ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Σχολή Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Πρόβλεψη της Δευτεροταγούς Δομής Πρωτεϊνών με τεχνικές Μηχανικής Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΧΡΥΣΟΥΛΑ Χ. ΚΟΣΜΑ

Επιβλέπων : Γεώργιος Στάμου
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2019

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Πρόβλεψη της Δευτεροταγούς Δομής Πρωτεϊνών με τεχνικές Μηχανικής Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΧΡΥΣΟΥΛΑ Χ. ΚΟΣΜΑ

Επιβλέπων : Γεώργιος Στάμου
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12η Νοεμβρίου 2019.

.....	.....	.....
Γεώργιος Στάμου	Γεώργιος-Ανδρέας Σταφυλοπάτης	Δημήτριος Φωτάκης
Αν. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2019

.....

ΧΡΥΣΟΥΛΑ ΚΟΣΜΑ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η βιοπληροφορική είναι το επιστημονικό πεδίο της ανάλυσης βιολογικών δεδομένων. Τα βιολογικά δεδομένα ποικίλλουν, από ακολουθίες DNA/RNA, οι οποίες μοντελοποιούνται ως ακολουθίες χαρακτήρων (αποτελούμενες από τέσσερις διαφορετικούς χαρακτήρες A, G, C, T) στην περιγραφή της δομής πρωτεϊνών και τις ταξινομήσεις διαφορετικών οργανισμών. Μια κοινή προσέγγιση για την ανάλυση αυτών των δεδομένων είναι η εξαντλητική μοντελοποίηση ή η στατιστική ανάλυση τους. Η προσέγγιση της στατιστικής ανάλυσης είναι ένας αποτελεσματικός τρόπος σε αυτά τα προβλήματα, καθώς η πολυπλοκότητα των βιολογικών συστημάτων, που επηρεάζουν τα βιολογικά δεδομένα, είναι υψηλή και πρέπει να ληφθούν υπόψη όλες οι πιθανές αλληλεπιδράσεις μεταξύ των υποσυστημάτων τους. Σε αυτή την κατεύθυνση, τα τελευταία χρόνια, έχουν πραγματοποιηθεί αρκετές εργασίες που προσπαθούν να αναλύσουν βιολογικά δεδομένα με χρήση μηχανικής μάθησης, αποδεικνύοντας ότι τα υπάρχοντα πρότυπα βιολογικών ακολουθιών μπορούν να μοντελοποιηθούν αποτελεσματικά.

Ανάμεσα στα πιο γνωστά προβλήματα βιολογικών ακολουθιών είναι το πρόβλημα της Πρόβλεψης της Δευτεροταγούς Δομής των Πρωτεϊνών, το οποίο στοχεύει στη απεικόνιση ακολουθιών πρωτεϊνών (αποτελούμενων από 22 διακριτούς χαρακτήρες) στις αντίστοιχες ακολουθίες της δευτεροταγούς δομής τους (η οποία συνήθως αποτελείται από 3 ή 8 κατηγορίες χαρακτήρων, που ορίζουν αντίστοιχα τις κωδικοποιήσεις Q3 και Q8). Σε αυτή την εργασία, το δύσκολο πρόβλημα της Q8 κωδικοποίησης της Δευτεροταγούς Δομής των Πρωτεϊνών εξετάζεται διεξοδικά. Οι πιο επιτυχημένες αρχιτεκτονικές που έχουν εφαρμοστεί στο πρόβλημα αυτό έχουν επιτύχει μια ακρίβεια ~71%, χρησιμοποιώντας μια ποικιλία μοντέλων, όπως βαθιά Συνελικτικά Νευρωνικά Δίκτυα, Επαναλαμβανόμενα Νευρωνικά Δίκτυα και μηχανισμούς Προσοχής, καθώς και συνδυασμούς των διαφόρων αρχιτεκτονικών.

Δεδομένου ότι το πρόβλημα Πρόβλεψης της Δευτεροταγούς Δομής των Πρωτεϊνών είναι ένα πρόβλημα πρόβλεψης ακολουθίας από ακολουθία, με τις ακολουθίες να αποτελούνται από χαρακτήρες, τα μοντέλα Επεξεργασίας Φυσικής Γλώσσας μπορούν να εφαρμοστούν στα δεδομένα και να τα χειριστούν ως ακολουθίες κειμένου. Σε αυτό το πλαίσιο, το πρόβλημα μπορεί να θεωρηθεί ως μια εργασία Μηχανικής Μετάφρασης από μια γλώσσα (αποτελούμενη από 22 χαρακτήρες για τα υπολείμματα των πρωτεϊνών) σε άλλη (αποτελούμενη από 8 διαφορετικούς χαρακτήρες που ορίζουν την ακολουθία της δευτεροταγούς δομής). Το μοντέλο με την μεγαλύτερη ακρίβεια στη Μηχανική Μετάφραση κειμένου είναι το μοντέλο του Μεταφραστή (Transformer), το οποίο και εφαρμόζεται σε αυτή την εργασία στο πρόβλημα Πρόβλεψης της Δευτεροταγούς Δομής των Πρωτεϊνών. Το μοντέλο αυτό επιτυγχάνει μια αξιοπρεπή ακρίβεια ~64.4% μετά από βασική ρύθμιση των υπερπαραμέτρων του και με τη χρήση ενός λεξιλογίου που αποτελείται από λέξεις ενός χαρακτήρα σε κάθε ακολουθία. Περαιτέρω βελτιώσεις σε αυτήν την αρχιτεκτονική, όπως πειράματα με διαφορετικά λεξιλόγια (με την εξαγωγή n-χαρακτήρων από τις ακολουθίες και τη χρήση τους ως λέξεις) ή χρήση προ-εκπαιδευμένων ενσωματώσεων από μεγαλύτερα σύνολα δεδομένων πρωτεϊνικών ακολουθιών, ενδεχομένως να επιτύχουν μεγαλύτερη ακρίβεια σε αυτό το πρόβλημα και να αναδείξουν το συνολικό μοτίβο της δομής των πρωτεϊνών.

Λέξεις Κλειδιά: Μοντελοποίηση Βιολογικών Ακολουθιών, Πρωτεϊνώματα, Πρόβλεψη της Δευτεροταγούς Δομής των Πρωτεϊνών, Μηχανική Μετάφραση, Συνελικτικά Νευρωνικά Δίκτυα, Επαναλαμβανόμενα Νευρωνικά Δίκτυα, Μηχανισμοί Προσοχής, Αρχιτεκτονικές Κωδικοποιητή-Αποκωδικοποιητή, Μοντέλα Ακολουθίας σε Ακολουθία, Επεξεργασία Φυσικής Γλώσσας.

Abstract

Bioinformatics is the scientific field of analyzing biological data. Biological data vary from DNA/RNA sequences, which can be modelled as character sequences (consisting of four distinct characters A, G, C, T) to sequences describing the protein structure and the taxonomies of different organisms. A common approach to analyze these data is by extensive modelling or by statistical analysis. The approach of statistical analysis is an effective way in these problems since the complexity of the biological systems affecting biological data is high and all the possible interactions between subsystems should be examined. In this direction, in the last years, several works that analyze biological data using machine learning (ML) have been applied, demonstrating that the existing patterns of biological sequences can be effectively modelled.

Among biological sequences' most well-known problems, lies the Protein Secondary Structure Prediction problem (PSSP), which aims to map sequences of proteins (consisting of 22 distinct characters) to their corresponding sequences of secondary structure (which is usually modelled by 3 or 8 classes of characters, defining the Q3 and Q8 encodings respectively). In this work, the more challenging Q8 class problem is thoroughly examined. State-of-the-art architectures have achieved an accuracy of ~71%, using a variety of models, consisting of deep CNNs, RNNs and attention layers and ensemble techniques.

Since the PSSP problem is a sequence-to-sequence problem, where sequences consist of characters, Natural Language Processing models can be applied to the data and handle them as text sequences. In these terms, the PSSP task can be considered a Machine translation task from one language (consisting of 22 characters for protein residues) to another (consisting of 8 different characters that define the sequence of secondary structure). The state-of-the-art model in Machine Translation, Transformer, is applied in this work to the PSSP problem proving to achieve a decent accuracy of ~64.4% with basic parameter tuning and a vocabulary consisting of 1-grams as words. Further improvements in this architecture, including experiments with different vocabularies (n-grams extraction from sequences) or the use of pretrained embeddings from larger protein datasets, are promising for achieving a higher accuracy on this task and for unravelling the unique context of protein sequences' structure.

Key Words: Biological Sequences Modelling, Proteomics, Protein Secondary Structure Q8, Machine Learning (ML), CNNs, RNNs, attention mechanism, Encoding-Decoding Neural Network Architectures, Sequence-to-sequence models, Natural Language Processing (NLP), Machine Translation, Transformer.

Ευχαριστίες

Καθοριστική στην εκπόνηση της συγκεκριμένης εργασίας ήταν η συνεισφορά πολλών ανθρώπων τους οποίους θα ήθελα να ευχαριστήσω σε αυτό το σημείο. Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής μου εργασίας, κύριο Γιώργο Στάμου, ο οποίος καθ' όλη τη διάρκεια της στάθηκε στο πλευρό μου παρέχοντας μου πολύτιμη καθοδήγηση, συμβουλές και έμπνευση για πειραματισμό στον τομέα της Επιστήμης Δεδομένων.

Ταυτόχρονα, θα ήθελα να ευχαριστήσω τον Έντμοντ Ντερβάκο και τον Αντώνη Κακολύρη για την ανταλλαγή γνώσεων, τις συμβουλές σε επίπεδο υλοποίησης και στην οργάνωση της εργασίας και των πειραμάτων. Μαζί με αυτούς, στην Ομάδα "Genomics and Deep Learning" του εργαστηρίου Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης της Σχολής, θα ήθελα να ευχαριστήσω την Νατάσα Σοφού και τον Βασίλη Τζουβάρα, για την οργάνωση των συναντήσεων, την παροχή γνώσεων σε αυτές και την καθοδήγηση για την πραγματοποίηση της εργασίας. Ήμουν πολύ τυχερή που είχα την ευκαιρία να μελετήσω τα επιμέρους θέματα της εργασίας στα πλαίσια μιας ερευνητικής ομάδας και να μάθω από την εμπειρία αυτών των ανθρώπων.

Επίσης, θα ήθελα να ευχαριστήσω τους καλούς φίλους (ξέρουν αυτοί) που στάθηκαν στο πλευρό μου και μου έδιναν θάρρος και τη θετική τους ενέργεια κατά τη διάρκεια της εργασίας, ιδιαίτερα στις στιγμές που χρειαζόμουν τη στήριξή τους. Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου και την αδερφή μου, που είναι πάντα δίπλα μου σε κάθε μου προσπάθεια.

Χρυσούλα Κοσμά

Πίνακας περιεχομένων

Κατάλογος Πινάκων

Κατάλογος Σχημάτων

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρο

Η μελέτη βιολογικών ακολουθιών (όπως ακολουθίες DNA, RNA και πρωτεϊνών) αποτελεί διαχρονικά έναν χώρο ακόρεστου επιστημονικού ενδιαφέροντος, που προσελκύει επιστήμονες από διαφορετικούς επιστημονικούς χώρους, όπως βιολόγους, γιατρούς, βιοφυσικούς, επιστήμονες που ασχολούνται με τη βιοπληροφορική κ.α. Στόχος είναι η ανακάλυψη των διαφόρων μοτίβων που παρουσιάζονται σε αυτές τις ακολουθίες και η επίδραση αυτών των χαρακτηριστικών στον βιολογικό τους ρόλο. Η βιοπληροφορική, η μελέτη των γονιδιωμάτων και των πρωτεϊνωμάτων προωθούν πεδία που ενοποιούν τα εργαλεία και τις γνώσεις από τη βιολογία, τη χημεία, την επιστήμη των υπολογιστών, τα μαθηματικά, τη φυσική και τη στατιστική, στην έρευνα και στον συνδυασμό των βιολογικών και πληροφοριακών επιστημών. Εμπνευσμένα από την μεγάλη ποσότητα βιολογικών ακολουθιών που είναι διαθέσιμη στις μέρες μας, αυτά τα νέα πεδία επιδιώκουν να θέσουν ερωτήματα και να δώσουν απαντήσεις σε βιολογικά ζητήματα, που ανέκαθεν θεωρούνταν πολύ περίπλοκα στην μελέτη τους. Η δυσκολία επεξεργασίας αυτών των δεδομένων, έγκειται στην διαχείριση ενός μεγάλου όγκου δεδομένων και μεγάλων σε μέγεθος ακολουθιών δεδομένων. Στα πλαίσια αυτά, οι μέθοδοι Μηχανικής Μάθησης, που γνωρίζουν σήμερα μεγάλη επιτυχία, χάρη σε αλγόριθμους που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν ακριβείς προβλέψεις σε αυτά, παρουσιάζουν τα τελευταία χρόνια μεγάλα ποσοστά επιτυχίας σε δύσκολα προβλήματα που περιλαμβάνουν βιολογικά δεδομένα, συμπεριλαμβανομένων προβλημάτων διαχείρισης μεγάλων ακολουθιών.

1.2 Αντικείμενο διπλωματικής

Ένα από τα πιο γνωστά προβλήματα στον τομέα ανάλυσης των πρωτεϊνωμάτων, είναι το πρόβλημα πρόβλεψης της Δευτεροταγούς Δομής των Πρωτεϊνών, η οποία αναπαρίσταται από ακολουθίες με χαρακτήρες, καθένας εκ των οποίων μπορεί να ανήκει σε 8 κλάσεις (Q8 κωδικοποίηση). Η πρόβλεψη της δομής αυτής γίνεται μέσω της αναζήτησης μοτίβων σε ακολουθίες πρωτεϊνών, που αποτελούνται από χαρακτήρες που μπορεί να ανήκουν σε 22 κλάσεις (όσα και τα αμινοξέα). Η μελέτη και η επίλυση αυτού του προβλήματος με τεχνικές

μηχανικής μάθησης γνωρίζει εδώ και δεκαετίες μεγάλο επιστημονικό ενδιαφέρον, μιας και στα πλέον διαδεδομένα σύνολα δεδομένων που έχουν χρησιμοποιηθεί για το πρόβλημα (CB6133 σύνολο δεδομένων εκπαίδευσης και CB513 σύνολο δεδομένων ελέγχου), δεν έχει επιτευχθεί ακρίβεια μεγαλύτερη της τάξης του ~71%, ενώ έχουν χρησιμοποιηθεί πολλοί συνδυασμοί νευρωνικών δικτύων με στόχο τη δημιουργία βαθιών δικτύων, για να αναγνωρίσουν τα μοτίβα σε ακολουθίες μεγάλου μήκους των συνόλων δεδομένων (έως 700 αμινοξέα). Συνεπώς, η δυσκολία του προβλήματος συνοψίζεται στα εξής σημεία:

1. Η πρόβλεψη ακολουθιών εξόδου από ακολουθίες εισόδου (sequence to sequence problem), που παρουσιάζουν διαφορετικά μήκη, εκ των οποίων πολλά είναι αρκετά μεγάλα (έως και 700 χαρακτήρες).
2. Το διαφορετικό πλήθος κλάσεων που ανήκουν οι χαρακτήρες των ακολουθιών εισόδου και εξόδου (22 και 8 αντίστοιχα).
3. Το σχετικά μικρό μέγεθος του συνόλου εκπαίδευσης (~5500 ακολουθίες) σε σχέση με τα μεγάλα μήκη των ακολουθιών που πρέπει να επεξεργαστούν από τα μοντέλα μηχανικής μάθησης.
4. Το γεγονός ότι οι ακολουθίες χαρακτήρων περιέχουν περισσότερες εμφανίσεις κάποιων κλάσεων έναντι κάποιων άλλων (imbalanced data) επίσης δυσκολεύουν την αποδοτικότητα της πρόβλεψης των μοντέλων μηχανικής μάθησης.

Το αντικείμενο μελέτης της συγκεκριμένης εργασίας είναι η αντιμετώπιση του προβλήματος με τη χρήση κλασσικών τεχνικών μετάφρασης κειμένου και πρόβλεψης ακολουθιών από ακολουθίες. Η πρόκληση της συγκεκριμένης προσέγγισης είναι οι διαφορές που παρουσιάζονται στα σύνολα δεδομένων κειμένου φυσικής γλώσσας και στα δεδομένα των πρωτεϊνικών ακολουθιών και της δομής τους. Ενώ στα πρώτα σύνολα μπορεί να εμφανίζονται χιλιάδες διαφορετικές λέξεις, το μέγεθος των προτάσεων είναι σχετικά μικρό (με τις μεγαλύτερες προτάσεις να παρουσιάζουν περίπου 20 λέξεις). Αντιθέτως στις ακολουθίες πρωτεϊνών, θεωρώντας ως λέξεις τους χαρακτήρες που περιέχουν, παρουσιάζονται 22 διαφορετικές κλάσεις στην είσοδο και μόνο 8 στην έξοδο, ενώ το μήκος των προτάσεων είναι αισθητά μεγαλύτερο, όπως αναφέραμε παραπάνω.

1.2.1 Συνεισφορά

Για την αντιμετώπιση του προβλήματος Πρόβλεψης της Δευτεροταγούς Δομής ακολουθιών πρωτεϊνών (ΠΔΔΠ) με κωδικοποίηση Q8, στα πλαίσια της συγκεκριμένης εργασίας, υλοποιήθηκε το πλέον σύγχρονο και αποδοτικό μοντέλο στον τομέα της Μηχανικής Μετάφρασης, το μοντέλο του Μεταφραστή (Transformer). Το μοντέλο αυτό, σε αντίθεση με τις υπόλοιπες εργασίες που χρησιμοποιούν συνδυασμούς Συνελικτικών Νευρωνικών Δικτύων

και Επαναλαμβανόμενων Νευρωνικών Δικτύων, δεν έχει δοκιμαστεί ξανά στο συγκεκριμένο πρόβλημα και παρουσιάζει μια καινοτόμα αρχιτεκτονική που το καθιστά ιδιαίτερα αποδοτικό και ακριβές σε κλασσικές εργασίες μηχανικής μετάφρασης σε μεγάλο όγκο δεδομένων κειμένου. Έτσι στη συγκεκριμένη εργασία:

1. Γίνεται μια ανασκόπηση διαφόρων τεχνικών και μοντέλων πρόβλεψης ακολουθιών από ακολουθίες εισόδου και μηχανικής μετάφρασης και αναλύεται η αποδοτικότητα αυτών.
2. Συνοψίζονται και επεξηγούνται οι πιο επιτυχείς αρχιτεκτονικές νευρωνικών δικτύων στο πρόβλημα ΠΔΔΠ.
3. Αναλύονται όλες οι τεχνικές λεπτομέρειες υλοποίησης του μοντέλου του Μεταφραστή (Transformer) για το πρόβλημα ΠΔΔΠ και επεξηγείται πώς μπορεί να πραγματοποιηθεί μετάφραση μεταξύ συνόλων βιολογικών ακολουθιών (εξαγωγή λέξεων, δημιουργία λεξικού).
4. Παρατίθενται και επεξηγούνται τα αποτελέσματα πολλών πειραμάτων της αρχιτεκτονικής του Μεταφραστή (Transformer) στο συγκεκριμένο πρόβλημα.
5. Παρατίθενται προτάσεις για βελτίωση της απόδοσης του συγκεκριμένου μοντέλου.

Συνεπώς, η συγκεκριμένη εργασία μπορεί να δώσει κίνητρο στην μεταφορά γνώσης από τις περιοχές της Επεξεργασίας Φυσικής Γλώσσας και της Μηχανικής Μετάφρασης στην διαχείριση βιολογικών ακολουθιών ως ακολουθίες κειμένου. Επίσης, δίνει επαρκή πληροφόρηση για την απόδοση του μοντέλου του Μεταφραστή (Transformer) στο προς εξέταση πρόβλημα και τους περιορισμούς αυτής της αρχιτεκτονικής, ενισχύοντας τον πειραματισμό για την επίτευξη μεγαλύτερης ακρίβειας στο πρόβλημα πρόβλεψης.

1.3 Οργάνωση κειμένου

Στην ενότητα αυτή θα αναφέρουμε συνοπτικά τι πραγματεύονται τα κεφάλαιο της εργασίας που ακολουθούν:

- Το Κεφάλαιο 2 κάνει μια εισαγωγή στις έννοιες της Επιβλεπόμενης Μηχανικής Μάθησης και στις βασικότερες αρχιτεκτονικές Νευρωνικών Δικτύων που γνωρίζουν μεγάλη εφαρμογή σήμερα. Ιδιαίτερη έμφαση δίνεται σε μοντέλα επεξεργασίας και διαχείρισης ακολουθιών, με λεπτομέρειες για την αρχιτεκτονική τους και πιθανές δυσκολίες και περιορισμούς που αυτά παρουσιάζουν.
- Το Κεφάλαιο 3 εισάγει τον αναγνώστη στα προβλήματα Μηχανικής Μετάφρασης. Δίνει μια σύνοψη των διαφόρων αρχιτεκτονικών που έχουν υλοποιηθεί στην περιοχή και εμβαθύνει στις λεπτομέρειες του μοντέλου του Μεταφραστή (Transformer).

- Το Κεφάλαιο 4 παρουσιάζει το πρόβλημα πρόβλεψης της Δευτεροταγούς Δομής Πρωτεϊνών, από βιολογική σκοπιά και ως πρόβλημα μηχανικής μάθησης. Παρουσιάζονται τα πιο διαδεδομένα σύνολα δεδομένων και η μετρική απόδοσης του προβλήματος πρόβλεψης. Επίσης, παρουσιάζονται τα πιο σύγχρονα και αποτελεσματικά μοντέλα που έχουν υλοποιηθεί για το πρόβλημα.
- Το Κεφάλαιο 5 παρέχει τις τεχνικές λεπτομέρειες υλοποίησης και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για τη δημιουργία και την εφαρμογή του μοντέλου του Μεταφραστή (Transformer) στο προς εξέταση πρόβλημα.
- Το Κεφάλαιο 6 παρουσιάζει τα πειράματα που διεξήχθησαν στο παραπάνω μοντέλο και τα αποτελέσματα αυτών των πειραμάτων καθώς και τα συμπεράσματα που προκύπτουν από αυτά.
- Το Κεφάλαιο 7 συνοψίζει τα διάφορα συμπεράσματα της εργασίας και παρουσιάζει προτάσεις για μελλοντικές επεκτάσεις στο μοντέλο του Μεταφραστή (Transformer) για το συγκεκριμένο πρόβλημα.

Κεφάλαιο 2

Εισαγωγή στα Προβλήματα Πρόβλεψης Ακολουθιών

Σε αυτό το κεφάλαιο γίνεται μια εισαγωγή στην Μηχανική Μάθηση με Επίβλεψη και ειδικότερα στα προβλήματα πρόβλεψης ακολουθιών εξόδου από ακολουθίες δεδομένων εισόδου. Επιπλέον, παρουσιάζονται οι Βασικές Αρχιτεκτονικές Νευρωνικών Δικτύων που εφαρμόζονται σε αυτή την περιοχή και στις οποίες θα αναφερθούμε εκτενώς στα επόμενα κεφάλαια, δίνοντας λεπτομέρειες για την υλοποίηση και την απόδοση τους στο αντικείμενο που μελετά η παρούσα εργασία.

2.1 Αρχές Επιβλεπόμενης Μάθησης

Ένας αλγόριθμος Μηχανικής Μάθησης (Machine Learning Algorithm) είναι ένας αλγόριθμος που μπορεί να μάθει από τα δεδομένα (data). Ένας συνοπτικός ορισμός για την έννοια της μάθησης δόθηκε από τον Mitchell (1997): "Ένα πρόγραμμα ηλεκτρονικών υπολογιστών λέγεται ότι μαθαίνει από την εμπειρία E σε σχέση με κάποια κλάση εργασίας T και ένα μέτρο επίδοσης P, αν η απόδοσή του στις εργασίες T, όπως αυτή μετριέται βάσει του P, βελτιώνεται με την εμπειρία E." (Mitchell, 1997)

Στη συνέχεια, θα αναφερθούμε συνοπτικά στις έννοιες της εργασίας T, του μέτρου επίδοσης P και της εμπειρίας E, όπως αυτές παρατίθενται στο βιβλίο (Ian Goodfellow, 2016, pp. 97-103).

Η Εργασία (Task), T

Οι εργασίες μηχανικής μάθησης περιγράφονται συνήθως με βάση τον τρόπο με τον οποίο το σύστημα μηχανικής μάθησης επεξεργάζεται ένα παράδειγμα. Ένα παράδειγμα είναι μια συλλογή χαρακτηριστικών που έχουν υπολογιστεί ποσοτικά από κάποιο αντικείμενο ή γεγονός που το σύστημα μηχανικής μάθησης καλείται να επεξεργαστεί. Συνήθως αντιπροσωπεύουμε ένα παράδειγμα με ένα διάνυσμα $x \in R^n$, όπου κάθε είσοδος x_i του διανύσματος είναι ένα άλλο χαρακτηριστικό. Για παράδειγμα, τα χαρακτηριστικά μιας εικόνας είναι συνήθως οι τιμές των εικονοστοιχείων (pixels) της εικόνας.

Ενδεικτικά, θα παρουσιάσουμε στη συνέχεια τρία βασικά είδη εργασιών, όπου εφαρμόζονται αλγόριθμοι μηχανικής μάθησης και στα οποία θα αναφερθούμε σε επόμενα κεφάλαια.

- **Ταξινόμηση (Classification):** Σε αυτό το είδος εργασίας, το πρόγραμμα καλείται να καθορίσει σε ποιες από τις k κατηγορίες ανήκει κάθε είσοδος. Για να πραγματοποιηθεί αυτή η εργασία, ο αλγόριθμος μηχανικής μάθησης καλείται να παράγει μια συνάρτηση

$f: R^n \rightarrow \{1, \ldots, k\}$. Όταν $y = f(x)$ το μοντέλο αντιστοιχεί μια είσοδο που περιγράφεται από τον διάνυσμα εισόδου x σε μια κατηγορία που έχει ετικέτα y. Υπάρχουν επίσης, παραλλαγές της εργασίας ταξινόμησης, όπως για παράδειγμα, όταν η συνάρτηση f βγάζει ως έξοδο μια κατανομή πιθανότητας της εισόδου πάνω στις κλάσεις.

- **Παλινδρόμηση (Regression):** Σε αυτό το είδος εργασίας, το πρόγραμμα καλείται να προβλέψει μια αριθμητική τιμή δεδομένης κάποιας εισόδου. Για να επιλύσει αυτή την εργασία, ο αλγόριθμος μηχανικής μάθησης καλείται να παράγει μια συνάρτηση $f: R^n \rightarrow R$. Αυτή η εργασία είναι παρόμοια με την ταξινόμηση, εκτός από το ότι η μορφή της εξόδου είναι διαφορετική.
- **Μετάφραση Μηχανής (Machine Translation):** Σε μια εργασία μηχανικής μετάφρασης, η είσοδος αναπαρίσταται ήδη από ακολουθίες συμβόλων σε κάποια γλώσσα και το πρόγραμμα καλείται να τη μετατρέψει σε ακολουθίες συμβόλων σε άλλη γλώσσα. Αυτό εφαρμόζεται συχνά σε φυσικές γλώσσες, όπως για παράδειγμα για τη μετάφραση κειμένου από τα Αγγλικά στα Γαλλικά. Η βαθιά μηχανική μάθηση έχει καταφέρει τα τελευταία χρόνια σημαντικά υψηλές επιδόσεις σε αυτό το είδος εργασιών, οι οποίες μαζί με τους μηχανισμούς τους θα αναλυθούν εκτενώς στο Κεφάλαιο 3.

Το Μέτρο Απόδοσης (Performance Measure), P

Για να αξιολογήσουμε τις αποτελεσματικότητα ενός αλγορίθμου μηχανικής μάθησης σε συγκεκριμένη εργασία που του ανατίθεται, πρέπει να σχεδιάσουμε ένα αριθμητικό μέτρο υπολογισμού της απόδοσής του. Συνήθως αυτό το μέτρο απόδοσης είναι συγκεκριμένο ως προς το έργο T που εκτελείται από το σύστημα.

Για εργασίες όπως ενδεικτικά η ταξινόμηση, που αναφέρθηκε παραπάνω, συχνά υπολογίζεται η ακρίβεια του μοντέλου. Η ακρίβεια (accuracy) είναι το ποσοστό των παραδειγμάτων εισόδου για τα οποία το μοντέλο παράγει τη σωστή έξοδο. Μπορούμε επίσης να λάβουμε ισοδύναμες πληροφορίες με τη μέτρηση του ρυθμού λάθους (error rate), το ποσοστό των παραδειγμάτων για τα οποία το μοντέλο παράγει εσφαλμένη έξοδο. Αναφέρουμε συχνά το ποσοστό σφάλματος ως την αναμενόμενη απώλεια μεταξύ 0-1. Αυτή, ενδεικτικά, σε ένα συγκεκριμένο παράδειγμα είναι 0 εάν είναι σωστά ταξινομημένο και 1 αν δεν είναι. Ωστόσο, η επιλογή της κατάλληλης μετρικής για κάθε είδος εργασίας δεν είναι πάντα τόσο προφανής, και χρειάζεται διεξοδική μελέτη για τον καθορισμό της, ώστε να αξιολογεί σωστά την απόδοση του μοντέλου. Στο Κεφάλαιο 4 παρουσιάζεται η μετρική απόδοσης που χρησιμοποιείται στο πρόβλημα Πρόβλεψης της Δευτεροταγούς Δομής Πρωτεϊνών που μελετάται στην εργασία.

Συνήθως μας ενδιαφέρει η επίδοση του αλγορίθμου μηχανικής μάθησης σε δεδομένα που το μοντέλο δεν έχει δει στο στάδιο εκπαίδευσής του. Συνεπώς, αξιολογούμε τα μέτρα απόδοσης των μοντέλων χρησιμοποιώντας ένα σύνολο δεδομένων ελέγχου (test set) διαφορετικό από το σύνολο των δεδομένων εκπαίδευσης (train set) πάνω στα οποία το σύστημα μαθαίνει.

Η Εμπειρία (Experience), E

Οι αλγόριθμοι μηχανικής μάθησης μπορούν να ταξινομηθούν ευρέως ως επιβλεπόμενοι και μη επιβλεπόμενοι, με βάση το είδος της εμπειρίας που τους επιτρέπεται να έχουν κατά τη διάρκεια της διαδικασίας εκμάθησης. Για το αντικείμενο της συγκεκριμένης εργασίας χρησιμοποιούνται αλγόριθμοι επιβλεπόμενης μάθησης, συνεπώς αυτοί θα αναλυθούν πιο διεξοδικά στη συνέχεια.

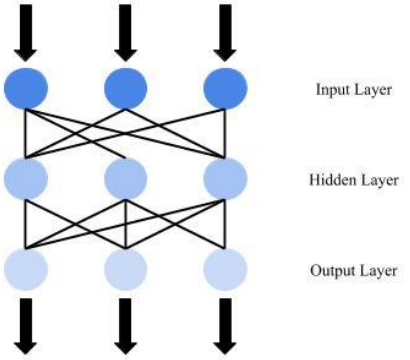
Οι **αλγόριθμοι επιβλεπόμενης μάθησης** μαθαίνουν ένα σύνολο δεδομένων εισόδου (dataset) που περιέχει χαρακτηριστικά, ενώ κάθε παράδειγμα εισόδου σχετίζεται επίσης με μια ετικέτα (label) από το σύνολο των διαθέσιμων ετικετών που ορίζεται από το πρόβλημα. Συνεπώς, στόχος του συστήματος είναι να μάθει μια συνάρτηση η οποία αντιστοιχεί τα δεδομένα εισόδου σε μια έξοδο, χρησιμοποιώντας ζεύγη παραδειγμάτων εισόδου-εξόδου (Stuart J. Russell, 2010). Άρα, στην επιβλεπόμενη μάθηση, κάθε παράδειγμα που μαθαίνει το σύστημα είναι ένα ζεύγος που αποτελείται από ένα αντικείμενο εισόδου (συνήθως ένα διάνυσμα) και μια επιθυμητή τιμή εξόδου (ετικέτα).

2.2 Βασικές Αρχιτεκτονικές Νευρωνικών Δικτύων

Στην ενότητα αυτή θα παρουσιάσουμε βασικές αρχιτεκτονικές νευρωνικών δικτύων, στα οποία θα γίνονται συχνά αναφορές στα επόμενα κεφάλαια της εργασίας, κατά τη μελέτη των διαφορετικών τρόπων προσέγγισης του προβλήματος που μελετάται. Στη συνέχεια, εξηγούνται συνοπτικά τα Δίκτυα Πρόσθιας Τροφοδότησης, τα Συνελικτικά Νευρωνικά Δίκτυα καθώς και τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα.

2.2.1 Δίκτυα Πρόσθιας Τροφοδότησης (Feedforward Neural Networks)

Ένα Νευρωνικό Δίκτυο Πρόσθιας Τροφοδότησης είναι ένα τεχνητό νευρωνικό δίκτυο όπου οι συνδέσεις μεταξύ των κόμβων τους δεν σχηματίζουν έναν κύκλο (Zell, 1994). Το συγκεκριμένο δίκτυο ήταν ο πρώτος και απλούστερος τύπος τεχνητού νευρικού δικτύου που επινοήθηκε (Schmidhuber J. , 2015). Σε αυτό, οι πληροφορίες μετακινούνται μόνο προς μία κατεύθυνση, προς τα εμπρός, από τους κόμβους εισόδου, μέσω των κρυφών κόμβων στους κόμβους εξόδου.



Σχήμα 2.1: Νευρωνικό Δίκτυο Πρόσθιας Τροφοδότησης με ένα κρυφό στρώμα.

Το μοντέλο αυτό μπορεί να παρομοιαστεί με ένα κατευθυνόμενο ακυκλικό γράφημα που περιγράφει τον τρόπο με τον οποίο διαφορετικές συναρτήσεις προσεγγίζονται μαζί. Για παράδειγμα, μπορούμε να έχουμε τρεις συναρτήσεις $f^{(1)}$, $f^{(2)}$, και $f^{(3)}$ συνδεδεμένες σε μια αλυσίδα, για να σχηματίσουμε τη συνάρτηση $f^{(3)}(f^{(2)}(f^{(1)}(\mathrm{x}))$. Σε αυτή την περίπτωση, το $f^{(1)}$ είναι το πρώτο στρώμα του δικτύου, το $f^{(2)}$ το δεύτερο στρώμα και ούτω καθεξής. Το συνολικό μήκος της αλυσίδας δίνει το βάθος του μοντέλου. Επειδή τα δεδομένα εκπαίδευσης του δεν εμφανίζουν την επιθυμητή έξοδο σε κανένα από αυτά τα στρώματα, αποκαλούνται κρυφά στρώματα (hidden layers). Τα κρυφά στρώματα απαιτούν την επιλογή συναρτήσεων ενεργοποίησης (activation functions) που θα χρησιμοποιηθούν για τον υπολογισμό των τιμών τους. Για τον καθορισμό της αρχιτεκτονικής ενός τέτοιου δικτύου, απαιτείται η επιλογή του αριθμού των στρωμάτων που περιέχει το δίκτυο, του τρόπου με τον οποίο συνδέονται αυτά τα επίπεδα μεταξύ τους, καθώς και το πλήθος των κόμβων σε κάθε επίπεδο. (Ian Goodfellow, 2016, pp. 164-165)

Μια κλασική οικογένεια μεθόδων που χρησιμοποιούνται για την αποτελεσματική εκπαίδευση των τεχνητών νευρωνικών δικτύων είναι οι **Αλγόριθμοι Οπισθοδιάδοσης Σφάλματος** (Backpropagation), οι οποίοι ακολουθούν αλγορίθμους που βασίζονται στον υπολογισμό κλίσεων (gradient-based optimization algorithms) με τη χρήση του κανόνα της αλυσίδας. Το κύριο χαρακτηριστικό τους είναι η επαναληπτικότητα, η αναδρομικότητα και η αποδοτικότητα των μεθόδων για τον υπολογισμό των ενημερώσεων των διαφόρων βαρών του δικτύου για τη βελτίωση της απόδοσής του κατά την εκπαίδευσή (Ian Goodfellow, 2016, σ. 196).

