ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Μελέτη Μεθόδων Βελτιστοποίησης και Αξιολόγηση Συμβατότητας Μοντέλων Μετασχηματιστών σε Κινητές Συσκευές

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΟΚΚΙΝΑΚΗ ΠΑΝΑΓΙΩΤΗ

Επιβλέπων: Ιάκωβος Στ. Βενιέρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2023

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Μελέτη Μεθόδων Βελτιστοποίησης και Αξιολόγηση Συμβατότητας Μοντέλων Μετασχηματιστών σε Κινητές Συσκευές

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΟΚΚΙΝΑΚΗ ΠΑΝΑΓΙΩΤΗ

Επιβλέπων: Ιάκωβος Στ. Βενιέρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26η Οκτωβρίου 2023.

.....	.....	.....
Ιάκωβος Στ. Βενιέρης Καθηγητής Ε.Μ.Π.	Δήμητρα-Θεοδώρα Κακλαμάνη Καθηγήτρια Ε.Μ.Π.	Αθανάσιος Δ. Παναγόπουλος Καθηγητής Ε.Μ.Π.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

(Υπογραφή)

.....

Κοκκινάκης Παναγιώτης

26η Οκτωβρίου 2023

Περίληψη

Τα μοντέλα Βαθιάς Μάθησης χρησιμοποιούνται πλέον σε πληθώρα εφαρμογών σε κάθε τομέα και γίνονται όλο και πιο διαδεδομένα με την πάροδο του χρόνου. Έτσι έχει δημιουργηθεί η ανάγκη να μπορούν τα μοντέλα αυτά να εκτελεστούν τοπικά σε φορητές κινητές συσκευές, ώστε να αποφεύγεται η μεταφορά των δεδομένων σε απομακρυσμένους εξυπηρετητές, γεγονός που αυξάνει την καθυστέρηση και εγείρει κινδύνους ασφαλείας. Τα σύγχρονα μοντέλα όμως χαρακτηρίζονται από μεγάλο μέγεθος και από υπολογιστική περιπλοκότητα, επομένως αποτελεί πρόκληση το να μπορούν να ενσωματωθούν αποτελεσματικά σε κινητές συσκευές, των οποίων οι δυνατότητες είναι περιορισμένες. Σκοπός της παρούσας διπλωματικής εργασίας είναι η μελέτη της τρέχουσας κατάστασης γύρω από την εκτέλεση μοντέλων Μετασχηματιστών σε κινητές συσκευές και η αξιολόγηση μεθόδων βελτιστοποίησης, για την μείωση του χώρου που απαιτείται για την αποθήκευση, του χρόνου εκτέλεσης και για τη συμβατότητα με τους διαθέσιμους επιταχυντές.

Για τις ανάγκες των πειραμάτων που πραγματοποιήθηκαν, εκπαιδεύτηκε ένας αριθμός από μοντέλα Μετασχηματιστών, ειδικευμένων στην Επεξεργασία Φυσικής Γλώσσας για να επιλύσουν το πρόβλημα της Ανάλυσης Συναισθήματος. Εφαρμόστηκαν αντικαταστάσεις και μέθοδοι βελτιστοποίησης, και ελήφθησαν μετρήσεις σχετικά με τον χρόνο και την ακρίβεια για τα αρχικά και τα τροποποιημένα μοντέλα.

Από τις μετρήσεις αυτές, προέκυψε το συμπέρασμα ότι με την αντικατάσταση της συνάρτησης ενεργοποίησης και της μεθόδου Κανονικοποίησης των μοντέλων με απλούστερες, επιτυγχάνεται σημαντική επιτάχυνση του χρόνου εκτέλεσης διατηρώντας ή και βελτιώνοντας την ακρίβεια των αρχικών μοντέλων για τη CPU και τη GPU, ενώ δοκιμάστηκε η διατήρηση του Πολλαπλασιασμού Πινάκων ανά Παρτίδα με ανάμεικτα αποτελέσματα. Οι παρατηρήσεις αυτές θέτουν ένα πλαίσιο για εύκολη βελτιστοποίηση μοντέλων Βαθιάς Μάθησης που μπορεί να επεκταθεί με περαιτέρω έρευνα σε επίπεδο υλικού και μοντέλων.

Λέξεις Κλειδιά

Βαθιά Μάθηση, Μετασχηματιστές, Αλγόριθμοι Συμπίεσης, Αλγόριθμοι Βελτιστοποίησης, Κβαντοποίηση, Συναρτήσεις Ενεργοποίησης, Επεξεργασία Φυσικής Γλώσσας, Κινητές Συσκευές

Abstract

Deep Learning Models are used in a variety of applications in every aspect and are becoming more widespread as time passes. The need has arisen for these models to be executable locally in portable mobile devices, in order to avoid unnecessary data transfers to remote servers, which increases latency and raises security risks. Modern Machine Learning Models are typically large in size and computationally complex making it challenging to effectively incorporate them in mobile devices, whose capabilities are limited. The aim of this thesis is to study the current state of Transformer models inference in mobile devices and evaluate optimization methods in order to decrease the storage space required and the execution latency, and to make them compatible with the available accelerators.

For the purposes of the experiments carried out, a number of Transformer Models, specialized in Natural Language Processing, were trained to solve the problem of Sentiment Analysis. Optimization methods and replacements were applied and measurements were taken regarding the time and accuracy of the original and modified models.

It was concluded from said measurements, that replacing the models' activation function and normalization method with simpler ones, achieves significant speedup in execution time while simultaneously preserving and sometimes even improving model accuracy for CPUs and GPUs. Batch Matrix Multiplication Folding was also considered with mixed results. The observations made on this paper can set a framework for easily implementable Transformer optimizations, which can be expanded through further research in the model and hardware levels.

Keywords

Deep Learning, Transformers, Compression Algorithms, Optimization Algorithms, Quantization, Activation Functions, Natural Language Processing, Mobile Devices

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή της παρούσας διπλωματικής εργασίας κ. Ιάκωβο Στ. Βενιέρη, που μου έδωσε την ευκαιρία να αναλάβω το συγκεκριμένο θέμα και που μου έδωσε το έναυσμα να γνωρίσω και να ασχοληθώ με τα πολύ ενδιαφέροντα πεδία του Κινητού Υπολογισμού και της Βαθιάς Μάθησης. Επίσης, θα ήθελα να ευχαριστήσω την καθηγήτρια κ. Δήμητρα-Θεοδώρα Κακλαμάνη και τον καθηγητή κ. Αθανάσιο Παναγόπουλο, για τη συμμετοχή τους στην τριμελή επιτροπή.

Ιδιαίτερες ευχαριστίες θα ήθελα να απευθύνω στον υποψήφιο διδάκτορα της ΣΗΜΜΥ κ. Ιωάννη Πανόπουλο, για την αμέριστη στήριξη, το ενδιαφέρον και την βοήθεια του. Η ενασχόληση του σε κάθε στάδιο της εκπόνησης της διπλωματικής αυτής εργασίας, από τη διαμόρφωση του θέματος μέχρι την υλοποίηση, υπήρξε καθοριστική.

Τέλος, ευχαριστώ την οικογένεια μου για τη στήριξη και την αγάπη τους, αλλά και τους φίλους και τα κοντινά μου πρόσωπα, που ήταν στο πλευρό μου όλα αυτά τα χρόνια.

Αθήνα, Οκτώβριος 2023

Κοκκινάκης Παναγιώτης

Πίνακας περιεχομένων

Κατάλογος Πινάκων 11
Κατάλογος Σχημάτων 14
1 Εισαγωγή 16
2 Βαθιά Μάθηση και Μετασχηματιστές 18
2.1 Βασικές Αρχές Βαθιάς Μάθησης 19
2.1.1 Perceptron 19
2.1.2 Νευρωνικά Δίκτυα 20
2.1.3 Επίπεδα 22
2.1.4 Συναρτήσεις Ενεργοποίησης 23
2.1.5 Εφαρμογές 29
2.2 Επεξεργασία Φυσικής Γλώσσας 30
2.2.1 Εφαρμογές της Επεξεργασίας Φυσικής Γλώσσας 30
2.2.1 Σύνολα Δεδομένων και Benchmarks 33
2.3 Αρχιτεκτονικές Μοντέλων Επεξεργασίας Φυσικής Γλώσσας 35
2.3.1 Επαναλαμβανόμενα Νευρωνικά Δίκτυα 35
2.3.2 Μετασχηματιστές 37
3 Κινητές Συσκευές, Περιορισμοί και Βελτιστοποίηση 41
3.1 Περιορισμοί και Προκλήσεις Εφαρμογών Βαθιά Μάθησης 41
3.1.1 Περιορισμοί Υλικού των Κινητών Συσκευών 41
3.1.2 Προκλήσεις στην Ανάπτυξη DL Μοντέλων σε Κινητά 42
3.2 Βελτιστοποιήσεις και Σχετικές Εργασίες 43
3.2.1 Βελτιστοποιήσεις Συνελικτικών Νευρωνικών Δικτύων 44
3.2.2 Βελτιστοποιήσεις Μετασχηματιστών 45
4 Προτεινόμενες Βελτιστοποιήσεις 46
4.1 Μέθοδοι Συμπίεσης 46
4.2 Μέθοδοι Βελτιστοποίησης για Συμβατότητα 47
5 Πειραματική Διάταξη 51

5.1 Τεχνολογίες 51
5.1.1 TensorFlow και TensorFlow Lite 51
5.1.2 Hugging Face 52
5.1.3 Android Studio 52
5.2 Αρχιτεκτονικές Transformer Μοντέλων 52
5.2.1 BERT 52
5.2.2 MobileBERT 53
5.2.3 ELECTRA 54
5.2.4 DistilBERT 55
5.2.5 RoBERTa 56
5.2.6 MiniLM 57
5.2.7 XtremeDistil 58
5.3 Σύνολα Δεδομένων 60
5.4 Μεθοδολογία Εκπαίδευσης και Μετατροπής 61
5.5 Κινητή Συσκευή 62
6 Μετρήσεις και Αποτελέσματα 63
6.1 Μετρικές 63
6.2 Αποτελέσματα 64
6.2.1 Συμβατότητα Επιταχυντών 64
6.2.2 Ακρίβεια 66
6.2.3 Επιτάχυνση Μοντέλων 73
7 Επίλογος 80
7.1 Συμπεράσματα 80
7.2 Μελλοντικές Κατευθύνσεις 81
Αναφορές 83

Κατάλογος Πινάκων

2.1: Benchmarks NLP 34

5.1: Το MobileBERT στο σημείο αναφοράς GLUE 54

5.2: Το Electra στο σημείο αναφοράς GLUE 55

5.3: Το DistilBERT στο σημείο αναφοράς GLUE 56

5.4: Το RoBERTa στο σημείο αναφοράς GLUE 57

5.5: Το MiniLM στο σημείο αναφοράς SQuAD2 και σε tasks του GLUE 57

5.6: Το XtremeDistil σε tasks του σημείου αναφοράς GLUE και στο σημείο αναφοράς SQuADv2 58

5.7: Τα μοντέλα που αναφέρθηκαν και που χρησιμοποιήθηκαν στην παρούσα εργασία 59

5.8: Χαρακτηριστικά συσκευής.....62

6.1: Το μέσο ποσοστό κόμβων που στέλνονται σε επιταχυντές για εκτέλεση (XNNPack, GPU) για όλα τα μοντέλα.....64

6.2: Επιτάχυνση εκτέλεσης των αρχικών μοντέλων για χρήση XNNPack και GPU σε σχέση με τη CPU.....65

6.3: Ακρίβεια και Χαρακτηριστικά των Μοντέλων που χρησιμοποιήθηκαν 66

6.4: Ακρίβεια Βελτιστοποιημένων Μοντέλων με ReLU για εκτέλεση σε GPU P100 66

6.5: Ακρίβεια Βελτιστοποιημένων Μοντέλων με ReLU6 για εκτέλεση σε GPU P100.....67

6.6: Ακρίβεια Βελτιστοποιημένων Μοντέλων με Leaky ReLU για εκτέλεση σε GPU P100 67

6.7: Ακρίβεια Βελτιστοποιημένων Μοντέλων με Sigmoid για εκτέλεση σε GPU P100 67

6.8: Ακρίβεια Βελτιστοποιημένων Μοντέλων με Swish για εκτέλεση σε GPU P100.....67

6.9: Ακρίβεια Βελτιστοποιημένων Μοντέλων με Tanh για εκτέλεση σε GPU P100.....68

6.10: Ακρίβεια των αρχικών και των βελτιστοποιημένων μοντέλων για εκτέλεση στη CPU της κινητής συσκευής.....69

6.11: Ακρίβεια των αρχικών και των βελτιστοποιημένων μοντέλων για εκτέλεση στη GPU της κινητής συσκευής.....70

6.12: Ακρίβεια αρχικών και βελτιστοποιημένων Folded μοντέλων στη CPU της κινητής συσκευής.....71

6.13: Ακρίβεια αρχικών και βελτιστοποιημένων Folded μοντέλων στη GPU της κινητής συσκευής.....72

6.14: Επιτάχυνση των μοντέλων για τα διάφορα σχήματα κβαντοποίησης.....73

6.15 : Επιτάχυνση αρχικών και βελτιστοποιημένων μοντέλων στη CPU της κινητής συσκευής.....75

6.16 : Επιτάχυνση αρχικών και βελτιστοποιημένων μοντέλων στη GPU της κινητής συσκευής.....76

6.17: Επιτάχυνση των Folded Μοντέλων στη CPU της κινητής συσκευή 78

Κατάλογος Σχημάτων

2.1: Η αρχιτεκτονική ενός Perceptron.....19

2.2: Η αρχιτεκτονική ενός MLP με δύο επίπεδα.....21

2.3: Γραφική Παράσταση της Συνάρτησης ReLU.....24

2.4: Γραφική Παράσταση της Συνάρτησης ReLU6.....25

2.5: Γραφική Παράσταση της Συνάρτησης Leaky ReLU.....26

2.6: Γραφική Παράσταση της Συνάρτησης Sigmoid.....26

2.7: Γραφική Παράσταση της Συνάρτησης SiLU.....27

2.8: Γραφική Παράσταση της Συνάρτησης Tanh.....28

2.9: Γραφική Παράσταση της Συνάρτησης GELU 28

2.10: Ένα Επαναλαμβανόμενο Νευρωνικό Δίκτυο.....35

2.11: Η αρχιτεκτονική ενός LSTM.....37

2.12: Η αρχιτεκτονική ενός μοντέλου Μετασχηματιστή.....38

2.13: Ο μηχανισμός Προσοχής ενός Μετασχηματιστή.....39

4.1: Το μοντέλο ELECTRA$_{SMALL}$ με BatchMatMul Folding 50

4.2: Το μοντέλο ELECTRA$_{SMALL}$ με BatchMatMul Unfolding 50

5.1: Η προσέγγιση απόσταξης με τη χρήση ενδιάμεσου δασκάλου.....57

5.2: Κατανομή των συναισθημάτων στο σύνολο δεδομένων.....60

6.1: Απόδοση της CPU στην κινητή συσκευή.....74

6.2: Μέσος όρος επιτάχυνσης για όλα τα μοντέλα ανά συνάρτηση ενεργοποίησης και τρόπο κβαντοποίησης σε CPU 76

6.3: Μέσος όρος επιτάχυνσης για όλα τα μοντέλα ανά συνάρτηση ενεργοποίησης και τρόπο κβαντοποίησης σε GPU 77

6.4: Μέσος όρος επιτάχυνσης για όλα τα Folded μοντέλα ανά συνάρτηση ενεργοποίησης και τρόπο κβαντοποίησης σε CPU 79

Κεφάλαιο 1ο

Εισαγωγή

Τα τελευταία χρόνια έχει υπάρξει τεράστια ανάπτυξη στον κλάδο της Τεχνητής Νοημοσύνης (Artificial Intelligence – AI) ερευνητικά. Τα AI μοντέλα πετυχαίνουν εξαιρετικά υψηλές επιδόσεις (συχνά σε σημείο που ξεπερνάνε τις αντίστοιχες ανθρώπινες), σε πολλούς τομείς, όπως η Ανάλυση Εικόνας, η Επεξεργασία Φυσικής Γλώσσας και άλλες χρήσιμες πτυχές της καθημερινής ζωής. Για το λόγο αυτό, έχει γεννηθεί η ανάγκη τα μοντέλα αυτά να είναι ενσωματώσιμα σε φορητές συσκευές και συσκευές παρυφών, όπως είναι τα έξυπνα τηλέφωνα (smartphones), οι συσκευές Διαδικτύου-των-Πραγμάτων (Internet of Things – IoT), οι φορητοί αισθητήρες, κ.α. Η πρόοδος στον τομέα αυτόν και η αποτελεσματικότητα των μοντέλων οφείλεται στη Βαθιά Μάθηση και στις δυνατότητες της οι οποίες είναι τεράστιες και αυξάνονται συνεχώς όσο εξελίσσεται η έρευνα στον τομέα.

Η εκτέλεση των υπολογιστικών μοντέλων τοπικά, στις παρυφές του δικτύου, έχει αρκετά οφέλη. Αρχικά, τα δεδομένα μπορούν να παραμείνουν στη συσκευή αντί να αποστέλλονται για επεξεργασία σε κάποιον εξυπηρετητή και συνεπώς προστατεύεται η ιδιωτικότητα του χρήστη. Πέραν αυτού, με το να εκτελούνται οι υπολογισμοί στη συσκευή, δεν προστίθεται η επιπλέον καθυστέρηση λόγω αποστολής και λήψης των δεδομένων, ενώ ταυτόχρονα αποφεύγεται η συμφόρηση του δικτύου και χρησιμοποιούνται πιο αποτελεσματικά οι υπολογιστικοί πόροι. Τέλος, δεν απαιτείται σύνδεση στο διαδίκτυο, η οποία μπορεί σε συγκεκριμένες περιπτώσεις να μην είναι εφικτή, ενώ η δυνατότητα εκτέλεσης σε κινητές συσκευές κάνει τις εφαρμογές που αναφέρθηκαν προσβάσιμες σε περισσότερο κόσμο.

Ένα χαρακτηριστικό των σύγχρονων μοντέλων Βαθιάς Μάθησης (Deep Learning – DL), και κυρίως των μοντέλων Μετασχηματιστών (Transformers), οι οποίοι αποτελούν την τελευταία λέξη της τεχνολογίας αυτή τη στιγμή στον τομέα αυτόν, είναι το μεγάλο τους μέγεθος και η απαιτητικότητα σε υπολογιστικούς πόρους, λόγω των παραμέτρων τους και των πράξεων και των συναρτήσεων που χρησιμοποιούν. Τα παραπάνω οδηγούν σε μεγάλες απαιτήσεις σε αποθηκευτικό χώρο, ενώ παράλληλα τα σύγχρονα υπολογιστικά συστήματα (κεντρικές μονάδες επεξεργασίας, κάρτες γραφικών, ειδικευμένοι επιταχυντές), δυσκολεύονται να συμβαδίσουν με τόσο αυξημένο αριθμό πράξεων, τις οποίες μάλιστα συχνά δεν είναι σχεδιασμένα να εκτελούν αποδοτικά. Τα προβλήματα αυτά είναι ακόμη πιο έντονα σε συσκευές παρυφών, καθώς διαθέτουν περιορισμένο αποθηκευτικό χώρο και υπολογιστικούς πόρους, λόγω του μικρού τους μεγέθους και της φορητότητας τους.

Δημιουργείται συνεπώς η ανάγκη για βελτιστοποίηση των μοντέλων Βαθιάς Μάθησης, ώστε να είναι εφικτή η εκτέλεση τους τοπικά στις συσκευές παρυφών. Στην παρούσα διπλωματική εργασία θα μελετηθούν μέθοδοι βελτιστοποίησης Transformers σε επίπεδο μοντέλου, σε συνδυασμό με μεθόδους συμπίεσης, ώστε να δημιουργηθούν μοντέλα, που να είναι καταλληλότερα για εκτέλεση σε κινητές συσκευές. Θα αναλυθεί η υπάρχουσα βιβλιογραφία πάνω σε θέματα συμβατότητας και μεθόδους βελτιστοποίησης και στη συνέχεια θα εφαρμοστούν πάνω σε ένα σύνολο μοντέλων Μετασχηματιστών στο έργο της Επεξεργασίας Φυσικής Γλώσσας. Θα ληφθούν οι απαραίτητες μετρήσεις κατά την εκτέλεση των μοντέλων σε μια κινητή συσκευή μέσω μιας εφαρμογής Android,

ώστε να εξαχθούν συμπεράσματα για την αποτελεσματικότητα και τη βιωσιμότητα των μεθόδων αυτών.

Κεφάλαιο 2ο

Βαθιά Μάθηση και Μετασχηματιστές

Ο όρος Τεχνητή Νοημοσύνη (Artificial Intelligence – AI) αναφέρεται στο πεδίο της επιστήμης υπολογιστών που ασχολείται με τη δημιουργία και τη μελέτη ευφυιών συστημάτων, δηλαδή συστημάτων που έχουν την ικανότητα να μαθαίνουν και να εξελίσσονται ώστε να μπορούν να διεκπεραιώσουν κάποιο έργο. Από την αρχή της ύπαρξης ψηφιακών υπολογιστών τη δεκαετία του 1940, έγινε ξεκάθαρο ότι τα υπολογιστικά συστήματα μπορούν να προγραμματίζονται ώστε να διεκπεραιώνουν περίπλοκες διεργασίες, και ο στόχος πάντοτε ήταν να δημιουργηθεί ένα σύστημα με νοημοσύνη που να συγκρίνεται με αυτή του ανθρώπου. Σήμερα, υπάρχουν αρκετά μοντέλα που ξεπερνούν τις ανθρώπινες επιδόσεις σε εξειδικευμένα προβλήματα, γεγονός που δείχνει την αξιοσημείωτη πρόοδο που έχει σημειωθεί στον τομέα.

Συχνά, υπάρχει σύγχυση σχετικά με τη χρήση του όρου αυτού και του όρου Μηχανική Μάθηση. Η Μηχανική Μάθηση (Machine Learning – ML) είναι το πεδίο μελέτης της Τεχνητής Νοημοσύνης, το οποίο πραγματεύεται τη δημιουργία συστημάτων, τα οποία έχουν την ικανότητα να μαθαίνουν, με παρόμοιο τρόπο με αυτόν που μαθαίνει ο άνθρωπος. Οι αλγόριθμοι Μηχανικής Μάθησης αναλύουν υπάρχοντα δεδομένα και σταδιακά εκπαιδεύονται πάνω σε αυτά, ώστε να μπορούν να γενικεύουν και να επιλύουν προβλήματα. Ο κλάδος της Μηχανικής Μάθησης είναι από τους πλέον αναπτυσσόμενους αυτή τη στιγμή, και χρησιμοποιείται σε πλήθος εφαρμογών.

Στο γενικότερο πεδίο της Μηχανικής Μάθησης, ένα ιδιαίτερα αξιοσημείωτο και ταχέως εξελισσόμενο υποπεδίο είναι η Βαθιά Μάθηση (Deep Learning - DL). Η DL αντιπροσωπεύει έναν εξειδικευμένο τομέα της ML που φέρνει επανάσταση στο τοπίο της τεχνητής νοημοσύνης. Σε αντίθεση με τους παραδοσιακούς αλγόριθμους μηχανικής μάθησης, τα μοντέλα βαθιάς μάθησης έχουν σχεδιαστεί για να προσομοιώνουν τα νευρωνικά δίκτυα του ανθρώπινου εγκεφάλου και υπερέχουν στην επεξεργασία τεράστιων ποσοτήτων δεδομένων για την εξαγωγή περίπλοκων μοτίβων και σχέσεων. Τα τελευταία χρόνια, η βαθιά μάθηση οδήγησε στην ανάπτυξη βαθιών νευρωνικών δικτύων, όπως είναι τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNNs) για ανάλυση εικόνας ή τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks - RNNs) για επεξεργασία σειριακών δεδομένων. Επιπλέον, πιο πρόσφατες καινοτομίες, όπως οι Μετασχηματιστές (Transformers), έχουν πυροδοτήσει σημαντικές προόδους σε διεργασίες επεξεργασίας φυσικής γλώσσας και μηχανικής μετάφρασης [1]–[3] [4], [5].

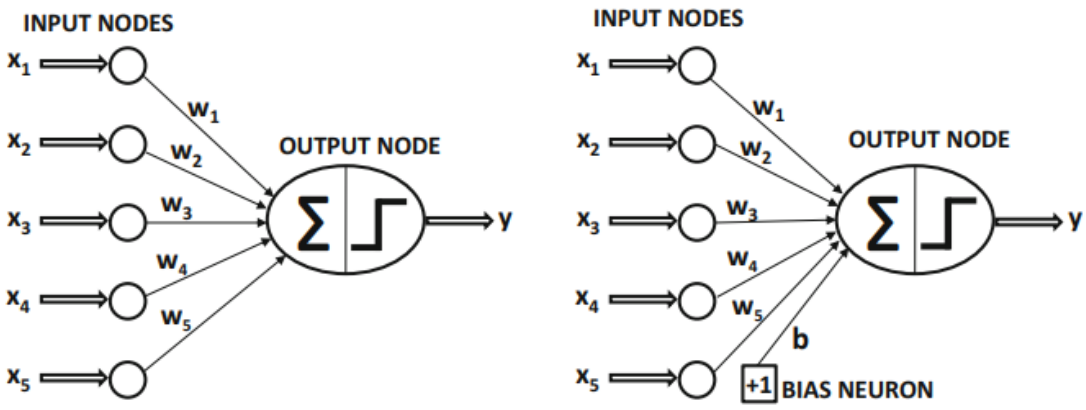
2.1 Βασικές Αρχές Βαθιάς Μάθησης

Η προσέγγιση της Βαθιάς Μάθησης (Deep Learning – DL) βασίζεται στη χρήση Τεχνητών Νευρωνικών Δικτύων για την εκμάθηση από δεδομένα. Αποτελεί υποπεδίο της Μηχανικής Μάθησης και αποπειράται να προσομοιάσει τον τρόπο με τον οποίο ο άνθρωπος επεξεργάζεται ερεθίσματα και τα μετατρέπει σε πληροφορία. Βρίσκει εφαρμογή στους τομείς της Όρασης Υπολογιστών, της Επεξεργασίας Φυσικής Γλώσσας, των Ιατρικών Διαγνώσεων, της Επεξεργασίας Φωνής, κ.α. Η λέξη «βαθύ» στον όρο Βαθιά Μάθηση αναφέρεται στο πλήθος των επιπέδων ή βάθος του Νευρωνικού Δικτύου (Neural Network - NN), το οποίο είναι μεγαλύτερο από 3, συμπεριλαμβανομένων των επιπέδων εισόδου και εξόδου. Τα Βαθιά Νευρωνικά Δίκτυα διαπρέπουν στην εξαγωγή χαρακτηριστικών από δεδομένα, ώστε να μοντελοποιούν περίπλοκες έννοιες και αναπαραστάσεις και έχουν επιτύχει εξαιρετικά υψηλές αποδόσεις σε πολλές διαφορετικές εφαρμογές.

Τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks – ANNs) ή απλά Νευρωνικά Δίκτυα είναι υπολογιστικά μοντέλα, τα οποία βασίζονται στον τρόπο λειτουργίας και μάθησης του ανθρώπινου εγκεφάλου. Η δομική τους μονάδα είναι ο Νευρώνας (Neuron) και οργανώνονται σε επίπεδα τα οποία συνδέονται μεταξύ τους μέσω τεχνητών συνάψεων. Τα Νευρωνικά Δίκτυα προσομοιάζουν ένα απλοποιημένο μοντέλο του τρόπου με τον οποίο επεξεργάζεται πληροφορίες ο άνθρωπος [3].

2.1.1 Perceptron

Στα δίκτυα ενός επιπέδου, ένα σύνολο εισόδων χαρτογραφείται απευθείας σε μία έξοδο χρησιμοποιώντας κάποια μη γραμμική συνάρτηση. Το απλό αυτό υπολογιστικό μοντέλο νευρώνα ονομάζεται Perceptron και αποτελεί το απλούστερο Νευρωνικό Δίκτυο. Στο Σχήμα 2.1 παρουσιάζεται η βασική αρχιτεκτονική του Perceptron.



Σχήμα 2.1: Η αρχιτεκτονική ενός Perceptron, Αριστερά: Ένα Perceptron χωρίς πόλωση, Δεξιά: ένα Perceptron με πόλωση [3]

Το επίπεδο εισόδου ενός Perceptron περιέχει έναν αριθμό από κόμβους εισόδου με βάρη. Αν θεωρήσουμε ότι το διάνυσμα εισόδου είναι το $X^T = [x_1, \dots, x_d]$ και τα βάρη του νευρώνα είναι το διάνυσμα $W^T = [w_1, \dots, w_d]$, όπου d η διάσταση της εισόδου του Perceptron, η έξοδος υπολογίζεται στον κόμβο εξόδου και είναι η εξής:

$$y = sign\,(W^T \cdot X) = sign\left(\sum_{j=1}^{d} w_j x_j\right)$$

Σε πολλές περιπτώσεις, υπάρχει ένα αμετάβλητο μέρος της πρόβλεψης, το οποίο αναφέρεται ως πόλωση ή μεροληψία (bias). Στην περίπτωση αυτή ενσωματώνουμε μία πρόσθετη μεταβλητή b, η οποία αποτελεί το υποκειμενικό όριο διαχωρισμού, και η έξοδος του Perceptron θα είναι η εξής:

$$y = sign(W^T \cdot X + b) = sign\left(\sum_{j=1}^{d} w_j x_j + b\right)$$

Το Perceptron χρησιμοποιεί τη συνάρτηση προσήμου, η οποία αντιστοιχίζει τις εισόδους στις πραγματικές τιμές +1 ή -1, και ενδείκνυται για δυαδική ταξινόμηση. Ο νευρώνας «ενεργοποιείται» όταν η συνάρτηση αυτή λάβει θετικό όρισμα, και επομένως είναι ίση με +1, και για το λόγο αυτό η συνάρτηση ονομάζεται και συνάρτηση ενεργοποίησης (activation function). Τόσο τα βάρη όσο και η τιμή της μεροληψίας καθορίζονται από τα δεδομένα μέσω της διαδικασίας εκμάθησης.

Το Perceptron, λόγω της απλότητας του, έχει περιορισμένες δυνατότητες, και μπορεί να χρησιμοποιηθεί μόνο για προβλήματα τα οποία είναι γραμμικά διαχωρίσιμα. Για πιο σύνθετα προβλήματα χρησιμοποιούνται Νευρωνικά Δίκτυα τα οποία αποτελούνται από περισσότερους Νευρώνες [3].

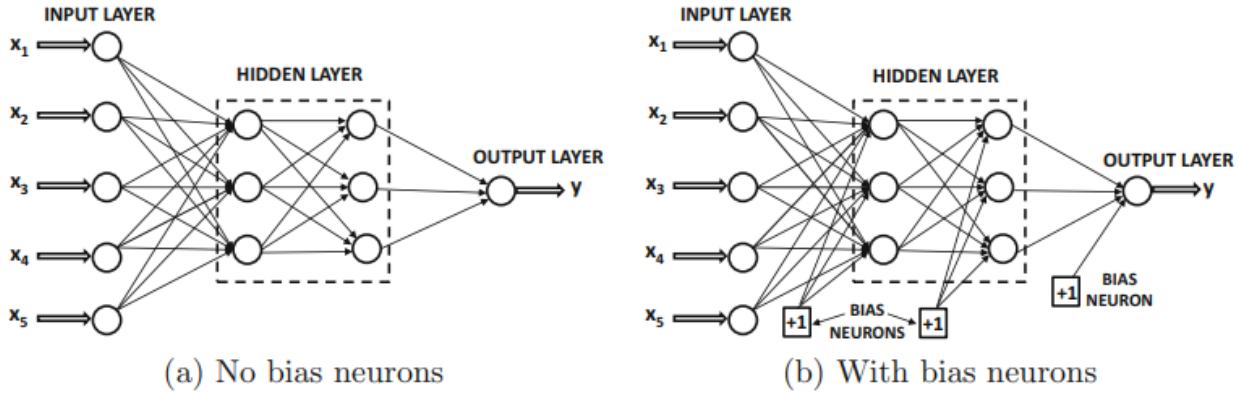
2.1.2 Νευρωνικά Δίκτυα

Όπως αναφέρθηκε προηγουμένως, οι Νευρώνες είναι η δομική μονάδα ενός Νευρωνικού Δικτύου. Οργανώνονται σε στρώματα, και συνδέονται μεταξύ τους μέσω συνάψεων ώστε να προκύψει το υπολογιστικό δίκτυο. Κάθε Νευρωνικό Δίκτυο αποτελείται από τρία μέρη: το επίπεδο εισόδου, το οποίο αποτελείται από τους νευρώνες εισόδου, ένα ή περισσότερα κρυφά επίπεδα, και το επίπεδο εξόδου, που έχει είτε μια ή περισσότερες μονάδες στόχους. Οποιοδήποτε νευρωνικό δίκτυο έχει παραπάνω από ένα κρυφό επίπεδο, ονομάζεται βαθύ Νευρωνικό Δίκτυο (Deep Neural Network - DNN).

