

Υπερανάλυση Εικόνας με χρήση των
Generative Adversarial Networks
και του αλγόριθμου βελτιστοποίησης ILO

ΔΡΙΒΑΣ ΠΑΝΑΓΙΩΤΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων καθηγητής: Λυσίμαχος Κόντης

Ιωάννινα, Φεβρουάριος 2022



**ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA**

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο της ολοκλήρωσης των σπουδών μου στο προπτυχιακό πρόγραμμα του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής της Πολυτεχνικής Σχολής του Πανεπιστημίου Ιωαννίνων.

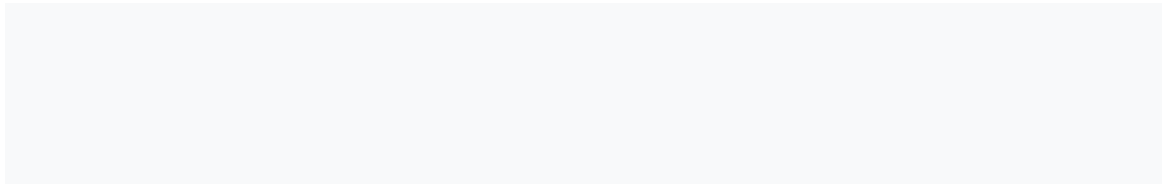
Ευχαριστώ τον επιβλέποντα καθηγητή μου, κύριο Λυσίμαχο Κόντη, για τη καθοδήγηση, τη καλή συνεργασία μας και την υποστήριξη που μου παρείχε καθ' όλη την διάρκεια διεκπεραίωσης της παρούσας διπλωματικής εργασίας.

Περίληψη

Η επεξεργασία εικόνας είναι ένας κλάδος ο οποίος τις τελευταίες δεκαετίες έχει γνωρίσει μεγάλη ανάπτυξη και δημοφιλία. Πρωτίστως λόγω της εξέλιξης της τεχνολογίας, αλλά και του μεγάλου πλήθους των εφαρμογών που αυτή χρησιμοποιείται. Σκοπός λοιπόν αυτής της διπλωματικής εργασίας είναι η παρουσίαση του τομέα της επεξεργασίας εικόνας που ονομάζεται υπερανάλυση εικόνας και βίντεο, μέσα από την μελέτη και την υλοποίηση μεθόδων βαθιάς μάθησης (deep learning), όπως είναι τα συνελκτικά νευρωνικά δίκτυα τα οποία και θα υλοποιηθούν για τον σκοπό της παρούσας εργασίας. Τα συνελκτικά νευρωνικά δίκτυα είναι σημαντικά εργαλεία τα οποία έχουν παρεισφρήσει και έχουν βοηθήσει τα τελευταία χρόνια ολόκληρο τον κλάδο. Επιπλέον για την επίτευξη του στόχου μας, γίνεται επίσης χρήση των adversarial generative models ή εν συντομία GANs. Τα μοντέλα αυτά αποτελούν την δημοφιλέστερη προσέγγιση του deep generative modeling χρησιμοποιώντας τη βαθιά μάθηση και τα συνελκτικά νευρωνικά δίκτυα. Πιο συγκεκριμένα τώρα το GAN που θα παρουσιαστεί θα είναι το StyleGAN₂, που σε συνδυασμό με τον αλγόριθμο βελτιστοποίησης Intermediate Layer Optimization θα μας δείξει πως μπορούμε να λύσουμε με πολύ αποδοτικό τρόπο το πρόβλημα της υπερανάλυσης εικόνας. Τέλος θα δούμε και πως αυτή η λύση αντιπετωπίζει και άλλα inverse problems, όπως είναι αυτό του inpainting.

Abstract

Image processing is an industry that has experienced great growth and popularity in recent decades. Primarily due to the evolution of technology, but also the large number of applications used. The purpose of this diplomatic work is to present the field of image processing called super-resolution of image and video, through the study and implementation of deep learning methods, such as convolutional neural networks, which will be implemented for purpose of this work. Convolutional neural networks are important tools tha have penetrated and helped the whole industry in recent years. In addition, to achieve our goal, also we use adversarial generative models or GANs for short. These models are most popular approach to deep generative modeling using deep learning and convolutional networks. More specifically, now the GAN that will be presented will be StyleGAN2, which in combination with the optimization algorithm Intermediate Layer Optimization will show us how can solve the problem of super resolution of image in a very efficient way. Finall, we will se how this solution addresses other inverse problems, such as inpainting.



Πίνακας Περιεχομένων

Κεφάλαιο 1. Εισαγωγή	1
1.1 Πεδίο της εργασίας	1
1.2 Δομή της εργασίας	1
Κεφάλαιο 2 Θεωρία	3
2.1 Βαθιά Μάθηση	3
2.2 Τεχνητά Νευρωνικά Δίκτυα	4
2.2.1 Συνάρτηση Ενεργοποίησης	6
2.2.2 Συνάρτηση Κόστους	8
2.2.3 Αλγόριθμος Backpropagation	9
2.2.4 Εκπαίδευση ΤΝΔ	10
2.2.5 Αρχιτεκτονική των ΤΝΔ	11
2.3 Συνελικτικά Νευρωνικά Δίκτυα	12
2.3.1 Συνέλιξη	12
2.3.2 Συγκεντρωτικό Επίπεδο	13
2.3.3 Πλήρως Συνδεδεμένο Επίπεδο	14
2.3.4 Επίπεδο Εξόδου	14
2.4 Generative Adversarial Networks	15
2.4.1 Μοντέλο Γεννήτριας και Διαχωριστή	15
2.4.2 Feedback	17
2.4.3 StyleGAN2	18
2.4.3.1 Γεννήτρια του StyleGAN2	19
2.4.3.2 Λύση του Προβλήματος των Artifacts	21
2.4.3.3 Προοδευτική Ανάπτυξη	22
2.5 Intermediate Layer Optimization	22
2.5.1 Σύνθεση Αλγορίθμου	23

2.5.2 Προσαρμογή του ILO με το StyleGAN2	24
2.5.3 Συναρτήσεις κόστους	24

Κεφάλαιο 3 Υλοποίηση και Πειράματα

3.1 Υλοποίηση	26
3.2 Πειράματα και αποτελέσματα	27
3.2.1 Επιπρόσθετα Πειράματα	32

Κεφάλαιο 4

4.1 Συμπεράσματα και προεκτάσεις	37
--	----

Κεφάλαιο 1

Εισαγωγή

1.1 Πεδίο της εργασίας

Υπερανάλυση εικόνας ονομάζεται η διαδικασία κατά την οποία γίνεται ανάκτηση μίας εικόνας υψηλής ανάλυσης (high resolution) από μία εικόνα χαμηλής ανάλυσης (low resolution). Πρόκειται για ένα πεδίο το οποίο έχει πάρα πολλές εφαρμογές σε πολλούς κλάδους, από την εφαρμογή σε applications που αφορούν την ψυχαγωγία, μέχρι την χρησιμοποίηση του στον τομέα της ιατρικής (μαγνητική τομογραφία), αλλά και στον τομέα των επικοινωνιών και των μαζικών μέσων ενημέρωσης.

Μία επιπλέον προώθηση τώρα σε αυτόν τον κλάδο έχει δώσει τις τελευταίες δεκαετίες, η μεγάλη ανάπτυξη και πρόοδος στις τεχνικές της βαθιάς μάθησης, οι οποίες χρησιμοποιούνται πλέον σε μεγάλο βαθμό στην επεξεργασία εικόνας. Πλέον έτσι έχουμε στη διάθεση μας μοντέλα βαθιάς μάθησης που είναι βασισμένα πάνω στο super resolution και πετυχαίνουν πολύ καλές επιδόσεις. Τέτοια είναι τα CNN (συνελικτικά νευρωνικά δίκτυα), αλλά και τα πιο σύγχρονα GANs με τα οποία και θα ασχοληθούμε στην παρούσα εργασία.

1.2 Δομή της εργασίας

Η εργασία αποτελείται από 4 κεφάλαια. Στο δεύτερο κεφάλαιο θα παρουσιαστούν οι τεχνολογίες και οι αλγόριθμοι τα οποία χρησιμοποιούνται για την λύση του προβλήματος. Δηλαδή θα δείξουμε γενικά τι είναι οι μέθοδοι βαθιάς μάθησης, τι είναι και πως δουλεύουν τα CNN, αλλά και τα GANs και πιο συγκεκριμένα το StyleGAN2. Ακόμα θα παρουσιαστεί πως υλοποιήθηκε και τι προσφέρει ο αλγόριθμός Intermediate Layer Optimization στο πρόβλημα του super resolution και πως αυτός συνδυάστηκε με το StyleGAN2.

Στο τρίτο κεφάλαιο θα δείξουμε την υλοποίηση ILO σε συνδυασμό με το StyleGAN2. Επίσης θα δείξουμε τα πειράματα που πραγματοποιήσαμε και τα αποτελέσματα αυτών, πειράζοντας κατά διαστήματα διάφορες

παραμέτρους παρατηρώντας τις αλλαγές που επέφεραν, και πως ή αν αυτά οδήγησαν σε ένα καλύτερο αποτέλεσμα. Ακόμα θα δούμε πως το συγκεκριμένο μοντέλο που έχει εκπαιδευτεί να παράγει πρόσωπα αντιδρά και τι αποτελέσματα μας δίνει όταν το βάλουμε να παράξει κάποια διαφορετική κατηγορία εικόνων.

Τέλος στο τέταρτο κεφάλαιο θα δούμε τι προσέφερε αυτή η εργασία στο πρόβλημα του super resolution. Επίσης θα δούμε μελλοντικές επεκτάσεις που μπορούν να γίνουν.

Κεφάλαιο 2

Βαθιά Μάθηση

2.1 Βαθιά Μάθηση

Με τον όρο βαθιά μάθηση αναφερόμαστε στην τεχνική μηχανικής μάθησης και τεχνητής νοημοσύνης με την οποία ένας υπολογιστής διδάσκεται να μαθαίνει με τον ίδιο τρόπο με τον οποίο μαθαίνει ένας άνθρωπος, δηλαδή μέσω παραδειγμάτων. Ακριβώς όπως ένας άνθρωπος, ένα μικρό παιδί, θα μάθει να αναγνωρίζει π.χ. ένα τραπέζι, έτσι και ένας υπολογιστής είναι σε θέση πλέον να προσημειώνει την ίδια διαδικασία. Ένα παιδί το οποίο οι γονείς του θα του υποδεικνύουν αν το αντικείμενο που του δείχνουν είναι ή όχι ένα τραπέζι, θα το βοηθήσουν να αναγνωρίζει σιγά σιγά και χρησιμοποιώντας την γνώση που έχει χτίσει σταδιακά, χτίζοντας έτσι ασυνείδητα μια ιεραρχία, τα χαρακτηριστικά που απαρτίζουν ένα τραπέζι. Παρόμοια ένα πρόγραμμα σε έναν υπολογιστή χρησιμοποιώντας και αυτό μια δομή ιεραρχίας, της οποίας ο κάθε αλγόριθμος χρησιμοποιεί αυτά που μαθαίνει σταδιακά, όπως το παιδί, για να δημιουργήσει μία έξοδο, ενώ όλο αυτό επαναλαμβάνεται έως ότου η έξοδος αποκτήσει μία αποδεκτή ακρίβεια. Ακρίβεια που στις μέρες μας έχουμε καταφέρει να έχει φτάσει σε ένα πολύ υψηλό επίπεδο. Για να φτάσουμε όμως σε αυτό το επίπεδο ακρίβειας χρειάζονται μεγάλες βάσεις δεδομένων αλλά και επεξεργαστικής ισχύς, λόγοι οι οποίοι εξηγούν και γιατί η βαθιά μάθηση άρχισε να αξιοποιείται τα τελευταία χρόνια, ενώ είχε παρουσιαστεί σε θεωρητικό επίπεδο ήδη από το 1980.

