ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Είναι αυτή η πινακίδα για εμένα; Αυτόματη αναγνώριση οδικής σήμανσης»

Του φοιτητή
Τσιμερίκα Ανδρέα
Αρ. Μητρώου: 164808

Επιβλέπων
Διαμαντάρας Κωνσταντίνος
Βαθμίδα Καθηγητής

Ημερομηνία 30-1-2024

Τίτλος Δ.Ε. Είναι αυτή η πινακίδα για εμένα; Αυτόματη αναγνώριση οδικής σήμανσης.
Κωδικός Δ.Ε. 22203
Ονοματεπώνυμο φοιτητή Ανδρέας Τσιμερίκας
Ονοματεπώνυμο εισηγητή Κωνσταντίνος Διαμαντάρας
Ημερομηνία ανάληψης Δ.Ε. 30-3-2022
Ημερομηνία περάτωσης Δ.Ε. 30-1-2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Πρόλογος

Η ανίχνευση και η κατηγοριοποίηση των σημάτων οδικής κυκλοφορίας είναι ένα πολύ σημαντικό κομμάτι της αλληλεπίδρασης των αυτόνομων οχημάτων με το περιβάλλον τους με ασφαλή τρόπο. Γι' αυτό το λόγο, αυτή η εργασία σκοπεύει να διερευνήσει την βελτίωση της ανίχνευσης των πιο άμεσα εφαρμόσιμων σημάτων οδικής κυκλοφορίας για την καλύτερη λήψη αποφάσεων από το όχημα σε κάθε περίπτωση, επιτρέποντας στο μοντέλο να δίνει έμφαση στα σήματα που το αφορούν. Επιπλέον, για να επιτευχθεί ο προηγούμενος στόχος επιχειρείται χρήση της αρχιτεκτονικής των Transformers, που ανήκουν στην κατηγορία των νευρωνικών δικτύων και της μηχανικής μάθησης, με μία προσαρμοσμένη συνάρτηση κόστους. Με το τέλος της εργασίας μπορεί να εξαχθεί ένα πρώτο συμπέρασμα αν αυτή η τεχνική μπορεί να έχει πρακτική εφαρμογή και μπορεί να βοηθήσει ένα τέτοιο μοντέλο στην καλύτερη αναγνώριση των σημαντικότερων σημάτων κάθε κατάστασης.

Περίληψη

Η μηχανική μάθηση έχει συμβάλλει σημαντικά στην εξέλιξη του τομέα της αυτόνομης οδήγησης οχημάτων. Με την εξέλιξη, παράλληλα, των Νευρωνικών Δικτύων και χρησιμοποιώντας την νέα αρχιτεκτονική των Transformers, για ανάλυση εικόνας, αυξάνονται οι δυνατότητες για την καλύτερη και ασφαλέστερη λήψη αποφάσεων από τα οχήματα. Έτσι, προέκυψε και η προβληματική της παρούσας εργασίας, που διερευνά, χρησιμοποιώντας ένα υποσύνολο του LISA Traffic Sign Dataset, την βελτίωση ανίχνευσης αποκλειστικά των σημάτων οδικής κυκλοφορίας που αφορούν πιο άμεσα το αυτόνομο όχημα, σε κάθε περίπτωση. Χρησιμοποιήθηκε ,ακόμα, μία προσαρμοσμένη συνάρτηση κόστους του Deformable DETR που να ευνοεί την ανίχνευση των σημαντικότερων σημάτων. Παρατηρήθηκε μικρή βελτίωση στην ικανότητα του μοντέλου να ανιχνεύει και να κατηγοριοποιεί τα σήματα, μεταφέροντας μέρος του υπάρχοντος σφάλματος στα σήματα που δεν αφορούν άμεσα το όχημα. Σύμφωνα με τα αποτελέσματα, λόγο περιορισμένου όγκου δεδομένων και hardware, προτείνεται περαιτέρω διερεύνηση του προβλήματος με καλύτερη κατανομή των σημαντικών και μη σημάτων και μεγαλύτερο όγκο δεδομένων.

«Is this sign for me? Automatic traffic sign recogniction»

«Andreas Tsimerikas»

Abstract

Machine learning has contributed significantly to the development of the field of autonomous driving. With the evolution of Neural Networks and by using the new architecture of Transformers, for image analysis, the possibilities for better and safer decision making by vehicles are increasing. With that been said, the current thesis investigates, the improvement of detection exclusively of road traffic signs that are more directly related to the autonomous vehicle, in each case by using a subset of the LISA Traffic Sign Dataset. An adapted cost function of Deformable DETR was also used to encourage the detection of the most important signs. A small improvement was observed in the model's ability to detect and categorize the signs, transferring some of the existing error to the vehicle non-related signs. According to the results, due to the limited amount of data and hardware, it is suggested to further investigate the problem with a better distribution of important and non-important signs with more cases of it’s type.

Περιεχόμενα

Κατάλογος Εικόνων και Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
SVM	Support Vector Machines
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RCNN	Recurrent Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformer
FFN	Feed-Forward neural Network
ReLU	Rectified Linear Unit
R-CNN	Region-based Convolutional Neural Network
RPN	Region Proposal Network
R-FCN	Region-based Fully Convolutional Network
DETR	DEtection TRansformer
YOLO	You Only Look Once
SSD	Single Shot MultiBox Detector

Κεφάλαιο 1ο: Εισαγωγή στην προβληματική της εργασίας

1.1 Εισαγωγή και στόχος εργασίας

Η ανίχνευση σημάτων οδικής κυκλοφορίας με τη χρήση της τεχνητής νοημοσύνης σε αυτόνομα οχήματα αποτελεί ένα κρίσιμο τμήμα της αυτόνομης οδήγησης που έχει πολλά υποσχόμενα πλεονεκτήματα, όπως η μείωσή του κόστους των μετακινήσεων, των επιπτώσεων της περιβαλλοντικής μόλυνσης αλλά και την μείωση των αυτοκινητικών ατυχημάτων [1]. Η ικανότητα ενός αυτόνομου οχήματος να αναγνωρίζει και να ερμηνεύει τα οδικά σήματα είναι κρίσιμη για την ασφαλή και αποτελεσματική λειτουργία του και για την αντίδρασή του σε διάφορες καταστάσεις. Φυσικά, υπάρχουν και πιο περίπλοκοι παράγοντες που χρειάζεται να λαμβάνει υπόψη όπως γραμμές στο οδόστρωμα, την κατάσταση του οδοστρώματος, την θέση άλλων οχημάτων, πεζών αλλά και άλλων αντικειμένων και καταστάσεις που είναι δύσκολο να προβλεφθούν. Οι δρόμοι στους οποίους καλείτε να κινηθεί ένα αυτόνομο όχημα έχουν κατασκευαστεί για την καλύτερη χρήση από ανθρώπους οδηγούς, οπότε σε αυτήν την περίπτωση το αυτοκινούμενο όχημα χρειάζεται να μιμηθεί έναν τέτοιο οδηγό, το οποίο περιλαμβάνει πολλές λειτουργίες όπως η επιλογή κατεύθυνσης ανάλογα τον τελικό μας προορισμό, η επιλογή λωρίδας ανάλογα με την κατεύθυνση που θέλουμε να πάρουμε, ο έλεγχος των σημάτων οδικής κυκλοφορίας ή των τροχονόμων αλλά και διάφορες άλλες μικρο-λειτουργίες που μπορεί να κάνουμε και ασυναίσθητα. Σε αυτήν την εργασία δεν θα αναλυθούν όλοι οι παράγοντες παρά μόνο η αναγνώριση των σημάτων οδικής κυκλοφορίας καθώς και η αναγνώριση από το όχημα αν θα πρέπει να ληφθούν υπόψη.

Έτσι, η τεχνητή νοημοσύνη με τεχνικές βαθιάς μάθησης και τη χρήση πολλαπλών εργαλείων έχει επιτρέψει την ανάπτυξη συστημάτων που μπορούν να αναγνωρίζουν και να ερμηνεύουν ένα ευρύ φάσμα σημάτων οδικής κυκλοφορίας, προσφέροντας την ικανότητα στα αυτόνομα οχήματα να αντιλαμβάνονται το περιβάλλον τους και να λαμβάνουν ακριβείς αλλά και ασφαλείς αποφάσεις. Απαιτούνται, επίσης, μεγάλα σύνολα δεδομένων που περιέχουν εικόνες απεικόνισης σημάτων σε πραγματικές συνθήκες οδήγησης. Τα δίκτυα αυτά μπορούν να εκπαιδευτούν να ανιχνεύουν και να αναγνωρίζουν τα σήματα με υψηλή ακρίβεια και ταχύτητα, παρέχοντας έτσι την κατάλληλη απόκριση του οχήματος.

Για να γίνει επιτυχώς η ανίχνευση και η αναγνώριση του σήματος και έπειτα ο βαθμός εφαρμοσιμότητας στο αυτόνομο όχημα, επιχειρείται η μέγιστη δυνατή αναγνώριση των σημάτων με διάφορα είδη χαρακτηριστικών όπως, η θέση στην εικόνα, το μέγεθος και η στροφή του σήματος στο χώρο που μπορεί να βοηθήσουν το μοντέλο να κάνει σωστή πρόβλεψη.

Τα εργαλεία που θα χρησιμοποιηθούν είναι τα νευρωνικά δίκτυα, των οποίων η πολυπλοκότητα συνεχώς αυξάνεται. Αυτή η αύξηση στην πολυπλοκότητα ενός τόσο σημαντικού συστατικού της μηχανικής μάθησης έχει ως συνέπεια πολλές φορές το αποτέλεσμα των αλγορίθμων να μην είναι το αναμενόμενο και να μην μπορεί να εξηγηθεί. Επομένως, κρίνεται απαραίτητο οι αποφάσεις του αυτόνομου οχήματος να περνούν από μια διαδικασία αλληλοεπαλήθευσης πολλών διαφορετικών πηγών και μεθόδων [2].

Κεφάλαιο 2ο: Μηχανική Μάθηση

2.1 Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη είναι ένα πεδίο της επιστήμης των υπολογιστών που έχει απασχολήσει ιδιαίτερα τους επιστήμονες τα τελευταία χρόνια και ορίζεται ως "η μελέτη και η δημιουργία προγραμμάτων ηλεκτρονικού υπολογιστή που σκοπό έχουν να συμπεριφέρονται έξυπνα". [3] Ο όρος έξυπνα δεν έχει συγκεκριμένη ερμηνεία, αλλά για τον χώρο της τεχνητής νοημοσύνης μπορεί να ερμηνευθεί σαν: ικανότητα να λύνουν προβλήματα, να μαθαίνουν από προηγούμενες εκτελέσεις, να "καταλαβαίνουν" δύσκολες καταστάσεις και δεδομένα, και αντιλαμβάνονται διαφορές και ομοιότητες μεταξύ καταστάσεων".

Αυτές οι μηχανές σχεδιάζονται να μιμούνται την ανθρώπινη νοημοσύνη και να εκτελούν καθήκοντα όπως η αναγνώριση εικόνας, η φωνητική αναγνώριση, η αναγνώριση προτύπων, η λήψη αποφάσεων, και πολλά άλλα. Γενικά έχουν επικρατήσει δύο τύποι τεχνητής νοημοσύνης [4] που ομαδοποιούν τα προγράμματα και είναι οι παρακάτω:

1. Η Κλασική Τεχνητή Νοημοσύνη.

2. Η Υπολογιστική Νοημοσύνη ή Ευέλικτος Λογισμός που μέρος της είναι η βαθιά μάθηση που θα αξιοποιηθεί ως εργαλείο για αυτήν την εργασία.

2.2 Μηχανική Μάθηση

Μέρος της γενικότερης κατηγορίας της τεχνητής νοημοσύνης είναι και η μηχανική μάθηση, στην οποία αναπτύσσονται συστήματα όπου μέσω ειδικών αλγορίθμων και τεχνικών, αναπροσαρμόζονται χρησιμοποιώντας τα διαθέσιμα δεδομένα για την λύση κάποιου προβλήματος. Η μηχανική μάθηση προσφέρει λύσεις σε προβλήματα που η περιπλοκότητα τους τα καθιστά ακατάλληλα για επίλυση με την συμβατική μέθοδο προγραμματισμού. Εξαιτίας της ποικιλίας και της πολυπλοκότητας των προβλημάτων θα ήταν αδύνατον να παρέχουμε απλά κάθε δυνατό κανόνα και εντολή που απαιτείται για την επίλυση.

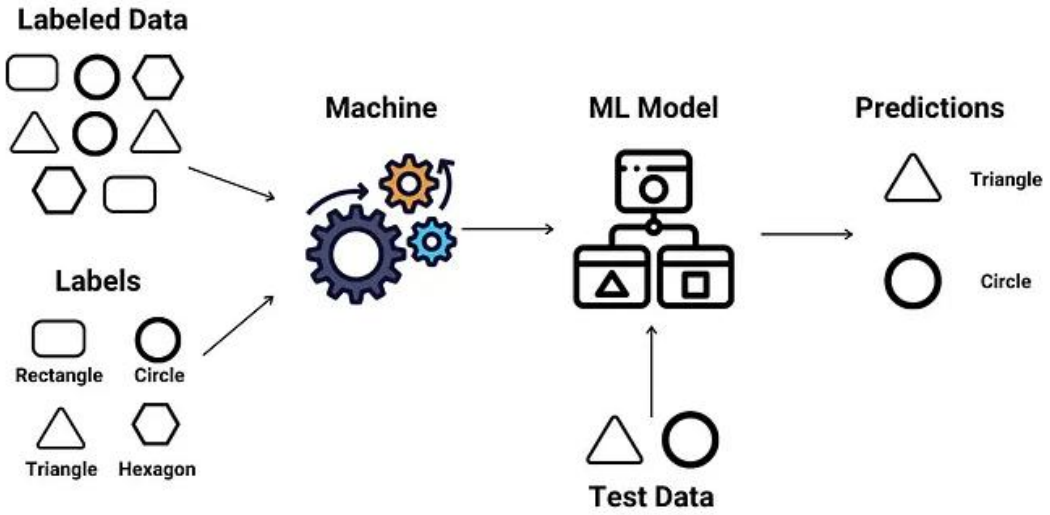
Η συμβατική μέθοδος λειτουργεί σε δύο βασικά βήματα. Το πρώτο είναι η παραγωγή ενός σχεδίου εντολών ή κανόνων που καταρτίζουν την λειτουργία του προγράμματος και την διαδικασία που θα ακολουθήσει για τη λύση του προβλήματος. Έπειτα, επιλέγεται κάποια γλώσσα προγραμματισμού για την υλοποίηση του σχεδίου. Αντίθετα, η μηχανική μάθηση λειτουργεί διαφορετικά. Στο πρόγραμμα διοχετεύεται ένα μεγάλος όγκος δεδομένων που αφορούν το πρόβλημα καθώς και τα αποτελέσματα στα οποία αναμένουμε να καταλήξει. Έτσι, το πρόγραμμα περνάει από μια διαδικασία "μάθησης" για να καταλήξει στους κανόνες και τις εντολές που απαιτούνται για την λύση του προβλήματος.

Η ιδέα, λοιπόν, της μηχανικής μάθησης είναι να επιτραπεί στους υπολογιστές να αντλούν συμπεράσματα από τα δεδομένα χωρίς να προγραμματίζονται συγκεκριμένα για κάθε εργασία αλλά αντίθετα να καταλήγουν σε ενέργειες μετά από μια πορεία "μάθησης". Παράδειγμα ενός τέτοιου προβλήματος είναι η διαλογή της αλληλογραφίας σε ένα e-mail και η ταξινόμηση της σε ανεπιθύμητη. Ο όγκος και η περιπλοκότητα των παραμέτρων που περιλαμβάνονται δεν επιτρέπουν την κατάρτιση ενός λεπτομερούς κώδικα εντολών και κανόνων.

2.3 Τύποι Μηχανικής Μάθησης

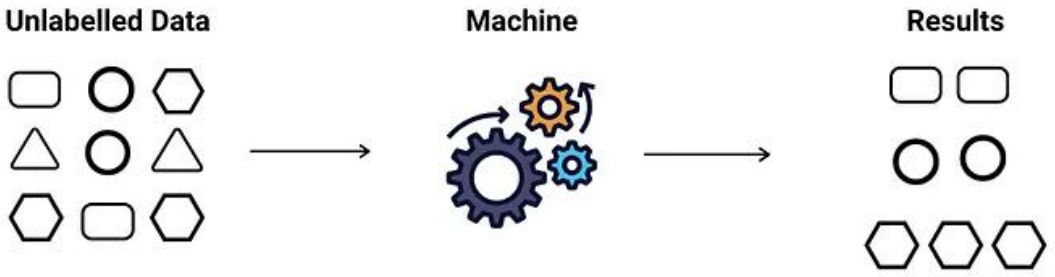
Στα πλαίσια της μηχανικής μάθησης, υπάρχουν τρεις βασικές κατηγορίες που αφορούν στην ποικιλία στην εκπαίδευση του μοντέλου μηχανικής μάθησης:

Επιβλεπόμενη Μάθηση (Supervised Learning): Το μοντέλο εκπαιδεύεται χρησιμοποιώντας κάποια δεδομένα εισόδου και εξόδου, που περιλαμβάνουν ετικέτες για την εκπαίδευση των αλγορίθμων. Στόχος είναι να συνδέει σωστά αυτά τα δεδομένα ώστε να προβλέπει σωστά τις αποκρίσεις για νέες εισόδους. Συνήθως τα δεδομένα που διοχετεύονται σε τέτοια μοντέλα είναι με τη μορφή tensor. Τα προβλήματα που λύνει χωρίζονται σε προβλήματα ταξινόμησης και παλινδρόμησης.



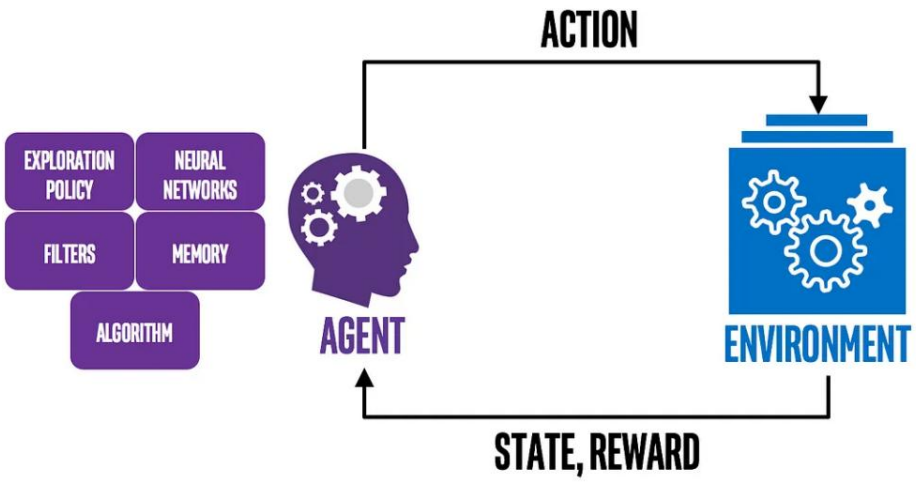
Εικόνα 2.1: Παράδειγμα μεθοδολογίας μάθησης με επίβλεψη. [5]

Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning): Εδώ, το μοντέλο εκπαιδεύεται χωρίς να έχει ετικέτες για τα δεδομένα με τα οποία το τροφοδοτούμε. Ο στόχος είναι να ανακαλύψει μοτίβα ή να συσχετίσει τα δεδομένα με τις ετικέτες τους. Έπειτα, έχει τη δυνατότητα να παράγει σωστά αποτελέσματα, προβλέποντας τις ετικέτες για καινούρια δεδομένα για τα οποία δεν εκπαιδεύτηκε. Τα προβλήματα που καλείται να λύσει αποκαλούνται προβλήματα εκτίμησης κατανομής και συσταδοποίησης (clustering).



Εικόνα 2.2: Παράδειγμα μεθοδολογίας μάθησης χωρίς επίβλεψη. [5]

Μάθηση με ενίσχυση (Reinforcement Learning): Αυτά τα μοντέλα επικεντρώνονται στην διαδικασία λήψεις αποφάσεων. Σε αυτήν την περίπτωση, ένα μοντέλο αλληλεπιδρά με ένα περιβάλλον, και ο στόχος του είναι να μάθει να πραγματοποιεί ενέργειες που μεγιστοποιούν μια ανταμοιβή και αποφεύγουν την τιμωρία. Ένα τέτοιο παράδειγμα είναι ένα μοντέλο που έχει εκπαιδευτεί να παίζει σκάκι. Παρέχοντας του χιλιάδες πραγματοποιηθέντα παιχνίδια και το αποτέλεσμα τους, το εκπαιδεύουμε να κερδίζει στο παιχνίδι και να επιβραβεύεται αντιστοίχως.

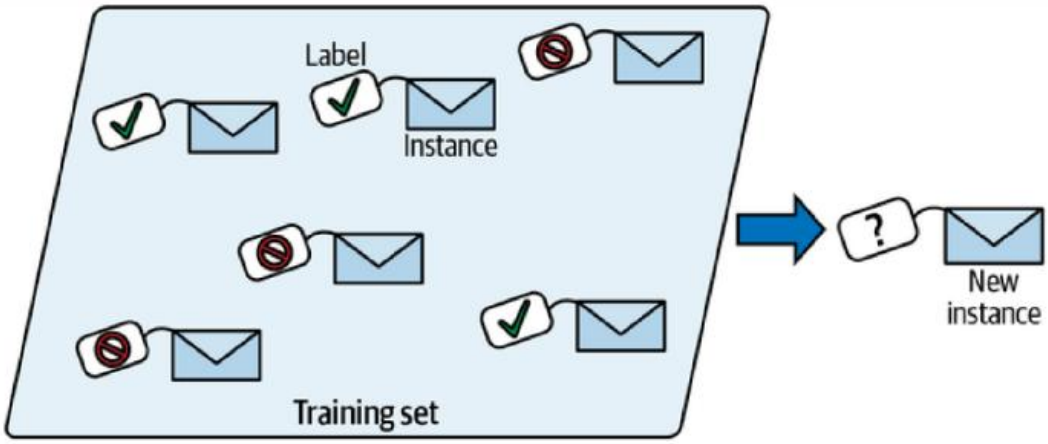


Εικόνα 2.3: Παράδειγμα μεθοδολογίας μάθησης με ενίσχυση. [6]

2.3.1 Supervised Learning

Στην Επιβλεπόμενη μάθηση [7][8][9], το μοντέλο διαχειρίζεται δεδομένα με ετικέτες. Η ανίχνευση μοτίβων είναι ο κύριος τρόπος διαχείρισης των δεδομένων που τροφοδοτούμε στο μοντέλο. Επιπλέον, εκπαιδεύεται στην ανίχνευση σχέσεων μεταξύ των δεδομένων και των ετικετών τους, προσαρμόζοντας αντίστοιχα τα βάρη. Σκοπός είναι να είναι σε θέση το μοντέλο να παράγει σωστά αποτελέσματα για νέα δεδομένα για τα οποία δεν το έχουμε εκπαιδεύσει ποτέ.

Η επιβλεπόμενη μάθηση λύνει δύο τύπους προβλημάτων. Ο πρώτος τύπος περιλαμβάνει τα προβλήματα **ταξινόμησης (classification)**. Οι αλγόριθμοι ταξινόμησης εκπαιδεύονται με σκοπό να ταξινομούν αποτελεσματικά τα δεδομένα που εισάγονται σε έναν ορισμένο αριθμό κλάσεων, βασιζόμενοι στις ετικέτες για τις οποίες έχει εκπαιδευτεί, όπως φαίνεται στην παρακάτω εικόνα από το βιβλίο του Aurelien Geron (2023):



Εικόνα 2.4: Πρόβλημα classification: Ταξινόμηση e-mail σε spam ή όχι. [8]

Αυτοί οι αλγόριθμοι βρίσκουν εφαρμογή σε δυαδικές ταξινομήσεις όπως το φιλτράρισμα για spam ή όχι spam e-mail που εισέρχονται σε μια διεύθυνση ηλεκτρονικού ταχυδρομείου, αναγνώριση και διαχωρισμό μιας γάτας από ένα σκύλο κ.α.

Σε αυτού του τύπου προβλήματα απαιτείται από τον αλγόριθμο να αποφασίσει σε ποια κατηγορία ανήκουν τα δεδομένα που εισάγουμε. Στο παρακάτω τύπο θεωρούμε f το μοντέλο που εκπαιδεύεται και x τα δεδομένα που εισάγουμε:

$$f: \mathbb{R}^n \rightarrow \{1, \dots k\}$$

(2.1) [9]

Όσον αφορά τα μέτρα εκτίμησης, στους αλγορίθμους ταξινόμησης χρησιμοποιούνται οι πίνακες σύγχυσης. Έτσι, η πρόβλεψη του μοντέλου μπορεί να είναι μία από τις ακόλουθες:

1) True positive (TP): η πρόβλεψη του μοντέλου είναι πραγματικά θετική
2) True negative (TN): η πρόβλεψη του μοντέλου είναι πραγματικά αρνητική
3) False positive (FP): η πρόβλεψη του μοντέλου είναι εσφαλμένα θετική
4) False negative (FN): η πρόβλεψη του μοντέλου είναι εσφαλμένα αρνητική

Ο υπολογισμός όλων των μέτρων εκτίμησης που απαιτούνται για ένα πρόβλημα ταξινόμησης, μπορούν να υπολογιστούν μέσω του πίνακα σύγχυσης. Επίσης, η ακρίβεια (accuracy) είναι ένα από τα πιο κοινά μέτρα εκτίμησης που χρησιμοποιείται στα προβλήματα ταξινόμησης και υπολογίζει το ποσοστό επιτυχίας με το οποίο ο αλγόριθμος προβλέπει τις κλάσεις. Σε περιπτώσεις, βέβαια, που το σύνολο δεδομένων (dataset) περιλαμβάνει μη ισορροπημένες κλάσεις δεδομένων, αυτό το μέτρο εκτίμησης είναι αδύναμο. Για αυτόν τον λόγο συνυπολογίζονται και άλλοι τύποι όπως η ακρίβεια, η ευστοχία, η ανάκληση και το F1-score, όπως φαίνονται παρακάτω:

$$\boldsymbol{Accuracy} = \frac{TN + TP}{TN + TP + FP + FN}$$

(2.2) [9]

$$\boldsymbol{Precision} = \frac{TP}{TP + FP}$$

(2.3) [9]

$$\boldsymbol{Recall} = \frac{TP}{TP + FN}$$

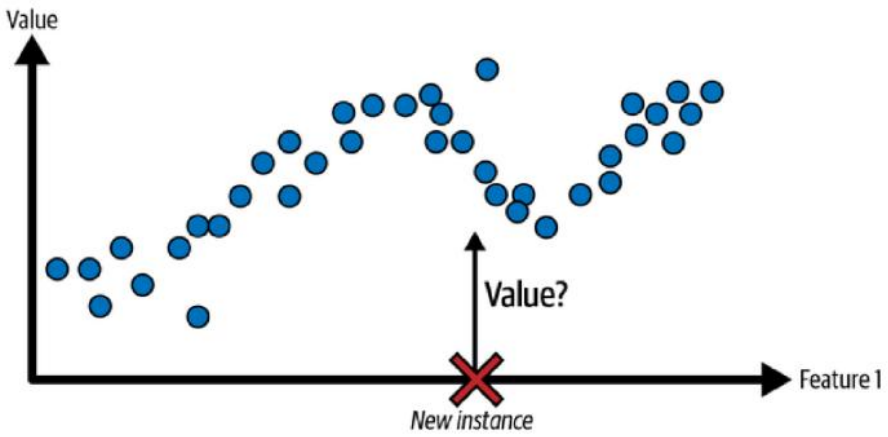
(2.4) [9]

$$\boldsymbol{F1 - score} = 2 * \frac{Precision * Recall}{Presicion + Recall}$$

(2.5) [9]

Το μέτρο εκτίμησης F1-score, είναι από τα πιο ισορροπημένα μέτρα εκτίμησης που βελτιώνει τα σημεία στα οποία τα άλλα μέτρα εκτίμησης δεν αποδίδουν τα επιθυμητά αποτελέσματα.

