



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Π.Μ.Σ. «ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ & ΥΠΗΡΕΣΙΕΣ»
ΚΑΤΕΥΘΥΝΣΗ «ΜΕΓΑΛΑ ΔΕΔΟΜΕΝΑ & ΑΝΑΛΥΤΙΚΗ»

Διπλωματική Εργασία

*Μεταφορά Γνώσης με χρήση Νευρωνικών Δικτύων
και Συστήματα Συστάσεων*

Γιαννοπούλου Μαρία
ΑΜ: ΜΕ2006

ΕΠΙΒΛΕΠΟΥΣΑ ΚΑΘΗΓΗΤΡΙΑ: ΜΑΡΙΑ ΧΑΛΚΙΔΗ

Οκτώβριος 2022

Περίληψη

Τα συνελκτικα νευρωνικα δικτυα (CNN) είναι η ραχοκοκαλιά της κατηγοριοποίησης εικόνων, ένα φαινόμενο βαθιάς μάθησης που παίρνει μια εικόνα και της αποδίδει μια κλάση και μια ετικέτα που την κάνει μοναδική. Η ταξινόμηση εικόνων με χρήση CNN αποτελεί σημαντικό μέρος των πειραμάτων μηχανικής μάθησης.

Μαζί με τη χρήση των συνελκτικών νευρωνικών δικτύων και των επαγόμενων δυνατοτήτων του, χρησιμοποιείται πλέον ευρέως για μια σειρά εφαρμογών - από την προσθήκη ετικετών στο Facebook έως τις συστάσεις προϊόντων της Amazon και τις εικόνες υγειονομικής περίθαλψης έως τα αυτόματα αυτοκίνητα. Ο λόγος που τα ΣΝΔ είναι τόσο δημοφιλή είναι ότι απαιτούν πολύ λίγη προεπεξεργασία, που σημαίνει ότι μπορούν να διαβάσουν 2D εικόνες εφαρμόζοντας φίλτρα που άλλοι συμβατικοί αλγόριθμοι δεν μπορούν. Θα εμβαθύνουμε στη διαδικασία του πώς λειτουργεί η ταξινόμηση εικόνων χρησιμοποιώντας ΣΝΔ.

Στην παρούσα διπλωματική εργασία, ερευνούμε τις εφαρμογές της μεταφοράς γνώσης στο πεδίο των νευρωνικών δικτύων. Πιο συγκεκριμένα, προσπαθήσαμε να εφαρμόσουμε τεχνικές μεταφοράς γνώσης σε εργασίες κατηγοριοποίησης εικόνας και συστήματα συστάσεων. Για την υλοποίηση, συγκρίναμε προεκπαιδευμένα μοντέλα (VGG19, ResNet50, InceptionV3) με απλά συνελκτικά μοντέλα για να διαπιστώσουμε αν όντως η μεταφορά γνώσης πραγματικά βελτιώνει αισθητά την απόδοση των μοντέλων και το στόχο των εργασιών μας. Στο πεδίο της υπολογιστικής όρασης και πιο συγκεκριμένα της αναγνώρισης εικόνας βρίσκουμε αρκετά τέτοια αποτελέσματα όπου μπορούμε να καταλάβουμε ουσιαστικά τη βοήθεια που προσφέρει η μεταφορά γνώσης από τα προεκπαιδευμένα μοντέλα. Η είσοδος στον αλγόριθμό μας θα είναι μια εικόνα οποιουδήποτε αντικειμένου που ο πελάτης θέλει να αγοράσει. Οι δύο εργασίες που θέλουμε να επιτύχουμε είναι:

- η κατηγοριοποίηση της φωτογραφίας ενός προϊόντος, σε σχέση με προϊόντα που είναι πιθανότερο να μοιάζουν
- η σύσταση παρόμοιων προϊόντων με αυτό που έχουμε εισάγει

Ο στόχος μας είναι αφού εκτελέσουμε τα δύο διαφορετικά πειράματα, να συγκρίνουμε τα προεκπαιδευμένα μοντέλα, δηλαδή αυτά τα μοντέλα που έχουν ήδη εκπαιδευτεί στο σύνολο δεδομένων ImageNet, με αυτά που είναι μη προεκπαιδευμένα όσον αφορά την ακρίβεια και την απώλεια και εν τέλει στο πόσες "σωστές" προβλέψεις κάνουν και αν επιτυγχάνεται ο τελικός στόχος κάθε εργασίας.

Περιεχόμενα

Λίστα Πινάκων	1
1 Εισαγωγή	1
1.1 Σκοπός της Διπλωματικής Εργασίας	1
1.2 Διάρθρωση της Διπλωματικής Εργασίας	1
2 Θεωρητικό Υπόβαθρο-Βιβλιογραφική Ανασκόπηση	2
2.1 Μεταφορά Γνώσης	2
2.2 Συστήματα Συστάσεων	4
2.2.1 Είδη Συστημάτων Συστάσεων	5
2.2.2 Μέτρα Ομοιότητας στα Συστήματα Συστάσεων	6
2.3 Νευρωνικά Δίκτυα	7
2.4 Τύποι Νευρωνικών Δικτύων	9
2.5 Η Χρήση των Βοηθητικών Εργασιών στη Μεταφορά Γνώσης	21
2.5.1 SELAR Framework	23
2.6 Μεταφορά Γνώσης σε Νευρωνικά Δίκτυα Γραφών	26
2.6.1 Η Μέθοδος της Προεκπαίδευσης στα Νευρωνικά Δίκτυα Γράφων	26
3 Ορισμός Προβλήματος-Μεθοδολογία	30
3.1 Ορισμός Προβλήματος	30
3.2 Η Χρήση των Προεκπαιδευμένων Μοντέλων στη Μεταφορά Γνώσης	30
4 Ένα Απλό Παράδειγμα Μεταφοράς Γνώσης	36
4.1 Αποτελέσματα	37
4.1.1 Πείραμα 1ο - Δυαδική Κατηγοριοποίηση Εικόνας	37
4.1.2 Πείραμα 2ο: Content-Based Recommendation System	52
4.2 Συμπεράσματα	63
Λίστα Εικόνων και Διαγραμμάτων	65
Λίστα Πινάκων	67

Συμβολισμοί	68
Αναφορές	69

1 Εισαγωγή

1.1 Σκοπός της Διπλωματικής Εργασίας

Τα νευρωνικά δίκτυα έχουν αποδειχθεί αποτελεσματικά στην εκμάθηση αναπαραστάσεων για διάφορες εργασίες, όπως ταξινόμηση κόμβων, πρόβλεψη συνδέσμων, κατηγοριοποίηση εικόνας, συστήματα συστάσεων, φυσική και ανακάλυψη φαρμάκων. Στην παρούσα διπλωματική εργασία προσπαθήσαμε να υλοποιήσουμε εργασίες που αφορούν τα συστήματα συστάσεων και την κατηγοριοποίηση εικόνας με χρήση νευρωνικών δικτύων και με τη βοήθεια της μεταφοράς γνώσης και προεκπαιδευμένων μοντέλων. Η προεκπαίδευση βελτιώνει σημαντικά την εκμάθηση των αναπαραστάσεων για αυτό και συγκρίναμε την απόδοση των προεκπαιδευμένων μοντέλων σε σχέση με την απόδοση του απλού μοντέλου συνελκτικού νευρωνικού δικτύου χωρίς προεκπαίδευση. Αναμένουμε, ότι η χρήση των προεκπαιδευμένων μοντέλων να μας δώσει καλύτερα αποτελέσματα στην ακρίβεια και στην απώλεια αλλά και στην υλοποίηση της εργασίας συνολικά. Στο πειραματικό κομμάτι της εργασίας θα εκτελέσουμε δύο διαφορετικά πειράματα, ένα που αφορά την δυαδική κατηγοριοποίηση εικόνας και ένα που αφορά την κατασκευή ενός συστήματος συστάσεων και θα συγκρίνουμε τις αποδόσεις των μοντέλων.

1.2 Διάρθρωση της Διπλωματικής Εργασίας

Η παρούσα μεταπτυχιακή εργασία θα αναπτυχθεί σε τέσσερα συνολικά κεφάλαια. Η διάρθρωση της εργασίας είναι η εξής:

- Στο Κεφάλαιο 2, παρατίθεται το θεωρητικό υπόβαθρο και η βιβλιογραφική ανασκόπηση η οποία είναι απαραίτητη για την παρούσα εργασία.
- Στο Κεφάλαιο 3 πραγματοποιείται ο ορισμός του προβλήματος μας και επεξηγούνται οι βασικές μέθοδοι που θα ακολουθήσουμε. Πιο συγκεκριμένα, αναλύεται η χρήση των βοηθητικών εργασιών στη μεταφορά γνώσης και γίνεται αναφορά στη χρήση προεκπαιδευμένων μοντέλων ώστε να πετύχουμε μεταφορά γνώσης σε εργασίες που αφορούν την κατηγοριοποίηση εικόνας και τα συστήματα συστάσεων.
- Στο Κεφάλαιο 4 υλοποιούμε τα πειράματά μας και διατυπώνονται τα συμπεράσματα και τα αποτελέσματα που προκύπτουν από τη μελέτη καθώς και σύγκριση των μοντέλων που χρησιμοποιήθηκαν.

2 Θεωρητικό Υπόβαθρο-Βιβλιογραφική Ανασκόπηση

2.1 Μεταφορά Γνώσης

Αυτή η ενότητα περιέχει τους βασικούς ορισμούς και τις διάφορες κατηγορίες της μεταφοράς γνώσης (transfer learning), μερικές από τις πιο σημαντικές σχετικές μελέτες σε κάθε κατηγορία και μια ανασκόπηση σχετικά με το είδος των εφαρμογών για τις οποίες έχει χρησιμοποιηθεί. Πριν δώσουμε τον ορισμό της μεταφοράς γνώσης, είναι απαραίτητο να ορίσουμε την έννοια του τομέα (domain) και της εργασίας (task).

Ορισμός: Ένας τομέας (domain) D αποτελείται από δύο μέρη: έναν χώρο χαρακτηριστικών X και την οριακή κατανομή $P(X)$. Με άλλα λόγια, $D = X, P(X)$ και το σύμβολο X υποδηλώνει ένα σύνολο περιπτώσεων, που ορίζεται ως $X = \{x \mid x_i \in X, i = 1, \dots, n\}$.

Δεδομένου ενός συγκεκριμένου τομέα, $D = X, P(X)$, μια εργασία (task) αποτελείται από δύο στοιχεία: έναν χώρο ετικέτας Y και μία προγνωστική συνάρτηση f (σημειώνεται με $T = Y, f()$), η οποία δεν παρατηρείται αλλά μπορεί να μαθευτεί από τα εκπαιδευμένα δεδομένα, τα οποία αποτελούνται από ζεύγη $\{x_i, y_i\}$, όπου $x_i \in X$ και $y_i \in Y$. Η συνάρτηση $f()$ μπορεί να χρησιμοποιηθεί για να προβλέψουμε την αντίστοιχη ετικέτα (label) $f(x)$, μιας νέας περίπτωσης x . Από πιθανολογική άποψη, το $f(x)$ μπορεί να γραφτεί ως $P(y|x)$, το οποίο αντιπροσωπεύει τις υπό συνθήκη προβλεπόμενες κατανομές των περιπτώσεων. Δίνουμε τώρα έναν ενοποιημένο ορισμό της μεταφοράς γνώσης, όπως δίνεται στην [1].

Ορισμός: Δεδομένου ενός τομέα προέλευσης (source domain) D_s , μιας εργασίας μάθησης T_s , ενός τομέα στόχου (target domain) D_t και μιας εργασίας μάθησης T_t , η μεταφορά γνώσης προσπαθεί να βελτιώσει την εκμάθηση της προγνωστικής συνάρτησης του στόχου $f_t()$ χρησιμοποιώντας τη γνώση στο D_s και T_s , όπου $D_s \neq D_t$ ή $T_s \neq T_t$. Εφόσον $D = X, P(X)$, η συνθήκη $D_s \neq D_t$ υποδηλώνει ότι είτε $X_s \neq X_t$ είτε $P_s(X) \neq P_t(X)$. Ομοίως, εφόσον $T = (Y, P(Y|X))$, η συνθήκη $T_s \neq T_t$ υποδηλώνει ότι είτε $Y_s \neq Y_t$ είτε $P(Y_s|X_s) \neq P(Y_t|X_t)$.

Η διαδικασία της μεταφοράς γνώσης μπορεί να χωριστεί σε δύο κατηγορίες:

- στην **ομογενή (homogeneous)** μεταφορά γνώσης, όπου οι τομείς είναι του ίδιου χώρου χαρακτηριστικών και ορισμένες μελέτες υποθέτουν ότι οι τομείς διαφέρουν μόνο σε οριακές κατανομές. Για να επιλυθεί το πρόβλημα αυτό, προσαρμόζουν τους τομείς διορθώνοντας τη μεροληψία επιλογής δείγματος ή τη μετατόπιση συμμεταβλητών.
- και στην **ετερογενή (heterogeneous)** μεταφορά γνώσης, όπου οι τομείς έχουν διαφορετικούς χώρους χαρακτηριστικών και απαιτείται προσαρμογή χώρου χαρακτηριστικών.

Προχωρώντας στη κατηγοριοποίηση με βάση τις ετικέτες, έχουμε τρεις κατηγορίες:

- **Inductive Transfer Learning**, όπου τα δεδομένα με ετικέτα είναι διαθέσιμα στον τομέα στόχου, ενώ τα δεδομένα με ετικέτα ή τα δεδομένα χωρίς ετικέτα είναι δια-

θέσιμα στον τομέα προέλευσης. Σε αυτήν τη ρύθμιση, ο τομέας στόχου και ο τομέας προέλευσης διαφέρουν.

- **Transductive Transfer Learning**, όπου τα δεδομένα με ετικέτα είναι διαθέσιμα μόνο στον τομέα προέλευσης. Σε αυτή τη συνθήκη, ο τομέας είναι διαφορετικά.
- **Unsupervised Transfer Learning**, όπου δεν υπάρχουν διαθέσιμα δεδομένα με ετικέτα. Εκεί, οι εργασίες προέλευσης και στόχου είναι διαφορετικές αλλά σχετίζονται.

Στο πεδίο της μεταφοράς γνώσης ανακύπτουν τρία ερευνητικά θέματα: **Τι μπορεί να μεταφερθεί**. Αυτό το ερώτημα απαντάει στο τι μέρος της γνώσης μπορεί να μεταφερθεί κατά μήκος τομέων ή εργασιών. Κάποια γνώση είναι ειδικά για μεμονωμένους τομείς ή εργασίες και ορισμένες γνώσεις μπορεί να είναι κοινές μεταξύ διαφορετικών τομέων, έτσι ώστε μπορεί να βοηθήσουν στη βελτίωση της απόδοσης για τον τομέα-στόχο ή έργο. Αφού ανακαλύψουμε ποιες γνώσεις μπορούν να μεταφερθούν, αναπτύσσονται αλγόριθμοι μάθησης για τη μεταφορά της γνώσης, το οποίο απαντά στο ερώτημα **πώς να μεταφέρω γνώση**. Το ερώτημα **πότε μεταφέρουμε γνώση** ρωτά σε ποιες περιπτώσεις είναι απαραίτητες κάποιες ικανότητες μεταφοράς. Ομοίως, μας ενδιαφέρει να γνωρίζουμε σε ποιες καταστάσεις δεν πρέπει να μεταφέρεται η γνώση. Σε ορισμένες περιπτώσεις, όταν ο τομέας προέλευσης και ο τομέας στόχου δεν σχετίζονται μεταξύ τους, η μεταφορά γνώσης μπορεί να είναι ανεπιτυχής. Στη χειρότερη περίπτωση, μπορεί ακόμη και να βλάψει την απόδοση της μάθησης στο target domain, μια κατάσταση που συχνά αναφέρεται ως αρνητική μεταφορά, όπως έχουμε αναφέρει και παραπάνω. Οι περισσότερες εργασίες για τη μεταφορά γνώσης επικεντρώνονται στο «Τι να μεταφέρω» και «Πώς να μεταφέρω», υποθέτοντας σιωπηρά ότι οι τομείς προέλευσης και στόχος σχετίζονται μεταξύ τους. Με βάση την κατηγοριοποίηση στην οποία προχωρήσαμε παραπάνω, υπάρχουν διαφορετικές προσεγγίσεις ως προς τη μεταφορά γνώσης.

- **Instance Transfer**, η οποία υποθέτει ότι ορισμένα μέρη τα δεδομένα στο source domain μπορούν να επαναχρησιμοποιηθούν για εκμάθηση στο target domain με επαναστάθμιση.
- **Feature-Representation Transfer**, όπου μαθαίνουμε μια «καλή» αναπαράσταση χαρακτηριστικών για τον τομέα στόχου. Σε αυτήν την περίπτωση, η γνώση που θα μεταφερόταν μεταξύ των τομέων κωδικοποιείται τώρα στην αναπαράσταση χαρακτηριστικών που έχουν ήδη περάσει στο στάδιο της εκμάθησης. Με αυτή τη νέα αναπαράσταση χαρακτηριστικών, η απόδοση του τομέα στόχου αναμένεται να βελτιωθεί σημαντικά.
- **Parameter Transfer**, που υποθέτει ότι οι τομείς προέλευσης και τομέας στόχου μοιράζονται ορισμένες παραμέτρους ή προηγούμενες κατανομές των υπερπαραμέτρων των μοντέλων. Έτσι, η μεταφερόμενη γνώση κωδικοποιείται στις κοινές παραμέτρους ή προηγούμενες.
- **Relational Knowledge Transfer**. Η βασική υπόθεση πίσω από αυτό το πλαίσιο είναι ότι υπάρχει κάποια σχέση μεταξύ των δεδομένων στον τομέα προέλευσης και τομέα στόχου. Έτσι, η γνώση που πρέπει να μεταφερθεί είναι η σχέση μεταξύ των δεδομένων

2.2 Συστήματα Συστάσεων

Αυτό το κεφάλαιο ξεκινά με τους βασικούς ορισμούς και κατηγοριοποιήσεις σχετικά με τα συστήματα συστάσεων και συνεχίζει με τη χρήση της μεταφοράς γνώσης για συστήματα συστάσεων.

Τα συστήματα συστάσεων είναι συστήματα που έχουν σχεδιαστεί για να προτείνουν πράγματα στον χρήστη με βάση πολλούς διαφορετικούς παράγοντες. Αυτά τα συστήματα προβλέπουν το προϊόν που είναι πιο πιθανό να αγοράσουν οι χρήστες και αυτό για το οποίο ενδιαφέρονται πιο πολύ. Εταιρείες όπως το Netflix και η Amazon χρησιμοποιούν συστήματα συστάσεων για να βοηθήσουν τους χρήστες τους να εντοπίσουν το σωστό προϊόν ή ταινίες.

Τα συστήματα συστάσεων ασχολούνται με μεγάλο όγκο πληροφοριών φιλτράροντας τις πιο σημαντικές πληροφορίες με βάση δεδομένα που παρέχει ένας χρήστης και άλλους παράγοντες που έχουν να κάνουν με την προτίμηση και το ενδιαφέρον του χρήστη. Ουσιαστικά, ανακαλύπτουν την αντιστοιχία μεταξύ ενός χρήστη και ενός αντικειμένου και εντοπίζουν τις ομοιότητες μεταξύ χρηστών και στοιχείων για σύσταση. Υπάρχουν πολλοί διαφορετικοί τρόποι για τη δημιουργία συστημάτων συστάσεων. Μερικοί χρησιμοποιούν αλγοριθμικές και τυπικές προσεγγίσεις όπως ο αλγόριθμος Page Rank, ενώ άλλοι χρησιμοποιούν περισσότερο μοντελοποιημένες προσεγγίσεις, όπως collaborative filtering, content based, link prediction κ.λπ. Όλες αυτές οι προσεγγίσεις μπορεί να διαφέρουν σε πολυπλοκότητα, αλλά η πολυπλοκότητα δεν μεταφράζεται απαραίτητα σε «καλή» απόδοση. Συχνά απλές λύσεις και υλοποιήσεις αποδίδουν ισχυρότερα. Για παράδειγμα, μεγάλες εταιρείες όπως η Reddit, η Hacker News και η Google έχουν χρησιμοποιήσει απλές τυποποιημένες υλοποιήσεις μηχανών συστάσεων για την προώθηση περιεχομένου στην πλατφόρμα τους. Για να αξιολογήσουμε ένα σύστημα συστάσεων μας βοηθάει ο αριθμός των “καλών” συστάσεων. Η ποιότητα μιας σύστασης μπορεί να αξιολογηθεί μέσω διαφόρων τακτικών που μετρούν την κάλυψη και την ακρίβεια.

- **Η ακρίβεια (accuracy)** είναι το κλάσμα των σωστών συστάσεων προς τις συνολικές πιθανές συστάσεις. Τα κοινά μέτρα στατιστικής ακρίβειας για την αξιολόγηση της ακρίβειας ενός συστήματος είναι τα RMSD (Root Mean Square Deviation), MAE (Mean Absolute Error) και K-Fold Cross Validation.
- **Η κάλυψη (coverage)** είναι το ποσοστό των πιθανών συστάσεων που μπορεί να παράγει ένας εφαρμοσμένος αλγόριθμος συστάσεων. Αυτό το μέτρο είναι απαραίτητο από την άποψη του νέου στοιχείου που προτείνεται, γιατί δίνει μια ιδέα για το πόσο γρήγορα θα εμφανιστεί ένα νέο στοιχείο στη λίστα συστάσεων.

Η μέθοδος αξιολόγησης μιας σύστασης εξαρτάται αποκλειστικά από το σύνολο δεδομένων και την προσέγγιση που χρησιμοποιείται για τη δημιουργία της σύστασης. Τα συστήματα συστάσεων μοιράζονται αρκετές εννοιολογικές ομοιότητες με το πρόβλημα της μοντελοποίησης ταξινόμησης και παλινδρόμησης. Σε μια ιδανική κατάσταση, θα θέλαμε να δούμε πώς αντιδρούν οι πραγματικοί χρήστες στις προτάσεις και να παρακολουθούμε μετρήσεις γύρω από τον χρήστη η πρότασή μας να βελτιωθεί, ωστόσο, αυτό είναι αρκετά δύσκολο να επιτευχθεί.

2.2.1 Είδη Συστημάτων Συστάσεων

- 1. Collaborative Filtering Systems:** Τα Collaborative Filtering Systems είναι η διαδικασία πρόβλεψης των ενδιαφερόντων ενός χρήστη με τον εντοπισμό προτιμήσεων και πληροφοριών από πολλούς χρήστες. Αυτό γίνεται με το φιλτράρισμα δεδομένων για πληροφορίες ή μοτίβα χρησιμοποιώντας τεχνικές που περιλαμβάνουν τη συνεργασία μεταξύ πολλαπλών παραγόντων και πηγών δεδομένων. Η υποκείμενη διαίσθηση πίσω από το collaborative filtering είναι ότι εάν οι χρήστες A και B έχουν παρόμοιο γούστο σε ένα προϊόν, τότε οι A και B είναι πιθανό να έχουν παρόμοια γεύση και σε άλλα προϊόντα. Υπάρχουν δύο συνήθεις τύποι προσεγγίσεων στο collaborative filtering, η προσέγγιση βάσει μνήμης και η προσέγγιση βάσει μοντέλου.
 - **Προσέγγιση βάσει μνήμης (Neighbourhood Collaborative Filtering).** Ουσιαστικά, οι αξιολογήσεις (ratings) των συνδυασμών χρήστη-αντικειμένων προβλέπονται με βάση τις γειτονιές τους. Αυτό μπορεί να χωριστεί περαιτέρω σε collaborative filtering βάσει χρήστη και collaborative filtering βάσει στοιχείων. Βάσει χρήστη ουσιαστικά σημαίνει ότι οι χρήστες με παρόμοια σκέψη θα παράσχουν ισχυρές και παρόμοιες συστάσεις. Το collaborative filtering βάσει στοιχείων συνιστά στοιχεία με βάση την ομοιότητα μεταξύ των στοιχείων που υπολογίζονται με βάση τις αξιολογήσεις χρηστών αυτών των στοιχείων.
 - **Προσέγγιση βάσει μοντέλου.** Τα χαρακτηριστικά που σχετίζονται με το σύνολο δεδομένων παραμετροποιούνται ως είσοδοι του μοντέλου για την προσπάθεια επίλυσης ενός προβλήματος που σχετίζεται με τη βελτιστοποίηση. Οι προσεγγίσεις που βασίζονται σε μοντέλα περιλαμβάνουν τη χρήση πραγμάτων όπως δέντρα αποφάσεων, προσεγγίσεις βασισμένες σε κανόνες και μοντέλα λανθάνοντος παράγοντα.
- 2. Content Based Systems:** Τα Content Based Systems δημιουργούν προτάσεις με βάση τις προτιμήσεις και το προφίλ των χρηστών. Προσπαθούν να ταιριάξουν τους χρήστες με αντικείμενα που τους άρεσαν στο παρελθόν. Το επίπεδο ομοιότητας μεταξύ των στοιχείων καθορίζεται γενικά με βάση τα χαρακτηριστικά των αντικειμένων που αρέσουν στον χρήστη. Σε αντίθεση με τα περισσότερα μοντέλα collaborative filtering που αξιοποιούν τις αξιολογήσεις μεταξύ του χρήστη-στόχου και άλλων χρηστών, τα μοντέλα που βασίζονται στο περιεχόμενο επικεντρώνονται στις αξιολογήσεις που παρέχονται από τον ίδιο τον χρήστη-στόχο. Τα συστήματα βάσει περιεχομένου απαιτούν τις ακόλουθες πηγές δεδομένων:
 - **Πηγή δεδομένων σε επίπεδο στοιχείου:** χρειαζόμαστε μια πηγή δεδομένων που να σχετίζεται με τα χαρακτηριστικά του στοιχείου. Όσο περισσότερες πληροφορίες γνωρίζουμε σχετικά με το αντικείμενο, τόσο πιο ωφέλιμο θα είναι για το σύστημα.
 - **Πηγή δεδομένων σε επίπεδο χρήστη:** χρειαζόμαστε κάποιου είδους σχόλια από τους χρήστες με βάση το στοιχείο για το οποίο παρέχονται προτάσεις. Αυτό το επίπεδο ανατροφοδότησης μπορεί να είναι είτε σιωπηρό είτε ρητό. Όσο περισσότερα σχόλια των χρηστών, τόσο πιο ωφέλιμο θα είναι για το σύστημά σας.
- 3. Hybrid Method Systems:** Ένα υβριδικό σύστημα συστάσεων (hybrid method system) είναι ένας ειδικός τύπος συστήματος συστάσεων που μπορεί να θεωρηθεί ως

ο συνδυασμός της μεθόδου περιεχομένου (content based system) και συνεργατικού φιλτραρίσματος (collaborative filtering system). Οι προσεγγίσεις του συστήματος υβριδικών συστάσεων μπορούν να εφαρμοστούν με διάφορους τρόπους, όπως με τη χρήση μεθόδων που βασίζονται στο περιεχόμενο και τη συνεργασία για τη δημιουργία προβλέψεων ξεχωριστά και στη συνέχεια συνδυάζοντας την πρόβλεψη ή μπορούμε απλώς να προσθέσουμε τις δυνατότητες των μεθόδων που βασίζονται σε συνεργασία σε μια προσέγγιση που βασίζεται στο περιεχόμενο (και αντίστροφα).

2.2.2 Μέτρα Ομοιότητας στα Συστήματα Συστάσεων

Σε αυτή την ενότητα, περιγράφουμε τη θεωρητική βάση κάποιων μέτρων ομοιότητας που συχνά χρησιμοποιούμε. Για κάθε μέτρο παρέχουμε μια σύντομη περιγραφή και τον αντίστοιχο τύπο του:

- **Ομοιότητα Συνημιτόνου [53]:** Αυτή η τεχνική παρουσιάζει έναν χρήστη ως διάνυσμα με βαθμολογίες που βαθμολογήθηκαν από τον ίδιο και ένα στοιχείο ως διάνυσμα βαθμολογιών που βαθμολογήθηκαν από το σύνολο των χρηστών. Το συνημίτονο μεταξύ δύο διανυσμάτων που αντιπροσωπεύουν δύο χρήστες (ή στοιχεία) υποδηλώνει την ομοιότητα αξία μεταξύ τους. Μια τιμή κοντά στο 1 υποδηλώνει ότι υπάρχει ισχυρή συσχέτιση μεταξύ των δύο μεταβλητών. Μια τιμή κοντά στο 0 δείχνει ότι δεν υπάρχει συσχέτιση (ανεξάρτητες μεταβλητές).

$$\text{Cosine}(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{u \in I_u} r_{ui}^2} \sqrt{\sum_{v \in I_v} r_{vi}^2}}$$

όπου I_u και I_v δηλώνουν τα σύνολα αντικειμένων που βαθμολογήθηκαν από τους χρήστες u και v , αντίστοιχα, και το I_{uv} υποδηλώνει το σύνολο των αντικειμένων που συνήθως βαθμολογούνται από τόσο το u όσο και το v , r_{ui} και r_{vi} είναι οι τιμές βαθμολογιών στο στοιχείο που δίνω από χρήστες u και v , αντίστοιχα.

