Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Συστηματων Μεταδοσης Πληροφοριας και Τεχνολογιας Υλικων

Ευφυές σύστημα παραγωγής λεκτικής περιγραφής εικόνας

Μελέτη και υλοποίηση

Διπλωματικη Εργασια

του

ΚΑΤΑΚΗ ΝΙΚΗΦΟΡΟΥ ΕΜΜΑΝΟΥΗΛ

Επιβλέπων: Ι. Στ. Βενιέρης
Καθηγητής

Αθήνα, Νοέμβριος 2020

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομεας Συστηματων Μεταδοσης Πληροφοριας και Τεχνολογιας Υλικων

Ευφυές σύστημα παραγωγής λεκτικής περιγραφής εικόνας

Μελέτη και υλοποίηση

Διπλωματικη Εργασια

του

ΚΑΤΑΚΗ ΝΙΚΗΦΟΡΟΥ ΕΜΜΑΝΟΥΗΛ

Επιβλέπων: Ι. Στ. Βενιέρης
Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5 Νοεμβρίου 2020.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Ι. Στ. Βενιέρης	Δ. - Θ. Κακλαμάνη	Γ. Ματσόπουλος
Καθηγητής	Καθηγήτρια	Καθηγητής

Αθήνα, Νοέμβριος 2020

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομεας Συστηματων Μεταδοσης Πληροφοριας και Τεχνολογιας Υλικων

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

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

(Υπογραφή)

.....
Κατάκης Νικηφόρος
Εμμανουήλ

5 Νοεμβρίου 2020

Περίληψη

Το θέμα της παρούσας διπλωματικής εργασίας είναι η δημιουργία ενός ευφυούς συστήματος παραγωγής λεκτικής περιγραφής εικόνας (Image Captioning), εφαρμόζοντας τεχνικές που εμπίπτουν στην περιοχή της Μηχανικής Μάθησης (Machine Learning), και συγκεκριμένα Βαθιάς Μηχανικής Μάθησης (Deep Learning). Τα τελευταία χρόνια, με την τεράστια ανάπτυξη της Τεχνητής Νοημοσύνης (Artificial Intelligence), το συγκεκριμένο πρόβλημα έχει τραβήξει την προσοχή πολλών ερευνητών, χάρη στην εφαρμογή που βρίσκει σε ένα ευρύ φάσμα τομέων, και έχει γίνει ένα ενδιαφέρον και επίπονο έργο.

Καθημερινά προκύπτει ένας τεράστιος όγκος ψηφιακών δεδομένων, κάτι που κρίνει αναγκαία την βαθύτερη κατανόηση της δομής τους και την ανακάλυψη τρόπων επεξεργασίας και εξαγωγής χρήσιμης πληροφορίας από αυτά. Η παραγωγή λεκτικής περιγραφής μίας εικόνας μπορεί να φανεί πολύ χρήσιμη σε διάφορους κλάδους, όπως η παραγωγή εφαρμογών για την βοήθεια ανθρώπων με προβλήματα όρασης, βελτίωση διαφόρων στοιχείων των πλατφορμών κοινωνικής δικτύωσης, περιγραφή ενός βίντεο frame by frame και βελτίωση των μηχανών αναζήτησης που ασχολούνται με εικόνες.

Προκειμένου να προσεγγίσουμε το θέμα του Image Captioning, αρχικά θα γίνει μία ανάπτυξη όλων των απαραίτητων θεωρητικών γνώσεων στους τομείς της μηχανικής μάθησης και των βαθιών νευρωνικών δικτύων. Έπειτα, θα παρουσιάσουμε ένα σύνολο μεθόδων και διαφορετικών αρχιτεκτονικών που έχουν χρησιμοποιηθεί για την επίλυση του προβλήματος, μελετώντας τις επιδόσεις που έχουν σημειώσει και τελικά θα περιγράψουμε την δική μας αρχιτεκτονική.

Στην συνέχεια θα υλοποιήσουμε το δικό μας μοντέλο παραγωγής λεκτικής περιγραφής εικόνας, το οποίο θα αποτελείται από έναν κωδικοποιητή, βασισμένο στα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNN) και έναν αποκωδικοποιητή, βασισμένο στα Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks - RNN), χρησιμοποιώντας ακόμα τον μηχανισμό της Προσοχής Attention, για την παραγωγή όσο το δυνατόν καλύτερων αποτελεσμάτων.

Τέλος, θα κατασκευάσουμε ένα web app που θα χρησιμοποιεί το παραπάνω μοντέλο για την παραγωγή λεκτικής περιγραφής εικόνων, αλλά και δύο ακόμα pretrained μοντέλα για να δώσουμε στο application ακόμα περισσότερες δυνατότητες, όπως αυτή της αναζήτησης με βάση την παραγόμενη λεζάντα και την κατηγοριοποίηση των εικόνων σε μία γκαλερί με διαφορετικές κατηγορίες βασιζόμενοι στις λεκτικές περιγραφές των εικόνων.

Λέξεις Κλειδιά

Λεκτική περιγραφή εικόνας, βαθιά μηχανική μάθηση, συνελικτικά νευρωνικά δίκτυα, α-

ναδρομικά νευρωνικά δίκτυα, όραση υπολογιστών, επεξεργασία φυσικής γλώσσας, μηχανή αναζήτησης, διαδικτυακή εφαρμογή, μηχανισμός προσοχής

Abstract

This diploma thesis deals with the problem of Image Captioning utilizing Deep Learning techniques. In recent years, with the rapid development of artificial intelligence, image caption has gradually attracted the attention of many researchers in the field of artificial intelligence and has become an interesting and arduous task.

The huge amount of image data that is generated on an everyday basis has made the need, to further understand and extract useful information from them, essential. Automated Image Captioning can be extremely useful to many applications, like those that are created for people that are visually impaired. In addition, it can improve many Social Media Platforms, explain the contents of a video frame by frame and improve Image Search Engines.

In order to grasp the problem of Image Captioning, we will first understand the theory of machine learning and deep neural networks.Next, we will discuss the various methods and models that have been used to solve this problem, examining their performances and making assumptions, and then propose our own architecture.

Moving forward, we will implement our own Image Captioning Model, which will consist of an encoder, based on Convolutional Neural Networks (CNN) and a decoder, based on Recurrent Neural Networks (RNN), while also utilizing the Attention Mechanism. In the end, we will create a web application, that will utilize the aforementioned model and two other pretrained models to produce captions, build a search engine and a gallery that will all be set up based on the produced captions.

Keywords

Image Captioning, Deep Learning, Convolutional Neural Networks, Reccurent Neural Networks, Computer vision, Natural Language Processing, search engine, Attention Mechanism, Web application

στους γονείς μου

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. Ιάκωβο Βενιέρη για την επίβλεψη αυτής της διπλωματικής εργασίας, για την ευκαιρία που μου έδωσε να εμβαθύνω στο κομμάτι της μηχανικής μάθησης και για τις γνώσεις που μου προσέφερε. Επίσης ευχαριστώ ιδιαίτερα την κ. Σοφία Καπελάκη για την εξαιρετική συνεργασία που είχαμε και για την καθοδήγησή της, όπως επίσης και το Εθνικό Δίκτυο Υποδομών Τεχνολογίας και Έρευνας(ΕΔΥΤΕ) για την προσφορά των απαραίτητων υπολογιστικών πόρων, χωρίς των οποίων δεν θα ήταν δυνατά τα αποτελέσματα μας. Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια, αλλά και τους φίλους μου και την Ε. που ήταν πάντα στο πλευρό μου όταν τους είχα ανάγκη.

Αθήνα, Νοέμβριος 2020

Κατάκης Νικηφόρος Εμμανουήλ

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Η σύγχρονη εποχή χαρακτηρίζεται από πολλούς ανθρώπους και ως η Εποχή της Πληροφορίας. Με την ραγδαία ανάπτυξη του διαδικτύου, ο καθημερινός όγκος πληροφορίας που είναι διαθέσιμος στους ανθρώπους είναι όλο και μεγαλύτερος. Ένας βασικός πυλώνας πληροφορίας είναι και οι εικόνες, με αποτέλεσμα να γίνεται αναγκαία η ύπαρξη συστημάτων για όσο το δυνατόν καλύτερη ερμηνεία και εκμετάλλευση τους.

Η ανάπτυξη μηχανών που έχουν την δυνατότητα να καταλαβαίνουν το νόημα των εικόνων, αλλά και την σημασία της φυσικής γλώσσας αποτελούν βασικά ζητούμενα της τεχνητής νοημοσύνης. Τα τελευταία χρόνια, η έρευνα τόσο στον τομέα της επεξεργασίας φυσικής γλώσσας (natural language processing - NLP), όσο και στον τομέα της όρασης υπολογιστών (computer vision - CV) έχει κάνει μεγάλες προόδους. Η εφαρμογή τεχνικών βαθιών νευρωνικών δικτύων έχει συμβάλει στην βελτίωση της επίδοσης εργασιών και στους δύο τομείς. Παρόλο που οι περιοχές της επεξεργασίας φυσικής γλώσσας και της όρασης υπολογιστών, συνήθως μελετώνται ξεχωριστά, παρατηρείται όλο και μεγαλύτερο ενδιαφέρον στους τρόπους με τους οποίους τα δύο πεδία μπορούν να συνδυαστούν. Αυτές οι τεχνικές ανήκουν στην κατηγορία της πολυτροπικής μάθησης (multimodal learning), που σημαίνει ότι η πληροφορία προέρχεται απο διαφορετικές κατηγορίες, στην περίπτωσή μας συγκεκριμένα, εικόνες και κείμενο.

Υπάρχει μεγάλο κίνητρο για τον άνθρωπο να γεφυρώσει την φυσική γλώσσα με την κατανόηση των εικόνων με σκοπό να δημιουργηθούν συστήματα που θα επιτρέπουν την αλληλεπίδραση ανθρώπου και μηχανής, χρησιμοποιώντας ως έναυσμα τον φυσικό κόσμο. Οι άνθρωποι χρησιμοποιούν την φυσική γλώσσα για να επικοινωνήσουν και θα είναι πολύ χρήσιμο ευφυείς μηχανές να μπορούν, να χρησιμοποιήσουν φυσική γλώσσα για να περιγράψουν οπτική πληροφορία.

Τα περισσότερα μοντέλα που χρησιμοποιούνται για τις βασικές NLP εργασίες μαθαίνουν τις σημασιολογικές ερμηνείες των λέξεων χρησιμοποιώντας μόνο κείμενα. Η μεθοδολογία αυτή μπορεί να παρέχει καλά αποτελέσματα για απλά προβλήματα, όμως δημιουργείται το εξής πρόβλημα: Τα μοντέλα προσπαθούν να κατανοήσουν το νόημα των λέξεων σε σχέση με τις υπόλοιπες λέξεις που χρησιμοποιούνται γύρω τους, όμως έτσι δεν υπάρχει δυνατότητα να συνδεθούν με τα αντικείμενα του πραγματικού κόσμου στα οποία αναφέρονται, παρά μόνο σε σύμβολα, δηλαδή άλλες λέξεις. Αυτό το πρόβλημα έχει τις ρίζες του αρκετές δεκαετίες πριν και ονομάζεται Symbol Grounding Problem [4]

Η σημασία της σύνδεσης της σημασίας των λέξεων με φυσικά αντικείμενα έχει οδηγήσει στην ανάπτυξη πολλών νέων τεχνικών στο πεδίο της επεξεργασίας φυσικής γλώσσας. Οι

μεγάλοι πρόοδοι που έχουν γίνει και στα βαθιά νευρωνικά δίκτυα δίνουν την δυνατότητα να αντιμετωπιστούν νέα προβλήματα, που βασίζονται στην σύνδεση της φυσικής γλώσσας με οπτικά ερεθίσματα. Ένα από αυτά τα προβλήματα είναι και το Image Captioning με το οποίο θα ασχοληθούμε στα πλαίσια της παρούσας διπλωματικής.

Ο σκοπός μας δεν είναι να σταματήσουμε, όμως, εκεί. Ένα από τα μεγαλύτερα πλεονεκτήματα της Τεχνητής Νοημοσύνης και της Μηχανικής Μάθησης είναι η λύση προβλημάτων που μέχρι τώρα φάνταζαν άλυτα. Για να εκμεταλλευτούμε πραγματικά αυτές τις δυνατότητες θα πρέπει να περάσουμε από την θεωρία στην πράξη. Πολλές φορές η έρευνα τέτοιον προβλημάτων οδηγεί σε δημοσιεύσεις και νούμερα, αλλά όχι σε ένα αποτέλεσμα που θα μπορούσε να το χρησιμοποιήσει ο καθένας για τα δικά του προβλήματα.

Στην διπλωματική, λοιπόν, αυτή θα αναπτύξουμε και μία διαδικτυακή εφαρμογή που θα επιτυγχάνει ακριβώς αυτό. Την μεταφορά μοντέλων μηχανικής μάθησης από την θεωρία στην πράξη, δίνοντας την δυνατότητα στον χρήστη να εκμεταλλευτεί τόσο το μοντέλο του Image Captioning, το οποίο θα αναπτύξουμε αλλά και μοντέλα για διαφορετικές εργασίες. Με αυτό τον τρόπο θα γίνει πραγματικά κατανοητή η αξία της Μηχανικής Μάθησης στην πραγματική, καθημερινή ζωή.

1.1 Οργάνωση του τόμου

Η εργασία αυτή είναι οργανωμένη σε επτά κεφάλαια: Στο κεφάλαιο 2 δίνεται το απαραίτητο θεωρητικό υπόβαθρο των βασικών τεχνολογιών που σχετίζονται με την διπλωματική αυτή, όσον αφορά στο κομμάτι της μηχανικής μάθησης. Αρχικά, περιγράφεται η έννοια της Μηχανικής Μάθησης και οι διάφορες μέθοδοι της. Στην συνέχεια ασχολούμαστε συγκεκριμένα με την Επιβλεπόμενη Μάθηση και τα Βαθιά Νευρωνικά Δίκτυα που αποτελούν τα θεμέλια της υπόλοιπης εργασίας. Στο κεφάλαιο 3 γίνεται μία εις βάθος εισαγωγή στα συγκεκριμένα κομμάτια της βαθιάς μάθησης, μέσω των οποίων θα προσεγγίσουμε το θέμα. Στο κεφάλαιο 4 παρουσιάζουμε το σύνολο δεδομένων που θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου, καθώς και την βασική του αρχιτεκτονική. Στο κεφάλαιο 5, γίνεται αναλυτική περιγραφή της πειραματικής διαδικασίας, παρουσιάζονται συγγενικές εργασίες με το πρόβλημα μας και συγκρίνουμε τα αποτελέσματα μας με εκείνες. Επίσης γίνεται και ανάλυση των αποφάσεων που πήραμε για τις παραμέτρους του συστήματος και έχουμε και μία οπτικοποίηση κάποιων αποτελεσμάτων του μοντέλου μας. Στο κεφάλαιο 6 παρουσιάζεται η διαδικτυακή εφαρμογή, με περιγραφή όλων των δυνατοτήτων της και των παραπάνω μοντέλων που χρησιμοποιήθηκαν. Τέλος, στο κεφάλαιο 7 παραθέτονται τα τελικά συμπεράσματα, καθώς και οι διάφορες μελλοντικές επεκτάσεις της παρούσας διπλωματικής.

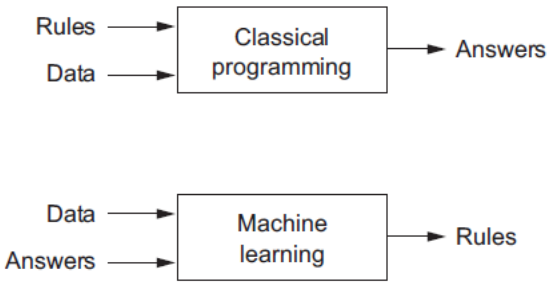
Κεφάλαιο 2

Θεωρητικό υπόβαθρο Μηχανικής Μάθησης

2.1 Μηχανική Μάθηση

2.1.1 Ορισμός Μεθόδων Μηχανικής Μάθησης

Η Μηχανική Μάθηση (Machine learning - ML) είναι ένα πεδίο της Τεχνητής Νοημοσύνης (Artificial Intelligence - AI) που στοχεύει στην δημιουργία συστημάτων με την ικανότητα να μαθαίνουν αυτόματα και να βελτιώνονται. Ένα τέτοιο σύστημα θα πρέπει να έχει την ικανότητα να βελτιώνει την απόδοσή του, όταν λαμβάνει νέα ερεθίσματα, χωρίς να χρειάζεται να προγραμματιστεί εξαρχής. Οι αλγόριθμοι της μηχανικής μάθησης επιτρέπουν στους υπολογιστές να εκπαιδεύονται πάνω σε δεδομένα εισόδου και χρησιμοποιούν στατιστική ανάλυση, ώστε να εξάγουν τιμές οι οποίες εμπίπτουν σε ένα συγκεκριμένο εύρος. Ο Tom M. Mitchell πρότεινε έναν πιο επίσημο ορισμό που χρησιμοποιείται ευρέως: «Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης Π, αν η επίδοσή του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E». [5] Η διαδικασία της μάθησης ξεκινάει με παραδείγματα ή οδηγίες, για να αναγνωριστούν πρότυπα στα δεδομένα και να ληφθούν καλύτερες αποφάσεις στο μέλλον, με βάση τα παραδείγματα που διαθέτουμε. Ο βασικός σκοπός είναι να επιτρέψουμε στους υπολογιστές να μαθαίνουν αυτόματα, χωρίς ανθρώπινη παρέμβαση ή βοήθεια, και να προσαρμόζουν τις πράξεις τους κατάλληλα. Αυτό το χαρακτηριστικό της μηχανικής μάθησης αποτελεί και την βασικότερη διαφορά με τον κλασικό προγραμματισμό, στον οποίο απλά τροφοδοτούμε το πρόγραμμα μας με δεδομένα και παίρνουμε τα αποτελέσματα. [6]



Εικόνα 2.1: *Διαφορές Μηχανικής Μάθησης και Κλασσικού Προγραμματισμού*

Οι αλγόριθμοι της μηχανικής μάθησης χωρίζονται σε 3 μεγάλες κατηγορίες. Οι κατηγορίες αυτές βασίζονται στον τρόπο με τον οποίο λαμβάνεται η μάθηση ή στον τρόπο με τον οποίο

δίνεται η ανάδραση στην εκμάθηση στο ανεπτυγμένο σύστημα, και είναι οι εξής:

- **Επιβλεπόμενη μάθηση (Supervised Learning)**: Το υπολογιστικό πρόγραμμα δέχεται τις παραδειγματικές εισόδους καθώς και τα επιθυμητά αποτελέσματα από έναν «δάσκαλο», και ο στόχος είναι να μάθει έναν γενικό κανόνα προκειμένου να αντιστοιχίσει τις εισόδους με τα αποτελέσματα. Οι αλγόριθμοι της Επιβλεπόμενης μάθησης μπορούν να χωριστούν σε δύο κατηγορίες με βάση την επιθυμητή έξοδο.
  - Classification, που αφορά τα προβλήματα, στα οποία τα προβλεπόμενα αποτελέσματα αποτελούν μέρος ενός διακριτού συνόλου, οι λεγόμενες κλάσεις. Σε περίπτωση που έχουμε δύο πιθανές κλάσεις, τότε το πρόβλημα ονομάζεται δυαδική κατηγοριοποίηση (binary classification).
  - Regression, που αφορά τα προβλήματα που η προβλεπόμενη έξοδος είναι μία πραγματική τιμή μέσα σε μία συγκεκριμένη εμβέλεια.

  Σε περισσότερες λεπτομέρειες για την Επιβλεπόμενη θα προχωρήσουμε στο επόμενο υποκεφάλαιο.
- **Μη Επιβλεπόμενη μάθηση (Unsupervised Learning)**: Όταν τα δεδομένα που διαχειριζόμαστε δεν έχουν αντιστοιχηθεί σε κάποια κλάση ή ετικέτα, δηλαδή δεν υπάρχει από πριν κάποια γνώση ή εμπειρία από αυτά [7]. Σκοπός είναι να εκπαιδεύσουμε αλγορίθμους που θα ανακαλύπτουν την δομή των δεδομένων εισόδου και θα βρίσκουν μοτίβα πάνω σε αυτά. Η πληροφορία παρέχεται στο δίκτυο χωρίς να συμβάλουμε με κάποιον έλεγχο και το ίδιο το δίκτυο προχωράει στην διόρθωση των δεδομένων μέσω του μηχανισμού της ανάδρασης (feedback). Τα δύο επικρατέστερα είδη της μη επιβλεπόμενης είναι τα παρακάτω:
  - Clustering, αναφέρεται στην ομαδοποίηση των δεδομένων με κατάλληλο τρόπο, ώστε τα δεδομένα που ανήκουν στην ίδια ομάδα (cluster) να έχουν παρόμοιες ιδιότητες ή χαρακτηριστικά μεταξύ τους και, συγχρόνως να έχουν διαφορές με τα δεδομένα των άλλων ομάδων.
  - Dimensionality reduction, αναφέρεται στην σύμπτυξη των δεδομένων μέσω της αφαίρεσης μεταβλητών χωρίς να χάνεται η σημασία και η δομή του συνόλου των δεδομένων. Με αυτό τον τρόπο επιτυγχάνεται μεγαλύτερη ευκολία στην αποθήκευση των δεδομένων, σε καλύτερη οπτική αναπαράστασή τους και σε ταχύτερη εκτέλεση υπολογισμών πάνω σε αυτά.
- **Ενισχυτική μάθηση (Reinforced Learning)**: Ένα πρόγραμμα υπολογιστή αλληλεπιδρά με ένα δυναμικό περιβάλλον στο οποίο πρέπει να επιτευχθεί ένας συγκεκριμένος στόχος, χωρίς κάποιος δάσκαλος να του λέει ρητά αν έχει φτάσει κοντά στο στόχο του [8]. Οι αλγόριθμοι ενισχυτικής μάθησης χρησιμοποιούν ένα σύστημα επιβράβευσης (reward system) και συνεχείς δοκιμές με σκοπό να μεγιστοποιηθεί η τελική επιβράβευση ενός πράκτορα (agent). Για παράδειγμα, σε ένα βιντεοπαιχνίδι ο πράκτορας είναι ο χαρακτήρας που χειρίζεται ο παίκτης. Ο αλγόριθμος της ενισχυτικής μάθησης θα πρέπει να αποφασίζει ποια είναι η καλύτερη διαδρομή που θα ακολουθήσει ο παίκτης, ώστε να πάρει την μέγιστη βαθμολογία.

2.2 Supervised Machine Learning

Το θέμα της παρούσας διπλωματικής, δηλαδή το πρόβλημα του Image Captioning ανήκει στην κατηγορία προβλημάτων επιβλεπόμενης μάθησης. Οι επόμενες ενότητες θα παρέχουν βασικές γνώσεις για τις κυριότερες μεθόδους επιβλεπόμενης μάθησης.

Όπως αναφέραμε και παραπάνω στο Supervised Learning, έχουμε δεδομένα εισόδου έστω X και την έξοδο, έστω Y. Σκοπός είναι να κατασκευαστεί ένας αλγόριθμος που θα μαθαίνει την συνάρτηση με την οποία θα αντιστοιχίζεται η είσοδος στην έξοδο.

$$Y = f(X) \tag{2.1}$$

Μέσω της μάθησης, αναζητείται όσο το δυνατόν καλύτερη συνάρτηση αντιστοίχησης, ώστε όταν νέα δεδομένα εισόδου X δίνονται στο μοντέλο, η έξοδος Y να μπορεί να προβλεφθεί σωστά.

2.2.1 Loss Function

Για να καταφέρει ο αλγόριθμος να μάθει την βέλτιστη συνάρτηση εισάγεται η έννοια της συνάρτησης κόστους (loss function). Η συνάρτηση αυτή, (συμβολίζεται ως $L(y', y)$) υπολογίζει την απώλεια, η οποία ορίζεται σαν ένας αριθμός, που υπήρξε όταν το μοντέλο προέβλεψε **y'** όταν η σωστή ετικέτα είναι **y**. Η συνάρτηση κόστους πρέπει να είναι κάτω φραγμένη και η ελάχιστη τιμή να επιτυγχάνεται όταν η πρόβλεψη του μοντέλου είναι σωστή. Οι παράμετροι της συνάρτησης που έχει μάθει το μοντέλο θα τεθούν στη συνέχεια με τέτοιο τρόπο ώστε να ελαχιστοποιείται η συνάρτηση κόστους L στα δεδομένα της εκπαίδευσης. Ο πιο συνηθισμένος τρόπος είναι να ελαχιστοποιούμε το άθροισμα των απωλειών όλων των διαφορετικών παραδειγμάτων της εκπαίδευσης.

Έστω ένα επισημασμένο σύνολο εκπαίδευσης $(x_1 : n, y_1 : n)$, μία συνάρτηση κόστους ανά δείγμα L και μία παραμετροποιημένη συνάρτηση $f(x; \Theta)$. Ώς συνολική απώλεια πάνω στο σύνολο δεδομένων σε σχέση με τις παραμέτρους Θ ορίζεται η μέση απώλεια πάνω σε όλα τα δεδομένα της εκπαίδευσης:

$$L(\Theta) = -\frac{1}{N}\sum_{i=1}^{N} L(f(x_i; \Theta), y_i) \tag{2.2}$$

Τα δεδομένα εισόδου έχουν σταθερές τιμές, και οι τιμές των παραμέτρων ορίζουν την απώλεια. Ο αλγόριθμος μάθησης έχει σκοπό να δώσει τέτοιες τιμές στις παραμέτρους Θ ώστε η τιμή του L να ελαχιστοποιηθεί.

$$\hat{\Theta} = arg_{\Theta} min L(\Theta) = arg_{\Theta} min \frac{1}{N}\sum_{i=1}^{N} L(f(x_i; \Theta), y_i) \tag{2.3}$$

Προχωρώντας ορίζεται η έννοια της *εντροπίας* (entropy) [9]. Ας υποθέσουμε ότι θέλουμε να επικοινωνήσουμε ένα σύνολο ν γεγονότων από μια συγκεκριμένη κατανομή πιθανότητας p. Η εντροπία πληροφορίας είναι το μέσο ελάχιστο μέγεθος κωδικοποίησης της πληροφορίας ώστε να επικοινωνήσουμε τα γεγονότα.

$$H(p) = \sum_x p(x) log \frac{1}{p(x)} \tag{2.4}$$

Αν η τιμή της εντροπίας είναι υψηλή (το μέγεθος κωδικοποίησης είναι κατά μέσο όρο μεγάλο), σημαίνει πως έχουμε πολλά δεδομένα εισόδου με μικρή πιθανότητα. Η εντροπία μπορεί να θεωρηθεί ένας τρόπος μέτρησης της αβεβαιότητας, εκτός από τρόπος μέτρησης της ποσότητας της πληροφορίας.

Η cross-entropy αποτελεί το μέσο ελάχιστο μέγεθος κωδικοποίησης της πληροφορίας του να επικοινωνήσουμε ένα γεγονός από μία κατανομή πιθανότητας σε μία άλλη. Ορίζεται ως εξής:

$$H_p(q) = \sum_x q(x) log \frac{1}{p(x)} \tag{2.5}$$

Στην περίπτωση που έχουμε δύο κατηγορίες, ορίζεται η δυαδική εντροπία (binary entropy). Πρόκειται για την εντροπία μιας διαδικασίας Bernoulli με πιθανότητα p που μπορεί να πάρει δύο τιμές. Έστω η τυχαία μεταβλητή X που μπορεί να πάρει δύο τιμές, 0 και 1. Τότε αν η πιθανότητα(X=1) = p, η πιθανότητα(X=0) = 1 - p και ορίζεται η εντροπία:

$$H(X) = plog\frac{1}{p} + (1-p)log\frac{1}{1-p} = -plogp - (1-p)log(1-p) \tag{2.6}$$

Η δυαδική συνάρτηση κόστους cross-entropy χρησιμοποιείται στην δυαδική ταξινόμηση με δεσμευμένες πιθανότητες εξόδου. Έστω σύνολο δεδομένων με δύο κλάσεις με ετικέτες 0 και 1, με τη σωστή ετικέτα $y \in (0, 1)$. Η έξοδος του ταξινομητή $\bar{y}$ μετασχηματίζεται με την χρήση της σιγμοειδούς (αλλιώς λογιστικής) συνάρτησης $\sigma(x) = \frac{1}{1+e^{-x}}$ στο διάστημα [0, 1], και σχηματίζει την δεσμευμένη πιθανότητα $\bar{y} = \sigma(\bar{y}) = P(y = 1|x)$. Προκύπτει ο κανόνας πρόβλεψης:

$$prediction = \begin{cases} 0, & if \quad \bar{y} < 0.5 \\ 1, & if \quad \bar{y} >= 0.5 \end{cases} \tag{2.7}$$

Το δίκτυο προπονείται με σκοπό να μεγιστοποιήσει τον λογάριθμο της δεσμευμένης πιθανότητας $logP(y = 1|x)$ για κάθε δεδομένο εκπαίδευσης (x, y). Η λογιστική απώλεια ορίζεται ως:

$$L_{logistic}(\hat{y}, y) = -ylog\hat{y} - (1-y)log(1-\hat{y}) \tag{2.8}$$

Όταν χρησιμοποιείται η λογιστική συνάρτηση κόστους, θεωρείται ότι το εξωτερικό επίπεδο μετασχηματίζεται με τη χρήση της σιγμοειδούς συνάρτησης.

