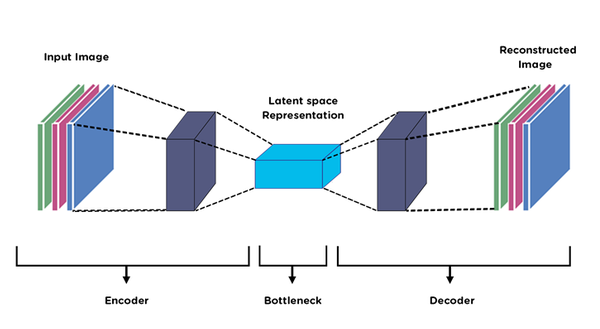
Autoencoders



Ένας autoencoder πρακτικά συμπιέζει την πληροφορία σε λιγότερες διαστάσεις (encoding part) και έχει ως στόχο την αναπαράσταση της πίσω στην αρχική μέσω αυτών των ελαχιστοποιημένων διαστάσεων(decoding part).

Μέσω των autoencoders δίνεται η ευκαιρία να μελετηθούν πολλά πράγματα όπως το αν η συμπιεσμένη αυτή πληροφορία είναι χρήσιμη και αν μπορεί να χρησιμοποιηθεί το συμπιεσμένο πλέον dataset ώστε να εκπαιδευτεί ένα δίκτυο λειτουργώντας ως feature extractor για παράδειγμα. Αυτή την μελέτη έχω κάνει έως ένα βαθμό στην πρώτη εργασία. Μελετώντας περεταίρω ανακάλυψα πως για να πετύχει κάτι τέτοιο πρέπει να μεταβούμε σε β-Variational Autoencoders ή αλλιώς disentangled autoencoders όπου πετυχαίνουν ακριβώς τη δυνατότητα του feature extraction σε ένα dataset. Άλλες δυνατότητες των autoencoders (με την κατάλληλη τροποποίηση) είναι το image denoising, δηλαδή να αφαιρούμε τον θόρυβο από εικόνες, και το image generation όπου χρησιμοποιούμε το decoding part του autoencoder για να παράξουμε νέες εικόνες.

Όπως γενικά ισχύει στα νευρωνικά δίκτυα έτσι και στους autoencoders, όντας νευρωνικά δίκτυα και αυτοί, υπάρχουν διάφορες αρχιτεκτονικές που μπορεί να χρησιμοποιήσει κανείς για την υλοποίηση τους, αναλόγως τους στόχους και τις ανάγκες του εκάστοτε project. Στην περίπτωση μας, όπου το dataset μας (cifar-10) αποτελείται από έγχρωμες εικόνες, λογικό είναι να απευθυνθούμε και σε convolutional(-deconvolutional) αρχιτεκτονικές για καλύτερη διαχείριση της πληροφορίας. Παρακάτω θα γίνει μελέτη επάνω σε **dense** και **convolutional** autoencoders για

* Reconstruction
* Image denoising
* Variational Autoencoder(VAE) for image generation

**Image Reconstructing AE**

Τα πειράματα ξεκίνησαν με την υλοποίηση **dense fully connected** δικτύων **AE** που είχαν ως στόχο το ίδιο το input που λάμβαναν ως train set. Σύντομα έγινε αντιληπτό πως **ρηχές αρχιτεκτονικές** ήταν πιο αποτελεσματικές από βαθύτερες αρχιτεκτονικές.(4-5% διαφορά και χειρότερο runtime) Επίσης η **relu** σαν activation function στο latent dimension είχε πολύ χειρότερα αποτελέσματα (και στην διάρκεια των 3εργασιών έχουμε διαπιστώσει πως δεν είναι σωστό να χρησιμοποιούμε την ReLU στα τελευταία layers διότι δυσκολεύει το train επειδή δεν αφήνει τις παραγώγους άρα και την πληροφορία να γυρίσει προς τα πίσω και να διορθωθεί). Επισημαίνουμε επίσης ότι ο συνδυασμός του **adam** σαν optimizer με το **mean squared error** ως loss function είναι ο βέλτιστος καθώς οποιαδήποτε άλλη αλλαγή είχε χειρότερα αποτελέσματα. Το **latent dim** παίζει σημαντικό στην αποδοτικότητα του autoencoder και βλέπουμε πως μείωση στις 512 διαστάσεις μας δίνει ένα έξτρα 5% (περίπου) στην απόδοση. Το καλύτερο μοντέλο που εκπαιδεύσαμε είναι με 1 μόνο layer ( αυτό του latent dimension) και πετυχαίνει accuracy 80%.

