Απαλλακτική Εργασία Ανάλυσης Δεδομένων

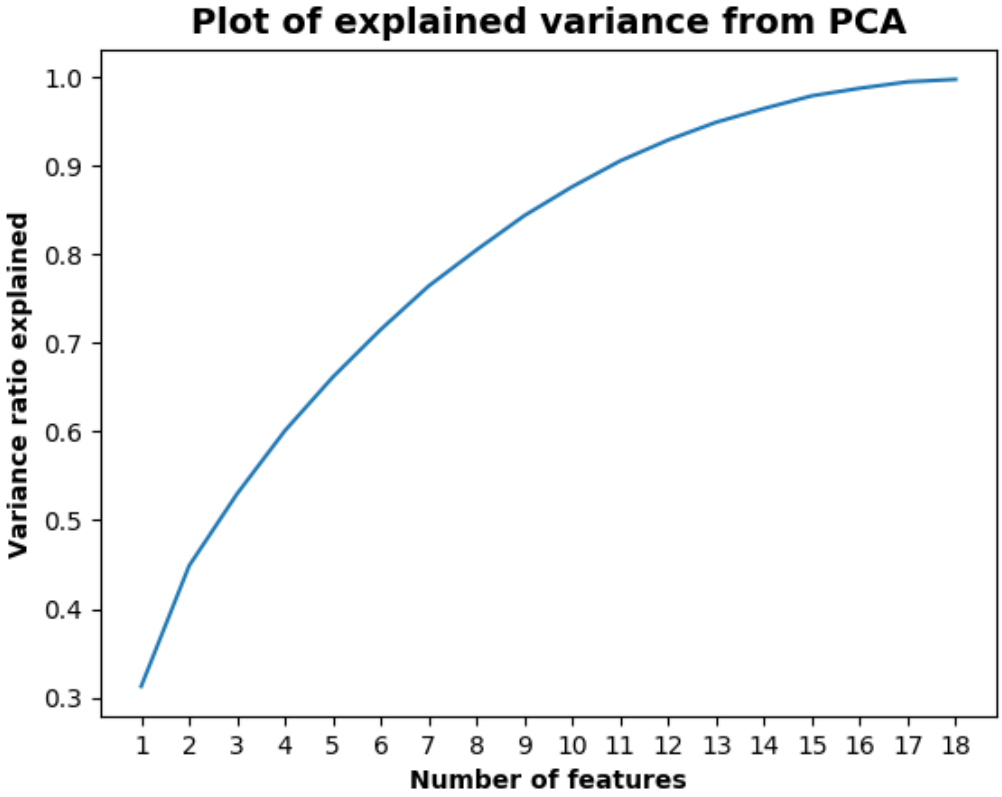
Άσκηση 1

Προπαρασκευή Δεδομένων

Το αρχείο για την πρώτη εργασία κατέβηκε από το UCI ML Repository και ήταν σε μορφή .xlsx το οποίο περιήχε 3 tabs. Για την ανάλυση των δεδομένων το tab με το όνομα 'data' χρησιμοποιήθηκε και φορτώθηκε στην Python, με τη χρήση της βιβλιοθήκης pandas. Τα δεδομένα αποτελούνταν από 2,129 γραμμές και 42 στήλες. Σύμφωνα με τις οδηγίες τις εργασίας, μόνο 21 κολώνες θα έπρεπε να χρησιμοποιηθούν σαν διάνυσμα χαρακτηριστικών και άλλες 2 κολώνες (CLASS, NSP) που περιέχουν τις ομάδες που οι αλγόριθμοι θα πρέπει να προβλέψουν. Άρα το πρώτο στάδιο προπαρασκευής δεδομένων ήταν ο καθαρισμός των στηλών που δεν θα χρησιμοποιηθούν για την ανάλυση.
Το επόμενο βήμα ήταν η αντιμετώπιση των ελλιπών τιμών. Παρατηρήθηκε ότι, όλες οι στήλες είχαν το πολύ 3 ελλιπής τιμές. Να σημειωθεί ότι υπάρχουν αρκετοί τρόποι να συμπληρωθούν οι ελλιπής τιμές, όπως να αντικατασταθούν από τον μέσο ή τη διάμεσο της στήλης που βρίσκονται, ωστόσο παρατηρήθηκε ότι αν διαγράψουμε τελείως τις γραμμές που περιέχουν κενές τιμές, δεν χάνουμε πολύ πληροφορία από τα δεδομένα. Συγκεκριμένα διαγράφονται μόνο 3 γραμμές, πράγμα που σημαίνει ότι όλες οι ελλιπής τιμές που παρατηρήθηκαν στις στήλες ήταν στις ίδιες γραμμές.
Στη συνέχεια, τα δεδομένα κανονικοποιήθηκαν. Η βιβλιοθήκη scikit-learn της python περιλαμβάνει την κλάση StandardScaler που χρησιμοποιείται για την κανονικοποίηση των δεδομένων. Συγκεκριμένα κάθε τιμή μετατρέπεται με βάση την παρακάτω εξίσωση:

