Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τομέας Τεχνολογίας, Πληροφορικής και Υπολογιστών
Εργαστήριο Ευφυών Συστημάτων, Περιεχομένου και Αλληλεπίδρασης

Πρόβλεψη της εξέλιξης της νόσου Covid-19 ως χρονοσειρά με τη χρήση μηχανικής μάθησης

Διπλωματική Εργασία

του

Άδμητου-Ραφαήλ
Α. Πασσαδάκη

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Συνεπιβλέπουσα: Παρασκευή Τζούβελη
Εργαστηριακό και Διδακτικό Προσωπικό Ε.Μ.Π.

Αθήνα, Ιούνιος 2021

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας, Πληροφορικής και Υπολογιστών
Εργαστήριο Ευφυών Συστημάτων, Περιεχομένου και Αλληλεπίδρασης

Πρόβλεψη της εξέλιξης της νόσου Covid-19 ως χρονοσειρά με τη χρήση μηχανικής μάθησης

Διπλωματική Εργασία

του

Άδμητου-Ραφαήλ
Α. Πασσαδάκη

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την $7^{η}$ Ιουνίου 2021

.....	.....	.....
Στέφανος Κόλλιας Καθηγητής Ε.Μ.Π.	Ανδρέας Γ. Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Γεώργιος Στάμου Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2021

.....
Άδμητος-Ραφαήλ Πασσαδάκης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών, Ε.Μ.Π.

Ευχαριστίες

Η ολοκλήρωση της παρούσας Διπλωματικής Εργασίας σηματοδοτεί το τέλος των προπτυχιακών μου σπουδών. Για αυτό το ταξίδι αισθάνομαι υποχρέωση να ευχαριστήσω μια σειρά ανθρώπων που στάθηκαν δίπλα μου όλον αυτόν τον καιρό.

Για αρχή, θα ηθέλα να ευχαριστήσω ιδιαίτερα τον καθηγητή μου κ. Στέφανο Κόλλια, που μου έδωσε την ευκαιρία να εκπονήσω την διπλωματική μου στο Εργαστήριο Ευφυών Συστημάτων αλλά και τους κ. Ανδρέα - Γεώργιο Σταφυλοπάτη και Γέωργιο , που μου έκαναν την τιμή να βρίσκονται στην τριμελή εξεταστική επιτροπή. Θα ήθελα επίσης, βαθύτατα να ευχαριστήσω την συνεπιβλέπουσα κ. Παρασκευή Τζούβελη, χωρίς την βοήθεια, την καθοδήγηση και την συμβολή της, η παρούσα εργασία δεν θα μπορούσε να ολοκληρωθεί. Οι ιδέες της και η αμέριστη συμπαράσταση της σε όλα τα στάδια της εργασίας στάθηκαν καθοριστικές.

Ένα τόσο μακρύ ταξίδι, όσο οι προπτυχιακές σπουδές, δεν θα μπορούσαν να ολοκληρωθούν χωρίς την συνεχή συνεργασία, ανταλλαγή ιδεών αλλά πάνω από όλα την παρέα των φίλων μου. Μαζί με αυτούς περάσαμε αξέχαστες στιγμές τα τελευταία 5 χρόνια, τόσο εντός όσο και εκτός σχολής. Προσωπικά ευχαριστώ τους Κλεάνθη Α., Άγγελο Α., Βασιλική Β., Αναστάση Β., Γιάννη Θ., Γιώργο Κ., Παναγιώτη Κ., Άγγελο Κ., Βασίλη Μ. και Δαμιανό Ρ. για την στήριξή τους. Ταυτόχρονα, θα ήθελα να ευχαριστήσω τα αδέρφια μου (Κλειώ Π., Γιώργο Π. και Σοφία Τζ.), τους συγγενείς μου και ιδιαιτέρως τις θείες μου (Ευγγελία Π. και Δήμητρα Ρ.) όπου ο καθένας τους, με τον τρόπο του, με βοήθησε ξεχωριστά σε αυτό το ταξίδι.

Πάνω από όλα όμως οφείλω να ευχαριστήσω την μητέρα μου, Παναγιώτα, για τις αμέτρητες θυσίες που έκανε όλα αυτά τα χρόνια, για να φτάσω σε αυτό το σημείο σήμερα, μέσα από την αγάπη της, το ενδιαφέρον της, τον χρόνο της και την υποστήριξή της σε κάθε μου επιλογή. Γι' αυτό, θα ήθελα να αφιερώσω σε αυτήν, αλλά και στους εκλιπόντες πατέρα και παππού μου (Αθανάσιο Π. και Αθανάσιο Ρ.) την παρούσα εργασία.

Άδμητος-Ραφαήλ Α. Πασσαδάκης
Αθήνα, 17 Ιουνίου 2021

Περίληψη

Από τις αρχές του 2020 ολόκληρος ο κόσμος μαστίζεται από την πανδημία του κορωνοϊού, ενώ σε καθημερινή βάση ενδιαφερόμαστε για τα δεδομένα της νόσου αλλά και την μελλοντική της εξέλιξη. Σκοπός της παρούσας διπλωματικής εργασίας είναι η εξοικείωση με τα δεδομένα της Covid-19 και στη συνέχεια η χρήση μοντέλων μηχανικής μάθησης προκειμένου να προβλέψουμε την εξέλιξη της πανδημίας ως προς τα κρούσματα και τους θανάτους. Το πρόβλημα αυτό, αντιμετωπίζεται πρωτοποριακώς, ως πρόβλημα χρονοσειρών και έτσι υπάγεται στα προβλήματα πρόβλεψης χρονοσειρών. Τα προβλήματα αυτά είναι ευρέως διαδεδομένα σε πολλούς τομείς της Επιστήμης των Δεδομένων.

Πιο συγκεκριμένα, η εργασία μας χωρίζεται σε 2 κεντρικούς άξονες. Πρώτα, γνωρίζουμε μέσα από μαθηματικές σχέσεις τα δεδομένα αυτής της ασθένειας και έπειτα τα αποτυπώνουμε με την βοήθεια στατιστικών απεικονίσεων. Στο δεύτερο και εκτενέστερο σκέλος, στοχεύουμε στην γνωριμία και βασική μαθηματική θεμελίωση ειδικών μοντέλων μηχανικής μάθησης, των νευρωνικών δικτύων. Εξ αυτών, περιγράφουμε αναλυτικά τα επαναληπτικά και συνελικτικά νευρωνικά δίκτυα (RNNs και CNNs), τις ζεύξεις αυτών και τις δομές κωδικοποιητών-αποκωδικοποιητών με μηχανισμό Προσοχής.

Στη συνέχεια, αφού γνωρίσουμε και προτείνουμε διάφορες μεθόδους πρόβλεψης χρονικά ακολουθιακών δεδομένων, χρησιμοποιούμε αυτές, σε συνδυασμό με τα νευρωνικά δίκτυα για να προβλέψουμε τα επικείμενα κρούσματα και θανάτους της νόσου Covid-19 για διάφορες χώρες αλλά και παγκόσμια. Επιπρόσθετα, πραγματοποιούμε πειράματα κατά τα οποία τα μοντέλα μας χρησιμοποιώντας μηχανισμούς ανατροφοδότησης προβλέπουν την εξέλιξη της νόσου πέρα από το χρονικό ορίζοντα του συνόλου δεδομένων αλλά και κατασκευάζουμε δίκτυα αποκαλούμενα ως Autoencoders που μας δίνουν τη δυνατότητα ανίχνευσης ακραίων τιμών στα δεδομένα (πιθανά σημεία απότομης έξαρσης/συρρίκνωσης της πανδημιάς). Τέλος, προτείνουμε μια μέθοδο, όπου μέσα από την αποτελέσματικότητα των μοντέλων, μπορούμε να εξάγουμε έως ένα βαθμό, ορισμένα γεωγραφικά χαρακτηριστικά και συμπεράσματα ως προς την εικόνα της πανδημίας ανά τις χώρες της υφηλίου.

Λέξεις Κλειδιά: Κρούσματα/Θάνατοι Covid-19, Πρόβλεψη Χρονοσειρών, Μηχανική Μάθηση, Νευρωνικά Δίκτυα, Επαναληπτικά Νευρωνικά Δίκτυα, Μηχανισμός Προσοχής, Μηχανισμός Ανατροφοδότησης, Autoencoders, Γεωγραφικά Χαρακτηριστικά.

Abstract

Since the beginning of 2020 the whole world has been plagued by the coronavirus pandemic, while on a daily basis we are interested in the data of the disease and its future development. The purpose of this diploma thesis is the familiarization with Covid-19 data and then the usage of machine learning models to predict the evolution of the pandemic in terms of cases and deaths. This problem is being treated innovatively as a time series problem and thus, can be regarded as a time series forecasting problem. These problems are well known in many areas of Data Science.

More specifically, our work is divided into 2 main axes. First, we learn through mathematical formulas the data of this disease and then we illustrate them via statistical plots and graphs. In the second and more extensive part, we aim at the acquaintance and basic mathematical foundation of specific models of machine learning, neural networks. Out of them, we describe in detail the recurrent and convolutional neural networks (RNNs and CNNs), their couplings and Encoder-Decoder architectures with an Attention mechanism.

Then, after the introduction and proposal of various methods of predicting temporal sequence data, we use these, in conjunction with neural networks in order to predict the upcoming cases and deaths of Covid-19 disease for different countries as well as worldwide. Furthermore, we conduct experiments in which our models use rolling update (feedback) mechanisms to predict the evolution of the disease beyond the time horizon of the dataset and we also construct special networks such as Autoencoders that are enabled to detect extreme values (anomalies) in the data (possible signs of sudden outbreak/shrinking of the pandemic). Lastly, we propose a method where, through the efficacy of the models, we can draw to some extent, certain geographical characteristics and inferences as to the image of the pandemic across the countries of the world.

Keywords: Cases/Deaths of Covid-19, Time Series Forecasting, Machine Learning, Neural Networks, Recurrent Neural Networks, Attention Mechanism, Rolling Update Mechanism, Autoencoders, Geographical Features.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

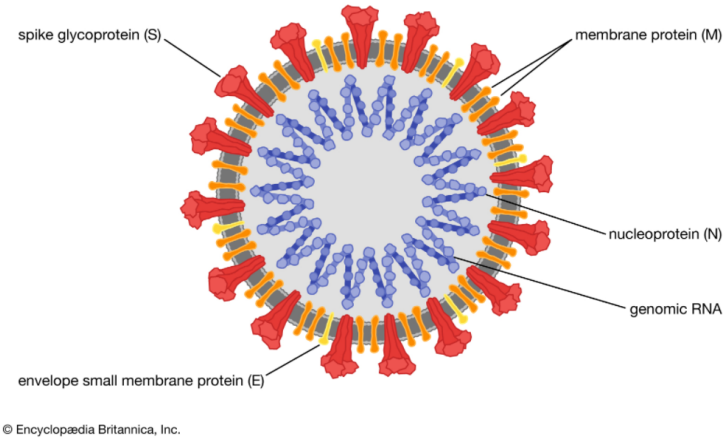
Εισαγωγή

1.1 Πληροφορίες για την ασθένεια COVID-19

Η πανδημία COVID-19, γνωστή ως πανδημία κορονοϊού, είναι μια νόσος σοβαρή ως θανατηφόρα, που προκαλείται από το οξύ αναπνευστικό σύνδρομο coronovarius 2 ή SARS-COV-2. Αυτή η ασθένεια εντοπίστηκε για πρώτη φορά τον Δεκέμβριο του 2019 στη Γουχάν της Κίνας, όταν το Εθνικό Γραφείο του ΠΟΥ στην Κίνα ενημερώθηκε για πνευμονία άγνωστης αιτίας [1]. Το ξέσπασμα ξεκίνησε στη Γουχάν τον Ιανουάριο του 2020 , με τις κινεζικές αρχές να αναφέρουν την πρώτη περίπτωση κρούσματος στις 4-1-2020 [2]. Αργότερα τον Μάρτιο του 2020, έχοντας εξαπλωθεί σε όλο τον κόσμο, ο Παγκόσμιος Οργανισμός Υγείας (ΠΟΥ) ανακήρυξε την ασθένεια Covid-19, ως πανδημία [3]. Ο πρώτος αναφερόμενος θάνατος από την Covid-19 ήταν επίσης στη Γουχάν στις 11-1-2020 [4].Έως και τον Μάρτιο του 2021, έχουν επιβεβαιωθεί περισσότερα από 120 εκατομμύρια κρούσματα (δηλαδή περίπου το 2% του παγκόσμιου πληθυσμού), με περισσότερους από 2.5 εκατομμύρια θανάτους να αποδίδονται στην COVID-19.

Κατά το πρώτο εξάμηνο του 2020, ολόκληρος ο κόσμος αντιμετώπισε πρωτοφανείς καταστάσεις όταν οι περισσότερες χώρες οδηγήθηκαν σε ολικό ή μερικό κλείσιμο (lockdown) αλλά και σε καθολική απαγόρευση των διεθνών πτήσεων προκειμένου να προστατεύσουν τους πολίτες και τα συστήματα υγείας τους. Δυστυχώς, πολλά συστήματα υγείας κατέρρευσαν, συμπεριλαμβανομένων της Ιταλίας, της Ισπανίας και ορισμένων πολιτειών των ΗΠΑ [5][6][7] υπό την τεράστια πίεση των αμέτρητων ασθενών που χρειάζονταν νοσηλεία και μονάδες εντατικής θεραπείας με αποτέλεσμα χιλιάδες θανάτους. Από το καλοκαίρι του 2020 και μετά, όταν άρχισε να εξαπλώνεται το δεύτερο κύμα του ιού, οι κυβερνήσεις προσπάθησαν σιγά-σιγά να αποφύγουν τα lockdown λόγω του τεράστιου αντικτύπου στην οικονομία τους. Εφαρμόστηκαν διάφορα μέτρα για να αποφευχθεί η υπερβολική μετάδοση του ιού, όπως περιορισμοί ταξιδιών, καραντίνες, αναβολές και ακυρώσεις εκδηλώσεων, κοινωνικές αποστάσεις και αποφυγή συνωστισμού. Ωστόσο, αυτά τα μέτρα δεν μπόρεσαν να κρατήσουν τον αριθμό των κρουσμάτων σε ικανοποιητικό επίπεδο για μεγάλο χρονικό διάστημα. Γύρω στον Οκτώβριο του 2020, ένα μεγάλο μέρος της Ευρώπης επέβαλε για άλλη μία φορά διάφορα περιοριστικά μέτρα, συμπεριλαμβανομένου και lockdown [8], για να σταματήσει (ή να μετριαστεί) η διασπορά

του κορονοϊού. Παρ' όλα αυτά, πολλές χώρες, συμπεριλαμβανομένης και της Ελλάδας, επλήγησαν βαρύτατα από το 2ο κύμα της πανδημίας κατά τις αρχές του χειμώνα του 2020 μετρώντας πολυάριθμες απώλειες σε ανθρώπινες ζωές [9].
Ύφεση της εξάπλωσης του κορονοϊού του δεύτερου κύματος παρατηρήθηκε την περίοδο των Χριστουγέννων του 2020. Όμως στα τέλη Ιανουαρίου 2021 άρχισε να κορυφώνεται το τρίτο κύμα της πανδημίας, με τον ιό να παρουσιάζει μεταλλάξεις [10], να γίνεται πιο μεταδότικός, αλλά παράλληλα, ξεκίνησε παγκόσμια και ο εμβολιασμός κατά της νόσου [11]. Επιδίωξη των κυβερνήσεων παγκόσμια είναι η αναχαίτηση του ιού, η θωράκιση του πληθυσμού με το εμβόλιο και η επιστροφή στην κανονικότητα.



Εικόνα 1.1: Αναπαράσταση του αναπνευστικού ιού SARS-COV-2 [120]

Εικόνα 1.2: Η πόλη της Αθήνας σε καθεστώς lockdown τον Απρίλιο 2020 [121]

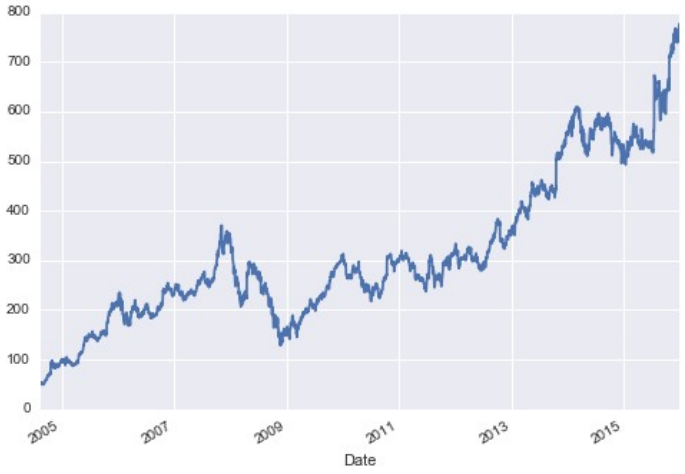
Εικόνα 1.3: Η ασφυκτική πίεση στο σύστημα υγείας, στο Πέργκαμο της Ιταλίας[122]

Όσον αφορά την ίδια την ασθένεια, τα πιο κοινά συμπτώματα περιλαμβάνουν πυρετό, βήχα, κόπωση, δυσκολίες στην αναπνοή και απώλεια όσφρησης-γεύσης. Σε πιο σοβαρές καταστάσεις της νόσου, οι επιπλοκές μπορεί να περιλαμβάνουν πνευμονία και οξεία αναπνευστική λοίμωξη. Η περίοδος επώασης είναι συνήθως περίπου πέντε (5) ημέρες, αλλά μπορεί να κυμαίνεται από μία (1) έως δεκατέσσερις (14) ημέρες [12]. Μέχρι σήμερα, υπάρχουν διάφορα εγκεκριμένα εμβόλια κατά της νόσου Covid-19 (δηλαδή έχουν εγκριθεί και γίνει αποδεκτά παγκοσμίως [13]. Ταυτόχρονα, γίνονται πολλές προσπάθειες, διεθνώς, για την διεύρυνση του αριθμού κατάλληλων εμβολίων που θα καλύπτουν και τις μεταλλάξεις του ιού. Επιπλέον, καταβάλλονται παγκόσμια τεράστιες προσπάθειες από τους ερευνητές για την εξεύρεση καταλλήλων θεραπειών για τη νόσο Covid-19, μια προσπάθεια που φαίνεται να αποδίδει (μονοκλωνικά αντισώματα-αναρρωτικό πλάσμα) σε συνδυασμό με άλλες αντιϊκές θεραπείες [14]. Ο αγώνας για τη θεραπεία ,τα εμβόλια ,τον εμβολιασμό του πληθυσμού και τον περιορισμό του ιού συνεχίζονται ακατάπαυστα από τις κυβερνήσεις και την επιστημονική κοινότητα σε όλο τον πλανήτη.

1.2 Χρονικά Ακολουθιακά Δεδομένα

Τους τελευταίους δεκατέσσερις (14) μήνες (κατά τη συγγραφή της εργασίας) οι ζωές της πλειόνοτητας των ανθρώπων ανά την υφήλιο άλλαξαν. Στην καθημερινότητα των περισσοτέρων από εμάς μπήκαν ερώτησεις όπως: 'Πόσα κρούσματα έχουμε σήμερα;', 'Πόσα κρούσματα θα έχουμε τις επόμενες μέρες;', 'Πόσοι άνθρωποι έχασαν τη ζωή τους;', 'Πόσες νοσηλείες αναμένεται να υπάρχουν τις επομένες μέρες στα νοσοκομεία;' και πολλές άλλες. Οι απαντήσεις σε αυτές τις ερώτησεις καθόριζαν και τη ζωή της επόμενης ημέρας. Τα καθημερινά και τα αθροιστικά κρούσματα, οι καθημερινοί και οι αθροιστικοί θάνατοι, τα διεξαγώμενα τεστ, οι νοσηλείες σε απλές κλίνες ή κλίνες ΜΕΘ-Covid κ.α αποτελούν τα *δεδομένα* της πανδημίας. Τα δεδομένα αυτά τα αποκαλούμε *χρονοσειρές* (time series). Χρονοσειρά λοιπόν είναι ένα σύνολο σημε-

ίων από δεδομένα που καταχωρούνται σε ένα ευρετήριο ή/και αναπαρίστανται γραφικά με χρονική σειρά. Συνήθως, μία χρονοσειρά είναι μία ακολουθία που λαμβάνεται διαδοχικά σε εξίσου κατανεμημένα χρονικά σημεία. Συνεπώς, μιλάμε για μια ακολουθία διακριτών χρονικών δεδομένων. Η εξέλιξη των δεδομένων της νόσου Covid-19 τους τελευταίους δεκατέσσερις (14) μήνες αποτελεί ακριβώς μία τέτοια χρονόσειρα. Άλλα παραδείγματα χρονοσειρών είναι τα ύψη των ωκεάνιων παλιρροιών, ο αριθμός των ηλιακών κηλίδων και η ημερήσια τιμή κλεισίματος (close price) του δείκτη Dow Jones. Οι χρονοσειρές συναντιούνται συχνά στην στατιστική, επεξεργασία σημάτων, αναγνώριση προτύπων, οικονομετρική ανάλυση, οικονομική μαθηματική ανάλυση, πρόγνωση καιρού, πρόβλεψη σεισμών, ηλεκτροεγκεφαλογραφία, μηχανική ελέγχου, αστρονομία, μηχανική επικοινωνιών και σε μεγάλο βαθμό σε οποιονδήποτε τομέα εφαρμοσμένης επιστήμης και μηχανικής που περιλαμβάνει χρονικές μετρήσεις. Στην εικόνα 1.4 παρουσιάζουμε μία τυπική χρονοσειρά που μας δείχνει πως κινήθηκε η τιμή κλεισίματος της μετοχής της Google από το 2004 έως και το 2016.



Εικόνα 1.4: Τυπική χρονοσειρά που δείχνει το κλείσιμο της τιμής μετοχής της Google[123]

Η *ανάλυση χρονοσειρών* περιλαμβάνει μεθόδους ανάλυσης δεδομένων χρονοσειρών για την εξαγωγή σημαντικών στατιστικών και άλλων χαρακτηριστικών μεγεθών. Η *πρόβλεψη χρονοσειρών* (time series forecasting) είναι η χρήση μοντέλων και εργαλείων για την πρόβλεψη μελλοντικών τιμών σε χρονοσειρές με βάση τιμές που έχουν παρατηρηθεί προηγουμένως. Στην ανάλυση χρονοσειρών, μπορούμε να διακρίνουμε δύο (2) βασικές μεθόδους ανάλυσης: μέθοδοι αναφερόμενες στο πεδίο της συχνότητας και μέθοδοι αναφερόμενες στο πεδίο του χρόνου. Οι πρώτες περιλαμβάνουν φασματική ανάλυση (spectral analysis) και ανάλυση κύματος (wavelet analysis), ενώ οι δεύτερες περιλαμβάνουν ανάλυση αυτοσυσχέτισης και ετεροσυσχέτισης (auto-correlation and cross-correlation analysis).
Ένας ακόμα διαχωρισμός γίνεται με βάση το αν τα δεδομένα παρουσιάζουν γραμμική συμπεριφορά ή όχι. Σε αρκετούς από τους κλάδους που αναφέρθηκαν πρωταρχικός στόχος είναι η πρόβλεψη. Σε ορισμένους, όπως οι τηλεπικοινωνίες, η επεξεργασία σήματος κ.α

στόχος είναι η ανίχνευση σήματος. Σε άλλες εφαρμογές όπως η εξόρυξη γνώσης από δεδομένα (data mining), η μηχανική μάθηση (machine learning) και η αναγνώριση προτύπων ο σκοπός είναι η ομαδοποίηση ή η ταξινόμηση δεδομένων (clustering and classification), η ανίχνευση ακραίων σημείων αλλά και η πρόβλεψη [15].

Μία από τις σημαντικότερες μεθόδους στην στατιστική μοντελοποίηση είναι η *παλινδρομική ανάλυση* (regression analysis) που αποτελεί ένα σύνολο στατιστικών διαδικασιών για την εκτίμηση των σχέσεων μεταξύ μίας εξαρτημένης μεταβλητής (συχνά αποκαλείται «μεταβλητή αποτελέσματος») και μίας ή περισσότερων ανεξάρτητων μεταβλητών (συχνά αποκαλούνται «χαρακτηριστικά»). Η πιο γνωστή μέθοδος παλινδρόμησης είναι η *γραμμική παλινδρόμηση* (linear regression) στην οποία κάποιος προσπαθεί να βρει μία ευθεία ή ένα σύνολο ευθειών που να ταιριάζουν καλύτερα στα δεδομένα του. Σε αρκετές περιπτώσεις ακολουθείται και η γνωστή *μέθοδος ελαχίστων τετραγώνων* κατά την οποία, η ευθεία που ψάχνουμε ώστε να ταιριάζει καλύτερα στα δεδομένα ελαχιστοποιεί το άθροισμα τετραγώνων της απόστασης μεταξύ των δεδομένων και της ίδιας της ευθείας.

Υπάρχουν και άλλες μέθοδοι παλινδρόμησης, οι οποίες συνήθως είναι και αποτελεσματικότερες, καθώς τα δεδομένα στην πλειονότητα των περιπτώσεων παρουσιάζουν μη γραμμική συμπεριφορά. Κάποιες από αυτές είναι η *πολυωνυμική παλινδρόμηση* (polynomial regression), *παλινδρόμηση με διανύσματα υποστήριξης* (support vector regression), *Μπεϋζιανή παλινδρόμηση, γραμμική και μη*, (Bayesian regression) και άλλες. Οι μέθοδοι παλινδρόμησης χρησιμοποιούνται κυρίως για δύο εννοιολογικά διαφορετικούς σκοπούς. Πρώτον, η παλινδρόμηση εφαρμόζεται ευρέως στην πρόβλεψη χρονοσειρών, όπου η χρήση της έχει σημαντική επικάλυψη με το πεδίο της μηχανικής μάθησης. Δεύτερον, σε ορισμένες περιπτώσεις η ανάλυση παλινδρόμησης μπορεί να χρησιμοποιηθεί για να συναχθούν αιτιώδεις σχέσεις μεταξύ της ανεξάρτητης και εξαρτημένης μεταβλητής [16]. Η μέθοδος των ελαχίστων τετραγώνων που αναφέρθηκε νωρίτερα ανήκει στην ευρύτερη κατηγορία του *ταιριάγματος καμπύλης* (curve fitting) κατά την οποία ψάχνουμε μία καμπύλη που ταιριάζει καλύτερα στα δεδομένα μας και αποτελεί άλλον έναν τρόπο ανάλυσης/πρόβλεψης χρονοσειρών.

Τις τελευταίες δεκαετίες τα πιο διαδεδομένα μοντέλα για την αναλύση και πρόβλεψη χρονοσειρών είναι τα μοντέλα *αυτόματης παλινδρόμησης* (auto-regressive-AR), τα μοντέλα *ενσωμάτωσης* (integrated-I) και τα μοντέλα *κινούμενου μέσου όρου* (moving average-MA). Συνδυασμός αυτών των ειδών μοντέλων έχουν καταγράψει μεγάλες επιτυχίες στον χώρο της πρόβλεψης χρονοσειρών. Τέτοια μοντέλα είναι τα ARMA (auto-regressive moving average), ARIMA (auto-regressive integrated moving average) και ARFIMA (auto-regressive fractionally integrated moving average) [17][18]. Ακόμα πιο σύγχρονος τρόπος αντιμετώπισης του προβλήματος αυτού, και που έχει ξεκινήσει τα τελευταία χρόνια, είναι η χρήση μοντέλων μηχανικής μάθησης. Στον άξονα αυτόν κινήθηκε και η παρούσα εργάσια. Αργότερα, στα Κεφάλαια 3 και 4 θα γνωρίσουμε πως είναι δυνατόν να χρησιμοποιήσουμε τέτοια μοντέλα προκειμένου να προβλέψουμε χρονοσειρές.

1.3 Στόχος και Δομή της Εργασίας

Το αντικείμενο της παρούσας εργασίας χωρίζεται σε δύο (2) διακριτά μέρη, την ανάλυση αλλά και την πρόβλεψη χρονοσειρών που αφόρουν την νόσο Covid-19. Το δεύτερο κομ-

μάτι είναι εκεί που η εργασία εστιάζει περισσότερο. Όπως αναφέρθηκε τα δεδομένα της νόσου είναι οι χρονοσειρές που θα χρησιμοποιήσουμε και κυρίως τα κρούσματα και οι θάνατοι που αποτελούν τις δύο (2) πιο σημαντικές μεταβλητές της πανδημίας. Στην αρχή, στόχος είναι η εισαγωγή δεδομένων για την νόσο Covid-19 από εγκεκριμένες πηγές όπως ο Παγκόσμιος Οργανισμός Υγείας (ΠΟΥ) [19], ο Οργανισμός Ηνωμένων Εθνών (ΟΗΕ) [20], η ιστοσελίδα Our World in Data (OWID) [21] και η ιστοσελίδα Covid Tracking Project[22] που αφορά κυριώς τις Ηνωμένες Πολιτείες Αμερικής (ΗΠΑ), την χώρα που έχει πληγεί όσο καμία αλλή από τον κορωνοϊό. Στη συνέχεια, εφαρμόζοντας κάποιες κλασσικές μεθόδους διαχείρισης δεδομένων αλλά και με την βοήθεια της προγραμματιστικής γλώσσας Python σκοπός είναι η οπτικοποίηση αυτών των δεδομένων σε απλά διαγράμματα, γραφήματα πίτας, ραβδογράμματα, ιστογράμματα, γεωγραφικοί 'θερμοχάρτες' (Geographic Heat Maps) κ.α αλλά και εξαγωγή διαφόρων χρήσιμων στατιστικών μετρικών και συμπερασμάτων που μας βοηθάνε στην καλύτερη κατανόηση της εξάπλωσης και σοβαρότητας της πανδημίας. Σε αυτές τις μετρικές ανήκουν τα κρούσματα/θάνατοι ανά εκατομμύριο, τα ποσοστά θνητότητας (Case Fatality Rate-CFR [23]) ανά χώρα, πιθανότητα εισαγωγής σε ΜΕΘ κ.λ.π

Στο δεύτερο και σημαντικότερο μέρος, ο σκοπός της εργασίας μετατοπίζεται στην πρόβλεψη της εξέλιξης της νόσου με την χρήση και τη βοήθεια μοντέλων μηχανικής μάθησης. Συγκεριμένα, κάνουμε χρήση των επαναληπτικών νευρωνικών δικτύων που διαθέτουν την ικανότητα να διατηρούν (σαν μνήμη) πληροφορία που έχουν γνωρίσει νωρίτερα κατά την εκπαίδευσή τους προκειμένου να κάνουν μία πρόβλεψη για το μέλλον. Αυτό το είδος νευρωνικού δικτύου, λόγω των συγκεκριμένων δυνατοτήτων αποτελεί τη ραχοκοκαλιά της εργασίας πάνω στην οποία διάφορα είδη μοντέλων δοκιμάστηκαν για το σκοπό της πρόβλεψης.
Χρησιμοποιούμε επίσης, κάποιες στοχευμένες τεχνικές, γνωστές στον κόσμο της πρόβλεψης χρονοσειρών, (μέθοδοι παραθυροποίησης, περιοδικών διαφορών κ.α.) έτσι ώστε να τροφοδοτήσουμε καταλλήλως τα δίκτυά μας αλλά και να τα ενισχύσουμε κατά την προσπάθεια πρόβλεψης. Ταυτόχρονα, γίνεται χρήση πληθώρας αρχιτεκτονικών και τεχνικών για να εξετάσουμε (και πιθανώς να βελτιώσουμε) την αποτελεσματικότητά μας κατά την πρόβλεψη και για αυτό εισάγουμε τα συνελικτικά νευρωνικά δίκτυα, την έννοια των αμφίδρομων νευρωνικών δικτύων αλλά και τον μηχανισμό της προσοχής. Θα πρέπει να τονιστεί ότι αυτές οι επιπλέον προσθήκες αποτελούν επεκτάσεις των επαναληπτικών νευρωνικών δικτύων υπό την έννοια ότι δεν χρησιμοποιούνται αυτόνομα.
Με την βοήθεια διάφορων βιβλιοθηκών κατασκευάζονται αρκετές εκδοχές (απλών ή ενισχυμένων) τέτοιων δικτύων που αξιολογούνται μέσα από πειραματική διαδικασία. Σημειώνοντας πειραματική διαδικασία εννοούμε την εκπαίδευση και την πρόβλεψη πάνω σε διάφορα είδη καμπυλών (κρουσμάτων ή θανάτων) από διάφορες χώρες του κόσμου ή/και την υφήλιο ολόκληρη. Κάτα την πρόβλεψη χρησιμοποιούμε διάφορες μετρικές αποτελεσματικότητας με τις οποίες ποσοστικοποιούμε και την απόδοση των μοντέλων μας. Αυτός είναι και ο απώτερος στόχος της εργασίας.
Επιπλέον, ειδικές υλοποιήσεις συνιστούν η κατασκευή πολυπλοκότερων αρχιτεκτονικών εφοδοδιασμένων με επιπλέον μηχανισμούς (Attention) που βοηθούν στην καλύτερη πρόβλεψη ή μοντέλων που με ανατροφοδότηση των προβλέψεών τους φεύγουν πέραν του συγκεκριμένου ορίζοντα δεδομένων που έχουμε εισάγει, η κατασκευή δικτύων (αποκαλούμενοι διεθνώς ως autoencoders) που εξετάζουν σημεία των δεδομένων μας που

ενδεχομένως να παρουσίαζουν κάποια ανωμαλία (εύρεση ακραίων τιμών) αλλά και η πρόταση και η εφαρμογή μιας μεθόδου όπου μέσα από την αποτελεσματικότητα των μοντέλων μπορούμε να εξάγουμε έως ένα βαθμό ορισμένα (γεωγραφικά) χαρακτηριστικά για την εικόνα της πανδημίας ανά την υφήλιο.

Η υπόλοιπη εργασία διαρθρώνεται ως εξής. Στο Κεφάλαιο 2 οπτικοποιούμε τα δεδομένα της Covid-19 και τα παρουσίαζουμε μέσα από τα γραφήματα και τα σχεδιαγράμματα που προαναφέραμε. Ο αναγνώστης σε αυτό το κεφάλαιο θα γνωρίσει για το πως έχει εξελιχθεί η πανδημία τον τελευταίο χρόνο μέσα από διάφορα στατιστικά στοιχεία. Στο Κεφάλαιο 3 κάνουμε μία εισαγωγή στην μηχανική μάθηση και τα νευρωνικά δίκτυα και γνωρίζουμε το απαραίτητο θεωρητικό υπόβαθρο των μοντέλων που χρησιμοποιήθηκαν. Ξεκινώντας από τη γραμμική παλινδρόμηση που γνωρίσαμε στην ενότητα 1.2 καταλήγουμε στις πιο σύνθετες και περίπλοκες δομές επαναληπτικών νευρωνικών δικτύων εφοδιασμένες με επιπρόθετους ενισχυτικούς μηχανισμούς.
Στο Κεφάλαιο 4 αποτυπώνεται η πρακτική εφαρμόγη των μοντέλων που αναλύθηκαν στο κεφάλαιο 3. Περιγράφουμε την υλοποιήση όλων των εμπλεκόμενων αρχιτεκτονικών της εργασίας και τις παρουσιάζουμε με τη βοήθεια της Python [24]. Επίσης, γίνεται η ανάλυση συγκεκριμένων τεχνικών που χρησιμοποιήθηκαν προκειμένου να βελτιώσουμε την απόδοση των μοντέλων. Σε συνέχεια, στο Κεφάλαιο 5 αναδεικνύονται τα αποτελέσματα των διαφόρων αρχιτεκτονικών και τεχνικών μέσα από κατάλληλες μετρικές, σχολιάζονται και έρχονται σε αντιδιαστολή με σκοπό την εύρεση του αποδότικότερου. Τέλος, στο Κεφάλαιο 6 καταγράφουμε τα βασικά συμπεράσματα της εργασίας καθώς και πιθανές μελλοντικές ερευνητικές επεκτάσεις σε ένα πρόβλημα που ούτως ή άλλως έχει την ιδιαιτερότητα ότι θα αποσχολεί την ερευνητική κοινότητα συνεχώς.

Κεφάλαιο 2

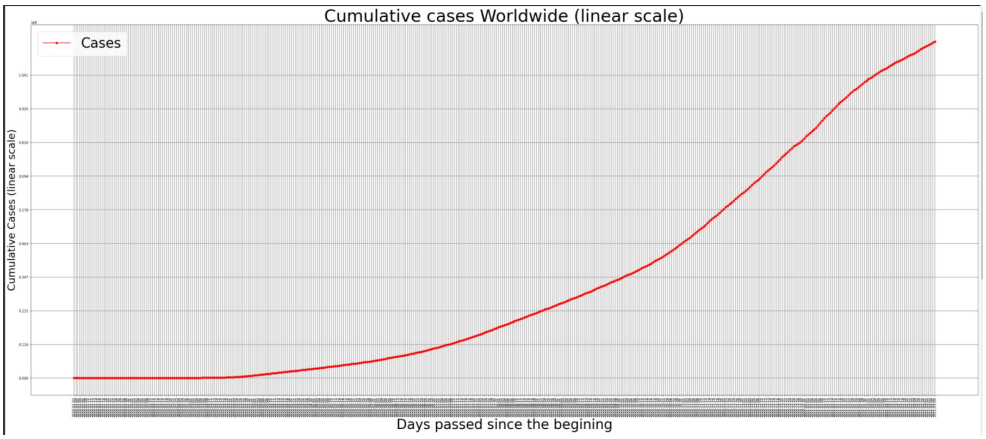
Οπτικοποίηση των δεδομένων της Covid-19

2.1 Η παγκόσμια εικόνα της νόσου

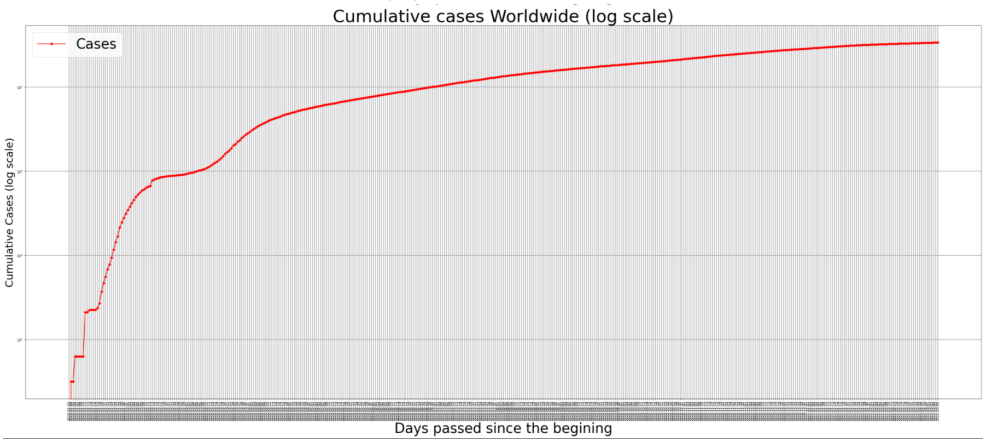
Από την έναρξη της πανδημίας, τον Ιανουάριο του 2020, έως και τον Μάρτιο του 2021 (κατά τη συγγραφή της εργασίας) έχουν νοσήσει περί τα 125 εκατομμύρια άνθρωποι από την ασθένεια Covid-19, ενώ έχουν καταλήξει περίπου 2.75 εκατομμύρια. Ταυτόχρονα, ξεκινώντας από τον Δεκέμβριο του 2020 εώς σήμερα, έχουν εμβολιαστεί περίπου 250 εκατομμύρια άνθρωποι, ήτοι το 3.5% του παγκόσμιου πληθυσμού. Στη συνέχεια, θα δοθεί μία πολύ σύντομη παρουσίαση της εικόνας που είχε η πανδημία σε παγκόσμιο επίπεδο κατά τον Μάρτιο του 2021. Θα πρέπει να τονιστεί σε αυτό το σημείο ότι τα δεδομένα της νόσου έχουν μορφή ημερήσιας συλλογής και κατά την εισαγωγή τους αφορούσαν την κάθε χώρα ξεχωριστά από την εναρκτήρια μέρα της πανδημίας στη συγκεκριμένη χώρα έως και την τελευταία ημέρα συλλογής.

Επειδή λοιπόν, η συλλογή των δεδομένων είχε ως βάση το σύνολο των χωρών, η αναγωγή τους σε παγκόσμια κλίμακα έγινε με την απλή πρόσθεση όλων των δεδομένων, όλων των χωρών καθ' όλη τη διάρκεια της πανδημίας. Αξίζει επίσης, να σημειωθεί ότι για λόγους ευκολίας διαχείρισης (όπως θα δούμε παρακάτω) και απεικόνισης των δεδομένων η χρονική διάρκεια της πανδημίας ορίστηκε για όλες τις χώρες αλλά και για την υφήλιο συνολικά ως το διάστημα από την ημέρα ανίχνευσης του 1ου κρούσματος στην Γιουχάν της Κίνας έως και την τελευταία μέρα συλλογής των δεδομένων.

Τα δεδομένα που απεικονίζουμε προέρχονται από τον ΠΟΥ (κρούσματα/θάνατοι) και τον ΟΗΕ (χρήση του πληθυσμού χωρών). Για την απεικόνιση τους έγινε χρήση των πολύ ισχυρών βιβλιοθηκών της Python, Matplotlib και Seaborn [25][26], οι οποίες προσφέρουν μεγάλη άνεση στην οπτικοποίηση δεδομένων. Στις εικόνες 2.1α′ και 2.1β′ βλέπουμε την αθροιστική καμπύλη των κρουσμάτων τόσο σε γραμμική όσο και σε λογαριθμική κλίμακα. Αντίστοιχα, στις εικόνες 2.2α′, 2.2β′ τις ίδιες καμπύλες για τους αθροιστικούς θανάτους. Παίρνοντας το κλάσμα των συνολικών θανάτων προς τα συνολικά κρούσματα για κάποια χρονική στιγμή έχουμε το λεγόμενο ποσοστό θνησιμότητας



(α′) Γραμμική κλίμακα



(β′) Λογαριθμική κλίμακα

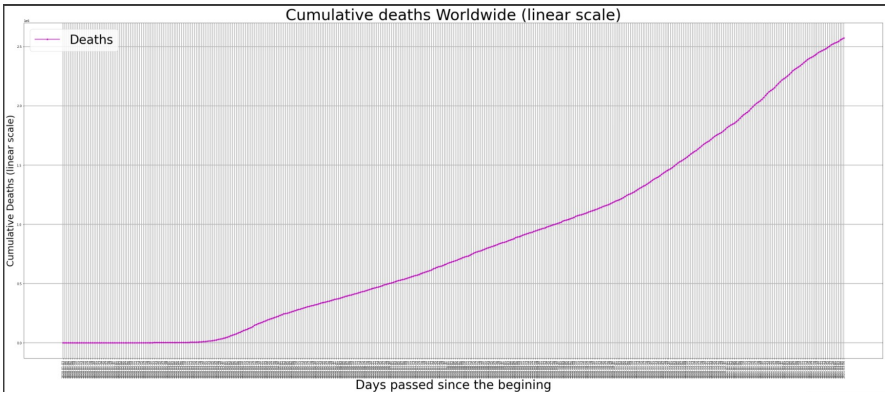
Εικόνα 2.1: Αθροιστικές καμπύλες κρουσμάτων

με βάση τα κρούσματα (Case Fatality Rate - CFR), δηλαδή:

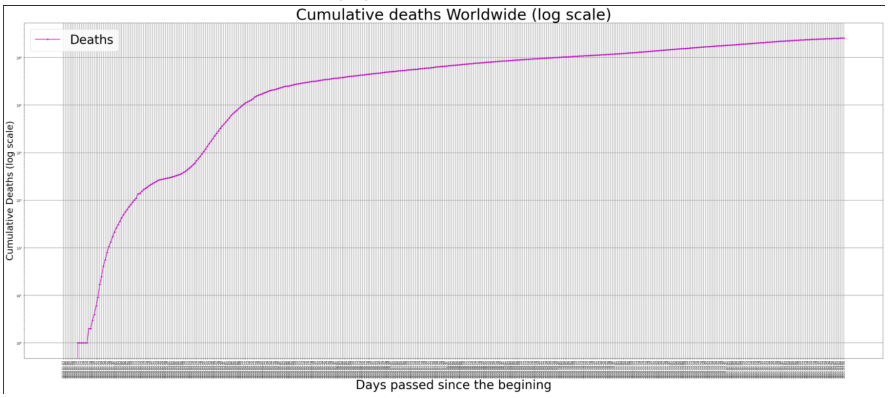
$$CFR = \frac{Total \quad Deaths}{Total \quad Cases} \tag{2.1.1}$$

Στην εικόνα 2.3 βλέπουμε το CFR της ασθένειας Covid-19 κατά τη διάρκεια του χρόνου. Βλέπουμε ότι κατά την αρχή της πανδημίας η θνησιμότητα της νόσου ξεπέρασε ακόμα και το **7%**, ενώ στη συνέχεια μειώθηκε φτάνοντας στο επίπεδο του **2.2%**

Περισσότερο ενδιαφέρον από τις αθροιστικές καμπύλες των κρουσμάτων και των θανάτων έχουν οι καμπύλες που αποτυπώνουν τα καθημερινά κρούσματα και τους καθημερινούς θανάτους, δίνοντας και καλύτερη εικόνα για την εξέλιξη της νόσου. Αυτές οι καμπύλες φαίνονται στα ραβδογράμματα των εικόνων 2.4α′,2.4β′. Επίσης, σε αυτές τις εικόνες βλέπουμε τον επταήμερο κυλιόμενο μέσο όρο των κρουσμάτων και των θανάτων. Τονίζουμε ότι ο επταήμερος κυλιόμενος μέσος όρος για μία παρατήρηση X μήκους N

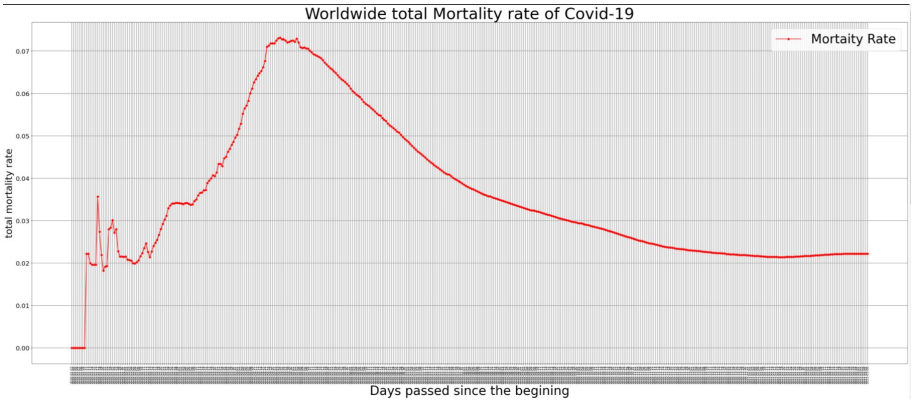


(α′) Γραμμική κλίμακα



(β′) Λογαριθμική κλίμακα

Εικόνα 2.2: Αθροιστικές καμπύλες θανάτων

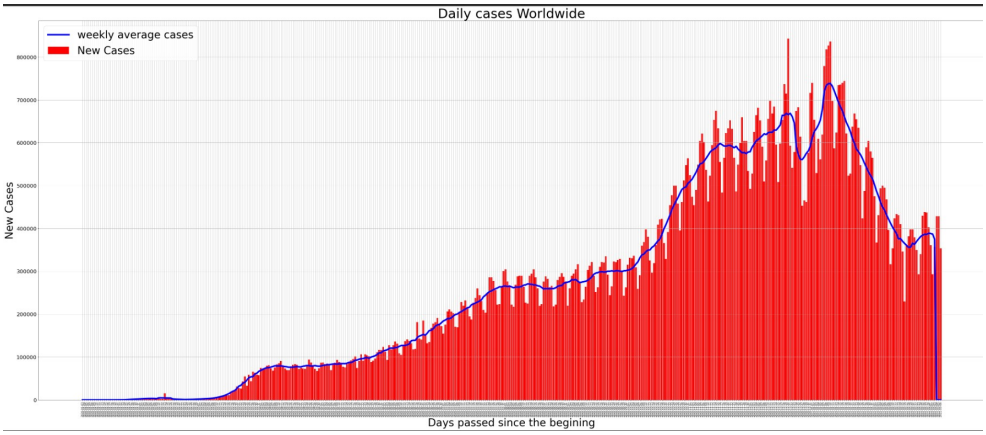


Εικόνα 2.3: Η θνησιμότητα της Covid-19 ως συνάρτηση του χρόνου (CFR)

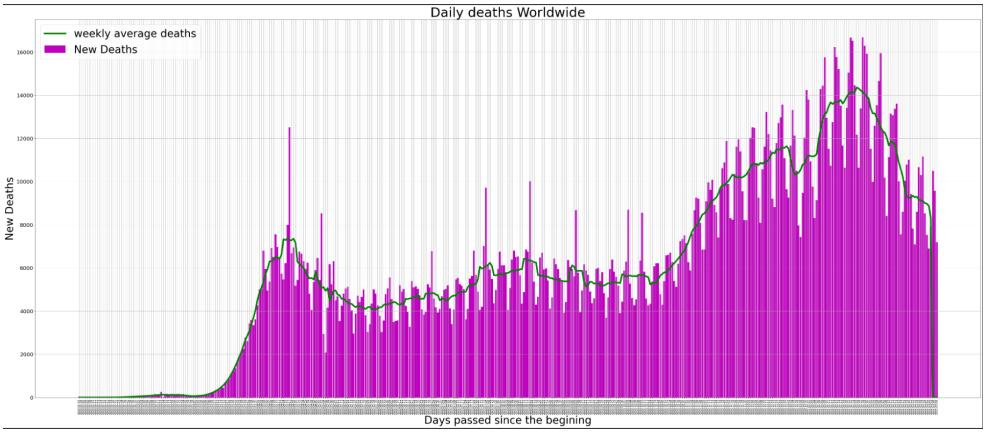
δίνεται από τον παρακάτω τύπο:

$$M_i = \frac{X_{i-3} + X_{i-2} + X_{i-1} + X_i + X_{i+1} + X_{i+2} + X_{i+3}}{7} \quad \text{για} \quad 3 \leq i \leq N-3 \tag{2.1.2}$$

Για $i = 0, 1, 2$ και για $i = N-2, N-1, N$ έχουμε $M_i = 0$



(α′) Καθημερινά Κρούσματα



(β′) Καθημερινοί Θάνατοι

Εικόνα 2.4: Καθημερινή αποτύπωση των δεδομένων της νόσου

Αξίζει να σταθούμε λίγο στις εικόνες 2.4α′,2.4β′. Όσον αφορά τα κρούσματα μπορούμε να δούμε ότι η υφήλιος 'χτυπήθηκε' από τρία (3) επιδημικά κύματα όπου χρονικά τοποθετούνται περίπου στον Μάρτιο, Ιούλιο και Νοέμβριο του 2020. Ευτυχώς, από την έναρξη του 2021 παρατηρείται μία σταδιακή υποχώρηση της νόσου. Αναφορικά με τους θανάτους, μπορούμε να διακρίνουμε δύο (2) απότομες αυξήσεις του επταήμερου κυλιόμενου μέσου όρου που αντιστοιχούν σε δύο (2) κύματα. Αυτά τοποθετούναι στις αρχές της πανδημίας και στον Νοέμβριο του 2020.

2.2 Διαχωρισμός ανά χώρα

Δυστυχώς, οι περισσότερες χώρες του κόσμου αντιμετώπισαν σοβαρό πρόβλημα σε επίπεδο υγείας και οικονομίας από τη λοίμωξη Covid-19. Όσον αφόρα τον τομέα της υγείας η χώρα με το μεγαλύτερο μακράν πλήγμα είναι οι ΗΠΑ, τόσο στον αριθμό κρουσμάτων όσο και σε αυτόν των θανάτων. Ακολουθούν διάφορες πολυπληθείς χώρες όπως η Ινδία, η Βραζίλια, η Ρωσία αλλά και χώρες της Ευρώπης. Στις εικόνες 2.5α′, 2.5β′ παρουσιάζουμε τις δέκα (10) πιο σφοδρά 'χτυπημένες' χώρες από τον ιό σε επίπεδο κρουσμάτων και θανάτων μέσα από δύο (2) γραφήματα πίτας με τους ακριβείς αριθμούς των κρουσμάτων και θανάτων για αυτές τις δέκα (10) χώρες καθώς και τις υπόλοιπες συγκεντρωτικά (που είναι εκτός δεκάδας). Επιπλέον, παρουσιάζεται η ποσοστιαία κατοχή της πίτας από κάθε χώρα (τα δεδομένα αφορούν έως και τις 8/3/2021).

Από τις εικόνες 2.5 μπορούμε να δούμε ότι κατά τους πρώτους δεκατέσσερις (14) μήνες πανδημίας τόσο ο συνολικός αριθμός κρουσμάτων όσο και ο αντίστοιχος αριθμός των θανάτων διαμοιράζεται κατά ποσοστό 65% και άνω στις πρώτες δέκα (10) χώρες και κάτα 35% σε όλες τις υπόλοιπες χώρες. Να τονιστεί βεβαίως, ότι αυτές οι δέκα (10) χώρες έχουν και ένα αρκετά μεγάλο μέρος του παγκόσμιου πληθυσμού (εξαιρείται η Κίνα). Επίσης, όσον αφορά τα κρούσματα μπορούμε να δούμε ότι οι ΗΠΑ από μόνες τους έχουν σχεδόν το 1/4 του συνολικού αριθμού κρουσμάτων και ακολουθούν η Ινδία και η Βραζιλία με σχεδόν 10%. Αναφορικά με τους θανάτους και πάλι οι ΗΠΑ κατέχουν από μόνες τους το 1/5 των συνολικών απωλειών ακολουθούμενες από τη Βραζιλία με 10%. Μία ιδιαίτερα χρήσιμη μεταβλητή προκειμένου να κατανοήσουμε καλύτερα τη διασπορά του ιού σε κάθε χώρα είναι οι περιπτώσεις της νόσου ανά εκατομμύριο πληθυσμού ($\frac{c}{m}$), όπου η αυξημένη τιμή αυτής της μεταβλητής για μία χώρα δείχνει μεγάλη διασπορά της νόσου στο εσωτερικό της. Αντίστοιχα ορίζεται και ο αριθμός των θανάτων ανά εκατομμύριο ($\frac{d}{m}$).

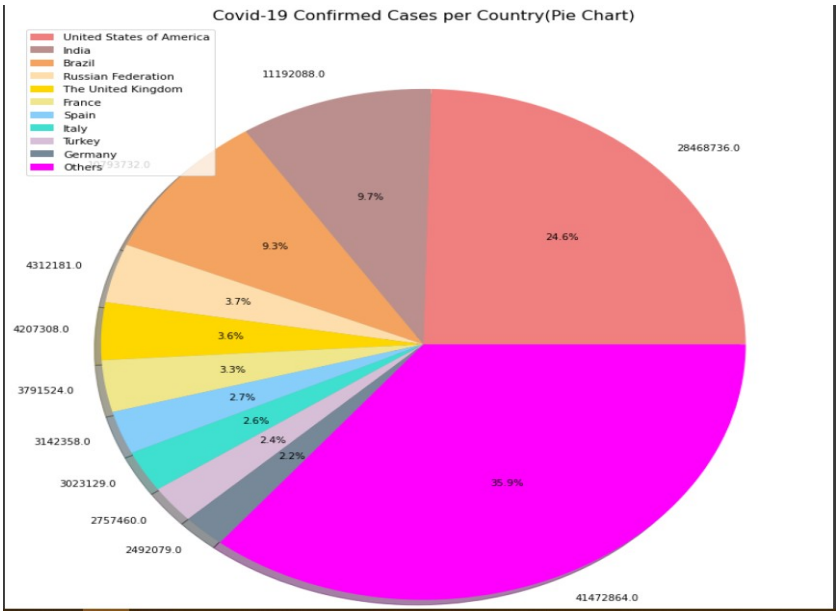
Στις εικόνες 2.6α′, 2.6β′, 2.6γ′ παρουσιάζονται οι χώρες που βρίσκονται στις πρώτες δέκα (10) θέσεις σε σχέση με αυτές τις μετρικές, αλλά και με βάση το ποσοστό θνησιμότητας. Αναφέρουμε επίσης, ότι μία ποσότητα, έστω κρούσματα (c), δίνεται ανά εκατομμύριο ($/m$) από τον τύπο:

$$\frac{c}{m} = \frac{Total \quad cases \quad in \quad the \quad country}{Country's \quad total \quad population} \times 10^6 \tag{2.2.1}$$

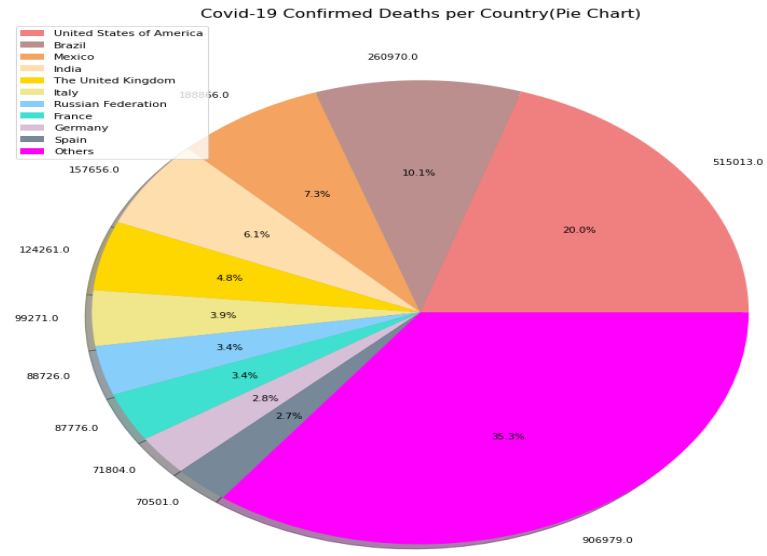
Η αντίστοιχη σχέση ισχύει και για τους θανάτους (d), ως:

$$\frac{d}{m} = \frac{Total \quad deaths \quad in \quad the \quad country}{Country's \quad total \quad population} \times 10^6 \tag{2.2.2}$$

Από την εικόνα 2.6α′ μπορούμε να δούμε ότι η Υεμένη, είναι η χώρα που έχει μακράν το υψηλότερο ποσοστό θνησιμότητας ξεπερνώντας το 25%! Όλες οι χώρες που ακολουθούν έχουν ποσοστό μικρότερο του 10%. Όσον αφορά τα κρούσματα ανά εκατομμύριο (εικόνα 2.6β′) παρατηρούμε ότι συνήθως μικρές πληθυσμιακά αλλά και σε έκταση χώρες, καταλαμβάνουν τις πρώτες θέσεις, χωρίς όμως να λείπουν οι εξαιρέσεις (ΗΠΑ). Επίσης, βλέπουμε ότι οι χώρες της Ευρώπης κυριαρχούν καθώς καταλαμβάνουν τις πρώτες οκτώ (8) θέσεις της δεκάδας. Τέλος, στην εικόνα 2.6γ′ βλέπουμε ότι οι ευρωπαϊκές χώρες καταγράφουν εξ' ολοκλήρου τη χειρότερη απόδοση αναφορικά με τους θανάτους ανά

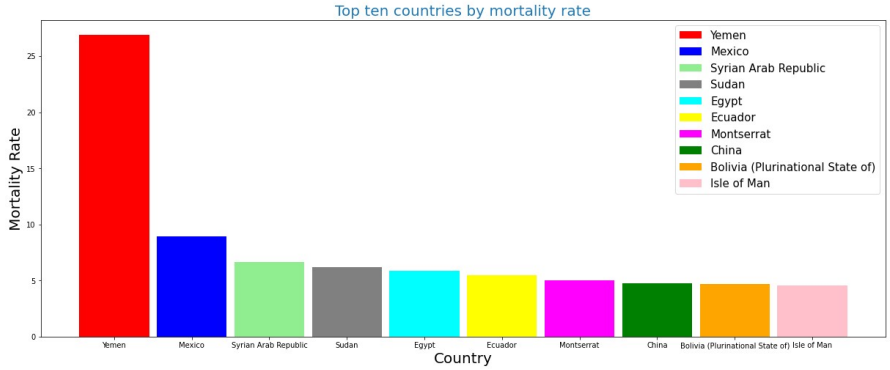


(α′) Οι δέκα (10) πρώτες χώρες ως προς τα κρούσματα

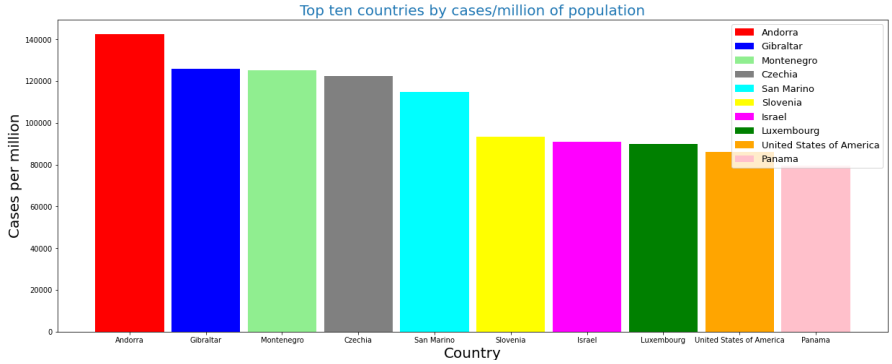


(β′) Οι δέκα (10) πρώτες χώρες ως προς τους θανάτους

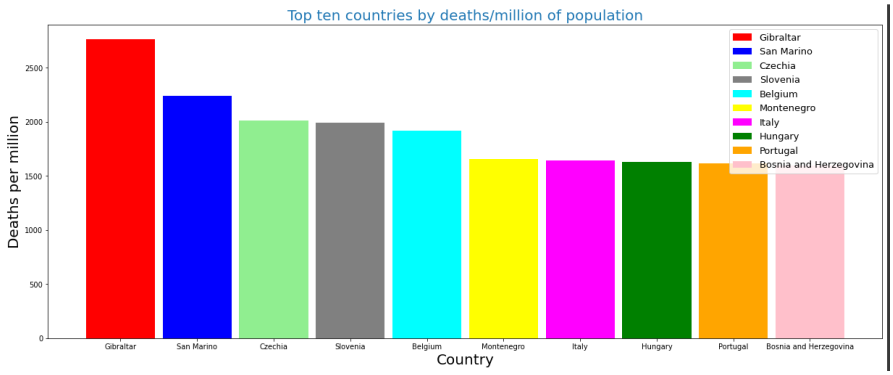
Εικόνα 2.5: Οι δέκα (10) πρώτες χώρες που επλήγησαν περισσότερο από την πανδημία



(α′) Οι δέκα (10) πρώτες χώρες ως προς το ποσοστό θνησιμότητας



(β′) Οι δέκα (10) πρώτες χώρες ως προς τα κρούσματα/εκατομμύριο



(γ′) Οι δέκα (10) πρώτες χώρες ως προς τους θανάτους/εκατομμύριο

Εικόνα 2.6: Οι δέκα (10) πρώτες χώρες αναφορικά με τους δείκτες CFR, $\frac{c}{m}$ και $\frac{d}{m}$

εκατομμύριο.
Στην εικόνα 2.7, ο πίνακας δείχνει αναλυτικά για πενήντα πέντε (55) χώρες τον αριθμό κρουσμάτων/θανάτων, τις μετρικές των κρουσμάτων/θανάτων ανά εκατομμύριο και το ποσοστό θνησιμότητας. Επισημαίνει με έντονο καφέ χρώμα τις περιπτώσεις όπου κάποια χώρα έχει αυξημένη την αντίστοιχη μετρούμενη ποσότητα (στήλη). Η ταξινόμηση των χωρών του πίνακα γίνεται με βάση τον αριθμό των κρουσμάτων[1].

	Country Name	Number of Confirmed Cases	Number of Deaths	Mortality Rate	Cases per million	Deaths per million
0	United States of America	28468736	515013	1.809048	86007.578060	1555.918071
1	India	11192088	157656	1.408638	8110.182925	114.243115
2	Brazil	10793732	260970	2.417792	50779.836333	1227.750873
3	Russian Federation	4312181	88726	2.057567	29548.750857	607.985256
4	The United Kingdom	4207308	124261	2.953456	0.000000	0.000000
5	France	3791524	87776	2.315059	58086.716707	1344.741493
6	Spain	3142358	70501	2.243570	67209.337706	1507.888508
7	Italy	3023129	99271	3.283717	50000.621880	1641.878906
8	Turkey	2757460	28901	1.048102	32694.931283	342.676307
9	Germany	2492079	71804	2.881289	29744.111476	857.013835
10	Colombia	2266211	60189	2.655931	44537.786026	1182.892856
11	Argentina	2141854	52784	2.464407	47390.578106	1167.896726
12	Mexico	2112508	188866	8.940369	16384.572196	1464.841133
13	Poland	1766490	44912	2.542443	46674.992380	1186.685041
14	Iran (Islamic Republic of)	1673470	60512	3.615960	19923.933380	720.441392
15	South Africa	1518979	50566	3.328947	25611.407030	852.590067
16	Ukraine	1394061	26919	1.930977	31876.084560	615.519924
17	Indonesia	1368069	37026	2.706442	5001.648468	135.366737
18	Peru	1349847	47306	3.504545	40939.382041	1434.739201
19	Czechia	1312164	21558	1.642935	122529.293634	2013.076500
20	Netherlands	1110213	15762	1.419728	64792.601614	919.878426
21	Canada	878391	22151	2.521770	23273.471095	586.903393
22	Chile	845450	20928	2.475368	44226.865274	1094.777735
23	Romania	820931	20785	2.531881	42673.072567	1080.431624
24	Portugal	808405	16486	2.039324	79280.987480	1616.796481
25	Israel	786613	5792	0.736321	90879.703533	669.166722
26	Belgium	783010	22215	2.837129	67561.341118	1916.802075
27	Iraq	719121	13537	1.882437	17878.574091	336.552899
28	Sweden	684961	13003	1.898356	67822.822838	1287.518801
29	Philippines	587704	12423	2.113819	5363.188364	113.368105
30	Pakistan	587014	13128	2.236403	2657.466637	59.431669
31	Switzerland	559627	9278	1.657890	64662.241592	1072.028829
32	Bangladesh	549184	8441	1.537008	3334.665477	51.254063
33	Morocco	485567	8673	1.786159	13155.233253	234.973419
34	Serbia	478878	4525	0.944917	54808.025756	517.890395
35	Austria	466693	8513	1.824111	51817.929472	945.216735
36	Hungary	459816	15765	3.428545	47598.275425	1631.928450
37	Japan	437892	8178	1.867584	3462.241171	64.660255
38	Jordan	417934	4862	1.163342	40961.311910	476.519973
39	United Arab Emirates	405277	1296	0.319781	40976.805791	131.036156
40	Lebanon	390053	4971	1.274442	57146.921767	728.304482
41	Saudi Arabia	379092	6519	1.719635	10889.109216	187.252970
42	Panama	343281	5895	1.717252	79559.549899	1366.237999
43	Slovakia	322104	7739	2.402640	58997.264107	1417.491950
44	Malaysia	310097	1159	0.373754	9580.949736	35.809185
45	Belarus	293103	2020	0.689178	31018.419207	213.771974
46	Ecuador	291070	15997	5.495929	16497.705047	906.702125
47	Nepal	274608	3010	1.096108	9424.779818	103.305757
48	Georgia	272617	3567	1.308429	68339.192941	894.169847
49	Kazakhstan	265929	3389	1.274400	14162.707018	180.489582
50	Bulgaria	258385	10571	4.091182	37186.017879	1521.347582
51	Bolivia (Plurinational State of)	252360	11761	4.660406	21619.067339	1007.536262
52	Croatia	246120	5585	2.269218	59952.236979	1360.447113
53	Dominican Republic	242087	3150	1.301185	22316.476989	290.378676
54	Tunisia	236356	8130	3.439727	19998.615743	687.897688

Εικόνα 2.7: Συγκεντρωτικός πίνακας των χωρών με βάση τα κρούσματα

[1] Το Ηνωμένο Βασίλειο έχει μηδενικούς τους δείκτες c/m και d/m στην εικόνα 2.7 καθώς το φύλλο δεδομένων του O.H.E που χρησιμοποιήθηκε δεν περιείχε για τη συγκεκριμένη χρονική περίοδο τον πληθυσμό του ως σύνολο

2.3 Η περίπτωση των Η.Π.Α.

