Ανίχνευση ανωμαλιών σε δεδομένα υψηλής διάστασης

Βάμβας Ιωάννης

<Διπλωματική Εργασία>

Επιβλέπων: Λυσίμαχος – Παύλος Κόντης

Ιωάννινα, Οκτώβριος, 2021

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UNIVERSITY OF IOANNINA

Περίληψη

Στην ανάλυση δεδομένων, η ανίχνευση ανωμαλιών είναι ο προσδιορισμός σπάνιων στοιχείων, γεγονότων ή παρατηρήσεων που δημιουργούν υποψίες διαφέροντας σημαντικά από την πλειοψηφία των δεδομένων. Συνήθως τα ανώμαλα στοιχεία θα μεταφραστούν σε κάποιο είδος προβλήματος όπως ένα δομικό ελάττωμα ή ως λάθη σε ένα κείμενο. Οι ανωμαλίες επίσης αναφέρονται ως ακραίες τιμές, θόρυβος και αποκλίσεις. Για αυτούς τους λόγους η ανίχνευση ανωμαλιών συγκεντρώνει το ενδιαφέρον πολλών ερευνητών τα τελευταία χρόνια. Ωστόσο, οι περισσότεροι αλγόριθμοι των υφιστάμενων μελετών είναι ανίσχυροι για δεδομένα μεγάλης κλίμακας και μεγάλης διάστασης. Επιπλέον τα ενδιάμεσα δεδομένα που εξάγονται με ορισμένες μεθόδους που μπορούν να χειριστούν δεδομένα υψηλής διάστασης καταλαμβάνουν μεγάλο χώρο αποθήκευσης. Στην παρούσα διπλωματική εργασία ασχολούμαστε με την ανίχνευση ανωμαλιών σε βίντεο. Η μέθοδος βασίζεται σε κρυμμένες αναπαραστάσεις που λαμβάνονται από έναν αυτοκωδικοποιητή μεταβολής, οι οποίες κωδικωποιούνται ως αραιά διανύσματα. Για σύνολα δεδομένων μεγάλης κλίμακας, η αραιή κωδικοποίηση μπορεί να παίξει τον ρόλο της μείωσης των διαστάσεων για τη λήψη κρυφών πληροφοριών και την εξαγωγή πιο υψηλού επιπέδου χαρακτηριστικών. Ταυτόχρονα, για την αποθήκευση κανονικών πληροφοριών, το κόστος του χώρου μπορεί να μειωθεί πολύ. Για να επαληθεύσουμε την ευελιξία και την απόδοση του προτεινόμενου αλγορίθμου μάθησης, χρησιμοποιήσαμε το σύνολο δεδομένων moving MNIST.

Λέξεις κλειδιά: Ανίχνευση ανωμαλιών, αυτοκωδικοποιητής μεταβολής, δίκτυο LISTA

Abstract

In data analysis, anomaly detection is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data. Typically the anomalous items will translate to some kind of problem as a structural defect or errors in a text. Anomalies are also referred to as outliers, noise, deviations and exceptions. For these reasons anomaly detection has attracted the interest of many researchers in recent years. However, most algorithms of the existing studies are powerless for large-scale and high-dimensional data, and the intermediate data extracted by some methods that can handle high-dimensional data, occupy a large storage space. In the present diploma thesis we deal with the anomaly detection in video. The method is based on hidden representations obtained from a variational autoencoder, which are encoded as sparse representations. For large-scale datasets, it can play the role of dimensionality reduction to obtain hidden information, and extract more high-level features than hand-crafted features. At the same time, for the storage of normal information, the space cost can be greatly reduced. To verify the versatility and performance of the proposed learning algorithm, we use the moving MNIST dataset.

Key words: Anomaly detection, variational autoencoder, LISTA network

Πίνακας Περιεχομένων

Κεφάλαιο 1.

Εισαγωγή

Η ανίχνευση μη φυσιολογικών συμβάντων είναι ένα δημοφιλές ερευνητικό θέμα. Αυτή η έρευνα γίνεται όλο και πιο σημαντική λόγω της πολυπλοκότητας των πληροφοριών δικτύου, την καθολικότητα των εφαρμογών και την δυσκολία παρακολούθησης των διαδικασιών . Τα τελευταία χρόνια, κάποιες παραδοσιακές μέθοδοι εκμάθησης λεξικών και εφαρμογής μοντέλων δικτύου που επωφελούνται από την βαθιά μάθηση στον τομέα της ανίχνευσης ανωμαλιών έχουν σημειώσει μεγάλη πρόοδο.

Οι παραδοσιακές μέθοδοι εξάγουν πρώτα χονδροειδή χαρακτηριστικά από πρωτότυπα δεδομένα και στη συνέχεια εκπαιδεύουν μία πιθανοτική κατανομή ή ένα στατιστικό μοντέλο για προσαρμογή σε κανονικά δεδομένα και ανίχνευση των αποκλίσεων των δεδομένων από το εκπαιδευμένο μοντέλο.

Οι μέθοδοι αυτού του είδους είναι πολύ καλές, αλλά υπάρχουν ακόμα περιθώρια βελτίωσης στην επιλογή χαρακτηριστικών, τέτοια ώστε, να γίνει η αλλαγή των χειροποίητων χαρακτηριστικών σε υψηλό επίπεδο.

Όμως, αν και αυτές οι μέθοδοι είναι καλές σε πειραματικό επίπεδο, υπάρχουν δύο προβλήματα που δεν μπορούν να αποφευχθούν. Πρώτον, αν τα πειραματικά δεδομένα δεν είναι αρκετά ή τα δεδομένα είναι ελλιπή, θα επηρεαστεί η προγνωστική ικανότητα του μοντέλου. Δεύτερον, αν η διάσταση των δεδομένων είναι πολύ υψηλή, τότε η υπολογιστική ισχύς του μοντέλου θα είναι ο μεγαλύτερος περιορισμός του μοντέλου. Έτσι, για να αποφύγουμε αυτά τα προβλήματα, οδηγούμαστε στην χρήση των νευρωνικών δικτύων.

Οι μέθοδοι που βασίζονται στην βαθιά μάθηση εστιάζουν στην αναζήτηση των κατάλληλων αναπαραστάσεων των δεδομένων, και πολλές από αυτές έχουν εφαρμοστεί με επιτυχία στο πρόβλημα της ανίχνευσης ανωμαλιών. Προς το παρόν, η εξαγωγή ορισμένων χαρακτηριστικών από κάποιες μεθόδους μπορεί να χωριστεί σε τρεις κατηγορίες: (1) Χαμηλό επίπεδο: τα χαρακτηριστικά χαμηλού επιπέδου εξάγονται άμεσα από την κίνηση ή την εμφάνιση πληροφοριών εισαγωγής, όπως το ιστόγραμμα της κλίσης, η οπτική ροή σε σημεία ενδιαφέροντος, και ιστογράμματα ορίου κίνησης. (2) Ενδιάμεσο επίπεδο: τα χαρακτηριστικά του ενδιάμεσου επίπεδου είναι ελαφρώς πιο σημασιολογικά από αυτά του χαμηλού επιπέδου. Η μέθοδος (που υλοποιούμε), βασίζεται στη βαθιά μάθηση, η οποία επιτυγχάνει καλά αποτελέσματα σε πολλούς τομείς, θεωρείται επίσης πως είναι ένας αποτελεσματικός τρόπος εξαγωγής λειτουργιών υψηλού επιπέδου.

Όπως αποδεικνύεται στα πειράματα, η προτεινόμενη μέθοδος μπορεί να εφαρμοστεί με επιτυχία στο πρόβλημα της ανίχνευσης ανωμαλιών εικόνας.

Επιπρόσθετα, την μέθοδο που φτιάξαμε την εφαρμόσαμε στο πρόβλημα της ανίχνευσης ανωμαλιών σε βίντεο. Χρησιμοποιήσαμε το σύνολο δεδομένων moving MNIST, το οποίο περιέχει βίντεο με χειρόγραφα ψηφία. Το MNIST είναι συντομογραφία του Modified National Institute of Standards and Technology.

Επίσης, το παρών κείμενο αποτελείται από 6 κεφάλαια. Στο 1ο κεφάλαιο υπάρχει μία γενική εισαγωγή για το αντικείμενο του θέματος με το οποίο ασχολούμαστε. Στο 2ο κεφάλαιο περιγράφουμε τα τεχνητά νευρωνικά δίκτυα αλλά και κάποιους όρους που σχετίζονται με αυτά, όπως η οπισθοδρόμηση. Επιπρόσθετα αναλύουμε τους αυτοκωδικοποιητές και τους αυτοκωδικοποιητές μεταβολής. Στο 3ο κεφάλαιο αναλύεται κυρίως η χρήση των λεξικών και του δίκτυου LISTA και η συσχέτιση τους με το πρόβλημα της επεξεργασίας εικόνας. Επιπλέον αναλύονται και οι αραιές αναπαραστάσεις. Στο 4ο

κεφάλαιο γίνεται περιγραφή του αλγορίθμου που υλοποιήθηκε, όπως επίσης και του μοντέλου, δηλαδή των παραμέτρων και της αρχιτεκτονικής του, που στηριχθήκαμε ώστε να υλοποιηθεί ο αλγόριθμος. Στο 5ο κεφάλαιο αναλύονται τα πειράματα που κάναμε για να πιστοποιήσουμε πως το μοντέλο μας δουλεύει όπως θα θέλαμε, όπως επίσης τα σύνολα δεδομένων που χρησιμοποιήσαμε αλλά και οι λεπτομέρειες υλοποίησης των μοντέλων μας. Τέλος, στο 6ο κεφάλαιο γίνεται περιγραφή των συμπερασμάτων που βγάλαμε από τα πειράματα του κεφαλαίου 5 αλλά και γενικά από την λειτουργία του μοντέλου μας.

Κεφάλαιο 2.

Τεχνητά Νευρωνικά Δίκτυα

2.1 Εισαγωγή στα Τεχνητά Νευρωνικά Δίκτυα

Οι ηλεκτρονικοί υπολογιστές κατέστησαν εφικτή την ταχύτατη αναζήτηση και επεξεργασία δεδομένων, με αποτέλεσμα την εμφάνιση ευφυών λειτουργιών από μηχανές κατά την έννοια του Turing, δηλαδή σε συγκεκριμένα προβλήματα μια μηχανή μπορεί να δίνει εξίσου καλές απαντήσεις με ένα άνθρωπο. Μέσα στο συγκεκριμένο πλαίσιο, εμφανίστηκε ένας αριθμός μεθοδολογιών υπό τον κοινό τίτλο υπολογιστική νοημοσύνη. Συγκεκριμένα, ως υπολογιστική νοημοσύνη ορίζουμε μία συνεχώς εξελισσόμενη συνέργια μεθοδολογιών επεξεργασίας αριθμητικών δεδομένων, οι οποίες είναι υλοποιήσιμες σε υπολογιστή για τη λήψη αποφάσεων κοινής λογικής.

Μία κλασική μεθοδολογία υπολογιστικής νοημοσύνης προσομοιώνει παραμετρικά κάποια λειτουργία ενός βιολογικού οργάνου/οργανισμού. Στην περίπτωση μας, τα τεχνητά νευρωνικά δίκτυα, που είναι μία κλασική μεθοδολογία υπολογιστικής νοημοσύνης, προσομοιώνουν την λειτουργία του εγκεφάλου. Ο ανθρώπινος εγκέφαλος αποτελείται από πολλά δισεκατομμύρια απλές μονάδες επεξεργασίας πληροφοριών (όπως τα νευρικά κύτταρα) που ονομάζονται νευρώνες. Κάθε νευρώνας μπορεί να επικοινωνεί με χιλιάδες άλλους μέσω αποφυάδων που ονομάζονται δενδρίτες, μεγάλο μέρος της επιφάνειας των οποίων καλύπτεται από τα συναπτικά άκρα αποφυάδων άλλων νευρώνων. Κατά την λειτουργία του εγκεφάλου ηλεκτρική διέγερση ενός νευρώνα μπορεί να μεταφερθεί προς κάθε νευρώνα με τον οποίο αυτός είναι διασυνδεδεμένος. Έτσι λειτουργεί ο ανθρώπινος εγκέφαλος, εκτελώντας μία παράλληλη και κατανεμημένη επεξεργασία ηλεκτρικών σημάτων. Αυτό κάνει τον ανθρώπινο εγκέφαλο, και ευρύτερα τον άνθρωπο, να είναι πιο αποτελεσματικός από τον ηλεκτρονικό υπολογιστή, σε προβλήματα όπως η αναγνώριση αντικειμένων/καταστάσεων, καθώς και οι συσχετίσεις αυτών. Επίσης, ο άνθρωπος μπορεί να μαθαίνει εμπειρικά ενώ ο ηλεκτρονικός υπολογιστής θα πρέπει να προγραμματιστεί. Αντίθετα, ο ηλεκτρονικός υπολογιστής είναι πιο ακριβής από έναν άνθρωπο στους αριθμητικούς υπολογισμούς και μπορεί να αποθηκεύσει αξιόπιστα μεγάλο όγκο δεδομένων.

Στη συνέχεια, μπορούμε να περάσουμε στην ανάλυση της λειτουργίας ενός τεχνητού νευρωνικού δικτύου. Ουσιαστικά, ένα τεχνητό νευρωνικό δίκτυο υλοποιεί μία συνάρτηση f: $R^N \rightarrow$ T, όπου το πεδίο ορισμού R είναι το σύνολο των πραγματικών αριθμών, ενώ το πεδίο τιμών T μπορεί να είναι είτε T ≡ R^M (σε προβλήματα παλινδρόμησης), είτε ένα σύνολο από ετικέτες (σε προβλήματα αναγνώρισης), όπως εξηγείται στην συνέχεια.

Δοθέντος ενός συνόλου ζευγών (x_1,f(x_1)), … , (x_n,f(x_n)), ο σκοπός ενός τυπικού μοντέλου υπολογιστικής νοημοσύνης είναι να υπολογίσει μία βέλτιστη προσέγγιση $\hat{f} : R^N \rightarrow$ T της συνάρτησης f: $R^N \rightarrow$ T, ώστε για κάθε $x_0 \in R^N$, με $x_0 \neq x_i$, i ∈{1, …, n}, το εκτιμώμενο $\hat{f}(x_0)$ να είναι όσο το δυνατόν πιο «κοντά», υπό κάποια έννοια στο πραγματικό f(x_0). Η διαδικασία υπολογισμού της συνάρτησης $\hat{f} : R^N \rightarrow$ T από το σύνολο των ζευγών (x_1,f(x_1)), … , (x_n,f(x_n)) στο πλαίσιο της υπολογιστικής νοημοσύνης καλείται μάθηση ή εκπαίδευση, ενώ η διαδικασία υπολογισμού της τιμής $\hat{f}(x_0)$, για $x_0 \neq x_i$, i ∈ {1, …, n}, καλείται γενίκευση. Βασικός στόχος κάθε διαδικασίας μάθησης στην υπολογιστική νοημοσύνη είναι μία αποδεκτή ικανότητα γενίκευσης.

Οι εφαρμογές υπολογιστικής νοημοσύνης γενικά χωρίζονται σε τρεις κατηγορίες:

α) Ομαδοποίηση, όπου το μοντέλο υπολογιστικής νοημοσύνης εκπαιδεύεται για να υπολογίζει σμήνη σε δεδομένα x_1, …, x_n χωρίς να δίνονται οι τιμές $f(x_1)$, …,f(x_n).

β) Κατηγοριοποίηση, όπου ένα μοντέλο $\hat{f} : R^N \rightarrow$ T υπολογίζεται για αναγνώριση προτύπων, δηλαδή το T είναι ένα σύνολο από ετικέτες.

γ) Παλινδρόμηση (στατιστική), όπου ένα μοντέλο υπολογιστικής νοημοσύνης $\hat{f} : R^N \rightarrow R^M$ υπολογίζεται για πρόβλεψη

Επιπρόσθετα, αξίζει να αναφερθεί πως ένα τεχνητό νευρωνικό δίκτυο υλοποιεί την συνάρτηση f , που περιγράφτηκε παραπάνω, χρησιμοποιώντας αρχιτεκτονική παρόμοια με την αρχιτεκτονική του ανθρώπινου εγκεφάλου. Όποτε τώρα μπορούμε να περάσουμε και στην ανάλυση της αρχιτεκτονικής ενός τεχνητού νευρωνικού δικτύου. Μία συγκεκριμένη αρχιτεκτονική τεχνητού νευρωνικού δικτύου τυπικά αναλύεται και σχεδιάζεται όπως περιγράφεται στη συνέχεια.

Το σχήμα 1.1 παρουσιάζει το μοντέλο ενός τεχνητού νευρώνα. Οι αριθμητικές είσοδοι $x_1, \ldots, x_n$ του τεχνητού νευρώνα πολλαπλασιάζονται με τα βάρη $w_1, \ldots, w_n$ αντίστοιχα και στη συνέχεια αθροίζονται στη συσκευή του αθροιστή (Σ). Μία επιπλέον είσοδος του αθροιστή Σ είναι σταθερή, δηλαδή $x_{n+1} = 1$ και πολλαπλασιάζεται με τη σταθερά πόλωσης β, η οποία αποτελεί το n+1 βάρος του τεχνητού νευρώνα, δηλαδή $w_{n+1} = \beta$. Συνεπώς, η έξοδος (σ) του αθροιστή υπολογίζεται ως εξής:

$\sigma = \sum_{i=1}^{n} w_i x_i + \beta = \sum_{i=1}^{n+1} w_i x_i = w^T \cdot x$, όπου το $w^T \cdot x$ παριστάνει το εσωτερικό γινόμενο του διανύσματος $x = (x_1, \ldots., x_i, 1)^T$ εισόδου του τεχνητού νευρώνα επί το διάνυσμα $w = (w_1, \ldots., w_n, w_{n+1})^T$ των βαρών.

Το σταθμισμένο (γραμμικό) άθροισμα σ των εισόδων του τεχνητού νευρώνα στη συνέχεια οδηγείται σε ένα (μη-γραμμικό) στοιχείο παραμόρφωσης f(σ), που ονομάζεται συνάρτηση ενεργοποίησης (activation function), όπως φαίνεται στο σχήμα 1.1. Μερικές από τις συναρτήσεις μεταφοράς f(σ) που έχουν προταθεί στην βιβλιογραφία παρουσιάζονται στην συνέχεια.