$$x_i' = \frac{x_i - \bar{x}}{\sigma}$$

Όπου το x'$_i$ είναι η νέα τιμή, το x είναι η παλιά τιμή, το $\bar{x}$ είναι ο μέσος της στήλης και το σ είναι η τυπική απόκλιση της στήλης.
Στο τελικό στάδιο της προπαρασκευής των δεδομένων έγιναν πειράματα για τη μείωση των διαστάσεων των δεδομένων χρησιμποιώντα την κλάση PCA της βιβλιοθήκης scikit-learn που εφαρμόζει ανάλυση κυρίων συνιστωσών. Ο αριθμός των συνιστωσών επιλέχθηκε με βάση το ποσοστό της διακύμανσης των αρχικών δεδομένων που εξηγούν. Το παρακάτω διάγραμμα απεικονίζει πως μεταβάλεται το ποσοστό της διακύμανσης που εξηγέιται αναλόγως με τον αριθμό των συνιστωσών που επιλέγεται κάθε φορά.



Με βάση τη βιβλιογραφία, αν εξηγήται το 85% της διακύμανσης των δεδομένων τότε είναι ένας καλό αριθμός για την επιλογή του αριθμού των συνιστωσών. Με βάση αυτό επιλέχθηκαν 10 συνιστώσες που εξηγούν το 87.6% της συνολικής διακύμανσης των δεδομένων.

Ομαδοποίηση

Το επόμενο ζητούμενο της πρώτης άσκησης ήταν η επιλογή και εφαρμογή 3 αλγορίθμων ομαδοποίησης, καθώς και ο πειραματισμός με τις παραμέτρους τους ώστε να έχουν όσο το δυνατόν μεγαλύτερη ακρίβεια με βάση τις ομάδες που περιέχουν οι στήλες CLASS ή NSP. Λόγο του ότι η στήλη CLASS αποτελείται από 10 διαφορετικές κλάσεις, χρησιμοποιήθηκε η στήλη NSP για να ελέγξουμε την απόδοση των αλγορίθμων. Για να εξετάσουμε την απόδοση των αλγορίθμων, χρησιμοποιήσαμε τις ετικέτες τις στήλης NSP για να τις συγκρίνουμε με αυτές που θα έβγαιναν ως αποτέλεσμα από τους αλγόριθμους ομδοποίησης. Οι αλγόριθμοι συγκρίνονται μεταξύ τους χρησιμοποιώντας το μέτρο της ακρίβειας.