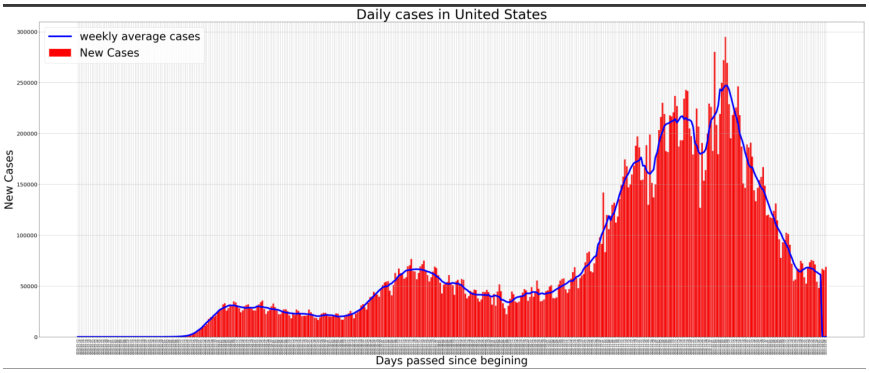
Στην ενότητα 1.3 αναφέραμε ότι μία εκ των πηγών των δεδομένων μας αναφέρεται αποκλειστικά στις ΗΠΑ. Η επιλογή αυτή έγινε αφενός επειδή, όπως είδαμε, η συγκεκριμένη χώρα έχει υποστεί βαρύτατο πλήγμα από την πανδημία και αφετέρου εξαιτίας του γεγονότος ότι μας προσφέρει μεγαλύτερη πληθώρα δεδομένων (πλήθος διεξαγόμενων τεστ, καθημερινά διεξαγόμενα τεστ, εισαγωγές στα νοσοκομεία, ασθενείς σε ΜΕΘ, ασθενείς διασωληνωμένοι κ.α) χωρίς εκλιπούσες τιμές (missing values). Ορισμένες χώρες για παράδειγμα, δε δίνουν αριθμό των τεστ ή των ασθενών σε ΜΕΘ ή τον παρέχουν μία φορά την εβδομάδα. Στη συγκεκριμένη πηγή πληροφοριών (dataset [22]) μπορούμε αναλυτικά να βρούμε αθροιστικούς αριθμούς τεστ, κρουσμάτων, εισαγωγών στα νοσοκομεία, ασθενών σε ΜΕΘ, ασθενών που χρήζονται αναπνευστήρα αλλά και θανάτων. Επίσης, υπάρχουν οι καθημερινές αντίστοιχες παρατηρήσεις. Έχοντας τον αριθμό των τεστ σε καθημερινή βάση είμαστε σε θέση να μετρήσουμε μία ακόμα σημαντική μετρική, το ποσοστό θετικότητας, που ορίζεται ως η διαίρεση του αριθμού κρουσμάτων με τον αριθμό των τεστ:

$$Positivity \quad Rate = \frac{Nunber \quad of \quad cases}{Number \quad of \quad tests} \times 100\% \tag{2.3.1}$$

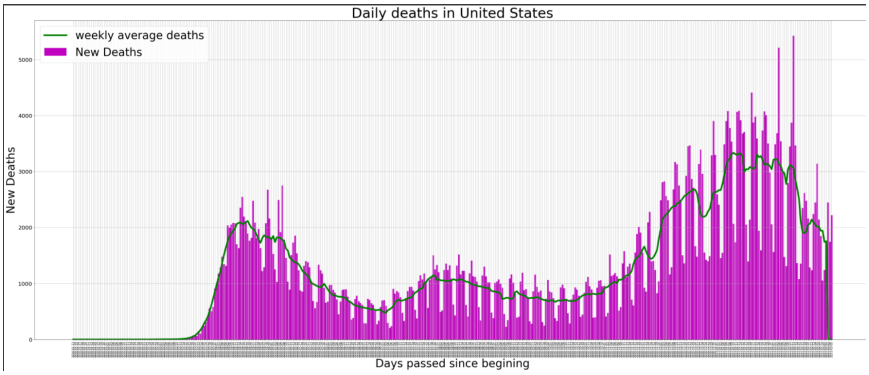
Η συγκεκριμένη ποσότητα βοηθά στην κατανόηση της εξάπλωσης του ιού στην κοινότητα (για αρκούντος μεγάλο αριθμό τεστ). Στις εικόνες 2.8 παρουσιάζονται τα καθημερινά κρούσματα και οι καθημερινοί θάνατοι των ΗΠΑ ενώ στις εικόνες 2.9 παρουσιάζονται τα καθημερινά τεστ και η διακύμανση του ποσοστού θετικότητας κατά τη διάρκεια του χρόνου για την ίδια χώρα.

Βλέπουμε ότι η εικόνα των κρουσμάτων και των θανάτων έχουν παρόμοια μορφή, ενώ είναι αξιοσημείωτη και η μερική ομοιότητα των κρουσμάτων με τα συνολικά παγκόσμια κρούσματα της εικόνας 2.4α′. Αυτό είναι αναμενώμενο κάθως, όπως είδαμε νωρίτερα, οι ΗΠΑ από μόνες τους κατέχουν ενά μεγάλο μερίδιο των συνολικών κρουσμάτων (εικόνα 2.5α′). Από τις εικόνες 2.9 βλέπουμε ότι τα τεστ ακολουθούν το μοτίβο των κρουσμάτων ενώ ο δείκτης θετικότητας ξεπέρασε ακόμα και το 20% στις αρχές της πανδημίας (οι αρχικές μεγάλες διακυμάνσεις δεν προσμετρούνται, πιθάνως να οφείλονται σε κάποια μεμονωμένα τεστ που διεξήχθησαν πριν ο ιός φτάσει για τα καλά στην αμερικανική ήπειρο), ενώ στη συνέχεια σταθεροποιήθηκε κάτω από το 10% με λιγόστες εξαιρέσεις.

Στη συνέχεια, χρησιμοποιούμε τα δεδομένα που σχετίζονται με τους ασθένεις σε νοσοκομεία, μονάδες εντατικής θεραπείας και τους διασωληνωμένους ασθενείς. Η εικόνα 2.10α′ αναδεικνύει ακριβώς αυτήν την τριπλή πληροφορία. Βλέπουμε ότι οι αυξομειώσεις των τριών (3) καμπυλών σχετίζονται άμεσα, ενώ μεγάλη είναι και η ομοιότητα με το γραφήμα των θανάτων στην εικόνα 2.8β′. Στην εικόνα 2.10β′ απεικονίζουμε (κατά τη διάρκεια του χρόνου) την μέση πιθανότητα ένας ασθένης που εισάγεται στο νοσοκομείο να καταλήξει στην ΜΕΘ αλλά και στην διασωλήνωση. Έστω H, I και V τα ενδεχόμενα κάποιος ασθενής να εισαχθεί σε νοσοκομείο,να εισαχθεί στην εντατική και να διασωληνωθεί αντίστοιχα, τότε οι ζητούμενες δεσμευμένες πιθανότητες υπολογίζονται από τις

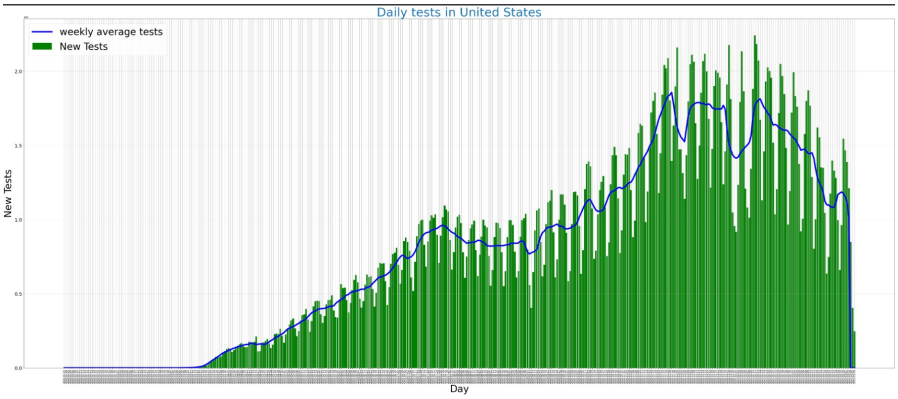


(α′) Τα καθημερινά κρούσματα των ΗΠΑ

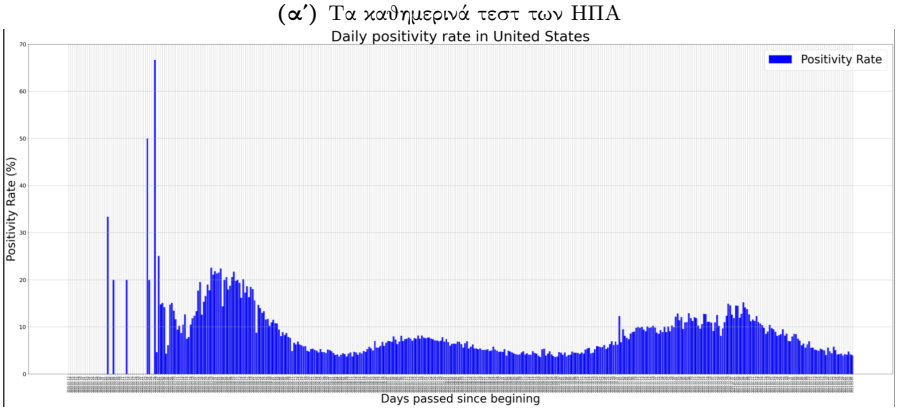


(β′) Οι καθημερινοί θάνατοι των ΗΠΑ.

Εικόνα 2.8: Τα κρούσματα και οι θάνατοι στις ΗΠΑ



(α′) Τα καθημερινά τεστ των ΗΠΑ



(β′) Καθημερινό ποσοστό θετικότητας στις ΗΠΑ.

Εικόνα 2.9: Τεστ και ποσοστό θετικότητας στις ΗΠΑ

εξής σχέσεις:

$$P_I(I|H) = \frac{N_I}{N_H} \times 100\%, \quad P_V(V|H) = \frac{N_V}{N_H} \times 100\% \tag{2.3.2}$$

όπου N_H, N_I και N_V ο αριθμός των ασθενών που νοσηλεύονται σε νοσοκομείο, μονάδα εντατικής και βρίσκονται διασωληνωμένοι αντίστοιχα.

Στην εικόνα 2.10γ′ μπορούμε να δούμε το CFR των Ηνωμένων Πολιτειών που στην αρχή βρέθηκε ακόμα και στο 6%, αλλά στη συνέχεια ελαττώθηκε σταδιακά έως και το 2% περίπου (εξαιρείται η αρχική, κάθετη σχεδόν ανύψωση).

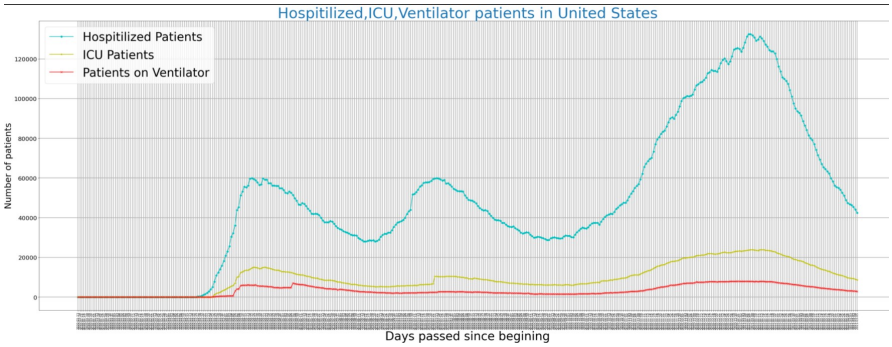
2.4 Γεωγραφικοί Θερμοχάρτες

Οι 'γεωγραφικοί θερμοχάρτες' αν και πολύ διαδεδομένοι, τόσο στην περίπτωση της πανδημίας όσο και γενικότερα, αποτελούν μία ιδιαίτερη υλοποίηση για τη συγκεκριμένη εργασία, μιας και αποτελούν αφενός πρωτόγνωρη εμπειρία και αφετέρου αναπόσπαστο κομμάτι της Επιστήμης των Δεδομένων σε μια διπλωματική που ούτως ή άλλως διασταυρώνεται σε πολλά σημεία με το συγκεκριμένο κλάδο. Για την πραγματοποίηση τους χρησιμοποιήθηκε η γλώσσα Python και μεταξύ άλλων η βιβλιοθήκη *folium* [27] καθώς και γεωγραφικά δεδομένα των χωρών ανά την υφήλιο που βρίσκονται στο [28]. Τα δεδομένα αυτά συνδυάστηκαν στη συνέχεια με τα επιδημιολογικά προκειμένου να γίνει σωστά η αποτύπωση τους. Ένα ξεχωριστό σημείο που στάθηκε εμπόδιο κατά την υλοποίηση των χαρτών (κυρίως όσον αφορά τη σταδιακή κλιμάκωση των χρωμάτων πάνω στο χάρτη) είναι το γεγόνος ότι οι διάφορες χώρες είχαν τρομακτικά μεγάλες διαφορές στους αριθμούς κρουσμάτων και θανάτων τους. Για παράδειγμα, για τις ειδικώς προαναφερόμενες Ηνωμενές Πολιτείες, ο αριθμός κρουσμάτων (κατά τη συγγραφή της εργασίας) άγγιζε τα 28.5 εκατομμύρια (εικόνα 2.7), ενώ για την πατρίδα μας στον αντίποδα ήταν μόλις 201 χιλιάδες. Αυτό δημιουργεί μια διαφορά ως προς την τάξη μεγέθους των αριθμών μεγαλύτερη του 100, πράγμα που δεν βοηθούσε στην αποτύπωση. Για το λόγο αυτό επιλέχθηκε γραμμικοποίηση των εν λόγω χαρακτηριστικών (κρουσμάτων, θανάτων) με τη χρήση του φυσικού λογαρίθμου, δηλαδή:

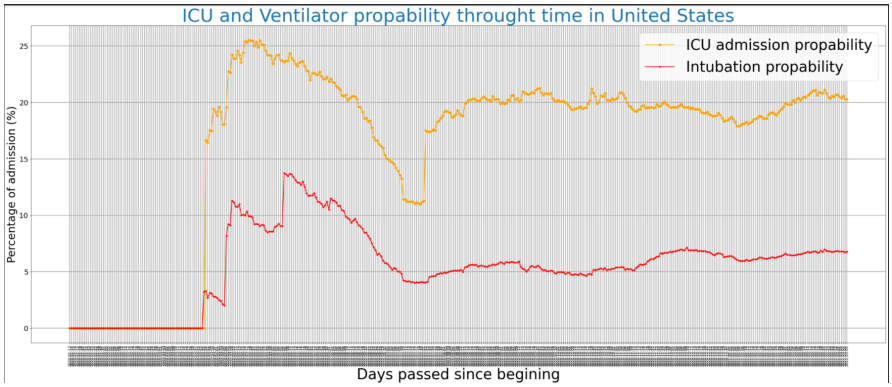
$$\text{number of cases}_{lin} = \ln(\text{number of cases}) \tag{2.4.1}$$

$$\text{number of deaths}_{lin} = \ln(\text{number of deaths}) \tag{2.4.2}$$

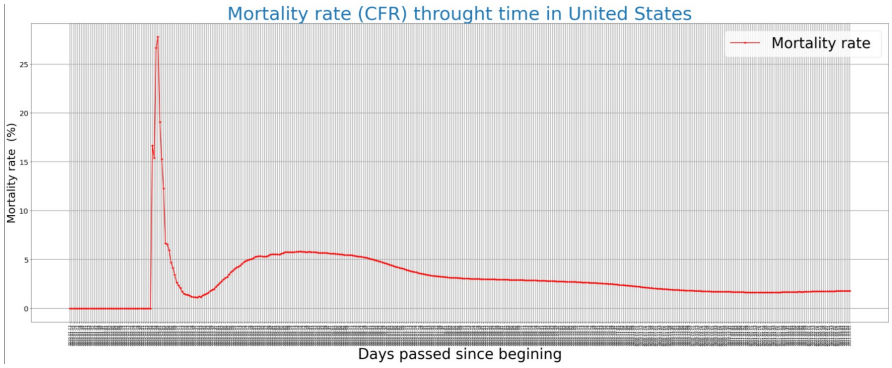
Στις εικόνες 2.11 έως 2.15 παρουσιάζονται κατά σειρά οι παγκόσμιοι χάρτες για τα κρούσματα, τους θανάτους, το ποσοστό θνησιμότητας, τα κρούσματα ανά εκατομμύριο και τους θανάτους ανά εκατομμύριο. Στις τρεις τελευταίες μεταβλητές λόγω της ήδη κανονικοποιήσής τους, κατά κάποιον τρόπο, δεν χρειάστηκε κάποια μέθοδος γραμμικοποίησης. Κάθε χώρα τονίζεται με εντόνοτερο χρώμα όταν διαθέτει αυξημένη την αντίστοιχη μεταβλητή.



(α′) Οι καμπύλες των νοσηλευομένων, ασθενών σε ΜΕΘ και διασωληνωμένων ασθενών

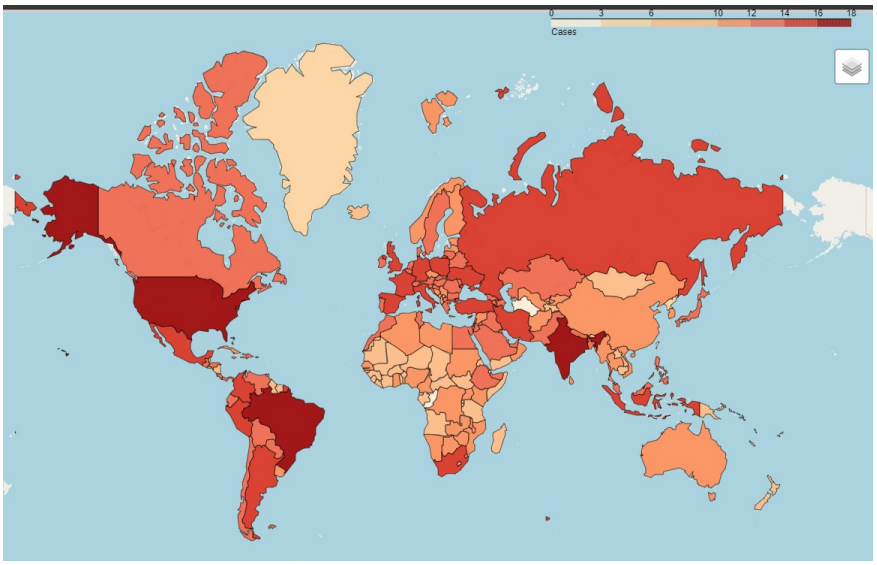


(β′) Η πιθανότητα εισαγωγής σε ΜΕΘ και διασωλήνωσης με δεδομένη την εισαγωγή σε νοσοκομείο

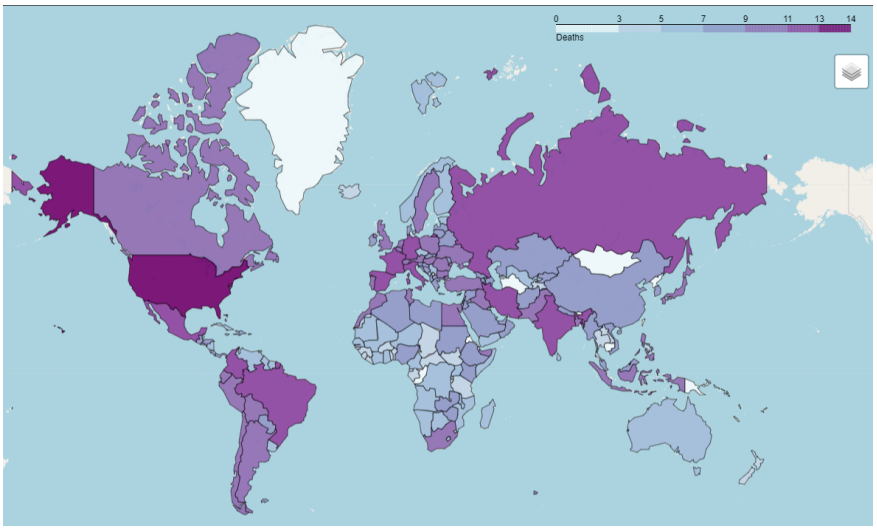


(γ′) Το ποσοστό θνησιμότητας στις ΗΠΑ

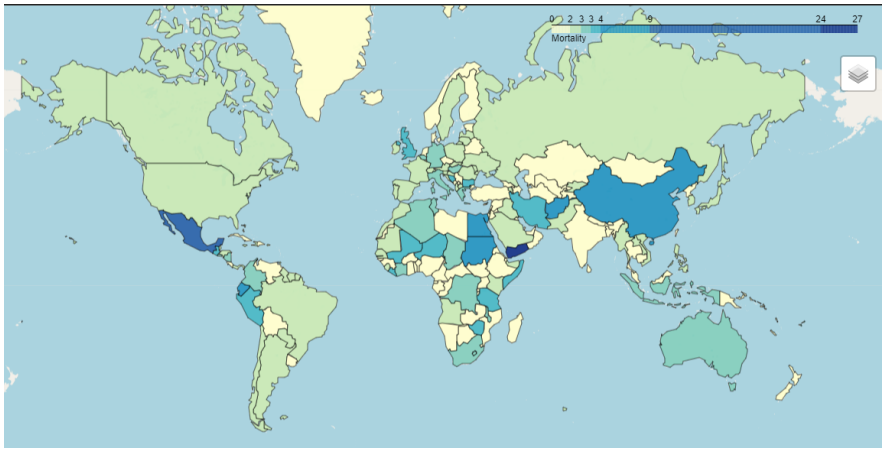
Εικόνα 2.10: Διάφορες μετρικές των ΗΠΑ κατά τη διάρκεια του χρόνου



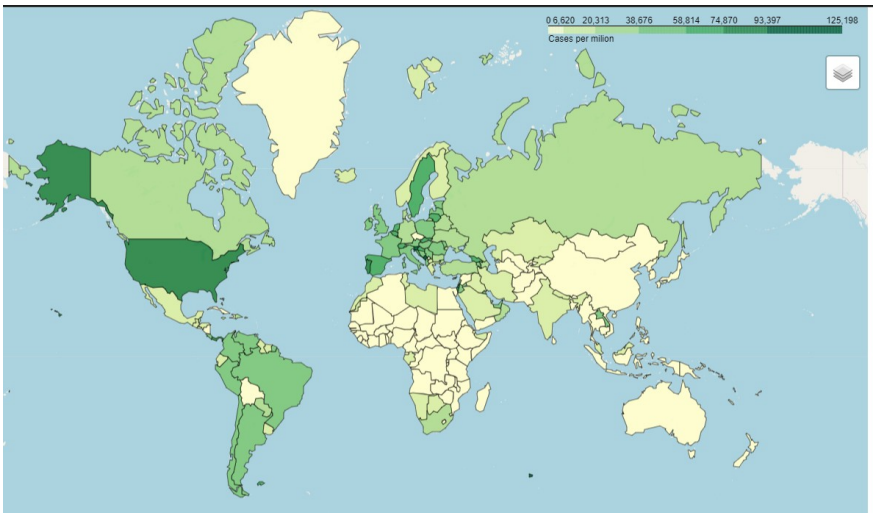
Εικόνα 2.11: Γεωγραφικός θερμοχάρτης του απόλυτου αριθμού κρουσμάτων με γραμμικοποιημένο υπόμνημα



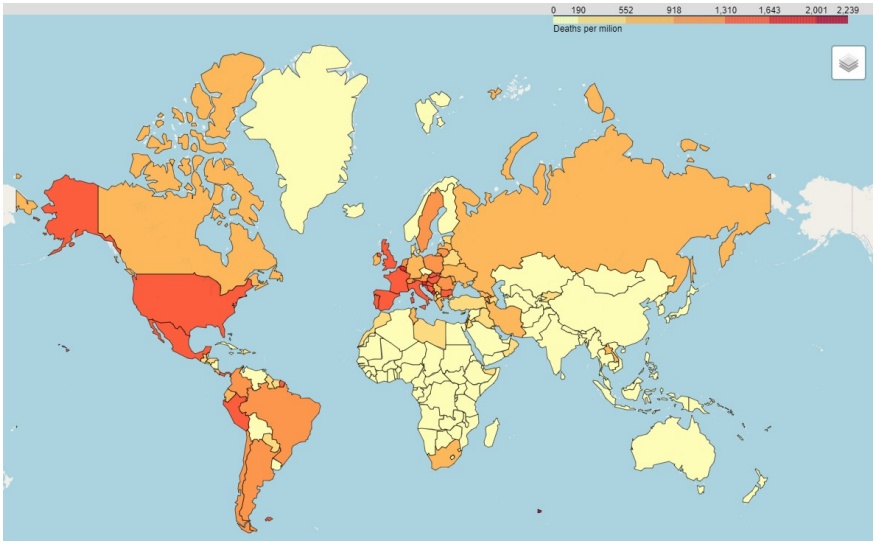
Εικόνα 2.12: Γεωγραφικός θερμοχάρτης του απόλυτου αριθμού θανάτων με γραμμικοποιημένο υπόμνημα



Εικόνα 2.13: Γεωγραφικός θερμοχάρτης του ποσοστού θνησιμότητας



Εικόνα 2.14: Γεωγραφικός θερμοχάρτης του αριθμού των κρουσμάτων ανά εκατομμύριο



Εικόνα 2.15: Γεωγραφικός θερμοχάρτης του αριθμού των θανάτων ανά εκατομμύριο.

Όσον αφορά στο σχολιασμό των παραπάνω εικόνων επιβεβαιώνουμε τα όσα έχουμε δει νωρίτερα, αναφορικά με τα κρούσματα και τους θανάτους, με τις ΗΠΑ να κατέχουν την πρωτιά και την Βραζίλια και την Ινδία να ακολουθούν. Για το ποσοστό θνησιμότητας βλέπουμε, ότι ευτυχώς, σε λίγες χώρες ήταν αυξημένο, ήτοι μεγαλύτερο από 5% με πρώτη την Υεμένη στην Αραβική χερσόνησο. Τέλος, από τις μετρικές ανά εκατομμύριο μπορούμε να δούμε τόσο από τα κρούσματα όσο κυρίως από τους θανάτους ότι σε συνολική εικόνα το μεγαλύτερο πρόβλημα απέναντι στην πανδήμια είχαν οι χώρες της Ευρώπης, οι χώρες της Λατινικής Αμερικής και φυσικά οι Ηνωμένες Πολιτείες.

Κεφάλαιο 3

Νευρωνικά Δίκτυα

3.1 Εισαγωγή στη Μηχανική Μάθηση

3.1.1 Από τη γραμμική παλινδρόμηση στα νευρωνικά δίκτυα

Στην ενότητα 1.2 αναφέραμε τη γραμμική παλινδρόμηση ως μέσο προκειμένου να συσχετίσουμε μία/πολλές μεταβλητή/ες παρακολούθησης, έστω $x_1, ..., x_d$, με μία άλλη εξαρτημένη μεταβλητή, έστω y, και εν συνεχεία να προβλέψουμε αυτή τη μεταβλητή y βασιζόμενοι στις παρατηρήσεις μας. Ο τρόπος με τον οποίο η γραμμική παλινδρόμηση πετυχαίνει αυτό το στόχο είναι η παρακάτω πολύ απλή εξίσωση:

$$\hat{y} = w_1 x_1 + ... + w_d x_d + b \tag{3.1.1}$$

όπου $\hat{y}$ η εκτίμησή μας για τη μεταβλητή y, $w_1, \ldots, w_d$ είναι τα λεγόμενα *βάρη* και b μία σταθερά διόρθωσης, συχνά αποκαλούμενη *bias*.

Ο απώτερος σκοπός της γραμμικής παλινδρόμησης αλλά και γενικότερα όλων των τεχνικών μηχανικής μάθησης είναι η κατά το δυνατόν καλύτερη εύρεση των παραμέτρων $w_1, \ldots, w_d$ και b ώστε το $\hat{y}$ να προσεγγίζει καλύτερα το y. Μπορούμε να γράψουμε την εξίσωση (3.1.1) ως εξής:

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b \tag{3.1.2}$$

όπου πλέον $\mathbf{x}, \mathbf{w} \in \mathbb{R}^d$. Η εξίσωση (3.1.2), ωστόσο αναφέρεται μονάχα σε ένα δείγμα εισόδου. Υποθέτοντας, n παραδείγματα είσοδου τότε η είσοδός μας μπορεί να εκφραστεί στη μορφή $X \in \mathbb{R}^{n \times d}$. Προφανώς, αναμένουμε και n παραδείγματα εξόδου και έτσι καταλήγουμε στην εξίσωση:

$$\hat{\mathbf{y}} = X\mathbf{w} + b \tag{3.1.3}$$

όπου $\hat{\mathbf{y}} \in \mathbb{R}^n$. Προκειμένου να ποσοτικοποιήσουμε την απόσταση μεταξύ της πραγματικής και της προβλεπόμενης τιμής, χρησιμοποιούμε τη *συνάρτηση απωλειών (loss function)*. Οι απώλειες είναι συνήθως ένας μη αρνητικός αριθμός όπου προφανώς μικρότερες τιμές δηλώνουν καλύτερες προβλέψεις, ενώ απώλειες ίσες με μηδέν (0) μεταφράζονται σε τέλεια πρόβλεψη. Η πιο δημοφιλής συνάρτηση απωλειών στα προβλήματα

παλινδρόμησης είναι αυτή του *τετραγωνικού σφάλματος* [29]. Έτσι, όταν η πρόβλεψή μας για ένα παράδειγμα i είναι $\hat{y}^{(i)}$ και η αντίστοιχη πραγματική ετικέτα είναι $y^{(i)}$, η συνάρτηση απωλειών δίνεται από την:

$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2 \tag{3.1.4}$$

Ο συντελεστής $1/2$ συνδράμει κατά την παραγώγιση της συνάρτησης απωλειών όταν και εξαλείφεται από τον τετραγωνικό εκθέτη. Συμπεριλαμβάνοντας όλα τα n παραδείγματα εισόδου μπορούμε να γενικεύσουμε την εξίσωση (3.1.4) και να ορίσουμε την ολική συνάρτηση απωλειών με απλή άθροιση όλων των επιμέρους απωλειών ως:

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2 \tag{3.1.5}$$

Ο κατ' εξοχήν τρόπος για βελτιστοποίηση των προβλέψεών μας στα πλαίσια της μηχανικής μάθησης είναι η συνεχής ελάττωση του λάθους, ανανεώνοντας συνεχώς τις παραμέτρους των βαρών (και του bias) στην κατεύθυνση όπου η συνάρτηση απωλείων L σταδιακά μειώνεται. Αυτό σημαίνει ότι ψάχνουμε τα καταλληλότερα βάρη $(\mathbf{w}^*, b^*)$ που ελαχιστοποιούν το εξής κριτήριο: $(\mathbf{w}^*, b^*) = \underset{\mathbf{w},b}{\operatorname{argmin}}\{L(\mathbf{w}, b)\}$. Ο αλγόριθμος που χρησιμοποιούμε γι' αυτήν τη διαδικασία ονομάζεται *κατάβαση πλαγιάς* (gradient descent) [30]. Έτσι, θεωρώντας ένα δείγμα της εισόδου που περιέχει κάποιο σταθερό αριθμό παραδειγμάτων (συχνά το αποκαλούμε *δέσμη εισόδου* (input minibatch)) $\mathcal{B}$, υπολογίζουμε την παράγωγο της μέσης απώλειας σε αυτή τη δέσμη ως προς τα βάρη και το bias. Στη συνέχεια, πολλαπλασιάζουμε την κλίση με μία προκαθορισμένη θετική τιμή η και αφαιρούμε τον προκύπτον όρο από τις τρέχουσες τιμές βαρών/biases. Μαθηματικώς συμβολίζουμε ως εξής (το ∂ δηλώνει τη μερική παράγωγο):

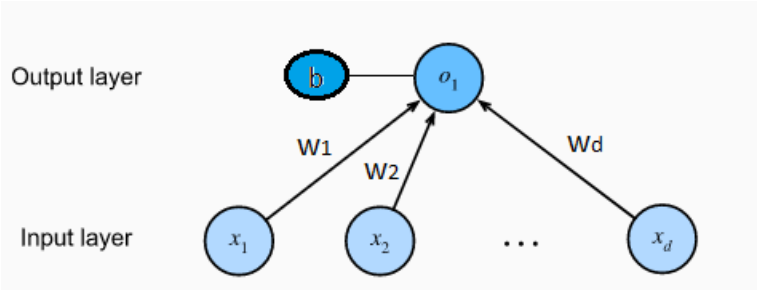
$$(\mathbf{w}, b) \leftarrow (\mathbf{w}, b) - \frac{\eta}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \partial_{(\mathbf{w},b)} l^{(i)}(\mathbf{w}, b) \tag{3.1.6}$$

Το η είναι μία καθοριζόμενη από τον χρήστη σταθερά και ονομάζεται *ρυθμός εκπαίδευσης*, ενώ με $|\mathcal{B}|$ συμβολίζουμε το πλήθος παραδειγμάτων μέσα στην επιλεγμένη δέσμη, αναφερόμενο σαν batch size. Στην εικόνα 3.1 φαίνεται ένα μοντέλο γραμμικής παλινδρόμησης που μπορεί να θεωρηθεί και ως ένα απλό νευρωνικό δίκτυο, στο οποίο κάθε χαρακτηριστικό εισόδου x_i, $i \in \{1, d\}$ πολλαπλασιάζεται με το αντίστοιχο βάρος w_i και έπειτα προστίθεται η σταθερά b για να πάρουμε την έξοδο o_1.

Παλινδρόμηση Softmax

Διευρευνώντας το συλλογισμό μας, υποθέτουμε ότι αντιμετωπίζουμε ένα πρόβλημα ταξινόμησης εικόνων 2×2 σε 3 κατηγορίες. Αυτό σημαίνει ότι έχουμε 4 χαρακτηρηστικά $(x_1, x_2, x_3, x_4) = \mathbf{x}$ ενώ οι 3 πιθανές κατηγορίες είναι $(o_1, o_2, o_3) = \mathbf{o}$. Συνεπώς, προκειμένου για κάθε κατηγορία να λάβουμε πληροφορία από κάθε χαρακτηριστικό, θα χρειαστούμε 12 βάρη (w_{ij}) και 3 διορθωτικές σταθερές (b_i). Άρα, θα έχουμε:

$$o_1 = x_1 w_{11} + x_2 w_{12} + x_3 w_{13} + x_4 w_{14} + b_1,$$



Εικόνα 3.1: Η γραμμική παλινδρόμηση είναι ένα νευρωνικό δίκτυο ενός μόνο επιπέδου και μίας εξόδου [31]

$$o_2 = x_1 w_{21} + x_2 w_{22} + x_3 w_{23} + x_4 w_{24} + b_2,$$
$$o_3 = x_1 w_{31} + x_2 w_{32} + x_3 w_{33} + x_4 w_{34} + b_3.$$

ή μπορούμε να γράψουμε, ισοδύναμα, την ακόλουθη διανυσματική εξίσωση ως:

$$\mathbf{o} = W\mathbf{x} + \mathbf{b} \tag{3.1.7}$$

όπου W ένας 3×4 πίνακας και $\mathbf{b}$ ένα 3×1 διάνυσμα.

Ωστόσο, για να μπορέσει να λειτουργήσει ορθώς ένα τέτοιο δίκτυο ταξινόμησης θα πρέπει να ερμηνεύσουμε τις εξόδους του μοντέλου μας ως πιθανότητες. Οποιαδήποτε έξοδος $\hat{y}^j$ ερμηνεύεται ως η πιθανότητα το δεδομένο στοιχείο να ανήκει στην κλάση j. Επομένως, μπορούμε να επιλέξουμε ως την τελική μας πρόβλεψη την κατηγορία με τη μεγαλύτερη πιθανότητα. Το πρόβλημα όμως, είναι πως οι έξοδοι $\mathbf{o}$ είναι ακαθόριστες υπό την έννοια ότι μπορεί να είναι είτε θετικές είτε αρνητικές και δεν αθροίζονται στο 1 ώστε να αντιπροσωπεύουν πιθανότητες. Για να αντιμετωπίσουμε αυτό το πρόβλημα εισάγουμε τη συνάρτηση *softmax* η οποία εφευρέθηκε το 1959 από τον κοινωνικό επιστήμονα R. Duncan Luce στο πλαίσιο των μοντέλων επιλογής [32]. Οπότε, ορίζουμε ως:

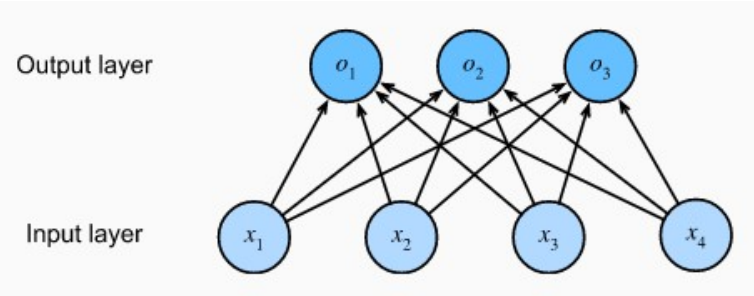
$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{o}) \quad \text{όπου} \quad \hat{y}_j = \frac{\exp(o_j)}{\sum_k \exp(o_k)} \tag{3.1.8}$$

Για $0 \leq \hat{y}_j \leq 1$ έχουμε την απαιτούμενη συνθήκη $\sum_j \hat{y}_j = 1$.
Γενικεύοντας, υποθέτουμε ότι για μία δέσμη της εισόδου X έχουμε αριθμό παραδειγμάτων d και μέγεθος δέσμης n (άρα θα είναι $X \in \mathbb{R}^{n \times d}$) καθώς, και ότι η διάσταση εξόδου (αριθμός κατηγοριών/κλάσεων) είναι q. Επομένως, η εξίσωση (3.1.7) μετατρέπεται ως εξής:

$$O = XW + \mathbf{b} \tag{3.1.9α′}$$
$$\hat{Y} = \text{softmax}(O) \tag{3.1.9β′}$$

όπου $W \in \mathbb{R}^{d \times q}$ είναι ο πίνακας των βαρών και $\mathbf{b} \in \mathbb{R}^{1 \times q}$ είναι το διάνυσμα των τιμών bias, ενώ τελικά η έξοδος πιθανοτήτων $\hat{Y}$ έχει διάσταση $n \times q$. Είναι αναγκαίο να τονιστεί ότι κατά τη βελτιστοποίηση του μοντέλου ισχύει: $\text{argmax}_j \hat{y}_j = \text{argmax}_j o_j$,

πράγμα που σημαίνει ότι μας επιτρέπεται να διαλέξουμε την κατηγορία εξόδου με τη μεγαλύτερη πιθανότητα $\hat{y}^j$. Στην εικονα 3.2 μπορούμε να δούμε ένα μοντέλο παλινδρόμησης softmax που αποτελεί ένα μονοστρωματικό νευρωνικό δίκτυο ταξινόμησης 3 κλάσεων. Η λειτουργία της συνάρτησης softmax υπονοείται ότι συμβαίνει στις εξόδους $o1, o2, o3$



Εικόνα 3.2: Η παλινδρόμηση softmax ως αναπαράσταση νευρωνικού δικτύου ενός στρώματος [33]

3.1.2 Τεχνητά Νευρωνικά Δίκτυα

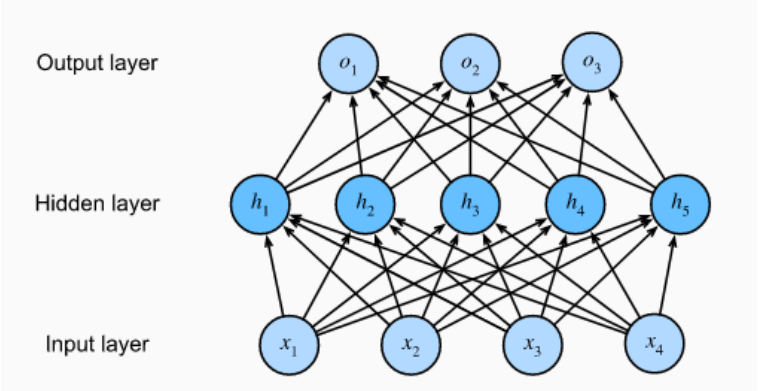
Το νευρωνικά δίκτυα των εικόνων 3.1, 3.2 είναι φανερό ότι παρουσιάζουν μία γραμμικότητα. Οι είσοδοι, πολλαπλασιαζόμενες με κάποια βάρη, καταλήγουν στην έξοδο από όπου και παίρνουμε το αποτέλεσμα (ενδεχομένως με κάποια επιπρόσθετη τιμή bias). Ας ανατρέξουμε σε ένα παράδειγμα που αναδεικνύει την ανικανότητα διαχείρισης μη γραμμικών δεδομένων από τέτοιου είδους δίκτυα.

Υποθέτουμε ότι θέλουμε να προβλέψουμε την πιθανότητα θανάτου με βάση τη θερμοκρασία του σώματος (βαριά νόσος από Covid-19). Για άτομα με θερμοκρασία σώματος άνω των 37 °C , οι υψηλότερες θερμοκρασίες υποδεικνύουν μεγαλύτερο κίνδυνο. Αντιθέτως, για άτομα με θερμοκρασίες σώματος κάτω από 37 °C, οι υψηλότερες θερμοκρασίες δείχνουν χαμηλότερο κίνδυνο! Σε αυτή την περίπτωση, μπορεί να επιλύσουμε το πρόβλημα με μία έξυπνη προεπεξεργασία. Θα μπορούσαμε να χρησιμοποιήσουμε την απόσταση από τους 37 °C ως χαρακτηριστικό διαφοροποίησης.

Μπορούμε να ξεπεράσουμε αυτόν τον περιορισμό των γραμμικών μοντέλων και να χειριστούμε δεδομένα γενικότερης μορφής, ενσωματώνοντας ένα ή περισσότερα *κρυφά στρώματα ή επίπεδα* (hidden layers) στα δίκτυα. Ως κρυφό στρώμα ορίζουμε ένα επιπλέον επίπεδο νευρώνων που δεν βρίσκεται σε άμεση επαφή με κάποια είσοδο ή έξοδο του δικτύου.

Ο ευκολότερος τρόπος για να επεκτείνουμε το δίκτυό μας είναι απλώς να στοιβάξουμε ένα πλήρως διασυνδεδεμένο επίπεδο ανάμεσα στην είσοδο και την έξοδο, όπως φαίνεται στην εικόνα 3.3. Ένα δίκτυο σαν αυτό το ονομάζουμε *πολυστρωματικό νευρωνικό δίκτυο* (Multi Layer Perceptron - MLP [34]). Φυσικά, τέτοια δίκτυα μπορεί να απαρτίζονται από περισσότερα του ενός, κρυφά επίπεδα, λόγου χάρη L. Σε αυτή την περίπτωση τα $L-1$ (από την είσοδο προς την έξοδο) κρυφά επίπεδα αναπαριστούν

το σώμα του δικτύου που επιτελεί μη γραμμικές λειτουργίες και το τελευταίο στρώμα συνιστά τη γραμμική έξοδο.



Εικόνα 3.3: Το δίκτυο MLP με ένα κρυφό επίπεδο και πέντες κρυφούς νευρώνες [35]

Με μαθηματικές εκφράσεις[35], θεωρούμε ότι η είσοδος $X \in \mathbb{R}^{n\times d}$ αποτελεί μια δέσμη εισόδου με n παραδείγματα, όπου κάθε παράδειγμα έχει d εισόδους χαρακτηριστικών. Για ένα Multilayer Perceptron με ένα κρυφό επίπεδο που απαρτίζεται από h σε πλήθος νευρώνες σημειώνουμε με $H \in \mathbb{R}^{n\times h}$ την έξοδο του κρυφού του επιπέδου. Αναφερόμαστε επίσης στο H ως *κρυφή αναπαράσταση ή κρυφή μεταβλητή*. Η εξίσωση που μας δίνει την κρυφή αυτή μεταβλητή είναι:

$$H = \phi(XW_{xh} + b_h) \tag{3.1.10}$$

όπου ϕ μία *συνάρτηση ενεργοποίησης*, $b_h \in \mathbb{R}^{1\times h}$ το διάνυσμα των τιμών bias, $W_{xh} \in \mathbb{R}^{h\times h}$ η μήτρα βαρών για το κρυφό επίπεδο. Αντίστοιχα, στο στρώμα εξόδου έχουμε:

$$O = HW_{hq} + b_q \tag{3.1.11}$$

όπου $W_{hq} \in \mathbb{R}^{h\times q}$ η μήτρα βαρών για το στάδιο εξόδου, $b_q \in \mathbb{R}^{1\times q}$ το διάνυσμα των όρων bias για την έξοδο, q το πλήθος των εξόδων και $H \in \mathbb{R}^{h\times h}$ η έξοδος του κρυφού επιπέδου της εξίσωσης (3.1.10).

Συνάρτηση Ενεργοποίησης

Νωρίτερα, στην εξίσωση (3.1.10) ονομάσαμε τη συνάρτηση ϕ ως συνάρτηση ενεργοποίησης. Οι συναρτήσεις ενεργοποίησης καθορίζουν αν ένας νευρώνας πρέπει να ενεργοποιηθεί ή όχι υπολογίζοντας ένα σταθμισμένο άθροισμα και προσθέτοντας επιπλεόν την τιμή bias σε αυτό. Είναι τμηματικά παραγωγίσιμες συναρτήσεις προκείμενου να μετατρέπουν τα σήματα εισόδου σε διακριτές τιμές εξόδου, ενώ οι περισσότερες από αυτές είναι μη γραμμικές. Μερικές από τις πιο γνωστές συναρτήσεις ενεργοποίησης αναφέρονται παρακάτω:

α) *διορθωτική γραμμική μονάδα (rectified linear unit - ReLU)* που ορίζεται ως η μέγιστη τιμή μεταξύ της εισόδου x και του 0,

$$\mathrm{ReLU}(x) = \max(x, 0) \tag{3.1.12}$$

β) *η σιγμοειδής συνάρτηση* (sigmoid), η οποία περιορίζει την είσοδό της στο διάστημα $[0, 1]$ και έτσι αποτελεί χρήσιμη επιλογή σε περιπτώσεις που επιθυμούμε για παράδειγμα το δίκτυο να υπολογίζει πιθανότητες.

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \tag{3.1.13}$$

γ) *η συνάρτηση υπερβολικής εφαπτομένης* (tanh), η οποία επίσης περιορίζει τις εισόδους της στο διάστημα $[-1, 1]$.

$$\tanh(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)} \tag{3.1.14}$$

Μία εναλλακτική συνάρτηση ενεργοποίησης, που χρησιμοποιείται σε νευρωνικά δίκτυα, όταν αντιμετωπίζουμε προβλήματα κατηγοριοποίησης δεδομένων εισόδου σε N κλάσεις που είναι άνω των δύο, είναι η ήδη γνωστή μας από την εξίσωση (3.1.8) συνάρτηση softmax όπου αναδιατυπώνοντας τον τύπο της έχουμε:

$$Softmax(x_i) = \frac{\exp(x_i)}{\sum_{n=1}^{N} \exp(x_i)} \tag{3.1.15}$$

3.2 Επαναληπτικά Νευρωνικά Δίκτυα

3.2.1 Απλά Επαναληπτικά Νευρωνικά Δίκτυα

Έχοντας γνωρίσει τις βασικές αρχές των νευρωνικών Δικτύων σε αυτή την υποενότητα θα περάσουμε σε μία νέα ομάδα, πιο σύγχρονων και πιο εξελιγμένων δικτύων, τα ***Επαναληπτικά Νευρωνικά Δίκτυα*** (Recurrent Neural Networks-RNNs)[36].

Τα δίκτυα αυτά αποτελούν και τη βάση της εργασίας γύρω από τα οποία πραγματοποιήθηκαν τα διάφορα πειράματα. Τα RNNs είναι μία ειδική κατηγορία δικτύων που έχουν τη δυνατότητα να διαχειρίζονται με μεγάλη αποτελεσματικότητα ακολουθιακά δεδομένα μεταβλητού μήκους εισόδου, καθώς διαθέτουν μία μεταβλητή κατάστασης που ονομάζεται *Κρυφή Κατάσταση* (Hidden State) και η οποία τους επιτρέπει να αποθηκεύουν πληροφορία από το παρελθόν (δηλαδή πληροφορίες από δεδομένα που έχουν συναντήσει νωρίτερα). Έτσι, μπορούν να χρησιμοποιήσουν αυτήν την πληροφορία μαζί με τις πιο πρόσφατες εισόδους προκειμένου να καθορίσουν την επόμενη έξοδο. Έχει αποδειχθεί ότι μπορούν να αποδώσουν σε περιπτώσεις όπως η αναγνώριση γραφής ή η αναγνώριση ομιλίας[37].

Θα πρέπει να τονίσουμε επίσης τη διαφορά μεταξύ των Hidden States που μόλις αναφέραμε και των Hidden Layers που έχουμε ήδη γνωρίσει νωρίτερα. Ένω τα Hidden Layers είναι στρώσεις του δικτύου που δεν είναι ορατές από την είσοδο και την έξοδο, τα Hidden States είναι καταστάσεις που αποτελούν οι ίδιες τις εισόδους του δικτύου σε κάθε στάδιο ξετυλίγματός (unroll) του και χρησιμοποιούν πληροφορία από προηγούμενα στάδια.

Τα RNNs χωρίζονται σε δύο (2) κατηγορίες ανάλογα με τον τρόπο αναπαράστασής τους. Τα δίκτυα πεπερασμένης ώθησης που δύνανται να ξετυλιχθούν και να αποτυπωθούν ως κατευθυνόμενος μη κυκλικός γράφος και τα δίκτυα άπειρης ώθησης που δε

γίνεται να ξετυλιχθούν και ως εκ τούτου αποτυπώνονται ως κατευθυνόμενος κυκλικός γράφος[38].
Και τα δύο αυτά είδη RNN μπορούν να έχουν επιπλέον καταστάσεις (states) αποθήκευσης πληροφορίας που ελέγχονται από το ίδιο το νευρωνικό δίκτυο. Αυτά τα states αναφέρονται συνήθως ως πύλες (gates) και θα τις αναλύσουμε στη συνέχεια, όταν θα αναφερθούμε σε πολυπλοκότερες αρχιτεκτονικές επαναληπτικών νευρωνικών δικτύων. Για να κατανοήσουμε τη σπουδαιότητα των επαναληπτικών νευρωνικών δικτύων, θα χρησιμοποιήσουμε ένα παράδειγμα γλωσσικού μοντέλου (language model). Σε ένα τέτοιο μοντέλο ο στόχος είναι ο υπολογισμός της από κοινού πιθανότητας $P(x_1, x_2, ..., x_T)$, όπου $x_1, ...x_T$ ακολουθίες κειμένου (συνήθως ονομάζονται tokens ή characters) και T το συνολικό μήκος κειμένου.
Η πιθανότητα εμφάνισης μίας ακολουθίας κειμένου x_t τη δεδομένη χρονική στιγμή $t \in \{1, T\}$ είναι $P(x_t|x_1, ..., x_{t-1})$. Αντίστοιχα, αν θέλουμε να ενσωματώσουμε την πιθανή επίδραση των tokens νωρίτερα από το χρονικό βήμα $t-(n-1)$, όπου $n \geq 2$, στη x_t, η πιθανότητα τροποποιείται ως εξής $P(x_t|x_{t-1}, ..., x_{t-n+1})$. Ωστόσο, σε κάθε περίπτωση, ο αριθμός των παραμέτρων ενός μοντέλου που θα υπολογίσει αυτήν την πιθανότητα αυξάνει εκθετικά με το n, και ιδιαίτερα στην περίπτωση όπου $n = t$, καθώς πρέπει να αποθηκεύσουμε $|\mathbf{V}|^n$ αριθμούς για ένα σύνολο λεξιλογίου (σύνολο από tokens) $\mathbf{V}$. Ως εκ τούτου, θα ήταν πρότιμότερο να χρησιμοποιήσουμε ένα ελαφρώς λανθάνον μοντέλο που θα κάνει την εξής προσέγγιση:

$$P(x_t|x_1, ..., x_{t-1}) \approx P(x_t|h_{t-1}),$$

όπου το h_{t-1} είναι η κρυφή κατάσταση και αποθηκεύει τις πληροφορίες της ακολουθίας έως το χρονικό βήμα $t-1$. Στη γενική περίπτωση, το Hidden State σε κάθε χρονικό βήμα t μπορεί να υπολογιστεί με βάση τόσο την τρέχουσα είσοδο x_t όσο και από την προηγούμενη κρυφή κατάσταση, h_{t-1} ως:

$$h_t = f(x_t, h_{t-1}),$$

όπου f μία καταλλήλως ορισμένη σύναρτηση. Η κρυφή κατάσταση h_t είναι αυτή που εμπεριέχει όλη την πληροφορία που έχουμε δει έως τη χρονική στιγμή t.

Ας δούμε τώρα πως τροποποιούνται οι εξισώσεις (3.1.10),(3.1.11) για ένα επαναληπτικό νευρωνικό δίκτυο που διαθέτει κάποια τέτοια κρυφή κατάσταση, h_t. Ας υποθέσουμε, ότι έχουμε μια δέσμη (minibatch) της εισόδου, $X_t \in \mathbb{R}^{n \times d}$ τη χρονική στιγμή t. Με άλλα λόγια, για ένα minibatch που εμπεριέχει n σε πλήθος παραδείγματα ανά είσοδο, κάθε γράμμη του X_t αντιστοιχεί σε ένα τέτοιο παράδειγμα για το συγκεκριμένο χρονικό βήμα t. Ακόμη, με $H_t \in \mathbb{R}^{n \times h}$ σημειώνουμε την κρυφή κατάσταση κατά το χρονικό βήμα t.

Σε αντίθεση με το MLP, εδώ έχουμε αποθηκεύσει τη κρυφή κατάσταση H_{t-1} από το προηγούμενο χρονικό βήμα και για αυτό το λόγο εισάγουμε μία νέα παράμετρο βάρους $W_{hh} \in \mathbb{R}^{h \times h}$ για να περιγράψουμε το πως θα χρησιμοποιήσουμε αυτή τη κρυφή κατάσταση του προηγούμενου χρονικού βήματος στο τρέχον βήμα. Συγκεκριμένα, ο υπολογισμός της κρυφής κατάστασης του τρέχοντος χρονικού βήματος καθορίζεται τόσο από την είσοδο του τρέχοντος χρονικού βήματος όσο και από την κρυφή κατάσταση του προηγούμενου χρονικού βήματος ως εξής:

$$H_t = \phi(X_t W_{xh} + H_{t-1} W_{hh} + b_h) \tag{3.2.1}$$

Από τη σχέση (3.2.1) βλέπουμε ότι καθώς κάθε Hidden State H_t εξαρτάται από το (χρονικά) προηγουμένο του H_{t-1} και αυτό από αμέσως πιο πριν κ.ο.κ η ιστορικότητα της πληροφορίας των ακολουθιών εισόδου διατηρείται έως το τρέχον βήμα σε ικανοποιητικό βαθμό. Επομένως, αυτός είναι και ο λόγος που η μεταβλητή H ονομάζεται *κρυφή κατάσταση*.

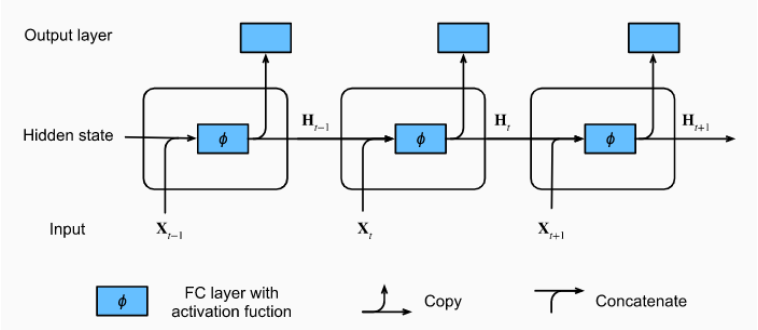
Η επανάληψη της σχέσης (3.2.1) σε κάθε χρονικό βήμα t είναι που καθιστά το δίκτυο *επαναληπτικό*. Οποιοδήποτε επίπεδο ενός νευρωνικού δικτύου εκτελεί τον υπολογισμό της εξίσωσης (3.2.1) ονομάζεται *επαναληπτικό επίπεδο*, ενώ στο σύνολό του το δίκτυο ονομάζεται *επαναληπτικό νευρωνικό δίκτυο*. Αντίστοιχα, με την εξίσωση (3.1.11) η έξοδος ενός RNN δίνεται για τη χρονική στιγμή t από την εξίσωση:

$$O_t = H_t W_{hq} + b_q \tag{3.2.2}$$

Στην εξίσωση (3.2.1) οι παράμετροι που χρησιμοποιήθηκαν περιλαμβάνουν τις κατάλληλες μήτρες βαρών $W_{xh} \in \mathbb{R}^{d\times h}, W_{hh} \in \mathbb{R}^{h\times h}$ και τον όρο bias $b_h \in \mathbb{R}^{1\times h}$ για το κρυφό επίπεδο και στην εξίσωση (3.2.2) περιλαμβάνουν τη μήτρα βαρών $W_{hq} \in \mathbb{R}^{h\times q}$ και τον όρο bias $b_q \in \mathbb{R}^{1\times q}$ για το επίπεδο εξόδου. Αξίζει να σημειωθεί ότι ακόμη και σε διαφορετικά χρονικά βήματα, τα RNNs χρησιμοποιούν πάντα αυτές τις παραμέτρους μοντέλου. Επομένως, το κόστος παραμετροποίησης ενός RNN δεν αυξάνεται καθώς αυξάνεται ο αριθμός των βημάτων χρόνου.

Στην εικόνα 3.4 απεικονίζουμε ένα RNN σε τρία συνεχόμενα χρονικά βήματα. Σε οποιαδήποτε χρονική στιγμή t η έξοδος του Hidden State μπορεί να υπολογιστεί ως εξής: i) συνενώνοντας (concatenation) την είσοδο X_t της τρέχουσας χρονικής στιγμής t και την κρυφή κατάσταση H_{t-1} στο προηγούμενο χρονικό βήμα $t-1$ και ii) τροφοδοτώντας το αποτέλεσμα της συνένωσης σε ένα πλήρως συνδεδεμένο επίπεδο (fully-connected layer) με συνάρτηση ενεργοποίησης ϕ. Η έξοδος ενός τέτοιου πλήρους συνδεδεμένου επιπέδου είναι η κρυφή κατάσταση H_t του τρέχοντος χρονικού βήματος t.

Στη συνένωση των X_t και H_{t-1} συμμετείχαν οι μήτρες βαρών W_{xh} και W_{hh} και ο όρος bias b_h, όλα από την εξίσωση (3.2.1). Η κρυφή κατάσταση του τρέχοντος βήματος t, H_t, θα συμμετάσχει στον υπολογισμό της κρυφής κατάστασης H_{t+1} του επόμενου βήματος $t+1$. Επιπλέον, θα τροφοδοτηθεί στο πλήρως συνδεδεμένο στρώμα εξόδου για να υπολογιστεί η έξοδος O_t για αυτήν τη χρονική στιγμή t.



Εικόνα 3.4: Ένα RNN με ένα hidden layer σε τρία χρονικά βήματα [39]

3.2.2 Αλγόριθμοι Εκπαίδευσης Επαναληπτικών Νευρωνικών Δικτύων

Ο τρόπος εκπαίδευσης των επαναληπτικών νευρωνικών δικτύων γίνεται μέσω του αλγορίθμου *οπισθοδιάδοσης διαμέσω χρόνου* (Back Propagation through time). Είναι μία ειδική εφαρμογή του αλγορίθμου *οπισθοδιάδοσης* (Back Propagation) [46] που αποτελεί τον αλγόριθμο εκπαίδευσης των τεχνητών νευρωνικών δικτύων όπως το MLP. Στην παράγραφο αυτή παρουσιάζεται ο αλγόριθμος εκπαίδευσης για τα επαναληπτικά νευρωνικά δίκτυα [40].
Για λόγους απλοποίησης, θεωρούμε ένα RNN χωρίς όρους bias του οποίου η συνάρτηση ενεργοποίησης στο κρυφό επίπεδο είναι η ταυτοτική συνάρτηση ($\phi(x) = x$). Για κάθε χρονικό βήμα t, θεωρούμε το διάνυσμα εισόδου και την αντίστοιχη ετικέτα να είναι $x_t \in \mathbb{R}^d$ και y_t. Έτσι, η κρυφή κατάσταση θα είναι $h_t \in \mathbb{R}^h$, η έξοδος $o_t \in \mathbb{R}^q$ και υπολογίζονται ως:

$$h_t = W_{hx}x_t + W_{hh}h_{t-1} \tag{3.2.3α'}$$

$$o_t = W_{qh}h_t \tag{3.2.3β'}$$

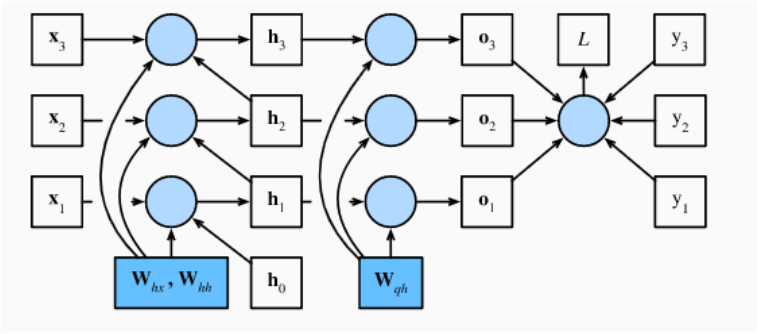
όπου $W_{hx} \in \mathbb{R}^{h\times d}$, $W_{hh} \in \mathbb{R}^{h\times h}$ και $W_{qh} \in \mathbb{R}^{q\times h}$. Συμβολίζουμε με $l(o_t, y_t)$ την *απώλεια* (loss) στο στιγμιαίο βήμα t, δηλαδή τη διαφορά της εξόδου o_t και της ετικέτας y_t. Συνεπώς, η αντικειμενική *συνάρτηση απωλειών* (loss function) για συνολικά T χρονικά βήματα από την αρχή της ακολουθίας εισόδου θα είναι λοιπόν:

$$L = \frac{1}{T}\sum_{t=1}^{T} l(o_t, y_t) \tag{3.2.4}$$

Προκειμένου να απεικονίσουμε τις εξαρτήσεις μεταξύ των μεταβλητών και των παραμέτρων του μοντέλου κατά τον υπολογισμό της εξόδου του RNN, μπορούμε να σχεδιάσουμε το υπολογιστικό γράφημα για το μοντέλο, όπως φαίνεται στο σχήμα της εικόνας 3.5. Για παράδειγμα, ο υπολογισμός της κρυφής κατάστασης του τρίτου χρονικού βήματος, h_3, εξαρτάται από τις παραμέτρους του μοντέλου W_{hx} και W_{hh}, την κρυφή κατάσταση του προηγούμενου βήματος h_2 και την είσοδο του τρέχοντος χρονικού βήματος x_3.

Γενικά, η εκπαίδευση ενός τέτοιου μοντέλου απαιτεί τον υπολογισμό της κλίσης της συνάρτησης απωλειών σε σχέση με αυτές τις παραμέτρους, δηλαδή: $\partial L/\partial W_{hx}$, $\partial L/\partial W_{hh}$ και $\partial L/\partial W_{qh}$. Για να 'φανερωθούν' οι εξαρτήσεις μεταξύ των μεγεθών, μπορούμε να 'διασχίσουμε' το γράφημα της εικόνας 3.5 προς την αντίθετη κατεύθυνση των βελών, υπολογίζοντας και αποθηκεύοντας τις εκάστοτε κλίσεις (παραγώγους).
Για μεγαλύτερη ευκολία στην έκφραση του πολλαπλασιασμού πινάκων και διανυσμάτων διαφορετικών διαστάσεων στον κανόνα της αλυσίδας, εισάγουμε τον τελεστή **prod**. Υποθέτοντας, ότι έχουμε 2 συναρτήσεις $\mathsf{Y} = f(\mathsf{X})$ και $\mathsf{Z} = g(\mathsf{Y})$, όπου οι είσοδοι και οι έξοδοι $\mathsf{X}, \mathsf{Y}, \mathsf{Z}$ είναι τανυστές αυθαίρετων διαστάσεων. Χρησιμοποιώντας τον κανόνα της αλυσίδας, μπορούμε να υπολογίσουμε το παράγωγο του Z ως προς το X μέσω της σχέσης:

$$\frac{\partial \mathsf{Z}}{\partial \mathsf{X}} = \text{prod}\left(\frac{\partial \mathsf{Z}}{\partial \mathsf{Y}}, \frac{\partial \mathsf{Y}}{\partial \mathsf{X}}\right) \tag{3.2.5}$$



Εικόνα 3.5: Υπολογιστικό γράφημα που αναδεικνύει τις εξαρτήσεις για ένα μοντέλο RNN σε τρία χρονικά βήματα. Τα πλαίσια αντιπροσωπεύουν μεταβλητές (όχι σκιασμένα) ή παραμέτρους (σκιασμένα) και οι κύκλοι αντιπροσωπεύουν τελεστές πράξεων [41]

Η χρήση του τελεστή prod εμπερικλείει όλες τις απαραίτητες ενέργειες που θα πρέπει να γίνουν στους τανυστές-ορίσματα X, Y, Z, όπως αντιστροφή και η εναλλαγή θέσης των εισόδων. Για εισόδους διανύσματα, αυτό είναι απλό καθώς μιλάμε για πολλαπλασιασμό πινάκων. Για τανυστές υψηλότερης διάστασης, αυτό γίνεται ιδιαίτερα περίπλοκο. Με τη βοήθεια αυτού του τελεστή μπορούμε να συνεχίσουμε την ανάλυση μας.
Η διαφόριση της συνάρτησης απωλείων ως προς την έξοδο του μοντέλου o_t για κάποια χρονική στιγμή t είναι αρκετά απλή και δίνεται από τη σχέση:

$$\frac{\partial L}{\partial o_t} = \frac{\partial l(o_t, y_t)}{T \cdot \partial o_t} \in \mathbb{R}^q \tag{3.2.6}$$

Μπορούμε να υπολογίσουμε την κλίση της συνάρτησης απωλειών ως προς τα βάρη W_{qh} στο επίπεδο εξόδου, $\partial L/\partial W_{qh} \in \mathbb{R}^{q \times h}$. Βασιζόμενοι στην εικόνα 3.5 ξέρουμε ότι L εξαρτάται από το W_{qh} μέσω των εξόδων $o_1, \ldots, o_T$, οπότε χρησιμοποιώντας τον κανόνα της αλυσίδας έχουμε:

$$\frac{\partial L}{\partial W_{qh}} = \sum_{t=1}^{T} \text{prod}\left(\frac{\partial L}{\partial o_t}, \frac{\partial o_t}{\partial W_{qh}}\right) = \sum_{t=1}^{T} \frac{\partial L}{\partial o_t} h_t^\top \tag{3.2.7}$$

όπου η μερική παράγωγος $\partial L/\partial o_t$ δίνεται από την (3.2.6).

Στη συνέχεια, όπως φαίνεται και πάλι στην εικόνα 3.5, στο τελευταίο χρονικό βήμα T η συνάρτηση απωλειών L εξαρτάται από την κρυφή κατάσταση h_T μόνο μέσω της o_T. Συνεπώς, η κλίση $\partial L/\partial h_T \in \mathbb{R}^h$ μπορεί να βρεθεί εύκολα κάνοντας και πάλι χρήση του κανόνα αλυσίδας:

$$\frac{\partial L}{\partial h_T} = \text{prod}\left(\frac{\partial L}{\partial o_T}, \frac{\partial o_T}{\partial h_T}\right) = W_{qh}^\top \frac{\partial L}{\partial o_T} \tag{3.2.8}$$

Τα πράγματα περιπλέκονται ελαφρώς για οποιοδήποτε $t < T$, όπου η συνάρτηση απωλειών L εξαρτάται από το h_t μέσω του h_{t+1} και του o_t.

