Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Εφαρμοσμενων Μαθηματικων και Φυσικων Επιστημων
Τομεας Μαθηματικων

Κατανεμημένα Ζεύγη Νευρωνικών Δικτύων

για την Αποδοτική Εκτέλεση Εφαρμογών Βαθιάς Μάθησης σε Κινητές Συσκευές

Διπλωματικη Εργασια

του

ΝΙΚΟΛΑΪΔΗ Ν. ΣΩΚΡΑΤΗ

Επιβλέπων: Ιάκωβος Βενιέρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2021

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Εφαρμοσμενων Μαθηματικων και Φυσικων Επιστημων
Τομεας Μαθηματικων

Κατανεμημένα Ζεύγη Νευρωνικών Δικτύων

για την Αποδοτική Εκτέλεση Εφαρμογών Βαθιάς Μάθησης σε Κινητές Συσκευές

Διπλωματικη Εργασια

του

ΝΙΚΟΛΑΪΔΗ Ν. ΣΩΚΡΑΤΗ

Επιβλέπων: Ιάκωβος Βενιέρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 28η Σεπτεμβρίου 2021.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Ιάκωβος Βενιέρης	Δήμητρα-Θεοδώρα Κακλαμάνη	Αντώνιος Συμβώνης
Καθηγητής Ε.Μ.Π.	Καθηγήτρια Ε.Μ.Π	Καθηγητής Ε.Μ.Π

Αθήνα, Σεπτέμβριος 2021

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Εφαρμοσμενων Μαθηματικων και Φυσικων Επιστημων
Τομεας Μαθηματικων

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....
Νικολαΐδης Σωκράτης

28 Σεπτεμβρίου 2021

Περίληψη

Παραδοσιακά, για την εκπαίδευση ή τη συμπερασματολογία Νευρωνικών Δικτύων για Βαθιά Μάθηση υπεύθυνη ήταν μία μόνο υπολογιστική μηχανή. Ωστόσο, σήμερα, η αποθήκευση μεγάλου όγκου δεδομένων για την εκπαίδευση Νευρωνικών Δικτύων καθώς και η έλλειψη υπολογιστικών πόρων σε μεμονωμένες μηχανές για την επεξεργασία τους, καθιστούν τη χρήση μίας μόνο υπολογιστικής μηχανής αναποτελεσματική. Καθώς ο όγκος δεδομένων και η πολυπλοκότητα των σύγχρονων μοντέλων είναι αναγκαία για την αύξηση της ακρίβειας, η ανάπτυξη ιδεών οι οποίες θα αντιμετωπίζουν τα προβλήματα που δημιουργεί η χρήση μίας μόνο υπολογιστικής μηχανής είναι απαραίτητη.

Με τις αναβαθμίσεις στον χώρο των κινητών συσκευών τα τελευταία χρόνια, οι οποίες έχουν επιφέρει μεγαλύτερη υπολογιστική ισχύ, μνήμη και καλύτερη διαχείριση της μπαταρίας, ένας νέος κλάδος αρχίζει και κερδίζει όλο και περισσότερο έδαφος. Η Κατανεμημένη Μηχανική Μάθηση σε Κινητές Συσκευές ξεφεύγει από τα όρια του νέφους αξιοποιώντας τα κατανεμημένα συστήματα για να ικανοποιήσει την ανάγκη για έξυπνες εφαρμογές στις σύγχρονες κινητές συσκευές. Ένα χαρακτηριστικό παράδειγμα τέτοιου συστήματος είναι τα Κατανεμημένα Ζεύγη Νευρωνικών Δικτύων, τα οποία χρησιμοποιούν δύο διαφορετικά νευρωνικά δίκτυα, από τα οποία το πρώτο εκτελείται σε έναν ισχυρό εξυπηρετητή στο νέφος ή στην άκρη του δικτύου και το δεύτερο στην κινητή συσκευή του χρήστη. Τα μοντέλα που χρησιμοποιούνται στον εξυπηρετητή είναι υπολογιστικά πιο «βαριά» αλλά ταυτόχρονα αποφέρουν μεγαλύτερη ακρίβεια, ενώ στην κινητή συσκευή ενσωματώνονται λιγότερο απαιτητικά μοντέλα με χαμηλότερη ακρίβεια, ώστε να είναι δυνατό να εκτελεστούν. Συγκριτικά με την εκτέλεση μόνο στην κινητή συσκευή, ένα Κατανεμημένο Ζεύγος μπορεί να βελτιώσει σημαντικά την απόδοση σε σχέση με διάφορες παραμέτρους, όπως είναι η ακρίβεια, το αποτύπωμα μνήμης ή η ενεργειακή κατανάλωση της κινητής συσκευής.

Στόχος της παρούσας διπλωματικής εργασίας είναι η μοντελοποίηση και ανάπτυξη ενός συστήματος Κατανεμημένου Ζεύγους Νευρωνικών Δικτύων, το οποίο θα λαμβάνει υπόψιν έναν μεγάλο αριθμό από παραμέτρους και μετρικές, ώστε να επιτρέπει την αποδοτική εκτέλεση εφαρμογών Βαθιάς Μάθησης σε κινητές συσκευές με βάση τις ανάγκες του χρήστη.

Λέξεις Κλειδιά

Μηχανική Μάθηση, Βαθιά Μάθηση, Κατανεμημένα Συστήματα, Ζεύγη Νευρωνικών Δικτύων, Κινητός Υπολογισμός, Edge Computing

Abstract

Traditionally, only a single computational machine was responsible for training and inference on Deep Neural Networks. However, today's need to store huge amounts of data for training along with the lack of computational resources, makes the deployment of a single machine for data processing obsolete. Since the use of big amounts of data and the increased complexity of the models are necessary to achieve high accuracy, new ideas to solve the problems that the use of a single machine creates are needed.

With the ever increasing development of mobile devices the last years, a new field is gaining ground. Mobile Distributed Machine Learning is able to escape the need for cloud computing and satisfy the demand for smart applications on modern mobile devices by using distributed systems. An interesting example of such a system is the Distributed Neural Network Pair, which takes advantage of two distinct neural networks, of which the first is on a powerful cloud server or a server on the network edge and the latter is on a mobile device. The server models are heavy but provide high accuracy while the device models are less demanding and provide lower accuracy. Using a distributed pair instead of just a mobile device can drastically improve the performance of many metrics such as the accuracy, the memory footprint or the energy consumption of the device.

This diploma thesis aims to model and develop such a system which will take into account a large number of variables and metrics to achieve efficient execution of deep learning applications on mobile devices.

Keywords

Machine Learning, Deep Learning, Distributed Systems, Neural Network Pair, Mobile Computing, Edge Computing

στους γονείς μου

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όσους συνέβαλαν στην εκπόνησή της.

Αρχικά, θα ήθελα να απευθύνω τις ευχαριστίες μου στον καθηγητή του Ε.Μ.Π. κ. Ιάκωβο Βενιέρη για την εμπιστοσύνη που μου έδειξε καθώς και τη δυνατότητα που μου προσέφερε να εκπονήσω τη διπλωματική μου εργασία πάνω σε έναν κλάδο στον οποίο ήθελα να εργαστώ και να εμβαθύνω. Ευχαριστώ ιδιαίτερα τον κ. Ιωάννη Πανόπουλο, υποψήφιο διδάκτορα ΣΗΜΜΥ ΕΜΠ, για την καθοδήγηση και την υπομονή του χαρις τις οποίες ήταν δυνατή η ολοκλήρωση της διπλωματικής εργασίας. Επίσης, ευχαριστώ θερμά τον Δρ. Στυλιανό Βενιέρη, ερευνητή στο κέντρο Τεχνητής Νοημοσύνης της Samsung στο Cambridge για τις πολύτιμες γνώσεις που μου προσέφερε κατά τη διάρκεια της εκπόνησης. Ακόμη, θα ήθελα να ευχαριστήσω τους καθηγητές κ. Δήμητρα-Θεοδώρα Κακλαμάνη και κ. Αντώνιο Συμβώνη για τη συμμετοχή τους στην τριμελή εξεταστική επιτροπή.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου η οποία μου προσέφερε πάντα απλόχερα ό,τι χρειαζόμουν και τους φίλους μου οι οποίοι ήταν πάντα δίπλα μου.

Αθήνα, Σεπτέμβριος 2021

Νικολαΐδης Σωκράτης

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Όταν έγινε η πρώτη σύλληψη της ιδέας ενός προγραμματίσιμου υπολογιστή, οι εφευρέτες αναρωτήθηκαν αν μια τέτοια μηχανή θα μπορούσε κάποια στιγμή στο μέλλον να είναι ευφυής. Σήμερα, η Τεχνητή Νοημοσύνη (Artificial Intelligence) είναι ένας ακμάζων τομέας με πληθώρα πρακτικών εφαρμογών και ενεργών θεμάτων έρευνας [1].

Η επιτυχία της Τεχνητής Νοημοσύνης οφείλεται σε μεγάλο βαθμό στη Βαθιά Μάθηση. Η Βαθιά Μάθηση επιτρέπει σε υπολογιστικά μοντέλα τα οποία αποτελούνται από πολλαπλά στρώματα επεξεργασίας να μαθαίνουν αναπαραστάσεις δεδομένων με πολλά επίπεδα αφαίρεσης. Αυτές οι μέθοδοι έχουν βελτιώσει δραματικά τις τεχνολογίες της αναγνώρισης ομιλίας, της αναγνώρισης οπτικών αντικειμένων, του εντοπισμού αντικειμενών και πολλών άλλων [2].

Τα τελευταία χρόνια έχουμε βιώσει μια εκρηκτική ανάπτυξη στις κινητές συσκευές οι οποίες έχουν διεισδύσει σε κάθε πλευρά της καθημερινότητάς μας. Με την αυξανόμενη χρήση κινητών συσκευών και «έξυπνων» εφαρμογών, καθώς και την τεράστια επιτυχία της Βαθιάς Μάθησης, είναι φυσική η τάση ώθησης της Βαθιάς Μάθησης σε εφαρμογές κινητών συκευών. Παρ' όλα αυτά, υπάρχουν πολλές προκλήσεις που πρέπει να μελετηθούν, όπως είναι η αντίθεση ανάμεσα στη μικρή υπολογιστικά φύση των κινητών συσκευών και την ανάγκη για υπολογιστικούς πόρους που έχουν τα Βαθιά Νευρωνικά Δίκτυα [3].

Η Κατανεμημένη Μηχανική Μάθηση (Distributed Machine Learning) στοχεύει να επιλύσει το πρόβλημα της αποθήκευσης του όγκου δεδομένων και της ανάγκης υπολογιστικών πόρων λόγω της αυξημένης πολυπλοκότητας των μοντέλων μέσα από τη συνεργασία πολλών εξυπηρετητών. Η βελτίωση της υπολογιστικής ισχύος και μνήμης των κινητών συσκευών επιτρέπει τη χρήση τους σε τέτοια συστήματα κατανεμημένης Μηχανικής Μάθησης [4].

Ένα Κατανεμημένο Ζεύγος Νευρωνικών Δικτύων ανήκει στον τομέα της Κατανεμημένης Μηχανικής Μάθησης και αποτελείται από δύο διαφορετικά σημεία εκτέλεσης, τον εξυπηρετητή (server) και την κινητή συσκευή (mobile device). Τα μοντέλα Βαθιάς Μάθησης στα σημεία εκτέλεσης είναι διαφορετικά, στον εξυπηρετητή ένα μοντέλο με υψηλή ακρίβεια αλλά και αυξημένες υπολογιστικές απαιτήσεις ενώ στην κινητή συσκευή ένα μοντέλο με μικρότερη ανάγκη υπολογιστικών πόρων αλλά και μειωμένη ακρίβεια.

1.1 Αντικείμενο διπλωματικής εργασίας

Οι τεχνολογικές εξελίξεις τόσο στην υπολογιστική ισχύ των κινητών συσκευών όσο και στα δίκτυα δεδομένων (π.χ. 5G, WiFi 5GHz) τα οποία επιτρέπουν τη γρήγορη και συνεπή

μεταφορά δεδομένων, έχουν επιτρέψει την ανάπτυξη συστημάτων Κατανεμημένων Ζευγών Νευρωνικών Δικτύων.

Τέτοια συστήματα αποσκοπούν στη βέλτιστη εκτέλεση εφαρμογών Βαθιάς Μάθησης ανάλογα με τις ανάγκες του χρήστη. Για να γίνει αυτό πρέπει να ληφθούν υπόψιν πολλές μετρικές όπως η ακρίβεια, ο χρόνος απόκρισης, η ενεργειακή απόδοση καθώς και οι περιορισμοί του συστήματος όπως είναι οι υπολογιστικοι πόροι, ο φόρτος εργασίας ή ακόμη και η θερμοκρασιά ή η μπαταρία της κινητής συσκευής.

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η μοντελοποίηση και η ανάπτυξη ενός συστήματος ζεύγους κατανεμημένων νευρωνικών δικτύων, με σκοπό την αποδοτική εκτέλεση εφαρμογών Βαθιάς Μάθησης σε κινητές συσκευές. Βασικοί στόχοι της μοντελοποίησης είναι η δημιουργία ενός συστήματος το οποία θα χαρακτηρίζεται από (α) πληρότητα, με την μοντελοποίηση πληθώρας μεταβλητών και τη χρήση των κατάλληλων μετρικών, ώστε το σύστημα να περιγράφεται όσο το δυνατόν καλύτερα και να αντικατοπτρίζει ρεαλιστικά σενάρια και (β) ευελιξία και επεκτασιμότητα, ώστε να μπορεί κανείς εύκολα να μεταβάλλει ή να επεκτείνει το υπάρχον σύστημα.

1.2 Οργάνωση τόμου

Η διπλωματική εργασία έχει οργανωθεί σε 6 κεφάλαια.

Το 2ο Κεφάλαιο περιλαμβάνει το θεωρητικό υπόβαθρο πάνω στο οποίο βασίστηκε η εργασία. Γίνεται μια σύντομη σύγκριση ανάμεσα στη Μηχανική και τη Βαθιά Μάθηση και εξηγείται στο που οφείλεται η επιτυχία της Βαθιάς Μάθησης καθώς και οι τεχνολογικές εξελίξεις που έχουν επιτρέψει την ευρεία χρήση της. Έπειτα, γίνεται μια σύντομη παρουσίαση των Τεχνητών Νευρωνικών Δικτύων, τα οποία αποτελούν την πλέον διαδεδομένη δομή μοντέλων Βαθιάς Μάθησης. Βασικό σημείο του θεωρητικού υποβάθρου είναι τα Συνελικτικά Νευρωνικά Δίκτυα, τα οποία ειδικεύονται στην αναγνώριση προτύπων σε εικόνες. Τέλος αναφέρονται τα Κατανεμημένα Συστήματα Μηχανικής Μάθησης με έμφαση στα Κατανεμημένα Ζεύγη Νευρωνικών Δικτύων.

Στο 3ο Κεφάλαιο παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για τη μοντελοποίηση και την ανάπτυξη του συστήματος. Αυτές είναι η προγραμματιστική γλώσσα Python, η βιβλιοθήκη ανοιχτού λογισμικού Tensorflow, η βιβλιοθήκη Keras και η βάση δεδομένων ImageNet.

Στο 4ο Κεφάλαιο γίνεται εκτενής παρουσίαση του συστήματος. Αρχικά παρουσιάζονται οι συναρτήσεις που το αποτελούν ως προς το σκοπό τους, τα στοιχεία εισόδου και εξόδου και τη λειτουργία τους. Ύστερα, εξηγείται λεπτομερώς η συνεργασίας τους μέσα από παρουσίαση της ροής εκτέλεσης του συστήματος.

Στο 5ο Κεφάλαιο παρουσιάζονται οι μετρήσεις που έγιναν για την αξιολόγηση και κατανόηση του συστήματος.

Τέλος, στο Κεφάλαιο 6 έχουμε τον επίλογο στον οποίο δίνονται κάποια τελικά συμπεράσματα καθώς και μελλοντική έρευνα που μπορεί να γίνει πάνω στο σύστημα.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Στο δεύτερο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο στο οποίο βασίζεται η διπλωματική εργασία.

2.1 Βαθιά Μάθηση

Η Βαθιά Μάθηση υπάγεται στον ευρύτερο κλάδο της Μηχανικής Μάθησης, η οποία με τη σειρά της είναι ένα υποσύνολο της Τεχνητής Νοημοσύνης όπως απεικονίζεται στην Εικόνα 2.1. Η Μηχανική Μάθηση αποσκοπεί στην αυτοματοποίηση των μεθόδων ανάλυσης δεδομένων. Συγκεκριμένα, είναι το σύνολο των μεθόδων που μπορούν αυτόματα να αναγνωρίσουν μοτίβα στα δεδομένα και στη συνέχεια χρησιμοποιώντας αυτά τα μοτίβα, να προβλέψουν μελλοντικά δεδομένα ή να λάβουν αποφάσεις σε καταστάσεις αβεβαιότητας [5].

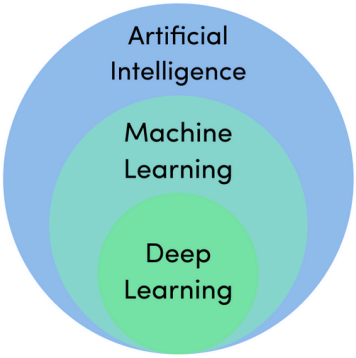
Η Βαθιά Μάθηση διαφέρει από τη Μηχανική Μάθηση σε τρία βασικά σημεία:

1. Οι αλγόριθμοι Μηχανικής Μάθησης είναι αρκετά απλοί σε αντίθεση με το δίκτυο αλγορίθμων της Βαθιάς Μάθησης οι οποίοι περιλαμβάνουν πολλαπλά επίπεδα και προσπαθούν να μιμηθούν τη λειτουργία του ανθρώπινου εγκεφάλου.

2. Οι αλγόριθμοι Μηχανικής Μάθησης αδυνατούν να επεξεργαστούν ακατέργαστα δεδομένα, οπότε είναι ευθύνη του ερευνητή να προεπεξεργαστεί τα δεδομένα και να εξάγει τα κατάλληλα χαρακτηριστικά, τα οποία θα τροφοδοτηθούν στη συνέχεια στα μοντέλα. Οι αλγόριθμοι Βαθιάς Μάθησης από την άλλη εκτελούν αυτόματα εξαγωγή χαρακτηριστικών, συνεπώς η ανάγκη για ανθρώπινη παρέμβαση μειώνεται σημαντικά.

3. Λόγω της περίπλοκης αρχιτεκτονικής πολλών επιπέδων, οι αλγόριθμοι Βαθιάς Μάθησης χρειάζονται πολύ μεγαλύτερα σύνολα δεδομένων σε σύγκριση με τους αλγορίθμους Μηχανικής Μάθησης [6].

Η επιτυχία κάθε μεθόδου η οποία λειτουργεί με γνώμονα τα δεδομένα, εξαρτάται κυρίως από τη γνώση του τι πρέπει να μετρηθεί αλλά και πώς αυτό πρέπει να μετρηθεί. Ωστόσο, αν και πολύ σημαντική για τη Μηχανική Μάθηση, η διαδικασία επιλογής και σχεδίασης χαρακτηριστικών δεν είναι αυτοματοποιημένη, αλλά συνήθως γίνεται από ειδικούς πάνω στο θέμα μελέτης από όπου προέρχονται τα δεδομένα. Αυτό έχει σαν αποτέλεσμα η σχεδίαση και προετοιμασία του συνόλου δεδομένων πολλές φορές να καταναλώνει το μεγαλύτερο μέρος του χρόνου και των πόρων ενός προβλήματος Μηχανικής Μάθησης [7].

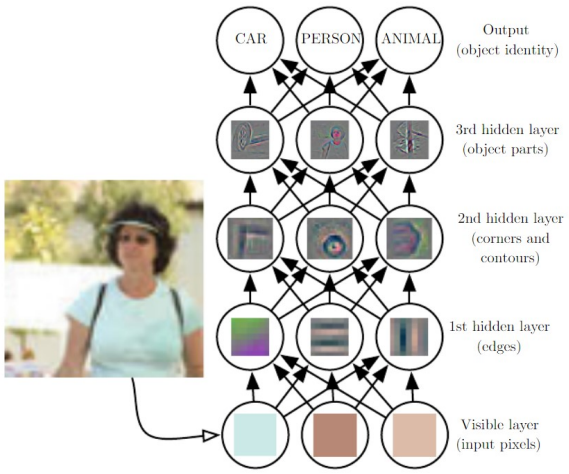


Εικόνα 2.1: *Σχέση Τεχνητής Νοημοσύνης, Μηχανικής Μάθησης, Βαθιάς Μάθησης*

Οι αλγόριθμοι Βαθιάς Μάθησης έχουν αποδειχθεί εξαιρετικά αποτελεσματικοί στην αυτόματη αναγνώριση χαρακτηριστικών από τα ακατέργαστα δεδομένα. Μάλιστα, αν έχουν πρόσβαση σε αρκετά μεγάλα σύνολα δεδομένων, οι μοντέρνοι αλγόριθμοι Βαθιάς Μάθησης πετυχαίνουν μεγαλύτερη ακρίβεια από αλγορίθμους Μηχανικής Μάθησης που χρησιμοποιούν χαρακτηριστικά που έχουν επιλεχθεί από ανθρώπους [7].