Τα δεδομένα εισέρχονται στο επίπεδο εισόδου και οι τιμές τους διαδίδονται στα κρυφά επίπεδα μέσω των συνάψεων. Οι υπολογισμοί που εκτελούνται σε αυτά δεν είναι ορατοί στο χρήστη, ενώ αφού ενεργοποιούνται οι αντίστοιχοι νευρώνες, βάσει της εκάστοτε συνάρτησης ενεργοποίησης, η έξοδος τους μεταδίδεται κάθε φορά στο επόμενο επίπεδο. Τα δίκτυα που χρησιμοποιούν τη συγκεκριμένη αρχιτεκτονική καλούνται και δίκτυα εμπρόσθιας τροφοδότησης (feed-forward networks), επειδή το ένα στρώμα συνδέεται με το επόμενο του, με κατεύθυνση από την είσοδο προς την έξοδο.

Στην έξοδο του δικτύου, υπολογίζεται η συνάρτηση απώλειας (loss function). Η συνάρτηση απώλειας αξιολογεί το βαθμό στον οποίο οι προβλέψεις του μοντέλου ευθυγραμμίζονται με τις πραγματικές τιμές. Χρησιμοποιείται σε συνδυασμό με έναν βελτιστοποιητή (optimizer). Ο optimizer είναι ένας αλγόριθμος βάσει του οποίου ανανεώνονται οι παράμετροι του μοντέλου κατά την εκπαίδευση, ώστε να ελαχιστοποιηθεί η συνάρτηση απώλειας. Η ανανέωση των βαρών των νευρώνων γίνεται μέσω της διαδικασίας της διάδοσης προς τα πίσω (back propagation), ξεκινώντας από τα τελευταία επίπεδα με κατεύθυνση προς την είσοδο. Έτσι το νευρωνικό δίκτυο εκπαιδεύεται και ρυθμίζεται δοσμένου ενός διανύσματος εισόδου $X = [x_1, \dots, x_d]$ διάστασης d, ώστε να μπορεί να κάνει σωστές προβλέψεις και να παράγει μία πρόβλεψη $\hat{y}$ για έναν στόχο y.

Στο σχήμα 2.2 φαίνεται η αρχιτεκτονική ενός Perceptron Πολλών Επιπέδων (Multi Layer Perceptron – MLP) με 5 νευρώνες εισόδου, 2 κρυφά επίπεδα και 1 νευρώνα εξόδου. Τα MLPs είναι τα απλούστερα Νευρωνικά Δίκτυα πολλών επιπέδων [3].



Σχήμα 2.2: Η αρχιτεκτονική ενός MLP με δύο κρυφά επίπεδα,

Αριστερά: MLP χωρίς μεροληψία, Δεξιά: MLP με μεροληψία [3]

2.1.3 Επίπεδα

Όπως αναφέρθηκε τα Νευρωνικά Δίκτυα αποτελούνται από συνδεδεμένα επίπεδα. Αναφέρθηκαν επίσης τα MLPs και τα επίπεδα τους τα οποία ονομάζονται Πλήρως Συνδεδεμένα. Ο τρόπος με τον οποίο οργανώνονται σε επίπεδα οι νευρώνες και που συνδέονται μεταξύ τους, καθορίζει και τη λειτουργία και τις ιδιότητες του νευρωνικού δικτύου, ώστε να προκύπτουν οι διαφορετικοί τύποι μοντέλων που υπάρχουν όπως τα Συνελικτικά Νευρωνικά Δίκτυα, τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα και οι Μετασχηματιστές. Παρακάτω παρατίθενται μερικά σημαντικά επίπεδα, τα οποία είναι δομικές μονάδες των μοντέλων που θα αναλυθούν και παρακάτω:

- **Πλήρως Συνδεδεμένο Επίπεδο (Fully Connected Layer)**: Στα Πλήρως Συνδεδεμένα Επίπεδα, όλοι οι νευρώνες είναι συνδεδεμένοι με κάθε νευρώνα του προηγούμενου επιπέδου. Χρησιμοποιούνται συχνά στα τελευταία επίπεδα ενός Νευρωνικού Δικτύου για κατηγοριοποίηση, επειδή έχουν την ικανότητα να κατατάσσουν δεδομένα.
- **Επίπεδο Συγκέντρωσης (Pooling Layer)**: Τα Επίπεδα Συγκέντρωσης χρησιμοποιούνται για να μειώσουν τις διαστάσεις του διανύσματος εισόδου, διατηρώντας σε ένα βαθμό την πληροφορία που αυτό περιέχει.
- **Επίπεδο Κανονικοποίησης (Normalization Layer)**: Στο Επίπεδο Κανονικοποίησης εφαρμόζεται στα δεδομένα εισόδου κάποια συνάρτηση κανονικοποίησης, ώστε αυτά να κανονικοποιηθούν σε κάποιο επιθυμητό εύρος τιμών με σταθερή κατανομή. Η κανονικοποίηση βελτιώνει την ικανότητα γενίκευσης και είναι χρήσιμη για την αντιμετώπιση των Προβλημάτων Έκρηξης και Εξαφάνισης Κλίσης, που είναι συχνά στα Βαθιά Νευρωνικά Δίκτυα. Επίσης βοηθάει στην ταχύτερη σύγκλιση κατά την εκπαίδευση, η οποία μπορεί να δυσχεραίνεται από την εσωτερική μετατόπιση των συμμεταβλητών.
- **Επίπεδο Συνέλιξης (Convolutional Layer)**: Τα Επίπεδα Συνέλιξης είναι το βασικό δομικό στοιχείο των Συνελικτικών Νευρωνικών Δικτύων. Μιμούνται την ικανότητα του ανθρώπινου οπτικού συστήματος να αντιλαμβάνεται σχήματα και υφές. Στα επίπεδα αυτά εφαρμόζεται η πράξη της Συνέλιξης σε δύο διαστάσεις ανάμεσα στην είσοδο και μία σειρά από φίλτρα μικρότερης διάστασης από αυτήν της εισόδου. Έτσι παράγονται οι χάρτες ενεργοποίησης, οι οποίοι αντιστοιχούν στην απόκριση κάθε φίλτρου για την είσοδο.

Αξίζει να γίνει ειδική αναφορά σε δύο μεθόδους που εφαρμόζονται για το Επίπεδο Κανονικοποίησης και που μελετώνται στην παρούσα εργασία:

- **Κανονικοποίηση Παρτίδας (Batch Normalization)**: Τα επίπεδα Κανονικοποίησης Παρτίδας χρησιμοποιούνται ευρέως στα CNNs, προστίθενται μεταξύ των κρυφών επιπέδων και έχουν σκοπό τη δημιουργία χαρακτηριστικών με παρόμοια διακύμανση ώστε να ενισχύεται η σταθερότητα του δικτύου και να διευκολύνεται η σύγκλιση. Χρησιμοποιούν δύο εκπαιδεύσιμες παραμέτρους, τις β και γ, καθώς και τις μέσες τιμές και διακυμάνσεις m και v. Οι παράμετροι β και γ μαθαίνονται κατά την εκπαίδευση ενώ υπολογίζονται σε κάθε βήμα και οι τιμές των m και v, ώστε να υπάρχει μία εκτίμηση της μέσης τιμής και της διακύμανσης όλων των δεδομένων. Κατά τη συμπερασματολογία, γίνεται η κανονικοποίηση χρησιμοποιώντας τις τιμές που μαθεύτηκαν κατά την εκπαίδευση, οπότε η έξοδος του επιπέδου είναι ένας απλός γραμμικός μετασχηματισμός.

- **Κανονικοποίηση Στρώματος (Layer Normalization)**: Η Κανονικοποίηση Στρώματος χρησιμοποιείται συνήθως όταν το μέγεθος δέσμης είναι μικρό ή ίσο με 1. Στο στρώμα αυτό υπολογίζονται η μέση τιμή μ_i και η τυπική απόκλιση σ_i, για όλα τα χαρακτηριστικά ενός δείγματος κατά μήκος της παρτίδας και βάσει αυτών υπολογίζονται οι νέες κανονικοποιημένες τιμές. Διαφέρει από την Κανονικοποίηση Παρτίδας, επειδή κανονικοποιεί κατά μήκος κάθε δείγματος στην παρτίδα, ενώ η τελευταία κανονικοποιεί κατά μήκος των χαρακτηριστικών για όλη την παρτίδα. Χρησιμοποιείται ευρέως στα Επαναληπτικά Νευρωνικά Δίκτυα και στους Μετασχηματιστές επειδή αφορούν ακολουθιακά δεδομένα[3], [6].

2.1.4 Συναρτήσεις Ενεργοποίησης

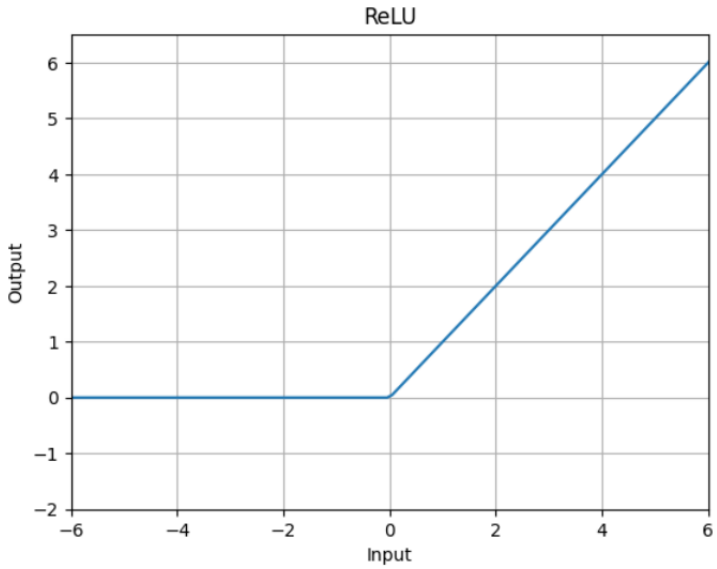
Εξαιρετικά σημαντικός είναι και ο ρόλος των συναρτήσεων ενεργοποίησης των νευρώνων, για τη συνολική λειτουργία του Νευρωνικού Δικτύου. Οι συναρτήσεις αυτές καθορίζουν εάν ο νευρώνας θα ενεργοποιηθεί, και κατ' επέκταση εάν η είσοδος του νευρώνα αυτού είναι σημαντική για την πρόβλεψη του μοντέλου.

Η συνάρτηση προσήμου των Perceptron, μπορεί να είναι χρήσιμη για την ταξινόμηση δυαδικών δεδομένων, όμως για πιο περίπλοκες εργασίες (tasks), απαιτούνται και πιο περίπλοκες συναρτήσεις, με πραγματική, μη γραμμική έξοδο. Οι μη γραμμικές συναρτήσεις ενεργοποίησης έχουν τη δυνατότητα, λόγω της παραγώγου τους, να επιτρέπουν το back propagation, ώστε να ανανεώνονται καταλλήλως τα βάρη. Επιπλέον μπορούν να μάθουν περίπλοκες σχέσεις μεταξύ της εισόδου και της εξόδου, για τις οποίες οι γραμμικές συναρτήσεις δεν είναι επαρκείς, συνδυάζοντας παράλληλα και τις εισόδους πολλών νευρώνων.

Παρακάτω αναλύονται μερικές από τις συχνότερα χρησιμοποιούμενες συναρτήσεις ενεργοποίησης και οι παραλλαγές τους, οι οποίες θα χρησιμοποιηθούν και στο πλαίσιο της εργασίας αυτής:

- **ReLU**

$$f(x) = \max(0, x)$$



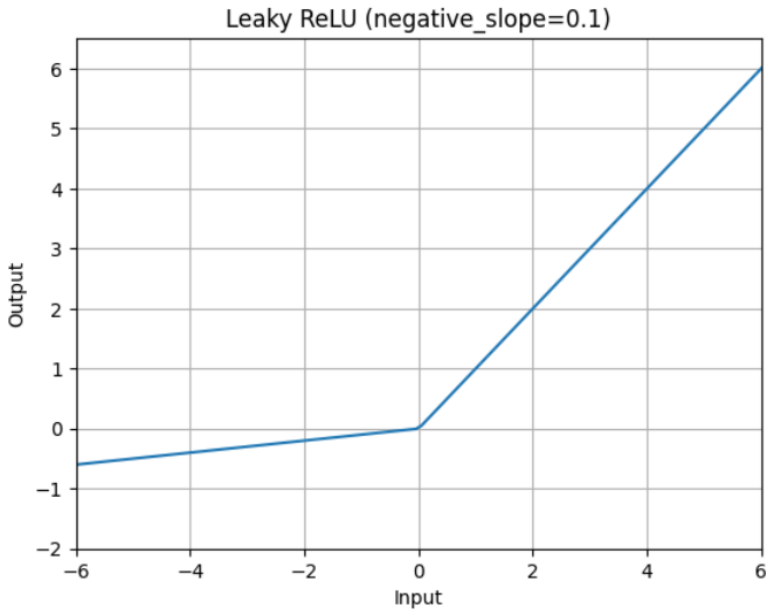
Σχήμα 2.3: Γραφική Παράσταση της συνάρτησης ReLU

Η συνάρτηση ReLU (Rectified Linear Unit), είναι γραμμική για θετικές και μηδενίζει για αρνητικές τιμές. Είναι μη γραμμική και είναι παραγωγίσιμη (σε κάθε σημείο εκτός από το 0), πράγμα που αποτελεί απαραίτητη προϋπόθεση για την λειτουργία του back propagation. Οι νευρώνες ενεργοποιούνται μόνο για θετικές τιμές, γεγονός που την καθιστά υπολογιστικά αποδοτική. Λόγω της απλότητας και της αποτελεσματικότητας της, είναι από τις πλέον χρησιμοποιούμενες.

Επειδή ένα ποσοστό νευρώνων δεν ενεργοποιούνται, και η παράγωγος της συνάρτησης για αρνητικές τιμές είναι ίση με μηδέν, αντιμετωπίζει το Dying ReLU Problem, κατά το οποίο τα βάρη και οι μεροληψίες κάποιων νευρώνων δεν ανανεώνονται. Αυτό έχει ως συνέπεια να μην ενεργοποιούνται ποτέ και να δημιουργούνται «νεκροί» νευρώνες. Για την αντιμετώπιση του προβλήματος αυτού, υπάρχουν παραλλαγές της συνάρτησης ReLU, όπως η Leaky ReLU.

- **Leaky ReLU**

$$f(x) = \max(0.1x, x)$$

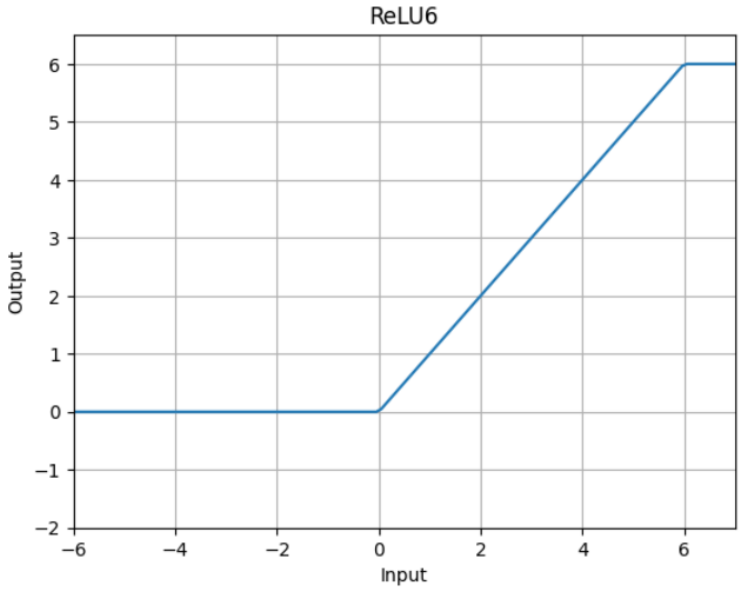


Σχήμα 2.4: Γραφική Παράσταση της συνάρτησης Leaky ReLU

Η Leaky ReLU έχει όλα τα πλεονεκτήματα της ReLU, όμως επιτρέπει το back propagation, ακόμη και για αρνητικές τιμές, αποτρέποντας με αυτόν τον τρόπο την ύπαρξη νεκρών νευρώνων. Παρόλα αυτά οι μικρές τιμές εισόδου μπορεί να δημιουργούν σφάλμα στις προβλέψεις, ενώ εξαιτίας της χαμηλής παραγώγου για αρνητικές τιμές, το μοντέλο αργεί περισσότερο να συγκλίνει σε τοπικό ελάχιστο κατά την εκπαίδευση.

- **ReLU6**

$$f(x) = \min\left(\max(0, x), 6\right)$$

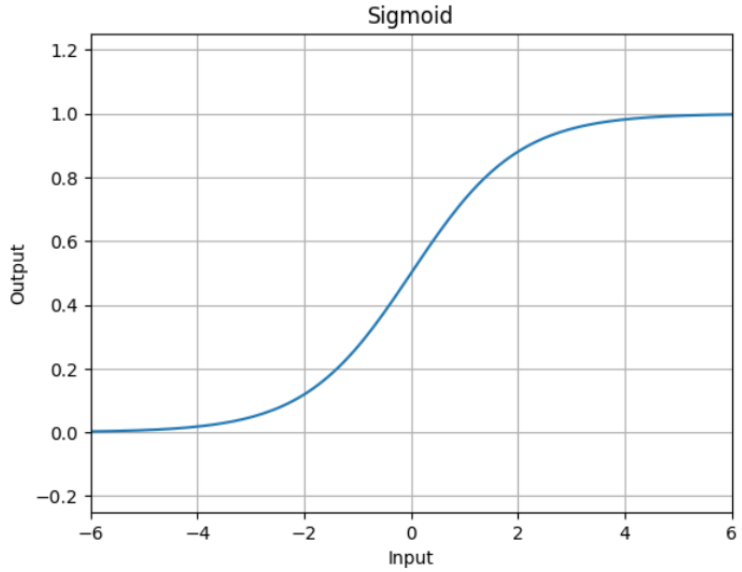


Σχήμα 2.5: Γραφική Παράσταση της συνάρτησης ReLU6

Η συνάρτηση ReLU6, είναι άλλη μία παραλλαγή της απλής ReLU. Η μέγιστη τιμή της εξόδου περιορίζεται στο 6, γεγονός που είναι βοηθητικό καθώς διατηρεί την έξοδο σε ένα μικρό εύρος τιμών αλλά χρησιμεύει και για συστήματα που δεν υποστηρίζουν υπολογισμούς μεγάλων μεταβλητών.

- **Σιγμοειδής**

$$sigmoid(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$

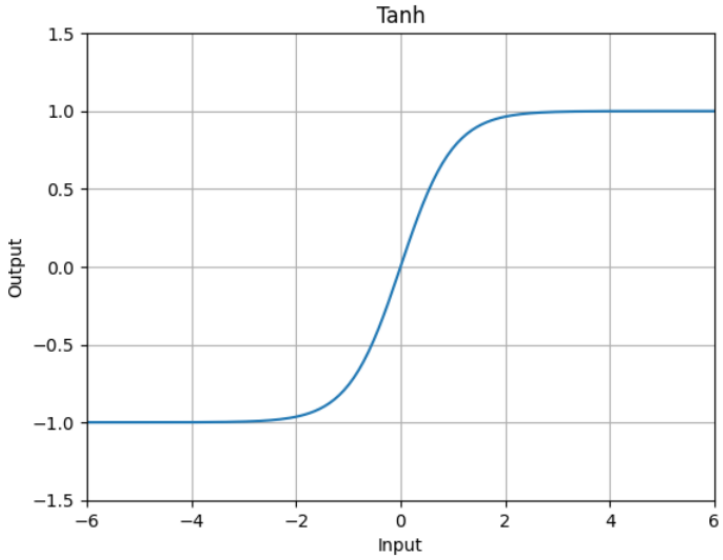


Σχήμα 2.6: Γραφική Παράσταση της συνάρτησης Sigmoid

Η συνάρτηση αυτή έχει ως έξοδο κάποια τιμή από 0 μέχρι 1 για οποιαδήποτε είσοδο. Χρησιμοποιείται συχνά για τον υπολογισμό πιθανοτήτων επειδή το πεδίο τιμών της είναι το [0, 1]. Υποφέρει από το πρόβλημα της Εξαφάνισης Κλίσης (Vanishing Gradient), επειδή η παράγωγος της συνάρτησης έχει πολύ χαμηλές τιμές εκτός του εύρους -3 έως 3.

- **Υπερβολική Εφαπτομένη**

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

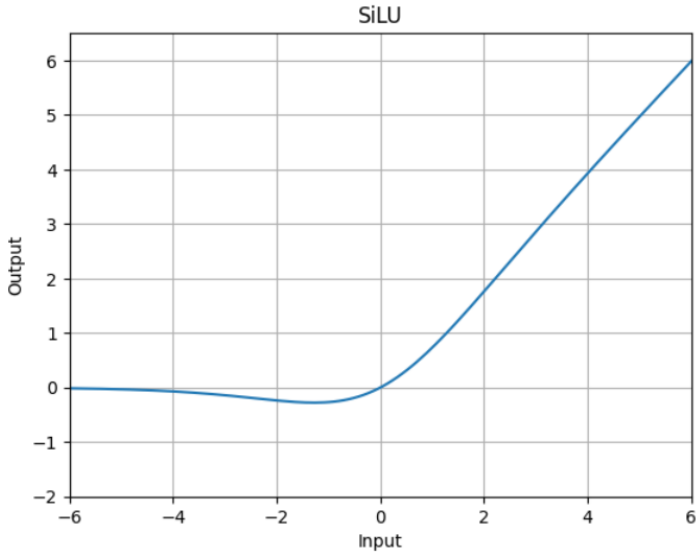


Σχήμα 2.7: Γραφική Παράσταση της συνάρτησης Tanh

Η συνάρτηση Υπερβολικής Εφαπτομένης είναι παρόμοια με τη σιγμοειδή, αλλά έχει πεδίο τιμών από το -1 μέχρι το +1, ενώ η σιγμοειδής έχει πεδίο τιμών από το 0 μέχρι το +1. Είναι κεντραρισμένη γύρω από την αρχή των αξόνων, γεγονός που επιτρέπει την κατηγοριοποίηση των εξόδων σε θετικές, αρνητικές και ουδέτερες. Υποφέρει από το Vanishing Gradient Problem, όπως και η σιγμοειδής.

- **Swish**

$$swish(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}}$$

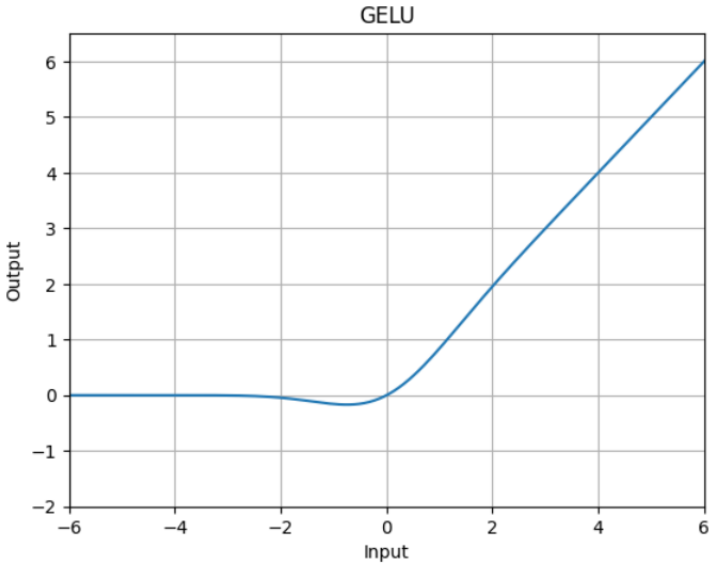


Σχήμα 2.8: Γραφική Παράσταση της συνάρτησης Swish

Η συνάρτηση Swish, γνωστή και ως Sigmoid Linear Unit (SiLU) έχει υψηλές αποδόσεις σε μία σειρά από tasks και αποδίδει καλύτερα από την κλασσική ReLU. Οι μεγάλες σε απόλυτη τιμή αρνητικές είσοδοι, γίνονται μηδέν, ενώ διατηρούνται οι μικρότερες οι οποίες μπορεί να παίζουν σημαντικό ρόλο.

- **Gaussian Error Linear Unit (GELU)**

$$GELU(x) = x \cdot \Phi(x) = x \cdot P(X \leq x) \; \alpha\nu \; X \sim N(0,1)$$



Σχήμα 2.9: Γραφική Παράσταση της συνάρτησης GELU

Η GELU αναπτύχθηκε πρόσφατα και είναι η συνάρτηση που χρησιμοποιείται κυρίως σε Μετασχηματιστές και άλλα μοντέλα Επεξεργασίας Φυσικής Γλώσσας. Αποδίδει καλύτερα από τις προαναφερθείσες συναρτήσεις ενεργοποίησης σε ό,τι task έχει δοκιμαστεί στους τομείς της Όρασης Υπολογιστών, της Αναγνώρισης Φωνής και της Επεξεργασίας Φυσικής Γλώσσας.

Η GELU ζυγίζει με μη γραμμικό τρόπο τις εισόδους βάσει του εκατοστημορίου στο οποίο ανήκουν, κρατώντας όσες περνάνε ένα συγκεκριμένο κατώφλι. Χρησιμοποιεί την αθροιστική συνάρτηση της Κανονικής κατανομής $\Phi(x)$ και μπορεί να προσεγγιστεί ικανοποιητικά με τον παρακάτω τύπο [3], [7]–[9]:

$$GELU(x) \approx 0.5x\left(1+\tanh\left[\sqrt{\frac{2}{\pi}}(x+0.044715x^3)\right]\right)$$

2.1.5 Εφαρμογές

Τα Νευρωνικά Δίκτυα είναι πάρα πολύ ισχυρά υπολογιστικά εργαλεία και βρίσκουν εφαρμογές σε πολλούς τομείς της καθημερινότητας. Μπορούν να χρησιμοποιηθούν για:

- **Ταξινόμηση**: ο στόχος της ταξινόμησης είναι η πρόβλεψη της κλάσης στην οποία ανήκει ένα αντικείμενο. Οι κλάσεις είναι προκαθορισμένα σύνολα στα οποία μπορεί να ανήκει ένα δείγμα.
- **Παλινδρόμηση**: ο στόχος της παλινδρόμησης είναι η πρόβλεψη μίας συνεχούς τιμής βάσει των δοσμένων δεδομένων εισόδου. Μπορεί να λάβει οποιαδήποτε πραγματική τιμή μέσα σε ένα δεδομένο εύρος.
- **Ομαδοποίηση**: ο στόχος της ομαδοποίησης είναι η δημιουργία συνόλων δειγμάτων με κοινά χαρακτηριστικά, σε δεδομένα για τα οποία δεν έχουμε κάποια προηγούμενη γνώση.
- **Συσχέτιση**: ο στόχος της συσχέτισης είναι η εύρεση σχέσεων μεταξύ μεταβλητών τιμών και η δημιουργία μοτίβων.

Τα Νευρωνικά Δίκτυα αποτελούν το ισχυρότερο εργαλείο Μηχανικής Μάθησης αυτή τη στιγμή, ενώ οι αρχιτεκτονικές των Συνελικτικών Νευρωνικών Δικτύων, των Επαναλαμβανόμενων Νευρωνικών Δικτύων και των Μετασχηματιστών παράγουν εντυπωσιακά αποτελέσματα σε ό,τι benchmark δοκιμαστούν.

2.2 Επεξεργασία Φυσικής Γλώσσας

Η Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing – NLP) είναι ο κλάδος της επιστήμης υπολογιστών, και πλέον λόγω των πρόσφατων τεχνολογικών εξελίξεων κυρίαρχα ο κλάδος της Τεχνητής Νοημοσύνης, που πραγματεύεται την ερμηνεία, την επεξεργασία, τη διαχείριση και την κατανόηση γραπτού κειμένου από υπολογιστικά συστήματα. Οι τρόποι με τους οποίους επιτυγχάνεται αυτό, είναι η μοντελοποίηση της ανθρώπινης γλώσσας, μέσω στατιστικής ανάλυσης ή/και μηχανικής μάθησης, και η γλωσσολογική ανάλυση των δοσμένων κειμένων.

Ο τομέας του NLP αναπτύσσεται διαρκώς τα τελευταία χρόνια, και αυτό σε συνδυασμό με την ανάπτυξη της Μηχανικής Μάθησης, την επιτάχυνση του υλικού και τα μεγάλα και εξειδικευμένα σύνολα δεδομένων εκπαίδευσης έχει ως αποτέλεσμα τα σύγχρονα μοντέλα να έχουν υψηλή ακρίβεια σε χαμηλό χρόνο εκτέλεσης. Τα πρώτα αποτελεσματικά μοντέλα NLP βασίζονταν στην Βαθιά Μάθηση και εκμεταλλεύονταν τα Βαθιά Νευρωνικά Δίκτυα (DNNs) για την εξαγωγή συμπερασμάτων. Η Βαθιά Μάθηση έφερε επανάσταση στον τομέα και όρισε το δείκτη αναφοράς για εργασίες που αφορούν το NLP. Τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα (RNNs), τα οποία χρησιμοποιούν πολλαπλά επίπεδα και επαναλαμβανόμενες, ανατροφοδοτούμενες μονάδες, αποτέλεσαν την εξέλιξη των DNNs στην Επεξεργασία Φυσικής Γλώσσας, καθώς μπορούν να επεξεργαστούν αποτελεσματικά σειριακά δεδομένα και να αναγνωρίσουν πρότυπα βάσει της θέσης των δεδομένων εισόδου. Κατέστη με τον τρόπο αυτό δυνατή η εξαγωγή πιο σύνθετων χαρακτηριστικών και η αύξηση της ακρίβειας των μοντέλων, εις βάρος όμως της πολυπλοκότητας και της ταχύτητας υπολογισμού. Η αρχιτεκτονική των Μετασχηματιστών (Transformers) ήρθε να λύσει τα προβλήματα αυτά, μέσω του μηχανισμού αυτό-προσοχής, και αποτελεί σήμερα την τελευταία λέξη της τεχνολογίας στο NLP.

Οι εφαρμογές του NLP είναι πολυπληθείς και συνεχώς αυξάνονται. Από τους Έξυπνούς Βοηθούς (AI Assistants) και τα Ρομπότ Συνομιλητές (Chatbots) μέχρι τα μοντέλα μετάφρασης κειμένου, υπάρχει η δυνατότητα για ανάλυση, κατανόηση, σύνοψη και κατηγοριοποίηση κειμένου, απάντηση ερωτήσεων και εξαγωγή πληροφορίας. Για την ανάπτυξη τους, έχουν δημιουργηθεί ένας αριθμός από σύνολα δεδομένων και benchmarks, μέσω των οποίων μπορούν να αξιολογούνται αξιόπιστα τα μοντέλα, βάσει του εκάστοτε task [10]–[12].

2.2.1 Εφαρμογές της Επεξεργασίας Φυσικής Γλώσσας

Στην ενότητα αυτή θα αναλύσουμε κάποιες από τις κύριες εφαρμογές της Επεξεργασίας Φυσικής Γλώσσας, που έχουν ως κύριο στόχο την ανάλυση και την επεξεργασία γραπτού κειμένου.

Αυτόματη Μετάφραση

Η αυτόματη μετάφραση στοχεύει στην μετατροπή ενός κειμένου από μία γλώσσα σε μία άλλη διατηρώντας το νόημα και τη συνοχή περιεχομένου του αρχικού κειμένου. Οι κυριότερες υποκατηγορίες είναι οι εξής:

- Μη επιβλεπόμενη αυτόματη Μετάφραση: Η κατηγορία αυτή περιλαμβάνει εφαρμογές μετατροπής κειμένου από μία αρχική γλώσσα σε μία γλώσσα στόχο, χρησιμοποιώντας κατά την εκπαίδευση κείμενα και από τις δύο γλώσσες, που όμως δεν σχετίζονται μεταξύ τους. Με τον

τρόπο αυτό το μοντέλο αναγκάζεται να μάθει τις σχέσεις μεταξύ των γλωσσών, και μπορεί να παράξει μεταφράσεις σε ζεύγη γλωσσών που πιθανόν τα ήδη υπάρχοντα μεταφρασμένα κείμενα να μην επαρκούν.

- Μεταγραμματισμός (Transliteration): Η κατηγορία αυτή περιλαμβάνει εφαρμογές μετατροπής κειμένου από ένα αλφάβητο σε ένα άλλο, με τρόπο τέτοιο ώστε να διατηρείται φωνητικά το νόημα, αντί να μεταφράζεται από τη μία γλώσσα στην άλλη. Οι εφαρμογές αυτές είναι χρήσιμες για γλώσσες που δεν χρησιμοποιούν το λατινικό αλφάβητο, τη στιγμή που τα πληκτρολόγια αρκετών συσκευών υποστηρίζουν μόνο αυτό [10], [12]–[16].