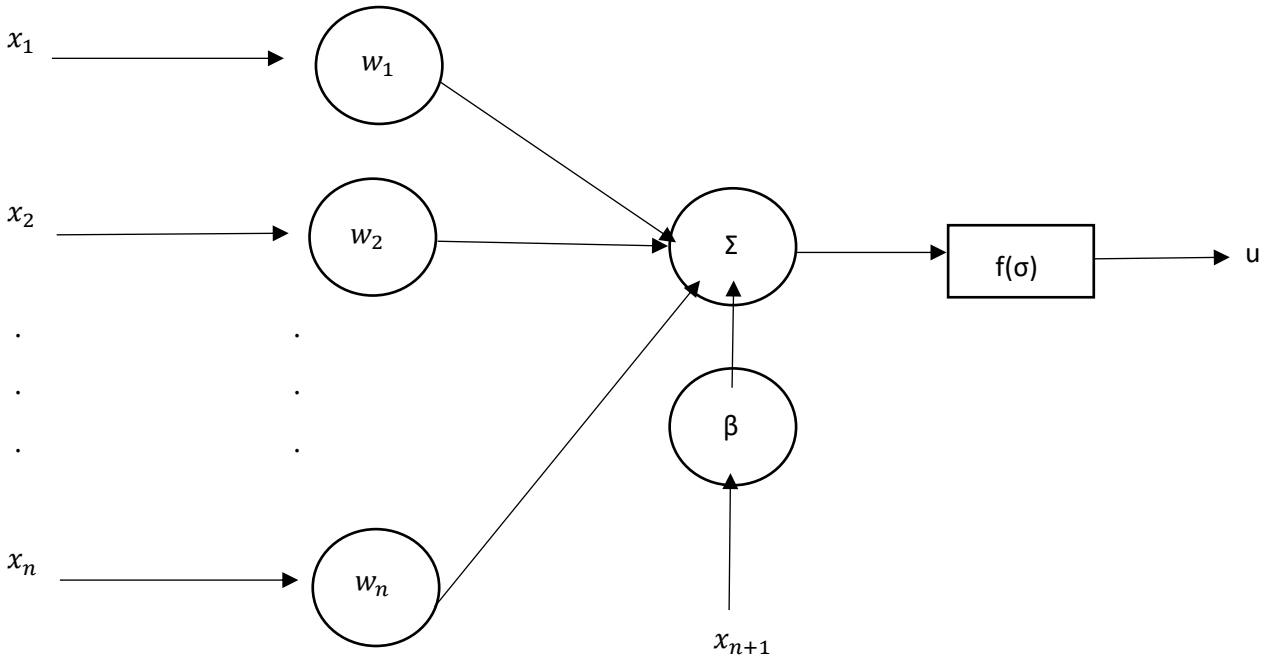
α) Γραμμική συνάρτηση ενεργοποίησης: f(σ) = σ

β) Συνάρτηση ενεργοποίησης τύπου perceptron: $f(\sigma) = \begin{cases} \sigma, & \sigma \geq 0 \\ 0, & \sigma < 0 \end{cases}$

γ) Δυαδική (δίτιμη) συνάρτηση ενεργοποίησης με κατώφλι T: $f(\sigma) = \begin{cases} 1, & \sigma \geq T \\ 0, & \sigma < T \end{cases}$

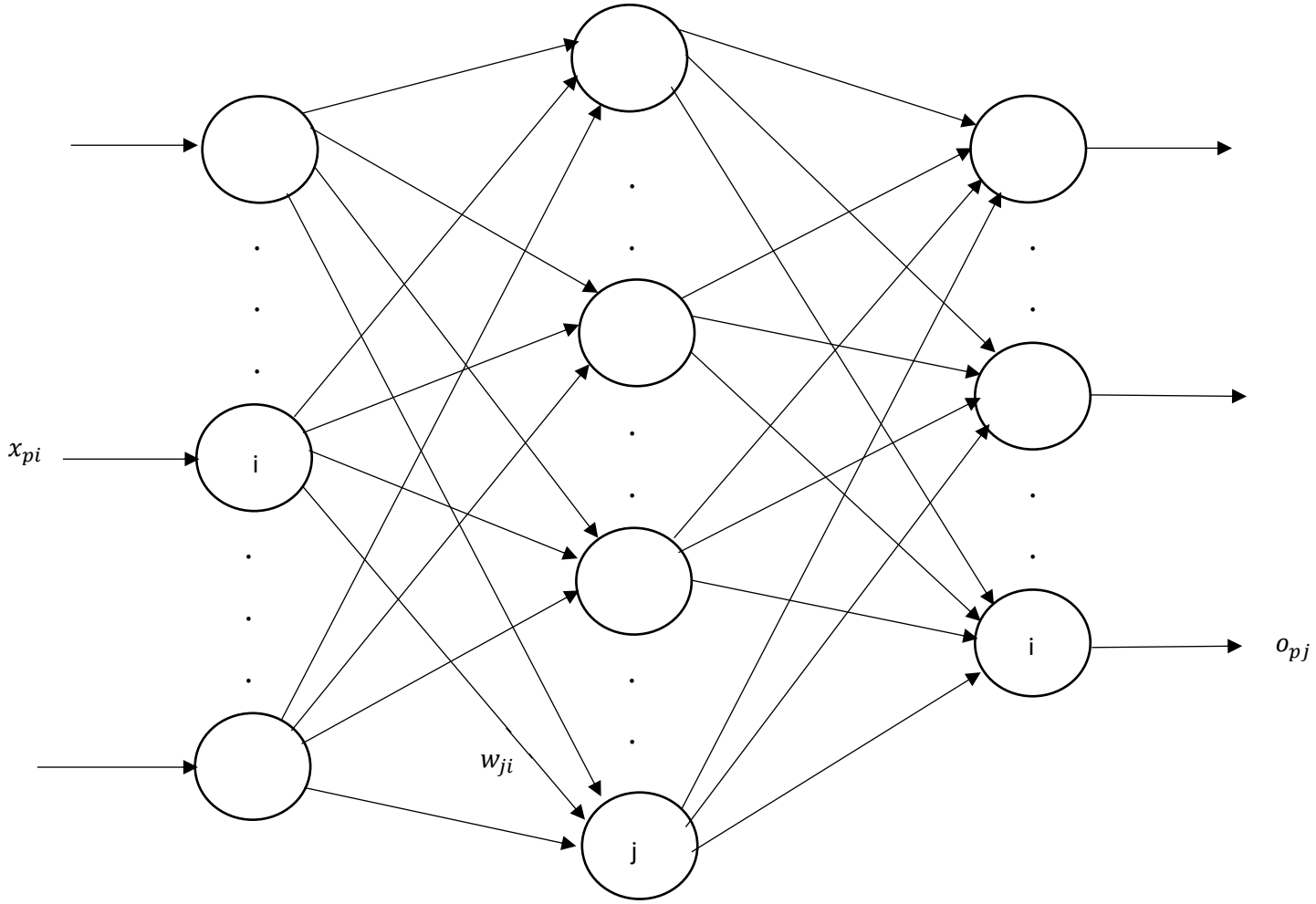
δ) Σιγμοειδής (ή λογιστική) συνάρτηση ενεργοποίησης: $f(\sigma) = \frac{1}{1+e^{-\sigma}}$

ε) Συνάρτηση ενεργοποίησης τύπου υπερβολικής εφαπτομένης: $f(\sigma) = \frac{1-e^{-\sigma}}{1+e^{-\sigma}}$



Σχήμα 2.1: Μοντέλο τεχνητού νευρώνα

Επιπλέον, μετά την παρουσίαση του μοντέλου ενός τεχνητού νευρώνα, μπορούμε να περάσουμε και στην διασύνδεση των νευρώνων μεταξύ τους μέσα σε ένα τεχνητό νευρωνικό δίκτυο. Συγκεκριμένα, θεωρούμε ότι οι νευρώνες ενός τεχνητού νευρωνικού δικτύου διατάσσονται σε διαδοχικά στρώματα τα οποία αλληλοσυνδέονται πλήρως όπως φαίνεται στο σχήμα 2.2, όπου απεικονίζεται ένα προσωτροφοδοτούμενο τεχνητό νευρωνικό δίκτυο τριών στρωμάτων με την πληροφορία (βλ. αριθμούς) να μεταδίδεται από τα χαμηλότερα προς τα υψηλότερα στρώματα. Επίσης, ένας νευρώνας σε κάθε στρώμα είναι πλήρως διασυνδεδεμένος υπό την έννοια ότι διασυνδέεται μέσω κάποιου κατευθυνόμενου συνδέσμου με κάθε νευρώνα του επόμενου στρώματος (αν υπάρχει τέτοιο στρώμα), αλλά και με κάθε νευρώνα του αμέσως προηγούμενου στρώματος (αν υπάρχει τέτοιο στρώμα). Αξίζει να σημειωθεί πως κανένας νευρώνας του στρώματος εισόδου δεν έχει συνάρτηση ενεργοποίησης, ενώ κάθε άλλος νευρώνας έχει συνάρτηση ενεργοποίησης και πως διαφορετικές συναρτήσεις ενεργοποίησης θα μπορούσαν να θεωρηθούν για διαφορετικούς νευρώνες.



Σχήμα 2.2: Προσωτροφοδοτούμενο τεχνητό νευρωνικό δίκτυο τριών στρωμάτων νευρώνων. Η είσοδος που είναι το διάνυσμα αριθμών x_p μεταδίδεται από το χαμηλότερα προς τα υψηλότερα στρώματα (στην περίπτωση μας από τα αριστερά προς τα δεξιά) κατά την φορά που δείχνουν τα βέλη. Στο στρώμα εξόδου υπολογίζεται το διάνυσμα αριθμών o_p. Με w_{ji} αναγράφεται το βάρος του κατευθυνόμενου συνδέσμου w_{ji} που καταλήγει στον νευρώνα j, ξεκινώντας από τον νευρώνα i του προηγούμενου στρώματος.

Η λεπτομερής υλοποίηση ενός νευρώνα που φαίνεται στο σχήμα 2.2 είναι αυτή που φαίνεται στο σχήμα 2.1. Ένας νευρώνας, έστω i, του στρώματος εισόδου δεν κάνει κανέναν υπολογισμό, παρά μόνο προωθεί την αντίστοιχη αριθμητική του είσοδο x_{pi} (του διανύσματος x_p εισόδου) προς κάθε νευρώνα του κρυμμένου στρώματος. Σημειώστε ότι κάθε κατευθυνόμενος σύνδεσμος έχει ένα συγκεκριμένο βάρος (w).

Επιπρόσθετα, μάθηση ενός τεχνητού νευρωνικού δικτύου ονομάζεται η διαδικασία υπολογισμού των βαρών των συνδέσμων ενός τεχνητού νευρωνικού δικτύου. Υπάρχουν δύο τύποι μάθησης τεχνητού νευρωνικού δικτύου που περιλαμβάνουν α) μάθηση με εποπτεία ή επίβλεψη, η οποία χρησιμοποιείται τυπικά για κατηγοριοποίηση και β) μάθηση χωρίς εποπτεία, η οποία τυπικά χρησιμοποιείται για την ομαδοποίηση. Συγκεκριμένα, στη μάθηση με εποπτεία το τεχνητό νευρωνικό δίκτυο μαθαίνει συγκεκριμένα ζεύγη διανυσμάτων (x_i, t_i), i ∈ {1, … , n} (εισόδου, εξόδου) υπό την εξής έννοια. Όταν το διάνυσμα x_i εφαρμοστεί στην είσοδο του εκπαιδευμένου τεχνητού νευρωνικού δικτύου, τότε υπολογίζεται το αντίστοιχο διάνυσμα t_i, i ∈{1, … , n} στην έξοδο του τεχνητού νευρωνικού δικτύου.

Όμως πολλές φορές δεν είναι δυνατός ο ακριβής υπολογισμός του διανύσματος t_i, οπότε μεθοδεύουμε μία εκπαίδευση του τεχνητού νευρωνικού δικτύου τέτοια ώστε να υπολογίζεται το διάνυσμα o_i, το οποίο αποτελεί μία βέλτιστη προσέγγιση του διανύσματος t_i , i ∈ {1, … , n} μέσα στο πλαίσιο των ελαχίστων τετραγώνων. Αντίθετα, στην μάθηση χωρίς εποπτεία είναι διαθέσιμα τα διανύσματα x_i, i ∈ {1, … , n} για εκπαίδευση, αλλά δεν είναι διαθέσιμα τα διανύσματα t_i, i ∈ {1, … , n}.

Τέλος, με βάση τα παραπάνω αλλά και από την βιβλιογραφία μπορούμε να καταλήξουμε σε κάποια πλεονεκτήματα των τεχνητών νευρωνικών δικτύων, τα οποία είναι τα εξής:

1) Μάθηση από εμπειρία (βλ. καταγεγραμμένα δεδομένα) συναρτήσεων εισόδου-εξόδου. Συγκεκριμένα, η μάθηση των τεχνητών νευρωνικών δικτύων πραγματοποιείται με τη μεταβολή των βαρών των συνδέσμων, έτσι ώστε να ελαχιστοποιείται ένα καλώς ορισμένο σφάλμα.
2) Προσέγγιση μη γραμμικών συναρτήσεων εισόδου-εξόδου.
3) Ανεκτικότητα σε βλάβες λόγω της παράλληλης δομής και λειτουργίας τους.
4) Ικανότητα γενίκευσης.
5) Κατανεμημένη και παράλληλη τοπολογία.

2.2 Οπισθόδρομη μάθηση (Δίκτυα οπισθόδρομης μάθησης)

Στην μηχανική μάθηση, η οπισθοδρόμηση είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος για την εκπαίδευση προσωτροφοδοτούμενων νευρωνικών δικτύων. Υπάρχουν γενικεύσεις της οπισθοδρόμησης για άλλα τεχνητά νευρωνικά δίκτυα και για συναρτήσεις γενικά. Αυτές οι κατηγορίες αλγορίθμων αναφέρονται γενικά ως «οπισθοδρόμηση». Κατά την προσαρμογή ενός νευρωνικού δικτύου, η οπισθοδρόμηση υπολογίζει την κλίση της συνάρτησης λάθους με σεβασμό στα βάρη του δικτύου για ένα μόνο παράδειγμα εισόδου-εξόδου, και το κάνει αποτελεσματικά, σε αντίθεση με έναν απλό μέσο υπολογισμό της κλίσης με σεβασμό σε κάθε βάρος ξεχωριστά. Αυτή η αποτελεσματικότητα καθιστά εφικτή την χρήση μεθόδων κλίσης για την εκπαίδευση δικτύων πολλαπλών επιπέδων, την ενημέρωση βαρών για την ελαχιστοποίησης της απώλειας. Ο gradient descent όπως και ο στοχαστικός gradient descent χρησιμοποιούνται συχνά. Ο αλγόριθμος οπισθοχώρησης λειτουργεί υπολογίζοντας την κλίση της συνάρτησης απώλειας με σεβασμό σε κάθε βάρος από τον κανόνα της αλυσίδας, υπολογίζοντας την κλίση ενός επιπέδου κάθε φορά, επαναλαμβάνοντας προς τα πίσω από το τελευταίο επίπεδο για να αποφευχθούν περιττοί υπολογισμοί των ενδιάμεσων όρων στον κανόνα της αλυσίδας. Αυτό είναι ένα παράδειγμα δυναμικού προγραμματισμού.

Επιπρόσθετα, ο όρος οπισθοδρόμηση αναφέρεται αυστηρά στον αλγόριθμο για τον υπολογισμό της κλίσης και όχι στον τρόπο με τον οποίο χρησιμοποιείται η κλίση. Ωστόσο, ο όρος χρησιμοποιείται για να αναφερθεί σε ολόκληρο τον αλγόριθμο εκμάθησης, συμπεριλαμβανομένου του τρόπου με τον οποίο χρησιμοποιείται η κλίση, όπως στον στοχαστικό gradient descent. Η οπισθοδρόμηση γενικεύει τον υπολογισμό της κλίσης από τον κανόνα δέλτα, που είναι η έκδοση της οπισθοχώρησης ενός επιπέδου, και με τη σειρά του γενικεύεται με αυτόματη διαφοροποίηση, όπου η οπισθοχώρηση είναι μία ειδική περίπτωση αντιστροφής συσσώρευσης (ή “αντίστροφη λειτουργεία”).

Στη συνέχεια, αυτή η ενότητα περιγράφει έναν τρόπο μάθησης τεχνητού νευρωνικού δικτύου με οπισθόδρομη μάθηση των βαρών, ώστε να ελαχιστοποιείται το τετραγωνικό σφάλμα εξόδου. Η συγκεκριμένη τεχνική, έγινε αργότερα γνωστή ως γενικευμένος κανόνας δέλτα και θα γίνει περιγραφή της στην επόμενη παράγραφο.

Έστω ότι εφαρμόζουμε το διάνυσμα x_p στην είσοδο ενός τεχνητού νευρωνικού δικτύου τριών στρωμάτων νευρώνων (βλ. σχήμα 2.2) και επιθυμούμε να υπολογιστεί το διάνυσμα t_p στην έξοδο. Δεδομένων των συγκεκριμένων τιμών που έχουν τα βάρη, στην έξοδο του τεχνητού νευρωνικού

δικτύου θα υπολογιστεί διάνυσμα o_p το οποίο στην γενική περίπτωση θα είναι $o_p \neq t_p$. Το ερώτημα είναι πως να μεταβάλουμε το κάθε βάρος w_{ji} του τεχνητού νευρωνικού δικτύου ώστε να μειωθεί το τετραγωνικό σφάλμα εξόδου $E_p = \frac{1}{2}\sum_j (t_{pj} - o_{pj})^2$ και συνεπώς το διάνυσμα o_p να προσεγγίσει το επιθυμητό διάνυσμα t_p.

Το σφάλμα E_p είναι συνάρτηση όλων των βαρών w_{ji} του τεχνητού νευρωνικού δικτύου. Προκειμένου να μειωθεί το σφάλμα, αρκεί να μεταβληθεί το κάθε βάρος w_{ji} του τεχνητού νευρωνικού δικτύου κατά μία μικρή ποσότητα $\Delta_p w_{ji}$ ανάλογη με την παράγωγο $-\frac{\partial E_p}{\partial w_{ji}}$. Θεωρούμε πως η παράγωγος που αναλύθηκε στην προηγούμενη πρόταση υπάρχει και την υπολογίζουμε στην συνέχεια. Πιο συγκεκριμένα, $-\frac{\partial E_p}{\partial w_{ji}} = -\frac{\partial E_p}{\partial \sigma_{pj}} \frac{\partial \sigma_{pj}}{\partial w_{ji}}$, όπου $\sigma_{pj} = \sum_k (w_{jk} o_{pk})$. Δηλαδή, το σ_{pj} είναι η έξοδος του αθροιστή (βλ. Σχήμα 2.1) του νευρώνα j, w_{jk} είναι το βάρος του συνδέσμου που καταλήγει στον νευρώνα j ξεκινώντας από τον νευρώνα k του προηγούμενου στρώματος και o_{pk} είναι η έξοδος του νευρώνα k. Ως αποτέλεσμα των παραπάνω, τώρα έχουμε πως $-\frac{\partial E_p}{\partial w_{ji}} = -\frac{\partial E_p}{\partial \sigma_{pj}} \frac{\partial}{\partial w_{ji}} \sum_k (w_{jk} o_{pk}) = -\frac{\partial E_p}{\partial \sigma_{pj}}\, o_{pi}$. Η ποσότητα $-\frac{\partial E_p}{\partial \sigma_{pj}}$ ονομάζεται δέλτα (δ_{pj}) και χαρακτηρίζει τον νευρώνα j. Τελικά, $\Delta_p w_{ji} = \eta \delta_{pj} o_{pi}$, όπου η σταθερά "η" είναι ένας μικρός αριθμός. Το πρόβλημα τώρα είναι να υπολογιστεί η ποσότητα δ_{pj} όπως παρουσιάζεται στη συνέχεια:

$\delta_{pj} = -\frac{\partial E_p}{\partial \sigma_{pj}} = -\frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial \sigma_{pj}}$, όπου $o_{pj} = f_j(\sigma_{pj})$. Άρα, $\delta_{pj} = -\frac{\partial E_p}{\partial o_{pj}} f_j{}'(\sigma_{pj})$, όπου $f_j{}'(\sigma_{pj})$ είναι η παράγωγος της συνάρτησης f(σ_{pj}) ως προς σ_{pj}. Διακρίνουμε 2 περιπτώσεις:

1) ο νευρώνας j είναι στο στρώμα εξόδου. Τότε έχουμε:

$-\frac{\partial E_p}{\partial o_{pj}} = -\frac{\partial}{\partial o_{pj}}\left[\frac{1}{2}\sum_j (t_{pj} - o_{pj})^2\right] = (t_{pj} - o_{pj})$. Οπότε $\delta_{pj} = (t_{pj} - o_{pj}) f_j{}'(\sigma_{pj})$.

2) ο νευρώνας j είναι στο κρυμμένο στρώμα. Τότε έχουμε:

Το $-\frac{\partial E_p}{\partial o_{pj}}$ είναι ανάλογο του όρου $-\sum_k (\frac{\partial E_p}{\partial \sigma_{pk}} \frac{\partial \sigma_{pk}}{\partial o_{pj}})$ και έτσι καταλήγουμε πως

$-\frac{\partial E_p}{\partial o_{pj}} = -\sum_k (\frac{\partial E_p}{\partial \sigma_{pk}} \frac{\partial}{\partial o_{pj}} \left[\sum_i (w_{ki} * o_{pi})\right]) = \sum_k (\frac{-\partial E_p}{\partial \sigma_{pk}} w_{kj}) = \sum_k (\delta_{pk} w_{kj})$. Άρα, το δ_{pj} είναι ανάλογο του $\sum_k (\delta_{pk} w_{kj})\, f_j{}'(\sigma_{pj})$.

Σημειώστε ότι ο γενικευμένος κανόνας δέλτα βελτιστοποιεί αναλυτικά τα βάρη ενός τεχνητού νευρωνικού δικτύου.

2.3 Μέθοδος καθόδου με βάση την κλίση (Gradient Descent)

Η μέθοδος καθόδου με βάση την κλίση είναι ένας επαναληπτικός αλγόριθμος βελτιστοποίησης πρώτης τάξης για την εύρεση ενός τοπικού ελαχίστου από μία παραγωγίσιμη συνάρτηση. Η ιδέα είναι να κάνουμε συνεχόμενα βήματα στην αντίθετη κατεύθυνση της κλίσης της συνάρτησης στο τρέχον σημείο, επειδή αυτή είναι η κατεύθυνση της απότομης καθόδου. Αντίθετα, κάνοντας βήματα προς την κατεύθυνση της κλίσης, τότε θα οδηγηθούμε σε ένα τοπικό μέγιστο της συνάρτησης. Η διαδικασία αυτή είναι γνωστή ως gradient ascent.