Σύμφωνα και πάλι με τον κανόνα της αλυσίδας, η κλίση $\partial L/\partial h_t \in \mathbb{R}^h$ για οποιοδήποτε χρονικό βήμα $t < T$ μπορεί να υπολογιστεί επαναληπτικά ως:

$$\frac{\partial L}{\partial h_t} = \text{prod}\left(\frac{\partial L}{\partial h_{t+1}}, \frac{\partial h_{t+1}}{\partial h_t}\right) + \text{prod}\left(\frac{\partial L}{\partial o_t}, \frac{\partial o_t}{\partial h_t}\right) = W_{hh}^\top \frac{\partial L}{\partial h_{t+1}} + W_{qh}^\top \frac{\partial L}{\partial o_t} \quad (3.2.9)$$

Ξεδιπλώνοντας την προηγούμενη αναδρομική σχέση (3.2.9) για οποιοδήποτε χρονικό βήμα $1 \leq t \leq T$ μπορούμε να πάρουμε:

$$\frac{\partial L}{\partial h_t} = \sum_{i=t}^{T} \left(W_{hh}^\top\right)^{T-i} W_{qh}^\top \frac{\partial L}{\partial o_{T+t-i}} \quad (3.2.10)$$

Ωστόσο, μπορούμε να δούμε ότι η σχέση (3.2.10) ενέχει κάποια προβλήματα: όταν οι ακολουθίες εισόδου του μοντέλου έχουν μεγάλο μήκος τότε οδηγούμαστε δυνητικά σε πολύ μεγάλες δυνάμεις του $W_{hh}^\top$. Οι ιδιότιμές αυτής της μήτρας που είναι μικρότερες από 1 μετά από αλλεπάλληλες αναδρομές εξαφανίζονται, ενώ οι ιδιότιμές μεγαλύτερες του 1 αποκλίνουν. Έτσι, οδηγούμαστε σε αριθμητική αστάθεια, το οποίο εκδηλώνεται είτε υπό τη μορφή εξαφανιζόμενων είτε υπό τη μορφή εκρηγνυόμενων κλίσεων (παραγώγων) καθώς αυξάνονται οι επαναλήψεις. Ένας τρόπος για να αντιμετωπιστεί αυτό είναι να περικοπούν τα χρονικά βήματα σε ένα υπολογιστικά βολικό μέγεθος. Στην πράξη, αυτή η περικοπή πραγματοποιείται με κατάργηση του υπολογισμού της κλίσης από ένα δεδομένο χρονικό βήμα και μετά.

Τέλος, το σχήμα της εικόνας 3.5 δείχνει ότι η συνάρτηση απωλειών L εξαρτάται από τις παραμέτρους του μοντέλου W_{hx} και W_{hh} στο κρυφό επίπεδο μέσω των κρυφών καταστάσεων $h_1, \ldots, h_T$. Για να υπολογίσουμε την κλίση της L ως προς αυτές τις παραμέτρους, $\partial L/\partial W_{hx} \in \mathbb{R}^{h \times d}$ και $\partial L/\partial W_{hh} \in \mathbb{R}^{h \times h}$, εφαρμόζουμε για άλλη μία φορά τον κανόνα της αλυσίδας ως:

$$\frac{\partial L}{\partial W_{hx}} = \sum_{t=1}^{T} \text{prod}\left(\frac{\partial L}{\partial h_t}, \frac{\partial h_t}{\partial W_{hx}}\right) = \sum_{t=1}^{T} \frac{\partial L}{\partial h_t} x_t^\top \quad (3.2.11\alpha')$$

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \text{prod}\left(\frac{\partial L}{\partial h_t}, \frac{\partial h_t}{\partial W_{hh}}\right) = \sum_{t=1}^{T} \frac{\partial L}{\partial h_t} h_{t-1}^\top \quad (3.2.11\beta')$$

όπου η ποσότητα $\partial L/\partial h_t$ που υπολογίζεται επαναληπτικά από τις σχέσεις (3.2.8) και (3.2.9) είναι ο βασικός παράγοντας που επηρεάζει την αριθμητική σταθερότητα.

Αν και ιδιαιτέρως περίπλοκος και χρονοβόρος ο αλγόριθμος της οπισθοδιάδοσης διαμέσω χρόνου έχει ένα θετικό σημείο: αποθηκεύει τις ενδιάμεσες μεταβλητές του αλγορίθμου με τη σειρά υπολογισμού τους για να αποφεύγονται οι αχρείαστοι επιπρόσθετοι υπολογισμοί. Για παράδειγμα, η αποθήκευση της $\partial L/\partial h_t$ που χρησιμοποιείται στον υπολογισμό των $\partial L/\partial W_{hx}$ και $\partial L/\partial W_{hh}$ επιταχύνει σημαντικά το χρόνο εκτέλεσης του αλγορίθμου.

Ωστόσο, όπως προαναφέρθηκε, τα σοβαρότερο πρόβλημα που ενδέχεται να προκαλέσει ο αλγόριθμος Back Propagation through time στα RNNs είναι ότι για ιδιαίτερα υψήλες δυνάμεις πινάκων μπορούμε να οδηγηθούμε σε αποκλίνουσες ή εξαφανιζόμενες

ιδιότιμές. Αυτό σημαίνει, ότι οι κλίσεις (παράγωγοι) κατά την εκτέλεση του αλγορίθμου εκρήγνυνται σε πολύ μεγάλες τιμές ή εξαφανίζονται σε πολύ μικρές. Αποτέλεσμα αυτού, είναι ότι τα πρώτα στάδια ενός δικτύου αποτελούμενο από n στρώσεις θα έχουν είτε πολύ μικρές ανανεώσεις στα βάρη τους και άρα δεν θα εκπαιδεύονται ικανοποιητικά είτε πολύ μεγάλες, με συνέπεια το δίκτυο να φτάσει σε σημείο κορεσμού και να σταματήσει την εκπαίδευσή του. Το προβλημά αυτό είναι γνωστό με το όνομα *Εξαφάνιση ή Έκρηξη της παραγώγου* (Vanishing/Exploding gradient Problem)[42] και συναντάται σύχνα στα τεχνητά Νευρωνικά Δίκτυα που εκπαιδεύονται με βάση τους αλγοριθμούς της *Κατάβασης Πλαγιάς* (Gradient Descent) και του Back Propagation. Για το λόγο αυτό υπήρξε ανάγκη για εξεύρεση περιπλοκότερων ακολουθιακών μοντέλων που θα μπορούσαν να αντιμετωπίσουν αυτό το πρόβλημα.

3.3 Επεκτάσεις των Επαναληπτικών Νευρωνικών Δικτύων

Ο κύριος λόγος για τον οποίο υπήρξε ανάγκη για επέκταση των RNNs ήταν η αντιμέτωπιση του προβλήματος εξαφανιζόμενης παραγώγου στα πιο περίπλοκα πρόβλήματα. Η λύση για πρώτη φόρα σε αυτό το πρόβλημα δόθηκε από τους Sepp Hochreiter και Jürgen Schmidhuber το 1997 [43] οι οποίοι ανέπτυξαν ένα είδος επαναληπτικού νευρωνικού δικτύου με πολυπλοκότερη αρχιτεκτονική από το συνηθισμένο RNN, το οποίο ονομάστηκε δίκτυο *Μακράς Βραχυπρόθεσμης Μνήμης*, (Long Short-term Memory-LSTM Neural Network).

Σχεδόν δύο δεκαετίες αργότερα, το 2014 ο Kyunghyun Cho [44] πρότεινε μια απλούστερη αρχιτεκτονική δικτύου από αυτή του LSTM, που όμως έδειξε την ίδια απότελεσματικότητα σε πληθώρα προβλημάτων. Η αρχιτεκτονική αυτή ονόμαστηκε δίκτυο *Φραγμένο Επαναληπτικό Δίκτυο* (Gated Recurrent Unit-GRU). Και τα δύο είδη δικτύων είναι αρκετά παρόμοια και βασίζονται σε καταστάσεις που τους επιτρέπουν να διατηρούν ή να διαχειρίζονται την εισαγόμενη πληροφορία και ονομάζονται *πύλες*, ενώ έχουν δείξει ιδιαίτερα ενθαρρυντικά αποτελέσματα σε προβλήματα, όπως η ταξινόμηση στατικών δεδομένων, η επεξεργασία ακολουθιακών δεδομένων και φυσικής γλώσσας ή η πρόβλεψη σε δεδομένα χρονοσειρών (κατηγορία στην οποία έγκειται και το δικό μας πρόβλημα).

3.3.1 Φραγμένα Επαναληπτικά Δίκτυα (GRUs)

Η διαφορά του απλού RNN με το **φραγμένο επαναληπτικό δίκτυο** (GRU) είναι ότι το τελευταίο υποστηρίζει πολυπλοκότερο μηχανισμό πύλης στο δικό του Hidden State που του επιτρέπει να ανανεώνει ή να επαναφέρει την απαραίτητη πληροφορία. Ο μηχανισμός αυτός χωρίζεται σε 2 υπο-πύλες: α) Την *Πύλη Ανανέωσης* (Update Gate) η οποία ελέγχει το βαθμό στον οποίο η νέα κρυφή κατάσταση ομοιάζει με αυτή του προηγούμενου χρονικού βήματος t και β) *την Πύλη Επαναφοράς* (Reset Gate) η οποία καθορίζει πόση πληροφορία από το παρελθόν θέλουμε να κρατήσουμε στην κρυφή κατάσταση, του επόμενου χρονικού βήματος $t-1$.

Οι είσοδοι και των δύο πυλών σε κάθε χρονικό βήμα είναι η συνένωση της εισόδου στο τρέχον χρονικό βήμα με την κρυφή κατάσταση της προηγούμενης χρονικής στιγμής.

Οι έξοδοι των δύο πυλών δίδονται από δύο πλήρως διασυνδεδεμένα στρώματα με συνάρτηση ενεργοποίησης τη σιγμοειδή συνάρτηση (simoid (σ)). Οπότε, αν υποθέσουμε ότι τη χρονική στιγμή t, έχουμε ένα minibatch εισόδου $X_t \in \mathbb{R}^{n\times d}$ (n: τα παραδείγματα ανά είσοδο, d: ο αριθμός εισόδων) και την κρυφή κατάσταση του προηγούμενου χρονικού βήματος είναι $H_{t-1} \in \mathbb{R}^{n\times h}$ (h : ο αριθμός κρυφών νευρώνων). Τότε, η πύλη επαναφοράς $R_t \in \mathbb{R}^{n\times h}$ και η πύλη ανανέωσης $Z_t \in \mathbb{R}^{n\times h}$ υπολογίζονται ως εξής:

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \tag{3.3.1}$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \tag{3.3.2}$$

όπου $W_{xr}, W_{xz} \in \mathbb{R}^{d\times h}$ και $W_{hr}, W_{hz} \in \mathbb{R}^{h\times h}$ είναι παράμετροι βαρών σε μητρική μορφή και $b_r, b_z \in \mathbb{R}^{1\times h}$ είναι όροι bias επίσης σε μητρική μορφή. Χρησιμοποιούμε τη σιγμοειδή συνάρτηση για να μετατρέψουμε τις τιμές εισόδου στο διάστημα $(0, 1)$.

Στη συνέχεια, ενσωματώνουμε την πύλη επαναφοράς R_t στο μηχανισμό ενημέρωσης λανθάνουσας κατάστασης της εξίσωσης (3.2.1). Αυτό οδηγεί στη δημιουργία μίας νέας κατάστασης που αποκαλείται *υποψήφια κρυφή κατάσταση* (candidate hidden state) $\tilde{H}_t \in \mathbb{R}^{n\times h}$ για την χρονική στιγμή t και δίνεται από την εξίσωση:

$$\tilde{H}_t = \tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h) \tag{3.3.3}$$

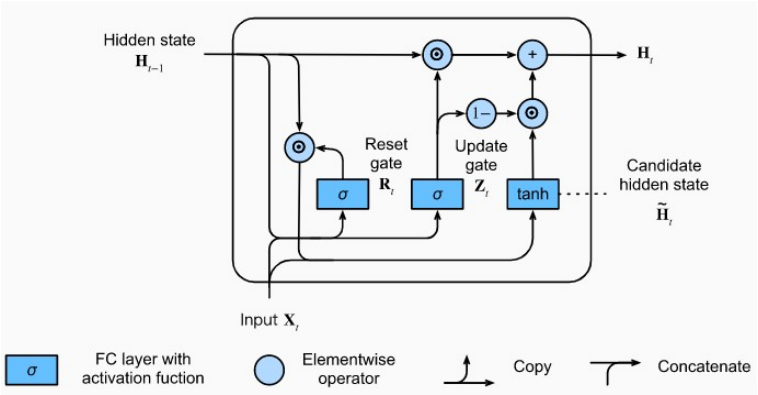
Και πάλι $W_{xh} \in \mathbb{R}^{d\times h}$ και $W_{hh} \in \mathbb{R}^{h\times h}$ είναι παράμετροι βαρών σε μητρική μορφή ενώ $b_h \in \mathbb{R}^{1\times h}$ είναι ο όρος bias. Να τονιστεί, ότι το σύμβολο $\odot$ είναι ο τελεστής Hadamard και υποδηλώνει τον πολλαπλασιασμό πινάκων στοιχείο με στοιχείο (elementwise product operator), ενώ χρησιμοποιούμε τη μη γραμμική συνάρτηση tanh για να διασφαλίσουμε ότι η τιμή της υποψήφιας κρυφής κατάστασης θα παραμείνει στο διάστημα $(-1, 1)$. Το αποτέλεσμα αυτό το αποκαλούμε *υποψήφιο*, καθώς δεν έχουμε ακόμη ενσωματώσει τη δράση της πύλης ανανέωσης στον υπολογισμό της κρυφής κατάστασης για τη χρονική στιγμή t. Σε σύγκριση με την (3.2.1), τώρα η επίδραση των προηγούμενων καταστάσεων μπορεί να μειωθεί με το στοιχειώδη πολλαπλασιασμό των R_t και H_{t-1}. Κάθε φορά που οι καταχωρήσεις στην πύλη επαναφοράς R_t είναι κοντά στο 1, ανακτούμε ένα απλό RNN όπως αυτό της εξίσωσης (3.2.1). Για όλες τις καταχωρήσεις της πύλης επαναφοράς R_t που είναι κοντά στο 0, η υποψήφια κρυφή κατάσταση είναι το αποτέλεσμα ενός MLP με είσοδο X_t, όπως αυτό της εξίσωσης (3.1.10). Έτσι, οποιαδήποτε προϋπάρχουσα κρυφή κατάσταση επαναφέρεται σε μία αρχική τιμή που είχε τεθεί κάποια χρονική στιγμή t στο παρελθόν.

Τέλος, πρέπει να συμπεριλάβουμε την επίδραση της πύλης ανανέωσης Z_t. Αυτή καθορίζει την έκταση στην οποία η νέα κρυφή κατάσταση ομοιάζει με την παλιά κατάσταση H_{t-1} και κατά πόσο χρησιμοποιείται η νέα υποψήφια κατάσταση $\tilde{H}_t$. Λαμβάνοντας τον πολλαπλασιασμό στοιχείο προς στοιχείο της Z_t με τα H_{t-1}, $\tilde{H}_t$ οδηγούμαστε στην τελική εξίσωση ενημέρωσης της κρυφής κάτασταση για το GRU:

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t \tag{3.3.4}$$

Όταν η πύλη ενημέρωσης Z_t είναι κοντά στο 1, διατηρούμε απλώς την παλιά κατάσταση. Σε αυτήν την περίπτωση, οι πληροφορίες από το X_t ουσιαστικά αγνοούνται, παρακάμπτοντας έτσι το βήμα t στην αλυσίδα εξάρτησης της πληροφορίας. Αντιθέτως, όποτε η

Z_t είναι κοντά στο 0, το νέα κρυφή κατάσταση H_t πλησιάζει πιο πολύ στην υποψήφια λανθάνουσα κατάσταση $\tilde{H}_t$. Κατά αυτόν τον τρόπο, η αρχιτεκτονική της μονάδας GRU μας βοηθά να αντιμετωπίσουμε το πρόβλημα της *εκλιπούσας παραγώγου* που συναντάται στα απλά RNN, αλλά και να συγκρατήσουμε καλύτερα τις αλληλοεξαρτήσεις ακολουθιών με μεγάλες αποστάσεις χρονικών βημάτων. Στην εικόνα 3.6 βλέπουμε τη δομή ενός κελιού GRU στο οποίο η έξοδος είναι η νέα κρύφη κατάσταση H_t.



Εικόνα 3.6: Το κελί GRU για τον υπολογίσμο του H_t [45]

3.3.2 Δίκτυα Μακράς Βραχυπρόθεσμης Μνήμης (LSTMs)

Η πρόκληση για την αντιμετώπιση της μακροπρόθεσμης διατήρησης πληροφοριών αλλά και η ανάγκη για απαλλαγή από το πρόβλημα της εκλιπούσας παραγώγου υπήρχε για πολύ καιρό, ενώ όπως προαναφέρθηκε, λύση για πρώτη φορά σε αυτό το θέμα δόθηκε το 1997 με τη δημιουργία του επαναληπτικού νευρωνικού δικτύου **μακράς βραχυπρόθεσμης μνήμης** - LSTM. Το LSTM μοιράζεται πολλές από τις ιδιότητες του GRU, έχοντας όμως ελαφρώς πολυπλοκότερο μηχανισμό. Το LSTM εισάγει ένα νέο είδος κελιού/κατάστασης που ονομάζεται *κελί μνήμης* (memory cell), που σχεδιάστηκε για την καταγραφή πρόσθετων πληροφοριών, και ομοιάζει αρκέτα με την γνωστή μας κρυφή κατάσταση. Για να ελέγξουμε το κελί μνήμης χρειαζόμαστε κάποιες πύλες. Απαιτείται μία πύλη, για να διαβάσει τις καταχωρήσεις από το κελί μνήμης, την οποία θα αναφέρουμε και ως *Πύλη Εξόδου* (Output Gate). Ακόμη, μια δεύτερη πύλη χρειάζεται για να αποφασίσουμε πότε να διαβάσουμε δεδομένα από το κελί μνήμης, το όνομα της οποίας είναι *Πύλη Εισόδου* (Input Gate). Τέλος, χρειαζόμαστε ένα μηχανισμό για επαναφορά του περιεχομένου του κελιού μνήμης. Τη δουλειά αυτή αναλαμβάνει η λεγόμενη *Πύλη Λήθης* (Forget Gate).

Ακριβώς όπως στην περίπτωση του GRU, τα δεδομένα που εισάγονται στις πύλες του LSTM είναι η συνένωση της εισόδου στο τρέχον χρονικό βήμα με την κρυφή κατάσταση της προηγούμενης χρονικής στιγμής. Εκεί, τα δεδόμενα επεξεργάζονται από 3 πλήρως διασυνδεδεμένα στρώματα με συνάρτηση ενεργοποίησης τη σιγμοειδή συνάρτηση. Η έξοδος αυτών των στρώματων είναι οι 3 πύλες που μόλις αναφέραμε. Αποτέλεσμα της χρήσης της σιγμοειδούς είναι οι τιμές των πυλών να είναι εντός του διαστήματος $(0, 1)$.

Μαθηματικώς, ας υποθέσουμε ότι έχουμε h σε αριθμό κρυφούς νευρώνες, το μέγεθος του batch size είναι n και ο αριθμός των εισόδων είναι d. Έτσι, η είσοδος θα είναι $X_t \in \mathbb{R}^{n\times d}$ και η κρυφή κατάσταση του προηγούμενου χρονικού βήματος θα είναι $H_{t-1} \in \mathbb{R}^{n\times h}$. Αντίστοιχα, οι 3 πύλες στο χρονικό βήμα t ορίζονται ως εξής: η πύλη εισόδου ως $I_t \in \mathbb{R}^{n\times h}$, η πύλη λήθης ως $F_t \in \mathbb{R}^{n\times h}$ και η πύλη εξόδου ως $O_t \in \mathbb{R}^{n\times h}$. Υπολογίζονται δε, ως:

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \tag{3.3.5}$$

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f) \tag{3.3.6}$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \tag{3.3.7}$$

όπου $W_{xi}, W_{xf}, W_{xo} \in \mathbb{R}^{d\times h}$ και $W_{hi}, W_{hf}, W_{ho} \in \mathbb{R}^{h\times h}$ είναι παράμετροι βαρών σε μητρική μορφή και $b_i, b_f, b_o \in \mathbb{R}^{1\times h}$ είναι παράμετροι bias επίσης σε μητρική μορφή.

Σε αντιστοιχία με την *υποψήφια κρυφή κατάσταση* στο GRU, η αρχιτεκτονική του LSTM διαθέτει μία ενδιάμεση κατάσταση προκείμενου να ελέγξει το πώς και κατά πόσο θα ανανεώσει το κελί μνήμης. Αυτή η κατάσταση ονομάζεται *υποψήφιο κελί μνήμης* και συμβολίζεται με $\tilde{C}_t \in \mathbb{R}^{n\times h}$. Ονομάζεται δε, υποψήφιο κάθως ακόμα δεν έχει ολοκληρωθεί η ροή της πληροφόριας μέσα στο κελί LSTM δεδόμενου ότι δεν έχουμε ανάφερει όλες τις λειτουργίες των πυλών. Ο υπολογισμός για αυτό το κελί είναι παρόμοιος, με αυτόν των τριών πυλών που αναφέρθηκαν παραπάνω, αλλά εδώ χρησιμοποιούμε τη συνάρτηση $tanh$ με εύρος τιμών το $(-1, 1)$ ως συνάρτηση ενεργοποίησης. Αυτό μας οδηγεί στην ακόλουθη εξίσωση για το χρονικό βήμα t:

$$\tilde{C}_t = tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c) \tag{3.3.8}$$

όπου $W_{xc} \in \mathbb{R}^{d\times h}$ και $W_{hc} \in \mathbb{R}^{h\times h}$ μήτρες βαρών και $b_c \in \mathbb{R}^{1\times h}$ το διάνυσμα παραμέτρων bias.

Στο GRU, είχαμε ένα μηχανισμό που διέπει την είσοδο αλλά και το πόση πληροφορία παραλείπουμε στη συνέχεια. Ομοίως, στο LSTM έχουμε δύο αποκλειστικές πύλες για αυτούς τους σκοπούς. Η πύλη εισόδου I_t είναι αυτή που καθορίζει το πόσο λαμβάνουμε υπόψη μας τα νέα δεδομένα μέσω του υποψήφιου κελιού $\tilde{C}_t$, ενώ η πύλη λήθης F_t διευθετεί την ποσότητα που θα διατηρήσουμε από το παλαίο κελί μνήμης (της προηγούμενης χρονικής στιγμής) $C_{t-1} \in \mathbb{R}^{n\times h}$. Χρησιμοποιώντας τον ίδιο πολλαπλασιασμό (στοιχείο προς στοιχείο) όπως πριν, φτάνουμε στην ακόλουθη εξίσωση ενημέρωσης του κελιού μνήμης:

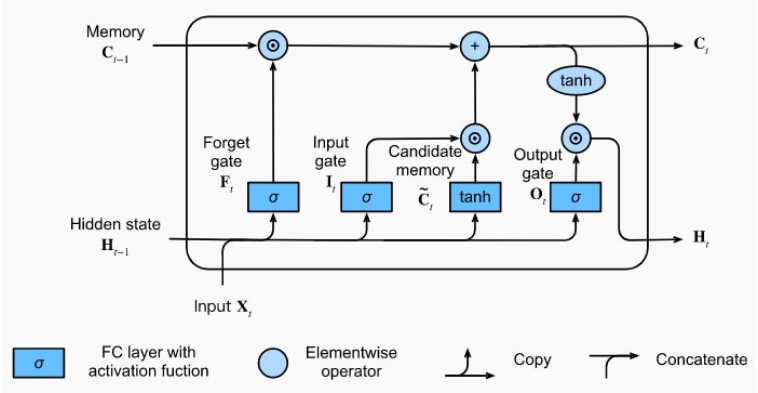
$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t \tag{3.3.9}$$

Εάν η πύλη λήθης είναι κοντά στο 1 και η πύλη εισόδου είναι κοντά στο 0, το προηγούμενο κελί μνήμης C_{t-1} θα αποθηκευτεί και θα περάσει αυτό στο τρέχον βήμα χρόνου t. Καθ' όλη τη διάρκεια της ανάλυσής μας για την αρχιτεκτονική του κελιού LSTM δεν αναφερθήκαμε σχεδόν καθόλου στο βασικό θεμέλιο των επαναληπτικών νευρωνικών δικτύων, την *κρυφή κατάσταση* (Hidden State). Την ορίζουμε και εδώ ως $H_t \in \mathbb{R}^{n\times h}$ και αποτελεί το τελευταίο στοιχείο που θα ανανεωθεί κατά τη διάρκεια του

τρέχοντος βήματος. Για τον υπολογισμό της συμβάλλει τόσο το μόλις ανανεωμένο κελί μνήμης, το οποίο όμως εισάγουμε από την συνάρτηση $tanh$, όσο και η πύλη εξόδου O_t. Αυτό διασφαλίζει ότι οι τιμές της H_t είναι πάντα στο διάστημα $(-1, 1)$. Συνεπώς, καταλήγουμε στην ακόλουθη εξίσωση (που όπως φαίνεται κάνουμε και πάλι χρήση του πολλαπλασιασμού στοιχείο προς στοιχείο):

$$H_t = O_t \odot \tanh(C_t) \tag{3.3.10}$$

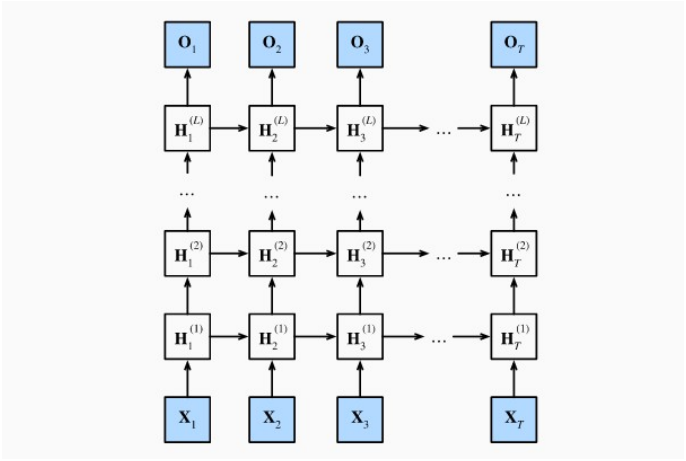
Οποτεδήποτε η πύλη εξόδου πλησιάζει την τιμή 1, μεταδίδουμε αποτελεσματικά όλες τις πληροφορίες από τα νεοεισαχθέντα δεδομένα εισόδου στο επόμενο χρονικό βήμα, ενώ για τιμές της πύλης εξόδου κοντά στο 0 διατηρούμε όλες τις πληροφορίες που έχουμε μέσα στο κελί μνήμης και δεν πραγματοποιούμε περαιτέρω επεξεργασία. Ο σχεδιασμός αυτός λοιπόν, που περιγράφεται από τις παραπάνω εξισώσεις, προτάθηκε για την αντιμετώπιση του προβλήματος της εξαφανιζόμενης παραγώγου αλλά και για την καλύτερη καταγραφή εξαρτήσεων μεγάλης εμβέλειας εντός των ακολουθιών. Τέλος, μία αντιπροσωπευτική εικόνα για τη δομή του LSTM παρουσιάζουμε στην εικόνα 3.7 στην οποία η έξοδος είναι τόσο η νέα κρύφη κατάσταση H_t όσο και το νέο κελί μνήμης C_t.



Εικόνα 3.7: Το κελί LSTM για τον υπολογίσμο των C_t,H_t [47]

3.3.3 Βαθιά Επαναληπτικά Νευρωνικά Δίκτυα (Deep RNNs)

Μέχρι τώρα, συζητήσαμε για RNN με ένα μόνο κρυφό επίπεδο. Θα μπορούσαμε ωστόσο, να στοιβάξουμε πολλαπλά κρυφά επίπεδα από RNNs το ένα πάνω στο άλλο, δημιουργώντας μία στοίβα από κρυφά επίπεδα. Αυτό οδηγεί σε έναν ευέλικτο μηχανισμό, λόγω του συνδυασμού πολλών απλών κρυφών επιπέδων. Συγκεκριμένα, τα δεδομένα σε διαφορετικά επίπεδα της στοίβας ενδέχεται να είναι συσχετισμένα καλύτερα με συνέπεια στην έξοδο να έχουμε και καλύτερα αποτελέσματα. Κατ' αυτόν τον τρόπο εισάγουμε τον όρο *βαθύ επαναληπτικό νευρωνικό δίκτυο* [48]. Μία απλουστευμένη απεικόνιση ενός βαθιού επαναληπτικού νευρωνικού δικτύου φαίνεται στην εικόνα 3.8, όπου περιγράφεται ένα βαθύ RNN με L κρυφά επίπεδα. Βλέπουμε, ότι κάθε κρυφή κατάσταση μεταφέρεται συνεχώς τόσο στο επόμενο βήμα χρόνου του τρέχοντος επιπέδου όσο και στο τρέχον βήμα χρόνου του επόμενου επιπέδου.



Εικόνα 3.8: Η αρχιτεκτονική ενός βαθιού RNN [49]

Σε αυτό το σημείο θα παρουσιάσουμε τις εξισώσεις που διέπουν ένα τέτοιας μορφής δίκτυο, όπως αυτό της εικόνας 3.8. Η ανάλυσή μας επικεντρώνεται στο απλό μοντέλο RNN από L κρυφά επίπεδα, αλλά ισχύει και για τα άλλα μοντέλα που περιγράψαμε. Υποθέτουμε ότι έχουμε μία δέσμη της εισόδου (minibatch), $X_t \in \mathbb{R}^{n\times d}$ (με n τα παραδείγματα ανά είσοδο και d τον αριθμό εισόδων) τη χρονική στιγμή t. Για την ίδια χρονική στιγμή, σημειώνουμε την κρυφή κατάσταση του l-ϊοστού (l^{th}, $l = 1, \ldots, L$) κρυμμένου επιπέδου ως $H_t^{(l)} \in \mathbb{R}^{n\times h}$ (h ο αριθμός κρυφών νευρώνων) και την έξοδο ως $O_t \in \mathbb{R}^{n\times q}$ (q το πλήθος εξόδων). Επίσης, θεωρώντας ότι $H_t^{(0)} = X_t$, δηλαδή ότι το κρυφό επίπεδο του 'μηδενικού' επιπέδου είναι η είσοδος, παίρνουμε ότι η κρυφή κατάσταση του l_{th} κρυφού επιπέδου που έχει ως συνάρτηση ενεργοποίησης την ϕ_l θα δίνεται από την εξίσωση:

$$H_t^{(l)} = \phi_l(H_t^{(l-1)}W_{xh}^{(l)} + H_{t-1}^{(l)}W_{hh}^{(l)} + b_h^{(l)}) \tag{3.3.11}$$

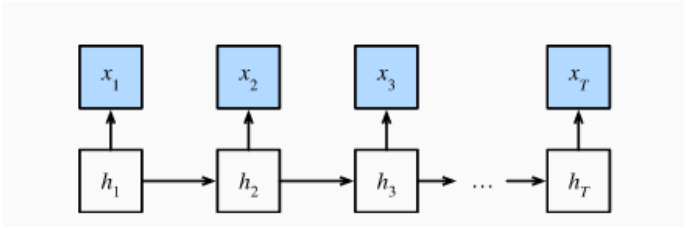
όπου τα βάρη $W_{xh}^{(l)} \in \mathbb{R}^{h\times h}$ και $W_{hh}^{(l)} \in \mathbb{R}^{h\times h}$ μαζί με τον όρο bias $b_h^{(l)} \in \mathbb{R}^{1\times h}$ είναι παράμετροι του μοντέλου στο l-ϊοστό κρυφό επίπεδο. Μία σημαντική παρατήρηση είναι ότι τελικά η έξοδος θα εξαρτάται μόνον από την κρυφή κατάσταση του τελευταίου (L_{th}) hidden layer ως:

$$O_t = H_t^{(L)}W_{hq} + b_q \tag{3.3.12}$$

όπου τα βάρη $W_{hq} \in \mathbb{R}^{h\times q}$ και ο όρος bias $b_q \in \mathbb{R}^{1\times q}$ είναι παράμετροι του μοντέλου στο στάδιο εξόδου. Όπως προαναφέραμε, μπορούμε εύκολα να πάρουμε ένα βαθύ RNN με πύλες αντικαθιστώντας τον υπολογισμό της κρυφής κατάστασης στην (3.3.11) με αυτόν, των GRU ή LSTM από τις εξίσωσεις (3.3.4) ή (3.3.10).

3.3.4 Αμφίδρομα Επαναληπτικά Νευρωνικά Δίκτυα (Bidirectional RNNs)

Προκειμένου να αποκτήσουμε κίνητρο για να χρησιμοποιήσουμε επιπλέον εναλλακτικές στην επίλυση προβλημάτων μηχανικής μάθησης θα ήταν χρήσιμο να κάνουμε μία πολύ σύντομη αναφορά στην έννοια του δυναμικού προγραμματισμού στις αλυσίδες Markov [50]. Έστω, ότι έχουμε ένα σύνολο T (κρυφών) καταστάσεων για κάθε μία από τις οποίες έχουμε και μία αντίστοιχη παρατήρηση (observation). Το μοντέλο *Κρυφής αλύσιδας Markov* (Hidden Markov model - HMM) υποθέτει ότι οι παρατηρήσεις μας καθορίζονται από αυτές τις κρυφές κατάστασεις. Στόχος είναι μέσα από ένα πιθανότικό μοντέλο να εξάγουμε πληροφορία για την παρατήρηση, έχοντας ως δεδομένο τη γνώση για την κρυφή κατάσταση. Θεωρούμε λοιπόν, για κάθε χρονική στιγμή $t \in \{1, T\}$ ότι η κρυφή κατάσταση είναι h_t καθορίζει την αντίστοιχη παρατήρησή μας, x_t μέσα από την πιθανότητα $P(x_t|h_t)$. Επιπρόσθετα, κάθε μετάβαση $h_t \to h_{t+1}$ δίνεται από κάποια πιθανότητα μετάβασης κατάστασης $P(h_{t+1}|h_t)$. Το πιθανότικό αυτό πρόβλημα μπορεί να αναπαρασταθεί γραφικά όπως στην εικόνα 3.9, στην οποία απεικονίζουμε ένα απλό μοντέλο Markov.



Εικόνα 3.9: Ένα κρυφό μοντέλο Markov [51]

Έτσι, για την ακολουθία των T παρατηρήσεων έχουμε την ακόλουθη από κοινού κατανομή πιθανότητας στις παρατηρούμενες και κρυφές καταστάσεις:

$$P(x_1, \ldots, x_T, h_1, \ldots, h_T) = \prod_{t=1}^{T} P(h_t \mid h_{t-1}) P(x_t \mid h_t) \tag{3.3.13}$$

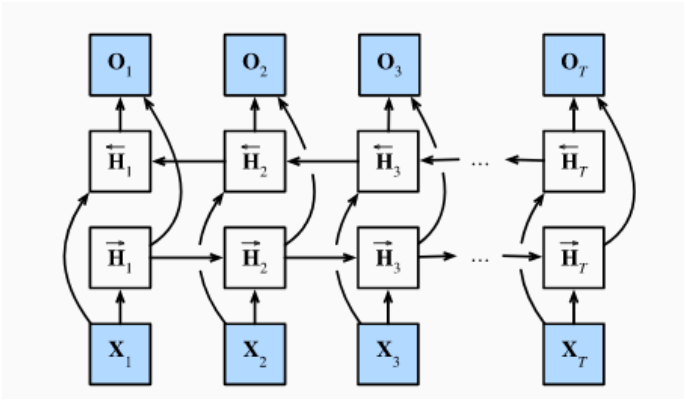
όπου $P(h_1 \mid h_0) = P(h_1)$. Χωρίς να επεκταθούμε περαιτέρω, θα αναφέρουμε το σκοπό αυτού του προβλήματος και την αρωγή του *Δυναμικού Προγραμματισμού* σε αυτό. Υποθέτοντας ότι παρατηρούμε όλες τις κατάστασεις x_i με εξαίρεση μίας κατάστασης x_j, προσπαθούμε να υπολογίσουμε την $P(x_j|x_{-j})$ όπου $x_{-j} = (x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_T)$. Ο υπολογισμός αυτής της πιθανότητας μπορεί να καταστεί πολύ περίπλοκος και χρονοβόρος δεδομένου ότι θα πρέπει να καλύψουμε όλους τους δυνατούς συνδυασμούς για τις κρυφές καταστάσεις $h_1, \ldots, h_T$. Ευτυχώς, με τη βοήθεια του δυναμικού προγραμματισμού καταλήγουμε στην παρακάτω εξίσωση που μας δίνει την πιθανότητα η αλυσίδα Markov να βρεθεί στην κατάσταση h_{t+1} τη στιγμή $t + 1$.

$$\pi_{t+1}(h_{t+1}) = \sum_{h_t} \pi_t(h_t) P(x_t \mid h_t) P(h_{t+1} \mid h_t) \tag{3.3.14}$$

με την αρχικοποίηση $\pi_1(h_1) = P(h_1)$. Η εξίσωση (3.3.14) ονομάζεται *εμπρόσθια αναδρομή* (forward recursion) καθώς για να προχωρήσουμε μπροστά σε χρονικά βήματα, βασιζόμαστε στα δεδομένα του προηγούμενου βήματος χρόνου. Εντελώς ανάλογα ορίζεται η *οπίσθια αναδρομή* (backward recursion) ως:

$$\rho_{t-1}(h_{t-1}) = \sum_{h_t} P(h_t \mid h_{t-1}) P(x_t \mid h_t) \rho_t(h_t) \tag{3.3.15}$$

με την αρχικοποίηση $\rho_T(h_T) = 1$. Εν αντιθέσει με την (3.3.14), στην εξίσωση (3.3.15) ξεκινάμε από το τέλος της ακολουθίας δεδομένων και προσπαθούμε να εξάγουμε πληροφορία για τις πρωτύτερες χρονικές στιγμές. Είναι εύκολο να απλοποιήσουμε τις εξισώσεις (3.3.14), (3.3.15) ως $\pi_{t+1} = f(\pi_t, x_t)$ και $\rho_{t-1} = g(\rho_t, x_t)$ για κάποιες κατάλληλες συναρτήσεις f, g. Οι τελευταίες δύο (2) εξισώσεις θυμίζουν αρκετά την εξίσωση της κρυφής κατάστασης ενός RNN με τη διαφορά ότι η δεύτερη 'τρέχει προς τα πίσω'. Αυτός ακριβώς είναι ο μηχανισμός που αναζητούμε προκειμένου να ενισχύσουμε τα μονόδρομα επαναληπτικά νευρωνικά δίκτυα ώστε να έχουν το πλεονέκτημα της γνώσης του μέλλοντος της ακολουθίας. Η μετατροπή ενός μονόδρομου RNN σε αμφίδρομο γίνεται με την προσθήκη ενός ακόμα κρυφού επιπέδου που όμως διαχειρίζεται τα δεδομένα είσοδου με αντίθετη σειρά από ότι το πρώτο κρυφό επίπεδο. *Τα αμφίδρομα RNNs* [52] έχουν το πλεονέκτημα ότι γνωρίζουν την πληροφορία τόσο από το παρελθόν προς το μέλλον όσο και από το μέλλον προς το παρελθόν και κάτα συνέπεια σε αρκέτες περιπτώσεις να μπορούν να κάνουν πιο ευέλικτους και ακριβείς υπολογισμούς. Η εικόνα 3.10 απεικονίζει την αρχιτεκτονική ενός αμφίδρομου RNN με ένα κρυφό επίπεδο.



Εικόνα 3.10: Η αρχιτεκτονική ενός αμφίδρομου RNN [51]

Ας υποθέσουμε ότι έχουμε μία δέσμη της εισόδου, $X_t \in \mathbb{R}^{n \times d}$ (n: τα παραδείγματα ανά είσοδο, d: τον αριθμό εισόδων ανά παράδειγμα) και η συνάρτηση ενεργοποίησης του κρυφού επιπέδου είναι ϕ. Στην αμφίδρομη αρχιτεκτονική, υποθέτουμε επίσης ότι οι εμπρόσθιες και οπίσθιες κρυφές καταστάσεις για αυτό το χρονικό βήμα είναι αντίστοιχα οι $\overrightarrow{H}_t \in \mathbb{R}^{n \times h}$ και $\overleftarrow{H}_t \in \mathbb{R}^{n \times h}$ (h: ο αριθμός των κρυφών νευρώνων). Η ενημέρωση των κρυφών καταστάσεων προς τα εμπρός και προς τα πίσω γίνεται ως:

$$\overrightarrow{H}_t = \phi(X_t W_{xh}^{(f)} + \overrightarrow{H}_{t-1} W_{hh}^{(f)} + b_h^{(f)}) \tag{3.3.16}$$

$$\overleftarrow{H}_t = \phi(X_t W_{xh}^{(b)} + \overleftarrow{H}_{t+1} W_{hh}^{(b)} + b_h^{(b)}) \tag{3.3.17}$$

όπου τα βάρη $W_{xh}^{(f)} \in \mathbb{R}^{d\times h}, W_{hh}^{(f)} \in \mathbb{R}^{h\times h}, W_{xh}^{(b)} \in \mathbb{R}^{d\times h}$ και $W_{hh}^{(b)} \in \mathbb{R}^{h\times h}$ και οι παράμετροι bias $b_h^{(f)} \in \mathbb{R}^{1\times h}, b_h^{(b)} \in \mathbb{R}^{1\times h}$ είναι όλα παράμετροι του μοντέλου. Στη συνέχεια, συνενώνουμε (concatenate) τις εμπρόσθιες και οπίσθιες κρυφές καταστάσεις $\overrightarrow{H}_t$ και $\overleftarrow{H}_t$ για να πάρουμε τη συνολική κρυφή κατάσταση $H_t \in \mathbb{R}^{n\times 2h}$, η οποία θα τροφοδοτηθεί στο στάδιο εξόδου. Στα βαθιά αμφίδρομα RNN με πολλαπλά κρυφά επίπεδα, αυτή η πληροφορία μεταβιβάζεται ως είσοδος στο επόμενο αμφίδρομο κρυφό επίπεδο. Τέλος, στο επίπεδο εξόδου, υπολογίζεται η έξοδος $O_t \in \mathbb{R}^{n\times q}$ (q: αριθμός εξόδων) ως:

$$O_t = H_t W_{hq} + b_q \tag{3.3.18}$$

Εδώ, η μήτρα βαρών $W_{hq} \in \mathbb{R}^{2h\times q}$ και το διάνυσμα bias $b_q \in \mathbb{R}^{1\times q}$ είναι παράμετροι του σταδίου εξόδου. Αξίζει να τονιστεί, ότι τα δύο (2) κρυφά επίπεδα αντίθετης κατεύθυνσης μπορούν να έχουν διαφορετικό πλήθος κρυφών νευρώνων.

3.4 Υβριδικά Νευρωνικά Δίκτυα (Hybrid Neural Networks)

Όπως έχουμε αναφέρει, σκόπος αυτής της εργασίας είναι η ανάδειξη μοντέλων που μπορούν να επιλύσουν προβλήματα πρόβλεψης ακολουθιακών δεδομένων. Τα επαναληπτικά νευρωνικά δίκτυα αποτελούν τον ακρογωνιαίο λίθο στην προσπάθεια επίλυσης τέτοιων προβλήματων εξαιτίας της προσθήκης της διάστασης του χρόνου κατά την επίλυση του πρόβληματος, αλλά και της ικανότητάς τους να διατηρούν πληροφορία από το παρελθόν προκειμένου να την αξιοποιήσουν για καλύτερη απόδοση. Θα λέγαμε με μία φράση ότι τα RNNs είναι κατάλληλα στο να 'συγκρατούν' χρονική πληροφορία.

Ωστόσο θα θέλαμε να κάνουμε ένα βήμα πάραπανω σε αυτή μας την προσπάθεια εισάγοντας λιγότερο συνηθισμένες αρχιτεκτονικές νευρωνικών δικτύων στην αντιμετώπιση του προβλήματος. Τα μοντέλα αυτά τα ονομάζουμε ***Υβριδικά Νευρωνικά Δίκτυα (HNNs)*** [53] και απαρτίζονται από τα γνωστά μας επαναληπτικά δίκτυα όσο και από μία άλλη ευρέως διαδεδομένη κατηγορία νευρωνικών δικτύων, τα ***Συνελικτικά Νευρωνικά Δίκτυα CNNs*** [54].

Σε αυτήν την εργασία όμως, σκόπος δεν είναι η ενδελεχής διευρεύνηση των CNNs. Για αυτό θα αρκεστούμε σε μία σύντομη παρουσίαση και θα εξηγήσουμε για το πως μπορούμε να τα συνδυάσουμε με τα RNNs, προκειμένου να λάβουμε αρχιτεκτονικές που θα μπορούν να αντιμετωπίσουν με ικανοποιητικό τρόπο το πρόβλημά μας.

3.4.1 Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks-CNNs)

Τα συνελικτικά δίκτυα, σε αντίθεση με τα επαναληπτικά, είναι ιδιαίτερα δυνατά στο να συγκρατούν χωρική πληροφορία, εξάγοντας τα σημαντικά χαρακτηριστικά από το σύνολο των δεδομένων που αναλύουν. Αυτά τα δίκτυα έχουν αποδειχθεί χρήσιμα και κατάλληλα στην ταξινόμηση και ανάλυση εικόνων, στην αναγνώριση προτύπων και βίντεο, στην

κατασκευή συστήματος συστάσεων και σε πολλούς άλλους τομείς [55] [56]. Η συνέλιξη στα μαθηματικά μεταξύ 2 συναρτήσεων $f, g : \mathbb{R}^d \to \mathbb{R}$ ορίζεται ως:

$$(f * g)(x) = \int f(z)g(x - z)dz \tag{3.4.1}$$

ενώ για το διακριτό χρόνο αντίστοιχα, αλλάζουμε το ολοκλήρωμα με άθροιση και έτσι έχουμε:

$$(f * g)(i) = \sum_a f(a)g(i - a) \tag{3.4.2}$$

Τέλος, αν το πρόβλημά μας είναι δισδιάστατο, έχουμε ένα άθροισμα με δείκτες (a, b) για f και αντίστοιχο με δείκτες $(i - a, j - b)$ για g. Η συνέλιξη ορίζεται ως:

$$(f * g)(i, j) = \sum_a \sum_b f(a, b)g(i - a, j - b) \tag{3.4.3}$$

Ας θεωρήσουμε λοιπόν σε αυτό το σημείο, ότι έχουμε δισδιάστατα δεδομένα (εικόνες) για τις οποίες συμβολίζουμε, το κάθε pixel στη θέση (i, j), με $[X]_{i,j}$ για την εικόνα εισόδου και με $[H]_{i,j}$ και την αντίστοιχη αναπαράστασή της στο κρυφό επίπεδο. Αναπαράγοντας τον παραπάνω τύπο (3.4.3) μπορούμε να δούμε πως μεταβάλλονται τα δεδομένα μας στο κρυφό επίπεδο ύστερα από το συνελικτικό στρώμα. Έτσι, στη θέση της συνάρτησης g έχουμε το pixel εισόδου και στη θέση της f τα αντίστοιχα βάρη κάθε pixel, που όμως θεωρούμε, ότι είναι ανεξάρτητα από τη θέση (i, j) του pixel, δηλαδή $[W]_{i,j,a,b} = [W]_{a,b}$. Συνεπώς, παίρνουμε:

$$[H]_{i,j} = u + \sum_a \sum_b [W]_{a,b}[X]_{i+a,j+b} \tag{3.4.4}$$

όπου u μια σταθερή (bias) τιμή. Με αυτόν τον τρόπο σταθμίζουμε αποτελεσματικά τα pixel $(i + a, j + b)$ στην περιοχή του (i, j) πολλαπλασιάζοντας τα με τους συντελεστές $[W]_{a,b}$ προκειμένου να λάβουμε την έξοδο $[H]_{i,j}$. Παρακινούμενοι ωστόσο, από το γεγονός ότι αυτού τους είδους δίκτυα πρέπει να διαθέτουν *τοπικότητα*, θεωρούμε ότι, δεν θα πρέπει να φύγουμε μακριά από την περιοχή (i, j), προκειμένου να συλλέξουμε τις σχετικές πληροφορίες που θα τις τροφοδότησουμε στο κρύφο επίπεδο $[H]_{i,j}$. Αυτό σημαίνει ότι εκτός κάποιου εύρους, $|a| > \Delta$ ή $|b| > \Delta$, θέτουμε $[V]_{a,b} = 0$. Ισοδύναμα, μπορούμε να ξαναγράψουμε την εξίσωση (3.4.4) ως:

$$[H]_{i,j} = u + \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} [W]_{a,b}[X]_{i+a,j+b} \tag{3.4.5}$$

Η εξίσωση (3.4.5) αποτελεί ένα *συνελικτικό στρώμα (convolutional layer)*. Τα δίκτυα που χρησιμοποιούν τέτοια στρώματα ονομάζονται *συνελικτικά νευρωνικά δίκτυα*. Στους όρους της μηχανικής μάθησης, το W αναφέρεται ως *πυρήνας* συνέλιξης ή *φίλτρο* ή απλά βάρη του επιπέδου και είναι συχνά παράμετροι εκπαίδευσης. Όταν η τοπική περιοχή είναι μικρή, η διαφορά σε σύγκριση με ένα πλήρως συνδεδεμένο δίκτυο μπορεί να είναι

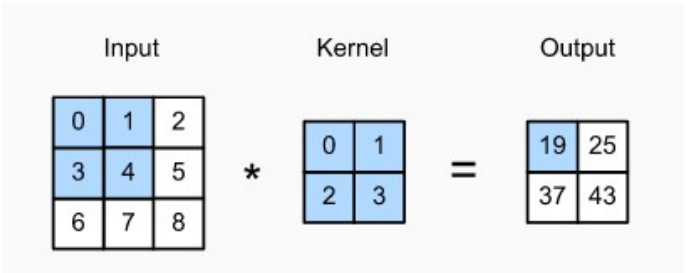
δραματική. Ενώ ένα πλήρως διασυνδεδεμένο δίκτυο θα χρειαζόταν εκατομμύρια παραμέτρους για να επεξεργαστεί ένα μόνο επίπεδο, σε ένα δίκτυο επεξεργασίας εικόνων χρειαζόμαστε τυπικά μερικές εκατοντάδες, χωρίς να αλλάξουμε τη διαστατικότητα των εισόδων ή των κρυφών αναπαραστάσεων. Το αντίτιμο που καταβάλλεται για αυτήν τη δραστική μείωση των παραμέτρων είναι, ότι πλέον τα χαρακτηριστικά είναι αμετάβλητα στη μετακίνηση της εικόνας (translation invariant) [59] και ότι το συνελικτικό επίπεδο μπορεί να ενσωματώσει μόνο τοπικές πληροφορίες όταν καθορίζει την έξοδο $[H]_{i,j}$.

Σε πολλές περιπτώσεις εφαρμογών τα δεδομένα εισόδου (εικόνες) αποτελούνται, πέρα από το ύψος και το πλάτος, από τα λεγόμενα *κανάλια (channels)*, π.χ. εικόνες RGB διάστασης $1024 \times 1024 \times 3$. Τα κανάλια μπορούν να θεωρηθούν ως εκχώρηση τρίτης διάστασης αναπαράστασης σε κάθε θέση pixel. Συνεπώς, θα πρέπει να επεκτείνουμε την είσοδο X ως $[X]_{i,j,k}$ και το φίλτρο της συνέλιξης W ως $[W]_{a,b,c}$. Ωστόσο, επειδή η είσοδός μας αποτελείται από έναν τανυστή τρίτης τάξης, θα ήταν καλή ιδέα να ορίσουμε ομοίως και την κρυφή αναπαράσταση H ως τανυστή ίδιας τάξης. Συνεπώς, θα πρέπει να είμαστε σε θέση να υποστηρίξουμε πολλαπλά κανάλια τόσο για την είσοδο X όσο και για την κρυφή κατάσταση H. Για να το κανούμε αυτό, θα χρειαστεί να επεκτείνουμε επιπλέον τα βάρη (φίλτρο) W ως $[W]_{a,b,c,d}$ και την εξίσωση (3.4.5) ως:

$$[H]_{i,j,d} = \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} \sum_{c} [W]_{a,b,c,d} [X]_{i+a,j+b,c} \tag{3.4.6}$$

όπου ο δείκτης d υποδεικνύει τα κανάλια εξόδου της κρυφής κατάστασης H [57].

Ας δούμε στο σημείο αυτό, πως πραγματοποιείται η συνέλιξη για δισδιάστατα δεδομένα (αγνοώντας τα κανάλια) και πως αποτυπώνονται αυτά στην κρυφή κατάσταση. Στην εικόνα 3.11 παρουσιάζουμε μία εικόνα εισόδου που είναι ένας δισδιάστατος τανυστής 3×3, καθώς και τον πυρήνα της συνέλιξης (ή παράθυρο συνέλιξης) που είναι επίσης δισδιάστατος τανυστής 2×2.



Εικόνα 3.11: Δισδιάστατη αναπαράσταση της πράξης cross-correlation. Τα σκιασμένα τμήματα είναι το πρώτο στοιχείο εξόδου καθώς και τα στοιχεία του τανυστή εισόδου και του πυρήνα που χρησιμοποιούνται για τον υπολογισμό: $0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3 = 19$ [58]

Η αναπαράσταση της πράξης της αλληλοσυσχέτισης (cross-correlation) σε 2 διαστάσεις, γίνεται 'σέρνοντας' το παράθυρο συνέλιξης που βρίσκεται στην επάνω αριστερή γωνία του τανυστή εισόδου τόσο από αριστερά προς τα δεξιά όσο και από πάνω προς τα κάτω. Έτσι, για την εικόνα 3.11 έχουμε τον ακόλουθο αναλυτικό υπολογισμό:

$$0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3 = 19,$$

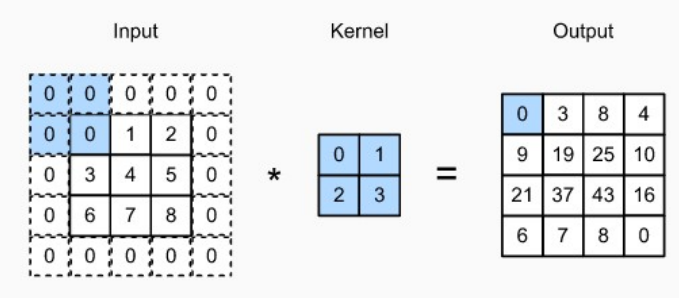
$$1 \times 0 + 2 \times 1 + 4 \times 2 + 5 \times 3 = 25,$$

$$3 \times 0 + 4 \times 1 + 6 \times 2 + 7 \times 3 = 37,$$

$$4 \times 0 + 5 \times 1 + 7 \times 2 + 8 \times 3 = 43.$$

Παρατηρούμε επίσης, ότι η έξοδος είναι ένας τανυστής 2×2. Επειδή ο πυρήνας έχει πλάτος και ύψος μεγαλύτερο από ένα, μπορούμε να υπολογίσουμε σωστά την πράξη του cross-correlation μόνο όταν ο πυρήνας ταιριάζει πλήρως στην εικόνα (δηλαδή βρίσκεται εξ' όλοκλήρου εντός της εικόνας). Γενικότερα, το μέγεθος εξόδου δίνεται από το μέγεθος εισόδου $n_h \times n_w$ μείον το μέγεθος του πυρήνα της συνέλιξης $k_h \times k_w$ μέσω της σχέσης $(n_h - k_h + 1) \times (n_w - k_w + 1)$.

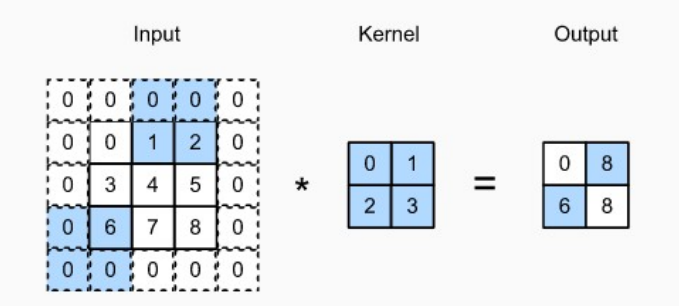
Όπως φαίνεται στην εικόνα 3.11, ο τανυστής της εξόδου μπορεί να έχει αρκετά μειωμένη διαστάση σε σχέση με αυτή της εισόδου, πράγμα που εξαρτάται από το μέγεθος του παραθύρου της συνέλιξης. Αυτό μπορεί να μας οδηγήσει στο να χάνουμε σημαντικά pixel στην περίμετρο της εικόνας μας. Μία απλή λύση σε αυτό το πρόβλημα είναι να προσθέσουμε επιπλέον pixel περιμετρικά της εικόνας εισόδου, αυξάνοντας έτσι το πραγματικό μέγεθος της εικόνας. Τη μέθοδο αυτή την ονομάζουμε *γέμισμα (padding)* και συνήθως χρησιμοποιούμε μηδενικά για να επαυξήσουμε την είκονα. Έτσι, εάν προσθέσουμε ένα σύνολο p_h γραμμών γεμίσματος (συνήθως οι μισές πάνε στην πάνω κορυφή και οι άλλες μισές στο κάτω μέρος) και ένα σύνολο p_w στηλών γεμίσματος (επίσης οι μισές στα αριστερά και οι υπόλοιπες στα δεξιά), τότε η διάσταση εξόδου θα είναι: $(n_h - k_h + p_h + 1) \times (n_w - k_w + p_w + 1)$. Σε πολλές περιπτώσεις, ορίζουμε $p_h = k_h - 1$ και $p_w = k_w - 1$ για να δώσουμε στην είσοδο και στην έξοδο το ίδιο ύψος και πλάτος. Στην εικόνα 3.12 βλέπουμε ένα τυπικό γέμισμα μίας εικόνας εισόδου για την οποία επαυξάνουμε τη διάσταση από 3×3 σε 5×5, ώστε να λάβουμε στην εξόδο μία εικόνα διάστασης 4×4. Στο επισκιασμένο μέρος της εικόνας κάνουμε τον εξής υπολογισμό $0 \times 0 + 0 \times 1 + 0 \times 2 + 0 \times 3 = 0$.



Εικόνα 3.12: Δισδιάστατη αναπαράσταση της πράξης cross-correlation εφοδιασμένη με γέμισμα (padding) [60]

Αναφέραμε νωρίτερα ότι 'σέρνουμε' το παράθυρο συνέλιξης κατά μήκος και κατά πλάτος της εικόνας εισόδου. Ο ρυθμός με τον οποίο διασχίζουμε τις γραμμές και τις στήλες ονομάζεται *βήμα (stride)*. Μέχρι στιγμής, έχουμε χρησιμοποιήσει βήμα ίσο με 1, τόσο για ύψος όσο και για πλάτος. Ωστόσο, μερικές φορές, είτε για υπολογιστική απόδοση είτε επειδή θέλουμε να μειώσουμε αρκετά τη διαστατικότητα των δεδομένων

μας, μετακινούμε το παράθυρο, περισσότερα από ένα στοιχεία κάθε φορά, παραλείποντας τις ενδιάμεσες περιοχές της εικόνας. Η εικόνα 3.13 δείχνει τη συνέλιξη μεταξύ μίας εικόνας είσοδου και ενός πυρήνα με βήμα 3 κάθετα και 2 οριζόντια. Τα σκιασμένα τμήματα αντιστοιχούν στις ακόλουθες 2 πράξεις μεταξύ εισόδου και πυρήνα: $0\times0+0\times1+1\times2+2\times3=8$, $0\times0+6\times1+0\times2+0\times3=6$. Βλέπουμε ότι η ανωτέρα σκίαση απέχει 3 γραμμές από την κατωτέρω σκίαση.



Εικόνα 3.13: Αναπαράσταση της πράξης cross-correlation με βήμα (stride) 3 και 2 για ύψος και πλάτος, αντίστοιχα [60]

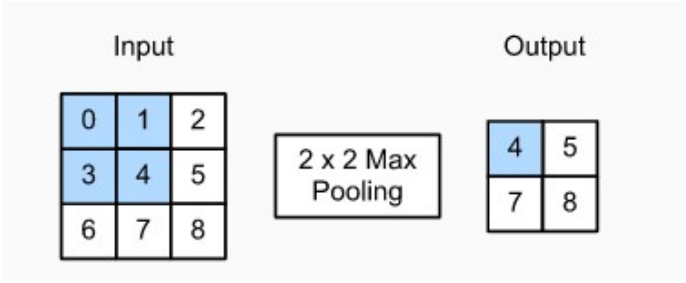
Γενικά, όταν το ύψος για το βήμα είναι s_h και το αντίστοιχο πλάτος είναι s_w,η διάσταση της εξόδου θα είναι:

$$\lfloor \frac{n_h - k_h + p_h + s_h}{s_h} \rfloor \times \lfloor \frac{n_w - k_w + p_w + s_w}{s_w} \rfloor \tag{3.4.7}$$

Τα σύμβολα $\lfloor \quad \rfloor$ δηλώνουν τη στρογγυλοποίηση προς τα κάτω. Εάν ορίσουμε και πάλι $p_h = k_h - 1$ και $p_w = k_w - 1$, τότε το σχήμα εξόδου θα απλοποιηθεί σε $\lfloor \frac{n_h+s_h-1}{s_h} \rfloor \times \lfloor \frac{n_w+s_w-1}{s_w} \rfloor$. Η σχέση (3.4.7) δίνει στη γενική μορφή τη διάσταση της εικόνας που λαμβάνουμε στην έξοδο ενός συνελικτικού στρώματος. Οι παραπάνω διαδικασίες του cross-correlation, padding και stride είναι ανάλογες και για δεδομένα με πολλαπλό αριθμό καναλιών και για αυτό δεν θα τις αναλύσουμε εδώ. Σχετικές πληροφορίες μπορούν να βρεθούν εδώ [58],[60].

Στην πλειονότητα των περιπτώσεων τα συνελικτικά δίκτυα απαρτίζονται από παραπάνω από ένα συνελικτικά στρώματα με συνέπεια, να υπάρχει μεγάλη πιθανότητα υπερπροσαρμογής του δικτύου σε ορισμένα τοπικά χαρακτηριστικά των δεδομένων. Για αυτό το λόγο είναι αναγκαία η υπάρξη ενός μηχανισμού που θα μετριάζει την ευαισθησία των συνελικτικών επιπέδων στην τοπικότητα και στην υπερβολική χωρική μείωση της διαστατικότητας των δεδομένων. Αυτόν το μηχανισμό υλοποιούν τα λεγόμενα *στρώματα συγκέντρωσης (pooling layers)*. Όπως και τα συνελικτικά στρώματα, τα στρώματα συγκέντρωσης αποτελούνται από ένα παράθυρο σταθερού σχήματος που ολισθαίνει σε όλες τις περιοχές της εισόδου σύμφωνα με κάποιο βήμα, υπολογίζοντας μία μόνο έξοδο για κάθε θέση που διασχίζεται από το παράθυρο. Σε ορισμένες περιπτώσεις, το παράθυρο αυτό ονομάζεται *παράθυρο συγκέντρωσης (pooling window)*. Ωστόσο, σε αντίθεση με την πράξη του cross-correlation μεταξύ της εισόδου και του πυρήνα στο συνελικτικό στρώμα, το στρώμα συγκέντρωσης δεν περιέχει παραμέτρους. Υπάρχουν 2 λειτουργίες

pooling, η *συγκέντρωση μεγιστοποίησης (maximum pooling)* και *συγκέντρωση μέσης τιμής (average pooling)*. Στην εικόνα 3.14 παρουσιάζεται η διαδικασία του maximum pooling με ένα παράθυρο συγκέντρωσης 2×2.

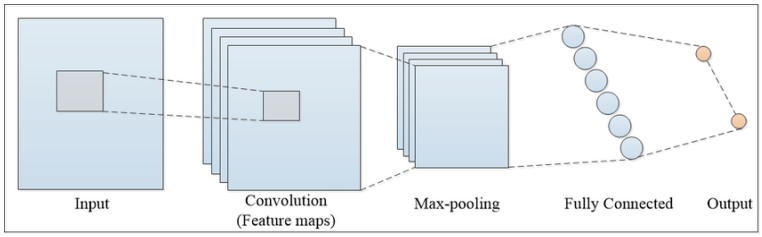


Εικόνα 3.14: Αναπαράσταση Max pooling με παράθυρο συγκέντρωσης 2×2. Τα σκιασμένα τμήματα είναι το πρώτο στοιχείο εξόδου αλλά και τα στοιχεία του τανυστή εισόδου που εμπεριέχονται στον υπολογισμό του παραθύρου συγκέντρωσης: $\max(0, 1, 3, 4) = 4$ [61]

Παρατηρούμε ότι η έξοδος έχει και αυτή διάσταση 2×2. Τα τέσσερα στοιχεία προέρχονται από την εξαγωγή της μέγιστης τιμής σε κάθε ολίσθηση του παραθύρου συγκέντρωσης. Αναλυτικά, οι υπολογισμοί γίνονται ως εξής:

$$\max(0, 1, 3, 4) = 4, \max(1, 2, 4, 5) = 5, \max(3, 4, 6, 7) = 7, \max(4, 5, 7, 8) = 8.$$

Ένα στρώμα συγκέντρωσης με παράθυρο διάστασης $p \times q$, ονομάζεται (p, q) συγκεντρωτικό στρώμα, ενώ η πράξη που επιτελεί ονομάζεται συγκέντρωση διάστασης (p, q). Τέλος, στην εικόνα 3.15 παρουσιάζουμε ένα συνελικτικό δίκτυο το οποίο ακολουθείται από ένα στρώμα μέγιστης συγκέντρωσης και από ένα πλήρως διασυνδεδεμένο στρώμα



Εικόνα 3.15: Αναπαράσταση ενός τυπικού συνελικτικού δικτύου με Max Pooling και ένα πλήρως διασυνδεδεμένο στρώμα [62]

3.4.2 Ζεύξη Επαναληπτικών και Συνελικτικών Νευρωνικών Δικτύων (CNN-RNNs)

Σε αυτή την παράγραφο θα συνδυάσουμε τα συνελικτικά δίκτυα με τα επαναληπτικά των ενοτήτων 3.2 και 3.3. Ο συνδυασμός των CNN με τα RNN ενδείκνυται για την αντιμετώπιση προβλημάτων πρόβλεψης ακολουθίας, όπου τα δεδομένα είναι στη μορφή

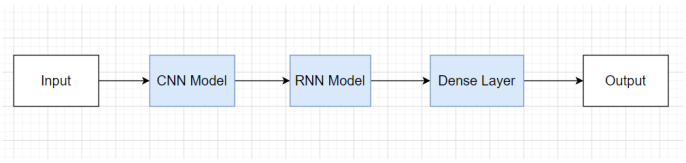
χωρικών εικόνων και βίντεο. Θα μπορούσαμε να πούμε, ότι συνδυάζοντας αυτές τις δύο αρχιτεκτονικές έχουμε τη δυνατότητα τόσο να εξάγουμε τα σημαντικότερα από τα χαρακτηριστικά των δεδομένων μας, όσο και να αξιοποιήσουμε αυτά σε μία χρονική σειρά διατηρώντας την όποια πληροφορία χρειαζόμαστε από το παρέλθον. Σε αυτή την εργασία παρουσιάζουμε δύο τρόπους με τους οποίους μπορούμε να ενώσουμε αυτές τις 2 κατηγορίες νευρωνικών δικτύων. Ο πρώτος τρόπος είναι η απλή τοποθέτηση για αρχή του συνελικτικού δικτύου (για το φιλτράρισμα των σημαντικότερων χαρακτηριστικών των δεδομένων είσοδου) και εν συνεχεία επαναληπτικού δικτύου που θα διαχειριστεί αυτά τα χαρακτηριστικά ως χρονική ακολουθία (και όχι το σύνολο των δεδομένων όπως ένα απλό RNN) [63]. Ο δεύτερος τρόπος είναι μία παραλλαγή των μοντέλων μακράς βραχυπρόθεσμης μνήμης (LSTM) τα οποία ονομάζουμε *συνελικτικά μοντέλα μακράς βραχυπρόθεσμης μνήμης* (Convolutional LSTM) [65].

CNN-RNN αρχιτεκτονική

Η αρχιτεκτονική αυτή είναι ιδιαιτέρως απλή, από την άποψη ότι απλώς τοποθετεί σε σειρά τα δύο γνωστά μας είδη δικτύων με τη μόνη παραπάνω προσοχή να έρχεται στο κομμάτι της εξόδου του $1^{ου}$ δικτύου που αποτελεί είσοδο για το $2^{ο}$. Τα CNN-RNNs αναπτύχθηκαν για προβλήματα πρόβλεψης χρονοσειρών και την εφαρμογή δημιουργίας περιγραφικών κειμένων από ακολουθίες εικόνων (π.χ. βίντεο) [64]. Κάποια κλασσικά είδη προβλημάτων είναι:

- Αναγνώριση Δραστηριότητας: Ανίχνευσης κάποιου συγκεκριμένου μοτίβου είκονας σε ένα σύνολο εικόνων ή βίντεο
- Περιγραφή Εικόνων/Βίντεο: Αναπαραγωγή περιγραφής κειμένου για ένα σύνολο εικόνων ή βίντεο
- Επεξεργασία φυσικής γλώσσας, Ανγνώριση ομιλίας ή πρόβλεψη χρονοσειρών [109] όπου το CNN χρησιμοποιείται ως εξολκέας χαρακτηριστικών και το RNN ως ο κύριος επεξεργαστής των δεδομένων.

