4η Εργασία
Αναγνώριση Προτύπων
Μπούρχα Ιωάννα 58019

Επιβλέπων καθηγητής: Ηλίας Θεοδωρακόπουλος

Ακαδημαϊκό έτος: 2023 - 2024

ΠΕΡΙΕΧΟΜΕΝΑ

ΑΣΚΗΣΗ 1

Το IRIS data set περιέχει μορφολογικές μετρήσεις για 150 φυτά iris (είδος κρίνου, αγριόκρινο). Τα δεδομένα είναι της μορφής: (μήκος σέπαλου, πλάτος σέπαλου, μήκος πετάλου, πλάτος πετάλου) σε cm. Από αυτά τα 150 φυτά, 50 είναι Iris Setosa (ω1), 50 είναι Iris Versicolour (ω2) και 50 είναι Iris Virginica (ω3).

Εφόσον το iris dataset βρίσκεται στην βιβλιοθήκη της sklearn, επιλέγω να το φορτώσω από εκεί.

Ερώτημα Α
Να χωριστεί το dataset σε 80% training και 20% test δείγματα με τυχαίο τρόπο.

Ο διαχωρισμός του dataset σε train και test γίνεται με την έτοιμη συνάρτηση της βιβλιοθήκης sklearn.

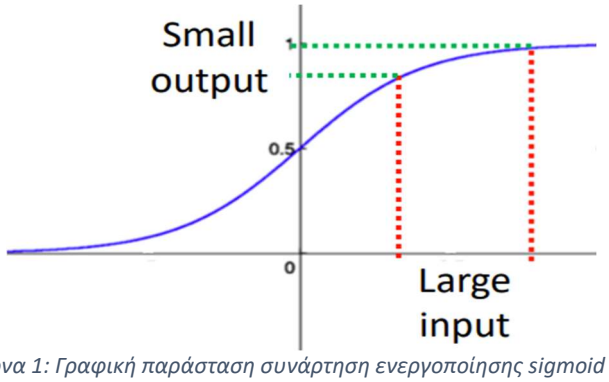
Ερώτημα Β
Να εκπαιδευτεί νευρωνικό δίκτυο 2 επιπέδων για την ταξινόμηση των δειγμάτων, όπου το 1ο layer θα έχει 30 νευρώνες και συνάρτηση ενεργοποίησης θα είναι sigmoid.

Για την δημιουργία του νευρωνικού δικτύου χρησιμοποιήθηκε η βιβλιοθήκη PyTorch. Μέσα σε μία κλάση ορίζω την αρχιτεκτονική που ζητείται και δημιουργώ ένα στιγμιότυπο του μοντέλου μου. Το πρώτο hidden layer έχει νευρώνες εισόδου όσο το πλήθος των χαρακτηριστικών των δεδομένων και νευρώνες εξόδου του είναι 30. Το δεύτερο hidden layer έχει νευρώνες εισόδου όσοι οι νευρώνες εξόδου του προηγούμενου και νευρώνες εξόδου όσες οι κλάσεις του προβλήματος.

Το βασικό δομικό στοιχείο της βιβλιοθήκης PyTorch είναι οι tensors, τύπος δεδομένων παρόμοιος με τους πίνακες στην βιβλιοθήκη numpy με την διαφορά ότι οι tensors μπορούν να επιταχυνθούν με την χρήση GPUs. Η μετατροπή των δεδομένων γίνεται με τις αντίστοιχες συναρτήσεις. Για τα δεδομένα οι τιμές του tensor είναι float32, ενώ για τις ετικέτες long. Έπειτα, για τα δεδομένα εκπαίδευσης (train data) δημιουργώ το dataset σε μορφή tensor με την αντίστοιχη συνάρτηση της βιβλιοθήκης.

Έχοντας ολοκληρώσει την προεπεξεργασία των δεδομένων, εκπαιδεύω το δίκτυο και εκτυπώνω τα αποτελέσματα για την απώλεια και την ακρίβεια για κάθε 10 εποχές. Η εκπαίδευση των νευρωνικών δικτύων στηρίζεται στην διαδικασία του backpropagation, δηλαδή την ανάδραση των σφαλμάτων από την έξοδο του δικτύου προς την είσοδο του, ώστε να προσαρμόζονται τα βάρη με την χρήση της αλυσιδωτής παραγώγου (chain rule) του διαφορικού λογισμού. Η παράγωγος της συνάρτησης ενεργοποίησης του νευρώνα χρησιμοποιείται για να υπολογιστεί η αλλαγή των βαρών του ώστε να μειωθεί το συνολικό σφάλμα του δικτύου.

Η μαθηματική σχέση της συνάρτησης ενεργοποίησης sigmoid είναι $\sigma(x) = \frac{1}{1+e^{-x}}$ περιορίζοντας την είσοδο x (πραγματικός αριθμός) σε μία τιμή μεταξύ του 0 και του 1. Η μη γραμμική αυτή έξοδος την καθιστά χρήσιμη για μοντελοποίηση πιθανοτήτων, ενώ παράλληλα επιτρέπει στα νευρωνικά δίκτυα να εντοπίσουν πολύπλοκες σχέσεις στα δεδομένα τους. Επίσης, είναι πλήρως διαφορίσιμη, $\sigma'(x) = \frac{\sigma(x)}{1-\sigma(x)}$, απαραίτητο στοιχείο για την εφαρμογή του chain rule. Το μειονέκτημά της είναι ότι η παράγωγός της παίρνει μικρές τιμές τόσο για χαμηλές όσο και για υψηλές τιμές ενεργοποιητικών, καθυστερώντας την ενημέρωση των βαρών με ενδεχόμενο ακόμα και να σταματήσει η εκπαίδευση. Ως ενεγροποιητικό ονομάζεται η τιμή της συνάρτησης ενεργοποίησης που παράγεται από τον νευρώνα.



Εικόνα 1: Γραφική παράσταση συνάρτηση ενεργοποίησης sigmoid

Για την εκπαίδευση του δικτύου αποφάσισα να χρησιμοποιήσω την συνάρτηση απωλειών CrossEntropyLoss, μία συνάρτηση που χρησιμοποιείται συχνά σε εφαρμογές ταξινόμησης πολλών κλάσεων. Η συνάρτηση απωλειών, γνωστή και ως συνάρτηση κόστος ή κριτήριο, είναι μία μαθηματική συνάρτηση που μετράει πόσο καλές είναι οι προβλέψεις του μοντέλου σε σύγκριση με τις πραγματικές τιμές. Με άλλα λόγια, μετράει την απόκλιση της προβλεπόμενης από την πραγματική τιμή. Όσο χαμηλότερη είναι η τιμή αυτής της απώλειας, τόσο περισσότερο το προσεγγίζει το μοντέλο την πραγματικότητα. Κατά την εκπαίδευση, η αναπροσαρμογή των βαρών γίνεται αποσκοπώντας στην μείωση της απώλειας της συνάρτησης αυτής. Επομένως, η επιλογή της συνάρτησης απωλειών είναι σημαντική για την απόδοση του μοντέλου μου. Η μαθηματική σχέση της Cross Entropy Loss (H) είναι:

$$H(p,q) = -\sum_x p(x)\log(x)$$

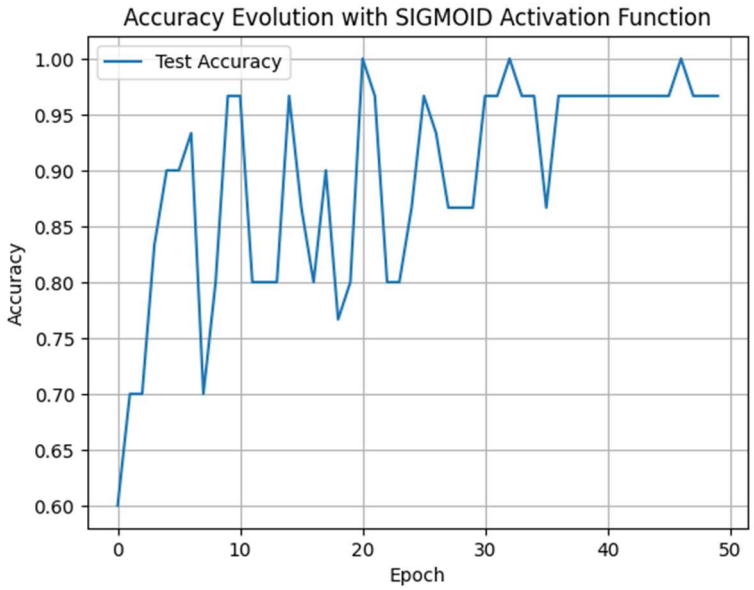
όπου $p(x)$ είναι η πραγματική τιμή και $q(x)$ η προβλεπόμενη τιμή του μοντέλου.

```
Epoch [10/50],    Test Loss: 0.9301,      Test Accuracy: 80.0000%
Epoch [20/50],    Test Loss: 0.7477,      Test Accuracy: 80.0000%
Epoch [30/50],    Test Loss: 0.6961,      Test Accuracy: 80.0000%
Epoch [40/50],    Test Loss: 0.3427,      Test Accuracy: 96.6667%
Epoch [50/50],    Test Loss: 0.4148,      Test Accuracy: 100.0000%
```

Ερώτημα Γ

Να αποτυπωθεί η εξέλιξη του test accuracy μετά από κάθε epoch εκπαίδευσης σε διάγραμμα, και να υπολογιστεί ο πίνακας σύγχυσης του τελικού μοντέλου για το test set.

Για την δημιουργία του διαγράμματος και του πίνακα σύγχυσης αξιοποιήθηκε η βιβλιοθήκη `matplotlib.pyplot`. Τα αντίστοιχα διαγράμματα φαίνονται στις εικόνες 2 και 3 αντίστοιχα. Στο σημείο αυτό τονίζω ότι μετά από κάθε εποχή εκπαίδευσης, δοκιμάζονται τα αποτελέσματά της στο test set από όπου προκύπτει η ακρίβεια (accuracy). Η ακρίβεια είναι μία μετρική που συνήθως χρησιμοποιείται για την αξιολόγηση της επίδοσης μοντέλων ταξινόμησης και αναπαρίσταται σε ποσοστό από το 0 έως το 1, όπου το 1 σημαίνει τέλεια ακρίβεια.

