Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Βελτιστοποίηση Νευρωνικών Δικτύων με χρήση Κβαντοποίησης

Διπλωματικη Εργασια

της

ΜΥΡΣΙΝΗΣ ΝΤΟΚΟΥ

Επιβλέπων: Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εργαστηριο Τεχνητης Νοημοσυνης και Συστηματων Μαθησης

Αθήνα, Οκτώβριος 2024

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Τεχνητής Νοημοσύνης και Συστημάτων Μάθησης

Βελτιστοποίηση Νευρωνικών Δικτύων με χρήση Κβαντοποίησης

Διπλωματικη Εργασια

της

ΜΥΡΣΙΝΗΣ ΝΤΟΚΟΥ

Επιβλέπων: Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 7η Νοεμβρίου 2024.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Ανδρέας-Γεώργιος Σταφυλοπάτης	Γεώργιος Στάμου	Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2024

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Τεχνητής Νοημοσύνης και Συστημάτων Μάθησης

(Υπογραφή)

.....

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

Περίληψη

Η κβαντοποίηση αποτελεί ένα ευρέως διαδεδομένο πεδίο έρευνας για την βελτιστοποίηση νευρωνικών δικτύων, με στόχο τη μείωση των υπολογιστικών απαιτήσεων και του αποτυπώματος μνήμης των μοντέλων βαθιάς μάθησης. Αυτή η διπλωματική εργασία εξετάζει την αποτελεσματικότητα της κβαντοποίησης μετά την εκπαίδευση (Post-Training Quantization - PTQ) ως μέθοδο βελτιστοποίησης σε αρχιτεκτονικές συνελικτικών νευρωνικών δικτύων (CNN) και μεγάλων γλωσσικών μοντέλων. Με τη χρήση της κβαντοποίησης, τα βάρη και οι ενεργοποιήσεις των μοντέλων μετατρέπονται σε αναπαραστάσεις χαμηλότερης ακρίβειας, μειώνοντας σημαντικά τον χρόνο εκτέλεσης και το αποτύπωμα σε μνήμη, χωρίς σημαντικές απώλειες ακρίβειας. Με την εξέταση διαφόρων μεθόδων κβαντοποίησης, αναδεικνύεται ο τρόπος με τον οποίο τα CNN και τα γλωσσικά μοντέλα ανταποκρίνονται στις μειώσεις ακρίβειας αναπαράστασης. Με πειράματα πάνω σε δημοφιλή σύνολα δεδομένων αξιολογούνται οι διαφορές στα μεγέθη των μοντέλων, την ταχύτητα εξαγωγής αποτελεσμάτων και την ακρίβεια. Τα ευρήματα αποκαλύπτουν ότι με την κβαντοποίηση μπορεί να επιτευχθεί σημαντική εξοικονόμηση πόρων διατηρώντας παράλληλα υψηλή ακρίβεια, ενισχύοντας έτσι την δυνατότητα ανάπτυξης προηγμένων μοντέλων σε συσκευές με περιορισμένους πόρους.

Λέξεις Κλειδιά

Κβαντοποίηση, Συνελικτικά Νευρωνικά Δίκτυα, Γλωσσικά Μοντέλα, Κβαντοποίηση Μετά την Εκπαίδευση, Βελτιστοποίηση Μοντέλων.

Abstract

Quantization is a widespread area of research in neural network optimization, aiming to reduce the computational requirements and memory footprint of deep learning models. This thesis examines the effectiveness of Post-Training Quantization (PTQ) as an optimization method in Convolutional Neural Network (CNN) architectures and large language models. Through the application of quantization, model weights and activations are converted to lower precision representations, significantly reducing execution time and memory footprint while maintaining a high level of accuracy. Examining various quantization methods highlights how CNN and language models respond to representation accuracy reductions in different ways. Experiments on popular datasets evaluate differences in model sizes, inference time, and accuracy. The findings reveal that significant resource savings can be achieved with quantization while maintaining high accuracy, thereby enhancing the ability to develop advanced models on resource-constrained devices.

Keywords

Quantization, Convolutional Neural Networks (CNNs), Language Models, Post-Training Quantization (PTQ), Model Optimization.

Ευχαριστίες

Θα ήθελα καταρχάς να ευχαριστήσω τον επιβλέποντα καθηγητή κ.Ανδρέα Σταφυλοπάτη, για την επίβλεψη αυτής της εργασίας και την ευκαιρία να εμβαθύνω στο συγκεκριμένο αντικείμενο. Επίσης, ευχαριστώ ιδιαίτερα τον κ. Γεώργιο Σιόλα για τις πολύτιμες συμβουλές που μου παρείχε, την καθοδήγηση του και την άψογη συνεργασία που είχαμε. Τέλος, θα ήθελα να ευχαριστήσω θερμά τους γονείς μου, και τους φίλους μου για τη βοήθεια και τη στήριξη που μου προσέφεραν όλα αυτά τα χρόνια.

Περιεχόμενα

Περίληψη **7**

Abstract **9**

Ευχαριστίες **11**

Περιεχόμενα **14**

Κατάλογος Σχημάτων **16**

Κατάλογος Πινάκων **17**

1 Εισαγωγή **19**
1.1 Εισαγωγή στο Πρόβλημα 19
1.2 Αντικείμενο Διπλωματικής Εργασίας 20
1.3 Διάρθρωση Εργασίας 20

2 Θεωρητικό Υπόβαθρο **21**
2.1 Μηχανική Μάθηση 21
2.2 Νευρωνικά Δίκτυα 22
2.2.1 Αρχιτεκτονική Νευρωνικών Δικτύων 22
2.2.2 Νευρώνας και Λειτουργία του 22
2.2.3 Συναρτήσεις Ενεργοποίησης 23
2.2.4 Εκπαίδευση Νευρωνικών Δικτύων 24
2.2.5 Το προβλημα των vanishing gradients 25
2.2.6 Overfitting και Κανονικοποίηση 26
2.2.7 Αξιολογηση Αποδοσης Νευρωνικου 26
2.3 Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNN) . . 27
2.3.1 Επίπεδο Εισόδου 28
2.3.2 Συνελικτικό Επίπεδο (Convolutional layer) 28
2.3.3 Επίπεδα Υποδειγματοληψίας (Pooling Layers) 29
2.3.4 Πλήρως Συνδεδεμένα Επίπεδα (Fully Connected Layers) 30
2.4 Μεταφορά Μάθησης (Transfer Learning) 30
2.5 Αρχιτεκτονικές Συνελικτικών Νευρωνικών Δικτύων 31

2.5.1 ResNet . . . 31
2.5.2 MobileNet . . . 32
2.6 Μετασχηματιστές (Transformers) . . . 33
2.6.1 BERT . . . 35
2.7 Αναπαράσταση Αριθμητικών Τιμών . . . 36
2.7.1 Αναπαράσταση κινητης υποδιαστολής (Floating Point) . . . 36
2.7.2 Αναπαράσταση Ακεραίων (Integer) . . . 37
2.8 Μέθοδοι Βελτιστοποίησης Νευρωνικών Δικτύων . . . 37
2.8.1 Κλάδεμα (Pruning) . . . 38
2.8.2 Απόσταξη γνώσης (Distillation) . . . 38
2.8.3 Αποσύνθεση Τανυστών (Tensor Decomposition) . . . 38
2.8.4 Παραγοντοποίηση Χαμηλής Τάξης (Low-Rank Factorization) . . . 39
2.8.5 Κβαντοποίηση (Quantization) . . . 39

3 Κβαντοποίηση **41**
3.1 Εισαγωγή στην Κβαντοποίηση . . . 41
3.2 Συμμετρική Κβαντοποίηση . . . 42
3.3 Μη-Συμμετρική Κβαντοποίηση . . . 44
3.4 Κβαντοποίηση μετά την Εκπαίδευση (Post-Training Quantization - PTQ) . . . 45
3.4.1 Δυναμική Κβαντοποίηση (Dynamic Quantization) . . . 45
3.4.2 Στατική Κβαντοποίηση (Static Quantization) . . . 45
3.5 Κβαντοποίηση με Ευαισθητοποίηση (Quantization-Aware Training - QAT) . . . 46

4 Πειραματική Διαδικασία **47**
4.1 Περιγραφή - Εργαλεία . . . 47
4.2 Δεδομένα και Μοντέλα . . . 48
4.3 Προεπεξεργασία Δεδομένων . . . 49
4.4 Πειραματική Διαδικασία . . . 50
4.4.1 Μοντέλα CNN . . . 50
4.4.2 BERT . . . 51
4.5 Αποτελέσματα . . . 51

5 Σύνοψη **61**
5.1 Συμπερασματα . . . 61
5.2 Μελλοντικές Επεκτάσεις . . . 62

Βιβλιογραφία **65**

Γλωσσάριο **69**

Κατάλογος Σχημάτων

2.1 Δομή Single-layer Perceptron και Multi-layer Perceptron 23
2.2 Παραδείγματα συναρτήσεων ενεργοποίησης 24
2.3 Δομή Τεχνητού Νευρώνα - Περσεπτρον 25
2.4 Παράδειγμα $3x3$ πίνακα σύγχυσης 27
2.5 Δομή Συνελικτικού Νευρωνικού Δικτύου (CNN) 27
2.6 Συνελικτικό Επίπεδο Νευρωνικού Δικτύου (CNN) 29
2.7 Εφαρμογή Δειγματοληψίας σε (Image Map) 29
2.8 Εφαρμογή Transfer Learning 31
2.9 Δομικό στοιχείο ResNet 32
2.10 Αρχιτεκτονική MobileNetv2 32
2.11 Αρχιτεκτονική μετασχηματιστή BERT 34
2.12 Διαδικασία προ-εκπαίδευσης και fine-tuning του BERT πάνω στο σύνολο δεδομένων SQuAD (Stanford Question Answering Dataset) 36

3.1 Top-1 accuracy σε σχέση με τον αριθμό operations που απαιτούνται για ένα single forward pass 42
3.2 Σχεδιάγραμμα Συμμετρικής Κβαντοποιησης 43
3.3 Διαδικασία κβαντοποίησης και αποκβαντοποίησης 43
3.4 Σχεδιάγραμμα Μη-Συμμετρικής Κβαντοποιησης 44
3.5 Σύγκριση διαδικασίας Κβαντοποίησης με Ευαισθητοποίηση (QAT) (αριστερά) και Κβαντοποίησης μετά την Εκπαίδευση (PTQ) (δεξιά) 46

4.1 Δείγμα εικόνων του CIFAR-100 πριν και μετά την επεξεργασία 50
4.2 Σύγκριση στην απόδοση το μέγεθος μοντέλου και τον χρόνο inference ανά μέθοδο κβαντοποίησης για το ResNet18 52
4.3 Σύγκριση στην απόδοση το μέγεθος μοντέλου και τον χρόνο inference ανά μέθοδο κβαντοποίησης για το ResNet34 54
4.4 Σύγκριση στην απόδοση το μέγεθος μοντέλου και τον χρόνο inference ανά μέθοδο κβαντοποίησης για το ResNet50 55
4.5 Σύγκριση στην απόδοση το μέγεθος μοντέλου και τον χρόνο inference ανά μέθοδο κβαντοποίησης για το MobileNetV2 56

4.6 Σύγκριση στην απόδοση το μέγεθος μοντέλου και τον χρόνο inference ανά μέθοδο κβαντοποίησης για τα μοντέλα CNN 58

Κατάλογος Πινάκων

4.1 Περιγραφή χρήσης Μοντέλων 49
4.2 Αποτελέσματα ResNet18 52
4.3 Αποτελέσματα ResNet34 53
4.4 Αποτελέσματα ResNet50 54
4.5 Αποτελέσματα MobileNetV2 56
4.6 Αποτελέσματα BERT 58

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή στο Πρόβλημα

Καθώς οι τεχνολογίες μηχανικής μάθησης και βαθιάς μάθησης αναπτύσσονται ραγδαία, οι υπολογιστικές ανάγκες που απαιτούνται για την εκπαίδευση και εκτέλεση νευρωνικών δικτύων γίνονται ολοένα και πιο απαιτητικές. Τα νευρωνικά δίκτυα χρησιμοποιούνται σε ένα πλήθος εφαρμογών, όπως αναγνώριση εικόνων, επεξεργασία φυσικής γλώσσας, πρόβλεψης χρονοσειρών και ανάλυσης μεγάλων δεδομένων. Ωστόσο, ένα βασικό μειονέκτημα των δικτύων αυτών είναι η υψηλή κατανάλωση μνήμης και υπολογιστικής ισχύος, ιδίως όταν πρόκειται για μεγάλα μοντέλα. Αυτό το πρόβλημα είναι πιο εμφανές στα βαθιά νευρωνικά δίκτυα (Deep Neural Networks - DNNs) τα οποία περιέχουν εκατομμύρια παραμέτρους, γεγονός που οδηγεί σε μεγάλο αποτύπωμα μνήμης και υψηλή υπολογιστική πολυπλοκότητα. Ο λόγος που τα νευρωνικά δίκτυα έχουν τόσο υψηλές απαιτήσεις σε μνήμη και υπολογιστική ισχύ είναι η δομή τους. Κάθε νευρώνας συνδέεται με άλλους νευρώνες μέσω βαρών, τα οποία απαιτούν αποθήκευση και εκτέλεση υπολογισμών σε κάθε βήμα. Στην περίπτωση των βαθιών νευρωνικών δικτύων, ο αριθμός των βαρών αυξάνεται εκθετικά με την προσθήκη νέων επιπέδων, με αποτέλεσμα να απαιτείται πολύ περισσότερη μνήμη και υπολογιστική ισχύς. Για παράδειγμα, συνελικτικά μοντέλα όπως το ResNet-50 μπορεί να έχουν πάνω από 25 εκατομμύρια παραμέτρους, ενώ το BERT, ένα μοντέλο φυσικής επεξεργασίας γλώσσας (Natural Language Processing - NLP), μπορεί να έχει εκατοντάδες εκατομμύρια παραμέτρους. Η μεγάλη κατανάλωση χώρου και μνήμης στα νευρωνικά δίκτυα καθιστά δύσκολη την εφαρμογή τους σε πραγματικό χρόνο, κυρίως σε περιβάλλοντα με περιορισμένους πόρους, όπως οι κινητές συσκευές και τα ενσωματωμένα συστήματα. Επομένως, η ανάγκη για αποδοτική χρήση των πόρων καθίσταται κρίσιμη για να μπορούν τα περίπλοκα μοντέλα να αναπτυχθούν και να χρησιμοποιηθούν σε αυτές τις πλατφόρμες. Η υπολογιστική πολυπλοκότητα των νευρωνικών δικτύων δεν επηρεάζει μόνο τις επιδόσεις των συστημάτων αλλά έχει και σημαντικές περιβαλλοντικές και οικονομικές συνέπειες. Η αυξημένη κατανάλωση ενέργειας που απαιτείται για την εκπαίδευση αυτών των μοντέλων είναι σημαντική, ειδικά σε περιβάλλοντα cloud, όπου οι ενεργειακές απαιτήσεις είναι ήδη υψηλές. Η ανάγκη για εξειδικευμένο hardware, όπως οι μονάδες GPU (Graphics Processing Units) και TPU (Tensor Processing Units), αυξάνει σημαντικά το κόστος των εφαρ-

μογών μηχανικής μάθησης. Έτσι, τα τελεύταία χρόνια προέκυψε η αναζήτηση και η ανάπτυξη τεχνικών βελτιστοποίησης, όπως η κβαντοποίηση, καθώς η ανάγκη εύρεσης αποδοτικότερων λύσεων καθίσταται επιτακτική.

1.2 Αντικείμενο Διπλωματικής Εργασίας

Το αντικειμενο της παρουσας διπλωματικης εργασιας αφορα την τεχνική της κβαντοποίησης για να μειώσουμε το αποτύπωμα του δικτύου σε χώρο και μνήμη. Η κβαντοποίηση προσφέρει σημαντικές βελτιώσεις στην απόδοση των μοντέλων σε μνήμη και υπολογιστικούς πόρους, χωρίς να επηρεάζεται σε μεγάλο βαθμό η ακρίβεια των προβλέψεων. Η διαδικασία υλοποίησης περιλαμβάνει σε πρώτο στάδιο την εκπαίδευση των μοντέλων πάνω στα σύνολα δεδομένων. Έπειτα τρέχουμε inference και υπολογίζουμε την ακρίβεια, τον χρόνο inference και μέγεθος του κάθε μοντέλου. Εφαρμόζουμε κβαντοποίηση (στατική και δυναμική) στα μοντέλα και ξανατρέχουμε ινφερενςε για τα ίδια αποτελέσματα ώστε να συγκρίνουμε την αποδοτικότητα της κβαντοποίησης.

1.3 Διάρθρωση Εργασίας

Η παρούσα Διπλωματική Εργασία διαρθρώνεται σε 5 κεφάλαια. Στο Κεφάλαιο 1, πραγματοποιηθηκε μια εισαγωγη στο θεμα, αλλα και τον λογο που ειναι απαραιτητη η ερευνα στο αντικειμενο. Στο Κεφάλαιο 2, καλυπτεται το θεωρητικο υποβαθρο που απαιτειται ωστε να ειναι δυνατη η κατανοηση εννοιων γυρω απο τα συνελικτικά νευρωνικά δίκτυα και μοντέλα επεξεργασιας φυσικης γλωσσας όπως οι μετασχηματιστές, καθως και την μοντελοποιηση των εφαρμογων τους απο υπολογιστικα συστηματα. Στο Κεφάλαιο 3, επεξηγουνται βασικες εννοιες αναφορικα με την μέθοδο της κβαντοποίησης, ενώ στο Κεφάλαιο 4, ακολουθει η αναλυτικη περιγραφη της πειραματικης διαδικασιας που πραγματοποιηθηκε, καθως και η αξιολογηση των αποτελεσματων που προεκυψαν απο αυτη. Το Κεφάλαιο 5, ολοκληρωνει την παρουσα εργασια με τα τελικα συμπερασματα που προεκυψαν και τις μελλοντικες κατευθυνσεις στις οποιες θα μπορουσε να επεκταθει η συγκεκριμενη μελετη.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Μηχανική Μάθηση

Η Μηχανική Μάθηση, ένας υποκλάδος της Τεχνητής Νοημοσύνης και της επιστήμης των υπολογιστών γενικότερα, αφορά τη σχεδίαση και ανάπτυξη αλγορίθμων που επεξεργάζονται εμπειρικά δεδομένα, αναζητώντας πρότυπα μέσω στατιστικών μεθόδων, με στόχο την πρόβλεψη των μηχανισμών που παρήγαγαν αυτά τα δεδομένα. Εναλλακτικά, μπορεί να οριστεί ως το επιστημονικό πεδίο που επιτρέπει στους υπολογιστές να αποκτούν ικανότητες μάθησης χωρίς να είναι άμεσα προγραμματισμένοι για αυτόν τον σκοπό (Samuel, 1959). Σύμφωνα με έναν πιο κλασικό ορισμό της Μηχανικής Μάθησης, ένα πρόγραμμα υπολογιστών μαθαίνει από την εμπειρία E σχετικά με μια διεργασία T και μια μέτρηση απόδοσης P, όταν η απόδοση του σε εργασίες T, όπως μετριέται μέσω της P, βελτιώνεται όσο αποκτά την εμπειρία E (Mitchell, 1997). Γενικά, η Μηχανική Μάθηση περιλαμβάνει τρεις κύριους τρόπους μάθησης, οι οποίοι ανταναικλούν τρόπους με τους οποίους μαθαίνει ο άνθρωπος: επιβλεπόμενη μάθηση, μη επιβλεπόμενη μάθηση και ενισχυτική μάθηση.

- **Επιβλεπόμενη μάθηση (Supervised Learning)**: Κατά τη διαδικασία της επιβλεπόμενης μάθησης, ένας αλγόριθμος δημιουργεί μια συνάρτηση που συνδέει συγκεκριμένες εισόδους (εκπαιδευτικό σύνολο) με επιθυμητές εξόδους, με σκοπό τη γενίκευση της συνάρτησης και για άλλες εισόδους με άγνωστες εξόδους. Χρησιμοποιείται σε προβλήματα όπως η ταξινόμηση (classification), η πρόβλεψη (prediction), η διερμηνεία (interpretation) και η παλινδρόμηση (regression).

- **Μη επιβλεπόμενη μάθηση (Unsupervised Learning)**: Σε αυτή τη μέθοδο, ο αλγόριθμος δημιουργεί ένα μοντέλο για ένα σύνολο εισόδων υπό μορφή παρατηρήσεων, χωρίς να γνωρίζει εκ των προτέρων τις επιθυμητές εξόδους. Χρησιμοποιείται για ανάλυση συσχετισμών (association analysis) και ομαδοποίηση (clustering).