Η τελευταία από τις 3 παραπάνω περιγραφές είναι και αυτή που μοιάζει πιο πολύ με την αρχιτεκτονική που εμείς θα χρησιμοποιήσουμε. Μια τέτοια δομή ενός CNN-RNN παρουσιάζεται στην εικόνα 3.16, στην οποία μπορούμε να δούμε ότι το συνολικό μας δίκτυο απαρτίζεται από 3 διακριτά μέρη, το CNN μοντέλο, RNN μοντέλο , και το πλήρως διασυνδεδεμένο (Dense) στρώμα εξόδου. Να τονιστεί ότι ως επαναληπτικό δίκτυο μπορεί να χρησιμοποιηθεί οποιαδήποτε από τις αρχιτεκτονικές που έχουμε γνωρίσει.



Εικόνα 3.16: Η αρχιτεκτονική ενός CNN-RNN δικτύου

Θα πρέπει να τονιστεί ότι από το 2019, αρχιτεκτονικές CNN και CNN-RNN έχουν εφαρμοστεί για την ανίχνευση COVID-19 μέσω ανάλυσης ιατρικών εικόνων και ειδικότερα ακτινογραφιών ή αξονικών τομογραφιών. Η προηγούμενη δραστηριότητα του Εργαστηρίου Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης, όπου εκπονήθηκε η παρούσα εργασία, περιλαμβάνει την ανάπτυξη και χρήση τεχνικών που αφορούν στην εφαρμογή νευρωνικών δικτύων για πρόβλεψη ασθενειών μέσω ανάλυσης ιατρικών εικόνων [128][129][130][131][132][133][134], στην εξαγωγή λανθανουσών μεταβλητών από βαθιά νευρωνικά δίκτυα για διαφανείς προβλεψεις [135][136][137], όπως και για επανεκπαίδευση και προσαρμογή βαθιών νευρωνικών δικτύων σε διαφορετικά σετ δεδομένων [138][139][140][141]. Ταυτόχρονα, έχει γίνει προσπάθεια σύνδεσης βαθιών νευρωνικών δικτύων και σημασιολογικών τεχνικών [142][143][144][145][146][147], χρήση τεχνικών Προσοχής (Attention) σε δίκτυα [148][149] (έναν μηχανισμό που θα γνωρίσουμε και αργότερα στην παρούσα εργασία) αλλά και προσπάθεια σύνθεσης νέων αναπαραστάσεων από ήδη υπάρχουσες [150][151].

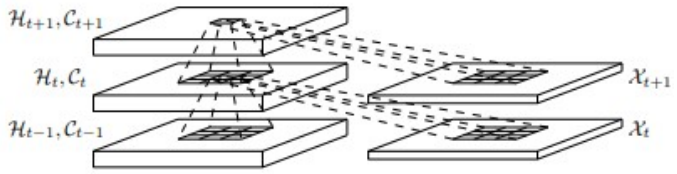
Αρχιτεκτονική Convolutional LSTM

Όπως έχει ήδη αναφερθεί η αρχιτεκτονική Convolutional LSTM (ή για συντομία ConvLstm) είναι μια παραλλαγή του απλού LSTM. Η βασική τους διαφορά είναι ότι αυτό το μοντέλο εκτελεί συνελίξεις μεταξύ των τανυστών εισόδου αντί για πολλαπλασιασμό (συνένωση). Αυτό σημαίνει, ότι τα δεδομένα που ρέουν κατά μήκος των κελιών ConvLstm διατηρούν το μέγεθος της εισόδου ($3D$ ή $4D$) και δεν είναι απλώς διανύσματα. Παρ' όλο την ικανότητα των απλών RNN, GRU και LSTM να συσχετίζουν καλά χρονικά ακολουθιακά δεδομένα, τα μοντέλα αυτά έχουν περιορισμένες δυνατότητες όταν η συζήτηση έρχεται στη διαχείριση χωροχρονικών δεδομένων (πχ. πρόβλεψη εικόνων ή frames σε βίντεο). Αυτό συμβαίνει εξαιτίας του γεγονότος ότι σε ένα LSTM κατά τις μεταβάσεις μεταξύ εισόδου-κελιού μνήμης και κελιού μνήμης-κρυφής κατάστασης δεν κωδικοποιούμε καθόλου χωρική πληροφορία. Για να ξεπεραστεί αυτό το πρόβλημα, στη δομή ConvLstm όλες οι είσοδοι $X_1, ..., X_t$, τα κελιά μνήμης $C_1, ..., C_t$, οι κρυφές καταστάσεις $H_1, ..., H_t$ καθώς και οι πύλες I_t, F_t και O_t αναπαρίστανται ως τρισδιάστατοι (ή και τετραδιάστατοι αναλόγως το πρόβλημα) τανυστές ($3D$ tensors), στους οποίους οι δύο τελευταίες διαστάσεις είναι χωρικές διαστάσεις (σειρές και στήλες) [65].
Προκειμένου να καταλάβουμε καλύτερα μπορούμε να φανταστούμε την είσοδο, το cell state και το hidden state ως διανύσματα που στέκονται σε ένα χωρικό πλέγμα. Το ConvLSTM καθορίζει τη μελλοντική κατάσταση ενός συγκεκριμένου κελιού στο πλέγμα από τις εισόδους της τρέχουσας χρονικής στιγμής και τις προηγούμενες (χρονικά) καταστάσεις των τοπικών γειτόνων του. Αυτό μπορεί εύκολα να επιτευχθεί με τη χρήση της συνέλιξης στις μεταβάσεις μεταξύ εισόδου-κελιού μνήμης και κελιού μνήμης-κρυφής κατάστασης. Στην εικόνα 3.17 παρουσιάζεται μια σχηματική αναπαράσταση αυτής της λειτουργίας.

Οι βασικές εξίσωσεις που διέπουν τη λειτουργία του ConvLSTM είναι οι ακόλουθες (εντελώς ανάλογες με τις (3.3.5) έως (3.3.10)):

$$I_t = \sigma(X_t * W_{xi} + H_{t-1} * W_{hi} + W_{ci} \odot C_{t-1} + b_i) \tag{3.4.8}$$

$$F_t = \sigma(X_t * W_{xf} + H_{t-1} * W_{hf} + W_{cf} \odot C_{t-1} + b_f) \tag{3.4.9}$$



Εικόνα 3.17: Αναπαράσταση των πράξεων μεταξύ των X_t, C_t, H_t στο εσωτερικό της δομής ConvLSTM [65]

$$O_t = \sigma(X_t * W_{xo} + H_{t-1} * W_{ho} + W_{co} \odot C_t + b_o) \tag{3.4.10}$$

$$\tilde{C}_t = tanh(X_t * W_{xc} + H_{t-1} * W_{hc} + b_c) \tag{3.4.11}$$

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t \tag{3.4.12}$$

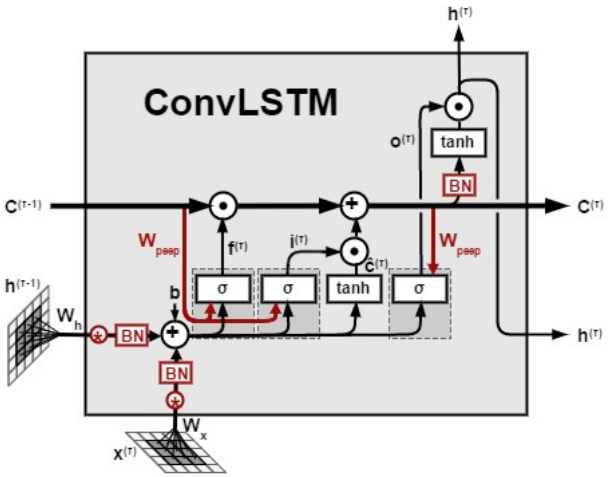
$$H_t = O_t \odot \tanh(C_t) \tag{3.4.13}$$

όπου και πάλι $\odot$ είναι ο τελεστής Hadamard και $*$ η πράξη της συνέλιξης.

Προσέξτε ότι πρώτες 3 εξισώσεις (3.4.8), (3.4.9) και (3.4.10) σε σχέση με τις (3.3.5), (3.3.6) και (3.3.7) έχουν έναν όρο γινομένου Hadamard παραπάνω ($W \odot C$). Αυτό υποδηλώνει ότι οι πύλες εισόδου, λήθης και εξόδου λαμβάνουν υπόψη τους και το κελί μνήμης είτε της προηγούμενης χρονικής στιγμής είτε αυτής (C_{t-1}, C_t) αφού αυτό πρώτα σταθμιστεί με τα βάρη $W_{ci}, W_{cf}, W_{co} \in \mathbb{R}^{h \times h}$. Στην εικόνα 3.18 φαίνεται η αρχιτεκτονική του κελιού ConvLSTM στην οποία έχει προστεθεί και η διαφοροποίηση που μόλις αναφέραμε (τονίζεται με κόκκινα βέλη). Τέλος, αξίζει να τονισθεί ότι αν υποθέσουμε ότι η είσοδος, το cell state και το hidden state σε ένα απλό LSTM είναι τρισδιάστατοι τανυστές με τις 2 τελευταίες διαστάσεις να είναι 1, τότε το LSTM είναι μια ειδική περίπτωση του ConvLSTM όπου όλα τα χαρακτηριστικά που στέκονται σε ένα μόνο κελί.

Temporal Convolutional Network (TCN)

Τα TCN είναι ένας ξεχωριστός τύπος δικτύου και ανήκει στην οικογένεια των συνελικτικών δικτύων. Εμφανίστηκε για πρώτη φορά στο [67] από τον Shaojie Bai το 2018. Προκείται για ένα μοντέλο που έχει τη δυνατότητα διαχείρισης ακολουθιακών δεδομένων στο οποίο η έξοδος έχει το ίδιος μήκος με την είσοδο (όπως και στα RNN) ενώ αποτρέπει τη διαρροή πληροφορίας από το παρέλθον στο μέλλον (casual convolutions). Το πλεονέκτημά τους σε σχέση με τα LSTM είναι ότι έχουν την ικανότητα να δέχονται μεγάλου μήκους ακολουθίες αξιοποιώντας καλύτερα παρελθοντικές πληροφορίες. Τα τρία βασικά χαρακτηριστικά τους είναι οι casual συνελίξεις, οι διεσταλμένες συνελίξεις και οι υπολειμματικές συνδέσεις.



Εικόνα 3.18: Η αρχιτεκτονική του κελιού ConvLSTM [66]

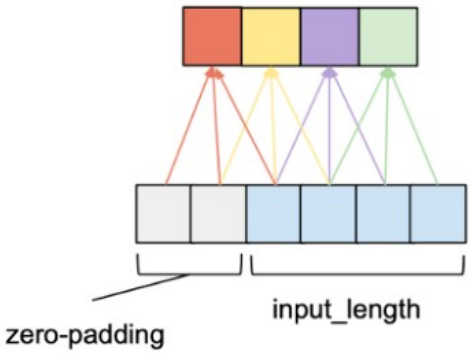
Casual Convolutions

Πριν ορίσουμε τη δομή του δικτύου, θα πρέπει πρώτα να περιγράψουμε την έννοια της ακολουθιακής μοντελοποίησης (sequence modeling). Έστω λοιπόν, μια ακολουθία εισόδου $x_0, ..., x_T$, και μια ακολουθία εξόδου $y_0, ..., y_T$ την οποία θέλουμε να προβλέψουμε. Η έννοια του casual συνελικτικού στρώματος μας υποδεικνύει ότι η πρόβλεψη μιας τιμής y_t στην έξοδο θα πρέπει να εξαρτάται μόνο από στοιχεία της εισόδου που έχουν προηγηθεί χρονικά, δηλαδή $x_0, ..., x_t$. Οπότε, ένα δίκτυο ακολουθιακής μοντελοποίησης είναι οποιαδήποτε συνάρτηση $f : \mathcal{X}^{T+1} \to \mathcal{Y}^{T+1}$ η οποία αντίστοιχίζει:

$$\hat{y_0}, ..., \hat{y_T} = f(x_0, ..., x_T) \tag{3.4.14}$$

αν η κάθε έξοδος y_T εξαρτάται μόνο από στοιχεία της εισόδου έως τη χρονική στιγμή t, δηλαδή τα $x_0, ..., x_t$ και όχι από κάποια είσοδο από τις $x_{t+1}, ..., x_T$. Ο στόχος εκμάθησης της ακολουθιακής μοντελοποίησης είναι να βρεθεί η κατάλληλη συνάρτηση (δίκτυο) f που ελαχιστοποιεί κάποιο λάθος μεταξύ των πραγματικών τιμών και των προβλέψεων, $L(\{y_0, ..., y_T\}, \{f(x_0, ..., x_T)\})$.

Ο παραπάνω περιορισμός επιτυγχάνεται μέσα από τις casual συνελίξεις οι οποίες σε κάθε χρονική στιγμή t, χρησιμοποιούν φίλτρα συνέλιξης έτσι ώστε κάθε εξόδος να εξαρτάται μόνο από στοιχεία της εισόδου νωρίτερα από αυτήν τη χρονική στιγμή. Στην εικόνα 3.19 φαίνεται γραφικά ένα παράδειγμα casual συνέλιξης. Από την εικόνα αυτή, μπορεί να εξαχθεί το συμπέρασμα ότι προκειμένου η έξοδος να έχει το ίδιο μήκος με την είσοδο, θα πρέπει η τελευταία να 'γεμιστεί' εξ αριστερών με μηδενικά (zero padding) πλήθους (kernel size-1).



Εικόνα 3.19: Αναπαράσταση των casual convolutions [68]

Διεσταλμένες Συνελίξεις (Dilated Convolutions)

Μέσω των casual συνελίξεων μπορούμε να κοιτάξουμε πίσω στο σύνολο της ιστορικής πληροφορίας αυξάνοντας το βάθος του δικτύου με γραμμικό τρόπο. Κάτι τέτοιο όμως προκαλεί πρόβλημα στη διαχείριση μεγάλου μήκους ακολουθιών καθώς απαιτούνται πολλαπλά στρώματα συνελίξεων επιβαρύνοντας σημαντικά το δίκτυο. Το πρόβλημα αυτό μπορεί να επιλυθεί με τη χρήση διεσταλμένων συνελίξεων που επιτρέπουν την αύξηση του *δεκτικού πεδίου* του δικτύου (receptive field) [67][69][70]. Ως δεκτικό πεδίο ορίζεται το σύνολο των στοιχείων της αρχικής εισόδου που επηρεάζουν ένα συγκεκριμένου στοιχείο εξόδου. Οπότε, για μια ακολουθία $\mathbf{x} \in \mathbb{R}^n$ και για ένα φίλτρο $f : \{0, ..., k-1\} \to \mathbb{R}$ η διεσταλμένη συνέλιξη F στο στοχείο s της ακολουθίας ορίζεται:

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-di} \tag{3.4.15}$$

όπου d ο συντελεστής διαστολής, k το μέγεθος του φίλτρου και το $s - di$ αντιπροσωπεύει την κατεύθυνση του παρελθόντος. Έτσι η διαστολή είναι ισοδύναμη με την εισαγωγή ενός σταθερού βήματος μεταξύ δύο διαδοχικών στοιχείων του πυρήνα. Για $d = 1$ η διεσταλμένη συνέλιξη γίνεται κανονική. Αυξάνοντας τον συντελεστή διαστολής, αυξάνεται ανάλογα και το δεκτικό πεδίο και έτσι μας επιτρέπεται μεγαλύτερη ιστορικότητα στα δεδομένα εισόδου χωρίς να μεγαλώνει εκθετικά η πολυπλοκότητα του δικτύου. Για n επιπέδα δικτύου το μέγεθος του δεκτικού πεδίου, r, δίνεται από την: $r = 1 + n \cdot (k-1) = 1 + d \cdot (k-1)$. Ωστόσο, αν το d παραμένει σταθερό ($\neq 1$) τότε θα χρειαζόμασταν ακόμα γραμμικό αριθμό επιπέδων, ώστε να καλύψουμε πλήρως το ιστορικό εισόδου. Για αυτό το λόγο, θα πρέπει ο συντελεστής διαστολής να αυξάνεται εκθετικά από επίπεδο σε επίπεδο ως $d = b^i$, όπου b η *βάση διαστολής* και i ο δείκτης του επιπέδου (πρώτο επίπεδο $\to i = 0$). Οπότε, κάθε επιπρόσθετο επίπεδο προσθέτει μια τιμή $d \cdot (k-1)$ στο τρέχον εύρος του δεκτικού πεδίου, και κατά συνέπεια το τελικό

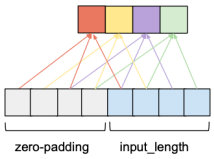
εύρος w του δεκτικού πεδίου υπολογίζεται ως:

$$w = 1 + \sum_{i=0}^{n-1}(k-1)b^i = 1 + (k-1)\cdot\frac{b^n-1}{b-1} \tag{3.4.16}$$

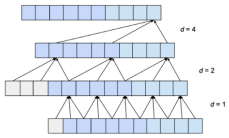
Τέλος, για να έχουμε πλήρη κάλυψη του ιστορικού εισόδου θα πρέπει $w \geq l$, όπου l το μήκος ακολουθίας εισόδου και συνεπώς:

$$1 + (k-1)\cdot\frac{b^n-1}{b-1} \geq l \tag{3.4.17}$$

Στις εικόνες 3.20α′, 3.20β′ φαίνεται μια συνέλιξη με συντελεστή διαστολής $d = 2$ και ένα βαθύ συνελικτικό δίκτυο τεσσάρων επιπέδων με βάση διαστολής $b = 2$ αντίστοιχα.



(α′) Διεσταλμένη συνέλιξη με d=2



(β′) Αναπαράσταση πολλών συνελικτικών επιπέδων με βάση διαστολής $b = 2$

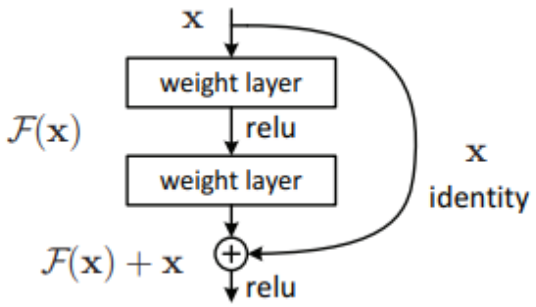
Εικόνα 3.20: Αναπαράσταση δικτύων που χρησιμοποιούν διεσταλμένες συνελίξεις [68]

Υπολειμματικές συνδέσεις (Residual Connections)

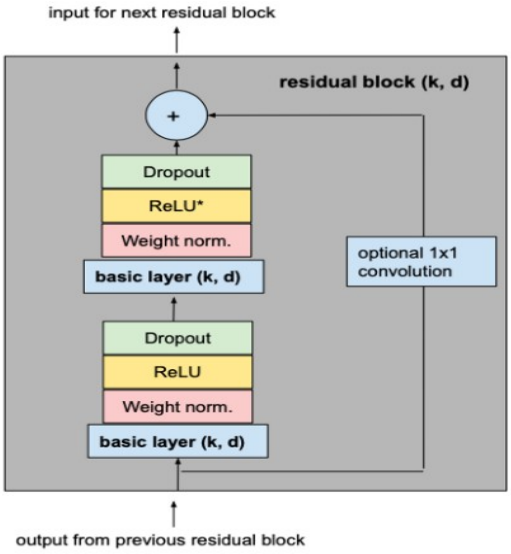
Συχνά τα βαθιά νευρωνικά δίκτυα παρουσιάζουν δυσκολίες κατά την εκπαίδευση τους λόγω της πολυπλοκότητας της αρχικτεκτονικής τους. Συγκεκριμένα, ενώ σε γενικές γραμμές, το μεγαλύτερο βάθος βοηθά, υπάρχει πάντα ένα σημείο κορεσμού όπου η ακρίβεια του δικτύου δεν βελτιώνεται περαιτέρω (πρόβλημα εκφυλισμού - degradation problem) [71]. Ακόμα, προσθέτοντας παραπάνω επίπεδα συχνά ερχόμαστε αντιμέτωποι με το πρόβλημα της υπερπροσαρμογής (overfitting), δηλαδή αύξηση του σφάλματος εκπαίδευσης. Ένας τρόπος αντιμετώπισης αυτού του προβλήματος είναι η χρήση των υπολειμματικών συνδέσεων που κάνει το TCN. Αυτού του είδους οι συνδέσεις επιτρέπουν την αύξηση του βάθους του δικτύου χωρίς να εγκυμονεί ο κίνδυνος του εκφυλισμού. Έτσι, αν η επιθυμητή έξοδος του δικτύου μας είναι $\mathcal{H}(x)$, τότε αφήνουμε το δίκτυο να τροφοδοτεί μια άλλη έξοδο $\mathcal{F}(x) = \mathcal{H}(x) - \mathbf{x}$ και συνεπώς η αρχική αναπαράσταση της εξόδου γίνεται $\mathcal{F}(x) + \mathbf{x}$, το οποίο αποδεικνύεται ότι είναι ευκολότερο να βελτιστοποιηθεί απ' ότι η $\mathcal{H}(x)$. Αυτό μπορούμε γραφικά να το παραστήσουμε πάνω στο δίκτυο με *συνδέσεις παράκαμψης* (shortcut connection). Αυτές οι συνδέσεις παρακάμπτουν κάποια στρώματα του δικτύου χωρίς να αυξάνουν την πολυπλοκότητα του μοντέλου. Ταυτόχρονα, θέτοντας $\mathcal{F}(x) = 0$ λαμβάνουμε την είσοδο στην έξοδο και κατά αυτόν τον τρόπο επιτρέπουμε την καλύτερη συσχέτιση της εξόδου $\mathcal{H}(x)$ με την είσοδο. Ένα παράδειγμα υπολειμματικής σύνδεσης φαίνεται στην εικόνα 3.21.

Επιπρόσθετα, στην εικόνα 3.22 φαίνεται η δομή ενός κελιού TCN. Μεταξύ άλλων βλέπουμε τη χρήση της ReLu ως συναρτήσης ενεργοποίησης που προσδίδει μη γραμμικότητα στο δίκτυο, επίπεδα Dropout που βοηθούν στην αποφυγή της υπερπροσαρμογής του

μοντέλου, επίπεδα κανονικοποίησης των βαρών και φυσικά τα μπλοκ συνέλιξης που καθορίζονται από το μέγεθος φίλτρου k και το συντελεστή διαστολής d. Βλέπουμε ακόμα, και ένα μπλοκ 1×1 συνέλιξης που διασφαλίζει ότι το μέγεθος του τανυστή εξόδου θα παραμείνει ίδιο με αυτό της εισόδου.



Εικόνα 3.21: Γραφική αναπαράσταση των residual connections σε ένα δίκτυο [71]



Εικόνα 3.22: Η εσωτερική δομή του κελιού TCN [68]

3.5 Βελτιώσεις των Επαναληπτικών Μοντέλων

Στην παράγραφο 3.2.1 αναφερθήκαμε στο παράδειγμα ενός γλωσσικού μοντέλου στο οποίο στόχος ήταν η πρόβλεψη εμφάνισης μιας ακολουθίας κειμένου x_t. Τα RNNs και αργότερα τα πιο περίπλοκα LSTMs και GRUs εφοδιασμένα με την κρυφή κατάσταση, H_t

και τους μηχανισμούς πυλών έδωσαν μια πρώτη λύση σε αυτό το πρόβλημα. Ωστόσο, εύκολα καταλαβαίνει κανείς πως το συγκεκριμένο ζήτημα είναι ιδαίτερα περίπλοκο καθώς μετά από μια πρόταση κειμένου οι πιθανές λέξεις που μπορούν να ακολουθήσουν είναι εκατοντάδες. Για παράδειγμα η πρόταση "βγήκα έξω για..." έχει πολλές πιθανές εκβάσεις όπως: "τρέξιμο", "βόλτα", "ψώνια" ,"δουλειές" κ.α. Τα επαναληπτικά νευρωνικά δίκτυα παρ' όλο τον μηχανισμό μνήμης που διαθέτουν δεν είναι ικανά σε όλες τις περιπτώσεις να κάνουν ακριβείς προβλέψεις πάνω σε τέτοιου είδους προβλήματα. Το συγκεκριμένο παράδειγμα προέρχεται από το πεδίο της *Επεξεργασίας Φυσικής Γλώσσας - (Natural Language Processing-NLP)*, ένα χώρο που τα τελευταία χρόνια η μηχανική μάθηση βρίσκει ολοένα και περισσότερο εφαρμογή [72] [73].
Ιδιαίτερα πρωτοποριακή μέθοδος σε αυτόν τον τομέα, που εκτόξευσε την επιτύχια των μοντέλων τόσο στην επεξεργασία γλώσσας όσο και στην μηχανική μετάφραση (machine translation), υπήρξε ο ***μηχανισμός της προσοχής (attention mechanism)***. Πρωτοεμφανίστηκε το 2015 στην δημοσίευση [74]. Ο μηχανισμός αυτός μπορεί να ερμηνευθεί ως ένα διάνυσμα από σημαντικά βάρη που έχουν εξαχθεί από την ήδη επεξεργασμένη πληροφορία. Το διάνυσμα αυτό μπορούμε να το χρησιμοποιήσουμε προκειμένου να προβλέψουμε ή να συμπεράνουμε τη συνέχεια μιας ακολουθίας και ονομάζεται *διάνυσμα προσοχής*. Στο παράδειγμα που αναφέραμε νωρίτερα, ένα νευρωνικό δίκτυο που χρησιμοποιεί τον μηχανισμό προσοχής θα δώσει ιδιαίτερη σημασία σε λέξεις που δηλώνουν ενδεχομένως κάποια δραστηριότητα και θα αποκλείσει άλλες όπως αυτές που ανήκουν στην κατηγορία φαγητών, χρωμάτων κ.α.
Παρ' όλο την θεαματική επιτυχία του μηχανισμού προσοχής στην επεξεργασία φυσικής γλώσσας η εφαρμογή του σε προβλήματα χρονοσειρών, όπως της παρούσας διπλωματικής, δεν είναι ακόμα τόσο διαδεδομένη. Ωστόσο, αποτελεί ένα πολλά υποσχόμενο βήμα και έτσι στα πλαίσια εμβάθυνσης της εργασίας εφαρμόζουμε τη συγκεκριμένη τεχνική (Κεφάλαια 4, 5) αφού πρώτα την αναλύσουμε θεωρητικά. Πρωτού γίνει αυτό θα πρέπει να γνωρίσουμε το μοντέλο του ***κωδικοποιητή-αποκωδικοποιητή (encoder-decoder)*** που αποτέλεσε την βάση για τον μηχανισμό της προσοχής.

3.5.1 Η δομή Κωδικοποιητών-Αποκωδικοποιητών (Encoders - Decoders)

Σε πολλές περιπτώσεις σε προβλήματα μηχανικής μάθησης θέλουμε να προβλέψουμε μια ακολουθία εξόδου δεδομένης μιας ακολουθίας εισόδου διαφορετικού μήκους, χωρίς αντιστοιχία μεταξύ κάθε εισόδου και κάθε εξόδου. Αυτή η τεχνική ονομάζεται *χαρτογράφηση ακολουθίας προς ακολουθίας* (sequence to sequence mapping) [75] και βρίσκεται πίσω από πολλές εφαρμογές όπως η επεξεργασία γλώσσας και η μηχανική μετάφραση. Η αρχιτεκτονική του κωδικοποιητή-αποκωδικοποιητή ήταν η πρώτη που μπόρεσε να αντιμετωπίσει το πρόβλημα της συσχέτισης ακολουθιών διαφορετικού μήκους. Παρουσιάστηκε για πρώτη φορά το 2014 στο [76]. Ένας κωδικοποιητής-αποκωδικοποιητής παίρνει μια ακολουθία ως είσοδο και δημιουργεί την πιο πιθανή επόμενη ακολουθία ως έξοδο. Όπως υποδηλώνει το όνομα, το μοντέλο αυτό αποτελείται από δύο υπο-μοντέλα:

- Κωδικοποιητής (Encoder): Είναι υπεύθυνος για τη διέλευση και την κωδικοποίηση μιας ακολουθίας εισόδου μέσα από διαδοχικά βήματα χρόνου σε ένα διάνυσμα σταθερού μήκους που ονομάζεται διάνυσμα περιβάλλοντος (context vector).

- Αποκωδικοποιητής (Decoder): Είναι υπεύθυνος για την πάροδο της πληροφορίας μέσα από διαδοχικά χρονικά βήματα εξόδου κατά την ανάγνωση του διανύσματος περιβάλλοντος.

Κωδικοποιητής

Ο κωδικοποιητής αποτελείται από μια ή παραπάνω επαναλαμβανόμενες μονάδες, που μπορεί να είναι απλά RNNs, κελιά LSTMs ή κελιά GRUs. Ας θεωρήσουμε τώρα ένα παράδειγμα ακολουθίας με μέγεθος δέσμης ίσο με 1. Ας υποθέσουμε ότι η ακολουθία εισόδου είναι $x_1, \ldots, x_T$ έτσι ώστε το x_t να δηλώνει την t^{th} ακολουθία εισόδου. Στο χρονικό βήμα t , το εκάστοτε RNN δίκτυο μετατρέπει το διάνυσμα $\mathbf{x_t}$ της αντίστοιχης ακολουθίας εισόδου x_t και την κρυφή κατάσταση $\mathbf{h_{t-1}}$ από το προηγούμενο βήμα, στην τρέχουσα κρυφή κατάσταση $\mathbf{h_t}$. Μπορούμε να χρησιμοποιήσουμε μια μη γραμμική συνάρτηση f για να εκφράσουμε το μετασχηματισμό του επαναληπτικού στρώματος του RNN ως:

$$\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{h}_{t-1}) \tag{3.5.1}$$

Γενικά, ο κωδικοποιητής μετατρέπει τις κρυφές καταστάσεις ανά πάσα χρονικό βήμα στη μεταβλητή περιβάλλοντος ($\mathbf{c}$) μέσω μιας προσαρμοσμένης μη γραμμικής συνάρτησης q:

$$\mathbf{c} = q(\mathbf{h}_1, \ldots, \mathbf{h}_T) \tag{3.5.2}$$

Λόγου χάρη, αν επιλέξουμε $q(\mathbf{h_1}, ..., \mathbf{h_T}) = \mathbf{h_T}$ τότε η μεταβλητή περιβάλλοντος θα είναι απλώς η κρυφή κατάσταση $\mathbf{h_T}$ της ακολουθίας εισόδου στο τελικό βήμα χρόνου T.

Αποκωδικοποιητής

Μέχρι αυτό το σημείο, η μεταβλητή περιβάλλοντος $\mathbf{c}$ της εξόδου του κωδικοποιητή, κωδικοποιεί ολόκληρη την ακολουθία εισόδου $x_1, ..., x_T$. Δεδομένης της ακολουθίας εξόδου $y_1, y_2, ..., y_{T'}$ από το σύνολο δεδομένων εκπαίδευσης, για κάθε βήμα t' (προσοχή, το σύμβολο διαφέρει από το χρονικό βήμα t των ακολουθιών εισόδου του κωδικοποιητή λόγω πιθανής διαφοράς στο μήκος ακολουθίας), η πιθανότητα εξόδου αποκωδικοποιητή να είναι $y_{t'}$ εξαρτάται από τις προηγούμενες εξόδους $y_1, ..., y_{t'-1}$ και τη μεταβλητή περιβάλλοντος $\mathbf{c}$ ως: $P(y_{t'} \mid y_1, \ldots, y_{t'-1}, \mathbf{c})$.
Για να μοντελοποιήθει αυτή η υπό συνθήκη πιθανότητα σε ακολουθίες, μπορούμε να χρησιμοποιήσουμε ένα άλλο RNN ή LSTM ή GRU ως αποκωδικοποιητή. Σε κάθε χρονικό βήμα t' στην ακολουθία εξόδου, το εκάστοτε δίκτυο RNN παίρνει την έξοδο $y_{t'-1}$ από το προηγούμενο χρονικό βήμα και τη μεταβλητή περιβάλλοντος $\mathbf{c}$ ως είσοδο, στη συνέχεια μετασχηματίζει αυτά και την προηγούμενη κρυφή κατάσταση $s_{t'-1}$ (συμβολισμός s για την έξοδο) στην κρυφή κατάσταση $s_{t'}$ του τρέχοντος χρονικού βήματος. Μπορούμε να χρησιμοποιήσουμε μια ακόμα μη γραμμική συνάρτηση g για να εκφράσουμε αυτόν το μετασχηματισμό της κρυφής στρώσης του αποκωδικοποιητή:

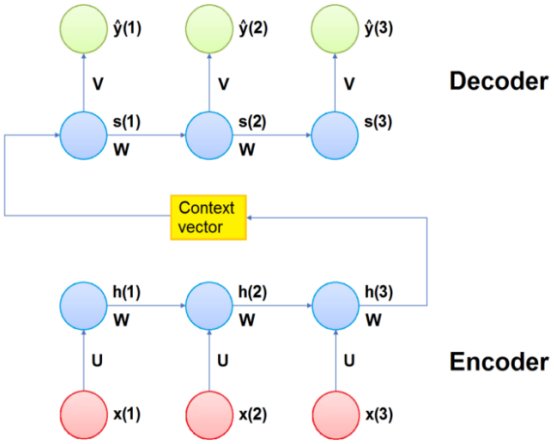
$$\mathbf{s}_{t'} = g(y_{t'-1}, \mathbf{c}, \mathbf{s}_{t'-1}) \tag{3.5.3}$$

Μετά την απόκτηση της κρυφής κατάστασης του αποκωδικοποιητή $\mathbf{s}$, μπορούμε να χρησιμοποιήσουμε ένα επίπεδο εξόδου και μια συνάρτηση ενεργοποίησης softmax για να

υπολογίσουμε την υπό συνθήκη πιθανότητα $P(y_{t'} \mid y_1, \ldots, y_{t'-1}, \mathbf{c})$ για την έξοδο κατά το χρονικό βήμα t'. Αυτό μπορεί να επιτευχθεί με την παρακάτω εξίσωση:

$$\hat{y} = softmax(Vs(t')) \tag{3.5.4}$$

όπου V κατάλληλη μήτρα βαρών στο στάδιο εξόδου του επαναληπτικού δικτύου στον αποκωδικοποιητή. Στην εικόνα 3.23 φαίνεται η δομή ενός απλού μοντέλου κωδικοποιητή - αποκωδικοποιητή. Η δύναμη αυτού του μοντέλου έγκειται στο ότι μπορεί να χαρτογραφήσει ακολουθίες διαφορετικών μηκών μεταξύ τους, καθώς οι είσοδοι και οι έξοδοι δεν συσχετίζονται και τα μήκη τους μπορεί να διαφέρουν. Ωστόσο, αυτό επιτυγχάνεται για μικρού μήκους ακολουθίες. Όταν το μήκος της ακολουθίας αυξάνεται είναι πολύ δύσκολο να συνοψίσουμε μια μακρά σε μήκος πληροφορία σε ένα μόνο διάνυσμα, με συνέπεια το μοντέλο συχνά να ξεχνά τα προηγούμενα μέρη της ακολουθίας εισόδου κατά την επεξεργασία των τελευταίων τμημάτων [77].



Εικόνα 3.23: Η αρχιτεκτονική ενός απλού Encoder-Decoder [77]

3.5.2 Ο μηχανισμός της Προσοχής (Attention Mechanism)

Για να αντιμετωπιστεί το προαναφερθέν πρόβλημα μπορούμε να χρησιμοποιήσουμε τον *μηχανισμό της προσοχής*. Αυτός ο μηχανισμός αντιμετωπίζει ακριβώς το πρόβλημα των πολύ μεγάλων ακολουθιών σε μήκος. Υπάρχουν 3 διαφορετικά είδη μηχανισμών προσοχής: α) Self Attention, β) Global (Hard) Attention και γ) Local (Soft) Attention.

- Self Attention: Η ιδέα είναι να συσχετίσουμε διαφορετικές θέσεις της ίδιας κρυφής κατάστασης που έχει προέλθει από την ακολουθία εισόδου προκειμένου να δημιουργηθεί μια αναπαράσταση, όπου κάθε στοιχείο της θα έχει συσχετισθεί (σε μικρότερο ή μεγαλύτερο βαθμό) με όλα τα υπόλοιπα στοιχεία του συνόλου [78].
- Global Attention: Η ιδέα είναι να αντλήσουμε ένα διάνυσμα περιβάλλοντος c_t, με βάση όλες τις κρυφές καταστάσεις του κωδικοποιητή. Ως εκ τούτου, αυτός ο μηχανισμός προσοχής παρακολουθεί όλο το χώρο εισόδου [74][79].

- Local Attention: Η ιδέα είναι η εξάλειψη του κόστους του Global Attention εστιάζοντας μόνο σε ένα μικρό υποσύνολο από τις ακολουθίες εισόδου. Θεωρώντας μια συγκεκριμένη θέση ευθυγράμμισης p_t, δημιουργούμε ένα παράθυρο $[p_t - D, p_t + D]$ επί της ακολουθίας εισόδου, όπου D το ήμισυ του μήκους του παραθύρου και αγνοούμε την πληροφορία που ξεπερνά τα όρια αυτά. Το p_t επιλέγεται είτε με: α) μονοτονική ευθυγράμμιση:

$$p_t = t \tag{3.5.5}$$

είτε με β) προβλεπτική ευθυγράμμιση:

$$p_t = S \cdot \sigma({\boldsymbol{v}_p}^T \cdot tanh(\mathbf{W}_p \mathbf{h}_t)) \tag{3.5.6}$$

όπου S το μήκος της ακολουθίας εισόδου, ${\boldsymbol{v}_p}^T$ και $\mathbf{W}_p$ κατάλληλες παράμετροι του μοντέλου προς εκπαίδευση και $\mathbf{h}_t$ η κρυφή κατάσταση του μοντέλου που εφαρμόζουμε το μηχανισμό [80].

Στην παρούσα εργασία πλησιάσαμε περισσότερο προς το Global Attention λαμβάνοντας σε κάποια σημεία, χρήσιμα στοιχεία και από τα υπόλοιπα είδη. Πιο αναλυτικά λοιπόν, ο βασικός στόχος είναι να επιτραπεί στον αποκωδικοποιητή να έχει επιλεκτική πρόσβαση σε πληροφορίες από τον κωδικοποιητή κατά την αποκωδικοποίηση. Αυτό μπορεί να συμβεί με την οικοδόμηση ενός διαφορετικού διανύσματος περιβάλλοντος για κάθε χρονικό βήμα του αποκωδικοποιητή, το οποίο υπολογίζουμε σε συνάρτηση με την προηγούμενη κρυφή κατάσταση (του αποκωδικοποιητή) αλλά και όλων των κρυφών καταστάσεων του κωδικοποιητή, εκχωρώντας τους κατάλληλα βάρη. Ο μηχανισμός αυτός δίνει περισσότερη σημασία στα πιο σημαντικά στοιχεία της ακολουθίας εισόδου, αποδίδοντάς τους μεγαλύτερη προσοχή.
Διατηρώντας την δομή του Encoder-Decoder μπορούμε να εφοδιάσουμε το δικτύο μας με το μηχανισμό αυτόν. Συνεπώς, η έξοδος του κωδικοποιητή θα έχει την ίδια μορφή με την εξίσωση (3.5.1). Ωστόσο, όπως αναφέραμε, πλέον το διάνυσμα περιβάλλοντος $\mathbf{c}(t)$ είναι συνάρτηση του κάθε βήματος t του αποκωδικοποιητή. Προκειμένου να λάβουμε αυτήν την μεταβλητή ακολουθείται η παρακάτω διαδικασία [77].

Αρχικά, για κάθε συνδυασμό του χρονικού βήματος j του κωδικοποιητή και του χρονικού βήματος t του αποκωδικοποιητή, το *σκορ ευθυγράμμισης* $\mathbf{e}(j,t)$ υπολογίζονται με το ακόλουθο σταθμισμένο άθροισμα:

$$\mathbf{e}(j,t) = \boldsymbol{V}^T tanh(\mathbf{W_2 s}(t-1), + \mathbf{W_1 h}(j) + \mathbf{b}) \tag{3.5.7}$$

Τα $\mathbf{W_1}$ και $\mathbf{W_2}$ είναι κατάλληλα βάρη που σχετίζονται με τις κρυφές καταστάσεις του κωδικοποιητή και του αποκωδικοποιητή αντίστοιχα, το $\mathbf{V}$ είναι βάρη κατάλληλα για το τελικό σκορ ευθυγράμμισης ενώ το $\mathbf{b}$ αποτελεί διάνυσμα bias που συνήθως αρχικοποιείται στο μηδέν (ή/και αγνοείται). Με $\mathbf{h}$ και $\mathbf{s}$ συμβολίζουμε και πάλι τις κρυφές καταστάσεις του Encoder και του Decoder αντίστοιχα.
Για κάθε χρονικό βήμα t του αποκωδικοποιητή, τα σκορ $\mathbf{e}(j,t)$ κανονικοποιούνται χρησιμοποιώντας τη συνάρτηση softmax για κάθε χρονικό βήμα j του κωδικοποιητή, και έτσι λαμβάνουμε τα λεγόμενα *βάρη προσοχής ή διάνυσμα προσοχής* $\boldsymbol{\alpha}(j,t)$:

$$\boldsymbol{\alpha}(j,t) = softmax(\mathbf{e}(j,t)) = \frac{exp(\mathbf{e}(j,t))}{\sum_{j=1}^{N} exp(\mathbf{e}(j,t))} \tag{3.5.8}$$

Το N στη σχέση (3.5.8) υποδηλώνει το συνολικό πλήθος των χρονικών βημάτων στον κωδικοποιητή. Το κάθε στοιχείο του διανύσματος προσοχής $\boldsymbol{\alpha}(j,t)$ καταγράφει τη σημασία της εισόδου στο χρονικό βήμα j, ως προς την αποκωδικοποίηση της εξόδου στο χρονικό βήμα t. Με άλλα λόγια, κατά την αποκωδικοποίηση της πληροφορίας τη χρονική στιγμή t καταγράφει πόση προσοχή χρειάζεται να δώσουμε στην κωδικοποιημένη είσοδο του βήματος j. Στη συνέχεια, το διάνυσμα περιβάλλοντος $\mathbf{c}(t)$ υπολογίζεται ως το σταθμισμένο άθροισμα όλων των κρυφών τιμών του κωδικοποιητή σύμφωνα με τα βάρη προσοχής:

$$\mathbf{c}(t) = \sum_{j=1}^{T} \boldsymbol{\alpha}(k,t)\mathbf{h}(j) \tag{3.5.9}$$

Στη σχέση (3.5.9) το T δηλώνει το συνολικό πλήθος των χρονικών βημάτων στον αποκωδικοποιητή. Αυτό το διάνυσμα περιβάλλοντος πλέον, επιτρέπει να δοθεί μεγαλύτερη προσοχή στις πιο σχετικές ακολουθίες εισόδου, δηλαδή αυτές που έχουν μεταξύ τους περισσότερη εξάρτηση. Έπειτα το διάνυσμα περιβάλλοντος $\mathbf{c}(t)$ μεταβιβάζεται στον αποκωδικοποιητή, ο οποίος υπολογίζει την κατανομή πιθανότητας της επόμενης πιθανής εξόδου. Αυτή η διαδικασία αποκωδικοποίησης λαμβάνει χώρα για όλα τα χρονικά βήματα που υπάρχουν στην είσοδο (του αποκωδικοποιητή). Ως εκ τούτου, η τρέχουσα κρυφή κατάσταση $\mathbf{s}(t)$ υπολογίζεται σύμφωνα με την περιοδική συνάρτηση μονάδας, λαμβάνοντας ως είσοδο το διάνυσμα περιβάλλοντος $\mathbf{c}(t)$, την κρυφή κατάσταση $\mathbf{s}(t-1)$ και την έξοδο $\hat{y}(t-1)$ του προηγούμενου χρονικού βήματος, ως:

$$\mathbf{s}(t) = g(\mathbf{c}(t), \mathbf{s}(t-1), \hat{y}(t-1)) \tag{3.5.10}$$

Με αυτόν τον τρόπο, το μοντέλο είναι σε θέση να βρει τους πιο στενούς συσχετισμούς μεταξύ διαφορετικών τμημάτων της ακολουθίας εισόδου και αντίστοιχων τμημάτων της ακολουθίας εξόδου και να χρησιμοποιήσει αυτή την πληροφορία για να κάνει πιο ακριβή πρόβλεψη. Την ίδια στιγμή είναι σε θέση να συγκρατήσει αυτήν την εξάρτηση ακόμα και για πολύ μεγάλου μήκους ακολουθίες.
Το σκορ ευθυγράμμισης από τη σχέση (3.5.7) δεν είναι ο μοναδικός τρόπος να 'ευθυγραμμίσουμε' την δεδομένη πληροφορία. Σύμφωνα και με την βιβλιογραφία, υπάρχουν αρκετοί διαφορετικοί τρόποι [81][82]. Για παράδειγμα, ένας εναλλακτικός τρόπος είναι να αλληλοσυσχετίσουμε το δοθέν κομμάτι πληροφορίας από τον Encoder με κάποια δέσμη, $\mathbf{x}_k = ({x_1}^k, {x_2}^k, ...{x_m}^k) \in \mathbb{R}^m$, από το διάνυσμα εισόδου. Υποθέτοντας ότι ο κωδικοποιητής μας είναι ένα LSTM τότε στην έξοδο αυτού θα έχουμε την κρυφή κατάσταση $\mathbf{h}_t$ και το κελί μνήμης $\mathbf{C}_t$. Συνενώνοντας αυτά τα 2 μπορούμε να αναπαράξουμε τη σχέση (3.5.7) και να ευθυγραμμίσουμε την πληροφορία του Encoder με την είσοδο ως εξής:

$$\mathbf{e}_{k,t} = \mathbf{V}^T tanh(\mathbf{W}_1[\mathbf{h}_{t-1}, \mathbf{C}_{t-1}] + \mathbf{W}_2\mathbf{x}_k + \mathbf{b}) \tag{3.5.11}$$

Σε αυτήν την περίπτωση λαμβάνουμε το διάνυσμα περιβάλλοντος ως το γινόμενο πολλαπλασιασμού στοιχείου προς στοιχείο (dot-product) μεταξύ των βαρέων προσοχής και του αρχικού διανύσματος εισόδου ως:

$$\mathbf{c}_t = (\boldsymbol{\alpha}_{j,t} \cdot \mathbf{x}_k)^T = ({\alpha_t}^1 {x_t}^1, {\alpha_t}^2 {x_t}^2, ..., {\alpha_t}^n {x_t}^n)^T \tag{3.5.12}$$

Οι εξισώσεις (3.5.7) και (3.5.11) αποτέλουν μια τροποποιημένη εκδοχή της *προσθετικής προσοχής* (additive attention) που πρότεινε ο Bahdanau στο [74]. Αργότερα ο Luong

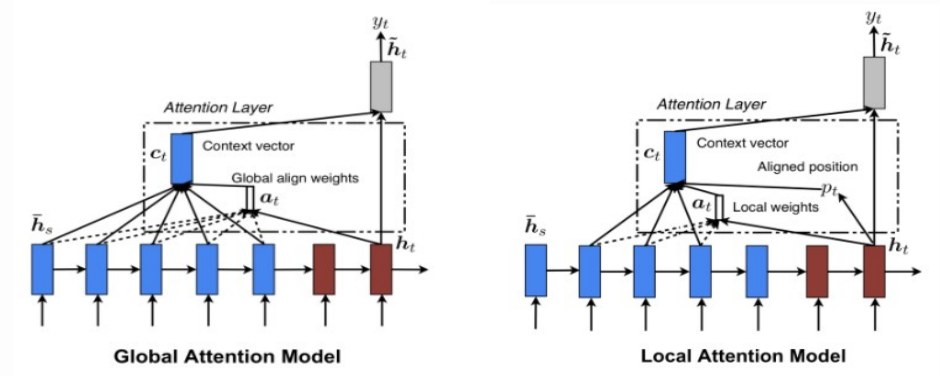
[80] πρότεινε επιπλέον τρόπους με τους οποίους μπορεί να υπολογιστεί το σκορ ευθυγράμμισης. Θεωρώντας και πάλι ως $\mathbf{h}_j$ και $\mathbf{s}_t$, τις κρυφές καταστάσεις του Encoder και του Decoder και με j, t τις χρονικές τους μεταβλητές τους αντίστοιχα μπορούμε να ορίσουμε τα σκορ ευθυγράμμισης ως:

$$\text{align score} = \mathbf{e}(j,t) = \begin{cases} \boldsymbol{s}_t{}^T \cdot \mathbf{h}_j, & \text{dot-product} \\ \boldsymbol{s}_t{}^T \mathbf{W}_e \mathbf{h}_j, & \text{general} \end{cases} \tag{3.5.13}$$

όπου $\mathbf{W}_e$ κατάλληλη μήτρα βαρών προς εκπαίδευση. Τέλος, στην περίπτωση που θέλουμε να εισάγουμε το στοιχείο της τοπικότητας στον μηχανισμό μας (local attention) θα πρέπει να περιορίσουμε το μήκος του διανύσματος προσοχής. Αυτό μπορεί να συμβεί χρησιμοποιώντας μια Γκαουσιανή κατανομή γύρω από τη θέση ευθυγράμμισης p_t (μέση τιμή) και τυπική απόκλιση σ. Από τη σχέση (3.5.6) έχουμε ήδη ότι $p_t \in [0, S]$ οπότε το σκορ ευθυγράμμισης μπορεί να υπολογιστεί από την εξής σχέση:

$$\mathbf{e}(j,t)_L = \mathbf{e}(j,t) \exp\left(-\frac{(u-p_t)^2}{2\sigma^2}\right) \tag{3.5.14}$$

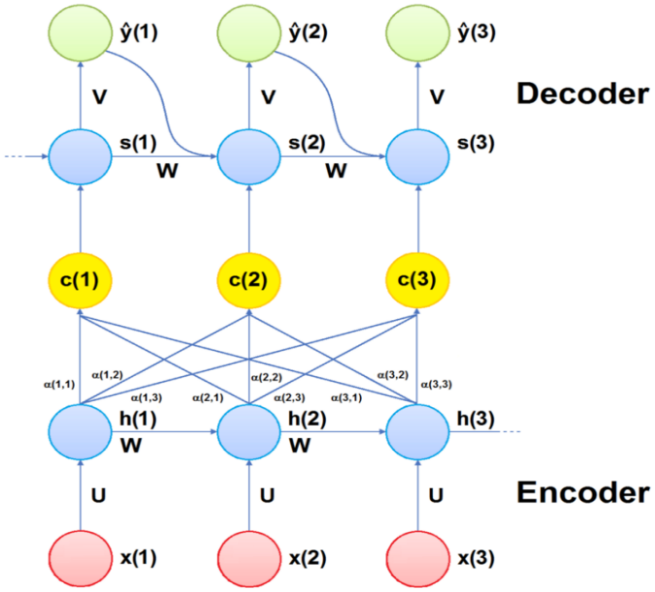
όπου u είναι ένα υποσύνολο εστίασης πάνω στην κρυφή κατάσταση $\mathbf{h}$ του κωδικοποιητή. Στις εικόνες 3.24α′ και 3.24β′ μπορούμε να δούμε 2 μοντέλα που υλοποιούν τους μηχανισμούς Global και Local Attention αντίστοιχα. Τέλος, στην εικόνα 3.25 βλέπουμε μια επέκταση του μοντέλου κωδικοποιητή-αποκωδικοποιητή της εικόνας 3.23, εφοδιασμένο με τον μηχανισμό της προσοχής που περιγράφουν οι εξισώσεις (3.5.7) έως και (3.5.10).



(α′) Μοντέλο που υλοποιεί τον μηχανισμό Global Attention

(β′) Μοντέλο που υλοποιεί τον μηχανισμό Local Attention

Εικόνα 3.24: Αναπαράσταση των μηχανισμών Global και Local Attention [80]



Εικόνα 3.25: Η αρχιτεκτονική ενός Attention Encoder-Decoder [77]

Κεφάλαιο 4

Ανάλυση τεχνικών και αρχιτεκτονικών

4.1 Λογισμικό και Προεπεξεργασία των Δεδομένων

4.1.1 Βιβλιοθήκες λογισμικού

Η μελέτη, η ανάπτυξη και εκπαίδευση μοντέλων νευρωνικών δικτύων που είδαμε στο Κεφάλαιο 3 , επιτυγχάνεται σχετικά απλά με τη χρήση λογισμικού υψηλού επιπέδου και βιβλιοθηκών μηχανικής μάθησης, που είναι εξειδικευμένες στην εφαρμογή και στην εκτέλεση σχετικών λειτουργιών. Πιο συγκεκριμένα, το TensorFlow [83], το Mxnet [84] και η PyTorch [85] είναι τα πιο δημοφιλή λογισμικά μηχανικής μάθησης ανοιχτού κώδικα και μπορούν να χρησιμοποιηθούν από τη γλώσσα προγραμματισμού Python ως βιβλιοθήκες. Στην παρούσα εργασία έγινε χρήση του TensorFlow και συγκεκριμένα του Keras [86], μιας βιβλιοθήκης του TensorFlow, που αποτελεί Διεπαφή Προγραμματισμού Εφαρμογών - ΔΠΕ (API) υψηλού επιπέδου για μηχανική μάθηση. Με τη βοήθεια του Keras, η δημιουργία και η εκπαίδευση των πολύπλοκων νευρωνικών δικτύων, όπως τα LSTMs ή τα αμφίδρομα LSTMs, γίνεται αρκετά ευκολότερη και ταχύτερη. Το Keras είναι σχεδιασμένο να επιτελεί επιταχυνόμενους υπολογισμούς (accelerated computations) σε ξεχωριστές μονάδες υπολογιστών (hardware). Ακόμη, υπάρχει η δυνατότητα χρήσης μονάδων επιτάχυνσης υπολογισμών που επιτρέπουν θεαματική βελτίωση στους χρόνους εκτέλεσης, όπως η GPU [87] και η TPU [88] έως και 100 φορές.
Οι βασικές δομές δεδομένων του Keras είναι τα *στρώματα* (layers) και τα *μοντέλα* (models). Στην πρώτη κατηγορία, ανήκουν όλα τα προαναφερθέντα είδη νευρωνικών δικτύων τα οποία χρησιμοποιούνται με μεγάλη ευκολία ως στρώματα παρατεταγμένα το ένα μετά το άλλο δημιουργώντας πολλαπλές στρώσεις δικτύων. Αυτές οι πολλαπλές στρώσεις αποτελούν τα λεγόμενα *ακολουθιακά μοντέλα* (sequential models). Πιο περίπλοκες αρχιτεκτονικές απαιτούν τη χρήση *λειτουργικών ΔΠΕ* (functional API), που μας δίνουν την δυνατότητα να κατασκευαστούν μοντέλα σε μορφή γράφων με πληθώρα διασυνδέσεων ανάμεσα σε τύπους στρωμάτων ή και μόντελα από την αρχή (functional

models, from scratch). Στην εργασία χρησιμοποιήθηκαν και οι δύο τύποι μοντέλων.

Στο Keras οι μονάδες επεξεργασίας, οποιουδήποτε τύπου μοντέλων και αν επιλέξουμε, είναι οι τανυστές (tensors) οι οποίοι στο συγκεκριμένο λογισμικό (TensorFlow-Keras) μπορούν να θεωρηθούν και ως απλοί πολυδιάστατοι πίνακες[1] της βιβλιοθήκης Numpy [89]. Η τελευταία αποτελεί τη θεμελιώδη βιβλιοθήκη για αλγεβρικούς και πολλούς άλλους μαθηματικούς υπολογισμούς της Python. Οι πίνακες αυτοί θα πρέπει να είναι τουλάχιστον 3-διάστατης μορφής ή και παραπάνω για να μπορούν να είναι διαχειρίσιμοι από τα επαναληπτικά και τα συνελικτικά νευρωνικά δίκτυα που παρέχει το Keras. Στις εικόνες 4.1 και 4.2 βλέπουμε ένα ακουλουθιακού τύπου μοντέλο και ένα σύνολο τανυστών αντίστοιχα, οι οποίοι θα μπορούσαν να αποτελούν είσοδο στο δίκτυο της εικόνας 4.1

```python
# Define Sequential model with 3 layers
model = keras.Sequential(
    [
        layers.Dense(2, activation="relu", name="layer1"),
        layers.Dense(3, activation="relu", name="layer2"),
        layers.Dense(4, name="layer3"),
    ]
)
# Call model on a test input
x = tf.ones((3, 3))
y = model(x)
```

Εικόνα 4.1: Ακολουθιακού τύπου μοντέλο στο Keras [90]

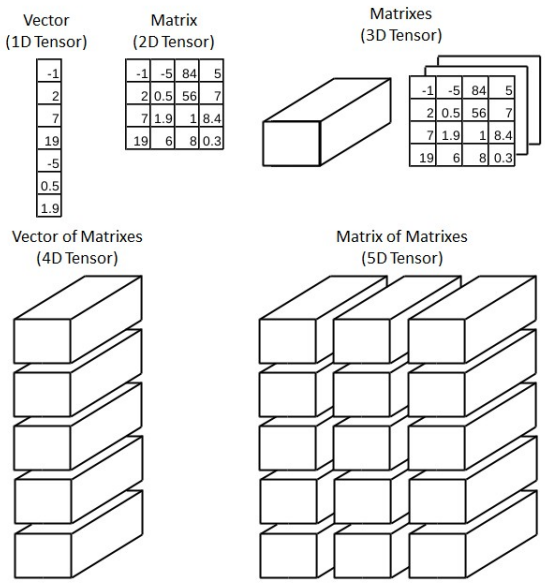
4.1.2 Παρουσίαση και βασικές τεχνικές προεπεξεργασίας του Dataset

Γνωριμία με το Dataset

Όπως έχει προαναφερθεί η συλλογή των δεδομένων έγινε από συγκεκριμένες πηγές ενώ κατά τη διάρκεια της εργασίας σχηματίστηκαν και νέες μεταβλητές όπως είδαμε στο Κεφάλαιο 2 (κρούσματα ανά εκατομμύριο, ποσοστό θνησιμότητας κλπ). Η διαθεσιμότητα όλων των δεδομένων (κυρίως η εγκυρότητα των διεξαγόμενων τεστ και των εισαγωγών στα νοσοκομεία και τις ΜΕΘ) για όλες τις χώρες δεν ήταν δυνατή, για αυτό σε ορισμένες περιπτώσεις βασιστήκαμε μόνο στα κυριότερα χαρακτηριστικά, δηλαδή τους θανάτους και τα κρούσματα, που καθολικά κατά συντριπτική πλειοψηφία τους, δόθηκαν ορθώς στον Παγκόσμιο Οργανισμό Υγείας. Σε περιπτώσεις που τα δεδομένα και από άλλες μετρικές ήταν πλήρη, χρησιμοποιήθηκαν επιπλέον μεταβητές στην προσπάθεια ακριβέστερης πρόβλεψης.

Κατά τη διάρκεια της εργασίας οι μεταβλητές που έγινε προσπάθεια να προβλεθούν ήταν οι μεταβλητές των κρουσμάτων και των θανάτων τόσο λόγω της αρτιότητας τους ως προς την παροχή όσο και από το γεγονός ότι καθ' όλη την διάρκεια της πανδημίας το μεγαλύτερο ενδιαφέρον συγκεντρώθηκε το πως θα εξελιχθεί η νόσος σε επίπεδο μολύνσεων και απωλειών. Μεταξύ των πολλών χαρακτηριστικών του συνόλου δεδομένων (Dataset) ήταν: αθροιστικά κρούσματα, αθροιστικοί θάνατοι, αθροιστικά τεστ, καθημερινά κρούσματα, καθημερινοί θάνατοι, καθημερινά τεστ, αριθμός νοσηλειών, αριθμός

[1] σε αντίθεση για παράδειγμα με την Pytorch που η χρήση τανυστών είναι υποχρεωτική



Εικόνα 4.2: Τανυστές διαφόρων μεγεθών ως πίνακες [91]

διασωληνωμένων, αριθμός ασθενών σε ΜΕΘ [2], ποσοστό θνησιμότητας, καθημερινό ποσοστό θετικότητας, αθροιστικοί εμβολιασμοί, καθημερινοί εμβολιασμοί κ.α.
Θα πρέπει σε αυτό το σημείο να τονιστεί, ούτως ή αλλιώς, ότι η πρόβλεψη χρονοσειρών χωρίζεται σε 2 είδη: α) στην *μονοδιάστατη* πρόβλεψη χρονοσειρών (univariate time series forecasting) και β) στην *πολυδιάστατη* πρόβλεψη χρονοσειρών (multivariate time series forecasting) [92][93]. Στην πρώτη μέθοδο, η εξαρτημένη μεταβλητή (η μεταβλητή που θέλουμε να προβλέψουμε) είναι ταυτόχρονα και μεταβλητή παρατήρησης, ενώ στη δεύτερη, ενδεχομένως και πάλι, η εξαρτημένη μεταβλητή να είναι ταυτόχρονα και μεταβλήτη παρατήρησης αλλά σε τούτη την περίπτωση στις μεταβλητές παρατήρησης συγκαταλέγονται επίσης και άλλες μεταβλητές που ίσως έχουν στενή συσχέτιση με το χαρακτηριστικό που θέλουμε να προβλέψουμε.
Για παράδειγμα, στην περίπτωσή μας, μια μονοδιάστατη πρόβλεψη θα βασίζονταν στην παρατήρηση των κρουσμάτων του παρελθόντος προκειμένου να προβλέψουμε τα κρούσματα του μέλλοντος. Αντίστοιχα, μια πολυδιάστατη πρόβλεψη θα βασίζονταν και σε άλλες μεταβλητές προκειμένου να προβλεφθούν τα κρούσματα του μέλλοντος. Απτά παραδείγματα τέτοιας εξάρτησης αποτελούν οι αριθμοί των τεστ με τα κρούσματα ή το πως έχει εξελιχθεί η πανδημία έως αυτό το χρονικό σημείο σε μια συγκεκριμένη τοποθεσία (έμμεσα, δηλαδή, υπονοείται εξάρτηση από το δείκτη των κρουσμάτων/εκατομμύριο). Ακόμη, αδιαμφισβήτητα οι άνθρωποι που βρίσκονται στα νοσοκομεία και στις ΜΕΘ μας βοηθάνε στο να κάνουμε μια καλύτερη εκτίμηση των απωλειών (θανάτων).

[2] Όπου ο αριθμός αυτός διέφερε από τον αριθμό των διασωληνωμένων

Πίνακες Αλληλοσυσχέτισης

Προκειμένου να αποφανθούμε κατά πόσο, όλα τα δεδομένα που έχουμε στη διάθεσή μας αλληλοσχετίζονται μεταξύ τους αξιοποιούμε τη γνωστή βιβλίοθηκη της Python, Pandas [94], που μας δίνει την δυνατατότητα με απλό τρόπο να κατασκευάσουμε τον *πίνακα αλληλοσυσχέτισης* (Correlation Matrix) των χαρακτηριστικών.

Ο πίνακας αλληλοσυσχέτισης μας δείχνει κατά πόσο σχετίζονται τα δεδομένα μεταξύ τους, είναι συμμετρικός και θετικά ημι-ορισμένος και η κύρια διαγώνιος του περιέχει τις τιμές αυτοσυσχέτισης (δηλαδή, τη συσχέτιση κάθε στοιχείου με τον εαυτό του). Συνεπώς οι τιμές αυτές είναι 1. Όλες οι τιμές του πίνακα αυτού κυμαίνονται στο διάστημα $[-1, 1]$. Στην πραγματικότητα για μια παρατήρηση $\mathbf{X} = [X_i, ..., X_n]$, ο πίνακας αυτός είναι $n \times n$ και εμπεριέχει τις κανονικοποιημένες μέσες τιμές των τυχαίων μεταβλητών X_i δια την τυπική απόκλιση σ, δηλαδή, $X_i/\sigma(X_i)$ για $i = 1, ..., n$. Οπότε παίρνουμε:

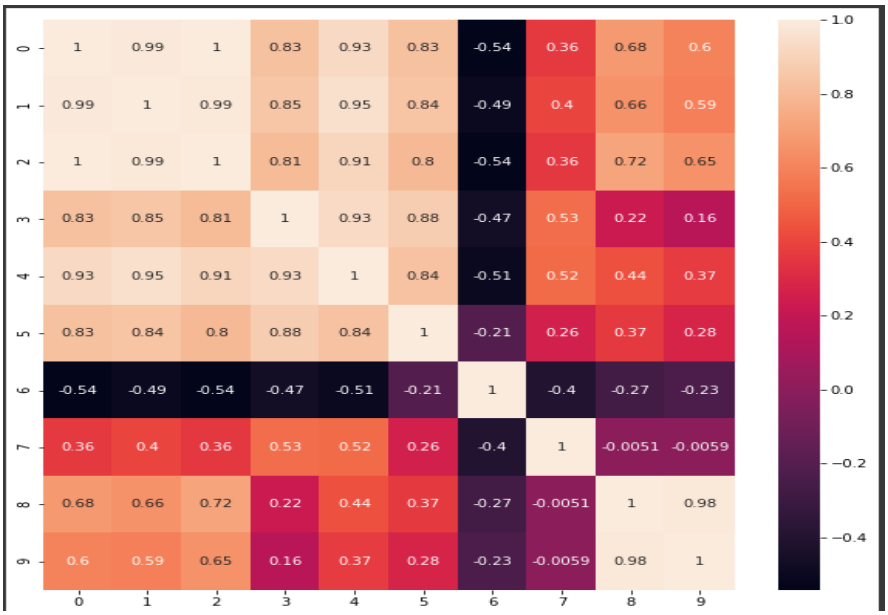
$$corr(\mathbf{X}) = \begin{bmatrix} 1 & \frac{E[(X_1-\mu_1)(X_2-\mu_2)]}{\sigma(X_1)\sigma(X_2)} & \dots & \frac{E[(X_1-\mu_1)(X_2-\mu_2)]}{\sigma(X_1)\sigma(X_2)} \\ \frac{E[(X_2-\mu_2)(X_1-\mu_1)]}{\sigma(X_2)\sigma(X_1)} & 1 & \dots & \frac{E[(X_2-\mu_2)(X_n-\mu_n)]}{\sigma(X_2)\sigma(X_n)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{E[(X_n-\mu_n)(X_1-\mu_1)]}{\sigma(X_n)\sigma(X_1)} & \frac{E[(X_n-\mu_n)(X_2-\mu_2)]}{\sigma(X_n)\sigma(X_2)} & \dots & 1 \end{bmatrix} \tag{4.1.1}$$

όπου μ_i είναι η μέση τιμή της τυχαίας μεταβλητής X_i.
Τιμές συσχέτισης κοντά στο 1 δηλώνουν ότι τα 2 χαρακτηριστικά έχουν μέγαλη συσχέτιση ενώ τιμές κοντά στο -1 δηλώνουν αντίθετη συσχέτιση μεταξύ των μεταβλητών. Τέλος τιμές κοντά στο 0 δηλώνουν αμυδρή εξάρτηση μεταξύ των μεταβλητών.
Στις εικόνες 4.3α′, 4.3β′ απεικονίζονται 2 τυπικοί πίνακες αλληλοσυσχέτισης για τα παγκόσμια δεδομένα της νόσου (10 μεταβλητές) και για τα δεδόμενα των ΗΠΑ ([22], 16 μεταβλητές) αντίστοιχα. Στην πρώτη εικόνα τα χαρακτηριτικά είναι υπό αριθμητική κωδικοποίηση και με τη σειρά είναι: αθροιστικά κρούσματα, αθροιστικοί θάνατοι, αθροιστικά τέστ, καθημερινά κρούσματα, καθημερινά τεστ, καθημερινοί θάνατοι,ποσοστό θνησιμότητας, καθημερινό ποσοστό θετικότητας, αθροιστικοί εμβολιασμοί, καθημερινοί εμβολιασμοί. Στη δεύτερη εικόνα τα χαρακτηριστικά εμφανίζονται ονοματικώς.