Η διακριτή συνάρτηση κόστους cross-entropy (γνωστή και ως αρνητική λογιστική συνάρτηση πιθανοφάνειας) χρησιμοποιείται όταν είναι επιθυμητή μία πιθανοτική ερμηνεία των αποτελεσμάτων. Έστω $y = y_{|1|}, ..., y_{|n|}$ ένα διάνυσμα που αναπαριστά την πραγματική multinominal κατανομή στις ετικέτες 1,..,n και έστω $\hat{y} = \hat{y}_{|1|}, ..., \hat{y}_{|n|}$ η έξοδος του γραμμικού ταξινομητή, η οποία μετασχηματίζεται από την συνάρτηση softmax και αναπαριστά την δεσμευμένη κατανομή του να ανήκει στην κλάση ένα δείγμα στην κλάση i, $\hat{y}_{|i|} = P(y = 1|x)$.

Τότε η διακριτή συνάρτηση κόστους cross-entropy για το n-οστό δείγμα είναι:

$$L_{cross-entropy}(\hat{y}_i, y_i) = -y_{|i|} log \hat{y_{|i|}} \tag{2.9}$$

Τελικώς, για να βελτιστοποιήσουμε τις παραμέτρους του μοντέλου μας, σκόπος μας είναι να μεγιστοποιήσουμε την πιθανοφάνειά του, ή αλλιώς να ελαχιστοποιήσουμε το μέσο όρο της αρνητικής λογιστικής πιθανοφάνειας όλων των διαθέσιμων N δειγμάτων εκπαίδευσης. Η αντικειμενική συνάρτηση (συνάρτηση κόστους) παίρνει την εξής μορφή (N ο αριθμός δειγμάτων της εκπαίδευσης):

$$L_{cross-entropy}(\hat{y}_i, y_i) = -\frac{1}{N} \sum_{i=1}^{N} y_{|i|} log(\hat{y}_{|i|}) \tag{2.10}$$

Η παραπάνω εξίσωση είναι αρκετά χρήσιμη για την εκπαίδευση νευρωνικών δικτύων (NN - neural networks), καθώς η τιμή της συνάρτησης κόστους επιτρέπει τον υπολογισμό του σφάλματος του νευρωνικού δικτύου, ως προς τις αποφάσεις ταξινόμησης που έλαβε για N δείγματα. Για να μάθει το NN τις βέλτιστες υπερπαραμέτρους χρησιμοποιείται ένας optimizer, ο οποίος βασίζεται στον υπολογισμό του ανάδελτα (gradient) $\nabla_\theta L(\hat{y}, y)$ για την εύρεση ενός τοπικού ελαχίστου. Ο πιο διάσημος αλγόριθμος για την βελτιστοποίηση των βαρών είναι αυτός του backpropagation [10]. Ο αλγόριθμος backpropagation στηρίζεται στον επαναλαμβανόμενο υπολογισμό των μερικών παραγώγων (gradients) κάθε επιπέδου ενός NN σε σχέση με τις παραμέτρους που χρειάζεται να ρυθμιστούν χρησιμοποιώντας τον κανόνα αλυσίδας, για να ελαχιστοποιηθεί η απώλεια. Τα βάρη του δικτύου ενημερώνονται αντίστοιχα. Το σφάλμα που υπολογίζεται από τις μερικές παραγώγους διαμορφώνει το κατά πόσο θα μεταβληθούν τα βάρη.

Στην πραγματικότητα, αυτό που προσπαθούμε να κάνουμε όταν χρησιμοποιούμε τον αλγόριθμο backpropagation είναι να προσεγγίσουμε το τοπικό ελάχιστο ενός μη-γραμμικού προβλήματος ελαχιστοποίησης. Το πρόβλημα αυτό δε μπορεί να λυθεί σε πολυωνυμικό χρόνο από κανέναν αλγόριθμο(ανήκει σε μια κατηγορία προβλημάτων που ονομάζονται ΝΠ-προβλήματα).

2.2.2 Λογιστική Παλινδρόμηση (Logistic Regression)

Η λογιστική παλινδρόμηση (Logistic Regression - LR) αποτελεί ένα μοντέλο γραμμικού ταξινομητή. Η LR υπολογίζει τις πιθανότητες για πρόβλημα ταξινόμησης με δύο δυνατά αποτελέσματα, εφαρμόζοντας την σιγμοειδή συνάρτηση πάνω στην γραμμική συνάρτηση παλινδρόμησης f, ώστε να λάβουμε την τελική απόφαση ταξινόμησης. Η σιγμοειδής συνάρτηση (ή αλλιώς λογιστική συνάρτηση) συμπιέζει ένα διάνυσμα στό διάστημα (0, 1). Για ένα πρόβλημα δυαδικής ταξινόμησης, η πιθανότητα μία εκ των κλάσεων για ένα διάνυσμα χαρακτηριστικών (feature vector) $x \in R^d$ υπολογίζεται ως:

$$P(y = 1|x) = \frac{1}{1 + e^{-f(x)}} \tag{2.11}$$

όπου f είναι μία γραμμική συνάρτηση με παραμέτρους w_i:

$$f(x) = w_0 + w_1 x_1 + .. + w_d x_d \tag{2.12}$$

Η πιθανότητα της άλλης κλάσης θα είναι $P(y = 0|x) = 1 - P(y = 1|x)$. Σε κάθε είσοδο αντιστοιχίζεται η ετικέτα της κλάσης, στην οποία η πιθανότητα είναι μεγαλύτερη από 0.5.

Οι παράμετροι της γραμμικής συνάρτησης υπολογίζονται ελαχιστοποιώντας το κόστος της σύναρτησης απώλειας cross-entropy J, που ορίζεται:

$$J(w) = -(ylog(P(y = 1|x)) + (1 - y)log(1 - P(y = 1|x))) \tag{2.13}$$

Το πρόβλημα βελτιστοποίησης μπορεί να λυθεί χρησιμοποιώντας και τον αλγόριθμο κατάβασης πλαγιάς (gradient descent), για τον οποίο θα μιλήσουμε σε παρακάτω κεφάλαιο.

2.2.3 Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines - SVMSs)

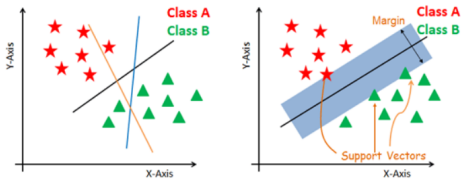
Ας υποθέσουμε ότι έχουμε ένα δυαδικό πρόβλημα ταξινόμησης, όπως και παραπάνω, με γραμμικό μοντέλο της μορφής:

$$f(x) = w^T \phi(x) + b \tag{2.14}$$

όπου $\phi(x)$ είναι ένας μετασχηματισμός στο χώρο των χαρακτηριστικών και η παράμετρος β (bias) είναι ορισμένη. Το σύνολο των δεδομένων εκπαίδευσης αποτελείται από N διανύσματα εισόδου $y = x_1, ..., x_N$ με αντίστοιχες τιμές εξόδου $y = y_1, ..., y_n$, όπου $y_i \in -1, 1$, και οποιοδήποτε νέο δείγμα x ταξινομείται με βάση το πρόσημο της συνάρτησης 2.14.

Υποθέτουμε ότι τα δεδομένα εκπαίδευσης είναι γραμμικά διαχωρίσιμα στο χώρο χαρακτηριστικών, ούτως ώστε εξ ορισμού υπάρχει τουλάχιστον μία επιλογή παραμέτρων ω και β, τέτοια ώστε μία συνάρτηση της μορφής 2.14 να ικανοποιεί την ανισότητα $f(x_i > 0)$ για δεδομένα που έχουν $y_i = +1$ και $f(x_i < 0)$ για δεδομένα που έχουν $y_i = -1$, έτσι ώστε το γινόμενο $y_i f(x_i > 0)$ για όλα τα δεδομένα της εκπαίδευσης.

Στο παρακάτω σχήμα 2.2 μπορούμε να παρατηρήσουμε ένα παράδειγμα τέτοιων δεδομένων εκπαίδευσης. Στην κατηγορία A ανήκουν τα κόκκινα αστέρια και στην κατηγορία B τα πράσινα αστέρια.



Εικόνα 2.2: *Παράδειγμα δυαδικού διαχωρισμού δύο γραμμικά διαχωρίσιμων κλάσεων*

Υπάρχει ένα άπειρο πλήθος πιθανών ευθειών που διαχωρίζουν τις παραπάνω 2 κλάσεις. Ο σκοπός του αλγόριθμου SVM είναι να βρει τον πιο γενικό ταξινομητή. Προσπαθεί, δηλαδή, να βρει το υπερεπίπεδο για το οποίο η ελάχιστη απόσταση μεταξύ των δύο κλάσεων (margin) έχει την μέγιστη δυνατή τιμή. Αυτό το υπερεπίπεδο μπορεί να παρατηρηθεί στην δεξιά εικόνα 2.2.

Αν η f(x) διαχωρίζει τα δείγματα, τότε η γεωμετρική απόσταση μεταξύ ενός σημείου x_i και

του υπερεπιπέδου f(x) = 0 θα είναι ίση με $\frac{|f(x_i)|}{||w||}$. Ενδιαφερόμαστε μόνο για τις λύσεις για τις οποίες όλα τα δείγματα ταξινομούνται σωστά, ούτως ώστε $y_i f(x_i > 0)$ για κάθε i. Έπειτα, η απόσταση μεταξύ ενός σημείου x_i και του βέλτιστου υπερεπιπέδου δίνεται από:

$$\frac{y_i f(x_i)}{||w||} = \frac{y_i(w^T \phi(x_i + b))}{||w||} \tag{2.15}$$

Το margin δίνεται από την κάθετη απόσταση στο κοντινότερο σημείο x_n από το σύνολο δεδομένων, και επιθυμούμε να βελτιστοποιήσουμε τις παραμέτρους w και b για να μεγιστοποιήσουμε αυτή την απόσταση. Λύνοντας την παρακάτω εξίσωση βρίσκουμε το μέγιστο margin.

$$L(w, b) = arg_{w,b} max(\frac{1}{||w||} min_i(y_i(w^T \phi(x_i + b)))) \tag{2.16}$$

Η μεγιστοποίηση του $\frac{1}{||w||}$ ισοδυναμεί με ελαχιστοποίηση του $\frac{1}{2}||w||^2$. Οπότε το πρόβλημα μετασχηματίζεται στα εξής δύο υποπροβλήματα.

$$L(w, b) = arg_{w,b} max(\frac{1}{2}||w||^2) \tag{2.17}$$

$$y_i(w^T \phi(x_i + b) >= 1, \qquad i = 1, ..., N \tag{2.18}$$

Η λύση της 2.17 δίνεται μέσω των πολλαπλασιαστών Lagrange [11].

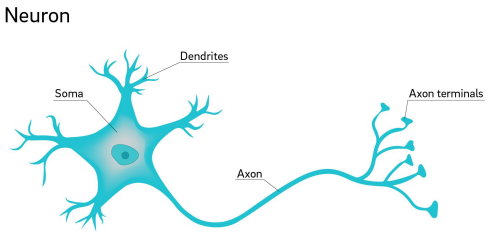
2.3 Βαθιά Μάθηση (Deep Learning)

2.3.1 Τεχνητά Νευρωνικά Δίκτυα

Για να έχουμε μια σφαιρική εικόνα της βαθιάς μάθησης (Deep Learning - DL), θα ασχοληθούμε πρώτα με την έννοια των τεχνητών νευρωνικών δικτύων.

Τα *τεχνητά νευρωνικά δίκτυα (artificial neural networks - ANN)* αποτελούν τα θεμέλια της βαθιάς μάθησης και ο συνδυασμός τους δημιουργεί τα βαθιά νευρωνικά δίκτυα. Αποτελούν ένα υπολογιστικό μοντέλο εμπνευσμένο από την βιολογία του ανθρώπινου εγκεφάλου [12]. Ο όρος ¨τεχνητά νευρωνικά δίκτυα' δεν περιγράφει έναν συγκεκριμένο αλγόριθμο, αλλά ουσιαστικά μια δομή για τον σχεδιασμό και την εκπαίδευση μοντέλων μηχανικής μάθησης, για μια πληθώρα διαφορετικών εργασιών. Τα ANN μπορούν να επεξεργαστούν οποιαδήποτε μορφή πληροφορίας η οποία παρουσιάζεται ως διάνυσμα, όπως κείμενο, εικόνα, ήχος κλπ. Αυτό γίνεται επειδή τα ANN έχουν την δυνατότητα να βγάζουν χαρακτηριστικά από ακατέργαστα δεδομένα εισόδου και να προπονούνται από αρχή μέχρι τέλους. Το γεγονός αυτό εξαφανίζει την ανάγκη για την παραγωγή χειροποίητων χαρακτηριστικών, μία διαδικασία που απαιτεί υψηλή γνώση του εκάστοτε τομέα, αλλά και είναι πολύ χρονοβόρα.

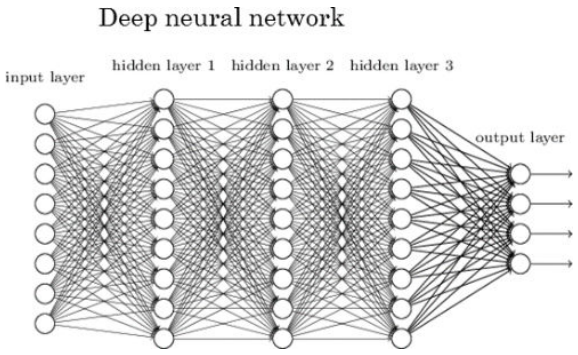
Η δημιουργία των ANNs, όπως αναφέρθηκε επηρεάστηκε σε μεγάλο βαθμό από την λειτουργία του ανθρώπινου εγκεφάλου. Η βασική υπολογιστική μονάδα του εγκεφάλου είναι ο *νευρώνας* 2.3. Υπάρχουν δισεκατομμύρια νευρώνες στο ανθρώπινο νευρικό σύστημα. Κάθε νευρώνας λαμβάνει σήματα εισόδου από τους *δενδρίτες* και παράγει σήματα εξόδου πάνω στον *άξονά* του. Ο άξονας συνδέει μέσω συνάψεων τους δενδρίτες των νευρώνων. Αντίστοιχα, στο



Εικόνα 2.3: *Παράδειγμα βιολογικού νευρώνα*

υπολογιστικό μοντέλο του νευρώνα, τα σήματα που ταξιδεύουν κατά μήκος των αξόνων, έστω x_0, βάσει της δύναμης της σύναψης, έστω w_0, αλληλεπιδρούν πολλαπλασιαστικά με τους δενδρίτες του άλλου νευρώνα , δηλαδή έστω w_0x_0. Η ιδέα είναι ότι οι δυνάμεις των συνάψεων (ή αλλιώς βάρη) μαθαίνονται και ελέγχουν την δύναμη της επιρροής του ενός νευρώνα στον άλλο. Στο βασικό μοντέλο, οι δενδρίτες μεταφέρουν το σήμα στο σώμα του κυττάρου, όπου όλα αθροίζονται. Αν το τελικό άθροισμα έχει τιμή μεγαλύτερη από ένα συγκεκριμένο όριο, ο νευρώνας ενεργοποιείται, στέλνοντας σήμα κατά μήκος του άξονα. Στο υπολογιστικό μοντέλο, υποθέτουμε ότι μας ενδιαφέρει μόνο η συχνότητα που στέλνονται αυτά τα σήματα. Επομένως, μοντελοποιούμε το ρυθμό αποστολής σημάτων του νευρώνα με μια *συνάρτηση ενεργοποίησης activation function* f, η οποία αναπαριστά τη συχνότητα αποστολής των σημάτων αυτών στον άξονα. Μία απο τις πιο συνηθισμένες επιλογές είναι και η σιγμοειδής συνάρτηση σ, που αναπτύξαμε παραπάνω.

Με την σχεδίαση και την υλοποίηση αρχιτεκτονικών που συνδυάζουν διαφορετικούς τεχνητούς νευρώνες, τα μοντέλα μπορούν να μάθουν πολύπλοκες μη γραμμικές συναρτήσεις. Αυτές οι αρχιτεκτονικές ονομάζονται πολυεπίπεδα αντίληπτρα (Multi-Layer Perceptrons - MLPs).



Εικόνα 2.4: *Βαθύ νευρωνικό δίκτυο*

Κάθε βαθύ νευρωνικό δίκτυο αποτελείται από τα εξής επίπεδα, όπως τα παρατηρούμε και στο σχήμα 2.4

- **Επίπεδο εισόδου (Input layer)**: Το αρχικό επίπεδο που λαμβάνει τα δεδομένα εισόδου. Παρέχει πληροφορίες από τον ¨έξω' κόσμο στο δίκτυο χωρις περαιτέρω επεξεργασία και οι κόμβοι απλά περνούν την πληροφορία στο επόμενο επίπεδο.

- **Κρυφά επίπεδα (Hidden layers)**: Τα αμέσως επόμενα επίπεδα, στα οποία η είσοδος περνάει επεξεργασία και εξάγονται τα χαρακτηριστικά της. Όσο προχωράμε προς ανώτερα κρυμμένα επίπεδα, εξάγονται χαρακτηριστικά ανωτέρου σημασιολογικού περιεχομένου.

- **Επίπεδο εξόδου (Output layer)**: Το τελευταίο επίπεδο που μετά την επεξεργασία των δεδομένων λαμβάνεται η απόφαση από το δίκτυο.

Συναρτήσεις Ενεργοποίησης (Activation Functions)

Οι συναρτήσεις ενεργοποίησης είναι κόμβοι που αποφασίζουν αν ένας νευρώνας πρέπει να ενεργοποιηθεί ή όχι. Τέτοιες συναρτήσεις είναι μη γραμμικές. Η επιλογή μη γραμμικών συναρτήσεων είναι απαραίτητη, γιατί χωρίς αυτές η χωρητικότητα του μοντέλου δεν θα μεγάλωνε παρά την αύξηση του βάθους του δικτύου. Ας έχουμε για παράδειγμα ένα γραμμικό νευρωνικό δίκτυο με ένα μόνο κρυφό επίπεδο με βάρη w_1 και w_2 και ένα επίπεδο εξόδου με βάρος w_0. Τότε έχουμε:

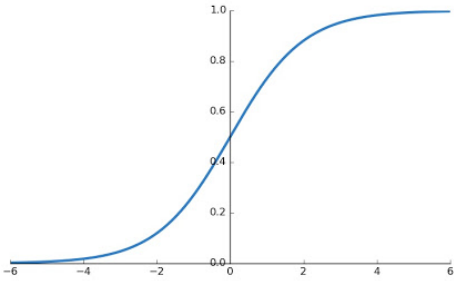
$$y = w_{01}(w_{11}x_1 + ... + w_{1n}x_n) + w_{02}(w_{21}x_1 + ... + w_{2n}x_n) = w_1x_1 + ... + w_nx_n \quad (2.19)$$

Ό,τι θα είχαμε δηλαδή και με ένα νευρωνικό δίκτυο χωρίς κρυμμένα επίπεδα. Μερικές από τις πιο συνηθισμένες συναρτήσεις ενεργοποίησης είναι οι παρακάτω:

Σιγμοειδής (Sigmoid)

Μια πολύ διαδεδομένη συνάρτηση ενεργοποίησης που συναντήσαμε και σε παραπάνω κεφάλαια είναι η σιγμοειδής. Ορίζεται ως εξής:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.20)$$



Εικόνα 2.5: *Σιγμοειδής συνάρτηση*

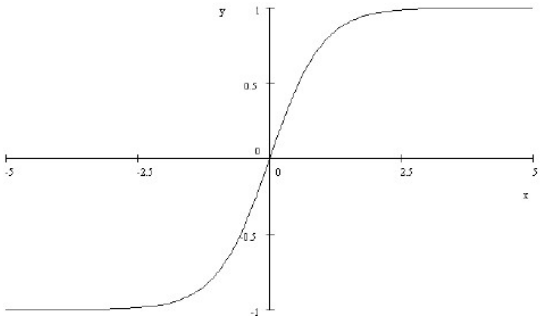
Όπως παρατηρούμε στην 2.5, η σιγμοειδής συνάρτηση συμπιέζει πραγματικούς αριθμούς στο διάστημα (0, 1). Συγκεκριμένα, μεγάλοι θετικοί αριθμοί παίρνουν τιμή 1 και αντίστοιχα, μεγάλοι αρνητικοί αριθμοί παίρνουν τιμή 0. Το γεγονός αυτό όμως δημιουργεί το πρόβλημα του λεγόμενου vanishing gradients. Σε περιοχές κορεσμού, δηλαδή, η κλίση (gradient) είναι

σχεδόν 0 που σημαίνει ότι τα βάρη κατά την διάρκεια της προπόνησης δεν θα αλλάζουν σχεδόν καθόλου. Επίσης, η χρήση της σιγμοειδούς ενεργοποίησης κάνει την διαδικασία μάθησης πολύ ευαίσθητη στην αρχικοποίηση των βαρών. Αν τα αρχικά βάρη είναι πολύ μεγάλα ή πολύ μικρά, τότε οι μικρές κλίσεις θα δρουν αρνητικά στην μάθηση.

Υπερβολική Εφαπτομένη (Hyperbolic tangent)

Η συνάρτηση υπερβολικής εφαπτομένης ορίζεται ως εξής:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.21}$$



Εικόνα 2.6: *Συνάρτηση Υπερβολικής Εφαπτομένης*

Η συγκεκριμένη συνάρτηση, όπως φαίνεται από την 2.6 μοιάζει αρκετά με την σιγμοειδή, αλλά έχει ως κέντρο της το μηδέν. Η υπερβολική εφαπτομένη συμπιέζει τις τιμές πραγματικών αριθμών στο διάστημα (-1, 1) και είναι ουσιαστικά μία κλιμακωμένη σιγμοειδή:

$$tanh(x) = 2 * \sigma(2x) - 1 \tag{2.22}$$

Επομένως, και αυτή υποφέρει από το πρόβλημα των vanishing gradients.

Rectified Linear Unit(ReLU)

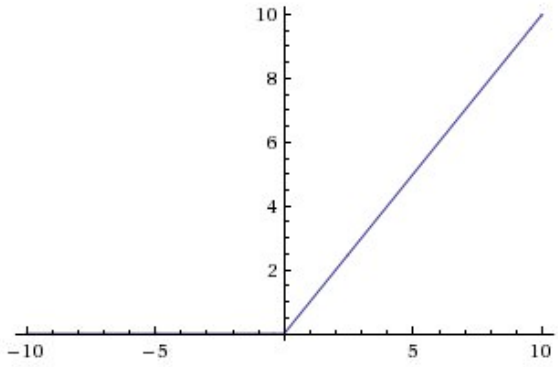
Η συνάρτηση ReLU ορίζεται ως εξής:

$$f(x) = max(0, x) \tag{2.23}$$

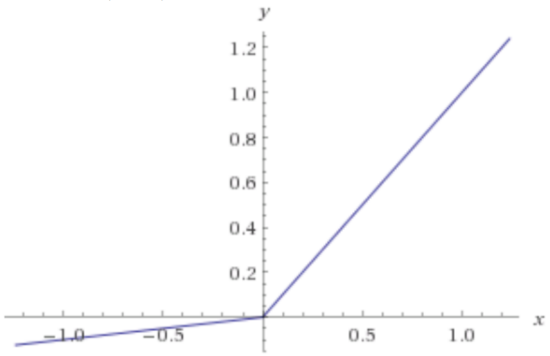
Η συνάρτηση ReLU αποτελεί μία από τις πιο συνηθισμένες συναρτήσεις ενεργοποίησης στα νευρωνικά δίκτυα. Δεν έχει υπολογιστικά ακριβές πράξεις (εκθέτες) και συγκλίνει γρηγορότερα. Επιπλέον, έχει χαμηλή πυκνότητα, γεγονός που προκύπτει αφού με κάθε αρνητική είσοδο, η συνάρτηση δεν ενεργοποιείται. Αυτό σημαίνει ότι οι νευρώνες ενεργοποιούνται μόνο όταν επεξεργάζονται ενδιαφέρουσες εισόδους για το πρόβλημά τους. Αποφεύγει επίσης το πρόβλημα των vanishing gradients, λόγω της γραμμικότητας της. Παρόλα αυτά τα αρνητικό της είναι ότι όταν έχουμε αρνητικές τιμές η έξοδος είναι πάντα 0 και ο νευρώνας ουσιαστικά νεκρώνεται, και δεν είναι οριοθετημένη, επομένως μπορεί να οδηγήσει σε εκρήξεις σε βαθιά δίκτυα.

Για να λυθεί το πρόβλημα των νεκρωμένων νευρώνων, προτάθηκε η leaky ReLU, η οποία επιτρέπει μία μικρή τιμή σε περίπτωση αρνητικών εισόδων:

$$prediction = \begin{cases} x, & if \quad x > 0 \\ ax, & if \quad x < 0 \end{cases} \tag{2.24}$$



Εικόνα 2.7: *Συνάρτηση ReLU*



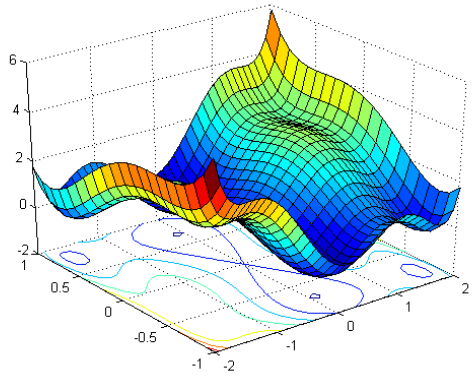
Εικόνα 2.8: *Συνάρτηση "leaky" ReLU*

2.3.2 Προπόνηση τεχνητών νευρωνικών δικτύων

Στο κεφάλαιο Loss Function μιλήσαμε για την συνάρτηση απωλειών. Η προπόνηση των νευρωνικών δικτύων είναι ουσιαστικά η διαδικασία που μειώνει την μέση συνάρτηση απωλειών J(w) πάνω στα δεδομένα εκπαίδευσης. Για να γίνει αυτό εφαρμόζονται τεχνικές βασισμένες στα gradients. Gradient ενός συγκεκριμένου σημείου ορίζουμε την κλίση της εφαπτομένης της συνάρτησης σε εκείνο το σημείο και έχει κατεύθυνση προς τη μεγαλύτερη αύξηση της συνάρτησης.

Ο αλγόριθμος **Απότομης Καθόδου (gradient descent - GD)** είναι ένας από τους πιο δημοφιλείς αλγορίθμους βελτιστοποίησης στα νευρωνικά δίκτυα. Ο GD είναι μία επαναληπτική διαδικασία. Οι παράμετροι w αρχικοποιούνται τυχαία σε κάθε επανάληψη και προσαρμόζονται στον χώρο των παραμέτρων κάνοντας ένα βήμα ακολουθώντας την κατεύθυνση της αρνητικής κλίσης της συνάρτηση κόστους $\nabla J(w)$. Η μεγαλύτερη διαφορά της εκπαίδευσης ενός γραμμικού μοντέλου και ενός βαθιού νευρωνικού δικτύου είναι ότι η μη γραμμικότητα

των νευρωνικών δικτύων οδηγεί στην επιφάνεια κόστους να γίνεται μη κυρτή 2.9. Αυτό σημαίνει ότι δεν είναι σίγουρο ότι μία μέθοδος βασισμένη στις κλίσεις θα οδηγήσει σε ένα ολικό ελάχιστο όπου η συνάρτηση κόστους θα είναι 0. Μάλιστα το πρόβλημα της βελτιστοποίησης των βαθιών νευρωνικών δικτύων ανήκει στην κατηγορία των NP-hard [13]. Αυτό σημαίνει ότι δεν υπάρχει αλγόριθμος που βρίσκει σε πολυωνιμικό χρόνο τις βέλτιστες υπερπαραμέτρους ενός βαθιού NN.



Εικόνα 2.9: *Μη κυρτή (non-convex) συνάρτηση κόστους*

Οι παράμετροι w στην i-οστή επανάληψη υπολογίζονται ως εξής:

$$w_{i+1} = w_i - \alpha \nabla J(w_i) \tag{2.25}$$

όπου α είναι ο ρυθμός μάθησης. Η κλίση δείχνει την κατεύθυνση που πρέπει να αναβαθμίσουμε τα βάρη αλλά δεν δίνει πληροφορία για το μέγεθος του βήματος. Αυτό ελέγχεται από τον ρυθμό μάθησης, ο οποίος αποτελεί μία από τις πιο σημαντικές υπερπαραμέτρους της προπόνησης των ANN. Από την μία μεριά, αν ο ρυθμός είναι πολύ μεγάλος τότε μπορεί να περάσει το ολικό ελάχιστο και να πηγαίνει μπρος πίσω στην επιφάνεια του κόστους. Από την άλλη, αν είναι πολύ μικρός μπορεί να πάρει πάρα πολύ να φτάσει σε κάποιο τοπικό ελάχιστο και τελικά να κολλήσει ο αλγόριθμος σε κάποιο μη ιδανικό τοπικό ελάχιστο.