Αυτό το πλεονέκτημα της Βαθιάς Μάθησης είναι εξαιρετικά χρήσιμο, ιδιαίτερα σε προβλήματα που περιέχουν μεγάλο αριθμό δεδομένων. Για παράδειγμα, η εξαγωγή χαρακτηριστικών για ένα πρόβλημα αναγνώρισης προσώπων σε φωτογραφίες είναι μια υπερβολικά δύσκολη πρόκληση για έναν άνθρωπο. Με τη Βαθιά Μάθηση όμως, η εξαγωγή αυτή γίνεται αυτόματα [7].

Τα μοντέλα Βαθιάς Μάθησης προσπαθούν να προσεγγίσουν μια αρκετά περίπλοκη συνάρτηση από τις τιμές εισόδου στις τιμές εξόδου, η οποία μπορεί να διασπαστεί σε μια σειρά από απλούστερες εμφωλευμένες συναρτήσεις που περιγράφονται από τα διάφορα επίπεδα του μοντέλου. Μερος της αντιμετώπισης ενός προβλήματος Βαθιάς Μάθησης είναι η εκμάθηση και η αξιολόγηση αυτής της συνάρτησης με κάθε σειρά από επίπεδα του μοντέλου να μπορεί να εξάγει ολοένα και πιο αφηρημένα χαρακτηριστικά [1].



Εικόνα 2.2: *Παράδειγμα μοντέλου Βαθιάς Μάθησης*

Για παράδειγμα αν οι τιμές εισόδου είναι ένα σύνολο από pixels μιας εικόνας και οι τιμές εξόδου είναι ταυτότητες αντικειμένων, η υπολογιστική μηχανή δεν μπορεί να καταλάβει απευθείας το νόημα των εισαγώμενων τιμών. Σε ένα μοντέλο Βαθιάς Μάθησης όμως, τα πρώτα επίπεδα μπορεί να αναγνωρίζουν από τις εισόδους ακμές, τα επόμενα χρησιμοποιώντας τις τιμές που εξάγονται από τα προηγούμενα να αναγνωρίζουν γωνίες και περιγράμματα, τα επόμενα επίπεδα πιο βαθιά στην ιεραρχεία να αναγνωρίζουν μέρη αντικειμένων και τελικά οι τιμές που εξάγονται να μπορούν να αντιστοιχηθούν σε κάποια ταυτότητα αντικειμένου. Αυτή η διαδικασία φαίνεται στην Εικόνα 2.2 [1].

Οι εφαρμογές Βαθιάς Μάθησης μπορούν να αναλυθούν σε δύο βασικές διαδικασίες, την εκπαίδευση και τη συμπερασματολογία [3].

Εκπαίδευση (training): Κατά τη διαδικασία της εκπαίδευσης δίνονται στο μοντέλο Βαθιάς Μάθησης γνωστά δεδομένα και αυτό κάνει προβλέψεις για το τι αναπαριστούν τα δεδομένα. Κάθε λάθος στις προβλέψεις του μοντέλου, χρησιμοποιείται για να ενημερωθούν οι παράμετροί του. Όσο προχωράει η διαδικασία της εκπαίδευσης, οι παράμετροι ρυθμίζονται περαιτέρω μέχρι οι προβλέψεις του μοντέλου να έχουν ικανοποιητική ακρίβεια.

Συμπερασματολογία (inference): Συμπερασματολογία είναι η διαδικασία κατά την οποία χρησιμοποιείται ένα μοντέλο Βαθιάς Μάθησης για να κάνει προβλέψεις σε δεδομένα που δεν έχει ξαναδεί. Συμπερασματολογία εκτελείται και κατά τη διάρκεια της εκπαίδευσης, καθώς κάθε φορά που δίνεται μια εικόνα στο μοντέλο, αυτό προσπαθεί να την κατηγοριοποιήσει. Η αξιοποίηση ενός εκπαιδευμένου μοντέλου Βαθιάς Μάθησης για συμπερασματολογία είναι εύκολη αφού μπορούμε απλώς να χρησιμοποιήσουμε ένα αντίγραφο του εκπαιδευμένου μοντέλου όπως είναι [8].

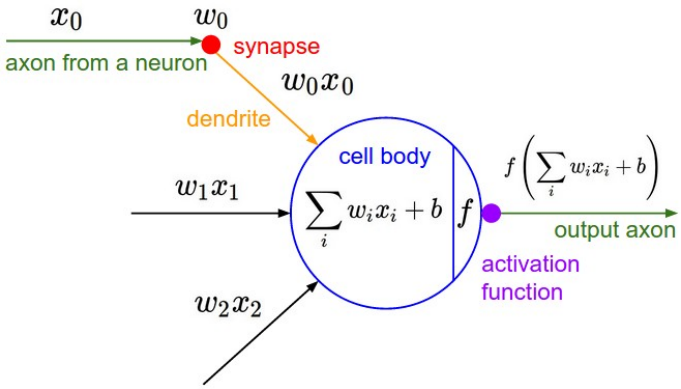
Η Βαθιά Μάθηση είχε ξεκινήσει σαν ιδέα από τις δεκαετίες 1940-1960 και λόγω διάφορων εξελίξεων χρησιμοποίειται ευρέως τις τελευταίες δεκαετίες. Οι βασικοί λόγοι για τη διάδοσή της είναι:

- Η δραματική αύξηση των δεδομένων που είναι διαθέσιμα για την εκπαίδευση των μοντέλων. Αυτή η αύξηση οφείλεται κυρίως στην εισαγωγή του διαδικτύου σε κάθε τομέα της ζωής του ανθρώπου.
- Η αύξηση του μεγέθους των μοντέλων με την εξέλιξη τόσο του hardware όσο και του software που χρησιμοποιούνται για την εφαρμογή μοντέλων Βαθιάς Μάθησης.
- Η επίλυση όλο και πιο περίπλοκων προβλημάτων με αυξανόμενη ακρίβεια κάνοντας χρήση Βαθιάς Μάθησης [1].

Αυτή τη στιγμή οι περισσότερες διαδικτυακές εταιρείες και τεχνολογίες χρησιμοποιούν Βαθιά Μάθηση για διάφορες εφαρμογές. Για παράδειγμα, η Facebook τη χρησιμοποιεί για ανάλυση κειμένου σε συζητήσεις στο διαδίκτυο, η Google, η Baidu και η Microsoft τη χρησιμοποιούν για αναγνώριση εικόνων και μετάφραση. Όλα τα μοντέρνα έξυπνα κινητά (smartphones) εκτελούν εφαρμογές που χρησιμοποιούν αναγνώριση φωνής και αναγνώριση προσώπου τα οποία βασίζονται στη Βαθιά Μάθηση. Υπάρχουν ακόμα πάρα πολλές εφαρμογές σε πολλούς άλλους τομείς, όπως στον τομέα της υγείας για επεξεργασία εικόνων για διάγνωση ή στην αυτοκινητοβιομηχανία για την ανάπτυξη αυτοοδηγούμενων αυτοκινήτων [7].

2.2 Τεχνητά Νευρωνικά Δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα είναι η πιο διαδεδομένη δομή μοντέλων Βαθιάς Μάθησης. Το θεμελιώδες στοιχείο του Νευρωνικού Δικτύου είναι ο Νευρώνας (Neuron) ή perceptron. Ένας νευρώνας παράγει έναν γραμμικό συνδυασμό από τις εισόδους του και στη συνέχεια τον διοχετεύει σε μια μη γραμμική συνάρτηση ενεργοποίησης ώστε να παραχθεί η έξοδος [9]. Η έμπνευση για τη δομή των νευρώνων των Τεχνητών Νευρωνικών Δικτύων προήλθε από τον τρόπο λειτουργίας των νευρώνων ενός εγκεφάλου. Η αρχιτεκτονική και η λειτουργία ενός Νευρώνα παρουσιάζονται στην Εικόνα 2.3.



Εικόνα 2.3: *Αρχιτεκτονική νευρώνα*

Οι πιο γνωστές συναρτήσεις ενεργοποίησης είναι:

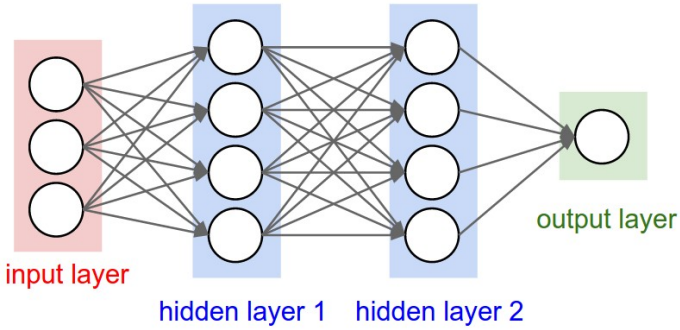
- Σιγμοειδής (Sigmoid): $\sigma(x) = 1/(1 + e^{-x})$
- Υπερβολική εφαπτομένη (Tanh): $tanh(x) = 2\sigma(2x) - 1$
- Rectified Linear Unit (ReLU): $f(x) = max(0, x)$

Τα Τεχνητά Νευρωνικά Δίκτυα μοντελοποιούνται ως συλλογές νευρώνων οι οποίοι είναι συνδεδεμένοι σε ένα ακυκλικό γράφημα. Η έξοδος (output) κάποιου νευρώνα μπορεί να είναι η είσοδος (input) ενός άλλου. Συνήθως οι νευρώνες οργανώνονται σε επίπεδα (layers) με το πιο συνηθισμένο επίπεδο να είναι το πλήρες συνδεδεμένο (fully connected). Σε αυτό, οι νευρώνες δύο γειτονικών επιπέδων είναι πλήρως συνδεδεμένοι και κανένας νευρώνας δεν συνδέεται με νευρώνα που βρίσκεται στο ίδιο επίπεδο. Αυτή η αρχιτεκτονική φαίνεται στην Εικόνα 2.4.

Το επίπεδο εισόδου (input layer) είναι το επίπεδο στο οποίο εισάγονται τα δεδομένα, ενώ επίπεδο εξόδου (output layer) είναι το επίπεδο στο οποίο εξάγεται το συμπέρασμα και συνήθως δεν περιλαμβάνει συνάρτηση ενεργοποίησης. Τα υπόλοιπα ενδιάμεσα επίπεδα αναφέρονται ως κρυφά επίπεδα [10].

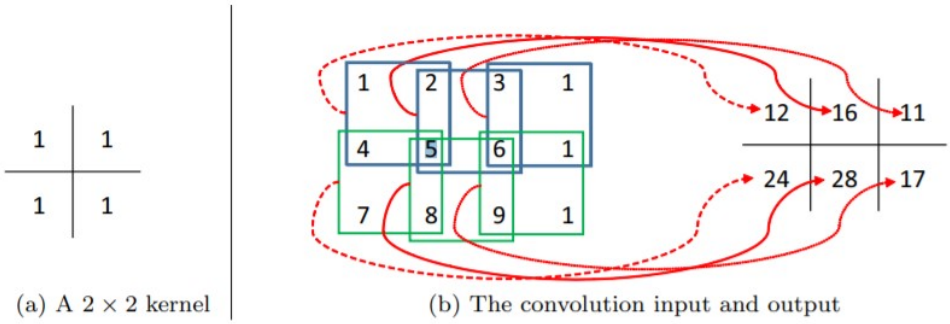
2.3 Συνελικτικά Νευρωνικά Δίκτυα

Τα Συνελικτικά Νευρωνικά Δίκτυα (CNNs) είναι ένα είδος Νευρωνικών Δικτυών τα οποία ειδικεύονται στην επεξεργασία δεδομένων σε μορφή πλέγματος, όπως είναι οι εικόνες, οι οποίες μπορούν να εκφραστούν ως ένα δισδιάστατο ή τρισδιάστατο πλέγμα από pixels [1].



Εικόνα 2.4: *Αρχιτεκτονική νευρωνικού δικτύου πολλαπλών επιπέδων*

Πήραν την ονομασία τους από τη συνέλιξη (convolution), μια γραμμική μαθηματική πράξη μεταξύ πινάκων [11]. Όπως φαίνεται και στην Εικόνα 2.5, ο πίνακας εισόδου (input) 3×4 συνελλίσεται με τον πυρήνα (kernel) 2×2 και προκύπτει ο πίνακας εξόδου 2×3. Ο πυρήνας διασχίζει την είσοδο και σε κάθε βήμα υπολογίζεται το γραμμικό άθροισμα των τιμών τους [12].



Εικόνα 2.5: *Παράδειγμα δισδιάστατης συνέλιξης*

Ο γενικός τύπος της συνέλιξης ανάμεσα σε ένα δισδιάστατο δείγμα εισόδου I κι έναν δισδιάστατο πυρήνα K είναι:

$$(I * K)(i,j) := \sum_m \sum_n I(m,n)K(i-m, j-n)\ [1].$$

Κατά τη χρήση παραδοσιακών Νευρωνικών Δικτύων για επεξεργασία δεδομένων εικόνων παρατηρήθηκαν δύο σημαντικά θέματα. Το πρώτο είναι ότι λόγω της αρχιτεκτονικής των πλήρως συνδεδεμένων επιπέδων, αν τα δεδομένα εισόδου είναι πολυδιάστατα (όπως είναι οι εικόνες), το δίκτυο καταλήγει να έχει υπερβολικά μεγάλο αριθμό παραμέτρων. Ο δεύτερος είναι ότι τα παρδοσιακά Νευρωνικά Δίκτυα δεν παρέχουν τη δυνατότητα αναγνώρισης τοπικών χαρακτηριστικών, κάτι που έχει αποδειχθεί ότι είναι απαραίτητο.

Καθώς τα Συνελικτικά Νευρωνικά Δίκτυα προορίζονται πρωταρχικά για αναγνώριση προτύπων σε εικόνες, μπορούν να γίνουν συμβιβασμοί στην αρχιτεκτονική τους οι οποίοι οδηγούν σε ικανά μοντέλα με μειωμένο αριθμό παραμέτρων [13].

2.3.1 Επίπεδα

Τα Συνελικτικά Νευρωνικά Δίκτυα έχουν πολλαπλά είδη επιπέδων. Τα βασικότερα από αυτά είναι: το συνελικτικό επίπεδο, το επίπεδο ενεργοποίησης, το επίπεδο συγκέντρωσης και το πλήρως συνδεδεμένο επίπεδο. Τα συνελικτικά και πλήρως συνδεδεμένα επίπεδα περιλαμβάνουν παραμέτρους, σε αντίθεση με τα επίπεδα ενεργοποίησης και συγκέντρωσης [11]. Όταν αυτά τα επίπεδα τοποθετούνται σε σειρά, έχουμε μια αρχιτεκτονική Συνελικτικού Νευρωνικού Δικτύου. Κάθε επίπεδο εκτελεί μια διαφορετική λειτουργία την οποία χρειάζεται το μοντέλο [13].

Συνελικτικό Επίπεδο

Το Συνελικτικό Επίπεδο έχει βασικό ρόλο στον τρόπο λειτουργίας των Συνελικτικών Νευρωνικών Δικτύων. Οι παράμετροι του επιπέδου εστιάζουν γύρω από τη χρήση εκπαιδεύσιμων φίλτρων (kernels). Αυτά τα φίλτρα είναι συνήθως μικρά σε διαστάσεις, αλλά εξαπλώνονται προς όλες τις διαστάσεις της εισόδου. Όταν τα δεδομένα φτάνουν σε ένα συνελικτικό επίπεδο, κάθε φίλτρο συνελίσσεται κατά μήκος και πλάτος των χωρικών διαστάσεων της εισόδου, δηλαδή το φίλτρο ολισθαίνει πάνω στην είσοδο και σε κάθε θέση υπολογίζεται το εσωτερικό γινόμενο του φίλτρου με την αντίστοιχη περιοχή της εισόδου, παράγοντας έναν δισδιάστατο χάρτη ενεργοποίησης (activation map). Έτσι το δίκτυο μαθαίνει φίλτρα τα οποία ενεργοποιούνται όταν βλέπουν ένα συγκεκριμένο χαρακτηριστικό σε μια συγκεκριμένη θέση της εισόδου. Κάθε φίλτρο έχει έναν αντίστοιχο χάρτη ενεργοποίησης και αυτοί οι χάρτες στοιβάζονται κατά μήκος της διάστασης του βάθους σχηματίζοντας την έξοδο του συνελικτικού επιπέδου.

Τα συνελικτικά επίπεδα μπορούν να βελτιστοποιήσουν την έξοδό τους μέσα από τρεις υπερπαραμέτρους, το βάθος (depth), τον βηματισμό (stride) και την προσθήκη μηδενικών στο σύνορο (zero-padding).

- Το **βάθος** είναι ο αριθμός των φίλτρων που έχει το επίπεδο και αντιστοιχεί στον αριθμό των διαφορετικών χαρακτηριστικών που μπορεί να μάθει.
- Ο **βηματισμός** καθορίζει τη μετατόπιση του φίλτρου σε αριθμό pixels κατά την ολίσθησή του πάνω στον πίνακα εισόδου. Όσο μεγαλύτερος είναι ο βηματισμός, τόσο μικρότερο είναι το μέγεθος της εξόδου.
- Η **προσθήκη μηδενικών** στο σύνορο της εισόδου γίνεται είτε επειδή υπάρχει πιθανότητα το φίλτρο να μην χωράει απόλυτα στην είσοδο, είτε για να ελεγχθούν οι διαστάσεις της εξόδου. Λίγα μηδενικά στο σύνορο της εισόδου μεταβάλλουν ελάχιστα το αποτέλεσμα της συνέλιξης, οπότε η προσθήκη μηδενικών είναι αποδοτική μέθοδος αντιμετώπισης αυτών των δύο περιπτώσεων.

Ο τρόπος λειτουργίας του συνελικτικού επιπέδου, δηλαδή η συνέλιξη των φίλτρων πάνω στην εικόνα, συνδέει κάθε νευρώνα του επιπέδου με μια μικρή μόνο περιοχή της εισόδου. Το μέγεθος αυτής της περιοχής συχνά αναφέρεται ως μέγεθος δεκτικού πεδίου (receptive filed size). Αυτό του επιτρέπει να αποφύγει το μεγάλο αριθμό παραμέτρων αλλά κυριότερα δίνει τη δυνατότητα στο δίκτυο να αναγνωρίζει τοπικές περιοχές ενδιαφέροντος.

Ακόμα και με όσα αναφέρθηκαν παραπάνω, ο αριθμός των παραμέτρων συνεχίζει να είναι τεράστιος. Η μέθοδος του **διαμοιρασμού παραμέτρων (parameter sharing)**, επιτρέπει εκτενή περικοπή των παραμέτρων του συνελικτικού επιπέδου. Βασίζεται στην υπόθεση ότι αν είναι χρήσιμο να υπολογιστεί ένα χαρακτηριστικό σε μια συγκεκριμένη περιοχή, τότε είναι πιθανό να είναι χρήσιμο και σε μια άλλη περιοχή. Περιορίζοντας κάθε φίλτρο στο να χρησιμοποιεί τις ίδιες παραμέτρους για κάθε νευρώνα, παρατηρείται σημαντική μείωση του αριθμού των παραμέτρων που παράγονται από ένα συνελικτικό επίπεδο.

Επίπεδο Συγκέντρωσης

Το επίπεδο συγκέντρωσης στοχεύει στη σταδιακή μείωση των διαστάσεων των χαρτών ενεργοποίησης και συνεπώς στη μείωση του αριθμού των παραμέτρων και της υπολογιστικής πολυπλοκότητας του μοντέλου. Το επίπεδο συγκέντρωσης δρα σε μικρές περιοχές κάθε χάρτη ενεργοποίησης (συνήθως 2×2 με βηματισμό 2), περνώντας τις τιμές αυτών από κάποια συνάρτηση συγκέντρωσης η οποία τις μετατρέπει σε μία μόνο τιμή. Αυτό συμπυκνώνει τους χάρτες ενεργοποίησης (στη περίπτωση που αναφέρθηκε κατά 75%) χωρίς να μειώσει το πλήθος τους. Οι δημοφιλέστερες συναρτήσεις συγκέντρωσης είναι:

- **Μέγιστη Συγκέντρωση (max pooling)**, επιστρέφει τη μέγιστη τιμή της περιοχής.
- **Συγκέντρωσή μέσου όρου (average pooling)**, επιστρέφει τον μέσο όρο της περιοχής.
- **Συγκέντρωση της L^2 νόρμας (L^2-norm pooling)**, επιστρέφει την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των τιμών της περιοχής.

Πλήρως Συνδεδεμένο Επίπεδο

Η λογική του Πλήρως Συνδεδεμένου Επιπέδου στα Συνελικτικά Δίκτυα είναι η ίδια με αυτή στα παραδοσιακά Τεχνητά Νευρωνικά Δίκτυα που περιγράφηκαν στην Ενότητα 2.2. Οι νευρώνες ενός τέτοιου επιπέδου είναι πλήρως συνδεδεμένοι με τους νευρώνες γειτονικών Πλήρως Συνδεδεμένων Επιπέδων, ενώ νευρώνες του ίδιου επιπέδου δεν έχουν καμία επικοινωνία μεταξύ τους [13].

2.3.2 Γνωστές Αρχιτεκτονικές

Η πιο παραδοσιακή τεχνική κατασκευής Συνελικτικών Νευρωνικών Δικτύων είναι μια σειρά από Συνελικτικά Επίπεδα ακολουθούμενα από ένα Επίπεδο Συγκέντρωσης. Αυτό το σχήμα επαναλαμβάνεται αρκετές φορές μέχρι το μέγεθος της εικόνας να έχει συμπυκνωθεί αρκετά. Στη συνέχεια, γίνεται αλλαγή σε Πλήρως Συνδεδεμένα Επίπεδα. Μετά από κάθε Συνελικτικό Επίπεδο ή Πλήρως Συνδεδεμένο Επίπεδο υπάρχει κι ένα Επίπεδο Ενεργοποίησης εκτός από το τελευταίο Πλήρως Συνδεδεμένο Επίπεδο το οποίο ακολουθεί ένα softmax επίπεδο ώστε να προκύψουν οι πεποιθήσεις του μοντέλου για κάθε κατηγορία [10].