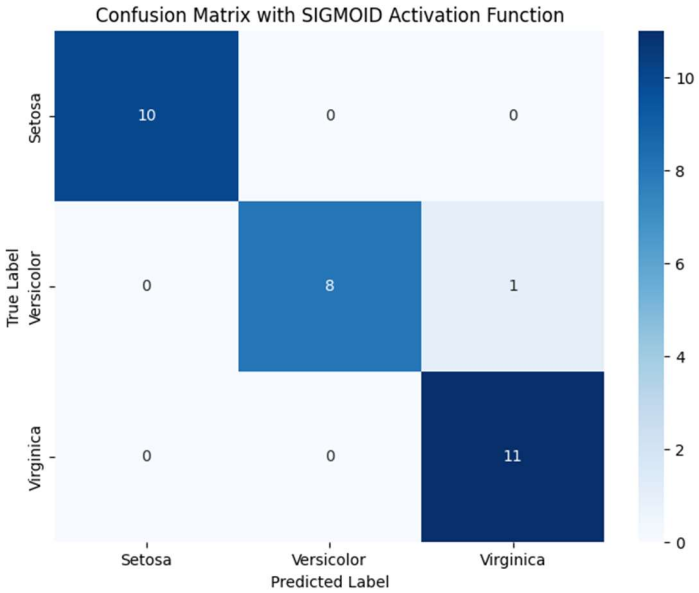


Εικόνα 2: Εξέλιξη του test accuracy κάθε epoch για το μοντέλο με συνάρτηση ενεργοποίησης sigmoid.

Το μοντέλο δείχνει να έχει σημαντική βελτίωση στην αρχική φάση της εκπαίδευσης, με την ακρίβεια να ανεβαίνει απότομα και στη συνέχεια να παρουσιάζει μια πιο ομαλή αλλά σημαντική αύξηση, που ενδεχομένως υποδηλώνει ότι το μοντέλο διανύει φάσεις υπερπροσαρμογής (overfitting) ή αναζήτησης τοπικών βελτιστοποιήσεων στον χώρο των παραμέτρων.

Ο πίνακας σύγχυσης (confusion matrix) χρησιμοποιείται για την αξιολόγηση της επίδοσης ενός ταξινομητή και την κατανόηση του τύπου των λαθών που κάνει. Στον οριζόντιο άξονα αναγράφονται οι προβλεπόμενες κατηγορίες (Predicted Label) και στον κάθετο άξονα οι πραγματικές κατηγορίες (True Label).

Οι κατηγορίες που χρησιμοποιούνται είναι: Setosa, Versicolor και Virginica. Κάθε κελί του πίνακα δείχνει τον αριθμό των δειγμάτων που ταξινομήθηκαν σε κάθε συνδυασμό πραγματικής και προβλεπόμενης κατηγορίας. Το μοντέλο προέβλεψε σωστά 10 δείγματα της κατηγορίας Setosa (κανένα λάθος), 8 δείγματα της κατηγορίας Versicolor (με 1 λάθος που ταξινομήθηκε ως Virginica), και 11 δείγματα της κατηγορίας Virginica (χωρίς λάθη). Η διαβάθμιση του χρώματος από ανοιχτό σε σκούρο μπλε αντιστοιχεί στον αριθμό των δειγμάτων: όσο πιο σκούρο το χρώμα, τόσο μεγαλύτερος ο αριθμός των δειγμάτων. Αυτός ο πίνακας δείχνει ότι το μοντέλο έχει μια πολύ καλή επίδοση, με πολύ λίγα λάθη στην ταξινόμηση.



Εικόνα 3: Πίνακας σύγχυσης για το μοντέλο με συνάρτηση ενεργοποίησης sigmoid.

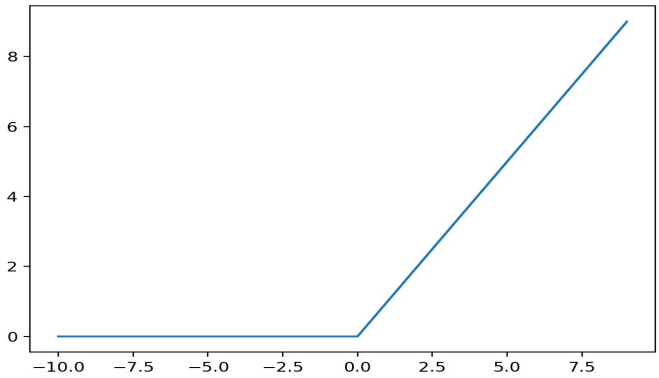
Ερώτημα Δ
Να επαναληφθεί η εκπαίδευση με συνάρτηση ενεργοποίησης την ReLU. Τί παρατηρείτε;

Η συνάρτηση ενεργοποίησης ReLU (Rectified Linear Unit) είναι μια από τις πιο διαδεδομένες συναρτήσεις ενεργοποίησης στον τομέα της μηχανικής μάθησης για βαθιά νευρωνικά δίκτυα. Ο τύπος της είναι:

$$f(x) = max(0, x)$$

Με άλλα λόγια, για θετική είσοδο x, η συνάρτηση επιστρέφει την τιμή αυτή ως έξοδο, ενώ αν η είσοδος είναι μη θετική (μηδέν ή αρνητική) η έξοδος είναι μηδέν. Η γραμμική αυτή έξοδος για θετική είσοδος επιτρέπει στο δίκτυο να μαθαίνει γρήγορα. Ακόμη, λόγω της απλότητας του μαθηματικού της τύπου, τα δίκτυα που την χρησιμοποιούν μπορούν να πιο γρήγορα τις απαιτούμενες πράξεις σε σχέση με άλλα που χρησιμοποιούν συναρτήσεις ενεργοποίησης με πιο περίπλοκο μαθηματικό τύπο.

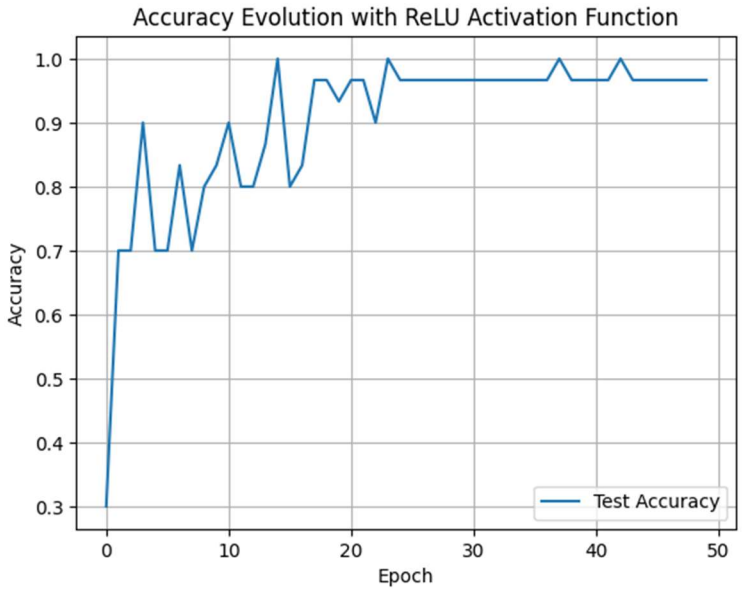
Συγκρίνοντας τις γραφικές παραστάσεις για τις συναρτήσεις ενεργοποίησης sigmoid και ReLU, παρατηρώ ότι η δεύτερη δεν είναι κορεσμένη για θετικές τιμές εισόδου. Κατά την ενημέρωση των βαρών, η ReLU δεν εμφανίζει μικρές τιμές της παραγώγου.



Εικόνα 4: Γραφική παράσταση συνάρτησης ενεργοποίησης ReLU.

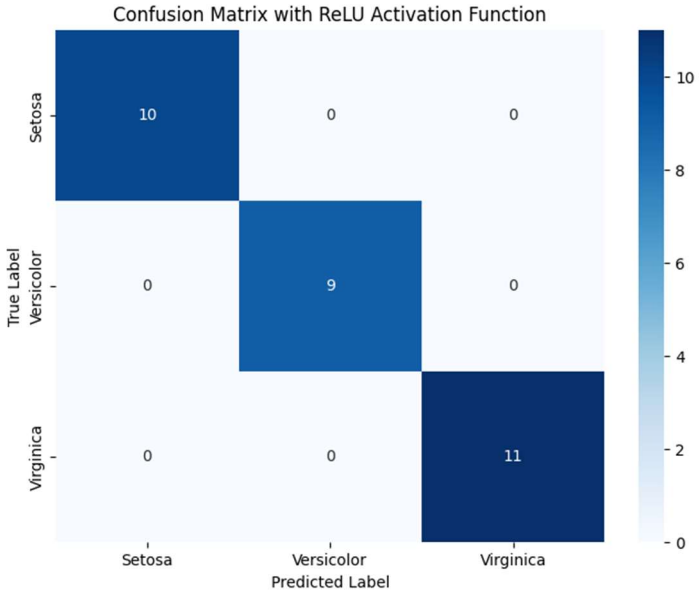
Το μειονέκτημα της ReLU είναι οι νεκροί νευρώνες. Οι νευρώνες που δέχονται αρνητική τιμή δεν ενεργοποιούνται, με αποτέλεσμα να μην συμβάλλουν στην εκμάθηση. Η λύση αυτού του προβλήματος δίνεται από τις παραλλαγές της ReLU που επιτρέπουν μικρές μη μηδενικές τιμές για αρνητικές εισόδους ώστε να διατηρούνται ενεργοί οι νευρώνες.

```
Epoch [10/50],    Test Loss: 0.6173,      Test Accuracy: 83.3333%
Epoch [20/50],    Test Loss: 0.4818,      Test Accuracy: 90.0000%
Epoch [30/50],    Test Loss: 0.2844,      Test Accuracy: 100.0000%
Epoch [40/50],    Test Loss: 0.2183,      Test Accuracy: 100.0000%
Epoch [50/50],    Test Loss: 0.1603,      Test Accuracy: 100.0000%
```



Εικόνα 5: Εξέλιξη του test accuracy κάθε epoch για το μοντέλο με συνάρτηση ενεργοποίησης ReLU.

Συγκρίνοντας την εξέλιξη του test accuracy κάθε εποχής για το ίδιο μοντέλο με συνάρτηση ενεργοποίησης sigmoid (εικόνα 2) και ReLU (εικόνα 5) παρατηρώ ότι η πρώτη έχει μεγάλη διακύμανση. Η ακρίβεια αυξάνεται απότομα στις πρώτες εποχές και στην συνέχεια διακυμαίνεται σε υψηλότερα επίπεδα. Αντίθετα, στην δεύτερη περίπτωση με συνάρτηση ενεργοποίησης ReLU εντοπίζεται μικρότερη διακύμανση καθώς η ακρίβεια φαίνεται να σταθεροποιείται πιο γρήγορα. Η χρήση της σιγμοειδούς συνάρτησης παρέχει καλύτερη τελική ακρίβεια αλλά με μεγαλύτερη διακύμανση στην επίδοση, ενώ η ReLU δίνει λιγότερο μεταβλητά αποτελέσματα και σταθεροποιείται πιο γρήγορη σε υψηλό επίπεδο ακρίβειας.