Ο επόμενος τύπος προβλημάτων για τα οποία χρησιμοποιείται η μάθηση με επίβλεψη είναι τα προβλήματα **παλινδρόμησης (regression)**. Σε αυτά τα προβλήματα ο αλγόριθμος χρησιμοποιώντας κάποια δεδομένα εισόδου προσπαθεί να προβλέψει μια αριθμητική έξοδο. Το κύριο χαρακτηριστικό που διαφοροποιεί αυτούς τους αλγορίθμους από τους αλγορίθμους ταξινόμησης είναι η μορφή των δεδομένων εξόδου. Στην παρακάτω εικόνα φαίνεται ένα πρόβλημα ταξινόμησης και πάλι από το βιβλίο του Aurelien Geron (2023):



Εικόνα 2.5: Πρόβλημα regression: πρόβλεψη τιμής με βάση κάποια δεδομένα εισαγωγής. [8]

Χαρακτηριστικά παραδείγματα παλινδρόμησης είναι η πρόβλεψη τιμών για διάφορα αγαθά, ακόμα και για ακίνητα, η πρόβλεψη καιρικών συνθηκών (θερμοκρασία, υγρασία, βροχή, κλπ) κ.α. Σε αυτά τα προβλήματα προσπαθούμε να εκτιμήσουμε τη συνάρτηση f η οποία παράγει την επιθυμητή τιμή για το εκάστοτε πρόβλημα.

Τα μέτρα εκτίμησης στα προβλήματα παλινδρόμησης είναι στην ουσία η ίδιες συναρτήσεις σφάλματος, που θα αναλυθούν σε παρακάτω κεφάλαιο, καθώς οι προβλέψεις είναι πραγματικοί αριθμοί και δεν έχει κάποιο νόημα η μέτρηση της σωστής πρόβλεψης αλλά της απόστασης από τον στόχο. Παρακάτω αναφέρονται εν συντομία η ονομασία και οι τύποι υπολογισμού τους:

Mean squared error (MSE):

$$(y_i - \bar{y}_i)J_{MSE} = \sum_{p=1}^{N} \left\| t_p - y_p \right\|^2 = \sum_{p=1}^{N} \sum_{i=1}^{m} \left(t_{p,i} - y_{p,i} \right)^2$$

(2.6) [9]

Mean absolute error (MAE):

$$J_{MAE} = \sum_{p=1}^{N} \sum_{i=1}^{m} \left| t_{p,i} - y_{p,i} \right|$$

(2.7) [9]

Cosine similarity:

$$J_{cos} = \frac{t^T y}{\|t\| \|y\|} = \frac{\sum_{i=1}^{m} t_i y_i}{\sqrt{\sum_{i=1}^{m} t^2} \sqrt{\sum_{i=1}^{m} y^2}}$$

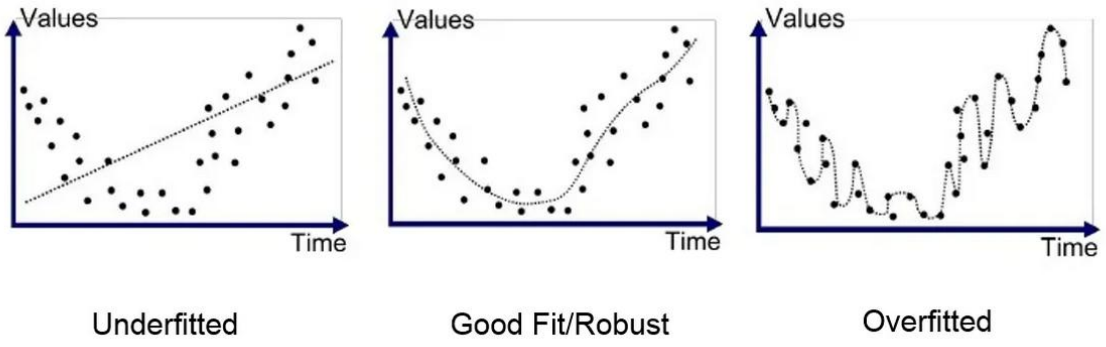
(2.8) [9]

Pearson similarity:

$$J_P = \frac{(t - \bar{t})^2 (y - \bar{y})}{\|t - \bar{t}\| \|y - \bar{y}\|} = \frac{\sum_{i=1}^{m} (t_i - \bar{t}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^{m} (t_i - \bar{t}_i)^2} \sqrt{\sum_{i=1}^{m} (y_i - \bar{y}_i)^2}}$$

(2.9) [9]

Για να κρίνουμε αν το μοντέλο μας κάνει καλή γενίκευση, δηλαδή είναι ικανό να μας δίνει σωστές προβλέψεις για άγνωστα δεδομένα στα οποία δεν έχει εκπαιδευτεί, χρησιμοποιούμε τα παραπάνω μέτρα εκτίμησης που αναφέρθηκαν. Αν το μοντέλο δεν καταφέρνει να μας δώσει σωστές προβλέψεις τότε ή το μοντέλο είναι πολύ απλό και δεν καταφέρνει να προσαρμοστεί στα δεδομένα ή συμβαίνει το αντίστροφο και το μοντέλο προσαρμόζεται υπερβολικά καλά στα δεδομένα εκπαίδευσης και αδυνατεί να κάνει σωστές προβλέψεις για άλλα δεδομένα στα οποία δεν έχει εκπαιδευτεί. [10] Οι δύο αυτές καταστάσεις ονομάζονται Underfitting και Overfitting αντίστοιχα και όταν καταφέρνουμε να τις αναγνωρίζουμε μας βοηθούν στο να κάνουμε τις απαραίτητες τροποποιήσεις στα δεδομένα εκπαίδευσης ή στην αρχιτεκτονική του μοντέλου για να πετύχουμε καλύτερη γενίκευση.



Εικόνα 2.6: Γραφήματα σύγκρισης περιπτώσεων Underfitting/Overfitting με περίπτωση καλής γενίκευσης. [10]

Τρόποι μείωσης Underfitting:

- Καλύτερη επιλογή δεδομένων εκπαίδευσης με καλύτερη προ επεξεργασία και έλεγχο για άχρηστα δείγματα επιτυγχάνοντας ομοιόμορφη κατανομή των δειγμάτων χωρίς ακραία δείγματα.
- Αύξηση της πολυπλοκότητας του μοντέλου με τροποποιήσεις στα κρυφά στρώματα.
- Αύξηση στην διάρκεια εκπαίδευσης

Τρόποι μείωσης Overfitting:

- Καλύτερη επιλογή δεδομένων εκπαίδευσης με καλύτερη προ επεξεργασία και έλεγχο για άχρηστα δείγματα επιτυγχάνοντας ομοιόμορφη κατανομή των δειγμάτων χωρίς ακραία δείγματα.
- Μείωση της πολυπλοκότητας του μοντέλου με τροποποιήσεις στα κρυφά στρώματα, εισαγωγή βοηθητικών στρωμάτων που χρησιμοποιούνται για την μείωση του overfitting όπως το Dropout που θα αναλυθεί σε μεταγενέστερο κεφάλαιο.
- Μείωση στην διάρκεια της εκπαίδευσης και την ένταση κατά την διάρκεια της.

2.3.2 Unsupervised Learning

Σε αντίθεση με τη μάθηση με επίβλεψη [11][12][9], η μάθηση χωρίς επίβλεψη περιλαμβάνει δεδομένα μη ταξινομημένα και χωρίς ετικέτες τα οποία ο αλγόριθμος καλείται να επεξεργαστεί. Στόχος του μοντέλου είναι από την διαδικασία μάθησης να εξάγει πληροφορίες για τα δεδομένα, εκτιμώντας την κατανομή πιθανότητας μέσω δειγμάτων. [9] Έτσι, η επεξεργασία στην οποία υποβάλλονται έχει σκοπό την ομαδοποίηση των μη ταξινομημένων δεδομένων, εντοπίζοντας μοτίβα, ομοιότητες και διαφορές. Το μοντέλο δεν έχει υποβληθεί σε προηγούμενη εκπαίδευση για τα δεδομένα που εισάγονται. Οι δύο τύποι προβλημάτων που λύνει αφορούν την ομαδοποίηση ή συσταδοποίηση (clustering) και στους κανόνες συσχετίσεων (association rules).

Διακρίνοντας τις ομοιότητες και τις διαφορές των δεδομένων, η τεχνική της **ομαδοποίησης (clustering)** κατανέμει σε ομάδες τα δεδομένα χωρίς ετικέτα. Οι ομάδες στις οποίες

κατηγοριοποιούνται τα δεδομένα χαρακτηρίζονται από δομές ή μοτίβα που προκύπτουν κατά την διαδικασία μάθησης του αλγορίθμου. Οι τύποι αλγορίθμων ομαδοποίησης είναι οι εξής:

- συγκεκριμένα αποκλειστικοί (specifically exclusive)
- επικαλυπτόμενος (overlapping)
- ιεραρχικούς (hierarchical)
- πιθανολογικούς (probabilistic)

Η ομαδοποίηση έχει εφαρμογή σε διάφορους κλάδους όπως στην εμπορία, τη διαφήμιση, τις ψηφιακές βιβλιοθήκες,στον κλάδο της ασφάλισης, στη βιολογία, στη σεισμολογία κ.α. [9]

Οι κανόνες συσχέτισης **(association rules)** σκοπεύουν στην εύρεση συσχετίσεων και συν-εμφανίσεων ανάμεσα στα σύνολα των δεδομένων. Οι συσχετίσεις που επιτυγχάνει απεικονίζονται με τη μορφή κανόνων ή συχνών συνόλων στοιχείων (frequent itemsets).

2.3.3 Reinforcement Learning

Η μάθηση με ενίσχυση [13][9], αποτελεί ένα μοντέλο μάθησης όπου ο στόχος είναι η λήψη αποφάσεων. Η ιδέα βασίζεται στην ύπαρξη ενός μοντέλου και ενός περιβάλλοντος. Ο πράκτορας (agent) όπως, αποκαλείται το μοντέλο, αλληλεπιδρά με το περιβάλλον εκτελώντας ενέργειες (actions) από τις οποίες αποκομίζει ανταμοιβή (reward). [9] Η ανταμοιβή μπορεί να γίνει διαθέσιμη στον πράκτορα άμεσα (απευθείας μετά την ενέργεια) αλλά και έμμεσα (έπειτα από ένα σύνολο ορθών ενεργειών). Παρέχονται στο μοντέλο τα απαραίτητα δεδομένα, όπως και οι επιβραβεύσεις ή οι τιμωρίες. Εκπαιδεύεται έτσι στην αποφυγή της ήττας αλλά και στην προσπάθεια για επιβράβευση.

Οι βασικοί συντελεστές της ενισχυτικής μάθησης είναι ο πράκτορας που λαμβάνει αποφάσεις (agent), το περιβάλλον στο οποίο λαμβάνονται οι αποφάσεις, οι δράσεις που εκτελούνται από τον επικεφαλής, η ανταμοιβή ή η τιμωρία που προκύπτει από κάθε δράση, και ο στόχος του επικεφαλής, που είναι συνήθως η μεγιστοποίηση του συνολικού κέρδους. Η ενισχυτική μάθηση περιλαμβάνει τη χρήση αλγορίθμων που μαθαίνουν από τα αποτελέσματα των δράσεών τους, και προσαρμόζουν τη στρατηγική τους για να βελτιώσουν την απόδοσή τους στον χρόνο. Κάποιες βασικές έννοιες στην ενισχυτική μάθηση περιλαμβάνουν:

- **Κατάσταση (State)**: Περιγράφει την τρέχον κατάσταση του περιβάλλοντος, που επηρεάζει την απόδοση του επικεφαλής.
- **Δράση (Action)**: Είναι η ενέργεια που αποφασίζει ο επικεφαλής να εκτελέσει σε μια συγκεκριμένη κατάσταση.
- **Κίνητρο (Reward)**: Είναι η ανταμοιβή ή η τιμωρία που λαμβάνει ο επικεφαλής μετά από μια δράση, που χρησιμεύει ως κριτήριο για την προσαρμογή της στρατηγικής του.
- **Πολιτική (Policy)**: Είναι ο τρόπος με τον οποίο ο επικεφαλής επιλέγει δράσεις σε κάθε κατάσταση.

Δύο πολύ σημαντικές έννοιες αυτού του μοντέλου μάθηση είναι η εξερεύνηση (exploration) και η εκμετάλλευση (exploitation). Η εξερεύνηση αφορά στην δοκιμή ποικίλων επιλογών προς ενέργεια με σκοπό την διερεύνηση των πιθανών ανταμοιβών που θα προκύψουν από αυτές ενώ η εκμετάλλευση

επικεντρώνεται στην αξιοποίηση της γνώσης που έχει αποκομίσει ο πράκτορας για να επιλέξει την καλύτερη ενέργεια σε κάθε περίπτωση.

Διάφοροι αλγόριθμοι ενισχυτικής μάθησης περιλαμβάνουν τον Q-learning, τον πολιτική-αξιολόγηση (policy evaluation), και τα βαθιά ενισχυτικά νευρωνικά δίκτυα (deep reinforcement learning). Εφαρμογές της ενισχυτικής μάθησης περιλαμβάνουν τον έλεγχο ρομπότ, τα παιχνίδια και την αυτόνομη οδήγηση.

2.3.4 Βαθιά Μάθηση

Η βαθιά μάθηση (deep learning) [9][14] αναφέρεται σε ένα υποσύνολο της μηχανικής μάθησης όπου τα μοντέλα, γνωστά ως βαθιά νευρωνικά δίκτυα, αποτελούνται από πολλά επίπεδα νευρώνων που λειτουργούν αντιπροσωπευτικά. Το θεώρημα του καθολικού προσεγγιστή υποστηρίζει πως σε ένα τεχνητό νευρωνικό δίκτυο ένα και μόνο κρυφό στρώμα σε συνδυασμό με επαρκές αλλά πεπερασμένο πλήθος νευρώνων δύναται να προσομοιώσει οποιαδήποτε συνεχής συνάρτηση. [15] Η βαθιά μάθηση έχει καταφέρει να επιλύσει πολύπλοκα προβλήματα και να πετύχει εντυπωσιακά αποτελέσματα σε πολλούς τομείς, όπως η αναγνώριση εικόνων, η φωνητική αναγνώριση, η μηχανική μετάφραση, και πολλά άλλα. Πρακτικά βέβαια, η πολυπλοκότητα των προβλημάτων που καλείται να λύσει η βαθιά μάθηση είναι πολύ μεγαλύτερη. Περιλαμβάνεται πλήθος παραμέτρων αλλά και εξοπλισμού που να ανταπεξέρχονται σε πολύ απαιτητικές λειτουργίες. Μία από τις βασικές έννοιες της βαθιάς μάθησης είναι η **Βαθιά Αρχιτεκτονική Νευρωνικών Δικτύων** (Deep Neural Network - DNN).

2.3.5 Νευρωνικά Δίκτυα

Τα νευρωνικά δίκτυα αναπτύχθηκαν εμπνευσμένα από τον τρόπο με τον οποίο λειτουργεί το ανθρώπινο εγκέφαλο. Είναι μια κατηγορία μοντέλων μηχανικής μάθησης που χρησιμοποιούνται ευρέως σε διάφορους τομείς. Τα νευρωνικά δίκτυα ανήκουν στην οικογένεια της βαθιάς μάθησης και συνίστανται από συνδεδεμένους κόμβους, επίσης γνωστούς ως νευρώνες. Αυτοί οι κόμβοι οργανώνονται σε επίπεδα, και τα σήματα περνούν μέσα από το δίκτυο από το είσοδο στην έξοδο, όπου γίνεται η τελική απόφαση ή πρόβλεψη. Τα νευρωνικά δίκτυα έχουν κατορθώσει να λύσουν πολύπλοκα προβλήματα και να επιτύχουν εντυπωσιακή απόδοση σε πολλούς τομείς, κυρίως χάρη στη δυνατότητά τους να αναγνωρίζουν πολύπλοκα πρότυπα σε μεγάλα σύνολα δεδομένων.

Ορισμένα κύρια στοιχεία των νευρωνικών δικτύων περιλαμβάνουν:

Νευρώνας (Neuron): Ο βασικός υπολογιστικός μονάδα σε ένα νευρωνικό δίκτυο. Κάθε νευρώνας λαμβάνει είσοδο από πολλούς άλλους νευρώνες, εκτελεί κάποιους υπολογισμούς, και παράγει μια έξοδο.

Στρώμα (Layer): Οι νευρώνες οργανώνονται σε στρώματα. Το επίπεδο εισόδου αποτελεί το πρώτο στρώμα, το επίπεδο εξόδου αποτελεί το τελευταίο στρώμα, και τα ενδιάμεσα επίπεδα αποτελούν τα κρυφά (hidden) στρώματα.

Συνάρτηση Ενεργοποίησης (Activation Function): Εφαρμόζεται σε κάθε νευρώνα για να εισάγει μη γραμμικότητα στο δίκτυο. Οι δημοφιλείς συναρτήσεις ενεργοποίησης περιλαμβάνουν την βηματική, Relu, και την σιγμοειδή συνάρτηση ενεργοποίησης.

Βάρη (Weights- Connected Nodes): Οι νευρώνες ενός επιπέδου συνδέονται με τους νευρώνες του επόμενου επιπέδου. Οι συνδέσεις αυτές έχουν βάρη που προσαρμόζονται κατά τη διάρκεια της εκπαίδευσης.

Συναρτήσεις Κόστους (Cost Functions): κατά την εκπαίδευση, η απόδοση του δικτύου αξιολογείται χρησιμοποιώντας μια συνάρτηση κόστους που μετρά την απόκλιση των προβλέψεων από τις πραγματικές τιμές.

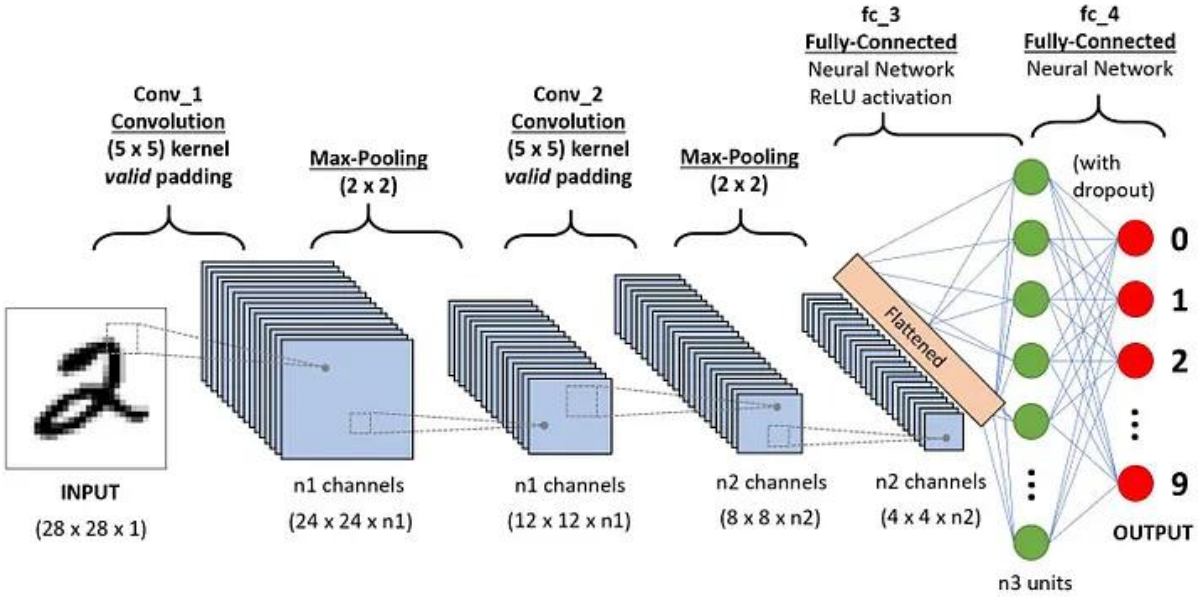
Εκπαίδευση (Training): η διαδικασία κατάρτισης συμπεριλαμβάνει την παροχή εκπαιδευτικών δεδομένων στο δίκτυο, ώστε να προσαρμόσει τα βάρη του για την εκμάθηση συγκεκριμένων προτύπων.

2.3.5.1 Συνελικτικά Νευρωνικά Δίκτυα

Τα συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks - CNNs) είναι ένα είδος νευρωνικών δικτύων που έχουν σχεδιαστεί ειδικά για να επεξεργάζονται δομημένα (structured data) δεδομένα, όπως εικόνες ή βίντεο. Τα CNN έχουν επαναπροσδιορίσει τον τρόπο με τον οποίο αντιλαμβανόμαστε και επεξεργαζόμαστε την πληροφορία σε εικόνες, επιτρέποντας την αυτόματη εξαγωγή χαρακτηριστικών και προτύπων. Ανήκουν στην κατηγορία των τεχνητών νευρωνικών δικτύων. Στην αρχιτεκτονική των CNN, κωδικοποιούνται ειδικά χαρακτηριστικά για την εικόνα καθιστώντας το δίκτυο κατάλληλο για τον επιθυμητό σκοπό, γεγονός που οδηγεί σε μείωση των παραμέτρων που απαιτούνται για την εκπαίδευση του μοντέλου.

Ουσιώδη ρόλο στην αρχιτεκτονική των CNN διαδραματίζει η λειτουργία της συνέλιξης. Η συνέλιξη είναι μια γραμμική λειτουργία που χρησιμοποιείται σε συνελικτικά νευρωνικά δίκτυα για την εξαγωγή χαρακτηριστικών από εικόνες ή άλλα δομημένα δεδομένα. Η διαδικασία αυτή εφαρμόζει ένα φίλτρο (kernel) στην είσοδο (όπως μια εικόνα) και εφαρμόζει αυτό το φίλτρο σε όλο τον χώρο της εισόδου, μετακινώντας το κατά μικρά βήματα (stride). Κάθε εφαρμογή του φίλτρου στην είσοδο παράγει έναν "χάρτη ενεργοποίησης" (activation map) που αντιστοιχεί στον τρόπο που το φίλτρο αντιδρά σε διάφορα τμήματα της εισόδου. Ο στόχος είναι να εξάγουμε χαρακτηριστικά από την είσοδο, όπως γωνίες, ακμές, χρώματα, που μπορούν να αναγνωριστούν στην επεξεργασία της εικόνας. Κάθε φίλτρο έχει συγκεκριμένα βάρη (weights) που ορίζουν τον τρόπο με τον οποίο το φίλτρο αντιδρά σε διάφορα χαρακτηριστικά. Κατά τη διάρκεια της συνέλιξης, το φίλτρο εφαρμόζεται στον πίνακα της εισόδου, και οι πολλαπλές εφαρμογές του παράγουν διάφορα χάρτες ενεργοποίησης, καθένα αποτελώντας μια παραπάνω αφαίρεση χαρακτηριστικών από την είσοδο. [9]

Οι διάφορες τεχνικές της συνέλιξης, όπως η χρήση διάφορων μεγεθών φίλτρων (kernel), τα βήματα μετακίνησης του φίλτρου στην είσοδο (stride), και η προσθήκη zero-padding γύρω από την είσοδο για να διατηρηθεί η διάσταση, μπορούν να επηρεάσουν τον τρόπο με τον οποίο λειτουργεί η συνέλιξη στα διάφορα συνελικτικά επίπεδα του νευρωνικού δικτύου.



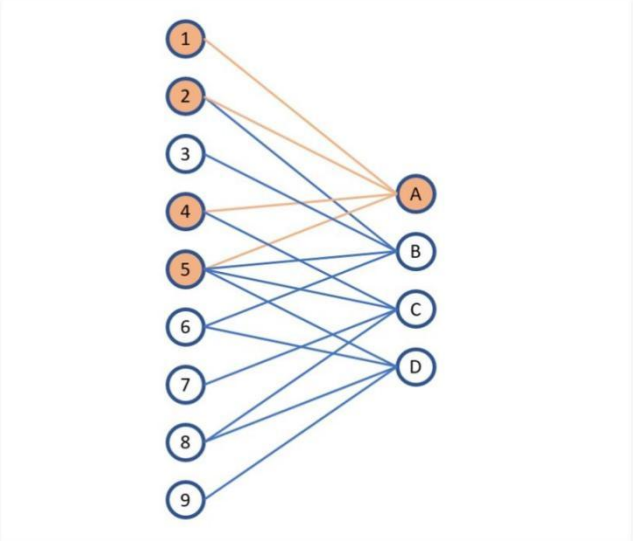
Εικόνα 2.7: Αρχιτεκτονική συνελικτικού δικτύου για αναγνώριση χειρόγραφων αριθμών. [16]

2.3.5.1.1 Αρχιτεκτονική Συνελικτικών Δικτύων

Βασικά στοιχεία της αρχιτεκτονικής τους είναι τα παρακάτω:

Στρώμα Εισόδου (Input Layer): Στο επίπεδο εισόδου, γίνεται είσοδος ολόκληρου του νευρωνικού δικτύου. Αν η είσοδος είναι εικόνα, αυτή μεταφράζεται ως pixels σε μορφή δισδιάστατου πίνακα, με κάθε στοιχείο του πίνακα να αποτελεί ένα διαφορετικό pixel της εικόνας.

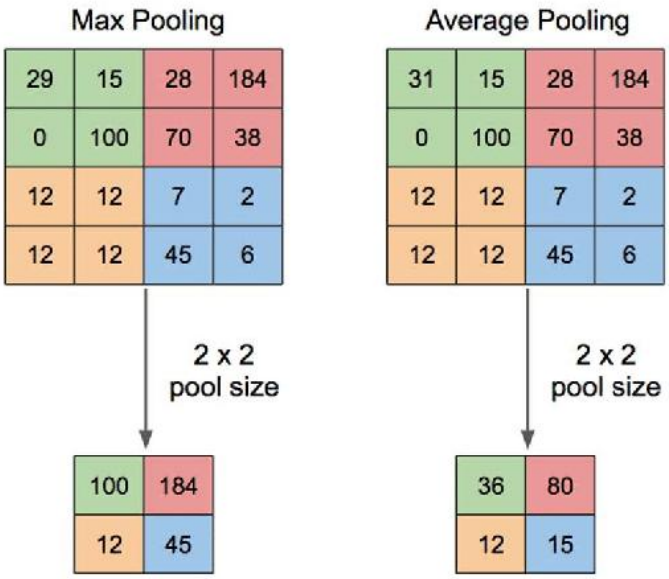
Συνελικτικό Στρώμα (Convolutional Layer): Τα πρώτα επίπεδα είναι συνελικτικά επίπεδα που εφαρμόζουν την συνέλιξη στην εικόνα για την εξαγωγή χαρακτηριστικών. Το επίπεδο συνέλιξης είναι βασικό στοιχείο του νευρωνικού δικτύου, φέρει το μεγαλύτερο υπολογιστικό όγκο και είναι το σημαντικότερο επίπεδο. Η συνέλιξη έχεις ως στόχο να εξάγει διαφορετικά χαρακτηριστικά από εκείνα της εισόδου. Ως γραμμική λειτουργία, η συνέλιξη περιλαμβάνει τον πολλαπλασιασμό βαρών (φίλτρα) με την είσοδο. Τα φίλτρα (kernel), συνδράμουν στην εξαγωγή τοπικών χαρακτηριστικών και αξιοποιούνται για την εξαγωγή τοπικών χαρακτηριστικών που διαμορφώνουν έναν χάρτη χαρακτηριστικών(feature map). Έτσι, όταν τα δεδομένα εισόδου εισέρχονται στο επίπεδο της συνέλιξης, ενεργοποιείται η συνέλιξη μεταξύ της εισόδου και του φίλτρου, βαθμιαία ελαττώνεται το φίλτρο της εισόδου σε πλάτος και σε ύψος, υπολογίζοντας το εσωτερικό γινόμενο μεταξύ του φίλτρου και του αντίστοιχου τμήματος της εικόνας. Ο πολλαπλασιασμός του φίλτρου και μέρος τμήματος της εικόνας δίνει μία τιμή, η οποία αναφέρεται στον συσχετισμό των pixels, εφόσον θωρήσουμε ως δεδομένο πως ως είσοδο έχουμε μια εικόνα. Επιπρόσθετη παράμετρο του επιπέδου συνέλιξης είναι τα stride[17], που είναι ο αριθμός των pixels που σταδιακά μετακινεί το φίλτρο πάνω στην είσοδο.



