# ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

2η Σειρά Ασκήσεων

ΔΗΜΗΤΡΙΑΔΗΣ ΓΕΩΡΓΙΟΣ: ΑΜ – 5209
ΔΗΜΗΤΡΙΟΥ ΑΡΙΣΤΟΤΕΛΗΣ: ΑΜ - 5211

COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF IOANNINA

# Πίνακας περιεχομένων

## Προαπαιτούμενες ενέργειες για την εκτέλεση του κώδικα

Για την ορθή διεκπεραίωση του κώδικα χρειάστηκε να προσθέσουμε τις παρακάτω βιβλιοθήκες:

- import numpy as np
- from tensorflow.keras.datasets import fashion_mnist
- from sklearn.decomposition import PCA
- import matplotlib.pyplot as plt
- from tensorflow.keras.layers import Input, Dense
- from tensorflow.keras.models import Model
- from sklearn.model_selection import train_test_split
- from sklearn.neighbors import KNeighborsClassifier
- from sklearn.metrics import accuracy_score
- from sklearn.cluster import KMeans
- from sklearn.metrics import silhouette_score
- from sklearn.metrics import confusion_matrix
- from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score

# 1η Άσκηση

## *Α΄ Ερώτημα*

- Φόρτωση του συνόλου δεδομένων Fashion MNIST: Φορτώνεται το σύνολο δεδομένων Fashion MNIST από το TensorFlow Keras. Το Fashion MNIST περιέχει 60.000 εικόνες ρούχων για εκπαίδευση και 10.000 εικόνες για το σύνολο ελέγχου.

- Επιλογή τυχαίων εικόνων από το σύνολο εκπαίδευσης: Αυτό το τμήμα του κώδικα επιλέγει τυχαία έναν ίσο αριθμό εικόνων από κάθε κατηγορία του συνόλου εκπαίδευσης. Αυτός ο συμβολισμός np.random.seed(42) θέτει το seed της γεννήτριας τυχαίων αριθμών, εξασφαλίζοντας ότι οι επόμενες τυχαίες επιλογές θα είναι ίδιες κάθε φορά που εκτελείται ο κώδικας με τον ίδιο seed.

- Μετατροπή των εικόνων σε διανύσματα: Αυτή η γραμμή μετατρέπει κάθε εικόνα σε ένα μονοδιάστατο διάνυσμα. Αυτό είναι απαραίτητο για την εφαρμογή του PCA.

- Κανονικοποίηση των εικόνων: Εδώ γίνεται η κανονικοποίηση των εικόνων, διαιρώντας κάθε τιμή εικόνας με το μέγιστο δυνατό pixel value (255) για να κανονικοποιηθούν στο διάστημα [0, 1].

- Δημιουργία αντικειμένου PCA: Δημιουργεί ένα αντικείμενο PCA, το οποίο θα χρησιμοποιηθεί για τη μείωση της διαστατικότητας των δεδομένων.

- Εφαρμογή PCA: Εφαρμόζει το PCA στα κανονικοποιημένα δεδομένα, μετασχηματίζοντάς τα σε ένα νέο χώρο χαμηλότερης διαστατικότητας.

- Βρίσκουμε τον αριθμό των κύριων συνιστωσών: Αυτή η γραμμή εμφανίζει τον αριθμό των κύριων συνιστωσών που διατηρούν το 90% της διασποράς. Αυτός ο αριθμός κύριων συνιστωσών θα χρησιμοποιηθεί στη συνέχεια για τη μείωση της διαστατικότητας.

- Μείωση της διάστασης των δεδομένων με χρήση PCA: Εφαρμόζει το PCA στα δεδομένα για να μειώσει την διαστατικότητα τους στον αριθμό των κύριων συνιστωσών που βρέθηκε.

- Απεικόνιση των πρώτων 10 μειωμένων εικόνων: Αντιστρέφει τη μείωση διαστάσεων και απεικονίζει τις πρώτες 10 εικόνες μετά τη μείωση της διάστασης. Αυτό γίνεται για να δούμε πώς μοιάζουν οι εικόνες μετά τη μείωση της διαστατικότητας.

```
Η διάσταση του νέου χώρου προβολής είναι: 82
```

*Β΄ Ερώτημα*

- Ορισμός της αρχικής διάστασης d και της νέας διάστασης M: Σε αυτό το σημείο, καθορίζονται η αρχική διάσταση των δεδομένων d και η νέα διάσταση M.