- **Ενισχυτική μάθηση (Reinforcement Learning)**: Εδώ, ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσω άμεσης αλληλεπίδρασης με το περιβάλλον. Χρησιμοποιείται κυρίως σε προβλήματα σχεδιασμού (planning), όπως η κίνηση ρομπότ και η βελτιστοποίηση εργασιών σε βιομηχανικούς χώρους.

2.2 Νευρωνικά Δίκτυα

Τα Νευρωνικά Δίκτυα (Neural Networks) αποτελούν τον πυρήνα της βαθιάς μάθησης (deep learning) και είναι εμπνευσμένα από τη δομή και λειτουργία του ανθρώπινου εγκεφάλου. Αποτελούνται από διασυνδεδεμένους νευρώνες (perceptrons), οργανωμένους σε επίπεδα (layers), όπου κάθε νευρώνας επεξεργάζεται πληροφορίες και τις μεταδίδει σε άλλους νευρώνες στο δίκτυο. Μέσα από αυτή τη διαδικασία, τα Νευρωνικά Δίκτυα μπορούν να μάθουν πολύπλοκες σχέσεις από τα δεδομένα και να επιλύσουν διάφορα προβλήματα, όπως ταξινόμηση, παλινδρόμηση, και πρόβλεψη.

2.2.1 Αρχιτεκτονική Νευρωνικών Δικτύων

Για την επιλυση πιο συνθετων προβληματων, οι επιμερους νευρωνες (single layer perceptrons) συνδυαζονται και δημιουργουν πολυεπιπεδες αρχιτεκτονικες νευρωνων. Οι δομες αυτες καλουνται πολυεπίπεδα δίκτυα (multilayer perceptrons) (βλ. Σχημα 2.1), και τα επιπεδα στα οποια οργανωνονται οι νευρωνες χωριζονται στις εξης κατηγοριες:

- **Επίπεδο Εισόδου (Input Layer)**: Το επίπεδο που δέχεται τα δεδομένα εισόδου, τα οποία μπορεί να είναι εικόνες, κείμενο, ή αριθμητικά δεδομένα.
- **Κρυφά Επίπεδα (Hidden Layers)**: Ένα ή περισσότερα επίπεδα νευρώνων, όπου πραγματοποιείται η επεξεργασία των δεδομένων μέσω ενεργοποιητικών συναρτήσεων.
- **Επίπεδο Εξόδου (Output Layer)**:Το επίπεδο που παράγει την τελική πρόβλεψη του δικτύου.

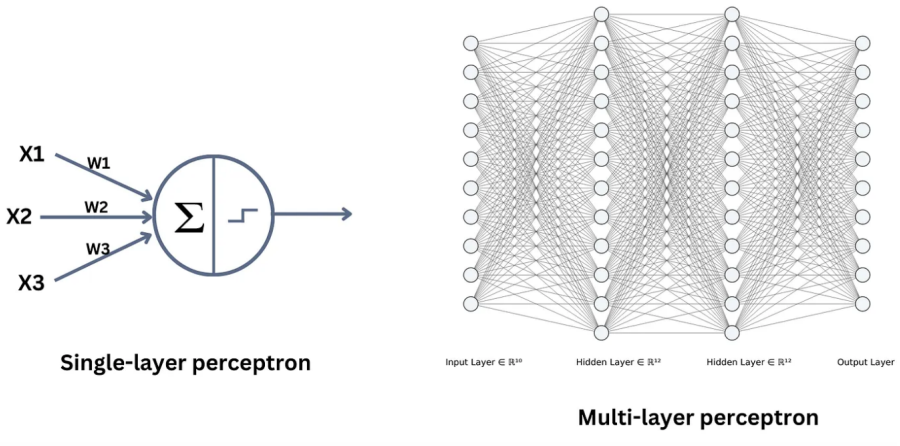
Τα πολυεπιπεδα δικτυα μπορουν να ειναι πληρως συνδεδεμενα (fully connected), δηλαδή καθε νευρωνας ενος επιπεδου να συνδεεται με ολους τους νευρωνες του επομενου επιπεδου, η αντίθετα μερικως συνδεδεμενα (partially connected). Επισης, τα δικτυα μπορουν να χωριστουν σε δικτυα προσθιας τροφοδοτησης (feedforward), στην περιπτωση που η πληροφορια μεταβιβαζεται σειριακα από το επιπεδο εισοδου στο επιπεδο εξοδου, η σε αναδρομικα δικτυα (recurrent) που περιλαμβανουν βροχους αναδρασης.

2.2.2 Νευρώνας και Λειτουργία του

Ο βασικός δομικός λίθος ενός Νευρωνικού Δικτύου είναι ο νευρώνας. Κάθε νευρώνας δέχεται ένα σύνολο εισόδων $x_1, x_2, ..., x_n$, οι οποίες πολλαπλασιάζονται με αντίστοιχα βάρη $w_1, w_2, ..., w_n$ και στη συνέχεια υπολογίζεται το συνολικό άθροισμα των σταθμισμένων εισόδων:

$$z = w_1x_1 + w_2x_2 + ... + w_nx_n + \beta = \sum_{i=1}^{n} w_ix_i + b \tag{2.1}$$

όπου β είναι η παράμετρος πόλωσης (bias), η οποία βοηθά στη μετατόπιση της ενεργοποιητικής συνάρτησης. Μετά τον υπολογισμό του z, εφαρμόζεται μια συνάρτηση ενεργοποίησης



Σχήμα 2.1: Δομή Single-layer Perceptron και Multi-layer Perceptron

$f(z)$, η οποία προσθέτει μη-γραμμικότητα στο μοντέλο. Το αποτέλεσμα του νευρώνα είναι:

$$y = f(z) \tag{2.2}$$

2.2.3 Συναρτήσεις Ενεργοποίησης

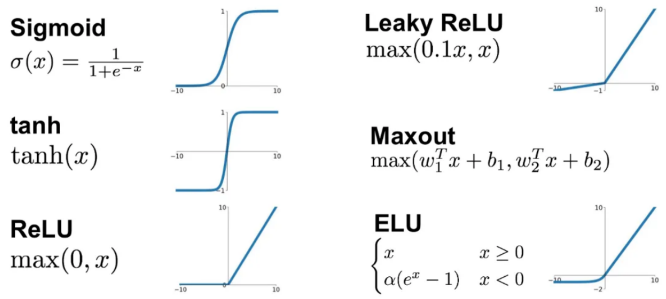
Ένα από τα βασικά στοιχεία ενός νευρώνα αποτελεί η Συναρτηση Ενεργοποιησης (Activation Function), όπως φαίνεται και στο σχημα 2.2. Μεσω αυτων των συναρτήσεων καθορίζεται η τιμή της εξόδου που θα διαβιβαστεί στον επόμενο νευρώνα με βάση την είσοδο. Χρησιμοποιώντας μη γραμμικές συναρτήσεις, επιτρέπουμε στους νευρώνες να μάθουν πολύπλοκες σχέσεις και απεικονίσεις μεταξύ εισόδου - εξόδου. Πιο ειδικά, το άθροισμα των σημάτων εισόδου σε έναν κόμβο, σταθμισμένων από τα αντίστοιχα βάρη, αντιστοιχίζεται και ερμηνεύεται πλέον σε ένα επιθυμητό διάστημα. Το διάστημα αυτό ορίζεται από την εκάστοτε συνάρτηση ενεργοποίησης.

Ορισμένες από τις πιο δημοφιλείς συναρτήσεις ενεργοποίησης είναι οι εξής:

- **Sigmoid**: Η σιγμοειδής συνάρτηση, που ονομάζεται και logistic, συμπιέζει την έξοδο του νευρώνα στο διάστημα (0,1). Είναι χρήσιμη για προβλήματα δυαδικής ταξινόμησης, αλλά συχνά υποφέρει από το πρόβλημα της εξαφάνισης κλίσης (vanishing gradients).

- **Tanh (Hyperbolic Tangent)**: Η tanh συμπιέζει τις τιμές στο διάστημα (-1,1), προσφέροντας καλύτερα αποτελέσματα από τη sigmoid, αλλά εξακολουθεί να έχει το ίδιο πρόβλημα εξαφάνισης κλίσης σε βαθιά δίκτυα.

- **ReLU (Rectified Linear Unit)**: Η συνάρτηση ReLU είναι η πιο δημοφιλής συνάρτηση ενεργοποίησης, καθώς επιλύει το πρόβλημα της εξαφάνισης κλίσης και είναι υπολογιστικά αποδοτική. Ωστόσο, μπορεί να προκαλέσει το πρόβλημα των ¨νεκρών' νευρώνων (dead neurons), όπου ορισμένοι νευρώνες παύουν να μαθαίνουν αν η τιμή z είναι πάντοτε αρνητική.



Σχήμα 2.2: Παραδείγματα συναρτήσεων ενεργοποίησης

2.2.4 Εκπαίδευση Νευρωνικών Δικτύων

Η εκπαίδευση ενός Νευρωνικού Δικτύου γίνεται μέσω της διαδικασίας forward propagation και backpropagation.

Forward Propagation: Στο forward propagation, τα δεδομένα εισόδου περνούν μέσα από τα επίπεδα του δικτύου και οι νευρώνες υπολογίζουν τα αποτελέσματά τους με βάση τις ενεργοποιητικές συναρτήσεις. Το τελικό αποτέλεσμα στο επίπεδο εξόδου είναι η πρόβλεψη του δικτύου.

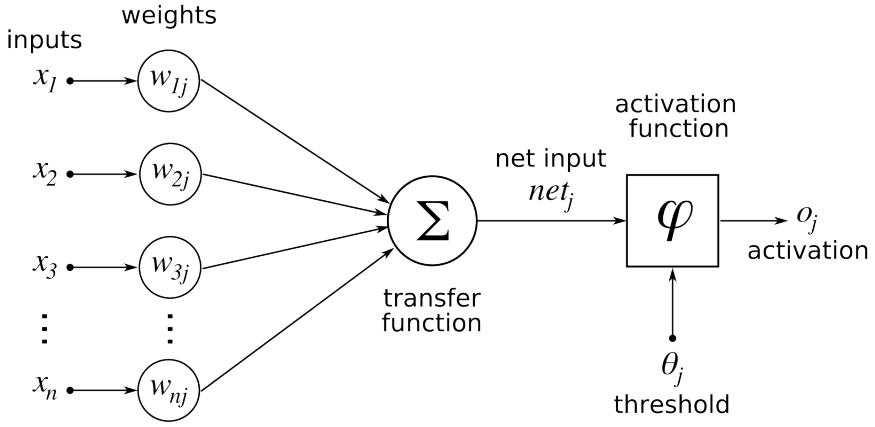
Συναρτήσεις Κόστους (Loss Functions:) Σε επόμενο βήμα, υπολογίζεται η συνάρτηση κόστους (loss function), η οποία μετρά τη διαφορά μεταξύ της πρόβλεψης του δικτύου (prediction) και της πραγματικής τιμής της ετικέτας στα δεδομένα εκπαίδευσης (label). Για προβλήματα δυαδικής ταξινόμησης, μια κοινή συνάρτηση κόστους είναι η Cross Entropy Loss:

$$L = -(ylog(p) + (1-y)log(1-p)) \tag{2.3}$$

Έστω ότι ένα δείγμα που ανήκει στην κατηγορία α, αποδίδεται από το νευρωνικό ότι ανήκει στην κατηγορία α, με πιθανότητα p. Στην περίπτωση αυτή, η πρόβλεψη είναι σωστή ($y = 1$) και ο τύπος γίνεται $-log(p)$. Το επιθυμητό είναι να λάβουμε L = 0, δηλαδή οι σωστές προβλέψεις να δίνονται με μεγάλη πιθανότητα. Αντίθετα, έστω ότι ένα δείγμα που ανήκει στην κατηγορία α, αποδίδεται από το νευρωνικό ότι ανήκει στην κατηγορία β, με πιθανότητα p. Στην περίπτωση αυτή, η πρόβλεψη είναι λάθος (y = 0) και ο τύπος μετασχηματίζεται σε log(1p). Ξανά το επιθυμητό είναι να λάβουμε L = 0, δηλαδή οι λάθος προβλέψεις να δίνονται με μικρή πιθανότητα. Τα παραπάνω μπορούν να γενικευτούν σε περιπτώσεις όπου διαθέτουμε περισσότερες από δύο κατηγορίες (έστω M) για ταξινόμηση.

Αλγόριθμος Backpropagation Backpropagation ονομάζεται η διαδικασία κατά την οποία το σφάλμα της συνάρτησης κόστους διαδίδεται πίσω μέσω του δικτύου για να ενημερωθούν τα βάρη. Η τιμη του βαρους w_{ij} ανανεωνεται συμφωνα με τη σχεση

$$w_{ij} = w_{ij} - \frac{\partial L}{\partial w_{ij}} \tag{2.4}$$



Σχήμα 2.3: Δομή Τεχνητού Νευρώνα - Περσεπτρον

Με χρήση του κανόνα της αλυσίδας για τη μερική παράγωγο, υπολογίζουμε την παράγωγο της συνάρτησης κόστους ως προς κάθε βάρος w_{ij} προκειμένου να ενημερωθούν κατάλληλα:

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial o_j}\frac{\partial o_j}{\partial w_{ij}} = \frac{\partial L}{\partial o_j}\frac{\partial o_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}} \tag{2.5}$$

Εάν ο νευρώνας βρίσκεται στο πρώτο στρώμα μετά το στρώμα εισόδου, τότε το o_i είναι απλά το x_i Το αθροισμα net_j οταν παραγωγιστει μερικως ως προς w_{ij} θα δωσει:

$$\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial}{w_{ij}}(\sum_{k=1}^{n} w_{kj}o_k) = \frac{\partial}{\partial w_{ij}} w_{ij}o_i = o_i \tag{2.6}$$

Η εξοδος o_i προκυπτει απο την ενεργοποιηση του αθροισματος net_j με βαση καποια συναρτηση φ. $(o_i = \varphi(\mathrm{net}_j))$:

$$\frac{\partial o_i}{\partial net_j} = \frac{\partial f(net_j)}{\partial net_j} \tag{2.7}$$

Στη συνέχεια, τα βάρη ενημερώνονται μέσω του αλγόριθμου βελτιστοποίησης της στοχαστικής βαθμίδωσης (Stochastic Gradient Descent - SGD):

$$w_i(t) = w_i(t-1) - \eta\frac{\partial L}{\partial w_i} \tag{2.8}$$

Η νεα τιμη του βαρους διαφερει απο την προηγουμενη κατα ενα παραγοντα

$$\eta\frac{\partial L}{\partial w_i} \tag{2.9}$$

όπου η είναι ο ρυθμος εκμαθησης (learning rate).

2.2.5 Το προβλημα των vanishing gradients

Κατα την διαρκεια του backpropagation, η τιμη του καθε βαρους προκυπτει αφαιρωντας απο την τρεχουσα τιμη την μερικη παραγωγο της συναρτησης κοστους αναφορικα με το συγκεκριμενο βαρος. Η τελευταια τιμη υπολογιζεται με τον κανονα της αλυσιδας. Εξαιτιας των

διαδοχικων πολλαπλασιασμων , ειναι πιθανο η τιμη που θα προκυψει να ειναι αρκετα κοντα στο μηδεν. Αυτο πρακτικα σημαινει, οτι η νεα τιμη βαρους θα ειναι ιση με την προηγουμενη. Γενικευοντας αυτο το προβλημα και για τα υπολοιπα βαρη του δικτυου, το νευρωνικο δικτυο πλεον δεν εκπαιδευεται.

2.2.6 Overfitting και Κανονικοποίηση

Κατά την εκπαίδευση των Νευρωνικών Δικτύων, ένα συχνό φαινόμενο είναι το overfitting, όπου το μοντέλο αποδίδει εξαιρετικά καλά στα δεδομένα εκπαίδευσης, αλλά αποτυγχάνει να γενικεύσει σε νέα δεδομένα. Για την αποφυγή του overfitting, χρησιμοποιούνται τεχνικές κανονικοποίησης, όπως:

1. **Dropout**: Κατά την εκπαίδευση, ορισμένοι νευρώνες απενεργοποιούνται τυχαία με πιθανότητα π, μειώνοντας την εξάρτηση του μοντέλου από συγκεκριμένους νευρώνες.

2. **Κανονικοποίηση L2 (Weight Decay)**: Ένας όρος προστίθεται στη συνάρτηση απώλειας που τιμωρεί τα μεγάλα βάρη, ενθαρρύνοντας το μοντέλο να μάθει μικρότερα και πιο σταθερά βάρη:
$$L_{total} = L + \lambda \sum_{i=1}^{n} w_i^2 \tag{2.10}$$
όπου λ είναι η παράμετρος κανονικοποίησης.

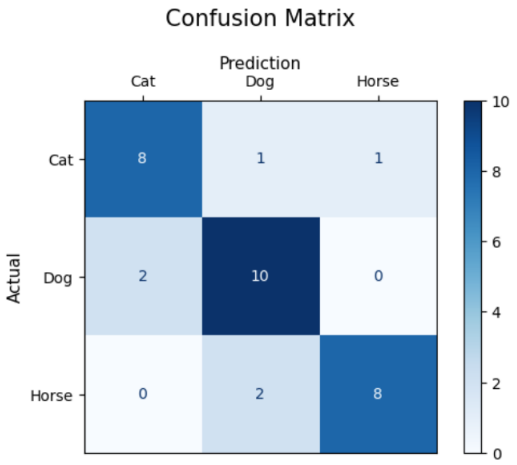
2.2.7 Αξιολογηση Αποδοσης Νευρωνικου

Προκειμένου να ποσοτικοποιηθεί το πόσο επιτυχης ήταν η εκπαίδευση ενός νευρωνικού μοντέλου, χρησιμοποιούνται κάποιες μετρικές αξιολόγησης στο σύνολο ελέγχου (test set).

Μετρικές Απόδοσης

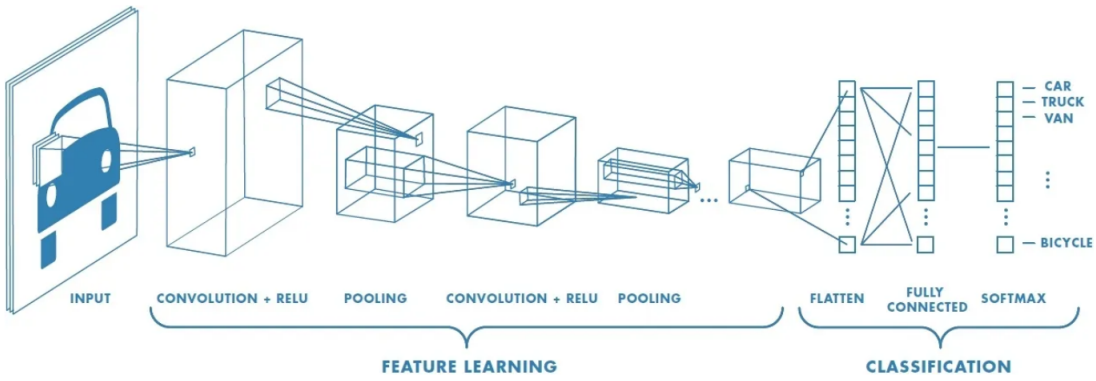
Ορθότητα - Accuracy : Πρόκειται για μια από τις βασικότερες μετρικές αξιολόγησης ενός μοντέλου. Ορίζεται ως το κλάσμα των ορθών προβλέψεων προς τον αριθμό των συνολικών εκτιμήσεων που πραγματοποιήθηκαν. Η μετρική αυτή, ωστόσο δεν είναι αντιπροσωπευτική όταν το dataset δεν είναι ισορροπημένο (balanced). Έστω ότι η κατηγορία Α ενός προβλήματος δυαδικής ταξινόμησης έχει 990 δείγματα, και η κατηγορία Β 10. Αν όλα τα δείγματα της κατηγορίας Β ταξινομηθούν λανθασμένα στην κατηγορία Α, η ακρίβεια του μοντέλου θα είναι πολύ υψηλή 990 / (990 + 10) = 99%. Για τον λόγο αυτό χρησιμοποιούνται στην πράξη και άλλες μετρικές απόδοσης.

Πίνακας Σύγχυσης : Πρόκειται για μια αναπαράσταση, συγκεκριμένα έναν τετραγωνικό πίνακα μεγέθους NxN, που δίνει πληροφορία για τα λάθη που συμβαίνουν μεταξύ των N κλάσεων. Πιο ειδικά κάθε τιμή αυτού του πίνακα αναφέρεται στον αριθμό των προβλέψεων που αποδόθηκαν σε μια κλάση (Predicted Class), δεδομένου ότι τα δείγματα άνηκαν σε μια άλλη κλάση (Actual Class). Στόχος είναι ο πίνακας σύγχυσης να έχει μηδενικά στα στοιχεία εκτός της κύριας διαγωνίου. Στο παρακάτω παράδειγμα (Σχήμα 2.4) διαθέτουμε 3 κλάσεις (3x3) πίνακας.