Απάντηση Ερωτήσεων

Οι εφαρμογές απάντησης ερωτήσεων εστιάζουν στην εξαγωγή πληροφοριών από κείμενα, και η παρουσίαση αυτών σε μορφή κειμένου ως απάντηση σε μία ερώτηση που έθεσε ο χρήστης. Τα συστήματα απάντησης ερωτήσεων χωρίζονται σε υποκατηγορίες ανάλογα από που αντλούν πληροφορίες και τι είδους απάντηση είναι εκπαιδευμένα να δίνουν. Παρακάτω αναλύονται μερικές από τις πιο συνηθισμένες υποκατηγορίες εφαρμογών απάντησης ερωτήσεων:

- Επιλογής Απάντησης: Οι εφαρμογές αυτές αναλύουν ένα σύνολο από πιθανές απαντήσεις για μία δοσμένη ερώτηση και εκπαιδεύονται ώστε να επιλέγουν την καταλληλότερη.
- Κατανόησης κειμένου: Οι εφαρμογές αυτές καλούνται να κατανοήσουν ένα σώμα κειμένου που τους δίνεται, και να αντλήσουν από αυτό σχετικές πληροφορίες, ώστε να είναι σε θέση να απαντήσουν σε ερωτήσεις πάνω στο κείμενο αυτό
- Απάντηση Ερωτήσεων σε Μορφή Συζήτησης: Οι εφαρμογές αυτές πρέπει να είναι σε θέση να κατανοήσουν το περιεχόμενο και το πλαίσιο στο οποίο γίνεται μια συζήτηση, και να δίνουν απαντήσεις σε ερωτήσεις, χωρίς να διαταράσσουν τη ροή της συζήτησης αυτής, και διατηρώντας τον τόνο.
- Απάντηση ερωτήσεων από το Open Domain: Οι εφαρμογές αυτές απαντάνε στις ερωτήσεις του χρήστη, αναλύοντας μεγάλο όγκο κειμένου, όπως π.χ. όλα τα άρθρα της Wikipedia, ή και ολόκληρο το διαδίκτυο (Internet). Πρέπει επίσης να είναι σε θέση να συνθέσουν πληροφορίες από διαφορετικές πηγές, ώστε να παράξουν την βέλτιστη απάντηση στη δοσμένη ερώτηση [13].

Ανάλυση Συναισθήματος

Οι εφαρμογές Ανάλυσης Συναισθήματος εστιάζουν στην ερμηνεία των συναισθημάτων που εκφράζονται σε ένα δοσμένο κείμενο. Τα μοντέλα που εκπαιδεύονται στο task αυτό είναι σε θέση να αναθέσουν την αντίστοιχη ετικέτα (label) σε κείμενα βάσει του περιεχομένου, είτε για θετικό ή αρνητικό συναίσθημα, είτε πιο εξειδικευμένα για χαρά, λύπη, θυμό, έκπληξη και για όσα συναισθήματα μπορεί να εκπαιδεύεται το μοντέλο μας. Αυτό επιτυγχάνεται με την δημιουργία σχέσεων μεταξύ της ύπαρξης λέξεων ή φράσεων με το αντίστοιχο συναίσθημα που εκφράζουν. Οι εφαρμογές αυτές είναι χρήσιμες στους τομείς της Εξυπηρέτησης Πελατών, το Μάρκετινγκ, της Αξιολόγησης Παροχής Υπηρεσιών, κ.α. [10], [11], [13].

Αναγνώριση Ονομαστικών Οντοτήτων

Ο όρος αυτός αναφέρεται σε εφαρμογές NLP που έχουν σκοπό την αναγνώριση και κατηγοριοποίηση ονομάτων σε ένα δοσμένο κείμενο. Έστω για παράδειγμα ότι δίνεται η πρόταση «Ο Νίκος έχει καταγωγή από τα Ιωάννινα και τα επισκέπτεται συχνά». Μία εφαρμογή Αναγνώρισης Ονομαστικών Οντοτήτων που λάμβανε την παραπάνω πρόταση ως είσοδο θα πρέπει να επιστρέψει τις οντότητες «Νίκος» και «Ιωάννινα». Αναλόγως την εφαρμογή, υπάρχει και η δυνατότητα κατηγοριοποίησης των εξόδων βάσει προκαθορισμένων τάξεων. Στο παράδειγμα μας η οντότητα «Νίκος» κατηγοριοποιείται στην τάξη «Πρόσωπο» και η οντότητα «Ιωάννινα» στην τάξη «Πόλη». Οι εφαρμογές αυτές αποτελούν απαραίτητο ενδιάμεσο στάδιο για πιο περίπλοκες διεργασίες, όπως είναι η Απάντηση Ερωτήσεων και η Ταξινόμηση Κειμένου [10], [11], [13].

Παραγωγή Κειμένου

Τα μοντέλα Παραγωγής Κειμένου είναι σε θέση να παράγουν κείμενο βάσει μίας πρότασης-προτροπής (prompt) που τους δίνεται. Το νέο αυτό κείμενο που παράγεται ως έξοδος πρέπει να προκύπτει βάσει λογικής από το prompt, ενώ οι υπόλοιποι παράγοντες όπως το μέγεθος, ο τόνος ή η αναλυτικότητα της απάντησης βασίζονται στον τρόπο εκπαίδευσης και στην είσοδο που δίνεται. Οι εφαρμογές Παραγωγής Κειμένου έχουν γίνει πλέον εξαιρετικά διαδεδομένες λόγω της κυκλοφορίας των ChatGPT, Bing Chat, Bard, κ.α., τα οποία προσφέρουν τη δυνατότητα Παραγωγής Κειμένου, Απάντησης Ερωτήσεων και άλλες NLP εφαρμογές σε μία συμπαγή μορφή. Μερικές υποκατηγορίες της Παραγωγής Κειμένου είναι οι παρακάτω:

- Παραγωγή Κώδικα: Οι εφαρμογές αυτές εστιάζουν στην παραγωγή λειτουργικού κώδικα, βάσει της εισόδου που τους δίνεται σε μορφή κειμένου, η οποία συνήθως είναι κάποια περιγραφή ενός προβλήματος λογισμικού σε φυσική γλώσσα. Συνήθως αυτού του είδους τα προγράμματα, μπορεί και να εξειδικεύονται σε μία συγκεκριμένη γλώσσα προγραμματισμού για πιο περίπλοκες εισόδους, είτε και να κατέχουν γενικευμένη θεωρητική γνώση, ώστε να καθοδηγούν τον χρήστη στη λύση με τη χρήση ψευδογλώσσας. Χρησιμοποιούνται εκτενώς μέχρι στιγμής για την παραγωγή κυρίως απλού κώδικα και συναρτήσεων.
- Παραγωγή Κειμένου από Δεδομένα: Οι εφαρμογές αυτές δέχονται ως είσοδο δεδομένα σε μορφή πινάκων, υπολογιστικών φύλλων ή και γράφων και έχουν τη δυνατότητα να παράξουν κείμενο βάσει αυτών. Χρησιμοποιούνται μεταξύ άλλων για την σύνοψη και οπτικοποίηση δεδομένων ή για εφαρμογές προσβασιμότητας, σε συνδυασμό με εφαρμογές που μετατρέπουν Κείμενο σε Φωνή [13], [17].

Ταξινόμηση Κειμένου

Τα μοντέλα ταξινόμησης στοχεύουν στο να λύσουν το πρόβλημα της ταξινόμησης κειμένων σε προκαθορισμένες κατηγορίες, για να είναι πιο εύκολη η αρχειοθέτηση και η οργάνωση τους. Εκπαιδεύονται πάνω σε κείμενα, τα οποία περιέχουν σχόλια και ετικέτες για το είδος στο οποίο ανήκει το κείμενο, ώστε να είναι σε θέση να αναγνωρίσουν τη σχέση μεταξύ του περιεχομένου και της ετικέτας. Μερικές από τις κύριες υποκατηγορίες Ταξινόμησης Κειμένου είναι οι εξής:

- Ταξινόμηση εγγράφων: Οι εφαρμογές αυτές ταξινομούν κείμενα-έγγραφα σε κατηγορίες όπως Άρθρο, Επιστημονική Δημοσίευση, Δημοσίευση σε Μέσο Κοινωνικής Δικτύωσης. Μπορούν με προσεκτική ρύθμιση (fine-tuning) να εξειδικευτούν σε συγκεκριμένους τομείς, όπως π.χ. σε Νομικά Έγγραφα και να επιτύχουν υψηλή ακρίβεια ταξινόμησης.
- Ταξινόμηση Αιτίου και Αποτελέσματος: Οι εφαρμογές αυτές είναι σε θέση να εντοπίσουν τη σχέση αιτίου-αποτελέσματος μεταξύ δύο γεγονότων σε ένα δοσμένο κείμενο. Χρησιμοποιούνται συνήθως για προεπεξεργασία και σαν αρχικό στάδιο εκπαίδευσης σε πιο περίπλοκες εφαρμογές [13].

2.2.1 Σύνολα Δεδομένων και Benchmarks

Η ανάπτυξη που έχει βιώσει το πεδίο του NLP τα τελευταία χρόνια οφείλεται εν μέρει και στην ύπαρξη ποιοτικών και μεγάλων σε μέγεθος συνόλων δεδομένων, τα οποία επιτρέπουν στα μοντέλα να εκπαιδεύονται πιο αποτελεσματικά. Ταυτόχρονα υπάρχουν κάποια διαδεδομένα σύνολα δεδομένων τα οποία λειτουργούν και ως εργαλεία κατάταξης των μοντέλων, ώστε να υπάρχει ένας ενιαίος τρόπος να συγκριθεί η απόδοση τους. Παρακάτω αναφέρονται συνοπτικά μερικά από τα βασικά Σύνολα Δεδομένων και τις εφαρμογές στις οποίες χρησιμοποιούνται:

- RACE: Το σύνολο δεδομένων RACE [18] περιέχει δεδομένα που έχουν συλλεχθεί από εξετάσεις Αγγλικών για μαθητές Γυμνασίου και Λυκείου της Κίνας. Αποτελείται από αποσπάσματα κειμένου και ερωτήσεις πάνω σε αυτά, οι οποίες έχουν γραφτεί από φυσικά πρόσωπα και χρησιμοποιείται για την εκπαίδευση και αξιολόγηση μοντέλων Κατανόησης Κειμένου. Η απόδοση των NLP μοντέλων για το συγκεκριμένο σύνολο δεδομένων πλέον μπορεί να προσεγγίζει την μέγιστη απόδοση ενός ανθρώπου που δοκιμάζεται πάνω σε αυτό [12], [18].
- GLUE: Το GLUE (General Language Understanding Evaluation) είναι benchmark για την αξιολόγηση της Κατανόησης Κειμένου για NLP μοντέλα. Αποτελείται από 9 διαφορετικά σύνολα δεδομένων, τα οποία αξιολογούν τα μοντέλα σε διαφορετικά tasks, ώστε να προκύψει μια σφαιρική εικόνα της απόδοσης του μοντέλου. Τα δύο από αυτά έχουν ως είσοδο μια πρόταση, τα τρία ελέγχουν την ομοιότητα των εισόδων, και τα υπόλοιπα τέσσερα ελέγχουν την εξαγωγή συμπερασμάτων βάσει της εισόδου. Το GLUE είναι σχεδιασμένο με τέτοιο τρόπο ώστε μοντέλα, τα οποία έχουν γενικευμένη κατανόηση κειμένων, και που μπορούν να διαπρέψουν σε διαφορετικά tasks, χωρίς την ανάγκη ύπαρξης μεγάλου όγκου δεδομένων εκπαίδευσης, να αποδίδουν καλύτερα [12], [19].
- SQuAD: Το SQuAD (Stanford Question Answering Dataset) είναι ένα σύνολο δεδομένων, που αποτελείται από ερωτήσεις και απαντήσεις, που έχουν παραχθεί με crowdsourcing, και οι οποίες βασίζονται σε αποσπάσματα κειμένων της Wikipedia. Χρησιμοποιείται για την εκπαίδευση μοντέλων Απάντησης Ερωτήσεων [20].

Παρακάτω παρατίθενται στον πίνακα 2.1 συγκεντρωμένα τα στοιχεία για τα παραπάνω benchmarks:

Benchmark	Task	Σύνολο Δεδομένων	Μέγεθος	Διαχωρισμός
RACE	Κατανόηση Κειμένου	RACE	24.26 MiB	train 18,728 validation 1021 test 1045
GLUE	Ταξινόμηση Προτάσεων	CoLA	368.14 KiB	train 8551 validation 1043 test 1063
		SST-2	7.09 MiB	train 67,349 validation 872 test 1821
	Ομοιότητα και Παράφραση	MRPC	1.43 MiB	train 3668 validation 40 test 1725
		STS-B	784.05 KiB	train 5749 validation 1500 test 1379
		QQP	57.73 MiB	train 363,849 validation 40,430 test 390,965
	Εξαγωγή Συμπερασμάτων	MNLI	298.29 MiB	train 392,702 validation 9815 test 9796
		QNLI	10.14 MiB	train 104,743 validation 5463 test 5463
		RTE	680.81 KiB	train 2490 validation 277 test 3000
		WNLI	28.32 KiB	train 635 validation 71 test 146
SQuAD	Απάντηση Ερωτήσεων	SQuAD1.0 SQuAD2.0	94.04 MiB	train 87,599 validation 10,570

Πίνακας 2.1: NLP Benchmarks [12]

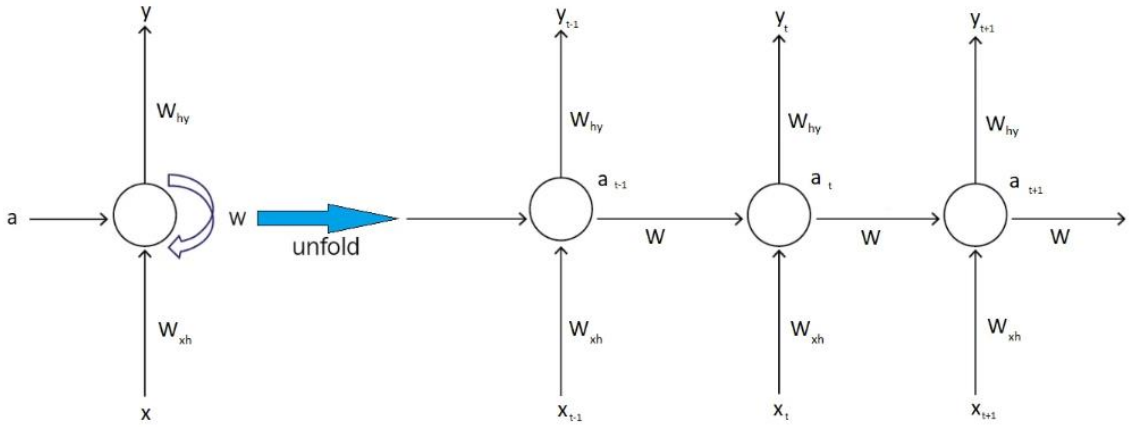
2.3 Αρχιτεκτονικές Μοντέλων Επεξεργασίας Φυσικής Γλώσσας

Όπως αναφέραμε παραπάνω οι τεχνολογίες στις οποίες βασίζονται οι εφαρμογές Επεξεργασίας Φυσικής Γλώσσας έχουν εξελιχθεί σημαντικά τα τελευταία χρόνια, και βασίζονται στη Βαθιά Μάθηση. Οι πλέον χρησιμοποιούμενες αρχιτεκτονικές στο NLP είναι αυτές των Επαναλαμβανόμενων Νευρωνικών Δικτύων και των Μετασχηματιστών. Η βελτιωμένη απόδοση τους οφείλεται στο γεγονός ότι δέχονται ως είσοδο σειριακά δεδομένα και διατηρούν την πληροφορία της θέσης κάθε λέξης στην είσοδο, σε αντίθεση με προηγούμενες τεχνολογίες, που αντιμετώπιζαν το κείμενο ως σάκο λέξεων. Οι σάκοι λέξεων κρατούν την πληροφορία της συχνότητας εμφάνισης των λέξεων, αλλά όχι την πληροφορία της θέσης τους. Ωστόσο, σε αρκετές εφαρμογές, και ιδίως σε αυτές όπου το μέγεθος του τμήματος κειμένου είναι σχετικά μικρό, η προσέγγιση αυτή είναι ανεπαρκής, καθώς μπορεί να χαθεί σημαντικά το νόημα ή να θεωρηθούν παρόμοια ή και πανομοιότυπα μεταξύ τους αποσπάσματα τα οποία έχουν εντελώς διαφορετική σημασία.

2.3.1 Επαναλαμβανόμενα Νευρωνικά Δίκτυα

Τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks – RNNs) είναι ένας τύπος Νευρωνικού Δικτύου στον οποίο το κρυφό στρώμα του Δικτύου λαμβάνει ως είσοδο όχι μόνο την έξοδο των προηγούμενων κόμβων, αλλά και την ίδια του την έξοδο. Με τον τρόπο αυτό δημιουργούνται σχέσεις εξάρτησης μεταξύ των προηγούμενων και των παρόντων δεδομένων εισόδου.

Στο σχήμα 2.10 (α) παρακάτω φαίνεται το απλούστερο επαναλαμβανόμενο νευρωνικό δίκτυο. Η είσοδος x αντιπροσωπεύει την είσοδο στο κρυφό στρώμα και η έξοδος y τις πιθανότητες προβλεπόμενης λέξης. Παρατηρούμε στο σχήμα αυτό την ύπαρξη του αυτό-βρόγχου που μεταβάλει την κρυφή κατάσταση του νευρωνικού μετά την είσοδο μίας νέας λέξης στην αλληλουχία. Οι αλληλουχίες που δίνουμε όμως ως είσοδο σε αυτά τα μοντέλα είναι πεπερασμένου μήκους, συνεπώς έχει νόημα να «ξεδιπλώσουμε» τον βρόγχο αυτό σε ένα μακρύτερο δίκτυο με «χρονικά επίπεδα» το οποίο προσομοιάζει ένα παραδοσιακό προς τα εμπρός τροφοδοτούμενο δίκτυο. Το δίκτυο που προκύπτει φαίνεται στο Σχήμα 2.10 (β).



(α) (β)

χήμα 2.10: Ένα επαναλαμβανόμενο νευρωνικό δίκτυο (α) και το ισοδύναμο δίκτυο που προκύπτει αν το «ξετυλίξουμε» στο χρόνο (β) [21]

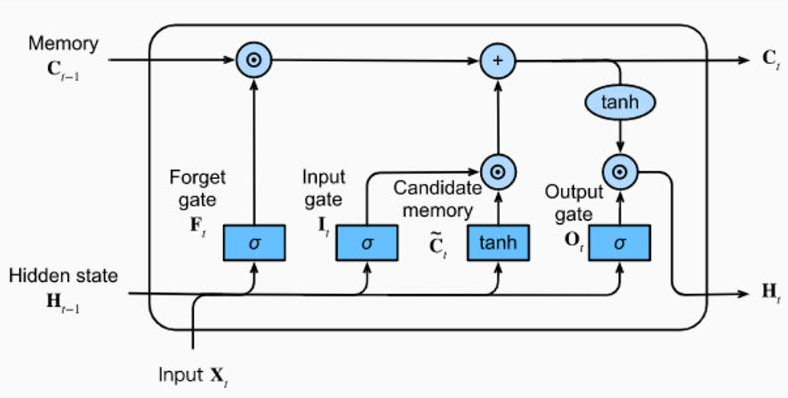
Η δυνατότητα των Επαναλαμβανόμενων Νευρωνικών Δικτύων να διατηρούν την πληροφορία προηγούμενων χρονικών βαθμίδων μας επιτρέπει να αναλύουμε καλύτερα σειριακά δεδομένα, και θεωρητικά τα RNNs μπορούν να διατηρούν την πληροφορία ανεξαρτήτως του μεγέθους της εισόδου. Στην πράξη όμως προκύπτουν μια σειρά από προκλήσεις στην εκπαίδευση των δικτύων αυτών. Τα RNNs είναι εξαιρετικά δύσκολο να εκπαιδευτούν, εξαιτίας του πλήθους των χρονικών επιπέδων, το οποίο εξαρτάται από το μήκος της εισόδου. Πέραν αυτού, η συνάρτηση απώλειας εμφανίζει διαφορετική ευαισθησία ανά διαφορετικό χρονικό επίπεδο, ενώ όλα τα επίπεδα μοιράζονται τους ίδιους πίνακες παραμέτρων. Ο συνδυασμός αυτός, δηλαδή η μεταφορά του υπολογιζόμενου σφάλματος, σε προηγούμενες βαθμίδες (λόγω του back propagation), το αυξημένο πλήθος βαθμίδων με ίδιες παραμέτρους, προκαλεί το Πρόβλημα της Εξαφάνισης και Εκτόξευσης της Κλίσης. Αντιμετωπίζουμε τα προβλήματα αυτά όταν η κλίση της συνάρτησης απώλειας καταλήγει να είναι εξαιρετικά μικρή ή εξαιρετικά μεγάλη αντίστοιχα, λόγω των πολλών συνεχόμενων πολλαπλασιασμών, των συναρτήσεων ενεργοποίησης και άλλων παραγόντων που συναντάμε στα RNNs. Αποτέλεσμα αυτού είναι να μην υπάρχει πραγματική εξάρτηση της εξόδου από παλαιότερες χρονικά εισόδους και το μοντέλο μας να μην μπορεί να διαχειριστεί μεγάλα σε μήκος δεδομένα εισόδου. Τα προβλήματα δημιουργούν δυσκολίες στην ανάπτυξη των RNNs και θέτουν ρεαλιστικά όρια στις αποδόσεις που μπορούν να επιτευχθούν με αυτά [3], [21].

Δίκτυα Long Short Term Memory (LSTM)

Τα δίκτυα LSTM αναπτύχθηκαν για να αντιμετωπίσουν το παραπάνω πρόβλημα, αποτελούν την εξέλιξη των RNNs και είναι σχεδιασμένα ώστε να διατηρούν μακροπρόθεσμες εξαρτήσεις. Λειτουργούν αλλάζοντας τις συνθήκες του αυτό-βρόγχου και τον τρόπο με τον οποίο διαδίδονται οι κρυφές καταστάσεις. Αυτό το επιτυγχάνουν χρησιμοποιώντας έναν μηχανισμό πύλης που ελέγχει τη ροή της πληροφορίας στο Νευρωνικό Δίκτυο. Για το σκοπό αυτό επιστρατεύουν ένα κρυφό διάνυσμα, το οποίο αναφέρεται ως κατάσταση κυψέλης, συμβολίζεται με c, και το οποίο αποτελεί ένα είδος μακροπρόθεσμης μνήμης, που διατηρεί ένα μέρος της πληροφορίας σε προηγούμενες καταστάσεις κυψέλης. Τα LSTM χρησιμοποιούν 3 πύλες:

- την πύλη εισόδου I: ελέγχει την ροή πληροφορίας από προηγούμενες καταστάσεις
- την πύλη εξόδου O: ελέγχει την ροή πληροφορίας σε επόμενες καταστάσεις
- την πύλη ξεχάσματος F: ελέγχει πόση από την πληροφορία των προηγούμενων καταστάσεων θα ξεχαστεί

Παρακάτω παρατίθεται στο σχήμα 2.11 μία κυψέλη LSTM.



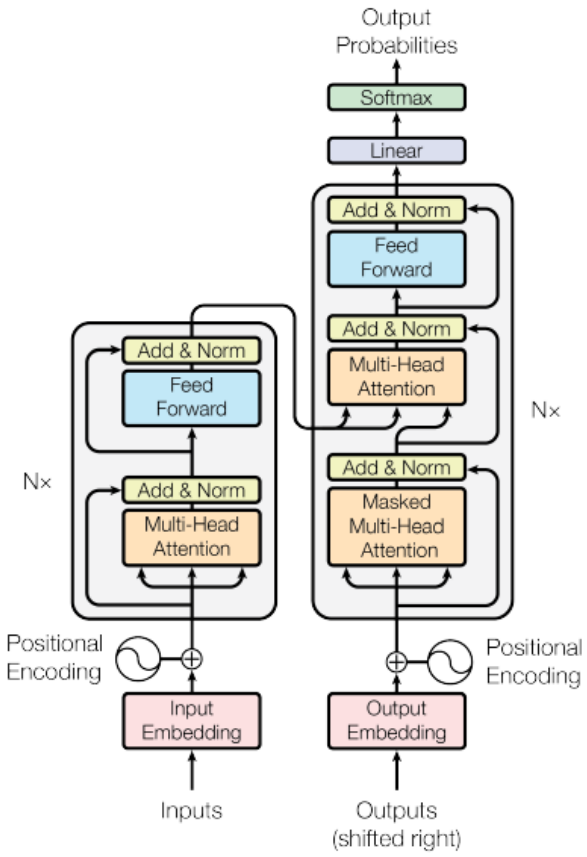
Σχήμα 2.11: Η αρχιτεκτονική ενός LSTM [22]

Η δυνατότητα της κατάστασης κυψέλης να ελέγχει την πληροφορία μεταξύ των χρονικών βαθμίδων, επιτρέπει στα LSTM να διατηρούν εξαιρετικά μακροπρόθεσμες εξαρτήσεις χωρίς να επηρεάζονται από το Φαινόμενο Εξαφάνισης-Έκρηξης Κλίσης. Έτσι τα LSTM έχουν πετύχει βελτιωμένες αποδόσεις σε σχέση με τα παραδοσιακά RNNs σε εφαρμογές Επεξεργασίας Φυσικής Γλώσσας και όχι μόνο. Εξακολουθούν παρόλα αυτά να είναι χρονοβόρα και δύσκολα στην εκπαίδευση και στη συμπερασματολογία [3], [22], [23].

2.3.2 Μετασχηματιστές

Η αρχιτεκτονική των Μετασχηματιστών (Transformers) προτάθηκε για πρώτη φορά από τους Vaswani, et al [4] το 2017. Βασίζεται στον μηχανισμό αυτό-προσοχής (self-attention) για κάθε λέξη, ώστε να διαπιστώσει τη σημασία όλων των υπόλοιπων λέξεων της πρότασης σε σχέση με την λέξη αυτή. Οι Μετασχηματιστές δε χρησιμοποιούν Επανάληψη, και για το λόγο αυτό η εκτέλεσή τους μπορεί να γίνει παράλληλα και αποδοτικά.

Αποτελούνται από ένα ζεύγος Κωδικοποιητή και Αποκωδικοποιητή, όπως φαίνεται στο Σχήμα 2.12. Ο Κωδικοποιητής λαμβάνει ως είσοδο δεδομένα μεταβλητού μήκους και τα κωδικοποιεί, ώστε ο Αποκωδικοποιητής να λάβει την είσοδο και το περιεχόμενο της πρότασης μέχρι στιγμής και να προβλέψει τα επόμενα σύμβολα βάσει αυτών.



Σχήμα 2.12: Η αρχιτεκτονική ενός μοντέλου Μετασχηματιστή. Αριστερά η μονάδα Κωδικοποιητή και δεξιά η μονάδα Αποκωδικοποιητή [4]

Ο παράγοντας διαφοροποίησης με άλλες τεχνολογίες είναι η συμπερίληψη στο ζεύγος Κωδικοποιητή-Αποκωδικοποιητή της Αυτό-Προσοχής. Ο μηχανισμός προσοχής δημιουργεί μία χαρτογράφηση μεταξύ ενός ερωτήματος (Query) και ενός συνόλου κλειδιών (Keys) και τιμών (Values) σε μία έξοδο. Ως είσοδος στο μηχανισμό δίνονται τα 3 διανύσματα Query, Keys και Values, τα οποία προκύπτουν από τα αντίστοιχα βάρη και τα δεδομένα εισόδου του μοντέλου και αντιπροσωπεύουν τα κομμάτια της πρότασης στα οποία εστιάζουμε, τα κομμάτια της πρότασης που σχετίζονται με αυτά, και την πρόβλεψη για τα επόμενα σύμβολα αντίστοιχα. Τα διανύσματα αυτά στη συνέχεια υποβάλλονται σε διαδοχικές πράξεις τις οποίες αναλύουμε παρακάτω ώστε να προκύψει ως έξοδος η τιμή της προσοχής (attention). Ο τύπος για την προσοχή δίνεται παρακάτω, όπου Q το διάνυσμα Query, K το διάνυσμα Keys, V το διάνυσμα Values και d_k η διάσταση του διανύσματος Keys:

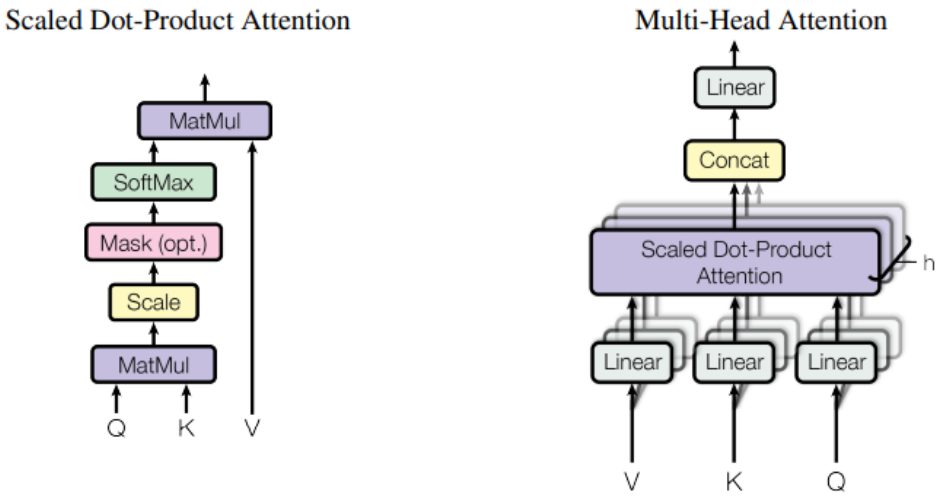
$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Ο παραπάνω μηχανισμός προσοχής λέγεται Scaled Dot Product Attention καθώς χρησιμοποιεί την πράξη του εσωτερικού γινομένου και κλιμακώνεται βάσει της διάστασης d_k.

Οι μετασχηματιστές χρησιμοποιούν Προσοχή Πολλαπλών Κεφαλών (Multi-Head-Attention), προβάλλοντας τα διανύσματα που αναφέραμε γραμμικά και υπολογίζοντας πολλές διαφορετικές Scaled Dot Product Attentions, όπως φαίνεται στο Σχήμα 2.13. Με αυτό επιτυγχάνεται το να γίνεται παράλληλα ο υπολογισμός και να διατηρείται καλύτερα η πληροφορία από πολλά σημεία της εισόδου, η οποία πιθανόν να χάνεται λόγω της χρήσης του μέσου όρου στο Scaled Dot Product Attention. Ο τύπος για την Προσοχή Πολλών Κεφαλών δίνεται παρακάτω, όπου W_i^Q, W_i^K, W_i^V τα βάρη για την κεφαλή Προσοχής i, τα οποία πολλαπλασιάζουμε με τα διανύσματα Query, Key και Value αντίστοιχα, και W^O τα βάρη με τα οποία πολλαπλασιάζουμε το συνενωμένο διάνυσμα που προκύπτει από τις εξόδους των Κεφαλών Προσοχής. Για καλύτερη παραλληλοποίηση του υπολογισμού γίνεται ταυτόχρονος πολλαπλασιασμός πινάκων κατά τον υπολογισμό της Προσοχής για όλα τα δείγματα της παρτίδας, αυξάνοντας τη διαστατικότητα κατά μία διάσταση, η οποία έχει τιμή ίση με το πλήθος των δειγμάτων της παρτίδας. Με τον τρόπο αυτό, επιτυγχάνεται μείωση του απαιτούμενου χρόνου υπολογισμού ενώ παράλληλα λειτουργεί καλύτερα ο μηχανισμός προσοχής.