2.2.2 Συνελικτικά Νευρωνικά Δίκτυα

Τα Συνελικτικά Νευρωνικά Δίκτυα (ΣΝΔ) είναι ένα είδος Νευρωνικών Δικτύων πρόσθιας τροφοδότησης, που χρησιμοποιούνται εκτεταμένα στο πεδίο της Όρασης Υπολογιστών, αλλά

παράλληλα έχουν πολλές εφαρμογές σε εργασίες επεξεργασίας φυσικής γλώσσας και αναγνώριση μουσικής. Τα ΣΝΔ έχουν πάρει το όνομά τους από τον τύπο των κρυφών επιπέδων από τα οποία αποτελούνται. Τα κρυμμένα στρώματα ενός ΣΝΔ αποτελούνται τυπικά από συνελικτικά στρώματα (convolutional layers), πλήρως συνδεδεμένα στρώματα (fully connected layers), στρώματα συγκέντρωσης (pooling layers) και στρώματα κανονικοποίησης (normalization layers). Μετά την συνέλιξη, ένα μη γραμμικό στρώμα (ή συνάρτηση ενεργοποίησης) εφαρμόζεται στον χάρτη ενεργοποίησης. Ο στόχος αυτού του στρώματος είναι να αφαιρέσει το θόρυβο και να ενισχύσει τα σημαντικά σήματα. Η συνάρτηση ενεργοποίησης μετά από ένα συνελικτικό στρώμα είναι συνήθως η ReLU[1] ή κάποια άλλη μη-γραμμική συνάρτηση (όπως Sigmoid, Tanh). Ένα από τα μεγαλύτερα πλεονεκτήματα που έχει η συνάρτηση ενεργοποίησης ReLU έναντι των υπολοίπων είναι ότι δεν ενεργοποιεί ταυτόχρονα όλους τους νευρώνες. Μετατρέπει όλες τις αρνητικές εισροές στο μηδέν και ο νευρώνας δεν ενεργοποιείται. Αυτό το καθιστά πολύ αποτελεσματικό υπολογιστικά καθώς λίγοι νευρώνες ενεργοποιούνται κάθε φορά.

Τα ΣΝΔ παίρνουν ως είσοδο ένα n-διάστατο διάνυσμα (π.χ. έναν πίνακα) και επιστρέφουν στην έξοδο έναν άλλο m-διάστατο διάνυσμα (π.χ. έναν πίνακα). Η έξοδος των ΣΝΔ διαβιβάζεται γενικά είτε σε άλλο κομμάτι του δικτύου (που περιλαμβάνει ένα άλλο ΣΝΔ) είτε σε ένα πυκνό στρώμα (dense layer), το οποίο μπορεί να τροφοδοτηθεί τελικά σε μια συνάρτηση SoftMax για τον υπολογισμό της κατανομής πιθανοτήτων πάνω στις ετικέτες της προς εκμάθησης εργασίας. Στις εργασίες ταξινόμησης εικόνων, οι είσοδοι αποτελούνται είτε από δισδιάστατα (ασπρόμαυρες εικόνες) είτε τρισδιάστατα (έγχρωμες εικόνες) διανύσματα. Τα δισδιάστατοι διανύσματα μπορούν να οπτικοποιηθούν ως πίνακες, οι αξίες των οποίων περιγράφουν το αν ένα εικονοστοιχείο είναι μαύρο ή όχι. Τα τρισδιάστατα διανύσματα δεν διαφέρουν ουσιαστικά. Περιέχουν μόνο μία επιπλέον διάσταση (δηλαδή βάθος ή κανάλια), των οποίων το μέγεθος είναι γενικά 3 και αναπαριστά τρεις μήτρες που ονομάζονται R, B και G (Κόκκινο, Μπλε και Πράσινο), καθεμιά από τις οποίες περιέχει τιμές που περιγράφουν την αντίστοιχη ένταση χρώματος κάθε εικονοστοιχείου.

Στρώμα Συνέλιξης (Convolution Layer): Το στρώμα συνέλιξης είναι το κεντρικό δομικό στοιχείο του ΣΝΔ, καθώς αναλαμβάνει τον κύριο φόρτο των υπολογισμών του δικτύου. Εφαρμόζοντας την πράξη της συνέλιξης, υπολογίζει το βαθμωτό γινόμενο μεταξύ δύο μητρών, όπου η μία μήτρα είναι το σύνολο παραμέτρων προς μάθηση (learnable parameters) που είναι γνωστές ως πυρήνας (kernel) και η άλλη μήτρα είναι ένα τμήμα του δεκτικού πεδίου (receptive

[1] Η συνάρτηση ενεργοποίησης ReLU (Rectified Linear Units) ορίζεται ως $R(z) = max(0, z)$, δηλαδή $R(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases}$ και παράγωγο $R'(z) = \begin{cases} 1, & z > 0 \\ 0, & z < 0 \end{cases}$.

field). Ο πυρήνας είναι χωρικά μικρότερος από μια εικόνα, αλλά έχει μεγαλύτερο βάθος. Αυτό σημαίνει ότι εάν η εικόνα αποτελείται από τρία κανάλια (π.χ. RGB), το ύψος και το πλάτος του πυρήνα θα είναι μικρότερα σε μέγεθος από την εικόνα, αλλά το βάθος θα εκτείνεται έως και στα τρία κανάλια.

Κατά τη διάρκεια της προς τα εμπρός διέλευσης, ο πυρήνας ολισθαίνει κατά μήκος του ύψους και του πλάτους της εικόνας, παράγοντας μια αναπαράσταση εικόνας για αυτή την περιοχή υποδοχής (receptive region). Αυτό παράγει μια δισδιάστατη αναπαράσταση της εικόνας που είναι γνωστή ως χάρτης ενεργοποίησης (activation or feature map), που δίνει την απόκριση του πυρήνα σε διαφορετικά σημεία της εικόνας. Το μέγεθος κατά το οποίο ο πυρήνας σύρεται κατά το μήκος και το ύψος της εικόνας ονομάζεται βήμα (stride). Το βήμα γενικά ρυθμίζεται με τρόπο που επιτρέπει στο φίλτρο να εξετάσει ολόκληρο το διάνυσμα εισόδου (π.χ. εάν το βήμα είναι πολύ μεγάλο, το φίλτρο μπορεί να μην είναι σε θέση να πλησιάσει στα όρια ή στις γωνίες του διανύσματος εισόδου). Επειδή το φίλτρο δεν μπορεί να βρεθεί εν μέρει έξω από το διάνυσμα εισόδου, το διάνυσμα εισόδου είναι συχνά διευρυμένο, γεμισμένο με τιμές (padding). Το γέμισμα γίνεται με την επέκταση μιας ή περισσότερων διαστάσεων των διανυσμάτων εισόδου με ορισμένες τιμές (συνήθως μηδενικά), έτσι ώστε το φίλτρο να μπορεί να ολισθήσει επίσης κατά μήκος των ορίων και των γωνιών του διανύσματος εισόδου. Ένας τρόπος για να υπολογιστεί αυτό το γέμισμα είναι χρησιμοποιώντας τον τύπο $padding = \frac{K-1}{2}$, όπου K είναι το μέγεθος του φίλτρου. Το μέγεθος της εξόδου ενός ΣΝΔ μπορεί να υπολογιστεί από τη σχέση $Output\_dim = \frac{W-K+2 \cdot P}{S} + 1$, όπου Output_dim: ύψος/μήκος εξόδου, W: ύψος/μήκος εισόδου, P: γέμισμα και S: βήμα.

Στρώμα Συγκέντρωσης (Pooling Layer):

Οι συνελίξεις και οι μη γραμμικές προβολές ακολουθούνται γενικά από ένα στρώμα υποδειγματοληψίας (downsampling) ή συγκέντρωσης (pooling). Το στρώμα συγκέντρωσης αντικαθιστά την έξοδο του δικτύου σε ορισμένες θέσεις, αντλώντας μια στατιστική σύνοψη των κοντινών εξόδων. Αυτό βοηθά στη μείωση του χωρικού μεγέθους της αναπαράστασης, γεγονός που μειώνει την ποσότητα των αναγκαίων υπολογισμών και βαρών. Η λειτουργία της συγκέντρωσης γίνεται μεμονωμένα σε κάθε κομμάτι της αναπαράστασης. Υπάρχουν διάφορες λειτουργίες συγκέντρωσης όπως ο μέσος όρος της ορθογώνιας γειτονιάς, η νόρμα[2] L2 της ορθογώνιας γειτονιάς και ένας σταθμισμένος μέσος όρος με βάση την απόσταση από το κεντρικό εικονοστοιχείο. Ωστόσο, η πιο δημοφιλής διαδικασία είναι η μέγιστη συγκέντρωση (max pooling), η οποία επιστρέφει τη μέγιστη έξοδο από τη γειτονιά. Παρόμοια με τα

[2] Η L2 νόρμα ενός διανύσματος $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ είναι ίση με $||v||_2 = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2}$.

στρώματα συνέλιξης, τα στρώματα συγκέντρωσης έχουν ένα σύνολο υπερπαραμέτρων που θα καθοριστούν, συμπεριλαμβανομένου του μεγέθους του φίλτρου, του μήκους του βήματος. Τα στρώματα συγκέντρωσης πρέπει επίσης να δηλώσουν τη συνάρτηση που πρέπει να εφαρμοστεί, από τις παραπάνω που αναφέρθηκαν. Τα στρώματα συγκέντρωσης έχουν γενικά φίλτρα μεγέθους $2 \times 2 \times Channels$ (κανάλια), τα οποία ολισθαίνουν κατά 2 βήματα. Ο υπολογισμός του μεγέθους της εξόδου είναι ίδιος με αυτόν που χρησιμοποιείται για τα στρώματα συνέλιξης, με την εξαίρεση ότι δεν υπάρχει γέμισμα (padding), δηλαδή $Output\_dim = \frac{W-K}{S} + 1$, όπου Output_dim: ύψος/μήκος εξόδου, W: ύψος/μήκος εισόδου και S: βήμα.

Πλήρως συνδεδεμένο στρώμα (fully connected layer): Οι νευρώνες σε αυτό το στρώμα έχουν πλήρη συνδεσιμότητα με όλους τους νευρώνες στο προηγούμενο και επόμενο. Αυτός είναι ο λόγος για τον οποίο μπορεί να υπολογιστεί ως συνήθως με πολλαπλασιασμό μήτρας που ακολουθείται από μια σταθερά πόλωσης (bias). Το στρώμα αυτό συμβάλλει στη χαρτογράφηση της αναπαράστασης μεταξύ της εισόδου και της εξόδου.

2.2.3 Επαναλαμβανόμενα Νευρωνικά Δίκτυα

Τα επαναλαμβανόμενα νευρωνικά δίκτυα (Recurrent Neural Networks - RNNs) είναι μια κατηγορία τεχνητών νευρωνικών δικτύων όπου οι συνδέσεις μεταξύ των κόμβων σχηματίζουν ένα κατευθυνόμενο γράφημα κατά μήκος μιας χρονικής ακολουθίας. Αυτό τους επιτρέπει να παρουσιάζουν χρονική δυναμική συμπεριφορά. Σε αντίθεση με τα δίκτυα πρόσθιας τροφοδότησης, τα RNNs μπορούν να χρησιμοποιούν την εσωτερική τους κατάσταση (μνήμη) για να επεξεργάζονται ακολουθίες εισόδων. Αυτό τα καθιστά εφαρμόσιμα σε εργασίες όπως αναγνώριση χειρόγραφου (Graves A. L. M., 2009) ή αναγνώριση ομιλίας (Sak Hasim, 2014), (Li Xiangang, 2014).

Ο όρος Επαναλαμβανόμενο Νευρωνικό Δίκτυο χρησιμοποιείται για την αναφορά σε δύο μεγάλες κατηγορίες δικτύων με παρόμοια γενική δομή, όπου το ένα έχει πεπερασμένες καταστάσεις και το άλλο άπειρες. Και οι δύο κατηγορίες δικτύων παρουσιάζουν χρονική δυναμική συμπεριφορά (Miljanovic, 2012). Ένα πεπερασμένο δίκτυο επαναλαμβανόμενων καταστάσεων είναι ένα κατευθυνόμενο ακυκλικό γράφημα που μπορεί “να ξετυλιχθεί” και να αντικατασταθεί από ένα δίκτυο πρόσθιας τροφοδότησης, ενώ ένα άπειρο επαναλαμβανόμενο δίκτυο είναι ένα κατευθυνόμενο κυκλικό γράφημα που δεν μπορεί να ξετυλιχθεί. Τόσο τα πεπερασμένα όσο και τα άπειρα επαναλαμβανόμενα δίκτυα μπορούν να έχουν επιπρόσθετη αποθηκευμένη κατάσταση και η αποθήκευση μπορεί να βρίσκεται υπό άμεσο έλεγχο από το

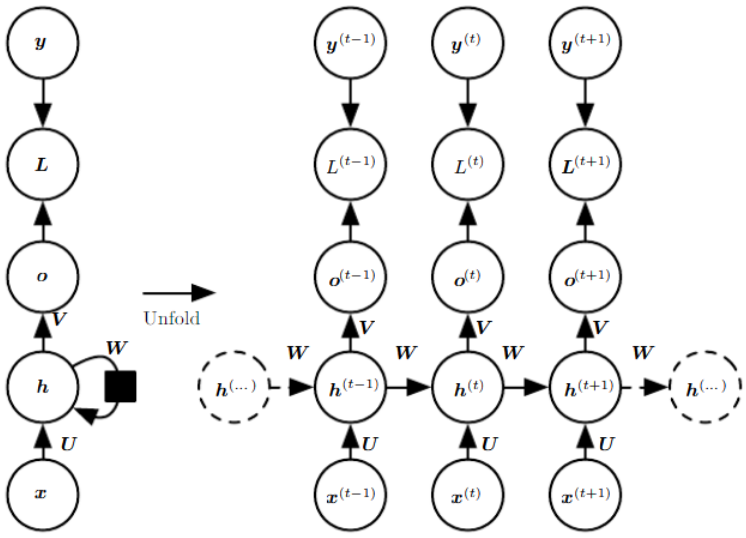
νευρικό δίκτυο. Η αποθήκευση μπορεί επίσης να αντικατασταθεί από άλλο δίκτυο ή γράφημα, εάν αυτό ενσωματώνει χρονικές καθυστερήσεις ή έχει βρόχους ανατροφοδότησης. Αυτές οι ελεγχόμενες καταστάσεις αναφέρονται ως φραγμένες καταστάσεις ή με φραγμένη μνήμη και αποτελούν μέρος των δικτύων Μακράς βραχυχρόνιας μνήμης (Long Short-term Memory – LSTMs) και των Φραγμένων επαναλαμβανόμενων νευρωνικών δικτύων (Gated Recurrent Neural Networks – GRUs).

Στη συνέχεια, θα αναφερθούμε στην αρχιτεκτονική και το θεωρητικό υπόβαθρο βασικών αρχιτεκτονικών επαναλαμβανόμενων νευρωνικών δικτύων, όπως αυτές παρατίθενται στο βιβλίο (Ian Goodfellow, 2016, σσ. 372-376). Ορισμένα παραδείγματα σημαντικών σχεδιαστικών μοτίβων για τα επαναλαμβανόμενα νευρωνικά δίκτυα περιλαμβάνουν τα εξής:

1. Επαναλαμβανόμενα δίκτυα που παράγουν μια έξοδο σε κάθε χρονικό βήμα και έχουν επαναλαμβανόμενες συνδέσεις μεταξύ κρυφών μονάδων, όπως απεικονίζονται στο Σχήμα 2.2.
2. Επαναλαμβανόμενα δίκτυα που παράγουν μια έξοδο σε κάθε χρονικό βήμα και έχουν επαναλαμβανόμενες συνδέσεις μόνο από την έξοδο σε ένα χρονικό βήμα προς τις κρυφές μονάδες στο επόμενο βήμα, που απεικονίζονται στο Σχήμα 2.3.
3. Επαναλαμβανόμενα δίκτυα με επαναλαμβανόμενες συνδέσεις μεταξύ κρυφών μονάδων, που διαβάζουν μια ολόκληρη ακολουθία και στη συνέχεια παράγουν μία μόνο έξοδο, που απεικονίζονται στο Σχήμα 2.4.

Το επαναλαμβανόμενο νευρωνικό δίκτυο του Σχήματος 2.2 και της εξίσωσης 2.1 είναι καθολικό με την έννοια ότι οποιαδήποτε λειτουργία υπολογιζόμενη από μια μηχανή Turing μπορεί να υπολογιστεί από ένα τέτοιο επαναλαμβανόμενο δίκτυο σταθερού μεγέθους. Η έξοδος μπορεί να διαβαστεί από το RNN μετά από έναν αριθμό βημάτων που είναι ασυμπτωτικά γραμμικά με τον αριθμό των χρονικών βημάτων που χρησιμοποιήθηκαν από τη μηχανή Turing και ασυμπτωτικά γραμμικά στο μήκος της εισόδου. Οι λειτουργίες που υπολογίζονται από μια μηχανή Turing είναι διακριτές, επομένως τα αποτελέσματα αυτά αφορούν την ακριβή υλοποίηση της λειτουργίας, όχι τις προσεγγίσεις. Το RNN, όταν χρησιμοποιείται ως μηχανή Turing, παίρνει μια δυαδική ακολουθία ως είσοδο και οι έξοδοί της πρέπει να είναι διακριτοποιημένες ώστε να παρέχουν μια δυαδική έξοδο. Είναι δυνατόν να υπολογίσουμε όλες τις λειτουργίες σε αυτό το δίκτυο χρησιμοποιώντας ένα συγκεκριμένο μέγεθος RNN. Η "είσοδος" της μηχανής Turing είναι μια ειδική περίπτωση της συνάρτησης που πρέπει να υπολογιστεί, επομένως το ίδιο δίκτυο που προσομοιώνει τη μηχανή Turing είναι επαρκές για όλα τα προβλήματα. Τώρα αναπτύσσουμε τις εξισώσεις διάδοσης για το RNN που απεικονίζεται στο Σχήμα 2.2. Η εικόνα δεν καθορίζει την επιλογή της συνάρτησης

ενεργοποίησης για τις μονάδες που βρίσκονται σε λειτουργία. Εδώ υποθέτουμε τη συνάρτηση ενεργοποίησης της υπερβολικής εφαπτομένης. Επίσης, το σχήμα δεν καθορίζει ακριβώς ποια μορφή λαμβάνει η έξοδος και η συνάρτηση απωλειών (loss function). Εδώ υποθέτουμε ότι η έξοδος είναι διακριτή, σαν να χρησιμοποιείται το RNN για την πρόβλεψη λέξεων ή χαρακτήρων.



Σχήμα 2.2: Το υπολογιστικό γράφημα για τον υπολογισμό της απώλειας εκπαίδευσης ενός επαναλαμβανόμενου δικτύου που αντιστοιχεί μια ακολουθία εισόδου των τιμών x σε μια αντίστοιχη ακολουθία τιμών εξόδου o. Μια απώλεια L μετρά πόσο μακριά είναι κάθε ο από τον αντίστοιχο στόχο εκπαίδευσης y. Όταν χρησιμοποιούμε εξόδους softmax, υποθέτουμε ότι ο είναι οι μη κανονικοποιημένες log πιθανότητες. Η απώλεια L υπολογίζει εσωτερικά το $y = softmax(o)$ και το συγκρίνει με το στόχο y. Το RNN έχει τις εισόδους προς τις κρυφές συνδέσεις παραμετροποιημένες από ένα μήτρα βάρους U, τις κρυφές σε κρυφές (hidden-to-hidden) επαναλαμβανόμενες συνδέσεις παραμετροποιημένες από μια μήτρα βάρους W και τις κρυφές συνδέσεις προς την έξοδο παραμετροποιημένες από μια μήτρα βάρους V. (Αριστερά) Το RNN και η απώλειά του σχεδιασμένα με επαναλαμβανόμενες συνδέσεις. (Δεξιά) Το ίδιο ως χρονικά ξεδιπλωμένος υπολογιστικός γράφος, όπου κάθε κόμβος συσχετίζεται τώρα με μία συγκεκριμένη χρονική στιγμή, *(Ian Goodfellow, 2016, σ. 373)*.

Ένας φυσικός τρόπος για να αναπαραστήσουμε τις διακριτές μεταβλητές είναι να θεωρήσουμε ότι η έξοδος δίνει τις μη κανονικοποιημένες λογαριθμικές πιθανότητες κάθε πιθανής τιμής της διακριτής μεταβλητής. Στη συνέχεια, μπορούμε να εφαρμόσουμε τη συνάρτηση softmax ως βήμα μετά την επεξεργασία για να πάρουμε ένα διάνυσμα κανονικοποιημένων πιθανοτήτων της εξόδου. Η διάδοση προς τα εμπρός (forward propagation) ξεκινά με τον προσδιορισμό της αρχικής κατάστασης $h^{(0)}$. Στη συνέχεια, για κάθε βήμα από $t = 1$ έως $t = \tau$, εφαρμόζουμε τις ακόλουθες εξισώσεις ενημέρωσης:

$$\alpha^{(t)} = b + Wh^{(t-1)} + Ux^{(t)}, \quad (2.1)$$

$$h^{(t)} = tanh(\alpha^{(t)}),$$

$$o^{(t)} = c + Vh^{(t)},$$

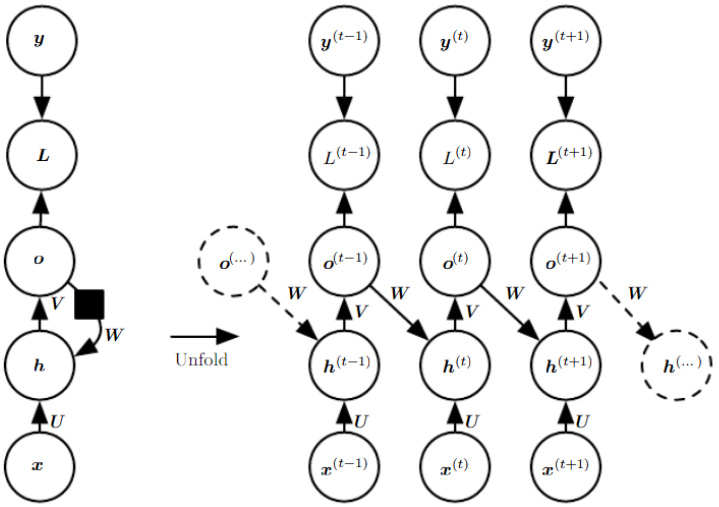
$$\hat{y}^{(t)} = softmax(o^{(t)})$$

όπου οι παράμετροι είναι οι σταθερές b, c μαζί με τους πίνακες βαρών U, V και W, αντίστοιχα, για τις συνδέσεις καταστάσεων εισόδου προς κρυφές καταστάσεις, κρυφών καταστάσεων προς καταστάσεις εξόδου και κρυφών καταστάσεων προς κρυφές καταστάσεις. Αυτό είναι ένα παράδειγμα ενός επαναλαμβανόμενου δικτύου που αντιστοιχεί μια ακολουθία εισόδου σε μια ακολουθία εξόδου του ίδιου μήκους. Η συνολική απώλεια για την ακολουθούμενη ακολουθία των τιμών x σε συνδυασμό με μια ακολουθία τιμών y θα είναι τότε μόνο το άθροισμα των απωλειών (losses) σε όλα τα χρονικά βήματα. Για παράδειγμα, αν $L^{(t)}$ είναι η αρνητική λογαριθμική πιθανότητα (log-likelihood) του $y^{(t)}$ δεδομένων των $x^{(1)}, \dots, x^{(t)}$ τότε:

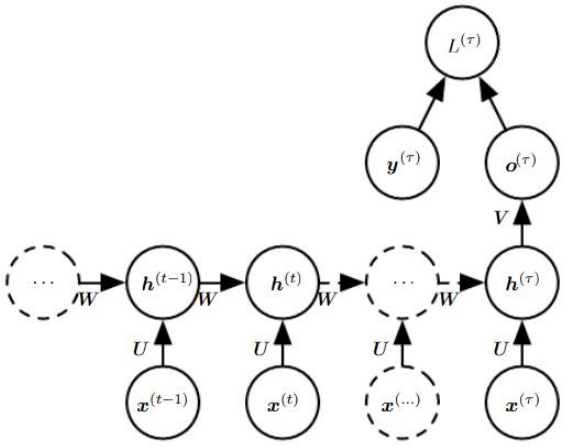
$$L\left(\{x^{(1)}, \dots, x^{(\tau)}\}, \{y^{(1)}, \dots, y^{(\tau)}\}\right) = \sum_t L^{(t)}$$

$$= -\sum_t logp_{model}\left(y^{(t)} | \{x^{(1)}, \dots, x^{(\tau)}\}\right)$$

Όπου το $p_{model}\left(y^{(t)} | \{x^{(1)}, \dots, x^{(\tau)}\}\right)$ δίνεται με την ανάγνωση της εισόδου $y^{(t)}$ από τον εξόδου του μοντέλου $\hat{y}^{(t)}$. Ο υπολογισμός της κλίσης αυτής της συνάρτησης σφάλματος σε σχέση με τις παραμέτρους είναι μια διαδικασία με μεγάλο κόστος. Ο υπολογισμός της κλίσης συνεπάγεται ένα πέρασμα προς τα εμπρός που μετακινείται από αριστερά προς τα δεξιά μέσω της απεικόνισης του ξετυλιγμένου γραφήματος στο Σχήμα 2.2, ακολουθούμενος από μια οπισθοδιάδοση που μετακινείται από δεξιά προς τα αριστερά μέσω του γραφήματος. Ο χρόνος εκτέλεσης είναι $O(\tau)$ και δεν μπορεί να μειωθεί με παραλληλισμό επειδή το γράφημα διάδοσης προς τα εμπρός είναι εγγενώς διαδοχικό, δηλαδή κάθε χρονικό βήμα μπορεί να υπολογιστεί μόνο μετά το προηγούμενο. Οι καταστάσεις που υπολογίζονται στη διάδοση προς τα εμπρός πρέπει να αποθηκευτούν μέχρι να επαναχρησιμοποιηθούν κατά τη διάρκεια της οπισθοδιάδοσης, οπότε το κόστος μνήμης είναι επίσης $O(\tau)$. Ο αλγόριθμος οπισθοδιάδοσης που εφαρμόστηκε στο ξετυλιγμένο γράφημα με το κόστος $O(\tau)$ ονομάζεται οπισθοδιάδοση μέσω χρόνου (Back propagation through time - BPTT).



Σχήμα 2.3: Ένα RNN του οποίου η μόνη επανάληψη είναι η σύνδεση ανατροφοδότησης από την έξοδο στο κρυφό στρώμα. Σε κάθε χρονικό βήμα t, η είσοδος είναι x_t, οι ενεργοποιήσεις των κρυφών στρωμάτων είναι $h^{(t)}$, οι έξοδοι είναι $o^{(t)}$, οι στόχοι είναι $y^{(t)}$, και η απώλεια είναι $L^{(t)}$. (αριστερά) Διάγραμμα κυκλώματος. (δεξιά) Ο ξεδιπλωμένος υπολογιστικός γράφος. Ένας τέτοιο RNN είναι λιγότερο ισχυρό (μπορεί να εκφράσει μικρότερο σύνολο συναρτήσεων) από αυτούς στο σχήμα 2.2. Το RNN στο σχήμα 2.2 μπορεί να επιλέξει να θέσει οποιαδήποτε πληροφορία θέλει για το παρελθόν στην κρυφή του αναπαράσταση h και να μεταδώσει την h στο μέλλον. Το RNN σε αυτό το σχήμα εκπαιδεύεται για να θέσει μια συγκεκριμένη τιμή εξόδου στο o , και το o είναι η μόνη πληροφορία που επιτρέπεται να σταλεί στο μέλλον. Δεν υπάρχουν άμεσες συνδέσεις από το h προς τα εμπρός. Το προηγούμενο h συνδέεται με το παρόν μόνο έμμεσα, μέσω των προβλέψεων για την παραγωγή των οποίων χρησιμοποιήθηκε. Αν δεν είναι πολύ μεγάλης διάστασης και πλούσιο το o, συνήθως δεν θα έχει σημαντικές πληροφορίες από το παρελθόν. Αυτό καθιστά το RNN σε αυτό το σχήμα λιγότερο ισχυρό, αλλά μπορεί να είναι πιο εύκολο να εκπαιδευτεί γιατί κάθε βήμα μπορεί να εκπαιδευτεί μεμονωμένα από τα υπόλοιπα, επιτρέποντας μεγαλύτερο παραλληλισμό κατά τη διάρκεια της εκπαίδευσης *(Ian Goodfellow, 2016, σ. 375)*.



Σχήμα 2.4: Χρονικά ξετυλιγμένο επαναλαμβανόμενο νευρωνικό δίκτυο με μία μόνο έξοδο στο τέλος της ακολουθίας. Ένα τέτοιο δίκτυο μπορεί να χρησιμοποιηθεί για να συνοψίσει μια ακολουθία και να παράγει μια αναπαράσταση σταθερού μεγέθους που χρησιμοποιείται ως είσοδος για περαιτέρω επεξεργασία. Μπορεί να υπάρχει στόχος δεξιά στο τέλος (όπως απεικονίζεται εδώ), ή η κλίση στην έξοδο $o^{(t)}$ μπορεί να ληφθεί με οπισθοδιάδοση από παρακάτω μέρη *(Ian Goodfellow, 2016, σ. 376)*.

2.2.4 Αρχιτεκτονικές Κωδικοποιητή-Αποκωδικοποιητή για την Πρόβλεψη Ακολουθίας από Ακολουθία

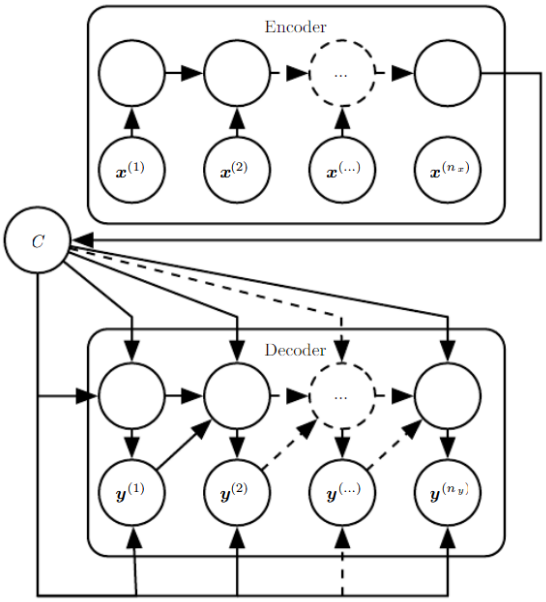
Στη συνέχεια, θα εξηγηθεί πώς ένα RNN μπορεί να εκπαιδευτεί για να απεικονίσει μια ακολουθία εισόδου σε μια ακολουθία εξόδου, η οποία δεν είναι απαραιτήτως του ίδιου μήκους με την είσοδο (Ian Goodfellow, 2016, σσ. 390-408). Αυτή η ιδέα μπορεί να χρησιμοποιηθεί σε πολλές εφαρμογές, όπως η αναγνώριση ομιλίας, η μηχανική μετάφραση και οι ερωτο-απαντήσεις, όπου οι ακολουθίες εισόδου και εξόδου στο σύνολο εκπαίδευσης δεν έχουν γενικά το ίδιο μήκος (αν και τα μήκη τους μπορεί να σχετίζονται). Συχνά καλούμε την είσοδο στο RNN "Το νοηματικό πλαίσιο" (context). Στόχος είναι η παραγωγή μιας απεικόνισης αυτού του πλαισίου, έστω C. Το σύμβολο C είναι ένα διάνυσμα ή μια ακολουθία από διανύσματα που συνοψίζουν την ακολουθία εισόδου των $X = \left(x^{(1)}, \ldots, x^{(n_x)}\right)$. Η πρώτη απλή αρχιτεκτονική RNN για την απεικόνιση ακολουθίας μεταβλητού μήκους σε άλλη ακολουθία μεταβλητού μήκους προτάθηκε πρώτα από τους Cho et al. (2014α) και μετά από τους Sutskever et al. (2014), οι οποίοι ανέπτυξαν ανεξάρτητα αυτή την αρχιτεκτονική και ήταν πρώτοι που απέκτησαν μετάφραση πολύ καλής ακρίβειας χρησιμοποιώντας αυτήν την προσέγγιση. Το προηγούμενο σύστημα βασίζεται σε προτάσεις βαθμολόγησης που δημιουργούνται από ένα σύστημα μηχανικής μετάφρασης (περισσότερα για τη μηχανική μετάφραση εξηγούνται στο Κεφάλαιο 3), ενώ ο τελευταίος χρησιμοποιεί ένα επαναλαμβανόμενο δίκτυο για την παραγωγή των μεταφράσεων. Αυτοί οι συγγραφείς ονόμασαν αυτή την αρχιτεκτονική, που απεικονίζεται στο Σχήμα 2.5, αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή (encoder-decoder architecture) ή αρχιτεκτονική πρόβλεψης ακολουθίας από ακολουθία (sequence-to-sequence architecture).

Η ιδέα αυτού του μοντέλου είναι πολύ απλή:

1. Ένας κωδικοποιητής (encoder) ή αναγνώστης (reader) ή είσοδος (input) του RNN επεξεργάζεται την ακολουθία εισόδου. Ο κωδικοποιητής επιστρέφει το πλαίσιο C, συνήθως ως απλή συνάρτηση της τελικής κρυφής κατάστασής του.
2. Ένας αποκωδικοποιητής (decoder) ή συγγραφέας (writer) ή έξοδος (output) του RNN ρυθμίζεται στο εν λόγω διάνυσμα σταθερού μήκους (ακριβώς όπως στο Σχήμα 2.5) για να παράγει την ακολουθία εξόδου $Y = \left(y^{(1)}, \ldots, y^{(n_y)}\right)$.

Η καινοτομία αυτού του είδους της αρχιτεκτονικής σε σχέση με τα απλά RNN είναι ότι τα μήκη n_x και n_y μπορεί να διαφέρουν μεταξύ τους. Στην αρχιτεκτονική πρόβλεψης ακολουθίας από ακολουθία, τα δύο RNN εκπαιδεύονται από κοινού για να μεγιστοποιήσουν το μέσο όρο του $\log P\left(y^{(1)}, \ldots, y^{(n_y)} \mid x^{(1)}, \ldots, x^{(n_x)}\right)$ πάνω σε όλα τα ζεύγη των x, y στο σύνολο εκπαίδευσης. Η τελευταία κατάσταση h_{n_x} του κωδικοποιητή RNN χρησιμοποιείται συνήθως

ως μια αναπαράσταση της ακολουθίας εισόδου που παρέχεται ως είσοδος στον αποκωδικοποιητή RNN.



Σχήμα 2.5: Παράδειγμα μιας αρχιτεκτονικής RNN κωδικοποιητή – αποκωδικοποιητή ή ακολουθίας σε ακολουθία, για την εκμάθηση της παραγωγής μιας ακολουθίας εξόδου $(y^{(1)},\dots,y^{(n_y)})$ δοθείσας μιας ακολουθίας εισόδου $(x^{(1)},\dots,x^{(n_x)})$. Αποτελείται από έναν κωδικοποιητή RNN που διαβάζει την ακολουθία εισόδου και έναν αποκωδικοποιητή RNN που παράγει την ακολουθία εξόδου (ή υπολογίζει την πιθανότητα μιας δοθείσας ακολουθίας εξόδου). Η τελική κρυφή κατάσταση του κωδικοποιητή RNN χρησιμοποιείται για τον υπολογισμό μιας γενικής μεταβλητής με σταθερό μέγεθος για το νοηματικό πλαίσιο C, που αναπαριστά μια νοηματική περίληψη της ακολουθίας εισόδου και δίνεται ως είσοδος στον αποκωδικοποιητή RNN, *(Ian Goodfellow, 2016, σ. 391)*.

Εάν το πλαίσιο C είναι ένα διάνυσμα, τότε ο αποκωδικοποιητής RNN είναι απλά ένα διάνυσμα στην ακολουθία RNN. Υπάρχουν τουλάχιστον δύο τρόποι ένα διάνυσμα στην ακολουθία RNN να λάβει την είσοδο. Η είσοδος μπορεί να παρέχεται στην αρχική κατάσταση του RNN ή η είσοδος μπορεί να συνδεθεί με τις κρυμμένες μονάδες σε κάθε χρονικό βήμα. Αυτοί οι δύο τρόποι μπορούν επίσης να συνδυαστούν. Δεν υπάρχει κανένας περιορισμός στο ότι ο κωδικοποιητής πρέπει να έχει το ίδιο μέγεθος κρυμμένων στρωμάτων με τον αποκωδικοποιητή. Ένας σαφής περιορισμός αυτής της αρχιτεκτονικής εμφανίζεται όταν η έξοδος του πλαισίου C από τον κωδικοποιητή RNN έχει μια διάσταση που είναι πολύ μικρή για να συνοψίσει σωστά μια μεγάλη ακολουθία. Αυτό το φαινόμενο παρατηρήθηκε από τον Bahdanau et al. (2015) στη μηχανική μετάφραση πλαισίου. Πρότειναν να κάνουν το πλαίσιο C μια ακολουθία μεταβλητού μήκους από ότι ένα διάνυσμα σταθερού μεγέθους. Επιπλέον, εισήγαγαν ένα μηχανισμό προσοχής (attention mechanism) που μαθαίνει να συσχετίζει στοιχεία της ακολουθίας C με τα στοιχεία της ακολουθίας εξόδου. Ο μηχανισμός αυτός αναλύεται εκτενώς στο Κεφάλαιο 3.