Η ικανότητα μάθησης των Συνελικτικών Νευρωνικών Δικτύων έχει βελτιωθεί δραματικά λόγω εκμετάλλευσης διάφορων δομικών τροποποιήσεων. Τα τελευταία χρόνια, η βελτίωση στην απόδοση των μοντέλων έχει επιτευχθεί αντικαθιστώντας τη συμβατική αρχιτεκτονική των επιπέδων με πιο περίπλοκα blocks.

Ιστορικά, οι πιο γνωστές και ενδιαφέρουσες αρχιτεκτονικές Συνελικτικών Νευρωνικών Δικτύων οι οποίες σχεδιάστηκαν για Κατηγοριοποίηση Εικόνας είναι:

- **LeNet** [14]. Η πιο γνωστή αρχιτεκτονική από τον Yann LeCun, ο οποίος ανέπτυξε τις πρώτες επιτυχημένες εφαρμογές Συνελικτικών Δικτύων τη δεκαετία του 1990.
- **AlexNet** [15]. Η αρχιτεκτονική που έκανε δημοφιλή τη χρήση Συνελικτικών Δικτύων στον τομέα της Όρασης Υπολογιστών.
- **ZF Net** [16]. Βασίστηκε πάνω στο AlexNet κάνοντας μικρές διορθώσεις στις υπερπαραμέτρους.
- **GoogleLeNet** [17]. Γνωστό και ως Inception, ήταν Νικητής της πρόκλησης μεγάλης κλίμακας εικονικής αναγνώρισης ImageNet το 2014 (ILSVRC2014). Κατάφερε να μειώσει δραματικά τον αριθμό των παραμέτρων, σε μόλις 4 εκατομμύρια, αξιοποιώντας διάφορες τεχνικές. Συγκριτικά, το AlexNet για παράδειγμα χρησιμοποιεί 60 εκ. παραμέτρους.
- **VGGNet** [18]. Ήρθε δεύτερο στην πρόκληση μεγάλης κλίμακας εικονικής αναγνώρισης ImageNet το 2014 (ILSVRC2014). Έδειξε ότι το βάθος του μοντέλου είναι κρίσιμο για την καλή απόδοση. Οι περισσότερες από τις 140 εκ. παραμέτρους του βρίσκονται στα πρώτα Πλήρως Συνδεδεμένα Επίπεδα και αργότερα βρέθηκε ότι αυτά μπορούν να αφαιρεθούν χωρίς να μειωθεί η απόδοση.
- **ResNet** [19]. Είναι απ' τα πιο προηγμένα μοντέλα από πλευράς τεχνικών αρχιτεκτονικής και μέχρι και σήμερα από τις βασικές επιλογές για πρακτική χρήση Συνελικτικών Δικτύων [10].
- **SqueezeNet** [20]. Δημιουργήθηκε το 2016 και επιτυγχάνει παρόμοια ακρίβεια με το AlexNet αλλά με 50 φορές λιγότερες παραμέτρους.
- **Xception** [21]. Μια βελτιωμένη μορφή του Inception η οποία κάνει χρήση τροποποιημένης συνέλιξης διαχωρίσιμης κατά βάθος.
- **EfficientNet** [22]. Κλιμακώνει ομοιόμορφα το πλάτος, το βάθος και την ανάλυση του δικτύου με βάση ένα σύνολο προεπιλεγμένων συντελεστών κλιμάκωσης.
- **MobileNet** [23]. Μικρά μοντέλα με μικρό χρόνο απόκρισης και χαμηλές απαιτήσεις ενέργειας τα οποία έχουν παραμετροποιηθεί ώστε να ικανοποιούν διάφορες περιπτώσεις περιορισμένων υπολογιστικών πόρων, όπως αυτές των κινητών συσκευών.
- **NASNet** [24]. Επιτυγχάνει αξιοσημείωτη ακρίβεια θέτοντας το πρόβλημα της εύρεσης της βέλτιστης αρχιτεκτονικής σαν ένα πρόβλημα Ενισχυτικής Μάθησης.

2.4 Κατανεμημένη Μηχανική Μάθηση

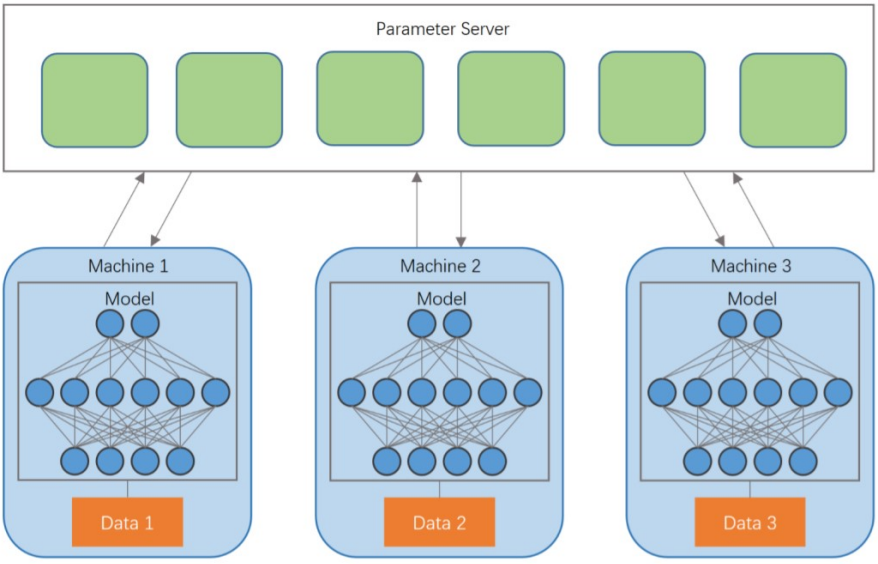
Η βασική ιδέα της Μηχανικής Μάθησης είναι η χρήση μεγάλων συνόλων δεδομένων για τη δημιουργία ενός μοντέλου το οποίο θα ανταποκρίνεται καλά σε εισόδους που δεν έχει

ξαναδεί. Με την αύξηση του όγκου δεδομένων και της πολυπλοκότητας των μοντέλων, γίνεται όλο και πιο δύσκολο να ολοκληρωθούν διεργασίες Μηχανικής Μάθησης σε μία μόνο μηχανή. Για τη διευθέτηση αυτού του προβλήματος αναπτύχθηκε η Κατανεμημένη Μηχανική Μάθηση (Distributed ML). Μια τυπική διεργασία Κατανεμημένης Μηχανικής Μάθησης ολοκληρώνεται μέσα από τη συνεργασία πολλών εξυπηρετητών (servers). Η πρώτη ιδέα για Κατανεμημένη Βαθιά Μάθηση χωρίς κοινή χρήση συνόλων δεδομένων προτάθηκε το 2015 από ερευνητές της Google.

Με την εξέλιξη των κινητών συσκευών και την αύξηση του αριθμού τους, είναι εφικτή η δημιουργία ενός ολοκληρωμένου και συμπαγούς συστήματος Κατανεμημένης Μηχανικής Μάθησης σε Κινητές Συσκευές (Mobile Distributed Machine Learning) το οποίο θα μπορούσε να ελαττώσει το φόρτο εργασίας των εξυπηρετητών [4].

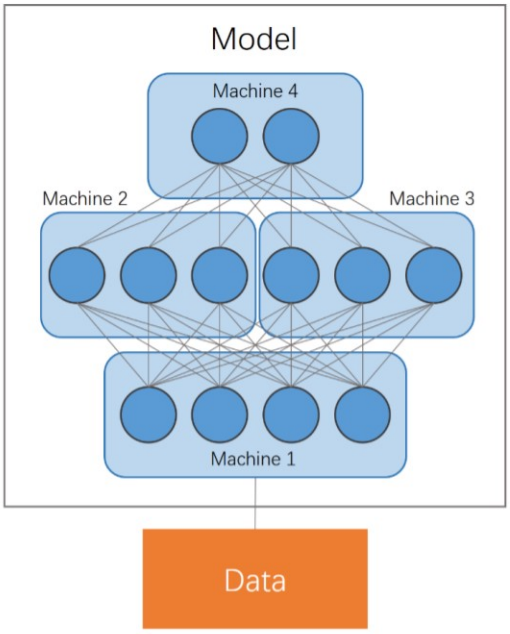
2.4.1 Παραδοσιακή Κατανεμημένη Μηχανική Μάθηση

Τα τελευταία χρόνια, τα σύνολα δεδομένων για διεργασίες Μηχανικής Μάθησης είναι τόσο μεγάλα που δεν μπορούν να αποθηκευτούν και να επεξεργαστούν από μια μόνο υπολογιστική μηχανή. Για την επίλυση αυτού του προβλήματος έχουν προταθεί σχήματα παραλληλισμού σε επίπεδο δεδομένων (data parallelism). Σε αυτά τα σχήματα, το σύνολο δεδομένων διαχωρίζεται σε υποσύνολα τα οποία αποθηκεύονται σε διαφορετικές μηχανές. Κάθε μηχανή έχει ένα αντίγραφο του μοντέλου το οποίο εκπαιδεύει στο τοπικό της υποσύνολο των δεδομένων. Μετά από αρκετές επαναλήψεις εκπαίδευσης, τα τοπικά μοντέλα που βρίσκονται στις μηχανές καταλήγουν να διαφέρουν μεταξύ τους. Οι πληροφορία από τα επιμέρους μοντέλα συγκεντρώνεται σε ένα εξυπηρετητή παραμέτρων (parameter server), ο οποίος παράγει ένα τελικό καθολικό μοντέλο (Global Model). Αυτή η μέθοδος καλείται *συγκέντρωση δεδομένων* (data aggregation) και λύνει το πρόβλημα της αποθήκευσης μεγάλου όγκου δεδομένων, ενώ παράλληλα βελτιώνει την αποδοτικότητα της διαδικασίας εκπαίδευσης. Το σχήμα παραλληλισμού σε επίπεδο δεδομένων φαίνεται στην Εικόνα 2.6.



Εικόνα 2.6: *Σχήμα παραλληλισμού σε επίπεδο δεδομένων*

Αντίστοιχα, ένα μοντέλο Μηχανικής Μάθησης μπορεί να είναι τόσο πολύπλοκο που δεν είναι εφικτό να εκπαιδευτεί και να εκτελεστεί σε μία μόνο μηχανή. Κατανεμημένα Βαθιά Νευρωνικά Δίκτυα προσπαθούν να λύσουν αυτό το πρόβλημα υιοθετώντας μεθόδους παραλληλισμού σε επίπεδο μοντέλου (model parallelism), οι οποίες χρησιμοποιούνται για την εκπαίδευση μεγάλων μοντέλων με δισεκατομμύρια παραμέτρους. Σε αυτό το σχήμα, κάθε μηχανή έχει αποθηκευμένο ένα μικρό μέρος του συνολικού μοντέλου. Κατά τη διάρκεια της εκπαίδευσης, τα δεδομένα περνάνε από κάθε μηχανή ώστε να επεξεργαστούν από τα τοπικά υπο-μοντέλα. Στις περισσότερες περιπτώσεις, κάθε γύρος εκπαίδευσης χρειάζεται τη συνεργασία κάθε μηχανής, οπότε η διαδικασία χρειάζεται να γίνει σειριακά καθώς τα δεδομένα εισόδου μια μηχανής εξαρτώνται από τα δεδομένα εξόδου μιας άλλης. Το σχήμα παραλληλισμού σε επίπεδο μοντέλου φαίνεται στην Εικόνα 2.7. Σε σχέση με τον παραλληλισμό σε επίπεδο δεδομένων, ο παραλληλισμός σε επίπεδο μοντέλου είναι πιο περίπλοκος και πιο δύσκολος να εφαρμοστεί λόγω της ισχυρής συνεργασίας ανάμεσα στις μηχανές. Και τα δύο σχήματα όμως είναι σημαντικά καθώς λύνουν δύο σοβαρούς περιορισμούς της Κατανεμημένης Μηχανικής Μάθησης [4].



Εικόνα 2.7: *Σχήμα παραλληλισμού σε επίπεδο μοντέλου*

2.4.2 Βαθιά Μάθηση σε Εξυπηρετητές

Τα δεδομένα που παράγονται από τις κινητές συσκευές και τις συσκευές του διαδικτύου (Internet of Things - IoT) έχουν επιτρέψει την εκπαίδευση εξαιρετικά αποτελεσματικών μοντέλων και η αξιοποίησή τους είναι πια απαραίτητη για μοντέρνες εφαρμογές Βαθιάς Μάθησης. Η πιο συνηθισμένη αρχιτεκτονική για την επεξεργασία αυτών των δεδομένων είναι κεντρικοποιημένης μορφής. Τα δεδομένα που παράγονται σε κινητές συσκευές, αυτοκίνητα, κάμερες παρακολούθησης και άλλες συσκευές IoT προωθούνται και συγκεντρώνονται σε έναν ισχυρό εξυπηρετητή, ο οποίος βρίσκεται στο υπολογιστικό νέφος ή στην άκρη του

δικτύου. Εκεί, με τη χρήση αυτών των δεδομένων αλλά και ισχυρών επεξεργαστών, εκτελείται πρώτα η εκπαίδευση Βαθιών Νευρωνικών Δικτύων και στη συνέχεια με αντίστοιχο τρόπο εκτελείται και η συμπερασματολογία [25].

Αυτή η πρακτική δημιουργεί βασικά προβλήματα όπως είναι ο μεγάλος χρόνος μεταφοράς που μπορεί να είναι απαγορευτικός για εφαρμογές πραγματικού χρόνου, το υψηλό κόστος, η αποτυχία προστασίας της ιδιωτικότητας των δεδομένων τα οποία είναι ευάλωτα σε επιθέσεις κατά τη μεταφορά τους και η συνεχής απαίτηση για σύνδεση στο διαδίκτυο από την κινητή συσκευή [26].

2.4.3 Βαθιά Μάθηση σε Κινητές Συσκευές

Όπως αναφέρθηκε και νωρίτερα, εμπνευσμένοι από τις εξαιρετικές επιδόσεις της Βαθιάς Μάθησης αλλά και λόγω του όγκου των δεδομένων που παράγονται στις κινητές συσκευές καθημερινά, οι ερευνητές ωθούν τις εφαρμογές της στις κινητές συσκευές. Με αυτόν τον τρόπο μπορούν να αποφευχθούν οι καθυστερήσεις, η έκθεση των δεδομένων σε ενδεχόμενες επιθέσεις και η ανάγκη για σύνδεση στο διαδίκτυο.

Λόγω της περιορισμένης υπολογιστικής ισχύος, μνήμης και μπαταρίας των κινητών συσκευών, η εκπαίδευση Βαθιών Νευρωνικών Δικτύων από την αρχή (from scratch) σε αυτές είναι πρακτικά αδύνατη. Ωστόσο, πρόσφατα έχουν γίνει προσπάθειες για τη δημιουργία προσωποποιημένων μοντέλων με χρήση Μεταφοράς Μάθησης (Transfer Learning) αποκλειστικά στην κινητή συσκευή του χρήστη τόσο σε ερευνητικό επίπεδο [27][28][29], όσο και σε επίπεδο εργαλείων [30]. Η Μεταφορά Μάθησης βασίζεται στο γεγονός ότι ένα πρόβλημα με λίγα δεδομένα ("data-poor" task) μπορεί να αξιοποιήσει τη γνώση ενός νευρωνικού δικτύου το οποίο έχει εκπαιδευτεί σε ένα παρόμοιο πρόβλημα με πάρα πολλά δεδομένα ("data-rich" task). Συγκεκριμένα, αυτό που γίνεται συνήθως είναι τα πρώτα επίπεδα του δικτύου που είναι υπεύθυνα για την εξαγωγή χαρακτηριστικών να παραμένουν ως έχουν, ενώ τα τελευταία επίπεδα να εκπαιδεύονται ξανά πάνω στα λίγα δεδομένα του προβλήματος που μας ενδιαφέρει. Επομένως, με αυτόν τον τρόπο μπορούν να προκύψουν μοντέλα προσαρμοσμένα στους χρήστες, ακόμη και με περιορισμένα δεδομένα εκπαίδευσης και υπολογιστικούς πόρους, διατηρώντας παράλληλα την ιδιωτικότητά τους.

Η εκτέλεση συμπερασματολογίας από την άλλη, είναι γενικά εφικτή, όμως μπορεί σε κάποιες περιπτώσεις να μην υπάρχει εγγύηση για την ποιότητα υπηρεσίας. Αυτό μπορεί να συμβεί επειδή (α) υπάρχει μεγάλη ετερογένεια μεταξύ των κινητών συσκευών, οπότε ένα μοντέλο Βαθιάς Μάθησης δεν παρουσιάζει τα ίδια αποτελέσματα σε όλες τις συσκευές που κυκλοφορούν και (β) μια κινητή συσκευή είναι ένα δυναμικό περιβάλλον, με τον επεξεργαστή πολλές φορές να πρέπει να εκτελεί διάφορες άλλες διεργασίες παράλληλα με την εφαρμογή Βαθιάς Μάθησης, επομένως το φορτίο (load) του να μην είναι σταθερό και να επηρεάζει παραμέτρους όπως είναι η θερμοκρασία του ή συχνότητα ρολογιού.

Τόσο η χρήση Μεταφοράς Μάθησης, όσο και η εκτέλεση συμπερασματολογίας στη συσκευή είναι εφικτά τα τελευταία χρόνια χάρη:

- Στην αναβάθμιση του υλικού (hardware) των κινητών συσκευών. Εκτός από τους καλύτερους επεξεργαστές, την μεγαλύτερη μνήμη και την καλύτερη διαχείριση της μπαταρίας, οι σύγχρονες κινητές συσκευές είναι εφοδιασμένες με Μονάδες Επεξερ-

γασίας Νευρώνων (Neural Processing Units - NPUs), οι οποίες έχουν σχεδιαστεί και κατασκευαστεί αποκλειστικά για την επιτάχυνση της εκτέλεσης βαθιών νευρωνικών δικτύων.

- Σε μεθόδους συμπίεσης και βελτιστοποίησης των μοντέλων Βαθιάς Μάθησης. Ένα μοντέλο μπορεί να συμπυκνωθεί είτε σχεδιάζοντάς το από την αρχή για αυτόν τον σκοπό, είτε χρησιμοποιώντας τεχνικές όπως είναι το «κλάδεμα» (pruning) και η κβαντοποίηση (quantization). Αυτές οι μέθοδοι μπορεί να μειώσουν το μέγεθος του μοντέλου κατά 50% ή 75% καθώς και να επιταχύνουν τη συμπερασματολογία κατά 2 ή 3 φορές εισάγοντας όμως και μια μικρή μείωση στην ακρίβεια.

2.4.4 Κατανεμημένο Ζεύγος Νευρωνικών Δικτύων

Ένα Κατανεμημένο Ζεύγος Νευρωνικών Δικτύων είναι ένα υβριδικό κατανεμημένο σύστημα το οποίο δίνει τη δυνατότητα χρήσης διαφορετικών Νευρωνικών Δικτύων μέσα από την επιλεκτική εκτέλεση μοντέλων είτε στον εξυπηρετητή είτε στην κινητή συσκευή:

- Η εκτέλεση συμπερασματολογίας στον εξυπηρετητή προσφέρει υψηλή ακρίβεια, καθώς τα μοντέλα είναι πιο πολύπλοκα και αξιοποιούν τους ισχυρότερους υπολογιστικούς πόρους που παρέχει ο εξυπηρετητής.

- Η εκτέλεση συμπερασματολογίας στην κινητή συσκευή αποφέρει χαμηλότερη ακρίβεια αλλά αποφεύγει τη μεταφορά δεδομένων, κάνοντας χρήση πιο απλών και συμπυκνωμένων μοντέλων.

Η δυνατότητα απόφασης του τρόπου εκτέλεσης της συμπερασματολογίας είναι το βασικότερο πλεονέκτημα ενός τέτοιου συστήματος. Η απόφαση αυτή μπορεί να λαμβάνεται με στόχο τη βελτιστοποίηση διαφόρων μετρικών, όπως η ακρίβεια και ο χρόνος απόκρισης, χρησιμοποιώντας πληθώρα μεταβλητών εισόδου οι οποίες μοντελοποιούν τα δυναμικά χαρακτηριστικά του συστήματος και τις ανάγκες του χρήστη. Καθώς το σύστημα συνδυάζει τους δύο παραδοσιακούς τρόπους εκτέλεσης συμπερασματολογίας, στον εξυπηρετητή (on-server) και στη συσκευή (on-device), θα πρέπει να ικανοποιεί και τους διάφορους περιορισμούς τους, όπως η μέγιστη χρήση μνήμης ή η μπαταρία. Ο εξυπηρετητής μπορεί να βρίσκεται στην άκρη του δικτύου (edge) με χαμηλότερη υπολογιστική ισχύ και μνήμη αλλά μικρότερη καθυστέρηση στη μεταφορά δεδομένων ή στο υπολογιστικό νέφος (cloud) με δυνατότητα για εκτέλεση πιο απαιτητικών υπολογισμών αλλά και με μεγαλύτερη καθυστέρηση στη μεταφορά δεδομένων. Πρόσφατες τεχνολογικές πρόοδοι όπως είναι το 5G και οι αναβαθμίσεις των κινητών συσκευών καθιστούν την ανάπτυξη ενός τέτοιου συστήματος εκτός από αναγκαία και εφικτή.

