



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Εφαρμοσμένων Μαθηματικών  
και Φυσικών Επιστημών  
Τομέας Μαθηματικών

# Μελέτη και Αξιολόγηση Μεθόδων Συμπύεσης Βαθιών Νευρωνικών Δικτύων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μαγδαληνή Παύλου

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

Αθήνα, Φεβρουάριος 2023





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Εφαρμοσμένων Μαθηματικών  
και Φυσικών Επιστημών  
Τομέας Μαθηματικών

# Μελέτη και Αξιολόγηση Μεθόδων Συμπύεσης Βαθιών Νευρωνικών Δικτύων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μαγδαληνή Παύλου

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

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 23<sup>η</sup> Φεβρουαρίου 2023.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....  
Ιάκωβος Βενιέρης  
Καθηγητής Ε.Μ.Π.

.....  
Δήμητρα Θεοδώρα Κακλαμάνη  
Καθηγήτρια Ε.Μ.Π.

.....  
Αντώνιος Συμβώνης  
Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2023





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Εφαρμοσμένων Μαθηματικών  
και Φυσικών Επιστημών  
Τομέας Μαθηματικών

Copyright ©–All rights reserved  
Μαγδαληνή Παύλου, 2023.

Με επιφύλαξη παντός δικαιώματος. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τη συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τη συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

(Υπογραφή)

.....

Μαγδαληνή Παύλου  
23 Φεβρουαρίου 2023



# Περίληψη

Η εκτέλεση μοντέλων νευρωνικών δικτύων τοπικά, σε μια συσκευή, είναι ωφέλιμη, τόσο σε θέματα ιδιωτικότητας, όσο και στην αποφυγή επιπλέον καθυστερήσεων, που θα οφείλονταν στη μεταφορά των δεδομένων από και προς έναν απομακρυσμένο εξυπηρετητή. Στη σημερινή εποχή, τα μοντέλα νευρωνικών δικτύων που χρησιμοποιούνται στις σύγχρονες εφαρμογές έχουν σημαντικά υψηλή απόδοση, η οποία σε προβλήματα όπως η ταξινόμηση, μπορεί ακόμα να συγκριθεί ή και να ξεπεράσει την ανθρώπινη. Για να επιτευχθεί όμως η επιθυμητή αυτή ακρίβεια, ο αριθμός των παραμέτρων που περιέχουν τα εν λόγω μοντέλα είναι αρκετά μεγάλος και αυτό έχει αρνητικές συνέπειες, τόσο στο υπολογιστικό κόστος, αφού ο αριθμός των πράξεων που εκτελούνται είναι τεράστιος, όσο και στο χώρο που αυτά καταλαμβάνουν σε μια συσκευή. Σκοπός της παρούσας διπλωματικής εργασίας είναι η μελέτη και αξιολόγηση αλγορίθμων βελτιστοποίησης νευρωνικών δικτύων, όπως η κβαντοποίηση, το κλάδεμα και η ομαδοποίηση ή και συνδυασμών αυτών, οι οποίοι αποσκοπούν στη μείωση του χώρου που καταλαμβάνουν τα μοντέλα και του χρόνου απόκρισής τους κατά την εκτέλεση της συμπερασματολογίας, επιτυγχάνοντας ταυτόχρονα τη μικρότερη δυνατή πτώση στην ακρίβεια.

Για τις ανάγκες των πειραμάτων που διεξήχθησαν, συγκεντρώθηκε ένας αριθμός από συνεκτικά μοντέλα που επιλύουν το πρόβλημα της κατηγοριοποίησης εικόνων, τα οποία είναι ήδη εκπαιδευμένα στο σύνολο δεδομένων *ImageNet*, στο οποίο οι εικόνες αντιστοιχίζονται σε 1000 κλάσεις. Οι μετρήσεις που πάρθηκαν κατά τη διάρκεια της διπλωματικής εργασίας αφορούν την ακρίβεια, το μέγεθος, το πλήθος των πράξεων κινητής υποδιαστολής και το χρόνο εκτέλεσης των μοντέλων πριν και μετά την εφαρμογή των αλγορίθμων σε αυτά, για διάφορες τιμές των παραμέτρων που αντιστοιχούν σε καθέναν από αυτούς.

Με βάση τα τελικά αποτελέσματα, προκύπτει το συμπέρασμα ότι οι υπό μελέτη αλγόριθμοι συμπίεσης παρουσιάζουν ποικίλα πλεονεκτήματα, όσον αφορά το μέγεθος και τον χρόνο απόκρισης των μοντέλων, χωρίς να διακυβεύεται σημαντικά η ακρίβειά τους. Μια τέτοια παρατήρηση είναι ζωτικής σημασίας, καθώς επιτρέπει την καθιέρωση ενός κριτηρίου που μπορεί να χρησιμοποιηθεί για τον προσδιορισμό του καταλληλότερου αλγόριθμου συμπίεσης και των βέλτιστων παραμέτρων του, ανάλογα με τις απαιτήσεις επιτάχυνσης και συμπίεσης του προβλήματος, το οποίο, στη δική μας περίπτωση, είναι κάποια εφαρμογή για έξυπνες κινητές συσκευές.

## **Λέξεις Κλειδιά:**

Βαθιά Μάθηση, Αλγόριθμοι Συμπίεσης, Κλάδεμα, Ομαδοποίηση, Κβαντοποίηση, Μεταφορά Γνώσης, Κατηγοριοποίηση Εικόνων





# Abstract

Executing neural network models locally, on a device, is beneficial both in terms of privacy and in avoiding additional delays that would be caused by transferring data to and from a remote server. In today's era, the neural network models used in modern applications have significantly high performance, which in problems such as classification, can be compared to or even exceed human performance. However, in order to achieve this desired accuracy, the number of parameters contained in these models is fairly large and this has negative consequences, both in terms of computational cost, since the number of operations is huge, and in the space they occupy on a device. The aim of this thesis is to study and evaluate neural network optimization algorithms, such as quantization, pruning and clustering or combinations of the above, which aim to reduce the space occupied by the models and their response time during inference, while achieving the smallest possible reduction in terms of accuracy.

For the needs of the experiments conducted, a number of convolutional models solving the image classification problem were collected, which are already trained on the ImageNet dataset, in which images are assigned to 1000 classes. The measurements obtained during the thesis project concern the accuracy, size, number of floating-point operations (FLOPs) and execution time of the models before and after applying the optimization algorithms, for different values of the parameters corresponding to each one of them.

Based on the final results, it can be inferred that the compression algorithms under study exhibit varied advantages concerning the size and latency of the models, without significantly compromising their accuracy. Such an observation is crucial since it allows for the establishment of a criterion that can be utilized to determine the most suitable compression algorithm and its optimal parameters, depending on the acceleration and compression requirements of the problem at hand, which, in our case, is a mobile application.

**Keywords:**

Deep Learning, Compression Algorithms, Pruning, Clustering, Quantization, Transfer Learning, Image Classification



# Ευχαριστίες

Στο σημείο αυτό, θα ήθελα να ευχαριστήσω όλους τους ανθρώπους οι οποίοι στάθηκαν δίπλα μου και με βοήθησαν σε όλη τη διάρκεια των σπουδών μου.

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή κύριο Ιάκωβο Βενιέρη, ο οποίος μου έδωσε την ευκαιρία να ασχοληθώ με ένα πολύ ενδιαφέρον αντικείμενο, στο οποίο μάθαινα καινούρια πράγματα καθ' όλη τη διάρκεια της ενασχόλησής μου με αυτό, καθώς και το εργαστήριο Ευφών Επικοινωνιών και Δικτύων Ευρείας Ζώνης. Επίσης, θα ήθελα να ευχαριστήσω τους καθηγητές κύριο Αντώνιο Συμβώνη και κυρία Δήμητρα Θεοδώρα Κακλαμάνη, για τη συμμετοχή τους στην τριμελή επιτροπή.

Έπειτα, θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα της ΣΗΜΜΥ, κύριο Ιωάννη Πανόπουλο, για τη στήριξη, το ενδιαφέρον, τις ιδέες και τη βοήθεια που μου παρείχε καθ' όλη τη διάρκεια ενασχόλησής μου με τη διπλωματική εργασία.

Επίσης, ευχαριστώ την οικογένειά μου, για τη στήριξη που μου παρείχε όλα αυτά τα χρόνια. Τέλος, ευχαριστώ τους φίλους μου, οι οποίοι ήταν πάντα στο πλευρό μου, αλλά και όλους τους ανθρώπους τους οποίους συνάντησα και περάσαμε χρόνο μαζί, για όλες τις όμορφες στιγμές που μου χάρισαν, τις οποίες θα θυμάμαι για πάντα!

Αθήνα, Φεβρουάριος 2023

Μαγδαληνή Παύλου



# Πίνακας Περιεχομένων

Περίληψη.....	1
Abstract.....	1
Κατάλογος Σχημάτων.....	1
Κατάλογος Πινάκων.....	1
Πρόλογος.....	1
1. Βαθιά Μάθηση.....	2
1.1 Τεχνητά Νευρωνικά Δίκτυα.....	2
1.1.1 Νευρώνας McCulloch - Pitts.....	3
1.1.2 Perceptron.....	4
1.1.3 ADALINE.....	8
1.1.4 Perceptron Πολλαπλών Επιπέδων.....	9
1.2 Εκπαίδευση και Εκμάθηση Βαρών.....	12
1.2.1 Επιβλεπόμενη Μάθηση.....	12
1.2.2 Θεωρία Βελτιστοποίησης.....	14
1.2.3 Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων.....	15
1.3 Άλλα Είδη Μάθησης.....	22
1.3.1 Μη Επιβλεπόμενη Μάθηση.....	23
1.3.2 Ενισχυτική Μάθηση.....	24
1.3.3 Μεταφορά Γνώσης.....	24
1.4 Συνελκτικά Νευρωνικά Δίκτυα.....	27
1.4.1 Συνέλιξη.....	28
1.4.2 Δομικά Στοιχεία Συνελκτικών Δικτύων.....	29
2. Μέθοδοι Συμπίεσης Βαθιών Νευρωνικών Δικτύων.....	35
2.1 Κλάδεμα.....	35
2.2 Κβαντοποίηση.....	37
2.3 Ομαδοποίηση – Διαμοιρασμός Βαρών.....	39
3. Τεχνολογίες.....	41
3.1 Android.....	41
3.2 Android Studio.....	42
3.3 Java.....	42
3.4 Python.....	43
3.4.1 Keras.....	43
3.5 TensorFlow Lite.....	44
3.6 ImageNet.....	45
4. Περιβάλλον Αξιολόγησης.....	46
4.1 Σύνολα Δεδομένων.....	46
4.1.1 Μεταφορά Γνώσης – Διαδικασία.....	47
4.2 Αρχιτεκτονικές Μοντέλων Βαθιάς Μάθησης.....	47
4.3 Μετρικές.....	51
4.4 Κινητή Συσκευή.....	52
4.5 Παράμετροι Αλγορίθμων Συμπίεσης.....	53
4.5.1 Κλάδεμα.....	53
4.5.2 Κβαντοποίηση.....	55
4.5.3 Ομαδοποίηση.....	56
5. Αξιολόγηση Μεθόδων Συμπίεσης.....	57
5.1 ImageNet.....	57

5.2 CIFAR10.....	62
5.2.1 Κλάδεμα.....	63
5.2.2 Ομαδοποίηση.....	68
6. Επίλογος.....	78
6.1 Συμπεράσματα.....	78
6.2 Μελλοντική Εργασία.....	80
Παράρτημα Α.....	83
Παράρτημα Β.....	89
Ξενογλώσση Ορολογία.....	99
Βιβλιογραφία.....	103

## Κατάλογος Σχημάτων

Σχήμα 1: Νευρώνας McCulloch – Pitts [21].....	3
Σχήμα 2: Επίπεδο διαχωρισμού για τη συνάρτηση OR [21].....	4
Σχήμα 3: Βηματική Συνάρτηση - Step Function [13].....	5
Σχήμα 4: Ευθεία Διαχωρισμού Δύο Κλάσεων [24].....	6
Σχήμα 5: Γενική Μορφή Νευρώνα [8].....	9
Σχήμα 6: Κατανομή των δεδομένων σύμφωνα με τη συνάρτηση XOR [26].....	9
Σχήμα 7: Μοντέλο Τεχνητών Νευρωνικών Δικτύων που προσεγγίζει τη συνάρτηση XOR [62].....	10
Σχήμα 8: Σιγμοειδής [10].....	11
Σχήμα 9: ReLU [10].....	11
Σχήμα 10: Μετασχηματισμός Δεδομένων Εισόδου σε Γραμμικά Διαχωρίσιμα [1].....	12
Σχήμα 11: Αντιστάθμιση Μεροληψίας και Διασποράς σε σχέση με το Συνολικό Σφάλμα [38].....	14
Σχήμα 12: Μέθοδος Ομαλής Καθόδου - Γραφική Αναπαράσταση Δύο Διαστάσεων [2].....	17
Σχήμα 13: Οπισθοδιάδοση Σφάλματος (Backpropagation) [20].....	20
Σχήμα 14: Παράδειγμα Συνέλιξης σε Δύο Διαστάσεις [34].....	29
Σχήμα 15: Μοντέλο Συνελκτικών Δικτύων για Ταξινόμηση [35].....	34
Σχήμα 16: Μη Δομικό και Δομικό Κλάδεμα - Γραφική Αναπαράσταση [18].....	36
Σχήμα 17: Μη Δομικό και Δομικό Κλάδεμα Αναπαράσταση Πινάκων [19].....	36
Σχήμα 18: Συμμετρική και Μη Συμμετρική Κβαντοποίηση με Αναπαράσταση σε 8 bits [42].....	39
Σχήμα 19: Επίδραση Αλγόριθμων Κβαντοποίησης στην Ακρίβεια.....	58
Σχήμα 20: Επίδραση Αλγόριθμων Κβαντοποίησης στο Μέγεθος.....	58
Σχήμα 21: Κατανομή του Χρόνου Απόκρισης - Κβαντοποίηση.....	59
Σχήμα 22: Κατανομή του Χρόνου Απόκρισης - Κβαντοποίηση.....	59
Σχήμα 23: Κατανομή του Χρόνου Απόκρισης - Κβαντοποίηση.....	60
Σχήμα 24: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Κβαντοποίηση.....	61
Σχήμα 25: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Κβαντοποίηση.....	61
Σχήμα 26: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Κβαντοποίηση.....	61
Σχήμα 27: Κλάδεμα – Ελαφριά Μοντέλα (Ποσοστό Αραιότητας - Ακρίβεια, Μέγεθος).....	65
Σχήμα 28: Κλάδεμα – Βαριά Μοντέλα (Ποσοστό Αραιότητας - Ακρίβεια, Μέγεθος).....	65
Σχήμα 29: Κατανομή του Χρόνου Απόκρισης - Κλάδεμα.....	67
Σχήμα 30: Κατανομή του Χρόνου Απόκρισης - Κλάδεμα.....	67
Σχήμα 31: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Κλάδεμα.....	68
Σχήμα 32: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Κλάδεμα.....	68
Σχήμα 33: Σύγκριση της Επίδρασης των Αλγορίθμων Αρχικοποίησης ως προς την Ακρίβεια.....	69
Σχήμα 34: Σύγκριση της Επίδρασης της Επανεκπαίδευσης ανά Αλγόριθμο Αρχικοποίησης.....	70
Σχήμα 35: Σύγκριση της Επίδρασης του Πλήθους Κεντροειδών ως προς την Ακρίβεια.....	70
Σχήμα 36: Ομαδοποίηση - Βαριά Μοντέλα (Ακρίβεια - Μέγεθος).....	72
Σχήμα 37: Ομαδοποίηση - Ελαφριά Μοντέλα (Ακρίβεια - Μέγεθος).....	72
Σχήμα 38: Κατανομή του Χρόνου Απόκρισης - Ομαδοποίηση.....	74
Σχήμα 39: Κατανομή του Χρόνου Απόκρισης - Ομαδοποίηση.....	74
Σχήμα 40: Κατανομή του Χρόνου Απόκρισης - Ομαδοποίηση.....	74
Σχήμα 41: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Ομαδοποίηση.....	75
Σχήμα 42: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Ομαδοποίηση.....	75
Σχήμα 43: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Ομαδοποίηση.....	76





## Κατάλογος Πινάκων

Πίνακας 1: Χαρακτηριστικά Μοντέλων Βαθιάς Μάθησης.....	50
Πίνακας 2: Χαρακτηριστικά Κινητής Συσκευής.....	52
Πίνακας 3: Επίδραση Ομαδοποίησης στην Επιτάχυνση.....	77
Πίνακας 4: Τελικά Συμπεράσματα.....	80
Πίνακας 5: Ακρίβεια Μοντέλων (ImageNet).....	83
Πίνακας 6: Μέγεθος Μοντέλων (ImageNet).....	84
Πίνακας 7: Χρόνος Απόκρισης (ms) - CPU (ImageNet).....	84
Πίνακας 8: Χρόνος Απόκρισης (ms) - GPU (ImageNet).....	85
Πίνακας 9: Χρόνος Απόκρισης (ms) - NNAPI (ImageNet).....	85
Πίνακας 10: Χρόνος Απόκρισης (ms) – Dynamic Range Quantization - CPU (ImageNet).....	85
Πίνακας 11: Χρόνος Απόκρισης (ms) - Dynamic Range Quantization - GPU (ImageNet).....	86
Πίνακας 12: Χρόνος Απόκρισης (ms) - Dynamic Range Quantization - NNAPI (ImageNet).....	86
Πίνακας 13: Χρόνος Απόκρισης (ms) – Float Fallback Quantization - CPU (ImageNet).....	86
Πίνακας 14: Χρόνος Απόκρισης (ms) - Float Fallback Quantization - GPU (ImageNet).....	87
Πίνακας 15: Χρόνος Απόκρισης (ms) - Float Fallback Quantization - NNAPI (ImageNet).....	87
Πίνακας 16: Χρόνος Απόκρισης (ms) - Full Integer - CPU (ImageNet).....	87
Πίνακας 17: Χρόνος Απόκρισης (ms) - Full Integer - GPU (ImageNet).....	87
Πίνακας 18: Χρόνος Απόκρισης (ms) - Full Integer - NNAPI (ImageNet).....	88
Πίνακας 19: Ακρίβεια Μοντέλων (CIFAR10).....	90
Πίνακας 20: Μέγεθος Μοντέλων (CIFAR10).....	92
Πίνακας 21: Χρόνος Απόκρισης (ms) - CPU (CIFAR10).....	93
Πίνακας 22: Χρόνος Απόκρισης (ms) - GPU (CIFAR10).....	93
Πίνακας 23: Χρόνος Απόκρισης (ms) - NNAPI (CIFAR10).....	94
Πίνακας 24: Χρόνος Απόκρισης (ms) - Κλάδεμα - CPU (CIFAR10).....	94
Πίνακας 25: Χρόνος Απόκρισης (ms) - Κλάδεμα - GPU (CIFAR10).....	95
Πίνακας 26: Χρόνος Απόκρισης (ms) - Κλάδεμα - NNAPI (CIFAR10).....	95
Πίνακας 27: Χρόνος Απόκρισης (ms) - Ομαδοποίηση - CPU (CIFAR10).....	96
Πίνακας 28: Χρόνος Απόκρισης (ms) - Ομαδοποίηση - GPU (CIFAR10).....	96
Πίνακας 29: Χρόνος Απόκρισης (ms) - Ομαδοποίηση - NNAPI (CIFAR10).....	97



# Πρόλογος

Τα βαθιά νευρωνικά δίκτυα αποτελούν έναν γρήγορα αναπτυσσόμενο τομέα ερευνητικά, αφού πετυχαίνουν σημαντικά υψηλή απόδοση σε δύσκολες εργασίες όπως όρασης υπολογιστών, επεξεργασίας φυσικής γλώσσας και λήψης αποφάσεων. Τα παραπάνω θέματα βρίσκουν εφαρμογές σε πολλούς τομείς της καθημερινής ζωής. Ένα παράδειγμα είναι η ανίχνευση ανωμαλιών σε μια εικόνα. Στο πλαίσιο της βιομηχανίας, αυτό μπορεί να σημαίνει την ανάλυση δεδομένων αισθητήρων με στόχο τον εντοπισμό απροσδόκητων αλλαγών σε κραδασμούς ή στη θερμοκρασία, που μπορεί να υποδηλώνουν πιθανή βλάβη του εξοπλισμού, ενώ στο πλαίσιο της ιατρικής μπορεί να μεταφραστεί ως την ανίχνευση όγκων σε ακτινογραφίες ή μαγνητικές τομογραφίες. Ανάγκη για χρήση της τεχνητής νοημοσύνης υπάρχει και στις εφαρμογές οι οποίες είναι ενσωματωμένες σε κινητές συσκευές.

Η εκτέλεση των υπολογισμών των μοντέλων τοπικά, δηλαδή σε μία συσκευή, μπορεί να έχει αρκετά οφέλη. Αρχικά, δεν είναι απαραίτητη η σύνδεση στο διαδίκτυο. Έπειτα, διασφαλίζεται η προστασία των προσωπικών δεδομένων, αφού τα δεδομένα του χρήστη δεν μεταφέρονται μέσω του διαδικτύου σε έναν απομακρυσμένο εξυπηρετητή, ώστε να εκτελεστούν σε αυτόν οι υπολογισμοί. Τέλος, δεν υπάρχει επιπλέον καθυστέρηση που να αφορά στην απόκριση ενός μοντέλου, αφού ο χρόνος αναμονής για την αποστολή ή λήψη δεδομένων ή παραμέτρων του μοντέλου από και προς τη συσκευή είναι μηδενικός.

Παρ' όλα αυτά, η χρήση εφαρμογών που περιλαμβάνουν μοντέλα βαθιάς μάθησης σε κινητές συσκευές, δεν είναι ένα τετριμμένο θέμα, αφού ο αριθμός των παραμέτρων από τις οποίες αποτελούνται μπορεί να είναι τεράστιος. Το γεγονός αυτό οδηγεί στην κατανάλωση αρκετά μεγάλου μέρους του αποθηκευτικού χώρου και έχει αρνητικές συνέπειες στο υπολογιστικό κόστος, λόγω του υπέρογκου αριθμού πράξεων που εκτελούνται κατά τη διαδικασία της συμπερασματολογίας. Επομένως, υπάρχει η ανάγκη για βελτιστοποίηση των μοντέλων βαθιάς μάθησης, ώστε η τοπική εκτέλεση των υπολογισμών να μην είναι απαγορευτική. Για το λόγο αυτό, στην παρούσα διπλωματική εργασία θα μελετήσουμε μεθόδους “συμπίεσης” μοντέλων βαθιάς μάθησης, όπως η κβαντοποίηση, το κλάδεμα και η ομαδοποίηση και θα αξιολογήσουμε πειραματικά την απόδοσή τους στο πρόβλημα της ταξινόμησης εικόνων.

Αρχικά, οι παραπάνω αλγόριθμοι θα μελετηθούν σε θεωρητική βάση και ύστερα θα αξιολογηθούν πειραματικά. Οι αλγόριθμοι θα εφαρμοστούν σε έναν αριθμό από συνελκτικά μοντέλα και θα παρθούν μετρήσεις για το μέγεθος, τον αριθμό πράξεων κινητής υποδιαστολής, το χρόνο απόκρισης και την ακρίβεια που πετυχαίνουν πριν και μετά την εφαρμογή των μεθόδων συμπίεσης. Ο χρόνος απόκρισης θα μετρηθεί κατά την εκτέλεση των μοντέλων σε μια εφαρμογή Android σε κινητή συσκευή. Έπειτα, θα ακολουθήσει σύγκριση των μεθόδων και εξαγωγή συμπερασμάτων.

# 1. Βαθιά Μάθηση

Η τεχνητή νοημοσύνη αποτελεί ερευνητικό κλάδο της πληροφορικής, στον οποίο γίνεται η προσπάθεια για τη δημιουργία “έξυπνων” μηχανών. Ο παραπάνω όρος αναφέρεται στην ιδιότητα της ανάπτυξης κριτικής σκέψης, δηλαδή στην ικανότητα λήψης αποφάσεων για ένα δοσμένο πρόβλημα. Ένας τρόπος με τον οποίον μπορεί αυτό να επιτευχθεί είναι διατυπώνοντας εκ των προτέρων τις αποκρίσεις για διαφορετικές εκδοχές του. Αυτό μπορεί να είναι αρκετά αποτελεσματικό, έστω και προσεγγιστικά, σε προβλήματα για τα οποία ο χώρος των εκδοχών (ενδεχομένων) είναι αρκετά μικρός. Παρ’ όλα αυτά, είναι αρκετά δύσκολο να διατυπωθούν ντετερμινιστικοί κανόνες έτσι ώστε να περιγράψουν τον κόσμο μας με ακρίβεια.

Μια άλλη προσέγγιση είναι η εξαγωγή προτύπων που προέρχονται από δεδομένα. Αυτή η τεχνική ονομάζεται *μηχανική μάθηση* και περιλαμβάνει πληθώρα μεθόδων. Η απόδοση των μοντέλων μηχανικής μάθησης βασίζεται ιδιαίτερα στον τρόπο με τον οποίο γίνεται η *αναπαράσταση των δεδομένων* που έχουμε στη διάθεσή μας. Κάθε είδος πληροφορίας που χρησιμοποιείται για την αναπαράσταση ενός στοιχείου εισόδου, ονομάζεται *χαρακτηριστικό*. Πολλά προβλήματα τεχνητής νοημοσύνης βασίζονται στην εύρεση των χαρακτηριστικών που περιγράφουν καλύτερα ένα στοιχείο εισόδου και στη μετέπειτα εισαγωγή τους σε έναν αλγόριθμο μηχανικής μάθησης. Όμως, υπάρχουν περιπτώσεις στις οποίες είναι αρκετά δύσκολη η εύρεση τέτοιων χαρακτηριστικών, αφού αυτά αποτελούν συνήθως αφηρημένες έννοιες, που δεν μπορούν να παρατηρηθούν στο φυσικό μας κόσμο.

Μια λύση σε αυτό το πρόβλημα είναι η χρήση αλγορίθμων μηχανικής μάθησης, όχι μόνο για την τελική εύρεση του αποτελέσματος, αλλά και για τον προσδιορισμό της κατάλληλης αναπαράστασης και άρα των χαρακτηριστικών μιας εισόδου. Η διαδικασία αυτή ονομάζεται *εκμάθηση αναπαραστάσεων* (representation learning). Εδώ έρχεται ο τομέας της *βαθιάς μάθησης*, που λύνει το συγκεκριμένο πρόβλημα. Η βαθιά μάθηση είναι ένα επιστημονικό πεδίο που περιλαμβάνει ένα υποσύνολο μοντέλων της μηχανικής μάθησης, τα οποία “μαθαίνουν” την κατάλληλη αναπαράσταση της εισόδου ενός προβλήματος με αυτοματοποιημένο τρόπο. Επιτυγχάνει μεγάλη ισχύ και ευελιξία, μαθαίνοντας να αναπαραστή τον κόσμο ως μια *εμφωλευμένη ιεραρχία εννοιών* και αναπαραστάσεων, με κάθε έννοια να ορίζεται σε σχέση με απλούστερες έννοιες και πιο αφηρημένες αναπαραστάσεις να υπολογίζονται σε σχέση με λιγότερο αφηρημένες [1].

## 1.1 Τεχνητά Νευρωνικά Δίκτυα

Ένα αρκετά αντιπροσωπευτικό παράδειγμα της παραπάνω συλλογιστικής, για την αναπαράσταση των εννοιών, αποτελούν τα *τεχνητά νευρωνικά δίκτυα*. Τα τεχνητά νευρωνικά δίκτυα είναι υπολογιστικά μοντέλα, τα οποία είναι εμπνευσμένα από τον τρόπο με τον οποίον έχει παρατηρηθεί και θεωρηθεί ότι γίνεται η εκμάθηση στον εγκέφαλο. Αποτελούνται από νευρώνες και τεχνητές συνάψεις και οργανώνονται σε επίπεδα, τα οποία συμβολίζουν την ιεραρχική εκμάθηση αναπαραστάσεων που περιγράφηκε παραπάνω.

Ο νευρώνας συμβολίζει μια συνάρτηση  $F(x_1, x_2, \dots, x_n) = y$ , η οποία συνδυάζει και μετασχηματίζει δοσμένες τιμές  $x_i$  σε μία τιμή εξόδου  $y$ . Ανάλογα με τη μορφή της συνάρτησης  $F$  μπορούμε να έχουμε νευρώνες διαφορετικού τύπου [1].

### 1.1.1 Νευρώνας McCulloch - Pitts

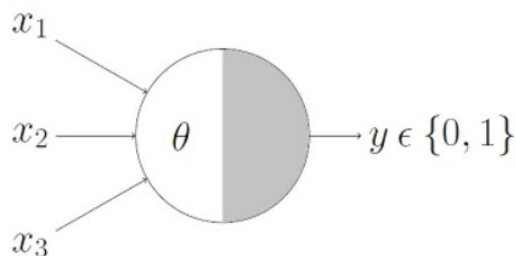
Το πρώτο υπολογιστικό μοντέλο ενός νευρώνα προτάθηκε από τους Warren McCulloch και Walter Pitts το 1943. Αποτελεί μια συνάρτηση  $F$ , η οποία αθροίζει  $n$  δυαδικές τιμές εισόδου  $(x_1, x_2, \dots, x_n) \in \{0,1\}^n$  και έχει ως έξοδο μια επίσης δυαδική τιμή  $y \in \{0,1\}$ .

$$F : \{0,1\}^n \rightarrow \{0,1\}$$

Η τιμή της εξόδου εξαρτάται από την τιμή ενός κατωφλίου  $\theta$ . Επομένως έχουμε την εξής συνάρτηση:

$$F(x) = F(x_1, x_2, \dots, x_n) = \begin{cases} 1, & x_1 + x_2 + \dots + x_n \geq \theta \\ 0, & x_1 + x_2 + \dots + x_n < \theta \end{cases} \quad (1)$$

Ένας νευρώνας McCulloch – Pitts με τρεις εισόδους φαίνεται στο Σχήμα 1.



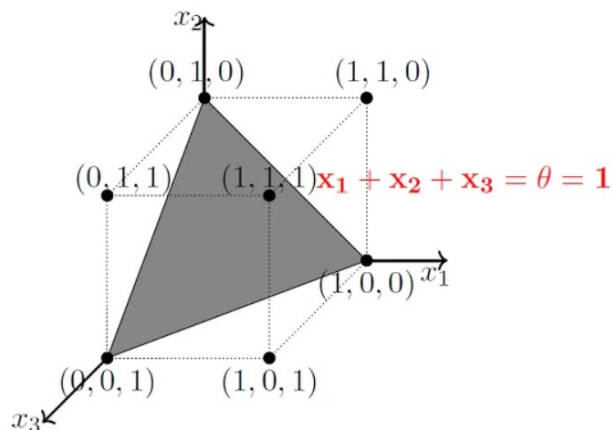
Σχήμα 1: Νευρώνας McCulloch – Pitts [21]

Ένα τέτοιο μοντέλο μπορεί να χρησιμοποιηθεί για την επίλυση προβλημάτων λήψης αποφάσεων. Επίσης, μέσω αυτού, μπορούν να μοντελοποιηθούν λογικές συναρτήσεις, όπως οι OR, AND, NOT, NOR και NAND. Για παράδειγμα, για  $\theta = 1$  έχουμε τη συνάρτηση OR.

Παρατηρούμε ότι η εξίσωση  $x_1 + x_2 + \dots + x_n = \theta$  ορίζει ένα υπερεπίπεδο. Μπορούμε, λοιπόν, να περιγράψουμε το χώρο των εισόδων, στα προβλήματα που επιλύει ο παραπάνω νευρώνας, ως εξής: “Όλες οι εισόδοι που παράγουν 1 βρίσκονται στη μία πλευρά του υπερεπιπέδου και όλες οι εισόδοι που παράγουν 0 βρίσκονται στην άλλη πλευρά του υπερεπιπέδου.” Το υπερεπίπεδο αυτό αποτελεί έναν διαχωριστή. Τα προβλήματα στα οποία ο χώρος των δεδομένων μπορεί να διαχωριστεί από ένα

υπερεπίπεδο σαν το παραπάνω ονομάζονται *γραμμικά διαχωρίσιμα*. Για συναρτήσεις με ακριβώς δύο εισόδους, ο διαχωριστής είναι μια ευθεία [21].

Μπορούμε να το δούμε σχηματικά, για τη συνάρτηση OR με τρεις εισόδους στο Σχήμα 2:



Σχήμα 2: Επίπεδο διαχωρισμού για τη συνάρτηση OR [21]

Αφού η έξοδος του νευρώνα εξαρτάται από τη θέση της αναπαράστασης ενός στοιχείου εισόδου ως προς το υπερεπίπεδο διαχωρισμού, τότε μπορεί κανείς να πει πως ένας νευρώνας με τις παραπάνω ιδιότητες, επιλύει ένα πρόβλημα *κατηγοριοποίησης* της εισόδου σε δύο ομάδες –κλάσεις– οι οποίες αντιστοιχούν στις τιμές εξόδου 0 και 1.

### 1.1.2 Perceptron

Το μοντέλο *Perceptron* δημιουργήθηκε από τους Minsky και Papert τη δεκαετία του 1950 και αποτελεί επέκταση του νευρώνα McCulloch – Pitts. Ξεπερνά ορισμένους από τους περιορισμούς του προηγούμενου νευρώνα, εισάγοντας την έννοια των *αριθμητικών βαρών* – ενός μέτρου σημαντικότητας για τις εισόδους. Τα βάρη στο μοντέλο Perceptron προσαρμόζονται με βάση τα δεδομένα, επομένως απαιτείται και ένας μηχανισμός εκμάθησης αυτών. Επίσης, οι εισοδοί είναι πραγματικοί αριθμοί, γεγονός που καθιστά το Perceptron πιο χρήσιμο και γενικευμένο.

Επομένως, η συνάρτηση  $F : \mathbb{R}^n \rightarrow \{0,1\}$  η οποία περιγράφει το μοντέλο Perceptron είναι η εξής:

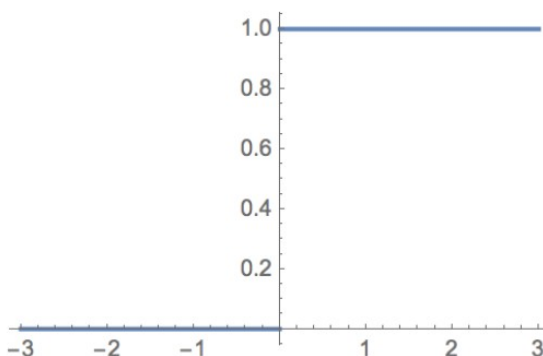
$$F(x) = F(x_1, x_2, \dots, x_n) = \begin{cases} 1, & w_1 x_1 + w_2 x_2 + \dots + w_n x_n \geq \theta \\ 0, & w_1 x_1 + w_2 x_2 + \dots + w_n x_n < \theta \end{cases}$$

όπου  $x_1, \dots, x_n \in \mathbb{R}$  οι τιμές του διανύσματος εισόδου,  $\theta$  το κατώφλι και  $w_1, \dots, w_n \in \mathbb{R}$  τα αντίστοιχα βάρη.

Αν θέσουμε  $b = -\theta$ , τότε θα έχουμε:

$$F(x_1, x_2, \dots, x_n) = \begin{cases} 1, & w^T \cdot x + b \geq 0 \\ 0, & w^T \cdot x + b < 0 \end{cases}$$

όπου  $w^T \cdot x$  είναι το εσωτερικό γινόμενο των βαρών με τις αντίστοιχες εισόδους. Ο όρος  $b$  ονομάζεται *μεροληψία*, αφού αποτελεί ένα υποκειμενικό όριο διαχωρισμού των δεδομένων. Η συνάρτηση την οποία περιγράφει το μοντέλο Perceptron ονομάζεται βηματική και μπορούμε να την αναπαραστήσουμε σχηματικά, όπως στο Σχήμα 3.



Σχήμα 3: Βηματική Συνάρτηση - Step Function [13]

Αν θέσουμε  $z = w^T x + b$  η παραπάνω συνάρτηση μπορεί να γραφεί ως εξής:

$$F(z) = \text{step}(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

Σε αντιστοιχία με τη λειτουργία του εγκεφάλου, λέμε ότι ένας νευρώνας "ενεργοποιείται" όταν για την έξοδο  $y'$  της συνάρτησης  $F$  ισχύει  $y' = 1$ . Για αυτό το λόγο, η βηματική συνάρτηση που εφαρμόζεται ονομάζεται και *συνάρτηση ενεργοποίησης* του νευρώνα.

Μπορεί να παρατηρήσει κανείς ότι η εξίσωση  $w^T \cdot x + b = 0$  αποτελεί επίσης εξίσωση επιπέδου. Επομένως, το Perceptron επιλύει μόνο γραμμικά διαχωρίσιμα προβλήματα. Τα βάρη και η μεροληψία είναι παράμετροι που εξαρτώνται από τα δεδομένα, επομένως οι τιμές τους μπορούν να καθοριστούν από αυτά, μέσω της παρακάτω διαδικασίας εκμάθησης [22].

### Εκμάθηση Βαρών για το Μοντέλο Perceptron

Για την περιγραφή του αλγορίθμου εκμάθησης των βαρών για το μοντέλο Perceptron, η συνάρτηση  $F$  θα έρθει στη μορφή:  $F(x_1, \dots, x_n, 1) = F(x') = \begin{cases} 1, & w'^T \cdot x' \geq 0 \\ 0, & w'^T \cdot x' < 0 \end{cases}$ , όπου  $w' = (w_1, w_2, \dots, w_n, b)$

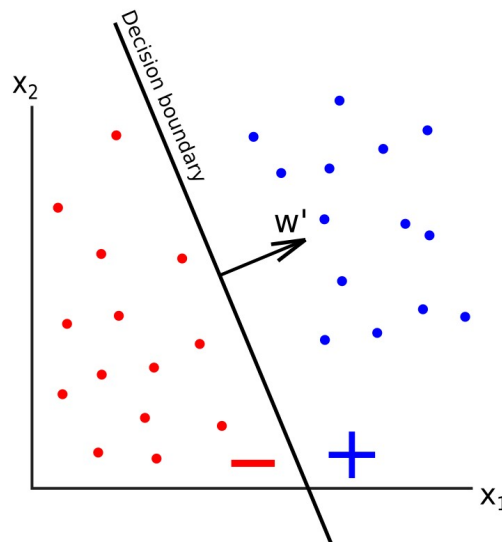
και το  $b$  ως όρος στο εσωτερικό γινόμενο, πολλαπλασιάζεται με τον αριθμό 1.

Ο αλγόριθμος εκμάθησης των βαρών, για ένα πρόβλημα κατηγοριοποίησης με 2 πιθανές εξόδους, αποτελεί μια επαναληπτική διαδικασία, η οποία έχει ως σκοπό να βρεθεί ένα υπερεπίπεδο, το οποίο διαχωρίζει τις δύο ομάδες εισόδων και περιγράφεται από την εξίσωση  $w'^T \cdot x' = 0$ . Ισχύει ότι το διάνυσμα των βαρών  $w'$  είναι κάθετο στο εν λόγω υπερεπίπεδο και αυτό φαίνεται από την τελευταία εξίσωση. Επομένως, αν έχει βρεθεί ένας κατάλληλος διαχωριστής για το παραπάνω πρόβλημα, η εσωτερική γωνία του διανύσματος των βαρών, με το διάνυσμα κάθε στοιχείου από την μία ομάδα θα είναι μικρότερη από  $90^\circ$ , ενώ για τα στοιχεία που ανήκουν στην άλλη ομάδα, η γωνία αυτή θα είναι μεγαλύτερη από  $90^\circ$  (και μικρότερη από  $180^\circ$  ή ίση με  $180^\circ$ ). Η προηγούμενη πρόταση ισχύει, λόγω της καθετότητας του διανύσματος  $w'$  στο επίπεδο.

Από τον ορισμό του εσωτερικού γινομένου δύο τυχαίων διανυσμάτων  $v_1, v_2$  :  $v_1 \cdot v_2 = \|v_1\| \|v_2\| \cos(\widehat{v_1, v_2})$ , παρατηρούμε ότι η τιμή του εξαρτάται από το συνημίτονο της μεταξύ τους γωνίας. Έτσι, σύμφωνα με τα παραπάνω έχουμε:

- $w'^T \cdot x' \geq 0$ , για την πρώτη ομάδα δεδομένων
- $w'^T \cdot x' < 0$ , για τη δεύτερη ομάδα δεδομένων [23]

Τα παραπάνω φαίνονται ως εξής σε δύο διαστάσεις, στο Σχήμα 4:



Σχήμα 4: Ευθεία Διαχωρισμού Δύο Κλάσεων [24]

Επομένως, για την εκμάθηση των βαρών έχουμε την εξής διαδικασία:

- Για κάθε στοιχείο  $x$  του συνόλου δεδομένων:
  - Αν το στοιχείο  $x'$  ανήκει στην πρώτη ομάδα και ισχύει  $w'^T \cdot x' < 0$ , τότε μεταβάλλουμε το  $w$  ως εξής:  $w'_{new} \leftarrow w' + x'$ .



- Αν το στοιχείο  $x$  ανήκει στη δεύτερη ομάδα και ισχύει  $w'^T \cdot x' \geq 0$ , τότε μεταβάλλουμε το  $w$  ως εξής:  $w'_{new} \leftarrow w' - x'$ .

Η διαδικασία συνεχίζεται μέχρι να βρεθεί το κατάλληλο υπερεπίπεδο για τον διαχωρισμό των δεδομένων εισόδου. Συνήθως αυτό επιτυγχάνεται όταν ικανοποιείται ένα κριτήριο τερματισμού.

Διασθητικά, μπορεί κανείς να καταλάβει τη μεταβολή που γίνεται στο διάνυσμα βαρών ως εξής (ισχύει:  $\cos(\widehat{w', x}) \propto (w')^T \cdot x$ : το συνημίτονο της γωνίας μεταξύ του διανύσματος των βαρών  $w'$  και ενός διανύσματος χαρακτηριστικών είναι ανάλογο με το εσωτερικό τους γινόμενο) [23]:

- 1<sup>η</sup> Περίπτωση:  $w'_{new} \leftarrow w' + x'$ ,

$$\cos(\widehat{w'_{new}, x'}) \propto (w'_{new})^T \cdot x' = (w' + x')^T x' = w'^T \cdot x' + x'^T \cdot x' = w'^T \cdot x' + \|x'\|_2^2$$

$$\text{άρα: } \cos(\widehat{w'_{new}, x'}) > \cos(\widehat{w', x'}) \Rightarrow (\widehat{w'_{new}, x'}) < (\widehat{w', x'})$$

- 2<sup>η</sup> Περίπτωση:  $w'_{new} \leftarrow w' - x'$ ,

$$\cos(\widehat{w'_{new}, x'}) \propto (w'_{new})^T \cdot x' = (w' - x')^T x' = w'^T \cdot x' - x'^T \cdot x' = w'^T \cdot x' - \|x'\|_2^2$$

$$\text{άρα: } \cos(\widehat{w'_{new}, x'}) < \cos(\widehat{w', x'}) \Rightarrow (\widehat{w'_{new}, x'}) > (\widehat{w', x'})$$

Ο κανόνας εκμάθησης των βαρών, για το μοντέλο Perceptron, μπορεί να γραφεί και με έναν πιο γενικό τρόπο, ο οποίος συνδέει τη μεταβολή του διανύσματος των βαρών, με το σφάλμα στην τελική πρόβλεψη ως εξής:

$$w'_{new} \leftarrow w' + (y - y')x', \text{ με } y' = \text{step}(z)$$

όπου  $y$  είναι η ορθή τιμή που αντιστοιχεί στην είσοδο  $x$ , την οποία προσπαθούμε να προβλέψουμε και  $y'$  είναι η πρόβλεψη του μοντέλου.