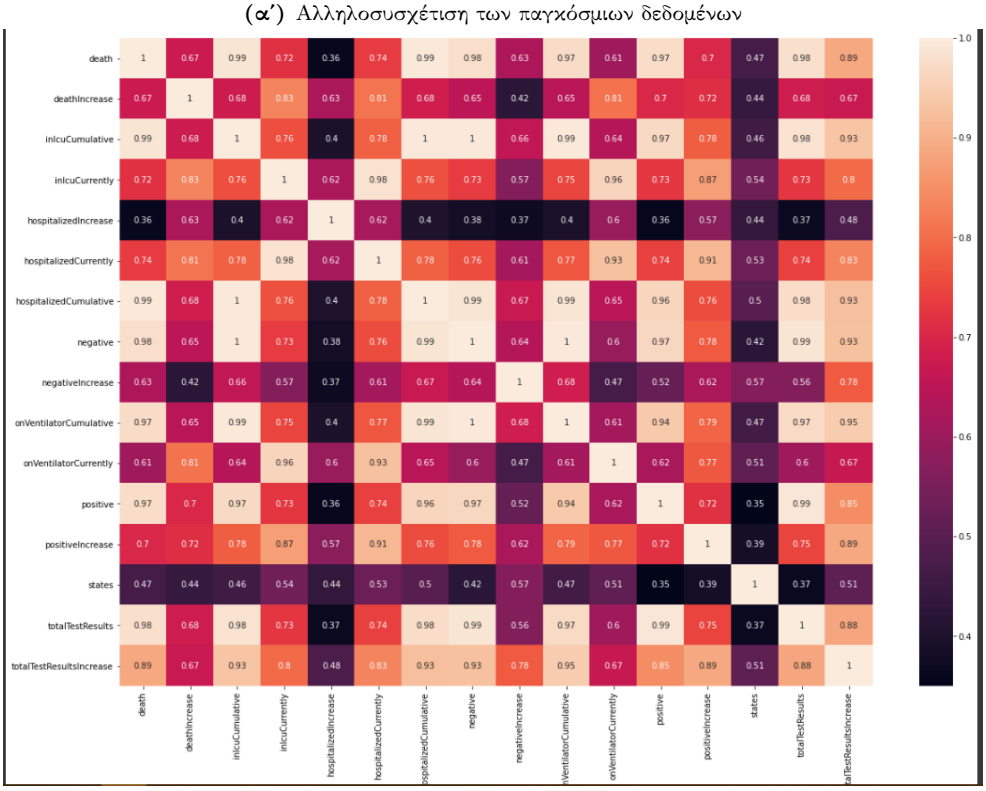
Από την εικόνα 4.3α′ είναι εμφανές, ότι τα πρώτα 6 χαρακτηριστικά παρουσιάζουν μεγάλη μεταξύ τους συσχέτιση, πράγμα που σημαίνει ότι ενδέχομενως να βοηθήσει η χρήση της παραπάνω από μιας μεταβλητής κατά την πρόβλεψη. Αντίστοιχα, στην εικόνα 4.3β′ υπάρχουν μεμονωμένα χαρακτηριστικά που εμφανίζουν υψηλή αλληλοσυσχέτιση. Αξίζει να σημειωθεί ότι οι πίνακες αυτοί θα μπορούσαν να είναι κάτω τριγωνικοί μιας και είναι συμμετρικοί.

Διάκριση μεταξύ αθροιστικών καμπυλών και χρονοσειρών

Σε πολλά σημεία έως τώρα έχουμε αναφέρει τις έννοεις 'αθροιστικός' και 'καθημερινός'. Υπάρχει μια ειδοποιός διαφορά μεταξύ αυτών των 2 χαρακτηρισμών. Οι αθροιστικές καμπύλες είναι συνεχώς αυξανόμενες και σε πολλές περιπτώσεις ομοιάζουν με κάποια εκθετική ή λογαριθμική συνάρτηση και δηλώνουν το συσσωρευτικό αριθμό της μετρούμενης παρατήρησης. Για το λόγο αυτό δεν αποτελούν χρονοσειρά. Αντίθετα, οι καθη-



(α′) Αλληλοσυσχέτιση των παγκόσμιων δεδομένων

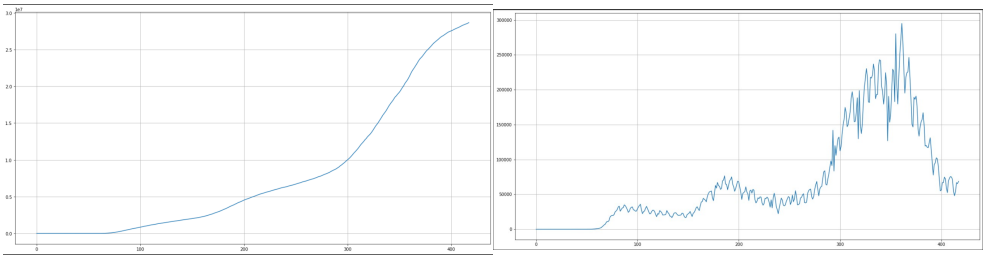


(β′) Αλληλοσυσχέτιση δεομένων των ΗΠΑ

Εικόνα 4.3: Αλληλοσυσχέτιση διαφόρων δεδομένων

μερινές καμπύλες εμφανίζουν συνεχώς τις αυξομειώσεις της μετρούμενης παρατήρησης και για αυτό αποτελούν χρονοσειρά. Η συνεχώς αυξητική πορεία των αθροιστικών καμπυλών ενδεχομένως να ενέχει μεγαλύτερη ευκολία κατά την πρόβλεψη και για αυτό αποφεύγεται η εκτενής αναλύση της.

Η εργασία, αντίθετα με την βιβλιογραφία, εστιάζει στην πρόβλεψη καθημερινών καμπυλών πράγμα ιδιαιτέρως πιο απαιτητικό, ρεαλιστικό και προκλητικό [95] [96] [97][98][110]. Στις εικόνες 4.4α′ και 4.4β′ βλέπουμε δυο τέτοιες καμπύλες αθροιστικών και καθημερινών κρουσμάτων στις ΗΠΑ. Από τις εικόνες αυτές γίνεται εύκολα αντιληπτό ότι η προσπάθεια προβλέψης δυσχεραίνεται στην περίπτωση της εικόνας 4.4β′



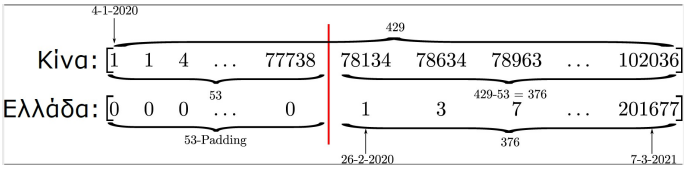
(α′) Αθροιστική καμπύλη (β′) Καθημερινής καμπύλη

Εικόνα 4.4: Διαφορά Αθροιστικής και Καθημερινή καμπύλης

Μέθοδοι Padding και Smoothing

Νωρίτερα στο Κεφάλαιο 3 είχαμε αναφέρει την μέθοδο του γεμίσματος (padding) για τα συνελικτικά νευρωνικά δίκτυα ενώ στο Κεφάλαιο 2 αναφέραμε ότι επειδή η πανδημία δεν ξέσπασε σε όλες τις χώρες την ίδια και χρονική στιγμή υπήρξε η ανάγκη για 'ευθυγράμμιση' των δεδομένων. Η αρχή της πανδημίας του κορωνοϊού έγινε στην Κίνα στις 4-1-2020 ενώ σε όλες τις υπόλοιπες χώρες του κόσμου η νόσος έφτασε μεταγενέστερα με συνέπεια η επιδημιολογική επιτήρηση της νόσου να είναι χρονικά μικρότερη σε διάρκεια από τα αντίστοιχα δεδομένα της Κίνας. Η μέθοδος του γεμίσματος που υλοποιήθηκε, είναι αυτή με την οποία όλα τα δεδόμενα όλων των χωρών αποκτούν το ίδιο μήκος ακολουθίας με τα δεδομένα της Κίνας. Το γέμισμα γίνεται με μηδενικές τιμές που στο πλήθος είναι όσες οι ημέρες που μεσολάβησαν από τις 4-1-2020 έως και την ημερομηνία έναρξης της πανδημίας στην εκάστοτε χώρα.
Για παράδειγμα στην πατρίδα μας, την Ελλάδα, το πρώτο καταγεγραμμένο κρούσμα αναφέθηκε στις 26-2-2020 [99]. Αυτό σημαίνει ότι ο ιός έφτασε (τυπικά) στη χώρα μας 53 ημέρες μετά από την εμφάνισή του στη Γιουχάν της Κίνας. Συνεπώς, όλες οι ακολουθίες δεδομένων της Ελλάδας 'γεμίστηκαν' με 53 μηδενικά στην αρχή τους προκειμένου να γίνει μεταξύ άλλων και αρκέτα ευκολότερη η διαχείριση των δεδομένων αυτών. Στην εικόνα 4.5 παρουσιάζουμε ένα παράδειγμα της μεθόδου padding για τα αθροιστικά κρούσματα της Ελλάδας σε σχέση με τους αντίστοιχους αριθμούς της Κίνας.

Μια ακόμα διαδικασία που χρειάστηκε μεμονωμένα και σε λίγες περιπτώσεις κάτα την διάρκεια της εργασίας είναι αυτή της *εξομάλυνσης* των δεδομένων (smoothing). Κάτα την διαδικασία αυτή όταν τα δεδομένα μας παρουσίασαν ακραία μεγάλες διακύμανσεις



Εικόνα 4.5: Αναπαράσταση μεθόδου γεμίσματος padding στα δεδομένα

από παρατήρηση σε παρατήρηση (από μέρα σε μέρα) τότε αντικαθιστούσαμε την ακραία παρατήρηση με τον επταήμερο κυλιόμενο μέσον όρο των γειτονικών παρατηρήσεων ($\pm$ενδεχομένως μια σταθερά $c > 0$ προκειμένου να 'διατηρηθεί η απότομη αύξηση ή μείωση' της χρονοσειράς σε εκείνο το σημείο).
Θα πρέπει να σημειωθεί ότι η συγκεκριμένη μέθοδος εφαρμόστηκε μόνο όταν αποφανθήκαμε ότι υπάρχει λάθος στα δεδομένα, καθώς σε αντίθετη περίπτωση, όπως προαναφέραμε, η ύπαρξη απότομων αυξομειώσεων και η επικείμενη δυσκολία πρόβλεψης αυτών από τα μοντέλα αποτέλεσαν μέρος των πειραματισμών. Ένα τέτοιο παράδειγμα συνέβη στα τέλη Νοεμβρίου 2020 όταν οι τουρκικές αρχές συμπεριέλαβαν στις καταγραφές τους, ασυμπτωματικά κρούσματα για όλο το προηγούμενο διάστημα αυξάνοντας τα παγκόσμια κρούσματα για εκείνο το διάστημα με πολύ απότομο τρόπο [100].

4.1.3 Μέθοδοι παραθυροποίησης και κανονικοποίησης στις χρονοσειρές

Μέθοδος παραθυροποίησης

Όπως έχει γίνει ίσως ήδη αντιληπτό τα δεδομένα εισαγωγής, αρχικά είναι στη μορφή ενός διανύσματος, αν χρησιμοποιούμε ένα μόνο χαρακτηριστικό, ή στη μορφή δισδιάστατου πίνακα αν γίνεται χρήση παραπάνω του ενός χαρακτηριστικού. Στις γραμμές αυτών των πινάκων βρίσκονται οι ημερομηνίες από την αρχή της πανδημίας έως και την τελευταία ημέρα της παρατήρησης. Στις στήλες βρίσκονται το ένα ή παραπάνω χαρακτηριστικά που χρησιμοποιήθηκαν (κρούσματα, θάνατοι κλπ). Στην εικόνα 4.6α΄ φαίνεται ένας διασδιάστατος πίνακας όπου στις στήλες του φαίνονται 4 χαρακτηριστικά: τα αθροιστικά και καθημερινά κρούσματα και θάνατοι της Κίνας σε συνδυασμό με τις ημερομηνίες στις γραμμές. Ωστόσο, τα δεδομένα αυτά όπως τονίσαμε και στην παράγραφο 4.1.1, προκειμένου να μπορούν να αναλυθούν και να επεξεργαστούν από τα μοντέλα των νευρωνικών δικτύων θα πρέπει να έχουν τουλάχιστον 3-διάστατη μορφή.
Προκειμένου να το πετύχουμε αυτό, στην ανάλυση και πρόβλεψη χρονοσειρών εφαρμόζουμε την *Μέθοδο Παραθυροποίησης*. Σε αυτήν, όπως αναφέρει και το όνομά της, επιλέγουμε ένα μήκος παραθύρου, έστω l_w, το οποίο καθορίζει το μήκος πληροφορίας με το οποίο εκπαιδεύεται το δίκτυό μας πάνω στην εκάστοτε χρονοσειρά (πίνακα). Το/τα στοιχείο/α της θέσης l_w+1 (κατά μήκος των γραμμών του πίνακα) αποτελεί *την ετικέτα* (label) της εκπαίδευσης με βάση την οποία καθορίζεται και η συνάρτηση απωλειών. Στη συνέχεια, κυλίουμε το παράθυρο κατά μια γραμμή κάτω και έτσι τα l_w στοιχεία του πίνακα από την δεύτερη έως και την $(l_w + 1)$-οστή θέση αποτελούν τα νέα δεδομένα εκπαίδευσης του δικτύου. Αυτήν την φορά η ετικέτα του πίνακα είναι το/τα στοιχείο/α

της θέσης $l_w + 2$. Αυτή η διαδικασία συνεχίζεται έως ότου εξαντληθεί το μήκος της χρονοσειράς (δηλαδή το πλήθος των γραμμών του πίνακα). Κατά αυτόν τον τρόπο στο τέλος της διαδικασίας θα έχουν παραχθεί $n-l_w$ στοιχεία πληροφορίας, όπου n το μήκος της χρονοσειράς (του αρχικού πίνακα). Κάθε τέτοιο στοιχείο πληροφορίας θα είναι ένα $2D$ πίνακας με μήκος l_w και πλάτος τον αριθμό χαρακτηρηστικών που έχουμε επιλέξει (έστω m). Αντίστοιχα, θα έχουν δημιουργηθεί και $n - l_w$ ετικέτες.
Στο σημείο αυτό είναι εφικτό να σχηματίσουμε έναν τρισδιάστατο τανυστή που θα εμπεριέχει αυτά τα $n-l_w$ στοιχεία πληροφορίας και θα έχει διάσταση $(n-l_w,l_w,m)$. Το μήκος l_w είναι παράμετρος που επίλεγεται από το χρήστη αναλόγως το πείραμα. Στην περίπτωση των δεδομένων της πανδημίας του κορωνοϊού τόσο εμπειρικά από τα αποτελέσματα όσο και από τη γνώση μας για τον ϊό επιλέξαμε μήκη παραθύρου από 7 έως 21 (μέρες). Αυτό σημαίνει ότι τα δικτυά μας 'έβλεπαν' στο παρελθόν από 1 έως 3 βδομάδες για να κάνουν τις προβλέψεις τους. Σε αρκετές περιπτώσεις το μήκος παραθύρου l_w αποκαλείται και *βήμα χρόνου* (time step). Στην εικόνα 4.6β΄ απεικονίζουμε τον ίδιο πίνακα με την εικόνα 4.6α΄, ωστόσο πάνω σε αυτήν τώρα φαίνεται γραφικά η μέθοδος της παραθυροποίησης.

	Ημερομηνία	Νέα Κρούσματα	Αθροιστικά Κρούσματα	Νέοι Θάνατοι	Αθροιστικοί Θάνατοι
0	2020-01-03	0.0	0.0	0.0	0.0
1	2020-01-04	1.0	1.0	0.0	0.0
2	2020-01-05	0.0	1.0	0.0	0.0
3	2020-01-06	3.0	4.0	0.0	0.0
4	2020-01-07	0.0	4.0	0.0	0.0
5	2020-01-08	0.0	4.0	0.0	0.0
6	2020-01-09	0.0	4.0	0.0	0.0
7	2020-01-10	0.0	4.0	0.0	0.0
8	2020-01-11	41.0	45.0	1.0	1.0
9	2020-01-12	0.0	45.0	0.0	1.0
10	2020-01-13	0.0	45.0	0.0	1.0
11	2020-01-14	0.0	45.0	0.0	1.0
12	2020-01-15	0.0	45.0	0.0	1.0
13	2020-01-16	0.0	45.0	0.0	1.0
14	2020-01-17	4.0	49.0	1.0	2.0
15	2020-01-18	17.0	66.0	0.0	2.0
16	2020-01-19	59.0	125.0	1.0	3.0
17	2020-01-20	77.0	202.0	1.0	4.0
18	2020-01-21	93.0	295.0	2.0	6.0

(α΄) Τυπικά αρχικά δεδομένα

(β΄) Αναπαράσταση μεθόδου παραθυροποίησης

Εικόνα 4.6: Αναπαράσταση (τυπικών) αρχικών δεδομένων και μεθόδου παραθυροποίησης

Διαδικασία Κανονικοποίησης

Στο Κεφάλαιο 3 σχεδόν σε όλες μας τις αναφορές στα επαναληπτικών νευρωνικών δικτύων χρησιμοποιούνται συναρτήσεις *sigmoid* και *tanh* οι οποίες περιορίζουν τις εξόδους τους στο $(0, 1)$ και στο $(-1, 1)$ αντίστοιχα. Από την άλλη πλευρά, τα δεδομένα μας έχουν τιμές τάξεως από μερικές εκατοντάδες έως ακόμα και 10^8. Αυτή η πολύ μεγάλη διαφορά τάξεως στα μέγεθη είναι πολύ πιθανό να προκαλέσει σημαντικά προβλήματα κατά την εκπαίδευση αλλά και την προβλέψη των δικτύων. Μερικά από τα προβλήματα αυτά είναι η αργή και ασταθής εκπαίδευση του μοντέλου, η απόκλιση κατά την επιτέλεση του αλγορίθμου εκπαίδευσης ή ακόμα και το πρόβλημα των εκρηγνυόμενων παραγώγων (κλίσεων) που έχουμε ήδη συναντήσει.

Για τους λόγους αυτούς, καθίσταται απαραίτητη η κανονικοποίηση του συνόλου δεδομένων σε ένα διάστημα όπου θα είναι 'διαχειρίσιμα' από τα νευρωνικά μας δίκτυα. Οι καλύτερες και συνηθέστερες επιλογές γενικά σε προβλήματα παλινδρόμησης είναι τα διαστήματα $(0, 1)$ και $(-1, 1)$. Στην εργασία επιλέχθηκε το δεύτερο έπειτα από πειραματισμούς.

Προκειμένου να επιτευχθεί αυτό έγινε χρήση έτοιμων βιβλιοθηκών της Python και συγκεκριμένα της Scikit-Learn [101] που παρέχει διαφόρων ειδών κανονικοποιητών όπως StandardScaler και MinMaxScaler. Μέσα από επίσης πειραματική διαδικασία επιλέχθηκε το δεύτερο είδος scaler. Η πράξη κανονικοποίησης που επιτελεί ο MinMaxScaler φαίνεται στην παρακάτω εξίσωση:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{4.1.2}$$

όπου X είναι το αρχικό διάνυσμα δεδομένων, έστω διάστασης $1 \times n$ με n το πλήθος παρατηρήσεων και X_{max}, X_{min} η μέγιστη και η ελάχιστη παρατήρηση στο διάνυσμα X αντίστοιχα. Κατ' επέκταση η μεταβλήτη X_{scaled} θα είναι και αυτή διάνυσμα διάστασης $1 \times n$. Αξίζει να τονιστεί ότι κανονικοποίηση υπέστησαν τόσο τα δεδομένα εκπαίδευσης όσο και οι αντίστοιχες ετικέτες τους διευκολύνοντας έτσι την εκπαίδευση.

Τελική μορφή των δεδομένων

Πρώτου τα δεδομένα, ως είσοδοι, εισέλθουν στο εκάστοτε μοντέλο προκειμένου αυτό να εκπαιδευθεί, χρείαστηκε ο διαμοιρασμός τους σε δεδομένα εκπαίδευσης (training data) και δεδομένα αξιολόγησης (test data). Τα πρώτα όπως φανερώνει και το ονομά τους χρησιμοποιούνται για την εκπαίδευση του μοντέλου (X_{train}) με βάση κάποιο σύνολο ετικετών (y_{train}), ενώ τα δεύτερα (X_{test}) είναι αυτά στα οποία το μοντέλο αξιολογείται χωρίς να τα έχει δει καθόλου κατά την εκπαίδευση πριν (άγνωστα δεδομένα). Κατά την διαδικασία αξιολόγησης δεν υπάρχουν ετικέτες, ωστόσο χρησιμοποιούνται μετέπειτα για την τελική μέτρηση αποτελεσματικότητας του δικτύου (y_{test}). Στα περισσότερα προβλήματα μηχανικής μάθησης τα δεδομένα εκπαίδευσης είναι αρκετά μεγαλύτερα από αυτά της αξιολόγησης σε ποσοστό 70% έως 90% και 30% έως 10% αντίστοιχα προκειμένου να δώσουμε την ευκαρία στα μοντέλα μας να εκπαιδευτούν καλύτερα και να κάνουν ακριβέστερες προβλέψεις. Στην παρούσα εργασία λόγω της μικρής διαθεσιμότητας δεδομένων[3] αλλά και λόγω της φύσεως του προβλήματος, ως πρόβλημα παλινδρόμησης σε

[3] η νόσος Covid-19 είναι ένα φαινόμενο που παρατηρείται μόνο 1 έτος περίπου

χρονοσειρές, επιλέχθηκε ο διαμοιρασμός σε ποσοστά 90%-10%.
Οπότε για συνολικό μήκος δεδομένων l_{data} τα μήκη των X_{train} και y_{train}, δίνονται από τις παρακάτω εξισώσεις:

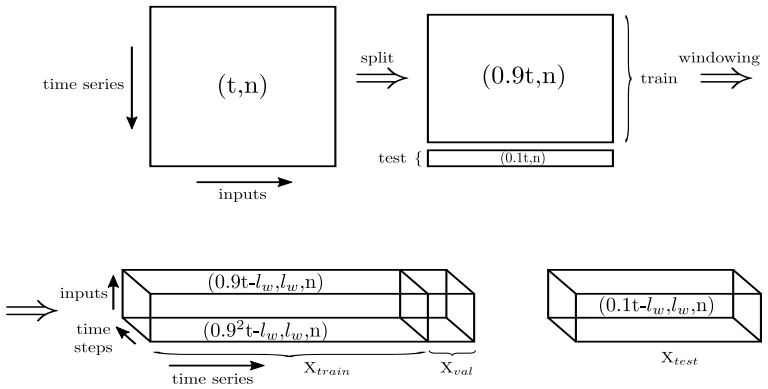
$$l_{Xtrain} = l_{ytrain} = 0.9 \times l_{data} - \text{time step}. \tag{4.1.3}$$

Επίσης στη σχέση (4.1.3) αν n_i ο αριθμός των χαρακτηριστικών που χρησιμοποιούμε, είναι $X_{train} \in \mathbb{R}^{l_{Xtrain} \times \text{time step} \times n_i}$ ενώ $y_{train} \in \mathbb{R}^{l_{ytrain} \times 1}$.
Για τα δεδόμενα αξιολόγησης θα έχουμε:

$$l_{Xtest} = 0.1 \times l_{data} + \text{time step} \tag{4.1.4}$$

$$l_{ytest} = 0.1 \times l_{data} \tag{4.1.5}$$

Στις (4.1.4) και (4.1.5) είναι $X_{test} \in \mathbb{R}^{l_{Xtest} \times \text{time step} \times n_i}$ ενώ $y_{train} \in \mathbb{R}^{l_{ytest} \times 1}$. Τέλος κατά την εκπαίδευση των δικτύων μας, τις περισσότερες φορές, είναι χρήσιμο να επαληθεύουμε την εκπαίδευση των μοντέλων σε άγνωστα δεδομένα που προέρχονται όμως από το σύνολο εκπαίδευσης. Τα δεδομένα αυτά τα ονομάζουμε δεδομένα επαλήθευσης (validation data ή $X_{validation}$). Αντίστοιχα, οι ετικέτες κατά τη διαδικασία επαλήθευσης ονομάζονται $y_{validation}$. Συνήθως, αυτή η διαδικασία βοηθά τα δίκτυα να πετυχαίνουν καλύτερη γενίκευση καθώς 'βλέπουν' άγνωστα δεδομένα κατά την εκπαίδευσή τους. Στην εργασία έπειτα από πειραματική διερεύνηση βρέθηκε ότι η χρήση δεδομένων επαλήθευσης θα πρέπει να είναι σχετικά μικρή σε σχέση με το πλήθος των δεδομένων εκπαίδευσης: $l_{Xvalidation} = 0.05 \times l_{Xtrain}$ και σε ορισμένες περιπτώσεις $l_{Xvalidation} = 0.1 \times l_{Xtrain}$. Δηλαδή το 5% ή το 10% των δεδομένων εκπαίδευσης χρησιμοποιήθηκαν ως δεδομένα επαλήθευσης.
Ανακεφαλαιώνοντας, παρουσιάζουμε τη μορφοποίηση των δεδομένων από τη συλλογή τους μέχρι και την εξαγωγή των τελικών τανυστών X_{train}, y_{train}, $X_{validation}$, $y_{validation}$, X_{test} και y_{test} στην εικόνα 4.7.



Εικόνα 4.7: Σχηματική αναπαράσταση των δεδομένων στους ζητούμενους tensors

4.2 Τεχνικές Τροποποίησης των Δεδομένων

Σε αυτήν την ενότητα θα γνωρίσουμε διάφορες τεχνικές που είναι γνωστές κατά την πρόβλεψη χρονοσειρών και υλοποιήθηκαν κατά την διάρκεια της εργασίας προκειμένου (ενδεχομένως) να βελτιώσουν την αποτελεσματικότητα και την ακρίβεια των μοντέλων μας κατά την πρόβλεψη. Ανάμεσα σε αυτές τις τεχνικές είναι η γνωστή *μέθοδος περιοδικών διαφορών* (Differencing) [102][103] αλλά και η μέθοδος της *επισήμανσης* (labelling) που προτείναμε κατά τη διάρκεια της εργασίας.

4.2.1 Μέθοδος επισήμανσης

Μια από τις μεγαλύτερες δυσκολίες στην πρόβλεψη χρονοσειρών είναι όταν αυτές παρουσιάζουν ακραίες τιμές (outliers), δηλαδή πολύ απότομες αυξήσεις ή μειώσεις ανάμεσα σε 2 παρατηρήσεις. Για το λόγο αυτό προτείνουμε την μέθοδο της επισήμανσης.
Σύμφωνα με αυτή τη μέθοδο, όταν τα δεδομένα έχουν υποστεί την προεξεργασία της κανονικοποίησης, σε αυτά προσθέτουμε ένα επιπλέον χαρακτηριστικό (στήλη) σε μορφή αληθούς-ψευδούς (0 ή 1) που δηλώνει αν η χρονοσειρά στο τρέχον βήμα t έχει μεγαλύτερη απόκλιση (σε απόλυτη τιμή) κατά μια τιμή $thres$ από την παρατήρησή μας σε κάποιο προηγούμενο χρονικό βήμα $t-k$. Αν $k = 1$ τότε προφανώς η σύγκριση γίνεται σε σχέση με την ακριβώς προηγούμενη παρατήρηση. Οπότε, αν θεωρήσουμε ότι η παρατήρηση τη χρονική στιγμή t είναι $x(t)$, η συνάρτηση επίσημανσης F_{label} μπορεί μαθηματικώς να διατυπωθεί ως:

$$F_{label}(t) = \begin{cases} 1, & |x(t) - x(t-k)| \geq thres \\ 0, & |x(t) - x(t-k)| < thres \end{cases} \tag{4.2.1}$$

για κάθε $t \geq k$ ενώ $F_{label}(t) = 0$ για κάθε $t < k$. Στην εικόνα 4.8 παρουσιάζουμε ένα τυπικό δείγμα της μεθόδου της επισήμανσης για τα δεδομένα της Ελλάδας για επιλογή $k = 1$ και $thres = 500$ για μια περίοδο 2 εβδομάδων. Έτσι με αυτήν την μέθοδο δίνουμε την ευκαιρία στα μοντέλα μας να συγκρατήσουν καλύτερα τις απότομες διακυμάνσεις στα δεδομένα και να αποδώσουν καλύτερα κατά τη διαδικασία αξιολόγησης.

4.2.2 Μέθοδος περιοδικών διαφορών

Σε αυτήν την παράγραφο θα δούμε μια αρκετά δημοφιλή τεχνική στην προβλέψη 'ανώμαλων' χρονοσειρών που εμφανίζουν κάποιου είδους περιοδικότητα και έχουν μία ενδεδειγμένη τάση. Σχεδόν όλον το πρώτο χρόνο της πανδημίας (Ιανουαριος 2020 - Ιανουαρίος 2021) σε παγκόσμιο επίπεδο η πανδημία παρουσίαζε μια αυξητική τάση και μια μερική περιοδικότητα γύρω στις 7 ημέρες, όπως φαίνεται και στην εικόνα 4.9α'. Η περιοδικότητα αυτή ενδεχομένως να οφείλεται στο γεγονός ότι τα περισσότερα κράτη του κόσμου συνηθίζουν να διεξάγουν μεγαλύτερο αριθμό τεστ τις καθημερινές και μικρότερο αριθμό τα σαββατοκύριακα.

Κατ' επέκταση και ο αριθμός κρουσμάτων φαίνεται να ακολουθεί την ίδια τάση. Για το λόγο αυτό, έχει παρατηρηθεί ότι σε χρονοσειρές που παρουσίαζουν τέτοιου είδους τάσεις μια μέθοδος που βοηθά ενδεχομένως στην εκπαίδευση και την πρόβλεψη είναι η *μέθοδος των περιοδικών διαφορών*. Σύμφωνα με αυτή τη μέθοδο αν η υπό εξέταση χρονοσειρά παρουσιάζει μια περίοδο T και με $X_{ob} \in \mathbb{R}^{n \times 1}$ συμβολίζουμε το διάνυσμα

Ημερομηνία	Νέα Κρούσματα	Δείκτης Επισήμανσης
2020-11-01	2055.0	0.0
2020-11-02	1678.0	0.0
2020-11-03	1151.0	1.0
2020-11-04	2166.0	1.0
2020-11-05	2646.0	0.0
2020-11-06	2915.0	0.0
2020-11-07	2447.0	0.0
2020-11-08	2555.0	0.0
2020-11-09	1889.0	1.0
2020-11-10	1489.0	0.0
2020-11-11	2383.0	1.0
2020-11-12	2751.0	0.0
2020-11-13	3316.0	1.0
2020-11-14	3038.0	0.0

Εικόνα 4.8: Αναπαράσταση της τεχνικής labelling στα δεδομένα

των παρατηρήσεων όπου n το πλήθος τους, τότε μπορούμε να 'θυσιάσουμε' μία περίοδο από τις παρατηρήσεις[4] για να δημιουργήσουμε μια νέα χρονοσειρά $Z_{diff} \in \mathbb{R}^{(n-T)\times 1}$ ως εξής:

$$Z_{diff}(t) = X_{ob}(t) - X_{ob}(t-T), \quad t \geq T. \tag{4.2.2}$$

Από την εξίσωση (4.2.2) φαίνεται ότι η νέα παρατήρηση ορίζεται μόνο για $t \geq T$. Στη σύνεχεια, αξιοποιούμε την νέα αυτή χρονοσειρά $Z_{diff}(t)$ προκειμένου να εκπαιδεύσουμε το μοντέλο μας σε αυτά τα νέα δεδομένα (λ.χ τα δεδομένα εκπαίδευσης θα είναι $Z_{diff-train}$) αλλά και να το αξιολογήσουμε (λ.χ με κάποια δεδομένα $Z_{diff-test}$). Η πρόβλεψη που θα πάρουμε ωστόσο (έστω $Z_{diff-pred}(t)$), από το μοντέλο θα είναι σχετική με τα δεδομένα $Z_{diff}(t)$ όποτε χρειάζεται να επαναφέρουμε τις προβλέψεις μας στην αρχική τους μορφή. Αυτό επιτυγχάνεται με την παρακάτω εξίσωση:

$$Z_{inv}(t) = Z_{diff-pred}(t) + X_{ob}(t-T), \quad t \geq T. \tag{4.2.3}$$

Η εξίσωση (4.2.3) μας δείχνει ότι οι τελικές προβλέψεις, $Z_{inv}(t)$, μπορούν να μας δωθούν από αυτές που μας έχει δώσει το μόντελο, $Z_{diff-pred}(t)$, συν τις πραγματικές παρατηρήσεις μια χρονική περίοδο T νωρίτερα. Στην πραγματικότητα δηλαδή, βασιζόμαστε στις παρατήρησεις μας μια χρονική περίοδο πίσω προκείμενου να κάνουμε ορθότερες προβλέψεις.
Η μέθοδος αυτή έχει αποδειχθεί ότι μπορεί να αποδώσει αρκετά καλά αποτελέσματα σε περιπτώσεις όπου τα δεδομένα εμφανίζουν περιοδικότητα και δεν είναι στάσιμα (stationary data) [104][105]. Σε περιπτώσεις που τα δεδομένα ακόμα και μετά την εφαρμογή της σχέσεως (4.2.2) εμφανίζουν ισχυρή μη-στασιμότητα τότε μπορούμε να

[4] Όταν το μήκος αυτών είναι ικανοποιητικά μεγάλο

εφαρμόσουμε ξανά την εξίσωση (4.2.2) ως:

$$\begin{aligned} Z_{2-diff}(t) &= Z_{diff}(t) - Z_{diff}(t-T) \\ &= X_{ob}(t) - X_{ob}(t-T) - X_{ob}(t-T) + X_{ob}(t-2T) \\ &= X_{ob}(t) - 2X_{ob}(t-T) + X_{ob}(t-2T) \end{aligned} \tag{4.2.4}$$

Προφανώς σε αυτήν την περίπτωση θα χρειαστεί ένας επιπλέον αντίστροφος μετασχηματισμός της (4.2.3) για την επαναφορά των δεδομένων από $Z_{2-inv}(t)$ σε $Z_{inv}(t)$. Στην εικόνα 4.9α′ απεικονίζεται η καμπύλη των καθημερινών κρουσμάτων παγκοσμίως (X_{ob}). Φαίνεται ότι η καμπύλη παρουσιάζει σταδιακή αύξηση έως και τις 360 μέρες περίπου ενώ με κόκκινα βέλη δείχνουμε τη σχετική περιοδικότητα που εμφανίζει η χρονοσειρά η οποία προσεγγίζει αρκετά το $T = 7$. Συγκεκριμένα βλεπούμε ότι σε διάστημα περίπου 50 ημερών, από την 250ή έως και την 300ή ημέρα, σχηματίζονται χονδρικά 7 ισομήκη διαστήματα. Όπως εύκολα μπορεί να δει κανείς αυτό μπορεί να εφαρμοστεί και σε αλλά σημεία της καμπύλης.
Στην εικόνα 4.9β′ βλέπουμε την παραχθείσα καμπύλη, Z_{diff}, από την παραπάνω X_{ob} αν πάρουμε $T = 7$. Είναι φανερό ότι η καμπύλη πλέον είναι πιο στάσιμη και δεν εμφανίζει τόσο ξεκάθαρη περιοδικότητα, ωστόσο υπάρχουν και πάλι δύο ξεκάθαρες ακραίες τιμές (outliers).

4.3 Αρχιτεκτονικές Μοντέλων Χρονοσειρών

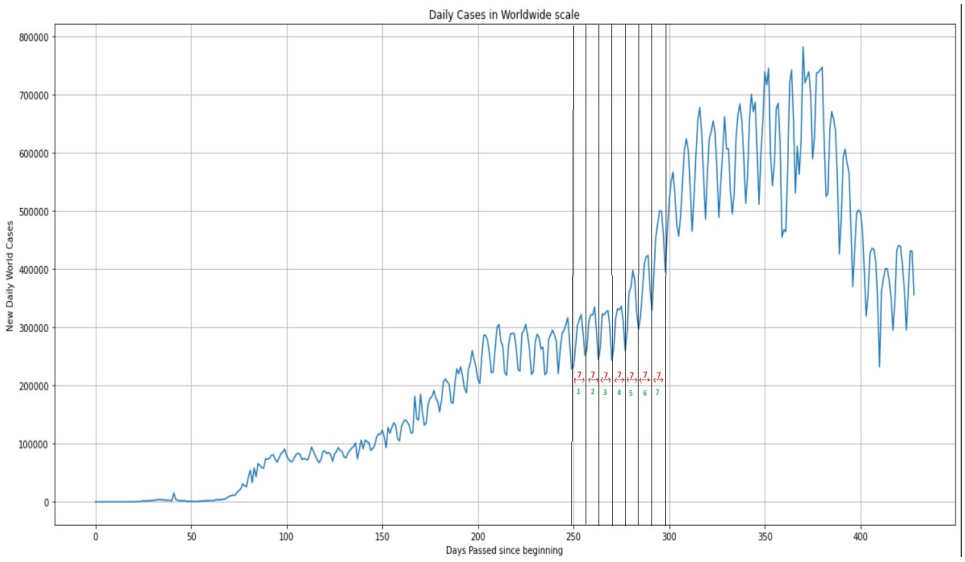
4.3.1 Επαναληπτικά Μοντέλα

Στην πρώτη προσπάθεια πρόβλεψης χρησιμοποιήσαμε Επαναληπτικά Νευρωνικά Δίκτυα όπως αυτά τα γνωρίσαμε στις ενότητες 3.2 και 3.3. Όπως ήδη γνωρίσαμε για την εκπαίδευση και την αξιολόγηση τους τα δεδομένα χωρίστηκαν στις τρεις υποομάδες εκπαίδευσης, επαλήθευσης και αξιολόγησης. Τα δεδομένα που χρησιμοποιήθηκαν από τον Παγκόσμιο Οργανισμό Υγείας αφορούσαν 429 ημέρες με το 90% να αποτελεί τα δεδομένα εκαίδευσης (386 ημέρες) και τα υπόλοιπα αφέθηκαν ως δεδομένα αξιολόγησης (43 ημέρες $\approx$ 6 εβδομάδες). Τα μήκη των X_{train}, y_{train}, $X_{validation}$, $y_{validation}$, X_{test} και y_{test} καθορίζονται από την επιλογή του μήκους παραθύρου l_w = time step όπως γνωρίσαμε στην παράγραφο 4.1.3 η οποία κυμάνθηκε από 7 έως 21 ημέρες. Για τα δεδομένα των ΗΠΑ μεμονωμένα, τα μήκη (σε ημέρες) ήταν 418, 376 και 42 των συνολικών δεδομένων, των δεδομένων εκπαίδευσης και των δεδομένων αξιολόγησης αντίστοιχα.

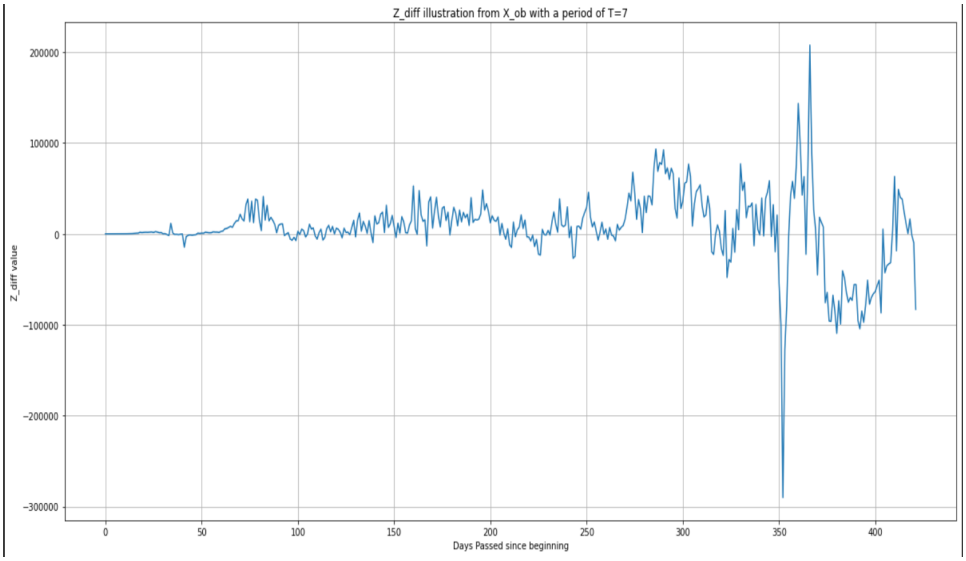
Απλά επαναληπτικά μοντέλα

Όλα τα είδη δικτύων που κατασκευάστηκαν κατά τη διάρκεια της εργασίας είχαν μόνον μια έξοδο[5] καθώς στόχος ήταν η πρόβλεψη μίας και μοναδικής τιμής εξόδου (κρουσμάτων ή θανάτων). Τα επαναληπτικά νευρωνικά δίκτυα έχουν πληθώρα *υπερπαραμέτρων* (hyperparameters), ιδιαίτερα τα πιο περίπλοκα GRUs και LSTMs. Κάποιες από αυτές είναι ο αριθμός των μονάδων (νευρώνων) στο δίκτυο (στρώμα), η συνάρτηση

[5]με εξαίρεση τα μοντέλα ανατροφοδότησης



(α′) Αρχική καμπύλη καθημερινών κρουσμάτων X_{ob}



(β′) Προκύπτουσα καμπύλη Z_{diff}

Εικόνα 4.9: Η μέθοδος περιοδικών διαφορών για τα παγκόσμια καθημερινά κρούσματα

ενεργοποίησης του δικτύου, η πιθανότητα dropout[6]. ο αριθμός των πλήρως διασυνδεδεμένων στρωμάτων στην έξοδο του δικτύου (Dense layers) κ.α.

Ως προς την δομή του νευρωνικού δικτύου υπάρχουν επιπλέον υπερπαράμετροι που καθορίζουν αν το επαναληπτικό μοντέλο θα επιστρέφει το σύνολο των κρυφών καταστάσεων, H, για κάθε χρονικό βήμα t (return sequences=True) ή αν θα επιστρέφει μόνο το διάνυσμα κρυφής κατάστασης του τελευταίου βήματος χρόνου (return sequences=False), αν θα είναι αμφίδρομο ή όχι αλλά και το πόσα στρώματα από το εκάστοτε δίκτυο χρησιμοποιούμε. Επιπρόσθετα, για τα δίκτυα LSTMs έχουμε μία ακόμα παράμετρο που μας δείχνει αν θέλουμε να επιστραφεί το τελευταίο κελί μνήμης C ή όχι (return state=True or False).

Αναφορικά και με την εκπαίδευση του μοντέλου όμως θα πρέπει να οριστεί το μέγεθος της δέσμης (batch size) με το οποίο θα τραφοδοτηθούν τα δεδομένα στο δίκτυο στη μορφή τρισδιάστατου τανυστή (batch size, timestep, features), αλλά και ο αριθμός των εποχών εκπαίδευσης (epochs). Τέλος, θα πρέπει να γίνει επιλογή αλγορίθμου βελτιστοποίησης (βλ. Παράρτημα Α′) και συνάρτησης κόστους.

Για την επιλογή των καταλληλότερων υπερπαραμέτρων έγινε εξαντλητική αναζήτηση (gridsearch) σε όλους τους συνδυασμούς των τιμών υπερπαραμέτρων που φαίνονται στους Πίνακες 4.1, 4.2, 4.3 για τα 3 είδη επαναληπτικών δικτύων (RNN, GRU, LSTM). Σε αυτούς, στην πρώτη στήλη τους βλέπουμε τα διάφορα είδη υπερπαραμέτρων που ελέχθησαν, στη δεύτερη τις διάφορες εναλλακτικές και στην τρίτη την βέλτιστη τιμή που πήραμε μετά την εφαρμογή του gridsearch.

Μοντέλο RNN		
Υπερπαράμετρος	*Συνδυασμός*	*Βέλτιστο*
Neurons	[16,32,64,128,256,512]	256
Dropout	[None,0.1,0.2]	0.2
Dense layers	[1,2,3]	1
Batch size	[8,16,32,48,64]	16 ή 32
Epochs	[50,100,150,200]	150
Optimizer	Adam, SGD	Adam
Loss Function	MSE, MAE	MSE
Activation Function	tanh, ReLu	tanh

Πίνακας 4.1: Υπερπαράμετροι μοντέλου απλού RNN

Αμφίδρομα και Βαθιά Επαναληπτικά Μοντέλα

Για τα αμφίδρομα επαναληπτικά μοντέλα η διαδικασία επιλογής υπερπαραμέτρων είναι πανομοιότυπη με τα παραπάνω μοντέλα με τη διάφορα ότι το δίκτυο που χρησιμοποιήσαμε είναι αμφίδρομο. Στον Πίνακα 4.4 βλέπουμε τις καλύτερες υπερπαραμέτρους που

[6]Ένα στρώμα Dropout θέτει τυχαία κάποιες εισόδους ίσες με το 0 κατά την εκπαίδευση του μοντέλου πράγμα που βοηθά στην αποφυγή της υπερπροσαρμογής (overfitting) με συχνότητα fr. Οι είσοδοι που δεν τίθονται μηδέν ,κανονικοποιούνται στη τιμή $1/(1-fr)$, ώστε το άθροισμα όλων των εισόδων να παραμείνει ανεπηρέαστο

Μοντέλο GRU		
Υπερπαράμετρος	*Συνδυασμός*	*Βέλτιστο*
Neurons	[16,32,64,128,256]	128
Dropout	[None,0.1,0.2]	0.1
Dense layers	[1,2,3]	1 ή 2
Batch size	[8,16,32,48,64]	16
Epochs	[50,100,150,200]	100
Optimizer	Adam, SGD	Adam
Loss Function	MSE, MAE	MSE
Activation Function	tanh, ReLu	tanh

Πίνακας 4.2: Υπερπαράμετροι μοντέλου GRU

Μοντέλο LSTM		
Υπερπαράμετρος	*Συνδυασμός*	*Βέλτιστο*
Neurons	[16,32,64,128,256]	128
Dropout	[None,0.1,0.2]	0.1
Dense layers	[1,2,3]	1 ή 2
Batch size	[8,16,32,48,64]	16 ή 32
Epochs	[50,100,150,200]	100
Optimizer	Adam, SGD	Adam
Loss Function	MSE, MAE	MSE
Activation Function	tanh, ReLu	tanh

Πίνακας 4.3: Υπερπαράμετροι μοντέλου LSTM

προέκυψαν τόσο για αμφίδρομα μοντέλα GRU όσο και LSTM. Σε αυτό το δίκτυο ο συνολικός αριθμός πολυπλοκότητας είναι όσος ο αριθμός των νευρωνών επί 2 ($\times 2$) λόγω της αμφίδρομης στρώσης.
Όσον αφορά τα βαθιά επαναληπτικά δίκτυα, τα οποία μπορεί να αποτελούνται είτε από απλές στρώσεις δικτύων είτε από αμφίδρομες, η γκάμα επιλογών των υπερπαραμέτρων που έχουμε παραμένει η ίδια με την προσθήκη του αριθμού των επιπέδων-στρώσεων από επαναληπτικά δίκτυα που χρησιμοποιήθηκαν.
Θα πρέπει επίσης να τονιστεί ότι στην περίπτωση αυτών των μοντέλων στις ενδιάμεσες στρώσεις η μεταβλητή return sequences τίθεται ίση με True προκειμένου να είναι εφικτή η είσοδος ενός $3D$ τανυστή στην επόμενη στρώση. Αντίθετα, στην τελευταία στρώση συνήθως αυτή η παράμετρος τίθεται ίση με False καθώς μας ενδιαφέρει η τελευταία κρυφή κατάσταση του δικτύου. Ακόμα, σε αυτού το είδους μοντέλα η επιβάρυνση του συστήματος πολλαπλασιάζεται με τις στρώσεις οπότε στα επιπλέον επίπεδα προσπαθήσαμε να κρατήσουμε την πολυπλοκότητα στο χαμηλότερο δυνατό βαθμό (μικρότερος αριθμός νευρώνων).
Στον Πίνακα 4.5 παρουσιάζουμε την επιλογή υπερπαραμέτων για τα βαθιά επαναληπτικά

δίκτυα που κατασκευάστηκαν στο πλαίσιο της εργασίας και απαρτίζονταν επίσης από μονάδες LSTM και GRU. Τέλος, στην εικόνα 4.10 βλέπουμε, τη δομή ενός βαθιού LSTM δικτύου που χρησιμοποιήθηκε. Παρατηρούμε ότι το μήκος ακολουθίας είναι 15 (ημέρες) και το πλήθος χαρακτηριστικών 1. Επίσης, βλέπουμε ότι τα μοντέλα LSTM ακολουθούνται από ένα πλήρως διασυνδεδεμένο στρώμα εξόδου ενώ η έξοδος του τελευταίου επίπεδου επαναληπτικού δικτύου μας δίνει έναν τανυστή διάστασης 2.

Bidirectional GRU και LSTM		
Υπερπαράμετρος	*Συνδυασμός*	*Βέλτιστο*
Neurons	[16,32,64,128,256]	64
Total Neurons	–	128
Dropout	[None,0.1,0.2]	0.1
Dense layers	[1,2,3]	1 ή 2
Batch size	[8,16,32,48,64]	16 ή 32
Epochs	[50,100,150,200]	100
Optimizer	Adam, SGD	Adam
Loss Function	MSE, MAE	MSE
Activation Function	tanh, ReLu	tanh

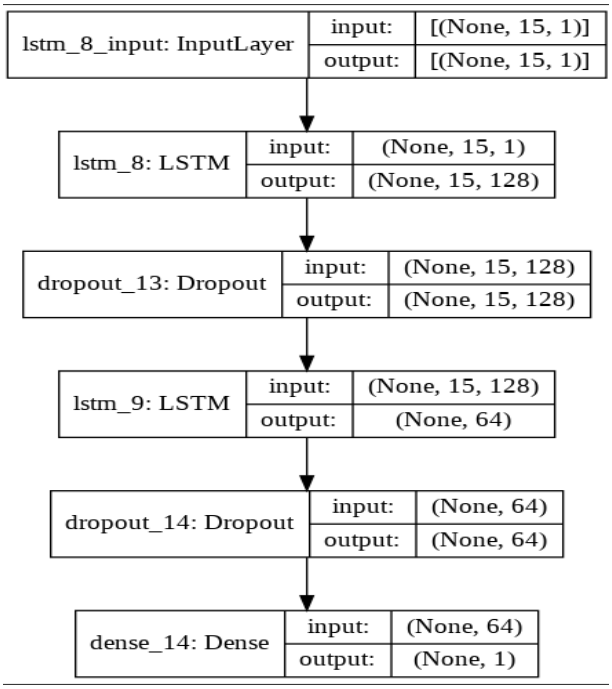
Πίνακας 4.4: Υπερπαράμετροι Αμφίδρομων μοντέλων LSTM και GRU

Βαθία Επαναληπτικά Μοντέλα GRU και LSTM		
Υπερπαράμετρος	*Συνδυασμός*	*Βέλτιστο*
Recurrent layers	[1,2,3]	2
Neurons $1^{ου}$ σταδίου	[16,32,64,128,256]	128
Neurons $2^{ου}$ σταδίου	[16,32,64,128,256]	64 ή 128
Dropout μετά από κάθε στρώση	[None,0.1,0.2]	0.1
Activation Function σε κάθε στρώση	tanh, ReLu	tanh
Dense layers	[1,2,3]	1 ή 2
Batch size	[8,16,32,48,64]	16
Epochs	[50,100,150,200]	150
Optimizer	Adam, SGD	Adam
Loss Function	MSE, MAE	MSE

Πίνακας 4.5: Υπερπαράμετροι Βαθιών Επαναληπτικών μοντέλων (LSTM,GRU)

4.3.2 Υβριδικά-Συνελικτικά Μοντέλα

Τα μοντέλα που απαρτίζονταν από συνδυασμό επαναληπτικών και συνελικτικών δικτύων ανήκουν στην κατηγορία που παρουσιάστηκε στην ενότητα 3.4. Σε αντίθεση με πριν, εδώ εκτός από τις παραμέτρους του επαναληπτικού μοντέλου θα πρέπει να διαλέξουμε



Εικόνα 4.10: Σχηματική απεικόνιση ενός βαθιού δικτύου LSTM αποτελούμενο από 2 στρώματα

και τις παραμέτρους του συνελικτικού. Συνεπώς, ακολουθήθηκε και εδώ η διαδιδικασία του gridsearch. Μερικές από τις υπερπαραμέτρους του συνελικτικού δικτύου είναι ο αριθμός των φίλτρων, το βήμα της συνέλιξης (stride), το μέγεθος πυρήνα, η χρήση padding ή όχι και φυσικά η συνάρτηση ενεργοποίησης του συνελικτικού στρώματος. Όσον αφορά τις παραμέτρους για Optimizer και Loss Function αυτές παρέμειναν ίδιες.

RNN-CNN Μοντέλο

Σε αυτό το είδος δικτύου, γνωρίσαμε στην παράγραφο 3.4.2 ότι συνήθως το συνελικτικό δίκτυο προηγείται του επαναληπτικού προκειμένου να υλοποιήσει την κατάλληλη εξαγωγή χωρικών χαρακτηριστικών των δεδομένων και να τροφοδοτήσει στο RNN μοντέλο. Αυτό είναι και το λογικό. Στα πλαίσια πειραματισμού ωστόσο, δοκιμάστηκε και η αντίθετη τεχνική. Δηλαδή, η τοποθέτηση του επαναληπτικού δικτύου στην αρχή και έπειτα του συνελικτικού. Κάτι τέτοιο όμως αποφανθήκαμε ότι, όχι μόνο δεν βοηθά, αλλά αποδίδει χειρότερα και συνέπως η τεχνική αυτή εγκαταλήφθηκε. Επίσης, συνολικά κατασκευάστηκαν 2 μοντέλα τύπου CNN-RNN με μερικές σημειακές διαφορές.
Η βασική διαφορά των 2 μοντέλων ήταν ότι στο δεύτερο δίκτυο έγινε χρήση και στρώματος συγκέντρωσης (Max pooling). Στον Πίνακα 4.6 παρουσιάζουμε τις υπερπαραμέτρους σε αυτά τα 2 μοντέλα διαχωριζόμενα από μία κάθετη γραμμή στη στήλη 'Βέλτιστο'.

CNN-RNN Model		
Υπερπαράμετρος	*Συνδυασμός*	*Βέλτιστο*
CNN Filters	[16,32,64,128]	64|64
Kernel Size	[2,3,4,6]	3|2
Stride	[1,2,3]	1|1
CNN activation	[relu,softmax]	relu|relu
Padding	[valid,same,casual]	same|valid
Pool Size	[2,3,4]	None|2
RNN Neurons	[16,32,64,128,256]	64|64
Dropout	[None,0.1,0.2]	0.1|0.1
Dense layers	[1,2,3]	1|1
Batch size	[8,16,32,48,64]	32|32
Epochs	[50,100,150,200]	50|50
Optimizer	Adam, SGD	Adam|Adam
Loss Function	MSE, MAE	MSE|MSE
RNN Activation	tanh, ReLu	tanh|tanh

Πίνακας 4.6: Υπερπαράμετροι μοντέλου CNN-RNN

Μοντέλο Conv-LSTM

Το μοντέλο Conv-LSTM παρέχεται και αυτό έτοιμο από τις βιβλιοθήκες του Keras. Συνδυάζει σε πολύ μεγάλο βαθμό τα χαρακτηριστικά και τις παραμέτρους των CNN και LSTM. Θα πρέπει να τονιστεί, ωστόσο, ότι η μονάδα Conv-LSTM σε αντίθεση τα απλά συνελικτικά στρώματα[7] που χρησιμοποιήθηκαν αλλά και τα επαναληπτικά στρώματα δέχεται ως εισόδους τανυστές 5 διαστάσεων (αντί για 3) ως (batch size, timestep, rows, columns, channels). Για το λόγο αυτό, στη χρήση αυτού του δικτύου κρίθηκε σκόπιμο να γίνει επιλογή παραθύρου (timestep) που να μην είναι πρώτος αριθμός προκειμένου στη συνέχεια να γίνει αναδιάταξη του τανυστή (Reshape). Στη θέση των columns είχαμε τα features ενώ για πλήθος καναλιών επιλέχθηκε το 1 [66].

Οπότε, αν για παράδειγμα στα προηγούμενα δίκτυα είχαμε έναν τρισδιάστατο τανυστή $X = (\text{batch size,timestep,features}) = (32, 15, 2)$ ως είσοδο (το 15 δεν είναι πρώτος), ο ίδιος tesnor τροποποιήθηκε προκειμένου να τροφοδοτηθεί στο Conv-LSTM ως: $X' = (\text{batch size, a, b, features, channels}) = (32, 3, 5, 2, 1)$. Προφανώς βλέπουμε ότι $a \times b = 3 \times 5 = \text{timestep} = 15$.

Τέλος, θα πρέπει να τονιστεί ότι και αυτή η μονάδα διαθέτει την μεταβλητή return sequences. Στην περίπτωση True η έξοδος θα είναι διάστασης 5 όπως και η είσοδος (batch size,a,b,channels,filters), ενώ σε περίπτωση False θα είναι δίαστασης 4 ως (batch size,b,channels,filters). Σε κάθε περίπτωση είναι αναγκαία η χρήση ενός στρώματος λείανσης (Flatten). Στον Πίνακα 4.7 παρουσιάζουμε τις υπερπαραμέτρους του μοντέλου αυτού που βρήκαμε ως βέλτιστες.

[7] Conv1D

Conv-LSTM Model		
Υπερπαράμετρος	*Συνδυασμός*	*Βέλτιστο*
Conv-LSTM Filters	[16,32,64,128]	64
Kernel Size	[(1,2),(2,1),(2,2)]	(1,2)
Stride	[1,2,3]	1
Conv-LSTM activation	[relu,tanh,softmax]	tanh
Pool Size	[2,3,4]	2
Dropout	[None,0.1,0.2]	0.1
Dense layers	[1,2,3]	1
Batch size	[8,16,32,48,64]	32
Epochs	[50,100,150,200]	150
Optimizer	Adam,SGD,Rmsprop	Adam
Loss Function	MSE,MAE	MSE

Πίνακας 4.7: Υπερπαράμετροι μοντέλου Conv-LSTM

Μοντέλο TCN

Το μοντέλο TCN είναι αισθητά διαφορετικό από τα μοντέλα που έχουμε δει, όπως γνωρίσαμε επίσης στην παράγραφο 3.4.2. Το μήκος της ακολουθίας εισόδου (l) μας καθορίζει κάποιες από τις παραμέρους του μοντέλου όπως το μέγεθος του πυρήνα της συνέλιξης και το πλήθος των διεσταλμένων συνελίξεων, σύμφωνα με τη σχέση (3.4.17). Όπως έχουμε τονίσει το μέγιστο μήκος ακολουθίας είναι 21 μέρες, συνεπώς θα πρέπει να διαλέξουμε βάση διαστολής b, πλήθος επιπέδων n και μέγεθος πυρήνα k ώστε:

$$1 + (k-1) \cdot \frac{b^n - 1}{b-1} \geq 21.$$

Δεδομένου ότι η βάση διαστολής είναι συνήθως $b = 2$ τότε θα πρέπει:

$$1 + (k-1) \cdot (2^n - 1) \geq 21.$$

Επομένως μας μένει να κάνουμε κατάταλλη επιλογή του n και του k. Στον Πίνακα 4.8 φαίνεται η επιλογή για το πλήθος των διαστολών (Dilations)[68].

Η παράμετρος n ισούται με το πλήθος των στοιχείων σε αυτό το διάνυσμα μείον 1. Για την ελάχιστη επιλογή $((1, 2, 4, 8))$ έχουμε $n = 3$ όποτε απαιτείται και $k - 1 = 3 \Rightarrow k = 4$ και τελικά:

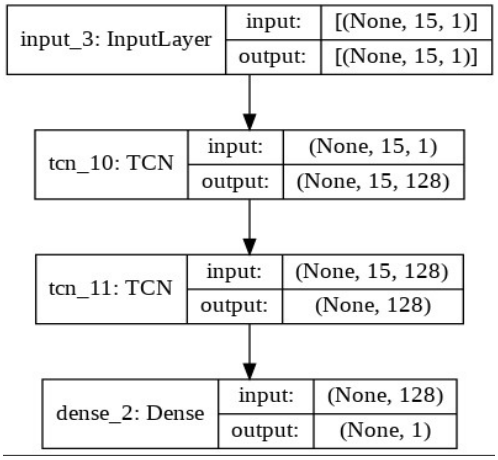
$$1 + (4-1) \cdot (2^3 - 1) \geq 21 \Leftrightarrow 22 \geq 21.$$

Προφανώς οι συνδυασμοί γίνεται να είναι περισσότεροι αρκεί να τηρείται αυτός ο περιορισμός. Ως προς τις υπόλοιπες παραμέτρους, το Skip Connections μας επιτρέπει να υλοποιήσουμε τις υπολειμματικές συνδέσεις, ενώ η μέθοδος Padding επιλέγεται casual με σκοπό να μην υπάρχει διαρροή πληροφορίας από το μέλλον. Οι υπόλοιπες παράμετροι είναι ίδιες με αυτές από τα απλά συνελικτικά δίκτυα. Τέλος, υπάρχει και εδώ η παράμετρος return sequences που στην περίπτωση ενδίαμεσων TCN επιπέδων τίθεται

TCN Model		
Υπερπαράμετρος	*Συνδυασμός*	*Βέλτιστο*
TCN Layers	[1,2,3,4]	2
TCN Filters	[16,32,64,128,256]	64
Kernel Size	[2,3,4,5]	4
Padding	[causal,same,valid]	causal
Skip Connections	[True,False]	True
Dilations	[(1,2,4,8), (1,2,4,8,16), (1,2,4,8,16,32)]	(1,2,4,8,16)
Activation function	[relu,tanh]	relu
Dense layers	[1,2,3]	1
Batch size	[8,16,32,48,64]	32
Epochs	[50,100,150,200]	150
Optimizer	Adam,SGD,Rmsprop	Rmsprop
Loss Function	MSE,MAE	MSE

Πίνακας 4.8: Υπερπαράμετροι μοντέλου TCN

True, ειδάλλως False. Το σύνολο των επιλογών φαίνεται στον Πίνακα 4.8 και η δομή ενός τέτοιου δικτύου στην εικόνα 4.11.



Εικόνα 4.11: Σχηματική απεικόνιση ενός δικτύου TCN αποτελούμενο από 2 στρώματα

4.3.3 Μοντέλα εφοδιασμένα με Attention

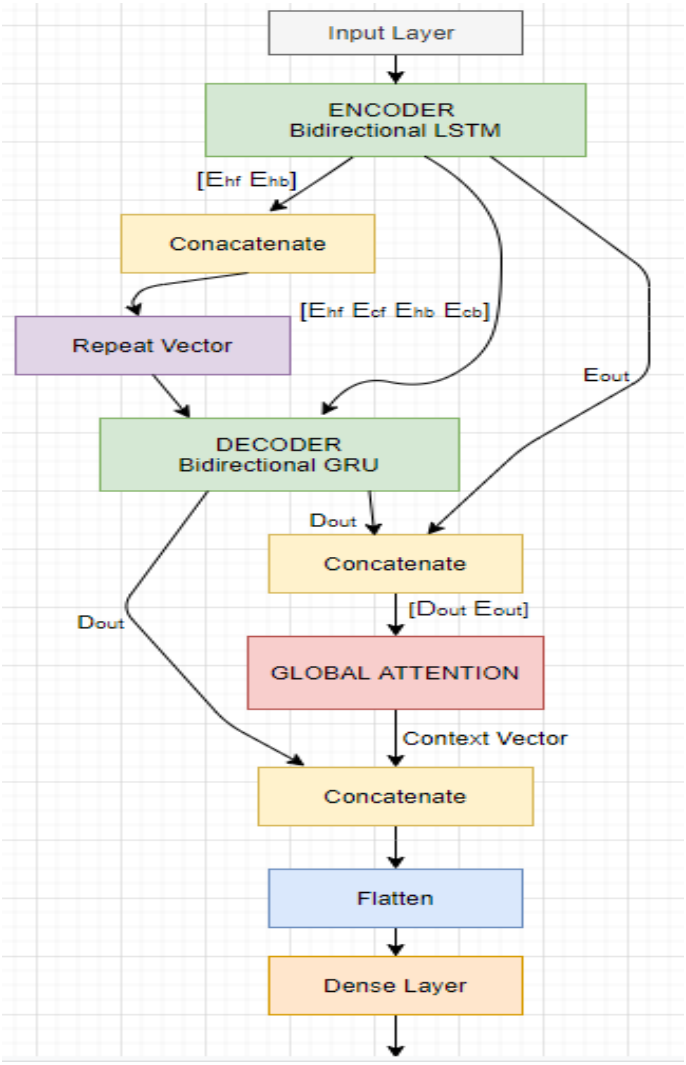
Τα μοντέλα που κατασκευάστηκαν στα πλαίσια της εργασίας και είχαν ενσωματωμένα τον μηχανισμό της προσοχής ήταν 2 ειδών. Σε γενικό πλαίσιο ο μηχανισμός αυτός ενσωματώθηκε στο εκάστοτε δίκτυο ως στρώμα όπως όλοι οι υπόλοιποι μηχανισμοί (RNNs,LSTMs, πλήρως συνδεδεμένα στρώματα κλπ). Ο διαχωρισμός που προαναφέρα-

με έγκειται στο είδος και στον τρόπο που εισήχθη αυτή η λειτουργία στο δίκτυο. Στην παράγραφο 3.5.2 αναφέραμε 3 από τα είδη Attention (Self,Global και Local Attention). Στην παρούσα διπλωματική κατασκευάστηκε μια κλάση from scratch (εξ' αρχής) που υλοποιούσε την τεχνική του Global Attention. Ταυτόχρονα, οι τεράστιες δυνατότητες του Keras προσφέρουν έτοιμες στρώσεις Προσοχής που μπορούν να υλοποιήσουν οποιοδήποτε είδος μέσα από παραμετροποίηση. Συγκεκριμένα, με αυτόν τον τρόπο υλοποιήσαμε τη λειτουργία του Local Attention.
Επίσης, σύμφωνα και πάλι με την παράγραφο 3.5.2 ο τρόπος που είσηχθη ο μηχανισμός αυτός στο δίκτυο είχε ως προαπαιτούμενο την ύπαρξη του μοντέλου Encoder-Decoder (όπως περίπου τον γνωρίσαμε στην παράγραφο 3.5.1). Ειδικότερα, χρησιμοποιήσαμε μια παραλλαγή αυτού του μοντέλου στο οποίο ενσωματώσαμε τον μηχανισμό της Προσοχής. Προσπαθήσαμε ώστε το είδος αυτό του Encoder-Decoder να προσεγγίζει αρκετά τα μοντέλα *από ακολουθία σε ακολουθία* (sequence to sequence - ευρέως διαδεδομένα στον τομέα Επεξεργασίας Φυσικής Γλώσσας). Στη θέση του Encoder είχαμε ένα Bidirectional LSTM (χρήση των h_t και C_t) ενώ σε αυτήν του Decoder ένα Bidirectional GRU.
Για τον αποκωδικοποιητή, η χρήση οποιουδήποτε επαναλήπτικού δικτύου είναι δυνατή. Στην εικόνα 4.12 φαίνεται η αρχιτεχτονική του μοντέλου Attention που προτάθηκε κατά τη διάρκεια της διπλωματικής [106][107]. Για την κατασκευή του βασιστήκαμε στο Functional API του Keras. Στη στρώση της Προσοχής μπορεί να υπονοείται είτε η αυτοδημιούργητη κλάση είτε κάποια έτοιμη από αντίστοιχη βιβλιοθήκη. Στην περίπτωση της δεύτερης ήταν δυνατή και η χρήση Local Attention. Και τα 2 είδη στρωμάτων λειτουργούν όπως λ.χ. τα επαναληπτικά στρώματα του Keras τα οποία δέχονται έναν τρισδιάστατο τανυστή ως είσοδο μεγέθους (batch size,time step,features) και δίνουν έναν τρισδιάστατο ή δισδιάστατο τανύστη στην έξοδο σύμφωνα με παραμετροποίηση.
Στην ίδια εικόνα βλέπουμε και άλλα στοιχεία όπως το Repeat Vector που μετατρέπει ένα δίανυσμα σε $3D$ πίνακα, αρκετές στρώσεις συνένωσης (Concatenation) αλλά και στρώμα λείανσης (Flatten). Ταυτόχρονα βλέπουμε ότι οι κρυφές καταστάσεις και τα κελιά μνήμης από τον κωδικοποιητή ($[E_{hf}; E_{cf}; E_{hb}; E_{cb}]$) χρησιμοποιούνται ως αρχικοποίηση μνήμης στον αποκωδικοποιητή, ενώ οι είσοδοι σε αυτόν, είναι οι τελευταίες κρυφές καταστάσεις του κωδικοποιητή ($[E_{hf}; E_{hb}]$). Τέλος, οι βασικές έξοδοι τόσο του Encoder όσο και του Decoder σε κάθε βήμα χρόνου συνενώνονται προκειμένου να περάσουν από το στρώμα Προσοχής. Πριν την έξοδο η συσχετισμένη πληροροφορία (Context Vector) από το στρώμα προσοχής συνενώνεται με την έξοδο του αποκωδικοποιητή για να περάσει από το πλήρες συνδεδεμένο στρώμα που θα ξεχωρίσει όλη τη σημαντική πληροφορία.
Στον Πίνακα 4.9 φαίνεται αναλυτικά το σύνολο των υπερπαραμέτρων τόσο για τον Encoder και τον Decoder όσο και για το δίκτυο συνολικά. Όπως έχει γίνει ήδη αντιληπτό, σε αυτό το μοντέλο χρειαζόμαστε την παράμετρο Return Sequences των Encoder και Decoder στο True. Για τον Encoder επιπλέον, χρειαζόμαστε και την παράμετρο Return State (C_t). Όσον αφορά το στρώμα Προσοχής αξίζει να σημειώσουμε ότι η μεταβλητή Attention Width δηλώνει το μήκος του παραθύρου που θέλουμε να 'εστιάσουμε' στην περίπτωση Local Attention. Μετά από πειραματισμό προέκυψε ότι χονδρικά είναι καλό να 'προσέξουμε' περίπου τόσα χρονικά βήματα πίσω όσο το time step των δεδομένων και για αυτό κατάληξαμε στο 5 έως 15 (εννοείται ημέρες).