Δυσκολίες της Απότομης Καθόδου

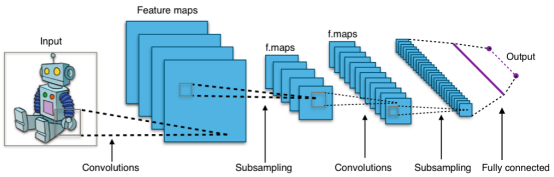
Ο GD αντιμετωπίζει δύο συχνά προβλήματα που πηγάζουν από την βελτιστοποίηση μίας μη κυρτής συνάρτησης. Αυτά τα προβλήματα είναι τα τοπικά ελάχιστα και τα saddle points. Υπάρχει πιθανότητα, λόγω της κλίσης, να φτάσει σε ένα μη ιδανικό σημείο που να μην μπορεί να ξεφύγει καθώς η κλίση γίνεται πολύ μικρή. Τα τοπικά ελάχιστα είναι επίσης αρκετά προβληματικά, σε περίπτωση που έχουν πολυ μεγαλύτερο κόστος σε σχέση με το ολικό ελάχιστο. Τα τελευταία χρόνια, όμως, έχουν αναπτυχθεί διάφοροι νέοι αλγόριθμοι βασισμένοι πάνω στον ΓΔ, που αντιμετωπίζουν αυτά τα προβλήματα. Τέτοιοι είναι ο Adadelta [14], ο Adagrad [15] και ο Adam [16]. αλγόριθμος σε κάποιο μη ιδανικό τοπικό ελάχιστο.

Οπίσθια διάδοση (Backpropagation)

Για την ελαχιστοποίηση της συνάρτησης κόστους $J(w)$ ενός νευρωνικού δικτύου χρησιμοποιώντας το βέλτιστο σύνολο τιμών για τα βάρη w πρέπει να υπολογιστεί το gradient. Χρησιμοποιώντας τον κανόνα της αλυσίδας μπορεί να υπολογιστεί το gradient, όμως σε μπορεί να οδηγήσει σε σφάλματα για πολύπλοκα δίκτυα. Η λύση δίνεται αποδοτικά με τον αλγόριθμο του backpropagation [17]. Ο αλγόριθμος αυτός υπολογίζει συστηματικά της παραγώγους μιας περίπλοκης μαθηματικής έκφρασης χρησιμοποιώντας τον κανόνα αλυσίδας και αποθηκεύοντας τα ενδιάμεσα αποτελέσματα.

2.3.3 Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNNs)

Τα συνελικτικά νευρωνικά δίκτυα είναι ένας τύπος τροφοδοτικού δικτύου που επεξεργάζεται δομημένα δεδομένα με την μέθοδο της συνέλιξης. Τα CNN είναι εμπνευσμένα από τον οπτικό φλοιό τον ζώων. Τα ζώα έχουν δύο ειδών οπτικών νευρωνικών δικτύων, ένα που ανιχνεύει βασικά σχήματα σε ένα μικρό δεκτικό πεδίο και ένα σε ένα μεγαλύτερο δεκτικό πεδίο για πιο χωρικά αμετάβλητες απαντήσεις. Κατά συνέπεια, τα CNN χρησιμοποιήθηκαν κυρίως στο πεδίο της όρασης υπολογιστών, όμως τα τελευταία χρόνια έχουν δείξει ότι μπορούν να αντεπεξέλθουν και σε διάφορα άλλα προβλήματα που σχετίζονται με ήχο και κείμενο. Γενικά τα CNN μπορούν να επεξεργαστούν οποιαδήποτε δεδομένο το οποίο έχει δομή πλέγματος. Για παράδειγμα, οι εικόνες συνήθως κωδικοποιούνται σαν 3-διάστατες ποσότητες από ένα κόκκινο, πράσινο και μπλε κανάλι (RGB encoding) ή ο ήχος μπορεί να πάρει την μορφή σπεκτρογράμματος και ούτω καθεξής.



Εικόνα 2.10: *Τυπική αρχιτεκτονική συνελικτικού νευρωνικού δικτύου*

Ένα επίπεδο ονομάζεται συνελικτικό, αν τα βάρη του εφαρμόζονται στην είσοδο με την διαδικασία της συνέλιξης και όχι του πολλαπλασιασμού πινάκων που είδαμε παραπάνω. Η συνέλιξη δύο συναρτήσεων είναι ο εσωτερικός μετασχηματισμός που μετράει πόσο μια συνάρτηση τροποποιεί την άλλη, όταν η μία περνάει πάνω από την άλλη. Με συνάρτηση εισόδου x, συνάρτηση βαρών w, τότε η συνέλιξη $x * w$ ορίζεται ως:

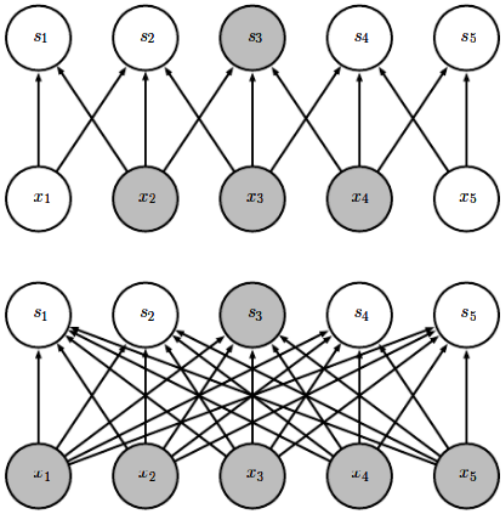
$$s(t) = x(t) * w(t) = \int x(h)w(t-h)dh \tag{2.26}$$

Η συνάρτηση βαρών συχνά αποκαλείται φίλτρο (filter) ή πυρήνας (kernel) και εν συνέχει η έξοδος της συνέλιξης αποκαλείται χάρτης χαρακτηριστικών (feature map). Το μέγεθος των kernels είναι συνήθως αρκετά μικρότερο της εισόδου. Με αυτό τον τρόπο τα CNN έχουν την δυνατότητα να μάθουν τις ιεραρχίες των μοτίβων. Τα πρώτα επίπεδα των CNN εξάγουν απλά, τοπικά μοτίβα τα οποία σταδιακά προχωράνε σε πιο πολύπλοκα και αφηρημένα καθώς

κινούμαστε βαθύτερα στο δίκτυο. Τα επίπεδα συνέλιξης έχουν συνήθως τα εξής στάδια:

- **Στάδιο συνέλιξης**: Το πρώτο βήμα είναι να εφαρμοστεί η συνέλιξη μεταξύ της εισόδου και ενός συνόλου από kernels.
- **Στάδιο ανίχνευσης**: Το δεύτερο βήμα είναι μία εφαρμογή μιας συνάρτησης ενεργοποίησης, όπως η ReLU.
- **Στάδιο pooling**: Το τελευταίο βήμα στο οποίο γίνεται στα χαρακτηριστικά υποδειγματοληψία. Κάθε χαρακτηριστικό στον χάρτη χαρακτηριστικών εξόδου αποτελεί στατιστικό άθροισμα από μια γειτονιά χαρακτηριστικών.

Η διαδικασία της ομαδοποίησης (pooling) είναι εξαιρετικά σημαντική για την έννοια των CNNs, καθώς επιτρέπει σε συνεχόμενα επίπεδα να επεξεργάζονται αντιπροσωπεύσεις από όλο και μεγαλύτερα υποκομμάτια της αρχικής εισόδου. Επίσης τα στάδια ομαδοποίησης μειώνουν το μέγεθος της εξόδου που ως συνέπεια μειώνει και τις παραμέτρους του μοντέλου. Η ομαδοποίηση μοιάζει με την συνέλιξη με την έννοια ότι επιδρούν πάνω σε γειτονιές χαρακτηριστικών, όμως η διαφορά τους είναι ότι αντί να χρησιμοποιούν ένα φίλτρο που έχουν μάθει, χρησιμοποιούν μια 'hardcoded' συνάρτηση, η οποία συνήθως είναι max ή average pooling. Η πρώτη επιλέγει το χαρακτηριστικό με την μέγιστη τιμή, ενώ η δεύτερη υπολογίζει τον μέσο όρο της γειτονιάς. Γενικά στην πράξη προτιμάται το max pooling.



Εικόνα 2.11: *Πυκνή συνδεσιμότητα (κάτω), Αραιή συνδεσιμότητα (πάνω)*

Τα CNNs έχουν 3 σημαντικές ιδιότητες:

- *Αραιή συνδεσιμότητα*: Στα fully-connected επίπεδα, κάθε έξοδος υπολογίζεται συνάρτηση όλων των τιμών εισόδου. Κάτι τέτοιο οδηγεί σε προβλήματα όταν η είσοδος είναι πολλαπλών διαστάσεων. Τα CNNs αντιμετωπίζουν αυτό το πρόβλημα μέσω αραιής συνδεσιμότητας 2.11. Αυτό σημαίνει ότι κάθε νευρώνας εξόδου συνδέεται με μία περιοχή των νευρώνων εισόδου. Η έκταση της συνδεσιμότητας ορίζεται από το μέγεθος του πυρήνα (kernel) και συμβάλει στην μείωση των παραμέτρων του μοντέλου, κάτι που εν

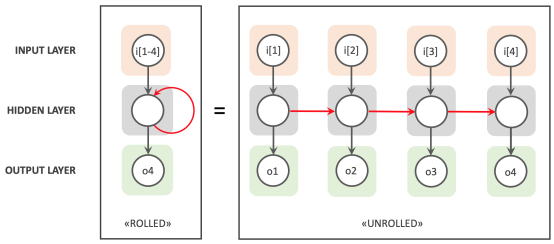
γένει μειώνει τον κίνδυνο για υπερπροπόνηση (overfitting), αλλά και τις απαιτήσεις σε μνήμη.

- *Μοιρασιά παραμέτρων*: Ένα άλλο πλεονέκτημα των CNNs είναι ότι μοιράζονται οι παράμετροι. Είναι επιθυμητό να έχουμε την εξαγωγή παρόμοιων χαρακτηριστικών σε διαφορετικές περιοχές της εισόδου. Ο πυρήνας είναι ουσιαστικά ένα φίλτρο που περνάει από όλη την είσοδο και κάθε στοιχείο του πυρήνα εφαρμόζεται σε κάθε στοιχείο της εισόδου. Αυτό αποτελεί μία τεχνική που μειώνει σημαντικά το πλήθος των παραμέτρων του μοντέλου.

- *Ισοδύναμες αναπαραστάσεις*: Ένα αποτέλεσμα της μοιρασιάς των παραμέτρων είναι ότι οι αναπαραστάσεις που μαθαίνονται είναι ισοδύναμες με μία μετάφραση. Μία συνάρτηση f είναι ισοδύναμη με μία g, αν $f(g(x)) = g(f(x))$. Αυτό σημαίνει ότι εφαρμόζοντας ένα συνελικτικό επίπεδο σε μία μεταφρασμένη είσοδο, θα φέρει τα ίδια αποτελέσματα με την μετάφραση της εξόδου της συνέλιξης. Αυτή η ιδιότητα εξηγεί γιατί ένας πυρήνας εντοπίζει ένα συγκεκριμένο πρότυπο στην είδοσο. Για να βρούμε πολλά χαρακτηριστικά σε κάθε επίπεδο, δεν χρησιμοποιούμε έναν μόνο πυρήνα, αλλά ένα σύνολο από αυτούς.

2.3.4 Αναδρονομικά Νευρωνικά Δίκτυα (Recurrent Neuran Networks)

Συχνά, στα Νευρωνικά Δίκτυα θεωρούμε ότι όλες οι είσοδοι και οι έξοδοί τους είναι ανεξάρτητες μεταξύ τους. Αυτό δεν συμβαίνει όμως σε κάθε πρόβλημα, όπως για παράδειγμα, αν θέλουμε να προβλέψουμε την επομένη λέξη σε μία πρόταση, χρειάζεται να γνωρίζουμε τις λέξεις που προηγήθηκαν. Σε αυτό το πρόβλημα απαντάνε τα *Αναδρομικά Νευρωνικά Δίκτυα*. Τα Αναδρομικά Νευρωνικά Δίκτυα (RNNs) αποτελούν έναν ισχυρό τύπο νευρωνικών δικτύων, λόγω της μοναδικής τους ιδιότητας να έχουν εσωτερική μνήμη. Οι συνδέσεις μεταξύ των μονάδων σε ένα RNN δημιουργούν ένα κατευθυνόμενο γράφο σε μία ακολουθία, κάτι που επιτρέπει στο δίκτυο να παρουσιάζει δυναμική χρονική συμπεριφορά σε μία ακολουθία.



Εικόνα 2.12: *Αρχιτεκτονική RNN*

Τα RNNs έχουν την δυνατότητα να θυμούνται πληροφορίες σχετικά με την είσοδο που έχουν δεχθεί, γεγονός που τους επιτρέπει να κάνουν ακριβείς προβλέψεις για τα δεδομένα που θα ακολουθήσουν. Τα βασικά RNN είναι κόμβοι οργανωμένοι σε διαδοχικά επίπεδα 2.12. Η διαδικασία λειτουργεί ως εξής:

Το RNN παίρνει το x_0 από την ακολουθία εισόδου και παράγει το h_0. Στην συνέχεια το h_0 μαζί με το x_1 αποτελούν την είσοδο για το επόμενο βήμα και ούτω καθεξής. Με αυτόν τον τρόπο το RNN μπορεί να θυμάται το περιεχόμενο της εισόδου που έχει δεχθεί ήδη κατά την διάρκεια της εκπαίδευσης.

Με μαθηματικούς όρους την χρονική στιγμή t ισχύουν για το RNN:

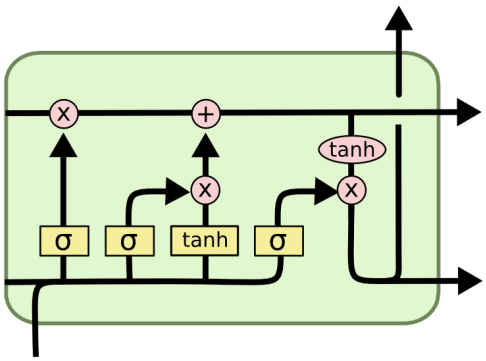
$$h_t = f_h(W_{hh}h_{t-1} + W_{hx}x_t + b_h) \tag{2.27}$$

$$y_t = f_y * (W_{yh}ht + b_y) \tag{2.28}$$

όπου h_t η κρυφή κατάσταση την στιγμή t, x_t το διάνυσμα εισόδου την στιγμή t, b_h, το bias για το h, b_y, το bias για το y, και $f(x), f(h)$ οι συναρτήσεις ενεργοποίησης για τα x, h αντίστοιχα. Υπάρχουν τρεις διαφορετικοί πίνακες βαρών. W_{hx} για τα βάρη από την είσοδο στο κρυφό επίπεδο, W_{hh} για τα βάρη από το κρυφό στο κρυφό επίπεδο και W_{yh} για τα βάρη από το κρυφό επίπεδο προς την έξοδο.

Δίκτυο μακράς βραχυπρόθεσμης μνήμης (Long Short-Term Memory - LSTM) unit

Θεωρητικά, τα RNNs μπορούν να διατηρήσουν πληροφορία από συνδέσεις μεγάλων αποστάσεων μεταξύ των ακολουθιών εισόδου. Στην πράξη όμως εμφανίζεται ένα υπολογιστικό πρόβλημα. Κατά την διάρκεια της εκπαίδευσης ενός RNN με τον αλγόριθμο του back propagation, τα gradients που διαδίδονται προς τα πίσω είναι πιθανό να πάρουν πολύ μικρές τιμές (τείνουν στο μηδέν) ή να εκραγούν (τείνουν στο άπειρο), επειδή οι υπολογισμοί που γίνονται κατά τη διαδικασία αυτή χρησιμοποιούν αριθμού πεπερασμένες ακρίβειας.



Εικόνα 2.13: *Κελί LSTM*

Στην εικόνα 2.13 παρατηρούμε την αρχιτεκτονική του LSTM [18]. Τα LSTM ξεπερνούν σε μεγάλο βαθμό τα παραπάνω προβλήματα προφυλάσσοντας τις συνδέσεις μεγάλων αποστάσεων ανάμεσα σε λέξεις και διαγράφουν πληροφορίες που δεν είναι σημαντικές από την πύλη του κυττάρου (cell gate), μέσω του επιπέδου της λήθης (forget gate).

Αν έχουμε μία ακολουθία $x_1, x_2, ..., x_t, ..., x_n$ διανυσμάτων μια ακολουθίας εισόδου, τότε για το διάνυσμα x_t, με εισόδους h_{t-1} και c_{t-1}, υπολογίζονται τα h_t και c_t ως εξής:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.29}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2.30}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.31}$$

$$u_t = tanh(W_u x_t + U_u h_{t-1} + b_u) \tag{2.32}$$

$$c_t = f_t \bigodot c_{t-1} + i_t \bigodot u_t \tag{2.33}$$

$$h_t = o_t \bigodot tanh(c_t) \tag{2.34}$$

- *Πύλη λήθης (Forget gate) (f_t)*: Η πύλη αυτή αποφασίζει για το ποιες πληροφορίες θα διατηρηθούν και ποιες θα διαγραφούν. Πληροφορίες από την προηγούμενη κατάσταση h_{t-1} και πληροφορίες από την τωρινή είσοδο x_t περνούν από μια σιγμοειδή συνάρτηση ενεργοποίησης. Όσο πιο κοντά στο 0 είναι οι τιμές τόσο πιο πιθανόν να διαγραφούν, ενώ όσο πιο κοντά στο 1 να διατηρηθούν.

- *Πύλη ειδόσου (Input gate) (i_t)*: Η κρυφή κατάσταση και η τωρινή είσοδος περνούν επίσης στην συνάρτηση tanh για να λάβουν τιμές ανάμεσα στο -1 και το 1 (u_t). Τελικά η έξοδος της σιγμοειδούς πολλαπλασιάζεται με την έξοδο της tanh. Η σιγμοειδής φιλτράρει τις σημαντικές πληροφορίες της tanh.

- *Κατάσταση κυττάρου (Cell state) (c_t)*: Η κατάσταση κυττάρου πολλαπλασιάζεται με το διάνυσμα λήθης. Με αυτό τον τρόπο απορρίπτονται οι τιμές αν πολλαπλασιαστούν με τιμές που τείνουν στο 0. Έπειτα, παίρνουμε την έξοδο της πύλης εισόδου και τις αθροίζουμε, οπότε λαμβάνουμε νέες τιμές για την κατάσταση κυττάρου, οι οποίες είναι πιο σχετικές με το πρόβλημά μας.

- *Πύλη εξόδου (Output gate) (o_t)*: Η πύλη εξόδου καθορίζει ποια θα είναι η επόμενη κρυφή κατάσταση. Αρχικά, η προηγούμενη κρυφή κατάσταση και η τωρινή είσοδος περνούν στην σιγμοειδή. Έπειτα, η αλλαγμένη κατάσταση του κυττάρου περνάει στην συνάρτηση tanh. Πολλαπλασιάζουμε την έξοδο της tanh με τη σιγμοειδή έξοδο, για να αποφασιστεί ποιες πληροφορίες θα πρέπει να διατηρήσει η κρυφή κατάσταση. Η έξοδος είναι η νέα κρυφή κατάσταση. Η νέα κατάσταση κυττάρου και η νέα κρυφή κατάσταση προχωρούν και στην επόμενη χρονική στιγμή.

2.3.5 Μηχανισμοί Προσοχής (Attention Mechanisms)

Ένα ακόμα πολύ χρήσιμο εργαλείο στο πεδίο της βαθιάς μάθησης, το οποίο έχει καθιερωθεί τα τελευταία χρόνια είναι η *Προσοχή (Attention)*. Αρχικά, attention χρησιμοποιήθηκε στο πρόβλημα του Neural Machine Translation [19], ώστε να ακολουθεί δυναμικά την αναπαράσταση της ακολουθίας εισόδου και να προβλέπει την επόμενη λέξη στην μεταφρασμένη

ακολουθία. Από τότε οι μηχανισμοί attention χρησιμοποιούνται σε πολλά διαφορετικά προβλήματα με τεράστια επιτυχία και state-of-art αποτελέσματα σε πολλά από αυτά.

Ο ρόλος του attention είναι να δίνει μεγαλύτερη σημασία στα σημαντικότερα κομμάτια της εισόδου υπολογίζοντας βάρη σημασίας για την αναπαράστασή της. Για παράδειγμα, αυτά τα σημαντικά κομμάτια μπορεί να είναι περιοχές μιας εικόνας ή λέξεις σε ένα κείμενο. Προκειμένου να εντοπίσουμε τα σημαντικά διανύσματα της εισόδου, θέτουμε ένα βάρος a_i, στο κρυφό βήμα που αντιστοιχεί σε κάθε διάνυσμα h_i. Υπολογίζουμε την πεπερασμένη αναπαράσταση ρ όλης της ακολουθίας εισόδου, ως το σταθμισμένο άθροισμα όλων των κρυφών καταστάσεων.

$$e_i = tanh(W_h h_i + b_h) \tag{2.35}$$

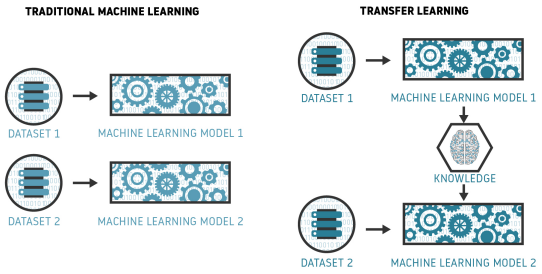
$$a_i = \frac{exp(e_i)}{\sum_{t=1}^{T} exp(e_i)}, \quad \sum_{i=1}^{T} a_i = 1 \tag{2.36}$$

$$r = \sum_{i=1}^{T} a_i h_i \tag{2.37}$$

όπου W_h και b_h είναι τα βάρη του επιπέδου attention.

2.4 Μεταφορά Μάθησης (Transfer Learning - TL)

Στη μηχανική μάθηση, και ειδικά στην βαθιά μάθηση, υπάρχει ένα σημαντικό πρόβλημα. Τα δίκτυα που επιλύουν περίπλοκα προβλήματα απαιτούν πολύ μεγάλες ποσότητες δεδομένων που πολλές φορές η απόκτηση τους για τα επιβλεπόμενα μοντέλα είναι αδύνατη λόγω χρονικών ή/και υπολογιστικών περιορισμών. Επίσης, μοντέλα που έχουν εκπαιδευθεί σε μικρά και ειδικά σύνολα δεδομένων δεν έχουν την δυνατότητα να χρησιμοποιηθούν για να αντιμετωπίσουν άλλα προβλήματα, εκτός του αρχικού που έχουν εκπαιδευτεί, ακόμα και αν αυτά είναι αρκετά παρεμφερή.



Εικόνα 2.14: *Η ιδέα της Μεταφοράς Μάθησης*

Λόγω αυτής της ανάγκης, άρχισε να αναπτύσσεται η τεχνική της *Μεταφοράς Μάθησης Tranfer Learning*. Ο στόχος του transfer learning είναι να βελτιώσει την εκμάθηση ενός προβλήματος χρησιμοποιώντας γνώση από ένα διαφορετικό πρόβλημα, εικόνα 2.14.

Στην μεταφορά μάθησης [20] έχουμε την έννοια του domain και του task. Ένας τομέας (domain) D αποτελείται από ένα χώρο χαρακτηριστικών X και μία περιθώρια κατανομή πιθανότητας P(X) στο χώρο των χαρακτηριστικών. Δεδομένου ενός τομέα $D = \{X, P(X)\}$

ένα Task T αποτελείται από τον χώρο ετικετών Y και μία δεσμευμένη κατανομή πιθανότητας $P(Y|X)$, η οποία είναι το αποτέλεσμα της μάθησης με από τα δεδομένα εκπαίδευση που ανήκουν στο X.

Δεδομένου ενός τομέα πηγής (D_s), ενός προβλήματος πηγής (T_s), ενός τομέα στόχου (D_t) και ενός προβλήματος στόχου (T_s), η μεταφορά μάθησης έχει ως σκοπό, να μας επιτρέψει να μάθουμε την δεσμευμένη κατανομή πιθανότητας του στόχου $P(Y_t|X_t)$ στον D_t με τις πληροφορίες που έχουν συλλεχθεί στον D_s και T_s.

Υπάρχουν τρεις τρόποι με τους οποίους συνήθως η μεταφορά μάθησης βελτιώνει τη διαδικασία εκπαίδευσης [21].

- Η αρχική απόδοση που επιτυγχάνεται στο target task χρησιμοποιώντας μόνο την γνώση που έχει μεταφερθεί από το source task, προτού εκπαιδευτεί παραπάνω, σε σχέση με την αρχική απόδοση ενός τυχαία αρχικοποιημένου μοντέλου
- Ο χρόνος που χρειάζεται για να εκπαιδευτεί πλήρως το μοντέλο στο target task δεδομένης της γνώσης που έχει μεταφερθεί, σε σχέση με το πόσο χρόνο χρειάζεται για να το μάθει εξαρχής.
- Η τελική απόδοση στο target task με transfer learning, σε σχέση με την τελική απόδοση χωρίς transfer learning.

Κεφάλαιο 3

Προσέγγιση θέματος

Η κατανόηση της ψηφιακής πληροφορίας από τον άνθρωπο είναι μία διαδικασία που συμβαίνει αβίαστα μέσω των αισθήσεων μας, και κυρίως της όρασης. Μπορούμε πολύ εύκολα να αναγνωρίσουμε αντικείμενα και γεγονότα που απεικονίζονται σε μία εικόνα, και να ανακαλέσουμε εμπειρίες που έχουμε βιώσει στο παρελθόν ώστε να εξάγουμε γνώση για αυτό που παρατηρούμε.Εν συνεχεία, είναι εύκολο να παράξουμε μία ακριβή περιγραφή για το τι διαδραματίζεται στην εν λόγω εικόνα. Από την άλλη, οι υπολογιστές δεν έχουν αυτή την ικανότητα και δεν είναι σε θέση να εκτελέσουν αυτές τις λειτουργίες του ανθρώπινου εγκεφάλου.

Η *Όραση Υπολογιστών (Computer Vision - CV)* και η *Εξεργασία Φυσικής Γλώσσας (Natural Language Processing - NLP)* αποτελούν δύο κλάδους της τεχνητής νοημοσύνης που προσπαθούν να αναπαράγουν μέσω αλγορίθμων λειτουργίες του ανθρώπινου εγκεφάλου, όπως η όραση και η ομιλία. Από την μία μεριά μέσω του CV μπορούμε να παράξουμε συστήματα για περιγραφή και ταξινόμηση εικόνων, ενώ μέσω του NLP, συστήματα για παραγωγή και κατανόηση γραπτού λόγου. Ο συνδυασμός και των δύο μπορεί να οδηγήσει και στο ζητούμενο σύστημα του Image Captioning.

Στο παρών κεφάλαιο θα ασχοληθούμε παραπάνω με τους δύο αυτούς κλάδους και την χρησιμότητα τους στην συγκεκριμένη διπλωματική και επίσης θα γίνει μία εισαγωγή στα μοντέλα sequence-to-sequence.

3.1 Όραση Υπολογιστών (Computer Vision)

Ο κλάδος της Όρασης Υπολογιστών (Computer Vision - CV) ασχολείται με την αλγοριθμική αναπαράσταση τρισδιάστατων σκηνών, μέσω της επεξεργασίας δισδιάστατων εικόνων που προέρχονται από οπτικούς αισθητήρες (παράδειγμα [22]). Εν συνεχεία στόχος είναι η παραγωγή κάποιος πρόβλεψης ή κάποιου συμπεράσματος για την αρχική σκηνή. Τα συστήματα της Μηχανική Όρασης δεν έχουν κάποιο ενσωματωμένο τρόπο αναγνώρισης προτύπων στις εικόνες και δεν μπορούν να εντοπίσουν μικροαλλαγές στην τοποθέτηση της κάμερας ή στα διάφορα απεικονιζόμενα αντικείμενα. Η μόνη ικανότητα που έχουν είναι να διαχειρίζονται διανύσματα τιμών, που συχνά οι τιμές αυτές περιλαμβάνουν και θόρυβο. Οι εφαρμογές, λοιπόν, της όρασης υπολογιστών προσπαθεί να χρησιμοποιήσει αυτό το θορυβώδες σύνολο τιμών για να εξαγάγει συμπεράσματα για την πραγματική εικόνα και για τα χαρακτηριστικά που διαθέτει.

Για να επιτευχθεί αυτό υπάρχουν τριών ειδών αλγόριθμοι.

- *Αλγόριθμοι χαμηλού επιπέδου*: Αλγόριθμοι, οι οποίοι δέχονται σαν είσοδο μία εικόνα και παράγουν μία τροποποιημένη εικόνα σαν έξοδο.
- *Αλγόριθμοι ενδιάμεσου επιπέδου*: Αλγόριθμοι που με είσοδο μία εικόνα, παράγουν χαρακτηριστικά ανωτέρου επιπέδου, όπως για παράδειγμα οι ακμές μιας εικόνας.
- *Αλγόριθμοι υψηλού επιπέδου*: Αλγόριθμοι που επικεντρώνονται στον εντοπισμό συγκεκριμένων χαρακτηριστικών και αντικειμένων των εικόνων εισόδου.

