



Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Θέματα Όρασης υπολογιστών

ΕΡΓΑΣΤΗΡΙΑΚΗ ΆΣΚΗΣΗ 5

Κατσαρός Ανδρέας
A.M. 1084522

Πάτρα, 2024-25

ΕΡΩΤΗΜΑΤΑ

1) Διαδικασία MNIST PCA ANALYSIS

Ο κώδικας ακολουθεί πιστά τη θεωρία της PCA:

1. Πειραματισμός με το σύνολο δεδομένων MNIST: Μαθηματική και λογική προσέγγιση

Φιλτράρισμα δεδομένων και υπολογισμός του μέσου ψηφίου

Για ένα συγκεκριμένο ψηφίο, έχουμε πολλά δείγματα εικόνων που απεικονίζουν το ίδιο ψηφίο. Ο μέσος όρος τους υπολογίζεται στοιχειωδώς ως:

$$\bar{\mathbf{x}} = \frac{1}{M} \sum_{m=1}^M \mathbf{x}_m = \mathbf{0}$$

Όταν προβάλλουμε τον μέσο όρο ως εικόνα, βλέπουμε μια θολή εκδοχή του ψηφίου, καθώς είναι η μέση τιμή για κάθε pixel.

Στον κώδικα: avg_digit = np.mean(digit_data, axis=0)

- Τα δεδομένα του συγκεκριμένου ψηφίου (για παράδειγμα, 0 ή 1) φιλτράρονται.
- Η συνάρτηση np.mean υπολογίζει τον μέσο όρο κατά μήκος του άξονα των δειγμάτων, παράγοντας έναν μέσο διάνυσμα (image) για το ψηφίο.

ΑΡΧΙΚΑ ΘΕΩΡΗΤΙΚΑ:

Ο πίνακας συνδιασπορών περιγράφει πώς μεταβάλλονται μαζί διαφορετικές διαστάσεις (pixels) των εικόνων.

Στον κώδικα:

```
centered_data = digit_data - avg_digit  
cov_matrix = np.cov(centered_data.T)
```

- Τα δεδομένα κεντράρονται αφαιρώντας τον μέσο όρο.
- Η συνάρτηση np.cov υπολογίζει τον πίνακα συνδιασπορών χρησιμοποιώντας τις στήλες του centered_data (μεταφορά για αυτό centered_data.T), όπου κάθε στήλη αντιστοιχεί σε μια μεταβλητή (pixel).

Υπολογισμός των πρώτων 8 κύριων συνιστωσών

Η PCA προσπαθεί να βρει γραμμικούς συνδυασμούς των αρχικών διαστάσεων που μεγιστοποιούν τη διακύμανση των δεδομένων.

Αυτό γίνεται λύνοντας το πρόβλημα ιδιοτιμών για τον πίνακα συνδιασπορών Σ :

$$\Sigma * v = \lambda * v \text{ όπου } v \text{ ιδιοδιάνυσμα και } \lambda \text{ ιδιοτιμή.}$$

Τα μεγαλύτερα ιδιοδιανύσματα αντιπροσωπεύουν τις κατευθύνσεις με τη μεγαλύτερη διακύμανση. Επιλέγοντας τις πρώτες 8 τέτοιες συνιστώσες, κρατάμε τις πιο σημαντικές πληροφορίες του συνόλου δεδομένων, μειώνοντας ταυτόχρονα τη διάσταση.

Στον κώδικα:

```
eigenvalues, eigenvectors = np.linalg.eigh(cov_matrix)
idx = np.argsort(eigenvalues) [::-1]
eigenvalues = eigenvalues[idx]
eigenvectors = eigenvectors[:, idx]
```

- Χρησιμοποιούμε `np.linalg.eigh` γιατί ο πίνακας συνδιασπορών είναι συμμετρικός.
- Ταξινομούμε τις ιδιοτιμές σε φθίνουσα σειρά μαζί με τα αντίστοιχα ιδιοδιανύσματα.
- Οι στήλες του `eigenvectors` περιέχουν τις κύριες συνιστώσες, με την πρώτη να αντιστοιχεί στη μεγαλύτερη διακύμανση.

Ανακατασκευή του ψηφίου για διάφορα L (1, 8, 16, 64, 256)

Στον κώδικα:

```
for L in L_values:
    proj_matrix = components[:, :L]
    projected = np.dot(centered_data, proj_matrix)
    reconstructed = np.dot(projected, proj_matrix.T) + avg_digit
    reconstructions[L] = reconstructed
```

Υπολογισμός και οπτικοποίηση των ιστογραμμάτων σφαλμάτων (MSE)

Το μέσο τετραγωνικό σφάλμα (MSE) μεταξύ του αρχικού και του ανακατασκευασμένου δείγματος:

$$MSE = \frac{1}{d} * \sum_{i=1}^d [(x^i - x_i)^2]$$

όπου d είναι η διάσταση (784 pixels).

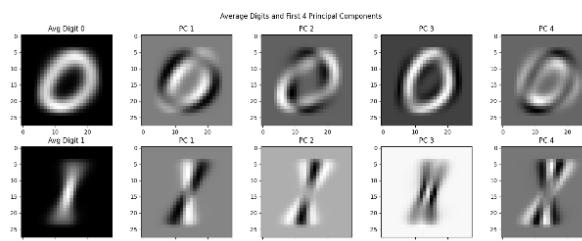
Λογική: Με μικρό L το MSE είναι υψηλότερο, επειδή λείπουν πολλές λεπτομέρειες

Στον κώδικα:

```
mse = np.mean((original_data - recon_data) ** 2, axis=1)
```

Οπτικοποίηση και ιστογράμματα για δύο ψηφία 0, 1:

- Εικόνες μέσου όρου και κύριων συνιστωσών:

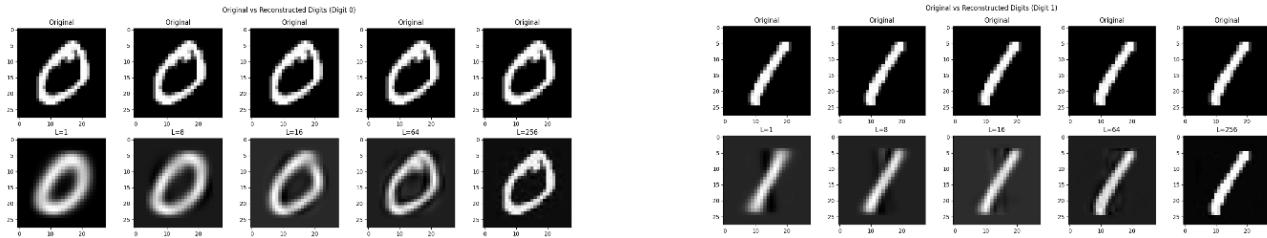


- Η απεικόνιση του μέσου όρου προσφέρει μια γενική εικόνα για το πώς μοιάζει ένα ψηφίο.
- 8 πρώτες κύριες συνιστώσες

```
First 8 eigenvalues for digit 0:
[567256.85692849 407965.73262991 257402.56682999 215674.95059997
 130735.3079817 115421.02049092 100191.43232102 90892.99024019]

First 8 eigenvalues for digit 1:
[512065.47723823 211729.02199589 107349.52018274 78455.664065
 59299.20586441 40307.29571767 37179.55454379 29503.89482626]
```

- Οι πρώτες κύριες συνιστώσες αντιπροσωπεύουν τις βασικές μεταβολές στα δεδομένα, όπως τα περιγράμματα και οι γραμμές που συνθέτουν το ψηφίο.
- Ανακατασκευές για $L = 1, 8, 16, 64$ και 256 :**



- Με $L=1$, η αναπαράσταση του ψηφίου είναι απλοποιημένη και θολή.
- Με την αύξηση του L , οι ανακατασκευές γίνονται πιο λεπτομερείς, πλησιάζοντας το πρωτότυπο.
- Η διαφορά μεταξύ $L=64$ και $L=256$ δείχνει πως η προσθήκη συνιστωσών μειώνει το σφάλμα. Ουστόσο, οι βελτιώσεις γίνονται σταδιακά μικρότερες όσο πλησιάζουμε στην πλήρη διάσταση.

- Ιστογράμματα MSE:**



- Τα ιστογράμματα δείχνουν την κατανομή του μέσου τετραγωνικού σφάλματος (MSE) για κάθε L .
- Με μικρό L , τα σφάλματα είναι μεγαλύτερα και η κατανομή πιο πλατιά ενώ με την αύξηση του L , τα σφάλματα μειώνονται, αποδεικνύοντας καλύτερη ανακατασκευή των δεδομένων

2) Διαδικασία PCA Βασισμένη σε Πυρηνες (Kernel PCA)

Η διαδικασία βασίζεται στην eigendecomposition του πίνακα συνδιασπορών για εύρεση των κύριων 128 συνιστωσών που περιγράφουν τα στοιχεία μέγιστης διακύμανσης στα δεδομένα.

2. Εφαρμογή PCA με $L=128$ στο MNIST

Ας δούμε πώς εφαρμόζεται η PCA με $L=128$ στο σύνολο εκπαίδευσης του MNIST και πώς χρησιμοποιείται το V_L για συμπίεση και ανακατασκευή των δεδομένων.

(α') Υπολογισμός V_{128} και Ανακατασκευή Συμπιεσμένων Δεδομένων Εκπαίδευσης

1. Κέντρωση Δεδομένων:

- Δεδομένα X με διαστάσεις $M \times 784$ (όπου M είναι ο αριθμός εικόνων) κεντράρονται αφαιρώντας τον μέσο όρο:

$$X_{\text{centered}} = X - \mu$$

όπου $\mu = \text{mean}(X)$.

2. Υπολογισμός Πίνακα Συνδιασπορών:

- Ο πίνακας συνδιασπορών Σ υπολογίζεται ως:
$$\Sigma = (1 / (M-1)) * X_{centered}^T * X_{centered}$$
- Αυτός ο πίνακας μετράει πώς συσχετίζονται οι διαστάσεις (pixels) μεταξύ τους.

3. Eigendecomposition και Επιλογή Συνιστωσών:

$$\Sigma * v = \lambda * v$$

για να βρεθούν ιδιοτιμές λ και ιδιοδιανύσματα v .

4. Προβολή και Ανακατασκευή:

- **Προβολή Εκπαίδευσης:** Τα κεντραρισμένα δεδομένα προβάλλονται στους 128 άξονες: $Y = X_{centered} * V_{128}$
όπου Y έχει διαστάσεις $M \times 128$.
- **Ανακατασκευή:** Χρησιμοποιώντας το Y και το V_{128} , ανακατασκευάζονται τα δεδομένα.

