

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

**ΘΕΜΑΤΑ ΟΡΑΣΗΣ ΥΠΟΛΟΓΙΣΤΩΝ & ΓΡΑΦΙΚΗΣ**

**ΑΝΑΦΟΡΑ ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ**

**5Η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ**

ΑΓΓΕΛΟΣ ΝΙΚΟΛΑΟΣ ΠΟΤΑΜΙΑΝΟΣ

Α.Μ. 1084537

up1084537@ac.upatras.gr

Πάτρα, 2024

**ΑΠΟΔΕΙΞΕΙΣ**

**Αποδείξτε ότι η επιθυµητή λύση προκύπτει από τη λύση**

**του ακόλουθου µε περιορισµούς προβλήµατος ϐελτιστοποίησης :**

Βήμα 1: Διατύπωση του προβλήματος χρησιμοποιώντας τη συνάρτηση Lagrange

Για να μεγιστοποιήσουμε τη συνάρτηση υπό τον περιορισμό , χρησιμοποιούμε τη μέθοδο των πολλαπλασιαστών Lagrange. Ορίζουμε τη συνάρτηση Lagrange ως εξής:

όπου είναι ο πολλαπλασιαστής Lagrange.

Βήμα 2: Εύρεση των κρίσιμων σημείων

Για να βρούμε τα μέγιστα, παίρνουμε τις μερικές παραγώγους της ως προς και και τις θέτουμε ίσες με μηδέν.

Παράγωγος ως προς :

Διαιρώντας με το 2, έχουμε:

Αυτή η εξίσωση δείχνει ότι είναι ιδιοδιάνυσμα του πίνακα με ιδιοτιμή

Παράγωγος ως προς :

Αυτό επιβεβαιώνει τον περιορισμό ότι το διάνυσμα έχει μονάδα νόρμα.

Βήμα 3: Εύρεση της μέγιστης ιδιοτιμής

Το ποσό που μεγιστοποιούμε είναι:

Δεδομένου ότι ισούται με .

Επομένως, για να μεγιστοποιήσουμε το , πρέπει να επιλέξουμε το ως τη μέγιστη ιδιοτιμή του . Το αντίστοιχο ιδιοδιάνυσμα είναι το διάνυσμα που αναζητούμε.

**Εκτίμηση του L όταν δεν μας δίδεται**

Όταν δεν καθορίζεται εκ των προτέρων ο αριθμός των διαστάσεων L για τη μείωση των διαστάσεων μέσω της Ανάλυσης Κυρίων Συνιστωσών (PCA), μπορούμε να εκτιμήσουμε το L χρησιμοποιώντας διάφορες μεθόδους που βασίζονται στην εξήγηση της διασποράς των δεδομένων. Μια κοινή προσέγγιση είναι η **εξέταση του ποσοστού της συνολικής διασποράς που εξηγείται από τις πρώτες L κύριες συνιστώσες**. Η διαδικασία είναι η εξής:

1. **Υπολογισμός των Ιδιοτιμών:**
   * Εκτελούμε PCA στα δεδομένα και υπολογίζουμε τις ιδιοτιμές  του πίνακα συνδιασπορών S, ταξινομημένες κατά φθίνουσα σειρά.
2. **Καθορισμός του Κρίσιμου Ποσοστού Διασποράς:**
   * Επιλέγουμε ένα όριο για το ποσοστό της συνολικής διασποράς που θέλουμε να διατηρήσουμε, π.χ., 95%.
3. **Υπολογισμός της Αθροιστικής Διασποράς:**
   * Υπολογίζουμε την αθροιστική διασπορά και τη συγκρίνουμε τη συνολική διασπορά .
4. **Επιλογή του L:**
   * Επιλέγουμε το μικρότερο L τέτοιο ώστε το λόγο να είναι ίσο ή μεγαλύτερο από το προκαθορισμένο ποσοστό (π.χ., 95%).

**Αιτιολόγηση:**

Αυτή η μέθοδος διασφαλίζει ότι οι επιλεγμένες κύριες συνιστώσες διατηρούν το μεγαλύτερο δυνατό ποσοστό της αρχικής διασποράς των δεδομένων, μειώνοντας έτσι την απώλεια πληροφοριών. Επιπλέον, η επιλογή του L με βάση την αθροιστική διασπορά βοηθά στην αποφυγή του φαινομένου του overfitting και στη βελτίωση της γενίκευσης των μοντέλων που δημιουργούνται με τα μειωμένα δεδομένα.

**Αν τα δεδομένα δεν είναι μηδενικής μέσης τιμής**

Εάν τα δεδομένα **δεν** έχουν μηδενική μέση τιμή , είναι απαραίτητο να **κεντράρουμε** τα δεδομένα πριν από την εφαρμογή της Ανάλυσης Κυρίων Συνιστωσών (PCA).

1. **Υπολογισμός του Μέσου Διανύσματος:**
   * Υπολογίζουμε το μέσο διανύσμα των δεδομένων:

1. **Κεντράρισμα των Δεδομένων:**
   * Αφαιρούμε το μέσο διανύσμα από κάθε παρατήρηση:
   * Έτσι, τα νέα κεντραρισμένα δεδομένα έχουν μηδενική μέση τιμή:

**Αποδείξτε επίσης ότι η λύση του παραπάνω προβλήματος είναι το**

**ιδιοδιάνυσμα που αντιστοιχεί στην µέγιστη ιδιοτιμή (έστω ) του µητρώου S**

Για να αποδείξουμε ότι η λύση του προβλήματος βελτιστοποίησης

είναι το ιδιοδιάνυσμα που αντιστοιχεί στη μέγιστη ιδιοτιμή του πίνακα S, ακολουθούμε τα εξής βήματα:

**Βήμα 1: Διατύπωση του Προβλήματος με τη Μέθοδο των Πολλαπλασιαστών Lagrange**

Ορίζουμε τη μέθοδο Lagrange για το πρόβλημα βελτιστοποίησης ως εξής:

όπου:

* είναι ο πολλαπλασιαστής Lagrange.
* είναι το διάνυσμα προς βελτιστοποίηση.

**Βήμα 2: Εύρεση των Κρίσιμων Σημείων μέσω Παραγώγων**

Για να βρούμε τα ακρότατα, υπολογίζουμε τις μερικές παραγώγους της ως προς και και τις θέτουμε ίσες με το μηδέν.

1. **Παράγωγος ως προς :**

Απλοποιώντας:

Αυτή η εξίσωση δείχνει ότι είναι **ιδιοδιάνυσμα** του πίνακα S με αντίστοιχη **ιδιοτιμή** .

1. **Παράγωγος ως προς :**

Αυτό επιβεβαιώνει τον περιορισμό ότι το διάνυσμα έχει **μοναδιαία νόρμα**.

**Βήμα 3: Επίλυση της Εξίσωσης**

Η εξίσωση είναι η τυπική μορφή για εύρεση ιδιοδιανυσμάτων και ιδιοτιμών ενός πίνακα. Συνεπώς:

* **Ιδιοδιανύσματα ():** Διάνυσμα που ικανοποιεί την εξίσωση .
* **Ιδιοτιμές ():** Κλίμακες που συνδέουν τα ιδιοδιανύσματα με τον πίνακα S.

**Βήμα 4: Επιλογή της Μέγιστης Ιδιοτιμής**

Το πρόβλημα βελτιστοποίησης στοχεύει στο να **μεγιστοποιήσει** το υπό τον περιορισμό . Παρατηρούμε ότι:

Δεδομένου ότι , το ισούται με . Για να μεγιστοποιήσουμε , πρέπει να επιλέξουμε την **μεγαλύτερη ιδιοτιμή** του πίνακα S, δηλαδή .

Συνεπώς, η μέγιστη τιμή του επιτυγχάνεται όταν είναι το ιδιοδιάνυσμα που αντιστοιχεί στη μέγιστη ιδιοτιμή .

**Βήμα 5: Ιδιότητες του Πίνακα Συνδιασπορών S**

**Παρατήρηση:** Ο πίνακας συνδιασπορών είναι εξ ορισμού ένα **μη αρνητικά ορισμένο** (positive semi-definite) μητρώο. Αυτό σημαίνει ότι όλες οι ιδιοτιμές του S είναι **μη αρνητικές** ().

**Απόδειξη της Μη Αρνητικότητας:**

Για οποιοδήποτε μη μηδενικό διάνυσμα :

Επομένως, ο πίνακας S είναι **μη αρνητικά ορισμένος**.

**Ας υποθέσουμε τώρα ότι θέλουμε να λύσουμε το πρόβλημα για L=2.  
Σε αυτή την περίπτωση, αποδείξτε ότι η επιθυμητή λύση προκύπτει από τη λύση του ακόλουθου προβλήματος βελτιστοποίησης με περιορισμούς:**

**Υπό τους περιορισμούς:**

**όπου είναι η -στάθμη του διανύσματος z, και είναι ο τελεστής εσωτερικού γινομένου.**

**Βήμα 1: Διατύπωση του Προβλήματος με τη Μέθοδο Lagrange**

Για να βρούμε το μέγιστο του υπό τους περιορισμούς ∣∣ και , χρησιμοποιούμε τη μέθοδο των πολλαπλασιαστών Lagrange. Ορίζουμε τον Lagrange ως εξής:

όπου:

* είναι ο πολλαπλασιαστής Lagrange για τον περιορισμό της μονάδας νόρμας.
* είναι ο πολλαπλασιαστής Lagrange για τον περιορισμό της ορθογωνιότητας με τη πρώτη κύρια συνιστώσα .

**Βήμα 2: Εύρεση των Κρίσιμων Σημείων μέσω Παραγώγων**

Για να βρούμε τα ακρότατα, υπολογίζουμε τις μερικές παραγώγους της ως προς , και και τις θέτουμε ίσες με το μηδέν.

1. **Παράγωγος ως προς :**

Απλοποιώντας:

1. **Παράγωγος ως προς :**
2. **Παράγωγος ως προς :**

**Βήμα 3: Επίλυση της Εξίσωσης**

Από την εξίσωση , μπορούμε να απομονώσουμε το :

Αυτή η εξίσωση δείχνει ότι συνδέεται με τη Πρώτη κύρια συνιστώσα μέσω της δεύτερης ιδιοτιμής .