K-MEANS

Σαν πρώτο αλγόριθμο, χρησιμοποιήθηκε ο K-MEANS. Ο συγκεκριμένος αλγόριθμος, παίρνει σαν παράμετρο τον αριθμό των ομάδων που θέλουμε να χωριστούν τα δεδομένα. Ξεκινάει θέτοντας τυχαία 3 κέντρα, και αναθέτει τα δεδομένα στην ομάδα που το κέντρο της έχει τη μικρότερη απόσταση από την κάθε παρατήρηση. Στη συνέχεια τα κέντρα

ανανεώνονται με βάση τη μέση τιμή των παρατηρήσεων που βρίσκονται στην κάθε ομάδα. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να μην υπάρχουν μετακινήσεις των δεδομένων μεταξύ των ομάδων.
Από τη στιγμή που ο συγκεκριμένος αλγόριθμος παίρνει μόνο μια παράμετρο (των αριθμό των ομάδων), έγινε μόνο ένα πείραμα, όπου εξετάστηκε αν ο αλγόριθμος έχει καλύτερη απόδοση, χρησιμοποιώντας τα κανονικά δεδομένα ή τα δεδομένα μετά την ανάλυση κυρίων συνιστωσών. Αποδείχθηκε ότι χρησιμοποιώντας τις κύριες συνιστώσες ο αλγόριθμος πετυχαίνει ακρίβεια 0.472, ενώ με τα κανονικά δεδομένα 0.336. Τα αντίστοιχα confusion matrices είναι τα εξής:

	0	1	2
0	952	1	702
1	254	0	41
2	119	6	51

Πίνακας 1: Confusion matrix χρησιμοποιώντας τις κύριες συνιστώσες. Στις γραμμές είναι οι πραγματικές ετικέτες και στις στήλες οι ετικέτες από τους αλγόριθμους.

	0	1	2
0	569	991	95
1	244	40	11
2	69	2	105

Πίνακας 2: Confusion matrix χρησιμοποιώντας τα κανονικά δεδομένα. Στις γραμμές είναι οι πραγματικές ετικέτες και στις στήλες οι ετικέτες από τους αλγόριθμους.

DBSCAN
Ο επόμενος αλγόριθμος που χρησιμοποιήθηκε είναι ο DBSCAN. Ο συγκεκριμένος αλγόριθμος, προσπαθεί να ομαδοποιήσει τα δεδομένα που έχουν κοινά χαρακτηριστικά μεταξύ τους και είναι ικανός να βρει σχήματα οποιασδήποτε μορφής στα clusters. Ο DBSCAN λαμβάνει υπόψιν του 2 μεταβλητές για την ομαδοποίηση των δεδομένων:
1. Την απόσταση (eps): 2 σημεία θεωρούνται γείτονες αν η απόσταση μεταξύ τους είναι μικρότερη ή ίση από μια προκαθορισμένη τιμή
2. Τον ελάχιστο αριθμό των παρατηρήσεων που ένα cluster μπορεί να έχει (MinPoints).

Με βάση αυτά τα 2 οι παρατηρήσεις χωρίζονται σε 3 κατηγορίες:
1. Core Points: Είναι οι παρατηρήσεις που έχει το λιγότερο MinPoints παρατηρήσεις γύρω τους, σε μια περιοχή με ακτίνα eps
2. Border Points: Είναι οι παρατηρήσεις που βρίσκονται σε απόσταση eps από κάποιο Core Point αλλά δεν έχουν MinPoints παρατηρήσεις γύρω τους, σε μια περιοχή με ακτίνα eps
3. Outlier: Είναι οι παρατηρήσεις που δεν καλύπτουν καμία από τις παραπάνω περιπτώσεις.

Ο αλγόριθμος, ξεκινάει διαλέγοντας στην τύχη ένα σημείο και καθορίζει την γειτονιά του με βάση την απόσταση (eps) που έχει δωθεί από τον χρήστη. Έπειτα το σημείο