Μπορούμε να έχουμε τις προηγούμενες μεταβολές για τις εξής περιπτώσεις:

1. Η πρόβλεψη του μοντέλου είναι σωστή:  $y = y' \Rightarrow y - y' = 0$ , άρα δεν υπάρχει κάποια μεταβολή στα βάρη.
2. Η πρόβλεψη του μοντέλου είναι λανθασμένη, επομένως, υπάρχουν οι εξής υποπεριπτώσεις:
  - $y - y' = 1$ , άρα έχουμε τη μεταβολή  $w'_{new} \leftarrow w' + x'$
  - $y - y' = -1$ , άρα έχουμε τη μεταβολή  $w'_{new} \leftarrow w' - x'$

### 1.1.3 ADALINE

Το μοντέλο *ADALINE* (Adaptive Linear Neuron) δημιουργήθηκε τη δεκαετία του 1960 και έχει αρκετές ομοιότητες με το Perceptron. Η διαφορά μεταξύ των δύο μοντέλων βρίσκεται κυρίως στη διαδικασία εκμάθησης των βαρών. Στο Perceptron, για να μεταβάλλουμε την τιμή του διανύσματος των βαρών, ελέγχουμε, για κάθε στοιχείο  $x$ , αν ικανοποιείται ένα ορισμένο κριτήριο από το διάνυσμα βαρών  $w'$ , το οποίο καθορίζει αν η ταξινόμηση γίνεται ορθά ή λανθασμένα. Στο *ADALINE* ελέγχεται ένα μέτρο της λανθασμένης ταξινόμησης. Δηλαδή, ακόμα και αν ένα στοιχείο  $x$  ταξινομείται σωστά, η μεταβολή του διανύσματος των βαρών  $w'$  δεν θα είναι μηδενική. Αυτό γίνεται χρησιμοποιώντας τη συνεχή γραμμική συνάρτηση ως συνάρτηση ενεργοποίησης:

$$f(z) = z \quad , \quad \text{με} \quad z = w^T \cdot x + b$$

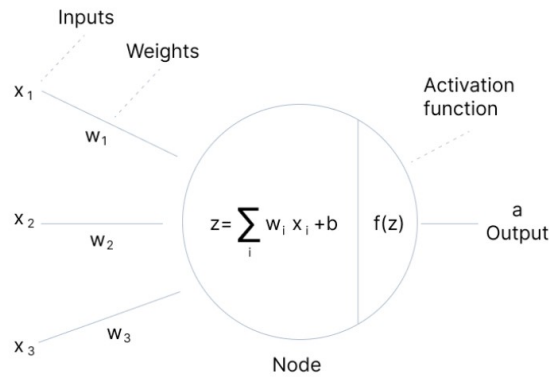
Για την εκπαίδευση των βαρών για το μοντέλο *ADALINE*, για κάθε στοιχείο δεδομένων  $x$ , υπολογίζουμε τη μεταβολή κάθε συνιστώσας του διανύσματος βαρών όπως και στο μοντέλο Perceptron, με μόνη διαφορά πως εισάγουμε επιπλέον τον όρο του ρυθμού εκμάθησης  $\eta$  ως έναν θετικό και αρκετά μικρό αριθμό ως εξής:

$$w_{new} \leftarrow w + \eta(y - y')x \quad , \quad \text{με} \quad y' = z \quad \text{και} \quad \eta > 0$$

Μετά από την εκπαίδευση των βαρών, στα δεδομένα εισόδου εφαρμόζεται η ίδια συνάρτηση που έχουμε δει και στις προηγούμενες περιπτώσεις νευρώνων [25].

Επομένως, μπορεί να δει κανείς πως η διαφορά μεταξύ των δύο παραπάνω μοντέλων έγκειται στο είδος της συνάρτησης ενεργοποίησης. Συγκεκριμένα, στην ιδιότητα της συνέχειας της συνάρτησης αυτής, η οποία επηρεάζει τη συνέχεια της συνάρτησης της διαφοράς της πρόβλεψης του μοντέλου με την ορθή τιμή που θέλουμε να προβλέψουμε:  $y - y'$ . Στην περίπτωση του Perceptron έχουμε τη βηματική συνάρτηση, η οποία είναι διακριτή. Επομένως, λόγω αυτής της ιδιότητας γίνεται μεταβολή των βαρών μόνο όταν η τιμή  $y'$  είναι λανθασμένη, δηλαδή δεν ταυτίζεται με την  $y$ . Αντίθετα, στο μοντέλο *ADALINE*, η συνάρτηση ενεργοποίησης είναι η ταυτοτική γραμμική, η οποία είναι συνεχής. Η ιδιότητα αυτή έχει ως αποτέλεσμα η συνάρτηση  $y - y'$  να είναι επίσης συνεχής και έτσι τα βάρη να μεταβάλλονται ακόμη και όταν το αποτέλεσμα της τελικής συνάρτησης του νευρώνα  $y'$  είναι ορθό. Το γεγονός αυτό, μας δίνει ένα μέτρο ως προς το πόσο καλά προσεγγίζεται η τελική τιμή  $y$  [2].

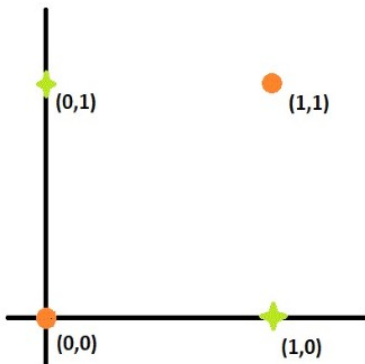
Τέλος, μπορούμε να αναπαραστήσουμε ένα νευρώνα, στη γενική του μορφή, όπως φαίνεται στο Σχήμα 5.



Σχήμα 5: Γενική Μορφή Νευρώνα [8]

### 1.1.4 Perceptron Πολλαπλών Επιπέδων

Τα προηγούμενα μοντέλα έχουν σοβαρούς περιορισμούς. Ένας από αυτούς είναι πως δεν μπορούν να επιλύσουν μη γραμμικά διαχωρίσιμα προβλήματα και έτσι δεν υπάρχει συνέπεια στην περιγραφή του φυσικού κόσμου. Το πιο απλό παράδειγμα αποτελεί η συνάρτηση XOR, για την οποία δεν υπάρχει επίπεδο το οποίο να διαχωρίζει τις δύο κλάσεις δεδομένων. Αυτό φαίνεται και από το Σχήμα 6:



Σχήμα 6: Κατανομή των δεδομένων σύμφωνα με τη συνάρτηση XOR [26]

Αν υπήρχε υπερεπίπεδο:  $w_1 x_1 + w_2 x_2 + b = 0$ , το οποίο να διαχωρίζει τα δεδομένα εισόδου θα έπρεπε να ικανοποιεί τις εξής σχέσεις:

$$x_1=0 \wedge x_2=0 : b < 0 \quad (\text{i})$$

$$x_1=1 \wedge x_2=0 : w_1 + b \geq 0 \quad (\text{ii})$$

$$x_1=0 \wedge x_2=1 : w_2 + b \geq 0 \quad (\text{iii})$$

$$x_1=1 \wedge x_2=1 : w_1 + w_2 + b < 0 \quad (\text{iv})$$

Παρατηρούμε πως αν αθροίσουμε τις ανισώσεις (ii) και (iii), έχουμε  $w_1 + w_2 + b \geq -b$ , για το οποίο, λόγω της σχέσης (i):  $b < 0$  ισχύει  $w_1 + w_2 + b > 0$  και έρχεται σε αντίθεση με τη σχέση (iv) [22].

Τη δεκαετία του 1980 με την ανάπτυξη της παράλληλης κατανεμημένης επεξεργασίας και σαν αποτέλεσμα της διαθεσιμότητας περισσότερων υπολογιστικών πόρων, κυριάρχησε η λογική πως “ένας

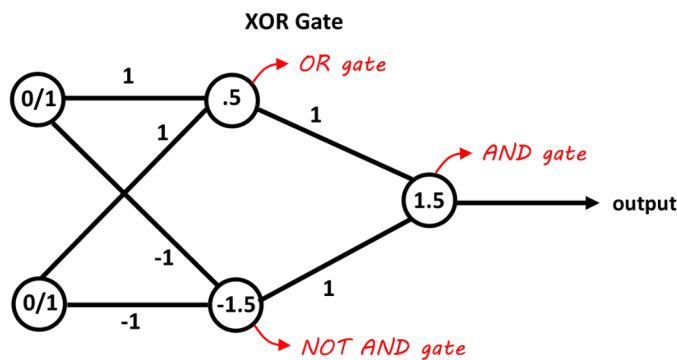
μεγάλος αριθμός από υπολογιστικές μονάδες -νευρώνες- μπορεί να επιτύχει ευφυή συμπεριφορά με τη μεταξύ τους σύνδεση”. Αυτό οδήγησε στην εφαρμογή της ιδέας πως κάθε είσοδος σε ένα σύστημα πρέπει να αναπαρίσταται από πολλά χαρακτηριστικά και κάθε χαρακτηριστικό θα πρέπει να εμπλέκεται στην αναπαράσταση πολλών διαφορετικών στοιχείων εισόδου. Η ιδέα αυτή ονομάζεται *κατανεμημένη αναπαράσταση* και χρησιμοποιείται μέχρι και σήμερα.

Η κύρια εφαρμογή της παραπάνω ιδέας είναι τα *τεχνητά νευρωνικά δίκτυα* ή *Perceptron* πολλαπλών επιπέδων, τα οποία αποτελούν παραμετρικές συναρτήσεις, που ορίζονται από τον συνδυασμό άλλων παραμετρικών συναρτήσεων. Κάθε μία από τις επιμέρους συναρτήσεις έχει πολλαπλές εισόδους και εξόδους. Σύμφωνα με την ορολογία που χρησιμοποιείται στη θεωρία των τεχνητών νευρωνικών δικτύων, κάθε τέτοια παραμετρική συνάρτηση ονομάζεται *στρώμα* του μοντέλου και κάθε βαθμωτό μέγεθος της εξόδου ενός στρώματος ονομάζεται *μονάδα* ή *χαρακτηριστικό*. Τα στρώματα ενός μοντέλου που αντιστοιχούν σε χαρακτηριστικά και όχι στις τελικές προβλέψεις ονομάζονται *κρυφά στρώματα*. Παρά το γεγονός η σχέση που περιγράφει την εξάρτηση μίας μονάδας από την είσοδό της είναι μία απλή συνάρτηση, η συνάρτηση η οποία συνδέει την έξοδο του μοντέλου με τα δεδομένα εισόδου, τις περισσότερες φορές, είναι ιδιαίτερα περίπλοκη. [1] Με τον τρόπο αυτό ένα μοντέλο βαθιάς μάθησης μπορεί να μαθαίνει την κατάλληλη αναπαράσταση των χαρακτηριστικών, μέσω των μετασχηματισμών που γίνονται στα κρυφά στρώματα.

Σαν αποτέλεσμα, τα μοντέλα τεχνητών νευρωνικών δικτύων μπορούν να χρησιμοποιηθούν για την επίλυση μη γραμμικά διαχωρίσιμων προβλημάτων, όπως το πρόβλημα προσέγγισης της συνάρτησης XOR η οποία δεν μπορούσε να περιγραφεί από το μοντέλο Perceptron. Ένας τρόπος προσέγγισης είναι η διαίρεση του προβλήματος αυτού σε μικρότερα υποπροβλήματα, τα οποία είναι γραμμικά διαχωρίσιμα. Στην περίπτωση αυτή αυτό μπορεί να γίνει ως εξής:

$$\text{XOR}(x_1, x_2) = (x_1 \vee x_2) \wedge \neg(x_1 \wedge x_2)$$

Επομένως, ένα μοντέλο το οποίο λύνει το παραπάνω πρόβλημα και μπορεί να ερμηνευθεί από την παραπάνω γραφή της συνάρτησης XOR, φαίνεται στο Σχήμα 7 [26].



Σχήμα 7: Μοντέλο Τεχνητών Νευρωνικών Δικτύων που προσεγγίζει τη συνάρτηση XOR [62]

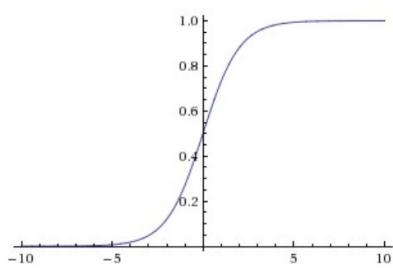
Στο μοντέλο του παραπάνω σχήματος, οι αριθμοί που φαίνονται στους νευρώνες του κρυφού στρώματος και στον νευρώνα εξόδου αποτελούν τα κατώφλια για καθέναν από αυτούς και η συνάρτηση ενεργοποίησης είναι η βηματική. Μία ερμηνεία του παραπάνω μοντέλου ως επίλυση του προβλήματος της συνάρτησης XOR είναι η εξής:

1. Έχουμε ως είσοδο τα  $x_1, x_2$ .
2. Στον πρώτο νευρώνα του πρώτου κρυφού στρώματος εκτελείται η πράξη OR  $(x_1 \vee x_2)$ .
3. Στον δεύτερο νευρώνα, του πρώτου κρυφού στρώματος εκτελείται η πράξη NAND  $\neg(x_1 \wedge x_2)$ .
4. Στον νευρώνα του στρώματος εξόδου εκτελείται η πράξη AND  $(x'_1 \wedge x'_2)$ .

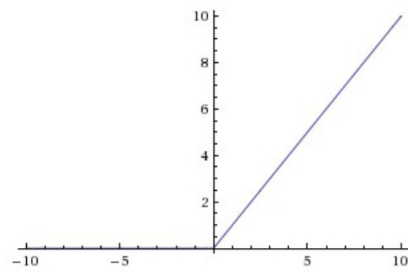
Όπως αναφέρεται παραπάνω, οι συναρτήσεις ενεργοποίησης αποτελούν ένα επιπλέον βήμα στους υπολογισμούς ενός νευρώνα και προσθέτουν μη γραμμική εξάρτηση του αποτελέσματος από την είσοδο. Η μη γραμμικότητα συνδέεται άμεσα με την εξαγωγή χαρακτηριστικών και έμμεσων συνδέσεων από τα δεδομένα, αφού αν δεν υπήρχε, κάθε νευρώνας θα εκτελούσε μόνο έναν γραμμικό μετασχηματισμό στις εισόδους. Σαν αποτέλεσμα, η έξοδος του μοντέλου θα ήταν γραμμική συνάρτηση της εισόδου του, αφού η σύνθεση δύο γραμμικών συναρτήσεων είναι και η ίδια μια γραμμική συνάρτηση. Έτσι, το μοντέλο θα εκφυλιζόταν σε επίπεδο νευρώνα [9]. Κάποιες από τις συναρτήσεις ενεργοποίησης που χρησιμοποιούνται συνήθως είναι οι εξής:

1. Σιγμοειδής Συνάρτηση:  $\sigma(x) = \frac{1}{1+e^{-x}}$
2. Υπερβολική Εφαπτομένη:  $\tanh(x) = 2\sigma(2x) - 1$
3. Ανορθωμένη Γραμμική Μονάδα (Rectified Linear Unit – ReLU):  $f(x) = \max(0, x)$
4. Softmax:  $f_i(x) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$ , χρησιμοποιείται ως μέτρο πιθανότητας

Κάποιες από τις προαναφερθείσες συναρτήσεις φαίνονται παρακάτω.

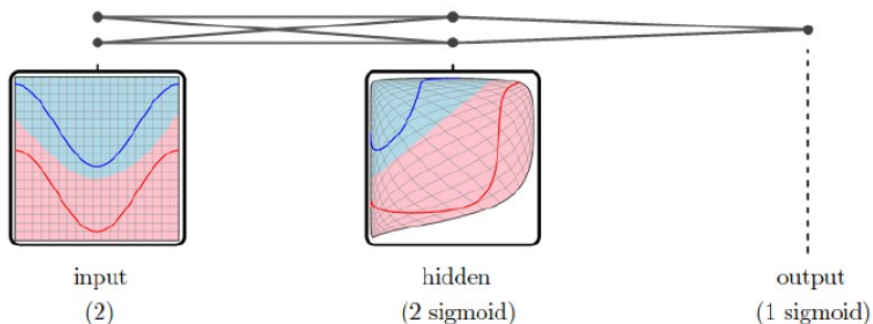


Σχήμα 8: Σιγμοειδής [10]



Σχήμα 9: ReLU [10]

Μπορούμε να δούμε μια γραφική αναπαράσταση ενός μετασχηματισμού του χώρου δεδομένων ενός προβλήματος, από ένα μοντέλο τεχνητών νευρωνικών δικτύων, στην παρακάτω εικόνα, όπου μέσω των μετασχηματισμών που γίνονται στα κρυφά στρώματα το σύνολο δεδομένων μετατρέπεται σε γραμμικά διαχωρίσιμο. Η συνάρτηση ενεργοποίησης των νευρώνων στην περίπτωση αυτή είναι η σιγμοειδής.



Σχήμα 10: Μετασχηματισμός Δεδομένων Εισόδου σε Γραμμικά Διαχωρίσιμα [1]

## 1.2 Εκπαίδευση και Εκμάθηση Βαρών

Όπως είχαμε δει κι από τα μοντέλα τα οποία αποτελούνται από έναν νευρώνα, όπως το Perceptron ή το ADALINE, σε ένα μοντέλο μπορεί να γίνει ρύθιση των βαρών χρησιμοποιώντας δεδομένα. Επομένως, για ένα πρόβλημα και ένα μέτρο απόδοσης ενός μοντέλου ως προς αυτό, η *εκπαίδευση* μπορεί να οριστεί ως η διαδικασία εύρεσης των παραμέτρων -βαρών- ενός μοντέλου, οι οποίες βελτιστοποιούν το μέτρο απόδοσής του στο δοσμένο πρόβλημα. Δηλαδή, αποτελεί, όπως περιγράφηκε και παραπάνω, μια διαδικασία *εκμάθησης* των βαρών. Η εκπαίδευση ενός μοντέλου μπορεί να αναχθεί σε ένα πρόβλημα βελτιστοποίησης.

### 1.2.1 Επιβλεπόμενη Μάθηση

Η *επιβλεπόμενη μάθηση* είναι μια υποκατηγορία της μηχανικής μάθησης και ορίζεται από τη χρήση επισημειωμένων συνόλων δεδομένων, για την εκπαίδευση αλγορίθμων που ταξινομούν δεδομένα ή προβλέπουν αποτελέσματα [27].

Η μεταβολή των βαρών κατά την εκπαίδευση γίνεται χρησιμοποιώντας ένα σύνολο δεδομένων  $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ , όπου κάθε  $x_i$  αποτελεί ένα διάνυσμα χαρακτηριστικών  $m$  διαστάσεων και

κάθε  $y_i$  είναι η αντίστοιχη έξοδος που θέλουμε, η οποία καλείται και *ετικέτα*. Επίσης, γίνεται η υπόθεση ότι τα ζεύγη εισόδου  $(x_i, y_i)$  ακολουθούν μια κατανομή  $D$ ,  $(x_i, y_i) \sim D$  για κάθε  $i = 1, \dots, n$ .

Μπορούμε να συνοψίσουμε τον στόχο της επιβλεπόμενης μάθησης, ως την εύρεση μιας συνάρτησης  $h : \mathbb{R}^m \rightarrow \mathbb{C}$ , τέτοιας ώστε για κάθε νέο ζεύγος εισόδου/εξόδου  $(x_i, y_i)$  που λαμβάνεται δειγματοληπτικά από την  $D$ , να ισχύει  $h(x_i) \approx y_i$  [3].

## Αντιστάθμιση Μεροληψίας και Διασποράς

Τόσο κατά την κατασκευή, όσο και κατά την εκπαίδευση ενός μοντέλου, στόχος είναι αρχικά αυτό να έχει σωστές προβλέψεις για τα δεδομένα με βάση τα οποία έχει γίνει η μεταβολή των βαρών. Από την άλλη πλευρά, δεν θα πρέπει οι προβλέψεις να εξειδικεύονται σε μεγάλο βαθμό στο σύνολο εκπαίδευσης, γιατί αυτό θα αφαιρούσε τη δυνατότητα αυτές να γενικευθούν σε νέα δεδομένα. Η πρόβλεψη ενός μοντέλου αποτελεί στοχαστικό μέγεθος, επομένως μπορεί να περιγραφεί σαν μία τυχαία μεταβλητή  $\hat{y}$ .

Με βάση την ιδιότητα αυτή, μπορούμε με μαθηματικό τρόπο να ορίσουμε τις παρακάτω έννοιες:

- **Μεροληψία:** Ονομάζεται το μέγεθος της διαφοράς μεταξύ της μέσης πρόβλεψης ενός μοντέλου  $E[\hat{y}]$  και της πραγματικής τιμής  $y$ .
- **Διασπορά:** Ονομάζεται η μεταβλητότητα της πρόβλεψης ενός μοντέλου για μία δεδομένη είσοδο και μας δίνει πληροφορίες για την εξάπλωση των δεδομένων.

Ένα παράδειγμα είναι αυτό της παλινδρόμησης, όπου ένα μοντέλο έχει ως στόχο να προσεγγίσει μία άγνωστη, συνεχή συνάρτηση  $f$ . Το συνολικό σφάλμα σε αυτήν την περίπτωση είναι:

$$E[(y - \hat{y})^2] = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

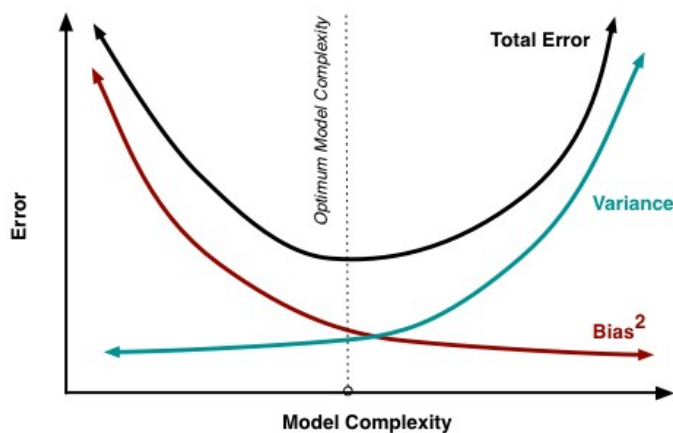
όπου  $n$  είναι το πλήθος των δειγμάτων που έχουμε.

Το παραπάνω άθροισμα μπορεί να γραφεί και ως εξής:

$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (E[\hat{y}] - y_i)^2 + E[(E[\hat{y}] - \hat{y})^2] + \varepsilon$$

Παρατηρούμε πως ο πρώτος όρος αποτελεί το τετράγωνο της μεροληψίας, ο δεύτερος όρος αποτελεί τη διασπορά, ενώ το  $\varepsilon$  είναι το σφάλμα που προέρχεται από το θόρυβο των δεδομένων και είναι πάντα θετικό,  $\varepsilon > 0$ . Το συνολικό σφάλμα εξαρτάται από κάθε μία από τις τρεις αυτές ποσότητες [39].

Μπορούμε να παρατηρήσουμε πώς αλληλεπιδρούν τα παραπάνω μεγέθη ως προς την πολυπλοκότητα του μοντέλου (δηλαδή το πλήθος των παραμέτρων από τις οποίες αποτελείται) στο Σχήμα 11.



Σχήμα 11: Αντιστάθμιση Μεροληψίας και Διασποράς σε σχέση με το Συνολικό Σφάλμα [38]

### Υποπροσαρμογή και Υπερπροσαρμογή

Στη ρίζα της, η ελαχιστοποίηση της μεροληψίας και της διακύμανσης αντίστοιχα, αφορά στην πραγματικότητα, την αντιμετώπιση της υποπροσαρμογής και της υπερπροσαρμογής ενός μοντέλου στα δεδομένα εκπαίδευσης. Καθώς προστίθενται όλο και περισσότερες παράμετροι σε ένα μοντέλο, η πολυπλοκότητα του μοντέλου αυξάνεται και η διακύμανση γίνεται το κύριο μέλημά μας, ενώ η μεροληψία μειώνεται σταθερά [38].

### 1.2.2 Θεωρία Βελτιστοποίησης

Η βελτιστοποίηση αποτελεί κλάδο των μαθηματικών ο οποίος ασχολείται με προβλήματα ελαχιστοποίησης ή μεγιστοποίησης μιας συνάρτησης  $f(x)$ , βρίσκοντας το σημείο  $x$  το οποίο την ελαχιστοποιεί. Συνήθως, ο όρος “βελτιστοποίηση” αναφέρεται σε προβλήματα ελαχιστοποίησης. Τα προβλήματα μεγιστοποίησης μιας συνάρτησης  $f(x)$ , εύκολα μπορούν να αναχθούν στα παραπάνω, αν ελαχιστοποιήσουμε την  $-f(x)$ . Η συνάρτηση που θέλουμε να βελτιστοποιήσουμε ονομάζεται συνάρτηση στόχου ή συνάρτηση κόστους ή συνάρτηση σφάλματος.

Χρήσιμο εργαλείο για τη βελτιστοποίηση μιας συνάρτησης  $f$  είναι η παράγωγός της, αφού δίνει την κλίση της  $f(x)$  στο σημείο  $x$ , για μια πολύ μικρή θετική μεταβολή στη μεταβλητή  $x$ . Με άλλα



λόγια, ορίζει πώς να κλιμακώσουμε μία μικρή αλλαγή στην είσοδο της συνάρτησης, έτσι ώστε να έχουμε την αναμενόμενη αλλαγή στην έξοδο. Αυτό φαίνεται και από τον ορισμό της παραγώγου ως εξής:

$$f(x+\varepsilon) \approx f(x) + \varepsilon f'(x)$$

Τα σημεία για τα οποία δεν μπορούμε να βγάλουμε κάποιο συμπέρασμα σύμφωνα με την παραπάνω σχέση είναι αυτά στα οποία μηδενίζεται η συνάρτηση  $f'(x)$ . Αυτά τα σημεία ανήκουν σε μία από τις παρακάτω κατηγορίες:

- Τοπικά ή ολικά ελάχιστα
- Τοπικά ή ολικά μέγιστα
- Σαγματικά σημεία (σημεία στα οποία μηδενίζεται η παράγωγος, αλλά δεν είναι ελάχιστα)

Τα σύγχρονα προβλήματα βελτιστοποίησης, συνήθως, αφορούν συναρτήσεις πολλών μεταβλητών. Για το λόγο αυτό, χρησιμοποιούμε τις μερικές παραγώγους και ορίζουμε την έννοια της κλίσης (Gradient) μιας συνάρτησης  $f$   $n$  μεταβλητών ως εξής:

$$\nabla f(x_1, \dots, x_n) = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)$$

Σε πολλά προβλήματα βελτιστοποίησης, όπως και στην εκπαίδευση τεχνητών νευρωνικών δικτύων, βελτιστοποιούμε συναρτήσεις που μπορεί να έχουν πολλά τοπικά ελάχιστα, τα οποία δεν είναι βέλτιστα και πολλά σαγματικά σημεία που περιβάλλονται από πολύ επίπεδες περιοχές. Όλα αυτά καθιστούν τη διαδικασία της βελτιστοποίησης πολύ δύσκολη, ειδικά όταν η είσοδος της συνάρτησης είναι πολυδιάστατη. Επομένως, συνήθως συμβιβάζομαστε με την εύρεση μιας τιμής της  $f$  που είναι πολύ χαμηλή, αλλά όχι απαραίτητα ελάχιστη με οποιαδήποτε τυπική έννοια (τοπικό ελάχιστο) [1]. Σαν αποτέλεσμα χρησιμοποιούμε επαναληπτικούς αλγορίθμους με κάποιο κριτήριο τερματισμού, το οποίο οδηγεί σε ένα επιθυμητό αποτέλεσμα.

### 1.2.3 Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων

Η εκπαίδευση τεχνητών νευρωνικών δικτύων επιλύει έμμεσα ένα πρόβλημα βελτιστοποίησης. Μας ενδιαφέρει η μεγιστοποίηση ενός μεγέθους απόδοσης  $\mathcal{P}$ , το οποίο δεν ξέρουμε πώς ακριβώς να επηρεάσουμε με άμεσο τρόπο, επομένως ελαχιστοποιούμε μια συνάρτηση κόστους  $J(\theta)$ , με την προσδοκία ότι, μέσω αυτού, θα βελτιωθεί το  $\mathcal{P}$  [1]. Αυτό γίνεται γιατί το μέτρο απόδοσης δεν είναι σε κάθε περίπτωση μια ομαλή συνάρτηση των βαρών ενός μοντέλου, επομένως για μία ορισμένη είσοδο,

μία μικρή μεταβολή στα βάρη δεν συνεπάγεται απαραίτητα μικρή μεταβολή στην συνάρτηση απόδοσης [2].

### Συναρτήσεις Κόστους

Υπάρχουν πολλοί τρόποι αναπαράστασης του σφάλματος στην εκμάθηση των βαρών ενός μοντέλου σε μια διαδικασία εκπαίδευσης και η επιλογή εξαρτάται από το πρόβλημα προς επίλυση. Κάποιες από αυτές τις συναρτήσεις παρουσιάζονται παρακάτω.

Αρχικά, έχουμε το άθροισμα της διαφοράς, της τιμής που δίνει το μοντέλο από την τιμή της ετικέτας, υψωμένης στο τετράγωνο. Έτσι, προκύπτει το μέσο τετραγωνικό σφάλμα, το οποίο χρησιμοποιείται σε προβλήματα παλινδρόμησης και έχει τον εξής τύπο:

$$L = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

όπου  $n$  είναι το πλήθος των δειγμάτων,  $y$  η παρατήρηση που θέλουμε να πάρουμε από το μοντέλο και  $\hat{y}$  το αποτέλεσμα που δίνει το μοντέλο [4].

Μία άλλη επιλογή συνάρτησης κόστους αποτελεί η εγκάρσια εντροπία, η οποία χρησιμοποιείται σε προβλήματα κατηγοριοποίησης. Η εγκάρσια εντροπία αποτελεί μέτρο της διαφοράς μεταξύ δύο κατανομών πιθανότητας, για μια δεδομένη τυχαία μεταβλητή ή σύνολο γεγονότων. Στα προβλήματα ταξινόμησης, γνωρίζουμε την κατανομή πιθανότητας στόχου  $D$  για μια είσοδο, ως την ετικέτα κλάσης 0 ή 1 που ερμηνεύεται ως πιθανότητες ως "αδύνατο" ή "βέβαιο" αντίστοιχα. Ένα μοντέλο μπορεί να εκτιμήσει την πιθανότητα ένα παράδειγμα να ανήκει σε κάθε μία από τις δοσμένες κλάσεις. Η εγκάρσια εντροπία μπορεί στη συνέχεια να χρησιμοποιηθεί για τον υπολογισμό της διαφοράς μεταξύ των δύο κατανομών πιθανότητας [5]. Η συνάρτηση αυτή, για κάθε είσοδο, περιγράφεται από τον εξής τύπο:

$$CE = - \sum_{i=1}^k p_i \log(q_i)$$

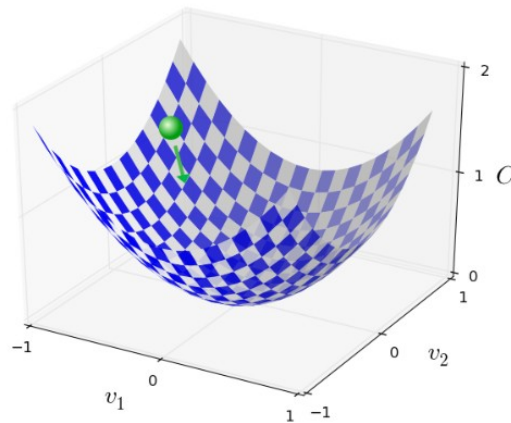
όπου,  $k$  είναι το πλήθος των εξόδων ενός μοντέλου,  $p_i$  η δοσμένη πιθανότητα και  $q_i$  η πιθανότητα που προβλέπεται από το μοντέλο για κάθε κλάση  $i$ .

### Αλγόριθμοι Καθόδου

## Μέθοδος Ομαλής Καθόδου

Η μέθοδος ομαλής καθόδου είναι μια επαναληπτική μέθοδος ελαχιστοποίησης μιας συνάρτησης κόστους  $J(\theta)$ , που εξαρτάται από τις παραμέτρους  $\theta \in \mathbb{R}^d$  του μοντέλου, ενημερώνοντάς τις προς την αντίθετη κατεύθυνση της κλίσης της συνάρτησης κόστους:  $\nabla_{\theta} J(\theta)$ . Ο ρυθμός εκμάθησης  $\eta$  καθορίζει το μέγεθος των βημάτων που κάνουμε για να φτάσουμε σε ένα ελάχιστο. Μπορούμε, πιο διαισθητικά, να πούμε πως ακολουθούμε την κατεύθυνση της κλίσης της επιφάνειας της συνάρτησης κόστους προς τα κάτω μέχρι να φτάσουμε σε μια κοιλάδα (τοπικό ελάχιστο), όπως φαίνεται στο Σχήμα 12 [11].

$$\theta_{new} \leftarrow \theta - \eta \cdot \nabla_{\theta} J(\theta)$$



Σχήμα 12: Μέθοδος Ομαλής Καθόδου - Γραφική Αναπαράσταση Δύο Διαστάσεων [2]

Η διαδικασία της εκπαίδευσης χωρίζεται σε φάσεις, κατά τις οποίες στοιχεία από το σύνολο δεδομένων εισάγονται στο μοντέλο, μέσω των οποίων υπολογίζεται η τιμή της συνάρτησης κόστους και τα βάρη μεταβάλλονται σύμφωνα με τον αλγόριθμο εκπαίδευσης. Μία εποχή αποτελεί το χρονικό διάστημα μέσα στο οποίο το μοντέλο “βλέπει” όλο το σύνολο δεδομένων, κατά τη διάρκεια της εκπαίδευσης. Η διαδικασία της εκπαίδευσης περιλαμβάνει παραπάνω από μία εποχές, αφού αποτελεί μία επαναληπτική διαδικασία ως προς το σύνολο δεδομένων.

Υπάρχουν τρεις παραλλαγές της μεθόδου ομαλής καθόδου, οι οποίες διαφοροποιούνται ως προς το πλήθος των δεδομένων που χρησιμοποιούνται για τον υπολογισμό της κλίσης της συνάρτησης κόστους.

- Ομαδική Ομαλή Κάθοδος: Χρησιμοποιείται όλο το σύνολο εκπαίδευσης για την ενημέρωση των παραμέτρων του μοντέλου.

- Στοχαστική Ομαλή Κάθοδος: Οι παράμετροι του μοντέλου ενημερώνονται με κάθε νέο ζεύγος δεδομένων εισόδου  $(x,y)$ . Μία συνέπεια αυτής της μεθόδου είναι πως η εκτίμηση της συνάρτησης κόστους δεν είναι τόσο καλή και σαν αποτέλεσμα η μεταβολή των βαρών έχει σχετικά μεγάλο σφάλμα.
- Ομαλή Κάθοδος με Δέσμες: Χρησιμοποιείται ένα υποσύνολο του συνόλου εκπαίδευσης – δέσμη– για την ενημέρωση των παραμέτρων του μοντέλου. Με τον τρόπο αυτό βελτιώνεται η προσέγγιση της συνάρτησης σφάλματος και έτσι γίνονται με μεγαλύτερη ακρίβεια οι μεταβολές στις παραμέτρους.

Υπάρχουν πολλοί αλγόριθμοι βελτιστοποίησης οι οποίοι βασίζονται στην παραπάνω μέθοδο. Κάποιοι από αυτούς είναι οι εξής:

- Ορμή (Momentum): Επινοήθηκε για τη βελτίωση της Στοχαστικής Ομαλής Καθόδου και ομαλοποιεί τη σύγκλιση. Αυτό γίνεται μέσω μιας πρόσθετης υπερπαραμέτρου  $\gamma$ , η οποία ονομάζεται *ορμή*. Η μεταβολή των παραμέτρων  $\theta$  περιγράφεται από τις παρακάτω εξισώσεις:

$$v_t \leftarrow \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta)$$

$$\theta_t \leftarrow \theta_{t-1} - v_t$$

Η υπερπαραμέτρος  $\gamma$  αυξάνεται για διαστάσεις των οποίων οι κλίσεις δείχνουν προς τις ίδιες κατευθύνσεις και μειώνει τις ενημερώσεις για διαστάσεις των οποίων οι κλίσεις αλλάζουν κατεύθυνση. Έτσι, οδηγούμαστε πιο γρήγορα σε σύγκλιση. Η τιμή του όρου  $\gamma$  είναι συνήθως κοντά στο 0.9 .

- Adagrad: Ένας αλγόριθμος ο οποίος μεταβάλλει δυναμικά τον ρυθμό μάθησης, χρησιμοποιώντας για το σκοπό αυτό την τιμή της πρώτης παραγώγου ως προς κάθε παράμετρο.
- RMSprop: Ο αλγόριθμος RMSprop επινοήθηκε ως βελτίωση του Adagrad, λόγω της πολύ γρήγορης ελάττωσης του ρυθμού εκμάθησης. Για το λόγο αυτό χρησιμοποιεί τη ροπή δεύτερης τάξης της εκτίμησης της πρώτης παραγώγου για κάθε παράμετρο.
- Adam: Η μέθοδος αυτή περιέχει στοιχεία από όλες τις προηγούμενες μεθόδους. Υπολογίζει δυναμικά τους ρυθμούς μάθησης για κάθε παράμετρο ξεχωριστά, από τις εκτιμήσεις της πρώτης και της δεύτερης ροπής των κλίσεων της συνάρτησης κόστους, οι οποίες φθίνουν εκθετικά ανά βήμα. Ο αλγόριθμος ενημερώνει τους μέσους όρους της κλίσης  $m_t$  και της τετραγωνικής κλίσης  $v_t$  , και οι υπερπαραμέτροι  $\beta_1, \beta_2 \in [0, 1)$  ελέγχουν τους εκθετικούς ρυθμούς αποσύνθεσης αυτών των κινητών μέσων όρων. Οι ροπές πρώτης και δεύτερης τάξης της κλίσης υπολογίζονται ως εξής:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

όπου  $g_t$  είναι η κλίση (gradient), της συνάρτησης κόστους ως προς τα βάρη, όπως υπολογίζεται σε κάθε βήμα  $t$ .

Έπειτα, κανονικοποιούνται ως εξής:  $\hat{m}_t = \frac{m_t}{1-\beta_1}$  ,  $\hat{v}_t = \frac{v_t}{1-\beta_2}$  και οι παράμετροι μεταβάλλονται τελικά ως εξής:

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

όπου για την σταθερα  $\epsilon$  προτείνεται η τιμή  $10^{-8}$  [11].

## Διάδοση Σφάλματος

Για να υπολογιστεί η μεταβολή για κάθε βάρος ενός μοντέλου στη μέθοδο ομαλής καθόδου, κατά την εκπαίδευση, θα πρέπει να υπολογιστεί η κατεύθυνση για κάθε συνιστώσα – βάρος, προς την οποία μειώνεται η τιμή της συνάρτησης κόστους  $J$ . Αυτό επιτυγχάνεται με τον υπολογισμό της μερικής παραγώγου  $\frac{\partial J}{\partial w}$ , όπως έχει ήδη σημειωθεί, για κάθε βάρος. Με τον αλγόριθμο οπισθοδιάδοσης (backpropagation), μπορεί κανείς να υπολογίσει την παραπάνω ποσότητα για κάθε νευρώνα, ξεκινώντας από το τελευταίο στρώμα του μοντέλου και έπειτα εφαρμόζοντας αναδρομικά τον κανόνα της αλυσίδας.

Συμβολίζουμε με:

- $w_{jk}^l$  : το βάρος που πολλαπλασιάζεται με την  $k$  ενεργοποίηση του στρώματος  $l-1$  και αποτελεί παράμετρο της  $j$  ενεργοποίησης του στρώματος  $l$
- $z_j^l$  : την  $j$  ενεργοποίηση του στρώματος  $l$
- $a_k^{l-1}$  : η  $k$  ενεργοποίηση του στρώματος  $l-1$
- $b_j^l$  : η  $j$  μεροληψία του στρώματος  $l$

Σύμφωνα με τον κανόνα της αλυσίδας, η μερική παράγωγος της συνάρτησης  $J$  ως προς το βάρος  $w_{jk}^l$ , μπορεί να γραφεί και ως εξής:

$$\frac{\partial J}{\partial w_{jk}^l} = \frac{\partial J}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial w_{jk}^l}$$

όπου  $z_j^l = \sum_{k=1}^n w_{jk}^l a_k^{l-1} + b_j^l$  και  $n$ : ο αριθμός των νευρώνων στο επίπεδο  $l-1$ .

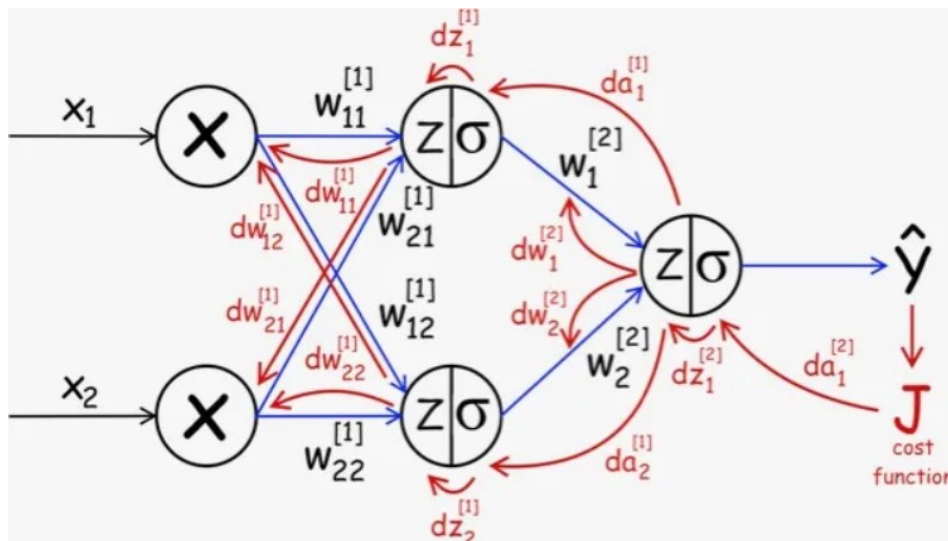
Εφαρμόζοντας τους κανόνες παραγωγίσης στην παραπάνω έκφραση έχουμε:  $\frac{\partial z_j^l}{\partial w_{jk}^l} = a_k^{l-1}$ ,

επομένως:

$$\frac{\partial J}{\partial w_{jk}^l} = \frac{\partial J}{\partial z_j^l} \cdot a_k^{l-1} \quad (4)$$

Η ποσότητα  $\frac{\partial J}{\partial z_j^l}$  μπορεί να υπολογιστεί επίσης με τον κανόνα της αλυσίδας [7].

Η διαδικασία αυτή περιγράφεται σχηματικά στο Σχήμα 13.



Σχήμα 13: Οπισθοδιάδοση Σφάλματος (Backpropagation) [20]

Επομένως, μπορούμε να συνοψίσουμε τη διαδικασία της εκπαίδευσης ενός μοντέλου τεχνητών νευρωνικών δικτύων στα ακόλουθα βήματα:

1. Τη συμπερασματολογία από το μοντέλο με τα τρέχοντα βάρη, για μία δέσμη στοιχείων από το σύνολο δεδομένων.
2. Τον υπολογισμό της τιμής της συνάρτησης σφάλματος με βάση τα βάρη αυτά.