2.2.5 Η πρόκληση των Εξαρτήσεων μεταξύ απομακρυσμένων χρονικών βημάτων (Long-term Dependencies) στα Επαναλαμβανόμενα Νευρωνικά Δίκτυα

Υπάρχει μια μαθηματική πρόκληση στην εκμάθηση των εξαρτήσεων μεταξύ των απομακρυσμένων χρονικών βημάτων των ακολουθιών στα επαναλαμβανόμενα δίκτυα. Το βασικό πρόβλημα είναι ότι οι κλίσεις που διαδίδονται σε πολλά στάδια τείνουν είτε να εξαφανιστούν (τις περισσότερες φορές) είτε να αυξηθούν (σπάνια, αλλά δημιουργώντας μεγάλα προβλήματα στη βελτιστοποίηση). Ακόμη και αν υποθέσουμε ότι οι παράμετροι είναι τέτοιες ώστε το επαναλαμβανόμενο δίκτυο να είναι σταθερό (μπορεί να αποθηκεύσει τις μνήμες, με κλίσεις που δεν αυξάνονται), η δυσκολία με τις μακροχρόνιες εξαρτήσεις προκύπτει από τα εκθετικά μικρότερα βάρη που δίδονται στις μακροχρόνιες αλληλεπιδράσεις (που περιλαμβάνουν τον πολλαπλασιασμό πολλών Ιακωβιανών πινάκων (Jacobians)) σε σύγκριση με τις βραχυπρόθεσμες. Τα επαναλαμβανόμενα δίκτυα περιλαμβάνουν τη σύνθεση της ίδιας συνάρτησης πολλές φορές, μία φορά ανά βήμα χρόνου. Αυτές οι συνθέσεις μπορούν να οδηγήσουν σε εξαιρετικά μη γραμμική συμπεριφορά.

Μπορούμε να σκεφτούμε την σχέση επανάληψης:

$$h^{(t)} = W^T h^{(t-1)}$$

ως ένα πολύ απλό επαναλαμβανόμενο νευρωνικό δίκτυο που δεν έχει μια μη γραμμική λειτουργία ενεργοποίησης και δεν έχει υπολειπόμενες εισόδους x. Μπορεί να απλοποιηθεί στη σχέση:

$$h^{(t)} = (W^t)^T h^{(0)}$$

και αν αντικαταστήσουμε το W με την ανάλυση σε ιδιοτιμές (eigendecomposition) $W = Q\Lambda Q^T$, με χρήση του ορθογώνιου πίνακα Q, η επαναληπτικότητα απλοποιείται στο εξής $h^{(t)} = Q^T \Lambda^t Q h^{(0)}$. Οι ιδιοτιμές υψώνονται στη δύναμη t, με αποτέλεσμα οι ιδιοτιμές με τιμές μικρότερες από ένα να τείνουν στο μηδέν και ιδιοτιμές με μέγεθος μεγαλύτερο από ένα να αυξάνονται υπερβολικά. Κάθε στοιχείο του $h^{(0)}$ που δεν είναι ευθυγραμμισμένο με το μεγαλύτερο ιδιοδιάνυσμα θα απορριφθεί τελικά. Αυτό το πρόβλημα είναι σύνηθες στα επαναλαμβανόμενα δίκτυα.

Το πρόβλημα εξαφάνισης και υπερβολική αύξησης που περιγράφηκε στα RNNs αποκαλύφθηκε από διάφορους ερευνητές (Hochreiter, 1991), (Bengio Y. F. P., 1993), (Bengio Y. S. P., 1994). Κάποιος ενδεχομένως να υποθέσει ότι το πρόβλημα μπορεί να αποφευχθεί απλώς παραμένοντας σε μια περιοχή του χώρου παραμέτρων όπου οι κλίσεις δεν τείνουν να εξαφανιστούν ή να αυξηθούν υπερβολικά. Δυστυχώς, ένα RNN πρέπει να εισέλθει σε μια περιοχή του χώρου των παραμέτρων όπου οι κλίσεις εξαφανίζονται (Bengio Y. F. P., 1993), (Bengio Y. S. P., 1994), ώστε να αποθηκεύσει τις μνήμες με τρόπο εύρωστο σε μικρές

διαταραχές. Ειδικότερα, κάθε φορά που το μοντέλο πρέπει να αναπαραστήσει μακροπρόθεσμες εξαρτήσεις, η κλίση μιας μακροπρόθεσμης αλληλεπίδρασης έχει εκθετικά μικρότερο μέγεθος από την κλίση μιας βραχυπρόθεσμης αλληλεπίδρασης. Αυτό σημαίνει όχι ότι είναι αδύνατο, αλλά ότι μπορεί να χρειαστεί πολύς χρόνος για την εκμάθηση μακροπρόθεσμων εξαρτήσεων, επειδή το σήμα σχετικά με αυτές τις εξαρτήσεις θα τείνει να κρύβεται από τις μικρότερες διακυμάνσεις που προκύπτουν από βραχυπρόθεσμες εξαρτήσεις. Στην πράξη, τα πειράματα των (Bengio Y. S. P., 1994) δείχνουν ότι καθώς αυξάνουμε το εύρος των εξαρτήσεων που πρέπει να αναλυθούν, η βελτιστοποίηση που βασίζεται στην κλίση γίνεται ολοένα και πιο δύσκολη, με την πιθανότητα επιτυχούς εκπαίδευσης ενός παραδοσιακού RNN μέσω SGD optimizer να προσεγγίζει γρήγορα το μηδέν για ακολουθίες μήκους μόνο 10 ή 20. Αν και έχουν προταθεί διάφορες προσεγγίσεις για να μειώσουν της δυσκολία της εκμάθησης των μακροχρόνιων εξαρτήσεων, το πρόβλημα αυτό παραμένει μια από τις κύριες προκλήσεις στη βαθιά μηχανική μάθηση.

2.2.6 Τα Δίκτυα Μακράς Βραχυχρόνιας μνήμης (Long Short-term Memory – LSTMs) και τα Φραγμένα Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Gated Recurrent Neural Networks – GRUs)

Τα πιο αποτελεσματικά επαναλαμβανόμενα δίκτυα που χρησιμοποιούνται σε πρακτικές εφαρμογές είναι τα ονομαζόμενα φραγμένα (gated) RNN. Αυτά περιλαμβάνουν μακρά βραχυχρόνια μνήμη (Long Short-term Memory) και δίκτυα που βασίζονται σε φραγμένη επαναλαμβανόμενη μονάδα (gated recurrent unit). Τα φραγμένα RNNs βασίζονται στην ιδέα της δημιουργίας διαδρομών μέσω του χρόνου, που έχουν παραγώγους που ούτε εξαφανίζονται ούτε αυξάνονται υπερβολικά. Γενικεύουν την ιδέα των συνδέσεων βαρών, σε βάρη σύνδεσης που μπορεί να αλλάζουν σε κάθε βήμα. Οι μονάδες διαρροής (Leaky Units) επιτρέπουν στο δίκτυο να συγκεντρώνει πληροφορίες (όπως στοιχεία για ένα συγκεκριμένο χαρακτηριστικό ή κατηγορία) για μεγάλη διάρκεια χρόνου. Μόλις αυτές οι πληροφορίες χρησιμοποιηθούν, ωστόσο, ίσως είναι χρήσιμο για το νευρικό δίκτυο να ξεχάσει μια παλιά κατάσταση. Για παράδειγμα, αν μια ακολουθία αποτελείται από υπακολουθίες και επιθυμούμε μια μονάδα διαρροής να συσσωρεύσει στοιχεία μέσα σε κάθε υπακολουθία, χρειαζόμαστε έναν μηχανισμό για να εμποδίσουμε την παλιά κατάσταση θέτοντας την στο μηδέν. Αντί να αποφασίσουμε χειροκίνητα για το πότε θα αποκλείσουμε αυτή την κατάσταση, θέλουμε το νευρικό δίκτυο να μάθει να αποφασίζει πότε πρέπει να το κάνει, πράγμα που κάνουν τα φραγμένα RNNs.

Δίκτυα Μακράς Βραχυχρόνιας μνήμης (LSTMs)

Τα επαναλαμβανόμενα δίκτυα LSTM έχουν LSTM κελιά που διαθέτουν εσωτερική επανάληψη (self-loop), εκτός από την εξωτερική επανάληψη του Επαναλαμβανόμενου Νευρωνικού Δικτύου. Κάθε κελί έχει τις ίδιες εισόδους και εξόδους σαν ένα συνηθισμένο επαναλαμβανόμενο δίκτυο, αλλά έχει και περισσότερες παραμέτρους και ένα σύστημα φραγμένων μονάδων (gating units) που ελέγχει την ροή πληροφοριών. Το πιο σημαντικό σημείο τους είναι οι μονάδες κατάστασης $s_i^{(t)}$, οι οποίες έχουν μια γραμμική εσωτερική επανάληψη (linear self-loop). Το βάρος της εσωτερικής επανάληψης (ή της συσχετιζόμενης χρονικής σταθεράς) ελέγχεται από μια μονάδα που ονομάζεται forget state unit $f_i^{(t)}$ (για το χρονικό βήμα t και τη μονάδα i), που θέτει αυτό το βάρος σε τιμή μεταξύ 0 και 1 μέσω μιας σιγμοειδούς μονάδας:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right)$$

Όπου $x^{(t)}$ είναι το τρέχον διάνυσμα εισόδου και το $h^{(t)}$ είναι το τρέχον διάνυσμα κρυφού στρώματος που περιέχει τις εξόδους όλων των κελιών LSTM και b^f, U^f, W^f είναι τα βάρη των σταθερών πόλωσης (biases), τα βάρη των εισόδων και τα επαναλαμβανόμενα βάρη για τα forget states. Επομένως, η εσωτερική κατάσταση του κελιού LSTM ενημερώνεται ως εξής, με ένα βάρος $f_i^{(t)}$ εσωτερικής επανάληψης:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right)$$

Όπου b, U, W ορίζουν τα βάρη των σταθερών πόλωσης (biases), τα βάρη των εισόδων και τα επαναλαμβανόμενα βάρη αντίστοιχα στο κελί LSTM. Η μονάδα εξωτερικής εισόδου external input gate $g_i^{(t)}$ υπολογίζεται με παρόμοιο τρόπο με τη μονάδα forget state (με μια μονάδα σιγμοειδούς για να επιτευχθεί τιμή μεταξύ 0 και 1), αλλά με τις δικές της παραμέτρους:

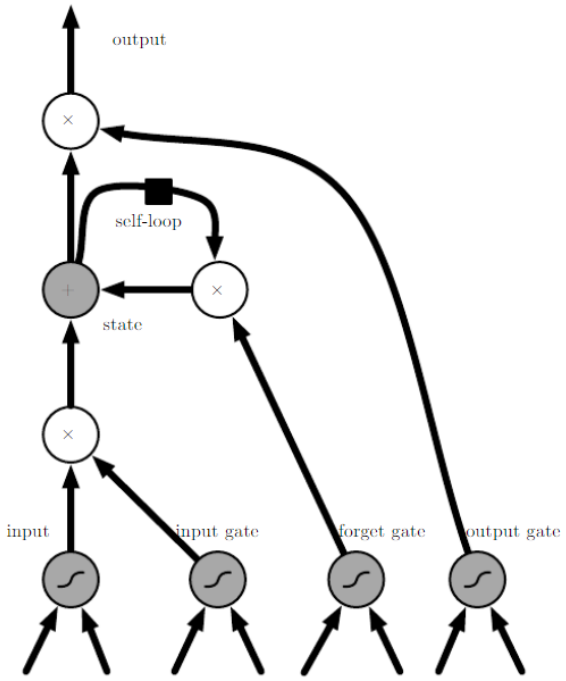
$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right)$$

Η έξοδος $f_i^{(t)}$του LSTM κελιού μπορεί να αποκλειστεί με μια μονάδα εξόδου output gate $g_i^{(t)}$, που επίσης χρησιμοποιεί μια σιγμοειδή μονάδα:

$$h_i^{(t)} = tanh\left(s_i^{(t)}\right) q_i^{(t)}$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right)$$

Που έχει παραμέτρους b^o, U^o, W^o για τα βάρη των σταθερών πόλωσης (biases), τα βάρη των εισόδων και τα επαναλαμβανόμενα βάρη αντίστοιχα. Τα δίκτυα LSTM έχουν αποδειχθεί ότι μαθαίνουν μακροπρόθεσμα εξαρτήσεις πιο εύκολα από τις απλές επαναλαμβανόμενες αρχιτεκτονικές, πρώτον σε τεχνητά σύνολα δεδομένων που έχουν σχεδιαστεί για να δοκιμάσουν την ικανότητα μάθησης μακροπρόθεσμων εξαρτήσεων (Bengio Y. S. P., 1994), (Hochreiter S. S. J., 1997), (Hochreiter S. B. Y., 2001) και στη συνέχεια, σε δύσκολες εργασίες επεξεργασίας ακολουθίας, όπου αποκτήθηκαν υπερσύγχρονες επιδόσεις (Graves, 2012), (Graves A. M. A., 2013), (Ilya Sutskever, 2014).



Σχήμα 2.6: Το διάγραμμα ενός LSTM επαναλαμβανόμενου δικτύου. Τα κελιά συνδέονται επαναληπτικά μεταξύ τους αντικαθιστώντας τις συνήθεις κρυφές μονάδες των επαναλαμβανόμενων δικτύων. Ένα χαρακτηριστικό της εισόδου υπολογίζεται με μια κανονική τεχνητή μονάδα νευρώνα. Η τιμή του μπορεί να συσσωρευτεί μέσα στην κατάσταση αν η σιγμοειδής πύλη εισόδου (gate) το επιτρέπει. Η μονάδα κατάστασης έχει μια γραμμική εσωτερική επανάληψη της οποίας τα βάρη ελέγχονται από τη forget gate. Η έξοδος του κελιού μπορεί να απενεργοποιηθεί από την πύλη εξόδου. Το μαύρο τετράγωνο δείχνει μια καθυστέρηση ενός απλού χρονικού βήματος *(Ian Goodfellow, 2016, σ. 405)*.

Φραγμένα Επαναλαμβανόμενα Νευρωνικά Δίκτυα (GRUs)

Σχετικά πρόσφατες δουλειές έχουν γίνει σε άλλες αρχιτεκτονικές Επαναλαμβανόμενων Νευρωνικών Δικτύων, όπως οι Φραγμένες Επαναλαμβανόμενες μονάδες (GRUs) (Cho K., 2014), (Junyoung Chung, 2014), (Chung J., 2015), (Jozefowicz R., 2015), (Chrupala G., 2015). Η βασική διαφορά με το LSTM είναι ότι μια μοναδική μονάδα πύλης (gate) ελέγχει ταυτόχρονα τον παράγοντα forget και την απόφαση για την ενημέρωση της μονάδας κατάστασης (state unit). Οι εξισώσεις ενημέρωσης των βαρών είναι οι εξής:

$$h_i^{(t)} = u_i^{(t-1)} h_i^{(t-1)} + (1 - u_i^{(t-1)}) \sigma \left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} r_j^{(t-1)} h_j^{(t-1)} \right)$$

Όπου η u είναι η πύλη ανανέωσης (update gate) και r η πύλη επαναφοράς (reset gate). Η τιμή τους ορίζεται ως:

$$u_i^{(t)} = \sigma \left(b_i^u + \sum_j U_{i,j}^u x_j^{(t)} + \sum_j W_{i,j}^u h_j^{(t)} \right)$$

$$r_i^{(t)} = \sigma \left(b_i^r + \sum_j U_{i,j}^r x_j^{(t)} + \sum_j W_{i,j}^r h_j^{(t)} \right)$$

Οι πύλες (gates) επαναφοράς και ενημέρωσης μπορούν ξεχωριστά να "αγνοήσουν" τμήματα του διανύσματος κατάστασης. Οι πύλες ενημέρωσης μπορούν να περιορίσουν γραμμικά οποιαδήποτε διάσταση, επιλέγοντας έτσι να την αντιγράψουν (σε ένα άκρο της σιγμοειδούς) ή να την αγνοήσουν τελείως (στο άλλο άκρο) αντικαθιστώντας τη με τη νέα κατάσταση "στόχου". Οι πύλες επαναφοράς που ελέγχουν τα τμήματα της κατάστασης συνηθίζουν να υπολογίζουν την επόμενη κατάσταση στόχου, εισάγοντας μια πρόσθετη μη γραμμική επιρροή στην σχέση μεταξύ προηγούμενης κατάστασης και μελλοντικής κατάστασης.

Κεφάλαιο 3

Η εφαρμογή των Νευρωνικών Δικτύων στην Μετάφραση Ακολουθιών Κειμένου

Σε αυτό το κεφάλαιο παρουσιάζεται το πεδίο έρευνας της Μηχανικής Μετάφρασης (machine translation) ακολουθιών κειμένου με τη χρήση Τεχνητών Νευρωνικών Δικτύων. Αναφέρονται συνοπτικά οι σημαντικότερες αρχιτεκτονικές δικτύων που έχουν εφαρμοστεί για την επίτευξη της συγκεκριμένης εργασίας, ενώ γίνεται μια εκτενής περιγραφή του Μοντέλου του Μεταφραστή (Transformer), ο οποίος και αποτελεί το δίκτυο με την μεγαλύτερη επιτυχία στην περιοχή. Η ανάλυση της αρχιτεκτονικής και των διαφόρων τεχνικών λεπτομερειών αυτού του δικτύου είναι απαραίτητη, καθώς η εφαρμογή του μοντέλου αυτού στο πρόβλημα της Πρόβλεψης της Δευτεροταγούς Δομής των Πρωτεϊνών είναι το κύριο αντικείμενο μελέτης της συγκεκριμένης εργασίας (η προσαρμογή και τα αποτελέσματα του μοντέλου στο πρόβλημα αυτό παρατίθενται σε επόμενα κεφάλαια).

3.1 Εισαγωγή στη Μηχανική Μετάφραση

Η Μηχανική μετάφραση είναι το πεδίο της υπολογιστικής γλωσσολογίας που μελετά την χρήση υπολογιστικών εργαλείων ώστε να μεταφράσει αυτόματα κείμενα από μία φυσική γλώσσα σε μία άλλη. Στο βασικό της επίπεδο, η μηχανική μετάφραση πραγματοποιεί απλές αντικαταστάσεις λέξεων από μία φυσική γλώσσα σε μία άλλη, παρόλα αυτά, αυτή η τεχνική από μόνη της αδυνατεί να παράγει ποιοτική μετάφραση ενός κειμένου, καθώς συνήθως χρειάζεται η αναγνώριση ολόκληρων φράσεων και των συμφραζόμενων λέξεων για μια πιο πιστή απόδοση της μετάφρασης. Η αντιμετώπιση αυτού του ζητήματος με σώματα κειμένων και στατιστικές τεχνικές αποτελεί ένα ραγδαία αναπτυσσόμενο πεδίο, το οποίο οδηγεί σε καλύτερες μεταφράσεις. Οι τεχνικές βαθιάς μηχανικής μάθησης αποτελούν ένα εργαλείο ιδιαίτερα χρήσιμο για την αντιμετώπιση της συγκεκριμένης εργασίας, μιας και η αρχιτεκτονική πολλών νευρωνικών δικτύων και συνδυασμών αυτών μπορεί να προσεγγίσει αποτελεσματικά το ζήτημα της πρόβλεψης μεγάλων ακολουθιών εξόδου από ακολουθίες εισόδου, όπως αποδεικνύεται με την ανάλυση που ακολουθεί.

Τα Επαναλαμβανόμενα νευρωνικά δίκτυα (Recurrent Neural Networks – RNNs), τα δίκτυα Μακράς βραχυχρόνιας μνήμης (Long Short-term Memory – LSTMs) (Schmidhuber S. H., 1997) και τα Φραγμένα επαναλαμβανόμενα νευρωνικά δίκτυα (Gated Recurrent Neural Networks – GRUs) (Junyoung Chung, 2014) είχαν εδραιωθεί μέχρι το 2017 (που εισηγήθηκε

το μοντέλο του Μεταφραστή – Transformer) ως οι πιο επιτυχείς προσεγγίσεις σε προβλήματα μοντελοποίησης ακολουθιών κειμένου και μηχανικής μετάφρασης (Ilya Sutskever, 2014) (Dzmitry Bahdanau, 2014) (Kyunghyun Cho, 2014). Πολυάριθμες προσπάθειες συνεχίζουν να εξελίσσουν επαναλαμβανόμενα μοντέλα για την επεξεργασία γλώσσας και αρχιτεκτονικές κωδικοποιητή-αποκωδικοποιητή (Encoder - Decoder architectures) (Yonghui Wu, 2016) (Minh-Thang Luong, 2015) (Rafal Jozefowicz, 2016).

Τα Επαναλαμβανόμενα νευρωνικά δίκτυα τυπικά επεκτείνουν τους υπολογισμούς κατά μήκος όλων των θέσεων των συμβόλων των ακολουθιών εισόδου και εξόδου. Αντιστοιχίζοντας τις θέσεις με τα βήματα που γίνονται κατά τον χρόνο υπολογισμού, παράγουν μια ακολουθία από κρυφές καταστάσεις h_t, ως συνάρτηση της προηγούμενης κρυφής κατάστασης h_{t-1} και της εισόδου για τη θέση t. Αυτή η διαδοχική επεξεργασία των θέσεων της εισόδου αποκλείει τον παραλληλισμό μέσα στα παραδείγματα εκπαίδευσης, γεγονός που σε μεγαλύτερο μήκος αλληλουχίας, δημιουργεί περιορισμούς μνήμης. Πρόσφατες εργασίες έχουν επιτύχει βελτιώσεις στην υπολογιστική αποτελεσματικότητα μέσω τεχνικών παραγοντοποίησης (factorization tricks) (Ginsburg, 2017) και μέσω υπολογισμού υπό όρους (Noam Shazeer, 2017). Ωστόσο, ο περιορισμός των διαδοχικών υπολογισμών παραμένει ένα σημαντικό πρόβλημα.

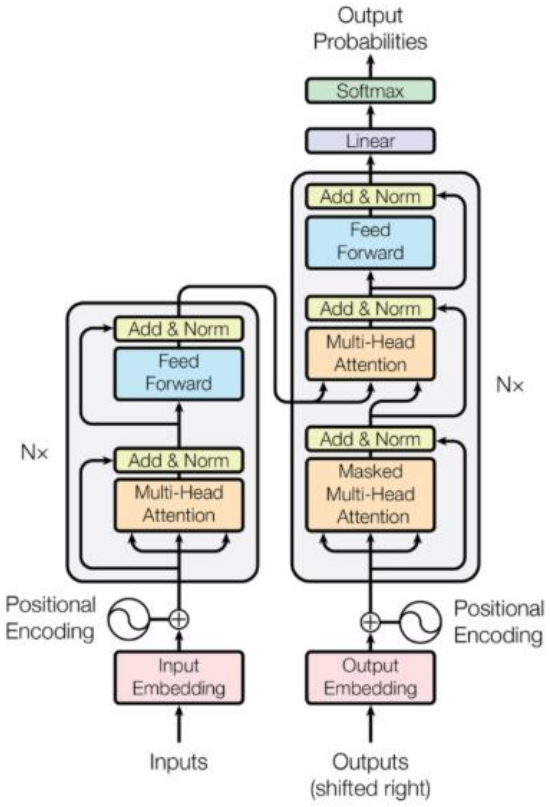
Οι μηχανισμοί προσοχής (attention mechanisms, η λειτουργία τους αναλύεται στη συνέχεια) έχουν γίνει αναπόσπαστο κομμάτι για τη μοντελοποίηση ακολουθιών και για μοντέλα μετάφρασης, επιτρέποντας τη μοντελοποίηση των εξαρτήσεων ανεξάρτητα από την απόσταση τους στις ακολουθίες εισόδου ή εξόδου (Dzmitry Bahdanau, 2014) (Yoon Kim, 2017). Σε όλες τις περιπτώσεις, εκτός από μερικές (Ankur Parikh, 2016), αυτοί οι μηχανισμοί προσοχής χρησιμοποιούνται σε συνδυασμό με ένα επαναλαμβανόμενο νευρωνικό δίκτυο.

3.2 Το Μοντέλο του Μεταφραστή (Transformer)

Το Δεκέμβρη του 2017, κυκλοφόρησε το μοντέλο του Μεταφραστή (Transformer) στο άρθρο “Attention is all you need” (Ashish Vaswani, 2017) το οποίο πέτυχε σημαντική βελτίωση στην ποιότητα μετάφρασης, παρέχοντας ταυτόχρονα μια νέα αρχιτεκτονική για πολλές άλλες εργασίες επεξεργασίας φυσικής γλώσσας (Natural Language Processing – NLP). Στη συνέχεια, γίνεται μια αναλυτική περιγραφή των κύριων συνιστωσών του μοντέλου και της εκπαίδευσής του στις επιμέρους εργασίες που έχει εφαρμοστεί.

Η Αρχιτεκτονική του Μοντέλου

Ο Μεταφραστής (Transformer) αποτελείται από μια συνιστώσα κωδικοποίησης (Encoder stacks), μια συνιστώσα αποκωδικοποίησης (Decoder stacks) και συνδέσεις μεταξύ αυτών. Η συνιστώσα κωδικοποίησης αποτελείται από μια στοίβα από κωδικοποιητές (encoders) και η συνιστώσα αποκωδικοποίησης από το αντίστοιχο πλήθος αποκωδικοποιητών (decoders). Οι κωδικοποιητές είναι όλοι ταυτόσημοι στη δομή τους (παρότι δεν μοιράζονται τα βάρη τους) και καθένας αποτελείται από δύο υποστρώματα: ένα Δίκτυο Αυτό-προσοχής (self-attention network) και ένα Δίκτυο Πρόσθιας Τροφοδότησης (feed-forward neural network). Οι είσοδοι κάθε κωδικοποιητή πρώτα περνούν από ένα στρώμα αυτό-προσοχής (self-attention layer), το οποίο βοηθά τον κωδικοποιητή να κοιτάξει σε άλλες λέξεις στην ακολουθία εισόδου, ενώ κωδικοποιεί μια συγκεκριμένη λέξη. Η έξοδος του στρώματος αυτο-προσοχής περνά μέσα από ένα Δίκτυο Πρόσθιας Τροφοδότησης, το οποίο και ανεξάρτητα εφαρμόζεται σε κάθε θέση. Ο κωδικοποιητής έχει και τα δύο αυτά υποστρώματα, αλλά μεταξύ τους υπάρχει ένα στρώμα προσοχής (attention) το οποίο του επιτρέπει να επικεντρώνεται σε σχετικά μέρη της ακολουθίας εισόδου.



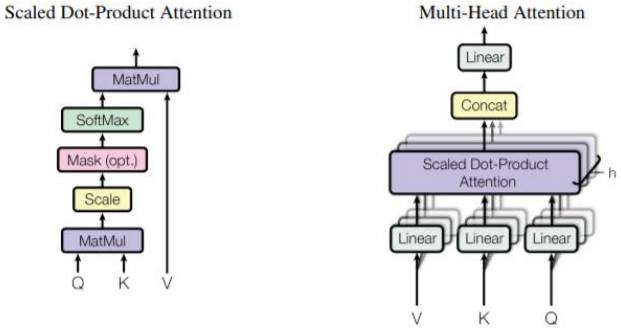
Σχήμα 3.1: Η αρχιτεκτονική του μοντέλου του Transformer**,** ***(Ashish Vaswani, 2017)*****.**

Οι Συνιστώσες του Κωδικοποιητή (Encoder) και του Αποκωδικοποιητή (Decoder)

Κωδικοποιητής: Ο κωδικοποιητής αποτελείται από μια στοίβα $N = 6$ πανομοιότυπων στρωμάτων. Κάθε στρώμα έχει δύο υποστρώματα. Το πρώτο είναι ένας μηχανισμός πολλαπλών κεφαλών (multi-head) στρωμάτων αυτο-προσοχής (self-attention layers), και το δεύτερο είναι ένα απλό, ως προς τη θέση (position-wise) πλήρως συνδεδεμένο δίκτυο πρόσθιας τροφοδότησης (fully-connected feed-forward neural network). Χρησιμοποιείται μια υπολειπόμενη σύνδεση (residual connection) (Kaiming He, 2016) γύρω από κάθε μια από αυτές τα δύο υποστρώματα, ακολουθούμενη από κανονικοποίηση στρώματος (layer normalization) (Jimmy Lei Ba, 2016). Δηλαδή, η έξοδος κάθε υποστρώματος είναι $LayerNorm(x + Sublayer(x))$, όπου το $Sublayer(x)$ είναι η λειτουργία που υλοποιείται από το υπόστρωμα το ίδιο. Για να διευκολυνθούν αυτές οι υπολειμματικές συνδέσεις, όλες τα υποστρώματα του μοντέλου, καθώς και τα στρώματα ενσωμάτωσης (embedding layers), παράγουν εξόδους με διάσταση d_{model}= 512 (Ashish Vaswani, 2017).

Αποκωδικοποιητής: Ο αποκωδικοποιητής αποτελείται επίσης από μια στοίβα N = 6 ίδιων στρωμάτων. Εκτός από τα δύο υποστρώματα σε κάθε στρώμα κωδικοποιητή, ο αποκωδικοποιητής εισάγει ένα τρίτο υπόστρωμα, το οποίο εφαρμόζει προσοχή πολλαπλών κεφαλών στην έξοδο της στοίβας του κωδικοποιητή. Παρόμοια με τον κωδικοποιητή, χρησιμοποιούνται υπολειμματικές συνδέσεις (residual connections) γύρω από κάθε υπόστρωμα, ακολουθούμενες από κανονικοποίηση στρώματος (layer normalization). Επίσης, τροποποιείται η αυτο-προσοχή στη στοίβα αποκωδικοποιητή για να εμποδίσει τις θέσεις να παρακολουθήσουν τις επόμενες θέσεις. Με την κάλυψη (masking) των επόμενων θέσεων, και με το γεγονός ότι οι ενσωματώσεις εξόδου μετατοπίζονται κατά μία θέση, εξασφαλίζει ότι το οι προβλέψεις για τη θέση i μπορούν να εξαρτώνται μόνο από τις γνωστές εξόδους σε θέσεις μικρότερες από i.

Η συνάρτηση Προσοχής (Attention)



Σχήμα 3.2: (αριστερά) Προσοχή Σταθμισμένου Βαθμωτού Γινομένου (Scaled Dot-Product Attention). (δεξιά) Η Προσοχή πολλαπλών κεφαλών (Multi-Head Attention) αποτελείται από διάφορα στρώματα προσοχής που εκτελούνται παράλληλα**,** ***(Ashish Vaswani, 2017)*****.**

Μια συνάρτηση προσοχής μπορεί να περιγραφεί ως μια αντιστοίχιση ενός ερωτήματος (query) και ενός συνόλου από ζεύγη κλειδιού-τιμής (key-value) σε μια έξοδο, όπου το ερώτημα, τα κλειδιά, οι τιμές και η έξοδος είναι όλα τα διανύσματα. Η έξοδος υπολογίζεται ως σταθμισμένο άθροισμα των τιμών, όπου το βάρος που αντιστοιχείται σε κάθε τιμή υπολογίζεται από μια συνάρτηση συμβατότητας του ερωτήματος με το αντίστοιχο κλειδί.

Η Προσοχή Σταθμισμένου Βαθμωτού Γινομένου (Scaled Dot-Product Attention)

Ο μηχανισμός προσοχής των (Ashish Vaswani, 2017) ονομάζεται "Προσοχή Σταθμισμένου Βαθμωτού Γινομένου" (παρουσιάζεται στο Σχήμα 3.2 αριστερά). Η είσοδος αποτελείται από τα ερωτήματα (queries) και τα κλειδιά (keys) της διάστασης d_k και τις τιμές (values) της διάστασης d_v. Υπολογίζονται τα βαθμωτά γινόμενα του ερωτήματος με όλα τα κλειδιά (keys), και κάθε αποτέλεσμα διαιρείται με την ποσότητα $\sqrt{d_k}$, και εφαρμόζεται η συνάρτηση softmax για να ληφθούν τα βάρη στις τιμές (values).

Στην πράξη, υπολογίζεται η συνάρτηση προσοχής σε ένα σύνολο ερωτημάτων ταυτόχρονα, τοποθετημένα μαζί σε μια μήτρα Q. Τα κλειδιά και οι τιμές είναι επίσης τοποθετημένα όλα μαζί σε πίνακες K και V. Υπολογίζεται η μήτρα των αποτελεσμάτων ως:

$$Attention(Q, V, K) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Οι δύο συνηθέστερες συναρτήσεις προσοχής που χρησιμοποιούνται είναι η προσθετική προσοχή (additive attention) (Dzmitry Bahdanau, 2014), και η προσοχή βαθμωτού γινομένου (πολλαπλασιαστική, dot-product). Η προσοχή βαθμωτού γινομένου είναι ίδια με αυτή που χρησιμοποιείται στο εν λόγω μοντέλο, εκτός από τον συντελεστή $\sqrt{d_k}$. Η προσθετική προσοχή υπολογίζει τη συνάρτηση συμβατότητας, που αναφέρθηκε παραπάνω, χρησιμοποιώντας ένα δίκτυο πρόσθιας τροφοδότησης με ένα μόνο κρυφό στρώμα. Ενώ και οι δύο τεχνικές είναι παρόμοιες σε θεωρητική πολυπλοκότητα, η προσοχή βαθμωτού γινομένου είναι πολύ ταχύτερη και πιο αποδοτική στην πράξη, δεδομένου ότι μπορεί να εφαρμοστεί με τη χρήση βελτιστοποιημένου κώδικα για τον πολλαπλασιασμό μητρών (Ashish Vaswani, 2017).

Ενώ για τις μικρές τιμές του $\sqrt{d_k}$ οι δύο μηχανισμοί έχουν παρόμοια απόδοση, η προσθετική προσοχή υπερέχει της προσοχής βαθμωτού γινομένου χωρίς την κλιμάκωση με το συντελεστή $\sqrt{d_k}$ για μεγαλύτερες τιμές d_k (Denny Britz, 2017). Οι συγγραφείς αναφέρουν ότι πιστεύουν ότι για μεγάλες τιμές του d_k, τα βαθμωτά γινόμενα μεγαλώνουν σε μέγεθος, ωθώντας τη συνάρτηση softmax σε περιοχές όπου έχει εξαιρετικά μικρές κλίσεις (gradients). Για να αντισταθμιστεί αυτό το αποτέλεσμα, πραγματοποιούν κλιμάκωση των γινομένων με το συντελεστή $\sqrt{d_k}$ (Ashish Vaswani, 2017).

Προσοχή Πολλαπλών Κεφαλών (Multi-head Attention)

Αντί να εκτελείται μία μόνο λειτουργία προσοχής με κλειδιά, τιμές και ερωτήματα διάστασης d_{model}, οι δημιουργοί του Μεταφραστή (Ashish Vaswani, 2017) επέλεξαν να προβάλλουν γραμμικά τα ερωτήματα, τα κλειδιά και τις τιμές h φορές με διαφορετικές, προς εκμάθηση γραμμικές προβολές στις διαστάσεις d_k, d_k, d_v αντίστοιχα. Σε καθεμία από αυτές τις προβολές των ερωτημάτων, των κλειδιών και των τιμών εφαρμόζεται στη συνέχεια η συνάρτηση προσοχής παράλληλα, παράγοντας τιμές εξόδου d_v-διάστασης. Αυτές συνενώνονται και προβάλλονται για άλλη μια φορά, με αποτέλεσμα τις τελικές τιμές, όπως που απεικονίζεται στο Σχήμα 3.2.

Η προσοχή πολλών κεφαλών επιτρέπει στο μοντέλο να παρακολουθεί από κοινού τις πληροφορίες από διαφορετικές αναπαραστάσεις υποχώρων σε διαφορετικές θέσεις, γεγονός που δεν είναι εφικτό με μία κεφαλή προσοχής.