χαρακτηρίζεται σαν Core, Border ή Outlier. Αν χαρακτήριστεί ως Core, αρχίζει η σύσταση του cluster. Όλα τα σημεία στη γειτονία του μπαίνουν στα ίδιο cluster. Αν περιέχονται και άλλα Core points μέσα στο cluster, τότε και η γειτονιά των υπόλοιπων Core points, μπαίνει στην ίδια ομάδα. Η διαδικασία αυτή επαλαμβλανεται μέχρι να εξταστούν όλες οι παρατηρήσεις των δεδομέμων.
Εξετάστηκαν πολλές διαφορετικές τιμές για τις 2 μεταβλητές, ωστόσο κανένας συνδυασμός δεν μπορούσε να βοηθήσει τον αλγόριθμο να καταλήξει σε παραπάνω από ένα cluster όταν χρησιμοποιήθηκαν τα κανονικά δεδομένα. Στην συνέχεια χρησιμοποιήθηκαν τα δεδομένα που προήλθαν από την ανάλυση κυριών συνιστωσών και δοκιμάζοντας διάφορες τιμες για τις 2 μεταβλητές, ο αλγόριθμος κατάφερε να ομαδοποίηση τα δεδομένα σε 2 clusters πετυχαίνοντας ακρίβεια 0.14. Να σημειωθεί ότι μεγαλύτερη επιρροή στα αποτελέσματα έχει η μεταβλητή eps, καθώς όταν τέθηκε ίση με 0.2 όσο και να άλλαζε η μεταβλητή MinPoints, οι ομάδες έμεναν αμετάβλητες.

	0	1	2
0	1	1654	0
1	0	295	0
2	6	170	0

Πίνακας 3: Confusion matrix χρησιμοποιώντας τις κύριες συνιστώσες. Στις γραμμές είναι οι πραγματικές ετικέτες και στις στήλες οι ετικέτες από τους αλγόριθμους.

OPTICS
Ο αλγόριθμος OPTICS εντάσεται στην ίδια κατηγορία με τον DBSCAN (density-based clustering) αλλά έχει ακόμα 2 έννοιες:

1. Core distance: Είναι η ελάχιστη απόσταση (eps) που πρέχει να έχει ένα σημείο από τουλάχιστον MinPoints σημεία, για να θεωρηθεί ως Core
2. Reachability distance: Το reachability distance ενός σημείου ο από ένα άλλο p, είναι η ελάχιστη απόσταση από το p αν το p είναι Core Point. Επίσης αυτή η απόσταση δεν πρέπει να είναι μικρότερη από το Core distance του p.

Για τον συγκεκριμένο αλγόριθμο, ελέγχθηκαν αρκετοί συνδυασμοί ώστε να ο αλγόριθμος να καταλήγει σε 3 ομάδες. Για τα κανονικά δεδομένα η παράμετρος του ελάχιστου αριθμου γειτόνων για να ένα σημείο ώστε να θεωρηθεί Core τέθηκε ίση με 14 για το κανονικά δεδομένα και 15 για τις κύριες συνιστώσες. Η μεταβλητή της απόστασης 2 σημείων ώστε να θεωρηθούν στην ίδια γειτονιά τέθηκε ίση με 3 και στις 2 περιπτώσης.
Η ακρίβεια του αλγόριθμου χρησιμοποιώντας τα κανονικά δεδομένα έφτασε στο 0.77 ενώ με τη χρήση κυρίων συνιστωσών 0.76. Τα αντίστοιχα confusion matrices, παρουσιάζονται στους επόμενους πίνακες:

	0	1	2
0	1627	14	14
1	283	12	0
2	176	0	0

Πίνακας 4: Confusion matrix χρησιμοποιώντας τα κανονικά δεδομένα. Στις γραμμές είναι οι πραγματικές ετικέτες και στις στήλες οι ετικέτες από τους αλγόριθμους.