Εικόνα 6: Πίνακας σύγχυσης για το μοντέλο με συνάρτηση ενεργοποίησης ReLU.

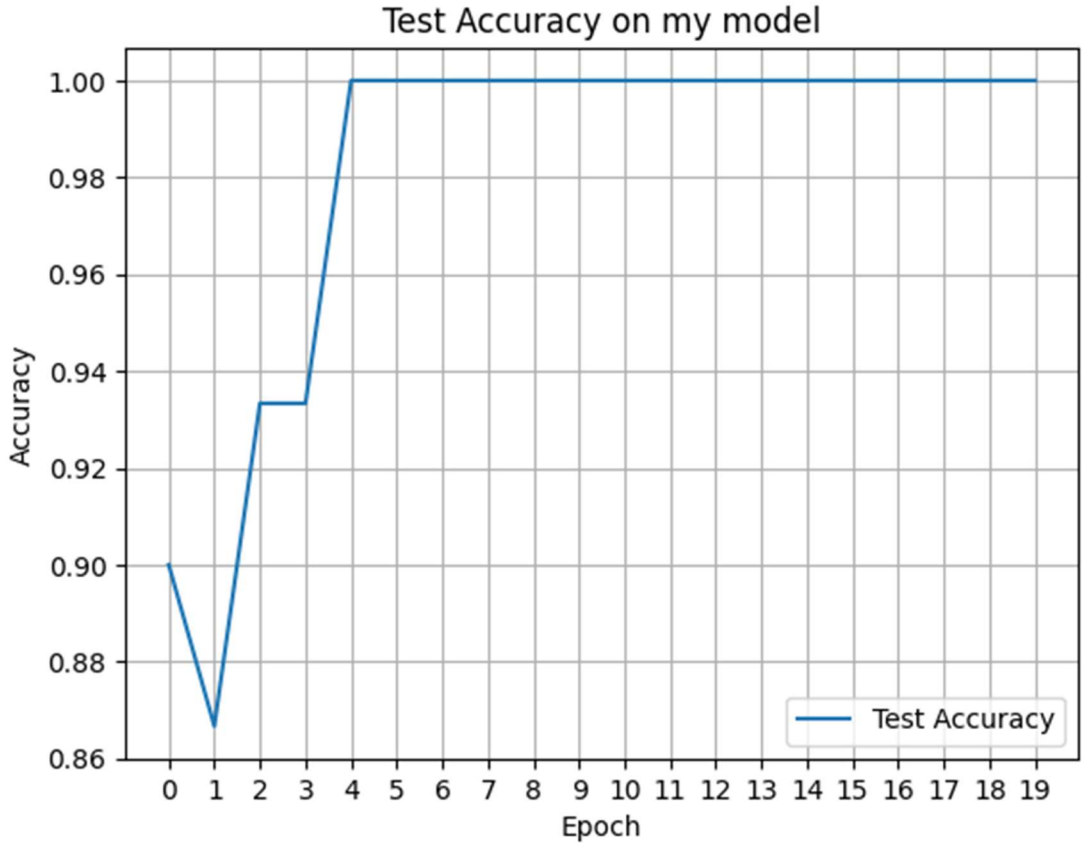
Το μοντέλο προέβλεψε σωστά 10 δείγματα της κατηγορίας Setosa (κανένα λάθος), 9 δείγματα της κατηγορίας Versicolor (κανένα λάθος) και 11 δείγματα της κατηγορίας Virginica (κανένα λάθος). Η διαβάθμιση του χρώματος από ανοιχτό σε σκούρο μπλε αντιστοιχεί στον αριθμό των δειγμάτων: όσο πιο σκούρο το χρώμα, τόσο μεγαλύτερος ο αριθμός των δειγμάτων. Αυτός ο πίνακας δείχνει ότι το μοντέλο έχει μια πολύ καλή επίδοση.

Ερώτημα Ε

Πειραματιστείτε με τις παραμέτρους σχεδίασης, και προτείνετε ένα δίκτυο που επιτυγχάνει καλύτερη ακρίβεια από τα προηγούμενα. Τεκμηριώστε το αποτέλεσμα με τα απαραίτητα γραφήματα και πίνακες σύγχυσης.

Αρχικά, αποφάσισα να προσθέσω ένα ενδιάμεσο hidden layer 15 νευρώνων. Το batch size κρατήθηκε σταθερό στην τιμή 10. Σταθερό επίσης διατηρήθηκε και το κριτήριο CrossEntropy. Μία σημαντική διαφορά είναι η κανονικοποίηση των δεδομένων με την συνάρτηση `StandardScaler` της `sklearn` ώστε να έχουν μηδενική μέση τιμή και μοναδιαία variance. Εφόσον τα δεδομένα αλλάζουν, δημιουργώ ξανά τους απαραίτητους τένσορες. Ακόμη, ο ρυθμός εκμάθησης ξεκινάει από 0,01 και υποδεκαπλασιάζεται κάθε 5 εποχές. Η εκπαίδευση γίνεται συνολικά για 20 εποχές.

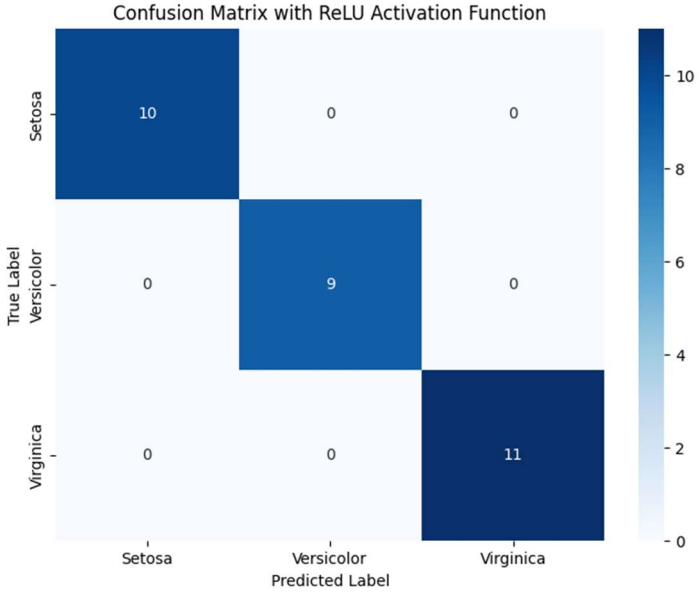
```
Epoch [1/20],    Loss: 0.7582,    Test Accuracy: 90.00%
Epoch [2/20],    Loss: 0.5006,    Test Accuracy: 86.67%
Epoch [3/20],    Loss: 0.3736,    Test Accuracy: 93.33%
Epoch [4/20],    Loss: 0.2467,    Test Accuracy: 93.33%
Epoch [5/20],    Loss: 0.2002,    Test Accuracy: 100.00%
Epoch [6/20],    Loss: 0.1607,    Test Accuracy: 100.00%
Epoch [7/20],    Loss: 0.1372,    Test Accuracy: 100.00%
Epoch [8/20],    Loss: 0.1649,    Test Accuracy: 100.00%
Epoch [9/20],    Loss: 0.1630,    Test Accuracy: 100.00%
Epoch [10/20],   Loss: 0.0637,    Test Accuracy: 100.00%
Epoch [11/20],   Loss: 0.1823,    Test Accuracy: 100.00%
Epoch [12/20],   Loss: 0.1198,    Test Accuracy: 100.00%
Epoch [13/20],   Loss: 0.1323,    Test Accuracy: 100.00%
Epoch [14/20],   Loss: 0.1620,    Test Accuracy: 100.00%
Epoch [15/20],   Loss: 0.1006,    Test Accuracy: 100.00%
Epoch [16/20],   Loss: 0.1696,    Test Accuracy: 100.00%
Epoch [17/20],   Loss: 0.1082,    Test Accuracy: 100.00%
Epoch [18/20],   Loss: 0.1845,    Test Accuracy: 100.00%
Epoch [19/20],   Loss: 0.0972,    Test Accuracy: 100.00%
Epoch [20/20],   Loss: 0.1881,    Test Accuracy: 100.00%
```



Εικόνα 6: : Εξέλιξη του test accuracy κάθε epoch για το δικό μου μοντέλο.

Συγκριτικά με τα δύο προαναφερόμενα μοντέλο, σε αυτό που έφτιαξα εγώ η ακρίβεια στο test set αυξάνεται πιο γρήγορα και φτάνει μάλιστα στο 100% σε λιγότερες εποχές. Αξιοσημείωτο είναι επίσης, ότι το αποτέλεσμα δεν παρουσιάζει διακύμανση.

Αναφορικά με τον πίνακα σύγχυσης, η απουσία αριθμών εκτός της κύριας διαγωνίου δηλώνει ότι το μοντέλο έχει άριστη επίδοση για τα δοθέντα δεδομένα. Το μοντέλο προέβλεψε σωστά 10 δείγματα της κατηγορίας Setosa (κανένα λάθος), 9 δείγματα της κατηγορίας Versicolor (κανένα λάθος) και 11 δείγματα της κατηγορίας Virginica (κανένα λάθος). Η διαβάθμιση του χρώματος από ανοιχτό σε σκούρο μπλε αντιστοιχεί στον αριθμό των δειγμάτων: όσο πιο σκούρο το χρώμα, τόσο μεγαλύτερος ο αριθμός των δειγμάτων. Αυτός ο πίνακας δείχνει ότι το μοντέλο έχει μια πολύ καλή επίδοση.



Εικόνα 7: Πίνακας σύγχυσης για το δικό μου μοντέλο.