Εικόνα 2.8: Συνελικτικό επίπεδο ενός συνελικτικού νευρωνικού δικτύου.[55]

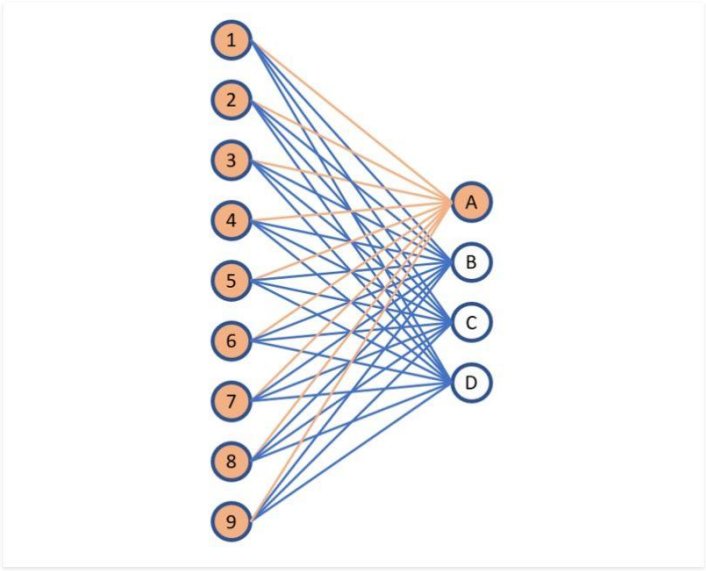
Συνάρτηση Ενεργοποίησης (Activation Function): Η απόφαση για το αν ο νευρώνας (κόμβος) θα ενεργοποιηθεί ή όχι είναι στα καθήκοντα της συνάρτησης ενεργοποίησης. Τα κρυφά επίπεδα του συνελικτικού νευρωνικού δικτύου ενεργοποιούνται μόλις γίνεται χρήση της συνάρτησης ενεργοποίησης.

Στρώματα Υποδειγματοληψίας (Pooling Layers): Το επίπεδο υποδειγματοληψίας τοποθετείται μετά την συνάρτηση ενεργοποίησης και αποτελεί μια κοινή δομή των συνελικτικών νευρωνικών δικτύων που μπορεί να χρησιμοποιηθεί περισσότερες από μία φορές στην αρχιτεκτονική του συνελικτικού. Στο εν λόγω επίπεδο, ένα πρόβλημα που αντιμετωπίζουν οι χάρτες χαρακτηριστικών (feature maps), στη θέση των χαρακτηριστικών εισόδου, είναι η ευαισθησία. Μικρές μεταβολές που πραγματοποιούνται στην θέση του χαρακτηριστικού της εικόνας εισόδου έχουν ως αποτέλεσμα την δημιουργία ενός νέου χάρτη χαρακτηριστικών. Η ευαισθησία που χαρακτηρίζει αυτό το επίπεδο, αντιμετωπίζεται με τεχνικές όπως η υποδειγματοληψία (down sampling). Αυτή η τεχνική, περιλαμβάνει μείωση της ανάλυσης της εικόνας εισόδου, δημιουργώντας ένα πανομοιότυπο της αρχικής, που περιέχει μόνο τα σημαντικά δομικά χαρακτηριστικά της εισόδου ενώ παραβλέπει τα ασήμαντα (δηλαδή αυτά που δεν χρειάζεται το μοντέλο για τον σκοπό που εκπαιδεύεται). [9] Η τεχνική της υποδειγματοληψίας επιτυγχάνεται αλλάζοντας το βήμα συνέλιξης (stride) σε όλη της εικόνα. Οι δύο πιο κοινές τεχνικές που εφαρμόζονται σε αυτό το επίπεδο είναι, όπως φαίνεται και στην Εικόνα 2.5, το **average pooling** και το **max pooling**.



Εικόνα 2.9: Average και Max Pooling συνελικτικών νευρωνικών δικτύων. [56]

Πλήρως Συνδεδεμένο Στρώμα (Fully Connected Layer): Τέλος, το πλήρως συνδεδεμένο επίπεδο, λειτουργεί όπως και οι νευρώνες ενός τεχνητού νευρωνικού δικτύου. Η διαφορά του με το αντίστοιχο των τεχνητών νευρωνικών δικτύων έγκειται στην σχέση του με το προηγούμενο επίπεδο, καθώς λειτουργεί σε πλήρη συσχέτιση με τους νευρώνες του επιπέδου συγκέντρωσης και βοηθάει στην εκμάθηση μη γραμμικών συσχετίσεων που προκύπτουν από τα προηγούμενα συνελικτικά επίπεδα



Εικόνα 2.10: Πλήρως συνδεδεμένο επίπεδο ενός συνελικτικού νευρωνικού δικτύου. [55]

Batch Normalization: Η κανονικοποίηση δέσμης ως τεχνική, χρησιμοποιείται στη βαθιά μάθηση για την κανονικοποίηση των εξόδων των συναρτήσεων ενεργοποίησης στα επίπεδα των νευρωνικών δικτύων. [18] Αυτή η τεχνική στοχεύει στη βελτίωση της απόδοσης και της σταθερότητας των νευρωνικών δικτύων, με έμφαση στην εκπαίδευση στα βαθιά δίκτυα με πολλαπλά επίπεδα. Κατά την εκπαίδευση στα βαθιά δίκτυα, οι ενεργοποιήσεις κάθε στρώματος μπορεί να διαφέρουν σημαντικά σε κάθε επίπεδο, με αποτέλεσμα να δυσχεραίνεται η εκπαίδευσή τους. Η κανονικοποίηση δέσμης

προσφέρει σημαντικά στην ελάττωση του συγκεκριμένου προβλήματος, κανονικοποιώντας τις ενεργοποιήσεις κάθε επιπέδου λαβαίνοντας υπόψη όλη τη δέσμη δεδομένων και κάνοντας χρήση της μέση τιμής και την τυπικής απόκλισης της δέσμης δεδομένων που επεξεργάζεται κάθε φορά. Έπειτα, τα επίπεδα ενεργοποίησης που έχουν κανονικοποιηθεί, υπόκεινται σε κλιμάκωση και μετατόπιση με σκοπό να έχουν μέση τιμή(μ) 0 και τυπική απόκλιση(σ) 1.

2.3.5.1.2 Τομείς Εφαρμογής Συνελικτικών Δικτύων

Τα συνελικτικά δίκτυα έχουν μεγάλο εύρος εφαρμογών λόγω τις ικανότητας τους να αντιλαμβάνονται πολύπλοκα χαρακτηριστικά από δεδομένα που έχουν ιεραρχική φύση, όπως οι εικόνες και τα βίντεο. Παρακάτω είναι μερικές σημαντικές χρήσεις των συνελικτικών δικτύων:

‘Οραση Υπολογιστών (Computer Vision):
- → Κατηγοριοποίηση Εικόνων (Image Classification)
- → Ανίχνευση Αντικειμένων σε Εικόνες (Object Detection)
- → Αναγνώριση Προσώπου (Facial Recognition)
- → Σημασιολογική Κατάτμιση (Semantic Segmenation)

Επεξεργασία φυσικής γλώσσας (Natural Language Processing):
- → Ανάλυση συναισθημάτων (Sentiment Analysis)
- → Ταξινόμηση κειμένου (Text Classification)

2.3.5.2 Αναδρομικά Νευρωνικά Δίκτυα

Τα αναδρομικά δίκτυα προσθέτουν την έννοια της κατάστασης όπου για να παραχθεί μία νέα έξοδος λαμβάνονται υπόψη και οι είσοδοι και έξοδοι προηγούμενης κατάστασης και από εκεί λαμβάνουν και το όνομα τους καθώς ανατρέχουν σε προηγούμενη κατάσταση για να παράγουν μία έξοδο. Αντίθετα με τα παραδοσιακά νευρωνικά δίκτυα όπου οι πληροφορίες κινούνται προς μία κατεύθυνση, από την είσοδο στην έξοδο, τα αναδρομικά νευρωνικά δίκτυα δημιουργούν βρόγχους και διατηρούν μνήμη προηγούμενων καταστάσεων.

Το Long Short-Term Memory (LSTM)[9] είναι ένα είδος αναδρομικού νευρωνικού δικτύου (RNN) που έχει σχεδιαστεί ειδικά για να διαχειρίζεται μακροπρόθεσμες εξαρτήσεις και να διατηρεί μνήμη για μεγάλα χρονικά διαστήματα. Τα LSTM είναι εξαιρετικά χρήσιμα για εφαρμογές που απαιτούν την αντιμετώπιση ή την κατανόηση μακροπρόθεσμων προτύπων ή εξαρτήσεων σε δεδομένα. Η ιδέα πίσω από τα LSTM είναι η χρήση πυλών (gates) για να ελέγξουν τη ροή των πληροφοριών μέσα στο νευρωνικό δίκτυο. Ένα LSTM έχει κάποιους βασικούς πυλώνες που ελέγχουν την είσοδο, την έξοδο και τη μνήμη του. Συγκεκριμένα, τα στοιχεία των LSTM περιλαμβάνουν:

Τα αναδρομικά συνελικτικά νευρωνικά δίκτυα (RCNN)[19-21] (recurrent convolutional neural networks ή RCNNs) αποτελούν μια ειδική κατηγορία νευρωνικών δικτύων που ενσωματώνουν την έννοια της αναδρομής ή της επανάληψης στη διαδικασία της συνέλιξης. Η αρχιτεκτονική των RCNNs συνήθως συνδυάζει στοιχεία από συνελικτικά επίπεδα (Convolutional Layers) και στοιχεία από αναδρομικά επίπεδα (Recurrent Layers) που επιτρέπουν την επεξεργασία αναδρομικής πληροφορίας.

Στα τεχνητά συνελικτικά νευρωνικά δίκτυα (CNNs), η διαδικασία συνέλιξης είναι μια μορφή τοπικής συνάψεως μεταξύ διαφόρων επιπέδων του δικτύου, ενώ η αναδρομή αναφέρεται στην ικανότητα ενός συστήματος να επαναφέρεται στον εαυτό του. Αφού παραχθεί επιτυχώς η έξοδος, αντιγράφεται και αποστέλλεται πίσω στο επαναλαμβανόμενο δίκτυο. Όταν καλείται να πάρει την απόφαση, λαμβάνει υπόψη την τρέχουσα είσοδο και την έξοδο που είχε μάθει με βάση την προηγούμενη είσοδο.

Στην περίπτωση των RCNNs, η αναδρομή χρησιμοποιείται για να δημιουργήσει εσωτερική αναπαράσταση των δεδομένων ή για να επιλύσει προβλήματα με αυθόρμητες δομές όπως οι γράφοι. Ένα παράδειγμα είναι τα συνελικτικά αναδρομικά δίκτυα που χρησιμοποιούνται για την επεξεργασία χειρογράφων. Αυτά τα δίκτυα μπορούν να είναι ιδιαίτερα χρήσιμα για την ανάλυση δεδομένων που διαθέτουν δομική πληροφορία, όπως τα κοινωνικά δίκτυα ή οι μοριακές δομές σε χημεία και βιολογία. Η επέκταση των συνελικτικών νευρωνικών δικτύων για αναδρομικές δομές ανοίγει νέους ορίζοντες στον τρόπο με τον οποίο τα νευρωνικά δίκτυα μπορούν να επεξεργαστούν δεδομένα που έχουν εσωτερικές συνδέσεις ή δομικές σχέσεις.

Προβλήματα αναδρομικών: Παρόλο που τα RNN αποτελούνται από φαινομενικά ισχυρή αρχιτεκτονική αντιμετωπίζουν ένα ουσιώδες πρόβλημα. Η περιορισμένη ικανότητα τους να μοντελοποιούν μακροπρόθεσμες εξαρτήσεις [21] συνήθως οφείλεται στην διαχείριση της μνήμης τους. Για να λυθεί αυτή η δυσκολία εισήχθησαν οι μακροπρόθεσμες μνήμες, Long Short-Term Memory. Αν και τα αναδρομικά επίπεδα των RCNNs διατηρούν κάποια μορφή μνήμης για να διατηρήσουν πληροφορία από προηγούμενα βήματα επεξεργασίας, η δυνατότητά τους για μακροπρόθεσμες εξαρτήσεις είναι περιορισμένη σε σχέση με άλλες αρχιτεκτονικές όπως τα αναδρομικά νευρωνικά δίκτυα μνήμης (RNNs) ή τα μοντέλα που βασίζονται σε μηχανισμούς μακροπρόθεσμης μνήμης όπως τα LSTM (Long Short-Term Memory) και GRU (Gated Recurrent Unit).

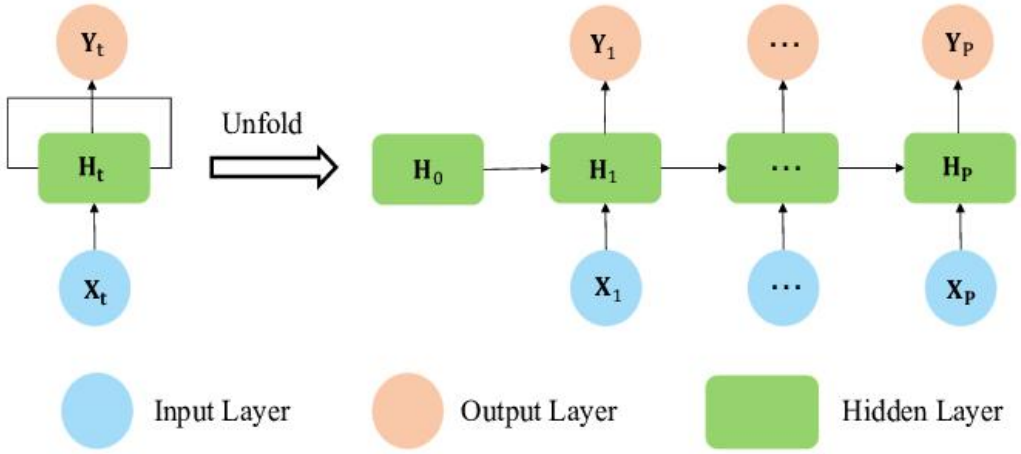
2.3.5.2.1 Αρχιτεκτονική Αναδρομικών Δικτύων

Τα παρακάτω αποτελούν στοιχεία της αρχιτεκτονικής RNN:

Κρυφά επίπεδα: Το κρυφό επίπεδο διατηρεί μια κρυφή κατάσταση που ενημερώνεται σε κάθε χρονικό βήμα. Αυτή η κρυφή κατάσταση περιλαμβάνει πληροφορίες από προηγούμενα χρονικά βήματα και λειτουργεί ως μια μορφή μνήμης.

Αναδρομικές συνδέσεις: Το κύριο χαρακτηριστικό των νευρωνικών δικτύων είναι οι αναδρομικές συνδέσεις που επιτρέπουν στην πληροφορία να περνά από ένα χρονικό βήμα στο επόμενο. Αυτό επιτρέπει στο δίκτυο να καταγράφει σειριακές εξαρτήσεις.

Επίπεδο εξόδου: Το επίπεδο εξόδου παράγει την έξοδο βασισμένο στην τρέχουσα είσοδο και την κρυφή μνήμη κατάσταση. Η έξοδος μπορεί να χρησιμοποιηθεί για προβλέψεις ή για περαιτέρω επεξεργασία.



Εικόνα 2.11: Διάγραμμα αρχιτεκτονικής απλού αναδρομικού δικτύου ανεπτυγμένο στον χρόνο. [57]

Τα παρακάτω αποτελούν στοιχεία της αρχιτεκτονικής LSTM:

Κυρίως Κύτταρο Μνήμης (Cell State): Αυτό είναι το σημαντικό στοιχείο του LSTM και διατηρεί την μακροπρόθεσμη μνήμη.

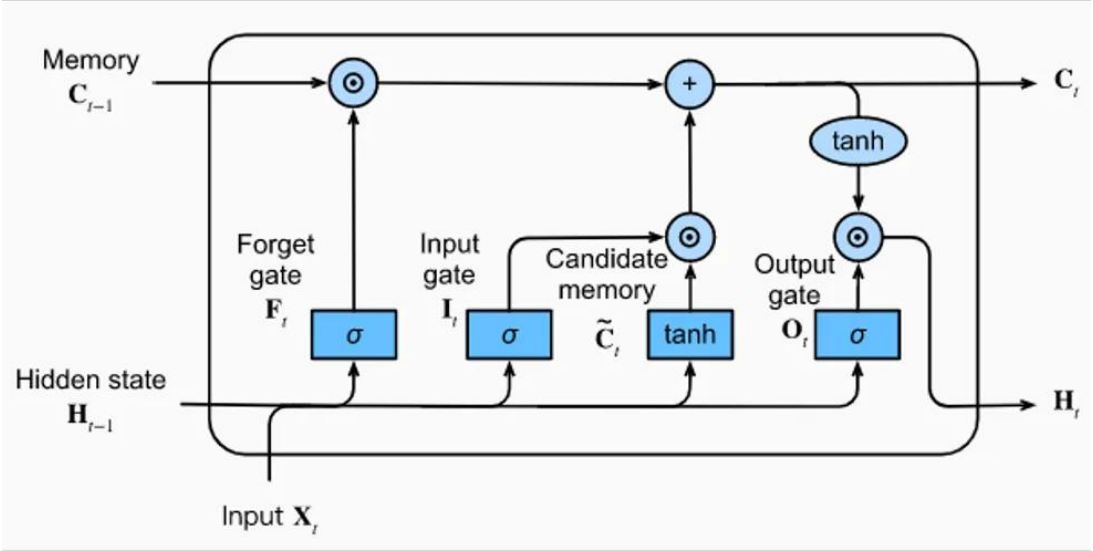
Πύλες (Gates): Τα LSTM χρησιμοποιούν τρεις πύλες για να ελέγξουν τη ροή των πληροφοριών.

→ Forget gate (Πύλη ανατροφοδότησης μνήμης): Η πύλη ξεχνάει ή θυμάται πληροφορίες από τον προηγούμενο κύκλο.

→ Output gate (Πύλη εισόδου): Η πύλη εισόδου αποφασίζει ποιες νέες πληροφορίες θα αποθηκευτούν στο κύτταρο μνήμης.

→ Output gate (Πύλη εξόδου): Η πύλη εξόδου ελέγχει την έξοδο του κυρίως κυττάρου μνήμης με βάση την είσοδο και τη μνήμη.

Η δομή αυτή επιτρέπει στα LSTM να μαθαίνουν και να διατηρούν σημαντικές πληροφορίες για μεγάλα χρονικά διαστήματα, αντιμετωπίζοντας το πρόβλημα της εξάντλησης της μνήμης που αντιμετωπίζουν τα πιο απλά RNNs. Τα LSTM έχουν εφαρμοστεί ευρέως σε πολλούς τομείς, όπως στη φυσική γλώσσα, την αναγνώριση φωνής, την αναγνώριση εικόνας, τη μετάφραση, την ανάλυση χρονοσειρών και άλλα, λόγω της ικανότητάς τους να διατηρούν μακροπρόθεσμες εξαρτήσεις και να μαθαίνουν αποτελεσματικά σύνθετες δομές και πρότυπα στα δεδομένα.

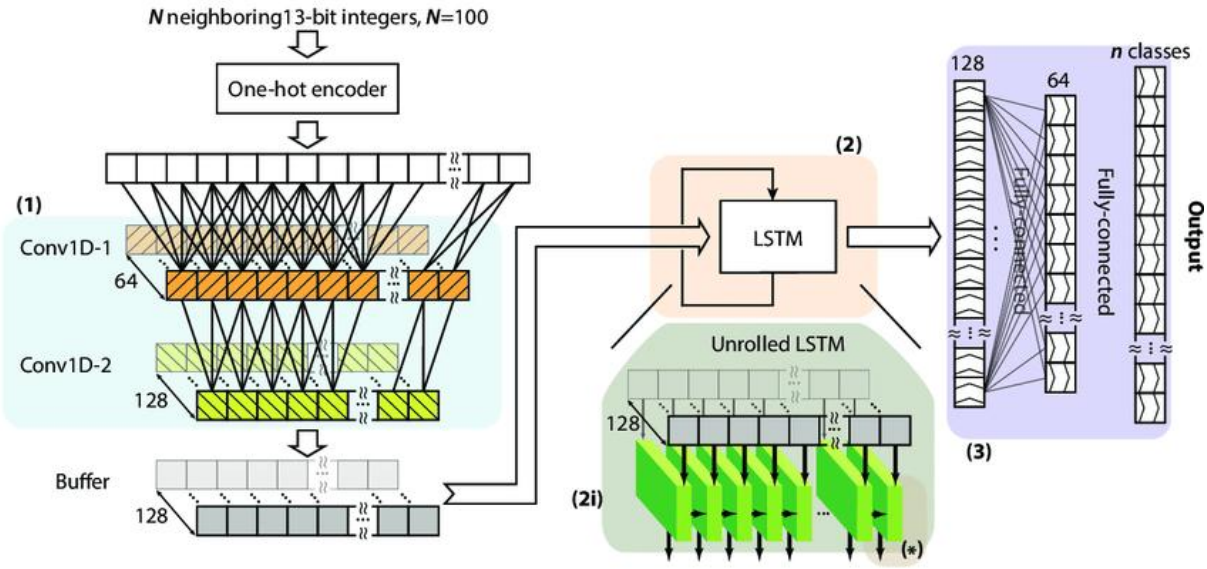


Εικόνα 2.12: Διάγραμμα αρχιτεκτονικής δικτύου LSTM. [58]

Τα παρακάτω αποτελούν στοιχεία της αρχιτεκτονικής RCNN :

Συνελικτικά Επίπεδα (Convolutional Layers): Τα συνελικτικά επίπεδα εφαρμόζουν συνελίξεις στην είσοδο (π.χ., εικόνες) για την εξαγωγή χαρακτηριστικών. Αυτά τα επίπεδα μπορεί να ανιχνεύουν ακμές, γωνίες, σχήματα, ή άλλα χαρακτηριστικά στα δεδομένα.

Αναδρομικά Επίπεδα (Recurrent Layers): Αυτά τα επίπεδα διατηρούν κάποια μορφή μνήμης ή εσωτερικής κατάστασης που επιτρέπει τη ροή της πληροφορίας πίσω στο δίκτυο. Η αναδρομή επιτρέπει την επεξεργασία δεδομένων που έχουν συνδέσεις ή σχέσεις μεταξύ τους. Συνήθως επιλέγονται τα LSTM (Long Short-Term Memory) ή GRU (Gated Recurrent Unit).



Εικόνα 2.13: Διάγραμμα αρχιτεκτονικής δικτύου RCNN με συνελικτικά επίπεδα και LSTM. [59]

Η συνδυασμένη αρχιτεκτονική αυτών των επιπέδων επιτρέπει στα RCNNs να λαμβάνουν υπόψη δομικές πληροφορίες και να επεξεργάζονται δεδομένα που έχουν εσωτερικές σχέσεις ή συνδέσεις

μεταξύ τους. Αυτό είναι ιδιαίτερα χρήσιμο σε περιπτώσεις όπως η αναγνώριση αντικειμένων σε εικόνες με πολλαπλά αντικείμενα που σχετίζονται μεταξύ τους ή σε δεδομένα με δομικές σχέσεις. Κάθε αρχιτεκτονική RCNN μπορεί να διαφέρει ανάλογα με την εφαρμογή και την επιλεγόμενη αναδρομική δομή για την κάθε περίπτωση. Οι ειδικές αρχιτεκτονικές προσαρμόζονται συνήθως στις απαιτήσεις της κάθε εφαρμογής ή των δεδομένων που χρησιμοποιούνται.

2.3.5.2.2 Τομείς Εφαρμογής Αναδρομικών Δικτύων

Τα αναδρομικά επίπεδα χρησιμοποιούνται κυρίως για επεξεργασία φυσικής γλώσσας αλλά και γενικότερα σε ακολουθιακά δεδομένα, καθώς μπορούν να αντιληφθούν συσχετίσεις ανάμεσα σε αυτά και να κάνουν ανάλογες προβλέψεις. Μερικές από τις πιο συνηθισμένες χρήσεις είναι οι παρακάτω:

Επεξεργασία φυσικής γλώσσας (Natural Language Processing):
- → Ανάλυση συναισθημάτων (Sentiment Analysis)
- → Δημιουργία Κειμένου (Text Generation)
- → Αναγνώριση Ονοματικών Οντοτήτων (Named Entity Recognition)

Επεξεργασία Ομιλίας (Speech Processing):
- → Μετατροπή Ομιλίας σε Κείμενο (Speech to Text)
- → Μετατροπή Κείμενου σε Ομιλία (Text to Speech)

Πρόβλεψη χρονοσειρών (Time Series Prediction):
- → Πρόβλεψη καιρού (Weather forecasting)

Επιπλέον συνδυάζοντας και αρχιτεκτονική των συνελικτικών δικτύων τα αναδρομικά δίκτυα επεκτείνονται και στην επεξεργασία εικόνων και βίντεο και μερικές από τις χρήσεις του φαίνονται παρακάτω:

Όραση Υπολογιστών (Computer Vision):
- → Αναγνώριση Χειρονομιών (Gesture Recognition)
- → Εκτίμηση Ανθρώπινης Πόζας (Human Pose Estimation)

2.3.5.3 Transformers

Οι Transformers είναι ένα είδος αρχιτεκτονικής νευρωνικών δικτύων που ξεχωρίζει στην επεξεργασία ακολουθιακών δεδομένων, όπως προτάσεις ή χρονοσειρές. Πρωτοεμφανίστηκαν σε ένα άρθρο του 2017 με τίτλο "Attention is All You Need". [22]

Η κύρια καινοτομία των Transformers είναι ο μηχανισμός Self-Attention. Αντί να επεξεργάζονται την είσοδο σειριακά, όπως στα παραδοσιακά αναδρομικά νευρωνικά δίκτυα (RNNs), οι Transformers μπορούν να λαμβάνουν υπόψη όλες τις λέξεις (ή tokens) σε μια ακολουθία ταυτόχρονα. Αυτό επιτυγχάνεται μέσω του μηχανισμού που προαναφέρθηκε, επιτρέποντας στο μοντέλο να εξάγει συσχετίσεις ακόμα και όταν θέσεις των δεδομένων, όπως οι λέξεις σε μία πρόταση, έχουν μεγάλη απόσταση μεταξύ τους.

Η αρχιτεκτονική του transformer που περιγράφεται στο άρθρο περιλαμβάνει ένα πλήθος κωδικοποιητών (Encoders) και αποκωδικοποιητών (Decoders), οι οποίοι περιλαμβάνουν τον μηχανισμό self-attention με την μορφή του Multi-Head-Attention, γραμμικά fully-connected νευρωνικά δίκτυα και καθένα από αυτά ακολουθείται από ένα επίπεδο πρόσθεσης και εξομάλυνσης (normalization layer). Στην κωδικοποιημένη ακολουθία εισόδου (Input Embedding) προστίθενται κωδικοποιήσεις των θέσεων (Positional Encodings) για να δώσουν πληροφορίες στο μοντέλο για τη σειρά της ακολουθίας.

Οι Transformers έχουν μεγάλη επιτυχία, ιδιαίτερα σε εργασίες επεξεργασίας φυσικής γλώσσας, λόγω της ικανότητάς τους να ανιχνεύουν μακροπρόθεσμες εξαρτήσεις και σχέσεις μέσα σε ακολουθίες. [22] Έχουν χρησιμοποιηθεί σε διάφορα προ-εκπαιδευμένα μοντέλα όπως το BERT (Bidirectional Encoder Representations from Transformers) και το GPT (Generative Pre-trained Transformer), τα οποία έχουν επιτύχει αποτελέσματα υψηλού επιπέδου σε διεργασίες όπως η κατανόηση γλώσσας, η μετάφραση, η περίληψη κειμένου και η απάντηση σε ερωτήσεις. Εκτός από την επεξεργασία φυσικής γλώσσας, οι Transformers έχουν εφαρμοστεί με επιτυχία και σε εργασίες όπως το όραση υπολογιστών και άλλους τομείς, επιδεικνύοντας την ευελιξία και την αποτελεσματικότητά τους.

