

Πανεπιστήμιο Πατρών  
Τμήμα Μηχανικών Η/Υ και Πληροφορικής

**Στατιστική Επεξεργασία Σήματος και  
Μάθηση**

2η Εργαστηριακή Άσκηση

Ακαδημαϊκό Έτος 2023/24

Ζήσης Σούρλας \*

22 Φεβρουαρίου 2024

---

\*AM: 1072477 Email: sourlas.zisis@ac.upatras.gr

## Περιεχόμενα

<b>1 Θεωρητικό υπόβαθρο και υλοποίηση των μοντέλων</b>	<b>2</b>
1.1 Ανάλυση Κύριων Συνιστώσων (PCA) . . . . .	2
1.2 Αυτοκωδικοποιητές (Autoencoders) . . . . .	3
1.2.1 Γραμμικοί Αυτοκωδικοποιητές . . . . .	3
1.2.2 Μη Γραμμικοί Αυτοκωδικοποιητές . . . . .	4
<b>2 Πειραματική σύγκριση - Συμπεράσματα</b>	<b>5</b>
<b>3 Βιβλιογραφία</b>	<b>8</b>

# 1 Θεωρητικό υπόβαθρο και υλοποίηση των μοντέλων

## 1.1 Ανάλυση Κύριων Συνιστωσών (PCA)

**Θεωρητικό υπόβαθρο** Η Ανάλυση Κύριων Συνιστωσών είναι μία γραμμική μέθοδος μείωσης της διαστατικότητας των δεδομένων. Βασίζεται στην υπόθεση ότι οι μεταβλήτες των δεδομένων εμφανίζουν γραμμικές συσχετίσεις. Στόχος είναι η προβολή τους σε ένα χώρο μικρότερης διάστασης, διατηρώντας ταυτόχρονα όσο το δυνατόν περισσότερη πληροφορία. Αυτό επιτυγχάνεται υπολογίζοντας τα ιδιοδιανύσματα (κύριες συνιστώσες) του μητρώου αυτοσυσχέτισης των δεδομένων. Αυτά τα ιδιοδιανύσματα υποδεικνύουν την κατεύθυνση της διασποράς των δεδομένων. Όσο μεγαλύτερη είναι η ιδιοτυπία που αντιστοιχεί σε ένα ιδιοδιανύσμα, τόσο μεγαλύτερη είναι και η διασπορά (άρα και η πληροφορία) των δεδομένων στην κατεύθυνση που αυτό ορίζει. Στόχος της PCA είναι η διατήρηση της μέγιστης δυνατής πληροφορίας, και αυτό επιτυγχάνεται με την επιλογή των ιδιοδιανύσμάτων με τις μεγαλύτερες ιδιοτυπίες ώστε να αποτελέσουν τη βάση του νέου χώρου χαμηλότερης διάστασης. Τα αρχικά δεδομένα προβάλλονται πάνω στα επιλεχθέντα ιδιοδιανύσματα και έτσι επιτυγχάνεται η μείωση της διαστατικότητας.(Haykin, 2009)[1]

**Υλοποίηση** Η υλοποίηση της (PCA) στα πλαίσια της παρούσας εργασίας έγινε σε **python** με χρήση της βιβλιοθήκης **numpy**. Δημιουργήθηκε μια κλάση με μοναδική ιδιότητα ένα πίνακα, όπου αποθηκεύονται τα ιδιοδιανύσματα του μητρώου συνδιασποράς των δεδομένων εκπαίδευσης ταξινομημένα κατά φύλνουσα σειρά βάσει των ιδιοτυπών τους. Επιπλέον η κλάση έχει τέσσερις μεθόδους: την **train**, την **encode**, την **decode** και την **run**:

