Ατομική Διπλωματική Εργασία

ΠΡΟΒΛΕΨΗ ΔΕΥΤΕΡΟΤΑΓΟΥΣ ΔΟΜΗΣ ΠΡΩΤΕΪΝΩΝ ΜΕ ΤΗ ΧΡΗΣΗ ΒΑΘΙΩΝ ΥΠΟΛΕΙΠΟΜΕΝΩΝ ΔΙΚΤΥΩΝ

Αντρούλλα Χρυσοστόμου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόβλεψη Δευτεροταγούς Δομής Πρωτεϊνών με τη χρήση βαθιών υπολειπόμενων δικτύων

Αντρούλλα Χρυσοστόμου

Επιβλέπων Καθηγητής

Δρ. Χρίστος Χριστοδούλου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2019

Ευχαριστίες

Πρώτα απ' όλα θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, Δρ. Χρίστος Χριστοδούλου που με έχει εμπιστευθεί για την εκπόνηση της παρούσας ατομικής εργασίας. Θέλω να τον ευχαριστήσω για την συνεχή υποστήριξή, την καθοδήγηση και τις πολύτιμες συμβουλές που μου πρόσφερε καθ' όλη την διάρκεια της εργασίας αυτής.

Θα ήθελα, επίσης να ευχαριστήσω τον διδακτορικό φοιτητή Μιχάλη Αγαθοκλέους και τον μεταπτυχιακό φοιτητή Διονυσίου Ανδρέας οι οποίοι συνέλαβαν και αυτοί με τις γνώσεις και τις εμπειρίες τους στο να ολοκληρωθεί αυτή η εργασία. Η συνεχής βοήθεια και οι διάφορες συμβουλές τους που μου παρείχαν με βοήθησαν σε μεγάλο βαθμό έτσι ώστε να επιλυθούν αρκετά προβλήματα που υπήρξαν καθ' όλη την διάρκεια της έρευνας.

Τέλος, ένα μεγάλο ευχαριστώ στην οικογένεια μου για την ενθάρρυνση και την στήριξη που μου πρόσφερε σε οποιαδήποτε δυσκολία που αντιμετώπισα μέχρις στιγμής.

Περίληψη

Στόχος της διπλωματικής εργασίας είναι η προσπάθεια επίλυσης του προβλήματος πρόβλεψης της δευτεροταγούς δομής των πρωτεϊνών με τη χρήση βαθιών υπολειπόμενων νευρωνικών δικτύων. Οι πρωτεΐνες αποτελούν ένα από τα βασικότερα συστατικά κάθε ζωντανού οργανισμού. Ο ρόλος τους είναι αρκετά σημαντικός διότι αυτές καθορίζουν τις λειτουργίες του οργανισμού. Επομένως η γνώση της δομής της πρωτεΐνες έχει τεράστια σημασία. Συγκεκριμένα η δομή της πρωτεΐνης χωρίζεται σε τέσσερα επίπεδα, την πρωτοταγή, τη δευτεροταγή, την τριτοταγή και την τεταρτοταγή δομή. Σημαντικότερη είναι η δομή στο τρισδιάστατο χώρο, η τριτοταγής δομή ,αφού η συγκεκριμένη δομή καθορίζει το βιολογικό ρόλο της πρωτεΐνης. Συνεπώς γνωρίζοντας την λειτουργία της πρωτεΐνης μπορούν να δημιουργηθούν διάφορες θεραπείες για την αντιμετώπιση πολλών παθήσεων. Δυστυχώς, οι μεθοδολογίες εξαγωγής της τρισδιάστατης δομής των πρωτεϊνών που έχουν αναπτυχθεί μέχρι σήμερα είναι αρκετά πολύπλοκες και χρονοβόρες διαδικασίες. Για να εξαχθεί η τριτοταγής δομή πρέπει πρώτα να καθοριστεί η δευτεροταγής δομή και γι' αυτό μελετούμε την δευτεροταγής δομή της πρωτεΐνης. Η πρόβλεψη της δευτεροταγούς δομής εξάγεται από την πρωτοταγής δομή η οποία αποτελείται από μια αλληλουχία αμινοξέων. Η παρούσα έρευνά ασχολείται με την μελέτη των βαθιών υπολειπόμενων δίκτυών και πώς αυτά μπορούν να βοηθήσουν στην επίλυση του προβλήματος της πρόβλεψης της δευτεροταγούς δομής των πρωτεϊνών. Πιο συγκεκριμένα το δίκτυο που μελετήθηκε παίρνει σαν είσοδο την πρωτοταγή δομή η οποία επεξεργάστηκε και τροποποιήθηκε και έπειτα το δίκτυο θα εκπαιδεύεται ως συνέπεια να εξαχθεί η προβλεπόμενη δευτεροταγής δομή. Τα βαθιά υπολειπόμενα δίκτυα ανήκουν στην κατηγορία των βαθιών νευρωνικών δικτύων που στην ουσία αποτελούνται από συνελικτικά επίπεδα με επιπρόσθετες συνδέσεις μεταξύ τους. Επιλεχθήκαν τα συγκεκριμένα δίκτυα διότι δεν ξαναχρησιμοποιηθήκαν για το συγκεκριμένο πρόβλημα λόγω του ότι έχουν εφευρεθεί πολύ πρόσφατα. Λόγω του ότι τα δίκτυα αυτά μπορούν να επιλύσουν κάποια προβλήματα που αντιμετωπίζουν τα βαθιά νευρωνικά δίκτυα μέχρις στιγμής. Το υψηλότερο ποσοστό επιτυχίας ήταν 74.26% για το αρχείο fold8, το οποίο είναι ένα υπόδειγμα του συνόλου δεδομένων CB513.

Περιεχόμενα

Κεφάλαιο **1** **Εισαγωγή**..... **1**

1.1 Η σημασία και ο σκοπός της έρευνας 2

1.2 Προηγούμενες σχετικές έρευνες 4

Κεφάλαιο **2** **Προ απαιτούμενη γνώση - Υπόβαθρο** **9**

2.1 Βιολογικό υπόβαθρο 10

2.1.1 Πρωτεΐνες και αμινοξέα 10

2.1.2 Δομή της πρωτεΐνης 13

2.1.2.1 Πρωτοταγής δομή 13

2.1.2.2 Δευτεροταγής δομή 14

2.1.2.3 Τριτοταγής δομή 15

2.1.2.4 Τεταρτοταγής δομή 17

2.1.3 Βιολογικός ρόλος της πρωτεΐνης 18

2.2 Τεχνητά Νευρωνικά Δίκτυα 19

2.2.1 Γενικά 19

2.2.2 Βιολογική έμπνευση 20

2.2.3 Αρχιτεκτονική και Λειτουργία 22

2.2.4 Εκπαίδευση νευρωνικών δικτύων 26

2.2.5 Μοντέλο Νευρώνα McCulloh και Pitts 28

2.2.6 Συναρτήσεις ενεργοποίησης 30

2.2.7 Πολυστρωματικά Δίκτυα Perceptron 33

2.2.7.1 Αλγόριθμος μάθησης Perceptron 35

2.2.7.2 Μέθοδος κατάβασης κλίσης 36

2.2.7.3 Αλγόριθμος ανάστροφης μετάδοσης σφάλματος 38

2.2.8 Συνελικτικά Νευρωνικά Δίκτυα 40

2.2.8.1 Γενικά 40

2.2.8.2 Αρχιτεκτονική 43

2.2.9 Υπολειπόμενα Νευρωνικά Δίκτυα 55

2.2.9.1 Γενικά 55
2.2.9.2 Αρχιτεκτονική 58
2.2.10 Πρόβλημα Vanishing Gradient 66

Κεφάλαιο **3** **Σχεδίαση και Υλοποίηση**……...…....**68**
3.1 Δεδομένα Εισόδου 69
3.1.1 Δεδομένα Εκπαίδευσης και Επαλήθευσης 69
3.1.2 Αναπαράσταση δεδομένων στα υπολειπόμμενα δίκτυα 71
3.1.2.1 Πρόβλημα αναπαράστασης δεδομένων στα CNN και η επίλυση του 72
3.1.2.2 DSSP Κωδικοποίηση 74
3.1.2.3 Αρχεία πολλαπλής στοίχισης (MSA) 74
3.1.2.4 Τεχνική Σημαντικότερων Γειτονικών Αμινοξέων 76
3.2 Βιβλιοθήκη Tensorflow 77
3.2.1 Γενικά 78
3.2.2 Προσαρμογή Βιβλιοθήκης στο PSSP Πρόβλημα 78
3.2.2.1 Βάση MNIST 79
3.2.2.2 Δεδομένα Εισόδου 81
3.2.2.3 Δεδομένα Εξόδου 81
3.2.2.4 Εκπαίδευση Δικτύου 82

Κεφάλαιο **4** **Πειράματα και Αποτελέσματα**…… **84**
4.1 Αποτελέσματα βάσης δεδομένων MNIST 85
4.2 10-fold Cross-validation 87
4.3 Πειράματα βελτιστοποίησης υπερπαραμέτρων του δικτύου 89
4.3.1 Χρήση Διαφορετικού Αριθμού Κρυφών Επιπέδων 89
4.3.2 Χρήση Διαφορετικών Μεγεθών Φίλτρου 93
4.3.3 Χρήση Διαφορετικού Αριθμού Παράλληλων Φίλτρων 96
4.3.4 Χρήση Διαφορετικής Τιμής Γεμίσματος 99
4.3.5 Χρήση Διαφορετικού Stride 102

4.3.6 Χρήση Διαφορετικού ρυθμού εκμάθησης 105
4.3.8 Χρήση max pooling layers σε συνδυασμό το αρχείο δεδομένων ονομασίας ‘fold8’ 111
Κεφάλαιο **5** **Συμπεράσματα και Μελλοντική Εργασία**…….…..……. **115**
5.1 Συμπεράσματα 116
5.2 Μελλοντική Εργασία 118

Α ν α φ ο ρ έ ς ……..……..……... **124**

Β ι β λ ι ο γ ρ α φ ί α …….…….. **129**

Π α ρ ά ρ τ η μ α Α…….……..……..…….…… **Α-1**

Π α ρ ά ρ τ η μ α Β……..…… **Β-1**

Κεφάλαιο 1

Εισαγωγή

1.1 Η σημασία και ο σκοπός της έρευνας 2
1.2 Προηγούμενες σχετικές έρευνες 4

Η σημασία και ο σκοπός της έρευνας

Οι πρωτεΐνες αποτελούν τα πιο πολυδιάστατα και βασικά στοιχεία του κάθε ζωντανού οργανισμού. Καταλαμβάνουν σημαντικό ρόλο στο ανθρώπινο σώμα αφού αυτές είναι οι υπεύθυνες για την σωστή ανάπτυξη και συντήρηση του ανθρώπινου αλλά και οποιουδήποτε άλλου ζωντανού οργανισμού. Συγκεκριμένα αποτελούνται από οργανικές ενώσεις που ονομάζονται αμινοξέα τα οποία συνδέονται μεταξύ τους με μακριές αλυσίδες. Η αλληλεπίδραση των αμινοξέων μεταξύ τους έχει ως αποτέλεσμα την δημιουργία αναδιπλώσεων της πρωτεΐνης σε μια συγκεκριμένη τρισδιάστατή δομή. Η δομή της πρωτεΐνης γενικότερα αποτελείται από 4 επίπεδα, την πρωτοταγή, δευτεροταγή, τριτοταγή και τεταρτοταγή δομή. Η πρωτοταγή δομή είναι μια ακολουθία των αμινοξέων, χιλιάδες μικρότερες μονάδες, οι οποίες ανάλογα με την σειρά που βρίσκονται στην πρωτεΐνη, καθορίζουν την τελική δομή. Η δευτεροταγή δομή μιας πρωτεΐνης αναφέρεται στην αναπαράσταση της ακολουθίας των αμινοξέων σε τοπικές αναδιπλούμενες δομές (α-έλικες, εκτεταμένους β-κλώνους). Η τριτοταγή δομή μιας πρωτεΐνης δημιουργείται συνδυάζοντας τις αναδιπλούμενες αυτές δομές με αποτέλεσμα να δημιουργηθεί το τελικό και λειτουργικό σχήμα της πρωτεΐνης στο χώρο. Τέλος η τεταρτοταγής δομή είναι η ένωση δύο ή περισσότερων αλυσίδων της τριτοταγής δομής.