2.3.5.3.1 Αρχιτεκτονική Transformers

Κύρια χαρακτηριστικά της αρχιτεκτονικής των Transformers είναι:

Self-Attention: Μια συνάρτηση προσοχής μπορεί να περιγράφει ως χαρτογράφηση ενός ερωτήματος (query) και ενός συνόλου ζευγών κλειδιού-τιμής (key-value) σε μια έξοδο, όπου τα ερωτήματα, τα κλειδιά, οι τιμές και οι έξοδοι είναι όλα διανύσματα. Η έξοδος υπολογίζεται ως ένα σταθμισμένο άθροισμα των τιμών, όπου το βάρος (weight) που εκχωρείται σε κάθε τιμή υπολογίζεται από μια συνάρτηση συμβατότητας τους ερωτήματος με το αντίστοιχο κλειδί. [22]

Scaled Dot-Product Attention: Η είσοδος που αποτελείται από διανύσματα ερωτήσεων και κλειδιών διαστάσεων dk, καθώς και τιμές διαστάσεων dv αντίστοιχα. Για κάθε θέση υπολογίζουμε τα εσωτερικά γινόμενα της ερώτησης της συγκεκριμένης θέση με όλα τα διανύσματα των κλειδιών, αυτό βοηθάει στην μέτρηση της σχετικότητας μεταξύ της θέσης αυτής και των υπόλοιπων θέσεων της ακολουθίας. Διαιρούμε το κάθε ένα από το √dk, που είναι ρίζα της διαστατικότητας του διανύσματος των κλειδιών, έτσι μένουν οι τιμές χαμηλές για καλύτερη σταθερότητα, και εφαρμόζουμε μια συνάρτηση softmax για να λάβουμε τα βάρη σε μία κατανομή που τα στοιχεία μαζί αθροίζουν στο 1 και να πολλαπλασιαστούν με τις τιμές. Στην πράξη, υπολογίζουμε για ένα σύνολο ερωτήσεων ταυτόχρονα, συγκεντρωμένων μαζί σε έναν πίνακα Q. Τα κλειδιά και οι τιμές συγκεντρώνονται επίσης σε πίνακες K και V. Υπολογίζουμε τον πίνακα των εξόδων ως:

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

(2.9) [22]

Multi-Head-Attention: Είναι μία επέκταση του self-attention που χρησιμοποιεί πολλαπλές "κεφαλές" προσοχής, επιτρέποντας στο μοντέλο να αναλύει διάφορες πτυχές των δεδομένων παράλληλα. Αντί να

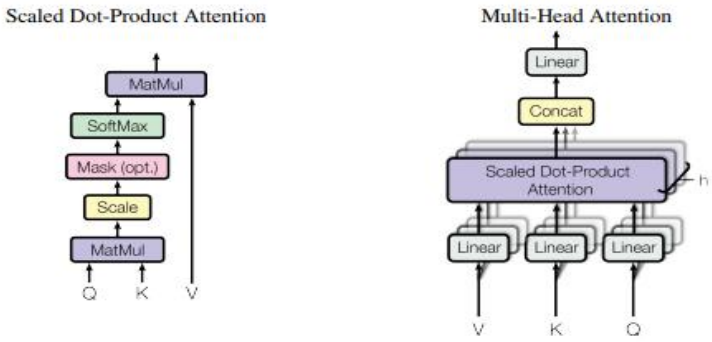
εξαρτάται από ένα σύνολο βαρών ενός self-attention μηχανισμού, χρησιμοποιεί πολλά σύνολα ταυτόχρονα με τα οποία εκτελούνται οι υπολογισμοί και τελικά συμπτύσσονται και μετασχηματίζονται ώστε να παραχθεί το τελικό αποτέλεσμα. Στο Multi-Head-Attention κάθε κεφαλή έχει τους αντίστοιχους πίνακες Q, K , V, και υπολογίζεται ως:

$$\text{MultiHead}(\text{Q}, \text{K}, \text{V}) = \text{Contact}(\text{head}_1, \ldots, \text{head}_\text{h})\text{W}^0$$

(2.10) [22]

Όπου:

- **h** είναι ο αριθμός των κεφαλών.
- **Head*i*** = Attention(*QWQi*, *KWKi*, *VWVi*) είναι η έξοδος της i-οστής κεφαλής.
- ***WQi*** ,***WKi***, ***WVi*** , είναι οι πίνακες των βαρών των αντίστοιχων πινάκων Q, K, V.
- ***WO***, είναι ο πίνακας βαρών για το τελικό στάδιο του γραμμικού μετασχηματισμού.



Εικόνα 2.14: Αρχιτεκτονική scaled dot product attention αριστερά και Multi-Head Attention δεξιά. [22]

Feed-Forward Neural Network: Το feed-forward neural network (FFN) αποτελεί ένα βασικό συστατικό εντός του μηχανισμού self-attention. Το FFN εφαρμόζεται ανεξάρτητα σε κάθε θέση της ακολουθίας, εισάγοντας μη γραμμικότητα στο μοντέλο και επιτρέποντας του να αντλήσει πιο περίπλοκα χαρακτηριστικά από τις αναπαραστάσεις που προκύπτουν από το στάδιο self-attention. Αυτό ενισχύει την εκφραστική δύναμη του μοντέλου, επιτρέποντάς του να μάθει πιο περίπλοκα χαρακτηριστικά από τις αυτόματες αναπαραστάσεις. Συνήθως ως συνάρτηση ενεργοποίηση χρησιμοποιείται η Rectified Linear Unit (ReLU). Η ReLU εισάγει μη γραμμικότητα στο μοντέλο, επιτρέποντας του να μάθει πιο πολύπλοκες σχέσεις.

$$\text{ReLU}(\text{Linear1}(\text{input}))$$

(2.11) [22]

Layer Normalization και Residual Connections: Το αποτέλεσμα του νευρωνικού δικτύου feedforward συχνά κανονικοποιείται χρησιμοποιώντας κανονικοποίηση επιπέδου (Layer Normalization). Επιπλέον για κάθε επίπεδο του Transformer, η είσοδος του επιπέδου (πριν την εφαρμογή των επεξεργασμένων αναπαραστάσεων) προστίθεται στην έξοδο του επιπέδου. Ο συνδυασμός αυτών των τεχνικών χρησιμοποιείται για να εξασφαλιστεί τη σταθερότητα της

εκπαίδευσης και τη ροή των πληροφοριών μέσα στο δίκτυο και να βελτιστοποιηθεί η απόδοση του μοντέλου.

Για μία ενεργοποίηση x η κανονικοποίηση υπολογίζεται ως:

$$\text{LayerNorm}(x) = \frac{\text{x} - \text{mean}(x)}{\sqrt{\text{var}(x) + \epsilon}} \cdot \gamma + \beta$$

(2.12) [22]

Όπου:

- **mean(x)** είναι η μέση τιμή των ενεργοποιήσεων.
- **var(x)** είναι η τυπική απόκλιση των ενεργοποιήσεων.
- **ϵ** είναι ένα πολύ μικρό θετικό αριθμητικό για να αποφευχθεί το διαίρεση με μηδέν.
- **γ** είναι μία μεταβλητή εκμάθησης.
- **β** είναι μία μεταβλητή εκμάθησης.

Positional Encoding: Προσθέτει πληροφορίες σχετικά με τη θέση των λέξεων στην ακολουθία, καθώς τα μοντέλα Transformers δεν διαθέτουν ενσωματωμένη κατανόηση της σειράς των δεδομένων. Καθώς το μοντέλο δεν περιλαμβάνει αναδρομή ή συνέλιξη, προκειμένου να εκμεταλλευτεί τη σειρά της ακολουθίας, πρέπει να ενσωματωθούν κάποιες πληροφορίες σχετικά με τη σχετική ή απόλυτη θέση των tokens στην ακολουθία. Για τον σκοπό αυτό, προσθέτουμε αυτές τις κωδικοποιήσεις των θέσεων (positional encodings) στις ενσωματώσεις εισόδου (input embeddings) και δημιουργούνται τα τελικά διανύσματα εισόδου του encoder και decoder. Οι κωδικοποιήσεις των θέσεων έχουν την ίδια διάσταση dmodel με τις ενσωματώσεις, ώστε οι δύο να μπορούν να αθροιστούν. Υπάρχουν πολλές επιλογές για τις κωδικοποιήσεις των θέσεων, είτε εκπαιδευόμενες είτε σταθερές.

Στο μοντέλο που εξετάζουμε υπολογίζονται ως:

$$\text{PE}(\text{pos}, 2\text{i}) = \sin\left(\frac{\text{pos}}{100000^{2\text{i}/\text{d}}\text{model}}\right)$$

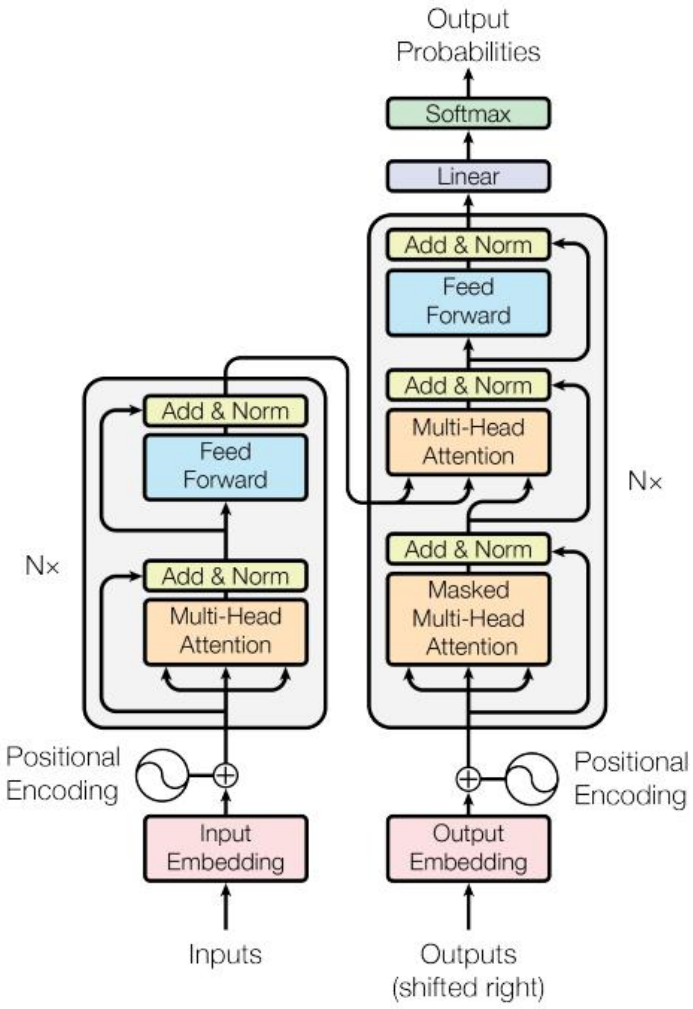
$$\text{PE}(\text{pos}, 2\text{i} + 1) = \cos\left(\frac{\text{pos}}{100000^{2\text{i}/\text{d}}\text{model}}\right)$$

(2.13) [22]

Όπου:

- ***PE(pos,2i)*** *και* ***PE(pos,2i+1)*** είναι τα στοιχεία σε άρτιες και περιττές θέσεις αντίστοιχα, για το διάνυσμα κωδικοποίησης θέσης που αντιστοιχεί στη θέση.
- ***pos*** είναι η θέση του τόκεν στην ακολουθία.
- ***i*** είναι ο δείκτης του στοιχείου κωδικοποίησης θέσης.
- ***dmodel*** είναι η διάσταση του μοντέλου.

Η χρήση συναρτήσεων sine και cosine με μεταβαλλόμενες συχνότητες βοηθούν το μοντέλο να αναλάβει διάφορες θέσεις. Η χρήση του όρου 1100002*i*/*d*model εξασφαλίζει ότι κάθε θέση λαμβάνει μία μοναδική κωδικοποίηση.



Εικόνα 2.15: Αρχιτεκτονική Transformer. [22]

Η δομή του Encoder και του Decoder μοιάζουν αρκετά καθώς αποτελούνται από τα ίδια υποεπίπεδα με διαφορά την προσθήκη ενός ακόμα Multi-Head-Attention. Το συγκεκριμένο Multi-Head-Attention επίπεδο δέχεται ως είσοδο τα δεδομένα εξόδου της στοίβας των Encoders και κάνει την κατάλληλη επεξεργασία σε αυτά μαζί με την έξοδο από ένα Masked-Multi-Head-Attention στο οποίο έχει προστεθεί μία μάσκα σε επόμενες θέσεις της ακολουθίας από την θέση που εξετάζεται κάθε φορά ώστε να αποφευχθούν οι συνδέσεις με μεταγενέστερες θέσεις στην ακολουθία. Με άλλα λόγια, διασφαλίζεται ότι οι προβλέψεις για τη θέση i μπορούν να εξαρτώνται μόνο από θέσεις μόνο μικρότερες του i.

2.3.5.3.2 Τομείς Εφαρμογής Transformers

Οι υλοποιήσεις της αρχιτεκτονικής των transformers διευρύνονται σε ολοένα και περισσότερους τομείς προβλημάτων. Η βασικός τομέας στον οποίο διακρίθηκαν οι transformers είναι η επεξεργασία φυσικής γλώσσας με μοντέλα όπως το BERT (Bidirectional Encoder Representations from Transformers) και το GPT (Generative Pre-trained Transformer) αλλά και πολλά άλλα όπου με

βελτιώσεις στην αρχιτεκτονική και έξυπνες προσθήκες έδειξαν αμέσως τις δυνατότητες αυτού του μηχανισμού. Έτσι τώρα χρησιμοποιούνται σε μια πληθώρα τύπων προβλημάτων όπως:

Επεξεργασία φυσικής γλώσσας (Natural Language Processing):
- → Ανάλυση συναισθημάτων (Sentiment Analysis)
- → Αναγνώριση Ονοματικών Οντοτήτων (Named Entity Recognition)
- → Απάντηση σε Ερωτήσεις (Question Answering)
- → Δημιουργία Περιεχομένου (Content Generation)
- → Πράκτορες Συνομιλίας (Conversational Agents)

Όραση Υπολογιστών (Computer Vision):
- → Κατηγοριοποίηση Εικόνων (Image Classification)
- → Ανίχνευση Αντικειμένων σε Εικόνες (Object Detection)

Επεξεργασία Ομιλίας (Speech Processing):
- → Φωνητικοί Βοηθοί (Voice Assistants)
- → Μετατροπή Ομιλίας σε Κείμενο (Speech to Text)

Πολυτροπικές Εφαρμογές (Multi- modal Applications):
- → Περιγραφή Εικόνων (Image Captioning)
- → Απάντηση σε Ερωτήσεις βάση Εικόνας (Visual Question Answering)

Αυτά είναι μερικά από τα προβλήματα που λύνουν οι transformers αλλά και πολλά άλλα, δείχνοντας την αποτελεσματικότητα τους στην αντίληψη πολύπλοκων προτύπων και εξαρτήσεων σε ακολουθιακά, χωρικά και πολυτροπικά δεδομένα.

Κεφάλαιο 3ο: Ανίχνευση Αντικειμένων (Object Detection)

3.1 Εισαγωγή

Η ανίχνευση αντικειμένων αποτελεί μία σημαντική εργασία στον τομέα της όρασης υπολογιστών, η οποία επικεντρώνεται στον εντοπισμό και την κατηγοριοποίηση αντικειμένων εντός εικόνων ή βίντεο. Τα αντικείμενα συνήθως συσχετίζονται με συγκεκριμένες κλάσεις ή κατηγορίες. Οι αλγόριθμοι ανίχνευσης όχι μόνο εντοπίζουν αντικείμενα, αλλά τους αναθέτουν ετικέτα κλάσης. Έχει εφαρμογές σε πολλούς τομείς, όπως η αυτόνομη οδήγηση και η ασφάλεια. Η ανίχνευση αντικειμένων αντιμετωπίζει προκλήσεις όπως η διαχείριση μεταβλητών μεγεθών αντικειμένων, η αντιμετώπιση της κάλυψης (occlusion) αντικειμένων, οι αλλαγές του γωνιακού προσανατολισμού, και η αντιμετώπιση ενός μεγάλου αριθμού κατηγοριών αντικειμένων.

Η ανίχνευση αντικειμένων στην ουσία περιλαμβάνει δύο προβλήματα, την δημιουργία ενός πλαισίου περιορισμού εντός της εικόνας που περιέχει το αντικείμενο προς ανίχνευση προβλέποντας θετικές συντεταγμένες, δηλαδή ένα πρόβλημα παλινδρόμησης, αλλά και την κατηγοριοποίηση του αντικειμένου που περιέχεται στο συγκεκριμένο πλαίσιο, δηλαδή ένα πρόβλημα ταξινόμησης.

Τα σύγχρονα μοντέλα ανίχνευσης αντιμετωπίζουν αυτό το πρόβλημα χρησιμοποιώντας δύο διαφορετικές προσεγγίσεις:

Ανιχνευτές Δύο Σταδίων (Two-Stage Detectors): Αυτή η κατηγορία ανιχνευτών περιλαμβάνει ένα στάδιο πρότασης περιοχών, το οποίο παράγει υποψήφιες περιοχές που πιθανώς περιέχουν αντικείμενα, και ένα στάδιο εκτίμησης που εκτελεί την τελική ανίχνευση. Παραδείγματα αποτελούν το Faster R-CNN (Faster Region-based Convolutional Neural Network) [23], που χρησιμοποιεί ένα δίκτυο πρότασης RPN (Region Proposal Network) το οποίο προτείνει πρώτα περιοχές που πιθανώς περιέχουν αντικείμενα και στη συνέχεια τα ταξινομεί και τα προσαρμόζει, και το R-FCN (Region-based Fully Convolutional Network) [24] που χρησιμοποιεί position-sensitive score maps για αποτελεσματική ανίχνευση.

Ανιχνευτές Ενός Σταδίου (One-Stage Detectors): Η συγκεκριμένη κατηγορία ανιχνευτών προσφέρει εναλλακτικές λύσεις χωρίς τη χρήση προκαθορισμένων αγκυρών, όπως οι μέθοδοι CenterNet [25] και FCOS [26]. Αυτοί οι ανιχνευτές προβλέπουν απευθείας τις θέσεις και τις κλάσεις των αντικειμένων χωρίς πρόταση περιοχών. Είναι συνήθως ταχύτεροι αλλά μπορεί να έχουν λιγότερη ακρίβεια. Παραδείγματα αποτελούν το DETR (DEtection Transformer) [27], που χρησιμοποιεί transformer για να επιτύχει ενιαία ανίχνευση χωρίς πρόταση περιοχών, το YOLO (You Only Look Once) [28] το οποίο προβλέπει άμεσα τις θέσεις και τις πιθανότητες κλάσης σε μία διέλευση, και το SSD (Single Shot MultiBox Detector) [29] που προβλέπει πολλαπλά πλαίσια περιορισμού και σκορ κλάσης σε διάφορες κλίμακες.

3.2 Μετρικές Αξιολόγησης

Παρακάτω περιλαμβάνονται μετρικές αξιολόγησης μοντέλων object detection:

Ευστοχία και Ανάκληση (Precision-Recall): Οι μετρικές αυτές καταγράφουν την ακρίβεια των θετικών προβλέψεων και την ικανότητα εντοπισμού όλων των θετικών περιπτώσεων.

mAP (Μέση Ευστοχία): Η mAP παρέχει μια ολοκληρωμένη εικόνα της απόδοσης του αλγορίθμου για διάφορες κατηγορίες. Υπολογίζεται ως ο μέσος όρος των τιμών ακρίβειας για κάθε κατηγορία σε ένα εύρος θετικών κατωφλιών πιθανοτήτων.

F1 Score: Το F1 Score χρησιμοποιείται όταν υπάρχει ανάγκη ισορροπίας μεταξύ ευστοχίας και ανάκλησης.

Επικάλυψη (IoU - Intersection over Union): Το IoU μετρά την επικάλυψη μεταξύ των προβλεπόμενων πλαισίων περιορισμού και των πραγματικών. Αποτελεί κρίσιμη μετρική για την αξιολόγηση της ακρίβειας της ανίχνευσης.

Καμπύλη Ευστοχία-Ανάκλησης: Η μέτρηση της απόδοσης γίνεται συνήθως με τη χρήση της καμπύλης ευστοχίας-ανάκλησης, που αναπαριστά τη σχέση μεταξύ της ευστοχίας και της ανάκλησης για διάφορα θετικά κατώφλια πιθανοτήτων.

3.3 DETR (DEtection TRansformer) και Deformable Detr

Οι transformer έχουν διακριθεί στην επεξεργασία φυσικής γλώσσας και έχουν προσαρμοστεί για εργασίες όπως η ανίχνευση αντικειμένων. Το μοντέλο DETR (DEtection TRansformer) αποτελεί μια πρωτοποριακή αρχιτεκτονική για τον τομέα αυτόν, σχεδιασμένη από το Facebook AI Research (FAIR). Οι κύριες λειτουργίες και τα συστατικά του μοντέλου DETR περιλαμβάνουν:

Συνελικτικό Νευρωνικό δίκτυο: Χρησιμοποιεί ένα τυπικό προεκπαιδευμένο συνελικτικό δίκτυο, (στην πιο απλή εκδοχή είναι ένα Resnet50), για την εξαγωγή ενός συμπιεσμένου χάρτη χαρακτηριστικών το οποίο τροφοδοτείται στον encoder αφού επαυξηθεί με τα εκπαιδευόμενα positional encodings, στην προκειμένη περίπτωση Object Queries.

Δομή Encoder-Decoder: Το DETR χρησιμοποιεί μια αρχιτεκτονική transformer, αποτελούμενη από έναν κωδικοποιητή και έναν αποκωδικοποιητή.

Πολυκεφαλική Αυτο-Προσοχή (Multi-Head Self-Attention): Τόσο ο κωδικοποιητής και όσο και ο αποκωδικοποιητής αποτελούνται από επίπεδα Multi-Head Self-Attention. Αυτό επιτρέπει στο μοντέλο να ανιχνεύει σχέσεις μεταξύ διαφορετικών στοιχείων στην είσοδο και είναι κρίσιμο για την αντιμετώπιση εξαρτήσεων μεγάλης εμβέλειας.

Νευρωνικά Δίκτυα Προώθησης (Feed-Forward-FFN): Σε κάθε επίπεδο του transformer, χρησιμοποιούνται νευρωνικά δίκτυα προώθησης για τον μη γραμμικό μετασχηματισμό των επιπέδων self-attention.

Μηχανισμός Self-Attention: Ο μηχανισμός αυτο-προσοχής του Transformer επιτρέπει στο μοντέλο να ζυγίζει τη σημασία διαφορετικών τμημάτων της εισόδου κατά την πρόβλεψη. Αυτό είναι κρίσιμο για την απόκτηση μακροπρόθεσμων εξαρτήσεων στις εικόνες.

Κωδικοποιήσεις Θέσης (Positional Encodings):_ Για να προσφέρει χωρικές πληροφορίες στον transformer, το DETR χρησιμοποιεί κωδικοποιήσεις θέσης. Αυτές προστίθενται στα διανύσματα εισόδου για να μεταδίδουν τις χωρικές σχέσεις μεταξύ των pixel στην εικόνα. Αντί να χρησιμοποιεί πλαίσια αγκυρών (anchor boxes) ή ένα δίκτυο πρότασης περιοχών (region proposal network - RPN), το DETR χρησιμοποιεί εκπαιδευόμενα ερωτήματα αντικειμένων. Αυτά τα Object Queries αντιπροσωπεύουν πιθανές περιοχές αντικειμένων και χρησιμοποιούνται για να επικεντρώνουν το ενδιαφέρον σε διάφορα μέρη της εικόνας.

Object Queries: Ο κωδικοποιητής επεξεργάζεται την είσοδο της εικόνας ως ένα σύνολο χαρτών χαρακτηριστικών. Το DETR εισάγει εκπαιδευόμενα ερωτήματα αντικειμένων (Object Queries) που αντιπροσωπεύουν δυνητικές τοποθεσίες αντικειμένων στην εικόνα.

Key, Value Embeddings: Ο κωδικοποιητής δημιουργεί ενσωματώσεις key και value για κάθε θέση στους χάρτες χαρακτηριστικών. Αυτές χρησιμοποιούνται στον μηχανισμό self-attention, επιτρέποντας στο μοντέλο να επικεντρωθεί περισσότερο στις πιο σχετικές πληροφορίες.

Διαδικασία Πρόβλεψης: Η διαδικασία πρόβλεψης γίνεται με την επαναληπτική ενημέρωση των ερωτημάτων αντικειμένων και των επιπέδων αυτο-προσοχής μέχρις ότου επιτευχθεί η σύγκλιση.

Κεφαλές ταξινόμησης και παλινδρόμησης (Classification και Regression Heads):

→ **Κεφαλίδα Κατηγοριοποίησης:** Αυτή η κεφαλίδα προβλέπει τις ετικέτες κλάσης για κάθε πιθανό Object Query.
→ **Κεφαλίδα Παλινδρόμησης Πλαισίου:** Προβλέπει τις συντεταγμένες του πλαισίου (π.χ., πάνω αριστερή γωνία και κάτω δεξιά γωνία) για κάθε Object Query.

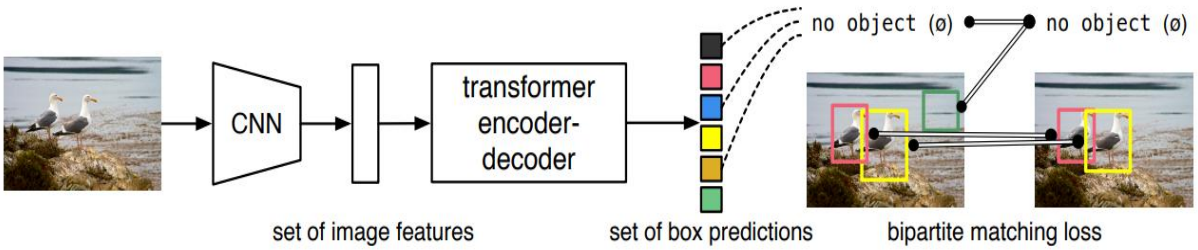
Συναρτήσεις Κόστους (Loss Function):

→ **Συνάρτηση Κόστους Κατηγοριοποίησης:**
Περιγραφή: Αυτή η συνάρτηση απώλειας εφαρμόζεται στις προβλεπόμενες πιθανότητες κλάσης για κάθε ερώτημα αντικειμένου.
Σκοπός: Η συνάρτηση κόστους Cross-Entropy μετρά τη διαφορά μεταξύ των προβλεπόμενων πιθανοτήτων κλάσης και των πραγματικών ετικετών κλάσης. Κινητοποιεί το μοντέλο να κατηγοριοποιεί σωστά τα αντικείμενα.

→ **Συνάρτηση Κόστους Παλινδρόμησης Πλαισίου Περιορισμού:**
Περιγραφή: Αυτή η συνάρτηση κόστους εφαρμόζεται στις προβλεπόμενες συντεταγμένες του πλαισίου περιορισμού (π.χ., πάνω αριστερή γωνία και κάτω δεξιά γωνία) για κάθε ερώτημα αντικειμένου.
Σκοπός: Η συνάρτηση κόστους Smooth L1 (ή Huber) σε συνδυασμό με την GIoU (Generalized Intersection over Union) για τον υπολογισμό της ομοιότητας μεταξύ των προβλεπόμενων και πραγματικών πλαισίων.