Ακόμα, η παράμετρος Return Attention δηλώνει αν θα μας επιστραφούν και τα βάρη Προσοχής (α_t), πράγμα το οποίο δεν φάνηκε να βοηθά. Στην περίπτωση Global Attention ο αριθμός των εποχών εκπαίδευσης του δικτύου ήταν μεγαλύτερος, πράγμα λογικό καθώς χρειάζοταν παραπάνω εκπαίδευση για να ευθυγραμμιστεί σωστά η απαραίτητη πληροφορία καθ' όλο το μήκος των δεδομένων εισόδου.
Κατά τα άλλα η επιλογή των υπόλοιπων υπερπαραμέτρων παρέμεινε ίδια σε σχέση με τα προηγούμενα μοντέλα. Τέλος, θα πρέπει να τονιστεί ότι η χρήση στρωμάτων Dropout ήταν προαιρετική και ότι το συγκεκριμένο δίκτυο, όπως εύκολα αντιλαμβάνεται κανείς, είναι ιδιαίτερα πιο περίπλοκο από τα προηγούμενα.

Encoder-Decoder με Attention		
Υπερπαράμετρος	*Συνδυασμός*	*Βέλτιστο*
Είδος Encoder	[RNN,LSTM,GRU,TCN,Bi-models]	Bi-LSTM
Είδος Decoder	[RNN,LSTM,GRU,TCN,Bi-models]	Bi-GRU
Return State (Encoder)	[True,False]	True
Return Sequences (Encoder/Decoder)	[True,False/True,False]	True/True
Neurons Encoder	[64,128,256,512,1024]	512
Neurons Decoder	[64,128,256,512,1024]	512
Dropout μετά από Encoder και Decoder	[None,0.1,0.2]	None ή 0.1
Activation Functions	[tanh, ReLu, sigmoid]	tanh
Attention Layers	[self-made, Keras]	both
Attention Type	[Global, Local, Self]	Global,Local
Attention Width	[1,2,...,21]	5 έως 15
Return Attention	[True,False]	False
Dense layers	[1,2,3]	1 ή 2
Batch size	[8,16,32,48,64]	32
Epochs	[50,100,150,200]	100 ή 150
Optimizer	Adam, SGD	Adam
Loss Function	MSE, MAE	MSE

Πίνακας 4.9: Υπερπαράμετροι μοντέλου εφοδιασμένο με Attention τύπου Encoder-Decoder



Εικόνα 4.12: Προτεινόμενη αρχιτεκτονική μοντέλου Encoder-Decoder με μηχανισμό Προσοχής

Κεφάλαιο 5

Πειραματικά αποτελέσματα

5.1 Μετρικές Αξιολόγησης

5.1.1 Μετρικές Τετραγωνικού Σφάλματος

Στην εξίσωση (3.1.4) της παραγράφου 3.1.1 είδαμε τη συνάρτηση απωλειών τετραγωνικού σφάλματος. Παίρνοντας το μέσο όρο αυτής για n παρατηρήσεις $y_{obs}^{(i)}$ και αντίστοιχα για n προβλέψεις $y_{pred}^{(i)}$ έχουμε τη μετρική *μέσου τετραγωνικού σφάλματος* (Mean Squared Error - (MSE)) ως:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_{pred}^{(i)} - y_{obs}^{(i)}\right)^2 \tag{5.1.1}$$

Το *μέσο τετραγωνικό σφάλμα ρίζας* ορίζεται εύκολα ως εξής (Root Mean Squared Error - (RMSE)):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{pred}^{(i)} - y_{obs}^{(i)}\right)^2} \tag{5.1.2}$$

Ωστόσο, οι μετρικές των εξισώσεων (5.1.1),(5.1.2) είναι σε γενικές γραμμές αυθαίρετες και καθορίζονται σε πολλές περιπτώσεις ανάλογα με το πείραμα και τη μετρούμενη ποσότητα. Για το λόγο αυτό χρειαζόμαστε μια πιο αντικειμενική μετρική αξιολόγησης. Έτσι, διαιρώντας κάθε διαφορά μεταξύ της πραγματικής παρατήρησης και της πρόβλεψής μας με την αντίστοιχη παρατήρηση και πολλαπλασιάζοντας επί 100% λαμβάνουμε το *ποσοστιαίο μέσο τετραγωνικό σφάλμα ρίζας* (Root Mean Squared Percentage Error - (RMSPE)):

$$\boxed{RMSPE = \left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_{pred}^{(i)} - y_{obs}^{(i)}}{y_{obs}^{(i)}}\right)^2}\right) \times 100\%} \tag{5.1.3}$$

5.1.2 Μετρικές Απόλυτου Σφάλματος

Αντίστοιχα με το μέσο τετραγωνικό σφάλμα της εξίσωσης (5.1.1) ορίζουμε το *μέσο απόλυτο σφάλμα* (Mean Absolute Error - (MAE)) ως εξής:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_{pred}^{(i)} - y_{obs}^{(i)}\right| \tag{5.1.4}$$

Ωστόσο για τους ίδιους λόγους με παραπάνω θα πρέπει να ποσοστικοποιήσουμε την παραπάνω μετρική για να γίνεται πιο εύκολα αντιληπτή. Συνεπώς, ορίζουμε το *μέσο απόλυτο ποσοστιαίο σφάλμα* (Mean Absolute Percentage Error - (MAPE)) με τον εξής τρόπο:

$$\boxed{MAPE = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_{pred}^{(i)} - y_{obs}^{(i)}}{y_{obs}^{(i)}}\right|\right) \times 100\%} \tag{5.1.5}$$

5.1.3 Μετρικές Ακρίβειας

Συντελεστής Προσδιορισμού

Στη στατιστική ένα μέτρο για να αξιολογήσουμε την απόδοση των μοντέλων παλινδρόμησης (όπως το πρόβλημα που επιλύουμε) είναι ο *συντελεστής προσδιορισμού ή 'R-τετράγωνο' ή* R^2 (determination coefficient or "R-squared"). Ο συντελεστής αυτός μας δείχνει την αλληλοσυσχέτιση των προβλέψεων του μοντέλου παλινδρόμησης με την ανεξάρτητη μεταβλητή (παρατηρήσεις) σε κλίμακα του $0 - 100\%$. Περιγραφικά, μπορεί να δοθεί από την εξής σχέση:

$$R^2 = \frac{model's \quad variance}{Total \quad variance} \tag{5.1.6}$$

Πιο συγκεκριμένα, για ένα σύνολο n γνωστών παρατηρήσεων $\mathbf{y}_{obs} = \{y_{obs}^{(1)}, ..., y_{obs}^{(n)}\}$ και αντίστοιχων προβλεπόμενων τιμών $\mathbf{y}_{pred} = \{y_{pred}^{(1)}, ..., y_{pred}^{(n)}\}$ ορίζουμε τα μεμονωμένα κατάλοιπα σφάλματος ως $e_i = y_{obs}^{(i)} - y_{pred}^{(i)}$ (residuals) και τη μέση τιμή των παρατηρήσεων μας ως $y_{\bar{obs}} = \frac{1}{n}\sum_i y_{obs}^{(i)}$. Τότε η ολική διασπορά του συνόλου δεδομένων μας μπορεί να δοθεί από τη σχέση:

$$SS_{tot} = \sum_i \left(y_{obs}^{(i)} - y_{\bar{obs}}\right)^2 \tag{5.1.7}$$

Αντίστοιχα, η συνολική διασπορά τετραγωνικού σφάλματος μπορεί να δοθεί από την εξής σχέση:

$$SS_{res} = \sum_i \left(y_{obs}^{(i)} - y_{pred}^{(i)}\right)^2 = \sum_i {e_i}^2 \tag{5.1.8}$$

Τέλος, μπορούμε να ορίσουμε το συντελεστή προσδιορισμού R^2 ως:

$$\boxed{R^2 = 1 - \frac{SS_{res}}{SS_{tot}}} \tag{5.1.9}$$

Από τη σχέση (5.1.9) άμεσα συνάγεται ότι για $R^2 \to 1$ το μοντέλο έχει συσχετιστεί σε υψηλό βαθμό με τα δεδομένα παρατηρήσεων το οποίο σημαίνει ότι το μοντέλο έχει μεγάλη ακρίβεια στις προβλέψεις του, ενώ για $R^2 \to 0$ το μοντέλο δεν έχει μεγάλο βαθμό συσχέτισης με τις παρατηρήσεις και συνεπώς οι προβλέψεις του είναι χαμηλής ακρίβειας. Τέλος αν, $R^2 = 0.5$ τότε το μοντέλο θεωρητικά μπορεί να ερμηνεύσει σωστά την μισή από την παρατηρούμενη διασπορά [108].

Ποσοστά αποτελεσματικότητας

Αφαιρώντας τα ποσοστιαία σφάλματα $MAPE$ και $RMSPE$ από το 100 λαμβάνουμε τα *ποσοστά αποτελεσματικότητας* (Accuracy) βασιζόμενα σε απόλυτο και τετραγωνικό λάθος αντίστοιχα. Μαθηματικώς, διατυπώνονται με τις εξής εκφράσεις:

$$A_{ab} = \text{Absolute Accuracy} = 100\% - MAPE, \tag{5.1.10α′}$$
$$A_{sq} = \text{Squared Accuracy} = 100\% - RMSPE \tag{5.1.10β′}$$

5.2 Αποτελέσματα και αξιολόγηση, μοντέλων και τεχνικών

Σε αυτήν την ενότητα θα δούμε τα αποτελέσματα που μας έδωσαν τα μοντέλα που είδαμε στην ενότητα 4.3. Σε αρκέτες περιπτώσεις τα αποτελέσματα, λόγω της φύσεως του προβλήματος (πρόβλημα παλινδρόμησης σε χρονοσειρές) αλλά και εξαίτιας της χρήσης του Adam ως optimizer, που είναι ένας προσαρμοστικός βελτιστοποιητής με αποτέλεσμα να 'ταλανώνεται' γύρω από το σημείο ισορροπίας (βλ. Παράρτημα Α′), προέρχονται ως μέσος όρος πολυάριθμων (συνήθως 5) δοκιμών στις οποίες προστίθεται/αφαιρείται μία τυπική απόκλιση σύμφωνα με τον τύπο (5.2.1), στην οποία υποθέτουμε n μετρήσεις με μέση τιμή $\bar{x}$, (x_i η i-οστή μέτρηση) επομένως:

$$s_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{5.2.1}$$

5.2.1 Αξιολόγηση σε καμπύλη κρουσμάτων

Το πρώτο πείραμα που διεξήχθη ήταν η πρόβλεψη των καθημερινών κρουσμάτων παγκοσμίως με τέσσερις διαφορετικές τεχνικές. Χρησιμοποιώντας μόνο τα κρούσματα (univariate), χρησιμοποιώντας κρούσματα σε συνδυασμό με τα διεξαγώμενα τεστ (multivariate), χρησιμοποιώντας την μέθοδο της επισήμανσης (labelling) όταν τα κρούσματα παρουσιάζουν απότομη αύξηση/μείωση από μέρα σε μέρα (μεγαλύτερη από το 10% της μέγιστης τιμής) και χρησιμοποιώντας την μέθοδο των διαφορών (differencing).

Αξιολογήθηκαν όλα τα μοντέλα που παρουσιάσαμε στην ενότητα 4.3 ως προς τις μετρικές της προηγούμενης ενότητας και συγκεκριμένα ως προς τις μετρικές των σχέσεων (5.1.3), (5.1.5) και (5.1.9). Στους Πίνακες 5.1, 5.2, 5.3 και 5.4 φαίνονται αναλυτικά τα σκορ όλων των μοντέλων ως προς τις μετρικές RMSPE,MAPE και R^2.[1] Σε κάθε μέθοδο τα καλύτερα σκορ τονίζονται με έντονη γραφή (ως προς το μέσο όρο).
Στις εικόνες 5.1, 5.2, 5.3 και 5.4 μπορούμε να δούμε σε διαγράμματα μπάρας τα αποτελέσματα που παρουσιάζονται στους προαναφεθέντες Πίνακες. Στην εικόνα 5.5 μπορούμε να δούμε τις προβλέψεις όλων των μοντέλων επί των παγκόσμιων κρουσμάτων σε 9 υπο-σχήματα. Σε κάθε ένα από τα υπο-σχήματα φαίνεται από τη μορφή της καμπύλης (όπως είχαμε δει και στην εικόνα 4.9α΄) ότι αυτή είναι ισχυρώς μη στάσιμη με περίοδο $T = 7$. Με μπλε απεικονίζεται το διάστημα του όπου τα μοντέλα εκπαιδεύθηκαν, με πορτοκαλί το σύνολο αξιολόγησης (σταθερά σε κάθε υπο-σχήμα). Με πράσινο χρώμα απεικονίζεται το καλύτερο σκορ (ως προς το R^2) που πέτυχε το κάθε μοντέλο και από τις 4 διαφορετικές τεχνικές και έρχεται σε αντιδιαστολή με την καμπύλη αξιολόγησης (στους Πίνακες το αντίστοιχο σκορ υπογραμμίζεται). Τέλος, στην εικόνα 5.6 φέρνουμε σε άμεση αντιδιαστολή τα αποτελέσματα του καλύτερου μοντέλου (Attention με Differencing) με τα πραγματικά κρούσματα σε γράφημα μπάρας ενώ στον οριζόντιο άξονα τονίζεται η πραγματική ημερομηνία.

Univariate Results			
Μοντέλο	**RMSPE(%)**	**MAPE(%)**	**R^2** (0 έως 1)
TCN	**11.44 ±1.56**	**7.53 ±1.01**	**0.826 ±0.006**
Attention	11.89 ±1.17	8.04 ±0.88	0.816 ±0.005
S-LSTM	17.59 ±1.00	12.13 ±0.91	0.579 ±0.015
Bi-GRU	18.17 ±1.24	12.53 ±0.92	0.558 ±0.013
LSTM	14.64 ±0.89	11.8 ±0.62	0.659 ±0.011
GRU	15.1 ±1.11	12.58 ±0.71	0.604 ±0.015
RNN	12.3 ±0.85	9.58 ±0.64	0.776 ±0.009
CNN-RNN	14.3±1.66	10.92 ±0.9	0.719±0.01
Conv-LSTM	14.71 ±1.14	11.3 ±0.76	0.661 ±0.012

Πίνακας 5.1: Αξιολόγηση στην τεχνική Univariate-Παγκόσμια Κρούσματα

Σχολιασμός

Τόσο από τους πίνακες όσο και από τις εικόνες μπορούμε να δούμε ότι στην προσπάθεια πρόβλεψης με τη χρήση της τεχνικής Univariate το μοντέλο TCN κυριαρχεί των υπολοίπων και στις 3 μετρικές αξιολόγησης, με δεύτερο να έρχεται το Attention based μοντέλο. Προχωρώντας στις τεχνικές Labelling και Multivariate βλέπουμε ότι το μοντέλο εφοδιασμένο με το μηχανισμό της προσοχής καταφέρνει να ανταποκριθεί καλύτερα σε σχέση με τα υπόλοιπα. Πιο συγκεκριμένα, στη τεχνική της επισήμανσης ενώ κάποια

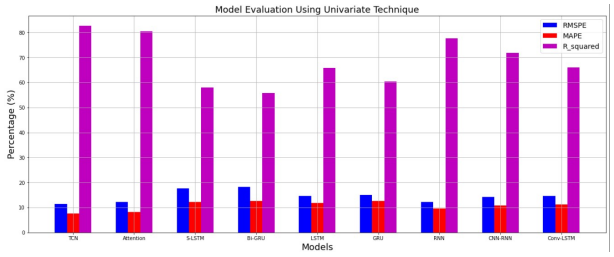
[1] Στο μοντέλο CNN-RNN η επισήμανση 'max' δηλώνει ότι έγινε χρήση του μοντέλου με στρώμα max pooling, όπως το $2^ο$ μοντέλο του Πίνακα 4.6. Σε αντίθετη περίπτωση υπονοείται χρήση του $1^{ου}$ μοντέλου

Labelling Results			
Μοντέλο	**RMSPE(%)**	**MAPE(%)**	**R^2** (0 έως 1)
TCN	14.28 ±1.1	9.72 ±0.78	0.7 ±0.007
Attention	**12.21 ±0.79**	**8.91 ±0.64**	**0.768 ±0.008**
S-LSTM	15.92 ±1.33	11.52 ±1.04	0.637 ±0.001
Bi-GRU	19.66 ±1.45	12.05 ±0.98	0.528 ±0.014
LSTM	13.25 ±1.11	10.31 ±0.72	0.709 ±0.013
GRU	12.51 ±0.91	9.97±0.67	0.734 ±0.005
RNN	12.92 ±0.88	10.73 ±0.61	0.689 ±0.011
CNN-RNN (max)	18.9 ±1.5	14.9 ±0.89	0.481 ±0.02
Conv-LSTM	14.05 ±1.09	9.48 ±0.73	0.732 ±0.013

Πίνακας 5.2: Αξιολόγηση στην τεχνική Labelling-Παγκόσμια Κρούσματα

Multivariate Results (with tests)			
Μοντέλο	**RMSPE(%)**	**MAPE(%)**	**R^2** (0 έως 1)
TCN	15.96 ±1.31	12.09 ±0.94	0.516 ±0.013
Attention	**12.19 ±0.68**	**9.36 ±0.57**	**0.722 ±0.006**
S-LSTM	20.55 ±0.87	15.13 ±0.72	0.372 ±0.021
Bi-GRU	21.88 ±1.44	16.15 ±0.99	0.344 ±0.019
LSTM	22.54 ±1.62	15.39 ±1.1	0.324 ±0.023
GRU	23.33 ±1.49	15.59 ±1.06	0.312 ±0.025
RNN	19.82 ±0.98	15.28 ±0.85	0.347±0.024
CNN-RNN	22.65 ±1.51	15.52 ±1.16	0.311 ±0.021
Conv-LSTM	19.35 ±0.97	14.97 ±0.7	0.428 ±0.015

Πίνακας 5.3: Αξιολόγηση στην τεχνική Multivariate-Παγκόσμια Κρούσματα

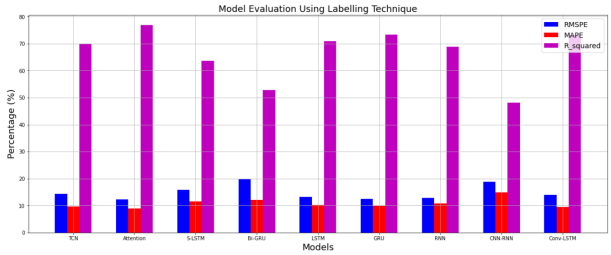


Εικόνα 5.1: Αξιολόγηση των μοντέλων στην τεχνική Univariate-Παγκόσμια Κρούσματα

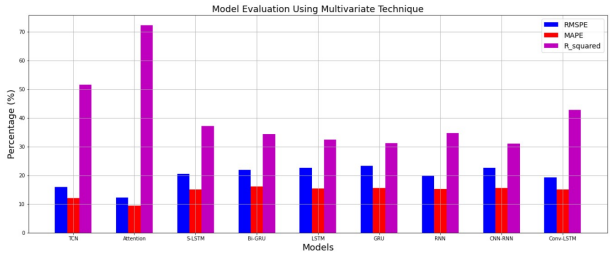
μοντέλα κατάφεραν να βελτιώσουν την απόδοσή τους σε σχέση με την μονομεταβλητή πρόβλεψη και το Attention μοντέλο έχασε στο σκορ του, αυτό παρέμεινε στην πρώτη θέση (και στις 3 μετρικές) ενώ στη δεύτερη ανέβηκε, το πιο περίπλοκο Conv-LSTM. Όσον αφορά την πολυμεταβλητή πρόβλεψη το μόνο μοντέλο που αποδίδει ικανοποιητικά ως προς τις 3 μετρικές είναι το μοντέλο με τον μηχανισμό της προσοχής. Αξίζει να τονιστεί ότι η εμπλοκή δεύτερης μεταβλητής δείχνει να μπέρδεψε τα μοντέλα αντί να τα βοηθήσει.

Differencing Results			
Μοντέλο	**RMSPE(%)**	**MAPE(%)**	**R^2** (0 έως 1)
TCN	9.61 ±1.04	7.2 ±0.67	0.816 ±0.007
Attention	**8.14 ±0.5**	5.66 ±0.21	**0.887 ±0.003**
S-LSTM	13.27 ±0.76	11.25±0.45	0.603 ±0.018
Bi-GRU	8.16 ±0.45	**5.48 ±0.25**	0.886 ±0.002
LSTM	12.31 ±0.82	10.52 ±0.51	0.668 ±0.013
GRU	14.33 ±0.88	12.07 ±0.5	0.532 ±0.018
RNN	10.45 ±0.61	8.15 ±0.38	0.759 ±0.008
CNN-RNN (max)	13.88 ±0.9	11.77 ±0.55	0.515 ±0.022
Conv-LSTM	12.72 ±0.85	10.65 ±0.53	0.699 ±0.011

Πίνακας 5.4: Αξιολόγηση στην τεχνική Differencing-Παγκόσμια Κρούσματα

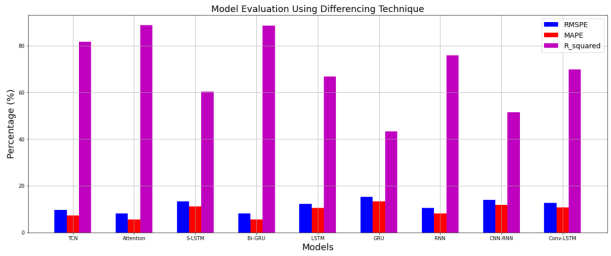


Εικόνα 5.2: Αξιολόγηση των μοντέλων στην τεχνική Labelling-Παγκόσμια Κρούσματα

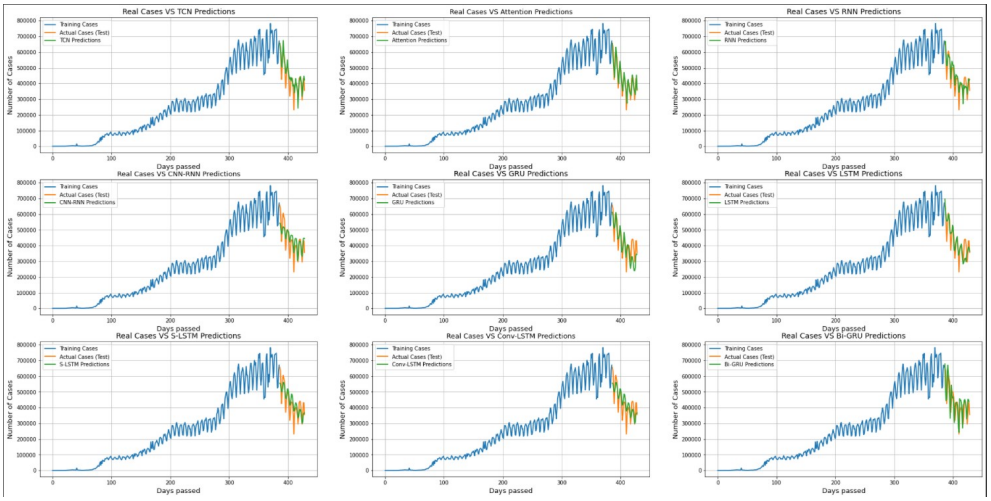


Εικόνα 5.3: Αξιολόγηση των μοντέλων στην τεχνική Multivariate-Παγκόσμια Κρούσματα

Αυτό ίσως συνέβη εξαιτίας του γεγονότος ότι τα δεδομένα για τον αριθμό των τεστ δεν προέρχονταν από όλες τις χώρες και στον σωστό ρυθμό, με συνέπεια τα μοντέλα να μην καταφέρουν να 'δουν' την άμεση εξάρτηση της μεταβλητής των κρουσμάτων από αυτή τη μεταβλητή. Ωστόσο, πείραματα με μεγαλύτερο αριθμό εποχών ενδεχομένως να βελτίωναν την απόδοσή τους. Έπειτα, στη μέθοδο των διαφορών φαίνεται το εξής ιδιαίτερο. Ενώ το μοντέλο Bidirectional GRU έδειξε στις 3 προηγούμενες τεχνικές ότι η χρήση αμφίδρομου επιπέδου νευρώνων δεν οφελεί το δίκτυο, σε αυτή την περίπτωση συμβαίνει το αντίθετο. Η 'διαφόριση' (με βήμα 7) της καμπύλης στη πραγματικότα μας δίνει την τάση του αριθμού των κρουσμάτων και το δίκτυο Bi-GRU δείχνει να αντιλαμ-



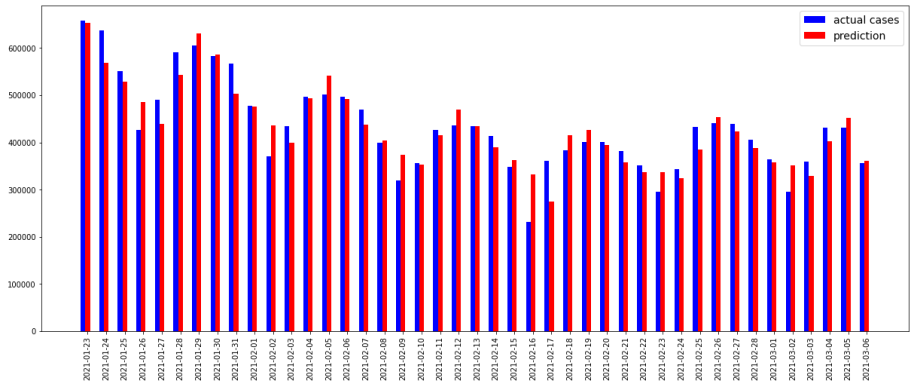
Εικόνα 5.4: Αξιολόγηση των μοντέλων στην τεχνική Differencing-Παγκόσμια Κρούσματα



Εικόνα 5.5: Γραφική αναπαράσταση των αποτελεσμάτων όλων των μοντέλων επί των παγκόσμιων κρουσμάτων

βάνεται πολύ καλά αυτή την τάση με αποτέλεσμα να υπερνικάει σε απόλυτο σφάλμα όλα τα υπόλοιπα μοντέλα, μένοντας οριακά δεύτερο πίσω από το Attention based μοντέλο στις μετρικές τετραγωνικού σφάλματους και συντελεστή προσδιορισμού.

Συνολικά, από όλες τις μετρήσεις φαίνεται ότι μοντέλο της προσοχής δείχνει την μεγαλύτερη αξιοπιστία σε αυτήν την καμπύλη. Ως προς τις τεχνικές και θεωρώντας ως βασική μας τεχνική την Univariate βλέπουμε ότι τα μοντέλα κατά βάση βοηθήθηκαν από τη μέθοδο Differencing μιας και όπως τονίσαμε η συγκεκριμένη καμπύλη είναι μη στάσιμη. Επίσης, σε κάποιες περιπτώσεις βλέπουμε βελτίωση και από τη μέθοδο Labelling. Τέλος, βλέπουμε ότι αναλογικά τα μοντέλα με συνελικτικά χαρακτηριστικά παρουσιάζουν ελαφρώς υψηλότερη διακύμανση, ενώ στη μέθοδο περιοδικών διαφορών η διακύμανση στα αποτελέσματα των μοντέλων μειώθηκε αισθητά.



Εικόνα 5.6: Σύγκριση των πραγματικών κρουσμάτων με το καλύτερο μοντέλο Attention με Differencing

5.2.2 Αξιολόγηση σε καμπύλη θανάτων

Σε αυτή την παράγραφο θα δούμε την αποτελεσματικότητα των μοντέλων σε μια καμπύλη θανάτων. Η καμπύλη αυτή αφορά τους θανάτους που συνέβησαν στην Ιταλία. Η επιλογή αυτή έγινε καθώς η συγκεκριμένη χώρα παρείχε σωστά και αξιόπιστα δεδομένα στις πηγές απ' όπου αντλήσαμε τα δεδομένα κυρίως ως προς τους ανθρώπους που νοσηλεύονται σε νοσοκομείο ή βρίσκονται σε ΜΕΘ. Και πάλι αξιολογήθηκαν οι 4 διαφορετικές τεχνικές με την μόνη διαφορά ότι στην τεχνική Multivariate χρησιμοποιήθηκαν ταυτόχρονα με τους ίδιους τους θανάτους, οι νοσηλείες σε νοσοκομείο (απλές κλίνες) και οι εισαγωγές σε μονάδες εντατικής θεραπείας. Από την άλλη πλευρά, αξιολογήθηκαν τα ίδια 9 μοντέλα και οι ίδιες 3 μετρικές.
Στους Πίνακες 5.5, 5.6, 5.7 και 5.8 παρουσιάζουμε τα αποτελέσματα του πειράματος (με έντονη γραφή φαίνονται τα καλύτερα σκορ ως προς τον μέσο όρο), ενώ στις εικόνες 5.7, 5.8, 5.9 και 5.10 μπορούμε να δούμε την ίδια πληροφορία σε διαγράμματα μπάρας.
Στην εικόνα 5.11 μπορούμε να δούμε τις προβλέψεις των μοντέλων επί της καμπύλης των θανάτων της Ιταλίας σε 9 υπο-σχήματα. Σε αντίθεση με πριν η καμπύλη αυτή είναι μερικώς μη στάσιμη αλλά και πάλι κατά την μέθοδο περιοδικών διαφορών χρησιμοποιήθηκε $T = 7$[2]. Με μπλε δείχνουμε το μέρος εκπαίδευσης και με πορτοκαλί το κομμάτι αξιολόγησης. Με πράσινο χρώμα απεικονίζεται η καλύτερη επίδοση (ως προς την μετρική R^2) από κάθε μοντέλο σε οποιαδήποτε από τις 4 τεχνικές και συγκρίνεται με το κομμάτι αξιολόγησης, την οποία υπογραμμίζουμε και στους Πίνακες.

Τέλος, στην εικόνα 5.12 φέρνουμε σε άμεση αντιδιαστολή τα αποτελέσματα του καλύτερου μοντέλου (TCN με Univariate) με τους πραγματικούς θανάτους σε γράφημα μπάρας ενώ στον οριζόντιο άξονα τονίζεται και εδώ η πραγματική ημερομηνία.

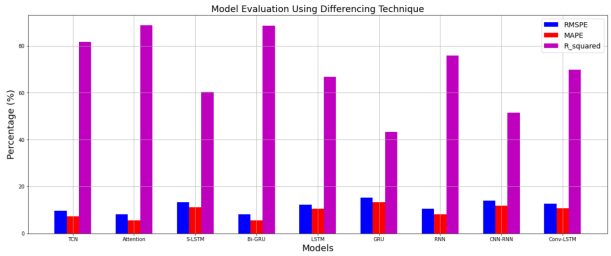
[2]η επιλογή του T, ιδιαίτερα σε περιπτώσεις όπου δεν είναι ξεκάθαρη, γίνεται με βάση το σημείο μεγιστοποίησης (ως προς τ) της συνάρτησης αυτοσυσχέτισης $E[X_t X_{t+\tau}]$ της δοθείσας χρονοσειράς X_t. Από τον υπολογισμό του μεγίστου εξαιρείται το σημείο 0 [126].

Univariate Results			
Μοντέλο	**RMSPE(%)**	**MAPE(%)**	**R^2** (0 έως 1)
TCN	12.13 ±0.76	**9.11 ±0.61**	**0.793 ±0.005**
Attention	15.18 ±1.05	10.74 ±0.8	0.735 ±0.007
S-LSTM	13.27 ±0.95	10.12 ±0.65	0.761 ±0.009
Bi-GRU	21.36 ±1.52	15.59 ±0.97	0.436 ±0.014
LSTM	**12.12 ±0.67**	9.26 ±0.56	0.757 ±0.006
GRU	13.33 ±0.77	10.34 ±0.55	0.754 ±0.006
RNN	13.65 ±0.82	10.61 ±0.6	0.752 ±0.007
CNN-RNN	16.88 ±1.11	13.25 ±0.89	0.656 ±0.01
Conv-LSTM	21.82 ±1.67	15.38 ±1.00	0.511 ±0.012

Πίνακας 5.5: Αξιολόγηση στην τεχνική Univariate-Θάνατοι Ιταλίας

Labelling Results			
Μοντέλο	**RMSPE(%)**	**MAPE(%)**	**R^2** (0 έως 1)
TCN	26.04 ±2.15	18.24 ±1.48	0.243 ±0.034
Attention	17.12 ±0.97	13.75 ±0.73	0.585 ±0.012
S-LSTM	16.52 ±0.92	12.18 ±0.72	0.659 ±0.011
Bi-GRU	23.75 ±1.88	16.5 ±1.27	0.397 ±0.024
LSTM	**13.65 ±0.9**	**10.75 ±0.64**	**0.742 ±0.007**
GRU	14.97 ±1.08	11.37 ±0.74	0.704 ±0.01
RNN	14.12 ±0.85	10.78 ±0.62	0.733 ±0.008
CNN-RNN (max)	24.78 ±1.98	16.76 ±1.4	0.276 ±0.029
Conv-LSTM	22.12 ±1.65	15.44 ±1.15	0.422 ±0.019

Πίνακας 5.6: Αξιολόγηση στην τεχνική Labelling-Θάνατοι Ιταλίας



Εικόνα 5.7: Αξιολόγηση των μοντέλων στην τεχνική Univariate-Θάνατοι Ιταλίας

Σχολιασμός

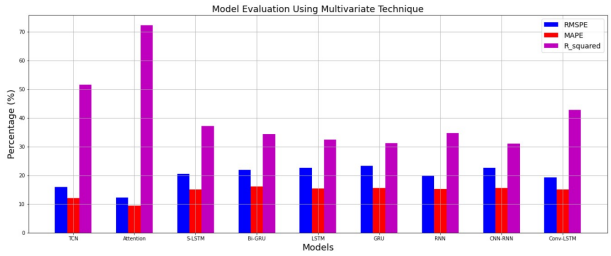
Τόσο από τους Πίνακες όσο και από τις εικόνες μπορούμε να δούμε ότι τα αποτελέσματα είναι ιδιαιτέρως πιο περίπλοκα από την περίπτωση των κρουσμάτων. Δεν φαίνεται να υπάρχει υπεροχή κάποιου μοντέλου επί των υπολοίπων καθώς τα σκορ των μοντέλων κυμαίνονται σε μικρότερο εύρος. Ο λόγος για αυτό, είναι το γεγονός ότι η συγκεκριμένη

Multivariate Results (with Hospitalized,ICU patients)			
Μοντέλο	**RMSPE(%)**	**MAPE(%)**	**R^2** (0 έως 1)
TCN	16.83 ±1.1	12.63 ±0.81	0.657 ±0.011
Attention	16.34 ±1.06	12.06 ±0.75	0.66 ±0.01
S-LSTM	12.69 ±0.91	9.98 ±0.55	0.751 ±0.007
Bi-GRU	19.6 ±1.2	14.47 ±0.88	0.553 ±0.016
LSTM	13.41 ±0.89	10.74 ±0.63	0.731 ±0.007
GRU	12.31 ±0.93	9.77 ±0.66	**0.774 ±0.006**
RNN	**12.27 ±0.84**	**9.7 ±0.6**	0.747 ±0.008
CNN-RNN	18.36 ±1.32	13.03 ±1.02	0.588 ±0.013
Conv-LSTM	16.96 ±1.18	13.14 ±0.81	0.583 ±0.015

Πίνακας 5.7: Αξιολόγηση στην τεχνική Multivariate-Θάνατοι Ιταλίας

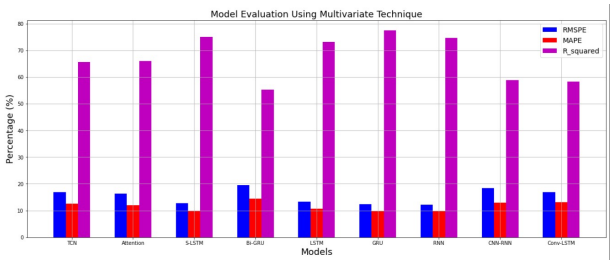
Differencing Results			
Μοντέλο	**RMSPE(%)**	**MAPE(%)**	**R^2** (0 έως 1)
TCN	15.12 ±1.07	12.22 ±0.74	0.685 ±0.011
Attention	14.32 ±0.71	11.75 ±0.49	0.712 ±0.009
S-LSTM	14.57 ±0.73	11.48 ±0.48	0.703 ±0.01
Bi-GRU	14.04 ±0.62	11.59 ±0.43	0.717 ±0.008
LSTM	14.51 ±0.78	11.89 ±0.5	0.697 ±0.009
GRU	14.42 ±0.69	11.79 ±0.52	0.699 ±0.009
RNN	14.76 ±0.71	12.00 ±0.55	0.687 ±0.01
CNN-RNN	**14.00 ±0.66**	11.28 ±0.42	0.713 ±0.008
Conv-LSTM	14.11 ±0.65	**11.04 ±0.4**	**0.719 ±0.007**

Πίνακας 5.8: Αξιολόγηση στην τεχνική Differencing-Θάνατοι Ιταλίας

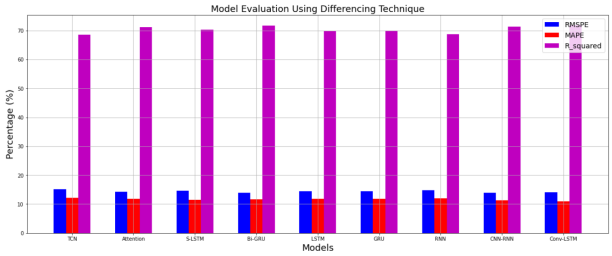


Εικόνα 5.8: Αξιολόγηση των μοντέλων στην τεχνική Labelling-Θάνατοι Ιταλίας

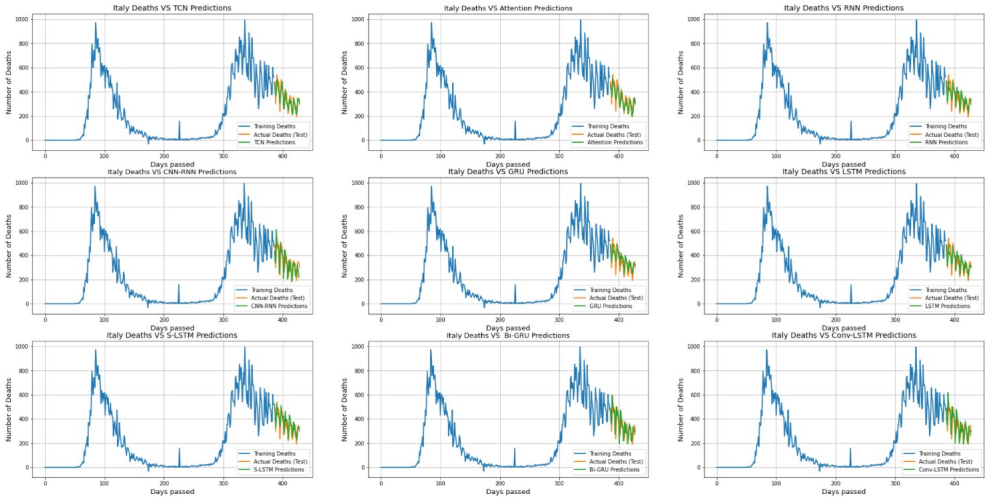
καμπύλη θανάτων είναι μερικώς μη στάσιμη, πράγμα το οποίο σημαίνει ότι είναι ιδιαίτερα πιο ακανόνιστη και δεν εμφανίζει κάποια περίοδο (με τόσο εμφανή τρόπο όσο οι ισχυρώς μη στάσιμες καμπύλες).
Με αλλά λόγια η καμπύλη αυτή είναι πιο στάσιμη και επιτρέπει στα μοντέλα (τα επαναληπτικά κυρίως δίκτυα -RNN,GRU,LSTM) που στο προηγούμενο πείραμα είχαν



Εικόνα 5.9: Αξιολόγηση των μοντέλων στην τεχνική Multivariate-Θάνατοι Ιταλίας

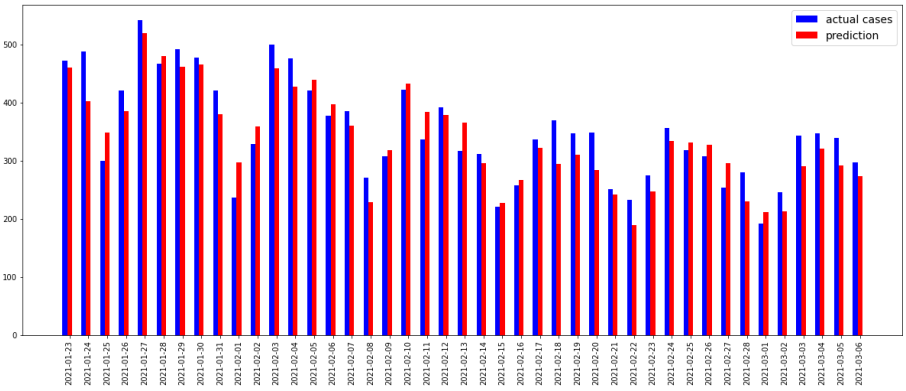


Εικόνα 5.10: Αξιολόγηση των μοντέλων στην τεχνική Differencing-Θάνατοι Ιταλίας



Εικόνα 5.11: Γραφική αναπαράσταση των αποτελεσμάτων όλων των μοντέλων επί των θανάτων της Ιταλίας

πιο αδύναμη απόδοση να αλληλοσυσχετίζουν τα δεδομένα με σωστότερο τρόπο και να προβλέπουν με μεγαλύτερη ακρίβεια. Ειδικότερα, στην τεχνική Univariate το μοντέλο TCN παραμένει στην κορυφή έχοντας την ίδια απόδοση ωστόσο ως προς το τετραγωνικό σφάλμα από το LSTM. Στις τεχνικές Labelling και Multivariate τα αμιγώς επαναλη-



Εικόνα 5.12: Σύγκριση των πραγματικών θανάτων της Ιταλίας με το καλύτερο μοντέλο TCN με Univariate

πτικά μοντέλα κυριαρχούν, καθώς τόσο το LSTM και το S-LSTM όσο και τα RNN, GRU εμφανίζουν υψηλότερη απόδοση. Θα πρέπει να υπογραμμισθεί ότι σε αντίθεση με το προηγούμενο πείραμα η τεχνική Multivariate βοηθά ορισμένα μοντέλα να πετύχουν μεγαλύτερα σκορ, ακριβώς για το λόγο ότι η καμπύλη πλέον ώντας πιο στάσιμη, δίνει την ευκαιρία στα μοντέλα να αλληλοσυσχετίσουν καλύτερα την επιπλέον πληροφορία αλλά και λόγω του ότι τα 2 έξτρα χαρακτηριστικά φαίνονται να συνδέονται άρρηκτα με τους θάνατους καθώς εστιάσαμε σε μια μεμονωμένη χώρα.

Η μέθοδος των περιοδικών διαφορών δείχνει να μην προσφέρει κάτι πέραν του γεγονότος ότι όλα τα μοντέλα καταφέρνουν να αποδώσουν σχετικά ικανοποιητικά στο ίδιο επίπεδο (περίπου 0.7 ως προς το R^2). Πιο συγκεκριμένα, τα υβριδικά δίκτυα και το Bidirectional GRU ανεβάζουν την αποδόση τους στο ίδιο επίπεδο με τα υπόλοιπα. Ωστόσο, επειδή η καμπύλη είναι σχετικά στάσιμη η συγκεκριμένη μέθοδος δεν μας βοηθά σε κάτι παραπάνω και κατά συνέπεια τα μοντέλα που ήδη τα πήγαιναν καλά, δεν βλέπουν κάποια περαιτέρω βελτίωση. Να τονιστεί ότι και σε αυτό το πείραμα η διακύμανση των αποτελεσμάτων στην συγκεκριμένη μέθοδο ήταν μικρότερη. Συνολικά λοιπόν, ως προς τις τεχνικές βλέπουμε ότι τα Differencing και Labelling δεν μας προσφέρουν κάτι παράπανω, ενώ στον αντίποδα η χρήση παραπάνω χαρακτηριστικών μπορεί να αποδειχθεί βοηθητική, ενώ και πάλι μπορούμε να ισχυριστούμε, ότι την υψηλότερη διακύμανση αποδόσεων παρουσιάζουν τα μοντέλα με συνελικτικές προσθήκες.

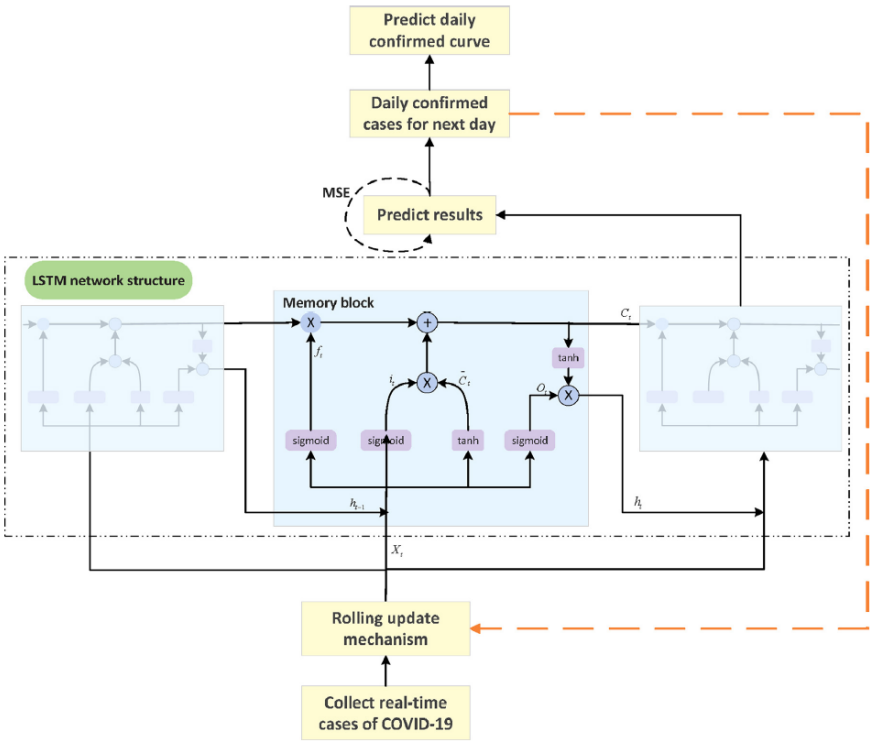
5.3 Πρόσθετοι Πειραματισμοί

Όπως έχει ήδη αναφερθεί στο πλαίσιο της εργασίας έγιναν και κάποιοι πρόσθετοι πειραματισμοί, που μας βοήθησαν να αποκτήσουμε καλύτερα εικόνα των δεδομένων, των μοντέλων αλλά και της πανδημίας.

5.3.1 Τεχνική ανατροφοδότησης, επέκταση πέρα από το Dataset

Εώς τώρα, όλες οι τεχνικές και τα πειράματα των μοντέλων απαιτούν ένα σύνολο δεδομένων εκπαίδευσης (X_{train}) και ένα σύνολο δεδομένων αξιολόγησης (X_{test}). Ωστόσο, υπάρχουν σημαντικές, μεμονωμένες, προσπάθειες που έχουν εστιάσει στην πρόβλεψη της εξέλιξης της πανδημίας σε περίοδους πέρα από αυτές που αναφέρονται στο σύνολο δεδομένων (Dataset). Μια ιδιαίτερα ξεχωριστή τεχνική για να επιτευχθεί αυτό είναι η κατασκευή μοντέλων μηχανικής μάθησης που χρησιμοποιούν μηχανισμούς ανατροφοδότησης [111]. Στην τεχνική αυτή, το σύνολο του Dataset χρησιμοποιείται ως δεδομένο εκπαίδευσης, προκειμένου να δώσουμε όσο το δυνατόν παραπάνω πληροφορία στο δίκτυο. Πλέον αντί για την διαδικασία αξιολόγησης κρατάμε την τελευταία είσοδο του X_{train} την οποία τροφοδότουμε στο δίκτυο προκειμένου να κάνει την πρώτη πρόβλεψη που ξεπερνά το όριο του Dataset.
Το συγκεκριμένο σημείο χρειάζεται διττή προσοχή. Πρώτον, η προβλεπόμενη τιμή θα πρέπει να συνενωθεί με την τελευταία είσοδο του X_{train} και στη συνέχεια να κυλίσουμε το επιλέγμενο παράθυρο (l_w), ώστε να συμπεριλάβουμε στην επόμενη εισόδο μας την νεοπαραχθείσα πρόβλεψή μας. Δεύτερον, σε περίπτωση πολυμεταβλήτης πρόβλεψης, για να επιτευχθεί η συνένωση θα πρέπει να έχουμε τιμές για όλες τις εισόδους. Με άλλα λόγια εξαναγκαζόμαστε να κάνουμε προβλέψεις όχι μόνο για την μεταβλητή ενδιαφέροντος αλλά για όλες τις εισόδους. Αυτό συχνά, έχει ως απότελεσμα όσο προχωράμε σε βάθος χρόνου οι προβλέψεις μας να τείνουν προς σταθερεποίηση.
Στην εικόνα 5.13 μπορούμε να δούμε την διάρθρωση του μοντέλου που χρησιμοποιήθηκε κάνοντας την παραδοχή ότι το καθ' αυτό μοντέλο που χρησιμοποιήθηκε είναι LSTM. Στην πραγματικότητα μπορεί να είναι οποιόδηποτε μόντελο έχουμε γνωρίσει έως τώρα. Ο μηχανισμός ανατροφοδότησης στην εικόνα 5.13 αναφέρεται ως Rolling Update Mechanism, ενώ στην έξοδο του μοντέλου μας δίνονται τα κρούσματα της επόμενης ημέρας. Στα πλαίσια της εργασίας έγινε χρήση της συγκεκριμένης τεχνικής για τα κρούσματα των Ηνωμένων Πολιτειών της Αμερικής και της Ελλάδας με χρήση κάποιων από τα μοντέλα που παρουσιάσαμε νωρίτερα. Η επέκταση έγινε για χρονικό ορίζοντα ενός μήνα (30 ημερών). Συνεπώς, στην μεν πρώτη από τις 418 ημέρες του βασικού Dataset επεκταθήκαμε στις 448 ενώ στη δεύτερη από τις 429 ημέρες του βασικού Dataset επεκταθήκαμε στις 459. Τα μοντέλα που πήραν θέση στο πείραμα ήταν τα TCN, Attention based, Bi-GRU, Conv-LSTM, GRU και LSTM . Ταυτόχρονα, για λόγους πληρότητας και ενδιαφέροντος συλλέξαμε σε ύστερη ημερομηνία, την εξέλιξη της πανδημίας σε αυτές τις χώρες και συγκρίναμε την πραγματικότητα με τις εικασίες των μοντέλων μας.

Στην εικόνα 5.14α′ και 5.15α′ φαίνεται η απόδοση των 6 προαναφερθέντων μοντέλων επί των ΗΠΑ και της Ελλάδας. Με την κόκκινη γραμμή δείχνουμε το χρονολογικό τέλος του Dataset και στη συνέχεια (πάνω στην ίδια καμπύλη με μπλε χρώμα) προσκολλάμε τις προβλέψεις του εκάστοτε μοντέλου. Με κίτρινο χρώμα παρουσιάζεται η πραγμάτικη συνέχεια της πανδημίας σε αυτές τις χώρες. Όπως έχουμε ήδη αναφέρει το τέλος του Dataset ημερολογιακά τοποθετείται στις 7 ή 8-3-2021 (ανάλογα τη χώρα). Η προβλέψεις των εικόνων 5.14α′ και 5.15α′ αφορούν το διάστημα 7-3-2021 έως 6-4-2021. Αντίστοιχα, στις εικόνες 5.14β′ και 5.15β′ δείχνουμε τις προβλέψεις του θεωρητικώς καλύτερου μοντέλου (Attention based). Στον ορίζοντιο άξονα φαίνεται η πραγματική ημερομήνια.

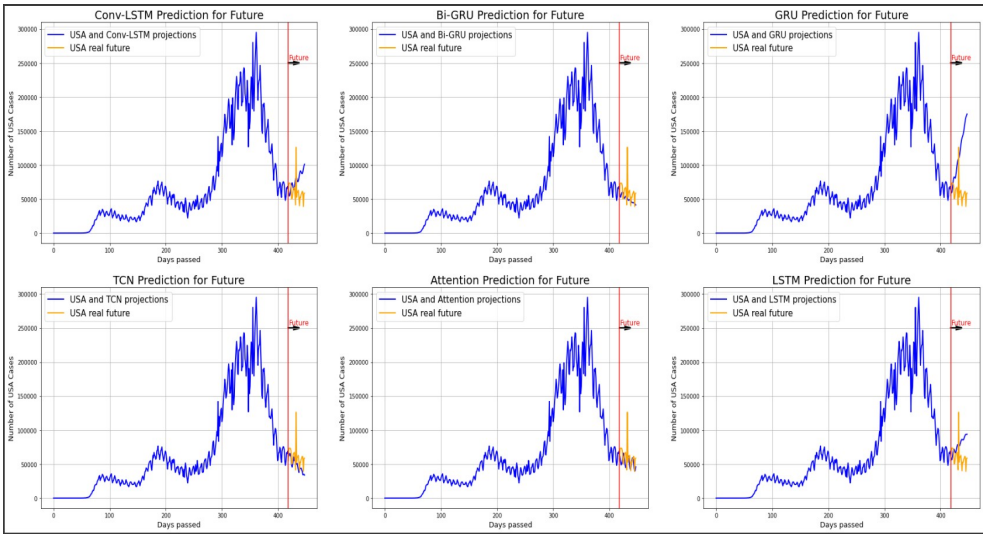


Εικόνα 5.13: Μοντέλο LSTM εφοδιασμένο με τον μηχανισμό ανατροφοδότησης [111]

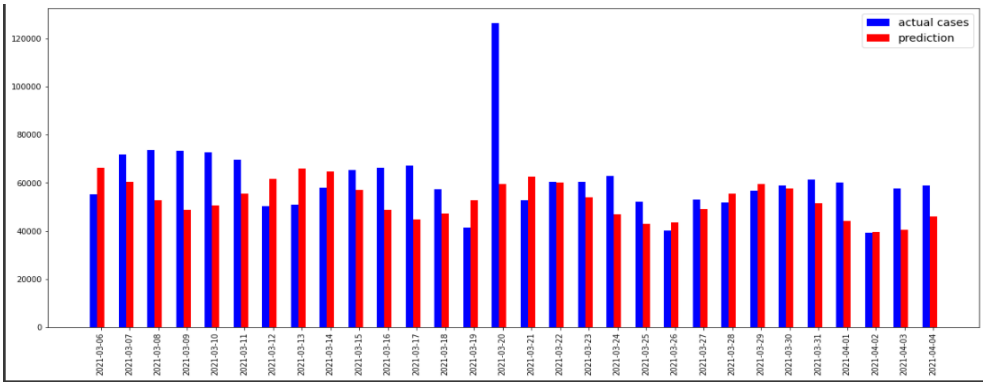
Σχολιασμός

Από τις παραπάνω εικόνες μπορούμε δούμε ότι για την εξέλιξη της νόσου στις ΗΠΑ οι προβλέψεις των μοντέλων ποικίλουν με τα Conv-LSTM και LSTM να 'βλέπουν' μια σταδιακή αύξηση του επιδημικού κύματος. Ταυτόχρονα, το μοντέλο GRU δείχνει ότι η πανδημία θα ακολουθούσε απότομη αύξηση με ένα ενδέχομενο νέο κύμα για τις ΗΠΑ. Ευτυχώς κάτι τέτοιο δε συνέβη. Σε αντίθεση, τα TCN και Bi-GRU προβλέπουν μια σταδιακή μείωση του αριθμού των κρουσμάτων που προσεγγίζει πιο πολύ την πραγματικότητα. Τέλος, το μοντέλο με το μηχανισμό προσοχής δείχνει να πλησιάζει σε αρκετά ικανοποιητικό βαθμό την πραγματική εξέλιξη της νόσου (με εξαίρεση το απότομο spike) όντας πιο κοντά σε αυτή από όλα τα υπόλοιπα μοντέλα. Αυτό επιβεβαιώνεται και από την εικόνα 5.14β΄

Όσον αφορά τη χώρα μας οι προβλέψεις των μοντέλων δεν διαφέρουν σε τόσο μεγάλο βαθμό. Συγκεκριμένα όλα τα μοντέλα προέβλεψαν ότι το $3^ο$ κύμα του ιού στη χώρα θα ήταν πιο συρρικνωμένο σε σχέση με το $2^ο$, καθώς 'θεώρησαν' ότι θα υπάρξει μείωση στο διάστημα Μαρτίου-Απριλίου. Μάλιστα, το μοντέλο Conv-LSTM προέβλεψε απότομη υποχώρηση του επιδημικού κύματος σε σχέση με τα υπόλοιπα μοντέλα που προέβλεψαν πιο σταδιακή μείωση του ικού φορτίου. Για κακή μας τύχη, όπως φαίνεται και από την πορτοκαλί προέκταση, κάτι τέτοιο δεν συνέβη με την Ελλάδα που υπέφερε σφοδρά από

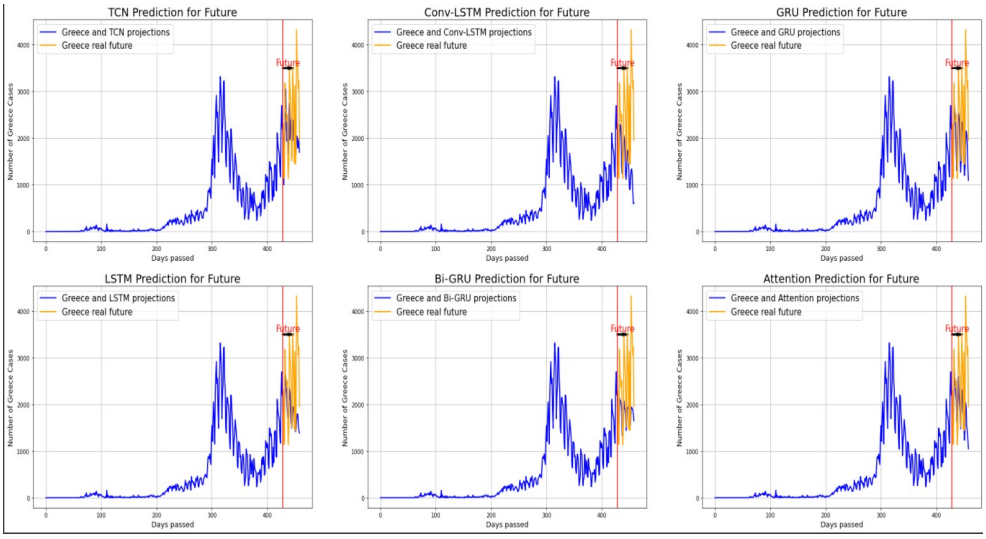


(α′) Προεκτάσεις των 6 μοντέλων επί των κρουσμάτων των ΗΠΑ

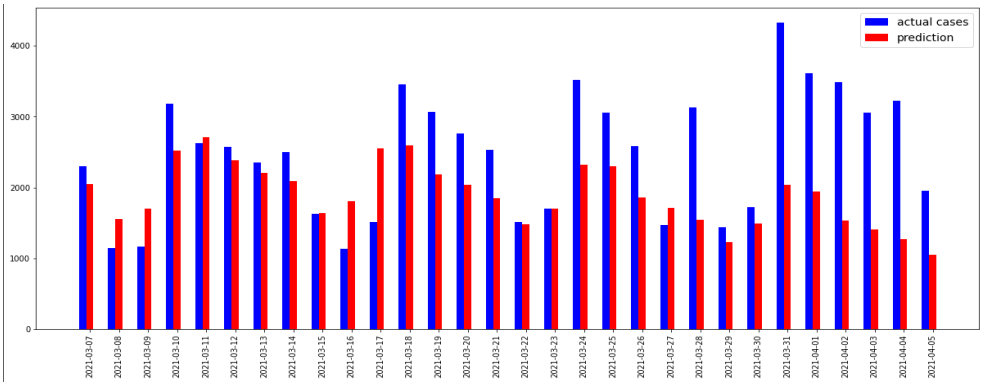


(β′) Σύγκριση προβλέψεων Attention με τα πραγματικά κρούσματα των ΗΠΑ

Εικόνα 5.14: Προβλέψεις μοντέλων για την πορεία της πανδημίας στις ΗΠΑ



(α′) Προεκτάσεις των 6 μοντέλων επί των κρουσμάτων της Ελλάδας



(β′) Σύγκριση προβλέψεων Attention με πραγματικά κρούσματα της Ελλάδας

Εικόνα 5.15: Προβλέψεις μοντέλων για την πορεία της πανδημίας στην Ελλάδα

το $3^ο$ κύμα της πανδημίας την περίοδο Μαρτίου-Απριλίου. Η σχετικά μεγάλη απόκλιση των προβλέψεων με την πραγματική εξέλιξη μπορεί να φανεί και από την εικόνα 5.15β′.

5.3.2 Εύρεση ακραίων τιμών

Όπως έχουμε δει έως τώρα, τόσο σε καμπύλες κρουσμάτων όσο και σε θανάτων οι διακυμάνσεις από ημέρα σε ημέρα ενδέχεται να είναι αρκετά μεγάλες. Σε αυτή την υποενότητα στόχος μας είναι η κατασκευή μοντέλων που θα μπορούν να ανιχεύσουν τέτοια σημεία πάνω στα δεδομένα μας και συνεπώς να τα χαρακτηρίζουμε ως 'ανώμαλα σημεία' ή 'ακραία σημεία'. Για να το πετύχουμε αυτό χρησιμοποιήσαμε ένα ειδικό μοντέλο κωδικοποιητών - αποκωδικοποιήτων τους λεγόμενους *autoencoders* [112].

Autoencoders

Οι autoencoders είναι ένα νευρωνικό δίκτυο που έχει σκοπό να ανακατασκευάσει με επαρκή τρόπο την είσοδο. Η αρχιτεκτονική τους είναι σαν τους απλούς Encoders-Decoders που γνωρίσαμε στην παράγραφο 3.5.1. Ο Encoder μαθαίνει ικανοποιητικές αναπαραστάσεις της εισόδου X μέσα από κωδικοποίηση. Αυτή η κωδικοποιημένη πληροφορία, έστω $\Phi(X)$, βρίσκεται στο τελευταίο στάδιο του κωδικοποιητή που ονομάζεται Bottleneck (αντίστοιχο με το Context Vector). Ο Decoder στη συνέχεια, παράγει μια αναπαράσταση της εισόδου $\Psi(\Phi(X))$ χρησιμοποιώντας την πληροφόρια που βρίσκεται στο Bottleneck. Οι autoencoders συνεπώς μαθαίνουν τη μορφή των δεδομένων στην είσοδο τους και αυτό τους καθιστά data specific, πράγμα που σημαίνει, ότι η κωδικοποίηση και η αποκωδικοποίηση μπορεί να γίνει μόνο σε δεδομένα που το μοντέλο γνωρίζει και έχει εκπαιδευτεί.

Η βασική μαθηματική διατύπωση ενός απλού autoencoder φαίνεται παρακάτω όπου με ϕ και ψ αναφέρουμε τις δράσεις του Encoder και του Decoder.

$$\phi : X \to F(X) \tag{5.3.1}$$

$$\psi : F(X) \to X \tag{5.3.2}$$

$$\phi, \psi = \underset{\phi,\psi}{\operatorname{argmin}} ||X - \psi(\phi(X))|| \tag{5.3.3}$$

Για την κατασκευή των autoencoder χρειάζεται ένας Encoder και ένας Decoder που συνήθως και οι 2 αναπαρίστανται από ένα επαναληπτικό δίκτυο (RNN, GRU, LSTM) ή οποιόδηποτε άλλο δίκτυο έχει τη δυνατότητα να κωδικοποίησει σωστά πληροφορία (λόγου χάρη το TCN). Στα πλαίσια της εργασίας κατασκευάσαμε 3 μοντέλα από autoencoders βασισμένους σε GRU, LSTM και TCN αντίστοιχα. Αυτούς τους ελέγξαμε για 3 διαφορετικές καμπύλες, 2 κρουσμάτων (Παγκόσμια, Ιταλίας) και 1 θανάτων (Ελλάδας).

Διαδικασία εύρεσης ακραίων τιμών

Προκειμένου να εντοπίσουμε πιθανά ακραία σημεία στα δεδομένα μας, ακολουθούμε τη γνωστή διαδικασία εκπαίδευσης του μοντέλου πάνω σε κάποιο υποσύνολο των συνολικών δεδομένων, τα δεδομένα εκπαίδευσης (X_{train}), ενώ το υπόλοιπο μέρος το αφήνουμε στα δεδομένα αξιολόγησης (X_{test}). Εδώ σε αντίθεση με πριν, χρησιμοποιήσαμε αναλογία 80%-20% καθώς χρειαζόμαστε μεγαλύτερο κομμάτι αξιολόγησης για ενδεχόμενη ύπαρξη

ανώμαλων σημείων. Έπειτα από την εκπαίδευση του μοντέλου, ακολουθεί η αξιολόγηση αυτού επί γνωστών δεδομένων, όπως του X_{train} και πάλι.

Από αυτήν την αξιολόγηση ($X_{train-pred}$), λαμβάνουμε το απόλυτο σφάλμα (Absolute Error) από την πραγματική παρατήρηση και την νεοπαραχθείσα πρόβλεψη, δηλαδή $E_{train} = |X_{train} - X_{train-pred}|$. Ακολούθως, με βάση αυτό το σφάλμα διαλέγουμε κάποιο ανώτατο κατώφλι (threshold,Th) από το οποίο θεωρούμε ότι πέρα από αυτό το σφάλμα το υπό εξέταση σημείο θεωρείται ανώμαλο. Συνήθως το κατώφλι αυτό λαμβάνεται ως το ανώτερο (ή ένα από τα ανώτερα) σφάλμα που πραγματοποιήσε το μοντέλο κατά την πρόβλεψη γνωστών δεδομένων ($Th \geq \max(E_{train})$).

Υπό αυτό το πρίσμα, είναι σειρά πλέον να προβλέψουμε τα άγνωστα δεδομένα του X_{test}. Η προβλεψή μας αυτή ($X_{test-pred}$) θα αφαιρεθεί από την πραγματική παρατήρηση έτσι ώστε να πάρουμε και πάλι το απόλυτο σφάλμα, δηλαδή $E_{test} = |X_{test} - X_{test-pred}|$. Διαθέτοντας αυτό το σφάλμα, μπορούμε πλέον να εξετάσουμε, που το μοντέλο μας κατά την πρόβλεψη άγνωστων δεδομένων ξεπέρασε το κατώφλι: $E_{test} > Th$. Αν για κάποιο δεδομένο του X_{test} ισχύει η τελευταία ανισότητα, τότε το συγκεκριμένο δεδομένο κατηγοριοποείται ως ανώμαλο.

Παρακάτω παρουσιάζουμε αναλυτικά πως ένας LSTM autoencoder κατηγοριοποίησε κάποια σημεία ως ακραία επί των παγκόσμιων κρουσμάτων. Προκειμένου να γίνει αντιληπτή η παραπάνω διαδικασία παρουσιάζουμε στις εικόνες 5.16α΄, 5.16β΄ και 5.16γ΄, ιστογράμματα με τα λάθη E_{train} και E_{test} αλλά και μια γραφική αναπαράσταση του κατωφλίου Th επί του E_{test}. Από την πρώτη από αυτές τις 3 εικόνες είναι φανερό, ότι για το σύνολο εκπαίδευσης ένα άνω φράγμα για την τιμή λάθους μπορεί να είναι το 0.15.