**Βήμα 4: Χρήση της Ορθογωνιότητας με τη Πρώτη κύρια συνιστώσα**

Δεδομένου ότι πρέπει να είναι ορθογώνιο με τη Πρώτη κύρια συνιστώσα , μπορούμε να αξιοποιήσουμε αυτή την ιδιότητα:

Επίσης, γνωρίζουμε ότι τα κύριες συνιστώσες είναι ιδιοδιανύσματα του πίνακα συνδιασπορών SS και ότι ο πίνακας S είναι συμμετρικός και μη αρνητικά ορισμένος.

**Βήμα 5: Επιλογή της Δεύτερης Μεγαλύτερης Ιδιοτιμής**

Για να μεγιστοποιήσουμε υπό τους περιορισμούς και , πρέπει να επιλέξουμε το ως το ιδιοδιάνυσμα του S που έχει τη δεύτερη μεγαλύτερη ιδιοτιμή .

**Δικαιολόγηση:**

* Με τη Πρώτη κύρια συνιστώσα έχει ήδη επιλεγεί ως το ιδιοδιάνυσμα που αντιστοιχεί στη μέγιστη ιδιοτιμή .
* Για να βρούμε τη δεύτερη κύρια συνιστώσα , πρέπει να μεγιστοποιήσουμε τη διακύμανση υπό τον περιορισμό ότι είναι ορθογώνιο σε .
* Αυτό οδηγεί στο συμπέρασμα ότι το πρέπει να είναι το ιδιοδιάνυσμα που αντιστοιχεί στη δεύτερη μεγαλύτερη ιδιοτιμή του S.

**Βήμα 6: Ιδιότητες του Πίνακα Συνδιασπορών S**

**Παρατήρηση:** Ο πίνακας συνδιασπορών είναι εξ ορισμού ένα **μη αρνητικά ορισμένο** (positive semi-definite) μητρώο. Αυτό σημαίνει ότι όλες οι ιδιοτιμές του S είναι **μη αρνητικές**

().

**Απόδειξη της Μη Αρνητικότητας:**

Για οποιοδήποτε μη μηδενικό διάνυσμα

Επομένως, ο πίνακας S είναι **μη αρνητικά ορισμένος**.

**Σημαντικές Παρατηρήσεις:**

1. Οι κύριες συνιστώσες είναι αμοιβαία ορθογώνια (ορθοκανονικά) μεταξύ τους, πράγμα που εξασφαλίζει ότι δεν υπάρχει επικάλυψη στην πληροφορία που μεταφέρονται.
2. Εφόσον ο πίνακας S είναι μη αρνητικά ορισμένος, οι ιδιοτιμές του είναι μη αρνητικές, και έτσι η μέγιστη διακύμανση είναι πάντα μη αρνητική.
3. Η ίδια προσέγγιση μπορεί να εφαρμοστεί για να βρεθούν περισσότερα κύριες συνιστώσες(L > 2), πάντα εξασφαλίζοντας την ορθογωνιότητα με τις προηγούμενες κύριες συνιστώσες.

**Αποδείξτε επίσης ότι η λύση του παραπάνω προβλήματος είναι το**

**ιδιοδιάνυσµα που αντιστοιχεί στην δεύτερη µεγαλύτερη ιδιοτιµή (έστω ) του µητρώου S.**

**Χρησιμοποιώντας την αρχή της επαγωγής, διατυπώστε το τελικό**

**(L= L πρόβλημα βελτιστοποίησης και εξηγείστε αναλυτικά τις επιλογές σας.**

**Βήμα 1: Διατύπωση του Προβλήματος με τη Μέθοδο των Πολλαπλασιαστών Lagrange**

Όπως και για L = 1, θέλουμε να μεγιστοποιήσουμε τη διακύμανση υπό τους περιορισμούς:

1. **Μονάδα Νόρμας:**

1. **Ορθογωνιότητα με τη Πρώτη κύρια συνιστώσα:**

Χρησιμοποιούμε τη μέθοδο των πολλαπλασιαστών Lagrange για να συμπεριλάβουμε αυτούς τους περιορισμούς στη συνάρτηση βελτιστοποίησης. Ορίζουμε τον Lagrange ως:

όπου:

* είναι ο πολλαπλασιαστής Lagrange για τον περιορισμό της μονάδας νόρμας.
* είναι ο πολλαπλασιαστής Lagrange για τον περιορισμό της ορθογωνιότητας με τη Πρώτη κύρια συνιστώσα .

**Βήμα 2: Εύρεση των Κρίσιμων Σημείων μέσω Παραγώγων**

Για να βρούμε τα ακρότατα, υπολογίζουμε τις μερικές παραγώγους της ως προς , και και τις θέτουμε ίσες με το μηδέν.

1. **Παράγωγος ως προς :**

Απλοποιώντας:

1. **Παράγωγος ως προς :**
2. **Παράγωγος ως προς :**

**Βήμα 3: Επίλυση της Εξίσωσης**

Από την εξίσωση:

μπορούμε να γράψουμε:

Όπου I είναι ο μοναδιαίος πίνακας. Παρατηρούμε ότι αυτή η εξίσωση συνδέει το με τη Πρώτη κύρια συνιστώσα .

**Βήμα 4: Χρήση της Ορθογωνιότητας με τη Πρώτη κύρια συνιστώσα**

Δεδομένου ότι πρέπει να είναι ορθογώνιο με το , δηλαδή:

και γνωρίζουμε ότι είναι το ιδιοδιάνυσμα που αντιστοιχεί στη μεγαλύτερη ιδιοτιμή , μπορούμε να συμπεράνουμε ότι πρέπει να είναι το ιδιοδιάνυσμα που αντιστοιχεί στη δεύτερη μεγαλύτερη ιδιοτιμή .

**Βήμα 5: Επιλογή της Δεύτερης Μεγαλύτερης Ιδιοτιμής**

Το πρόβλημα βελτιστοποίησης στοχεύει στο να **μεγιστοποιήσει** το υπό τους περιορισμούς και . Αυτό επιτυγχάνεται όταν είναι το ιδιοδιάνυσμα που αντιστοιχεί στη **δεύτερη μεγαλύτερη ιδιοτιμή** του πίνακα S.

**Aιτιολόγηση:**

* Με τη Πρώτη κύρια συνιστώσα έχει ήδη επιλεγεί ως το ιδιοδιάνυσμα που αντιστοιχεί στη μέγιστη ιδιοτιμή .
* Για να βρούμε τη δεύτερη κύρια συνιστώσα , πρέπει να μεγιστοποιήσουμε τη διακύμανση υπό τον περιορισμό ότι είναι ορθογώνιο προς .
* Αυτό οδηγεί στο συμπέρασμα ότι το πρέπει να είναι το ιδιοδιάνυσμα που αντιστοιχεί στη δεύτερη μεγαλύτερη ιδιοτιμή του S.

**Βήμα 6: Ιδιότητες του Πίνακα Συνδιασπορών S**

**Παρατήρηση:** Ο πίνακας συνδιασπορών είναι εξ ορισμού ένα **μη αρνητικά ορισμένο** (positive semi-definite) μητρώο. Αυτό σημαίνει ότι όλες οι ιδιοτιμές του S είναι **μη αρνητικές** για κάθε i).

**Απόδειξη της Μη Αρνητικότητας:**

Για οποιοδήποτε μη μηδενικό διάνυσμα :

Επομένως, ο πίνακας S είναι **μη αρνητικά ορισμένος**.

**Συμπέρασμα**

Αποδείξαμε ότι για L = 2, η λύση του προβλήματος βελτιστοποίησης:

είναι το **ιδιοδιάνυσμα** που αντιστοιχεί στη **δεύτερη μεγαλύτερη ιδιοτιμή** του πίνακα συνδιασπορών S.

**Γενίκευση με τη Μέθοδο της Επαγωγής για Αρχικό L**

Θέλουμε να δείξουμε ότι για κάθε L, η λύση του προβλήματος βελτιστοποίησης:

υπό τους περιορισμούς:

είναι το ιδιοδιάνυσμα που αντιστοιχεί στη L-οστη μεγαλύτερη ιδιοτιμή του πίνακα S.

**Βήμα 1: Βάση της Επαγωγής (L = 1)**

Η βάση της επαγωγής ήδη αποδείχθηκε από το προηγούμενο ερώτημα:

Για L = 1, η λύση είναι το ιδιοδιάνυσµα που αντιστοιχεί στη μεγαλύτερη ιδιοτιμή .

**Βήμα 2: Υπόθεση της Επαγωγής**

Υποθέτουμε ότι για κάποιο , η λύση του προβλήματος για L=k είναι το ιδιοδιάνυσμα που αντιστοιχεί στη k-οστη μεγαλύτερη ιδιοτιμή του S, υπό τους περιορισμούς:

**Βήμα 3: Απόδειξη για L = k + 1**

Θέλουμε να δείξουμε ότι η λύση του προβλήματος για L = k + 1 είναι το ιδιοδιάνυσμα που αντιστοιχεί στη (k+1)-οστη μεγαλύτερη ιδιοτιμή του πίνακα S, υπό τους περιορισμούς:

**Διατύπωση του Προβλήματος:**

όπου είναι οι πολλαπλασιαστές Lagrange για τους περιορισμούς ορθογωνιότητας με τις προηγούμενες κύριες συνιστώσες.

**Εύρεση των Κρίσιμων Σημείων:**

Παίρνοντας τις παραγώγους της ως προς και τους πολλαπλασιαστές Lagrange και θέτοντάς τες ίσες με το μηδέν:

1. **Παράγωγος ως προς :**

Απλοποιώντας:

1. **Παράγωγοι ως προς τους πολλαπλασιαστές Lagrange:**

Δηλαδή:

**Επίλυση της Εξίσωσης:**

Από την εξίσωση:

και γνωρίζοντας ότι τα είναι ιδιοδιανύσματα του S που αντιστοιχούν στις ιδιοτιμές , μπορούμε να συμπεράνουμε ότι πρέπει να είναι το ιδιοδιάνυσμα που αντιστοιχεί στη

(k+1)-οστη μεγαλύτερη ιδιοτιμή .