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

όπου $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$



Σχήμα 2.13: Αριστερά: Ο μηχανισμός προσοχής εσωτερικού γινομένου, Δεξιά: Η Προσοχή Πολλών Κεφαλών με τη χρήση πολλών παράλληλων Επιπέδων Προσοχής [4]

Συγκεκριμένα, τα χαρακτηριστικά που εξάγουν τα ενδιάμεσα μπλοκ Μετασχηματιστών και άρα οι ποσότητες Q, K και V έχουν διαστάσεις (N, S, H), όπου N το μέγεθος παρτίδας (batch size), S το μήκος της ακολουθίας εισόδου (sequence length) και H το κρυφό μήκος (hidden size). Για την εφαρμογή Προσοχής Πολλαπλών Κεφαλών, αρχικά η τελευταία διάσταση των παραπάνω πινάκων χωρίζεται ώστε κάθε κεφαλή να βλέπει μόνο ένα τμήμα της κρυφής διάστασης:

$$(N, S, H) \longrightarrow \left(N, A, S, \frac{H}{A}\right)$$

όπου A το πλήθος των κεφαλών. Στη συνέχεια χρησιμοποιείται ένα πρώτο επίπεδο Πολλαπλασιασμού Πινάκων σε Παρτίδες (Batch Matrix Multiplication – Batch MatMul), το οποίο είναι απαραίτητο για τον πολλαπλασιασμό των πινάκων Q και K^T, λόγω του πλήθους των διαστάσεων τους. Αναλυτικότερη περιγραφή της πράξης αυτής γίνεται σε επόμενο κεφάλαιο. Το αποτέλεσμα του Batch MatMul δίνεται παρακάτω:

$$\underset{Q}{\left(N, A, S, \frac{H}{A}\right)} \times \underset{K^T}{\left(N, A, \frac{H}{A}, S\right)} \longrightarrow (N, A, S, S)$$

Μετά την κλιμάκωση (scaling) και την εφαρμογή της συνάρτησης softmax στο παραπάνω ενδιάμεσο αποτέλεσμα, θα ακολουθήσει ένα δεύτερο BatchMatMul επίπεδο για τον πολλαπλασιασμό με τον πίνακα V, ώστε να προκύψουν τα νέα χαρακτηριστικά και να μειωθεί η διάσταση της εξόδου:

$$(N, A, S, S) \times \underset{V}{\left(N, A, S, \frac{H}{A}\right)} \longrightarrow \left(N, A, S, \frac{H}{A}\right) \longrightarrow (N, S, H)$$

Οι μετασχηματιστές αυτή τη στιγμή αποτελούν την state-of-the-art αρχιτεκτονική όσον αφορά τα μοντέλα Επεξεργασίας Φυσικής Γλώσσας. Ο μηχανισμός προσοχής τους επιτρέπει να αντιλαμβάνονται καλύτερα από κάθε άλλη υπάρχουσα αρχιτεκτονική τις μακροπρόθεσμες εξαρτήσεις μεταξύ λέξεων, ώστε να μοντελοποιούν τη γλώσσα αποτελεσματικά, ενώ είναι υπολογιστικά σημαντικά λιγότερο απαιτητικοί. Είναι ένα διαρκώς αναπτυσσόμενο πεδίο έρευνας και ανάπτυξης την στιγμή και έχουν φτάσει να έχουν εξαιρετικά επιδόσεις σε όλα τα benchmarks, τόσο ώστε να έχουν σχεδόν αντικαταστήσει τελείως τη χρήση οποιασδήποτε άλλης αρχιτεκτονικής για NLP [4], [5]

Κεφάλαιο 3ο

Κινητές Συσκευές, Περιορισμοί και Βελτιστοποίηση

Οι κινητές συσκευές έχουν αναπτυχθεί εντυπωσιακά την τελευταία δεκαπενταετία. Οι τεχνολογίες που σχετίζονται με αυτές βελτιώνονται συνεχώς, ενώ κυκλοφορούν διαρκώς νέα μοντέλα με αναβαθμισμένες δυνατότητες υπολογισμού και απεικόνισης. Ταυτόχρονα αποτελούν αναπόσπαστο κομμάτι της σύγχρονης καθημερινότητας, και σε συνδυασμό με την άνοδο του Διαδικτύου των Πραγμάτων (Internet-of-Things - IoT) και των έξυπνων συσκευών, είναι σημαντικό, το υλικό τους να εξελίσσεται ώστε να συμβαδίζει με τις αυξημένες απαιτήσεις των σύγχρονων εφαρμογών.

3.1 Περιορισμοί και Προκλήσεις Εφαρμογών Βαθιά Μάθησης

Η ανάπτυξη εφαρμογών που χρησιμοποιούν Βαθιά Μάθηση σε Κινητές Συσκευές αναπτύσσεται ταχύτατα, όμως υπάρχουν δύο κύρια προβλήματα: οι περιορισμοί του υλικού των Κινητών Συσκευών (Μονάδες Επεξεργασίας, μνήμη, κ.α) και η πολυπλοκότητα των μοντέλων Βαθιάς Μάθησης (μέγεθος, πράξεις, κ.α).

3.1.1 Περιορισμοί Υλικού των Κινητών Συσκευών

Παρά την ανάπτυξη του υλικού των κινητών συσκευών και την έκρηξη στην υπολογιστική τους ισχύ τα τελευταία χρόνια, οι δυνατότητες τους παραμένουν περιορισμένες. Τα κύρια ζητήματα που αντιμετωπίζουν είναι τα εξής:

- **Περιορισμένες Δυνατότητες Κεντρικών Μονάδων Επεξεργασίας και Καρτών Γραφικών**: Λόγω του μικρού τους μεγέθους, στις κινητές συσκευές είναι αδύνατη η χρήση των πλακετών (System-on-Chip – SoC) που χρησιμοποιούνται σε φορητούς ή σταθερούς υπολογιστές. Οι εξειδικευμένες πλακέτες που έχουν αναπτυχθεί για χρήση σε κινητές συσκευές έχουν περιορισμένες υπολογιστικές δυνατότητες, χαμηλότερες ταχύτητες ρολογιού και λιγότερους πυρήνες σε σχέση με τις αντίστοιχες των σταθερών υπολογιστών.
- **Περιορισμένη Μνήμη**: Οι κινητές συσκευές διαθέτουν λιγότερη μνήμη από τους σταθερούς υπολογιστές και τους εξυπηρετητές, τόσο όσον αφορά το σκληρό τους δίσκο, όσο και τη RAM. Έτσι καθίσταται δύσκολη η αποθήκευση μεγάλων σε μέγεθος εφαρμογών.
- **Περιορισμοί Ισχύος**: Οι κινητές συσκευές, όντας φορητές, διαθέτουν μπαταρία, η οποία πρέπει να χρησιμοποιείται αποδοτικά. Η μπαταρία μίας κινητής συσκευής πέρα από την περιορισμένη χωρητικότητα, επίσης δεν μπορεί να παράξει τόσο υψηλή τάση όσο ένα τροφοδοτικό προσωπικού υπολογιστή, γεγονός που θέτει όριο στην υπολογιστική ισχύ που μπορεί να επιτύχει η πλακέτα.
- **Ετερογένεια**: Υπάρχει αυτή τη στιγμή μεγάλη ποικιλία όσον αφορά τις τεχνολογίες στις Κινητές Συσκευές, είτε πρόκειται για το κομμάτι του λογισμικού και του λειτουργικού συστήματος τους

(Android και iOS), είτε πρόκειται για το υλικό (Exynos, Snapdragon, A Series, κ.λπ.). Αυτό έχει ως αποτέλεσμα να μην είναι ενιαίες οι δυνατότητες των σύγχρονων Κινητών Συσκευών, και να απαιτείται ρύθμιση και βελτιστοποίηση των μοντέλων και των εφαρμογών που αναπτύσσονται, εξειδικευμένα για συγκεκριμένα Λειτουργικά Συστήματα, Μονάδες Επεξεργασίας ή και μεμονωμένες συσκευές. Έτσι αυξάνεται το έργο που απαιτείται για την δημιουργία προγραμμάτων, που να εκμεταλλεύονται στο έπακρο τις υπολογιστικές δυνατότητες των σύγχρονων συσκευών [24]–[27].

3.1.2 Προκλήσεις στην Ανάπτυξη DL Μοντέλων σε Κινητά

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, οι εφαρμογές της Μηχανικής Μάθησης πληθαίνουν και όπως είναι λογικό επεκτείνονται και στις κινητές συσκευές. Τα προγράμματα Αναγνώρισης και Επεξεργασίας Εικόνας, οι Εφαρμογές Επεξεργασίας Φυσικής Γλώσσας και Αναγνώρισης Φωνής, έχουν γίνει αναπόσπαστη βοήθεια στις έξυπνες συσκευές και ενσωματώνουν όλο και περισσότερο τεχνολογίες Μηχανικής και Βαθιάς Μάθησης.

Η μετάβαση, για την λύση προβλημάτων που απαιτούν Μηχανική Μάθηση, στα Βαθιά Νευρωνικά Δίκτυα ξεκίνησε το 2012 με το AlexNet [28], το οποίο κέρδισε τον ετήσιο διαγωνισμό ILSVRC [29]. Το συγκεκριμένο μοντέλο πέτυχε πολύ σημαντική βελτίωση στην απόδοση σε σχέση με προηγούμενους νικητές, και αποτέλεσε την αρχή για την ανάπτυξη των Συνελικτικών Νευρωνικών Δικτύων (Convolutional Neural Networks – CNNs). Το αρχικό μοντέλο είχε 61 εκατομμύρια παραμέτρους και 8 επίπεδα. Τα CNNs έχουν εξελιχθεί σημαντικά όμως υπάρχει συνεχώς η τάση, κυρίως αλλά όχι αποκλειστικά, στις εφαρμογές Όρασης Υπολογιστών, να χρησιμοποιούνται όλο και μεγαλύτερα και πιο περίπλοκα Βαθιά Νευρωνικά Δίκτυα. Χαρακτηριστικό παράδειγμα αποτέλεσε το VGG-19 [30] με 138 εκατομμύρια παραμέτρους. Το μέγεθος των μοντέλων αυτών έχει καταλήξει να είναι απαγορευτικό για χρήση σε κινητές συσκευές, καθώς η συμπερασματολογία παίρνει εξαιρετικά πολύ χρόνο, με αποτέλεσμα οι εφαρμογές να μην ενδείκνυνται για χρήση σε πραγματικό χρόνο. Επιπροσθέτως, το μέγεθος των μοντέλων αυτών επιβαρύνει σε υπερβολικό βαθμό τις μνήμες και τη μπαταρία των κινητών συσκευών.

Το πρόβλημα αυτό εντείνεται ακόμη περισσότερο, όταν μιλάμε για αρχιτεκτονικές Μετασχηματιστών (Transformers). Ιδίως για εργασίες Επεξεργασίας Φυσικής Γλώσσας, που είναι το κύριο πεδίο χρήσης των Transformers αυτή τη στιγμή, τα μοντέλα καλούνται συνεχώς να επιλύσουν όλο και πιο απαιτητικά tasks, χρησιμοποιούν όλο και μεγαλύτερα σύνολα δεδομένων, και συνεπώς για να ανταπεξέλθουν μεγαλώνουν συνεχώς σε μέγεθος και πολυπλοκότητα, ακολουθώντας την τάση των CNNs. Χαρακτηριστικά από τις 110 εκατομμύρια παραμέτρους του base μοντέλου BERT [31], έχουμε φτάσει στις 175 δισεκατομμύρια παραμέτρους του GPT-3 [32] και στις 1.76 τρισεκατομμύρια παραμέτρους του GPT-4 [33]. Πέρα από το μέγεθος, τα σύγχρονα Transformer μοντέλα, χρησιμοποιούν το επίπεδο Κανονικοποίησης Στρώματος (LayerNorm) και τη συνάρτηση ενεργοποίησης GELU. Η μαθηματική πολυπλοκότητα των δύο πράξεων αυτών συμβάλλει σημαντικά στην καθυστέρηση (latency) της εκτέλεσης, ενώ για αρκετές κάρτες γραφικών (GPUs), που χρησιμοποιούνται σε κινητές συσκευές, οι πράξεις αυτές δεν υποστηρίζονται. Αυτό έχει αποτέλεσμα το μοντέλο ουσιαστικά να μην μπορεί να χρησιμοποιηθεί.

3.2 Βελτιστοποιήσεις και Σχετικές Εργασίες

Έχουν προταθεί αρκετές λύσεις, οι οποίες έχουν σκοπό την βελτιστοποίηση μοντέλων CNN και Transformers σε κινητές συσκευές και όχι μόνο. Ο κλάδος της βελτιστοποίησης είναι ένα αναδυόμενο ερευνητικό πεδίο που συνοδεύει την ανάπτυξη των μοντέλων Μηχανικής Μάθησης. Ο όρος αυτός αναφέρεται στη βελτίωση της αποδοτικότητας και της επίδοσης των μοντέλων, μέσω της χρήσης τεχνικών, οι οποίες μειώνουν την απαιτούμενη υπολογιστική ισχύ, τον χρόνο εκτέλεσης και άλλες μετρικές.

Τα βελτιστοποιημένα μοντέλα, συχνά έχουν ισάξια ή και καλύτερη απόδοση από τα πρωτότυπα μοντέλα. Ταυτόχρονα, λόγω του μικρότερου μεγέθους και της ταχύτητας τους μπορούν να εκπαιδεύονται και να εκτελούνται σε μεγαλύτερο εύρος συσκευών και υπολογιστικών συστημάτων, χρησιμοποιώντας λιγότερη ενέργεια, αποθηκευτικό χώρο και υπολογιστικούς πόρους. Η βελτιστοποίηση καθίσταται ακόμη πιο απαραίτητη για συσκευές παρυφών, όπως smartphones και IoT συσκευές, οι οποίες διαθέτουν πολύ περιορισμένους πόρους και δεν έχουν πάντα τη δυνατότητα χρήσης του υπολογιστικού νέφους.

Κάποιες από τις κύριες τεχνικές που χρησιμοποιούνται για την βελτιστοποίηση μοντέλων μηχανικής μάθησης είναι οι εξής:

- **Απόσταξη γνώσης (Knowledge Distillation)**: Ο όρος αυτός αναφέρεται στη διαδικασία εκπαίδευσης ενός μικρού σε μέγεθος και αριθμό παραμέτρων μοντέλου, ώστε να μιμείται τις προβλέψεις και την απόδοση ενός μεγαλύτερου, πιο περίπλοκου μοντέλου. Με αυτόν τον τρόπο δημιουργούνται συμπαγή και αποδοτικά μοντέλα, χωρίς να έχουμε σοβαρή πτώση της ακρίβειας [34].
- **Κλάδεμα παραμέτρων (Pruning)**: Στο κλάδεμα παραμέτρων αφαιρούνται συνάψεις και βάρη, τα οποία δε συμβάλλουν στην ακρίβεια του μοντέλου, ώστε να μειωθεί το μέγεθος χωρίς να μειωθεί σοβαρά η απόδοση [35].
- **Κβαντοποίηση (Quantization)**: Η κβαντοποίηση είναι τεχνική μείωσης της αριθμητικής ακρίβειας των βαρών και των ενεργοποιήσεων, ώστε να καταλαμβάνουν λιγότερο χώρο στη μνήμη και να είναι πιο εύκολη η εκτέλεση του μοντέλου. Συνήθως τα βάρη και οι ενεργοποιήσεις μετατρέπονται από τον αρχικό τους τύπο που είναι 32 bit κινητής υποδιαστολής σε κάποιον με χαμηλότερη ακρίβεια, όπως 16 bit κινητής υποδιαστολής ή 8 bit ακέραιο[35].

Σημαντική πρόοδος έχει γίνει και στον τομέα του υλικού των κινητών συσκευών. Συγκεκριμένα η ανάπτυξη πλακετών για κινητά ακολουθεί τις τάσεις που υπάρχουν στην ανάπτυξη πλακετών για σταθερούς υπολογιστές και εξυπηρετητές. Στις κεντρικές μονάδες επεξεργασίας έχει υιοθετηθεί η χρήση πολλαπλών πυρήνων που μπορούν να παραλληλοποιήσουν τον υπολογισμό, ενώ υπάρχουν επίσης εξειδικευμένες βιβλιοθήκες και εκπρόσωποι (delegates). Οι delegates [36] μπορούν να αξιοποιήσουν επιταχυντές της συσκευής ώστε να διευκολύνεται ο υπολογισμός των πράξεων σε μοντέλα Νευρωνικών Δικτύων. Μία τέτοια βιβλιοθήκη που χρησιμοποιεί εκπροσώπους είναι η XNNPack [37], η οποία προσφέρει μια σειρά από βελτιστοποιημένους τελεστές (operators) για την συμπερασματολογία Νευρωνικών Δικτύων. Επιπροσθέτως, ενδείκνυται και η χρήση Καρτών Γραφικών (GPUs) για τη συμπερασματολογία μοντέλων Μηχανικής Μάθησης, καθώς σε πολλές περιπτώσεις χρήσης ξεπερνούν τις CPUs σε επιδόσεις. Τέλος, είναι αρκετά διαδεδομένο στις νέες

κυκλοφορίες κινητών συσκευών να ενσωματώνονται Μονάδες Επεξεργασίας Νευρωνικών Δικτύων (NPUs). Οι εξειδικευμένες αυτές μονάδες βρίσκονται ακόμη σε πρώιμο στάδιο και δεν μπορούν να αναλάβουν πάντα την εκτέλεση ολόκληρων μοντέλων, παρά μόνο μερικών κόμβων τους, όμως στοχεύουν να αναλάβουν μελλοντικά τον ρόλο της εκτέλεσης μοντέλων Νευρωνικών Δικτύων, αναλαμβάνοντας όλους τους απαραίτητους υπολογισμούς.

Οι παραπάνω τρόποι μπορούν να χρησιμοποιηθούν για οποιαδήποτε μοντέλα μηχανικής μάθησης και επιτυγχάνουν σημαντικές βελτιώσεις. Υπάρχουν όμως και εξειδικευμένες τεχνικές και έρευνες που στοχεύουν στη βελτιστοποίηση των αρχιτεκτονικών Συνελικτικών Νευρωνικών Δικτύων και των Μετασχηματιστών, που βρίσκονται προς το παρόν στο επίκεντρο της έρευνας όσον αφορά τη Μηχανική Μάθηση [3], [35].

3.2.1 Βελτιστοποιήσεις Συνελικτικών Νευρωνικών Δικτύων

Έχουν προταθεί αρκετές αρχιτεκτονικές και βελτιώσεις για Συνελικτικά Νευρωνικά Δίκτυα. Η οικογένεια μοντέλων VGG-Net [30] χρησιμοποιεί αποκλειστικά συνελίξεις 3 x 3 για να προσεγγίσει τα συνελικτικά φίλτρα 7 x 7. Απαίτουνται όμως περισσότερα στρώματα, για να επιτευχθεί ισάξια απόδοση, οπότε η όποια επιτάχυνση αντισταθμίζεται από την αύξηση στο μέγεθος. Επόμενη προσπάθεια αποτέλεσε το μοντέλο MobileNet [24], το οποίο χρησιμοποιεί διαχωρίσιμη συνέλιξη βάθους (Depthwise Separable Convolution), για να αντικαταστήσει την κλασική, χρονοβόρα και υπολογιστικά απαιτητική Συνέλιξη. Ο τύπος συνέλιξης αυτός χρησιμοποιεί δύο υποστρώματα: το στρώμα Συνέλιξης Κατά Βάθος (Depthwise Convolution) και το Στρώμα Σημειακής Συνέλιξης (Pointwise Convolution). Το πρώτο στρώμα εφαρμόζει ένα φίλτρο στην είσοδο για κάθε διάσταση της εισόδου, ενώ το δεύτερο εφαρμόζει συνέλιξη με φίλτρο 1 x 1 ώστε να αλλάξει τη διάσταση της εισόδου. Πέραν αυτού, το MobileNet χρησιμοποιεί δύο μεταβλητές, τον πολλαπλασιαστή πλάτους (Width Multiplier) και τον πολλαπλασιαστή ανάλυσης (Resolution Multiplier), ώστε να μικραίνει περαιτέρω το δίκτυο και την είσοδο αντίστοιχα. Με αυτές τις βελτιστοποιήσεις το MobileNet αποτελεί ένα συμπαγές και αποδοτικό μοντέλο με μόνο 1.32 εκατομμύρια παραμέτρους στη βασική μορφή του, και κατάφερε να ξεπεράσει το AlexNet σε επιδόσεις.

Άλλες επιτυχημένες προσπάθειες αποτελούν το SqueezeNet [38] και η έξελιξη του SqueezeNext [25]. Συγκεκριμένα το δεύτερο κατάφερε να ξεπεράσει σε ακρίβεια το VGG-19 με κατά 31 φορές λιγότερες παραμέτρους, χωρίς να χρησιμοποιεί Depthwise Separable Convolutions, οι οποίες δεν υποστηρίζονται από όλα τα μοντέλα επεξεργαστών κινητών συσκευών. Χρησιμοποιεί μια δομή με σημεία συμφόρησης (bottlenecks) για να μειώσει τη διαστατικότητα της εισόδου και υιοθετεί εξειδικευμένη αρχιτεκτονική, ώστε επιτυγχάνει τη βέλτιστη επίδοση με χρήση επιταχυντή Νευρωνικών Δικτύων.

Μία προσέγγιση που κυριάρχησε είναι να αναπτύσσονται μοντέλα έχοντας ως δεδομένο την περιορισμένη διαθεσιμότητα πόρων, και στη συνέχεια να γίνεται κλιμάκωση τους εάν υπάρχει η αντίστοιχη δυνατότητα. Η δυνατότητα αυτή μελετήθηκε σε βάθος για να δημιουργηθεί η οικογένεια μοντέλων EfficientNet [39] που αποτελεί αυτή τη στιγμή το state-of-the-art στα CNNs.

Η έρευνα πάνω στις βελτιστοποιημένες αρχιτεκτονικές, σε συνδυασμό με τις μεθόδους συμπίεσης που αναφέρθηκαν παραπάνω, έχουν ως αποτέλεσμα να υπάρχουν πλέον αποδοτικά

συνελικτικά μοντέλα τα οποία να λειτουργούν ικανοποιητικά σε κινητές ή έξυπνες συσκευές. Χαρακτηριστικά παραδείγματα αποτελούν οι εφαρμογές κάμερας Τεχνητής Νοημοσύνης για κινητά και οι έξυπνοι οπτικοί αισθητήρες που «τρέχουν» σε ειδικές, συμπαγείς συσκευές.

3.2.2 Βελτιστοποιήσεις Μετασχηματιστών

Η αρχιτεκτονική των Μετασχηματιστών είναι πιο πρόσφατη από αυτή των Συνελικτικών Νευρωνικών Δικτύων. Για το λόγο αυτό, το πεδίο της βελτιστοποίησης τους δεν είναι εξίσου εξελιγμένο. Επιπροσθέτως, τα tasks για τα οποία χρησιμοποιούνται οι Μετασχηματιστές και που έχουν να κάνουν συνήθως με Επεξεργασία Φυσικής Γλώσσας, είναι πιο απαιτητικά από αυτά για τα οποία χρησιμοποιούνται τα CNNs, με αποτέλεσμα τα μεγαλύτερα Μοντέλα Μετασχηματιστών να είναι τάξεις μεγέθους μεγαλύτερα από τα αντίστοιχα Μοντέλα CNN.

Παρόλα αυτά, γίνεται ήδη προσπάθεια και υπάρχει ανάπτυξη στο πεδίο της Βελτιστοποίησης Transformers. Οι τεχνικές και οι μέθοδοι που αναφέρθηκαν παραπάνω, όπως η Απόσταξη Γνώσης, η Συμπίεση και ο σχεδιασμός με επίγνωση υλικού (Hardware Aware Design) εφαρμόζονται στα ήδη υπάρχοντα μοντέλα και στα αναπτυσσόμενα, ενώ υπάρχουν ήδη έρευνες που παρουσιάζουν ειδικευμένα μοντέλα για κινητές συσκευές. Το πιο αντιπροσωπευτικό παράδειγμα είναι το μοντέλο MobileBERT [40]. Το MobileBERT, βασίζεται στο μοντέλο BERT, αλλά έχει υιοθετήσει μία σειρά από τεχνικές συμπίεσης και προσεγγίσεων. Έχει εκπαιδευτεί με απόσταξη γνώσης. Σε αυτήν τη διαδικασία, η οποία αναφέρεται συχνά και ως «δάσκαλος-μαθητής», το ρόλο του δασκάλου είχε το πρωτότυπο μοντέλο BERT και το ρόλο του μαθητή προς τον οποίο γίνεται η μεταφορά γνώσης το MobileBERT. Αναλυτικότερη αναφορά στη λειτουργία του θα γίνει σε επόμενο κεφάλαιο. Η έρευνα στην οποία παρουσιάστηκε το MobileBERT, δείχνει ότι έχει επιτύχει αποτελέσματα συγκρίσιμα με το $\text{BERT}_{\text{BASE}}$, αλλά με 4.3 φορές μικρότερο μέγεθος και 5.5 φορές ταχύτερα.

Παρά την πολλά υποσχόμενη απόδοση του MobileBERT, το πεδίο της βελτιστοποίησης μοντέλων Transformers είναι ακόμη σε πρώιμο στάδιο, και απέχουμε από ένα μέλλον στο οποίο θα είναι δυνατή η συμπερασματολογία περίπλοκων μοντέλων στις κινητές συσκευές.

Κεφάλαιο 4ο

Προτεινόμενες Βελτιστοποιήσεις

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η επεξεργασία και η βελτιστοποίηση Transformer μοντέλων για τη χρήση σε κινητές συσκευές. Θα μελετηθεί η επίδραση γνωστών τεχνικών βελτιστοποίησης και συμπίεσης μεγέθους στην επίδοση, στην καθυστέρηση (latency) των μοντέλων και στη συμβατότητα τους με υλικό (CPUs, GPUs). Παράλληλα θα πραγματοποιηθούν μετρήσεις σε CPU και GPU, με τη χρήση επιταχυντών και πολλαπλών νημάτων, ώστε να εξακριβωθεί η απόδοση των παραπάνω τεχνικών για διαφορετικούς τρόπους εκτέλεσης και συμπερασματολογίας.

4.1 Μέθοδοι Συμπίεσης

Για τη συμπίεση, τη μείωση του μεγέθους και του χρόνου εκτέλεσης των μοντέλων εφαρμόστηκε κβαντοποίηση στα μοντέλα που χρησιμοποιήθηκαν. Τα αρχικά μοντέλα χρησιμοποιούν ακρίβεια κινητής υποδιαστολής 32 bit (float32 / FP32) τόσο για τα βάρη όσο και για τις ενεργοποιήσεις και την είσοδο. Εφαρμόστηκαν 4 διαφορετικές μέθοδοι κβαντοποίησης:

- **Κινητής Υποδιαστολής 16 bit (float16 / FP16)**: Στην κβαντοποίηση αυτού του τύπου, τα βάρη του μοντέλου μετατρέπονται από μορφή FP32 σε FP16. Αυτό έχει ως αποτέλεσμα 2 φορές μικρότερο μέγεθος από το πρωτότυπο. Σε συγκεκριμένες συσκευές, το υλικό στηρίζει την εκτέλεση των υπολογισμών σε αυτή τη μορφή, γεγονός που επιτρέπει και επιτάχυνση σε σχέση με την εκτέλεση του αρχικού μοντέλου. Σε κάθε περίπτωση, η διαδικασία της μετατροπής μπορεί να αντιστραφεί κατά την εκτέλεση, επιτρέποντας σημαντική μείωση του χώρου αποθήκευσης με μηδαμινή αύξηση της καθυστέρησης [41].
- **Δυναμικού εύρους (Dynamic Range / DR):** Στην κβαντοποίηση αυτού του τύπου, τα βάρη του μοντέλου μετατρέπονται από μορφή FP32 σε μορφή ακεραίου 8 bit (INT8) στατικά χωρίς τη χρήση κάποιου αντιπροσωπευτικού συνόλου δεδομένου για ρύθμιση. Παράλληλα οι ενεργοποιήσεις μετατρέπονται επίσης σε INT8, αλλά με δυναμικό τρόπο βάσει του εύρους τους και της κατανομής τους για κάθε στρώμα. Η μέθοδος αυτή επιτυγχάνει 4 φορές μικρότερο μέγεθος σε σχέση με το αρχικό, και επειδή ο υπολογισμός πραγματοποιείται σε 8 bit υπάρχει σημαντική επιτάχυνση [42][.
- **Ακεραίου (Integer / INT):** Στην κβαντοποίηση αυτού του τύπου, τα βάρη και οι ενεργοποιήσεις του μοντέλου μετατρέπονται από μορφή FP32 σε μορφή ακεραίου 8 bit (INT8). Για να βρεθεί το εύρος τιμών των διανυσμάτων που θα κβαντιστούν και να ρυθμιστούν οι παράμετροι κβαντοποίησης, χρησιμοποιείται ένα αντιπροσωπευτικό σύνολο δεδομένων μεγέθους λίγων εκατοντάδων δειγμάτων, θεωρώντας δεδομένο ότι τα δείγματα αυτά ακολουθούν παρόμοια κατανομή με το συνολικό σύνολο δεδομένων. Όπου υπάρχει ανάγκη λόγω μη υλοποίησης για ακεραίους, ξαναγίνεται μετατροπή σε κινητή υποδιαστολή (float fallback). Η μέθοδος αυτή επιτυγχάνει 4 φορές μικρότερο μέγεθος και επειδή ο υπολογισμός πραγματοποιείται μόνο σε 8 bit από την αρχή μέχρι το τέλος υπάρχει σημαντική επιτάχυνση [42].

- **Πλήρως Ακεραίου (Full Integer / FULL):** Αυτός ο τύπος κβαντοποίησης μοιάζει με την Integer κβαντοποίηση, καθώς χρησιμοποιείται και εδώ αντιπροσωπευτικό σύνολο δεδομένων όμως μετατρέπονται σε ακέραιους η είσοδος και η έξοδος και δεν υπάρχει δυνατότητα χρήσης αριθμητικής κινητής υποδιαστολής, σε περίπτωση που δεν έχουν υλοποιηθεί ακέραιοι πυρήνες. Η μέθοδος αυτή επιτυγχάνει και αυτή 4 φορές μικρότερο μέγεθος και σημαντική επιτάχυνση σε σχέση με το αρχικό [42].

Με τις παραπάνω μεθόδους επιτεύχθηκε σημαντική μείωση του χώρου που απαιτείται για την αποθήκευση, που είναι καίριας σημασίας για τις κινητές συσκευές. Ο υπολογισμός γίνεται επίσης ταχύτερος, πράγμα απαραίτητο για τις αδύναμες σχετικά μονάδες επεξεργασίας των κινητών, όμως για την εκτέλεση σε μεγαλύτερο εύρος υλικού κινητών συσκευών πρέπει να γίνουν περαιτέρω βελτιστοποιήσεις [41], [42].

4.2 Μέθοδοι Βελτιστοποίησης για Συμβατότητα

Τα μοντέλα Transformers που συναντώνται κατά κύριο λόγο στις σύγχρονες εφαρμογές Επεξεργασίας Φυσικής Γλώσσας χρησιμοποιούν και τις πιο πρόσφατες εξελίξεις όσον αφορά τις συναρτήσεις ενεργοποίησης και τα διάφορα στρώματα, ώστε να επιτύχουν τις εξαιρετικά υψηλές αποδόσεις τους σε τόσο μεγάλα σύνολα δεδομένων. Συνεπώς, χρησιμοποιούν τη συνάρτηση ενεργοποίησης GELU και την μέθοδο Κανονικοποίησης Στρώματος, που αναλύθηκαν στο Κεφάλαιο 2. Ωστόσο υπάρχουν έρευνες [25] που δείχνουν ότι η χρήση των παραπάνω σε κινητές συσκευές, μπορεί να αυξήσει σημαντικά τον χρόνο εκτέλεσης, ενώ εμφανίζονται και προβλήματα συμβατότητας της Κανονικοποίησης Στρώματος με κάρτες γραφικών, στις οποίες παρατηρήθηκε απαγορευτικά χαμηλότερη ακρίβεια. Για τον λόγο αυτό προτείνονται και δοκιμάζονται μια σειρά από αντικαταστάσεις με σκοπό να διατηρηθούν οι επιδόσεις για εκτέλεση σε κάρτα γραφικών και όχι μόνο, και με βελτίωση του χρόνου εκτέλεσης.

Αντικατάσταση Κανονικοποίησης Στρώματος με Κανονικοποίηση Παρτίδας

Η Κανονικοποίηση Στρώματος (LayerNorm) εμφανίζει προβλήματα συμβατότητας με αρκετές κάρτες γραφικών και συχνά δεν υποστηρίζεται, έχοντας ως αποτέλεσμα τη μείωση της ακρίβειας. Αυτό οφείλεται στις πράξεις (αντίστροφο της τετραγωνικής ρίζας Rsqrt και τετράγωνο διαφοράς SquaredDifference), τις οποίες χρησιμοποιεί η Κανονικοποίηση Στρώματος. Είναι επίσης σχετικά αργή καθώς δεν μπορεί να εκμεταλλευτεί την παραλληλοποίηση, εφόσον εκτελείται κατά μήκους όλων των χαρακτηριστικών ενός δείγματος, ενώ δεν υπάρχει προς το παρόν βελτιστοποίηση και επιτάχυνση υλικού ειδικευμένα για τη μέθοδο αυτή.

Για τους παραπάνω λόγους δοκιμάστηκε η αντικατάσταση της με την Κανονικοποίηση Παρτίδας (Batch Normalization). Η Κανονικοποίηση Παρτίδας, είναι μια διαδικασία βελτιστοποιημένη για τις σύγχρονες πλακέτες, μπορεί να εκμεταλλευτεί την παραλληλοποίηση λόγω των πολλαπλών παρτίδων και μπορεί να πετύχει τα επιθυμητά αποτελέσματα κανονικοποίησης χωρίς σημαντική διαφορά στην απόδοση.

Αντικατάσταση GELU με άλλη συνάρτηση ενεργοποίησης

Η συνάρτηση GELU απαιτεί για την εκτέλεση της τον υπολογισμό της τετραγωνικής ρίζας και της συνάρτησης σφάλματος, πράξεις οι οποίες είναι υπολογιστικά χρονοβόρες, ειδικά σε σχέση με πιο απλές υπολογιστικά συναρτήσεις ενεργοποίησης όπως η ReLU. Επιπλέον, οι περισσότερες κάρτες γραφικών που κυκλοφορούν δεν υποστηρίζουν προς το παρόν επιτάχυνση υλικού για τη συγκεκριμένη συνάρτηση.