Ένα από τα πλεονεκτήματα της τώρα είναι ότι σε αντίθεση με την παραδοσιακή μηχανική μάθηση στην οποία πρέπει να είμαστε πολύ προσεκτικοί και συγκεκριμένοι όταν λέμε στον υπολογιστή τι χρειάζεται να αναζητήσει για να αποφασίσει αν π.χ. μια εικόνα περιέχει ένα τραπέζι, διαδικασία η οποία ονομάζεται εξαγωγή χαρακτηριστικών και η οποία εξαρτάται πλήρως από τις ικανότητες του εκάστοτε προγραμματιστή να μπορεί να υλοποιήσει με ακρίβεια τα παραπάνω, η βαθιά μάθηση έχει απαλλαχτεί από αυτό, είναι δηλαδή μία μέθοδος χωρίς επίβλεψη (unsupervised), καθώς το ίδιο το πρόγραμμα έχει πλέον τη δυνατότητα να το κάνει αυτό, γεγονός που προσδίδει στην βαθιά μάθηση ταχύτητα αλλά και το πιο σημαντικό ακρίβεια.

Ας δούμε τώρα με λίγα λόγια πως λειτουργούν όλα αυτά. Αρχικά στο πρόγραμμα θα του δοθεί ως είσοδος ένα σύνολο από εικόνες ή αλλιώς τα δεδομένα της εκπαίδευσής του, το οποίο θα περιέχει εικόνες με τραπέζια και θα το βοηθήσει να εκπαιδευτεί ώστε να μάθει να τα αναγνωρίζει. Μέσα από την διαδικασία της εκπαίδευσης το πρόγραμμα θα δημιουργήσει ένα μοντέλο πρόβλεψης και θα μπορεί έτσι να αναγνωρίσει σε μια εικόνα αν σε αυτή περιέχεται ένα τραπέζι. Όσο πιο πολλές φορές γίνεται αυτή η διαδικασία, τόσο περισσότερο περίπλοκο αλλά και πιο ακριβές θα γίνεται το μοντέλο μας.

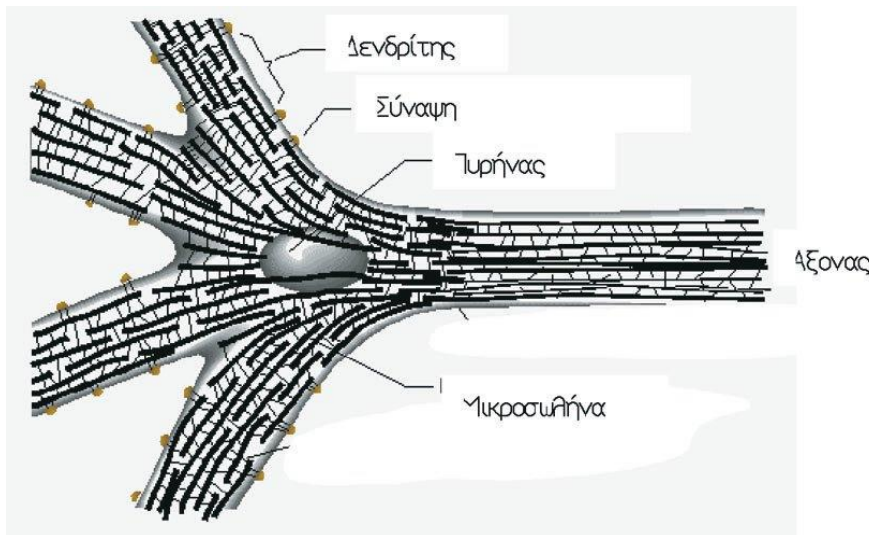
Ένας από τους πιο γνωστούς αλγορίθμους μηχανικής μάθησης, που στηρίζει τα περισσότερα μοντέλα βαθιά μάθησης, είναι τα τεχνητά νευρωνικά δίκτυα. Η πιο δημοφιλής μορφή αυτών των δικτύων με την οποία και θα ασχοληθούμε στην παρούσα εργασία είναι τα συνελκτικά νευρωνικά δίκτυα.

2.2 Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα αποτελούν μια προσομοίωση του ανθρώπινου νευρικού κεντρικού συστήματος, προσπαθώντας με αυτόν τον τρόπο να συνδυάσουμε τον ανθρώπινο τρόπο σκέψης με τον μαθηματικό. Έχουμε φτάσει έτσι στο να χρησιμοποιούμε ρήματα που μέχρι πρότινος χαρακτηρίζαν ανθρώπινες ενέργειες, έτσι τώρα μπορούμε να πούμε ότι ένα ΤΝΔ μαθαίνει και εκπαιδεύεται.

Για να κατανοήσουμε καλύτερα αυτήν την αντιστοιχία, αλλά και τα ίδια τα ΤΝΔ ας δούμε σύντομα τι είναι ένα βιολογικό νευρωνικό δίκτυο. Ένα βιολογικό νευρωνικό δίκτυο απαρτίζεται από ένα μεγάλο αριθμό μονάδων, τους οποίους ονομάζουμε νευρώνες. Η κύρια δομή ενός νευρώνα ονομάζεται σώμα μέσα στο οποίο βρίσκεται ο πυρήνας του κυττάρου αυτού. Ένας νευρώνας αποτελεί το βασικό δομικό στοιχείο του ανθρώπινου εγκεφάλου. Όλα αυτά τα δομικά στοιχεία συνδέονται μεταξύ τους με συνδέσεις. Το σύνολο των νευρώνων και των συνδέσεων τους, οι οποίες ονομάζονται συνάψεις αποτελούν ένα νευρωνικό δίκτυο και όλο αυτό το πλήθος των νευρωνικών δικτύων, απαρτίζουν το Κεντρικό Νευρικό Σύστημα του ανθρώπινου οργανισμού. Μέσω τώρα αυτών των νευρώνων, ο εγκέφαλος μεταφέρει πληροφορίες στα υπόλοιπα συστήματα του οργανισμού. Κάθε νευρώνας έχει περισσότερες από μια εισόδους και μια εξόδο μέσω των οποίων οι νευρώνες ανταλλάσσουν μεταξύ τους πληροφορίες. Αυτά τα σημεία εισόδου και εξόδου σε έναν νευρώνα ονομάζονται δενδριτές και άξονας αντίστοιχα. Οι πληροφορίες που διαχέονται μέσα στο Κεντρικό Νευρικό Σύστημα μέσω των νευρώνων έχουν την μορφή ηλεκτρικών παλμών, και η συνολική ποσότητα τους την οποία λαμβάνει ένας νευρώνα στην είσοδο του την ονομάζουμε συναπτικό βάρος.

Ένας νευρώνας θα βρίσκεται πάντα σε δύο καταστάσεις, είτε θα είναι ενεργός είτε θα είναι μη-ενεργός. Ενεργός καθίσταται όταν πυροδοτεί ένα ηλεκτρικό σήμα, έτσι ώστε να μεταδώσει τις πληροφορίες που έχουν έρθει σε αυτόν από κάποιο γειτονικό του νευρώνα. Ενώ όταν είναι μη-ενεργός δεν κατέχει κάποια πληροφορία προς μετάδοση, άρα δεν πυροδοτεί κάποιο ηλεκτρικό σήμα. Μέσα λοιπόν από τις εισόδους του, τους δενδρίτες, ο νευρώνας παίρνει όλες αυτές τις πληροφορίες, όλο αυτά τα ηλεκτρικά σήματα και είναι έτοιμος να τα μεταφέρει εάν όλη αυτή η ποσότητα της πληροφορίας είναι μεγαλύτερη από ένα κάτω όριο. Όσο πιο μεγάλη, πιο έντονη είναι η σύνδεση μεταξύ δύο νευρώνων οι οποίοι ανταλλάσσουν μια πληροφορία τόσο πιο μεγάλη είναι η συνεισφορά αυτής στο σύνολο του όγκου της πληροφορίας που έχει ο νευρώνας.

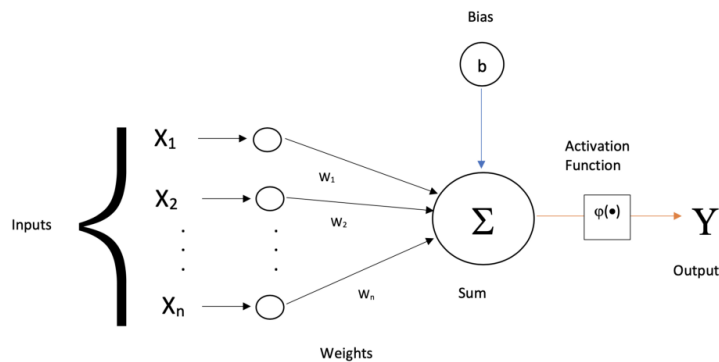


Εικόνα 2.1 : Το μοντέλο του βιολογικού νευρώνα.

Ένα τεχνητό νευρωνικό δίκτυο λοιπόν αποτελείται από πολλούς νευρώνες οι οποίοι έχουν και αυτοί αντίστοιχα εισόδους και μία έξοδο, μέσω των οποίων ένας νευρώνας λαμβάνει και στέλνει σήματα εισόδου και εξόδου. Σε κάθε μία από τις εισόδους ενός νευρώνα αντιστοιχεί και ένα βάρος, το οποίο ουσιαστικά μας δείχνει πόσο στενά είναι συνδεδεμένοι αυτοί οι δύο νευρώνες που συνδέονται με αυτό το βάρος. Κατά την διάρκεια της εκπαίδευσης ενός νευρωνικού δικτύου οι τιμές αυτών αλλάζουν, ενώ συνήθως τιμές που παίρνουν τα βάρη βρίσκονται στο διάστημα -1 έως 1 . Ανάλογα τώρα με το αν τα βάρη έχουν αρνητικές ή θετικές τιμές, αυτό είναι που καθιστά ένα νευρώνα ενεργό ή μη-ενεργό, δηλαδή αν έτοιμος να μεταδώσει ή να μην μεταδώσει κάποια πληροφορία.

2.2.1 Συνάρτηση ενεργοποίησης

Όταν λοιπόν ένας νευρώνας καταστεί ενεργός, αρχικά θα αθροίσει τα σήματα εισόδου τα οποία επηρεάζονται από τα βάρη που έχει κάθε είσοδος του, ενώ στην συνέχεια υπολογίζει μια συνάρτηση, η οποία ονομάζεται συνάρτηση ενεργοποίησης f . Υπολογίζοντας την τιμή αυτής της συνάρτησης με βάση το άθροισμα των εισόδων του και συγκρίνοντας την με ένα κάτω όριο, το οποίο είναι χαρακτηριστικό για τον συγκεκριμένο νευρώνα, θα υπολογίσει την έξοδο του και θα την στείλει σε κάποιον γείτονα του αν και μόνο αν αυτή η τιμή είναι μεγαλύτερη από αυτό το κατώφλι.



Εικόνα 2.2: Το μοντέλο του τεχνητού νευρώνα

Ας περιγράψουμε τώρα μερικές από τις πιο γνωστές και διαδεδομένες συναρτήσεις ενεργοποίησης, οι οποίες χρησιμοποιούνται στα ΤΝΔ:

1. Βηματική Συνάρτηση:

Ο γενικός τύπος είναι:

$$g(x) = a \text{ αν } x < 0 \text{ ή } g(x) = b \text{ αν } x > 0 \quad (2.1)$$

Οι πιο συνηθισμένες τιμές των a και b είναι $a = 0$ και $b = 0$ ή $a = -1$ και $b = 1$. Το μεγάλο μειονέκτημα αυτής της συνάρτησης είναι ότι η παραγωγός της μηδενίζεται και για αυτό το λόγο δεν είναι χρήσιμη για την ενεργοποίηση των νευρώνων σε ένα ΤΧΔ.