- **Ευκλείδεια Απόσταση:** Η Ευκλείδεια απόσταση που συμβολίζεται με d , από ένα χρήστη u σε ένα χρήστη v (ή από ένα στοιχείο i σε ένα στοιχείο j) είναι το μήκος του ευθυγράμμου τμήματος μεταξύ των δύο χρηστών (ή στοιχείων) στον Ευκλείδειο Χώρο. Πρακτικά, κάθε χρήστης αντιπροσωπεύεται από τις καρτεσιανές του συντεταγμένες σε σχέση με τη βάση των στοιχείων (το ίδιο για ένα στοιχείο που αντιπροσωπεύεται σε σχέση με τη βάση των χρηστών) και η απόσταση μεταξύ δύο χρηστών (ή δύο στοιχείων) είναι η απόλυτη τιμή της αριθμητικής διαφοράς των συντεταγμένων τους.

$$d(u, v) = \sqrt{\sum_{i \in I_{uv}} (r_{vi} - r_{ui})^2}$$

όπου το I_{uv} δηλώνει το σύνολο των αντικειμένων που συνήθως βαθμολογούνται από το u και το v . Τα r_{ui} και r_{vi} υποδηλώνουν τη βαθμολογία του χρήστη u και v , αντίστοιχα, για το στοιχείο i .

2.3 Νευρωνικά Δίκτυα

Τα νευρωνικά δίκτυα [2] και η βαθιά μάθηση είναι μεγάλα θέματα στην Επιστήμη των Υπολογιστών και στον κλάδο της τεχνολογίας, παρέχουν επί του παρόντος τις καλύτερες λύσεις σε πολλά προβλήματα στην αναγνώριση εικόνας, την αναγνώριση ομιλίας και την επεξεργασία φυσικής γλώσσας. Ο ορισμός ενός νευρωνικού δικτύου, πιο σωστά αναφερόμενος ως «τεχνητό» νευρωνικό δίκτυο (artificial neural network) (ANN), παρέχεται από τον εφευρέτη ενός από τους πρώτους νευροϋπολογιστές, τον Dr. Robert Hecht-Nielsen.

Ορισμός: Ένα νευρωνικό δίκτυο είναι ένα διασυνδεδεμένο συγκρότημα απλών στοιχείων επεξεργασίας, μονάδων ή κόμβων, των οποίων η λειτουργικότητα βασίζεται στον ζωικό νευρώνα. Η ικανότητα επεξεργασίας του δικτύου αποθηκεύεται στις δυνάμεις ή στα βάρη της σύνδεσης μεταξύ των μονάδων, που λαμβάνονται από μια διαδικασία προσαρμογής ή μάθησης από ένα σύνολο προτύπων εκπαίδευσης.

Από τον παραπάνω ορισμό, μπορούμε να συμπεράνουμε ότι ένα νευρωνικό δίκτυο αποτελείται από νευρώνες. Βιολογικά οι νευρώνες συνδέονται μέσω συνάψεων όπου ρέουν οι πληροφορίες (βάρη για το υπολογιστικό μοντέλο) και όταν εκπαιδεύουμε ένα νευρωνικό δίκτυο θέλουμε οι νευρώνες να ενεργοποιούνται όποτε μαθαίνουν συγκεκριμένα μοτίβα από τα δεδομένα και μοντελοποιούμε τον ρυθμό χρησιμοποιώντας μια συνάρτηση ενεργοποίησης.

Αν κάνουμε μια αντιστοίχιση της αρχιτεκτονικής που επικρατεί στο χώρο των νευρώνων του ανθρώπινου εγκεφάλου, τότε για τα νευρωνικά δίκτυα θα καταλήξουμε στο εξής: Τα τεχνητά ισοδύναμα των βιολογικών νευρώνων είναι οι κόμβοι (nodes) ή οι μονάδες (units) στο δικό μας ορισμό. Οι συνάψεις μοντελοποιούνται από έναν μόνο αριθμό ή βάρος (weight) έτσι ώστε κάθε είσοδος να πολλαπλασιάζεται με ένα βάρος πριν αποσταλεί στο ισοδύναμο του κυτταρικού σώματος.

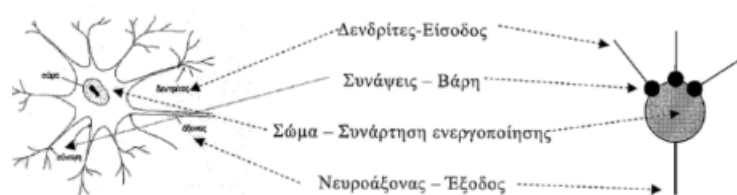


Figure 1: Αντιστοίχιση Βιολογικού με Τεχνητού Νευρώνα [3]

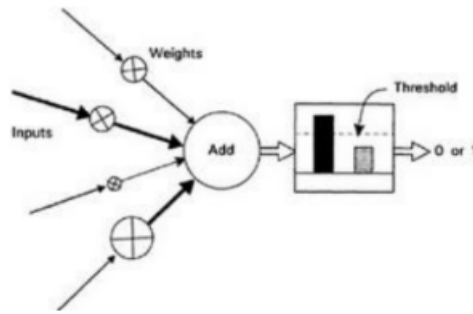


Figure 2: Μηχανισμός Τεχνητού Νευρώνα [3]

Οι τεχνητοί νευρώνες (perceptron) αποτελούνται από:

- **Input**
- **Weight**
- **Bias**
- **Activation Function**
- **Output**

Θα δούμε τι συμβαίνει σε κάθε στάδιο του δικτύου μας,

- **Input Nodes (Input Layer):** Δεν γίνεται κανένας υπολογισμός σε αυτό το επίπεδο, απλώς περνούν οι πληροφορίες στο επόμενο επίπεδο (hidden layer). Ένα σύνολο κόμβων ονομάζεται επίσης στρώμα.
- **Hidden Nodes (Hidden Layer):** Στα κρυφά επίπεδα γίνεται οι ενδιάμεσοι υπολογισμοί, και στη συνέχεια μεταφέρουν τα βάρη (σήματα ή πληροφορίες) από το επίπεδο εισόδου στο επόμενο επίπεδο (άλλο κρυφό επίπεδο ή στο επίπεδο εξόδου). Είναι δυνατό να έχουμε ένα νευρωνικό δίκτυο χωρίς κρυφό στρώμα.
- **Output Nodes (Output Layer):** Εδώ τελικά χρησιμοποιούμε μια συνάρτηση ενεργοποίησης (activation function) που αντιστοιχίζεται στην επιθυμητή μορφή εξόδου.
- **Connections and Weights:** Το δίκτυο αποτελείται από συνδέσεις, κάθε σύνδεση μεταφέρει την έξοδο ενός νευρώνα i στην είσοδο ενός νευρώνα j . Με αυτή την έννοια το i είναι ο προκάτοχος του j και το j είναι ο διάδοχος του i . Σε κάθε σύνδεση εκχωρείται ένα βάρος W_{ij} .
- **Activation Function:** Η συνάρτηση ενεργοποίησης ενός κόμβου καθορίζει την έξοδο αυτού του κόμβου με δεδομένη μια είσοδο ή ένα σύνολο εισόδων.
- **Learning Rule:** Ο κανόνας εκμάθησης είναι ένας κανόνας ή ένας αλγόριθμος που τροποποιεί τις παραμέτρους του νευρωνικού δικτύου, προκειμένου μια δεδομένη είσοδος στο δίκτυο να παράγει μια προτιμώμενη έξοδο. Αυτή η διαδικασία μάθησης συνήθως ισοδυναμεί με την τροποποίηση των βαρών και των ορίων.

Εκπαίδευση Νευρωνικού Δικτύου: Όταν ξεκινάμε με το νευρωνικό μας δίκτυο, αρχικοποιούμε τα βάρη μας τυχαία. Στη διαδικασία της εκπαίδευσης, θέλουμε να ξεκινήσουμε με ένα νευρωνικό δίκτυο κακής απόδοσης και να ολοκληρώσουμε το δίκτυο με υψηλή ακρίβεια. Όσον αφορά τη συνάρτηση απώλειας, θέλουμε η συνάρτηση απώλειας μας να είναι πολύ χαμηλότερη στο τέλος της εκπαίδευσης. Η συνάρτηση απώλειας είναι μια συνάρτηση που μας λέει πόσο καλό είναι το νευρωνικό μας δίκτυο για μια συγκεκριμένη εργασία.

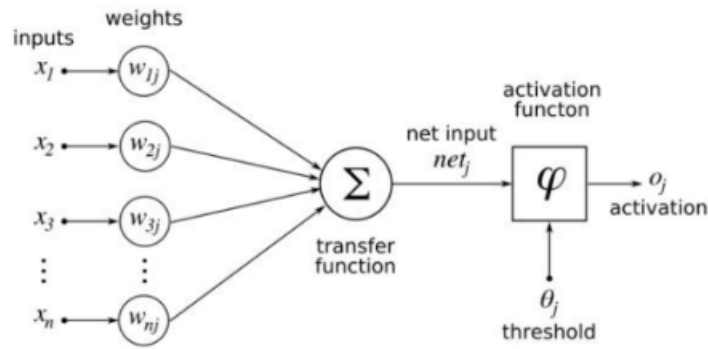


Figure 3: Νευρωνικό Δίκτυο [57]

Ορίζουμε σαν συνάρτηση απώλειας,

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

όπου y αντιπροσωπεύει τον αριθμό που θέλουμε να πάρουμε από το δίκτυο, ενώ \hat{y} ο αριθμός που πήραμε πραγματικά περνώντας το παράδειγμά μας μέσω του δικτύου. Η βελτίωση του δικτύου είναι δυνατή, γιατί μπορούμε να αλλάξουμε τη λειτουργία του ρυθμίζοντας τα βάρη. Θέλουμε να βρούμε μια άλλη συνάρτηση που να αποδίδει καλύτερα από την αρχική. Το πρόβλημα της εκπαίδευσης είναι ισοδύναμο με το πρόβλημα της ελαχιστοποίησης της συνάρτησης απώλειας. Υπάρχουν πολλοί αλγόριθμοι που βελτιστοποιούν τις συναρτήσεις. Αυτοί οι αλγόριθμοι μπορούν να βασίζονται σε κλίση ή όχι, με την έννοια ότι δεν χρησιμοποιούν μόνο τις πληροφορίες που παρέχονται από τη συνάρτηση, αλλά και από τη διαβάθμιση της.

Στην πραγματικότητα, η εκπαίδευση ενός νευρωνικού δικτύου σημαίνει την εύρεση των κατάλληλων βαρών των νευρωνικών συνδέσεων καθώς όλοι οι νευρώνες ενός δεδομένου στρώματος παράγουν μια έξοδο, αλλά δεν έχουν το ίδιο βάρος για το επόμενο στρώμα νευρώνων. Αυτό σημαίνει ότι εάν ένας νευρώνας σε ένα στρώμα παρατηρήσει ένα δεδομένο μοτίβο μπορεί να σημαίνει λιγότερο για τη συνολική εικόνα και θα είναι μερικώς ή πλήρως σε σίγαση. Με άλλα λόγια ένα μεγάλο βάρος σημαίνει ότι η είσοδος είναι σημαντική και φυσικά ένα μικρό βάρος σημαίνει ότι πρέπει να το αγνοήσουμε. Κάθε νευρωνική σύνδεση μεταξύ των νευρώνων θα έχει ένα σχετικό βάρος. Σε όλες τις περιπτώσεις όταν το δίκτυο σταματάει να αλλάζει τις τιμές των βαρών, τότε θεωρούμε ότι η εκπαίδευση έχει επιτευχθεί. Αυτό συμβαίνει επειδή το λάθος στην έξοδο γίνεται μηδέν ή είναι πολύ κοντά (τείνει) στο μηδέν.

2.4 Τύποι Νευρωνικών Δικτύων

Feedforward Neural Network: Ένα νευρωνικό δίκτυο τροφοδοσίας [2] είναι ένα τεχνητό νευρωνικό δίκτυο όπου οι συνδέσεις μεταξύ των μονάδων δεν σχηματίζουν κύκλο. Σε

αυτό το δίκτυο, οι πληροφορίες κινούνται προς μία μόνο κατεύθυνση, προς τα εμπρός, από τους κόμβους εισόδου, μέσω των κρυφών κόμβων (εάν υπάρχουν) και στους κόμβους εξόδου. Δεν υπάρχουν κύκλοι ή βρόχοι στο δίκτυο.

- **Single Layer Perceptron:** Αυτό είναι το απλούστερο νευρωνικό δίκτυο τροφοδοσίας και δεν περιέχει κανένα κρυφό επίπεδο, πράγμα που σημαίνει ότι αποτελείται μόνο από ένα μόνο στρώμα κόμβων εξόδου. Λέγεται ότι είναι απλό, επειδή όταν μετράμε τα επίπεδα δεν συμπεριλαμβάνουμε το επίπεδο εισόδου, ο λόγος για αυτό είναι επειδή στο επίπεδο εισόδου δεν γίνονται υπολογισμοί, οι εισοδοί τροφοδοτούνται απευθείας στις εξόδους μέσω μιας σειράς βαρών.
- **Multi-Layer Perceptron:** Αυτή η κατηγορία δικτύων αποτελείται από πολλαπλά υπολογιστικά επίπεδα. Κάθε νευρώνας σε ένα στρώμα συνδέεται με τους νευρώνες του επόμενου στρώματος. Τα MLP είναι πολύ πιο χρήσιμα και ένας καλός λόγος είναι ότι, μπορούν να μάθουν μη γραμμικές αναπαραστάσεις (τις περισσότερες φορές τα δεδομένα που μας παρουσιάζονται δεν είναι γραμμικά διαχωρισμένα).

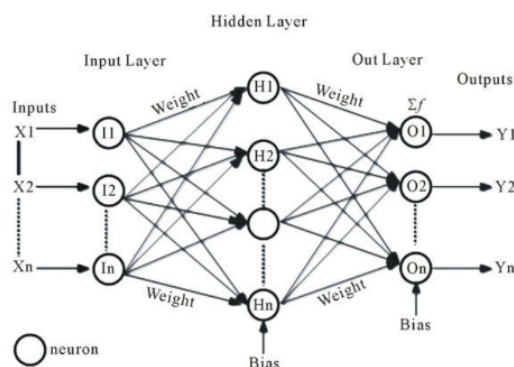


Figure 4: Multi-Layer Perceptron

- **Convolutional Neural Network (CNN):** Τα συνελικτικά νευρωνικά δίκτυα (ΣΝΔ) μοιάζουν πολύ με τα συνηθισμένα νευρωνικά δίκτυα, αποτελούνται και αυτά από νευρώνες που έχουν βάρη. Στο συνελικτικό νευρωνικό δίκτυο (CNN, ή ConvNet ή αναλλοίωτο μετατόπισης ή αναλλοίωτο χώρου) το μοτίβο συνδεσιμότητας μονάδας εμπνέεται από την οργάνωση του οπτικού φλοιού, οι μονάδες ανταποκρίνονται σε ερεθίσματα σε μια περιορισμένη περιοχή του χώρου, γνωστή ως δεκτικό πεδίο. Τα δεκτικά πεδία επικαλύπτονται εν μέρει, καλύπτοντας όλο το οπτικό πεδίο. Η απόκριση μονάδας μπορεί να προσεγγιστεί μαθηματικά με μια λειτουργία συνέλιξης. Οι ευρείες εφαρμογές τους είναι στην αναγνώριση εικόνας και βίντεο, συστήματα συστάσεων και επεξεργασία φυσικής γλώσσας. Τα CNN (ΣΝΔ) απαιτούν μεγάλα δεδομένα για εκπαίδευση. Κάθε επίπεδο ενός ΣΝΔ μετασχηματίζει τον τρισδιάστατο όγκο εισόδου (input volume), σε έναν τρισδιάστατο όγκο εξόδου (output volume), που παράγεται από τις ενεργοποιήσεις των νευρώνων του. Τα διαφορετικά επίπεδα του συνελικτικού νευρωνικού δικτύου που χρησιμοποιούνται είναι:

- **Επίπεδο εισόδου:** Το πρώτο επίπεδο κάθε CNN που χρησιμοποιείται είναι το «επίπεδο εισόδου» που λαμβάνει εικόνες, αλλάζει το μέγεθός τους για μεταβίβαση περαιτέρω στρώματα για εξαγωγή χαρακτηριστικών.
- **Επίπεδο συνέλιξης:** Τα επόμενα επίπεδα είναι «επίπεδα συνέλιξης» που λειτουργούν ως φίλτρα για εικόνες, επομένως βρίσκουμε και εξάγουμε χαρακτηριστικά από εικόνες και χρησιμοποιείται επίσης για τον υπολογισμό των κοινών χαρακτηριστικών κατά τη διάρκεια της δοκιμής.
- **Επίπεδο συγκέντρωσης:** Τα εξαγόμενα σύνολα χαρακτηριστικών μεταβιβάζονται στη συνέχεια στο «επίπεδο συγκέντρωσης». Αυτό το στρώμα λαμβάνει μεγάλες εικόνες και τις συρρικνώνει, διατηρώντας παράλληλα τις πιο σημαντικές πληροφορίες σε αυτά. Διατηρεί τη μέγιστη τιμή από κάθε παράθυρο, διατηρεί τις καλύτερες προσαρμογές κάθε χαρακτηριστικού μέσα στο παράθυρο.
- **Rectified Linear Unit Layer:** Το επόμενο επίπεδο «Rectified Linear Unit» ή ReLU ανταλλάσσει κάθε αρνητικό αριθμό του επιπέδου συγκέντρωσης με 0. Αυτό βοηθά το CNN να παραμείνει μαθηματικά σταθερό, εμποδίζοντας τις μαθημένες τιμές να κολλήσουν κοντά στο 0 ή να φουσκώνουν προς το άπειρο.
- **Πλήρως συνδεδεμένο επίπεδο:** Το τελικό επίπεδο είναι το πλήρως συνδεδεμένο επίπεδο που λαμβάνει τις φιλτραρισμένες εικόνες υψηλού επιπέδου και τις μεταφράζει σε κατηγορίες με ετικέτες

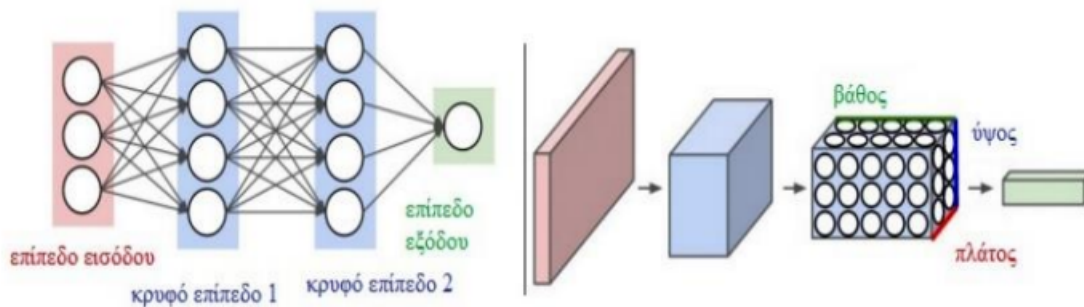


Figure 5: Σύγκριση ενός κλασικού Νευρωνικού Δικτύου με ένα CNN

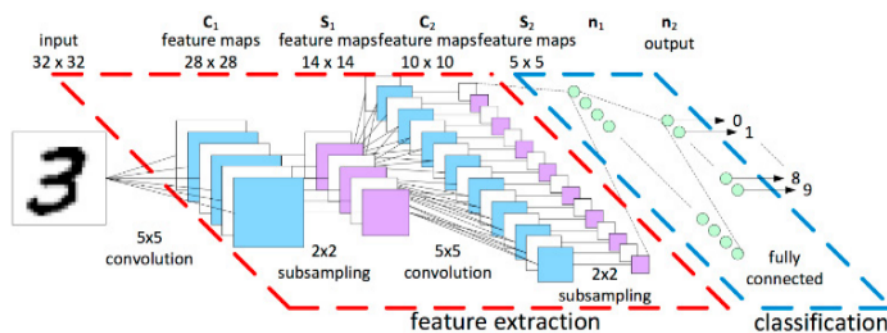


Figure 6: Επίπεδα Συνελικτικού Νευρωνικού Δικτύου [58]

Recurrent Neural Networks: Στο επαναλαμβανόμενο νευρωνικό δίκτυο (RNN), οι συνδέσεις μεταξύ των μονάδων σχηματίζουν έναν κατευθυνόμενο κύκλο (διαδίδουν δεδομένα προς τα εμπρός, αλλά και προς τα πίσω, από τα μεταγενέστερα στάδια επεξεργασίας στα

προηγούμενα στάδια). Αυτό του επιτρέπει να επιδεικνύει δυναμική χρονική συμπεριφορά. Σε αντίθεση με τα νευρωνικά δίκτυα τροφοδοσίας, τα RNN μπορούν να χρησιμοποιήσουν την εσωτερική τους μνήμη για να επεξεργαστούν αυθαίρετες ακολουθίες εισόδων. Αυτό τα καθιστά εφαρμόσιμα σε εργασίες όπως η μη τμηματοποιημένη, η συνδεδεμένη αναγνώριση χειρογράφου, η αναγνώριση ομιλίας και άλλοι γενικοί επεξεργαστές ακολουθίας.

Νευρωνικά Δίκτυα Γράφων Πριν προχωρήσουμε στα νευρωνικά δίκτυα γράφων, θα ορίσουμε το γράφο. Ο γράφος [6] είναι μία δομή δεδομένων που αποτελείται από δύο συστατικά, τις κορυφές και τις ακμές. Ένας γράφος G , δηλαδή περιγράφεται από ένα σύνολο κορυφών V και ακμών E . Πιο συγκεκριμένα,

$$G = (V, E)$$

Οι ακμές μπορεί να είναι κατευθυνόμενες ή μη κατευθυνόμενες. Συνήθως τις κορυφές τις αποκαλούμε κόμβους.

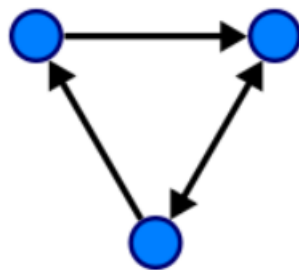


Figure 7: Κατευθυνόμενος Γράφος

Τα νευρωνικά δίκτυα γράφων είναι τύποι νευρωνικών δικτύων που εμφανίζονται με τη δομή γραφημάτων.

Τυπικές εφαρμογές των νευρωνικών δικτύων γράφων είναι η ταξινόμηση κόμβων και η πρόβλεψη συνδέσμου. Σε ένα γράφο, κάθε κόμβος ορίζεται φυσικά από τα χαρακτηριστικά του και τους σχετικούς κόμβους. Ο στόχος των νευρωνικών δικτύων γράφων είναι να μάθουν να αναπαριστούν κάθε κόμβο υπό τη μορφή μιας ενσωμάτωσης h_v , η οποία περιέχει πληροφορίες για την γειτονιά του κόμβου. Πιο συγκεκριμένα,

$$h_v = f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]})$$

$$o_v = g(h_v, x_v)$$

όπου $x_{co[v]}$ υποδηλώνει τα χαρακτηριστικά των ακμών που συνδέονται με v , $h_{ne[v]}$ υποδηλώνει την ενσωμάτωση των γειτονικών κόμβων του v και $x_{ne[v]}$ υποδηλώνει τα χαρακτηριστικά των γειτονικών κόμβων του v . Η f είναι η συνάρτηση μετάβασης (transition function) που προβάλλει αυτές τις εισόδους σε ένα d -διάστατο χώρο. Εστω τα H, O, X και X_N να είναι τα διανύσματα που κατασκευάζονται με τη συνένωση όλων των καταστάσεων, όλων των εξόδων, όλων των χαρακτηριστικών και όλων των χαρακτηριστικών του κόμβου, αντίστοιχα. Στη συνέχεια έχουμε την παρακάτω μορφή:

$$H = F(H, X)$$

$$O = G(H, X_N)$$

όπου F , η global συνάρτηση μετάβασης, και G , η παγκόσμια συνάρτηση εξόδου είναι στοιβαγμένες εκδόσεις των f και g για όλους τους κόμβους σε ένα γράφο αντίστοιχα. Η τιμή του H είναι το σταθερό σημείο και ορίζεται μοναδικά με την υπόθεση ότι το F είναι ένας χάρτης συστολής (contraction map).

Εφ'όσον ψάχνουμε μία μοναδικά αναπαράσταση για το h_v , η παραπάνω εξίσωση γράφεται μετά από επανάληψεις ως,

$$H^{t+1} = F(H^t, X)$$

όπου H^t δηλώνει την t -οστή επανάληψη του H .

Οι υπολογισμοί που περιγράφονται στα f και g μπορούν να ερμηνευθούν ως εμπρός τροφοδότηση. Όταν έχουμε το λειτουργικό πλαίσιο του ΝΔΓ, η επόμενη ερώτηση είναι πώς να μάθουμε τις παραμέτρους των f και g . Με τις πληροφορίες στόχου (t_v για έναν συγκεκριμένο κόμβο) για την επίβλεψη, η απώλεια μπορεί να γραφτεί ως εξής:

$$loss = \sum_{i=1}^p (t_i - o_i)$$

όπου p είναι ο αριθμός των εποπτευόμενων κόμβων. Η απώλεια μπορεί να βελτιστοποιηθεί μέσω του αλγόριθμου διαβάθμισης-κλίσης.

Όπως περιγράφεται και στο [4], υπάρχουν κάποιοι περιορισμοί σε αυτή την εκδοχή των νευρωνικών δικτύων γράφων.

- Εάν η υπόθεση του "σταθερού σημείου" είναι χαλαρή, είναι δυνατό να αξιοποιηθεί το Multi-layer Perceptron για να μάθουμε μια πιο σταθερή αναπαράσταση και να καταργήσουμε τη διαδικασία επαναληπτικής ενημέρωσης. Αυτό συμβαίνει επειδή, στην αρχική πρόταση, διαφορετικές επαναλήψεις χρησιμοποιούν τις ίδιες παραμέτρους της συνάρτησης μετάβασης f , ενώ οι διαφορετικές παράμετροι σε διαφορετικά επίπεδα του MLP επιτρέπουν την ιεραρχική εξαγωγή χαρακτηριστικών.
- Δεν μπορεί να επεξεργαστεί πληροφορίες ακμών (π.χ. διαφορετικές ακμές σε ένα γράφημα γνώσης μπορεί να υποδηλώνουν διαφορετική σχέση μεταξύ κόμβων)
- Το σταθερό σημείο μπορεί να αποθαρρύνει τη διαφοροποίηση της κατανομής κόμβων, και επομένως μπορεί να μην είναι κατάλληλο για εκμάθηση αναπαράστασης κόμβων.

Είδη Νευρωνικών Δικτύων Γράφων Τα νευρωνικά δίκτυα γραφημάτων (GNN) όπως είδαμε και παραπάνω είναι μια προσπάθεια εφαρμογής τεχνικών βαθιάς μάθησης σε γραφήματα. Ο όρος GNN αναφέρεται συνήθως σε μια ποικιλία διαφορετικών αλγορίθμων και όχι σε μια ενιαία αρχιτεκτονική. Όπως θα δούμε, μια πληθώρα διαφορετικών αρχιτεκτονικών έχουν αναπτυχθεί όλα αυτά τα χρόνια. Παρατίθεται ένα διάγραμμα [4] που παρουσιάζει τις πιο σημαντικές εργασίες στο πεδίο.