- Ορισμός του μοντέλου Autoencoder: Το autoencoder είναι ένα δίκτυο νευρώνων που χρησιμοποιείται για την αυτόματη μείωση της διάστασης των δεδομένων. Εδώ, ορίζεται το μοντέλο του autoencoder με χρήση του Keras. Υπάρχουν δύο στάδια:

    - Κωδικοποίηση (encoding)

    - Αποκωδικοποίηση (decoding).

- Εκπαίδευση του μοντέλου Autoencoder: Το μοντέλο autoencoder εκπαιδεύεται με τη χρήση της μεθόδου fit για έναν αριθμό εποχών (epochs). Το loss function που χρησιμοποιείται είναι το binary crossentropy.

- Χρήση του encoder για τη μείωση της διάστασης των δεδομένων: Το εκπαιδευμένο μοντέλο autoencoder χρησιμοποιείται για να μειώσει τη διάσταση των δεδομένων εισόδου. Οι εικόνες εισόδου normalized_images προβάλλονται στον encoder, και τα αποτελέσματα της προβολής αποθηκεύονται στη μεταβλητή encoded_images.

```
Epoch 1/50
63/63 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - loss: 0.5321 - val_loss: 0.5285
Epoch 2/50
63/63 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3459 - val_loss: 0.4612
Epoch 3/50
63/63 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3143 - val_loss: 0.4119
Epoch 4/50
63/63 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3043 - val_loss: 0.3925
Epoch 5/50
63/63 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2993 - val_loss: 0.3765
Epoch 6/50
63/63 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2910 - val_loss: 0.3725
```

## *Γ΄ Ερώτημα*

- Διαχωρισμός των δεδομένων: Οι μειωμένες εικόνες από το PCA και το Autoencoder διαχωρίζονται σε σύνολο εκπαίδευσης και σύνολο ελέγχου, χρησιμοποιώντας τη συνάρτηση train_test_split. Το ποσοστό του συνόλου ελέγχου ορίζεται στο 20%, και η τυχαιότητα του διαχωρισμού εξασφαλίζεται με την παράμετρο random_state=42.

- Εκπαίδευση του μοντέλου k-NN για PCA: Εδώ, δημιουργείται και εκπαιδεύεται ένα μοντέλο k-NN χρησιμοποιώντας τον μειωμένο χώρο χαρακτηριστικών που προκύπτει από το PCA.

- Πρόβλεψη και υπολογισμός της ακρίβειας για PCA: Το μοντέλο k-NN που εκπαιδεύτηκε με χρήση του PCA χρησιμοποιείται για την πρόβλεψη των ετικετών του συνόλου ελέγχου PCA. Στη συνέχεια, υπολογίζεται η ακρίβεια της ταξινόμησης για το PCA.

- Εκπαίδευση του μοντέλου k-NN για Autoencoder: Προχωρούμε με τη δημιουργία και εκπαίδευση ενός μοντέλου k-NN χρησιμοποιώντας τον μειωμένο χώρο χαρακτηριστικών που προκύπτει από το Autoencoder.

- Πρόβλεψη και υπολογισμός της ακρίβειας για το Autoencoder: Το μοντέλο k-NN που εκπαιδεύτηκε με χρήση του Autoencoder χρησιμοποιείται για την πρόβλεψη των ετικετών του συνόλου ελέγχου Autoencoder. Στη συνέχεια, υπολογίζεται η ακρίβεια της ταξινόμησης για το Autoencoder.

```
Ακρίβεια ταξινόμησης με PCA: 0.8245
Ακρίβεια ταξινόμησης με Autoencoder: 0.8235
```

- Μείωση της διάστασης των δεδομένων με χρήση PCA: Αρχικά, τα δεδομένα εισόδου normalized_images περνούν μέσα από το μοντέλο Autoencoder για να προβλεφθούν οι μειωμένες αναπαραστάσεις τους στον μειωμένο χώρο χαρακτηριστικών. Στη συνέχεια, αυτές οι μειωμένες αναπαραστάσεις υπόκεινται σε ακόμα μία διαδικασία μείωσης διαστάσεων με χρήση του PCA. Συγκεκριμένα, χρησιμοποιείται η συνάρτηση PCA() με παράμετρο το 0.90, προκειμένου να διατηρηθεί το 90% της διακύμανσης των μειωμένων αναπαραστάσεων.