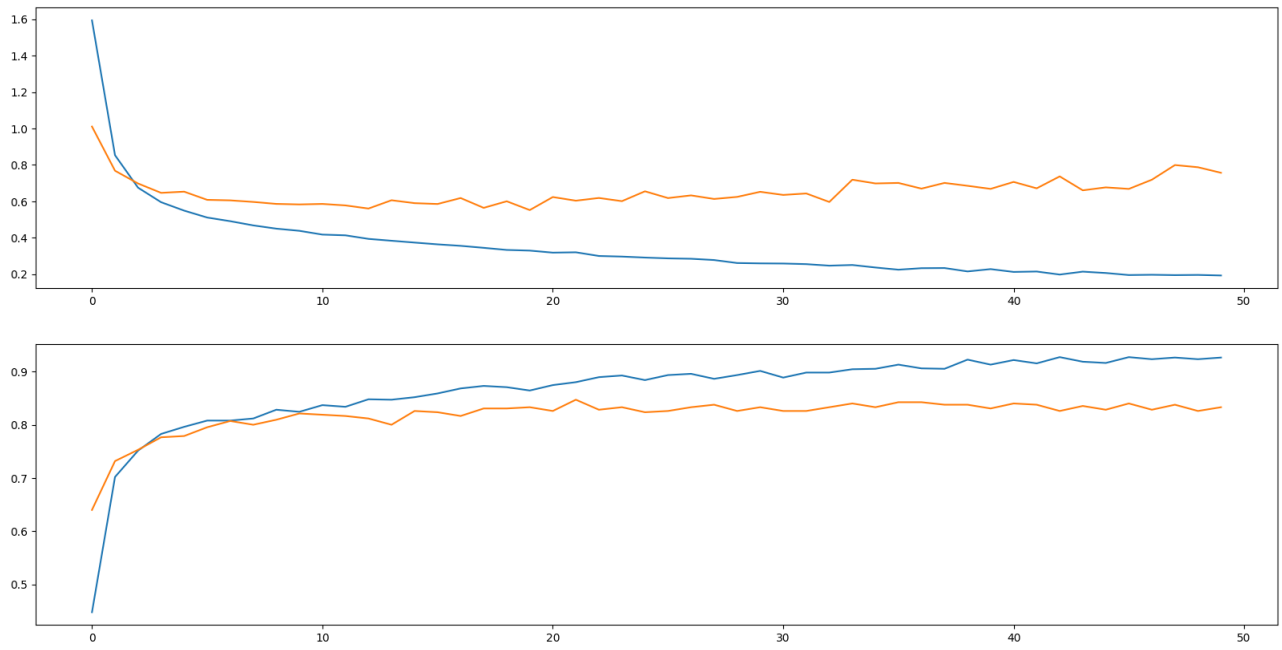
	0	1	2
0	1615	21	19
1	294	1	0
2	175	1	0

Πίνακας 5: Confusion matrix χρησιμοποιώντας τις κύριες συνιστώσες. Στις γραμμές είναι οι πραγματικές ετικέτες και στις στήλες οι ετικέτες από τους αλγόριθμους.