Στην εικόνα 5.17α΄ βλέπουμε την τελική θέση των ακραίων σημείων στα δεδομένα αξιολόγησης που μας έδωσε ο LSTM autoencoder. Τέλος, στις εικόνες 5.17β΄ και 5.17γ΄ βλέπουμε την ίδια τοποθέτηση ακραίων σημείων από τους GRU και TCN autoencoders. Για το συγκεκριμένο πείραμα και προκείμενου όλοι οι autoencoders να έχουν μια κοινή βάση χρησιμοποιήθηκε ένα κατώφλι $Th = 0.15$ που ήταν σύμφωνο και με τα 3 μοντέλα. Στη συνέχεια, στις εικόνες 5.18α΄, 5.18β΄ και 5.18γ΄ βλέπουμε την τοποθέτηση των ακραίων από τα 3 είδη autoencoder ως προς τα κρούσματα της Ιταλίας. Σε αυτό το πείραμα η εξαγωγή της τιμής του κατωφλίου έγινε και πάλι λαμβάνοντας υπόψη το απολύτο σφάλμα στο Train set και των 3 μοντέλων και έτσι επιλέχθηκε $Th = 0.25$. Στις εικόνες 5.19α΄, 5.19β΄ και 5.19γ΄ δείχνουμε τα αποτελέσματα των 3 autoencoder στους καθημερινούς θανάτους της χώρας μας. Σε αυτό το πείραμα επιλέχθηκε κατώφλι, $Th = 0.3$

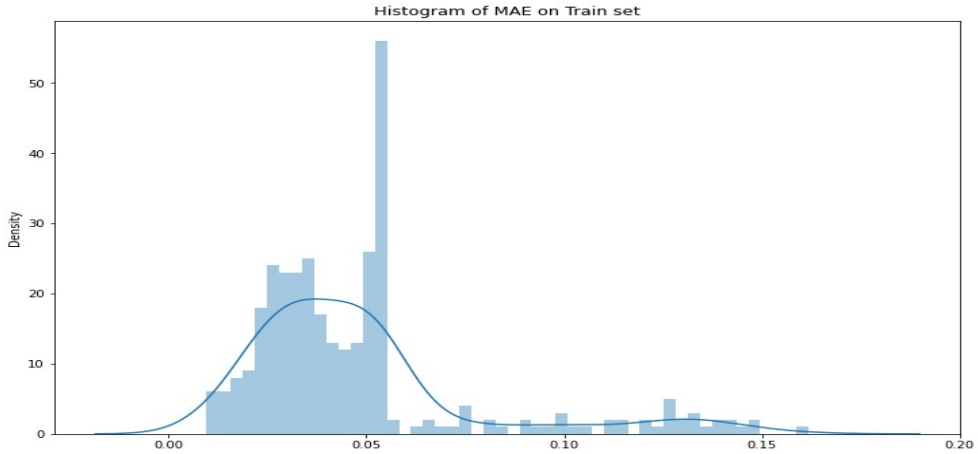
Σχολιασμός

Από τις εικόνες 5.17 βλέπουμε ότι το μοντέλο LSTM έχει κάποια δυσκολία στο να αναγνωρίσει τις απότομες αυξήσεις των κρουσμάτων από το εκάστοτε τοπικό ελάχιστο στην επόμενη μέρα (8 ανώμαλα σημεία). Το μοντέλο GRU χονδρικά αντιμετωπίζει το ίδιο πρόβλημα με την προσθήκη ότι σε ορισμένες περιπτώσεις 'χάνει' και το τοπικό ελάχιστο (13 ανώμαλα σημεία). Το μοντέλο TCN από την άλλη αντιμετωπίζει πολύ λιγότερα προβλήματα, αναδεικνύοντας μόλις 3 σημεία ως ανώμαλα.
Από τις εικόνες 5.18 φαίνεται ότι τόσο ο LSTM όσο και ο TCN autoencoder δυσκολεύονται ιδιαίτερα στην πρόβλεψη της τελευταίας (χονδρικά) εβδομάδας του Test set αναδεικνύοντας όλα τα ακραία τους σημεία στο συγκεκριμένο διάστημα (7 και 6 ανώμαλα

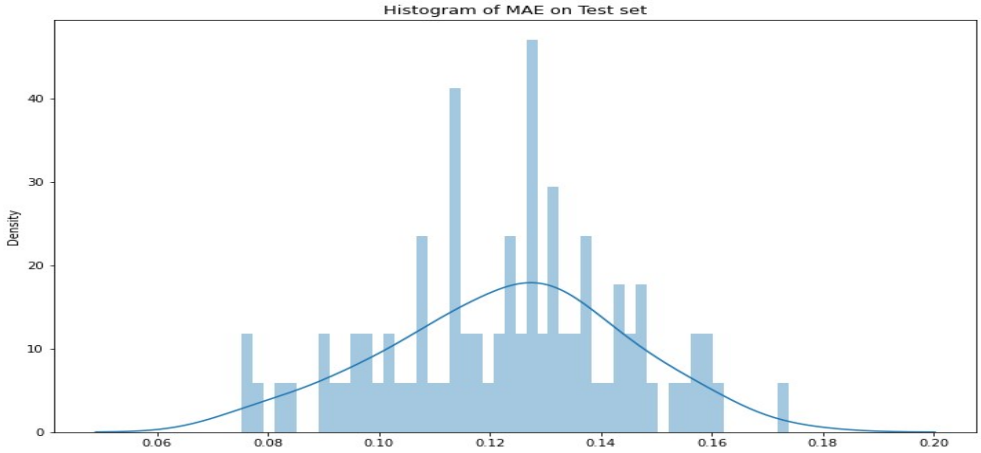
σημεία αντίστοιχα). Αξιοσημείωτο είναι το γεγονός ότι την ίδια ώρα ο GRU autoencoder δεν 'βλέπει' κανένα ανώμαλο δεδομένο καθ' όλο το μήκος της καμπύλης. Θα πρέπει δε, να τονιστεί ότι στα πλαίσια πειραματισμού μειώσαμε το κατώφλι σε $Th = 0.2$ μόνο για το μοντέλο αυτό. Ωστόσο, και πάλι κανένα σημείο δε χαρακτηρίστηκε ως ανώμαλο από συγκεκριμένο το δίκτυο.
Από τις εικόνες 5.19 βλέπουμε μια σχετική ομοιομορφία των αντιδράσεων των 3 μοντέλων στους θανάτους της Ελλάδας. Και τα τρία είδη autoencoder δείχνουν να δυσκολεύονται περισσότερο στο δεύτερο ήμισυ της καμπύλης (κυρίως κατά το μήκος 75-85 επί του Test set). Ειδικότερα, το μοντέλο LSTM εμφανίζει 10 ανώμαλα σημεία σχεδόν όλα στο προαναφερθέν διάστημα. Ταυτόχρονα τα δίκτυα GRU και TCN έχουν ακόμα μεγαλύτερη ομοίοτητα στην τοποθέτηση των ακραίων τους σημείων σε αυτή την καμπύλη με το πρώτο να αριθμεί ένα σύνολο 11 ακραίων σημείων και το δεύτερο 13 σημείων.
Σαν γενικό σχόλιο μπορούμε να ανάφερουμε ότι τα τρία είδη autoencoder έχουν αρκετά παρόμοια αντιμετώπιση των δεδομένων με καθέναν από αυτούς να αναδεικνύεται καλύτερος σε μία από τις τρεις εξεταζόμενες καμπύλες. Ταυτόχρονα, βλέπουμε ότι με την αύξηση της δυσκολίας στο είδος της καμπύλης (μεγαλύτερες και πιο απότομες διακυμάνσεις από ημέρα σε ημέρα) η τιμή του κατωφλίου αυξήθηκε και για τα τρία είδη μοντέλων.

5.3.3 Γεωγραφικά χαρακτηριστικά

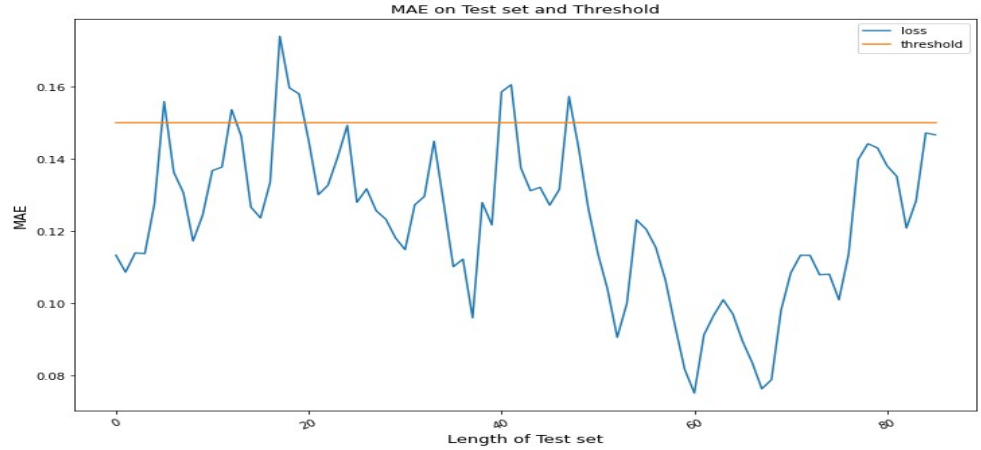
Σε αυτήν την τελευταία παράγραφο των πειραμάτων, στόχος είναι να εξάγουμε κάποια γεωγραφικά χαρακτηριστικά για την πανδημία Covid-19 ανά την υφήλιο με βάση την εξέλιξη των κρουσμάτων σε διάφορες χώρες. Νωρίτερα, στην ενότητα 2.4, είδαμε μέσα από τους γεωγραφικούς θερμοχάρτες που αφορούν τις μετρικές (κρούσματα/θανάτους) ανά εκατομμύριο, στις εικόνες 2.14 και 2.15, την παγκόσμια αποτύπωση της νόσου έως και τον Μάρτιο του 2021. Είδαμε (θεωρητικά) πως η πανδημία έπληξε πολύ περισσότερο τις χώρες της Αμερικανικής Ηπείρου (Βόρεια και Νότια) και της Ευρώπης λιγότερο από τις χώρες τις Ασίας και της Αφρικής.
Να τονιστεί σε αυτό το σημείο πως για λόγους αποφυγής πολιτικοκοινωνικού περιεχομένου στην εργασία, θεωρούμε πως τα δεδομένα των Αφρικανικών και Ασιατικών κρατών είναι ισάξια σε αξιοπιστία με αυτά των δυτικών χωρών.
Έως τώρα, όλα τα πειράματά μας αφορούσαν μία χώρα ανά πείραμα, όπου κάποιο μέρος της καμπύλης των κρουσμάτων ή θανάτων (+ ενδεχομένως κάποιων βοηθητικών χαρακτηριστικών) της υπό εξέτασης χώρας, αποτελούσε τα δεδομένα εκπαίδευσης (X_{train}) και το υπόλοιπο μέρος αφήνονταν για τα δεδομένα αξιολόγησης (X_{test}). Εν αντιθέσει, κατά τη διάρκεια αυτής της πειραματικής διαδικασίας, προτείνουμε μια τεχνική για την εξαγωγή γεωγραφικών χαρακτηριστικών μεταξύ χωρών. Με βάση αυτήν, συλλέξαμε ένα σύνολο, έστω **A**, χωρών που αποτέλεσαν το σύνολο των δεδομένων εκπαίδευσης και πάνω σε αυτές εκπαιδεύσαμε (κάποια από) τα μόντελα μας. Ταυτόχρονα, επιλέξαμε μια (ή δύο) χώρα/ες ως δεδομένα αξιολόγησης (έστω σύνολο **B**). Τελικός στόχος ήταν η όσο το δυνατόν καλύτερη πρόβλεψη ολόκληρης της επιδημιολογικής καμπύλης των κρουσμάτων του/των κράτους/ών του συνόλου **B**.



(α′) Ιστόγραμμα του απόλυτου σφάλματος επι του Train set

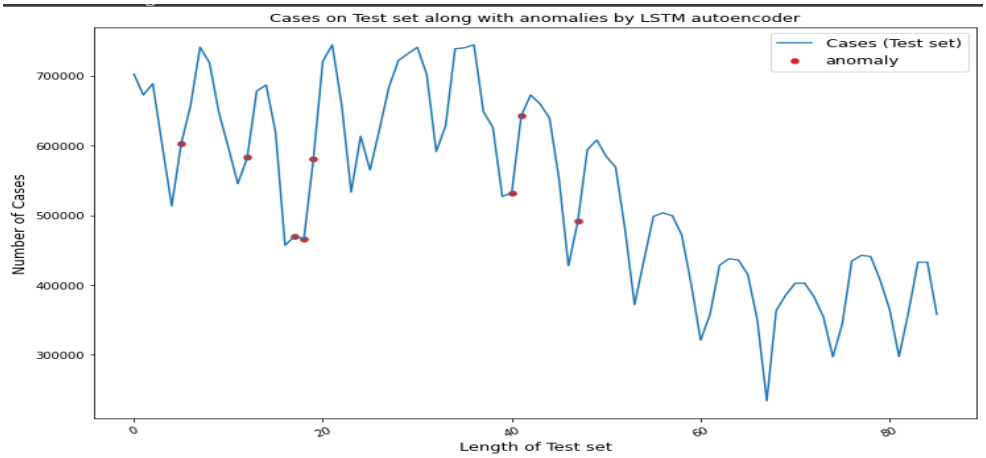


(β′) Ιστόγραμμα του απόλυτου σφάλματος επι του Test set

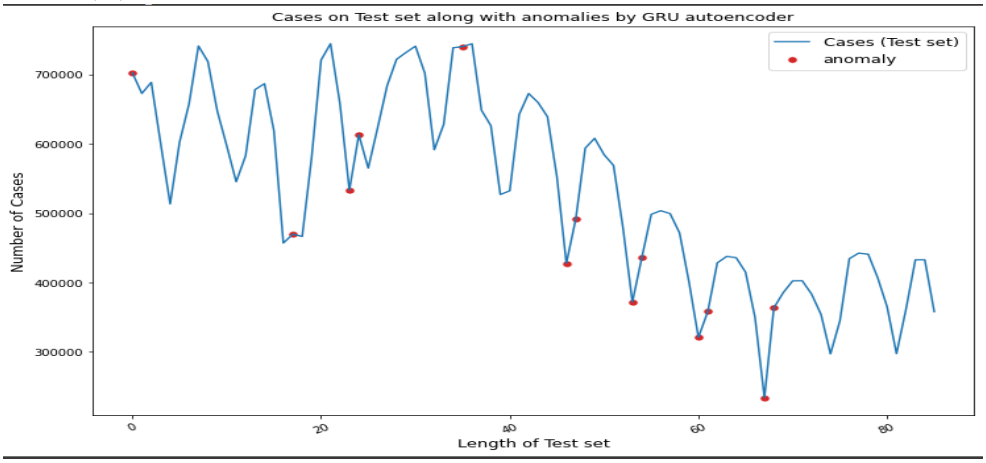


(γ′) Απόλυτο σφάλμα με κατώφλι $Th = 0.15$

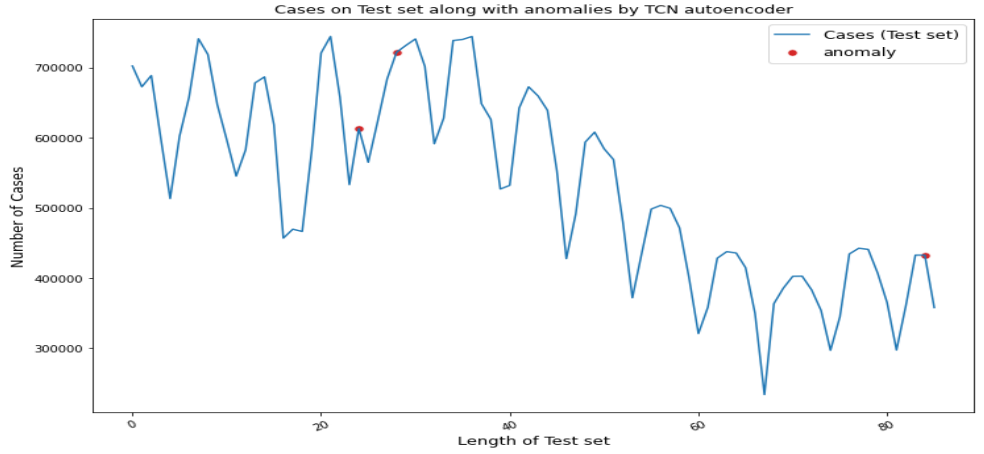
Εικόνα 5.16: Εξαγωγή ακραίων σημείων με autoencoders



(α′) Τοποθέτηση ακραίων σημείων από LSTM autoencoder-Παγκόσμια Κρούσματα

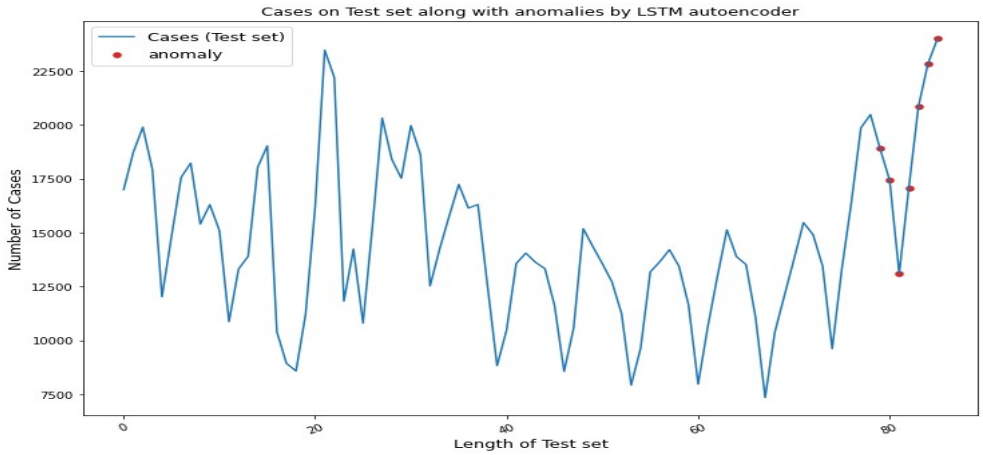


(β′) Τοποθέτηση ακραίων σημείων από GRU autoencoder-Παγκόσμια Κρούσματα

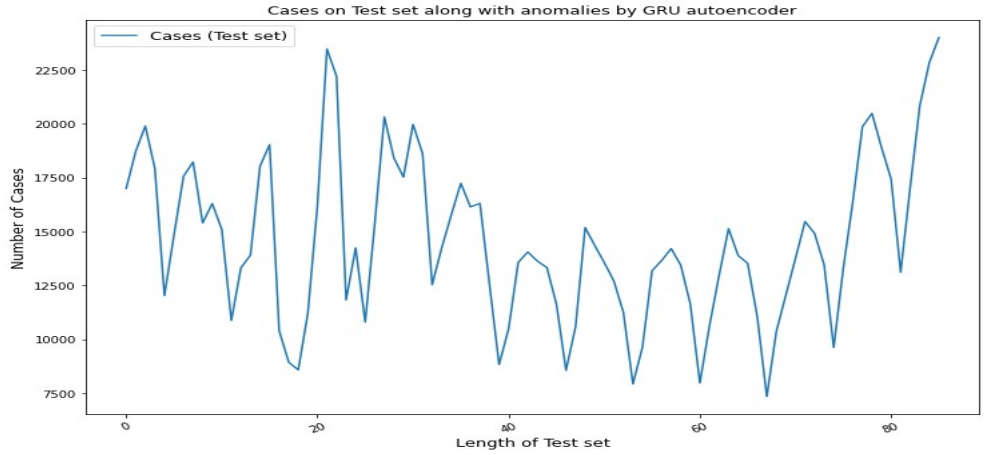


(γ′) Τοποθέτηση ακραίων σημείων από TCN autoencoder-Παγκόσμια Κρούσματα

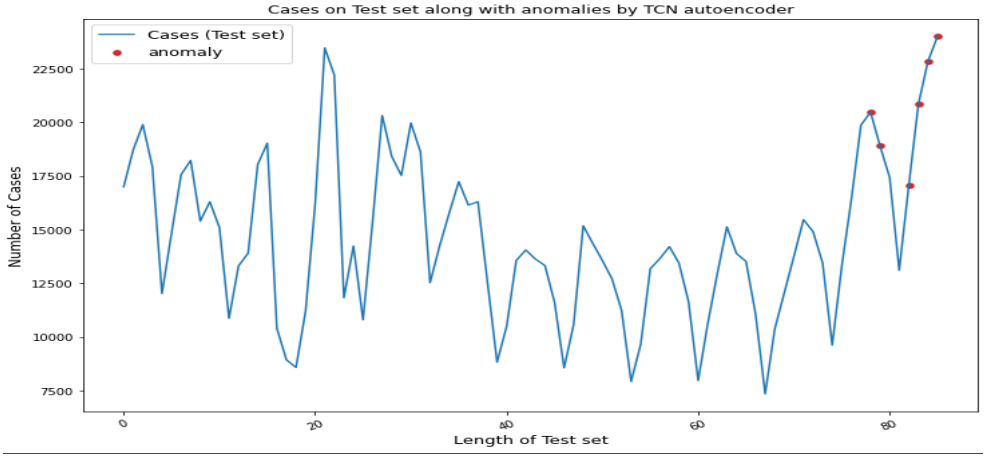
Εικόνα 5.17: Τοποθέτηση ακραίων σημείων από τους 3 autoencoders στα Παγκόσμια Κρούσματα με $Th = 0.15$



(α′) Τοποθέτηση ακραίων σημείων από LSTM autoencoder-Κρούσματα Ιταλίας

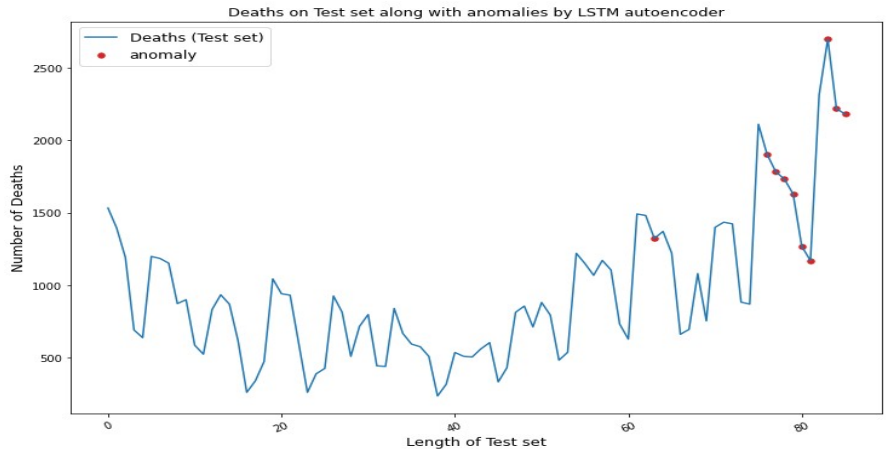


(β′) Τοποθέτηση ακραίων σημείων από GRU autoencoder-Κρούσματα Ιταλίας

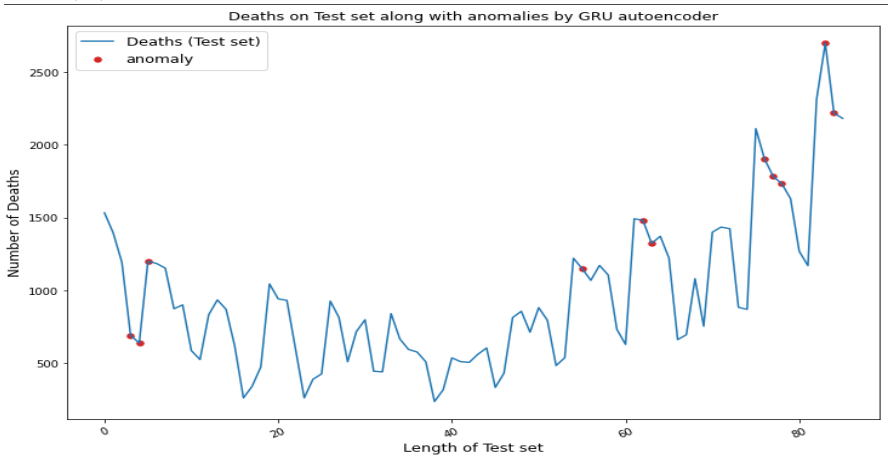


(γ′) Τοποθέτηση ακραίων σημείων από TCN autoencoder-Κρούσματα Ιταλίας

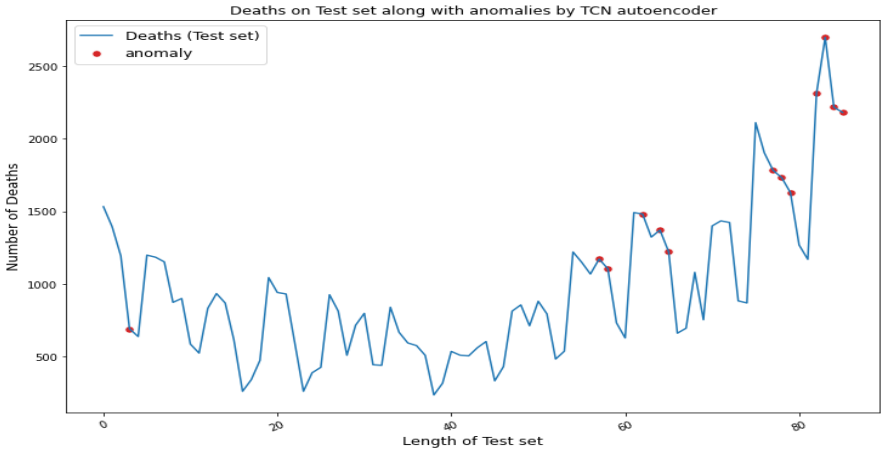
Εικόνα 5.18: Τοποθέτηση ακραίων σημείων από τους 3 autoencoders στα Κρούσματα της Ιταλίας με $Th = 0.25$



(α′) Τοποθέτηση ακραίων σημείων από LSTM autoencoder-Θάνατοι Ελλάδας



(β′) Τοποθέτηση ακραίων σημείων από GRU autoencoder-Θάνατοι Ελλάδας



(γ′) Τοποθέτηση ακραίων σημείων από TCN autoencoder-Θάνατοι Ελλάδας

Εικόνα 5.19: Τοποθέτηση ακραίων σημείων από τους 3 autoencoders στους Θανάτους της Ελλάδας με $Th = 0.3$

Πείραμα 1

Σύμφωνα και με τους χάρτες των εικόνων 2.14 και 2.15 αλλά και με βάση ενδελεχή έλεγχο για την πορεία της πανδημίας σε συγκεκριμένες χώρες, αποφασίσαμε ότι οι χώρες του συνόλου **A** (Train set) θα πρέπει να έχουν σχετικά παρόμοια εικόνα της πανδημίας. Αυτό, για παράδειγμα, συμβαίνει για τις χώρες της Ευρώπης που σε γενικές γραμμές είχαν έναν ενιαίο τρόπο αντιμετώπισης της διασπόρας της νόσου και πάνω κάτω αρκετά κοντινά αποτελέσματα (λ.χ. χονδρικά όλες οι χώρες τις Ευρωπής μαζί με τις ΗΠΑ και τον Καναδά είχαν 3-4 επιδημικά κύματα τα οποία χρονολογικά τα αντιμετώπισαν περίπου ταυτόχρονα).

Σε αντίθεση, η Κίνα για παράδειγμα είχε τελείως διαφορετικές πολιτικές και αποτελέσματα (μόλις ένα επιδημικό κύμα). Για το λόγο αυτό, σε πρώτη φάση, διαλέξαμε και την/τις χώρα/ες του συνόλου **B** (Test set) να έχουν σχετικά κοντινή εικόνα με αυτές του **A**. Επιπρόσθετα, τα μοντέλα που έλαβαν θέση σε αυτό το πείραμα ήταν τα LSTM, GRU, TCN και Attention based. Με βάση τις παραπάνω πληροφορίες, για το πρώτο από αυτά τα πειράματα, επιλέξαμε χώρες του δυτικού κόσμου για τα σύνολα $\mathbf{A}, \mathbf{B}$ τις οποίες αναλυτικά παρουσιάζουμε στον Πίνακα 5.9.

Χώρες συμμετοχής (σε Κρούσματα)	
Σύνολο **A** *(Train set)*	*Σύνολο* **B** *(Test set)*
Αυστρία, Βέλγιο, Βουλγαρία, Καναδάς, Δανία, Γαλλία, Ισλανδία, Ιρλανδία, Ιταλία, Λουξεμβούργο, Ολλανδία, Πορτογαλία, Σλοβενία, Ισπανία, Ηνωμένο Βασίλειο, Ηνωμένες Πολιτείες	Ελλάδα, Κύπρος

Πίνακας 5.9: Επιλογή χωρών για εκπαίδευση και αξιολόγηση σε κρούσματα με (σχετικώς) κοινά χαρακτηριστικά

Απ' όσο φαίνεται και από τον Πίνακα 5.9 οι χώρες του συνόλου εκπαίδευσης ήταν 16 και αυτές προς πρόβλεψη ήταν 2. Επιπρόσθετα, για κάθε χώρα και για κάθε μοντέλο επιλέξαμε ένα συγκεκριμένο και κοινό αριθμό εποχών, μεγέθους δέσμης κλπ. Το σύνολο των επιλογών φαίνεται στον Πίνακα 5.10. Η εκπαίδευση των δικτύων ως προς τις χώρες ήταν σειριακή, ενώ για λόγους απλότητας επιλέχθηκαν μόνον οι τεχνικές Univariate και Labelling (από τις οποίες κρατήσαμε αυτή που απέδωσε καλύτερα).

Στις εικόνες 5.20α′ και 5.20β′ φαίνονται οι τελικές καμπύλες πρόβλεψης που μας έδωσαν τα μοντέλα (με πορτοκαλί) επί των πραγματικών καμπυλών (με μπλε) για την Κύπρο και την Ελλάδα αντίστοιχα. Στις εικόνες 5.21α′ και 5.21β′ παρουσιάζουμε τις επιδόσεις του καλύτερου μοντέλου σε διαγράμματα μπάρας (με κόκκινο) φέρνοντάς το σε αντιδιαστολή με τα πραγματικά κρούσματα των χωρών αξιολόγησης (με σκούρο μπλε).

Με αυτόν τον τρόπο αντιλαμβανόμαστε καλύτερα κατά πόσο το μοντέλο αντιλήφθηκε την εξέλιξη της νόσου κατά τη διάρκεια του χρόνου. Στον ορίζοντιο άξονα υπάρχουν οι πραγματικές ημερομηνίες που όμως, λόγω του πολύ μεγάλου πλήθους τους, αλληλοκαλύπτονται (στην πραγματικότητα αναφέρονται όλες οι ημερομηνιές από 19-1-2020 έως 8-3-2021).

[3] Μόνο για το μοντέλο Attention

Συνδυασμός υπερπαραμέτρων των μοντέλων		
Υπερπαράμετρος	*Συνδυασμός*	*Βέλτιστο*
Neurons	[16,32,64,128,256]	64 ή 128
Neurons Decoder[3]	[32,64,128,256,512]	64 ή 128
Dropout	[None,0.1,0.2]	0.1
Dense layers	[1,2,3]	1
Batch size	[8,16,32,48,64]	16
Epochs	[50,100,150,200]	100
Optimizer	Adam, SGD, RMSprop	Adam
Loss Function	MSE, MAE	MSE

Πίνακας 5.10: Επιλογή υπερπαραμέτρων μοντέλων για το πείραμα του Πίνακα 5.9

Επίσης, στον Πίνακα 5.11 βλέπουμε αναλυτικά τα αποτελέσματα των τεσσάρων (4) μοντέλων που έλαβαν μέρος στο πείραμα ως προς τις μετρικές Absolute Accuracy (A_{ab}) και Squared Accuracy (A_{sq}). Τα καλύτερα σκορ ανά χώρα τονίζονται με έντονη γραφή.

Σχολιασμός

Παρατηρώντας κανείς τις εικόνες 5.20, μπορεί να ισχυριστεί ότι τα μοντέλα δείχνουν να ανταποκρίνονται αρκετά ικανοποιητικά στην συνολική πρόβλεψη των καμπυλών κρουσμάτων της Κύπρου και της Ελλάδας. Πράγματι, όλα τα μοντέλα χωρίς να έχουν καμία απολύτως πληροφορία για την υπό εξέταση χώρα κάνουν μια σχετικώς αξιόπιστη πρόβλεψη. Παρ' όλα αυτά θα πρέπει να προσέξουμε, ότι ενδεχομένως σε κάποια σημεία της καμπύλης οι προβλέψεις να έχουν χρονική καθυστέρηση σε σχέση με την πραγματικότητα, το οποίο στις εικόνες δεν είναι εύκολο να γίνει αντιληπτό.

Για τον λόγο αυτό, τα αποτελέσματα του Πίνακα 5.11 είναι πιο ρεαλιστικά. Από αυτόν βλέπουμε, ότι και στις 2 χώρες το μοντέλο Attention είναι αυτό που πετυχαίνει το υψηλότερο απόλυτο ποσοστό ακρίβειας (κάποιος πολύ παρατηρητικός ενδεχομένως να το καταλάβει αυτό και από τις εικόνες 5.20, όπου το συγκεκριμένο μοντέλο παρουσιάζει λιγότερες χρονικές καθυστερήσεις από τα υπόλοιπα και ταυτοχρόνως διατηρεί την συνολική τάση της επιδημιολογικής καμπύλης).

Από εκεί και πέρα, σε σχέση και πάλι με το A_{ab}, το μοντέλο TCN έρχεται δεύτερο για την Κύπρο ενώ τα LSTM, GRU έχουν κοντινά αποτελέσματα στις 2 τελευταίες θέσεις. Αναφορικά με τη χώρα μας τα LSTM, GRU υπερνικούν το TCN που έρχεται τελευταίο. Συνολικά, όσον αφορά το απόλυτο ποσοστό ακρίβειας, βλέπουμε ότι όλα τα μοντέλα κυμαίνονται γύρω στο 50% αποτελεσματικότητας, πράγμα που σημαίνει πως σε αυτού του είδους το πείραμα (όπου οι χώρες του συνόλου **A** και **B** είχαν μερικώς κοινή εικόνα), τα μοντέλα μετά την εκπαίδευση τους είχαν την δυνατότητα να προβλέψουν λίγο πάνω από 1/2 κρούσματα για τις 'άγνωστες' χώρες του Test set. Το μοντέλο Attention μάλιστα, κατάφερε να πάει και ένα βήμα πέρα από αυτό το φράγμα.

Συνεπώς, το αποτέλεσμά του αποτυπώνεται σε άμεση αντιπάραθεση με την πραγματική εικόνα, και στις εικόνες 5.21, όπου φαίνεται (ιδιαίτερα για την Ελλάδα) πόσο καλή εξέλιξη της νόσου επιτυγχάνει.

Στον Πίνακα 5.11 αναφέρεται και το τετραγωνικό ποσοστό ακρίβειας (A_{sq}). Παρατηρούμε το εξής παράδοξο σχετικά με αυτό. Τα μοντέλα μας πετυχαίνουν αρνητικό (!) ποσοστό ακρίβειας σε αυτή την μετρική. Σε πρώτη όψη αυτό μοιάζει παράλογο. Ωστόσο, θα πρέπει κάνεις να σκεφτεί ότι μόνο για μικρές τιμές απόλυτου σφάλματος τα 2 είδη λαθών (απόλυτο και τετραγωνικό) έχουν κοντινές τιμές. Κάθως το απόλυτο σφάλμα αυξάνεται, το τετραγωνικό εκτοξεύται σε πολύ μεγάλες τιμές (και παρά το ριζικό) καταλήγει να είναι υψηλότερο από το απόλυτο σφάλμα. Εξού, και ότι σε όλα τα πειραμάτά μας έως τώρα είχαμε πάντα ότι $RMSPE > MAPE$.

Στο συγκεκριμένο πείραμα, το $MAPE$ κυμάνθηκε χονδρικά στα μοντέλα μας από 40 έως 50%, πράγμα που σημαίνει ότι το $RMSPE$ είχε πολύ μεγαλύτερες τιμές και σε πολλές περιπτώσεις μεγαλύτερες του 100%. Απότοκο αυτού, ήταν τα αρνητικά ποσοστά τετραγωνικής ακρίβειας. Σε αυτές τις περιπτώσεις το 'λιγότερο' αρνητικό σκορ επιλέχθηκε ως καλύτερο. Τέλος, στο Παράρτημα Β′ παρουσιάζουμε αναλυτικά την σχέση μεταξύ των 2 ποσοστών ακρίβειας (τα οποία είναι ευθέως παράγωγα των μεταβλητών MAE και $RMSE$)

Χώρα	**Μοντέλο**	$\mathbf{A_{ab}}$ **(%)**	$\mathbf{A_{sq}}$ **(%)**
Κύπρος	Attention	**56.42**	**-3.47**
	TCN	52.68	-6.83
	LSTM	48.31	-12.97
	GRU	49.02	-18.62
Ελλάδα	Attention	**61.24**	**20.13**
	TCN	48.99	-11.92
	LSTM	56.68	8.49
	GRU	54.89	-3.99

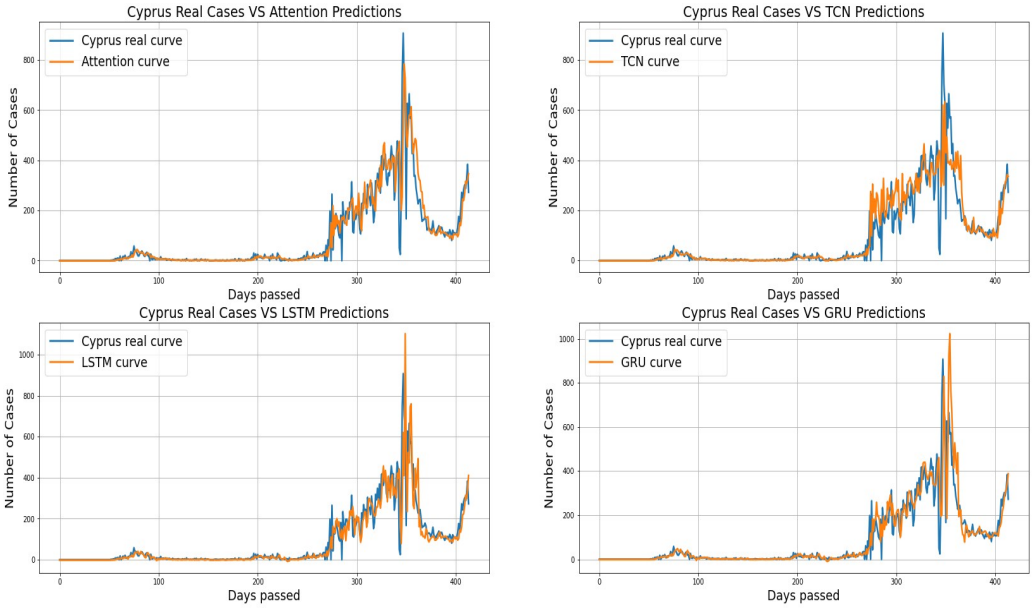
Πίνακας 5.11: Αξιολόγηση των μοντέλων στο πείραμα του Πίνακα 5.9

Πείραμα 2

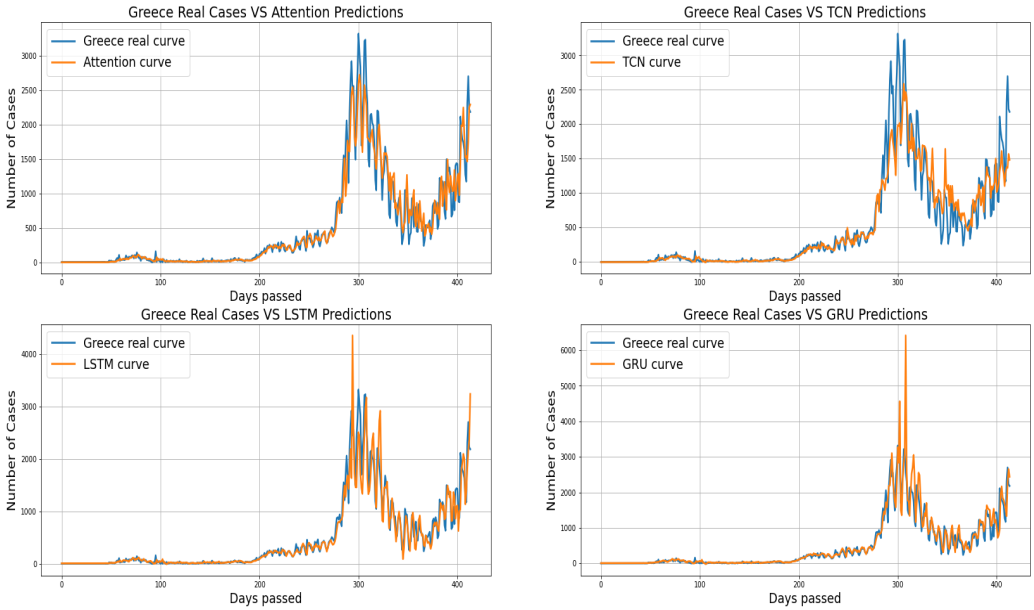
Σε αντίθεση με το προηγούμενο πείραμα όπου προσπαθήσαμε οι χώρες του συνόλου **A** και **B** να έχουν σχετικώς κοινά χαρακτηριστικά, τώρα θα δούμε τα αποτελέσματα που πήραμε κατά την διεξαγωγή πειράματος στο οποίο οι χώρες (θεωρητικά) δεν είχαν τόσο κοντινή είκονα, αναφορικά με την αντιμετώπιση της πανδημίας.

Αρχικά για λόγους άμεσης σύγκρισης θα διατηρήσουμε τις χώρες του συνόλου **B** ίδιες, ήτοι θα αξιολογήσουμε τα μοντέλα μας και πάλι στην Ελλάδα και την Κύπρο. Ωστόσο, οι χώρες του συνόλου **A** τροποποιήθηκαν ώστε πλέον να αποτελούνται από χώρες της Ασίας και της Αφρικής που όπως είδαμε και στους χάρτες των εικόνων 2.14 και 2.15 είχαν διαφορετικά αποτελέσματα από τις Ευρωπαϊκές. Οι χώρες αυτές φαίνονται αναλυτικά στον Πίνακα 5.12.

Επιπλέον, ούτως ή άλλως, γνωρίζουμε ότι οι πολιτικές αντιμετώπισης της νόσου από την Ελλάδα και την Κύπρο απέχουν από τις αντίστοιχες πολιτικές του συνόλου των χωρών αυτών (συγκριτικά πάντα με τις Ευρωπαϊκές). Οι χώρες του Train set παρέμειναν

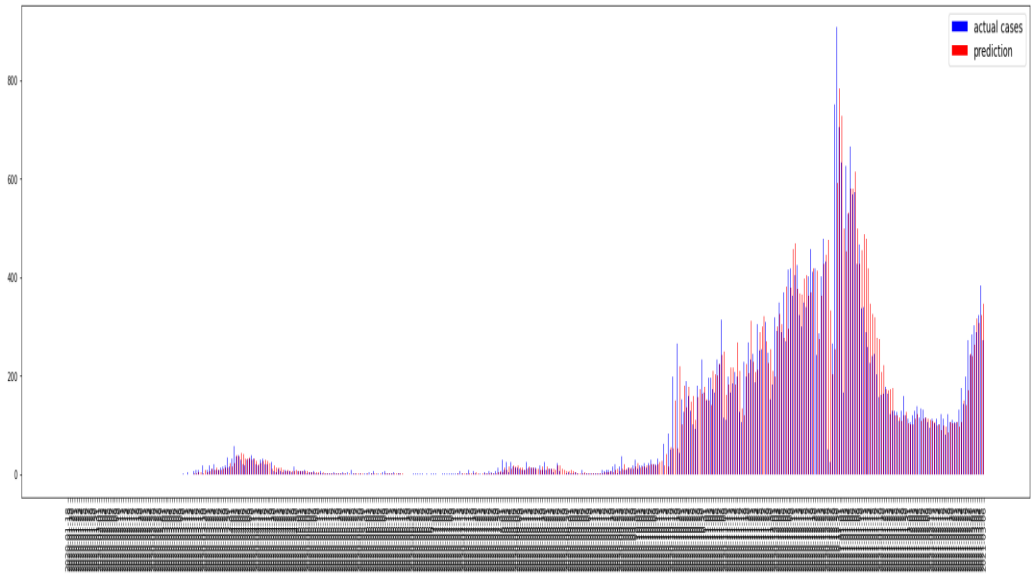


(α′) Αποτελέσμα μοντέλων επί των κρουσμάτων της Κύπρου - Πείραμα 1

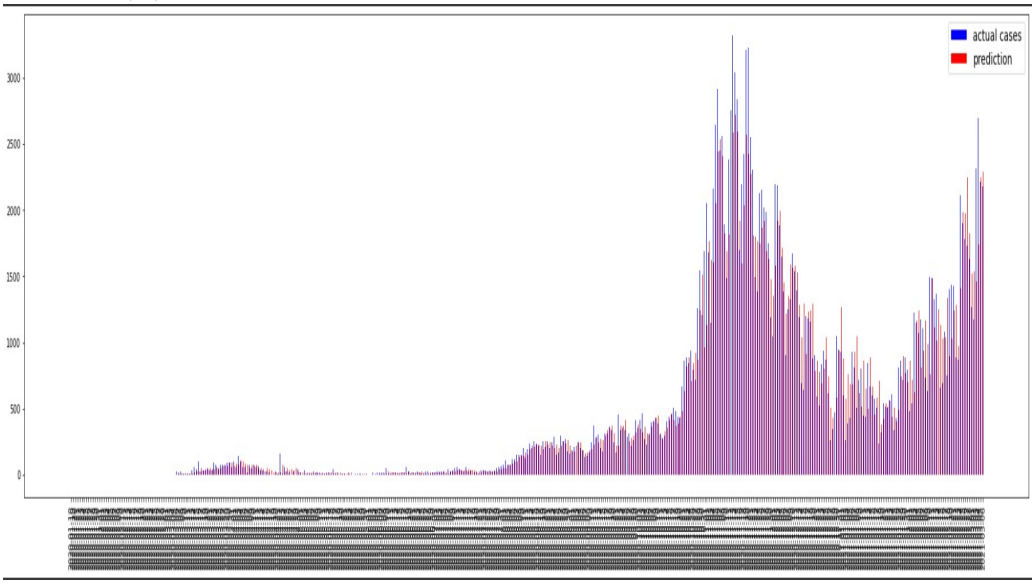


(β′) Αποτελέσμα μοντέλων επί των κρουσμάτων της Ελλάδας - Πείραμα 1

Εικόνα 5.20: Αποτελέσματα των 4 μοντέλων για την Κύπρο και την Ελλάδα στο πείραμα του Πίνακα 5.9



(α′) Σύγκριση του μοντέλου Attention με τα κρούσματα της Κύπρου - Πείραμα 1



(β′) Σύγκριση του μοντέλου Attention με τα κρούσματα της Ελλάδας - Πείραμα 1

Εικόνα 5.21: Σύγκριση του καλύτερου μοντέλου (Attention) με τα πραγματικά κρούσματα έπειτα από το πείραμα του Πίνακα 5.9

16 σε αριθμό. Επιπρόσθετα, τα μοντέλα που έλαβαν μέρος στο πείραμα ήταν τα ίδια τέσσερα με το προηγούμενο πείραμα (LSTM, GRU, TCN και Attention based) των οποίων η εκπαίδευση ήταν και πάλι σειριακή, ενώ και οι υπερπαράμετροί τους διατηρήθηκαν ίδιες όπως στον Πίνακα 5.10.

Χώρες συμμετοχής (σε Κρούσματα)	
Σύνολο **A** *(Train set)*	*Σύνολο* **B** *(Test set)*
Αλγερία, Ανγκόλα, Μπανγκλαντές, Κίνα, Δημοκρατία του Κονγκό, Αίγυπτος, Αιθιοπία, Ινδία, Ιράν, Καζακστάν, Μογγολία, Πακιστάν, Ρωσία, Σαουδική Αραβία, Σουδάν, Τουρκία	Ελλάδα, Κύπρος

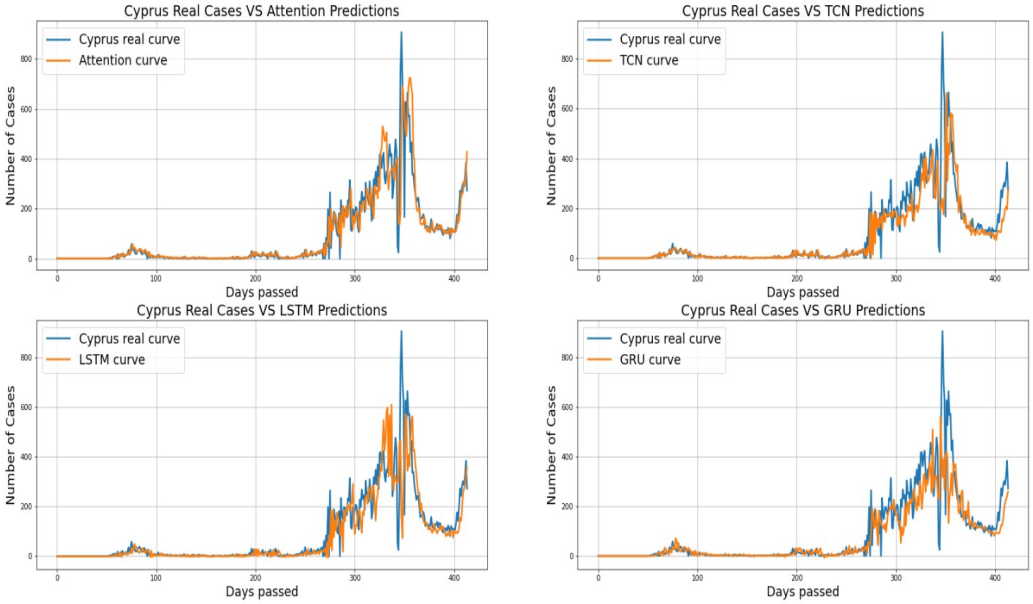
Πίνακας 5.12: Επιλογή χωρών για εκπαίδευση και αξιολόγηση σε κρούσματα με (θεωρητικώς) ανόμοια χαρακτηριστικά

Στις εικόνες 5.22α′ και 5.22β′ φαίνονται με πορτοκαλί οι τελικές καμπύλες πρόβλεψης που μας έδωσαν τα μοντέλα επί των πραγματικών καμπυλών (με μπλε), για την Κύπρο και την Ελλάδα αντίστοιχα. Στον Πίνακα 5.13 βλέπουμε αναλυτικά τα αποτελέσματα των τεσσάρων (4) μοντέλων, και πάλι ως προς τις μετρικές Absolute Accuracy (A_{ab}) και Squared Accuracy (A_{sq}). Τα καλύτερα σκορ ανά χώρα τονίζονται με έντονη γραφή, ενώ οι αρνητικές τιμές για το τετραγωνικό σκορ ακρίβειας οφείλονται στους λόγους που προαναφέρθηκαν. Τέλος, στις εικόνες 5.23α′ και 5.23β′, με κόκκινο παρουσιάζουμε τις επιδόσεις του καλύτερου μοντέλου σε διαγράμματα μπάρας. Ταυτόχρονα, με σκούρο μπλε φαίνονται τα πραγματικά κρούσματα των χωρών αξιολόγησης.

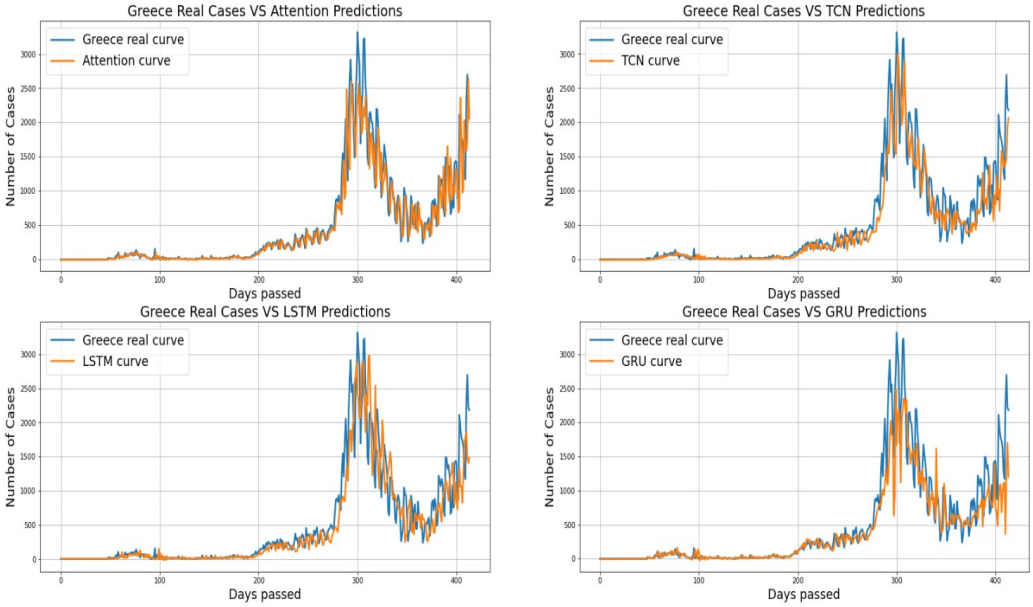
Έτσι, είμαστε και πάλι σε θέση να καταλάβουμε κατά πόσο αυτό το μοντέλο αντιλήφθηκε την εξέλιξη της νόσου κατά τη διάρκεια του χρόνου σε αυτό το 2^o πείραμα. Στον οριζόντιο αξόνα έχουν τοποθετηθεί εκ νέου, οι (πολυάριθμες) πραγματικές ημερομηνίες.

Χώρα	**Μοντέλο**	**A_{ab} (%)**	**A_{sq} (%)**
Κύπρος	Attention	**50.16**	**0.49**
	TCN	45.32	-9.73
	LSTM	42.94	-17.75
	GRU	42.44	-15.7
Ελλάδα	Attention	**56.14**	4.97
	TCN	51.79	**5.71**
	LSTM	49.15	-8.58
	GRU	45.31	-23.82

Πίνακας 5.13: Αξιολόγηση των μοντέλων στο πείραμα του Πίνακα 5.12

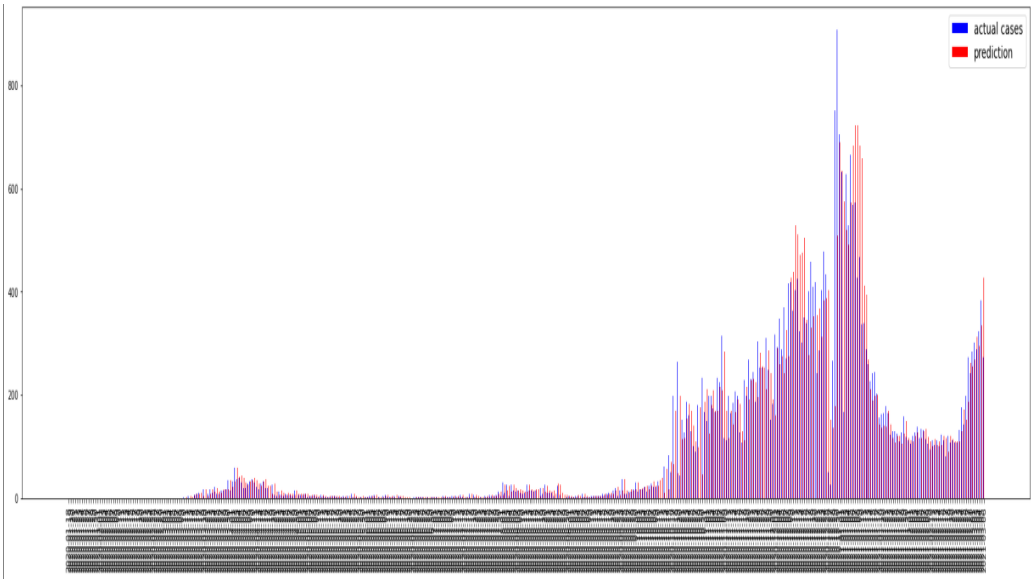


(α′) Αποτελέσμα μοντέλων επί των κρουσμάτων της Κύπρου - Πείραμα 2

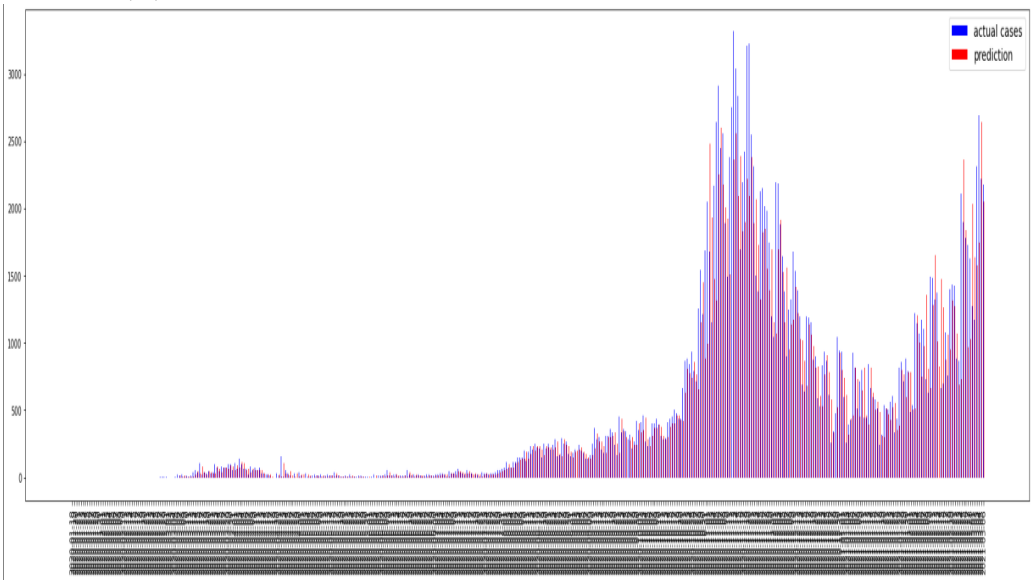


(β′) Αποτελέσμα μοντέλων επί των κρουσμάτων της Ελλάδας - Πείραμα 2

Εικόνα 5.22: Αποτελέσματα των 4 μοντέλων για την Κύπρο και την Ελλάδα στο πείραμα του Πίνακα 5.12



(α′) Σύγκριση του μοντέλου Attention με τα κρούσματα της Κύπρου - Πείραμα 2

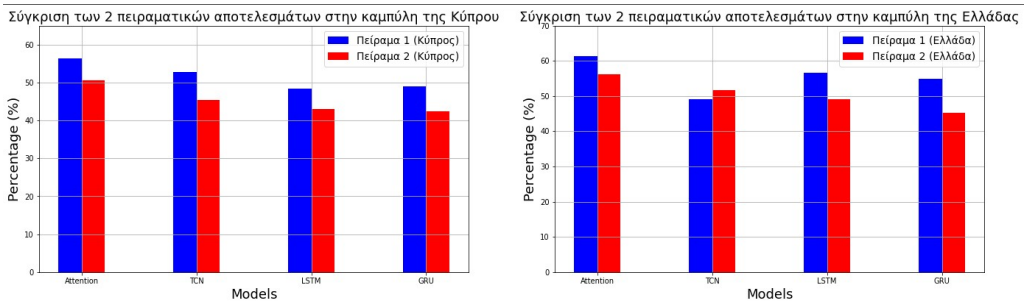


(β′) Σύγκριση του μοντέλου Attention με τα κρούσματα της Ελλάδας - Πείραμα 2

Εικόνα 5.23: Σύγκριση του καλύτερου μοντέλου (Attention) με τα πραγματικά κρούσματα έπειτα από το πείραμα του Πίνακα 5.12

Σχολιασμός

Παρατηρώντας τις εικόνες 5.22, βλέπουμε ότι τα μοντέλα συνεχίζουν να αποτυπώνουν ικανοποιητικά την εξέλιξη της πανδημίας για τις χώρες αξιολόγησης. Ωστόσο, σε σχέση με το Πείραμα 1 φαίνεται ότι δεν αποδίδουν με την ίδια ακρίβεια την πραγματικότητα, τόσο (κυρίως) στην περίπτωση της Κύπρου, όσο και (λιγότερο) στην περίπτωση της χώρα μας. Σε κάθε περίπτωση, η πραγματική απόδοση των μοντέλων καταγράφεται στον Πίνακα 5.13. Συγκρίνοντας αυτά τα αποτελέσματα με τα αντίστοιχα του Πίνακα 5.11 φαίνεται ότι στην πλειοψηφία τους τα σκορ των μοντέλων, ως προς το απόλυτο ποσοστό ακρίβειας, χειροτέρεψαν σε αυτό το πείραμα. Εξαίρεση αποτελεί το μοντέλο TCN κατά την πρόβλεψη των ελληνικών κρουσμάτων.
Θα πρέπει να τονισθεί ότι κατά τη διάρκεια αυτού του πειράματος παρατηρήθηκε ισχυρή παρουσία του προβλήματος χρονικής καθυστέρησης κατά την πρόβλεψη από τα μοντέλα. Αυτός είναι και ένας από τους λόγους μείωσης των ποσοστών των μοντέλων στο A_{ab}. Ηχηρό παράδειγμα σε αυτό αποτελεί το Attention based δίκτυο όπου κατά την πρόβλεψη στην Ελλάδα δείχνει (από τις εικόνες 5.22) να τα πηγαίνει πολύ καλά. Παρ' όλα αυτά το σκορ του είναι μειωμένο σε σχέση με το Πείραμα 1. Το πρόβλημα αυτό παρατηρείται εμφανώς και στο δίκτυο LSTM (και πάλι κατά την πρόβλεψη της Ελλάδας).
Όσον αφορά την απόδοση των μοντέλων μεταξύ τους, παρατηρούμε και εδώ πρωτιά του μοντέλου με τον μηχανισμό της προσοχής (ανεπαίσθητη εξαίρεση αποτελεί το τετραγωνικό ποσοστό ακρίβειας για την Ελλάδα όπου έρχεται πρώτο το μοντέλο TCN). Τη δεύτερη θέση καταλαμβάνει το TCN ενώ τελευταία έρχονται τα LSTM και GRU. Για το λόγο αυτό, το αποτέλεσμά του αποτυπώνεται και μόνο του στις εικόνες 5.23.
Συνολικά, σε σχέση με το Πείραμα 1 βλέπουμε μία μείωση του ποσοστού (απόλυτης) ακρίβειας από τα μοντέλα της τάξεως του 5 έως 10%. Συνεπώς, βλέπουμε ότι στο πείραμα όπου οι χώρες των συνόλων **A** και **B** είχαν (θεωρητικώς) διαφορετικά χαρακτηριστικά, τα μοντέλα μας (κατά μέσο όρο) πλέον προβλέπουν με λιγότερο από 50% αποτελεσματικότητα. Αυτό επαληθεύει την αρχική μας πεποίθηση, ότι δηλαδή δίνοτας για εκπαίδευση στα μοντέλα χώρες, με διαφορετική εικόνα κρουσμάτων και της πανδημίας γενικά, σε σχέση με τις χώρες αξιολόγησης, θα δυσκόλευε και εν τέλει θα χειροτέρευε την απόδοση αυτών. Παρόμοια συμπεριφορά βλέπουμε και στο ποσοστό ακρίβειας A_{sq}. Τέλος, στην εικόνα 5.24 βλέπουμε συγκεντρωτικά την άμεση σύγκριση των αποτελεσμάτων των μοντέλων κατά την διάρκεια των δύο πειραμάτων ως προς το A_{ab}.



Εικόνα 5.24: Αναπαράσταση της απόδοσης των μοντέλων στα 2 πειράματα που βοηθά στην εξαγωγή γεωγραφικών συμπερασμάτων για τις χώρες στους Πίνακες 5.9 και 5.12

Κεφάλαιο 6

Συμπεράσματα και μελλοντικές επεκτάσεις

6.1 Συμπεράσματα

Στόχος αυτής της διπλωματικής ήταν η πρόβλεψη των κρουσμάτων και των θανάτων της νόσου Covid-19 με βάση την παρελθοντική εξέλιξη αυτών αλλά και με τη βοήθεια κάποιων τεχνικών ή επιπλέον χαρακτηριστικών. Σε αυτό το κεφάλαιο, θα μιλήσουμε για τα σημαντικότερα συμπεράσματα που εξήχθησαν από τα πειράματα του Κεφαλαίου 5. Έχουμε πάρει ήδη μια πρώτη εικόνα από τους σχολιασμούς του εκάστοτε πειράματος. Η εξαγωγή των συμπερασμάτων θα γίνει σε αντιστοιχία με το κάθε πείραμα.

6.1.1 Συμπεράσματα πειραματικής διαδικασίας σε καμπύλες κρουσμάτων και θανάτων

Τα πειράματα της ενότητας 5.2 χωρίστηκαν σε 2 είδη. Στο πρώτο πείραμα κάναμε πρόβλεψη πάνω σε καμπύλη κρουσμάτων, ενώ στο δεύτερο σε καμπύλη θανάτων. Η επίλογες αυτές δεν έγιναν τυχαία. Η καμπύλη κρουσμάτων είδαμε ότι είναι πιο περιοδική με πιο συκγεκριμένη τάση, πράγμα που την καθιστά μη στάσιμη. Όπως έχει αναφερθεί, αυτό συνέβη (πιθανώς) εξαιτίας του γεγονότος ότι η πλειοψηφία των χωρών διεξήγαγε τεστ με περιοδικό ρυθμό. Σε αντίθεση, η καμπύλη των θανάτων δεν παρουσιάζει την ίδια περιοδικότητα, πράγμα λογικό, καθώς το γεγονός αυτό δεν συμβαίνει περιοδικά. Η καμπύλη αυτή είναι περισσότερο στάσιμη. Κατά την διάρκεια του πειράματος στην μη στάσιμη καμπύλη (κρούσματα) είδαμε:

- Το μοντέλο TCN ανταποκρίνεται καλύτερα στην μονομεταβλητή πρόβλεψη με το προτεινόμενο μοντέλο Attention να έρχεται δεύτερο. Στα υπόλοιπα 3 είδη (Multivariate, Labelling και Differencing)) το μοντέλο Attention επικρατεί. Με βάση αυτά, συνάγουμε το συμπέρασμα, ότι η προτεινόμενη αρχιτεκτονική με τον μηχανισμό Προσοχής μπορεί να ανταποκριθεί καλύτερα κατά την πρόβλεψη μη στάσιμων δεδομένων. Η υπεροχή αυτή φαίνεται καλύτερα και στην μέθοδο των

περιοδικών διαφορών όπου το συγκεκριμένο μοντέλο και το Bidirectional GRU δείχνουν να αφομοιώνουν καλύτερα την τάση της καμπύλης.

- Τα μοντέλα με συνελικτικές προσθήκες δεν έδειξαν να προσφέρουν κάτι το ιδιαίτερο στην προβλεψη αυτή. Εξαίρεση αποτελεί το μοντέλο TCN που σε κάποιες περιπτώσεις έδειξε πολύ υψηλή απόδοση. Τα ίδια μοντέλα έδειξαν να έχουν και ελαφρώς μεγαλύτερη διακύμανση αποτελεσμάτων.

- Η προσθήκη πολλαπλών επιπέδων επαναληπτικών μοντέλων (S-LSTM) δεν έδειξε να βελτιώνει το τελικό αποτέλεσμα. Θα πρέπει να τονισθεί ότι ενδεχομένως τέτοια δικτύα με μεγαλύτερο αριθμό νευρώνων να βοηθούσαν. Η κατασκευή τους ωστόσο κρίνεται ιδιαίτερα δαπανηφόρα σε υπλογιστικούς πόρους. Ταυτόχρονα, η προσθήκη αμφίδρομου στρώματος βοηθά μονό στην τεχνική Differencing.