Επίσης, η μέθοδος καθόδου με βάση την κλίση στηρίζεται στην παρατήρηση πως η συνάρτηση πολλών μεταβλητών F(x) είναι ορισμένη και παραγωγίσιμη σε μία γειτονιά του σημείου α και στη συνέχεια η

F(x) μειώνεται ταχύτερα, εάν κάποιος πάει από το a προς την κατεύθυνση της αρνητικής κλίσης της F στο α, -$\nabla F(a)$. Επομένως, αν:

$a_{n+1} = a_n - \gamma \nabla F(a_n)$

για ένα αρκετά μικρό $\gamma \in R_+$, τότε έχουμε πως $F(a_n) \geq F(a_{n+1})$. Με άλλα λόγια, ο όρος $\gamma \nabla F(a_n)$ αφαιρείται από τον a γιατί θέλουμε να μετακινηθούμε αντίθετα από την κλίση, προς το τοπικό ελάχιστο. Έχοντας κατά νου αυτή την παρατήρηση, κάνουμε μία εικασία x_0 για ένα τοπικό ελάχιστο F, και θεωρώντας ακολουθία $x_0, x_1, x_2, \ldots$ τέτοια ώστε:

$x_{n+1} = x_n - \gamma_n \nabla F(x_n)$, $n \geq 0$

Έχουμε μία μονότονη ακολουθία:

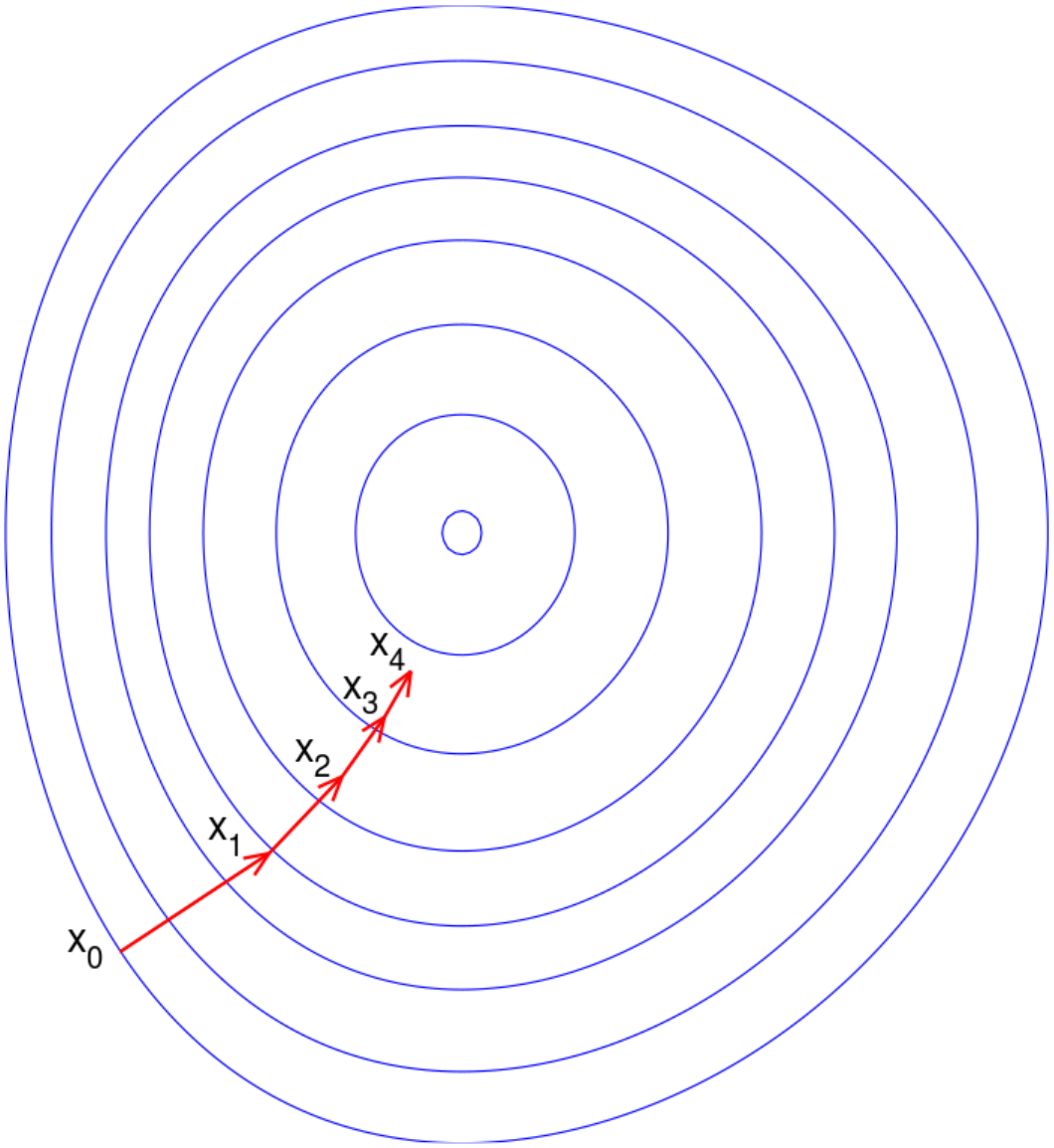
$F(x_0) \geq F(x_1) \geq F(x_2) \geq \ldots$,

οπότε θεωρητικά η ακολουθία (x_n) συγκλίνει στο επιθυμητό τοπικό ελάχιστο. Σημειώστε πως η τιμή του βήματος γ επιτρέπεται να αλλάξει σε κάθε επανάληψη. Με ορισμένες παραδοχές σχετικά με την συνάρτηση F (για παράδειγμα ότι η F είναι κυρτή και το ∇F ικανοποιεί την συνθήκη Lipschitz) και συγκεκριμένες επιλογές για το γ (για παράδειγμα, είτε πως επιλέχθηκε μέσω μίας γραμμικής αναζήτησης που ικανοποιεί τις συνθήκες Wolfe είτε από την μέθοδο Barzilai-Borwein που θα δείξουμε στη συνέχεια),

$$\gamma_n = \frac{|(x_n - x_{n-1})^T [\nabla F(x_n) - \nabla F(x_{n-1})]|}{||\nabla F(x_n) - \nabla F(x_{n-1})||^2}$$

μπορεί να διασφαλιστεί η σύγκλιση στο τοπικό ελάχιστο. Όταν η F είναι κυρτή, τότε όλα τα τοπικά ελάχιστα είναι και ολικά ελάχιστα, όποτε σε αυτή την περίπτωση η μέθοδος καθόδου με βάση την κλίση μπορεί να εξασφαλίσει μία ολική λύση.

Αυτή η διαδικασία απεικονίζεται στο σχήμα 2.3. Εδώ η συνάρτηση F θεωρείται ότι ορίζεται στο επίπεδο και ότι το γράφημα έχει τη μορφή καμπύλης Ball. Οι μπλε καμπύλες είναι οι γραμμές περιγράμματος, δηλαδή, οι περιοχές στις οποίες η συνάρτηση F είναι σταθερή. Ένα κόκκινο βέλος που προέρχεται από ένα σημείο δείχνει την κατεύθυνση της αρνητικής κλίσης σε αυτό το σημείο. Σημειώστε ότι η αρνητική κλίση σε ένα σημείο είναι ορθογώνια προς την γραμμή περιγράμματος που διέρχεται από αυτό το σημείο. Βλέπουμε ότι ο gradient descent μας οδηγεί στο κάτω μέρος της καμπύλης, δηλαδή, στο σημείο όπου η τιμή της συνάρτησης F είναι ελάχιστη.



Σχήμα 2.3: Απεικόνιση του gradient descent πάνω σε μία σειρά από σύνολα επιπέδων

Τέλος, αξίζει να αναφερθεί πως η διαδικασία gradient descent εμφανίζεται με τη μορφή του βήματος οπισθοχώρησης όπου υπολογίζουμε τα διανύσματα σφάλματος δ προς τα πίσω, ξεκινώντας από το τελευταίο επίπεδο. Ανάλογα με την συνάρτηση ενεργοποίησης, προσδιορίζουμε πόση αλλαγή απαιτείται παίρνοντας την μερική παράγωγο της συνάρτησης με ως προς το βάρος (w). Η τιμή αλλαγής πολλαπλασιάζεται με το ποσοστό εκμάθησης. Ως μέρος της εξόδου, αφαιρούμε αυτή την τιμή από την προηγούμενη έξοδο για να πάρουμε την ενημερωμένη τιμή. Συνεχίζουμε αυτή την διαδικασία μέχρι να φτάσουμε στην σύγκλιση.

2.4 Αυτοκωδικοποιητής

Ένας αυτοκωδικοποιητής είναι ένας τύπος τεχνητού νευρωνικού δικτύου που χρησιμοποιείται για την αποτελεσματική εκμάθηση κωδικοποιήσεων δεδομένων με έναν μη εποπτευόμενο τρόπο. Ο στόχος ενός αυτοκωδικοποιητή είναι να μάθει μία αναπαράσταση (κωδικοποίηση) για ένα σύνολο δεδομένων, συνήθως για μείωση διάστασης, εκπαιδεύοντας το δίκτυο να αγνοεί το θόρυβο. Εκτός της μείωσης, γίνεται εκμάθηση και της ανακατασκευής, όπου ο αυτοκωδικοποιητής προσπαθεί να δημιουργήσει από την μειωμένη κωδικοποιήση μία αναπαράσταση όσο το δυνατόν πιο κοντά στην αρχική του είσοδο, εξ' ου και το όνομα της (ανακατασκευή). Υπάρχουν παραλλαγές, οι οποίες στοχεύουν στην παραγωγή αναπαραστάσεων που πληρούν κάποιες ιδιότητες. Παραδείγματα είναι οι κανονικοποιημένοι αυτοκωδικοποιητές (αραιός, denoising και συμβατικός), οι οποίοι είναι αποτελεσματικοί στην εκμάθηση αναπαραστάσεων για ζητήματα ταξινόμησης, και οι αυτοκωδικοποιτές μεταβολής, με εφαρμογές ως γενετικά μοντέλα. Γενικά, οι αυτοκωδικοποιητές εφαρμόζονται σε πολλά προβλήματα, από την αναγνώριση προσώπου έως και την εύρεση της σημασιολογίας των λέξεων (δηλαδή τι σημαίνει μία λέξη).

Στη συνέχεια, περνάμε στην ανάλυση της αρχιτεκτονικής ενός αυτοκωδικοποιητή. Η απλούστερη μορφή ενός αυτοκωδικοποιητή είναι ένα προσωτροφοδοτούμενο, μη επαναλαμβανόμενο νευρωνικό δίκτυο παρόμοιο με τα ενός επιπέδου perceptrons που συμμετέχουν σε πολυεπίπεδα perceptrons (multilayer perceptron - MLP), χρησιμοποιώντας ένα επίπεδο εισόδου και ένα επίπεδο εξόδου που είναι συνδεδεμένο με ένα ή περισσότερα κρυμμένα επίπεδα. Το επίπεδο εξόδου έχει τον ίδιο αριθμό κόμβων (νευρώνων) με το επίπεδο εισόδου. Σκοπός του αυτοκωδικοποιητή είναι να ανακατασκευάσει τις εισόδους του (ελαχιστοποιώντας τη διαφορά της εισόδου με την έξοδο) αντί να προβλέψει επιθυμητή τιμή Y για δεδομένες εισόδους X. Επομένως, οι αυτοκωδικοποιητές είναι μη ελεγχόμενα μοντέλα μάθησης.

Ένας αυτοκωδικοποιητής αποτελείται από δύο μέρη, τον κωδικοποιητή και τον αποκωδικοποιητή, που μπορούν να οριστούν ως μεταβάσεις φ και ψ τέτοιες ώστε:

$\varphi: X \rightarrow F$

$\psi: F \rightarrow X$

$\varphi, \psi := \arg\min_{\varphi,\psi} ||X - (\psi \circ \varphi)X||^2$

Στην απλούστερη περίπτωση, με δεδομένο πως έχουμε ένα κρυφό επίπεδο, ο κωδικοποιητής του αυτοκωδικοποιητή παίρνει την είσοδο $x \in R^d = X$ και παράγει το $h \in R^d = F$:

$h = \sigma(Wx + b)$

Αυτή η εικόνα h συνήθως αναφέρεται ως κώδικας, μεταβλητή κρυμμένου χώρου κωδικοποίησης ή αναπαράσταση κρυμμένου χώρου κωδικοποίησης. Εδώ, το σ είναι μία συνάρτηση ενεργοποίησης όπως η σιγμοειδής συνάρτηση. Το W είναι ένας πίνακας βαρών και το b είναι το διάνυσμα πόλωσης. Τα βάρη και οι πολώσεις συνήθως αρχικοποιούνται τυχαία και στη συνέχεια ενημερώνονται επαναληπτικά κατά την διάρκεια της εκπαίδευσης μέσω της οπισθοδρόμησης (backpropagation). Μετά από αυτό, ο αποκωδικοποιητής του αυτοκωδικοποιητή οδηγεί το h στην ανακατασκευή x′, που έχει την ίδια διάσταση με το x:

$x' = \sigma'(W'h + b')$

όπου τα σ′, W′και b′ για τον αποκωδικοποιητή μπορεί να μην συσχετίζονται με τα σ, W και b του κωδικοποιητή.

Οι αυτοκωδικοποιητές εκπαιδεύονται για να ελαχιστοποιήσουν τα σφάλματα ανακατασκευής (όπως τα τετραγωνικά σφάλματα), που συχνά αναφέρονται ως «απώλεια»:

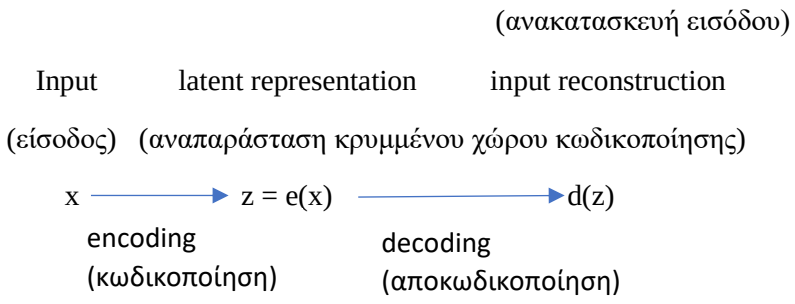
$L(x, x') = ||x - x'||^2 = ||x - \sigma'(W'(\sigma(Wx + b)) + b')||^2$

όπου το x συνήθως υπολογίζεται ως μέσος όρος κάποιων στοιχείων εισόδου του συνόλου εκπαίδευσης.

Όπως αναφέρθηκε προηγουμένως, η εκπαίδευση ενός αυτοκωδικοποιητή πραγματοποιείται μέσω οπισθοδρόμησης του λάθους, ακριβώς όπως σε ένα κανονικό προσωτροφοδοτούμενο νευρωνικό δίκτυο.

Αν ο χώρος κωδικοποίησης F έχει χαμηλότερη διάσταση από τον χώρο εισόδου X, το χαρακτηριστικό διάνυσμα φ(x) μπορεί να θεωρηθεί ως συμπιεσμένη αναπαράσταση της εισόδου. Αυτή είναι η περίπτωση των μη ολοκληρωμένων αυτοκωδικοποιητών. Αν τα κρυμμένα επίπεδα είναι μεγαλύτερα από το επίπεδο εισόδου (υπερπλήρης αυτοκωδικοποιητής), ή ίσα με το επίπεδο εισόδου ή οι κρυμμένες μονάδες έχουν αρκετή χωρητικότητα, ένας αυτοκωδικοποιητής μπορεί δυνητικά να μάθει την ταυτοτική συνάρτηση και να γίνει άχρηστος. Ωστόσο, τα πειραματικά αποτελέσματα έχουν δείξει ότι οι αυτοκωδικοποιητές ενδέχεται να εξακολουθούν να μαθαίνουν χρήσιμες λειτουργίες και σε αυτές τις περιπτώσεις. Στην ιδανική κατάσταση, κάποιος θα πρέπει να μπορεί να προσαρμόζει την διάσταση του κώδικα (ή του κρυμμένου επιπέδου κωδικοποίησης) και τη χωρητικότητα του μοντέλου με βάση την πολυπλοκότητα της κατανομής δεδομένων που θα μοντελοποιηθεί. Ένας τρόπος για να γίνει αυτό είναι να εκμεταλλευτούμε τις παραλλαγές του μοντέλου που είναι γνωστές ως κανονικοποιημένοι αυτοκωδικοποιητές.

Τέλος, ακολουθεί η σχηματική αναπαράσταση λειτουργίας ενός αυτοκωδικοποιητή:



2.5 Αυτοκωδικοποιητής Μεταβολής

Η ανάγκη χρήσης των αυτοκωδικοποιητών μεταβολής (variational autoencoders-VAE) πηγάζει από το γεγονός πως οι αυτόματοι κωδικοποιητές έχουν περιορισμούς ως προς την δημιουργία περιεχομένου. Πιο συγκεκριμένα όταν κατασκευάζουμε έναν αυτόματο κωδικοποιητή, έχουμε έναν κωδικοποιητή και έναν αποκωδικοποιητή, όμως δεν υπάρχει τρόπος για την παραγωγή νέου περιεχομένου. Μία λύση θα μπορούσε να ήταν να πάρουμε ένα σημείο τυχαία από τον χώρο κρυμμένης κωδικοποίησης (latent space), θεωρώντας πως αυτός ο χώρος είναι αρκετά κανονικός, και να το αποκωδικοποιήσουμε για να πάρουμε ένα νέο περιεχόμενο. Όμως αυτή η λύση φαίνεται πως δεν είναι εφικτή γιατί η κανονικότητα του χώρου κρυμμένης κωδικοποίησης για έναν αυτόματο κωδικοποιητή είναι ένα δύσκολο σημείο που εξαρτάται από την κατανομή των δεδομένων στον αρχικό χώρο, τη διάσταση του χώρου κρυμμένης κωδικοποίησης και την αρχιτεκτονική του κωδικοποιητή. Έτσι είναι σχεδόν αδύνατο να διασφαλιστεί πως ο κωδικοποιητής θα οργανώσει αρκετά έξυπνα τον χώρο κρυμμένης κωδικοποίησης ώστε να μπορεί να γίνει η διαδικασία που αναλύθηκε παραπάνω. Οπότε οδηγούμαστε στην χρήση των VAE.

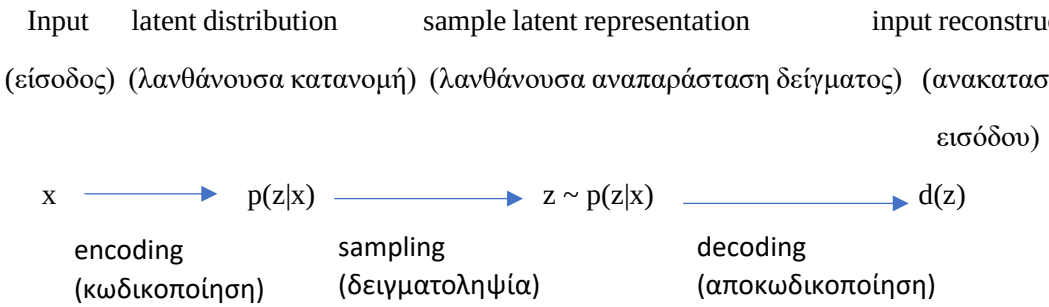
Ένα VAE μπορεί να οριστεί ως ένας αυτόματος κωδικοποιητής του οποίου η εκπαίδευση είναι κανονικοποιημένη για να αποφευχθεί το πρόβλημα της αποστήθισης και να διασφαλιστεί πως ο χώρος κρυμμένης κωδικοποίησης θα έχει καλές ιδιότητες που θα επιτρέπουν την παραγωγή περιεχομένου. Στη συνέχεια, αναλύοντας την αρχιτεκτονική ενός VAE, βλέπουμε πως αποτελείται από τον κωδικοποιητή και τον αποκωδικοποιητή και έχει εκπαιδευτεί για να ελαχιστοποιεί το σφάλμα

ανακατασκευής μεταξύ των κωδικοποιημένων-αποκωδικοποιημένων δεδομένων και των αρχικών δεδομένων. Ουσιαστικά ως εδώ, η αρχιτεκτονική του VAE είναι ίδια με αυτή του αυτοκωδικοποιητή. Όμως για να εισάγουμε κάποια κανονικοποίηση στον χώρο κρυμμένης κωδικοποίησης, προχωράμε σε μία τροποποίηση στην διαδικασία κωδικοποίησης-αποκωδικοποίησης και τώρα αντί να κωδικοποιούμε μία είσοδο ως ένα απλό μόνο σημείο, την κωδικοποιούμε ως κατανομή στον χώρο κρυμμένης κωδικοποίησης. Αφού αναλύθηκε και η αρχιτεκτονική ενός VAE, τώρα μπορούμε να περάσουμε και στο πως εκπαιδεύεται.