Ερώτημα ΣΤ

Στο καλύτερο δίκτυο από αυτά που εκπαιδεύσατε, υπολογίστε και αποτυπώστε τις καμπύλες precision-recall για τις τρεις κλάσεις, μεταβάλλοντας το κατώφλι της πιθανότητας κάθε κλάσης. Ποια κλάση είναι πιο εύκολα διαχωρίσιμη με βάση το AUC (Area Under Curve) που προκύπτει από τις καμπύλες;

Η καμπύλη Precision – Recall αξιοποιείται για την αξιολόγηση της απόδοσης μοντέλων ταξινόμησης. Το Precision (ακρίβεια) αναφέρεται στο ποσοστό των πραγματικά θετικών προβλέψεων ως προς το σύνολο των προβλέψεων που ταξινομήθηκαν ως σωστά. Η μετρική αυτή δηλώνει πόσο αξιόπιστες είναι οι θετικές προβλέψεις του μοντέλου.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

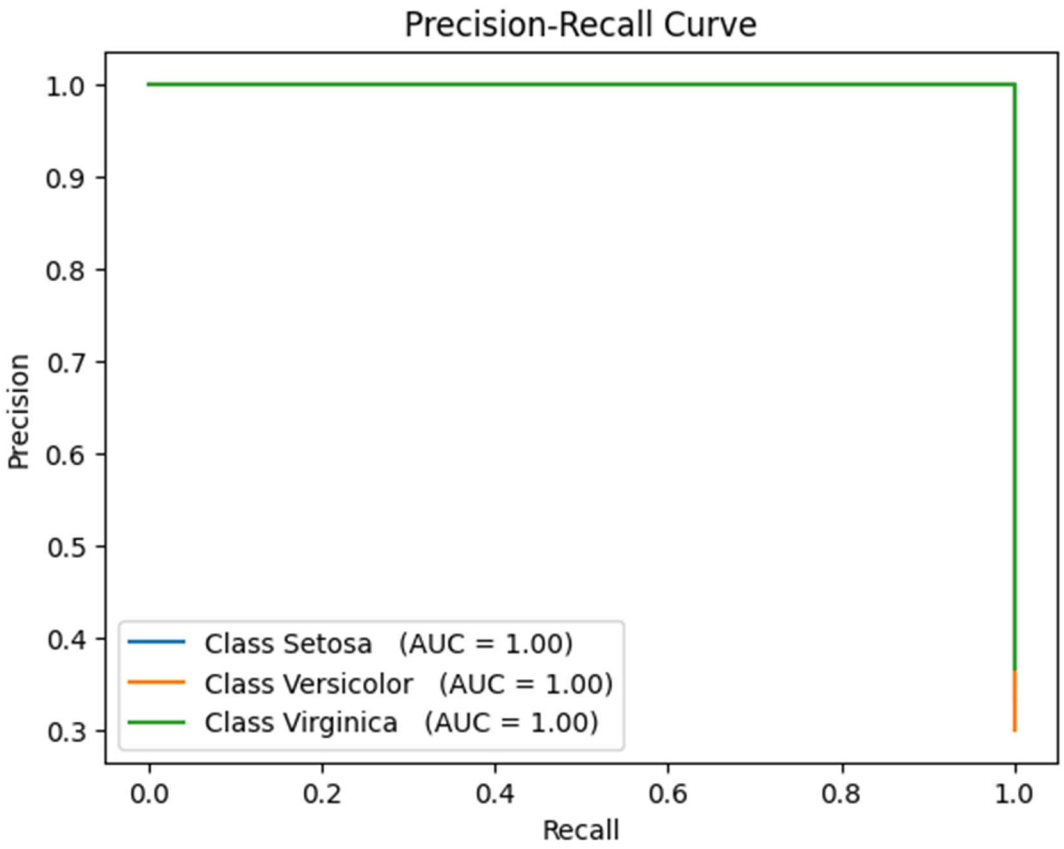
Το Recall (ευαισθησία) αναφέρεται στο ποσοστό των πραγματικά θετικών προβλέψεων ως προς το σύνολο των σωστών αποτελεσμάτων. Η μετρική αυτή δηλώνει πόσο ικανό είναι το μοντέλο στο να εντοπίσει όλα τα θετικά αποτελέσματα.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Η καμπύλη Precision-Recall απεικονίζει την σχέση μεταξύ των δύο αυτών μετρικών για διαφορετικά κατώφλια ταξινόμησης. Στον άξονα x τοποθετείται η ευαισθησία και στον άξονα y η ακρίβεια. Συνήθως, όταν το κατώφλι αυξάνεται, εντοπίζεται αύξηση στην ακρίβεια και μείωση στην ευαισθησία. Μέσω αυτής της καμπύλης μπορούμε να επιλέξουμε το βέλτιστο κατώφλι για την ταξινόμηση. Το AUC (Area Under Curve) της καμπύλης Precision-Recall αναφέρεται στο εμβαδόν κάτω από αυτήν την καμπύλη. Η τιμή του προσδιορίζει την συνολική απόδοση του μοντέλου για όλα τα πιθανά κατώφλια. Υψηλή τιμή σημαίνει ότι το μοντέλο διατηρεί υψηλή απόδοση ακόμα και όταν αυξάνεται η ευαισθησία, γεγονός ιδιαίτερα σημαντικό σε περιπτώσεις όπου τα False Positive έχουν κάποιο κόστος ή όταν τα True Positive είναι καθοριστικής σημασίας.

Αναφορικά με την υλοποίηση σε κώδικα, πρώτα πρέπει να υπολογίσω τις τιμές precision και recall για τα διάφορα πιθανά κατώφλια για κάθε κλάση. Ο υπολογισμός των πιθανοτήτων γίνεται με την χρήση της συνάρτησης softmax στις εξόδου του μοντέλου και οι τιμές τους αποθηκεύονται στην μεταβλητή `y_prob`. Έπειτα, καλώ τις συναρτήσεις `precision_recall_curve` και `auc` της βιβλιοθήκης `sklearn`.

Το καλύτερο μοντέλο είναι εκείνο που έτυχε την μικρότερη απώλεια. Όπως φαίνεται από τα παραπάνω, το μοντέλο αυτό είναι εκείνο του ερωτήματος Ε.



Εικόνα 8: Καμπύλη Precision-Recall με AUC για το μοντέλο του ερωτήματος Ε

Στην εικόνα 8 φαίνεται ότι για όλες τις καμπύλες το AUC είναι μονάδα, δηλαδή το μέγιστο δυνατό. Έχει επιτευχθεί η ιδανική κατάσταση όπου συνυπάρχει υψηλή ακρίβεια και υψηλή ανάκληση, δηλαδή το μοντέλο προβλέπει σωστά τις κλάσεις χωρίς λάθη. Η κατάσταση αυτή είναι αναμενόμενη δεδομένου το accuracy που επιτυγχάνει.

Στο σημείο αυτό σημειώνω ότι η τέλεια απόδοση είναι σπάνια σε πραγματικές εφαρμογές και ενδέχεται να υποδηλώνει ότι το μοντέλο έχει υπερεκπαιδευτεί στα δεδομένα (overfitting) ή ότι τα δεδομένα του test set δεν είναι αντιπροσωπευτικά των πραγματικών συνθηκών. Στην συγκεκριμένη εργασία, εφόσον εργαζόμαστε για ένα μικρό σύνολο δεδομένων το οποίο δεν παρουσιάζει προκλήσεις στα δεδομένα του, ακόμα και με ένα απλό νευρικό δίκτυο μπορούν να επιτευχθούν ιδανικές καταστάσεις.

ΑΣΚΗΣΗ 2

Να εκπαιδεύσετε ένα νευρωνικό δίκτυο με αρχιτεκτονική autoencoder, με 3 επίπεδα κωδικοποίησης 128, 32 και 3 νευρώνων και 3 αντίστοιχα επίπεδα αποκωδικοποίησης, χρησιμοποιώντας το training set της βάσης δεδομένων MNIST Digits που χρησιμοποιήθηκε στην εργαστηριακή άσκηση 6.

Ο autoencoder είναι ένας τύπος νευρωνικών δικτύων που χρησιμοποιείται για την απεικονίσεων των δεδομένων σε χαμηλότερες διαστάσεις. Η αρχιτεκτονική ενός autoencoder αποτελείται συνήθως από τρία κύρια μέρη: τον κωδικοποιητή (encoder), τον κωδικό (code) και τον αποκωδικοποιητή (decoder).

Κωδικοποιητής (Encoder):
Σύνολο νευρωνικών επιπέδων (neural layers) που λαμβάνει τα εισερχόμενα δεδομένα και τα μετατρέπει σε μια αναπαράσταση με μικρότερες διαστάσεις, γνωστή ως κωδικός. Η διαδικασία αυτή ονομάζεται κωδικοποίηση. Συνήθως τα νευρωνικά επίπεδα που την αποτελούν είναι πλήρως συνδεδεμένα.

Κωδικός (Code):
Είναι η αναπαράσταση των δεδομένων σε χαμηλότερες διστάσεις στοχεύοντας να διατηρήσει τις πιο σημαντικές πληροφορίες. Με άλλα λόγια αποτελεί μια συμπιεσμένη μορφή των αρχικών εισόδων.

Αποκωδικοποιητής (Decoder):
Ο αποκωδικοποιητής παίρνει τον κωδικό και προσπαθεί να ανακατασκευάσει τα αρχικά δεδομένα. Συνήθως χρησιμοποιεί μια συμμετρική δομή νευρωνικών επιπέδων με τον κωδικοποιητή. Ο στόχος είναι η ανακατασκευασμένη έξοδος να είναι όσο το δυνατόν πιο κοντά στην αρχική είσοδο.

Οι autoencoders εκπαιδεύονται με τη χρήση ανατροφοδότησης (backpropagation), προσαρμόζοντας τα βάρη του δικτύου ώστε να ελαχιστοποιήσουν τη διαφορά μεταξύ των αρχικών εισόδων και των ανακατασκευασμένων εξόδων. Η διαφορά αυτή ορίζεται συνήθως από μια συνάρτηση απώλειας, όπως η μέση τετραγωνική απόκλιση.

