**Έκθεση: Ανακατασκευή μεθόδων Autoencoder και PCA Reconstruction στο MNIST**

**1. Εισαγωγή**

Η παρούσα εργασία επικεντρώνεται στη χρήση διαφόρων αλγορίθμων για την ανακατασκευή ψηφίων από το σύνολο δεδομένων MNIST. Οι μέθοδοι που χρησιμοποιήθηκαν περιλαμβάνουν:

1. Έναν κλασικό autoencoder.
2. Ανακατασκευή μέσω Ανάλυσης Κύριων Συνιστωσών (PCA).
3. Έναν autoencoder για την πρόβλεψη του επόμενου ψηφίου (Next Digit Autoencoder).
4. Έναν classifier για την αξιολόγηση των ανακατασκευασμένων ψηφίων.

Οι στόχοι της εργασίας είναι:

* Να περιγραφεί η υλοποίηση των παραπάνω μεθόδων.
* Να αξιολογηθεί η απόδοση των μοντέλων.
* Να συγκριθούν οι autoencoders με την ανακατασκευή μέσω PCA.

**2. Αλγόριθμοι και Υλοποιήσεις**

**2.1 Classic Autoencoder**

Ο κλασικός autoencoder σχεδιάστηκε με έναν γραμμικό encoder και έναν γραμμικό decoder. Χρησιμοποιήθηκε η συνάρτηση απώλειας MSELoss και η εκπαίδευση πραγματοποιήθηκε με early stopping για αποφυγή υπερεκπαίδευσης.

* **Αρχιτεκτονική**:
  + Encoder: Linear(784, 128)
  + Decoder: Linear(128, 784)
* **Υλοποίηση**:
  + Αρχικοποίηση των βαρών με τον βελτιστοποιητή Adam (lr=0.001).
  + Διακοπή εκπαίδευσης στην 263η εποχή με χρόνο εκπαίδευσης ~9.52 λεπτά.

**2.2 PCA Reconstruction**

Η PCA χρησιμοποιήθηκε για τη μείωση των διαστάσεων στα 128 components και την ανακατασκευή των αρχικών δεδομένων μέσω της αντίστροφης μετατροπής.

* **Διαδικασία**:
  + Εφαρμογή PCA με Scikit-learn.
  + Ανακατασκευή μέσω inverse\_transform.
* **Πλεονέκτημα**:
  + Εξαιρετικά γρήγορη διαδικασία (~4.13 δευτερόλεπτα).

**2.3 Next Digit Autoencoder**

Αυτός ο autoencoder επεκτείνει την ιδέα της ανακατασκευής προβλέποντας το επόμενο ψηφίο. Χρησιμοποιήθηκε η ίδια αρχιτεκτονική με τον κλασικό autoencoder, αλλά με δεδομένα εκπαίδευσης κατάλληλα διαμορφωμένα.

* **Υλοποίηση**:
  + Αρχικοποίηση με Adam (lr=0.001).
  + Διακοπή εκπαίδευσης στην 432η εποχή με χρόνο ~20.50 λεπτά.

**2.4 Classifier**

Για την αξιολόγηση των ανακατασκευασμένων ψηφίων χρησιμοποιήθηκε ένα νευρωνικό δίκτυο ταξινόμησης (classifier). Το δίκτυο εκπαιδεύτηκε στο σύνολο δεδομένων MNIST και η απόδοσή του χρησιμοποιήθηκε ως μετρική για να ελεγχθεί η ποιότητα των ανακατασκευών. Ο classifier περιλαμβάνει δύο πλήρως συνδεδεμένα επίπεδα:

* **Αρχιτεκτονική**:
  + Πρώτο επίπεδο: Linear(784, 128), ενεργοποίηση ReLU.
  + Δεύτερο επίπεδο: Linear(128, 10), έξοδος για κατηγοριοποίηση 10 κλάσεων.
* **Σκοπός**:
  + Να αξιολογήσει εάν το ανακατασκευασμένο ψηφίο μπορεί να αναγνωριστεί σωστά.

**3. Αποτελέσματα**

**3.1 Classic Autoencoder**

* **Ακρίβεια**:
  + Training Set: 99.91%
  + Test Set: 97.87%
* **Χαρακτηριστικά Παραδείγματα**:
  + **Καλύτερη κατηγορία**: 0 -> 1 με ακρίβεια 99.03%.
  + **Χειρότερη κατηγορία**: 8 -> 9 με ακρίβεια 96.73%.
  + Σωστές ανακατασκευές: 7, 2, 1, 0, 4.
  + Εσφαλμένες ανακατασκευές: 9, 4, 7.
* **Χρόνος Εκπαίδευσης**: 9.52 λεπτά.