Κεφάλαιο 3

Τεχνολογίες - Εργαλεία

Στο κεφάλαιο αυτό γίνεται μια σύντομη παρουσίαση των τεχνολογιών που χρησιμοποιήθηκαν για την εκπόνηση της διπλωματικής εργασίας.

3.1 Python

Η Python είναι μια διαδραστική, αντικειμενοστραφής προγραμματιστική γλώσσα διερμηνέα η οποία ενσωματώνει δομοστοιχεία (modules), εξαιρέσεις (exceptions), δυναμική τυπολόγηση (dynamic typing), δυναμικούς τύπους δεδομένων υψηλού επιπέδου (very high level dynamic data types) και κλάσεις (classes). Εκτός από αντικειμενοστραφή προγραμματισμό, υποστηρίζει και άλλα προγραμματιστικά πρότυπα όπως ο διαδικαστικός (procedural) και ο συναρτησιακός (functional) προγραμματισμός. Η Python συνδυάζει αξιοσημείωτη δύναμη με εξαιρετικά καθαρό συνακτικό. Καθώς είναι επεκτάσιμη στη C και στη C++, μπορεί να χρησιμοποιηθεί ως γλώσσα επέκτασης για εφαρμογές που χρειάζονται μια προγραμματίσιμη διεπαφή. Τέλος, ο κώδικας της Python είναι εύκολο να μεταφερθεί μεταξύ λειτουργικών συστημάτων αφού υποστηρίζεται σε πολλές παραλλαγές Unix όπως Linux, macOS και φυσικά Windows [31].

Η Python συγκρίνεται συχνά με άλλες προγραμματιστικές γλώσσες διερμηνέα όπως η Java, η JavaScript ή η C++.

- Σε σύγκριση με τη Java, τα προγράμματα της Python αναμένεται να τρέχουν πιο αργά, αλλά χρειάζονται πολύ λιγότερο χρόνο να αναπτυχθούν. Τα προγράμματα της Python είναι τυπικά 3 με 5 φορές μικρότερα από τα αντίστοιχα προγράμματα της Java. Αυτή η διαφορά αποδίδεται στους υψηλού επιπέδου τύπους δεδομένων καθώς και στη δυναμική τυπολόγηση της Python. Αυτό έχει σαν αποτέλεσμα η εκτέλεση της Python να χρειάζεται να δουλέψει πολύ πιο σκληρά σε σχέση με την αντίστοιχη εκτέλεση της Java. Γι' αυτούς τους λόγους, η Python λειτουργεί καλύτερα σαν συνδετική γλώσσα (glue language), ενώ η Java είναι πιο χρήσιμη σαν γλώσσα υλοποίησης χαμηλού επιπέδου. Οι δύο αυτές γλώσσες μπορούν να συνδυαστούν σε εφαρμογές και να συνεργαστούν εξαιρετικά.

- Το αντικειμενοστραφές υποσύνολο της Python είναι περίπου ισοδύναμο με αυτό της JavaScript. Όπως και η JavaScript, η Python υποστηρίζει ένα στυλ προγραμματισμού

το οποίο κάνει χρήση απλών συναρτήσεων και μεταβλητών χωρίς να εμπλέκει ορισμούς κλάσεων, σε αντίθεση με τη Java. Ενώ όμως η JavaScript σταματά εκεί, η Python υποστηρίζει ανάπτυξη πολύ μεγαλύτερων προγραμμάτων και καλύτερη επαναχρησιμοποίηση κώδικα μέσα από ένα πραγματικά αντικειμενοστραφές στυλ προγραμματισμού, όπου οι κλάσεις και η κληρονομικότητα παίζουν σημαντικό ρόλο.

- Σχεδόν όλα όσα ειπώθηκαν για τη Java ισχύουν και για τη C++, αλλά σε μεγαλύτερο βαθμό: ενώ τα προγράμματα της Python είναι περίπου 3 με 5 φορές μικρότερα από τα προγράμματα της Java, πολλές φορές είναι 5 με 10 φορές μικρότερα από τα αντίστοιχα προγράμματα που έχουν αναπτυχθεί σε C++. Η χρήση της Python ως συνδετικής γλώσσας για το συνδυασμό δομικών στοιχείων γραμμένων σε C++, αποφέρει συνήθως εξαιρετικά αποτελέσματα [32].

Τις τελευταίες δεκαετίες, η Python έχει γίνει η πρώτη επιλογή για πραγματοποίηση υπολογιστικής επιστημονικής έρευνας. Παρόλο που η Python δεν είχε σχεδιαστεί συγκεκριμένα για να εξυπηρετεί τις υπολογιστικές ανάγκες της επιστημονικής κοινότητας, σε πολύ μικρό χρονικό διάστημα από τη γέννησή της κατάφερε να προσελκύσει επιστήμονες και μηχανικούς. Παρά την εκφραστική της σύνταξη και την πλούσια συλλογή από τύπους δεδομένων, ήταν προφανές πως υπήρχε ανάγκη να προστεθεί σε αυτή τη συλλογή ένας τύπος δεδομένων μορφής πίνακα για αριθμητικές υπολογιστικές πράξεις.

Το 1995, σχηματίστηκε στην κοινότητα της Python η matrix-sig [33], μια ομάδα που είχε ως σκοπό τη δημιουργία ενός καινούριου τύπου δεδομένων με τις προδιαγραφές που αναφέρθηκαν. Τα πρώτα χρόνια, ο μεγάλος βαθμός αλληλεπίδρασης ανάμεσα στην ευρύτερη και στην επιστημονική κοινότητα της Python, είχε σαν αποτέλεσμα τον συνεχή εμπλουτισμό της γλώσσας με καινούρια γνωρίσματα και συντακτικό τα οποία εξυπηρετούσαν την επιστημονική κοινότητα.

Τα επόμενα πέντε χρόνια, μια μικρή αλλά αφοσιωμένη κοινότητα επιστημόνων και μηχανικών συνέχισε να βελτιώνει το Numeric, μια επέκταση της Python από τον Jim Hugunin, απόφοιτο του MIT, η οποία βασίστηκε πάνω στο αντικείμενο πίνακα (matrix object) του Jim Fulton. Αυτή η κοινότητα πέρα από τις βελτιώσεις του Numeric, ανέπτυσσε και διαμοιράζε επιπλέον πακέτα για επιστημονική υπολογιστική.

Την περίοδο του 2000, υπήρχε αυξανόμενο ενδιαφέρον για τη δημιουργία ενός πλήρους προγραμματιστικού περιβάλλοντος για επιστημονική υπολογιστική. Αυτό το ενδιαφέρον ώθησε την κοινότητα στη διεύρυνση της συλλογής των επεκτάσεων οι οποίες στόχευαν στην υλοποίηση αυτού του περιβάλλοντος. Τα επόμενα τρία χρόνια, διάφορες εξελίξεις βελτίωσαν την χρησιμότητα της Python ως προς την επιστημονική υπολογιστική. Οι Travis Oliphant, Eric Jones και Pearu Peterson ένωσαν τους κώδικες που είχαν γράψει και δημιούργησαν το πακέτο SciPy το οποίο παρείχε μια πρότυπη συλλογή από συνήθεις μαθηματικές πράξεις μαζί με τη δομή δεδομένων Numeric. Ο Fernando Perez ανέπτυξε την πρώτη εκδοχή του IPython, ενός προηγμένου διαδραστικού φλοιού αρκετά διαδεδομένου στην επιστημονική κοινότητα. Τέλος, ο John Hunter κυκλοφόρησε την πρώτη εκδοχή του matplotlib, την καθιερωμένη βιβλιοθήκη γραφικής αναπαράστασης για επιστημονική υπολογιστική.

Παρά τη χρησιμότητα του Numeric ως βάση για τα καινούρια πακέτα, ο βασικός του κώδικας ήταν δύσκολο να επεκταθεί, κάτι το οποίο επιβράδυνε σημαντικά την ανάπτυξη

αυτών των πακέτων. Λύση σε αυτό το πρόβλημα έδωσαν οι Perry Greenfield, Todd Miller και Rick White οι οποίοι ανέπτυξαν ένα καινούριο καινούριο πακέτο δομής δεδομένων μορφής πίνακα (array), το οποίο περιείχε πολλά πρωτοποριακά, χρήσιμα χαρακτηριστικά και ονομάστηκε numarray. Το χάσμα που δημιουργήθηκε στην κοινότητα ανάμεσα στο Numeric και στο numarray, κατάφερε να γεφυρώσει ο Travis Oliphant με την κυκλοφορία του NumPy, μιας σημαντικής εξέλιξης του Numeric η οποία υιοθέτησε τα πιο χρήσιμα χαρακτηριστικά του numarray. Αυτό το πακέτο είναι μέχρι και σήμερα το πιο διαδεδομένο πακέτο για μαθηματικές πράξεις στην Python. Από τότε η κοινότητα του SciPy μεγαλώνει ταχύτατα και η βασική συλλογή από εργαλεία βελτιώνεται και διευρύνεται με σταθερούς ρυθμούς [34].

Η Python, είναι μια εξαιρετική γλώσσα «πλοηγός» για επιστημονικούς κώδικες που έχουν γραφεί σε άλλες γλώσσες προγραμματισμού. Παρ' όλα αυτά, με επιπλέον βασικά εργαλεία, η Python μεταμορφώνεται σε μια γλώσσα υψηλού επιπέδου, κατάλληλη για επιστημονικό κώδικα ο οποίος είναι συχνά αρκετά γρήγορος και χρήσιμος αλλά και ευέλικτος ώστε να μπορεί να επιταχυνθεί με επιπλέον επεκτάσεις.

3.2 TensorFlow

Το TensorFlow είναι μια βιβλιοθήκη ανοιχτού λογισμικού για αριθμητική υπολογιστική η οποία επιταχύνει και διευκολύνει την ενσωμάτωση Μηχανικής Μάθησης συνδυάζοντας πλήθος μοντέλων και αλγορίθμων Μηχανικής και Βαθιάς Μάθησης. Παρέχει μια βολική διεπαφή προγραμματισμού εφαρμογών (API) κάνοντας χρήση της Python, ενώ εκτελεί αυτές τις εφαρμογές σε C++ υψηλής απόδοσης. Το TensorFlow επιτρέπει την εκπαίδευση και την εκτέλεση Βαθιών Νευρωνικών Δικτύων τα οποία χρησιμοποιούνται για πληθώρα εφαρμογών, όπως κατηγοριοποίηση εικόνων και βίντεο, εντοπισμό αντικειμένων σε εικόνες, εκτίμηση πόζας, κατάτμηση εικόνας, αναγνώριση δράσεων, αυτόνομα αυτοκίνητα, αναγνώριση φωνής, ερωταπαντήσεις, μετάφραση κειμένου, ανάλυση συναισθήματος, σύνοψη κειμένου, ανάλυση ιατρικών εικόνων, συστήματα συστάσεων, πρόβλεψη χρονοσειρών, κ.α. Με το TensorFlow οι προγραμματιστές μπορούν να δημιουργήσουν γραφήματα ροής δεδομένων, τα οποία εξηγούν πώς κινούνται τα δεδομένα. Κάθε κόμβος στο γράφημα αναπαριστά μια μαθηματική πράξη και κάθε σύνδεση μεταξύ των κόμβων ένα διάνυσμα δεδομένων πολλών διαστάσεων ή αλλιώς έναν τανυστή (tensor). Οι εφαρμογές του TensorFlow μπορούν να εκτελεστούν οπουδήποτε είναι βολικό: σε τοπικές μηχανές, σε ένα σύμπλεγμα στο νέφος, σε συσκευές iOS ή Android, σε Κεντρικές Μονάδες Επεξεργασίας (CPUs) ή Μονάδες Επεξεργασίας Γραφικών (GPUs). Το TensorFlow επιτρέπει στον προγραμματιστή να εστιάσει στη γενικότερη λογική της εφαρμογής χωρίς να χρειάζεται να ασχοληθεί με τις λεπτομέρειες της εφαρμογής του αλγορίθμου. Οι κύριοι ανταγωνιστές του TensorFlow είναι το PyTorch, το οποίο είναι παρόμοιο με το TensorFlow όμως το TensorFlow είναι αποδοτικότερο σε μεγάλα έργα, το CNTK και το Apache MXNET τα οποία είναι αρκετά δύσκολο να μάθει και να εφαρμόσει κανείς [35].

Το TensorFlow αναπτύχθηκε αρχικά από ερευνητές και μηχανικούς, που δούλευαν στην ομάδα Google Brain μέσα στον οργανισμό Machine Intelligence Research της Google, ώστε να τους επιτρέψει να πραγματοποιήσουν έρευνα πάνω στη Μηχανική Μάθηση και τα βαθιά νευρωνικά δίκτυα. Το ανοικτό λογισμικό TensorFlow είναι ένα περιεκτικό και ευέλικτο πε-

ριβάλλον εργαλείων, βιβλιοθηκών και πόρων της κοινότητας, το οποίο επιτρέπει σε ερευνητές να εξελίξουν τις τεχνολογίες αιχμής της Μηχανικής Μάθησης και στους προγραμματιστές, να δημιουργήσουν και να χρησιμοποιήσουν εφαρμογές που βασίζονται στη Μηχανική Μάθηση [36].

3.3 Keras

Το Keras είναι μια υψηλού επιπέδου βιβλιοθήκη της Python για Βαθιά Μάθηση, εύκολη στην κατανόηση και στη χρήση της. Επιτρέπει στον προγραμματιστή να εστιάσει στις βασικές αρχές της Βαθιάς Μάθησης όπως η δημιουργία επιπέδων (layers) του νευρωνικού δικτύου [37]. Επίσης παρέχει έτοιμες πολλές από τις πιο γνωστές αρχιτεκτονικές Συνελικτικών Νευρωνικών Δικτύων, όπως οι VGG16, ResNet, MobileNet, NasNetLarge κ.α. [38]. Το Keras διαμοιράζεται πια μαζί με το TensorFlow και μπορεί να χρησιμοποιηθεί από εκεί [39].

3.4 ImageNet

Το ImageNet είναι μια μεγάλης κλίμακας βάση δεδομένων που περιέχει εικόνες. Παρέχει δεκάδες εκατομμύρια εικόνες με ετικέτες οι οποίες έχουν οργανωθεί με βάση τη σημασιολογική ιεραρχεία του WordNet [40].

Η πρόκληση μεγάλης κλίμακας εικονικής αναγνώρισης ImageNet (ILSVRC) [41], είναι σημείο αναφοράς στην κοινότητα της Μηχανικής Μάθησης. Απαιτεί την κατηγοριοποίηση και αναγνώριση αντικειμένων σε εκατοντάδες κατηγορίες αντικειμένων και σε εκατομμύρια εικόνες. Διοργανώνονταν ετησίως από το 2010 μέχρι και το 2017, προσελκύοντας περισσότερα από 50 ιδρύματα, τα οποία ανέπτυξαν γι' αυτή πολλά από τα πιο γνωστά και πολυχρησιμοποιημένα μοντέλα [42].

Κεφάλαιο 4

Μοντελοποίηση Συστήματος

Στο κεφάλαιο αυτό παρουσιάζεται η μοντελοποίηση του συστήματος. Αρχικά γίνεται ανάλυση των συναρτήσεων που το αποτελούν κι έπειτα παρουσιάζεται η ροή εκτέλεσης του προγράμματος.

4.1 Περιγραφή Συναρτήσεων

Στην πρώτη ενότητα αναλύονται οι συναρτήσεις που απαρτίζουν το σύστημα. Γίνεται μια σύντομη παρουσίαση των στοιχείων εισόδου, του τρόπου λειτουργίας και των στοιχείων εξόδου κάθε συνάρτησης. Υπάρχουν συνολικά εννιά βασικές συναρτήσεις και τρεις βοηθητικές.

4.1.1 Μέτρηση Πεποίθησης

Η συνάρτηση μέτρησης πεποίθησης (conf_metric) σχεδιάστηκε ώστε να αποφασίζει αν υπάρχει εμπιστοσύνη στο αποτέλεσμα της κατηγοριοποίησης που έχει εκτελέσει το μοντέλο της κινητής συσκευής, δηλαδή προσπαθεί να προσομοιάσει την πιθανότητα σωστής πρόβλεψης του μοντέλου. Αυτό επιτυγχάνεται χρησιμοποιώντας τις τιμές πεποίθησης του αποτελέσματος της συμπερασματολογίας. Τα τρία στοιχεία εισόδου είναι:

- **Πρόβλεψη (prediction)**, μια λίστα με τιμές πεποίθησης των 5 πιο σίγουρων αποτελεσμάτων της συμπερασματολογίας.
- **Όριο (threshold)**, μια τιμή $t \in [0, 1]$ η οποία ορίζει την αυστηρότητα της μέτρησης.
- **Λειτουργία (mode)**, μεταβλητή που ορίζει τη μετρική με βάση την οποία λαμβάνεται η απόφαση.

Χάρις την ευελιξία του συστήματος είναι πολύ εύκολο να προστεθούν και να επιλεχθούν διάφορες μετρικές. Μετά από μετρήσεις, η μετρική που έφερε τα καλύτερα αποτελέσματα είναι η Best-vs-Second Best (BvSB) [40], η οποία είναι και η προκαθορισμένη τιμή για τη μεταβλητή mode. Αυτή η μετρική υπολογίζει τη διαφορά ανάμεσα στις τιμές των πεποιθήσεων της πρώτης και της δεύτερης πρόβλεψης της συμπερασματολογίας και ελέγχει την ανίσωση:

$$P_1 - P_2 > c_{thresh}$$

όπου P_1 η τιμή πεποίθησης της πρώτης πρόβλεψης, P_2 η τιμή πεποίθησης της δεύτερης πρόβλεψης και c_{thresh} το όριο πεποίθησης. Αν αυτή η διαφορά είναι μεγαλύτερη από το threshold, τότε θεωρείται ότι το αποτέλεσματα της συμπερασματολογίας είναι ικανοποιητικά πιθανό να είναι σωστό. Όσο πιο μικρή η τιμή του threshold, τόσο πιο εύκολο να θεωρηθεί ικανοποιητικά πιθανό το αποτέλεσμα να είναι σωστό. Η απόφαση αποθηκεύεται σε μια δυαδική τιμή (boolean), η οποία αν είναι αληθής δηλώνει πως μπορεί να θεωρηθεί σωστή η κατηγοριοποίηση, ενώ αν είναι ψευδής πως η κατηγοριοποίηση είναι πιθανότατα λανθασμένη.

4.1.2 Υπολογισμός Χρόνου Μεταφοράς

Η συνάρτηση υπολογισμού χρόνου μεταφοράς (data_trans_time) εκτιμά το χρόνο που χρειάζεται για να μεταφερθεί ένα αρχείο εικόνας από την κινητή συσκευή στον εξυπηρετητή (server). Δέχεται δύο στοιχεία εισόδου:

- **Το μέγεθος του αρχείου σε bits (data_size)**, το οποίο έχει προϋπολογιστεί από τη βοηθητική συνάρτηση «Μέγεθος Εικόνας» (βλ. υποενότητα 4.1.10).
- **Τα στοιχεία της σύνδεσης (connection)**, μια λίστα μήκους 3. Το πρώτο στοιχείο της λίστας είναι η καθυστέρηση (latency) σε δευτερόλεπτα, το δεύτερο το εύρος ζώνης (bandwidth) σε bits ανά δευτερόλεπτο και το τρίτο το εύρος ζώνης που διατίθεται (avail_bandwidth) όπου $BW_{av} \in [0, 1]$.

Η συνάρτηση υπολογίζει το χρόνο που χρειάζεται για να μεταφερθούν τα δεδομένα (elapsed time) με βάση τον τύπο:

$$t_e = l + d/(BW_{av} \cdot BW)$$

όπου l είναι η καθυστέρηση της σύνδεσης (latency), d το μέγεθος του αρχείου (data), BW_{av} το εύρος ζώνης που διατίθεται (available bandwidth) και BW το εύρος ζώνης της σύνδεσης (bandwidth).

4.1.3 Ακριβής Χρόνος Συμπερασματολογίας

Η συνάρτηση υπολογισμού του ακριβούς χρόνου συμπερασματολογίας στον εξυπηρετητή (get_server_inf_time_exact) υπολογίζει το μέσο χρόνο που χρειάζεται για να εκτελεστεί η συμπερασματολογία μιας εικόνας στον εξυπηρετητή. Τα δύο στοιχεία εισόδου είναι:

- Το **νευρωνικό μοντέλο που χρησιμοποιεί ο εξυπηρετητής (server_model)**.
- Τα **χαρακτηριστικά του εξυπηρετητή (server_specs)**, μια λίστα μήκους 3. Το πρώτο στοιχείο της λίστας είναι η υπολογιστική δύναμη που διαθέτει ο εξυπηρετητής σε TeraFLOP/s. Το δεύτερο στοιχείο της λίστας είναι η μνήμη τυχαίας προσπέλασης (RAM) που διαθέτει ο εξυπηρετητής σε Gigabytes (GB). Το τρίτο είναι ο φόρτος εργασίας του εξυπηρετητή, $L_s \in [1, \infty)$.