Η βασική ιδέα πίσω από τις περισσότερες αρχιτεκτονικές GNN είναι η συνέλιξη γραφήματος. Στην ουσία προσπαθούμε να γενικεύσουμε την ιδέα της συνέλιξης σε γραφήματα. Η συνέλιξη γραφήματος προβλέπει τα χαρακτηριστικά του κόμβου στο επόμενο επίπεδο ως

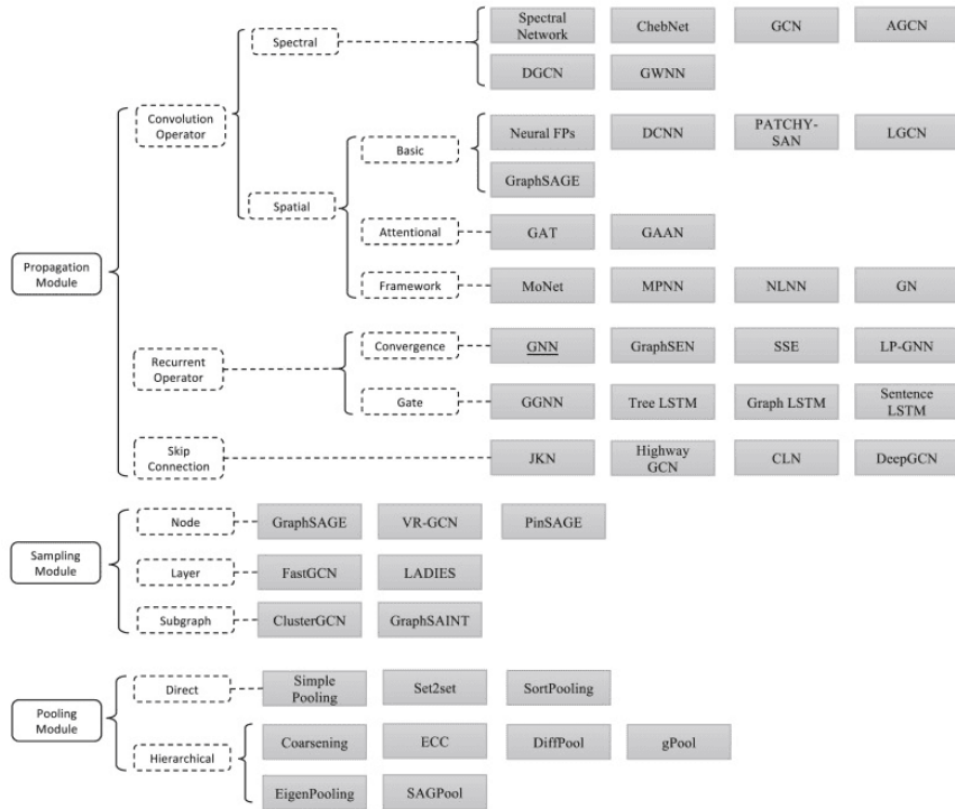


Figure 8: Αρχιτεκτονικές στα Νευρωνικά Δίκτυα Γράφων [4]

συνάρτηση των χαρακτηριστικών των γειτόνων. Μετατρέπει τα χαρακτηριστικά του κόμβου x_i σε έναν λανθάνοντα χώρο h_i που μπορεί να χρησιμοποιηθεί για διάφορους λόγους.

$$x_i \rightarrow h_i$$

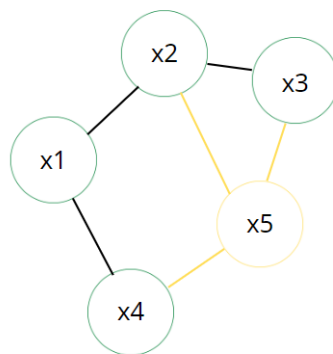


Figure 9: Απεικόνιση 1

$$h_5 = g(x_2, x_3, x_4)$$

Αυτοί οι παράγοντες h_i μας βοηθούν σε εφαρμογές που εμπίπτουν στις παρακάτω κατηγορίες:

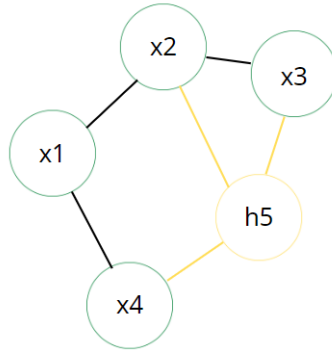


Figure 10: Απεικόνιση 2

- **Ταξινόμηση Κόμβου (Node Classification):** Εάν εφαρμόσουμε μια κοινή συνάρτηση f σε καθένα από τα λανθάνοντα διανύσματα, μπορούμε να κάνουμε προβλέψεις για κάθε κόμβο. Με αυτόν τον τρόπο μπορούμε να ταξινομήσουμε τους κόμβους με βάση τα χαρακτηριστικά τους: $Z_i = f(h_i)$
- **Ταξινόμηση Ακμής (Edge Classification):** Μπορούμε να τα χρησιμοποιήσουμε για να ταξινομήσουμε τις ακμές με βάση τα χαρακτηριστικά τους. Για να το πετύχουμε αυτό, χρειαζόμαστε γενικά τόσο τα διανύσματα γειτονικών κόμβων όσο και τα χαρακτηριστικά ακμών, εάν υπάρχουν. Μαθηματικά έχουμε: $Z_{ij} = f(h_i, h_j, e_{ij})$
- **Ταξινόμηση Γράφου (Graph Classification):** Μπορούμε να προβλέψουμε κάποιο χαρακτηριστικό για ολόκληρο το γράφημα συγκεντρώνοντας όλα τα χαρακτηριστικά του κόμβου και εφαρμόζοντας μια κατάλληλη συνάρτηση f . Η συνάρτηση είναι συνήθως μια αμετάβλητη συνάρτηση μετάθεσης, όπως ένα άθροισμα, μια μέση πράξη ή μια πράξη συγκέντρωσης. Εδώ έχουμε: $z_G = f(\sum h_i)$

Μια εννοιολογία στην βιβλιογραφία των νευρωνικών δικτύων γράφων που μπορεί να προκαλέσει σύγχυση είναι αυτή που αφορά το τρόπο μάθησης των δικτύων. Όπως αναφέραμε και στην περίπτωση της μεταφοράς γνώσης υπάρχουν δύο κατηγορίες:

- **Transductive Learning:** Σε αυτή την περίπτωση το μοντέλο έχει ήδη συναντήσει τόσο την εκπαίδευση όσο και στην διαδικασία της δοκιμής τα δεδομένα εισόδου. Στην περίπτωσή μας θέλουμε να προβλέψουμε τις ετικέτες των κόμβων ενός μεγάλου συνήθως γράφου. Εάν προστεθεί ένας νέος κόμβος στο γράφημα, πρέπει να επανεκπαιδύσουμε το μοντέλο.
- **Inductive Learning:** Σε αυτή την περίπτωση το μοντέλο βλέπει μόνο τα δεδομένα εκπαίδευσης. Έτσι, το παραγόμενο μοντέλο θα χρησιμοποιηθεί για την πρόβλεψη ετικετών γραφημάτων πάνω σε δεδομένα που δεν έχει δει και δεν γνωρίζει.

Στη συνέχεια θα παρουσιάσουμε κάποιες γνωστές αρχιτεκτονικές ΝΔΓ [7].

Φασματικές Μέθοδοι

Αυτές οι μέθοδοι βασίζονται στην επεξεργασία σήματος γραφήματος και ορίζουν τον τελεστή συνέλιξης στον φασματικό τομέα χρησιμοποιώντας τον μετασχηματισμό Fourier F . Το σήμα

γραφήματος x μετασχηματίζεται αρχικά στο φασματικό πεδίο από το μετασχηματισμό γραφήματος Fourier F . Στη συνέχεια, η πράξη συνέλιξης διεξάγεται με πολλαπλασιασμό βάσει στοιχείων. Μετά τη συνέλιξη, το προκύπτον σήμα μετατρέπεται ξανά χρησιμοποιώντας το αντίστροφο γράφημα μετασχηματισμού Fourier F^{-1} .

$$F(x) = U^T x$$

$$F^{-1}(x) = Ux$$

όπου U είναι ο πίνακας που ορίζεται από τις ιδιοτιμές του πίνακα L , με $L = U\Lambda U^T$. Ο Λ είναι ένας διαγώνιος πίνακας με στοιχεία τις ιδιοτιμές του $\Lambda = \text{diag}([\lambda_0, \dots, \lambda_1])$.

Ο τελεστής συνέλιξης τότε θα είναι ίσος με το παρακάτω:

$$g * x = F^{-1}(F(g)F(x)) = U(U^T g \cdot U^T x)$$

Το L είναι το κανονικοποιημένο γράφημα Laplace και κατασκευάζεται όπως απεικονίζεται παρακάτω:

$$L = I - D^{-\frac{1}{2}} A D^{\frac{1}{2}}$$

όπου $U^T g$ είναι το φίλτρο στο φασματικό πεδίο, D είναι ο πίνακας βαθμών και A είναι ο πίνακας γειτνίασης του γράφου.

- **Φασματικά Δίκτυα:** Τα φασματικά δίκτυα μειώνουν το φίλτρο στον φασματικό τομέα ώστε να είναι ένας διαγώνιος πίνακας g_w , όπου w είναι οι μαθησιακή παράμετρος του δικτύου. Ως αποτέλεσμα, μπορεί κανείς να δημιουργήσει ένα δίκτυο που μαθαίνει συνελικτικά φίλτρα για ταξινόμηση γραφημάτων. Υπάρχουν ορισμένα μειονεκτήματα σε αυτή την αρχιτεκτονική. Πρώτα το φίλτρο εφαρμόζεται σε ολόκληρο το γράφημα, οπότε δεν υπάρχει έννοια της εντοπιότητας που έχουμε στις εικόνες. Δεύτερον, είναι υπολογιστικά αναποτελεσματικό, ειδικά για μεγάλα γραφήματα.
- **ChebNets:** Για την επίλυση του προβλήματος της τοπικότητας, τα ChebNets προτείνουν ότι η αναπαράσταση χαρακτηριστικών οποιουδήποτε διανύσματος θα πρέπει να επηρεάζεται μόνο από τη γειτονιά k -hop. Χρησιμοποιώντας την επέκταση Chebyshev της τάξης K , μπορούμε να ορίσουμε μια k -τοπική συνέλιξη που μπορεί να χρησιμοποιηθεί για να σχηματίσει ένα συνελικτικό νευρωνικό δίκτυο.
- **Graph Convolutional Networks (GCN):** Το Graph Convolutional Networks είναι η πιο συχνά χρησιμοποιούμενη αρχιτεκτονική σε πραγματικές εφαρμογές. Στα GCN, η K -τοπική συνέλιξη που προτείνεται στα ChebNets απλοποιείται σε $K=1$. Τα GCN προτείνουν τις παρακάτω αλλαγές:
 - Επιβάλλουν αυτοσυνδέσεις προσθέτοντας τον πίνακα ταυτότητας I στον πίνακα γειτνίασης A .

$$\tilde{A} = A + I$$

- Χρησιμοποίησαν τη συμμετρική κανονικοποίηση του Laplacian L .

$$L_{norm} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$$

- Χρησιμοποίησαν ένα τέχνασμα επανακανονικοποίησης για να λύσουν προβλήματα κλίσης που εξαφανίζονται.

$$I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \rightarrow \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$$

όπου χρησιμοποιούμε τον πίνακα βαθμών D_{ij}

Με βάση τα παραπάνω, εάν ο H είναι ο πίνακας χαρακτηριστικών και ο W ο εκπαιδευσιμος πίνακας βάρους, ο κανόνας ενημέρωσης για το επίπεδο GCN γίνεται ο εξής:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

Από την οπτική των κόμβων, μπορούμε να γράψουμε το εξής:

$$h_i^{(l)} = \sigma\left(\sum_{j \in N_j} c_{ij}Wh_j\right)$$

όπου το $c_{ij} = \frac{1}{\sqrt{|N_i||N_j|}}$ και τα N_i, N_j είναι τα μεγέθη των γειτονιών των κόμβων.

Τα GCN υπολογιστικά είναι πολύ πιο αποτελεσματικά από τους προκατόχους τους και πιο απλά στην κωδικοποίηση, αλλά έχουν μερικούς περιορισμούς.

- Δεν υποστηρίζουν άμεσα χαρακτηριστικά ακμών.
- Παραλείπουν την έννοια των μηνυμάτων σε γραφήματα. Συνήθως, οι κόμβοι μπορούν να στέλνουν μηνύματα (αριθμητικά διανύσματα) κατά μήκος των άκρων του γραφήματος.

Χωρικές Μέθοδοι

Οι χωρικές προσεγγίσεις ορίζουν τις συνελίξεις απευθείας στο γράφημα με βάση την τοπολογία του γραφήματος. Συνήθως ακολουθούν το ίδιο μοτίβο:

- Τα διανύσματα χαρακτηριστικών του κόμβου μετασχηματίζονται χρησιμοποιώντας κάποιο είδος προβολής.
- Συγκεντρώνονται με μια αμετάβλητη συνάρτηση μετάθεσης
- Το διάνυσμα χαρακτηριστικών κάθε κόμβου ενημερώνεται με βάση τις τρέχουσες τιμές του και τη συγκεντρωτική αναπαράσταση γειτονιάς.
- **Message Passing Neural Networks (MPNN):** Τα νευρωνικά δίκτυα διέλευσης μηνυμάτων (MPNN) χρησιμοποιούν την έννοια των μηνυμάτων στα GNN. Ένα μήνυμα m_{ij} μπορεί να σταλεί στις άκρες i και j και υπολογίζεται χρησιμοποιώντας μια συνάρτηση

μηνύματος f_e . Η f_e είναι γενικά ένα μικρό MLP και λαμβάνει υπόψη τα χαρακτηριστικά τόσο των κόμβων όσο και των ακμών. Για δύο κόμβους i και j με χαρακτηριστικά ακμών e_{ij} ,

$$m_{ij} = f_e(h_i, h_j, e_{ij})$$

Στη συνέχεια, όλα τα μηνύματα που φτάνουν σε κάθε κόμβο συγκεντρώνονται χρησιμοποιώντας μια αμετάβλητη συνάρτηση μετάθεσης, όπως η άθροιση. Η συγκεντρωτική αναπαράσταση στη συνέχεια συνδυάζεται με τα υπάρχοντα χαρακτηριστικά κόμβου μέσω της f_v , καταλήγοντας σε μία ανανεωμένη διάνυσμα χαρακτηριστικών κόμβου. Μαθηματικά, έχουμε το παρακάτω,

$$h_i = f_v(h_i, \sum_{j \in N_i} m_{ij})$$

Τα MPNN είναι ένα ισχυρό framework και θεωρούνται μία από τις πιο γενικές αρχιτεκτονικές GNN. Ωστόσο, περιστασιακά υποφέρουν από προβλήματα επεκτασιμότητας, επειδή απαιτούν αποθήκευση και επεξεργασία μηνυμάτων ακμών καθώς και των χαρακτηριστικών του κόμβου. Γι' αυτό στην πράξη, έχουν εφαρμογή μόνο για μικρά γραφήματα.

- **Graph Attention Networks (GAT):** Για να κατανοήσουμε τα δίκτυα προσοχής γραφήματος, ας επανεξετάσουμε τον κανόνα ενημέρωσης κόμβων των GCN. Έχουμε αυτόν τον συντελεστή $\frac{1}{\sqrt{|N_i||N_j|}}$, ο οποίος πολλαπλασιάζεται με την προβολή των χαρακτηριστικών των κόμβων. Ο συντελεστής προέρχεται από τον πίνακα βαθμών του γράφου και εξαρτάται σε μεγάλο βαθμό από τη δομή του γραφήματος. Διαισθητικά, αντιπροσωπεύει πόσο σημαντικά είναι τα χαρακτηριστικά του κόμβου j για τον κόμβο i .

$$h_i^{(l)} = \sigma\left(\sum_{j \in N_i} \frac{1}{\sqrt{|N_i||N_j|}} W h_j\right)$$

Η κύρια ιδέα πίσω από το GAT είναι να υπολογιστεί αυτός ο συντελεστής σιωπηρά και όχι ρητά όπως κάνουν τα GCN. Με αυτόν τον τρόπο μπορούμε να χρησιμοποιήσουμε περισσότερες πληροφορίες εκτός από τη δομή του γραφήματος για να προσδιορίσουμε τη «σημασία» κάθε κόμβου.

Οι συγγραφείς πίσω από το GAT πρότειναν ότι ο συντελεστής, που στο εξής θα συμβολίζεται ως a_{ij} , θα πρέπει να υπολογίζεται με βάση τα χαρακτηριστικά του κόμβου, τα οποία στη συνέχεια μεταβιβάζονται σε μια συνάρτηση προσοχής. Λάβετε υπόψη ότι μπορούν επίσης να συμπεριληφθούν χαρακτηριστικά ακμών. Τέλος, η συνάρτηση softmax εφαρμόζεται στα βάρη προσοχής a_{ij} που καταλήγουν σε μια κατανομή πιθανότητας. Μαθηματικά έχουμε:

$$a_{ij} = \text{attention}(h_i, h_j)$$

$$a_{ij} = \frac{\exp(a_{ij})}{\sum_{k \in N_i} \exp(a_{ik})}$$

Ο κανόνας ενημέρωσης μπορεί να σχηματοποιηθεί όπως παρακάτω:

$$h_i^{(l)} = \sigma\left(\sum_{j \in N_i} a_{ij} W h_j\right)$$

Μέθοδοι Δειγματοληψίας

Ένα σημαντικό μειονέκτημα των περισσότερων αρχιτεκτονικών GNN είναι η επεκτασιμότητα. Γενικά, το διάνυσμα χαρακτηριστικών κάθε κόμβου εξαρτάται από ολόκληρη τη γειτονιά του. Αυτό μπορεί να είναι αρκετά αναποτελεσματικό για τεράστια γραφήματα με μεγάλες γειτονιές. Για την επίλυση αυτού του προβλήματος, έχουν ενσωματωθεί μονάδες δειγματοληψίας. Η κύρια ιδέα των δομοστοιχείων δειγματοληψίας είναι ότι αντί να χρησιμοποιούμε όλες τις πληροφορίες γειτονιάς, μπορούμε να δειγματίσουμε ένα υποσύνολο αυτών για να πραγματοποιήσουμε διάδοση.

- **GraphSAGE:** Το GraphSAGE έκανε δημοφιλή αυτή την ιδέα προτείνοντας το ακόλουθο πλαίσιο:

- Διαλέξτε ομοιόμορφα ένα δείγμα κόμβων από τη γειτονιά.
- Συγκεντρώστε τις πληροφορίες χαρακτηριστικών από γείτονες του δείγματος.
- Με βάση τη συνάνθριση, εκτελούμε ταξινόμηση γραφήματος ή ταξινόμηση κόμβων.

Σε κάθε επίπεδο, επεκτείνουμε το βάθος K-γειτονιάς, με αποτέλεσμα τη δειγματοληψία των χαρακτηριστικών του κόμβου K-hops μακριά. Καταλαβαίνουμε εύκολα πόσο υπολογιστικά αποδοτικότερο είναι αυτό σε σύγκριση με τη χρήση ολόκληρης της γειτονιάς. Αυτό ολοκληρώνει την εμπρός διάδοση του GraphSage.

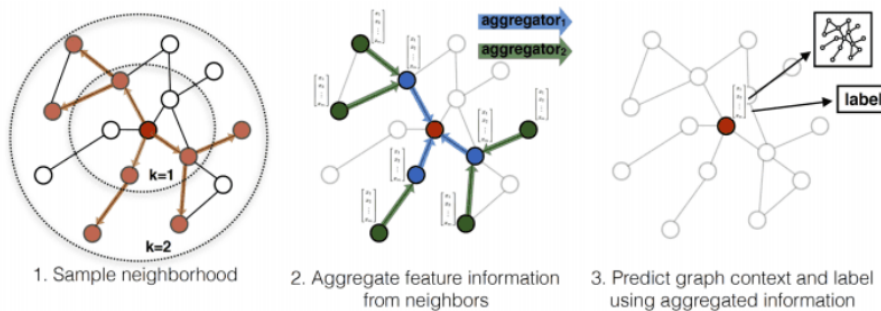


Figure 11: Αρχιτεκτονική GraphSAGE [7]

- **PinSAGE** Το PinSAGE είναι μια άμεση συνέχεια του GraphSAGE και μιας από τις πιο δημοφιλείς εφαρμογές GNN. Το PinSAGE είναι βασικά το GraphSAGE που εφαρμόζεται σε ένα πολύ μεγάλο γράφημα (3 δισεκατομμύρια κόμβοι και 18 δισεκατομμύρια άκρες). Προτείνεται από το Pinterest και χρησιμοποιείται στο σύστημα συστάσεων του.

Εκτός από την τεράστια μηχανική τους προσπάθεια,ας δούμε εν συντομία τις κύριες αρχές της αρχιτεκτονικής:

- Καθορίζουν τη γειτονιά του κόμβου χρησιμοποιώντας τυχαίους περιπάτους. Με την προσομοίωση τυχαίων περιπάτων ξεκινώντας από κόμβους-στόχους, μπορούν να επιλέξουν τους κορυφαίους κόμβους με τον υψηλότερο αριθμό επισκέψεων. Μια παρενέργεια είναι ότι τώρα σε κάθε κόμβο εκχωρείται μια βαθμολογία σπουδαιότητας που υποδεικνύει πόσο σημαντικός είναι για τον κόμβο-στόχο.
- Η συνάνθρωση πραγματοποιείται με τη χρήση «δειγματοληψίας σημαντικότητας». Στη δειγματοληψία σπουδαιότητας, απλώς κανονικοποιούμε και συνοψίζουμε τις βαθμολογίες σπουδαιότητας που δημιουργούνται από τους τυχαίους περιπάτους.
- Το μοντέλο εκπαιδεύεται με εποπτευόμενο τρόπο σε ένα σύνολο δεδομένων κόμβων που συνδέονται με βάση την ιστορική δέσμευση των χρηστών στο Pin-terest.

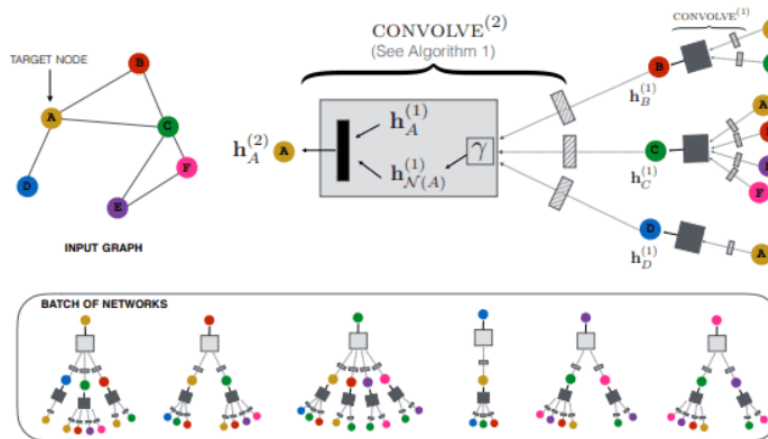


Figure 12: Αρχιτεκτονική PinSAGE [7]

Δυναμικοί Γράφοι

Τα δυναμικά γραφήματα είναι γραφήματα των οποίων η δομή αλλάζει συνεχώς με την πάροδο του χρόνου. Αυτό περιλαμβάνει τόσο κόμβους όσο και ακμές, οι οποίες μπορούν να προστεθούν, να τροποποιηθούν και να διαγραφούν. Παραδείγματα περιλαμβάνουν κοινωνικά δίκτυα, οικονομικές συναλλαγές και άλλα. Ένα δυναμικό γράφημα μπορεί να αναπαρασταθεί ως μια ταξινομημένη λίστα ή μια ροή γεγονότων με χρονική σήμανση που αλλάζουν τη δομή του γραφήματος.

Η έρευνα ML σε δυναμικά γραφήματα είναι πολύ νέα, αλλά υπάρχουν μερικές αξιοσημείωτες αρχιτεκτονικές.

- **Temporal Graph Networks (TGN):** Δεδομένου ότι τα δυναμικά γραφήματα αντιπροσωπεύονται ως μια λίστα με βάση το χρόνο, οι γειτονιές του κόμβου αλλάζουν με την πάροδο του χρόνου. Κάθε φορά t , μπορούμε να πάρουμε ένα στιγμιότυπο του γραφήματος. Η γειτονιά σε μια συγκεκριμένη χρονική στιγμή t ονομάζεται χρονική γειτονιά.

Ο στόχος του TGN είναι να προβλέψει τις ενσωματώσεις κόμβων σε μια συγκεκριμένη χρονική σήμανση. Αυτές οι ενσωματώσεις μπορούν να τροφοδοτηθούν σε ένα δίκτυο αποκωδικοποιητή που θα εκτελέσει τη συγκεκριμένη εργασία.

Η αρχιτεκτονική προτείνεται από το Twitter και εκπαιδεύεται στο γράφημα των tweets τους. Οι κόμβοι αντιπροσωπεύουν τα tweets και οι ακμές τις αλληλεπιδράσεις μεταξύ τους. Ο στόχος του μοντέλου είναι να προβλέψει τις αλληλεπιδράσεις που δεν έχουν συμβεί ακόμη στη χρονική σήμανση t με τη μορφή πιθανότητας. Με άλλα λόγια, έκαναν μια πρόβλεψη άκρων. Το δίκτυο εκπαιδεύεται με τρόπο αυτοεποπτευόμενο: κατά τη διάρκεια κάθε εποχής, ο κωδικοποιητής επεξεργάζεται τα γεγονότα με χρονολογική σειρά και προβλέπει την επόμενη αλληλεπίδραση με βάση τα προηγούμενα.

2.5 Η Χρήση των Βοηθητικών Εργασιών στη Μεταφορά Γνώσης

Βοηθητικές Εργασίες στη Μηχανική Μάθηση Στη μηχανική μάθηση, οι βοηθητικές εργασίες είναι εργασίες στα οποία προσπαθούμε να επιτύχουμε με μοναδικό στόχο την καλύτερη εκτέλεση μιας ή περισσότερων πρωταρχικών εργασιών. Αυτή η κατάσταση, η οποία αναφέρεται εδώ ως βοηθητική μάθηση, έρχεται σε αντίθεση με τη εκμάθηση πολλαπλών εργασιών (multi-task learning), για την οποία μας ενδιαφέρει να ολοκληρώσουμε καλά όλες τις εργασίες και την εκμάθηση μεμονωμένης εργασίας (single-task learning), για την οποία λαμβάνεται υπόψη μόνο μία εργασία.

	Εργασίες που πραγματοποιούνται κατά την εκπαίδευση	Εργασίες που λαμβάνονται υπόψη κατά την αξιολόγηση της απόδοσης
Εκμάθηση Μεμονωμένης Εργασίας	Μία Εργασία	Μία Εργασία
Εκμάθηση πολλαπλών εργασιών	Διάφορες Εργασίες	Διάφορες Εργασίες
Βοηθητική Εκμάθηση	Μία ή περισσότερες κύριες εργασίες, μία ή περισσότερες βοηθητικές εργασίες	Κύριες Εργασίες

Figure 13:

Βοηθητική εκμάθηση σε σύγκριση με την εκμάθηση μεμονωμένης εργασίας και την εκμάθηση πολλαπλών εργασιών

Οι βοηθητικές εργασίες έχουν χρησιμοποιηθεί με επιτυχία για την εκπαίδευση των νευρωνικών δικτύων. Ειδικότερα, βοηθούν ιδιαίτερα σε περιπτώσεις εποπτευόμενης μάθησης με

πολύ λίγα παραδείγματα τα οποία έχουν ετικέτα ή ενισχυτική εκμάθηση με αραιές ανταμοιβές. Ωστόσο, μπορούν να βλάψουν την απόδοση στην κύρια εργασία.

Η βοηθητική εκμάθηση έχει συχνά εφαρμοστεί με επιτυχία σε άλλες ρυθμίσεις, όπως στα παρακάτω παραδείγματα:

- Στο [24] χρησιμοποίησαν σαν τομέα τη Βαθιά Μάθηση, σαν κύρια εργασία την εκπαίδευση ενός βαθιά νευρωνικού δικτύου και σαν βοηθητικές εργασίες την ανακατασκευή κατεστραμμένων εκδόσεων των εισόδων του νευρωνικού δικτύου και των ενεργοποιήσεων των κρυφών επιπέδων του.
- Στο [24] χρησιμοποίησαν σαν τομέα την Υπολογιστική Όραση, σαν κύρια εργασία τον προσδιορισμό σημείων προσώπου σε εικόνες προσώπου και σαν βοηθητικές εργασίες την εκτίμηση της στάσης του κεφαλιού και πρόβλεψη χαρακτηριστικών του προσώπου («φορώντας γυαλιά», «χαμογελώντας» και «φύλο»).
- Στο [25] χρησιμοποίησαν σαν τομέα την Υπολογιστική Όραση, σαν κύρια εργασία την ανίχνευση αντικειμένων σε εσωτερικές σκηνές και σαν βοηθητικές εργασίες την πρόβλεψη ετικετών σκηνών και αξιολόγηση προσανατολισμού βάλτους και επιφάνειας σε επίπεδο pixel.
- Στο [29] χρησιμοποίησαν σαν τομέα την Ενισχυτική Εκμάθηση, σαν κύρια εργασία το να παίζουμε ένα βιντεοπαιχνίδι σε εσωτερικές σκηνές και σαν βοηθητικές εργασίες την πρόβλεψη της μελλοντικής κατάστασης με βάση την τρέχουσα κατάσταση και την τρέχουσα δράση

Τυπική προσέγγιση για τη βαθιά βοηθητική μάθηση: Ας υποθέσουμε ότι είμαστε στην περίπτωση που έχουμε μία κύρια εργασία και μία βοηθητική. Στην τυπική προσέγγιση εκμάθησης πολλαπλών εργασιών με νευρωνικό δίκτυα, ένα ενιαίο δίκτυο συνδυάζει πολλά υποδίκτυα, καθένα από αυτά αντιστοιχεί σε μια μεμονωμένη εργασία. Αυτά τα υποδίκτυα είτε μοιράζονται κάποιους νευρώνες (hard parameter sharing), είτε έχουν κάποιους από τους νευρώνες τους περιορισμένους να είναι παρόμοιοι. Ένα τέτοιο δίκτυο συνήθως εκπαιδεύεται ορίζοντας τη συνάρτηση απώλειας \mathcal{L}_i για κάθε εργασία i και στη συνέχεια με την ελαχιστοποίηση του σταθμισμένου αθροίσματος αυτών των απωλειών $\sum \lambda_i \mathcal{L}_i$, με τη βοήθεια του λαγορίθμου gradient descent.