3. Την οπισθοδιάδοση του σφάλματος και τον υπολογισμό της πρώτης παραγώγου της συνάρτησης σφάλματος ως προς κάθε εκπαιδύσιμη παράμετρο όπως τα βάρη και οι μεροληψίες.
4. Τη μεταβολή των βαρών και των μεροληψιών, με βάση τις αντίστοιχες πρώτες παραγώγους που υπολογίστηκαν και με κάποιον από τους παραπάνω αλγορίθμους καθόδου.

τα οποία επαναλαμβάνονται επαναληπτικά.

## Ομαλοποίηση

Η ομαλοποίηση μπορεί να οριστεί ως οποιαδήποτε συμπληρωματική τεχνική που στοχεύει στο να κάνει ένα μοντέλο να “γενικεύει” καλύτερα, δηλαδή να έχει καλύτερη επίδοση στο σύνολο ελέγχου (ή σε οποιοδήποτε σύνολο δεδομένων του οποίου τα στοιχεία δεν βρίσκονται στο σύνολο εκπαίδευσης) [45]. Αυτό επιτυγχάνεται μειώνοντας την πολυπλοκότητα του μοντέλου κατά τη διαδικασία αυτή. Κάποιες από αυτές τις μεθόδους περιγράφονται παρακάτω:

- L<sub>1</sub> – Ομαλοποίηση: Στη μέθοδο αυτή, μεταβάλλουμε τη συνάρτηση κόστους του μοντέλου, προσθέτοντας έναν επιπλέον όρο – *όρος ποιότης*. Η τελική συνάρτηση κόστους έχει την ακόλουθη μορφή:

$$J_{new}(w) = J_{old}(w) + \lambda \sum_{i=1}^n |w_i|$$

Το  $\lambda$  ονομάζεται βαθμός ομαλοποίησης και ισχύει  $\lambda > 0$ . Παρατηρούμε ότι όσο τα μέτρα των βαρών αυξάνονται κατά την εκπαίδευση, οδηγώντας σε υπερπροσαρμογή του μοντέλου, τόσο μεγαλύτερο τελικό σφάλμα λαμβάνουμε. Η μέθοδος αυτή έχει ως αποτέλεσμα οι συντελεστές των λιγότερο σημαντικών χαρακτηριστικών να συρρικνώνονται στο μηδέν, αφαιρώντας έτσι εντελώς κάποιο χαρακτηριστικό.

- L<sub>2</sub> – Ομαλοποίηση: Η μέθοδος αυτή αποτελεί μία παραλλαγή της παραπάνω μεθόδου. Η διαφορά τους είναι πως σε αυτή την περίπτωση, χρησιμοποιούμε τη νόρμα  $\| \cdot \|_2$  των βαρών.

$$J_{new}(w) = J_{old}(w) + \lambda \sum_{i=1}^n w_i^2$$

όπου  $\lambda > 0$  [40].

- Dropout: Η λειτουργία αυτής της μεθόδου είναι ότι κατά τη διάρκεια της εκπαίδευσης, ένα ορισμένο πλήθος νευρώνων ενός μοντέλου απενεργοποιείται, με κάποια πιθανότητα  $p$ . Αυτό μπορεί να συμβεί για πολλούς νευρώνες ταυτόχρονα. Επομένως, με τον τρόπο αυτόν, μπορεί να

γίνει φανερό ποιοι νευρώνες είναι σημαντικοί στην εκπαίδευση του μοντέλου από τις τελικές τιμές των αντίστοιχων βαρών. Το dropout θα μπορούσε επίσης να θεωρηθεί ως λειτουργία που εκπαιδεύει ξεχωριστά πολλά υπομοντέλα και παίρνει το μέσο όρο αυτών κατά τη φάση εξαγωγής συμπερασμάτων [17].

## **Υπερπαραμέτροι Μοντέλων Βαθιάς Μάθησης**

Εκτός από τις παραμέτρους ενός μοντέλου βαθιών νευρωνικών δικτύων που έχουμε συναντήσει μέχρι στιγμής, υπάρχουν επιπρόσθετα κάποια στοιχεία ενός μοντέλου τα οποία αποτελούν παραμέτρους, όμως δεν μεταβάλλονται κατά την εκπαίδευση. Τα στοιχεία αυτά ονομάζονται *υπερπαραμέτροι* και μεταβάλλονται πειραματικά, σύμφωνα με την τελική απόδοση του μοντέλου.

Για παράδειγμα, κατά την κατασκευή ενός Perceptron πολλαπλών επιπέδων έχουμε τις εξής υπερπαραμέτρους:

- Ο αριθμός στρωμάτων – *βάθος* του μοντέλου
- Το πλήθος των μονάδων σε κάθε στρώμα – *πλάτος* του μοντέλου

Κάποιες από τις υπερπαραμέτρους που αφορούν τη διαδικασία της εκπαίδευσης είναι οι εξής:

- Ο ρυθμός εκμάθησης
- Το πλήθος των εποχών
- Το μέγεθος της δέσμης δεδομένων
- Όροι που εξαρτώνται από τον αλγόριθμο βελτιστοποίησης που χρησιμοποιείται (π.χ. ορμή)

Για την ρύθμιση όλων των υπερπαραμέτρων, χρησιμοποιείται ένα ξεχωριστό σύνολο δεδομένων, το οποίο αξιολογεί την απόδοση του μοντέλου, ανά εποχή και ονομάζεται *σύνολο αξιολόγησης*.

## **1.3 Άλλα Είδη Μάθησης**

Εκτός από την επιβλεπόμενη μάθηση, η οποία παρουσιάστηκε στην **Υποενότητα 1.2.1**, υπάρχουν κι άλλα είδη εκπαίδευσης ενός μοντέλου μηχανικής μάθησης, τα οποία διαφοροποιούνται ως προς τη μορφή των δεδομένων που χρησιμοποιούνται, καθώς και ως προς τη γενικότερη διαδικασία εκμάθησης. Οι μέθοδοι εκπαίδευσης περιγράφονται παρακάτω.



### 1.3.1 Μη Επιβλεπόμενη Μάθηση

Η μη επιβλεπόμενη μάθηση χρησιμοποιεί αλγόριθμους μηχανικής μάθησης για την ανάλυση και ομαδοποίηση συνόλων δεδομένων, τα οποία δεν συνοδεύονται από κάποια ετικέτα. Αυτοί οι αλγόριθμοι ανακαλύπτουν κρυμμένα μοτίβα ή ομαδοποιήσεις δεδομένων, χωρίς την ανάγκη ανθρώπινης παρέμβασης. Δηλαδή, χωρίζουν τα δεδομένα σε κλάσεις, οι οποίες δεν είναι προβλεπόμενες [28].

Κάποιοι από τους λόγους για τους οποίους χρησιμοποιείται αυτή η τεχνική εκπαίδευσης είναι, αρχικά, πως όταν καθορίζεται εκ των προτέρων το αποτέλεσμα ενός μοντέλου, απορρίπτονται αυτόματα αποτελέσματα και πληροφορίες υποπεριπτώσεων που μπορεί να κρύβουν τα δεδομένα. Επίσης, στην επιβλεπόμενη μάθηση, η “ποιότητα” της εκπαίδευσης εξαρτάται από το πλήθος των δεδομένων στα οποία έχουν αποδοθεί ετικέτες. Εκτός από αυτό, η απόδοση ετικετών σε κάθε στοιχείο για ένα αρκετά μεγάλο σύνολο δεδομένων είναι μία αντίστοιχα χρονοβόρα διαδικασία [29].

Τα μοντέλα μη επιβλεπόμενης μάθησης χρησιμοποιούνται για τρεις κύριους σκοπούς: ομαδοποίηση, συσχέτιση και μείωση διαστάσεων. Παρακάτω περιγράφονται μερικά παραδείγματα τέτοιων αλγορίθμων.

- Αλγόριθμος K-Means: Η μέθοδος K-Means αναφέρεται ως “αποκλειστική” μέθοδος, όπου ως “αποκλειστικές”, εννοούμε τις μεθόδους ομαδοποίησης όπου κάθε στοιχείο δεν μπορεί να ανήκει σε παραπάνω από μία ομάδες. Τα στοιχεία δεδομένων (αποτελούν σημεία σε χώρο  $d$  - διαστάσεων) κατανέμονται σε  $K$  ομάδες, όπου το  $K$  αντιπροσωπεύει τον αριθμό των ομάδων, με βάση την απόσταση από το κέντρο κάθε ομάδας. Τα σημεία δεδομένων που βρίσκονται πλησιέστερα σε ένα δεδομένο κέντρο, θα ομαδοποιηθούν στην ίδια κατηγορία. Το πλήθος των κέντρων αποτελεί υπερπαραμέτρο. Με μια μεγαλύτερη τιμή του  $K$  θα δημιουργηθούν μικρότερες ομάδες με μεγαλύτερη ακρίβεια στην ταξινόμηση, ενώ με μια μικρότερη τιμή  $K$  θα δημιουργηθούν μεγαλύτερες ομάδες με μικρότερη ακρίβεια αντίστοιχα. Ωστόσο, μεγάλες τιμές του  $K$  μπορεί να οδηγήσουν σε υπερπροσαρμογή και αρκετά αυξημένο υπολογιστικό κόστος, οπότε η επιλογή των κέντρων πρέπει να γίνεται βέλτιστα με βάση το εκάστοτε πρόβλημα.
- Αυτόματοι Κωδικοποιητές: Ανήκουν στις μεθόδους μείωσης διαστάσεων. Αξιοποιούν μοντέλα νευρωνικών δικτύων, για να συμπιέζουν τα δεδομένα και στη συνέχεια να αναδημιουργούν μια νέα αναπαράσταση της εισόδου των αρχικών δεδομένων [28]. Ένας αυτόματος κωδικοποιητής αποτελείται από τα εξής τρία μέρη:
  1. Κωδικοποιητής: Μία μονάδα που συμπιέζει τα δεδομένα εισόδου σε μια κωδικοποιημένη αναπαράσταση που είναι συνήθως αρκετές τάξεις μεγέθους μικρότερη από τα δεδομένα εισόδου.
  2. Σημείο Συμφόρησης - Bottleneck: Ένα δομικό στοιχείο που περιέχει τις συμπιεσμένες αναπαραστάσεις γνώσης.

3. Αποκωδικοποιητής: Μια μονάδα που συμβάλλει στην "αποσυμπίεση" των αναπαραστάσεων γνώσης και στην ανακατασκευή των δεδομένων από την κωδικοποιημένη μορφή τους [54].

### 1.3.2 Ενισχυτική Μάθηση

Η προσέγγιση της ενισχυτικής μάθησης επικεντρώνεται πολύ περισσότερο στη μάθηση με βάση το στόχο μέσω της αλληλεπίδρασης, σε σχέση με άλλες προσεγγίσεις της μηχανικής μάθησης. Η οντότητα που μαθαίνει, δεν λαμβάνει οδηγίες για το ποιες ενέργειες πρέπει να εκτελέσει, αλλά αντίθετα πρέπει να ανακαλύψει μόνη της αυτές που παράγουν τη μεγαλύτερη ανταμοιβή, δοκιμάζοντάς τις και λαμβάνοντας την αντίστοιχη απόκριση. Επιπλέον, αυτές οι ενέργειες μπορούν να επηρεάσουν όχι μόνο την άμεση ανταμοιβή, αλλά και τις μελλοντικές, αφού οι τρέχουσες ενέργειες θα καθορίσουν τις μελλοντικές καταστάσεις.

Στην ενισχυτική μάθηση υπάρχουν δύο βασικές συνιστώσες:

1. Ο *πράκτορας*, ο οποίος είναι ένα πρόγραμμα υπολογιστή με μοναδικό ρόλο τη λήψη αποφάσεων (ενεργειών) για την επίλυση σύνθετων προβλημάτων λήψης αποφάσεων, υπό συνθήκες αβεβαιότητας.
2. Το *περιβάλλον*, το οποίο ανταποκρίνεται με συνέπειες αυτών των ενεργειών, οι οποίες είναι παρατηρήσεις ή καταστάσεις, και ανταμοιβές (θετικές ή αρνητικές).

Στην περίπτωση της βαθιάς μάθησης, ο πράκτορας είναι ένα μοντέλο βαθιών νευρωνικών δικτύων [30].

### 1.3.3 Μεταφορά Γνώσης

Παρακάτω, περιγράφεται μία τεχνική εκπαίδευσης, η οποία έχει ως στόχο να μειώσει τη διάρκεια και τους πόρους για την εκπαίδευση ενός μοντέλου.

Πολλές μέθοδοι μηχανικής μάθησης έχουν καλή απόδοση, μόνο κάτω από την προϋπόθεση ότι τα δεδομένα εκπαίδευσης και τα δεδομένα ελέγχου προέρχονται από τον ίδιο χώρο χαρακτηριστικών και ακολουθούν την ίδια κατανομή. Όταν η κατανομή αλλάζει, τα περισσότερα στατιστικά μοντέλα πρέπει να εκπαιδευτούν από την αρχή, χρησιμοποιώντας νέα δεδομένα εκπαίδευσης [31].

Πολλές φορές, για να επιλύσουμε ένα πρόβλημα που έρχεται στο προσκήνιο, χρησιμοποιούμε γνώσεις από σχετικά προβλήματα που έχουμε ήδη λύσει. Ένα παράδειγμα αυτής της “μεταφοράς γνώσης” είναι το εξής:

*αναγνώριση αυτοκινήτου → αναγνώριση φορτηγού*

Κίνητρο για τη χρήση της μεταφοράς γνώσης μεταξύ μοντέλων, στον τομέα της επιβλεπόμενης μάθησης, αποτελεί το γεγονός πως η δυσκολία ενός προβλήματος προς επίλυση είναι ανάλογη με τον όγκο των δεδομένων που χρειάζονται για την εκπαίδευση ενός μοντέλου βαθιών νευρωνικών δικτύων. Συνεπώς, για ένα δυσεπίλυτο πρόβλημα, θα χρειαστεί ένας αρκετά μεγάλος όγκος δεδομένων, τα οποία είναι δύσκολο να βρεθούν και να αποδοθεί κάποια ετικέτα σε καθένα από αυτά. Με την μεταφορά γνώσης το μέγεθος του συνόλου δεδομένων που απευθύνεται στο καινούριο πρόβλημα μπορεί να μειωθεί έως και εκθετικά. Για το λόγο αυτό, η τεχνική αυτή βελτιστοποιεί τη μετάβαση από ένα πιο ειδικό σε ένα πιο γενικό πρόβλημα [15].

Η μεταφορά γνώσης ορίζεται πιο αυστηρά παρακάτω:

#### Πεδίο:

Ένα πεδίο  $D$  ορίζεται με βάση ένα σύνολο χαρακτηριστικών  $X$  και μια κατανομή πιθανότητας  $P[x]$ , όπου  $x = (x_1, \dots, x_n) \in X$ .

#### Πρόβλημα:

Για ένα δοσμένο πεδίο  $D$ , ένα πρόβλημα  $T$  ορίζεται από ένα σύνολο ετικετών  $Y$  και μια συνάρτηση πρόβλεψης  $f(\cdot)$ , η οποία εκπαιδεύεται με ένα διάνυσμα  $x \in X$  του συνόλου χαρακτηριστικών και την αντίστοιχη ετικέτα  $y \in Y$ .

#### Μεταφορά Γνώσης:

Δεδομένου ενός αρχικού πεδίου  $D_S$ , αρχικού προβλήματος εκπαίδευσης  $T_S$ , ενός τελικού πεδίου  $D_T$  και ενός τελικού προβλήματος εκπαίδευσης  $T_T$  (πρόβλημα – στόχος), η μεταφορά γνώσης έχει ως στόχο τη βελτίωση της εκπαίδευσης της συνάρτησης πρόβλεψης του τελικού προβλήματος  $f_T(\cdot)$  στο πεδίο  $D_T$ , χρησιμοποιώντας τη γνώση από τα  $D_S$  και  $T_S$ , όπου  $D_S \neq D_T$  ή  $T_S \neq T_T$ .

Υπάρχουν διάφορες υποκατηγορίες προβλημάτων μεταφοράς γνώσης, οι οποίες εξαρτώνται από το τελικό πεδίο ή το τελικό πρόβλημα και περιγράφονται παρακάτω:

- Επαγωγική Μεταφορά Γνώσης: Στο πλαίσιο της *επαγωγικής μεταφοράς γνώσης* το πρόβλημα – στόχος είναι διαφορετικό από το αρχικό πρόβλημα, ανεξάρτητα από τη σχέση που έχουν το αρχικό πεδίο  $D_S$ , με το τελικό  $D_T$ . Δηλαδή ισχύει  $T_S \neq T_T$ . Σε αυτή την περίπτωση, είναι απαραίτητο να υπάρχουν στοιχεία δεδομένων από το τελικό πεδίο, στα οποία έχει αποδοθεί κάποια ετικέτα, για τη δημιουργία ενός μοντέλου πρόβλεψης  $f_T(\cdot)$ . Ανάλογα με την κατάσταση των στοιχείων δεδομένων με ή χωρίς ετικέτα, υπάρχουν οι εξής υποκατηγορίες:
  1. Πολλά από τα στοιχεία δεδομένων με ετικέτα στο αρχικό πεδίο είναι διαθέσιμα.
  2. Κανένα από τα στοιχεία δεδομένων με ετικέτα στο αρχικό πεδίο δεν είναι διαθέσιμο.
- Μεταγωγική Μεταφορά Γνώσης: Στο πλαίσιο της *μεταγωγικής μεταφοράς γνώσης*, το αρχικό και το τελικό πρόβλημα ταυτίζονται,  $T_S = T_T$ , ενώ το αρχικό και το τελικό πεδίο είναι διαφορετικά,  $D_S \neq D_T$ . Δεν υπάρχουν διαθέσιμα δεδομένα με ετικέτα στο τελικό πεδίο, ενώ υπάρχουν πολλά επισημασμένα δεδομένα στο αρχικό πεδίο. Επιπλέον, σύμφωνα με διαφορετικές συσχετίσεις μεταξύ του αρχικού και τελικού πεδίου, μπορούμε να κατηγοριοποιήσουμε περαιτέρω τη μεταγωγική μεταφορά γνώσης σε δύο περιπτώσεις.
  1. Ο χώρος των χαρακτηριστικών στο αρχικό πεδίο είναι διαφορετικός από αυτόν στο τελικό, δηλαδή  $X_S \neq X_T$ .
  2. Οι χώροι των χαρακτηριστικών στο αρχικό και το τελικό πεδίο ταυτίζονται, ισχύει:  $X_S = X_T$ , αλλά οι κατανομές των χαρακτηριστικών σε κάθε πεδίο διαφέρουν μεταξύ τους:  $P[X_S] \neq P[X_T]$ .
- Μεταφορά Γνώσης Χωρίς Επίβλεψη: Όσον αφορά τη *μεταφορά γνώσης χωρίς επίβλεψη*, το τελικό πρόβλημα είναι διαφορετικό από το αρχικό  $T_S \neq T_T$ , αλλά σχετίζεται με αυτό. Η μεταφορά γνώσης χωρίς επίβλεψη επικεντρώνεται στην επίλυση προβλημάτων μη επιβλεπόμενης μάθησης στο τελικό πεδίο, επομένως δεν υπάρχουν στοιχεία δεδομένων στα οποία να έχουν αποδοθεί ετικέτες, τόσο στο αρχικό, όσο και στο τελικό πεδίο [31].

Οι τρόποι με τους οποίους μπορούν να επιλυθούν τα παραπάνω προβλήματα μέσω της μεταφοράς γνώσης, περιγράφονται παρακάτω:

- Μεταφορά Στοιχείων: Η επαναχρησιμοποίηση της γνώσης από το αρχικό πεδίο στο τελικό πρόβλημα αποτελεί συνήθως ένα ιδανικό σενάριο. Στις περισσότερες περιπτώσεις, τα δεδομένα του αρχικού πεδίου δεν μπορούν να επαναχρησιμοποιηθούν άμεσα. Αντίθετα, υπάρχουν ορισμένες περιπτώσεις στοιχείων από το αρχικό πεδίο που μπορούν να επαναχρησιμοποιηθούν μαζί με τα δεδομένα του τελικού για τη βελτίωση των αποτελεσμάτων.
- Μεταφορά Παρουσίασης Χαρακτηριστικών: Αυτή η προσέγγιση αποσκοπεί στην ελαχιστοποίηση της διαφοροποίησης των πεδίων και στη μείωση των ποσοστών σφάλματος

μέσω του εντοπισμού καλών αναπαραστάσεων χαρακτηριστικών που μπορούν να χρησιμοποιηθούν από το αρχικό πεδίο στο πεδίο-στόχο. Ανάλογα με τη διαθεσιμότητα προσημασμένων δεδομένων, μπορούν να εφαρμοστούν επιβλεπόμενες ή μη επιβλεπόμενες μέθοδοι.

- Μεταφορά Παραμέτρων: Η προσέγγιση αυτή βασίζεται στην υπόθεση, ότι τα μοντέλα τα οποία επιλύουν συσχετιζόμενα προβλήματα μοιράζονται ορισμένες παραμέτρους ή την εκ των προτέρων κατανομή των υπερπαραμέτρων.
- Μεταφορά Σχεσιακής Γνώσης: Σε αντίθεση με τις τρεις προηγούμενες προσεγγίσεις, η μεταφορά σχεσιακής γνώσης επιχειρεί να διαχειριστεί δεδομένα που δεν είναι ανεξάρτητα και πανομοιότυπα κατανομημένα. Με άλλα λόγια, δεδομένα, όπου κάθε σημείο σχετίζεται με άλλα σημεία δεδομένων. Για παράδειγμα, τα δεδομένα κοινωνικών δικτύων χρησιμοποιούν τεχνικές μεταφοράς σχεσιακής γνώσης [15].

## 1.4 Συνελικτικά Νευρωνικά Δίκτυα

Σε προβλήματα όπου τα δεδομένα εισόδου μπορούν να απεικονιστούν με τη μορφή πλέγματος, όπως οι εικόνες, χρησιμοποιείται ένα είδος μοντέλων νευρωνικών δικτύων, τα οποία εκμεταλλεύονται τη γεωμετρική τοπολογία των δεδομένων και ονομάζονται *συνελικτικά νευρωνικά δίκτυα*. Για να επιτευχθεί αυτό, τα βάρη του δικτύου οργανώνονται σε τρεις διαστάσεις (πλάτος, ύψος, βάθος), ώστε να ταιριάζουν με την τοπολογία της αναπαράστασης των εικόνων [1].

Το όφελος που μπορεί να αποκομίσει κανείς από αυτόν τον τρόπο οργάνωσης των βαρών, είναι οι μειωμένες διαστάσεις στον χώρο των παραμέτρων, σε σχέση με κάποιο αντίστοιχο μοντέλο Perceptron πολλαπλών επιπέδων. Σαν αποτέλεσμα, ο χρόνος που απαιτείται για την εκπαίδευση αυτών των παραμέτρων μειώνεται. Οι νευρώνες στα κρυμμένα στρώματα συνδέονται μόνο με μια μικρή περιοχή του προηγούμενου στρώματος (σε αντίθεση με την πλήρως συνδεδεμένη δομή ενός τυπικού νευρωνικού δικτύου). Αυτός ο τύπος συνδεσιμότητας ονομάζεται *τοπική συνδεσιμότητα*, η οποία μας επιτρέπει να εξοικονομήσουμε ένα τεράστιο ποσό παραμέτρων στο μοντέλο [33]. Η μείωση των διαστάσεων επιτυγχάνεται με τη χρήση ενός ολισθαίνοντος παραθύρου με μέγεθος μικρότερο από αυτό του πίνακα εισόδου. Αυτό το τετραγωνικό τεμάχιο μετατοπίζεται συνεχώς από αριστερά προς τα δεξιά και από πάνω προς τα κάτω για να καλύψει την πλήρη εικόνα [32].

Επομένως, ανάλογα με τη μορφή της εισόδου, υπάρχει και ένας αντίστοιχος τύπος μοντέλου, ο οποίος βελτιστοποιεί την επεξεργασία. Για παράδειγμα, τα επαναληπτικά νευρωνικά δίκτυα χρησιμοποιούνται για την ανάλυση ακολουθιακών δεδομένων, όπως είναι οι χρονοσειρές και τα νευρωνικά δίκτυα γραφημάτων, χρησιμοποιούνται για την ανάλυση γραφημάτων και των σχέσεων που συνδέουν τους κόμβους μεταξύ τους.

### 1.4.1 Συνέλιξη

Αντί για τον πολλαπλασιασμό πινάκων, τα συνελκτικά νευρωνικά δίκτυα χρησιμοποιούν την πράξη της συνέλιξης, η οποία ορίζεται μαθηματικά ως εξής:

$$s(t) = (x * w)(t) = \int_{-\infty}^{+\infty} x(a)w(t-a) da$$

Στην περίπτωση διακριτού αθροίσματος, η παραπάνω συνάρτηση έχει την εξής μορφή:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{+\infty} x(a)w(t-a)$$

Για την εφαρμογή της συνέλιξης στη θεωρία των νευρωνικών δικτύων, η συνάρτηση  $x$  αναπαριστά την είσοδο του μοντέλου, η συνάρτηση  $w$  αναπαριστά τον πυρήνα που περιέχει τα βάρη του και το αποτέλεσμα  $s$  της πράξης ονομάζεται *χάρτης χαρακτηριστικών*.

Για μια εικόνα και έναν πυρήνα δύο διαστάσεων η συνέλιξη ορίζεται ως εξής:

$$S(i, j) = \sum_n \sum_m I(n, m)K(i-n, j-m)$$

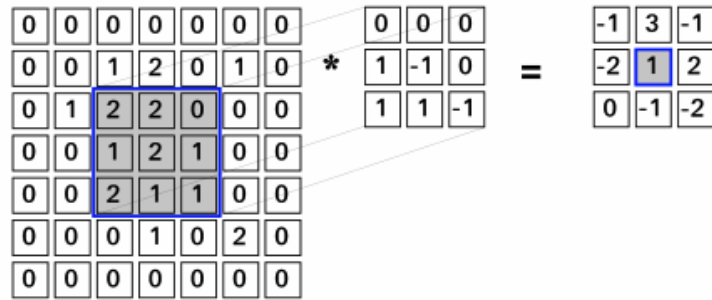
ή

$$S(i, j) = \sum_n \sum_m I(i-n, j-m)K(n, m)$$

Παρ' όλα αυτά, επειδή στη συνέλιξη αντιστρέφεται η σειρά με την οποία επιλέγουμε τα στοιχεία στη συνάρτηση  $K$  ή  $I$  αντίστοιχα, χρησιμοποιείται συχνά η εξής παραλλαγή, η οποία δίνει το ίδιο αποτέλεσμα:

$$S(i, j) = \sum_n \sum_m I(i+n, j+m)K(n, m) \quad (4)$$

και φαίνεται σχηματικά παρακάτω για κάθε τιμή των  $i = 1, 2, \dots, n'$  και  $j = 1, 2, \dots, m'$ , όπου  $n' \times m'$  οι διαστάσεις του χάρτη χαρακτηριστικών εξόδου [1].



Σχήμα 14: Παράδειγμα Συνέλιξης σε Δύο Διαστάσεις [34]

## 1.4.2 Δομικά Στοιχεία Συνελικτικών Δικτύων

Τα συνελικτικά νευρωνικά δίκτυα οργανώνονται σε επίπεδα, καθένα από τα οποία μετασχηματίζει ένα πλήθος από χάρτες χαρακτηριστικών με διαφορετικό τρόπο. Παρακάτω φαίνονται κάποιοι βασικοί τύποι επιπέδων.

### Συνελικτικό Επίπεδο

Το συνελικτικό επίπεδο αποτελεί το κύριο δομικό στοιχείο του μοντέλου και είναι αυτό στο οποίο γίνονται οι περισσότεροι υπολογισμοί. Αποτελείται από φίλτρα με εκπαιδευσίμα βάρη, με τις χωρικές διαστάσεις τους (ύψος, πλάτος) να είναι σημαντικά μικρότερες από τις διαστάσεις της εισόδου του επιπέδου, ενώ το βάθος είναι ίσο με αυτό της εισόδου. Εφαρμόζοντας την πράξη της συνέλιξης ανάμεσα στην είσοδο και σε καθένα από τα φίλτρα σχηματίζονται δισδιάστατοι χάρτες ενεργοποίησης (ή χάρτες χαρακτηριστικών), καθένας από τους οποίους απεικονίζει την απόκριση του αντίστοιχου φίλτρου σε κάθε χωρίο της εισόδου [12]. Στους χάρτες ενεργοποίησης, κάθε νευρώνας συνδέεται μόνο με μια τοπική περιοχή του όγκου εισόδου, το μέγεθος της οποίας ονομάζουμε *δεκτικό πεδίο* του νευρώνα [33]. Λέμε ότι ένα φίλτρο “ενεργοποιείται” όταν εντοπίσει κάποιο χαρακτηριστικό. Η έξοδος ενός συνελικτικού επιπέδου είναι η παράθεση των χαρτών ενεργοποίησης.

Στην απλούστερη εκδοχή της, η πράξη της συνέλιξης για μία είσοδο με διαστάσεις  $H \times W$ , όπου  $H$  είναι το ύψος,  $W$  το πλάτος και  $k \times k$  οι διαστάσεις ενός τετραγωνικού φίλτρου, πραγματοποιείται όπως φαίνεται στο Σχήμα 14, όπου το φίλτρο “σαρώνει” την είσοδο, εκτελώντας σε κάθε βήμα την πράξη του πολλαπλασιασμού μεταξύ των στοιχείων του και των αντίστοιχων στοιχείων της εισόδου και προσθέτοντας στο τέλος τα αντίστοιχα γινόμενα. Το αποτέλεσμα της διαδικασίας αυτής αποτελεί στοιχείο του πίνακα χαρακτηριστικών με θέση  $(i, j)$ , όπου  $i$  το πλήθος των βημάτων που έχουν γίνει ως προς την διεύθυνση του ύψους και  $j$  το πλήθος των βημάτων που έχουν γίνει ως προς την διεύθυνση

του πλάτους ως προς το ύψος  $i$ . Το φίλτρο “σαρώνει” την είσοδο με βήμα 1 pixel. Σαν αποτέλεσμα, το μέγεθος του παραγόμενου χάρτη χαρακτηριστικών θα είναι:

$$(H-k+1) \times (W-k+1)$$

Με βάση ορισμένες υπερπαραμέτρους, η παραπάνω διαδικασία μπορεί να διαφοροποιηθεί ως προς διάφορους παράγοντες, επηρεάζοντας έτσι το μέγεθος και τις διαστάσεις του παραγόμενου χάρτη χαρακτηριστικών. Οι βασικές υπερπαραμέτροι από τις οποίες καθορίζονται οι διαστάσεις της εξόδου είναι οι εξής:

1. Το *πλήθος των φίλτρων* που υπάρχουν στο συνελκτικό επίπεδο, το οποίο ταυτίζεται με το βάθος της εξόδου  $C_{out}$ . Σκοπός κάθε φίλτρου είναι να “μάθει” να αναγνωρίζει ένα διαφορετικό χαρακτηριστικό.
2. Οι *χωρικές διαστάσεις των φίλτρων*, καθώς με την αύξηση στο ύψος ή το πλάτος ενός φίλτρου, έχουμε μείωση στις αντίστοιχες διαστάσεις της εξόδου. Ταυτίζονται με το δεκτικό πεδίο καθενός από τους νευρώνες της εξόδου. Συνήθως επιλέγουμε διαστάσεις  $3 \times 3$ .
3. Ο *βηματισμός (stride)  $s$* , ο οποίος καθορίζει τη μετατόπιση ενός φίλτρου σε pixels κατά την πράξη της συνέλιξης. Μεγαλύτερος βηματισμός οδηγεί σε χάρτη ενεργοποίησης μικρότερων διαστάσεων. Συνήθως επιλέγουμε 1 ή 2 pixels, το οποίο έρχεται σε αναλογία με τις συνήθεις διαστάσεις των φίλτρων. Επομένως, οι διαστάσεις της εξόδου, όταν έχουμε ίδιο βηματισμό στις διευθύνσεις του ύψους και του πλάτους, μπορούν να προσδιοριστούν από τον ακόλουθο τύπο:

$$\left[ \frac{(H-k)}{s} + 1 \right] \times \left[ \frac{(W-k)}{s} + 1 \right] \times C_{out}$$

4. Το *μέγεθος του παραγεμίματος με μηδενικά (zero-padding)*, που αποτελεί μια τεχνική για τον έλεγχο των διαστάσεων της εξόδου. Για παράδειγμα, αν αυτό ταυτίζεται με 1, τότε το αρχικό ύψος και το πλάτος των χαρτών ενεργοποίησης αυξάνονται κατά 2 pixels και στις κενές θέσεις εισάγονται μηδενικά. Μια περίπτωση χρήσης της τεχνικής αυτής είναι ώστε οι διαστάσεις της εισόδου να ταυτίζονται με αυτές της εξόδου [12]. Αυτό είναι ιδιαίτερα χρήσιμο, στις βαθιές αρχιτεκτονικές συνελκτικών δικτύων που εφαρμόζονται πολλαπλά φίλτρα συνέλιξης διαδοχικά μεταξύ τους [33]. Επομένως, εφαρμόζοντας τη μέθοδο του παραγεμίματος με μηδενικά, με  $P$  σειρές μηδενικών σε κάθε διεύθυνση και βηματισμό  $s$ , οι διαστάσεις της εξόδου μπορούν να προσδιοριστούν ως εξής:

$$\left[ \frac{(H-k+2P)}{s+1} \right] \times \left[ \frac{(W-k+2P)}{s+1} \right] \times C_{out}$$

5. Το *μέγεθος της διαστολής  $l$*  που εφαρμόζεται σε μια συνέλιξη. Με βάση αυτήν την υπερπαραμέτρο, η συνέλιξη μπορεί να εφαρμοστεί σε στοιχεία του χάρτη ενεργοποίησης τα οποία απέχουν απόσταση  $l$  μεταξύ τους. Όσο μεγαλύτερη διαστολή εφαρμόζεται σε ένα συνελκτικό στρώμα, τόσο μικρότερος θα είναι ο παραγόμενος χάρτης ενεργοποίησης.



Εκτός από την απλή εφαρμογή της πράξης της συνέλιξης, υπάρχουν αρκετές παραλλαγές οι οποίες χρησιμοποιούνται και πολλές από αυτές προσφέρουν βελτίωση στο υπολογιστικό κόστος. Κάποιες από αυτές παρουσιάζονται παρακάτω:

- Συνέλιξη με Βηματισμό: Σε αυτόν τον τύπο συνέλιξης έχουμε βηματισμό μεγαλύτερο από 1, που ισχύει στην περίπτωση της απλής συνέλιξης.
- Συνέλιξη με Παραγέμισμα με Μηδενικά: Στην περίπτωση αυτή, χρησιμοποιείται η τεχνική του παραγεμίματος με μηδενικά, με σκοπό οι διαστάσεις της εξόδου ενός συνελκτικού επιπέδου να ταυτίζονται με αυτές της εισόδου.
- Διεσταλμένη Συνέλιξη: Στην απλή περίπτωση συνέλιξης ισχύει  $l=0$ . Αυτό μπορεί να είναι πολύ χρήσιμο, σε ορισμένες περιπτώσεις, για χρήση σε συνδυασμό με φίλτρα με διαστολή ίση με 0, αφού με αυτόν τον τρόπο μπορούν να συγχωνευθούν χωρικές πληροφορίες από τις εισόδους πολύ πιο επιθετικά χρησιμοποιώντας λιγότερα επίπεδα [12].
- Αποσυνέλιξη: Αποτελεί την αντίθετη πράξη της συνέλιξης και διαχέει την πληροφορία που υπάρχει από ένα pixel, σε διάφορα pixel [48].
- Κατά Βάθος Συνέλιξη: Για κάθε χάρτη ενεργοποίησης, υπάρχει διαφορετικό φίλτρο (δύο διαστάσεων) με το οποίο γίνεται η πράξη της συνέλιξης και δημιουργείται ένας καινούριος χάρτης. Με τον τρόπο αυτό έχουμε λιγότερες παραμέτρους.
- Ομαδοποιημένη Συνέλιξη: Στην περίπτωση αυτή, οι χάρτες χαρακτηριστικών χωρίζονται σε ομάδες σε σχέση με το βάθος τους και χρησιμοποιείται ένα φίλτρο για κάθε ομάδα.
- Σημειακή Συνέλιξη: Αποτελεί μία 1x1 χωρική συνέλιξη [12].
- Διαχωρίσιμη Συνέλιξη: Υπάρχουν δύο κατηγορίες διαχωρίσιμων συνελίξεων και είναι οι εξής:
  - Χωρικά Διαχωρίσιμη Συνέλιξη: Η χωρικά διαχωρίσιμη συνέλιξη ονομάζεται έτσι επειδή ασχολείται με τις χωρικές διαστάσεις ενός πυρήνα: το πλάτος και το ύψος. Μια χωρικά διαχωρίσιμη συνέλιξη, διαιρεί έναν πυρήνα σε δύο μικρότερους πυρήνες. Η πιο συνηθισμένη περίπτωση είναι να διαιρεθεί ένας πυρήνας 3x3 σε δύο πυρήνες διαστάσεων 3x1 και 1x3. Δηλαδή, αντί για μία συνέλιξη με 9 πολλαπλασιασμούς, έχουμε δύο συνελίξεις με 3 πολλαπλασιασμούς η κάθε μία (6 συνολικά) για να επιτύχουμε το ίδιο αποτέλεσμα. Επομένως, η υπολογιστική πολυπλοκότητα μειώνεται. Το μειονέκτημα της μεθόδου αυτής, είναι πως λίγοι πυρήνες έχουν την ιδιότητα να διαχωρίζονται με τον τρόπο αυτό και έτσι χρησιμοποιείται σπάνια στην πράξη [36].
  - Κατά Βάθος Διαχωρίσιμη Συνέλιξη: Αυτός ο τύπος συνέλιξης προήλθε από την ιδέα ότι το βάθος και οι χωρικές διαστάσεις ενός φίλτρου μπορούν να διαχωριστούν. Δηλαδή, εκτελούνται αρχικά οι κατά βάθος συνελίξεις και έπειτα μία σημειακή συνέλιξη. Αντίστοιχα με την παραπάνω περίπτωση μειώνεται ο αριθμός των παραμέτρων και σαν αποτέλεσμα το υπολογιστικό κόστος [37].

## Επίπεδο Ενεργοποίησης

Πολλές φορές, μετά από τις πράξεις που εκτελούνται σε ένα συνελκτικό επίπεδο, εφαρμόζεται μία συνάρτηση ενεργοποίησης στον παραγόμενο χάρτη χαρακτηριστικών. Αυτό μπορεί να αναπαρασταθεί σαν ένα ξεχωριστό επίπεδο. Ένα στρώμα ενεργοποίησης δέχεται έναν όγκο εισόδου και στη συνέχεια εφαρμόζει μία δεδομένη συνάρτηση ενεργοποίησης. Δεδομένου ότι η συνάρτηση ενεργοποίησης εφαρμόζεται κατά στοιχείο, η διάσταση της εξόδου ενός στρώματος ενεργοποίησης είναι πάντα η ίδια με τη διάσταση εισόδου [33].

## Επίπεδο Συγκέντρωσης

Το επίπεδο συγκέντρωσης έχει ως λειτουργία να μειώσει τις χωρικές διαστάσεις των χαρτών ενεργοποίησης και συνήθως βρίσκεται μεταξύ των συνελκτικών επιπέδων [12]. Αυτό μας επιτρέπει να μειώσουμε το πλήθος των παραμέτρων και των υπολογισμών στο μοντέλο [33]. Η μείωση των διαστάσεων έχει ως στόχο την ελάττωση των παραμέτρων του μοντέλου, καθώς και των μετέπειτα υπολογισμών και έτσι τον έλεγχο της υπερπροσαρμογής των παραμέτρων του μοντέλου στο σύνολο εκπαίδευσης.

Το επίπεδο αυτό ορίζεται από μια συνάρτηση ομαδοποίησης που δρα σε χωρία συγκεκριμένων διαστάσεων της εισόδου. Το μέγεθος της εξόδου εξαρτάται από τις διαστάσεις του χωρίου και το βήμα (stride) με το οποίο μετακινείται στο ύψος και στο πλάτος της εισόδου. Συνήθως οι διαστάσεις του χωρίου είναι  $2 \times 2$  με βήμα 2 pixels. Καμία από τις παραπάνω παραμέτρους δεν εξαρτάται από τα δεδομένα που επεξεργάζεται το μοντέλο, επομένως δεν είναι εκπαιδευσιμες. Οι κύριες συναρτήσεις που χρησιμοποιούνται είναι οι εξής:

1. Η συνάρτηση  $max\{\}$  στην οποία επιλέγεται η μέγιστη τιμή από κάθε χωρίου της εισόδου.
2. Η συνάρτηση  $avg\{\}$  στην οποία υπολογίζεται η μέση τιμή των τιμών που βρίσκονται στο εκάστοτε χωρίο της εισόδου [12].

## Επίπεδο Κανονικοποίησης

Υπάρχουν πολλοί τύποι επιπέδων κανονικοποίησης, οι οποίοι αναφέρονται τόσο στους χάρτες ενεργοποίησης όσο και στα βάρη. Η κανονικοποίηση που αφορά τους χάρτες ενεργοποίησης έχει ως στόχο η είσοδος και η έξοδος ενός συνελκτικού επιπέδου να έχουν κοινή κατανομή, ενώ οι μέθοδοι κανονικοποίησης βαρών να κανονικοποιήσουν τα βάρη έτσι ώστε η συνάρτηση κόστους να είναι πιο ομαλή. Συνήθως, για την πρώτη κατηγορία, τα επίπεδα κανονικοποίησης μετασχηματίζουν τα δεδομένα έτσι ώστε για κάθε χαρακτηριστικό να έχουν μέση τιμή ίση με 0 και διασπορά ίση με 1 [12].

Στη γενική του μορφή, ένα συνελικτικό νευρωνικό δίκτυο μπορεί να χωριστεί σε δύο μέρη, τον κωδικοποιητή και τον αποκωδικοποιητή. Ο κωδικοποιητής, η αρχιτεκτονική του οποίου είναι η ίδια για κάθε δίκτυο ανεξαρτήτως διεργασίας, είναι υπεύθυνος για την εξαγωγή των χαρακτηριστικών της εκάστοτε εικόνας εισόδου, ενώ ο αποκωδικοποιητής είναι υπεύθυνος για την περαιτέρω επεξεργασία των εξαγόμενων χαρακτηριστικών με βάση τη διεργασία την οποία το δίκτυο καλείται να εκτελέσει (π.χ. κατηγοριοποίηση εικόνων, εντοπισμός αντικειμένων, κατάτμηση εικόνας, εκτίμηση ανθρώπινης πόζας, παραγωγή εικόνων, κ.α.). Τα είδη επιπέδων που παρουσιάστηκαν παραπάνω χρησιμοποιούνται κατά κόρον στον κωδικοποιητή - εξαγωγέα χαρακτηριστικών των συνελικτικών δικτύων, ενώ, ανάλογα με τη διεργασία, ο αποκωδικοποιητής μπορεί να χρησιμοποιεί αυτά, είτε και άλλους τύπους επιπέδων. Για παράδειγμα, ένα συνελικτικό μοντέλο για κατηγοριοποίηση εικόνων συνήθως χρησιμοποιεί στον αποκωδικοποιητή του, ο οποίος ονομάζεται και ταξινομητής, τους δύο τύπους επιπέδων που περιγράφονται παρακάτω.