Ως χαρακτηριστικά (features) θεωρούμε ορισμένα κομμάτια πληροφορίας που στην περίπτωση των εικόνων μπορεί να είναι σημεία, ακμές, αλλαγές στην φωτεινότητα και ούτω καθεξής. Τα χαρακτηριστικά αυτά μπορεί να περιέχουν είτε πληροφορίες χαμηλού επιπέδου, όπως χρώμα, σχήμα και κίνηση, είτε χαρακτηριστικά υψηλού επιπέδου σχετικά με αντικείμενα ή γεγονότα που παρουσιάζονται στην εικόνα. Η περιγραφή και η παραγωγή αυτών των χαρακτηριστικών αποτελεί κινητήρια για πολλούς αλγορίθμους και τεχνικής της *Βαθιάς και Μηχανικής Μάθησης*. Παρακάτω θα δούμε την χρησιμότητα των CNNs (2.3.3) στον τομέα της Όρασης Υπολογιστών και συγκεκριμένα στην επεξεργασία εικόνων.

3.1.1 Συνελικτικά Νευρωνικά Δίκτυα και Επεξεργασία Εικόνων

Οι ψηφιακές εικόνες συχνά αναπαρίστανται σε τρεις διαστάσεις ως H x W x C, όπου H είναι το ύψος της εικόνας, W το βάθος και C τα κανάλια. Η πιο διαδεδομένη μορφή εικόνων είναι του RGB, σύμφωνα με το οποίο κάθε εικόνα έχει τρία κανάλια που ορίζουν το κόκκινο, πράσινο και μπλε κομμάτι του κάθε πίξελ. Όπως καταλαβαίνουμε αυτό σημαίνει ότι αυτές οι αναπαραστάσεις είναι πολύ μεγάλων διαστάσεων, με το πιο συνηθισμένο μέγεθος για είσοδο νευρωνικού δικτύου να είναι 224 x 224. Αν χρησιμοποιούσαμε ένα κλασσικό πλήρως συνδεδεμένο νευρωνικό δίκτυο για την επεξεργασία των εικόνων ο αριθμός των βαρών για το επίπεδο εισόδου θα ήταν 224 * 224 * 3 = 150,528. Εδώ καταλαβαίνουμε πόσο χρήσιμα μπορούν να είναι σε τέτοιες εργασίες με βάση και τις ιδιότητες τους που αναπτύξαμε στο κεφάλαιο 2.3.3.

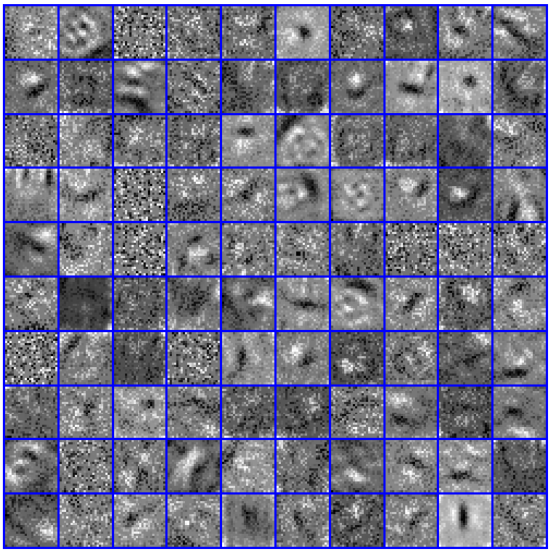
Στα προβλήματα της Όρασης Υπολογιστών έχει βοηθήσει σε πολύ μεγάλο βαθμό η μεταφορά μάθησης 2.4. Η προπόνηση βαθιών συνελικτικών δικτύων εξ αρχής χρειάζεται πολύ μεγάλο χρονικό διάστημα (έως και εβδομάδες) και έχει ανάγκη αυξημένους υπολογιστικού πόρους, συνθήκες που δεν είναι εύκολα εφικτές. Για τον παραπάνω λόγο είναι αρκετά διαδεδομένη η χρήση δικτύων, τα οποία έχουν γίνει pre-trained σε τεράστια dataset. Τα δύο πιο γνωστά pretraining προβλήματα είναι αυτά του *Image Classification* [23] και του *Object Detection* [24]. Το πρώτο πρόβλημα αναζητάει την κατάλληλη ετικέτα για μία εικόνα, π.χ αυτοκίνητο, σκύλος και ούτω καθεξής, ενώ το δεύτερο πρόβλημα στοχεύει στον εντοπισμό αντικειμένων και την θέση τους μέσα σε μία εικόνα.

Τα pretrained CNNs έχουν τις παρακάτω χρήσεις:

- *Fine-tuning*: Σε περιπτώσεις όπου τα δεδομένα του προβλήματος που σκοπεύουμε να λύσουμε είναι αρκετά διαφορετικά από αυτά που χρησιμοποιήθηκαν για την διαδικασία του pretraining, τότε προσπαθούμε να βελτιώσουμε το μοντέλο για το πρόβλημά μας. Για να γίνει αυτό χρησιμοποιούμε τα βάρη του pre-trained μοντέλου ως αρχικά, και στην συνέχεια εφαρμόζουμε πολύ μικρό ρυθμό μάθησης κατά την διάρκεια της εκπαίδευσης

- *Εξαγωγή χαρακτηριστικών*: Εξάγουμε χαρακτηριστικά για την εικόνα παίρνοντας την έξοδο ενός ενδιάμεσου επιπέδου του pre-trained CNN. Με αυτό τον τρόπο παράγεται ένας χάρτης χαρακτηριστικών, δηλαδή ένα σύνολο από χαρακτηριστικά που αντιπροσωπεύουν περιοχές της εικόνας εισόδου. Συνήθως για να πάρουμε μία συνολική αναπαράσταση της εικόνας, απλά αφαιρούμε το επίπεδο εξόδου του CNN, που παράγει τα αποτελέσματα για τις ετικέτες, και χρησιμοποιούμε όλο το υπόλοιπο ως εξαγωγέα χαρακτηριστικών.

Όπως είδαμε στο κεφάλαιο 2.3.3 το κομμάτι εξαγωγής χαρακτηριστικών των CNNs αποτελείται από επίπεδα συνέλιξης, ενεργοποίησης και pooling, τοποθετημένα το ένα πάνω από το άλλο. 'Ετσι, τα CNNs μαθαίνουν τις ιεραρχίες των μοτίβων στα δεδομένα. Οι πυρήνες (kernels) αναγνωρίζουν απλά και βασικά χαρακτηριστικά στα αρχικά επίπεδα, αλλά όσο βαθαίνει το δίκτυο τόσο πιο πολύπλοκα γίνονται τα χαρακτηριστικά.



Εικόνα 3.1: *Απλοϊκά χαρακτηριστικά από πρώτο επίπεδο CNN [1]*

3.2 Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing)

Η Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing - NLP) αποτελεί ένα κλάδο της επιστήμης των υπολογιστών που συνδυάζει στοιχεία της *μηχανικής μάθησης*, αλλά και της *γλωσσολογίας*, και ασχολείται με τις αλληλεπιδράσεις μεταξύ ηλεκτρονικών υπολογιστών και ανθρώπινων (φυσικών) γλωσσών. Αποσκοπεί στο να προσδιορίσει έναν υπολογιστικό μηχανισμό, για να παρουσιάζει διάφορες μορφές γλωσσικής συμπεριφοράς, αλλά και στον σχεδιασμό και υλοποίηση συστημάτων που εξεργάζονται φυσικές γλώσσες για πρακτικές εφαρμογές. Προφανώς κάτι τέτοιο είναι πολύ δύσκολο, λόγω της πολυπλοκότητας και της αφηρημένης φύσης της ανθρώπινης γλώσσας. Μία φράση, για παράδειγμα, μπορεί υπό μία οπτική να δείχνει αίσθημα χαράς, ενώ στην πραγματικότητα μπορεί να περιέχει σαρκασμό, κάτι που αποδεικνύει την δυσκολία του προβλήματος που θέλει να αντιμετωπίσει το NLP. Παρόλα

αυτά τα τελευταία χρόνια έχει γίνει μία τεράστια πρόοδος στα περισσότερα προβλήματα του NLP βασιζόμενη κυρίως στην μέθοδο μάθησης αναπαραστάσεων των λέξεων.

3.2.1 Διανύσματα Λέξεων (Word Embeddings)

Ο ευκολότερος τρόπος να αναπαρασταθεί μία λέξη σαν διάνυσμα θα είναι να κωδικοποιηθεί ως ο δείκτης της στο λεξιλόγιο. Αυτό μπορεί να γίνει έχοντας έναν διάνυσμα διαστάσεων όσο το μέγεθος του λεξιλογίου, και στην θέση που έχει στο λεξιλόγιο η λέξη θα παίρνει την τιμή *1*, ενώ σε όλες τις άλλες θέσεις την τιμή *0*. Η μέθοδος αυτή ονομάζεται one-hot encoding [25], [26]. Η απλοϊκή αυτή τεχνική εμφανίζει αρκετά προβλήματα. Αρχικά, το μέγεθος των διανυσμάτων αυξάνεται γραμμικά με το μέγεθος του λεξιλογίου, κάτι που μπορεί να οδηγήσει σε θέματα μνήμης, αλλά και να συμβάλλει σε υπερεκπαίδευση εξαιτίας του γεγονότος ότι ο χώρος των χαρακτηριστικών είναι εξαιρετικά αραιός. Ένα ακόμα πρόβλημα είναι ότι με αυτή την μέθοδο τα διανύσματα δεν κωδικοποιούν καμία πληροφορία για πιθανή ομοιότητα μεταξύ λέξεων. Κάθε διάνυσμα λέξης είναι ανεξάρτητο με κάθε άλλο και έτσι το σύστημα θεωρεί ότι η λέξη ΄σκύλος΄ είναι το ίδιο ανόμοια με την λέξη ΄γάτα΄ και την λέξη ΄αεροπλάνο΄.

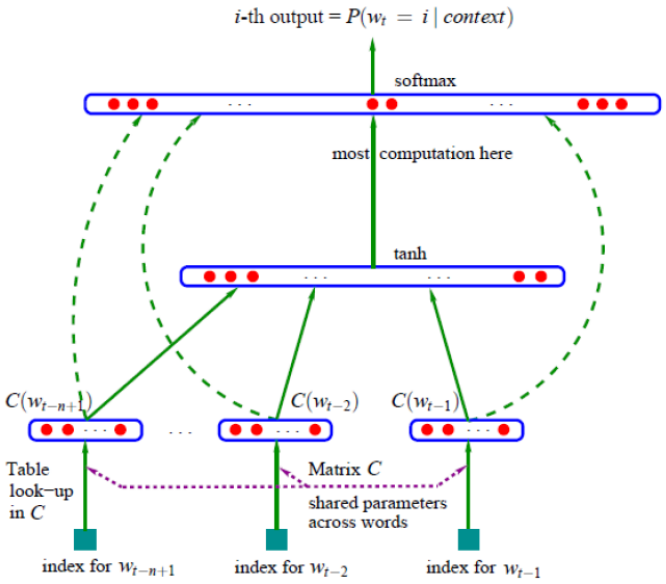
Για να αντιμετωπιστεί το παραπάνω δημιουργήθηκαν τα *Word Embeddings* που χρησιμοποιούνται από τα περισσότερα μοντέλα NLP για την αναπαράσταση λέξεων. Η ιδέα πίσω από τα word embeddings είναι ότι θέλουμε τα διανύσματα παρόμοιων λέξεων να έχουν κοντινές τιμές μεταξύ τους. Τα περισσότερα μοντέλα για την μάθηση word embeddings βασίζονται *στην κατανεμημένη υπόθεση (Distributional Hypothesis)* [27], σύμφωνα με την οποία λέξεις οι οποίες εμφανίζονται με παρόμοια συμφραζόμενα έχουν και παρόμοιο νόημα. Η διαδικασία μάθησης αυτών των διανυσμάτων αποτελεί ένα unsupervised πρόβλημα που αποσκοπεί στην πρόβλεψη μιας λέξης με βάση τα συμφραζόμενα της. Στο παράδειγμα που είδαμε παραπάνω, με την χρήση word embeddings θέλουμε τα διανύσματα του ΄σκύλου΄ και της γάτας να είναι σχετικά κοντά, αφού είναι και τα δύο διάσημα κατοικίδια, αλλά από την άλλη ο ΄σκύλος΄ και το αεροπλάνο΄ θα πρέπει να είναι απομακρυσμένα, αφού δεν έχουν καμία σημασιολογική ομοιότητα μεταξύ τους.

Η πρώτη προσπάθεια για την δημιουργία word embeddings βασίστηκε σε αρχιτεκτονική κλασσικού νευρωνικού δικτύου [2]. Έστω ότι έχουμε προτάσεις μήκους T και λέξεις $w_1, w_2, ..., w_T$ που ανήκουν σε λεξιλόγιο V με μέγεθος $|V|$. Συνήθως χρησιμοποιείται περιεχόμενο μήκους n λέξεων και συσχετίζουμε κάθε λέξη με μία είσοδο v_w με διάσταση d και έξοδο v'_w. Το τελευταίο στάδιο είναι η βελτιστοποίηση μίας συνάρτησης J_θ σε σχέση με τις παραμέτρους θ του μοντέλου και κάποιου σκορ εξόδου $f_\theta(x)$ για κάθε είσοδο x.

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} log f(w_t, w_{t-1}, ..., w_{t-n+1}) \tag{3.1}$$

όπου $f(w_t, w_{t-1}, ..., w_{t-n+1})$ η έξοδος του συστήματος $p(w_t|w_{t-1}, ..., w_{t-n+1})$ όπως υπολογίστηκε από το επίπεδο της softmax με n να συμβολίζει τον αριθμό των λέξεων που δόθηκαν στο μοντέλο.

Τα τελευταία χρόνια το παραπάνω εμπρόσθιο δίκτυο έχει αντικατασταθεί από αναδρομικά νευρωνικά (κεφάλαιο 2.3.4) δίκτυα για την δημιουργία γλωσσικών μοντέλων, με την λογική όμως να παραμένει ίδια.



Εικόνα 3.2: *Νευρωνικό δίκτυο παραγωγής embeddings [2]*

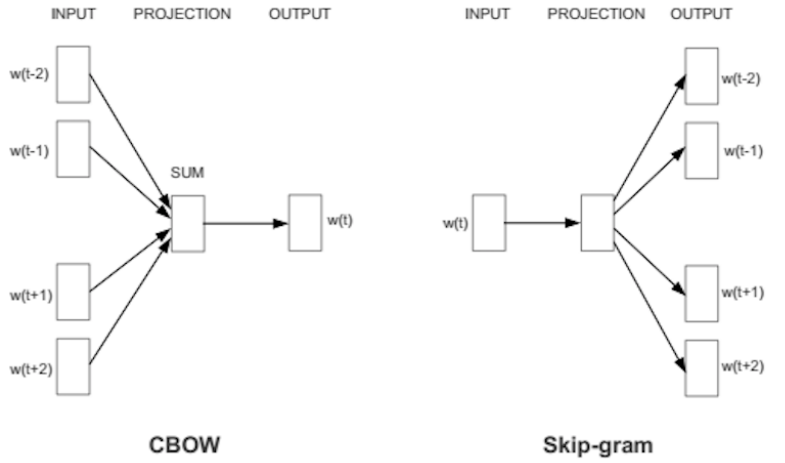
Word2Vec

Το μοντέλο του Word2Vec προτάθηκε το 2013 και από τότε είναι ένα από τα πιο διαδεδομένα μοντέλα για την μάθηση word embeddings [3]. Είναι ένα νευρωνικό δίκτυο 2 επιπέδων που επεξεργάζεται το κείμενο με στόχο την ανακατασκευή του γλωσσολογικού περιεχομένου των λέξεων. Παρόλο που δεν είναι βαθύ νευρωνικό δίκτυο, δέχεται ως είσοδο ένα μεγάλο έγγραφο και παράγει ένα χώρο διανυσμάτων υψηλών διαστάσεων (κοντά στις εκατοντάδες), όπου κάθε μοναδική λέξη του εγγράφου αντιστοιχίζεται σε ένα διάνυσμα στο χώρο αυτό. Τα διανύσματα λέξεων μπορούν έπειτα να χρησιμοποιηθούν από βαθιά νευρωνικά δίκτυα. Το word2vec είναι ένα πολύ φθηνό υπολογιστικά μοντέλο πρόβλεψης για την εκμάθηση διανυσμάτων λέξεων από απλό κείμενο.

Το μοντέλο αυτό έχει δύο εκδόσεις. Το Continuous Bag-of-Words (CBOW) model και το Skip-Gram model. Το CBOW μοντέλο μαθαίνει να προβλέπει την τρέχουσα λέξη με βάση τα συμφραζόμενα, ενώ το Skip-Gram μοντέλο μαθαίνει τις συμφραζόμενες λέξεις από την τρέχουσα λέξη, όπως βλέπουμε και στο σχήμα 3.3.

Τα embeddings που παράγονται από το Word2Vec έχει αποδειχθεί ότι κρατούν σημασιολογικές σχέσεις, κάτι που δίνει δυνατότητα και πράξεων μεταξύ των διανυσμάτων. Για παράδειγμα η διανυσματική πράξη *king - man + woman* θα μας οδηγήσει σε ένα διάνυσμα κοντά στο διάνυσμα της λέξης *queen*. Συνήθως τα word embeddings προπονούνται σε μεγάλες, μη annotated συλλογές κειμένων και δίνουν τελικώς σημασιολογικά σωστές και πυκνές αναπαραστάσεις των λέξεων. Για αυτό τον λόγο είναι πολύ συνηθισμένο να χρησιμοποιούνται pre-trained word embeddings για τα περισσότερα NLP προβλήματα, και αν όχι αυτούσια, τότε για την αρχικοποίηση του επιπέδου εισόδου.

Και αυτή η μέθοδος όμως παρουσιάζει κάποια προβλήματα:



Εικόνα 3.3: *Μοντέλο Word2Vec [3]*

- *Απουσία περιεχομένου (Lack of context)*: Ένα προφανές πρόβλημα που μπορεί να δημιουργηθεί χρησιμοποιώντας τεχνικές όπως του Word2Vec είναι ότι λέξεις που έχουν πολλές έννοιες (πολύσημες) δεν μπορούν να αναπαρασταθούν σωστά στον διανυσματικό χώρο.

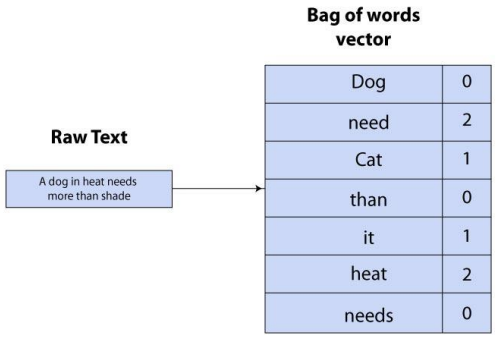
- *Παράθυρο γειτονικών λέξεων (Window of surrounding words)*: Ένας ακόμα περιορισμός προκύπτει όταν μαθαίνουμε διανύσματα λέξεων βάσει ενός μικρού παράθυρου από γειτονικές λέξεις. Σε τέτοιες περιπτώσεις παρατηρούμε ομαδοποίηση λέξεων που έχουν αντίθετο σημασιολογικό νόημα, αλλά έχουν κοινά συμφραζόμενα, όπως για παράδειγμα 'good', 'bad'.

Αξίζει, επίσης, να σημειωθεί ότι η διαφορά ανάμεσα στα κλασσικά word embeddings και του Word2Vec είναι ότι τα embeddings του Word2Vec κωδικοποιούν σημασιολογικές σχέσεις που είναι χρήσιμες για μικρότερα προβλήματα. Δεν θα ήταν όμως χρήσιμα για προβλήματα που δεν έχουν συνάφεια με σημασιολογία. Από την άλλη, τα τυπικά νευρωνικά δίκτυα παράγουν διανύσματα, τα οποία έχουν άμεση συσχέτιση με το πρόβλημα που λύνουν.

3.2.2 Αναπαράσταση Προτάσεων (Sentence Representation)

Αφού παραπάνω μελετήσαμε το πως μπορεί να αναπαρασταθεί μία λέξη, το επόμενο βήμα είναι η αναπαράσταση μίας πρότασης. Ο πιο απλός αλγόριθμος για τον υπολογισμό της αναπαράστασης μία πρότασης είναι το μοντέλο Bag of Words (BoW). Στο BoW, μία πρόταση αναπαρίσταται ως συσσωμάτωση (συνήθως του μέσου όρου) των λέξεων της, αγνοώντας πλήρως την σειρά τους μέσα στην πρόταση. Ο CBOW που είδαμε παραπάνω αποτελεί μία παραλλαγή του BoW, στην οποία οι λέξεις αναπαρίστανται από τα embeddings τους. Παρόλο που είναι απλοϊκός αλγόριθμος ο CBOW αποτελεί δυνατό θεμέλιο για πολλά προβλήματα κατηγοριοποίησης κειμένου.

Μετά την επιτυχία των μοντέλων διανυσμάτων λέξεων, έγιναν προσπάθειες να κωδικοποιηθούν και ολόκληρες προτάσεις σε διανύσματα λίγων διαστάσεων και συγκεκριμένου μεγέθους. Μία προσπάθεια σε αυτή την κατεύθυνση ήταν οι Skip-Thought vectors [28]. Η λογική που



Εικόνα 3.4: *Παράδειγμα BoW*

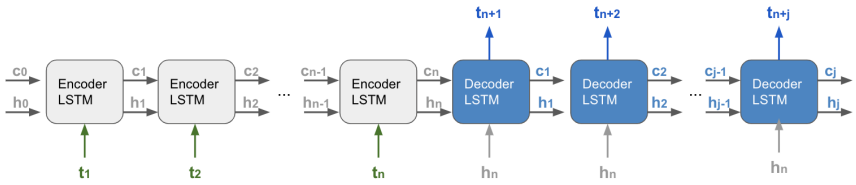
ακολουθούσε είναι ανάλογη με το Skip-gram Word2Vec μοντέλο, με την διαφορά αντί να προβλέπει συμφραζόμενες λέξεις, το Skip-Thought μοντέλο προβλέπει τις προτάσεις που περικυκλώνουν την τρέχουσα πρόταση.

Παρόλα αυτά, η συνηθισμένη προσέγγιση για τα NLP προβλήματα είναι να προπονούνται RNN δίκτυα end-to-end. Υποθέτοντας ότι έχουμε μία πρόταση με λέξεις $w_1, w_2, ..., w_T$ που θέλουμε να κατηγοριοποιήσουμε. Στην χρονική στιγμή t η λέξη εισόδου w_t περνάει από το embedding επίπεδο και το παραγόμενο διάνυσμα e_t δίνεται στο RNN. Η συνολική πρόταση αναπαρίσταται ως η κρυμμένη κατάσταση του RNN στο τελευταίο χρονικά βήμα της ακολουθίας. Τελικά η πρόταση περνάει σε έναν κατηγοριοποιητή (classifier), ο οποίος συνήθως είναι μία σειρά από πλήρως συνδεδεμένα επίπεδα. Πολλές τεχνικές μπορούν να ακολουθηθούν για την βελτίωση του αποτελέσματος, όπως η χρήση ενός επιπέδου *attention* (κεφάλαιο 2.3.5) πάνω από το RNN.

3.3 Μοντέλα Sequence-to-Sequence

Η ανάγκη αντιστοίχησης ακολουθιακών δεδομένων οδήγησε στην δημιουργία των μοντέλων Sequence to Sequence (seq2seq) [29]. Η μάθηση της μορφής Sequence to Sequence αποτελεί μία δομή για την αντιστοίχηση ακολουθιών εισόδου σε ακολουθίες εξόδου διαφορετικού μήκους. Η αντιστοίχηση ακολουθιακών δεδομένων εμφανίζεται σε διάφορους τομείς όπως η *μηχανική μετάφραση (machine translation)* [30] και οι *οπτικές ερωτοαπαντήσεις (visual question answering)* [31]. Το κοινό στοιχείο αυτών των προβλημάτων είναι ότι μία ακολουθία εισόδου αντιστοιχίζεται σε ακολουθία εξόδου με διαφορετικό μήκος και ίσως διαφορετικού τύπου δεδομένα. Για παράδειγμα, η αγγλική πρόταση 'I am dancing in the snow' μεταφράζεται στα ελληνικά ως 'Χορεύω στο χιόνι'. Παρατηρούμε ότι η ακολουθία εισόδου έχει 6 λέξεις, ενώ η αντίστοιχη εξόδου έχει μόνο 3. Επομένως είναι αδύνατο να χρησιμοποιήσουμε ένα απλό RNN για να μεταφράσουμε κάθε λέξη από τα Αγγλικά στα Ελληνικά.

Η γενική προσέγγιση των seq2seq μοντέλων είναι απλή. Το δίκτυο έχει δύο κομμάτια, έναν κωδικοποιητή (encoder) και έναν αποκωδικοποιητή (decoder). Ο κωδικοποιητής αντιστοιχίζει την ακολουθία εισόδου σε ένα διάνυσμα συγκεκριμένων διαστάσεων, έστω c (context vector), το οποίο με την σειρά του χρησιμοποιείται από τον αποκωδικοποιητή για την παραγωγή της



Εικόνα 3.5: *Αρχιτεκτονική Sequence-to-Sequence μοντέλου*

εξόδου. Έστω λοιπόν ότι έχουμε την ακολουθία εισόδου $x_1, x_2, ..., x_T$ και θέλουμε την έξοδο $y_1, y_2, ..., y_T$. Αυτή θα δίνεται ως εξής:

$$P(y_1, ..., y_T | x_1, ..., x_t) = \Pi_{t=1}^{T} P(y_t | c, y_1, ..., y_{t-1}) \tag{3.2}$$

Αφού και ο κωδικοποιητής και ο αποκωδικοποιητής έχουν να κάνουν με ακολουθίες, συνήθως κατασκευάζονται χρησιμοποιώντας RNNs (όχι όμως πάντα). Με μεγαλύτερη λεπτομέρεια:

- ***Κωδικοποιητής (Encoder)***: Η ακολουθία εισόδου $x_1, x_2, ..., x_T$ επεξεργάζεται από τον RNN κωδικοποιητή. Η τελευταία κρυφή κατάσταση αποτελεί την αναπαράσταση της εισόδου και χρησιμοποιείται για την αρχικοποίηση της κρυφής κατάστασης του RNN αποκωδικοποιητή. Ο ρόλος αυτού του διανύσματος είναι να περιγράφει την πληροφορία της εισόδου, με βάση την οποία ο αποκωδικοποιητής θα παράξει την έξοδο.

- ***Αποκωδικοποιητής (Decoder)***: Αφού αρχικοποιηθεί η κρυφή κατάσταση του αποκωδικοποιητή, ο αποκωδικοποιητής δέχεται σαν είσοδο ένα token που συμβολίζει την αρχή της ακολουθίας (συνήθως <START>, <SOS>), και αρχίζει να παράγει λέξεις μία προς μία. Κάθε χρονική στιγμή t, η επόμενη λέξη y_t παράγεται με βάση την τελευταία λέξη που έχει παραχθεί και την κρυφή κατάσταση του αποκωδικοποιητή. Η πρόβλεψη λέξεων συνεχίζεται μέχρι να προβλεφθεί token που σημαίνει την λήξη της ακολουθίας (συνήθως <END>, <EOS>).

Ένα πρόβλημα αυτής της τεχνικής είναι ότι η πληροφορία της εισόδου είναι προσβάσιμη μόνο από το διάνυσμα περιεχομένου. Προσπαθώντας να κωδικοποιήσουμε όλη την πληροφορία σε ένα διάνυσμα συγκεκριμένου μεγέθους μπορεί να οδηγήσει σε περιορισμό της πληροφορίας. Μία λύση σε αυτό το πρόβλημα είναι η εισαγωγή ένος attention μηχανισμού, ο οποίος συνδυάζει δυναμικά τις εξόδους του κωδικοποιητή για κάθε λέξη εισόδου κατά την διάρκεια της αποκωδικοποίησης. Αυτό σημαίνει ότι ο αποκωδικοποιητής έχει την δυνατότητα να συγκεντρωθεί στα πιο σχετικά κομμάτια της εισόδου.