Κατηγοριοποίηση με νευρωνικό δίκτυο MLP

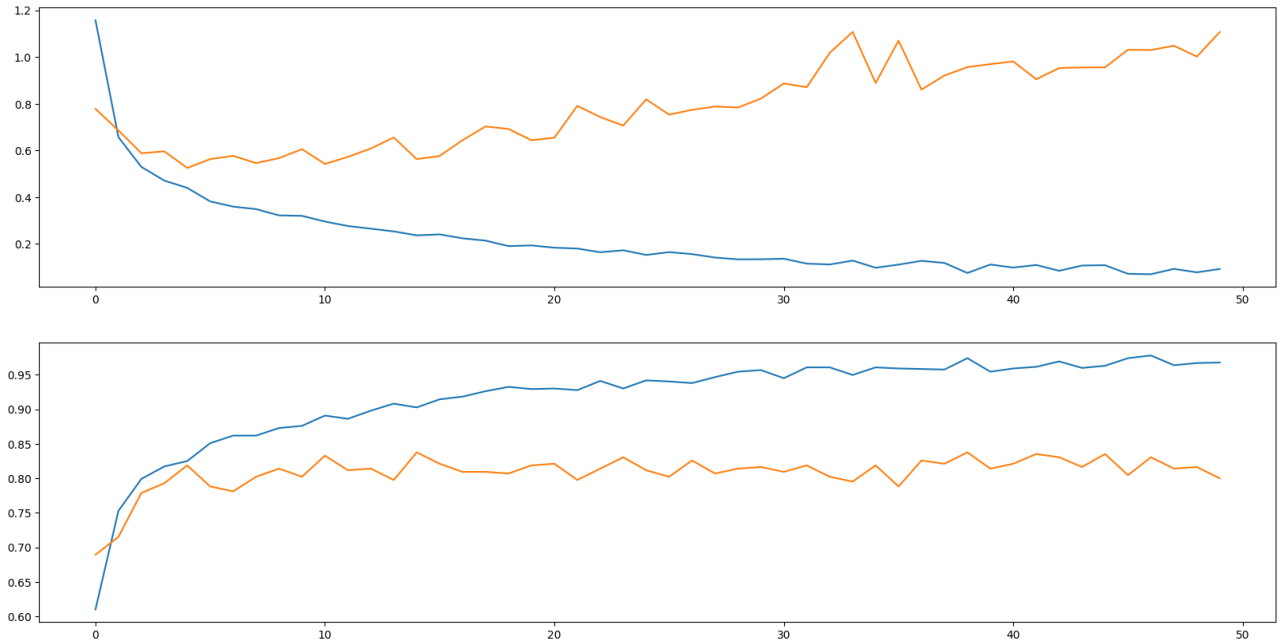
Το αντικείμενο αυτής της άσκησης είναι η κατηγοριοποίηση των δεδομένων με βάση την στήλη 'CLASS' η οποία περέχει 10 διαφορετικές κλάσεις. Διεξήχθηκαν διάφορα πειράματα όσων αφορά την αρχιτεκτονική του νευρωνικού δικτύου ωστόσο οι νευρώνες του στρώματος εισόδου και εξόδου έμειναν σταθεροί στους 21 και 10 αντίστοιχα. Σε όλες τις στοιβάδες χρησιμοποιήθηκε η relu ως activation function, εκτός από την τελευταία που χρησιμοποιήθηκε η softmax, μιας και έχουμε να προβλέψουμε παραπάνω από μια κλάσεις. Τα πειράματα αφορούσαν τον αριθμό των νευρώνων για τις 2 κρυφές στοιβάδες. Να σημειωθεί ότι ο αριθμός των εποχών σε όλα τα πειράματα ήταν σταθερός στις 50, σαν optimizer χρησιμοποιήθηκε ο adam και σαν loss function η categorical crossentropy.

Στο πρώτο πείραμε χρησιμοποίηθηκαν 10 νευρώνες και στις 2 κρυφές στοιβάδες. Τα αποτελέσματα ήταν αρκετά καλά καθώς το μοντέλο κατάφερε να πιάσει ακρίβεια 94.7% κατά τη διάρκεια της εκπαίδευσης και 84.5% κατά τη διάρκεια του test. Τα αντίστοιχα διαγράμματα φαίνονται παρακάτω