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O$$

$$\text{όπου } head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

$$\text{και } W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}, W^O \in \mathbb{R}^{hd_v \times d_{model}}$$

(MultiHead: προσοχή πολλαπλών κεφαλών, Concat: συνένωση, Attention: προσοχή)

Στο άρθρο του Μεταφραστή, χρησιμοποιούνται $h = 8$ παράλληλα στρώματα προσοχής ή κεφαλές. Για καθένα από αυτά χρησιμοποιείται $d_k = d_v = \frac{d_{model}}{h} = 64$. Λόγω της μειωμένης διάστασης κάθε κεφαλής, το συνολικό υπολογιστικό κόστος είναι παρόμοιο με εκείνο της προσοχής μιας μόνο κεφαλής με πλήρη διάσταση (Ashish Vaswani, 2017).

Η προσοχή πολλών κεφαλών βελτιώνει την απόδοση του στρώματος προσοχής με τους εξής τρόπους:

- Διευρύνει την ικανότητα του μοντέλου να εστιάζει σε διαφορετικές θέσεις.
- Δίνει στο στρώμα προσοχής πολλαπλούς "υποχώρους αναπαράστασης". Χρησιμοποιώντας προσοχή πολλαπλών κεφαλών δεν δημιουργείται μόνο ένα, αλλά πολλαπλά σύνολα ερωτημάτων, κλειδιών, τιμών πινάκων βάρους (για οκτώ κεφαλές προσοχής, υπάρχουν οκτώ σύνολα για κάθε κωδικοποιητή, αποκωδικοποιητή) . Κάθε ένα από αυτά τα σύνολα αρχικοποιείται τυχαία. Στη συνέχεια, μετά την εκπαίδευση, κάθε σύνολο χρησιμοποιείται για να προβάλλει τις ενσωματώσεις (embeddings) των εισόδων (ή τα διανύσματα επόμενων επιπέδων των κωδικοποιητών, αποκωδικοποιητών) σε έναν διαφορετικό υπόχωρο αναπαράστασης.

Οι εφαρμογές του Μηχανισμού Προσοχής στο Μοντέλο του Μεταφραστή

Ο Μεταφραστής χρησιμοποιεί προσοχή πολλαπλών κεφαλών με τρεις διαφορετικούς τρόπους:

- Στα στρώματα "προσοχής κωδικοποιητή-αποκωδικοποιητή", τα ερωτήματα προέρχονται από το προηγούμενο στρώμα αποκωδικοποιητή, και τα κλειδιά και οι τιμές μνήμης προέρχονται από την έξοδο του κωδικοποιητή. Αυτό επιτρέπει σε κάθε θέση στον αποκωδικοποιητή να παρακολουθεί όλες τις θέσεις στην ακολουθία εισόδου. Επίσης, μιμείται τους τυπικούς μηχανισμούς προσοχής κωδικοποιητή-αποκωδικοποιητή σε μοντέλα μετατροπής ακολουθίας σε ακολουθία όπως τα (Yonghui Wu, 2016), (Dzmitry Bahdanau, 2014), (Jonas Gehring, 2017).
- Ο κωδικοποιητής περιέχει στρώματα αυτο-προσοχής. Σε ένα στρώμα αυτο-προσοχής όλα τα κλειδιά, οι τιμές και τα ερωτήματα προέρχονται από τον ίδιο μέρος, στην περίπτωση αυτή την έξοδο του προηγούμενου στρώματος στον κωδικοποιητή. Κάθε θέση στον κωδικοποιητή μπορεί να παρακολουθεί όλες τις θέσεις στο προηγούμενο στρώμα του κωδικοποιητή.
- Ομοίως, τα στρώματα αυτο-προσοχής στον αποκωδικοποιητή επιτρέπουν σε κάθε θέση στον αποκωδικοποιητή να παρακολουθεί όλες τις θέσεις στον αποκωδικοποιητή μέχρι και τη θέση αυτή. Πρέπει να αποφευχθεί η ροή πληροφοριών προς τα αριστερά στον αποκωδικοποιητή για να διατηρηθεί η auto-regression ιδιότητα. Αυτό εφαρμόζεται στο εσωτερικό της προσοχής σταθμισμένου βαθμωτού γινομένου (scaled dot-product) με την κάλυψη (ρύθμιση σε $-\infty$) όλων των τιμών στην είσοδο της softmax που αντιστοιχούν σε παράνομες συνδέσεις (Βλ. Σχήμα 3.2).

Δίκτυο Πρόσθιας Τροφοδότησης ως προς τη θέση (position-wise Feed-Forward Neural Network)

Εκτός από τα υποστρώματα προσοχής, κάθε στρώμα στον κωδικοποιητή και στον αποκωδικοποιητή περιλαμβάνει ένα πλήρες συνδεδεμένο δίκτυο πρόσθιας τροφοδότησης, το οποίο εφαρμόζεται σε κάθε θέση χωριστά και πανομοιότυπα. Αυτό αποτελείται από δύο γραμμικούς μετασχηματισμούς με συνάρτηση ενεργοποίηση ReLU μεταξύ τους.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

Ενώ οι γραμμικοί μετασχηματισμοί είναι οι ίδιοι στις διαφορετικές θέσεις, χρησιμοποιούν διαφορετικές παραμέτρους από στρώμα σε στρώμα. Ένας άλλος τρόπος για να το περιγράψουμε είναι δύο συνελίξεις με μέγεθος πυρήνα ίσο με 1. Η διαστατικότητα της εισόδου και της εξόδου είναι $d_{model} = 512$, και το εσωτερικό στρώμα έχει διαστάσεις $d_{ff} = 2048$.

Ενσωματώσεις (Embeddings) και Softmax

Παρόμοια με άλλα μοντέλα μετάφρασης ακολουθιών, χρησιμοποιούνται ενσωματώσεις για να μετατρέψουν τα σημεία εισόδου (input tokens) και εξόδου (output tokens) σε διανύσματα διαστάσεων d_{model}. Χρησιμοποιείται επίσης ο συνηθισμένος γραμμικός μετασχηματισμός μάθησης (learned linear transformation) και η συνάρτηση softmax για τη μετατροπή της εξόδου του αποκωδικοποιητή στις προβλεπόμενες πιθανότητες επόμενου σημείου (next token). Στον Μεταφραστή, μοιράζεται ο ίδιος πίνακας βάρους μεταξύ των δύο στρωμάτων ενσωμάτωσης και του pre-softmax γραμμικού μετασχηματισμού, όπως στο (Wolf, 2016). Στα στρώματα ενσωμάτωσης, αυτά τα βάρη πολλαπλασιάζονται με $\sqrt{d_{model}}$.

Κωδικοποίηση Θέσης (Positional Encoding)

Δεδομένου ότι το μοντέλο του Μεταφραστή δεν περιλαμβάνει κάποια αναδρομή (recurrence) ή συνέλιξη (convolution), προκειμένου το μοντέλο να χρησιμοποιήσει τη σειρά της ακολουθίας, πρέπει να εισαχθούν κάποιες πληροφορίες σχετικά με τη σχετική ή απόλυτη θέση του των σημείων ή λέξεων (tokens) στην ακολουθία (Ashish Vaswani, 2017). Για το σκοπό αυτό, προστίθενται "κωδικοποιήσεις θέσης" στις ενσωματώσεις εισόδου στην αρχή των στοιβών του κωδικοποιητή και του αποκωδικοποιητή. Οι κωδικοποιήσεις θέσης έχουν την ίδια διάσταση d_{model} με τις ενσωματώσεις, έτσι ώστε να μπορούν να αθροιστούν με αυτές. Υπάρχουν πολλές επιλογές κωδικοποίησης θέσης, προς μάθηση (learned) και σταθερές (fixed) (Jonas Gehring, 2017). Στο άρθρο που εξετάζεται χρησιμοποιούνται οι συναρτήσεις ημιτόνου και συνημιτόνου με διαφορετικές συχνότητες:

$$PE_{(pos,2i)} = sin(pos/1000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = cos(pos/1000^{2i/d_{model}})$$

όπου pos είναι η θέση και i η διάσταση, PE: positional encoding (κωδικοποίηση θέσης).

Δηλαδή, κάθε διάσταση της κωδικοποίησης θέσης αντιστοιχεί σε ένα ημιτονοειδές. Τα μήκη κύματος σχηματίζουν μια γεωμετρική εξέλιξη από 2π έως $1000 \cdot 2\pi$. Επιλέγεται αυτή η συνάρτηση με την υπόθεση ότι θα επιτρέψει στο μοντέλο να μάθει εύκολα να παρακολουθεί σχετικές θέσεις, αφού για κάθε σταθερή μετατόπιση k, το PE_{pos+k} μπορεί να αναπαρασταθεί ως γραμμική συνάρτηση του PE_{pos} (Ashish Vaswani, 2017).

Επίσης, έγιναν πειράματα με τη χρήση ενσωματώσεων θέσεων προς μάθηση (learned positional embeddings) (Jonas Gehring, 2017), και διαπιστώθηκε ότι οι δύο οι εκδόσεις παρήγαγαν σχεδόν ταυτόσημα αποτελέσματα. Η ημιτονοειδής έκδοση επιλέχθηκε επειδή μπορεί να επιτρέψει στο μοντέλο να επεκτείνεται σε μήκη ακολουθίας μεγαλύτερα από αυτά που συναντήθηκαν κατά τη διάρκεια της εκπαίδευσης (Ashish Vaswani, 2017).

Εκπαίδευση του Μοντέλου του Μεταφραστή για μετάφραση κειμένου

Εκπαίδευση των Δεδομένων και χωρισμός σε υποσύνολα εκπαίδευσης (batching)

Το Μοντέλο του Μεταφραστή εκπαιδεύτηκε στο τυπικό σύνολο δεδομένων από τα Αγγλικά στα Γερμανικά, WMT 2014 English-German dataset που αποτελείται από περίπου 4,5 εκατομμύρια ζεύγη προτάσεων. Οι προτάσεις κωδικοποιήθηκαν χρησιμοποιώντας την κωδικοποίηση ζεύγους byte (byte-pair encoding) (Denny Britz, 2017), η οποία έχει κοινό λεξιλόγιο πηγής-στόχου με περίπου 37000 σημεία (tokens). Για μετάφραση από Αγγλικά στα Γαλλικά, χρησιμοποιήθηκε το σημαντικά μεγαλύτερο WMT 2014 English-French σύνολο δεδομένων αποτελούμενα από προτάσεις 36M και τα σημεία μοιράστηκαν σε λεξιλόγιο 32000 λέξεων (Yonghui Wu, 2016). Τα ζεύγη προτάσεων μπήκαν σε υποσύνολα εκπαίδευσης (batches) με βάση το κατά προσέγγιση μήκος ακολουθίας. Κάθε υποσύνολο εκπαίδευσης περιείχε ένα σύνολο ζευγών προτάσεων που περιείχαν περίπου 25000 σημεία πηγής και 25000 σημεία στόχου.

Αλγόριθμος Βελτιστοποίησης (χρήση Optimizer)

Για τις παραπάνω εργασίες χρησιμοποιήθηκε για βελτιστοποίηση ο Adam Optimizer (Diederik Kingma, 2015) με $\beta_1 = 0.9$, $\beta_2 = 0.98$ και $e = 10^{-9}$. Ο ρυθμός εκμάθησης[3] (lrate) μεταβλήθηκε κατά τη διάρκεια της εκπαίδευσης, σύμφωνα με τον τύπο:

$$lrate = d_{model}^{-0.5} \cdot min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

Αυτό αντιστοιχεί στην αύξηση του ρυθμού εκμάθησης γραμμικά για τα πρώτα βήματα εκπαίδευσης που ορίζονται από τη μεταβλητή $warmup\_steps$, και στη συνέχεια στη μείωση αναλογικά προς την αντίστροφη τετραγωνική ρίζα του αριθμού βήματος, που ορίζεται από τη μεταβλητή $step\_num$. Χρησιμοποιήθηκε $warmup\_steps = 4000$.

Κανονικοποίηση (regularization)

Χρησιμοποιούνται τρεις τύποι κανονικοποίησης κατά τη διάρκεια της εκπαίδευσης στο μοντέλο του Μεταφραστή.

[3] Ο ρυθμός εκμάθησης (learning rate) είναι μια υπερπαράμετρος που ελέγχει πόσο αλλάζει το μοντέλο σε απάντηση του εκτιμώμενου σφάλματος κάθε φορά που ενημερώνονται τα βάρη του μοντέλου.

- **Υπολειπόμενος αποκλεισμός (residual dropout):** Εφαρμόζεται αποκλεισμός (dropout) (Nitish Srivastava, 2014) στην έξοδο κάθε υποστρώματος, προτού προστεθεί στο υπόστρωμα εισόδου και κανονικοποιηθεί. Επιπλέον, εφαρμόζεται αποκλεισμός στα αθροίσματα των ενσωματώσεων και των κωδικοποιήσεων θέσης και στις δύο στοίβες του κωδικοποιητή και του αποκωδικοποιητή. Για το βασικό μοντέλο, χρησιμοποιείται ένα ποσοστό της τάξης του $P_{drop} = 0.1$.
- **Εξομάλυνση Ετικετών:** Κατά τη διάρκεια της εκπαίδευσης, πραγματοποιείται εξομάλυνση ετικετών της τιμής $e_{ls} = 0.1$ (Christian Szegedy, 2015). Αυτό πλήττει την πολυπλοκότητα (perplexity), καθώς το μοντέλο μαθαίνει να είναι πιο αβέβαιο, αλλά βελτιώνει την ακρίβεια και την βαθμολογία BLEU (BLEU score).

<u>Διαφοροποίηση της Διαδικασίας Εκπαίδευσης και της Διαδικασίας Μετάφρασης και Ελέγχου μετά την Εκπαίδευση</u>

Έστω ότι στόχος είναι η μετάφραση ενός συνόλου προτάσεων από τα Αγγλικά (source) στα Γερμανικά (target). Η κωδικοποιημένη είσοδος θα είναι μια αγγλική πρόταση και η είσοδος του αποκωδικοποιητή θα είναι μια γερμανική πρόταση. Ωστόσο, η είσοδος του αποκωδικοποιητή θα μετακινηθεί προς τα δεξιά κατά μία θέση. Ένας λόγος είναι ότι το μοντέλο δεν πρέπει να μάθει πώς να αντιγράφει την είσοδο του αποκωδικοποιητή κατά τη διάρκεια της εκπαίδευσης, αλλά πρέπει δεδομένης της ακολουθίας κωδικοποιητή και μιας συγκεκριμένης ακολουθίας αποκωδικοποιητή, την οποία έχει ήδη δει το μοντέλο, να προβλέπει την επόμενη λέξη. Εάν δεν μετακινηθεί η ακολουθία του αποκωδικοποιητή, το μοντέλο μαθαίνει απλά να αντιγράφει την είσοδο του αποκωδικοποιητή, αφού η λέξη / χαρακτήρας στόχος για τη θέση i θα είναι η λέξη / χαρακτήρας i στην είσοδο του αποκωδικοποιητή. Έτσι, με τη μετατόπιση της εισόδου του αποκωδικοποιητή κατά μία θέση, το μοντέλο πρέπει να προβλέψει τη λέξη / χαρακτήρα προορισμού για τη θέση i έχοντας δει μόνο τη λέξη / χαρακτήρες $1, \ldots, i-1$ στην ακολουθία αποκωδικοποιητή. Η πρώτη θέση της εισόδου του αποκωδικοποιητή συμπληρώνεται με ένα σύμβολο έναρξης πρότασης ("start-of-sentence"), επειδή αυτή η θέση θα ήταν διαφορετικά κενή λόγω της μετατόπισης δεξιά. Ομοίως, εισάγεται ένα σύμβολο τέλους πρότασης ("end-of-sentence") στην ακολουθία εισόδου του αποκωδικοποιητή για να επισημανθεί το τέλος αυτής της ακολουθίας και αυτό προστίθεται επίσης στην πρόταση στόχο εξόδου (target). Αυτό είναι χρήσιμο, για την εξαγωγή των αποτελεσμάτων. Εκτός από την μετατόπιση, ο Μεταφραστής εφαρμόζει μια μάσκα στην είσοδο της πρώτης μονάδας προσοχής πολλαπλών κεφαλών για να αποφύγει την εμφάνιση πιθανών μελλοντικών στοιχείων της ακολουθίας. Αυτό είναι απαραίτητο για την αρχιτεκτονική Transformer επειδή δεν πρόκειται για κάποιο RNN όπου η ακολουθία μπορεί να εισαχθεί διαδοχικά. Εδώ, ολόκληρη η πρόταση εισάγεται μαζί και αν δεν υπήρχε μάσκα, η προσοχή πολλαπλών κεφαλών θα θεωρούσε

ολόκληρη την ακολουθία εισόδου αποκωδικοποιητή σε κάθε θέση. Η ακολουθία στόχων που θέλουμε για τον υπολογισμό των απωλειών (loss function) είναι απλώς η είσοδος αποκωδικοποιητή (γερμανική πρόταση) χωρίς να την μετατοπίζουμε και με ένα σύμβολο τέλους ακολουθίας στο τέλος της.

Η διαδικασία της μετάφρασης για την τελική αξιολόγηση των αποτελεσμάτων που προβλέπονται από το μοντέλο, είναι διαφορετική από την εκπαίδευση, καθώς τελικά στόχος είναι να μεταφραστεί μια αγγλική πρόταση χωρίς την αντίστοιχη γερμανική. Το κλειδί σε αυτή τη διαδικασία είναι η επανατροφοδότηση της εξόδου στο μοντέλο για κάθε θέση της ακολουθίας εξόδου μέχρι να συναντήσουμε ένα σύμβολο τέλους της πρότασης.

Κεφάλαιο 4

Το Πρόβλημα πρόβλεψης της Δευτεροταγούς Δομής των Πρωτεϊνών (ΠΔΔΠ)

Στο κεφάλαιο αυτό αναλύεται το Πρόβλημα Πρόβλεψης της Δευτεροταγούς Δομής των Πρωτεϊνών (ΠΔΔΠ, Protein Secondary Structure Prediction – PSSP). Αφού γίνει η κατάλληλη σύνδεση του προβλήματος με τις αναγκαίες πληροφορίες σε βιολογικό επίπεδο, επεξηγούνται οι πιο επιτυχημένες προσεγγίσεις με χρήση Αλγορίθμων Μηχανικής Μάθησης που έχουν υλοποιηθεί.

4.1 Ορισμός του Προβλήματος

Για τον πλήρη ορισμό του προβλήματος που εξετάζεται στην παρούσα εργασία, σε αυτή την ενότητα, γίνεται αρχικά μια εισαγωγή στη θεωρία της τρισδιάστατης δομής των πρωτεϊνών (παράγραφος 4.1.1). Στη συνέχεια, παρουσιάζονται οι δύο μέθοδοι κωδικοποίησης της προς εξέτασης δομής των πρωτεϊνικών ακολουθιών (παράγραφος 4.1.2) και γίνεται μια σύντομη ανασκόπηση του προβλήματος με κωδικοποίηση Q3 (παράγραφος 4.1.3). Επιπλέον, επεξηγούνται τα πλέον διαδεδομένα σύνολα δεδομένων του προβλήματος (παράγραφος 4.1.3) και αναλύεται η μετρική απόδοσης που χρησιμοποιείται για τη μέτρηση της ακρίβειας των παραγόμενων μοντέλων.

4.1.1 Βιολογικό υπόβαθρο

Οι πρωτεΐνες είναι μεγάλα βιομόρια ή μακρομόρια, που αποτελούνται από μία ή περισσότερες μακριές αλυσίδες υπολειμμάτων αμινοξέων. Οι πρωτεΐνες εκτελούν μια τεράστια ποικιλία λειτουργιών εντός των οργανισμών, όπως καταλυτικές μεταβολικές αντιδράσεις, αντιγραφή DNA, αντίδραση σε ερεθίσματα, παροχή δομής σε κύτταρα και οργανισμούς και μεταφορά μορίων από τη μία θέση στην άλλη. Οι πρωτεΐνες διαφέρουν μεταξύ τους πρωτίστως στην ακολουθία των αμινοξέων που περιέχουν, η οποία υπαγορεύεται από την ακολουθία των νουκλεοτιδίων των γονιδίων τους. Αυτή η ακολουθία συνήθως οδηγεί σε αναδίπλωση της πρωτεΐνης σε μια συγκεκριμένη τρισδιάστατη δομή που καθορίζει τη δραστηριότητα της.

Το σχήμα στο οποίο διπλώνεται φυσιολογικά η πρωτεΐνη είναι γνωστό ως η φυσική διαμόρφωσή της (Murray, p. 36). Υπάρχουν τέσσερα διακριτά επίπεδα της δομής μιας πρωτεΐνης: η πρωτοταγής, η δευτεροταγής, η τριτοταγής και η τεταρτοταγής δομή με βάση το

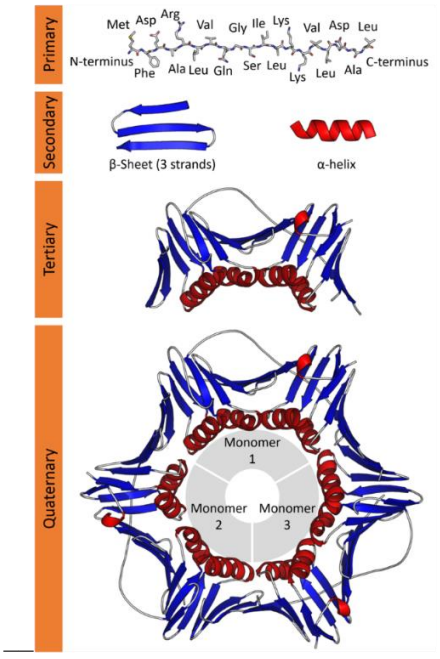
βαθμό της πολυπλοκότητας της πολυπεπτιδικής αλυσίδας. Οι χημικοί δεσμοί μεταξύ τμημάτων της πολυπεπτιδικής αλυσίδας παρέχουν το σχήμα της πρωτεΐνης και συμβάλλουν στη συγκράτησή του.

Ενδεικτικά, η Πρωτοταγής Δομή των Πρωτεϊνών περιγράφει τη μοναδική σειρά στην οποία τα αμινοξέα συνδέονται για να σχηματίσουν μια πρωτεΐνη (Anfinsen CB, 1961). Οι πρωτεΐνες κατασκευάζονται από ένα σύνολο 20 αμινοξέων.

Τα αμινοξέα έχουν τις ακόλουθες δομικές ιδιότητες:

1. Ένα άτομο άνθρακα (alpha carbon) που ενώνεται με τις τέσσερις ομάδες (που αναφέρονται στα 2-5)
2. Ένα άτομο υδρογόνου (H)
3. Μια ομάδα καρβοξυλίου (-COOH)
4. Μια αμινομάδα (-NH2)
5. Μια μεταβλητή ομάδα ή ομάδα R

Η Δευτεροταγής Δομή των Πρωτεϊνών, την οποία και θα μελετήσουμε στην συγκεκριμένη εργασία, αναφέρεται σε τοπικές αναδιπλωμένες δομές που σχηματίζονται μέσα σε ένα πολυπεπτίδιο λόγω αλληλεπιδράσεων μεταξύ των ατόμων του κύριου κορμού (δηλαδή το μέρος της πολυπεπτιδικής αλυσίδας εκτός από τις ομάδες R). Οι πιο συνηθισμένοι τύποι δευτεροταγών δομών είναι η α έλικα (alpha helix) και το β φύλλο (beta pleated sheet). Και οι δύο δομές συγκρατούνται σε σχήμα με δεσμούς υδρογόνου, οι οποίοι σχηματίζονται μεταξύ του καρβονυλικού οξυγόνου ενός αμινοξέος και του αμινικού υδρογόνου του άλλου.



Σχήμα 4.1: Σύνοψη της δομής των πρωτεϊνών (πρωτοταγής, δευτεροταγής, τριτοταγής, τεταρτοταγής). Πηγή Protein Data Bank (PDB).

4.1.2 Οι καταστάσεις κωδικοποίησης στην ΠΔΔΠ (προβλήματα Q3 & Q8)

Ανάμεσα στη μεγάλη ποικιλία πρωτεϊνικών δομών, βρίσκεται ένα σχετικά μικρό σύνολο επαναλαμβανόμενων μοτίβων από γωνίες στρέψης και δεσμούς υδρογόνου που επιτρέπουν στην πρωτεΐνη να διατηρεί τόσο τοπικούς δεσμούς (δηλαδή μεταξύ κοντινών θέσεων στην αλυσίδα) καθώς και πιο απομακρυσμένους (μεταξύ μακρυνότερων θέσεων της αλυσίδας) δεσμούς (Iddo Drori, 2018). Αυτά τα μοτίβα, τα οποία είναι γνωστά ως χαρακτηριστικά της δευτεροταγούς δομής, υποδηλώνουν την ταξινόμηση των υπολειμμάτων πρωτεΐνης σε σχετικά μικρό αριθμό δομικών κλάσεων, που είναι γνωστές ως η δευτεροταγής δομή.

Οι δευτεροταγείς δομές πρωτεϊνών χαρακτηρίζονται παραδοσιακά από 3 γενικές καταστάσεις (Q3 πρόβλημα): έλικα (helix, H), κλώνος (strand, E) και πηνίο (coil, C). Από αυτές τις γενικές τρεις καταστάσεις, το λεξικό της δευτεροταγούς δομής (DSSP) πρότεινε έναν πιο λεπτομερή χαρακτηρισμό των δευτεροταγών δομών, επεκτείνοντας τις τρεις καταστάσεις σε οκτώ καταστάσεις (Q8 πρόβλημα) (Wolfgang Kabsch, 1983): 310 έλικα (310 helix, code G), α-έλικα (α-helix, code H), π-έλικα (π-helix, code E), β-θέση (β-stand, code E), γέφυρα (bridge, code B), στροφή (turn, code T), κάμψη (bend, code S) και άλλα (C). Με την κωδικοποίηση αυτή από τα μέσα της δεκαετίας του '80 η πρόβλεψη της δευτεροταγούς δομής έχει αναδειχθεί σε ένα πολύ ενδιαφέρον αντικείμενο μελέτης του πεδίου αυτού.

4.1.3 Το πρόβλημα Q3 – Προσεγγίσεις και Ποσοστά Επιτυχίας

Η πρόβλεψη του προβλήματος Q3 (δηλαδή των τριών καταστάσεων από τις πρωτεϊνικές αλληλουχίες) έχει διερευνηθεί εντατικά για δεκαετίες χρησιμοποιώντας πολλές μεθόδους μηχανικής μάθησης, μεταξύ των οποίων μελετήθηκαν μοντέλα γράφων πιθανοτήτων (propability graph models) (Schmidler SC, 2000) (Chu W, 2004), διανυσματικοί φορείς υποστήριξης (support vector machines) (Hua S, 2001) (Guo J, 2004), κρυφά μαρκοβιανά μοντέλα (hidden markov models) (Asai K, 1993) (Aydin Z, 2006), τεχνητά νευρωνικά δίκτυα (artificial neural networks - ANNs) (Qian N, 1988) (DT, 1999) (Buchan DW, 2013) (Faraggi E, 2012) και επαναλαμβανόμενο νευρωνικό δίκτυο διπλής κατεύθυνσης (bidirectional recurrent neural network - BRNN) (Baldi P, 1999) (Chen J, 2007) (Mirabello C, 2013) (Torrisi M, 2018).

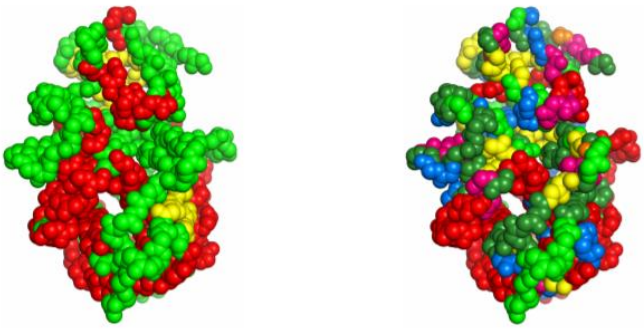
Στις αρχές της δεκαετίας του 90 οι Rost και Sander (Sander, Prediction of protein secondary structure at better than 70% accuracy, 1993) (Sander, Combining evolutionary information and neural networks to predict protein secondary structure, 1994), επαύξησαν τις ακολουθίες πρωτεϊνών βάζοντας μέσα τα προφίλ (profiles), χαρακτηριστικά τα οποία προκύπτουν από πολλαπλή ευθυγράμμιση (alignment) των αλληλουχιών ομόλογων πρωτεϊνών. Εισήγαγαν

επίσης πολυεπίπεδα νευρωνικά δίκτυα, με τα οποία, η πρόβλεψη είχε επιτυχία με ακρίβεια πάνω από 70% για την κωδικοποίηση με χρήση τριών καταστάσεων (Q3), υπερβαίνοντας δραματικά τις προηγούμενες προσεγγίσεις. Η επιτυχία άνοιξε το δρόμο για περαιτέρω μελέτες που παρείχαν πιο περίπλοκες υλοποιήσεις αυτών, αυξάνοντας το ποσοστό επιτυχίας του Q3 έως και 84%. Ωστόσο, καθώς οι επιδόσεις προσέγγιζαν το θεωρητικό όριο (85%-88%) (Yuedong Yang, 2018), το ενδιαφέρον για το πρόβλημα μειώθηκε.

4.1.4 Τα σύνολα δεδομένων Δευτεροταγούς Δομής Πρωτεϊνών

Το καθορισμένο σύνολο δεδομένων PSSP είναι το **CB6133** (Troyanskaya, CB6133 dataset, 2014) για εκπαίδευση (train set) και το **CB513** για δοκιμή (test set). Η δομή αυτών των συνόλων δεδομένων είναι πανομοιότυπη. Το CB6133 περιέχει 5534 αλληλουχίες (μετά την πιο πρόσφατη ανανέωση της βάσης (Iddo Drori, 2018, p. 3)), και για κάθε μία από αυτές, σε κάθε θέση, χρησιμοποιούμε 46 χαρακτηριστικά, ακολουθώντας τα πρότυπα των (Troyanskaya, Deep supervised and convolutional generative stochastic network for protein secondary structure prediction, 2014), (Iddo Drori, 2018, p. 3). Τα πρώτα 22 από αυτά τα χαρακτηριστικά περιέχουν την πληροφορία (κωδικοποιημένη σε μορφή one-hot[4], one-hot encoded) για τα υπολείμματα πρωτεΐνης (residues). Εκτός από τους βασικούς τύπους 20 υπολειμμάτων: A, C, E, D, G, F, I, H, K, M, L, N, Q, P, S, R, T, W, V και Y, (όπως αναφέρθηκαν στην ενότητα 4.1.1) χρησιμοποιείται επιπλέον ο κωδικός X για τα μη τυποποιημένα υπολείμματα και ο κωδικός noSeq. Ένα δεύτερο σύνολο 22 χαρακτηριστικών εμπεριέχεται σε κάθε ακολουθία, τα προφίλ (profiles) που όπως αναφέραμε παραπάνω προκύπτουν από το alignment των ακολουθιών με το εργαλείο PSI-BLAST. Τέλος, δίνονται δύο δυαδικά χαρακτηριστικά που σηματοδοτούν την πρώτη και την τελευταία θέση κάθε ακολουθίας. Σε όλες τις ακολουθίες, μετά το τέλος των χαρακτηριστικών τους, έχουν προστεθεί χαρακτήρες noSeq ώστε να έχουν όλες μήκος 700. Οι κλάσεις των αντίστοιχων ακολουθιών εξόδου που θέλουμε να προβλέψουμε για κάθε δευτεροταγή δομή, περιλαμβάνουν τις οκτώ κατηγορίες που ορίζονται για το Q8: L, B, E, G, I, H, S και T, όπως αναφέραμε στην ενότητα 4.1.2. Ομοίως, την ίδια δομή έχει το σύνολο δεδομένων CB513 (Barton, 1999) που αποτελείται από 513 ακολουθίες και τα ίδια χαρακτηριστικά ανά ακολουθία αξιοποιούνται στην παρούσα εργασία.

[4] Όταν έχουμε μια σειρά από χαρακτηριστικά (έστω για παράδειγμα τα x, y) σε μορφή one-hot encoding και η ακολουθία μας τα περιέχει και τα δύο με τη σειρά xy τότε ακολουθία εισόδου μας κωδικοποιείται σε ένα διάνυσμα της μορφής $\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$, με το 1 να αντιστοιχεί στην ύπαρξη ενός χαρακτήρα στη θέση i της ακολουθίας, και το 0 σε αντίθετη περίπτωση.



Σχήμα 4.2: (αριστερά) Q3, (δεξιά) Q8 δευτεροταγής δομή σφαιρών για την πρωτεΐνη 1AKD στο σύνολο δεδομένων CB513, *(Iddo Drori, 2018)*.

4.1.5 Η Μετρική Απόδοσης του προβλήματος ΠΔΔΠ

Μετά την ολοκλήρωση της διαδικασίας εκπαίδευσης του εκάστοτε μοντέλου που εξετάζεται στο σύνολο δεδομένων CB6133, ακολουθεί η διαδικασία μέτρησης της ακρίβειας της πρόβλεψης στο σύνολο ελέγχου CB513.Η ακρίβεια της πρόβλεψης της δευτεροταγούς δομής με κωδικοποίηση Q8 ορίζεται ανά ακολουθία ως το ποσοστό των υπολειμμάτων για τα οποία οι προβλεπόμενες δευτεροταγείς δομές είναι σωστές (το ίδιο ισχύει και για το πρόβλημα με κωδικοποίηση Q3). Συνεπώς, αν έχουμε, για παράδειγμα, μια ακολουθία για την οποία η Q8 κωδικοποίησή της στα πλαίσια της δευτεροταγούς δομής είναι ένα διάνυσμα έστω x, και το μοντέλο πρόβλεψης βγάζει ως έξοδο μια ακολουθία x' για την ακολουθία εισόδου πρωτεϊνικών υπολειμμάτων που δίνεται, η ακρίβεια της πρόβλεψης είναι το πλήθος των επιτυχημένων προβλέψεων από τη σύγκριση των x, x' θέση προς θέση, προς το μήκος της ακολουθίας. Για το σύνολο όλων των ακολουθιών ελέγχου (test set, CB513) η ακρίβεια υπολογίζεται τελικά ως ο μέσος όρος των επιμέρους ακριβειών που υπολογίζονται ανά προβλεπόμενη ακολουθία.

4.2 Τα πιο επιτυχή μοντέλα της ΠΔΔΠ με κωδικοποίηση Q8

Στην ενότητα αυτή αναλύονται οι αρχιτεκτονικές νευρωνικών δικτύων που έχουν πετύχει την υψηλότερη απόδοση στο πρόβλημα Πρόβλεψης της Δευτεροταγούς Δομής των Πρωτεϊνών με κωδικοποίηση Q8.

4.2.1 Εφαρμογή συνδυασμού (Ensemble) Νευρωνικών Δικτύων

Στο άρθρο “High Quality Protein Q8 Secondary Structure Prediction by Diverse Neural Network Architectures” που δημοσιεύτηκε το 2018 (Iddo Drori, 2018), αντιμετωπίστηκε το

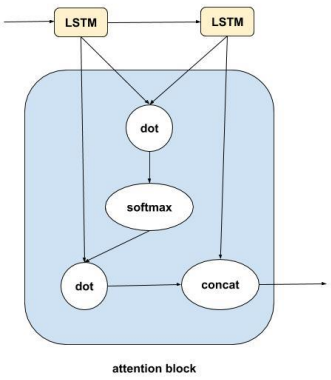
πρόβλημα της Πρόβλεψης της Δευτεροταγούς Δομής Πρωτεϊνών με κωδικοποίηση Q8 χρησιμοποιώντας έναν συνδυασμό επιτυχημένων αρχιτεκτονικών μοντέλων (χρήση της τεχνικής ensemble). Τα πειράματα εφαρμόστηκαν πάνω στα σύνολα CB6133 και CB513.