Διμερές Κόστος Αντιστοίχισης (Bipartite Matching Loss):

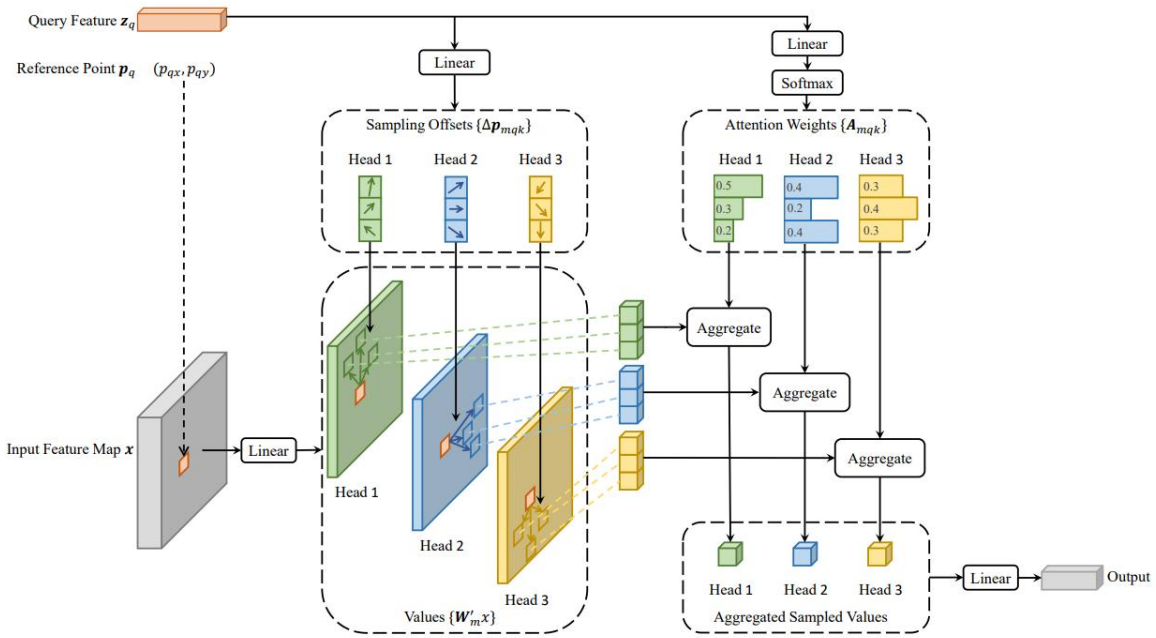
Περιγραφή: Το DETR χρησιμοποιεί έναν αλγόριθμο διμερούς αντιστοίχισης που βασίζεται στον Ουγγρικό αλγόριθμο για να συσχετίσει προβλεπόμενα πλαίσια με τα πραγματικά. Αυτό βοηθά στην εκπαίδευση του μοντέλου για κατηγοριοποίηση και παλινδρόμηση.
Σκοπός: Συνδυάζει τις άλλες δύο συναρτήσεις κόστους και εξασφαλίζει ότι κάθε προβλεπόμενο πλαίσιο συσχετίζεται με το πιο κατάλληλο πραγματικό πλαίσιο, ενθαρρύνοντας την εκπαίδευση των καθηκόντων ταξινόμησης και παλινδρόμησης.



Εικόνα 3.1: Διαδικασία ανίχνευσης αντικειμένων του DETR. [27]

Με τη χρήση των παραπάνω συστατικών, το DETR επιτυγχάνει την ανίχνευση αντικειμένων χωρίς την ανάγκη προκαθορισμένων πλαισίων αγκυρών ή άλλων προηγμένων μεθόδων πρότασης περιοχών. Όμως η αρχιτεκτονική του υποφέρει από μεγάλη πολυπλοκότητα το οποίο οδηγεί σε μεγάλες περιόδους εκπαίδευσης για να επιτευχθεί σύγκλιση και μειωμένη απόδοση σε μικρά αντικείμενα, τα οποία έρχεται να διορθώσει το Deformable DETR. [30]

Το Deformable DETR στοχεύει στην αντιμετώπιση της αργής σύγκλισης και της υψηλής πολυπλοκότητας που παρουσιάζει το DETR. Συνδυάζει αραιή χωρική δειγματοληψία από deformable convolution με τη δυνατότητα μοντελοποίησης σχέσεων των μοντέλων που βασίζονται σε Transformers. Εισαγάγει ένα πρότυπο προσοχής με εύκαμπτη λειτουργία, το οποίο επικεντρώνεται σε ένα μικρό σύνολο τοποθεσιών δειγματοληψίας για προ-φιλτράρισμα κυρίων στοιχείων από όλα τα εικονοστοιχεία του χάρτη χαρακτηριστικών. Χρησιμοποιεί (πολλαπλής κλίμακας) πρότυπα προσοχής με εύκαμπτη λειτουργία αντί των προσοχών του Transformer για την επεξεργασία των χαρτών χαρακτηριστικών. [30]



Εικόνα 3.2: Αρχιτεκτονική του deformable attetion module. [30]

Multi-scale Deformable Attention Module: Το προτεινόμενο προσαρμοστικό πρότυπο προσοχής (deformable attention module) για πολλαπλές κλίμακες χαρτών χαρακτηριστικών.

$$MSDeformAttn(z_p, \hat{p}_q, \{x^1\}_{i=1}^{L=1} = \sum_{m=1}^{M} W_m \left[\sum_{l=1}^{L}\sum_{k=1}^{K} A_{mlqk} \cdot W'_m x^l(\varphi_l(\hat{p}_q) + \Delta p_{mlqk})\right]$$

(2.14) [30]

Όπου:

- το **m** υποδεικνύει το πλήθος των attention head
- και το **l** υποδεικνύει επίπεδο εισόδου του δείγματος
- ενώ, το **k** υποδεικνύει το σημείο δειγματοληψίας
- $\Delta \boldsymbol{p_{mlqk}}$ και $\boldsymbol{A_{mlqk}}$ υποδεικνύουν την απόκλιση με την οποία λαμβάνονται δείγματα και attention weight του k^{th} σημείου δειγματοληψίας στο l^{th} επίπεδο δείγματος και το m^{th} attention head.
- Το attention weight $\boldsymbol{A_{mlqk}}$ είναι κανονικοποιημένο με $\sum_{l=1}^{L}\sum_{k=1}^{K} A_{mlqk}=1$.

Χρησιμοποιούνται η κανονικοποιημένες συντεταγμένες $\hat{\boldsymbol{p}}_{\boldsymbol{q}} \in [0,1]^2$, στις οποίες η κανονικοποιημένες συντεταγμένες (0,0) και (1,1) αναφέρονται στις γωνιές πάνω-αριστερά και κάτω-δεξιά της εικόνας.

- Η συνάρτηση $\boldsymbol{\varphi_1(\hat{p}_q)}$ επανακλιμακώνει τις κανονικοποιημένες συντεταγμένες $\hat{\boldsymbol{p}}_{\boldsymbol{q}}$ στον χάρτη χαρακτηριστικών της εισόδου l-th επιπέδου.

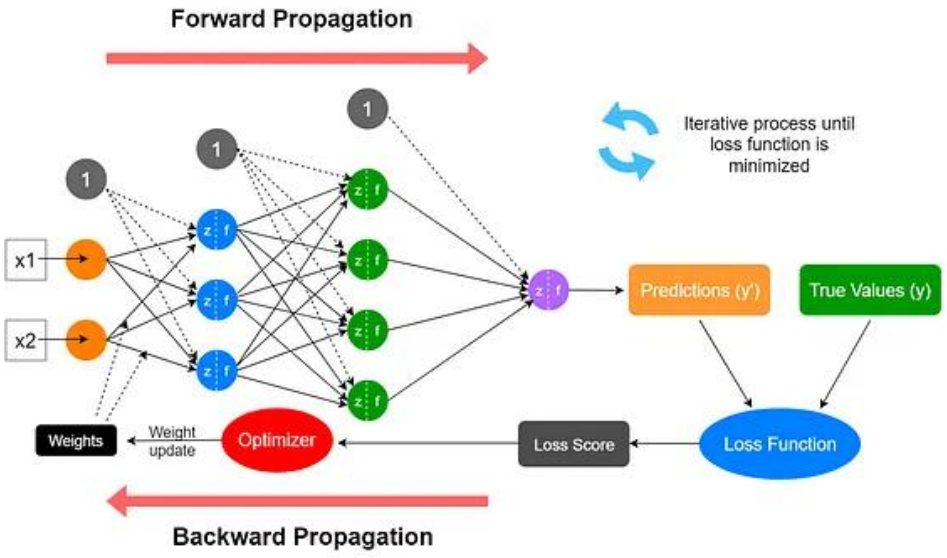
Κεφάλαιο 4ο: Εκπαίδευση Νευρωνικών Δικτύων

4.1 Εισαγωγή

Η εκπαίδευση των μοντέλων με τα οποία ασχολούμαστε ανήκουν στην γενικότερη κατηγορία της εκπαίδευσης με επίβλεψη και ακολουθούν την ροή που φαίνεται στην εικόνα 3.1 σε μία απλοποιημένη μορφή, κάθε δίκτυο ανάλογα την πολυπλοκότητα του μπορούν να διαφέρουν επιμέρους στοιχεία, για παράδειγμα πολλαπλές συναρτήσεις κόστους όπως στο deformable detr. [30]

Σε μία απλοποιημένη μορφή η διαδικασία που ακολουθείται είναι η παρακάτω [9], όπως φαίνεται και στην εικόνα 3.1:

1. Είσοδος ενός διανύσματος.
2. Εμπρόσθια τροφοδότηση υπολογίζοντας τα διανύσματα εξόδου και τις παραγώγους των συναρτήσεων ενεργοποίησης.
3. Σύγκριση εξόδου με στόχους και υπολογισμός σφάλματος με την συνάρτηση κόστους.
4. Πίσω διάδοση από την έξοδο μέχρι την είσοδο και υπολογισμός σφάλματος και μερικής παραγώγου για όλα τα ενδιάμεσα στρώματα.
5. Ενημέρωση βαρών στην κατεύθυνση αρνητικής κλίσης χρησιμοποιώντας ένα βήμα εκπαίδευσης ή ρυθμός μάθησης.



Εικόνα 4.1: Ροή εκπαίδευσης ενός απλού νευρωνικού δικτύου. [31]

4.2 Αρχικοποίηση Βαρών

Ξεκινώντας ένα νευρωνικό δίκτυο δεν γνωρίζει καμία συσχέτιση για τα δεδομένα εκπαίδευσης και με κατάλληλες προσαρμογές στα βάρη, μέσα από την διαδικασία της εκπαίδευσης, λύνει το εκάστοτε πρόβλημα για το οποίο εκπαιδεύτηκε. Αρχικό βήμα του αλγορίθμου της μάθησης είναι η αρχικοποίηση των βαρών, συνήθως με τυχαίες μικρές τιμές, προχωρώντας ο αλγόριθμος επαναληπτικά κάνει της απαραίτητες αυξομειώσεις στις τιμές αυτές με βάση κάποιες παραμέτρους, όπως τα αποτελέσματα των συναρτήσεων σφάλματος και των ρυθμό μάθησης(learning rate). Επιλέγονται μικρές τιμές για να υπάρχει ικανοποιητικό περιθώριο μεταβολής των βαρών, δηλαδή να μην φτάνει η έξοδος των νευρώνων κοντά στο όριο της εκάστοτε συνάρτησης ενεργοποίησης f . [9]

4.3 Συναρτήσεις σφάλματος

Η συνάρτηση σφάλματος ή συνάρτηση κόστους είναι ένα εργαλείο αξιολόγησης της απόδοσης του νευρωνικού δικτύου. Η αξιολόγηση πραγματοποιείται υπολογίζοντας την διαφορά μεταξύ των προβλέψεων του δικτύου και των πραγματικών εξόδων που είναι γνωστές μέσω των ετικετών. Η συνάρτηση σφάλματος έχει σκοπό ελαχιστοποιήσει αυτή την διαφορά. Επομένως, μικρές τιμές στη συνάρτηση σφάλματος, σημαίνει πως η διαφορά των αποτελεσμάτων του δικτύου με τα πραγματικά είναι πολύ μικρή. Υπάρχει ποικιλία συναρτήσεων που χρησιμοποιούνται ως συνάρτηση σφάλματος και η επιλογή εξαρτάται από τη φύση του προβλήματος.

4.3.1 Mean Square Error (MSE)

Το μέσο τετραγωνικό σφάλμα χρησιμοποιείται συνήθως σε δίκτυα που λύνουν προβλήματα που ανήκουν στην γενικότερη κατηγορία της παλινδρόμησης. Όπως φαίνεται στον τύπο 4.1, παίρνοντας πάντα θετική τιμή, η συνάρτηση κόστους μας επιστρέφει την ευκλείδεια απόσταση δύο διανυσμάτων με βέλτιστη τιμή το μηδέν, όπου σημαίνει ότι έχει βρεθεί η τέλεια καμπύλη που εφαρμόζει τέλεια πάνω στα δεδομένα, το οποίο δεν συμβαίνει συνήθως.

$$(y_i - \bar{y}_i)J_{MSE} = \sum_{p=1}^{N} \left\| t_p - y_p \right\|^2 = \sum_{p=1}^{N} \sum_{i=1}^{m} \left(t_{p,i} - y_{p,i} \right)^2$$

(4.1) [9]

Όπου:

- $\boldsymbol{t_p}$ είναι το διάνυσμα στόχος για το p-οστό πρότυπο με διάσταση m.
- $\boldsymbol{y_p}$ είναι το διάνυσμα εξόδου για το p-οστό πρότυπο με διάσταση m.
- $\boldsymbol{N}$ είναι ο αριθμός των προτύπων.

4.3.2 Cross-Entropy Loss

Η συνάρτηση κόστους Cross-Entropy Loss χρησιμοποιείται συνήθως σε δίκτυα που λύνουν προβλήματα που ανήκουν στην γενικότερη κατηγορία της κατηγοριοποίησης. Η συνάρτηση μετρά την απόκλιση μεταξύ των πιθανοτήτων που προβλέπει το δίκτυο και των πραγματικών κλάσεων των στόχων. Ανάλογα με το πλήθος των κλάσεων στόχους χρησιμοποιείται και η ανάλογη παραλλαγή της συνάρτησης, για δυαδικά προβλήματα χρησιμοποιείται η συνάρτηση Binary Cross Entropy όπως φαίνεται στον τύπο 4.2.

Binary Cross Entropy:

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} [y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))]$$

(4.2) [32]

Όπου:

- $\boldsymbol{y_i}$ είναι η πραγματική ετικέτα (0 ή 1) για το i-οστό δείγμα.
- $\boldsymbol{p(y_i)}$ είναι η προβλεπόμενη πιθανότητα της κλάσης 1 για το i-οστό δείγμα.
- $\boldsymbol{N}$ είναι ο αριθμός των δειγμάτων.

Προβλήματα που απαιτούν ταξινόμηση περισσότερων κλάσεων χρησιμοποιείται η παραλλαγή της συνάρτησης Categorical Cross Entropy όπως φαίνεται στον τύπο 4.3.

Categorical Cross Entropy:

$$CCE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \cdot log(p(y_{ij}))$$

(4.3) [32]

Όπου:

- $\boldsymbol{y_{ij}}$ είναι η πραγματική ετικέτα (0 ή 1) εάν είναι j είναι η σωστή κλάση για το i-οστό δείγμα.
- $\boldsymbol{p(y_{ij})}$ είναι η προβλεπόμενη πιθανότητα της κλάσης j για το i-οστό δείγμα.
- $\boldsymbol{N}$ είναι ο αριθμός των δειγμάτων.
- $\boldsymbol{M}$ είναι ο αριθμός των κλάσεων.

4.3.3 Focal Loss

Η συνάρτηση κόστους Focal Loss χρησιμοποιείται συνήθως σε δίκτυα που λύνουν προβλήματα που ανήκουν στην γενικότερη κατηγορία της κατηγοριοποίησης με μεγάλη επιτυχία ειδικότερα στα περίπλοκα προβλήματα ανίχνευσης αντικειμένων. Η συνάρτηση επικεντρώνεται στο να ταξινομηθούν σωστά δείγματα που σε διαφορετική περίπτωση είναι πολύ δύσκολο να ταξινομηθούν, αυτό επιτυγχάνεται χρησιμοποιώντας την ειδική υπερπαράμετρο γ, όπως φαίνεται στον τύπο 4.4. Η υπερπαράμετρος μικραίνει το βάρος των εύκολα ταξινομήσιμων δειγμάτων και ενισχύει των δύσκολων, προσπαθώντας να εξαλείψει το πρόβλημα της ανισορροπίας των κλάσεων κατά την διαδικασία της εκπαίδευσης.

$$FL = -\frac{1}{N}\sum_{i=1}^{N} (1 - p(y_i))^{\gamma} \cdot y_i \cdot log(p(y_i))$$

(4.4) [33]

Όπου:

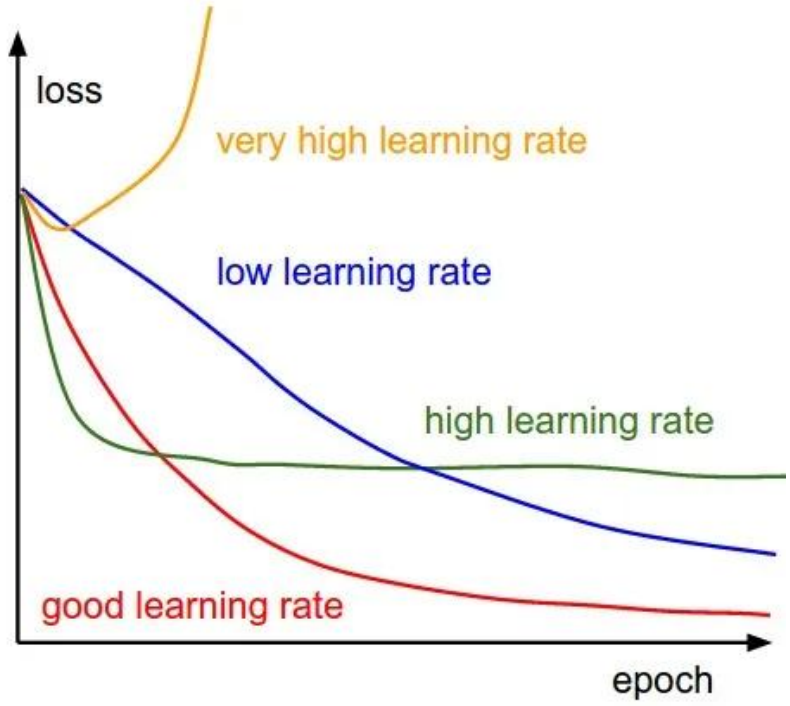
- $\boldsymbol{y_i}$ είναι η πραγματική ετικέτα (0 ή 1) για το i-οστό δείγμα.
- $\boldsymbol{p(y_{ij})}$ είναι η προβλεπόμενη πιθανότητα της κλάσης 1 για το i-οστό δείγμα.
- $\boldsymbol{\gamma}$ είναι μία υπερπαράμετρος που ελέγχει το βαθμό της εστίασης.
- $\boldsymbol{N}$ είναι ο αριθμός των δειγμάτων.

4.4 Ρυθμός Μάθησης

Ο ρυθμός μάθησης (learning rate) καθορίζει το μέγεθος της μεταβολής των παραμέτρων με σκοπό την ελαχιστοποίηση της συνάρτησης σφάλματος. Ο ρυθμός μάθησης παίζει πολύ σημαντικό ρόλο στην εκπαίδευση των νευρωνικών δικτύων και είναι μία από τις πιο σημαντικές υπερπαραμέτρους καθώς

μπορεί να επηρεάσει δραματικά την απόδοση του δικτύου. Κάθε φορά που γίνεται προσαρμογή των βαρών ο ρυθμός μάθησης δίνει το μέγεθος του βήματος που θα γίνει προς την εύρεση του ολικού ελαχίστου που μπορεί να πάρει η συνάρτηση κόστους.

Μία σωστή επιλογή ρυθμού μάθησης οδηγεί το δίκτυο σε σωστή σύγκλιση, με ομαλή εκπαίδευση, ελαχιστοποιώντας την συνάρτηση κόστους. Με μία πολύ μεγάλη τιμή του ρυθμού μάθησης προκύπτουν αστάθειες στην εκπαίδευση καθώς πραγματοποιούνται πολύ δραστικές προσαρμογές στα βάρη, ουσιαστικά προσπερνώντας το ελάχιστο της συνάρτησης κόστους. Με μία πολύ μικρή τιμή στον ρυθμό μάθησης πραγματοποιούνται ανεπαίσθητες προσαρμογές στα βάρη με αποτέλεσμα την πολύ αργή σύγκλιση αλλά και την μεγάλη περίπτωση συνάρτηση κόστους να βρεθεί σε ένα τοπικό ελάχιστο. [34]



Εικόνα 4.2: Γραφική αναπαράσταση επίδρασης διαφόρων τιμών learning rate στην απόδοση του μοντέλου.[35]

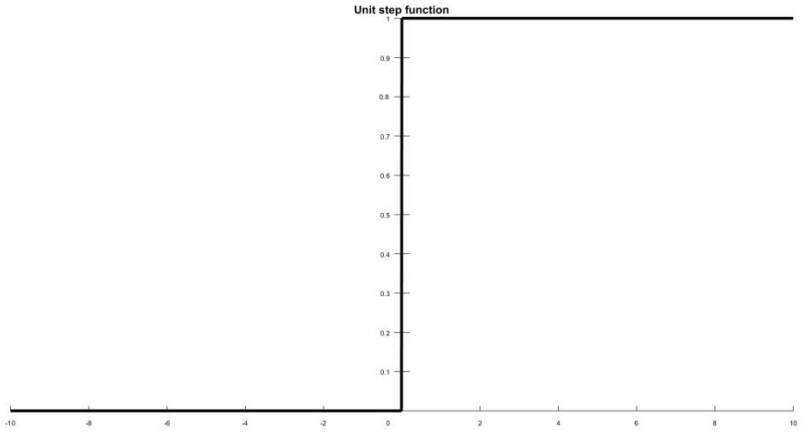
Για να βρεθεί η καλύτερη τιμή για τον ρυθμό μάθησης χρησιμοποιούνται συνήθως επαναλαμβανόμενα πειράματα με ένα εύρος από τιμές για τον ρυθμό μάθησης μέχρι να βρεθεί ο βέλτιστος. Επίσης, υπάρχουν μέθοδοι που προσαρμόζουν τον ρυθμό μάθησης (learning rate schedulers) κατά την διάρκεια της εκπαίδευσης χρησιμοποιώντας διάφορες παραμέτρους, όπως η εποχή που βρίσκεται η εκπαίδευση. [36]

4.5 Συναρτήσεις Ενεργοποίησης

Οι συναρτήσεις ενεργοποίησης είναι αναπόσπαστα στοιχεία των νευρωνικών δικτύων, καθώς εισάγουν την μη γραμμικότητα και μη γραμμικές σχέσεις στα νευρωνικά δίκτυα επιτυγχάνοντας την μοντελοποίηση πολύπλοκων συναρτήσεων και την εξαγωγή πολύπλοκων χαρακτηριστικών από τα δεδομένα εκπαίδευσης. Υπάρχουν πολλές διαφορετικές συναρτήσεις ενεργοποίησης με διάφορες ιδιαιτερότητες ανάλογα το δίκτυο στο οποίο θέλουμε να τις εντάξουμε.

4.5.1 Βηματική συνάρτηση (Step function)

Η συνάρτηση Step function ή βηματική είναι μια απλή συνάρτηση ενεργοποίησης που παίρνει μια τιμή εισόδου και ανάλογα με το αν η είσοδος είναι μεγαλύτερη ή ίση με ένα προκαθορισμένο κατώφλι, επιστρέφει μια σταθερή τιμή 1, ενώ αν είναι μικρότερη επιστρέφει 0.



Εικόνα 4.3: Γραφική αναπαράσταση της συνάρτησης Step Function με κατώφλι ίσο με το μηδέν. [60]

Ο τύπος της συνάρτησης:

$$f(x) = \begin{cases} 0\,, if\ x\ <\ κατώφλι \\ 1,\ if\ x\ \geq\ κατώφλι \end{cases}$$

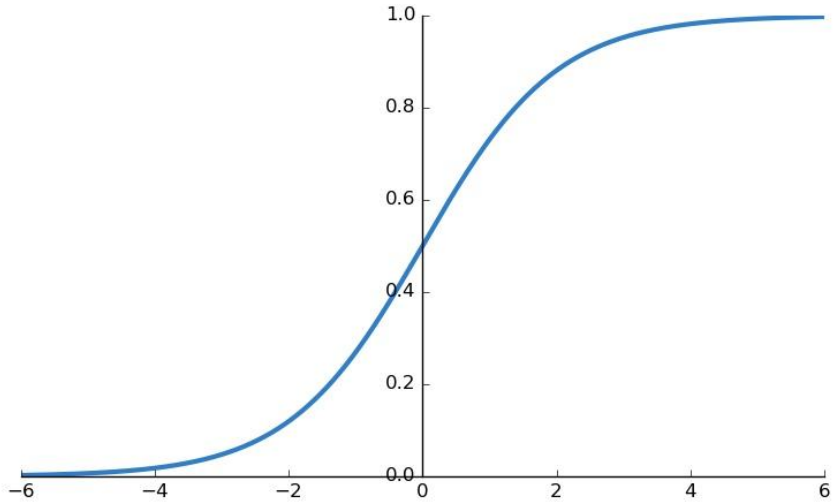
(4.5) [37]

Όπου:

- $\boldsymbol{x}$ είναι η τιμή εισόδου.

4.5.2 Σιγμοειδής συνάρτηση (Sigmoid function)

Η σιγμοειδής συνάρτηση είναι μια μη γραμμική συνάρτηση ενεργοποίησης και χρησιμοποιείται συνήθως σε δίκτυα που λύνουν προβλήματα δυαδικής ταξινόμησης, όπου η έξοδος πρέπει να ερμηνεύεται ως πιθανότητα. Η συνάρτηση μετασχηματίζει την είσοδο σε ένα εύρος μεταξύ 0 και 1, έτσι μπορεί να χρησιμοποιηθεί ως συνάρτηση ενεργοποίησης στο τελευταίο υπολογιστικό στρώμα ενός δικτύου για επίλυση προβλημάτων δυαδικής ταξινόμησης.



Εικόνα 4.4: Γραφική αναπαράσταση της συνάρτησης Sigmoid funtion. [61]

Ο τύπος της συνάρτησης:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.6}$$

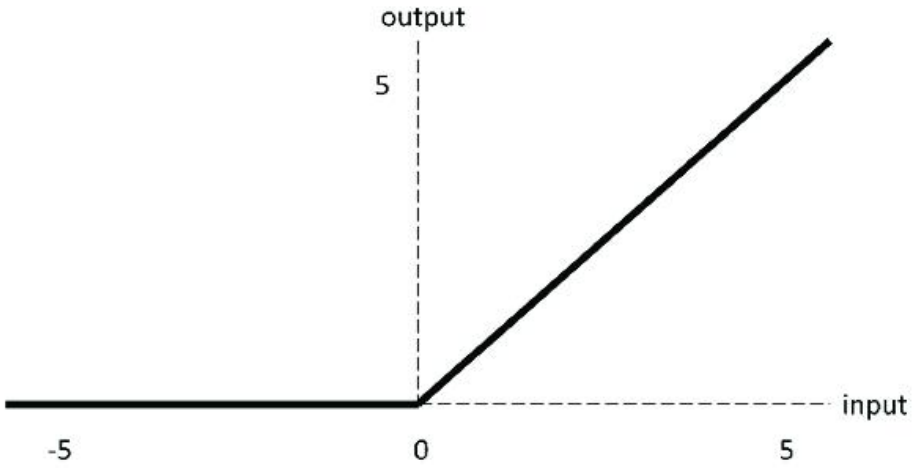
[38]

Όπου:

- $\boldsymbol{x}$ είναι η τιμή εισόδου.

4.5.3 Συνάρτηση Rectified Linear Unit (ReLU)

Η συνάρτηση Rectified Linear Unit είναι μία μη γραμμική συνάρτηση που έχει γίνει αρκετά δημοφιλής στα προβλήματα επεξεργασίας εικόνας και βίντεο. Το μεγάλο της πλεονέκτημα είναι η απλότητα της και στην ουσία η απόδοση της στην επιτάχυνση της εκπαίδευσης, καθώς χρησιμοποιείται σε πολλά κρυφά στρώματα. Η συνάρτηση επιστρέφει την είσοδο που έλαβε αν αυτή είναι θετική και 0 σε περίπτωση που η είσοδος είναι αρνητική ή μηδέν.