3.3.1 Μέθοδος Teacher Forcing

Κατά την διάρκεια εκπαίδευσης ενός seq2seq μοντέλου, ο decoder προβλέπει την επόμενη λέξη από την πρόβλεψη του προηγούμενου βήματος και την κρυφή κατάσταση που ενσωματώνει συνολικά την πληροφορία της ακολουθίας λέξεων. Πολλές φορές ο αποκωδικοποιητής δεν προβλέπει την σωστή λέξη, ένα σημαντικό πρόβλημα καθώς το δίκτυο εκπαιδεύεται τελικά σε λανθασμένες ακολουθίες. Για να αποφευχθεί αυτό το πρόβλημα ξεκίνησε να χρησιμοποιείται

η διαδικασία του *teacher forcing* [32]. Στο teacher forcing, αντί ο αποκωδικοποιητής να λαμβάνει ως είσοδο την λέξη από την πρόβλεψη του προηγούμενου βήματος, λαμβάνει ως είσοδο την επιθυμητή λέξη, δηλαδή αυτή που θα έπρεπε να είχε προβλέψει. Η μέθοδος αυτή διδάσκει (teacher) τον αποκωδικοποιητή την σωστή πρόβλεψη, εξαναγκάζοντας (forcing) την σωστή είσοδο σε κάθε επόμενο βήμα, ώστε να προκύπτει η σωστή ακολουθία. Η εκπαίδευση με teacher forcing κάνει το μοντέλο να συγκλίνει ταχύτερα.

Το πρόβλημα που δημιουργείται μέσω αυτού του αλγορίθμου είναι στην αξιολόγηση. Κατά την αξιολόγηση του μοντέλου, δεν υπάρχει διαθέσιμη κάποια επιθυμητή ακολουθία, οπότε ο αποκωδικοποιητής είναι αναγκασμένος να χρησιμοποιήσει την δική του πρόβλεψη ως είσοδο για την επόμενη. Αυτή η ασυμφωνία ανάμεσα στην εκπαίδευση και την αξιολόγηση μπορεί να οδήγησει σε αστάθεια του μοντέλου. Για να αντιμετωπιστεί αυτό το πρόβλημα μπορεί να χρησιμοποιείται το teacher forcing στα αρχικά στάδια της εκπαίδευσης, ώστε να δοθεί μια αρχική βοήθεια στο μοντέλο και στην συνέχεια σταδιακά να μειώνεται το ποσοστό χρήσης του, με σκοπό να καλυφθεί σιγά σιγά η ασυμφωνία των δύο τρόπων.

Κεφάλαιο 4

Image Captioning στο MS COCO dataset

Στο κεφάλαιο αυτό παρουσιάζεται το σύνολο των δεδομένων που χρησιμοποιήθηκαν για την εκπαίδευση και την αξιολόγηση του μοντέλου μας. Στην συνέχεια αναπτύσσεται ο σχεδιασμός του μοντέλου που συνδυάζει διάφορες αρχιτεκτονικές που είδαμε στα προηγούμενα κεφάλαια.

4.1 Δεδομένα - MS COCO dataset

Η μηχανική μάθηση έχει κάνει μεγάλα άλματα στον ερευνητικό τομέα της παραγωγής λεκτικής περιγραφής εικόνας (Image Captioning). Ωστόσο η παραγωγή των βέλτιστων αποτελεσμάτων εμποδίζεται από το μικρό πλήθος δεδομένων που είναι διαθέσιμα. Ανάμεσα στα διάφορα σύνολα δεδομένων που έχουν αναπτυχθεί για το συγκεκριμένο πρόβλημα, για την ανάπτυξη της παρούσας διπλωματικής εργασίας επιλέχτηκε το *MS COCO: Microsoft Common Objects in Context* [33]. Το dataset αυτό δημιουργήθηκε το 2014 με σκοπό την βελτίωση των state-of-the-art αλγορίθμων στο πρόβλημα του Object Recognition [24]. Η τακτική που ακολουθήθηκε για την συλλογή των δεδομένων ήταν η επιλογή εικόνων που περιέχουν πολύπλοκες καθημερινές σκηνές, αλλά και συνηθισμένα αντικείμενα στο φυσικό τους περιεχόμενο. Με τον καιρό, το dataset επεκτάθηκε και χρησιμοποιήθηκε και σε περισσότερα προβλήματα, όπως το keypoint detection [34] και το *image captioning*, με το οποίο θα ασχοληθούμε.

Το MS COCO αποτελεί μία τεράστια συλλογή δεδομένων, τα οποία συνδυάζουν μεγάλο αριθμό εικόνων και έναν αριθμό από λεζάντες για τις εικόνες, δηλαδή ακριβώς αυτό που χρειαζόμαστε για το training του μοντέλου μας. Συγκεκριμένα, το dataset έχει ήδη χωριστεί σε train, validation, testing sets, όπως βλέπουμε στον πίνακα 4.1

Εν συντομία, θα αναφερθούν και μερικά από τα άλλα διαθέσιμα dataset για το συγκεκριμένο πρόβλημα.

- *Flickr30k Dataset* [35] : Ένα επίσης αρκετά διαδεδομένο dataset για image captioning. Αποτελείται από 30,000 εικόνες που έχουν συλλεχθεί από το Flickr μαζί με 158,000 λεζάντες φτιαγμένες από ανθρώπους. Έχει χρησιμοποιηθεί και για προβλήματα object detection και classification.

- *Flickr8k Dataset* [36]: Από τα πρώτα dataset που είχαν σαν άμεσο πρόβλημα προς αντιμετώπιση το image captioning. Αποτελείται από 8,000 εικόνες, με 6,000 για training,

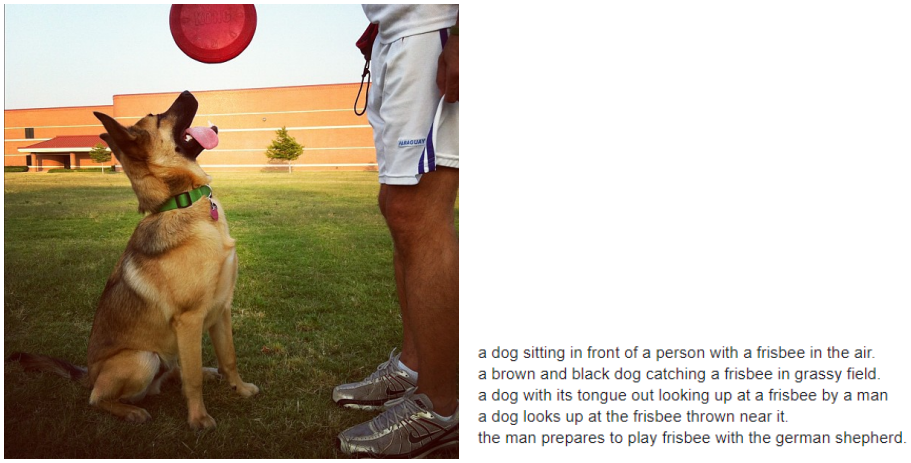
Πίνακας 4.1: *Διαχωρισμός δεδομένων στο MS COCO dataset*

Set	Images	Captions
Training	82,783	413,915
Evaluation	40,504	202,250
Testing	40,775	179,189

1,000 για validation και 1,000 για testing. Κάθε εικόνα έχει 5 λεζάντες από ανθρώπους.

- *Visual Genome Dataset* [37]: Σε αντίθεση με τα άλλα τρία dataset που έχουμε δει που κάθε εικόνα έχει μία λεζάντα που περιγράφει όλη την σκηνή, το Genome έχει ξεχωριστές λεζάντες για διαφορετικές περιοχές της εικόνας. Συνολικά περιέχει πάνω από 108,000 εικόνες, αλλά δεν έχει χρησιμοποιηθεί με μεγάλη επιτυχία προς το παρόν στον τομέα του image captioning

Λόγω του μεγαλύτερου πλήθους δεδομένων, αλλά και της έρευνας που έχει γίνει διαλέγουμε το MS COCO dataset.



Εικόνα 4.1: *Παράδειγμα εικόνας με λεζάντες από MS COCO*

4.1.1 Επεξεργασία Δεδομένων

Το MS COCO παρέχει τα δεδομένα των captions σε αρχεία τύπου ΄.json΄. Το πρώτο βήμα είναι να συγκεντρώσουμε όλα τα δοσμένα captions και να τα περάσουμε από μία μορφή προεπεξεργασίας. Θα γίνει μία διαδικασία tokenization ώστε να απομακρύνουμε περιττούς χαρακτήρες, όπως σημεία στίξης και διάφορα άλλα σύμβολα, π.χ. θαυμαστικά, διέσεις ή οτιδήποτε άλλο δεν συμβάλει στην σημασία της πρότασης. Επίσης γίνεται μία αντιστοίχηση μεταξύ φράσεων όπως *'it's', 'we've'* με τις συστατικές τους λέξεις, δηλαδή *'it is', 'we have'* αντίστοιχα. Προσθέτουμε σε κάθε caption ειδικά tokens για να γνωρίζει το μοντέλο την αρχή και το τέλος της ακολουθίας. Συγκεκριμένα τοποθετούμε token '<start>' στην αρχή της πρότασης και '<end>' στο τέλος της. Η εκπαίδευση του μοντέλου μας απαιτεί το μέγεθος των δεδομένων εισόδου (εν προκειμένω οι λεζάντες) να είναι σταθερού μήκους, όμως κάθε

πρόταση προφανώς και δεν έχει το ίδιο μήκος. Για να αντιμετωπίσουμε αυτό το πρόβλημα, ορίζουμε το token '<pad>', το οποίο τοποθετείται μετά το '<end>' στις προτάσεις που έχουν μήκος μικρότερο από ένα μέγιστο που έχουμε ορίσει, μετατρέποντας όλες τις ακολουθίες σε ακολουθίες ίσου μήκους.

4.1.2 Λεξιλόγιο (Vocabulary)

Το λεξιλόγιο αποτελεί μία μορφή λεξικού (dictionary) το οποίο αντιστοιχίζει έναν δείκτη i με μία λέξη w. Όπως θα δούμε και παρακάτω, το μοντέλο θα προβλέπει πιθανότητες πάνω στους δείκτες του λεξιλογίου. Αυτό σημαίνει ότι ολόκληρο το λεξιλόγιο θα πρέπει να παραμένει σταθερό κατά την διάρκεια τόσο της εκπαίδευσης, αλλά και της μετέπειτα χρήσης του μοντέλου. Για την δημιουργία του λεξιλογίου, ελέγχουμε όλες τις υπάρχουσες λεζάντες και καταχωρούμε στο dictionary τις λέξεις με βάση την συχνότητα εμφάνισης τους. Επειδή το λεξιλόγιο θα είναι συγκεκριμένου μεγέθους, πιθανό είναι να εμφανιστούν λέξεις οι οποίες θα είναι άγνωστες και μη καταχωρημένες στο λεξιλόγιο. Όλες αυτές οι λέξεις θα αντιστοιχίζονται σε ένα ακόμα special token, το '<unk>' που θα έχει δικιά του καταχώρηση στο λεξιλόγιο. Σημαντικό επίσης είναι να αναφερθεί, ότι η δημιουργία του λεξικού βασίζεται μόνο στα captions του training set και δεν λαμβάνονται καθόλου υπόψιν αυτές του validation και test set.

Στις εικόνες 4.2 και 4.3 παρατηρούμε τις πιο συχνές και τις λιγότερο συχνές από τις top 5000 λέξεις στο dataset. Όπως είναι φυσικό οι πιο συνηθισμένες λέξεις είναι άρθρα, 'a', 'an', 'the' και βασικά ρήματα και ουσιαστικά, όπως 'is', 'are', 'man', 'people'. Από την άλλη, λέξεις όπως 'pesto', 'injured', 'cracker' δεν αποτελούν βασικό κομμάτι των captions και χρησιμοποιούνται σε πολύ συγκεκριμένα σενάρια.

4.2 Αρχιτεκτονική Μοντέλου

Αφού πλέον έχουμε περιγράψει την απαραίτητη θεωρία, αλλά και τα δεδομένα που θα χρησιμοποιήσουμε για την εκπαίδευση του δικτύου μας, θα προχωρήσουμε στην περιγραφή της βασικής αρχιτεκτονικής (εικόνα 4.4). Το βασικό μας μοντέλο αποτελείται από δύο βασικά μέρη, τον *κωδικοποιητή εικόνων (image encoder)* και τον *αποκωδικοποιητή* που παράγει τις λεζάντες *(decoder)*. Ο κωδικοποιητής είναι ένα δίκτυο που έχει σκοπό να κωδικοποιεί τις εικόνες εισόδου. Δέχεται, δηλαδή, σε κάθε χρήση του μοντέλου την εικόνα και προσπαθεί να εξάγει από αυτήν τα χρήσιμα χαρακτηριστικά της, τα οποία είναι απαραίτητα για την μετέπειτα παραγωγή της λεκτικής περιγραφής της. Το τελικό του αποτέλεσμα θα είναι το διάνυσμα χαρακτηριστικών της εικόνας, το οποίο εμπεριέχει όλη την πληροφορία της δοσμένης εικόνας και τροφοδοτεί τον αποκωδικοποιητή. Ο αποκωδικοποιητής, με την σειρά του, δέχεται σε κάθε βήμα (timestep) τα χαρακτηριστικά της εικόνας και την κρυφή κατάσταση από τις λέξεις που έχουν παραχθεί ήδη. Εκείνος παράγει την τρέχουσα λέξη και την κρυφή κατάσταση που θα χρησιμοποιήσει στο επόμενο βήμα, φτιάχνοντας τελικά επαναληπτικά ολόκληρη την λεζάντα.

Γρήγορα όμως, έγινε κατανοητό ότι αυτή η αρχιτεκτονική δεν θα είναι αρκετή για να πετύχουμε βέλτιστα αποτελέσματα. Το πρόβλημα που γεννάται είναι, πως κάθε φορά που το μοντέλο προσπαθεί να παράξει μία καινούρια λέξη της λεζάντας, αυτή η λέξη συνήθως περι-

frequency	index	word
0.000000	1	a
-0.404588	2	.
-1.515484	3	on
-1.573031	4	of
-1.612007	5	the
-1.674996	6	in
-1.862457	7	with
-1.963443	8	and
-2.300812	9	is
-2.587460	10	man
-2.669145	11	to
-2.918806	12	sitting
-2.977400	13	an
-3.002140	14	two
-3.123198	15	standing
-3.134372	16	at
-3.136424	17	people
-3.166124	18	are
-3.204361	19	,

Εικόνα 4.2: *Συχνότερες λέξεις στα captions*

γράφει μόνο ένα κομμάτι της εικόνας και δεν είναι δυνατόν να πιάσει την ουσία ολόκληρης της εικόνας εισόδου. Επομένως, ταΐζοντας τον αποκωδικοποιητή μας σε κάθε βήμα με την συνολική αναπαράσταση της εικόνας, τον εμποδίζουμε από το να μπορεί να παράξει αποτελεσματικά διαφορετικές λέξεις για διαφορετικά κομμάτια της εικόνας. Για αυτό τον λόγο προσθέτουμε και ένα τρίτο βασικό μέρος στην αρχιτεκτονική μας, αυτό του *Attention*. Μέσω του μηχανισμού του attention θα καθοδηγείται ο αποκωδικοποιητής, με τέτοιο τρόπο ώστε να δίνει σημασία στο πιο σχετικό κομμάτι της εικόνας, κάθε φορά που θα παράγεται μία νέα λέξη (εικόνα 4.5).

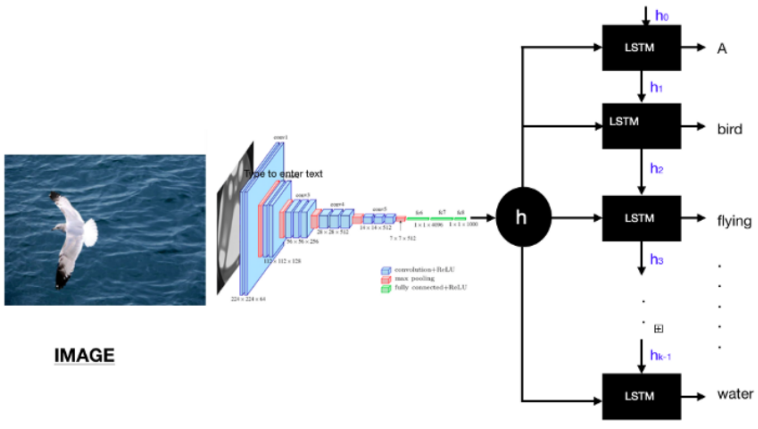
Προχωρώντας θα δούμε με μεγαλύτερη λεπτομέρεια το κάθε μέρος.

4.2.1 Κωδικοποιητής Εικόνας (Image Encoder)

Το μοντέλο μας παίρνει μία εικόνα και παράγει μία λεζάντα y κωδικοποιημένη ως μία ακολουθία από κωδικοποιημένες λέξεις

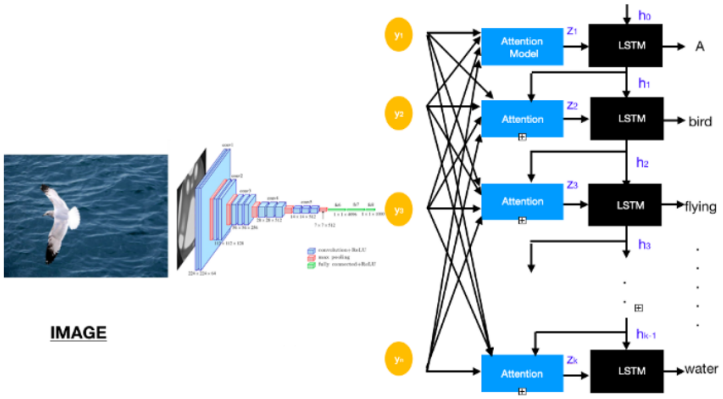
frequency	index	word
-10.646341	4980	grows
-10.646341	4981	wilted
-10.646341	4982	designated
-10.646341	4983	unopened
-10.646341	4984	whites
-10.646341	4985	lions
-10.646341	4986	fellow
-10.646341	4987	goblets
-10.646341	4988	bathed
-10.646341	4989	quilted
-10.646341	4990	injured
-10.646341	4991	scaffolding
-10.646341	4992	cracker
-10.646341	4993	coastal
-10.646341	4994	pesto
-10.646341	4995	magnifying
-10.646341	4996	unpacked
-10.646341	4997	rocket
-10.646341	4998	shipping

Εικόνα 4.3: *Λιγότερο συχνές λέξεις στα captions*



Εικόνα 4.4: *Αρχική αρχιτεκτονική μοντέλου Image Captioning*

$$y = \{y_1, ..., y_C\}, \quad y_i \in R^K \tag{4.1}$$



Εικόνα 4.5: *Αρχιτεκτονική μοντέλου Image Captioning με Attention*

όπου K είναι το μέγεθος του λεξιλογίου και C είναι το μήκος της λεζάντας.

Για την επεξεργασία της εικόνας εισόδου η οποία είναι σε μορφή $H\ x\ W\ x\ C$, όπου H είναι το ύψος της εικόνας, W το βάθος και C τα κανάλια θα χρησιμοποιήσουμε ένα Convolutional Neural Network (2.3.3). Το CNN χρησιμοποιείται με σκοπό να παραχθεί ένα σύνολο από διανύσματα χαρακτηριστικών (annotation vectors), το καθένα από τα οποία θα είναι μία D-διάστατη αναπαράσταση που θα αντιστοιχεί σε κάποια περιοχή της εικόνας.

$$a = \{a_1, ..., a_L\}, \quad a_i \in R^D \tag{4.2}$$

Προκειμένου να έχουμε μία αντιστοιχία μεταξύ των διανυσμάτων χαρακτηριστικών και των κομματιών της εικόνας, τα χαρακτηριστικά θα τα εξάγουμε από ένα χαμηλότερο επίπεδο συνέλιξης και όχι χρησιμοποιώντας ένα πλήρως συνδεδεμένο επίπεδο, που είναι η συνηθισμένη τακτική. Αυτό θα επιτρέπει στον αποκωδικοποιητή να δώσει μεγαλύτερη σημασία σε συγκεκριμένα κομμάτια της εικόνας επιλέγοντας ένα υποσύνολο όλων των διανυσμάτων χαρακτηριστικών.

4.2.2 Αποκωδικοποιητής και μηχανισμός Προσοχής (Decoder and Attention Mechanism)

Για την διαδικασία της αποκωδικοποίησης και τελικής παραγωγής της λεζάντας θα χρησιμοποιήσουμε ένα Recurrent Neural Network (2.3.4). Συγκεκριμένα, θα χρησιμοποιηθεί ένα Long Short-Term Memory (LSTM) δίκτυο που θα παράγει το caption, παράγοντας μία λέξη ανά χρονικό βήμα, εξαρτώμενο από ένα διάνυσμα συμφραζομένων (context vector), την προηγούμενη κρυφή κατάσταση και τις λέξεις που έχουν παραχθεί προηγουμένως.

Με απλούς όρους, ο context vector $\hat{z}_t$ αποτελεί μία δυναμική αναπαράσταση των σχετικών κομματιών της εικόνας εισόδου για μία χρονική στιγμή t. Ορίζουμε έναν μηχανισμό, ο οποίος υπολογίζει το $\hat{z}_t$ από τους annotation vectors $a_i, i = 1, .., L$, που αντιστοιχούν στα χαρακτηριστικά που έχουν εξαχθεί από διαφορετικές περιοχές της εικόνας. Για κάθε περιοχή i, ο μηχανισμός παράγει ένα θετικό βάρος α_i, το οποίο αποτελεί την πιθανότητα η περιοχή i να είναι και το σωστό μέρος για να ασχοληθεί ο αποκωδικοποιητής για την παραγωγή της επόμενης λέξης. Αυτό το βάρος α_i για κάθε annotation vector παράγεται μέσω ενός at-

tention μοντέλου f_{att}, για το οποίο χρησιμοποιείται ένα απλό feed forward neural network εξαρτώμενο από την προηγούμενη κρυφή κατάσταση h_{t-1}. Τελικά έχουμε

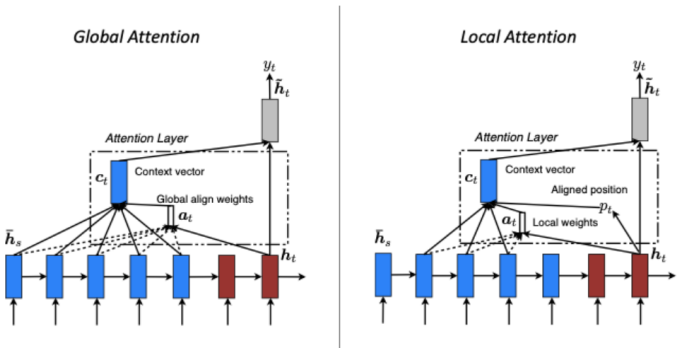
$$e_{ti} = f_{att}(a_i, h_{t-1}) \tag{4.3}$$

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})} \tag{4.4}$$

Αφού έχουν υπολογιστεί όλα τα βάρη, ο context vector υπολογίζεται ως συνάρτηση των α_i, a_i.

$$\hat{z}_t = \sum_{i=1}^{L} \alpha_{ti} a_i \tag{4.5}$$

Το συγκεκριμένο είδος attention ονομάζεται *soft* [38] και χαρακτηρίζεται από το γεγονός ότι ο context vector υπολογίζεται ως το weighted άθροισμα των κρυφών καταστάσεων του encoder (Στην περίπτωση μας, κρυφές καταστάσεις του encoder θεωρούμε τους annotation vectors που παράγουμε από το cnn). Στο μοντέλο μας, δοκιμάσαμε δύο διαφορετικά είδη του soft attention [39]. Πρώτα έχουμε το *Global Attention*, στο οποίο λαμβάνονται υπόψιν όλες οι κρυφές καταστάσεις του encoder. Αυτή η τακτική οδηγεί στο μειονέκτημα ότι είναι υπολογιστικά πολύ ακριβό όταν χρησιμοποιείται για πολλές προτάσεις. Για αυτό δοκιμάζεται και το *Local Attention* το οποίο χρησιμοποιεί ένα υποσύνολο των κρυφών καταστάσεων του κωδικοποιητή, μέσω ενός παραθύρου (εικόνα 4.6).



Εικόνα 4.6: *Αρχιτεκτονικές Global και Local Attention*

Τελικά, ο context vector και η κρυφή κατάσταση του decoder περνάνε διαδικασία συνένωσης (concatenation), για να παραχθεί η νέα έξοδος. Η παραπάνω διαδικασία επαναλαμβάνεται έως ότου να παραχθεί το token λήξης της πρότασης '<end>' ή αν η ακολουθία ξεπεράσει το μέγιστο όριο μεγέθους.

Κεφάλαιο 5

Πειραματική Διαδικασία, Αξιολόγηση και Αποτελέσματα

Στο κεφάλαιο αυτό θα ασχοληθούμε με την πειραματική ανάλυση του μοντέλου μας. Αρχικά θα γίνει μία μικρή αναφορά στις τεχνολογίες και τα framework που χρησιμοποιήθηκαν. Έπειτα, θα δούμε τα αποτελέσματα σε σχέση με διάφορες μετρικές αξιολόγησης και σε σύγκριση με state-of-the-art μοντέλα. Στην συνέχεια, θα δώσουμε κάποιες πληροφορίες για την διαδικασία της εκπαίδευσης και τις διάφορες υπερπαραμέτρους και διαφοροποιήσεις του μοντέλου μας. Τέλος, θα γίνει η οπτικοποίηση ορισμένων παραδειγμάτων χρήσης του μοντέλου μας.

5.1 Τεχνικό Υπόβαθρο Ανάπτυξης Νευρωνικών Δικτύων

Για την υλοποίηση του νευρωνικού δικτύου μας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python [40], η γλώσσα που χρησιμοποιείται κατά κόρων για διαδικασίες τεχνικές μάθησης. Παρακάτω θα αναλύσουμε σύντομα τα εργαλεία και τα frameworks που χρησιμοποιήθηκαν για την ανάπτυξη του μοντέλου μας.

5.1.1 Tensorflow

Το *Tensorflow* είναι μία δωρεάν και open-source βιβλιοθήκη λογισμικού για μηχανική μάθηση. Μπορεί να χρησιμοποιηθεί για μία πληθώρα προβλημάτων, αλλά δίνει ιδιαίτερη βάση στην προπόνηση και χρήση βαθιών νευρωνικών δικτύων. Φτιάχτηκε από την Google και η πρώτη του version έγινε release τον Φεβρουάριο του 2017. Είναι μία ευέλικτη αρχιτεκτονική που επιτρέπει εύκολο deployment σε μία μεγάλη γκάμα από πλατφόρμες (CPUs, GPUs, TPUs) και από υπολογιστές σε cluster από servers και σε κινητές συσκευές.

5.1.2 Keras

Το *Keras* είναι μία δωρεάν βιβλιοθήκη-API υψηλού επιπέδου που ξεκίνησε να αναπτύσσεται το 2015, γραμμένο σε γλώσσα προγραμματισμού Python. Το όνομά του προέρχεται από την ελληνική λέξη Κέρας και ο κύριος δημιουργός του είναι ο Francois Chollet [41], μηχανικός της Google. Το Keras μπορεί να τρέξει πάνω στο tensorflow και είναι παραμετροποιημένο ώστε να μπορεί να λειτουργήσει και σε CPU, αλλά και σε GPU. Η σχεδίαση του έχει γίνει με τέτοιο τρόπο, ώστε να είναι φιλικό προς το χρήστη στους πειραματισμούς του με Βαθιά Νευρωνικά Δίκτυα, να είναι εύκολα παραμετροποιήσιμο και να μπορεί να διευρυνθεί για άλλους σκοπούς.

5.2 Συγγενείς Εργασίες

Το πρόβλημα του Image Captioning είναι ένα πρόβλημα που απασχολεί την ερευνητική κοινότητα για πολλά χρόνια. Από τις πρώτες προσπάθειες που έγιναν, ήταν άμεσα αντιληπτό ότι απλές τεχνικές μηχανικής μάθησης δεν θα είναι αρκετές για την αντιμετώπιση ενός εξαιρετικά σύνθετου task. Η χρήση βαθιών νευρωνικών δικτύων ήταν μονόδρομος.

Η αρχιτεκτονική ενός sequence-to-sequence μοντέλου (κεφάλαιο 3.3) με έναν κωδικοποιητή και έναν αποκωδικοποιητή εκτόξευσε την επίδοση στο πρόβλημα του Image Captioning. Οι πρώτες πραγματικά καλές προσπάθειες άρχισαν χρησιμοποιώντας διαφορετικά RNN για το κομμάτι του αποκωδικοποιητή [42], [43]. Προχωρώντας η χρήση των LSTM αντί για vanilla RNNs συνέβαλε στην βελτίωση της απόδοσης [44]. Στην συνέχεια η εισαγωγή του μηχανισμού του attention βοήθησε στην ανάπτυξη ακόμα καλύτερων μοντέλων, αφού τώρα τα μοντέλα μπορούν να δώσουν σημασία στα σχετικά κομμάτια της εικόνας [45]. Στον τομέα του attention έχουν γίνει αρκετές προσεγγίσεις, όπως για παράδειγμα στην εργασία [46], όπου αναπτύχθηκε η έννοια του adaptive attention, μία τεχνική που επιτρέπει την εφαρμογή attention παραπάνω από μία φόρα για ένα decoding step. Μία ακόμα ενδιαφέρουσα τεχνική πάνω σε αυτόν τον τομέα είναι η εφαρμογή Bottom-Up και Top-Down Attention [47] για ακόμα βαθύτερη κατανόηση των εικόνων.