**Aιτιολόγηση:**

* Οι προηγούμενα κύριες συνιστώσες έχουν ήδη επιλεγεί ως τα ιδιοδιανύσματα που αντιστοιχούν στις μεγαλύτερες ιδιοτιμές
* Η προσθήκη του περιορισμού ορθογωνιότητας με όλα τα προηγούμενα κύριες συνιστώσες εξασφαλίζει ότι τη νέα κύρια συνιστώσα θα βρίσκεται στον υπόχωρο που είναι ορθογώνιος προς όλες τις προηγούμενες κύριες συνιστώσες.
* Αυτό οδηγεί στο συμπέρασμα ότι το πρέπει να είναι το ιδιοδιάνυσμα που αντιστοιχεί στη (k+1)-οστη μεγαλύτερη ιδιοτιμή .

**Τελική Διατύπωση του Προβλήματος για Γενικό L**

Για ένα γενικό L, το πρόβλημα βελτιστοποίησης διατυπώνεται ως εξής:

υπό τους περιορισμούς:

**Διατύπωση του Lagrange:**

**Επίλυση των Παραγώγων:**

Παίρνοντας τις παραγώγους της ως προς και τους πολλαπλασιαστές Lagrange και , και θέτοντάς τες ίσες με το μηδέν, καταλήγουμε σε:

και

Για ένα γενικό L, η διαδικασία επιλογής των LL κύριων συνιστωσών μέσω PCA διατυπώνεται ως εξής:

**Εκτέλεση PCA:**

Υπολογίζουμε τις ιδιοτιμές και τα αντίστοιχα ιδιοδιανύσματα του πίνακα συνδιασπορών S.

**Επιλογή των Κύριων Συνιστωσών:**

Επιλέγουμε τα πρώτα L ιδιοδιανύσματα που αντιστοιχούν στις L μεγαλύτερες ιδιοτιμές .

**Ορθογωνιότητα και Μονάδα Νόρμας:**

Διασφαλίζουμε ότι κάθε για έχει μονάδα νόρμα και είναι ορθογώνιο με όλα τα προηγούμενα για i < l.

**Δημιουργία Νέου Χώρου Συντεταγμένων:**

Οι L κύριες συνιστώσες σχηματίζουν ένα νέο υπόχωρο διάστασης L στον οποίο τα δεδομένα προβάλλονται, διατηρώντας τη μέγιστη δυνατή διασπορά και εξασφαλίζοντας τη γραμμική ασυσχέτιση μεταξύ των νέων διαστάσεων.

**Αποδείξτε ότι η Frobenius στάθµη ενός µητρώου, ισούται µε την l2 στάθµη του διανύσµατος που προκύπτει από τις γραµµές ή τις στήλες.**

**Δοθέν:**

* Έστω ένα μητρώο.
* Η **Frobenius στάθμη** του α ορίζεται ως:
* Η  **στάθμη** ενός διανύσματος με στοιχεία ορίζεται ως: όπου KK είναι το πλήθος των στοιχείων του διανύσματος.

**Βήμα 1: Διάνυσμα από τις Γραμμές του Μητρώου**

1. **Δημιουργία Διανύσματος από τις Γραμμές:**
   * Έστω ότι προβάλλουμε το μητρώο σε ένα διανύσμα τοποθετώντας τις γραμμές του διαδοχικά.
   * Δηλαδή, αν έχει γραμμές
   * Το διανύσμα έχει μήκος .
2. **Υπολογισμός της Στάθμης του Διανύσματος :**
   * Παρατηρούμε ότι το άθροισμα των τετραγώνων των στοιχείων του είναι το ίδιο με το άθροισμα των τετραγώνων των στοιχείων του , άρα:

**Βήμα 2: Διάνυσμα από τις Στήλες του Μητρώου**

1. **Δημιουργία Διανύσματος από τις Στήλες:**
   * Έστω ότι προβάλλουμε το μητρώο σε ένα διανύσμα τοποθετώντας τις στήλες του διαδοχικά.
   * Δηλαδή, αν έχει στήλες , τότε
   * Το διάνυσμα έχει επίσης μήκος .
2. **Υπολογισμός της Στάθμης του Διανύσματος :**

= =

Όπως και στο προηγούμενο βήμα, το άθροισμα των τετραγώνων των στοιχείων του ισούται με το άθροισμα των τετραγώνων των στοιχείων του , άρα:

**Επεξήγηση:**

* Οι γραμμές και οι στήλες ενός μητρώου περιέχουν τα ίδια στοιχεία, απλώς με διαφορετική διάταξη. Όταν τα στοιχίζουμε σε ένα διανύσμα, το συνολικό άθροισμα των τετραγώνων των στοιχείων παραμένει το ίδιο.
* Η Frobenius στάθμη είναι μια μέτρηση της συνολικής "μεγέθυνσης" ή "ενέργειας" ενός μητρώου, ανεξάρτητα από τη διάταξή του σε γραμμές ή στήλες. Η ισότητα με την στάθμη διανύσματος που προκύπτει από τις γραμμές ή τις στήλες αντικατοπτρίζει αυτή την ιδιότητα.
* Η Frobenius στάθμη χρησιμοποιείται συχνά στην PCA για την εκτίμηση της απόστασης μεταξύ του αρχικού μητρώου συνδιασπορών και της προσέγγισης χαμηλότερης κατάταξης, όπως φαίνεται στο πρόβλημα βελτιστοποίησης που αναφέρθηκε προηγουμένως.

**Αποδείξτε ότι η λύση του παραπάνω προβλήματος ελαχιστοποίησης είναι η ακόλουθη:**

**όπου:**

**\* : οι πρώτες L στήλες του μητρώου V, και**

**\* : ένα διαγώνιο μητρώο που περιέχει τις πρώτες L τιμές της διαγωνίου του μητρώου .**

Η απόδειξη βασίζεται στο **Θεώρημα του Eckart-Young–Mirsky**, το οποίο δηλώνει ότι η καλύτερη προσέγγιση χαμηλότερης κατάταξης (rank) ενός μητρώου σε σχέση με το Frobenius norm, επιτυγχάνεται μέσω της διατήρησης των μεγαλύτερων L ιδιαζουσών τιμών και των αντίστοιχων ιδιοδιανυσμάτων.

**Βήμα 1: Σύνδεση της Προσέγγισης Χαμηλότερης Κατάταξης με την SVD**

Έχουμε:

όπου είναι διαγώνιο με τις ιδιάζουσες τιμές

Η προσέγγιση χαμηλότερης κατάταξης L του δίνεται από την επιλογή των πρώτων L ιδιοτιμών και των αντίστοιχων ιδιοδιανυσμάτων. Δηλαδή:

όπου:

* περιέχει τις πρώτες L στήλες του ,
* είναι ένα διαγώνιο μητρώο με τις πρώτες L ιδιάζουσες τιμές .

**Βήμα 2: Εφαρμογή του Θεωρήματος του Eckart-Young–Mirsky**

Το θεώρημα αυτό λέει ότι για οποιοδήποτε μητρώο και οποιαδήποτε κατάταξη , η προσέγγιση χαμηλότερης κατάταξης που ελαχιστοποιεί το Frobenius norm είναι η προσέγγιση που περιλαμβάνει τις πρώτες L ιδιάζουσες τιμές και τα αντίστοιχα ιδιοδιανύσματα.

Στην περίπτωσή μας, αφού είναι συμμετρικό και μη αρνητικά ορισμένο, οι ιδιάζουσες τιμές είναι μη αρνητικές και τα ιδιοδιανύσματα είναι ορθογώνια.

**Βήμα 3: Υπολογισμός του Ελαχίστου Πόρου Frobenius**

Το Frobenius norm δίνεται από:

Επειδή περιλαμβάνει τις πρώτες L ιδιάζουσες τιμές και τα αντίστοιχα ιδιοδιανύσματα, το υπόλοιπο αποτελείται από τις υπόλοιπες ιδιαζουσές τιμές:

Εφόσον οι ιδιάζουσες τιμές μειώνονται φθίνουσα σειρά, αυτή η επιλογή ελαχιστοποιεί το άθροισμα των τετραγωνικών διαφορών.

**Βήμα 4: Συμπέρασμα**

Έτσι, αποδεικνύουμε ότι η λύση του προβλήματος:

είναι:

όπου και περιέχουν τις πρώτες L στήλες και τις αντίστοιχες ιδιάζουσες τιμές του πίνακα και αντίστοιχα.

**Επεξήγηση των Επιλογών:**

1. Η ΣVD διασπά το μητρώο σε ιδιοδιανύσματα και ιδιαζουσές τιμές, που επιτρέπουν την εύκολη ανάλυση της διασποράς των δεδομένων.
2. Η σύνδεση της συνδιασποράς με την SVD επιτρέπει την αναπαράσταση του ως , όπου περιέχει τις ιδιάζουσες τιμές.
3. **Θεώρημα του Eckart-Young–Mirsky:** Χρησιμοποιείται για να επιβεβαιώσει ότι η επιλογή των πρώτων L ιδιοτιμών και ιδιοδιανυσμάτων παρέχει την βέλτιστη χαμηλότερης κατάταξης προσέγγιση.
4. Η μείωση της κατάταξης μέσω της επιλογής των σημαντικότερων ιδιοδιανυσμάτων διατηρεί τη μέγιστη δυνατή διακύμανση στα δεδομένα, ενώ μειώνει την πολυπλοκότητα.

**Aποδείξτε ότι ισχύουν οι ακόλουθες σχέσεις:**

**και επομένως η σ.π. της τ.μ. θα είναι:**

**Δοθέντα:**

Η τυχαία μεταβλητή Z ακολουθεί την κανονική κατανομή (0, I), δηλαδή:

Η τυχαία μεταβλητή W (που αντιπροσωπεύει τον θόρυβο) ακολουθεί την κανονική κατανομή , δηλαδή:

Οι μεταβλητές Z και W είναι ανεξάρτητες.

Το μοντέλο γέννησης των δεδομένων δίνεται από τη σχέση:

όπου:

* W είναι το μητρώο βαρών,
* είναι το N-διάστατο διάνυσμα μετατόπισης,
* W (δεύτερος όρος) αντιπροσωπεύει τον θόρυβο.