Για τους παραπάνω λόγους δοκιμάστηκαν 6 εναλλακτικές συναρτήσεις ενεργοποίησης, επεκτείνοντας πάνω στην ιδέα του MobileBERT για χρήση της ReLU, και μελετήθηκε η επίδραση της αντικατάστασης αυτής, τόσο στο χρόνο εκτέλεσης, όσο και στην ακρίβεια του μοντέλου. Οι συναρτήσεις που δοκιμάστηκαν είναι οι εξής: ReLU, Leaky ReLU, ReLU6, Sigmoid, Swish, Tanh. Οι παραπάνω συναρτήσεις υποστηρίζονται από τις σύγχρονες μονάδες υπολογισμού και δεν περιέχουν εξίσου περίπλοκες πράξεις.

Ξεδίπλωμα Πολλαπλασιασμού Πινάκων Παρτίδας

Ο Πολλαπλασιασμός Πινάκων (Matrix Multiplication) είναι μία πράξη Γραμμικής Άλγεβρας για την εύρεση του γινομένου δύο δισδιάστατων πινάκων A και B. Αν ο πίνακας A έχει διαστάσεις (m, n) και ο πίνακας B έχει διαστάσεις (n, p), τότε ο πίνακας C θα έχει διαστάσεις (m, p). Τα στοιχεία του πίνακα C προκύπτουν από το εσωτερικό γινόμενο των σειρών του πίνακα A με τις στήλες του πίνακα B:

$$C_{i,j} = (A \times B)_{i,j} = \sum_{k=1}^{n} A_{i,k} \cdot B_{k,j}$$

Ο Πολλαπλασιασμός Πινάκων Παρτίδας (Batch Matrix Multiplication - BatchMatMul) επεκτείνει την παραπάνω πράξη, επιτρέποντας στους δύο πίνακες εισόδου να έχουν οποιοδήποτε πλήθος διαστάσεων. Ο δισδιάστατος πολλαπλασιασμός πινάκων εφαρμόζεται κατά τα γνωστά στις δύο τελευταίες διαστάσεις. Έτσι, αν ο πίνακας A έχει διαστάσεις $(\dots, m, n)$ και ο πίνακας B έχει διαστάσεις $(\dots, n, p)$, τότε ο πίνακας C θα έχει διαστάσεις $(\dots, m, p)$.

Στους Μετασχηματιστές, επίπεδα BatchMatMul χρησιμοποιούνται για τον υπολογισμό της προσοχής πολλών κεφαλών. Χρησιμοποιείται ένα πρώτο επίπεδο BatchMatMul επίπεδο, το οποίο είναι απαραίτητο για τον πολλαπλασιασμό των πινάκων Q και K^T και ένα δεύτερο BatchMatMul επίπεδο για τον πολλαπλασιασμό με τον πίνακα V, όπως αναφέρθηκε στην υποενότητα 2.3.2. Οι Μετασχηματιστές είναι οι πρώτες αρχιτεκτονικές μοντέλων βαθιάς μάθησης που απαιτούν τη χρήση BatchMatMul επιπέδων και λόγω αυτού κατά την περίοδο παρουσίασής τους δεν υπάρχει εγγύηση για τη συμβατότητα αυτών των νέων επιπέδων με το υλικό. Επί του παρόντος, γνωστές βιβλιοθήκες Μηχανικής Μάθησης, όπως η TensorFlow Lite, επιλέγουν να αποφεύγουν τη χρήση BatchMatMul επιπέδων στα μοντέλα τους και αντί αυτών να «ξεδιπλώνουν» (unfold) τον υπολογισμό χρησιμοποιώντας μια σειρά από Πλήρως Συνδεδεμένα Επίπεδα, των οποίων η σωστή λειτουργία είναι εγγυημένη λόγω εκτενών δοκιμών σε πρότερες αρχιτεκτονικές νευρωνικών δικτύων.

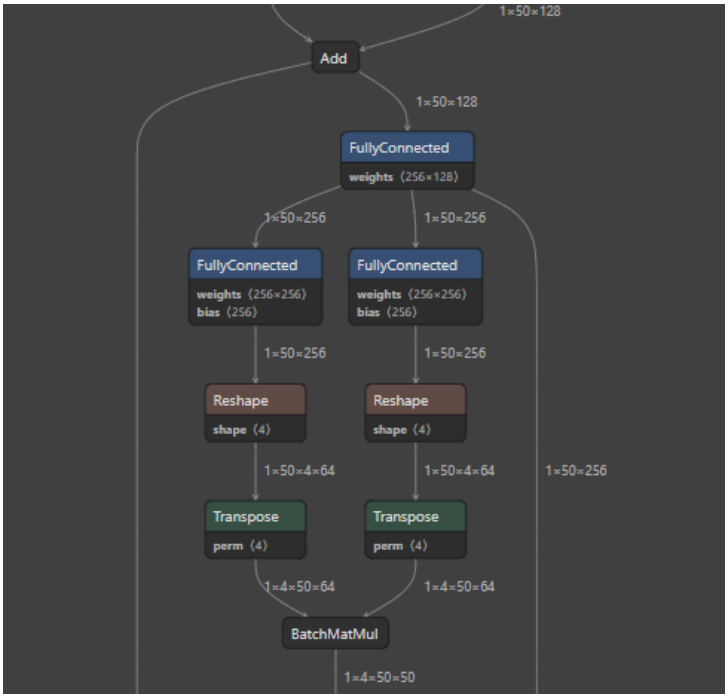
Ένα Πλήρως Συνδεδεμένο Επίπεδο μπορεί να εκτελέσει πολλαπλασιασμό πινάκων μεταξύ μιας εισόδου X αυθαίρετων διαστάσεων $(\dots, n)$ και ενός δισδιάστατου πίνακα βαρών W διαστάσεων (n, p), ώστε η έξοδος Y να έχει διαστάσεις $(\dots, p)$. Λόγω του περιορισμού των δύο διαστάσεων που πρέπει να έχουν τα βάρη, παρατηρούμε ότι για το ξεδίπλωμα ενός BatchMatMul επιπέδου στην περίπτωση των μετασχηματιστών απαιτούνται $N \times A$ πλήρως συνδεδεμένα επίπεδα, όπου N το μέγεθος παρτίδας (batch size) και A το πλήθος των κεφαλών προσοχής. Αυτός ο αριθμός μπορεί να προκύψει πολύ μεγάλος για μεγάλα πλήθη κεφαλών και μεγέθη παρτίδας και επομένως να αυξήσει σε υπερβολικό βαθμό την πολυπλοκότητα και το μέγεθος του μοντέλου.

Για παράδειγμα, αν S το μήκος της ακολουθίας εισόδου (sequence length) και H το κρυφό μήκος (hidden size), για τον πολλαπλασιασμό των Q και K^T θα ισχύει:

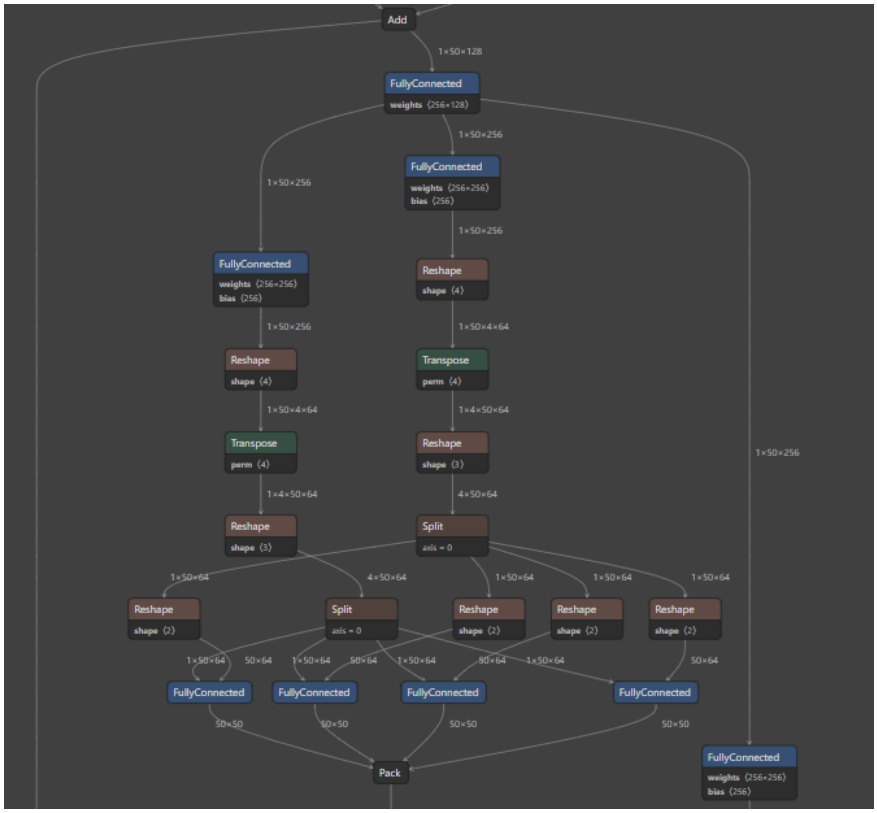
$$Q: \left(N, A, S, \frac{H}{A}\right) \longrightarrow N \times A \left(S, \frac{H}{A}\right)$$

$$K^T: \left(N, A, \frac{H}{A}, S\right) \longrightarrow N \times A \left(\frac{H}{A}, S\right)$$

Οι $N \times A$ υποπίνακες Q διαστάσεων $(S, H/A)$ θεωρούνται οι είσοδοι στα $N \times A$ πλήρως συνδεδεμένα επίπεδα και οι $N \times A$ υποπίνακες K^T διαστάσεων $(H/A\,, S)$ είναι τα βάρη τους. Στα σχήματα 4.1 και 4.2 φαίνεται το πρώτο BatchMatMul επίπεδο σε ένα από τα μπλοκ του μοντέλου ELECTRA με μέγεθος παρτίδας 1 και 4 κεφαλές προσοχής. Στην εργασία αυτή θα μελετηθεί η επίδραση της αρχικής μεθόδου (Unfolded μοντέλα) και της πιο συμπαγούς προσέγγισης (Folded μοντέλα) στην ακρίβεια και τον χρόνο εκτέλεσης των Μετασχηματιστών, καθώς και η συμβατότητα τους με το υλικό κινητών συσκευών [43].



Σχήμα 4.1: Το μοντέλο ELECTRA$_{\text{SMALL}}$ με 4 Κεφαλές Προσοχής, όταν χρησιμοποιεί BatchMatMul Folding στην TFLite μορφή



Σχήμα 4.2: Το μοντέλο ELECTRA με 4 Κεφαλές Προσοχής, όταν χρησιμοποιεί BatchMatMul Unfolding στην TFLite μορφή

Κεφάλαιο 5ο

Πειραματική Διάταξη

Παρακάτω παρατίθεται λεπτομερής περιγραφή του περιβάλλοντος υλοποίησης και αξιολόγησης των Transformer μοντέλων και των μεθόδων συμπίεσης και βελτιστοποίησης τους, οι οποίες παρουσιάστηκαν στο προηγούμενο Κεφάλαιο. Θα αναφερθούν οι τεχνολογίες, τα σύνολα δεδομένων που χρησιμοποιήθηκαν, η μεθοδολογία εκπαίδευσης, οι αρχιτεκτονικές των μοντέλων, καθώς και οι μετρικές, βάσει των οποίων θα γίνει η αξιολόγηση.

5.1 Τεχνολογίες

5.1.1 TensorFlow και TensorFlow Lite

Το TensorFlow είναι μία ολοκληρωμένη, δωρεάν βιβλιοθήκη ανοιχτού κώδικα για την ανάπτυξη εφαρμογών Μηχανικής Μάθησης και Τεχνητής Νοημοσύνης. Δημιουργήθηκε από την ομάδα της Google και κυκλοφόρησε δημόσια το 2016. Παρέχει μία σειρά από εργαλεία, μεθόδους και πόρους για την προώθηση της τελευταίας λέξης της τεχνολογίας στη Μηχανική Μάθηση. Μέσω του TensorFlow μπορεί κανείς να δημιουργήσει, να εκπαιδεύσει, να τροποποιήσει και να τεστάρει μοντέλα και εφαρμογές Μηχανικής Μάθησης, με ιδιαίτερη έμφαση στη Βαθιά Μάθηση. Διαθέτει τη διεπαφή Keras σε γλώσσα Python, που αποτελεί ένα API με σκοπό τη διευκόλυνση της χρήσης του TensorFlow από τον άνθρωπο. Το Keras προσφέρει τα βασικά εργαλεία και στοιχεία των Νευρωνικών Δικτύων, όπως είναι τα στρώματα, οι ενεργοποιήσεις και οι μετρικές σε συμπαγή μορφή, ώστε να χρησιμοποιούνται σαν δομικά στοιχεία και να απλοποιείται ο προγραμματισμός από το χρήστη. Το TensorFlow και το Keras ανανεώνονται διαρκώς, ώστε να συμβαδίζουν με τις πιο πρόσφατες τεχνολογικές εξελίξεις στον τομέα της Μηχανικής Μάθησης.

Το TensorFlow προσφέρει επίσης το TensorFlow Lite, το οποίο είναι μία ολοκληρωμένη πλατφόρμα, που προσφέρει τη δυνατότητα εκτέλεσης και συμπερασματολογίας μοντέλων TensorFlow σε κινητές συσκευές και άλλες συσκευές παρυφών, όπως είναι οι μικροελεγκτές. Οι κύριες δομές του είναι οι εξής:

1. TensorFlow Lite Μετατροπέας (Converter): Ο μετατροπέας δίνει τη δυνατότητα μετατροπής των μοντέλων TensorFlow σε FlatBuffers, μία μορφή με μικρότερο μέγεθος και ταχύτερη εκτέλεση (.tflite), διατηρώντας τη δομή και τις ιδιότητες τους ώστε να καθίσταται δυνατή η χρήση σε συσκευές με περιορισμένους υπολογιστικούς πόρους και μνήμη. Επιπλέον, μπορεί να εισάγει βελτιστοποιήσεις όπως κβαντοποίηση και κλάδεμα παραμέτρων.
2. TensorFlow Lite Διερμηνέας (Interpreter): Ο διερμηνέας εκτελεί τα TFlite μοντέλα σε μία σειρά από πιθανές συσκευές. Ταυτόχρονα διασφαλίζει κατά την εκτέλεση την βέλτιστη χρήση πόρων, αναθέτοντας κομμάτια του υπολογισμού σε επιταχυντές, όπως XNNPack, GPU, κ.α. για βέλτιστη απόδοση [44]–[47].

5.1.2 Hugging Face

Η Hugging Face είναι μία Γαλλοαμερικανική εταιρία με έδρα τη Νέα Υόρκη που δραστηριοποιείται στον τομέα της Επεξεργασίας Φυσικής Γλώσσας. Η πορεία της ξεκίνησε το 2016, με την ανάπτυξη μίας εφαρμογής συνομιλίας για εφήβους. Από το 2019 και ύστερα, έχει κάνει προσπάθεια να δημιουργήσει μία βιβλιοθήκη μοντέλων Επεξεργασίας Φυσικής Γλώσσας, ανοιχτή και εύκολη για χρήση από το κοινό. Ο σκοπός είναι η ανάπτυξη και δημοκρατικοποίηση του NLP, καθώς η εταιρία θεωρεί ότι ο τομέας αυτός ανοίγει νέους ορίζοντες στην επικοινωνία ανθρώπου και μηχανής και μπορεί να έχει εξαιρετικά ωφέλιμες εφαρμογές στην καθημερινότητα.

Η βιβλιοθήκη Transformers, που προέκυψε από την προσπάθεια αυτή, είναι εξαιρετικά δημοφιλής και παρέχει χιλιάδες διαφορετικά NLP μοντέλα και αρχιτεκτονικές για κάθε είδους εργασία. Υποστηρίζεται από τις τρεις κυριότερες βιβλιοθήκες Βαθιάς Μάθησης, τις PyTorch, TensorFlow και Jax. Τα Transformers μοντέλα μπορούν να λειτουργήσουν αυτοτελώς, καθώς είναι προεκπαιδευμένα ή να υποστούν επεξεργασία και fine-tuning ώστε να επιλύσουν άλλες εργασίες. Τέλος προσφέρεται η δυνατότητα χρήσης, λήψης και συμπερασματολογίας των μοντέλων σε μία σειρά από εφαρμογές μέσω API για ακόμη πιο εύκολη πρόσβαση [48].

5.1.3 Android Studio

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Ανακοινώθηκε το 2013 από τη Google και κυκλοφόρησε για πρώτη φορά το 2014. Είναι βασισμένο στον επεξεργαστή κώδικα και τα εργαλεία της Intellij IDEA. Η ανάπτυξη των εφαρμογών μπορεί να γίνει σε γλώσσα προγραμματισμού Java ή Kotlin. Προσφέρει πληθώρα δυνατοτήτων στον προγραμματιστή, συμπεριλαμβανομένου μιας διεπαφής για επεξεργασία και οπτικοποίηση του κώδικα, ενός εύχρηστου προσομοιωτή Android συσκευών, γρήγορη διασύνδεση με το GitHub, κ.α. [49].

5.2 Αρχιτεκτονικές Transformer Μοντέλων

5.2.1 BERT

Αρχικά πρέπει να γίνει μία σύντομη αναφορά στο μοντέλο BERT. Το BERT (Bidirectional Encoder Representations from Transformers) παρουσιάστηκε το 2019 από τους Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova στην εργασία “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” [31]. Λόγω του μεγάλου μεγέθους του, είναι δύσκολο να γίνει συμπερασματολογία του BERT σε κινητές συσκευές και για το λόγο αυτό δεν χρησιμοποιήθηκε στην παρούσα διπλωματική, όμως αποτελεί την αρχιτεκτονική-βάση για τα μοντέλα τα οποία χρησιμοποιήθηκαν. Το BERT έφερε επανάσταση στον τομέα του NLP, πετυχαίνοντας εξαιρετικά υψηλές επιδόσεις σε σχέση με τα μοντέλα και τις αρχιτεκτονικές που επικρατούσαν μέχρι τότε. Χρησιμοποιεί αμφίδρομη εκπαίδευση του Transformer, ώστε να αξιοποιεί ταυτόχρονα περιεχόμενο τόσο από τα δεξιά όσο και από τα αριστερά του κειμένου. Με τον τρόπο αυτό το μοντέλο μπορεί να δημιουργήσει πιο περίπλοκες γλωσσικές σχέσεις βάσει του περιεχομένου που του δίνεται, σε σύγκριση με τα μοντέλα τα οποία έχουν εκπαιδευτεί από τα

δεξιά προς τα αριστερά ή με συνδυασμό εκπαίδευσης και στις δύο κατευθύνσεις. Έτσι το μοντέλο μπορεί να μετεκπαιδευτεί, για εξειδίκευση σε κάποια συγκεκριμένη εργασία όπως η Απάντηση Ερωτήσεων ή η Κατανόηση Κειμένου προσθέτοντας απλά ένα κατάλληλο στρώμα εξόδου. Επειδή το BERT προορίζεται για κατανόηση γλώσσας μελετώντας τα συμφραζόμενα και όχι για παραγωγή κειμένου, χρησιμοποιεί μόνο τον Κωδικοποιητή (Encoder) της αρχιτεκτονικής των Μετασχηματιστών και δεν απαιτείται ο Αποκωδικοποιητής (Decoder). Κατά την εκπαίδευση χρησιμοποιεί δύο στρατηγικές:

- **Μοντελοποίηση Γλώσσας με Συγκαλύψεις (Masked Language Modeling - MLM)**: Στην Μοντελοποίηση Γλώσσας με Συγκαλύψεις ένα ποσοστό των λέξεων «καλύπτεται» με ένα διακριτικό [MASK]. Το μοντέλο αποπειράται να βρει την αρχική τιμή των λέξεων που καλύφθηκαν, χρησιμοποιώντας ως πληροφορία εισόδου το περιεχόμενο των υπόλοιπων λέξεων. Συγκεκριμένα στο BERT το ποσοστό των λέξεων το οποίο καλύπτεται είναι το 15% και η διεργασία αυτή αποτελεί το πρώτο στάδιο εκπαίδευσης.
- **Πρόβλεψη Επόμενης Πρότασης (Next Sentence Prediction - NSP):** Στην Πρόβλεψη Επόμενης Πρότασης δίνονται στο μοντέλο ως είσοδος ζεύγη προτάσεων και το μοντέλο καλείται να προβλέψει εάν η δεύτερη πρόταση είναι συνέχεια της πρώτης. Στο BERT στο 50% των ζευγών οι προτάσεις είναι συνέχεια η μία της άλλης, ενώ στο άλλο 50% η δεύτερη πρόταση έχει επιλεχθεί τυχαία από το αρχικό κείμενο.

Το BERT εκπαιδεύεται με στόχο την ελαχιστοποίηση της συνάρτησης απώλειας και για τις δύο στρατηγικές εκπαίδευσης παράλληλα. Η βασική εκδοχή του BERT, το $\text{BERT}_{\text{BASE}}$ έχει 12 στρώματα transformers με 12 κεφαλές αυτοπροσοχής, κρυφό μέγεθος 768 και περίπου 110 εκατομμύρια παραμέτρους συνολικά, ενώ υπάρχει και η μεγαλύτερη εκδοχή το $\text{BERT}_{\text{LARGE}}$. Στη συνέχεια θα δούμε κάποιες αρχιτεκτονικές, οι οποίες βασίστηκαν στο μοντέλο BERT αλλά εφάρμοσαν μεθόδους συμπίεσης ή εκπαίδευσης για να επιτύχουν ακόμη καλύτερα αποτελέσματα όσον αφορά την ακρίβεια ή το μέγεθος [31].

5.2.2 MobileBERT

Το MobileBERT, το οποίο αναφέρθηκε και στο κεφάλαιο 3, παρουσιάστηκε το 2020 από τους Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, και Denny Zhou [50]. Σχεδιάστηκε με σκοπό τη δημιουργία ενός ταχύτερου και ελαφρύτερου μοντέλου Transformer, βασισμένο στο BERT. Όπως και το πρωτότυπο μοντέλο, είναι “task-agnostic”, δηλαδή μπορεί να χρησιμοποιηθεί για πληθώρα από tasks Επεξεργασίας Φυσικής Γλώσσας, με απλό fine-tuning και επανεκπαίδευση. Το MobileBERT είναι μία ελαφρύτερη εκδοχή του $\text{BERT}_{\text{LARGE}}$ με σημεία συμφόρησης και μία προσεκτικά σχεδιασμένη ισορροπία μεταξύ των δικτύων αυτό-προσοχής και εμπρόσθιας τροφοδότησης. Εκπαιδεύτηκε με απόσταξη και μεταφορά γνώσης από ένα μοντέλο «δάσκαλο», που ήταν στην ουσία ένα μοντέλο $\text{BERT}_{\text{LARGE}}$ με αντίστροφη συμφόρηση. Οι επιδόσεις του στο GLUE Benchmark πλησιάζουν και σε κάποιες περιπτώσεις ξεπερνούν αυτές του $\text{BERT}_{\text{BASE}}$, με μέγεθος μικρότερο κατά 4.3 φορές και ταχύτερο χρόνο εκτέλεσης κατά 5.5 φορές. Όπως και το πρωτότυπο BERT, συνίσταται να χρησιμοποιείται για πρόβλεψη κρυμμένων tokens και για Κατανόηση Φυσικής Γλώσσας (Natural Language Understanding NLU), αντί για tasks Παραγωγής Κειμένου (Text Generation).

Το MobileBERT υιοθετεί, πέρα από την ειδικευμένη για ταχύτητα και μείωση μεγέθους αρχιτεκτονική του, και μία σειρά από αντικαταστάσεις και τεχνικές βελτιστοποίησης. Η Κανονικοποίηση Στρώματος (LayerNorm) αντικαθίσταται με τη NoNorm, μία προσαρμοσμένη από τους ερευνητές του MobileBERT μέθοδο κανονικοποίησης και η συνάρτηση ενεργοποίησης GELU αντικαθίσταται από την απλή συνάρτηση ενεργοποίησης ReLU. Οι αντικαταστάσεις της Κανονικοποίησης Στρώματος και της συνάρτησης ενεργοποίησης GELU με άλλες απλούστερες μεθόδους και συναρτήσεις δοκιμάστηκαν και στις επόμενες αρχιτεκτονικές και για το λόγο αυτό το MobileBERT θα αποτελεί σημείο αναφοράς για τα υπόλοιπα μοντέλα στην παρούσα διπλωματική εργασία [40], [50].

Παρακάτω στον πίνακα 5.1 παρατίθενται τα αποτελέσματα του MobileBERT σε σχέση με άλλα μοντέλα στο σημείο αναφοράς GLUE, όπως αυτά δίνονται στην εργασία παρουσίασης του μοντέλου. Ο αριθμός κάτω από κάθε task υποδηλώνει το πλήθος των εισόδων κατά την εκπαίδευση και το σύμβολο «*» υποδηλώνει την ύπαρξη fine-tuned εξειδικευμένου μοντέλου.

	#Params	#FLOPS	Latency	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m/mm	QNLI	RTE	GLUE
				8.5k	67k	3.7k	5.7k	364k	393k	108k	2.5k	
ELMo-BiLSTM-Attn	-	-	-	33.6	90.4	84.4	72.3	63.1	74.1/74.5	79.8	58.9	70.0
OpenAI GPT	109M	-	-	47.2	93.1	87.7	84.8	70.1	80.7/80.6	87.2	69.1	76.9
$BERT_{BASE}$	109M	22.5B	342 ms	**52.1**	**93.5**	**88.9**	**85.8**	71.2	**84.6/83.4**	90.5	66.4	78.3
$BERT_{BASE}$-6L-PKD*	66.5M	11.3B	-	-	92.0	85.0	-	70.7	81.5/81.0	89.0	65.5	-
$BERT_{BASE}$-4L-PKD†*	52.2M	7.6B	-	24.8	89.4	82.6	79.8	70.2	79.9/79.3	85.1	62.3	-
$BERT_{BASE}$-3L-PKD*	45.3M	5.7B	-	-	87.5	80.7	-	68.1	76.7/76.3	84.7	58.2	-
$DistilBERT_{BASE}$-6L†	62.2M	11.3B	-	-	92.0	85.0		70.7	81.5/81.0	89.0	65.5	-
$DistilBERT_{BASE}$-4L†	52.2M	7.6B	-	32.8	91.4	82.4	76.1	68.5	78.9/78.0	85.2	54.1	-
TinyBERT*	14.5M	1.2B	-	43.3	92.6	86.4	79.9	**71.3**	82.5/81.8	87.7	62.9	75.4
$MobileBERT_{TINY}$	15.1M	3.1B	40 ms	46.7	91.7	87.9	80.1	68.9	81.5/81.6	89.5	65.1	75.8
MobileBERT	25.3M	5.7B	62 ms	50.5	92.8	88.8	84.4	70.2	83.3/82.6	90.6	66.2	77.7
MobileBERT w/o OPT	25.3M	5.7B	192 ms	51.1	92.6	88.8	84.8	70.5	84.3/**83.4**	**91.6**	**70.4**	**78.5**

Πίνακας 5.1: Το MobileBERT στο σημείο αναφοράς GLUE [50]

5.2.3 ELECTRA

Το μοντέλο ELECTRA προτάθηκε στην εργασία με τίτλο " ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators", των Kevin Clark, et al το 2020 [51]. Εισήγαγε μία νέα προσέγγιση προεκπαίδευσης, στην οποία η εκπαίδευση γίνεται σε δύο Transformer μοντέλα: τη γεννήτρια (generator) και τον διαχωριστή (discriminator). Η γεννήτρια αντικαθιστά tokens σε μία ακολουθία, ώστε να εκπαιδευτεί σαν γλωσσικό μοντέλο με συγκαλύψεις (Masked Language Model – MLM). Τα μοντέλα αυτού του είδους εκπαιδεύονται κρύβοντας μέρος της εισόδου με το ειδικό token [MASK] και προσπαθώντας στη συνέχεια να προβλέψουν την αρχική τιμή των καλυμμένων λέξεων χρησιμοποιώντας την υπόλοιπη ακολουθία. Παράλληλα ο διαχωριστής προσπαθεί να αναγνωρίσει ποια tokens αντικαταστάθηκαν από τη γεννήτρια.

Οι δημιουργοί του ELECTRA έκαναν τη διαπίστωση ότι οι MLM μέθοδοι εκπαίδευσης, έχουν υψηλή απόδοση αλλά χρειάζονται μεγάλα σύνολα δεδομένων. Για το λόγο αυτό πρότειναν τη μέθοδο της ανίχνευσης tokens (token detection), στην οποία το μοντέλο προσπαθεί να αναγνωρίσει ποια tokens έχουν αντικατασταθεί, αντί να κρύβει την είσοδο και να εκπαιδεύεται

προσπαθώντας να μαντέψει τα tokens εισόδου. Η μελέτη τους καταλήγει ότι αυτή η μέθοδος εκπαίδευσης είναι πιο αποτελεσματική από τη μέθοδο των MLM, καθώς ο διαχωριστής μαθαίνει κοιτώντας όλη την είσοδο, αντί να περιορίζεται στο μικρό υποσύνολο το οποίο κρύφτηκε. Η προσέγγιση αυτή είναι ιδιαίτερα αποδοτική για μοντέλα μικρού μεγέθους και ξεπέρασε πολύ μεγαλύτερα μοντέλα σε επίδοση και ταχύτητα εκτέλεσης, αλλά μπορεί να κλιμακωθεί και για μεγαλύτερα μοντέλα.

Το ELECTRA είναι στην ουσία μία μέθοδος προεκπαίδευσης, πάνω στην υπάρχουσα αρχιτεκτονική του BERT, χωρίς ουσιαστικές διαφορές, πέρα από το μέγεθος των εμφυτευμάτων (embedding size), που είναι συνήθως μικρότερο, και την κρυφή διάσταση (hidden size), που είναι μεγαλύτερη και σκοπεύει στην βελτίωση της επίδοσης του αρχικού μοντέλου [51], [52].

Παρακάτω στον πίνακα 5.2 παρατίθενται τα αποτελέσματα του ELECTRA σε σχέση με άλλα μικρά μοντέλα στο σημείο αναφοράς GLUE, όπως αυτά δίνονται στην εργασία παρουσίασης του μοντέλου.

Model	Train / Infer FLOPs	Speedup	Params	Train Time + Hardware	GLUE
ELMo	3.3e18 / 2.6e10	19x / 1.2x	96M	14d on 3 GTX 1080 GPUs	71.2
GPT	4.0e19 / 3.0e10	1.6x / 0.97x	117M	25d on 8 P6000 GPUs	78.8
BERT-Small	1.4e18 / 3.7e9	45x / 8x	14M	4d on 1 V100 GPU	75.1
BERT-Base	6.4e19 / 2.9e10	1x / 1x	110M	4d on 16 TPUv3s	82.2
ELECTRA-Small	1.4e18 / 3.7e9	45x / 8x	14M	4d on 1 V100 GPU	79.9
50% trained	7.1e17 / 3.7e9	90x / 8x	14M	2d on 1 V100 GPU	79.0
25% trained	3.6e17 / 3.7e9	181x / 8x	14M	1d on 1 V100 GPU	77.7
12.5% trained	1.8e17 / 3.7e9	361x / 8x	14M	12h on 1 V100 GPU	76.0
6.25% trained	8.9e16 / 3.7e9	722x / 8x	14M	6h on 1 V100 GPU	74.1
ELECTRA-Base	6.4e19 / 2.9e10	1x / 1x	110M	4d on 16 TPUv3s	85.1

Πίνακας 5.2: Το ELECTRA στο σημείο αναφοράς GLUE [51]

5.2.4 DistilBERT

Το DistilBERT παρουσιάστηκε στην εργασία “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter” από τους Victor Sanh, Lysandre Debut, Julien Chaumond και Thomas Wolf το 2020 [53] με σκοπό την έρευνα για τη λειτουργία μοντέλων Επεξεργασίας Φυσικής Γλώσσας στις παρυφές ή/και σε περιβάλλοντα με περιορισμένους πόρους. Είναι μία μικρότερη, πιο γρήγορη και ελαφριά έκδοση του BERT, που εκπαιδεύτηκε με απόσταξη γνώσης από το μοντέλο $\text{BERT}_{\text{BASE}}$. Η διαφορά του με άλλες προσπάθειες χρήσης απόσταξης γνώσης με «δάσκαλο» το BERT είναι ότι η μεταφορά γνώσης γίνεται στη φάση της προεκπαίδευσης, και το μοντέλο μπορεί στη συνέχεια να εξειδικευτεί σε πληθώρα άλλων tasks με fine-tuning. Για την καλύτερη αξιοποίηση των επαγωγικών μεροληψιών, τις οποίες έχουν μάθει τα μεγαλύτερα μοντέλα μέσω της εκπαίδευσης τους, το DistilBERT χρησιμοποιεί ένα σύστημα τριπλής απώλειας, που συνδυάζει τη μοντελοποίηση γλώσσας, την απόσταξη και την απώλεια συνημιτονικής απόστασης. Το μοντέλο που προκύπτει έχει μικρότερο μέγεθος από το πρωτότυπο BERT κατά 40%, με το 97% των δυνατοτήτων, και είναι κατά 60% ταχύτερο στην εκτέλεση του [53], [54].