Εκτός όμως από τις παραπάνω τεχνικές, μέθοδοι βασισμένες σε Reinforcement Learning και GANs έχουν επίσης κάνει την εμφάνισή τους στο πρόβλημα του Image Captioning. Στην προσέγγιση της εργασίας [48] χρησιμοποιείται μία ανταγωνιστική adversarial μέθοδος εκπαίδευσης, οδηγώντας σε captions με πολύ μεγάλη ποικιλία. Τα captions είναι περισσότερο ανεξάρτητα από τα caption αναφοράς σε σχέση με τις προηγούμενες μεθόδους. Στην εργασία [49] χρησιμοποιείται Reinforcement Learning με μία αρχιτεκτονική actor-critic δίνοντας πολύ καλά αποτελέσματα.

Στον πίνακα 5.1 έχουμε τα αποτελέσματα των πιο καταξιωμένων εργασιών πάνω στο Image Captioning μαζί με τα αποτελέσματα του δικού μας μοντέλου.

5.2.1 Μετρικές αξιολόγησης

Σκοπός μας είναι να μπορούμε αυτόματα να αξιολογήσουμε για κάθε εικόνα την ποιότητα ενός παραγόμενου caption σε σχέση με ένα σύνολο caption αναφοράς, πρόβλημα εξαιρετικά δύσκολο λόγω της φύσης του αποτελέσματος που, για να αξιολογηθεί σωστά χρειάζεται η άποψη ενός ανθρώπου. Θα παρουσιάσουμε συνοπτικά τις διάφορες μετρικές που χρησιμοποιούνται για το πρόβλημα του Image Captioning και δίνουν μία εικόνα του παραπάνω σκοπού μας. Τα caption αναπαρίστανται σαν ένα σύνολο από n-grams, όπου n-gram είναι μία συλλογή από μία ή περισσότερες διατεταγμένες λέξεις.

BLUE

Η BLEU [52] είναι μία διάσημη μετρική, η οποία χρησιμοποιήθηκε κυρίως για το πρόβλημα του machine translation. Αναλύει τις συνυπάρξεις των n-grams μεταξύ των υποψήφιων

Πίνακας 5.1: *Πίνακας αποτελεσμάτων state of the art μοντέλων*

Results on MS COCO							
Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	CIDER
m-RNN [43]	0.670	0.491	0.344	0.243	0.239	-	-
Show and Tell [50]	-	-	-	0.277	0.237	-	0.854
Adversarial [48]	0.713	0.539	0.403	0.304	0.251	0.525	0.931
Hierarchical LSTMs and Adaptive Attention [51]	0.791	-	-	0.375	0.285	-	1.25
Bottom-Up and Top-Down attention [47]	0.79	0.641	0.491	0.359	0.276	0.571	1.17
Our Model	0.711	0.536	0.398	0.301	0.239	0.518	0.859

και των caption αναφοράς. Υπολογίζει n-gram precision σε επίπεδο συλλογής (corpus-level) μεταξύ προτάσεων ως εξής:

$$CP_n(C,S) = \frac{\sum_i \sum_k min(h_k(c_i), maxh_k(s_{ij}))}{\sum_i \sum_k h_k(c_i)} \tag{5.1}$$

Στην συνέχεια εντάσσεται ένα πέναλτι συντομίας (brevity penalty), καθώς το precision ευνοεί μικρές προτάσεις:

$$b(C,S) = \begin{cases} 1, & if \quad l_c > l_s \\ e^{1-l_s/l_c}, & if \quad l_c <= l_s \end{cases} \tag{5.2}$$

όπου l_c είναι το συνολικό μήκος των υποψήφιων προτάσεων και l_s είναι το μήκος της αναφοράς σε corpus-level. Το τελικό BLEU score υπολογίζεται ως:

$$BLEU_N(C,S) = b(C,S)exp(\sum_{n=1}^{N} w_n logCP_n(C,S)) \tag{5.3}$$

όπου N = 1,2,3,4. Γενικά η BLEU λειτουργεί χειρότερα όταν συγκρίνει ατομικές προτάσεις.

METEOR

Η METEOR [53] υπολογίζεται παράγοντας μία ευθυγράμμιση μεταξύ των λέξεων στις υποψήφιες και τις προτάσεις αναφοράς, με στόχο μία 1:1 αντιστοίχιση. Δίνοντας ένα σύνολο από αντιστοιχίσεις, m το METEOR score είναι ο αρμονικός μέσος μεταξύ του precision P_m και του recall R_m μεταξύ του καλύτερου υποψηφίου και του στόχου.

$$P_{en} = \gamma(\frac{ch}{m})^{\theta} \tag{5.4}$$

$$F_{mean} = \frac{P_m R_m}{\alpha P_m + (1-a) R_m} \tag{5.5}$$

$$P_m = \frac{|m|}{\sum_k h_k(c_i)} \tag{5.6}$$

$$R_m = \frac{|m|}{\sum_k h_k(s_{ij})} \tag{5.7}$$

$$METEOR = (1 - P_{en}) F_{mean} \tag{5.8}$$

ROUGE

Η ROUGE [54] είναι ένα σύνολο από μετρικές αξιολόγησης σχεδιασμένες για την αξιολόγηση αλγορίθμων σύνοψης κειμένου. Εμείς θα ασχοληθούμε με την μετρική $ROUGE_L$, η οποία χρησιμοποιεί ένα μέτρο που βασίζεται στην Μεγαλύτερη Κοινή Υπακολουθία (Longest Common Subsequence (LCS)). Μία LCS είναι ένα σύνολο από λέξεις που υπάρχουν σε δύο προτάσεις με την ίδια σειρά. Δίνοντας το μήκος $l(c_i, s_{ij})$ της LCS μεταξύ ενός ζευγαριού προτάσεων, η $ROUGE_L$ υπολογίζεται ως:

$$R_l = max_j \frac{l(c_i, s_{ij})}{|s_{ij}|} \tag{5.9}$$

$$P_l = max_j \frac{l(c_i, s_{ij})}{|c_i|} \tag{5.10}$$

$$ROUGE_L(c_i, S_i) = \frac{(1+\beta^2) R_l P_l}{R_l + \beta^2 P_l} \tag{5.11}$$

όπου R_l και P_l είναι το recall και το precision της LCS.

CIDEr

Η CIDER [55] μετρική μετράει την ομοφωνία σε captions εικόνων χρησιμοποιώντας στάθμιση Term Frequency Inverse Document Frequency (TF-IDF) για κάθε n-gram. Ο αριθμός των φορών που εμφανίζεται ένα n-gram σε μία πρόταση αναφοράς συμβολίζεται $h_k(s_{ij})$ και $h_k(c_i)$ για την υποψήφια πρόταση. Η στάθμιση TF-IDF $g_k(s_{ij})$ για κάθε n-gram υπολογίζεται ως:

$$g_k(s_{ij}) = \frac{h_k(s_{ij})}{\sum_{\omega_l} h_l(s_{ij})} log(\frac{|I|}{\sum_{I_p \in I} min(1, \sum_q h_k(s_p q))}) \tag{5.12}$$

Το $CIDER_n$ για n-grams μήκους n υπολογίζεται χρησιμοποιώντας την μέση ομοιότητα συνημιτόνου μεταξύ της υποψήφιας πρότασης και του στόχου:

$$CIDEr_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{g^n(c_i) g^n(s_{ij})}{||g^n(c_i)|| ||g^n(s_{ij})||} \tag{5.13}$$

όπου $g^n(c_i)$ είναι ένα διάνυσμα από τα $g_k(c_i)$ που αντιστοιχούν σε n-grams με μήκος n. Τα scores από n-grams διαφορετικού μήκους συνδυάζονται για το τελικό σκορ:

$$CIDEr(c_i, S_i) = \sum_{n=1}^{N} w_n CIDEr_n(c_i, S_i) \tag{5.14}$$

5.3 Ρυθμίσεις Πειράματος

Στο παρακάτω κομμάτι θα ασχοληθούμε με τα διάφορα πειράματα που τρέξαμε, τις υπερπαραμέτρους και τις επιλογές που οδήγησαν στο βέλτιστο αποτέλεσμα.

5.3.1 Διαδικασία Εκπαίδευσης

Το μοντέλο μας εκπαιδεύεται για *100 εποχές* χρησιμοποιώντας *adam optimizer* [16] με *learning rate = 0.0001*. Χρησιμοποιώντας μία GPU NVIDIA Tesla k40m η εκπαίδευση διαρκούσε σχεδόν 4 ημέρες. Η διαδικασία που ακολουθεί το μοντέλο σε κάθε βήμα της εκπαίδευσης είναι η παρακάτω:

- Αρχικά, η εικόνα εισόδου περνάει μέσα από τον encoder, όπου παράγονται τα annotation vectors που περιέχουν την πληροφορία της εικόνας.
- Έπειτα, η έξοδος του *encoder*, δηλαδή τα annotations vectors της εικόνας εισόδου, η κρυφή κατάσταση, η οποία αρχικοποιείται με την τιμή 0 και η είσοδος του *decoder*, η οποία είναι το token '<start>' περνάνε στον decoder.
- Στην συνέχεια, ο decoder επιστρέφει τις προβλέψεις και την κρυφή κατάστασή του.
- Η κρυφή κατάσταση του decoder περνιέται ξανά πίσω στο μοντέλο και οι προβλέψεις χρησιμοποιούνται για να υπολογιστούν οι απώλειες. Σαν συνάρτηση κόστους χρησιμοποιούμε την cross-entropy που συζητήσαμε και στο κεφάλαιο 2.2.1.
- Για να αποφύγουμε το μοντέλο μας να γίνεται train σε λανθασμένες ακολουθίες χρησιμοποιούμε κατά την διάρκεια της εκπαίδευσης την τεχνική teacher forcing (3.3.1), για να αποφασίσουμε την επόμενη είσοδο του decoder. Σε κάθε βήμα δίνουμε στον decoder την σωστή λέξη, ώστε να μάθει τις σωστές ακολουθίες και να διορθώνει τις στατιστικές ιδιότητες του μοντέλου γρήγορα.
- Το τελευταίο βήμα είναι να υπολογίσουμε τα gradients και να τα εφαρμόσουμε στον optimizer, για να γίνει η διαδικασία του backpropagation.

Ένα από τα μεγαλύτερα προβλήματα που αντιμετωπίσαμε κατά την διάρκεια της εκπαίδευσης ήταν η διάρκειά της που όπως αναφέραμε έφτανε τις 4 μέρες. Μία τεχνική, η οποία βελτίωνε ως έναν βαθμό τον χρόνο εκπαίδευσης ήταν η 'offline' εξαγωγή των features - annotation vectors των εικόνων και η αποθήκευσή τους. Στην συνέχεια μπορούσαμε να περάσουμε στην εκπαίδευση του αποκωδικοποιητή χωρίς να χρειάζεται να γίνεται πρώτα η επεξεργασία των εικόνων, κάτι που βελτίωνε αρκετά την χρονική επίδοση της εκπαίδευσης. Η μέθοδος αυτή όμως, δεν έλυνε τελείως το πρόβλημα καθώς για να χρησιμοποιήσουμε διαφορετικό image encoder, θα έπρεπε να επαναλάβουμε την διαδικασία εξαγωγής των annotation vectors.

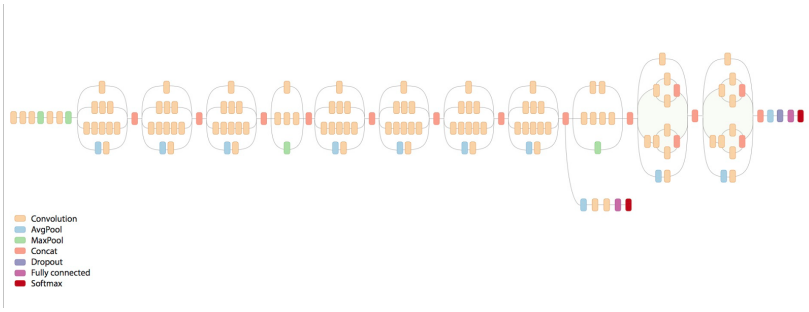
5.3.2 Πειραματική Διαδικασία

Σκοπός της πειραματικής διαδικασίας είναι η βαθύτερη κατανόηση των χαρακτηριστικών του μοντέλου και της αύξησης των μετρικών απόδοσής του. Θα αναπτύσσουμε στην συνέχεια τις πιο βασικές παραμέτρους που επηρεάζουν σε μεγαλύτερο βαθμό την επίδοση του μοντέλου. Η ανάλυση αυτή των παραμέτρων παρέχει μία ουσιαστική εποπτεία στην διαδικασία αξιολόγησης του μοντέλου.

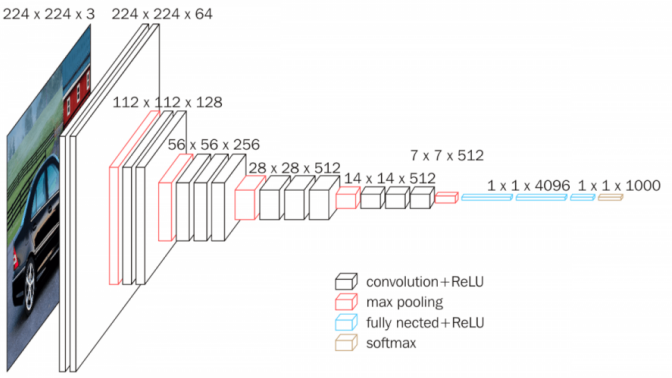
Επιλογή Image Encoder

Μία από τις σημαντικές αποφάσεις για την ανάπτυξη του μοντέλου μας είναι η επιλογή του κατάλληλου Image Encoder. Ο encoder θα είναι ένα pre-trained CNN, το οποίο θα χρησιμοποιείται για την εξαγωγή των annotation vectors.

Για το πείραμά μας χρησιμοποιήσαμε τρεις διαφορετικές αρχιτεκτονικές για την εξαγωγή των χαρακτηριστικών των εικόνων. Το InceptionV3 [56], το VGG-6 [57] και το Resnet-50 [58]. Οι τρεις παραπάνω αρχιτεκτονικές έχουν χρησιμοποιηθεί με τεράστια επιτυχία σε προβλήματα της όρασης Υπολογιστών, όπως Image Detection και Image Classification. Για να τις προσαρμόσουμε στο πρόβλημα μας, αφαιρούμε το τελικό πλήρως συνδεδεμένο επίπεδο που υπάρχει και στα τρία, παίρνοντας έτσι χαρακτηριστικά από τα προηγούμενα επίπεδα συνέλιξης. Παρατηρούμε τις αρχιτεκτονικές τους στις εικόνες 5.1, 5.2 και 5.3 αντίστοιχα.

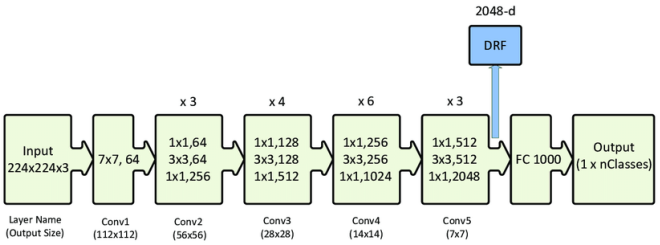


Εικόνα 5.1: *Αρχιτεκτονική InceptionV3*



Εικόνα 5.2: *Αρχιτεκτονική VGG-16*

Και οι 3 αρχιτεκτονικές που χρησιμοποιούμε έχουν εκπαιδευτεί πάνω στο ImageNet [59],



Εικόνα 5.3: *Αρχιτεκτονική Resnet50*

Πίνακας 5.2: *Πίνακας αποτελεσμάτων διαφορετικών Image Encoder*

Results based on Image Encoder		
Encoder	BLEU-4	CIDER
InceptionV3	0.298	0.844
VGG - 16	0.295	0.841
Resnet50	0.299	0.852

ένα από τα μεγαλύτερα και πιο διαδεδομένα στην ερευνητική κοινότητα dataset, για το πρόβλημα του Image Classification.

Τελικώς τα αποτελέσματα που προσέφεραν και οι 3 αρχιτεκτονικές ήταν παρόμοια και τα παρατηρούμε στον πίνακα 5.2 με βάση το validation set με όλες τις υπόλοιπες παραμέτρους του μοντέλου να μένουν σταθερές.

Συμπεραίνουμε λοιπόν ότι από άποψη απόδοσης ο καλύτερος Image Encoder είναι το Resnet50.

Επιλογή είδους Attention

Όπως αναφέραμε και στο κεφάλαιο 4.2.2 δοκιμάσαμε δύο διαφορετικά είδη soft attention το Global και το Local. Με το Global Attention λαμβάνουμε υπόψιν όλες τις κρυφές καταστάσεις του encoder, δηλαδή όλα τα annotation vectors, ενώ με το Local εξετάζουμε ορισμένους μόνο annotation vectors. Παρατηρούμε ότι το Global Attention αποδίδει ελαφρώς καλύτερα από το Local, όμως λόγω της υπολογιστικής του πολυπλοκότητας η διαδικασία της εκπαίδευσης γίνεται πολύ πιο επίπονη για το σύστημα. Επομένως επιλέγουμε Soft Local Attention. Στον πίνακα 5.3 παρατηρούμε τα αποτελέσματα για τα attentions, όπως επίσης και την σημαντική διαφορά που έχει το σύστημα, χωρίς καθόλου attention, αποδεικνύοντας την αναγκαιότητα του μηχανισμού.

Πίνακας 5.3: *Πίνακας αποτελεσμάτων Local, Global Attention και No Attention*

Attention	BLEU-4	CIDER
Local	0.299	0.850
Global	0.301	0.854
No Attention	0.257	0.793

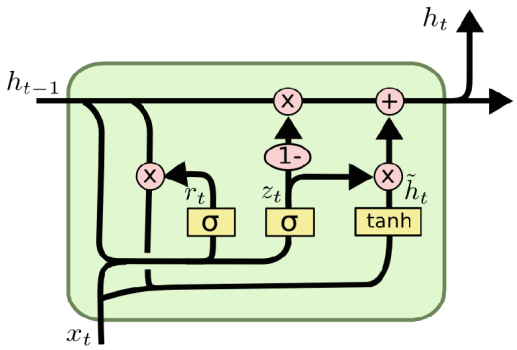
Πίνακας 5.4: *Πίνακας αποτελεσμάτων GRU και LSTM*

Decoder	BLEU-4	CIDER
GRU	0.279	0.822
LSTM	0.301	0.853

Επιλογή Decoder

Ένα ακόμα πολύ σημαντικό κομμάτι για το μοντέλο είναι να αποφασιστεί ποια θα είναι η βασική μονάδα επεξεργασίας του μοντέλου για τον αποκωδικοποιητή (decoder), δηλαδή οι νευρώνες RNN, LSTM ή GRU (Gated Recurrent Unit). Η επιλογή αυτή είναι σημαντική γιατί κάθε νευρώνας έχει διαφορετική πολυπλοκότητα και επηρεάζει την επεκτασιμότητα του μοντέλου. Στα πλαίσια του πειράματος μας, ασχοληθήκαμε με LSTM και GRU. Για τα κλασσικά RNN και το LSTM αναφερθήκαμε με λεπτομέρεια στο κεφάλαιο 2.3.4.

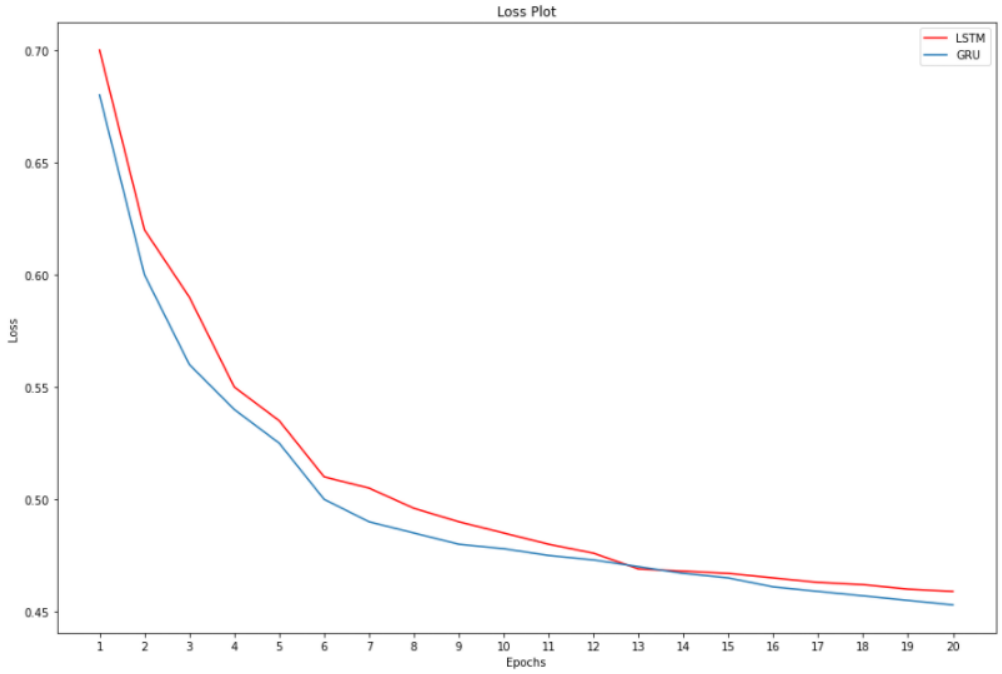
Τα GRU [60] (εικόνα 5.4) είναι παρόμοια με τα LSTMs με τη μόνη διαφορά ότι δεν έχουν πύλη εξόδου. Αυτό τα κάνει πολύ πιο αποδοτικά καθώς χρησιμοποιούνται λιγότεροι παράμετροι εσωτερικά του κυττάρου. Γενικά παρουσιάζουν πολύ καλή επίδοση σε μικρού όγκου δεδομένα.



Εικόνα 5.4: *Αρχιτεκτονική κυττάρου GRU*

Παρόλο που η επιλογή του GRU οδηγεί σε ένα αρκετά ταχύτερο μοντέλο, το μέγεθος των δεδομένων και το μήκος των προτάσεων οδηγεί σε πτώση της επίδοσης, όπως μπορούμε να δούμε από τον πίνακα 5.4, όποτε επιλέγουμε το LSTM.

Επίσης είχε ενδιαφέρον το γεγονός ότι η χρήση GRU δίνει μικρότερο loss στην εκπαίδευση, όπως παρατηρούμε στην εικόνα 5.5, αλλά όχι καλύτερα αποτελέσματα στο evaluation, κάτι που πιθανώς έχει να κάνει με την απλοϊκότερη αρχιτεκτονική του.



Εικόνα 5.5: *Γραφική Loss Function για LSTM, GRU για τις πρώτες 20 εποχές*

Πίνακας 5.5: *Πίνακας αποτελεσμάτων σε σχέση με μέγεθος λεξιλογίου*

Vocabulary Size	BLEU-4	CIDER
5,000	0.301	0.857
10,000	0.281	0.836
20,000	0.264	0.827

Μέγεθος Λεξιλογίου

Μία ακόμα σημαντική παράμετρος του μοντέλου μας είναι το μέγεθος του λεξιλογίου. Ενώ η λογική σκέψη θα ήταν να χρησιμοποιήσουμε όσο το δυνατό μεγαλύτερο λεξιλόγιο, ώστε να αποφεύγουμε τις περιπτώσεις όπου συναντάμε λέξη, η οποία δεν είναι καταχωρημένη στο λεξιλόγιο μας, δηλαδή τις λέξεις '<unk>' όπως αναφέραμε και στο κεφάλαιο 4.1.2.

Στην πράξη όμως δεν συμβαίνει αυτό. Επιλέξαμε λεξιλόγια μεγέθους 5000, 10000 και 20000 λέξεων και όπως βλέπουμε στον πίνακα 5.5 τα καλύτερα αποτελέσματα τα έχουμε με το λεξιλόγιο των 5000 λέξεων. Αυτό μπορεί να συμβαίνει λόγω του ότι οι βασικές λέξεις δεν ξεπερνάνε τις 5000 και όταν αυξάνεται το μέγεθος του λεξιλογίου δεν μπορεί να μάθει το μοντέλο τις κατάλληλες σχέσεις μεταξύ των λέξεων, οδηγώντας σε περισσότερες αστοχίες.

Περαιτέρω επεξεργασία λεξιλογίου

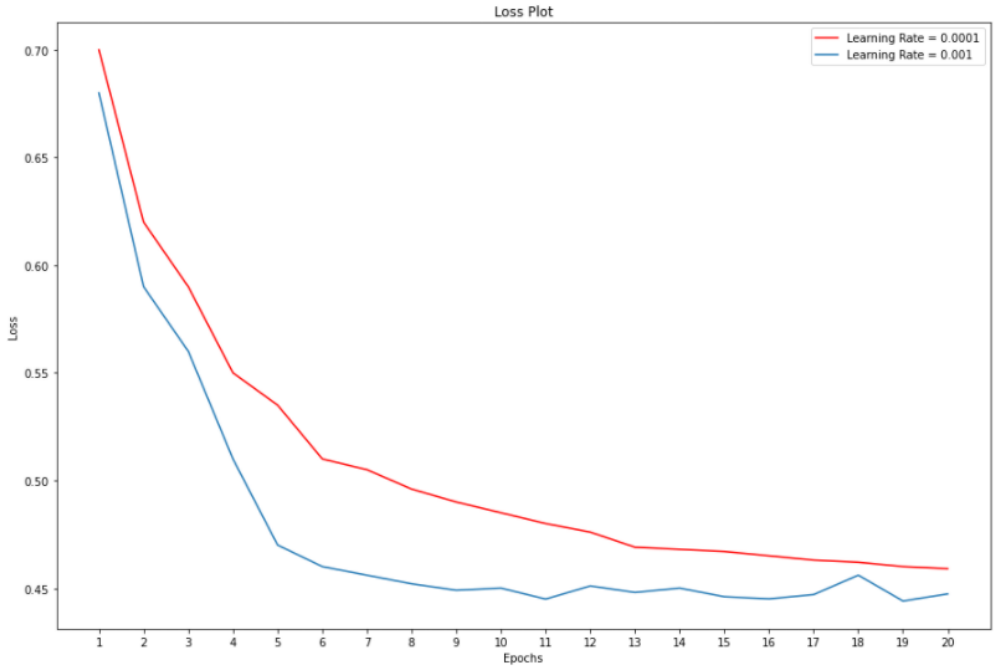
Μία ακόμα μέθοδος που χρησιμοποιήθηκε στην εκπαίδευση και χρήση του μοντέλου ήταν η αφαίρεση από το λεξιλόγιο των άρθρων 'a', 'an', 'the'. Η τακτική αυτή βοήθησε το μο-

ντέλο να παράγει captions με μεγαλύτερη σημασιολογική αξία, καθώς δεν χαραμιζόταν στον υπολογισμό άρθρων, που παρόλο που δεν προσφέρουν κάτι σημαντικό σημασιολογικό, αποτελούσαν τις λέξεις που εμφανιζόντουσαν πιο συχνά από οποιαδήποτε άλλη. Δυστυχώς, όμως με αυτή την τακτική δεν γινόταν να εφαρμοστούν οι μέθοδοι αξιολόγησης που χρησιμοποιήθηκαν παραπάνω, οπότε τα αποτελέσματά της κρίθηκαν ποιοτικά, μέσω της χρήσης του μοντέλου.

Λοιπές Υπερπαράμετροι Μοντέλου

Τέλος, θα ασχοληθούμε με τις πιο τεχνικές υπερπαραμέτρους του μοντέλου και θα συζητηθούν εν συντομία τα συμπεράσματα για αυτές.

Πρώτα από όλα θα αναφερθούμε στον *ρυθμό μάθησης learning rate*. Όπως αναφέρθηκε και πιο πάνω επιλέχτηκε η τιμή 0.0001 και αυτό έγινε για να υπάρχει ένα μέσο στην ταχύτητα εκμάθησης, ώστε να αποφεύγεται τόσο οι περιπτώσεις του overfitting, αλλά και του underfitting. Με τιμή 0.001 το μοντέλο συνέκλινε υπερβολικά σύντομα, μόνο στις περίπου 20 εποχές και τα αποτελέσματα ήταν απογοητευτικά στο validation set (εικόνα 5.6). Από την άλλη επιλογή της τιμής σε 0.00001 ήταν εξαιρετικά μικρή και το μοντέλο ακόμα και μετά από αρκετά μεγάλο χρονικό διάστημα δεν σημείωνε κάποια σημαντική πρόοδο.



Εικόνα 5.6: *Γραφική Loss Function για τις πρώτες 20 εποχές σε σχέση με learning rate*

Στην συνέχεια ορίστηκε σαν *μέγιστο μέγεθος ενός caption* οι 20 λέξεις. Η επιλογή αυτής της τιμής έγινε με βάση και τις επιλογές των διαφόρων άλλων state of the art μεθόδων για Image Captioning. Η μείωση του σε κάτω από 15 λέξεις οδηγούσε σε μικρή μείωση της απόδοσης, ενώ η αύξηση σε τιμές μεγαλύτερες του 20 δεν είχαν κάποια ουσιαστική δράση πάνω στο μοντέλο μας και οδηγούσαν σε μεγαλύτερη κατανάλωση μνήμης λόγω της διαδικασίας του padding που εφαρμόζουμε σε κάθε πρόταση.