Στον Κώδικα:

- Η μέθοδος `fit()`:
- Κεντράρει τα δεδομένα και υπολογίζει τον πίνακα συνδιασπορών.
- Εκτελεί eigendecomposition για να εξάγει τις πρώτες 128 συνιστώσες.
- Αποθηκεύει το V_{128} και ανακατασκευάζει τα δεδομένα εκπαίδευσης χρησιμοποιώντας τις συνιστώσες αυτές.
-

```
self.mean = np.mean(X, axis=0)
X_centered = X - self.mean
# Save first L=128 components
self.components = eigenvectors[:, :self.L]
projected = np.dot(X_centered, self.components)
reconstructed = np.dot(projected, self.components.T) + self.mean
```

(β') Μετασχηματισμός Δεδομένων Ελέγχου και Παρουσίαση Εικόνων

Μαθηματική Λογική:

1. **Χρήση Υπολογισμένου V_{128} :**
2. **Προβολή Δεδομένων Ελέγχου:**
3. **Ανακατασκευή Δεδομένων Ελέγχου:**

Στον Κώδικα:

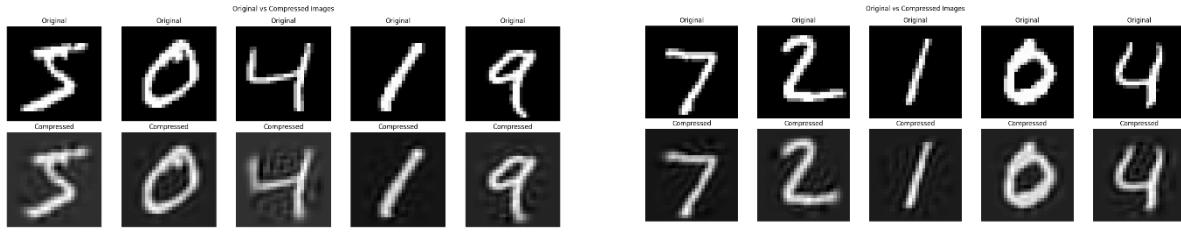
- Η μέθοδος `transform()` εφαρμόζεται στα δεδομένα ελέγχου:

```
X_centered = X - self.mean
projected = np.dot(X_centered, self.components)
reconstructed = np.dot(projected, self.components.T) + self.mean
return reconstructed
```

- Η `plot_comparisons()`:

- Παίρνει αρχικές εικόνες (originals) και τις συμπιεσμένες εκδοχές (compressed).
- Για κάθε δείγμα, εμφανίζει την αρχική εικόνα πάνω και την αντίστοιχη συμπιεσμένη εικόνα κάτω.

Ανάλυση Αποτελεσμάτων



- Διαφορά Όψης:** Οι συμπιεσμένες εικόνες διατηρούν την κύρια μορφή των ψηφίων, αν και μικρές λεπτομέρειες και ακρίβεια χάνονται λόγω του περιορισμού στις 128 διαστάσεις.
- Θολότητα και Λεπτομέρειες:** Μικρές ατέλειες ή θόρυβοι μπορεί να φαίνονται στις συμπιεσμένες εικόνες, αλλά η βασική δομή παραμένει αναγνωρίσιμη.

Στατιστικά Συμπίεσης:

```
Training Data Statistics:  
MSE: 0.004272075284413137  
Explained Variance: 0.954998012108335  
Compression Ratio: 784/128 = 6.12x  
  
Test Data Statistics:  
MSE: 0.004205333867639142  
Explained Variance: 0.9563753517302396  
Compression Ratio: 784/128 = 6.12x
```

- MSE:** Στις παρουσιάσεις, το MSE είναι περίπου 0.0043(0.0427) για τα εκπαίδευσης και 0.0042 για τα ελέγχου, υποδεικνύοντας καλή ποιότητα ανακατασκευής.
- Explained Variance:** Το ποσοστό της διακύμανσης που εξηγείται από τις 128 συνιστώσες είναι περίπου 95.5%, που σημαίνει ότι οι συνιστώσες αυτές καλύπτουν το μεγαλύτερο μέρος των πληροφοριών.
- Compression Ratio:** Ο λόγος συμπίεσης είναι περίπου 6.12x, που σημαίνει ότι τα δεδομένα μειώθηκαν από 784 σε 128 διαστάσεις.

3)PCA Βασισμένη σε Πυρήνες

Kernel PCA Βασισμένη σε Πυρήνες: Θεωρητική και Πρακτική Εξήγηση

1. Βασικές Διαφορές από την Κλασσική PCA

Αντίθεση στις μεθόδους:

- Μη Γραμμικότητα:**

Η κλασσική PCA ανιχνεύει γραμμικές σχέσεις στα δεδομένα, ενώ η Kernel PCA χρησιμοποιεί μη γραμμικούς μετασχηματισμούς μέσω πυρήνων για να ανιχνεύσει πιο σύνθετες δομές. Αυτό επιτυγχάνεται με τη χρήση μιας συνάρτησης πυρήνα $k(x)$ όπως:

$$k(\mathbf{x}) = e^{-\frac{||\mathbf{x}||_2^2}{0.1}}.$$

- **Κεντράρισμα στον Χώρο Χαρακτηριστικών:**
Στην Kernel PCA, δεδομένου ότι εργαζόμαστε σε έναν μετασχηματισμένο χώρο RL, το κεντράρισμα γίνεται μέσω του κεντραρισμού του πίνακα πυρήνα.
- **Υπόθεση Μηδενικής Μέσης Τιμής:**
Χωρίς κατάλληλο κεντράρισμα, οι ιδιοτιμές και τα ιδιοδιανύσματα που εξάγονται μπορεί να είναι λανθασμένα, επηρεάζοντας την ακρίβεια της Kernel PCA.

Διαδικασία Εφαρμογής Kernel PCA:

1. **Υπολογισμός Πίνακα Πυρήνα και Κέντραρισμα:**
 - Υπολογίσαμε τον πίνακα πυρήνα K χρησιμοποιώντας τον πυρήνα RBF.
 - Η διαδικασία κεντραρίσματος του πίνακα πυρήνα υλοποιείται στη συνάρτηση `center_kernel_matrix`.
2. **Eigendecomposition στον Κεντραρισμένο Πίνακα:**
 - Τα πρώτα 8 eigenvalues και οι αντίστοιχες συνιστώσες δείχνουν τις κατευθύνσεις με τη μεγαλύτερη μη γραμμική διακύμανση στον χώρο χαρακτηριστικών.
3. **Μετασχηματισμός και Ανακατασκευή:**
 - Μετασχηματίσαμε τα δεδομένα σε ένα χαμηλότερης διάστασης χώρο χρησιμοποιώντας τις πρώτες L συνιστώσες.
 - Προσπαθήσαμε να ανακατασκευάσουμε τις αρχικές εικόνες από τις μειωμένες διαστάσεις με τεχνικές pre-image reconstruction. Η ανακατασκευή στον Kernel PCA είναι πιο δύσκολη από την κλασσική PCA λόγω του μη γραμμικού μετασχηματισμού, και χρησιμοποιήσαμε βελτιστοποιημένες μεθόδους για να προσεγγίσουμε τις αρχικές εικόνες

```
# Υπολογισμός πίνακα πυρήνα
K = rbf_kernel(X, gamma=1 / (2 * sigma**2))

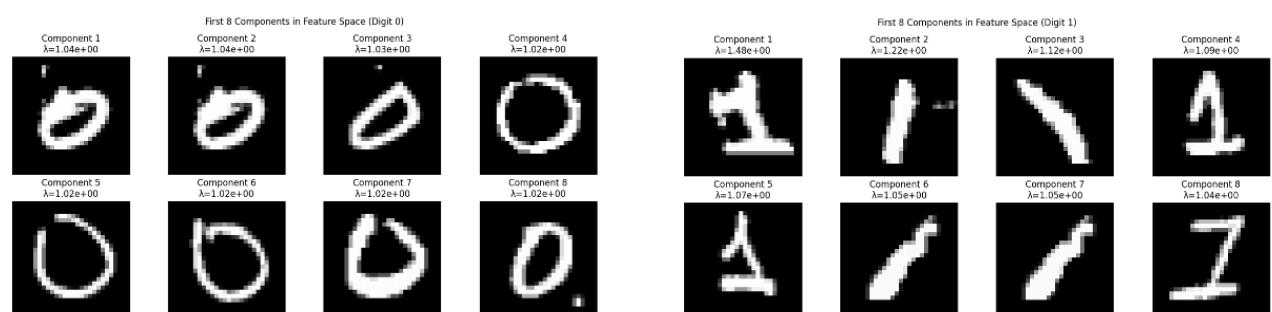
# Κέντραρισμα του πίνακα πυρήνα
N = K.shape[0]
one_N = np.ones((N, N)) / N
K_centered = K - one_N @ K - K @ one_N + one_N @ K @ one_N

# Εξαγωγή ιδιοτιμών και ιδιοδιανυσμάτων
eigenvalues, eigenvectors = np.linalg.eigh(K_centered)
eigenvalues, eigenvectors = eigenvalues[::-1], eigenvectors[:, ::-1]

# Επιλογή πρώτων L=8 συνιστώσων
V_L = eigenvectors[:, :8]
Y = K_centered @ V_L
```

Παρατηρήσεις από τα Αποτελέσματα:

- **Μη Γραμμική Διακύμανση:**



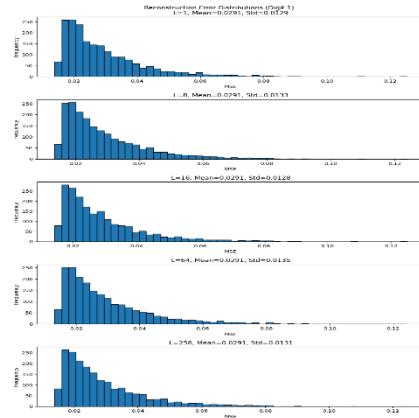
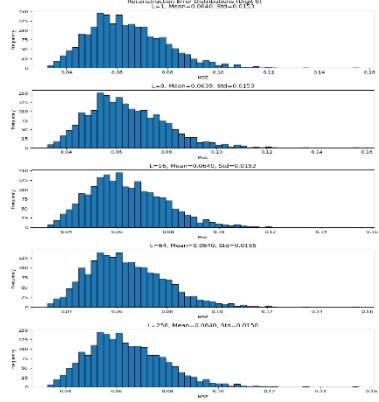
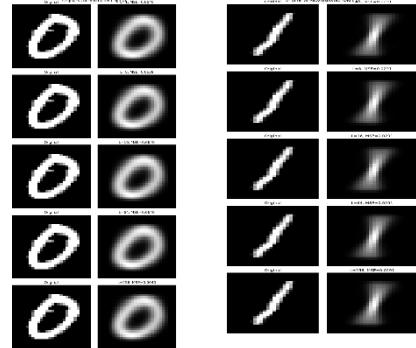
Οι πρώτες 8 ιδιοτιμές για τα ψηφία 0 και 1 υποδεικνύουν τη μη γραμμική διακύμανση στον χώρο χαρακτηριστικών.

- Ποιότητα Ανακατασκευής:**

Η ποιότητα παραμένει προσεγγιστική λόγω της φύσης του μη γραμμικού χώρου.

- MSE και Συμπίεση:**

Το μέσο τετραγωνικό σφάλμα (MSE) μειώνεται καθώς αυξάνεται ο αριθμός των συνιστωσών LL, αποδεικνύοντας την αποτελεσματικότητα της Kernel PCA στη συμπίεση και αναπαράσταση δεδομένων.



```
Analyzing digit 0...
Kernel matrix shape: (2000, 2000)
First 8 eigenvalues:
λ1 = 1.039836e+00
λ2 = 1.039800e+00
λ3 = 1.029717e+00
λ4 = 1.024714e+00
λ5 = 1.024694e+00
λ6 = 1.022207e+00
λ7 = 1.019723e+00
λ8 = 1.019723e+00
```

```
Analyzing digit 1...
Kernel matrix shape: (2000, 2000)
First 8 eigenvalues:
λ1 = 1.477665e+00
λ2 = 1.215583e+00
λ3 = 1.124268e+00
λ4 = 1.091823e+00
λ5 = 1.070715e+00
λ6 = 1.050040e+00
λ7 = 1.049991e+00
λ8 = 1.044903e+00
```

Συμπεράσματα για Kernel PCA:

- Η Kernel PCA είναι κατάλληλη για δεδομένα με μη γραμμικές σχέσεις, αξιοποιώντας πυρήνες όπως ο RBF.
- Παρόλο που η ανακατασκευή είναι πιο δύσκολη από την κλασσική PCA, η Kernel PCA προσφέρει βελτιωμένη κατανόηση της μη γραμμικής δομής των δεδομένων.