Ερώτημα Α
Ποια αρχιτεκτονική autoencoder πέτυχε το μικρότερο μέσο τετραγωνικό σφάλμα στα δεδομένα εκπαίδευσης; Αυτή που χρησιμοποιήθηκε στην εργαστηριακή άσκηση ή αυτή που εκπαιδεύσατε; Γιατί πιστεύετε ότι συνέβη αυτό;

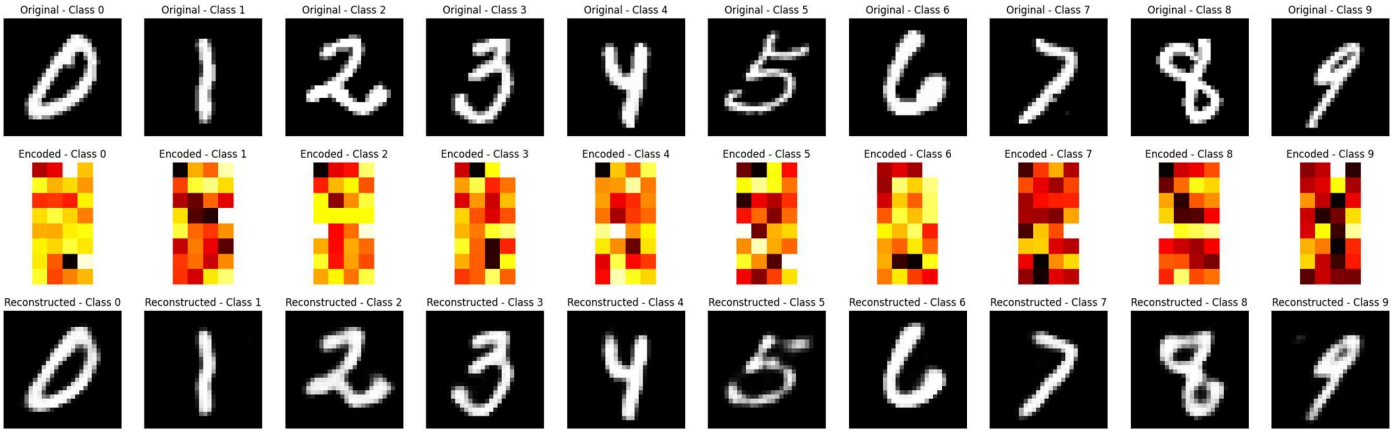
Προκειμένου να αποφανθώ για το μέσο τετραγωνικό σφάλμα ορίζω κατάλληλα το κριτήριο για την εκπαίδευση και των δύο δικτύων.

Μοντέλο εργαστηρίου	Μοντέλο άσκησης
Epoch [1/20], Loss: 0.0441	Epoch [1/20], Loss: 0.0742
Epoch [2/20], Loss: 0.0411	Epoch [2/20], Loss: 0.0645
Epoch [3/20], Loss: 0.0469	Epoch [3/20], Loss: 0.0647
Epoch [4/20], Loss: 0.0443	Epoch [4/20], Loss: 0.0701
Epoch [5/20], Loss: 0.0406	Epoch [5/20], Loss: 0.0743
Epoch [6/20], Loss: 0.0395	Epoch [6/20], Loss: 0.0637
Epoch [7/20], Loss: 0.0386	Epoch [7/20], Loss: 0.0660
Epoch [8/20], Loss: 0.0304	Epoch [8/20], Loss: 0.0710
Epoch [9/20], Loss: 0.0387	Epoch [9/20], Loss: 0.0716
Epoch [10/20], Loss: 0.0331	Epoch [10/20], Loss: 0.0721
Epoch [11/20], Loss: 0.0407	Epoch [11/20], Loss: 0.0624
Epoch [12/20], Loss: 0.0360	Epoch [12/20], Loss: 0.0645
Epoch [13/20], Loss: 0.0327	Epoch [13/20], Loss: 0.0636
Epoch [14/20], Loss: 0.0305	Epoch [14/20], Loss: 0.0646
Epoch [15/20], Loss: 0.0363	Epoch [15/20], Loss: 0.0662
Epoch [16/20], Loss: 0.0315	Epoch [16/20], Loss: 0.0684
Epoch [17/20], Loss: 0.0351	Epoch [17/20], Loss: 0.0702
Epoch [18/20], Loss: 0.0398	Epoch [18/20], Loss: 0.0673
Epoch [19/20], Loss: 0.0339	Epoch [19/20], Loss: 0.0619
Epoch [20/20], Loss: 0.0381	Epoch [20/20], Loss: 0.0692

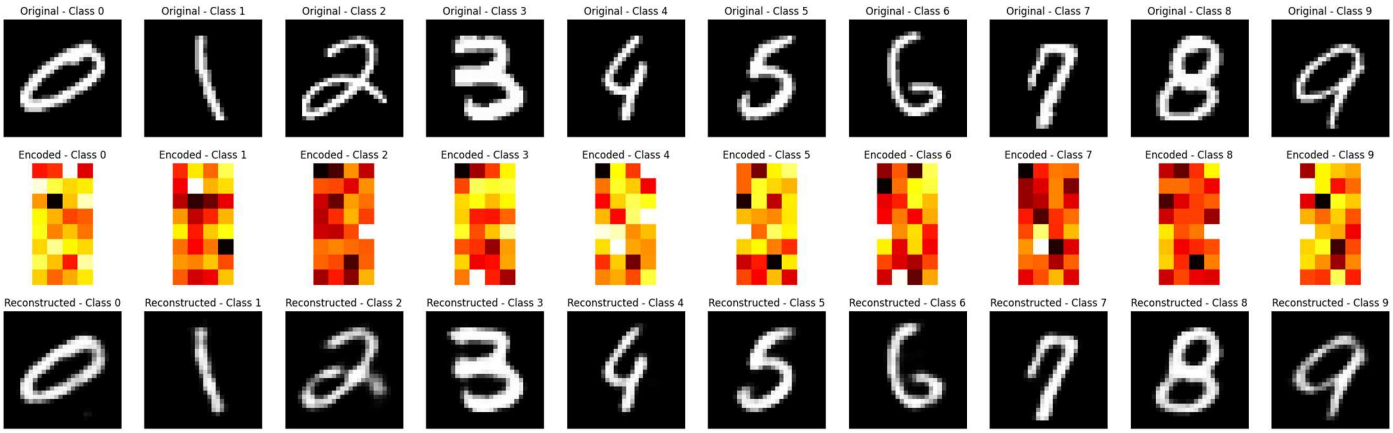
Παρατηρώ ότι χειρότερα αποτελέσματα δίνει το μοντέλο της άσκησης. Με την προσθήκη επιπλέον κρυφών επιπέδων αυξάνεται η πολυπλοκότητα του μοντέλου, γεγονός που ενδέχεται να οδηγήσει σε

υπερπροσαρμογή (overfitting). Ακόμη, η εκπαίδευση μεγάλων μοντέλων είναι λιγότερο αποτελεσματική εάν ο αλγόριθμος βελτιστοποίησης και ο ρυθμός εκμάθησης δεν είναι κατάλληλα προσαρμοσμένοι για ένα μοντέλο πολλών κρυφών επιπέδων. Δεδομένου ότι και τα δύο μοντέλα εκπαιδεύονται για το ίδιο πλήθος εποχών, μόλις 20, αυτό μπορεί να είναι αρκετό για το μοντέλο του εργαστηρίου αλλά όχι και για το μοντέλο της άσκησης που είναι πιο βαθύ.

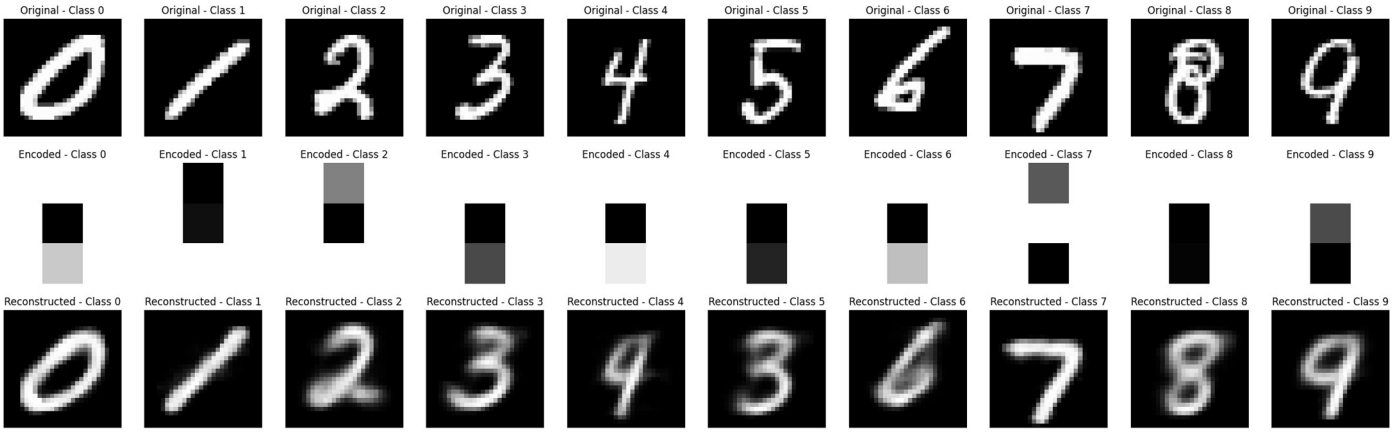
Σε συνέχεια αυτού του ερωτήματος, τροποποίησα την συνάρτηση του εργαστηρίου show_latten_space προκειμένου να απεικονίσω ένα δεδομένο για κάθε κλάση. Σε αυτήν τροφοδοτώ τόσο το μοντέλο του εργαστηρίου όσο και αυτό της άσκησης.



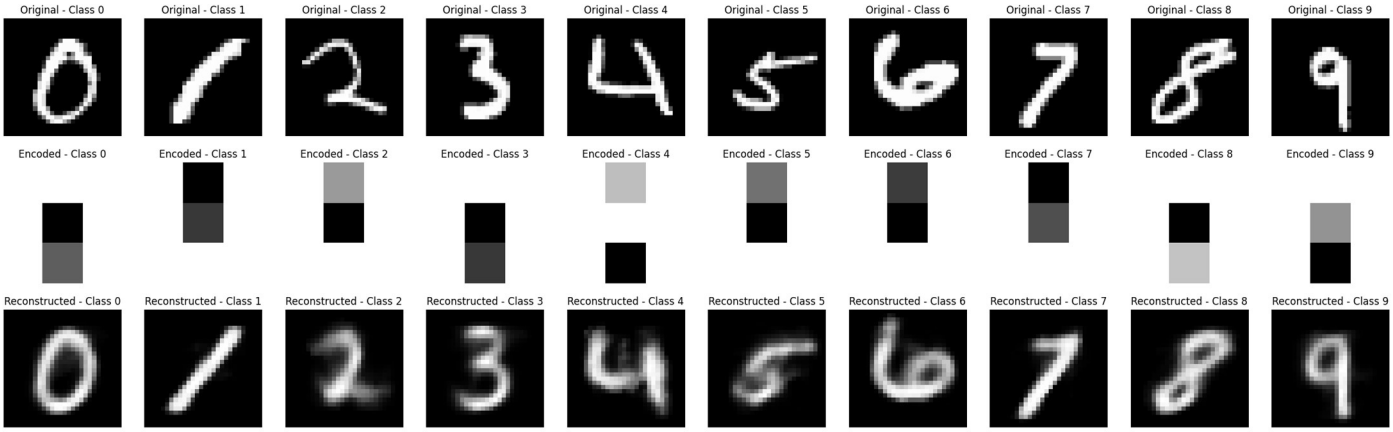
Εικόνα 10: Μοντέλο εργαστηρίου για δεδομένα train set.