Σημαντικότερη δομή της πρωτεΐνης λοιπόν είναι η τριτοταγή δομή αφού αυτή προσδιορίζει τις αναδιπλώσεις της, οι οποίες σε κάθε πρωτεΐνη έχουν διαφορετικό σχήμα, με αποτέλεσμα να καθορίζεται και η λειτουργία της. Γνωρίζοντας την πρωτοταγή δομή της πρωτεΐνης μπορούμε να εξάγουμε την δευτεροταγή δομή και συνεπώς την τριτοταγή αφού η τριτοταγής δομή της πρωτεΐνης είναι οι αναδιπλώσεις των α-ελίκων και β-κλώνων. Γνωρίζοντας λοιπόν την τριτοταγή δομή της πρωτεΐνης μπορούμε να βοηθήσουμε αρκετά το τομέα της φαρμακευτικής και της ιατρικής στο να αναπτυχθούν διάφορες μέθοδοί για την αντιμετώπιση και θεραπεία πολλών ασθενειών που αφορούν την λειτουργία των πρωτεϊνών. Συνεπώς η μελέτη των υφιστάμενων πρωτεϊνών έχει τεράστιο ρόλο αφού εμπλέκεται όχι μόνο στην επιστήμη της βιολογίας και της βιοπληροφορικής, αλλά και της ιατροφαρμακευτικής.

Το πρόβλημα που αντιμετωπίζεται τα τελευταία χρόνια είναι η διαδικασία εύρεσης της τριτοταγούς δομής της πρωτεΐνης. Δυστυχώς, οι μεθοδολογίες εξαγωγής της

τρισδιάστατης δομής των πρωτεϊνών που έχουν αναπτυχθεί μέχρι σήμερα είναι αρκετά πολύπλοκες και χρονοβόρες διαδικασίες και έχουν πολύ υψηλό κόστος. Είναι ένα αρκετά σοβαρό πρόβλημα γιατί από μόνη της η πρωτοταγή δομή δεν δίνει αρκετή πληροφορία, παρόλο που μπορεί εύκολα να εξαχθεί από κάθε πρωτεΐνη, ενώ η τριτοταγή δομή περιέχει την βασική λειτουργία της πρωτεΐνης. Αυτό έχει ως αποτέλεσμα την εμφάνιση ενός αριθμού υπολογιστικών τεχνικών και αλγορίθμων που προσπαθούν να επιλύσουν το πρόβλημα αυτό. Συγκεκριμένα προσπαθούν να προβλέψουν την δευτεροταγή δομή βάση της πρωτοταγούς, ούτως ώστε από την δευτεροταγή δομή να μπορέσουν να εξάγουν την τριτοταγή δομή πιο γρήγορα και φθηνότερα.

Μία από αυτές τις τεχνικές που χρησιμοποιήθηκαν σε αυτό το πρόβλημα είναι η χρήση αλγορίθμων Μηχανικής Μάθησης (Machine Learning). Οι αλγόριθμοι αυτοί σχετίζονται με υπολογιστικές στατιστικές και τεχνικές μαθηματικής βελτιστοποίησης οι οποίες εφαρμόζονται σε συστήματα που μπορούν να μαθαίνουν και να εκπαιδεύονται από τα δεδομένα εισόδου που θα τους παρουσιάζονται, με στόχο να μπορούν να προβλέπουν από μόνα τους νέα δεδομένα. Συγκεκριμένα σε αυτή την έρευνα θα επικεντρωθούμε στα τεχνητά νευρικά δίκτυα (ANN), και πιο συγκεκριμένα θα δοκιμαστούν τα βαθιά υπολειπόμενα νευρωνικά δίκτυα (Deep Residual Networks-ResNets) για να επιτευχθεί μια τυχόν καλύτερη δυνατή πρόβλεψη της δευτεροταγούς δομής των πρωτεϊνών από την πρωτοταγή τους δομή, έτσι ώστε να μπορούμε να προβλέψουμε την τριτοταγή τους δομή. Επιλέξαμε αυτά τα δίκτυα για δοκιμή διότι δεν έχουν ξαναχρησιμοποιηθεί στο συγκεκριμένο πρόβλημα (Protein Secondary Structure Prediction – PSSP) λόγω του ότι έχουν αναπτυχθεί πολύ πρόσφατα (2015) (He et al., 2016).

Τα βαθιά υπολειπόμενα δίκτυα ανήκουν στην κατηγορία των βαθιών νευρωνικών δικτύων (Deep Neural Networks-DNN) (He et al., 2016a). Το βάθος του δικτύου έχει κρίσιμη σημασία στις αρχιτεκτονικές νευρωνικών δικτύων, αλλά τα πιο βαθιά δίκτυα είναι πιο δύσκολο να εκπαιδευτούν (He et al., 2016a). Τα υπολειπόμενα βαθιά δίκτυα μπορούν να αντιμετωπίσουν αυτό το πρόβλημα που έχουν τα βαθιά δίκτυα. Συγκεκριμένα επιλέγοντας αυτά τα δίκτυα αντιμετωπίζεται καλύτερα το πρόβλημα της εξαφανιζόμενης κλίσης (vanishing problem) το οποίο πρόβλημα οφείλεται στην αύξηση

του βάθους του δικτύου. Αυτό το πρόβλημα οφείλεται στο ότι η κλίση είναι εξαιρετικά μικρή, εμποδίζοντας το βάρος να αλλάζει τιμή με αποτέλεσμα να οδηγηθούμε σε υψηλότερο σφάλμα εκπαίδευση (αναλύεται περισσότερο στο υποκεφάλαιο 2.2.10). Η υπολειπόμενη μάθηση όμως επιλύει αυτό το πρόβλημα επιτρέποντας το δίκτυο να είναι βαθύτερο και οδηγώντας το σε βελτιωμένα αποτελέσματα.

Συγκεκριμένα τα ResNets είναι τεχνητά νευρωνικά δίκτυα που χρησιμοποιούν συνδέσεις παράκαμψης για να μεταβούν σε επίπεδα που έχουν μεγαλύτερο βάθος βοηθώντας το δίκτυο να εκπαιδευτεί. Για να μπορούν να γινούν αυτές οι παραλήψεις (skip connections) πρέπει τα υπολειπόμενα βαθιά δίκτυα να είναι πιο βαθιά από τα κανονικά. Σε αυτό το ResNet λοιπόν η εκπαίδευση θα γινεί με την μέθοδο της ανάστροφής μετάδοσης σφάλματος δεχόμενη σαν είσοδο μια σειρά από αμινοξέα, συγκεκριμένα 15 αμινοξέα που το μεσαίο θα είναι το κύριο αμινοξύ που μελετούμε και τα υπόλοιπα οι γείτονες του, και σαν έξοδο θα είναι η πρόβλεψη της κατηγορίας της δευτεροταγούς δομής που ανήκει το μεσαίο αμινοξύ. Περισσότερα για την δομή των δεδομένων εισόδου και εξόδου θα εξηγηθούν στο κεφάλαιο 3. Χρησιμοποιώντας αυτή την αρχιτεκτονική λοιπόν, η οποία εξηγείται περεταίρω στο υποκεφάλαιο 2.2.9, θα εισέρχεται στο δίκτυο περισσότερη πληροφορία για ένα αμινοξύ με αποτέλεσμα το δίκτυο να εκπαιδεύεται καλύτερα παράγοντας και καλύτερα αποτελέσματα.

1.1 Προηγούμενες σχετικές έρευνες

Οι έρευνες για την πρόβλεψη της δευτεροταγούς δομής της πρωτεΐνης ξεκίνησαν πριν από πολλά χρόνια λόγω της τεράστιας σημασίας και ανάγκης επίλυσης του συγκεκριμένου προβλήματος. Με την πάροδο του χρόνου έχουν δημιουργηθεί και τροποποιηθεί αρκετοί αλγόριθμοι μάθησης για να προβλέψουν την συγκεκριμένη δομή χρησιμοποιώντας ένα συγκεκριμένο τρόπο αξιολόγησης. Αυτός ο τρόπος σχετίζεται με τον υπολογισμό του ποσοστού επιτυχίας Q3 (Εξίσωση 1.1) όπου υπολογίζεται μια σχέση μεταξύ του αριθμού των αμινοξικών καταλοίπων που προβλέπονται σωστά με τον ολικό αριθμό των αμινοξικών καταλοίπων που προβλέπονται σε μια πρωτεΐνη.

$$Q3 = \frac{\text{number of residues correctly predicted}}{\text{number of all residues}} * 100\%$$

Εξίσωση 1.1: Εξίσωση εύρεσης ποσοστού επιτυχίας Q3 για τις 3 κλάσεις/κατηγορίες(H-έλικες, E-κλώνοι, C-coil) , όπου number of all residues correctly predicted είναι ο αριθμός αμινοξικών καταλοίπων που προβλέπονται σωστά και number of all residues ο ολικός αριθμός αμινοξικών καταλοίπων.

Όπως προαναφέρθηκε δημιουργηθήκαν αρκετοί αλγόριθμοι που χρησιμοποιήθηκα για την επίλυση του PSSP προβλήματος από σημαντικούς ερευνητές. Οι Qian and Sejnowski το 1988 έχουν χρησιμοποιήσει ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο με το μέγεθος του παραθύρου εισόδου 13 (13 αμινοξέα ορθογώνιάς κωδικοποίησης) και ένα μόνο κρυφό επίπεδο. Το δίκτυο αυτό πρόβλεπε την κατηγορία της δευτεροταγούς δομής του κεντρικού αμινοξέως που βρισκόταν στο παράθυρο εισόδου. Με άλλα λόγια η έξοδος του δικτύου θα ήταν είτε H: α - έλικας, είτε E: β - πτυχωτή επιφάνεια, είτε C: αμινοξέα που δεν ανήκουν σε μια από τις άλλες κατηγορίες των πρωτεϊνών. Σημαντικό να αναφερθεί ότι δεν υλοποιήθηκε μόνο αυτό το δίκτυο αλλά χρησιμοποιήσαν ακόμη ένα για να βελτιώσουν τα δεδομένα εξόδου του πρώτου δίκτυού που χρησιμοποιήσαν. Το ποσοστό επιτυχίας Q3 της συγκεκριμένης έρευνάς κατάφερε να φτάσει μέχρι και το 63.3% παρόλο που αντιμετώπιζε προβλήματα υπερεκπαίδευσης.

Ακολούθως οι Rost και Sander το 1993 δημιουργήσαν το μοντέλο PHD(Profile network from HeiDelberg) το οποίο στην ουσία η αρχιτεκτονική του δικτύου ήταν η ίδια με το πλήρως συνδεδεμένο νευρωνικό δίκτυο που έχει αναφερθεί πιο πάνω με την διαφορά ότι οι ερευνητές αυτοί προσπάθησαν να βρούνε τεχνικές στο να αντιμετωπίσουν το πρόβλημα της υπερεκπαίδευσης. Κάποιες τεχνικές που χρησιμοποιήσαν ήταν το γρήγορο σταμάτημα, ensemble average και γενικά επεξεργασία δεδομένων εισόδου για εκπαίδευση και επαλήθευσή. Ως αποτέλεσμα , κατάφεραν να μειώσουν το πρόβλημα της υπερεκπαίδευσης αυξάνοντας το ποσοστό επιτυχίας στο 71.40%.

Στη συνέχεια, μια άλλη έρευνα που υπήρξε ήταν η δημιουργία του DSC (Discrimination of protein secondary structure class) από τον King και Sternberg το 1996. Σε αυτή την έρευνα αναπτύχθηκε ένας αλγόριθμος που ομαδοποιεί τα δεδομένα εξόδου του δίκτυού σε διάφορες κατηγορίες χρησιμοποιώντας κάποιους υπολογισμούς από γραμμικές και στατικές μεθόδους ούτως ώστε να προσεγγίσει την ακριβή δευτεροταγής δομής της πρωτεΐνης.

Ένας αλγόριθμος που ξεχώρισε και είχε την καλύτερη απόδοση εκείνο το καιρό ήταν ένα δίκτυο αμφίδρομης ανάδρασης (Bidirectional Recurrent Neural Network-BRRN) από τον Baldi et al. (1999) με ποσοστό επιτυχίας Q3 73.6%. Πιο συγκεκριμένα το δίκτυο αυτό πρόβλεπε το μεσαίο αμινοξύ από ένα κινητό παράθυρο, το οποίο παράθυρο ήταν η είσοδος στο δίκτυο, βάση τα γειτονικά αμινοξέα του. Το δίκτυο δεχόταν περισσότερη πληροφορία η οποία προερχόταν από τα γειτονικά αμινοξέα (αμινοξέα που προηγούνται και έπονται του συγκεκριμένου αμινοξέος που μελετάται) χρησιμοποιώντας αμφίδρομη ανάδραση με αποτέλεσμα να παραχθούν αυτά τα ψηλά αποτελέσματα.

Έγινε χρήση των BRNN και από τους Chen και Chaudhari το 2007 λαμβάνοντας υπόψη και τις μακρινές αλληλεξαρτήσεις μεταξύ των αμινοξικών καταλοίπων τονίζοντας ότι είχαν σημαντικό ρόλο στην αναδίπλωση της πρωτεΐνης. Συγκεκριμένα δημιουργηθήκαν δυο BRNN των οποίων η έξοδος του πρώτου ήταν η είσοδος του δεύτερου. Η μέθοδος αυτή είχε Q3 ποσοστό επιτυχίας 74.83%.