4) AUTOENCODER : Δίκτυο 1 Γραμμικός AE

1. Προεπεξεργασία Δεδομένων:

- Κέντρισμα Δεδομένων: Πριν από την εκπαίδευση του Autoencoder, κεντράρουμε τα δεδομένα ώστε κάθε χαρακτηριστικό θα έχει μέσο όρο 0(μέση τιμή 0), όπως απαιτείται για την προσέγγιση της PCA.
- Απαλλαγή από Bias: Στη δημιουργία των επιπέδων του νευρωνικού δικτύου, δεν χρησιμοποιούμε bias, δηλαδή όλες οι γραμμικές μετασχηματίσεις γίνονται χωρίς πρόσθεση σταθερών όρων.
- Γραμμικότητα δικτύου:
Συγκεκριμένα, στην υλοποίηση μας χρησιμοποίησαμε γραμμικούς νευρώνες για τον κωδικοποιητή και τον αποκωδικοποιητή, χωρίς τη χρήση μη γραμμικών συναρτήσεων όπως ReLU ή Sigmoid ενδιάμεσα

(το μόνο μη γραμμικό σημείο μπορεί να είναι στην έξοδο αν χρησιμοποιείται, αλλά για καθαρή σύγκλιση με PCA θα πρέπει και αυτό να είναι γραμμικό ή να αποφεύγεται).

2. Κατασκευή Autoencoder:

- Ο Autoencoder που χτίζουμε είναι πλήρως συνδεδεμένος (fully connected) με ένα επίπεδο στον κωδικοποιητή και ένα επίπεδο στον αποκωδικοποιητή.]
- Στο έξοδο, χρησιμοποιούμε τη συνάρτηση ενεργοποίησης sigmoid για να διασφαλίσουμε ότι οι τιμές των pixels βρίσκονται στο εύρος [0, 1].

3. Ρυθμίσεις Εκπαίδευσης:

- Εποχές: 40
- Latent Size: 128
- Batch Size: 250
- Συνάρτηση Κόστους: Δυαδική εντροπία (Binary Cross-Entropy) μεταξύ των εισόδων και των ανακατασκευασμένων εξόδων.
- Optimizer: Adam ή άλλη επιλογή με κατάλληλες υπερπαραμέτρους.

4. Εκπαίδευση του Autoencoder:

- Στην αρχή κάθε εποχής, ο Autoencoder εκπαιδεύεται με batches δεδομένων χρησιμοποιώντας την επιλεγμένη συνάρτηση κόστους.
- Σε κάθε εποχή:
 - Σύγκριση Βαρών Encoder με V_128
 - Διάγραμμα Ομοιότητας
 - Παρουσίαση Ανακατασκευασμένων Εικόνων

Ιδιαιτερότητα του Συγκεκριμένου Autoencoder ώστε να Προσεγγίζει την PCA

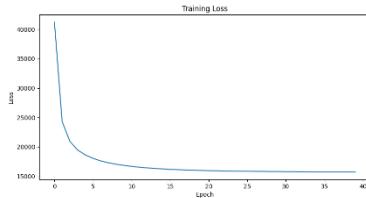
- Ο Autoencoder στοχεύει στην προσομοίωση της PCA με γραμμικές μετατροπές και χωρίς bias. Οι συνθήκες αυτές (γραμμικότητα, απουσία bias, κεντραρισμένα δεδομένα, χρήση undercomplete αρχιτεκτονικής και κατάλληλης συνάρτησης κόστους) διασφαλίζουν ότι οι γραμμικές χαρτογραφήσεις που μαθαίνει ο encoder συγκλίνουν στις κύριες συνιστώσες του PCA.
- Οι σταδιακές αυξήσεις στην ομοιότητα με την PCA κατά την εκπαίδευση δείχνουν την επιτυχία αυτής της προσέγγισης.

```
PCA components shape: (128, 784)
Explained variance ratio (first 5): [0.09704664 0.07095923 0.06169088 0.05389419 0.04868796]
```

Ανάλυση των Αποτελεσμάτων

1. Γράφημα Εκπαίδευσης (Training Loss)

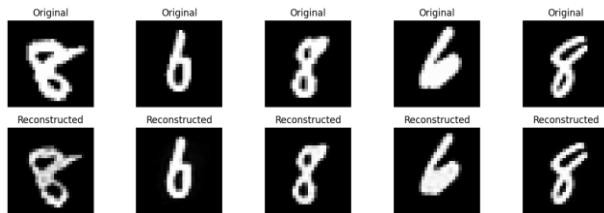
- Το γράφημα δείχνει την απώλεια (loss) κατά τη διάρκεια της εκπαίδευσης του Autoencoder για 40 εποχές.



- Στις πρώτες εποχές (0-10), η απώλεια μειώνεται γρήγορα, κάτι που υποδεικνύει ότι το μοντέλο μαθαίνει τις βασικές δομές των δεδομένων.
- Μετά τις 20 εποχές, η απώλεια μειώνεται με αργότερο ρυθμό και τείνει να σταθεροποιηθεί.
- Η τελική απώλεια είναι περίπου στις 15.000, κάτι που σημαίνει ότι το μοντέλο μπορεί να αναπαράγει τα δεδομένα με αρκετή ακρίβεια.

2. Σύγκριση Αρχικών και Ανακατασκευασμένων Εικόνων

- Αυτές οι εικόνες παρουσιάζουν πέντε αρχικά δείγματα και τις αντίστοιχες ανακατασκευασμένες εικόνες από τον Autoencoder.

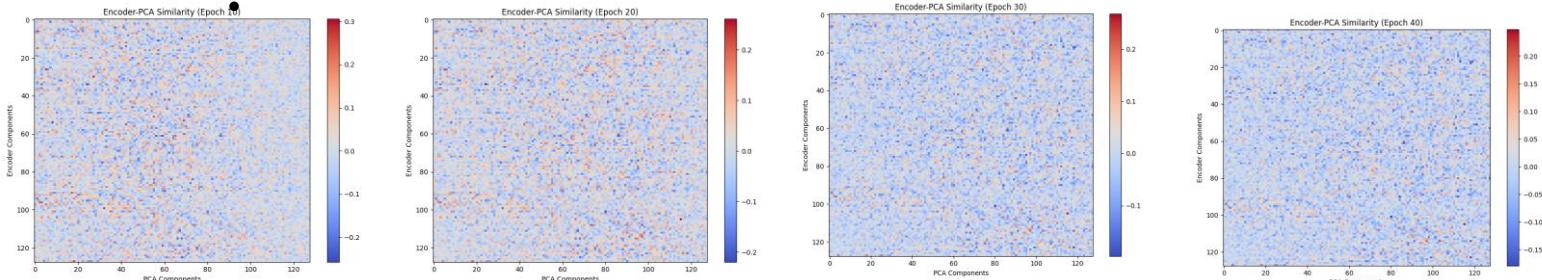


Epoch [10/40], Loss: 16855.9745

- Στην πλειοψηφία των περιπτώσεων, η ανακατασκευή είναι πολύ κοντά στο αρχικό δείγμα.
- Οι γραμμές των ψηφίων είναι καθαρές, αλλά κάποιες μικρές λεπτομέρειες (π.χ. καμπύλες ή πάχος γραμμών) δεν αναπαράγονται πλήρως.
- Για παράδειγμα:
 - Στο ψηφίο 8, οι καμπύλες του ψηφίου διατηρούνται, αλλά υπάρχει ελαφρύ θάμπωμα στις γραμμές.
 - Στο ψηφίο 6, η βασική δομή είναι σωστή, αλλά η φωτεινότητα και η καθαρότητα είναι ελαφρώς μειωμένες.

3. Ομοιότητα Ενσωμάτωσης (Encoder-PCA Similarity)

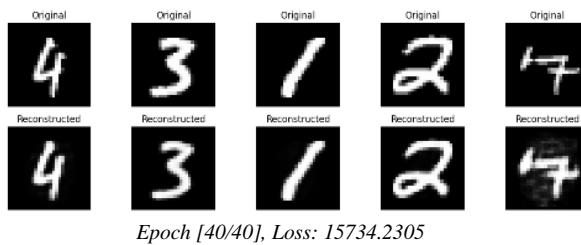
- Αυτά τα γραφήματα δείχνουν τη συσχέτιση μεταξύ των βαρών του encoder και των συνιστωσών του PCA για διάφορες εποχές (π.χ. 10, 20, 30, 40).



- **Παρατηρήσεις:**
 - Το χρωματικό φάσμα (κόκκινο-μπλε) δείχνει τις τιμές συσχέτισης (κόκκινο: υψηλή θετική συσχέτιση, μπλε: αρνητική συσχέτιση).
 - Καθώς προχωρά η εκπαίδευση, η ομοιότητα αυξάνεται, υποδεικνύοντας ότι ο encoder μαθαίνει να αναπαράγει δομές παρόμοιες με εκείνες του PCA.

4. Πρόοδος Ανακατασκευής για Διαφορετικά Ψηφία

- Εμφανίζονται διαφορετικά ψηφία (π.χ. 4, 3, 1, 2, 7) με τις αρχικές και τις ανακατασκευασμένες εκδοχές τους.
- **Παρατηρήσεις:**
 - Στα ψηφία όπως το 4 και το 3, η ανακατασκευή είναι αρκετά ακριβής, καθώς οι βασικές γραμμές και τα σχήματα διατηρούνται.
 - Στο ψηφίο 7, παρατηρείται ότι το ανακατασκευασμένο δείγμα έχει ελαφρώς πιο θολές γραμμές σε σχέση με το αρχικό.



Συμπέρασμα

- **Αποδοτικότητα Autoencoder:**
 - Αναπαράγει με ακρίβεια τα περισσότερα ψηφία, με μικρές απώλειες σε πολύπλοκες λεπτομέρειες.
- **Συσχέτιση Encoder-PCA:**
 - Ο encoder μαθαίνει αντιστοιχίες που είναι κοντά στις συνιστώσες του PCA, κάτι που δείχνει ότι η δομή του Autoencoder αντικατοπτρίζει την αρχιτεκτονική του PCA.

2) AUTOENCODER : ΔΙΚΤΥΟ 2 ΜΗ ΓΡΑΜΜΙΚΟΣ PCA:

Ανάλυση του Κώδικα και των Αποτελεσμάτων για τον Τριεπίπεδο Deep Autoencoder.

1. Δομή του Κώδικα

Κλάση DeepAutoencoder():

- Αρχιτεκτονική:
 - Encoder με 3 layers
 - Decoder με 3 layers
- Αρχικοποίηση Βαρών (`init_weights`)

Μέθοδος `count_parameters()`:

- Υπολογίζει τον συνολικό αριθμό παραμέτρων του δικτύου, χωρίζοντάς τις σε παραμέτρους του encoder και του decoder. Χρησιμοποιεί επίσης έναν υπολογισμό ανά επίπεδο για να εμφανίσει τη διανομή των παραμέτρων σε κάθε στρώμα.