2. Σιγμοειδής Συνάρτηση :

Ο γενικός τύπος της σιγμοειδούς συνάρτησης είναι:

$$s(x) = L / (1 + e^{-k \cdot (x - x_0)}) \quad (2.2)$$

Το L δηλώνει την μέγιστη τιμή της συνάρτησης, το k καθορίζει την καμπυλότητα της συνάρτησης και x_0 είναι το κεντρικό σημείο της καμπύλης της συνάρτησης. Το σύνολο τιμών της είναι το $[0, L]$ ενώ είναι γνησίως αύξουσα συνάρτηση και διαφορίσιμη στο πεδίο ορισμού της εξαλείφοντας έτσι το μειονέκτημα της βηματικής. Οι πιο ευρέως χρησιμοποιούμενες τιμές των παραμέτρων της είναι $L=1$, $x_0=0$, $k=1$, με τον τύπο να γίνεται $s(x) = 1 / (1 + e^{-x})$ και το σύνολο τιμών της να κλείνει στο $[0, 1]$. Αποτελεί έτσι την πιο δημοφιλή συνάρτηση ενεργοποίησης, καθώς μέσα από το σύνολο τιμών αντιστοιχίζονται οι δύο καταστάσεις ενός νευρώνα, δηλαδή ο όταν είναι ανενεργός και 1 όταν είναι ενεργός. Μειονέκτημα της τώρα είναι ότι όταν στο όριο της συνάρτησης που η κλίση γίνεται πολύ μεγάλη, η σιγμοειδής γίνεται βηματική.

3. Υπερβολική Εφαπτομένη Συνάρτηση :

Ο γενικός τύπος της συνάρτησης είναι:

$$\tanh(x) = (e^{ax} - e^{-ax}) / (e^{ax} + e^{-ax}) \quad (2.3)$$

Αποτελεί και αυτή μία διαφορίσιμη στο πεδίο ορισμού της συνάρτηση, αποβάλλοντας έτσι το μειονέκτημα της βηματικής συνάρτησης ενεργοποίησης. Παρά το γεγονός αυτό όμως η υπερβολική εφαπτομένη έχει αρχίσει και εγκαταλείπεται, αφού είναι μια συνάρτηση που οδηγείται σε κορεσμό.

4. Ανορθωμένη Γραμμική Συνάρτηση:

Ο γενικός τύπος της συνάρτησης είναι:

$$f(x) = \max(0, x) \quad (2.4)$$

Όπου το x είναι η είσοδος του νευρώνα. Τα τελευταία χρόνια αποτελεί την δημοφιλέστερη συνάρτηση ενεργοποίησης για τα βαθιά νευρωνικά δίκτυα. Η συνάρτηση ράμπας (ReLU), όπως αλλιώς αποκαλείται, δεν οδηγείται σε κορεσμό, ενώ αποτελεί μια απλή υπολογιστικά συνάρτηση και πολύ αποτελεσματική. Παρόλα αυτά το βασικό της μειονέκτημα είναι ότι σε περίπτωση εμφάνισης αρνητικού αθροίσματος εξόδου, τότε ο νευρώνας όπως φαίνεται και από τον γενικό τύπο της συνάρτησης θα παραμένει ανενεργός, με τελικό αποτέλεσμα να μην εκπαιδευτεί το νευρωνικό δίκτυο (dying neural networks).

5. Διαρρέουσα Ανορθωμένη Γραμμική (Leaky ReLU):

Ο γενικός τύπος της συνάρτησης είναι:

$$f(x)=\max(0.01x, x) \quad (2.5)$$

Η δημιουργία αυτής της συνάρτησης ήρθε να λύσει το πρόβλημα της συνάρτησης ράμπας(ReLU) με τα νεκρά νευρωνικά δίκτυα. Προσθέτοντας λοιπόν στην συνάρτηση την μεταβλητή x , πολλαπλασιαζόμενη από την σταθερά 0.01 καταφέραμε να δίνει έξοδο και σε αρνητικές τιμές.

2.2.2 Συνάρτηση Κόστους(Loss Function)

Η συνάρτηση κόστους έχει στόχο να μας δείξει την απόκλιση ανάμεσα στην έξοδο ενός ΤΝΔ καθώς εκπαιδεύεται και την πραγματική εικόνα (target) που έχει δοθεί στο δίκτυο μαζί με την είσοδό του. Κατά την διάρκεια της εκπαίδευσης το δίκτυο μαθαίνει να μειώνει αυτή την απόκλιση και να πλησιάζει κοντά στην πραγματική αλήθεια. Ας δούμε τώρα κάποιες συναρτήσεις κόστους :

1) Μέσο τετραγωνικό σφάλμα(MSE):

Όπως δηλώνει και το όνομα της, είναι ο μέσος όρος της διαφοράς στο τετράγωνο μεταξύ της πραγματικής τιμής και της τιμής που παράγει το δίκτυο. Μειονέκτημα της αποτελεί το γεγονός, ότι λόγω της ύψωσης των τιμών στο τετράγωνο, η συνάρτηση τείνει να τιμωρεί τιμές οι οποίες έχουν μεγάλη απόκλιση από την πραγματική σε σύγκριση με αυτές που έχουν μικρότερη απόκλιση. Ο τύπος της είναι:

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (2.6)$$

Όπου y_i είναι η πραγματική τιμή και \hat{y}_i καπέλο η προβλεπόμενη.

2) Cross Entropy Loss

Αποτελεί την πιο κοινή συνάρτηση απώλειας, κατά την οποία η απώλεια μειώνεται καθώς η προβλεπόμενη τιμή τείνει προς την πραγματική. Ο τύπος της είναι :

$$CrossEntropyLoss = -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (2.7)$$

Είναι επίσης γνωστή ως δυαδική διασταυρούμενη εντροπία καθώς η έξοδός της είναι 0 ή 1. Το κύριο χαρακτηριστικό αυτής είναι ότι τιμωρεί τις εξόδους που είναι αισιόδοξες αλλά λανθασμένες.

2.2.3 Αλγόριθμος Backpropagation

Σκοπός του αλγόριθμου Backpropagation είναι ελαχιστοποιήσει το σφάλμα που προκύπτει από την εκάστοτε συνάρτηση κόστους, χρησιμοποιώντας τον αλγόριθμο Gradient Descent, ο οποίος είναι ένας αλγόριθμος βελτιστοποίησης που ενημερώνει επαναληπτικά τις παραμέτρους που μπορούν να μάθουν σε ένα δίκτυο, δηλαδή τα βάρη και του πυρήνες-φίλτρα. Ακολουθώντας έτσι μία προς τα πίσω διαδικασία, έχοντας το σφάλμα από την συνάρτηση κόστους, ανανεώνει τις τιμές του νευρωνικού δικτύου. Μέσα λοιπόν από τον κανόνα της αλυσιδωτής παραγωγίσης, υπολογίζεται η παράγωγος της συνάρτησης σφάλματος ως προς τα βάρη κάθε νευρώνα, τα οποία ανανεώνονται με βάση τον τύπο:

$$w_i^+ = w_i - \text{step} * \partial E_{\text{total}} / \partial w_i \quad (2.8)$$

όπου $\partial E_{\text{total}} / \partial w_i$ είναι η μεταβολή του σφάλματος και το step είναι μια σταθερά που ονομάζεται ρυθμός ανανέωσης των βαρών.

2.2.4 Εκπαίδευση ΤΝΔ

Τα τεχνητά νευρωνικά δίκτυα κατασκευάστηκαν έτσι ώστε να προσομοιώνουν ανθρώπινα χαρακτηριστικά και να μας βοηθούν έτσι να αντιμετωπίζουμε προβλήματα, την λύση των οποίων θα είναι σε θέση να την αντιμετωπίσουν μόνοι τους. Αυτό λοιπόν το αποτέλεσμα επιτυγχάνεται μέσα από την διαδικασία της εκπαίδευσης ενός ΤΝΔ. Μέσα λοιπόν από την διαδικασία της εκπαίδευσης του δικτύου, αυτό που ουσιαστικά συμβαίνει και κάνει το δίκτυο να βελτιώνεται και να φτάνει μια αποδεκτή ακρίβεια, είναι οι αλλαγές που πραγματοποιούνται στα βάρη των εισόδων κάθε νευρώνα. Ο τρόπος με τον οποίο πραγματοποιούνται αυτές οι αλλαγές δεν είναι μοναδικός, αλλά υπάρχουν διάφορες μέθοδοι που επιτελούν αυτήν την δουλειά.

Οι μέθοδοι αυτοί είναι:

- 1) Η επιβλεπόμενη μάθηση

2) Η μη επιβλεπόμενη μάθηση

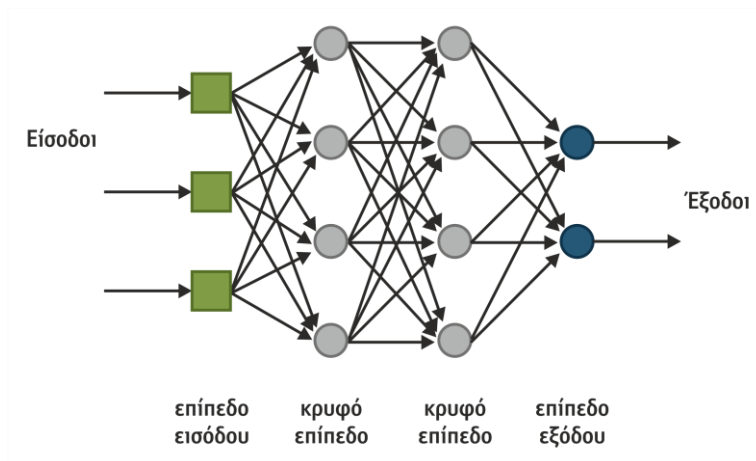
1) Πρόκειται για την πιο ευρέως χρησιμοποιούμενη μέθοδο για την εκπαίδευση ενός νευρωνικού δικτύου. Κατά την χρησιμοποίηση της επιβλεπόμενης μεθόδου ο προγραμματιστής πρέπει να δίνει στο δίκτυο δεδομένα εισόδου τα οποία έχουν επισημανθεί για μια συγκεκριμένη έξοδο. Το νευρωνικό δίκτυο λοιπόν θα εκπαιδεύεται και θα αλλάζει τα βάρη του με γνώμονα τις διαφορές που εντοπίζει ανάμεσα στις εξόδους που παράγει το ίδιο και τις αντίστοιχες επισημασμένες, επιτρέποντάς του έτσι να παράγει μετά την εκπαίδευση του εξόδους για δεδομένα που βλέπει πρώτη φορά, δηλαδή δεδομένα που δεν έχουν επισημανθεί στο δίκτυο.

2) Με την μη επιβλεπόμενη μέθοδο μάθησης, ο προγραμματιστής δεν χρειάζεται να επεμβαίνει, όπως στην περίπτωση της επιβλεπόμενης μάθησης, αφού οι είσοδοι-δεδομένα που παρέχονται στο σύστημα δεν χρειάζεται να συνοδεύονται από τις αντίστοιχες εξόδους τους. Με αυτήν την μέθοδο το νευρωνικό δίκτυο κατά τη διάρκεια της εκπαίδευσης προσπαθεί να βρει κάποια μοτίβα στις εισόδους που του έχουν δοθεί, τα οποία θα το οδηγήσουν να βγάλει εξόδους που θα προσεγγίζουν τις αντίστοιχες εισόδους. Με αυτόν τον τρόπο λοιπόν το δίκτυο διορθώνει τα σφάλματα του και αλλάζει τα βάρη του.

2.2.5 Αρχιτεκτονική των ΤΝΔ

1) Δίκτυα πρόσθιας τροφοδότησης :

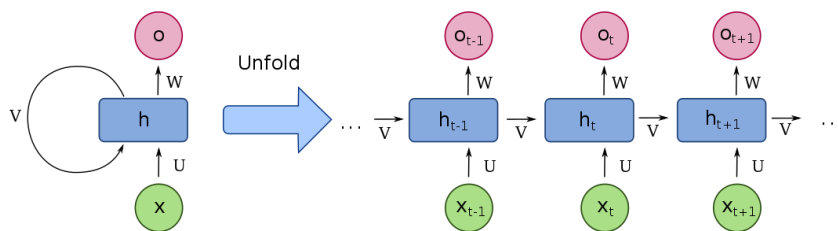
Αυτή η αρχιτεκτονική των τεχνητών δικτύων, αποτελείται από ένα επίπεδο εισόδου και ένα εξόδο, και κανένα ή πολλά ενδιάμεσα-κρυφά επίπεδα. Κάθε επίπεδο εισόδου ή εξόδου αποτελείται από τους κόμβους-νευρώνες του, όπως και τα κρυφά επίπεδα, οι νευρώνες των οποίων ονομάζονται κρυφοί. Η ύπαρξη των κρυφών επιπέδων μας προσφέρει την δυνατότητα της καλύτερης αφομοίωσης των πληροφοριών λόγω των περισσότερων συνδέσεων – συνάψεων, αλλά και της καλύτερης αλληλεπίδρασης που δημιουργείται. Το επίπεδο της εισόδου τώρα αποτελεί το διάνυσμα εισόδου στο δίκτυο, χωρίς να επιτελεί κάποια υπολογιστική εργασία. Κάθε κρυφό επίπεδο χρησιμοποιεί την έξοδο του προηγούμενου επιπέδου σαν δικιά του είσοδο, μέχρις ότου η πληροφορία μεταφερθεί στο επίπεδο εξόδου όπου και θα πραγματοποιηθεί ο υπολογισμός του διανύσματος εξόδου του δικτύου.