**Σημείωση:** Υποθέτουμε ότι το δεύτερο W στη σχέση αντιπροσωπεύει τον θόρυβο και όχι το ίδιο το μητρώο βαρών. Για αποσαφήνιση, ας αναδιατυπώσουμε τη σχέση ως εξής:

όπου .

**ΒΗΜΑ 1. Υπολογισμός του Expectation :**

Ξεκινάμε τη σχέση γέννησης των δεδομένων:

Υπολογίζουμε το expectation:

Εφαρμόζουμε την γραμμική ιδιότητα της αναμενόμενης τιμής:

= + +

Γνωρίζουμε ότι:

* είναι σταθερό διάνυσμα, άρα

Επομένως:

Άρα:

**ΒΗΜΑ 2. Υπολογισμός της Συνδιασποράς :**

Από το προηγούμενο βήμα, γνωρίζουμε ότι , άρα:

Αντικαθιστούμε τη σχέση γέννησης των δεδομένων:

Άρα:

Επεκτείνουμε το γινόμενο:

Χρησιμοποιούμε την ανεξαρτησία μεταξύ Z και , καθώς και την ιδιότητα ότι και :

Επομένως:

Υπολογίζουμε κάθε όρο ξεχωριστά:

1. **Υπολογισμός του :**

Δεδομένου ότι(0, I):

Άρα:

1. **Υπολογισμός του :**

(λόγω ότι )

Συνεπώς:

Άρα:

**ΒΗΜΑ 3. Συμπέρασμα για τη Συνάρτηση Πυκνότητας Πιθανότητας (x):**

Από τις παραπάνω σχέσεις, γνωρίζουμε ότι:

* X είναι γραμμική συνάρτηση της Z και του θορύβου ,
* Z και είναι κανονικά κατανεμημένες και ανεξάρτητες.

Άρα, η τυχαία μεταβλητή X ακολουθεί την κανονική κατανομή:

όπου:

Συνεπώς, η συνάρτηση πυκνότητας πιθανότητας της τυχαίας μεταβλητής X είναι:

**Αποδείξτε ότι η παραπάνω υπό συνθήκη σ.π.τ., με δεδομένο αυτό της Σχέσης (4), δίνεται από την ακόλουθη σχέση:**

**όπου .**

Δοθέντα:

Μοντέλο Γέννησης Δεδομένων: όπου:

* (λανθάνουσα μεταβλητή με κανονική κατανομή).
* (θόρυβος με κανονική κατανομή).
* W είναι ένα μητρώο βαρών.
* είναι ένα N-διάστατο διάνυσμα μετατόπισης.

Θεώρημα του Bayes: όπου:

* (x|z) είναι η συνάρτηση πιθανοφάνειας (likelihood).
* (z) είναι η εκ των προτέρων συνάρτηση πυκνότητας πιθανότητας της Z.
* (x) είναι η συνάρτηση πυκνότητας πιθανότητας της X.

**Βήμα 1: Καθορισμός των Κατανομών**

1. Συνάρτηση Πιθανοφάνειας (x|z):

Από το μοντέλο γέννησης δεδομένων:

και γνωρίζοντας ότι , έχουμε:

Άρα:

(x|z) = ()

1. Εκ των Προτέρων Συνάρτηση Πυκνότητας Πιθανότητας :

Δεδομένου ότι :

1. Συνάρτηση Πυκνότητας Πιθανότητας :

Το X είναι γραμμική συνάρτηση της Z και του θορύβου , άρα:

όπου:

Άρα:

**Βήμα 2: Εφαρμογή του Θεωρήματος του Bayes**

Χρησιμοποιώντας το Θεώρημα του Bayes:

(z|x) =

Αντικαθιστώντας τις εκ των προτέρων σχέσεις:

(z|x) =

**Βήμα 3: Παραγωγή της Εκ των Υστέρων Κατανομής (z|x)**

Για να βρούμε τη μορφή της (z|x), παρατηρούμε ότι η συνάρτηση πυκνότητας πιθανότητας είναι προϊόν δύο Γκαουσιανών, το οποίο παράγει επίσης Γκαουσιανή κατανομή. Θα χρησιμοποιήσουμε τις ιδιότητες των Γκαουσιανών για να ολοκληρώσουμε την απόδειξη.

1. Γκαουσιανές:
2. Γινόμενο των Πυκνοτήτων:

Το γινόμενο των δύο Γκαουσιανών (x|z) και είναι:

(x|z) (z)

Αφού ο κανονικοποιημένος σταθερός πολλαπλασιαστής θα εξαλείψει το , επικεντρωνόμαστε στο εκθέτη.

1. Ολοκλήρωση του Γινόμενου στον Εκθέτη:

Επεκτείνοντας τους όρους:

Συγκεντρώνοντας τους όρους που περιέχουν το z:

1. Ολοκλήρωση του Γινόμενου σε Μορφή Γκαουσιανής:

Θέλουμε να γράψουμε το εκθέτη σε μορφή ολοκληρωμένης Γκαουσιανής, δηλαδή:

όπου και είναι οι εκ των υστέρων μέσες τιμές και συνδιασπορά.

Συγκρίνοντας τους όρους, έχουμε:

Άρα:

Επομένως, η εκ των υστέρων κατανομή (z|x) είναι:

όπου:

**Συμπέρασμα**

Έχουμε αποδείξει ότι η εκ των υστέρων συνάρτηση πυκνότητας πιθανότητας (z|x) είναι Γκαουσιανή με:

* Μέση Τιμή (Mean):

Παρατηρούμε ότι η μέση τιμή εξαρτάται από το x.

* Συνδιασπορά (Covariance):

Η συνδιασπορά είναι ανεξάρτητη από το x.

* Σημαντικές Παρατηρήσεις:

1. Ανάγκη για ορθογωνιοποίηση του :

Η συνδιασπορά είναι πάντα θετικά ορισμένη, δεδομένου ότι είναι θετικά ημι-ορισμένο και είναι θετικά ορισμένο.

1. Ορθοκανονικότητα του W:

Εάν το μητρώο W είναι ορθοκανονικό (δηλαδή ), τότε η συνδιασπορά γίνεται , απλοποιώντας την εκ των υστέρων κατανομή.

1. Εφαρμογές στην Πιθανοτική PCA:

Η πιθανοτική PCA επεκτείνει την κλασική PCA εισάγοντας μια πιθανοτική ερμηνεία, όπου οι κρυφές μεταβλητές Z λαμβάνουν υπόψη τη διασπορά των δεδομένων μέσω του θορύβου . Η εκ των υστέρων κατανομή επιτρέπει την εκτίμηση των κρυφών παραμέτρων Z δεδομένων νέων παρατηρήσεων X.

Έχουμε δείξει ότι:

όπου:

και η μέση τιμή της εκ των υστέρων κατανομής εξαρτάται από το x, ενώ η συνδιασπορά δεν εξαρτάται από το x.

Αυτή η σχέση επιβεβαιώνει ότι η πιθανοτική PCA παρέχει μια γραμμική εκτίμηση των κρυφών μεταβλητών Z που λαμβάνει υπόψη τη διασπορά των δεδομένων μέσω της κατανομής θορύβου , εξασφαλίζοντας παράλληλα την αποδοτική αναπαράσταση των δεδομένων στο κύριο υπόχωρο.

Παρατηρήστε ότι: με την ισότητα να ισχύει όταν (αποδείξτε το).

**Δοθέντα:**

Το Kullback-Leibler Divergence (KLD) μεταξύ δύο κατανομών πιθανοτήτων και ορίζεται ως:

Παρατηρούμε ότι:

με την ισότητα να ισχύει όταν .

Η απόδειξη βασίζεται στην **ανισότητα του Gibbs**, η οποία είναι μια ειδική περίπτωση της **ανισότητας του Jensen** για συναρτήσεις κυρτών.

**Βήμα 1: Χρήση της Ανισότητας του Jensen**

Η συνάρτηση είναι **κυρτή** για όλες τις u > 0, διότι η δεύτερη παράγωγός της είναι αρνητική .

Η ανισότητα του Jensen για μια κυρτή συνάρτηση f και μια τυχαία μεταβλητή UU με αναμενόμενη τιμή είναι:

Αλλά επειδή η είναι κυρτή, η ανισότητα του Jensen ισχύει στην αντίθετη κατεύθυνση:

**Βήμα 2: Εφαρμογή της Ανισότητας του Jensen στην KLD**

Ας ορίσουμε τη μεταβλητή . Εφόσον και είναι κατανομές πιθανότητας, έχουμε:

Χρησιμοποιώντας την ανισότητα του Jensen για την κυρτή συνάρτηση :

Ωστόσο, η KLD ορίζεται με την ένδειξη , οπότε η ανισότητα αυτή πρέπει να αναθεωρηθεί. Στην πραγματικότητα, η συνάρτηση είναι **συγκεντρωτική**, οπότε η σωστή εφαρμογή της ανισότητας του Jensen για την κυρτή συνάρτηση οδηγεί στην **ανισότητα του Gibbs**, η οποία δίνει:

**Βήμα 3: Απόδειξη της Μη Αρνητικότητας μέσω του Θεωρήματος του Gibbs**

Η ανισότητα του Gibbs δηλώνει ότι για οποιεσδήποτε δύο κατανομές πιθανοτήτων p(x) και q(x):

με την ισότητα να ισχύει **μόνο** όταν p(x) = q(x) για κάθε x.

Ας δούμε γιατί ισχύει αυτό:

**1.Θεμελιώδης Ιδιότητα της Συνάρτησης :**

Για κάθε u > 0, ισχύει ότι:

με ισότητα μόνο όταν u = 1.

**2.Εφαρμογή της Ιδιότητας στη KLD:**

Θέτοντας , έχουμε:

**3.Αναμενόμενη Τιμή:**

Παίρνοντας την αναμενόμενη τιμή με βάση :

Επειδή:

με ισότητα όταν για κάθε x, προκύπτει:

και η ισότητα ισχύει **μόνο** όταν για κάθε x.