A picture containing calendar

Description automatically generated

Στην συνέχεια υλοποιήθηκαν convolutional δίκτυα στα οποία υπάρχουν 2 βασικές αρχιτεκτονικές. Η μία είναι fully convolutional- deconvolutional layers και η άλλη είναι convolutional-dense(latent dimension) – dense-deconvolutional.

Στη πρώτη περίπτωση παρατηρήθηκε πως όσο και να άλλαζαν οι παράμετροι(layers, filters, kernels, strides etc.) τα αποτελέσματα παρέμεναν πάνω κάτω σταθερά γύρω στο 73%. Μια σημαντική παρατήρηση όμως που αξίζει να σημειωθεί είναι πως τα reconstructed images με αυτή την αρχιτεκτονική φαίνονται πιο pixelated σε αντίθεση με τους dense AE όπου φαίνονται πιο blurred και ίσως όχι τόσο sharp. Αυτό μπορεί να οφείλεται στο πως λειτουργούν οι convolutional αρχιτεκτονικές καθώς χρησιμοποιούν τον convolutional kernel και πιθανότατα εκεί κρύβεται και η ύπαρξη αυτού του φαινομένου. Παρόλα αυτά το αποτέλεσμα παραμένει αναπάντεχο καθώς θα περιμέναμε μια convolutional αρχιτεκτονική να έχει καλύτερα αποτελέσματα σε εικόνες (όπου παίζουν ρόλο τα neighboring pixel) από μία dense αρχιτεκτονική αλλά τα αποτελέσματα δεν το υποστηρίζουν. Τέλος παρατηρήθηκε πως για κάποιο λόγο διαφορετικά batch sizes είχαν χειρότερη επίδραση στο δείγμα οπότε χρησιμοποιήθηκε σαν batch ολόκληρο το training set.

Όσο για τη δεύτερη περίπτωση έγιναν παρόμοια αποτελέσματα με την πρώτη και είδαμε ελάχιστα καλύτερες επιδόσεις της τάξης του 2-3%. Χρησιμοποιήθηκε convolutional αρχιτεκτονική στο μέχρι τώρα καλύτερο dense δίκτυο μας και το μοντέλο είχε χειρότερες επιδόσεις από το dense (accuracy 75%) αλλά καλύτερη από τις fully convolutional αρχιτεκτονικές. Άλλα πειράματα δεν έδειξαν κάποια άλλη ιδιαίτερη επίδραση στην απόδοση.

**ΣΥΜΠΕΡΑΣΜΑΤΑ**

Κάποια συμπεράσματα μελετών που επιβεβαίωσα και με τα δικά μου πειράματα είναι τα εξής :

* Οι ρηχές αρχιτεκτονικές φέρουν καλύτερα αποτελέσματα από βαθιές αρχιτεκτονικές όπου δεν καταφέρνουν να συμπιέσουν σωστά την πληροφορία και να την αναπαραστήσουν. Συγκεκριμένα το καλύτερο image reconstruction autoencoder δίκτυο που έχτισα ήταν με μόνο 1 layer.
* Ο dense autoencoder φαίνεται να τα πηγαίνει ελάχιστα καλύτερα από τους convolutional autoencoders(για image reconstruction).

Ένα άλλο εύρημα είναι πως όλα τα δίκτυα ξεκινούσαν με underfit, το οποίο δεν κατάφερα να φτιάξω.

Μια σημαντική παρατήρηση όμως που αξίζει να σημειωθεί είναι πως τα reconstructed images με αυτή την αρχιτεκτονική φαίνονται πιο pixelated σε αντίθεση με τους dense AE όπου φαίνονται πιο blurred και ίσως όχι τόσο sharp. Αυτό μπορεί να οφείλεται στο πως λειτουργούν οι convolutional αρχιτεκτονικές καθώς χρησιμοποιούν τον convolutional kernel και πιθανότατα εκεί κρύβεται και η ύπαρξη αυτού του φαινομένου.