Ο Wang και οι συνεργάτες του (Wang et al.,2016) κατάφεραν να δημιουργήσουν ένα βαθύ συνελικτικό νευρωνικό δίκτυο (Deep Convolutional Neural Network- DCNN) το οποίο κατάφερε να φτάσει μέχρι και το 84% ποσοστό επιτυχίας Q3. Το συγκεκριμένο δίκτυο ήταν μια επέκταση του Deep Learning από Conditional Neural Fields (CNF), η οποία είναι μια συνένωση των Conditional Random Fields (CRF) και των shallow neural networks. Επίσης καταλήξαν ότι τα δίκτυα αυτά είναι χρήσιμα στην πρόβλεψη άλλων ιδιοτήτων της δομής της πρωτεΐνης όπως για παράδειγμα disorder regions, and solvent accessibility.

Τα τελευταία 10 περίπου χρόνια εφευρεθήκαν πολλοί αλγόριθμοι για την μελέτη και επίλυση του προβλήματος της πρόβλεψης της δευτεροταγούς δομής της πρωτεΐνης και από άτομα στο Πανεπιστήμιο Κύπρου. Συγκεκριμένα, σύμφωνα με προηγούμενες διπλωματικές εργασίες μπορούμε να παρατηρήσουμε ότι έχει γίνει μια αρκετά σοβαρή έρευνα σχετικά με το συγκεκριμένο πρόβλημα από τον καθηγητή Χριστοδούλου Χρίστο και από το διδακτορικό φοιτητή Μιχάλη Αγαθοκλέους. Έχουν αναπτυχθεί αρκετοί αλγόριθμοι από πολλούς φοιτητές, με την βοήθεια του καθηγητή Χριστοδούλου και του Αγαθοκλέους, εξάγοντας ικανοποιητικά ποσοστά επιτυχίας.

Ο Αγαθοκλέους το 2009 ανέπτυξε ένα νευρωνικό δίκτυο αμφίδρομης ανάδρασης το οποίο βασιζόταν στην αρχιτεκτονική του δικτύου του Baldi et al. (1999) όπως έχει προαναφερθεί. Η εκπαίδευση είχε γινεί με τον αλγόριθμο ανάστροφης μετάδοσης λάθους με είσοδο μόνο την πρωτοταγή δομή και έξοδο την πρόβλεψη της δευτεροταγούς δομής. Το ποσοστό Q3 έφθασε μέχρι το 64% αλλά ο Αγαθοκλέους είχε τονίσει ότι το δίκτυο μαθαίνει σωστά και πως υπάρχουν ελπίδες στο μέλλον για βελτιώσεις αυτού του ποσοστού.

Συνεπώς, η Χριστοδούλου το 2010 ανάλαβε την συνέχεια της μελέτης που είχε κάνει ο Αγαθοκλέους με το νευρωνικό δίκτυο αμφίδρομης ανάδρασης. Με κάποιες αλλαγές και διαμορφώσεις πέτυχε να αυξήσει το ποσοστό επιτυχίας στο 74%.

Έπειτα το 2016 ο Παυλίδης προσπάθησε να επιλύσει το συγκεκριμένο πρόβλημα με την χρήση συνελικτικών νευρωνικών δικτύων. Λόγω του ότι τα συνελικτικά δίκτυα ανήκουν στην οικογένεια των βαθιών νευρωνικών δικτύων, το δίκτυο που είχε υλοποιήσει αντιμετώπιζε κάποια προβλήματα στην εκπαίδευση. Μετά από προσπάθεια να οπτικοποιήσει τα δεδομένα εισόδου για να αντιμετωπίσει τα προβλήματα που είχε κατά την εκπαίδευση το καλύτερο ποσοστό επιτυχίας Q3 που πήρε ήταν το 47.89%.

Ο Δημητρίου (2018), στην δική του διπλωματική εργασία, προσπάθησε να επιλύσει το συγκεκριμένο πρόβλημα με την χρήση clockwork νευρωνικών δίκτυών (CW-RNN) (Koutnik et al., 2014). Αυτά τα δίκτυα με την χρήση διάφορων clock speeds αντιμετώπισαν ένα σοβαρό πρόβλημα, το πρόβλημα εξαφανιζόμενης κλίσης (vanishing gradient) το οποίο προκαλούσε δυσκολίες στην εκπαίδευση του δικτύου. Τα CWRRNs

έδωσαν ποσοστό επιτυχίας Q3 75.74% και SOV [1] 72. Αλλά υπήρξαν καλύτερα αποτελέσματα αφού ο Δημητρίου έπειτα χρησιμοποίησε και αυτός φίλτρα, συγκεκριμένα τα SVMs και εξωτερικοί εμπειρικοί κανόνες, ως αποτέλεσμα το ποσοστό Q3 να φτάσει μέχρι το 76.44%.

Τέλος, η έρευνα του Διονυσίου (2018), είχε σκοπό να εξετάσει το πώς τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNNs), μπορούν να δώσουν λύση στο PSSP πρόβλημα σε συνδυασμό με τα φίλτρα Gabor και Support Vector Machines (SVMs). Τα συνελικτικά νευρωνικά δίκτυα ανήκουν στην κατηγορία των βαθιών νευρωνικών δικτύων και συνεπώς τα δίκτυα αυτά αναλύοντας τα δεδομένα εισόδου και εφαρμόζοντας πολλαπλά φίλτρα εξαγωγής δεδομένων σε κάθε επίπεδο, είναι σε θέση να εντοπίσουν πολύπλοκα χαρακτηριστικά στα δεδομένα εισόδου. Οπτικοποίησε τα δεδομένα εκπαίδευσης και επαλήθευσης και έπειτα χρησιμοποίησε φίλτρα Gabor δημιουργώντας μια νέα εμπλουτισμένης μορφής δεδομένα εισόδου που χρησιμοποιήθηκε για να παράξει υψηλότερα ποσοστά επιτυχίας στην πρόβλεψη. Δυστυχώς ο Διονυσίου κατέληξε στο ότι τα Gabor φίλτρα δεν ήταν ιδιαίτερα αποτελεσματικά και χρήσιμα στο πρόβλημα της πρόβλεψης της δευτεροταγούς δομής των πρωτεϊνών διότι δεν βελτίωναν τα ποσοστά επιτυχίας. Κατά συνέπεια είχε χρησιμοποιήσει ένα άλλο αλγόριθμο μάθησης για να φιλτράρει τα αποτελέσματα του PSSP πρόβλημα, ο οποίος έδωσε καλύτερα αποτελέσματα, το Support Vector Machine (SVM). Η μέθοδος αυτή κατάφερε να φτάσει στο 80.40% ποσοστό επιτυχίας Q3 και 73.6 SOV.

[1] SOV: Segment Overlap: μέθοδος βαθμολόγησης της προβλεπόμενης ακολουθίας της δευτεροταγούς δομής.

Κεφάλαιο 2

Προ απαιτούμενη γνώση - Υπόβαθρο

2.1 Βιολογικό υπόβαθρο 10
2.1.1 Πρωτεΐνες και αμινοξέα 10
2.1.2 Δομή της πρωτεΐνης 13
2.1.2.1 Πρωτοταγής δομή 13
2.1.2.2 Δευτεροταγής δομή 14
2.1.2.3 Τριτοταγής δομή 15
2.1.2.4 Τεταρτοταγής δομή 17
2.1.3 Βιολογικός ρόλος της πρωτεΐνης 18
2.2 Τεχνητά Νευρωνικά Δίκτυα 19
2.2.1 Γενικά 19
2.2.2 Βιολογική έμπνευση 20
2.2.3 Αρχιτεκτονική και Λειτουργία 22
2.2.4 Εκπαίδευση νευρωνικών δικτύων 26
2.2.5 Μοντέλο Νευρώνα McCulloh και Pitts 28
2.2.6 Συναρτήσεις ενεργοποίησης 30
2.2.7 Πολυστρωματικά Δίκτυα Perceptron 33
2.2.7.1 Αλγόριθμος μάθησης Perceptron 35
2.2.7.2 Μέθοδος κατάβασης κλίσης 36
2.2.7.3 Αλγόριθμος ανάστροφης μετάδοσης σφάλματος 38
2.2.8 Συνελικτικά Νευρωνικά Δίκτυα 40
2.2.8.1 Γενικά 40
2.2.8.2 Αρχιτεκτονική 43
2.2.9 Υπολειπόμενα Νευρωνικά Δίκτυα 55
2.2.9.1 Γενικά 55
2.2.9.2 Αρχιτεκτονική 58
2.2.10 Πρόβλημα Vanishing Gradient 66

2.1 Βιολογικό υπόβαθρο

2.1.1 Πρωτεΐνες και αμινοξέα

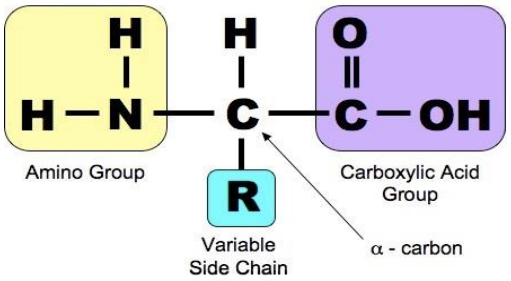
Γενικότερα στην επιστήμη της βιολογίας ένα κύτταρο αποτελείται από μια συστηματικά οργανωμένη ομάδα σωματιδίων. Ένα από αυτά τα μικρά σωματίδια που θεωρείται ως σημαντική δομική μονάδα του κυττάρου είναι το ριβόσωμα το οποίο αποτελείται από πρωτεΐνες και RNA και βρίσκεται ελεύθερο μέσα στο κυτταρόπλασμα ή στο αδρό ενδοπλασματικό δίκτυο. Ο κύριος του ρόλος είναι η σύνθεση πρωτεϊνών χρησιμοποιώντας γενετικές πληροφορίες που βρίσκονται στο αγγελιοφόρο RNA. Οι πρωτεΐνες αυτές αποτελούν το πιο διαδεδομένο μακρομόριο [2] καθώς και το πιο πολυδιάστατο στη μορφή και στη λειτουργία του. Ακόμα και σε ένα απλό κύτταρο, όπως αυτό των βακτηρίων, υπάρχουν εκατοντάδες διαφορετικές πρωτεΐνες, κάθε μία από τις οποίες έχει έναν ιδιαίτερο ρόλο στη ζωή του κυττάρου (είτε δομικό συστατικό, είτε εξυπηρετεί κάποια συγκεκριμένη λειτουργία του). Πιο συγκεκριμένα οι πρωτεΐνες αποτελούνται από μια ή περισσότερες πεπτιδικές αλυσίδες και περιέχουν άνθρακά, οξυγόνο , άζωτο και στις περισσότερες υπάρχει και θείο. Οι πεπτικές αυτές αλυσίδες, πολυπεπτίδιά, αποτελούνται από αλληλουχίες αμινοξέων με αποτέλεσμα να σχηματίζεται η πρώτη μορφή της πρωτεΐνης, η λεγόμενη πρωτοταγή δομή.

Ελληνική ονομασία	Διεθνής σύντμηση	Ελληνική ονομασία	Διεθνής σύντμηση
Αλανίνη	Ala	Ιστιδίνη	His
Αργινίνη	Arg	Κυστεΐνη	Cys
Ασπαραγίνη	Asn	Λευκίνη*	Leu
Ασπαραγινικό οξύ	Asp	Λυσίνη*	Lys
Βαλίνη*	Val	Μεθειονίνη*	Met
Γλουταμινικό οξύ	Glu	Προλίνη	Pro
Γλουταμίνη	Gln	Σερίνη	Ser
Γλυκίνη	Gly	Τρυπτοφάνη*	Trp
Θρεονίνη*	Thr	Τυροσίνη	Tyr
Ισολευκίνη*	Ile	Φαινυλαλανίνη*	Phe

Πίνακας 2.1: Τα 20 αμινοξέα που συνθέτουν τις πρωτεΐνες των ζωντανών οργανισμών. Τα ονόματα με αστερίσκο (*) είναι τα 8 βασικά αμινοξέα. (Wikipedia, 2019)

[2] Μακρομόρια: σύνθετες οργανικές ενώσεις μεγάλου μοριακού βάρους όπως πρωτεΐνες ,νουκλεϊκα οξέα , πολυσακχαρίτες και λιπίδια.