- **train:** Η μέθοδος αυτή δέχεται ως είσοδο τα δεδομένα εκπαίδευσης (εν προκειμένω τις εικόνες) και αφού τα μετατρέψει από μητρώα σε διανύσματα, υπολογίζει και αφαιρεί από αυτές τον μέσο όρο τους προκειμένου τα δεδομένα να έχουν μηδενική μέση τιμή. Αυτό το βήμα είναι απαραίτητο για την ορθή λειτουργία της PCA. Στη συνέχεια υπολογίζει το μητρώο συνδιασποράς των δεδομένων και τις ιδιοτυπίες και τα ιδιοδιανύσματά<sup>1</sup> του. Τέλος ταξινομεί τα ιδιοδιανύσματα κατά φύλνουσα σειρά βάσει των ιδιοτυπών τους και τα αποθηκεύει ως ιδιότητα του αντικειμένου προκειμένου να χρησιμοποιηθούν για την κωδικοποίηση / αποκωδικοποίηση δεδομένων.
- **encode:** Η μέθοδος αυτή δέχεται ως είσοδο τα δεδομένα που θα συμπληστούν και τον αφιθμό των κυριών συνιστωσών που θα χρησιμοποιηθούν (δηλαδή της νέας διάστασης των δεδομένων). Αφού τα μετατρέψει σε διάνυσμα πολλαπλασιάζει τα ιδιονύσματα που έχει αποθηκεύσει (μόνο όσα ορίζονται από την είσοδο) από την εκπαίδευση με τα δεδομένα εισόδου, προβάλλοντας

<sup>1</sup>Εδώ χρησιμοποιείται η ταχύτερη συνάρτηση `eigh` της `numpy` αντί της `eig` καθώς γνωρίζουμε ότι το μητρώο συνδιασποράς είναι συμμετρικό.

τα στο μικρότερο χώρο. Παράγει έτσι τα συμπιεσμένα δεδομένα τα οποία επιστρέφει μαζί με το αρχικό σχήμα των δεδομένων εισόδου.

- **decode:** Η μέθοδος αυτή δέχεται ως είσοδο τα συμπιεσμένα δεδομένα, το αρχικό τους σχήμα και τον αριθμό συνιστωσών που χρησιμοποιήθηκαν για να παραχθούν. Πολλαπλασιάζει τον ανάστροφο πίνακα ιδιοιδιανυσμάτων με τα συμπιεσμένα δεδομένα. Το αποτέλεσμα αυτής τη πράξης το μετασχηματίζει στο αρχικό σχήμα παράγοντας έτσι τα ανακατασκευασμένα (αποσυμπιεσμένα) δεδομένα τα οποία και επιστρέφει.
- **run:** Η μέθοδος αυτή εκτελεί διαδοχικά τις δύο παραπάνω μεθόδους προκειμένου να παράξει απευθείας τα ανακατασκευασμένα δεδομένα από τα αρχικά.

## 1.2 Αυτοκωδικοποιητές (Autoencoders)

### 1.2.1 Γραμμικοί Αυτοκωδικοποιητές

**Θεωρητικό υπόβαθρο** Οι αυτοκωδικοποιητές αποτελούν ένα είδος feed-forward νευρωνικού δικτύου που αποτελείται από δύο βασικά στοιχεία: έναν κωδικοποιητή και έναν αποκωδικοποιητή. Ο κωδικοποιητής έχει σχεδιαστεί για να συμπιέζει τα δεδομένα εισόδου σε μια αναπαράσταση χαμηλότερης διάστασης (bottleneck) ενώ ο ρόλος του αποκωδικοποιητή είναι να ανακατασκευάζει τα αρχικά δεδομένα από την έξοδο του κωδικοποιητή. Αυτή η διαδικασία συμπίεσης-αποσυμπίεσης πραγματοποιείται μέσω στρωμάτων με προοδευτικά μειούμενο αριθμό νευρώνων που στη συνέχεια αυξάνονται πίσω στην αρχική διάσταση στον αποκωδικοποιητή. Συνήθως, τα στρώματα αυτά έχουν συμμετρική δομή.

Στόχος είναι το νευρωνικό δίκτυο να διαχρίνει τις πιο κρίσιμες διαστάσεις ή συσχετίσεις εντός των δεδομένων εισόδου. Εφόσον εκπαιδευτούν σωστά, οι αυτοκωδικοποιητές αντιλαμβάνονται τα ουσιώδη χαρακτηριστικά για την αναπαράσταση των δεδομένων και είναι χρήσιμοι σε εφαρμογές όπως η μείωση της διαστατικότητας και η εξαγωγή χαρακτηριστικών.