Σχήμα 2.4: Παράδειγμα $3x3$ πίνακα σύγχυσης

2.3 Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNN)



Σχήμα 2.5: Δομή Συνελικτικού Νευρωνικού Δικτύου (CNN)

Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks, CNNs) αποτελούν μία από τις πιο δημοφιλείς αρχιτεκτονικές στη μηχανική μάθηση, κυρίως για εφαρμογές που σχετίζονται με την ανάλυση εικόνων, αλλά και για άλλα είδη δεδομένων που μπορούν να αναπαρασταθούν σε μορφή πινάκων. Τα CNN αξιοποιούν την μαθηματική πράξη της συνέλιξης (convolution) για την αυτόματη εξαγωγή χαρακτηριστικών από τις εισόδους τους, διατηρώντας την τοπική πληροφορία μέσα στα δεδομένα και εκμεταλλευόμενα την δομή τους. Τα κρυφά επίπεδα, από τα οποία απαρτίζονται δεν είναι όλα πανομοιότυπα μεταξύ τους, αλλά υπάρχει μια δομημένη ακολουθία στρωμάτων, όπου το καθένα επιτελεί από μια διαφορετική λειτουργία, συνδράμοντας όλα στην επίτευξη μιας «καλής» πρόβλεψης. Να σημειωθεί ότι αυτή τους η αρχιτεκτονική, δεν είναι μόνο καλή στην εκμάθηση χαρακτηριστικών, αλλά και επεκτάσιμη σε τεράστια σύνολα δεδομένων, διατηρώντας υψηλή ακρίβεια. Η δομή ενός CNN συνήθως

ακολουθεί το μοτίβο ενός αριθμού συνελικτικών επιπέδων που ακολουθούνται από επίπεδα max-pooling και στο τέλος μερικά πλήρως συνδεδεμένα επίπεδα. Στις επόμενες υποενότητες περιγράφονται εν συντομία τα πιο συχνά εμφανιζόμενα επίπεδα επεξεργασίας στα CNN.

2.3.1 Επίπεδο Εισόδου

Όπως σε όλα τα Νευρωνικά Δίκτυα, αποτελεί το πρώτο στρώμα του δικτύου, μέσω του οποίου εισέρχεται η πληροφορία από το περιβάλλον, με σκοπό την περαιτέρω επεξεργασία της από τα στρώματα που ακολουθούν. Στα Συνελικτικά Δίκτυα, σαν είσοδος μπορεί να χρησιμοποιηθεί μία ή περισσότερες εικόνες, η διάσταση των οποίων καθορίζει και τη «διάσταση» του επιπέδου αυτού. Σε αυτό το σημείο, να εξηγήσουμε ότι μια εικόνα είναι, στην ουσία, ένας πίνακας τιμών, με κάθε «κουτάκι» του να αντιπροσωπεύει την τιμή ενός εικονοστοιχείου της. Για να οριστεί χρειάζονται τρεις παράμετροι, το μήκος, το πλάτος και ο αριθμός καναλιών της. Ο τελευταίος εξαρτάται από τον αριθμό των τιμών που απαιτούνται για να αναπαραστήσουν ένα εικονοστοιχείο της. Πιο αναλυτικά, εάν είναι ασπρόμαυρη κάθε pixel αντιστοιχίζεται με 1 τιμή, οπότε η εικόνα έχει 1 κανάλι, ενώ εάν είναι έγχρωμη κάθε pixel αντιστοιχίζεται συνήθως με 3 τιμές, οπότε έχει 3 κανάλια. Βάσει αυτών, το επίπεδο εισόδου ενός CNN, δέχεται δεδομένα διαστάσεων: (Πλήθος Εικόνων) x (Μήκος Εικόνας) x (Πλάτος Εικόνας) x (Αριθμός Καναλιών).

2.3.2 Συνελικτικό Επίπεδο (Convolutional layer)

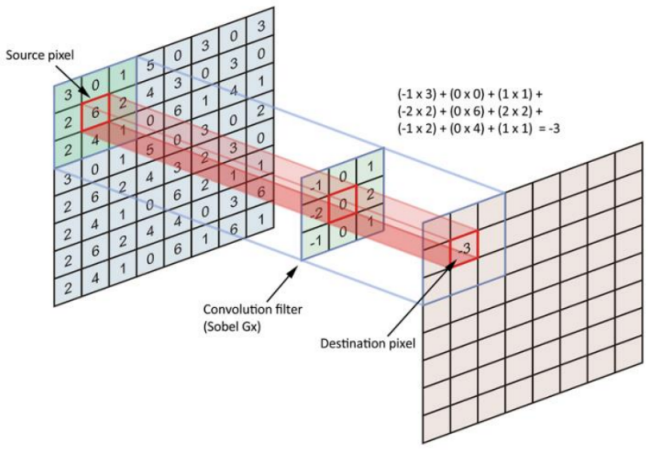
Το συνελικτικό επίπεδο αποτελεί το κύριο δομικό συστατικό ενός CNN, όπου λαμβάνει χώρα η διαδικασία εξαγωγής χαρακτηριστικών από τα δεδομένα εισόδου. Στα επίπεδα αυτά, μια μαθηματική πράξη συνέλιξης εφαρμόζεται μεταξύ των διανυσμάτων εισόδου (συνήθως μια εικόνα) και ενός φίλτρου (kernel), που συχνά ονομάζεται και φίλτρο (filter) για να παράγει έναν πίνακα χαρακτηριστικών (feature map). Σε μια δισδιάστατη εικόνα $I(x,y)$ η πράξη της συνέλιξης με έναν, επίσης, δισδιάστατο πυρήνα $K(x, y)$ περιγράφεται μαθηματικά ως εξής:

$$S_{ij} = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \tag{2.11}$$

όπου

- I είναι ο πίνακας εισόδου (π.χ. εικόνα).
- K είναι το φίλτρο (kernel).
- S_{ij} είναι η τιμή στο σημείο (i,j) του χάρτη χαρακτηριστικών που προκύπτει από τη συνέλιξη.

Η διαδικασία αυτή μετακινεί το φίλτρο πάνω από την είσοδο και εκτελείται ένας πολλαπλασιασμός και προσθετική πράξη μεταξύ των τιμών του φίλτρου και των τιμών της εισόδου. Το φίλτρο «συλλαμβάνει» τοπικά χαρακτηριστικά όπως ακμές, γωνίες ή πιο σύνθετα μοτίβα καθώς το δίκτυο εμβαθύνει.

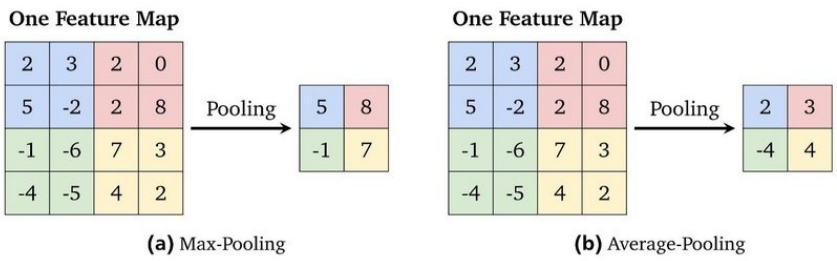


Σχήμα 2.6: Συνελικτικό Επίπεδο Νευρωνικού Δικτύου (CNN)

Σημαντικό να αναφερθεί ότι, όπως όλα τα NN, έτσι και τα Συνελικτικά Δίκτυα χαρακτηρίζονται από μη-γραμμικότητα. Αυτό καθίσταται εφικτό στα σύγχρονα CNN, περνώντας το αποτέλεσμα κάθε συνέλιξης μέσω μιας συνάρτησης ενεργοποίησης και πιο συγκεκριμένα της ReLU. Οι τιμές των image maps απεικονίζονται σε ένα διαφορετικό σύνολο τιμών επιτρέποντας την εκμάθηση πιο σύνθετων χαρακτηριστικών. Έτσι, οι τιμές που τελικά αποθηκεύονται στους χάρτες χαρακτηριστικών δεν είναι, στην πραγματικότητα, τα αθροίσματα των συνελίξεων, αλλά τα αποτελέσματα που προκύπτουν όταν η συνάρτηση ReLU εφαρμόζεται σε αυτά. Συγκεκριμένα η ReLU βοηθά στην αποφυγή του φαινομένου της εξαφάνισης των βαθμίδων, που παρατηρείται σε βαθιά δίκτυα, και επιτρέπει στο δίκτυο να μαθαίνει πιο γρήγορα.

2.3.3 Επίπεδα Υποδειγματοληψίας (Pooling Layers)

Μεταξύ των Convolutional Layers, είναι δυνατόν να παρεμβάλλονται κάποια επίπεδα που μειώνουν τις διαστάσεις των activation maps. Τα επίπεδα αυτά καλούνται pooling layers και στην πράξη χρησιμοποιούνται πιο συχνά δύο είδη υποδειγματοληψίας: Max pooling / Average pooling. Στις περιπτώσεις αυτές καθορίζεται το μέγεθος του παραθύρου στο οποίο θα εφαρμοστεί pooling.



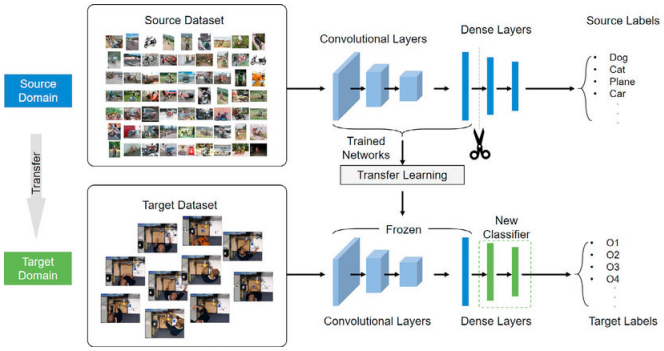
Σχήμα 2.7: Εφαρμογή Δειγματοληψίας σε (Image Map)

2.3.4 Πλήρως Συνδεδεμένα Επίπεδα (Fully Connected Layers)

Τα πλήρως συνδεδεμένα επίπεδα (Fully connected layers) αποτελούν τα τελευταία στρώματα των συνελικτικών δικτύων. Συναντώνται μετά από μια σειρά συνελικτικών και συγκεντρωτικών επιπέδων και οργανώνονται σε ομάδες ενός ή περισσότερων πλήρως συνδεδεμένων στρωμάτων. Όπως φαίνεται και από το όνομά τους, κάθε νευρώνας του επιπέδου αυτού συνδέεται με όλους τους νευρώνες του επόμενου, οπότε στην πραγματικότητα οι ομάδες αυτές λειτουργούν όπως και σε ένα πολυεπίπεδο νευρωνικό δίκτυο. Ως είσοδο λαμβάνουν τον τρισδιάστατο πίνακα χαρακτηριστικών που προέκυψε σαν έξοδος από το αμέσως προηγούμενο συγκεντρωτικό επίπεδο. Ωστόσο, όπως έχει προαναφερθεί, αυτού του είδους τα δίκτυα αναμένουν κάποια μονοδιάστατη είσοδο. Επομένως, εφαρμόζεται η μέθοδος της ισοπέδωσης (flatten) στα δεδομένα εισόδου του επιπέδου αυτού, κατά την οποία ο τρισδιάστατος πίνακας μετασχηματίζεται σε μονοδιάστατο διάνυσμα, χωρίς, φυσικά, να έχουμε απώλεια πληροφορίας. Τα πλήρως συνδεδεμένα επίπεδα αποσκοπούν στην αξιοποίηση των σημείων ενδιαφέροντος που εξήχθησαν από τα προηγούμενα συνελικτικά μπλοκ, ώστε να καταλήξουν στο τελικό αποτέλεσμα. Ο ρόλος τους έγκειται στην ικανότητα εκμάθησης μη γραμμικών συνδυασμών των σημαντικών χαρακτηριστικών της εισόδου, στην εξαγωγή προτύπων και κρυμμένων μοτίβων. Γι' αυτό και συχνά τα συνελικτικά δίκτυα χρησιμοποιούνται σε εφαρμογές ταξινόμησης εικόνων, όπου το τελευταίο πλήρως συνδεδεμένο επίπεδό τους, βάσει της χρήσιμης πληροφορίας, αναδεικνύει την κλάση της εκάστοτε εισόδου.

2.4 Μεταφορά Μάθησης (Transfer Learning)

Η Μεταφορά μάθησης είναι ένα ερευνητικό πρόβλημα στο πεδίο της Μηχανικής Μάθησης που εστιάζει στην αποθήκευση γνώσης, η οποία αποκτήθηκε κατά την επίλυση ενός συγκεκριμένου προβλήματος, και στην εφαρμογή της σε ένα διαφορετικό αλλά παρόμοιο πρόβλημα. Για παράδειγμα, η γνώση που αποκτήθηκε κατά την αναγνώριση ποδηλάτων, μπορεί να χρησιμοποιηθεί για την αναγνώριση μοτοσυκλετών. Τα παρακάτω πλέον, μπορούν να γίνουν κατανοητά, έχοντας θεμελιώσει βασικές έννοιες γύρω από τα νευρωνικά δίκτυα και γενικότερα τα ευφυή υπολογιστικά συστήματα. Με τον όρο απόκτηση γνώσης αναφερόμαστε στην κατάλληλη ρύθμιση βαρών - bias των επιπέδων ενός μοντέλου, που προκύπτουν με τη διαδικασία της εκπαίδευσης. Με τον όρο 'μεταφορά γνώσης' αναφερόμαστε στην διατήρηση των τιμών αυτών των βαρών (κυρίως των αρχικών επιπέδων) και στην σύνθεση τους με ορισμένα άλλα επίπεδα, τα βάρη των οποίων θα εκπαιδευτούν εξ αρχής. Η Μεταφορά μάθησης είναι ιδιαίτερα χρήσιμη όταν έχουμε στη διάθεση μας λίγα δεδομένα εκπαίδευσης για την επίλυση ενός προβλήματος. Όπως έχει αναφερθεί, τα πρώτα επίπεδα ενός δικτύου εξάγουν χαρακτηριστικά χαμηλού επιπέδου, όπως ακμές. Στα επόμενα επίπεδα, τα χαρακτηριστικά αυτά συντίθενται δημιουργώντας πιο σύνθετες αναπαραστάσεις. Για το λόγο, τα αρχικά επίπεδα μπορούν να χρησιμοποιηθούν για κάποιο άλλο (σχετικό) πρόβλημα. Ωστόσο, τα τελικά επίπεδα, που είναι πιο εξειδικευμένα πάνω στο εκάστοτε πρόβλημα, θα πρέπει να εκπαιδευτούν εκ νέου.



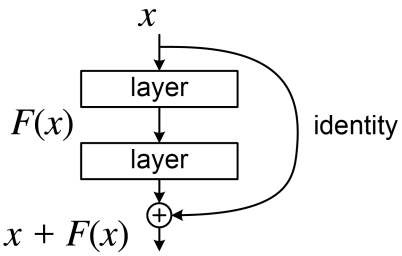
Σχήμα 2.8: Εφαρμογή Transfer Learning

2.5 Αρχιτεκτονικές Συνελικτικών Νευρωνικών Δικτύων

Στην συνεχεια περιγραφονται ορισμενες, εν γενει πολυεπιπεδες, αρχιτεκτονικες που εκπαιδευτηκαν με τα χαρακτηριστικα του παροντος συνολου δεδομενων.

2.5.1 ResNet

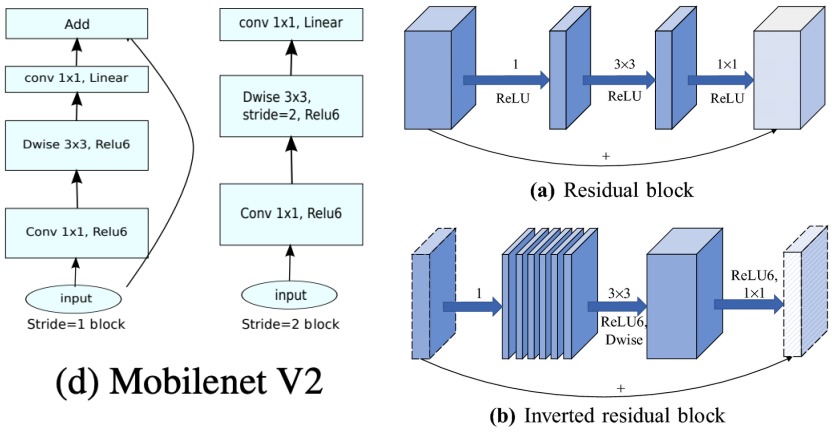
Η ιδέα της αρχιτεκτονικής ResNet[17], προέκυψε από την παρατήρηση ότι βαθύτερες αρχιτεκτονικές είναι πιο δύσκολο να εκπαιδευτούν. Το πρόβλημα αυτό είναι απόρροια των vanishing gradients, το γεγονός δηλαδή πως κατά το backpropagation η παράγωγος λόγω της τιμής των βαρών γίνεται τόσο μικρή με αποτέλεσμα να αδυνατεί να επηρεάσει τα υπόλοιπα βάρη και πρακτικά να επιβραδύνεται η εκπαίδευση του δικτύου ή ακόμη και να σταματάει εντελώς. Έτσι λοιπόν, είναι αρκετά πιθανό πολυεπίπεδες αρχιτεκτονικές να έχουν υψηλότερο training-test error σε σύγκριση με μικρότερες αρχιτεκτονικές. Την απάντηση στο πρόβλημα βρήκαν οι He et al. (2015a) μέσω των Residual Networks, ή ResNets. Το βασικό προτέρημα των ResNets είναι τα residual blocks τα οποία πρακτικά αποτελούν συνδέσεις οι οποίες επιτρέπουν την παράκαμψη ενός ή περισσοτέρων επιπέδων. Σε μια αρχιτεκτονική ResNet, παρεμβάλλονται διασυνδέσεις (shortcut connections) μεταξύ επιπέδων, τα οποία δεν είναι κατά ανάγκη γειτονικά. Έτσι λοιπόν μπορεί να αντιμετωπιστεί το πρόβλημα των vanishing gradients, καθώς κατά τη διαδικασία του backpropagation, η τιμή του gradient μπορεί να διαδοθεί μέσω των shortcut connections. Έτσι είναι λιγότερο πιθανό να συρρικνωθεί στην τιμή μηδέν. Υπάρχουν και μικρότερες εκδοχές των δικτύων όπως τα ResNet18 και ResNet34 τα οποία χρησιμοποιούν residual blocks δύο επιπέδων σε αντίθεση με τα μεγαλύτερα, τα οποία χρησιμοποιούν residual blocks τριών επιπέδων, και χρησιμοποιούνται συνήθως σε περιπτώσεις που τα ResNets με 50 επίπεδα και πάνω είναι πολύ μεγάλα ή/και "βαριά" για κάποιο υπολογιστικό σύστημα. Η δομή μιας μονάδας της αρχιτεκτονικής ResNet (residual block) απεικονίζεται ακολουθώς (Σχήμα 2.9):



Σχήμα 2.9: Δομικό στοιχείο ResNet

2.5.2 MobileNet

Το MobileNet είναι μια οικογένεια αρχιτεκτονικών συνελικτικών νευρωνικών δικτύων (CNN) η οποία αναπτύχθηκε για πρώτη φορά από την Google το 2017, για ταξινόμηση εικόνων, ανίχνευση αντικειμένων και άλλες διεργασίες όρασης υπολογιστών. Είναι σχεδιασμένα για μικρό μέγεθος, χαμηλό latency και χαμηλή κατανάλωση ενέργειας, καθιστώντας τα κατάλληλα για on-δειςε inference και edge computing σε συσκευές με περιορισμένους πόρους, όπως κινητά τηλέφωνα και ενσωματωμένα συστήματα. Στην συγκεκριμένη εργασία χρησιμοποιήθηκε το μοντέλο MobileNetV2 [18], το οποίο αναπτύχθηκε το 2018 και αποτελεί την εξέλιξη του αρχικού μοντέλου. Η βασική του καινοτομία είναι η χρήση inverted residual blocks, γραμμικών bottleneck και depthwise separable convolutions. Τα inverted residuals μειώνουν τις διαστάσεις του χάρτη χαρακτηριστικών χρησιμοποιώντας σημειακές ($1x1$) συνελίξεις, ενώ η κύρια επεξεργασία λαμβάνει χώρα στα επίπεδα κατά βάθος. Κάθε μπλοκ αποτελείται από τρία κύρια στάδια: μια σημειακή συνέλιξη $1x1$ που επεκτείνει τις διαστάσεις των χαρακτηριστικών, μια κατά βάθος διαχωρίσιμη συνέλιξη $3x3$ που εκτελεί το κύριο φιλτράρισμα και μια σημειακή συνέλιξη $1x1$ που μειώνει τις διαστάσεις των χαρακτηριστικών και συνδέεται με ένα γραμμικό bottleneck. Αυτά τα στοιχεία συνδέονται με residual blocks επιτρέποντας στο δίκτυο να μαθαίνει πιο αποτελεσματικά. Χρησιμοποιείται επίσης η ReLU6 ως συνάρτηση μη γραμμικότητας, αντί της απλής ReLU, για να περιοριστούν οι υπολογισμοί.