Εικόνα 12: Μοντέλο εργαστηρίου για δεδομένα test set.



Εικόνα 13: Μοντέλο άσκησης για δεδομένα train set.



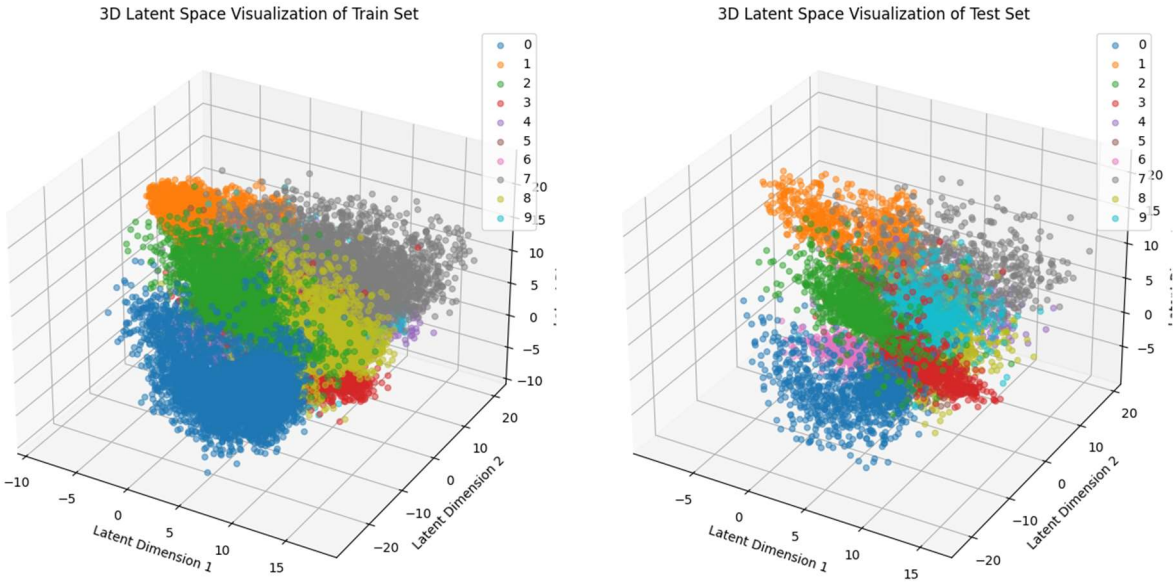
Εικόνα 14: Μοντέλο άσκησης για δεδομένα test set.

Συγκρίνοντας τις εικόνες 10 έως 14, συνειδητοποιώ ότι όντως το μοντέλο της άσκησης δίνει χειρότερα αποτελέσματα σε σχέση με εκείνο του εργαστηρίου τόσο για τα δεδομένα εκπαίδευσης όσο και για τα δεδομένα αξιολόγησης. Οι πιθανοί λόγοι έχουν προαναφερθεί.

Ερώτημα Β

Να απεικονίσετε κατάλληλα το σύνολο των εικόνων του training set όπως κωδικοποιούνται στο latent space, με κατάλληλο χρώμα που να αντιστοιχεί στο label καθεμιάς. Κάνετε το ίδιο και για τις εικόνες του test set. Διατηρούνται τα χαρακτηριστικά της κατανομής των απεικονίσεων και στα δύο σύνολα? Σχολιάστε.

Έχοντας μειώσει τις διαστάσεις των δεδομένων στις επιθυμητές του latten space, απεικονίζω σε αυτόν τον χώρο τα δεδομένα του train set και του test set τόσο για το μοντέλο της άσκησης.



Εικόνα 9: Αναπαράσταση στον χώρο του latent space των δεδομένων του train set (αριστερά) και του test set (δεξιά) για το μοντέλο της άσκησης.

Συγκρίνοντας τις δύο εικόνες παρατηρώ ότι διατηρούνται τα χαρακτηριστικά κατανομής και για τα δύο σύνολα (train και test set). Η μόνη διαφορά είναι ότι το διάγραμμα για το test set παρουσιάζει μεγαλύτερη αραιότητα.

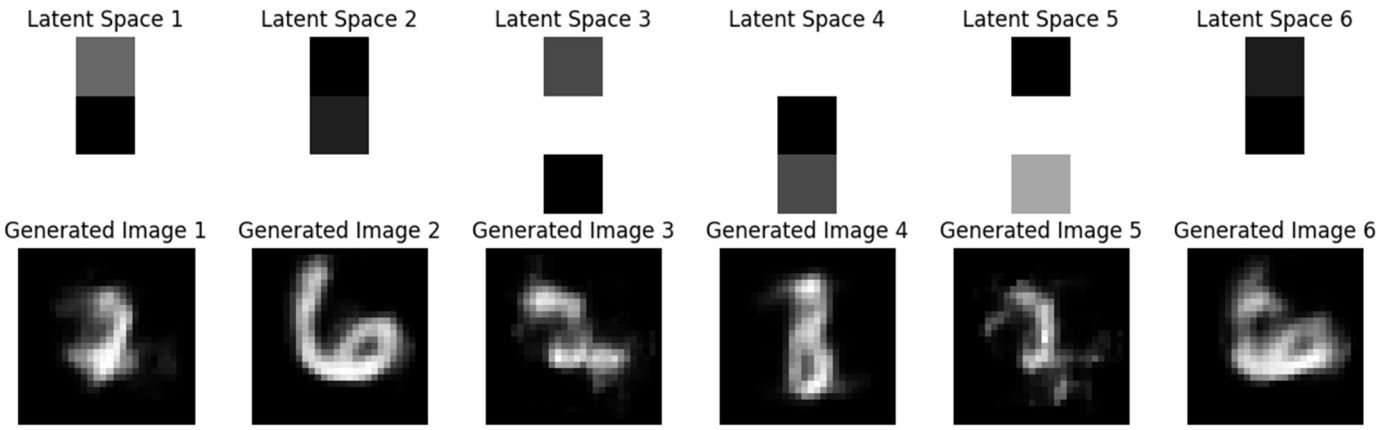
Ερώτημα Γ

Να χρησιμοποιήσετε τον κλάδο αποκωδικοποίησης του εκπαιδευμένου autoencoder ώστε να δημιουργήσετε εικόνες ξεκινώντας από τυχαία σημεία του latent space. Εξερευνήστε το latent space δοκιμάζοντας διάφορα σημεία του χώρου και παρατηρώντας τις εικόνες που παράγονται. Αντιστοιχούν όλες οι περιοχές του latent space σε πραγματικά ψηφία? Σχολιάστε τα ευρήματά σας παραθέτοντας παραδείγματα εικόνων.

Μετά την εκπαίδευση ενός μοντέλου encoder-decoder, ο κωδικοποιητής έχει μάθει να μετατρέπει τις εικόνες σε συμπυκνωμένη μορφή διαστάσεων που ορίζεται από το latent space, ενώ ο αποκωδικοποιητής έχει μάθει να ανακατασκευάζει εικόνες από αυτές τις μορφές.

Η δημιουργία νέων εικόνων από τυχαία σημεία του latent space υλοποιείται με την ανάθεση τυχαίων τιμών σε έναν τένσορα διαστάσεων που καθορίζονται από εκείνες του latent space. Οι τυχαίες αυτές τιμές ακολουθούν μία κατανομή η οποία ορίζεται από την συνάρτηση που επιλέγεται για την παραγωγή των τυχαίων τιμών. Στην παρούσα περίπτωση, επιλέχτηκε η συνάρτηση της βιβλιοθήκης pytorch, η οποία παράγει τυχαίες τιμές που ακολουθούν την κανονική κατανομή (γκαουσιανή κατανομή με μέσο όρο και τυπική απόκλιση 1). Θα σημειώνω ότι έγιναν προσπάθειες και με άλλες κατανομές, οι οποίες όμως δεν έδωσαν καλύτερα αποτελέσματα. Αυτές οι προσπάθειες βρίσκονται στο τέλος του σημειωματαρίου για την δεύτερη άσκηση. Οι κατανομές αυτές θα μπορούσαν να μελετηθούν περαιτέρω, ως προς τις τιμές των παραμέτρων και την επίδραση στο τελικό αποτέλεσμα. Ωστόσο, λόγω έλλειψης χρόνου και εφόσον δεν ζητείται στην εκφώνηση δεν έγινε περαιτέρω μελέτη.

Εφόσον έχουν παραχθεί τα τυχαία δεδομένα, τροφοδοτούνται στον εκπαιδευμένο αποκωδικοποιητή. Στην εικόνα 10 φαίνεται στο επάνω μέρος η απεικόνιση 6 τυχαίων δεδομένων στο latent space και στο κάτω μέρος το αποτέλεσμα της αποκωδικοποίησης.



Εικόνα 10: Αποτέλεσμα αποκωδικοποίησης τυχαίων σημείων του latent space

Από την εικόνα 10 συμπεραίνω ότι δεν αντιστοιχούν όλες οι περιοχές του latent space σε πραγματικά ψηφία. Συγκεκριμένα, στην εικόνα μονάχα το δεύτερο δείγμα είναι πιο κοντά σε έναν αριθμό, το 6. Τα υπόλοιπα, αν και μοιάζουν με κάποιον αριθμό, δεν είναι ιδιαίτερα κατανοητός.

ΑΣΚΗΣΗ 3

Το Breast Cancer Wisconsin είναι ένα σύνολο δεδομένων που περιέχει 30 μορφολογικά χαρακτηριστικά των καρκινικών κυττάρων από 569 βιοψίες μαστού, καθώς και το είδος του όγκου για κάθε δείγμα (κακοήθης: κλάση1 ή M \| καλοήθης: κλάση 0 ή B).

Το dataset αυτό βρίσκεται στην βιβλιοθήκη sklearn, οπότε το κάνω import από εκεί.

Ερώτημα Α
Να χωριστεί το dataset σε 70% training και 30% test δείγματα με τυχαίο τρόπο.