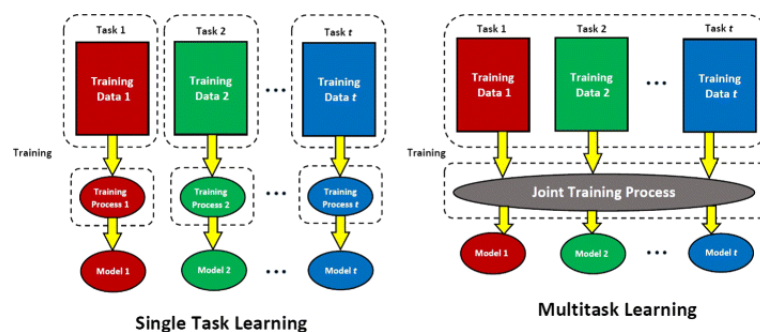


Figure 14:

Εκμάθηση Μεμονωμένης Εργασίας-Εκμάθηση Πολλαπλών Εργασιών [59]

2.5.1 SELAR Framework

Όπως αναφέραμε και παραπάνω οι βοηθητικές εργασίες που έχουν εφαρμογή σε ετερογενείς (heterogeneous) γράφους, που περιέχουν πλούσιες σημασιολογικές πληροφορίες με διάφορους τύπους κόμβων και ακμών, έχουν διερευνηθεί λιγότερο στη βιβλιογραφία. Για την εκμάθηση νευρωνικών δικτύων ετερογενών γράφων, στο [34] προτείνεται μια νέα αυτοεποπτευόμενη μέθοδος εκμάθησης βοηθητικών εργασιών (self-supervised auxiliary learning method) χρησιμοποιώντας μετα-μονοπάτια meta-paths, τα οποία είναι σύνθετες σχέσεις πολλαπλών τύπων ακμών. Η προτεινόμενη μέθοδος είναι η εκμάθηση της κύριας εργασίας προβλέποντας μετα-μονοπάτια ως βοηθητικές εργασίες. Αυτό μπορεί να θεωρηθεί ως ένας τύπος μετα-εκμάθησης. Υπενθυμίζουμε, ότι η μέθοδος της βοηθητικής εκμάθησης είναι μια στρατηγική μάθησης με χρήση βοηθητικών εργασιών για την υποβοήθηση της κύριας εργασίας. Είναι παρόμοια με την εκμάθηση πολλαπλών εργασιών, αλλά η βοηθητική εκμάθηση ενδιαφέρεται μόνο για την εκτέλεση της κύριας εργασίας.

Η μέθοδος της μετα-εκμάθησης στοχεύει στην αποτελεσματική εκμάθηση μοντέλων και γενικεύει τη στρατηγική εκμάθησης σε νέες εργασίες. Σε αντίθεση με τους κλασσικούς αλγόριθμους μάθησης που γενικεύουν σε δείγματα, η μετα-εκμάθηση γενικεύει εργασίες. Εδώ, χρησιμοποιούμε την μετα-εκμάθηση για να μεταφέρουμε τη γνώση από τις βοηθητικές εργασίες στην κύρια εργασία.

Παρακάτω θα ορίσουμε κάποιες βασικές έννοιες που είναι απαραίτητες για την κατανόηση του SELAR Framework. Η παρούσα μελέτη θα εστιάσει σε ετερογενείς γράφους. Να υπενθυμίσουμε ότι ετερογενείς γράφοι είναι αυτοί που έχουν πολλαπλούς τύπους κόμβων και ακμών. Πιο συγκεκριμένα,

Ορισμός Έστω $G = (V, E)$ ένα γράφος με ένα σύνολο κόμβων V και ακμών E . Ένα ετερογενής γράφος είναι ένας γράφος εξοπλισμένος με μια συνάρτηση αντιστοίχισης τύπου κόμβου $f_v : V \rightarrow T^v$ και μια συνάρτηση αντιστοίχισης τύπου ακμής $f_e : E \rightarrow T^e$, όπου T^v είναι ένα σύνολο τύπων κόμβων και T^e είναι ένα σύνολο τύπων ακμών. Κάθε κόμβος $v_i \in V$ (και ακμή $e_{ij} \in E$ αντίστοιχα) έχει έναν τύπο κόμβου, δηλαδή $f_v(v_i) \in T^v$, (και ένας τύπος ακμής $f_e(e_{ij}) \in T^e$ αντίστοιχα). Θα εξετάσουμε τα ετερογενή γραφήματα με $|T^e| > 1$ ή $|T^v| > 1$ Όταν $|T^e| = 1$ και $|T^v| = 1$, έχουμε έναν ομογενή γράφο.

Όπως αναφέραμε και πιο πάνω, ο αλγόριθμος SELAR χρησιμοποιεί την πρόβλεψη μετα-μονοπατιών σαν αυτο-εποπτευόμενη βοηθητική εργασία. Σαν ένα μετα-μονοπάτι ορίζουμε μια διαδρομή όπου μια ακολουθία κόμβων συνδέεται με ακμές. Πιο συγκεκριμένα,

Ορισμός Ένα μετα-μονοπάτι είναι μια διαδρομή σε έναν ετερογενή γράφο G όπου μια ακολουθία κόμβων συνδέεται με ετερογενείς ακμές, δηλ. $v_1 \xrightarrow{t_1} v_2 \xrightarrow{t_2} \dots \xrightarrow{t_l} v_{l+1}$, όπου $t_l \in T^e$ δηλώνει την l -η ακμή του μετα-μονοπατιού. Το μετα-μονοπάτι μπορεί να θεωρηθεί ως μια σύνθετη σχέση $R = t_1 \circ t_2 \dots \circ t_l$ μεταξύ κόμβου v_1 και v_{l+1} , όπου $R_1 \circ R_2$ υποδηλώνει τη σύνθεση της σχέσης R_1 και R_2 .

Ο ορισμός του μετα-μονοπατιού γενικεύει συνδέσεις πολλαπλών βημάτων και φαίνεται ότι είναι χρήσιμο για την ανάλυση ετερογενών γράφων. Για παράδειγμα, στο σύνολο δεδομένων Book-Crossing, το «user-item-written.series-item-user» υποδηλώνει ένα μετα-μονοπάτι που συνδέει χρήστες που τους αρέσει η ίδια σειρά βιβλίων.

Η μέθοδος της πρόβλεψης του μετα-μονοπατιού (meta-path prediction) είναι παρόμοια με την πρόβλεψη συνδέσμου (link prediction), αλλά τα μετα-μονοπατία επιτρέπουν ετερογενείς σύνθετες σχέσεις. Η πρόβλεψη μετα-μονοπατιού μπορεί να επιτευχθεί με τον ίδιο τρόπο όπως η πρόβλεψη συνδέσμου. Αν δύο κόμβοι u και v είναι συνδεδεμένοι με ένα μετα-μονοπάτι p με ετερογενείς ακμές (t_1, t_2, \dots, t_l) , τότε $y_{u,v}^p = 1$, αλλιώς $y_{u,v}^p = 0$. Οι ετικέτες μπορούν να δημιουργηθούν από τον $A_p = A_{t_l} \dots A_{t_2} A_{t_1}$, όπου A_t είναι ο πίνακας γειτνίασης (adjacency matrix) της ακμής τύπου t . Η binary τιμή στο (u, v) στο A_p υποδεικνύει εάν το u και το v συνδέονται μέσω του p . Θα χρησιμοποιήσουμε την πρόβλεψη του μετα-μονοπατιού σαν μία αυτο-εποπτευόμενη βοηθητική εργασία.

Αρχιτεκτονική Για τη βελτιστοποίηση του μοντέλου, όπως η πρόβλεψη συνδέσμου, είναι η διασταυρούμενη εντροπία μεταχειρισμένος. Το νευρωνικό δίκτυο γράφου f διαμοιράζεται μεταξύ της πρόβλεψης συνδέσμου και της πρόβλεψης μετα-μονοπατιού. Όπως κάθε μέθοδος βοηθητικής εκμάθησης, τα μετα-μονοπατία που τα χρησιμοποιούμε εδώ σαν βοηθητικές εργασίες θα πρέπει να επιλέγονται προσεκτικά και να σταθμίζονται σωστά, έτσι ώστε η πρόβλεψη του μετα-μονοπατιού να μην ανταγωνίζεται τη πρόβλεψη συνδέσμου, ειδικά όταν η χωρητικότητα του δικτύου είναι περιορισμένη. Για την αντιμετώπιση αυτών των ζητημάτων, προτείνεται ένα framework που αυτόματα επιλέγει μετα-μονοπατία και τα εξισορροπεί με τη πρόβλεψη συνδέσμου μέσω μετα-εκμάθησης.

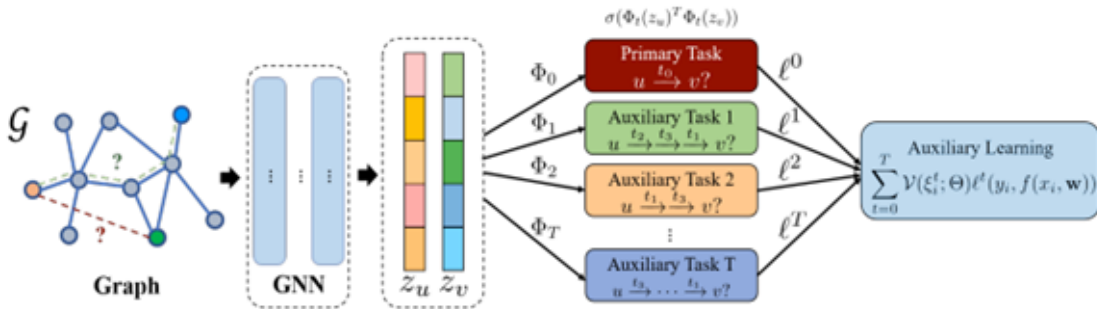


Figure 15:
SELAR Framework [34]

Το SELAR ξεκινά να μαθαίνει την πρωταρχική εργασία και έχει πολλαπλές βοηθητικές εργασίες που θα το βοηθήσουν σε αυτό το σκοπό. Αυτό το γράφουμε ως εξής:

$$\min_{\mathbf{w}, \Theta} E_{(x,y) \sim D^{pr}} [\mathcal{L}^{pr}(\mathbf{w}^*(\Theta))] \quad , \quad \text{όπου } \mathbf{w}^*(\Theta) = \operatorname{argmin}_{\mathbf{w}} E_{(x,y) \sim D^{pr}} [\mathcal{L}^{pr+au}(\mathbf{w}; \Theta)]$$

Το $\mathcal{L}^{pr}(\cdot)$ είναι η συνάρτηση απώλειας της πρωταρχικής εργασίας για την αξιολόγηση του εκπαιδευμένου μοντέλου $f(x; \mathbf{w}^*(\Theta))$ στα μετα-δεδομένα D^{pr} και \mathcal{L}^{pr+au} είναι η συνάρτηση

απώλειας για την εκπαίδευση ενός μοντέλου σε δεδομένα εκπαίδευσης D^{pr+au} με τις πρωταρχικές και βοηθητικές εργασίες.

Κάθε εργασία \mathcal{T}_t έχει N_t δείγματα και \mathcal{T}_0 και $\{\mathcal{T}_t\}_{t=1}^T$ είναι οι πρωταρχική και βοηθητικές εργασίες αντίστοιχα. Η προτεινόμενη διατύπωση μαθαίνει πώς να βοηθά την πρωταρχική εργασία βελτιστοποιώντας το Θ μέσω μετα-εκμάθησης. Το πρόβλημα ένθετης βελτιστοποίησης δοθέντος Θ είναι μια διαδικασία εκπαίδευσης με σωστά προσαρμοσμένες συναρτήσεις απώλειας για την εξισορρόπηση των πρωταρχικών και βοηθητικών εργασιών. Η διατύπωση μπορεί να γραφτεί πιο συγκεκριμένα ως:

$$\min_{w, \Theta} \sum_{i=1}^{M_0} \frac{1}{M_0} l^0(y_i^{(0,meta)}, f(x_i^{(0,meta)}; w^*(\Theta)))$$

όπου, l^t και f^t συμβολίζουμε τη συνάρτηση απώλειας και το μοντέλο για την εργασία t . Αντικαθιστούμε το l^t με την τιμή του, όπου $l^t = l^t(y_i^{(t,train)}, f^t(x_i^{(t,train)}; w)) \cdot \xi_i^{(t,train)}$, με το l^t να είναι το διάνυσμα ενσωμάτωσης του i -ου δείγματος για την εργασία t , και $\xi_i^{(t,train)} = [l^t; e_t; y_i^{(t,train)}]$.

Για να εξάγουμε τον αλγόριθμό μας, συντομεύουμε τις εξισώσεις ως $\mathcal{L}^{pr}(w^*(\Theta))$ και $\mathcal{L}^{pr+au}(w; \Theta)$. Με όλα αυτά καταλήγουμε στην παρακάτω διατύπωση:

$$\min_{w, \Theta} \mathcal{L}^{pr}(w^*(\Theta)) \text{ όπου, } w^*(\Theta) = \underset{w}{\operatorname{argmin}} \mathcal{L}^{pr+au}(w; \Theta)$$

Για να παρακάμψουμε τη δυσκολία της βελτιστοποίησης δύο επιπέδων (bi-level optimization), στο στάδιο της μετα-εκμάθησης την προσεγγίζουμε με τις ενημερωμένες παραμέτρους χρησιμοποιώντας την ενημέρωση που λαμβάνουμε μέσω της καθόδου βασισμένη στην κλίση (gradient descent) ως:

$$w^*(\Theta) \approx \hat{w}^k(\Theta^k) = w^k - \alpha \nabla_w \mathcal{L}^{pr+au}(w^k; \Theta^k)$$

όπου α είναι ο ρυθμός μάθησης για το w . Δεν αξιολογούμε αριθμητικά το $\hat{w}^k(\Theta)$ αντ' αυτού συνδέουμε το υπολογιστικό γράφημα του \hat{w}^k στο $\mathcal{L}^{pr}(w^*(\Theta))$ για να βελτιστοποιήσουμε το Θ . Έστω $\nabla_{\Theta} \mathcal{L}^{pr}(w^*(\Theta^k))$ η διαβάθμιση που αξιολογείται στο Θ^k . Στη συνέχεια, η ενημέρωση των παραμέτρων Θ δίνεται ως:

$$\Theta^{k+1} = \Theta^k - \beta \nabla_{\Theta} \mathcal{L}^{pr}(\hat{w}^k(\Theta^k))$$

όπου β είναι ο ρυθμός μάθησης για το Θ . Αυτή η ενημέρωση επιτρέπει την απαλή επιλογή χρησίμων βοηθητικών εργασιών και την εξισορρόπησή τους με την πρωταρχική εργασία για τη βελτίωση της απόδοσης της. Χωρίς εργασίες εξισορρόπησης με τη συνάρτηση στάθμισης $V(; \theta)$ οι βοηθητικές εργασίες μπορεί να κυριαρχήσουν στη διαδικασία της εκπαίδευσης και

να υποβαθμίσουν την απόδοση της πρωταρχικής εργασίας.

Οι παράμετροι του μοντέλου w^k μπορούν να ενημερωθούν με τα βελτιστοποιημένα Θ^{k+1} σύμφωνα με την παρακάτω σχέση,

$$w^{k+1} = w^k - \alpha \nabla_w \mathcal{L}^{pr+au}(w^k; \Theta^{k+1})$$

2.6 Μεταφορά Γνώσης σε Νευρωνικά Δίκτυα Γραφών

Τα νευρωνικά δίκτυα γραφημάτων χρησιμοποιούνται ευρέως για την εκμάθηση αποτελεσματικής αναπαράστασης δεδομένων υπό τη δομή γράφου. Εργασίες έχουν καταδείξει ότι η μεταφορά γνώσης από αυτοεποπτευόμενες εργασίες σε εργασίες μεταγενέστερης ροής θα μπορούσε να βελτιώσει περαιτέρω την αναπαράσταση γραφήματος. Σε διάφορες εργασίες έχει μελετηθεί η χρήση συστημάτων συστάσεων σε νευρωνικά δίκτυα γραφών ώστε να πετύχουμε τη σύσταση βάσει αλληλεπιδράσεων χρηστών (social recommendation). Ακόμη, σε πρόσφατες εργασίες μελετούν την προ-εκπαίδευση των νευρωνικών δικτύων γραφών και χρησιμοποιούν βοηθητικές εργασίες (auxiliary tasks) για την επίλυση προβλημάτων. Ο στόχος τους είναι να μεταφέρουν τη γνώση που αποκτήθηκε από τις αυτο-εποπτευόμενες εργασίες (self-supervised tasks) σε μεταγενέστερες εργασίες. Αυτές οι μέθοδοι αποδεικνύουν ότι η χρήση άφθονων δεδομένων χωρίς ετικέτα θα μπορούσε να βελτιώσει περαιτέρω την αναπαράσταση γραφήματος. Στην [16] προτείνεται ένα προσαρμοστικό βοηθητικό μοντέλο στάθμισης απώλειας για την εκμάθηση τα βαρών των βοηθητικών εργασιών ποσοτικοποιώντας τη συνέπεια μεταξύ των βοηθητικών εργασιών και της εργασίας στόχου, ενώ η εκμάθηση του μοντέλου πραγματοποιείται μέσω meta-learning.

2.6.1 Η Μέθοδος της Προεκπαίδευσης στα Νευρωνικά Δίκτυα Γράφων

Πριν προχωρήσουμε στην μέθοδο της προεκπαίδευσης, όπως αναφέραμε και παραπάνω, τα ΝΔΓ χρησιμοποιούν τη συνδεσιμότητα γραφήματος καθώς και τα χαρακτηριστικά του κόμβου και της ακμής για την εκμάθηση ενός διάνυσματος αναπαράστασης (δηλαδή, ενσωμάτωση) h_v για κάθε κόμβο $v \in G$ και ένα διάνυσμα h_G για ολόκληρο το γράφημα G . Μετά από k -επανλήψεις συνάθροισης, η αναπαράσταση του v καταγράφει τις δομικές πληροφορίες στη k -hop γειτονιά του. Τυπικά, το k -ο στρώμα ενός ΝΔΓ είναι:

$$h_v^{(k)} = \text{COMBINE}^{(k)}(h_v^{(k-1)}, \text{AGGREGATE}(\{h_u^{k-1}, h_u^{k-1}, e_{uv}\} : u \in N(v)))$$

όπου $h_v^{(k)}$ είναι η αναπαράσταση του κόμβου v στην k -η επανάληψη/στρώμα, e_{uv} είναι το διάνυσμα χαρακτηριστικών της ακμής μεταξύ u και v , και το $N(v)$ είναι ένα σύνολο γειτόνων του v . Αρχικοποιούμε την $h_v(0) = Xv$.

Για να λάβουμε ολόκληρη την αναπαράσταση του γραφήματος h_G , η συνάρτηση *READOUT* συναθροίζει χαρακτηριστικά των κόμβων από την τελική επανάληψη K ,

$$h_G = \text{READOUT}(\{h_v^{(K)} | v \in G\}).$$

όπου η *READOUT* [17][18] είναι μια αναλλοίωτη συνάρτηση μετάθεσης, όπως ο μέσος όρος ή μία πιο περίπλοκη συνάρτηση συγκέντρωσης σε επίπεδο γραφήματος.

Όπως είδαμε και παραπάνω η μεταφορά γνώσης αναφέρεται σε ένα περιβάλλον όπου ένα μοντέλο, αρχικά εκπαιδευμένο σε ορισμένες εργασίες, επαναπροσδιορίζεται σε διαφορετικές αλλά σχετικές εργασίες. Η διαδικασία της προ-εκπαίδευσης έχει τη δυνατότητα να προσφέρει λύση σε δύο προκλήσεις που αφορούν την εκμάθηση σε δεδομένα υπό τη δομή γράφων.

- Πρώτον, τα δεδομένα με ετικέτα για μία συγκεκριμένη εργασία μπορεί να είναι εξαιρετικά σπάνια. Αυτό το πρόβλημα επιδεινώνεται σε σημαντικά σύνολα δεδομένων γραφημάτων από επιστημονικούς τομείς, όπως η χημεία και η βιολογία, όπου η επισήμανση δεδομένων (π.χ. βιολογικά πειράματα σε υγρό εργαστήριο) απαιτεί πόρους και χρόνο.
- Δεύτερον, τα δεδομένα υπό τη μορφή γράφου από εφαρμογές του πραγματικού κόσμου συχνά περιέχουν δείγματα εκτός κατανομής, το οποίο σημαίνει ότι οι γράφοι στο σύνολο εκπαίδευσης θα είναι διαφορετικοί από αυτούς στο σύνολο δοκιμής. Είναι σημαντικό να σημειώσουμε ότι η πρόβλεψη εκτός κατανομής είναι σύνηθες σε σύνολα δεδομένων υπό τη δομή γράφου του πραγματικού κόσμου, για παράδειγμα, όταν κάποιος θέλει να προβλέψει τις χημικές ιδιότητες ενός ολοκαίνουργιου μορίου, το οποίο είναι διαφορετικό από όλα τα μόρια που έχουν συντεθεί μέχρι τώρα, και ως εκ τούτου διαφέρει από όλα τα μόρια στο σύνολο εκπαίδευσης.

Μια επιτυχημένη διαδικασία μεταφοράς γνώσης δεν είναι μόνο θέμα αύξησης του αριθμού των συνόλων δεδομένων προεκπαίδευσης με ετικέτα που προέρχονται από τον ίδιο τομέα με τη μεταγενέστερη εργασία. Αντίθετα, απαιτείται σημαντική τεχνογνωσία για τον τομέα μας ώστε να επιλέξουμε προσεκτικά παραδείγματα και ετικέτες που συσχετίζονται με τη μεταγενέστερη εργασία που μας ενδιαφέρει. Διαφορετικά, η διαδικασία της μεταφοράς γνώσης από σχετικές εργασίες προ-εκπαίδευσης σε ένα μεταγενέστερο έργο μπορεί να βλάψει τη γενίκευση, η οποία είναι γνωστή ως αρνητική μεταφορά (*negative transfer*).

Η προεκπαίδευση είναι ένας κοινός και αποτελεσματικός τρόπος μεταφοράς γνώσης από σχετικές εργασίες σε μια εργασία στόχο για τη βελτίωση της γενίκευσης. Μία από τις στρατηγικές που προτείνονται στο [17] την διαδικασία της προεκπαίδευσης είναι η ιδέα να προεκπαιδεύσουμε ένα νευρωνικό δίκτυο γράφου τόσο σε επίπεδο μεμονωμένων κόμβων όσο και σε ολόκληρο το γράφο. Αυτή η ιδέα δίνει τη δυνατότητα στο νευρωνικό δίκτυο γράφου να συλλάβει σημαντικά χαρακτηριστικά για τον τομέα μας και στα δύο επίπεδα.

Προεκπαίδευση σε επίπεδο κόμβου: Για τη διαδικασία της προεκπαίδευσης ενός νευρωνικού δικτύου γράφου σε επίπεδο κόμβου, η προσέγγισή που προτείνεται είναι να χρησιμοποιηθούν εύκολα προσβάσιμα δεδομένα χωρίς ετικέτα για να συλλάβουμε τις γνώσεις του συγκεκριμένου τομέα στο γράφο. Εδώ προτείνονται δύο μέθοδοι, η μέθοδος Πρόβλεψης Περιβάλλοντος (*Context Prediction*) και η Κάλυψη Χαρακτηριστικών (*Attribute Masking*).

- Πρόβλεψη Περιβάλλοντος:** Σύμφωνα με τη μέθοδο του Context Prediction, χρησιμοποιούμε υπογράφηματα (sub-graphs) για να προβλέψουμε τις δομές γραφημάτων που υπάρχουν γύρω τους. Ο στόχος είναι να εκπαιδεύσουμε εκ των προτέρων το νευρωνικό δίκτυο γράφου έτσι ώστε να χαρτογραφεί κόμβους που εμφανίζονται σε παρόμοια δομικά πλαίσια με κοντινές ενσωματώσεις (embeddings).

Για κάθε κόμβο n , ορίζουμε την γειτονιά (neighborhood) και το περιβάλλον (context) του γράφου ως εξής: Η γειτονιά K -hop (K -hop neighborhood) του κόμβου n περιέχει όλους τους κόμβους και τις ακμές που βρίσκονται το πολύ K -hop μακριά από τον κόμβο n στο γράφημα. Αυτό οφείλεται στο γεγονός ότι ένα νευρωνικό δίκτυο γράφου με K -επίπεδα συλλέγει πληροφορίες κατά μήκος της K -ης τάξης γειτονιάς του n , και επομένως η ενσω $h_n(K)$ του n -κόμβου εξαρτάται από τους κόμβους που απέχουν το πολύ K -hops απόσταση από τον κόμβο n . Ο γράφος περιβάλλοντος περιγράφεται από δύο παραμέτρους $r1$ και $r2$, και αντιπροσωπεύει ένα υπογράφημα που βρίσκεται στην περιοχή που είναι μεταξύ των γειτονιών $r1$ -hops και $r2$ -hops του κόμβου n .

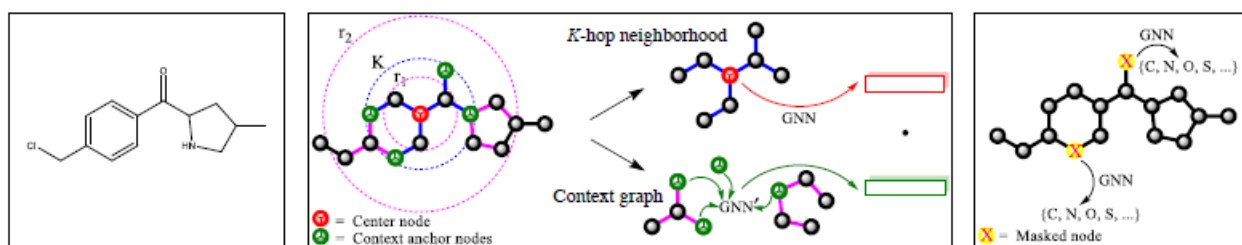


Figure 16: Προεκπαίδευση σε Επίπεδο Γράφου [17]

Για να προβλέψουμε το γράφο περιβάλλοντος κωδικοποιούμε τον γράφο περιβάλλοντος ως διανύσματα σταθερού μήκους. Για το σκοπό αυτό, χρησιμοποιούμε ένα βοηθητικό νευρωνικό δίκτυο γράφου, το οποίο αναφέρουμε ως νευρωνικό δίκτυο γράφου περιβάλλοντος. Εφαρμόζουμε πρώτα αυτό το δίκτυο, για να λάβουμε τις ενσωματώσεις κόμβων στο γράφο περιβάλλοντος. Στη συνέχεια χρησιμοποιούμε negative sampling για να μάθουμε από κοινού το κύριο νευρωνικό δίκτυο γράφου και το νευρωνικό δίκτυο γράφου περιβάλλοντος. Το κύριο ΝΔΓ κωδικοποιεί γειτονιές για να αποκτήσει ενσωματώσεις κόμβων. Αντίστοιχα, το ΝΔΓ περιβάλλοντος κωδικοποιεί γράφους περιβάλλοντος για να αποκτήσει τις ενσωματώσεις περιβάλλοντος. Συγκεκριμένα, ο μαθησιακός στόχος της Πρόβλεψης Περιβάλλοντος είναι μια δυαδική ταξινόμηση για το εάν μια συγκεκριμένη γειτονιά και ένα συγκεκριμένος γράφος περιβάλλοντος ανήκουν στον ίδιο κόμβο. Μετά την προεκπαίδευση, το κύριο ΝΔΓ διατηρείται ως το προεκπαιδευμένο μοντέλο μας.

- Κάλυψη Χαρακτηριστικών:** Κατά τη διάρκεια της διαδικασίας της προεκπαίδευσης με βάση τη μέθοδο της Κάλυψης Χαρακτηριστικών αποκρύπτουμε τα χαρακτηριστικά κόμβου/ακμής και μετά αφήνουμε τα ΝΔΓ να προβλέψουν αυτά τα χαρακτηριστικά με βάση τη γειτονική δομή. Πιο συγκεκριμένα, καλύπτουμε τυχαία τα χαρακτηριστικά εισόδου κόμβου/ακμής, για παράδειγμα τύπους ατόμων σε μοριακά γραφήματα, αντικαθιστώντας τα με ειδικούς δείκτες. Στη συνέχεια με τη βοήθεια ΝΔΓ λαμβάνουμε τις αντίστοιχες ενσωματώσεις κόμβου/ακμών (οι ενσωματώσεις των ακμών μπορούν να

ληφθούν ως άθροισμα των ενσωματώσεων των ακραίων κόμβων της ακμής). Τέλος, ένα γραμμικό μοντέλο εφαρμόζεται πάνω από τις ενσωματώσεις για την πρόβλεψη του κρυφού χαρακτηριστικού κόμβου/ακμής.

Προεκπαίδευση σε επίπεδο γράφου: Στόχος μας είναι να εκπαιδύσουμε εκ των προτέρων τα ΝΔΓ για τη δημιουργία χρήσιμων ενσωματώσεων που αποτελούνται από τις ενσωματώσεις του ουσιαστικού κόμβου. Θέλουμε είναι να διασφαλίσουμε ότι τόσο οι ενσωματώσεις κόμβων όσο και των γραφημάτων είναι υψηλής ποιότητας, έτσι ώστε οι ενσωματώσεις γράφων να είναι ισχυρές και να μπορούν να μεταφερθούν σε όλα τις μεταγενέστερες εργασίες. Υπάρχουν δύο επιλογές για τη διαδικασία της προεκπαίδευσης σε επίπεδο γράφου: να κάνουμε προβλέψεις σχετικά με τα χαρακτηριστικά ολόκληρων γράφων σε επίπεδο τομέα ή να κάνουμε προβλέψεις σχετικά με τη δομή του γράφου.