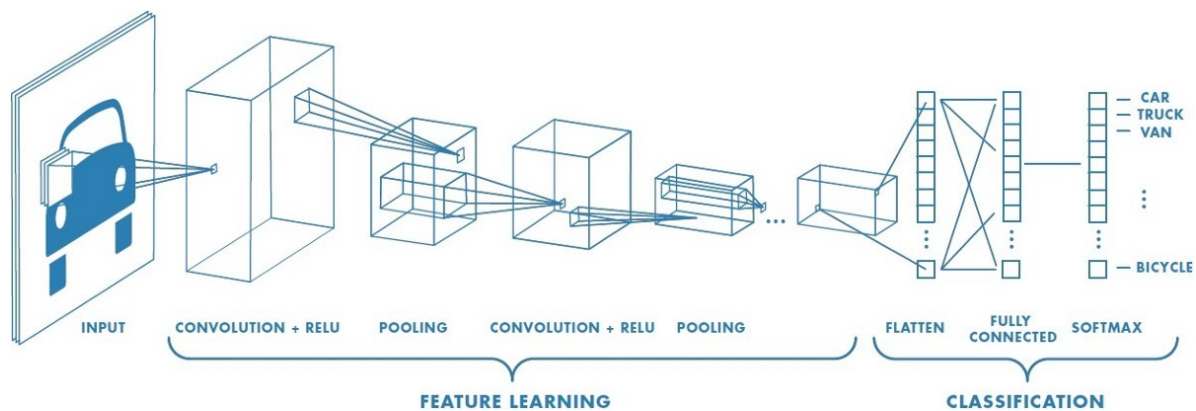
### **Επίπεδο Ολικής Συγκέντρωσης**

Εκτός από τις τεχνικές συγκέντρωσης που αναφέρονται παραπάνω, υπάρχει και αυτή της συνολικής συγκέντρωσης. Σε αυτήν την περίπτωση, συνήθως, χρησιμοποιείται η συνάρτηση ομαδοποίησης *avg*. Η συνάρτηση ομαδοποίησης στο επίπεδο αυτό εφαρμόζεται σε ολόκληρους τους χάρτες ενεργοποίησης. Η τεχνική αυτή έχει σχεδιαστεί για να αντικαταστήσει τον μεγάλο αριθμό πλήρως συνδεδεμένων στρωμάτων στα κλασικά συνελικτικά δίκτυα και έχει εφαρμογή σε προβλήματα όπως η ταξινόμηση εικόνων. Η βασική ιδέα είναι να δημιουργηθεί ένας χάρτης χαρακτηριστικών για κάθε αντίστοιχη κατηγορία ταξινόμησης στο τελευταίο στρώμα από νευρώνες. Αντί να προσθέσουμε πλήρως συνδεδεμένα στρώματα πάνω από τους χάρτες χαρακτηριστικών, παίρνουμε το μέσο όρο κάθε χάρτη και το διάνυσμα που προκύπτει τροφοδοτείται απευθείας στο στρώμα που εκτελεί την ταξινόμηση. Έτσι, οι χάρτες χαρακτηριστικών μπορούν εύκολα να ερμηνευθούν ως χάρτες εμπιστοσύνης κατηγοριών [49].

### **Πλήρως Συνδεδεμένο Επίπεδο**

Το πλήρως συνδεδεμένο επίπεδο αποτελείται από νευρώνες οι οποίοι συνδέονται με κάθε προηγούμενη ενεργοποίηση, όπως έχει αναφερθεί προηγουμένως. Έτσι οι νέες ενεργοποιήσεις υπολογίζονται τώρα με πολλαπλασιασμό πινάκων. Σε μοντέλα συνελικτικών δικτύων τα οποία έχουν ως σκοπό την ταξινόμηση της εισόδου ως προς δοσμένες κλάσεις, τα τελευταία επίπεδα είναι πλήρως συνδεδεμένα [12].

Ένα παράδειγμα ενός μοντέλου συνελκτικών δικτύων για ταξινόμηση φαίνεται στο Σχήμα 15.



Σχήμα 15: Μοντέλο Συνελκτικών Δικτύων για Ταξινόμηση [35]

## 2. Μέθοδοι Συμπίεσης Βαθιών Νευρωνικών Δικτύων

Τα τελευταία χρόνια, η βαθιά μάθηση έχει σημειώσει σημαντική πρόοδο σε περίπλοκα προβλήματα, όπως η ταξινόμηση εικόνων, ο εντοπισμός αντικειμένων σε μία εικόνα, η σύνθεση φωνής και η κατάτμηση εικόνας [17]. Παρ' όλα αυτά, οι αυξημένες επιδόσεις στην ακρίβεια συνδυάζονται, συνήθως, με αρκετά μεγάλο υπολογιστικό και αποθηκευτικό κόστος για τα μοντέλα βαθιάς μάθησης. Αυτές οι απαιτήσεις όχι μόνο αυξάνουν το κόστος των υποδομών, αλλά καθιστούν επίσης δύσκολη την ενσωμάτωση των μοντέλων σε περιβάλλοντα με περιορισμένους πόρους, όπως τα κινητά τηλέφωνα ή οι έξυπνες συσκευές [16].

Στην πραγματικότητα, σε πολλές περιπτώσεις, τα μοντέλα βαθιάς μάθησης περιλαμβάνουν περισσότερες παραμέτρους από όσες είναι απαραίτητες για να έχουν τη μέγιστη δυνατή ακρίβεια σε ένα δοσμένο πρόβλημα. Το φαινόμενο αυτό ονομάζεται *υπερπαραμετροποίηση*. Επομένως, με την αφαίρεση της περιττής πληροφορίας με αποδοτικό τρόπο, ένα τέτοιο μοντέλο μπορεί να “συμπιεστεί”, να έρθει δηλαδή σε μια πιο πρακτική μορφή, με μικρή απώλεια ακρίβειας. Σε ορισμένες, σπάνιες, περιπτώσεις η ακρίβεια μπορεί ακόμη και να βελτιωθεί [17].

Η παραπάνω συμπίεση μπορεί να επιτευχθεί με τη χρήση αλγορίθμων βελτιστοποίησης ή συμπίεσης βαθιών νευρωνικών δικτύων, οι οποίοι εκτός από μείωση στο μέγεθος των μοντέλων, μπορεί να οδηγήσουν και σε βελτίωση του χρόνου απόκρισης τους. Οποιαδήποτε μέθοδος συμπίεσης είναι αποδοτικότερο να εφαρμόζεται σε ήδη εκπαιδευμένα μοντέλα, καθώς η παράλληλη εφαρμογή μιας μεθόδου συμπίεσης κατά την αρχική εκπαίδευση ενός μοντέλου συνήθως οδηγεί σε υποβέλτιστη ακρίβεια. Τις περισσότερες φορές, για την εφαρμογή μιας μεθόδου, απαιτείται ένας νέος γύρος εκπαίδευσης ή ορθότερα προσαρμογής του μοντέλου στη μέθοδο, ώστε το μοντέλο να ανακτήσει όσο το δυνατόν μεγαλύτερο ποσοστό από την ακρίβεια που έχασε. Ωστόσο, όπως θα αναφερθεί και στην **Ενότητα 4.5** για κάθε μέθοδο ξεχωριστά, αυτό το επιπλέον βήμα προσαρμογής δεν είναι πάντα απαραίτητο, αλλά ούτε και πάντα εφικτό, για παράδειγμα όταν δεν έχουμε πρόσβαση στα δεδομένα εκπαίδευσης. Στην παρούσα διπλωματική εργασία, θα ασχοληθούμε με αλγορίθμους κλαδέματος, κβαντοποίησης και ομαδοποίησης, οι οποίοι παρουσιάζονται αναλυτικά στη συνέχεια.

### 2.1 Κλάδεμα

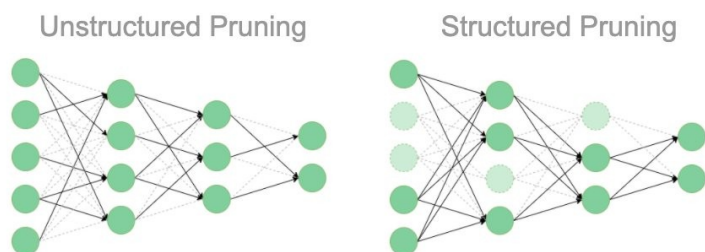
Το κλάδεμα αποτελεί μια αρκετά δημοφιλή μέθοδο συμπίεσης βαθιών νευρωνικών δικτύων. Σκοπός της είναι η αφαίρεση περιττών παραμέτρων ή νευρώνων που δεν συμβάλλουν στην ακρίβεια των αποτελεσμάτων. Αυτή η συνθήκη μπορεί να προκύψει όταν οι συντελεστές βάρους είναι μηδενικοί, κοντά στο μηδέν ή επαναλαμβάνονται (στην περίπτωση αφαίρεσης νευρώνων). Το κλάδεμα, κατά συνέπεια, μειώνει το αποθηκευτικό κόστος και την υπολογιστική πολυπλοκότητα. Εάν τα κλαδεμένα μοντέλα επανεκπαιδευτούν, τότε υπάρχει η δυνατότητα διαφυγής από ένα προηγούμενο τοπικό ελάχιστο και περαιτέρω βελτίωσης της ακρίβειας [17].

## Κατηγορίες Τεχνικών Κλαδέματος

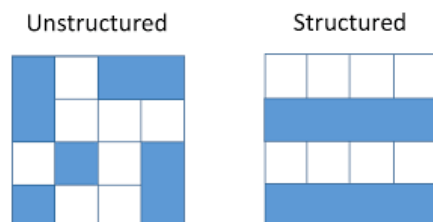
Υπάρχουν αρκετές τεχνικές εφαρμογής της παραπάνω μεθόδου, οι οποίες εξαρτώνται από τον τύπο των παραμέτρων στις οποίες εφαρμόζεται. Ως προς τον τύπο των παραμέτρων οι οποίες κλαδεύονται, μπορούμε να διακρίνουμε τις εξής τεχνικές:

1. Μη δομικό κλάδεμα: Στην περίπτωση αυτή, αξιολογούμε μεμονωμένα τα βάρη με κάποιο κριτήριο μηδενισμού. Με τον τρόπο αυτό παράγεται ένα αραιό -ως προς τις παραμέτρους- μοντέλο νευρωνικών δικτύων, το οποίο -αν και μικρότερο από πλευράς αριθμού παραμέτρων- μπορεί να μην είναι δομημένο με τρόπο που να ευνοεί την επιτάχυνση των υπολογισμών με τη χρήση σύγχρονων βιβλιοθηκών και υλικού.
2. Δομικό κλάδεμα: Στην περίπτωση αυτή, αξιολογούμε τη σημασία των ενεργοποιήσεων ή των χαρτών ενεργοποίησης, με κάποιο κριτήριο μηδενισμού, έτσι ώστε να μηδενιστούν όλα τα σχετικά βάρη, σε ομάδες. Με αυτόν τον τρόπο αφαιρούνται νευρώνες ή ακόμα και ολόκληροι χάρτες ενεργοποίησης, με την αφαίρεση φίλτρων (στα συνελκτικά νευρωνικά δίκτυα), για την εκμετάλλευση υλικού και λογισμικού βελτιστοποιημένου για πυκνούς υπολογισμούς [16].

Παρακάτω μπορούμε να δούμε πώς εφαρμόζεται κάθε τύπος κλαδέματος σε ένα μοντέλο Perceptron πολλαπλών επιπέδων. Αρχικά, φαίνεται μία σχηματική αναπαράσταση ενός τέτοιου μοντέλου στο οποίο έχει εφαρμοστεί μη δομικό και δομικό κλάδεμα (Σχήμα 16) και έπειτα φαίνεται, πιο αναλυτικά, η μορφή των πινάκων των βαρών, ενός τέτοιου μοντέλου, μετά από κάθε μορφή κλαδέματος αντίστοιχα (Σχήμα 17).



Σχήμα 16: Μη Δομικό και Δομικό Κλάδεμα - Γραφική Αναπαράσταση [18]



Σχήμα 17: Μη Δομικό και Δομικό Κλάδεμα Αναπαράσταση Πινάκων [19]

Σαν αποτέλεσμα, στην πρώτη περίπτωση κλαδέματος, περιμένουμε να έχουμε πολύ αραιούς πίνακες, δηλαδή πίνακες στους οποίους πάρα πολλά από τα στοιχεία να είναι ίσα με 0. Στη δεύτερη περίπτωση, όμως, περιμένουμε να έχουμε ένα συρρικνωμένο μοντέλο, το οποίο θα εκτελεί μικρότερο αριθμό πράξεων στα συνελκτικά επίπεδα, αφού ο αριθμός των φίλτρων θα είναι μειωμένος. Στην περίπτωση που είναι δυνατόν να αφαιρεθούν κάποιες σειρές από έναν κλαδεμένο πίνακα βαρών  $W$ , θα έχουμε στην πράξη μείωση διαστάσεων και κατ' επέκταση μείωση του υπολογιστικού κόστους και στα πλήρως συνδεδεμένα επίπεδα.

## Κριτήρια Κλαδέματος

Τα κριτήρια με τα οποία γίνεται το κλάδεμα αποτελούν αρκετά σημαντικό μέρος της διαδικασίας του αλγορίθμου. Μερικοί τύποι κριτηρίων περιγράφονται παρακάτω:

- Κλάδεμα με βάση το μέγεθος των βαρών: Ισχύει ότι τα εκπαιδευμένα βάρη με μεγάλες τιμές είναι σημαντικότερα από αυτά με μικρότερες τιμές. Ένα κριτήριο το οποίο είναι αρκετά διαισθητικό και απορρέει από την παραπάνω πρόταση, είναι το κλάδεμα των βαρών των οποίων η απόλυτη τιμή (μέγεθος) είναι η μικρότερη. Στο δομικό κλάδεμα, το κριτήριο αυτό εφαρμόζεται χρησιμοποιώντας ως μέτρο τη νόρμα  $l^p$ , η οποία ορίζεται ως εξής για ένα διάνυσμα  $x$  μεγέθους  $n$ :

$$\|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}$$

και πιο συγκεκριμένα τις νόρμες  $l^1, l^2$ , για τους πίνακες βαρών ή τους πυρήνες ή τα φίλτρα για τα συνελκτικά επίπεδα αντίστοιχα [17].

- Κλάδεμα βασισμένο σε ποινή: Πολλές μέθοδοι, αντί να κλαδεύουν χειροκίνητα τις συνδέσεις, εφαρμόζουν διάφορα είδη ποινών στα ίδια τα βάρη, ώστε να τα κάνουν να συρρικνώνονται προοδευτικά προς το 0 (σε μερικές περιπτώσεις εφαρμόζεται έπειτα η πρώτη μορφή κλαδέματος). Σήμερα, διάφορες μέθοδοι εφαρμόζουν διαφορετικού τύπου ομαλοποιήσεις ως ποινές, για να αυξήσουν περαιτέρω την αραιότητα στα βάρη ενός μοντέλου. Κάποια τέτοια παραδείγματα είναι η χρήση της νόρμας  $l_1$  ή της νόρμας  $l_2$  των βαρών, οι οποίες εφαρμόζονται στη συνάρτηση κόστους [41], όπως έχει περιγραφεί αναλυτικότερα στην **Υποενότητα 1.2.3**.

## 2.2 Κβαντοποίηση

Η *κβαντοποίηση* είναι γνωστή ως η διαδικασία προσέγγισης ενός συνεχούς σήματος, από ένα σύνολο διακριτών συμβόλων ή ακέραιων τιμών [17]. Στο πλαίσιο των μοντέλων νευρωνικών δικτύων, η ιδιότητα αυτή έχει ως στόχο τη μείωση του αριθμού των bit για την αναπαράσταση των τιμών που αναφέρονται στα βάρη, τις ενεργοποιήσεις και τις παραγώγους που υπολογίζονται για την οπισθοδιάδοση του σφάλματος. Σαν αποτέλεσμα, μειώνεται το μέγεθος και η υπολογιστική πολυπλοκότητα ενός μοντέλου.

Η μετατροπή των παραμέτρων κινητής υποδιαστολής σε τύπους δεδομένων χαμηλής αριθμητικής ακρίβειας για την κβαντοποίηση μοντέλων νευρωνικών δικτύων προτάθηκε ήδη από τη δεκαετία του 1990. Το ανανεωμένο ενδιαφέρον για την κβαντοποίηση ξεκίνησε τη δεκαετία του 2010, όταν τα βάρη

με τιμές ακρίβειας 8-bit αποδείχθηκε ότι επιταχύνουν την εξαγωγή συμπερασμάτων χωρίς σημαντική πτώση της ακρίβειας [17].

## Ομοιόμορφη Κβαντοποίηση

Η μέθοδος αυτή ονομάζεται ομοιόμορφη, αφού οι προκύπτουσες κβαντισμένες τιμές (ή αλλιώς επίπεδα κβαντοποίησης) είναι ομοιόμορφα κατανεμημένες, δηλαδή έχουν ίσες αποστάσεις ανά δύο μεταξύ τους.

Αρχικά, θα πρέπει να οριστεί ένα εύρος αρχικών τιμών  $[\alpha, \beta]$ , με βάση το οποίο θα εφαρμοστεί η μέθοδος της κβαντοποίησης. Η διαδικασία αυτή ονομάζεται *βαθμονόμηση*. Μια απλή επιλογή είναι η χρήση της ελάχιστης και της μέγιστης τιμής των δοσμένων τιμών  $[\alpha, \beta] = [\min, \max]$ . Μπορούμε επίσης να επιλέξουμε με διάφορους τρόπους τα  $\alpha, \beta$  ώστε να είναι συμμετρικά μεταξύ τους ως προς το 0 [42].

Έστω  $[\alpha, \beta]$  το εύρος των πραγματικών τιμών που επιλέγονται για κβαντοποίηση και  $b$  το πλήθος των bit της αναπαράστασης ενός τελικού προσημασμένου ακέραιου αριθμού. Η ομοιόμορφη κβαντοποίηση μετασχηματίζει την τιμή εισόδου  $x \in [\alpha, \beta]$ , ώστε να βρίσκεται εντός του εύρους  $[-2^{b-1}, 2^{b-1}-1]$ . Τα δεδομένα εισόδου που βρίσκονται εκτός αυτού του εύρους, προσεγγίζονται από το κοντινότερο όριο του εύρους των κβαντισμένων τιμών. Με βάση τα παραπάνω, υπάρχουν δύο επιλογές για τη συνάρτηση κβαντοποίησης και είναι οι εξής:

1. Μη συμμετρική ομοιόμορφη κβαντοποίηση:  $f(x) = \text{round}(s \cdot x) + z$ . Τα μεγέθη  $s, z$  ορίζονται ως εξής:

- $s = \frac{2^b - 1}{\beta - \alpha}$ , όπου το  $\frac{1}{s} = \frac{\beta - \alpha}{2^b - 1}$  παρατηρούμε πως αποτελεί την απόσταση (βήμα) μεταξύ των  $2^b - 1$  ακεραίων (επιπέδων κβαντοποίησης), ώστε να “χωρέσουν” στο αρχικό διάστημα  $[\alpha, \beta]$ .
- $z = -\text{round}(\beta \cdot s) - 2^{b-1}$ , αποτελεί την κβαντισμένη αναπαράσταση του σημείου 0

Η αποκωδικοποίηση γίνεται ως εξής:  $x = \frac{1}{s} x_q - z$ .

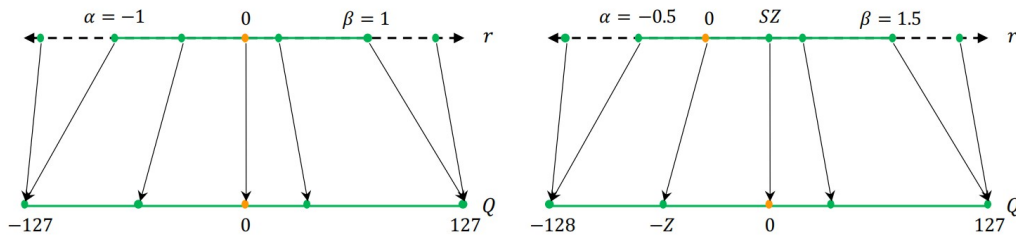
2. Συμμετρική Ομοιόμορφη Κβαντοποίηση:  $f(x) = \text{round}(s \cdot x)$ , η οποία αποτελεί ειδική περίπτωση της παραπάνω, όπου  $z = 0$ . Ονομάζεται συμμετρική, γιατί τα εύρη της εισόδου και της εξόδου είναι συμμετρικά ως προς το 0. Αυτό σημαίνει ότι μία κβαντισμένη τιμή, σε αυτή την περίπτωση, θα βρίσκεται στο εύρος  $[-2^{b-1}+1, 2^{b-1}-1]$ . Το μέγεθος  $s$  ορίζεται ως εξής:

$$s = \frac{2^{b-1} - 1}{\beta}$$



Η αποκωδικοποίηση γίνεται ως εξής:  $x = \frac{1}{S}x_q$  [43].

Παρακάτω φαίνονται σχηματικά οι έννοιες της συμμετρικής και της μη συμμετρικής κβαντοποίησης.



Σχήμα 18: Συμμετρική και Μη Συμμετρική Κβαντοποίηση με Αναπαράσταση σε 8 bits [42]

### Μη Ομοιόμορφη Κβαντοποίηση

Μία άλλη προσέγγιση της κβαντοποίησης είναι η μη ομοιόμορφη κβαντοποίηση, όπου η απόσταση ανάμεσα σε δύο διαδοχικά επίπεδα κβαντοποίησης επιτρέπεται να μην είναι ένα σταθερό μέγεθος. Η μέθοδος αυτή μπορεί να επιτύχει μεγαλύτερη ακρίβεια για ένα σταθερό πλήθος bit, επειδή μπορεί να συλλάβει καλύτερα τον τρόπο που κατανέμονται οι αρχικές τιμές, εστιάζοντας περισσότερο σε σημαντικές περιοχές ή βρίσκοντας κατάλληλα αρχικά εύρη  $[\alpha, \beta]$ .

Ωστόσο, τα μη ομοιόμορφα σχήματα κβαντοποίησης είναι συνήθως δύσκολο να αναπτυχθούν αποτελεσματικά σε υλικό γενικών υπολογισμών (GPU, CPU). Ως εκ τούτου, η ομοιόμορφη κβαντοποίηση είναι επί του παρόντος η κυρίαρχη μέθοδος λόγω της απλότητάς της και της αποδοτικής απεικόνισής της στο υλικό [42].

## 2.3 Ομαδοποίηση – Διαμοιρασμός Βαρών

Η *ομαδοποίηση* ή *διαμοιρασμός βαρών*, αποτελεί έναν αλγόριθμο για την επιλογή των αρχικών κεντροειδών του αλγορίθμου KMEANS. Η βασική ιδέα είναι τα αρχικά κεντροειδή να μην οριστούν τυχαία, αλλά με τέτοιο τρόπο ώστε να είναι πιο πιθανό να είναι αντιπροσωπευτικά των πραγματικών συστάδων στο σύνολο δεδομένων. Η βασική ιδέα των μεθόδων που ανήκουν στην κατηγορία αυτή, είναι η κατάταξη των βαρών ενός μοντέλου σε ομάδες, σύμφωνα με συγκεκριμένα κριτήρια που συνοδεύουν την κάθε μέθοδο, την απόδοση κάποιας κοινής τιμής στα βάρη κάθε ομάδας (διαμοιρασμός) και έπειτα τη συμπύεση του μοντέλου χρησιμοποιώντας κάποια αντίστοιχη μέθοδο, όπως για παράδειγμα τον αλγόριθμο Huffman [44].

Ένα παράδειγμα εφαρμογής της ομαδοποίησης, είναι η περίπτωση όπου τα βάρη ενός συνελκτικού δικτύου διαμοιράζονται σε όλα τα φίλτρα ενός στρώματος. Σε αυτό το πλαίσιο, η ομαδοποίηση έχει τα ακόλουθα αποτελέσματα:

- Ο αριθμός των βαρών που πρέπει να ρυθμιστούν μειώνεται, γεγονός που μειώνει το χρόνο και το κόστος εκπαίδευσης του μοντέλου.
- Η αναζήτηση χαρακτηριστικών είναι ανεξάρτητη από τη θέση τους σε μία εικόνα.

Έτσι, το σφάλμα στην εκπαίδευσης μειώνεται, με κόστος στην ευελιξία του μοντέλου. Ο διαμοιρασμός βαρών αποτελεί για όλες τις προθέσεις και τους σκοπούς του μια μορφή ομαλοποίησης. Επομένως, όπως άλλες μορφές ομαλοποίησης, μπορεί στην πραγματικότητα να αυξήσει την απόδοση ενός μοντέλου, σε ορισμένα σύνολα δεδομένων, με υψηλή διακύμανση στη θέση των χαρακτηριστικών, μειώνοντας τη διασπορά περισσότερο από ό,τι αυξάνει την μεροληψία ενός μοντέλου [45] [46].

Ο διαμοιρασμός των βαρών βασίζεται σε αλγορίθμους ομαδοποίησης, για την εύρεση κάποιων τιμών (κεντροειδή) με βάση τις οποίες θα διαμοιραστούν τα βάρη. Δηλαδή, η τελική τιμή ενός βάρους θα είναι ίση με την τιμή του κέντρου με το οποίο έχει τη μικρότερη απόσταση. Για το λόγο αυτό, η αρχικοποίηση των τιμών των κέντρων έχει μεγάλη σημασία στην τελική απόδοση ενός μοντέλου. Κάποια παραδείγματα αρχικοποίησης των κέντρων σε διαφορετικούς αλγορίθμους ομαδοποίησης είναι τα παρακάτω:

- Γραμμική Αρχικοποίηση: Ταξινομώντας τις τιμές από τη μικρότερη στη μεγαλύτερη επιλέγουμε τα κέντρα έτσι ώστε να είναι ομοιόμορφα κατανεμημένα μεταξύ τους.
- KMEANS++: Αποτελεί μια παραλλαγή του αλγορίθμου KMEANS, όπου τα αρχικά κεντροειδή δεν ορίζονται τυχαία, αλλά με τέτοιο τρόπο ώστε είναι πιο πιθανό να είναι αντιπροσωπευτικά των πραγματικών συστάδων στο σύνολο δεδομένων. Υποθέτοντας ότι  $X$  είναι το σύνολο των δοσμένων τιμών, τα βήματα του αλγορίθμου είναι τα εξής:
  1. Επιλέγουμε το πρώτο κεντροειδές από τα σημεία δεδομένων τυχαία.
  2. Για όλες τις υπόλοιπες τιμές  $x \in X$  (που δεν έχουν επιλεγεί ως κεντροειδή) υπολογίζουμε την απόσταση  $D(x)$ , δηλαδή την απόσταση μεταξύ της τιμής  $x$  και του κοντινότερου σε αυτήν κεντροειδούς.
  3. Επιλέγουμε το επόμενο κεντροειδές τυχαία με πιθανότητα:  $p(x) = \frac{D^2(x)}{\sum_{x \in X} D^2(x)}$ . Αυτό σημαίνει ότι τα σημεία που βρίσκονται πιο μακριά από τα υπάρχοντα κεντροειδή είναι πιο πιθανό να επιλεγούν ως νέα κεντροειδή.
  4. Επαναλαμβάνουμε τα βήματα 2 και 3 μέχρι να έχουν επιλεγεί συνολικά  $K$  κεντροειδή [47].

### 3. Τεχνολογίες

Παρακάτω περιγράφονται τα τεχνολογικά εργαλεία που χρησιμοποιήθηκαν στο πειραματικό μέρος της διπλωματικής εργασίας. Συγκεκριμένα, οι βιβλιοθήκες Keras (μέσω της βιβλιοθήκης TensorFlow) και TensorFlow Lite χρησιμοποιήθηκαν στη γλώσσα προγραμματισμού Python με σκοπό την πρόσβαση σε ένα σύνολο από προεκπαιδευμένα στο σύνολο δεδομένων ImageNet μοντέλα, την εφαρμογή των μεθόδων συμπίεσης και την κατασκευή των τελικών μοντέλων προς ανάπτυξη (deployment) αντίστοιχα, ενώ το Android Studio IDE χρησιμοποιήθηκε για την κατασκευή μιας εφαρμογής στη γλώσσα Java για την τοποθέτηση των συμπιεσμένων μοντέλων σε μια πραγματική έξυπνη συσκευή Android με στόχο τη μέτρηση του χρόνου απόκρισής τους.

#### 3.1 Android

Το *Android* αποτελεί λειτουργικό σύστημα ανοικτού κώδικα, το οποίο είναι βασισμένο στο Linux και χρησιμοποιείται για φορητές συσκευές, όπως τα κινητά τηλέφωνα. Αναπτύχθηκε από την Open Handset Alliance, η οποία αποτελεί μία επιχειρηματική συνεργασία διαφόρων εταιρειών, με επικεφαλής την Google και άλλες εταιρείες. Η πρώτη δοκιμαστική έκδοση του Android Software Development Kit (SDK) κυκλοφόρησε από την Google το 2007, ενώ η πρώτη εμπορική έκδοση, το Android 1.0, κυκλοφόρησε τον Σεπτέμβριο του 2008.

Το Android προσφέρει μια ενιαία προσέγγιση στην ανάπτυξη εφαρμογών για κινητές συσκευές. Αυτό σημαίνει ότι οι προγραμματιστές χρειάζεται να αναπτύσσουν εφαρμογές μόνο για το Android και αυτές οι εφαρμογές μπορούν να “τρέχουν” σε διαφορετικές συσκευές που λειτουργούν με αυτό [50].

Μία εφαρμογή που αναπτύσσεται για συσκευές που λειτουργούν με το παραπάνω λειτουργικό σύστημα έχει τα εξής κύρια δομικά στοιχεία:

1. Δραστηριότητες: Μια *δραστηριότητα* είναι το σημείο εισόδου μίας εφαρμογής για αλληλεπίδραση με τον χρήστη και αντιπροσωπεύει μια ενιαία οθόνη. Μπορεί να υπάρχουν παραπάνω από μία δραστηριότητες σε μία εφαρμογή, οι οποίες εξυπηρετούν διαφορετικό σκοπό. Για παράδειγμα, μια εφαρμογή ηλεκτρονικού ταχυδρομείου μπορεί να έχει δραστηριότητες για την εμφάνιση μιας λίστας νέων μηνυμάτων ηλεκτρονικού ταχυδρομείου, για τη σύνταξη ενός μηνύματος και για την ανάγνωση μηνυμάτων. Οι δραστηριότητες μιας εφαρμογής είναι ανεξάρτητες μεταξύ τους και έχουν συγκεκριμένο “κύκλο ζωής”.
2. Υπηρεσίες: Μια *υπηρεσία* είναι ένα σημείο εισόδου γενικής χρήσης για τη διατήρηση μιας εφαρμογής σε λειτουργία στο παρασκήνιο, με σκοπό την εκτέλεση μακροχρόνιων λειτουργιών ή την εκτέλεση εργασιών για απομακρυσμένες διεργασίες. Μία υπηρεσία δεν παρέχει διεπαφή για τον χρήστη.

3. Δέκτες Εκπομπής: Ο δέκτης εκπομπής είναι ένα στοιχείο που επιτρέπει στο σύστημα να παραδίδει συμβάντα στην εφαρμογή, εκτός της κανονικής ροής του χρήστη, επιτρέποντας στην εφαρμογή να ανταποκρίνεται σε ανακοινώσεις εκπομπής από όλο το σύστημα.
4. Πάροχοι Περιεχομένου: Ένας πάροχος περιεχομένου διαχειρίζεται ένα κοινόχρηστο σύνολο δεδομένων της εφαρμογής, τα οποία μπορεί να είναι αποθηκευμένα σε ένα σύστημα αρχείων, σε μια βάση δεδομένων SQLite, στον ιστό ή σε οποιαδήποτε άλλη μόνιμη θέση αποθήκευσης στην οποία μπορεί να έχει πρόσβαση η εφαρμογή. Μέσω του παρόχου περιεχομένου, άλλες εφαρμογές μπορούν να ζητούν ή να τροποποιούν τα δεδομένα, εάν ο πάροχος περιεχομένου το επιτρέπει [51].

## 3.2 Android Studio

Το *Android Studio* είναι το επίσημο Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE) για την ανάπτυξη εφαρμογών Android. Βασίζεται στον ισχυρό επεξεργαστή κώδικα και στα εργαλεία ανάπτυξης της IntelliJ IDEA [57]. Οι γλώσσες προγραμματισμού που χρησιμοποιούνται για την ανάπτυξη εφαρμογών στο Android Studio είναι η Kotlin και η Java.

## 3.3 Java

Η *Java* είναι μια ευρέως χρησιμοποιούμενη αντικειμενοστραφής γλώσσα προγραμματισμού και πλατφόρμα λογισμικού που εκτελείται σε δισεκατομμύρια συσκευές, όπως φορητούς υπολογιστές, κινητές συσκευές, κονσόλες παιχνιδιών, ιατρικές συσκευές και πολλές άλλες. Οι κανόνες και το συντακτικό της Java βασίζονται στις γλώσσες C και C++.

Ένα σημαντικό πλεονέκτημα της ανάπτυξης λογισμικού με τη Java είναι η “φορητότητά” της. Για παράδειγμα, ο κώδικας για ένα πρόγραμμα Java σε έναν υπολογιστή είναι πολύ εύκολο να μεταφερθεί σε μια κινητή συσκευή. Όταν η γλώσσα επινοήθηκε το 1991 από τον James Gosling της Sun Microsystems, ο πρωταρχικός στόχος ήταν να μπορεί κανείς να γράψει μία φορά ένα πρόγραμμα και αυτό να μπορεί να “τρέξει” οπουδήποτε (“write once, run everywhere”).

Σε ένα γενικότερο πλαίσιο, η Java είναι μια τεχνολογία που αποτελείται τόσο από μια γλώσσα προγραμματισμού, όσο και από μια πλατφόρμα λογισμικού. Η πλατφόρμα λογισμικού της Java αποτελείται από την Εικονική Μηχανή Java (Java Virtual Machine - JVM), τη Διεπαφή

Προγραμματισμού Εφαρμογών της Java (Java API) και ένα πλήρες περιβάλλον ανάπτυξης. Ένα πρόγραμμα στη γλώσσα προγραμματισμού Java, κωδικοποιείται από έναν μεταγλωττιστή σε μία ακολουθία από byte. Η JVM αναλύει και εκτελεί (“ερμηνεύει”) τον παραπάνω κώδικα. Η Διεπαφή Προγραμματισμού της Java αποτελείται από ένα εκτεταμένο σύνολο βιβλιοθηκών, που περιλαμβάνουν βασικά αντικείμενα, λειτουργίες δικτύωσης και ασφάλειας, παραγωγή XML (Extensible Markup Language) και υπηρεσίες ιστού. Σε συνδυασμό, η γλώσσα Java και η πλατφόρμα λογισμικού Java δημιουργούν μια ισχυρή τεχνολογία για την ανάπτυξη λογισμικού [52].

## 3.4 Python

Η Python είναι μια διερμηνεύσιμη, αντικειμενοστραφής, υψηλού επιπέδου γλώσσα προγραμματισμού, με δυναμική σημασιολογία. Οι ενσωματωμένες δομές δεδομένων υψηλού επιπέδου, σε συνδυασμό με τη δυναμική τυποποίηση (dynamic typing) και τη δυναμική δέσμευση (dynamic binding), την καθιστούν πολύ ελκυστική για την ταχεία ανάπτυξη εφαρμογών, καθώς και για χρήση ως γλώσσα σεναρίων ή ως γλώσσα για τη σύνδεση υφιστάμενων στοιχείων μεταξύ τους. Το απλό, εύχρηστο συντακτικό της Python δίνει έμφαση στην αναγνωσιμότητα και συνεπώς μειώνει το κόστος συντήρησης των προγραμμάτων. Επίσης, υποστηρίζει ενότητες και πακέτα, γεγονός που ενθαρρύνει την αρθρωσιμότητα του προγράμματος και την επαναχρησιμοποίηση του κώδικα. Ο διερμηνέας της Python και η εκτεταμένη πρότυπη βιβλιοθήκη είναι διαθέσιμα σε πηγαία ή δυαδική μορφή χωρίς χρέωση, για όλες τις κύριες πλατφόρμες και μπορούν να διανεμηθούν ελεύθερα [71].

### 3.4.1 Keras

Το *Keras* είναι μια βιβλιοθήκη νευρωνικών δικτύων υψηλού επιπέδου και ανοικτού κώδικα, η οποία είναι γραμμένη στη γλώσσα προγραμματισμού Python. Αναπτύχθηκε από έναν μηχανικό της Google, τον Francois Chollet. Είναι φιλική προς το χρήστη, επεκτάσιμη και αρθρωτή για τη διευκόλυνση του ταχύτερου πειραματισμού με βαθιά νευρωνικά δίκτυα. Υποστηρίζει, τα συνελκτικά νευρωνικά δίκτυα, τα επαναληπτικά νευρωνικά δίκτυα, αλλά και συνδυασμούς αυτών.

Το Keras, δεν μπορεί να χειριστεί υπολογισμούς χαμηλού επιπέδου, όπως είναι τα γινόμενα τανυστών ή οι συνελίξεις, οπότε χρησιμοποιεί μία εξειδικευμένη βιβλιοθήκη διαχείρισης τανυστών που είναι βελτιστοποιημένη για να χρησιμεύσει ως μηχανή στο “παρασκήνιο” (backend). Η βιβλιοθήκη αυτή λειτουργεί ως μία διεπαφή – “περιτύλιγμα” υψηλού επιπέδου για τη διεπαφή χαμηλού επιπέδου. Το Keras αποτελείται από τρεις τέτοιες μηχανές, οι οποίες είναι οι εξής:

1. **TensorFlow**: Το TensorFlow είναι ένα προϊόν της Google, και είναι ένα από τα πιο διάσημα εργαλεία βαθιάς μάθησης που χρησιμοποιούνται ευρέως στον ερευνητικό τομέα της μηχανικής μάθησης και των βαθιών νευρωνικών δικτύων. Κυκλοφόρησε στην αγορά στις 9 Νοεμβρίου 2015. Είναι κατασκευασμένο με τέτοιο τρόπο ώστε να μπορεί εύκολα να εκτελεστεί σε πολλαπλές CPU και GPU, καθώς και σε κινητά λειτουργικά συστήματα.
2. **Theano**: Το Theano είναι μια βιβλιοθήκη της Python ανοικτού κώδικα που χρησιμοποιείται ευρέως για την εκτέλεση μαθηματικών πράξεων σε πολυδιάστατους πίνακες ενσωματώνοντας τις βιβλιοθήκες `scipy` και `numpy`. Χρησιμοποιεί GPU για ταχύτερους υπολογισμούς και υπολογίζει αποτελεσματικά τις κλίσεις κατασκευάζοντας συμβολικά γραφήματα αυτόματα.
3. **CNTK**: Το Microsoft Cognitive Toolkit είναι το πλαίσιο ανοικτού κώδικα της βαθιάς μάθησης. Αποτελείται από όλα τα βασικά δομικά στοιχεία, τα οποία απαιτούνται για τον σχηματισμό ενός νευρωνικού δικτύου [53].

## 3.5 TensorFlow Lite

Στη παρούσα διπλωματική εργασία, για να γίνει χρήση της βιβλιοθήκης Keras, θα χρησιμοποιήσουμε σαν backend – βιβλιοθήκη την TensorFlow, η οποία παρουσιάστηκε παραπάνω. Για την ενσωμάτωση των μοντέλων σε μία κινητή συσκευή, θα χρησιμοποιηθεί το εργαλείο *TensorFlow Lite*. Το TensorFlow Lite αποτελεί ένα περιβάλλον βαθιάς μάθησης για κινητές συσκευές, που αναπτύχθηκε από την Google. Μόνο ένα μοντέλο που έχει αναπτυχθεί με τη βιβλιοθήκη TensorFlow μπορεί να βελτιστοποιηθεί με το εργαλείο αυτό. Τα δύο κύρια συστατικά του είναι ένας *μετατροπέας* και ένας *διερμηνέας*. Ο μετατροπέας χρησιμεύει ως μονάδα βελτιστοποίησης για τη διατήρηση της απόδοσης στο μοντέλο, ώστε να μπορεί αυτό να χρησιμοποιηθεί αποτελεσματικά. Η χρήση του διερμηνέα είναι να βοηθήσει στην εκτέλεση του βελτιστοποιημένου μοντέλου σε μία κινητή συσκευή.

Ένα από τα σημαντικότερα χαρακτηριστικά του TensorFlow Lite είναι ότι ο μετατροπέας του υποστηρίζει ένα σύνολο από *μεθόδους κβαντοποίησης χωρίς επανεκπαίδευση* κατά την μετατροπή ενός μοντέλου. Αυτές οι μέθοδοι μπορούν να μετατρέψουν τα αρχικά 32-bit βάρη του μοντέλου σε αριθμούς κινητής υποδιαστολής των 16 bit ή ακεραίους των 8 bit, για παράδειγμα. Όπως αναφέρθηκε και στο **Κεφάλαιο 2**, αυτό μπορεί να είναι πολύ χρήσιμο σε περιπτώσεις όπου η επανεκπαίδευση του μοντέλου μπορεί να είναι υπολογιστικά υπερβολικά κοστοβόρα (ακόμη και για λίγες εποχές) ή σε περιπτώσεις όπου δεν έχουμε πρόσβαση στο αρχικό σύνολο εκπαίδευσης [55].

## 3.6 ImageNet

Το ImageNet είναι μια βάση δεδομένων ταξινόμησης εικόνων μεγάλης κλίμακας που περιέχει πάνω από 14 εκατομμύρια εικόνες με περισσότερες από 20.000 κατηγορίες ή κλάσεις. Δημιουργήθηκε από μια ομάδα ερευνητών στο Πανεπιστήμιο του Πρίνστον το 2009 και έκτοτε έχει γίνει ένα τυπικό σύνολο δεδομένων αναφοράς για την αξιολόγηση αλγορίθμων ταξινόμησης εικόνων. Χρησιμοποιήθηκε για την εκπαίδευση και αξιολόγηση των νικητήριων μοντέλων στο ετήσιο ImageNet Large Scale Visual Recognition Challenge (ILSVRC), έναν διαγωνισμό όπου ομάδες από όλο τον κόσμο ανταγωνίζονται για την ανάπτυξη των καλύτερων μοντέλων ταξινόμησης εικόνων.

Το ImageNet έχει διαδραματίσει σημαντικό ρόλο στην προώθηση του τομέα της όρασης υπολογιστών και της μηχανικής μάθησης, καθώς έδωσε τη δυνατότητα στους ερευνητές να αναπτύξουν και να αξιολογήσουν όλο και πιο εξελιγμένα μοντέλα ταξινόμησης εικόνων. Τα τελευταία χρόνια, το ImageNet έχει χρησιμοποιηθεί ως σύνολο δεδομένων αναφοράς για την αξιολόγηση και άλλων διεργασιών, όπως η ανίχνευση αντικειμένων, η σημασιολογική κατάτμηση και η δημιουργία λεζάντας εικόνων. Η διαθεσιμότητα συνόλων δεδομένων μεγάλης κλίμακας όπως το ImageNet βοήθησε στην επιτάχυνση της έρευνας στην όραση υπολογιστών και συνέβαλε στην ανάπτυξη πραγματικών εφαρμογών όπως είναι τα αυτόνομα οχήματα και η ανάλυση ιατρικών εικόνων [72].

## 4. Περιβάλλον Αξιολόγησης

Παρακάτω περιγράφεται το περιβάλλον αξιολόγησης των μεθόδων συμπίεσης, οι οποίες παρουσιάστηκαν στο **Κεφάλαιο 2**. Γίνεται αναφορά στα σύνολα δεδομένων τα οποία χρησιμοποιούνται, στις αρχιτεκτονικές των μοντέλων, στα χαρακτηριστικά της κινητής συσκευής μέσω της οποίας γίνονται οι μετρήσεις, στις μετρικές που χρησιμοποιούνται και τέλος στους αλγορίθμους συμπίεσης, από κάθε κατηγορία μεθόδων, που εφαρμόστηκαν στα μοντέλα.