Οι γραμμικοί αυτοκωδικοποιητές έχουν γραμμικές συναρτήσεις ενεργοποίησης, πράγμα που ισοδυναμεί με απουσία συνάρτησης ενεργοποίησης καθώς η είσοδος περνά αυτούσια στην έξοδο. Αν η συνάρτηση απώλειας την οποία προσπαθεί να μηδενίσει ο γραμμικός αυτοκωδικοποιητής είναι το Μέσο Τετραγωνικό Σφάλμα, τότε καταλήγει εν τέλει να εντοπίζει τις κύριες συνιστώσες των δεδομένων εισόδου, δηλαδή να ισοδυναμεί με την PCA (Charte et al., 2018) [2].

**Τλοποίηση** Για την υλοποίηση του γραμμικού αυτοκωδικοποιητή χρησιμοποιήθηκε η γλώσσα **python** και έγινε χρήση της βιβλιοθήκης **tensorflow**. Δημιουργήθηκε μια κλάση η οποία βασίζεται πάνω στο γενικό Model του tensorflow. Ως ιδιότητες έχει το αρχικό σχήμα των δεδομένων εισόδου καθώς και το μέγεθος της νέας αναπαράστασης των δεδομένων. Αποτελείται από δύο βασικές μεθόδους: την encoder και την decoder.

Η encoder αποτελείται από δύο στρώματα. Ένα στρώμα εισόδου που μετατρέπει την είσοδο σε διάνυσμα και ένα δεύτερο στρώμα (χρυφό) με αριθμό νευρώνων ίσο με τη μειωμένη διάσταση. Το δεύτερο στρώμα έχει γραμμική συνάρτηση ενεργοποίησης και δεν έχει biases στους νευρώνες.

Η decoder αποτελείται επίσης από δύο στρώματα. Το πρώτο έχει μέγεθος όσο το διάνυσμα εισόδου (δηλαδή οι διαστάσεις της εικόνας πολλαπλασιασμένες) και έχει επίσης γραμμική συνάρτηση ενεργοποίησης και απουσία biases. Το δεύτερο στρώμα μετατρέπει τα δεδομένα στην αρχική τους μορφή.

Για την εκπαίδευση των γραμμικών αυτοκωδικοποιητών χρησιμοποιήθηκε ο adam optimizer και το μέσο τετραγωνικό σφάλμα ως συνάρτηση απώλειας.

### 1.2.2 Μη Γραμμικοί Αυτοκωδικοποιητές

**Θεωρητικό υπόβαθρο** Για τους μη γραμμικούς κωδικοποιητές ισχύει ό, τι και για τους γραμμικούς με τη διαφορά ότι χρησιμοποιούν μη γραμμικές συναρτήσεις ενεργοποίησης. Ακριβώς αυτή τους η ιδιότητα τους επιτρέπει να εντοπίσουν μη γραμμικές συσχετίσεις μεταξύ των δεδομένων. Λόγω αυτού μπορούν να θεωρηθούν και γενίκευση της PCA. (Charte et al., 2018) [2]

**Τλοιοποίηση** Η υλοποίηση του μη γραμμικού αυτοκωδικοποιητή είναι αντίστοιχη με αυτή του γραμμικού.

Στον encoder χρησιμοποιούνται δύο στρώματα πέραν του στρώματος που μετατρέπει την είσοδο σε διάνυσμα. Το πρώτο στρώμα χρησιμοποιεί ως συνάρτηση ενεργοποίησης την ReLU και αριθμό νευρώνων διπλάσιο του μεγέθους της διάστασης κωδικοποίησης. Το δεύτερο στρώμα χρησιμοποιεί την SELU (παραλλαγή της ReLU που δεν μηδενίζει τις αρνητικές τιμές στην είσοδο της) ως συνάρτηση ενεργοποίησης και έχει μέγεθος ίσο με τη μειωμένη διάσταση.

Ο decoder αντίστοιχα έχει δύο στρώματα πέραν του τελευταίου που μετασχηματίζει την έξοδο. Το πρώτο έχει μέγεθος ίδια με του αντίστοιχου στον encoder και συνάρτηση ενεργοποίησης την SELU. Το δεύτερο έχει μέγεθος ίσο με την αρχική διάσταση των δεδομένων και συνάρτηση ενεργοποίησης την ReLU.