- **Εποπτευόμενη πρόβλεψη ιδιοτήτων σε επίπεδο γραφήματος:** Καθώς η αναπαράσταση σε επίπεδο γραφήματος h_G χρησιμοποιείται απευθείας για τη διαδικασία fine-tuning στις μεταγενέστερες εργασίες, είναι επιθυμητό να κωδικοποιούμε απευθείας πληροφορίες που αφορούν τον τομέα μας στο h_G . Πιο συγκεκριμένα, εξετάζουμε μια πρακτική μέθοδο ώστε να προεκπαιδύσουμε αναπαραστάσεις γράφου: την προεκπαίδευση πολλαπλών εργασιών σε επίπεδο γράφου για την από κοινού πρόβλεψη ενός διαφορετικού συνόλου εποπτευόμενων ετικετών μεμονωμένων γραφημάτων. Για να προβλέψουμε από κοινού πολλές ιδιότητες γραφήματος, όπου κάθε ιδιότητα αντιστοιχεί σε δυαδικά κατηγοριοποίηση, εφαρμόζουμε γραμμικούς ταξινομητές πάνω στις αναπαραστάσεις του γράφου.
- **Πρόβλεψη δομικής ομοιότητας:** Μια δεύτερη προσέγγιση είναι να υλοποιήσουμε μια εργασία πρόβλεψης σε επίπεδο γράφου όπου ο στόχος θα ήταν να μοντελοποιηθεί η δομική ομοιότητα δύο γράφων. Παραδείγματα τέτοιων εργασιών περιλαμβάνουν τη μοντελοποίηση της απόστασης επεξεργασίας γράφου (graph edit distance) ή την πρόβλεψη ομοιότητας δομής γραφήματος.

Συνολικά, η στρατηγική μας για την προεκπαίδευση είναι πρώτα να εκτελούμε προεκπαίδευση σε επίπεδο κόμβου και στη συνέχεια εποπτευόμενη προεκπαίδευση πολλαπλών εργασιών σε επίπεδο γραφήματος. Όταν ολοκληρωθεί η προεκπαίδευση του ΝΔΓ, προσαρμόζουμε το προεκπαιδευμένο μοντέλο ΝΔΓ σε μεταγενέστερες εργασίες. Πιο συγκεκριμένα, προσθέτουμε γραμμικούς ταξινομητές πάνω από αναπαραστάσεις σε επίπεδο γραφήματος για να προβλέψουμε τις ετικέτες ετικέτες μεταγενέστερων γράφων. Το πλήρες μοντέλο, δηλ. το προεκπαιδευμένο μοντέλο ΝΔΓ και οι μεταγενέστεροι γραμμικοί ταξινομητές, στη συνέχεια προσαρμόζονται (στάδιο fine-tune) με ακρίβεια.

3 Ορισμός Προβλήματος-Μεθοδολογία

3.1 Ορισμός Προβλήματος

Η εργασία μας θα είναι να εκπαιδύσουμε ένα συνελικτικό νευρωνικό δίκτυο (CNN) που μπορεί να αναγνωρίζει αντικείμενα σε εικόνες. Θα εκτελέσουμε δύο διαφορετικές εργασίες

- μια δυαδική κατηγοριοποίηση εικόνας, δηλαδή θα ταξινομήσουμε μία δική φωτογραφία σε δύο κλάσεις
- ένα σύστημα συστάσεων που θα προτείνει στο χρήστη παρόμοια είδη σε σχέση με το αντικείμενο εισόδου

Και τις δύο εργασίες θα τις υλοποιήσουμε με χρήση συνελικτικών νευρωνικών δικτύων, και μάλιστα θα χρησιμοποιήσουμε και προεκπαιδευμένα μοντέλα ώστε να συγκρίνουμε την απόδοση με μεταφορά γνώσης και χωρίς. Τα νευρωνικά δίκτυα, όπως έχουμε ήδη αναφέρει έχουν αποδειχθεί αποτελεσματικά στην εκμάθηση αναπαραστάσεων για διάφορες εργασίες όπως ως ταξινόμηση κόμβου, η πρόβλεψη συνδέσμου και η εύρεση ισχυρών αναπαραστάσεων που μας οδηγούν σε κορυφαίες επιδόσεις σε μια ποικιλία εφαρμογών, συμπεριλαμβανομένης της ανάλυσης κοινωνικών δικτύων, ανάλυση δικτύου παραπομπών, οπτική κατανόηση, συστήματα συστάσεων κ.α.

Η προεκπαίδευση με μια βοηθητική εργασία είναι μια κοινή τεχνική για τα βαθιά νευρωνικά δίκτυα. Στην πραγματικότητα είναι το πρώτο βήμα στην επεξεργασία φυσικής γλώσσας και στην υπολογιστική όραση για την εκμάθηση δικτύων όπως το BERT [28] και το ResNet [29] αξιοποιώντας μεγάλα σύνολα δεδομένων όπως το BooksCorpus [30] και το ImageNet [31]. Σύμφωνα με τα παραπάνω, πρέπει να είμαστε ιδιαίτερα προσεκτικοί με την επιλογή των βοηθητικών εργασιών για τη διαδικασία της προεκπαίδευσης, και τελικά αυτές οι εργασίες να είναι με ουσιαστική γνώση και εξειδίκευση στον τομέα των χαρακτηριστικών του γράφου ώστε να βοηθηθεί πραγματικά η κύρια εργασία.

3.2 Η Χρήση των Προεκπαιδευμένων Μοντέλων στη Μεταφορά Γνώσης

Όπως έχουμε αναφέρει, η μεταφορά γνώσης είναι ένα υποπεδίο της μηχανικής μάθησης και της τεχνητής νοημοσύνης που στοχεύει στην εφαρμογή της γνώσης που αποκτάται από μια εργασία (εργασία προέλευσης) σε μια διαφορετική αλλά παρόμοια εργασία (εργασία στόχος).

Για παράδειγμα, η γνώση που αποκτήθηκε κατά την εκμάθηση της ταξινόμησης κειμένων της Wikipedia μπορεί να χρησιμοποιηθεί για την αντιμετώπιση προβλημάτων ταξινόμησης νομικών κειμένων.

Ένα προεκπαιδευμένο μοντέλο είναι ένα μοντέλο που δημιουργήθηκε και εκπαιδεύτηκε από κάποιον άλλο για να λύσει ένα πρόβλημα που μοιάζει με το δικό μας. Συνήθως επιλέγεται ένα πολύ μεγάλο σύνολο δεδομένων ως βασικό σύνολο δεδομένων, όπως το ImageNet ή το Corpus της Wikipedia. Στη συνέχεια, δημιουργούν ένα μεγάλο νευρωνικό δίκτυο (π.χ., το VGG19 έχει 143.667.240 παραμέτρους) για να λύσουν ένα συγκεκριμένο πρόβλημα (π.χ., αυτό το πρόβλημα είναι η ταξινόμηση εικόνας για το VGG19). Στη συνέχεια,

αυτό το προεκπαιδευμένο μοντέλο θα δημοσιοποιηθεί για να μπορέσουμε να το πάρουμε και να τα επαναχρησιμοποιήσουμε. Αφού αποκτήσουμε αυτά τα προεκπαιδευμένα μοντέλα, επαναχρησιμοποιούμε τη γνώση που αποκτήσαμε, η οποία περιλαμβάνει τα επίπεδα, τα χαρακτηριστικά, τα βάρη και τη μεροληψία.

Συχνά είναι καλύτερο να βελτιστοποιήσετε (fine-tune) το προεκπαιδευμένο μοντέλο για δύο λόγους:

- Για να μπορέσουμε να επιτύχουμε ακόμη μεγαλύτερη ακρίβεια.
- Το μοντέλο μας μετά το στάδιο του fine-tuning μπορεί να καταλήξει σε ένα αποτέλεσμα με πιο σωστή μορφή.

Σε γενικές γραμμές, σε ένα νευρωνικό δίκτυο, ενώ τα στρώματα κατώτερου και μεσαίου επιπέδου αντιπροσωπεύουν συνήθως γενικά χαρακτηριστικά, τα ανώτερα στρώματα αντιπροσωπεύουν τα ειδικά χαρακτηριστικά του προβλήματος. Δεδομένου ότι το νέο μας πρόβλημα είναι διαφορετικό από το αρχικό πρόβλημα, τείνουμε να απορρίπτουμε τα ανώτερα στρώματα.

Προσθέτοντας ειδικά επίπεδα για τα προβλήματά μας, μπορούμε να επιτύχουμε μεγαλύτερη ακρίβεια. Αφού απορρίψουμε τα πάνω στρώματα, πρέπει να τοποθετήσουμε τα δικά μας στρώματα έτσι ώστε να έχουμε την έξοδο που θέλουμε.

Για παράδειγμα, ένα μοντέλο που έχει εκπαιδευτεί με ImageNet μπορεί να ταξινομήσει έως και 1000 αντικείμενα. Εάν προσπαθούμε να ταξινομήσουμε χειρόγραφα ψηφία (π.χ. ταξινόμηση MNIST), ίσως είναι καλύτερο να καταλήξουμε σε ένα τελικό στρώμα με μόνο 10 νευρώνες. Αφού προσθέσουμε τα προσαρμοσμένα επίπεδα στο προεκπαιδευμένο μοντέλο, μπορούμε να το διαμορφώσουμε με ειδικές συναρτήσεις απώλειας και βελτιστοποιητές και να περάσουμε στη διαδικασία του fine-tuning με επιπλέον εκπαίδευση.

Η χρήση των προεκπαιδευμένων μοντέλων είναι ευρεία σε εργασίες υπολογιστικής όρασης όπως είναι: η δημιουργία εικόνας, η ταξινόμηση εικόνων η ανίχνευση ανωμαλιών κ.α.

Τα πιο συνηθισμένα προεκπαιδευμένα μοντέλα που χρησιμοποιούμε σε αυτό το στυλ εργασιών είναι τα παρακάτω:

- VGG19
- Inceptionv3 (GoogLeNet)
- ResNet50

VGG19 Το VGG19 [54] είναι ένα συνελικτικό νευρωνικό δίκτυο (CNN) που έχει βάθος 19 στρωμάτων. Κατασκευάστηκε και εκπαιδεύτηκε από την Karen Simonyan και τον Andrew Zisserman στο Πανεπιστήμιο της Οξφόρδης το 2014 και δημοσιεύτηκε στην εργασία τους, Very Deep Convolutional Networks for Large-Scale Image Recognition το 2015. Το δίκτυο VGG-19 εκπαιδεύεται χρησιμοποιώντας περισσότερες από 1 εκατομμύριο εικόνες από τη βάση δεδομένων ImageNet. Αυτό το προεκπαιδευμένο δίκτυο μπορεί να ταξινομήσει έως και 1000 αντικείμενα όπως πληκτρολόγιο, ποντίκι, μολύβι και πολλά ζώα. Το δίκτυο εκπαιδεύτηκε σε έγχρωμες εικόνες 224x224 pixels.

Πιο συγκεκριμένα για την αρχιτεκτονική του VGG19, όπως αναφέραμε πιο πάνω το VGG19 είναι μια παραλλαγή του μοντέλου VGG που αποτελείται εν συντομία από 19 επίπεδα

(16 στρώματα συνέλιξης, 3 πλήρως συνδεδεμένα στρώματα, 5 επίπεδα MaxPool και 1 στρώμα SoftMax). Το σημαντικό εδώ είναι ότι σε όλα τα στρώματα το kernel size είναι πάντα το ίδιο και ισούται με 3x3.

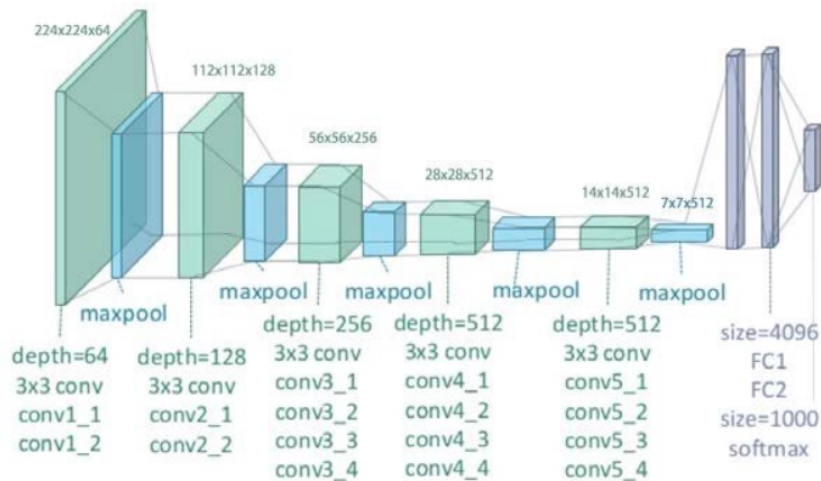


Figure 17:
Αρχιτεκτονική VGG19 [60]

InceptionV3 (GoogLeNet) Το InceptionV3 [56] είναι ένα συνελικτικό νευρωνικό δίκτυο (CNN) που έχει βάθος 50 επιπέδων. Κατασκευάστηκε, εκπαιδεύτηκε από την Google και είναι δημοσιευμένο στην εργασία, με τίτλο "Going more with convolutions". Η προεκπαιδευμένη έκδοση του InceptionV3 με τα βάρη ImageNet μπορεί να ταξινομήσει έως και 1000 αντικείμενα. Το μέγεθος εισόδου εικόνας αυτού του δικτύου ήταν 299x299 pixel, το οποίο είναι μεγαλύτερο από το δίκτυο VGG19. Η αρχιτεκτονική του δικτύου Inception αποτελείται από πολλές κομμάτια της ακόλουθης δομής,

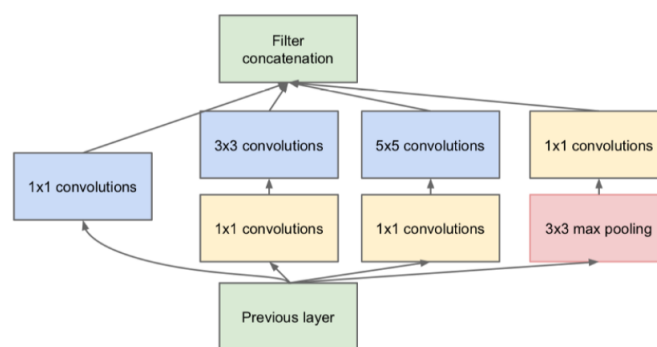


Figure 18:
Inception Module

Όπως φαίνεται και παραπάνω, κάθε μονάδα αποτελείται από τέσσερις παράλληλες λειτουργίες:

- Συνελικτικό Στρώμα 1x1
- Συνελικτικό Στρώμα 3x3

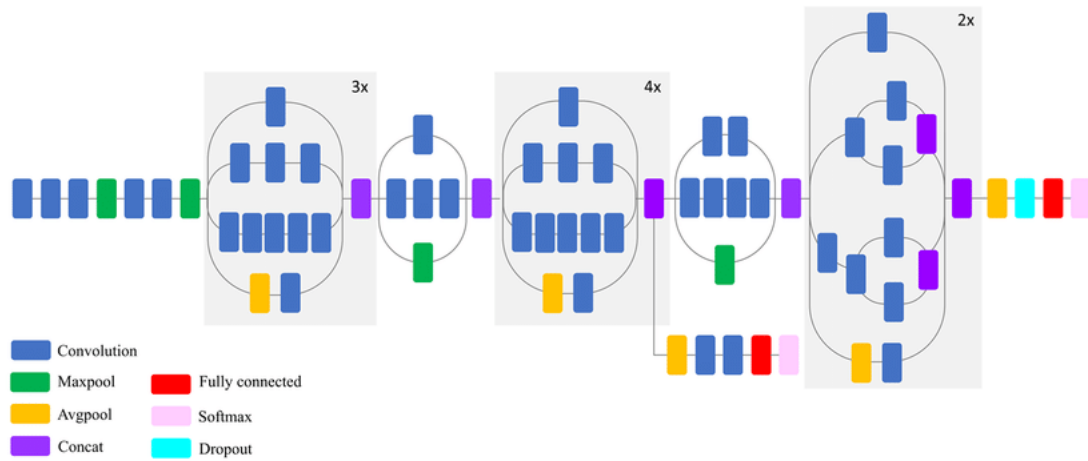


Figure 19:
Αρχιτεκτονική InceptionV3 [61]

- Συνελικτικό Στρώμα 5x5
- Max Pooling Στρώμα

Κάθε μονάδα μπορεί να καταγράψει σημαντικά χαρακτηριστικά σε διαφορετικά επίπεδα. Τα καθολικά χαρακτηριστικά καταγράφονται από το επίπεδο μετατροπής 5x5, ενώ το επίπεδο μετατροπής 3x3 είναι επιρρεπές στην καταγραφή κατανεμημένων χαρακτηριστικών. Η λειτουργία max-pooling είναι υπεύθυνη για τη λήψη χαρακτηριστικών χαμηλού επιπέδου που ξεχωρίζουν σε μια γειτονιά. Σε ένα δεδομένο επίπεδο, όλα αυτά τα χαρακτηριστικά εξάγονται και συνενώνονται πριν τροφοδοτηθεί στο επόμενο στρώμα. Τέλος, αφήνουμε στη διαδικασία της εκπαίδευσης να αποφασίσει ποια χαρακτηριστικά έχουν τις περισσότερες αξίες και βάρη ανάλογα.

ResNet50 (Residual Network) Το ResNet50 [55] είναι ένα συνελικτικό νευρωνικό δίκτυο που έχει βάθος 50 επιπέδων. Κατασκευάστηκε και εκπαιδεύτηκε από τη Microsoft το 2015 με την εργασία Deep Residual Learning for Image Recognition. Αυτό το μοντέλο εκπαιδεύεται σε περισσότερες από 1 εκατομμύριο εικόνες από τη βάση δεδομένων ImageNet. Ακριβώς όπως το VGG-19, μπορεί να ταξινομήσει έως και 1000 αντικείμενα και το δίκτυο εκπαιδεύτηκε σε έγχρωμες εικόνες 224x224 pixel.

Το σημαντικό που πέτυχε η αρχιτεκτονική ResNet είναι ότι χρησιμοποιεί συνδέσεις συντομεύσεων (shortcut connections) για να λύσει το πρόβλημα του vanishing gradient. Ένα από τα κύρια πλεονεκτήματα ενός πολύ βαθύ δικτύου είναι ότι μπορεί να αντιπροσωπεύει πολύ περίπλοκες λειτουργίες. Ωστόσο, ένα τεράστιο εμπόδιο για την εκπαίδευσή τους είναι οι κλίσεις που εξαφανίζονται: τα πολύ βαθιά δίκτυα έχουν συχνά ένα σήμα κλίσης που μηδενίζεται γρήγορα, καθιστώντας έτσι απαγορευτικά αργή την βαθμωτή κάθοδο (gradient descent). Πιο συγκεκριμένα, κατά την βαθμωτή κάθοδο, καθώς γυρίζουμε από το τελικό στρώμα πίσω στο πρώτο στρώμα, πολλαπλασιάζουμε με τον πίνακα των βαρών σε κάθε βήμα. Εάν οι διαβαθμίσεις είναι μικρές, λόγω μεγάλου αριθμού πολλαπλασιασμών, η διαβάθμιση

Μοντέλο	Χρονιά	Κύριο Χαρακτηριστικό
VGG	2014	Σταθερό Μέγεθος Πυρήνα
Inception	2014	Παράλληλοι Πυρήνες
ResNet	2015	Συνδέσεις Συντομεύσεων

Table 1: Σύγκριση Προεκπαιδευμένων Μοντέλων

μπορεί να μειωθεί εκθετικά γρήγορα στο μηδέν. Το βασικό δομικό στοιχείο του ResNet είναι ένα Residual block που επαναλαμβάνεται σε όλο το δίκτυο. Αντί να μάθει τη χαρτογράφηση από το $x \rightarrow F(x)$, το δίκτυο μαθαίνει τη χαρτογράφηση από το $x \rightarrow F(x)+G(x)$. Όταν η διάσταση της εισόδου x και της εξόδου $F(x)$ είναι η ίδια, η συνάρτηση $G(x) = x$ είναι συνάρτηση ταυτότητας και η σύνδεση συντόμευσης ονομάζεται σύνδεση ταυτότητας. Υπάρχουν δύο κύριοι τύποι μπλοκ ResNets: Το μπλοκ ταυτότητας και το συνελικτικό μπλοκ.

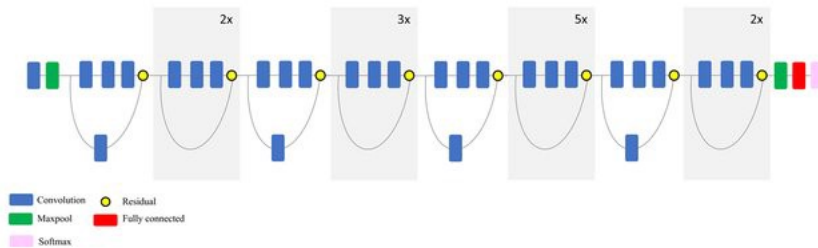


Figure 20:
Αρχιτεκτονική ResNet50 [61]

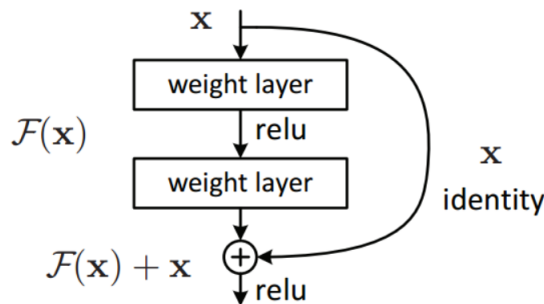


Figure 21:
Residual Block από το αρχικό paper

Αν θέλουμε να κάνουμε μία σύγκριση μεταξύ των τριών προεκπαιδευμένων μοντέλων που θα χρησιμοποιήσουμε μπορούμε να το δούμε ως εξής:

Με βάση αυτά τα προεκπαιδευμένα μοντέλα οδηγούμαστε σε πολύ καλά αποτελέσματα σε εργασίες που αφορούν την αναγνώριση εικόνας και μπορούμε να υλοποιήσουμε και μηχανισ-

μούς συστημάτων συστάσεων που έχουν εφαρμογή στην αναγνώριση εικόνας. Για αυτές τις υλοποιήσεις που θα δούμε και παρακάτω μας είναι απαραίτητη η μεταφορά γνώσης. Για παράδειγμα, η γνώση που αποκτήθηκε κατά την εκμάθηση της αναγνώρισης των αυτοκινήτων θα μπορούσε να εφαρμοστεί κατά την προσπάθεια αναγνώρισης φορτηγών.

4 Ένα Απλό Παράδειγμα Μεταφοράς Γνώσης

Ο σκοπός της παρούσας εργασίας ήταν να ερευνήσει τις εφαρμογές της μεταφοράς γνώσης στο πεδίο των νευρωνικών δικτύων και πιο συγκεκριμένα στα νευρωνικά δίκτυα γράφων. Οι εργασίες [34],[36] ερευνούν τη μεταφορά γνώσης στα δίκτυα υπό τη μορφή γράφου, και μάλιστα στη πρώτη αξιοποιούν τα μετα-μονοπάτια σαν βοηθητικές εργασίες, αλλά δεν έχουμε αποτελέσματα που να σχετίζονται με τα συστήματα συστάσεων. Στο πεδίο της υπολογιστικής όρασης και πιο συγκεκριμένα της αναγνώρισης εικόνας βρίσκουμε αρκετά τέτοια αποτελέσματα όπου μπορούμε να καταλάβουμε ουσιαστικά τη βοήθεια που προσφέρει η μεταφορά γνώσης απο τα προεκπαιδευμένα μοντέλα που έχουμε ήδη αναφέρει.

Η είσοδος στον αλγόριθμό μας θα είναι μια εικόνα οποιουδήποτε αντικείμενου που ο πελάτης θέλει να αγοράσει. Στη συνέχεια χρησιμοποιούμε ένα μοντέλο συνελικτικών νευρωνικών δικτύων (CNN) για την ταξινόμηση της κατηγορίας που αυτό το αντικείμενο πιθανότατα ανήκει και χρησιμοποιεί το διάνυσμα εισόδου του τελευταίου πλήρους συνδεδεμένου στρώματος (fully connected layer) ως διάνυσμα χαρακτηριστικών που τροφοδοτείται σε κάποιο υπολογισμό ομοιότητας για να βρούμε τα πλησιέστερα προϊόντα στη βάση δεδομένων μας. Πιο συγκεκριμένα, οι δύο εργασίες που θέλουμε να επιτύχουμε είναι:

- η κατηγοριοποίηση της φωτογραφίας ενός προϊόντος, σε σχέση με προϊόντα που είναι πιθανότερο να μοιάζουν, π.χ αν εισάγω μια φωτογραφία μίας τσάντας θα πρέπει να κατηγοριοποιηθεί ως "τσάντα"
- η σύσταση παρόμοιων προϊόντων με αυτό που έχουμε εισάγει. Στην ιδανική περίπτωση, όσοι αναζητούν iPhone θα έπρεπε να προτείνονται iPhone

Για οποιεσδήποτε εικόνες στο σύνολο δεδομένων, θα υπάρχει ένα αντίστοιχο διάνυσμα χαρακτηριστικών. Και αυτό το χαρακτηριστικό διάνυσμα θα είναι η συμβολή για το μοντέλο συστάσεων μας, στη συνέχεια υπολογίζουμε το βαθμό ομοιότητας μεταξύ του διανύσματος χαρακτηριστικών της εικόνας στόχου και των διανυσμάτων χαρακτηριστικών όλων των εικόνων στο σύνολό μας. Αυτό επιτυγχάνεται με την απόσταση συνημιτόνου και τη χρήση νευρικών δικτύων.

Σε αυτό το κομμάτι της εργασίας, θα προσπαθήσουμε να δούμε πως η μεταφορά γνώσης, στη δική μας περίπτωση η χρήση των προεκπαιδευμένων μοντέλων επηρεάζει θετικά όσο και την κατηγοριοποίηση μιας εικόνας όσο τις συστάσεις σε ένα σύστημα συστάσεων με βάση την ομοιότητα του συνημιτόνου.

4.1 Αποτελέσματα

4.1.1 Πείραμα 1ο - Δυαδική Κατηγοριοποίηση Εικόνας

Σε αυτό το πρώτο πείραμα θα ασχοληθούμε με το γνωστό σύνολο Dogs vs Cats, το οποίο παρέχεται από το site Kaggle. Για λόγους εξοικονόμησης χρόνου και πόρων θα πάρουμε ένα υποσύνολο 3.000 εικόνων από τις 25.000 που υπάρχουν συνολικά στο σύνολο δεδομένων, το οποίο μας απρέχεται από το Microsoft Research. Στόχος μας είναι να υλοποιήσουμε μία μηχανή ταξινόμησης εικόνων βασιζόμενη στις δύο κλάσεις που είναι χωρισμένες οι εικόνες. Στο παρόν δείγμα έχουν δύο κλάσεις, "γάτα (cat)" και "σκύλος (dog)". Θα πετύχουμε την ταξινόμηση με τη χρήση συνελικτικών νευρωνικών δικτύων (CNN). Θα συγκρίνουμε τα τρία πιο γνωστά προεκπαιδευμένα CNN μοντέλα με ένα μη προεκπαιδευμένο που θα υλοποιήσουμε εμείς.

Για όλες τις υλοποιήσεις μας χρησιμοποιήσαμε τη γλώσσα Python και το API Keras που είναι ένα από τα κορυφαία API νευρωνικών δικτύων. Είναι γραμμένο σε Python και υποστηρίζει πολλαπλές μηχανές υπολογισμού νευρωνικών δικτύων back-end.