### 4.1 Σύνολα Δεδομένων

Τα μοντέλα που χρησιμοποιήθηκαν για την πραγματοποίηση του πειραματικού μέρους της διπλωματικής εργασίας αντλήθηκαν από τη βιβλιοθήκη *applications* του Keras, προεκπαιδευμένα στο σύνολο δεδομένων *ImageNet-1k*. Το εν λόγω σύνολο δεδομένων είναι ένα υποσύνολο του μεγαλύτερου συνόλου δεδομένων ImageNet (**Ενότητα 3.6**) που αποτελείται από 1.000 κατηγορίες αντικειμένων με συνολικά πάνω από 1.2 εκατομμύρια εικόνες. Οι εικόνες στο σύνολο δεδομένων είναι διαφορετικών μεγεθών και αναλύσεων (μέση ανάλυση 256x256) και περιλαμβάνουν ένα ευρύ φάσμα αντικειμένων, όπως ζώα, οχήματα, έπιπλα και είδη οικιακής χρήσης. Χρησιμοποιήθηκε στον διαγωνισμό ILSVRC του 2012, που ήταν η πρώτη έκδοση του ετήσιου διαγωνισμού που είχε ως στόχο να ωθήσει τα όρια της απόδοσης ταξινόμησης εικόνων. Το ImageNet-1k είναι ένα αρκετά απαιτητικό σύνολο δεδομένων λόγω του μεγάλου αριθμού κλάσεων και της μεταβλητότητας των εικόνων σε κάθε κλάση, για αυτό και σήμερα θεωρείται ίσως το κορυφαίο σύνολο δεδομένων αναφοράς για την αξιολόγηση αλγορίθμων ταξινόμησης εικόνων, ιδιαίτερα βαθιών νευρωνικών δικτύων.

Ωστόσο, όπως έχει ήδη αναφερθεί, η εφαρμογή μεθόδων συμπίεσης συνήθως απαιτεί την επανεκπαίδευση του μοντέλου, ώστε να ανακτηθεί η χαμένη ακρίβεια και η εκπαίδευση ενός μοντέλου ταξινόμησης στο ImageNet-1k είναι μία αρκετά χρονοβόρα διαδικασία, λόγω του μεγάλο όγκου του συνόλου δεδομένων. Για αυτό το λόγο, χρησιμοποιήθηκε η τεχνική της μεταφοράς γνώσης στα προαναφερθέντα μοντέλα με στόχο αρχικά την προσαρμογή τους σε ένα μικρότερο σύνολο δεδομένων, το *CIFAR10*.

Το σύνολο δεδομένων CIFAR10 είναι μια συλλογή από 60.000 έγχρωμες εικόνες διαστάσεων 32x32 ταξινομημένες σε 10 κλάσεις (6.000 εικόνες ανά κλάση). Οι 10 κατηγορίες είναι οι: αεροπλάνο, αυτοκίνητο, πουλί, γάτα, ελάφι, σκύλος, βάτραχος, άλογο, πλοίο και φορτηγό. Μπορεί κανείς να παρατηρήσει ότι οι διαστάσεις των εικόνων στο CIFAR10 είναι αρκετά μικρές σε σύγκριση με το ImageNet, γεγονός που τις καθιστά πιο κατάλληλες για πειραματισμό. Αυτό βοηθά τον σκοπό της συγκεκριμένης διπλωματικής εργασίας.



### 4.1.1 Μεταφορά Γνώσης – Διαδικασία

Στη δεδομένη περίπτωση, έχουμε επαγωγική μεταφορά γνώσης μεταξύ των δοσμένων και των καινούριων μοντέλων, αφού το καινούριο πρόβλημα προς επίλυση (ταξινόμηση σε 10 κλάσεις) είναι διαφορετικό από το παλιό (ταξινόμηση σε 1000 κλάσεις).

Η διαδικασία που ακολουθήθηκε για κάθε μοντέλο περιγράφεται παρακάτω:

1. Κατασκευή μοντέλου: Το νέο μοντέλο έχει την ίδια αρχιτεκτονική με το αρχικό σε ό,τι αφορά τον εξαγωγέα χαρακτηριστικών, όμως διαφορετικό ταξινομητή, ώστε η έξοδος του να αντιστοιχεί σε 10 αντί για 1000 κλάσεις.
2. Αρχικοποίηση παραμέτρων: Οι παράμετροι του εξαγωγέα χαρακτηριστικών του αρχικού μοντέλου αντιγράφηκαν στα αντίστοιχα στρώματα του νέου μοντέλου, ενώ τα βάρη του ταξινομητή του νέου μοντέλου αρχικοποιήθηκαν τυχαία με βάση την ομοιόμορφη κατανομή σε διάστημα που εξαρτάται από το πλήθος των νευρώνων εισόδου και εξόδου και οι μεροληψίες αρχικοποιήθηκαν με μηδενικές τιμές.
3. Εκπαίδευση: Προσαρμογή του μοντέλου στο νέο σύνολο δεδομένων CIFAR10 αφήνοντας όλα τα στρώματα του δικτύου να μεταβάλλουν τα βάρη τους και χρησιμοποιώντας τον ίδιο ρυθμό μάθησης για όλο το δίκτυο.

Με βάση τα παραπάνω, μπορεί κανείς να καταλάβει πως ο τρόπος με τον οποίο εφαρμόστηκε η μεταφορά γνώσης είναι η μεταφορά παραμέτρων από τα αρχικά στα νέα μοντέλα, αφού τα αντίστοιχα προβλήματα προς επίλυση είναι συσχετιζόμενα. Σε ορισμένες περιπτώσεις μοντέλων με μέγεθος εισόδου μεγαλύτερο από 224x224, το μέγεθος αυτό μειώθηκε σε 224x224, μιας και οι εικόνες του CIFAR10 είναι πολύ μικρές, αλλά και για να μελετηθεί η επίδραση αυτής της μεταβολής στην τελική ακρίβεια.

## 4.2 Αρχιτεκτονικές Μοντέλων Βαθιάς Μάθησης

Για την αξιολόγηση των μεθόδων συμπίεσης βαθιών νευρωνικών δικτύων χρησιμοποιήθηκαν συνελκτικά μοντέλα διαφορετικών αρχιτεκτονικών, τα οποία επιλύουν το πρόβλημα της ταξινόμησης εικόνων. Οι αρχιτεκτονικές αυτές αναγράφονται παρακάτω:

- Inception: Η αρχιτεκτονική Inception δημοσιεύθηκε για πρώτη φορά το Σεπτέμβριο του 2014 και είχε ως στόχο την κατασκευή μοντέλων βαθιάς μάθησης με ακόμα περισσότερα βαθιά στρώματα και ταυτόχρονα την εύρεση αποδοτικότερων μεθόδων υπολογισμού των χαρτών ενεργοποίησης στα κρυφά επίπεδα. Ο δομικός λίθος των μοντέλων που μοιράζονται την

αρχιτεκτονική Inception ονομάζεται “Inception-μονάδα”, οι οποίες δημιουργούν μια αρχιτεκτονική στοίβας. Η μονάδα αυτή περιλαμβάνει μείωση διαστάσεων με 1x1 συνέλιξη και 3x3 και 5x5 συνέλιξεις σε συνδυασμό με προβολή [66].

- **ResNet:** Η αρχιτεκτονική ResNet δημοσιεύθηκε το Δεκέμβριο του 2015 και ήταν αυτή που εισήγαγε την έννοια των υπολοίπων στα μοντέλα βαθιάς μάθησης. Τα υπόλοιπα δημιουργήθηκαν ως ένας τρόπος για τη βελτίωση της εκπαίδευσης, όσο αυξάνονταν τα επίπεδα και προχωρούσαμε σε πιο “βαθιές” αρχιτεκτονικές. Η βασική ιδέα πίσω από την έννοια αυτή είναι η εφαρμογή ενός επιπέδου σε συνδυασμό με ένα ταυτοτικό επίπεδο σε ένα σύνολο χαρτών ενεργοποίησης. Ο συνδυασμός των αποτελεσμάτων γίνεται συνήθως με την εκτέλεση πρόσθεσης ανάμεσα στους παραγόμενους χάρτες ενεργοποίησης από κάθε επίπεδο [63].
- **DenseNet:** Η αρχιτεκτονική DenseNet δημοσιεύθηκε για πρώτη φορά τον Ιανουάριο 2016 και βασίζεται στην ιδέα των υπολοίπων που εισήγαγαν τα ResNet. Στην αρχιτεκτονική αυτή κάθε στρώμα συνδέεται με κάθε άλλο στρώμα με εμπρόσθια τροφοδότηση. Ενώ τα παραδοσιακά συνελκτικά δίκτυα με  $L$  στρώματα έχουν  $L$  συνδέσεις - μία μεταξύ κάθε στρώματος και των επόμενων στρωμάτων ένα μοντέλο με αρχιτεκτονική DenseNet έχει  $\frac{L(L+1)}{2}$  άμεσες συνδέσεις [64].
- **InceptionResNet:** Η αρχιτεκτονική InceptionResNet δημοσιεύθηκε τον Αύγουστο του 2016 και αποτελείται από μία δομική μονάδα η οποία συνδυάζει την έννοια των υπολοίπων με τη δομική μονάδα του Inception. Η αρχιτεκτονική σχηματίζεται από μία στοίβα, η οποία έχει ως δομικούς λίθους τις μονάδες αυτές [67].
- **MobileNet:** Η αρχιτεκτονική MobileNet δημοσιεύθηκε για πρώτη φορά τον Απρίλιο του 2017 και είχε ως σκοπό την ενσωμάτωση των μοντέλων βαθιάς μάθησης σε συστήματα κινητών τηλεφώνων και ενσωματωμένα συστήματα. Τα μοντέλα με την αρχιτεκτονική MobileNet βασίζονται σε μια βελτιωμένη αρχιτεκτονική που χρησιμοποιεί κατά βάθος διαχωρίσιμες συνέλιξεις για την κατασκευή ελαφριών μοντέλων βαθιών νευρωνικών δικτύων [65].
- **NASNet:** Η αρχιτεκτονική NASNet δημοσιεύθηκε τον Ιούλιο του 2017 και αναπτύχθηκε με βάση τον σχεδιασμό ενός νέου χώρου αναζήτησης (ο οποίος αποκαλείται “χώρος αναζήτησης NASNet”), ο οποίος επιτρέπει τη δυνατότητα μεταφοράς γνώσης. Μέσω μίας πειραματικής διαδικασίας έγινε αναζήτηση για το καλύτερο επίπεδο συνελκτικού δικτύου (ή “κύτταρο”) στο σύνολο δεδομένων CIFAR10 και στη συνέχεια αυτή η μονάδα εφαρμόστηκε στο σύνολο δεδομένων ImageNet συσσωρεύοντας περισσότερα αντίγραφα αυτής, το καθένα με τις δικές του παραμέτρους για να σχεδιαστεί μια συνελκτική αρχιτεκτονική, την οποία ονομάζουμε αρχιτεκτονική NASNet [68].
- **EfficientNet:** Η αρχιτεκτονική EfficientNet δημοσιεύθηκε για πρώτη φορά τον Μάιο του 2019 και βασίζεται σε μια νέα τεχνική κλιμάκωσης που ρυθμίζει ομοιόμορφα όλες τις διαστάσεις - βάθος/πλάτος/ανάλυση- χρησιμοποιώντας έναν απλό, αλλά εξαιρετικά αποτελεσματικό, μεικτό

συντελεστή. Χρησιμοποιήθηκε η αναζήτηση αρχιτεκτονικής (NAS) για να σχεδιαστεί ένα νέο βασικό μοντέλο δικτύου και να επεκταθεί ώστε να δημιουργηθεί μια οικογένεια μοντέλων, τα λεγόμενα EfficientNet, τα οποία επιτυγχάνουν πολύ υψηλότερη ακρίβεια και αποδοτικότητα από τα προηγούμενα συνελικτικά δίκτυα [69].

Από τις παραπάνω αρχιτεκτονικές επιλέχθηκε ένας αριθμός μοντέλων, τα οποία αναγράφονται στον Πίνακα 1. Στις στήλες του Πίνακα αναφέρονται τα βασικά χαρακτηριστικά κάθε μοντέλου: το μέγεθος εισόδου, η ακρίβεια, ο αριθμός παραμέτρων και το πλήθος των πράξεων κινητής υποδιαστολής, για κάθε σύνολο δεδομένων.

Μοντέλα	Μέγεθος Εισόδου		Ακρίβεια (%)		Αριθμός Παραμέτρων (10 <sup>6</sup> )		GFLOPs	
	ImageNet	CIFAR10	ImageNet	CIFAR10	ImageNet	CIFAR10	ImageNet	CIFAR10
MobileNetV2	224	224	71.3	94.55	3.5	2.22	0.602	0.599
InceptionV3	299	224	77.9	96.10	23.9	21.79	11.426	5.671
DenseNet121	224	224	75.0	94.55	8.1	6.89	5.668	5.666
DenseNet169	224	224	76.2	96.57	14.3	12.36	6.719	6.716
DenseNet201	224	224	77.3	96.56	20.2	17.90	8.582	8.578
EfficientNetB0	224	224	77.1	95.13	5.3	4.00	0.771	0.769
EfficientNetB2	260	224	80.1	95.87	9.2	7.68	1.987	1.315
EfficientNetB4	380	380	82.9	94.97	19.5	17.50	8.787	8.784
EfficientNetB6	528	528	84.0	96.47	43.3	40.65	38.135	38.130
EfficientNetLiteB0	224	-	75.1	-	4.7	-	0.770	-
EfficientNetLiteB2	260	-	77.6	-	6.1	-	1.719	-
EfficientNetLiteB4	300	-	81.5	-	13.0	-	5.106	-
InceptionResNetV2	299	299	80.3	94.88	55.9	54.29	26.311	26.308
NASNetMobile	224	224	74.4	95.19	5.3	4.24	1.127	1.125
NASNetLarge	331	331	82.5	95.96	88.9	84.74	47.567	47.559
ResNet101	224	224	76.4	93.89	44.7	42.47	15.140	15.136
ResNet152	224	224	76.6	92.59	60.4	58.09	22.564	22.560
ResNet50	224	224	74.9	92.50	25.6	23.50	7.715	7.711
ResNet152V2	224	224	78.0	91.03	60.4	58.09	21.813	21.809
ResNet101V2	224	224	77.2	91.33	44.7	42.47	14.389	14.384
ResNet50V2	224	224	76.0	91.85	25.6	23.50	6.964	6.960
VGG16	224	224	71.3	85.72	138.4	14.72	30.940	4.923
VGG19	224	224	71.3	87.93	143.7	20.03	39.264	4.657

Xception	299	224	79.0	96.53	22.9	20.82	16.714	9.101
----------	-----	-----	------	-------	------	-------	--------	-------

Πίνακας 1: Χαρακτηριστικά Μοντέλων Βαθιάς Μάθησης

### Παρατηρήσεις:

- Οι διαφορές που παρατηρούνται στα πλήθη των παραμέτρων των μοντέλων στα δύο σύνολα δεδομένων οφείλονται στους διαφορετικούς ταξινομητές αυτών και συγκεκριμένα στο πλήθος των νευρώνων εξόδου (1000 και 10). Αυτή η διαφορά οδηγεί συνήθως σε απόκλιση 1 με 2 εκατομμύρια παραμέτρων ανάλογα με το πλήθος των χαρακτηριστικών που παράγει κάθε μοντέλο. Από αυτή τη λογική βλέπουμε ότι απέχουν τα δύο VGG μοντέλα, καθώς:
  - ➔ Τα μοντέλα VGG16 και VGG19 που είναι εκπαιδευμένα στο ImageNet περιλαμβάνουν τέσσερα πλήρως συνδεδεμένα επίπεδα, όπου το τελευταίο είναι το στρώμα εξόδου, με 25.088, 4096, 4096 και 1000 νευρώνες το καθένα. Επομένως, έχουμε  $(25.088 \times 4096) + (4096 \times 4096) + (4096 \times 1000) = 123.637.760$  βάρη και  $4096 + 4096 + 1000 = 9192$  μεροληψίες. Άρα, συνολικά **123.646.952** παραμέτρους οι οποίες αφορούν στα πλήρως συνδεδεμένα επίπεδα. Στα αντίστοιχα μοντέλα που είναι εκπαιδευμένα στο CIFAR10, τα τρία παραπάνω στρώματα αντικαθίστανται με δύο πλήρως συνδεδεμένα με 512 και 10 νευρώνες στο καθένα, άρα συνολικά  $512 \times 10 + 10 = 5.130$  παραμέτρους στα πλήρως συνδεδεμένα στρώματα.
- Οι πράξεις κινητής υποδιαστολής, στην περίπτωση των μοντέλων VGG, μεταβάλλονται αναλόγως με το πλήθος των παραμέτρων, αφού η διαφορά των μοντέλων που είναι εκπαιδευμένα στο ImageNet με αυτά που είναι εκπαιδευμένα στο CIFAR10 έγκειται στο πλήθος των πλήρως συνδεδεμένων επιπέδων και στο πλήθος των νευρώνων που βρίσκονται σε αυτά, τα οποία στη δεύτερη περίπτωση ελαττώνονται και τα δύο αισθητά.
- Από τον παραπάνω πίνακα, μπορεί να παρατηρήσει κανείς πως το πλήθος των παραμέτρων και το πλήθος πράξεων κινητής υποδιαστολής ενός μοντέλου δεν είναι κατά κανόνα ανάλογα μεγέθη. Ένα παράδειγμα αποτελούν τα μοντέλα InceptionV3 και ResNet50V2, τα οποία ενώ έχουν περίπου τον ίδιο αριθμό παραμέτρων, διαφέρουν αισθητά στις πράξεις κινητής υποδιαστολής, λόγω διαφορών στην αρχιτεκτονική τους.
- Όπως μπορεί να παρατηρήσει κανείς, δεν έχει γίνει εκπαίδευση με μεταφορά γνώσης στα μοντέλα EfficientNetLite στο σύνολο δεδομένων CIFAR10. Αυτό συμβαίνει αφού τα βασικά μέρη της αρχιτεκτονικής τους είναι ίδια με αυτά των EfficientNet, με μόνες διαφορές ότι για τα πρώτα χρησιμοποιείται η συνάρτηση ενεργοποίησης ReLU6 και ότι αφαιρείται ένας συγκεκριμένος τύπος δομικού στοιχείου (building block), μετατρέποντάς τα σε καταλληλότερα για ενσωμάτωση σε κινητές συσκευές.

## 4.3 Μετρικές

Οι μετρήσεις που πάρθηκαν, κατά τη διάρκεια του πειραματικού μέρους της διπλωματικής, αφορούν τις παρακάτω μετρικές:

- Ακρίβεια: Το ποσοστό των δειγμάτων για τα οποία η πρόβλεψη ενός μοντέλου είναι ορθή. Για τα μοντέλα που είναι εκπαιδευμένα στο σύνολο δεδομένων ImageNet αυτή καθορίζεται από το σύνολο αξιολόγησης (validation set) του συνόλου, ενώ για τα μοντέλα που επανεκπαιδεύτηκαν με μεταφορά γνώσης στο σύνολο δεδομένων CIFAR10, καθορίζεται από το σύνολο ελέγχου (test set) του συνόλου.
- Μέγεθος: Ο χώρος που καταλαμβάνει ένα μοντέλο σε μία συσκευή ή στη μνήμη μιας συσκευής.
- Αριθμός πράξεων κινητής υποδιαστολής (FLOPs): Το πλήθος πράξεων (ή βημάτων) που εκτελούνται κατά τη συμπερασματολογία ενός μοντέλου για μία δεδομένη είσοδο. Αποτελεί καθαρό μέγεθος.
- Χρόνος απόκρισης: Ο χρόνος που χρειάζεται για να πραγματοποιηθεί η διαδικασία της συμπερασματολογίας ενός μοντέλου για μία δεδομένη είσοδο. Μετρείται σε χιλιοστά του δευτερολέπτου (ms).
  - ➔ Ο χρόνος απόκρισης κατά τη συμπερασματολογία ενός μοντέλου σε μια κινητή συσκευή στο πλαίσιο μιας εφαρμογής δεν είναι σταθερός, αλλά παρουσιάζει διακυμάνσεις καθώς επηρεάζεται από το δυναμικό περιβάλλον της συσκευής, δηλαδή τη θερμοκρασία της, τον στιγμιαίο φόρτο του επεξεργαστή, τη διαθέσιμη μνήμη, την μπαταρία και τις άλλες λειτουργίες που επιτελεί ταυτόχρονα το λογισμικό. Για αυτό το λόγο, ο χρόνος απόκρισης θεωρείται στοχαστικό μέγεθος και περιγράφεται από μια τυχαία μεταβλητή. Επομένως, έχει παρθεί ένας ορισμένος αριθμός μετρήσεων, από τις οποίες εξάγουμε τα εξής στατιστικά μεγέθη:
    - Ελάχιστη τιμή
    - Μέγιστη τιμή
    - Μέση τιμή
    - Τυπική απόκλιση
    - 90° Εκατοστημόριο (το 90% του πλήθους των μετρήσεων είναι μικρότερες από αυτήν την τιμή)

## 4.4 Κινητή Συσκευή

Οι μετρήσεις για το χρόνο απόκρισης κατά τη συμπεραματολογία των μοντέλων που έχουν περιγραφεί στην **Ενότητα 4.2**, έχουν παρθεί για την κινητή συσκευή Samsung Galaxy S20 FE, της οποίας τα χαρακτηριστικά φαίνονται στον παρακάτω πίνακα:

Κυκλοφορία	2 Οκτωβρίου, 2020
System on a Chip	Exynos 990 (7 nm+)
CPU	Octa-core (2x2.73 GHz Mongoose M5 & 2x2.50 GHz Cortex-A76 & 4x2.0 GHz Cortex-A55)
GPU	Mali-G77 MP11
RAM	6GB
Έκδοση Android	11

Πίνακας 2: Χαρακτηριστικά Κινητής Συσκευής

Αν και όταν κυκλοφόρησε κατατασσόταν στην οικογένεια των high-end κινητών συσκευών, σήμερα, λόγω παλαιότητας, η συγκεκριμένη συσκευή μπορεί να θεωρηθεί ότι αντιπροσωπεύει πολύ καλά την οικογένεια των mid-range προς high-end συσκευών. Ο χρόνος απόκρισης ενός μοντέλου κατά τη συμπεραματολογία, μετρήθηκε για κάθε μία από τις παρακάτω μονάδες επεξεργασίας της συσκευής:

1. CPU (στις περιπτώσεις με 1 και 2 νήματα – threads)
2. GPU
3. NPU

Η Μονάδα Επεξεργασίας Νευρωνικών Δικτύων (NPU) αποτελεί μία μονάδα επεξεργασίας η οποία έχει σχεδιαστεί για να επιταχύνει τους υπολογισμούς που εκτελούνται για τις διεργασίες μηχανικής μάθησης. Η NPU περιλαμβάνει μονάδες για πολλαπλασιασμό και πρόσθεση, λειτουργίες ενεργοποίησης, λειτουργίες δεδομένων 2D και αποσυμπίεση. Η μονάδα πολλαπλασιασμού και πρόσθεσης χρησιμοποιείται για τον υπολογισμό του πολλαπλασιασμού και της πρόσθεσης πινάκων, της συνέλιξης, του τετραγωνικού γινομένου και άλλων λειτουργιών [70].

Για την αξιοποίηση της NPU χρησιμοποιείται η Διεπαφή Νευρωνικών Δικτύων (NNAPI) του Android. Με βάση τις απαιτήσεις μιας εφαρμογής και τις δυνατότητες του υλικού μιας συσκευής Android, το NNAPI μπορεί να κατανέμει αποτελεσματικά τον υπολογιστικό φόρτο εργασίας στους διαθέσιμους επιταχυντές της συσκευής, συμπεριλαμβανομένου του ειδικού υλικού νευρωνικών δικτύων (DSP), των μονάδων επεξεργασίας γραφικών (GPU) και των επεξεργαστών ψηφιακού σήματος. Σε ορισμένες περιπτώσεις, το πρόγραμμα εκτέλεσης NNAPI μπορεί να εκτελέσει τα αιτήματα στη CPU [57].

Οι λεπτομέρειες και τα χαρακτηριστικά της NPU που διαθέτει ο Exynos 990 της συσκευής που χρησιμοποιήθηκε δεν είναι διαθέσιμες, για αυτό και λείπουν από τον Πίνακα 2. Επομένως, δεν θα μπορούσαμε να προβλέψουμε εκ των προτέρων την απόδοση της εκτέλεσης μοντέλων στην συγκεκριμένη μονάδα, αφού δεν είναι γνωστές οι δυνατότητές της, το σύνολο των επιπέδων που μπορεί να εκτελέσει ή η αριθμητική ακρίβεια που υποστηρίζει (32- ή 16-bit αριθμοί κινητής υποδιαστολής ή 8-bit ακέραιοι αριθμοί).

## 4.5 Παράμετροι Αλγορίθμων Συμπίεσης

Στην παρούσα Ενότητα περιγράφονται λεπτομέρειες σχετικά με τις μεθόδους συμπίεσης που εφαρμόστηκαν στα μοντέλα, όπως οι τρόποι υλοποίησής τους και οι παράμετροι που πρέπει να οριστούν πριν την εφαρμογή τους. Με βάση τις υλοποιήσεις που υπάρχουν διαθέσιμες, για τους αλγόριθμους κλαδέματος και ομαδοποίησης επιλέξαμε να αξιοποιήσουμε τη βιβλιοθήκη TFMOT (TensorFlow Model Optimization Toolkit), ενώ για τους αλγόριθμους κβαντοποίησης τις μεθόδους που προσφέρει ο μετατροπέας της βιβλιοθήκης TensorFlow Lite χωρίς επανεκπαίδευση, ώστε να μπορέσουμε να χρησιμοποιήσουμε και τα μοντέλα που είναι εκπαιδευμένα στο σύνολο δεδομένων ImageNet.

### 4.5.1 Κλάδεμα

Στα πλαίσια της διπλωματικής εργασίας, χρησιμοποιήθηκε μη δομικό κλάδεμα, δηλαδή τα βάρη κλαδεύτηκαν μεμονωμένα, με κριτήριο την απόσταση της απόλυτης τιμής τους από το μηδέν. Η υλοποίηση που προσφέρεται από τη βιβλιοθήκη TFMOT απαιτεί το κλάδεμα να γίνει με ταυτόχρονη εκπαίδευση. Για την περιγραφή της διαδικασίας που χρησιμοποιήθηκε, θα χρειαστούν οι παρακάτω όροι:

- Χρονοδιάγραμμα Κλαδέματος (Pruning Schedule): Το *χρονοδιάγραμμα κλαδέματος* ελέγχει το κλάδεμα κατά τη διάρκεια της εκπαίδευσης, ενημερώνοντας σε κάθε βήμα εάν τα βάρη ενός επιπέδου πρέπει να κλαδευτούν ή όχι, καθώς και την τελική αραιότητα (επί %) την οποία πρέπει να έχει ένα επίπεδο μετά το κλάδεμα. Στην περίπτωση αυτή, χρησιμοποιούμε κλάδεμα με βάση την *πολυωνυμική αποσύνθεση (Polynomial Decay)*, όπου η αραιότητα σε κάθε βήμα είναι η τιμή μιας πολυωνυμικής συνάρτησης, η οποία έχει ως τελική τιμή την αραιότητα που έχουμε ως στόχο. Αυτή η λειτουργία υλοποιείται, μέσω της βιβλιοθήκης keras, με την κλάση PolynomialDecay η οποία έχει τις εξής παραμέτρους:

- *initial\_sparsity*: Η αρχική αραιότητα με την οποία ξεκινά η διαδικασία του κλαδέματος.
- *final\_sparsity*: Η αραιότητα – στόχος την οποία θέλουμε να επιτύχουμε με το τέλος της διαδικασίας του κλαδέματος.
- *begin\_step*: Το βήμα της εκπαίδευσης στο οποίο αρχίζει το κλάδεμα με την αρχική αραιότητα που έχει οριστεί με βάση την παράμετρο *initial\_sparsity*.
- *end\_step*: Το βήμα στο οποίο τελειώνει η διαδικασία του κλαδέματος και εφαρμόζεται η αραιότητα – στόχος (*final\_sparsity*).
- *power*: Ο εκθέτης που εφαρμόζεται στο πολυώνυμο που καθορίζει την αραιότητα σε κάθε βήμα.
- *frequency*: Καθορίζει τη συχνότητα (σε βήματα) με την οποία εφαρμόζεται ο αλγόριθμος κλαδέματος.
- Βήματα Κλαδέματος (Pruning Steps): Κάθε φορά που επεξεργαζόμαστε μία καινούρια δέσμη έχουμε ένα καινούριο βήμα κλαδέματος. Αυτό καθορίζεται από την τιμή της παραμέτρου *frequency* που υπάρχει στην κλάση *PolynomialDecay*.

Επομένως, θεωρώντας μία διαδικασία εκπαίδευσης με έναν συγκεκριμένο αριθμό εποχών, σε κάθε βήμα  $s$ , μιας εποχής, στο οποίο γίνεται η επεξεργασία μιας καινούριας δέσμης (βήμα κλαδέματος) εφαρμόζεται ο αλγόριθμος του κλαδέματος με αραιότητα  $f(s)$ , όπου  $f$  είναι μία πολυωνυμική συνάρτηση, ο βαθμός της οποίας καθορίζεται από την παράμετρο *power* της κλάσης *PolynomialDecay*. Έπειτα, γίνεται η επεξεργασία της δέσμης και η ενημέρωση των βαρών και επαναλαμβάνεται η προηγούμενη διαδικασία, μέχρι η εκπαίδευση να φτάσει στο τέλος της [58].

Στον εφαρμογή της μεθόδου του κλαδέματος που περιγράφεται παραπάνω και που χρησιμοποιείται στη διπλωματική εργασία, σημειώνεται πως οι μεροληψίες δεν περιλαμβάνονται στα βάρη που μπορούν να κλαδευτούν και πως υπάρχουν κάποιες περιπτώσεις στρωμάτων στα οποία αυτό ισχύει καθολικά, για όλα τα βάρη, γιατί αυτό θα ήταν αρκετά επιζήμιο για την ακρίβεια των μοντέλων. Τα στρώματα αυτά είναι τα εξής:

1. Κατά βάθος συνελικτικό στρώμα
2. Στρώμα κανονικοποίησης [60]



## 4.5.2 Κβαντοποίηση

Από τα σχήματα κβαντοποίησης που προσφέρει ο μετατροπέας του TensorFlow Lite, στο πλαίσιο της εργασίας εφαρμόστηκαν τα ακόλουθα τρία, των οποίων κοινός παρονομαστής είναι η μετατροπή αριθμών κινητής υποδιαστολής των 32 bit σε ακέραιους αριθμούς των 8 bit.

1. Κβαντοποίηση με Δυναμικό Εύρος: Αυτός ο τύπος κβαντοποίησης, μετασχηματίζει μόνο τα βάρη – όχι τις ενεργοποιήσεις – από αριθμούς κινητής υποδιαστολής σε ακέραιους, γεγονός που παρέχει ακρίβεια 8 bit. Για να μειωθεί περαιτέρω η καθυστέρηση κατά τη διάρκεια της συμπερασματολογίας, οι τελεστές "δυναμικού εύρους" μετατρέπουν τις ενεργοποιήσεις σε ακέραιους των 8 bit με δυναμικό τρόπο, με κριτήριο το εύρος τους και εκτελούν υπολογισμούς με βάρη και ενεργοποιήσεις των 8 bit. Αυτή η βελτιστοποίηση παρέχει χρονικές αναμονές κοντά σε αυτές από υπολογισμούς με αναπαράσταση σταθερού αριθμού bit. Ωστόσο, οι έξοδοι εξακολουθούν να αποθηκεύονται με χρήση κινητής υποδιαστολής, οπότε η αυξημένη ταχύτητα των τελεστών δυναμικής εμβέλειας είναι μικρότερη από έναν πλήρη υπολογισμό σταθερής υποδιαστολής.
2. Κβαντοποίηση με πτώση σε αριθμούς κινητής υποδιαστολής: Η μέθοδος αυτή χρησιμοποιείται προκειμένου να είναι δυνατή η πλήρης κβαντοποίηση ενός μοντέλου, αλλά να χρησιμοποιούνται τελεστές κινητής υποδιαστολής όταν η λειτουργία ενός επιπέδου δεν μπορεί να εκτελεστεί με ακέραιους αριθμούς. Καθώς δίνουμε αυτή τη δυνατότητα "πτώσης" στο μοντέλο, οι είσοδοι και έξοδοι του μοντέλου παραμένουν κινητής υποδιαστολής και το ίδιο το μοντέλο αναλαμβάνει να εκτελεί την κβαντοποίηση και αποκβαντοποίηση του εκάστοτε δείγματος και εξόδου κάθε φορά. Πλήρης κβαντοποίηση σημαίνει κβαντοποίηση τόσο των βαρών, όσο και των ενδιαμέσων ενεργοποιήσεων ενός μοντέλου, οπότε για τη βαθμονόμηση των -δυναμικών- ενεργοποιήσεων απαιτείται ένα αντιπροσωπευτικό σύνολο δεδομένων, ώστε να εκτελεστεί ένας αριθμός συμπερασματολογιών και να καθοριστούν τα εύρη κβαντοποίησης. Συνήθως το σύνολο αυτό αποτελείται από 300-500 εικόνες που στην περίπτωση μας είναι ένα υποσύνολο του συνόλου αξιολόγησης του ImageNet.
3. Πλήρης κβαντοποίηση: Αποτελεί την πλήρη κβαντοποίηση των παραμέτρων και των ενεργοποιήσεων ενός μοντέλου σε ακέραιους αριθμούς των 8 bit, χωρίς τη δυνατότητα χρήσης οποιασδήποτε άλλης ακρίβειας. Στόχος αυτού του τύπου κβαντοποίησης είναι να προκύψουν μοντέλα με σκοπό την εκτέλεσή τους από μονάδες που υποστηρίζουν μόνο ακέραια αριθμητική. Για αυτό το λόγο, σε αυτή την περίπτωση, κατά τη μετατροπή κινητοποιούνται και οι είσοδοι - έξοδοι. Κι εδώ χρησιμοποιείται ένα αντιπροσωπευτικό σύνολο δεδομένων [59].

### 4.5.3 Ομαδοποίηση

Οι παράμετροι ενός αλγορίθμου ομαδοποίησης είναι ο αλγόριθμος που θα χρησιμοποιηθεί για την αρχικοποίηση των κεντροειδών και το πλήθος τους. Χρησιμοποιήθηκαν οι αλγόριθμοι:

1. Γραμμικής Αρχικοποίησης και
2. KMEANS++

που περιγράφηκαν στην **Ενότητα 2.3** και έγιναν πειράματα για διάφορες τιμές του πλήθους των κεντροειδών, οι οποίες επιλέχθηκαν με βάση την επίδοση κάθε μοντέλου. Η υλοποίηση που προσφέρει η βιβλιοθήκη TFMOT για την ομαδοποίηση δεν απαιτεί η εφαρμογή της να γίνει με ταυτόχρονη εκπαίδευση, όμως για τη μέγιστη ανάκτηση της χαμένης ακρίβειας, αυτό το βήμα συνιστάται ανεπιφύλακτα.

## 5. Αξιολόγηση Μεθόδων Συμπίεσης

Στο σημείο αυτό περιγράφεται το πειραματικό μέρος της διπλωματικής εργασίας, το οποίο έχει ως σκοπό την αξιολόγηση των αλγορίθμων συμπίεσης που αναφέρθηκαν παραπάνω. Η διαδικασία αυτή περιλαμβάνει την εφαρμογή των αλγορίθμων αυτών σε ένα σύνολο από μοντέλα βαθιάς μάθησης και την εξέταση διαφόρων μετρικών απόδοσής τους ως προς την ακρίβεια, το μέγεθος και τον χρόνο απόκρισης, κατά τη συμπερασματολογία. Για τη διεξαγωγή του πειραματικού μέρους της διπλωματικής εργασίας έγινε η εξής υπόθεση:

- Θεωρούμε ότι ένα μοντέλο είναι αποδεκτό μετά την εφαρμογή ενός αλγορίθμου συμπίεσης, με κριτήριο την ακρίβεια, εάν ισχύει  $acc_{init} - acc_{final} < 1\%$ , όπου  $acc_{init}$  και  $acc_{final}$  είναι η αρχική και τελική ακρίβεια ενός μοντέλου, τα οποία αξιολογήθηκαν στο σύνολο ελέγχου των CIFAR10 και ImageNet αντίστοιχα.

### 5.1 ImageNet

Λόγω του μεγάλου όγκου του συνόλου δεδομένων ImageNet, στα αντίστοιχα μοντέλα εφαρμόστηκαν μόνο οι τρεις αλγόριθμοι κβαντοποίησης που περιγράφηκαν στην Ενότητα 4.5.2 επειδή δεν απαιτούν επανεκπαίδευση του μοντέλου.

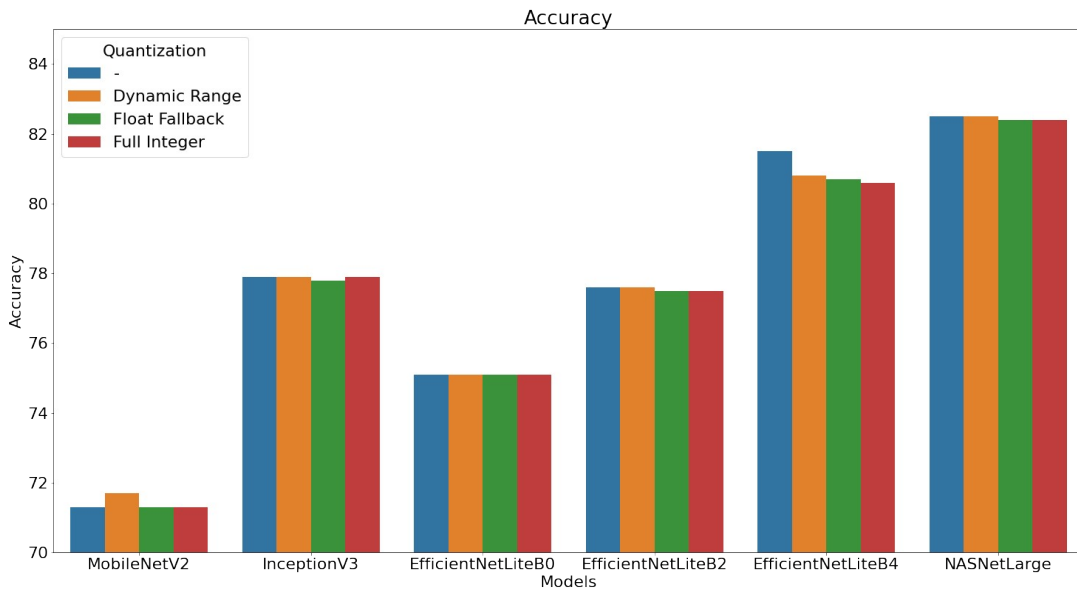
#### Ακρίβεια - Μέγεθος

Για κάθε έναν από τους παραπάνω αλγορίθμους έχει καταγραφεί η ακρίβεια και το μέγεθος των μοντέλων πριν και μετά την εφαρμογή κάθε ενός από αυτούς.

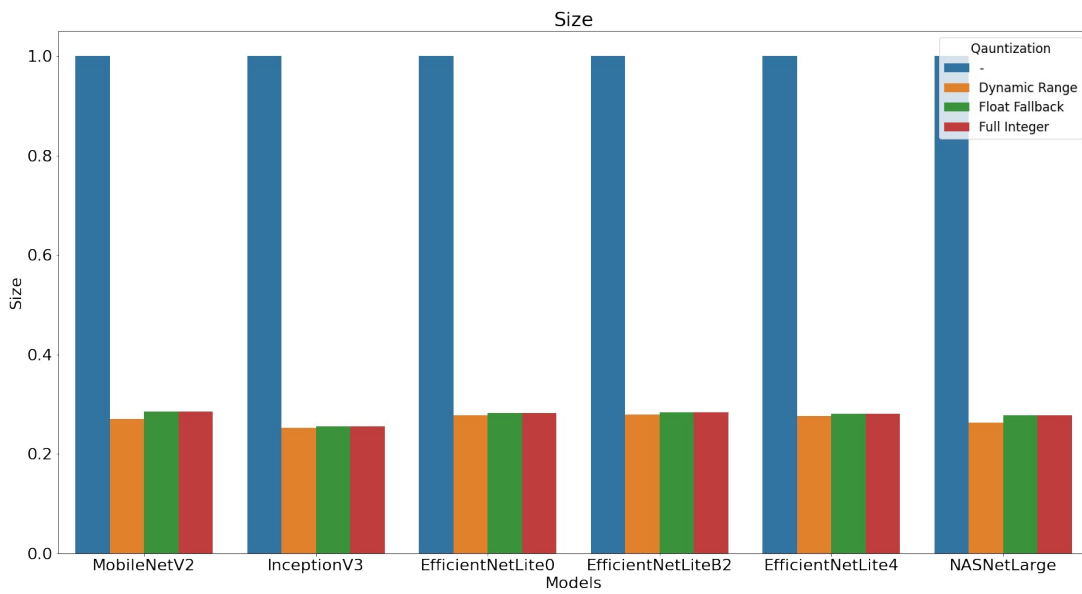
Τα συμπιεσμένα μοντέλα για τα οποία έχει επιτευχθεί επιτρεπτή ακρίβεια (σύμφωνα με την αντίστοιχη υπόθεση) σε όλες τις περιπτώσεις είναι τα ακόλουθα έξι:

- MobileNetV2
- InceptionV3
- EfficientNetLiteB0
- EfficientNetLiteB2
- EfficientNetLiteB4
- NASNetLarge

Στα Σχήματα 19 και 20, παρουσιάζονται γραφικά η επί % ακρίβεια και το ποσοστό συμπίεσης, αντίστοιχα, για καθένα από τα παραπάνω μοντέλα πριν και μετά από την εφαρμογή κάθε αλγορίθμου.



Σχήμα 19: Επίδραση Αλγορίθμων Κβαντοποίησης στην Ακρίβεια



Σχήμα 20: Επίδραση Αλγορίθμων Κβαντοποίησης στο Μέγεθος

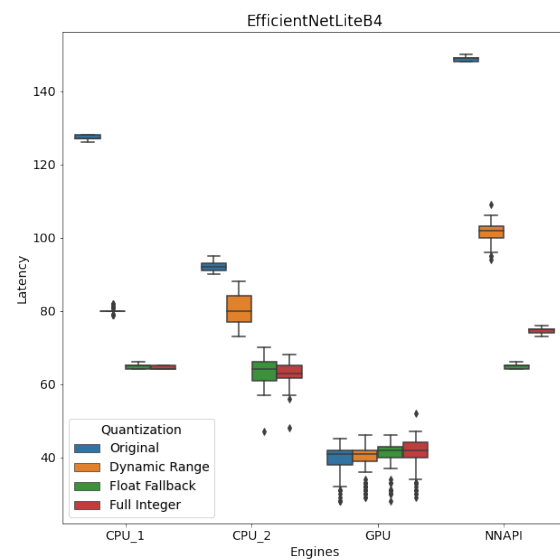
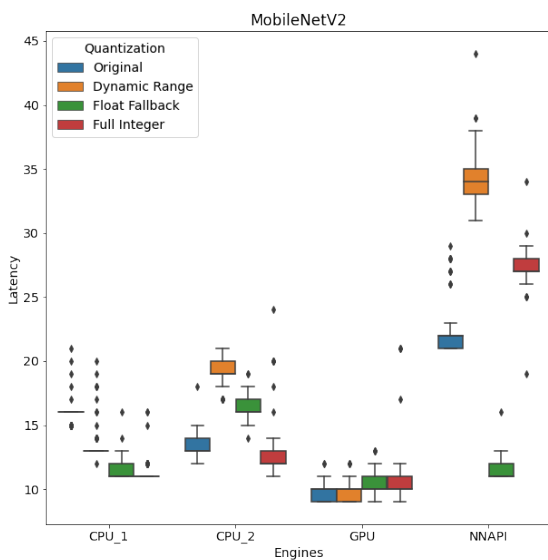
## Παρατηρήσεις:

- Η πτώση της ακρίβειας για καθένα από τα παραπάνω μοντέλα μπορεί να θεωρηθεί αμελητέα, με την εφαρμογή κάθε αλγορίθμου κβαντοποίησης από τους παραπάνω. Το γεγονός αυτό δεν είναι ένα τετριμμένο συμπέρασμα, αφού τα μοντέλα που ικανοποιούν αυτή την ιδιότητα δεν έχουν όλα την ίδια αρχιτεκτονική, ούτε ανήκουν στην ίδια κατηγορία μεγέθους.
- Η εφαρμογή καθενός από τους αλγορίθμους συμπίεσης στα παραπάνω μοντέλα έχει ως αποτέλεσμα τη δημιουργία ενός συμπιεσμένου μοντέλου με μέγεθος περίπου 4 φορές μικρότερο σε σύγκριση με το αρχικό. Στους αλγορίθμους Κβαντοποίησης με “Πτώση” σε Αριθμούς Κινητής Υποδιαστολής και Πλήρους Κβαντοποίησης, παρατηρείται μια μικρή αύξηση στο μέγεθος σε σχέση με αυτόν της Κβαντοποίησης με Δυναμικό Εύρος, η οποία πιθανόν οφείλεται στην επιπρόσθετη πληροφορία που διαθέτουν οι δύο πρώτοι αλγόριθμοι σχετικά με τα δυναμικά εύρη των ενεργοποιήσεων.