Η συνάρτηση έχει πρόσβαση σε μια βάση δεδομένων όπου είναι αποθηκευμένοι συνδυασμοί εξυπηρετητών, νευρωνικών μοντέλων και μέσων χρόνων εκτέλεσης συμπερασματολογίας σε μια εικόνα. Εξετάζει αν ο συνδυασμός εξυπηρετητή και μοντέλου που δόθηκαν υπάρχει ήδη στη βάση δεδομένων και αν υπάρχει, επιστρέφει το μέσο χρόνο συμπερασματολογίας ο οποίος είναι αποθηκευμένος σε αυτή. Αν δεν υπάρχει, ο εξυπηρετητής τον υπολογίζει και προσθέτει τον καινούριο συνδυασμό μαζί με το μέσο χρόνο συμπερασματολογίας που μόλις υπολογίστηκε στη βάση δεδομένων.

4.1.4 Συγκριτικός Χρόνος Συμπερασματολογίας

Η συνάρτηση συγκριτικού χρόνου συμπερασματολογίας στον εξυπηρετητή (get_server_inf_time_scaling) υπολογίζει τον μέσο χρόνο που χρειάζεται για να εκτελεστεί συμπερασματολογία σε μία εικόνα συγκρίνοντας τα στοιχεία του δοθέντος εξυπηρετητή με τα στοιχεία ενός εξυπηρετητή αναφοράς. Δέχεται δύο εισόδους:

- Το **νευρωνικό μοντέλο που χρησιμοποιεί ο εξυπηρετητής (server_model)**.
- Τα **χαρακτηριστικά του εξυπηρετητή (server_specs)**, όπως στην 4.1.3.

Η συνάρτηση βρίσκει το αντίστοιχο μοντέλο σε μια λίστα η οποία περιέχει τιμές μέσων χρόνων συμπερασματολογίας σε μία εικόνα για διάφορα μοντέλα. Η λίστα αυτή δημιουργήθηκε χρησιμοποιώντας σύστημα με μονάδα επεξεργασίας γραφικών NVIDIA TITAN Xp η οποία διαθέτει 12.15 TeraFLOP/s και τυχαία μνήμη προσπέλασης 16 GB. Με βάση το μέσο χρόνο συμπερασματολογίας της λίστας και τα χαρακτηριστικά του εξυπηρετητή που δόθηκαν, υπολογίζεται ένας συγκριτικός χρόνος μέσης συμπερασματολογίας για το συνδυασμό εξυπηρετητή-μοντέλου που δόθηκε.

Ο τρόπος υπολογισμού του μέσου χρόνου που χρειάζεται για να εκτελεστεί η συμπερασματολογία σε μια εικόνα με σύγκριση δεν είναι τόσο ακριβής όσο ο προηγούμενος αλλά υπερτερεί σε ταχύτητα και δέσμευση μνήμης.

4.1.5 Απόφαση Εκτέλεσης Συμπερασματολογίας στην Κινητή Συσκευή

Η συνάρτηση απόφασης εκτέλεσης συμπερασματολογίας στην κινητή συσκευή (stay_on_mobile), λαμβάνει υπόψιν τα χαρακτηριστικά της συσκευής και ταυτόχρονα ελέγχοντας αν πληρούνται οι απαιτήσεις απόδοσης αποφασίζει αν επιτρέπεται η εκτέλεση συμπερασματολογίας στη συσκευή. Δέχεται τέσσερα στοιχεία εισόδου κι επιστρέφει μια δυαδική τιμή, ανάλογα με το αν αυτή η τιμή είναι ψευδής ή αληθής τότε αντίστοιχα επιτρέπεται ή όχι η εκτέλεση συμπερασματολογίας στη συσκευή.

Τα στοιχεία εισόδου είναι:

- Τα **χαρακτηριστικά της συσκευής (mobile_specs)**, μια λίστα μήκους 4. Το πρώτο στοιχείο της λίστας είναι ο χρόνος που χρειάζεται για να εκτελεστεί συμπερασματολογία σε μια εικόνα σε δευτερόλεπτα. Το δεύτερο στοιχείο είναι ο φόρτος εργασίας (load) του επεξεργαστή $L_p \in [1, \infty)$, όπου $p \in P = \{CPU, GPU, NPU, ...\}$. Το τρίτο στοιχείο είναι η θερμοκρασία της συσκευής σε βαθμούς κελσίου και το τελευταίο είναι η υπολειπόμενη μπαταρία $B \in [0, 100]$.

- Το **όριο για το χρόνο συμπερασματολογίας (time_thresh)**.
- Το **όριο για τη θερμοκρασία (temp_thresh)**.
- Το **όριο για τη μπαταρία (battery_thresh)**.

Ο χρόνος εκτέλεσης συμπεραματολογίας στην κινητή συσκευή μπορεί να υπολογιστεί με χρήση του TFLite Model Benchmark Tool [43]. Η συνάρτηση ελέγχει τις παρακάτω ανισώσεις:

$$L_p \cdot t_{inf,m} < t_{thresh,m}$$
$$T < T_{thresh}$$
$$B > B_{thresh}$$

όπου $t_{inf,m}$ είναι ο χρόνος εκτέλεσης συμπερασματολογίας στην κινητή συσκευή, $t_{thresh,m}$ το όριο του χρόνου συμπερασματολογίας, T η θερμοκρασία της συσκευής, T_{thresh} το όριο της και B_{thresh} το όριο της μπαταρίας. Αν ισχύουν και οι τρεις, επιτρέπεται να εκτελεστεί η συμπερασματολογία στη συσκευή, δηλαδή η τιμή εξόδου είναι αληθής.

4.1.6 Απόφαση Εκτέλεσης Συμπερασματολογίας στον Εξυπηρετητή

Η συνάρτηση απόφασης εκτέλεσης συμπερασματολογίας στον εξυπηρετητή (send_to_server) αποφασίζει αν επιτρέπεται να εκτελεστεί συμπερασματολογία στον εξυπηρετητή. Επιστρέφει μια δυαδική τιμή, αν αυτή είναι αληθής τότε επιτρέπεται να εκτελεστεί συμπερασματολογία στον εξυπηρετητή διαφορετικά αν είναι ψευδής δεν επιτρέπεται να εκτελεστεί συμπερασματολογία στον εξυπηρετητή.

Τα έξι στοιχεία εισόδου είναι:

- Το **νευρωνικό μοντέλο που χρησιμοποιεί ο εξυπηρετητής (server_model)**.
- Τα **χαρακτηριστικά του εξυπηρετητή (server_specs)**, όπως στην 4.1.3.
- Τα **στοιχεία της σύνδεσης (connection)**, όπως στην 4.1.2.
- Το **μέγεθος του αρχείου εικόνας (image_size)**.
- Το **όριο της καθυστέρησης (thresh)**, που επιτρέπεται να προστεθεί αν η συμπερασματολογία εκτελεστεί στον εξυπηρετητή.
- Η **λειτουργία (inference_time_mode)** του τρόπου υπολογισμού του μέσου χρόνου που χρειάζεται για να εκτελεστεί η συμπερασματολογία μιας εικόνας στον εξυπηρετητή, με τιμές 'exact' ή 'scaling'.

Η συνάρτηση υπολογίζει τον χρόνο που χρειάζεται για να μεταφερθούν τα δεδομένα κάνοντας χρήση της συνάρτησης 4.1.2 και τον μέσο χρόνο που χρειάζεται για να εκτελεστεί η συμπερασματολογία από τις συναρτήσεις 4.1.3 ή 4.1.4. Η συνάρτηση ελέγχει την παρακάτω ανίσωση:

$$L_s \cdot t_{inf,s} + t_{trans} < t_{thresh,s}$$

όπου $t_{inf,s}$ είναι ο χρόνος εκτέλεσης συμπερασματολογίας στον εξυπηρετητή, t_{trans} ο χρόνος μεταφοράς των δεδομένων και $t_{thresh,s}$ το όριο του χρόνου συμπερασματολογίας. Αν ο συνολικός χρόνος καθυστέρησης είναι μικρότερος από το χρονικό όριο που δόθηκε, τότε η συμπερασματολογία επιτρέπεται να εκτελεστεί στον εξυπηρετητή, δηλαδή η τιμή εξόδου είναι αληθής.

4.1.7 Συμπερασματολογία στην Κινητή Συσκευή

Η συνάρτηση συμπερασματολογίας στην κινητή συσκευή (inference_on_mobile) εκτελεί τη συμπερασματολογία με το μοντέλο της κινητής συσκευής και επιστρέφει τα αναλυτικά αποτελέσματα (πεποιθήσεις για κάθε κατηγορία). Τα δύο στοιχεία εισόδου είναι:

- Το **νευρωνικό μοντέλο της κινητής συσκευής (mobile_model)**.
- Η **εικόνα στην οποία θα εκτελεστεί συμπερασματολογία (input_image)**, η οποία είναι προεπεξεργασμένη από τη βοηθητική συνάρτηση «Προετοιμασία εικόνας για το μοντέλο της συσκευής» (βλ. 4.1.10), όπως χρειάζεται το μοντέλο ώστε να εκτελέσει συμπερασματολογία.

4.1.8 Συμπερασματολογία στον Εξυπηρετητή

Αντίστοιχα με την προηγούμενη συνάρτηση, η συνάρτηση συμπερασματολογίας στον εξυπηρετητή (inference_on_server) εκτελεί τη συμπερασματολογία με το μοντέλο του εξυπηρετητή και επιστρέφει τα αναλυτικά αποτελέσματα (πεποιθήσεις για κάθε κατηγορία). Τα δυο στοιχεία εισόδου είναι:

- Το **νευρωνικό μοντέλο του εξυπηρετητή (server_model)**.
- Η **εικόνα στην οποία θα εκτελεστεί συμπερασματολογία (input_image)**, η οποία είναι προεπεξεργασμένη από τη βοηθητική συνάρτηση «Προετοιμασία εικόνας για το μοντέλο του εξυπηρετητή» (βλ. 4.1.10), όπως χρειάζεται το μοντέλο ώστε να εκτελέσει συμπερασματολογία.

4.1.9 Συμπερασματολογία

Η συνάρτηση συμπερασματολογίας (inference), είναι η βασική συνάρτηση του συστήματος. Δέχεται δεκα στοιχεία εισόδου και επιστρέφει τρία στοιχεία εξόδου: τα αποτελέσματα της συμπερασματολογίας στο κινητό (αν δεν εκτελέστηκε συμπερασματολογία στο κινητό επιστρέφει τιμή 0), τα αποτελέσματα της συμπερασματολογίας στον εξυπηρετητή (αν δεν εκτελέστηκε συμπερασματολογία στον εξυπηρετητή επιστρέφει τιμή 0) και μια δυαδική τιμή η οποία δηλωνει αν εκτελέστηκε συμπερασματολογία στον εξυπηρετητή.

Τα δέκα στοιχεία εισόδου είναι:

- Το **νευρωνικό μοντέλο της κινητής συσκευής (mobile_model)**.
- Το **νευρωνικό μοντέλο του εξυπηρετητή (server_model)**.
- Τα **χαρακτηριστικά του εξυπηρετητή (server_specs)**, όπως στην 4.1.3.
- Τα **χαρακτηριστικά της κινητής συσκευής (mobile_specs)**, όπως στην 4.1.5.
- Τα **στοιχεία της σύνδεσης (connection)**, όπως στην 4.1.2.
- Η **τοποθεσία της εικόνας (img_path)**.
- Ο **τύπος λειτουργίας (framework_type)**, με τιμές 'cascade' ή 'standard'.
- Η **λειτουργία (inference_time_mode)** του τρόπου υπολογισμού του μέσου χρόνου που χρειάζεται για να εκτελεστεί η συμπερασματολογία μιας εικόνας στον εξυπηρετητή, με τιμές 'exact' ή 'scaling'.
- Η **προτίμηση εκτέλεσης (inf_both)** σε περίπτωση που μπορεί να γίνει συμπερασματολογία και στην κινητή συσκευή και στον εξυπηρετητή. Αφορά μόνο τον κανονικό ('standard') τύπο λειτουργίας και μπορεί να έχει τιμές 'server' ή 'mobile'.
- Η **προτίμηση εκτέλεσης (inf_none)** σε περίπτωση που δεν μπορεί να γίνει συμπερασματολογία ούτε στην κινητή συσκευή ούτε στον εξυπηρετητή. Αφορά μόνο τον κανονικό ('standard') τύπο λειτουργίας και μπορεί να έχει τιμές 'server' ή 'mobile'.

Το σύνολο των στοιχείων εισόδου της inference συμβολίζεται ως:

$$I = \{M_m, M_s, S_s, S_m, C, I_p, F, t_{mode}, D_b, D_n\}$$

Η συνάρτηση αρχικά προετοιμάζει την εικόνα για συμπερασματολογία και στα δύο μοντέλα και αποθηκεύει το μέγεθος της εικόνας. Αν ο τύπος λειτουργίας είναι 'cascade', τότε εκτελείται συμπερασματολογία στο μοντέλο της κινητής συσκευής και στη συνέχεια ανάλογα με το αποτέλεσμα αν πρέπει και παράλληλα επιτρέπεται η εικόνα στέλνεται και στον εξυπηρετητή για συμπερασματολογία. Αν ο τύπος λειτουργίας είναι 'standard', υπολογίζεται αν επιτρέπεται να εκτελεστεί συμπερασματολογία στην κινητή συσκευή και στον εξυπηρετητή. Διακρίνονται 4 περίπτωσεις:

1. Αν επιτρέπεται εκτέλεση συμπερασματολογίας στη συσκευή και όχι στον εξυπηρετητή τότε εκτελείται συμπερασματολογία στο μοντέλο της συσκευής.
2. Αν επιτρέπεται εκτέλεση συμπερασματολογίας στον εξυπηρετητή και όχι στη συσκευή, τότε εκτελείται συμπερασματολογία στο μοντέλο του εξυπηρετητή.
3. Αν επιτρέπεται να εκτελεστεί συμπερασματολογία και στα δύο τότε εκτελείται στην επιλογή που υποδεικνύει το στοιχείο εισόδου inf_both.
4. Αν δεν επιτρέπεται να εκτελεστεί συμπερασματολογία σε κανένα, τότε πάλι εκτελείται στην επιλογή που υποδεικνύει το στοιχείο εισόδου inf_none.

4.1.10 Βοηθητικές Συναρτήσεις

Υπάρχουν τρεις βοηθητικές συναρτήσεις οι οποίες είναι απαραίτητες για τη λειτουργία του συστήματος αλλά δεν περιλαμβάνονται στις βασικές.

- **Μέγεθος εικόνας**. Αυτή η συνάρτηση δέχεται σαν είσοδο την τοποθεσία της εικόνας και επιστρέφει το μέγεθός της σε bits κάνοντας χρήση της βιβλιοθήκης os.
- **Προετοιμασία εικόνας για το μοντέλο της συσκευής**. Αυτή η συνάρτηση δέχεται σαν είσοδο την τοποθεσία της εικόνας, την προετοιμάζει (αναδιαμόρφωση στις διαστάσεις που ζητά το μοντέλο, μετατροπή σε διάνυσμα, κ.λπ.) και την επιστρέφει σε μορφή κατάλληλη για επεξεργασία από το μοντέλο.
- **Προετοιμασία εικόνας για το μοντέλο του εξυπηρετητή**. Όμοια με τη συνάρτηση προετοιμασίας εικόνας για το μοντέλο της συσκευής, απλώς με την προετοιμασία που χρειάζεται για το μοντέλο του εξυπηρετητή, για παράδειγμα διαφορετικές διαστάσεις κ.α.

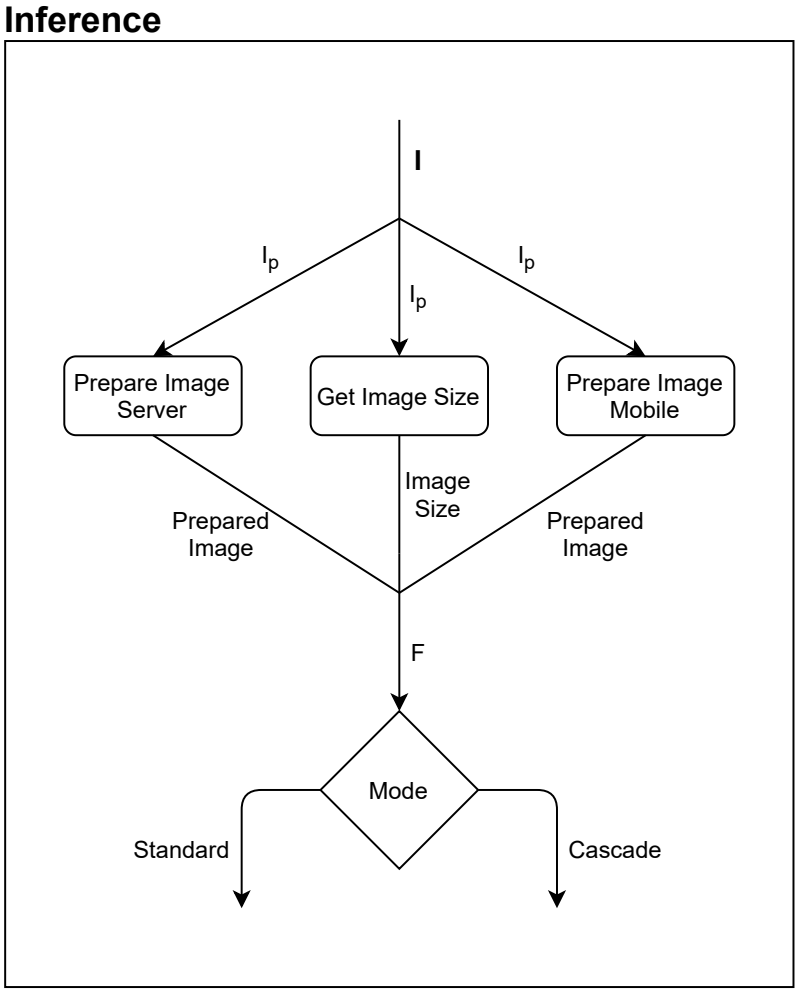
4.2 Εκτέλεση Προγράμματος

Σε αυτή την ενότητα, περιγράφεται η συνεργασία μεταξύ των συναρτήσεων και το πώς εκτελείται το πρόγραμμα από την είσοδο μέχρι και την έξοδο. Το πρόγραμμα ξεκινάει καλώντας τη συνάρτηση inference (βλ. 4.1.9) με όλα τα απαραίτητα στοιχεία εισόδου. Σε αυτά περιλαμβάνονται το νευρωνικό μοντέλο του εξυπηρετητή, το νευρωνικό μοντέλο της κινητής συσκευής, τα χαρακτηριστικά του εξυπηρετητή, τα χαρακτηριστικά της κινητής συσκευής, τα χαρακτηριστικά της σύνδεσης, η τοποθεσία της εικόνας, ο τύπος εκτέλεσης, ο τρόπος υπολογισμού του μέσου χρόνου εκτέλεσης συμπερασματολογίας για τον εξυπηρετητή και οι προτιμήσεις σε περίπτωση που μπορεί να εκτελεστεί συμπερασματολογία και στον εξυπηρετητή και στη συσκευή ή σε κανένα από τα δύο σημεία εκτέλεσης.

Η συνάρτηση συμπερασματολογίας προετοιμάζει τις εικόνες για το μοντέλο του εξυπηρετητή και το μοντέλο της κινητής συσκευής καλώντας τις βοηθητικές συναρτήσεις προετοιμασίας της εικόνας (prepare_image_mobile, prepare_image_server) (βλ. 4.1.10). Επίσης υπολογίζει το μέγεθος της εικόνας καλώντας τη συνάρτηση get_img_size (βλ. 4.1.10).

Έπειτα, η ροή της εκτέλεσης διαχωρίζεται και ακολουθεί διαφορετικό μονοπάτι ανάλογα με τον τύπο λειτουργίας που δόθηκε. Καθώς το σύστημα είναι ευέλικτο, στο μέλλον μπορούν να προστεθούν καινούριοι τύποι λειτουργίας και να έχουμε περισσότερα από δύο μονοπάτια ροής εκτέλεσης. Η αρχικοποίηση και ο διαχωρισμός φαίνονται στην Εικόνα 4.1.