Έχουν ανιχνευθεί πάνω από 170 διαφορετικά αμινοξέα από τα οποία όμως 20 μόνο αποτελούν συστατικό των πρωτεϊνών (Πίνακας 2.1). Τα αμινοξέα είναι τα βασικά δομικά στοιχεία των πρωτεϊνών που αυτά καθορίζουν τα χαρακτηριστικά των πρωτεϊνών και αναφέρονται με συντομογραφία τριών γραμμάτων (τρία πρώτα γράμματα του ονόματος τους) όπως φαίνεται στο πίνακα πιο πάνω. Τα αμινοξέα και πιο συγκεκριμένα τα πρωτεϊνικά αμινοξέα είναι χημικές ενώσεις που περιέχουν μια τουλάχιστον καρβονική ομάδα και μια τουλάχιστον αμινομάδα (Σχήμα 2.1). Τα πρωτεϊνικά αμινοξέα εστιάζονται κυρίως στα α-αμινοξέα τα οποία αποτελούνται από 3 σταθερά μέρη και ένα μεταβλητό μέρος (πλευρική ομάδα). Συγκεκριμένα η δομή των α-αμινοξέων είναι ένα άτομο άνθρακα το οποίο συνδέεται ομοιοπολικά με μία αμινομάδα στα αριστερά του (NH2), μια καρβοξυλομάδα στα δεξιά του (COOH), μια χημική δομή που περιέχει πολλά διαφορετικά άτομα η οποία ονομάζεται πλευρική δομή (R) και ένα άτομο υδρογόνου· δεδομένου ότι το C στο κέντρο είναι ένα άτομο α-άνθρακα, το H είναι ένα άτομο υδρογόνου, το O είναι ένα άτομο οξυγόνου και το N είναι ένα άτομο αζώτου. Η πλευρική ομάδα R είναι διαφορετική σε κάθε αμινοξύ με αποτέλεσμα αυτή να είναι υπεύθυνη να διαφοροποιεί τα αμινοξέα ως προς τα χαρακτηριστικά τους και ως συνέπειά να ομαδοποιούνται σε τέσσερις κατηγορίες. Η πρώτη κατηγορία είναι τα όξινα αμινοξέα των οποίων η πλευρική αλυσίδα είναι φορτισμένη αρνητικά, η δεύτερη είναι τα βασικά αμινοξέα των οποίων η πλευρική αλυσίδα είναι φορτισμένη θετικά, η τρίτη κατηγορία είναι τα αμινοξέα που δεν έχουν φορτίο αλλά η πλευρική αλυσίδα έχει δυο περιοχές αντίθετά φορτισμένες και τέλος η τέταρτη κατηγορία είναι τα αμινοξέα που δεν έχουν καθόλου φορτίο.



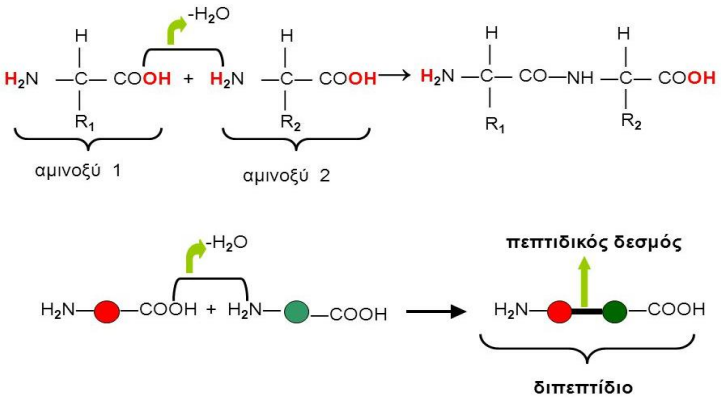
Σχήμα 2.1: Δομή αμινοξέως

Πάρθηκε από: http://www.vce.bioninja.com.au/aos-1-molecules-oflife/biomolecules/proteins.html

Σημαντικό να αναφερθεί ότι ανάλογα με τις ιδιότητες και τα χαρακτηριστικά που έχει η κάθε πρωτεΐνη, σχηματίζεται μια διαφορετική δομή στην κάθε πρωτεΐνη στο

τρισδιάστατο χώρο με αποτέλεσμα να υπάρχουν πολλές πρωτεΐνες με διαφορετικό σχήμα και διαφορετική λειτουργία.

Αν τοποθετήσουμε σε μια σειρά τα αμινοξέα δημιουργείται η αμινοξική ακολουθία σχηματίζοντας πεπτίδια (πρωτοταγής δομή). Αναλυτικότερα για να δημιουργηθεί αυτή η ακολουθία πολυπεπτιδίων πρέπει να γινεί η ένωση των αμινοξέων. Η διαδικασία αυτή ονομάζεται συμπύκνωση (Σχήμα 2.2). Για να συνδεθούν λοιπόν δυο αμινοξέα, αντιδρά η καρβοξυλομάδα (φορτισμένη αρνητικά) του ενός με την αμινομάδα (φορτισμένη θετικά) του επομένου με απόσπαση ενός μόριού νερού (H_2O). Ο δεσμός που σχηματίζεται λέγεται πεπτιδικός με αποτέλεσμα να παράγεται ένα διπεπτίδιο και τα αμινοξέα να ονομάζονται αμινοξικά κατάλοιπα. Όταν συνδεθούν περισσότερα από 50 αμινοξέα σχηματίζεται ένα πολυπεπτίδιο ή πολυπεπτιδική αλυσίδα δημιουργώντας την πρώτη δομή της πρωτεΐνης. Μετά την σύνθεση του πολυπεπτιδίου, δυστυχώς δεν είναι ικανό να εκδηλώσει το βιολογικό του ρόλο. Η ικανότητα αυτή αποκτάται, όταν η πολυπεπτιδική αλυσίδα πάρει την τελική της διαμόρφωση στο χώρο που αυτή η διαμόρφωση ονομάζεται τριτοταγής δομή της πρωτεΐνης. Οι πολυπεπτιδικές αλυσίδες έχουν την ιδιότητα να αναδιπλώνονται στο χώρο. Η αναδίπλωση της κάθε πολυπεπτιδικής αλυσίδας έχει σημαντικό ρόλο αφού και αυτή είναι ένας βασικός παράγοντας για τον καθορισμό την μοναδικής τρισδιάστατης μορφής την πρωτεΐνης και γενικότερα το τελικό σχήμα που θα έχει. Γενικά μπορούμε να πούμε ότι υπάρχουν αρκετοί παράγοντες που βάση αυτών δημιουργείται μια συγκεκριμένη πρωτεΐνη, για παράδειγμα η σειρά των αμινοξέων που θα τοποθετηθούν ή ο αριθμός των αμινοξέων σε μια πρωτεΐνη.

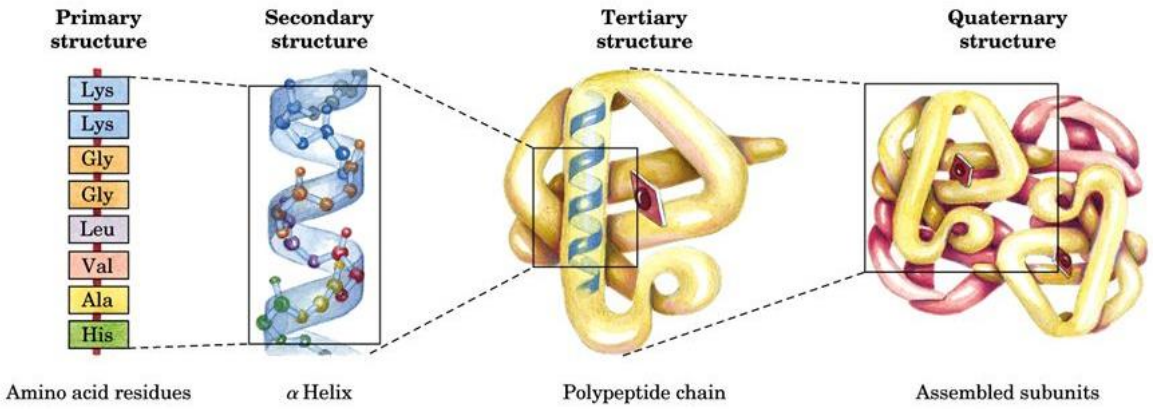


Σχήμα 2.2: Διαδικασία συμπύκνωσης / Πεπτικός δεσμός

Πάρθηκε από: https://slideplayer.gr/slide/2556368/

2.1.2 Δομή της πρωτεΐνης

Το πιο εντυπωσιακό ίσως χαρακτηριστικό της ζωής είναι το γεγονός ότι αυτή είναι οργανωμένη σε επίπεδα αυξανόμενης πολυπλοκότητας. Τα άτομα συνιστούν μόρια, τα μόρια, με τη σειρά τους, κυτταρικά οργανίδια, τα τελευταία σχηματίζουν κύτταρα κ. ο. κ. Εσωτερική οργάνωση συναντούμε και σε κάθε επιμέρους επίπεδο και φυσικά στο πιο στοιχειώδες από αυτά, δηλαδή το μοριακό επίπεδο. Οι χημικές ενώσεις οι οποίες συνθέτουν τους οργανισμούς μπορούν, ανάλογα με το μοριακό βάρος τους, να τοποθετηθούν σε μια ιεραρχική κλίμακα, στην οποία κάθε σκαλί προκύπτει από το προηγούμενο μέσω των αντιδράσεων του μεταβολισμού. Έτσι λοιπόν, διαχωρίζεται και η δομή της πρωτεΐνης σε επίπεδα οργάνωσης, με αποτέλεσμα μια καλύτερη παρακολούθηση της δομής της πρωτεΐνης στις διάφορες φάσεις του σχηματισμού της. Συγκεκριμένα αποτελείται από τέσσερα ιεραρχικά επίπεδα τα οποία αντιστοιχούν στην πρωτοταγή, δευτεροταγή, τριτοταγή και τεταρτοταγή δομή (Σχήμα 2.3).



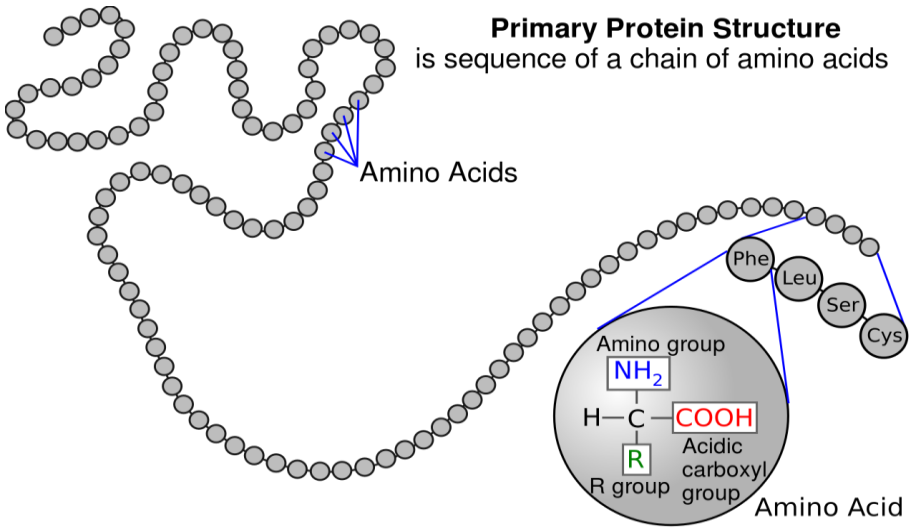
Σχήμα 2.3: Επίπεδα οργάνωσης της πρωτεΐνης

Πάρθηκε από: https://slideplayer.com/slide/7398871/

2.1.2.1 Πρωτοταγής δομή

Το πρώτο επίπεδο είναι η πρωτοταγής δομή η οποία αποτελείται από την αλληλουχία των αμινοξέων στην πολυπεπτιδική αλυσίδα με καθοριστικούς παράγοντες τα νουκλεϊκα οξέα, τα οποία φέρονται να ελέγχουν όλες τις λειτουργίες αλλά και τα

κληρονομικά γνωρίσματα των οργανισμών (Σχήμα 2.4). Αναλυτικότερα, η πρωτοταγής δομή των πρωτεϊνών προκύπτει από την ένωση των αμινοξέων με πεπτιδικούς ομοιοπολικούς δεσμούς οι οποίοι καθορίζονται από ένα γονίδιο και κωδικοποιούνται κατά τον γενετικό κώδικα DNA. Μία αλλαγή στην αλληλουχία DNA του γονιδίου μπορεί να οδηγήσει σε μια αλλαγή στην αλληλουχία αμινοξέων της πρωτεΐνης. Ακόμα και η αλλαγή μόνο ενός αμινοξέος σε μια αλληλουχία πρωτεΐνης μπορεί να επηρεάσει τη συνολική δομή και λειτουργία της πρωτεΐνης.