Εικόνα 2.3 : Δίκτυο πρόσθιας τροφοδότησης

2) Αναδρομικά δίκτυα:

Η κύρια διαφορά αυτής της μορφής δικτύων σε σχέση με τα πρόσθιας τροφοδότησης, είναι η ύπαρξη βρόχων ανάδρασης (feedback loop). Με την συγκεκριμένη προσθήκη ένας νευρώνας το δικτύου έχει πλέον την δυνατότητα να λαμβάνει σαν είσοδο την ίδια του την έξοδο. Δημιουργείται λοιπόν η έννοια της μνήμης στο δίκτυο, βοηθώντας το έτσι στην ικανότητα του να μαθαίνει, αλλά και στο κομμάτι της απόδοσης του. Επιπλέον οι βρόχοι περιλαμβάνουν τη χρήση στοιχείων καθυστέρησης σε κάποιες διασυνδέσεις που επιτρέπουν στο δίκτυο να αποκτήσει μια μη γραμμική δυναμική συμπεριφορά, ικανότητα σημαντική για την αντιμετώπιση συγκεκριμένων εφαρμογών.



Εικόνα 2.4: Αναδρομικό Δίκτυο

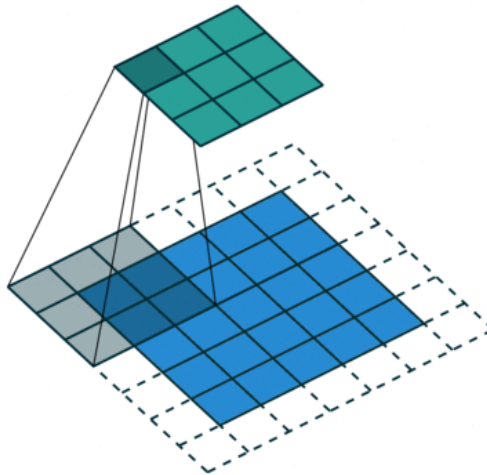
2.3 Συνελικτικά Νευρωνικά Δίκτυα:

Τα συνελικτικά νευρωνικά δίκτυα είναι ένα είδος τεχνητών νευρωνικών δικτύων, και άρα ισχύουν τα βασικά χαρακτηριστικά που έχουμε πει παραπάνω. Εκπαιδεύονται με τον ίδιο τρόπο και τα διέπουν οι ίδιες αρχές. Τα τελευταία χρόνια έχουν κυριαρχήσει στα προβλήματα της υπολογιστικής όρασης, όπως είναι η υπερανάληψη εικόνας, με την οποία ασχολούμαστε και εδώ. Η διαφορά τους από τα απλά νευρωνικά δίκτυα είναι ότι οι νευρώνες τους εκτελούν φίλτρα συνέλιξης με το διάνυσμα, στην περίπτωση μας εικόνα, που δέχονται ως είσοδο. Επίσης ανήκουν στην κατηγορία των unsupervised δικτύων, δηλαδή δεν είναι απαραίτητη η παρέμβαση του ανθρώπου-προγραμμαστή.

2.3.1 Συνέλιξη

Η συνέλιξη πραγματοποιείται στο συνελικτικό επίπεδο, το οποίο ακολουθεί τις περισσότερες φορές το επίπεδο εισόδου. Δοσμένης μιας εικόνας ως είσοδο, εφαρμόζεται σε αυτήν ένα φίλτρο (ή πυρήνας) μεγέθους $N \times N$, το οποίο μετατοπίζεται σύμφωνα με το επιλεγμένο μήκος διασκελισμού, καλύπτοντας κάθε φορά $N \times N$ pixel (stride) της εικόνας μέχρι την πλήρη κάλυψη της και πολλαπλασιάζοντας (matrix multiplication) τον πυρήνα με το επιλεγμένο τμήμα της εικόνας.

Κάθε νευρώνας λοιπόν, βάση αυτού του φίλτρου υπολογίζει το άθροισμα των βαρών προστιθέμενο με μια σταθερά, την κλίση (bias), άθροισμα το οποίο οδηγείται στην συνάρτηση ενεργοποίησης του νευρώνα δίνοντας του την έξοδο του. Στα συνελικτικά νευρωνικά δίκτυα η πιο συχνά χρησιμοποιούμενη συνάρτηση ενεργοποίησης είναι η ανορθωμένη γραμμική συνάρτηση. Μέσα λοιπόν από αυτήν την διαδικασία της συνέλιξης δημιουργείται το πρώτο κρυφό επίπεδο του ΣΝΔ, το οποίο αποτελείται από $N \times N$ νευρώνες. Η αντιστοίχιση τώρα της εισόδου και του φίλτρου προς το κρυφό επίπεδο ονομάζεται feature-map. Δεδομένου ότι μπορούμε να έχουμε πολλά φίλτρα, μας δίνεται η δυνατότητα πολλών feature-map, τα οποία μας προσφέρουν ένα κρυφό επίπεδο με πολλά κανάλια, όπου το καθένα αντιστοιχεί και σε ένα feature-map.



Εικόνα 2.5 : Λειτουργία της συνέλιξης

2.3.2 Συγκεντρωτικό επίπεδο (pooling)

Τα επόμενα επίπεδα του συνελκτικού δικτύου, είναι τα συγκεντρωτικά επίπεδα, ρόλος των οποίων είναι η μείωση της πληροφορίας του συνελκτικού επιπέδου για λόγους μείωσης της υπολογιστικής ισχύς, λειτουργώντας σαν μια συνάρτηση υποδειγματοληψίας.

Υπάρχουν 2 είδη συγκέντρωσης, η μέγιστη συγκέντρωση(max pooling) και η μέση συγκέντρωση (average pooling). Κατά την μέγιστη συγκέντρωση επιστρέφεται η μέγιστη τιμή από το τμήμα της εικόνας που σαρώνει σταδιακά το φίλτρο-πυρήνας, ενώ κατά την μέση συγκέντρωση μας δίνεται ο μέσος όρος των τιμών της εικόνας που σαρώνει το φίλτρο.

2.3.3 Πλήρως συνδεδεμένο επίπεδο(fully connected)

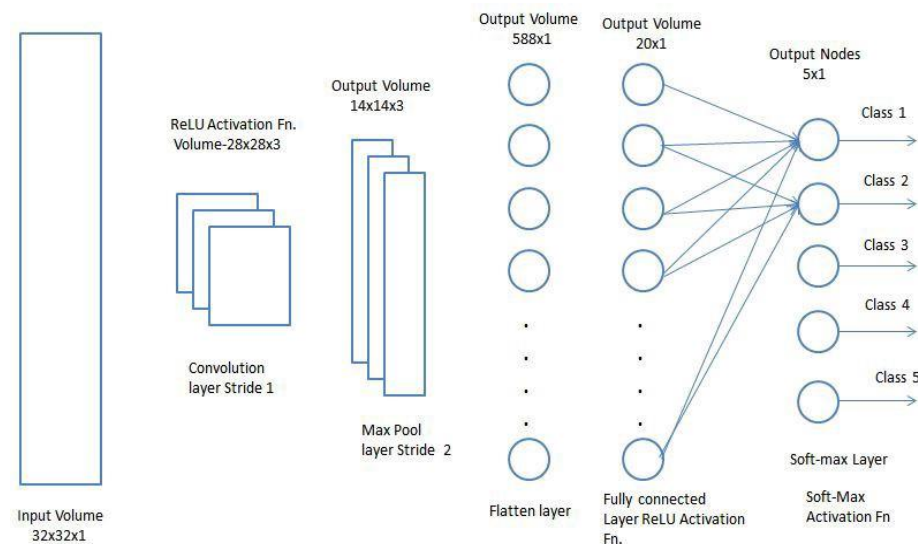
Πριν το επίπεδο εξόδου, υπάρχει ένα τελευταίο κρυφό επίπεδο, ένα πλήρως συνδεδεμένο επίπεδο. Σε αυτό το επίπεδο όλοι οι νευρώνες του είναι συνδεδεμένοι με όλους τους νευρώνες του αμέσως επόμενου επιπέδου. Η συνηθέστερη συνάρτηση ενεργοποίησης στα fully connected είναι η ανορθωμένη γραμμική συνάρτηση(ReLU). Σκοπός αυτού του επιπέδου είναι να παίρνει την πληροφορία του συνελκτικού επιπέδου και να την χρησιμοποιεί για την καλύτερη κατηγορία ταξινόμησης.

2.3.4 Επίπεδο Εξόδου

Στο επίπεδο εξόδου τώρα, γίνεται μετατροπή των τιμών που δέχεται σε πιθανότητες, ώστε να χρησιμοποιηθούν για την αξιολόγηση του δικτύου, ενώ η συνάρτηση ενεργοποίησης που χρησιμοποιείται σε αυτό το επίπεδο ονομάζεται softmax με τύπο :

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (2.9)$$

Όπου το x_i , αντιστοιχεί στην τιμή εξόδου του i -οστού νευρώνα.



Εικόνα 2.6: Συνελικτικό νευρωνικό επίπεδο

2.4 Generative Adversarial networks (GANs)

Τα generative adversarial networks αποτελούν μια προσέγγιση για το generative modeling χρησιμοποιώντας μεθόδους της βαθιάς μάθησης, όπως είναι τα συνελικτικά νευρωνικά δίκτυα και είναι αποτελεσματικά στην δημιουργία εικόνων υψηλής ανάλυσης. Το generative modeling είναι μια μέθοδος μάθησης χωρίς επίβλεψη (unsupervised), η οποία σου δίνει την δυνατότητα αυτόματης ανακάλυψης και εκμάθησης των μοτίβων των δεδομένων εισόδου, με τρόπο τέτοιο ώστε το μοντέλο στην συνέχεια να είναι σε θέση να παράγει εξόδους που θα μοιάζουν να προέρχονται από αυτό το σύνολο δεδομένων. Τα GANs λοιπόν είναι ένας έξυπνος τρόπος εκπαίδευσης ενός παραγωγικού μοντέλου και η χρησιμότητα

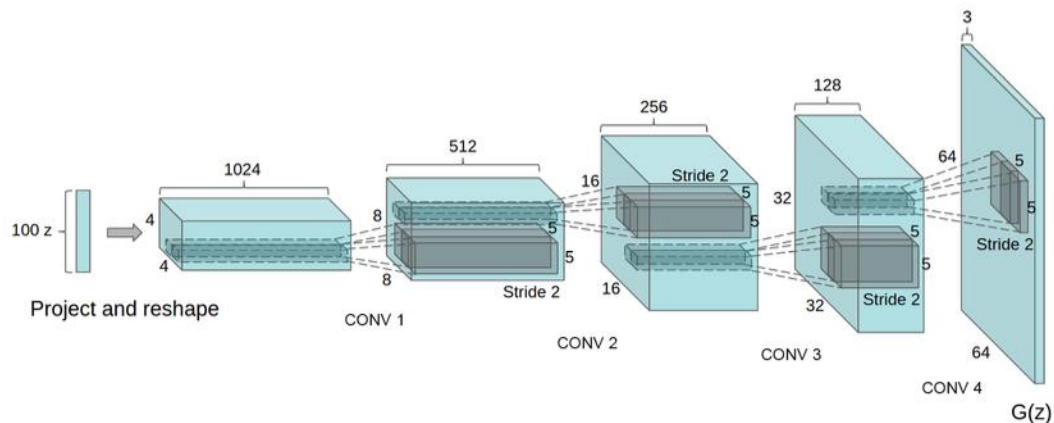
τους είναι να παράγουν-δημιουργούν δεδομένα από την αρχή, δηλαδή ουσιαστικά παράγουν τα δικά τους δεδομένα εκπαίδευσης, τα οποία είναι κυρίως εικόνες και οι οποίες δεν θα διαφέρουν από αυθεντικές-πραγματικές φωτογραφίες.