**Συμπέρασμα**

Από τα παραπάνω βήματα, αποδεικνύουμε ότι:

και η ισότητα ισχύει **μόνο** όταν:

Αυτό επιβεβαιώνει ότι το Kullback-Leibler Divergence είναι ένα μέτρο που εκφράζει την απόσταση (ή τη διαφορά) μεταξύ δύο κατανομών πιθανοτήτων και ότι είναι πάντα μη αρνητικό, με το μηδέν να επιτυγχάνεται μόνο όταν οι δύο κατανομές είναι ταυτόσημες.

**Περισσότερες Παρατηρήσεις:**

1. Το Kullback-Leibler Divergence δεν είναι συμμετρικό, δηλαδή γενικά
2. Το KLD χρησιμοποιείται ευρέως στη μηχανική μάθηση, στατιστική, πληροφορική, και άλλα πεδία για την αξιολόγηση της ομοιότητας ή της διαφοράς μεταξύ δύο κατανομών πιθανοτήτων.
3. Στη μηχανική μάθηση, το KLD μπορεί να ερμηνευτεί ως ένα μέτρο της "απώλειας" ή της "πληροφορίας" που χάνεται όταν η κατανομή χρησιμοποιείται ως προσέγγιση της αληθινής κατανομής .

**PCA - ΔΙΑΔΙΚΑΣΙΑ 1 ΣΕΛΙΔΑ 5 ([mnist\_part1.py](https://github.com/nickpotamianos/Computer_Vision_Lab/blob/main/CV_5-AUTOENCODERS/mnist_part1.ipynb))**

**Υπολογισμός του Μέσου Ψηφίου**

Η συνάρτηση calculate\_mean\_digit υπολογίζει το μέσο ψηφίο για κάθε επιλεγμένο ψηφίο.

**def** **calculate\_mean\_digit**(X, y, digit):  
 digit\_mask = y == digit  
 digit\_data = X[digit\_mask]  
 **return** torch.mean(digit\_data, dim=0)

Αυτή η συνάρτηση φιλτράρει τα δεδομένα για το συγκεκριμένο ψηφίο και υπολογίζει τον μέσο όρο κάθε pixel, παράγοντας έτσι το μέσο ψηφίο.

**Υπολογισμός του Πίνακα Συνδιασπορών**

Η συνάρτηση calculate\_covariance\_matrix υπολογίζει τον πίνακα συνδιασπορών των δεδομένων.

**def** **calculate\_covariance\_matrix**(X):  
 X\_centered = X - torch.mean(X, dim=0)  
 N = X.shape[0]  
 **return** torch.matmul(X\_centered.T, X\_centered) / (N - 1)

Πρώτα, τα δεδομένα κεντράρονται αφαιρώντας τον μέσο όρο. Στη συνέχεια, υπολογίζεται ο πίνακας συνδιασπορών ως το γινόμενο των κεντραρισμένων δεδομένων με τον μετασχηματισμό τους και διαιρούμενο με τον αριθμό των παρατηρήσεων μείον ένα.

**Υπολογισμός των Κύριων Συνιστωσών**

Η κλάση PCA υλοποιεί την διαδικασία της PCA, υπολογίζοντας τις κύριες συνιστώσες και τις ιδιοτιμές.

**class** **PCA**:  
 **def** **\_\_init\_\_**(self, n\_components):  
 self.n\_components = n\_components  
 self.components = None  
 self.mean = None  
 self.eigenvalues = None  
 self.explained\_variance\_ratio\_ = None  
  
 **def** **fit**(self, X):  
 # Μετατροπή σε PyTorch tensor αν δεν είναι ήδη  
 **if** **not** isinstance(X, torch.Tensor):  
 X = torch.tensor(X, dtype=torch.float32)  
  
 # Κεντράρισμα των δεδομένων  
 self.mean = torch.mean(X, dim=0)  
 X\_centered = X - self.mean  
  
 # Υπολογισμός του πίνακα συνδιασπορών  
 N = X.shape[0]  
 cov\_matrix = torch.matmul(X\_centered.T, X\_centered) / (N - 1)  
  
 # Υπολογισμός ιδιοτιμών και ιδιοδιανυσμάτων  
 eigenvalues, eigenvectors = torch.linalg.eigh(cov\_matrix)  
  
 # Ταξινόμηση των ιδιοτιμών και αντίστοιχων ιδιοδιανυσμάτων κατά φθίνουσα σειρά  
 idx = torch.argsort(eigenvalues, descending=True)  
 eigenvalues = eigenvalues[idx]  
 eigenvectors = eigenvectors[:, idx]  
  
 # Αποθήκευση των πρώτων n\_components ιδιοδιανυσμάτων  
 self.components = eigenvectors[:, :self.n\_components]  
 self.eigenvalues = eigenvalues[:self.n\_components]  
  
 **return** self  
  
 **def** **transform**(self, X):  
 **if** **not** isinstance(X, torch.Tensor):  
 X = torch.tensor(X, dtype=torch.float32)  
 X\_centered = X - self.mean  
 **return** torch.matmul(X\_centered, self.components)  
  
 **def** **inverse\_transform**(self, X\_transformed):  
 **return** torch.matmul(X\_transformed, self.components.T) + self.mean

Η κλάση PCA εκτελεί τα εξής βήματα:

**Κεντράρισμα**: Αφαιρεί τον μέσο όρο από τα δεδομένα.

**Υπολογισμός Συνδιασπορών**: Υπολογίζει τον πίνακα συνδιασπορών.

**Ιδιοτιμές και Ιδιοδιανύσματα**: Υπολογίζει τις ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα συνδιασπορών.

**Ταξινόμηση**: Ταξινομεί τις ιδιοτιμές κατά φθίνουσα σειρά και επιλέγει τα αντίστοιχα ιδιοδιανύσματα.

**Μετασχηματισμός**: Μετασχηματίζει τα δεδομένα στον νέο χώρο συνιστωσών.

**Ανακατασκευή**: Ανακατασκευάζει τα δεδομένα από τον μειωμένο χώρο.

**ΑΠΟΤΕΛΕΣΜΑΤΑ:**

A close-up of a ring

Description automatically generated A collage of images of a ring

Description automatically generated A white circle with black background

Description automatically generated A graph of error

Description automatically generated A blurry image of a light

Description automatically generated A collage of images of x

Description automatically generated A blurry image of a white line

Description automatically generated

A graph of error

Description automatically generated

**Μέσο Ψηφίο:** Το μέσο ψηφίο αντικατοπτρίζει την τυπική μορφή του κάθε ψηφίου, παρουσιάζοντας τα βασικά χαρακτηριστικά που είναι κοινά σε όλες τις παρατηρήσεις.

**Πίνακας Συνδιασπορών:** Ο πίνακας συνδιασπορών δείχνει την διασπορά των δεδομένων και τις σχέσεις μεταξύ των διαφορετικών pixels. Οι υψηλές τιμές συνδιασπορών υποδηλώνουν ότι οι αντίστοιχες διαστάσεις συσχετίζονται ισχυρά μεταξύ τους.

**Κύριες Συνιστώσες:** Οι πρώτες οκτώ κύριες συνιστώσες απεικονίζουν τις πιο σημαντικές δομές των δεδομένων. Παρατηρείται ότι οι πρώτες συνιστώσες συχνά αντιπροσωπεύουν τις βασικές γραμμές και καμπύλες των ψηφίων, ενώ οι επόμενες συνιστώσες προσθέτουν λεπτομέρειες.

**Ανακατασκευές Ψηφίων:**

* L = 1: Η ανακατασκευή με μία συνιστώσα αποδίδει μια γενική μορφή του ψηφίου, αλλά χάνει πολλές λεπτομέρειες.
* L = 8: Η ποιότητα της ανακατασκευής βελτιώνεται σημαντικά, διατηρώντας τα βασικά χαρακτηριστικά του ψηφίου.
* L = 16, 64, 256: Με αύξηση του αριθμού των συνιστωσών, η ανακατασκευή πλησιάζει όλο και περισσότερο το αρχικό ψηφίο, διατηρώντας περισσότερες λεπτομέρειες και μικρότερα χαρακτηριστικά.

**Κατανομή Σφαλμάτων Ανακατασκευής:** Τα ιστογράμματα δείχνουν ότι τα σφάλματα ανακατασκευής μειώνονται με αύξηση του αριθμού των συνιστωσών.

**ΔΙΑΔΙΚΑΣΊΑ 2 ([mnist\_part2.py](https://github.com/nickpotamianos/Computer_Vision_Lab/blob/main/CV_5-AUTOENCODERS/mnist_part2.ipynb))**

**Αποτελέσματα και Σχολιασμός**

Κατά την εφαρμογή της PCA με L = 128 στο σύνολο δεδομένων εκπαίδευσης του MNIST, παρατηρήθηκαν τα ακόλουθα αποτελέσματα:

1. Ο πίνακας διαμορφώθηκε ως ένας πίνακας διαστάσεων (784, 128), όπου κάθε στήλη αντιστοιχεί σε μια κύρια συνιστώσα. Οι πρώτες στήλες του πίνακα αυτών των συνιστωσών αντιπροσωπεύουν τις πιο σημαντικές διαστάσεις που εξηγούν τη μεγαλύτερη διακύμανση στα δεδομένα.
2. Η ανακατασκευή των δεδομένων εκπαίδευσης από τον συμπιεσμένο χώρο διαστάσεων 128 αποδίδει υψηλή ποιότητα ανακατασκευής, με πολύ μικρότερο μέσο τετραγωνικό σφάλμα (MSE) σε σύγκριση με τις πιο συμπιεσμένες ανακατασκευές του προηγούμενου μέρους της εργασίας. Το MSE υπολογίστηκε ως περίπου 0.006, υποδηλώνοντας ότι τα δεδομένα ανακατασκευάστηκαν με υψηλή ακρίβεια.
3. Οι ανακατασκευές των δεδομένων ελέγχου επίσης παρουσίασαν υψηλή ποιότητα, με το μέσο τετραγωνικό σφάλμα να είναι παρόμοιο με αυτό των δεδομένων εκπαίδευσης. Τα χαρακτηριστικά ζευγάρια εικόνων (αρχική και ανακατασκευασμένη) αποδεικνύουν ότι οι κύριες συνιστώσες διατηρούν τα βασικά χαρακτηριστικά των ψηφίων, ενώ παράλληλα καταφέρνουν να αποθηκεύσουν αρκετές λεπτομέρειες για να διατηρούν την αναγνωσιμότητα των ψηφίων.
4. Η κατανομή των σφαλμάτων ανακατασκευής για διάφορους αριθμούς συνιστωσών (L = 1, 8, 16, 64, 256) έδειξε σαφή μείωση του σφάλματος με την αύξηση των συνιστωσών. Αυτό επιβεβαιώνει ότι περισσότερες συνιστώσες εξηγούν μεγαλύτερη διακύμανση στα δεδομένα, βελτιώνοντας την ποιότητα της ανακατασκευής.
5. **Παράδειγμα Ανακατασκευών**

Για το ψηφίο **0**, **4**, και **9**, παρουσιάστηκαν τα ακόλουθα ζευγάρια εικόνων:

* **Αρχική Εικόνα:** Το πραγματικό ψηφίο όπως εμφανίζεται στο σύνολο δεδομένων.
* **Συμπιεσμένη Εικόνα:** Το ψηφίο ανακατασκευασμένο από τις συμπιεσμένες συνιστώσες (L=128L = 128L=128).