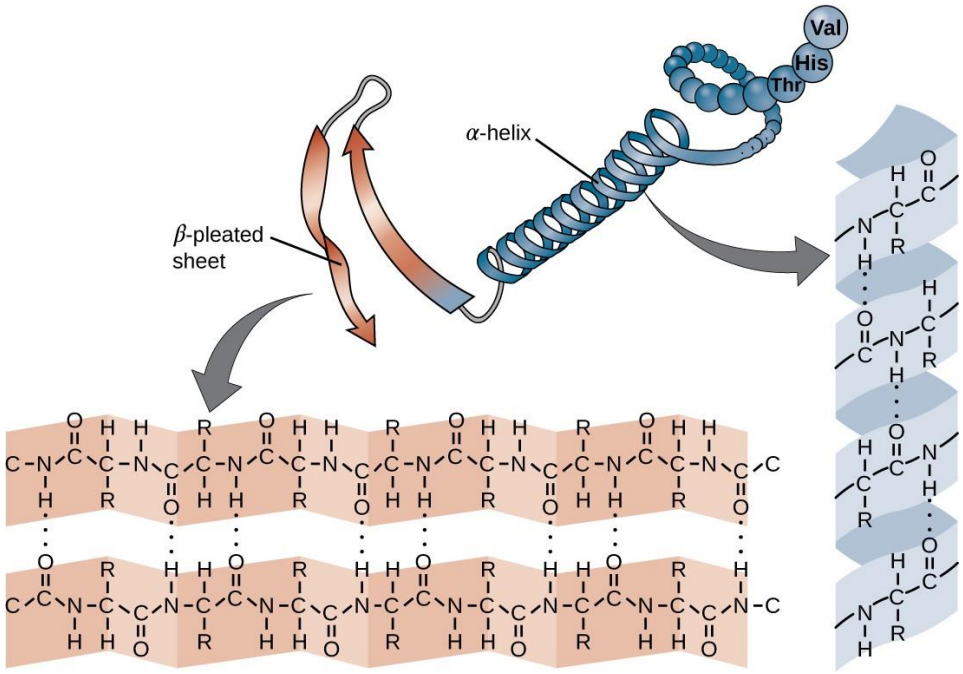
Σχήμα 2.4: Πρωτοταγής δομή πρωτεΐνης

Πάρθηκε από: https://en.wikibooks.org/wiki/Structural_Biochemistry/Proteins

2.1.2.2 Δευτεροταγής δομή

Η αλυσίδα αμινοξέων που ορίζει την πρωτοταγή δομή της πρωτεΐνης δεν είναι άκαμπτη, αλλά είναι εύκαμπτη εξαιτίας της φύσης των δεσμών που συγκρατούν τα αμινοξέα μαζί. Όταν η αλυσίδα είναι επαρκώς μακρά, μπορεί να δημιουργηθεί δεσμός υδρογόνου μεταξύ χαρακτηριστικών ομάδων αμίνης και καρβονυλίου εντός του σκελετού πεπτιδίων (εξαιρουμένης της πλευρικής ομάδας R), με αποτέλεσμα την τοπική αναδίπλωση της πολυπεπτιδικής αλυσίδας σε έλικες και φύλλα. Αυτά τα σχήματα αποτελούν την δευτεροταγή δομή της πρωτεΐνης. Η δευτεροταγής δομή περιορίζει την ελευθερία κίνησης των υδρογόνων της ένωσης και έτσι η πεπτιδική αλυσίδα λαμβάνει μια σταθερή διάταξη στον χώρο. Ο πλέον διαδεδομένος τύπος

τέτοιας μορφής πολυπεπτιδίου είναι η λεγόμενη "α-έλικα", δεξιόστροφη, όπου οι σπείρες διατηρούνται στη θέση τους με δεσμούς υδρογόνου μεταξύ των καρβοξυλομάδων και των αμινομάδων των αμινοξέων. Μια άλλη δευτεροταγής δομή είναι η λεγόμενη "β-πτυχωτή επιφάνεια" όπου στη περίπτωση αυτή διασταυρώνονται παράλληλες αλυσίδες πολυπεπτιδίων που ενώνονται στις διασταυρώσεις με δεσμούς υδρογόνου σχηματίζοντας έτσι μια εξαιρετικά σφιχτή δομή, όπως στο μετάξι (Σχήμα 2.5). Παραδείγματα πρωτεϊνών του οργανισμού με α-έλικα έχουμε τις πρωτεΐνες των μαλλιών του κεφαλιού, της μυοσίνης, του ινώδους, της αιμοσφαιρίνης. Ορισμένα αμινοξέα που χρησιμεύουν σαν πυρήνες για τον σχηματισμό της β-πτυχωτή επιφάνειας (πριονωειδή δομή) είναι η βαλίνη, η μεθειονίνη και η ισολευκίνη.



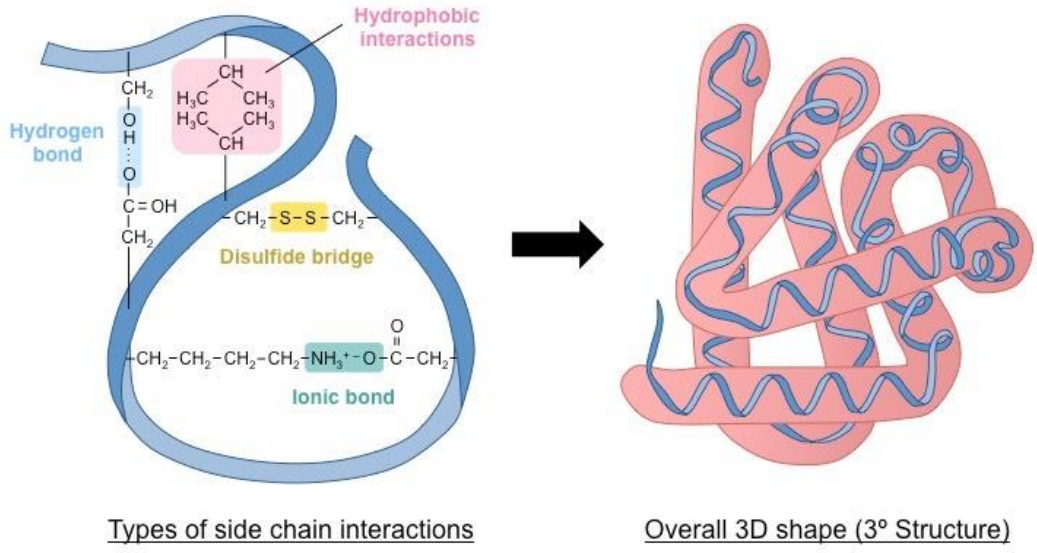
Σχήμα 2.5: Δευτεροταγής δομή πρωτεΐνης. Αριστερά: Παράδειγμα αναδίπλωσης β–πτυχωτής επιφάνειας. Δεξιά: Παράδειγμα αναδίπλωσης α–έλικας

Πάρθηκε από: https://courses.lumenlearning.com/microbiology/chapter/proteins/

2.1.2.3 Τριτοταγής δομή

Το επόμενο επίπεδο οργάνωσης πρωτεϊνών είναι η τριτοταγής δομή, η οποία είναι το μεγάλης κλίμακας τρισδιάστατο σχήμα μιας απλής πολυπεπτιδικής αλυσίδας. Η

τριτοταγής δομή προσδιορίζεται από αλληλεπιδράσεις μεταξύ υπολειμμάτων αμινοξέων που είναι πολύ μακριά στην αλυσίδα. Μία ποικιλία αλληλεπιδράσεων δημιουργεί την τριτοταγή δομή της πρωτεΐνης, όπως δισουλφιδικές γέφυρες, οι οποίες είναι δεσμοί μεταξύ των λειτουργικών ομάδων σουλφυδρυλίου (-SH) σε πλευρικές ομάδες αμινοξέων, δεσμούς υδρογόνου, ιοντικοί δεσμοί και υδρόφοβες αλληλεπιδράσεις μεταξύ μη πολικών πλευρικών αλυσίδων. Όλες αυτές οι αλληλεπιδράσεις, αδύναμες και ισχυρές, συνδυάζονται για να προσδιοριστεί το τελικό τρισδιάστατο σχήμα της πρωτεΐνης και συνεπώς η λειτουργία της (Σχήμα 2.6). Γενικότερα, με τον όρο τριτοταγής δομή, εννοούμε το τελικό και λειτουργικό σχήμα που αποκτά το πολυπεπτίδιο, οπότε πλέον ονομάζεται πρωτεΐνη. Συνεπώς αυτή η δομή είναι αρκετά σημαντική επειδή προσδιορίζει την λειτουργία της πρωτεΐνης. Αν η πρωτεΐνη αποτελείται από μία μόνο πολυπεπτιδική αλυσίδα, όπως η ριβονουκλεάση, το τελικό στάδιο της διαμόρφωσής της είναι η τριτοταγής δομή. Αν όμως αποτελείται από περισσότερες πολυπεπτιδικές αλυσίδες, το τελικό στάδιό της είναι η τεταρτοταγής δομή.

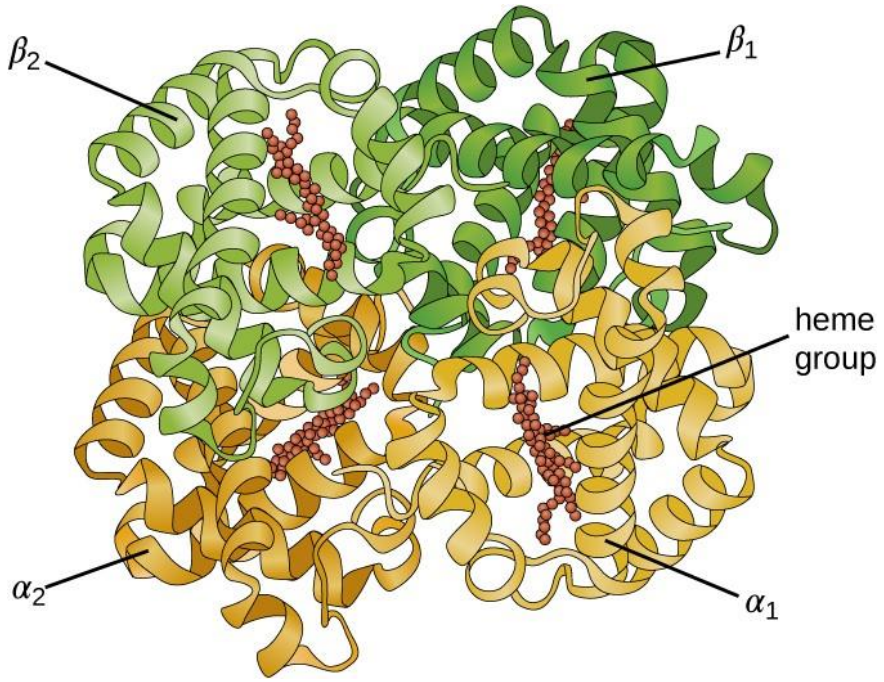


Σχήμα 2.6: Τριτοταγής δομή πρωτεΐνης. Αριστερά: Τύποι αλληλεπιδράσεων πλευρικής αλυσίδας. Δεξιά: Τρισδιάστατη μορφή της πρωτεΐνης

Πάρθηκε από: http://ib.bioninja.com.au/higher-level/topic-7-nucleic-acids/73-translation/protein-structure.html

2.1.2.4 Τεταρτοταγής δομή

Ενώ όλες οι πρωτεΐνες περιέχουν πρωτοταγείς, δευτεροταγείς και τριτοταγείς δομές, οι τεταρτοταγείς δομές προορίζονται για πρωτεΐνες που αποτελούνται από δύο ή περισσότερες πολυπεπτιδικές αλυσίδες. Συγκεκριμένα, ορισμένες πρωτεΐνες είναι συγκροτήματα διαφόρων χωριστών πολυπεπτιδίων, επίσης γνωστών ως πρωτεϊνικές υπομονάδες. Αυτές οι πρωτεΐνες λειτουργούν επαρκώς μόνο όταν υπάρχουν όλες οι υπομονάδες και διαμορφώνονται κατάλληλα. Οι αλληλεπιδράσεις που συγκρατούν αυτές τις υπομονάδες μαζί αποτελούν την τεταρτοταγή δομή της πρωτεΐνης. Η συνολική τεταρτοταγής δομή σταθεροποιείται από σχετικά αδύναμες αλληλεπιδράσεις. Η αιμοσφαιρίνη είναι ένα χαρακτηριστικό παράδειγμα τεταρτογενούς δομής που αποτελείται από δύο υπομονάδες άλφα και δύο βήτα υπομονάδες (Σχήμα 2.7).



Σχήμα 2.7: Τεταρτοταγής δομή πρωτεΐνης του μόριού αιμοσφαιρίνης (έχει δύο α και δύο β πολυπεπτίδια μαζί με τέσσερις ομάδες heme).