Αρχιτεκτονικές Δικτύων

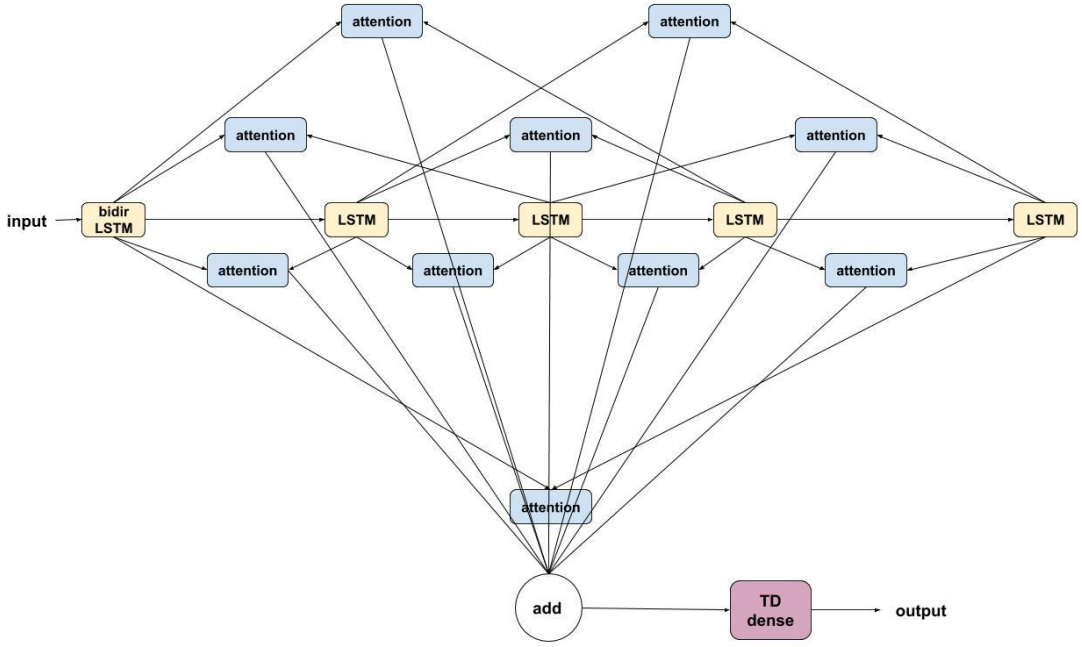
Δημιουργήθηκαν έξι διαφορετικές αρχιτεκτονικές νευρωνικών δικτύων. Ακολουθεί μια λεπτομερής περιγραφή και απεικόνιση της κάθε αρχιτεκτονικής, ακριβώς όπως αυτές παρατίθενται στο σχετικό άρθρο.

Δίκτυο Μακράς βραχυχρόνιας μνήμης (Long Short-term Memory – LSTM) διπλής κατεύθυνσης (Bidirectional) με προσοχή (attention)

Στα σχήματα 4.3, 4.4 που ακολουθούν παρουσιάζεται η αρχιτεκτονική για αυτό το μοντέλο. Η ενσωμάτωση κάθε ακολουθίας αμινοξέων της εισόδου συνενώνεται με τα χαρακτηριστικά των προφίλς (που εξηγήσαμε παραπάνω) και μεταβιβάζεται σε ένα Δίκτυο Μακράς βραχυχρόνιας μνήμης διπλής κατεύθυνσης (Bidirectional LSTM) με 75 μονάδες, ακολουθούμενο από τέσσερα Δίκτυα Μακράς βραχυχρόνιας μνήμης απλής κατεύθυνσης, που το καθένα έχει 150 μονάδες. Η αρχική κατάσταση κάθε Δικτύου Μακράς βραχυχρόνιας μνήμης αρχικοποιείται από την τελευταία κρυφή κατάσταση του προηγούμενου δικτύου (οι οποίες συνενώνονται στην περίπτωση του LSTM διπλής κατεύθυνσης). Για κάθε πιθανό ζεύγος LSTM δικτύων, εφαρμόζεται μηχανισμός προσοχής (Minh-Thang Luong, 2015) χρησιμοποιώντας την έξοδο του τελευταίου LSTM ως ερωτήματα και αποτελέσματα του πρώτου LSTM ως κλειδιά και τιμές. Αυτή η διαδικασία δημιουργεί δέκα εξόδους προσοχής, οι οποίες στη συνέχεια προστίθενται και μεταφέρονται σε δύο πλήρως συνδεδεμένα στρώματα (fully-connected layers). Αυτή είναι η πρώτη φορά που ο μηχανισμός προσοχής (Minh-Thang Luong, 2015), (Ashish Vaswani, 2017) χρησιμοποιείται για το πρόβλημα επιτυγχάνοντας πολύ καλά αποτελέσματα χωρίς τη χρήση συνελίξεων.



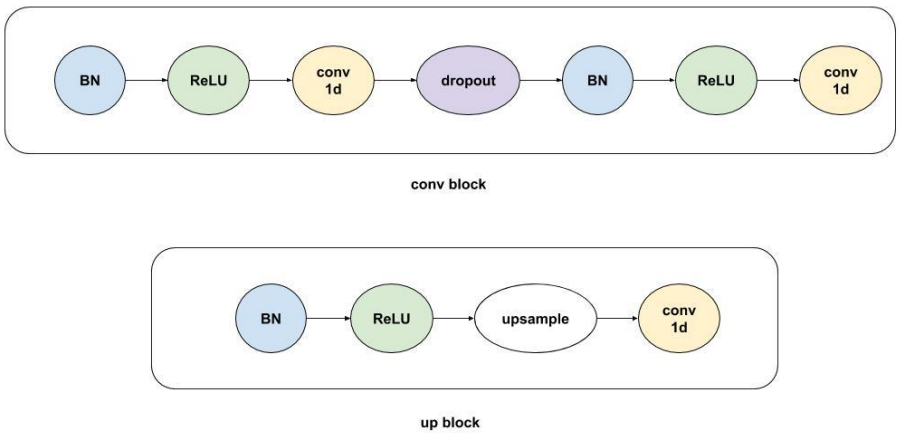
Σχήμα 4.3: Οι συνιστώσες του κομματιού της προσοχής (Attention block).

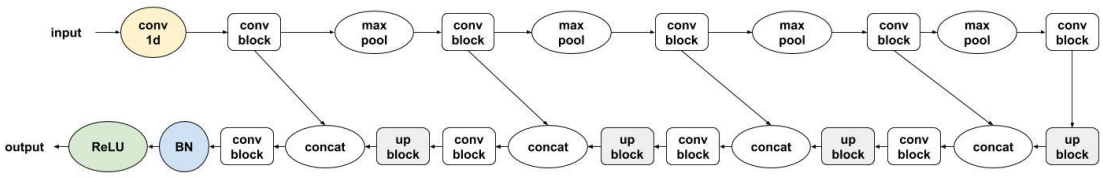


Σχήμα 4.4: LSTMs διπλής κατεύθυνσης με προσοχή.

Δίκτυο U-Net με συνελικτικά μέρη (convolutional blocks)

Στο σχήμα 4.5 που ακολουθεί παρουσιάζεται η αρχιτεκτονική του μοντέλου. Πρόκειται για ένα πλήρες συνδεδεμένο συνελικτικό μοντέλο, που χρησιμοποιεί ένα μονοδιάστατο U-Net (Olaf Ronneberger, 2015) με αποκλεισμό δικτύου (dropout) (Nitish Srivastava, 2014) και κανονικοποίηση συνόλων εκπαίδευσης (batch normalization) (Sergey Ioffe, 2015). Η μήτρα εισόδου που περιλαμβάνει τα προφίλς συνενώνεται με την έξοδο του στρώματος ενσωμάτωσης (embedding layer) και προωθείται στο πρώτο στρώμα του μονοδιάστατου U-Net.

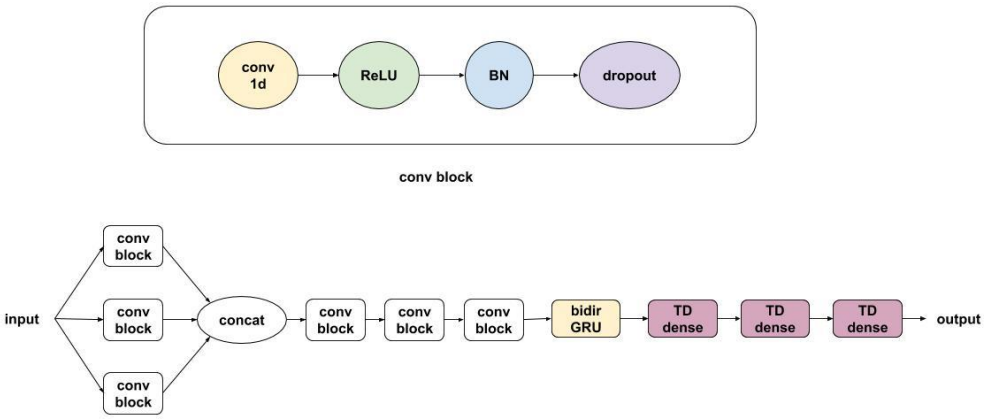




Σχήμα 4.5: U-Net με συνελικτικά μέρη (convolutional blocks).

Φραγμένο αναδρομικό νευρωνικό δίκτυο (Gated Recurrent Neural Network – GRU) διπλής κατεύθυνσης (Bidirectional) με συνελικτικά μέρη (convolutional blocks)

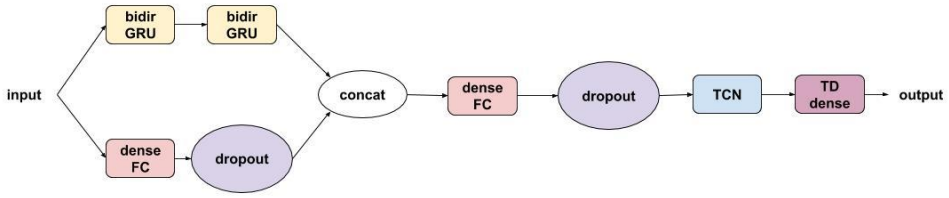
Στο Σχήμα 4.6 που ακολουθεί φαίνεται η αρχιτεκτονική του μοντέλου. Η συνένωση του κωδικοποιημένου υπολείμματος σε μορφή one-hot, της ενσωμάτωσης των υπολειμμάτων και των χαρακτηριστικών προφίλς των υπολειμμάτων μεταφέρονται σε συνελικτικά στρώματα πολλών κλιμάκων με διαφορετικό πυρήνα μεγέθους (3, 5, 7) για να αποκτήσουν πολλαπλές τοπικές απεικονίσεις των χαρακτηριστικών των συμφραζομένων (Zhen Li, 2016). Ακολουθεί μια σειρά επικαλυπτικών στρωμάτων συνέλιξης (cascading convolutional layers). Μια σειρά από 3 συνενωμένες μονοδιάστατες συνελίξεις (concatenated 1D convolutions) εφαρμόζονται. Κάθε συνέλιξη ακολουθείται από διάφορα στρώματα (Cholle, 2017): χρονικά κατανεμημένη ενεργοποίηση ReLU, κανονικοποίηση συνόλων εκπαίδευσης (batch normalization) και στρώματα αποκλεισμού δικτύου (dropout layers) (με ποσοστό 0.5). Αυτό περνάει από ένα Φραγμένο αναδρομικό νευρωνικό δίκτυο διπλής κατεύθυνσης (Bidirectional GRU) (Junyoung Chung, 2014) με 256 μονάδες με ένα l_2 επαναλαμβανόμενη κανονικοποίηση (recurrent regularizer). Η έξοδος παράγεται από δύο πλήρως συνδεδεμένα στρώματα ενεργοποίησης ReLU (μεγέθους 128 και 64) ακολουθούμενα από ένα στρώμα εξόδου softmax.



Σχήμα 4.6: GRU διπλής κατεύθυνσης με συνελικτικά μέρη (convolutional blocks).

Χρονικό Συνελικτικό Δίκτυο (Temporal convolutional network)

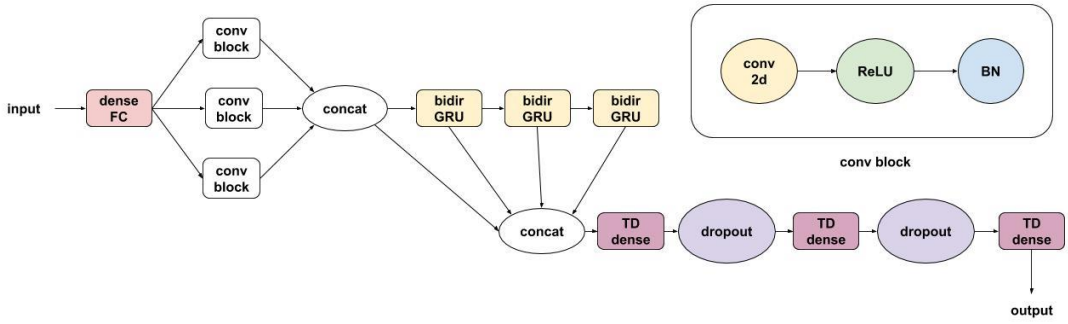
Στο σχήμα 4.7 που ακολουθεί παρουσιάζεται η αρχιτεκτονική του μοντέλου. Δύο στρώματα ενσωμάτωσης που τροφοδοτούνται με τους χαρακτήρες των αρχικών δεδομένων συνενώνονται με τα χαρακτηριστικά των προφίλς. Μια τέτοια συνενωμένη έξοδος τροφοδοτείται σε ένα πυκνό στρώμα (dense layer) ακολουθούμενο από αποκλεισμό δικτύου (dropout). Μια άλλη συνενωμένη έξοδος τροφοδοτείται σε δύο Φραγμένα Επαναλαμβανόμενα νευρωνικά δίκτυα διπλής κατεύθυνσης (Bidirectional GRUs). Αυτά τα δύο ξεχωριστά στρώματα (Dense, biGRU) συνενώνονται. Η έξοδος τροφοδοτείται σε ένα πυκνό στρώμα, ακολουθούμενο από αποκλεισμό δικτύου, ένα χρονικό συνελικτικό δίκτυο (Aäron Van Den Oord, 2016) και ένα χρονικά κατανεμημένο πυκνό στρώμα με ενεργοποίηση softmax.



Σχήμα 4.7: Χρονικό Συνελικτικό δίκτυο (Temporal Convolution Network, TCN).

Φραγμένο αναδρομικό νευρωνικό δίκτυο (Gated Recurrent Neural Network – GRU) διπλής κατεύθυνσης (Bidirectional) με δισδιάστατες συνελίξεις (2D convolutions)

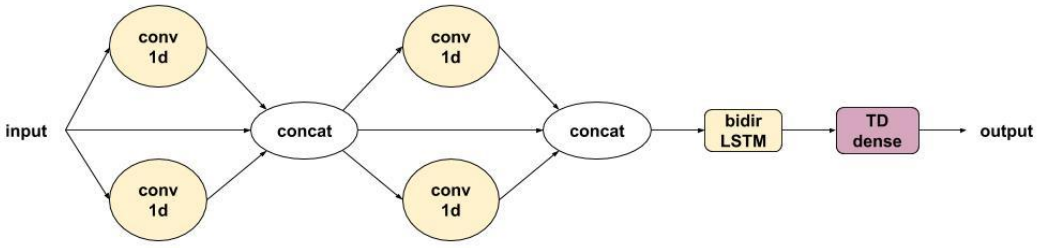
Στο σχήμα 4.8 που ακολουθεί φαίνεται η αρχιτεκτονική του μοντέλου. Το μοντέλο συνενώνει τα ακόλουθα χαρακτηριστικά στην είσοδο: ένα γραμμικό συνδυασμό των διανυσμάτων σε κωδικοποίηση one-hot των προηγούμενων αμινοξέων, ένα γραμμικό συνδυασμό των διανυσμάτων σε κωδικοποίηση one-hot των ακόλουθων αμινοξέων, το διάνυσμα σε κωδικοποίηση one-hot που αντιστοιχεί στο τρέχον αμινοξύ και τα χαρακτηριστικά προφίλς για το τρέχον αμινοξύ. Ένα πλήρως συνδεδεμένο στρώμα (με 128 μονάδες) αφαιρεί την αραιότητα από τα χαρακτηριστικά και οι έξοδοί του τροφοδοτούνται σε τρία στρώματα συνέλιξης πυρήνων (3, 7, 11) με 64 φίλτρα το καθένα. Μετά την κανονικοποίηση των συνόλων εκπαίδευσης των εξόδων (batch normalization), συνενώνονται και περνούν μέσω μιας στοίβας από τρία Bidirectional GRUs (με 32 μονάδες το καθένα). Η συνένωση των εξόδων των GRU με τις εξόδους των συνελικτικών στρωμάτων περνά μέσα από ένα πλήρως συνδεδεμένο δίκτυο δύο επιπέδων.



Σχήμα 4.8: GRUs διπλής κατεύθυνσης.

Συνελίξεις και Δίκτυο Μακράς βραχυχρόνιας μνήμης (Long Short-term Memory – LSTM) διπλής κατεύθυνσης (Bidirectional)

Στο σχήμα 4.9 που ακολουθεί φαίνεται η αρχιτεκτονική του μοντέλου. Το μοντέλο χρησιμοποιεί συνδέσεις παράκαμψης (skip connections), τροφοδοτώντας την κωδικοποιημένη είσοδο, σε δύο ανεξάρτητα στρώματα συνέλιξης 64 καναλιών το καθένα (με μέγεθος πυρήνα 11 και 7 αντίστοιχα). Στη συνέχεια, συνενώνονται και τα δύο με την είσοδο. Χρησιμοποιούνται και πάλι δύο ανεξάρτητα συνελικτικά στρώματα κάθε ένα με 64 κανάλια (με μέγεθος πυρήνα 5 και 3 αντίστοιχα). Και πάλι, συνενώνεται η είσοδος από την προηγούμενη συνένωση και η έξοδος των δύο συνελικτικών στρωμάτων. Στη συνέχεια, αυτή η συνένωση τροφοδοτείται σε ένα Bidirectional LSTM που παράγει μια έξοδο 128 μονάδων που τελικά χρησιμοποιείται για την παραγωγή της εξόδου χρησιμοποιώντας ένα πυκνό στρώμα (dense layer).



Σχήμα 4.9: Συνελίξεις και LSTM διπλής κατεύθυνσης.

Πειράματα και Αποτελέσματα

Οι αρχιτεκτονικές δικτύων που αναλύθηκαν παραπάνω εκπαιδεύτηκαν στο σύνολο δεδομένων CB6133 και η απόδοσή τους μετρήθηκε στο σύνολο ελέγχου CB513. Επιπλέον, δημιουργήθηκε ένας συνδυασμός αυτών των μοντέλων (τεχνική ensemble) η οποία και εμφάνισε την

υψηλότερη ακρίβεια μεταξύ των επιμέρους αρχιτεκτονικών. Τα αποτελέσματα εμφανίζονται αναλυτικά στον ακόλουθο πίνακα.

Πίνακας 4.1: Παρουσιάζεται η ακρίβεια των 6 μοντέλων που αναλύθηκαν και του συνδυασμού τους στο σύνολο CB513. *(Iddo Drori, 2018)*

Μέθοδος	Ακρίβεια(%)
Ensemble	**70.7**
Bidirectional GRU with convolution blocks	69.8
U-Net with convolution blocks	69.2
Temporal convolutional network	68.7
Bidirectional LSTMs with attention	68.4
Convolutions and Bidirectional LSTM	67.8
Bidirectional GRUs	67.4

4.2.2 Εφαρμογή Αναλλοίωτου ως προς την επεξεργασία Νευρωνικού Δικτύου (EINN) και βαθιών Συνελικτικών Νευρωνικών Δικτύων (CNNs) με συνενώσεις (concatenations)

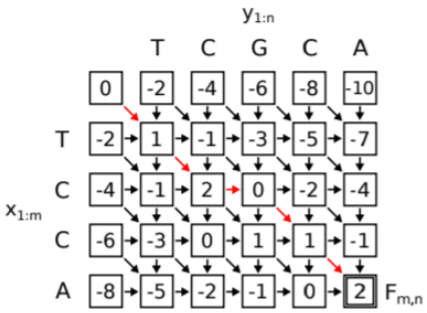
Η αναλλοίωτη ιδιότητα (invariance), η οποία αναγκάζει ένα μοντέλο πρόβλεψης να ικανοποιήσει μια επιθυμητή ιδιότητα για ένα συγκεκριμένο έργο, είναι σημαντική στα νευρωνικά δίκτυα. Για παράδειγμα, τα Συνελικτικά Νευρωνικά Δίκτυα (ΣΝΔ) με συγκεντρωτικά στρώματα (pooling layers) έχουν αναλλοίωτη ιδιότητα ως προς τη μετατόπιση (shift invariance) που θεωρείται σημαντική ιδιότητα για εργασίες αναγνώρισης εικόνων. Τα ΣΝΔ ήταν τα πρώτα που προτάθηκαν για να μιμηθούν την οργάνωση του οπτικού φλοιού (Fukushima, 1980). Αυτό χρησιμοποιείται συχνά για να εξηγήσει γιατί τα ΣΝΔ δουλεύουν σε εργασίες όρασης. Ομοίως, μελετάται και η αναλλοίωτη περιστροφή (rotation invariance) για εργασίες εικόνας (D. E. Worrall, 2017). Γενικά, είναι σημαντικό να μοντελοποιούνται οι κατάλληλες αναλλοίωτες ιδιότητες για ένα συγκεκριμένο πεδίο εφαρμογής.

Ποια είναι λοιπόν η αμετάβλητη ιδιότητα σε βιολογικές εργασίες; Όπως είναι γνωστό στην βιοπληροφορική, παρόμοιες αλληλουχίες τείνουν να παρουσιάζουν παρόμοιες λειτουργίες ή δομές (δηλαδή παρόμοιες ετικέτες κλάσεων από την άποψη της μηχανής μάθησης). Εδώ, η ομοιότητα εκτιμάται από την ευθυγράμμιση αλληλουχίας (sequence alignment), η οποία

σχετίζεται στενά με την επεξεργασία της απόστασης. Αυτό υπονοεί ότι οι ετικέτες που σχετίζονται με τις βιολογικές αλληλουχίες παρουσιάζουν μικρή αναλλοίωτη ιδιότητα σε σχέση με έναν μικρό αριθμό επεξεργασιών, δηλαδή εργασίες όπως η υποκατάσταση, η εισαγωγή και η διαγραφή. Το άρθρο "Neural Edit Operations for Biological Sequences" (Koide S., 2018) που δημοσιεύτηκε το 2018 έχει ως στόχο να ενσωματώσει τέτοιες αναλλοίωτες ιδιότητες, που ονομάστηκαν αναλλοίωτες ιδιότητες ως προς την επεξεργασία ακολουθιών (invariance edit), σε νευρωνικά δίκτυα. Σε αυτό δημιουργούνται δύο αρχιτεκτονικές νευρωνικών δικτύων που ενσωματώνουν τις λειτουργίες επεξεργασίας. Πρώτον, προτείνονται τα αναλλοίωτα ως προς την επεξεργασία νευρωνικά δίκτυα (Edit Invariant Neural Networks), με την αναπαραγωγή του αλγορίθμου Needleman-Wunsch (S. Needleman, 1970) ως παραγωγίσιμο (differentiable) νευρωνικό δίκτυο. Στη συνέχεια, αποδεικνύεται ότι τα βαθιά ΣΝΔ με αλληλουχίες μπορούν να χειριστούν κανονικές εκφράσεις χωρίς αστέρια Kleene, υποδεικνύοντας ότι τέτοια ΣΝΔ μπορούν να αντιληφθούν λειτουργίες επεξεργασίας, συμπεριλαμβανομένης της εισαγωγής/διαγραφής. Η ανάλυση που ακολουθεί βασίζεται πλήρως στη δουλειά που παρουσιάζεται στο αντίστοιχο άρθρο (Koide S., 2018).

Αναλλοίωτα ως προς την επεξεργασία Νευρωνικά Δίκτυα (Edit Invariant Neural Networks - EINNS)

Παραγωγίσιμη ευθυγράμμιση ακολουθίας (differentiable sequence alignment): Στη βιοπληροφορική, η ευθυγράμμιση αλληλουχιών είναι το κλειδί στη σύγκριση δύο βιολογικών ακολουθιών (π.χ. DNA, πρωτεϊνών). Ο αλγόριθμος NeedlemanWunsch (NW) (S. Needleman, 1970), ένας θεμελιώδης αλγόριθμος ευθυγράμμισης, υπολογίζει τη βαθμολογία ομοιότητας μεταξύ δύο ακολουθιών σε ένα αλφάβητο Σ. Όπως φαίνεται στο Σχήμα 4.10, η βαθμολογία ομοιότητας υπολογίζεται μέσω δυναμικού προγραμματισμού για τη μεγιστοποίηση της συνολικής βαθμολογίας (απεικονίζεται ως το τετράγωνο διπλού περιγράμματος) εισάγοντας ή διαγράφοντας χαρακτήρες (που απεικονίζονται ως κάθετα και οριζόντια βέλη, αντίστοιχα).



Σχήμα 4.10: Η ευθυγράμμιση NW. Η κόκκινη γραμμή είναι ένα μονοπάτι που μεγιστοποιεί τη βαθμολογία. Ο αριθμός των κελιών αντιστοιχεί στο $F_{i,j}$ στον Αλγόριθμο 1. *(Koide S., 2018, σ. 2)*

Αν και ο αρχικός αλγόριθμος NW είναι μια συνάρτηση που χρησιμοποιεί ακολουθίες ως ορίσματα, επεκτείνεται σε μια παραγωγίσιμη συνάρτηση χρησιμοποιώντας ενσωμάτωση. Ο Αλγόριθμος 1 δείχνει τον προτεινόμενο αλγόριθμο δυναμικού προγραμματισμού για τον υπολογισμό του σκορ NW $s_{NW}(x_{1:m}, y_{1:n}; g)$. Εδώ, η κλιμακωτή παράμετρος g αντιπροσωπεύει το κόστος εισαγωγής ή διαγραφής ενός χαρακτήρα. Οι διαφορές από τον αρχικό αλγόριθμο NW είναι οι εξής:

1. Οι ακολουθίες εισόδου $x_{1:m} = [x_1, \ldots, x_m]$ και $y_{1:n} = [y_1, \ldots, y_n]$ είναι d διάστασης ακολουθίες (δηλαδή τα x_i, y_j είναι διανύσματα στον $\mathbb{R}^d$) μήκους m και n αντίστοιχα.
2. Μετά την παραπάνω τροποποίηση, η συνάρτηση βαθμολογίας (score function) ορίζεται ως το εσωτερικό γινόμενο αντί για τον προκαθορισμένο πίνακα αναζήτησης a (Γραμμή 7 – Αλγόριθμος 1).
3. Η χρήση της συνάρτησης softmax $max^{\gamma}(x) = \gamma log(\sum_i exp(x_i/\gamma))$ αντί για τη συνάρτηση hard max (Γραμμή 10 – Αλγόριθμος 1).

Ακολούθως, παρατίθεται ο **Αλγόριθμος 1**:

Algorithm 1: Differentiable Needleman-Wunsch (forward): $s_{\text{NW}}(x_{1:m}, y_{1:n}; g)$

```
1  F ← 0;        // (m+2)x(n+2) zero matrix
2  for i = 0 ··· m do
3      F_{i,0} ← −ig
4  for j = 1 ··· n do
5      F_{0,j} ← −jg;
6      for i = 1 ··· m do
7          a ← F_{i−1,j−1} + x_i · y_j;
8          b ← F_{i−1,j} − g;
9          c ← F_{i,j−1} − g;
10         F_{i,j} ← max^γ (a, b, c)
11 return F_{m,n} as s_NW(x_{1:m}, y_{1:n}; g)
```

(Koide S., 2018, σ. 2)

Ο δυναμικός προγραμματισμός στον Αλγόριθμο 1 μπορεί να θεωρηθεί ως υπολογιστικός γράφος, επιτρέποντάς μας να διαφοροποιήσουμε τη βαθμολογία ομοιότητας NW $s_{NW}(x_{1:m}, y_{1:n}; g)$ ως προς τα $x_{1:m}$, $y_{1:n}$ και g. Καταρχήν, είναι δυνατό να εφαρμοστεί αυτόματη παραγώγιση για να υπολογιστεί η κλίση. Για να αποφευχθεί το υπολογιστικό κόστος που υπεισέρχεται από τα προς τα πίσω μονοπάτια στον υπολογιστικό γράφο, τροποποιήθηκε ο προς τα πίσω υπολογισμός με ορισμένες αλγεβρικές αντικαταστάσεις. Έτσι, μπορούν να υπολογιστούν οι παράγωγοι χρησιμοποιώντας δυναμικό προγραμματισμό, όπως φαίνεται στους Αλγορίθμους 2 και 3 που ακολουθούν. Με την μήτρα Q που υπολογίζεται στον Αλγόριθμο 2, μπορούμε να υπολογίσουμε την παράγωγο της βαθμολογίας NW ως προς τα $x_{1:m}$, $y_{1:n}$ ως εξής:

$$\frac{\partial s_{NW}}{\partial x_i} = \sum_{j=1}^{n} Q_{i,j} exp(H_{i,j}/\gamma) \cdot y_j \,, \frac{\partial s_{NW}}{\partial y_i} = \sum_{j=1}^{n} Q_{i,j} exp(H_{i,j}/\gamma) \cdot x_i$$

$$\text{όπου } H_{i,j} := F_{i-1,j-1} + x_i \cdot y_j - F_{i,j}$$

Αυτές οι παράγωγοι προκύπτουν παρόμοια με το SoftDTW (M. Cuturi, 2017), μια παραγωγίσιμη συνάρτηση απόστασης που αντιστοιχεί στη δυναμική χρονική κάλυψη (συνεπώς δεν εμπλέκεται το gapcost). Για την παράγωγο σε σχέση με το gapcost g, μπορεί να χρησιμοποιηθεί η μήτρα P στον Αλγόριθμο 3: $\frac{\partial s_{NW}}{\partial x_i} = P_{m,n}$. Όπως και στον αρχικό αλγόριθμο NW, η προτεινόμενη μέθοδος μπορεί να εξετάσει τις εισαγωγές / διαγραφές. Είναι γνωστό ότι το σκορ NW συνδέεται στενά με την απόσταση επεξεργασίας. Δεδομένων των ακολουθιών $x_{1:m}$, $y_{1:n}$ θα εξεταστεί μια τροποποιημένη ακολουθία $x'_{1:(m-1)}$ όπου ένα διάνυσμα χαρακτηριστικών x_t διαγράφεται από $x_{1:m}$. Σε μια τέτοια περίπτωση, τα υπολογιζόμενα αποτελέσματα $s_{NW}(x, y)$ και $s_{NW}(x', y)$ παρουσιάζουν παρόμοια αξία. Ονομάζουμε αυτήν την ιδιότητα την αναλλοίωτη ιδιότητα ως προς την επεξεργασία (edit invariance), η οποία αναμένεται να είναι σημαντική για εργασίες που περιλαμβάνουν βιολογικές ακολουθίες.

Ακολούθως, παρατίθενται οι **Αλγόριθμοι 2, 3**:

Algorithm 2: Calculation of Q (backward). We denote $\varphi_\gamma(a, b) := \exp((a-b)/\gamma)$.

$Q \leftarrow 0$; // (m+2) x (n+2) zero matrix
for $i = 1 \cdots m$ **do**
  $F_{i,n+1} \leftarrow \infty$
$F_{m+1,n+1} \leftarrow F_{m,n}$; $Q_{m+1,n+1} \leftarrow 1$;
for $j = n \cdots 1$ **do**
  $F_{m+1,j} \leftarrow \infty$;
  for $i = m \cdots 1$ **do**
    $a \leftarrow \varphi_\gamma(F_{i,j} + x_i \cdot y_j, F_{i+1,j+1})$;
    $b \leftarrow \varphi_\gamma(F_{i,j} - g, F_{i+1,j})$;
    $c \leftarrow \varphi_\gamma(F_{i,j} - g, F_{i,j+1})$;
    $Q_{i,j} \leftarrow aQ_{i+1,j+1} + bQ_{i+1,j} + cQ_{i,j+1}$
return Q

Algorithm 3: Calculation of P. We denote $\varphi_\gamma(a, b) := \exp((a-b)/\gamma)$.

$P \leftarrow 0$; // (m+2) x (n+2) zero matrix
for $i = 0 \cdots m$ **do**
  $P_{i,0} \leftarrow -i$
for $j = 1 \cdots n$ **do**
  $P_{0,j} \leftarrow -j$;
  for $i = 1 \cdots m$ **do**
    $a \leftarrow \varphi_\gamma(F_{i-1,j-1} + x_i \cdot y_j, F_{i,j})$;
    $b \leftarrow \varphi_\gamma(F_{i-1,j} - g, F_{i,j})$;
    $c \leftarrow \varphi_\gamma(F_{i,j-1} - g, F_{i,j})$;
    $P_{i,j} \leftarrow$
    $aP_{i-1,j-1} + b(P_{i-1,j} - 1) + c(P_{i,j-1} - 1)$
return P

(Koide S., 2018, σ. 3)

Συνελικτικό Αναλλοίωτο ως προς την επεξεργασία Νευρωνικό Δίκτυο (Convolutional EINN)

Τα παραδοσιακά ΣΝΔ επεκτείνονται με το σκορ NW που παρουσιάστηκε παραπάνω. Έστω μια ενσωματωμένη ακολουθία $X \in \mathbb{R}^{d \times L}$ μήκους L, και ένα συνελικτικό φίλτρο $w \in \mathbb{R}^{d \times K}$ με μέγεθος πυρήνα K. Έστω $x \in \mathbb{R}^{d \times K}$ ένα πλαίσιο μήκους K σε μια συγκεκριμένη θέση στην

ενσωματωμένη ακολουθία X. Στα ΣΝΔ, η ομοιότητα υπολογίζεται από το εσωτερικό γινόμενο (Frobenius), δηλαδή $w \cdot x$. Η ιδέα του συγκεκριμένου άρθρου είναι η αντικατάσταση της ομοιότητας που βασίζεται στο εσωτερικό γινόμενο με το παραπάνω προτεινόμενο $s_{NW}(x, w; g)$. Λαμβάνοντας ένα όριο καθώς $g \to \infty$ (δηλαδή η εισαγωγή και η διαγραφή απαγορεύονται), το s_{NW} σχετίζεται με συνέλιξη ως εξής:

Για κάθε $x \in \mathbb{R}^{d \times K}$ και κάθε $w \in \mathbb{R}^{d \times K}$, αποδεικνύεται ότι $w \cdot x = \log_{g \to \infty} s_{NW}(x, w; g)$ (Koide S., 2018).

Βαθιά Συνελικτικά Νευρωνικά Δίκτυα για την αναγνώριση Κανονικών Εκφράσεων

Αρχικά, αποδεικνύεται η σχέση μεταξύ των ΣΝΔ και των κανονικών εκφράσεων. Πρώτα, δίνεται ο ορισμός των εκφράσεων χωρίς άστρο Kleene, το οποίο είναι ένα υποσύνολο των τυπικών κανονικών εκφράσεων.

Ορισμός 1. Η κανονική έκφραση χωρίς άστρο Kleene είναι ένα σύνολο συμβολοσειρών σε ένα ορισμένο αλφάβητο Σ που ορίζεται αναδρομικά όπως ακολουθεί. Αρχικά, παρατίθενται οι κανονικές εκφράσεις χωρίς το άστρο Kleene:

1. Κενό σύνολο $\emptyset$.
2. Κενή συμβολοσειρά ε.
3. Ένας απλός χαρακτήρας $\forall a \in \Sigma$. Στη συνέχεια, έστω R και S τυπικές κανονικές εκφράσεις χωρίς άστρο Kleene. Τότε, τα ακόλουθα σύνολα συμβολοσειρών είναι επίσης κανονικές εκφράσεις χωρίς άστρο Kleene.
4. Συνένωση των συνόλων R και S, που δηλώνεται από το RS.
5. Μια ένωση των συνόλων R και S, που σημειώνεται από το $R|S$ (που ονομάζεται εναλλαγή, alternation). Επιπλέον, δεδομένης μιας συμβολοσειράς q, λέγεται ότι η q ταιριάζει με το R αν η q περιλαμβάνεται στο σύνολο R.

Εν ολίγοις, αυτό είναι ισοδύναμο με τις τυπικές κανονικές εκφράσεις χωρίς το άστρο Kleene, R^*, που δέχεται δυνητικά άπειρες επαναλήψεις των συμβολοσειρών στο R. Ακολουθώντας αυτόν τον ορισμό, μπορεί να δειχτεί ότι τα σύνολα συμβολοσειρών που αντιπροσωπεύονται από τα δύο μοτίβα που αναφέρθηκαν παραπάνω είναι οι κανονικές εκφράσεις χωρίς το άστρο Kleene.

Παρατίθενται κάποιοι συμβολισμοί εκφράσεων: Η κανονική έκφραση /a.b/ περιγράφει συμβολοσειρές όπως "a, ακολουθούμενο από οποιονδήποτε χαρακτήρα, ακολουθούμενο από b." Επιπλέον, η έκφραση /a[bc]a/, σημαίνει συμβολοσειρές όπως "a, ακολουθούμενο από b ή c, ακολουθούμενο από a," και /(abc|ac)/ σημαίνει "abc ή ac". Είναι αξιοσημείωτο ότι η τελευταία κανονική έκφραση /(abc|ac)/ είναι ίση με /ab?c/, όπου "?" σημαίνει μηδέν ή μία

εμφάνιση του προηγούμενου συμβόλου. Επειδή το άστρο Kleene "*" δεν λαμβάνεται εδώ υπόψιν, κανονικές εκφράσεις όπως "/ab*/", που περιγράφουν "a ακολουθούμενο από οποιοδήποτε αριθμό b" δεν λαμβάνονται επίσης υπόψιν.