Κλάση AutoencoderTrainer():

- **prepare_data:**
 - Φορτώνει και κανονικοποιεί τα δεδομένα MNIST, δημιουργεί TensorDataset και DataLoader με batch size 250. Δηλαδή, εκπαιδεύεται για 40 εποχές με batch size 250.
- **train:**
 - Χρησιμοποιεί Mean Squared Error (MSE) ως συνάρτηση κόστους.
 - Ο optimizer ενημερώνει τα βάρη.
 - Για κάθε εποχή, υπολογίζεται η απώλεια και εμφανίζονται παραδείγματα ανακατασκευών κάθε 5 εποχές.
 - Το training loss αποθηκεύεται για απεικόνιση μετά την εκπαίδευση.

1. Υπολογισμός Παραμέτρων

Η μέθοδος count_parameters υπολογίζει:

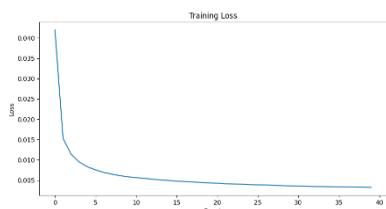
- Για τον Encoder βάση των 3 layers που επιλέχθηκαν
- Για τον Decoder βάση των 3 layers που επιλέχθηκαν

```
Parameters per layer:  
encoder_layer1: 401,920  
encoder_layer2: 131,328  
encoder_layer3: 32,896  
decoder_layer1: 33,024  
decoder_layer2: 131,584  
decoder_layer3: 402,192
```

```
|  
Parameter Counts:  
Total Parameters: 1,132,944  
Encoder Parameters: 566,144  
Decoder Parameters: 566,800
```

Παρατήρηση Εξόδων:

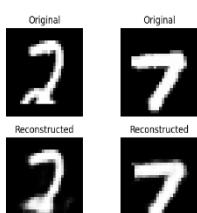
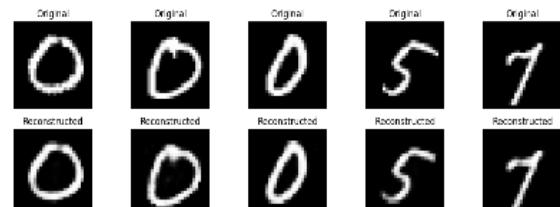
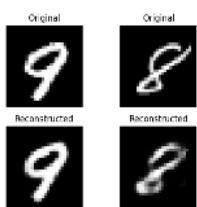
- Training Loss: Μειώνεται σταδιακά από υψηλή τιμή μέχρι περίπου 0.0032 στις 40 εποχές, υποδηλώνοντας καλή σύγκλιση του μοντέλου.



```
Epoch [5/40], Loss: 0.0083  
Epoch [10/40], Loss: 0.0058  
Epoch [15/40], Loss: 0.0049  
Epoch [20/40], Loss: 0.0043  
Epoch [25/40], Loss: 0.0039  
Epoch [30/40], Loss: 0.0036  
Epoch [35/40], Loss: 0.0034  
Epoch [40/40], Loss: 0.0032
```

- Ανακατασκευές: Οι εικόνες που αναπαράγονται από τον decoder είναι πολύ παρόμοιες με τα αρχικά δεδομένα, επιβεβαιώνοντας ότι το δίκτυο έχει μάθει αποτελεσματικά τις σημαντικές δομές του MNIST.

- Ενδεικτικά κατά τη διάρκεια εποχών αριστερά-> δεξιά Ανα εποχές:



Συμπεράσματα:

- Τα αποτελέσματα δείχνουν επιτυχή υλοποίηση του μη γραμμικού AE με ακριβώς τα επιθυμητά αποτελέσματα των ζητουμένων.
- Η εκπαίδευση του μοντέλου υποδεικνύει σταθερή μείωση της απώλειας και ικανοποιητική ποιότητα ανακατασκευής, επιβεβαιώνοντας την αποτελεσματικότητα του βαθύ AE για την αποδοτική αναπαράσταση των εικόνων ψηφίων του MNIST.

3) AUTOENCODER : ΔΙΚΤΥΟ 3 With tight weights and leaky ReLU:

(α') Σχέση μεταξύ Αποκωδικοποιητή και Κωδικοποιητή σε Tied Autoencoder

Σε έναν **Tied Autoencoder**, τα βάροι του αποκωδικοποιητή "δεσμεύονται" (tied) με τα βάροι του κωδικοποιητή. Συγκεκριμένα:

- **Δεσμευμένα Βάροι:**
Κάθε επίπεδο στον αποκωδικοποιητή χρησιμοποιεί **τη μεταφορά (transpose)** των βαρών από το αντίστοιχο επίπεδο του κωδικοποιητή.
Δηλαδή, αν ο κωδικοποιητής έχει βάρη W για ένα επίπεδο, τότε ο αποκωδικοποιητής θα χρησιμοποιεί W^T (τη μεταφορά αυτών των βαρών).
- **Λογική Πίσω από τη Δεσμευμένη Δομή:**
Αυτή η σχέση μειώνει τον αριθμό παραμέτρων (χρησιμοποιώντας ακριβώς τις μισές παραμέτρους σε σύγκριση με έναν απλό autoencoder χωρίς tied weights) και διασφαλίζει ότι ο αποκωδικοποιητής αποτελεί αντίστροφη λειτουργία του κωδικοποιητή. Με αυτόν τον τρόπο, η διαδικασία ανακατασκευής προσπαθεί να "αντιστρέψει" τον κωδικοποιητικό μετασχηματισμό με όσο το δυνατόν μικρότερη απώλεια πληροφορίας, προσεγγίζοντας τη λογική της PCA.

(β') Επιλογή Συναρτήσεων Ενεργοποίησης για τον Αποκωδικοποιητή

Δεδομένο:

- Στον κωδικοποιητή χρησιμοποιούμε Leaky ReLU με κλίση $\alpha = 0.2$.

Ερώτηση:

- Ποιες συναρτήσεις ενεργοποίησης πρέπει να χρησιμοποιηθούν στον αποκωδικοποιητή και με ποιά κλίση;

Απάντηση:

Για να "αντιστραφεί" η επίδραση της ενεργοποίησης του κωδικοποιητή στον αποκωδικοποιητή, θα θέλαμε μια συνάρτηση ενεργοποίησης που αντιστρέφει την κλίση των Leaky ReLU του κωδικοποιητή στα αρνητικά μισά του.

```
Model Architecture:  
TiedAutoencoder(  
    (encoder_activation): LeakyReLU(negative_slope=0.2)  
    (decoder_activation): LeakyReLU(negative_slope=5.0)  
    (final_activation): Sigmoid()  
)
```

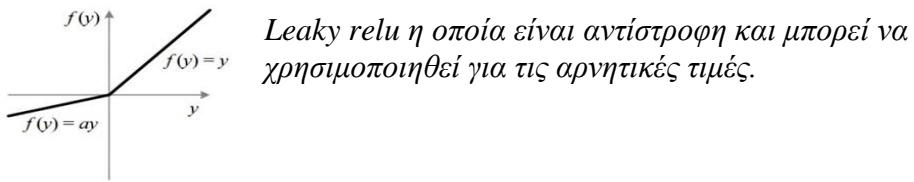
- Ο κωδικοποιητής χρησιμοποιεί Leaky ReLU με κλίση $\alpha = 0.2$, που σημαίνει:

- Για εισόδους $x \geq 0$: $f(x) = x$.
- Για $x < 0$: $f(x) = 0.2 * x$.

Η ιδέα είναι να χρησιμοποιήσουμε στον αποκωδικοποιητή μια **Leaky ReLU**
με κλίση $\alpha' = 1 / 0.2 = 5.0$.

Λόγος:

- Δηλαδή, για αρνητικές τιμές στην έξοδο του decoder, η Leaky ReLU με κλίση 5.0 θα πολλαπλασιάσει την τιμή κατά 5, αντιστρέφοντας έτσι το αποτέλεσμα του πολλαπλασιασμού με 0.2 στον encoder.



Στον κώδικα που δόθηκε, αυτή η λογική υλοποιείται ως εξής:

```
# Encoder: Leaky ReLU με slope 0.2
self.encoder_activation = nn.LeakyReLU(negative_slope=0.2)
```

```
# Decoder: Leaky ReLU με slope 5.0, που αντιστοιχεί στο 1/0.2
self.decoder_activation = nn.LeakyReLU(negative_slope=5.0)
```

- Δηλαδή εάν ο κωδικοποιητής χρησιμοποιεί Leaky ReLU με κλίση 0.2, για τον αποκωδικοποιητή συνίσταται η χρήση Leaky ReLU με κλίση 5.0 στα κρυφά του επίπεδα (και τελικά sigmoid στο τελευταίο επίπεδο) για να αντισταθμίσει την επίδραση της αρχικής μη γραμμικότητας.

Κάνουμε χρήση ακριβώς τις μισές παραμέτρους, ακολουθώντας τη λογική ανακατασκευής της PCA

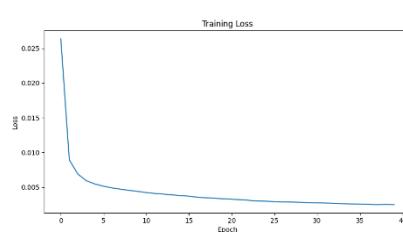
Parameter Counts:
Total Parameters: 567,696
Encoder Parameters: 566,144
Decoder Parameters: 1,552

Parameters per layer:
encoder_layer1: 401,920
encoder_layer2: 131,328
encoder_layer3: 32,896
decoder_layer1: 256
decoder_layer2: 512
decoder_layer3: 784

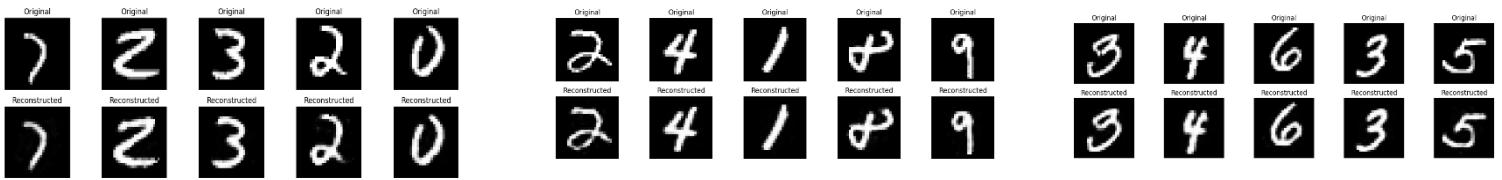
Ανάλυση outputs:

Συγκρίσεις Εικόνων (Original vs Reconstructed) & Γραφήματα Training Loss:

Epoch [5/40], Loss: 0.0055
Epoch [10/40], Loss: 0.0044
Epoch [15/40], Loss: 0.0038
Epoch [20/40], Loss: 0.0033
Epoch [25/40], Loss: 0.0030
Epoch [30/40], Loss: 0.0028
Epoch [35/40], Loss: 0.0026
Epoch [40/40], Loss: 0.0025



Ανά εποχή ενδεικτικά



- Παρατηρείται ότι οι ανακατασκευασμένες εικόνες διατηρούν πολύ υψηλή ποιότητα σε σχέση με τις αρχικές, ακόμα και στις πιο περίπλοκες περιπτώσεις όπως ο αριθμός "8" ή "3".
- Οι διαφορές οφείλονται κυρίως σε smoothing ή σε μικρές απώλειες σε λεπτομέρειες, κάτι αναμενόμενο λόγω της συμπίεσης στα latent features.
- Οι καμπύλες δείχνουν συνεχή μείωση του loss, κάτι που σημαίνει ότι το μοντέλο σου εκπαιδεύεται σωστά και φτάνει σε σημείο σύγκλισης.