- Ως προς τις τεχνικές, η παρουσία μη στάσιμης καμπύλης καθιστά την μέθοδο περιοδικών διαφορών ιδιαίτερα βοηθητική. Η τεχνική Univariate (ως βασική) δείχνει να είναι ικανοποιητική. Η προτεινόμενη μέθοδος Labelling δείχνει να μας βοηθά σε μεμονωμένες περιπτώσεις και (ενδεχομένως) να μειώνει ανεπαίσθητα την διακύμανση των αποτελεσμάτων. Επιπλέον, χρήση βοηθητικού χαρακτηριστικού δεν βοηθά σε αυτό το πείραμα. Οι πιθανοί λόγοι για αυτό αναφέρθηκαν αναλυτικά στον σχολιασμό της παραγράφου 5.2.1. Τέλος, η τεχνική Differencing δείχνει να μειώνει την διακύμανση των αποτελεσμάτων των μοντέλων.

Κατά την διάρκεια του πειράματος στην περισσότερο στάσιμη καμπύλη (θάνατοι) είδαμε:

- Το μοντέλο TCN ανταποκρίνεται και πάλι καλύτερα σε σχέση με τα υπόλοιπα κατά την τεχνική Univariate. Ωστόσο, σύμφωνα και με τις υπόλοιπες τεχνικές τα πράγματα δεν είναι τόσο ξεκάθαρα. Τα επαναληπτικά μοντέλα (RNN, LSTM, GRU) δείχνουν να βοηθιούνται από την ύπαρξη μιας περισσότερο στάσιμης καμπύλης κάθως τους είναι πιο εύκολο να συσχετίσουν τα πιο 'ακανόνιστα' δεδομένα. Το προτεινόμενο Attention based μοντέλο για τον ίδιο λόγο δεν δείχνει να μπορεί να αποδώσει την ίδια προσοχή στα στοιχεία της καμπύλης.

- Η τεχνική της επισήμανσης δεν δείχνει να βοηθά κανένα από τα μοντέλα μας, παρ' όλο που θεωρητικά αντιμετωπίζει το πρόβλημα των απότομων και μεγάλων αλλαγών. Τα έξτρα χαρακτηριστικά στην πολυμεταβλητή πρόβλεψη δείχνουν να βοηθούν ορισμένα μοντέλα. Ο λόγος για αυτό είναι ότι συνδέονταν άμεσα με την μεταβλήτη πρόβλεψης χώρις αποκλίσεις. Και στις 2 αυτές τεχνικές τα απλά επαναληπτικά μοντέλα δείχνουν να κυριαρχούν.

- Κατά τις 3 πρώτες τεχνικές η προσθήκη αμφίδρομου στρώματος νευρώνων δεν βοηθά. Στον αντίποδα, η χρήση έξτρα στρωμάτων βρέθηκε ότι σε ορισμένες περιπτώσεις να βελτιώνει την απόδοση (S-LSTM έναντι LSTM,RNN κ.λ.π). Τα υβριδικά δίκτυα και πάλι δεν δείχνουν να ξεχωρίζουν ενώ αναδεικνύουν ελαφρώς αυξημένη διακύμανση αποτελεσμάτων.

- Κατά την τεχνική Differencing δεν παρατηρήθηκε καμία αξιοσημείωτη βελτίωση, ενώ όλα τα μοντέλα είχαν κοντινές αποδόσεις. Η διακύμανση των αποτελεσμάτων

των μοντέλων μειώθηκε και πάλι, ενώ με αμυδρή διαφορά καλύτερη απόδοση παρουσιάζουν για πρώτη φορά τα μοντέλα με συνελικτικούς μηχανισμούς. Αυτό ενδεχομένως να συμβαίνει, επειδή τα συνελικτικά στρώματα κατάφεραν στο συγκεκριμένο πείραμα να αποσπάσουν την κατάλληλη χωρική πληροφορία και με την βοηθεία της μεθόδου, τα μοντέλα αυτά, να μειώσουν το ποσοστό λάθους.

Συνολικά ξεχωρίζουν τα εξής σημεία: α) Προτείνουμε την χρήση TCN για μονομεταβλήτη πρόβλεψη, β) την χρήση του προτεινόμενου μοντέλου Προσοχής για μη στάσιμες καμπύλες γ) την χρήση της μεθόδου Differencing επίσης για καμπύλες με μια ενδεδειγμένη τάση και περίοδο δ) την 'κλασσική' χρήση (όλων των ειδών) RNNs σε πιο 'ακανόνιστες' καμπύλες, δηλαδή πιο στάσιμες, ε) χρήση της τεχνικής Multivariate μόνον όταν τα χαρακτηριστικά είναι ισχυρώς συσχετισμένα χωρίς 'κενά' και στ) την χρήση της προτεινόμενης τεχνικής Labelling κατόπιν διερεύνησης.

6.1.2 Συμπεράσματα στην πειραματική διαδικασία της τεχνικής ανατροφοδότησης

Στο συγκεκριμένο πείραμα δεν υπήρξε αξιολόγηση των μοντέλων που έλαβαν μέρος ως προς κάποια μετρική. Ο λόγος που επιλέχθηκε αυτό ήταν η απουσία πραγματικών δεδομένων αξιολόγησης (X_{test}). Κατά την αξιολόγηση σε πραγματικά δεδομένα τα μοντέλα σε κάθε βήμα χρόνου (timestep) ακόμα και αν κάνουν μια αρκετά λανθάσμενη πρόβλεψη στο χρονικό βήμα t, στο επόμενο χρονικό βήμα, $t+1$, έχουν και πάλι ως βάση, μια πραγατική τιμή για την προβλεψή τους. Αυτό ενισχύεται στη πρόβλεψη χρονοσειρών καθώς την χρονική στιγμή $t+1$ το μοντέλο πλέον έχει ως δεδομένο και την τιμή που 'προσπάθησε' να προβλέψει στο χρονικό βήμα t. Έτσι, οι πιθανότητες να απομακρυνθούμε σε πολύ μεγάλο βαθμό από την πραγματικότητα λιγοστεύουν, σε περιπτώσεις που φυσικά, το μοντέλο δεν έχει υπερπροσαρμοστεί, δεν έχει αντιμετωπίσει το πρόβλημα εξαφάνισης/έκρηξης των κλίσεων κ.α.
Σε αυτό το πείραμα όπως γνωρίσαμε το κάθε μοντέλο προέβλεπε ένα χρονικό βήμα στο μέλλον (μετά το πέρας του Dataset-έστω πρόβλεψη t_0). Έπειτα λάμβανε στην είσοδο (ανατροφοδοτώντας) αυτήν την τιμή και προχωρούσε στην επόμενη πρόβλεψη (έστω t_1). Η διαδικάσια αυτή συνεχίζονταν για τον επιλεγμένο χρονικό ορίζοντα T (εμείς επιλέξαμε $T = 30$). Εδώ καταλαβαίνει κανείς πως όσο μεγαλύτερος είναι αυτός ο χρονικός ορίζοντας, τόσο πιο πολύ θα 'ξεφεύγει' το μοντέλο από τις πραγματικές μετρήσεις[1]. Αυτό είχε ως συνέπεια την μεγάλη απόκλιση των μοντέλων από την πραγματικότητα όσον αφορά τις μετρικές σφάλματος ή αποτελεσματικότητας. Για τον λόγο αυτό αποκλείσαμε τις μετρικές από το πείραμα αυτό.
Στόχος ήταν να δούμε ποιό μοντέλο ανταποκρίθηκε καλύτερα και προσέγγισε καλύτερα την πραγματικότητα. Όπως είδαμε και στις εικόνες 5.14α′ και 5.15α′, στην περίπτωση των ΗΠΑ, το προτεινόμενο μοντέλο με Attention ξακάθαρα πλησιάσε πιο πολύ την πραγματική επιδημιολογική εικόνα της χώρας, ενώ σε αυτήν της χώρας μας όλα τα μοντέλα 'έπεσαν έξω'. Σχετικά πιο κοντά ήταν και πάλι το μοντέλο της Προσοχής σε συνδυασμό με το μοντέλο GRU. Προτείνουμε τον πειραματισμό διαφορετικών μοντέλων σε τέτοιου είδους πειράματα με μεγαλύτερο χρονικό ορίζοντα T και ιδιαιτέρως μοντέλων με Attention, κομμάτι που δεν έχει ακόμα αναδειχθεί στην διεθνή βιβλιογραφία.

[1] Είναι, για παράδειγμα, σαν να προσπαθούμε να προβλέψουμε τον καιρό πολλές ημέρες στο μέλλον

6.1.3 Συμπεράσματα στην πειραματική διαδικασία εύρεσης ακραίων τιμών

Κατά την διάρκεια αυτού του πειράματος στόχος ήταν να βρούμε τα σημεία, των υπό εξέταση καμπυλών την εκάστοτε φορά, που παρουσίασαν κάποια 'ανωμαλία', με βάση τους τρεις βασικούς autoencoders που κατασκευάσαμε. Μέσα από αυτήν την διαδικασία, μπορούν να εξαχθούν συμπεράσματα για το ποιά είδη δεδομένων και ποιές κατανομές αυτών 'δυσκολεύουν' τα μοντέλα μας που βασίστηκαν στα LSTM, GRU και TCN αντίστοιχα.
Από τα 3 πειράματα που διεξήχθησαν (σε 3 διαφορετικές καμπύλες) είδαμε ότι σε γενικές γραμμές όλα τα είδη autoencoder αντιμετωπίζουν παρόμοιες δυσκολίες στην αναγνώριση σημείων ως ακραίων, υπό την έννοια ότι χονδρικά καταχώρησαν σχετικά κοντινά σημεία των καμπυλών ως ακραία. Στον Πίνακα 6.1 παρουσιάζουμε συνοπτικά όσα σχολιάσαμε στην παράγραφο 5.3.1.

Κατανομή ανώμαλων σημείων			
Autoencoder	Παγκόσμια Κρούσματα	Κρούσματα Ιταλίας	Θάνατοι Ελλάδας
LSTM	8	7	**10**
GRU	13	**0**	11
TCN	**3**	6	13

Πίνακας 6.1: Συγκεντρωτική παρουσίαση των ακραίων σημείων των μοντέλων στο πείραμα της παραγράφου 5.3.2

Με βάση αυτόν και τις εικόνες 5.17, 5.18 και 5.19 επαληθεύουμε τα όσα προαναφέρθηκαν. Επιπλέον, βλέπουμε ότι σε κάθε καμπύλη αναδεικνύεται καλύτερο και ένα από τα τρία μοντέλα autoencoder. Αθροίζοντας κατά γραμμές τον παραπάνω πίνακα, βλέπουμε ότι τα μοντέλα LSTM, GRU και TCN βρήκαν (κάτω από τις ίδιες συνθήκες) 25, 24 και 22 ανώμαλα σημεία αντίστοιχα. Με βάση αυτούς τους αριθμούς συμπεραίνουμε ότι οριακώς καλυτέρη κατανόηση των δεδομένων δείχνει ο TCN autoencoder. Όπως αναφέρθηκε και στο Κεφάλαιο 5 αξιοσημείωτη είναι η περίπτωση του μηδενός για το μοντέλο GRU στα κρούσματα της Ιταλίας.
Προτείνουμε την παιρετέρω διερεύνηση του συγκεκριμένου ζητήματος και σε αλλά είδη χρονοσειρών ενδεχομένως με λιγότερα σκαμπανεβάσματα. Επιπλέον, η χρήση TCN ως μέσου κωδικοποίησης-αποκωδικποίησης αν και εύκολη, χρίζει ιδιαίτερης πρωτοπορίας. Εφαρμογή αυτού, και άλλων τέτοιων μοντέλων, θα είχαν ενδιαφέρον στην εύρεση ακραίων τιμών σε μετοχές εταιρειών και γενικά χρονοσειρές σχετικές με την οικονομία.

6.1.4 Συμπεράσματα για την εξαγωγή γεωγραφικών χαρακτηριστικών

Στα 2 πειράματα της παραγράφου 5.3.3 στόχος ήταν αφενός να δούμε την αποτελέσματικότητα των μοντέλων σε εντέλως άγνωστα δεδομένα τα οποία προέρχονταν από διαφορετικές χώρες από αυτές που εκπαιδεύθηκαν και αφετέρου να δούμε κάτα πόσο και αν η εκπαίδευση των μοντέλων σε χώρες με θεωρητικά κοινά/διαφορετικά χαράκτηριστικά

(ως προς τα δεδομένα της πανδημίας, τους τρόπους αντιμετώπισης, πολιτικές κλπ) παίζει ρόλο στην απόδοσή τους. Από τα πειράματα αυτά, τα κυριότερα συμπεράσματα που εξήχθησαν ήταν:

- Το προτεινόμενο μοντέλο με τον μηχανισμό Attention ανταποκρίθηκε καλύτερα στην πρόβλεψη της συνολικής καμπύλης κρουσμάτων των υπό εξέταση χωρών σε σχέση με τα TCN, GRU και LSTM. Έδειξε, ότι κατά την εκπαίδευση του σε καμπύλες χωρών του συνόλου **A** (Train set) κράτησε τη σημαντικότερη πληροφορία προκειμένου να εκτελέσει μια πιο αξιόπιστη πρόβλεψη στις χώρες του συνόλου **B** (Test set).

- Σε σχέση με τα πειράματα της ενότητας 5.2 τα απόλυτα σφάλματα των μοντέλων ήταν πολύ μεγαλύτερα με συνέπεια τα τετραγωνικά σφάλματα να μπορούν να λάβουν τιμές ανώτερες του 100%.

- Το σημαντικότερο όμως κέρδος από αυτήν την διαδικασία ήταν ότι καταφέραμε, έστω ως ένα βαθμό, να επαληθεύσουμε, μέσα από την αποτελεσματικότητα των μοντέλων, την αρχική μας πεποίθηση, ότι σε κάποια μέρη του κόσμου η εικόνα της εξέλιξης της πανδημίας παρουσιάζει μια ομοιότητα σε σχέση με άλλα σημεία του πλανήτη. Έτσι, για παράδειγμα, είδαμε ότι τα μοντέλα αποδίδουν καλύτερα όταν πρόκειται να προβλέψουν 2 Ευρωπαϊκές χώρες, έχοντας εκπαιδευθεί σε άλλες χώρες με σχετικά κοντινότερη εικόνα (Πείραμα 1 παργράφου 5.3.3), παρά όταν έχουν εκπαιδευθεί σε χώρες με (θεωρητικώς) διαφορετικότερη εικόνα (Πείραμα 2 παργράφου 5.3.3).

Και εδώ προτείνουμε, τον επιπρόσθετο πειραματισμό σε ακόμα μεγαλύτερο συνδυασμό ομάδων χωρών αλλά και ενδεχομένως την χρήση επιπλέον μοντέλων για την εξαγωγή βαθύτερων συμπερασμάτων.

6.2 Μελλοντικές κατευθύνσεις

Στην ενότητα 6.1, στο τέλος κάθε παραγράφου, αναφέραμε ορισμένες κατευθύνσεις στις οποίες θα μπορούσε να υπάρξει ακόμα πιο ενδελεχής εξέταση των σημείων που ανέλυσε η παρούσα εργασία. Σε αυτήν την ενότητα θα παρουσιάσουμε, εν συντομία, 2 κεντρικούς τρόπους με τους οποίους θεωρήσαμε ότι θα γίνει το επόμενο βήμα στην εξέλιξη της παρούσας ανάλυσης.

6.2.1 Επεκτάσεις ως προς τις τεχνικές διαχείρισης των δεδομένων

Η πρώτη κατεύθυνση αφορά την χρήση επιπρόσθετων και ακόμα ρεαλιστικότερων χαρακατηριστικών στα δεδομένα. Μερικές από τις πιθανές επιλογές είναι οι παρακάτω:

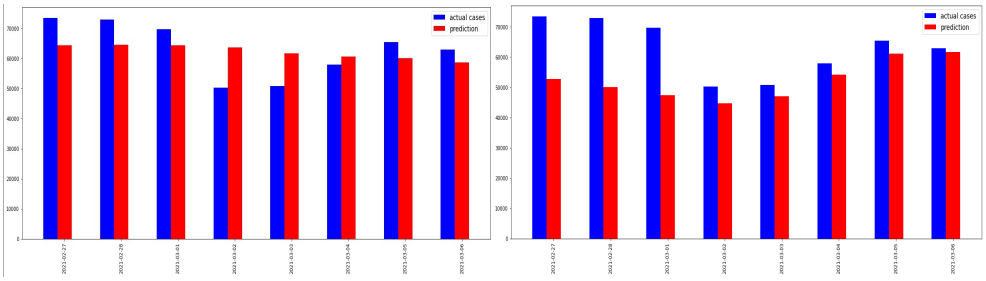
- Προς το τέλος της εργασίας όταν και οι εμβολιασμοί είχαν επεκταθεί σε έναν σημαντικό αριθμό παγκοσμίως διεξήγαμε ένα σύντομο και χωρίς ιδιαίτερη εμβάθυνση πείραμα πρόβλεψης 8 μόλις (λόγω της απουσίας σημαντικού αριθμού εμβολιασμών) ημερών κρουσμάτων για την Ιταλία. Τα σύντομα αποτελέσματα παρουσιάζονται

στην εικόνα 6.1 όπου στην περίπτωση χρήσης εμβολιαστικής μεταβλητής μετρήθηκε απόλυτο σφάλμα, $MAPE = 12.36\%$, ενώ στην απλή (μονομεταβλητή) πρόβλεψη μετρήθηκε $MAPE = 15.53\%$. Το συγκεκριμένο πείραμα αφορούσε μόλις ένα δίκτυο (LSTM), μια χώρα, λίγες ημέρες ενώ δεν έγιναν και περεταίρω της μίας επαναληπτικής μέτρησης. Η παραπάνω εμβάθυνση και τεκμηρίωση μένει προς διερεύνηση.

- Γενικά, η χρήση επιπρόσθετων χαρακτηριστικών είναι ένας τομέας επέκτασης της εργασίας. Στην ανώτερω περίπτωση των εμβολιασμών αναμένουμε αντιστρόφως ανάλογη σχέση της συγκεκριμένης μετάβλητης με την μεταβλήτη πρόβλεψης (κρούσματα/θάνατοι) → Αύξηση εμβολιασμών, μείωση του επιδημικού κύματος.

- Ταυτόχρονα, πολλά προσδοκώμενη είναι η χρήση 'μη συμβατικών' χαρακτηριστικών υπό την έννοια ότι αυτά δεν είναι χρονοσειρές. Τέτοια παραδείγματα, που φαίνεται να παίζουν ρόλο στην διασπορά του ϊού, είναι οι μεταβλητές θερμοκρασίας σε μια συγκεκριμένη περιοχή, οι δείκτες επιβολής περιοριστικών μέτρων και lockdowns στις διάφορες χώρες, η κίνηση γειτονικών πληθυσμών (μέσω ταξιδιών) ή χρήση δεικτών που δηλώνουν την περιοδικότητα διεξαγωγής των τέστ [113][114]. Το τελευταίο από αυτά, προσπαθήσαμε να επιτελέσουμε με την πρόταση και την εφαρμογή της μεθόδου επισήμανσης (Labelling).

- Αντίστοιχα στους θανάτους, παρόμοιες προτάσεις θα μπορούσαν να είναι η χρήση δεικτών που δηλώνουν την γήρανση του πληθυσμού μιας χώρας ή την πίεση του Συστήματος Υγείας της σε μια δεδομένη χρονική στιγμή.



(α′) Απόδοση μοντέλου με χρήση εμβολιαστικής μεταβλητής ($MAPE = 12.36\%$)

(β′) Απόδοση μοντέλου χωρίς χρήση εμβολιαστικής μεταβλητής ($MAPE = 15.53\%$)

Εικόνα 6.1: Διαφαινόμενη βελτίωση των αποδόσεων με χρήση εμβολιαστικής μεταβλητής

6.2.2 Επεκτάσεις ως προς την κατασκευή αρχιτεκτονικών

Η $2^η$ κατεύθυνση επέκτασης της παρούσας εργασίας έχει να κάνει με την υλοποίηση καλύτερων και πολυπλοκότερων αρχιτεκτονικών. Στην παράγραφο 4.3.3 προτάθηκε ένα μοντέλο που χρησιμοποιεί (κατά κάποιον τρόπο) τον μηχανισμό της Προσοχής *από ακολουθία σε ακολουθία* (sequence to sequence - τα μοντέλα αυτά είναι ιδαιτέρως διαδεδομένα στην Επεξεργασία Φυσικής Γλώσσας). Υπάρχουν πολλοί τρόποι με τους

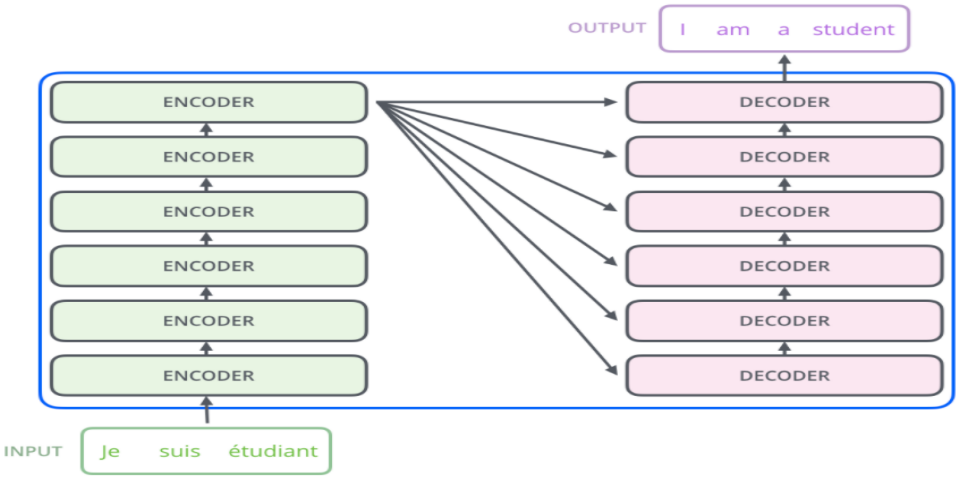
οποίους ένας μηχανισμός προσοσχής μπορεί να ενσωματωθεί σε ένα δίκτυο. Στην εργασία δοκιμάσθηκαν διάφοροι τρόποι με τον αποτελεσματικότερο να εμφανίζεται στην παράγραφο 4.3.3. Ωστόσο, είμαστε υποχρεώμενοι να προτείνουμε και επιπλέον εναλλακτικές χρήσεις του μηχανισμού Attention σε ένα πρόβλημα (πρόβλεψη χρονοσειρών) που ακόμα ο συγκεκριμένος μηχανισμός δεν έχει διερευνηθεί σε τόσο βάθος. Ενδεικτικά προτείνουμε:

- Πιο διευρυμένη χρήση του Local Attention
- Χρήση του Self Attention
- Εναλλακτικούς τρόπους τοποθέτησης επαναληπτικών δικτύων και στρώματος Προσοχής από αυτούς που απεικονίζονται στην εικόνα 4.12 [115].

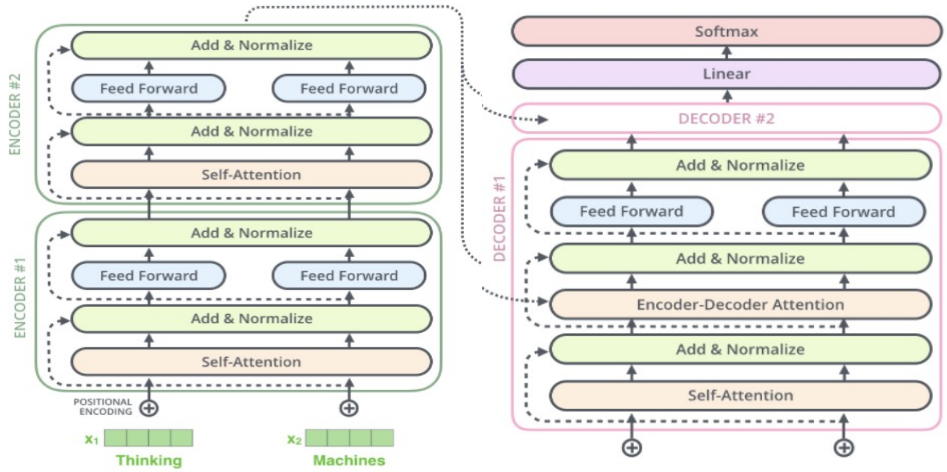
Σε επέκταση των παραπάνω, έρχεται η κατασκευή των πιο σύγχρονων, αποτελεσματικότερων αλλά και συνάμα πολυπλοκότερων δικτύων μηχανικής μάθησης. Αυτά είναι οι λεγόμενοι *Μετασχηματιστές* (Transformers). Πρωτοεμφανίστηκαν το 2017 στο [116] και έκτοτε έχουν κυριαρχίσει ένταντι των RNNs στους τομείς της Επεξεργασία Φυσικής Γλώσσας και Μηχανικής Μετάφρασης [117]. Ταυτόχρονα, από την μικρή τους εφαρμογή σε προβλήματα χρονοσειρών υπάρχει ένδειξη για πρωτοποριακή αντιμετώπιση του ζητήματος αλλά και πρωτοφανή αποτελεσματικότητα [118].

Η δομή ενός Transformer είναι βασισμένη σε μια σειρά κωδικοποιητών και μια σειρά αποκωδικοποιητών που χρησιμοποιούν με ενδεδειγμένο τρόπο τον μηχανισμό της Προσοχής (Self Attention και MultiHead Attention [116]). Η σειρά των κωδικοποιητών έχει το ρόλο της δημιουργίας μίας κωδικοποιημένης αναπαράστασης της πληροφορίας εισόδου, όπου κάθε στοιχείο αυτής θα είναι άμεσα συσχετισμένο με όλα τα υπόλοιπα στοιχεία της ακολουθίας εισόδου. Αυτό επιτυγχάνεται με τη χρήση στρωμάτων Προσοχής που προαναφέραμε αλλά και Τεχνητών Νευρωνικών Δικτύων (Feed Forward Neural Networks). Η τελική κωδικοποίηση της πληροφορίας περνά στην σειρά των αποκωδικοποιητών που είναι αυτή που θα συσχετίσει τις ακολουθίες εξόδου με αυτές τις εισόδου μέσω παρόμοιων μηχανισμών και θα συγκρατήσει όλη την σημαντική πληροφορία κατά την τελική πρόβλεψη η οποία ως γνωστόν θα περάσει από συναρτήσεις ενεργοποίησης. Θα πρέπει να τονισθεί ότι το σύγχρονο μοντέλο TCN, του οποίου έγινε χρήση κατά την παρούσα εργασία, ομοιάζει σε μικρό βαθμό με τις αρχικτεκτονικές των transformer υπό την έννοια ότι έχει πολλαπλές στρώσεις επεξεργασίας των δεδομένων.

Το μειονέκτημα εκπαίδευσης τέτοιων μοντέλων είναι ότι εξαιτίας της μεγάλης τους πολυπλοκότητας απαιτούν τεράστιο όγκο δεδομένων και, άρα, υπολογιστικών πόρων που είναι δύσκολα διαθέσιμοι στο πλαίσιο εκπόνησης μιας διπλωματικής εργασίας. Επίσης, στην εικόνα 6.2 παρουσιάζουμε την τυπική δομή ενός transformer που προέρχεται από το πεδίο της Μηχανικής Μετάφρασης. Τέλος, στις εικόνες 6.3α′ και 6.3β′ βλέπουμε πιο λεπτομερώς τις εσωτερικές δομές των Encoder και Decoder στις οποίες με (έντονες) διακεκομμένες γραμμές τονίζονται οι γνωστές μας υπολειμματικές συνδέσεις (από το μοντέλο TCN) που υπάρχουν και στην αρχιτεκτονική των transformer. Περισσότερες πηροφορίες, για αυτό το πολύ υποσχόμενο μοντέλο, μπορούν να βρεθούν εδώ [119]. Αποτελεί πραγματική πρόκληση διεθνώς η εύρεση τρόπων προσαρμογής αυτών των πολύ δυνατών μοντέλων σε προβλήματα αντιμετώπισης και πρόβλεψης χρονοσειρών.



Εικόνα 6.2: Αναπαράσταση της αρχιτεκτονικής ενός Transformer[119]



(α′) Το εσωτερικό των (πολλαπλών) κωδικοποιητών (Encoder)

(β′) Το εσωτερικό των (πολλαπλών) αποκωδικοποιητών (Decoder)

Εικόνα 6.3: Το εσωτερικό των Encoder και Decoder στην αρχιτεκτονική των Transformer[119]

Βιβλιογραφία

[1] Rolling updates on coronavirus disease (COVID-19).Pneumonia of unknown cause reported to WHO China Office, 31 December 2019 - World Health Organisation, Last access on: 29/5/2021

[2] Rolling updates on coronavirus disease (COVID-19).WHO responding to a cluster of pneumonia cases in Wuhan, 4 January 2020 - World Health Organisation, Last access on: 29/5/2021

[3] WHO Director-General's opening remarks at the media briefing on COVID-19, 11 March 2020 - World Health Organisation, Last access on: 29/5/2021

[4] Amy Qin and Javier C. Hernández, China Reports First Death From New Virus Published Jan. 10, 2020 Updated Jan. 21, 2020 - The New York Times, Last access on: 29/5/2021

[5] Gary Parkinson, Giulia Carbonaro, 03:24, 20-Mar-2020, How COVID-19 is pushing the Italian healthcare system to the brink of collapse - CGTN, Last access on: 29/5/2021

[6] Sam Jones in Madrid, Spain: doctors struggle to cope as 514 die from coronavirus in a day, Tue 24 Mar 2020 16.22 GMT - The Guardian, Last access on: 29/5/2021

[7] How coronavirus broke America's healthcare system - FINANCIAL TIMES, Last access on: 29/5/2021

[8] By Julia Belluz | @juliaoftoronto | Updated Oct 26, 2020, 1:56pm EDT European countries with spiraling Covid-19 outbreaks are shutting back down - Vox, Last access on: 29/5/2021

[9] COVID-19 pandemic in Greece, analytical information - WIKIPEDIA, Last access on: 29/5/2021

[10] Rapid increase of a SARS-CoV-2 variant with multiple spike protein mutations observed in the United Kingdom, 20 December 2020 - European Centre for Disease Prevention and Control (ECDPC), Last access on: 29/5/2021

[11] Coronavirus: Where the vaccine has been rolled out - Deutsche Welle, Last access on: 29/5/2021

[12] Questions and answers on COVID-19: Basic facts - European Centre for Disease Prevention and Control (ECDPC), Last access on: 29/5/2021

[13] Carl Zimmer, Jonathan Corum and Sui-Lee Wee, Coronavirus Vaccine Tracker - The New York Times, Last access on: 29/5/2021

[14] Treatments for COVID-19 - Harvard Health Publishing (Harvard Medical

School), Last access on: 29/5/2021

[15] Dr. A.W. van der Vaart. Time series notes, 1995-2010, Last access on: 29/5/2021

[16] Regression analysis - WIKIPEDIA, Last access on: 29/5/2021

[17] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, UK, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67., Last access on: 29/5/2021

[18] S.L Ho, M Xie, T.N Goh,A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction,Computers & Industrial Engineering, Volume 42, Issues 2–4, 2002, Pages 371-375, ISSN 0360-8352, Last access on: 29/5/2021

[19] WHO Coronavirus Disease (COVID-19) Dashboard - World Health Organisation, Last access on: 29/5/2021

[20] Department of Economic and Social Affairs Population Dynamics, World Population Prospects 2019 - United Nations, Last access on: 29/5/2021

[21] Cameron Appel, Diana Beltekian, Daniel Gavrilov, Charlie Giattino, Joe Hasell, Bobbie Macdonald, Edouard Mathieu, Esteban Ortiz-Ospina, Hannah Ritchie, Max Roser. Data on COVID-19 (coronavirus) - Our World in Data, Last access on: 29/5/2021

[22] The Data, Total for the US, CovidTracking.com Copyright © 2021 by The Atlantic Monthly Group - The COVID Tracking Project, Last access on: 29/5/2021

[23] Statistics and Research, Mortality Risk of COVID-19 - Our World in Data, Last access on: 29/5/2021

[24] Python.org-Welcome to Python, Last access on: 29/5/2021

[25] Matplotlib, Visualization with Python, Last access on: 29/5/2021

[26] Seaborn: statistical data visualization, Last access on: 29/5/2021

[27] folium: Python data, leaflet.js maps, Last access on: 29/5/2021

[28] Geographic data for all countries across the globe, github, Last access on: 29/5/2021

[29] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, https://d2l.ai, pages:87-91, Last access on: 29/5/2021

[30] Gradient Descent, By: IBM Cloud Education, 27 October 2020, Last access on: 29/5/2021

[31] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, page:94, Last access on: 29/5/2021

[32] Luce, R. Duncan. Individual choice behavior: A theoretical analysis. Courier Corporation, 2012., Last access on: 29/5/2021

[33] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, page:108, Last access on: 29/5/2021

[34] M.W Gardner, S.R Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, Atmospheric Environment, Volume 32, Issues 14–15, 1998, Pages 2627-2636, ISSN 1352-2310, Last access on: 29/5/2021

[35] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and

Alexander J. Smola, 2020, page:131, Last access on: 29/5/2021

[36] Rumelhart, David E; Hinton, Geoffrey E, and Williams, Ronald J (Sept. 1985). Learning internal representations by error propagation. Tech. rep. ICS 8504. San Diego, California: Institute for Cognitive Science, University of California., Last access on: 29/5/2021

[37] A. Graves, A. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6645-6649, Last access on: 29/5/2021

[38] V. Di Massa, G. Monfardini, L. Sarti, F. Scarselli, M. Maggini and M. Gori, "A Comparison between Recursive Neural Networks and Graph Neural Networks," The 2006 IEEE International Joint Conference on Neural Network Proceedings, 2006, pp. 778-785, doi: 10.1109/IJCNN.2006.246763., Last access on: 29/5/2021

[39] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, page:323, Last access on: 29/5/2021

[40] P. J. Werbos, "Backpropagation through time: what it does and how to do it," in Proceedings of the IEEE, vol. 78, no. 10, pp. 1550-1560, Oct. 1990, doi: 10.1109/5.58337., Last access on: 29/5/2021

[41] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, page:342, Last access on: 29/5/2021

[42] Kurtis Pykes,The Vanishing/Exploding Gradient Problem in Deep Neural Networks, May 17 2020 - towardsdatascience, Last access on: 29/5/2021

[43] Sepp Hochreiter, Fakult ät für Informatik, Technische Universität München, 80290 München, Germany - Jürgen Schmidhuber, IDSIA, Corso Elvezia 36, 6900 Lugano, Switzerland, Last access on: 29/5/2021

[44] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014)., Last access on: 29/5/2021

[45] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, page:349, Last access on: 29/5/2021

[46] Hecht-Nielsen, Robert. "Theory of the backpropagation neural network." Neural networks for perception. Academic Press, 1992.65-93., Last access on: 29/5/2021

[47] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, page:355, Last access on: 29/5/2021

[48] Pascanu, Razvan, et al. "How to construct deep recurrent neural networks." arXiv preprint arXiv:1312.6026 (2013)., Last access on: 29/5/2021

[49] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, page:360, Last access on: 29/5/2021

[50] Howard, Ronald A. "Dynamic programming and markov processes." (1960) John Wiley., Last access on: 29/5/2021

[51] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, pages:363,365, Last access on: 29/5/2021

[52] Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." IEEE transactions on Signal Processing 45.11 (1997): 2673-2681., Last access on: 29/5/2021

[53] Lin, T., Guo, T., & Aberer, K. (2017). Hybrid neural networks for learning the

trend in time series. In Proceedings of the twenty-sixth international joint conference on artificial intelligence (No. CONF, pp. 2273-2279)., Last access on: 29/5/2021

[54] Sumit Saha, December 15 2018 ,A Comprehensive Guide to Convolutional Neural Networks, the ELI5 way - towardsdatascience, Last access on: 29/5/2021

[55] Sainath, T. N., Mohamed, A. R., Kingsbury, B., & Ramabhadran, B. (2013, May). Deep convolutional neural networks for LVCSR. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 8614-8618). IEEE., Last access on: 29/5/2021

[56] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732)., Last access on: 29/5/2021

[57] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, pages:225-230, Last access on: 29/5/2021

[58] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, page:231, Last access on: 29/5/2021

[59] Translational symmetry - WIKIPEDIA, Last access on: 29/5/2021

[60] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, pages:237-239, Last access on: 29/5/2021

[61] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, page:246, Last access on: 29/5/2021

[62] Hussain, Saddam, Syed Muhammad Anwar, and Muhammad Majid. "Segmentation of glioma tumors in brain using deep convolutional neural network." Neurocomputing 282 (2018): 248-261., Last access on: 29/5/2021

[63] Jason Brownlee on August 21, 2017, CNN Long Short-Term Memory Networks - Machine Learning Mastery, Last access on: 29/5/2021

[64] Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016). Cnn-rnn: A unified framework for multi-label image classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2285-2294)., Last access on: 29/5/2021

[65] Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. arXiv preprint arXiv:1506.04214., Last access on: 29/5/2021

[66] Alexandre Xavier, March 25 2019, An introduction to ConvLSTM - neuronio.ai, Last access on: 29/5/2021

[67] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271., Last access on: 29/5/2021

[68] Francesco Lässig, 28 October 2020, Temporal Convolutional Networks and Forecasting - Unit8, Last access on: 29/5/2021

[69] Oord, A. V. D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499., Last access on: 29/5/2021

[70] Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated

convolutions. arXiv preprint arXiv:1511.07122., Last access on: 29/5/2021

[71] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778)., Last access on: 29/5/2021

[72] Manning, C., & Schutze, H. (1999). Foundations of statistical natural language processing. MIT press., Last access on: 29/5/2021

[73] Chowdhury, G. G. (2003). Natural language processing. Annual review of information science and technology, 37(1), 51-89., Last access on: 29/5/2021

[74] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473., Last access on: 29/5/2021

[75] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. arXiv preprint arXiv:1409.3215., Last access on: 29/5/2021

[76] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[77] Marco Del Pra, November 2 2020,Time Series Forecasting with Deep Learning and Attention Mechanism - towardsdatascience, Last access on: 29/5/2021

[78] Cheng, Jianpeng, Li Dong, and Mirella Lapata. "Long short-term memory-networks for machine reading." arXiv preprint arXiv:1601.06733 (2016)., Last access on: 29/5/2021

[79] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015, June). Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning (pp. 2048-2057). PMLR., Last access on: 29/5/2021

[80] Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025., Last access on: 29/5/2021

[81] Kim, S., & Kang, M. (2019). Financial series prediction using Attention LSTM. arXiv preprint arXiv:1902.10877., Last access on: 29/5/2021

[82] Zhang, Xuan, et al. "AT-LSTM: An attention-based LSTM model for financial time series prediction." IOP Conference Series: Materials Science and Engineering. Vol. 569. No. 5. IOP Publishing, 2019., Last access on: 29/5/2021

[83] TensorFlow, An end-to-end open source machine learning platform, Last access on: 29/5/2021

[84] mxnet, A FLEXIBLE AND EFFICIENT LIBRARY FOR DEEP LEARNING, Last access on: 29/5/2021

[85] PyTorch, FROM RESEARCH TO PRODUCTION, Last access on: 29/5/2021

[86] Keras, The Python Deep Learning API, Last access on: 29/5/2021

[87] Dive into Deep Learning, Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola, 2020, pages:538-540, Last access on: 29/5/2021

[88] Jouppi, Norman P., et al. "In-datacenter performance analysis of a tensor processing unit." Proceedings of the 44th annual international symposium on computer architecture. 2017., Last access on: 29/5/2021

[89] NumPy, Type annotation support - Performance improvements through multi-

platform SIMD, Last access on: 29/5/2021

[90] fchollet, Keras - The Sequential Model, 2020/04/12, Complete guide to the Sequential model., Last access on: 29/5/2021

[91] Victor Roman, January 19 2020, Deep Learning: Introduction to Tensors & TensorFlow - towardsdatascience, Last access on: 29/5/2021

[92] Ho, R. (2013). Handbook of univariate and multivariate data analysis with IBM SPSS. CRC press., Last access on: 29/5/2021

[93] Du Preez, J., & Witt, S. F. (2003). Univariate versus multivariate time series forecasting: an application to international tourism demand. International Journal of Forecasting, 19(3), 435-451., Last access on: 29/5/2021

[94] pandas - fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language., Last access on: 29/5/2021

[95] Shastri, Sourabh, et al. "Time series forecasting of Covid-19 using deep learning models: India - USA comparative case study." Chaos, Solitons & Fractals 140 (2020): 110227., Last access on: 29/5/2021

[96] Arora, P., Kumar, H., & Panigrahi, B. K. (2020). Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India. Chaos, Solitons & Fractals, 139, 110017., Last access on: 29/5/2021

[97] Kırbaş, İsmail, et al. "Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches." Chaos, Solitons & Fractals 138 (2020): 110015., Last access on: 29/5/2021

[98] Pal, Ratnabali, et al. "Neural network based country wise risk prediction of COVID-19." Applied Sciences 10.18 (2020): 6448., Last access on: 29/5/2021

[99] NATIONAL PUBLIC HEALTH ORGANIZATION, Current state of Covid-19 outbreak in Greece and timeline of key containment events, 04/03/2020, Last access on: 29/5/2021

[100] REUTERS, Reuters Stafff, Health News NOVEMBER 25 2020, Turkey announces asymptomatic coronavirus case numbers for first time since July, Last access on: 29/5/2021

[101] scikit-learn, Machine Learning in Python, Last access on: 29/5/2021

[102] Sutcliffe, A. (1994). Time-series forecasting using fractional differencing. Journal of Forecasting, 13(4), 383-393., Last access on: 29/5/2021

[103] Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Last access on: 29/5/2021

[104] Altunkaynak, A., & Nigussie, T. A. (2018). Monthly water demand prediction using wavelet transform, first-order differencing and linear detrending techniques based on multilayer perceptron models. Urban Water Journal, 15(2), 177-181., Last access on: 29/5/2021

[105] Tseng, F. M., Yu, H. C., & Tzeng, G. H. (2002). Combining neural network model with seasonal time series ARIMA model. Technological forecasting and social change, 69(1), 71-87., Last access on: 29/5/2021

[106] Huangwei Wieniawska, June 25 2020, Building Seq2Seq LSTM with Luong

Attention in Keras for Time Series Forecasting, gitconnected, Last access on: 29/5/2021

[107] Du, S., Li, T., Yang, Y., & Horng, S. J. (2020). Multivariate time series forecasting via attention-based encoder–decoder framework. Neurocomputing, 388, 269-279., Last access on: 29/5/2021

[108] Devaraj, Jayanthi, et al. "Forecasting of COVID-19 cases using deep learning models: Is it reliable and practically significant?." Results in Physics 21 (2021): 103817., Last access on: 29/5/2021

[109] Lu, Wenjie, et al. "A CNN-LSTM-Based Model to Forecast Stock Prices." Complexity 2020 (2020)., Last access on: 29/5/2021

[110] Wang, Peipei, et al. "Prediction of epidemic trends in COVID-19 with logistic model and machine learning technics." Chaos, Solitons & Fractals 139 (2020): 110058., Last access on: 29/5/2021

[111] Wang, Peipei, et al. "Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: Case studies in Russia, Peru and Iran." Chaos, Solitons & Fractals 140 (2020): 110214., Last access on: 29/5/2021

[112] An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE, 2(1), 1-18., Last access on: 29/5/2021

[113] Ibrahim, Mohamed R., et al. "Variational-LSTM Autoencoder to forecast the spread of coronavirus across the globe." PloS one 16.1 (2021): e0246120., Last access on: 29/5/2021

[114] Kim, Minseok, et al. "Hi-COVIDNet: Deep Learning Approach to Predict Inbound COVID-19 Patients and Case Study in South Korea." Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020., Last access on: 29/5/2021

[115] Shih, S. Y., Sun, F. K., & Lee, H. Y. (2019). Temporal pattern attention for multivariate time series forecasting. Machine Learning, 108(8), 1421-1441., Last access on: 29/5/2021

[116] Vaswani, Ashish, et al. "Attention is all you need." arXiv preprint arXiv:1706.03762 (2017)., Last access on: 29/5/2021

[117] Karita, Shigeki, et al. "A comparative study on transformer vs rnn in speech applications." 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019., Last access on: 29/5/2021

[118] Wu, Neo, et al. "Deep transformer models for time series forecasting: The influenza prevalence case." arXiv preprint arXiv:2001.08317 (2020)., Last access on: 29/5/2021

[119] Jay Alammar, The Illustrated Transformer, Written on June 27 2018, Last access on: 29/5/2021

[120] The coronavirus SARS-CoV-2, the cause of the COVID-19 pandemic. Encyclopedia Britannica, Inc./Patrick O'Neill Riley, Last access on: 29/5/2021

[121] ALEXANDROS MARAGOS, EERIE STILLNESS: DOCUMENTING ATHENS ON LOCKDOWN, Last access on: 29/5/2021

[122] By Jason Horowitz, Photographs by Fabio Bucciarelli, Published Nov. 29 2020, Updated Feb. 2 2021, THE LOST DAYS THAT MADE BERGAMO A CO-

RONAVIRUS TRAGEDY - New York Times, Last access on: 29/5/2021

[123] An excerpt from Python Data Science Handbook by Jake Vander, Working with Time Series, github:https://github.com/jakevdp/PythonDataScienceHandbook, Last access on: 29/5/2021

[124] Choi, D., Shallue, C. J., Nado, Z., Lee, J., Maddison, C. J., & Dahl, G. E. (2019). On empirical comparisons of optimizers for deep learning. arXiv preprint arXiv:1910.05446., Last access on: 29/5/2021

[125] SEBASTIAN RUDER, Januray 19 2016, An overview of gradient descent optimization algorithms, Last access on: 29/5/2021

[126] Pedregal, D. J., & Young, P. C. (2002). Statistical approaches to modelling and forecasting time series. Companion to economic forecasting, 69-104., Last access on: 29/5/2021

[127] Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. Geoscientific model development, 7(3), 1247-1250., Last access on: 29/5/2021

[128] Kollias, D., Tagaris, A., Stafylopatis, A., Kollias, S., & Tagaris, G. (2018). Deep neural architectures for prediction in healthcare. Complex & Intelligent Systems, 4(2), 119-131.

[129] Tagaris, A., Kollias, D., & Stafylopatis, A. (2017, August). Assessment of parkinson's disease based on deep neural networks. In International Conference on Engineering Applications of Neural Networks (pp. 391-403). Springer, Cham.

[130] Tagaris, A., Kollias, D., Stafylopatis, A., Tagaris, G., & Kollias, S. (2018). Machine learning for neurodegenerative disorder diagnosis—survey of practices and launch of benchmark dataset. International Journal on Artificial Intelligence Tools, 27(03), 1850011.

[131] Kollia, I., Stafylopatis, A. G., & Kollias, S. (2019, July). Predicting Parkinson's disease using latent information extracted from deep neural networks. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[132] Wingate, J., Kollia, I., Bidaut, L., & Kollias, S. (2020). Unified deep learning approach for prediction of Parkinson's disease. IET Image Processing, 14(10), 1980-1989.

[133] Tzouveli, P., Schmidt, A., Schneider, M., Symvonis, A., & Kollias, S. (2008, July). Adaptive reading assistance for the inclusion of students with dyslexia: The AGENT-DYSL approach. In 2008 Eighth IEEE International Conference on Advanced Learning Technologies (pp. 167-171). IEEE.

[134] Wallace, M., Tsapatsoulis, N., & Kollias, S. (2005). Intelligent initialization of resource allocating RBF networks. Neural Networks, 18(2), 117-122.

[135] Kollias, D., Bouas, N., Vlaxos, Y., Brillakis, V., Seferis, M., Kollia, I., ... & Kollias, S. (2020). Deep transparent prediction through latent representation analysis. arXiv preprint arXiv:2009.07044.

[136] Kollias, D., & Zafeiriou, S. P. (2020). Exploiting multi-cnn features in cnn-rnn based dimensional emotion recognition on the omg in-the-wild dataset. IEEE Transactions on Affective Computing.

[137] Yu, M., Kollias, D., Wingate, J., Siriwardena, N., & Kollias, S. (2021). Machine learning for predictive modelling of ambulance calls. Electronics, 10(4),

482.

[138] Kollias, D., & Zafeiriou, S. (2018, July). Training deep neural networks with different datasets in-the-wild: The emotion recognition paradigm. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE., 1-17.

[139] Kollias, D., Yu, M., Tagaris, A., Leontidis, G., Stafylopatis, A., & Kollias, S. (2017). Adaptation and contextualization of deep neural network models. In 2017 IEEE symposium series on computational intelligence (SSCI) (pp. 1-8). IEEE.

[140] Kollias, D., Tagaris, A., & Stafylopatis, A. (2016, December). On line emotion detection using retrainable deep neural networks. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-8). IEEE.

[141] Kollias, D., & Zafeiriou, S. (2021). Affect Analysis in-the-wild: Valence-Arousal, Expressions, Action Units and a Unified Framework. arXiv preprint arXiv:2103.15792.

[142] Kollias, D., Marandianos, G., Raouzaiou, A., & Stafylopatis, A. G. (2015, November). Interweaving deep learning and semantic techniques for emotion analysis in human-machine interaction. In 2015 10th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP) (pp. 1-6). IEEE.

[143] Kollia, I., Simou, N., Stafylopatis, A., & Kollias, S. (2010). Semantic image analysis using a symbolic neural architecture. Image Analysis & Stereology, 29(3), 159-172.

[144] Glimm, B., Kazakov, Y., Kollia, I., & Stamou, G. (2015, February). Lower and upper bounds for SPARQL queries over OWL ontologies. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 29, No. 1).

[145] Glimm, B., Kazakov, Y., Kollia, I., & Stamou, G. B. (2013). Using the TBox to Optimise SPARQL Queries. Description Logics, 1014(181-196), 11.

[146] Kollia, I., Kalantidis, Y., Rapantzikos, K., & Stafylopatis, A. (2012). Improving Semantic Search in Digital Libraries Using Multimedia Analysis. Journal of Multimedia, 7(2).

[147] Kollia, I., Glimm, B., & Horrocks, I. (2011, May). SPARQL query answering over OWL ontologies. In Extended Semantic Web Conference (pp. 382-396). Springer, Berlin, Heidelberg.

[148] Avrithis, Y., Tsapatsoulis, N., & Kollias, S. (2000, July). Broadcast news parsing using visual cues: A robust face detection approach. In 2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532) (Vol. 3, pp. 1469-1472). IEEE.

[149] Rapantzikos, K., Tsapatsoulis, N., Avrithis, Y., & Kollias, S. (2007). Bottom-up spatiotemporal visual attention model for video analysis. IET Image Processing, 1(2), 237-248.

[150] Kollias, D., Cheng, S., Pantic, M., & Zafeiriou, S. (2018). Photorealistic facial synthesis in the dimensional affect space. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops (pp. 0-0).

[151] Kollias, D., Cheng, S., Ververas, E., Kotsia, I., & Zafeiriou, S. (2020). Deep neural network augmentation: Generating faces for affect analysis. International Journal of Computer Vision, 1-30.

Παράρτημα Α΄

Αλγόριθμοι Βελτιστοποίησης (Optimizers)

Οι αλγόριθμοι βελτιστοποίησης είναι πλεόν αναπόσπαστο κομμάτι της μηχανικής μάθησης και αποτελούν τον κατ' εξοχήν τρόπο προκειμένου να βελτιώσουμε τα βάρη ενός νευρωνικού δικτύου, προκείμενου να ελαχιστοποιήσουμε κάποια συνάρτηση λάθους ή να μεγιστοποιήσουμε κάποια συνάρτηση στόχο [124]. Χωρίζονται σε 2 μεγάλες κατηγορίες, τους: α) αλγορίθμους κατάβασης κλίσης και β) τους προσαρμοστικούς αλγορίθμους [125].

Α΄.1 Αλγόριθμοι Κατάβασης Κλίσης (Gradient Descent)

Στους αλγορίθμους κατάβασης κλίσης στόχος είναι η ελαχιστοποίηση μιας αντικειμενικής συνάρτησης $J(\theta)$, όπου με $\theta \in \mathbb{R}^d$ συμβολίζουμε τις παραμέτρους του μοντέλου. Η ελαχιστοποίηση αυτή γίνεται ανανεώνοντας αυτές τις παραμέτρους στην αντίθετη κατεύθυνση από αυτήν της κλίσης της συνάρτηση κόστους $\nabla_\theta J(\theta)$ σε σχέση με το θ. Ο ρυθμός εκπαίδευσης η καθορίζει τα βήματα που κάνουμε σε κάθε επανάληψη εκμάθησης ώστε να φτάσουμε στο (τοπικό) ελάχιστο.

Vanilla (Batch) Gradient Descent

Αποτελεί τον πιο απλό αλγόριθμο βελτιστοποίησης και υπολογίζει τη συνάρτηση κόστους σε σχέση με τις παραμέτρους θ του μοντέλου για το σύνολο εκπαίδευσης, ως εξής:

$$\theta = \theta - \eta \nabla_\theta J(\theta) \tag{Α΄.1.1}$$

Πιο συγκεκριμένα, ο απλός αλγόριθμος κατάβασης κλίσης ακολουθεί την κατεύθυνση της καμπύλης που δημιουργείται από την αντικειμενική συνάρτηση εώς ότου αυτή ελαχιστοποιηθεί. Στις περισσότερες περιπτώσεις όμως καθώς χρείαζεται να υπολογίσουμε και να ανανεώσουμε τα βάρη του μοντέλου διατρέχοντας ολόκληρο το dataset ο αλγόριθμος αυτός αναδεικνύεται ιδιαίτερα αργός και δυσλειτουργικός.

Stochastic Gradient Descent

Από την άλλη πλευρά Stochastic gradient descent (SGD) πραγματοποιεί την ανανέωση των παραμέτρων για κάθε δείγμα εκπαίδευσης $x^{(i)}$ με αντίστοιχη ετικέτα $y^{(i)}$ ως:

$$\theta = \theta - \eta \nabla_\theta J(\theta; x^{(i)}; y^{(i)}) \tag{Α΄.1.2}$$

Καταλαβαίνει κανείς ότι ο SGD γίνεται λίγο πιο περίπλοκος από τον Vanilla Gradient Descent, ωστόσο σε αντίθεση με αυτόν, σε μεγάλα dataset δεν κάνει περιττούς επαναυπολογισμούς στις παραγώγους παρόμοιων παραδειγμάτων πριν από την ανανέωση κάθε παραμέτρου. Ο SGD παραλείπει αυτούς τους υπολογισμούς πραγματοποιώντας μόνον ένα κάθε φορά με συνέπεια να είναι σημαντικά γρηγορότερος. Ωστόσο, λόγου των συχνών ανανεώσεων με υψηλή διακύμανση ο SGD συχνά προκαλεί μεγάλη ταλάντευση της αντικειμενικής συνάρτησης J.
Ακόμα, ενώ ο Batch gradient descent συγκλίνει σε ένα ελάχιστο, ο SGD λόγω αυτών των ταλαντώσεων έχει την δυνατότητα μεταπήδησης σε κάποιο γειτονικό ελάχιστο ενδεχομένως καλύτερο από το προηγούμενο. Έχει δειχθεί ότι ο αλγόριθμος SGD για μικρό ρυθμό εκπαίδευσης συγκλίνει πάντα σε κάποιο τοπικό ή ολικό ελάχιστο.

Α΄.2 Προσαρμοστικοί Αλγόριθμοι (Adaptive Optimizers)

Σε όλους τους αλγορίθμους κατάβασης κλίσης χρείαζεται να θέσουμε εμείς το ρυθμό εκπαίδευσης η του μοντέλου. Κατά τη διάρκεια εκπαίδευσης όμως ο ρυθμός αυτός παραμένει σταθερός, και σε περίπτωση που δεν αποδίδει θα χρειαστεί αλλαγή. Οι προσαρμοστικοί αλγόριθμοι, όπως δηλώνει και το όνομά τους, έχουν την δυνατότητα να προσαρμόζουν το ρυθμό εκπαίδευσης ανάλογα με τις ανάγκες του μοντέλου. Το μόνο που χρειάζεται να δώσει ο χρήστης είναι μια αρχική τιμή, ενώ η λειτουργία του αλγορίθμου διασφαλίζει τη συνεχή ανανέωση του η κατά την διάρκεια εκπαίδευσης. Υπάρχουν διάφοροι αλγόριθμοι τέτοιου είδους όπως οι Adagrad, Adadelta, RMSprop και Adam. Εδώ θα αναλύσουμε τους 2 τελευταίους όπου και χρησιμοποιήσαμε.

RMSprop

Ο αλγόριθμος RMSprop, είναι ένας προσαρμοστικής μάθησης αλγόριθμος, που προτάθηκε από τον Geoff Hinton. Αναπτύχθηκε προκειμένου να αντιμετωπιστεί το βασικό πρόβλημα του αλγορίθμου Adagrad, σχετικά με την εκμηδένιση του ρυθμού εκπαίδευσης κατά την διάρκεια βελτιστοποίησης. Η μαθηματική διατύπωση αυτού του αλγορίθμου μπορεί να αποτυπωθεί στις παρακάτω εξισώσεις.

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2, \tag{Α΄.2.1α΄}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \tag{Α΄.2.1β΄}$$

Στις (Α΄.2.1α΄) και (Α΄.2.1β΄) με g_t συμβολίζουμε την κλίση της αντικειμενικής συνάρτησης J κατά τη χρονική στιγμή t της εκπαίδευσης, με $E[g^2]_t$ συμβολίζουμε τον μέσο

όρο των τετραγώνων των κλίσεων αυτών κατά τη χρονική στιγμή t, ενώ το ϵ είναι μια παράμετρος εξομάλυνσης, ώστε να αποφεύγεται η διαίρεση με το μηδέν και το γ είναι μια παράμετρος που καθορίζει το ρυθμό ανανέωσης των κλίσεων. Βλέπουμε επίσης, ότι σε αντίθεση με πριν, πλέον οι παράμετροι θ του μοντέλου έχουν χρονική εξάρτηση (θ_t). Από την δεύτερη σχέση φαίνεται ότι ο RMSprop διαιρεί το ρυθμό εκπαίδευσης με τον εκθετικά μειούμενο μέσο όρο των τετραγώνων των κλίσεων. Σύμφωνα με τον Geoff Hinton μια καλή επιλογή της παράμετρου γ είναι 0.9, ενώ μια καλή αρχικοποίηση του ρυθμού εκπαίδευσης είναι 0.001.

Adaptive Moment Estimation (Adam)

Στη συντριπτική πλειοψηφεία των πειραμάτων χρησιμοποιήσαμε τον αλγόριθμο βελτιστοποίησης Adam. Πρόκειται για ακόμα έναν προσαρμοστικό αλγόριθμο. Πέρα από την αποθήκευση του εκθετικά μειούμενου μέσου όρου των παρελθοντικών τετραγώνων των κλίσεων, αποθηκεύει και τον εκθετικά μειούμενο μέσον όρο των παρελθοντικών κλίσεων m_t (που αναφέρονται στη στιγμιαία κλίση g_t). Μαθηματικώς, αυτό το διατυπώνουμε ως εξής:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \tag{Α΄.2.2α΄}$$

$$u_t = \beta_2 u_{t-1} + (1-\beta_2) g_t^2 \tag{Α΄.2.2β΄}$$

Τα m_t και u_t καλούνται και εκτιμήσεις πρώτης (μέσης) και δεύτερης (μη κεντραρισμένης απόκλισης) ορμής των κλίσεων αντίστοιχα. Επειδή τα m_t, u_t αρχικοποιούνται ως διανύσματα με μηδενικά, παρατηρείται ότι είναι μεροληπτικά γύρω από την περιοχή του μηδενός και ιδιαίτερα κατά τα πρώτα στάδια της επαίδευσης. Για να αντιμετωπιστεί αυτή η δυσκολία στην βιβλιογραφία προτείνεται η επανεκτίμηση της πρώτης και της δεύτερης ορμής κατά το χρονικό βήμα t, ως εξής:

$$\hat{m_t} = \frac{m_t}{1-\beta_1^t}, \tag{Α΄.2.3α΄}$$

$$\hat{u_t} = \frac{u_t}{1-\beta_2^t} \tag{Α΄.2.3β΄}$$

Στη συνέχεια, χρησιμοποιούμε τις παραπάνω εκτίμησεις των ορμών για την ανανέωση των παραμέτρων θ του μοντέλου, ανάλογα με τον RMSprop:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{u_t}} + \epsilon} \hat{m_t} \tag{Α΄.2.4}$$

Σύμφωνα και πάλι με την βιβλιογραφία κάποιες καλές τιμές για τις παραμέτρους β_1, β_2 είναι 0.9 και 0.999 αντίστοιχα και 10^{-8} για το ϵ. Δηλαδή, οι τιμές των β_1, β_2 πρέπει να είναι κοντά στο 1 πράγμα που μειώνει το ρυθμό της εκθετικής μείωσης των κλίσεων και κατ' επέκταση την διακύμανση των ταλαντώσεων της πορείας προς το ελάχιστο.
Με βάση αυτές τις επιλογές έχει δειχθεί ότι ο Adam λειτουργεί καλά στην πράξη για ένα πολύ μεγάλο εύρος προβλημάτων μηχανικής μάθησης. Αυτό διαπιστώθηκε και κατά την εκπόνηση της εργάσιας σε ένα κατά κύριο λόγο πρόβλημα παλινδρόμησης.

Παράρτημα Β′

Απόδειξη MAE-RMSE

Κατά την διάρκεια των πειραμάτων είδαμε τα σφάλματα $RMSPE$ και $MAPE$. Παρατηρήσαμε ότι $RMSPE > MAPE$ και μάλιστα στην παράγραφο 5.3.3 το δικαιολογήσαμε στη ροή του κειμένου. Πράγματι, για τα σφάλματα MAE και $RMSE$ ισχύει πάντοτε η σχέση:

$$\boxed{RMSE \geq MAE} \tag{Β′.0.1}$$

Η σχέση Β′.0.1 υπάρχει στην διεθνή βιβλιογραφία [127], ωστόσο κάποια απόδειξη δεν υπέπεσε στην προσοχή μας. Αυτό το κεφάλαιο επικεντρώνεται στην μαθηματική απόδειξη αυτής της σχέσης. Στο Κεφάλαιο 5 είδαμε ότι:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{pred}^{(i)} - y_{obs}^{(i)}\right)^2} \quad \text{και} \quad MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_{pred}^{(i)} - y_{obs}^{(i)}\right| \tag{Β′.0.2}$$

Θεωρούμε λοιπόν, 2 διανύσματα παρατηρήσεων και προβλέψεων, $\mathbf{Y}_{obs}$ και $\mathbf{Y}_{pred}$ αντίστοιχα με μήκος $n > 1$ το καθένα. Συνεπώς: $\mathbf{Y}_{obs} = (y_{obs}^{(1)}, y_{obs}^{(2)}, ..., y_{obs}^{(n)})$ και $\mathbf{Y}_{pred} = (y_{pred}^{(1)}, y_{pred}^{(2)}, \ldots, y_{pred}^{(n)})$. Για κάθε παρατήρηση και πρόβλεψη i, ορίζουμε το μεμονωμένο σφάλμα: $u_i = y_{obs}^{(i)} - y_{pred}^{(i)}$ και έτσι το συνολικό διάνυσμα σφάλματος είναι $\mathbf{U} = \mathbf{Y}_{obs} - \mathbf{Y}_{pred} = (u_1, u_2, \ldots, u_n)$.

- Για την απλή περίπτωση όπου $n = 2$ και θεωρώντας ότι η Β′.0.1 είναι σε ισχύ τότε από τις σχέσεις Β′.0.2 παίρνουμε με αντικατάσταση:

  Απόδειξη.

  $$\sqrt{\frac{1}{2}\sum_{i=1}^{2} {u_i}^2} \geq \frac{1}{2}\sum_{i=1}^{2}|u_i| \Leftrightarrow {}^{1}\frac{1}{2}({u_1}^2 + {u_2}^2) \geq \frac{1}{4}(|u_1| + |u_2|)^2 \Leftrightarrow {}^{2}$$
  $$\Leftrightarrow 2{u_1}^2 + 2{u_2}^2 \geq {u_1}^2 + 2|u_1||u_2| + {u_2}^2 \Leftrightarrow {u_1}^2 - 2|u_1||u_2| + {u_2}^2 \geq 0 \Leftrightarrow$$
  $$\Leftrightarrow (|u_1| - |u_2|)^2 \geq 0 \quad \text{που ισχύει } \forall u_1, u_2$$

  □

- Για την περίπτωση όπου $n = 3$ και θεωρώντας και πάλι ότι η Β΄.0.1 είναι σε ισχύ τότε από τις σχέσεις Β΄.0.2 παίρνουμε με αντικατάσταση:

  Απόδειξη.

$$\sqrt{\frac{1}{3}\sum_{i=1}^{3} {u_i}^2} \geq \frac{1}{3}\sum_{i=1}^{3}|u_i| \Leftrightarrow \frac{1}{3}({u_1}^2 + {u_2}^2 + {u_3}^2) \geq \frac{1}{9}(|u_1| + |u_2| + |u_3|)^2 \Leftrightarrow$$
$$\Leftrightarrow 3{u_1}^2 + 3{u_2}^2 + 3{u_3}^2 \geq {u_1}^2 + {u_2}^2 + {u_3}^2 + 2|u_1||u_2| + 2|u_2||u_3| + 2|u_1||u_3| \Leftrightarrow$$
$$\Leftrightarrow 2{u_1}^2 + 2{u_2}^2 + 2{u_3}^2 - 2|u_1||u_2| - 2|u_2||u_3| - 2|u_1||u_3| \geq 0 \Leftrightarrow$$
$$\Leftrightarrow (|u_1| - |u_2|)^2 + (|u_2| - |u_3|)^2 + (|u_1| - |u_3|)^2 \geq 0 \quad \text{που ισχύει } \forall u_1, u_2, u_3$$

□

- Για την γενική περίπτωση όπου $n = m > 3$ ακολουθώντας εκ νέου την ίδια διαδικασία γνωρίζουμε ότι στην 3^η γραμμή της ακριβώς παραπάνω απόδειξης θα έχουμε: α) $(m-1)$ φορές, κάθε όρο της μορφής ${u_i}^2$ για $i \in [1, m]$, β) $\binom{m}{2} = \frac{m!}{2!(m-2)!} = \frac{m!}{2(m-2)!}$ ζεύγη της μορφής $-2|u_i||u_j|$, με $i \neq j$, όπου $i, j \in [1, m]$ και γ) θα υπάρχουν m όροι της μορφής $(m-1){u_i}^2$.
  Σύμφωνα με τα α) και γ) παραπάνω θα υπάρχουν ζευγάρια της μορφής $(|u_i|, |u_j|)$ με $i \neq j$ το πλήθος των οποίων θα είναι $\frac{m(m-1)}{2}$ (A). Από το β) παίρνουμε ότι οι όροι της μορφής $-2|u_i||u_j|$ θα είναι σε πλήθος $\frac{m!}{2(m-2)!} = \frac{m(m-1)(m-2)!}{2(m-2)!} = \frac{m(m-1)}{2}$ (B). Από τις (A) και (B) αποδεικνύεται ότι εν τέλει είναι δυνατή η κατασκευή $\frac{m(m-1)}{2}$ τέλειων τετραγώνων της μορφής $(|u_i| - |u_j|)^2$ με $i \neq j$.

Αναλυτικά θα έχουμε:

Απόδειξη.

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m} {u_i}^2} \geq \frac{1}{m}\sum_{i=1}^{m}|u_i| \Leftrightarrow \frac{1}{m}({u_1}^2 + \ldots + {u_m}^2) \geq \frac{1}{m^2}(|u_1| + \ldots + |u_m|)^2 \Leftrightarrow$$
$$\Leftrightarrow m{u_1}^2 + \ldots + m{u_m}^2 \geq {u_1}^2 + \ldots + {u_m}^2 + 2|u_1||u_2| + 2|u_1||u_3| + \ldots + 2|u_1||u_m| +$$
$$+2|u_2||u_3| + \ldots + 2|u_2||u_m| + \ldots + 2|u_{m-1}||u_m| \Leftrightarrow (m-1){u_1}^2 + \ldots + (m-1){u_m}^2 -$$
$$-2|u_1||u_2| - 2|u_1||u_3| - \ldots - 2|u_1||u_m| - 2|u_2||u_3| - \ldots - 2|u_2||u_m| - \ldots - 2|u_{m-1}||u_m| \Leftrightarrow$$
$$\Leftrightarrow (|u_1| - |u_2|)^2 + (|u_1| - |u_3|)^2 + \ldots + (|u_1| - |u_m|)^2 + (|u_2| - |u_3|)^2 + \ldots$$
$$+(|u_2| - |u_m|)^2 + \ldots + (|u_{m-1}| - |u_m|)^2 \geq 0 \quad \text{που ισχύει } \forall u_1, u_2, \ldots, u_m$$

□

[1] ισχύει πάντα ότι $RMSE \geq 0$ και $MAE \geq 0$
[2] ισχύει ότι $|x|^2 = x^2, \quad \forall x$