Σχήμα 2.10: Αρχιτεκτονική MobileNetv2

2.6 Μετασχηματιστές (Transformers)

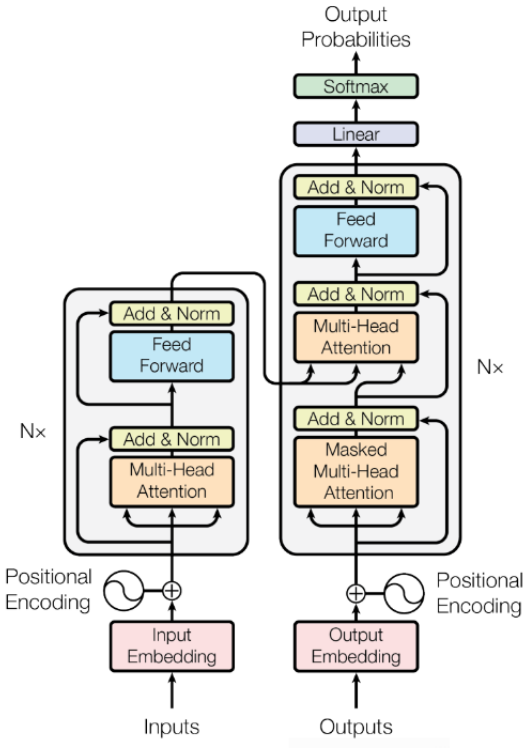
Οι μετασχηματιστές (transformers) αποτελούν σύγχρονα γλωσσικά μοντέλα βαθιάς μηχανικής μάθησης, τα οποία βασίζονται στην αρχιτεκτονική κωδικοποιητή - αποκωδικοποιητή (encoder - decoder) των αναδρομικών νευρικών δικτύων (Recurrent Neural Networks - RNNs) ακολουθίας-σε-ακολουθία (sequence-to-sequence - seq2seq). Βασική παράμετρος στην εκπαίδευση των μετασχηματιστών, είναι η έννοια της "προσοχής" (Attention). Ο μηχανισμός αυτός, λαμβάνει ως είσοδο κωδικοποιημένα διανύσματα και βοηθά το μοντέλο να ξεχωρίσει και να αξιοποιήσει τη χρήσιμη πληροφορία που περιέχουν, ανάλογα με τα συμφραζόμενα. Έτσι, οι αναπαραστάσεις δημιουργούνται με μεγάλη ακρίβεια, ενώ παράλληλα ο χρόνος επεξεργασίας μειώνεται σημαντικά.

- **Μηχανισμός Προσοχής**: Ο μηχανισμός προσοχής επιτρέπει στο μοντέλο να επικεντρωθεί στις λέξεις που σχετίζονται περισσότερο με τη λέξη που επεξεργάζεται. Στην αρχιτεκτονική του μετασχηματιστή, τόσο ο κωδικοποιητής όσο και ο αποκωδικοποιητής χρησιμοποιούν την τεχνική της αυτο-προσοχής (Self-Attention). Η αυτο-προσοχή, υπολογίζει την σημασιολογική σχέση της λέξης με κάθε έναν από τους όρους της ακολουθίας, εξετάζοντας όλους τους πιθανούς τρόπους που μπορεί να σχετίζεται με αυτές και επιλέγονταιεν τέλει αυτές με τις υψηλότερες βαθμολογίες. Στην υλοποίηση του αποκωδικοποιητή, εφαρμοζεται η προσοχή κωδικοποιητή-αποκωδικοποιητή (Encoder-Decoder Attention), μια παραλλαγή του μηχανισμού προσοχής. Ο μηχανισμός αυτός, λαμβάνει την έξοδο των κωδικοποιητών του μετασχηματιστή και σε συνδυασμό με τα αποτελέσματα του επιπέδου αυτο-προσοχής, διαμορφώνει την δική του βαθμολογία για τη σχετικότητα των συμφραζομένων όρων, προσθέτοντάς τη με τη σειρά του στην αναπαράσταση της λέξης.

- **Πολλαπλές Κεφαλές Προσοχής**: Κατά την υλοποίηση του Μετασχηματιστή, ο μηχανισμός προσοχής επαναλαμβάνει τους υπολογισμούς του πολλές φορές, σε παράλληλο χρόνο. Κάθε επανάληψη αντιστοιχεί σε μία κεφαλή προσοχής (Attention Head). Οι έξοδοι που προκύπτουν από τις κεφαλές, ενώνονται και παράγουν μια τελική βαθμολογία προσοχής (Attention score). Με τη χρήση πολλαπλών κεφαλών προσοχής (Multiple Attention Heads), ο μετασχηματιστής ενισχύει την αποτελεσματικότητα της σημασιολογικής κωδικοποίησης της λέξης με τα συμφραζόμενά της, διακρίνοντας τις λέξεις με τις οποίες σχετίζεται.

- **Αρχιτεκτονική**: Η βασική αρχιτεκτονική του μετασχηματιστή, αποτελείται από πολλαπλούς κωδικοποιητές και τους αντίστοιχους αποκωδικοποιητές τους. Κάθε κωδικοποιητής, όπως φαίνεται και στο Σχήμα 2.11, δέχεται ως είσοδο τα διανύσματα ενσωμάτωσης της ακολουθίας εισόδου (Embedding Layer) και την κωδικοποιημένη αναπαράσταση της θέσης της λέξης μέσα στο κείμενο (Position Encoding Layer). Η δομή κάθε επιπέδου κωδικοποίησης είναι πανομοιότυπη και περιλαμβάνει ένα επίπεδο αυτο-προσοχής που εκτιμά τις σχέσεις ανάμεσα στις διαφορετικές λέξεις της ακολουθίας εισόδου που

δέχεται, τα αποτελέσματα του οποίου τροφοδοτούν το δίκτυο πρόσθιας τροφοδότησης που ακολουθεί. Η διαδικασία αυτή επαναλαμβάνεται διαδοχικά, με κάθε επόμενο κωδικοποιητή να δέχεται ως είσοδο την έξοδο του προηγούμενου. Κατά την μετάδοση πληροφορίας από το ένα επίπεδο στο άλλο, πραγματοποιείται κανονικοποίηση επιπέδου (layer normalization), προκειμένου να σταθεροποιηθεί. Στην περίπτωση των αποκωδικοποιητών, ως αρχική είσοδος λαμβάνεται το διάνυσμα ενσωμάτωσης που αναπαριστά την επιθυμητή ακολουθία εξόδου του μοντέλου (target sequence). Όπως και στην περίπτωση των κωδικοποιητών, η αρχιτεκτονική είναι ίδια για κάθε αποκωδικοποιητή. Ανάμεσα στο επίπεδο αυτο-προσοχής και το νευρωνικό δίκτυο πρόσθιας τροφοδότησης, παρεμβάλλεται ένα επιπλέον επίπεδο κωδικοποίησης-αποκωδικοποίησης. Ο ρόλος του είναι ανάλογος με αυτόν του επιπέδου αυτο-προσοχής, με τη διαφορά ότι βασίζει τη λειτουργία του και στην έξοδο του συνόλου των κωδικοποιητών και όχι μόνο στην έξοδο του προηγούμενου επιπέδου. Τα αποτελέσματα του λειτουργούν ως είσοδος στο επίπεδο πρόσθιας τροφοδότησης που ακολουθεί με τον ίδιο τρόπο που ορίζεται στην αρχιτεκτονική των κωδικοποιητών, όπου είσοδος για κάθε επόμενο αποκωδικοποιητή θεωρείται το αποτέλεσμα του προηγούμενου, ενώ σε κάθε υπο-επίπεδο προηγείται η κανονικοποίηση (layer normalization) των επιμέρους εξόδων που προκύπτουν.



Σχήμα 2.11: Αρχιτεκτονική μετασχηματιστή BERT

Για τις ανάγκες αυτής της διπλωματικής εργασίας χρησιμοποιήθηκε ένα από τα δημοφιλέστερα μοντέλα προ-εκπαίδευσης:

2.6.1 BERT

Το μοντέλο BERT (Bidirectional Encoder Representations from Transformers)[26], είναι ένα γλωσσικό μοντέλο το οποίο είναι σχεδιασμένο να εκπαιδεύει βαθιές αμφίδρομες αναπαραστάσεις. Το BERT διαβάζει την ακολουθία εισόδου στο σύνολό της, λαμβάνοντας υπόψη τόσο τις προηγούμενες όσο και τις επόμενες λέξεις ενός όρου. Επομένως, με τη χρήση της αμφίδρομης προσέγγισης, αποκωδικοποιεί με μεγαλύτερη ακρίβεια το νόημα που προσδίδουν τα συμφραζόμενα στη λέξη.

Η αρχιτεκτονική του BERT βασίζεται σε αυτή των μετασχηματιστών, αλλά περιλαμβάνει μόνο τον κωδικοποιητή του μοντέλου, με έναν μηχανισμό αυτο-προσοχής και ένα δίκτυο πρόσθιας τροφοδότησης. Συγκεκριμένα, πρόκειται για μια πολυεπίπεδη δομή κωδικοποιητών, η οποία, βάσει των προηγούμενων και των επόμενων λέξεων μιας λέξης που ανήκει στην ακολουθία εισαγωγής, εκπαιδεύεται να ερμηνεύει σημασιολογικά τη λέξη και να αποδίδει την ακριβή αναπαράστασή της.

Υπάρχουν δύο βασικές εκδοχές του BERT μοντέλου, των οποίων οι δομές, ξεπερνούν σε μέγεθος την απλή έκδοση του μετασχηματιστή (6 επίπεδα κωδικοποιητών, 512 κρυμμένες μονάδες, 8 κεφαλές), καθώς διαθέτουν περισσότερα επίπεδα κωδικοποιητών και μεγαλύτερα δίκτυα πρόσθιας τροφοδότησης:

- BERT$_{BASE}$: Πρόκειται για το βασικό μοντέλο BERT. Αποτελείται από 12 κεφαλές προσοχής και 12 κωδικοποιητές, με 768 κρυφές μονάδες.

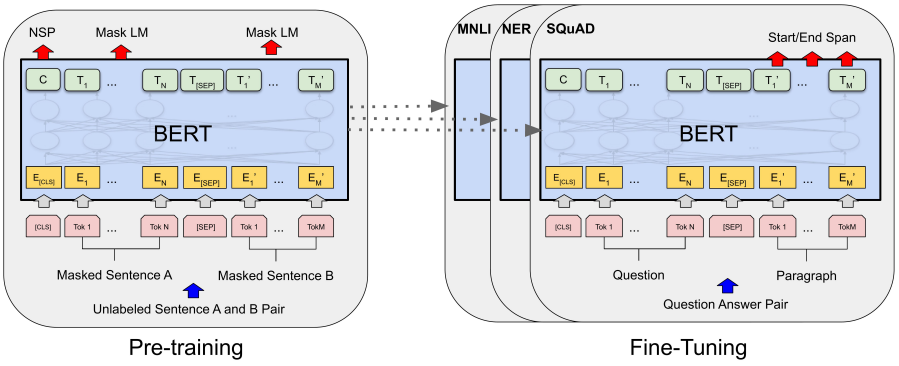
- BERT$_{LARGE}$: Ένα εκτενέστερο μοντέλο BERT μεγάλων διαστάσεων με 16 κεφαλές προσοχής, 24 κωδικοποιητές και 1024 κρυμμένων μονάδων.

Για την προ-εκπαίδευση του BERT, χρησιμοποιούνται 2 βασικές στρατηγικές χωρίς επίβλεψη· το γλωσσικό μοντέλο απόκρυψης (Masked Language Model - MLM) και η πρόβλεψη της επόμενης πρότασης (Next Sentence Prediction - NSP).

- Γλωσσικό μοντέλο απόκρυψης: Τα περισσότερα γλωσσικά μοντέλα εκπαιδεύονται είτε "από αριστερά προς τα δεξιά" είτε "από δεξιά προς τα αριστερά", καθώς η αμφίδρομη επεξεργασία θα είχε ως αποτέλεσμα η κάθε λέξη να "δει τον εαυτό της" και συνεπώς η πρόβλεψη της λέξης αυτής να μην έχει πλέον νόημα. Προκειμένου να γίνει η εκπαίδευση και από τις δύο κατευθύνσεις, αποκρύπτεται το 15% των λέξεων της ακολουθίας εισόδου και το 80% αυτών αντικαθίσταται από το ειδικό σύμβολο [MASK] του μετασχηματιστή. Από τους υπόλοιπους όρους, ένα 10% αντικαθίσταται από μια άλλη διαφορετική λέξη ενώ το υπόλοιπο (10%), παραμένει ίδιο. Έτσι, το μοντέλο μπορεί να εξετάσει αμφίδρομα όλα τα συμφραζόμενα της πρότασης και να επιχειρήσει να προβλέψει τη λέξη που λείπει. Η πρόβλεψη αυτή θα αποτελέσει και την έξοδο του μοντέλου.

- Πρόβλεψη της επόμενης πρότασης: Με τη χρήση της τεχνικής της πρόβλεψης της επόμενης πρότασης, σκοπός του μοντέλου είναι να προβλέψει την ύπαρξη σχέσης (ή μη) ανάμεσα σε 2 προτάσεις. Για την εκπαίδευση του μοντέλου χρησιμοποιείται ένα σύνολο

δεδομένων από ζεύγη προτάσεων. Στο 50% των περιπτώσεων, η δεύτερη πρόταση είναι ακριβώς η ίδια με αυτή που διαδέχεται την πρώτη στο πρωτότυπο κείμενο, ενώ στις υπόλοιπες περιπτώσεις η δεύτερη πρόταση επιλέγεται τυχαία.



Σχήμα 2.12: Διαδικασία προ-εκπαίδευσης και fine-tuning του BERT πάνω στο σύνολο δεδομένων SQuAD (Stanford Question Answering Dataset)

2.7 Αναπαράσταση Αριθμητικών Τιμών

Η αναπαράσταση των αριθμητικών τιμών σε νευρωνικά δίκτυα έχει κεντρικό ρόλο στη διαχείριση της αποδοτικότητας και της πολυπλοκότητας ενός συστήματος. Δεδομένου ότι τα βάρη και οι ενεργοποιήσεις των μοντέλων αποθηκεύονται και υπολογίζονται με βάση αριθμητικές τιμές, η ακρίβεια και το εύρος αυτών των αναπαραστάσεων επηρεάζουν άμεσα την κατανάλωση μνήμης και την υπολογιστική ταχύτητα. Οι υπολογιστές χρησιμοποιούν ένα συγκεκριμένο αριθμό bit για να αναπαραστήσουν οποιαδήποτε πληροφορία. Ένα string με n bit μπορεί να αναπαραστήσει έως 2^n αριθμούς.

2.7.1 Αναπαράσταση κινητης υποδιαστολής (Floating Point)

Τα περισσότερα σύγχρονα νευρωνικά δίκτυα χρησιμοποιούν την αναπαράσταση Floating Point για τα βάρη και τις ενεργοποιήσεις. Η FP32 (32-bit floating-point) αναπαράσταση είναι μια τυποποιημένη μορφή που χρησιμοποιείται για την αναπαράσταση πραγματικών αριθμών με μεγάλη ακρίβεια.

- FP32: Σε αυτήν την αναπαράσταση, οι πραγματικοί αριθμοί αποτελούνται από 32 bits. Τα bits χωρίζονται σε τρία μέρη:
  - 1 bit για το πρόσημο (0 για θετικό, 1 για αρνητικό αριθμό).
  - 8 bits για τον εκθέτη, που καθορίζει τη θέση της υποδιαστολής.
  - 23 bits για τη mantissa (ή συντελεστή), που περιέχει τις σημαντικές πληροφορίες του αριθμού.

Η μορφή FP32 επιτρέπει την αναπαράσταση πολύ μεγάλων ή πολύ μικρών αριθμών με μεγάλη ακρίβεια. Ωστόσο, καταλαμβάνει σημαντικό χώρο μνήμης και απαιτεί πολύπλοκους υπολογισμούς για την εκτέλεση πράξεων. Επίσης, η χρήση floating point αναπαραστάσεων απαιτεί περισσότερη ενέργεια από την εκτέλεση integer πράξεων, κάτι που καθιστά την FP32 ακατάλληλη για συσκευές με περιορισμένη μνήμη ή ισχύ.

2.7.2 Αναπαράσταση Ακεραίων (Integer)

Σε περιβάλλοντα που απαιτούν μικρότερη μνήμη και ταχύτερους υπολογισμούς, χρησιμοποιούνται οι αναπαραστάσεις ακεραίων, όπως το INT8. Οι integer τιμές αναπαριστούν ακέραιους αριθμούς και είναι πιο αποδοτικές στη μνήμη από τους floating-point αριθμούς.

- INT8: Χρησιμοποιεί 8 bits για την αναπαράσταση ακεραίων αριθμών από -128 έως 127. Τα bits χωρίζονται σε δύο μέρη:
  - 1 bit για το πρόσημο (0 για θετικό, 1 για αρνητικό αριθμό).
  - 7 bits για τη mantissa.

Παρόλο που έχει περιορισμένο εύρος αναπαράστασης σε σχέση με το FP32, προσφέρει μεγάλη εξοικονόμηση μνήμης και επιτάχυνση υπολογισμών. Χρησιμοποιείται συνήθως σε εφαρμογές εξαγωγής συμπερασμάτων (inference), όπου οι αριθμητικές πράξεις είναι περιορισμένες.

2.8 Μέθοδοι Βελτιστοποίησης Νευρωνικών Δικτύων

Ο κύριος στόχος της βελτιστοποίησης νευρωνικών δικτύων είναι η μείωση των απαιτήσεων σε μνήμη και υπολογιστική ισχύ, χωρίς σημαντική μείωση στην ακρίβεια του μοντέλου. Αυτό επιτρέπει στα δίκτυα να εκτελούνται πιο γρήγορα και να είναι πιο αποδοτικά, ενώ παράλληλα μειώνει την κατανάλωση ενέργειας και τη χρήση πόρων. Οι στρατηγικές βελτιστοποίησης αφορούν κυρίως:

- Την απλοποίηση της αρχιτεκτονικής των δικτύων.
- Την αφαίρεση των μη αναγκαίων παραμέτρων.
- Την αποτελεσματική αναπαράσταση των δεδομένων και των βαρών.

Η βελτιστοποίηση των νευρωνικών δικτύων έχει καταστεί ένα από τα πιο σημαντικά ζητήματα στην έρευνα και την ανάπτυξη αυτών των μοντέλων, και πολλές τεχνικές έχουν προταθεί για να αντιμετωπιστούν οι προκλήσεις αυτές. Μερικές από τις πιο κοινές προσεγγίσεις που χρησιμοποιούνται για τη μείωση του αποτυπώματος των νευρωνικών δικτύων τόσο σε μνήμη όσο και σε υπολογιστική πολυπλοκότητα είναι:

2.8.1 Κλάδεμα (Pruning)

Το κλάδεμα είναι μια τεχνική στη μηχανική μάθηση, κατά την οποία απομακρύνονται βάρη του δικτύου που έχουν μικρή ή μηδενική επίδραση στο τελικό αποτέλεσμα. Κατά τη διάρκεια της εκπαίδευσης, πολλά από τα βάρη που μαθαίνει το δίκτυο καταλήγουν να έχουν πολύ μικρές τιμές, και η αφαίρεσή τους μπορεί να μειώσει τον αριθμό των υπολογιστικών πράξεων που απαιτούνται, χωρίς να επηρεαστεί σημαντικά η απόδοση. Το κλάδεμα μπορεί να είναι ιδιαίτερα χρήσιμο για τεράστια και πολύπλοκα μοντέλα, όπου η μείωση του μεγέθους τους μπορεί να προκαλέσει σημαντικές βελτιώσεις στην ταχύτητα και την ικανότητά τους.

Υπάρχουν διάφορες τεχνικές κλαδέματος, όπως το δομημένο κλάδεμα (structured pruning), όπου εξαλείφονται ολόκληρα κανάλια συνελίξεων, φίλτρα ή νευρώνες, και το μη δομημένο κλάδεμα (unstructured pruning), όπου αφαιρούνται μεμονωμένα βάρη. Η απόφαση για το ποια τεχνική κλαδέματος θα χρησιμοποιηθεί εξαρτάται από διάφορους παράγοντες όπως ο τύπος του μοντέλου, η προσβασιμότητα των πόρων και ο επιθυμητός βαθμός ακρίβειας. Για παράδειγμα, το δομημένο κλάδεμα μπορεί να εφαρμοστεί πιο αποτελεσματικά σε μοντέλα με δομημένη αρχιτεκτονική, όπως συνελικτικά νευρωνικά δίκτυα (CNN), ενώ το μη δομημένο κλάδεμα είναι καταλληλότερο για μοντέλα χωρίς δομημένη αρχιτεκτονική, όπως πλήρως συνδεδεμένα δίκτυα εγκεφάλου.