Όποτε η εκπαίδευση ενός VAE γίνεται ως εξής:

1) Η είσοδος κωδικοποιείται ως κατανομή στον χώρο κρυμμένης κωδικοποίησης.
2) Γίνεται δειγματοληψία ενός σημείου του χώρου κρυμμένης κωδικοποίησης από αυτή την κατανομή.
3) Το σημείο που πήραμε από την δειγματοληψία αποκωδικοποιείται και υπολογίζεται το σφάλμα ανακατασκευής.
4) Το σφάλμα ανακατασκευής επαναπροσδιορίζεται μέσω του δικτύου. Ουσιαστικά γίνεται η διαδικασία backpropagation και το σφάλμα διαδίδεται προς τα πίσω στο δίκτυο.

Η παραπάνω διαδικασία, σχηματικά μπορεί να αναπαρασταθεί ως εξής:



Στην πράξη, στον VAE οι κωδικοποιημένες κατανομές επιλέγονται να είναι κανονικές έτσι ώστε ο κωδικοποιητής να μπορεί να εκπαιδευτεί για να επιστρέψει την μέση τιμή και τον πίνακα συνδιακύμανσης που περιγράφουν μία κανονική κατανομή. Ο λόγος για τον οποίο μία είσοδος κωδικοποιείται ως κάποια κατανομή με κάποια διακύμανση, αντί ως ένα μόνο σημείο, είναι ότι έτσι είναι δυνατό να εκφραστεί η κανονικοποίηση του χώρου κρυμμένης κωδικοποίησης: οι κατανομές που επιστρέφονται από τον κωδικοποιητή επιβάλλονται να είναι κοντά σε μία τυπική κανονική κατανομή.

Στη συνέχεια θα αναλυθεί η συνάρτηση λάθους που ελαχιστοποιείται κατά την εκπαίδευση ενός VAE. Αυτή η συνάρτηση αποτελείται από έναν όρο ανασυγκρότησης (στο τελικό επίπεδο), που τείνει να κάνει το σχήμα κωδικοποίησης-αποκωδικοποίησης όσο το δυνατόν πιο αποδοτικό. Αυτό τείνει να κανονικοποιήσει την οργάνωση του χώρου κρυμμένης κωδικοποίησης, κάνοντας τις κατανομές που επιστρέφονται από τον κωδικοποιητή να είναι κοντά σε μία τυπική κατανομή. Αυτός ο όρος κανονικοποίησης εκφράζεται ως η απόκλιση Kullback-Leibler (KL) μεταξύ της επιστρεφόμενης κατανομής και μια τυπικής κανονικής κατανομής (ή κατανομής Γκάους). Πιο συγκεκριμένα η συνάρτηση λάθους σε ένα VAE είναι της μορφής:

loss = $||x - \hat{x}||^2$ + KL[N(μ_x, σ_x),N(0,1)], όπου αυτός ο τύπος μπορεί να πάρει την μορφή:

loss = $||x - d(z)||^2$ + KL[N(μ_x, σ_x),N(0,1)], με βάση το σχηματικό ενός VAE που αναλύθηκε παραπάνω.

Από τον παραπάνω τύπο προκύπτει πως σε ένα VAE, η συνάρτηση απώλειας αποτελείται από έναν όρο ανακατασκευής (που καθιστά το σχήμα κωδικοποίησης-αποκωδικοποίησης αποτελεσματικό) και έναν όρο κανονικοποίησης (που καθιστά τον χώρο κρυμμένης κωδικοποίησης κανονικό).

Από εκεί και πέρα, η κανονικότητα που αναμένεται από τον χώρο κρυμμένης κωδικοποίησης προκειμένου να καταστεί δυνατή η παραγωγική διαδικασία μπορεί να εκφραστεί μέσω δύο κυρίων ιδιοτήτων: της συνέχειας, δηλαδή πως δύο κοντινά σημεία στον λανθάνοντα χώρο δεν πρέπει να δίνουν δύο εντελώς διαφορετικά περιεχόμενα μετά την αποκωδικοποίηση, και της πληρότητας, δηλαδή πως για μία επιλεγμένη κατανομή, ένα σημείο που λαμβάνεται ως δείγμα από τον λανθάνοντα χώρο θα πρέπει να δίνει ουσιαστικό περιεχόμενο μόλις αποκωδικοποιηθεί.

Εφόσον περιγράφτηκε και η συνάρτηση λάθους, μπορούμε να περάσουμε σε μία αναφορά που είχε γίνει παραπάνω, πως οι VAE κωδικοποιούν τις εισόδους ως κατανομές αντί για απλά ένα σημείο. Αυτό δεν αρκεί για να διασφαλιστεί η συνέχεια και η πληρότητα που αναφέρθηκαν στην προηγούμενη παράγραφο. Γενικά, χωρίς έναν καλά καθορισμένο όρο κανονικοποίησης, το μοντέλο μπορεί να μάθει, προκειμένου να ελαχιστοποιήσει το σφάλμα ανακατασκευής του και να αγνοήσει το γεγονός ότι οι κατανομές επιστρέφονται και συμπεριφέρονται σχεδόν σαν τους κλασικούς αυτόματους κωδικοποιητές (που οδηγούν σε αποστήθιση).

Έτσι, για να αποφύγουμε αυτή την συμπεριφορά, πρέπει να κανονικοποιήσουμε τόσο τον πίνακα συνδιακύμανσης όσο και την μέση τιμή των κατανομών που επιστρέφονται από τον κωδικοποιητή. Στην πράξη, αυτή η κανονικοποίηση πραγματοποιείται επιβάλλοντας τις κατανομές να πλησιάζουν μία τυπική κανονική κατανομή (κεντραρισμένη και μειωμένη). Με αυτό τον τρόπο, απαιτούμε οι πίνακες συνδιακύμανσης να είναι κοντά στον μοναδιαίο πίνακα, να αποτρέπουν τις τυπικές (ακριβείς) κατανομές και τον μέσο όρο να είναι κοντά στο 0, αποτρέποντας τις κωδικοποιημένες κατανομές να είναι πολύ μακριά μεταξύ τους.

Με αυτό τον όρο κανονικοποίησης, εμποδίζουμε το μοντέλο να κωδικοποιεί δεδομένα πολύ μακριά στον χώρο κρυμμένης κωδικοποίησης και να ενθαρρύνει όσο το δυνατόν περισσότερες επιστρεφόμενες κατανομές για “αλληλοεπικάλυψη”, ικανοποιώντας έτσι τις συνθήκες για την αναμενόμενη συνέχεια και πληρότητα. Φυσικά, όπως για κάθε όρο κανονικοποίησης, αυτό έρχεται μαζί με μία αύξηση στο σφάλμα ανακατασκευής στα δεδομένα εκπαίδευσης. Αλλά η αντιστάθμιση μεταξύ του σφάλματος ανακατασκευής και της απόκλισης KL μπορεί να προσαρμοστεί.

Στη συνέχεια ακολουθούν κάποιες εξηγήσεις σχετικά με τις μαθηματικές λεπτομέρειες των VAEs που θα βοηθήσουν και στην περαιτέρω κατανόηση της σχηματικής αναπαράστασης λειτουργίας ενός VAE. Αρχικά, δηλώνουμε με x την μεταβλητή που αντιπροσωπεύει τα δεδομένα μας και υποθέτουμε πως το x δημιουργείται από μία μεταβλητή z που ανήκει στον χώρο κρυμμένης κωδικοποίησης (η κωδικοποιημένη αναπαράσταση) που δεν παρατηρείται άμεσα. Έτσι, για κάθε σημείο δεδομένων, θεωρείται η ακόλουθη διαδικασία παραγωγής δύο βημάτων:

1) Λαμβάνεται δείγμα λανθάνουσας αναπαράστασης z από την προηγούμενη (prior) κατανομή p(z)

2) Γίνεται δειγματοληψία των δεδομένων x από την δεσμευμένη κατανομή πιθανότητας p(x|z)

Σε αυτό το σημείο, αξίζει να αναφερθεί πως ενώ ένας αυτόματος κωδικοποιητής υλοποιεί έναν ντετερμινιστικό κωδικοποιητή και αποκωδικοποιητή, τώρα έχουμε έναν πιθανοτικό κωδικοποιητή και αποκωδικοποιητή. Ο “πιθανοτικός” αποκωδικοποιητής ορίζεται φυσικά από το p(x|z), που περιγράφει την κατανομή της αποκωδικοποιημένης μεταβλητής δεδομένης της κωδικοποιημένης, ενώ ο “πιθανοτικός” κωδικοποιητής ορίζεται από το p(z|x), που περιγράφει την κατανομή της κωδικοποιημένης μεταβλητής δεδομένης της αποκωδικοποιημένης. Εδώ μπορούμε ήδη να παρατηρήσουμε πως η κανονικοποίηση του χώρου κρυμμένης κωδικοποίησης που μας λείπει σε απλούς αυτόματους κωδικοποιητές εμφανίζεται φυσικά εδώ στον ορισμό της διαδικασίας παραγωγής δεδομένων: οι κωδικοποιημένες αναπαραστάσεις z στον χώρο κρυμμένης κωδικοποίησης θεωρείται

πράγματι ότι ακολουθούν την προηγούμενη (prior) κατανομή p(z). Διαφορετικά, η σύνδεση μεταξύ της prior p(z), της likelihood p(x|z) και της posterior p(z|x) μπορεί να προκύψει από το γνωστό θεώρημα Bayes:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x|u)p(u)du}$$

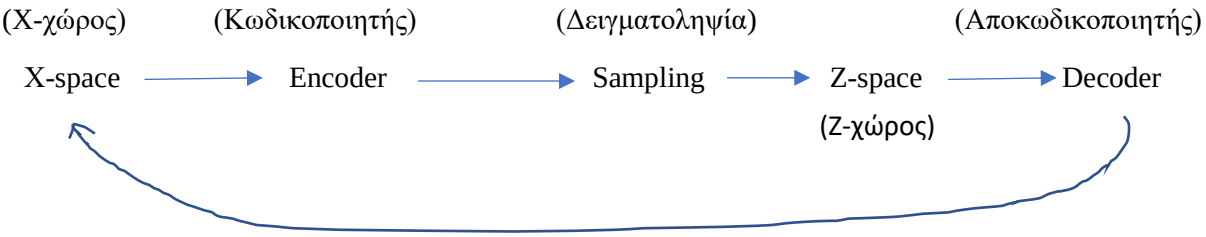
Γενικότερα, από όλα τα παραπάνω προκύπτει πως χρησιμοποιώντας έναν αυτοκωδικοποιητή, δεν γνωρίζουμε τίποτα για τα κωδικοποιημένα αντικείμενα που έχουν δημιουργηθεί από το δίκτυό μας. Θα μπορούσαμε να συγκρίνουμε διαφορετικά κωδικοποιημένα αντικείμενα, αλλά είναι απίθανο να καταλάβουμε τι συμβαίνει. Αυτό σημαίνει πως δεν θα μπορέσουμε να χρησιμοποιήσουμε τον αποκωδικοποιητή μας για τη δημιουργία νέων αντικειμένων. Απλώς δεν ξέρουμε πως θα πρέπει να εμφανίζονται οι είσοδοι.

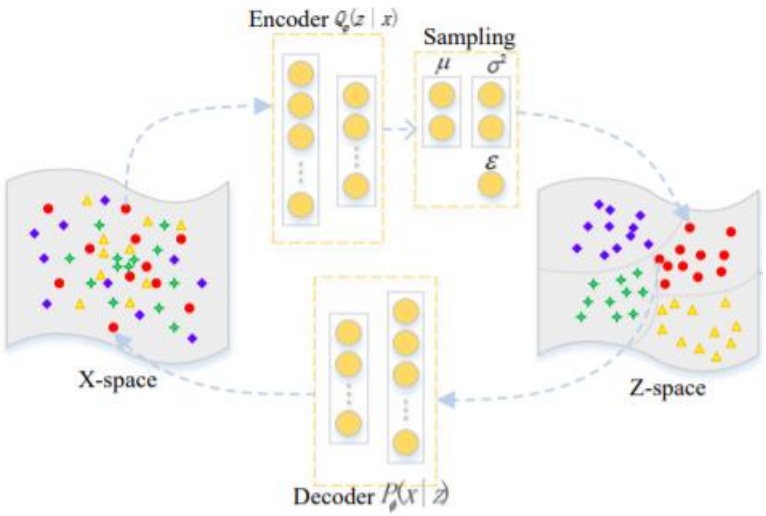
Αντίθετα χρησιμοποιώντας έναν VAE όπως αναλύθηκε παραπάνω, ακολουθούμε την αντίθετη προσέγγιση. Τώρα δεν προσπαθούμε να κάνουμε εικασίες σχετικά με την κατανομή που ακολουθείται από τα κρυμμένα διανύσματα. Απλώς λέμε στο δίκτυό μας πώς θέλουμε να είναι αυτή η κατανομή.

2.5.1 Αρχιτεκτονική του αυτοκωδικοποιητή μεταβολής

Εδώ θα αναλυθεί εκτενέστερα η εσωτερική δομή του αυτοκωδικοποιητή μεταβολής που χρησιμοποιούμε στο πλαίσιο αυτής της πτυχιακής.

Αρχικά αν θέλαμε να αναπαραστήσουμε σχηματικά την εσωτερική δομή του αυτοκωδικοποιητή μεταβολής, το σχήμα θα μπορούσε να είναι κάπως έτσι (παρουσιάζονται 2 εκδοχές):





Σχήμα 2.4: Σχηματική αναπαράσταση του αυτοκωδικοποιητή μεταβολής.

Ο X-χώρος αναπαριστά την είσοδο. Στη συνέχεια ακολουθεί ο κωδικοποιητής, στον οποίο περνάμε την είσοδο μέσα από κάποια επίπεδα με σκοπό την κωδικοποίηση της εισόδου. Στη συνέχεια μόλις παραχθούν οι παράμετροι z_mean και z_log_var που αντιστοιχούν στην μέση τιμή και την συνδιακύμανση της παραμέτρου z, περνάμε αυτές τις δύο παραμέτρους μέσα στην διαδικασία της δειγματοληψίας, όπου μέσω αυτής της διαδικασίας παράγεται ο Z-χώρος. Στη συνέχεια δίνουμε τον Z-χώρο ως είσοδο στον αποκωδικοποιητή του μοντέλου μας. Και ουσιαστικά μετά στόχος είναι ο αποκωδικοποιητής να μας επιστρέψει ως έξοδο την αρχική είσοδο. Επίσης στόχος είναι όταν εκπαιδευτεί το μοντέλο μας, να μας επιστρέφει λογικές εξόδους για άγνωστες εισόδους (άγνωστες ως προς την εκπαίδευση του μοντέλου μας πάντα). Δηλαδή δίνοντας άγνωστες εισόδους στον αποκωδικοποιητή μας, να μπορεί να αναπαράγει λογικές εξόδους που να βγάζουν νόημα και να είναι σωστές γενικά.

Στη συνέχεια θα προχωρήσουμε ακόμα πιο εσωτερικά στο μοντέλο. Μπορούμε να διακρίνουμε πως ο κωδικοποιητής και ο αποκωδικοποιητής χρησιμοποιούν ακριβώς τον ίδιο αριθμό επιπέδων. Επίσης η διαδικασία της δειγματοληψίας χρησιμοποιεί το λεγόμενο τρικ επαναπροσδιορισμού, το οποίο καθιστά δυνατή την χρήση του gradient descent παρά την τυχαία δειγματοληψία που συμβαίνει στα μισά της αρχιτεκτονικής και βασίζεται στο εξής: ότι εάν το z είναι μία τυχαία μεταβλητή που ακολουθεί την κανονική κατανομή με μέση τιμή μ και συνδιακύμανση σ^2 , τότε το z μπορεί να γραφτεί στη μορφή:

z = μ + σ^2 * ε, ε ~ N(0,1), όπου το ε ακολουθεί την κανονική κατανομή με μέση τιμή 0 και συνδιακύμανση 1.

2.5.2 Μεταβολική συμπερασματολογία

Τώρα θα ακολουθήσει μία ανάλυση για την μέθοδο της μεταβολικής συμπερασματολογίας (που από εκεί ουσιαστικά προκύπτει και ο τύπος του λάθος του VAE μας, που αναλύθηκε παραπάνω).

Πιο συγκεκριμένα, στην στατιστική, η μεταβολική συμπερασματολογία (variational inference-VI) είναι μία τεχνική για την προσέγγιση πολύπλοκων κατανομών. Η ιδέα είναι να οριστεί μία παραμετροποιημένη οικογένεια κατανομών (για παράδειγμα η οικογένεια κανονικών κατανομών με παραμέτρους την μέση τιμή και την συνδιακύμανση) και να αναζητηθεί η καλύτερη προσέγγιση της κατανομής που μας ενδιαφέρει μέσα σε αυτή την οικογένεια. Το καλύτερο στοιχείο μέσα στην οικογένεια είναι εκείνο που ελαχιστοποιεί μία δεδομένη μέτρηση σφάλματος προσέγγισης (τις περισσότερες φορές είναι η απόκλιση Kullback-Leibler μεταξύ προσέγγισης και στόχου) και εντοπίζεται από τον gradient descent ανάμεσα στις παραμέτρους που περιγράφουν την οικογένεια.

Εδώ προσεγγίζουμε το p(z|x), που περιγράφτηκε παραπάνω, με μία κανονική κατανομή q_x(z), της οποίας η μέση τιμή και η συνδιακύμανση ορίζονται από δύο συναρτήσεις g και h, που ανήκουν στις οικογένειες των συναρτήσεων G και H.

Έτσι μπορούμε να θεωρήσουμε πως

q_x(z) ≡ N(g(x),h(x)), όπου το N(μ, σ^2) συμβολίζει την κανονική κατανομή με μέση τιμή μ και συνδιακύμανση σ^2 και επίσης έχουμε πως το g ∈ G και h ∈ H (τα H και G αναλύονται παρακάτω). Έτσι καταλήγουμε πως το q_x(z) ακολουθεί την κανονική κατανομή με μέση τιμή g(x) και συνδιακύμανση h(x).

Οπότε τώρα έχουμε ορίσει μία οικογένεια υποψηφίων συναρτήσεων για τη μεταβολική συμπερασματολογία και πρέπει να βρούμε την καλύτερη προσέγγιση μεταξύ αυτής της οικογένειας, βελτιστοποιώντας τις συναρτήσεις g και h (στην πραγματικότητα τις παραμέτρους τους) για να ελαχιστοποιηθεί η απόκλιση Kullback-Leibler μεταξύ της προσέγγισης και του στόχου p(z|x). Με άλλα λόγια ψάχνουμε τα βέλτιστα g* και h* ώστε:

$$(g^*,h^*) = \arg\min_{(g,h)\in G\times H} KL(q_x(z),p(z|x)) \quad (2.1)$$

$$= \arg\min_{(g,h)\in G\times H} \left(E_{z\sim q_x}(\log q_x(z)) - E_{z\sim q_x}\left(\log \frac{p(x|z)p(z)}{p(x)}\right)\right) \quad (2.2)$$

$$= \arg\min_{(g,h)\in G\times H} (E_{z\sim q_x}(\log q_x(z)) - E_{z\sim q_x}(\log p(z)) - E_{z\sim q_x}\log p(x|z) + E_{z\sim q_x}(\log p(x))) \quad (2.3)$$

$$= \arg\max_{(g,h)\in G\times H} (E_{z\sim q_x}(log\ p(x|z)) - KL(q_x(z),p(z))) \quad (2.4)$$

$$= \arg\max_{(g,h)\in G\times H} \left(E_{z\sim q_x}\left(-\frac{||x-f(z)||^2}{2c}\right) - \mathrm{KL}(q_x(z), p(z))\right) \quad (2.5)$$

Στην εξίσωση 2.4, μπορούμε να παρατηρήσουμε την αντιστάθμιση που υπάρχει κατά την προσέγγιση του p(z|x) μεταξύ μεγιστοποίησης της πιθανότητας των "παρατηρήσεων" (μεγιστοποίηση της αναμενόμενης log-likelihood πιθανότητας, για τον πρώτο όρο) και παραμονής κοντά στην προηγούμενη κατανομή (ελαχιστοποίηση της απόκλισης KL μεταξύ $q_x(z)$ και p(z), για τον δεύτερο όρο). Αυτή η αντιστάθμιση είναι φυσική για το πρόβλημα της Μπεϋζιανής συμπερασματολογίας και εκφράζει την ισορροπία που πρέπει να βρεθεί μεταξύ της εμπιστοσύνης που έχουμε στα δεδομένα και της εμπιστοσύνης που έχουμε στο παρελθόν.