Σε ένα πρόβλημα κατηγοριοποίησης για να μπορέσουμε να αξιολογήσουμε σωστά την απόδοση του χρησιμοποιούμε διάφορες μετρικές, όπως είναι η accuracy, precision, recall. Τα δεδομένα μας όμως δεν έχουν ετικέτα, οπότε είναι δύσκολο να υπολογίζουμε τα συγκεκριμένα μέτρα, καθώς εξαρτώνται και από τις προβλεπόμενες τιμές αλλά και από τις πραγματικές. Για να μπορέσουμε εδώ να αξιολογήσουμε την απόδοση του μοντέλου θα εξετάσουμε λίγες εικόνες στις οποίες μπορούμε χειροκίνητα να δημιουργήσουμε ταμπέλα εξ αρχής. Στο πρόβλημά μας, έχουμε δυαδική κατηγοριοποίηση εικόνας, άρα έχουμε δύο κλάσεις, η κλάση 1-σκύλος και η κλάση 0-γάτα. Για να υπολογίσουμε τα μέτρα ορίζουμε ως:

- **TP (True Positive):** να απεικονίζεται στην πραγματικότητα σκύλος (κλάση 1) και να κατηγοριοποιείται ως σκύλος (κλάση 1)
- **TN (True Negative):** να απεικονίζεται στην πραγματικότητα γάτα (κλάση 0) και να κατηγοριοποιείται ως γάτα (κλάση 0)
- **FP (False Positive):** να απεικονίζεται στην πραγματικότητα γάτα (κλάση 0) και να κατηγοριοποιείται ως σκύλος (κλάση 1)
- **FN (False Negative):** να απεικονίζεται στην πραγματικότητα σκύλος (κλάση 1) και να κατηγοριοποιείται ως γάτα (κλάση 0)

Για την αξιολόγηση του μη προεκπαιδευμένου μοντέλου μας θα χρησιμοποιήσουμε 1000 εικόνες από το validation set μας. Όπως αναφέραμε αρχικά, χρησιμοποιούμε ένα υποσύνολο 3000 εικόνων που το διαμοιράζουμε σε 2000 εικόνες εκπαίδευσης και 1000 εικόνες για αξιολόγηση. Με βάση αυτές τις 1000 εικόνες υπολογίζουμε τα παρακάτω μέτρα αλλά και τον confusion matrix. Πιο συγκεκριμένα θα υπολογίσουμε:

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

- **Precision** = $\frac{TP}{TP+FP}$

- **Recall** = $\frac{TP}{TP+FN}$

- **F1-Score** = $\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

Όσον αφορά το κομμάτι των αρχιτεκτονικών, για την υλοποίηση των συνελικτικών νευρωνικών δικτύων, στο κομμάτι των προεκπαιδευμένων μοντέλων (InceptionV3, VGG19, ResNet50), προσθέσαμε ένα πλήρως συνδεδεμένο στρώμα, με 1024 μονάδες και τη συνάρτηση ενεργοποίησης ReLu. Στο τέλος προσθέσαμε το επίπεδο εξόδου και τη συνάρτηση ενεργοποίησης Sigmoid, η οποία δουλεύει καλύτερα σε εργασίες δυαδικής ταξινόμησης. Για το μη προεκπαιδευμένο δίκτυο ξεκινήσαμε με δοκιμές με την πιο απλή δομή, δηλαδή ένα στρώμα συνέλιξης, ένα pooling layer, ένα fully connected layer και τέλος το επίπεδο εξόδου. Η αρχική μορφή του δικτύου δεν παρουσίασε καλά αποτελέσματα για αυτό ενισχύσαμε την πολυπλοκότητα της αρχιτεκτονικής με άλλα δύο στρώματα συνέλιξης με τα αντίστοιχα pooling layers. Επίσης σε όλα τα συνελικτικά δίκτυα χρησιμοποιήσαμε σαν συνάρτηση απώλειας την binary cross entropy.

Μη Προεκπαιδευμένο Συνελικτικό Δίκτυο Σε αυτό το σετ δεδομένων οι εικόνες μας έχουν διαστάσεις (150,150,3). Για να κατασκευάσουμε το μοντέλο μας, χρησιμοποιούμε το Sequential μοντέλο του Tensorflow το οποίο είναι κατάλληλο για μια απλή στοίβα στρωμάτων όπου κάθε στρώμα έχει ακριβώς έναν διάνυσμα εισόδου και έναν διάνυσμα εξόδου. Δίνουμε τις κατάλληλες παραμέτρους που αφορούν το αριθμό των φίλτρων και το μέγεθος του πυρήνα (kernel size). Τα μικρότερα μεγέθη πυρήνα αποτελούνται από 1x1, 2x2, 3x3 και 4x4, ενώ το μεγαλύτερο αποτελείται από 5x5 και ούτω καθεξής, αλλά χρησιμοποιούμε μέχρι 5x5 για 2D Convolution. Το 2012, όταν εισήχθη η αρχιτεκτονική CNN AlexNet, χρησιμοποίησε 11x11 και 5x5 μεγέθη πυρήνα που κατανάλωναν δύο έως τρεις εβδομάδες στην εκπαίδευση. Έτσι, λόγω του εξαιρετικά μεγαλύτερου χρόνου εκπαίδευσης και του ακριβού κόστους, δεν χρησιμοποιούμε πλέον τόσο μεγάλα μεγέθη πυρήνων.

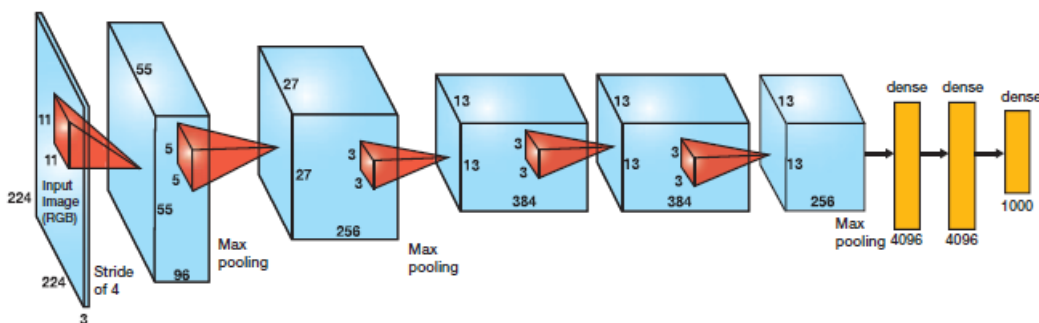


Figure 22:
Αρχιτεκτονική CNN Μοντέλου

Παράμετροι για το μη προεκπαιδευμένο μοντέλο:

- Μέγεθος Εικόνας για είσοδο: (150,150,3)
- Μέγεθος Πυρήνα:3x3
- Αριθμός Κόμβων:
 - 1ο στρώμα: 16
 - 2ο στρώμα: 32
 - 3ο στρώμα: 64
- Χρησιμοποιούμε για συνάρτηση ενεργοποίησης τη συνάρτηση ReLu. Η ReLu θα εξάγει απευθείας το αποτέλεσμα εάν είναι θετικό, διαφορετικά θα εξάγει μηδέν
- Για την αξιολόγηση του μοντέλου και επειδή έχουμε δυαδική ταξινόμηση χρησιμοποιούμε σαν συνάρτηση απώλειας τη δυαδική διασταυρούμενη εντροπία
- Ανάμεσα στα στρώματα συνελίξεων παρεμβάλλονται MaxPool στρώματα. Η μέγιστη συγκέντρωση (Max Pooling) είναι μια λειτουργία συγκέντρωσης που επιλέγει το μέγιστο στοιχείο από την περιοχή του χάρτη χαρακτηριστικών που καλύπτεται από το φίλτρο. Έτσι, η έξοδος μετά το επίπεδο max-pooling θα είναι ένας χάρτης χαρακτηριστικών που περιέχει τα πιο σημαντικά χαρακτηριστικά του προηγούμενου χάρτη χαρακτηριστικών. Έτσι, το στρώμα που γίνεται η συγκέντρωση μειώνει τον αριθμό των παραμέτρων για εκμάθηση και μειώνει τον υπολογισμό στο νευρωνικό δίκτυο. Εδώ επιλέγουμε να είναι μεγέθους 2.
- Για την κατασκευή του επιπέδου εξόδου χρησιμοποιείται η συνάρτηση ενεργοποίησης Sigmoid

Epochs	Running Time	Val Loss	Val Accuracy
1	29s	0.6881	0.5400
2	29s	0.6809	0.5300
3	29s	0.6738	0.5800
4	28s	0.6823	0.5125
5	28s	0.7045	0.5075
6	28s	0.6837	0.5225
7	28s	0.6786	0.5650
8	28s	0.6316	0.6675
9	28s	0.6151	0.6400
10	28s	0.6010	0.6850

Table 2: Αποτελέσματα για απλό CNN μοντέλο

- Χρησιμοποιούμε τον αλγόριθμο RMSProp για την ενημέρωση των βαρών των νευρωνικών δικτύων και του ρυθμού εκμάθησης. Οι βελτιστοποιητές χρησιμοποιούνται για την επίλυση προβλημάτων βελτιστοποίησης ελαχιστοποιώντας τη συνάρτηση.

Θα εκπαιδεύσουμε το CNN μοντέλο μας για 10 epochs, όπως και τα άλλα τρία προεκπαιδευμένα μοντέλα μας.

Όπως βλέπουμε καταφέραμε να επιτύχουμε ακρίβεια επικύρωσης 68.50%. Αναμένουμε να βελτιωθεί με τη χρήση των προεκπαιδευμένων μοντέλων. Από τα παρακάτω διαγράμματα μπορούμε να καταλάβουμε αρκετά για την απόδοση του μοντέλου. Στα διαγράμματα αναπαριστούμε με μπλε χρώμα την επικύρωση (validation) και με κόκκινο την διαδικασία της εκπαίδευσης (training). Φαίνεται ότι όσο αυξάνεται ο αριθμός των epochs αυξάνεται η ακρίβεια και μειώνεται η απώλεια και τόσο στην εκπαίδευση όσο και στην επικύρωση, κάτι που βοηθάει το μοντέλο αλλά σίγουρα μπορεί να βελτιωθεί όπως θα δούμε και παρακάτω. Η απώλεια υπολογίζεται με βάση την εκπαίδευση και την επικύρωση και η διάδρασή της είναι το πόσο καλά τα πάει το μοντέλο για αυτά τα δύο σύνολα, φαίνεται ότι όσον αφορά την απώλεια το μοντέλο ανάμεσα στην απώλεια και στην επικύρωση δουλεύει καλά, βέβαια θα μπορούσαμε να το εκπαιδεύσουμε και παραπάνω. Όσον αφορά το χρόνο εκτέλεσης είναι αρκετά χαμηλός.

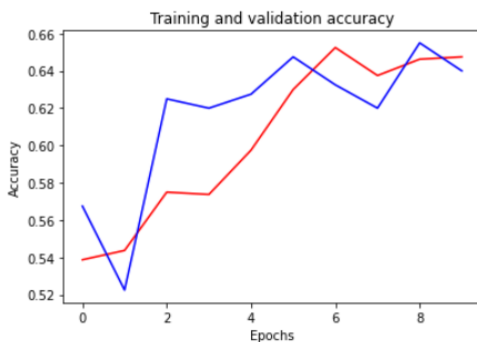


Figure 23: Train-Val Accuracy

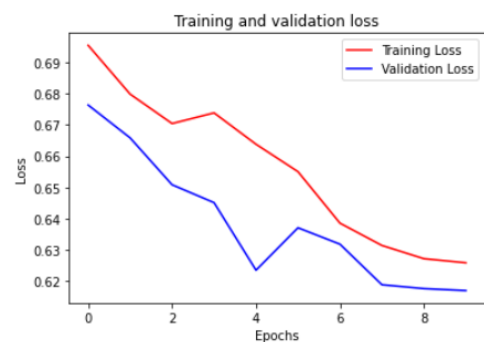


Figure 24: Train-Val Loss



Figure 25:

Η εικόνα αυτή κατηγοριοποιήθηκε λανθασμένα ως "σκύλος"

Παρακάτω παρουσιάζουμε το confusion matrix με βάση τις 1000 εικόνες που έχουμε για αξιολόγηση.

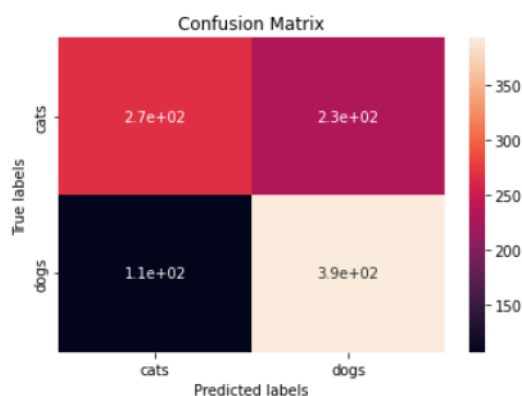


Figure 26:

Confusion Matrix για απλό μη προεκπαιδευμένο δίκτυο

Όπως βλέπουμε, με βάση τις 1000 εικόνες που έχουμε για αξιολόγηση, ο αριθμός των TN (πραγματική κλάση: 0, προβλεπόμενη κλάση: 0) ισούται με 270, ο αριθμός των FN (πραγματική κλάση: 1, προβλεπόμενη κλάση: 0) ισούται με 110, ο αριθμός των TP (πραγματική κλάση: 1, προβλεπόμενη κλάση: 1) ισούται με 390, και τέλος ο αριθμός των FP (πραγματική κλάση: 0, προβλεπόμενη κλάση: 1) ισούται με 230.

Επίσης λαμβάνουμε και το classification report, το οποίο παρουσιάζουμε παρακάτω. Από το συγκεκριμένο report λαμβάνουμε πληροφορίες για πολύ σημαντικές μετρικές που μας αφορούν, όπως precision, recall, accuracy, F1-score. Όπως βλέπουμε, το 63% των εικόνων που προβλέψαμε ότι ανήκουν στην κλάση 1 ("σκύλος"), απεικόνιζαν όντως σκύλο, ενώ το 79% των φωτογραφιών που απεικόνιζαν σκύλο προβλέφθηκε σωστά. Επίσης το F1-Score για την κλάση 0 είναι 61%, ενώ για την κλάση 1 70%. Ιδανικά, θα θέλαμε να είναι κοντά στο 100%, άρα παρατηρούμε ότι το μοντέλο δέχεται βελτιώσεις. Τέλος, έχουμε συνολικό accuracy 66%.

	Precision	Recall	F1-Score	Support
0	0.71	0.54	0.61	500
1	0.63	0.79	0.70	500
accuracy			0.66	1000
macro avg	0.67	0.66	0.66	1000
weighted avg	0.67	0.66	0.66	1000

Table 3: Classification Report για απλό CNN μοντέλο

Προεκπαιδευμένο Μοντέλο VGG19

Στη συνέχεια εκτελούμε το ίδιο πείραμα αλλά χρησιμοποιούμε το προεκπαιδευμένο μοντέλο VGG19. Όπως έχουμε ήδη αναφέρει μπορούμε να κατανοήσουμε το VGG ως διάδοχο του AlexNet, ενώ δημιουργήθηκε από μια ομάδα που ονομάζεται Visual Geometry Group στην Οξφόρδη και εξ ου και το όνομα VGG.

Παράμετροι για το VGG19:

- Το μοντέλο VGG19 αποτελείται από 19 επίπεδα (ανάμεσα σε αυτά είναι 16 στρώματα συνέλιξης και 3 πλήρως συνδεδεμένα στρώματα)
- Αριθμός Κόμβων:
 - 1ο-2ο στρώμα συνέλιξης: 64
 - 3ο-4ο στρώμα συνέλιξης: 128
 - 5ο-16ο στρώμα συνέλιξης: 512
- Χρησιμοποιείται η μέγιστη συγκέντρωση (Max Pooling) όπως και στο μη προεκπαιδευμένο δίκτυο
- Για την κατασκευή του επιπέδου εξόδου χρησιμοποιείται επίσης η συνάρτηση ενεργοποίησης Sigmoid
- Για την αξιολόγηση του μοντέλου και επειδή έχουμε δυαδική ταξινόμηση χρησιμοποιούμε σαν συνάρτηση απώλειας τη δυαδική διασταυρούμενη εντροπία

Epochs	Running Time	Val Loss	Val Accuracy
1	337s	0.5403	0.7525
2	336s	0.3570	0.8275
3	338s	0.3831	0.8313
4	336s	0.2869	0.8825
5	337s	0.3404	0.8525
6	336s	0.3559	0.8325
7	337s	0.2881	0.8700
8	338s	0.2940	0.8625
9	335s	0.3074	0.8700
10	335s	0.3439	0.8350

Table 4: Αποτελέσματα για VGG19 προεκπαιδευμένο μοντέλο

Όπως βλέπουμε καταφέραμε να επιτύχουμε ακρίβεια επικύρωσης 88.25%, πετύχαμε δηλαδή σημαντική βελτίωση σε σχέση με το απλό συνελκτικό δίκτυο. Σε αυτά τα διαγράμματα αναπαριστούμε με κόκκινο χρώμα την εκπαίδευση και με μπλε τη διαδικασία επικύρωσης, όπως και πριν. Από τα παρακάτω διαγράμματα φαίνεται ότι όσο αυξάνεται ο αριθμός των epochs αυξάνεται η ακρίβεια και μειώνεται η απώλεια και τόσο στην εκπαίδευση όσο και στην επικύρωση, όπως και στο προηγούμενο μοντέλο. Όσον αφορά το χρόνο εκτέλεσης είναι πιο υψηλός σε σχέση με πριν, ίσως και λόγω της διαφορετικής αρχιτεκτονικής που χρησιμοποιούμε. Τα καλά αποτελέσματα που πετύχαμε φαίνονται και στις προβλέψεις μας που είναι σαφώς καλύτερες από τις προηγούμενες.

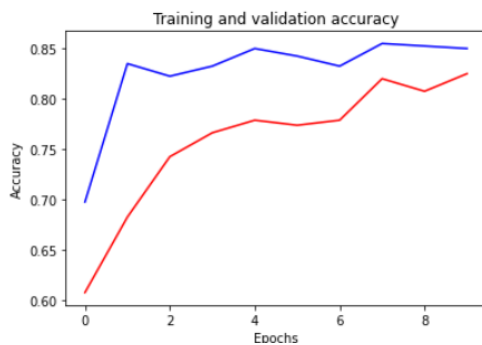


Figure 27: Train-Val Accuracy

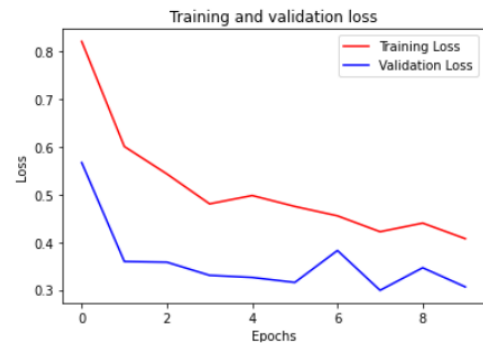


Figure 28: Train-Val Loss



Figure 29:

Η εικόνα αυτή κατηγοριοποιήθηκε σωστά ως "γάτα"

Παρακάτω παρουσιάζουμε το confusion matrix με βάση τις 1000 εικόνες που έχουμε για αξιολόγηση.

Όπως βλέπουμε, με βάση τις 1000 εικόνες που έχουμε για αξιολόγηση, ο αριθμός των TN (πραγματική κλάση: 0, προβλεπόμενη κλάση: 0) ισούται με 450, ο αριθμός των FN (πραγματική κλάση: 1, προβλεπόμενη κλάση: 0) ισούται με 81, ο αριθμός των TP (πραγματική κλάση: 1, προβλεπόμενη κλάση: 1) ισούται με 420, και τέλος ο αριθμός των FP (πραγματική κλάση: 0, προβλεπόμενη κλάση: 1) ισούται με 54.

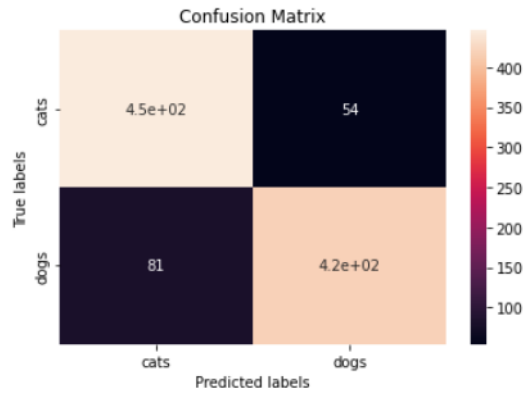


Figure 30:
Confusion Matrix για το προεκπαιδευμένο δίκτυο VGG19

Επίσης λαμβάνουμε και το classification report, το οποίο παρουσιάζουμε παρακάτω. Από το συγκεκριμένο report λαμβάνουμε πληροφορίες για πολύ σημαντικά μέτρα που μας αφορούν, όπως precision, recall, accuracy, F1-score. Όπως βλέπουμε, το 89% των εικόνων

	Precision	Recall	F1-Score	Support
0	0.85	0.89	0.87	500
1	0.89	0.84	0.86	500
accuracy			0.86	1000
macro avg	0.87	0.86	0.86	1000
weighted avg	0.87	0.86	0.86	1000

Table 5: Classification Report για VGG19 προεκπαιδευμένο μοντέλο

που προβλέψαμε ότι ανήκουν στην κλάση 1 ("σκύλος"), απεικόνιζαν όντως σκύλο, ενώ το 84% των φωτογραφιών που απεικόνιζαν σκύλο προβλέφθηκε σωστά. Επίσης το F1-Score για την κλάση 1 είναι 87% και για την κλάση 0 86%, που είναι κοντά στο 100%, πράγμα που σημαίνει το μοντέλο αυτό δουλεύει καλά. Τέλος το μοντέλο παρουσιάζει συνολικό accuracy που είναι 86%.

Προεκπαιδευμένο Μοντέλο InceptionV3

Το InceptionV3 είναι ένα μοντέλο βαθιάς μάθησης που βασίζεται σε συνελικτικά νευρωνικά δίκτυα, το οποίο χρησιμοποιείται για ταξινόμηση εικόνων. Το Inception V3 είναι μια ανώτερη έκδοση του βασικού μοντέλου Inception V1 που παρουσιάστηκε ως GoogLeNet το 2014. Όπως υποδηλώνει το όνομα, αναπτύχθηκε από μια ομάδα της Google. Όπως βλέπουμε η αρχική μορφή του μοντέλου InceptionV1 έχει ένα συνελικτικό στρώμα 5×5 το οποίο ήταν υπολογιστικά ακριβό. Έτσι, για να μειωθεί το υπολογιστικό κόστος, το συνελικτικό στρώμα 5×5 αντικαταστάθηκε από δύο συνελικτικές στρώσεις 3×3 όπως φαίνεται παρακάτω και έτσι δημιουργήθηκε το InceptionV3 που μελετάμε σήμερα. Το Inception v3 είναι ένα μοντέλο αναγνώρισης εικόνας που έχει αποδειχθεί ότι επιτυγχάνει μεγαλύτερη από 78,1% ακρίβεια στο σύνολο δεδομένων ImageNet. Το μοντέλο είναι το αποκορύφωμα πολλών ιδεών που αναπτύχθηκαν από πολλούς ερευνητές όλα αυτά τα χρόνια. Βασίζεται στην αρχική εργασία: «Rethinking the Inception Architecture for Computer Vision» των Szegedy, et. al. Το ίδιο το μοντέλο αποτελείται από συμμετρικά και ασύμμετρα δομικά στοιχεία, συμπεριλαμβανομένων των συνελίξεων, της μέσης συγκέντρωσης, της μέγιστης συγκέντρωσης, των συνενώσεων, των εγκαταλείψεων και των πλήρως συνδεδεμένων στρωμάτων. Η κανονικοποίηση παρτίδας χρησιμοποιείται εκτενώς σε όλο το μοντέλο και εφαρμόζεται στις εισόδους ενεργοποίησης. Η απώλεια υπολογίζεται χρησιμοποιώντας τη συνάρτηση Softmax. Το μοντέλο Inceptionv3, έχει συνολικά 42 επίπεδα και χαμηλότερο ποσοστό σφάλματος από τους προκατόχους του.

Η βασική μορφή του μοντέλου InceptionV1 αποτελείται από τέσσερα παράλληλα στρώματα.

- 1×1 στρώμα συνέλιξης
- 3×3 στρώμα συνέλιξης
- 5×5 στρώμα συνέλιξης
- 3×3 στρώμα συγκέντρωσης Max Pooling

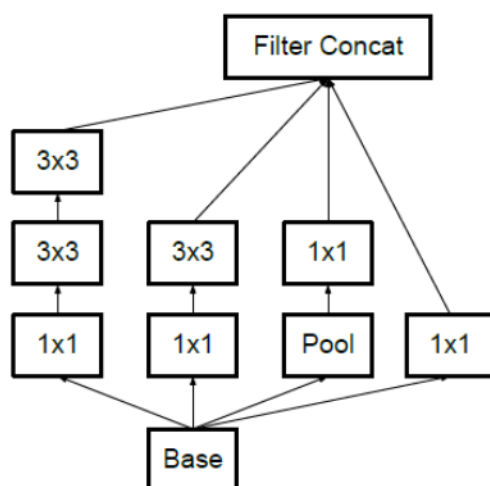


Figure 31:
Αρχιτεκτονική InceptionV3 [63]

Epochs	Running Time	Val Loss	Val Accuracy
1	65s	0.3160	0.9300
2	60s	0.1713	0.9470
3	59s	0.2016	0.9430
4	59s	0.5510	0.8900
5	59s	0.1325	0.9550
6	59s	0.1162	0.9570
7	60s	0.3310	0.9190
8	59s	0.4105	0.8860
9	59s	0.1695	0.9560
10	61s	0.4170	0.9050

Table 6: Αποτελέσματα για προεκπαιδευμένο μοντέλο InceptionV3

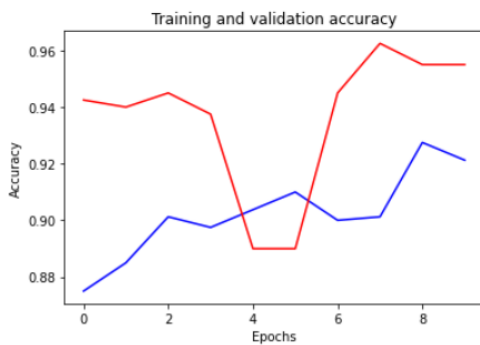


Figure 32: Train-Val Accuracy

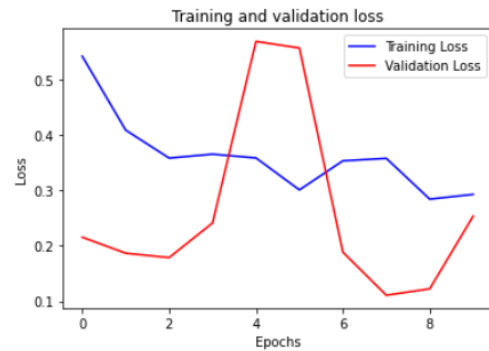


Figure 33: Train-Val Loss

Με τη χρήση του μοντέλου InceptionV3 καταφέραμε να επιτύχουμε την πιο υψηλή ακρίβεια επικύρωσης 95.70%, όμως από τα παρακάτω διαγράμματα, με μπλε σημειώνουμε τη διαδικασία εκπαίδευσης και με κόκκινο την επικύρωση, φαίνεται ότι όσο αυξάνεται ο αριθμός των epochs δεν αυξάνεται ομαλά ούτε η ακρίβεια και ούτε μειώνεται η απώλεια με καλό ρυθμό και τόσο στην εκπαίδευση όσο και στην επικύρωση, όπως και στο προηγούμενο μοντέλο. Ειδικότερα, στο διάγραμμα της ακρίβειας στην 3η περίπου epoch η ακρίβεια επικύρωσης μειώνεται αισθητά και αυξάνεται πάλι, ενώ στο διάγραμμα της απώλειας στο ίδιο ακριβώς σημείο η απώλεια επικύρωσης δείχνει να αυξάνεται σε σχέση με αυτή της εκπαίδευσης, κάτι που μας οδηγεί να υποθέσουμε ότι το μοντέλο μας είναι overfit, το οποίο σημαίνει ότι το μοντέλο «απομνημονεύει» τα παραδείγματα εκπαίδευσης και ίσως να μην είναι αποτελεσματικό όταν θα χρειαστεί να το γενικεύσουμε. Όσον αφορά το χρόνο εκτέλεσης είναι σχετικά χαμηλός.

Παρακάτω παρουσιάζουμε το confusion matrix με βάση τις 1000 εικόνες που έχουμε για αξιολόγηση.

Όπως βλέπουμε, με βάση τις 1000 εικόνες που έχουμε για αξιολόγηση, ο αριθμός των TN (πραγματική κλάση: 0, προβλεπόμενη κλάση: 0) ισούται με 480, ο αριθμός των FN (πραγματική κλάση: 1, προβλεπόμενη κλάση: 0) ισούται με 12, ο αριθμός των TP (πραγ-

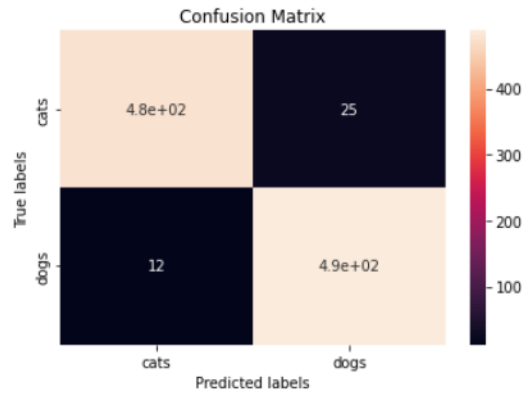


Figure 34:

Confusion Matrix για το προεκπαιδευμένο δίκτυο InceptionV3

ματική κλάση: 1, προβλεπόμενη κλάση: 1) ισούται με 490, και τέλος ο αριθμός των FP (πραγματική κλάση: 0, προβλεπόμενη κλάση: 1) ισούται με 25.