2.8.2 Απόσταξη γνώσης (Distillation)

Η απόσταξη γνώσης (knowledge distillation) είναι μια τεχνική που χρησιμοποιείται για τη μεταφορά γνώσης από ένα μεγάλο μοντέλο (το teacher model) σε ένα μικρότερο και πιο ελαφρύ μοντέλο (το student model). Κατά τη διαδικασία αυτή, το μικρότερο μοντέλο εκπαιδεύεται για να μιμηθεί τις προβλέψεις του μεγαλύτερου μοντέλου, διατηρώντας την ίδια απόδοση με μικρότερο αποτύπωμα. Η τεχνική αυτή καθίσταται εξαιρετικά χρήσιμη σε περιπτώσεις όπου οι πόροι είναι περιορισμένοι, όπως στα κινητά τηλέφωνα, ενώ ταυτόχρονα επιτρέπει την ανάπτυξη εφαρμογών που απαιτούν γρήγορη απόκριση και χαμηλή κατανάλωση μνήμης.

2.8.3 Αποσύνθεση Τανυστών (Tensor Decomposition)

Η αποσύνθεση τανυστών είναι μια ακόμη μέθοδος βελτιστοποίησης που μειώνει τον αριθμό των παραμέτρων σε ένα νευρωνικό δίκτυο με το να αναλύει τα βάρη του σε μικρότερες, ανεξάρτητες συνιστώσες. Στα περισσότερα μοντέλα νευρωνικών δικτύων, τα βάρη αποθηκεύονται με τη μορφή μεγάλων τανυστών (πινάκων πολλών διαστάσεων). Η αποσύνθεση αυτών των τανυστών μειώνει τις διαστάσεις τους, με αποτέλεσμα να απαιτούνται λιγότεροι υπολογισμοί. Η τεχνική αυτή είναι ιδιαίτερα χρήσιμη σε δίκτυα με μεγάλα συνελικτικά επίπεδα (CNN), όπου οι τανυστές βαρών μπορούν να καταλαμβάνουν τεράστιες ποσότητες μνήμης. Μέσα από την αποσύνθεση, μπορούμε να διατηρήσουμε την πληροφορία που περιέχεται στα βάρη, αλλά με πολύ χαμηλότερο αποτύπωμα.

2.8.4 Παραγοντοποίηση Χαμηλής Τάξης (Low-Rank Factorization)

Η παραγοντοποίηση χαμηλής τάξης είναι μια παραλλαγή της αποσύνθεσης τανυστών, κατά την οποία τα βάρη αναλύονται σε χαμηλότερης διάστασης πίνακες, οι οποίοι μπορούν να χρησιμοποιηθούν για να αναπαραστήσουν τα αρχικά βάρη με λιγότερες παραμέτρους. Η μέθοδος αυτή μπορεί να μειώσει σημαντικά τη μνήμη που απαιτείται για την αποθήκευση των βαρών, καθώς και τον αριθμό των πράξεων που χρειάζονται κατά την εκτέλεση του δικτύου.

2.8.5 Κβαντοποίηση (Quantization)

Η κβαντοποίηση[2] είναι μια τεχνική βελτιστοποίησης που μειώνει τη χρήση μνήμης και την υπολογιστική πολυπλοκότητα ενός νευρωνικού δικτύου, αντικαθιστώντας την αναπαράσταση των βαρών και των ενεργοποιήσεων από αριθμούς υψηλής ακρίβειας (συνήθως FP32) σε αριθμούς χαμηλότερης ακρίβειας, όπως INT8 ή FP16.

Κεφάλαιο 3

Κβαντοποίηση

3.1 Εισαγωγή στην Κβαντοποίηση

Η κβαντοποίηση (quantization)[2] είναι μια από τις πιο αποτελεσματικές τεχνικές για τη μείωση του αποτυπώματος μνήμης και την επιτάχυνση εκτέλεσης των νευρωνικών δικτύων, με πολύ μικρό αντίκτυπο στην ακρίβεια πρόβλεψης. Στόχος της είναι να μειώσει την ακρίβεια των αριθμητικών αναπαραστάσεων που χρησιμοποιούνται για τα βάρη (weights) και τις ενεργοποιήσεις (activations) του δικτύου, από πιο ακριβείς αναπαραστάσεις όπως τα 32-βιτ floating-point (FP32) σε τύπους δεδομένων χαμηλότερης ακρίβειας, όπως 16, 8 ή 4 bit.

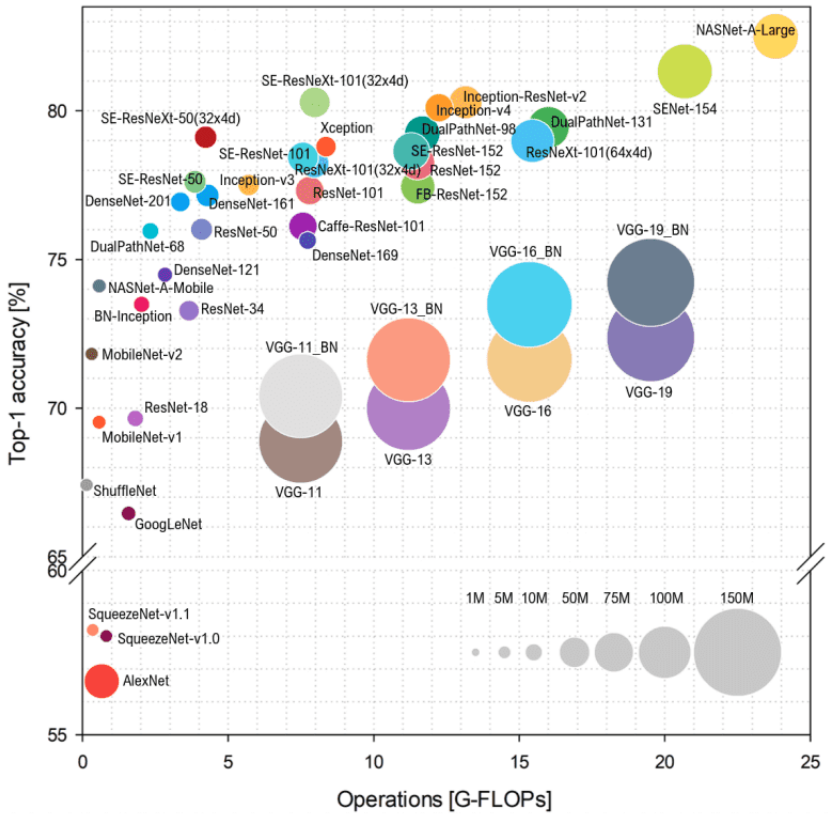
Η συνεχώς αυξανόμενη πολυπλοκότητα των σύγχρονων αρχιτεκτονικών νευρωνικών δικτύων, απαιτεί ραγδαία αυξανόμενη υπολογιστική ισχύ και πόρους μνήμης, ιδιαίτερα όταν εφαρμόζονται σε πραγματικό χρόνο ή σε περιβάλλοντα με περιορισμένους πόρους, όπως φορητές συσκευές, drones ή edge devices. Οι κύριοι λόγοι για τη χρήση της κβαντοποίησης είναι:

- **Μείωση Μνήμης**: Τα κβαντισμένα μοντέλα χρησιμοποιούν μικρότερο αποθηκευτικό χώρο και χαμηλότερη χρήση μνήμης κατά την εκτέλεση.

- **Επιτάχυνση Υπολογισμών**: Οι integer πράξεις (όπως INT8) είναι γρηγορότερες σε πολλές αρχιτεκτονικές υλικού σε σχέση με τις floating-point πράξεις.

- **Ενεργειακή Αποδοτικότητα**: Τα κβαντισμένα μοντέλα απαιτούν λιγότερη ενέργεια, καθιστώντας τα ιδανικά για εφαρμογές με περιορισμούς ισχύος, όπως οι κινητές συσκευές.

- **Υποστήριξη από το Υλικό**: Πολλά μοντέρνα hardware accelerators (π.χ., TPUs, NVIDIA Tensor Cores, ARM processors) είναι ειδικά σχεδιασμένα να εκτελούν κβαντισμένα μοντέλα, μεγιστοποιώντας τα οφέλη.



Σχήμα 3.1: Top-1 accuracy σε σχέση με τον αριθμό operations που απαιτούνται για ένα single forward pass

3.2 Συμμετρική Κβαντοποίηση

Στη συμμετρική κβαντοποίηση, το εύρος των αρχικών τιμών κινητής-υποδιαστολής αντιστοιχίζεται σε ένα συμμετρικό εύρος γύρω από το μηδέν στον κβαντισμένο χώρο. Αυτό σημαίνει ότι η κβαντισμένη τιμή για το μηδέν στον χώρο κινητής υποδιαστολής είναι ακριβώς μηδέν στον κβαντισμένο χώρο. Ένα παράδειγμα μιας μορφής συμμετρικής κβαντοποίησης ονομάζεται absolute maximum (absmax) κβαντοποίηση.

Δεδομένης μιας λίστας τιμών, λαμβάνουμε την υψηλότερη απόλυτη τιμή (α) ως το εύρος για την εκτέλεση της γραμμικής αντιστοίχισης.

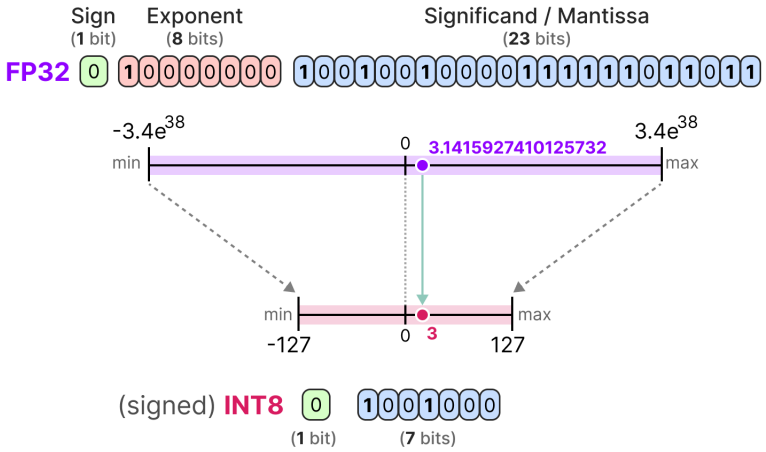
Αρχικά υπολογίζουμε τον συντελεστή κλίμακας (scale factor s):

$$s = \frac{2^{b-1} - 1}{\alpha} = \frac{127}{\alpha} \tag{3.1}$$

όπου

- b είναι ο αριθμός των βιτ στα οποία θέλουμε να κβαντίσουμε (8),
- α είναι η υψηλότερη απόλυτη τιμή,

Αξίζει να σημειωθεί πως το εύρος τιμών [-127, 127] αντιπροσωπεύει το περιορισμένο εύρος.



Σχήμα 3.2: Σχεδιάγραμμα Συμμετρικής Κβαντοποιησης

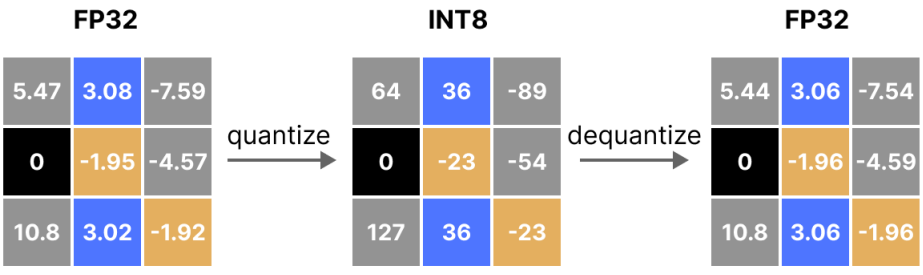
Το απεριόριστο εύρος είναι [-128, 127] και εξαρτάται από τη μέθοδο κβαντοποίησης. Στη συνέχεια, χρησιμοποιούμε το s για να κβαντίσουμε την είσοδο x:

$$x_{quant} = round(s * x) \tag{3.2}$$

Για να ανακτήσουμε τις αρχικές τιμές FP32, μπορούμε να χρησιμοποιήσουμε τον προηγουμένως υπολογισμένο συντελεστή s για να αποκβαντοποιήσουμε τις κβαντισμένες τιμές.

$$x_{dequant} = \frac{x_{quant}}{s} \tag{3.3}$$

Η εφαρμογή της διαδικασίας κβαντοποίησης και στη συνέχεια αποκβαντοποίησης για την ανάκτηση της αρχικής λίστας τιμών φαίνεται ως εξής (Σχήμα 3.3):



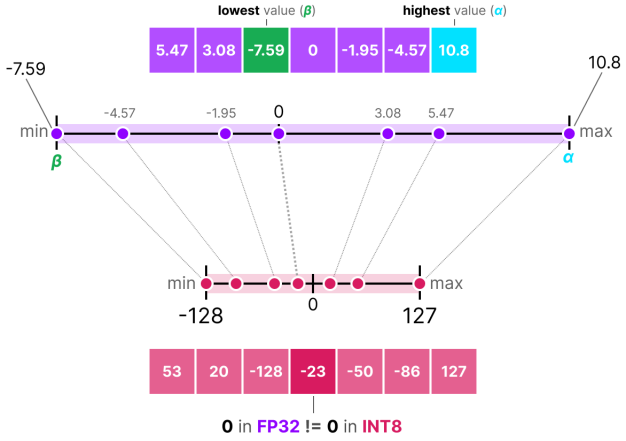
Σχήμα 3.3: Διαδικασία κβαντοποίησης και αποκβαντοποίησης

Όπως βλέπουμε, υπάρχει μια απόκλιση μεταξύ των αρχικών και τελικών τιμών. Αυτό αναφέρεται ως το σφάλμα κβαντοποίησης το οποίο μπορούμε να υπολογίσουμε βρίσκοντας τη διαφορά μεταξύ της αρχικής και της αποκβαντισμένης τιμής. Γενικά, όσο μικρότερος είναι ο αριθμός των bit, τόσο μεγαλύτερο σφάλμα κβαντοποίησης τείνουμε να έχουμε.

3.3 Μη-Συμμετρική Κβαντοποίηση

Η μη-συμμετρική κβαντοποίηση, όπως προδίδει το όνομα της, δεν είναι συμμετρική γύρω από το μηδέν. Αντίθετα, αντιστοιχίζει τις ελάχιστες και τις μέγιστες τιμές από το εύρος float στις ελάχιστες και μέγιστες τιμές του κβαντισμένου εύρους.

Η πιο διαδεδομένη μέθοδος μη-συμμετρικής κβαντοποίησης ονομάζεται κβαντοποίηση σημείου μηδέν (zero-point quantization).



Σχήμα 3.4: Σχεδιάγραμμα Μη-Συμμετρικής Κβαντοποιησης

Λόγω της μετατοπισμένης θέσης του 0, πρέπει να υπολογίσουμε το σημείο μηδέν (zero-point) για την περιοχή INT8 για να εκτελέσουμε τη γραμμική αντιστοίχιση. Όπως και πριν, πρέπει επίσης να υπολογίσουμε έναν συντελεστή κλίμακας, αλλά να χρησιμοποιήσουμε τη διαφορά του εύρους του INT8 [-128, 127]

$$s = \frac{255}{\alpha - \beta} \tag{3.4}$$

όπου α,β η μέγιστη και ελάχιστη τιμή αντίστοιχα. Υπολογίζουμε το zero-point z ως εξής:

$$z = -round(s * \beta) - 128 \tag{3.5}$$

$$x_{quant} = round(s * x + z) \tag{3.6}$$

Για να αποκβαντοποιήσουμε από το INT8 πίσω στο FP32, θα χρειαστεί να χρησιμοποιήσουμε τον προηγουμένως υπολογισμένο συντελεστή κλίμακας (s) και το σημείο μηδέν (z).

$$x_{dequant} = \frac{x_{quant} - z}{s} \tag{3.7}$$

Υπάρχουν δύο επικρατέστερες μέθοδοι κβαντοποίησης για τον υπολογισμό των βαρών και των ενεργοποιήσεων, οι οποίες διαφέρουν ανάλογα με τον τρόπο εφαρμογής τους και τον χρόνο κατά τον οποίο γίνεται η κβαντοποίηση.

3.4 Κβαντοποίηση μετά την Εκπαίδευση (Post-Training Quantization - PTQ)

Μία από τις πιο διαδεδομένες τεχνικές κβαντοποίησης είναι η κβαντοποίηση μετά την εκπαίδευση (PTQ)[3]. Περιλαμβάνει κβαντοποίηση των παραμέτρων ενός μοντέλου (βάρη και ενεργοποιήσεις) μετά την εκπαίδευση του μοντέλου χωρίς να απαιτείται επανεκπαίδευση. Η κβαντοποίηση των παραμέτρων σε χαμηλότερη ακρίβεια πραγματοποιείται είτε συμμετρικά είτε μη-συμμετρικά με τους τρόπους που αναφέρθηκαν προηγουμένως. Θα μελετήσουμε δύο μορφές κβαντοποίησης PTQ:

- Δυναμική κβαντοποίηση
- Στατική κβαντοποίηση

3.4.1 Δυναμική Κβαντοποίηση (Dynamic Quantization)

Η Δυναμική Κβαντοποίηση εφαρμόζεται κυρίως στα βάρη κατά τη διάρκεια της εκτέλεσης του μοντέλου, επιτρέποντας τη μετατροπή των δεδομένων από FP32 σε INT8 «on-the-fly», χωρίς να επηρεάζεται το μοντέλο κατά την εκπαίδευσή του. Αφού τα δεδομένα περάσουν από ένα κρυφό επίπεδο, συλλέγονται οι ενεργοποιήσεις. Η κατανομή των ενεργοποιήσεων αυτών χρησιμοποιείται για τον υπολογισμό του σημείου μηδέν (z) και του συντελεστή κλίμακας (s) με τον τρόπο που περιγράφηκε στο 3.2 και 3.3. Η διαδικασία επαναλαμβάνεται κάθε φορά που τα δεδομένα περνούν από ένα νέο επίπεδο. Επομένως, κάθε επίπεδο έχει τις δικές του ξεχωριστές τιμές z και s και, συνεπώς, διαφορετικά σχήματα κβαντοποίησης.

Σε αυτήν την προσέγγιση, τα βάρη αποθηκεύονται σε μορφή FP32 και κβαντίζονται σε INT8 μόνο κατά τη φάση της εξαγωγής αποτελεσμάτων (inference), ανάλογα με τις απαιτήσεις του εκάστοτε υπολογισμού.

3.4.2 Στατική Κβαντοποίηση (Static Quantization)

Σε αντίθεση με τη δυναμική κβαντοποίηση, η στατική κβαντοποίηση δεν υπολογίζει το σημείο μηδέν (z) και τον συντελεστή κλίμακας (s) κατά τη διάρκεια της εξαγωγής, αλλά εκ των προτέρων. Για να βρεθούν αυτές οι τιμές, χρησιμοποιείται ένα σύνολο δεδομένων calibration και δίνεται στο μοντέλο για τη συλλογή αυτών των πιθανών κατανομών. Αφού συλλεχθούν αυτές οι τιμές, μπορούμε να υπολογίσουμε τις απαραίτητες τιμές s και z για να γίνει η κβαντοποίηση κατά τη διάρκεια του inference.

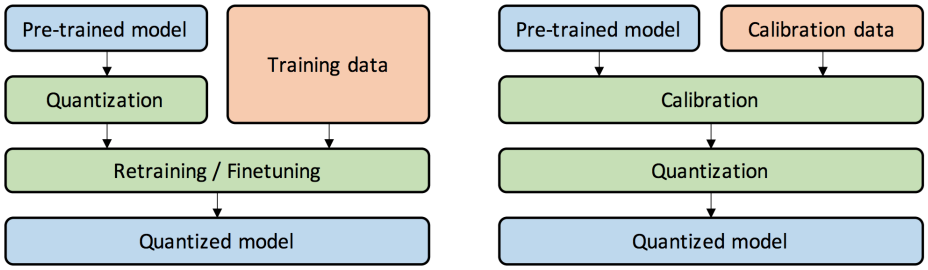
Κατά τη διάρκεια του inference, οι τιμές s και z δεν υπολογίζονται εκ νέου, αλλά χρησιμοποιούνται συνολικά σε όλες τις ενεργοποιήσεις για την κβαντοποίησή τους.

Γενικά, η δυναμική κβαντοποίηση τείνει να είναι πιο ακριβής, καθώς επιχειρεί να υπολογίσει τις τιμές s και z ανά κρυφό στρώμα. Ωστόσο, μπορεί να αυξήσει τον υπολογιστικό χρόνο καθώς αυτές οι τιμές πρέπει να υπολογιστούν. Αντίθετα, η στατική κβαντοποίηση είναι λιγότερο ακριβής, αλλά είναι πιο γρήγορη, καθώς γνωρίζει ήδη τις τιμές s και z που χρησιμοποιούνται για την κβαντοποίηση.