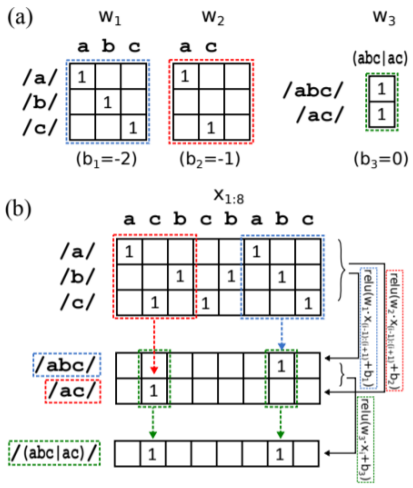
Απλές Κανονικές Εκφράσεις με Συνελικτικά Νευρωνικά Δίκτυα

Εδώ, αποκαλύπτεται η σχέση μεταξύ κανονικών εκφράσεων και των ΣΝΔ. Δίνεται ένα απλό παράδειγμα για να ελεγχθεί αν μια δεδομένη συμβολοσειρά εισόδου x μήκους L σε ένα αλφάβητο $\Sigma = \{a, b, c\}$ αντιστοιχεί σε μια κανονική έκφραση /abc/ για κάθε θέση. Έστω ότι χρησιμοποιείται μια κωδικοποίηση one-hot για την x, όπου κάθε διάσταση αντιστοιχεί σε έναν χαρακτήρα στο Σ.

Δημιουργείται ένα μονοδιάστατο συνελικτικό στρώμα, του οποίου ο πίνακας φίλτρου w_1 και η σταθερά bias [5] b_1 δίνονται από τη μία κωδικοποίηση one-hot $w_1 = (e_a, e_b, e_c)$ και $b_1 = -2$, αντίστοιχα, όπου e_a είναι το one-hot διάνυσμα του χαρακτήρα "a". Ο πίνακας φίλτρου w_1 φαίνεται στο Σχήμα 4.11(a). Χρησιμοποιώντας αυτό το φίλτρο, το η έξοδος του στρώματος στη θέση i είναι 1 αν η $x_{(i-1):(i+1)}$ ταιριάζει με την κανονική έκφραση /abc/, ή μικρότερη από 1 διαφορετικά (βλ. Σχήμα 4.11(b)). Επομένως, με τη χρήση ενεργοποίησης ReLU (δηλαδή $relu(w_1 \cdot x_{(i-1):(i+1)} - b_1)$ επιστρέφεται 1 για αντιστοίχιση και 0 για μη αντιστοίχιση. Αυτό δείχνει ότι μπορεί να μιμηθεί το ακριβές πρότυπο αντιστοίχισης χρησιμοποιώντας ένα μονό στρώμα μονοδιάστατων συνελίξεων. Για διευκόλυνση, η συνέλιξη υποδηλώνεται από μια πλειάδα (w_1, b_1).

Για παράδειγμα, για την αναγνώριση της κανονικής έκφρασης /ac/ μπορεί να χρησιμοποιηθεί ένα μονοδιάστατο συνελικτικό στρώμα μεγέθους πυρήνα $k = 3$ που αποτελείται από $w_2 = (e_a, e_c, 0)$ και $b_2 = -1$ (Σχήμα 4.11). Ομοίως, αν δοθεί μια κανονική έκφραση /ab?C/ = /(abc|ac)/ ως παράδειγμα, που αντιπροσωπεύει το μοτίβο 'abc', αλλά αποδέχεται τη διαγραφή του μεσαίου 'b'. Αυτό μπορεί να αναγνωριστεί από το παρακάτω πολυ-στρωματικό δίκτυο. Πρώτα, εφαρμόζονται δύο συνελίξεις (w_1, b_1) και (w_2, b_2).. Στη συνέχεια, χρησιμοποιώντας τις εξόδους αυτών των δύο φίλτρων ως είσοδο, ένα συνελικτικό στρώμα με μέγεθος πυρήνα 1 με την παράμετρο $w_3 = (e_{abc} + e_{ac}) = [1, 1]^T$ και $b_3 = 0$ εφαρμόζεται (βλέπε τον κάτω πίνακα στο Σχήμα 4.11(b)).

[5] Η παράμετρος bias είναι σαν τη σταθερά που προστίθεται σε μια γραμμική εξίσωση. Πρόκειται για μια επιπλέον παράμετρο στο Νευρικό Δίκτυο που χρησιμοποιείται για την προσαρμογή της εξόδου μαζί με το σταθμισμένο άθροισμα των εισόδων στον νευρώνα. Έτσι, η σταθερά αυτή βοηθά το μοντέλο να προσαρμοστεί καλύτερα στα δεδομένα εισόδου.



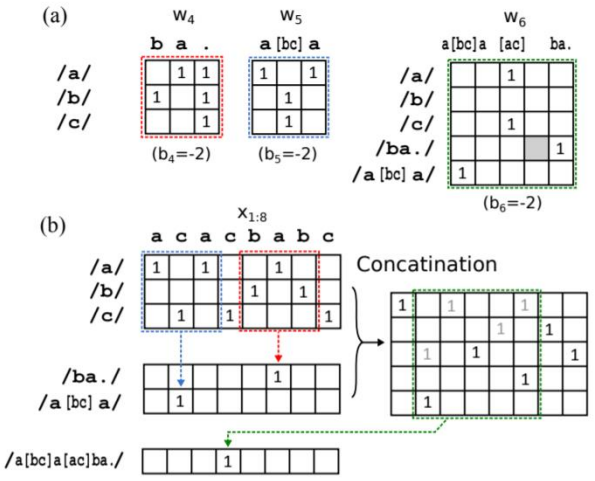
Σχήμα 4.11: Μονοδιάστατη συνελικτική αρχιτεκτονική που αποδέχεται μια κανονική έκφραση /(abc|ac)/. (a) Τα βάρη των συνελίξεων. (b) Εφαρμογή σε μια συμβολοσειρά acbcbabc. *(Koide S., 2018, σ. 5)*

Σχέση μεταξύ Συνελικτικών Νευρωνικών Δικτύων και Κανονικών Εκφράσεων χωρίς άστρο του Kleene

Δοθείσας μιας κανονικής έκφρασης χωρίς το άστρο του Kleene R, μπορεί να κατασκευαστεί ένα δυο επιπέδων συνελικτικό δίκτυο που δέχεται το R ομοίως. Έστω k το μέγιστο μήκος των συμβολοσειρών στο R. Το k είναι πεπερασμένο επειδή το R δεν περιλαμβάνει άστρο Kleene. Για τον ίδιο λόγο, το R είναι ένα πεπερασμένο σύνολο. Για κάθε συμβολοσειρά r στο R, κατασκευάζεται ένα συνελικτικό στρώμα με μέγεθος πυρήνα k που δέχεται r. Στη συνέχεια, οι έξοδοι αυτών των στρωμάτων είναι είσοδοι στο επόμενο στρώμα με μέγεθος πυρήνα 1, το οποίο πραγματοποιεί την λειτουργία OR (ένωση) παρόμοια με το φίλτρο (w_3, b_3) παραπάνω. Συνεπώς, διατυπώνεται η εξής πρόταση:

Πρόταση 2. (ΣΝΔ για την αναγνώριση κανονικής έκφρασης). Λαμβάνοντας μια κανονική έκφραση χωρίς άστρο Kleene R, υπάρχει ένα ΣΝΔ που μπορεί να ελέγξει αν μια δεδομένη συμβολοσειρά x αντιστοιχεί στο R για κάθε θέση της x.

Ωστόσο, η κατασκευή ενός ΣΝΔ είναι αναποτελεσματική όταν το $|R|$ είναι μεγάλο. Για παράδειγμα, ας θεωρήσουμε μια κανονική έκφραση /ba./, η οποία αποτελείται από $|\Sigma|$ συμβολοσειρές. Στην περίπτωση αυτή, η παραπάνω κατασκευή απαιτεί $|\Sigma|$ συνελικτικά φίλτρα. Στην πραγματικότητα, όμως μπορεί να αναπαρασταθεί από ένα μόνο φίλτρο, που αποτελείται από $w_4 = (e_b, e_a, e_a + e_b + e_c)$ και $b_4 = -2$ (Σχήμα 4.12 (a)). Επιπλέον, η έκφραση /a[bc]a/ αντιστοιχεί σε $w_5 = (e_a, e_b + e_c, e_a)$ και $b_5 = -2$ (Σχήμα 4.12 (a)). Αυτά τα παραδείγματα δείχνουν μια πιθανότητα ότι μια μεγάλη κανονική έκφραση R θα μπορούσε να συμπιεστεί σε ένα μικρό ΣΝΔ.



Σχήμα 4.12: Μονοδιάστατη συνελικτική αρχιτεκτονική που αποδέχεται μια κανονική έκφραση /a[bc]a[ac]ba./. (a) Τα βάρη των συνελίξεων. (b) Εφαρμογή σε μια συμβολοσειρά acabaabc. Τα 1 που δεν αντιστοιχούν στο w_6 σημειώνονται με γκρι. *(Koide S., 2018, σ. 5)*

Βαθύτερα Δίκτυα για πιο Σύνθετες Κανονικές Εκφράσεις

Σύμφωνα με την Πρόταση 2, ρηχά νευρικά δίκτυα μπορούν να αναγνωρίσουν μια αυθαίρετη κανονική έκφραση χωρίς το άστρο Kleene R. Εδώ, θα δούμε πώς το βάθος του νευρικού δικτύου σχετίζεται με τις κανονικές εκφράσεις. Περαιτέρω, ερευνάται η σημασία του DenseNet (G. Huang, 2017), ως η συνένωση (concatenation) των αποτελεσμάτων από διάφορα στρώματα. Το βάθος και η συνένωση είναι σημαντικά για την απόκτηση των κατανεμημένων αναπαραστάσεων των μοτίβων των συμβολοσειρών, όπως στην επεξεργασίας εικόνων. Συνδυάζοντας βαθιές συνελίξεις με συνένωση, μπορεί να κατασκευαστεί ένα μοντέλο που αναγνωρίζει εξαιρετικά πολύπλοκες κανονικές εκφράσεις από μικρά δομικά στοιχεία για απλές κανονικές εκφράσεις. Αυτό εξηγείται από το ακόλουθο παράδειγμα στο Σχήμα 4.12. Εκτός από τις ατομικές κανονικές εκφράσεις /a/, /b/ και /c/, έστω ότι δίνονται οι κανονικές εκφράσεις, /a[bc]a/ και /ba./, όπως αναφέρθηκε παραπάνω. Επιπλέον, έστω η κανονική έκφραση /a[bc]a[ac]ba./, η οποία είναι ακόμα πιο περίπλοκη, ένας συνδυασμός των παραπάνω. Αυτή η κανονική έκφραση μπορεί να χωριστεί σε τρία μέρη: 1) /a[bc] a / 2) /ba./ και 3) /[ac]/. Οι δύο πρώτες κανονικές εκφράσεις μπορούν να αναγνωριστούν από τα w_4 και w_5 που αναφέρθηκαν παραπάνω. Για να αναγνωριστεί η τελευταία, /[ac]/, χρησιμοποιείται η συνένωση δύο πινάκων, όπως φαίνεται στο Σχήμα 4.12(b). Συγκεκριμένα, χρησιμοποιώντας το συνελικτικό φίλτρο w_6 που φαίνεται στο Σχήμα 4.12(a) με $b = -2$, μπορεί να αναγνωριστεί η /a[bc]a[ac]ba./ όπως φαίνεται στον κάτω πίνακα στο Σχήμα 4.12(b). Ομοίως, αν το σκιασμένο στοιχείο του w_6 στο Σχήμα 4.12(a) είναι 1, μπορεί να αναπαρασταθεί μια άλλη κανονική έκφραση με διαγραφή, όπως η /a[bc]a[ac]?ba./.

Πειράματα και Αποτελέσματα

Στη συνέχεια αναλύονται τα διαφορετικά μοντέλα δικτύων που δοκιμάστηκαν στο συγκεκριμένο άρθρο καθώς και τα αποτελέσματα που έδωσαν στο πρόβλημα ΠΔΔΠ.

Απλοποιημένα Μοντέλα

Αρχικά, οι συγγραφείς μελέτησαν την επίδραση των Αναλλοίωτων ως προς την επεξεργασία Νευρωνικών Δικτύων **(**EINN) χρησιμοποιώντας απλοποιημένα μοντέλα και σύνολα δεδομένων. Χρησιμοποίησαν δύο τύπους μοντέλων: Ένα Μικρό ΣΝΔ (Tiny-CNN) και ένα Μικρό Αναλλοίωτο ως προς την επεξεργασία Νευρωνικό Δίκτυο (Tiny-EINN). Το Σχήμα 4.13(a) δείχνει το Tiny-CNN ενώ το Tiny-EINN λαμβάνεται αντικαθιστώντας τα στρώματα Conv-5 με το EINN συνελικτικά στρώματα που παρουσιάστηκαν παραπάνω.

Για την εκπαίδευση, χρησιμοποίησαν δεδομένα σε κωδικοποίηση one-hot για είσοδο και το 2% των δεδομένων εκπαίδευσης (ακολουθίες) που σύνολο δεδομένων CB6133. Χρησιμοποίησαν επίσης Adam optimizer, με μέγεθος συνόλων εκπαίδευσης (batches) 128, ρυθμό αρχικής εκμάθησης (learning rate) από 0,0002 (μειωμένο κατά 1/10 στην εποχή 15), και αποσύνθεση βάρους [6](weight decay) (10-5). Η ακρίβεια στο σύνολο ελέγχου CB513 αφορά την εποχή 30 και σταθερό gapcost $g = 2.5$ που έδωσε και την καλύτερη απόδοση, με το Tiny-EINN να έχει μεγαλύτερη ακρίβεια από το Tiny-CNN. Τα αποτελέσματα παρουσιάζονται στον ακόλουθο πίνακα.

Πίνακας 4.2: Παρουσιάζεται η ακρίβεια των μικρών μοντέλων που δημιουργήθηκαν στο σύνολο CB513. *(Koide S., 2018)*

Μέθοδος	Ακρίβεια(%)
Tiny-CNN	42.0
Tiny-EINN (g = 2.5)	43.0

[6] Κατά την εκπαίδευση ενός νευρωνικού δικτύου, είναι συνηθισμένο να χρησιμοποιείται η αποσύνθεση βάρους (weight decay), με την οποία τα βάρη πολλαπλασιάζονται με ένα συντελεστή ελαφρώς μικρότερο από 1 μετά από κάθε ενημέρωση τους. Αυτό εμποδίζει την υπερβολικά μεγάλη αύξηση των βαρών και μπορεί να θεωρηθεί ως κάθοδος κλίσης σε έναν τετραγωνικό όρο κανονικοποίησης.

Βαθύτερα Μοντέλα

Στη συνέχεια, παρουσιάζονται τα αποτελέσματα για βαθιά μοντέλα, συμπεριλαμβανομένου ενός μοντέλου που επιτυγχάνει την υψηλότερη ακρίβεια που έχει επιτευχθεί στο CB513. Σε όλα τα πειράματα, χρησιμοποιήθηκε το RMSProp για βελτιστοποίηση, με τον αρχικό ρυθμό εκμάθησης 0.00033 και μέγεθος συνόλων εκπαίδευσης 8 (batch size).

Τα μοντέλα εκπαιδεύονται για 150 εποχές και αναφέρεται η ακρίβεια στο σύνολο δοκιμής στην τελευταία εποχή. Δεν χρησιμοποιείται αποσύνθεση βάρους και ο ρυθμός εκμάθησης μειώνεται κατά 1/10 στην εποχή 100. Δεν χρησιμοποιούνται άλλες τεχνικές που περιλαμβάνουν ταξινόμηση βασισμένη σε ακτινωτή αναζήτηση (beam-search based classification) (A. Busia, 2017) ή συνδυασμό μοντέλων (ensemble) (A. Busia, 2017), (Zhen Li, 2016). Επιπλέον, διαπιστώθηκε από τους συγγραφείς ότι η αύξηση των δεδομένων βελτιώνει την ακρίβεια. Για τη δημιουργία νέων δεδομένων εκπαίδευσης, αντικατέστησαν τυχαία επιλεγμένες θέσεις στο διάνυσμα σε κωδικοποίηση one-hot με ένα αμινοξύ που προέρχεται από ομοιόμορφη κατανομή. Στα πειράματά τους, αντικατέστησαν τυχαία το 15% των υπολειμμάτων, βελτιώνοντας την απόδοση έως και 0,8 μονάδες στο σύνολο ελέγχου.

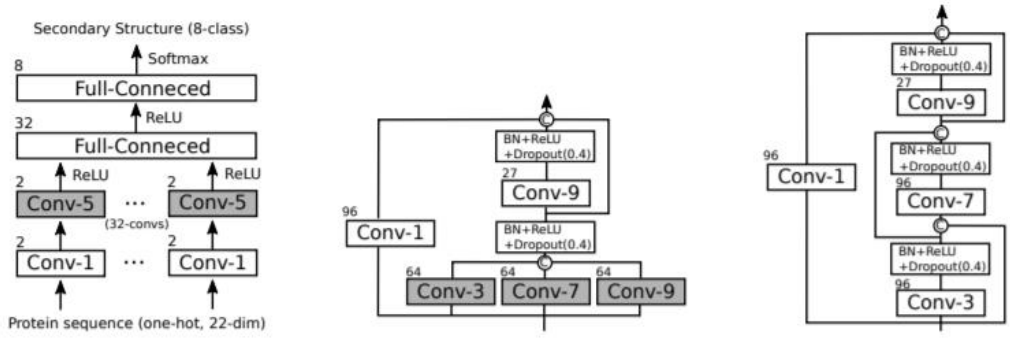
Σαν μοντέλο αναφοράς (baseline), χρησιμοποιήθηκε μια στοίβα από ConvBlocks που φαίνονται στο Σχήμα 4.13(b), παρόμοιο με το προηγούμενο επιτυχές μοντέλο που προτάθηκε στο (A. Busia, 2017). Σε αντίθεση με την αρχιτεκτονική αυτή, δεν χρησιμοποιήθηκε μη γραμμικότητα μετά Conv-1 επειδή διαπίστωσαν ότι επιδεινώνει την απόδοση όταν τοποθετείται σε μεγαλύτερο βάθος στο μοντέλο. Εφαρμόζονται πρώτα δύο ConvBlocks. Στη συνέχεια, σε κάθε θέση, εφαρμόζεται ένα πλήρως συνδεδεμένο στρώμα (μεγέθους 455), ακολουθούμενο από κανονικοποίηση συνόλων εκπαίδευσης (batch normalization), αποκλεισμό δικτύου (dropout σε ποσοστό p = 0,2) και συνάρτηση ενεργοποίησης ReLU. Τέλος, εφαρμόζεται ένα άλλο πλήρως συνδεδεμένο στρώμα που εξάγει τα αποτελέσματα των 8 κατηγοριών. Για να ερευνηθεί η επίδραση των EINN, αντικαθίστανται οι συνελίξεις στο Σχήμα 4.13(b) στο πρώτο ConvBlock με EINN με ίδιο μεγέθος φίλτρου και πυρήνα.

Πίνακας 4.3: Παρουσιάζεται η ακρίβεια των μοντέλων που δημιουργήθηκαν στο σύνολο CB513. Το σύμβολο ∗ αναφέρεται στη χρήση του μοντέλου για πολλαπλές εργασίες (multitasking) και το σύμβολο † στην εκπαίδευση του μοντέλου σε επαυξημένα δεδομένα. Η ονομασία MCNN αφορά τη χρήση τροποποιημένου ConvBlock (Σχήμα 4.13(c)), το οποίο περιλαμβάνει επίσης συνελικτικά στρώματα με συνενώσεις. *(Koide S., 2018)*

Μέθοδος	Ακρίβεια(%)
2-block CNN†	69.7
2- block EINN†	69.8
2-block CNN∗†	69.8
4-block CNN∗†	70.6

8-block CNN*†	71.2
12-block CNN*†	**71.5**
16-block CNN*†	71.3
8-block MCNN*†	71.3
12-block MCNN*†	**71.5**

Η ακρίβεια των δοκιμών για αυτά τα μοντέλα (CNN 2-block και 2-block EINN) δείχνουν ότι τα μοντέλα που βασίζονται σε αρχιτεκτονικές EINN είναι και πάλι καλύτερα, αλλά ο βαθμός βελτίωσης είναι πιο μικρός. Αυτό μπορεί να ερμηνευτεί ως εξής. Σύμφωνα με την ανάλυση παραπάνω (Βαθιά Συνελικτικά Νευρωνικά Δίκτυα για την αναγνώριση Κανονικών Εκφράσεων), το ίδιο το ConvBlock μπορεί αναγνωρίζει πολύπλοκα μοτίβα συμβολοσειρών. Είναι αδύνατο να αντικατασταθούν όλες οι συνελίξεις στο μοντέλο με EINNs για τους ακόλουθους λόγους. Πρώτον, τα EINN καταναλώνουν πολύ περισσότερη μνήμη απ' ότι τα ΣΝΔ, εμποδίζοντας έτσι την ευρεία εφαρμογή τους. Δεύτερον, ο χρόνος υπολογισμού των EINN είναι μεγαλύτερος από εκείνο των ΣΝΔ. Αν και έχουν εφαρμοστεί EINNs που χρησιμοποιούν μονάδα επεξεργασίας γραφικών (GPU), η ταχύτητα υπολογισμού είναι περισσότερο από δέκα φορές πιο αργή από αυτή των ΣΝΔ αν το μέγεθος του πυρήνα είναι $k = 5$. Αυτό οφείλεται στο γεγονός ότι ο εσωτερικός διπλός βρόχος του αλγορίθμου δεν μπορεί να παραλληλιστεί, με αποτέλεσμα να έχει πολυπλοκότητα του $O(k^2)$, ενώ σε ΣΝΔ ο υπολογισμός τρέχει σε $O(1)$ χρόνο χρησιμοποιώντας GPUs.



Σχήμα 4.13: Αρχιτεκτονικές του δικτύου. Το Conv-k είναι το μονοδιάστατο συνελικτικό στρώμα με μέγεθος πυρήνα k. Ο αριθμός πάνω αριστερά δείχνει το πλήθος των φίλτρων που χρησιμοποιούνται. Αντικαθιστώντας τις σκιασμένες συνελίξεις με τις συνιστώσες EINN του ίδιου μέγεθους πυρήνα, παίρνουμε το δίκτυο EINN. Εδώ το c σημαίνει συνένωση (concatenation) κατά μήκος της διάστασης του φίλτρου. (a) 32-convs σημαίνει μια συνέλιξη με 32 ομάδες. (b) Το ConvBlock στοιβάζεται βαθιά. Σε κάθε θέση, ένα πλήρως συνδεδεμένο στρώμα εφαρμόζεται για να δώσει ως έξοδο τις βαθμολογίες των 8 κλάσεων. (c) Ένα τροποποιημένο ConvBlock. (Koide S., 2018, σ. 9)

Κεφάλαιο 5

Τεχνικές Λεπτομέρειες Υλοποίησης του Μοντέλου του Μεταφραστή (Transformer) για την Πρόβλεψη της Δευτεροταγούς Δομής των Πρωτεϊνών (PSSP)

Στο κεφάλαιο αυτό παρατίθενται οι λεπτομέρειες υλοποίησης του Μοντέλου του Μεταφραστή για το πρόβλημα Πρόβλεψης της Δευτεροταγούς Δομής των Πρωτεϊνών. Αρχικά, στην Ενότητα 5.1 παρατίθενται κάποιες βασικές λεπτομέρειες της υλοποίησης του μοντέλου του Μεταφραστή (Transformer) σχετικά με τη διαχείριση των δεδομένων εκπαίδευσης και στην Ενότητα 5.2 συνοψίζονται τα προγραμματιστικά εργαλεία που βοήθησαν στην υλοποίηση του μοντέλου.

5.1 Λεπτομέρειες υλοποίησης

Προ-επεξεργασία συνόλων δεδομένων

Ως σύνολο εκπαίδευσης (train set) χρησιμοποιείται το CB6133 και ως σύνολο ελέγχου (test set) το CB513, για τα οποία έχουμε μιλήσει εκτενώς στο Κεφάλαιο 4 (ενότητα 4.1.5) και τα οποία χρησιμοποιήθηκαν στις επιτυχείς δουλειές που παρουσιάσαμε (Iddo Drori, 2018), (Koide S., 2018). Τα δεδομένα αυτά ανακτώνται ακολουθώντας τις οδηγίες που δίνονται στις δύο παραπάνω δουλειές και μετατρέπονται σε ακολουθίες, με μέγιστο μήκος 700 και αποθηκεύονται σε ακολουθίες από πεζά γράμματα (22 διαφορετικοί χαρακτήρες για τα πρωτεϊνικά υπολείμματα στις ακολουθίες εισόδου και 8 διαφορετικοί χαρακτήρες για την περιγραφή της Δευτεροταγούς Δομής στις ακολουθίες εξόδου), χωρίς κενά μεταξύ τους, για την ευκολότερη επεξεργασία τους στη συνέχεια. Όπως, αναφέρθηκε και σε προηγούμενα μοντέλα (ενότητα 4.2.1) που εφαρμόζουν ενσωματώσεις στην είσοδο, η μήτρα εισόδου που περιλαμβάνει τα επιπρόσθετα χαρακτηριστικά προφίλς συνενώνεται με την έξοδο του στρώματος ενσωμάτωσης (embedding layer) και προωθείται στα επόμενα μέρη του μοντέλου.

Δημιουργία Λεξικού για την εργασία της Μετάφρασης

Δημιουργήθηκε μια συνάρτηση, η οποία λαμβάνει τις ακολουθίες χαρακτήρων εισόδου και εξόδου, και παράγει τις διαφορετικές λέξεις μήκους n (n-grams extraction) που συνθέτουν κάθε ακολουθία που αντίστοιχα θα αντιμετωπιστεί σαν πρόταση από το μοντέλο του Transformer. Στα βασικά πειράματα που παρουσιάζονται στο Κεφάλαιο 6 κάθε λέξη ήταν ένας χαρακτήρας

(δηλαδή το λεξικό της πηγής, source, περιλάμβανε καταρχήν τις 22 λέξεις για τις ακολουθίες εισόδου και το λεξικό του στόχου, target, τις 8 λέξεις για τις ακολουθίες εξόδου). Επιπλέον χρησιμοποιήθηκαν στα δύο λεξικά οι επιπρόσθετοι χαρακτήρες, που αντιστοιχούν σε ειδικά tokens με συγκεκριμένη ερμηνεία και βοηθούν στην πλήρη πρόβλεψη κάθε παραγόμενης ακολουθίας εξόδου από ακολουθία εισόδου:

1. κενός χαρακτήρας (pad token) για το γέμισμα των ακολουθιών ώστε να έχουν όλες ίδιο μήκος ίσο με το μέγιστο (700)
2. άγνωστος χαρακτήρας (unknown token)
3. χαρακτήρας που δηλώνει την αρχή κάθε πρότασης
4. χαρακτήρας που δηλώνει το τέλος κάθε πρότασης

Οργάνωση και χρήση των συνόλων δεδομένων εκπαίδευσης και ελέγχου

Για την διευκόλυνση της εξαγωγής συμπερασμάτων για τη διαδικασία της εκπαίδευσης του μοντέλου κατά τη διάρκεια αυτής, από το σύνολο δεδομένων εκπαίδευσης (train set) λήφθηκε το 10% των δεδομένων το οποίο και αποτέλεσε το σύνολο επικύρωσης (validation set). Πρόκειται για ένα δείγμα δεδομένων που χρησιμοποιείται για την παροχή αμερόληπτης αξιολόγησης ενός μοντέλου στο σύνολο δεδομένων εκπαίδευσης, κατά τη ρύθμιση των υπερπαραμέτρων του μοντέλου, χωρίς να αναπροσαρμόζονται τα βάρη του μοντέλου βάσει αυτού. Το υπόλοιπο 90% του συνόλου δεδομένων CB6133 χρησιμοποιήθηκε για την εκπαίδευση του μοντέλου και την αναπροσαρμογή των βαρών του βάσει της συνάρτησης απωλειών (loss function) που υπολογίζεται σε αυτό και την επιθυμητή έξοδο σε κάθε εποχή. Επιπλέον, κατά την εκπαίδευση, σε κάθε εποχή το δίκτυο τροφοδοτείται με υποσύνολα των δεδομένων εκπαίδευσης (batches) (δηλαδή το σύνολο δεδομένων αντί να τροφοδοτηθεί ολόκληρο στο δίκτυο, μοιράζεται σε υποσύνολα, τα οποία ένα-ένα τροφοδοτούνται στο δίκτυο).

Διαδικασία Μετάφρασης για τη λήψη των τελικών προβλέψεων των ακολουθιών Δευτεροταγούς Δομής

Κατά τη διάρκεια της διαδικασίας της εκπαίδευσης πραγματοποιείται η προσαρμογή των βαρών των συνιστωσών του κωδικοποιητή και του αποκωδικοποιητή. Οι ακολουθίες εισόδου (source) σε κάθε εποχή διαδίδονται προς τα εμπρός στα επιμέρους στρώματα του κωδικοποιητή και στη συνέχεια περνούν στον αποκωδικοποιητή, όπου συνεχίζουν την πορεία τους μέσα στο δίκτυο μαζί με τις ακολουθίες εξόδου (target) που έχουν σε αυτό το σημείο περάσει από τις πολλαπλές κεφαλές προσοχής (masked multi-head attention) που είναι προσαρμοσμένες για την επεξεργασία τους (όπως φαίνεται στο Σχήμα 3.1 του Κεφαλαίου 3 και στην αντίστοιχη

ανάλυση που παρατίθεται σε αυτό). Για τη λήψη λοιπόν σωστών αποτελεσμάτων όσον αφορά την μετρική ακρίβειας που χρησιμοποιείται στην εκάστοτε εργασία με το μοντέλο του Transformer, είναι αναγκαία μετά την ολοκλήρωση της εκπαίδευσης να εφαρμοστεί μια διαδικασία μετάφρασης και υπολογισμού της ακρίβειας στο σύνολο δεδομένων ελέγχου (test set). Το εκπαιδευμένο μοντέλο, δηλαδή τα βάρη του, μετά το τέλος του της διαδικασίας της εκπαίδευσης αποθηκεύεται ώστε να χρησιμοποιηθεί στη συνέχεια για τη διαδικασία της μετάφρασης. Στη διαδικασία της μετάφρασης, το σύνολο δεδομένων ελέγχου και συγκεκριμένα οι ακολουθίες εισόδου (source) περνούν από τον εκπαιδευμένο κωδικοποιητή και στη συνέχεια από τον εκπαιδευμένο αποκωδικοποιητή. Το μοντέλα εξάγει μια κατανομή πιθανότητας πάνω από κάθε λέξη στο λεξιλόγιο για κάθε λέξη στην ακολουθία εξόδου. Στη συνέχεια, μια διαδικασία αποκωδικοποίησης καλείται να μετασχηματίσει τις πιθανότητες σε μια τελική ακολουθία λέξεων (με την έννοια της λέξεις εννοούμε στο πρόβλημα μας τα n-grams που έχουμε εξάγει κατά την προ-επεξεργασία των δεδομένων – 1 χαρακτήρας στην πιο απλή περίπτωση).

Το τελικό στρώμα στο μοντέλο νευρωνικού δικτύου έχει έναν νευρώνα για κάθε λέξη στο λεξιλόγιο εξόδου και χρησιμοποιείται μια λειτουργία ενεργοποίησης softmax για να εξάγει μια πιθανότητα κάθε λέξης στο λεξιλόγιο να είναι η επόμενη λέξη στην ακολουθία. Η αποκωδικοποίηση της πιο πιθανής ακολουθίας εξόδου περιλαμβάνει την αναζήτηση σε όλες τις πιθανές ακολουθίες εξόδου βάσει της πιθανότητας τους. Το μέγεθος του λεξιλογίου καθιστά το πρόβλημα αναζήτησης εκθετικό στο μήκος της ακολουθίας εξόδου και ανέφικτο (NP-complete πρόβλημα) στην περίπτωση πλήρους αναζήτησης. Στην πράξη, μέθοδοι ευριστικής αναζήτησης χρησιμοποιούνται για την επιστροφή μιας ή περισσοτέρων κατά προσέγγιση αποκωδικοποιημένων εξόδων για μια δεδομένη πρόβλεψη. Εδώ, θα αναλύσουμε τις δύο τεχνικές που εφαρμόζουμε στα πειράματα του μοντέλου:

1. **Άπληστη Αποκωδικοποίηση (Greedy Decoding):** Μια απλή προσέγγιση είναι η χρήση μια άπληστης αναζήτησης που επιλέγει την πιο πιθανή λέξη σε κάθε σημείο της ακολουθίας εξόδου. Αυτή η προσέγγιση έχει το πλεονέκτημα ότι είναι πολύ γρήγορη, αλλά η ποιότητα των τελικών ακολουθιών εξόδου μπορεί να απέχει πολύ από τη βέλτιστη.
2. **Ακτινική Αναζήτηση (Beam Search):** Μια άλλη συνηθισμένη ευριστική είναι η ακτινική αναζήτηση που επεκτείνει την ιδέα της άπληστης αναζήτησης και επιστρέφει μια λίστα πιθανών ακολουθιών εξόδου. Αντί να επιλεχθεί το πιο πιθανό επόμενο σημείο κατά την κατασκευή της ακολουθίας, η ακτινική αναζήτηση επεκτείνει όλα τα πιθανά επόμενα βήματα και διατηρεί τα k πιο πιθανά, όπου k είναι μια παράμετρος που καθορίζεται από το χρήστη και ελέγχει τον αριθμό των παράλληλων αναζητήσεων μέσω των πιθανοτήτων της ακολουθίας.

Βελτιστοποίηση (Optimizer) και Συνάρτηση Απωλειών (loss function) για την Εκπαίδευση και Υπολογισμός Ακρίβειας (Accuracy Metric)

Αλγόριθμος Βελτιστοποίησης (optimizer)

Ο στόχος της μηχανικής μάθησης και της βαθιάς μάθησης είναι η μείωση της διαφοράς μεταξύ της προβλεπόμενης εξόδου από ένα μοντέλο και της πραγματικής εξόδου, η οποία καλείται συνάρτηση απωλειών (loss function) και την έχουμε αναφέρει σε προηγούμενα Κεφάλαια. Με στόχο την ελαχιστοποίηση της συνάρτησης κόστους βρίσκοντας βελτιστοποιημένες τιμές για τα βάρη του δικτύου, πρέπει επίσης να διασφαλιστεί ότι ο αλγόριθμος γενικεύει καλά. Αυτό θα συμβάλει στην καλύτερη πρόβλεψη των δεδομένων στα οποία δεν έχει εκπαιδευτεί το μοντέλο. Οι αλγόριθμοι βελτιστοποίησης (optimization algorithms, optimizers) ενημερώνουν τις προς εκμάθηση παραμέτρους του μοντέλου για την ελαχιστοποίηση της συνάρτησης απωλειών. Για την εκπαίδευση του μοντέλου του Transformer χρησιμοποιείται ο αλγόριθμος βελτιστοποίησης Adam Optimizer με μεταβλητό ρυθμό εκμάθησης (learning rate) που χρησιμοποιήθηκε όταν εισηγήθηκε το μοντέλο στην μετάφραση κειμένου (Ashish Vaswani, 2017), η λειτουργία του οποίου αναλύθηκε στην ενότητα 3.2.