2.4.1 Μοντέλο γεννήτριας και διαχωρισμού(Generator and Discriminator)

Τα δύο κύρια δομικά συστατικά ενός GAN, είναι η γεννήτρια και ο διαχωριστής του, τα οποία αποτελούν δύο βαθιά συνελκτικά νευρωνικά δίκτυα.

Ας ξεκινήσουμε εξηγώντας τον ρόλο της γεννήτριας στο μοντέλο μας. Η αρχική είσοδος του μοντέλου μας είναι ουσιαστικά ένας θόρυβος z ο οποίος προέρχεται από δειγματοληψία χρησιμοποιώντας μια Gaussian ή ομοιόμορφη κατανομή. Δοσμένης λοιπόν της εισόδου μας και δίνοντας την στην γεννήτρια μας θα δημιουργηθεί μια εικόνα $G(z)$. Το z σαν είσοδος αντιπροσωπεύει τα ενδιάμεσα χαρακτηριστικά των εικόνων που παράγει η γεννήτρια, όπως είναι το χρώμα των ματιών, αλλά και το σχήμα ενός ανθρώπινου προσώπου. Αυτά τα ενδιάμεσα χαρακτηριστικά αποτελούν τα κρυφά χαρακτηριστικά, τα οποία είναι χρήσιμα για το μοντέλο μας αλλά δεν είναι άμεσα παρατηρήσιμα.

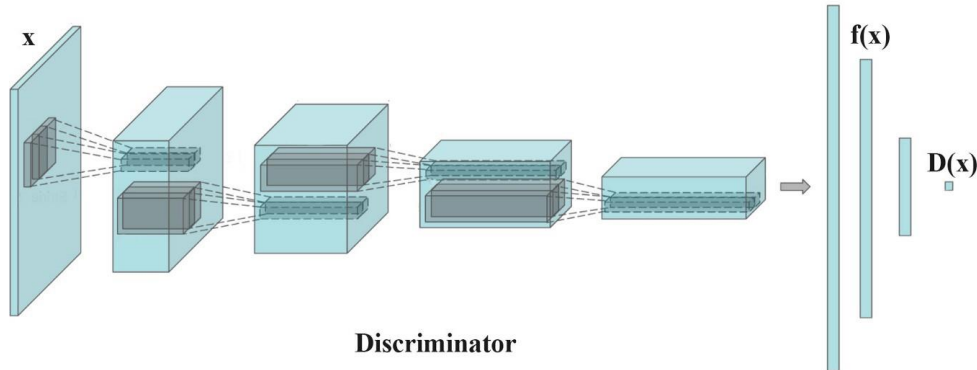
Όπως είπαμε και παραπάνω πρόκειται για ένα unsupervised μοντέλο, δηλαδή δεν ελέγχουμε τα χαρακτηριστικά που μαθαίνει το μοντέλο, όπως για παράδειγμα πιο byte του z καθορίζει το χρώμα ματιών ενός προσώπου.



Εικόνα 2.7 : Μοντέλο Γεννήτριας

Μέσα λοιπόν από μετατιθέμενα επίπεδα συνελίξεων η γεννήτρια μέσω του εισόδου z θα παράξει την εικόνα x . Στην συνέχεια η εικόνα περνιέται σαν είσοδος

στον διαχωριστή (discriminator), ο οποίος ακολουθώντας την αντίστροφη διαδικασία από αυτή της γεννήτριας, εξ



Εικόνα 2.8: Μοντέλο Διαχωριστή

-ετάζει τις πραγματικές εικόνες και τις παραγόμενες ξεχωριστά, θέλοντας να διακρίνει ποιες από αυτές είναι η πραγματική και ποια η παραγόμενη. Η έξοδος $D(x)$ του διαχωριστή, η οποία αποτελεί την πιθανότητα η είσοδος x να είναι πραγματική, θέλουμε να μας δίνει την τιμή 1 για την πραγματική εικόνα και 0 για παραγόμενη, εκπαιδεύοντας έτσι τον διαχωριστή να αναγνωρίζει τα χαρακτηριστικά που πλαισιώνουν μια πραγματική εικόνα. Άρα θέλουμε η γεννήτρια μας να δίνει εικόνες που θα 'ξεγελούν' τον διαχωριστή και θα δίνουν έξοδο $D(x) = 1$. Μέσω λοιπόν του αλγόριθμου backpropagation, διαδίδοντας την έξοδο του διαχωριστή πίσω προς την γεννήτρια, την εκπαιδεύουμε να δημιουργεί εικόνες που τον θα πείθουν ότι είναι πραγματικές. Μέσα από αυτό το παιχνίδι του ανταγωνισμού εκπαιδεύονται και τα δύο μέρη του μοντέλου μας έτσι ώστε να μας δίνουν ρεαλιστικές και αληθοφανείς εικόνες.

2.4.2 Feedback

Ας πάμε τώρα να δούμε τις εξισώσεις με τις οποίες γίνεται αυτό το feedback που βελτιώνει ένα GAN. Αρχικά δεδομένης της έξοδου $D(x)$ του διαχωριστή, θέλουμε να μετρήσουμε την απώλεια, χρησιμοποιώντας την διασταυρούμενη εντροπία, οπότε προκύπτει η αντικειμενική συνάρτηση του διαχωριστή:

$$\max_D V(D) = \underbrace{\mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)]}_{\text{recognize real images better}} + \underbrace{\mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]}_{\text{recognize generated images better}} \quad (2.10)$$

που θέλει να μεγιστοποιήσει την δυνατότητα να αναγνωρίζει τις πραγματικές ως πραγματικές και τις παραγόμενες ως ψεύτικες. Ενώ η αντικειμενική συνάρτηση της

γεννήτριας που θέλει να παράγει εικόνες με τιμές $D(x)$ που θα προσεγγίζουν το 1 και θα περνούν από την εξέταση του διαχωριστή ως πραγματικές προκύπτει ως:

$$\min_G V(G) = \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Optimize G that can fool the discriminator the most.

(2.11)

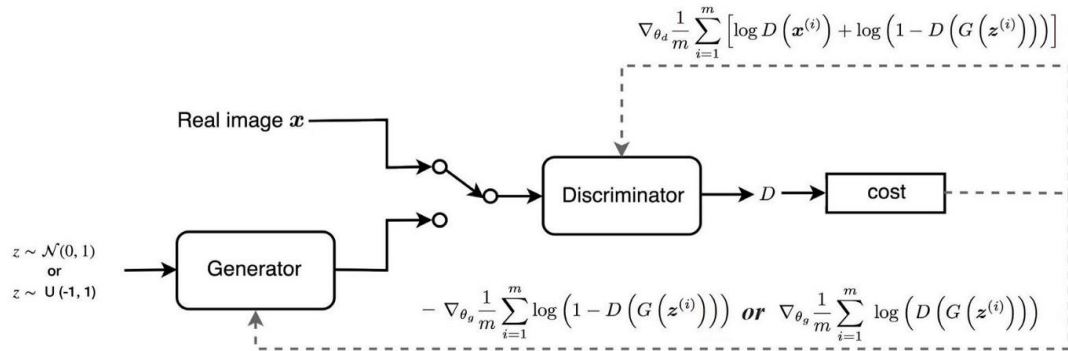
Πολλές φορές καθορίζουμε ένα GAN με μια minmax αντιμετώπιση όπου η γεννήτρια θέλει να ελαχιστοποιήσει το D και ο διαχωριστής να το μεγιστοποιήσει με την αντικειμενική συνάρτηση να γίνεται:

$$\max_D V(D) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

recognize real images better recognize generated images better

(2.12)

Στην συνέχεια χρησιμοποιώντας τον αλγόριθμο Gradient Descent διορθώνουμε τις παραμέτρους, όπως είναι τα βάρη, της γεννήτριας και επαναλαμβάνουμε όλη την παραπάνω διαδικασία μέχρι η γεννήτρια να παράγει εικόνες καλής ποιότητας.



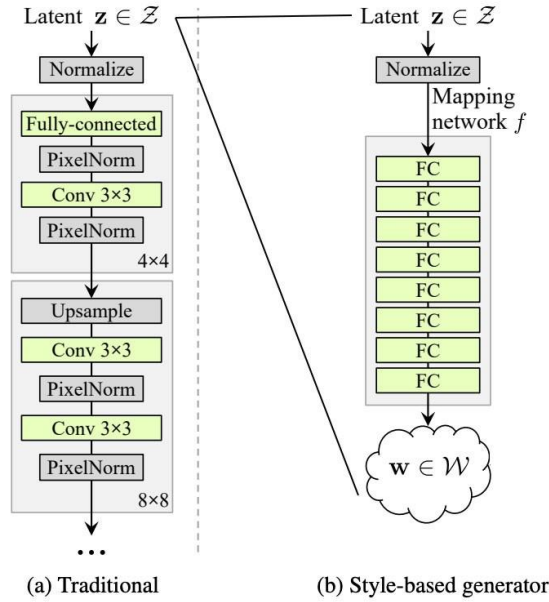
Εικόνα 2.9: Διαδικασία εκπαίδευσης ενός GAN

2.4.3 StyleGAN2

Το StyleGAN αποτελεί ένα από τα πιο δημοφιλή GANs μοντέλα, στον χώρο των inverse problems, το οποίο έχει αναπτυχθεί από την Nvidia. Είναι μια επέκταση της αρχιτεκτονικής των GANs που προτείνει μεγάλες αλλαγές στην γεννήτρια του μοντέλου και είχε σκοπό την παραγωγή ρεαλιστικών-πειστικών ψεύτικων ανθρώπινων προσώπων. Ας δούμε λοιπόν την αρχιτεκτονική του μοντέλου ξεκινώντας από την πρώτη έκδοση του (StyleGAN) και καταλήγοντας στο StyleGAN2.

Το StyleGAN εφαρμόζει ένα βαθύ δίκτυο, που ονομάζεται δίκτυο χαρτογράφησης το οποίο αποσκοπεί στην μετατροπή του ενδιάμεσου παράγοντα z σε έναν ενδιάμεσο χώρο w .

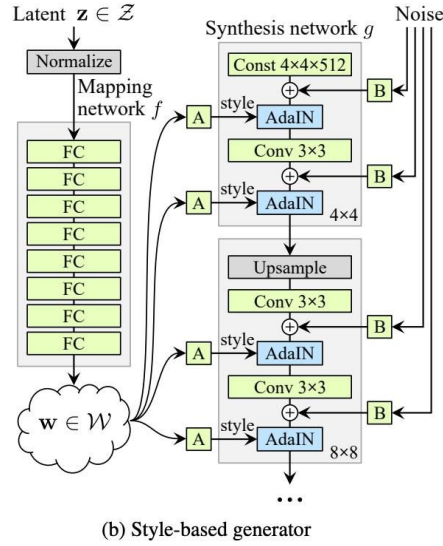
Στόχος αυτού του ενδιάμεσου χώρου είναι να προσφέρει στο μοντέλο ξεπερδεδεμένα χαρακτηριστικά τα οποία είναι εύκολο να παραχθούν από την γεννήτρια, έτσι ώστε να αποφύγουμε την παρουσίαση χαρακτηριστικών τα οποία δεν απαντώνται στο σύνολο εκπαίδευσης του δικτύου. Μέσα από την εισαγωγή του δικτύου αντιστοίχισης f θα μετατρέψουμε το z στον ενδιάμεσο χώρο w με την χρησιμοποίηση 8 πλήρους συνδεδεμένων επιπέδων, όπου το w τώρα θα θεωρείται ως το νέο z .



Εικόνα 2.10: Παραδοσιακή γεννήτρια GAN έναντι γεννήτρια στο StyleGAN

2.4.3.1 Γεννήτρια του StyleGAN

Σε αντίθεση με τα απλά GAN, όπου το z χρησιμοποιείται μόνο στο πρώτο στρώμα του δικτύου, εδώ θέλοντας να διατηρήσουμε τον ρόλο του προχωρώντας στο δίκτυο εφαρμόζουμε μια χωριστή αφινική(affine) A λειτουργία μάθησης ώστε να μετατρέψουμε το w σε κάθε επίπεδο.