## Χρόνος Απόκρισης

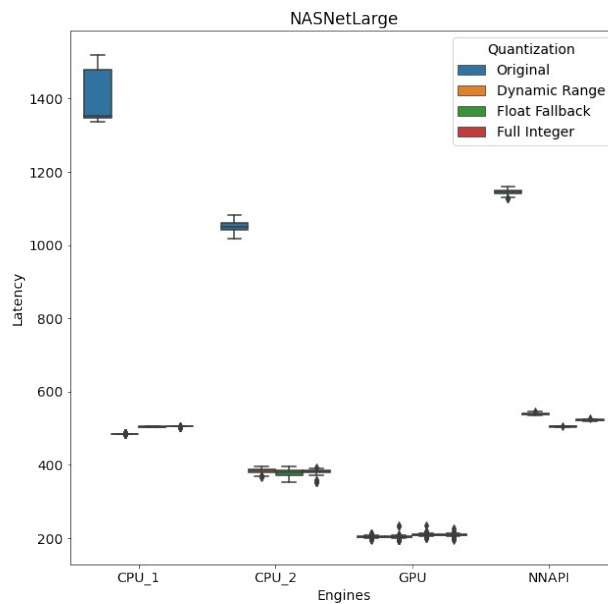
Ο προσδιορισμός του χρόνου απόκρισης των μοντέλων έχει γίνει κατά τη συμπερασματολογία σε μία κινητή συσκευή, ως προς τις μονάδες επεξεργασίας: CPU (με ένα και δύο νήματα - threads), GPU και NPU, παίρνοντας μετρήσεις για 100 εικόνες.

Κάποια ενδεικτικά αποτελέσματα από τις μετρήσεις για το χρόνο απόκρισης, στην περίπτωση των μοντέλων στα οποία έχουν εφαρμοστεί οι παραπάνω αλγόριθμοι κβαντοποίησης, για κάθε μία από τις μονάδες επεξεργασίας, φαίνονται παρακάτω. Στα Σχήματα 21, 22, 23 φαίνονται οι κατανομές των χρόνων απόκρισης κατά τη συμπερασματολογία που αναφέρονται στα μοντέλα MobileNetV2, EfficientNetLiteB4 και NASNetLarge, αντίστοιχα.



Σχήμα 21: Κατανομή του Χρόνου Απόκρισης - Κβαντοποίηση

Σχήμα 22: Κατανομή του Χρόνου Απόκρισης - Κβαντοποίηση

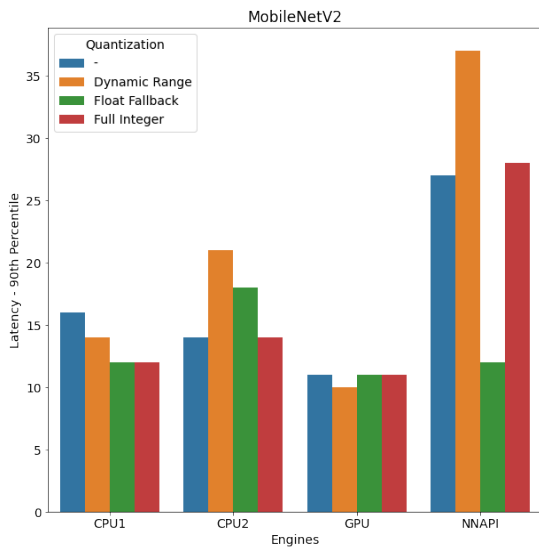


Σχήμα 23: Κατανομή του Χρόνου Απόκρισης - Κβαντοποίηση

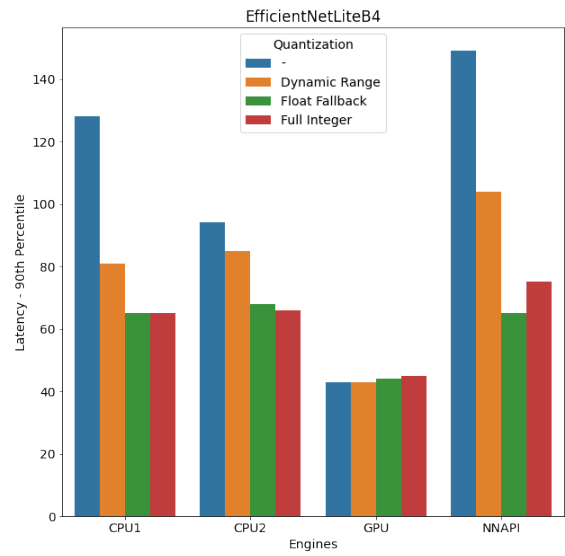
Παρατήρηση:

Δεν παρατηρείται μεγάλη διασπορά στις μετρήσεις που έχουν παρθεί για κάθε μία από τις περιπτώσεις συμπερασματολογίας.

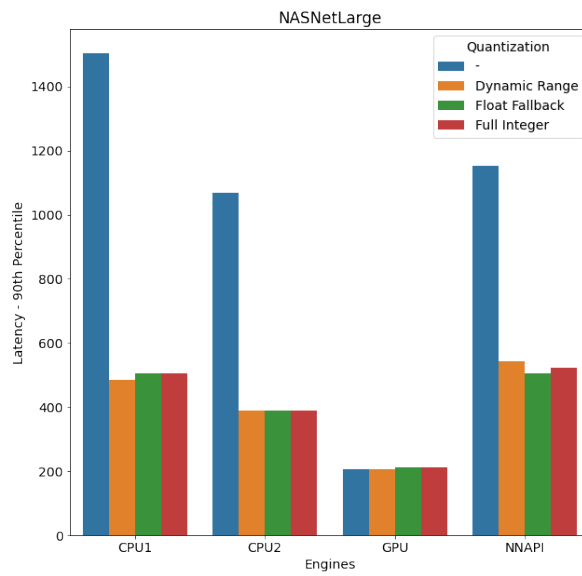
Ο χρόνος απόκρισης κατά τη συμπερασματολογία μπορεί να προσεγγιστεί με το 90ο εκατοστημόριο των μετρήσεων, όπως φαίνεται παρακάτω στα Σχήματα 24, 25 και 26 αντίστοιχα.



Σχήμα 24: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Κβαντοποίηση



Σχήμα 25: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Κβαντοποίηση



Σχήμα 26: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Κβαντοποίηση

Θεωρώντας την περίπτωση όπου παίρνουμε τις μετρήσεις κατά τη συμπερασματολογία στη CPU με 1 νήμα ως τη “βασική” μας περίπτωση, έχουμε τις εξής παρατηρήσεις οι οποίες αφορούν τις υπόλοιπες περιπτώσεις:

#### Παρατηρήσεις:

- CPU - 2 νήματα: Η αύξηση του πλήθους των νημάτων (threads), από 1 σε 2, φαίνεται να βελτιώνει όλα τα μοντέλα στα οποία οι πράξεις γίνονται με αριθμούς κινητής υποδιαστολής, όμως να επιδρά περισσότερο στα μεγαλύτερα κβαντισμένα μοντέλα, τα οποία έχουν και μεγαλύτερες δυνατότητες παραλληλοποίησης.
- GPU: Παρουσιάζει τον ελάχιστο χρόνο μεταξύ όλων των μονάδων, όμως η κβαντοποίηση δεν φαίνεται να επιταχύνει την εκτέλεση. Αυτό είναι λογικό γιατί η GPU δεν μπορεί να εκτελέσει υπολογισμούς σε ακεραίους των 8 bits (int8), οπότε για την εκτέλεση κβαντισμένων μοντέλων, πρώτα τα αποκβαντοποιεί και τα εκτελεί σε αριθμούς κινητής υποδιαστολής των 32 bits (fp32) ή αριθμούς κινητής υποδιαστολής με 16 bits των 16 bits (fp16).
- NPU: Σε όλα τα μοντέλα στα οποία έχει εφαρμοστεί ο αλγόριθμος κβαντοποίησης με “πτώση” σε αριθμούς κινητής υποδιαστολής και σε κάποια μοντέλα στα οποία έχει εφαρμοστεί πλήρης κβαντοποίηση, το NNAPI δεν “στέλνει” τα μοντέλα στην NPU, οπότε η συμπερασματολογία εκτελείται στη CPU με 1 νήμα και για το λόγο αυτό οι χρόνοι απόκρισης που παρατηρούνται στις δύο αυτές περιπτώσεις ταυτίζονται. Στην περίπτωση που τα στέλνει όμως στην NPU, οι χρόνοι απόκρισης που παρατηρούνται είναι μεγαλύτεροι από ότι στη CPU με 1 νήμα. Επομένως, φαίνεται ότι ίσως η NPU δεν είναι φτιαγμένη για κβαντισμένα μοντέλα.

## 5.2 CIFAR10

Εφόσον οι αλγόριθμοι κβαντοποίησης μελετήθηκαν στα μοντέλα που είναι εκπαιδευμένα στο ImageNet, στα μοντέλα που είναι εκπαιδευμένα στο CIFAR10 εφαρμόστηκαν αρχικά μόνο κλάδεμα και ομαδοποίηση.

Στην περίπτωση του Κλαδέματος, ο αλγόριθμος έχει εφαρμοστεί για διάφορες τιμές της τελικής αραιότητας του μοντέλου. Αρχικά, ο αλγόριθμος εφαρμόστηκε για 80% και 90% τελική αραιότητα στα στρώματα εξαγωγής χαρακτηριστικών και 80% στον ταξινομητή και έπειτα, στην περίπτωση όπου η τελική ακρίβεια ήταν αρκετά υψηλότερη από την αρχική, σε σχέση με τα υπόλοιπα μοντέλα, έγιναν δοκιμές για 95% και έπειτα 93% τελική αραιότητα, στην περίπτωση που η ακρίβεια στην προτελευταία περίπτωση δεν ήταν ικανοποιητική. Το κριτήριο επιλογής της αρχικής τιμής αραιότητας που χρησιμοποιήθηκε (80%) ήταν η ακρίβεια ενός μοντέλου να προσεγγίζει μία “αποδεκτή” τιμή, δηλαδή



να ικανοποιεί τη συνθήκη  $acc_{init} - acc_{final} < 1\%$  , η οποία παρουσιάστηκε στην Εισαγωγή του Κεφαλαίου, το οποίο ικανοποιήθηκε από όλα τα μοντέλα στα οποία εφαρμόστηκε, εκτός από τα MobileNetV2 και NASNetMobile.

Στην περίπτωση της ομαδοποίησης, έχουν εφαρμοστεί οι αλγόριθμοι Γραμμικής Αρχικοποίησης και KMEANS++ για διαφορετικά πλήθη κεντροειδών, με και χωρίς εκπαίδευση. Τα πλήθη των κέντρων για τα οποία εφαρμόστηκαν οι αλγόριθμοι ομαδοποίησης επιλέχθηκαν σε δυάδες για κάθε μοντέλο: ένα σχετικά μικρό πλήθος κέντρων και ένα πιο μεγάλο, ώστε να μελετηθεί η επίδρασή τους για κάθε αλγόριθμο. Το κριτήριο επιλογής που χρησιμοποιήθηκε ήταν η ακρίβεια ενός μοντέλου μετά την εκπαίδευση, για καθέναν από τους δύο αλγορίθμους να προσεγγίζει μία “αποδεκτή” τιμή, όπως στην περίπτωση του αλγορίθμου κλαδέματος. Η μικρή τιμή του πλήθους κέντρων είναι η ελάχιστη τιμή για την οποία έχει παρατηρηθεί πως ικανοποιείται η συνθήκη αυτή.

Μπορεί κανείς να καταλάβει αν ένα μοντέλο, στο οποίο έχει γίνει ομαδοποίηση με ένα συγκεκριμένο πλήθος κέντρων για κάθε στρώμα, μπορεί να φτάσει μια επιθυμητή τιμή στην ακρίβεια στην περίπτωση που εφαρμοστεί εκ νέου εκπαίδευση, αν η τιμή της συνάρτησης κόστους δεν είναι πάρα πολύ μεγάλη, *ανεξάρτητα από την τιμή της ακρίβειας*. Ένα παράδειγμα αποτελούν τα αποτελέσματα από την ομαδοποίηση των μοντέλων NASNetLarge και InceptionResNetV2, τα οποία βρίσκονται στον Πίνακα 19 που βρίσκεται στο **Παράρτημα Β**.

### 5.2.1 Κλάδεμα

Στους Πίνακες 19 και 20 που βρίσκονται στο **Παράρτημα Β**, υπάρχουν καταγεγραμμένα η ακρίβεια και το μέγεθος των μοντέλων πριν και μετά την εφαρμογή του αλγορίθμου Κλαδέματος. Στο σημείο αυτό, υπάρχουν κάποιες αρχικές παρατηρήσεις από τις μετρήσεις αυτές:

#### Παρατηρήσεις:

- Παρατηρείται επιθυμητή ακρίβεια μετά από την εφαρμογή του αλγορίθμου κλαδέματος, με 90% ή 93% ή 95% αραιότητα. Αυτό οφείλεται στο γεγονός πως τα μοντέλα αυτά έχουν εκπαιδευτεί στο σύνολο δεδομένων ImageNet, του οποίου οι τελικές κλάσεις είναι 100 φορές περισσότερες. Ένα πρόβλημα κατηγοριοποίησης με 1000 κλάσεις είναι αρκετά απαιτητικό και για το λόγο αυτό απαιτείται αντίστοιχα μεγάλος αριθμός παραμέτρων στο κομμάτι του μοντέλου το οποίο είναι υπεύθυνο για την εξαγωγή χαρακτηριστικών. Αντίθετα, ένα πρόβλημα κατηγοριοποίησης με 10 κλάσεις είναι αρκετά απλούστερο, άρα με τη χρήση των παραπάνω μοντέλων οδηγούμαστε στην υπερπαραμετροποίηση, η οποία επιλύεται με το κλάδεμα, όπως έχει ήδη αναφερθεί.

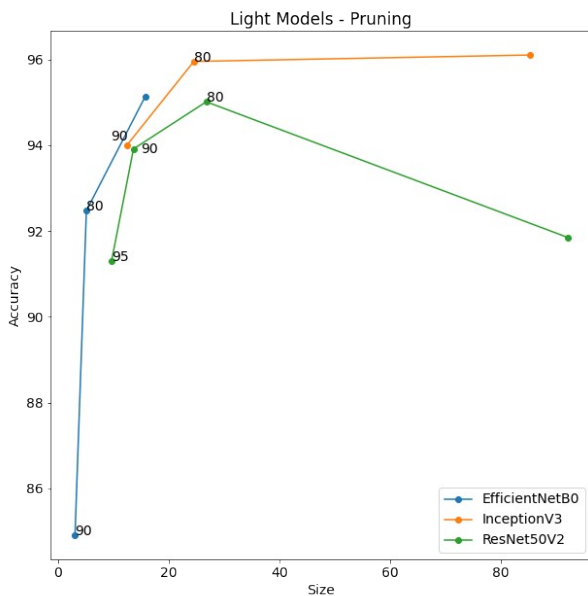
- Στην περίπτωση που έχουμε 95% αραιότητα, μπορεί να δει κανείς ότι τα μοντέλα τα οποία επιτυγχάνουν την επιθυμητή ακρίβεια έχουν την αρχιτεκτονική ResNet. Το αποτέλεσμα αυτό μπορεί να ερμηνευθεί από το γεγονός πως πολλά από τα βάρη στις υπολειμματικές συνδέσεις (residuals) δεν συνεισφέρουν ουσιαστικά στη συμπερασματολογία και επομένως δεν βλάπτουν το αποτέλεσμα όταν μηδενίζονται από τον αλγόριθμο κλαδέματος.
- Στην περίπτωση του κλαδέματος, από τα μοντέλα MobileNetV2 και NASNetMobile, τα οποία είναι ήδη αρκετά μικρότερα σε μέγεθος – αριθμό παραμέτρων – από τα υπόλοιπα, δεν μπορούν να “αφαιρεθούν” παράμετροι χωρίς αξιοσημείωτη πτώση στην ακρίβεια. Σαν συμπέρασμα έχουμε πως το πλήθος παραμέτρων των μοντέλων αυτών είναι ήδη βέλτιστο για το συγκεκριμένο πρόβλημα προς επίλυση.

### Ακρίβεια - Μέγεθος

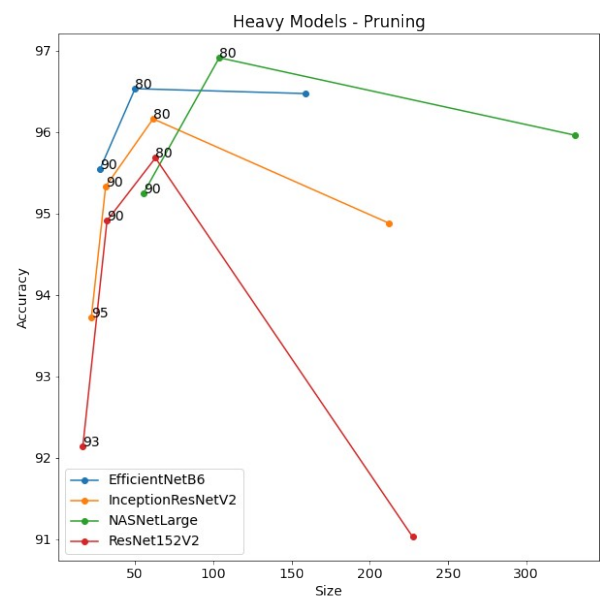
Για τη γραφική αναπαράσταση της σχέσης των μεγεθών ακρίβεια – μέγεθος (accuracy – size) για κάθε μοντέλο, πριν και μετά από την εφαρμογή του αλγορίθμου κλαδέματος, έχουμε χωρίσει τα μοντέλα στις εξής κατηγορίες:

1. **“Ελαφριά” Μοντέλα:** Τα μοντέλα των οποίων το αρχικό μέγεθος (δηλαδή το μέγεθος πριν την εφαρμογή των αλγορίθμων συμπίεσης) είναι συγκριτικά μικρό. Σε αυτή την κατηγορία έχουν προστεθεί τα εξής μοντέλα:
  - EfficientNetB0
  - InceptionV3
  - ResNet50V2
2. **“Βαριά” Μοντέλα:** Τα μοντέλα των οποίων το αρχικό μέγεθος (δηλαδή το μέγεθος πριν την εφαρμογή των αλγορίθμων συμπίεσης) είναι συγκριτικά μεγάλο. Σε αυτή την κατηγορία έχουν προστεθεί τα εξής μοντέλα:
  - NASNetLarge
  - EfficientNetB6
  - InceptionResNetV2
  - ResNet152V2

Με βάση τη συμπίεση που έχει επιτευχθεί από τον αλγόριθμο κλαδέματος που περιγράφηκε παραπάνω και τις δοκιμές που έχουν γίνει για κάθε μία από τις τιμές αραιότητας που προαναφέρθηκαν για κάθε μοντέλο, έχουμε τα διαγράμματα που φαίνονται στα Σχήματα 27 και 28 για τα ελαφριά και τα βαριά μοντέλα αντίστοιχα.



Σχήμα 27: Κλάδεμα – Ελαφριά Μοντέλα (Ποσοστό Αραιότητας - Ακρίβεια, Μέγεθος)



Σχήμα 28: Κλάδεμα – Βαριά Μοντέλα (Ποσοστό Αραιότητας - Ακρίβεια, Μέγεθος)

Για να δει κανείς τη χρονική πορεία των διαγραμμάτων στην περίπτωση του κλαδέματος, θα πρέπει να παρατηρήσει τα ζεύγη τιμών (μέγεθος, ακρίβεια) για καθένα από τα μοντέλα από το τέλος προς την αρχή. Με βάση την προηγούμενη πρόταση, σημειώνεται πως η τιμή της ακρίβειας για το μεγαλύτερο μέγεθος, για κάθε μοντέλο, αντιστοιχεί στο αρχικό, μη συμπιεσμένο μοντέλο. Με βάση τα παραπάνω, έχουμε τις εξής παρατηρήσεις:

### Παρατηρήσεις:

- Κάποια μοντέλα είναι πιο ανθεκτικά από άλλα. Για παράδειγμα το EfficientNetB0 έχει επίπτωση 10% στην ακρίβεια στην εφαρμογή του κλαδέματος με 90% τελική αραιότητα, ενώ την ίδια στιγμή υπάρχουν μοντέλα, τα οποία στην ίδια περίπτωση εφαρμογής του κλαδέματος έχουν αύξηση στην ακρίβεια που επιτυγχάνουν. Το γεγονός αυτό φαίνεται τόσο στα “ελαφριά” όσο και στα “βαριά” μοντέλα που έχουν επιλεγεί. Τα επίπεδα συμπίεσης, τα οποία έχουν αυτήν την ιδιότητα δεν είναι πάντα ανάλογα με το μέγεθος των μοντέλων και αυτό φαίνεται από την ιδιότητα των μοντέλων ResNet, που περιγράφεται στις αρχικές παρατηρήσεις. Αυτό μπορεί να οφείλεται σε κάθε μία από τις παρακάτω αιτίες:
  - Το πρόβλημα επίλυσης της κατηγοριοποίησης σε 10 κλάσεις είναι αρκετά απλοϊκό για τα παραπάνω μοντέλα, τα οποία έχουν εκπαιδευτεί για την επίλυση του ίδιου προβλήματος σε 1000 κλάσεις. Επομένως, με τη χρήση των μοντέλων αυτών οδηγούμαστε σε

υπερπαραμετροποίηση. Σαν αποτέλεσμα, περιμένουμε η ακρίβεια να μείνει σταθερή ή να αυξηθεί.

- Συμπληρωματικά με την παραπάνω πρόταση, η εκπαίδευση με μεταφορά γνώσης των μοντέλων στο σύνολο CIFAR10 δεν έγινε με το βέλτιστο τρόπο, ώστε να επιτευχθεί η μέγιστη ακρίβεια σε κάθε περίπτωση. Έτσι, με την εκ νέου ρύθμιση των βαρών που γίνεται ως βήμα σε καθέναν από τους παραπάνω αλγορίθμους επιτυγχάνεται αύξηση της ακρίβειας.
- Τα συμπιεσμένα μοντέλα, επιτυγχάνουν μέγιστη ακρίβεια για ένα κοινό εύρος μεγεθών πριν φτάσουν στο κατώφλι ενός κατώτερου μεγέθους, όπου η ακρίβεια πέφτει σε μεγάλο βαθμό. Με το γεγονός αυτό συμπεραίνουμε πως η ανθεκτικότητα στη συμπίεση εξαρτάται κυρίως από το πρόβλημα προς επίλυση και δευτερευόντως από τις ιδιότητες στην αρχιτεκτονική ενός μοντέλου βαθιάς μάθησης.

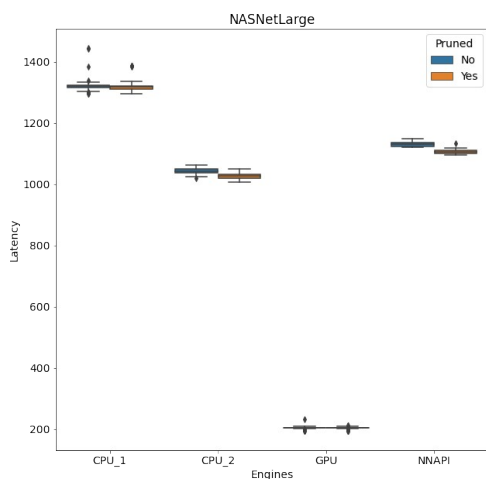
### Σημείωση:

Παρά το γεγονός ότι τα μοντέλα MobileNetV2 και NASNetMobile αποτελούν επίσης “ελαφριά” μοντέλα, δεν φαίνονται στο Σχήμα 19, καθώς δεν μπόρεσαν να φτάσουν μία τιμή επιθυμητής ακρίβειας με το αρχικό ποσοστό αραιότητας: 80%, αλλά ούτε και με χαμηλότερα ποσοστά όπως 70% ή 60%. Επομένως, μπορεί να συμπεράνει κανείς πως τα μοντέλα αυτά είναι ήδη βελτιστοποιημένα ως προς το πλήθος των παραμέτρων που διαθέτουν.

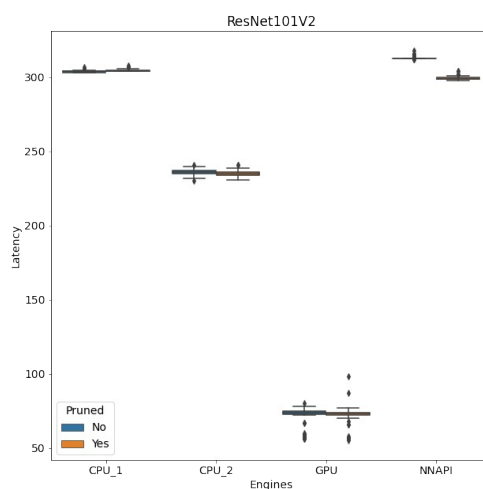
### **Χρόνος Απόκρισης**

Κάποια ενδεικτικά αποτελέσματα από τις μετρήσεις για το χρόνο απόκρισης, στην περίπτωση των μοντέλων στα οποία έχει εφαρμοστεί ο αλγόριθμος κλαδέματος, για κάθε μία από τις μονάδες επεξεργασίας, φαίνονται παρακάτω.

Παρακάτω, φαίνονται οι κατανομές για κάποια από τα μοντέλα στα οποία έχει εφαρμοστεί κλάδεμα με μεγάλη τελική αραιότητα - 90% ή 95%. Έχουν επιλεγεί τα μοντέλα NASNetLarge και ResNet101V2. Οι κατανομές τους φαίνονται στα Σχήματα 29 και 30 αντίστοιχα.



Σχήμα 29: Κατανομή του Χρόνου Απόκρισης - Κλάδεμα

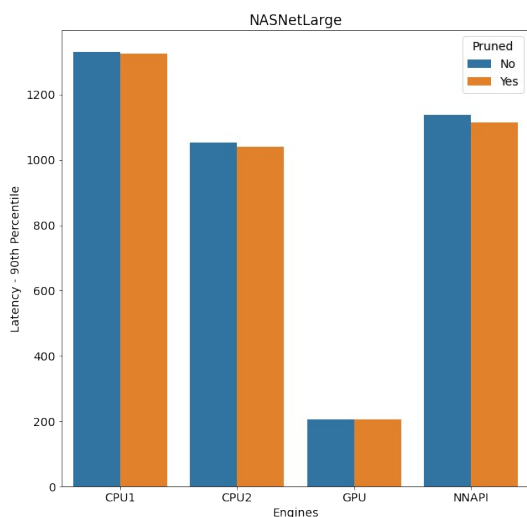


Σχήμα 30: Κατανομή του Χρόνου Απόκρισης - Κλάδεμα

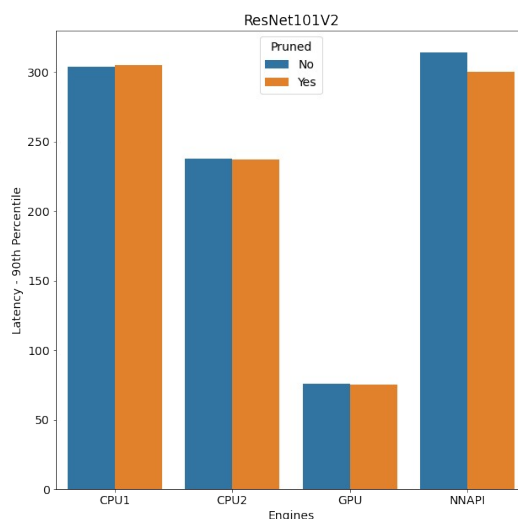
### Παρατηρήσεις:

- Δεν υπάρχει μεγάλη διασπορά στις μετρήσεις, στις περιπτώσεις και των δύο μοντέλων, επομένως μπορεί να προσδιοριστεί ο χρόνος απόκρισης με μεγαλύτερη ακρίβεια.
- Στην περίπτωση του ResNet101V2, όταν η συμπερασματολογία γίνεται με χρήση της NPU, μέσω του NNAPI, φαίνεται να υπάρχει διαφορά μεταξύ των 90ων εκατοστημορίων των μοντέλων πριν και μετά την εφαρμογή του αλγορίθμου κλαδέματος με 95% αραιότητα. Η διαφορά αυτή είναι 14ms, δηλαδή περίπου 4.6%. Επομένως, συμπεραίνουμε πως η διαφορά αυτή είναι αμελητέα.

Ένα ικανοποιητικό μέτρο προσδιορισμού για το χρόνο απόκρισης κάθε μοντέλου για κάθε μονάδα επεξεργασίας, είναι το 90ο εκατοστημόριο της κατανομής. Για κάθε μία από τις παραπάνω περιπτώσεις, αυτό φαίνεται ως εξής στο Σχήμα 31 και στο Σχήμα 32, με αντιστοιχία στα παραπάνω.



Σχήμα 31: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Κλάδεμα



Σχήμα 32: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Κλάδεμα

## 5.2.2 Ομαδοποίηση

Από τον Πίνακα 20 που βρίσκεται στο **Παράρτημα Β** και περιέχει το μέγεθος των μοντέλων πριν και μετά την εφαρμογή των αλγορίθμων Ομαδοποίησης, μπορεί να παρατηρήσει κανείς αρχικά πως το μέγεθος των μοντέλων, στα οποία έχει εφαρμοστεί αρχικοποίηση με KMEANS++ στα βάρη κάθε επιπέδου, είναι μεγαλύτερο από το μέγεθος των μοντέλων μετά την εφαρμογή Γραμμικής Αρχικοποίησης των βαρών.

### Υπερπαράμετροι Αλγορίθμων Ομαδοποίησης

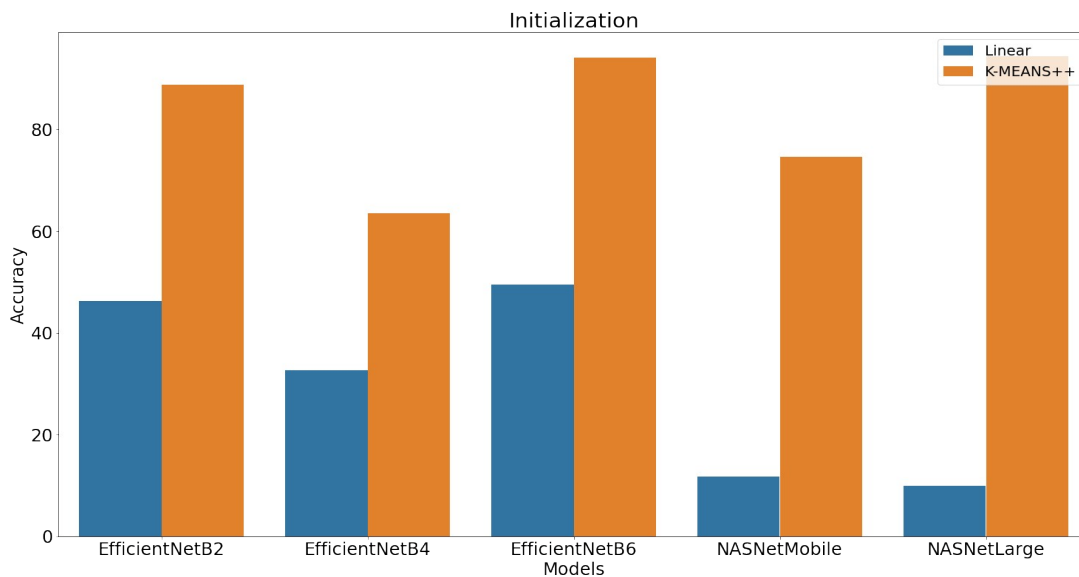
Εφόσον κάθε αλγόριθμος ομαδοποίησης χαρακτηρίζεται από τρεις παραμέτρους (αλγόριθμος αρχικοποίησης, πλήθος κεντροειδών, επανεκπαίδευση), αρχικά μελετάται η επίδραση κάθε μιας από αυτές στην τελική ακρίβεια για τα ακόλουθα πέντε μοντέλα:

- EfficientNetB2
- EfficientNetB4
- EfficientNetB6
- NASNetMobile

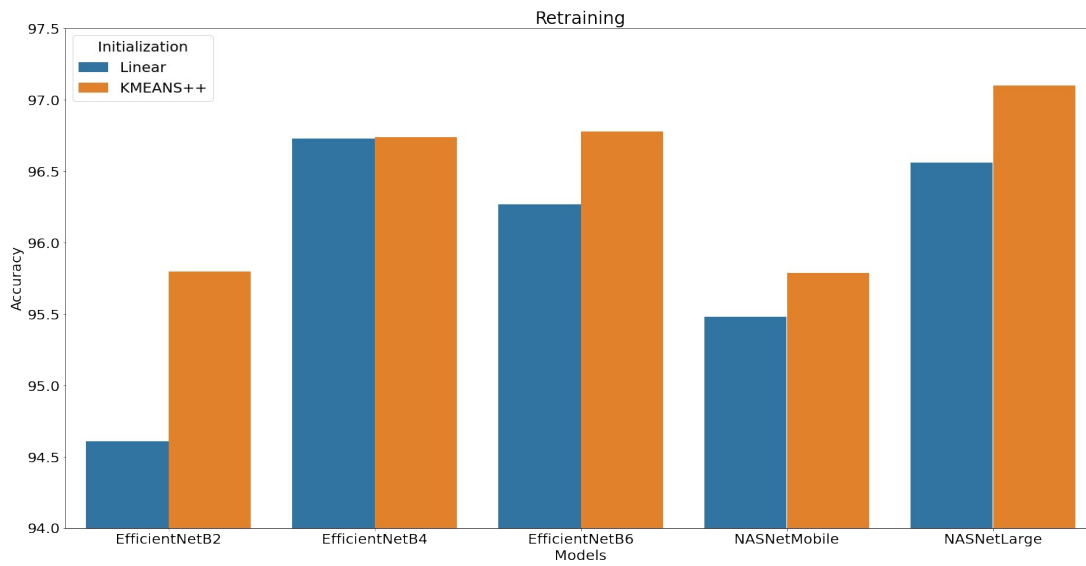
- NASNetLarge

Πιο συγκεκριμένα, στα ακόλουθα Σχήματα φαίνονται:

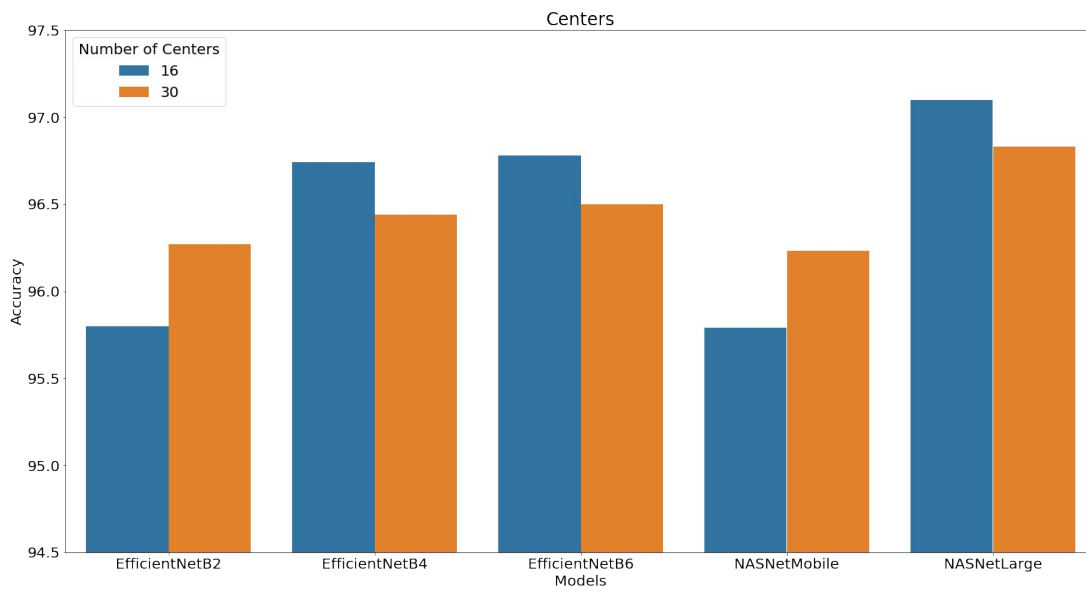
1. Η επίδραση της μεθόδου αρχικοποίησης των κεντροειδών (γραμμική ή KMEANS++), όταν το πλήθος των κεντροειδών είναι σταθερό και ίσο με 16 και στις περιπτώσεις όπου δεν έχει (Σχήμα 33) και έχει (Σχήμα 34) ακολουθήσει επανεκπαίδευση του μοντέλου μετά την εφαρμογή του αλγορίθμου.
2. Η επίδραση του πλήθους των κεντροειδών για τιμές 16 και 30 και για αρχικοποίηση με τον αλγόριθμο KMEANS++ και επανεκπαίδευση των βαρών των μοντέλων (Σχήμα 35).



Σχήμα 33: Σύγκριση της Επίδρασης των Αλγορίθμων Αρχικοποίησης ως προς την Ακρίβεια



Σχήμα 34: Σύγκριση της Επίδρασης της Επανεκπαίδευσης ανά Αλγόριθμο Αρχικοποίησης



Σχήμα 35: Σύγκριση της Επίδρασης του Πλήθους Κεντροειδών ως προς την Ακρίβεια



Με βάση τα παραπάνω διαγράμματα μπορούν να διατυπωθούν οι εξής παρατηρήσεις:

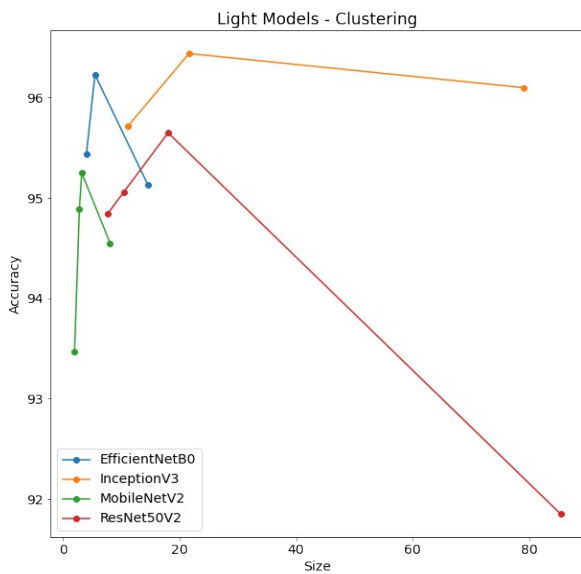
- Ο αλγόριθμος KMEANS++ επιφέρει καλύτερη αρχικοποίηση των βαρών ενός στρώματος σε σύγκριση με τον αλγόριθμο Γραμμικής Αρχικοποίησης, όπως φαίνεται από τις διαφορές στην ακρίβεια των μοντέλων (Σχήμα 33). Αυτό συμβαίνει γιατί ο αλγόριθμος KMEANS++ λαμβάνει υπ' όψη τη γεωμετρία μεταξύ των κέντρων που επιλέγονται, ενώ στη Γραμμική Αρχικοποίηση, τα κέντρα επιλέγονται ομοιόμορφα, με βάση το εύρος των τιμών χωρίς να λαμβάνεται υπ' όψη η κατανομή τους. Επιπρόσθετα, από το Σχήμα 34, μπορεί να δει κανείς πως τα μοντέλα των οποίων τα βάρη έχουν αρχικοποιηθεί με τον αλγόριθμο KMEANS++ έχουν υψηλότερη ακρίβεια, ακόμα και μετά την επανεκπαίδευση.
- Συγκρίνοντας τα αποτελέσματα από τα Σχήματα 33 και 34 μπορεί να παρατηρήσει κανείς ότι τα μοντέλα μετά την εκ νέου εκπαίδευση έχουν αρκετά αυξημένη απόδοση, ακόμη και σε περιπτώσεις όπου η αρχική ακρίβεια είναι πολύ χαμηλή. Για παράδειγμα, το μοντέλο NASNetLarge μετά την εφαρμογή Γραμμικής Αρχικοποίησης των βαρών σε 16 ομάδες ανά στρώμα, έχει ακρίβεια 10.00%, ενώ μετά την εκπαίδευση η ακρίβεια αυξάνεται στην τιμή 96.56%, η οποία αποτελεί μία “επιτρεπτή” τιμή στην ακρίβεια για το συγκεκριμένο μοντέλο. Παρόμοια συμπεριφορά έχει και το μοντέλο NASNetMobile, το οποίο είναι αρκετά μικρότερο σε μέγεθος.
- Τέλος, για την επίδραση του πλήθους των κέντρων στην ακρίβεια ενός μοντέλου, μετά την εκπαίδευση, στην περίπτωση που φαίνεται στο Σχήμα 35, για τις τιμές 16 και 30, παρατηρούμε πως είναι δυνατόν να επιτευχθεί μεγαλύτερη ακρίβεια με μικρότερο αριθμό κέντρων. Αυτό συμβαίνει για κάποιο από τα παρακάτω δύο ενδεχόμενα:
  - Τα βάρη ενός μοντέλου πριν την εκπαίδευση σχηματίζουν μικρό αριθμό ομάδων μεταξύ τους και έτσι η πτώση στην ακρίβεια δεν είναι μεγάλη εξ' αρχής
  - Η τιμή της συνάρτησης κόστους για την αξιολόγηση ενός μοντέλου δεν είναι μεγάλη, επομένως κατά την εκπαίδευση, μπορούν οι τιμές των ομαδοποιημένων βαρών να μετασχηματιστούν κατάλληλα, ώστε η τελική ακρίβεια να είναι επιτρεπτή.