Συνάρτηση Απωλειών (loss function)

Η συνάρτηση απωλειών που χρησιμοποιήθηκε κατά τη διαδικασία της εκπαίδευσης για τη συγκεκριμένη εργασία είναι η cross-entropy. Ο αποκωδικοποιητής βγάζει ως έξοδο διανύσματα με πιθανότητες πάνω στο λεξιλόγιο $p_i \in \mathbb{R}^V$ για κάθε σημείο της ακολουθίας. Για κάθε ακολουθία εξόδου ή στόχο $y_1, \ldots, y_n$ μπορούμε να υπολογίσουμε τη συνολική πιθανότητα σαν το γινόμενο των επιμέρους πιθανοτήτων κάθε σημείου που παράγεται:

$$P(y_1, \ldots, y_n) = \prod_{i=1}^{m} p_i[y_i]$$

Όπου $p_i[y_i]$ σημαίνει ότι εξάγεται το y_i σημείο του διανύσματος πιθανοτήτων p_i από το i-οστό βήμα αποκωδικοποίησης. Συγκεκριμένα, μπορούμε να υπολογίσουμε την πιθανότητα της πραγματικής ακολουθίας στόχων. Ένα τέλειο σύστημα θα έδινε μια πιθανότητα 1 σε αυτήν την ακολουθία στόχο, οπότε πρόκειται να εκπαιδεύσουμε το δίκτυό μας για να μεγιστοποιήσουμε την πιθανότητα της ακολουθίας στόχου, η οποία είναι η ίδια με την ελαχιστοποίηση:

$$\log P(y_1, \ldots, y_n) = -\log \prod_{i=1}^{m} p_i[y_i] = -\sum_{i=1}^{n} \log p_i[y_i]$$

Που πρόκειται για τη συνάρτηση cross-entropy. Συνεπώς, ελαχιστοποιούμε την cross-entropy μεταξύ της κατανομής στόχου (όλων των θερμών διανυσμάτων) και της προβλεπόμενης κατανομής που εξάγεται από το μοντέλο μας.

Μετρική Ακρίβειας των προβλέψεων του Μοντέλου του Μεταφραστή

Η μετρική απόδοσης του συγκεκριμένου προβλήματος ορίστηκε στην ενότητα 4.1.6. Η μετρική αυτή χρησιμοποιείται για τον υπολογισμό της ακρίβειας (ως μέσο όρο των επιμέρους ακριβειών για κάθε προβλεπόμενη ακολουθία), μεταξύ των προβλέψεων για τις ακολουθίες εξόδου που παράγει η διαδικασία της μετάφρασης που αναλύθηκε παραπάνω και τις πραγματικές ακολουθίες δευτεροταγούς δομής που αντιστοιχούν στις ακολουθίες εισόδου (για έλεγχο χρησιμοποιούμε πάντα σύνολο δεδομένων CB513).

Επιπλέον, κατά την διάρκεια του training κρίθηκε χρήσιμος, εκτός από τον υπολογισμό της συνάρτησης απωλειών μεταξύ προβλέψεων και ακολουθιών στόχων εξόδου, ο υπολογισμός μιας ακρίβειας των προβλέψεων στο σύνολο επικύρωσης (validation set). Το σύνολο αυτό, όπως αναφέραμε, δεν χρησιμοποιείται για εκπαίδευση αλλά για να δώσει μια εικόνα της απόδοσης του μοντέλου κατά τη διάρκεια της εκπαίδευσης και να βοηθήσει στην ρύθμιση των υπερπαραμέτρων του μοντέλου. Συγκεκριμένα, σε κάθε εποχή της εκπαίδευσης, γίνεται υπολογισμός της συνάρτησης απωλειών στο σύνολο εκπαίδευσης (train set) και στο σύνολο επικύρωσης (validation set). Επιπλέον, σε κάθε εποχή υπολογίζεται η καθιερωμένη μετρική απόδοσης του προβλήματος Πρόβλεψης της Δευτεροταγούς Δομής στο σύνολο επικύρωσης. Η μετρική αυτή δεν περιμένουμε ότι δίνει μια ακριβή πληροφορία για την ακρίβεια της πρόβλεψης κατά την διαδικασία εκπαίδευσης (εφόσον σε αυτή ο αποκωδικοποιητής παίρνει ως είσοδο τις ακολουθίες στόχους), μιας και πρέπει να γίνει η πρόβλεψη της εξόδου με τη διαδικασία της μετάφρασης, που αναλύθηκε παραπάνω, για την εξαγωγή αμερόληπτων προβλέψεων. Παρόλα αυτά, δείχνει την τάση της απόδοσης του μοντέλου και ήταν ιδιαίτερα χρήσιμη για τη επιλογή μοντέλων για τελική μετάφραση, διευκολύνοντας την όλη διαδικασία των πειραμάτων που αναλύεται στο Κεφάλαιο 6. Μαζί με αυτή την ακρίβεια υπολογίζαμε, χωρίς να τη λαμβάνουμε υπόψη σε κάποια επιμέρους διαδικασία, την καθιερωμένη μετρική σε εργασίες μετάφρασης κειμένου, αυτή που χρησιμοποιήθηκε από τους (Ashish Vaswani, 2017) στο μοντέλο του Μεταφραστή.

Έγκαιρη Διακοπή της διαδικασίας Εκπαίδευσης (Early Stopping) για αποφυγή υπερπροσαρμογής (overfitting) στα δεδομένα εκπαίδευσης και αποθήκευση του καλύτερου μοντελου (Checkpoint)

Ένα πρόβλημα με την εκπαίδευση των νευρωνικών δικτύων είναι η επιλογή του αριθμού των εποχών εκπαίδευσης που πρέπει να χρησιμοποιηθούν. Πάρα πολλές εποχές μπορούν να οδηγήσουν σε υπερπροσαρμογή (overfitting) του μοντέλου στο σύνολο δεδομένων εκπαίδευσης (δυσκολευοντάς το να γενικεύσει στο σύνολο ελέγχου), ενώ πολύ λίγες μπορεί να οδηγήσουν σε ένα μοντέλο με κακή απόδοση. Η έγκαιρη διακοπή (Early Stopping) είναι μια μέθοδος που επιτρέπει να οριστεί ένας αυθαίρετα μεγάλος αριθμός εποχών εκπαίδευσης και η

οποία σταματά την εκπαίδευση μόλις η απόδοση του μοντέλου σταματήσει να βελτιώνεται σε ένα σύνολο δεδομένων επικύρωσης (validation set). Η συνάρτηση απώλειας (loss function) που επιλέγουμε για να βελτιστοποιήσουμε το μοντέλο (cross entropy στην προκειμένη) υπολογίζεται στο τέλος κάθε εποχής. Η κλήση του Early Stopping επιτρέπει τον καθορισμό του μέτρου απόδοσης για την παρακολούθηση και την διακοπή της εκπαίδευσης με βάση το κριτήριο που ορίζεται. Στην συγκεκριμένη εργασία, όπως και στις περισσότερες, επιδιώκουμε την ελάχιστη τιμή της απώλειας στο σύνολο επικύρωσης. Η εκπαίδευση με αυτή την τεχνική θα σταματήσει όταν το επιλεγμένο μέτρο απόδοσης σταματήσει να βελτιώνεται. Συχνά, το πρώτο σημάδι μη περαιτέρω βελτίωσης μπορεί να μην είναι η καλύτερη στιγμή για να σταματήσει την εκπαίδευση. Αυτό οφείλεται στο γεγονός ότι το μοντέλο μπορεί να βρεθεί για μια μόνο εποχή σε ένα σημείο χωρίς βελτίωση ή ακόμη και να χειροτερεύσει σε μια σειρά εποχών για να βελτιωθεί αισθητά στη συνέχεια. Το πρόβλημα αυτό αντιμετωπίζεται προσθέτοντας μια καθυστέρηση στην ενεργοποίηση της διακοπής της εκπαίδευσης, επιλέγοντας συγκεκριμένο αριθμό εποχών στις οποίες υπάρχει ανοχή στην έλλειψη βελτίωσης της απόδοσης. Αυτό μπορεί να γίνει με τη ρύθμιση μιας παραμέτρου που ονομάζεται $patience$. Η τιμή της παραμέτρου ποικίλει ανάλογα με τα μοντέλα και τα προβλήματα. Από προεπιλογή, οποιαδήποτε βελτίωση στο μέτρο απόδοσης, ανεξάρτητα από την ακριβή τιμή, θα θεωρηθεί βελτίωση.

Η τεχνική Early Stopping θα σταματήσει την εκπαίδευση μόλις ενεργοποιηθεί, αλλά το μοντέλο στο τέλος της εκπαίδευσης μπορεί να μην είναι το μοντέλο με τις καλύτερες επιδόσεις στο σύνολο δεδομένων επικύρωσης. Απαιτείται μια επιπλέον μέθοδος που θα σώσει το καλύτερο μοντέλο που παρατηρήθηκε κατά τη διάρκεια της εκπαίδευσης για μελλοντική χρήση. Αυτή είναι η τεχνική της αποθήκευσης του μοντέλου (Model Checkpoint) (δηλαδή των βαρών του καλύτερου με κάποιο κριτήριο μοντέλου). Συνήθως αποθηκεύεται το καλύτερο ή τα καλύτερα μοντέλα που παρατηρούνται κατά τη διάρκεια της εκπαίδευσης, όπως ορίζεται από ένα επιλεγμένο μέτρο απόδοσης στο σύνολο δεδομένων επικύρωσης. Στην εργασία αυτή, και τα πειράματα που παρατίθενται στο επόμενο κεφάλαιο, το Model Checkpoint αποθηκεύει τα βάρη του μοντέλου που εμφάνισε το μικρότερο loss στο σύνολο δεδομένων επικύρωσης σε όλη τη διάρκεια της εκπαίδευσης.

5.2 Πλατφόρμες και προγραμματιστικά εργαλεία

Εργαλεία και Συνεισφορές στην Υλοποίηση του Μοντέλου του Μεταφραστή (Transformer)

Το μοντέλο του Μεταφραστή έχει εφαρμοστεί (όπως έχουμε εξηγήσει στο Κεφάλαιο 3) κυρίως σε εργασίες μετάφρασης κειμένου (με δομή προτάσεων και λέξεων) από μια γλώσσα σε μια άλλη. Μαζί με το άρθρο που εισήγαγε για πρώτη φορά αυτό το μοντέλο (Ashish Vaswani, 2017) δημοσιεύτηκε και ο αντίστοιχος κώδικας, που χρησιμοποιήθηκε για τα διάφορα πειράματα που παρατέθηκαν, στο σύνδεσμο https://github.com/ tensorflow/tensor2tensor. Η δοθείσα υλοποίηση πραγματοποιήθηκε στην προγραμματιστική γλώσσα Python, με χρήση της βιβλιοθήκης μηχανικής μάθησης TensorFlow[7]. Στη συγκεκριμένη διπλωματική εργασία, ο κώδικας που υλοποιεί το μοντέλο του Μεταφραστή προσαρμοσμένο στο πρόβλημα πρόβλεψης Δευτεροταγούς Δομής των Πρωτεϊνών (ΠΔΔΠ) υλοποιήθηκε σε Python και με τη χρήση της βιβλιοθήκης Pytorch[8]. Η επιλογή της συγκεκριμένης βιβλιοθήκης έγκειται στην ευκολία χρήση της σε επίπεδο υλοποίησης, την πληθώρα εργαλείων και επεκτάσεων που διαθέτει και το γεγονός ότι υπήρχε προηγούμενη εμπειρία στη χρήση της σε εργασίες μηχανικής μάθησης.

Αρχικά, η ανάπτυξη της υλοποίησης των βασικών μερών του μοντέλου, βασίστηκε στην καθοδήγηση και σε κομμάτια κώδικα που παρατίθενται στο tutorial “The annotated Transformer” στο σύνδεσμο https://nlp.seas.harvard.edu/2018/04/03/attention.html από την ομάδα HarvardNlp, που έκαναν μια απόπειρα μετατροπής του πρωτο-δημοσιευμένου κώδικα με τη βιβλιοθήκη TensorFlow σε χρήση της βιβλιοθήκης Pytorch, δίνοντας και σχετικές επεξηγήσεις για τη λειτουργία των διαφόρων επιμέρους κομματιών του μοντέλου. Εφόσον επιτεύχθηκε η εξοικείωση με το μοντέλο και η επιτυχής ανάπτυξη των βασικών του συνιστωσών, η αναζήτηση μιας πιο δομημένης υλοποίησης οδήγησε στην εύρεση ενός

[7] Το TensoFlow είναι μια βιβλιοθήκη ανοιχτού κώδικα για τον αριθμητικό υπολογισμό και την μηχανική μάθηση μεγάλης κλίμακας που δημιουργήθηκε από την ομάδα Google Brain (Google Brain team). Δημοσιεύτηκε τον Νοεμβρίου του 2015. Επιτρέπει υλοποίηση σε γλώσσες Python C++.

[8] Το Pytorch είναι μια βιβλιοθήκη ανοιχτού κώδικα για μηχανική μάθηση βασισμένη στη βιβλιοθήκη Torch και χρησιμοποιείται για εφαρμογές όπως όραση υπολογιστών και επεξεργασία φυσικής γλώσσας. Αναπτύχθηκε πρωτίστως από την ερευνητική ομάδα τεχνητής νοημοσύνης του Facebook. Πρόκειται για δωρεάν λογισμικό ανοιχτού κώδικα που δημοσιεύτηκε τον Οκτώβρη του 2016. Παρόλο που η υλοποίηση σε γλώσσα Python ήταν ο κύριος στόχος της ανάπτυξης αυτού του εργαλείου, υλοποιείται επίσης σε γλώσσα προγραμματισμού C++.

δημοσιευμένου κώδικα στο Github[9], στο σύνδεσμο https://github.com/jadore801120/attention-is-all-you-need-pytorch, που έχει δεχτεί θετικές κριτικές από χιλιάδες χρήστες που το έχουν χρησιμοποιήσει για την ανάπτυξη του μοντέλου Transformer. Αυτή η υλοποίηση, που παρέχει μια πολύ δομημένη οργάνωση των διαφόρων κομματιών και διαδικασιών του μοντέλου, προσαρμόστηκε τελικά στη λήψη και την επεξεργασία των δεδομένων των πρωτεϊνικών ακολουθιών για την μετάφρασή τους στην δευτεροταγή δομή τους.

Διαχείριση Υπολογιστικών Πόρων και Βελτιστοποίηση της επεξεργασίας των δεδομένων

Στον τομέα της βαθιάς μηχανικής μάθησης, ο παραλληλισμός των δεδομένων είναι η πιο διαδεδομένη μέθοδος για τη διαίρεση των εργασιών εκπαίδευσης των μοντέλων μεταξύ των πολλαπλών GPU. Αρχικά, θα εξηγήσουμε πώς ο παραλληλισμός των δεδομένων λειτουργεί με τη χρήση μικρών υποσυνόλων εκπαίδευσης (mini batches) και τη μέθοδο στοχαστικής καθόδου κλίσης (stochastic gradient decent) ως παράδειγμα. Υποθέτουμε ότι υπάρχουν πολλαπλές μονάδες GPU σε μια μηχανή. Με βάση το μοντέλο που θα εκπαιδευτεί, κάθε GPU θα διατηρήσει ανεξάρτητα ένα πλήρες σύνολο παραμέτρων του μοντέλου. Σε κάθε επανάληψη της εκπαίδευσης του μοντέλου, με δεδομένο τυχαίο mini batch, διαιρούμε τα παραδείγματα στο batch σε k τμήματα και κάθε τμήμα διανέμεται σε κάθε GPU. Στη συνέχεια, κάθε GPU θα υπολογίσει την τοπική κλίση των παραμέτρων του μοντέλου με βάση το υποσύνολο mini batches που της έχει ανατεθεί και τις παραμέτρους μοντέλου που διατηρεί. Στη συνέχεια, προσθέτουμε μαζί τις τοπικές κλίσεις στις k GPU για να πάρουμε την τρέχουσα στοχαστική κλίση του mini batch. Μετά από αυτό, κάθε GPU χρησιμοποιεί αυτή τη στοχαστική κλίση του mini batch για να ενημερώσει το πλήρες σύνολο των παραμέτρων του μοντέλου που διατηρεί.

Η χρήση πολλαπλών GPU για τη διαδικασία εκπαίδευσης (multi-GPU training) έχει τα εξής πλεονεκτήματα:

- Η χρήση μεγάλων μοντέλων βαθιών Νευρωνικών Δικτύων, πολλά εκ των οποίων καταλαμβάνουν τεράστιο χώρο στη μνήμη, έχει συχνά αποτέλεσμα να μην μπορούν αυτά να χωρέσουν σε μια κανονική GPU. Το επιτρέπουν multi-GPU training επιτρέπει σε πολλαπλές GPU να μοιραστούν την υπολογιστική μνήμη της διαδικασίας εκπαίδευσης, επιτρέποντας την εκπαίδευση μεγαλύτερων δικτύων.

[9] Το GitHub είναι μια υπηρεσία που φιλοξενεί ανοικτό πηγαίο κώδικα. Υποστηρίζει τον πηγαίο κώδικα διαφόρων εργασιών χρηστών σε διαφορετικές γλώσσες προγραμματισμού και παρακολουθεί, αποθηκεύει τις διάφορες αλλαγές που γίνονται κατά την ανάπτυξη της υλοποίησης.

- Η επιτάχυνση της εκπαίδευσης των βαθιών Νευρωνικών Δικτύων είναι επίσης ένα πολύ θετικό αποτέλεσμα της χρήσης πολλαπλών GPU, όπως προκύπτει από την παραλληλοποίηση που εξηγήθηκε παραπάνω.

Οι μηχανισμοί προσοχής που περιλαμβάνει το μοντέλο του Μεταφραστή (Transformer) απαιτούν μια σύνθετη υπολογιστική διαδικασία, με μεγάλο κόστος σε επίπεδο της μνήμης που καταλαμβάνει το μοντέλο. Για το σκοπό αυτό το multi-GPU training καθίσταται αναγκαίο, σε περιπτώσεις που το μέγεθος του μοντέλου αυξάνεται (για πλήθος N στρωμάτων κωδικοποιητή/αποκωδικοποιητή μεγαλύτερο από 2). Για να είναι εφικτά και αποδοτικά τα πειράματα που περιγράφονται στο επόμενο κεφάλαιο, πραγματοποιήθηκε multi-GPU training κατά την εκπαίδευση. Χρησιμοποιήθηκαν 2 GPUs GeForce RTX 2080 Ti (11Gb Memory), 1545 Mhz.

Κεφάλαιο 6

Αξιολόγηση προτεινόμενου Μοντέλου και Ρύθμιση Υπερπαραμέτρων

Στο Κεφάλαιο 5, αναλύθηκαν εκτενώς οι λεπτομέρειες υλοποίησης του μοντέλου του Μεταφραστή (Transformer) για το πρόβλημα Πρόβλεψης της Δευτεροταγούς Δομής Πρωτεϊνών (ΠΔΔΠ). Βάσει των λεπτομερειών και των μεθόδων που εξηγήθηκαν, στις επόμενες ενότητες παρατίθενται τα αποτελέσματα απόδοσης του μοντέλου σε διάφορα σύνολα υπερπαραμέτρων καθώς και το μοντέλο που πέτυχε την υψηλότερη ακρίβεια πρόβλεψης στο σύνολο ελέγχου.

6.1 Υπερπαράμετροι προς Ρύθμιση

Στο Κεφάλαιο 3 επεξηγήθηκε η αρχιτεκτονική του μοντέλου του Μεταφραστή καθώς και οι υπερπαράμετροι που προκύπτουν από τις διάφορες συνιστώσες που περιέχει. Τις αναφέρουμε συνοπτικά μιας και τα πειράματα που ακολουθούν πραγματοποιήθηκαν για τη ρύθμιση τους:

- d_model: ο αριθμός των αναμενόμενων χαρακτηριστικών των διανυσμάτων στις εισόδους του κωδικοποιητή και του αποκωδικοποιητή, που συμπίπτει με το μέγεθος των ενσωματώσεων που παράγουν τα στρώματα ενσωμάτωσης (embedding layers).
- d_inner_hid: η διάσταση του δικτύου πρόσθιας τροφοδότησης (Feed Forward Nural Network) που ακολουθεί τα στρώματα πολλαπλών κεφαλών προσοχής (multi-head attention) στις συνιστώσες κωδικοποιητή – αποκωδικοποιητή.
- d_k, d_v: διάσταση του πίνακα των κλειδιών (keys) και διάσταση του πίνακα των τιμών (values) αντίστοιχα, που χρησιμοποιούνται στα στρώματα προσοχής πολλαπλών κεφαλών.
- N: ο αριθμός των συνιστωσών κωδικοποιητών και αποκωδικοποιητών, που μπορεί να μην ταυτίζεται απαραίτητα.
- h: το πλήθος των κεφαλών στα στρώματα πολλαπλών κεφαλών προσοχής, που περιέχονται τόσο στο κομμάτι των κωδικοποιητών όσο και στο κομμάτι των αποκωδικοποιητών του δικτύου.
- dropout: το ποσοστό αποκλεισμού του δικτύου κατά την εκπαίδευση.
- batch_size: το μέγεθος (πλήθος παραδειγμάτων) των υποσυνόλων δεδομένων που χρησιμοποιούνται κατά την εκπαίδευσης.

- epoch: το μέγιστο πλήθος εποχών για την εκπαίδευση του μοντέλου (μέγιστο μιας και η διαδικασία της εκπαίδευσης παρακολουθείται και μπορεί να διακοπεί νωρίτερα από την μέθοδο (Early Stopping) που εξηγήθηκε στην ενότητα 6.1)
- Οι σταθερές β_1, β_2 και e για τον Adam Optimizer και το πλήθος βημάτων $warmup\_steps$ που καθορίζουν την αύξηση ή μείωση του ρυθμού εκμάθησης κατά τη διάρκεια της εκπαίδευσης, όπως αναλύθηκε στην ενότητα 3.2, στην παράγραφο για την μέθοδο Βελτιστοποίησης που χρησιμοποιήθηκε. Για το ρυθμό εκμάθησης χρησιμοποιείται ο τύπος γραμμικής μεταβολής του $lrate = d_{model}^{-0.5} \cdot min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$, όπως προτάθηκε στο άρθρο που εισήγαγε το μοντέλο.
- label_smoothing: αν η λειτουργία εξομάλυνσης ετικετών είναι ενεργοποιημένη ή όχι μαζί με την παράμετρο e_{ls} που την καθορίζει.

Συνολικά, λοιπόν, έχουμε 15 παραμέτρους που επιθυμούμε να ρυθμίσουμε για την επίτευξη όσο το δυνατόν μεγαλύτερης απόδοσης στο πρόβλημα που επιλύουμε, συνεπώς θέλει ιδιαίτερη προσοχή στην επιλογή των υπερπαραμέτρων προς εξέταση για τη διεξαγωγή των πειραμάτων, ώστε αυτά να οδηγήσουν στην εύρεση ενός αποτελεσματικού μοντέλου, μετά από ένα λογικό χρονικό διάστημα δοκιμής των διαφορετικών συνδυασμών τους.

6.2 Σύστημα αξιολόγησης και Οργάνωση των πειραμάτων

Αξιολόγηση μοντέλων

Για την αξιολόγηση της αποδοτικότητας των προς εξέταση μοντέλων χρησιμοποιούνται, αρχικά, οι τιμές της συνάρτησης απωλειών και της μετρικής απόδοσης του προβλήματος PSSP (που εξηγήθηκαν στο Κεφάλαιο 5) στο σύνολο επικύρωσης κατά την εκπαίδευση. Αυτές μας δίνουν ένα δείγμα της πορείας της εκπαίδευσης των διαφόρων μοντέλων και επιτρέπουν τις συγκρίσεις μεταξύ τους, ώστε για την χρονοβόρα διαδικασία της μετάφρασης να επιλέγονται σε κάθε σύνολο πειραμάτων τα πιο υποσχόμενα μοντέλα.

Μελέτη επίδρασης υπερπαραμέτρων και ρύθμιση των διαστημάτων αναζήτησης τιμών

Για τη διευκόλυνση της διεξαγωγής των πειραμάτων, σε πρώτη φάση ελέγχθηκε η επιρροή των διαφόρων παραμέτρων στην απόδοση του μοντέλου. Συγκεκριμένα, μετά από τυχαία αναζήτηση των συνδυασμών των παραμέτρων σε ένα μεγάλο σύνολο τιμών, ορίστηκαν κάποια διαστήματα τιμών στα οποία αυτές ελέγχθηκαν με οργανωμένα πειράματα, με εξαντλητική αναζήτηση των διαφόρων συνδυασμών για την τελική επιλογή τους (τα οποία πειράματα

παρατίθενται στη συνέχεια). Από τα αποτελέσματα της τυχαίας αναζήτησης αποφασίστηκε να τεθούν σταθερές οι εξής παράμετροι κατά τη διάρκεια της εξαντλητικής αναζήτησης που ακολούθησε για τους υπόλοιπους συνδυασμούς παραμέτρων:

- $epoch = 200$, μιας και παρατηρήθηκε ότι για διαφορετικούς συνδυασμούς των υπολοίπων παραμέτρων, η γραφική παράσταση της συνάρτησης απωλειών του μοντέλου ανά εποχή συγκλίνει σε σταθερή τιμή (καταδεικνύοντας ότι περίπου μετά από αυτό τον αριθμό εποχών η διαδικασία της εκπαίδευσης πρέπει να διακοπεί εμποδίζοντας την υπερπροσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης).
- $patience = 10, 20$, στη μέθοδο Early Stopping που αναλύεται στην ενότητα 5.1, για τη διακοπή του μοντέλου μετά από 10 ή 20 εποχές χωρίς βελτίωση του κριτηρίου απόδοσης που έχουμε επιλέξει.
- $warmup\_steps = 4000$, $\beta_1 = 0.9$, $\beta_2 = 0.98$ και $e = 10^{-9}$ για τη λειτουργία του Adam Optimizer με μεταβλητό ρυθμό εκμάθησης (learning rate). Καταλήξαμε στις συγκεκριμένες τιμές κατά την διάρκεια της τυχαίας αναζήτησης, μιας και με αυτές οι καμπύλες της συνάρτησης απωλειών σε κάθε εποχή, έδειξαν μια ομαλή πτώση της απώλειας του μοντέλου και μια σταθερή βελτίωση της απόδοσης κατά τη διάρκεια της εκπαίδευσης, δείχνοντας ότι μπορεί να συμβάλει στην γενίκευση του μοντέλου και σε καλύτερη απόδοση στο σύνολο των δεδομένων ελέγχου.
- $e_{ls} = 0.1$ σε περίπτωση που ενεργοποιείται η επιλογή της εξομάλυνσης ετικετών

Με τις παραπάνω τιμές σταθερές, προχωρήσαμε σε ορισμό των τιμών που θα ελεγχθούν σε εξαντλητική αναζήτηση για τις υπόλοιπες 9 παραμέτρους (d_model, d_inner_hid, d_k, d_v, N, h, dropout, batch_size, label_smoothing=True/False), που φάνηκε από τα πρώτα τυχαία πειράματα που τρέξαμε ότι επηρεάζουν σημαντικά τη συμπεριφορά του μοντέλου στο πρόβλημα για το οποίο εκπαιδεύεται.

Επίσης κατά τη διάρκεια της τυχαίας αναζήτησης, παρατηρήθηκε ότι συγκεκριμένοι συνδυασμοί των τιμών των παραμέτρων d_model, d_inner_hid, d_k, d_v, οδηγούν σε καλύτερες περιγραφές των ακολουθιών εισόδου στο χώρο χαρακτηριστικών και στη λήψη ακριβέστερων συσχετίσεων μεταξύ διαφορετικών λέξεων μέσω της συνάρτησης προσοχής. Το άρθρο που εισήγαγε το μοντέλο του Μεταφραστή (Ashish Vaswani, 2017), προτείνει τις εξής σχέσεις μεταξύ των μεταβλητών για καλύτερη απόδοση: $d\_inner\_hid = 4 \cdot d\_model, d\_k = d\_v = d\_model/h$. Ως πλήθος πολλαπλών κεφαλών προσοχής ερευνήθηκαν μεγάλες τιμές, καθώς λόγω του μεγάλου μήκους των ακολουθιών εισόδου (~700) μεγαλύτερα h μπορούν να αποκωδικοποιήσουν αποτελεσματικότερα τις εξαρτήσεις μεταξύ λέξεων. Πράγματι από την τυχαία αναζήτηση καλά αποτελέσματα έδινε η τιμή $h = 8$ (αισθητά καλύτερα από μικρότερες τιμές που δοκιμάστηκαν), ενώ σε ένα σύνολο πειραμάτων δοκιμάστηκε και μια πολύ μεγάλη

τιμή $h = 16$. Όσον αφορά το μέγεθος των υποσυνόλων εκπαίδευσης (batch_size) επιλέχθηκαν μικρές τιμές (10, 18 ή 20), μιας και παρατηρήθηκε ότι οδηγούσαν σε καλύτερα αποτελέσματα στη συνάρτηση απωλειών και στη μετρική κατά την εκπαίδευση και επίσης διευκόλυναν την εκπαίδευση του μοντέλου από άποψη μνήμης (μιας και ο συνδυασμός μεγάλων σε μέγεθος παραμέτρων ενδεχομένως να οδηγήσει και σε αδυναμία εκπαίδευσης λόγω περιορισμών μνήμης των GPUs, ακόμα και με multi-GPU training).

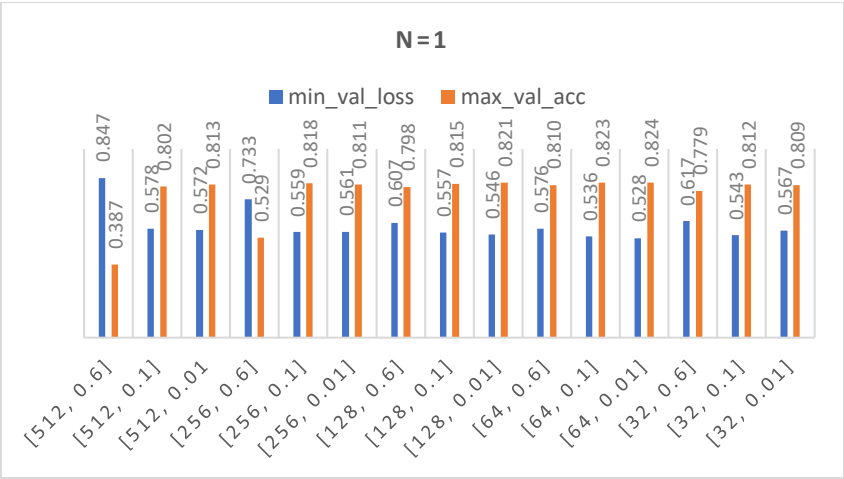
1. Στο πρώτο σύνολο πειραμάτων εξαντλητικής αναζήτησης δοκιμάστηκαν οι τιμές: $h = 8,16$, $d\_inner\_hid = 4 \cdot d\_model$, $d\_k = d\_v = d\_model/4$, $dropout = 0.01,0.1,0.6$, $N = 1,2,3,4,5,6$, $d\_model = 32,64,128,256,512$, $label\_smoothing = True, False$, και $batch\_size = 10$. Κάποια μοντέλα με υψηλά ποσοστά στο σύνολο επικύρωσης δοκιμάστηκαν επίσης με $batch\_size = 18$.
2. Στο δεύτερο σύνολο πειραμάτων εξαντλητικής αναζήτησης δοκιμάστηκαν οι τιμές: $h = 8$, $d\_inner\_hid = 4 \cdot d\_model$, $d\_k = d\_v = d\_model/2$, $dropout = 0.01,0.1$, $N = 1,2$, $d\_model = 32,64,128,256,512$, $label\_smoothing = False$, και $batch\_size = 20$.

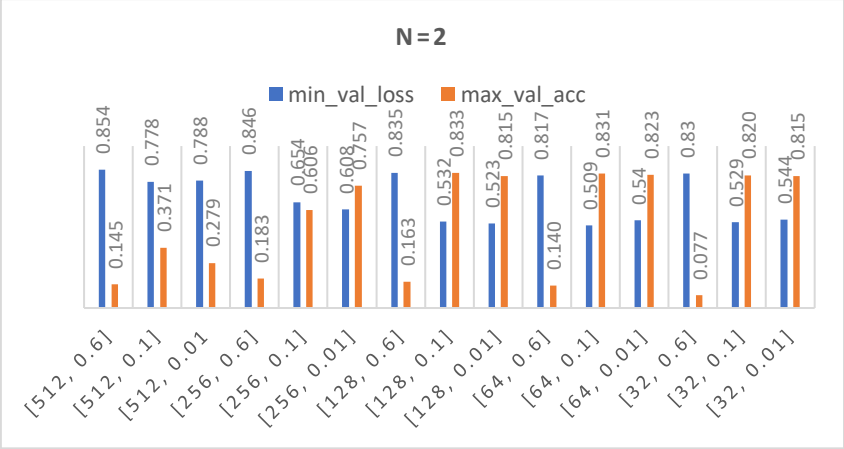
6.3 Αποτελέσματα

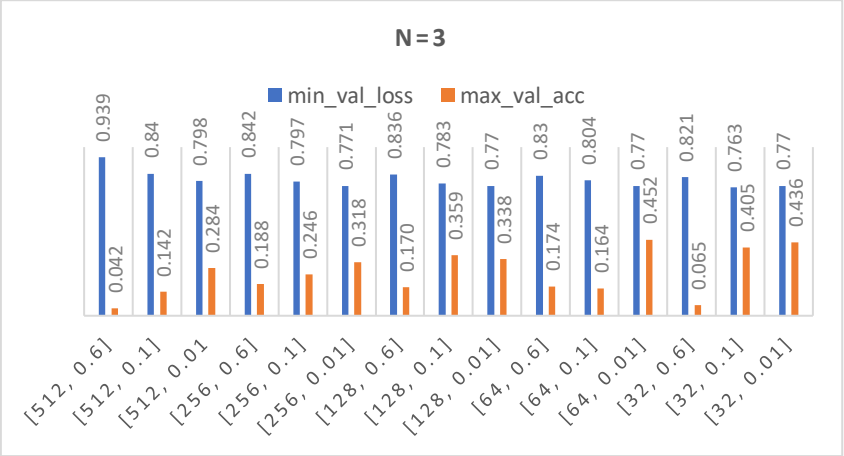
Πρώτο σύνολο πειραμάτων

Στη συνέχεια, από το πρώτο σύνολο πειραμάτων εξαντλητικής αναζήτησης, για κάθε μοντέλο που εκπαιδεύτηκε για $N = 1, 2, 3$, $h = 8$, $d\_inner\_hid = 4 \cdot d\_model$, $d\_k = d\_v = d\_model/4$, $label\_smoothing = False$, $batch\_size = 10$ παρουσιάζονται η μικρότερη τιμή στη συνάρτηση απωλειών (minimum validation loss) στο σύνολο δεδομένων επικύρωσης (validation set) και η υψηλότερη τιμή της ακρίβειας PSSP (maximum validation accuracy) στο σύνολο δεδομένων επικύρωσης κατά τη διάρκεια της εκπαίδευσης. Αυτός είναι ένας τρόπος να δούμε πώς πήγε σε γενικές γραμμές η εκπαίδευση για διάφορους συνδυασμούς υπερπαραμέτρων. Τα αποτελέσματα για $N = 4, 5, 6$ δεν παρατίθενται μιας και τα αντίστοιχα δίκτυα δεν εκπαιδεύτηκαν ουσιαστικά και παρουσιάσαν overfitting σε πολύ λίγες εποχές.

Στα παρακάτω τρία γραφήματα, στον οριζόντιο άξονα δίνονται οι συνδυασμού των υπερπαραμέτρων $[d\_model, dropout]$, όπου $d\_model = 32,64,128,256,512$, $dropout = 0.01,0.1,0.6$.







<u>Παρατηρήσεις</u>:

- **Για $N = 1$:** Παρατηρούμε αρκετά υψηλές τιμές ακρίβειας και χαμηλές τιμές απώλειας στο σύνολο δεδομένων επικύρωσης για όλους τους συνδυασμούς παραμέτρων. Οι τιμές αυτές βελτιώνονται σε γενικές γραμμές καθώς τα $dropout$ και d_model μειώνονται, με πολύ μικρές διακυμάνσεις ιδιαίτερα στους συνδυασμούς με $dropout =$

0.01,0.1. Η υψηλότερη ακρίβεια σημειώνεται για $d\_model = 64,128$ και $dropout = 0.01$.