**Convolutional AE:** (not from the same model)

PowerPoint

Description automatically generated with low confidence

A picture containing text

Description automatically generatedA picture containing background pattern

Description automatically generated

**Dense AE:**

A picture containing different, window, several

Description automatically generated

A picture containing text

Description automatically generatedA close up of a person's face

Description automatically generated with low confidence

**PCA Image Reconstruction**

A person wearing a hat

Description automatically generated with low confidence

**PCA:** Από τις πιο διαδομένες και επιτυχημένες τεχνικές για dimensionality reduction. Το Principal Component Analysis πετυχαίνει να μειώσει σημαντικά τις διαστάσεις, κρατώντας το μεγαλύτερο δυνατό ποσοστό της «σημαντικής» πληροφορίας. Έτσι το dataset γίνεται πιο εύκολα διαχειρίσιμο, ο χρόνος της εκπαίδευσης μειώνεται δραματικά και γίνεται πιο εύκολο visualization των δεδομένων. Η λειτουργία του μπορεί να περιγραφεί από μερικά απλά βήματα. Αρχικά πρέπει να κάνουμε standardization ώστε κάθε μεταβλητή να συνεισφέρει όμοια στην ανάλυση και να μην υπάρχει bias. Μετέπειτα πρέπει να βρούμε τον covariance matrix ώστε να δούμε τις συσχετίσεις μεταξύ τους και να δούμε που υπάρχει άχρηστη πληροφορία. Τέλος πρέπει να υπολογίσουμε τις ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα για να πάρουμε τα principal components οι οποίοι είναι γραμμικοί συνδυασμοί από τις αρχικές μεταβλητές.

Chart, diagram

Description automatically generated

Αντιστρέφοντας βέβαια την διαδικασία του PCA δημιουργούμε ένα νέο μοντέλο το οποίο κάνει ακριβώς ότι προσπαθούμε να πετύχουμε στα προηγούμενα παραδείγματα με τους Autoencoders, δηλαδή πετυχαίνει να κάνει reconstruct images που προέρχονται από ελαχιστοποιημένες διαστάσεις. Ο τύπος που μπορούμε να χρησιμοποιήσουμε για να πετύχουμε αυτή τη λειτουργία είναι πολύ απλά :

PCA reconstruction=PC scores⋅Eigenvectors(transpose)+Mean

Έγιναν 3 πειράματα για να μελετηθεί η χρησιμότητα και η επιτυχία τέτοιων μοντέλων στο ένα κρατήθηκε 90% του variance ενώ στο άλλο το 95% και στο τελευταίο χρησιμοποιήθηκε ο ίδιος αριθμός διαστάσεων που είχαν και οι AE δηλαδή 512 διαστάσεις.

Στο πρώτο πείραμα κρατήσαμε το 90% του variance και τα αποτελέσματα δεν ήταν πολύ καλά(blurry images). Μοιάζουν αρκετά με τα αποτελέσματα από τα κακά δίκτυα των autoencoders παρόλα αυτά καταφέρνουν να πετύχουν ένα ποσοστό deconstruct και χρησιμοποιώντας μόνο 99 διαστάσεις.

nComp = 99 (variance = 90%)

A picture containing different

Description automatically generated

Στο δεύτερο πείραμα κρατήσαμε το 95% του variance που είναι ένα ικανοποιητικό ποσοστό και θα έπρεπε να αντιπροσωπεύει το dataset μας σωστά. Αυτό βλέπουμε και από τα αποτελέσματα όπου φαίνεται πως το reconstruction έχει γίνει σε αρκετά καλό βαθμό και χρησιμοποιώντας μόνο 217 διαστάσεις.

nComp = 217 (variance=95%)

A picture containing different

Description automatically generated

Στο 3ο και τελευταίο πείραμα χρησιμοποιήσαμε 512 διαστάσεις που είναι και το νούμερο του latent space στους AE ώστε να υπάρξει και μια καλύτερη σύγκριση. Αυτός ο αριθμός principal components καταφέρνει να διατηρήσει το 98.3% του variance και δίνει αποτελέσματα που σχεδόν δεν ξεχωρίζουν από τις αρχικές εικόνες και φαίνεται να είναι η καλύτερη επιλογή σε σχέση με τους Auto Encoders.

nComp = 512 (variance=98.3%)

**A picture containing different

Description automatically generated**

**Image Denoising AE**

A picture containing old

Description automatically generated

Στα denoising AE οι αρχιτεκτονικές είναι ίδιες με αυτές των AE. Το δίκτυο δεν αλλάζει καθόλου, πάλι έχουμε ένα encoding και ένα decoding part και μπορούμε να απευθυνθούμε είτε σε dense είτε σε convolutional αρχιτεκτονικές. Αυτό που προσπαθούμε να πετύχουμε είναι να εισάγουμε μια εικόνα με θόρυβο και έχουμε ως στόχο να λάβουμε μια εικόνα reconstructed χωρίς θόρυβο. Το μόνο που έχουμε να κάνουμε λοιπόν είναι να πάρουμε το dataset και να εισάγουμε θόρυβο σε αυτό και με αυτό να κάνουμε την εκπαίδευση των μοντέλων μας. Τον θόρυβο τον κάνουμε sample από κανονική κατανομή με διασπορά 0.3. Έτσι παίρνουμε τις εικόνες που βλέπουμε στο επάνω sample.