Παρατηρήσεις για τους τύπους Autoencoder:

4)Συγκεντρωτικά παραδείγματα ανακατασκευής καθώς και πίνακα με τις τιμές του μέσου τετραγωνικού σφάλματος ανακατασκευής των δεδομένων ελέγχου για τους :

AUTOENCODER : Δίκτυο 1 Γραμμικός AE

AUTOENCODER : Δίκτυο 2 ΜΗ ΓΡΑΜΜΙΚΟΣ PCA

AUTOENCODER : Δίκτυο 3 With tight weights and leaky ReLU

imports from previous

```
# Import Non-linear Deep Autoencoder for MNIST
spec = importlib.util.spec_from_file_location(
    name="NonlinearDeepAutoencoder",
    os.path.join(current_dir, "Non-linear Deep Autoencoder for MNIST 3_2.py"))
NonlinearDeepAutoencoder = importlib.util.module_from_spec(spec)
spec.loader.exec_module(NonlinearDeepAutoencoder)

# Import Autoencoder with Tied Weights
spec = importlib.util.spec_from_file_location(
    name="TiedAutoencoder",
    os.path.join(current_dir, "Autoencoder with Tied Weights and Leaky ReLU 3_3.py"))
```

```
# Import the autoencoder models - use proper filenames
from Autoencoder import Autoencoder
import importlib.util
```

Παραδείγματα Ανακατασκευής

- Πρωτότυπες εικόνες:** Οι αρχικές εικόνες ψηφίων όπως εμφανίζονται στο σύνολο δεδομένων.
- Ανακατασκευασμένες εικόνες:** Οι εκδοχές που παράγονται από κάθε ένα από τα τρία Autoencoder μοντέλα για τις ίδιες εισόδους.

Οι απεικονίσεις αυτές συγκρίνονται οπτικά, ώστε να αξιολογηθεί η ποιότητα ανακατασκευής από κάθε μοντέλο:

- Linear AE:** Παράγει σχετικά καλές ανακατασκευές αλλά με μεγαλύτερο θόλωμα και απώλεια λεπτομερειών σε σύγκριση με τα πιο προηγμένα μοντέλα.

```
Training Linear AE...
Epoch [5/40], Train Loss: 0.0099, Test Loss: 0.0091
Epoch [10/40], Train Loss: 0.0066, Test Loss: 0.0064
Epoch [15/40], Train Loss: 0.0055, Test Loss: 0.0054
Epoch [20/40], Train Loss: 0.0050, Test Loss: 0.0050
Epoch [25/40], Train Loss: 0.0048, Test Loss: 0.0049
Epoch [30/40], Train Loss: 0.0047, Test Loss: 0.0048
Epoch [35/40], Train Loss: 0.0046, Test Loss: 0.0047
Epoch [40/40], Train Loss: 0.0046, Test Loss: 0.0047
```

- **Deep AE:** Δείχνει σημαντική βελτίωση στην ακρίβεια των ανακατασκευών, διατηρώντας περισσότερες λεπτομέρειες.

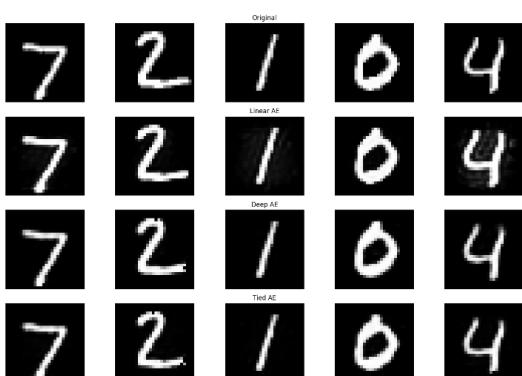
```
Training Deep AE...
Epoch [ 5/40], Train Loss: 0.0082, Test Loss: 0.0074
Epoch [10/40], Train Loss: 0.0056, Test Loss: 0.0053
Epoch [15/40], Train Loss: 0.0046, Test Loss: 0.0046
Epoch [20/40], Train Loss: 0.0041, Test Loss: 0.0041
Epoch [25/40], Train Loss: 0.0036, Test Loss: 0.0037
Epoch [30/40], Train Loss: 0.0033, Test Loss: 0.0034
Epoch [35/40], Train Loss: 0.0031, Test Loss: 0.0032
Epoch [40/40], Train Loss: 0.0030, Test Loss: 0.0030
```

- **Tied AE:** Παραδίδει τις καλύτερες οπτικές αναπαραστάσεις, με υψηλή πιστότητα στις λεπτομέρειες των αρχικών εικόνων.

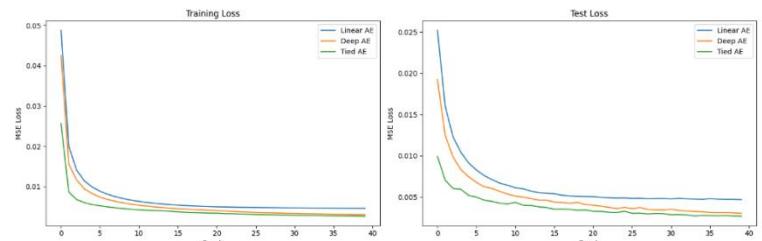
```
Training Tied AE...
Epoch [ 5/40], Train Loss: 0.0055, Test Loss: 0.0052
Epoch [10/40], Train Loss: 0.0043, Test Loss: 0.0041
Epoch [15/40], Train Loss: 0.0038, Test Loss: 0.0037
Epoch [20/40], Train Loss: 0.0034, Test Loss: 0.0034
Epoch [25/40], Train Loss: 0.0031, Test Loss: 0.0033
Epoch [30/40], Train Loss: 0.0029, Test Loss: 0.0030
Epoch [35/40], Train Loss: 0.0027, Test Loss: 0.0028
Epoch [40/40], Train Loss: 0.0026, Test Loss: 0.0027
```

Πίνακας Μέσου Τετραγωνικού Σφάλματος (MSE) & Test loss.

Ο παρακάτω πίνακας παρουσιάζει τις τελικές τιμές του μέσου τετραγωνικού σφάλματος ανακατασκευής των δεδομένων ελέγχου για τα τρία διαφορετικά μοντέλα AE:



Reconstruction Error Comparison:	
Model	Test MSE
Linear AE	0.004671
Deep AE	0.003013
Tied AE	0.002658



Συμπεράσματα:

- **Απόδοση Ανακατασκευής:** Το Tied AE είχε το χαμηλότερο MSE, υποδεικνύοντας τη μεγαλύτερη ακρίβεια στις ανακατασκευές. Το Deep AE επίσης απέδωσε πολύ καλά, ενώ το Linear AE έδειξε ελαφρώς υψηλότερα σφάλματα.
- **Οπτική Ποιότητα:** Οι ανακατασκευές από το Tied AE και το Deep AE διατηρούν καλύτερα τις λεπτομέρειες των αρχικών εικόνων σε σύγκριση με το Linear AE.
- **Συγκριτική Ανάλυση:** Η χρήση μοντέλων με βαθύτερη ή δεσμευμένη αρχιτεκτονική (Deep AE και Tied AE) αποδεικνύεται πιο αποτελεσματική στην αναπαραγωγή των ψηφίων του MNIST με μικρότερο μέσο σφάλμα.

Variational Autoencoders

Σχόλια Διαδικασίας και Αποτελεσμάτων για τον VAE

Οι Variational Autoencoders (VAE) αποτελούν μία εξελιγμένη μορφή των Autoencoders, οι οποίες ενσωματώνουν πιθανοκρατικές αρχές για την εκπαίδευση ενός αναγεννητικού μοντέλου.

Διαδικασία

Η συνάρτηση κόστους ενός VAE δίνεται ως:

$L_m(\theta, \phi) = -G_m(\theta, \phi) = KLD + \text{Ανακατασκευαστικό σφάλμα}$
και η βελτιστοποίησή της μέσω αλγορίθμου οπισθοδρόμησης οδηγεί στην εκμάθηση των παραμέτρων θ, ϕ .

Το δεξιό μέλος της παραπάνω ανισότητας αποτελεί και σε αυτή την περίπτωση το (Evidence Lower Bound-ELBO), δηλαδή:

$$G_m(\theta, \phi) = -D_{KL}(\hat{f}_{\mathcal{Z}|\mathcal{X}}(\mathbf{z}|\mathbf{x}_m; \theta) || f_{\mathcal{Z}}(\mathbf{z})) + \mathbb{E}_{\sim f_{\mathcal{Z}|\mathcal{X}}(\mathbf{z}|\mathbf{x}_m; \theta)} (\log [\hat{f}_{\mathcal{X}|\mathcal{Z}}(\mathbf{x}_m|\mathbf{z}; \phi)]) \quad (14)$$

1. Σχεδίαση και Εκπαίδευση VAE:

- Δημιουργήθηκε ένας VAE με τρία επίπεδα τόσο στον κωδικοποιητή όσο και στον αποκωδικοποιητή.
- Ο κωδικοποιητής μετατρέπει μια είσοδο 784-διαστάσεων σε έναν λανθάνοντα χώρο δύο διαστάσεων (latent size = 2) μέσω μιας αλληλουχίας γραμμικών επιπέδων με ενεργοποίηση ReLU.
- Ο αποκωδικοποιητής επανακατασκευάζει την είσοδο από το λανθάνοντα χώρο, τελειώνοντας με sigmoid για παραγωγή τιμών μεταξύ 0 και 1.
- Η εκπαίδευση διεξήχθη για 100 εποχές με batch size 250, χρησιμοποιώντας τη δυαδική εντροπία συνδυαστικά με τον KL όρο ως συνάρτηση κόστους.

Ο κώδικας δημιουργεί και εκπαιδεύει έναν VAE με latent size 2, χρησιμοποιεί δυαδική εντροπία και KL divergence ως συνάρτηση κόστους. Επίσης:

- generate_reconstructions(epoch):** Παράγει εικόνες από fixed noise σε συγκεκριμένες εποχές.
- visualize_latent_space(test_loader):** Απεικονίζει τον λανθάνοντα χώρο με scatter plot και εκτυπώνει στατιστικά για κάθε κατηγορία ψηφίων.

Αποτελέσματα

(a') Εξέλιξη Ανακατασκευών από Θόρυβο:



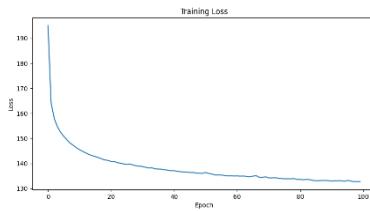
- Epoch 1:** Οι εικόνες που παράχθηκαν από τον αποκωδικοποιητή μέσω τυχαίου θορύβου ήταν ασαφείς και μη αναγνωρίσιμες.