**A comparison of a white and black image

Description automatically generated with medium confidenceA black and white image of a letter

Description automatically generated**

**A close-up of a test

Description automatically generated**

**Σχολιασμός:** Τα ανακατασκευασμένα ψηφία διατηρούν τα βασικά χαρακτηριστικά των αρχικών, όπως οι κύριες γραμμές και καμπύλες, ενώ παρατηρείται ελάχιστη απώλεια λεπτομερειών. Αυτό επιβεβαιώνει την αποτελεσματικότητα της PCA στην αποθήκευση της βασικής δομής των δεδομένων με μειωμένη διάσταση.

**KERNEL PCA - ΔΙΑΔΙΚΑΣΙΑ 1 ΣΕΛΙΔΑ 7 ([mnist\_kernel.py](https://github.com/nickpotamianos/Computer_Vision_Lab/blob/main/CV_5-AUTOENCODERS/mnist_kernel.ipynb))**

**Βασικές Διαφορές μεταξύ Κλασσικής PCA και Kernel PCA**

Η **κλασσική PCA** λειτουργεί με γραμμικούς μετασχηματισμούς, προσπαθώντας να βρει έναν νέο χώρο διαστάσεων όπου οι διαστάσεις είναι οι κύριες συνιστώσες που εξηγούν τη μέγιστη δυνατή διακύμανση των δεδομένων. Ωστόσο, αυτή η προσέγγιση περιορίζεται στην ικανότητά της να αποτυπώνει μόνο γραμμικές σχέσεις μεταξύ των χαρακτηριστικών.

Αντίθετα, η **Kernel PCA** επεκτείνει την κλασσική PCA χρησιμοποιώντας μη γραμμικούς μετασχηματισμούς μέσω της χρήσης πυρήνων (kernels). Αυτό επιτρέπει την εφαρμογή της PCA σε ένα υψηλότερο διαστατικό χώρο χαρακτηριστικών χωρίς την ανάγκη απευθείας υπολογισμού των μετασχηματισμένων δεδομένων, αξιοποιώντας το **θεώρημα του πυρήνα** (kernel trick). Με αυτόν τον τρόπο, η Kernel PCA μπορεί να αποτυπώσει πολύπλοκες, μη γραμμικές δομές στα δεδομένα, κάτι που η κλασσική PCA δεν μπορεί να κάνει.

**Υπόθεση Αναμενόμενης Τιμής και Επιπτώσεις της Μη Ικανοποίησης της**

Στην κλασσική PCA, υποθέτουμε ότι τα δεδομένα έχουν **μηδενική αναμενόμενη τιμή** (zero mean), δηλαδή ότι ο μέσος όρος κάθε χαρακτηριστικού είναι μηδέν. Αυτή η υπόθεση είναι κρίσιμη για τον σωστό υπολογισμό του πίνακα συνδιασπορών και των κύριων συνιστωσών.

Στην **Kernel PCA**, η υπόθεση αυτή εξακολουθεί να ισχύει για τα **μετασχηματισμένα δεδομένα** στον χώρο των χαρακτηριστικών. Αν αυτή η υπόθεση **δεν ικανοποιείται**, δηλαδή αν τα μετασχηματισμένα δεδομένα δεν έχουν μηδενική αναμενόμενη τιμή, οι υπολογισμοί του πίνακα συνδιασπορών θα είναι λανθασμένοι, επηρεάζοντας αρνητικά την απόδοση της PCA. Συγκεκριμένα, οι κύριες συνιστώσες μπορεί να μην αντανακλούν σωστά τη διακύμανση των δεδομένων, οδηγώντας σε ανεπαρκείς αναπαραστάσεις.

**Μη Γραμμικός Μετασχηματισμός**

Ο μη γραμμικός μετασχηματισμός εφαρμόζεται component-wise σε κάθε δεδομένο:

**def** **kernel\_transform**(X, alpha=0.1):  
 """  
 Εφαρμογή του μη γραμμικού μετασχηματισμού k(x) = exp(-x^2 / alpha)  
 σε κάθε στοιχείο του X.  
 X έχει σχήμα [N, M] (N παρατηρήσεις, M χαρακτηριστικά/pixels).  
 Επιστρέφει ένα tensor του ίδιου σχήματος.  
 """  
 **return** torch.exp(-(X \*\* 2) / alpha)

Ο μετασχηματισμός k(x) εφαρμόζεται σε κάθε pixel του κάθε ψηφίου, μετατρέποντας τα δεδομένα σε έναν νέο, μη γραμμικό χώρο χαρακτηριστικών.

**Υπολογισμός του Πίνακα Συνδιασπορών στον Χώρο των Χαρακτηριστικών**

**def** **calculate\_covariance\_matrix\_kspace**(X\_k):  
 """  
 Υπολογισμός του πίνακα συνδιασπορών στον χώρο των χαρακτηριστικών μετά τον μετασχηματισμό.  
 """  
 Xk\_centered = X\_k - X\_k.mean(dim=0)  
 N = X\_k.shape[0]  
 **return** torch.matmul(Xk\_centered.T, Xk\_centered) / (N - 1)

Μετά τον μετασχηματισμό, τα δεδομένα κεντράρονται αφαιρώντας τον μέσο όρο κάθε χαρακτηριστικού. Στη συνέχεια, υπολογίζεται ο πίνακας συνδιασπορών ​ ως το γινόμενο των κεντραρισμένων δεδομένων με τον μετασχηματισμό τους, διαιρούμενο με τον αριθμό των παρατηρήσεων μείον ένα. A close up of a circle

Description automatically generated

A blurry image of a person's face

Description automatically generated

**AUTOENCODERS - ΔΙΑΔΙΚΑΣΙΑ 1 ΣΕΛΙΔΑ 9 (**[**autoencoder\_mnist\_1.py**](https://github.com/nickpotamianos/Computer_Vision_Lab/blob/main/CV_5-AUTOENCODERS/autoencoder_mnist_1.ipynb)**)**

Οι **Autoencoders (AE)** είναι νευρωνικά δίκτυα που εκπαιδεύονται για να **αντιγράφουν** την είσοδό τους στην έξοδο, με σκοπό να μάθουν μια αποδοτική, συμπιεσμένη αναπαράσταση των δεδομένων εισόδου. Αποτελούνται από δύο βασικά μέρη: τον **κωδικοποιητή** (encoder) που μετασχηματίζει τα δεδομένα εισόδου σε έναν χαμηλότερης διάστασης χώρο (latent space) και τον **αποκωδικοποιητή** (decoder) που ανακατασκευάζει τα αρχικά δεδομένα από αυτήν την αναπαράσταση.

Ορισμός του Autoencoder

# Autoencoder Model  
**class** **Autoencoder**(nn.Module):  
 **def** **\_\_init\_\_**(self, input\_size=784, latent\_size=128):  
 super(Autoencoder, self).\_\_init\_\_()  
  
 # Encoder  
 self.encoder = nn.Linear(input\_size, latent\_size, bias=False)  
  
 # Decoder  
 self.decoder = nn.Linear(latent\_size, input\_size, bias=False)  
  
 # Initialize weights  
 nn.init.xavier\_uniform\_(self.encoder.weight)  
 nn.init.xavier\_uniform\_(self.decoder.weight)  
  
 **def** **forward**(self, x):  
 # Flatten input  
 x = x.view(x.size(0), -1)  
  
 # Encode  
 encoded = self.encoder(x)  
  
 # Decode  
 decoded = torch.sigmoid(self.decoder(encoded))  
  
 **return** decoded

**Encoder**: Μετατρέπει τα δεδομένα εισόδου από διαστάσεις 784 σε διαστάσεις 128 χωρίς bias.

**Decoder**: Ανακατασκευάζει τα δεδομένα από τις συμπιεσμένες διαστάσεις 128 πίσω στις αρχικές 784 διαστάσεις, χρησιμοποιώντας sigmoid ενεργοποίηση για την έξοδο.

Η αρχικοποίηση των βαρών γίνεται με τη χρήση της τεχνικής Xavier Uniform, η οποία βοηθά στην αποφυγή προβλημάτων όπως το vanishing ή exploding gradients κατά την εκπαίδευση.