3.5 Κβαντοποίηση με Ευαισθητοποίηση (Quantization-Aware Training - QAT)

Η κβαντοποίηση με ευαισθητοποίηση (QAT)[1] θεωρείται η πιο αποτελεσματική μορφή κβαντοποίησης, καθώς ενσωματώνει τις επιδράσεις της κβαντοποίησης κατά τη διάρκεια της εκπαίδευσης του μοντέλου. Το μοντέλο μαθαίνει να προσαρμόζεται στα σφάλματα που εισάγονται από την κβαντοποίηση, γεγονός που οδηγεί σε μικρότερη πτώση της ακρίβειας συγκριτικά με τη μέθοδο PTQ, όμως ως διαδικασία απαιτεί περισσότερους υπολογιστικούς πόρους. Η QAT εφαρμόζεται κυρίως σε εφαρμογές που απαιτούν υψηλή ακρίβεια μετά την κβαντοποίηση, όπως σε πολύπλοκα δίκτυα βαθιάς μάθησης για αναγνώριση εικόνων ή επεξεργασία φυσικής γλώσσας.

Κατά τη διάρκεια της εκπαίδευσης, εισάγονται τα λεγόμενα «ψεύτικα» quants. Αυτή είναι η διαδικασία κβαντοποίησης των βαρών σε, για παράδειγμα, INT8 και μετά αποκβαντοποίησης πίσω στο FP32. Σε αυτή την περίπτωση, η κβαντοποίηση μοντελοποιείται μέσα στον βρόχο της εκπαίδευσης, και τα βάρη προσαρμόζονται μέσω της διαδικασίας backpropagation έτσι ώστε να ελαχιστοποιηθούν οι απώλειες λόγω κβαντοποίησης.



Σχήμα 3.5: Σύγκριση διαδικασίας Κβαντοποίησης με Ευαισθητοποίηση (QAT) (αριστερά) και Κβαντοποίησης μετά την Εκπαίδευση (PTQ) (δεξιά)

Κεφάλαιο 4

Πειραματική Διαδικασία

4.1 Περιγραφή - Εργαλεία

Για την υλοποίηση της παρούσας διπλωματικής εργασίας έγινε χρήση της γλώσσας προγραμματισμού Python στο προγραμματιστικό περιβάλλον Google Colaboratory. Οι βασικές βιβλιοθήκες που συμπεριλήφθηκαν κατά την εκτέλεση συνοψίζονται ακολούθως:

- numpy: Βιβλιοθήκη της Python που παρέχει υποστήριξη για πολυδιάστατους πίνακες και συναρτήσεις μαθηματικών πράξεων.
- matplotlib: Περιεκτική βιβλιοθήκη για την οπτικοποίηση των αποτελεσμάτων.
- time: Η βιβλιοθήκη αυτή επιτρέπει την ακριβή μέτρηση και διαχείριση του χρόνου, παρέχοντας λειτουργίες για χρονικές καθυστερήσεις και παρακολούθηση χρονικών στιγμών.
- torch: Βιβλιοθηκη για εφαρμογες μηχανικης μαθησης, με εστιαση στην εκπαιδευση νευρωνικων δικτυων.
- torchvision: Επέκταση της βιβλιοθήκης torch, προσφέροντας προκαθορισμένα σύνολα δεδομένων, μοντέλα και συναρτήσεις επεξεργασίας εικόνων για εφαρμογές όρασης υπολογιστών.

Η επιτάχυνση της εκτέλεσης αριθμητικών πράξεων, και γενικότερα η εκπαίδευση των μοντέλων, έγινε εφικτή με τη χρήση κάρτας γραφικών GPU. Το περιβάλλον Google Colaboratory παρέχει διαφορετικούς τύπους GPU κάθε φορά, χωρίς ωστόσο να δίνεται η δυνατότητα επιλογής. Οι διαθέσιμες κάρτες γραφικών είναι: Nvidia K80s, T4s, P4s και P100s. Οι υλοποιήσεις χβαντοποίησης του PyTorch είναι ωστόσο κατά κύριο λόγο βελτιστοποιημένες για CPU inference, ειδικά για στατική και δυναμική χβαντοποίηση. Οι χβαντισμένες λειτουργίες σε GPU εξακολουθούν να περιορίζονται σε συγκεκριμένες περιπτώσεις χρήσης και υλικό, αλλά βελτιώνονται με νέες ενημερώσεις στο CUDA και το cuDNN.

4.2 Δεδομένα και Μοντέλα

Χρησιμοποιήθηκαν δύο σύνολα δεδομένων για την δεδομένη εργασία

- **CIFAR-100**: Το CIFAR-100 είναι ένα δημοφιλές σύνολο δεδομένων που χρησιμοποιείται ευρέως στην έρευνα μηχανικής μάθησης και ειδικότερα σε εφαρμογές της υπολογιστικής όρασης. Περιέχει 100 κλάσεις, όπου κάθε μία περιλαμβάνει 600 έγχρωμες εικόνες ανάλυσης $32x32$ pixels, οι οποίες χωρίζονται σε 500 εικόνες εκπαίδευσης και 100 εικόνες δοκιμής. Οι κλάσεις αυτές είναι οργανωμένες σε 20 υπερκλάσεις (superclasses), όπου κάθε μία περιλαμβάνει πέντε κατηγορίες. Κάθε εικόνα συνοδεύεται από ένα fine label (η τάξη στην οποία ανήκει) και ένα coarse label (η υπερκλάση στην οποία ανήκει). Για παράδειγμα, η υπερκατηγορία ¨ερπετά' περιλαμβάνει κατηγορίες όπως σαύρα, φίδι, κροκόδειλος, δεινόσαυρος. Οι εικόνες στο CIFAR-100 χαρακτηρίζονται από ποικιλία σε χρώματα, υφές και σύνθεση, γεγονός που καθιστά το σύνολο δεδομένων ιδιαίτερα χρήσιμο για την εκπαίδευση και αξιολόγηση μοντέλων βαθιάς μάθησης, όπως τα Convolutional Neural Networks (CNNs). Το CIFAR-100 θεωρείται πιο απαιτητικό από τον πρόγονό του, το CIFAR-10, καθώς διαθέτει περισσότερες κατηγορίες και μεγαλύτερη ποικιλία οπτικών χαρακτηριστικών. Στην παρούσα εργασία χρησιμοποιήθηκε για την εκπαίδευση και αξιολόγηση των μοντέλων ResNet-50, ResNet-34, ResNet-18 και MobileNetV2.

- **SQuAD Dataset**: Το SQuAD (Stanford Question Answering Dataset) είναι ένα από τα πιο δημοφιλή σύνολα δεδομένων για την εκπαίδευση και αξιολόγηση μοντέλων επεξεργασίας φυσικής γλώσσας, ιδίως για εφαρμογές ερωταπαντήσεων. Πρόκειται για ένα σύνολο δεδομένων κατανόησης ανάγνωσης, που αποτελείται από ερωτήσεις που έθεσαν οι εργαζόμενοι στο πλήθος σε ένα σύνολο άρθρων της Wikipedia, όπου η απάντηση σε κάθε ερώτηση είναι ένα τμήμα κειμένου από το αντίστοιχο απόσπασμα ανάγνωσης ή η ερώτηση μπορεί να είναι αναπάντητη. Η πρώτη έκδοση του SQuAD, γνωστή ως SQuAD 1.1, περιέχει πάνω από 100.000 ερωταπαντήσεις, όπου κάθε ερώτηση συνοδεύεται από ένα αντίστοιχο χωρίο κειμένου από το οποίο μπορεί να εξαχθεί η απάντηση. Οι απαντήσεις σε αυτήν την έκδοση είναι πάντα τμήματα του χωρίου, και το σύνολο δεδομένων είναι σχεδιασμένο έτσι ώστε τα μοντέλα να πρέπει να εντοπίζουν τις σχετικές πληροφορίες μέσα σε ένα συγκεκριμένο πλαίσιο. Στη συνέχεια, κυκλοφόρησε η δεύτερη έκδοση, το SQuAD 2.0, το οποίο περιλαμβάνει επιπλέον 50.000 ερωτήσεις που δεν έχουν απαντήσεις μέσα στα χωρία κειμένου, προκειμένου να ελεγχθεί η ικανότητα των μοντέλων να χειρίζονται αναπάντητα ερωτήματα. Το SQuAD θεωρείται πρότυπο για την αξιολόγηση της ικανότητας των μοντέλων στην κατανόηση κειμένου και στην εξαγωγή πληροφοριών και έχει χρησιμοποιηθεί ευρέως για την εκπαίδευση μεγάλων γλωσσικών μοντέλων όπως το BERT στην προκειμένη περίπτωση.

Στην παρούσα εργασία, επιλέξαμε να μελετήσουμε τη βελτιστοποίηση των ακόλουθων αρχιτεκτονικών νευρωνικών δικτύων, τα οποία είναι γνωστά για την απόδοσή τους αλλά και για την πολυπλοκότητά τους:

- **ResNet-50, ResNet-34, ResNet-18**: Αυτές οι παραλλαγές του ResNet (Residual Network) έχουν χρησιμοποιηθεί ευρέως στην αναγνώριση εικόνων λόγω της ικανότητάς τους να διαχειρίζονται το πρόβλημα των vanishing gradient. Ωστόσο, η πολυπλοκότητά τους αυξάνει το αποτύπωμα μνήμης και την ανάγκη για υπολογιστική ισχύ, ιδιαίτερα στην περίπτωση του ResNet-50, το οποίο έχει μεγαλύτερο αριθμό επιπέδων και παραμέτρων.

- **MobileNetV2**: Το MobileNetV2 έχει σχεδιαστεί για να είναι ελαφρύ και αποδοτικό, καθιστώντας το κατάλληλο για κινητές συσκευές. Παρ΄ όλα αυτά, ακόμα και αυτό το μοντέλο μπορεί να επωφεληθεί από περαιτέρω βελτιστοποίηση, ιδίως σε εφαρμογές όπου η ακόμη μεγαλύτερη μείωση του αποτυπώματος είναι απαραίτητη.

- **BERT-Base-Cased-Squad2**: Το μοντέλο που χρησιμοποιήθηκε σε αυτή την εργασία είναι μια παραλλαγή του BERT (Bidirectional Encoder Representations from Transformers), γνωστό ως 'Q-A-Model-Bert-Base-Cased-Squad2'. Το BERT, που προτάθηκε αρχικά από την Google, είναι γνωστό για τις αξιοσημείωτες προόδους του στην Επεξεργασία Φυσικής Γλώσσας (NLP) λόγω της αμφίδρομης συνείδησής του. Αυτή η συγκεκριμένη παραλλαγή του BERT έχει υποστεί fine-tuning στο SQuAD 2.0, ένα γνωστό σημείο αναφοράς για την αξιολόγηση των δυνατοτήτων απάντησης ερωτήσεων.

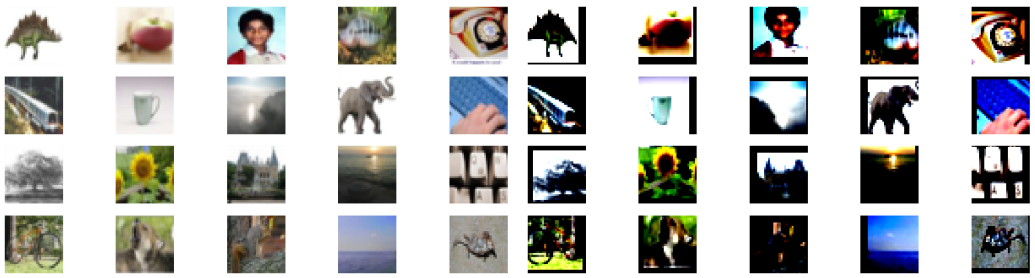
Task	Architecture	parameters (millions)
Image Classification	ResNet-50	25.6
	ResNet-34	21.8
	ResNet-18	11.7
	MobileNetV2	3.5
Question Answering	BERT-base-cased (SQuAD)	110

Πίνακας 4.1: Περιγραφή χρήσης Μοντέλων

4.3 Προεπεξεργασία Δεδομένων

Δεδομένου του ότι επιλέξαμε το CIFAR-100 ως το σύνολο δεδομένων μας, χρειάστηκε να ρυθμίσουμε το DataLoader, με σκοπό την προετοιμασία των εικόνων για χρήση με τα προεκπαιδευμένα μοντέλα που περιγράφηκαν, καθώς τα συγκεκριμένα μοντέλα έχουν εκπαιδευτεί αρχικά στο σύνολο δεδομένων ImageNet. Αρχικά, εφαρμόζονται διάφορες μετασχηματισμοί στις εικόνες με τη χρήση της βιβλιοθήκης torchvision.transforms. Συγκεκριμένα, εφαρμόζεται τυχαία οριζόντια αναστροφή (RandomHorizontalFlip) και περικοπή (RandomCrop), που ενισχύουν τα δεδομένα (data augmentation) ώστε το μοντέλο να γενικεύσει καλύτερα. Στη συνέχεια, η μετατροπή των εικόνων σε tensor και η κανονικοποίησή τους με βάση τις μέσες τιμές και τις τυπικές αποκλίσεις των καναλιών RGB του CIFAR-100 (mean και std), προσαρμόζουν το εύρος τιμών των δεδομένων ώστε να ταιριάζει περισσότερο με τα δεδομένα

στα οποία εκπαιδεύτηκαν τα μοντέλα. Αν και το ImageNet περιλαμβάνει εικόνες υψηλότερης ανάλυσης ($224x224$) και ευρύτερης ποικιλίας σε σύγκριση με το CIFAR-100 ($32x32$), η εφαρμογή των προσαρμογών στις εικόνες του CIFAR-100 βοηθά στο να προσομοιωθούν οι συνθήκες των εικόνων του ImageNet. Η κανονικοποίηση επιτρέπει καλύτερη σύγκλιση κατά τη διαδικασία εκπαίδευσης και προσαρμογής (fine-tuning) στο CIFAR-100, βελτιώνοντας την απόδοση του προ-εκπαιδευμένου μοντέλου σε νέα δεδομένα. Χρησιμοποιώντας αυτό το σύνολο δεδομένων, το μοντέλο μπορεί να διατηρήσει τις αρχικές δυνατότητες εκμάθησης που αναπτύχθηκαν στο ImageNet, ενώ παράλληλα προσαρμόζεται στα χαρακτηριστικά των μικρότερων, πιο εξειδικευμένων εικόνων του CIFAR-100.



Σχήμα 4.1: Δείγμα εικόνων του CIFAR-100 πριν και μετά την επεξεργασία

4.4 Πειραματική Διαδικασία

4.4.1 Μοντέλα CNN

Μετά την προεπεξεργασία των δεδομένων, τα μοντέλα εκπαιδεύτηκαν στο CIFAR-100 σε 50 epochs ώστε να μην έχουμε overfitting. Στη συνέχεια υπολογίστηκε το μέγεθος του κάθε μοντέλου στην fp32 μορφή του (baseline μοντέλο), η ακρίβεια πάνω στο σύνολο δεδομένων test και ο χρόνος inference. Για τη διαδικασία της κβαντοποίησης, το μοντέλο φορτώνεται στη συσκευή εκτέλεσης (device) και τίθεται σε κατάσταση αξιολόγησης (evaluation mode), κάτι που είναι απαραίτητο για την κβαντοποίηση, καθώς απενεργοποιεί τα τυχόν στοιχεία εκπαίδευσης (π.χ., dropout).

- **Στατική Κβαντοποίηση**: Χρησιμοποιούμε μια λίστα modules_to_fuse, που περιέχει ζεύγη ή τριάδες επιπέδων του μοντέλου, ανάλογα με την αρχιτεκτονική του, τα οποία θέλουμε να συγχωνεύσουμε για καλύτερη αποδοτικότητα. Κάθε ζεύγος (conv, bn) αναφέρεται σε ένα επίπεδο συνέλιξης (conv) και το αντίστοιχο επίπεδο κανονικοποίησης παρτίδας (bn) που μπορούν να συγχωνευτούν. Η συγχώνευση μειώνει την πολυπλοκότητα των υπολογισμών και βελτιώνει την ταχύτητα του μοντέλου σε περιβάλλοντα με περιορισμένους πόρους. Η εντολή torch.quantization.prepare προσθέτει ειδικά επίπεδα παρακολούθησης (observers) στο μοντέλο, ώστε να μετρούνται τα στατιστικά στοιχεία κάθε επιπέδου κατά τη φάση του calibration (διαδικασία συλλογής στατιστικών). Το μοντέλο στη συνέχεια περνάει από τα δεδομένα εκπαίδευσης (train_loader) ώστε να

συλλεχθούν στατιστικά στοιχεία σχετικά με το εύρος των τιμών κάθε επιπέδου. Αυτά τα στατιστικά βοηθούν στον καθορισμό των κβαντικών περιοχών τιμών. Τέλος, με τη χρήση της torch.quantization.convert, το μοντέλο μετατρέπεται σε κβαντοποιημένη μορφή με βάση τα συλλεχθέντα στατιστικά. Στο δεύτερο πείραμα παραλείφθηκε η συγχώνευση των modules με σκοπό τη σύγκριση της αποδοτικότητας της κβαντοποίησης.

- **Δυναμική Κβαντοποίηση**: Η δυναμική κβαντοποίηση εφαρμόζεται μόνο στα linear επίπεδα με τη χρήση της torch.quantization.quantize_dynamic. Σε αυτήν την περίπτωση, όπως έχει αναφερθεί προηγουμένως, η κβαντοποίηση στα βάρη του μοντέλου γίνεται κατά την εκτέλεση (runtime), επομένως δεν απαιτεί καλιμπράρισμα στα δεδομένα.

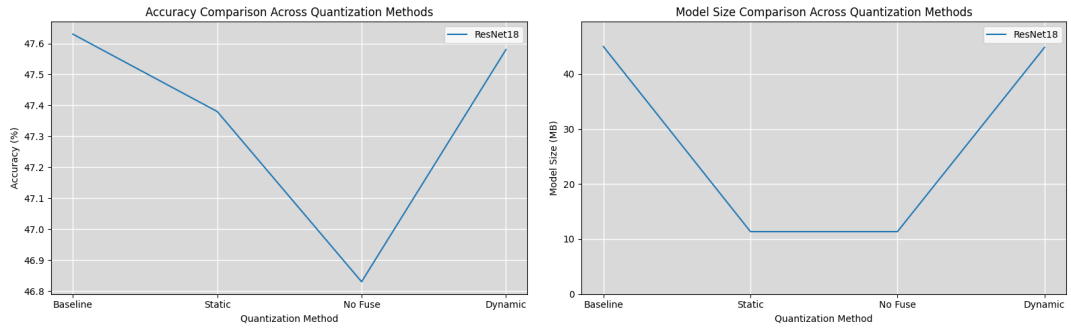
Τέλος υπολογίζουμε ξανά τα νέα μεγέθη των μοντέλων, την ακρίβεια και τον χρόνο inference.

4.4.2 BERT

Το μοντέλο BertForQuestionAnswering και το tokenizer φορτώνονται από την προκαθορισμένη αρχιτεκτονική του BERT (deepset/bert-base-cased-squad2). Το μοντέλο τίθεται σε κατάσταση αξιολόγησης (model.eval()) για να απενεργοποιηθούν οι λειτουργίες εκπαίδευσης (π.χ., dropout). Καθώς επιλέξαμε το μοντέλο BERT-Base-Cased-Squad2, το οποίο ήταν ήδη εκπαιδευμένο στο dataset που χρησιμοποιήθηκε, δεν έγινε κάποια προεπεξεργασία στα δεδομένα και περαιτέρω εκπαίδευση. Εφαρμόστηκε δυναμική κβαντοποίηση στα Linear επίπεδα του μοντέλου, με τύπο δεδομένων int8 με τη χρήση της torch.quantization.quantize_dynamic. Έγινε χρήση ενός μέρους του συνόλου δεδομένων SQuAD για να αξιολογηθεί η ακρίβεια του μοντέλου και υπολογίστηκαν επίσης το μέγεθος του μοντέλου πριν και μετά την κβαντοποίηση, όπως και ο χρόνος inference.