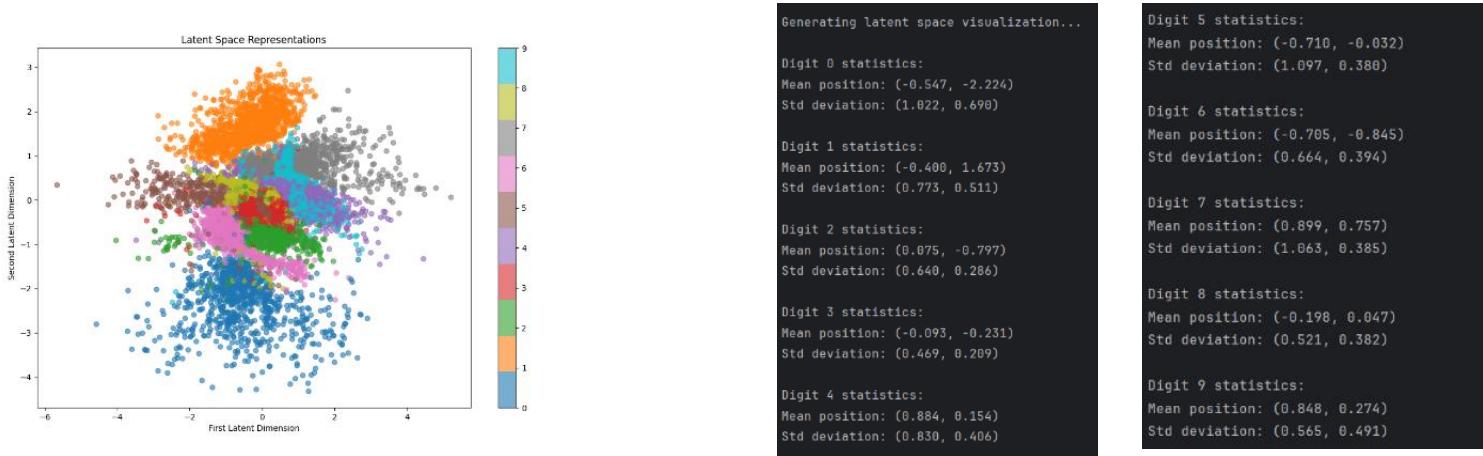
- **Epoch 50:** Οι ανακατασκευές από το ίδιο noise batch βελτιώθηκαν σημαντικά, παρουσιάζοντας περισσότερο δομημένες και αναγνωρίσιμες μορφές ψηφίων.



- **Epoch 100:** Οι εικόνες είναι αρκετά καθαρές και μοιάζουν με έγκυρα ψηφία, αποδεικνύοντας ότι ο αποκωδικοποιητής έχει μάθει να μετατρέπει τυχαία δείγματα από $N(0, I)$ σε ρεαλιστικές εικόνες MNIST.



(6') Λανθάνουσα Αναπαράσταση των Δεδομένων Ελέγχου:



- Το scatter plot του λανθάνοντα χώρου (με διαστάσεις 2) δείχνει πώς οι διάφοροι αριθμοί κατηγοριοποιούνται σε διαφορετικές περιοχές.
- Για κάθε ψηφίο, οι μέσες θέσεις και οι τυπικές αποκλίσεις δείχνουν τη συγκέντρωση και την ποικιλία των σημείων στον χώρο.

Παραδείγματα:

- **Digit 0:** Μέση θέση (-0.547, -2.224), με αστάθεια κατά μήκος της πρώτης διάστασης.
- **Digit 1:** Μέση θέση (-0.400, 1.673), με μικρότερη διασπορά στη δεύτερη διάσταση.
- **Άλλα ψηφία:** Δείχνουν ιδιαίτερες κατευθύνσεις και διασπορές, υποδεικνύοντας ομαδικές κατανομές.

ΑΠΟΔΕΙΞΕΙΣ ΑΡΧΕΙΟΥ:

Για να αρχίσουμε, ας υποθέσουμε ότι θέλουμε να λύσουμε το πρόβλημα για $L = 1$. Αποδείξτε ότι η επιθυμητή λύση προκύπτει από τη λύση του ακόλουθου με περιορισμούς προβλήματος βελτιστοποίησης:

$$\begin{aligned} & \max v_1^T S v_1 \\ & \text{subject to: } \|v_1\|_2 = 1 \end{aligned}$$

Αποδείξτε επίσης ότι η λύση του παραπάνω προβλήματος είναι το ιδιοδιάνυσμα v_1 που αντιστοιχεί στην μέγιστη ιδιοτιμή (έστω λ_1) του μη τρώου S_3 .

Πρόβλημα:

Προσπαθούμε να βρούμε τη "κατεύθυνση" στην οποία τα δεδομένα μας είναι πιο "διασκορπισμένα". Μαθηματικά, αυτό σημαίνει ότι θέλουμε να μεγιστοποιήσουμε τη διασπορά των δεδομένων όταν τα προβάλλουμε σε αυτή την κατεύθυνση.

Ψάχνουμε για εκείνη τη γραμμή στην οποία τα δεδομένα διασκορπίζονται όσο περισσότερο. Η συνάρτηση $v_1^T S v_1$ μετρά πόση διασπορά υπάρχει κατά μήκος της γραμμής που ορίζεται από το v_1 .

Έτσι, μεγιστοποιώντας $v_1^T S v_1$ υπό τη συνθήκη $\|v_1\|_2 = 1$, βρίσκουμε ακριβώς αυτήν τη μέγιστη διασπορά.

Η λύση σε αυτό το πρόβλημα δίνει το πρώτο κύριο συστατικό: το v_1 που αντιστοιχεί στη μέγιστη ιδιοτιμή του S .

Για $L=1$, πρέπει να βρούμε το διάνυσμα v_1 τέτοιο ώστε να μεγιστοποιείται η διακύμανση των προβολών:

$$\max v_1^T S v_1 \text{ με } \|v_1\|_2 = 1$$

Απόδειξη:

1. Καθορισμός Διακύμανσης Προβολής:

- Η προβολή κάθε x_m πάνω στον άξονα v_1 δίνεται από $v_1^T x_m$
- Η διακύμανση των προβολών είναι:

$$\text{Var}(v_1^T x) = 1/M \sum_{i=1}^M (v_1^T x_i)^2 = v_1^T (1/M \sum_{i=1}^M x_i x_i^T) v_1 = v_1^T S v_1$$

2. Μέγιστη Διακύμανση υπό Περιορισμό:

- Θέλουμε να επιλέξουμε v_1 με $\|v_1\|_2 = 1$ ώστε να μεγιστοποιείται $v_1^T S v_1$
- Το πρόβλημα μεγιστοποίησης με περιορισμό μπορεί να διατυπωθεί ως:

$$\max v_1^T S v_1 \text{ με } \|v_1\|_2 = 1$$

3. Χρήση Μεθόδου Lagrange:

- Ορίζουμε τη Lagrange συνάρτηση:

$$\mathcal{L}(v_1, \lambda) = v_1^T S v_1 - \lambda(v_1^T v_1 - 1)$$
- Παίρνουμε την παράγωγο ως προς v_1 και θέτουμε ίση με το μηδέν:

$$\nabla_{v_1} \mathcal{L} = 2Sv_1 - 2\lambda v_1 = 0 \Rightarrow Sv_1 = \lambda v_1$$
- Έτσι, v_1 είναι ιδιοδιάνυσμα του S και η τιμή Lagrange λ είναι η αντίστοιχη ιδιοτιμή.

4. Μέγιστη Διακύμανση:

- Για τη μέγιστη διακύμανση, πρέπει να επιλέξουμε το v_1 που δίνει τη μεγαλύτερη τιμή $v_1^T S v_1$
- Αυτό επιτυγχάνεται όταν λ είναι η μεγαλύτερη ιδιοτιμή του S , δηλαδή v_1 είναι το ιδιοδιάνυσμα του S με τη μεγαλύτερη ιδιοτιμή.

Συμπέρασμα:

Η λύση του προβλήματος μεγιστοποίησης $\max \|v_1\| = 1$ $v_1^T S v_1$ είναι το ιδιοδιάνυσμα v_1^* του πίνακα S που αντιστοιχεί στην μεγαλύτερη ιδιοτιμή. Αυτό το v_1^* είναι η πρώτη κύρια συνιστώσα των δεδομένων σύμφωνα με την PCA.

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, με περιορισμούς προβλήματος δελτιστοποίησης: $\max_{v_2} v_2^T S v_2$ subject to: $\|v_2\|_2 = 1 = 0$

Όταν $L=2$, θέλουμε να βρούμε **δύο κύριες συνιστώσες** (διάνυσμα v_1 και διάνυσμα v_2) που μεγιστοποιούν τη διακύμανση των δεδομένων όταν προβάλλονται σε έναν υποχώρο δύο διαστάσεων. Για τη δεύτερη κύρια συνιστώσα (v_2), ισχύουν οι εξής περιορισμοί:

1. Το v_2 πρέπει να είναι μοναδιαίο ($\|v_2\|_2 = 1$).
2. Το v_2 πρέπει να είναι ορθογώνιο στο πρώτο κύριο διάνυσμα (v_1^*), δηλαδή $\langle v_1^*, v_2 \rangle = 0$.

Δοσμένα

1. Πίνακας συνδιασπορών S :

- Περιγράφει πώς σχετίζονται οι διαστάσεις των δεδομένων.
- Είναι συμμετρικός και ημιθετικά ορισμένος ($S \geq 0$).

2. Περιορισμός για v_2 :

- Θέλουμε να μεγιστοποιήσουμε το $v_2^T S v_2$, που είναι η διακύμανση της προβολής στον άξονα v_2 , υπό τους περιορισμούς:
 - $\|v_2\|_2 = 1$
 - $\langle v_1^*, v_2 \rangle = 0$ (ορθογωνιότητα)

Απόδειξη

1. Υπενθύμιση από την Περίπτωση $L=1$

- Στην περίπτωση $L=1$, βρήκαμε ότι η πρώτη κύρια συνιστώσα v_1^* είναι το ιδιοδιάνυσμα του S που αντιστοιχεί στη μεγαλύτερη ιδιοτιμή (λ_1).
- Αυτό συμβαίνει γιατί μεγιστοποιεί τη διακύμανση $v_1^T S v_1$ υπό τον περιορισμό $\|v_1\|_2 = 1$.

2. Υποχώρος Ορθογωνιότητας

- Ορίζουμε τον υποχώρο U , ο οποίος περιλαμβάνει όλα τα διανύσματα v που είναι ορθογώνια προς v_1^* :
$$U = \{v \in \mathbb{R}^N \mid \langle v_1^*, v \rangle = 0\}$$
- Το U είναι ο πλήρης ορθοσυμπληρωματικός υποχώρος του v_1^* .

3. Μέγιστη Διακύμανση στον Υπόχωρο U

- Περιορίζουμε το πρόβλημα μεγιστοποίησης στον υποχώρο U :
 - Βρίσκουμε το v_2 που μεγιστοποιεί το $v_2^T S v_2$, υπό τον περιορισμό $v_2 \in U$ και $\|v_2\|_2 = 1$.
 - Μέσα στον U , ο S παραμένει συμμετρικός και ημιθετικά ορισμένος.
 - Η μέγιστη διακύμανση στον U επιτυγχάνεται από το ιδιοδιάνυσμα του S με τη δεύτερη μεγαλύτερη ιδιοτιμή (λ_2).

4. Περιορισμός Ορθογωνιότητας

- Ο περιορισμός $\langle v_1^*, v_2 \rangle = 0$ εξασφαλίζει ότι το v_2 δεν περιέχει καμία συνιστώσα στην κατεύθυνση του v_1^* .
- Έτσι, η μεγιστοποίηση της διακύμανσης γίνεται μόνο στις διαστάσεις που είναι ανεξάρτητες από το v_1^* .