Επίσης λαμβάνουμε και το classification report, το οποίο παρουσιάζουμε παρακάτω. Από το συγκεκριμένο report λαμβάνουμε πληροφορίες για πολύ σημαντικά μέτρα που μας αφορούν, όπως precision, recall, accuracy, F1-score. Όπως βλέπουμε, το 96% των εικόνων

	Precision	Recall	F1-Score	Support
0	0.98	0.95	0.96	500
1	0.95	0.98	0.96	500
accuracy			0.96	1000
macro avg	0.96	0.96	0.96	1000
weighted avg	0.96	0.96	0.96	1000

Table 7: Classification Report για InceptionV3 προεκπαιδευμένο μοντέλο

που προβλέψαμε ότι ανήκουν στην κλάση 1 ("σκύλος"), απεικόνιζαν όντως σκύλο, ενώ το 98% των φωτογραφιών που απεικόνιζαν σκύλο προβλέφθηκε σωστά. Επίσης το F1-Score και για τις δύο κλάσεις είναι 96%, που είναι πολύ κοντά στο 100%, πράγμα που σημαίνει το μοντέλο αυτό δουλεύει αρκετά καλά. Τέλος αυτό επιβεβαιώνεται και από το συνολικό accuracy που είναι 96%.

Μέχρι στιγμής έχουμε δοκιμάσει ένα απλό συνελικτικό νευρωνικό δίκτυο και δύο προεκπαιδευμένα (VGG19, InceptionV3), τα οποία βάσει των επιδόσεων τους όντως βελτιστοποίησαν την εργασία ταξινόμησής μας.

Θα δοκιμάσουμε άλλα ένα προεκπαιδευμένο μοντέλο το ResNet50.

Προεκπαιδευμένο μοντέλο ResNet50

Το 2012 στον διαγωνισμό ταξινόμησης LSVRC2012, το AlexNet κέρδισε την πρώτο βραβείο, μετά από αυτό το ResNet ήταν το πιο ενδιαφέρον πράγμα που συνέβη στην υπολογιστική όραση και στον κόσμο της βαθιάς μάθησης.

Λόγω του πλαισίου που παρουσίασε το ResNet, κατέστη δυνατή η εκπαίδευση εξαιρετικά βαθιών νευρωνικών δικτύων, δηλαδή δικτύων που μπορεί να περιέχουν εκατοντάδες ή χιλιάδες επίπεδα και να εξακολουθεί να επιτυγχάνει εξαιρετική απόδοση.

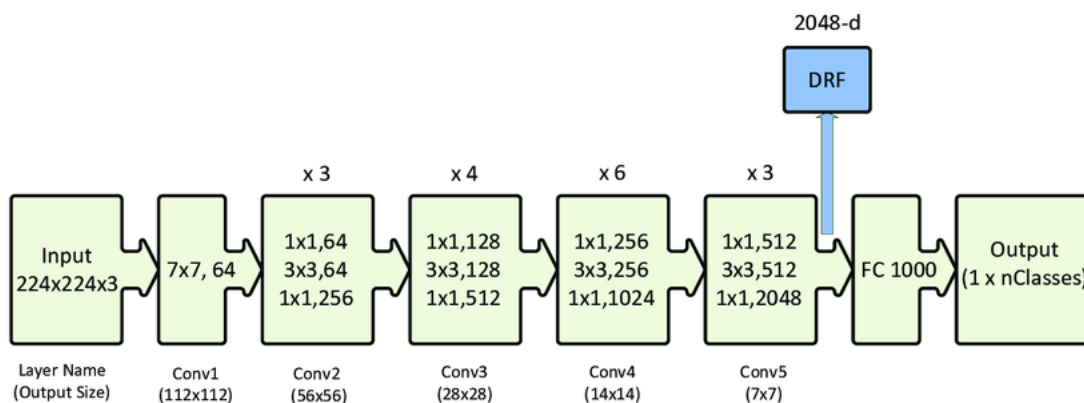


Figure 35:
Αρχιτεκτονική ResNet50 Μοντέλου [62]

Αρχιτεκτονική ResNet50:

- Μια συνένωση με μέγεθος πυρήνα $7 * 7$ και 64 διαφορετικούς πυρήνες, όλα με διασκελισμό μεγέθους 2 που μας δίνει 1 στρώμα.
- Στη συνέχεια έχουμε ένα επίπεδο συγκέντρωσης max pooling με επίσης μέγεθος βήματος 2.
- Στην επόμενη συνέλιξη υπάρχει ένας πυρήνας $1 * 1$, 64 που ακολουθεί έναν πυρήνα $3 * 3$, 64 και στο τέλος ένας πυρήνας $1 * 1$, 256. Αυτά τα τρία επίπεδα επαναλαμβάνονται συνολικά 3 φορές, οπότε μας δίνουν 9 επίπεδα σε αυτό το βήμα.
- Στη συνέχεια βλέπουμε πυρήνα $1 * 1$, 128 μετά από αυτό έναν πυρήνα $3 * 3$, 128 και έναν πυρήνα $1 * 1$, 512 αυτό το βήμα επαναλήφθηκε 4 φορές, δίνοντάς μας 12 επίπεδα σε αυτό το βήμα.
- Μετά από αυτό υπάρχει ένας πυρήνας $1 * 1$, 256 και δύο ακόμη πυρήνες με $3 * 3$, 256 και $1 * 1$, 1024 και αυτό επαναλαμβάνεται 6 φορές δίνοντας μας συνολικά 18 επίπεδα.
- Και μετά πάλι ένας πυρήνας $1 * 1$, 512 με δύο ακόμη $3 * 3$, 512 και $1 * 1$, 2048 και αυτό επαναλήφθηκε 3 φορές δίνοντάς μας συνολικά 9 επίπεδα.
- Μετά από αυτό κάνουμε ένα average pool και το τελειώνουμε με ένα πλήρως συνδεδεμένο στρώμα που περιέχει 1000 κόμβους και στο τέλος μια συνάρτηση softmax οπότε αυτό μας δίνει 1 στρώμα.

Συνολικά αν αθροίσουμε τον αριθμό των επιπέδων έχουμε 50, εξού και το όνομα ResNet50.

Με τη χρήση του μοντέλου ResNet50 καταφέραμε να επιτύχουμε ακρίβεια επικύρωσης 68.70%, όμως από τα παρακάτω διαγράμματα, με μπλε σημειώνουμε τη διαδικασία εκ-

Epochs	Running Time	Val Loss	Val Accuracy
1	117s	0.6789	0.5975
2	109s	1.2064	0.4750
3	110s	0.9001	0.5050
4	109s	0.6994	0.5075
5	110s	0.6764	0.5775
6	110s	0.6329	0.6870
7	111s	0.6814	0.6050
8	109s	0.7020	0.5050
9	111s	0.6791	0.6050
10	109s	0.6772	0.6000

Table 8: Αποτελέσματα για ResNet50 προεκπαιδευμένο μοντέλο



Figure 36: Train-Val Accuracy

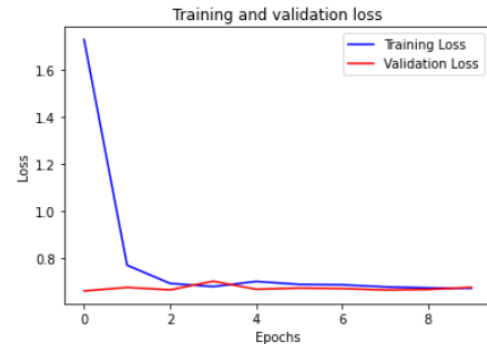


Figure 37: Train-Val Loss

παίδευσης και με κόκκινο την επικύρωση, φαίνεται ότι όσο αυξάνεται ο αριθμός των epochs η ακρίβεια επικύρωσης δεν βελτιώνεται σταθερά, ενώ η απώλεια εκπαίδευσης και επικύρωσης ελαχιστοποιούνται όσο περνάνε οι epochs. Θα μπορούσαμε να επιλύσουμε κάποια από τα παραπάνω ζητήματα αν εκπαιδεύαμε το μοντέλο για παραπάνω epochs, αφού και ο χρόνος εκτέλεσης δεν είναι καταστροφικά μεγάλος.

Παραπάνω παρουσιάζουμε το confusion matrix με βάση τις 1000 εικόνες που έχουμε για αξιολόγηση.

Όπως βλέπουμε, με βάση τις 1000 εικόνες που έχουμε για αξιολόγηση, ο αριθμός των TN (πραγματική κλάση: 0, προβλεπόμενη κλάση: 0) ισούται με 400, ο αριθμός των FN (πραγματική κλάση: 1, προβλεπόμενη κλάση: 0) ισούται με 310, ο αριθμός των TP (πραγματική κλάση: 1, προβλεπόμενη κλάση: 1) ισούται με 190, και τέλος ο αριθμός των FP (πραγματική κλάση: 0, προβλεπόμενη κλάση: 1) ισούται με 100.

Επίσης λαμβάνουμε και το classification report, το οποίο παρουσιάζουμε παρακάτω. Από το συγκεκριμένο report λαμβάνουμε πληροφορίες για πολύ σημαντικά μέτρα που μας αφορούν, όπως precision, recall, accuracy, F1-score.

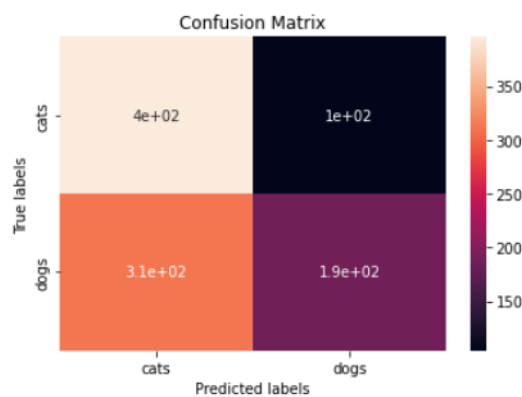


Figure 38:
Confusion Matrix για προεκπαιδευμένο δίκτυο ResNet50

	Precision	Recall	F1-Score	Support
0	0.56	0.79	0.66	500
1	0.64	0.38	0.47	500
accuracy			0.58	1000
macro avg	0.60	0.58	0.57	1000
weighted avg	0.60	0.58	0.57	1000

Table 9: Classification Report για ResNet50 προεκπαιδευμένο μοντέλο

Όπως βλέπουμε, το 64% των εικόνων που προβλέψαμε ότι ανήκουν στην κλάση 1 ("σκύλος"), απεικόνιζαν όντως σκύλο, ενώ το 38% των φωτογραφιών που απεικόνιζαν σκύλο προβλέφθηκε σωστά. Επίσης το F1-Score για την κλάση 0 είναι 66%, ενώ για την κλάση 1 47%. Ιδανικά, θα θέλαμε να είναι κοντά στο 100%, άρα παρατηρούμε ότι το μοντέλο δέχεται βελτιώσεις. Τέλος, έχουμε συνολικό accuracy 58%.

Παρακάτω θα συγκρίνουμε τις ακρίβειες επικύρωσης των τεσσάρων μοντέλων:

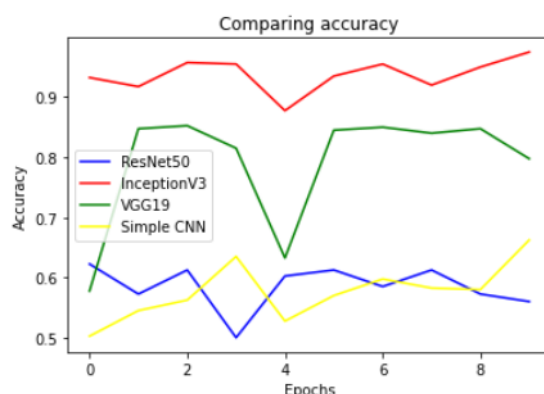


Figure 39: Σύγκριση Μοντέλων

Στο παρακάτω διάγραμμα συγκρίνουμε τις ακρίβειες επικύρωσης των τεσσάρων μοντέλων που χρησιμοποιήσαμε. Είναι εμφανές ότι καλύτερη ακρίβεια επικύρωσης έχει πετύχει

το προεκαπιδευμένο μοντέλο InceptionV3 που όπως αναφέραμε παραπάνω συμπεριφέρεται "καλύτερα", στη συνέχεια ακολουθεί το προεκαπιδευμένο μοντέλο VGG19. Τέλος, παρατηρούμε ότι τα άλλα δύο μοντέλα το απλό συνελικτικό και το προεκαπιδευμένο μοντέλο ResNet50 ακολουθούν με χαμηλότερες επιδόσεις στην ακρίβεια επικύρωσης.

4.1.2 Πείραμα 2ο: Content-Based Recommendation System

Στο δεύτερο πείραμα μας θα συγκρίνουμε την απόδοση ενός απλού μη προεκπαιδευμένου μοντέλου και δύο προεκπαιδευμένων (InceptionV3, ResNet50).

Το σετ δεδομένων που θα χρησιμοποιήσουμε είναι το Fashion Product Images Dataset, το οποίο περιέχει 44.000 εικόνες από διάφορα είδη ρουχισμού. Στο σύνολό μας περιέχονται και άλλες πληροφορίες που αφορούν το χρώμα, τη μάρκα αλλά αυτή τη στιγμή δε θα μας απασχολήσουν.

Για όλες τις υλοποιήσεις μας χρησιμοποιήσαμε πάλι το API Keras, και πλήθος βιβλιοθηκών που προσφέρει. Επίσης χωρίσαμε τα δεδομένα μας σε 80% δεδομένα εκπαίδευσης και 20% επικύρωσης με τη βοήθεια του ImageGenerator και για λόγους εξοικονόμησης πόρων και χρόνου δεν εκπαιδεύσαμε τα μοντέλα σε ολόκληρο το σύνολο δεδομένων αλλά χρησιμοποιήσαμε δείγμα 2000 φωτογραφιών.

Για την υλοποίηση των συνελικτικών νευρωνικών δικτύων ακολουθήσαμε παρόμοιες τεχνικές με το πρώτο πείραμα. Στο μη προεκπαιδευμένο συνελικτικό δίκτυο έχουμε τρία επίπεδα συνέλιξης, ένα πλήρως συνδεδεμένο επίπεδο, και ένα επίπεδο εξόδου με συνάρτηση ενεργοποίησης Softmax λόγω του multiclass classification. Χρησιμοποιήσαμε παρόμοιες αρχιτεκτονικές για τα προεκπαιδευμένα μοντέλα, όπως και στο πρώτο πείραμα απλά αντί για τη συνάρτηση Sigmoid επιλέξαμε τη Softmax. Μια διαφορά σε σχέση με το πρώτο πείραμα, είναι ότι στο πρώτο πείραμα χρησιμοποιήσα τη συνάρτηση απώλειας binary cross entropy ενώ εδώ τη categorical cross entropy. Όσον αφορά το σύνολο δεδομένων, με την εντολή sample πήραμε ένα υποσύνολο 2000 εικόνων, τις οποίες χωρίσαμε σε 1600 εικόνες εκπαίδευσης και 400 αξιολόγησης, όπως αναφέραμε και παραπάνω.

Για όλες τις υλοποιήσεις αφού αξιολογήσαμε τα μοντέλα στο τέλος προχωρήσαμε στη δημιουργία ενός συστήματος συστάσεων με βάση κάποιο μέτρο ομοιότητας. Πιο συγκεκριμένα, προχωράμε στη πρόβλεψη από το μοντέλο και μετατρέπουμε όλες τις εικόνες σε διανύσματα στη συνέχεια με βάση την ομοιότητα τους προτείνονται παρόμοιες εικόνες. Στην δικιά μας περίπτωση θέλουμε αν έχουμε στη φωτογραφία ένα συγκεκριμένο είδος ρουχισμού να μας προταθούν ανάλογα. Δηλαδή αν στη φωτογραφία έχουμε γυναικεία μπλούζα, να μας προταθούν γυναικείες μπλούζες. Στην παρούσα έρευνα χρησιμοποιούμε την ομοιότητα συνημιτόνου, την οποία έχουμε αναφέρει και ορίσει στο Κεφάλαιο 2.

Προεκπαιδευμένο Μοντέλο InceptionV3

Θα εκπαιδεύσουμε το μοντέλο για 10 epochs όπως και στο προηγούμενο πείραμα. Το μέγεθος της εικόνας μας είναι (224,224,3). Πάλι θα παγώσουμε τα ενδιάμεσα layers, και για το output χρησιμοποιούμε global average pooling layer. Για την αξιολόγηση της απώλειας χρησιμοποιούμε τη συνάρτηση Categorical Cross-Entropy και για βελτιστοποιητή τον αλγόριθμο Adam.

Epochs	Running Time	Val Loss	Val Accuracy
1	114s	24.3784	0.4609
2	98s	10.2226	0.5195
3	96s	7.4763	0.5039
4	95s	5.7274	0.5117
5	96s	3.2226	0.5664
6	96s	2.8627	0.6289
7	96s	2.0242	0.6484
8	96s	1.7953	0.6445
9	94s	1.8246	0.6484
10	95s	1.6098	0.6367

Table 10: Αποτελέσματα για προεκπαιδευμένο μοντέλο InceptionV3

Όπως βλέπουμε το προεκπαιδευμένο μοντέλο InceptionV3 δεν παρουσιάζει πολύ καλά αποτελέσματα. Για την εκπαίδευση χρησιμοποιήσαμε 20 βήματα ανά εποχή, και επίσης όχι ολόκληρο το σύνολο δεδομένων. Για να δούμε καλύτερα αποτελέσματα θα μπορούσαμε να αυξήσουμε το σύνολο εκπαίδευσης, τον αριθμό των βημάτων.

Από τα παρακάτω διαγράμματα, με μπλε σημειώνουμε τη διαδικασία εκπαίδευσης και με κόκκινο την επικύρωση, είναι εύκολο να παρατηρήσουμε ότι η ακρίβεια επικύρωσης του μοντέλου είναι χαμηλή, ενώ η ακρίβεια εκπαίδευσης υψηλή, και ότι στο διάγραμμα απώλειας φαίνεται ξεκάθαρα ότι το μοντέλο InceptionV3 είναι underfit. Ένα underfit μοντέλο υποδεικνύει ότι το μοντέλο δεν ήταν σε θέση να μάθει καθόλου το σύνολο δεδομένων εκπαίδευσης. Αυτό μπορεί να συμβαίνει όταν το μοντέλο δεν έχει την κατάλληλη χωρητικότητα για την πολυπλοκότητα του συνόλου δεδομένων. Συνοψίζοντας, το προεκπαιδευμένο μοντέλο δεν έχει "καλή" απόδοση και φαίνεται και στο σύστημα συστάσεων που υλοποιούμε παρακάτω.

Θα προβλέψουμε τις συστάσεις για τις πρώτες 10 εικόνες του συνόλου δεδομένων μας. Στον παρακάτω πίνακα φαίνονται τα μέτρα ομοιότητας για κάθε πρόβλεψη. Εδώ είναι σημαντικό να σημειώσουμε ότι οι παρακάτω ομοιότητες συνημιτόνου, υπολογίζονται με βάση τις ενσωματώσεις (embeddings) που προβλέπει και υπολογίζει το νευρωνικό δίκτυο. Άρα θα μπορούσαμε να συνδέσουμε τους υπολογισμούς αυτούς με την επιτυχία ή την αποτυχία του μοντέλου.

Παρακάτω παραθέτουμε και τις συστάσεις των προϊόντων γραφικά. Όπως βλέπουμε γραφικά επιβεβαιώνονται οι μετρικές ομοιότητας σε σχέση με τα προϊόντα που προτείνονται.

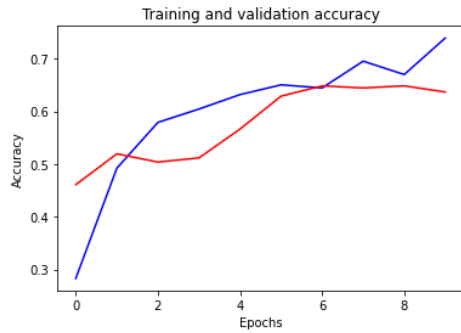


Figure 40: Train-Val Accuracy

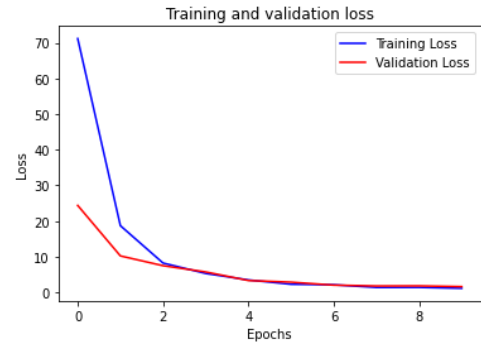


Figure 41: Train-Val Loss

Εικόνα	Σύσταση 1	Σύσταση 2	Σύσταση 3	Σύσταση 4	Σύσταση 5	Σύσταση 6
1	0.99937016	0.9993059	0.99894875	0.9979685	0.99792224	0.99791312
2	0.93320304	0.90883017	0.906594	0.90648514	0.90317136	0.90235768
3	0.9794764	0.9719831	0.9666896	0.9642074	0.9632404	0.9621871
4	0.9869534	0.98512095	0.9794982	0.9753118	0.97201985	0.96335687
5	0.9941272	0.993582	0.98854136	0.9881924	0.9842353	0.9827689
6	0.99811035	0.9930333	0.9814514	0.97313356	0.9641891	0.9632567
7	0.99906135	0.99708885	0.9956081	0.99320483	0.9911238	0.9910685
8	0.99998355	0.999893	0.9998913	0.99987745	0.9998706	0.9985465
9	0.995195	0.99295914	0.9923822	0.9920556	0.9919858	0.9915763
10	0.97843194	0.9753174	0.9744554	0.97207594	0.96949387	0.9641891

Table 11: Συστάσεις για προεκπαιδευμένο μοντέλο InceptionV3

Για παράδειγμα εκεί που έχουμε χαμηλότερο δείκτη ομοιότητας, το προϊόν δεν είναι τόσο σχετικό.



Figure 42:
Συστάσεις των προϊόντων 1,2,3



Figure 43:
Συστάσεις των προϊόντων 4,5,6



Figure 44:
Συστάσεις των προϊόντων 7,8,9,10

Προεκπαιδευμένο μοντέλο ResNet50

Θα εκπαιδεύσουμε το μοντέλο για 10 epochs όπως και στο προηγούμενο πείραμα. Το μέγεθος της εικόνας μας είναι (224,224,3). Πάλι θα παγώσουμε τα ενδιάμεσα layers, και για το output χρησιμοποιούμε global average pooling layer. Για την αξιολόγηση της απώλειας χρησιμοποιούμε τη συνάρτηση Categorical Cross-Entropy και για βελτιστοποιητή τον αλγόριθμο Adam. Με ακριβώς τις ίδιες παραμέτρους το ResNet αποδίδει πολύ καλύτερα.

Epochs	Running Time	Val Loss	Val Accuracy
1	107s	0.6890	0.8008
2	101s	0.5228	0.8594
3	102s	0.4784	0.8750
4	102s	0.5167	0.8594
5	103s	0.4341	0.8750
6	102s	0.5719	0.8633
7	102s	0.3983	0.8828
8	102s	0.3423	0.9062
9	101s	0.4593	0.9023
10	101s	0.3541	0.9102

Table 12: Αποτελέσματα για προεκπαιδευμένο μοντέλο ResNet50

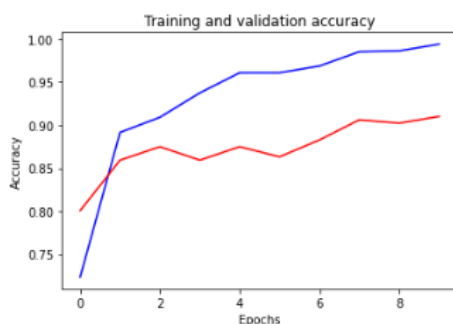


Figure 45: Train-Val Accuracy

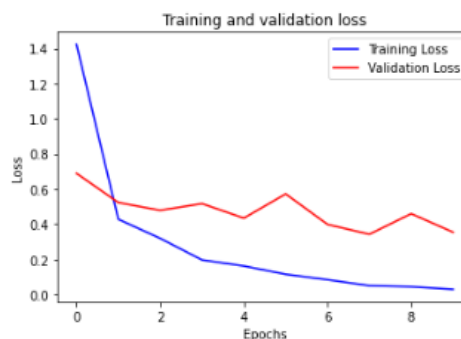


Figure 46: Train-Val Loss

Σύμφωνα με τα παραπάνω διαγράμματα, με μπλε σημειώνουμε τη διαδικασία εκπαίδευσης και με κόκκινο την επικύρωση, με τη χρήση του μοντέλου ResNet50 έχουμε πετύχει καλύτερη ακρίβεια επικύρωσης σε σχέση με το προηγούμενο μοντέλο 91.02%, επίσης παρατηρούμε ότι στο διάγραμμα η απώλεια εκπαίδευσης μειώνεται και ταυτόχρονα η απώλεια επικύρωσης παρουσιάζει μια μικρή τάση αύξησης και αυτό συμβαίνει μετά τα τέσσερα epochs. Τα αποτελέσματα που παίρνουμε όμως είναι καλύτερα από του προηγούμενου μοντέλου αναφορικά με τις προβλέψεις.

Θα προβλέψουμε τις συστάσεις για τις πρώτες 10 εικόνες του συνόλου δεδομένων μας. Στον παρακάτω πίνακα φαίνονται τα μέτρα ομοιότητας για κάθε πρόβλεψη, τα μέτρα αυτά υπολογίστηκαν με βάση το embedding της δοθείσας εικόνας και κάθε φορά με το embedding της εικόνας που προτείνει το δίκτυο.

Εικόνα	Σύσταση 1	Σύσταση 2	Σύσταση 3	Σύσταση 4	Σύσταση 5	Σύσταση 6
1	0.9999998	0.99999917	0.99999905	0.999999	0.9999987	0.9999976
2	0.9999998	0.999999	0.99999905	0.999999	0.9999987	0.9999976
3	0.9999999	0.999999	0.99999976	0.9999997	0.99999964	0.999999
4	0.9999999	0.9999998	0.99999976	0.9999997	0.99999964	0.9999996
5	0.999999	0.999999	0.999999	0.999999	0.999999	0.9999931
6	0.99811035	0.9930333	0.9814514	0.97313356	0.9641891	0.9632567
7	0.99906135	0.99708885	0.9956081	0.99320483	0.9911238	0.9910685
8	0.9999997	0.99999964	0.99999875	0.99999875	0.99999475	0.9999931
9	0.9999998	0.9999998	0.99999934	0.999999	0.9999976	0.9999971
10	0.99999243	0.99998766	0.9999756	0.9999742	0.9999696	0.9999689

Table 13: Συστάσεις για προεκπαιδευμένο μοντέλο InceptionV3

Παρακάτω παραθέτουμε και τις συστάσεις των προϊόντων γραφικά. Όπως βλέπουμε γραφικά επιβεβαιώνονται οι μετρικές ομοιότητας σε σχέση με τα προϊόντα που προτείνονται. Για παράδειγμα εκεί που έχουμε χαμηλότερο δείκτη ομοιότητας, το προϊόν δεν είναι τόσο σχετικό.



Figure 47:
Συστάσεις των προϊόντων 1,2,3



Figure 48:
Συστάσεις των προϊόντων 4,5,6



Figure 49:
Συστάσεις των προϊόντων 7,8,9,10

Μη Προεκπαιδευμένο Συνελικτικό Δίκτυο

Θα χρησιμοποιήσουμε εδώ πάλι το ίδιο σύνολο δεδομένων. Για την κατασκευή του συνελικτικού μοντέλου χρησιμοποιήσαμε 3 στρώματα με διαφορετικό αριθμό φίλτρων αλλά όλα μεγέθους 3×3 . Ακόμη, έχουμε 3 επίπεδα είναι μια λειτουργία συγκέντρωσης (pooling layer) με μέγεθος φίλτρου 2×2 . Τέλος, έχουμε ένα πλήρως συνδεδεμένο επίπεδο με 35 φίλτρα και το επίπεδο εξόδου το οποίο ενεργοποιείται με τη συνάρτηση softmax. Χωρίζουμε τα δεδομένα μας σε 80% εκπαίδευσης και 20% δοκιμής και εκπαιδεύουμε τα δεδομένα μας για 10 epochs. Και σε αυτή την υλοποίηση θα χρησιμοποιήσουμε σαν μετρική ομοιότητας την ομοιότητα συνημιτόνου. Παρακάτω παρουσιάζουμε τα αποτελέσματα:

Epochs	Running Time	Val Loss	Val Accuracy
1	78s	52.8325	0.3516
2	77s	2.1621	0.4883
3	76s	1.9144	0.5391
4	75s	1.9367	0.4528
5	75s	1.6334	0.6172
6	75s	1.7228	0.6133
7	79s	1.9017	0.6406
8	78s	0.9098	0.6680
9	77s	0.8433	0.6992
10	79s	0.8339	0.7070

Table 14: Αποτελέσματα για απλό CNN μοντέλο



Figure 50: Train-Val Accuracy

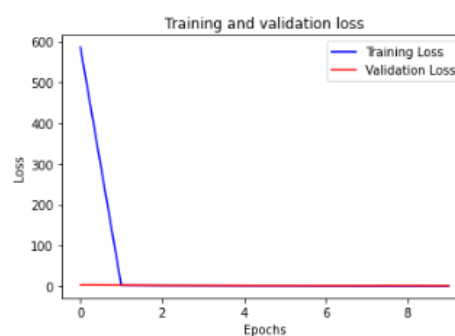


Figure 51: Train-Val Loss

Με τα παραπάνω διαγράμματα, με μπλε σημειώνουμε τη διαδικασία εκπαίδευσης και με κόκκινο την επικύρωση, παρατηρούμε ότι η ακρίβεια επικύρωσης δεν είναι και πολύ υψηλή, πετύχαμε 70.7%. Παρατηρώντας το διάγραμμα απώλειας δεν μπορούμε να βγάλουμε σαφή συμπεράσματα, όμως από τα αποτελέσματα μπορούμε να γνωρίζουμε ότι μετά την πρώτη epoch σε όλη τη διάρκεια εκπαίδευσης η απώλεια εκπαίδευσης είναι πάντα μικρότερη από την απώλεια επικύρωσης αλλά και οι δύο έχουν την τάση να μειώνονται που αυτό μας οδηγεί να συμπεράνουμε ότι το μοντέλο μας τείνει να είναι underfit, δηλαδή το μοντέλο μας δεν μπορεί να καταγράψει τη σχέση μεταξύ των μεταβλητών εισόδου και εξόδου με

ακρίβεια, δημιουργώντας υψηλό ποσοστό σφάλματος τόσο στο σύνολο εκπαίδευσης όσο και στα μη ορατά δεδομένα. Μπορούμε να προχωρήσουμε σε κινήσεις που θα βλετιώσουν την απόδοση του απλού συνελικτικού μοντέλου, όπως να πειραματιστούμε με το learning rate ή να προσθέσουμε κρυφά επίπεδα στο συνελικτικό μας δίκτυο.