4.5 Αποτελέσματα

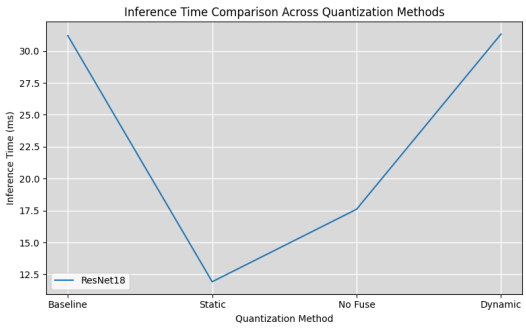
ResNet-18



Το ResNet18 είναι ένα μοντέλο βάθους 18 επιπέδων, με αρχιτεκτονική που χρησιμοποιεί residual blocks για την αποτροπή του φαινομένου vanishing gradients κατά την εκπαίδευση

	Model Size (MB)	Model Size Difference (%)	Accuracy (%)	Accuracy Difference (%)	Elapsed Time (ms)	Time Difference (%)
Baseline (FP32)	44.98	-	47.63%	-	31.20	-
Static Quantization	11.36	-74.75%	47.38%	-0.5%	11.93	-61.78%
Static (No-Fuse)	11.36	-74.75%	46.83%	-1.6%	17.60	-43.60%
Dynamic Quantization	44.83	-0.34%	47.58%	-0.1%	31.32	+0.37%

Πίνακας 4.2: Αποτελέσματα ResNet18



Σχήμα 4.2: Σύγκριση στην απόδοση το μέγεθος μοντέλου και τον χρόνο inference ανά μέθοδο κβαντοποίησης για το ResNet18

βαθιών νευρωνικών δικτύων. Με τη στατική κβαντοποίηση, το μέγεθος του μοντέλου μειώνεται σε 11.36 MB, προσφέροντας μια μείωση μνήμης σχεδόν 75%, καθώς η ακρίβεια των αριθμητικών αναπαραστάσεων μειώθηκε από 32bit σε 8. Ο χρόνος εξαγωγής αποτελεσμάτων βελτιώνεται αισθητά κατά 62%, λόγω της μείωσης των υπολογιστικών απαιτήσεων και αυτό επιτεύχθηκε με ελάχιστη μείωση της ακρίβειας κατά 0.5%. Η μείωση της ακρίβειας οφείλεται στην απώλεια πληροφοριών, λόγω της κβαντοποίησης των ενεργοποιήσεων σε 8-bit βάρη, που δεν μπορούν να αναπαραστήσουν όλες τις διαφοροποιήσεις στις τιμές των δεδομένων όπως η αναπαράσταση FP32. Επιπλέον, το ResNet18 βασίζεται στις εναλλαγές μεταξύ των επιπέδων για την εξαγωγή χαρακτηριστικών, και η σταθερή κβαντοποίηση μπορεί να περιορίσει αυτή την ικανότητα του μοντέλου.

Στο δεύτερο πείραμα που εκτελέστηκε, όπου δεν εφαρμόστηκε συγχώνευση των modules, το μέγεθος του μοντέλου παραμένει 11.36 MB, παρόμοιο με την αρχική στατική κβαντοποίηση με fusion, αλλά ο μέσος χρόνος εκτέλεσης αυξάνεται στα 17.60 ms και έχουμε μείωση της ακρίβειας κατά 1.6%, δηλαδή 1.1% διαφορά στην απόδοση και 18% στον χρόνο inference.

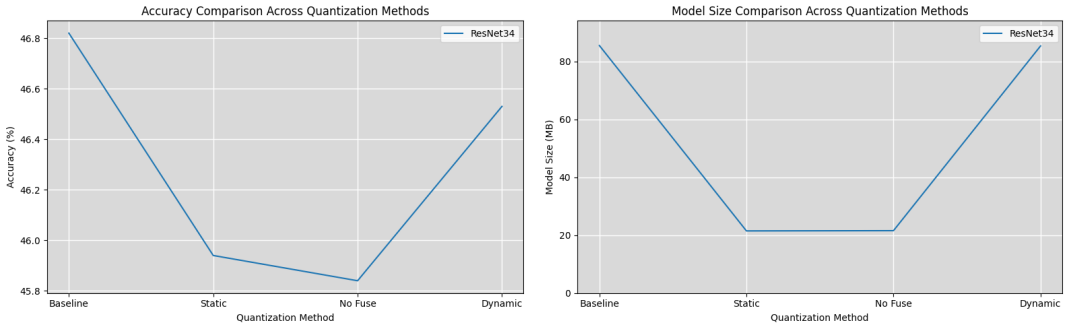
Με τη δυναμική κβαντοποίηση, το μέγεθος του μοντέλου παραμένει παρόμοιο με το Base-

line (fp32), στα 44.83 MB, καθώς οι ενεργοποιήσεις δεν κβαντοποιούνται εκ των προτέρων αλλά υπολογίζονται κατά την εκτέλεση. Η ακρίβεια του μοντέλου και ο μέσος χρόνος εκτέλεσης επηρεάζονται κάτω από 0.5%, παραμένουν παρόμοια δηλαδή με τα αρχικά metrics. Η δυναμική κβαντοποίηση μειώνει μόνο την ακρίβεια των βαρών σε INT8 κατά την εκτέλεση, αφήνοντας τις ενεργοποιήσεις σε FP32. Αυτό επιτρέπει στο μοντέλο να διατηρήσει ακρίβεια παρόμοια με το Baseline, καθώς οι ενεργοποιήσεις έχουν τη δυνατότητα να προσαρμόζονται στις εισόδους κατά τη διάρκεια εκτέλεσης, αλλά δεν επιτυγχάνει μείωση του αποτυπώματος μνήμης.

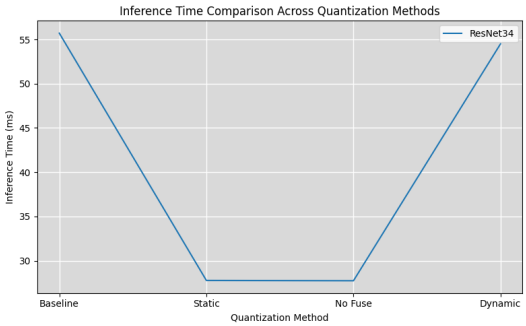
ResNet-34

	Model Size (MB)	Model Size Difference (%)	Accuracy (%)	Accuracy Difference (%)	Elapsed Time (ms)	Time Difference (%)
Baseline (FP32)	85.47	-	46.82%	-	55.72	-
Static Quantization	21.48	-74.87%	45.94%	-1.87%	27.76	-50.19%
Static (No-Fuse)	21.57	-74.77%	45.84%	-2.09%	27.73	-50.23%
Dynamic Quantization	85.32	-0.18%	46.53%	-0.62%	54.51	-2.17%

Πίνακας 4.3: Αποτελέσματα ResNet34



Το ResNet34 είναι ένα πιο βαθύ μοντέλο από το ResNet18, με 34 επίπεδα, και χρησιμοποιεί επίσης residual blocks για την αποτροπή του φαινομένου των vanishing gradients, αλλά με περισσότερες παραμέτρους λόγω του πρόσθετου βάθους. Η στατική κβαντοποίηση στο ResNet34 μειώνει το μέγεθος του μοντέλου σε 21.48 MB, προσφέροντας μια μείωση μνήμης σχεδόν 75%, καθώς τα βάρη και οι ενεργοποιήσεις κβαντοποιούνται σε 8-bit αναπαραστάσεις. Αυτή η μείωση του μεγέθους συνοδεύεται από επιτάχυνση στον χρόνο εξαγωγής αποτελεσμάτων (inference), μείωση κατά περίπου 50% στον χρόνο εκτέλεσης, σε σχέση με το αρχικό



Σχήμα 4.3: Σύγκριση στην απόδοση το μέγεθος μοντέλου και τον χρόνο inference ανά μέθοδο κβαντοποίησης για το ResNet34

μοντέλο. Παρατηρείται μια μικρή αλλά μεγαλύτερη μείωση στην ακρίβεια σε σχέση με το ResNet18, της τάξης του 1.87%. Αποδεικνύεται λοιπόν πως οι επιπτώσεις της κβαντοποίησης στην ακρίβεια των μοντέλων είναι πιο έντονες σε πιο βαθιές αρχιτεκτονικές δικτύων.

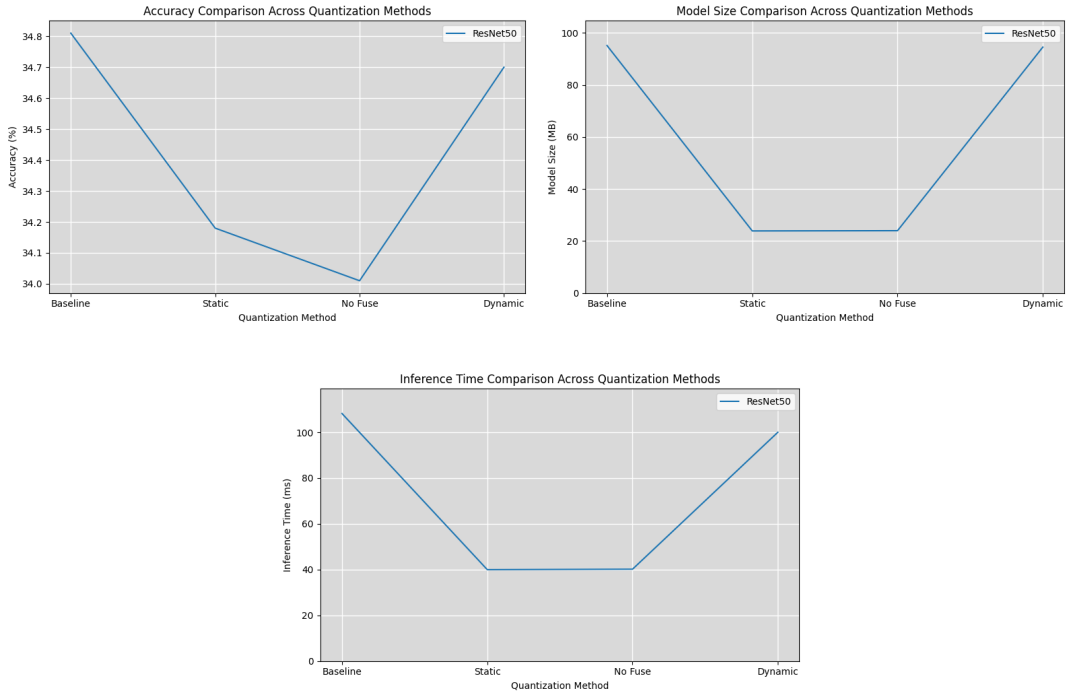
Στο πείραμα όπου η στατική κβαντοποίηση εφαρμόστηκε χωρίς fusion στα modules, το μέγεθος του μοντέλου παρουσιάζει μια πολύ μικρή αύξηση σε σχέση με τη στατική κβαντοποίηση με fusion (21.57 MB). Παράλληλα, ο χρόνος εκτέλεσης αυξάνεται ελαφρώς, φτάνοντας στα 27.73 ms, ενώ παρατηρείται και ελαφρώς μεγαλύτερη μείωση της ακρίβειας, κατά 2.09%.

Η δυναμική κβαντοποίηση στο ResNet34 αφήνει τις ενεργοποιήσεις σε FP32 και κβαντοποιεί μόνο τα βάρη σε INT8 κατά την εκτέλεση, με αποτέλεσμα το μέγεθος του μοντέλου να παραμένει κοντά στο αρχικό (85.32 MB). Αυτή η μέθοδος έχει ως αποτέλεσμα σχεδόν μηδενική μείωση της ακρίβειας (0.62%), ενώ ο χρόνος εκτέλεσης παραμένει κοντά στο baseline.

ResNet-50

	Model Size (MB)	Model Size Difference (%)	Accuracy (%)	Accuracy Difference (%)	Elapsed Time (ms)	Time Difference (%)
Baseline (FP32)	95.15	-	34.81%	-	108.22	-
Static Quantization	23.90	-74.87%	34.12%	-1.97%	39.99	-63.05%
Static (No-Fuse)	24.00	-74.78%	34.01%	-2.29%	40.19	-62.86%
Dynamic Quantization	94.54	-0.64%	34.8%	-0.02%	100.02	-7.58%

Πίνακας 4.4: Αποτελέσματα ResNet50



Σχήμα 4.4: Σύγκριση στην απόδοση το μέγεθος μοντέλου και τον χρόνο inference ανά μέθοδο χβαντοποίησης για το ResNet50

Το ResNet50 είναι ένα πολύπλοκο μοντέλο που διαθέτει αρχιτεκτονική με bottleneck blocks, αντί για τα βασικά (basic) βλοχς που χρησιμοποιούνται στα 2 προηγούμενα μοντέλα. Η χρήση αυτών των bottleneck blocks καθιστά το μοντέλο πιο ευαίσθητο στις επιπτώσεις της χβαντοποίησης, καθώς απαιτείται υψηλή ακρίβεια για να αποδοθούν τα συμπιεσμένα χαρακτηριστικά που περνούν μέσα από αυτά.

Με τη στατική χβαντοποίηση και το fusion των επιπέδων, το μέγεθος του ResNet50 μειώνεται από 95.15 MB σε 23.90 MB, επιτυγχάνοντας μείωση μνήμης σχεδόν 75%, παρόμοια με τα μικρότερα μοντέλα της αρχιτεκτονικής ResNet. Ο χρόνος εκτέλεσης βελτιώνεται αισθητά, από 108.22 ms σε 39.99 ms (63%), λόγω της χβαντοποίησης και της μείωσης των υπολογιστικών απαιτήσεων. Παρόλα αυτά, το ResNet50 εμφανίζει μεγαλύτερη μείωση στην ακρίβεια σε σύγκριση με τα ResNet18 και ResNet34. Παρατηρείται πτώση 1.97% στην ακρίβεια, η οποία είναι πιο αισθητή λόγω της πιο πολύπλοκης δομής του και της υψηλότερης εξάρτησης από την ακρίβεια στους υπολογισμούς.

Χωρίς το fusion των επιπέδων, το μέγεθος παρουσιάζει μια πολύ μικρή απόκλιση στα 24 MB, αλλά ο μέσος χρόνος εκτέλεσης αυξάνεται ελαφρώς σε 40.19 ms (0.4%). Η ακρίβεια μειώνεται περαιτέρω κατά 2.29% σε σχέση με το baseline. Αυτό συμβαίνει επειδή τα bottleneck blocks επηρεάζονται από την παράλειψη της συγχώνευσης, που επιβαρύνει τους επιπλέον υπολογισμούς και την ακρίβεια στις συμπιεσμένες αναπαραστάσεις.

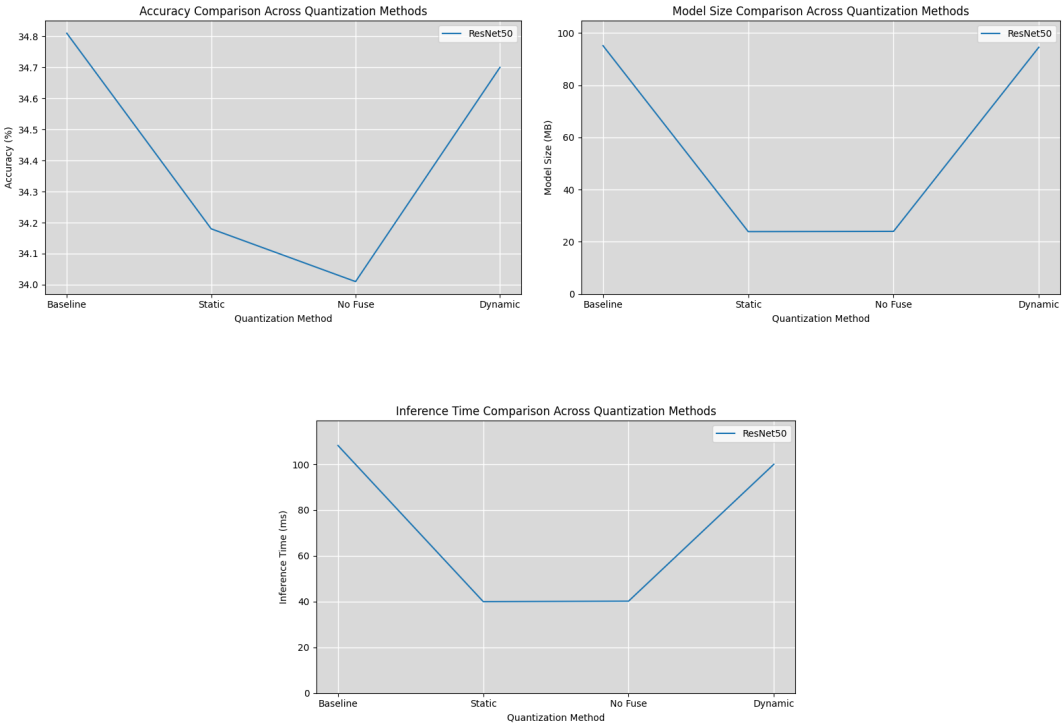
Με τη δυναμική χβαντοποίηση, το μέγεθος του μοντέλου παραμένει κοντά στο baseline στα 94.54 MB, καθώς οι ενεργοποιήσεις δεν χβαντοποιούνται προκαταβολικά, αλλά μετατρέπονται κατά την εκτέλεση. Αυτό επιτρέπει στο ResNet50 να διατηρήσει την ακρίβειά του κοντά στο

baseline με πτώση μόλις 0.2%, ενώ ο χρόνος εκτέλεσης βελτιώνεται ελαφρώς κατά 7.5%.

MobileNetV2

	Model Size (MB)	Model Size Difference (%)	Accuracy (%)	Accuracy Difference (%)	Elapsed Time (ms)	Time Difference (%)
Baseline (FP32)	9.63	-	61.77%	-	96.90	-
Static Quantization	2.47	-74.35%	60.93%	-1.35%	50.06	-48.33%
Static (No-Fuse)	2.76	-71.31%	54.88%	-11.15%	60.50	-37.55%
Dynamic Quantization	9.25	-3.97%	61.6%	-0.27%	107.35	+10.79%

Πίνακας 4.5: Αποτελέσματα MobileNetV2



Σχήμα 4.5: Σύγκριση στην απόδοση το μέγεθος μοντέλου και τον χρόνο inference ανά μέθοδο χβαντοποίησης για το MobileNetV2

Το MobileNetV2 είναι ένα μοντέλο που σχεδιάστηκε ειδικά για να είναι ελαφρύ και αποδοτικό, επιτυγχάνοντας υψηλή ακρίβεια σε περιορισμένα υπολογιστικά περιβάλλοντα. Χρησιμοποιεί αρχιτεκτονική με inverted residual blocks και linear bottlenecks, που επιτρέπουν στο μοντέλο να διατηρεί την αποδοτικότητα και τη δυνατότητα επεξεργασίας σε μικρές συσκευές.

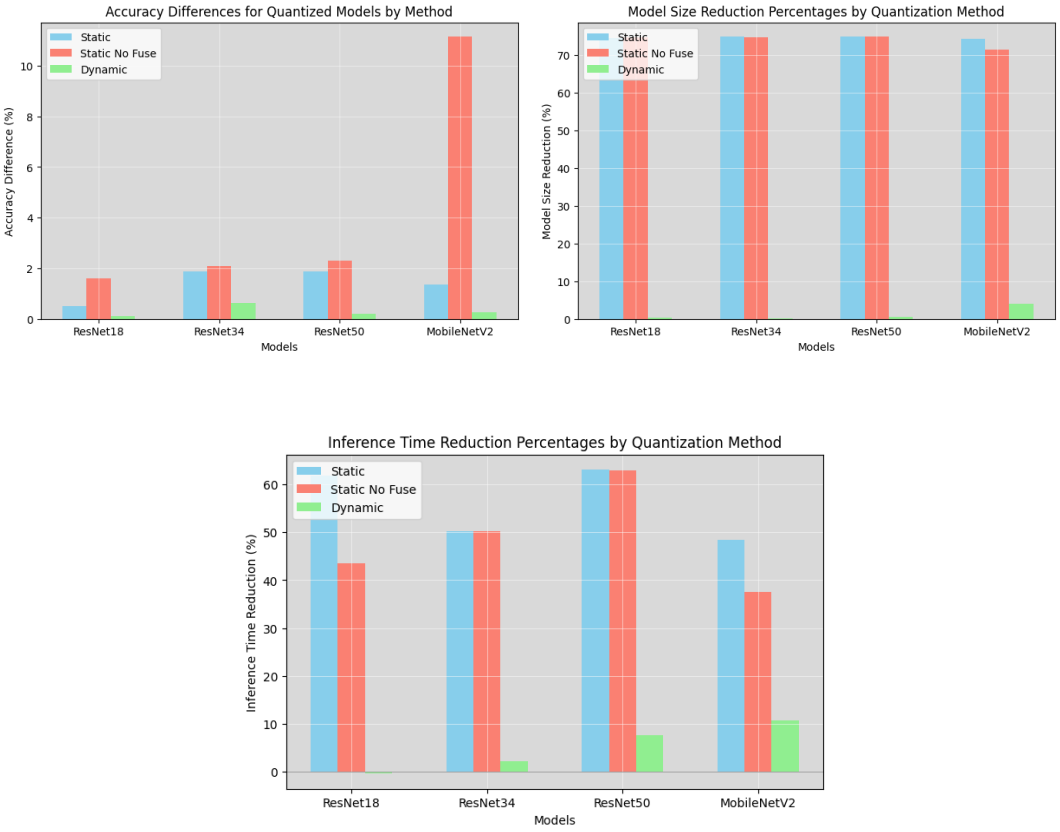
Με τη στατική κβαντοποίηση και την εφαρμογή του fusion των επιπέδων, το μέγεθος του μοντέλου μειώνεται σε 2.47 MB, επιτυγχάνοντας μείωση μνήμης περίπου 74% σε σύγκριση με το αρχικό μοντέλο (baseline).Παρατηρείται επιπλέον, ότι ο χρόνος εκτέλεσης μειώνεται σε 50.06 ms, προσφέροντας βελτίωση περίπου 48% σε σύγκριση με το baseline. Ωστόσο, η ακρίβεια μειώνεται κατά 1.35%, γεγονός που μπορεί να αποδοθεί στην απώλεια λεπτομερειών κατά τη διαδικασία κβαντοποίησης.