Ακολουθώ την ίδια λογική με εκείνη του πρώτου ερωτήματος της πρώτης άσκησης της παρούσας εργασίας.

Ερώτημα Β
Να μετατρέψετε το 10% των training δεδομένων σε missing values με τυχαία ομοιόμορφη κατανομή, και να εκπαιδεύσετε έναν δενδρικό ταξινομητή (tree classifier) με μέγιστο βάθος 5 επίπεδα. Υπολογίστε την ακρίβεια πρόβλεψης του ταξινομητή στο test set.

Τα missing values δημιουργούνται με την βοήθεια της `rand.uniform` της βιβλιοθήκης `numpy`, όπως ακριβώς αναφέρθηκε και στα πλαίσια του πέμπτου εργαστηρίου.

Τα δέντρα αποφάσεων είναι μία δομή δεδομένων ιεραρχικής φύσης που χρησιμοποιείται για την αποθήκευση πληροφοριών σε οργανωμένη μορφή. Ένα δέντρο αποτελείται από κόμβους και ακμές. Ο κορυφαίος κόμβος ονομάζεται ρίζα. Κάθε κόμβος μπορεί να ενώνεται με άλλους κόμβους κατώτερου επιπέδου οι οποίοι καλούνται παιδιά του. Ένας κόμβος ενδέχεται να μην έχει παιδιά. Ο αρχικός κόμβος καλείται πατέρας. Μονάχα η ρίζα του δέντρου δεν μπορεί να έχει πατέρα. Τα δέντρα των οποίων οι κόμβοι έχουν το πολύ δύο παιδιά καλούνται δυαδικά. Οι κόμβοι οι οποίοι δεν έχουν παιδιά ονομάζονται φύλλα του δέντρου. Η ιεραρχική αυτή δομή επιτρέπει την αποδοτική οργάνωση και αναζήτηση των δεδομένων.

Αναφορικά με την χρήση των δέντρων στο πεδίο της μηχανικής μάθησης, αυτά χρησιμοποιούνται ως μοντέλα αποφάσεων και για αυτό καλούνται δέντρα αποφάσεων. Με τα δέντρα μπορούμε να απεικονίσουμε μια σειρά αποφάσεων καθώς και των πιθανών εκβάσεών τους. Στα δέντρα απόφασης κάθε κόμβος αντιπροσωπεύει ένα ερώτημα πάνω σε ένα χαρακτηριστικό και οι ακμές προς τα παιδιά του τις δυνατές απαντήσεις. Σαφώς, τα φύλλα του δέντρου αντιπροσωπεύουν τις τελικές αποφάσεις.

Η δημιουργία ενός δενδρικού ταξινομητή γίνεται με την συνάρτηση `DecisionTreeClassifier` της `sklearn`. Εφόσον ορίσω το βάθος σύμφωνα με τα δεδομένα της άσκησης, εκπαιδεύω τον ταξινομητή στα τροποποιημένα δεδομένα εκπαίδευσης. Η ακρίβεια που επιτυγχάνει στα test data είναι αρκετή υψηλή για όλο το σύνολο των δοκιμών που έτρεξα.

Η υψηλή αυτή ακρίβεια είναι αναμενόμενη καθώς το Breast Cancer Wisconsin είναι καλά δομημένο και περιέχει σχετικά καθαρά και συγκεκριμένα δεδομένα με ισορροπημένη κατανομή των κλάσεων. Ακόμη, το βάθος του δέντρου σε μόλις 5 επίπεδα είναι αρκετό για να μάθει το μοντέλο μας χωρίς να υποπέσει σε κατάσταση υπερπροσαρμογής (overfitting). Σε αυτό βοηθάει επίσης και η εισαγωγή των missing values στα δεδομένα εκπαίδευσης. Τέλος, το γεγονός ότι τα missing values ακολουθούν τυχαία ομοιόμορφη κατανομή διασφαλίζει ότι δεν υπάρχει συστηματική προκατάληψη στον τρόπο με τον οποίο αντιμετωπίζονται τα δεδομένα.

Ερώτημα Γ
Με τα ίδια training και test sets να εκπαιδεύσετε ένα random forest (RF) με 100 ταξινομητές μέγιστου βάθους 3, χρησιμοποιώντας 5 features ανά δένδρο χωρίς bootstrapping των training δεδομένων. Να υπολογίσετε την ακρίβεια πρόβλεψης του RF στο test set.

Το Random Forest είναι ένας αλγόριθμος που αποτελείται από πολλά ίδια δέντρα αποφάσεων. Σε κάθε δέντρο δίνονται ορισμένα χαρακτηριστικά αυξάνοντας έτσι την δημιουργία διαφορετικών και ανεξάρτητων δέντρων αποφάσεων. Με άλλα λόγια, αυξάνεται η ανθεκτικότητα του μοντέλου. Κάθε δέντρο εκπαιδεύεται ανεξάρτητα στα χαρακτηριστικά που του αντιστοιχούν. Για την τελική απόφαση συνδυάζονται οι αποφάσεις όλων των δέντρων. Τελικά επικρατεί εκείνη με τις περισσότερες ψήφους (majority voting). Γενικά, η χρήση του Random Forest αναμένεται να οδηγήσει σε ένα πιο ανθεκτικό μοντέλο με μεγαλύτερη ικανότητα

γενίκευσης και άρα υψηλότερη ακρίβεια. Αυτό οφείλεται στο γεγονός ότι η συγκέντρωση πολλών δέντρων αποφάσεων επιφέρει την μείωση της διακύμανσης και της πιθανότητας υπερπροσαρμογής.

Ερώτημα Δ

Υπολογίστε το feature importance για τον δενδρικό ταξινομητή και το RF. Σχολιάστε τα μέχρι τώρα αποτελέσματα και τις διαφορές που παρατηρείτε.

Για την εύρεση της σημαντικότητας κάθε χαρακτηριστικού χρησιμοποίησα την αντίστοιχη συνάρτηση της βιβλιοθήκης. Αναφορικά με το Random Forest, σημειώνεται ότι η σημαντικότερα ενός χαρακτηριστικού καθορίζεται από το σύνολο της σημαντικότητας που παρουσιάζει σε όλα τα δέντρα που συμμετέχει.

Σημαντικά χαρακτηριστικά για το δέντρο	Σημαντικά χαρακτηριστικά για το δάσος
mean concave points: 0.4648	mean concave points: 0.0017
worst concave points: 0.2182	worst perimeter: 0.0016
worst radius: 0.0990	worst radius: 0.0010
worst area: 0.0977	worst concave points: 0.0009
mean radius: 0.0358	mean perimeter: 0.0008
worst texture: 0.0205	mean area: 0.0007
worst smoothness: 0.0159	worst area: 0.0006
mean texture: 0.0149	worst concavity: 0.0004
mean concavity: 0.0112	mean concavity: 0.0002
smoothness error: 0.0101	mean texture: 0.0002
mean fractal dimension: 0.0099	worst compactness: 0.0002
texture error: 0.0019	area error: 0.0002
mean symmetry: 0.0000	worst smoothness: 0.0002
worst concavity: 0.0000	worst texture: 0.0002
mean perimeter: 0.0000	radius error: 0.0002
mean area: 0.0000	worst symmetry: 0.0002
mean smoothness: 0.0000	mean compactness: 0.0001
mean compactness: 0.0000	perimeter error: 0.0001
worst compactness: 0.0000	mean radius: 0.0001
worst perimeter: 0.0000	mean smoothness: 0.0001
fractal dimension error: 0.0000	worst fractal dimension: 0.0001
symmetry error: 0.0000	fractal dimension error: 0.0000
radius error: 0.0000	mean symmetry: 0.0000
perimeter error: 0.0000	mean fractal dimension: 0.0000
area error: 0.0000	symmetry error: 0.0000
worst symmetry: 0.0000	smoothness error: 0.0000
compactness error: 0.0000	compactness error: 0.0000
concavity error: 0.0000	concavity error: 0.0000
concave points error: 0.0000	concave points error: 0.0000
worst fractal dimension: 0.0000	texture error: 0.0000

Σύμφωνα με τα παραπάνω αποτελέσματα, ο δενδρικός ταξινομητής φαίνεται πως βασίζεται σε λιγότερα χαρακτηριστικά τα οποία όμως παρουσιάζουν υψηλή σημαντικότητα. Σύμφωνα με το όνομα των χαρακτηριστικών αυτών, συμπεραίνω ότι ο δενδρικός ταξινομητής βασίζεται σε χαρακτηριστικά που αφορούν την μορφολογία και τις διαστάσεις του κυττάρου.

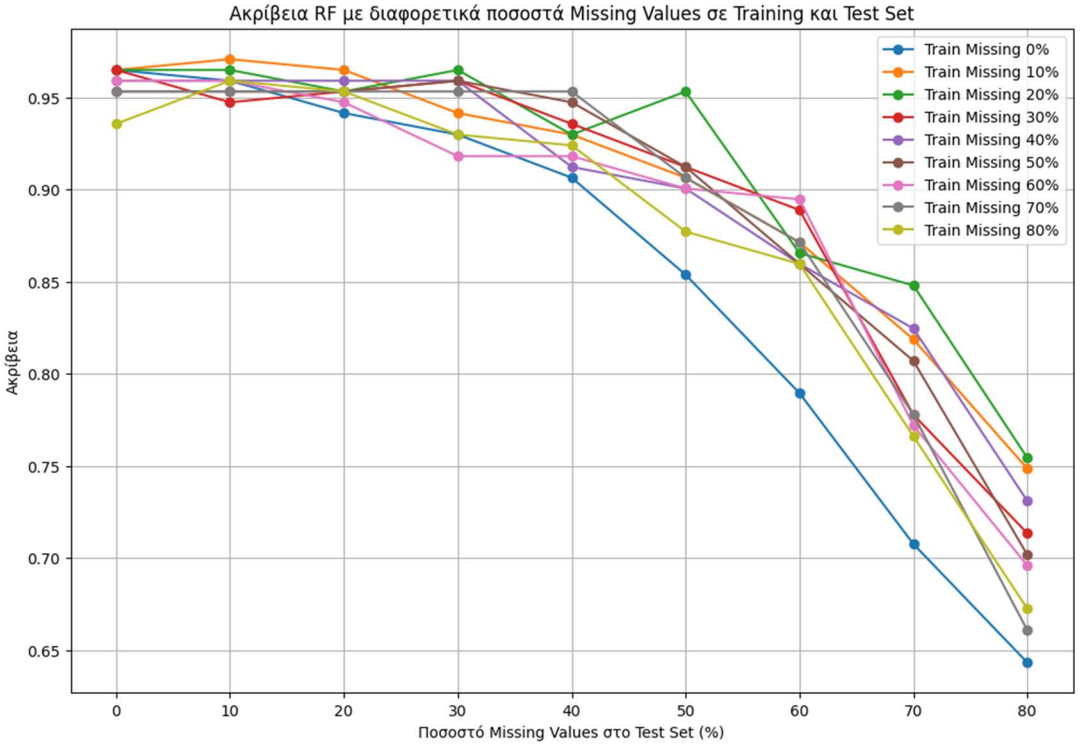
Αντίθετα, το Random Forest βασίζεται σε χαρακτηριστικά με αισθητά μικρότερη τιμή σημαντικότητας. Επομένως, το Random Forest λαμβάνει υπόψη μια ευρεία ποικιλία χαρακτηριστικών για την τελική απόφαση, καθώς κανένα χαρακτηριστικό δεν κυριαρχεί σημαντικά.