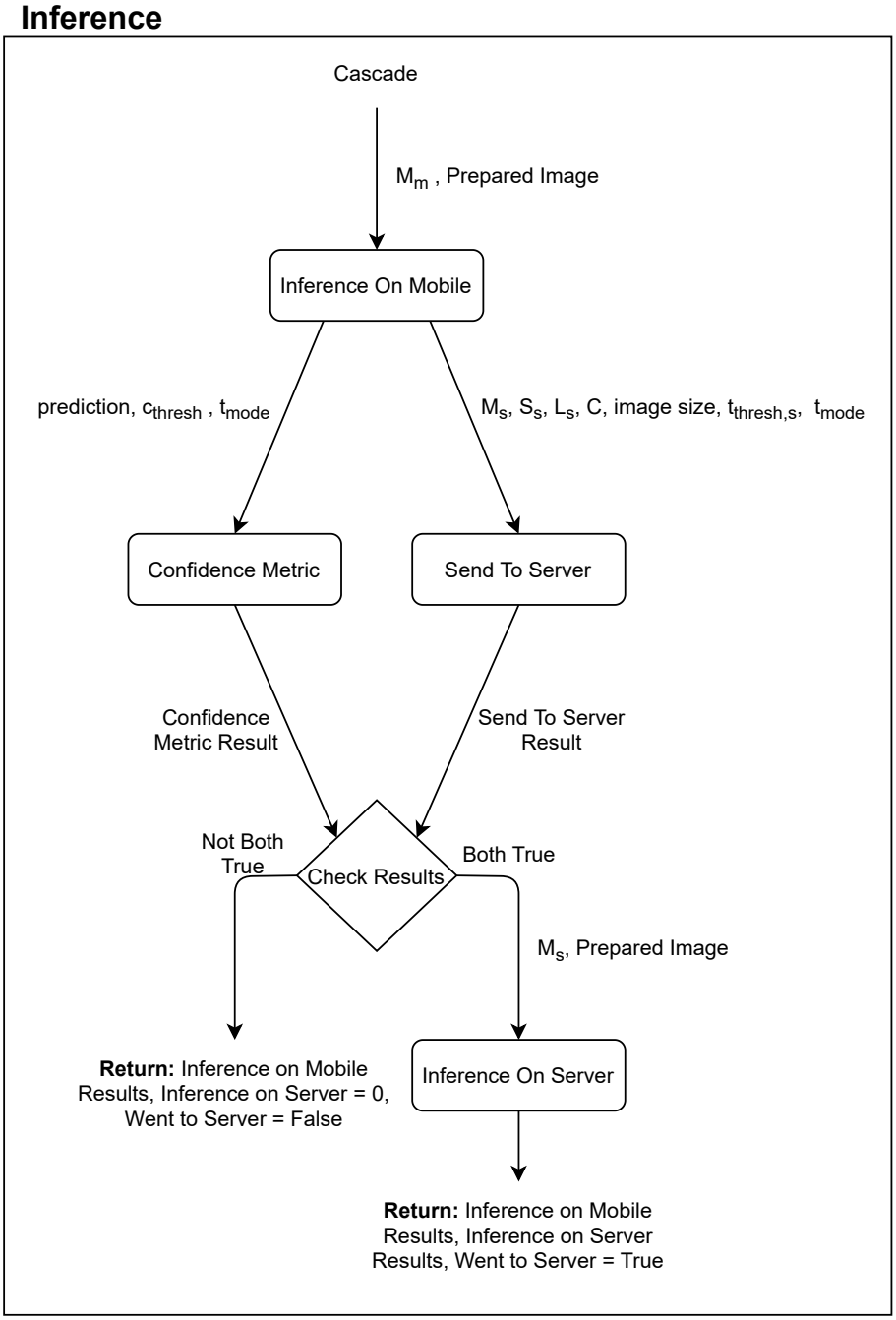
Αν ο τύπος λειτουργίας είναι Cascade, τότε πρώτα εκτελείται συμπερασματολογία στο μοντέλο της κινητής συσκευής καλώντας την inference_on_mobile. Ύστερα, υπολογίζεται αν υπάρχει εμπιστοσύνη στα αποτελέσματα καλώντας την conf_metric (βλ. 4.1.1). Αν αυτή επιστρέψει ψευδές (false) αποτέλεσμα και επιτρέπεται να γίνει συμπερασματολογία στον εξυπηρετητή, το οποίο βρίσκει καλώντας την send_to_server (βλ. 4.1.6), τότε εκτελείται συμπερασματολογία και στον εξυπηρετητή καλώντας τη συνάρτηση inference_on_server (βλ.



Εικόνα 4.1: *Διάγραμμα Ροής Εκτέλεσης 1*

4.1.8). Επιστρέφει τα αποτελέσματα των συμπερασματολογιών (σε περίπτωση που δεν εκτελέστηκε συμπερασματολογία στον εξυπηρετητή τα αποτελέσματά του είναι 0), καθώς και αν έγινε συμπερασματολογία στον εξυπηρετητή. Η εκτέλεση του Cascade μοντέλου φαίνεται στην Εικόνα 4.2.

Αν ο τύπος λειτουργίας είναι Standard, τότε υπολογίζεται αν μπορεί να εκτελεστεί συμπερασματολογία στην κινητή συσκευή και στον εξυπηρετητή καλώντας τις συναρτήσεις stay_on_mobile (βλ. 4.1.5) και send_to_server (βλ. 4.1.6), αντίστοιχα. Σε περίπτωση που μπορεί να εκτελεστεί συμπερασματολογία στην κινητή συσκευή και όχι στον εξυπηρετητή, υπολογίζονται τα αποτελέσματα συμπερασματολογίας του μοντέλου της συσκευής καλώντας τη συνάρτηση inference_on_mobile (βλ. 4.1.7). Σε περίπτωση που δεν μπορεί να εκτελεστεί συμπερασματολογία στη συσκευή αλλά μπορεί να εκτελεστεί στον εξυπηρετητή, υπολογίζονται τα αποτελέσματα του μοντέλου του εξυπηρετητή καλώντας την inference_on_server (βλ. 4.1.8). Σε περίπτωση που μπορεί να εκτελεστεί και στα δύο, παίρνουμε τα αποτελέσματα του μοντέλου που έχει επιλέγει στα στοιχεία εισόδου της inference (βλ. 4.1.9), ομοίως αν δεν μπορεί να γίνει σε κανένα από τα δύο. Επιστρέφονται τα αποτελέσματα, τα οποία είναι 0 για το μοντέλο στο οποίο δεν εκτελέστηκε συμπερασματολογία και αν έκτελέστηκε συμπερα-

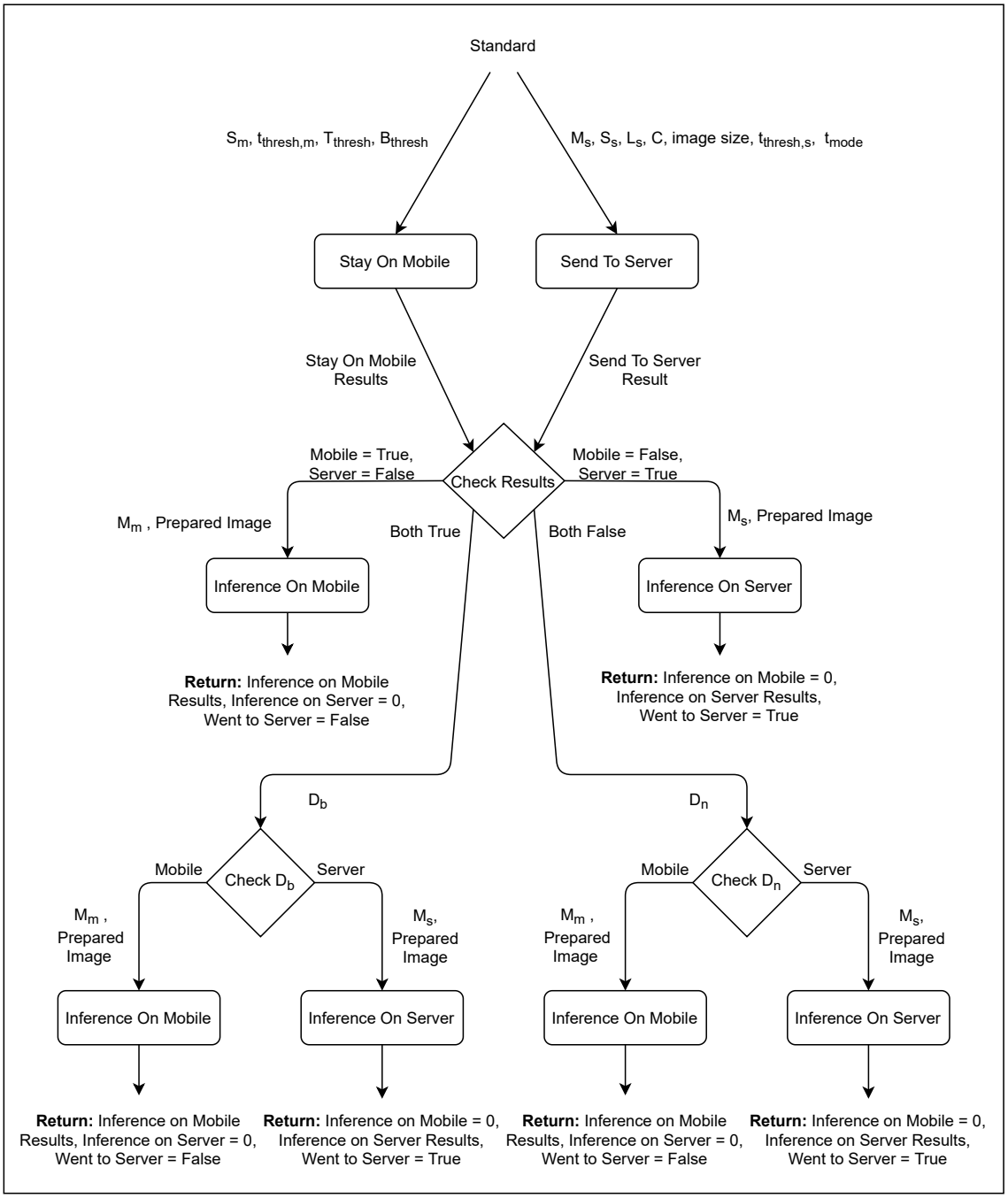


Εικόνα 4.2: *Διάγραμμα Ροής Εκτέλεσης 2*

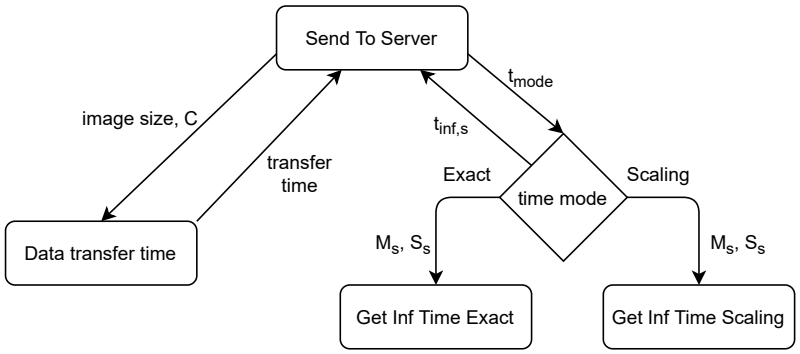
σματολογία στον εξυπηρετητή. Η εκτέλεση του Standard μοντέλου φαίνεται στην Εικόνα 4.3.

Κατά την εκτέλεση της συνάρτησης send_to_server (βλ. 4.1.6) γίνεται κλήση των συναρτήσεων get_server_inf_time_exact (βλ. 4.1.3) ή get_server_inf_time_scaling (βλ. 4.1.4) και της data_trans_time (βλ. 4.1.2). Η επιλογή κλήσης ανάμεσα στις get_server_inf_time_exact και get_server_inf_time_exact γίνεται με βάση τη μεταβλητή t_{mode}. Η εκτέλεση της send_to_server φαίνεται στην Εικόνα 4.4

Inference



Εικόνα 4.3: *Διάγραμμα Ροής Εκτέλεσης 3*



Εικόνα 4.4: *Διάγραμμα Ροής Εκτέλεσης 4*

Κεφάλαιο 5

Αξιολόγηση

Στο κεφάλαιο αυτό παρουσιάζονται οι μετρήσεις που έγιναν για μια πρώτη αξιολόγηση και κατανόηση της λειτουργίας του συστήματος σε διάφορους τομείς.

5.1 Μεθοδολογία Ελέγχου

Από τους δύο τρόπους λειτουργίας του συστήματος (cascade και standard) ιδιαίτερο ενδιαφέρον παρουσιάζει ο cascade. Καθώς εκτελείται πρώτα συμπερασματολογία στο μοντέλο της κινητής συσκευής, η πρόσβαση στις πεποιθήσεις της κατηγοριοποίησης καθιστά πιο δυναμικό το πρόβλημα της απόφασης του αν θα εκτελεστεί συμπερασματολογία και στο μοντέλο του εξυπηρετητή.

Οι δύο βασικότερες παράμετροι που επηρεάζουν τη λειτουργία του συστήματος, πέρα από τα νευρωνικά μοντέλα και τα χαρακτηριστικά της συσκευής και του εξυπηρετητή, είναι (α) το όριο με βάση το οποίο λαμβάνεται η απόφαση εμπιστοσύνης του αποτελέσματος του μοντέλου της κινητής συσκευής και (β) ο μέγιστος χρόνος καθυστέρησης που επιτρέπεται να έχει η εκτέλεση συμπερασματολογίας στον εξυπηρετητή.

Αυτές οι δύο μεταβλητές μελετήθηκαν ξεχωριστά χρησιμοποιώντας σαν σύνολο δεδομένων για τις μετρήσεις το σύνολο επικύρωσης (validation set) της πρόκλησης μεγάλης κλίμακας εικονικής αναγνώρισης ImageNet του 2012 (ILSVRC2012) [44], το οποίο περιέχει 50000 εικόνες με ετικέτες.

Οι μετρικές που χρησιμοποιήθηκαν για τη μελέτη του ορίου πεποίθησης ήταν η ακρίβεια για την 1η πρόβλεψη και για τις πρώτες 5 προβλέψεις. Συγκεκριμένα:

- **Ακρίβεια για την 1η πρόβλεψη (top-1 accuracy)**: το ποσοστό των εικόνων του συνόλου, για τις οποίες η κατηγορία στην οποία έδωσε τη μεγαλύτερη πεποίθηση η συμπερασματολογία είναι η σωστή.
- **Ακρίβεια για τις 5 πρώτες προβλέψεις (top-5 accuracy)**: το ποσοστό των εικόνων του συνόλου, για τις οποίες κάποια από τις 5 κατηγορίες με τις μεγαλύτερες πεποιθήσεις είναι σωστή.

Επίσης, αποθηκεύονταν ο αριθμός των εκτελέσεων συμπερασματολογίας στον εξυπηρετητή.

Η καθυστέρηση που προστίθεται από την εκτέλεση συμπερασματολογίας στον εξυπηρετητή εξαρτάται από το μοντέλο, τα χαρακτηριστικά του εξυπηρετητή και από το είδος της σύνδεσης. Μοντελοποιήθηκαν 7 διαφορετικοί τύποι συνδέσεων, οι οποίοι φαίνονται στον Πίνακα 5.2 και για αυτούς υπολογίστηκαν οι χρόνοι μεταφοράς κάθε εικόνας του validation set. Σε αυτό το σύνολο των χρόνων υπολογίστηκαν οι μετρικές:

- **Μέσος**: Ο μέσος όρος όλων των χρόνων μεταφοράς.
- **Τυπική απόκλισή**: Η τυπική απόκλιση των χρόνων μεταφοράς.
- **Ελάχιστο**: Ο ελάχιστος χρόνος μεταφοράς.
- **Μέγιστο**: Ο μέγιστος χρόνος μεταφοράς.
- **90th percentile**: Ο χρόνος κάτω από τον οποίο βρίσκεται το 90% των υπόλοιπων χρόνων.

Τέλος, για διάφορες τιμές του ορίου πεποίθησης μελετήθηκαν τρεις τύποι μη αποδοτικών συμπερασματολογιών (inefficiencies). Αυτές είναι εκτελέσεις συμπερασματολογίας όπου θα μπορούσε να έχει γίνει εξοικονόμηση χρόνου και πόρων ή ακόμα και αύξηση της ακρίβειας αν δεν είχε επιλεγεί εκτέλεση συμπερασματολογίας και στον εξυπηρετητή.

5.2 Στοιχεία Εισόδου

Για όλες τις μετρήσεις χρησιμοποιήθηκε σαν μοντέλο κινητής συσκευής το MobileNet [45] ένα μικρό, χαμηλής καθυστέρησης, χαμηλής ενέργειας μοντέλο το οποίο χρειάζεται λίγους πόρους για να λειτουργήσει (βλ. 2.3.2) και σαν μοντέλο εξυπηρετητή το NASNetLarge [46] ένα βαρύ μοντέλο με υψηλή ακρίβεια (βλ. 2.3.2). Τα δύο μοντέλα παρέχονται από τη βιβλιοθήκη Keras. Στον Πίνακα 5.1 έχουν συγκεντρωθεί τα χαρακτηριστικά των δύο μοντέλων που χρησιμοποιήθηκαν με τις ακρίβειες να έχουν υπολογιστεί στο σύνολο επικύρωσης του ImageNet.

Πίνακας 5.1: *Χαρακτηριστικά και ακρίβεια μοντέλων*

Models	Top-1	Top-5	Number of Parameters	Size	Input Size	FLOPs
MobileNet	0.684	0.883	4,253,864	16 MB	224×224	0.569B
NASNetLarge	0.816	0.956	88,949,818	343 MB	331×331	24B

Για τα χαρακτηριστικά της σύνδεσης μοντελοποιήθηκαν 7 διαφορετικοί τύποι συνδέσεων, όπως φαίνονται στον Πίνακα 5.2.

Για τις μετρήσεις του ορίου πεποίθησης, σαν σύνδεση χρησιμοποιήθηκε το WiFi 5GHz ώστε να μην επηρεάζεται το αν επιτρέπεται να εκτελεστεί συμπερασματολογία στον εξυπηρετητή από τη σύνδεση. Για τα χαρακτηριστικά του εξυπηρετητή επιλέχτηκαν 12.15 TeraFLOP/s, 16 GB RAM και πολλαπλασιαστής φόρτου εργασίας 1. Αυτές οι τιμές προέρχονται από σύστημα με μονάδα επεξεργασίας γραφικών την NVIDIA TITAN Xp, το οποίο χρειάζεται κατά μέσο όρο 0.033 δευτερόλεπτα για να εκτελέσει συμπερασματολογία σε μία

Πίνακας 5.2: *Χαρακτηριστικά συνδέσεων*

Connection	Latency	Bandwidth
3G	0.25	400000
3G HSPA+	0.25	3000000
4G	0.042	10000000
4G advanced	0.042	25000000
5G	0.015	100000000
WiFi 2.4 GHz	0.034	150000000
WiFi 5 GHz	0.008	500000000

εικόνα στο μοντέλο NASNetLarge (βλ. Πίνακα μοντέλων εξυπηρετητή στο Παράρτημα Β'). Γενικά τα χαρακτηριστικά της σύνδεσης και του εξυπηρετητή δεν επηρέασαν την απόφαση για εκτέλεση συμπερασματολογίας στον εξυπηρετητή, οπότε η απόφαση λήφθηκε με βάση το όριο πεποίθησης.

5.3 Όριο Πεποίθησης

Για τιμές του ορίου πεποίθησης από το 0 έως το 1, υπολογίστηκαν οι ακρίβειες για την 1η πρόβλεψη, τις πρώτες 5 προβλέψεις, καθώς και ο αριθμός εκτελέσεων συμπερασματολογίας στον εξυπηρετητή. Τα αποτελέσματα φαίνονται στον Πίνακα 5.3.

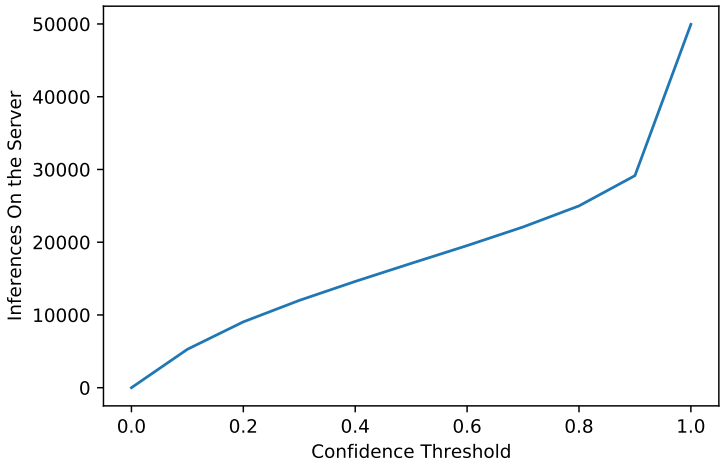
Πίνακας 5.3: *Μετρήσεις ορίου πεποίθησης*

Confidence Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Top 1 Accuracy	0.684	0.725	0.752	0.769	0.781	0.790	0.799	0.805	0.810	0.814	0.816
Top 5 Accuracy	0.883	0.906	0.920	0.930	0.934	0.939	0.944	0.948	0.951	0.953	0.956
Inferences On Server	0	5279	9050	11999	14607	17089	19534	22093	24991	29151	49948

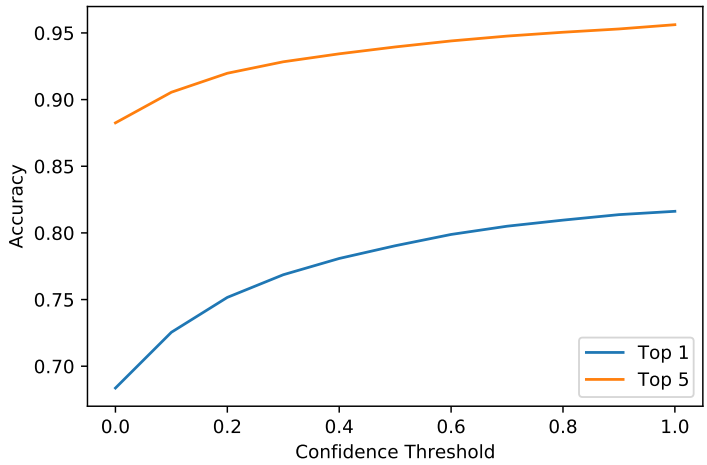
Με την αύξηση του ορίου πεποίθησης, έχουμε εκθετική αύξηση του αριθμού των συμπερασματολογιών στον εξυπηρετητή αλλά λογαριθμική αύξηση της ακρίβειας όπως φαίνεται στις Εικόνες 5.1 και 5.2.