## **Ακρίβεια – Μέγεθος**

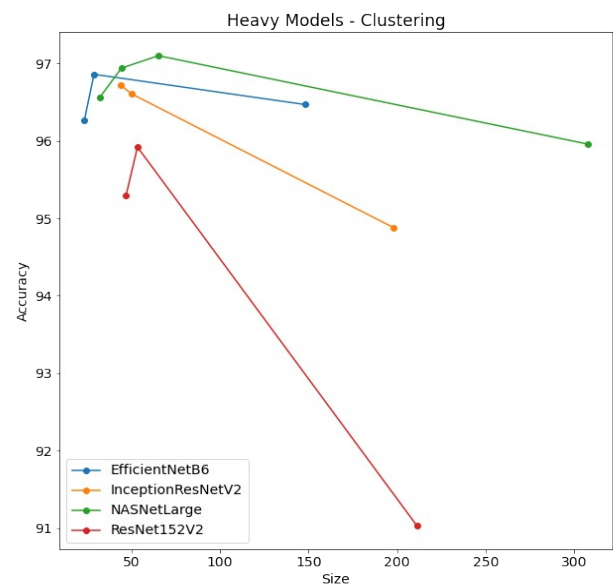
Για να φανεί η επίδραση των αλγορίθμων ομαδοποίησης, αφού δεν επηρεάζονται από αυτούς εκ των προτέρων η αρχιτεκτονική ενός μοντέλου ή η αναπαράσταση των αριθμών σε bits, έχουμε εφαρμόσει σε μοντέλα, τα οποία βρίσκονται σε μορφή TFLite, τον αλγόριθμο συμπίεσης ZIP για τη μέτρηση του μεγέθους. Για τη γραφική αναπαράσταση της σχέσης των μεγεθών *ακρίβεια – μέγεθος* (accuracy – size) για κάθε μοντέλο, πριν και μετά από την εφαρμογή των αλγορίθμων ομαδοποίησης, έχουμε χωρίσει τα μοντέλα στις εξής κατηγορίες:

1. **“Ελαφριά” Μοντέλα:** Τα μοντέλα των οποίων το αρχικό μέγεθος (δηλαδή το μέγεθος πριν την εφαρμογή των αλγορίθμων συμπίεσης) είναι συγκριτικά μικρό. Σε αυτή την κατηγορία έχουν προστεθεί τα εξής μοντέλα:
  - MobileNetV2
  - EfficientNetB0
  - InceptionV3
  - ResNet50V2
- **“Βαριά” Μοντέλα:** Τα μοντέλα των οποίων το αρχικό μέγεθος (δηλαδή το μέγεθος πριν την εφαρμογή των αλγορίθμων συμπίεσης) είναι συγκριτικά μεγάλο. Σε αυτή την κατηγορία έχουν προστεθεί τα εξής μοντέλα:
  - NASNetLarge
  - EfficientNetB6
  - InceptionResNetV2
  - ResNet152V2

Με βάση τη συμπίεση που έχει επιτευχθεί από τους αλγορίθμους ομαδοποίησης που περιγράφηκαν παραπάνω και τις δοκιμές που έχουν γίνει για κάθε μία από τις υπερπαραμέτρους που προαναφέρθηκαν, για κάθε μοντέλο, έχουμε τα διαγράμματα που φαίνονται στα Σχήματα 36 και 37 για τα ελαφριά και τα βαριά μοντέλα αντίστοιχα.



Σχήμα 37: Ομαδοποίηση - Ελαφριά Μοντέλα (Ακρίβεια - Μέγεθος)



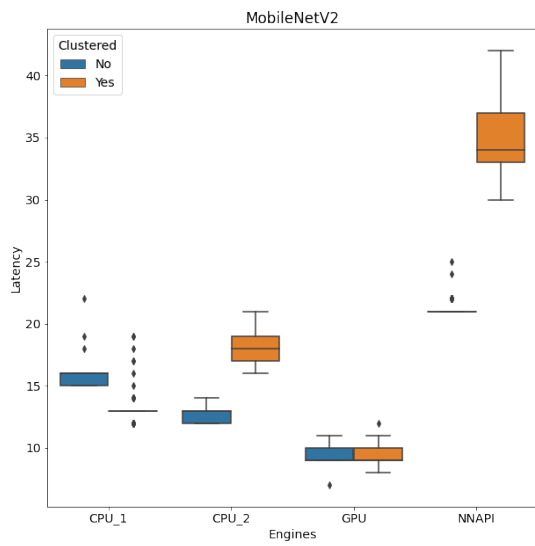
Σχήμα 36: Ομαδοποίηση - Βαριά Μοντέλα (Ακρίβεια - Μέγεθος)

Για να δει κανείς τη χρονική πορεία των διαγραμμάτων στην περίπτωση του κλαδέματος, θα πρέπει να παρατηρήσει τα ζεύγη τιμών (μέγεθος, ακρίβεια) για καθένα από τα μοντέλα από το τέλος προς την αρχή. Επομένως, σημειώνεται πως η τιμή της ακρίβειας για το μεγαλύτερο μέγεθος, για κάθε μοντέλο, αντιστοιχεί στο αρχικό, μη συμπιεσμένο μοντέλο. Με βάση τα παραπάνω, οι παρατηρήσεις που έχουμε ταυτίζονται με αυτές που έχουν διατυπωθεί στην περίπτωση του αλγορίθμου κλαδέματος, με μόνη διαφορά πως στην ομαδοποίηση τα μικρότερα μοντέλα (MobileNetV2 και NASNetMobile, όπως φαίνεται στον Πίνακα 19 στο **Παράρτημα Β**), πετυχαίνουν συμπίεση με αποδεκτές τιμές στην ακρίβεια, ακόμα και αύξηση αυτής. Επίσης, λόγω της επανεκπαίδευσης που μπορεί να εκτελεστεί, τα μοντέλα μετά την εφαρμογή ομαδοποίησης και επανεκπαίδευσης έχουν μεγαλύτερη ακρίβεια σε σύγκριση με τα αντίστοιχα στα οποία έχει εφαρμοστεί ο αλγόριθμος κλαδέματος, πετυχαίνοντας αντίστοιχη χωρική συμπίεση.

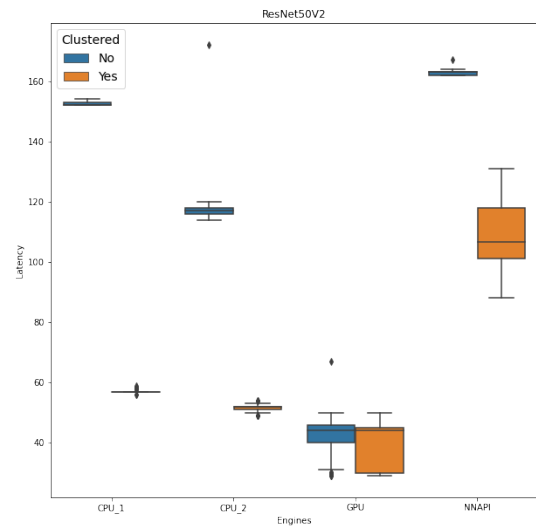
### **Χρόνος Απόκρισης**

Κατά τη μέτρηση του χρόνου απόκρισης μοντέλων στα οποία είχε εφαρμοστεί κάποιος αλγόριθμος ομαδοποίησης, παρατηρήθηκε γρήγορα πως η συγκεκριμένη μέθοδος δεν επιταχύνει την εκτέλεση των μοντέλων (ακριβώς όπως και συνέβη και στο κλάδεμα). Για τον παραπάνω λόγο, επιλέχθηκε στα ομαδοποιημένα μοντέλα να εφαρμοστεί συνδυαστικά και Κβαντοποίηση Δυναμικού Εύρους, ώστε εκτός από τη μείωση στο μέγεθος, να μετρηθεί και η βελτίωση στον χρόνο απόκρισης. Σχετικά με την ακρίβεια των τελικών μοντέλων, αναμένουμε ελάχιστη έως και καθόλου πτώση καθώς τα βάρη στην ουσία είναι ήδη κβαντισμένα, απλά όχι σε ακέραιες τιμές. Για παράδειγμα, για το μοντέλο MobileNetV2 η ακρίβεια πριν και μετά την εφαρμογή κβαντοποίησης δυναμικού εύρους είναι 95.25% και 94.94% αντίστοιχα. Δηλαδή έχουμε διαφορά  $0.31\% < 1\%$ .

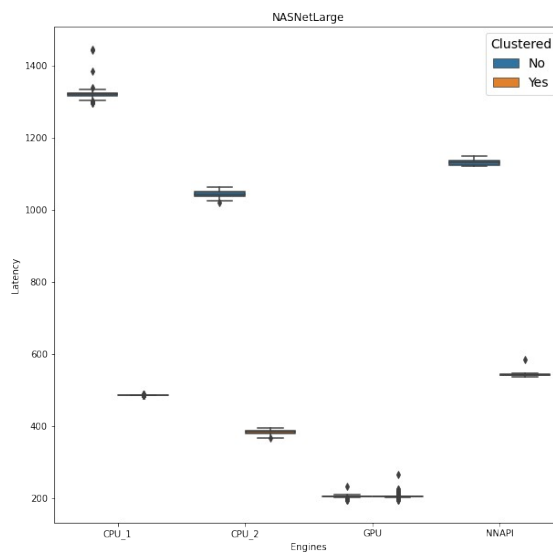
Ενδεικτικά, φαίνονται οι κατανομές για τα μοντέλα MobileNetV2, ResNet50V2 και NASNetLarge, τα οποία αποτελούν ένα μικρό, ένα μεγάλο και ένα μεσαίου μεγέθους μοντέλο, στα Σχήματα 38, 39 και 40 αντίστοιχα.



Σχήμα 38: Κατανομή του Χρόνου Απόκρισης - Ομαδοποίηση



Σχήμα 39: Κατανομή του Χρόνου Απόκρισης - Ομαδοποίηση

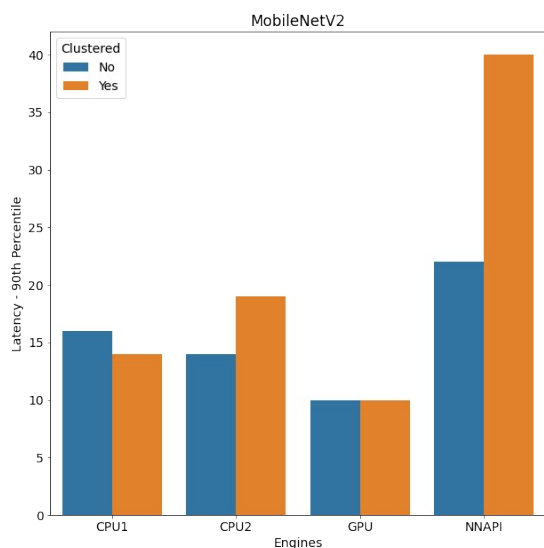


Σχήμα 40: Κατανομή του Χρόνου Απόκρισης - Ομαδοποίηση

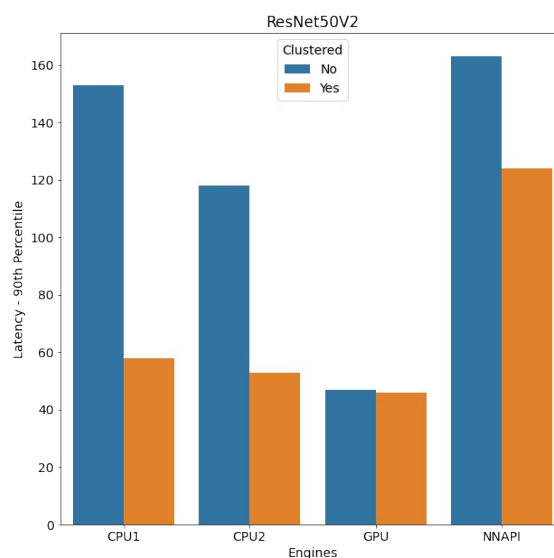
### Παρατήρηση:

Παρατηρούμε πως και στα τρία μοντέλα και ιδιαίτερα στην περίπτωση του NASNetLarge, δεν υπάρχει μεγάλη διασπορά στις τιμές των μετρήσεων για κάθε μία από τις επιμέρους κατανομές, επομένως ο χρόνος απόκρισης μπορεί να προσεγγιστεί με αρκετά καλή ακρίβεια.

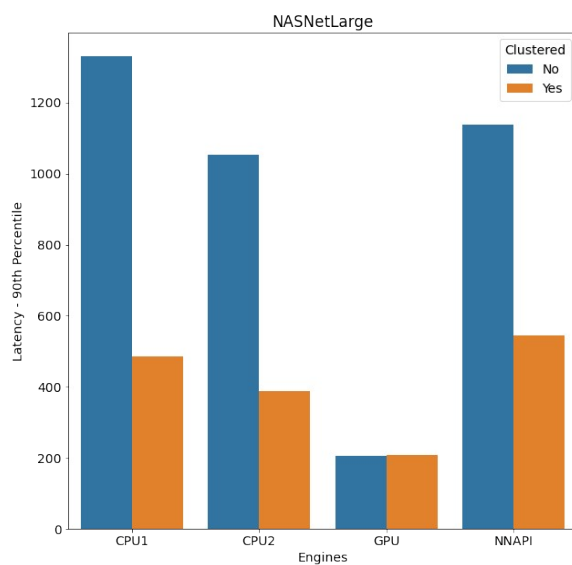
Για την προσέγγιση του χρόνου απόκρισης κάθε μοντέλου, για κάθε μονάδα επεξεργασίας, θα χρησιμοποιηθεί το 90ο εκατοστημόριο της κατανομής. Για κάθε μία από τις παραπάνω περιπτώσεις, αυτό φαίνεται ως εξής στα Σχήματα 41, 42 και 43, με αντιστοιχία στα παραπάνω.



Σχήμα 41: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Ομαδοποίηση



Σχήμα 42: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Ομαδοποίηση



Σχήμα 43: 90ο Εκατοστημόριο του Χρόνου Απόκρισης - Ομαδοποίηση

### Παρατήρηση:

Από τις μετρήσεις για το χρόνο απόκρισης που έγιναν για τα παραπάνω μοντέλα, μπορεί να παρατηρήσει κανείς πως έχουμε την ίδια εικόνα, με πολύ μικρή απόκλιση, με τα μοντέλα που έχουν εκπαιδευτεί στο ImageNet και στα οποία έχει εφαρμοστεί η Κβαντοποίηση Δυναμικού Εύρους.

Η παραπάνω παρατήρηση μας παρακινεί να προσδιορίσουμε τη συνεισφορά του κάθε αλγορίθμου (Ομαδοποίησης, Κβαντοποίηση Δυναμικού Εύρους) στη χρονική επιτάχυνση των συμπιεσμένων μοντέλων. Η επίδραση ενός αλγορίθμου συμπίεσης στη μείωση του χρόνου απόκρισης κατά τη συμπερασματολογία, ως προς μία μονάδα επεξεργασίας, μπορεί να προσδιοριστεί από τον λόγο των 90ων εκατοστημορίων πριν και μετά την εφαρμογή του, δηλαδή:

$$\lambda = \frac{90^{th} \text{ Percentile}_{\text{χωρίς συμπίεση}}}{90^{th} \text{ Percentile}_{\text{με συμπίεση}}}$$

Σε προηγούμενη παράγραφο της **Υποενότητας 4.6.1** έχει παρουσιαστεί η επίδραση της κβαντοποίησης δυναμικού εύρους για τα μοντέλα βαθιάς μάθησης που είναι εκπαιδευμένα στο σύνολο δεδομένων ImageNet. Επομένως, συγκρίνοντας τη μέση τιμή του λόγου των 90ων εκατοστημορίων για την περίπτωση της κβαντοποίησης δυναμικού εύρους και για την περίπτωση της κβαντοποίησης δυναμικού εύρους σε συνδιασμό με την ομαδοποίηση, θα βρεθεί η συνεισφορά του κάθε αλγορίθμου στη δεύτερη περίπτωση.

Σαν αποτέλεσμα, υπολογίζοντας το μέσο όρο των λόγων για τα μοντέλα που αξιολογήθηκαν σε κάθε μία από τις παραπάνω περιπτώσεις διαμορφώνεται ο εξής πίνακας:

$\lambda$	CPU1	CPU2	GPU	NNAPI
Κβαντοποίηση	2.01	1.56	1	1.31
Ομαδοποίηση και Κβαντοποίηση	1.73	1.35	1.02	1.24

Πίνακας 3: Επίδραση Ομαδοποίησης στην Επιτάχυνση

Παρατήρηση:

Μπορεί να παρατηρήσει κανείς, πως η κβαντοποίηση δυναμικού εύρους σε συνδυασμό με την ομαδοποίηση επιβραδύνει το χρόνο απόκρισης ενός μοντέλου κατά τη συμπερασματολογία, δεδομένης μίας μονάδας επεξεργασίας, όπως αυτή ορίζεται παραπάνω, αφού η τιμή του λόγου  $\lambda$  σε αυτήν την περίπτωση είναι σε όλες τις περιπτώσεις μικρότερη από την τιμή του λόγου όταν εφαρμόστηκε μόνο η κβαντοποίηση.

## 6. Επίλογος

Στο κεφάλαιο αυτό, περιγράφονται τα τελικά συμπεράσματα τα οποία μπορούμε να εξάγουμε από την αξιολόγηση που πραγματοποιήθηκε για κάθε μία από τις μεθόδους συμπίεσης βαθιών νευρωνικών δικτύων: *κλάδεμα*, *ομαδοποίηση* και *κβαντοποίηση*.

### 6.1 Συμπεράσματα

Για τη μελέτη και αξιολόγηση των αλγορίθμων συμπίεσης βαθιών νευρωνικών δικτύων όσον αφορά στην παρούσα διπλωματική εργασία, έχουν ληφθεί υπ' όψη αρκετοί παράγοντες, όπως η ακρίβεια, το μέγεθος, το πλήθος των πράξεων κινητής υποδιαστολής (FLOPs) και ο χρόνος απόκρισης των μοντέλων κατά τη συμπερασματολογία. Παρακάτω, αναφέρονται κάποια γενικότερα και χρήσιμα συμπεράσματα για κάθε μέθοδο συμπίεσης.

#### Κλάδεμα

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, όπου τα μοντέλα που χρησιμοποιήθηκαν για την αξιολόγηση της μεθόδου του κλαδέματος αποτελούνταν από υπέρογκο αριθμό παραμέτρων σε σχέση με το τελικό πρόβλημα προς επίλυση, το οποίο ήταν η ταξινόμηση σε 10 κλάσεις, ο αλγόριθμος κατάφερε μία αρκετά καλή μείωση του μεγέθους των μοντέλων, σε συνδυασμό με “επιτρεπτή” ακρίβεια, ως προς την αρχική μας υπόθεση. Σε ορισμένες περιπτώσεις, μπορεί να παρατηρήσει κανείς πως είναι δυνατή και η *αύξηση* της ακρίβειας των μοντέλων, όπως για παράδειγμα στα μοντέλα ResNet50V2, ResNet101V2 και ResNet152V2. Αξίζει να σημειωθεί πως με τη χρήση του αλγορίθμου κλαδέματος, με 95% αραιότητα στα στρώματα, για τα μοντέλα ResNet101V2, ResNet152 και ResNet152V2 παρατηρείται επίσης επιτρεπτή ακρίβεια. Παρ' όλα αυτά, δεν παρατηρήθηκε βελτίωση στο χρόνο απόκρισης κατά τη συμπερασματολογία, όπως αυτός μετρήθηκε πειραματικά. Αυτό οφείλεται στο γεγονός πως ο αλγόριθμος που χρησιμοποιήθηκε ανήκει στην κατηγορία των μεθόδων του μη δομικού κλαδέματος και επομένως δεν επηρεάζει την αρχιτεκτονική ενός μοντέλου, παρά μόνο τις τιμές των βαρών μεμονωμένα.

#### Ομαδοποίηση

Η ομαδοποίηση αποτελεί τη μέθοδο συμπίεσης με την οποία επιτεύχθηκε η μεγαλύτερη ακρίβεια σε σύγκριση με το κλάδεμα, όπως φαίνεται στο *Σχήμα 31*. Αυτό οφείλεται, όπως σημειώθηκε και παραπάνω, στην εκ νέου εκπαίδευση των βαρών, μετά το τέλος της διαδικασίας των αλγορίθμων που



ανήκουν σε αυτή την κατηγορία. Επιπλέον, σημειώθηκε αρκετά μεγάλη βελτίωση στο μέγεθος των μοντέλων, αφού, όπως μπορεί να δει κανείς, υπάρχουν περιπτώσεις όπου η ομαδοποίηση έχει επιτύχει 10 φορές μικρότερο μέγεθος μοντέλου σε σχέση με το αρχικό. Παρά την πολύ καλή απόδοση των αλγορίθμων ομαδοποίησης, όσον αφορά τόσο στην ακρίβεια όσο και στο μέγεθος ενός μοντέλου, ο χρόνος απόκρισης κατά τη συμπερασματολογία δεν επηρεάζεται από τους αλγορίθμους, αφού αυτοί επηρεάζουν μόνο τις τιμές των βαρών και όχι την κωδικοποίησή τους σε bits, όπως συμβαίνει για παράδειγμα στην περίπτωση της κβαντοποίησης.

### Κβαντοποίηση

Η κβαντοποίηση αποτελεί μία αρκετά διαδεδομένη μέθοδο συμπίεσης βαθιών νευρωνικών δικτύων. Παρατηρήθηκε πως ο αλγόριθμος κβαντοποίησης δυναμικού εύρους μειώνει το μέγεθος των αρχικών μοντέλων κατά 4 φορές περίπου, χωρίς να επηρεάζει σημαντικά την ακρίβεια. Για τους αλγορίθμους κβαντοποίησης με “πτώση” σε αριθμούς κινητής υποδιαστολής και πλήρους κβαντοποίησης, η ακρίβεια στα περισσότερα μοντέλα μειώνεται αισθητά, ενώ η μείωση στο μέγεθος δεν διαφέρει σημαντικά από αυτήν που επιφέρει η κβαντοποίηση με δυναμικό εύρος. Όσον αφορά στο χρόνο απόκρισης των μοντέλων, οι αλγόριθμοι κβαντοποίησης συνεισφέρουν σημαντικά στη μείωσή του. Παρατηρείται μάλιστα πως, σε ορισμένες περιπτώσεις, δεν έχουν την ίδια επίδραση σε μοντέλα με διαφορετικό μέγεθος. Το γεγονός αυτό φαίνεται αν συγκρίνει κανείς τους αρχικούς με τους τελικούς χρόνους απόκρισης ως προς τη CPU με 2 νήματα και την NPU (μέσω του NNAPI) για το MobileNetV2, το οποίο αποτελεί ένα μικρό σχετικά μοντέλο και για το NASNetLarge, το οποίο είναι το μεγαλύτερο σε μέγεθος από τα αρχικά μοντέλα. Τέλος, οι αλγόριθμοι κβαντοποίησης δεν φαίνεται να επηρεάζουν το χρόνο απόκρισης κατά τη συμπερασματολογία στη GPU.

Τα προηγούμενα συμπεράσματα φαίνονται πιο συγκεντρωμένα στον Πίνακα 4, όπου για κάθε μέθοδο (Κλάδεμα, Ομαδοποίηση, Κβαντοποίηση) οι οποίες αποτελούν κάθε μία από τις σειρές του πίνακα, σημειώνονται η μέγιστη επιτάχυνση που παρουσιάστηκε σε κάθε μοντέλο για τους αλγορίθμους Κλαδέματος και Ομαδοποίησης, η μέση επιτάχυνση που παρουσιάστηκε για τους αλγορίθμους Κβαντοποίησης και η μέση μείωση μεγέθους για κάθε μία από τις τρεις παραπάνω περιπτώσεις. Οι μέσες τιμές που αναφέρονται έχουν υπολογιστεί ως προς κάθε μία από τις μονάδες επεξεργασίας. Τέλος, για τον υπολογισμό των μεγεθών που αναφέρθηκαν, ο χρόνος απόκρισης προσεγγίστηκε από το 90ο εκατοστημόριο για κάθε σύνολο μετρήσεων που έχει ληφθεί για κάθε μοντέλο.

	Επιτάχυνση	Μείωση Μεγέθους
Κλάδεμα	x1.05	x15.07
Ομαδοποίηση	x5.08	x11.02
Κβαντοποίηση	x2.12	x3.75

Πίνακας 4: Τελικά Συμπεράσματα

### Σημειώνονται τα εξής:

- Από τα στατιστικά μεγέθη που έχουν υπολογιστεί για κάθε μοντέλο, μπορεί να δει κανείς πως η μεγαλύτερη επιτάχυνση συμβαίνει για τις μετρήσεις στη CPU και ιδιαίτερα στην περίπτωση με 1 νήμα (thread).
- Η μέγιστη επιτάχυνση που αναφέρεται για το Κλάδεμα αναφέρεται στην NPU, για το μοντέλο ResNet101V2. Αντίστοιχα, για τη μέθοδο της ομαδοποίησης αυτή επιτεύχθηκε στη συμπερασματολογία στη CPU με 1 νήμα (thread), για το μοντέλο VGG19. Για την κβαντοποίηση, η μέγιστη μέση επιτάχυνση παρατηρήθηκε για τις μετρήσεις που έγιναν στη CPU με 1 νήμα.
- Η μέση μείωση μεγέθους, ως προς το κλάδεμα, πάρθηκε για τα μοντέλα στα οποία έχει εφαρμοστεί ο αλγόριθμος με 95% αραιότητα, ως ακραία περίπτωση. Για τον υπολογισμό της μέσης μείωσης μεγέθους ως προς την ομαδοποίηση έχει παρθεί ο μέσος όρος των μεγεθών, για τα μοντέλα στα οποία έχει εφαρμοστεί Γραμμική Αρχικοποίηση με το μικρότερο από τα δύο κεντροειδή, σε συνδυασμό με εκ νέου εκπαίδευση.
- Τέλος, επαληθεύεται πως οι αλγόριθμοι Κβαντοποίησης προσφέρουν κατά μέσο όρο 4 φορές μικρότερα σε μέγεθος μοντέλα.

Τα παραπάνω φαίνονται αναλυτικότερα στους πίνακες στα **Παραρτήματα Α και Β**.

## 6.2 Μελλοντική Εργασία

Η συμπίεση των βαθιών νευρωνικών δικτύων αποτελεί ένα θέμα, το οποίο έχει αρκετές πρακτικές εφαρμογές, όπως η πιο αποδοτική ενσωμάτωση μοντέλων βαθιάς μάθησης σε συσκευές μειωμένων υπολογιστικών και χωρητικών δυνατοτήτων, όπως είναι τα κινητά τηλέφωνα, αλλά και πολλές θεωρητικές επεκτάσεις.

Αρχικά, οι μέθοδοι συμπίεσης που παρουσιάστηκαν θα μπορούσαν να εφαρμοστούν σε μοντέλα με διαφορετικές αρχιτεκτονικές, όπως είναι για παράδειγμα τα Transformers.

Μία διαφορετική επέκταση της εργασίας αυτής, αποτελεί η εξέταση των συνδυασμών των αλγορίθμων συμπίεσης βαθιών νευρωνικών δικτύων ως προς την επίπτωση που έχουν στην ακρίβεια και τη βελτιστοποίηση στο μέγεθος ενός μοντέλου και το χρόνο απόκρισης κατά τη συμπερασματολογία, σε σχέση με την εφαρμογή ενός μόνο αλγορίθμου συμπίεσης.

Επίσης, για τις μεθόδους κλάδεμα και ομαδοποίηση (που δεν παρουσιάζουν επιτάχυνση), μία ιδέα είναι να πάμε ένα βήμα πιο κοντά στη λειτουργία των μονάδων επεξεργασίας και να δούμε πως ακριβώς γίνεται η υλοποίηση της εκτέλεσης ενός μοντέλου νευρωνικών δικτύων, ώστε να καταλάβουμε τους λόγους για τους οποίους δεν έχουμε επιτάχυνση και τι θα μπορούσαμε να κάνουμε - αλλάξουμε ώστε να έχουμε. Για παράδειγμα, για τον πολλαπλασιασμό αραιών πινάκων που έχουν προκύψει από το κλάδεμα, μπορούν να χρησιμοποιηθούν αλγόριθμοι πολλαπλασιασμού αραιού πίνακα, όπως οι αλγόριθμοι συμπιεσμένης αραιής γραμμής (compressed sparse row - CSR) ή συμπιεσμένης αραιής στήλης (compressed sparse column - CSC). Αντίστοιχα, για τα ομαδοποιημένα μοντέλα η πιο γνωστή μέθοδος επιτάχυνσης είναι η κβαντοποίηση, που εξετάστηκε σε κάποιο βαθμό και στην παρούσα διπλωματική.

Ως προς το κομμάτι της μνήμης μιας συσκευής την οποία καταλαμβάνει ένα μοντέλο, ειδικά για τη μέθοδο της ομαδοποίησης, θα ήταν ενδιαφέρον να δούμε αν υπάρχει κάποια διαφορά σε σχέση με το μέγεθος.



# Παράρτημα Α

**Παρουσίαση των αποτελεσμάτων για τους αλγορίθμους Κβαντοποίησης στο σύνολο δεδομένων ImageNet.**

## Ακρίβεια Μοντέλων:

Στον παρακάτω πίνακα φαίνεται η ακρίβεια των μοντέλων, πριν και μετά την εφαρμογή καθενός από τους αλγορίθμους κβαντοποίησης, που παρουσιάστηκαν σε προηγούμενο Κεφάλαιο.

Μοντέλα	Ακρίβεια (%)			
	Αρχική	DR	FF	INT8
DenseNet121	75.0	73.148	65.632	66.54
DenseNet169	76.2	74.966	70.956	71.74
DenseNet201	77.3	75.968	65.214	70.29
InceptionResNetV2	80.3	80.336	73.15	75.30
InceptionV3	77.9	77.862	77.812	77.89
MobileNetV2	71.3	71.648	71.324	71.26
NASNetLarge	82.5	82.524	82.426	82.42
NASNetMobile	74.4	74.296	66.07	3.55
ResNet152V2	78.0	72.64	72.054	-
ResNet101V2	77.2	72.326	71.684	-
ResNet50V2	76.0	70.01	69.408	-
Xception	79.0	79.104	64.37	64.53
EfficientNetB0-Lite	75.1	75.118	75.068	75.058
EfficientNetB2-Lite	77.6	77.604	77.49	77.508
EfficientNetB4-Lite	81.5	80.784	80.734	80.606

Πίνακας 5: Ακρίβεια Μοντέλων (ImageNet)

## Μέγεθος Μοντέλων:

Στον παρακάτω πίνακα φαίνεται το μέγεθος των μοντέλων, πριν και μετά την εφαρμογή καθενός από τους αλγορίθμους κβαντοποίησης, που παρουσιάστηκαν σε προηγούμενο Κεφάλαιο.

Μοντέλα	Μέγεθος (MB)			
	Αρχικό	DR	FF	INT
DenseNet121	31.24	8.24	8.18	8.19
DenseNet169	55.37	14.56	14.36	14.36
DenseNet201	78.30	20.54	20.19	20.20
InceptionResNetV2	218.18	55.79	56.70	56.70
InceptionV3	93.09	23.57	23.78	23.78
MobileNetV2	13.66	3.70	3.90	3.90
NASNetLarge	331.00	86.82	91.78	91.79
NASNetMobile	20.81	5.91	6.32	6.32
ResNet152V2	235.35	60.34	60.96	-
ResNet101V2	174.16	44.58	45.02	-
ResNet50V2	99.89	25.49	25.72	-
Xception	89.23	23.03	23.70	23.70
EfficientNetB0-Lite	18.12	5.04	5.12	5.12
EfficientNetB2-Lite	23.73	6.63	6.74	6.74
EfficientNetB4-Lite	50.65	13.98	14.20	14.20

Πίνακας 6: Μέγεθος Μοντέλων (ImageNet)

### Χρόνος Απόκρισης Μοντέλων:

Στους παρακάτω πίνακες φαίνονται τα στατιστικά μεγέθη που υπολογίστηκαν με βάση τις μετρήσεις που έγιναν για το χρόνο απόκρισης, για κάθε ένα από τα μοντέλα που φαίνονται παρακάτω.

Μοντέλα	Ελάχιστη Τιμή		Μέγιστη Τιμή		Μέση Τιμή		Τυπική Απόκλιση		90° Εκατοστημόριο	
	1	2	1	2	1	2	1	2	1	2
Threads										
MobileNetV2	15.0	12.0	21.0	18.0	15.93	13.35	0.87	0.88	16.0	14.0
InceptionV3	228.0	164.0	229.0	176.0	228.66	170.5	0.47	2.66	229.0	174.0
EfficientNetLiteB0	21.0	16.0	23.0	20.0	21.2	18.01	0.42	0.57	22.0	19.0
EfficientNetLiteB4	126.0	90.0	128.0	95.0	127.25	92.35	0.48	1.31	128.0	94.0
NASNetLarge	1336.0	1018.0	1519.0	1081.0	1390.0	1050.38	66.99	12.69	1503.0	1067.0

Πίνακας 7: Χρόνος Απόκρισης (ms) - CPU (ImageNet)

Μοντέλα	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση	90° Εκατοστημόριο
MobileNetV2	9.0	12.0	9.86	0.69	11.0

InceptionV3	47.0	66.0	61.44	4.77	65.0
EfficientNetLiteB0	8.0	15.0	11.29	0.9	12.0
EfficientNetLiteB4	28.0	45.0	39.24	4.35	43.0
NASNetLarge	195.0	213.0	203.9	3.2	206.0

Πίνακας 8: Χρόνος Απόκρισης (ms) - GPU (ImageNet)

Μοντέλα	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση	90° Εκατοστημόριο
MobileNetV2	21.0	29.0	22.51	2.04	27.0
InceptionV3	127.0	197.0	160.18	16.73	174.0
EfficientNetLiteB0	27.0	29.0	27.18	0.41	28.0
EfficientNetLiteB4	148.0	150.0	148.76	0.53	149.0
NASNetLarge	1127.0	1160.0	1144.99	7.1	1153.0

Πίνακας 9: Χρόνος Απόκρισης (ms) - NNAPI (ImageNet)

## Κβαντοποίηση με Δυναμικό Εύρος

Μοντέλα	Ελάχιστη Τιμή		Μέγιστη Τιμή		Μέση Τιμή		Τυπική Απόκλιση		90° Εκατοστημόριο	
	1	2	1	2	1	2	1	2	1	2
Threads										
MobileNetV2	12.0	17.0	20.0	21.0	13.48	19.36	1.36	1.02	14.0	21.0
InceptionV3	78.0	57.0	81.0	71.0	79.36	65.96	0.66	2.9	80.0	69.0
EfficientNetLiteB0	15.0	20.0	22.0	32.0	16.12	23.49	0.71	1.62	16.0	25.0
EfficientNetLiteB4	79.0	73.0	82.0	88.0	80.12	80.27	0.59	3.93	81.0	85.0
NASNetLarge	484.0	367.0	487.0	395.0	485.04	382.25	0.68	6.91	486.0	388.0

Πίνακας 10: Χρόνος Απόκρισης (ms) – Dynamic Range Quantization - CPU (ImageNet)

Μοντέλα	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση	90° Εκατοστημόριο
MobileNetV2	9.0	12.0	9.67	0.69	10.0
InceptionV3	48.0	67.0	61.06	4.57	64.0
EfficientNetLiteB0	10.0	14.0	11.38	0.82	13.0

EfficientNetLiteB4	29.0	46.0	39.72	4.15	43.0
NASNetLarge	194.0	234.0	203.64	5.6	206.0

Πίνακας 11: Χρόνος Απόκρισης (ms) - Dynamic Range Quantization - GPU (ImageNet)

Μοντέλα	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση	90° Εκατοστημόριο
MobileNetV2	31.0	44.0	34.3	2.1	37.0
InceptionV3	102.0	113.0	106.48	2.41	109.0
EfficientNetLiteB0	33.0	48.0	39.83	2.71	43.0
EfficientNetLiteB4	94.0	109.0	101.17	2.73	104.0
NASNetLarge	536.0	546.0	539.69	1.96	542.0

Πίνακας 12: Χρόνος Απόκρισης (ms) - Dynamic Range Quantization - NNAPI (ImageNet)

## Κβαντοποίηση με “Πτώση” σε Αριθμούς Κινητής Υποδιαστολής

Μοντέλα	Ελάχιστη Τιμή		Μέγιστη Τιμή		Μέση Τιμή		Τυπική Απόκλιση		90° Εκατοστημόριο	
	1	2	1	2	1	2	1	2	1	2
Threads										
MobileNetV2	11.0	14.0	16.0	19.0	11.5	16.45	0.77	0.98	12.0	18.0
InceptionV3	73.0	49.0	75.0	67.0	74.31	60.69	0.48	3.54	75.0	64.0
EfficientNetLiteB0	14.0	15.0	18.0	24.0	14.93	19.1	0.78	1.47	15.0	21.0
EfficientNetLiteB4	64.0	47.0	66.0	70.0	64.58	63.7	0.51	3.47	65.0	68.0
NASNetLarge	503.0	352.0	506.0	369.0	504.25	376.42	0.54	12.42	505.0	389.0

Πίνακας 13: Χρόνος Απόκρισης (ms) – Float Fallback Quantization - CPU (ImageNet)

Μοντέλα	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση	90° Εκατοστημόριο
MobileNetV2	9.0	13.0	10.54	0.81	11.0
InceptionV3	48.0	79.0	62.05	5.19	66.0
EfficientNetLiteB0	11.0	15.0	12.15	0.73	13.0
EfficientNetLiteB4	28.0	46.0	40.69	4.22	44.0
NASNetLarge	199.0	235.0	208.96	4.36	212.0



Πίνακας 14: Χρόνος Απόκρισης (ms) - Float Fallback Quantization - GPU (ImageNet)

Μοντέλα	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση	90° Εκατοστημόριο
MobileNetV2	11.0	16.0	11.43	0.68	12.0
InceptionV3	73.0	75.0	74.05	0.43	75.0
EfficientNetLiteB0	14.0	20.0	14.76	0.69	15.0
EfficientNetLiteB4	64.0	66.0	64.69	0.48	65.0
NASNetLarge	504.0	507.0	504.73	0.56	505.0

Πίνακας 15: Χρόνος Απόκρισης (ms) - Float Fallback Quantization - NNAPI (ImageNet)

## Πλήρης Κβαντοποίηση

Μοντέλα	Ελάχιστη Τιμή		Μέγιστη Τιμή		Μέση Τιμή		Τυπική Απόκλιση		90° Εκατοστημόριο	
	1	2	1	2	1	2	1	2	1	2
Threads										
MobileNetV2	11.0	11.0	16.0	24.0	11.3	12.94	0.85	2.08	12.0	14.0
InceptionV3	74.0	49.0	75.0	67.0	74.37	62.6	0.48	2.56	75.0	65.0
EfficientNetLiteB0	14.0	14.0	16.0	29.0	14.49	16.7	0.52	2.48	15.0	17.0
EfficientNetLiteB4	64.0	48.0	65.0	68.0	64.33	62.94	0.47	3.06	65.0	66.0
NASNetLarge	504.0	353.0	506.0	392.0	504.85	378.69	0.43	10.48	505.0	388.0

Πίνακας 16: Χρόνος Απόκρισης (ms) - Full Integer - CPU (ImageNet)

Μοντέλα	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση	90° Εκατοστημόριο
MobileNetV2	9.0	21.0	10.54	2.04	11.0
InceptionV3	47.0	73.0	61.1	5.71	65.0
EfficientNetLiteB0	11.0	15.0	12.0	0.69	13.0
EfficientNetLiteB4	29.0	52.0	40.89	4.55	45.0
NASNetLarge	196.0	227.0	208.32	4.16	211.0

Πίνακας 17: Χρόνος Απόκρισης (ms) - Full Integer - GPU (ImageNet)

Μοντέλα	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση	90° Εκατοστημόριο
MobileNetV2	19.0	34.0	27.22	1.35	28.0
InceptionV3	3134.0	3221.0	3200.59	11.04	3210.0
EfficientNetLiteB0	31.0	37.0	33.61	0.84	34.0
EfficientNetLiteB4	73.0	76.0	74.7	0.64	75.0
NASNetLarge	520.0	528.0	522.96	1.06	524.0

Πίνακας 18: Χρόνος Απόκρισης (ms) - Full Integer - NNAPI (ImageNet)

# Παράρτημα Β

Παρουσίαση των αποτελεσμάτων για τους αλγορίθμους Κλαδέματος και Ομαδοποίησης για το σύνολο δεδομένων CIFAR10.

Ακρίβεια Μοντέλων:

Στον παρακάτω πίνακα, φαίνεται η ακρίβεια των μοντέλων μετά την εκπαίδευση με μεταφορά γνώσης στο σύνολο δεδομένων CIFAR10 και την εφαρμογή των αλγορίθμων κλαδέματος και ομαδοποίησης. Οι επιτρεπτές ακρίβειες έχουν σημειωθεί με πορτοκαλί χρώμα.

Μοντέλα	Σχήμα Εισόδου	Τελική Ακρίβεια (%)	Κλάδεμα - Ακρίβεια (%)				Ομαδοποίηση - Ακρίβεια (%)							
			Αραιότητα (%)				Γραμμική				KMEANS++			
			80	90-80	95	93	Χωρίς Εκπαίδευση		Με Εκπαίδευση		Χωρίς Εκπαίδευση		Με Εκπαίδευση	
						20	30	20	30	20	30	20	30	
MobileNetV2	224	94.55	-	-	-	-	12.06	41.21	93.47	95.09	71.77	89.19	94.89	95.25
InceptionV3	224	96.10	95.95	94.00	-	-	30	50	30	50	30	50	30	50
							89.58	94.20	95.72	95.48	91.38	95.19	96.44	95.50
DenseNet121	224	94.55	95.72	94.59	-	-	16	28	16	28	16	28	16	28
							11.16	88.13	95.08	94.29	86.15	91.59	96.06	94.65
DenseNet169	224	96.57	96.06	95.41	-	-	20	50	20	50	20	50	20	50
							45.97	93.45	95.53	95.55	92.42	94.01	96.50	95.22
DenseNet201	224	96.56	96.31	95.99	-	-	16	36	16	36	16	36	16	36
							10.40	90.25	95.94	96.29	90.65	93.45	96.35	96.17
EfficientNetB0	224	95.13	92.47	84.90	-	-	20	30	20	30	20	30	20	30
							26.50	54.35	93.70	95.44	81.73	91.04	95.33	96.23
EfficientNetB2	224	95.87	88.49	66.43	-	-	16	30	16	30	16	30	16	30
							46.21	90.18	94.61	96.48	88.77	94.71	95.80	96.27
EfficientNetB4	380	94.97	96.12	93.53	-	-	16	30	16	30	16	30	16	30
							32.73	76.53	96.73	96.40	63.50	85.86	96.74	96.44
EfficientNetB6	528	96.47	96.53	95.54	-	-	16	30	16	30	16	30	16	30
							49.48	94.98	96.27	96.86	94.08	95.97	96.78	96.50

<b>InceptionResNet V2</b>	299	94.88	96.16	95.33	-	93.72	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>
							10.00	10.00	-	-	91.39	89.97	96.72	96.61
<b>NASNetMobile</b>	224	95.19	-	-	-	-	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>
							11.76	77.26	95.48	96.14	74.53	86.44	95.79	96.23
<b>NASNetLarge</b>	331	95.96	96.91	95.24	-	-	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>
							10.00	95.37	96.56	96.94	94.36	95.72	97.10	96.83
<b>ResNet101</b>	224	93.89	95.53	92.89	-	-	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>
							10.00	10.09	82.93	94.67	14.32	93.12	96.20	96.36
<b>ResNet152</b>	224	92.59	95.91	95.52	93.38	-	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>
							9.87	9.64	95.40	95.35	25.61	90.82	95.80	96.42
<b>ResNet50</b>	224	92.50	94.88	91.45	-	-	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>
							10.0	17.69	93.68	95.57	82.55	88.44	93.01	95.43
<b>ResNet152V2</b>	224	91.03	95.68	94.91	92.14	-	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>
							10.00	8.92	-	-	90.60	90.36	95.29	95.92
<b>ResNet101V2</b>	224	91.33	95.55	95.17	94.53	-	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>
							10.00	10.00	74.12	-	89.81	91.31	95.89	96.32
<b>ResNet50V2</b>	224	91.85	95.01	93.91	-	91.30	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>
							15.12	85.36	94.84	95.06	86.33	91.38	95.65	95.12
<b>VGG16</b>	224	85.72	90.36	89.51	-	-	<b>26</b>	<b>32</b>	<b>26</b>	<b>32</b>	<b>26</b>	<b>32</b>	<b>26</b>	<b>32</b>
							82.69	84.00	89.92	92.29	85.04	85.05	91.69	92.69
<b>VGG19</b>	224	87.93	89.52	80.90	-	-	<b>20</b>	<b>28</b>	<b>20</b>	<b>28</b>	<b>20</b>	<b>28</b>	<b>20</b>	<b>28</b>
							32.41	84.99	92.75	92.80	86.45	87.18	93.09	93.29
<b>Xception</b>	224	96.53	95.92	94.17	-	-	<b>20</b>	<b>38</b>	<b>20</b>	<b>38</b>	<b>20</b>	<b>38</b>	<b>20</b>	<b>38</b>
							9.78	92.77	96.16	96.72	91.38	95.71	96.08	96.57

Πίνακας 19: Ακρίβεια Μοντέλων (CIFAR10)

### Μέγεθος Μοντέλων:

Στον παρακάτω πίνακα, φαίνεται η συμπίεση των μοντέλων (σε TFLite μορφή) ως προς τη χωρητικότητά τους, μετά την εκπαίδευση με μεταφορά γνώσης στο σύνολο δεδομένων CIFAR10 και την εφαρμογή των αλγορίθμων κλαδέματος και ομαδοποίησης.