Στο δεύτερο πείραμα, όπου δεν εφαρμόστηκε module fusion, το μέγεθος του μοντέλου παραμένει ελαφρώς μεγαλύτερο στα 2.76 MB, και ο μέσος χρόνος εκτέλεσης αυξάνεται στα 60.50 ms. Παρατηρείται απότομη πτώση της ακρίβειας, κατά 11.15%.

Με τη δυναμική κβαντοποίηση, το μέγεθος του μοντέλου παραμένει κοντά στο baseline, στα 9.25 MB, καθώς οι ενεργοποιήσεις υπολογίζονται κατά την εκτέλεση. Αυτή η μέθοδος επιτρέπει στο MobileNetV2 να διατηρεί την ακρίβειά του, με πτώση μόλις 0.27%, ωστόσο, η συνολική μνήμη του μοντέλου και ο χρόνος εκτέλεσης δεν μειώνονται αισθητά.

Παρόλο που το MobileNetV2 σχεδιάστηκε αρχικά με γνώμονα την αποτελεσματικότητα, υπάρχουν μερικοί λόγοι για τους οποίους μπορεί να αποδώσει χειρότερα υπό κβαντοποίηση σε σχέση με τις προηγούμενες αρχιτεκτονικές ResNet: Χρησιμοποιεί συνελίξεις σε βάθοςοι οποίες περιλαμβάνουν λιγότερες παραμέτρους και μικρότερες τιμές ενεργοποίησης, γεγονός που μπορεί να οδηγήσει σε υψηλότερο σφάλμα κβαντοποίησης σε αναπαραστάσεις χαμηλής ακριβείας όπως το INT8. Επιπλέον, το MobileNetV2 χρησιμοποιεί τη συνάρτηση ενεργοποίησης ReLU6, η οποία περιορίζει την έξοδο σε ένα εύρος μεταξύ 0 και 6. Αυτό είναι χρήσιμο για τον έλεγχο των τιμών ενεργοποίησης, αλλά στην περίπτωση της κβαντοποίησης είναι ένας περιοριστικός παράγοντας, καθώς οι ενεργοποιήσεις αντιστοιχίζονται σε μειωμένο εύρος, καθιστώντας το πιο επιρρεπές σε σφάλματα. Αυτό μπορεί να έρχεται σε αντίθεση με τα μοντέλα ResNet, τα οποία συνήθως χρησιμοποιούν απλή ReLU, δίνοντας περισσότερο χώρο για κατά προσέγγιση τιμές χωρίς μεγάλο σφάλμα. Επιπλέον, το MobileNetV2 εξαρχής βελτιστοποιήθηκε για λειτουργίες κινητής υποδιαστολής σε υλικό φορητών συσκευών (π.χ. επεξεργαστές ARM με FP16/FP32). Η κβαντοποίηση μιας τέτοιας αρχιτεκτονικής μπορεί να μην αξιοποιεί πλήρως τις βελτιστοποιήσεις που χρησιμοποιούνται σε υπολογισμούς κινητής υποδιαστολής, οδηγώντας σε υψηλότερη απώλεια κβαντοποίησης. Οι αρχιτεκτονικές ResNet, από την άλλη πλευρά, είναι πιο ευέλικτες και τείνουν να γενικεύονται καλύτερα σε κβαντισμένες και μη κβαντισμένες υλοποιήσεις. Τέλος, αξίζει να σημειωθεί πως τα αποτελέσματα κβαντοποίησης μπορεί επίσης να διαφέρουν με βάση το σύνολο δεδομένων που χρησιμοποιείται, τις παραμέτρους κβαντοποίησης και τα δεδομένα βαθμονόμησης. Το MobileNetV2 μπορεί να απαιτεί λεπτομέρεια ή layer-wise quantization (όπου διαφορετικά επίπεδα χρησιμοποιούν διαφορετικά επίπεδα ακρίβειας) για να επιτευχθεί βέλτιστη ακρίβεια μετά την κβαντοποίηση.

Στα σχήματα που ακολουθούν βλέπουμε και συγκριτικά τις διαφορές σε Model Size, Accuracy και Inference Time για τα μοντέλα CNN που χρησιμοποιήθηκαν.



Σχήμα 4.6: Σύγκριση στην απόδοση το μέγεθος μοντέλου και τον χρόνο inference ανά μέθοδο χβαντοποίησης για τα μοντέλα CNN

BERT Dynamic Quantization on SQuAD

	Baseline (FP32)	Dynamic Quantization	%Difference
Size(MB)	410.99	168.04	-59.11%
Inference Time (ms)	374.86	238.37	-36.41%
Accuracy (%)	36.5%	35.7%	-2.19%

Πίνακας 4.6: Αποτελέσματα BERT

Το μοντέλο BERT στην αρχική του μορφή με floating-point (FP32) έχει μέγεθος 410.99 MB και χρόνο inference 374.86 ms, και ακρίβειά 36.50%, ωστόσο παρουσιάζει υψηλές απαιτήσεις πόρων. Μετά την εφαρμογή δυναμικής χβαντοποίησης INT8, το μέγεθος του μοντέλου μειώνεται σημαντικά στα 168.04 MB, δείχνοντας μείωση περίπου 59% στο αποτύπωμα μνήμης.

Όσον αφορά το χρόνο εκτέλεσης, το κβαντοποιημένο μοντέλο παρουσιάζει σημαντική βελτίωση, με τον μέσο χρόνο να μειώνεται σε 238.77 ms, δηλαδή βελτίωση περίπου 36%. Ωστόσο, αυτό συνοδεύεται από μικρής τάξης πτώση στην ακρίβεια της τάξης του 2%. Η πτώση στην ακρίβεια είναι σχετικά μικρή και αποδεικνύει ότι η διαδικασία κβαντοποίησης, ενώ μειώνει την ακρίβεια από 32-bit σε 8-bit, έχει διατηρήσει την ικανότητα του μοντέλου να λειτουργεί αποτελεσματικά για πρακτικές εφαρμογές.

Αξίζει να σημειωθεί πως τα αποτελέσματα για τη δυναμική κβαντοποίηση σε αρχιτεκτονικές συνελικτικών νευρωνικών δικτύων (CNN) δεν δείχνουν την ίδια αποτελεσματικότητα όπως παρατηρείται με το μοντέλο BERT. Για παράδειγμα, η εφαρμογή δυναμικής κβαντοποίησης σε αρχιτεκτονικές CNN, όπως το ResNet, δεν παρήγαγε σημαντικές βελτιώσεις στο αποτύπωμα, το χρόνο εξαγωγής αποτελεσμάτων ή τη μνήμη. Αντιθέτως, τα γλωσσικά μοντέλα όπως το BERT είναι εγγενώς πιο ανθεκτικά σε τέτοιες επιδράσεις κβαντοποίησης λόγω της εξάρτησής τους από τους μηχανισμούς προσοχής, οι οποίοι τους επιτρέπουν να συλλαμβάνουν αποτελεσματικά τις σχέσεις στα δεδομένα. Αυτή η ανθεκτικότητα επιτρέπει στο BERT να προσαρμόζεται καλύτερα στην μειωμένη ακρίβεια του INT8 χωρίς σημαντική υποβάθμιση της απόδοσης.

Κεφάλαιο 5

Σύνοψη

5.1 Συμπερασματα

Στην παρούσα διπλωματική εργασία, πραγματοποιήθηκε έρευνα για την βελτιστοποίηση νευρωνικών δικτύων με χρήση της κβαντοποίησης για τη μείωση του αποτυπώματος σε χρόνο και μνήμη. Χρησιμοποιήθηκε η μέθοδος της κβαντοποίησης μετά την εκπαίδευση και πιο συγκεκριμένα στατική και δυναμική κβαντοποίηση. Στα συνελικτικά νευρωνικά δίκτυα (CNN), όπως αναλύθηκε, κυριαρχούν οι πράξεις συνέλιξης και τα πλήρως συνδεδεμένα επίπεδα, επομένως οι ενεργοποιήσεις τείνουν να ακολουθούν πιο προβλέψιμες κατανομές στους χάρτες χαρακτηριστικών. Όπως επιβεβαιώθηκε και πειραματικά, η στατική κβαντοποίηση λειτουργεί καλύτερα σε αυτή την περίπτωση επειδή η κατανομή των δεδομένων εισόδου είναι σχετικά σταθερή και προβλέψιμη και οι προ-υπολογισμένες παράμετροι κβαντοποίησης που προκύπτουν από το στάδιο του calibration μπορούν να χρησιμοποιηθούν χωρίς απώλεια σημαντικής ακρίβειας.

Μεταξύ των αρχιτεκτονικών που χρησιμοποιήθηκαν, παρατηρήσαμε ότι τα πιο μικρά μοντέλα ResNet, όπως το ResNet18, παρουσίασαν καλύτερη απόδοση σε σχέση με το MobileNetV2. Αυτό οφείλεται στη σχεδίαση των ResNet, η οποία περιλαμβάνει πιο σύνθετες και βαθιές αρχιτεκτονικές με residual blocks, που βοηθούν στην αποφυγή συσσωρευμένων σφαλμάτων κβαντοποίησης. Αντίθετα, το MobileNetV2, αν και είναι σχεδιασμένο με λιγότερες παραμέτρους και γραμμικά bottlenecks για βελτιστοποίηση ανάπτυξης σε κινητές συσκευές, φάνηκε να είναι πιο ευαίσθητο στην απώλεια ακρίβειας που προκαλείται από την κβαντοποίηση, καθώς οι περιορισμοί στην αναπαράσταση των αριθμών στις χαμηλότερες ακρίβειες μπορεί να περιορίσουν την ικανότητα του μοντέλου να αποτυπώσει τις διαφοροποιήσεις των χαρακτηριστικών. Σε γενικές γραμμές, η πιο περίπλοκη δομή των ResNet τους παρέχει πλεονέκτημα στην κατανόηση και εκπαίδευση πάνω στις πληροφορίες, γεγονός που τα καθιστά πιο ανθεκτικά στις τεχνικές κβαντοποίησης, με αποτέλεσμα να επιτυγχάνεται καλύτερη ακρίβεια και συνολική απόδοση. Όπως παρατηρήθηκε, σε όλα τα μοντέλα, η παράλειψη του module fusion επηρεάζει τη βελτιστοποίηση του υπολογιστικού κόστους, καθώς δεν επιτρέπει τη συγχώνευση ορισμένων επιπέδων (όπως convolutions με Batch Normalization), κάτι που θα βελτίωνε την απόδοση και κυρίως τον χρόνο εκτέλεσης, καθώς με τη συγχώνευση απαιτούνται λιγότερες αριθμητικές πράξεις. Όσον αφορά τα γλωσσικά μοντέλα, λειτουργούν σε δεδομένα κειμένων,

τα οποία παρουσιάζουν μεγάλη μεταβλητότητα στο μήκος εισόδου όπως και στη σημασιολογία. Η κατανομή των ενεργοποιήσεων σε αυτά τα μοντέλα μπορεί να ποικίλλει σημαντικά μεταξύ διαφορετικών τύπων εισαγωγών κειμένου, καθιστώντας δυσκολότερη τη στατική εφαρμογή των παραμέτρων κβαντοποίησης.

Αποδείχθηκε και πειραματικά πως η δυναμική κβαντοποίηση καθίσταται κατάλληλη για γλωσσικά μοντέλα επειδή προσαρμόζεται στην υψηλή μεταβλητότητα των δεδομένων εισόδου και των ενεργοποιήσεων. Συνολικά από το πείραμα και όλες τις περιπτώσεις που εξετάστηκαν, αποδείχτηκε πόσο ισχυρή μέθοδος βελτιστοποίησης είναι η κβαντοποίηση. Παρατηρήθηκε μείωση του χρόνου εξαγωγής αποτελεσμάτων έως και 65% σε κάποια μοντέλα και μείωση του μεγέθους των μοντέλων έως και 75% με μικρή απώλεια ακρίβειας, γεγονός που καθιστά πολύ χρήσιμη την ανάπτυξη και επέκτασή της.

5.2 Μελλοντικές Επεκτάσεις

Η κβαντοποίηση αποτελεί έναν από τους πιο χρήσιμους τομείς έρευνας και ανάπτυξης για τη βελτιστοποίηση νευρωνικών δικτύων, ειδικά όσον αφορά τη μείωση του υπολογιστικού κόστους και της μνήμης που απαιτείται για την εκτέλεση μοντέλων βαθιάς μάθησης. Με την αυξανόμενη ανάγκη για γρήγορες και αποδοτικές εφαρμογές, οι μελλοντικές επεκτάσεις της κβαντοποίησης στοχεύουν στην περαιτέρω βελτίωση της απόδοσης χωρίς σημαντικές απώλειες ακρίβειας. Μερικές από τις πλέον υποσχόμενες κατευθύνσεις είναι η χρήση πιο ακραίας κβαντοποίησης, σε 4-bit (INT4) και τα δυαδικά νευρωνικά δίκτυα (Binary Neural Networks, BNNs). Στην περίπτωση της κβαντοποίησης INT4, τα βάρη και οι ενεργοποιήσεις των μοντέλων αντιπροσωπεύονται με ακρίβεια 4-bit, μειώνοντας ακόμα περισσότερο τον απαιτούμενο αποθηκευτικό χώρο και αυξάνοντας την ταχύτητα επεξεργασίας. Αυτή η προσέγγιση μπορεί να εφαρμοστεί ιδιαίτερα πάνω στα CNNs, τα οποία λόγω της φύσης της αρχιτεκτονικής τους με τα συνελικτικά φίλτρα, απαιτούν μεγάλο όγκο υπολογισμών. Παράλληλα, η ανάπτυξη των δυαδικών νευρωνικών δικτύων (BNNs) τα τελευταία χρόνια παρουσιάζει ιδιαίτερο ενδιαφέρον, καθώς σε αυτά τα μοντέλα χρησιμοποιούνται μόνο δύο επίπεδα τιμών (συνήθως -1 και 1) για την αναπαράσταση των βαρών. Αυτό έχει ως αποτέλεσμα ακόμα μεγαλύτερη μείωση της χρήσης μνήμης και των υπολογιστικών απαιτήσεων. Τα BNNs έχουν αρχίσει να εφαρμόζονται σε ελαφριά μοντέλα CNN, προσφέροντας ταχύτητα και εξοικονόμηση πόρων, αλλά το κύριο πρόβλημα που αντιμετωπίζουν είναι η σχετική πτώση στην ακρίβεια συγκριτικά με τα μοντέλα υψηλότερης ακρίβειας. Όσον αφορά τα Large Language Models (LLMs), οι μελλοντικές επεκτάσεις της κβαντοποίησης επικεντρώνονται σε υβριδικές μεθόδους που προσαρμόζουν δυναμικά την ακρίβεια των δεδομένων ανάλογα με την πολυπλοκότητα της ακολουθίας. Καθώς τα LLMs είναι πιο ευαίσθητα σε αλλαγές ακριβείας, καθίσταται σημαντικό να βρεθεί ισορροπία μεταξύ της ταχύτητας και της διατήρησης των πληροφοριών. Στο μέλλον, αναμένεται η ενσωμάτωση πιο δυναμικών συστημάτων κβαντοποίησης, όπου τα μοντέλα θα μπορούν να αποφασίζουν αυτόματα τον αριθμό των bits που απαιτούνται για κάθε επίπεδο

βάθους, φιλτράρισμα ή ενεργοποίηση. Επίσης, η βελτίωση του hardware, όπως οι μονάδες GPU και TPU, θα επιτρέψουν την καλύτερη υποστήριξη των κβαντισμένων μοντέλων και των εφαρμογών τους σε πραγματικό χρόνο.

Βιβλιογραφία

[1] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y. Binarized neural networks. In *Advances in Neural Information Processing Systems*, 4114-4122, 2016.

[2] Han, S., Mao, H., Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[3] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Kalenichenko, D Adam, H. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 2704-2713), 2018.

[4] Mundichipparakkal, J.: Evaluation of Quantization Methods for Neural Networks, 22 Dec 2022.

[5] Wu, S., Zhang, G., Huang, Y. Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*, 2020.

[6] Choi, Y., El-Khamy, M., Lee, J. Learning low precision deep neural networks through regularization In *In International Conference on Learning Representations.*, 2018.

[7] Banner, R., Nahshan, Y., Hoffer, E., Soudry, D. Post-training 4-bit quantization of convolutional networks for rapid-deployment In *In Advances in Neural Information Processing Systems.*, 2019.

[8] Shen, S., Dong, Z., Ye, J., Ma, L., Huang, Z., Li, J. Q-BERT: Hessian based ultra low precision quantization of BERT. In *Proceedings of the AAAI Conference on Artificial Intelligence.* (Vol. 34, No. 05, pp. 8815-8821), 2020.

[9] Wang, K., Du, Z., Wang, J., Zhang, D., Wen, H. Hessian-Aware . Quantization of Deep Neural Networks with Mixed-Precision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 6128-6136), 2020.

[10] Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M., Keutzer, K. A Survey of Quantization Methods for Efficient Neural Network Inference *arXiv preprint arXiv:2103.13630*, 2021

[11] Stock, P., Joulin, A. Reducing the Computational Cost of Deep Generative Models with Quantization and Pruning *arXiv preprint arXiv:2006.13721.*, 2020

[12] Krishnamoorthi, R. Quantizing deep convolutional networks for efficient inference: A whitepaper *arXiv preprint arXiv:1806.08342*, 2018

[13] Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J. Quantized convolutional neural networks for mobile devices. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016

[14] Choukroun, Y., Kravchik, E., Kisilev, P. Low-bit quantization of neural networks for efficient inference *arXiv preprint arXiv:1902.06822*, 2019

[15] Jain, S., Blalock, D., Guttag, J. Compressing neural networks with probabilistic data structures. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020

[16] Banner, R., Hubara, I., Hoffer, E., Soudry, D. Scalable methods for 8-bit training of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018

[17] He, K., Zhang, X., Ren, S., Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016

[18] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018

[19] Krizhevsky, A. Learning multiple layers of features from tiny images. Technical Report, University of Toronto. , 2009

[20] Rajpurkar, P., Jia, R., Liang, P. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018

[21] Bhandare, A., Baskar, M. K., Chinthakindi, A., Menon, V., Gysel, P. Efficient 8-bit quantization of transformer neural machine language translation model. *arXiv preprint arXiv:1906.00532*, 2019

[22] Zhou, A., Yao, A., Guo, Y., Xu, L., Chen, Y. Incremental network quantization: Towards lossless CNNs with low-precision weights. *arXiv preprint arXiv:1702.03044*, 2016

[23] Nagel, M., Amjad, R. A., Blankevoort, T. Up or down? Adaptive rounding for post-training quantization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021

[24] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, Daniel Soudry Accurate Post Training Quantization With Small Calibration Sets. In *Proceedings of the 38th International Conference on Machine Learning*, PMLR 139:4466-4475, 2021

[25] S. Haykin Νευρωνικα Δικτυα και Μηχανικη Μαθηση. Παπασωτηριου, 2010

[26] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR, abs/1810.04805*, 2018.

[27] M Nagel, M Fournarakis, RA Amjad, Y Bondarenko, M Van Baalen, T Blankevoort A White Paper on Neural Network Quantization *arXiv preprint arXiv:2106.08295*, 2021

Γλωσσάριο

Ελληνικός όρος	Αγγλικός όρος
ακέραιος	integer
ακρίβεια	accuracy
αναπαράσταση	represantation
αρχιτεκτονική	architecture
βάρη	weights
δυναμική	dynamic
έλεγχος	test
εκπαίδευση	training
ενεργοποίηση	activation
εξαγωγή αποτελεσμάτων	inference
κλάση	class
κβαντοποίηση	quantization
κινητή υποδιαστολή	floating point
μετασχηματιστής	transformer
μεταφορά μάθησης	transfer learning
μηχανική μάθηση	machine learning
νευρωνικό δίκτυο	neural network
παράμετροι	parameters
στατική	static
συνέλιξη	convolution
σύνολο δεδομένων	dataset