Αυτό είναι λογικό καθώς στις περιπτώσεις των συμπερασματολογιών που εκτελούνται στον εξυπηρετητή όταν το όριο πεποίθησης είναι χαμηλό, είναι μεγάλη η πιθανότητα το μοντέλο της συσκευής να έχει κάνει λάθος κατηγοριοποίηση και το μοντέλο του εξυπηρετητή που έχει γενικά καλύτερη ακρίβεια να κάνει σωστή κατηγοριοποίηση. Όσο μεγαλώνει το όριο πεποίθησης, στέλνονται στον εξυπηρετητή και αποτελέσματα για τα οποία το μοντέλο της συσκευής είναι αρκετά σίγουρο, δηλαδή υπάρχει μεγάλη πιθανότητα να τα έχει κατηγοριοποιήσει σωστά, με αποτέλεσμα να μην έχουμε ιδιαίτερη αύξηση στην ακρίβεια.

Συστήματα τα οποία απαιτούν υψηλή ακρίβεια ή δεν προσθέτουν μεγάλη καθυστέρηση με αποστολή των εικόνων στον εξυπηρετητή επωφελούνται από υψηλό όριο πεποίθησης. Συστήματα τα οποία έχουν μέτριες απαιτήσεις ακρίβειας ή που προσθέτουν μια μέτρια καθυστέρηση με την αποστολή εικόνων στον εξυπηρετητή επωφελούνται από ένα μεσαίο όριο πεποίθησης. Συστήματα με χαμηλές απαιτήσεις ακρίβειας ή που προσθέτουν μεγάλη καθυστέρηση με αποστολή εικόνων στον εξυπηρετητή επωφελούνται από ένα μικρό όριο πεπο-



Εικόνα 5.1: *Πλήθος εκτελέσεων συμπερασματολογίας στον εξυπηρετητή*



Εικόνα 5.2: *Ακρίβεια ζεύγους συναρτήσει του ορίου πεποίθησης*

ίθησης. Ακόμα και για μικρές τιμές του ορίου πεποίθησης παρατηρείται αισθητή αύξηση στην ακρίβεια ενώ τιμές πάνω από το 0.7 δεν τη μεταβάλουν ιδιαίτερα και καλό θα ήταν να αποφεύγονται.

5.4 Χρόνοι Μεταφοράς

Για να εξεταστεί η αποτελεσματικότητα των μοντέρνων δικτύων δεδομένων ως προς το χρόνο μεταφοράς, μελετήθηκαν τα στατιστικά των χρόνων μεταφοράς για τους διάφορους τύπους συνδέσεων των οποίων οι μοντελοποιήσεις φαίνονται στον Πίνακα 5.2.

Το βασικό στατιστικό που μας ενδιαφέρει είναι το 90th percentile, το οποίο δείχνει το χρόνο κάτω από τον οποίο βρίσκεται το 90% όλων των χρόνων. Αυτό σημαίνει ότι το 90% των εικόνων χρειάζεται λιγότερο από αυτό τον χρόνο για να σταλθεί. Με βάση αυτό βλέπουμε ποιες συνδέσεις έχουν ικανοποιητικό χρόνο μεταφοράς για το μεγαλύτερο μέρος των εικόνων.

Από τα αποτελέσματα στους Πίνακες 5.4, 5.5 και 5.6, βλέπουμε ότι οι συνδέσεις WiFi 5GHz, WiFi 2.4 GHz και 5G δεν προσθέτουν σχεδόν καθόλου καθυστέρηση με το 90% των εικόνων να μπορούν να μεταφερθούν σε χρόνο μικρότερο από 50 msec. Είναι λογικό να υποθέσουμε ότι αυτές οι συνδέσεις λειτουργούν πολύ καλά με το σύστημα σε οποιαδήποτε περίπτωση.

Πίνακας 5.4: *Στατιστικά χρόνων μεταφοράς σύνδεσης Wifi 5GHz*

Connection type	Wifi 5GHz
Mean	10.15
Standard Deviation	1.94
Min	0.8
Max	129.2
90th percentile	11.2

Πίνακας 5.5: *Στατιστικά χρόνων μεταφοράς σύνδεσης Wifi 2.4GHz*

Connection type	Wifi 2.4GHz
Mean	41.15
Standard Deviation	6.47
Min	34.1
Max	438.2
90th percentile	44.6

Πίνακας 5.6: *Στατιστικά χρόνων μεταφοράς σύνδεσης 5G*

Connection type	5G
Mean	25.73
Standard Deviation	9.71
Min	15.1
Max	621.2
90th percentile	31

Στις συνδέσεις 4G advanced και 4G το 90% των εικόνων μπορούν να μεταφερθούν σε λιγότερο από 110 και 210 msec αντίστοιχα, τα αποτελέσματα φαίνονται στους Πίνακες 5.7, 5.8. Για τις περισσότερες περιπτώσεις αυτές οι συνδέσεις περιμένουμε να είναι επαρκείς.

Πίνακας 5.7: *Στατιστικά χρόνων μεταφοράς σύνδεσης 4G advanced*

Connection type	4G advanced
Mean	84.9
Standard Deviation	38.8
Min	42.3
Max	2467
90th percentile	105.9

Πίνακας 5.8: *Στατιστικά χρόνων μεταφοράς σύνδεσης 4G*

Connection type	4G
Mean	149.3
Standard Deviation	97.1
Min	42.8
Max	6104.5
90th percentile	201.7

Τέλος η σύνδεση 3G HSPA+ είναι αρκετά αργή, με το 90% των εικόνων να χρειάζονται από 800 msec και κάτω για να μεταφερθούν και θα μπορεί να χρησιμοποιηθεί σε ειδικές περιπτώσεις μόνο, η 3G είναι πολύ αργή και δύσκολα θα μπορούσε να χρησιμοποιηθεί. Τα αποτελέσματα φαίνονται στους Πίνακες 5.9 και 5.10 αντίστοιχα.

Πίνακας 5.9: *Στατιστικά χρόνων μεταφοράς σύνδεσης 3G hspa+*

Connection type	3G hspa+
Mean	607.7
Standard Deviation	323.6
Min	252.6
Max	20458.2
90th percentile	782.2

Πίνακας 5.10: *Στατιστικά χρόνων μεταφοράς σύνδεσης 3G*

Connection type	3G
Mean	2932.6
Standard Deviation	2427.2
Min	269.7
Max	151811.7
90th percentile	4241.7

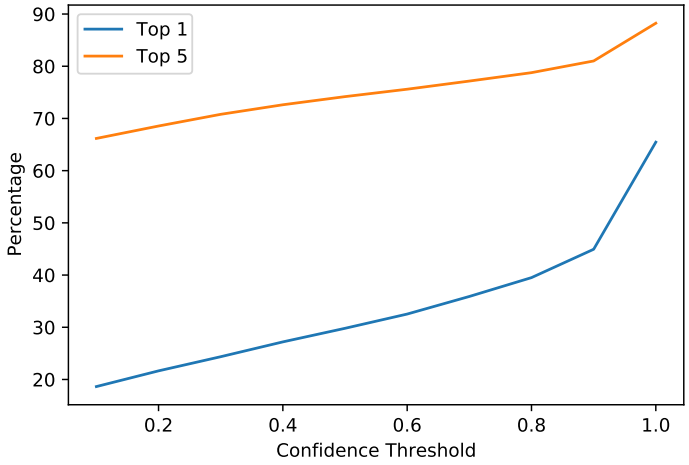
5.5 Μετρήσεις Μη Αποτελεσματικών Συμπερασματολογιών

Οι μετρήσεις μη αποτελεσματικών συμπερασματολογιών για το σύστημα στη λειτουργία cascade βοηθούν πολύ στην κατανόηση του τρόπου λειτουργίας του ως προς τη μεταβλητή του ορίου πεποίθησης. Οι μετρήσεις μη αποτελεσματικών συμπερασματολογιών χωρίζονται σε τρεις τύπους και μπορούν να υπολογιστούν μόνο στην περίπτωση που εκτελεστεί συμπερασματολογία και στον εξυπηρετητή.

Για κάθε τιμή του ορίου πεποίθησης, υπολογίστηκε ο αριθμός των εκτελέσεων συμπερασματολογίας που ανήκουν σε κάθε τύπο και το ποσοστό που αντιπροσωπεύουν από το σύνολο των συμπερασματολογιών που εκτελέστηκαν στον εξυπηρετητή. Στις Εικόνες 5.3, 5.4 και 5.5 βλέπουμε πώς επηρεάζονται τα ποσοστά των μη αποδοτικών εκτελέσεων συμπερασματολογίας καθώς μεταβάλεται το όριο πεποίθησης.

5.5.1 Τύπος 1

Σωστή πρόβλεψη και στα δύο μοντέλα. Τόσο το μοντέλο της συσκευής, όσο και το μοντέλο του εξυπηρετητή κάνουν σωστή κατηγοριοποίηση της εικόνας. Καθώς η αρχική πρόβλεψη του μοντέλου της συσκευής είναι σωστή, η εκτέλεση συμπερασματολογίας στον εξυπηρετητή είναι περιττή.



Εικόνα 5.3: *Μη αποδοτικές εκτελέσεις συμπερασματολογίας Τύπου 1*

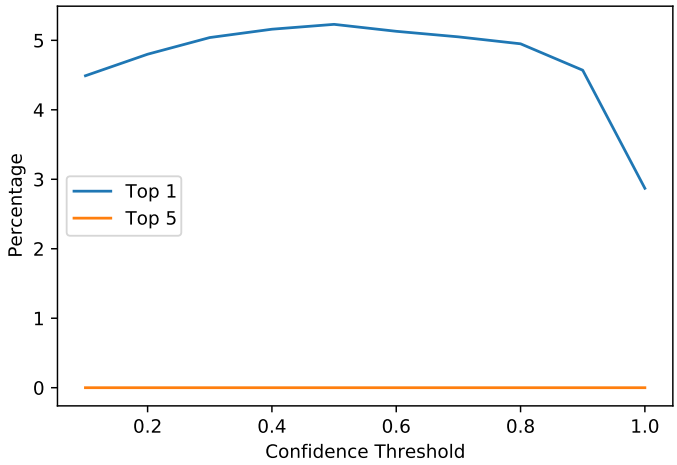
Οι μη αποδοτικές εκτελέσεις τύπου 1, αυξάνονται με την αύξηση του ορίου πεποίθησης. Αυτό συμβαίνει γιατί όσο αυξάνεται το όριο πεποίθησης, στέλνονται για συμπερασματολογία στον εξυπηρετητή και εικόνες τις οποίες υπάρχει μεγάλη πιθανότητα να έχει κατηγοριοποιήσει σωστά το μοντέλο της συσκευής. Συνεπώς ο αριθμός των εικόνων που κατηγοριοποιούν σωστά και το μοντέλο της συσκευής και το μοντέλο του εξυπηρετητή αυξάνεται.

5.5.2 Τύπος 2

Σωστή πρόβλεψη μοντέλου συσκευής, λανθασμένη πρόβλεψη μοντέλου εξυπηρετητή. Εφόσον το μοντέλο του εξυπηρετητή έχει μεγαλύτερη ακρίβεια από το μοντέλο της κινητής συσκευής, όποτε εκτελείται συμπερασματολογία και στον εξυπηρετητή, επιλέγεται το δικό του αποτέλεσμα και το αποτέλεσμα του μοντέλου της συσκευής αγνοείται. Συνεπώς το αποτέλεσμα που θα χρησιμοποιηθεί, αν έχουμε μη αποτελεσματική συμπερασματολογία τύπου 2, είναι λανθασμένο ενώ αρχικά ήταν σωστό. Αυτός ο τύπος μη αποδοτικής συμπερασματολογίας είναι και ο πιο επιβλαβής καθώς δεν προσθέτει καθυστέρηση (όπως οι άλλοι τύποι) αλλά μειώνει και την ακρίβεια.

Οι μη αποδοτικές εκτελέσεις τύπου 2 ακολουθούν μια παραβολική καμπύλη στην ακρίβεια της μίας πρόβλεψης με τα ποσοστά να κυμαίνονται ανάμεσα στο 3% και 5% (βλ. Γράφημα 5.4). Τα νούμερα είναι αρκετά μικρά αλλά θα θέλαμε να πλησιάζουν το 0. Το ποσοστό στην ακρίβεια των 5 πρώτων προβλέψεων είναι πάντα 0.

Για την καλύτερη κατανόηση των αποτελέσματων αυτού του τύπου έγινε ανάλυση στις κατηγορίες που πρόεβλεψε το μοντέλο του εξυπηρετητή και στις πραγματικές ετικέτες (labels) των εικόνων. Για τις μη αποδοτικές συμπερασματολογίες τύπου 2, υπολογίστηκε το πλήθος



Εικόνα 5.4: *Μη αποδοτικές εκτελέσεις συμπερασματολογίας Τύπου 2*

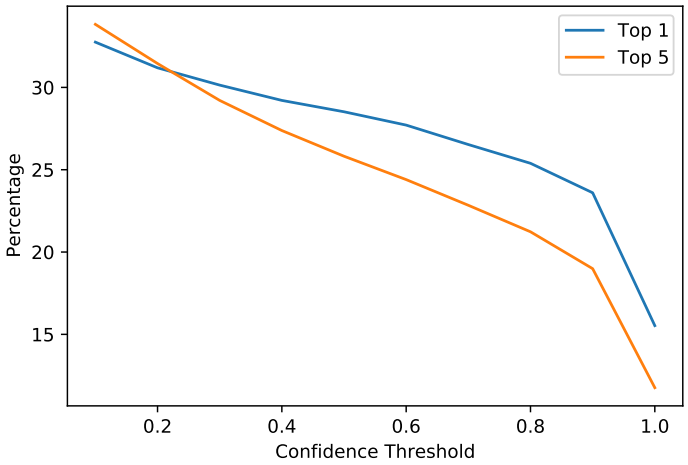
των μοναδικών ετικετών. Στο σύνολο των εικόνων που ανήκουν σε κατηγοριοποίηση τύπου 2, περιλαμβάνονται οι 635 από τις 1000 κατηγορίες. Ομοίως, υπολογίστηκε το πλήθος των μοναδικών κατηγοριοποιήσεων από το μοντέλου του εξυπηρετητή. Στο σύνολο των προβλέψεων του μοντέλου του εξυπηρετητή οι οποίες ανήκουν σε μη αποδοτική συμπερασματολογία τύπου 2, περιλαμβάνονται οι 627 από τις 1000 κατηγορίες. Και για το πλήθος των μοναδικών ετικετών και για το πλήθος των μοναδικών κατηγοριοποιήσεων, ο μέγιστος αριθμός επαναεμφανίσεων μίας κατηγορίας είναι 13. Συνεπώς, στον τύπο 2 και οι πραγματικές ετικέτες των εικόνων και οι προβλέψεις περιέχουν μεγάλο ποσοστό των κατηγοριών. Αυτό σε συνδυασμό με τον μικρό αριθμό επανεμφανίσεων των κατηγοριών μας ωθεί στο συμπέρασμα ότι οι μη αποδοτικές συμπερασματολογίες τύπου 2 φαίνεται να συμβαίνουν τυχαία και δεν εξαρτώνται από τις ετικές των εικόνων ή τις προβλέψεις του μοντέλου.

5.5.3 Τύπος 3

Λανθασμένη πρόβλεψη και στα δύο μοντέλα. Από τη στιγμή που είναι λανθασμένη και η πρόβλεψη του μοντέλου της συσκευής και η πρόβλεψη του μοντέλου του εξυπηρετητή, η εκτέλεση συμπερασματολογίας σε αυτόν σπαταλά χρόνο.

Οι μη αποδοτικές εκτελέσεις τύπου 3 μειώνονται όσο αυξάνεται το όριο πεποίθησης (βλ. Γράφημα 5.5). Αυτό συμβαίνει γιατί για μικρές τιμές του ορίου πεποίθησης υπάρχει μεγάλη πιθανότητα να έχει γίνει λάθος κατηγοριοποίηση από το μοντέλο της συσκευής. Όσο αυξάνεται το όριο πεποίθησης υπάρχει μεγαλύτερη πιθανότητα να έχει κάνει σωστή κατηγοριοποίηση με αποτέλεσμα να μειώνεται το ποσοστό των μη αποδοτικών εκτελέσεων τύπου 3 ως προς των αριθμό των εικόνων στις οποίες εκτελέστηκε συμπερασματολογία και στον εξυπηρετητή.

Η γωνία που παρατηρείται στα γραφήματα γύρω από την τιμή 0.9 του ορίου πεποίθησης, υπάρχει λόγω της μεγάλης αύξησης του αριθμού των εικόνων στις οποίες εκτελείται συμπερασματολογία και στον εξυπηρετητή, από 29151 στην τιμή 0.9 του ορίου πεποίθησης σε 49948 στην τιμή 1. Αυτές οι τιμές του ορίου πεποίθησης είναι πολύ υψηλές και δεν έχει



Εικόνα 5.5: *Μη αποδοτικές εκτελέσεις συμπερασματολογίας Τύπου 3*

νόημα η περαιτέρω μελέτη του συγκεκριμένου διαστήματος.

Κεφάλαιο 6

Επίλογος

Στο κεφάλαιο αυτό παρατίθενται γενικά συμπεράσματα της διπλωματικής, καθώς και η μελλοντική εργασία που μπορεί να γίνει με βάση αυτή.

6.1 Συμπεράσματα

Το σύστημα κατανεμημένου ζεύγους νευρωνικών δικτύων που δημιουργήθηκε σε αυτή τη διπλωματική είναι ευέλικτο, εύχρηστο και ταυτόχρονα κατάφερε να παραμετροποιηθεί σε ικανοποιητικό βαθμό. Το πλήθος των παραμέτρων και των συναρτήσεων καθιστούν ικανή τη μοντελοποίηση πληθώρας λειτουργιών.

Από τις μετρήσεις φάνηκε ότι η διάταξη του ζεύγους είναι αρκετά χρήσιμη και μπορεί να αυξήσει κατά πολύ την ακρίβεια του μοντέλου της συσκευής ακόμα και με λίγες μόνο εκτελέσεις συμπερασματολογίας στον εξυπηρετητή. Για παράδειγμα αν επιλέξουμε το όριο πεποίθησης έτσι ώστε το μοντέλο της συσκευής να μην είναι σίγουρο για το 24% του συνόλου των εικόνων, η ακρίβεια για την πρώτη πρόβλεψη αυξάνεται από 68.4% σε 76.9%, ενώ η ακρίβεια για τις 5 πρώτες προβλέψεις αυξάνεται από 88.3% σε 93% (βλ. Πίνακα 5.3).

Επίσης, από τις μετρήσεις των συνδέσεων δικτύων, φαίνεται ότι τα πιο σύγχρονα δίκτυα προσθέτουν αμελητέο χρόνο καθυστέρησης ενώ παλιότερα δίκτυα είναι πιθανό ανά περιπτώσεις να αυξάνουν τόσο το συνολικό χρόνο απόκρισης ώστε να καθιστούν την εκτέλεση συμπερασματολογίας στον εξυπηρετητή ασύμφορη. Ένα τέτοιο σύστημα συνεπώς, μπορεί να χρησιμοποιηθεί άμεσα σε πολλές εφαρμογές χωρίς να αποτελεί ζήτημα ο χρόνος μεταφοράς και στο μέλλον να αποτελέσει τον βασικό τρόπο συμπερασματολογίας στο νέφος ή στην άκρη του δικτύου.

Τέλος, με τις μετρήσεις των μη αποδοτικών εκτελέσεων συμπερασματολογίας φαίνεται ότι το σύστημα μπορεί να βελτιωθεί ως προς τον τρόπο επιλογής των εικόνων οι οποίες θα αποσταλούν στον εξυπηρετητή. Αυτό μπορεί να γίνει μέσα από περαιτέρω μελέτη αυτών των εκτελέσεων ως προς το περιεχόμενο των εικόνων για κάθε ξεχωριστό ζεύγος νευρωνικών μοντέλων. Ακόμα, η διαδικασία επιλογής των εικόνων οι οποίες θα αποσταλούν στον εξυπηρετητή μπορεί να αυτοματοποιηθεί εκπαιδεύοντας ένα νευρωνικό δίκτυο χρησιμοποιώντας σαν σύνολο εκπαίδευσης τις μετρήσεις μη αποδοτικών συμπερασματολογιών.

6.2 Μελλοντική Εργασία

Η παρούσα διπλωματική εργασία αποτελεί γερή βάση για μελλοντική εργασία προς πάρα πολλές κατευθύνσεις. Όπως αναφέρθηκε στα συμπεράσματα, η διάταξη μπορεί να βελτιώσει κατά πολύ την ακρίβεια των προβλέψεων με μικρή χρήση του εξυπηρετητή. Αξίζει λοιπόν το σύστημα να λαμβάνει υπόψιν κι άλλες μετρικές βελτιστοποίησης όπως η ενέργεια που καταναλώνεται ή το αποτύπωμα μνήμης, ώστε να μπορεί να ικανοποιεί τις ανάγκες πληθώρας εφαρμογών με διαφορετικούς στόχους επίδοσης. Οι κύριες κατευθύνσεις τις οποίες μπορεί να ακολουθήσει η παρούσα εργασία είναι:

- **Επέκταση του συστήματος**. Αν και το σύστημα έχει παραμετροποιηθεί σε ικανοποιητικό βαθμό, πάντα μπορούν να προστεθούν κι άλλες παράμετροι. Επίσης, μπορεί να γίνει επέκταση ως προς τις συναρτήσεις και τις λειτουργίες ώστε να καλύπτεται κάθε πιθανή περίπτωση εφαρμογής. Το βασικό στοιχείο είναι ότι το σύστημα είναι ευέλικτο οπότε θα είναι πολύ εύκολο να γίνει οποιαδήποτε προσθήκη. Για παράδειγμα, μια τεχνική που θα μπορούσε να χρησιμοποιηθεί στην περίπτωση που το αποτέλεσμα της κινητής συσκευής δεν είναι σίγουρο, όμως ταυτόχρονα δεν υπάρχει η δυνατότητα για αποστολή του δείγματος στον εξυπηρετητή, είτε λόγω φόρτου, είτε λόγω κακής σύνδεσης, είναι η αξιοποίηση των πεποιθήσεων της συμπερασματολογίας της κινητής συσκευής για την επιλογή μιας πιο γενικής (υψηλότερου επιπέδου) κατηγορίας από την ειδική που προέβλεψε το μοντέλο. Αυτό είναι εφικτό διότι οι κλάσεις αντικειμένων του ImageNet είναι δομημένες με βάση την ιεραρχία του Wordnet, οπότε για κάθε αντικείμενο έχουμε εκτός από την ειδική του κλάση και πιο γενικές. Με αυτή την τεχνική, η ακρίβεια του ζεύγους μπορεί να αυξηθεί επιλέγοντας την πιο γενική αλλά σωστή κατηγορία σε σχέση με την πιο ειδική και πιθανώς λανθασμένη.