Εικόνα 4.5: Γραφική αναπαράσταση της συνάρτησης ReLU. [62]

Ο τύπος της συνάρτησης:

$$\mathrm{ReLU}(x) = \max(0, x)$$

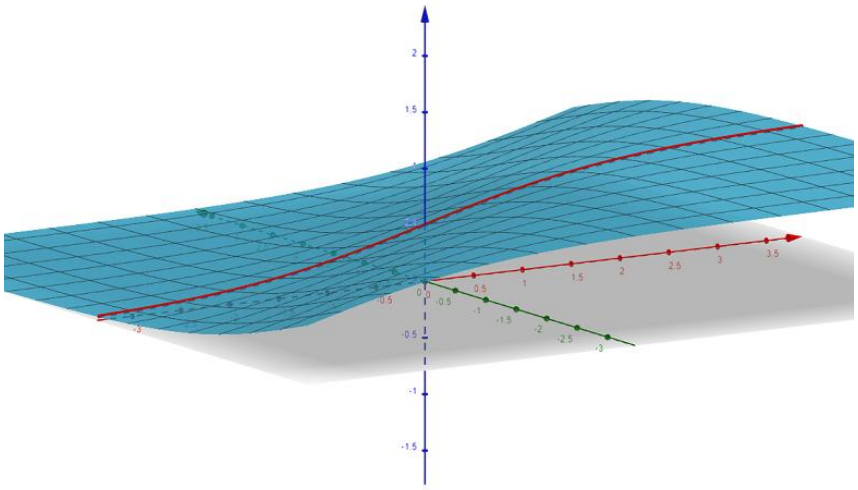
(4.7) [39]

Όπου:

- $\boldsymbol{x}$ είναι η τιμή εισόδου.

4.5.4 Συνάρτηση Softmax

Η συνάρτηση Softmax είναι μία μη γραμμική συνάρτηση που συνήθως χρησιμοποιείται στο τελευταίο υπολογιστικό στρώμα ενός νευρωνικού δικτύου για προβλήματα ταξινόμησης πολλαπλών κλάσεων. Η συνάρτηση μετασχηματίζει ένα διάνυσμα εισόδου σε ένα διάνυσμα ίδιου μεγέθους, όπου το κάθε στοιχείο του νέου διανύσματος αντιστοιχεί στην πιθανότητα της εισόδου να ανήκει σε μία συγκεκριμένη τις κλάσεις στόχου και τα στοιχεία του διανύσματος εξόδου αθροίζουν σε 1.



Εικόνα 4.6: Γραφική αναπαράσταση της συνάρτησης Softmax. [63]

Ο τύπος της συνάρτησης:

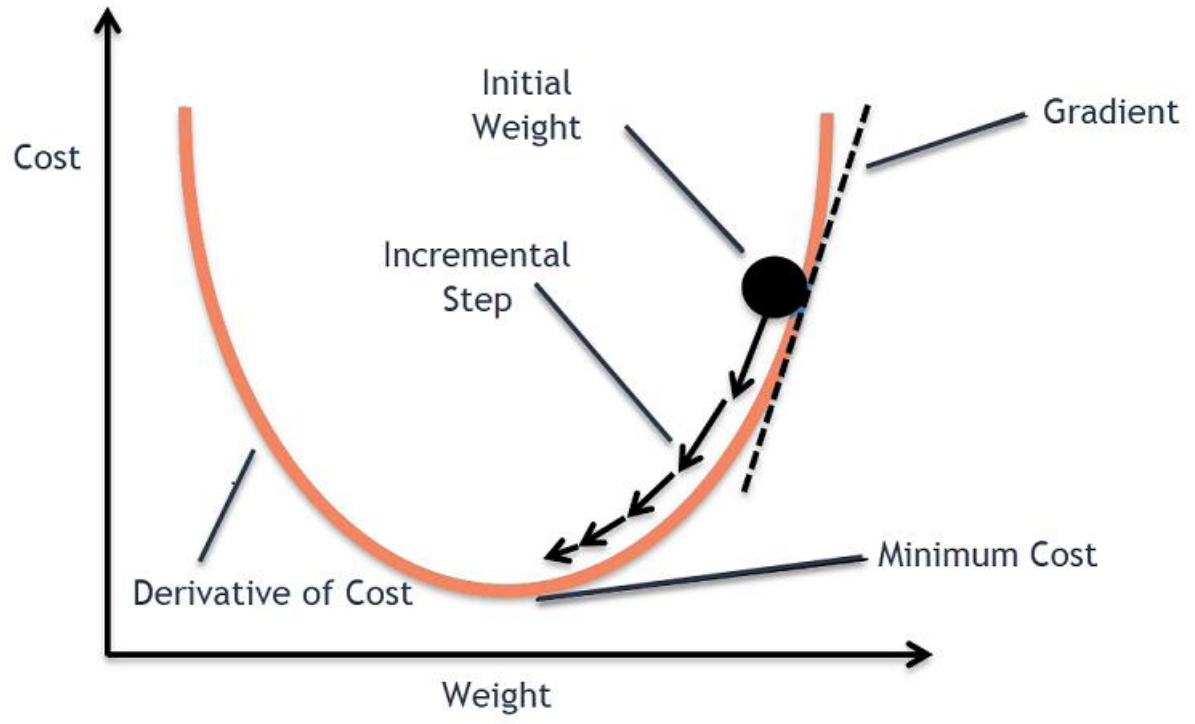
$$Softmax(x)_i = \frac{e^{x_i}}{\sum_{k=1}^{N} e_k^x} \quad (4.7)$$ [37]

Όπου:

- $\boldsymbol{x}$ είναι το διάνυσμα εισόδου.
- $\boldsymbol{x_i}$ είναι η τιμή του i-οστού στοιχείου του διανύσματος εισόδου.

4.6 Ελαχιστοποίηση συνάρτησης σφάλματος

Για την ελαχιστοποίηση αυτής της συνάρτησης χρησιμοποιείται η μέθοδος κατάβασης δυναμικού (Gradient Descent). Ο στόχος του είναι να βρει το ελάχιστο μιας συνάρτησης κόστους (ή σφάλματος) προσαρμόζοντας επαναληπτικά τις παραμέτρους του μοντέλου. Στην κατάβαση δυναμικού, η συνάρτηση κόστους *J* αξιολογεί την απόδοση του μοντέλου για όλα τα παραδείγματα εκπαίδευσης, και ο στόχος είναι η ελαχιστοποίηση αυτής της συνάρτησης κόστους.



Εικόνα 4.7: Παράδειγμα γραφικής αναπαράστασης του αλγορίθμου gradient decent. [31]

Ο αλγόριθμος χρησιμοποιεί την κλίση (gradient) για να βρει την κατεύθυνση με τη μεγαλύτερη αύξηση, και κινείται αντίστροφα αυτής της κατεύθυνσης για να ελαχιστοποιήσει τη συνάρτηση:

Η ανανέωση των παραμέτρων γίνεται με τη βοήθεια της κλίσης της συνάρτησης κόστους *J* ως εξής:

$$\theta = \theta - \alpha \cdot \nabla J\ (\theta)$$

(4.8) [38]

Όπου:

- **θ** είναι το σύνολο των παραμέτρων που προσαρμόζονται κατά τη διάρκεια της εκπαίδευσης.
- **α** είναι ο ρυθμός μάθησης (learning rate), που καθορίζει το μέγεθος του βήματος προς την κατεύθυνση της κλίσης.
- **∇ J** (**θ**) είναι η κλίση (gradient) της συνάρτησης κόστους, που υποδεικνύει την κατεύθυνση της μέγιστης αύξησης.

4.6.1 SGD και MB-GD

Στο πλαίσιο της μηχανικής μάθησης, η SGD (Stochastic Gradient Descent) είναι μέθοδος βελτιστοποίησης που χρησιμοποιείται για την εκπαίδευση νευρωνικών δικτύων [39]. Με παραλλαγή της μεθόδου και ορισμένες βελτιώσεις προκύπτει η MB-GD (Mini-Batch Gradient Descent). Οι βασικές έννοιες για κάθε μέθοδο είναι:

Stochastic Gradient Descent (SGD): Είναι μια μέθοδος βελτιστοποίησης που εφαρμόζει τον αλγόριθμο κατάβασης δυναμικού σε κάθε βήμα εκπαίδευσης, αλλά αντί να λαμβάνει υπόψη το σύνολο δεδομένων, χρησιμοποιεί κάθε δείγμα ξεχωριστά για τον υπολογισμό της κλίσης και κάνει την ενημέρωση των βαρών για το καθένα.

→ **Πλεονεκτήματα:** Κατάλληλος για μεγάλα σύνολα δεδομένων καθώς έχει χαμηλή απαιτητικότητα σε πόρους και μπορεί να επιτύχει γρηγορότερα σύγκλιση καθώς προσαρμόζει τα βάρη πάρα πολύ συχνά.

→ **Μειονεκτήματα:** Αυξημένη διακύμανση στην εκπαίδευση λόγο των συχνών προσαρμογών στα βάρη με κάθε δείγμα και ενδεχομένως να κολλήσει σε τοπικά ελάχιστα.

Mini-Batch Gradient Descent (SGD): Η MB-GD έχει την ίδια φιλοσοφία με την SGD αλλά αντί να λαμβάνει υπόψη κάθε δείγμα ξεχωριστά για τον υπολογισμό της κλίσης, κάνει την ενημέρωση των βαρών για ένα τυχαίο υποσύνολο του συνόλου εκπαίδευσης.

→ **Πλεονεκτήματα:** Τα μικρά υποσύνολα βοηθούν στην σταθερότητα της εκπαίδευσης, διατηρούν την ταχύτητα σε μεγάλο βαθμό και διατηρούν σχετικά χαμηλά την χρήση πόρων.

→ **Μειονεκτήματα:** Χρειάζεται την προσαρμογή μιας επιπλέον παραμέτρου, που είναι το μέγεθος των υποσυνόλων (batch size).

Η πιο κοινή μέθοδος από τις δύο είναι MB-GD καθώς είναι μια καλή μέση λύση ανάμεσα στην απλή κατάβαση δυναμικού και την SGD, προσφέροντας έναν καλό συμβιβασμό ανάμεσα στην χρήση πόρων και την γρήγορη και σωστή σύγκλιση ενός νευρωνικού δικτύου.

4.6.2 Ορμή

Η Ορμή (Momentum) είναι ένας σημαντικός όρος που χρησιμοποιείται στο πλαίσιο του γραμμικού και του στοχαστικού κατευθυνόμενης κατάβασης δυναμικού (Gradient Descent - GD) στη βελτιστοποίηση των μοντέλων μηχανικής μάθησης. Στον αλγόριθμο της Ορμής, κατά την ενημέρωση των παραμέτρων ενός μοντέλου κατά τη διάρκεια της εκπαίδευσης, λαμβάνεται υπόψη ο μέσος όρος των προηγούμενων βημάτων για να επηρεαστεί η κατεύθυνση κίνησης. Συγκεκριμένα, η ορμή παρέχει ένα μηχανισμό που δίνει "ταχύτητα" στον αλγόριθμο GD για να αποφύγει τον τοπικό ελάχιστο και να συγκλίνει προς το γενικό ελάχιστο της συνάρτησης κόστους. Η βασική ιδέα της Ορμής είναι η χρήση της τρέχουσας ταχύτητας ενημέρωσης (velocity) για να αλλάξει το βήμα ενημέρωσης των παραμέτρων. Κατά την εκτέλεση του αλγορίθμου, η τρέχουσα ταχύτητα είναι ένας πολλαπλασιαστής του προηγούμενου βήματος ενημέρωσης και του τρέχοντος κλίση (gradient), λαμβάνοντας υπόψη την κατεύθυνση και το μέγεθος της προηγούμενης αλλαγής.

Η χρήση της Ορμής μπορεί να βελτιώσει τη σύγκλιση του αλγορίθμου GD, ιδίως σε περιπτώσεις όπου η συνάρτηση κόστους έχει πολύ καμπύλη ή έχει πολλά τοπικά ελάχιστα. Αυτό επιτυγχάνεται με την "ομαλοποίηση" της κατεύθυνσης κίνησης, επιτρέποντας στον αλγόριθμο να ξεφύγει από τα τοπικά ελάχιστα και να προχωρήσει προς το γενικό ελάχιστο.

Είναι σημαντικό να προσαρμόζεται προσεκτικά ο συντελεστής ορμής κατά την εκπαίδευση ενός μοντέλου, καθώς αυτός μπορεί να επηρεάσει την ταχύτητα σύγκλισης και την απόδοση του αλγορίθμου. Ο τύπος της Ορμής (Momentum) στον αλγόριθμο βελτιστοποίησης Gradient Descent περιγράφεται ως εξής:

Έστω $\boldsymbol{\theta}$ οι παράμετροι που ενημερώνουμε σε κάθε βήμα του Gradient Descent. Η ορμή $\boldsymbol{v}t$ στο χρόνο $\boldsymbol{t}$ για την ενημέρωση των παραμέτρων ορίζεται ως εξής:

$$v_t = \beta v_{t-1} + \eta \nabla_\theta J(\theta_t)$$

(4.9) [40]

Όπου:

- $\boldsymbol{\beta}$ είναι ο συντελεστής ορμής (momentum coefficient), συνήθως κινείται στο διάστημα [0, 1].
- $\boldsymbol{v_{t-1}}$ είναι η προηγούμενη τιμή της ορμής στο χρόνο t-1.
- $\boldsymbol{\eta}$ είναι ο ρυθμός μάθησης (learning rate).
- $\boldsymbol{\nabla_\theta J(\theta_t)}$ είναι ο γραμμικός παράγοντας (gradient) της συνάρτησης κόστους J ως προς τις παραμέτρους θ στο σημείο θ.

4.6.3 ADAM και ADAMW

ADAM: Ο ADAM (Adaptive Moment Estimation) είναι ένας δημοφιλής αλγόριθμος βελτιστοποίησης που χρησιμοποιείται συχνά στην εκπαίδευση νευρωνικών δικτύων. Είναι μια παραλλαγή του αλγορίθμου Stochastic Gradient Descent (SGD) και στοχεύει στη βελτίωση της απόδοσης και της ταχύτητας σύγκλισης.

Το βασικό χαρακτηριστικό του ADAM είναι η χρήση δύο κύριων μηχανισμών:

Ορμή (Momentum): Αντίστοιχα με άλλους αλγορίθμους, χρησιμοποιεί την έννοια της ορμής (momentum) για να κινεί τις ενημερώσεις των παραμέτρων στην κατεύθυνση που δείχνει η κλίση.
Κλιμάκωση (Scaling): Προσαρμόζει τον ρυθμό μάθησης (learning rate) για κάθε παράμετρο ξεχωριστά, με βάση την παρατήρηση της πορείας της κλίσης.

Η συνδυαστική χρήση αυτών των μηχανισμών επιτρέπει στον ADAM να είναι αποτελεσματικός σε μια ποικιλία διαφορετικών προβλημάτων εκπαίδευσης. Επιπλέον, ο ADAM είναι λιγότερο ευαίσθητος σε παραμέτρους όπως ο ρυθμός μάθησης σε σύγκριση με άλλες μεθόδους βελτιστοποίησης. Παρ' όλα αυτά, η απόδοση του ADAM μπορεί να διαφέρει ανάλογα με το πρόβλημα και την αρχιτεκτονική του δικτύου, οπότε συνίσταται η δοκιμή και η παρακολούθηση της απόδοσης σε διάφορες παραμέτρους.

ADAMW: Ο ADAMW είναι μια παραλλαγή του βελτιστοποιητή ADAM που προτείνεται για την εκπαίδευση νευρωνικών δικτύων. Αυτή η παραλλαγή προσθέτει έναν όρο βαρύτητας (weight decay term) στη λειτουργία του ADAM με σκοπό να αντιμετωπίσει το πρόβλημα του overfitting και να βελτιώσει τη γενίκευση του μοντέλου. Ο όρος βαρύτητας προστίθεται στη συνάρτηση κόστους κατά την εκπαίδευση. Αυτή η παράμετρος έχει το ρόλο να προσθέσει μια επιπλέον ποινή στις μεγάλες τιμές των βαρών του μοντέλου, ενθαρρύνοντας τις τιμές αυτές να παραμείνουν μικρές, βοηθώντας έτσι στον περιορισμό του overfitting.

Η προσθήκη του όρου βαρύτητας στον ADAM δίνει την ονομασία ADAMW, όπου το "W" αναφέρεται στη βαρύτητα (Weight decay). Η χρήση αυτής της παραλλαγής ADAMW μπορεί να βελτιώσει την απόδοση και τη γενίκευση του μοντέλου σε κάποιες περιπτώσεις συγκριτικά με τον κανονικό ADAM, ειδικά όταν το μοντέλο παρουσιάζει προσαρμοστικά βάρη που μπορούν να οδηγήσουν σε μεγάλες τιμές βαρών και ενδεχομένως σε overfitting.

4.7 Προ-Εκπαιδευμένα Μοντέλα και Μάθηση Μεταφοράς

Τα προ-εκπαιδευμένα μοντέλα και η μάθηση μεταφοράς (Transfer Learning) [41] αποτελούν δύο σημαντικές τεχνικές στον χώρο της μηχανικής μάθησης και της εκμάθησης νευρωνικών δικτύων.

Προεκπαιδευμένα Μοντέλα (Pretrained Models):

Τα προ-εκπαιδευμένα μοντέλα είναι νευρωνικά δίκτυα τα οποία έχουν εκπαιδευτεί σε μεγάλα σύνολα δεδομένων για πολλαπλές εργασίες, όπως η αναγνώριση εικόνων σε γενικό επίπεδο, η αναγνώριση αντικειμένων, η αναγνώριση προσώπων, κ.λ.π. Κατά τη διάρκεια της εκπαίδευσής τους, τα προ-εκπαιδευμένα μοντέλα έχουν αναπτύξει κατανοητές αναπαραστάσεις των δεδομένων που τους έχουν παρουσιαστεί. Υπάρχουν πολλά διαθέσιμα προεκπαιδευμένα μοντέλα. Μερικά από αυτά τα μοντέλα έχουν εκπαιδευτεί σε δημοφιλή σύνολα δεδομένων εικόνας, όπως το ImageNet [42] και το COCO [43], ενώ κάποια, έχουν εκπαιδευτεί σε μεγάλα σύνολα δεδομένων κειμένου, όπως η Wikipedia.

Μάθηση Μεταφοράς (Transfer Learning):

Η μάθηση μεταφοράς είναι η τεχνική της χρήσης προ-εκπαιδευμένων μοντέλων για την επίλυση νέων προβλημάτων ή την εκπαίδευση μοντέλων σε νέα σύνολα δεδομένων που δεν έχουν εκπαιδευτεί αρχικά. Σε αυτήν την τεχνική, τα προ-εκπαιδευμένα μοντέλα χρησιμοποιούνται ως αρχικά σημεία εκκίνησης και οι προηγούμενες γνώσεις που έχουν αποκτηθεί από την προηγούμενη εκπαίδευση μεταφέρονται στο νέο πρόβλημα. Τα προ-εκπαιδευμένα μοντέλα, λόγω της εκπαίδευσής τους σε

μεγάλα σύνολα δεδομένων, έχουν μάθει να εξάγουν γενικές χαρακτηριστικές αναπαραστάσεις από τα δεδομένα.

Η μάθηση μεταφοράς επωφελείται από αυτές τις γενικές αναπαραστάσεις, επιτρέποντας στο μοντέλο να μάθει αποτελεσματικά από πολύ λιγότερα δεδομένα σε νέες εργασίες ή προβλήματα. Η συνήθης διαδικασία της μάθησης μεταφοράς περιλαμβάνει τη χρήση του προ-εκπαιδευμένου μοντέλου για την εκπαίδευση ενός νέου μοντέλου σε ένα νέο πρόβλημα, είτε μέσω της προσαρμογής ορισμένων στρωμάτων του μοντέλου είτε μέσω της προσθήκης νέων στρωμάτων σε αυτό.

Κεφάλαιο 5ο: Υλοποίηση

5.1 Εισαγωγή

Το κομμάτι που επιχειρεί να βελτιώσει αυτή η εργασία είναι η καλύτερη αναγνώριση και κατηγοριοποίηση τον πιο άμεσα εφαρμόσιμων σημάτων που θα συναντήσει ένα αυτόνομο όχημα στο δρόμο. Για τον σκοπό αυτόν θα χρησιμοποιήσουμε αρχικά ένα προεκπαιδευμένο νευρωνικό δίκτυο για αναγνώριση και ανίχνευση αντικειμένων ενός σταδίου, το οποίο θα εκπαιδεύσουμε σε ένα σχετικά μικρό σύνολο δεδομένων από συνθήκες οδήγησης, ώστε να αναγνωρίζει τα σήματα που περιέχονται στα δεδομένα. Εφόσον έχουμε αυτό το μοντέλο ως βάση, θα εισάγουμε μία ακόμα παράμετρο στο σύνολο δεδομένων για κάθε σήμα που περιλαμβάνει κάθε εικόνα, ώστε να ορίσουμε αν το εκάστοτε σήμα είναι άμεσα εφαρμόσιμο στο όχημα. Διευκρινίζοντας, ως σημαντικό αναφέρεται ένα σήμα οδικής κυκλοφορίας το οποίο μπορεί να έχει άμεση επίδραση στο όχημα μας, για παράδειγμα σε περίπτωση που το σήμα αναφέρεται στην λωρίδα που κινείται το όχημα πρέπει να καθοριστεί σημαντικό. Επιπλέον, αν είναι πολλά σήματα στο πεδίο ανίχνευσης, για παράδειγμα σε δύο αλλεπάλληλες διασταυρώσεις, καθορίζουμε ως σημαντικά αυτά που έχουν την πιο άμεση εφαρμογή στο όχημα.

Χρησιμοποιώντας το νέο σύνολο δεδομένων θα επαναλάβουμε ακριβώς την ίδια εκπαίδευση στο δίκτυο με τροποποιήσεις σε μία από τις συναρτήσεις κόστους, ώστε να λαμβάνεται υπόψη η παράμετρος άμεσης εφαρμοσιμότητας που προστέθηκε. Τέλος θα συγκρίνουμε τα αποτελέσματα των δύο μοντέλων στην ανίχνευση και κατηγοριοποίηση σημάτων αλλά και τις διαφορές στην κατηγοριοποίηση των άμεσα εφαρμόσιμων σημάτων.

5.2 Εργαλεία

Για την υλοποίηση των πειραμάτων εκπαίδευσης των νευρωνικών δικτύων θα χρησιμοποιηθεί η online πλατφόρμα goole colab η οποία μας προσφέρει ένα περιβάλλον με προ εγκατεστημένη την γλώσσα προγραμματισμού Python και βιβλιοθήκες της. Το περιβάλλων μας παρέχει πόρους συστήματος που απευθύνονται στην εκπαίδευση νευρωνικών και είναι ικανοποιητικοί ακόμα και στην δωρεάν έκδοση.

5.2.1 Πόροι συστήματος

Τα νευρωνικά δίκτυα και γενικότερα τα μοντέλα μηχανικής μάθησης που επιτυγχάνουν τις καλύτερες επιδόσεις στην κατηγορία προβλημάτων που περιλαμβάνουν όραση, δηλαδή κάποια μορφή επεξεργασίας εικόνας συνήθως συνοδεύονται και από ένα αρκετά μεγάλο όγκο απαιτήσεων σε πόρους συστήματος για την εκπαίδευση τους.

Για την εκπαίδευση των νευρωνικών δικτύων που επιλέχθηκαν χρειάζονται αρκετά μεγάλο αποθηκευτικό χώρο. Πρώτον, λόγο του μεγάλου όγκου που καταλαμβάνουν τα ίδια και σε περιπτώσει που χρειάζεται να αποθηκεύσουμε διαφορετικά στάδια της εκπαίδευσης τους ο χώρος καταλαμβάνεται σχετικά γρήγορα και δεύτερον λόγο τον μεγάλο όγκο τον συνόλων δεδομένων που είναι αναγκαία για την εκπαίδευση τους.

Επιπλέον, για την εκπαίδευση σχεδόν απαιτήσει η ύπαρξη ικανοποιητικής κάρτας γραφικών (GPU) με αρκετή μνήμη για να ικανοποιήσει της ανάγκες των μεγαλύτερων, σε όγκο, παραμετρικά δικτύων και να εκτελεί την διαδικασία της εκπαίδευσης σε ένα λογικό χρονικό πλαίσιο.
Όταν μετακινούνται αυτοί οι μεγάλοι όγκοι δεδομένων κατά την εκπαίδευση των δικτύων προκύπτει η ανάγκη για χρήση της κύριας μνήμης RAM, καθώς χρειάζεται πολλές φορές να γίνουν επεξεργασίας στα δεδομένα εκπαίδευσης σε διάφορα στάδια. Η χρήση του της κεντρικής μονάδας επεξεργασίας μπαίνει σε δεύτερη μοίρα εφόσον υλοποιούμε εκπαίδευση με GPU, όμως ένας πολύ αργός επεξεργαστής μπορεί να μας καθυστερήσει αρκετά στην επεξεργασία που απαιτείται να γίνει στα δεδομένα σε ένα κύκλο εκπαίδευσης.

Πίνακας 5.1: Πόροι συστήματος εκπαίδευσης.

RAM	12.7 GB
ΔΙΣΚΟΣ	78.2 GB
CPU	1vCPU
GPU	Tesla T4 16 GB

Στον πίνακα 5.1 φαίνονται οι πόροι του συστήματος που χρησιμοποιήθηκε για την επεξεργασία όλων των συνόλων δεδομένων αλλά και την εκπαίδευση των νευρωνικών δικτύων.

5.2.2 Python

Το βασικό εργαλείο που θα χρησιμοποιήσουμε είναι η γλώσσα Python με την οποία θα γίνουν και τα πειράματα. Η Python είναι μια δημοφιλής γλώσσα προγραμματισμού υψηλού επιπέδου που δημιουργήθηκε από τον Guido van Rossum και πρωτοκυκλοφόρησε το 1991. Είναι σχεδιασμένη για να είναι ευανάγνωστη και ευκολομνημόνευτη, με έμφαση στην αναγνωσιμότητα του κώδικα. Η Python υποστηρίζει πολλαπλές τεχνικές προγραμματισμού, συμπεριλαμβανομένων του αντικειμενοστραφούς προγραμματισμού, του δομημένου προγραμματισμού και του συναρτησιακού προγραμματισμού. Είναι επίσης γνωστή για την μεγάλη κοινότητα της, τα εκτεταμένα πακέτα βιβλιοθηκών και το μεγάλο οικοσύστημα υποστήριξης. [44]

Η γλώσσα έχει χρησιμοποιηθεί ευρέως για εφαρμογές όπως οι ιστοσελίδες, η μηχανική μάθηση, η τεχνητή νοημοσύνη γενικότερα, ο έλεγχος εκδόσεων, η ανάλυση δεδομένων, και πολλά άλλα. Η απλή σύνταξη και η ευελιξία της την καθιστούν μια δημοφιλή επιλογή για προγραμματιστές σε διάφορους τομείς. Το βασικό μειονέκτημα είναι οι επιδόσεις της σε σύγκριση με γλώσσες όπως η C ή C++ καθώς είναι γλώσσα που χρησιμοποιεί διερμηνέα (Interpreter) και όχι μεταγλωττιστή όπως η γλώσσες που προαναφέραμε, καθιστώντας την υποδεέστερη επιλογή σε εφαρμογές ό χρόνος απόκρισης είναι πολύ σημαντικός.