Dense Denoising AE(best model)

A picture containing window, different

Description automatically generated

Στους Denoising Dense AE μετά από στοχευμένα test παρατηρήθηκε πως τα συμπεράσματα που απορρέουν είναι τα ίδια για τους απλούς Reconstructing AE. Για παράδειγμα τα ρηχά δίκτυα αποδίδουν καλύτερα από τα πιο βαθιά κλπ.

Dense Denoising AE

A picture containing window, different

Description automatically generated

Conv-Dense-DeConv Denoising AE

A picture containing window, different

Description automatically generated

Conv(U-net inspired) Denoising AE

A picture containing window, different

Description automatically generated

Παρατηρούμε πως αυτή η αρχιτεκτονική ανεβάζει αρκετά την επίδοση των convolutional autoencoders και κάνει αρκετά καλό reconstruction με θόρυβο(αν και αρκετές εικόνες είναι πολύ θολές). Βέβαια το κόστος αυτής της επίδοσης είναι ο χρόνος καθώς με gpu enabled στο google colab ήθελε τουλάχιστον 40 second για κάθε εποχή του train. Γενικά παρατηρείται πως βαθύτερες αρχιτεκτονικές σε convolutional δίκτυα και με σωστό normalization φέρουν καλύτερα αποτελέσματα.

**Variational Autoencoders**

Οι VAE μπορούν να χαρακτηριστούν ως απλοί AE των οποίων το training κανονικοποιείται για να αποφύγουν το overfitting και να εξασφαλίσουν ότι το latent space τους θα έχει καλές ιδιότητες που ενισχύουν το generative process. Έτσι έχουν και αυτά ένα encoding-decoding part αλλά ο encoder επιστρέφει μια sampled τιμή από κάποια κατανομή και ο decoder κάνει reconstruct τα σημεία αυτά που προέρχονται από κάποια κατανομή και άρα είναι regularized. Σαν loss έχουν το κλασικό reconstruction loss και προστίθεται το KL divergence που χρησιμεύει στο να γίνεται το regularization. Για την εκπαίδευση χρησιμοποιείται το reparameterization trick όπου εισάγουμε μία ακόμα μεταβλητή ώστε να γίνεται σωστά η εκπαίδευση και να έχουν νόημα οι παράγωγοι.

Η βασική χρήση των variational autoencoders είναι το image generation και αυτό προσπαθήσαμε να πετύχουμε με τα δικά μας πειράματα. Όπως οι απλοί AE έτσι και αυτοί μπορούν να χτιστούν είτε με Dense είτε με Convolutional αρχιτεκτονικές. Στα πειράματα που έγιναν φάνηκε να υπάρχει πολύ μεγάλο loss και στις 2 αρχιτεκτονικές και η αναπαράσταση νέων εικόνων ή και απλών reconstructed δεν έχει νόημα καθώς δεν επιτεύχθηκε σωστή μάθηση.

**Γενικές παρατηρήσεις**

Οι ρηχοί dense autoencoders φαίνεται να εκπαιδεύονται αρκετά καλά και να έχουν ικανοποιητικά αποτελέσματα σε πολύ λίγο χρόνο ειδικά σε σχέση με απλά dense convolutional δίκτυα. Βαθύτερα dense δίκτυα φαίνεται να έχουν καλά αποτελέσματα αλλά με αυξημένο χρονικό κόστος.

Το image reconstruction του PCA φαίνεται να είναι το πιο αποτελεσματικό αναλογίζοντας τόσο την ακρίβεια των εικόνων όσο και τον χρόνο που χρειάζεται για να επιτευχθεί και την μείωση των διαστάσεων που πετυχαίνει. Σημειώνεται βέβαια πως το PCA δεν είναι τόσο versatile όπως οι AE και για παράδειγμα δεν θα μπορούσε να χρησιμοποιηθεί για να κάνει denoise εικόνες ούτε για image generation.

Για image denoising βλέπουμε μεγάλη διαφορά μεταξύ deep convolutional αρχιτεκτονικών και dense και βλέπουμε οι πρώτες να υπερισχύουν.