Εικόνα 2.11 : Μοντέλο γεννήτριας στο StyleGAN

Ας δούμε τώρα συγκεντρωτικά τις προσθήκες και τις αλλαγές που διαφοροποιούν την παραδοσιακή γεννήτρια σε ένα GAN από αυτήν σε ένα StyleGAN:

- 1) Η πρώτη βελτίωση είναι η αντικατάσταση της μεθόδου παρεμβολής του πλησιέστερου γείτονα(nearest neighbor) τόσο στον διαχωριστή όσο και στην γεννήτρια με την μέθοδο της διγραμμικής παρεμβολής.
- 2) Η δεύτερη αλλαγή είναι η προσθήκη του δικτύου χαρτογράφησης και του στυλ, όπου το στυλ αναφέρεται στο χαρακτηριστικά των δεδομενων εκπαίδευσης, όπως είναι το χρώμα των μαλλιών ενός προσώπου ή οι φακίδες, ενώ το `pixelnorm` θα αντικατασταθεί από το AdaIN(προσαρμοστική κανονικοποίηση), η οποία επιτελεί το έργο της μεταφοράς του στυλ μια εικόνας στο περιεχόμενο μια άλλης και ο τύπος της ορίζεται ως:

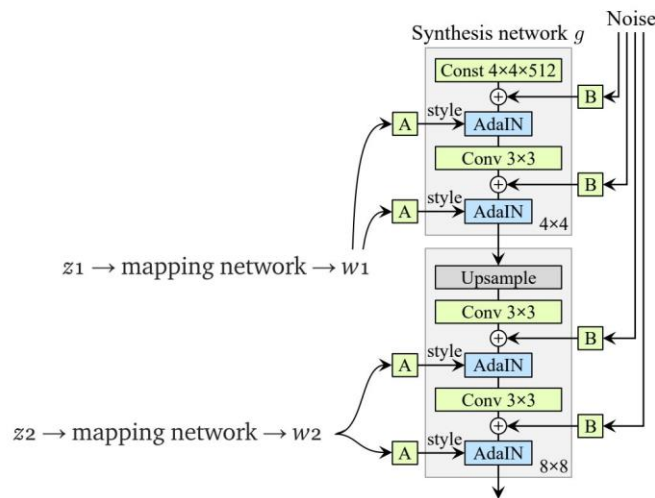
$$\text{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}$$

feature map i the mean & standard deviation of the feature map i
 normalize the feature map value
 (instance normalization)

(2.13)

- 3) Επειδή τα εμπειρικά αποτελέσματα έδειξαν ότι η είσοδος μιας μεταβλητής σαν είσοδο στο πρώτο επίπεδο του StyleGAN δεν προσφέρει κάτι ουσιαστικό, ο παράγοντας z σαν πρώτη είσοδος στο πρώτο επίπεδο αντικαταστάθηκε από μια σταθερά με διαστάσεις $4 \times 4 \times 512$.

- 4) Άλλη μια βελτίωση είναι αυτή της προσθήκης θορύβου στα χωρικά δεδομένα με σκοπό την δημιουργία στοχαστικής παραλλαγής(stochastic variation), η οποία μετριάζει επαναλαμβανόμενα μοτίβα αλλά και δημιουργεί παραλλαγές χαρακτηριστικών, όπως η τοποθέτηση φακίδων σε ένα ανθρώπινο πρόσωπο.
- 5) Η τελευταία προσθήκη αφορά την ανάμειξη τακτοποίησης(mixing regularization) κατά την οποία μεταβαίνουμε σε ένα διαφορετικό παράγοντα, έστω z_2 με σκοπό να εξαχθεί το στυλ αφού φτάσει σε μια δεδομένη χωρική ανάλυση.



Εικόνα 2.12: Τακτοποίηση ανάμειξης

Ωστόσο παρά την μεγάλη βελτίωση στην παραγωγή εικόνων με το StyleGAN, παρατηρήθηκαν κάποιες ανωμαλίες(artifacts) τις οποίες ήρθε να διορθώσει η δεύτερη έκδοση, το StyleGAN2 στο οποίο εντοπίστηκε το πρόβλημα, που οφειλόταν στην λειτουργία AdaIN, που κανονικοποιεί τον μέσο όρο και την διακύμανση στο κάθε χάρτη χαρακτηριστικών(feature map), καταστρέφοντας πιθανόν έτσι κάποιες πληροφορίες στα μεγέθη των χαρακτηριστικών που είναι συσχετιζόμενα μεταξύ τους.

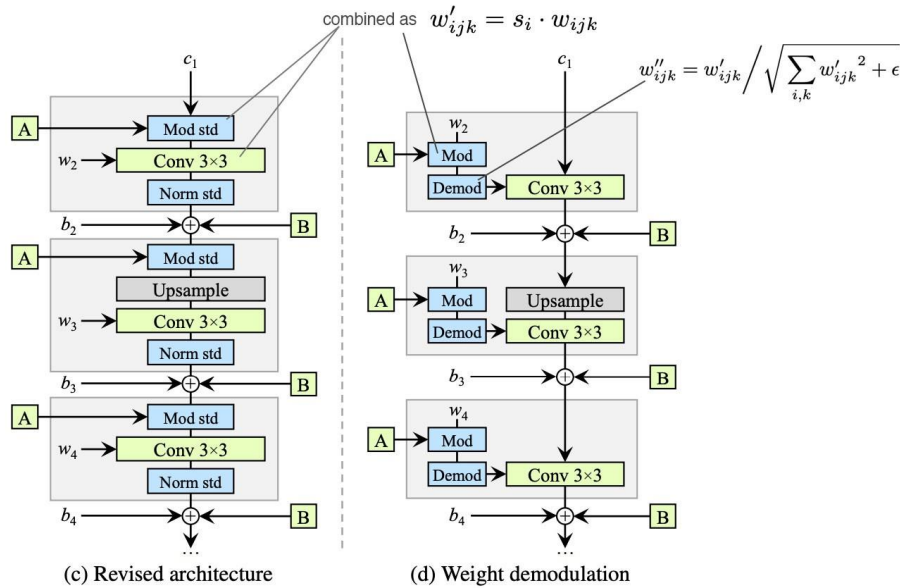
2.4.3.2 Λύση του προβλήματος των artifacts

Μέσα από τα πειραματικά αποτελέσματα οι αλλαγές που έγιναν στο StyleGAN2 για την εξάλειψη του προβλήματος είναι: 1) η αφαίρεση του τρόπου επεξεργασίας της σταθερά στην αρχή του επιπέδου 2) ο μέσος όρος δεν χρειάζεται

πλέον για την κανονικοποίηση των χαρακτηριστικών 3) η αφαίρεση της μονάδας θορύβου έξω από το στυλ 4) γίνεται απλοποίηση της σχεδίασης του μοντέλου της γεννήτριας μέσω της αποδιαμόρφωσης του βάρους κατά την οποία αντικαθιστούμε την μέθοδο της στιγμιαίας κανονικοποίησης με μια άλλη μέθοδο η οποία εξαλείφει τα artifacts.

Πιο αναλυτικά τώρα μέσα από την αποδιαμόρφωση του βάρους επέρχονται οι εξής αλλαγές:

- 1) Η διαμόρφωση (mod std) συγχωνεύεται με την συνέλιξη(3x3) και εφαρμόζεται σαν κλιμάκωση των βαρών συνέλιξης(mod)
- 2) Γίνεται ομαλοποίηση(demod) και κανονικοποίηση του βάρους



Εικόνα 2.13 : Αποδιαμόρφωση του βάρους

Αλλαγές που τα πειράματα έδειξαν ότι εξαφάνισαν τα artifacts.

2.4.3.3 Προοδευτική Ανάπτυξη(Progressive growth)

Το StyleGAN χρησιμοποιεί την ιδέα της προοδευτικής ανάπτυξης με σκοπό να σταθεροποιηθεί η εκπαίδευση για εικόνες υψηλής ανάλυσης. Με την προοδευτική ανάπτυξη το StyleGAN2 ψάχνει για εναλλακτικά σχέδια που επιτρέπουν στο δίκτυο να αποκτήσει πιο βαθιά μορφή και καλύτερη εκπαιδευτική σταθερότητα. Για αυτά τα σχέδια χρησιμοποιούμε το διγραμμικό φιλτράρισμα για

να πραγματοποιήσουμε υπο/αναδειγματοληψία στα προηγούμενα στρώματα και να προσπαθήσουμε να μάθουμε τις υπολλειματικές τιμές των επόμενων στρωμάτων.

2.5 Intermediate Layer Optimization

Ο αλγόριθμος Intermediate Layer Optimization αποτελεί ένα αλγόριθμο βελτιστοποίησης για την αντιμετώπιση των inverse problems. Αντίθετα με την συνήθη τακτική, κατά την οποία γίνεται βελτιστοποίηση μόνο πάνω στον κύριο ενδιάμεσο κώδικα, με αυτήν την προσέγγιση αλλάζουμε σταδιακά τα στρώματα εισόδου πετυχαίνοντας έτσι πιο εκφραστικές γεννήτριες. Για την εύρεση υψηλών διαστάσεων χωρικών δεδομένων ο αλγόριθμος αυτός επικεντρώνεται στα όρια μιας σφαίρας ακτίνας l_1 γύρω από την πολλαπλή επαγωγή των προηγούμενων στρωμάτων. Η ακτίνα αυτής της σφαίρας πρέπει να κυμαίνεται σε μικρές τιμές, ώστε να μην ξεπεραστούν τα όρια της γεννήτριας με αποτέλεσμα να δίνει μη ρεαλιστικές εικόνες. Γενικά αυτή η μέθοδος επικεντρώνεται σε μη-εποπτευόμενες τεχνικές που βασίζονται πάνω σε προ εκπαιδευόμενες γεννήτριες.

Το κεντρικό πρόβλημα βελτιστοποίησης που εμφανίζεται στον μη επιβλεπόμενο ανασχηματισμό των εικόνων είναι η αντιστροφή ενός παραγωγικού μοντέλου. Με την έννοια της αντιστροφής ενός παραγωγικού μοντέλου περιγράφουμε την διαδικασία κατά την οποία, δεν τροφοδοτούμε πλέον το μοντέλο μας με ένα θόρυβο z που προέκυψε από μια κατανομή, αλλά τώρα δίνουμε στο μοντέλο μας μια εικόνα x και ψάχνουμε να βρούμε ένα ενδιάμεσο διάνυσμα z , για το οποίο το μοντέλο θα μας δώσει ένα αποτέλεσμα, μια παραγόμενη εικόνα $G(z)$, η οποία θα προσεγγίζει με μεγάλη ακρίβεια την x , δεδομένης μια συνάρτησης κόστους της μορφής $||G(z)-x||$.

Ο αλγόριθμος που προτείνουμε λοιπόν αλλάζει προσαρμοστικά ποιο στρώμα της γεννήτριας θα βελτιστοποιείται κάθε φορά, κινούμενοι πάντα από τον αρχικό ενδιάμεσο κώδικα προς τα ενδιάμεσα στρώματα που είναι πιο κοντά στα pixel της εικόνας. Με το να βελτιστοποιούμε λοιπόν τα ενδιάμεσα στρώματα του μοντέλου μας, προεκτείνουμε το εύρος της γεννήτριας μας, το οποίο όπως είπαμε και παραπάνω δεν πρέπει να ξεπερνάει τα όρια της σφαίρας ακτίνας l_1 ώστε η γεννήτρια μας να μην μας δίνει μη ρεαλιστικές εικόνες.

2.5.1 Σύνθεση του αλγόριθμου

Δεδομένου μια προ εκπαιδευμένης γεννήτριας έστω $G(z) \in R^n$, η οποία παράγει εικόνες, δοσμένου ενός ενδιάμεσου κώδικα $z \in R^k$, αναλύουμε το $G = G_1 \circ G_2$ όπου $G_1 : R^k \rightarrow R^p$ και $G_2 : R^p \rightarrow R^n$. Ο ενδιάμεσος παράγοντας z προέρχεται από μια

Gaussian κατανομή, ενώ οι παρατηρήσεις μας προέρχονται από τον πίνακα μέτρησης $y = Ax + \text{noise}$ όπου $A : R^{m \times n}$ και x είναι η πραγματική εικόνα.