5.2.3 Pytorch και Pytorch Lightning

Το **PyTorch** είναι μία βιβλιοθήκη ανοιχτού κώδικα μηχανικής μάθησης για την ανάπτυξη και την εκπαίδευση νευρωνικών δικτύων. Δημιουργήθηκε από το Facebook και έχει αποκτήσει δημοτικότητα λόγω της ευκολίας χρήσης, της ευελιξίας και της δυνατότητας επιτάχυνσης της ανάπτυξης μοντέλων μηχανικής μάθησης. [45] Επιπλέον, χρησιμοποιεί ένα δυναμικό

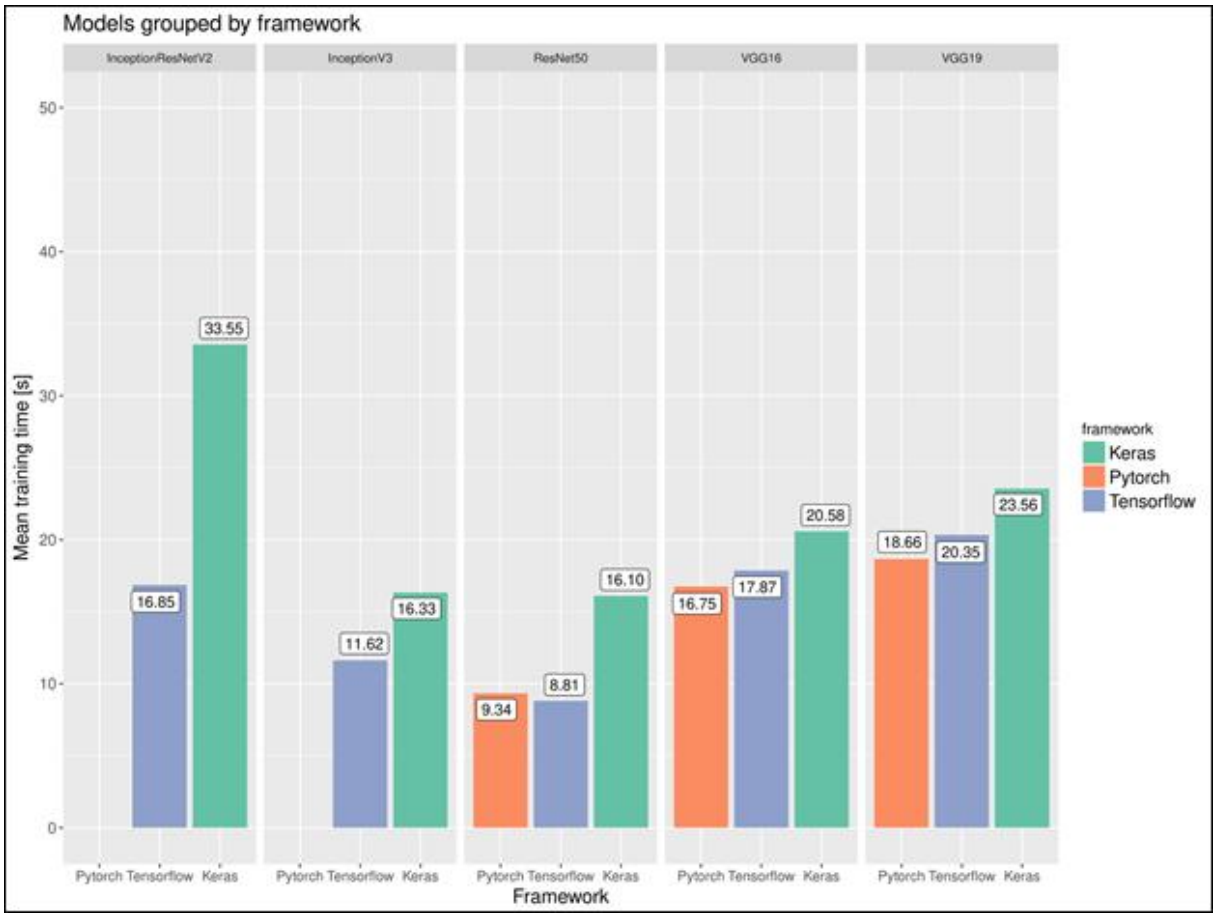
υπολογιστικό γράφημα, που σημαίνει ότι ο χρήστης μπορεί να ορίσει και να τροποποιήσει το γράφημα κατά τη διάρκεια της εκτέλεσης, πράγμα που καθιστά ευκολότερη την αποσφαλμάτωση και τον πειραματισμό. Διαθέτει μια ενεργή κοινότητα και ένα πλούσιο σύνολο τεκμηρίωσης, καθώς και πολλά παραδείγματα κώδικα για την εκμάθηση και την εφαρμογή των διαφόρων λειτουργιών καθιστώντας την μία από της δημοφιλέστερες βιβλιοθήκες για την ανάπτυξη και την εκπαίδευση μοντέλων μηχανικής μάθησης.

Το **Pytorch Lighning** είναι ένα ελαφρύ περίβλημα (wrapper) για το PyTorch που απλοποιεί τη διαδικασία εκπαίδευσης και έρευνας για μοντέλα βαθιάς μάθησης. Παρέχει μια υψηλού επιπέδου διεπαφή με προεγκατεστημένα στοιχεία, επιτρέποντας σε ερευνητές και προγραμματιστές να επικεντρωθούν περισσότερο στη δημιουργία και στην πειραματική διαδικασία με τα μοντέλα τους, αντί να ασχολούνται με κώδικα που επαναλαμβάνεται για τη διαδικασία εκπαίδευσης, την καταγραφή και άλλες εργασίες. Το PyTorch Lightning είναι σχεδιασμένο να είναι ελαφρύ και εύκολο στην χρήση του ανάλογα τις ανάγκες του καθένα. Σας επιτρέπει να οργανώνετε τον κώδικά σας σε ξεχωριστά τμήματα για τα δεδομένα, το μοντέλο και τη λογική εκπαίδευσης, καθιστώντας τη βάση κώδικα πιο αρθρωτή και ευανάγνωστη. [46]

Σύγκριση Pytorch με άλλες πλατφόρμες ανταγωνιστές:

Κάθε μία από τις επιλογές έχει τα δικά τις πλεονεκτήματα και μειονεκτήματα. [47] Το PyTorch είναι γνωστό για το δυναμικό υπολογιστικό γράφημα, προσφέροντας ευελιξία και ευανάγνωστο κώδικα, και έχει κερδίσει δημοφιλία στον ερευνητικό χώρο. Το TensorFlow [48], που παρέχει είτε στατικά είτε δυναμικά υπολογιστικά γραφήματα, διαθέτει ένα εκτενές οικοσύστημα και ευρεία βιομηχανική χρήση. Το Keras [49], αρχικά ένα υψηλού επιπέδου API, ενσωματώθηκε πλέον στο TensorFlow, παρέχοντας μια φιλική προς τον χρήστη διεπαφή κατάλληλη για το στάδιο ανάπτυξης πρωτοτύπων. Η επιλογή μεταξύ αυτών των πλαισίων εξαρτάται από τις προτιμήσεις του κάθε ατόμου και τις απαιτήσεις του έργου, με το PyTorch να επικρατεί για το δυναμικό του γράφημα και τη δημοφιλία του στην έρευνα, το TensorFlow για το εκτενές οικοσύστημά του και τη βιομηχανική υιοθέτηση, και το Keras για την απλότητά του, πλέον στενά ενσωματωμένο με το TensorFlow.

Η διαφορά στην απόδοση και την αποτελεσματικότητα εκπαίδευσης φαίνεται στο σχήμα 5.1, όπου συγκρίνονται το Pytorch, Keras και το Tensoflow στον χρόνο εκπαίδευσης, ώστε να γίνει αντιληπτή η σωστότερη διαχείριση πόρων. Όπως γίνεται αντιληπτό το Pytorch με το Tensorflow είναι αρκετά κοντά στην απόδοση τους με το Keras να καθυστερεί αισθήτα περισσότερο στην εκπαίδευση.



Σχήμα 5.1: Σύγκριση των Pytorch, Tensorflow και Keras στον χρόνο εκπαίδευσης. Στον οριζόντιο άξονα βρίσκονται τα μοντέλα με τις ανάλογες βιβλιοθήκες και στον κάθετο ο μεσος χρόνος εκπαίδευσης. [50]

5.2.4 Weights and Biases (wandb)

Για την οπτικοποίηση και την καταγραφή των πειραμάτων εκτός από βιβλιοθήκες της Python θα χρησιμοποιήσουμε και την πλατφόρμα Weights and Biases (wandb). Η χρήση της είναι πολύ εύκολη καθώς υποστηρίζεται από πολλές βιβλιοθήκες μηχανικής μάθησης στην Python αλλά και την αυτόνομη βιβλιοθήκη της. Παρέχει μία online πλατφόρμα όπου καταγράφονται και οπτικοποιούνται όλα τα δεδομένα των πειραμάτων που εκτελούμε αλλά και ένα ολοκληρωμένο οικοσύστημα για την εκπαίδευση μοντέλων μηχανικής μάθησης, εμείς θα την χρησιμοποιήσουμε μόνο για την καταγραφή και οπτικοποίηση των πειραμάτων. [51]

5.2.5 Cuda

Για την εκπαίδευση των νευρωνικών δικτύων χρησιμοποιούνται κάρτες γραφικών οι οποίες έχουν ειδικά κατασκευασμένους πυρήνες για την επιτάχυνση της εκπαίδευσης, μία τέτοια τεχνολογία είναι και το CUDA. Όταν αναφερόμαστε στο CUDA της NVIDIA αναφερόμαστε σε μια πλατφόρμα παράλληλου υπολογισμού που αναπτύχθηκε από την NVIDIA για την επιτάχυνση των υπολογισμών χρησιμοποιώντας τις κάρτες γραφικών (GPU) της ομώνυμης εταιρείας. Το CUDA επιτρέπει στους προγραμματιστές να εκμεταλλευτούν την ισχύ των GPU για να επιτύχουν υψηλή απόδοση σε εφαρμογές παράλληλου υπολογισμού. [52]

Ορισμένα βασικά στοιχεία του CUDA περιλαμβάνουν:

Πυρήνες CUDA (CUDA Cores): Οι μονάδες επεξεργασίας στις GPU που είναι υπεύθυνες για την εκτέλεση των παράλληλων υπολογισμών. Οι πυρήνες CUDA είναι σχεδιασμένοι για να εκτελούν πολλαπλούς υπολογισμούς ταυτόχρονα.

CUDA Toolkit: Ένα σύνολο εργαλείων και βιβλιοθηκών που παρέχονται από την NVIDIA για την ανάπτυξη εφαρμογών που χρησιμοποιούν την τεχνολογία CUDA.

CUDA C: Μια εκδοχή της γλώσσας προγραμματισμού C που επεκτείνεται για την υποστήριξη παράλληλων υπολογισμών με χρήση του CUDA Toolkit.

GPU-accelerated Libraries: Προηγμένες βιβλιοθήκες που έχουν βελτιστοποιηθεί για να εκμεταλλεύονται την υψηλή απόδοση των GPU. Περιλαμβάνουν βιβλιοθήκες για μαθηματικούς υπολογισμούς, επεξεργασία εικόνας, επεξεργασία σήματος και άλλες εφαρμογές.

Με το CUDA, οι προγραμματιστές έχουν τη δυνατότητα να αξιοποιήσουν την ισχύ των GPU για ευρεία γκάμα εφαρμογών, συμπεριλαμβανομένων των επιστημονικών υπολογισμών, της τεχνητής νοημοσύνης, της επεξεργασίας εικόνας και πολλών άλλων.

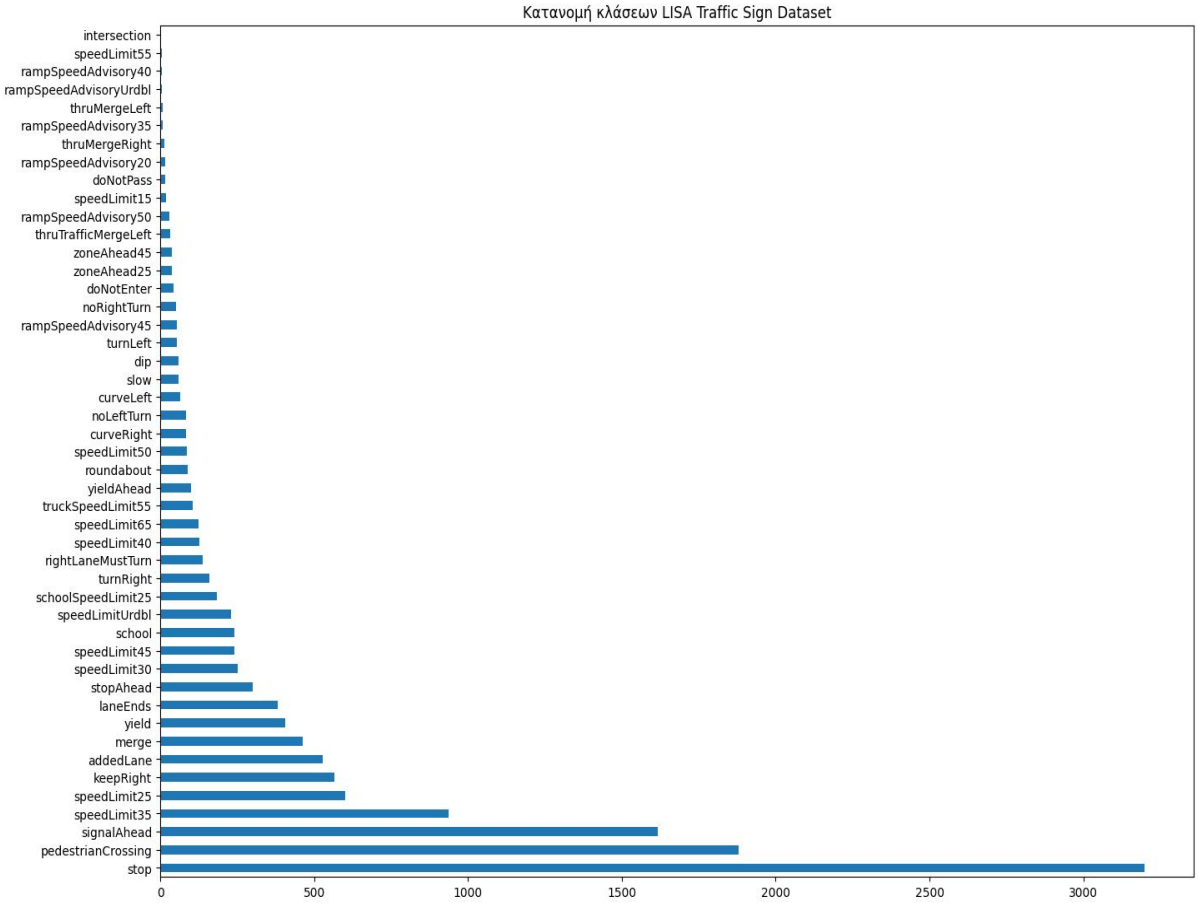
5.3 Dataset

Το σύνολο δεδομένων που θα χρησιμοποιηθεί θα είναι ένα υποσύνολο από το LISA Traffic Sign Dataset [53] , το οποίο περιέχει εικόνες από σενάρια πραγματικής οδήγησης από μία εμπρόσθια κάμερα από την ευρύτερη περιοχή του san diego των ηνωμένων πολιτειών της Αμερικής. Στο σύνολο δεδομένων εμπεριέχονται εικόνες ασπρόμαυρες αλλά και έγχρωμες όπως φαίνεται στις εικόνες 5.1 και 5.2 αντίστοιχα.

Εικόνα 5.1: Τυχαία έγχρωμη εικόνα από το σύνολο δεδομένων Lisa Traffic Sign Dataset.

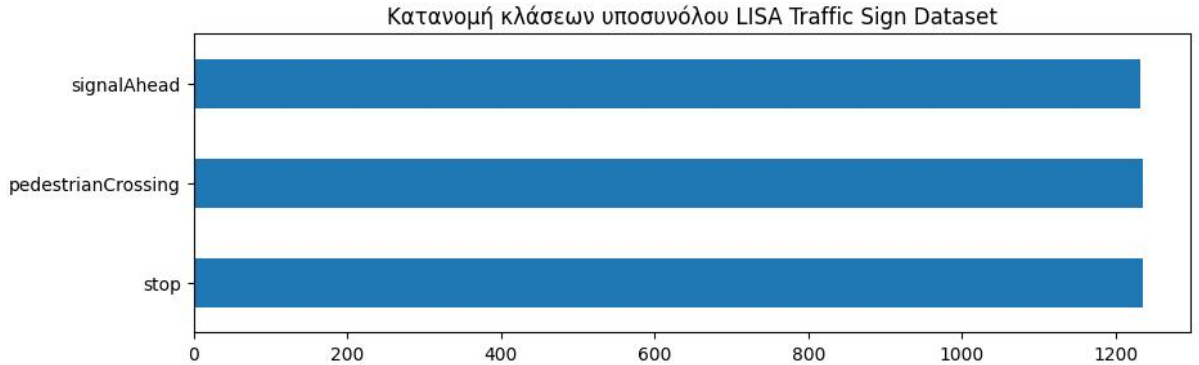
Εικόνα 5.2: Τυχαία ασπρόμαυρη εικόνα από το σύνολο δεδομένων Lisa Traffic Sign Dataset.

Το σύνολο δεδομένων περιέχει 7.855 εικόνες και για κάθε εικόνα τις συντεταγμένες των πλαισίων όπου υπάρχουν σήματα οδικής κυκλοφορίας και την κατηγορία τους, με 46 κλάσεις σημάτων με την κατανομή που φαίνεται στο Σχήμα 5.2.



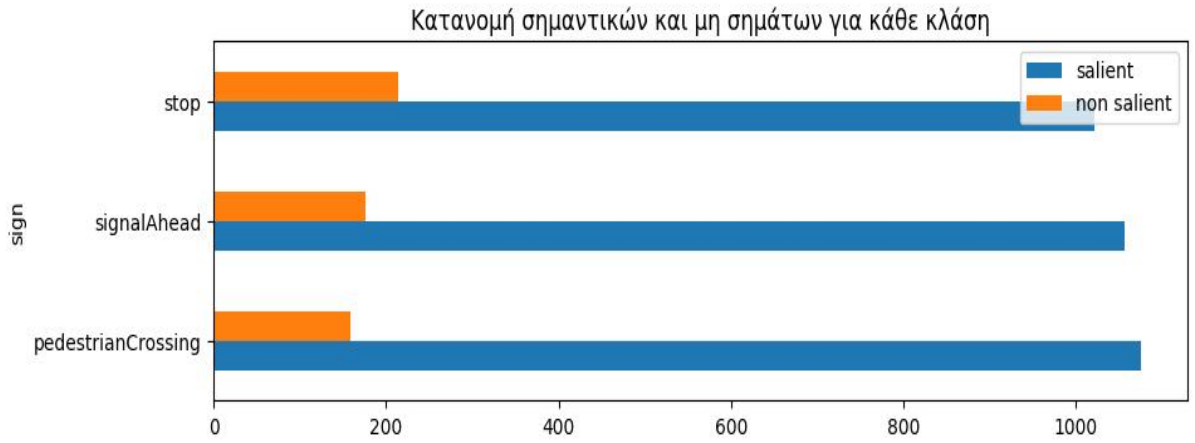
Σχήμα 5.2: Διάγραμμα με το πλήθος των δειγμάτων κάθε κλάσης στο σύνολο δεδομένων LISA Traffic Sign Dataset.

Για την δημιουργία του υποσυνόλου για την εκπαίδευση επιλέχθηκαν οι κλάσεις signalAhead, pedestrianCrossing και stop, οι οποίες είχαν τις μεγαλύτερες εμφανίσεις στα δεδομένα ώστε να προκύψει ένα υποσύνολο με ομοιόμορφη κατανομή των κλάσεων και αποφευχθούν προβλήματα κατά την εκπαίδευση. Διαχωρίστηκαν 3,703 σήματα σε 3,178 εικόνες οι οποίες περιέχουν έστω και μία από τις κλάσεις σημάτων που επιλέχθηκαν με την κατανομή που φαίνεται στο Σχήμα 5.3 και σε επόμενο βήμα ο περαιτέρω διαχωρισμός του συνόλου σε τρία υποσύνολο για τα στάδια του training, validation και test, με 3001, 405 και 297 σήματα αντίστοιχα.



Σχήμα 5.3: Διάγραμμα με το πλήθος των δειγμάτων κάθε κλάσης στο διαχωρισμένο υποσύνολο δεδομένων από το LISA Traffic Sign Dataset.

Στην συνέχεια δημιουργούμε μία νέα εκδοχή του υποσυνόλου που διαχωρίσαμε και το επαυξάνουμε με μία ακόμα παράμετρο (salience) [54] και για κάθε σήμα, όπου ελέγχουμε αν είναι άμεσα εφαρμόσιμο και του δίνουμε την τιμή 1 και 0 αν δεν είναι. Οι πλειονότητα των σημάτων στο σύνολο δεδομένων είναι σημαντικά όπως φαίνεται και στο Σχήμα 5.4, με αποτέλεσμα ίσως να καλύπτεται η διαφορά που θα μπορούσε να προκύψει με μία πιο ισομερή κατανομή.



Σχήμα 5.4: Διάγραμμα του πλήθους των δειγμάτων με την επιπρόσθετη παράμετρο σημαντικότητας (salience) στο διαχωρισμένο υποσύνολο δεδομένων από το LISA Traffic Sign Dataset.

5.4 Νευρωνικό δίκτυο

Το μοντέλο που επιλέχθηκε είναι το Deformable Detr ως μετεξέλιξη του DETR εφόσον πατάει πάνω στην αρχιτεκτονική του μειώνοντας την πολυπλοκότητα και βελτιώνοντας την ανίχνευση στα μικρά αντικείμενα καθώς χρησιμοποιεί μια παραλλαγή του μηχανισμού attention που επικεντρώνεται σε ένα υποσύνολο σημείων για να γίνει ανίχνευση αντικειμένων επιτυγχάνοντας πολύ καλά αποτελέσματα στο μεγάλο σύνολο δεδομένων COCO Dataset. Για την εκπαίδευση στα υποσύνολα δεδομένων που δημιουργήσαμε θα χρησιμοποιηθεί το Deformable Detr προεκπαιδευμενο στο σύνολο δεδομένων Coco Dataset, δηλαδή θα χρησιμοποιήσουμε την τεχνική transfer learning για χρησιμοποιήσουμε την ήδη υπάρχουσα γνώση που έχει το μοντέλο και να το εξειδικεύσουμε με τα μικρά σύνολα δεδομένων που δημιουργήσαμε. Χρησιμοποιήθηκαν οι προκαθορισμένες ρυθμίσεις του μοντέλου με το resnet50 προεκπαιδευμένο στο dataset imagenet για την εξαγωγή χαρακτηριστικών από την εικόνα, με μοναδική διαφοροποίηση σε μία από τις συναρτήσεις κόστους του Deformable Detr, την sigmoid focal loss, που φαίνεται στην σχέση (4.1).

$$\mathrm{FL}(\mathrm{p_t}) = -\,\alpha_{\mathrm{FL}}(1-\mathrm{p_t})^{\mathrm{y}}\log \mathrm{p_t} \tag{5.1}$$

Η σχέση τροποποιείται με βάση την έρευνα [54] για να ευνοεί τα σημαντικά σήματα κάθε περίπτωσης για το αυτόνομο όχημα. Η παραπάνω συνάρτηση κόστους που φαίνεται στην σχέση 4.1 επαυξάνεται με ένα ακόμα βάρος για τα σημαντικά σήματα όπως φαίνεται στην σχέση (4.2).

$$FL(d, p_t) = -\alpha_{FL} w_{SS}(d)(1 - p_t)^y \log p_t$$

(5.2) [54]

Όπου, $w_{SS}(d)$ με d το καλύτερο ανιχνευόμενο πλαίσιο στο οποίο αν το πραγματικό με το οποίο αντιστοιχίζεται αφορά σημαντικό σήμα του δίνουμε μία τιμή 2 αλλιώς παίρνει την τιμή 1, δηλαδή στην ουσία αφήνει την συνάρτηση κόστους αυτούσια. Στην έρευνα [54] αναφέρεται ότι η τιμή 4 έδωσε τα καλύτερα αποτελέσματα για τα σημαντικά σήματα όμως στην περίπτωση μας μετά από δοκιμές είχε ασταθή αποτελέσματα στο μικρότερο σύνολο δεδομένων που χρησιμοποιήθηκε.

5.5 Εκπαίδευση

Για την εκπαίδευση χρησιμοποιήθηκαν οι βιβλιοθήκες pytorch, pytorch lightning, για την καταγραφή των πειραμάτων ή εξαγωγή μετρικών οι βιβλιοθήκες wandb, fiftyone και η βιβλιοθήκη transformers από την ηλεκτρονική πλατφόρμα huggingface όπου υπάρχουν πολλά προ-εκπαιδευμένα μοντέλα και το συγκεκριμένο που θα χρησιμοποιήσουμε. Στο πρώτο στάδιο θα γίνει εκπαίδευση του μοντέλου χωρίς τροποποιήσεις στην συνάρτηση κόστους με τις παρακάτω παραμέτρους.

Υπερπαράμετροι εκπαίδευσης:

- **adamw learning rate:** 2e-5
- **adamw resnet50 learning rate:** 1e-6
- **adamw weight decay**: 1e-4
- **gradient clip value**: 1.0
- **batch size**: 2
- **batch gradients accumulation**: 4

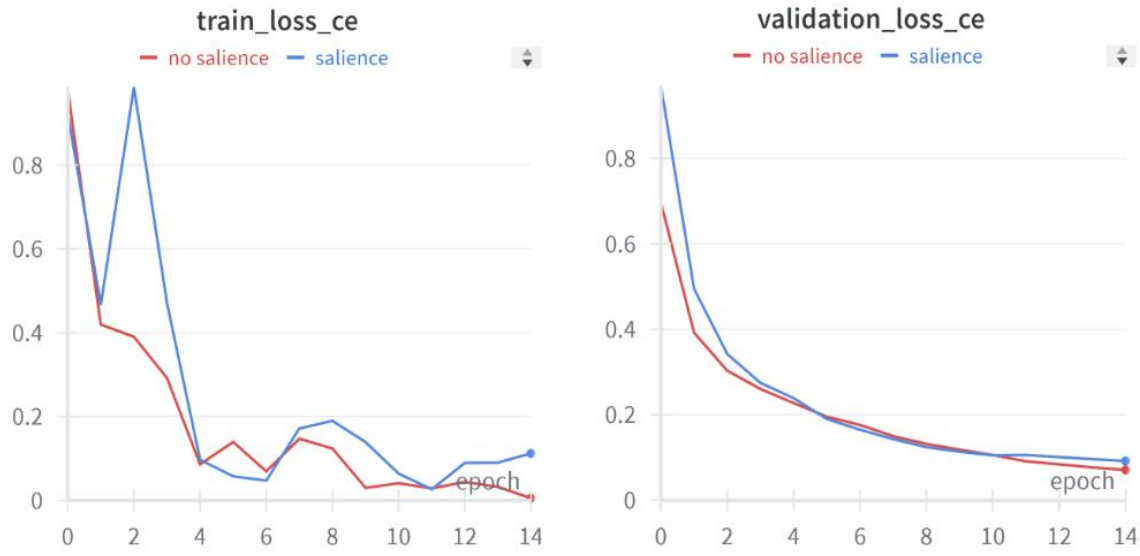
Όπου, adamw learning rate και resnet50 learning rate είναι ο ρυθμός εκπαίδευσης του transformer και του συνελικτικού που παράγει τα features από τις εικόνες, το weight decay και gradient clip value για περιορισμό του προβλήματος των τεράστιων παραγωγών που μπορεί να προκύψουν στην διαδικασία του back propagation και batch size με batch gradients accumulation γιατί το hardware όπου έγινε η εκπαίδευση δεν υποστήριζε μεγαλύτερο batch size για να "βλέπει" το δίκτυο περισσότερα δείγματα προτού προχωρήσει στην ενημέρωση των βαρών με το back propagation. Μετά από αρκετές δοκιμές φαίνεται να δίνουν καλά και σταθερά αποτελέσματα για να γίνει σωστά η σύγκριση που θέλουμε.