**3.2 PCA Reconstruction**

* **Ακρίβεια**:
  + Training Set: 99.46%
  + Test Set: 97.99%
* **Χαρακτηριστικά Παραδείγματα**:
  + **Καλύτερη κατηγορία**: 9 -> 0 με ακρίβεια 99.18%.
  + **Χειρότερη κατηγορία**: 7 -> 8 με ακρίβεια 97.13%.
  + Σωστές ανακατασκευές: 7, 2, 1, 0, 4.
  + Εσφαλμένες ανακατασκευές: 4, 5, 6.
* **Χρόνος Εκτέλεσης**: 4.13 δευτερόλεπτα.

**3.3 Next Digit Autoencoder**

* **Ακρίβεια**:
  + Training Set: 99.73%
  + Test Set: 95.40%
* **Χαρακτηριστικά Παραδείγματα**:
  + **Καλύτερη κατηγορία**: 1 -> 2 με ακρίβεια 97.97%.
  + **Χειρότερη κατηγορία**: 4 -> 5 με ακρίβεια 93.48%.
  + Σωστές προβλέψεις: 8, 3, 2, 1, 5.
  + Εσφαλμένες προβλέψεις: 6, 4, 7.
* **Χρόνος Εκπαίδευσης**: 20.50 λεπτά.

**4. Σύγκριση Autoencoder και PCA**

**4.1 Απόδοση**

Η PCA εμφανίζει ελαφρώς καλύτερη ακρίβεια στο Test Set (97.99% έναντι 97.87%), αλλά ο autoencoder παρέχει μεγαλύτερη ευελιξία στην εκμάθηση μη γραμμικών σχέσεων.

**4.2 Χρόνος Εκτέλεσης**

Η PCA είναι σαφώς ταχύτερη από τον autoencoder, με χρόνο εκτέλεσης μόλις 4.13 δευτερόλεπτα, ενώ ο Classic Autoencoder χρειάζεται 9.52 λεπτά για την εκπαίδευση. Η διαφορά αυτή οφείλεται στο γεγονός ότι η PCA βασίζεται σε στατιστική επεξεργασία και όχι σε διαδικασία εκμάθησης μέσω βαρών. Στη PCA, τα δεδομένα μετασχηματίζονται γραμμικά σε έναν χαμηλότερων διαστάσεων χώρο και η ανακατασκευή γίνεται μέσω της αντίστροφης διαδικασίας. Αντίθετα, ο autoencoder χρειάζεται να μάθει τις αναπαραστάσεις μέσω επαναλαμβανόμενων εποχών, κάτι που αυξάνει σημαντικά το χρόνο εκτέλεσης. Παρόλα αυτά, ο χρόνος εκπαίδευσης του autoencoder μπορεί να μειωθεί με χρήση πιο ισχυρών υπολογιστικών συστημάτων ή βελτιστοποιημένων τεχνικών, όπως convolutional layers.

**4.3 Εφαρμογές**

Ο Classic Autoencoder είναι πιο κατάλληλος για σύνθετα δεδομένα, καθώς μπορεί να μάθει μη γραμμικές σχέσεις που δεν μπορεί να συλλάβει η PCA. Για παράδειγμα, σε δεδομένα με υψηλή πολυπλοκότητα ή σε περιπτώσεις όπου απαιτείται η κατανόηση σύνθετων προτύπων (όπως σε εικόνες υψηλής ανάλυσης), ο autoencoder υπερτερεί. Από την άλλη, η PCA είναι ιδανική για εφαρμογές όπου η ταχύτητα είναι σημαντική και τα δεδομένα μπορούν να περιγραφούν ικανοποιητικά μέσω γραμμικών σχέσεων, όπως η ανάλυση μεγάλων συνόλων δεδομένων ή η αρχική μείωση διαστάσεων πριν την εφαρμογή άλλων μοντέλων. Έτσι, η επιλογή εξαρτάται από τις απαιτήσεις της εφαρμογής.

**5. Συμπεράσματα**

Η εργασία απέδειξε ότι:

* Ο Classic Autoencoder επιτυγχάνει υψηλή ακρίβεια και είναι κατάλληλος για μη γραμμικές σχέσεις, αλλά απαιτεί μεγαλύτερο χρόνο εκπαίδευσης.
* Η PCA είναι εξαιρετικά γρήγορη και αποδοτική σε απλούστερα δεδομένα, αλλά περιορίζεται από τη γραμμική φύση της.
* Ο Next Digit Autoencoder παρουσίασε ικανοποιητική απόδοση, αλλά λόγω της πολυπλοκότητάς του και της πρόβλεψης επόμενου ψηφίου, η ακρίβειά του ήταν ελαφρώς χαμηλότερη.

Σε γενικές γραμμές, η επιλογή της κατάλληλης μεθόδου εξαρτάται από τη φύση των δεδομένων και τις απαιτήσεις της εκάστοτε εφαρμογής. Η εργασία απέδειξε ότι τόσο η PCA όσο και οι autoencoders μπορούν να αποδώσουν εξαιρετικά, υπό τις κατάλληλες συνθήκες.