Αρχικά γίνεται βελτιστοποίηση ενός πολυδιάστατου ενδιαμέσου κώδικα z^k το οποίο αποτελεί την είσοδο του πρώτου στρώματος της γεννήτριας. Στην συνέχεια βελτιστοποιούμε την συνάρτηση κόστους $\|AG(z^k) - Ax\|$ χρησιμοποιώντας τον gradient descent για την ανανέωση των βαρών του μοντέλου μας. Από την λύση αυτής της διαδικασίας προκύπτει το \hat{z}^k το οποίο αντιστοιχούμε σε έναν p -dimensional χώρο χρησιμοποιώντας το G_1 μέσα από το οποίο προκύπτει η ενδιαμέση αναπαράσταση $\hat{z}^p = G_1(\hat{z}^k)$.

Μετά από αυτήν την διαδικασία ο αλγόριθμος μας αρχίζει και εκτελεί επαναλήψεις, όπου στην αρχή κάθε επανάληψης γίνεται βελτιστοποίηση πάνω στην είσοδο του p -dimensional χώρου του G_2 . Όπως είπαμε και προηγουμένως επιτρέπονται μόνο οι λύσεις μέσα στα όρια της σφαίρας l_1 , ενώ επιτρέπονται οι αποκλίσεις μόνο μέσα εύρος του G_1 που έχει σκοπό την αύξηση της εκφραστικότητας του μοντέλου. Εν συνέχεια βρίσκοντας τον ενδιαμέσο κώδικα \hat{z}^p ψάχνουμε για τον ενδιαμέσο κώδικα z^k ο οποίος προσεγγίζει όσο το δυνατόν περισσότερο το \hat{z}^p . Αυτό το πετυχαίνουμε βελτιστοποιώντας το z^p σε \hat{z}^p και μετά ελαχιστοποιούμε το κόστος $\|G_1(z^k) - \hat{z}^p\|$ χρησιμοποιώντας ξανά τον αλγόριθμο gradient descent. Μέσα από αυτό προκύπτει ένα καινούριο διάνυσμα \hat{z}^k το οποίο προβάλλεται ξανά στον ενδιαμέσο κώδικα ($\hat{z}^p = G_1(\hat{z}^k)$). Με αυτόν τον αλγόριθμο λοιπόν προσπαθούμε να βρούμε το κατάλληλο z το οποίο θα μας δώσει την καλύτερη απομίμηση της εικόνας x που έχουμε δώσει ως είσοδο στο μοντέλο μας.

Algorithm 1 ILO for one layer of the generator

```

// CSGM solution
1  $\hat{z}^k \leftarrow \operatorname{argmin}_{z^k \in B_2^k(r_1)} \|AG(z^k) - Ax\|_2$ 
2  $\hat{z}^p \leftarrow G_1(\hat{z}^k)$ 
3 for  $t \leftarrow 0$  to  $r$  do
    // Best solution within an  $l_1$  ball
    // centered around the prev. solution
4  $\hat{z}^p \leftarrow \operatorname{argmin}_{z^p \in \hat{z}^p \oplus B_1^p(r_2)} \|AG_2(z^p) - Ax\|$ 
    // Projection back to the range
5  $\hat{z}^k \leftarrow \operatorname{argmin}_{z^k \in B_2^k(r_1)} \|G_1(z^k) - \hat{z}^p\|$ 
6  $\hat{z}^p \leftarrow G_1(\hat{z}^k)$ 
end
// Return the best solution within an  $l_1$ 
// ball of some point in the range
7 return  $G_2(\hat{z}^p)$ 

```

Εικόνα 2.14: Ο αλγόριθμος βελτιστοποίησης ILO +

2.5.2 Προσαρμογή του ILO με το StyleGAN2

Αρχικά χρειάζεται να χωριστούν τα ενδιάμεσα στρώματα της γεννήτριας του StyleGAN2, το οποίο έχει παρατηρηθεί ότι δίνει καλύτερα αποτελέσματα. Έτσι χωρίζουμε την γεννήτρια σε 18 μέρη $G_1 \circ G_2 \circ \dots \circ G_{18}$ και τρέχουμε τον αλγόριθμό μας σε γύρους, όπου σε κάθε γύρο το αρχικό στρώμα απορρίπτεται ώστε να μπει το επόμενο, ενώ για να διασφαλίσουμε ότι μένουμε μέσα στα όρια της σφαίρας ακτίνας l_1 χρησιμοποιούμε τον αλγόριθμο Projected Gradient Descent. Για να εφαρμόσουμε την προβολή πάνω σε μια σφαίρα ακτίνας l_1 γύρω από την δεδομένη καλύτερη λύση χρησιμοποιείται η μέθοδος του Duchi. Επιπλέον καθώς προχωράμε σε ενδιάμεσα χωρικά δεδομένα μεγαλύτερων διαστάσεων αυξάνουμε την απόκλιση στο μέγιστο επιτρεπτό όριο της.

ενδιάμεσο δηλαδή μπορούμε να τα βελτιστοποιούμε δοσμένης μιας ακολουθίας $\{w_i^k\}_{i=1}^{18}$. Οι αποκλίσεις αυτές σταθεροποιούνται με έναν επιπλέον όρο στην συνάρτηση κόστους, μέσα από την οποία αποτυπώνεται η απόσταση μεταξύ των διανυσμάτων. Η μέθοδος ξεκινάει λοιπόν βελτιστοποιώντας μόνο τα πέντε πρώτα διανύσματα θορύβου, αλλά σταδιακά επιτρέπεται η βελτιστοποίηση των υπολοίπων ενδιάμεσων διανυσμάτων όσο προχωράμε σε ενδιάμεσα χωρικά δεδομένα υψηλότερων διαστάσεων.

2.5.3 Συναρτήσεις κόστους

Στην υπερανάλυση εικόνας χρησιμοποιείται ένας σταθμισμένος μέσος όρος της μετρικής LPIPS και της συνάρτησης κόστους MSE. Η μετρική LPIPS έχει παρατηρηθεί ότι έχει πολύ καλά αποτελέσματα με εικόνες μεγέθους 256x256, οπότε για την σύγκριση των εικόνων υψηλής ανάλυσης και χαμηλής κάνουμε υποδειγματοληψία της εικόνας χρησιμοποιώντας την δικυβική παρεμβολή. Ενώ στο inpainting αν έχουμε αρκετά δεδομένα από την εικόνα χρησιμοποιούμε μόνο την συνάρτηση κόστους MSE, αλλιώς χρησιμοποιούμε και την μετρική LPIPS. Όσον αφορά το πρόβλημα της αποθορυβοποίησης, αυτό αντιμετωπίζεται κατά την διαδικασία της βελτιστοποίησης της γεννήτριας προσθέτοντας Gaussian θόρυβο στην παραγόμενη εικόνα πριν την εκτέλεση της συνάρτησης κόστους.

Κεφάλαιο 3

Υλοποίηση και Πειράματα

3.1 Υλοποίηση

Σε αυτήν την εργασία όπως είπαμε και παραπάνω δεν θα μπορούμε στην διαδικασία να εκπαιδεύσουμε ένα δίκτυο ούτε να δείξουμε πως γίνεται αυτό, αλλά θα χρησιμοποιήσουμε ένα προ-εκπαιδευμένο δίκτυο το έχει εκπαιδευτεί να παράγει ανθρώπινα πρόσωπα. Επίσης θα δοκιμάσουμε την απόκριση του αλγόριθμου σε διάφορες κατηγορίες εικόνων.

Ας δούμε πρώτα μέσα από τον κώδικα πως υλοποιείται ο αλγόριθμος ILO αλλά και το πως φορτώνουμε τα προ εκπαιδευμένα δίκτυα.

```
--
61
62 class LatentOptimizer(torch.nn.Module):
63     def __init__(self, config):
64         super().__init__()
65         self.config = config
66         if config['image_size'][0] != config['image_size'][1]:
67             raise Exception('Non-square images are not supported yet.')
68
69         device = config['device']
70         self.downsampler_1024_256 = BicubicDownSample(4)
71         self.downsampler_1024_image = BicubicDownSample(1024 // config['image_size'][0])
72         self.downsampler_image_256 = BicubicDownSample(config['image_size'][0] // 256)
73
74         # Load models and pre-trained weights
75         gen = Generator(1024, 512, 8)
76         gen.load_state_dict(torch.load(config["ckpt"])[ "g_ema"], strict=False)
77         gen.eval()
78         self.gen = gen.to(device)
79         self.gen.start_layer = config['start_layer']
80         self.gen.end_layer = config['end_layer']
81         self.mpl = MappingProxy(torch.load('gaussian_fit.pt'))
82         self.percept = lpips.PerceptualLoss(model="net-lin", net="vgg",
83                                             use_gpu=device.startswith("cuda"))
84         self.init_state()
85
```

Εικόνα 3.1 : Φόρτωση προ εκπαιδευμένου δικτύου

Στην παραπάνω εικόνα βλέπουμε πως φορτώνουμε τα προ εκπαιδευμένα βάρη αλλά και το προ εκπαιδευμένο μοντέλο(ckpt) από τις γραμμές 75-82, αλλά ετοιμάζουμε και τις αντίστοιχες δικυβικές παρεμβολές ώστε να κάνουμε downsample στις παραγόμενες εικόνες έτσι ώστε να υπολογίσουμε την μετρική LPIPS.

```

210         loss += mse_loss
211     else:
212         # downsample generated images
213         downsampled = self.downsampler_1024_image(img_gen)
214         # mask
215         masked = downsampled * mask
216         # compute loss
217         diff = torch.abs(masked - self.original_imgs) - self.config['dead_zone_linear_alpha']
218         loss += self.config['dead_zone_linear'][index] * torch.max(torch.zeros(diff.shape, device=diff.device), diff).mean()
219
220     mse_loss = F.mse_loss(masked, self.original_imgs)
221     loss += self.config['mse'][index] * mse_loss
222     if self.config['pe'][index] != 0:
223         if self.config['lpips_method'] == 'mask':
224             p_loss = self.percept(self.downsampler_image_256(masked),
225                                   self.downsampler_image_256(self.original_imgs)).mean()
226         elif self.config['lpips_method'] == 'fill':
227             filled = mask * self.original_imgs + (1 - mask) * downsampled
228             p_loss = self.percept(self.downsampler_image_256(img_gen), self.downsampler_image_256(filled)).mean()
229         else:
230             raise NotImplementedError('LPIPS policy not implemented')
231
232     loss += self.config['pe'][index] * p_loss
233
234     loss += self.config['geocross'] * loss_geocross(self.latent_z[2 * start_layer:])
235

```

Εικόνα 3.2: Υπολογισμός μετρικής LPIPS

3.2 Πειράματα και Αποτελέσματα

Ας δούμε τώρα τα πειράματα που εκτελέσαμε δοκιμάζοντας τον αλγόριθμο τόσο με τις default τιμές, αλλά και κάτω από διάφορες αλλαγές που κάναμε στις παραμέτρους του αλγορίθμου.