- Δημιουργία και εκπαίδευση του ταξινομητή k-NN: Τα δεδομένα που προκύπτουν από το PCA χρησιμοποιούνται για τον διαχωρισμό των δεδομένων σε σύνολο εκπαίδευσης και σύνολο ελέγχου. Ένας ταξινομητής k-NN (με k=5) δημιουργείται και εκπαιδεύεται χρησιμοποιώντας το σύνολο εκπαίδευσης.

- Αξιολόγηση του μοντέλου στα δεδομένα ελέγχου: Το μοντέλο αξιολογείται στο σύνολο ελέγχου, και η ακρίβεια της ταξινόμησης υπολογίζεται και εκτυπώνεται.

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 0s 771us/step
Ακρίβεια ταξινόμησης στον μειωμένο χώρο μετά την εφαρμογή PCA: 0.8155
```

Όπως μπορούμε να παρατηρήσουμε το accuracy μειώνεται!

# 2η Άσκηση

*Α΄ Ερώτημα*

- Εύρεση βέλτιστου Κ για τη μέθοδο PCA: Χρησιμοποιείται η μέθοδος K-Means clustering για να προσαρμοστεί στα μειωμένα δεδομένα που προέκυψαν από το PCA. Για κάθε τιμή του Κ στο εύρος K_range, υπολογίζεται το silhouette score. Η βέλτιστη τιμή του Κ επιλέγεται ως αυτή που μεγιστοποιεί το silhouette score.

- Εύρεση βέλτιστου Κ για τη μέθοδο Autoencoder: Πραγματοποιείται παρόμοια διαδικασία για τα δεδομένα που έχουν περάσει από το Autoencoder. Τα μειωμένα δεδομένα που προκύπτουν από το Autoencoder χρησιμοποιούνται για την εκτίμηση του K-Means clustering. Και πάλι, υπολογίζεται το silhouette score για κάθε τιμή του Κ στο εύρος K_range, και η βέλτιστη τιμή του Κ επιλέγεται ως αυτή που μεγιστοποιεί το silhouette score.