- **Βελτιστοποίηση των Παραμέτρων**. Ένα πολύ βασικό βήμα που μπορεί να γίνει είναι η βελτιστοποίηση όλων των παραμέτρων. Στη διπλωματική αυτή είδαμε πώς αλλάζουν τη λειτουργία του συστήματος οι πιο βασικές παράμετροι, οι οποίες είναι εύκολο να μεταβληθούν. Μια πιο δύσκολη βελτιστοποίηση θα ήταν η βελτιστοποίηση των νευρωνικών δικτύων, η εύρεση δηλαδή του καλύτερου ζεύγους νευρωνικών δικτύων τόσο για τη συσκευή όσο και για τον εξυπηρετητή.

- **Επιλογή εικόνων**. Η μετρική BvSB φαίνεται να λειτουργεί αρκετά καλά σαν δείκτης επιλογής των εικόνων οι οποίες είναι πιθανό να έχουν κατηγοριοποιηθεί λάθος από την κινητή συσκευή, υπάρχουν όμως κι άλλες ιδέες που αξίζει να διερευνηθούν. Μια μέθοδος είναι η εκπαίδευση ενός δυαδικού ταξινομητή με βάση τα χαρακτηριστικά των εικόνων, όπως αυτά προκύπτουν από ένα ενδιάμεσο σημείο της αρχιτεκτονικής της κινητής συσκευής (early exit). Το σημείο εξαγωγής των χαρακτηριστικών θα αποτελεί παράμετρο προς βελτιστοποίηση και οι ετικέτες εκπαίδευσης θα προκύπτουν από το αν το μοντέλο της κινητής συσκευής έχει ταξινομήσει σωστά ή όχι την κάθε εικόνα του συνόλου εκπαίδευσης. Μια άλλη ιδέα είναι η χρήση μεθόδων ή αλγορίθμων που δεν έχουν σχέση με τη Βαθιά Μάθηση, ώστε να αποφασίζεται αν ένα δείγμα θα ταξινομηθεί σωστά από την κινητή συσκευή χωρίς να γίνει καθόλου inference ή μέρος αυτού.

- **Ανάπτυξη του συστήματος ώστε να λειτουργεί σε πραγματικές κινητές συσκευές.** Αυτή τη στιγμή το σύστημα εκτελείται σε μία μόνο υπολογιστική μηχανή η οποία αναλαμβάνει το ρόλο και του εξυπηρετητή και της κινητής συσκευής. Η ανάπτυξη του συστήματος ώστε να εκτελείται σε πραγματικές κινητές συσκευές και εξυπηρετητές είναι ένα φυσικό επόμενο βήμα.

- **Ιδιωτικότητα**. Η επίτευξη της διατήρησης της ιδιωτικότητας των δεδομένων θα είναι μια πολύ σημαντική επέκταση του συστήματος. Ειδικότερα, για να λειτουργήσει ένα τέτοιο σύστημα σε μεγάλη κλίμακα, η διατήρηση της ιδιωτικότητας κρίνεται απαραίτητη. Με χρήση Early-Exit μοντέλων για παράδειγμα, αν χρειαστεί το δείγμα να αποσταλεί στον εξυπηρετητή, τότε αντί για την εικόνα καθαυτή, μπορεί να αποστέλλονται τα ενδιάμεσα χαρακτηριστικά και το μοντέλο του server να εκτελεί συμπερασματολογία με βάση αυτά. Σε αυτή την περίπτωση, είναι ενδιαφέρον να διερευνηθεί με ποιους τρόπους το μοντέλο του εξυπηρετητή προσαρμόζεται ώστε να μπορεί να αξιοποιεί τα χαρακτηριστικά που του στέλνει η κινητή συσκευή.

Παραρτήματα

Παράρτημα Α΄

Μετρήσεις Μη Αποδοτικών Συμπερασματολογιών

Παρακάτω παραθέτονται αναλυτικά οι πίνακες των μετρήσεων για τις μη αποδοτικές συμπερασματολογίες. Κάθε πίνακας αντιστοιχεί σε διαφορετικό όριο πεποίθησης και μετρική ακρίβειας. Αναφέρονται ο αριθμός των εικόνων που αντιστοιχούν σε κάθε Τύπο καθώς και το ποσοστό τους ως προς τον αριθμό των εικόνων που στάλθηκαν στον εξυπηρετητή.

Πίνακας Α΄.1: *Top-1 Accuracy, στάλθηκαν 5279 εικόνες για όριο πεποίθησης 0.1*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	984	237	1729
Percentage	18.64	4.49	32.75

Πίνακας Α΄.2: *Top-5 Accuracy, στάλθηκαν 5279 εικόνες για όριο πεποίθησης 0.1*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	3493	0	1786
Percentage	66.17	0.00	33.83

Πίνακας Α΄.3: *Top-1 Accuracy, στάλθηκαν 9050 εικόνες για όριο πεποίθησης 0.2*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	1959	434	2823
Percentage	21.65	4.80	31.19

Πίνακας Α΄.4: *Top-5 Accuracy, στάλθηκαν 9050 εικόνες για όριο πεποίθησης 0.2*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	6204	0	2846
Percentage	68.55	0.00	31.45

Πίνακας Α΄.5: *Top-1 Accuracy, στάλθηκαν 11999 εικόνες για όριο πεποίθησης 0.3*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	2923	605	3616
Percentage	24.36	5.04	30.14

Πίνακας Α΄.6: *Top-5 Accuracy, στάλθηκαν 11999 εικόνες για όριο πεποίθησης 0.3*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	8494	0	3505
Percentage	70.79	0.00	29.21

Πίνακας Α΄.7: *Top-1 Accuracy, στάλθηκαν 14607 εικόνες για όριο πεποίθησης 0.4*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	3973	754	4266
Percentage	27.20	5.16	29.21

Πίνακας Α΄.8: *Top-5 Accuracy, στάλθηκαν 14607 εικόνες για όριο πεποίθησης 0.4*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	10607	0	4000
Percentage	72.62	0.00	27.38

Πίνακας Α΄.9: *Top-1 Accuracy, στάλθηκαν 17089 εικόνες για όριο πεποίθησης 0.5*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	5092	893	4874
Percentage	29.80	5.23	28.52

Πίνακας Α΄.10: *Top-5 Accuracy, στάλθηκαν 17089 εικόνες για όριο πεποίθησης 0.5*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	12676	0	4413
Percentage	74.18	0.00	25.82

Πίνακας Α΄.11: *Top-1 Accuracy, στάλθηκαν 19534 εικόνες για όριο πεποίθησης 0.6*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	6355	1003	5412
Percentage	32.53	5.13	27.71

Πίνακας Α΄.12: *Top-5 Accuracy, στάλθηκαν 19534 εικόνες για όριο πεποίθησης 0.6*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	14768	0	4766
Percentage	75.60	0.00	24.40

Πίνακας Α'.13: *Top-1 Accuracy, στάλθηκαν 22093 εικόνες για όριο πεποίθησης 0.7*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	7931	1115	5862
Percentage	35.90	5.05	26.53

Πίνακας Α'.14: *Top-5 Accuracy, στάλθηκαν 22093 εικόνες για όριο πεποίθησης 0.7*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	17048	0	5045
Percentage	77.16	0.00	22.84

Πίνακας Α'.15: *Top-1 Accuracy, στάλθηκαν 24991 εικόνες για όριο πεποίθησης 0.8*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	9873	1237	6346
Percentage	39.51	4.95	25.39

Πίνακας Α'.16: *Top-5 Accuracy, στάλθηκαν 24991 εικόνες για όριο πεποίθησης 0.8*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	19685	0	5306
Percentage	78.77	0.00	21.23

Πίνακας Α'.17: *Top-1 Accuracy, στάλθηκαν 29151 εικόνες για όριο πεποίθησης 0.9*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	13101	1333	6880
Percentage	44.94	4.57	23.60

Πίνακας Α'.18: *Top-5 Accuracy, στάλθηκαν 29151 εικόνες για όριο πεποίθησης 0.9*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	23614	0	5537
Percentage	81.01	0.00	18.99

Πίνακας Α'.19: *Top-1 Accuracy, στάλθηκαν 49948 εικόνες για όριο πεποίθησης 1*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	32695	1434	7756
Percentage	65.46	2.87	15.53

Πίνακας Α'.20: *Top-5 Accuracy, στάλθηκαν 49948 εικόνες για όριο πεποίθησης 1*

Inefficiency Type	Type 1	Type 2	Type 3
Instancies of inefficiency	44075	0	5873
Percentage	88.24	0.00	11.76

Παράρτημα Β΄

Χαρακτηριστικά Μοντέλων Εξυπηρετητή

Παρακάτω παρατίθεται ο Πίνακας χαρακτηριστικών των μοντέλων [47] [48].

Πίνακας Β΄.1: *Μοντέλα Εξυπηρετητή*

Model	Size	Top-1	Top-5	Time (ms)	Memory (GB)
DenseNet121	33 MB	0.75	0.923	8.93	0.67
DenseNet169	57 MB	0.762	0.932	13.03	0.87
DenseNet201	80 MB	0.773	0.936	17.15	0.72
InceptionResNetV2	215 MB	0.803	0.953	25.94	0.87
InceptionV3	92 MB	0.779	0.937	10.1	0.72
MobileNet	16 MB	0.704	0.895	2.45	0.63
MobileNetV2	14 MB	0.713	0.901	3.34	0.63
NASNetLarge	343 MB	0.825	0.96	32.3	1.09
NASNetMobile	23 MB	0.744	0.919	22.36	0.63
ResNet50	98 MB	0.749	0.921	5.1	0.74
ResNet101	171 MB	0.764	0.928	8.9	0.87
ResNet152	232 MB	0.766	0.931	14.31	0.82
VGG16	528 MB	0.713	0.901	5.17	1.46
VGG19	549 MB	0.713	0.9	5.5	1.49
Xception	88 MB	0.79	0.945	6.44	1.03

Βιβλιογραφία

[1] Ian Goodfellow, Yoshua Bengio και Aaron Courville. *Deep Learning*. The MIT Press, Cambridge, Massachusetts, 1η έκδοση, 2016.

[2] Yann LeCun, Yoshua Bengio και Geoffrey Hinton. *Deep learning*. *Nature*, 521, 2015.

[3] Ji Wang, Bokai Cao, Philip Yu, Lichao Sun, Weidong Bao και Xiaomin Zhu. *Deep Learning towards Mobile Applications*. *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, σελίδες 1385–1393, Vienna, Austria, 2018.

[4] Renjie Gu, Shuo Yang και Fan Wu. *Distributed Machine Learning on Mobile Devices: A Survey*. *CoRR*, abs/1909.08329, 2019.

[5] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, Massachusetts, 1η έκδοση, 2012.

[6] *Deep learning vs. machine learning - What's the difference?* https://levity.ai/blog/difference-machine-learning-deep-learning. Ημερομηνία πρόσβασης: 25-08-2021.

[7] John D. Kelleher. *Deep Learning*. The MIT Press, Cambridge, Massachusetts, 1η έκδοση, 2019.

[8] *Deep Learning Training and Inferencel*. https://www.intel.com/content/www/us/en/artificial-intelligence/posts/deep-learning-training-and-inference.html. Ημερομηνία πρόσβασης: 16-09-2021.

[9] Warren S. Sarle. *Neural Networks and Statistical Models*. *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, Cary, NC, USA, 1994.

[10] *Notes for the Stanford CS class "CS231n: Convolutional Neural Networks for Visual Recognition"*. https://cs231n.github.io/. Ημερομηνία πρόσβασης: 25-08-2021.

[11] Saad Albawi, Tareq Abed Mohammed και Saad Al-Zawi. *Understanding of a convolutional neural network*. *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 2017.

[12] Wu Jianxin. *Introduction to convolutional neural networks*. *National Key Lab for Novel Software Technology, Nanjing University*, 2017.

[13] Keiron O'Shea και Ryan Nash. *An Introduction to Convolutional Neural Networks*. *CoRR*, abs/1511.08458, 2015.

[14] Y. Lecun, L. Bottou, Y. Bengio και P. Haffner. *Gradient-based learning applied to document recognition*. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Alex Krizhevsky, Ilya Sutskever και Geoffrey E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, σελίδα 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[16] Matthew D. Zeiler και Rob Fergus. *Visualizing and Understanding Convolutional Networks*. *CoRR*, abs/1311.2901, 2013.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke και Andrew Rabinovich. *Going Deeper with Convolutions*. *CoRR*, abs/1409.4842, 2014.

[18] Karen Simonyan και Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. *CoRR*, abs/1409.1556, 2014.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. *Deep Residual Learning for Image Recognition*. *CoRR*, abs/1512.03385, 2015.

[20] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally και Kurt Keutzer. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size*. *CoRR*, abs/1602.07360, 2016.

[21] François Chollet. *Xception: Deep Learning with Depthwise Separable Convolutions*. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, σελίδες 1800–1807, 2017.

[22] Mingxing Tan και Quoc Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. *Proceedings of the 36th International Conference on Machine Learning*, τόμος 97 στο *Proceedings of Machine Learning Research*, σελίδες 6105–6114. PMLR, 2019.

[23] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto και Hartwig Adam. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. *CoRR*, abs/1704.04861, 2017.

[24] Barret Zoph, Vijay Vasudevan, Jonathon Shlens και Quoc V. Le. *Learning Transferable Architectures for Scalable Image Recognition*. *CoRR*, abs/1707.07012, 2017.

[25] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo και Junshan Zhang. *Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing*. *Proceedings of the IEEE*, 107(8):1738–1762, 2019.

[26] Sawsan Abdul Rahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, Chamseddine Talhi και Mohsen Guizani. *A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond*. *IEEE Internet Things J.*, 8(7):5476–5497, 2021.

[27] Olivier Valery, Pangfeng Liu και Jan Jan Wu. *CPU/GPU Collaboration Techniques for Transfer Learning on Mobile Devices*. *2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, σελίδες 477–484, 2017.

[28] Kieran Woodward, Eiman Kanjo, David J. Brown και T. M. McGinnity. *On-Device Transfer Learning for Personalising Psychological Stress Modelling using a Convolutional Neural Network*. *Proceedings of the 2nd SysML Conference*, Palo Alto, CA, USA, 2019.

[29] Han Cai, Chuang Gan, Ligeng Zhu και Song Han. *TinyTL: Reduce Memory, Not Parameters for Efficient On-Device Learning*. *Advances in Neural Information Processing Systems*H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan και H. Lin, επιμελητές, τόμος 33, σελίδες 11285–11297. Curran Associates, Inc., 2020.

[30] *Example on-device model personalization with TensorFlow Lite*. https://blog.tensorflow.org/2019/12/example-on-device-model-personalization.html. Ημερομηνία πρόσβασης: 27-09-2021.

[31] *General Python FAQ*. https://docs.python.org/3/faq/general.html#id2. Ημερομηνία πρόσβασης: 13-09-2021.

[32] *Comparing Python to Other Languages*. https://www.python.org/doc/essays/comparisons/. Ημερομηνία πρόσβασης: 13-09-2021.

[33] *Matrix-SIG Archives*. https://mail.python.org/pipermail/matrix-sig/. Ημερομηνία πρόσβασης: 24-08-2021.

[34] K. Jarrod Millman και Michael Aivazis. *Python for Scientists and Engineers*. *IEEE Computing in Science Engineering*, 13(2):9–12, 2011.

[35] *What is TensorFlow? The machinge learning library explained*. https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html. Ημερομηνία πρόσβασης: 13-09-2021.

[36] *TensorFlow*. https://github.com/tensorflow/tensorflow. Ημερομηνία πρόσβασης: 24-08-2021.

[37] Navin Kumar Manaswi. *Understanding and Working with Keras*. Apress, Berkeley, CA, 1η έκδοση, 2018.

[38] *Keras API reference*. https://keras.io/api/. Ημερομηνία πρόσβασης: 24-08-2021.

[39] *Keras 2.4.0*. https://github.com/keras-team/keras/releases/tag/2.4.0. Ημερομηνία πρόσβασης: 24-08-2021.

[40] Jia Deng, Wei Dong, Richard Socher, Li Jia Li, Kai Li και Li Fei-Fei. *ImageNet: A large-scale hierarchical image database*. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009.

[41] *ImageNet Large Scale Visual Recognition Challenge*. https://www.image-net.org/challenges/LSVRC/index.php. Ημερομηνία πρόσβασης: 24-08-2021.

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg και Li Fei-Fei. *ImageNet Large Scale Visual Recognition Challenge*. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[43] *TFLite Model Benchmark Tool*. https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/tools/benchmarkl. Ημερομηνία πρόσβασης: 23-09-2021.

[44] *ImageNet Large Scale Visual Recognition Challenge 2012*. https://www.image-net.org/challenges/LSVRC/2012/index.php. Ημερομηνία πρόσβασης: 24-08-2021.

[45] *MobileNet and MobileNetV2*. https://keras.io/api/applications/mobilenet/. Ημερομηνία πρόσβασης: 27-09-2021.

[46] *NasNetLarge and NasNetMobile*. https://keras.io/api/applications/nasnet/. Ημερομηνία πρόσβασης: 27-09-2021.

[47] Simone Bianco, Rémi Cadène, Luigi Celona και Paolo Napoletano. *Benchmark Analysis of Representative Deep Neural Network Architectures*. *IEEE Access*, 6:64270–64277, 2018.

[48] *Keras Applications*. https://keras.io/api/applications/. Ημερομηνία πρόσβασης: 24-09-2021.

Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια

βλ.	βλέπε
εκ.	εκατομμύρια
κ.α	και άλλα
κ.λπ	και λοιπά
π.χ.	παραδείγματος χάρη
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
CNN	Convolutional Neural Network
CPU	Central Processing Unit
GPU	Graphics Processing Unit
ms	millisecond
NPU	Neural Processing Unit
RAM	Random-Access Memory
ReLU	rectified linear unit
RNN	Recurrent Neural Network

Απόδοση Ξενόγλωσσων Όρων

Απόδοση	Ξενόγλωσσος όρος
ακατέργαστος	raw
άκρη του δικτύου	network edge
ακρίβεια	accuracy
Αναδρομικά Νευρωνικά Δίκτυα	Recurrent Neural Networks
Βαθιά Μάθηση	Deep Learning
βάθος	depth
βηματισμός	stride
δεδομένα	data
διαμοιρασμός παραμέτρων	parameter sharing
διατήρηση της ιδιωτικότητας	privacy-preserving
Διεπαφή Προγραμματισμού Εφαρμογών	Application Programming Interface
διεργασία	task
διερμηνέας	interpreter
δομικό στοιχείο	component
δομή	framework
εκατοστημόριο	percentile
εκπαίδευση	training
ελάχιστο	minimum
επίπεδο	layer
εφαρμογή	application
είσοδος	application
ετικέτα	label
ευελιξία	flexibility
ζεύγος νευρωνικών δικτύων	neural network pair
καθυστέρηση	latency
κανονικοποίηση	normalization
κατανεμημένος	distributed
κβαντοποίηση	quantization
Κεντρική Μονάδα Επεξεργασίας	Central Processing Unit
κλιμάκωση	scaling
κινητή συσκευή	mobile device
μέγεθος	size
μέγιστο	maximum
μέσος	mean

μη-αποδοτικός	inefficient
Μηχανική Μάθηση	Machine Learning
Μονάδα Επεξεργασίας Γραφικών	Graphics Processing Unit
Μονάδα Επεξεργασίας Νευρώνων	Neural Processing Unit
μοντέλο	model
νόρμα	norm
Περιβάλλον	Environment
περικοπή	pruning
πεποίθηση	confidence
πλάτος	width
πλήρως συνδεδεμένο	fully-connected
προσθήκη μηδενικών στο σύνορο	zero-padding
πυρήνας	kernel
σιγμοειδής	sigmoid
συμπερασματολογία	inference
συνάρτηση ενεργοποίησης	activation function
Συνελικτικά Νευρωνικά Δίκτυα	Convolutional Neural Networks
συνέλιξη	convolution
συγκέντρωση	pooling
συνολικός	total
σύνολο δεδομένων	dataset
σχήμα	schema
Τεχνητή Νοημοσύνη	Artificial Intelligence
τεχνητό νευρωνικό δίκτυο	artificial neural network
τοπικό δεκτικό πεδίο	local receptive field
ύψος	height
φίλτρο	filter
χαρακτηριστικό	feature
χιλιοστό του δευτερολέπτου	millisecond