Επιπρόσθετα, με βάση και τον παραπάνω τύπο που προέκυψε, αν η κανονικότητα εξαρτάται κυρίως από την προηγούμενη κατανομή που λαμβάνεται πάνω στο χώρο κρυμμένης κωδικοποίησης, η απόδοση του συνολικού σχήματος κωδικοποίησης-αποκωδικοποίησης εξαρτάται σε μεγάλο βαθμό από την επιλογή της συνάρτησης f (που φαίνεται στην τελευταίο τύπο που προέκυψε στην παραπάνω σχέση). Πράγματι, αφού το p(z|x) μπορεί να προσεγγιστεί (λόγω της μεταβολικής συμπερασματολογίας) από το p(z) και το p(x|z) και καθώς το p(z) είναι μία απλή τυπική κατανομή, οι μόνοι δύο μοχλοί που έχουμε στη διάθεσή μας στο μοντέλο για να κάνουμε βελτιστοποιήσεις είναι η παράμετρος c (που καθορίζει την διακύμανση της πιθανότητας) και η συνάρτηση f (που καθορίζει την μέση τιμή της πιθανότητας).

Ας θεωρήσουμε λοιπόν πως μπορούμε να πάρουμε για οποιαδήποτε συνάρτηση f στο F (καθένας ορίζει έναν διαφορετικό πιθανοτικό αποκωδικοποιητή p(x|z)) την καλύτερη προσέγγιση του p(z|x), που υποδηλώνεται ως $q_x^*(z)$. Παρά την πιθανοτική φύση του, αναζητούμε ένα σχήμα κωδικοποίησης-αποκωδικοποίησης όσο το δυνατόν πιο αποτελεσματικό και στη συνέχεια θέλουμε να επιλέξουμε την συνάρτηση f που μεγιστοποιεί την αναμενόμενη log-likelihood πιθανότητα του x δεδομένου του z όταν το z λαμβάνεται με δειγματοληψία από το $q_x^*(z)$. Με άλλα λόγια, για μία δεδομένη είσοδο x, θέλουμε να μεγιστοποιήσουμε την πιθανότητα να έχουμε $\hat{x} = x$ όταν κάνουμε δειγματοληψία του z από την κατανομή $q_x^*(z)$ και τότε κάνουμε δειγματοληψία του $\hat{x}$ από την κατανομή p(x|z). Επιπλέον, ψάχνουμε το βέλτιστο f* έτσι ώστε:

$$f^* = \arg\max_{f\in F} E_{z\sim q_x^*}(\log p(x|z)) = \arg\max_{f\in F} E_{z\sim q_x^*}\left(-\frac{||x-f(z)||^2}{2c}\right)$$

όπου το q_x^* εξαρτάται από την συνάρτηση f και λαμβάνεται όπως περιγράφηκε σε προηγούμενη παράγραφο.

Συνδυάζοντας τους παραπάνω τύπους, καταλήγουμε στην αναζήτηση των βέλτιστων f*, g* και h*, έτσι ώστε:

$$(f^*, g^*, h^*) = \arg\max_{(f,g,h)\in F\times G\times H} \left(E_{z\sim q_x}\left(-\frac{||x-f(z)||^2}{2c}\right) - \mathrm{KL}(q_x(z), p(z))\right)$$

Στην παραπάνω συνάρτηση μπορούμε να διακρίνουμε το σφάλμα ανακατασκευής μεταξύ του x και του f(z) και τον όρο κανονικοποίησης που δίνεται από την απόκλιση KL μεταξύ q_x(z) και p(z) (που είναι μια τυπική κατανομή). Μπορούμε επίσης να διακρίνουμε τη σταθερά c που ελέγχει την ισορροπία μεταξύ των δύο προηγούμενων όρων που περιγράφηκαν. Όσο υψηλότερο είναι το c τόσο περισσότερο υποθέτουμε μία μεγάλη διακύμανση γύρω από το f(z) για τον πιθανό αποκωδικοποιητή στο μοντέλο μας και, έτσι, τόσο περισσότερο προτιμούμε τον όρο κανονικοποίησης έναντι του όρου ανασυγκρότησης (και το αντίθετο ισχύει αν το c είναι μικρό).

Αφού αναλύθηκε και η συνάρτηση που πρέπει να βελτιστοποιήσουμε, τώρα μπορούμε να περάσουμε στον ορισμό των συναρτήσεων f, g και h που δεν έχουν οριστεί παραπάνω. Αρχικά, επειδή δεν μπορούμε να βελτιστοποιήσουμε εύκολα ολόκληρο τον χώρο των συναρτήσεων, περιορίζουμε τον τομέα βελτιστοποίησης και εκφράζουμε τις συναρτήσεις f, g και h ως νευρωνικά δίκτυα. Έτσι, τα F, G και H αντιστοιχούν στις οικογένειες των συναρτήσεων που ορίζονται από τις αρχιτεκτονικές των δικτύων και η βελτιστοποίηση γίνεται με βάση τις παραμέτρους αυτών των δικτύων.

Στην πράξη, τα g και h δεν ορίζονται από δύο εντελώς ανεξάρτητα δίκτυα, αλλά μοιράζονται ένα μέρος της αρχιτεκτονικής τους και των βαρών τους, έτσι ώστε να έχουμε:

g(x) = $g_2(g_1(x))$ h(x) = $h_2(h_1(x))$ $g_1(x) = h_1(x)$

Καθώς ορίζεται ο πίνακας συνδιακύμανσης του q_x(z), το h(x) υποτίθεται πως είναι ένας τετραγωνικός πίνακας. Ωστόσο, προκειμένου να απλοποιήσουμε τον υπολογισμό και να μειώσουμε τον αριθμό των παραμέτρων, κάνουμε την πρόσθετη υπόθεση ότι η προσέγγιση μας για το p(z|x), q_x(z), είναι μία πολυδιάστατη κανονική κατανομή με διαγώνιο πίνακα συνδιακύμανσης (υπόθεση ανεξαρτησίας μεταβλητών). Με αυτή την υπόθεση, το h(x) είναι απλώς το διάνυσμα των διαγώνιων στοιχείων του πίνακα συνδιακύμανσης και τότε έχει το ίδιο μέγεθος με το g(x). Ωστόσο, μειώνοντας με αυτό τον τρόπο την οικογένεια των κατανομών που παίρνουμε υπόψιν μας για την μεταβολική συμπερασματολογία, τότε η προσέγγιση του p(z|x) μπορεί να είναι λιγότερο ακριβής.

Σε αντίθεση με το τμήμα κωδικοποιητή που μοντελοποιεί το p(z|x) και για το οποίο θεωρήσαμε μία κανονική κατανομή με μέση τιμή και συνδιακύμανση που είναι συναρτήσεις του x (g και h), το μοντέλο μας υποθέτει για το p(z|x) μία κανονική κατανομή με σταθερή συνδιακύμανση. Η συνάρτηση f της μεταβλητής z που ορίζει τη μέση τιμή της κανονικής κατανομής μοντελοποιείται από ένα νευρωνικό δίκτυο.

Κεφάλαιο 3.

Αραιές αναπαραστάσεις και δίκτυα deep unfolding

Σε αυτό το κεφάλαιο αρχικά αναλύεται ο όρος της αραιής κωδικοποίησης και η συσχέτιση του με την επεξεργασία εικόνας. Η αραιή κωδικοποίηση μπορεί να αποτελέσει μία αποτελεσματική λύση αντιμετώπισης του προβλήματος των δεδομένων υψηλής διάστασης (όπως μία εικόνα). Επίσης, σε αυτή τη διπλωματική εργασία, η μέθοδος που υλοποιούμαι βασίζεται στην αραιή αναπαράσταση μίας μεταβλητής. Επιπρόσθετα, σε αυτό το κεφάλαιο αναλύεται και η εκμάθηση λεξικών, η οποία αποτελεί βασική προϋπόθεση ώστε μία αραιή αναπαράσταση μίας μεταβλητής να είναι καλή. Όμως και η εκμάθηση λεξικού και ο υπολογισμός μίας αραιής αναπαράστασης χαρακτηρίζονται από μεγάλο υπολογιστικό κόστος. Έτσι προκύπτει η ανάγκη εύρεσης ενός αλγορίθμου (ή μίας μεθόδου) ο οποίος θα μειώσει το υπολογιστικό κόστος. Αυτός ο αλγόριθμος είναι το δίκτυο LISTA, το όποιο χρησιμοποιούμε στην μέθοδο μας για την εύρεση της αραιής αναπαράστασης μίας μεταβλητής.

3.1 Αραιή κωδικοποίηση για αντίστροφα γραμμικά προβλήματα στην επεξεργασία εικόνας

Τα αντίστροφα γραμμικά προβλήματα στην επεξεργασία εικόνας συνήθως διατυπώνονται ως εξής:

$y = Lx + \eta$, (3.1)

όπου το $x \in R^k$ είναι μία διανυσματική μορφή της άγνωστης πηγαίας εικόνας, το $y \in R^n$ υποδηλώνει τις υποβαθμισμένες παρατηρήσεις (θολές) και το $\eta \in R^n$ είναι ο θόρυβος. Το πρόβλημα (3.1) εμφανίζεται σε πολλές εφαρμογές επεξεργασίας εικόνας συμπεριλαμβανομένης της ανάκτησης εικόνας και της αποθορυβοποίησης.

3.1.1 Αραιή Κωδικοποίηση

Ακόμη και όταν δίνεται ο τελεστής γραμμικής παρατήρησης L, το πρόβλημα 3.1 δεν είναι καλώς ορισμένο και απαιτείται κανονικοποίηση για την λύση του. Η αραιότητα έχει χρησιμοποιηθεί ευρέως ως κανονικοποιητής που οδηγεί στο γνωστό πρόβλημα αραιής προσέγγισης. Αντί να λύσουμε άμεσα ως προς x, βασιζόμαστε σε μία αραιή μοντελοποίηση. Υποθέτουμε οτί το x μπορεί να γραφεί ως $x = \Psi u$, $\Psi \in R^{k \times m}$, όπου το $u \in R^m$ είναι ένα αραιό διάνυσμα που μπορεί να υπολογιστεί με την επίλυση του προβλήματος:

$$\min_u \frac{1}{2}||x - \Psi u||_2^2 + \lambda||u||_1, \text{ (3.2)}$$

όπου το λ είναι η παράμετρος κανονικοποίησης, και το $||u||_1 = \sum_{i=1}^{N} |u_i|$ είναι η l_1 νόρμα, που προάγει την αραιότητα. Στη συνέχεια, το να βρούμε το x από το παρατηρούμενο διάνυσμα y, ανάγεται στο πρόβλημα:

$$\min_u \frac{1}{2}||y - L\Psi u||_2^2 + \lambda||u||_1. \text{ (3.3)}$$

Θέτοντας $D = L\Psi$, $D \in R^{n \times m}$, η σχέση (3.3) μπορεί να ξαναγραφτεί ως εξής:

$$\min_u \frac{1}{2}||y - Du||_2^2 + \lambda||u||_1. \text{ (3.4)}$$

Το πρόβλημα (3.4) επιλύεται με αριθμητικές μεθόδους οι οποίες έχουν συνήθως μεγάλο υπολογιστικό κόστος.

3.1.2 Εκμάθηση Λεξικών

Η θεωρία εκμάθησης λεξικών, ασχολείται με το πρόβλημα της κατασκευής του λεξικού και της αναπαράστασης των δεδομένων μέσω του λεξικού. Δοθέντος ενός λεξικού D = $[d_1, d_2, \ldots, d_k] \in R^{n\times K}$, κάθε $d_k \in R^n$ ονομάζεται άτομο. Στη συνέχεια, το σύνολο δεδομένων εισόδου X = $[x_1, x_2, \ldots, x_k] \in R^{d\times N}$ μπορεί να αναπαρασταθεί από έναν γραμμικό συνδυασμό πολλών ατόμων, ο πίνακας A = $[a_1, a_2, \ldots, a_N] \in R^{K\times N}$ περιλαμβάνει τους συντελεστές αναπαράστασης που αντιστοιχούν στο X με βάση το λεξικό D. Εάν το A έχει μόνο s μη μηδενικούς συντελεστές, ονομάζεται s-sparse(s-αραιός). Το κύριο μέλημα της εκμάθησης λεξικού είναι να λύσει το ακόλουθο πρόβλημα βελτιστοποίησης για να πάρουμε τη βέλτιστη βάση διανυσμάτων D και τον βέλτιστο αραιό συντελεστή A:

$\min_{D,A} ||X - DA||_F^2 + \lambda R(A)$,

όπου, το λ είναι μία θετική σταθερά κλίμακας και το R(A) είναι ο όρος κανονικοποίησης που μετρά το πόσο αραιός είναι ο πίνακας συντελεστών A.

Το πρόβλημα επιλύεται επαναληπτικά ως εξής. Κάθε επανάληψη αποτελείται από δύο βήματα. Στο πρώτο βήμα, το λεξικό D παραμένει σταθερό και γίνεται βελτιστοποίηση των συντελεστών A. Στο δεύτερο βήμα, ο πίνακας συντελεστών A παραμένει σταθερός και γίνεται βελτιστοποίηση του λεξικού D. Πραγματοποιούνται τόσες επαναλήψεις ώστε το σφάλμα ανακατασκευής να είναι αρκετά μικρό.

3.2 Αραιές αναπαραστάσεις με την μέθοδο deep unfolding

Οι αναλυτικές προσεγγίσεις για τον υπολογισμό αραιών αναπαραστάσεων είναι συνήθως εξοπλισμένες με θεωρητικές εγγυήσεις, ωστόσο, το κύριο μειονέκτημα τους είναι η υψηλή υπολογιστική πολυπλοκότητα τους. Σε ορισμένες εφαρμογές, τα λεξικά που έχουν αναπτυχθεί επίσης πρέπει να μαθευτούν, αυξάνοντας τον υπολογιστικό φόρτο. Για την αντιμετώπιση της υψηλής πολυπλοκότητας της αριθμητικής βελτιστοποίησης, προτάθηκε ένα νευρωνικό δίκτυο που εκτελεί παρόμοιες ενέργειες με έναν επαναληπτικό proximal αλγόριθμο για την προσέγγιση της αραιότητας, που ονομάζεται επαναληπτικός αλγόριθμος soft thresholding (ISTA). Για την λύση της εξίσωσης (3.4), η t-οστή επανάληψη του ISTA υπολογίζει το εξής:

$u^t = \varphi_\gamma(u^{t-1} - \frac{1}{L}D^T(Du^t - y))$, $u^0 = 0$, (3.5)

όπου το L είναι μία κατάλληλη σταθερά και το φ_γ είναι ο τελεστής soft thresholding που ορίζεται ως εξής:

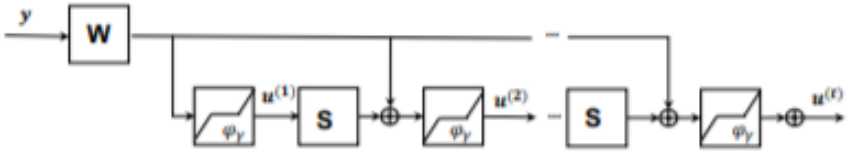
$\varphi_\gamma(u_i) = sign(u_i)\max\{0, |u_i| - \gamma\}$, i = {1, …, k}. (3.6)

Μία μορφή του ISTA μπορεί να ληφθεί γράφοντας κάθε επανάληψη στην ακόλουθη μορφή:

$u^t = \varphi_\gamma(Su^{t-1} + Wy)$, $u^0 = 0$, (3.7)

όπου S = $I - \frac{1}{L}D^T D$, W = $\frac{1}{L}D^T$. Η σχέση (3.7) εκφράζει το επίπεδο t του νέου αλγόριθμου που ονομάζεται LISTA (Learned ISTA). Το δίκτυο μπορεί να εκπαιδευτεί με εποπτευόμενο τρόπο ώστε να αντιστοιχεί ένα σήμα εισόδου y σε αραιές αναπαραστάσεις u. Οι παράμετροι S ∈ $R^{m\times m}$, W ∈ $R^{m\times n}$

και γ > 0 του LISTA μπορούν να μαθευτούν από τα δεδομένα (ενώ στον ISTA είναι σταθερές). Ως αποτέλεσμα, ο LISTA πετυχαίνει μεγάλη ακρίβεια σε μόλις λίγες επαναλήψεις. Μία γραφική αναπαράσταση του μοντέλου φαίνεται στο σχήμα 3.1. Η μετατροπή ενός επαναληπτικού αλγορίθμου σε νευρωνικό δίκτυο είναι μια νέα τεχνική που είναι γνωστή με τον όρο deep unfolding.

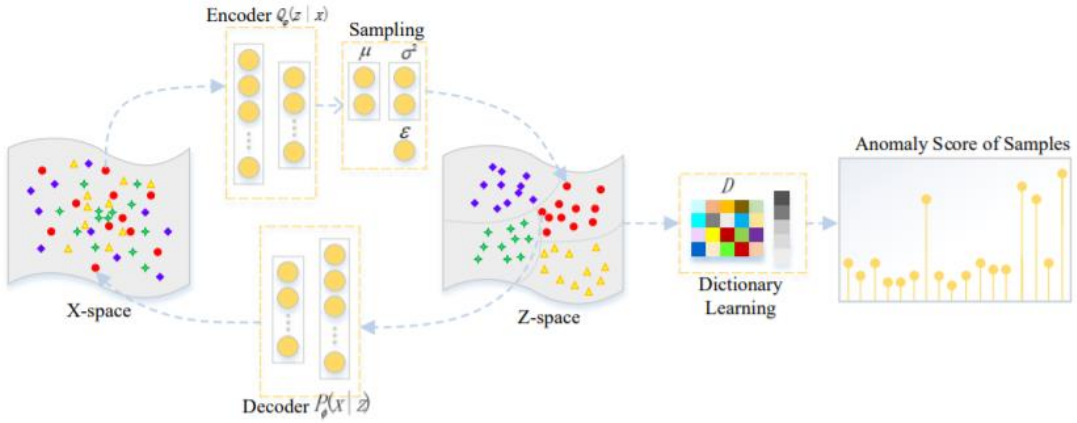


Σχήμα 3.1: Αραιή κωδικοποίηση με το μοντέλο LISTA. Το μοντέλο υπολογίζει αραιές αναπαραστάσεις u^t από μία διανυσματική αναπαράσταση y μιας εικόνας.

Κεφάλαιο 4.

Εντοπισμός ανωμαλιών σε βίντεο με χρήση του αυτοκωδικοποιητή μεταβολής

4.1 Υλοποίηση του αυτοκωδικοποιητή μεταβολής



Σχήμα 4.1: Σχηματική αναπαράσταση της μεθόδου που υλοποιούμε.

Σε αυτό το κεφάλαιο θα παρουσιάσουμε μια μέθοδο εντοπισμού ανωμαλιών σε βίντεο με χρήση του αυτοκωδικοποιητή μεταβολής. Η μέθοδος βασίζεται στην αραιή αναπαράσταση της κρυμμένης μεταβλητής z. Σύμφωνα με όσα έχουν αναφερθεί στο κεφάλαιο 3, ο υπολογισμός της αραιής αναπαράστασης του z μπορεί να επιτευχθεί με δύο τρόπους: α) Με εκμάθηση κατάλληλου λεξικού και χρήση ενός επαναληπτικού αλγορίθμου π.χ. ADMM β) Με την ενσωμάτωση του δικτύου LISTA στο μοντέλου του αυτοκωδικοποιητή μεταβολής. Στην παρούσα διπλωματική εργασία θα χρησιμοποιήσουμε τη δεύτερη μέθοδο. Συγκεκριμένα, θα χρησιμοποιήσουμε ένα αυτοκωδικοποιητή LISTA ο οποίος θα δίνει στην έξοδό του την ανακατασκευή του z. Για τον εντοπισμό ενός μη ομαλού βίντεο, ελέγχουμε το σφάλμα αραιής ανακατασκευής της κρυμμένης μεταβλητής z. Αν το σφάλμα είναι μεγάλο τότε θεωρούμε ότι έχουμε μη ομαλή σκηνή βίντεο γιατί η αραιή αναπαράσταση της κρυμμένης μεταβλητής z που μας δίνει ο αλγόριθμος LISTA δεν επιτυγχάνει καλή ανακατασκευή του z, συνεπώς οδηγεί σε μεγάλο σφάλμα και έτσι στην συνέχεια υπάρχει πολύ μεγάλη πιθανότητα ο αποκωδικοποιητής να μην μας δώσει πίσω την αρχική εικόνα (όπως θα ήταν επιθυμητό) και να έχουμε μεγάλο σφάλμα. Αυτό επίσης μας δείχνει πως η εικόνα που έχουμε, είναι πολύ πιθανό (ή και σχεδόν σίγουρο) να έχει αλλοιωθεί σημαντικά λόγω θορύβου ή οποιοδήποτε άλλου φαινομένου και έτσι να μπορέσουμε να δούμε σε πρώιμο στάδιο πως η εικόνα που έχουμε στο δίκτυο μας είναι αλλοιωμένη.