```
Η βέλτιστη τιμή Κ για τη μέθοδο PCA είναι: 13
Η βέλτιστη τιμή Κ για τη μέθοδο Autoencoder είναι: 13
```

*Β΄ Ερώτημα*

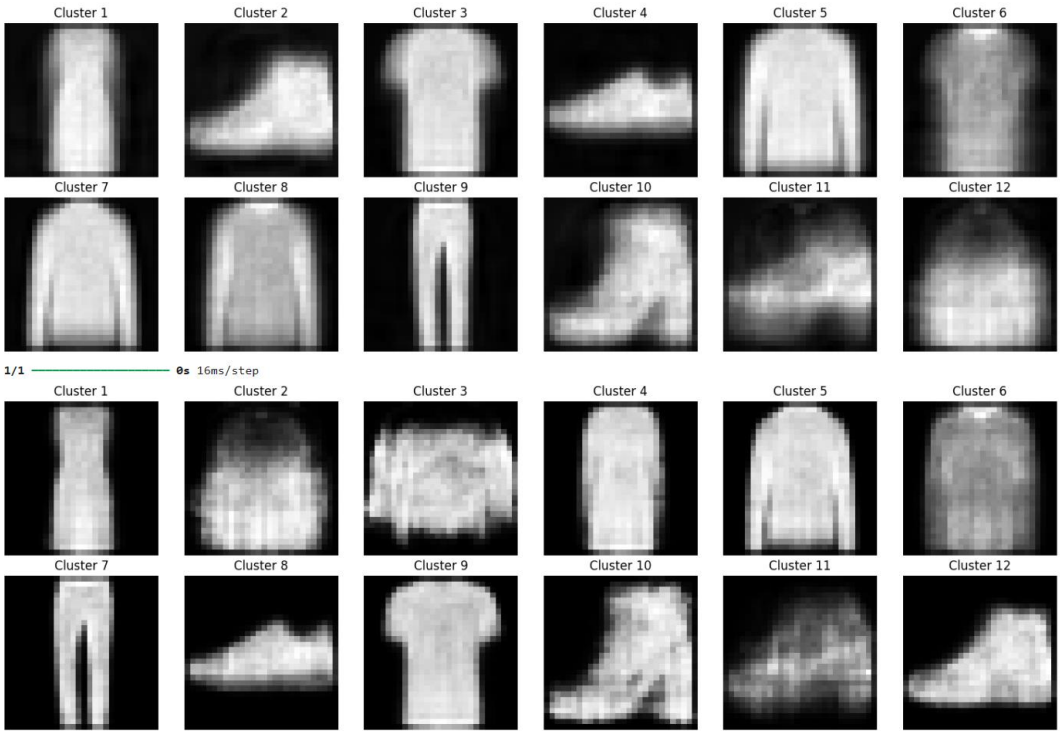
*Για το PCA:*

- Εύρεση των κέντρων των clusters: Χρησιμοποιείται ο αλγόριθμος K-Means για να ομαδοποιήσει τα δεδομένα σε clusters, χρησιμοποιώντας τη βέλτιστη τιμή Κ που προέκυψε από το προηγούμενο βήμα. Τα κέντρα των clusters καθορίζονται και αποθηκεύονται.
- Αντίστροφη μετατροπή PCA: Χρησιμοποιείται η αντίστροφη μετατροπή PCA για να μετατραπούν τα κέντρα των clusters από τον μειωμένο χώρο στον αρχικό χώρο των δεδομένων.
- Εμφάνιση των κέντρων των clusters: Δημιουργείται ένα γράφημα με τις εικόνες των κέντρων των clusters. Για κάθε εικόνα, προστίθεται ένας τίτλος που δείχνει τον αντίστοιχο αριθμό του cluster. Αυτό μας επιτρέπει να οπτικοποιήσουμε την τυπική εικόνα που αντιστοιχεί σε κάθε ομάδα, βοηθώντας μας να κατανοήσουμε καλύτερα τη δομή των ομάδων στον αρχικό χώρο των δεδομένων.

*Για το Autoencoder:*

Τα βήματα είναι ουσιαστικά τα ίδια με τα παραπάνω, με μια μικρή διαφοροποίηση στο δεύτερο βήμα:

- Αντίστροφη μετατροπή Autoencoder: Χρησιμοποιείται η αντίστροφη μετατροπή του Autoencoder για να μετατραπούν τα κέντρα των clusters από τον μειωμένο χώρο του Autoencoder στον αρχικό χώρο των δεδομένων.




*Γ΄ Ερώτημα*

- Υπολογισμός καθαρότητας (purity):

  - Για κάθε cluster, βρίσκονται οι πραγματικές ετικέτες των δεδομένων που ανήκουν σε αυτό το cluster.
  - Δημιουργείται ο πίνακας σύγχυσης (confusion matrix) για το κάθε cluster, ο οποίος δείχνει πόσα δείγματα ανήκουν σε κάθε κλάση.

- ❖ Η καθαρότητα για κάθε cluster υπολογίζεται ως ο λόγος του μέγιστου αριθμού στοιχείων σε μια κλάση προς τον συνολικό αριθμό των στοιχείων στο cluster.
- ❖ Τέλος, η καθαρότητα του κάθε cluster χωρίζεται με τον αριθμό των clusters για να υπολογιστεί η τελική καθαρότητα του συνόλου.

- Υπολογισμός F-measure:
  - Για κάθε cluster, υπολογίζονται τα True Positives (TP), False Positives (FP) και False Negatives (FN) χρησιμοποιώντας τον πίνακα σύγχυσης.
  - Υπολογίζονται οι μετρικές Precision και Recall για κάθε cluster.
  - Το F-measure υπολογίζεται ως ο αρμονικός μέσος των Precision και Recall.
  - Τέλος, το F-measure για κάθε cluster υπολογίζεται ως ο μέσος όρος των F-measure των διαφόρων clusters.

```
Purity για PCA: 0.6873113309837685
F-measure για PCA: 0.07293071736043442
Purity για Autoencoder: 0.7013129924310371
F-measure για Autoencoder: 0.1666307452002436
```

## Αποτελέσματα χρόνου:

Συνολικά για την εκτέλεση ολόκληρου του κώδικα απαιτήθηκε περίπου 1 λεπτό.

Kernel status: Idle

Executed 7 cells

Elapsed time: 61 seconds

*Figure 1: Jupyter info*