Από τα παραπάνω επιβεβαιώνεται ότι τα δέντρα αποφάσεων είναι πιο επιρρεπή στην υπερπροσαρμογή έναντι του Random Forest, το οποίο προσφέρει μια πιο ισορροπημένη προσέγγιση.

Ερώτημα Ε

Να εκπαιδεύσετε το ίδιο RF για μεταβλητό ποσοστό missing values (0% έως 80% ανά 10%) και για κάθε forest να υπολογίσετε το classification accuracy εισάγοντας στο test set μεταβλητό ποσοστό missing values (0% έως 80% ανά 10%). Απεικονίστε κατάλληλα τη συνολική συμπεριφορά της ακρίβειας του RF ταξινομητή σε σχέση με το ποσοστό missing values.

Αρχικά δημιουργώ πάλι ένα Random Forest όπως σε προηγούμενο ερώτημα, Το μοντέλο αυτό εκπαιδεύεται για ένα σύνολο δεδομένων στο οποίο έχει εισαχθεί ένα ποσοστό missing values. Έπειτα, το μοντέλο αξιολογείται για τα δεδομένα του test set στο οποίο πάλι έχει εισαχθεί ένα ποσοστό missing values.η διαδικασία αυτή υλοποιείται για όλους τους πιθανούς συνδυασμούς ποσοστών missing values στο train και στο test set. Στο τέλος δημιουργώ ένα διάγραμμα όπου απεικονίζεται η ακρίβεια του μοντέλου για τις διαφορετικά ποσοστά missing values στο test set.



Εικόνα 11: Ακρίβεια Random Forest με διαφορετικά ποσοστά missing values στα train και test set.

Στο διάγραμμα της εικόνας 11 κάθε γραμμή διαφορετικού χρώματος αντιστοιχεί στην εφαρμογή ενός διαφορετικού ποσοστού missing values στο train set. Το ποσοστό που αντιστοιχεί σε κάθε κατηγορία αναγράφεται στην λεζάντα.

Σχολιάστε την αντοχή του random forest στα missing data, τόσο στο training όσο και στο testing. Σε ποιο από τα δύο είδη missing data είναι περισσότερο ευαίσθητος.

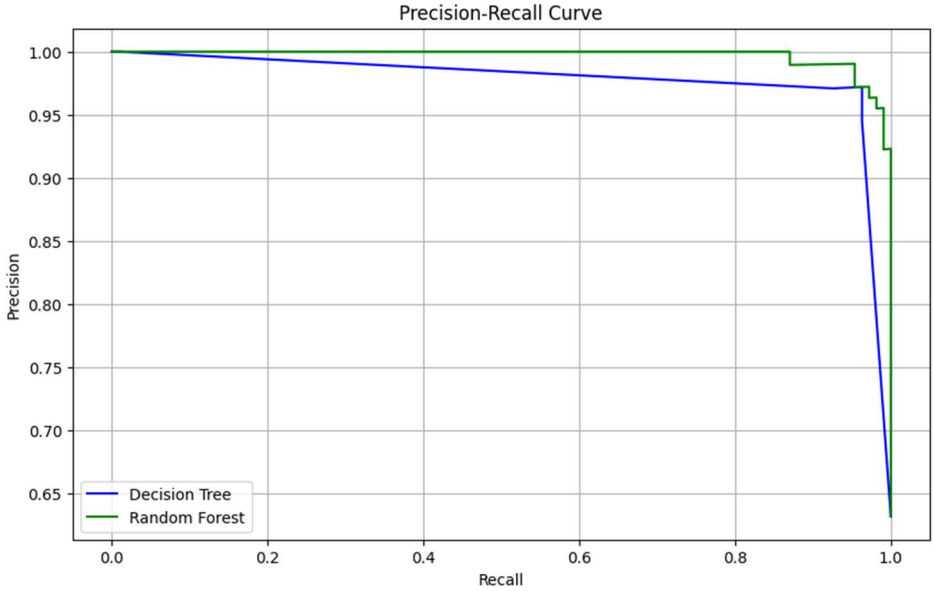
Σύμφωνα με την εικόνα 11, το Random Forest φαίνεται ότι διατηρεί σχετικά καλή ακρίβεια ακόμα και όταν υπάρχουν missing values στα δεδομένα εκπαίδευσης. Αυτό οφείλεται στην μέθοδο λειτουργίας των δέντρων αποφάσεων. Η ακρίβεια στο test set παρουσιάζει σταθερή μείωση της τιμής του καθώς αυξάνεται το ποσοστό missing values. Φαίνεται ότι η επίδραση των missing values είναι πιο έντονη στο test set παρά στο train set, ειδικά μάλιστα για πολύ υψηλά ποσοστά. Αυτό οφείλεται στο γεγονός ότι τα δέντρα έχουν εκπαιδευτεί να λαμβάνουν αποφάσεις βάσει των διαθέσιμων δεδομένων, η απουσία των οποίων στο test set ενδέχεται να παραπλανήσει το μοντέλο. Συνεπώς, το μοντέλο φαίνεται να παρουσιάζει μεγαλύτερη ευαισθησία στα missing data του test set παρά στο train set. Η προσαρμοστικότητα και η ανθεκτικότητα του μοντέλου στα missing values του train set δεν μεταφράζεται απαραίτητα και σε ισχυρή ανοχή στην ύπαρξη missing values και στο test set.

Ερώτημα ΣΤ

Να εκπαιδεύσετε έναν δενδρικό ταξινομητή και έναν RF ταξινομητή όπως στα προηγούμενα, για 10% missing data στο training set. Υπολογίστε τις καμπύλες precision-recall για τους δύο ταξινομητές (χωρίς missing data στο test set).

Ακολουθώ την ίδια διαδικασία για την δημιουργία και την εκπαίδευση του δενδρικού ταξινομητή και του Random Forest όπως στα προηγούμενα ερωτήματα. Με όμοιο τρόπο όπως και στο τελευταίο ερώτημα της πρώτης άσκησης δημιουργώ τις καμπύλες precision-recall.

Η ακρίβεια (precision) αναφέρεται στο ποσοστό των θετικών προβλέψεων που ήταν πραγματικά θετικές και η ευαισθησία (recall) αναφέρεται στο ποσοστό των θετικών περιπτώσεων που ταξινομήθηκαν σωστά. Το ιδανικό σενάριο είναι να έχω υψηλή ακρίβεια και υψηλή ευαισθησία.



Εικόνα 12: Καμπύλες Precision-Recall για τον δενδρικό ταξινομητή και το Random Forest.

Ποιος ταξινομητής έχει καλύτερη συμπεριφορά και γιατί?

Ως καλύτερος ταξινομητής κρίνεται εκείνος του οποίου η καμπύλη είναι πιο κοντά στην ιδανική. Σύμφωνα με το διάγραμμα της εικόνας 12 υψηλότερη ακρίβεια παρουσιάζει το Random Forest και μάλιστα σε όλο το φάσμα. Και οι δύο οι ταξινομητές ξεκινούν από πολύ υψηλή ακρίβεια. Ωστόσο, καθώς αυξάνεται η ανάκληση, η ακρίβεια του δένδρικού ταξινομητή μειώνεται πιο γρήγορα σε σχέση με εκείνη του Random Forest. Επομένως, ο ταξινομητής Random Forest είναι πιο ακριβής στην πρόβλεψη των θετικών κλάσεων και είναι λιγότερο πιθανό να κατατάξει αρνητικές περιπτώσεις ως θετικές, για αυτό η καμπύλη του είναι σχεδόν ιδανική.

Δεδομένου ότι οι ταξινομητές που εκπαιδεύσατε αποτελούν συστήματα πρόβλεψης της κακοήθειας όγκων, ποια χαρακτηριστικά της καμπύλης έχουν περισσότερη βαρύτητα κατά την πρακτική εφαρμογή?

Για ταξινομητές σε συστήματα πρόβλεψης της κακοήθειας όγκων τόσο η υψηλή ακρίβεια όσο και η υψηλή ευαισθησία είναι κρίσιμης σημασίας.

Υψηλή ακρίβεια σημαίνει ότι πολλές περιπτώσεις που ταξινομούνται ως κακοήθεις είναι όντως κακοήθεις. Έτσι, μειώνονται οι πιθανότητες ένας καλοήθης όγκος να ταξινομηθεί ως κακοήθης, γεγονός που ενδεχομένως να οδηγούσε σε άσκοπες επεμβάσεις και ψυχολογική ταλαιπωρία του ασθενούς, αλλά και περιττό κόστος για το σύστημα υγείας.

Υψηλή ευαισθησία αφορά το ποσοστό των πραγματικών κακοήθων όγκων που έχουν εντοπιστεί από τον ταξινομητή. Σαφώς, ο εντοπισμός όσο των δυνατόν περισσότερων κακοηθειών εξασφαλίζει ότι ο ασθενής πρόκειται να λάβει την αναγκαία θεραπεία το συντομότερο δυνατόν. Αντίθετα, χαμηλή ευαισθησία σημαίνει ότι οι περισσότεροι κακοήθεις όγκοι δεν θα εντοπιστούν, με αποτέλεσμα να αυξάνεται ο κίνδυνος για την υγεία του ασθενούς αφού καθυστερεί η έναρξη της θεραπείας.

Σε περιπτώσεις κακοήθειας όγκων θεωρώ ότι η υψηλή ευαισθησία είναι προτιμότερη, καθώς η παράλειψη της κακοήθειας έχει μεγαλύτερο κόστος (ενδεχόμενο απώλειας ανθρώπινης ζωής).