Οι λόγοι που μας οδηγούν στην χρησιμοποίηση του LISTA αντί μιας συμβατικής μεθόδου υπολογισμού αραιών αναπαραστάσεων είναι οι ακόλουθοι

Η χρήση αναπαραστάσεων των δεδομένων με χρήση νευρωνικών δικτύων αποτελεί σήμερα μια δημοφιλή λύση γιατί επιτυγχάνει την κωδικοποίηση της σημαντικής πληροφορίας που εμπεριέχεται στα δεδομένα. Παρόλα αυτά το πρόβλημα της υψηλής διάστασης μπορεί να υφίσταται και στο χώρο αναπαράστασης. Η αραιή κωδικοποίηση αποτελεί μια αποτελεσματική λύση αντιμετώπισης του προβλήματος των δεδομένων υψηλής διάστασης όπως είναι το βίντεο. Ωστόσο, μια επιτυχημένη αραιή αναπαράσταση προϋποθέτει την εκμάθηση ενός κατάλληλου λεξικού. Όμως, τόσο η εκμάθηση του

λεξικού όσο και ο ίδιος ο υπολογισμός μιας αραιής αναπαράστασης είναι διαδικασίες μεγάλου υπολογιστικού κόστους. Επομένως, προτείνουμε ένα σχέδιο για την κατασκευή του λεξικού D μέσα στον χώρο κρυμμένης κωδικοποίησης του αυτοκωδικοποιητή μεταβολής, που όχι μόνο υπερνικά το ζήτημα των αναπαραστάσεων υψηλής διάστασης, αλλά επίσης ανακαλύπτει με επιτυχία την κατανομή και κάποια δομικά χαρακτηριστικά των δεδομένων.

Ο αλγόριθμος LISTA χρησιμοποιείται ουσιαστικά στην φάση της εκμάθησης λεξικού (όπως φαίνεται στο σχήμα 4.1), παίρνοντας ως είσοδο την κρυμμένη μεταβλητή z που παράγεται στην διαδικασία της δειγματοληψίας. Εδώ να τονιστεί πως ο LISTA χρησιμοποιείται πριν περάσουμε την μεταβλητή z στον αποκωδικοποιητή μας. Επίσης, ο LISTA είναι φτιαγμένος έτσι ώστε με λίγες επαναλήψεις να μπορεί να δώσει τις αραιές αναπαραστάσεις της κρυμμένης μεταβλητής z.

Τώρα μπορούμε να περάσουμε και στην ανάλυση του σχήματος 4.1. Αυτό το σχήμα αποτελεί μία απεικόνιση της μεθόδου που υλοποιούμε. Το αριστερό μέρος του, που αποτελείται από τον X-χώρο, τον κωδικοποιητή, την διαδικασία της δειγματοληψίας, τον Z-χώρο και τον αποκωδικοποιητή, αναλύθηκε στο Σχ. 2.4 και ουσιαστικά είναι ο αυτοκωδικοποιητής μεταβολής. Αυτό που δεν αναλύθηκε στο Σχ. 2.4, είναι πως το z που παράγεται από τον κωδικοποιητή, αποτελεί είσοδο του αποκωδικοποιητή VAE αλλά και ενός αυτοκωδικοποιητή LISTA. Στη φάση εκπαίδευσης, ο αυτοκωδικοποιητής LISTA μαθαίνει να ανακατασκευάζει την κρυμμένη μεταβλητή z. Στη φάση δοκιμών, το σφάλμα ανακατασκευής χρησιμοποιείται ως κριτήριο εμφάνισης ανωμαλιών.

Ακολουθεί μια λεπτομερής περιγραφή του προτεινόμενου μοντέλου.

Με δεδομένο ένα σύνολο δεδομένων εκπαίδευσης, X = $[x_1, x_2, \ldots, x_k] \in R^{d \times N}$, αρχικά το δίκτυο μας υλοποιεί έναν κωδικοποιητή. Ο κωδικοποιητής αποτελείται από M επίπεδα, που το καθένα περιέχει r_{end}^m νευρώνες $\forall m = 1, 2, \ldots, M$. Υποθέτοντας πως το $W_{end}^m \in R^{r^m \times r^{m-1}}$ και το $b_{end}^m \in R^{r^m}$ είναι ο πίνακας βαρών και το διάνυσμα πόλωσης αντίστοιχα του m-ιοστού επιπέδου, η είσοδος του m επιπέδου, η οποία είναι επίσης και η έξοδος του (m-1)-ιοστού επιπέδου, μπορεί να αναπαρασταθεί ως εξής:

$h^m = W_{end}^m \times h^{m-1} + b_{end}^m$,

όπου το h^0 είναι η είσοδος x. Στη συνέχεια, σύμφωνα με την μέση τιμή $\mu \in R^{r^M}$ και την συνδιακύμανση $\sigma^2 \in R^{r^M}$ του τελευταίου επιπέδου εξόδου του κωδικοποιητή, η τυχαία μεταβλητή $z \in R^{r^M \times N}$ υπόκειται στην αντίστοιχη κανονική κατανομή που δημιουργείται ως κρυμμένος χώρος κωδικοποίησης. Αυτή η διαδικασία ονομάζεται "τρικ επαναπροσδιορισμού", δηλαδή $z^i = \mu + \sigma^2 \varepsilon$, με το ε να ακολουθεί την κανονική κατανομή με μέση τιμή 0 και συνδιακύμανση 1 (N(0,1)).

Στη συνέχεια, η μεταβλητή z του κρυμμένου χώρου κωδικοποίησης, θα διαδοθεί στα δύο επόμενα μέρη του μοντέλου, στον αποκωδικοποιητή και στον αραιό αυτοκωδικοποιητή LISTA. Ο αποκωδικοποιητής είναι ένα πολυεπίπεδο νευρωνικό δίκτυο με τον ίδιο αριθμό των επιπέδων του κωδικοποιητή, και κάθε επίπεδο έχει $r_{dec}^{M-l} = r_{end}^{l+1}$ νευρώνες για όλα τα l = 0, 1, 2, …., M-1. Προφανώς, ο αποκωδικοποιητής υπολογίζεται με τον ίδιο τρόπο όπως και ο κωδικοποιητής, εκτός από την είσοδο που είναι διαφορετική. Με άλλα λόγια, απεικονίζουμε την εκπαίδευση των δεδομένων X στην μεταβλητή του κρυμμένου χώρου κωδικοποίησης z από τον κωδικοποιητή f(X), και στη συνέχεια τα νέα δεδομένα $\hat{X}$ παράγονται από τον αποκωδικοποιητή g(Z), ο οποίος παραμετροποιείται από το σύνολο $\{W_{dec,}^m b_{dec}^m\}_{m=1}^M$. Κατά την αποκωδικοποίηση της μεταβλητής z του κρυμμένου χώρου κωδικοποίησης, χρησιμοποιούμε και τον αυτοκωδικοποιητή LISTA, ο οποίος λαμβάνει ως είσοδο το z, υπολογίζει μια αραιή αναπαράσταση α και στη συνέχεια ανακατασκευάζει το $\hat{z}$ = Dα. Το D είναι παράμετρος του αυτοκωδικοποιητή LISTA. Επομένως, εκτός από την διασφάλιση του ελαχίστου σφάλματος κατανομής μεταξύ της μεταβλητής z

του κρυμμένου χώρου κωδικοποίησης και της μεταβλητής παρατήρησης X και το σφάλμα ανάμεσα στην αρχική μεταβλητή X και στα παραγόμενα δεδομένα $\hat{X}$ είναι το ελάχιστο, επίσης το μοντέλο χρειάζεται να διασφαλίσει ότι το σφάλμα ανασυγκρότησης του αυτοκωδικοποιητή LISTA θα είναι το ελάχιστο. Λαμβάνοντας υπόψη όλα τα παραπάνω, το μοντέλο μας μπορεί να διατυπωθεί ως το ακόλουθο πρόβλημα βελτιστοποίησης:

Το D = $[d_1, d_2, \ldots, d_k] \in R^{r^m \times K}$ αντιπροσωπεύει ένα λεξικό που εκπαιδεύτηκε μέσα στο z-διανυσματικό χώρο, όπου κάθε a_i μέσα στο A = $[a_1, a_2, \ldots, a_N] \in R^{K \times N}$ είναι ένα αραιό διάνυσμα συντελεστών που αντιστοιχεί σε κάθε διάνυσμα στο z, διασφαλίζοντας πως οι περισσότερες σειρές του A είναι μηδενικά διανύσματα για να εξασφαλιστεί το φιλτράρισμα έγκυρων δεδομένων στο z-διανυσματικό χώρο.

$\min_{\theta,\varphi,D,A}$ J = $J_1 + \lambda_1 J_2 + \lambda_3 J_3$ =

$= D_{KL}[Q_\varphi(\text{z|X})\| P_\theta(\text{z})] - E_{Q_\varphi(\text{z|X})}[\log P_\theta(\text{X|z})]$

$+ \lambda_1 \frac{1}{N} \sum_{i=1}^{N} (||z_i - D a_i||_2^2 + \lambda_2 ||a_i||_1) + \lambda_3 \text{R(W,b)}$

τέτοιο ώστε $||d_i||^2 \leq 1$, ∀ i = 1, 2, …, K,

(4.1)

όπου το θ και το φ είναι παράμετροι του αποκωδικοποιητή και του κωδικοποιητή αντίστοιχα. Στη συνέχεια, με βάση το τρικ επαναπροσδιορισμού που αναφέρθηκε παραπάνω, οι όροι $D_{KL}[Q_\varphi(\text{z|X})\| P_\theta(\text{z})]$ και $E_{Q_\varphi(\text{z|X})}[\log P_\theta(\text{X|z})]$ υπολογίζονται ως εξής:

$D_{KL}[Q_\varphi(\text{z|X})\| P_\theta(\text{z})] = -\frac{1}{2} \sum_{i=1}^{N} (1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2)$,

$E_{Q_\varphi(\text{z|X})}[\log P_\theta(\text{X|z})] = \frac{1}{L} \sum_{i=1}^{L} \log P(X|z^{(t)})$,

όπου, το L είναι ο αριθμός της βοηθητικής μεταβλητής θορύβου. Η συνθήκη περιορισμού $||d_i||^2 \leq 1$, ∀ i = 1, 2, …, K συμβάλλει στην αποφυγή ασήμαντων λύσεων και για την σταθεροποίηση της λύσης. Τέλος, το R(W,b) αντιπροσωπεύει την κανονικοποίηση παραμέτρων για τον κωδικοποιητή και τον αποκωδικοποιητή του μοντέλου, χρησιμοποιώντας γενικά την l_2 νόρμα.

Το πρόβλημα βελτιστοποίησης της αντικειμενικής συνάρτησης (4.1) δεν είναι πραγματικά ένα κυρτό πρόβλημα, αλλά αν δούμε κάθε μεταβλητή ξεχωριστά, η βέλτιστη λύση του προβλήματος μπορεί να επιτευχθεί. Επομένως, για την επίλυση του παραπάνω προβλήματος, εξετάζουμε το ενδεχόμενο χρησιμοποίησης μίας εναλλακτικής μεθόδου βελτιστοποίησης. Ο αλγόριθμος 4.1 (που θα παρουσιαστεί παρακάτω) δείχνει όλα τα βήματα της εκπαίδευσης του μοντέλου μας.

Αλγόριθμος 4.1 Εκπαιδευτικό πλαίσιο του προτεινόμενου μοντέλου

Προαπαιτούμενα:

Σύνολο δεδομένων εκπαίδευσης X = $[x_1, x_2, \ldots, x_k] \in R^{d \times N}$;

Παράμετροι:

Μέγεθος λεξικού K;

Ρυθμοί μάθησης $\lambda_1, \lambda_2, \lambda_3$;

Σφάλμα σύγκλισης ε;

Αριθμός των νευρώνων $r^{(m)}$ για όλα τα επίπεδα;

Τυχαία αρχικοποιημένα D;

Αρχικοποίηση των παραμέτρων του αποκωδικοποιητή P_θ και του κωδικοποιητή Q_φ.

Διαδικασία εκμάθησης

Όσο τα {θ, φ} δεν συγκλίνουν τότε:

1) Δειγματοληψία των $x_1, x_2, \ldots, x_n$ από το σύνολο δεδομένων εκπαίδευσης

2) Ενημερώστε τις παραμέτρους του αποκωδικοποιητή και του κωδικοποιητή {θ, φ} καθώς και τις παραμέτρους του αυτοκωδικοποιητή LISTA (συμπεριλαμβανομένου του λεξικού D) με φθίνουσα σειρά χρησιμοποιώντας την οπισθοδρόμηση:

J = $D_{KL}[Q_\varphi(\text{z|X})\| P_\theta(\text{z})] - E_{Q_\varphi(\text{z|X})}[\log P_\theta(\text{X|z})] + \lambda_1 \frac{1}{N} \sum_{i=1}^{N}(||z_i - Da_i||_2^2 + C) + \lambda_3 \text{R(W,b)}$;

3) Επιδιόρθωση των {θ, φ}.

Κεφάλαιο 5.

Παρουσίαση των πειραμάτων και των αποτελεσμάτων

5.1 Σύνολα δεδομένων

Αρχικά τα σύνολα εκπαίδευσης και ελέγχου κατασκευάστηκαν ως εξής:

Για το σύνολο εκπαίδευσης, αρχικά φορτώσαμε το σύνολο δεδομένων moving MNIST, το οποίο είναι ένα σύνολο δεδομένων που περιέχει 10000 βίντεο, που το κάθε βίντεο αποτελείται από frames διάστασης 64x64. Στη συνέχεια από το σύνολο δεδομένων μας πήραμε τα πρώτα 8000 βίντεο ως σύνολο εκπαίδευσης. Από αυτά επιλέξαμε τυχαία τα 5000 βίντεο ώστε η εκπαίδευση του μοντέλου μας να γίνει πάνω σε 5000 βίντεο. Μετέπειτα, μετατρέψαμε τα 5000 βίντεο σε ένα σύνολο από 100000 δείγματα, τα οποία αντιστοιχούν σε εικόνες, ώστε να καταλήξουμε στην μορφή (100000,64,64). Τώρα ως τελικό βήμα, μετατρέψαμε κάθε εικόνα από 64x64 σε 28x28 με την χρήση της παρεμβολής του πιο κοντινού γείτονα (nearest neighbor interpolation), ώστε να καταλήξουμε σε διάσταση (100000,28,28) και στη συνέχεια με μία ακόμα μετατροπή καταλήγουμε σε διάσταση (100000,784). Η μετατροπή της διάστασης των καρέ των βίντεο έγινε, ώστε το μοντέλο μας να μπορεί να εκπαιδευτεί πάνω σε αυτά τα καρέ.

Για το σύνολο ελέγχου φτιάξαμε ένα νέο αρχείο, όπου εκεί φορτώνουμε πάλι το σύνολο δεδομένων moving MNIST. Στη συνέχεια παίρνουμε τα 2000 τελευταία βίντεο του συνόλου δεδομένων ως σύνολο ελέγχου. Από αυτά, τα πρώτα 1000 βίντεο τα κρατάμε ως κανονικό σύνολο ελέγχου και δεν κάνουμε καμία περαιτέρω μετατροπή. Στα υπόλοιπα 1000 βίντεο, που αποτελούν το παρεφθαρμένο (corrupted) σύνολο ελέγχου, αυτό που κάνουμε είναι το εξής. Για κάθε βίντεο, σε όλα τα καρέ του τοποθετούμε ένα τετράγωνο διάστασης 6x6, σε τυχαία επιλεγμένη θέση μέσα στο καρέ, ώστε να προσομοιώσουμε την εμφάνιση ενός ανώμαλου γεγονότος.

Για το μοντέλο VAE ισχύουν τα εξής ως προς το σύνολο εκπαίδευσης και το σύνολο ελέγχου:

- Ως σύνολο εκπαίδευσης χρησιμοποιούμε αυτό που περιγράφτηκε.
- Ως σύνολο ελέγχου χρησιμοποιούμε τα βίντεο του κανονικού συνόλου ελέγχου moving MNIST

Για το μοντέλο LISTA και το μοντέλο ανίχνευσης ανωμαλιών ισχύουν τα εξής ως προς το σύνολο εκπαίδευσης και το σύνολο ελέγχου:

- Για να φτιαχτεί το σύνολο εκπαίδευσης θα πρέπει να γίνουν κάποια βήματα:
    - Αρχικά φορτώνεται το μοντέλο VAE, μαζί με τα βάρη που προέκυψαν κατά την εκπαίδευση του.
    - Στη συνέχεια, χρησιμοποιώντας τον κωδικοποιητή του μοντέλου VAE με είσοδο το σύνολο εκπαίδευσης που περιγράφτηκε παραπάνω, παίρνουμε ως έξοδο τα z με τα οποία θα γίνει η εκπαίδευση των 2 μοντέλων.
- Για το μοντέλο LISTA χρησιμοποιείται ως σύνολο ελέγχου τα βίντεο του κανονικού συνόλου ελέγχου moving MNIST.
- Για το μοντέλο ανίχνευσης ανωμαλιών χρησιμοποιείται ως σύνολο ελέγχου και τα βίντεο του κανονικού συνόλου ελέγχου moving MNIST αλλά και τα βίντεο του ανώμαλου συνόλου ελέγχου corrupted moving MNIST.

5.2 Λεπτομέρειες υλοποίησης

Σε αυτή την ενότητα θα ακολουθήσει ανάλυση των λεπτομερειών σχεδίασης και εκπαίδευσης κάθε μοντέλου.

5.2.1 Λεπτομέρειες σχεδίασης

Για το μοντέλο VAE, στο κομμάτι του κωδικοποιητή χρησιμοποιούμε ένα επίπεδο εισόδου. Στη συνέχεια χρησιμοποιούμε ένα dense επίπεδο με 400 units και άλλα δύο Dense επίπεδα, που αποτελούνται από 392 units το καθένα, που αντιστοιχούν στην διάσταση για τον χώρο κρυμμένης κωδικοποίησης. Έτσι προκύπτουν η μέση τιμή και η συνδιακύμανση του z-χώρου. Μετά παίρνουμε την μέση τιμή και την συνδιακύμανση που προέκυψαν στο προηγούμενο βήμα και της βάζουμε σε μία συνάρτηση δειγματοληψίας μέσω ενός Lambda επιπέδου και έτσι προκύπτει ο z-χώρος. Σε αυτό το βήμα εκμεταλλευόμαστε το τρικ επαναπροσδιορισμού που περιγράφτηκε στην ενότητα 2.5.2 και το υλοποιεί η συνάρτηση δειγματοληψίας. Μέσω του Lambda επιπέδου το αποτέλεσμα της δειγματοληψίας μπορεί να χρησιμοποιηθεί ως επίπεδο στο μοντέλο μας. Ο αποκωδικοποιητής του VAE μοντέλου αποτελείται από 2 Dense επίπεδα, το πρώτο έχει 400 units, όπως το αντίστοιχο Dense επίπεδο του κωδικοποιητή και παίρνει ως είσοδο μέρος του z-χώρου που προέκυψε από τον κωδικοποιητή και το δεύτερο Dense επίπεδο έχει 784 units και μάς δίνει μία εικόνα σαν αποτέλεσμα.

Για το μοντέλο LISTA, η σχέση που υπολογίζει κάθε επίπεδο είναι η σχέση 3.7. Επίσης η διάσταση του λεξικού που δημιουργείται μέσω των δύο αυτών μοντέλων είναι (392, 392), όπου το 392 είναι η διάσταση του χώρου κρυμμένης κωδικοποίησης. Τέλος η διάσταση του πίνακα των βαρών των μοντέλων μας είναι (784, 392) και η διάσταση του πίνακα S είναι (784, 784).

5.2.2 Λεπτομέρειες εκπαίδευσης

Το μοντέλο VAE το εκπαιδεύουμε με μία προσαρμοσμένη συνάρτηση λάθους. Αυτή η συνάρτηση υπολογίζει το σφάλμα ανακατασκευής μεταξύ των εικόνων που έχουμε σαν είσοδο και σαν έξοδο από το μοντέλο μας και υπολογίζει επίσης την απόκλιση Kullback Leibler μεταξύ της εκπαιδευμένης κατανομής του χώρου κρυμμένης κωδικοποίησης και της προηγούμενης κατανομής, λειτουργώντας ως όρος κανονικόποιησης.

$loss = \frac{\sum_{j=1}^{J} -\frac{1}{2} \sum_{i=1}^{N}(1+\log(\sigma_i{}^2) - \mu_i{}^2 - \sigma_i{}^2) - \frac{1}{L}\sum_{i=1}^{L} \log P(X|z^{(t)})}{J}$, όπου το $\sigma_i{}^2$ είναι η συνδιακύμανση και το μ_i είναι η μέση τιμή του χώρου κρυμμένης κωδικοποίησης, το J είναι η διάσταση του χώρου κρυμμένης κωδικοποίησης, το L είναι ο αριθμός της βοηθητικής μεταβλητής θορύβου και το $P(X|z^{(t)})$ είναι η πιθανότητα να πάρουμε την είσοδο X του μοντέλου μας, δεδομένου του z-χώρου.