- **Για $N = 2$:** Παρατηρούμε αρκετά υψηλές τιμές ακρίβειας για τους περισσότερους συνδυασμούς παραμέτρων, φανερώνοντας ότι οι δύο συνιστώσες κωδικοποιητών και αποκωδικοποιητών μπορούν να ανακαλύψουν αποτελεσματικότερα τα μοτίβα των ακολουθιών, για το δοθέν σύνολο δεδομένων και το λεξιλόγιο που χρησιμοποιείται. Οι χαμηλότερες τιμές ακρίβειας εμφανίζονται σε δίκτυα με $dropout = 0.6$ και οι υψηλότερες για $dropout = 0.1$, καθώς το d_model παίρνει μικρότερες τιμές, με καταλληλότερη την τιμή $d\_model = 64$.
- **Για $N = 3$:** Παρατηρούμε ότι όλα τα δίκτυα παρουσιάζουν αρκετά χαμηλή ακρίβεια στο σύνολο επικύρωσης κατά τη διάρκεια της εκπαίδευσης, για το σύνολο των συνδυασμών που επιλέχθηκαν. Καθώς το d_model μειώνεται από την τιμή 512, η ακρίβεια αυξάνεται, με την υψηλότερη να παρουσιάζεται για $d\_model = 32, 64$. Ομοίως η μείωση του $dropout$ αυξάνει την ακρίβεια, ιδιαίτερα καθώς το d_model επίσης μειώνεται, εμφανίζοντας υψηλότερες τιμές ακρίβειας για $dropout = 0.01$.
- **Για $N > 3$:** Τα δίκτυα δεν εκπαιδεύονται ομαλά και παρουσιάζουν overfitting σε λίγες εποχές εκπαίδευσης, οδηγώντας σε τιμές ακρίβειας αρκετά χαμηλές (χαμηλότερες από αυτές που παρουσιάστηκαν για $N = 3$, οι οποίες μειώνονται δραματικά καθώς το N αυξάνεται). Για $N = 4$ εμφανίζονται λίγο χειρότερα αποτελέσματα από αυτά που παρουσιάσαμε για $N = 3$.

Τα μοντέλα, από το πρώτο σύνολο πειραμάτων εξαντλητικής αναζήτησης, που εμφάνισαν την υψηλότερη ακρίβεια PSSP και τις μικρότερες τιμές στη συνάρτηση απωλειών (loss) στο σύνολο δεδομένων επικύρωσης (validation set) κατά τη διάρκεια της εκπαίδευσης, στη συνέχεια χρησιμοποιήθηκαν για την μετάφραση του συνόλου ελέγχου και την παραγωγή των τελικών προβλέψεων. Τα αποτελέσματα με greedy decoding ($beam\_size = 1$), $batch\_size = 10$ και $patience = 10$ για το Early Stopping παρουσιάζονται στον παρακάτω πίνακα:

Πίνακας 6.1: Αποτελέσματα στο CB513 για τα πιο επιτυχή μοντέλα του πρώτου συνόλου πειραμάτων με $batch\_size = 10$, $patience = 10$.

N	d_model	d_k,d_v	d_inner	h	dropout	LabelS.	acc(%), beam_s=1
1	32	8	128	8	0.1	False	59.19
2	128	32	512	8	0.1	False	**61.14**
1	256	64	1,024	8	0.01	False	55.8

1	64	16	256	8	0.01	False	60.38
1	64	16	256	8	0.1	False	59.89
1	128	32	512	8	0.01	False	59.35
1	128	32	512	8	0.1	False	58.49
1	512	128	2,048	8	0.01	False	56.76
1	32	8	128	16	0.01	False	57.05
1	32	8	128	16	0.1	False	58.85
1	64	16	256	16	0.01	False	57.89
1	64	16	256	16	0.1	False	**60.38**
1	64	16	256	16	0.6	False	56.49
1	128	32	512	16	0.01	False	60.27
1	128	32	512	16	0.1	False	58.82
1	256	64	1,024	16	0.01	False	56.18
1	256	64	1,024	16	0.1	False	58.72
1	32	8	128	8	0.1	True	58.78
2	128	32	512	8	0.1	True	58.62
1	512	128	2,048	8	0.01	True	59.2
1	256	64	1,024	8	0.1	True	58.65
1	256	64	1,024	8	0.01	True	55.99
1	128	32	512	8	0.01	True	57.39
1	128	32	512	8	0.1	True	58.57
1	128	32	512	8	0.6	True	60.62
1	64	16	256	8	0.1	True	**60.85**

<u>Παρατηρήσεις</u>:

Παρατηρούμε ότι τα καλύτερα μοντέλα και των τριών κατηγοριών που εμφανίζονται στον παραπάνω πίνακα (με $h = 8$ και $label\_smoothing = False$, με $h = 16$ και $label\_smoothing = False$, με $h = 8$ και $label\_smoothing = True$), εμφανίζουν παρεμφερή ποσοστά ακρίβειας που κυμαίνονται μεταξύ των τιμών ~56% και ~61%. Υψηλότερη ακρίβεια ανάμεσα στα παραπάνω μοντέλα με greedy decoding δίνουν οι υπερπαράμετροι $N = 2$, $d\_model = 128$, $d\_k = d\_v = 32$, $d\_inner\_hid = 512$, $dropout = 0.1$, $label\_smoothing = False$.

Συγκρίνοντας την απόδοση των κοινών συνδυασμών υπερπαραμέτρων των παραπάνω κατηγοριών, μπορεί κάποιος να παρατηρήσει ότι:

- Για $h = 16$ αντί $h = 8$ υπάρχει σε ορισμένους συνδυασμούς μικρή αύξηση της ακρίβειας, μικρότερης του 0.1%.
- Με την επιλογή $label\_smoothing = True$ αντί $label\_smoothing = False$ και $h = 8$ οδηγεί στους μισούς περίπου κοινούς συνδυασμούς σε μια μικρή αύξηση της ακρίβειας, μικρότερης του 0.1%, με εξαίρεση έναν συνδυασμό που οδηγεί σε σημαντική αύξηση της τάξης του 3%.

Επίσης από τα μοντέλα του προηγούμενου πίνακα, επιλέχθηκαν τα πρώτα 8, τα οποία εκπαιδεύτηκαν με $batch\_size = 18$ και στη συνέχεια αποκωδικοποιήθηκαν (με $beam\_size = 1,5,10$) ώστε να συγκριθούν τα αποτελέσματα με διαφορετικό $batch\_size$ για τους συνδυασμούς παραμέτρων που έχουν δώσει καλά αποτελέσματα στα παραπάνω πειράματα ($patience = 10$ για το Early Stopping):

Πίνακας 6.2: Αποτελέσματα στο CB513 για τα πιο επιτυχή μοντέλα του πρώτου συνόλου πειραμάτων με $batch\_size = 18$, $patience = 10$. Το σύμβολο ↑ δείχνει αύξηση σε σχέση με τα ποσοστά των αντίστοιχων μοντέλων στον Πίνακα 6.1, διαφοροποιώντας μόνο το $batch\_size$.

N	d_model	d_k,d_v	d_inner	h	dropout	LabelS.	acc(%), beam_s=1	acc(%), beam_s=5	acc(%), beam_s=10
1	32	8	128	8	0.1	False	60.17 ↑	61.11	60.92
2	128	32	512	8	0.1	False	44.55	49.93	49.56
1	256	64	1,024	8	0.01	False	59.4 ↑	60.53	60.41
1	64	16	256	8	0.01	False	59.19	60.80	60.76
1	64	16	256	8	0.01	False	61.27 ↑	**62.49**	62.4
1	128	32	512	8	0.01	False	58.22	59.35	58.93
1	128	32	512	8	0.1	False	61.03 ↑	62.12	62.0
1	512	128	2,048	8	0.01	False	58.24↑	60.1	59.74

<u>Παρατηρήσεις</u>:

Παρατηρούμε ότι στην πλειοψηφία των διαφορετικών συνδυασμών παραμέτρων (στους 5 από τους 8) με σταθερό $h = 8$, η αύξηση του $batch\_size$ από 10 σε 18 οδήγησε σε αύξηση της ακρίβειας με greedy decoding, από 1% έως 4%. Συνεπώς, η αύξηση του $batch\_size$ είναι μια επιλογή (ιδιαίτερα στις περιπτώσεις των μικρών δικτύων που η μνήμη το επιτρέπει) που αξίζει να διερευνηθεί για την αύξηση της ακρίβειας πρόβλεψης του μοντέλου. Επιπλέον, παρατηρούμε ότι η εφαρμογή beam search κατά την αποκωδικοποίηση με μεγαλύτερο $beam\_size = 5$ οδήγησε σε όλα τα παραπάνω μοντέλα σε αύξηση της ακρίβειας της τάξης του

1.5% κατά μέσο όρο. Αντίθετα περαιτέρω αύξηση της παραμέτρου για $beam\_size = 10$, δεν βελτίωσε την ακρίβεια, αλλά αντιθέτως τα ποσοστά ακρίβειας εμφανίστηκαν ελαφρώς μικρότερα (από 0.1% έως 0.4%).

Δεύτερο σύνολο πειραμάτων

Από το δεύτερο σύνολο πειραμάτων εξαντλητικής αναζήτησης προέκυψαν τα εξής αποτελέσματα με $batch\_size = 20$, $patience = 20$ για το Early Stopping:

Πίνακας 6.3: Αποτελέσματα στο CB513 για όλα τα μοντέλα του δεύτερου συνόλου πειραμάτων με $N = 2$, $batch\_size = 20$, $patience = 20$.

N	d_model	d_k,d_v	d_inner	h	dropout	LabelS.	acc(%), beam_s=5	acc(%), beam_s=10
2	512	256	2,048	8	0.1	False	33.63	29.77
2	512	256	2,048	8	0.01	False	29.36	28.24
2	256	128	1,024	8	0.1	False	60.43	60.51
2	256	128	1,024	8	0.01	False	**64.22**	**64.36**
2	128	64	512	8	0.1	False	63.30	63.22
2	128	64	512	8	0.01	False	62.34	62.55
2	64	32	256	8	0.1	False	62.85	62.59
2	64	32	256	8	0.01	False	58.96	58.45
2	32	16	128	8	0.1	False	63.7	63.5

<u>Παρατηρήσεις</u>:

Παρατηρούμε ότι η επιλογή της συσχέτισης $d\_k = d\_v = d\_model/2$ δίνει για $N = 2$, $h = 8$ εξαιρετική βελτίωση στην απόδοση των μοντέλων, με τα περισσότερα να εμφανίζουν ακρίβεια μεγαλύτερη του 60%, που δεν είχε εμφανιστεί σε πολλά από τα προηγούμενα πειράματα. Δοκιμάζονται $dropout = 0.01{,}0.1$, ενώ παραλείπεται η επιλογή του label smoothing ($label\_smoothing = False$) μιας και από τα προηγούμενα πειράματα φάνηκε να μην δίνει ουσιαστική βελτίωση στην απόδοση για τα μικρά δίκτυα που δοκιμάζουμε. Επίσης χρησιμοποιήθηκε σταθερό $batch\_size = 20$, μιας και από προηγούμενα πειράματα φάνηκε να βοηθά στην εκπαίδευση μικρών μοντέλων. Η υψηλότερη απόδοση εμφανίστηκε για $N = 2$, $d\_model = 256$, $d\_k = d\_v = 128$, $d\_inner\_hid = 1{,}024$, $dropout = 0.01$, $label\_smoothing = False$, $h = 8$ με $beam\_size = 10$, ενώ ακολουθούν οι συνδυασμοί $d\_model = 32$, $d\_k = d\_v = 16$, $d\_inner\_hid = 128$, $dropout = 0.1$ και $d\_model = 128$,

$d\_k = d\_v = 64$, $d\_inner\_hid = 512$, $dropout = 0.1$. Εξαίρεση αποτελούν τα αποτελέσματα για $d\_model = 512$ που ήταν σημαντικά χαμηλά. Κι εδώ, όπως παραπάνω, παρατηρούμε ότι η αύξηση του $beam\_size$ από 5 σε 10, δεν δίνει ουσιαστική βελτίωση στην ακρίβεια, αλλά αυξάνει σημαντικά το χρόνο ολοκλήρωσης της αποκωδικοποίησης.

Τα ίδια πειράματα του Πίνακα 6.2 πραγματοποιήθηκαν επίσης με $N = 1$, μιας και τα δίκτυα με μία συνιστώσα κωδικοποιητή και μία συνιστώσα αποκωδικοποιητή έχουν δώσει σταθερά υψηλά αποτελέσματα στην πλειοψηφία των πειραμάτων, οπότε εδώ θέλουμε να δούμε αν θα ξεπεράσουν την ακρίβεια των πειραμάτων με $N = 2$ και $d\_inner\_hid = d\_model/2$. Παρατίθενται τα αποτελέσματα μόνο με greedy decoding και με beam search με $beam\_size = 5$, μιας και παρατηρήσαμε ότι η αποκωδικοποίηση με μεγαλύτερο $beam\_size$ δεν δίνει ουσιαστική βελτίωση και ταυτόχρονα είναι αρκετά χρονοβόρα διαδικασία.

Πίνακας 6.4: Αποτελέσματα στο CB513 για όλα τα μοντέλα του δεύτερου συνόλου πειραμάτων με $N = 1$, $batch\_size = 20$, $patience = 20$.

N	d_model	d_k,d_v	d_inner	h	dropout	LabelS.	acc(%), beam_s=1	acc(%), beam_s=5
1	512	256	2,048	8	0.1	False	58.39	60.12
1	512	256	2,048	8	0.01	False	58.36	59.39
1	256	128	1,024	8	0.1	False	58.36	58.87
1	256	128	1,024	8	0.01	False	57.07	59.21
1	128	64	512	8	0.1	False	56.81	58.86
1	128	64	512	8	0.01	False	56.75	58.39
1	64	32	256	8	0.1	False	58.96	60.6
1	64	32	256	8	0.01	False	59.35	60.55
1	32	16	128	8	0.1	False	**62.03**	**63.04**
1	32	16	128	8	0.01	False	59.54	62.72

<u>Παρατηρήσεις</u>:

Παρατηρούμε ότι και στην περίπτωση του $N = 1$, η επιλογή της συσχέτισης $d\_k = d\_v = d\_model/2$ με $h = 8$ δίνει παρόμοια και ελαφρώς καλύτερα αποτελέσματα σε κάποιους συνδυασμούς σε σχέση με αυτά που παρουσιάστηκαν στο πρώτο σύνολο πειραμάτων. Συγκριτικά με τα αποτελέσματα του Πίνακα 6.2 για $N = 2$, τα αποτελέσματα ακρίβειας για $N = 1$ είναι λίγο μικρότερα (της τάξης του 1%) στους περισσότερους συνδυασμούς. Εξαίρεση αποτελούν οι συνδυασμοί $d\_model = 512$, $d\_k = d\_v = 256$, $d\_inner\_hid = 2{,}048$,

$dropout = 0.01$ και $dropout = 0.1$, που παρουσιάζουν πολύ καλύτερα αποτελέσματα για $N = 1$ (κατά 26.5% και 30% αντίστοιχα) από ότι για $N = 2$.

6.4 Σύνοψη συμπερασμάτων των πειραμάτων

Από το σύνολο των πειραμάτων που παρουσιάστηκαν και επεξηγήθηκαν στην ενότητα 6.3, θα μπορούσαμε να συμπεράνουμε τα παρακάτω σχετικά με την αποδοτικότητα των διαφόρων συνδυασμών υπερπαραμέτρων του μοντέλου μας:

- Αρχικά, η εκπαίδευση είναι αποτελεσματικότερη για μικρό αριθμό συνιστωσών κωδικοποιητή και αποκωδικοποιητή ($N = 1,2$) και οδηγεί σε ποσοστά PSSP ακρίβειας ~60% για τους περισσότερους συνδυασμούς που δοκιμάσαμε για τις υπόλοιπες υπερπαραμέτρους. Μεγαλύτερα δίκτυα ($N \geq 3$) εμφάνισαν αδυναμία εκπαίδευσης και υπερπροσαρμογή στα δεδομένα εκπαίδευσης οδηγώντας σε πολύ χαμηλά ποσοστά ακρίβεια στα σύνολα επικύρωσης και ελέγχου.
- Στα περισσότερα δίκτυα που εκπαιδεύτηκαν στα προηγούμενα πειράματα με $N = 1,2$, η εκπαίδευση ολοκληρωνόταν γύρω στις 80-100 εποχές με την τεχνική Early Stopping, με συνολική διάρκεια εκπαίδευσης ~40-60 λεπτά (30sec ή λιγότερο για κάθε εποχή). Η διαδικασία του decoding παρουσίαζε τους εξής χρόνους εκτέλεσης: το greedy decoding ($beam\_size = 1$) ολοκληρωνόταν σε περίπου 10 λεπτά, το beam search με $beam\_size = 5$ σε περίπου 40 λεπτά και με $beam\_size = 10$ σε περίπου μια ώρα. Οι συνδυασμοί υπερπαραμέτρων που παρουσίασαν ποσοστό της τάξης του 60%, φάνηκε ότι κατάφερναν να προβλέπουν το μήκος των ακολουθιών εξόδου (του συνόλου ελέγχου) στην πλειοψηφία τους. Συγκεκριμένα, από τις 515 ακολουθίες ελέγχου αστοχία σε μήκος εμφανίζονταν σε μόλις 5-10 για τα πιο αποδοτικά μοντέλα, ενώ κάποια μοντέλα που πέτυχαν ακρίβεια πάνω από 62% με beam search δεν εμφάνισαν λάθη στα μήκη των ακολουθιών που προέβλεπαν.
- Ο ορισμός των $d\_k = d\_v = d\_model/2$ έδωσε σημαντική αύξηση στην ακρίβεια των μοντέλων που εκπαιδεύτηκαν με $N = 1,2$, φτάνοντάς τα στο 62%-64% με beam search. Καλά αποτελέσματα παρουσίασε και η επιλογή των $d\_k = d\_v = d\_model/4$ αλλά χαμηλότερης ακρίβειας.
- Η επιλογή του $h = 8$ (πλήθος πολλαπλών κεφαλών προσοχής) είναι σημαντική για την αποδοτικότητα του μοντέλου, μιας και επιτρέπει την καλύτερη εκμάθηση πιο σύνθετων εξαρτήσεων μεταξύ των ακολουθιών. Τα πειράματα με μεγαλύτερο $h = 16$ δεν έδωσαν ουσιαστική βελτίωση στην ακρίβεια των μοντέλων που δοκιμάστηκαν, ενώ ταυτόχρονα αυξάνουν το χώρο μνήμης που καταλαμβάνουν οι υπολογισμοί.

Πειραματισμοί με μικρότερο $h = 4$ που δοκιμάστηκαν στην αρχική τυχαία αναζήτηση, έδωσαν χαμηλότερα αποτελέσματα για μικρά μοντέλα, γεγονός που δείχνει ότι χρειάζεται μεγάλος αριθμός κεφαλών προσοχής για την εκμάθηση των εξαρτήσεων στα μεγάλα μήκη ακολουθιών εισόδου.

- Η αύξηση του $batch\_size$ από 8 σε 20, βελτίωσε την ακρίβεια των μοντέλων στους περισσότερους συνδυασμούς υπερπαραμέτρων. Επιπλέον αύξηση της παραμέτρου δεν ήταν εφικτό να δοκιμαστεί λόγω περιορισμών μνήμης. Ταυτόχρονα, στην καλύτερη εκπαίδευση των μικρών δικτύων που δοκιμάσαμε ($N = 1,2$) βοηθά η χρήση μικρού $dropout$ (0.01 ή 0.1 ανάλογα την περίπτωση).
- Η εφαρμογή Label Smoothing ($label\_smoothing = True$) δεν οδήγησε σε ουσιαστική βελτίωση της αποδοτικότητας των μοντέλων, γεγονός που ενδεχομένως να οφείλεται στο μικρό πλήθος κλάσεων και λεξικού που ορίζεται από το πρόβλημα με τη χρήση 1 χαρακτήρα ως λέξη.

Η υψηλότερη ακρίβεια που σημειώσαμε στα παραπάνω παραδείγματα προκύπτει από τις υπερπαραμέρους: $N = 2$, $d\_model = 256$, $d\_k = d\_v = 128$, $d\_inner\_hid = 1,024$, $dropout = 0.01$, $label\_smoothing = False$, $h = 8$, $batch\_size = 20$ με αποκωδικοποίηση με $beam\_size = 10$.

Κεφάλαιο 7

Επίλογος

7.1 Σύνοψη και συμπεράσματα

Συμπερασματικά, το μοντέλο του Μεταφραστή (Transformer) παρουσιάζει μια καλή ακρίβεια απόδοσης, της τάξης του 64.4%, (αν σκεφτεί κανείς πόσο βαθιά μοντέλα έχουν δοκιμαστεί και το φράγμα του ~71% που δεν έχουν ξεπεράσει) στο ιδιαίτερα δύσκολο πρόβλημα της Πρόβλεψης της Δευτεροταγούς Δομής των Πρωτεϊνών, το οποίο έγκειται:

1. Στο μικρό πλήθος δειγμάτων του συνόλου εκπαίδευσης για το σύνθετο μοντέλο του Transformer, που οδηγεί σε πολύ γρήγορη υπερπροσαρμογή των δεδομένων σε μεγάλα δίκτυα (με 3-6 συνιστώσες κωδικοποιητή και αποκωδικοποιητή).
2. Το μεγάλο μήκος των ακολουθιών εισόδου (~700) και το μικρό μέγεθος του λεξιλογίου εισόδου και εξόδου (~22 και ~8 αντίστοιχα).
3. Τους περιορισμούς μνήμης που εισάγει η χρήση πολλών πολλών κεφαλών προσοχής σε συνδυασμό με μεγάλες τιμές στις παραμέτρους d_model, d_k, d_v που μας περιορίζει στη χρήση μικρού $batch\_size$.
4. Το πολύ μεγάλο πλήθος υπερπαραμέτρων προς ρύθμιση, που καθιστά αδύνατη την μελέτη όλων των δυνατών συνδυασμών τους, αφήνοντας ανοιχτό το ενδεχόμενο να υπάρχει κάποιο μοντέλο που πετυχαίνει ακρίβεια μεγαλύτερη από τις τιμές που δόθηκαν στο Κεφάλαιο 6.

Από την άλλη, τα πιο επιτυχή μοντέλα του Μεταφραστή, που παρουσιάστηκαν παραπάνω, έχουν γρήγορη σύγκλιση σε ακρίβεια μεγαλύτερη από 60% (σε περίπου 80 εποχές) και έχουν σχετικά μικρό μέγεθος ($N = 1,2$ συνιστώσες κωδικοποιητή - αποκωδικοποιητή) και μικρούς χρόνους εκπαίδευσης (~40 λεπτά), συνεπώς η μελέτη και η επέκτασή τους παρουσιάζει μεγάλο ενδιαφέρον.

7.2 Μελλοντικές επεκτάσεις

Πειραματισμός με Διαφορετικά Σύνολα 'Λέξεων' (εξαγωγή n-grams από ακολουθίες εισόδου, εξόδου)

Μια άμεση επέκταση του μοντέλου που παρουσιάστηκε προηγουμένως, μπορεί να επιτευχθεί με τον πειραματισμό με διαφορετικά λεξιλόγια για τις ακολουθίες εισόδου (ακολουθίες πρωτεϊνών) και τις ακολουθίες εξόδου (ακολουθίες δευτεροταγούς δομής). Στο υλοποιημένο μοντέλο εξάγονται λέξεις μήκους ενός χαρακτήρα (δημιουργώντας λεξιλόγιο με 22 λέξεις για την είσοδο και 8 για την έξοδο). Μπορούμε να πειραματιστούμε με την εξαγωγή λέξεων δύο χαρακτήρων ή περισσότερων (n-grams) με επικάλυψη ή χωρίς για τον εμπλουτισμό του λεξιλογίου. Ο εμπλουτισμός αυτός ενδεχομένως να επιτρέψει την αποδοτικότερη εκπαίδευση βαθύτερων δικτύων ή την καλύτερη περιγραφή του χώρου χαρακτηριστικών των ακολουθιών, οδηγώντας σε υψηλότερη ακρίβεια των πειραμάτων μας.

Εξαγωγή αναπαραστάσεων (representations from pre-training) από μεγαλύτερα σύνολα δεδομένων και Μεταφορά της μάθησης (Transfer Learning) στο κύριο μοντέλο

Στη συγκεκριμένη εργασία, οι ενσωματώσεις των λέξεων του λεξιλογίου που δημιουργούμε αρχικοποιούνται τυχαία. Μια ιδέα εμπνευσμένη από τις σύγχρονες τεχνικές επεξεργασίας φυσικής γλώσσας είναι, λοιπόν, να εξάγουμε με μη επιβλεπόμενη μάθηση τις αναπαραστάσεις των λέξεων μας, πριν τις χρησιμοποιήσουμε για την εκπαίδευση του κύριου μοντέλου. Χαρακτηριστικές τεχνικές στον τομέα της επεξεργασίας φυσικής γλώσσας για την εξαγωγή αναπαραστάσεων, βάσει του νοηματικού πλαισίου, είναι οι BERT (Bidirectional Encoder Representations from Transformers), που παράγει αναπαραστάσεις με χρήση κωδικοποιητή Transformer διπλής κατεύθυνσης καθώς και τα LSTMs διπλής κατεύθυνσης που χρησιμοποιούνται για κωδικοποίηση.

Τέτοιες ιδέες μοντέλων θα μπορούσαν να εκπαιδευτούν σε ένα μεγάλο σύνολο ακολουθιών πρωτεϊνών (που είναι ευκολότερο να βρεθεί από ότι ένα σύνολο Δευτεροταγούς Δομής με ακολουθίες εξόδου, μιας και η εκπαίδευση είναι μη επιβλεπόμενη) και στη συνέχεια οι αναπαραστάσεις των ακολουθιών να χρησιμοποιηθούν για την εκπαίδευση του δικτύου. Ειδικότερα, αν διατηρήσουμε ως κύριο μοντέλο για τη μηχανική μετάφραση τον Transformer, και χρησιμοποιήσουμε ως pretraining την ιδέα του BERT, θα μπορούσαμε να χρησιμοποιήσουμε το αρχικό training για να αρχικοποιήσουμε τα βάρη του Encoder (transfer learning). Φυσικά, η δυσκολία αυτής της επέκτασης έγκειται στην εύρεση ενός αρκετά μεγάλου συνόλου δεδομένων πρωτεϊνικών ακολουθιών για την εξαγωγή των αναπαραστάσεων. Γενικά,

υπάρχουν αρκετές τεχνικές εκμάθησης αναπαραστάσεων που θα μπορούσαν να διερευνηθούν ανάλογα με τα διαθέσιμα δεδομένα για την μεταφορά γνώσης στο κύριο νευρωνικό δίκτυο που υλοποιούμε.

Βιβλιογραφία

A. Busia, N. J. (2017). *Next-Step Conditioned Deep Convolutional Neural Networks Improve Protein Secondary Structure Prediction.* Ανάκτηση από CoRR, abs/1702.03865.

Aäron Van Den Oord, S. D. (2016). Wavenet: A generative model for raw audio. *SSW*, (σ. 125).

Anfinsen CB, H. E. (1961). The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proc Natl Acad Sci U S A, 47*(9). doi:10.1073/pnas.47.9.1309

Ankur Parikh, O. T. (2016). A decomposable attention model. *In Empirical Methods in Natural Language Processing.*

Asai K, H. S. (1993). Prediction of protein secondary structure by the hidden markov model. *Bioinformatics, 9*(2), σ. 141. doi:10.1093/bioinformatics/9.2.141

Ashish Vaswani, N. S. (2017). Attention is all you need. *Advances in neural information processing systems,* (σσ. 5998-6008). Ανάκτηση από https://arxiv.org/abs/1706.03762

Aydin Z, A. Y. (2006). Protein secondary structure prediction for a single-sequence using hidden semi-markov models. *BMC Bioinformatics, 7*(1), σ. 178. doi:10.1186/1471-2105-7-178

Baldi P, B. S. (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics, 15*(11), σσ. 937–46. doi:10.1093/bioinformatics/15.11.937

Barton, J. A. (1999). Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics, 34*(4), σσ. 508-519.

Bengio Y., F. P. (1993). The problem of learning long-termdependencies in recurrent networks. *IEEE International Conference on NeuralNetworks* (σσ. 1183–1195). San Francisco: IEEE Press.

Bengio Y., S. P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Tr. Neural Nets*.

Buchan DW, M. F. (2013). Scalable web services for the psipred protein analysis workbench. Nucleic Acids Res. *41*(Web Server issue), σσ. 349–57. doi:10.1093/nar/gkt381

Chen J, C. N. (2007). Cascaded bidirectional recurrent neural networks for protein secondary structure prediction. *IEEE/ACM Trans Comput Biol Bioinforma, 4*(4), σσ. 572–82. doi:10.1109/tcbb.2007.1055

Cho K., V. M. (2014). *On the properties of neural machine translation: Encoder-decoder approaches.* Ανάκτηση από ArXiv e-prints, abs/1409.1259.

Cholle, F. (2017). *Deep learning with Python.* Manning Publications.

Christian Szegedy, V. V. (2015). *Rethinking the inception architecture for computer vision.* Ανάκτηση από CoRR, abs/1512.00567.

Chrupala G., K. A. (2015). *Learning language through pictures.* Ανάκτηση από arXiv 1506.03694.

Chu W, G. Z. (2004). A graphical model for protein secondary structure prediction. *Proceedings 21st Annual International Conference on Machine Learning(ICML).* New York: ACM.

Chung J., G. Ç. (2015). Gated feedback recurrent neural networks. *ICML' 15.*

D. E. Worrall, S. J. (2017). Harmonic Networks: Deep Translation and Rotation Equivariance. *Proc. CVPR'17*, (σσ. 5028–5037).

Denny Britz, A. G.-T. (2017). *Massive exploration of neural machine translation architectures.* Ανάκτηση από CoRR, abs/1703.03906.

Diederik Kingma, J. B. (2015). Adam: A method for stochastic optimization. *ICLR.*

DT, J. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol, 292*(2), σ. 195. doi:10.1006/jmbi.1999.3091

Dzmitry Bahdanau, K. C. (2014). *Neural machine translation by jointly learning to align and translate.* Ανάκτηση από CoRR abs/1409.0473.

Faraggi E, A. E. (2012). Spine x: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *J Comput Chem, 33*(3), σσ. 259–67. doi:10.1002/jcc.21968

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics, 36*(4), σσ. 193-202.

G. Huang, Z. L. (2017). Densely Connected Convolutional Networks. *Proc. CVPR'17*, (σσ. 2261–2269).

Ginsburg, O. K. (2017). *Factorization tricks for LSTM networks.* Ανάκτηση από arXiv preprint arXiv:1703.10722.

Graves. (2012). Supervised Sequence Labelling with Recurrent Neural Networks. Στο *Studies in Computational Intelligence.* Springer.

Graves A., L. M. (2009). A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(5), σσ. 855–868. doi:10.1109/tpami.2008.137

Graves A., M. A. (2013). Speech recognition with deep recurrentneural networks. *ICASSP' 2013*, (σσ. 6645–6649).

Guo J, C. H. (2004). A novel method for protein secondary structure prediction using dual-layer svm and profiles. *Protein Struct Funct Bioinform, 54*(4), σσ. 738–43. doi:10.1002/prot.10634

Hochreiter S., B. Y. (2001). Gradient flow in recurrent nets: thedifficulty of learning long-term dependencies. *Field Guide to Dynamical Recurrent Networks, IEEE Press.*

Hochreiter S., S. J. (1997). Long short-term memory. *Neural Computation, 9*(8), σσ. 1935-1780.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. *Diploma Thesis, T.U. München*.

Hua S, S. Z. (2001). A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *J Mol Biol, 308*(2), σσ. 397–407. doi:10.1006/jmbi.2001.4580

Ian Goodfellow, Y. B. (2016). *Deep Learning.* MIT Press.

Iddo Drori, I. D.-F. (2018). *High Quality Prediction of Protein Q8 Secondary Structure by Diverse Neural Network Architectures.* Ανάκτηση από arXiv preprint arXiv:1811.07143.

Ilya Sutskever, O. V. (2014). Sequence to sequence learning with neural networks. *In Advances in Neural Information Processing Systems*, (σσ. 3104–3112).

Jimmy Lei Ba, J. R. (2016). *Layer normalization.* Ανάκτηση από arXiv preprint arXiv:1607.06450.

Jonas Gehring, M. A. (2017). *Convolutional sequence to sequence learning.* Ανάκτηση από arXiv preprint arXiv:1705.03122v2.

Jozefowicz R., Z. W. (2015). An empirical evaluation of recurrentnetwork architectures. *ICML' 2015.*

Junyoung Chung, Ç. G. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling.* Ανάκτηση από CoRR abs/1412.3555.

Kaiming He, X. Z. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (σσ. 770–778).

Koide S., K. K. (2018). Neural Edit Operations for Biological Sequences. *Advances in Neural Information Processing Systems 31, (NeurIPS), 49654975*. Ανάκτηση από http://papers.nips.cc/paper/7744-neural-edit-operations-for-biological-sequences.pdf

Kyunghyun Cho, B. v. (2014). *Learning phrase representations using rnn encoder-decoder for statistical machine translation.* Ανάκτηση από CoRR abs/1406.1078.

Li Xiangang, W. X. (2014, 10 15). *Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition.* Ανάκτηση από arXiv:1410.4281.

M. Cuturi, M. B. (2017). Soft-DTW: a Differentiable Loss Function for Time-Series. *Proc. ICML'17*, (σσ. 894–903).

Miljanovic, M. (2012). Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction. *Indian Journal of Computer and Engineering, 3*(1).

Minh-Thang Luong, H. P. (2015). *Effective approaches to attention based neural machine translation.* Ανάκτηση από arXiv preprint arXiv:1508.04025.

Mirabello C, P. G. (2013). paleale 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics, 29*(16), σσ. 2056–8. doi:10.1093/bioinformatics/btt344

Mitchell, T. M. (1997). *Machine Learning.* New York: McGraw-Hill.

Murray, e. a. (χ.χ.).

Nitish Srivastava, G. E. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(1), 1929–1958.

Noam Shazeer, A. M. (2017). *Outrageously large neural networks: The sparsely-gated mixture-of-experts layer.* Ανάκτηση από arXiv preprint arXiv:1701.06538.

Olaf Ronneberger, P. F. (2015). U-Net: Convolutional networks for biomedical image segmentation. *In International Conference on Medical Image Computing and Computer-Assisted Intervention* (σσ. 234–241). Springer.

Qian N, S. T. (1988). Predicting the secondary structure of globular proteins using neural network models. *J Mol Biol, 202*(4), σσ. 865-84. doi:10.1016/0022-2836(88)90564-5

Rafal Jozefowicz, O. V. (2016). *Exploring the limits of language modeling.* Ανάκτηση από arXiv preprint arXiv:1602.02410.

S. Needleman, C. W. (1970, 3). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology, 48*(3), 443-453.

Sak Hasim, S. A. (2014). *Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling.* Ανάκτηση από https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf

Sander, B. R. (1993). Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology, 232*(2), 584-599.

Sander, B. R. (1994). Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins: Structure, Function, and Bioinformatics, 19*(1), σσ. 55–72.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*(61), σσ. 85–117. doi:10.1016/j.neunet.2014.09.003

Schmidhuber, S. H. (1997). Long short-term memory. *Neural computation, 9*(8), σσ. 1735–1780.

Schmidler SC, L. J. (2000). Bayesian segmentation of protein secondary structure. *J Comput Biol A J Comput Mol Cell Biol, 7*(1-2), σσ. 48-233. doi:10.1089/10665270050081496

Sergey Ioffe, C. S. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning.*

Stuart J. Russell, P. N. (2010). *Artificial Intelligence: A Modern Approach* (Third Edition εκδ.). Prentice Hall.

Torrisi M, K. M. (2018). *state-of-the-art ab initio prediction of protein secondary structure in 3 and 8 classes.* Ανάκτηση από https://www.biorxiv.org/content/early/2018/03/30/289033.

Troyanskaya, J. Z. (2014). CB6133 dataset. Ανάκτηση από https://www.princeton.edu/~jzthree/datasets/ICML2014/dataset_readme.txt

Troyanskaya, J. Z. (2014). Deep supervised and convolutional generative stochastic network for protein secondary structure prediction., (σσ. 745–753).

Wolf, O. P. (2016). *Using the output embedding to improve language models.* Ανάκτηση από arXiv preprint arXiv:1608.05859.

Wolfgang Kabsch, C. S. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules, 22*(12), σσ. 637-2577. doi:10.1002/bip.360221211

Yonghui Wu, M. S. (2016). *Google's neural machine translation system: Bridging the gap between human and machine translation.* Ανάκτηση από arXiv preprint arXiv:1609.08144.

Yoon Kim, C. D. (2017). Structured attention networks. *In International Conference on Learning Representations,*.

Yuedong Yang, J. G. (2018). Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in Bioinformatics*, *19*(3), σσ. 482–494.

Zell, A. (1994). *Simulation Neuronaler Netze [Simulation of Neural Networks] (in German)* (First Edition εκδ.). Addison-Wesley.

Zhen Li, Y. Y. (2016). Protein secondary structure prediction using cascaded convolutional and recurrent neural networks. *International Joint Conference on Artificial Intelligence*, (σσ. 2560–2567).