Πάρθηκε από: https://courses.lumenlearning.com/microbiology/chapter/proteins/

2.1.3 Βιολογικός ρόλος της πρωτεΐνης

Σύμφωνα με τους μετριοπαθέστερους υπολογισμούς, στο ανθρώπινο σώμα υπάρχουν περισσότερες από 30.000 διαφορετικές πρωτεΐνες. Καθεμιά από αυτές, εμφανίζει έναν ιδιαίτερο βιολογικό ρόλο. Γενικά, μπορούμε να πούμε ότι οι πρωτεΐνες είναι απαραίτητες για όλους τους ζωντανούς οργανισμούς και συμμετέχουν σε κάθε διαδικασία μέσα στα κύτταρα. Οι πρωτεΐνες, με κριτήριο τη λειτουργία τους, διακρίνονται σε δύο ευρύτερες κατηγορίες. Τις δομικές, που αποτελούν δομικά συστατικά των κυττάρων και κατ' επέκταση των οργανισμών, και τις λειτουργικές, που συμβάλλουν στις διάφορες λειτουργίες του οργανισμού.

Πιο συγκεκριμένα οι δομικές πρωτεΐνες έχουν σαν κύριο ρόλο την στήριξη της υφής, της δομής και της σταθερότητας των διαφόρων ιστών και οργάνων των οργανισμών για παράδειγμα κολλαγόνο, κερατίνες. Οι λειτουργικές πρωτεΐνες συμμετέχουν στις διάφορες λειτουργίες των οργανισμών με το να αναγνωρίζουν και να δεσμεύουν εκλεκτικά διάφορα μόρια ή ιόντα, τα οποία στην προκειμένη περίπτωση αποτελούν υποκατάστατες ή προσθήματα και καλούνται ligands. Ανάλογα με το λειτουργικό αποτέλεσμα που προκύπτει από την δέσμευση ή / και αποδέσμευση των ligands οι λειτουργικές πρωτεΐνες διακρίνονται σε διάφορες ομάδες. Μερικές από αυτές απαρτίζονται στο πιο κάτω πίνακα (Πίνακας 2.2).

ΕΙΔΟΣ ΠΡΩΤΕΪΝΩΝ	ΠΑΡΑΔΕΙΓΜΑ	ΛΕΙΤΟΥΡΓΙΑ
Α.ΔΟΜΙΚΕΣ	Κολλαγόνο	Συστατικό του συνδετικού ιστού (οστά, χόνδροι, τένοντες)
	Ελαστίνη	Συστατικό των συνδέσμων των οστών
Β.ΛΕΙΤΟΥΡΓΙΚΕΣ		
ΜΕΤΑΦΕΡΟΥΣΕΣ	Αιμοσφαιρίνη	Μεταφορά οξυγόνου και διοξειδίου του άνθρακα στο αίμα σπονδυλωτών
	Μυοσφαιρίνη	Μεταφορά οξυγόνου και προσωρινή αποθήκευση στους μυς σπονδυλωτών

ΣΥΣΤΑΛΤΕΣ	Μυοσίνη	Συστατικό των μυϊκών κυττάρων
	Ακτίνη	Συστατικό των μυϊκών κυττάρων
ΑΠΟΘΗΚΕΥΤΙΚΕΣ	Καζεΐνη	Αποθήκη ασβεστίου στο γάλα
	Αλβουμίνη	Πηγή αμινοξέων για το αναπτυσσόμενο έμβρυο (στο ασπράδι των αβγών)
ΟΡΜΟΝΙΚΕΣ	Ινσουλίνη	Ρύθμιση του σακχάρου του αίματος. Εκκρίνεται από το πάγκρεας
	Γλυκαγόνη	Ρύθμιση του σακχάρου του αίματος. Εκκρίνεται από το πάγκρεας
ΕΝΖΥΜΙΚΕΣ	Εξοκινάση	Ένζυμο της γλυκόλυσης
	RNA πολυμεράση	Ένζυμο της μεταγραφής του DNA σε RNA

Πίνακας 2.2: Διάκριση των πρωτεϊνών και λειτουργίες που αυτές επιτελούν. Πάρθηκε από: http://ebooks.edu.gr/modules/ebook/show.php/DSGL-B106/726/4800,21687/

2.2 Τεχνητά νευρωνικά δίκτυα

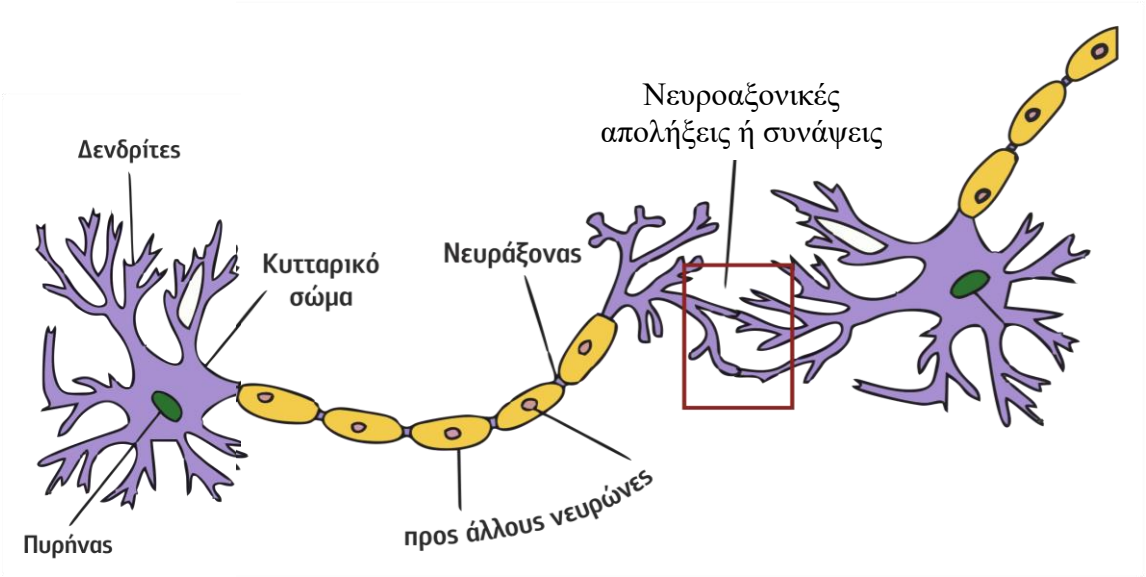
2.2.1 Γενικά

Τα νευρωνικά δίκτυα (Neural Network) αποτελούν μια σχετικά νέα περιοχή στις φυσικές επιστήμες, καθ' όσον έχουν γίνει γνωστά και έχουν αναπτυχθεί μόνο κατά τα τελευταία σαράντα περίπου χρόνια. Εν τούτοις, η περιοχή αυτή έχει δει μια μεγάλη άνθηση, κρίνοντας από την μεγάλη ανάπτυξη που έχει παρατηρηθεί, από τον αριθμό των επιστημόνων που ασχολούνται με αυτά τα θέματα, και βέβαια από τα πολύ σημαντικά επιτεύγματα που έχουν συμβάλλει στο να γίνουν γνωστά σε ένα ευρύτερο κύκλο. Αποτελούν επομένως ένα θέμα με μεγάλο ενδιαφέρον στις τεχνολογικές επιστήμες. Το κύριο χαρακτηριστικό τους είναι ότι οι πρώτες αρχές και λειτουργίες τους βασίζονται στο νευρικό σύστημα των ζώντων οργανισμών (και φυσικά του ανθρώπου), αλλά η μελέτη και η χρήση τους έχει προχωρήσει πολύ πέρα από τους βιολογικούς οργανισμούς, και σήμερα τα νευρωνικά δίκτυα χρησιμοποιούνται για να

λύσουν κάθε είδους προβλήματα με ηλεκτρονικό υπολογιστή. Η φιλοσοφία τους όμως είναι διαφορετική από τον τρόπο με τον οποίο δουλεύουν οι κλασσικοί υπολογιστές. Η λειτουργία τους προσπαθεί να συνδυάσει τον τρόπο σκέψης του ανθρώπινου εγκεφάλου με τον αφηρημένο μαθηματικό τρόπο σκέψης. Έτσι στα νευρωνικά δίκτυα χρησιμοποιούμε ιδέες όπως, π.χ. ένα δίκτυο μαθαίνει και εκπαιδεύεται, θυμάται ή ξεχνά μια αριθμητική τιμή, κλπ. πράγματα που μέχρι τώρα τα αποδίδουμε μόνο στην ανθρώπινη σκέψη. Αλλά βέβαια μπορούν και χρησιμοποιούν επί πλέον και περίπλοκες μαθηματικές συναρτήσεις και κάθε είδους εργαλεία από την μαθηματική ανάλυση. Ένα ιδιαίτερο χαρακτηριστικό είναι ότι οι επιστήμονες στην περιοχή των νευρωνικών δικτύων προέρχονται σχεδόν από όλες τις περιοχές των φυσικών επιστημών, όπως την ιατρική, την επιστήμη μηχανικών, την φυσική, την χημεία, τα μαθηματικά, την επιστήμη υπολογιστών, ηλεκτρολογία, κλπ. Αυτό δείχνει ότι για την ανάπτυξή τους απαιτούνται ταυτόχρονα γνώσεις και θέματα από πολλές περιοχές, ενώ το ίδιο ισχύει και για τις τεχνικές και τις μεθόδους που χρησιμοποιούνται. Έτσι καταλαβαίνει κανείς ότι τα νευρωνικά δίκτυα δίνουν μια νέα πρόκληση στις επιστήμες, καθ' όσον οι νέες γνώσεις που απαιτούνται είναι από τις πιο χρήσιμές στον άνθρωπο, τόσο για την ζωή και την ιατρική, όσο και στην τεχνολογία.

2.2.2 Βιολογική έμπνευση

Η έμπνευση για κάθε μορφής νευρωνικό δίκτυο ξεκινά από την βιολογία. Οι ζώντες οργανισμοί, από τους πιο απλούς μέχρι τον άνθρωπο, έχουν ένα νευρικό σύστημα, το οποίο είναι υπεύθυνο για μια πλειάδα από διεργασίες, όπως είναι η επαφή με τον εξωτερικό κόσμο, η μάθηση, η μνήμη. Το νευρικό σύστημα των οργανισμών αποτελείται από πολλά νευρωνικά δίκτυα τα οποία είναι εξειδικευμένα στις διεργασίες αυτές. Η κεντρική μονάδα του νευρικού συστήματος είναι, οπωσδήποτε, ο εγκέφαλος, ο οποίος επίσης αποτελείται από νευρωνικά δίκτυα. Κάθε νευρωνικό δίκτυο αποτελείται από ένα μεγάλο αριθμό μονάδων, που λέγονται νευρώνες, ή νευρώνια (neurons). Ο νευρώνας είναι η πιο μικρή ανεξάρτητη μονάδα του δικτύου, όπως π.χ. το άτομο είναι η πιο μικρή μονάδα της ύλης. Οι νευρώνες συνεχώς και ασταμάτητα επεξεργάζονται πληροφορίες, παίρνοντας και στέλνοντας ηλεκτρικά σήματα σε άλλους νευρώνες.



Σχήμα 2.8: Δομή ενός τυπικού βιολογικού νευρώνα και η φυσική σύνδεση με γειτονικό νευρώνα μέσω σύναψης.

Πάρθηκε από: http://repfiles.kallipos.gr/html_books/93/04a-main.html

Αναλυτικότερα, ο ανθρώπινος εγκέφαλος αποτελείται κατά κύριο λόγο από ένα ευρύ φάσμα νευρώνων (86.000.000.000 κατά προσέγγιση) (Wikipedia, 2019), οι οποίοι είναι μαζικά διασυνδεδεμένοι με ένα μέσο όρο από διάφορες χιλιάδες διασυνδέσεις ανά νευρώνα. Κάθε νευρώνας είναι ένα εξειδικευμένο κύτταρο το οποίο έχει τη δυνατότητα μετάδοσης ενός ηλεκτροχημικού σήματος. Κάθε νευρώνας αποτελείται από 3 κύρια τμήματα, όπως παρατηρείται στο Σχήμα 2.8, τους δενδρίτες (dendrites), οι οποίοι λειτουργούν ως κανάλια εισόδου για το νευρώνα, το κυρίως κυτταρικό σώμα (cell body) και τον άξονα του κυττάρου-νευροάξονα (axon). Οι άξονες ενός κυττάρου συνδέονται με τους δενδρίτες ενός άλλου, μέσω του σημείου ένωσης που ονομάζεται νευροαξονική απόληξη ή σύναψη (synapse). Ένας νευρώνας μπορεί να λάβει σήματα από ένα σύνολο γειτονικών νευρώνων μέσω των δενδριτών, να τα επεξεργαστεί και να τροφοδοτήσει την έξοδό του μέσω του άξονα προς ένα άλλο σύνολο γειτονικών νευρώνων. Τα σήματα που έρχονται μέσω των δενδριτών «ζυγίζονται» και τα αποτελέσματα αθροίζονται. Όταν το άθροισμα υπερβεί ένα συγκεκριμένο επίπεδο (τιμή κατωφλίου), ο νευρώνας δημιουργεί μια έξοδο ή αλλιώς πυροδοτεί ένα ηλεκτροχημικό σήμα κατά μήκος του άξονα του, το οποίο εν συνεχεία μέσω των συνάψεων θα μεταφερθεί στους γειτονικούς νευρώνες.