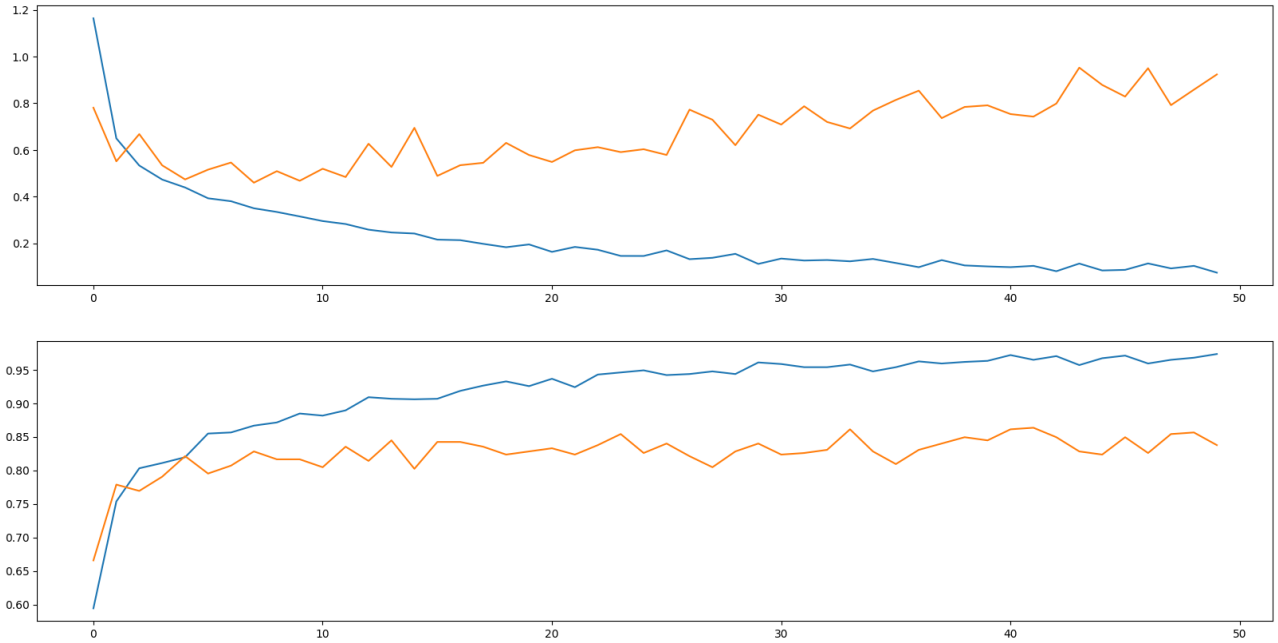
Εικόνα 2: Τα διαγράμματα loss function (πάνω) και ακρίβειας (κάτω) με το πέρασμα των εποχών όταν χρησιμοποιήθηκαν 10 νευρώνες στις 2 κρυφές στοιβάδες.

Στη συνέχεια για οι νευρώνες στις κρυφές στοιβάδες αυκήθηκαν στους 50 για να παρατηρηθεί η απόδοση ενός πολύπλοκου μοντέλου με περισσότερες παραμέτρους. Το συγκεκριμένο μοντέλο είχε ελάχιστα χειρότερη απόδοση από το προηγούμενο κατά τη διάρκεια του test (81.6%) και ελάχιστα καλύτερη κατά την εκπαίδευση (94.9%). Το φαινόμενο αυτό δημιούργησε υποψίες overfitting γι αυτό το επόμενο πείραμα ήταν με 100 νευρώνες.



Εικόνα 3: Τα διαγράμματα loss function (πάνω) και ακρίβειας (κάτω) με το πέρασμα των εποχών όταν χρησιμοποιήθηκαν 50 νευρώνες στις 2 κρυφές στοιβάδες.

Το μοντέλο με τους 100 νευρώνες στις κρυμμένες στοιβάδες πέτυχε ακρίβεια 96.78% κατά την εκπαίδευση και 82.8% κατά τη διάρκεια του test.



Εικόνα 4: Τα διαγράμματα loss function (πάνω) και ακρίβειας (κάτω) με το πέρασμα των εποχών όταν χρησιμοποιήθηκαν 100 νευρώνες στις 2 κρυφές στοιβάδες.

Από τα πειράματα δεν φαίνεται κάποια σημαντική διαφορά στις αποδόσεις των μοντέλων.

Άσκηση 2

Προπαρασκευή Δεδομένων

Για τα δεδομένα Istanbul Stock Exchange, σύμφωνα με τις οδηγίες, κρατήθηκαν οι πρώτες 530 γραμμές, κανονικοποιήθηκαν και δημιουργήθηκε μια συνάρτηση που μετασχηματίζει στη μορφή [samples, steps, features].