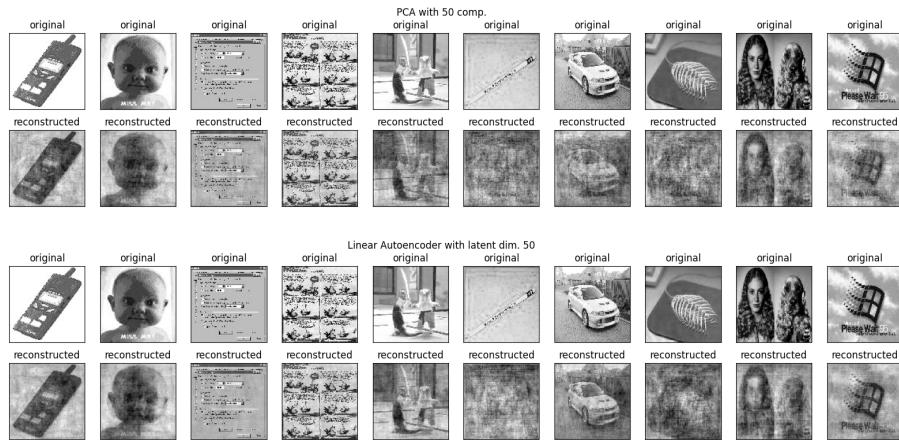
Παρά την παρατήρηση ότι η ReLU μειώνει την απόδοση των AE και παρότι θεωρητικά οι σιγμοειδείς συναρτήσεις προτιμώνται (Charte et al., 2018) [2], βρήκαμε πειραματικά ότι αυτός ο συνδιασμός ReLU και SELU οδηγεί σε καλύτερα αποτελέσματα.

Για την εκπαίδευση των μη γραμμικών αυτοκωδικοποιητών χρησιμοποιήθηκε ο adam optimizer και το μέσο τετραγωνικό σφάλμα ως συνάρτηση απώλειας.

## 2 Πειραματική σύγχριση - Συμπεράσματα



Figure 1: Αρχικές και ανακατασκευασμένες εικόνες σε χώρο 100 διαστάσεων για τις τρεις μεθόδους



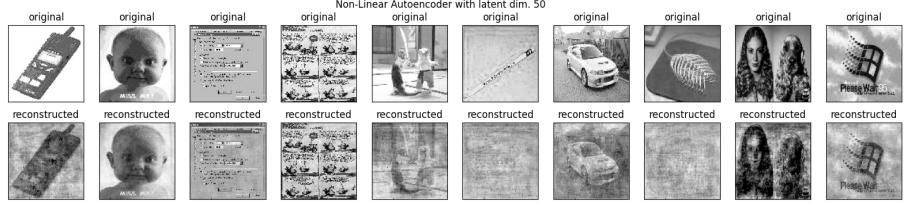


Figure 2: Αρχικές και ανακατασκευασμένες εικόνες σε χώρο 50 διαστάσεων για τις τρεις μεθόδους



Figure 3: Αρχικές και ανακατασκευασμένες εικόνες σε χώρο 10 διαστάσεων για τις τρεις μεθόδους

Παρατηρούμε ότι βάσει του διαγράμματος 4 ο γραμμικός αυτοκωδικοποιητής είναι σταθερά η καλύτερη μέθοδος μείωσης διαστατικότητας ανεξάρτητως μεγέθους του χώρου προβολής. Ο μη γραμμικός αυτοκωδικοποιητής είναι σταθερά κατώτερος του γραμμικού ενώ παρουσιάζει ίδια ποσοστά επιτυχούς ανάκτησης με την PCA με εξαίρεση τον χώρο μεγέθους 100.

Με μια πρώτη ματιά τα αποτελέσματα αυτά ίσως φαντάζουν κάπως παράδοξα. Ενώ αναμένεται ο γραμμικός αυτοκωδικοποιητής να προσεγγίζει την PCA, αυτός παρουσιάζει καλύτερη απόδοση. Αυτό πιστεύουμε ότι οφείλεται σε δύο λόγους.