Στη συνέχεια εκπαιδεύουμε το μοντέλο μας για είσοδο το σύνολο εκπαίδευσης που περιγράφτηκε παραπάνω και για 100 εποχές και λέμε πως η έξοδος του μοντέλου μας θέλουμε να μοιάζει με την είσοδο του μοντέλου μας. Επιπρόσθετα χρησιμοποιούμε μέγεθος συνόλου εκπαίδευσης 128, βελτιστοποιητή τον Adam με ρυθμό μάθησης 0.001 και εκπαιδεύουμε το μοντέλο μας για 100 εποχές.

Για το μοντέλο LISTA, όπως και για το μοντέλο ανίχνευσης ανωμαλιών, το μέγεθος του συνόλου εκπαίδευσης και ο βελτιστοποιητής είναι ίδιοι με το μοντέλο VAE. Αυτό που αλλάζει σε αυτά τα δύο μοντέλα, είναι πως βελτιστοποιούνται με βάση την συνάρτηση λάθους του μέσου τετραγωνικού σφάλματος της εισόδου και της εξόδου των μοντέλων μας.

MSE $= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})$, όπου το y_i αντιστοιχεί στην έξοδο του μοντέλου μας , το $\hat{y_i}$ στην είσοδο του μοντέλου μας και το N αντιστοιχεί στο μέγεθος του δείγματος.

Αλλάζει επίσης και ο αριθμός των εποχών που τώρα είναι 10.

5.3 Αποτελέσματα ανακατασκευής

Για τον VAE, ως μέτρο ανακατασκευής χρησιμοποιούμε την μετρική PSNR (peak signal noise ratio) μεταξύ της αρχικής εικόνας που δίνουμε ως είσοδο στο μοντέλο μας και της εικόνας που πήραμε ως αποτέλεσμα από το μοντέλο μας.

$PSNR = 10log_{10}\left(\frac{(L-1)^2}{MSE}\right) = 20log_{10}\left(\frac{(L-1)}{RMSE}\right)$, όπου το L είναι ο αριθμός των μέγιστων δυνατών επίπεδων έντασης (το ελάχιστο επίπεδο έντασης υποθέτουμε πως είναι ίσο με 0) σε 1 εικόνα. Εδώ το MSE περιγράφεται από την εξής τύπο:

$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(O(i,j) - D(i,j))^2$, όπου το O αντιπροσωπεύει τα δεδομένα (σε μορφή πίνακα) της κανονικής εικόνας, το D αντιπροσωπεύει τα δεδομένα (σε μορφή πίνακα) της υποβαθμισμένης εικόνας. Το m αντιπροσωπεύει τον αριθμό των σειρών των εικονοστοιχείων και το i αντιπροσωπεύει τον δείκτη της σειράς της εικόνας. Το n αντιπροσωπεύει τον αριθμό των στηλών των εικονοστοιχείων και τον j αντιπροσωπεύει τον δείκτη της στήλης της εικόνας. Το RMSE είναι η ρίζα του MSE.

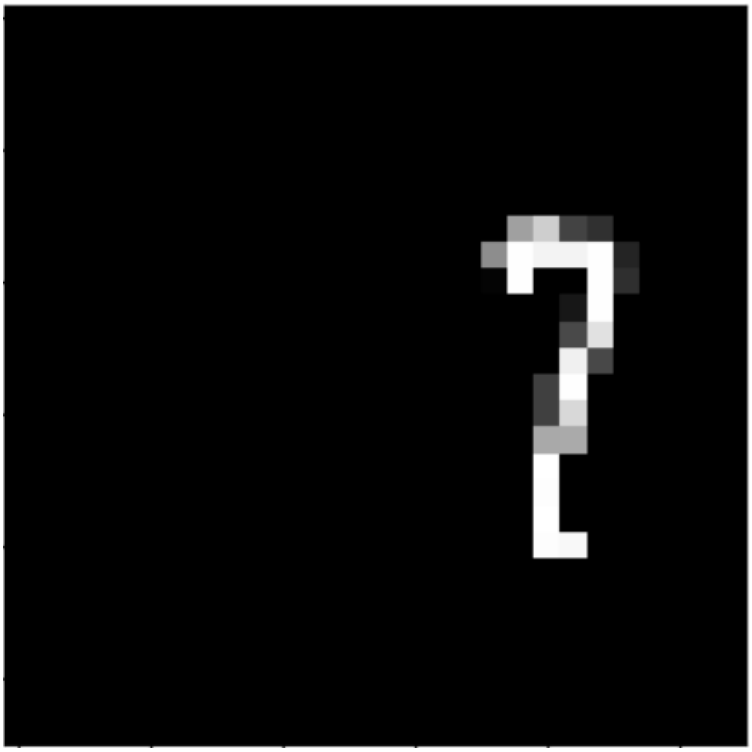
Τέλος αυτή η μετρική προκύπτει ως ο μέσος όρος των PSNR των 20 καρέ που έχει κάθε βίντεο.

Το αποτέλεσμα που προέκυψε για το PSNR είναι 26.522. Ενδεικτικά ακολουθούν 4 καρέ, στις Εικόνες 5.1 και 5.2, όπου το πρώτο καρέ είναι το καρέ που έχουμε ως δείγμα και το δεύτερο είναι αυτό που μας δίνει το μοντέλο VAE σαν πρόβλεψη.

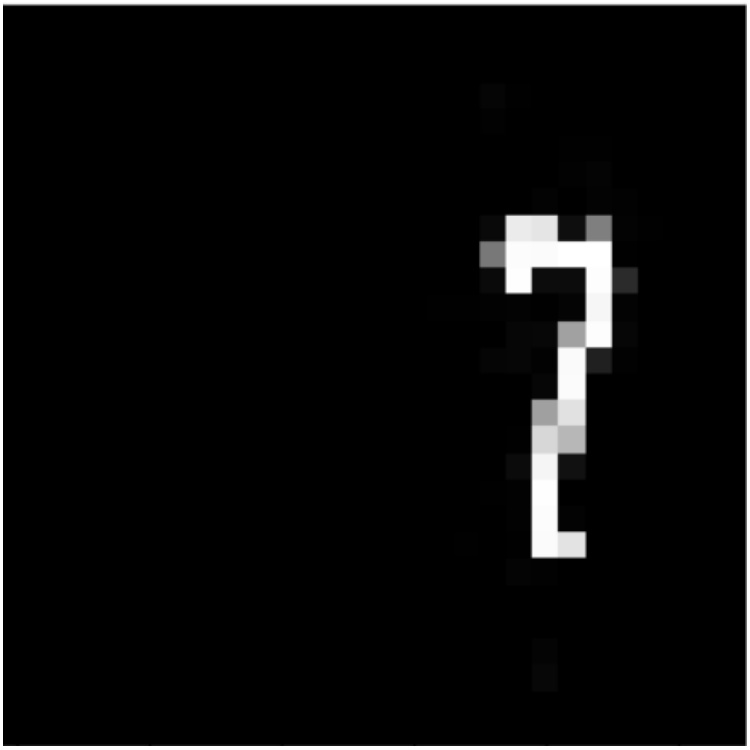
Για το LISTA, ως αποτέλεσμα ανακατασκευής εννοούμε την μετρική MSE (mean squared error) που περιγράφτηκε στην ενότητα 5.2.2. Τέλος αυτή η μετρική προκύπτει ως ο μέσος όρος των MSE των 20 καρέ που έχει κάθε βίντεο. Το αποτέλεσμα που πήραμε είναι 5.574e-05. Ενδεικτικά ακολουθεί η Εικόνα 5.3, που δείχνουμε τον αρχικό z-χώρο και τον z-χώρο που παράγει ο LISTA ως πρόβλεψη.

Επίσης ακολουθεί η εικόνα 5.4 και η εικόνα 5.5 που μας δείχνει την ανακατασκευή ανώμαλης εικόνας με το μοντέλο VAE και η εικόνα 5.6 και 5.7 που μας δείχνουν τις αντίστοιχες ανακατασκευές ανώμαλων εικόνων του μοντέλου LISTA.

Καρέ εισόδου:

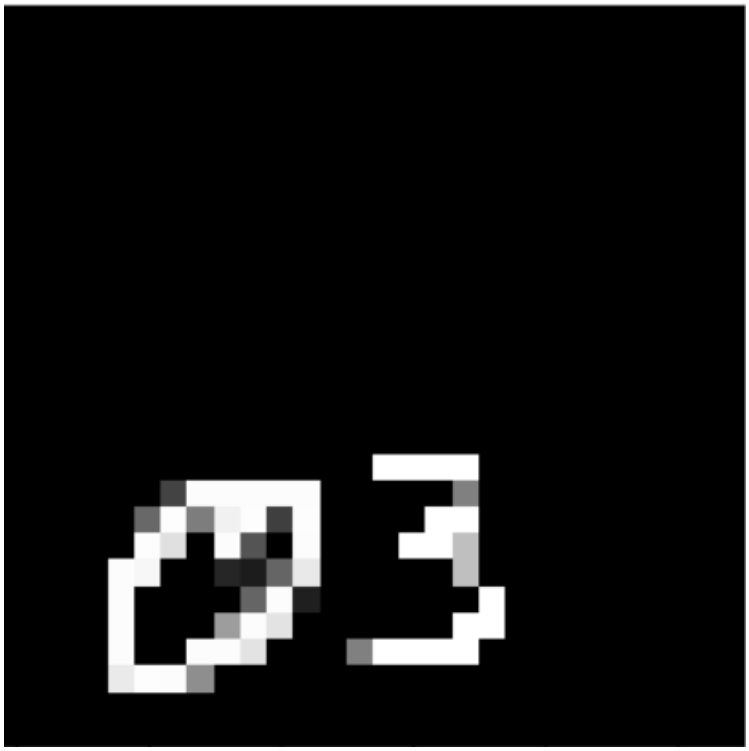


Πρόβλεψη του μοντέλου VAE:

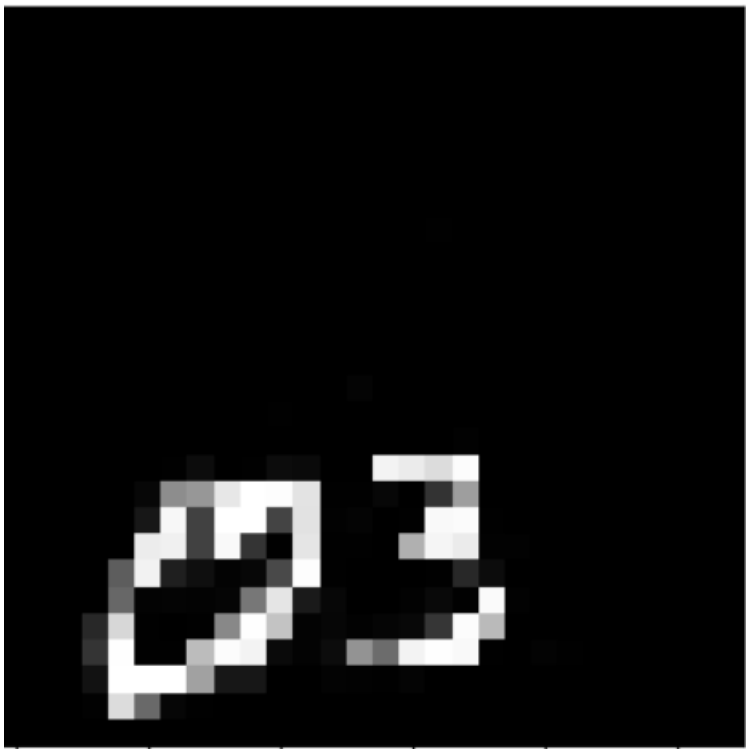


Εικόνα 5.1: Παράδειγμα ανακατασκευής με το μοντέλο VAE

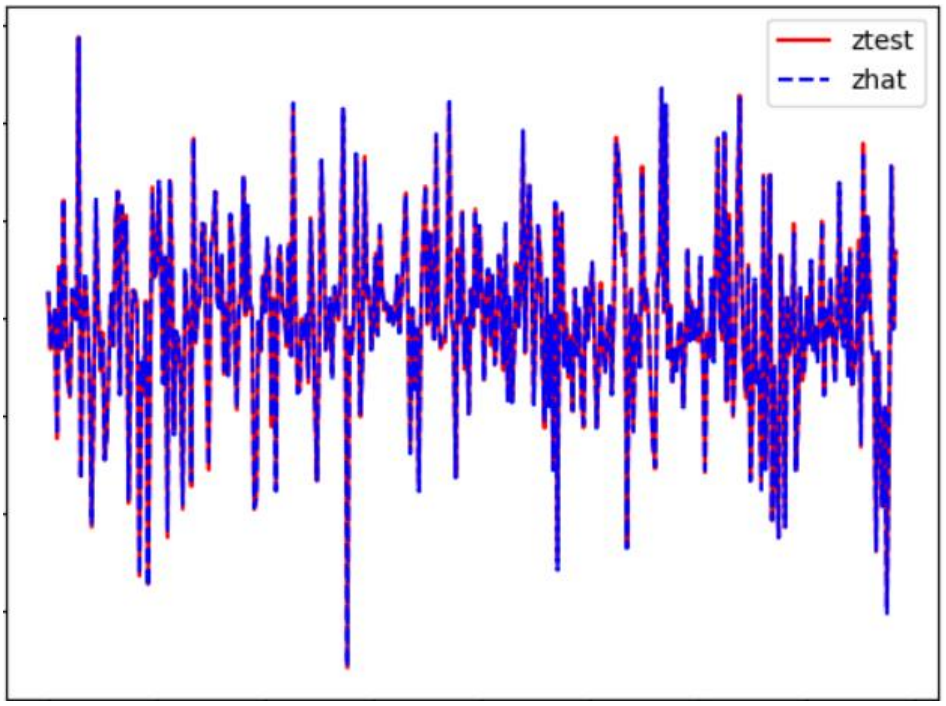
Καρέ εισόδου:



Πρόβλεψη του μοντέλου VAE:

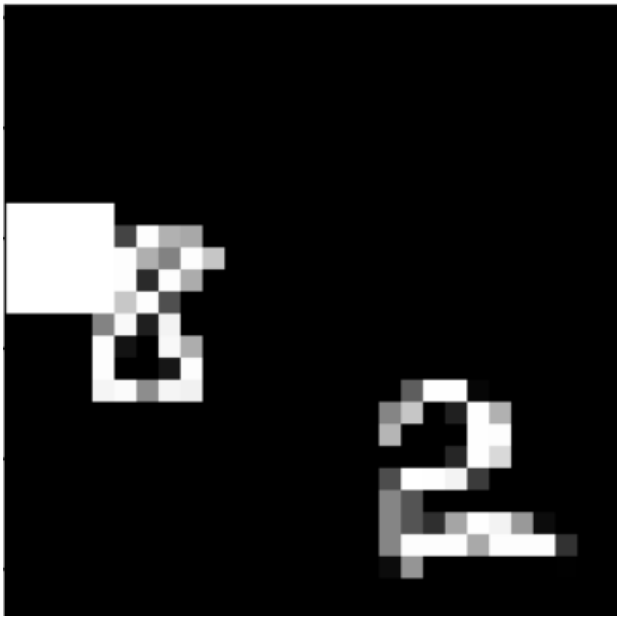


Εικόνα 5.2: Παράδειγμα ανακατασκευής με το μοντέλο VAE.

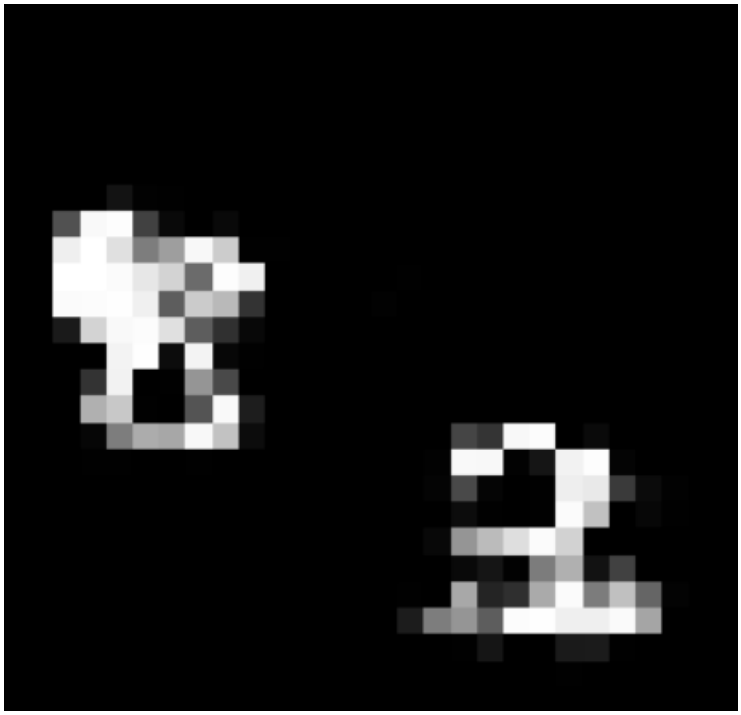


Εικόνα 5.3: Παράδειγμα ανακατασκευής με το μοντέλο LISTA.

Αρχικό καρέ:

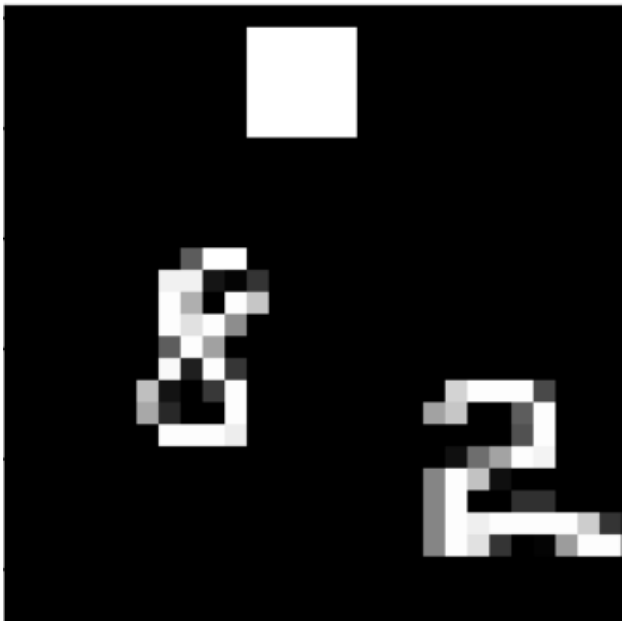


Πρόβλεψη VAE:

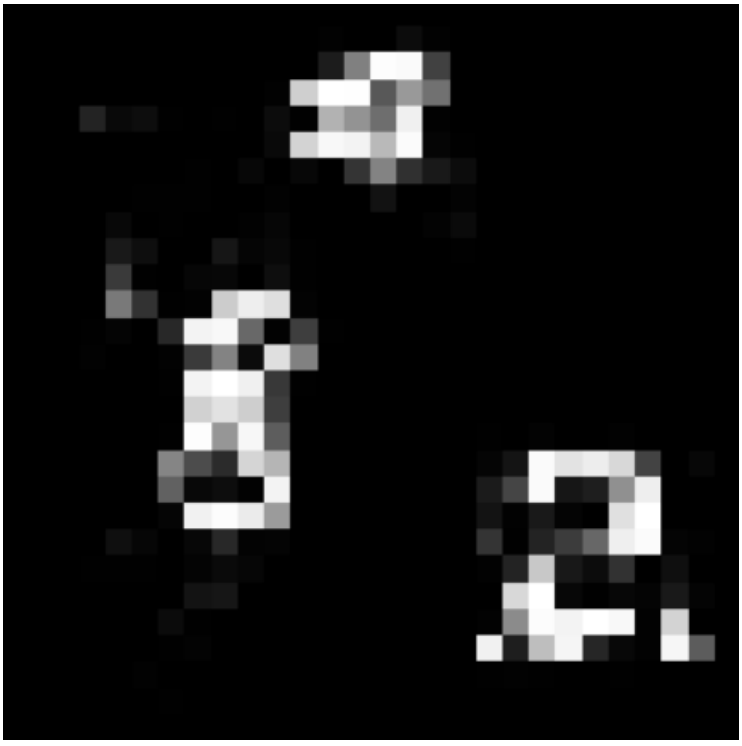


Εικόνα 5.4: Παράδειγμα ανακατασκευής ανώμαλης εικόνας με το μοντέλο VAE.