**Εκπαίδευση του Autoencoder**

**def** **train\_model**(model, train\_loader, num\_epochs=40, learning\_rate=0.001):  
 device = torch.device("cuda" **if** torch.cuda.is\_available() **else** "cpu")  
 print(f"Using device: {device}")  
 model = model.to(device)  
  
 criterion = nn.BCELoss()  
 optimizer = optim.Adam(model.parameters(), lr=learning\_rate)  
  
 # Lists to store losses  
 train\_losses = []  
  
 # Training loop  
 **for** epoch **in** range(num\_epochs):  
 model.train()  
 running\_loss = 0.0  
  
 **for** data, \_ **in** train\_loader:  
 data = data.to(device)  
  
 # Forward pass  
 output = model(data)  
 loss = criterion(output, data.view(data.size(0), -1))  
  
 # Backward pass and optimize  
 optimizer.zero\_grad()  
 loss.backward()  
 optimizer.step()  
  
 running\_loss += loss.item()  
  
 epoch\_loss = running\_loss / len(train\_loader)  
 train\_losses.append(epoch\_loss)  
  
 # Print epoch statistics  
 print(f'Epoch [{epoch + 1}/{num\_epochs}], Loss: {epoch\_loss:.6f}')  
  
 # Compare encoder weights with PCA matrix  
 **if** epoch % 10 == 0:  
 compare\_weights\_with\_pca(model.encoder.weight.data)  
  
 # Plot training loss  
 plt.figure(figsize=(10, 5))  
 plt.plot(train\_losses)  
 plt.title('Training Loss Over Time')  
 plt.xlabel('Epoch')  
 plt.ylabel('Loss')  
 plt.grid(True)  
 plt.show()

Η συνάρτηση train\_model εκπαιδεύει τον Autoencoder για 40 εποχές χρησιμοποιώντας τον Adam optimizer με learning rate 0.001 και τη δυαδική εντροπία ως συνάρτηση κόστους. Σε κάθε εποχή, υπολογίζεται το μέσο τετραγωνικό σφάλμα (loss) και αποθηκεύεται για την παρακολούθηση της διαδικασίας εκπαίδευσης.

Επιπλέον, κάθε 10 εποχές, καλείται η συνάρτηση compare\_weights\_with\_pca για να συγκρίνει τα βάρη του κωδικοποιητή με το μητρώο ​ που προέκυψε από την προηγούμενη ανάλυση PCA. Αυτή η σύγκριση βοηθά στην αξιολόγηση του κατά πόσο ο Autoencoder μαθαίνει τα ίδια χαρακτηριστικά με την PCA.

**Σύγκριση Βαρών Κωδικοποιητή με Μητρώο PCA**

**def** **compare\_weights\_with\_pca**(encoder\_weights):  
 # Here you would compare with VL matrix from PCA  
 print(f"Encoder weights shape: {encoder\_weights.shape}")

Η συνάρτηση compare\_weights\_with\_pca έχει ως σκοπό να συγκρίνει τα βάρη του κωδικοποιητή με το μητρώο VLVL​ από την PCA. Στον παρόντα κώδικα, η σύγκριση περιορίζεται στην εκτύπωση του σχήματος των βαρών.

Κατά την εκπαίδευση του Autoencoder με τα καθορισμένα υπερπαραμέτρους (40 εποχές, latent size=128, batch size=250) και χρήση της δυαδικής εντροπίας ως συνάρτηση κόστους, παρατηρήθηκαν τα εξής:

**Μείωση του Loss:**

Κατά τη διάρκεια των 40 εποχών, το loss παρουσίασε συνεχή μείωση, επιβεβαιώνοντας ότι το μοντέλο μαθαίνει να ανακατασκευάζει τα δεδομένα εισόδου με αυξανόμενη ακρίβεια.

**Σύγκριση Βαρών Κωδικοποιητή με PCA ​:**

Μετά από κάθε 10 εποχές, τα βάρη του κωδικοποιητή συγκρίνονται με το μητρώο ​ από την PCA. Αν και στον παρόντα κώδικα η σύγκριση περιορίζεται στην εκτύπωση των σχημάτων, η πλήρης σύγκριση απαιτεί την εφαρμογή μιας μετρικής ομοιότητας.

Παράδειγμα: Αν τα βάρη του κωδικοποιητή πλησιάζουν τις κύριες συνιστώσες της PCA, αυτό υποδηλώνει ότι ο Autoencoder μαθαίνει τα πιο σημαντικά χαρακτηριστικά των δεδομένων, όπως προκύπτει από την PCA.

**Ανακατασκευασμένες Εικόνες:**

Οι ανακατασκευασμένες εικόνες παρουσιάζουν υψηλή ποιότητα, με το Autoencoder να καταφέρνει να αναπαράγει τα βασικά χαρακτηριστικά των ψηφίων, διατηρώντας παράλληλα την αναγνωσιμότητά τους.

Παράδειγμα: Το ανακατασκευασμένο ψηφίο "0" διατηρεί την κυκλική του μορφή, ενώ το ψηφίο "1" παραμένει λεπτό και ευθύγραμμο, όπως το αρχικό.

**Ιδιαιτερότητα του Autoencoder για Προσέγγιση της PCA:**

Λειτουργικές Ιδιότητες:

Γραμμικότητα: Ο κωδικοποιητής και ο αποκωδικοποιητής είναι γραμμικά επίπεδα χωρίς μη γραμμικές συναρτήσεις ενεργοποίησης, γεγονός που επιτρέπει στο Autoencoder να μαθαίνει γραμμικές σχέσεις μεταξύ των χαρακτηριστικών, όπως η PCA.

Undercomplete Κατασκευή: Με latent size μικρότερο από το αρχικό μέγεθος των δεδομένων, ο Autoencoder αναγκάζεται να μαθαίνει τις πιο σημαντικές διαστάσεις των δεδομένων, αποθηκεύοντας την πλειοψηφία της διακύμανσης τους.

Χωρίς Bias: Η απουσία bias στα επίπεδα βοηθά στη διατήρηση της μηδενικής μέσης τιμής των δεδομένων, όπως απαιτείται για την ορθή εκτέλεση της PCA.

A graph with a line

Description automatically generated

A group of white letters in black squares

Description automatically generated

**AUTOENCODERS - ΔΙΑΔΙΚΑΣΙΑ 2 ΣΕΛΙΔΑ 9 (**[**autoencoder\_mnist\_2.py**](https://github.com/nickpotamianos/Computer_Vision_Lab/blob/main/CV_5-AUTOENCODERS/autoencoder_mnist_2.ipynb)**)**

Η υλοποίηση βασίστηκε στην κλάση ThreeLayerAutoencoder, όπου:

Ο **κωδικοποιητής** έχει τα επίπεδα: 784 → 512 (ReLU) → 256 (ReLU) → 128.

Ο **αποκωδικοποιητής** αναστρέφει τη διαδικασία: 128 → 256 (ReLU) → 512 (ReLU) → 784 (Sigmoid).

Οι πίνακες βαρών των γραμμικών επιπέδων δεν χρησιμοποιούν bias. Με βάση τις διαστάσεις αυτές, το σύνολο των εκπαιδεύσιμων παραμέτρων υπολογίζεται από τη συνάρτηση count\_parameters(). Στο παρόν παράδειγμα, το πλήθος των παραμέτρων ήταν 1.130.496. Αυτό προκύπτει από το άθροισμα των μηκών των βαρών σε κάθε γραμμικό επίπεδο του δικτύου:

Encoder: 784×512 + 512×256 + 256×128

Decoder: 128×256 + 256×512 + 512×784

Καθώς δεν χρησιμοποιούμε bias, οι μόνοι παράμετροι είναι τα συνδεδεμένα βάρη.

A graph of a training loss

Description automatically generated

**AUTOENCODERS - ΔΙΑΔΙΚΑΣΙΑ 3 ΣΕΛΙΔΑ 9 ([autoencoder\_mnist\_3.py](https://github.com/nickpotamianos/Computer_Vision_Lab/blob/main/CV_5-AUTOENCODERS/autoencoder_mnist_3.ipynb))**

**(αʹ) Σχέση Αποκωδικοποιητή με τον Κωδικοποιητή:**

Σε έναν Autoencoder με tied weights (όπου μοιράζονται οι παράμετροι του κωδικοποιητή και του αποκωδικοποιητή), η αρχιτεκτονική προϋποθέτει ότι ο αποκωδικοποιητής δεν έχει ανεξάρτητα βάρη. Αντ’ αυτού, κάθε γραμμικό επίπεδο στον αποκωδικοποιητή χρησιμοποιεί τα βάρη του αντίστοιχου επιπέδου του κωδικοποιητή, αλλά με μετασχηματισμό transpose. Δηλαδή:

Για κάθε στρώμα στον αποκωδικοποιητή, τα βάρη είναι το transpose των βαρών του αντίστοιχου στρώματος στον κωδικοποιητή.

Αυτό σημαίνει ότι ο αποκωδικοποιητής ανακτά τις τιμές αναστροφής των γραμμικών μετασχηματισμών που έλαβε ο κωδικοποιητής, ακολουθώντας τη λογική της PCA, όπου η ανακατασκευή γίνεται με χρήση των transposed ιδιοδιανυσμάτων (κύριες συνιστώσες). Η χρήση των tied weights μειώνει σημαντικά τον αριθμό των παραμέτρων του δικτύου, καθώς δεν αποθηκεύονται ξεχωριστά βάρη για τον αποκωδικοποιητή.

**(ϐʹ) Επιλογή Συναρτήσεων Ενεργοποίησης στον Αποκωδικοποιητή:**

Αν στον κωδικοποιητή χρησιμοποιήσουμε LeakyReLU με κλίση 0.2, τότε για να προσεγγίσουμε την αναστροφή της λειτουργίας αυτής στον αποκωδικοποιητή, πρέπει να χρησιμοποιήσουμε μια συνάρτηση ενεργοποίησης που να αντιστρέφει ή να “αντιμετωπίζει ανάποδα” τη συμπεριφορά του LeakyReLU.

Στην περίπτωση του παρεχόμενου κώδικα για τον TiedAutoencoder, η προσέγγιση είναι ως εξής:

Στον κωδικοποιητή: Χρησιμοποιείται LeakyReLU με κλίση 0.2.

Στον αποκωδικοποιητή: Ορίζεται μια συνάρτηση decoder\_activation, η οποία λειτουργεί ως “αντίστροφη” του LeakyReLU:

self.decoder\_activation = **lambda** x: torch.where(x < 0, x / 0.2, x)

Αυτή η συνάρτηση ενεργοποίησης εφαρμόζεται μετά από κάθε γραμμικό επίπεδο του αποκωδικοποιητή, εκτός από την τελική έξοδο που χρησιμοποιεί Sigmoid.