Time series prediction με MLP

Το πρώτο αντικείμενο της δεύτερης άσκησεις είναι να προβλέψουμε τις μελλοντικές τιμές μιας χρονοσειράς. Μιας και πρόκειται για πρόβλημα παλλινδρόμησης για την αξιολόγηση των μοντέλων χρησιμοποιήθηκαν οι εξής στατιστικοί δείκτες:

1. Mean Squared Error (MSE)
2. R-square

Τα πειράματα έγιναν κρατώντας σταθερή την αρχιτεκτονική του μοντέλου, ώστε να διαπιστωθεί κατά πόσο επηρεάζεται η απόδοση του αλγόριθμου από τις αλλαγές στα δεδομένα (αλλάζοντας το βήμα). Συγκεκριμένα, το δίκτυο αποτελείται από τη στοιβάδα εισόδου με αριθμό νευρώνων ίσο με τον αριθμό των παρατηρήσεων των δεδομένων, και με activation function ReLU. Στη συνέχεια, ακολουθεί μια κρυμμένη στοιβάδα με 50 νευρώνες και ίδια activation function και στο τέλο υπάρχει η στοιβάδα εξόδου με ένα νευρώνα και γραμμική activation function. Σαν loss function ορίστηκε το MSE, σαν optimiser ο Adam και το κάθε δίκτυο εκπαιδεύτηκε για 50 εποχές. Για να αξιολογηθεί η απόδοση των μοντέλων, τα δεδομένα χωρίστηκαν σε 2 σύνολα. Τα δεδομένα εκπαίδευσης (80%) και τα δεδομένα του test (20%). Στον παρακάτων πίνακα φαίνονται οι επιδόσεις του μοντέλου για τα διάφορα βήματα που δοκιμάστηκαν.

	Train				Test		
	1	3	6		1	3	6
MSE	0.0001	0.0001	0.0001		0.0002	0.0002	0.0003
R- square	0.81	0.78	0.64		0.34	0.38	0.33

Πίνακας 6: Τα αποτελέσματα από τα πειράματα με τη χρήση MLP.

Time series prediction με SimpleRNN

Στο δεύτερο μέρος αυτής της άσκησης, προσπαθούμε να προβλέψουμε την τιμή της χρονοσειράς με τη χρήση ενός νευρωνικού δικτύου που περιέχει RNN cells. Τα RNN cells είναι ιδανικά για τις χρονοσειρές καθώς έχουν την ιδιότητα να κρατούν στη μνήμη πληροφορία από προηγούμενες χρονικές περιόδους.
Για να είναι δυνατή η σύγκριση με το παραπάνω δίκτυο, τα πειράματα έγιναν αλλάζοντας τις χρονικές στιγμές από το παρελθόν που βλέπει το δίκτυο. Η αρχιτεκτονική του δικτύου περιλαμβάνει ένα στρώμα εισόδου με RNN units, ένα κρυφό fully connected layer με 50 νευρώνες και ένα στρώμα εξόδου με ένα νευρώνα. Το κάθε δίκτυο εκπαιδεύεται για 50 εποχές, σαν optimizer χρησιμοποιείται ο adam που προσπαθεί να ελαχιστοποιήσει το mean square error. Στον παρακάτω πίνακα φαίνονται οι επιδόσεις των μοντέλων.

	Train				Test		
	1	3	6		1	3	6
MSE	0.0001	0.0001	0.00006		0.0002	0.0002	0.0002
R- square	0.81	0.76	0.85		0.296	0.45	0.52

Πίνακας 7: Τα αποτελέσματα από τα πειράματα με τη χρήση RNN units

Από τα παραπάνω πειράματα φαίνεται ότι κανένα μοντέλο δεν έχει καλή απόδοση αφού σχεδόν όλα τα R-square είναι κάτω του 0.5. Ωστόσο παρατηρείται ότι όσο περισσότερα παρελθοντικά βήματα δώσουμε στα μοντέλα τόσο καλύτερα πηγαίνουν κατά τη διάρκεια του test.