5. Συμπέρασμα

- Η λύση του προβλήματος μεγιστοποίησης $\max\{v_2\} v_2^T S v_2$, υπό τους περιορισμούς $\|v_2\|_2 = 1$ και $\langle v_1^*, v_2 \rangle = 0$, είναι το ιδιοδιάνυσμα v_2^* του S που αντιστοιχεί στη δεύτερη μεγαλύτερη ιδιοτιμή (λ_2).

Γενική Εικόνα

1. Η πρώτη κύρια συνιστώσα v_1^* μεγιστοποιεί τη διακύμανση.
2. Η δεύτερη κύρια συνιστώσα v_2^* :
 - Είναι ορθογώνια προς την πρώτη ($\langle v_1^*, v_2^* \rangle = 0$).
 - Μεγιστοποιεί τη διακύμανση στον υποχώρο που δεν περιέχει το v_1^* .

To προβλήμα μας τώρα μπορεί να διατυπωθεί ως το ακόλουθο πρόβλημα δελτιστοποίησης : $\min A //S - A//2 F$ subject to: $\text{rank}\{A\}=L$ óπου $\|C\|/F$, η Frobenius στάθμη των μητρώου C

4. Αποδείξτε ότι η λύση του παραπάνω προβλήματος ελαχιστοποίησης είναι η ακόλουθη: $A \star = V_L \Sigma_L V_L T$ óπου: • V_L οι πρώτες L στήλες του μητρώου V και • Σ_L ένα $L \times L$ διαγώνιο μητρώο που περιέχει τις πρώτες L τιμές της διαγωνίου του μητρώου Σ_0

Θέλουμε να βρούμε μια προσέγγιση χαμηλού βαθμού (rank L) του πίνακα συνδιασπορών $S = XX^T$, η οποία ελαχιστοποιεί την **Frobenius απόσταση** μεταξύ του S και της προσέγγισης. Η Frobenius απόσταση $\|S - A\|^2$ είναι η συνολική τετραγωνική διαφορά μεταξύ των στοιχείων των πινάκων S και A .

Δοσμένα:

1. **SVD του πίνακα X:** Ο πίνακας X μπορεί να γραφτεί ως:

$$X = V \Sigma U^T$$
 όπου:
 - Σ είναι ένας διαγώνιος πίνακας διαστάσεων $N \times M$, με τις ιδιάζουσες τιμές s_k στη διαγώνιο.
 - V είναι ένας ορθοκανονικός πίνακας $N \times N$, του οποίου οι στήλες είναι τα αριστερά ιδιάζοντα διανύσματα του X .
 - U είναι ένας ορθοκανονικός πίνακας $M \times M$, του οποίου οι στήλες είναι τα δεξιά ιδιάζοντα διανύσματα του X .
 -
2. **Πίνακας συνδιασπορών S:** Από την SVD του X , ο S γράφεται ως:

$$S = XX^T = V \Sigma \Sigma^T V^T$$

Ο πίνακας $\Sigma \Sigma^T$ είναι διαγώνιος και περιέχει τα τετράγωνα των ιδιάζουσων τιμών του X .

Πρόβλημα Βελτιστοποίησης:

Θέλουμε να λύσουμε:

$$\min\{A\} \|S - A\|^2, \text{ με περιορισμό } \text{rank}(A) = L$$

- Το A πρέπει να έχει βαθμό (rank) L , δηλαδή να εξαρτάται μόνο από L γραμμικά ανεξάρτητες συνιστώσες.

Απόδειξη:

1. **Frobenius Norm:**
 Η Frobenius απόσταση $\|S - A\|^2$ δίνεται από:

$$\|S - A\|^2 = \sum_{ij} (s_{ij} - a_{ij})^2$$

Το πρόβλημα είναι να επιλέξουμε Α έτσι ώστε να ελαχιστοποιήσουμε αυτή την ποσότητα.

2. Θεώρημα Eckart-Young-Mirsky:

Το θεώρημα Eckart-Young-Mirsky δηλώνει ότι η καλύτερη προσέγγιση χαμηλού βαθμού (rank L) ενός πίνακα, σύμφωνα με το Frobenius norm, προκύπτει αν διατηρήσουμε τις L μεγαλύτερες ιδιάζουσες τιμές του πίνακα και μηδενίσουμε τις υπόλοιπες.

3. Εφαρμογή στο S:

- Από την SVD του $S = V\Sigma V^T$, ο πίνακας Σ περιέχει τις ιδιοτιμές του S (που είναι τα τετράγωνα των ιδιάζουσων τιμών του X).
- Επιλέγοντας τις L μεγαλύτερες τιμές από τη διαγώνιο του Σ και μηδενίζοντας τις υπόλοιπες, έχουμε:

$$A^* = V_1 \Sigma_1 V_1^T$$

όπου:

- V_1 περιέχει τις πρώτες L στήλες του V (τα L πρώτα ιδιοδιανύσματα).
- Σ_1 είναι ένας διαγώνιος πίνακας $L \times L$ με τις L μεγαλύτερες ιδιοτιμές.

4. Ιδιότητες της Βέλτιστης Λύσης:

- Το A^* έχει βαθμό (rank) L γιατί εξαρτάται μόνο από L ιδιοδιανύσματα.
- Η Frobenius απόσταση $\|S - A^*\|_F^2$ είναι η ελάχιστη δυνατή, επειδή διατηρούμε μόνο τις συνιστώσες με τη μεγαλύτερη διακύμανση (μέσω των μεγαλύτερων ιδιοτιμών).

Συμπέρασμα:

Η βέλτιστη προσέγγιση χαμηλού βαθμού A^* του S είναι:

$$A^* = V_1 \Sigma_1 V_1^T$$

όπου:

- V_1 είναι ο πίνακας με τις πρώτες L στήλες του V,
- Σ_1 είναι ο διαγώνιος πίνακας με τις πρώτες L μεγαλύτερες ιδιοτιμές του S.

Αυτή η προσέγγιση διατηρεί τη μέγιστη δυνατή πληροφορία (διακύμανση) του S, ενώ μειώνει τον βαθμό του πίνακα στο L.

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

Δοσμένο:

Έστω $A \in \mathbb{R}^{mn}$ ένας πίνακας με στοιχεία a_{ij} .

Θέλουμε να αποδείξουμε:

Η Frobenius στάθμη του πίνακα A, δηλαδή

$$\|A\|_F = \sqrt{(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2)}$$

είναι ίση με την ℓ_2 στάθμη του διανύσματος που προκύπτει από την στοίχιση όλων των γραμμών ή όλων των στήλων του A.

Απόδειξη:

1. Ορισμός της Frobenius Στάθμης:

Η Frobenius στάθμη του πίνακα A είναι ορίζεται ως:

$$\|A\|_F = \sqrt{(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2)}$$

Αυτή είναι η ρίζα του αθροίσματος των τετραγώνων όλων των στοιχείων του A.

2. Δημιουργία Διανύσματος από τις Γραμμές ή τις Στήλες του A:

- Μπορούμε να δημιουργήσουμε ένα διάνυσμα $\text{vec}(A) \in \mathbb{R}^{mn}$ τοποθετώντας όλα τα στοιχεία του πίνακα A σε μία διάσταση (vectorization). Για παράδειγμα:
$$\text{vec}(A) = [a_{11} \ a_{12} \ \dots \ a_{1n} \ a_{21} \ a_{22} \ \dots \ a_{mn}]^T$$
- Το $\text{vec}(A)$ είναι το διάνυσμα που προκύπτει από την στοίχιση όλων των στοιχείων του A.

3. Υπολογισμός της ℓ_2 Στάθμης του $\text{vec}(A)$:

Η ℓ_2 στάθμη του διανύσματος $\text{vec}(A)$ είναι:

$$\|\text{vec}(A)\|_2 = \sqrt{(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2)}$$

4. Σύγκριση της Frobenius Στάθμης με την ℓ_2 Στάθμη:

Παρατηρούμε ότι οι τύποι για τη Frobenius στάθμη $\|A\|^f$ και την ℓ_2 στάθμη $\|\text{vec}(A)\|_2$ είναι απολύτως ταυτόσημοι:

$$\|A\|^f = \sqrt{(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2)} = \|\text{vec}(A)\|_2$$

Συμπέρασμα:

Αποδείξαμε ότι η Frobenius στάθμη ενός πίνακα A είναι ίση με την ℓ_2 στάθμη του διανύσματος που προκύπτει από την στοίχιση όλων των στοιχείων του A .

Εφαρμογή:

Αυτό το αποτέλεσμα χρησιμοποιείται συχνά στη γραμμική άλγεβρα και στην αριθμητική ανάλυση για να συνδέουμε πίνακες και διανύσματα, καθώς η Frobenius στάθμη ενός πίνακα είναι η φυσική επέκταση της ℓ_2 στάθμης από διανύσματα σε πίνακες.

Είναι προφανές (αλλά ωστόσο αποδείξτε το) ότι ισχύουν οι ακόλουθες σχέσεις:

Μέση τιμή:

$$E\{X\} = E\{WZ + \mu + W\} = \mu$$

Πίνακας συνδιασποράς:

$$\begin{aligned} \Sigma_{xx} &= E\{(WZ + W)(WZ + W)^T\} \\ &= WE\{ZZ^T\}W^T + E\{WW^T\} \\ &= WW^T + \sigma^2 I \end{aligned}$$

Συνάρτηση πυκνότητας πιθανότητας της τ.μ. X :

$$f_x(x) = N(x; \mu, \Sigma_{xx})$$

Απόδειξη των Σχέσεων στην Πιθανοτική PCA

Δοσμένα:

Λανθάνουσα μεταβλητή Z με κατανομή $f_Z(z) = N(z; 0, I)$

Υπό συνθήκη κατανομή X δεδομένου $Z=z$:

$$f_{X|Z}(x|z) = N(x; Wz + \mu, \sigma^2 I)$$

Αυτό υποδηλώνει ότι:

$$X = WZ + \mu + \varepsilon$$

όπου $\varepsilon \sim N(0, \sigma^2 I)$ είναι ανεξάρτητος θόρυβος.

Θέλουμε να δείξουμε:

$$E\{X\} = \mu$$

$$\Sigma_{XX} = \text{Cov}(X) = WW^T + \sigma^2 I$$

$$\text{Καταληκτικά, } f_X(x) = N(x; \mu, \Sigma_{XX})$$

Απόδειξη:

1. Υπολογισμός $E\{X\}$:

Δεδομένου ότι:

$$X = WZ + \mu + \varepsilon$$

και γνωρίζοντας ότι:

$$E\{Z\} = 0 \text{ και } E\{\varepsilon\} = 0$$

υπολογίζουμε:

$$E\{X\} = E\{WZ + \mu + \varepsilon\} = W \cdot E\{Z\} + \mu + E\{\varepsilon\} = W \cdot 0 + \mu + 0 = \mu$$

Άρα, $E\{X\} = \mu$

2. Υπολογισμός Σ_{XX} (Συνδιασποράς):

Η συνδιασπορά $\Sigma_{XX} = Cov(X)$ δίνεται από:

$$\Sigma_{XX} = E\{(X - E\{X\})(X - E\{X\})^T\}$$

Αφού $E\{X\} = \mu$, έχουμε:

$$X - \mu = WZ + \varepsilon$$

Τότε:

$$\begin{aligned}\Sigma_{XX} &= E\{(WZ + \varepsilon)(WZ + \varepsilon)^T\} \\ &= E\{WZ(WZ)^T\} + E\{WZ\varepsilon^T\} + E\{\varepsilon(WZ)^T\} + E\{\varepsilon\varepsilon^T\}\end{aligned}$$

Λόγω της ανεξαρτησίας του Z και ε και ότι $E\{\varepsilon\} = 0$, τα διασταυρωμένα $E\{WZ\varepsilon^T\}$ και $E\{\varepsilon(WZ)^T\}$ είναι μηδέν. Επίσης, $E\{WZ(WZ)^T\} = W \cdot E\{ZZ^T\} \cdot W^T$.