A graph with a line

Description automatically generated

A black square with white text

Description automatically generated

**AUTOENCODERS - ΔΙΑΔΙΚΑΣΙΑ 4 ΣΕΛΙΔΑ 10 (**[**autoencoder\_mnist\_4.py**](https://github.com/nickpotamianos/Computer_Vision_Lab/blob/main/CV_5-AUTOENCODERS/autoencoder_mnist_4.ipynb)**)**

Ο κώδικας που παρέχεται υλοποιεί έναν PseudoinverseAutoencoder, όπου ο κωδικοποιητής αποτελείται από τρία γραμμικά επίπεδα με LeakyReLU ενεργοποίηση. Ο αποκωδικοποιητής δεν αποθηκεύει ξεχωριστά βάρη. Αντ’ αυτού, κατά τη διάρκεια της προώθησης (forward pass), χρησιμοποιεί τη συνάρτηση torch.pinverse() για να υπολογίσει τον Moore-Penrose ψευδοαντίστροφο των βαρών κάθε επιπέδου του κωδικοποιητή. Αυτή η προσέγγιση επιτρέπει στον αποκωδικοποιητή να ανακτά τις τιμές αναστροφής των γραμμικών μετασχηματισμών που έλαβε ο κωδικοποιητής, ακολουθώντας τη λογική της PCA όπου η ανακατασκευή γίνεται με χρήση των transposed ιδιοδιανυσμάτων (κύριες συνιστώσες). Η χρήση των ψευδοαντιστρόφων βαρών μειώνει σημαντικά τον αριθμό των παραμέτρων του δικτύου, καθώς δεν αποθηκεύονται ξεχωριστά βάρη για τον αποκωδικοποιητή, ενώ παράλληλα προσφέρει αυξημένη σταθερότητα και καλύτερη ανακατασκευή σε περιπτώσεις όπου τα βάρη του κωδικοποιητή δεν είναι τετραγωνικά ή δεν έχουν πλήρη βαθμίδα.

1. **Συνάρτηση Forward στον PseudoinverseAutoencoder**

Στον forward method, κάθε επίπεδο αποκωδικοποίησης υπολογίζει το αποτέλεσμα της γραμμικής μεταβολής με χρήση του ψευδοαντίστροφου των βαρών του αντίστοιχου επιπέδου κωδικοποιητή. Οι ενεργοποιήσεις decoder\_activation εφαρμόζονται μετά από κάθε γραμμικό επίπεδο του αποκωδικοποιητή, εκτός από την τελική έξοδο που χρησιμοποιεί Sigmoid. Αυτό διασφαλίζει ότι οι μη γραμμικές λειτουργίες του κωδικοποιητή αντιστρέφονται σωστά στον αποκωδικοποιητή, υποστηρίζοντας την προσέγγιση της PCA μέσω της μείωσης των παραμέτρων και της διατήρησης γραμμικών σχέσεων όσο το δυνατόν πιο πιστά.

1. **Διαφορές και Επιδράσεις της Χρήσης Ψευδοαντιστροφών**

Η χρήση του ψευδοαντίστροφου επιτρέπει στον Autoencoder να χρησιμοποιεί αποτελεσματικά την πληροφορία του κωδικοποιητή για την ανακατασκευή των δεδομένων χωρίς να χρειάζεται ξεχωριστά βάρη αποκωδικοποιητή. Αυτό μειώνει το συνολικό αριθμό των παραμέτρων του δικτύου, καθώς ο αποκωδικοποιητής δεν έχει ανεξάρτητα βάρη, αλλά χρησιμοποιεί τα ψευδοαντίστροφα των βαρών του κωδικοποιητή. Επιπλέον, η χρήση του Moore-Penrose ψευδοαντίστροφου προσφέρει μεγαλύτερη σταθερότητα και ακρίβεια στην ανακατασκευή, ειδικά σε περιπτώσεις όπου οι πίνακες βαρών δεν είναι τετραγωνικοί ή δεν έχουν πλήρη βαθμίδα. Αυτή η μέθοδος χειρίζεται καλύτερα τις περιπτώσεις όπου ο κωδικοποιητής δεν μπορεί να μετασχηματίσει τα δεδομένα σε έναν πλήρως βαθμωτό χώρο, επιτρέποντας μια πιο αξιόπιστη αναστροφή και ανακατασκευή των δεδομένων.

A graph with a line

Description automatically generated

A black square with white text

Description automatically generated with medium confidence

**AUTOENCODERS - ΔΙΑΔΙΚΑΣΙΑ 5 ΣΕΛΙΔΑ 10 (**[**compare\_autoencoder.py**](https://github.com/nickpotamianos/Computer_Vision_Lab/blob/main/CV_5-AUTOENCODERS/python%20compare_autoencoders.ipynb)**)**

Comparing autoencoder variants...  
  
Training Three Layer AE...  
Epoch [10/40], Loss: 0.073688  
Epoch [20/40], Loss: 0.069300  
Epoch [30/40], Loss: 0.067676  
Epoch [40/40], Loss: 0.066718  
  
Training Tied Weights AE...  
Epoch [10/40], Loss: 0.070830  
Epoch [20/40], Loss: 0.068180  
Epoch [30/40], Loss: 0.067308  
Epoch [40/40], Loss: 0.066746  
  
Training Pseudoinverse AE...  
Epoch [10/40], Loss: 0.073334  
Epoch [20/40], Loss: 0.070542  
Epoch [30/40], Loss: 0.069282  
Epoch [40/40], Loss: 0.068518  
  
Reconstruction MSE Comparison:  
--------------------------------------------------  
Model MSE   
--------------------------------------------------  
Three Layer AE 0.002259   
Tied Weights AE 0.002038   
Pseudoinverse AE 0.002308   
--------------------------------------------------

A number in black squares

Description automatically generated with medium confidence

**AUTOENCODERS - ΔΙΑΔΙΚΑΣΙΑ 1 ΣΕΛΙΔΑ 17 (**[**VAE.py**](https://github.com/nickpotamianos/Computer_Vision_Lab/blob/main/CV_5-AUTOENCODERS/VAE.ipynb)**)**

**(αʹ) Δημιουργία και Χρήση Fixed Noise Batch στον Αποκωδικοποιητή**

Στον κώδικα, η μεταβλητή fixed\_noise δημιουργείται μία φορά πριν από την έναρξη της εκπαίδευσης, χρησιμοποιώντας την κανονική κατανομή N(0,I) με διαστάσεις 64×2. Αυτή η σταθερή θορυβώδης είσοδος χρησιμοποιείται στον αποκωδικοποιητή για να παραχθούν ανακατασκευές στο τέλος συγκεκριμένων εποχών:

fixed\_noise = torch.randn(64, latent\_dim).to(device)  
  
Η συνάρτηση train\_epoch καλεί τη visualize\_fixed\_noise\_samples στις εποχές 1, 50 και 100. Μέσα σε αυτή τη συνάρτηση, ο αποκωδικοποιητής δέχεται το ίδιο batch θορύβου fixed\_noise και παράγει ανακατασκευασμένες εικόνες:  
  
**def** **visualize\_fixed\_noise\_samples**(model, fixed\_noise, device, epoch):  
 model.eval()  
 **with** torch.no\_grad():  
 samples = model.decode(fixed\_noise).cpu()  
 ...  
 plt.suptitle(f'Fixed Noise Samples - Epoch {epoch}')

Η παραπάνω διαδικασία δείχνει πως εξελίσσεται η ανακατασκευή που προκύπτει από το σταθερό batch θορύβου σε διαδοχικές εποχές. Συγκεκριμένα, τα γραφικά που παράγονται στις εποχές 1, 50 και 100 απεικονίζουν πώς βελτιώνεται η ικανότητα του δικτύου να παράγει ποιοτικές εικόνες χρησιμοποιώντας την ίδια είσοδο θορύβου, αντανακλώντας την εκμάθηση της γενετικής ικανότητας του αποκωδικοποιητή.

**(ϐʹ) Οπτικοποίηση Λανθάνουσας Αναπαράστασης των Δεδομένων Ελέγχου**

Μετά την ολοκλήρωση της εκπαίδευσης, η συνάρτηση visualize\_latent\_space χρησιμοποιείται για να δημιουργήσει την λανθάνουσα αναπαράσταση όλων των δεδομένων ελέγχου. Η διαδικασία αυτή περιλαμβάνει τα εξής βήματα:

Για κάθε batch δοκιμαστικών δεδομένων, ο κωδικοποιητής του VAE εξάγει τα μέσα μ των κατανομών των λανθάνουσων μεταβλητών.

Αυτές οι τιμές μ συγκεντρώνονται και απεικονίζονται σε scatter plot, όπου κάθε σημείο αντιστοιχεί σε μια εικόνα, και το χρώμα του σημείου καθορίζεται από την ετικέτα του ψηφίου.

**def** **visualize\_latent\_space**(model, test\_loader, device):  
 model.eval()  
 z\_points = []  
 labels = []  
  
 **with** torch.no\_grad():  
 **for** data, y **in** test\_loader:  
 data = data.to(device)  
 mu, \_ = model.encode(data) # Λήψη των μέσων των λανθάνουσων  
 z\_points.append(mu.cpu().numpy())  
 labels.append(y.numpy())  
  
 z\_points = np.concatenate(z\_points, axis=0)  
 labels = np.concatenate(labels, axis=0)  
  
 plt.figure(figsize=(10, 8))  
 scatter = plt.scatter(z\_points[:, 0], z\_points[:, 1], c=labels, cmap='tab10')  
 plt.colorbar(scatter)  
 plt.title('Latent Space Visualization')  
 plt.xlabel('z[0]')  
 plt.ylabel('z[1]')  
 plt.show()

Η παραπάνω συνάρτηση παράγει ένα scatter plot που απεικονίζει την κατανομή των ψηφίων στον χώρο των λανθάνουσων διαστάσεων zz (με διαστάσεις 2). Στο plot, τα σημεία χρωματίζονται ανάλογα με το ποιο ψηφίο αντιπροσωπεύουν, επιτρέποντας την οπτική εξέταση της ομαλοποίησης του χώρου και του πώς σχηματίζονται συμπλέγματα ανάλογα με την κατηγορία του ψηφίου.

A colorful circle with many dots

Description automatically generated with medium confidence