Παρακάτω στον πίνακα 5.3 παρατίθενται τα αποτελέσματα του DistilBERT σε σχέση με το πρωτότυπο μοντέλο BERT και το ELMo στο σημείο αναφοράς GLUE, όπως αυτά δίνονται στην εργασία παρουσίασης του μοντέλου. Το ELMo είναι ένας πρωτοπόρος τρόπος αναπαράστασης λέξεων σε εμφυτεύματα.

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Πίνακας 5.3: Το DistilBERT στο σημείο αναφοράς GLUE [53]

5.2.5 RoBERTa

Το μοντέλο RoBERTa παρουσιάστηκε στην εργασία με τίτλο "RoBERTa: A Robustly Optimized BERT Pretraining Approach", από τους Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer και Veselin Stoyanov το 2019 [55]. Η έρευνα αυτή κατέληξε, μέσω μίας προσεκτικής μελέτης της επίδρασης των κύριων υπερπαραμέτρων και του μεγέθους του συνόλου δεδομένων εκπαίδευσης, στο συμπέρασμα ότι το πρωτότυπο μοντέλο BERT είναι υποεκπαιδευμένο, και ότι έχει τη δυνατότητα να επιτύχει πολύ υψηλότερες επιδόσεις, τις οποίες το RoBERTa στοχεύει να επιτύχει.

Το RoBERTa εκπαιδεύτηκε με ένα σύνολο δεδομένων 160 GB κειμένου, που είναι παραπάνω από 10 φορές μεγαλύτερο από το σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση του BERT, και αφιερώθηκε πολύ περισσότερος χρόνος στη διαδικασία αυτή. Η αρχιτεκτονική του είναι σχεδόν ίδια με αυτή του BERT με 12 επίπεδα, κρυφό μέγεθος 768 και 12 κεφαλές αυτό-προσοχής. Έγιναν όμως και μία σειρά από αλλαγές στη διαδικασία εκπαίδευσης και στην αρχιτεκτονική. Αρχικά, το μοντέλο δεν εκπαιδεύτηκε με το στόχο της Πρόβλεψης της Επόμενης Πρότασης, καθώς παρατηρήθηκε ότι η αφαίρεση αυτού του στόχου, βελτιώνει την ικανότητα γενίκευσης του μοντέλου. Επιπλέον, το RoBERTa εκπαιδεύτηκε με μεγαλύτερες ακολουθίες εισόδου, σε μεγαλύτερου μεγέθους παρτίδες. Τέλος, έγινε χρήση δυναμικού masking για tokens, ώστε να αποφευχθεί η στατικότητα και να μαθαίνει το μοντέλο διαφορετικά πρότυπα.

Οι επιδόσεις του βελτιστοποιημένου μοντέλου είναι εφάμιλλες με άλλα σύγχρονα και βελτιστοποιημένα μοντέλα. Παρακάτω στον πίνακα 4.4 παρατίθενται τα αποτελέσματα του RoBERTa σε σχέση με άλλα μοντέλα εφάμιλλου μεγέθους [55]–[57].

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-
Ensembles on test (from leaderboard as of July 25, 2019)										
ALICE	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8	86.3
MT-DNN	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2/89.8	98.6	90.3	86.3	**96.8**	**93.0**	67.8	91.6	**90.4**	88.4
RoBERTa	**90.8/90.2**	**98.9**	90.2	**88.2**	96.7	92.3	67.8	**92.2**	89.0	**88.5**

Πίνακας 5.4: Το RoBERTa στο σημείο αναφοράς GLUE [55]

5.2.6 MiniLM

Το μοντέλο MiniLM παρουσιάστηκε στην εργασία με τίτλο “MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers” των Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang και Ming Zhou το 2020 [58], η οποία έθεσε τα θεμέλια για την απόσταξη γνώσης μεγάλων γλωσσικών μοντέλων. Εισήγαγε μία νέα προσέγγιση απόσταξης, που διέφερε από προηγούμενες προσπάθειες (DistillBERT, MobileBERT, κ.α.). Αρχικά προτάθηκε η χρήση ενός ενδιάμεσου «δασκάλου-βοηθού» σε περιπτώσεις που ο «δάσκαλος» έχει περισσότερα στρώματα από τον «μαθητή». Χαρακτηριστικά, αν ο «δάσκαλος» και ο «μαθητής» έχουν L και M στρώματα και κρυφό μέγεθος d και d_i αντίστοιχα, παρατίθεται παρακάτω στο σχήμα 5.1 η προσέγγιση απόσταξης, για $M \leq \frac{1}{2}L$ και $d_i \leq \frac{1}{2}d$.

Σχήμα 5.1: Η προσέγγιση απόσταξης με τη χρήση ενδιάμεσου δασκάλου [58]

Τα αποτελέσματα της μεθόδου αυτής είναι εντυπωσιακά, σε αρκετές περιπτώσεις καλύτερα από τα πιο σύγχρονα μοντέλα, ενώ διατηρείται το 99% της ακρίβειας για τα κυριότερα Benchmarks, με το μισό μέγεθος του μοντέλου «δασκάλου». Παρακάτω στον πίνακα 5.5 παρατίθενται τα αποτελέσματα του MiniLM σε σχέση με άλλα μοντέλα απόσταξης γνώσης, όπως παρατίθενται στην εργασία παρουσίασης του μοντέλου [58], [59].

Model	#Param	SQuAD2	MNLI-m	SST-2	QNLI	CoLA	RTE	MRPC	QQP	Average
$\text{BERT}_{\text{BASE}}$ [12] (teacher)	109M	76.8	84.5	93.2	91.7	58.9	68.6	87.3	91.3	81.5
$\text{BERT}_{\text{SMALL}}$ [41]	66M	73.2	81.8	91.2	89.8	53.5	67.9	84.9	90.6	79.1
Truncated $\text{BERT}_{\text{BASE}}$ [12]	66M	69.9	81.2	90.8	87.9	41.4	65.5	82.7	90.4	76.2
DistilBERT [35]	66M	70.7	82.2	91.3	89.2	**51.3**	59.9	87.5	88.5	77.6
TinyBERT [20]	66M	73.1	83.5	91.6	90.5	42.8	**72.2**	**88.4**	90.6	79.1
MINILM	66M	**76.4**	**84.0**	**92.0**	**91.0**	49.2	71.5	**88.4**	**91.0**	**80.4**

Πίνακας 5.5: Το MiniLM στο σημείο αναφοράς SQuAD2 και σε tasks του GLUE [58]

5.2.7 XtremeDistil

Το τελευταίο μοντέλο που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία είναι το XtremeDistil. Το μοντέλο αυτό παρουσιάστηκε στην εργασία "XtremeDistilTransformers: Task Transfer for Task-agnostic Distillation" το 2021 [60] από την ομάδα της Microsoft. Βασίζεται σε μεθόδους συμπίεσης και απόσταξης γνώσης από προηγούμενες έρευνες (συμπεριλαμβανομένου και αυτών του MiniLM), οι οποίες δημιούργησαν αποδοτικά και συμπαγή task-specific μοντέλα, ώστε να προκύψει ένα πλαίσιο απόσταξης για task-agnostic μοντέλα, με ισχυρή δυνατότητα γενίκευσης και μεταφοράς από ένα task σε ένα άλλο.

Για τη δημιουργία του πλαισίου αυτού μελετήθηκε το κατά πόσο μεταφέρεται η γνώση σε μία σειρά από πηγαία tasks, ώστε να βρεθούν τα βέλτιστα για μεταφορά γνώσης. Στη συνέχεια έγινε το ίδιο για διάφορες αρχιτεκτονικές. Το πλαίσιο που προκύπτει προσφέρει υψηλή δυνατότητα συμπίεσης για απόσταξη συγκεκριμένων tasks και μπορεί να έχει ευρεία εφαρμογή για συνολικά μοντέλα πολλών tasks. Πέραν αυτού μελετώνται και παρουσιάζονται τεχνικές επαύξησης των υπάρχοντων δεδομένων για τη βελτίωση της απόδοσης.

Παρακάτω στον πίνακα 5.6 παρατίθενται τα αποτελέσματα του XtremeDistil σε σχέση με άλλα συμπιεσμένα μοντέλα και μοντέλα απόσταξης γνώσης, όπως παρατίθενται στην εργασία παρουσίασης του μοντέλου.

Models	Params	Speedup	MNLI	QNLI	QQP	RTE	SST	MRPC	SQuADv2	Avg
BERT (R)	109	1x	84.5	91.7	91.3	68.6	93.2	87.3	76.8	84.8
BERT-Trun (R)	66	2x	81.2	87.9	90.4	65.5	90.8	82.7	69.9	81.2
DistilBERT (R)	66	2x	82.2	89.2	88.5	59.9	91.3	87.5	70.7	81.3
TinyBERT (R)	66	2x	83.5	90.5	90.6	72.2	91.6	88.4	73.1	84.3
MiniLM (R)	66	2x	84.0	91.0	91.0	71.5	92.0	88.4	76.4	84.9
MiniLM (R)	22	5.3x	82.8	90.3	90.6	68.9	91.3	86.6	72.9	83.3
BERT (HF)	109	1x	84.4	91.4	91.2	66.8	93.2	83.8	74.8	83.7
MiniLM (HF)	22	5.3x	82.7	89.4	90.3	64.3	90.8	84.1	71.5	81.9
XtremeDistilTransf. (HF)	**22**	**5.3x**	**84.5**	**90.2**	**90.4**	**77.3**	**91.6**	**89.0**	**74.4**	**85.3**
XtremeDistilTransf. (HF)	14	9.4x	81.8	86.9	89.5	74.4	89.9	86.5	63.0	81.7

Πίνακας 5.6: Το XtremeDistil σε tasks του σημείου αναφοράς GLUE και στο σημείο αναφοράς SQuADv2 [60]

Παρακάτω παρατίθενται στον πίνακα 5.7 τα μοντέλα τα οποία αναφέρθηκαν καθώς και αυτά που χρησιμοποιήθηκαν (με bold) στην παρούσα εργασία:

Model	Params
BERT_BASE	110M
DistilBERT	66M
MiniLMv1-L12-H384	33.19M
RoBERTa_TINY	27.75M
Xdistil-L12-H384	33.19M
ELECTRA_SMALL	14M
MobileBERT	25.3M

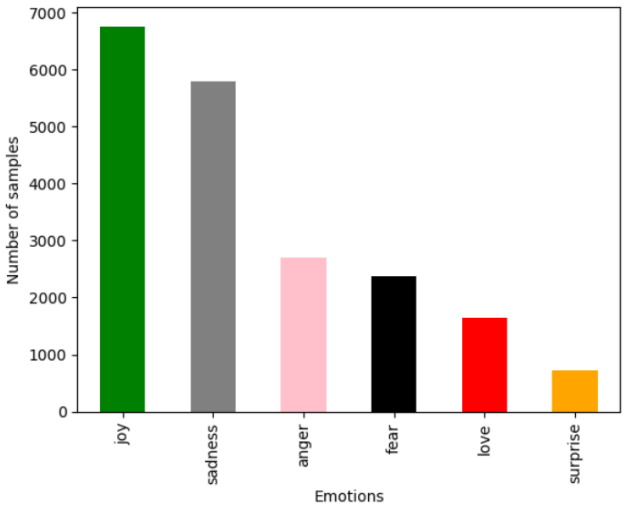
Πίνακας 5.7: Τα μοντέλα που αναφέρθηκαν και που χρησιμοποιήθηκαν στην παρούσα εργασία

Οι μέθοδοι που αναλύθηκαν παράγουν μοντέλα, των οποίων το μέγεθος είναι συνήθως απαγορευτικό για χρήση σε κινητές συσκευές, ιδίως το RoBERTa και το ELECTRA, καθώς σχεδιάστηκαν με στόχο την αύξηση της ακρίβειας και όχι με γνώμονα το μικρότερο μέγεθος. Εξαιτίας αυτού για τα RoBERTa και ELECTRA χρησιμοποιήθηκαν οι υπάρχουσες εκδοχές τους $\text{RoBERTa}_{\text{TINY}}$ και $\text{ELECTRA}_{\text{SMALL}}$ αντίστοιχα που έχουν λιγότερα στρώματα, κεφαλές αυτό-προσοχής, και παραμέτρους, ώστε να είναι μικρότερα σε μέγεθος και πιο συμπαγή ενώ για τα μοντέλα Xdistil και MiniLM επίσης χρησιμοποιήθηκαν μικρότερες εκδόσεις από τις αρχικές. Οι υπόλοιπες αρχιτεκτονικές είναι έως έναν βαθμό βελτιστοποιημένες, είτε μέσω της χρήσης απόσταξης γνώσης (DistilBERT), είτε με άλλες μεθόδους μείωσης μεγέθους και υπολογιστικού φόρτου (MobileBERT) επομένως χρησιμοποιήθηκαν ως έχουν.

5.3 Σύνολα Δεδομένων

Τα μοντέλα που χρησιμοποιήθηκαν για την πραγματοποίηση του πειραματικού μέρους της παρούσας εργασίας αντλήθηκαν από τη βιβλιοθήκη transformers του Hugging Face. Τα μοντέλα αυτά είναι προεκπαιδευμένα είτε σε κάποιο συγκεκριμένο task, είτε σε ένα εύρος από tasks, όπως η Κατανόηση Φυσικής Γλώσσας, πάνω σε πολύ μεγάλα σύνολα δεδομένων κειμένου, όπως αναλύθηκε στην προηγούμενη ενότητα για κάθε αρχιτεκτονική. Επιλέχθηκε να γίνει επανεκπαίδευση και fine-tuning των μοντέλων που χρησιμοποιήθηκαν, ώστε να εξειδικευτούν στο task της Κατηγοριοποίησης Συναισθημάτων. Συγκεκριμένα χρησιμοποιήθηκε το σύνολο δεδομένων Emotions Dataset for NLP [61], το οποίο περιέχει προτάσεις στα αγγλικά και την ετικέτα του συναισθήματος τους. Κάθε πρόταση μπορεί να έχει ταξινομηθεί σε ένα από έξι συναισθήματα-κατηγορίες: χαρά, λύπη, θυμό, φόβο, αγάπη και έκπληξη. Είναι χωρισμένο σε σύνολο εκπαίδευσης, επικύρωσης και δοκιμής με 16,000, 2,000 και 2,000 δείγματα αντίστοιχα. Παρακάτω στο σχήμα 5.2 παρατίθεται η κατανομή των συναισθημάτων για τα δείγματα, για όλο το σύνολο δεδομένων.

Το σύνολο δεδομένων υπέστη προεπεξεργασία με tokenization ώστε να μετατραπεί στην κατάλληλη μορφή η είσοδος και να δοθεί στα μοντέλα μας. Για να γίνει αυτό χρησιμοποιήθηκε ο tokenizer WordPiece, και τέθηκε το όριο των 50 tokens για την είσοδο στο σύνολο δεδομένων εκπαίδευσης, συμπεριλαμβανομένων των tokens αρχής, διαχωρισμού και γεμίσματος (padding), ώστε να υπάρχει σταθερή είσοδος. Έτσι η εκπαίδευση εν τέλει πραγματοποιήθηκε με 15,596 δείγματα, μόνο δηλαδή όσα πληρούσαν την προϋπόθεση του μικρότερου μεγέθους από 50 μετά το tokenization. Έγινε η επιλογή να παραμείνουν τα δείγματα με μέγεθος πάνω από 50 tokens στο σύνολο δεδομένων δοκιμής, αλλά η είσοδος να περιοριστεί στα 50 πρώτα tokens, καθώς υπάρχουν πραγματικές περιπτώσεις χρήσης, στις οποίες αυτό μπορεί να συναντάται.



Σχήμα 5.2: Κατανομή των συναισθημάτων στο σύνολο δεδομένων

5.4 Μεθοδολογία Εκπαίδευσης και Μετατροπής

Η εκπαίδευση των Transformer μοντέλων αποτέλεσε σημαντικό κομμάτι της εργασίας, καθώς είναι καίριας σημασίας το να διατηρηθεί σε υψηλό επίπεδο η απόδοση, παρά τις τροποποιήσεις που έγιναν. Η μεθοδολογία που χρησιμοποιήθηκε είναι η εξής:

- Φόρτωση των προεκπαιδευμένων μοντέλων από το Hugging Face μέσω της βιβλιοθήκης transformers της Python και επεξεργασία τους, ώστε να χρησιμοποιηθούν ως μοντέλα ταξινόμησης. Για την επίτευξη του στόχου αυτού έγιναν στα μοντέλα δύο τροποποιήσεις:
  1. Προσθήκη στρώματος εισόδου δεδομένων με μέγεθος εισόδου 50, μέγεθος παρτίδας 64 και τύπο εισόδου ακεραίους 32 bit.
  2. Προσθήκη Κεφαλής Ταξινόμησης (Classification Head) με ένα στρώμα Συγκέντρωσης (για όσες αρχιτεκτονικές δεν το είχαν ήδη ενσωματωμένο στην έξοδο τους), ένα στρώμα εγκατάλειψης (Dropout) με ποσοστό εγκατάλειψης από το σύνολο τιμών [0.1 , 0.2, 0.25], ένα πλήρως συνδεδεμένο στρώμα για 6 κλάσεις και μία συνάρτηση ενεργοποίησης Softmax.
- Fine-tuning των βαρών των μοντέλων για την εργασία της Ανάλυσης Συναισθήματος στο συγκεκριμένο σύνολο δεδομένων με μετεκπαίδευση. Το πρόγραμμα εκπαίδευσης ήταν το εξής:
  - Ρυθμός Μάθησης από το σύνολο τιμών [1e-5, 2.e-5, 2.5e-5, 1e-4, 2e-4, 2.5e-4].
  - Ρυθμός Αποσύνθεσης (decay rate) ίσος με 0.98.
  - Βελτιστοποιητής Adam ή RMSProp με ρυθμό αποσύνθεσης βαρών ίσο με 0.3.
  - Αριθμός εποχών εκπαίδευσης ίσος με 50.
  - Απώλεια Διασταυρούμενης Εντροπίας ως συνάρτηση απώλειας.
  - Ακρίβεια (Accuracy) ως μετρική αξιολόγησης. Η ακρίβεια είναι το ποσοστό των προβλέψεων που πέτυχε το μοντέλο μας. Συγκεκριμένα μπορεί να εκφραστεί ως εξής [62]:

$$Ακρίβεια = \frac{\text{Πλήθος Σωστών Προβλέψεων}}{\text{Πλήθος Συνολικών Προβλέψεων}}$$

  Αξίζει να σημειωθεί ότι για τα βελτιστοποιημένα μοντέλα μετά την αντικατάσταση της συνάρτησης ενεργοποίησης GELU και της Κανονικοποίησης Στρώματος, η μετεκπαίδευση έγινε με fine-tuning των βαρών από το ήδη εκπαιδευμένο πρωτότυπο μοντέλο.
- Κβαντοποίηση των εκπαιδευμένων μοντέλων και μετατροπή τους στη μορφή TFLite αρχείου για την αρχική FP32 μορφή και στα 4 είδη κβαντοποίησης που αναλύθηκαν στο προηγούμενο κεφάλαιο (FP16, DR, INT, FULL).

5.5 Κινητή Συσκευή

Οι μετρήσεις για την εφαρμογή πραγματοποιήθηκαν στη συσκευή Samsung Galaxy A54 5G. Ανήκει στην κατηγορία των Μέτριων Συσκευών (Mid-range), για αυτό ενδείκνυται η χρήση του στην παρούσα διπλωματική, καθώς διαθέτει παρόμοιες δυνατότητες με την πλειοψηφία των συσκευών που χρησιμοποιούνται αυτή τη στιγμή. Τα αναλυτικά χαρακτηριστικά παρατίθενται στον πίνακα 5.8 παρακάτω:

Μοντέλο	Samsung Galaxy A54 5G
Κυκλοφορία	24 Μαρτίου 2023
System-on-Chip	Exynos 1380 (5 nm)
CPU	Octa-core (4x2.4 GHz Cortex-A78 & 4x2.0 GHz Cortex-A55)
GPU	Mali-G68 MP5
NPU	✓
Μνήμη RAM	8 GB
Έκδοση Android	13

Πίνακας 5.8: Χαρακτηριστικά συσκευής

Κεφάλαιο 6ο

Μετρήσεις και Αποτελέσματα

Στην ενότητα αυτή θα γίνει περιγραφή του πειραματικού μέρους της διαδικασίας, ώστε να αξιολογηθούν τα επιλεγμένα μοντέλα στις μεθόδους βελτιστοποίησης που αναλύθηκαν παραπάνω. Για να γίνει η αξιολόγηση ελήφθησαν μετρήσεις σε διαφορετικές συνθήκες εκτέλεσης για κάθε μοντέλο μέσω ειδικής εφαρμογής και αποτιμήθηκε η απόδοση βάσει διαφόρων μετρικών.

6.1 Μετρικές

Η αξιολόγηση των αποτελεσμάτων των μετρήσεων χρόνου έγινε με βάση τις παρακάτω μετρικές:

1. Μέση Τιμή (Average): Ο μέσος όρος των τιμών όλων των μετρήσεων.
2. Διάμεσος (Median): Η τιμή που χωρίζει το σύνολο τιμών σε δύο υποσύνολα με ίδιο πλήθος.
3. Ελάχιστος (Minimum): Η ελάχιστη τιμή όλων των μετρήσεων.
4. Μέγιστος (Maximum): Η μέγιστη τιμή όλων των μετρήσεων.
5. 90ο, 95ο εκατοστημόριο (90^{th}, 95^{th} percentile): Η τιμή κάτω από την οποία βρίσκεται το 90% και το 95% όλων των μετρήσεων αντίστοιχα.
6. Απόδοση (Throughput): Ο αριθμός των μετρήσεων που ολοκληρώνονται στη μονάδα ενός δευτερολέπτου.

Για να εξαχθούν τα δεδομένα και να υπολογιστούν οι παραπάνω τιμές, έγιναν 100 εκτελέσεις για κάθε περίπτωση συμπερασματολογίας, με 5 ακόμη για ζέσταμα των οποίων τα αποτελέσματα δεν ελήφθησαν. Αυτό μας εξασφαλίζει μεγαλύτερη αξιοπιστία των αποτελεσμάτων καθώς περιορίζεται η επίδραση εξωτερικών παραγόντων (π.χ. θερμοκρασία συσκευής, άλλες διεργασίες, κ.α) που μπορεί να επηρεάσουν πιο εύκολα μία μεμονωμένη ή έναν μικρό αριθμό μετρήσεων, ενώ μας επιτρέπει να λάβουμε και στατιστικά στοιχεία για την κατανομή των χρόνων εκτέλεσης για διαφορετικές εισόδους.

Όσον αφορά την επίδοση των μοντέλων, ελήφθησαν επίσης μετρήσεις για τον υπολογισμό των παρακάτω μετρικών:

1. Αριθμός κόμβων που οδηγούνται για υπολογισμό σε κάποιον επιταχυντή (XNNPack ή GPU) μέσω εκπροσώπων (delegates).
2. Αριθμός Υπολογισμών Κινητής Υποδιαστολής που εκτελούνται (floating point operations–FLOPs), που αποτελεί μετρική βάσει της οποίας αξιολογείται η απαιτητικότητα διεργασιών [63]
3. Αριθμός Παραμέτρων (Number of Parameters): το πλήθος των βαρών του μοντέλου.
4. Ακρίβεια: το κλάσμα των προβλέψεων που πέτυχε το μοντέλο μας στο σύνολο δεδομένων δοκιμής (Test Set).

Για την εξαγωγή της επιτάχυνσης ενός μοντέλου σε σχέση με ένα άλλο συγκρίνονται οι μέσες τιμές για τον χρόνο εκτέλεσης τους.

Οι μετρήσεις αυτές έγιναν σε κάθε στάδιο της διαδικασίας, από τα πρωτότυπα μοντέλα μέχρι τα τροποποιημένα, σε πανομοιότυπες συνθήκες υπολογισμού (θερμοκρασία, υπολογιστικός φόρτος, χωρίς παράλληλες διεργασίες, κλπ.). Με τον τρόπο αυτό μπορεί να γίνει αντικειμενική σύγκριση μεταξύ των μοντέλων, ώστε να μελετηθεί η επίδραση των βελτιστοποιήσεων.

6.2 Αποτελέσματα

6.2.1 Συμβατότητα Επιταχυντών

Ποσοστό Συμβατότητας Εκπροσώπων

Μία αρχική παράμετρος αξιολόγησης μπορεί να θεωρηθεί το ποσοστό του μοντέλου που μπορεί να εκτελεστεί από τους διαθέσιμους επιταχυντές. Για τη συγκεκριμένη συσκευή επιταχυντές θεωρούνται η βιβλιοθήκη XNNPack η οποία εφαρμόζεται στη CPU, η GPU και η NPU. Στον πίνακα 6.1 παρατίθεται το μέσο ποσοστό των κόμβων των μοντέλων που υποστηρίζονται από τη βιβλιοθήκη XNNPack και τη GPU για κάθε μέθοδο κβαντοποίησης. Τα αντίστοιχα ποσοστά για την NPU είναι μηδενικά, επομένως η εκτέλεση στην NPU για τη συγκεκριμένη συσκευή δεν υποστηρίζεται.

Quantization	XNNPack	GPU
FP32	72.33	99.52
FP16	76.68	81.62
DR	63.87	99.55
INT	66.61	99.42
FULL	66.56	99.52

Πίνακας 6.1: Το μέσο ποσοστό κόμβων που στέλνονται σε επιταχυντές για εκτέλεση (XNNPack, GPU) για όλα τα μοντέλα.

Παρατηρήσεις:

Όπως φαίνεται από τον πίνακα 6.1, στη GPU γίνεται ο υπολογισμός όλων σχεδόν των κόμβων που απαιτούνται για την εκτέλεση των Transformer μοντέλων μας, σε σχέση με τον επιταχυντή XNNPack, που υποστηρίζει μόνο ένα ποσοστό της τάξης του 70 τοις εκατό.

Απόδοση Επιταχυντών

Στον πίνακα 6.2 παρατίθενται τα δεδομένα για την επιτάχυνση της εκτέλεσης των αρχικών μοντέλων για επιταχυντές σε σχέση με την εκτέλεση σε CPU για ένα νήμα.

Model	Accelerator	Quantization				
		FP32	FP16	DR	INT	FULL
MobileBERT	XNNPack	1.69	1.70	1.26	1.35	1.36
	GPU	1.80	1.83	0.97	1.02	1.02
Xdistil-L12-H384	XNNPack	1.42	1.41	1.10	1.08	1.09
	GPU	2.10	2.15	1.07	0.99	0.99
RoBERTa$_{TINY}$	XNNPack	1.45	1.45	1.13	1.16	1.18
	GPU	1.84	1.84	0.96	0.99	0.99
DistilBERT	XNNPack	1.59	1.56	1.21	1.23	1.22
	GPU	3.41	3.26	1.36	1.05	1.07
ELECTRA$_{SMALL}$	XNNPack	1.48	1.49	1.18	1.15	1.15
	GPU	1.80	1.78	1.02	1.00	1.01
MiniLMv1-L12-H384	XNNPack	1.46	1.45	1.17	1.09	1.13
	GPU	2.04	2.10	1.04	0.99	0.99

Πίνακας 6.2: Επιτάχυνση εκτέλεσης των αρχικών μοντέλων για χρήση XNNPack και GPU σε σχέση με τη CPU.

Παρατηρήσεις:

Όπως αναμένεται από τα δεδομένα του πίνακα 6.10, επιτυγχάνεται μεγαλύτερη επιτάχυνση μέσω της χρήσης της GPU σε σχέση με τη χρήση του XNNPack, εφόσον η πρώτη είχε μεγαλύτερη ενσωμάτωση των πράξεων. Παρόλα αυτά η βελτίωση αυτή περιορίζεται στα FP32 και FP16 μοντέλα, καθώς η εκτέλεση για τη GPU γίνεται σε μορφή floating point 16 byte ανεξαρτήτως κβαντοποίησης. Αποτέλεσμα αυτού είναι να έχουμε μηδαμινή ή καθόλου επιτάχυνση για τις DR, INT και FULL κβαντοποιήσεις.

Όσον αφορά τη χρήση του XNNPack, προσφέρει βελτίωση της ίδιας περίπου τάξης μεγέθους ανεξαρτήτως κβαντοποίησης και μοντέλου, και είναι επομένως πιο συνεπής.

6.2.2 Ακρίβεια

Ακρίβεια Αρχικών Μοντέλων

Στον πίνακα 6.3 παρουσιάζονται οι 6 αρχιτεκτονικές που χρησιμοποιήθηκαν στην παρούσα εργασία μαζί με τα χαρακτηριστικά τους και την ακρίβεια τους στο task της Ανάλυσης Συναισθήματος. Η εκτέλεση των μοντέλων έγινε μέσω της πλατφόρμας Kaggle με τη χρήση επιταχυντή GPU μοντέλου P100 [64] για κέντρα δεδομένων.

Model	FLOPs	Params	Accuracy (%)				
			FP32	FP16	DR	INT	FULL
MobileBERT	2.06 G	24.33 M	93.10	93.10	92.95	93.10	92.65
Xdistil-L12-H384	2.18 G	33.19 M	93.15	93.15	93.00	86.35	85.25
RoBERTa$_{TINY}$	1.28 G	27.75 M	93.15	93.15	93.25	91.85	92.30
DistilBERT	4.30 G	66.01 M	93.30	93.30	93.30	92.65	92.30
ELECTRA$_{SMALL}$	0.98 G	13.46 M	93.35	93.35	93.15	92.95	93.30
MiniLMv1-L12-H384	2.18 G	33.19 M	93.55	93.55	93.30	92.15	92.10

Πίνακας 6.3: Ακρίβεια των Μοντέλων που χρησιμοποιήθηκαν για εκτέλεση σε GPU P100

Παρατηρήσεις:

Από τις μετρήσεις που έγιναν για την ακρίβεια των μοντέλων, παρατηρούμε ότι η επίδοση όλων τους είναι σε κοντινά επίπεδα. Επιλέχθηκαν μοντέλα με σχετικά μικρό μέγεθος (<250MB), καθώς επίκεντρο της έρευνας αυτής είναι οι κινητές συσκευές, οι οποίες διαθέτουν μικρότερο αποθηκευτικό χώρο. Επίσης σημαντικό είναι να αναφερθεί το γεγονός ότι η κβαντοποίηση λειτουργεί για την πλειοψηφία των μοντέλων, προκαλώντας σχετικά μικρή πτώση στην ακρίβεια, της τάξης του 1% με 2%, με εξαίρεση το μοντέλο Xdistil, στο οποίο έχουμε σημαντική πτώση της επίδοσης για κβαντοποίηση INT και FULL.

Ακρίβεια Βελτιστοποιημένων Μοντέλων

Παρακάτω, στους πίνακες 6.4, 6.5, 6.6, 6.7, 6.8, 6.9 παρατίθεται η ακρίβεια για τους διάφορους τύπους κβαντοποίησης, μετά την αντικατάσταση της Κανονικοποίησης Στρώματος (Layer Normalization) με την Κανονικοποίηση Παρτίδας (Batch Normalization) και της συνάρτησης ενεργοποίησης GELU με τις συναρτήσεις ReLU, ReLU6, Leaky ReLU, Sigmoid, Swish, Tanh αντίστοιχα. Με bold σημειώνονται οι ακρίβειες που είναι εξίσου ψηλές ή υψηλότερες από την αντίστοιχη ακρίβεια του πρωτότυπου μοντέλου.