Θα προβλέψουμε τις συστάσεις για τις πρώτες 10 εικόνες του συνόλου δεδομένων μας. Στον παρακάτω πίνακα φαίνονται τα μέτρα ομοιότητας για κάθε πρόβλεψη, τα οποία υπολογίστηκαν με τον τρόπο που έχουμε ήδη περιγράψει.

Εικόνα	Σύσταση 1	Σύσταση 2	Σύσταση 3	Σύσταση 4	Σύσταση 5	Σύσταση 6
1	0.99933064	0.99872935	0.99842364	0.99841326	0.9983516	0.99834394
2	0.93330675	0.9323562	0.92838764	0.9278087	0.92078376	0.9176374
3	0.9962066	0.99579084	0.99505687	0.9947887	0.99415475	0.99411875
4	0.9991956	0.9991562	0.99906045	0.99900335	0.9989956	0.9988768
5	0.9999723	0.9999551	0.99994916	0.9999431	0.9999421	0.9999421
6	0.99800944	0.99745154	0.99692976	0.9962692	0.99303806	0.9917012
7	0.99977714	0.9996979	0.9996975	0.99967945	0.99966115	0.9996534
8	0.9713972	0.9707801	0.9598353	0.9596516	0.9595351	0.9576949
9	0.99997973	0.9999515	0.9999489	0.9999421	0.9999269	0.9999188
10	0.9983738	0.99794495	0.99791604	0.9975929	0.99727565	0.99576205

Table 15: Συστάσεις για απλό CNN μοντέλο

Παρακάτω παραθέτουμε και τις συστάσεις των προϊόντων γραφικά. Τα αποτελέσματα που παίρνουμε σε σχέση με τα προηγούμενα μοντέλα είναι καλύτερα από αυτά του InceptionV3 αλλά δεν υπερτερούν σαυτά του ResNet50.



Figure 52:
Συστάσεις των προϊόντων 1,2,3



Figure 53:
Συστάσεις των προϊόντων 4,5,6

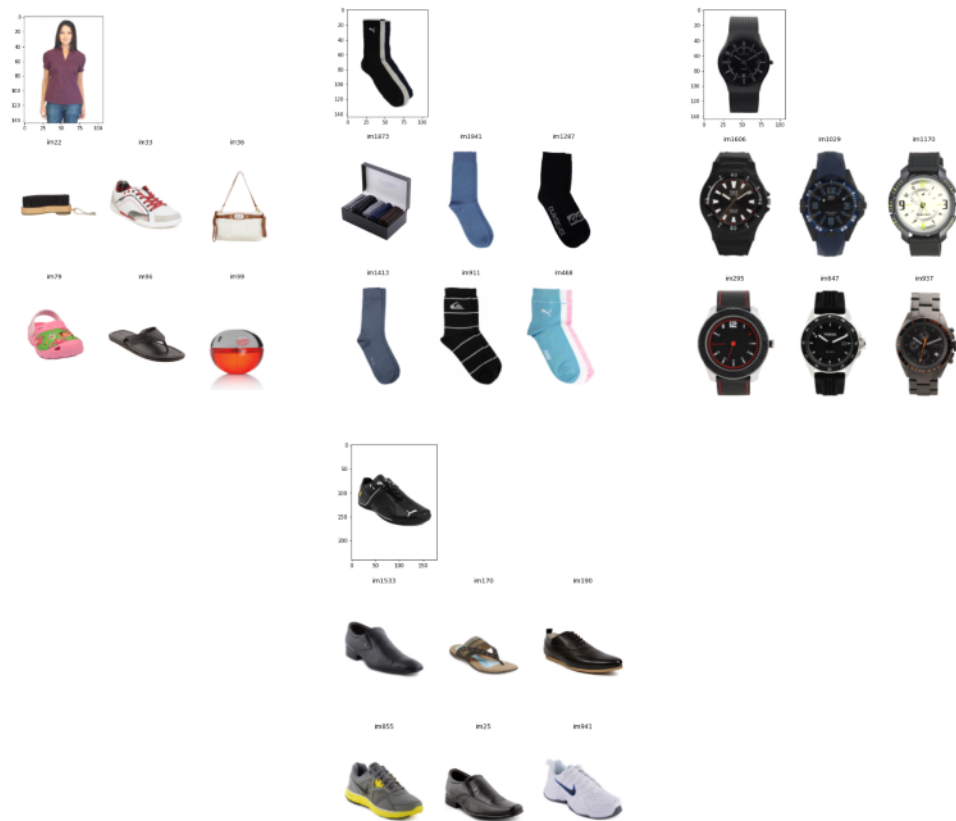


Figure 54:
Συστάσεις των προϊόντων 7,8,9,10

4.2 Συμπεράσματα

Παρατηρούμε ότι και στις περιπτώσεις, στην δυαδική κατηγοριοποίηση εικόνας και στο σύστημα συστάσεων τα προεκπαιδευμένα μοντέλα ως επί το πλείστον απέδωσαν καλύτερα από τα μη προεκπαιδευμένα συνελικτικά δίκτυα. Εκτός του ότι κάποια από τα προεκπαιδευμένα μοντέλα είχαν καλύτερη ακρίβεια (InceptionV3 στη δυαδική κατηγοριοποίηση και ResNet50 στο σύστημα συστάσεων) είχαμε τελικά και καλύτερα αποτελέσματα στους στόχους των εργασιών.

Στο πρώτο πείραμα το προεκπαιδευμένο μοντέλο InceptionV3 κατάφερε τις καλύτερες επιδόσεις, με πολύ καλή πρόβλεψη και κατηγοριοποίηση στο σύνολο των φωτογραφιών για αξιολόγηση που δώσαμε.

Επίσης, με χρήση του ResNet50 καταφέραμε να πάρουμε καλύτερες συστάσεις σε σχέση με το μη προεκπαιδευμένο συνελικτικό δίκτυο και το InceptionV3 που ενώ είχαμε δώσει σαν εισόδο γυναικεία μπλούζα μας πρότεινε μόνο αντρικές.

Θα μπορούσαμε να πετύχουμε ακόμη καλύτερα αποτελέσματα αν προχωρούσαμε σε συνδυασμό των προεκπαιδευμένων μοντέλων, αν εκπαιδεύαμε τα μοντέλα μας ή αν πειραματιζόμασταν με το learning rate για περισσότερες epochs εφόσον σε όλες τις υλοποιήσεις η εκπαίδευση κράτησε 10 epochs για λόγους οικονομίας πόρων και χρόνου.

Σαν συμπέρασμα μπορούμε να πούμε ότι και στα δύο πειράματα τα προεκπαιδευμένα μοντέλα απέδωσαν καλύτερα από τα απλά συνελικτικά νευρωνικά δίκτυα. Όσον αφορά τα προεκπαιδευμένα μοντέλα, αυτό που ήταν πιο βαθύ, δηλαδή το ResNet50 (50 στρώματα) ωφελήθηκε εκεί που το σύνολο δεδομένων είχε περισσότερες κλάσεις, αφού ο αριθμός των εικόνων για εκπαίδευση ήταν περίπου ίδιος. Αυτό είναι κάτι που έχει παρατηρηθεί στα νευρωνικά δίκτυα, δηλαδή ο μεγαλύτερος αριθμός στρωμάτων δε σημαίνει και απαραίτητα ότι θα έχουμε ωφέλιμα αποτελέσματα στην ακρίβεια και στην απώλεια. Το συγκεκριμένο επιβεβαιώνεται και στο 1ο Πείραμα που το ResNet50 δεν απέδωσε το ίδιο καλά καθώς μόνο δύο κλάσεις. Για το InceptionV3 και το VGG19 μπορούμε να πούμε ότι το VGG19 είναι σίγουρα μια καλή αρχιτεκτονική νευρωνικών δικτύων, αλλά μπορεί να μην έχει καλή απόδοση για τις δύσκολες εργασίες, καθώς είναι μια απλή στοίβα συνελικτικών και max-pooling επιπέδων που ακολουθούνται το ένα από το άλλο και τελικά πλήρως συνδεδεμένα επίπεδα. Με πιο απλά λόγια, δεν είναι σε θέση να εξαγάγει πολύ σύνθετα χαρακτηριστικά. Από την άλλη πλευρά, το InceptionV3 έχει τα inception modules που αποτελούνται από φίλτρα 1X1 και ακολουθούνται από συνελικτικές στρώσεις με διαφορετικά μεγέθη φίλτρων που εφαρμόζονται ταυτόχρονα. Αυτό επιτρέπει να μαθαίνουν πιο σύνθετα χαρακτηριστικά. Έχουν περισσότερα κρυφά επίπεδα σε σύγκριση με το VGG19. Ως εκ τούτου, τα καταφέρνουν καλύτερα σε σύνθετες εργασίες, όπως και έγινε.

Μέσα από τις παραπάνω εργασίες θέλαμε να διαπιστώσουμε κατά πόσο η μεταφορά γνώσης βοηθάει και βελτιστοποιεί εργασίες όπως η υπολογιστική όραση και η κατηγοριοποίηση εικόνας και τα αποτελέσματά μας φαίνεται να ενισχύουν την αρχική μας προσδοκία ως προς την ανωτερότητα των προεκπαιδευμένων μοντέλων. Θα μπορούσαμε στο μέλλον να εξετάσουμε το συνδυασμό των προεκπαιδευμένων μοντέλων ώστε να καταλήξουμε σε ακόμη καλύτερα αποτελέσματα και στις δύο εργασίες. Ακόμη, είναι δυνατό να υλοποιήσουμε κάποια

εφαρμογή που θα βελτιστοποιεί το μηχανισμό του συστήματος συστάσεων.

Όσον αφορά το κομμάτι της μεταφοράς γνώσης, μελλοντικά μπορούμε να αξιοποιήσουμε τη γνώση από ένα σύνολο εικόνων όπως αυτά που επεξεργαστήκαμε στην παρούσα εργασία και σε άλλες εφαρμογές συστημάτων συστάσεων όπως για παράδειγμα να εκπαιδεύσουμε ένα μοντέλο για σύσταση ταινιών και να το αξιοποιήσουμε σε σύνολα που αφορούν βιβλία ώστε να υλοποιήσουμε ένα σύστημα συστάσεων που να προτείνει βιβλία.

Λίστα Εικόνων και Διαγραμμάτων

- Εικόνα 1, σελίδα 7 - Αντιστοίχιση Βιολογικού με Τεχνητού Νευρώνα
- Εικόνα 2, σελίδα 8 - Μηχανισμός Τεχνητού Νευρώνα
- Εικόνα 3, σελίδα 9 - Νευρωνικό Δίκτυο
- Εικόνα 4, σελίδα 10 - Multi-Layer Perceptron
- Εικόνα 5, σελίδα 11 - Σύγκριση ενός κλασικού Νευρωνικού Δικτύου με ένα CNN
- Εικόνα 6, σελίδα 11 - Επίπεδα Συνελικτικού Νευρωνικού Δικτύου
- Εικόνα 7, σελίδα 12 - Κατευθυνόμενος Γράφος
- Εικόνα 8, σελίδα 14 - Αρχιτεκτονικές στα Νευρωνικά Δίκτυα Γράφων
- Εικόνα 9, σελίδα 14 - Απεικόνιση 1
- Εικόνα 10, σελίδα 15 - Απεικόνιση 2
- Εικόνα 11, σελίδα 19 - Αρχιτεκτονική GraphSage
- Εικόνα 12, σελίδα 20 - Αρχιτεκτονική PinSage
- Εικόνα 13, σελίδα 21 - Σύγκριση Βοηθητικής Εκμάθησης
- Εικόνα 14, σελίδα 22 - Εκμάθηση Μεμονωμένης Εργασίας-Πολλαπλών Εργασιών
- Εικόνα 15, σελίδα 24 - SELAR Framework
- Εικόνα 16, σελίδα 28 - Προεκπαίδευση σε Επίπεδο Γράφου
- Εικόνα 17, σελίδα 32 - Αρχιτεκτονική του VGG19
- Εικόνα 18, σελίδα 32 - Inception Module
- Εικόνα 19, σελίδα 33 - Αρχιτεκτονική του InceptionV3
- Εικόνα 20, σελίδα 34 - Αρχιτεκτονική του ResNet50
- Εικόνα 21, σελίδα 34 - Residual Block
- Εικόνα 22, σελίδα 39 - Αρχιτεκτονική CNN Μοντέλου
- Εικόνα 23, σελίδα 40 - Διάγραμμα Ακρίβειας CNN μοντέλου
- Εικόνα 24, σελίδα 40 - Διάγραμμα Απώλειας CNN μοντέλου
- Εικόνα 25, σελίδα 41 - Εικόνα Κατηγοριοποίησης 1
- Εικόνα 26, σελίδα 41 - Confusion Matrix για απλό μη προεκπαιδευμένο
- Εικόνα 27, σελίδα 44 - Διάγραμμα Ακρίβειας VGG19 μοντέλου
- Εικόνα 28, σελίδα 44 - Διάγραμμα Απώλειας VGG19 μοντέλου
- Εικόνα 29, σελίδα 44 - Εικόνα Κατηγοριοποίησης 3
- Εικόνα 30, σελίδα 44 - Confusion Matrix για VGG19
- Εικόνα 31, σελίδα 46 - Αρχιτεκτονική InceptionV1
- Εικόνα 32, σελίδα 47 - Αρχιτεκτονική InceptionV3
- Εικόνα 33, σελίδα 47 - Διάγραμμα Ακρίβειας InceptionV3 μοντέλου
- Εικόνα 34, σελίδα 47 - Διάγραμμα Απώλειας InceptionV3 μοντέλου
- Εικόνα 35, σελίδα 48 - Confusion Matrix για InceptionV3
- Εικόνα 36, σελίδα 50 - Αρχιτεκτονική ResNet50 μοντέλου
- Εικόνα 37, σελίδα 51 - Διάγραμμα Ακρίβειας ResNet50 μοντέλου
- Εικόνα 38, σελίδα 51 - Διάγραμμα Απώλειας ResNet50 μοντέλου
- Εικόνα 39, σελίδα 51 - Confusion Matrix για ResNet50
- Εικόνα 40, σελίδα 52 - Σύγκριση Μοντέλων
- Εικόνα 41, σελίδα 56 - Διάγραμμα Ακρίβειας InceptionV3 μοντέλου
- Εικόνα 42, σελίδα 56 - Διάγραμμα Απώλειας InceptionV3 μοντέλου
- Εικόνα 43, σελίδα 56 - Σύσταση 1,2,3 με χρήση InceptionV3 μοντέλου
- Εικόνα 44, σελίδα 57 - Σύσταση 4,5,6 με χρήση InceptionV3 μοντέλου

- Εικόνα 45, σελίδα 57 - Σύσταση 7,8,9,10 με χρήση InceptionV3 μοντέλου
- Εικόνα 46, σελίδα 58 - Διάγραμμα Ακρίβειας ResNet50 μοντέλου
- Εικόνα 47, σελίδα 58 - Διάγραμμα Απώλειας ResNet50 μοντέλου
- Εικόνα 48, σελίδα 59 - Σύσταση 1,2,3 με χρήση ResNet50 μοντέλου
- Εικόνα 49, σελίδα 60 - Σύσταση 4,5,6 με χρήση ResNet50 μοντέλου
- Εικόνα 50, σελίδα 60 - Σύσταση 7,8,9,10 με χρήση ResNet50 μοντέλου
- Εικόνα 51, σελίδα 61 - Διάγραμμα Ακρίβειας μη προεκπαιδευμένου μοντέλου
- Εικόνα 52, σελίδα 61 - Διάγραμμα Απώλειας μη προεκπαιδευμένου μοντέλου
- Εικόνα 53, σελίδα 62 - Σύσταση 1,2,3 με χρήση μη προεκπαιδευμένου μοντέλου
- Εικόνα 54, σελίδα 63 - Σύσταση 4,5,6 με χρήση μη προεκπαιδευμένου μοντέλου
- Εικόνα 55, σελίδα 63 - Σύσταση 7,8,9,10 με χρήση μη προεκπαιδευμένου μοντέλου

Λίστα Πινάκων

- Πίνακας 1, σελίδα 33 - Σύγκριση Προεκπαιδευμένων Μοντέλων
- Πίνακας 2, σελίδα 40 - Αποτελέσματα για απλό CNN μοντέλο
- Πίνακας 3, σελίδα 42 - Classification Report για απλό CNN μοντέλο
- Πίνακας 4, σελίδα 42 - Αποτελέσματα για μοντέλο VGG19
- Πίνακας 5, σελίδα 44 - Classification Report για μοντέλο VGG19
- Πίνακας 6, σελίδα 46 - Αποτελέσματα για μοντέλο InceptionV3
- Πίνακας 7, σελίδα 47 - Classification Report για μοντέλο InceptionV3
- Πίνακας 8, σελίδα 49 - Αποτελέσματα για μοντέλο ResNet50
- Πίνακας 9, σελίδα 50 - Classification Report για μοντέλο ResNet50
- Πίνακας 10, σελίδα 53 - Αποτελέσματα για μοντέλο InceptionV3
- Πίνακας 11, σελίδα 54 - Συστάσεις για μοντέλο InceptionV3
- Πίνακας 12, σελίδα 57 - Αποτελέσματα για μοντέλο ResNet50
- Πίνακας 13, σελίδα 58 - Συστάσεις για μοντέλο ResNet50
- Πίνακας 14, σελίδα 60 - Αποτελέσματα για απλό CNN μοντέλο
- Πίνακας 15, σελίδα 61 - Συστάσεις για απλό CNN μοντέλο

Συμβολισμοί

h_G	ενσωμάτωση
GNN	Graph Neural Network
CNN	Convolutional Neural Network
ΣΝΔ	Συνελικτικά Νευρωνικά Δίκτυα
ΝΔΓ	Νευρωνικά Δίκτυα Γράφων
\mathcal{T}	Εργασία
\mathcal{D}	Τομέας
\mathcal{L}	Συνάρτηση Απώλειας
G	Γράφος
TGN	Temporal Graph Networks
GAN	Graph Attention Networks
MPNN	Message Passing Neural Networks
GCN	Graph Convolutional Networks
ANN	Artificial Neural Networks

Αναφορές

1. A Comprehensive Survey on Transfer Learning, Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Senior Member, Hui Xiong, Fellow, and Qing He, arXiv:1911.02685v3, June 2020
2. A Gentle Introduction to Neural Networks, published in Towards Data Science in August 2017
3. An Introduction To Neural Networks, Kevin Gurney, University of Sheffield, published in 1997 by UCL Press
4. Παπαδόπουλος Αθανάσιος, "Συνελικτικά Νευρωνικά Δίκτυα στην Υπολογιστική Όραση", Απρίλιος 2016, Πανεπιστήμιο Πατρών
5. Graph neural networks: A review of methods and applications, Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, Maosong Sun, 2020, Chinese Roots Global Impact
6. A Gentle Introduction to Graph Neural Networks, published in Towards Data Science August 2019
7. Best Graph Neural Network architectures: GCN, GAT, MPNN and more, Sergios Karagiannakos, published in AI Summer in September 2021
8. Bruna, Joan, et al. "Spectral Networks and Locally Connected Networks on Graphs." ArXiv:1312.6203 [Cs], May 2014
9. Defferrard, Michaël, et al. "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering." ArXiv:1606.09375 [Cs, Stat], Feb. 2017
10. Kipf, Thomas N., and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks." ArXiv:1609.02907 [Cs, Stat], Feb. 2017
11. Gilmer, Justin, et al. "Neural Message Passing for Quantum Chemistry." ArXiv:1704.01212 [Cs], June 2017
12. Veličković, Petar, et al. "Graph Attention Networks." ArXiv:1710.10903 [Cs, Stat], Feb. 2018
13. Hamilton, William L., et al. "Inductive Representation Learning on Large Graphs." ArXiv:1706.02216 [Cs, Stat], Sept. 2018
14. Ying, Rex, et al. "Graph Convolutional Neural Networks for Web-Scale Recommender Systems." ArXiv:1806.01973 [Cs, Stat], June 2018
15. Rossi, Emanuele, et al. "Temporal Graph Networks for Deep Learning on Dynamic Graphs." ArXiv:2006.10637 [Cs, Stat], Oct. 2020
16. Xueting Han, Zhenhuan Huang, Bang An, Jing Bai et al. "Adaptive Transfer Learning on Graph Neural Networks" ArXiv:2107.08765v2 [cs.LG] Jul. 2021
17. Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, Jure Leskovec et al. "Strategies for Pre-Training Graph Neural Networks" ArXiv:1905.12265v3 [cs.LG] Feb. 2020
18. Andrey Gritsenko, Yuan Guo, Kimia Shayestehfard, Armin Moharrer, Jennifer Dy, Stratis Ioannidis et al. "Graph Transfer Learning"
19. Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen, "An end-to-end deep learning architecture for graph classification" In AAAI Conference on Artificial Intelligence, pp. 4438–4445, 2018.
20. Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec, "Graph convolutional neural networks for web-scale recom-

- mender systems” In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 974–983, 2018a.
21. Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec, ”Hierarchical graph representation learning with differentiable pooling”, In Advances in Neural Information Processing Systems (NeurIPS), 2018b.
 22. Vivien Tran-Thien, ”Censoring the auxiliary loss gradient to mitigate risks of negative transfer”
 23. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A. (2008). ”Extracting and composing robust features with denoising autoencoders”. Proceedings of the 25th International Conference on Machine Learning, 1096–1103.
 24. Zhang, Z., Luo, P., Loy, C. C., Tang, X. (2014). ”Facial landmark detection by deep multi-task learning”, In ECCV. 94–108.
 25. Mordan, T., Thome, N., Henaff, G., Cord, M. (2018), ”Revisiting Multi-Task Learning with ROCK: a Deep Residual Auxiliary Block for Visual Detection”, Advances in Neural Information Processing Systems, 1317–1329.
 26. Trinh, T. H., Dai, A. M., Luong, T., Le, Q. V. (2018), ”Learning longer-term dependencies in rnns with auxiliary losses”, ArXiv Preprint ArXiv:1803.00144.
 27. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013), ”Distributed representations of words and phrases and their compositionality”, Advances in Neural Information Processing Systems, 3111–3119.
 28. Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., Kavukcuoglu, K. (2016), ”Reinforcement learning with unsupervised auxiliary tasks”, ArXiv Preprint ArXiv:1611.05397.
 29. Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., Efros, A. A. (2018), ”Large-scale study of curiosity-driven learning”, ArXiv Preprint ArXiv:1808.04355.
 30. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. ”BERT: pre-training of deep bidirectional transformers for language understanding”, In NAACL-HLT, pages 4171–4186, 2019.
 31. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, ”Deep residual learning for image recognition”, In CVPR, pages 770–778, 2016.
 32. Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler, ”Aligning books and movies: Towards story-like visual explanations by watching movies and reading books”, In ICCV, pages 19–27, 2015.
 33. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, ”Imagenet: A large-scale hierarchical image database”, In CVPR, pages 248–255. Ieee, 2009.
 34. Dasol Hwang, Jinyoung Park, Sunyoung Kwon, Kyung-Min Kim, Jung-Woo Ha, Hyunwoo J. Kim, ”Self-supervised Auxiliary Learning with Meta-paths for Heterogeneous Graphs”, arXiv:2007.08294v5 [cs.LG], Feb 2021
 35. Masoud Mahdianpari, Bahram Salehi, Mohammad Rezaee, Fariba Mohammadianesh, Yun Zhang, ”Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery”, July 2018
 36. Xueting Han, Zhenhuan Huang, Bang An, Jing Bai, ”Adaptive Transfer Learning on Graph Neural Networks”, ArXiv:2107.08765

37. Kou, F., Du, J., He, Y., Ye, L. (2016) "Social Network Search Based on Semantic Analysis and Learning." *CAAI Transactions on Intelligence Technology*.
38. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Garcia-Rodriguez, J. (2017) "A Review on Deep Learning Techniques Applied to Semantic Segmentation."
39. Li, L. J., Su, H., Lim, Y., Li, F. F. (2010, September) "Objects as Attributes for Scene Classification." *ECCV Workshops(57-69)*. Krizhevsky, A., Hinton, G. (2009) "Learning multiple layers of features from tiny images."
40. LeCun, Y., Jackel, L. D., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., ... Vapnik, V. (1995) "Learning algorithms for classification: A comparison on hand-written digit recognition." *Neural networks: the statistical mechanics perspective(pp 261-276)*.
41. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998) "Gradient-based learning applied to document recognition." *proceedings of the IEEE 86(11): 2278-2324*.
42. Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014) "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research 15(1): 1929-1958*.
43. Eitz, M., Hays, J., Alexa, M. (2012) "How do humans sketch objects?" *ACM Trans. Graph., 31(4)*.
44. Ballester, P., deAraújo, R. M. (2016, February) "On the Performance of GoogLeNet and AlexNet Applied to Sketches." in *AAAI*.
45. Yang, Y., Hospedales, T. M. (2015) "Deep neural networks for sketch recognition".
46. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L. (2014) "Large-scale video classification with convolutional neural networks." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
47. Ciresan, Dan, Ueli Meier, and JurgenSchmidhuber, (2012) "Multi-column deep neural networks for image classification." *2012 IEEE Conference on Computer Vision and Pattern Recognition*.
48. Ciresan, Dan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, and JurgenSchmidhuber. (2011) "Flexible, High Performance Convolutional Neural Networks for Image Classification." *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Two: 1237-1242*.
49. Lawrence, Steve, C. Lee Giles, Ah Chung Tsoi, and Andrew D. Back. (1997) "Face Recognition: A Convolutional Neural Network Approach." *IEEE Transactions on Neural Networks, Volume 8; Issue 1*.
50. James Le, The 4 Convolutional Neural Network Models That Can Classify Your Fashion Image, published in *Towards Data Science* in October 2018
51. Image recommendation engine — leverage transfer learning, Alexandre Wrg published in *Towards Data Science* in October 2019
52. Neha Sharma, Vibhor Jain, Anju Mishra, "An Analysis of Convolutional Neural Networks For Image Classification", *International Conference on Computational Intelligence and Data Science 2018*
53. Definition of Cosine Similarity on *Wikipedia*
54. Understanding the VGG19 Architecture as published in *opengenius.org*
55. Understanding the ResNet50 Architecture as published in *opengenius.org*

56. Understanding the InceptionV3 Architecture as published in openaccess.thecvf.com
57. Performance Analysis of Open Source Machine Learning Frameworks for Various Parameters in Single-Threaded and Multi-Threaded Modes, Yuriy P, Kochura, Sergii Stirenko, Oleg Alienin, Michail Novotarskiy, August 2017
58. How to choose CNN Architecture MNIST, published on Kaggle
59. Mridula Verma, D. R. Sahu, K. K. Shukla, VAGA: a novel viscosity-based accelerated gradient algorithm Convergence analysis and applications Springer Science+Business Media, LLC, part of Springer Nature 2017
60. Yufeng Zheng, Clifford K. Yang, Breast cancer screening using convolutional neural network and follow-up digital mammography, Computational Imaging III, May 2018
61. Masoud Mahdianpari, Bahram Salehi, Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery, Article in Remote Sensing · July 2018
62. Ammar Mahmood, Ana Giraldo, Mohammed Bennamoun, Senjian An, Automatic Hierarchical Classification of Kelps Using Deep Residual Features
63. Going deeper with convolutions, Christian Szegedy Google Inc., Wei Liu University of North Carolina, Chapel Hill Yangqing Jia Google Inc. Pierre Sermanet Google Inc. Scott Reed University of Michigan Dragomir Anguelov Google Inc. Dumitru Erhan Google Inc. Vincent Vanhoucke Google Inc. Andrew Rabinovich Google Inc. , arXiv:1409.4842v1, Sept 2014