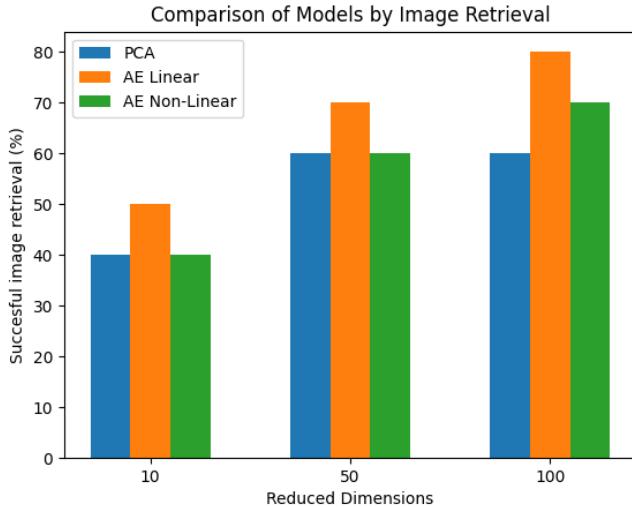


Figure 4: Σύγκριση ποσοστών επιτυχούς ανάκτηση σε διαφορετικούς χώρους για τις τρεις μεθόδους

Πρώτον ότι ο υπολογισμός των ιδιοτυπών και των ιδιοδιανυσμάτων στην PCA δεν είναι απόλυτα ακριβής<sup>2</sup>. Δεύτερον ότι το ποσοστό επιτυχούς ανάκτησης ίσως να μην είναι απολύτως ενδεικτικό της ποιότητας των μεθόδων. Οι εικόνες στα train και test datasets εμφανίζουν διαφορές ως προς την φωτεινότητα, το χρώμα και άλλα στοιχεία. Συνεπώς, μπορεί μια μέθοδος η οποία χάνει λίγο περισσότερη πληροφορία κατά τη συμπίεση, να οδηγεί σε μεγαλύτερο ποσοστό επιτυχούς ανάκτησης ακριβώς γιατί εξαλείφει σε μεγαλύτερο βαθμό αυτή την πληροφορία που διαφοροποιεί τα datasets. Αυτό ενδεχομένως εξηγεί και την μεγαλύτερη διαφορά που υπάρχει μεταξύ PCA και γραμμικού AE στον χώρο των 100 διαστάσεων.

Σε σχέση με τον μη γραμμικό AE παρατηρούμε ότι οι επιδόσεις του είναι κατώτερες από αυτές του γραμμικού. Αυτό υποθέτουμε ότι οφείλεται σε δύο πιθανούς παράγοντες. Πρώτον στο ότι τα δεδομένα μας ενδεχομένως να μην περιέχουν πολλές μη γραμμικές συσχετίσεις με αποτέλεσμα να μη συνεισφέρει κάτι η μη γραμμικότητα του μοντέλου στο πρόβλημα. Ο δεύτερος παράγοντας έχει να κάνει με την ίδια την υλοποίηση του μοντέλου. Ενδεχομένως η υλοποίηση μας να μην είναι βέλτιστη καθώς χρησιμοποιήσαμε ως συνάρτηση ενεργοποίησης τη ReLU, η οποία φαίνεται να έχει αρνητική επίπτωση στις επιδόσεις των AE (Charte et al., 2018) [2], αντί κάποιας σιγμοειδούς που προτείνεται για αυτά τα μοντέλα. Ωστόσο όπως προαναφέρθηκε οι πειραματισμοί μας μας οδήγησαν σε αυτές τις επιλογές.

<sup>2</sup>Είναι χαρακτηριστικό πως αν χρησιμοποιήσουμε την eig αντί της eigh, η οποία είναι υπολογιστικά πολύ ακριβότερη, το σφάλμα λόγω στρογγυλοποιήσεων είναι τόσο μεγάλο ώστε να εμφανίζονται μιγαδικά ιδιοδιανύσματα και ο υπολογισμός της PCA να καθίσταται αδύνατος.

Τέλος αξίζει να αναφερθεί πως παρά τις πιο πάνω παρατηρήσεις, τόσο ο όγκος και η ποικιλία των δεδομένων μας όσο και η μετρική που χρησιμοποιήθηκε δεν επαρκούν για να εξαχθούν γενικευμένα συμπεράσματα για αυτά τα μοντέλα.

### 3 Βιβλιογραφία

#### References

- [1] S. Haykin, *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*. Παπασωτηρίου, 2009, pp. 373–380.
- [2] D. Charte, F. Charte, S. García, M. J. del Jesus, and F. Herrera, “A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines”, *Information Fusion*, vol. 44, pp. 78–96, 2018, ISSN: 1566-2535. DOI: <https://doi.org/10.1016/j.inffus.2017.12.007>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1566253517307844>.