- **ReLU**

Model	Accuracy (%)				
	FP32	FP16	DR	INT	FULL
MobileBERT	93.10	93.10	92.95	93.10	92.65
Xdistil-L12-H384	**93.25**	**93.25**	93.05	**93.15**	**92.75**
RoBERTa$_{TINY}$	92.95	92.95	92.90	**92.95**	**92.50**
DistilBERT	91.80	91.80	91.80	91.35	91.45
ELECTRA$_{SMALL}$	93.00	93.00	92.85	92.85	92.90
MiniLMv1-L12-H384	93.40	93.40	93.05	**93.10**	**93.05**

Πίνακας 6.4: Ακρίβεια Βελτιστοποιημένων Μοντέλων με ReLU για εκτέλεση σε GPU P100

- **ReLU6**

Model	Accuracy (%)				
	FP32	FP16	DR	INT	FULL
Xdistil-L12-H384	**93.70**	**93.70**	**93.50**	**93.60**	**93.60**
RoBERTa$_{TINY}$	93.00	93.00	92.80	**93.10**	**92.90**
ELECTRA$_{SMALL}$	92.75	92.75	92.75	92.50	92.70
MiniLMv1-L12-H384	93.45	93.45	93.15	**93.25**	**93.20**

Πίνακας 6.5: Ακρίβεια Βελτιστοποιημένων Μοντέλων με ReLU6 για εκτέλεση σε GPU P100

- **Leaky ReLU**

Model	Accuracy (%)				
	FP32	FP16	DR	INT	FULL
Xdistil-L12-H384	**93.20**	**93.20**	92.90	**92.95**	**93.20**
RoBERTa$_{TINY}$	93.00	93.00	93.05	**93.00**	**93.25**
ELECTRA$_{SMALL}$	92.70	92.70	92.65	92.60	92.60
MiniLMv1-L12-H384	93.05	93.05	92.80	**93.05**	**93.15**

Πίνακας 6.6: Ακρίβεια Βελτιστοποιημένων Μοντέλων με Leaky ReLU για εκτέλεση σε GPU P100

- **Sigmoid**

Model	Accuracy (%)				
	FP32	FP16	DR	INT	FULL
Xdistil-L12-H384	92.25	92.25	91.35	**91.40**	**91.40**
RoBERTa$_{TINY}$	**93.15**	**93.15**	92.80	**92.65**	**92.85**
DistilBERT	92.90	92.90	92.75	92.60	**92.60**
ELECTRA$_{SMALL}$	90.95	90.95	90.60	90.90	91.00
MiniLMv1-L12-H384	92.10	92.10	90.00	90.65	90.50

Πίνακας 6.7: Ακρίβεια Βελτιστοποιημένων Μοντέλων με Sigmoid για εκτέλεση σε GPU P100

- **Swish**

Model	Accuracy (%)				
	FP32	FP16	DR	INT	FULL
Xdistil-L12-H384	90.60	90.60	89.90	**87.65**	**87.90**
RoBERTa$_{TINY}$	**93.25**	**93.20**	**93.35**	**93.40**	**93.55**
DistilBERT	92.85	92.85	92.80	92.60	**92.70**
ELECTRA$_{SMALL}$	92.95	92.95	92.90	92.85	92.90
MiniLMv1-L12-H384	93.15	93.15	93.00	**93.00**	**92.90**

Πίνακας 6.8: Ακρίβεια Βελτιστοποιημένων Μοντέλων με Swish για εκτέλεση σε GPU P100

- **Tanh**

Model	Accuracy (%)				
	FP32	FP16	DR	INT	FULL
Xdistil-L12-H384	91.35	91.35	90.15	**90.10**	**90.15**
RoBERTa$_{TINY}$	92.95	92.95	93.00	**92.95**	**92.85**
DistilBERT	87.90	87.90	87.80	88.00	87.95
ELECTRA$_{SMALL}$	89.95	89.95	90.05	90.00	90.15
MiniLMv1-L12-H384	92.40	92.40	92.15	**92.25**	**92.15**

Πίνακας 6.9: Ακρίβεια Βελτιστοποιημένων Μοντέλων με Tanh για εκτέλεση σε GPU P100

Παρατηρήσεις:

Από τις μετρήσεις ακρίβειας προκύπτει ότι οι βελτιστοποιήσεις που έγιναν μπορούν να επηρεάσουν την ακρίβεια, όμως με την χρήση της κατάλληλης συνάρτησης ενεργοποίησης, η μείωση που προκαλούν μπορεί να αντισταθμιστεί. Μάλιστα σε κάποιες περιπτώσεις (π.χ. μοντέλο Xdistil με ReLU6, μοντέλο RoBERTa με Swish), η επίδοση ανεβαίνει σε σχέση με το πρωτότυπο.

Παράλληλα για όλες τις συναρτήσεις ενεργοποίησης παρατηρούμε ότι η επίδραση της κβαντοποίησης στην ακρίβεια μειώνεται σημαντικά, και τα μοντέλα προσεγγίζουν σε ακρίβεια τα αντίστοιχα FP32. Για το μοντέλο Xdistil δεν παρατηρείται, όπως στο πρωτότυπο, η κατακόρυφη μείωση της ακρίβειας αλλά αντίθετα υπάρχει περίπτωση (Xdistil με LeakyReLU), όπου οι FULL και INT κβαντοποιήσεις έχουν την ίδια ακρίβεια με την FP32.

Τέλος, δεν υπάρχει κάποια συνάρτηση ενεργοποίησης που να προσφέρει βέλτιστες επιδόσεις για όλα τα μοντέλα, καθώς αλληλεπιδρούν με διαφορετικό τρόπο με κάθε αρχιτεκτονική και μπορεί να βελτιώνουν κάποια, ενώ επιδεινώνουν κάποια άλλα.

Ακρίβεια Βελτιστοποιημένων Μοντέλων στην Κινητή Συσκευή

Στη συνέχεια παρατίθενται στους πίνακες 6.10, 6.11 τα αποτελέσματα των μετρήσεων για την ακρίβεια των αρχικών μοντέλων και των βελτιστοποιημένων μοντέλων σε CPU και GPU. Οι μετρήσεις έγιναν στην κινητή συσκευή Samsung A54 5G, τα χαρακτηριστικά της οποίας αναλύθηκαν στην ενότητα 5.6.

Σημειώνεται ότι στους επόμενους πίνακες λείπουν οι μετρήσεις που αναφέρονται στο μοντέλο DistilBERT με συνάρτηση ενεργοποίησης ReLU6 και Leaky ReLU καθώς οι συναρτήσεις αυτές δεν υποστηρίζονταν στο Hugging Face για το συγκεκριμένο μοντέλο, λόγω του τρόπου υλοποίησης του.

Model	Quantization	CPU On-Device Accuracy (%)						
		GELU	ReLU	ReLU6	L. ReLU	Sigmoid	SiLU	Tanh
Xdistil-L12-H384	FP32	93.15	93.25	93.70	93.20	92.25	90.60	91.35
	FP16	93.15	93.25	93.70	93.20	92.25	90.60	91.35
	DR	93.10	93.05	93.50	92.90	91.30	89.95	90.15
	INT	84.35	93.10	93.70	93.05	91.40	87.65	90.00
	FULL	85.20	92.85	93.55	92.95	91.35	87.30	90.00
RoBERTa$_{TINY}$	FP32	93.15	92.95	93.00	93.00	93.15	93.25	92.95
	FP16	93.15	92.95	93.00	93.00	93.15	93.2	92.95
	DR	93.25	92.9	92.80	93.05	92.75	93.35	93.00
	INT	92.10	92.90	92.90	92.9	92.80	93.10	93.00
	FULL	92.05	92.65	92.90	93.05	93.00	93.20	92.90
DistilBERT	FP32	93.30	91.80			92.90	92.85	87.90
	FP16	93.30	91.80			92.90	92.85	87.90
	DR	93.25	91.80			92.80	92.85	87.80
	INT	92.20	91.20			92.50	92.65	87.80
	FULL	92.50	90.95			92.50	92.65	87.75
ELECTRA$_{SMALL}$	FP32	93.35	93.00	92.75	92.70	90.95	92.95	89.95
	FP16	93.35	93.00	92.75	92.70	90.95	92.95	89.95
	DR	93.20	92.85	92.75	92.65	90.60	92.90	90.05
	INT	92.90	92.90	92.55	92.60	90.80	92.75	89.95
	FULL	92.90	92.85	92.6	92.70	91.00	92.80	89.95
MiniLMv1-L12-H384	FP32	93.55	93.40	93.45	93.05	92.10	93.15	92.40
	FP16	93.55	93.40	93.45	93.05	92.10	93.15	92.40
	DR	93.30	93.05	93.15	92.80	90.05	93.05	92.10
	INT	92.40	93.15	93.45	92.95	90.45	93.05	92.10
	FULL	92.10	93.05	93.35	93.10	90.40	93.05	92.30

Πίνακας 6.10: Ακρίβεια αρχικών και βελτιστοποιημένων μοντέλων στη CPU της κινητής συσκευής

Model	Quantization	GPU On-Device Accuracy (%)						
		GELU	ReLU	ReLU6	L. ReLU	Sigmoid	SiLU	Tanh
Xdistil-L12-H384	FP32	13.75	93.30	93.65	93.20	92.25	90.6	91.35
	FP16	13.75	93.25	93.65	93.20	92.25	90.6	91.35
	DR	29.05	93.05	93.50	92.90	91.20	89.95	90.15
	INT	34.75	93.1	93.45	92.95	91.30	88.40	90.05
	FULL	34.75	92.85	93.55	92.85	91.40	88.50	90.20
RoBERTa$_{TINY}$	FP32	13.75	92.95	92.95	93.00	93.15	93.25	92.95
	FP16	13.75	92.95	92.95	93.00	93.15	93.25	92.95
	DR	34.75	92.90	92.95	93.10	92.70	93.35	92.95
	INT	34.75	92.90	92.90	92.85	92.75	93.15	93.00
	FULL	34.75	92.65	93.00	93.25	92.80	93.15	92.85
DistilBERT	FP32	7.95	91.80			92.90	92.90	87.90
	FP16	7.95	91.80			92.90	92.90	87.90
	DR	11.20	91.80			92.85	92.85	87.80
	INT	10.65	91.20			92.65	92.80	87.75
	FULL	10.95	90.95			92.65	92.80	87.65
ELECTRA$_{SMALL}$	FP32	34.75	93.00	92.75	92.70	91.00	92.95	89.95
	FP16	34.75	93.00	92.75	92.70	90.95	92.95	89.95
	DR	34.75	92.85	92.75	92.60	90.65	92.90	90.15
	INT	34.75	92.90	92.65	92.50	90.85	92.80	89.95
	FULL	34.75	92.85	92.6	92.50	90.85	92.80	89.90
MiniLMv1-L12-H384	FP32	34.75	93.40	93.45	93.10	92.05	93.20	92.40
	FP16	34.75	93.40	93.45	93.10	92.05	93.20	92.40
	DR	34.75	93.05	93.15	92.90	90.20	93.00	92.20
	INT	34.75	93.15	93.20	93.10	89.90	92.70	92.25
	FULL	34.75	93.05	93.30	93.10	89.90	93.00	92.15

Πίνακας 6.11: Ακρίβεια αρχικών και βελτιστοποιημένων μοντέλων στη GPU της κινητής συσκευής

Παρατηρήσεις:

Παρατηρείται ότι η ακρίβεια διατηρείται για την εκτέλεση στη CPU για όλα τα μοντέλα και με όλους τους τρόπους κβαντοποίησης, με πολύ μικρές αποκλίσεις σε σχέση με την εκτέλεση στην P100, της τάξης του 0.05-0.10%. Παρατηρείται επίσης ότι υπάρχει πτώση της ακρίβειας για την εκτέλεση των αρχικών μοντέλων σε GPU, όπως αναμένεται από τα αποτελέσματα ερευνών [26]. Η ακρίβεια επανέρχεται (με επίσης πολύ μικρές αποκλίσεις) με τις βελτιστοποιήσεις που εφαρμόστηκαν και συγκεκριμένα με την αντικατάσταση του Layer Normalization με το Batch Normalization, όπως διαπιστώθηκε από δοκιμές που πραγματοποιήθηκαν.

Ακρίβεια Folded Μοντέλων στην Κινητή Συσκευή

Μέχρι στιγμής οι μετρήσεις μας αφορούσαν τα Unfolded μοντέλα, καθώς το Batch MatMul Unfolding αποτελεί προεπιλογή στα TFLite μοντέλα. Παρακάτω στους πίνακες 6.12 και 6.13, παρατίθενται τα αποτελέσματα ακρίβειας των Folded μοντέλων, αρχικών και βελτιστοποιημένων σε CPU και GPU. Οι μετρήσεις έγιναν στην κινητή συσκευή Samsung A54 5G, τα χαρακτηριστικά της οποίας αναλύθηκαν στην ενότητα 5.6.

Model	Quantization	CPU On-Device Accuracy (%)						
		GELU	ReLU	ReLU6	L. ReLU	Sigmoid	SiLU	Tanh
MobileBERT	FP32		93.10					
	FP16		93.00					
	DR		92.80					
	INT		92.95					
	FULL		93.25					
Xdistil-L12-H384	FP32	91.35	93.25	93.70	93.20	92.25	90.60	91.35
	FP16	91.35	93.25	93.70	93.20	92.25	90.60	91.35
	DR	90.15	93.05	93.50	92.90	91.30	89.95	90.15
	INT	90.05	93.00	93.55	92.75	91.40	87.60	90.05
	FULL	90.00	92.80	93.60	93.20	91.20	87.55	90.00
RoBERTa$_{TINY}$	FP32	93.15	92.95	93.00	93.00	93.15	93.25	92.95
	FP16	93.15	92.95	93.00	93.00	93.15	93.20	92.95
	DR	93.25	92.90	92.80	93.05	92.75	93.35	93.00
	INT	92.15	92.80	93.05	92.90	92.75	93.30	92.90
	FULL	92.00	92.80	92.90	93.15	92.80	93.40	92.90
DistilBERT	FP32	93.30	91.80			92.90	92.85	87.90
	FP16	93.30	91.80			92.90	92.85	87.90
	DR	93.30	91.80			92.80	92.85	87.80
	INT	92.30	91.25			92.90	92.55	87.70
	FULL	92.15	91.35			92.50	92.60	87.75
ELECTRA$_{SMALL}$	FP32	93.35	93.00	92.75	92.70	90.95	92.95	89.95
	FP16	93.35	93.00	92.75	92.70	90.95	92.95	89.95
	DR	93.20	92.85	92.75	92.65	90.60	92.90	90.05
	INT	92.80	92.80	92.45	92.45	90.80	92.95	89.90
	FULL	93.05	92.85	92.55	92.85	90.90	92.75	90.05
MiniLMv1-L12-H384	FP32	93.55	93.40	93.45	93.05	92.10	93.15	92.40
	FP16	93.55	93.40	93.45	93.05	92.10	93.15	92.40
	DR	93.30	93.05	93.15	92.80	90.05	93.05	92.15
	INT	92.35	92.95	93.05	92.90	90.45	93.05	92.30
	FULL	92.35	92.90	93.15	93.00	90.35	92.95	92.10

Πίνακας 6.12: Ακρίβεια αρχικών και βελτιστοποιημένων Folded μοντέλων στη CPU της κινητής συσκευής.

Model	Quantization	GPU On-Device Accuracy (%)						
		GELU	ReLU	ReLU6	L. ReLU	Sigmoid	SiLU	Tanh
MobileBERT	FP32		14.10					
	FP16		13.85					
	DR		13.95					
	INT		13.20					
	FULL		13.95					
Xdistil-L12-H384	FP32	13.75	34.75	38.95	31.05	36.15	27.50	13.75
	FP16	13.75	34.75	38.95	31.05	36.15	27.50	13.75
	DR	25.10	43.35	38.10	41.10	34.75	19.65	25.10
	INT	22.30	42.45	38.15	29.25	37.65	21.00	22.30
	FULL	24.20	43.10	37.60	27.95	14.50	21.35	24.20
RoBERTa$_{TINY}$	FP32	16.00	31.50	18.60	47.65	37.40	37.20	25.45
	FP16	15.95	31.50	18.60	47.65	37.40	37.20	25.50
	DR	11.20	31.15	16.55	44.95	33.30	36.00	21.50
	INT	11.20	33.75	16.20	43.80	34.50	34.70	22.10
	FULL	11.20	34.10	15.60	44.00	34.35	35.50	22.30
DistilBERT	FP32	13.75	34.75			34.75	13.75	33.90
	FP16	13.75	34.75			34.75	13.75	33.90
	DR	13.75	34.75			34.75	13.75	33.70
	INT	13.70	34.75			34.75	20.60	13.75
	FULL	13.65	34.75			34.75	20.10	13.75
ELECTRA$_{SMALL}$	FP32	3.30	48.35	40.95	15.05	38.45	39.55	50.75
	FP16	13.75	48.30	41.00	15.05	38.45	39.55	50.70
	DR	3.30	47.05	40.80	14.95	38.15	39.70	51.15
	INT	11.20	38.85	38.70	15.30	39.40	40.15	50.55
	FULL	11.20	38.40	39.40	15.10	39.70	40.70	51.05
MiniLMv1-L12-H384	FP32	34.75	31.25	14.85	23.95	13.75	20.85	39.50
	FP16	34.75	13.75	14.90	24.00	24.30	20.80	39.45
	DR	7.95	34.75	16.50	26.95	14.55	15.30	30.75
	INT	7.95	34.70	21.35	21.30	16.95	15.45	40.80
	FULL	7.95	39.90	21.05	21.50	17.30	15.55	41.00

Πίνακας 6.13: Ακρίβεια αρχικών και βελτιστοποιημένων Folded μοντέλων στη GPU της κινητής συσκευής.

Παρατηρήσεις:

Παρατηρείται ότι η ακρίβεια των Folded μοντέλων είναι παρόμοια με αυτή των Unfolded με πολύ μικρές διακυμάνσεις για εκτέλεση στη CPU. Όσον αφορά την ακρίβεια για εκτέλεση σε GPU από τα δεδομένα προκύπτει ότι η χρήση Batch Matrix Multiplication Folding την επηρεάζει αρνητικά σε σημείο που καθίσταται απαγορευτική η χρήση GPU, ακόμα και για τα βελτιστοποιημένα μοντέλα για τα οποία είχε αποκατασταθεί η ακρίβεια με τις βελτιστοποιήσεις.

6.2.3 Επιτάχυνση Μοντέλων

Κβαντοποίηση

Παρακάτω στον πίνακα 6.14 παρουσιάζονται τα αποτελέσματα επιτάχυνσης της εκτέλεσης των μοντέλων στην CPU της κινητής συσκευής, για τους διάφορους τρόπους κβαντοποίησης σε σχέση με τα αντίστοιχα FP32 μοντέλα.

Model	Latency Speedup			
	FP16	DR	INT	FULL
MobileBERT	1.00	2.20	2.61	2.61
Xdistil-L12-H384	0.99	2.30	2.12	2.12
RoBERTa-tiny	0.99	2.44	2.41	2.41
DistilBERT	1.03	2.81	2.89	2.89
ELECTRA$_{SMALL}$	0.99	2.02	1.78	1.78
MiniLMv1-L12-H384	1.00	2.38	2.10	2.10

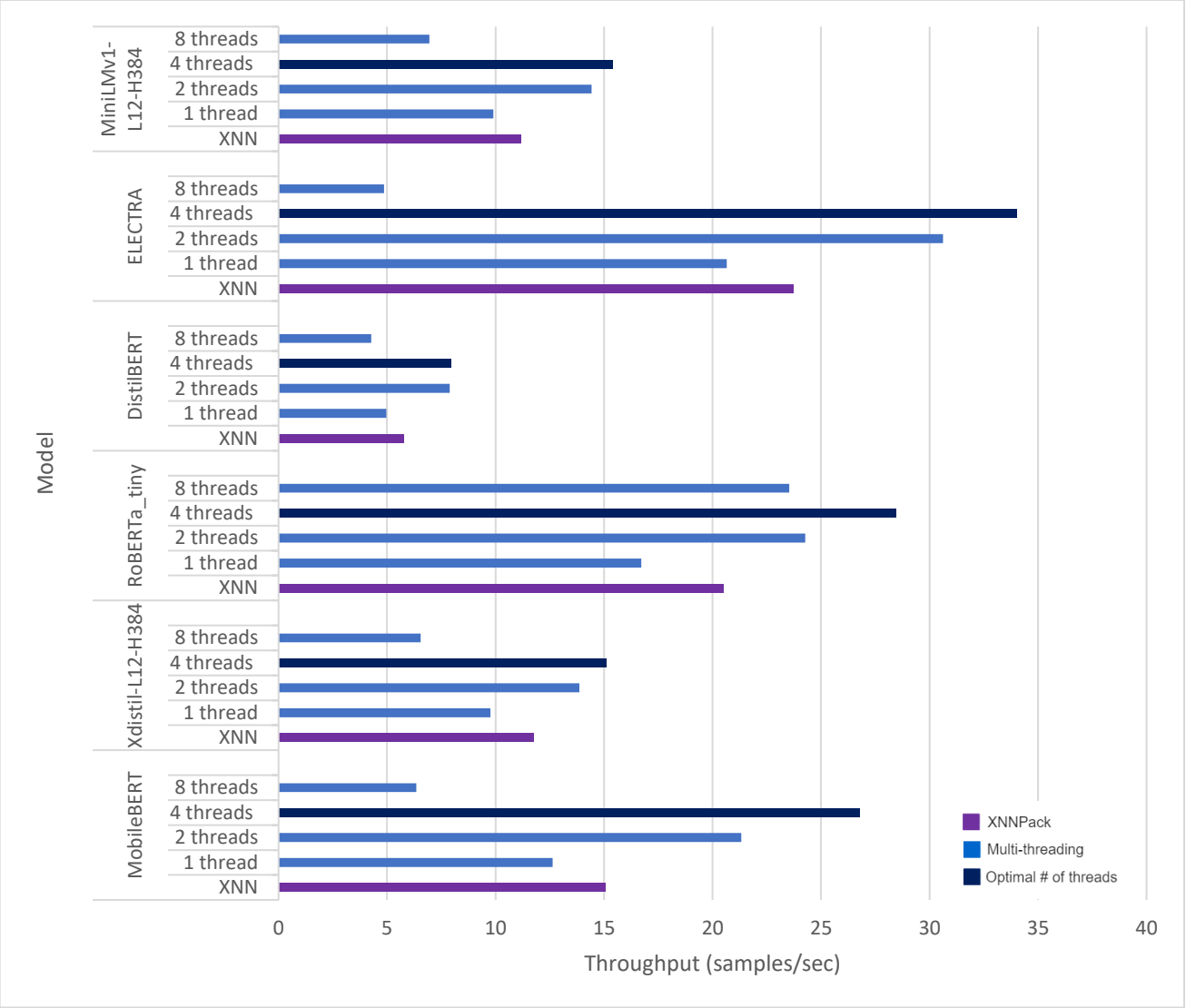
Πίνακας 6.14: Επιτάχυνση των μοντέλων για τα διάφορα σχήματα κβαντοποίησης

Παρατηρήσεις:

Όπως είναι αναμενόμενο δεν υπάρχει κάποια επιτάχυνση για Floating Point 16 κβαντοποίηση, παρόλα αυτά υπάρχει νόημα στη χρήση της καθώς το μέγεθος πέφτει στο ½ του αρχικού. Για τους υπόλοιπους τύπους κβαντοποίησης, η μείωση αυτή είναι στο ¼ και παρατηρείται επιτάχυνση περίπου x2 σε σχέση με το αρχικό FP32 μοντέλο. Η βελτίωση στον χρόνο εκτέλεσης είναι σχετικά παρόμοια για όλες τις αρχιτεκτονικές, με διακυμάνσεις.

Απόδοση CPU

Στο σχήμα 6.1 παρατίθεται η ρυθμαπόδοση (Throughput), δηλαδή το πλήθος δειγμάτων για τα οποία γίνεται συμπερασματολογία ανά δευτερόλεπτο, για όλα τα FP32 μοντέλα, με τη χρήση ενός ή πολλαπλών νημάτων (threads) και με χρήση του επιταχυντή XNNPack. Ο επιταχυντής αυτός χρησιμοποιεί νηματοποίηση ως προεπιλογή, και για το λόγο αυτό δεν συνίσταται η χρήση του με πολλαπλά νήματα με άλλο τρόπο. Με γαλάζιο σημειώνεται η εκτέλεση για πολλαπλά νήματα, με μωβ η εκτέλεση με τη χρήση XNNPack και με σκούρο μπλε ο ιδανικός αριθμός νημάτων.



Σχήμα 6.1: Απόδοση της CPU στην κινητή συσκευή.

Παρατηρήσεις:

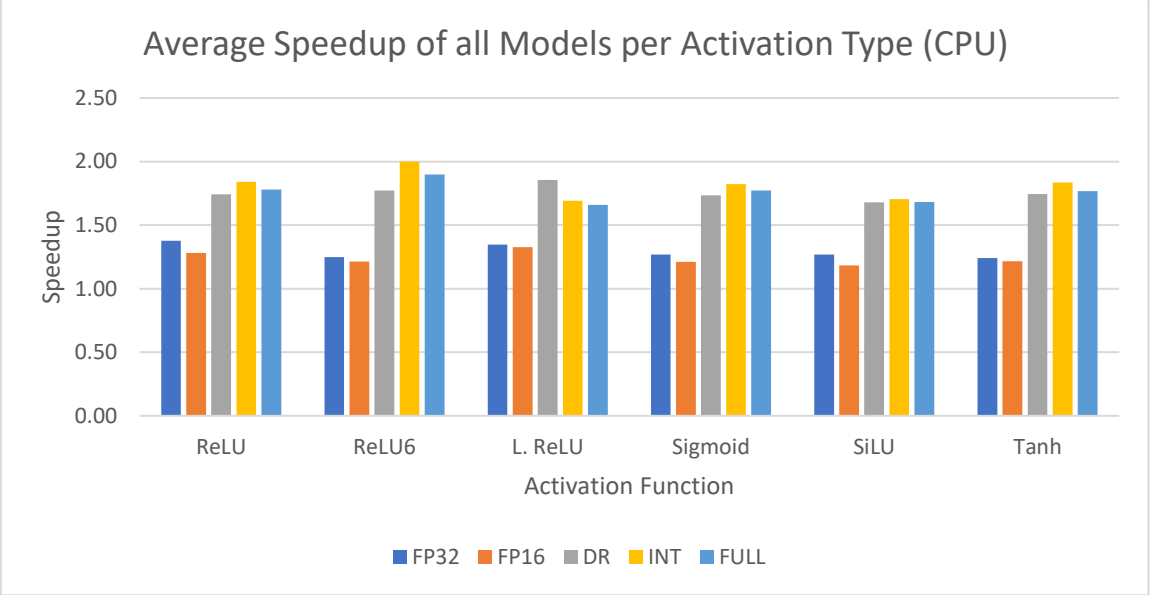
Όπως φαίνεται από το σχήμα 6.1, η βέλτιστη ρύθμιση των παραμέτρων εκτέλεσης για όλα τα μοντέλα είναι η χρήση 4 νημάτων, αφού επιτυγχάνεται η βέλτιστη παραλληλοποίηση, και έχουμε τη μέγιστη απόδοση. Η απόδοση που επιτυγχάνεται είναι έως και 2 φορές μεγαλύτερη από αυτή που επιτυγχάνεται με τη χρήση 1 νήματος. Παρατηρείται επίσης ότι ενώ η χρήση του XNNPack μπορεί να προσφέρει επιτάχυνση σε σχέση με τη χρήση 1 νήματος, δεν προσφέρει βελτίωση σε σχέση με τη χρήση περισσότερων νημάτων.

Επιτάχυνση Βελτιστοποιημένων Μοντέλων στην Κινητή Συσκευή

Στη συνέχεια παρατίθενται στους πίνακες 6.15, 6.16 και στα σχήματα 6.2, 6.3 τα δεδομένα για την επιτάχυνση των βελτιστοποιημένων μοντέλων σε σχέση με τα αρχικά σε CPU και GPU. Οι μετρήσεις έγιναν στην κινητή συσκευή Samsung A54 5G και η επιτάχυνση που αναφέρεται είναι σε σχέση με το αρχικό μοντέλο χωρίς καμία τροποποίηση για το βέλτιστο αριθμό νημάτων (για τη CPU).

Model	Quantization	CPU Average Speedup					
		ReLU	ReLU6	L. ReLU	Sigmoid	SiLU	Tanh
Xdistil-L12-H384	FP32	1.26	1.25	1.22	1.21	1.26	1.21
	FP16	1.27	1.27	1.23	1.22	1.27	1.14
	DR	1.47	1.72	1.76	1.70	1.66	1.72
	INT	2.09	2.08	1.75	2.10	1.99	2.07
	FULL	1.89	1.90	1.60	1.91	1.83	1.90
RoBERTa$_{TINY}$	FP32	1.43	1.38	1.40	1.39	1.38	1.26
	FP16	1.34	1.28	1.28	1.32	1.27	1.20
	DR	1.87	1.85	1.95	1.81	1.78	1.90
	FULL	2.13	1.93	1.53	2.09	1.81	2.16
	FULL	1.99	1.72	1.54	2.00	1.87	2.05
DistilBERT	FP32	1.49			1.37	1.31	1.25
	FP16	1.21			1.16	1.05	1.15
	DR	1.74			1.70	1.60	1.72
	INT	1.03			1.01	0.98	1.01
	FULL	1.03			1.03	0.97	0.99
ELECTRA$_{SMALL}$	FP32	1.58	1.15	1.56	1.18	1.22	1.43
	FP16	1.54	1.12	1.60	1.21	1.20	1.48
	DR	1.97	1.89	2.00	1.89	1.79	1.81
	INT	2.03	2.05	1.87	2.00	1.90	2.04
	FULL	2.06	2.05	1.88	2.01	1.90	2.01
MiniLMv1-L12-H384	FP32	1.12	1.21	1.21	1.20	1.18	1.05
	FP16	1.05	1.18	1.20	1.14	1.13	1.11
	DR	1.66	1.63	1.71	1.57	1.57	1.58
	INT	1.92	1.94	1.62	1.91	1.84	1.90
	FULL	1.93	1.92	1.62	1.91	1.84	1.89

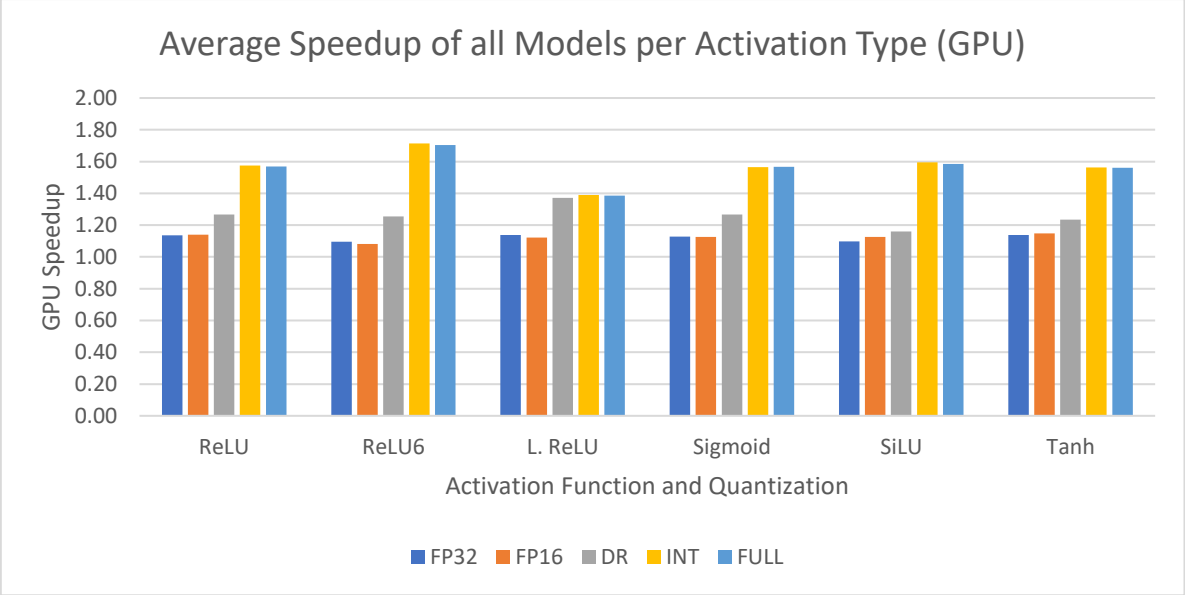
Πίνακας 6.15 : Επιτάχυνση αρχικών και βελτιστοποιημένων μοντέλων στη CPU της κινητής συσκευής



Σχήμα 6.2: Μέσος όρος επιτάχυνσης για όλα τα μοντέλα ανά συνάρτηση ενεργοποίησης και τρόπο κβαντοποίησης στη CPU

Model	Quantization	GPU Average Speedup					
		ReLU	ReLU6	L. ReLU	Sigmoid	SiLU	Tanh
Xdistil-L12-H384	FP32	1.10	1.14	1.11	1.16	1.09	1.12
	FP16	1.14	1.10	1.11	1.11	1.11	1.12
	DR	1.25	1.23	1.26	1.23	1.21	1.25
	INT	1.82	1.83	1.49	1.80	1.77	1.81
	FULL	1.81	1.80	1.47	1.81	1.75	1.80
RoBERTa$_{TINY}$	FP32	1.23	1.15	1.17	1.14	1.07	1.07
	FP16	1.15	1.08	1.19	1.14	1.12	1.14
	DR	1.43	1.19	1.49	1.40	1.27	1.43
	INT	1.71	1.71	1.43	1.70	1.65	1.69
	FULL	1.71	1.70	1.44	1.70	1.67	1.69
DistilBERT	FP32	1.09			1.18	1.27	1.21
	FP16	1.10			1.22	1.24	1.22
	DR	1.11			1.24	1.17	1.20
	INT	1.03			1.03	1.03	1.02
	FULL	1.03			1.04	1.06	1.03
ELECTRA$_{SMALL}$	FP32	1.15	0.95	1.13	0.97	0.95	1.14
	FP16	1.18	1.05	1.08	1.03	1.04	1.13
	DR	1.31	1.36	1.38	1.28	0.98	1.08
	INT	1.50	1.49	1.16	1.49	1.76	1.48
	FULL	1.50	1.51	1.17	1.50	1.69	1.49
MiniLMv1-L12-H384	FP32	1.11	1.14	1.14	1.19	1.11	1.15
	FP16	1.13	1.10	1.11	1.13	1.12	1.13
	DR	1.23	1.24	1.36	1.18	1.17	1.21
	INT	1.82	1.83	1.48	1.81	1.77	1.81
	FULL	1.80	1.81	1.46	1.79	1.76	1.79