Μοντέλα	Σχήμα Εισόδου	Μέγεθος(MB)		Κλάδεμα - Μέγεθος (MB)				Ομαδοποίηση - Μέγεθος (MB)											
								Αραιότητα (%)				Γραμμική				KMEANS++			
								TFLITE	TFLITE + ZIP	80	90-80	95	93	Χωρίς Εκπαίδευση		Με Εκπαίδευση		Χωρίς Εκπαίδευση	
<b>MobileNetV2</b>	224	8.71	8.07	-	-	-	-	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>				
								-	-	1.96	2.29	-	-	2.79	3.20				
<b>InceptionV3</b>	224	85.17	79.04	24.51		-	-	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>				
								-	-	11.17	14.64	-	25.90	21.61	25.65				
<b>DenseNet121</b>	224	27.29	25.46	8.10	4.31	-	-	<b>16</b>	<b>28</b>	<b>16</b>	<b>28</b>	<b>16</b>	<b>28</b>	<b>16</b>	<b>28</b>				
								-	-	3.38	4.49	-	-	5.70	6.92				
<b>DenseNet169</b>	224	48.95	45.91	14.70	-	-	-	<b>20</b>	<b>50</b>	<b>20</b>	<b>50</b>	<b>20</b>	<b>50</b>	<b>20</b>	<b>50</b>				
								-	-	6.63	10.04	-	-	10.73	14.10				
<b>DenseNet201</b>	224	70.89	66.56	21.40	11.31	-	-	<b>16</b>	<b>36</b>	<b>16</b>	<b>36</b>	<b>16</b>	<b>36</b>	<b>16</b>	<b>36</b>				
								-	-	8.22	12.35	-	-	14.08	18.30				
<b>EfficientNetB0</b>	224	15.72	14.56	-	-	-	-	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>				
								-	-	-	4.00	-	-	4.83	5.46				
<b>EfficientNetB2</b>	224	30.14	27.94	-	-	-	-	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>				
								-	-	5.46	6.99	-	9.96	8.17	9.95				
<b>EfficientNetB4</b>	380	68.56	63.67	22.05	-	-	-	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>				
								-	-	12.58	16.08	-	-	17.85	21.63				
<b>EfficientNetB6</b>	528	159.03	147.87	50.02	27.57	-	-	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>				
								-	-	23.34	28.76	-	-	33.52	38.83				
<b>InceptionResNet V2</b>	299	212.25	198.00	61.56	31.23	-	-	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>				
								-	-	-	-	-	-	43.83	49.76				
<b>NASNetMobile</b>	224	16.74	15.42	-	-	-	-	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>				
								-	-	3.72	4.68	-	-	5.35	6.49				
<b>NASNetLarge</b>	331	331.00	307.48	103.83	55.24	-	-	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>	<b>16</b>	<b>30</b>				
								-	44.61	31.98	44.25	-	76.50	65.18	-				
<b>ResNet101</b>	224	165.98	155.00	46.39	23.48	-	-	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>	<b>30</b>	<b>50</b>				
								-	-	-	29.53	-	51.86	41.93	49.52				
<b>ResNet152</b>	224	227.05	211.14	62.68	31.73	16.28	-	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>				
								-	-	31.10	35.57	-	-	55.08	-				
<b>ResNet50</b>	224	91.85	85.37	26.65	-	-	-	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>	<b>20</b>	<b>30</b>				
								-	-	8.60	11.06	-	-	20.39	23.26				
<b>ResNet152V2</b>	224	227.43	211.22	63.05	32.10	16.64	-	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>	<b>26</b>	<b>36</b>				
								-	-	-	-	50.24	55.87	46.80	53.32				

Model	Threads	Min	Q1	Q2	Q3	Max	Std	30	50	30	50	30	50	30	50
								Min	Q1	Q2	Q3	Max	Min	Q1	Q2
ResNet101V2	224	166.25	154.44	46.62	23.71	12.28	-	30	50	30	50	30	50	30	50
								-	-	-	-	-	44.82	37.72	43.27
ResNet50V2	224	91.98	85.44	26.75	13.60	-	9.65	20	30	20	30	20	30	20	30
								-	-	7.60	10.42	-	20.90	18.03	20.39
VGG16	224	57.51	53.42	16.84	8.50	-	-	26	32	26	32	26	32	26	32
								-	-	4.76	5.63	11.38	12.04	10.98	11.75
VGG19	224	78.25	72.69	23.00	-	-	-	20	28	20	28	20	28	20	28
								-	-	4.95	6.67	14.40	15.80	13.89	15.35
Xception	224	81.31	75.56	25.08	-	-	-	20	38	20	38	20	38	20	38
								-	-	10.85	14.84	-	24.93	19.82	24.82

Πίνακας 20: Μέγεθος Μοντέλων (CIFAR10)

### Χρόνος Απόκρισης Μοντέλων:

Στους παρακάτω πίνακες φαίνονται τα στατιστικά μεγέθη που υπολογίστηκαν με βάση τις μετρήσεις που έγιναν για το χρόνο απόκρισης, για κάθε ένα από τα μοντέλα που φαίνονται παρακάτω.

Models	Min		Max		Mean		STD		90 <sup>th</sup> Percentile	
	1	2	1	2	1	2	1	2	1	2
MobileNetV2	15	12	22	14	15.77	12.82	0.88	0.75	16	14
InceptionV3	114	79	118	122	114.65	82.52	0.73	4.22	115	84
DenseNet121	127	92	131	101	127.87	95.44	0.64	1.47	128	97
DenseNet169	151.0	113.0	154.0	121.0	151.66	116.62	0.6	1.46	152	118
DenseNet201	193.0	148.0	197.0	154.0	194.3	151.58	0.61	1.23	195.0	153
EfficientNetB0	33.0	28.0	36.0	33.0	33.49	30.9	0.64	0.78	34.0	32.0
EfficientNetB2	52.0	38.0	55.0	46.0	53.04	41.32	0.53	1.62	53.0	43.0
EfficientNetB4	313.0	223.0	326.0	242.0	314.54	229.41	2.26	3.04	317.0	232.0
InceptionResNetV2	534.0	370.0	542.0	401.0	537.4	382.12	2.45	5.68	541.0	389.0
NASNetMobile	57.0	37.0	64.0	50.0	57.9	41.97	1.2	4.19	58.0	48.0
NASNetLarge	1295.0	1020.0	1445.0	1063.0	1322.01	1043.18	20.36	8.03	1330.0	1054.0
ResNet101	313.0	237.0	316.0	253.0	313.28	241.41	0.53	2.72	314.0	244.0
ResNet152	463.0	352.0	481.0	380.0	467.46	358.58	2.88	4.58	469.0	363.0
ResNet50	163.0	123.0	167.0	128.0	163.96	125.1	0.68	1.28	164.0	127.0
ResNet152V2	456.0	348.0	488.0	367.0	459.75	354.03	4.34	3.19	461.0	357.0
ResNet101V2	303.0	230.0	307.0	241.0	303.76	236.16	0.65	1.83	304.0	238.0
ResNet50V2	152.0	114.0	154.0	172.0	152.37	117.31	0.54	5.65	153.0	118.0

VGG16	631.0	487.0	655.0	518.0	639.52	501.72	4.14	7.34	644.0	512.0
VGG19	815.0	636.0	869.0	677.0	829.21	658.41	9.77	8.05	837.0	670.0
Xception	202.0	138.0	208.0	149.0	202.59	141.95	1.21	1.79	203.0	144.0

Πίνακας 21: Χρόνος Απόκρισης (ms) - CPU (CIFAR10)

Models	Min	Max	Mean	STD	90 <sup>th</sup> Percentile
MobileNetV2	7.0	11.0	9.48	0.64	10.0
InceptionV3	31.0	53.0	41.61	7.17	48.0
DenseNet121	37.0	60.0	52.9	7.01	58.0
DenseNet169	53.0	82.0	68.14	7.99	74.0
DenseNet201	70.0	95.0	85.26	7.15	90.0
EfficientNetB0	14.0	20.0	15.06	1.01	16.0
EfficientNetB2	15.0	30.0	21.93	2.08	24.0
EfficientNetB4	52.0	82.0	66.68	6.39	75.0
InceptionResNetV2	99.0	117.0	111.31	5.08	116.0
NASNetMobile	17.0	31.0	25.67	3.81	28.0
NASNetLarge	193.0	231.0	203.54	4.02	205.0
ResNet101	56.0	98.0	70.88	6.86	76.0
ResNet152	81.0	110.0	95.62	7.56	102.0
ResNet50	30.0	49.0	42.03	6.74	48.0
ResNet152V2	82.0	106.0	96.68	6.54	102.0
ResNet101V2	56.0	80.0	70.62	6.9	76.0
ResNet50V2	29.0	67.0	41.56	7.16	47.0
VGG16	37.0	87.0	56.19	14.93	82.0
VGG19	45.0	99.0	67.34	17.84	92.0
Xception	35.0	77.0	47.49	7.07	53.0

Πίνακας 22: Χρόνος Απόκρισης (ms) - GPU (CIFAR10)

<b>Models</b>	<b>Min</b>	<b>Max</b>	<b>Mean</b>	<b>STD</b>	<b>90<sup>th</sup> Percentile</b>
MobileNetV2	21.0	25.0	21.23	0.6	22.0
InceptionV3	92.0	140.0	113.7	10.48	136.0
DenseNet121	147.0	153.0	147.77	1.03	148.0
DenseNet169	174.0	177.0	175.31	0.56	176.0
DenseNet201	220.0	224.0	221.05	0.65	222.0
EfficientNetB0	52.0	61.0	54.14	1.71	55.0
EfficientNetB2	69.0	71.0	69.92	0.5	70.0
EfficientNetB4	405.0	414.0	407.74	2.24	441.0
InceptionResNetV2	270.0	329.0	299.99	14.64	313.0
NASNetMobile	48.0	52.0	49.11	0.49	50.0
NASNetLarge	1121.0	1148.0	1130.82	6.82	1139.0
ResNet101	323.0	327.0	324.44	0.62	325.0
ResNet152	471.0	475.0	472.02	0.66	473.0
ResNet50	175.0	204.0	176.44	3.21	177.0
ResNet152V2	462.0	466.0	462.94	0.75	464.0
ResNet101V2	312.0	318.0	313.26	0.78	314.0
ResNet50V2	162.0	167.0	162.8	0.63	163.0
VGG16	144.0	258.0	228.48	34.98	254.0
VGG19	181.0	295.0	256.81	39.32	291.0
Xception	79.0	150.0	108.4	20.14	130.0

Πίνακας 23: Χρόνος Απόκρισης (ms) - NNAPI (CIFAR10)

## Κλάδεμα

<b>Models</b>	<b>Min</b>		<b>Max</b>		<b>Mean</b>		<b>STD</b>		<b>90<sup>th</sup> Percentile</b>	
	<b>1</b>	<b>2</b>	<b>1</b>	<b>2</b>	<b>1</b>	<b>2</b>	<b>1</b>	<b>2</b>	<b>1</b>	<b>2</b>
ResNet101V2	304.0	231.0	308.0	241.0	304.49	235.77	0.77	1.86	305.0	237.0
NASNetLarge	1295.0	1008.0	1388.0	1051.0	1318.82	1027.31	17.71	9.39	1326.0	1040.0

Πίνακας 24: Χρόνος Απόκρισης (ms) - Κλάδεμα - CPU (CIFAR10)



Models	Min	Max	Mean	STD	90 <sup>th</sup> Percentile
ResNet101V2	55.0	98.0	71.64	6.6	75.0
NASNetLarge	193.0	212.0	203.28	3.68	206.0

Πίνακας 25: Χρόνος Απόκρισης (ms) - Κλάδεμα - GPU (CIFAR10)

Models	Min	Max	Mean	STD	90 <sup>th</sup> Percentile
ResNet101V2	298.0	304.0	299.49	0.93	300.0
NASNetLarge	1096.0	1134.0	1105.51	6.92	1114.0

Πίνακας 26: Χρόνος Απόκρισης (ms) - Κλάδεμα - NNAPI (CIFAR10)

## Ομαδοποίηση

Models	Min		Max		Mean		STD		90 <sup>th</sup> Percentile	
	1	2	1	2	1	2	1	2	1	2
MobileNetV2	12.0	16.0	19.0	21.0	13.23	17.97	1.25	1.01	14.0	19.0
InceptionV3	41.0	37.0	46.0	44.0	41.79	41.56	0.75	1.17	42.0	43.0
DenseNet121	58.0	53.0	63.0	57.0	59.11	55.43	0.6	0.89	60.0	57.0
DenseNet169	68.0	60.0	71.0	75.0	68.95	65.29	0.61	2.06	70.0	67.0
DenseNet201	84.0	73.0	91.0	85.0	85.0	79.04	1.01	2.63	86.0	83.0
EfficientNetB0	28.0	35.0	36.0	42.0	29.25	36.92	1.45	1.12	30.0	38.0
EfficientNetB2	42.0	41.0	47.0	53.0	42.78	49.62	0.82	1.82	43.0	51.0
EfficientNetB4	227.0	221.0	229.0	241.0	228.16	231.23	0.46	3.94	229.0	236.0
InceptionResNetV2	180.0	126.0	182.0	137.0	180.53	129.91	0.56	3.0	181.0	134.0
NASNetMobile	34.0	44.0	40.0	57.0	34.71	50.78	1.17	3.09	35.0	55.0
NASNetLarge	484.0	366.0	487.0	394.0	485.16	381.9	0.66	7.33	486.0	389.0
ResNet101	101.0	70.0	107.0	87.0	101.73	80.7	0.81	2.78	102.0	84.0
ResNet152	146.0	99.0	153.0	120.0	148.8	109.52	2.39	2.78	152.0	113.0
ResNet50	58.0	49.0	61.0	71.0	58.94	53.3	0.66	2.48	60.0	55.0
ResNet152V2	147.0	108.0	150.0	119.0	147.95	112.69	0.52	2.28	148.0	116.0
ResNet101V2	102.0	70.0	104.0	87.0	102.27	81.3	0.55	2.93	103.0	84.0
ResNet50V2	56.0	49.0	59.0	54.0	57.2	51.53	0.47	1.2	58.0	53.0
VGG16	167.0	106.0	171.0	112.0	168.13	108.54	0.58	1.0	169.0	110.0
VGG19	205.0	130.0	212.0	134.0	206.24	131.51	0.97	0.84	207.0	132.0

Xception	73.0	54.0	77.0	73.0	73.78	66.19	0.78	2.5	74.0	69.0
----------	------	------	------	------	-------	-------	------	-----	------	------

Πίνακας 27: Χρόνος Απόκρισης (ms) - Ομαδοποίηση - CPU (CIFAR10)

Models	Min	Max	Mean	STD	90 <sup>th</sup> Percentile
MobileNetV2	8.0	12.0	9.56	0.68	10.0
InceptionV3	29.0	52.0	42.14	6.93	49.0
DenseNet121	37.0	87.0	54.4	9.74	59.0
DenseNet169	54.0	111.0	70.68	9.56	77.0
DenseNet201	69.0	121.0	86.76	10.07	93.0
EfficientNetB0	14.0	19.0	15.06	0.9	16.0
EfficientNetB2	15.0	32.0	21.63	2.72	24.0
EfficientNetB4	53.0	72.0	62.13	4.34	65.0
InceptionResNetV2	96.0	123.0	112.36	4.82	116.0
NASNetMobile	18.0	32.0	26.27	3.42	29.0
NASNetLarge	194.0	264.0	204.62	7.65	209.0
ResNet101	55.0	111.0	70.29	8.53	76.0
ResNet152	80.0	104.0	95.25	7.37	102.0
ResNet50	30.0	49.0	42.75	6.37	48.0
ResNet152V2	82.0	115.0	97.7	6.28	103.0
ResNet101V2	57.0	78.0	71.73	6.0	76.0
ResNet50V2	29.0	50.0	40.06	6.96	46.0
VGG16	37.0	87.0	57.24	16.05	83.0
VGG19	44.0	96.0	67.0	16.2	90.0
Xception	35.0	58.0	46.88	6.43	52.0

Πίνακας 28: Χρόνος Απόκρισης (ms) - Ομαδοποίηση - GPU (CIFAR10)

Models	Min	Max	Mean	STD	90 <sup>th</sup> Percentile
MobileNetV2	30.0	42.0	35.01	2.83	40.0
InceptionV3	58.0	72.0	64.2	2.67	67.0
DenseNet121	82.0	96.0	89.32	3.01	94.0
DenseNet169	93.0	110.0	99.37	3.35	104.0
DenseNet201	110.0	127.0	116.41	3.09	120.0
EfficientNetB0	58.0	82.0	64.7	3.69	68.0

EfficientNetB2	64.0	83.0	72.77	4.24	78.0
EfficientNetB4	388.0	437.0	392.84	5.2	396.0
InceptionResNetV2	202.0	215.0	208.99	3.0	213.0
NASNetMobile	50.0	77.0	60.89	5.22	66.0
NASNetLarge	536.0	585.0	541.74	4.84	544.0
ResNet101	128.0	160.0	141.62	5.2	147.0
ResNet152	174.0	195.0	185.88	4.37	191.0
ResNet50	83.0	114.0	100.18	6.21	107.0
ResNet152V2	183.0	237.0	193.9	9.01	197.0
ResNet101V2	135.0	154.0	145.52	4.23	151.0
ResNet50V2	88.0	131.0	108.7	10.7	124.0
VGG16	204.0	236.0	212.44	4.94	217.0
VGG19	241.0	275.0	250.9	5.26	256.0
Xception	88.0	98.0	90.89	1.89	93.0

Πίνακας 29: Χρόνος Απόκρισης (ms) - Ομαδοποίηση - NNAPI (CIFAR10)



# Ξενόγλωσση Ορολογία

Τεχνητά Νευρωνικά Δίκτυα	Artificial Neural Networks
Βαθιά Μάθηση	Deep Learning
Μηχανική Μάθηση	Machine Learning
Νευρώνας	Neuron
Βάρος	Weight
Μεροληψία	Bias
Στρώμα/Επίπεδο	Layer
Συνάρτηση Ενεργοποίησης	Activation Function
Εκπαίδευση	Training
Υπερπροσαρμογή	Overfitting
Υποπροσαρμογή	Underfitting
Ομαλή Κάθοδος	Gradient Descent
Ρυθμός Εκμάθησης	Learning Rate
Στοχαστική Ομαλή Κάθοδος	Stochastic Gradient Descent
Δέσμη	Batch
Μέγεθος Δέσμης	Batch Size
Συνέλιξη	Convolution
Συνελικτικά Δίκτυα	Convolutional Networks
Φίλτρο	Filter
Βήμα	Stride
Παραγέμισμα με μηδενικά	Zero-Padding
Χάρτης Ενεργοποίησης	Activation Map
Διαστολή	Dilation
Κατά Βάθος Συνέλιξη	Depthwise Convolution
Διαχωρίσιμη Συνέλιξη	Separable Convolution
Κατά Βάθος Διαχωρίσιμη Συνέλιξη	Depthwise Separable Convolution
Αλγόριθμοι Συμπίεσης	Compression Algorithms
Κβαντοποίηση	Quantization
Εύρος	Range

Κβαντοποίηση με Δυναμικό Εύρος	Dynamic Range Quantization
Αριθμός Κινητής Υποδιαστολής	Floating Point Number
Κβαντοποίηση με “Πτώση” σε Αριθμούς Κινητής Υποδιαστολής	Float Fallback Quantization
Κλάδεμα	Pruning
Δομικό Κλάδεμα	Structured Pruning
Μη Δομικό Κλάδεμα	Unstructured Pruning
Ομαδοποίηση	Clustering
Ακέραιος	Integer
Πλήρης Κβαντοποίηση	Full Integer Quantization
Μεταφορά Γνώσης	Transfer Learning
Πεδίο	Domain
Εποχή	Epoch
Σύνολο Εκπαίδευσης	Training Set
Σύνολο Ελέγχου	Test Set
Σύνολο Αξιολόγησης	Validation Set
Αρχιτεκτονική	Architecture
Κατώφλι	Threshold
Είσοδος	Input
Έξοδος	Output
Υπερεπίπεδο	Hyperplane
Γραμμικά Διαχωρίσιμος	Linearly Separable
Ταξινόμηση	Classification
Κατάτμηση	Segmentation
Υπερπαραμέτρος	Hyperparameter
Βηματική Συνάρτηση	Step Function
Σιγμοειδής Συνάρτηση	Sigmoid Function
Ανορθωμένη Γραμμική Μονάδα	Rectified Linear Unit (ReLU)
Κατανεμημένη Αναπαράσταση	Distributed Representation
Επιβλεπόμενη Μάθηση	Supervised Learning
Μη Επιβλεπόμενη Μάθηση	Unsupervised Learning
Ενισχυτική Μάθηση	Reinforcement Learning
Συμπερασματολογία	Inference

Οπισθοδιάδοση	Backpropagation
Πλάτος	Width
Ύψος	Height
Βάθος	Depth
Τοπική Συνδεσιμότητα	Local Connectivity
Μη γραμμικότητα	Non Linearity
Διασπορά	Variance
Πυρήνας	Kernel
Χωρική Συνέλιξη	Spatial Convolution
Διεσταλμένη Συνέλιξη	Dilated Convolution
Αποσυνέλιξη	Deconvolution
Σημειακή Συνέλιξη	Pointwise Convolution
Συνέλιξη με Βηματισμό	Strided Convolution
Συνέλιξη με Παραγέμισμα με Μηδενικά	Padded Convolution
Ομαδοποιημένη Συνέλιξη	Group Convolution
Χωρικά Διαχωρίσιμη Συνέλιξη	Spatial Separable Convolution
Δραστηριότητα	Activity
Υπηρεσία	Service
Δέκτης Εκομπής	Broadcast Receiver
Πάροχος Περιεχομένου	Content Provider
Σύνολο Δεδομένων	Dataset
Πλήρως Συνδεδεμένο Επίπεδο	Dense Layer
Επίπεδο Κανονικοποίησης	Normalization Layer
Κανονικοποίηση με Δέσμες	Batch Normalization
Καθολική Συγκέντρωση	Global Pooling
Καθολική Μέση Συγκέντρωση	Global Average Pooling





# Βιβλιογραφία

- [1] Ian Goodfellow, Yoshua Bengio and Aaron Courville, “Deep Learning.” [www.deeplearningbook.org/](http://www.deeplearningbook.org/), MIT Press, 2016. Ημερομηνία Πρόσβασης: 19 Ιανουαρίου 2023.
- [2] M. A. Nielsen, “Neural Networks and Deep Learning,” 2015, Available: [neuralnetworksanddeeplearning.com](http://neuralnetworksanddeeplearning.com), Ημερομηνία Πρόσβασης: 19 Ιανουαρίου 2023.
- [3] D. Brown, “What is Supervised Learning? A Mathematical Perspective,” *Dorian Brown*, Sep. 03, 2019. [dorianbrown.dev/what-is-supervised-learning/](http://dorianbrown.dev/what-is-supervised-learning/). Ημερομηνία Πρόσβασης: 19 Ιανουαρίου 2023.
- [4] V. Bushaev, “How do we ‘train’ neural networks?,” *Medium*, Oct. 22, 2018. [towardsdatascience.com/how-do-we-train-neural-networks-edd985562b73](https://towardsdatascience.com/how-do-we-train-neural-networks-edd985562b73). Ημερομηνία Πρόσβασης: 21 Ιανουαρίου 2023.
- [5] J. Brownlee, “A Gentle Introduction to Cross-Entropy for Machine Learning,” *MachineLearningMastery.com*, Oct. 20, 2019. [machinelearningmastery.com/cross-entropy-for-machine-learning/](https://machinelearningmastery.com/cross-entropy-for-machine-learning/). Ημερομηνία Πρόσβασης: 21 Ιανουαρίου 2023.
- [6] S. Doshi, “Various Optimization Algorithms For Training Neural Network,” *Medium*, Aug. 03, 2020. [towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6](https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6) Ημερομηνία Πρόσβασης: 21 Ιανουαρίου 2023.
- [7] S. Kostadinov, “Understanding Backpropagation Algorithm,” *Medium*, Aug. 12, 2019. [towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd](https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd). Ημερομηνία Πρόσβασης: 19 Ιανουαρίου 2023.
- [8] “Activation Functions in Neural Networks [12 Types & Use Cases].” [www.v7labs.com/blog/neural-networks-activation-functions](http://www.v7labs.com/blog/neural-networks-activation-functions), [www.v7labs.com/blog/neural-networks-activation-functions](http://www.v7labs.com/blog/neural-networks-activation-functions). Ημερομηνία Πρόσβασης: 22 Ιανουαρίου 2023.
- [9] S. Sharma, S. Sharma, and A. Athaiya, “ACTIVATION FUNCTIONS IN NEURAL NETWORKS,” *IJEAST*, vol. 04, no. 12, σελίδες: 310–316, May 2020, doi: 10.33564/IJEAST.2020.v04i12.054.
- [10] “CS231n Convolutional Neural Networks for Visual Recognition.” <https://cs231n.github.io/neural-networks-1/>. Ημερομηνία Πρόσβασης: 22 Ιανουαρίου 2023.
- [11] S. Ruder, “An overview of gradient descent optimization algorithms,” *CoRR*, vol. abs/1609.04747, 2016. Available: <http://arxiv.org/abs/1609.04747>. Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.

- [12] “CS231n Convolutional Neural Networks for Visual Recognition.” [cs231n.github.io/convolutional-networks/](https://cs231n.github.io/convolutional-networks/). Ημερομηνία Πρόσβασης: 23 Ιανουαρίου 2023..
- [13] Kom Soo "Deep Learning Bible - 2. Classification - 한글", (2023), WikiDocs
- [14] “CS231n Convolutional Neural Networks for Visual Recognition.” [cs231n.github.io/transfer-learning/](https://cs231n.github.io/transfer-learning/). Ημερομηνία Πρόσβασης: 23 Ιανουαρίου 2023.
- [15] D. (DJ) Sarkar, “A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning,” *Medium*, Nov. 17, 2018. [towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a](https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a). Ημερομηνία Πρόσβασης: 23 Ιανουαρίου 2023.
- [16] D. W. Blalock, J. J. G. Ortiz, J. Frankle, and J. V. Gutttag, “What is the State of Neural Network Pruning?,” in *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*, 2020. [Online]. Available: [proceedings.mlsys.org/book/296.pdf](https://proceedings.mlsys.org/book/296.pdf). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.
- [17] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, “Pruning and quantization for deep neural network acceleration: A survey,” *Neurocomputing*, vol. 461, σελίδες: 370–403, 2021, doi: 10.1016/j.neucom.2021.07.045.
- [18] neuralmagic, “Part 1: What is Pruning in Machine Learning? - Neural Magic,” Jul. 31, 2020. [neuralmagic.com/blog/pruning-overview/](https://neuralmagic.com/blog/pruning-overview/). Ημερομηνία Πρόσβασης: 23 Ιανουαρίου 2023.
- [19] K. Neklyudov, D. Molchanov, A. Ashukha, and D. P. Vetrov, “Structured Bayesian Pruning via Log-Normal Multiplicative Noise,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017*, pp. 6775–6784. Available: [proceedings.neurips.cc/paper/2017/hash/dab49080d80c724aad5ebf158d63df41-Abstract.html](https://proceedings.neurips.cc/paper/2017/hash/dab49080d80c724aad5ebf158d63df41-Abstract.html). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.
- [20] “Understand and Implement the Backpropagation Algorithm From Scratch In Python,” *A Developer Diary*, Apr. 18, 2019. [www.adeveloperdiary.com/data-science/machine-learning/understand-and-implement-the-backpropagation-algorithm-from-scratch-in-python/](https://www.adeveloperdiary.com/data-science/machine-learning/understand-and-implement-the-backpropagation-algorithm-from-scratch-in-python/). Ημερομηνία Πρόσβασης: 25 Ιανουαρίου 2023.
- [21] A. L. Chandra, “McCulloch-Pitts Neuron — Mankind’s First Mathematical Model Of A Biological Neuron,” *Medium*, Sep. 27, 2022. [towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1](https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1). Ημερομηνία Πρόσβασης: 25 Ιανουαρίου 2023.
- [22] A. L. Chandra, “Perceptron: The Artificial Neuron (An Essential Upgrade To The McCulloch-Pitts Neuron),” *Medium*, Jun. 03, 2020. [towardsdatascience.com/perceptron-the-artificial-neuron-4d8c70d5cc8d](https://towardsdatascience.com/perceptron-the-artificial-neuron-4d8c70d5cc8d). Ημερομηνία Πρόσβασης: 26 Ιανουαρίου 2023.

- [23] A. L. Chandra, “Perceptron Learning Algorithm: A Graphical Explanation Of Why It Works,” *Medium*, Sep. 27, 2022. [towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975](https://towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975). Ημερομηνία Πρόσβασης: 27 Ιανουαρίου 2023.
- [24] D. Lazar, “Perceptron: Explanation, Implementation and a Visual Example,” *Medium*, May 14, 2021. [towardsdatascience.com/perceptron-explanation-implementation-and-a-visual-example-3c8e76b4e2d1](https://towardsdatascience.com/perceptron-explanation-implementation-and-a-visual-example-3c8e76b4e2d1). Ημερομηνία Πρόσβασης: 27 Ιανουαρίου 2023.
- [25] S. Raschka, “What is the difference between a Perceptron, Adaline, and neural network model?,” *Sebastian Raschka, PhD*. [sebastianraschka.com/faq/docs/diff-perceptron-adaline-neuralnet.html](https://sebastianraschka.com/faq/docs/diff-perceptron-adaline-neuralnet.html) (accessed Jan. 28, 2023).
- [26] “Solving XoR problem using MLP.” [www.niser.ac.in/~smishra/teach/cs460/2020/lectures/lec19/](http://www.niser.ac.in/~smishra/teach/cs460/2020/lectures/lec19/). Ημερομηνία Πρόσβασης: 28 Ιανουαρίου 2023.
- [27] “What is Supervised Learning? | IBM.” [www.ibm.com/topics/supervised-learning](http://www.ibm.com/topics/supervised-learning). Ημερομηνία Πρόσβασης: 28 Ιανουαρίου 2023.
- [28] “What is Unsupervised Learning? | IBM.” [www.ibm.com/topics/unsupervised-learning](http://www.ibm.com/topics/unsupervised-learning). Ημερομηνία Πρόσβασης: 29 Ιανουαρίου 2023.
- [29] J. Shaikh, “Unsupervised Deep Learning | Deep Learning Essentials,” *Analytics Vidhya*, May 06, 2018. [www.analyticsvidhya.com/blog/2018/05/essentials-of-deep-learning-trudging-into-unsupervised-deep-learning/](http://www.analyticsvidhya.com/blog/2018/05/essentials-of-deep-learning-trudging-into-unsupervised-deep-learning/). Ημερομηνία Πρόσβασης: 29 Ιανουαρίου 2023.
- [30] J. TORRES.AI, “DRL 01: A gentle introduction to Deep Reinforcement Learning,” *Medium*, May 10, 2021. [towardsdatascience.com/drl-01-a-gentle-introduction-to-deep-reinforcement-learning-405b79866bf4](https://towardsdatascience.com/drl-01-a-gentle-introduction-to-deep-reinforcement-learning-405b79866bf4). Ημερομηνία Πρόσβασης: 30 Ιανουαρίου 2023.
- [31] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.
- [32] P. Mishra, “Why are Convolutional Neural Networks good for image classification?,” *Medium*, Jul. 20, 2019. [medium.datadriveninvestor.com/why-are-convolutional-neural-networks-good-for-image-classification-146ec6e865e8](https://medium.datadriveninvestor.com/why-are-convolutional-neural-networks-good-for-image-classification-146ec6e865e8). Ημερομηνία Πρόσβασης: 31 Ιανουαρίου 2023.
- [33] A. Rosebrock, “Convolutional Neural Networks (CNNs) and Layer Types,” *PyImageSearch*, May 14, 2021. [pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/](https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/). Ημερομηνία Πρόσβασης: 31 Ιανουαρίου 2023.
- [34] “Convolutional neural networks introduction - Machine Learning Group - Cosmos.” [www.cosmos.esa.int/web/machine-learning-group/convolutional-neural-networks-introduction](http://www.cosmos.esa.int/web/machine-learning-group/convolutional-neural-networks-introduction). Ημερομηνία Πρόσβασης: 31 Ιανουαρίου 2023.

- [35] Prabhu, “Understanding of Convolutional Neural Network (CNN) — Deep Learning,” *Medium*, Nov. 21, 2019. [medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148). Ημερομηνία Πρόσβασης: 31 Ιανουαρίου 2023.
- [36] C.-F. Wang, “A Basic Introduction to Separable Convolutions,” *Medium*, Aug. 14, 2018. [towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728](https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728). Ημερομηνία Πρόσβασης: 1 Φεβρουαρίου 2023.
- [37] A. Pandey, “Depth-wise Convolution and Depth-wise Separable Convolution,” *Medium*, Sep. 09, 2018. [medium.com/@zurister/depth-wise-convolution-and-depth-wise-separable-convolution-37346565d4ec](https://medium.com/@zurister/depth-wise-convolution-and-depth-wise-separable-convolution-37346565d4ec). Ημερομηνία Πρόσβασης: 1 Φεβρουαρίου 2023.
- [38] “Understanding the Bias-Variance Tradeoff.” [scott.fortmann-roe.com/docs/BiasVariance.html](https://scott.fortmann-roe.com/docs/BiasVariance.html). Ημερομηνία Πρόσβασης: 1 Φεβρουαρίου 2023.
- [39] S. Singh, “Understanding the Bias-Variance Tradeoff,” *Medium*, Oct. 09, 2018. [towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229](https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229). Ημερομηνία Πρόσβασης: 1 Φεβρουαρίου 2023.
- [40] U. Tewari, “Regularization — Understanding L1 and L2 regularization for Deep Learning,” *Analytics Vidhya*, Nov. 10, 2021. [medium.com/analytics-vidhya/regularization-understanding-l1-and-l2-regularization-for-deep-learning-a7b9e4a409bf](https://medium.com/analytics-vidhya/regularization-understanding-l1-and-l2-regularization-for-deep-learning-a7b9e4a409bf). Ημερομηνία Πρόσβασης: 1 Φεβρουαρίου 2023.
- [41] H. Tessier, “Neural Network Pruning 101,” *Medium*, Sep. 13, 2021. [towardsdatascience.com/neural-network-pruning-101-af816aaea61](https://towardsdatascience.com/neural-network-pruning-101-af816aaea61). Ημερομηνία Πρόσβασης: 1 Φεβρουαρίου 2023.
- [42] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, “A Survey of Quantization Methods for Efficient Neural Network Inference,” *CoRR*, vol. abs/2103.13630, 2021. Available: [arxiv.org/abs/2103.13630](https://arxiv.org/abs/2103.13630). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.
- [43] H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, “Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation,” *CoRR*, vol. Abs/2004.09602, 2020. Available: [arxiv.org/abs/2004.09602](https://arxiv.org/abs/2004.09602). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.
- [44] H. Cai *et al.*, “Enable Deep Learning on Mobile Devices: Methods, Systems, and Applications,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 27, no. 3, σελίδες: 1–50, May 2022, doi: 10.1145/3486618.
- [45] J. Kukacka, V. Golkov, and D. Cremers, “Regularization for Deep Learning: A Taxonomy,” *CoRR*, vol. abs/1710.10686, 2017, 2023. [Online]. Available: [arxiv.org/abs/1710.10686](https://arxiv.org/abs/1710.10686). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.

- [46] “Notes on weight sharing.” [kaggle.com/code/residentmario/notes-on-weight-sharing](https://kaggle.com/code/residentmario/notes-on-weight-sharing) Ημερομηνία Πρόσβασης: 2 Φεβρουαρίου 2023.
- [47] D. Arthur and S. Vassilvitskii, “k-means++: the advantages of careful seeding,” in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, 2007, σελίδες: 1027–1035. Available: [dl.acm.org/citation.cfm?id=1283383.1283494](https://dl.acm.org/citation.cfm?id=1283383.1283494). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.
- [48] P. Kumar, “DeConvolution and How it works and its importance?,” *Geek Culture*, Aug. 23, 2021. [medium.com/geekculture/deconvolution-and-how-it-works-and-its-importance-80ce8f137e1b](https://medium.com/geekculture/deconvolution-and-how-it-works-and-its-importance-80ce8f137e1b). Ημερομηνία Πρόσβασης: 3 Φεβρουαρίου 2023.
- [49] “Papers with Code - Global Average Pooling Explained.” [paperswithcode.com/method/global-average-pooling](https://paperswithcode.com/method/global-average-pooling) Ημερομηνία Πρόσβασης: 3 Φεβρουαρίου 2023.
- [50] “Android - Overview,” *TutorialsPoint*. [www.tutorialspoint.com/android/android\\_overview.html](http://www.tutorialspoint.com/android/android_overview.html) Ημερομηνία Πρόσβασης: 9 Φεβρουαρίου 2023.
- [51] “Application Fundamentals,” *Android Developers*. [developer.android.com/guide/components/fundamentals](https://developer.android.com/guide/components/fundamentals). Ημερομηνία Πρόσβασης: 9 Φεβρουαρίου 2023.
- [52] “What is Java ? | IBM.” [www.ibm.com/topics/java](http://www.ibm.com/topics/java). Ημερομηνία Πρόσβασης: 9 Φεβρουαρίου 2023.
- [53] “Keras Tutorial | Deep Learning with Python - Javatpoint,” [www.javatpoint.com](http://www.javatpoint.com). [www.javatpoint.com/keras](http://www.javatpoint.com/keras). Ημερομηνία Πρόσβασης: 10 Φεβρουαρίου 2023.
- [54] “Autoencoders in Deep Learning: Tutorial & Use Cases [2023].” [www.v7labs.com/blog/autoencoders-guide](http://www.v7labs.com/blog/autoencoders-guide), [www.v7labs.com/blog/autoencoders-guide](http://www.v7labs.com/blog/autoencoders-guide). Ημερομηνία Πρόσβασης: 16 Φεβρουαρίου 2023.
- [55] D.-J. Shin and J.-J. Kim, “A Deep Learning Framework Performance Evaluation to Use YOLO in Nvidia Jetson Platform,” *Applied Sciences*, vol. 12, no. 8, Art. no. 8, Jan. 2022, doi: 10.3390/app12083734.
- [56] “What are Threads in Computer Processor or CPU?,” *GeeksforGeeks*, Mar. 05, 2020. [www.geeksforgeeks.org/what-are-threads-in-computer-processor-or-cpu/](http://www.geeksforgeeks.org/what-are-threads-in-computer-processor-or-cpu/). Ημερομηνία Πρόσβασης: 18 Φεβρουαρίου 2023.
- [57] “Neural Networks API | Android NDK,” *Android Developers*. [developer.android.com/ndk/guides/neuralnetworks](https://developer.android.com/ndk/guides/neuralnetworks). Ημερομηνία Πρόσβασης: 18 Φεβρουαρίου 2023.
- [58] C. Versloot, “Machine learning articles.” Available: [github.com/christianversloot/machine-learning-articles/blob/6b0d8425bbb5b1dd28618d2c5dfdaf2a24f63513/tensorflow-pruning-](https://github.com/christianversloot/machine-learning-articles/blob/6b0d8425bbb5b1dd28618d2c5dfdaf2a24f63513/tensorflow-pruning-)

schedules-constantsparsity-and-polynomialdecay.md. Ημερομηνία Πρόσβασης: 20 Φεβρουαρίου 2023.

- [59] “Post-training quantization | TensorFlow Lite.” [www.tensorflow.org/lite/performance/post\\_training\\_quantization](http://www.tensorflow.org/lite/performance/post_training_quantization). Ημερομηνία Πρόσβασης: 20 Φεβρουαρίου 2023.
- [60] “TensorFlow Model Optimization.” [www.tensorflow.org/model\\_optimization](http://www.tensorflow.org/model_optimization). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.
- [61] J. O’Neill, “An Overview of Neural Network Compression,” *CoRR*, vol. abs/2006.03669, 2020. Available: [arxiv.org/abs/2006.03669](http://arxiv.org/abs/2006.03669). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.
- [62] A. V. D, “Neural Network XOR Application and Fundamentals,” *Medium*, May 02, 2017. [becominghuman.ai/neural-network-xor-application-and-fundamentals-6b1d539941ed](https://becominghuman.ai/neural-network-xor-application-and-fundamentals-6b1d539941ed). Ημερομηνία Πρόσβασης: 28 Φεβρουαρίου 2023.
- [63] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, σελίδες: 770–778. doi: 10.1109/CVPR.2016.90.
- [64] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, Jul. 2017, σελίδες: 2261–2269. doi: 10.1109/CVPR.2017.243.
- [65] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *CoRR*, vol. abs/1704.04861, 2017, [Online]. Available: [arxiv.org/abs/1704.04861](http://arxiv.org/abs/1704.04861). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.
- [66] C. Szegedy *et al.*, “Going deeper with convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, σελίδες: 1–9. doi: 10.1109/CVPR.2015.7298594.
- [67] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, 2017*, σελίδες 4278–4284. Available: [aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806](http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.
- [68] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning Transferable Architectures for Scalable Image Recognition,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, σελίδες: 8697–8710. doi: 10.1109/CVPR.2018.00907.

- [69] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 2019, vol. 97, σελίδες 6105–6114. Available: [proceedings.mlr.press/v97/tan19a.html](https://proceedings.mlr.press/v97/tan19a.html). Ημερομηνία Πρόσβασης: 7 Μαρτίου 2023.
- [70] Utmel, “Neural Processing Unit (NPU) Explained.” [www.utmel.com/blog/categories/integrated%20circuit/neural-processing-unit-npu-explained](http://www.utmel.com/blog/categories/integrated%20circuit/neural-processing-unit-npu-explained). Ημερομηνία Πρόσβασης: 4 Μαρτίου 2023.
- [71] “What is Python? Executive Summary,” *Python.org*. [www.python.org/doc/essays/blurb/](http://www.python.org/doc/essays/blurb/). Ημερομηνία Πρόσβασης: 6 Μαρτίου 2023.
- [72] Russakovsky, O. et al. “ImageNet Large Scale Visual Recognition Challenge”. *International Journal of Computer Vision (IJCV)*. (2015). 115(3), σελίδες: 211-252.