Κρατάμε το μοντέλο με το χαμηλότερο συνδυασμένο validation loss που προκύπτει από τις συναρτήσεις κόστους του μοντέλου για το υποσύνολο των δεδομένων που χρησιμοποιούμε για επαλήθευση.

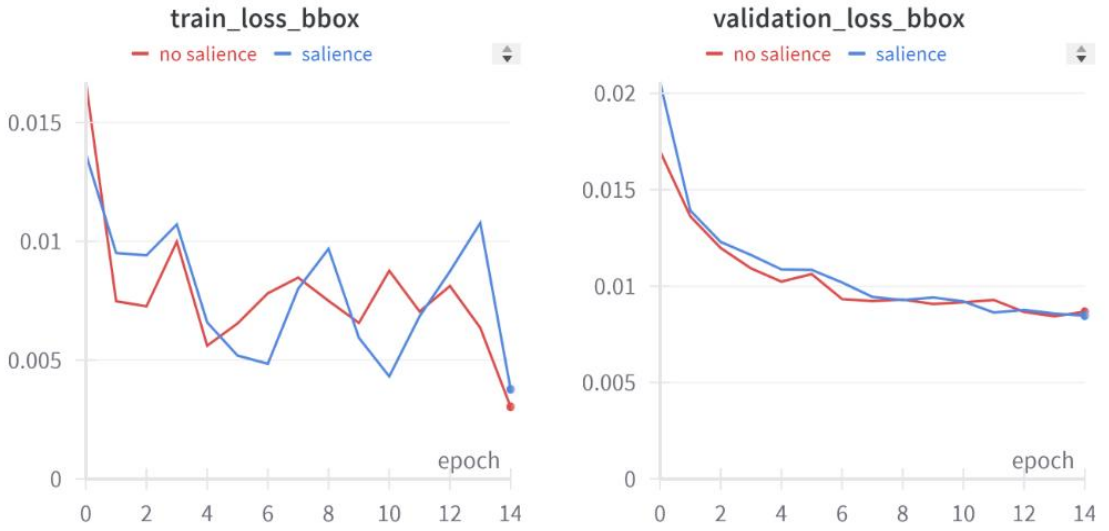
Σε επόμενο στάδιο χρησιμοποιούμε ακριβώς τις ίδιες παραμέτρους αλλά με την τροποποιημένη συνάρτηση κόστους sigmoid focal loss και κάνουμε ακριβώς την ίδια διαδικασία χωρίς να πειράζουμε το σύνολο δεδομένων με κανένα τρόπο και διατηρώντας την ίδια ακριβώς σειρά των δειγμάτων στο υποσύνολο εκπαίδευσης ώστε να έχουμε ακριβώς τις ίδιες συνθήκες εκπαίδευσης.

Κρατάμε πάλι το μοντέλο με το χαμηλότερο συνδυασμένο validation loss. Τα αποτελέσματα της εκπαίδευσης φαίνονται στα διαγράμματα 5.5-5.9.

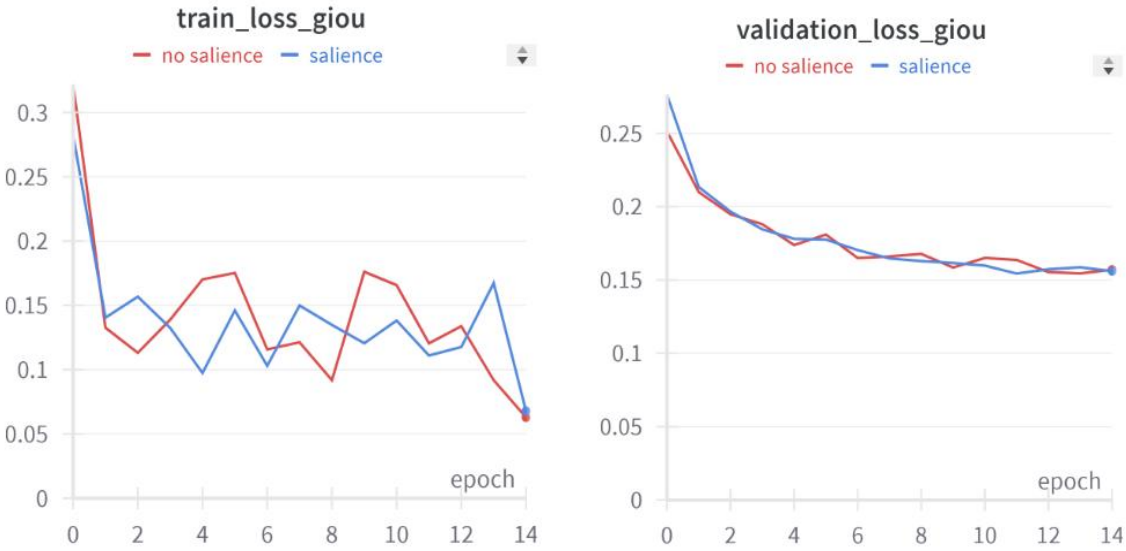
Σχήμα 5.5: Διαγράμματα με το ομαδοποιημένο σφάλμα κατηγοριοποιήσεις (loss_ce) και πλαισίων (loss_bbox και loss_giou) για κάθε εποχή εκπαίδευσης στο υποσύνολο εκπαίδευσης και στο επαλήθευσης αντίστοιχα.



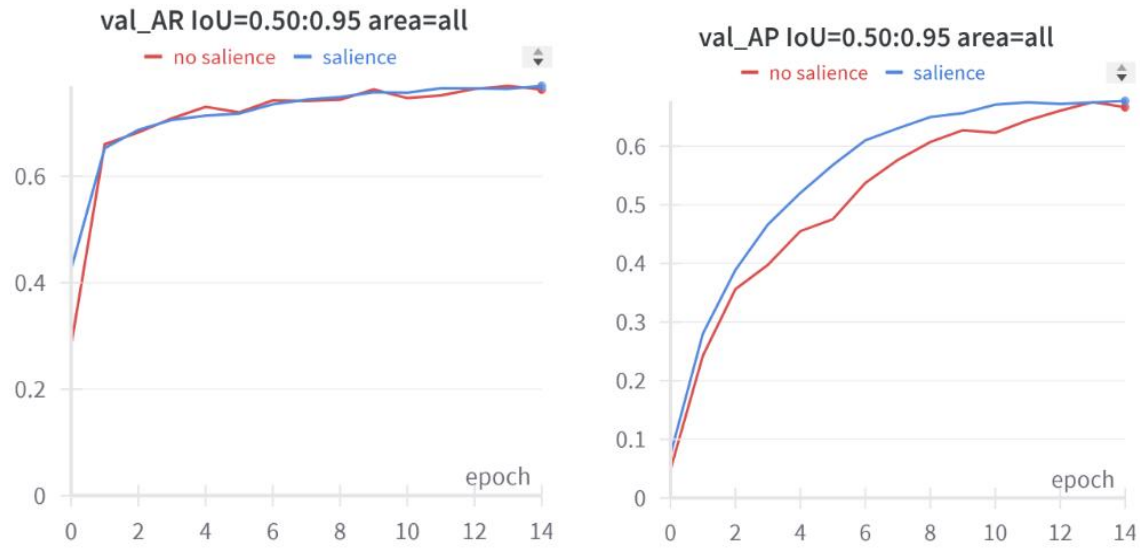
Σχήμα 5.6: Διαγράμματα με το σφάλμα κατηγοριοποιήσεις (loss_ce) για κάθε εποχή εκπαίδευσης στο υποσύνολο εκπαίδευσης και στο επαλήθευσης αντίστοιχα.



Σχήμα 5.7: Διαγράμματα με το σφάλμα πλαισίων (loss_bbox ή L1 loss) για κάθε εποχή εκπαίδευσης στο υποσύνολο εκπαίδευσης και στο επαλήθευσης αντίστοιχα.



Σχήμα 5.8: Διαγράμματα με το σφάλμα επικάλυψης πλαισίων (loss_giou) για κάθε εποχή εκπαίδευσης στο υποσύνολο εκπαίδευσης και στο επαλήθευσης αντίστοιχα.



Σχήμα 5.9: Διαγράμματα με την μέση ανάκληση (average recall) και την μέση ευστοχία (average precision) αντίστοιχα, για κατώφλια επικάλυψης πλαισίων από 0.50 έως 0.95, για όλα τα μεγέθη πλαισίων, με μέγιστο όριο ανιχνεύσεων ίσο με 100 για κάθε εποχή εκπαίδευσης στο υποσύνολο επαλήθευσης.

Εφαρμόζοντας το καλύτερο μοντέλο κάθε περιπτώσεις στο υποσύνολο δεδομένων που κρατήθηκε για testing προκύπτουν οι παρακάτω πίνακες 4.1 και 4.2. Ο πίνακας 5.2 δείχνει το συνδυασμένο σφάλμα των συναρτήσεων κόστους (test loss), το σφάλμα κατηγοριοποίησης (test loss ce), το σφάλμα πλαισίων (test loss bbox ή L1 loss) και το σφάλμα επικάλυψης πλαισίων αντίστοιχα για το καλύτερο μοντέλο των δύο περιπτώσεων χρήσης ή μη της ειδικής παραμέτρου σημαντικότητας (salience) στο υποσύνολο δεδομένων που διατηρήθηκε για testing. Ο πίνακας 5.3 μας δείχνει την μέση ανάκληση (average recall) και την μέση ευστοχία (average precision) αντίστοιχα, για κατώφλια επικάλυψης πλαισίων από 0.50 έως 0.95, για όλα τα μεγέθη πλαισίων, με μέγιστο όριο ανιχνεύσεων ίσο με 100 για το καλύτερο μοντέλο των δύο περιπτώσεων χρήσης ή μη της ειδικής παραμέτρου σημαντικότητας (salience) στο υποσύνολο δεδομένων που διατηρήθηκε για testing.

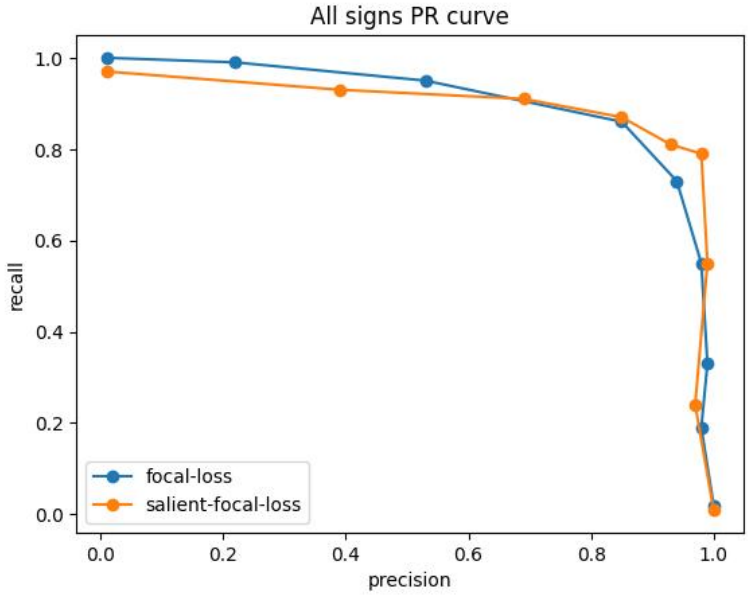
Πίνακας 5.2: Αποτελέσματα συναρτήσεων κόστους στο υποσυνόλου test.

	Without Salience	**With Salience**
loss	0.5379509925842285	0.534494936466217
loss ce	0.1626903563737869	0.161407351493835
loss bbox	0.0091726770624518	0.009119536727666
loss giou	0.1646986305713653	0.163744926452636

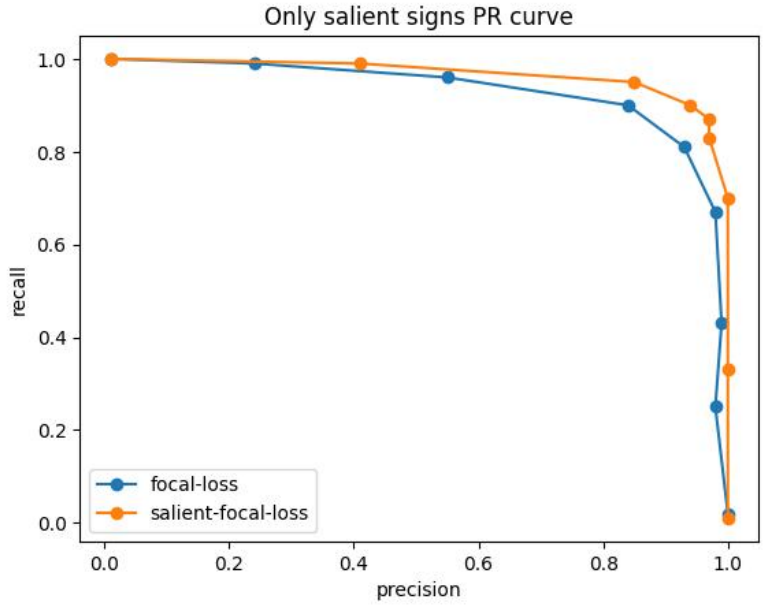
Πίνακας 5.3: Αποτελέσματα μέσης ανάκλησης και ευστοχίας για όλες τις κλάσεις και τα κατώφλια επικάλυψης (iou thresholds) από 0.5 έως 0.95.

	Without Salience	With Salience
Mean Average Precision	0.621	0.654
Mean Average Recall	0.630	0.648

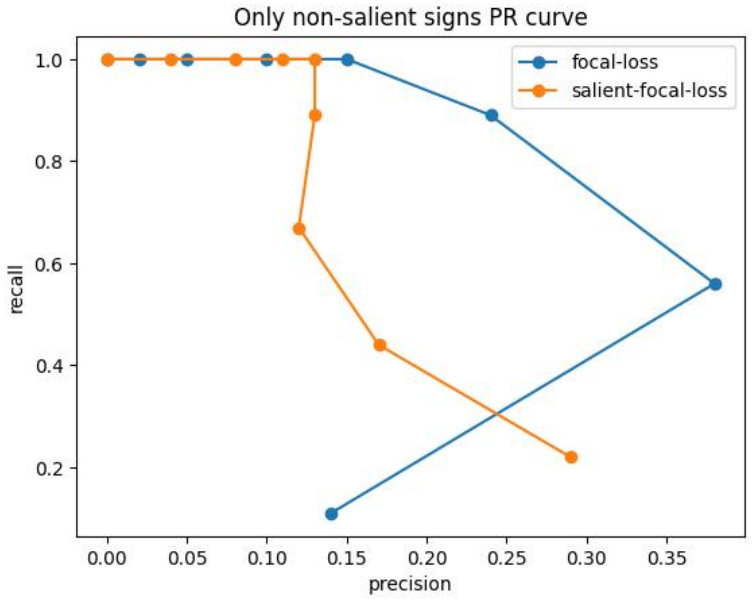
Στην συνέχεια τα γραφήματα 5.10-5.12 δείχνουν τα αποτελέσματα που έδωσε το μοντέλο για το υποσύνολο δεδομένων που διατηρήθηκε για testing σε μορφή καμπυλών με την μέση ανάκληση και ευστοχία όλων των κλάσεων για ένα εύρος από confidence thresholds από 0 έως 1 με βήμα 0.1 και σταματώντας αν υπάρχουν μηδενικές προβλέψεις.



Σχήμα 5.10: Διάγραμμα καμπυλών precision-recall με την μέση ανάκληση (average recall) και την μέση ευστοχία (average precision) όλων των κλάσεων για τα δύο μοντέλα, με την παράμετρο σημαντικότητας στην συνάρτηση κόστους (salient focal loss) και χωρίς (focal loss), χωρίς κατώφλι επικάλυψης πλαισίων, με μέγιστο όριο ανιχνεύσεων ίσο με 100 για το ένα εύρος από confidence thresholds από 0 έως 1 με βήμα 0.1 για ολόκληρο το υποσύνολο που διατηρήθηκε για testing.



Σχήμα 5.11: Διάγραμμα καμπυλών precision-recall με την μέση ανάκληση (average recall) και την μέση ευστοχία (average precision) όλων των κλάσεων για τα δύο μοντέλα, με την παράμετρο σημαντικότητας στην συνάρτηση κόστους (salient focal loss) και χωρίς (focal loss), χωρίς κατώφλι επικάλυψης πλαισίων, με μέγιστο όριο ανιχνεύσεων ίσο με 100 για το ένα εύρος από confidence thresholds από 0 έως 1 με βήμα 0.1 μόνο για τα δείγματα που είναι σημαντικά (δηλαδή έχουν θετική την παράμετρο salience) στο υποσύνολο που διατηρήθηκε για testing.



Σχήμα 5.12: Διάγραμμα καμπυλών precision-recall με την μέση ανάκληση (average recall) και την μέση ευστοχία (average precision) όλων των κλάσεων για τα δύο μοντέλα, με την παράμετρο σημαντικότητας στην συνάρτηση κόστους (salient focal loss) και χωρίς (focal loss), χωρίς κατώφλι επικάλυψης πλαισίων, με μέγιστο όριο ανιχνεύσεων ίσο με 100 για το ένα εύρος από confidence thresholds από 0 έως 1 με βήμα 0.1 μόνο για τα δείγματα που δεν είναι σημαντικά (δηλαδή έχουν αρνητική την παράμετρο salience) στο υποσύνολο που διατηρήθηκε για testing.

Κεφάλαιο 6ο: Συμπεράσματα και Προτάσεις βελτίωσης

6.1 Συμπεράσματα

Από τα αποτελέσματα φαίνεται πως το μοντέλο που χρησιμοποιεί την προσαρμοσμένη συνάρτηση κόστους διατηρεί και βελτιώνει γενικά τις προβλέψεις του, σε σχέση με το βασικό, βρίσκοντας περισσότερα σήματα και με υψηλότερη ακρίβεια, όπως φαίνεται στο γράφημα 5.10. Επιπλέον, παρατηρείται ότι είναι αρκετά καλύτερο στο να βρίσκει τα σημαντικά σήματα με εμφανείς βελτίωση στην καμπύλη precision-recall που φαίνεται στο γράφημα 5.11 και τέλος το μέρος του σφάλματος φαίνεται να έχει περάσει στα μη σημαντικά σήματα όπου το μοντέλο ήδη δυσκολεύεται αρκετά όπως φαίνεται στο γράφημα 5.12.

Ολοκληρώνοντας, με βάση τις παρατηρήσεις η μέθοδος φαίνεται να επιτυγχάνει την βελτίωση της ανίχνευσης των πιο άμεσα σημαντικών σημάτων. Χρησιμοποιώντας το μικρό επαυξημένο σύνολο δεδομένων με την ειδική παράμετρο σημαντικότητας (salience) που προστέθηκε, ενθαρρύνοντας την ανίχνευση των σημαντικών σημάτων στα οποία βασίζεται ένα αυτόνομο για να λάβει κρίσιμες αποφάσεις. Όμως, λαμβάνοντας υπόψη το μικρό μέγεθος των δεδομένων και την μη ομοιόμορφη κατανομή των δύο κλάσεων, σημαντικών και μη σημαντικών σημάτων, χρειάζεται περαιτέρω δοκιμές.

6.2 Προτάσεις Βελτίωσης

Βελτιώσεις μπορούν να περιλαμβάνουν συλλογή και δημιουργία ενός συνόλου δεδομένων με σκοπό να περιέχουν περισσότερες περιπτώσεις μη σημαντικών σημάτων και δύσκολων σεναρίων που θα μπορούσε να έρθει ένα οποιοδήποτε όχημα. Επιπλέον, θα μπορούσαν να προστεθούν παράμετροι για καταστάσεις που πιθανόν θα μπορούσαν να βοηθήσουν στην ανίχνευση, όπως την λωρίδα που βρίσκεται το όχημα και την επόμενη κίνηση που έχει σκοπό να κάνει, να συνεχίσει ευθεία, να στρίψει αριστερά και ούτω καθεξής. Σημαντική είναι επίσης και η βελτίωση της παράκαμψης των σημάτων που δεν έχουν άμεσο ενδιαφέρον για το όχημα και χρειάζεται ίσως με την εφαρμογή κάποιας μορφής penalty κατά την διάρκεια της εκπαίδευσης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία και Άρθρα

[1] Anderson, J. M., Nidhi, K., Stanley, K. D., Sorensen, P., Samaras, C., & Oluwatola, O. A. (2014). *Autonomous vehicle technology: A guide for policymakers.* Rand Corporation.

[2] Huang, Y., & Chen, Y. (2020, December). *Survey of state-of-art autonomous driving technologies with deep learning.* In 2020 IEEE 20th international conference on software quality, reliability and security companion (QRS-C) (pp. 221-228). IEEE

[3] Nilsson, N. J. (1998). *Artificial intelligence: a new synthesis*. Morgan Kaufmann.

[4] Καμπουρλάζος, Β., & Παπακώστας, Γ. (2015). *Εισαγωγή στην Υπολογιστική Νοημοσύνη. Αθήνα: ΣΕΑΒ.*

[5] Metehan, K. (2021) Supervised and Unsupervised Learning (an Intuitive Approach). Medium. Available at: https://medium.com/@metehankozan/supervised-and-unsupervised-learning-an-intuitive-approach-cd8f8f64b644

[6] Dan, L. (2019) Reinforcement Learning, Part 1: A Brief Introduction. Medium. Available at: https://medium.com/ai%C2%B3-theory-practice-business/reinforcement-learning-part-1-a-brief-introduction-a53a849771cf

[7] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning by Christopher M. Bishop*. Springer Science+ Business Media, LLC.

[8] Géron, A. (2022).” *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* " O'Reilly Media, Inc.".

[9] Διαμαντάρας, Κ., & Μπότσης, Δ. (2019). Μηχανική Μάθηση. Κλειδάριθμος, Αθήνα, Ιούλιος.

[10] A. Bhande, "What is underfitting and overfitting in machine learning and how to deal with it.," *GreyAtom,* 2018.

[11] Müller, A. C., & Guido, S. (2016). Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.".

[12] Bengio, Y., Goodfellow, I., & Courville, A. (2017). Deep learning (Vol. 1). Cambridge, MA, USA: MIT press.

[13] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

[14] Janai, J., Güney, F., Behl, A., & Geiger, A. (2020). Computer vision for autonomous vehicles: Problems, datasets and state of the art. Foundations and Trends® in Computer Graphics and Vision, 12(1–3), 1-308.

[15] Παπαδημητρίου, Α. Ι. (2011). Νευρωνικά δίκτυα και αναγνώριση προτύπων (Doctoral dissertation, University of Piraeus (Greece)).

[16] Saha, S. (2022) A comprehensive guide to Convolutional Neural Networks - the eli5 way, Medium.Towards Data Science. Available at: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[17] Namatēvs, I. (2017). Deep convolutional neural networks: Structure, feature extraction and training. Information Technology and Management Science, 20(1), 40-47.

[18] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). pmlr.

[19] Bui, T. D., Ravi, S., & Ramavajjala, V. (2018, February). Neural graph learning: Training neural networks using graphs. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (pp. 64-71).

[20] Pinheiro, P., & Collobert, R. (2014, January). Recurrent convolutional neural networks for scene labeling. In International conference on machine learning (pp. 82-90). PMLR

[21] .Lai, S., Xu, L., Liu, K., & Zhao, J. (2015, February). Recurrent convolutional neural networks for text classification. In Proceedings of the AAAI conference on artificial intelligence (Vol. 29, No. [21]Memory, L. S. T. (2010). Long short-term memory. Neural computation, 9(8), 1735-1780.

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). *Attention is all you need*. Advances in neural information processing systems, 30.

[23] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

[24] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, *29*.

[25] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6569-6578).

[26] Tian, Z., Shen, C., Chen, H., & He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9627-9636).

[27] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213-229). Cham: Springer International Publishing.

[28] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[29] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21-37). Springer International Publishing.

[30] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

[31] Crypto1 (2023) How does the gradient descent algorithm work in machine learning?, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/#h-what-is-gradient-descent.

[32] Amber, R. (2023) Binary Cross Entropy: Where To Use Log Loss In Model Monitoring. Arize.com Available at: https://arize.com/blog-course/binary-cross-entropy-log-loss/

[33] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).

[34] Brownlee, J. (2020) Understand the impact of learning rate on neural network performance, MachineLearningMastery.com. Available at: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/.

[35] Aditya, R. (2019) Understanding Learning Rate. Medium. Available at: https://towardsdatascience.com/https-medium-com-dashingaditya-rakhecha-understanding-learning-rate-dd5da26bb6de

[36] Sharma, S. (2022) Activation functions in neural networks, Medium. Available at: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[37] Paras, D. (2017) Softmax and Cross Entropy Loss. parasdahal.com. Available at: https://www.parasdahal.com/softmax-crossentropy

[38] Haji, S. H., & Abdulazeez, A. M. (2021). *Comparison of optimization techniques based on gradient descent algorithm: A review*. PalArch's Journal of Archaeology of Egypt/Egyptology, 18(4), 2715-2743.

[39] Ian, G., Yoshua, B., & Aaron, C. (2016). Deep learning (adaptive computation and machine learning series).

[40] Chen, J., & Kyrillidis, A. (2019). *Decaying momentum helps neural network training*.

[41] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[42] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

[43] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13* (pp. 740-755). Springer International Publishing.

[44] Python, "What is Python? Executive Summary | Python.org," Python. Available at: https://www.python.org/doc/essays/blurb/.

[45] S. L. Kinza Yasar, "What is PyTorch," Techtarget, Available at: https://www.techtarget.com/searchenterpriseai/definition/PyTorch.

[46] Pytorch Lightning, "About Lightning", lightning.ai, Available at: https://lightning.ai/about

[47] J. Terra, "Pytorch Vs Tensorflow Vs Keras: Here are the Difference You Should Know", Simplilearn, (2023). Available at: https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article.

[48] TensorFlow, "TensorFlow," tensorflow.org. Available at: https://www.tensorflow.org/.

[49] Keras, "Keras," keras.io. Available at: https://keras.io/

[50] J. Johnson, "TensorFlow vs PyTorch: Choosing Your ML Framework," BMC, (2022). Available at: https://www.bmc.com/blogs/tensorflow-vs-keras/.

[51] Weights and Biases, "About us", wandb.ai, Available at: https://wandb.ai/site/company/about-us

[52] F. Oh, "What Is CUDA | NVIDIA Official Blog," Nvidia, (2012). Available at:https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/.

[53] Møgelmose, A., Liu, D., & Trivedi, M. M. (2014, October). Traffic sign detection for us roads: Remaining challenges and a case for tracking. 17th International IEEE Conference on Intelligent Transportation Systems (ITSC) (pp. 1394-1399)

[54] Greer, R., Gopalkrishnan, A., Deo, N., Rangesh, A., & Trivedi, M. (2023). Salient sign detection in safe autonomous driving: Ai which reasons over full visual context. *arXiv preprint arXiv:2301.05804*.

Internet Site

[55] https://builtin.com/machine-learning/fully-connected-layer

[56] https://www.researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max_fig2_333593451

[57] https://www.researchgate.net/figure/The-folded-and-unfolded-structure-of-recurrent-neural-networks-1-RNN-Similar-to-a_fig5_341639694

[58] https://miro.medium.com/v2/resize:fit:4800/format:webp/1*Mb_L_slY9rjMr8-IADHvwg.png
[59] https://www.researchgate.net/figure/Recurrent-convolutional-neural-network-RCNN-model-Two-convolutional-layers-marked-by_fig4_332932587
[60] https://www.oreilly.com/api/v2/epubs/9781788397872/files/assets/34427a44-9fee-4bc3-813d-8a599f32a08c.png
[61] http://ronny.rest/media/blog/2017/2017_08_10_sigmoid/sigmoid_plot.jpg
[62] https://vidyasheela.com/web-contents/img/post_img/40/ReLU-activation-function-new.png
[63] https://themaverickmeerkat.com/img/softmax/softmax_2d.png

ΠΑΡΑΡΤΗΜΑ Α : Κώδικας Εργασίας

Ο κώδικας που χρησιμοποιήθηκε υπάρχει στο παρακάτω google colab notebook:
https://colab.research.google.com/drive/1gnOuMPGs5AcWOjZXj4a_UbRWllCZPnLc?usp=sharing