Πίνακας 6.16: Επιτάχυνση αρχικών και βελτιστοποιημένων μοντέλων στη GPU της κινητής συσκευής



Σχήμα 6.3 : Μέσος όρος επιτάχυνσης για όλα τα μοντέλα ανά συνάρτηση ενεργοποίησης και τρόπο κβαντοποίησης στη GPU

Παρατηρείται για την επιτάχυνση που επιτεύχθηκε, ότι δεν υπάρχει κάποιο μοτίβο που να ταιριάζει σε όλα τα μοντέλα, για όλες τις συναρτήσεις ενεργοποίησης και όλες τις κβαντοποιήσεις, ούτε να καθοριστεί κάποια τροπολογία ως βέλτιστη. Μπορούν όμως να γίνουν οι εξής παρατηρήσεις:

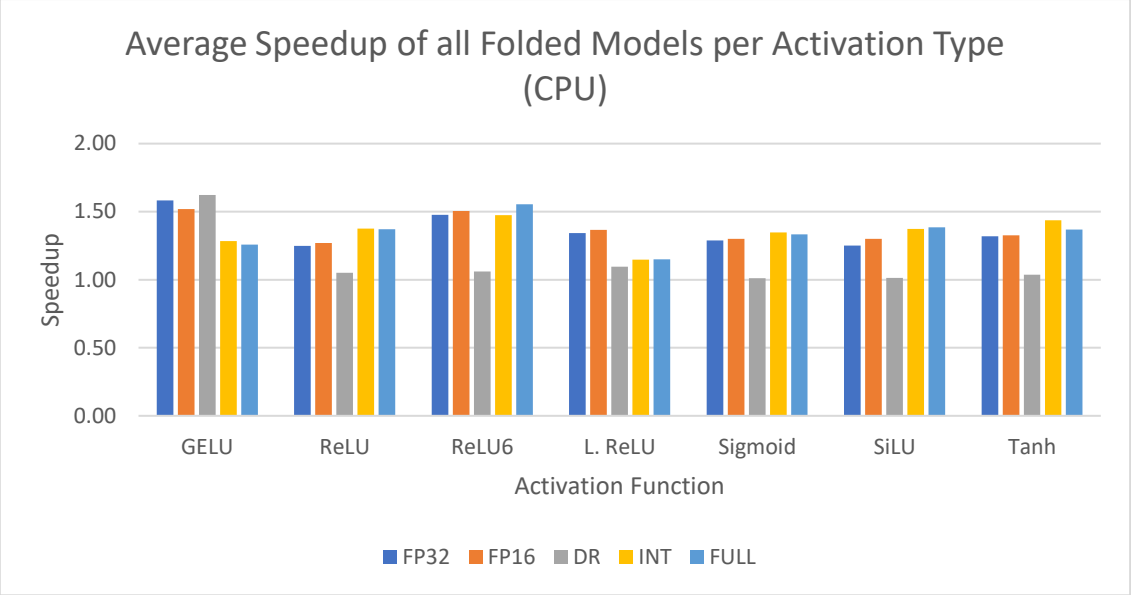
- Με τις βελτιστοποιήσεις μας επιταχύνθηκε το μοντέλο, στην τάξη των 1.2 με 1.4 φορές για FP32 και FP16, ενώ για τους υπόλοιπους τρόπους κβαντοποίησης η επιτάχυνση συνήθως είναι μεγαλύτερη (φτάνει σε περιπτώσεις τον διπλασιασμό της απόδοσης) αλλά δεν είναι ενιαία.
- Παρατηρείται ότι η επιτάχυνση είναι περίπου της ίδιας τάξης για το ίδιο μοντέλο και τον ίδιο τρόπο κβαντοποίησης ανεξαρτήτως της συνάρτησης ενεργοποίησης που χρησιμοποιείται. Εξαιρέσεις αποτελούν οι συναρτήσεις ενεργοποίησης Leaky ReLU και Swish, στις οποίες παρατηρείται χαμηλότερη επιτάχυνση της εκτέλεσης για κάποια μοντέλα στους τρόπους κβαντοποίησης που χρησιμοποιούν ακέραια αριθμητική (FULL και INT).
- Το μοντέλο DistilBERT δείχνει να μην επωφελείται από τις βελτιστοποιήσεις όσον αφορά τον χρόνο εκτέλεσης του για FULL και INT κβαντοποιήσεις για όλες τις διαφορετικές συναρτήσεις ενεργοποίησης.
- Η επιτάχυνση των βελτιστοποιημένων μοντέλων σε σχέση με το αρχικό για εκτέλεση με GPU, είναι συνήθως μικρότερη από την αντίστοιχη επιτάχυνση για εκτέλεση σε CPU.
- Τα μοντέλα Xdistil και MiniLM, τα οποία χρησιμοποιούν την ίδια αρχιτεκτονική, αλλά έχουν εκπαιδευτεί με διαφορετική μέθοδο απόσταξης γνώσης, εμφανίζουν παρόμοια συμπεριφορά όσον αφορά την επιτάχυνση τους.

Επιτάχυνση Folded Μοντέλων στην Κινητή Συσκευή

Παρακάτω στον πίνακα 6.17, και στο σχήμα 6.3, παρατίθενται τα αποτελέσματα επιτάχυνσης των Folded μοντέλων, αρχικών και βελτιστοποιημένων σε CPU. Οι μετρήσεις έγιναν στην κινητή συσκευή Samsung A54 5G, Η επιτάχυνση που αναγράφεται είναι για εκτέλεση σε CPU με τον βέλτιστο αριθμό νημάτων σε κάθε περίπτωση. Λόγω της χαμηλής ακρίβειας για εκτέλεση σε GPU δεν θεωρήθηκε προωθητικό να ληφθούν μετρήσεις για την επιτάχυνση που επιτυγχάνεται με τη χρήση του Batch MatMul Folding.

Model	Quantization	CPU Average Speedup						
		GELU	ReLU	ReLU6	L. ReLU	Sigmoid	SiLU	Tanh
MobileBERT	FP32		0.97					
	FP16		0.97					
	DR		0.80					
	INT		1.09					
	FULL		1.08					
Xdistil-L12-H384	FP32	1.19	1.16	1.16	1.18	1.12	1.06	1.07
	FP16	1.22	1.19	1.19	1.20	1.14	1.08	1.17
	DR	1.03	1.08	0.90	0.92	0.84	0.84	0.81
	INT	1.25	1.20	1.21	0.96	1.14	1.16	1.47
	FULL	1.12	1.22	1.22	0.94	1.19	1.18	1.14
RoBERTa$_{TINY}$	FP32	1.63	1.26	1.36	1.28	1.05	1.09	1.22
	FP16	1.55	1.31	1.39	1.34	1.01	1.12	1.22
	DR	1.81	1.15	1.11	1.11	1.09	1.06	1.04
	INT	1.31	1.47	1.60	1.37	1.10	1.30	1.13
	FULL	1.31	1.53	1.87	1.34	1.00	1.29	1.19
DistilBERT	FP32	1.57	1.13			1.20	1.12	1.27
	FP16	1.31	1.07			1.23	1.30	1.27
	DR	1.76	1.07			1.05	1.07	1.06
	INT	1.39	1.41			1.43	1.45	1.43
	FULL	1.39	1.40			1.40	1.43	1.44
ELECTRA$_{SMALL}$	FP32	1.83	1.35	1.87	1.34	1.60	1.50	1.34
	FP16	1.84	1.39	1.94	1.36	1.58	1.52	1.38
	DR	1.84	1.08	1.10	1.15	0.99	1.03	1.11
	INT	1.22	1.54	1.53	1.03	1.54	1.47	1.57
	FULL	1.23	1.44	1.57	1.08	1.53	1.51	1.52
MiniLMv1-L12-H384	FP32	1.69	1.62	1.52	1.57	1.47	1.49	1.70
	FP16	1.67	1.69	1.50	1.56	1.54	1.48	1.59
	DR	1.67	1.12	1.13	1.20	1.09	1.07	1.17
	INT	1.25	1.55	1.56	1.23	1.53	1.49	1.58
	FULL	1.24	1.56	1.56	1.24	1.55	1.51	1.55

Πίνακας 6.17: Επιτάχυνση των Folded Μοντέλων στη CPU της κινητής συσκευή



Σχήμα 6.3: Μέσος όρος επιτάχυνσης για όλα τα Folded μοντέλα ανά συνάρτηση ενεργοποίησης και τρόπο κβαντοποίησης σε CPU

Παρατηρήσεις:

Παρατηρείται επιτάχυνση στην πλειοψηφία των περιπτώσεων χωρίς όμως να μπορεί να ληφθεί κάποιο συνολικό συμπέρασμα. Μοντέλα όπως το MobileBERT και το Xdistil δεν επωφελούνται σε ίδιο βαθμό με άλλα από τη χρήση του Batch MatMul Folding, αντίθετα μπορεί να επιδεινώνεται ο χρόνος εκτέλεσης, ενώ άλλα μοντέλα όπως το ELECTRA φτάνουν ακόμα και σε διπλασιασμό της απόδοσης. Η επιτάχυνση είναι υψηλότερη για τις αρχικές εκδόσεις των μοντέλων, με την συνάρτηση ενεργοποίησης GELU και την Κανονικοποίηση Παρτίδας σε σχέση με την βελτίωση στα βελτιστοποιημένα μοντέλα. Όσον αφορά τους τρόπους κβαντοποίησης παρατηρείται ότι για DR κβαντοποίηση η επιτάχυνση είναι ελάχιστη και είναι αισθητά μικρότερη από όλους τους υπόλοιπους τρόπους.

Κεφάλαιο 7ο

Επίλογος

Στο κεφάλαιο αυτό θα αναλυθούν τα συμπεράσματα που προέκυψαν από την αξιολόγηση που πραγματοποιήθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας, η συνεισφορά της στο ερευνητικό πεδίο και οι προεκτάσεις που μπορεί να έχει μελλοντικά.

7.1 Συμπεράσματα

Στο προηγούμενο κεφάλαιο παρουσιάστηκαν τα αποτελέσματα των μετρήσεων για τα επιλεγμένα μοντέλα, μελετήθηκε και αξιολογήθηκε η επίδοση τους και η επίδραση των μεθόδων βελτιστοποίησης που εφαρμόστηκαν στην ακρίβεια και τον χρόνο εκτέλεσης τους.

Οι σύγχρονες CPUs είναι ευέλικτες και υποστηρίζουν ένα ευρύτατο σύνολο πράξεων. Επιπλέον είναι πλέον διαδεδομένη η ύπαρξη πολλαπλών πυρήνων και συνεπώς ο χρόνος εκτέλεσης μπορεί να μειωθεί σημαντικά με παραλληλοποίηση των υπολογισμών. Παρατηρήθηκε ότι η χρήση πολλαπλών νημάτων μπορεί να επιφέρει μέχρι και διπλασιασμό της απόδοσης, χωρίς να απαιτείται κάποια τροποποίηση του αρχικού μοντέλου, επομένως μπορεί να λειτουργεί σαν προεπιλεγμένη ρύθμιση για μοντέλα τα οποία δεν έχουν δεχτεί άλλη βελτιστοποίηση. Ο συγκεκριμένος αριθμός παρόλα αυτά εξαρτάται από τα χαρακτηριστικά της συσκευής και του μοντέλου που εκτελείται, και για το λόγο αυτό θα πρέπει να πραγματοποιούνται εξιδεικευμένες μετρήσεις, όπως αυτές που πραγματοποιήθηκαν στο πλαίσιο της παρούσας διπλωματικής.

Για την περαιτέρω μείωση του χρόνου εκτέλεσης και του χώρου αποθήκευσης που απαιτείται στις κινητές συσκευές, μπορεί να χρησιμοποιηθεί κβαντοποίηση. Παρατηρήθηκε σημαντική επιτάχυνση για τους τρόπους κβαντοποίησης που χρησιμοποιούν ακέραια αριθμητική (DR, INT, FULL), ενώ παράλληλα μειώνεται στο ¼ ο απαιτούμενος χώρος αποθήκευσης.

Παρά την ευρεία υιοθέτηση και χρήση Μοντέλων Μετασχηματιστών σε Κινητές Συσκευές, το κομμάτι της εκτέλεσης τους σε επιταχυντές δεν έχει συμβαδίσει. Τόσο η βιβλιοθήκη XNNPack, η οποία δεν προσφέρει αρκετά σημαντικές επιταχύνσεις, όσο και οι GPUs, οι οποίες δεν επιταχύνουν όλες τις αριθμητικές και δεν υποστηρίζουν όλες τις πράξεις που χρησιμοποιούν οι Μετασχηματιστές, δεν αποτελούν συνολικές λύσεις και απαιτούνται περαιτέρω τροποποιήσεις στα μοντέλα.

Στο πλαίσιο αυτής της διπλωματικής εργασίας έγινε μία προσπάθεια να αντικατασταθούν τα επίπεδα και οι ενεργοποιήσεις που δεν υποστηρίζονται από τις πιο διαδεδομένες GPUs. Όπως διαπιστώθηκε, οι αντικαταστάσεις αυτές σε συνδυασμό, με το κατάλληλο fine-tuning μπορούν να φτάσουν ή ακόμη και να ξεπεράσουν σε ακρίβεια τα αρχικά μοντέλα. Ωστόσο από τις 6 συναρτήσεις ενεργοποίησης που δοκιμάστηκαν (ReLU, ReLU6, Leaky ReLU, Sigmoid, Swish, Tanh), δεν διαπιστώθηκε ενιαία διαφορά στην ακρίβεια για όλα τα μοντέλα σε σχέση με τα πρωτότυπα, οπότε η διαδικασία εύρεσης της βέλτιστης συνάρτησης ενεργοποίησης είναι αναγκάια.

Παρατηρείται ότι οι τροποποιήσεις μας επαναφέρουν την ακρίβεια για τη GPU, και επιτυγχάνουν επιτάχυνση της εκτέλεσης του μοντέλου. Η επιτάχυνση είναι περίπου ενιαία για όλα τα μοντέλα και τις συναρτήσεις ενεργοποίησης, όμως εμφανίζει διακυμάνσεις ανάλογα τον τρόπο κβαντοποίησης. Συμπεραίνουμε λοιπόν ότι τόσο η αρχιτεκτονική του μοντέλου όσο ο τρόπος μετεκπαίδευσης του, καθορίζουν την επίδραση που θα έχει η αλλαγή της συνάρτησης ενεργοποίησης και της μεθόδου κανονικοποίησης στην ακρίβεια και την επιτάχυνση, και απαιτείται εξατομίκευση για κάθε περίπτωση χρήσης.

Δοκιμάστηκε επίσης η χρήση Batch Matrix Multiplication Folding αντί για το προεπιλεγμένο Batch Matrix Multiplication Unfolding για τα αρχικά και τα βελτιστοποιημένα μοντέλα. Παρότι η χρήση Folded μοντέλων στη GPU δεν συνίσταται λόγω της πτώσης ακρίβειας που προκαλούν, επιταχύνουν την εκτέλεση σε CPU σημαντικά σε κάποιες περιπτώσεις.

7.2 Μελλοντικές Κατευθύνσεις

Η βελτιστοποίηση των Transformer μοντέλων για συμπερασματολογία σε κινητές συσκευές αποτελεί ένα πεδίο με ερευνητικό ενδιαφέρον και αρκετές προεκτάσεις. Βρίσκεται ακόμη σε αρχικό στάδιο και μπορεί να αναπτυχθεί και να συμβάλλει στην ανάπτυξη και την εξάπλωση τόσο του τομέα της Μηχανικής Μάθησης όσο και του Internet-of-Things.

Τα αποτελέσματα των μετρήσεων που πραγματοποιήθηκαν στο πλαίσιο της διπλωματικής εργασίας υποδεικνύουν ότι υπάρχει ακόμη ανάγκη για έρευνα στο πεδίο. Είναι συνεπώς σημαντικό να υπάρξουν εξελίξεις τόσο σε επίπεδο Μετασχηματιστών, όσο και σε επίπεδο Κινητών Συσκευών.

Οι αντικαταστάσεις που προτάθηκαν, σε συνδυασμό με τις ήδη υπάρχουσες μεθόδους για βελτιστοποίηση των μοντέλων όπως είναι η απόσταξη και η μεταφορά γνώσης, θέτουν μια καλή βάση για περαιτέρω έρευνα στον τομέα. Τα μοντέλα Μετασχηματιστών μέχρι στιγμής εστιάζουν στην βελτίωση της επίδοσης, μέσω της χρήσης όλο και μεγαλύτερων μοντέλων και συνόλων δεδομένων εκπαίδευσης. Θα πρέπει να δοθεί εξίσου μεγάλη έμφαση στην δημιουργία συμπαγών και αποτελεσματικών αρχιτεκτονικών. Ταυτόχρονα θα πρέπει να μελετηθεί η επίδραση της αντικατάστασης άλλων στοιχείων των μοντέλων με απλούστερα ώστε να γίνει ευκολότερη και ταχύτερη η εκτέλεση και να είναι δυνατή σε όλες τις διαθέσιμες υπολογιστικές μονάδες (CPU, GPU, NPU, κτλ.).

Τα ευρήματα μας είναι σημαντικά και θα πρέπει να επεκταθούν στα παραγωγικά (generative) μοντέλα, κύριο χαρακτηριστικό των οποίων είναι το πολύ μεγάλο μέγεθος τους. Μέχρι στιγμής το μέγεθος αυτό καθιστά απαγορευτική την εκτέλεση τους σε κινητές συσκευές. Καθώς όμως γίνονται όλο και πιο διαδεδομένα, θα πρέπει να μελετηθούν λύσεις συμπίεσης και βελτιστοποίησης τους, ώστε να γίνουν ελαφρύτερα και να καταστεί δυνατή η χρήση τους σε συσκευές παρυφών και σε πραγματικό χρόνο [65]–[67].

Μείζονος σημασίας είναι όμως να αναπτυχθούν και οι δυνατότητες του υλικού των κινητών συσκευών. Η κυκλοφορία συσκευών με ενσωματωμένη NPU είναι ένα σημαντικό βήμα, όμως θα πρέπει οι μονάδες αυτές να εξελιχθούν σε σημείο που να μπορούν να αντικαταστήσουν τις CPU ή και τις GPU για εκτέλεση μοντέλων Βαθιάς Μάθησης. Προς το παρόν το ποσοστό των επιπέδων

που μεταφέρονται σε NPU σε μία τυπική συσκευή είναι απαγορευτικά χαμηλό και περιορίζεται σε συγκεκριμένες περιπτώσεις. Επιπλέον θα πρέπει να υπάρξει έρευνα όσον αφορά την αναβάθμιση των δυνατοτήτων των ήδη υπαρχόντων επιταχυντών, όπως το XNNPack, ώστε μελετηθούν και πιθανόν να αναβαθμιστούν οι δυνατότητες τους και τα μοντέλα να σχεδιάζονται βάσει και αυτών.

Για το υπάρχον υλικό μπορούν να δοκιμαστούν τεχνικές κατανομής του υπολογισμού μεταξύ CPU και GPU (δυνητικά και NPU), ώστε να εκτελούνται στη CPU διεργασίες και πράξεις, που δεν υποστηρίζονται από τη GPU (π.χ. LayerNorm, Batch MatMul). Με αυτόν τον τρόπο θα συνδυαστεί η ευελιξία και η αποτελεσματικότητα των CPUs με την ταχύτητα των GPUs, ώστε να επιτευχθούν βέλτιστα αποτελέσματα.

Τέλος, παρατηρήθηκε σημαντική επιτάχυνση των μοντέλων με τη χρήση πολλαπλών νημάτων για παραλληλοποίηση των υπολογισμών. Στις μετρήσεις μας τα 4 νήματα είχαν καλύτερη απόδοση από τα 8, συνεπώς υπάρχει βελτίωση και καλύτερη αξιοποίηση των πόρων. Περαιτέρω έρευνα μπορεί να πραγματοποιηθεί πάνω στην ταυτόχρονη εκτέλεση πολλαπλών παρτίδων δειγμάτων (batches) και στην παραλληλοποίηση του υπολογισμού αυτών, ώστε να έχουμε ακόμη καλύτερη απόδοση.

Αναφορές

[1] I. El Naqa and M. J. Murphy, '***What is machine learning?***', *Machine Learning in Radiation Oncology: Theory and Applications*, σσ. 3–11, 2015.

[2] '***artificial intelligence | Britannica***'. Ημερομηνία πρόσβασης: 18 Οκτώβριος 2023. Διαθέσιμο στο: https://www.britannica.com/technology/artificial-intelligence

[3] C. C. Aggarwal, ***Neural Networks and Deep Learning***. Cham: Springer International Publishing, 2018.

[4] A. Vaswani *κ.ά.*, '***Attention Is All You Need***', Ιουνίου 2017

[5] '***Attention is all you need: Discovering the Transformer paper | Medium***'. Ημερομηνία πρόσβασης: 15 Σεπτέμβριος 2023. Διαθέσιμο στο: https://towardsdatascience.com/attention-is-all-you-need-discovering-the-transformer-paper-73e5ff5e0634

[6] J. L. Ba, J. R. Kiros, και G. E. Hinton, '***Layer Normalization***', Ιουλίου 2016

[7] '***PyTorch Documentation, Activation Function***'. Ημερομηνία πρόσβασης: 18 Οκτώβριος 2023. Διαθέσιμο στο: https://pytorch.org/docs/stable/nn.html

[8] '***Activation Functions in Neural Networks [12 Types & Use Cases] | V7***'. Ημερομηνία πρόσβασης: 18 Οκτώβριος 2023. Διαθέσιμο στο: https://www.v7labs.com/blog/neural-networks-activation-functions

[9] D. Hendrycks και K. Gimpel, '***Gaussian Error Linear Units (GELUs)***', Ιουνίου 2016

[10] '***What Is Natural Language Processing (NLP)? | AWS***'. Ημερομηνία πρόσβασης: 12 Σεπτέμβριος 2023. Διαθέσιμο στο: https://aws.amazon.com/what-is/nlp/

[11] '***What is natural language processing? | IBM***'. Ημερομηνία πρόσβασης: 18 Οκτώβριος 2023. Διαθέσιμο στο: www.ibm.com/topics/natural-language-processing

[12] A. Rahali και M. A. Akhloufi, '***End-to-End Transformer-Based Models in Textual-Based NLP***', *AI*, τ. 4, τχ. 1, σσ. 54–110, Ιανουαρίου 2023,

[13] N. Patwardhan, S. Marrone, και C. Sansone, '***Transformers in the Real World: A Survey on NLP Applications***', *Information*, τ. 14, τχ. 4, σ. 242, Απριλίου 2023,

[14] '***What is Transliteration? How is it different from Translation? | Medium***', Ημερομηνία πρόσβασης: 12 Σεπτέμβριος 2023. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://medium.com/neuralspace/what-is-transliteration-how-is-it-different-from-translation-16fff6c3e0cc

[15] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, και H. Jégou, '***Word Translation Without Parallel Data***', Οκτωβρίου 2017

[16] '***Unsupervised machine translation: A novel approach to provide fast, accurate translations for more languages | Engineering at Meta***'. Ημερομηνία πρόσβασης: 12 Σεπτέμβριος 2023. Διαθέσιμο στο: https://engineering.fb.com/2018/08/31/ai-research/unsupervised-machine-translation-a-novel-approach-to-provide-fast-accurate-translations-for-more-languages/

Αναφορές

[17] J. Rudolph, S. Tan, και S. Tan, '***War of the chatbots: Bard, Bing Chat, ChatGPT, Ernie and beyond. The new AI gold rush and its impact on higher education***', *Journal of Applied Learning & Teaching*, τ. 6, τχ. 1, σσ. 364–389, Απριλίου 2023,

[18] G. Lai, Q. Xie, H. Liu, Y. Yang, και E. Hovy, '***RACE: Large-scale ReAding Comprehension Dataset From Examinations***', Απριλίου 2017

[19] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, και S. Bowman, '***GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding***', στο *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, σσ. 353–355.

[20] P. Rajpurkar, R. Jia, και P. Liang, '***Know What You Don't Know: Unanswerable Questions for SQuAD***', Ιουνίου 2018

[21] '***Introduction to the Architecture of Recurrent Neural Networks (RNNs) | Medium***'. Ημερομηνία πρόσβασης: 15 Σεπτέμβριος 2023. Διαθέσιμο στο: https://pub.towardsai.net/introduction-to-the-architecture-of-recurrent-neural-networks-rnns-a277007984b7

[22] '***An Intuitive Explanation of LSTM | Medium***'. Ημερομηνία πρόσβασης: 15 Σεπτέμβριος 2023. Διαθέσιμο στο: https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c

[23] '***Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies***', στο *A Field Guide to Dynamical Recurrent Networks*, IEEE, 2009.

[24] A. G. Howard *κ.ά.*, '***MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications***', Απριλίου 2017

[25] A. Gholami *κ.ά.*, '***SqueezeNext: Hardware-Aware Neural Network Design***', Μαρτίου 2018

[26] I. Panopoulos, S. Nikolaidis, S. I. Venieris, και I. S. Venieris, '***Exploring the Performance and Efficiency of Transformer Models for NLP on Mobile Devices***', Ιουνίου 2023

[27] M. Almeida, S. Laskaridis, A. Mehrotra, L. Dudziak, I. Leontiadis, και N. D. Lane, '***Smart at what cost? Characterising Mobile Deep Neural Networks in the wild***', Σεπτεμβρίου 2021

[28] A. Krizhevsky, I. Sutskever, και G. E. Hinton, '***ImageNet Classification with Deep Convolutional Neural Networks***', *Adv Neural Inf Process Syst*, τ. 25, 2012

[29] '***ImageNet Large Scale Visual Recognition Challenge (ILSVRC)***'. Ημερομηνία πρόσβασης: 23 Οκτώβριος 2023. Διαθέσιμο στο: https://image-net.org/challenges/LSVRC/

[30] K. Simonyan και A. Zisserman, '***Very Deep Convolutional Networks for Large-Scale Image Recognition***', Σεπτεμβρίου 2014

[31] J. Devlin, M.-W. Chang, K. Lee, και K. Toutanova, '***BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding***', Οκτωβρίου 2018

[32] T. B. Brown *κ.ά.*, '***Language Models are Few-Shot Learners***', Μαΐου 2020

[33] OpenAI, '***GPT-4 Technical Report***', Μαρτίου 2023

[34] '***Knowledge Distillation: Principles, Algorithms, Applications | Neptune AI***'. Ημερομηνία πρόσβασης: 11 Οκτώβριος 2023. Διαθέσιμο στο: https://neptune.ai/blog/knowledge-distillation

Αναφορές

[35] '***Quantization and Pruning | Scaler***'. Ημερομηνία πρόσβασης: 18 Σεπτέμβριος 2023. Διαθέσιμο στο: https://www.scaler.com/topics/quantization-and-pruning/

[36] '***TensorFlow Lite Delegates***'. Ημερομηνία πρόσβασης: 24 Οκτώβριος 2023. Διαθέσιμο στο: https://www.tensorflow.org/lite/performance/delegates

[37] '***XNNPack | GitHub***', Ημερομηνία πρόσβασης: 24 Οκτώβριος 2023. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://github.com/google/XNNPACK

[38] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, και K. Keutzer, '***SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size***', Φεβρουαρίου 2016

[39] M. Tan και Q. V. Le, '***EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks***', Μαΐου 2019

[40] '***MobileBERT | Hugging Face***'. Ημερομηνία πρόσβασης: 25 Σεπτέμβριος 2023. Διαθέσιμο στο: https://huggingface.co/docs/transformers/model_doc/mobilebert

[41] '***Post-training float16 quantization | TensorFlow***'. Ημερομηνία πρόσβασης: 24 Σεπτέμβριος 2023. Διαθέσιμο στο: https://www.tensorflow.org/lite/performance/post_training_float16_quant

[42] '***Post-training quantization | TensorFlow***'. Ημερομηνία πρόσβασης: 24 Σεπτέμβριος 2023. Διαθέσιμο στο: www.tensorflow.org/lite/performance/post_training_quantization

[43] '***What Is TensorFlow Batch Matmul and How Does It Work? | SaturnCloud***'. Ημερομηνία πρόσβασης: 24 Σεπτέμβριος 2023. Διαθέσιμο στο: https://saturncloud.io/blog/what-is-tensorflow-batch-matmul-and-how-does-it-work/#what-is-batch_matmul

[44] '***TensorFlow***'. Ημερομηνία πρόσβασης: 27 Σεπτέμβριος 2023. Διαθέσιμο στο: https://www.tensorflow.org/about

[45] '***TensorFlow Lite***'. Ημερομηνία πρόσβασης: 27 Σεπτέμβριος 2023. Διαθέσιμο στο: https://www.tensorflow.org/lite/guide

[46] '***Keras***'. Ημερομηνία πρόσβασης: 27 Σεπτέμβριος 2023. Διαθέσιμο στο: https://keras.io

[47] '***Model Conversion Overview | TensorFlow Lite***'. Ημερομηνία πρόσβασης: 27 Σεπτέμβριος 2023. Διαθέσιμο στο: www.tensorflow.org/lite/models/convert

[48] '***Hugging Face | Wikipedia***'. Ημερομηνία πρόσβασης: 27 Σεπτέμβριος 2023. Διαθέσιμο στο: https://en.wikipedia.org/wiki/Hugging_Face

[49] '***Android Studio***'. Ημερομηνία πρόσβασης: 27 Σεπτέμβριος 2023. Διαθέσιμο στο: https://www.techtarget.com/searchmobilecomputing/definition/Android-Studio

[50] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, και D. Zhou, '***MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices***', Απριλίου 2020

[51] K. Clark, M.-T. Luong, Q. V. Le, και C. D. Manning, '***ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators***', Μαρτίου 2020

[52] '***ELECTRA | Hugging Face***'. Ημερομηνία πρόσβασης: 27 Σεπτέμβριος 2023. Διαθέσιμο στο: https://huggingface.co/docs/transformers/model_doc/electra

[53] V. Sanh, L. Debut, J. Chaumond, και T. Wolf, '***DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter***', Οκτωβρίου 2019

Αναφορές

[54] '***DistilBERT| Hugging Face***'. Ημερομηνία πρόσβασης: 25 Σεπτέμβριος 2023. Διαθέσιμο στο: https://huggingface.co/docs/transformers/model_doc/distilbert

[55] Y. Liu *κ.ά.*, '***RoBERTa: A Robustly Optimized BERT Pretraining Approach***', Ιουλίου 2019

[56] '***RoBERTa | Hugging Face***'. Ημερομηνία πρόσβασης: 26 Σεπτέμβριος 2023. Διαθέσιμο στο: https://huggingface.co/docs/transformers/model_doc/roberta

[57] '***Overview of ROBERTa model | GeeksForGeeks***'. Ημερομηνία πρόσβασης: 26 Σεπτέμβριος 2023. Διαθέσιμο στο: https://www.geeksforgeeks.org/overview-of-roberta-model

[58] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, και M. Zhou, '***MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers***', *Adv Neural Inf Process Syst*, τ. 33, σσ. 5776–5788, 2020

[59] '***MiniLM***'. Ημερομηνία πρόσβασης: 27 Σεπτέμβριος 2023. Διαθέσιμο στο: http://mohitmayank.com/a_lazy_data_science_guide/natural_language_processing/minilm/

[60] S. Mukherjee, A. H. Awadallah, και J. Gao, '***XtremeDistilTransformers: Task Transfer for Task-agnostic Distillation***', Ιουνίου 2021

[61] '***Emotions Dataset for NLP | Kaggle***'. Ημερομηνία πρόσβασης: 25 Σεπτέμβριος 2023. Διαθέσιμο στο: https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp

[62] '***Classification: Accuracy***'. Ημερομηνία πρόσβασης: 27 Σεπτέμβριος 2023. Διαθέσιμο στο: https://developers.google.com/machine-learning/crash-course/classification/accuracy

[63] '***Floating-Point Operations Per Second (FLOPS) | WikiChip***'. Ημερομηνία πρόσβασης: 18 Οκτώβριος 2023. Διαθέσιμο στο: https://en.wikichip.org/wiki/flops

[64] '***NVIDIA Tesla P100***'. Ημερομηνία πρόσβασης: 30 Σεπτέμβριος 2023. Διαθέσιμο στο: https://www.nvidia.com/en-us/data-center/tesla-p100/

[65] W. Niu *κ.ά.*, '***Real-Time Execution of Large-scale Language Models on Mobile***', Σεπτεμβρίου 2020

[66] Y.-H. Chen *κ.ά.*, '***Speed Is All You Need: On-Device Acceleration of Large Diffusion Models via GPU-Aware Optimizations***', Απριλίου 2023

[67] Y. Li *κ.ά.*, '***SnapFusion: Text-to-Image Diffusion Model on Mobile Devices within Two Seconds***', Ιουνίου 2023