Εικόνα 3.3: Obama (s.r 64x64 to 1024x1024)

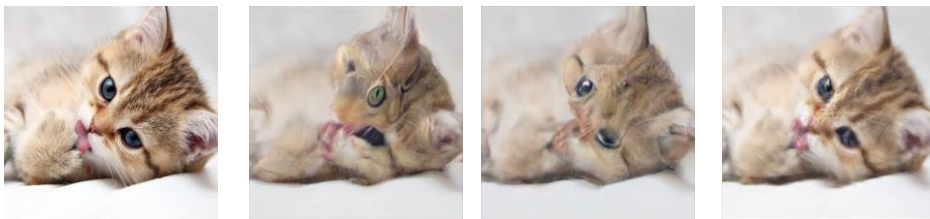
Πρώτα θα τεστάρουμε το μοντέλο στην παραγωγή προσώπων. Στην εικόνα 17 βλέπουμε το αποτέλεσμα που μας δίνει το μοντέλο μας, δοσμένης μιας εικόνας 64 pixel κατά την οποία έχουμε εκτελέσει 250 steps per layer, παράμετρος που μέσω

των πειραμάτων φάνηκε ότι ουσιαστικά αποτελεί την πιο σημαντική παράμετρο για την μεταβολή των αποτελεσμάτων, σε καθένα από τα στρώματα του αλγόριθμου. Όπως φαίνεται η παραγόμενη εικόνα έχει σχετικά μεγάλη ακρίβεια και προσεγγίζει την πραγματική, αν και παρατηρούμε ορισμένα artifacts, δεδομένου ότι η εικόνα που δόθηκε σαν είσοδος στο μοντέλο μας είχε αρκετά μικρή ανάλυση, ενώ στην εικόνα 18 φαίνεται η εικόνα που μας δίνει το μοντέλο αν για κάθε στρώμα του τώρα εκτελέσουμε 25 steps per layer. Όπως είναι εμφανές ότι η αύξηση της συγκεκριμένης παραμέτρου οδήγησε σε αύξηση της ακρίβειας του αποτελέσματος.



Εικόνα 3.4: : Obama (s.r 64x64 to 1024x1024)

Τώρα ας δοκιμάσουμε τον αλγόριθμο αλλάζοντας τα προ εκπαιδευμένα δίκτυα και δοκιμάζοντας άλλες κατηγορίες εικόνων.



Εικόνα 3.5 : Cat (s.r 64x64 to 1024x1024)

Ξεκινώντας με την πραγματική εικόνα στην εικόνα 19 βλέπουμε ότι χρησιμοποιώντας τώρα ένα δίκτυο το οποίο έχει εκπαιδευτεί να παράγει γάτες το η γεννητριά που έχει εκπαιδευτεί πάνω στα πρόσωπα δίνει σχετικά καλά αποτελέσματα, αλλά όχι παρόμοια με αυτά των προσώπων, ενώ βλέπουμε ότι η εμφάνιση artifacts εδώ είναι πιο έντονη. Η δεύτερη εικόνα προκύπτει με την χρήση

της παραμέτρου `steps per layer` στην τιμή 25 και για τα 4 στρώματα , η τρίτη με τις τιμές 50 για τα 2 δύο πρώτα στρώματα και 25 για τα 2 τελευταία, ενώ η τέταρτη με την τιμή 250 για όλα τα στρώματα.



Εικόνα 3.6

Προσπαθώντας να βελτιώσουμε την παραπάνω απόδοση, αποθηκεύσαμε και χρησιμοποιήσαμε κάποιες μεταβλητές βελτιστοποίησης μέσα στα αρχεία `noises.pt`, `latent.pt` και `gen_out.pt` που όπως φαίνεται στην εικόνα 20 βελτίωσε αρκετά την απόδοση του μοντέλου μας, καθώς βλέπουμε ότι έχουν εξαφανιστεί αρκετά τα artifacts. Παρακάτω θα δούμε όλες τις κατηγορίες που αναπαρήγαγε τον μοντέλο μας χρησιμοποιώντας και κάποιες πιο περίεργες κατηγορίες εικόνων, ξεκινώντας δίνοντας μικρές τιμές στην παράμετρο `steps per layer` και αυξάνοντας την παράμετρο μέχρι και την τιμή 250, καθώς διαπιστώθηκε πειραματικά ότι τιμές μεγαλύτερες από αυτήν δεν παρουσιάζουν κάποια σημαντική βελτίωση, ενώ χρειάζεται αρκετή ώρα για την εκτέλεση του αλγορίθμου.



Εικόνα 3.7 : Car (s.r 64x64 to 1024x1024)



Εικόνα 3.8: Car (s.r 64x64 to 1024x1024) used optimization variables



Εικόνα 3.9: Church (s.r 64x64 to 1024x1024)



Εικόνα 3.10: Church (s.r 64x64 to 1024x1024) used optimization variables



Εικόνα 3.11: Horse (s.r 64x64 to 1024x1024)



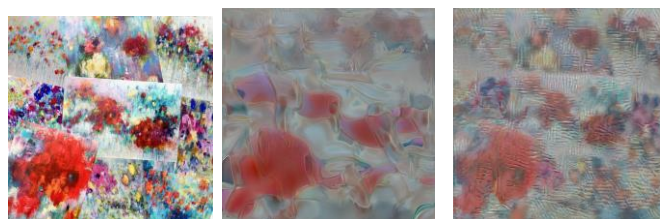
Εικόνα 3.12: Horse (s.r 64x64 to 1024x1024) used optimization variables



Εικόνα 3.13: Figure (s.r 64x64 to 1024x1024)



Εικόνα 3.14: Figure (s.r 64x64 to 1024x1024) used optimization variables



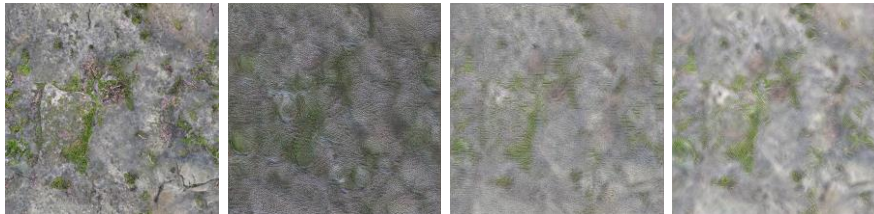
Εικόνα 3.15: Abstract art (s.r 64x64 to 1024x1024)



Εικόνα 3.16: Abstract art (s.r 64x64 to 1024x1024) used optimization variables



Εικόνα 3.17: Tex (s.r 64x64 to 1024x1024)



Εικόνα 3.18: Tex (s.r 64x64 to 1024x1024) used optimization variables

3.2.1 Επιπρόσθετα πειράματα

Ας δούμε τώρα πως ανταποκρίνεται ο αλγόριθμος στο inverse problem του inpainting, τόσο στην κατηγορία των προσώπων όσο και στις υπόλοιπες κατηγορίες που είδαμε και παραπάνω. Αρχικά όπως και προηγουμένως θα ξεκινάμε με την τιμή της steps per layer στα 25 και φτάνουμε μέχρι την τιμή 250 για κάθε ομάδα εικόνων. Επίσης θα χρησιμοποιήσουμε ξανά τις optimization variables για να δούμε να προκύψει κάποια βελτίωση, ενώ θα εφαρμόσουμε και την μετρική κόστους LPIPS.



Εικόνα 3.19 : Obama inpainting



Εικόνα 3.20 : Obama inpainting used optimization variables and LPIPS



Εικόνα 3.21 : Luci Lu inpainting



Εικόνα 3.22 : Cat inpainting



Εικόνα 3.23 : Cat inpainting used optimization variables and LPIPS



Εικόνα 3.24 : Church inpainting



Εικόνα 3.25 : Church inpainting used optimization variables and LPIPS



Εικόνα 3.26 : Car inpainting



Εικόνα 3.27 : Car inpainting used optimization variables and LPIPS



Εικόνα 3.28 : Figure inpainting



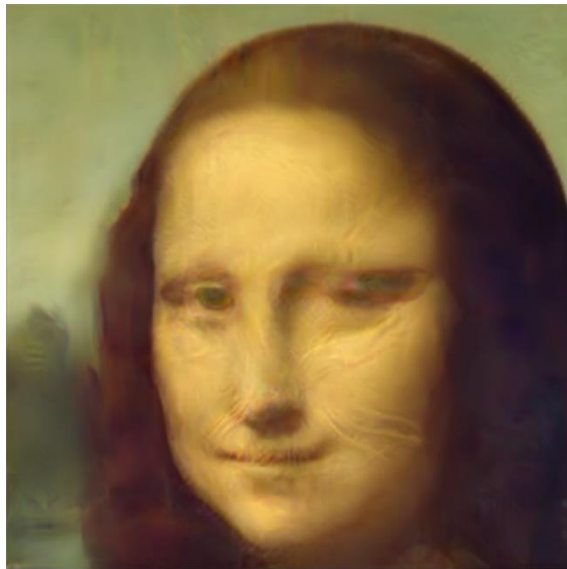
Εικόνα 3.29 : Figure inpainting used optimization variables and LPIPS



Εικόνα 3.30 : Abstract art inpainting



Εικόνα 3.31 : Abstract art inpainting used optimization variables and LPIPS



Εικόνα 3.32 : Mona Lisa (s.r from 64x64 to 1024x1204) from pretrained model for cats

Όπως φαίνεται από τα παραπάνω παραδείγματα, το μοντέλο ανταποκρίνεται με αρκετά καλή ακρίβεια στην κατηγορία των προσώπων, ακόμα και αν η εικόνα εισόδου έχει πολύ μικρή ανάλυση, αλλά όσο προχωράμε σε κατηγορίες εικόνων πολύ διαφορετικές από το dataset με το οποίο έχει εκπαιδευτεί η γεννήτρια που χρησιμοποιούμε εδώ, εκεί βλέπουμε ότι τα αποτελέσματα είναι αρχίζουν σιγά σιγά να απομακρύνονται αρκετά από αυτά των πραγματικών εικόνων, χωρίς όμως τα παραγόμενα αποτελέσματα να είναι απογοητευτικά.

Κεφάλαιο 4

Συμπεράσματα και προεκτάσεις

4.1 Συμπεράσματα και Προεκτάσεις

Ο αλγόριθμος Intermediate Layer Optimization σε συνδυασμό με το StyleGAN2 αποδείχτηκε ότι αποτελεί μια πολύ καλή προσθήκη στον τομέα των inverse problems καθώς ανταποκρίνεται με μεγάλη ακρίβεια στην κατηγορία των εικόνων που απεικονίζουν πρόσωπα, τόσο στο super resolution αλλά και στο inpainting. Ικανοποιητικά είναι και τα αποτελέσματα στην κατηγορίες εικόνων που περιέχουν ζώα ή αντικείμενα, ενώ δεδομένου της φύσης της γεννήτριας που χρησιμοποιούμε, αξιοπρεπή είναι και τα αποτελέσματα στις πιο περίεργες κατηγορίες εικόνων.

Μελλοντικές προεκτάσεις πάνω στην παρούσα εργασία θα ήταν σίγουρα βελτίωση στις υπόλοιπες κατηγορίες εικόνων που παρουσιάστηκαν στα επιπρόσθετα πειράματα, αλλά και η χρησιμοποίηση του αλγορίθμου στον κλάδο της ιατρικής, μέσα από την υπαράνάλυση εικόνα πάνω, παραδείγματος χάρη σε τομείς την ιατρικής όπως είναι η μαγνητικές τομογραφίες. Επίσης ενδιαφέρον θα είχε και ο συνδυασμός του αλγορίθμου ILO με άλλες δίκτυα GAN όπως είναι το BigGAN.

Βιβλιογραφία

Giannis Daras , Joseph Dean , Ajil Jalal, Alexandros G. Dimakis. Intermediate Layer

Optimization for Inverse Problems using Deep Generative Model. URL
<https://arxiv.org/pdf/2102.07364.pdf> (2021)

Giannis Daras, ILO. URL <https://github.com/giannisdaras/ilo> (2021)

Chao Dong, Chen Change Loy, Member, IEEE, Kaiming He, Member, IEEE, and

Xiaoou Tang, Fellow, IEEE. Image Super-Resolution Using Deep
Convolutional Networks URL <https://arxiv.org/pdf/1501.00092.pdf> (2014)

Ian J. Goodfellow, Jean Poulet-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley,

Sherjil Ozair† , Aaron Courville, Yoshua Bengio. Generative Adversarial
Networks URL <https://arxiv.org/pdf/1406.2661.pdf> (2014)

Justin Pinkey . Awesome Pretrained StyleGAN2 URL

<https://github.com/justinpinkney/awesome-pretrained-stylegan2> (2020)

Kate Brush. Deep Learning URL

<https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network> (2021)

NVLABS. StyleGAN2 Official TensorFlow Implementation. Url

<https://github.com/NVlabs/stylegan2> (2020)

Jonathan Hui. StyleGAN- StyleGAN2 URL

<https://jonathan-hui.medium.com/gan-stylegan-stylegan2-479bdf256299> (2020)