Πίνακας 5.6: *Πίνακας αποτελεσμάτων σε σχέση με πλάτος ακτινωτής αναζήτησης*

Beam Size	BLEU-4	CIDER
1	0.298	0.847
3	0.297	0.853
5	0.301	0.857
7	0.294	0.836

Το μέγεθος του *batch size* ορίστηκε ίσο με 32. Επίσης, ο αριθμός των εποχών τέθηκε στο 100. Αποδείχτηκε ο απαραίτητος αριθμός για να προλάβει το μοντέλο να φτάσει στα βέλτιστα επίπεδά του, αλλά παραπάνω εποχές θα οδηγούσαν σε overfitting και μείωση της συνολικής απόδοσης του συστήματος.

5.3.3 Διαδικασία Decoding

Για την διαδικασία του decoding, δηλαδή της παραγωγή ενός caption για μία δοσμένη εικόνα, χρησιμοποιήσαμε την μέθοδο της *ακτινωτής αναζήτησης (beam search)* [61]. Η λογική αυτής της τεχνικής είναι αντί σε κάθε βήμα της ακολουθίας να διαλέγουμε άπληστα το επόμενο βήμα, με την ακτινωτή αναζήτηση επεκτείνονται όλα τα δυνατά επόμενα βήματα και κρατιούνται τα k πιο πιθανά, όπου το k το ορίζουμε εμείς. Οι πιο συνηθισμένες τιμές για το k είναι 1 για μία άπληστη αναζήτηση και ανεβαίνουν για πιο πολύπλοκα προβλήματα. Η διαδικασία αναζήτησης τερματίζει όταν κάθε υποψήφιος έχει φτάσει σε '<end>' token. Για τα πειράματά μας δοκιμάσαμε τις τιμές που φαίνονται στον πίνακα 5.6 με το καλύτερο αποτέλεσμα για την τιμή 5.

5.4 Οπτικοποίηση Αποτελεσμάτων

Στο παρακάτω υποκεφάλαιο θα δούμε ορισμένα από τα αποτελέσματα που έδωσε σε εικόνες. Θα τα χωρίσουμε σε υποκατηγορίες με βάση την σημασιολογική τους αξία.

Καλό Αποτέλεσμα

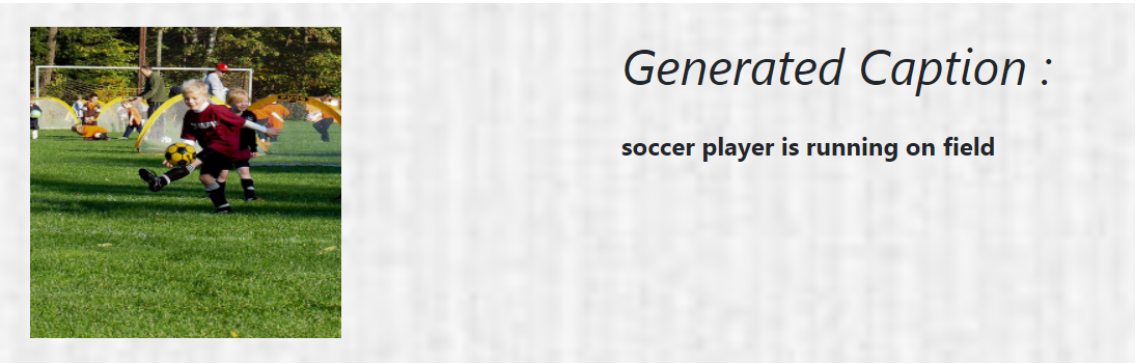
Ένα caption το θεωρούμε καλό αν κατανοεί το περιεχόμενο της εικόνας και παράγει μία περιγραφή, η οποία είναι γραμματικά σωστή.

Η εικόνα 5.7 αποτελεί ένα χαρακτηριστικό παράδειγμα, καθώς εντοπίζονται όλα τα απαραίτητα χαρακτηριστικά της εικόνας, δηλαδή ότι είναι ένα μεγάλο τζετ, ότι ο ουρανός είναι συννεφιασμένος και ότι το εικονιζόμενο αεροπλάνο πετάει. Επίσης η γραμματική είναι άριστη.

Εικόνα 5.7: *Καλό αποτέλεσμα χρήσης του μοντέλου*

Μέτριο Αποτέλεσμα

Ένα μέτριο caption είναι αυτό που δεν κατανοεί και δεν περιγράφει πλήρως το περιεχόμενο της εικόνας.



Εικόνα 5.8: *Μέτριο αποτέλεσμα χρήσης του μοντέλου*

Για παράδειγμα στην εικόνα 5.8 το μοντέλο κατανοεί η σκηνή έχει σχέση με ποδόσφαιρο και καταλαβαίνει ότι είναι σε γήπεδο, αλλά χάνει ένα πολύ σημαντικό χαρακτηριστικό της εικόνας, το γεγονός ότι ο ποδοσφαιριστής είναι ένα μικρό παιδί.

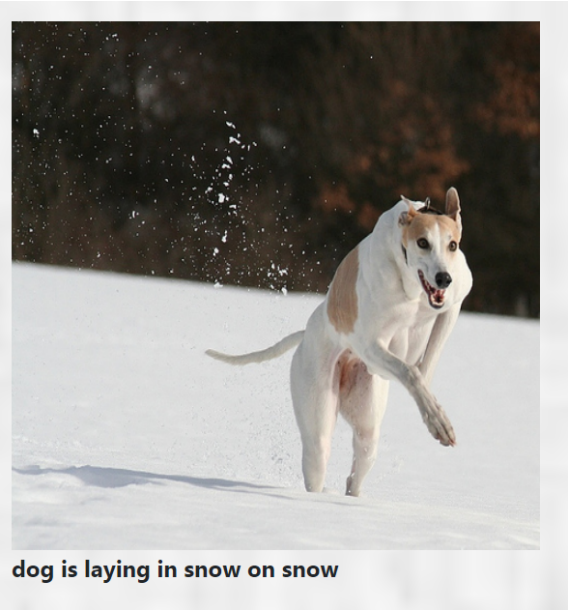
Κακό Αποτέλεσμα

Κακό μπορεί να χαρακτηριστεί ένα caption όταν χάνει βασικά χαρακτηριστικά της εικόνας και έχει και γραμματικά λάθη.

Στην εικόνα 5.9 το μοντέλο εντοπίζει τον σκύλο και το χιόνι, όμως δεν καταλαβαίνει ότι ο σκύλος τρέχει και περιέχει και γραμματικό λάθος, που οδηγεί τελικά σε μία τελείως λάθος πρόταση.

5.5 Training στο σύστημα ARIS

Σε αυτό το σημείο είναι σημαντικό να αναφερθεί πού έγινε η εκπαίδευση του παραπάνω μοντέλου. Έγινε άμεσα αντιληπτό ότι η εκπαίδευση σε ένα τέτοιο μεγάλο όγκο δεδομένων, ενός αρκετά πολύπλοκου συστήματος δεν θα ήταν εφικτό στον τοπικό υπολογιστή. Η πρώτη



Εικόνα 5.9: *Κακό αποτέλεσμα χρήσης του μοντέλου*

σκέψη ήταν να χρησιμοποιηθούν τα notebooks του Google Colab ή του Kaggle που δίνουν την δυνατότητα στον χρήστη να χρησιμοποιήσει GPUs στο cloud για την εκπαίδευση μοντέλων. Όμως και εκέι οι διαθέσιμοι πόροι δεν ήταν αρκετοί για την εκτέλεση των πειραμάτων μας.

Η λύση δόθηκε από το Εθνικό Δίκτυο Υποδομών Τεχνολογίας και Έρευνας(ΕΔΥΤΕ), μέσω του ARIS. Το ΕΔΥΤΕ παρέχει υπολογιστικούς πόρους υψηλών επιδόσεων στις ελληνικές και διεθνείς επιστημονικές και ερευνητικές κοινότητες για την πραγματοποίηση επιστημονικής έρευνας. Η υποδομή ARIS αποτελείται από τέσσερις νησίδες υπολογιστικών συστημάτων βασισμένους σε αρχιτεκτονική Intel x86 διασυνδεδεμένους σε ένα ενιαίο δίκτυο που προσφέρουν πολλαπλές δυνατότητες και αρχιτεκτονικές επεξεργασίας.

Στο εργαστήριο μας δόθηκε πρόσβαση στην νησίδα κόμβων επιταχυντών GPU που αποτελείται από 44 εξυπηρετητές Dell PowerEdge R730. Κάθε εξυπηρετητής περιέχει 2 επεξεργαστές Intel Xeon E5-2660v3, 64 GB μνήμης και 2 κάρτες GPU NVidia K40. Στην συγκεκριμένη υποδομή μπορέσαμε να τρέξουμε τα πειράματά μας χρησιμοποιώντας shell scripts και τα resources ήταν αρκετά για να γίνει επαρκής εκπαίδευση και έλεγχος του μοντέλου.

Για αυτό τον λόγο ευχαριστούμε τον ΕΔΥΤΕ, καθώς χωρίς την συμβολή του η διεκπεραίωση των πειραμάτων θα ήταν αδύνατη.

Κεφάλαιο 6

Διαδικτυακή Εφαρμογή

Στο κεφάλαιο αυτό θα γίνει η παρουσίαση της διαδικτυακής εφαρμογής που κατασκευάσαμε, η οποία βασίζεται στο μοντέλο του Image Captioning. Εκτός όμως από το μοντέλο μας, χρησιμοποιήθηκαν και άλλα δύο pretrained μοντέλα για την ενίσχυση των δυνατοτήτων της εφαρμογής. Σκοπός μας είναι να δείξουμε τις ευκαιρίες που προσφέρουν τα μοντέλα μηχανικής μάθησης σε μία πραγματική διαδικτυακή εφαρμογή.

6.1 Τεχνικό Υπόβαθρο Ανάπτυξης Διαδικτυακών Εφαρμογών

Για την υλοποίησης της διαδικτυακής εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python, που είδαμε και στα νευρωνικά δίκτυα, και επίσης οι γλώσσες προγραμματισμού *Javascript [62], HTML [63], CSS [64]*, οι οποίες αποτελούν τους βασικούς πυλώνες σχεδόν για κάθε διαδικτυακή εφαρμογή.

6.1.1 Εφαρμογές REST

Το *REpresentational State Transfer (REST)* [65] αποτελεί το πρωτόκολλο που χρησιμοποιεί η διαδικτυακή μας εφαρμογή. Είναι ένα στυλ αρχιτεκτονικής λογισμικού που αποτελείται από κατευθυντήριες γραμμές και πρακτικές για τη δημιουργία επεκτάσιμων υπηρεσιών διαδικτύου. Το REST είναι ένα σύνολο περιορισμών που εφαρμόζεται στην σχεδίαση των μερών σε ένα διαμοιρασμένο σύστημα υπερμέσων, με σκοπό να οδηγήσει σε μία πιο αποδοτική και εύκολα συντηρήσιμη αρχιτεκτονική. Είναι πλέον ευρέως αποδεκτό και αποτελεί μία απλούστερη εναλλακτική στο SOAP [66]. Τα RESTful συστήματα επικοινωνούν μέσω του πρωτοκόλλου HTTP με τα ίδια HTTP verbs (GET, POST, PUT, DELETE) που χρησιμοποιούν και οι web browsers για να λάβουν σελίδες του διαδικτύου και να στείλουν δεδομένα σε απομακρυσμένους servers. Οι απαντήσεις του server σε κάθε REST call συνηθίζουν να είναι της μορφής HTML, XML και κυρίως σε μορφή JSON. Πλέον, η παράλληλη ανάπτυξη της ίδιας εφαρμογής σε πολλαπλές πλατφόρμες (web, android, ios κτλ) έχει οδηγήσει στην καθιέρωση του REST πρωτοκόλλου καθώς με ένα μόνο backend τύπου REST μπορεί να υπάρξει επικοινωνία με κάθε μορφής frontend.

6.1.2 Flask Backend

Το *Flask* [67] αποτελεί ένα backend micro-framework γραμμένο σε Python. Χαρακτηρίζεται ως microframework επειδή δεν χρειάζεται ιδιαίτερα εργαλεία ή βιβλιοθήκες για να λειτουργήσει. Αρχικά η ιδέα του Flask ήταν ένα πρωταπριλιάτικο αστείο, που τελικά απέκτησε τέτοια φήμη ώστε να γίνει μία σοβαρή εφαρμογή. Πλέον, είναι ένα από τα διασημότερα framework της Python, αφού τον Οκτώβριο του 2020 έχει τα δεύτερα περισσότερα αστέρια στο Github όσον αφορά τα web-development frameworks, ελαφρώς πίσω από το Django. Χάρη στην απλότητα και την ευελιξία του χρησιμοποιείται και από κολοσσούς στην αγορά της τεχνολογίας, όπως το Pinterest [68] και το LinkedIn. Το Flask, λοιπόν, χρησιμοποιήθηκε για το στήσιμο του backend της εφαρμογής μας.

6.1.3 Bootstrap Frontend

Το *Bootstrap* [69] είναι δωρεάν και open-source CSS framework που στοχεύει σε responsive, mobile-first front-end web development. Περιέχει templates σε CSS και Javascript για τυπογραφία, φόρμες, κουμπιά, navigation και αλλά κομμάτια μίας responsive διεπαφής. Το Bootstrap αναπτύχθηκε αρχικά στο Twitter, ως ένα framework που να ενθαρρύνει την συνοχή μεταξύ των εσωτερικών εργαλείων. Πριν το bootstrap χρησιμοποιούνταν πολλές διαφορετικές βιβλιοθήκες για ανάπτυξη διεπαφών, κάτι που οδηγούσε σε ασυνέπειες και μεγάλο βάρος συντήρησης. Πλέον, είναι το πιο διάσημο HTML, CSS και Javascript framework για ανάπτυξη διεπαφών και χρησιμοποιείται από τεράστιες εταιρίες, όπως το Spotify κα το Twitter [70]. Αυτό είναι το framework που χρησιμοποιήθηκε για τον σχεδιασμό του frontend της διαδικτυακής εφαρμογής μας.

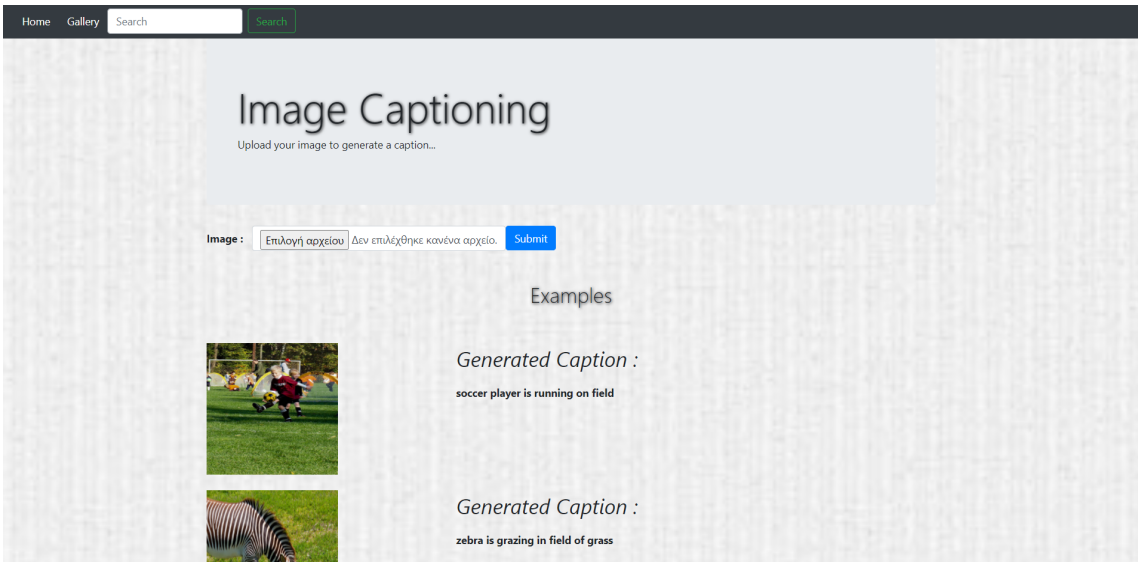
6.2 Περιήγηση στην Εφαρμογή

Παρακάτω θα δούμε τις διάφορες καρτέλες της διαδικτυακής εφαρμογής μας και τις δυνατότητες που προσφέρει στους χρήστες. Κάθε δυνατότητα της εφαρμογής είναι συνδεδεμένη και με ένα διαφορετικό μοντέλο, που τονίζει την ιδιαιτερότητα της εφαρμογής.

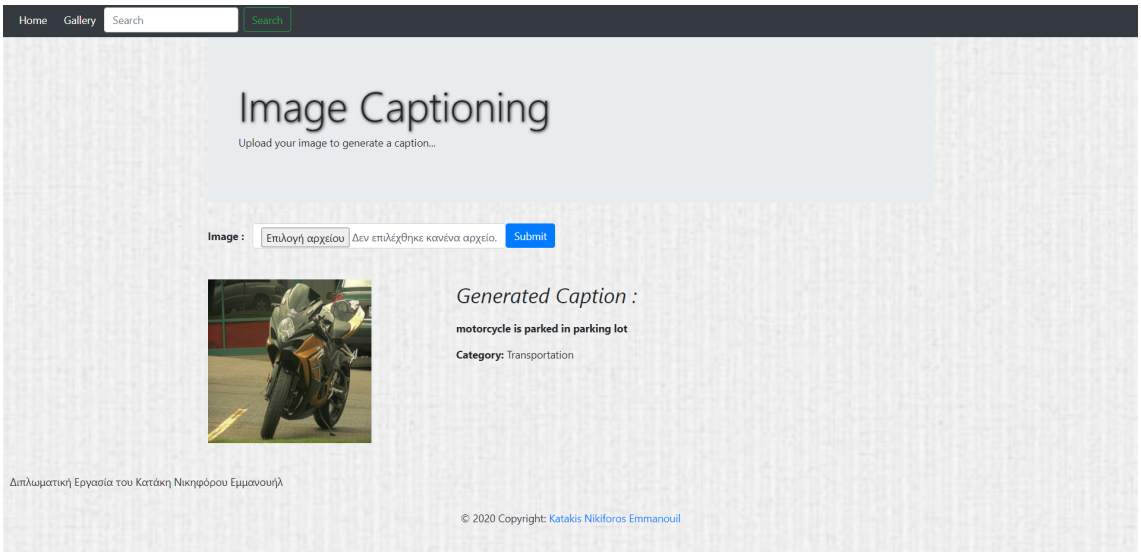
6.2.1 Αρχική Σελίδα

Η αρχική σελίδα είναι αυτή που αντικρίζει ο χρήστης όταν εισέρχεται στην σελίδα. Όπως μπορούμε να δούμε και στην εικόνα υπάρχει ένα navigation bar με τις επιλογές *Home, Gallery, Search*. Τις δύο τελευταίες θα τις αναλύσουμε παρακάτω με λεπτομέρεια.

Όπως βλέπουμε στην αρχική σελίδα 6.1 ο χρήστης έχει την δυνατότητα να κάνει upload μία εικόνα από τον προσωπικό του υπολογιστή. Η εικόνα αυτή αποθηκεύεται στον server και αφού δεχτεί την απαραίτητη προεπεξεργασία, δηλαδή την μετατροπή των διαστάσεων της σε 224x224 εισέρχεται στο μοντέλο όπου παράγεται το κατάλληλο caption. Επίσης υπάρχουν και κάποια παραδείγματα με captions που έχουν παραχθεί από το μοντέλο μας. Το μοντέλο φορτώνεται με το που εκκινεί ο server και είναι σε έτοιμότητα΄ να δεχτεί τα request του χρήστη. Το σύστημα, επίσης, αποθηκεύει τις εικόνες μαζί με τα caption τους για τις υπόλοιπες λειτουργίες της εφαρμογής.



Εικόνα 6.1: *Αρχική σελίδα (Home)*



Εικόνα 6.2: *Σελίδα αποτελέσματος captioning*

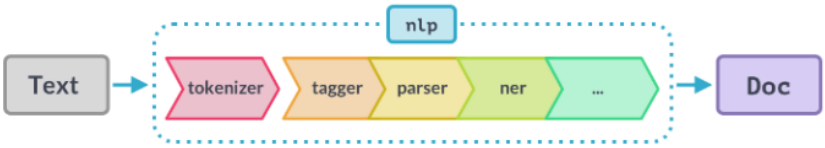
Στην εικόνα 6.2 παρατηρούμε το αποτέλεσμα μετά το upload κάποιας εικόνας. Στο συγκεκριμένο παράδειγμα ο χρήστης ανέβασε μία παρκαρισμένη μηχανή και πήρε το αντίστοιχο caption. Εκτός από το caption βλέπουμε και μία επιπλέον πληροφορία, το *category*, με το οποίο θα ασχοληθούμε στην συνέχεια.

6.2.2 Σελίδα αναζήτησης

Στην αρχική σελίδα είδα ότι υπάρχει και η επιλογή Search. Στο πεδία πάνω στο navigation bar ο χρήστης μπορεί να εισάγει μία πρόταση και στην συνέχεια θα του γυρίσει μία λίστα με τις πιο σχετικές εικόνες. Το ιδιαίτερο σε αυτή την αναζήτηση είναι ότι γίνεται με βάση τα caption που έχουν παραχθεί από το μοντέλο μας. Οπότε για να γίνει σωστή αναζήτηση θα πρέπει να υπάρχει ένα σύστημα βαθμολόγησης της ομοιότητας δύο διαφορετικών προτάσεων, και για αυτό το λόγο χρησιμοποιούμε ένα ακόμα pretrained νευρωνικό δίκτυο.

Μοντέλο spaCy

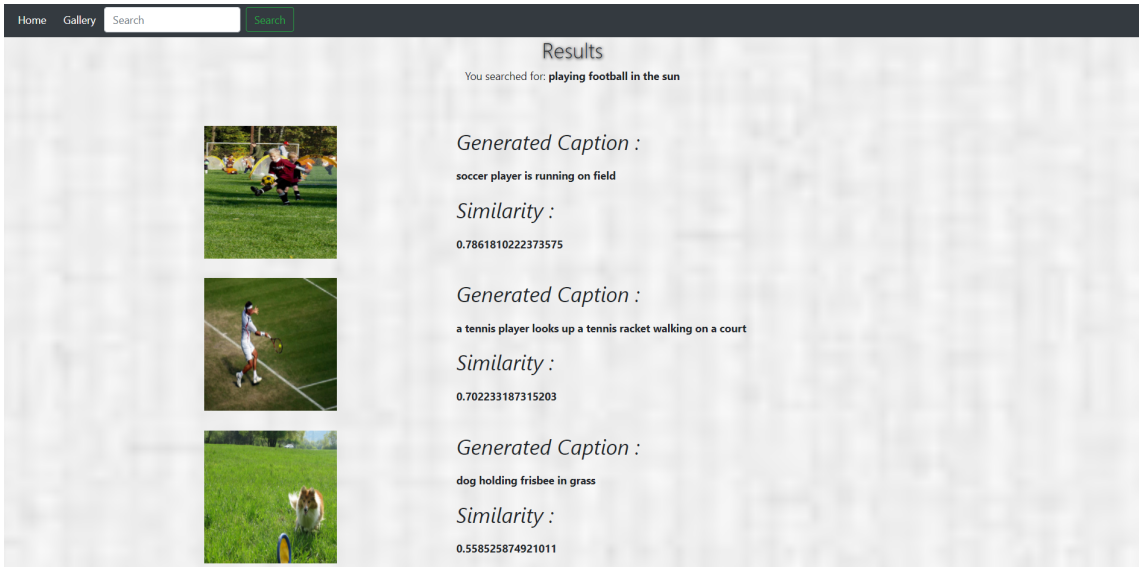
Για την διαδικασία ομοιότητας μεταξύ προτάσεων χρησιμοποιήσαμε το spaCy [71]. Το spaCy είναι μία open-source βιβλιοθήκη για προχωρημένη επεξεργασία φυσικής γλώσσας, γραμμένη στην γλώσσα προγραμματισμού Python. Σε αντίθεση με άλλες βιβλιοθήκες που ασχολούνται με την εκπαίδευση και την έρευνα στο πεδίο του Natural Language Processing (NLP), το spaCy επικεντρώνεται στην παροχή λογισμικού έτοιμο για χρήση στην παραγωγή. Παρέχει pretrained βαθιά νευρωνικά δίκτυα, για διάφορα NLP tasks, όπως part-of-speech tagging και dependency parsing, σε πάρα πολλές γλώσσες όπως Αγγλικά, Γερμανικά, Ελληνικά και Ισπανικά.



Εικόνα 6.3: *spaCy Pipeline*

Για την δική μας περίπτωση εκμεταλλευόμαστε τα pretrained CNN models του spaCy για την αγγλική γλώσσα [72] και συγκεκριμένα το *en_core_web_lg*. Μέσω του pretrained μοντέλου μπορούμε να παράγουμε διανύσματα για τις δύο προτάσεις, ακολουθώντας το pipeline της εικόνας 6.3, και στην συνέχεια με ομοιότητα συνημιτόνου να παράγουμε την ομοιότητά τους.

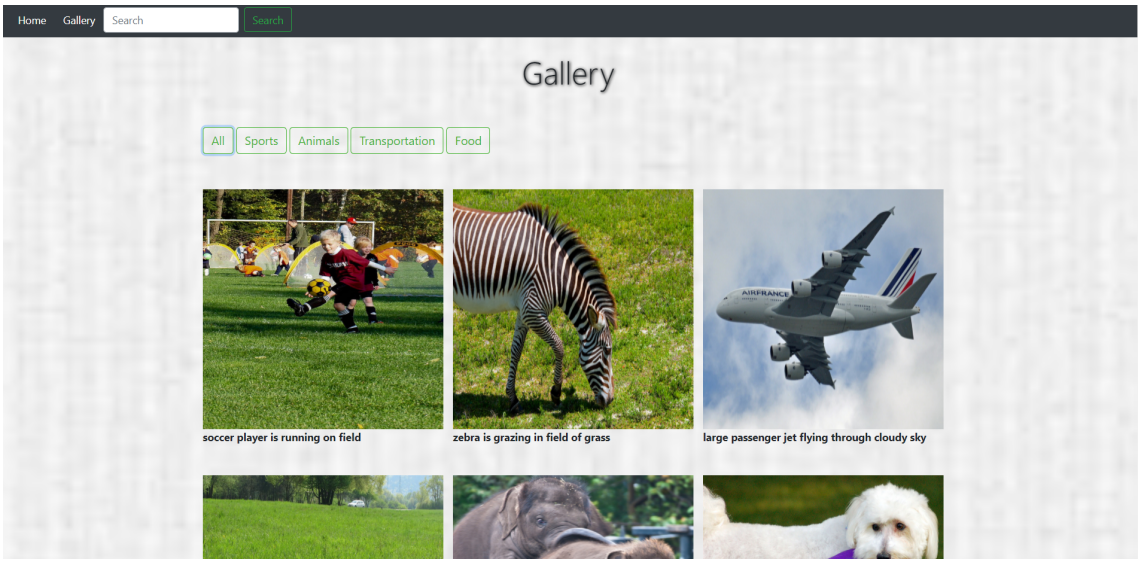
Αφού γίνει η παραπάνω διαδικασία παράγουμε τις ομοιότητες με κάθε λεζάντα που είναι αποθηκευμένη στον server και επιστρέφονται τα αποτελέσματα ταξινομημένα για τον χρήστη, όπως βλέπουμε στην εικόνα 6.4.



Εικόνα 6.4: *Σελίδα αποτελέσματος αναζήτησης*

6.2.3 Σελίδα Gallery

Η τελευταία επιλογή που είναι διαθέσιμη για τον χρήστη είναι αυτή της Gallery. Όταν ο χρήστης κάνει κλικ πάνω στον κουμπί Gallery μεταφέρεται στην σελίδα της γκαλερί, όπου μπορεί να παρατηρήσει όλες τις εικόνες που έχουν αποθηκευτεί στον server μαζί με το caption που έχει παραχθεί για την καθεμία, όπως βλέπουμε και στην εικόνα 6.5.