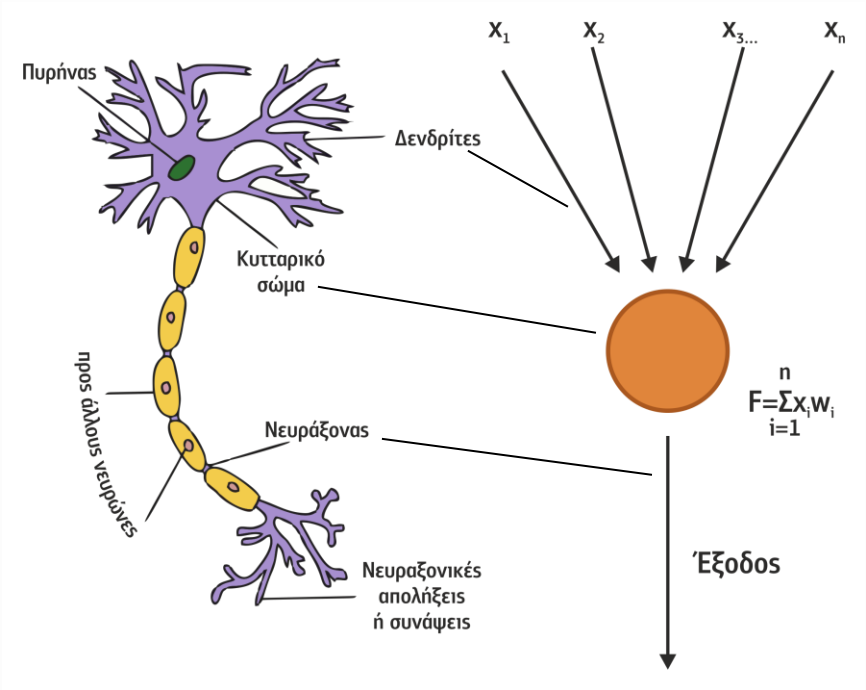
Έτσι λοιπόν, τα νευρωνικά δίκτυα των ζώντων οργανισμών τα ονομάζουμε βιολογικά νευρωνικά δίκτυα, ενθυμούμενοι ότι αυτά είναι και τα πρώτα δίκτυα που μελετήθηκαν, καθ'όσον υπάρχουν σε όλους τους ζώντες οργανισμούς. Οι διεργασίες που επιτελούνται από τα βιολογικά νευρωνικά δίκτυα στους οργανισμούς είναι πολύ περίπλοκες, αλλά και τόσο χρήσιμές στην καθημερινή ζωή του ανθρώπου. Μερικές από αυτές είναι εργασίες ρουτίνας, και τις οποίες ο ανθρώπινος εγκέφαλος εκτελεί με ελάχιστη ή μηδαμινή προσπάθεια, και η αναγνώριση εικόνας. Το ερώτημα που προκύπτει λοιπόν είναι: Μπορούν οι ηλεκτρονικοί υπολογιστές να ανταπεξέλθουν στις ικανότητες που διαθέτει το ανθρώπινο μυαλό;

Δυστυχώς, λόγω του ότι η δομή των υπολογιστών είναι πάρα πολύ διαφορετική από την δομή του εγκεφάλου, δεν μπορεί ευκολά ένας ηλεκτρονικός υπολογιστής να παρέχει τις λειτουργίες που έχει ένα ανθρώπινο μυαλό. Αυτό το ερώτημα έχει οδηγήσει στο να γίνουν κάποιες πρώτες σκέψεις μήπως είναι δυνατόν να δημιουργηθούν κάποια πρότυπα-μοντέλα του νευρωνικού συστήματος του ανθρώπου, τα οποία θα περιέχουν όλα τα χαρακτηριστικά που είναι γνωστά μέχρι σήμερα, και τα οποία θα μπορούσαν από μόνα τους να επιτελέσουν τις εργασίες αυτές, με τον ίδιο τρόπο που γίνονται στα βιολογικά νευρωνικά δίκτυα. Τα μοντέλα αυτά ονομάζονται τεχνητά νευρωνικά δίκτυα (ΤΝΔ-Artificial Neural Networks-ANN).

2.2.3 Αρχιτεκτονική και Λειτουργία

Ένα νευρωνικό δίκτυο είναι ένας μαζικά παράλληλος και διαμοιρασμένος επεξεργαστής, αποτελούμενος από απλούς κόμβους διασυνδεδεμένοι μεταξύ τους, οι οποίοι ονομάζονται τεχνητοί νευρώνες και αντιστοιχούν στους βιολογικούς νευρώνες του ανθρώπινου εγκέφαλου. Κάθε τεχνητός νευρώνας, που στο εξής θα αναφέρεται απλά και ως νευρώνας, λαμβάνει σήματα από το περιβάλλον ή από άλλους νευρώνες, τα συσσωρεύει με αποτέλεσμα να μεταδίδει ένα σήμα σε όλους τους νευρώνες που συνδέονται μαζί του. Οι συνδέσεις αυτές αντιστοιχούν στις συνάψεις ενός βιολογικού δικτύου και ονομάζονται βάρη. Ανάλογα με την θετική ή αρνητική αριθμητική τιμή που έχουν τα βάρη, τα σήματα που εισέρχονται σε ένα νευρώνα ενισχύονται ή υποβαθμίζονται. Η ενεργοποίηση του νευρώνα και η ένταση του σήματος που

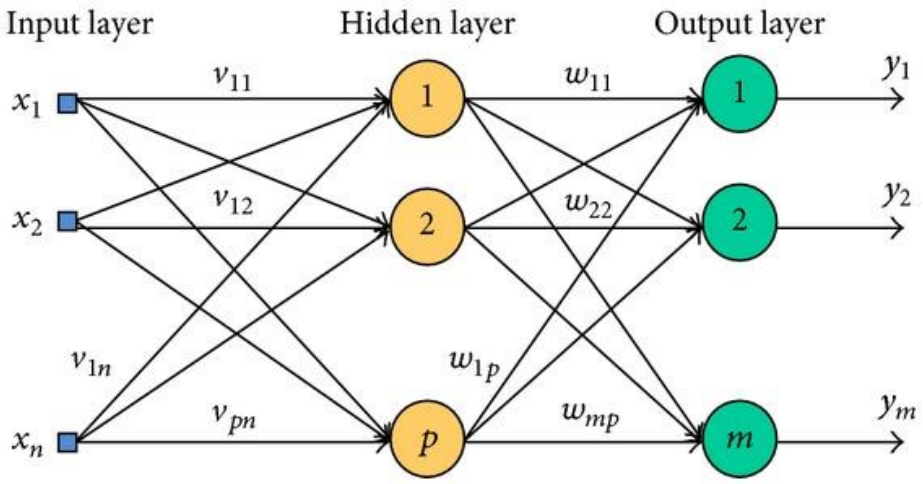
εξέρχεται από το νευρώνα ελέγχονται μέσω μιας συνάρτησης που ονομάζεται συνάρτηση ενεργοποίησης. Ο νευρώνας συλλέγει όλα τα εισερχόμενα σήματα και υπολογίζει το σήμα εισόδου που χρησιμοποιείται ως όρισμα για τη συνάρτηση ενεργοποίησης, η οποία στη συνέχεια υπολογίζει το σήμα εξόδου του νευρώνα.



Σχήμα 2.9: Σύγκριση βιολογικού νευρώνα με τεχνητό νευρώνα. Αριστερά: βιολογικός/φυσικός νευρώνας. Δεξιά: Τεχνητός νευρώνας. Οι δενδρίτες στα βιολογικά νευρωνικά δίκτυα, αναλογούν στα διασυνδεδεμένα βάρη από έναν νευρώνα σε έναν άλλο, το κυτταρικό σώμα του νευρώνα στο βιολογικό νευρώνα, αναλογεί στον τεχνητό κόμβο νευρώνα του τεχνητού νευρωνικού δικτύου. Ο άξονας στο βιολογικό νευρώνα αναλογεί με τη μονάδα εξόδου σε ένα τεχνητό νευρωνικό δίκτυο.
Πάρθηκε από: http://repfiles.kallipos.gr/html_books/93/04a-main.html

Αναλυτικότερα οι νευρώνες χωρίζονται σε 3 είδη: οι νευρώνες εισόδου, οι νευρώνες εξόδου και οι κρυφοί νευρώνες. Οι νευρώνες εισόδου δεν επιτελούν κανέναν υπολογισμό, μεσολαβούν απλώς ανάμεσα στις περιβαλλοντικές εισόδους του δικτύου και στους υπολογιστικούς νευρώνες. Οι νευρώνες εξόδου διοχετεύουν στο περιβάλλον τις τελικές αριθμητικές εξόδους του δικτύου. Οι κρυφοί νευρώνες πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο συναπτικό βάρος και υπολογίζουν το ολικό άθροισμα των γινομένων (Σχήμα 2.9). Το άθροισμα αυτό τροφοδοτείται ως όρισμα στη

συνάρτηση ενεργοποίησης, την οποία υλοποιεί εσωτερικά κάθε κόμβος. Η τιμή που λαμβάνει η συνάρτηση για το εν λόγω όρισμα είναι και η έξοδος του νευρώνα για τις τρέχουσες εισόδους και βάρη λαμβάνοντας υπόψη και την τιμή του κατωφλίου. Η τιμή του κατωφλίου καθορίζει αν ο συγκεκριμένος νευρώνας μπορεί να ενεργοποιηθεί ή όχι. Δηλαδή αν η τιμή της συνάρτησης είναι μεγαλύτερη από την τιμή κατωφλίου, τότε ο νευρώνας υπολογίζει την έξοδο, την οποία προωθεί ως είσοδο στον επόμενο νευρώνα (ή στους επόμενους νευρώνες).

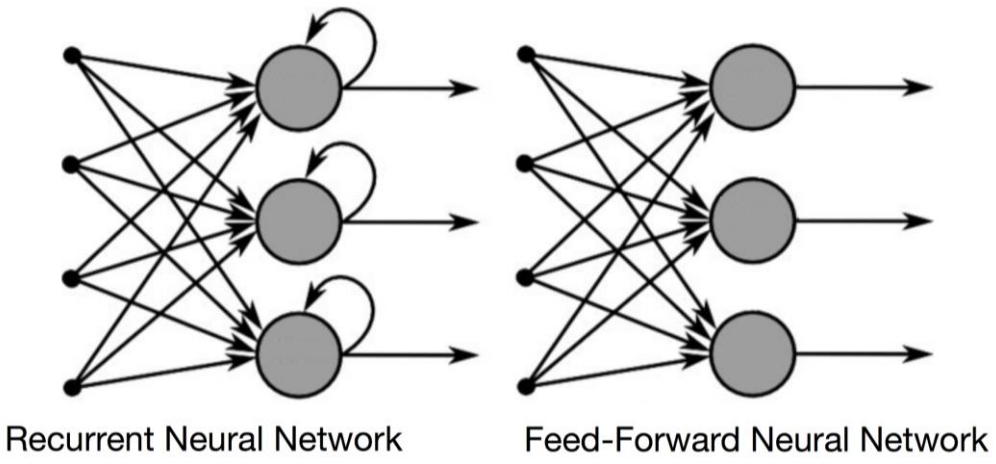


Σχήμα 2.10: Ένα τεχνητό νευρωνικό δίκτυο με ένα κρυφό επίπεδο (hidden layer). Αριστερά: επίπεδο εισόδου (input layer) με 3 τιμές εισόδου (x_1, x_2, x_3) Δεξιά: επίπεδο εξόδου (output layer) με 3 τιμές εξόδου (y_1, y_2, y_3). Τα v_{11}- v_{pn} και w_{11}-w_{mp} είναι οι συνδέσεις, ή αλλιώς τα λεγόμενα βάρη, μεταξύ των νευρώνων.
Πάρθηκε από: https://www.researchgate.net/figure/Neural-network-with-one-hidden-layer-of-neurons_fig5_258524366

Γενικά, ο τρόπος με τον οποίο είναι διασυνδεδεμένοι οι νευρώνες ενός δικτύου ονομάζεται τοπολογία ή αρχιτεκτονική του τεχνητού νευρωνικού δικτύου. Οι νευρώνες είναι λογικά χωρισμένοι σε επίπεδα, με τους νευρώνες του ίδιου επιπέδου να έχουν την ίδια συνάρτηση ενεργοποίησης και συνεπώς να έχουν παρόμοια συμπεριφορά. Συγκεκριμένα, τα τεχνητά νευρωνικά δίκτυα είναι οργανωμένα επίπεδα στα οποία το σήμα μεταφέρεται από το πρώτο επίπεδο, το οποίο είναι το επίπεδο εισόδου (input layer) και αποτελείται από τους νευρώνες εισόδου, στο κρυφό επίπεδο (hidden layer),

το οποίο αποτελείται από τους κρυφούς νευρώνες, και έπειτα το σήμα καταλήγει στο τελευταίο επίπεδο, το όποιο είναι το επίπεδο εξόδου (output layer) που περιέχει τους νευρώνες εξόδου (Σχήμα 2.10).