Επειδή $Z \sim N(0, I)$, έχουμε $E\{ZZ^T\} = I$. Επιπλέον, $E\{\varepsilon\varepsilon^T\} = \sigma^2 I$.

Άρα:

$$\begin{aligned}\Sigma_{XX} &= W \cdot E\{ZZ^T\} \cdot W^T + E\{\varepsilon\varepsilon^T\} \\ &= WIW^T + \sigma^2 I \\ &= WW^T + \sigma^2 I\end{aligned}$$

3. Κατανομή X :

Η παραπάνω απόδειξη μας δίνει:

$$E\{X\} = \mu \text{ και } \Sigma_{XX} = WW^T + \sigma^2 I$$

Αυτό σημαίνει ότι η κατανομή της τυχαίας μεταβλητής X είναι κανονική (Gaussian) με μέση τιμή μ και συνδιασπορά Σ_{XX} , δηλαδή:

$$f_X(x) = N(x; \mu, WW^T + \sigma^2 I)$$

Συμπέρασμα:

Μέσω της πιθανοτικής προσέγγισης της PCA, αποδείξαμε ότι η κατανομή των δεδομένων X είναι $N(x; \mu, WW^T + \sigma^2 I)$, έχοντας υπολογίσει τόσο τη μέση τιμή όσο και τη συνδιασπορά με βάση τις παραδοχές για τη λανθάνουσα μεταβλητή Z και τον θόρυβο ε .

Αν γνωρίζουμε την σππ $f_X(x)$, τότε από το θεώρημα του Bayes εύκολα βρίσκουμε ότι:

$$f_{\{Z|X\}}(z|x) = f_{\{X|Z\}}(x|z)f_Z(z)/f_X(x)$$

όπου $f_{\{X|Z\}}(x|z)$ η συνάρτηση πιθανοφάνειας (*likelihood*).

Αποδείξτε ότι η παραπάνω υπό συνθήκη σππ, με δεδομένο αντό της Σχέσης (4), δίνεται από την ακόλουθη σχέση:

$$f_{\{Z|X\}}(z|x) = N(z; \Sigma_I^{-1}W^T(x-\mu), \sigma^2\Sigma_I^{-1})$$

όπου $\Sigma_I = W^TW + \sigma^2I$.

Αξιοσημείωτο είναι ότι η εκ των νστέρων αναμενόμενη τιμή εξαρτάται από το x ενώ το εκ των νστέρων μητρώο συνδιασπορών όχι.

Δεδομένα του Μοντέλου

- **Προτεραιότητα (prior):**

$$f_Z(z) = N(z; 0, I_L)$$

- **Συνθήκη (likelihood):**

$$f_{\{X|Z\}}(x|z) = N(x; Wz + \mu, \sigma^2 I_N)$$

- **Σύμφωνα με το Bayes:**

$$f_{\{Z|X\}}(z|x) = f_{\{X|Z\}}(x|z) \cdot f_Z(z)/f_X(x)$$

όπου $f_X(x)$ είναι ένας παραγοντοποιητικός σταθερός όρος όσον αφορά το z (κανονικοποίηση).

1. Γράφουμε τους Όρους

Αρχικά εκφράζουμε τις πυκνότητες:

Likelihood:

$$f_{\{X|Z\}}(x|z) = (2\pi\sigma^2)^{-N/2} \exp(-\frac{1}{2}\sigma^{-2}(x-\mu-Wz)^T(x-\mu-Wz))$$

Prior:

$$f_Z(z) = (2\pi)^{-L/2} \exp(-\frac{1}{2}z^Tz)$$

2. Υπολογισμός των Γινομένου $f_{\{X|Z\}}(x|z)f_Z(z)$

$$f_{\{X|Z\}}(x|z)f_Z(z) \propto \exp(-\frac{1}{2}\sigma^{-2}(x-\mu-Wz)^T(x-\mu-Wz) - \frac{1}{2}z^Tz)$$

3. Επεξεργασία του Εκθετικού Όρου

$$-\frac{1}{2}\sigma^{-2}(x-\mu-Wz)^T(x-\mu-Wz) - \frac{1}{2}z^Tz = -\frac{1}{2}\sigma^{-2}[(x-\mu)^T(x-\mu) - 2(x-\mu)^TWz + z^TW^TWz] - \frac{1}{2}z^Tz$$

Οργανώνουμε τους όρους που εξαρτώνται από το z :

$$-\frac{1}{2}z^T(W^TW/\sigma^2 + I_L)z + \sigma^{-2}(x-\mu)^TWz + σταθερός όρος$$

4. Ολοκλήρωση Τετραγώνου

Θέλουμε να φέρουμε την παράσταση στη μορφή:

$$-\frac{1}{2}(z-m)^T A(z-m) + \text{σταθερός όρος}$$

όπου:

- $A = W^T W / \sigma^2 + I_L L$
- $Am = \sigma^{-2} W^T (x - \mu)$
- άρα: $m = A^{-1} \sigma^{-2} W^T (x - \mu)$

5. Υπολογισμός A^{-1} και Ταυτότητας με Σ_1

Παρατηρούμε ότι:

$$A = W^T W / \sigma^2 + I_L L = (W^T W + \sigma^2 I_L L) / \sigma^2$$

Ορίζοντας:

$$\Sigma_1 = W^T W + \sigma^2 I_L L$$

είναι:

$$A = \Sigma_1 / \sigma^2 \quad A^{-1} = \sigma^2 \Sigma_1^{-1}$$

6. Υπολογισμός των Μέσου m και της Συμβατικής Matrix

$$m = A^{-1} \sigma^{-2} W^T (x - \mu) = \sigma^2 \Sigma_1^{-1} \sigma^{-2} W^T (x - \mu) = \Sigma_1^{-1} W^T (x - \mu)$$

Η συμβατική μήτρα της posterior κατανομής:

$$\text{Cov}(Z|X=x) = \sigma^2 A^{-1} = \sigma^2 (\sigma^2 \Sigma_1^{-1}) = \sigma^2 \Sigma_1^{-1}$$

7. Συμπέρασμα για την Posterior Κατανομή

$$f_{\{Z|X\}}(z|x) = N(z; \Sigma_1^{-1} W^T (x - \mu), \sigma^2 \Sigma_1^{-1})$$

$$\text{όπου } \Sigma_1 = W^T W + \sigma^2 I_L L$$

Παρατήρηση

- **Μέση Τιμή Posterior:**

$$E\{Z|X=x\} = \Sigma_1^{-1} W^T (x - \mu)$$

εξαρτάται από την παρατήρηση x.

- **Συμβατική Matrix Posterior:**

$$\text{Cov}(Z|X) = \sigma^2 \Sigma_1^{-1}$$

δεν εξαρτάται από x — είναι σταθερή για όλες τις παρατηρήσεις.

To Kullback-Leibler Divergence (KLD) είναι ένα πολύ γνωστό μέτρο ομοιότητας (similarity measure), το οποίο χρησιμοποιείται στην μέτρηση της ομοιότητας δύο κατανομών.

To KLD για τις συναρτήσεις πυκνότητας πιθανότητας $p_X(x)$ και $q_X(x)$ ορίζεται ως ακολούθως:

$$D_{KL}(p_X(x) // q_X(x)) = E_{\sim p_X} \log [p_X(x)/q_X(x)] \\ = \int p_X(x) \log [p_X(x)/q_X(x)] dx$$

Παρατηρήστε ότι:

$$D_{KL}(p_X(x) // q_X(x)) \geq 0$$

με την ισότητα να ισχύει όταν $p_X(x) = q_X(x)$ (αποδείξτε το).

Απόδειξη της Μη Αρνητικότητας του Kullback-Leibler Divergence και Εφαρμογές

Μη αρνητικότητα του Kullback-Leibler Divergence (KLD)

Ορισμός του D_{KL} :

Ο D_{KL} μεταξύ δύο κατανομών $p_X(x)$ και $q_X(x)$ ορίζεται ως:

$$D_{KL}(p_X(x) // q_X(x)) = \int p_X(x) \log [p_X(x)/q_X(x)] dx$$

Στόχος:

Θέλουμε να δείξουμε ότι:

$$D_{KL}(p_X // q_X) \geq 0$$

$D_{KL}(p_X // q_X) = 0$ αν και μόνο αν $p_X(x) = q_X(x)$ σχεδόν παντού

Χρήση της Ανισότητας του Gibbs:

Η ανισότητα του Gibbs λέει ότι:

$$\log t \leq t - 1 \text{ για κάθε } t > 0$$

με ισότητα αν και μόνο αν $t = 1$.

Αν θέσουμε $t = q_X(x)/p_X(x)$, τότε:

$$\log [q_X(x)/p_X(x)] \leq q_X(x)/p_X(x) - 1$$

Εφαρμογή στο D_{KL} :

$$D_{KL}(p_X(x) // q_X(x)) = -\int p_X(x) \log [q_X(x)/p_X(x)] dx$$

Από την ανισότητα του Gibbs και ολοκληρώνοντας:

$$-\int p_X(x) \log [q_X(x)/p_X(x)] dx \geq \\ -\int p_X(x) (q_X(x)/p_X(x) - 1) dx$$

2. Χρήση του D_{KL} για Έλεγχο Ανεξαρτησίας

Στόχος:

Θέλουμε να ελέγξουμε αν δύο τυχαίες μεταβλητές X και Y είναι ανεξάρτητες.

Οι X και Y είναι ανεξάρτητες αν: $f_{\{X,Y\}}(x,y) = f_X(x)f_Y(y) \forall x,y$

Χρήση του D_{KL} :

Ορίζουμε:

$$p_{\{X,Y\}}(x,y) = f_{\{X,Y\}}(x,y)$$

$$q_{\{X,Y\}}(x,y) = f_X(x)f_Y(y)$$

Σύνδεση με την Αμοιβαία Πληροφορία:

Η ποσότητα ονομάζεται αμοιβαία πληροφορία $I(X;Y)$ μεταξύ των X και Y :

$$I(X;Y) = D_{KL}(f_{\{X,Y\}}(x,y) \parallel f_X(x)f_Y(y))$$

Ιδιότητες της αμοιβαίας πληροφορίας:

$$I(X;Y) \geq 0$$

$I(X;Y) = 0$ αν και μόνο αν $f_{\{X,Y\}}(x,y) = f_X(x)f_Y(y)$, δηλαδή οι X και Y είναι ανεξάρτητες

Συμπέρασμα:

Ο $D_{KL}(p_X \parallel q_X) \geq 0$, με ισότητα όταν $p_X = q_X$ σχεδόν παντού.

Για έλεγχο ανεξαρτησίας, ορίζουμε: $D_{KL}(f_{\{X,Y\}} \parallel f_X f_Y) = I(X;Y)$, που είναι μηδέν αν και μόνο αν οι μεταβλητές είναι ανεξάρτητες.