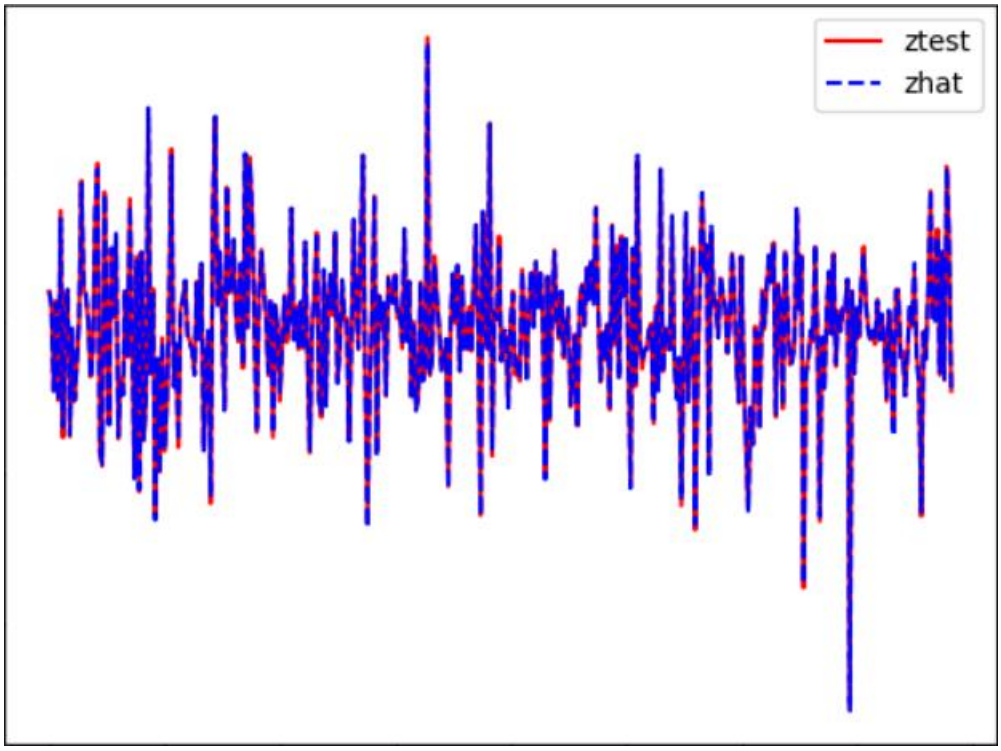
Αρχικό καρέ:



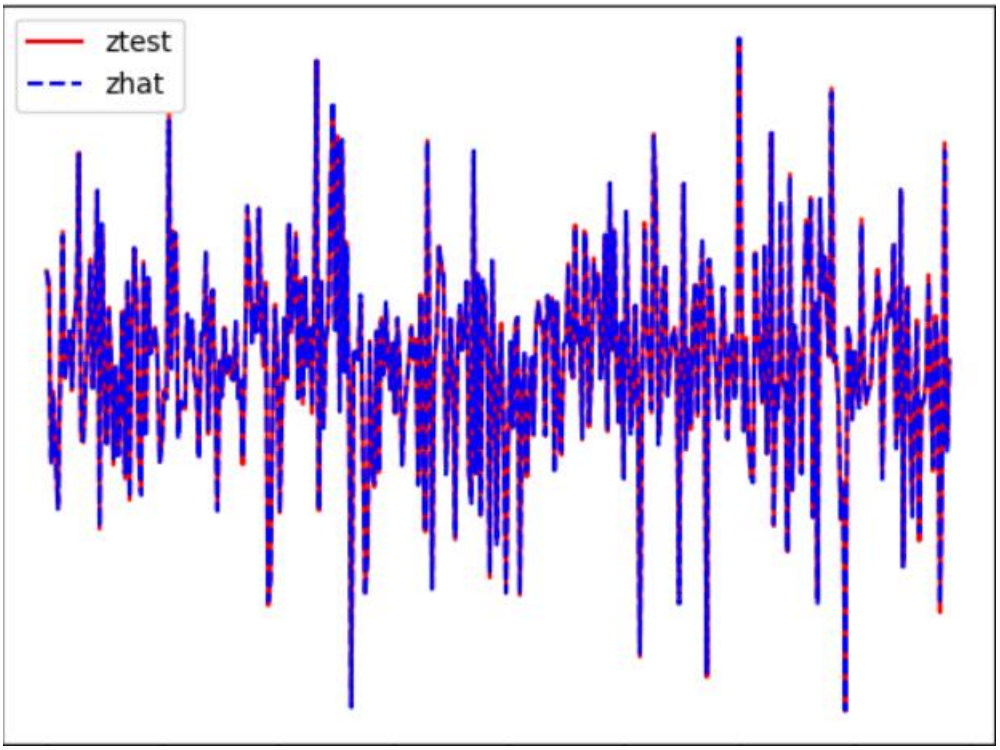
Πρόβλεψη VAE:



Εικόνα 5.5: Παράδειγμα ανακατασκευής ανώμαλης εικόνας με το μοντέλο VAE.



Εικόνα 5.6: Ανακατασκευή με βάση το αρχικό καρέ της εικόνας 5.4 με τον LISTA.



Εικόνα 5.7: Ανακατασκευή με βάση το αρχικό καρέ της εικόνας 5.5 με τον LISTA.

5.4 Αποτελέσματα ταξινόμησης

Αυτή η ενότητα αφορά τα αποτελέσματα ταξινόμησης που μας δίνει το μοντέλο ανίχνευσης ανωμαλιών που αναλύθηκε παραπάνω.

Αρχικά για την ταξινόμηση των εικόνων χρησιμοποιήσαμε τα εξής κριτήρια. (1) θεωρήσαμε πως αν το μέσο τετραγωνικό σφάλμα μεταξύ του αρχικού χώρου κρυμμένης κωδικοποίησης και του κρυμμένου χώρου κωδικοποίησης που έχουμε ως έξοδο από το LISTA είναι μεγαλύτερο από το κατώφλι mse_thershold (mse_th) που έχουμε ορίσει, τότε το βίντεο είναι ανώμαλο. (2) θεωρήσαμε πως αν το psnr του βίντεο, που αναλύθηκε στην προηγούμενη ενότητα πως προκύπτει, είναι μικρότερο από το κατώφλι psnr_thershold (psnr_th), τότε το βίντεο είναι ανώμαλο. (3) αποτελεί συνδυασμό των κριτηρίων (1) και (2) και ουσιαστικά θεωρούμε πως αν ισχύει είτε η συνθήκη του κριτηρίου (1) είτε η συνθήκη του κριτηρίου (2), τότε ένα βίντεο είναι ανώμαλο. (4) αποτελεί επίσης συνδυασμό των κριτηρίων (1) και (2) και ουσιαστικά θεωρούμε πως αν ισχύει και η συνθήκη του κριτηρίου (1) και η συνθήκη του κριτηρίου (2), τότε ένα βίντεο είναι ανώμαλο.

Για να αξιολογήσουμε την αποτελεσματικότητα του μοντέλου της ανίχνευσης ανωμαλιών, στηριχθήκαμε σε κάποιες μετρικές. Αυτές είναι οι accuracy, precision, recall και f1 score. Όμως πρώτα θα πρέπει να αναλυθούν κάποιες παράμετροι που χρησιμοποιήσαμε για να καταλήξουμε στα αποτελέσματα των μετρικών που αναφέρθηκαν.

Αρχικά, ορίζουμε ως κλάση την anomalous, δηλαδή στην περίπτωση μας είναι ένα ανώμαλο βίντεο. True Positives (TP) είναι οι σωστά προβλεπόμενες θετικές τιμές, που σημαίνει ότι ενώ το βίντεο είναι ανώμαλο, του μοντέλο ανίχνευσης ανωμαλιών ταξινομεί το βίντεο ως ανώμαλο. True Negatives (TN) είναι οι σωστά προβλεπόμενες αρνητικές τιμές, που σημαίνει πως ενώ το βίντεο είναι κανονικό, το μοντέλο ανίχνευσης ανωμαλιών ταξινομεί το βίντεο ως κανονικό. False Positives (FP) σημαίνει πως ενώ το βίντεο είναι κανονικό, το μοντέλο της ανίχνευσης ανωμαλιών ταξινομεί το βίντεο ως ανώμαλο. False Negatives (FN) σημαίνει πως ενώ το βίντεο είναι ανώμαλο, το μοντέλο της ανίχνευσης ανωμαλιών ταξινομεί το βίντεο ως κανονικό.

Η μετρική accuracy είναι ο λόγος σωστά προβλεπόμενων παρατηρήσεων προς το σύνολο των παρατηρήσεων και προκύπτει από την σχέση

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

Η μετρική precision είναι ο λόγος των σωστά προβλεπόμενων ανώμαλων βίντεο προς το σύνολο των ανώμαλων βίντεο και προκύπτει από την σχέση

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

Η μετρική recall είναι ο λόγος των σωστά προβλεπόμενων ανώμαλων βίντεο προς όλες τις παρατηρήσεις στην κλάση anomalous, δηλαδή των βίντεο που ενώ ήταν ανώμαλα, το μοντέλο της ανίχνευσης ανωμαλιών τα ταξινόμησε ως ανώμαλα και των βίντεο που ενώ είναι ανώμαλα, το μοντέλο ανίχνευσης ανωμαλιών τα ταξινόμησε ως κανονικά.

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

Τέλος η μετρική f1 score, είναι ο σταθμισμένος μέσος όρος των μετρικών recall και precision. Επιπρόσθετα αυτή η μετρική παίρνει υπόψιν της και τα ψευδώς ανώμαλα βίντεο και τα ψευδώς κανονικά βίντεο και προκύπτει από την σχέση

$$F1\ score = \frac{2(RecallPrecision)}{(Recall + Precision)}$$

Όμως, πριν εκτελέσουμε τα πειράματα για να αξιολογήσουμε τις μετρικές accuracy, precision, recall και f1_score, εκτελέσαμε 2 πειράματα ώστε να εκτιμήσουμε τις κατάλληλες τιμές για το κατώφλι του psnr, psnr_th, αλλά και για το κατώφλι του mse, mse_th. Το πρώτο πείραμα εκτελέστηκε για 100 βίντεο του συνόλου ελέγχου των κανονικών βίντεο και το δεύτερο πείραμα εκτελέστηκε για 100 βίντεο του συνόλου ελέγχου των παρεφθαρμένων βίντεο. Στα δύο αυτά πειράματα εκτιμήσαμε τις μετρικές psnr και mse και καταλήξαμε πως οι ιδανικές αρχικές τιμές για το κατώφλι του psnr ήταν 25.5 και για το κατώφλι του mse ήταν 4.2e-05.

Στα επόμενα πειράματα που κάναμε, διαχωρίσαμε τα βίντεο σε κανονικά ή ανώμαλα με βάση τα κριτήρια (1), (2), (3) και (4) που αναλύθηκαν στην αρχή της ενότητας και αλλάζοντας κάθε φορά είτε το κατώφλι του psnr είτε το κατώφλι του mse, είτε και τα δύο μαζί, καταλήξαμε σε κάποια αποτελέσματα για τις μετρικές accuracy, precision, recall και f1 score.

Τιμές Κατωφλίου		Accuracy			
PSNR	z-MSE	(1)	(2)	(3)	(4)
24.5	4.2e-05	0.746	0.945	0.745	**0.946**
24.5	4.7e-05	0.868	0.944	0.871	0.941
24.5	5e-05	**0.908**	**0.948**	**0.926**	0.930
25.0	4.2e-05	0.737	0.919	0.732	0.924
25.0	4.7e-05	0.874	0.919	0.858	0.935
25.0	5e-05	0.896	0.920	0.897	0.919
25.5	4.2e-05	0.737	0.871	0.723	0.885
25.5	4.7e-05	0.879	0.873	0.829	0.923
25.5	5e-05	0.897	0.875	0.857	0.914

Πίνακας 5.1: Αποτελέσματα ανίχνευσης ανωμαλιών με βάση τέσσερα διαφορετικά κριτήρια και για διαφορετικές τιμές κατωφλίου.

Τιμές Κατωφλίου		Precision			
PSNR	z-MSE	(1)	(2)	(3)	(4)
24.5	4.2e-05	0.667	**0.951**	0.663	0.953
24.5	4.7e-05	0.806	0.944	0.799	0.959
24.5	5e-05	**0.897**	0.949	**0.884**	**0.968**
25.0	4.2e-05	0.655	0.875	0.651	0.883
25.0	4.7e-05	0.810	0.877	0.782	0.915
25.0	5e-05	0.887	0.876	0.834	0.941
25.5	4.2e-05	0.656	0.801	0.644	0.820
25.5	4.7e-05	0.817	0.803	0.746	0.889
25.5	5e-05	0.890	0.806	0.781	0.927

Πίνακας 5.2: Αποτελέσματα ανίχνευσης ανωμαλιών με βάση τέσσερα διαφορετικά κριτήρια και για διαφορετικές τιμές κατωφλίου.

Τιμές Κατωφλίου		Recall			
PSNR	z-MSE	(1)	(2)	(3)	(4)
24.5	4.2e-05	0.999	0.939	0.999	0.939
24.5	4.7e-05	0.969	0.944	0.992	0.921
24.5	5e-05	0.922	0.948	0.981	0.889
25.0	4.2e-05	**1.000**	0.977	**1.000**	0.977
25.0	4.7e-05	0.977	0.975	0.994	0.958
25.0	5e-05	0.907	0.978	0.991	0.894
25.5	4.2e-05	0.999	0.987	0.999	**0.987**
25.5	4.7e-05	0.977	**0.988**	0.998	0.967
25.5	5e-05	0.906	0.987	0.993	0.900

Πίνακας 5.3: Αποτελέσματα ανίχνευσης ανωμαλιών με βάση τέσσερα διαφορετικά κριτήρια και για διαφορετικές τιμές κατωφλίου.

Τιμές Κατωφλίου		F1 score			
PSNR	z-MSE	(1)	(2)	(3)	(4)
24.5	4.2e-05	0.797	0.945	0.797	**0.946**
24.5	4.7e-05	0.880	0.944	0.885	0.940
24.5	5e-05	**0.909**	**0.948**	**0.930**	0.927
25.0	4.2e-05	0.792	0.923	0.788	0.928
25.0	4.7e-05	0.885	0.924	0.875	0.936
25.0	5e-05	0.897	0.924	0.906	0.917
25.5	4.2e-05	0.792	0.884	0.783	0.896
25.5	4.7e-05	0.890	0.886	0.854	0.926
25.5	5e-05	0.898	0.887	0.874	0.913

Πίνακας 5.4: Αποτελέσματα ανίχνευσης ανωμαλιών με βάση τέσσερα διαφορετικά κριτήρια και για διαφορετικές τιμές κατωφλίου

Κεφάλαιο 6.

Συμπεράσματα

Το πρόβλημα της ανίχνευσης ανωμαλιών σε δεδομένα υψηλής διάστασης, μπορεί να οριστεί ως ο προσδιορισμός σπάνιων στοιχείων, γεγονότων ή παρατηρήσεων που δημιουργούν υποψίες διαφέροντας σημαντικά από την πλειοψηφία των δεδομένων. Αυτό στον πραγματικό κόσμο μπορεί να εκφραστεί ως η εμφάνιση ενός αυτοκινήτου σε έναν δρόμο ή ως η εμφάνιση ενός ανθρώπου σε έναν δρόμο με αμάξια. Για αυτό το πρόβλημα, προτείνουμε ένα νέο πλαίσιο επίλυσης του προβλήματος χωρίς επίβλεψη που ενσωματώνει την αραιή ανακατασκευή μέσα στον αυτοκωδικοποιητή μεταβολής, το οποίο μπορεί να διαχωριστεί σε δύο μέρη: την μοντελοποίηση κρυφών πληροφοριών και την αραιή κωδικοποίηση αυτών των πληροφοριών. Οι πληροφορίες του χώρου κρυμμένης κωδικοποίησης στο μοντέλο κατασκευάστηκαν από τον αυτοκωδικοποιητή μεταβολής για να εξαχθούν οι χαρακτηριστικές πληροφορίες των δεδομένων εισόδου. Εν τω μεταξύ οι πληροφορίες του χώρου κρυμμένης κωδικοποίησης μπορούν να παίξουν τον ρόλο της μείωσης διαστάσεων για δεδομένα μεγάλης κλίμακας. Η κατασκευή λεξικού των δεδομένων του χώρου κρυμμένης κωδικοποίησης, χρησιμοποιείται για την αξιολόγηση του βαθμού ανωμαλίας των δειγμάτων ελέγχου. Αυτή η μέθοδος όχι μόνο μειώνει το κόστος χώρου, αλλά παρέχει επίσης μη γραμμική είσοδο για εκμάθηση λεξικού, που καθιστά το μοντέλο πιο σταθερό. Τα αποτελέσματα των πειραμάτων που έγιναν με τα βίντεο του συνόλου ελέγχου του moving MNIST και τα βίντεο του συνόλου ελέγχου του corrupted moving MNIST, δείχνουν πως το μοντέλο μας ταξινομεί σε πολύ μεγάλο βαθμό σωστά τα βίντεο σε ανώμαλα ή κανονικά αλλά και πως έχει πολύ καλά αποτελέσματα συγκριτικά με άλλες μεθόδους.

Βιβλιογραφία

[BBo1988] Barzilai, J., & Borwein, J. M. (1988). Two-point step size gradient methods. IMA journal of numerical analysis, 8(1), 141-148.

[Cou1994] Courant, R. (1994). Variational methods for the solution of problems of equilibrium and vibrations. Lecture notes in pure and applied mathematics, 1-1.

[Cur1944] Curry, H. B. (1944). The method of steepest descent for non-linear minimization problems. Quarterly of Applied Mathematics, 2(3), 258-261.

[DDD2004] Daubechies, I., Defrise, M., & De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, 57(11), 1413-1457.

[Fle2005] Fletcher, R. (2005). On the barzilai-borwein method. In Optimization and control with applications (pp. 235-256). Springer, Boston, MA.

[GBC2016] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[GBC2016] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Back-propagation and other differentiation algorithms. Deep Learning

[GLe2010] Gregor, K., & LeCun, Y. (2010, June). Learning fast approximations of sparse coding. In Proceedings of the 27th international conference on international conference on machine learning (pp. 399-406).

[HAu2004] Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. Artificial intelligence review, 22(2), 85-126.

[HKW2011] Hinton, G. E., Krizhevsky, A., & Wang, S. D. (2011, June). Transforming auto-encoders. In International conference on artificial neural networks (pp. 44-51). Springer, Berlin, Heidelberg.

[KBa2004] Kussul, E., & Baidyk, T. (2004). Improved method of handwritten digit recognition tested on MNIST database. Image and Vision Computing, 22(12), 971-981.

[KDW2019] Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. arXiv preprint arXiv:1906.02691.

[KPa2015] Kampourlazos, V., & Papakostas, G. (2015). INTRODUCTION TO COMPUTATIONAL INTELLIGENCE.

[Kra1991] Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. AIChE journal, 37(2), 233-243.

[KTP2018] Kiran, B. R., Thomas, D. M., & Parakkal, R. (2018). An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. Journal of Imaging, 4(2), 36.

[LCL2014] Liou, C. Y., Cheng, W. C., Liou, J. W., & Liou, D. R. (2014). Autoencoder for words. Neurocomputing, 139, 84-96.

[Lem2012] Lemaréchal, C. (2012). Cauchy and the gradient method. Doc Math Extra, 251(254), 10.

[LHY2008] Liou, C. Y., Huang, J. C., & Yang, W. C. (2008). Modeling word perception using the Elman network. Neurocomputing, 71(16-18), 3150-3157.

[LIM2018] Lucas, A., Iliadis, M., Molina, R., & Katsaggelos, A. K. (2018). Using deep neural networks for inverse problems in imaging: beyond analytical methods. IEEE Signal Processing Magazine, 35(1), 20-36.

[Pol1987] Polyak, B. T. (1987). Introduction to optimization. optimization software. Inc., Publications Division, New York, 1.

[RSc2008] Ribes, A., & Schmitt, F. (2008). Linear inverse problems in imaging. IEEE Signal Processing Magazine, 25(4), 84-99.

[SWX2018] Sun, J., Wang, X., Xiong, N., & Shao, J. (2018). Learning sparse representation with variational auto-encoder for anomaly detection. IEEE Access, 6, 33353-33361.

[TWr2010] Tropp, J. A., & Wright, S. J. (2010). Computational methods for sparse solution of linear inverse problems. Proceedings of the IEEE, 98(6), 948-958.

[YWH2010] Yang, J., Wright, J., Huang, T. S., & Ma, Y. (2010). Image super-resolution via sparse representation. IEEE transactions on image processing, 19(11), 2861-2873.

[VLL2010] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., & Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(12).

[ZHZ2016] Zhu, P., Hu, Q., Zhang, C., & Zuo, W. (2016, March). Coupled dictionary learning for unsupervised feature selection. In Thirtieth AAAI Conference on Artificial Intelligence.

[ZSc2017] Zimek, A., & Schubert, E. (2017). Outlier detection, encyclopedia of database systems. Springer, 10, 978-1.