Εικόνα 6.5: *Σελίδα Gallery*

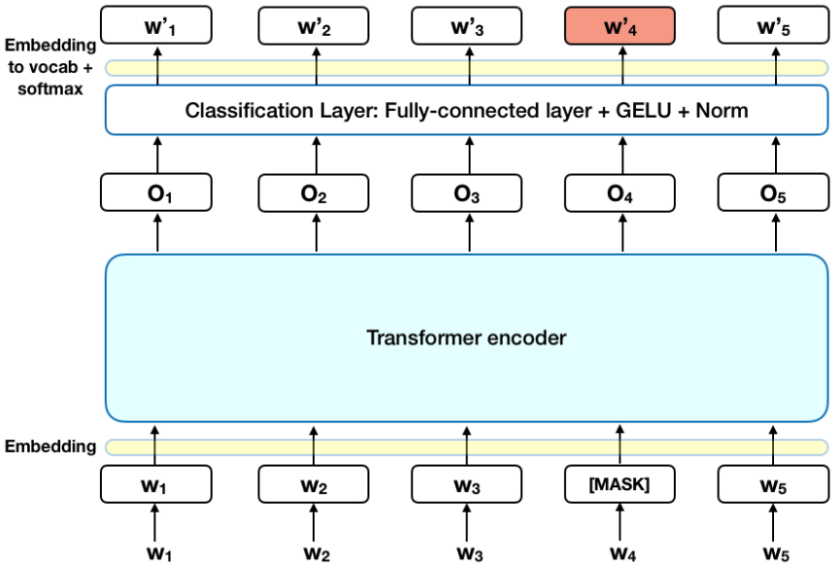
Αυτό όμως δεν αποτελεί το βασικότερο feature αυτής της σελίδας. Σίγουρα είναι χρήσιμο να μπορεί να δει όλες τις εικόνες με τις λεζάντες τους, αλλά θεωρήσαμε ότι ένα πολύ ενδιαφέρον βήμα παραπάνω θα ήταν να μπορεί να δει τις εικόνες που ανήκουν σε συγκεκριμένες κατηγορίες, όπως Animals, Sports, Food ή Transportation. Όμως αυτό που θέλουμε επίσης να κάνουμε είναι να εκμεταλλευόμαστε την πληροφορία του παραγόμενου caption, για αυτό και αναζητήσαμε ένα μοντέλο που θα μπορεί να κατηγοριοποιεί μία πρόταση, με βάση το περιεχόμενό της.

Μοντέλο BERT

Το BERT [73] αποτελεί ένα σύγχρονο μοντέλο που δημοσιεύτηκε από ερευνητές στο Google AI Language. Έχει προκαλέσει μεγάλη ταραχή στην κοινότητα της μηχανικής μάθησης καθώς παρουσιάζει state-of-the-art αποτελέσματα σε μία πληθώρα NLP προβλημάτων, όπως το Question Answering [74] και το Natural Language Inference [75] μεταξύ άλλων.

Η βασική τεχνική καινοτομία του BERT είναι ότι εφαρμόζει αμφίδρομη εκπαίδευση του Transformer [76], ενός διάσημου μοντέλου attention, για γλωσσικά μοντέλα. Αυτή η οπτική είναι αντίθετη με παλιότερες προσπάθειες που βλέπανε μία ακολουθία είτε από αριστερά προς τα δεξιά, ή συνδύαζαν left-to-right και right-to-left εκπαίδευση. Τα αποτελέσματα της δημοσίευσης έδειξαν ότι ένα μοντέλο γλώσσας που είναι αμφίδρομα εκπαιδευμένο μπορεί να έχει μία βαθύτερη συναίσθηση του νοήματος της γλώσσας, από ότι ένα μοντέλο που προπονείται μονόδρομα.

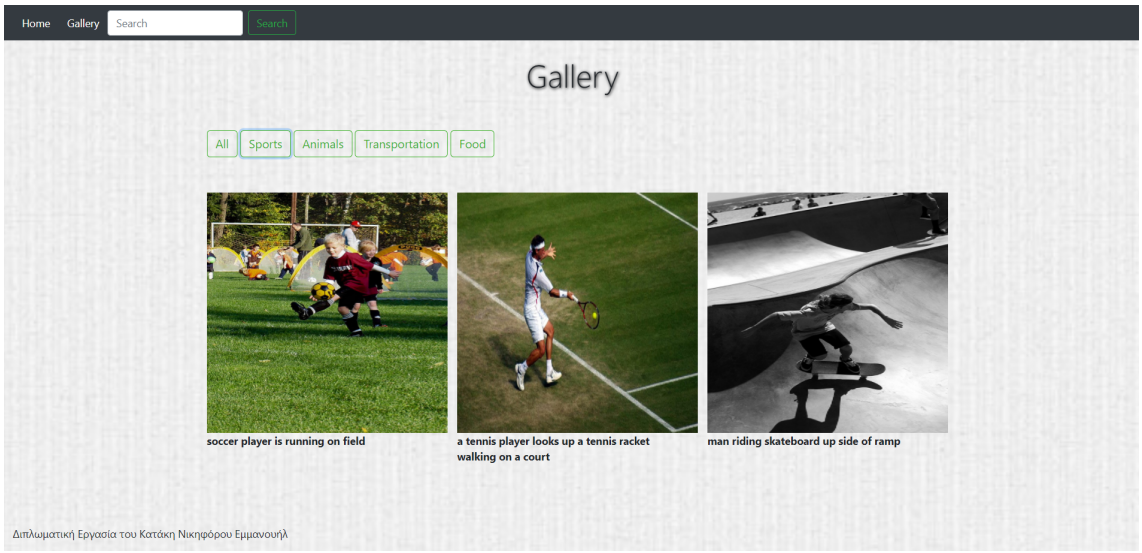
Το μεγάλο πλεονέκτημα του BERT είναι ότι μπορεί να χρησιμοποιηθεί για μια πληθώρα



Εικόνα 6.6: *Βασική αρχιτεκτονική μοντέλου BERT*

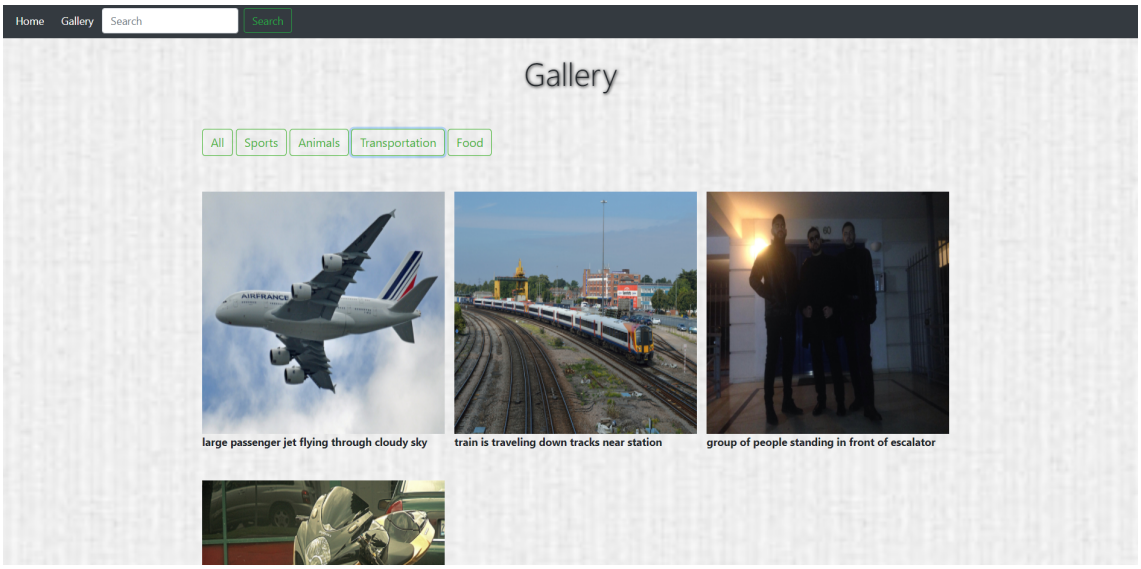
γλωσσικών προβλημάτων προσθέτοντας ένα μόνο μικρό επίπεδο στον πυρήνα του μοντέλου.

Για την δική μας περίπτωση θα χρησιμοποιήσουμε ένα pretrained BERT μοντέλο που θα είναι υπεύθυνο να εντάσσει κάθε εικόνα σε κατηγορίες με βάση την λεζάντα της. Για να αποφύγουμε την παρουσία μίας εικόνας θα την τοποθετούμε σε αυτήν που σημειώνει μεγαλύτερο σκορ με βάση το BERT. Στις εικόνες 6.7 και 6.8 έχουμε τα αποτελέσματα για τις κατηγορίες αθλητισμού και μεταφορών.



Εικόνα 6.7: *Σελίδα Gallery - Κατηγορία Sports*

Στην περίπτωση της κατηγορίας μεταφορών 6.8 παρατηρούμε ότι η τρίτη εικόνα δεν είναι πολύ σχετική, αλλά ταιριάζει περισσότερο σε σχέση με τις άλλες κατηγορίες που έχουμε δώσει. Ένα ακόμα ενδιαφέρον feature είναι ότι αυτές οι κατηγορίες είναι πολύ εύκολο να αλλάξουν. Δηλαδή, σε περίπτωση που ο διαχειριστής της σελίδας θεωρήσει ότι δεν τον ικανοποιούν, μπορεί offline με μία μόνο γραμμή κώδικα να ξανατρέξει την διαδικασία για όποιες κατηγορίες



Εικόνα 6.8: *Σελίδα Gallery - Κατηγορία Transportation*

θέλει εκείνος.

Κεφάλαιο 7

Συμπεράσματα και Μελλοντικές Επεκτάσεις

7.1 Συμπεράσματα

Στην παρούσα διπλωματική παρουσιάσαμε την αρχιτεκτονική ενός μοντέλου που βασίζεται στην μέθοδο του encoder-decoder για την παραγωγή λεκτική περιγραφής μία εικόνας εισόδου, με την προσθήκη ενός μηχανισμού προσοχής (attention mechanism), όπως επίσης και την υλοποίηση μίας διαδικτυακής εφαρμογής, η οποία βασίζεται στην χρήση του μοντέλου μας, αλλά και κάποιων επιπλέον συμπληρωματικών μοντέλων.

Το Image Captioning αποδείχθηκε ένα πολύ δύσκολο πρόβλημα που χρειάστηκε αρκετή μελέτη και κόπο για να αντιμετωπιστεί. Στην ανάπτυξη του μοντέλου μας ασχολούμαστε με την χρήση διαφορετικών αρχιτεκτονικών συνελικτικών νευρωνικών δικτύων, όπως το ResNet και το VGG και προσπαθούμε να τα ενσωματώσουμε στο δικό μας πρόβλημα του Image Captioning. Δοκιμάζουμε διαφορετικές αρχιτεκτονικές RNN για τον αποκωδικοποιητή, όπως αυτή του LSTM και του GRU. Επιπλέον εισάγουμε και μπαίνουμε σε μεγαλύτερο βάθος στους μηχανισμούς προσοχής και ασχολούμαστε με την τεχνική του soft attention και δύο υποπεριπτώσεις του, το Global και το Local attention. Για την προπόνηση του μοντέλου χρησιμοποιούμε την πλατφόρμα ARIS που μας δίνει την δυνατότητα να προπονούμε τα μοντέλα μας σε ισχυρές μονάδες GPU. Όπως είδαμε και στο κεφάλαιο 5.2 παράγουμε ανταγωνιστικά αποτελέσματα με τις state-of-the-art αρχιτεκτονικές, αλλά υπάρχει ακόμα αρκετό μέλλον για την περαιτέρω εξέλιξη του μοντέλου. Ένα από τα πιο σημαντικά συμπεράσματα που μπορούμε να βγάλουμε για το Image Captioning είναι η δυσκολία αξιολόγησης των μοντέλων. Η αξιολόγηση των αποτελεσμάτων συστημάτων παραγωγής φυσικής γλώσσας είναι μία πολύ δύσκολη διαδικασία. Ο βέλτιστος τρόπος να αξιολογηθεί η ποιότητα κειμένου που έχει παραχθεί αυτόματα είναι υποκειμενική αξιολόγηση από γλωσσολόγους, που προφανώς είναι δύσκολο να επιτευχθεί. Για να βελτιστοποιηθεί όμως και η απόδοση του συστήματος, θα πρέπει να βελτιστοποιηθούν και οι μέθοδοι αξιολόγησης. Ένα ακόμα σημαντικό πρόβλημα που παρουσιάστηκε ήταν η ταχύτητα της εκπαίδευσης και του τεσταρίσματος που ξεπερνούσε τις τρεις μέρες.

Πέρα όμως από την ανάπτυξη του μοντέλου, σημαντικό κομμάτι της διπλωματικής ήταν και η ανάπτυξη της διαδικτυακής εφαρμογής που εφαρμόζει το μοντέλο. Κατά την διάρκεια της ανάπτυξης της έγινε κατανοητό, πόσο σημαντική είναι η εφαρμογή νευρωνικών δικτύων σε πραγματικά use cases. Τα νευρωνικά δίκτυα και ο κλάδος της Μηχανικής Μάθησης έχουν ως στόχο να προσφέρουν δυνατότητες που μέχρι πρότινος δεν ήταν υπαρκτές. Με το να μένουν

τα μοντέλα μόνο σε θεωρητικό επίπεδο χάνεται και το μεγαλύτερο κομμάτι του νοήματος, κάτι που μας οδήγησε και στην ανάπτυξη της εφαρμογής και την χρήση διαφόρων μοντέλων εκτός από το δικό μας.

7.2 Μελλοντικές Επεκτάσεις

Το μοντέλο που αναπτύχθηκε στα πλαίσια αυτής της διπλωματικής εργασίας αλλά και το αντίστοιχο web application θα μπορούσε να βελτιωθεί και να επεκταθεί περαιτέρω, ως προς πολλές κατευθύνσεις.

Όσον αφορά το κομμάτι των *νευρωνικών δικτύων*, μπορούν να γίνουν διάφοροι πειραματισμοί για την βελτίωση του υπάρχοντος μοντέλου. Για παράδειγμα, μπορούν να χρησιμοποιηθούν διαφορετικών ειδών feature generators για τις εικόνες, εκτός από τους CNN μηχανισμούς που προτείνουμε παραπάνω. Τέτοιοι μηχανισμοί είναι τα R-CNN, που προσφέρουν εκπληκτικά αποτελέσματα σε προβλήματα object detection [77] ή bottom-up και top-down attention που μπορούν να βοηθήσουν στον καλύτερο εντοπισμό πληροφορίας στην εικόνα εισόδου [47]. Ένας άλλος τρόπος για την περαιτέρω βελτίωση του μοντέλου θα ήταν η εφαρμογή ενός early-stopping αλγορίθμου που θα δίνει την δυνατότητα στο δίκτυο να κάνει πιο σωστά fine-tune τις υπερπαραμέτρους του. Στο επίπεδο του decoding η εφαρμογή του μοντέλου BERT για την παραγωγή των word embeddings θα μπορούσε να οδηγήσει πιθανώς, σε σημασιολογικά καλύτερα αποτελέσματα. Προχωρώντας, μία πιθανή επέκταση θα ήταν να μπορούμε να δίνουμε στον χρήστη την δυνατότητα να ΄κατευθύνει΄ το μοντέλο, δίνοντας εκείνος πληροφορίες για το που θα ήθελε να εστιάσει το δίκτυο πάνω στην εκάστοτε εικόνα. Με αυτό τον τρόπο, θα μπορεί να έχει μεγαλύτερο έλεγχο στην έξοδο του μοντέλου και έτσι, να την προσαρμόζει στις ανάγκες του.

Εν συνεχεία μία πολύ ενδιαφέρουσα επέκταση του μοντέλου αυτού, είναι η εκ νέου παραγωγή εικόνων, μέσω των λεκτικών περιγραφών που κατασκευάζονται από το μοντέλο μας. Το πρόβλημα της σύνθεσης εικόνων μέσα από κείμενο είναι ένα πρόβλημα μηχανικής μάθησης που έχει τραβήξει σε μεγάλο βαθμό την προσοχή της ερευνητικής κοινότητας τα τελευταία χρόνια. Ο συνδυασμός των τομέων του Computer Vision και του Natural Language Processing, όπως είδαμε και μέσω του Image Captioning, μπορεί να οδηγήσει σε μοντέλα με τεράστιες δυνατότητες. Τα Generative Adversarial Networks - GANs, που λειτουργούν συνδυάζοντας δύο νευρωνικά δίκτυα που διαγωνίζονται το ένα το άλλο, αποτελούν την βάση για τα προβλήματα παραγωγής εικόνων και μπορούν να χρησιμοποιηθούν και για παραγωγή εικόνων μέσω κειμένου [78].

Όσον αφορά το κομμάτι του web application οι επεκτάσεις που μπορούν να γίνουν είναι πολλές. Αρχικά, το ήδη υπάρχον application μπορεί να αναπτυχθεί οριζόντια, εντάσσοντας ακόμα περισσότερα μοντέλα ή/και διαφορετικά interfaces. Μία εφαρμογή, η οποία θα μπορεί να στεγάζει και να εκμεταλλεύεται πολλά διαφορετικά μοντέλα, συνδυάζοντας τα το ένα με το άλλο μπορεί να παράγει χρήσιμα εργαλεία για διάφορους τομείς.

Με την τεράστια ανάπτυξη των smartphones ένα από τα πιο ενδιαφέροντα και ίσως σημαντικά βήματα για την συνέχεια της εφαρμογής μας είναι η ανάπτυξη ενός mobile application. Πλέον κάθε άνθρωπος έχει πρόσβαση σε android ή ios smartphones, οπότε γεννιέται και η ανάγκη όσες εργασίες κάνει στον υπολογιστή του, να μπορεί να της κάνει και στο κινητό του.

Ιδανικά, το app θα πρέπει να έχει όλες τις δυνατότητες της διαδικτυακής μας εφαρμογής, δηλαδή την παραγωγή λεκτικής περιγραφής εικόνας μέσω του μοντέλου μας, την κατηγοριοποίηση των εικόνων και την δυνατότητα αναζήτησης στην γκαλερί χρησιμοποιώντας τα pre-trained μοντέλα που εισαγάγουμε στο web app. Ένα βήμα παραπάνω θα ήταν να εισαχθεί και η κάμερα της κινητής συσκευής στην εφαρμογή, ώστε ο χρήστης να είναι σε θέση να βγάλει μία φωτογραφία από το κινητό του και αυτόματα να γίνονται οι παραπάνω διαδικασίες.

Βιβλιογραφία

[1] Mina Khoshdeli, Richard Cong και Bahram Parvin. *Detection of Nuclei in H and E Stained Sections Using Convolutional Neural Networks*. *IEEE International Conference on Biomedical Health Informatics*, 2017.

[2] Y. Bengio, R. Ducharme, P. Vincent και C. Janvin. *A Neural Probabilistic Language Model*. *JMLR.org*, 3, 2003.

[3] T. Mikolov, G. Corrado, K. Chen και J. Dean. *Efficient Estimation of Word Representations in Vector Space*. *Proceedings of Workshop at ICLR*, 1, 2013.

[4] S. Harnad. *The Symbol Grounding Problem*. *CNLS*, 1989.

[5] T. M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997.

[6] F. Chollet. *Deep Learning with Python*. Manning, 2017.

[7] H.B. Barlow. *Unsupervised Learning*. *Neural Computation*, 1(3):295–311, 1989.

[8] R. S. Sutton και A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2014.

[9] C. Olah. *Visual Information Theory*. https://colah.github.io/posts/2015-09-Visual-Information/, 2014.

[10] R. Rojas. *Neural Networks*, κεφάλαιο 7. Springer-Verlag, 1996.

[11] C.M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2011.

[12] M. A. Nielsen. *Neural networks and deep learning*. http://neuralnetworksanddeeplearning.com/, 2018.

[13] R. Zadeh. *The hard thing about deep learning*. https://www.oreilly.com/radar/the-hard-thing-about-deep-learning/, 2016.

[14] M. D. Zeiler. *ADADELTA: An Adaptive Learning Rate Method*. *https://arxiv.org/abs/1212.5701v1*, 2012.

[15] S. Ruder. *An overview of gradient descent optimization algorithms*. https://ruder.io/optimizing-gradient-descent/, 2016.

[16] D. P. Kingma και J. L. Ba. *Adam: A method for stochastic optimization*. *ICLR 2015*, 2017.

[17] P. Munro. *Encyclopedia of Machine Learning*. *Springer US*, σελίδες 73–73, 2010.

[18] S. Hochreiter και J .Schmidhuber. *Long Short-Term Memory*. *Neural Computation*, 9(8):1735–1780, 1997.

[19] D. Bahdanau, K. Cho και Y. Bengio. *Neural machine translation by jointly learning to align and translate*. *ICLR 2015*, 2014.

[20] J. Brownlee. *A Gentle Introduction to Transfer Learning for Deep Learning*. https://machinelearningmastery.com/transfer-learning-for-deep-learning/, 2018.

[21] L. Torrey και J. Shavlik. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, κεφάλαιο Transfer Learning. Information Science Reference - Imprint of: IGI Publishing, 2010.

[22] Moeslund, B. Thomas και E. Graum. *A Survey of Computer Vision-Based Human Motion Capture*. *Elsevier Science Inc.*, 2001.

[23] N. Sharma, V. Jain και A. Mishra. *An Analysis Of Convolutional Neural Networks For Image Classification*. *Procedia Computer Science*, 132:377 – 384, 2018.

[24] R. Verschae και J. Ruizdel Solar. *Object Detection: Current and Future Directions*. *Frontiers in Robotics and AI*, 2:29, 2015.

[25] M. Walker. *Hands On Natural Language Processing with TensorFlow: Concepts and Applications*. CreateSpace Independent Publishing Platform, 2018.

[26] P. Klarith και S. Tanachutiwat. *Thai Clickbait Detection Algorithms Using Natural Language Processing with Machine Learning Techniques*. *2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, 2018.

[27] J.R. Firth. *A Synopsis of Linguistic Theory, 1930-1955*. https://books.google.gr/books?id=T8LDtgAACAAJ, 1957.

[28] R. Kiros, Y. Zhu, R. Salakhutdinov, S. Richard, A. Torralba, R. Urtasun και S. Fidler. *Skip-Thought Vectors*. MIT Press, 2015.

[29] I. Sutskever, O. Vinyalis και Q. V. Le. *Sequence to Sequence Learning with Neural Networks*. *CoRR*, abs/1409.3215, 2014.

[30] F. Gaspari, H. Almaghout και S. Doherty. *A survey of machine translation competences: Insights for translation technology educators and practitioners*. *Perspectives Studies in Translatology*, 23, 2015.

[31] Q. Wu, D. Teney, P. Wang, C. Shen, A. R. Dick και A.van den Hengel. *Visual Question Answering: A Survey of Methods and Datasets*. *CoRR*, abs/1607.05910, 2016.

[32] J. Brownlee. *What is Teacher Forcing for Recurrent Neural Networks?* https://machinelearningmastery.com/teacher-forcing-for-recurrent-neural-networks/, 2017.

[33] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, P. Dollar και C. L. Zitnick. *Microsoft COCO: Common Objects in Context*. *CoRR*, abs/1405.0312, 2014.

[34] D. Hanbin, Z. Liangbo, Z. Feng, Z. Zhengyu, H. Hong, Z. Xiatian και Y. Mao. *Joint COCO and Mapillary Workshop at ICCV 2019 Keypoint Detection Challenge Track Technical Report: Distribution-Aware Coordinate Representation for Human Pose Estimation*. *arXiv*, 2020.

[35] B. A. Plummer, L. Wang, C. Cervantes, M. Chris, J. Caicedo, J. Hockenmaier και S. Lazebnik. *Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models*. *Kluwer Academic Publishers*, 123:74–93, 2017.

[36] M. Hodosh, P. Young και J. Hockenmaier. *Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics*. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[37] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L. Li, D. A. Shamma, M. S. Bernstein και F. Li. *Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations*. *CoRR*, abs/1602.07332, 2016.

[38] D. Bahdanau, K. Cho και Y. Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. *3rd International Conference on Learning Representations*. ICLR 2015, 2015.

[39] M. T. Luong, H. Pham και C. D. Manning. *Effective Approaches to Attention-based Neural Machine Translation*. *CoRR*, 2015.

[40] *Python Tutorial*. https://www.tutorialspoint.com/python/index.htm/, 2019.

[41] F. Chollet. *Keras*. https://github.com/keras-team/keras, 2015.

[42] J. Mao, W. Xu, Y. Yang, J. Wang και A. Yuille. *Explain Images with Multimodal Recurrent Neural Networks*. *ArXiv*, 2014.

[43] J. Mao, W. Xu, Y. Yang, J. Wang και A. Yuille. *Deep captioning with multimodal recurrent neural networks (m-rnn)*. *International Conference on Learning Representations (ICLR)*, 2015.

[44] X. Jia, E. Gavves, B. Fernando και T. Tuytelaars. *Guiding Long-Short Term Memory for Image Caption Generation*. *CoRR*, 2015.

[45] J. Jin, K. Fu, R. Cui, F. Sha και C. Zhang. *Aligning where to see and what to tell: image caption with region-based attention and scene factorization*. *ArXiv*, abs/1506.06272, 2015.

[46] J. Lu, C. Xiong, D. Parikh και R. Socher. *Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning*. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[47] P. Anderson, H. Xiaodong, C. Buehler και L. Zhang. *Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[48] R. Shetty, M. Rohrbach, L. A. Hendricks, M. Fritz και B. Schiele. *Speaking the Same Language: Matching Machine to Human Captions by Adversarial Training. IEEE International Conference on Computer Vision (ICCV)*, 2018.

[49] L. Zhang, F. Sung, F. Liu, T. Xiang, S. Gong, Y. Yang και T. M. Hospedales. *Actor-Critic Sequence Training for Image Captioning. ArXiv*, 2018.

[50] O. Vinyals, A. Toshev, S. Bengio και D. Erhan. *Show and tell: A neural image caption generator. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[51] J. Song, X. Li, L. Gao και H. T. Shen. *Hierarchical LSTMs with Adaptive Attention for Visual Captioning. IEEE Trans. Pattern Anal. Mach. Intell*, 2019.

[52] K. Papineni, S. Roukos, T. Ward και W. J. Zhu. *BLEU: a Method for Automatic Evaluation of Machine Translation.* Association for Computational Linguistics, 2002.

[53] M. Denkowski και A. Lavie. *Meteor Universal: Language Specific Translation Evaluation for Any Target Language. Proceedings of the Ninth Workshop on Statistical Machine Translation.* Association for Computational Linguistics, 2014.

[54] C. Lin. *ROUGE: A Package for Automatic Evaluation of Summaries. Text Summarization Branches Out*, σελίδες 74–81. Association for Computational Linguistics, 2004.

[55] R. Vedantam, C. L. Zitnick και D. Parikh. *CIDEr: Consensus-based Image Description Evaluation. CoRR*, 2014.

[56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens και Z. Wojna. *Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, σελίδες 2818–2826, 2016.

[57] S. Liu και W. Deng. *Very deep convolutional neural network based image classification using small training sample size. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, σελίδες 730–734, 2015.

[58] K. He, X. Zhang, S. Ren και J. Sun. *Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, σελίδες 770–778, 2016.

[59] J. Deng, W. Dong, R. Socher, L. Li και L. Fei-Fei. *ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition*, σελίδες 248–255, 2009.

[60] J. Chung, K. Cho και Y. Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. NIPS 2014 Deep Learning and Representation Learning Workshop*, 2014.

[61] S. Russell και P. Norvig. *Artificial Intelligence: A Modern Approach*, σελίδες 125–126. Prentice Hall Press, 2009.

[62] *Javascript Tutorial*. https://www.w3schools.com/js/DEFAULT.asp, 2019.

[63] *HTML Tutorial*. https://www.w3schools.com/html/, 2020.

[64] *CSS Tutorial*. https://www.w3schools.com/css/, 2019.

[65] T. Roy και N. Richard. *Architectural Styles and the Design of Network-Based Software Architectures*. Διδακτορική Διατριβή, University of California, Irvine, 2000.

[66] M. Rouse. *SOAP (Simple Object Access Protocol)*. https://searchapparchitecture.techtarget.com/definition/SOAP-Simple-Object-Access-Protocol, 2014.

[67] A. Romacher. *Opening the flask*. http://mitsuhiko.pocoo.org/flask-pycon-2011.pdf, 2011.

[68] S. Cohen. *What challenges has Pinterest encoutered with Flask?* https://www.quora.com/What-challenges-has-Pinterest-encountered-with-Flask/answer/Steve-Cohen?srid=hXZd&share=1, 2015.

[69] M. Otto. *Bootstrap from Twitter*. https://blog.twitter.com/developer/en_us/a/2011/bootstrap-twitter.html, 2011.

[70] *Who uses Bootstrap?* https://stackshare.io/bootstrap, 2020.

[71] M. Honnibal. *Introducing Spacy*. https://explosion.ai/blog/introducing-spacy, 2015.

[72] *spaCy Models*. https://spacy.io/models/en.

[73] J. Devlin, M. Chang, K. Lee και K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, σελίδες 4171–4186. Association for Computational Linguistics, 2019.

[74] I. Annamoradnejad, M. Fazli και J. Habibi. *Predicting Subjective Features from Questions on QA Websites using BERT. 2020 6th International Conference on Web Research (ICWR)*, σελίδες 240–244, 2020.

[75] N. Jiang και M.de Marneffe. *Evaluating BERT for natural language inference: A case study on the CommitmentBank. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference*

on Natural Language Processing (EMNLP-IJCNLP), σελίδες 6086–6091. Association for Computational Linguistics, 2019.

[76] R. Agarwal. *What Are Transformer Models in Machine Learning?* https://lionbridge.ai/articles/what-are-transformer-models-in-machine-learning/, 2020.

[77] H. Guangxing, Z. Xuan και L. Chongrong. *Revisiting Faster R-CNN: A Deeper Look at Region Proposal Network*. *ICONIP*, 2017.

[78] S. Reed, A. Zeynep, Y. Xinchenand, B. Schiele L. Logeswaran και H. Lee. *Generative Adversarial Text to Image Synthesis*. *ArXiv*, 2016.