Όσον αφορά το πώς είναι συνδεδεμένοι οι νευρώνες μεταξύ τους, υπάρχουν δυο βασικές κατηγορίες ΤΝΔ, την πρόσθια τροφοδότησης (feed forward) και την οπίσθια τροφοδότησης (feed backward). Στα νευρωνικά δίκτυα πρόσθιας τροφοδότησης, οι νευρώνες είναι οργανωμένοι σε διαφορετικά επίπεδα, ώστε οι νευρώνες του ενός επιπέδου να τροφοδοτούν τους νευρώνες του επόμενου επιπέδου, έως ότου τροφοδοτηθούν και οι νευρώνες του τελευταίου επιπέδου (Σχήμα 2.11). Δηλαδή, δεν υπάρχει νευρώνας εξόδου ενός επιπέδου που να αποτελεί νευρώνας εισόδου του ίδιου ή προηγούμενων επιπέδου. Τέτοια ΤΝΔ είναι τα δίκτυα οπισθοδιάδοσης (backpropagation). Στα οπισθίως τροφοδοτούμενα δίκτυα, που καλούνται και ανατροφοδοτούμενα ΤΝΔ (recurrent ANN), επιτρέπεται στους νευρώνες ενός επιπέδου να τροφοδοτούν και νευρώνες του ιδίου επιπέδου ή και προηγούμενων επιπέδων. Αν και τα ανατροφοδοτούμενα δίκτυα είναι πολύ χρήσιμα, τα περισσότερα δίκτυα ακολουθούν την πρόσθια τροφοδότηση.



Σχήμα 2.11: Σύγκριση νευρωνικών δικτύων πρόσθιας και οπίσθιάς τροφοδότησης. Αριστερά: Οπισθίως τροφοδοτούμενο δίκτυο (Recurrent Neural Network), Δεξιά: νευρωνικό δίκτυο πρόσθιας τροφοδότησης (Feed-Forward Neural Network). Πάρθηκε από: https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5

2.2.4 Εκπαίδευση νευρωνικών δικτύων

Όπως προαναφέρθηκε, τα τεχνητά νευρωνικά δίκτυα αποτελούν μια προσπάθεια προσέγγισης της λειτουργίας του ανθρώπινου εγκεφάλου, με αποτέλεσμα να υιοθετούν κάποια κοινά χαρακτηριστικά του ανθρώπινου εγκεφάλου. Όπως για παράδειγμά, έχουν την ικανότητα να μαθαίνουν και να μοντελοποιούν μη γραμμικές και πολύπλοκες σχέσεις, πράγμα που είναι πραγματικά σημαντικό επειδή στην πραγματική ζωή, πολλές από τις σχέσεις μεταξύ εισροών και εξόδων είναι μη γραμμικές και πολύπλοκες. Επίσης τα τεχνητά νευρωνικά δίκτυα έχουν τη δυνατότητα για υψηλή ανοχή σφάλματος και έχουν την ιδιότητα της γενίκευσης. Δηλαδή ένα τεχνητό νευρωνικό δίκτυο, αφού μάθει από τις αρχικές εισόδους και τις σχέσεις μεταξύ τους, μπορεί να συμπεράνει αόρατες σχέσεις και σε αόρατα δεδομένα, καθιστώντας έτσι το μοντέλο γενικευμένο και προβλέψιμο σε αόρατα δεδομένα. Με βάση τα πιο πάνω, υπάρχει ένα μεγάλο εύρος προβλημάτων που επιλύονται αποδοτικότερα από τα νευρωνικά δίκτυα. Εφαρμογή των νευρωνικών δικτύων έχει γίνει σε πεδία όπως η προσέγγιση συναρτήσεων, η πρόγνωση και η πρόβλεψη, η δημιουργία μοντέλων μη γραμμικών συστημάτων, η σύνθεση και η αναγνώριση φωνής, η συμπίεση ήχου και εικόνας.

Όμως για να μπορούν να πραγματοποιηθούν τα πιο πάνω χαρακτηριστικά των νευρωνικών δικτύων πρέπει να εκπαιδευτούν, όπως συμβαίνει και σε ένα ανθρώπινο εγκέφαλο, ούτως ώστε να μπορούν να μάθουν. Συγκεκριμένα, κατά την διάρκεια της εκπαίδευσης το μόνο πράγμα που αλλάζει είναι η τιμές των βαρών, των συνδέσεων των νευρώνων. Οι αλλαγές στις τιμές των βαρών δεν γίνονται πάντα με τον ίδιο τρόπο, αλλά εξαρτάται σημαντικά από την μέθοδο που θα χρησιμοποιηθεί (το είδος μάθησης που θα έχει το δίκτυο).

Επιβλεπόμενη μάθηση (Supervised Learning)

Η εκπαίδευση αυτή γίνεται με το να παρουσιάσουμε μια ομάδα από πρότυπα στο δίκτυο, αντιπροσωπευτικά ή παρόμοια με αυτά που θέλουμε να μάθει το δίκτυο. Αυτό σημαίνει ότι δίνουμε στο δίκτυο ως εισόδους κάποια πρότυπα για τα οποία ξέρουμε ποια πρέπει να είναι η επιθυμητή έξοδος στο δίκτυο. Το δίκτυο με τα δεδομένα αυτά τροποποιεί την εσωτερική του δομή ώστε να κάνει την ίδια αντιστοιχία που του δώσαμε

εμείς. Ακολούθως, αφού βρει την σωστή εσωτερική δομή, τότε θα μπορεί να λύνει και άλλα ανάλογα προβλήματα τα οποία δεν τα έχει δει προηγουμένως, δηλ. δεν έχει εκπαιδευθεί στα πρότυπα των προβλημάτων αυτών. Γενικότερα, ένα δίκτυο δέχεται τις παραδειγματικές εισόδους καθώς και τα επιθυμητά αποτελέσματα από έναν «δάσκαλο», και ο στόχος είναι να μάθει έναν γενικό κανόνα προκειμένου να αντιστοιχίσει τις εισόδους με τα αποτελέσματα. Αυτή η διαδικασία εκπαίδευσης χρησιμοποιείται σε προβλήματα ταξινόμησης, πρόγνωσης και διερμηνείας (http://repfiles.kallipos.gr/html_books/93/04a-main.html). Μερικοί αλγόριθμοί μάθησης υπό επίβλεψή είναι ο κανόνας δέλτα και ο αλγόριθμος ανάστροφής μετάδοσης λάθους (back propagation) οι οποίοι θα αναλυθούν περισσότερα στην συνέχεια.

Μη επιβλεπόμενη μάθηση (Unsupervised Learning)

Η εκπαίδευση χωρίς επίβλεψη δεν χρησιμοποιεί κάποιον «δάσκαλο» και βασίζεται μόνο σε τοπικές πληροφορίες καθ' όλη τη διάρκεια της εκπαίδευσής του τεχνητού νευρωνικού δικτύου. Αναφέρεται επίσης ως αυτό-οργάνωση, με την έννοια ότι αυτό-οργανώνει τα δεδομένα που παρουσιάζονται στο δίκτυο και εντοπίζει σημαντικές συλλογικές ιδιότητες τους. Συγκεκριμένα το δίκτυο προσπαθεί από μόνο του να εντοπίζει κοινά χαρακτηριστικά των δεδομένων εισόδου που του παρουσιάζονται, χωρίς να του είναι γνωστά τα επιθυμητά αποτελέσματα εξόδου, ούτως ώστε να τα ομαδοποιήσει σε διάφορες κατηγορίες. Αυτή η διαδικασία εκπαίδευσης χρησιμοποιείται σε προβλήματα ανάλυσης συσχετισμών και ομαδοποίησης (clustering). Παραδείγματα μη επιβλεπόμενης μάθησης είναι το δίκτυο Kohonen και ο αλγόριθμος Hebbian (Wikipedia, 2019).

Ενισχυτική μάθηση (Reinforcement Learning)

Η μάθηση ενίσχυσης είναι ένας τομέας μηχανικής μάθησης που αφορά τον τρόπο με τον οποίο οι υπολογιστικοί πράκτορες πρέπει να αναλάβουν δράση σε ένα περιβάλλον ώστε να μεγιστοποιήσουν κάποια έννοια αθροιστικής ανταμοιβής. Πιο συγκεκριμένα το δίκτυο που εκπαιδεύεται με ενισχυτική μάθηση δεν χρειάζεται να του παρουσιαστούν

τα ζεύγη εισόδου και εξόδου αλλά από μόνο του μαθαίνει βάση κάποιας ανατροφοδότησης που παίρνει από ένα κριτή. Δηλαδή, ο κριτής κρίνει την κίνηση που έχει κάνει ο πράκτορας και την αξιολογά είτε θετικά (επιβράβευση-award) είτε αρνητικά (τιμωρία-penalty) και βάση αυτής της κριτικής ο πράκτορας μαθαίνει και αποφασίζει ποια θα είναι η επόμενη κίνησή που θα εκτελέσει ούτως ώστε να ελαχιστοποιήσει τις τιμωρίες. Η εκπαίδευση αυτή χρησιμοποιείται κυρίως σε προβλήματα Σχεδιασμού (Planning), όπως για παράδειγμα ο έλεγχος κίνησης ρομπότ και η βελτιστοποίηση εργασιών σε εργοστασιακούς χώρους.

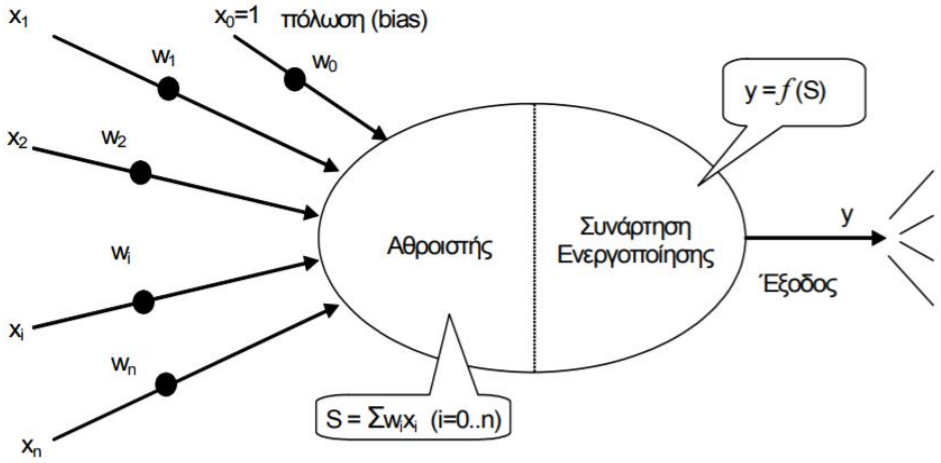
2.2.5 Μοντέλο Νευρώνα McCulloh και Pitts

Το πρώτο υπολογιστικό μοντέλο ενός νευρώνα προτάθηκε από τον Warren MuCulloch (neuroscientist) και τον Walter Pitts (logician) το 1943. Όπως προαναφέρθηκε ένας τεχνητός νευρώνας παίρνει δεδομένα εισόδου και τα επεξεργάζεται με αποτέλεσμα να παραχθούν συγκεκριμένα αποτελέσματα. Συγκεκριμένα όπως φαίνεται και στην εικόνα πιο κάτω (Σχήμα 2.12), ο νευρώνας δέχεται ένα διάνυσμα τιμών εισόδων (x_1, x_2, x_i, x_n) το οποίο πολλαπλασιάζεται με το αντίστοιχο διάνυσμα βαρών (w_1, w_2, w_i, w_n) και έπειτα συναθροίζονται όλα τα γινόμενα που προκύψαν (Εξίσωση 2.1).

$$\text{Sum} = \sum_{i=1}^{n} \text{Xi} * \text{Wi}$$

Εξίσωση 2.1: Εξίσωση συνολικής τιμής εισόδου ενός τεχνητού νευρώνα, όπου Xi είναι η τιμή εισόδου και Wi το αντίστοιχο βάρος

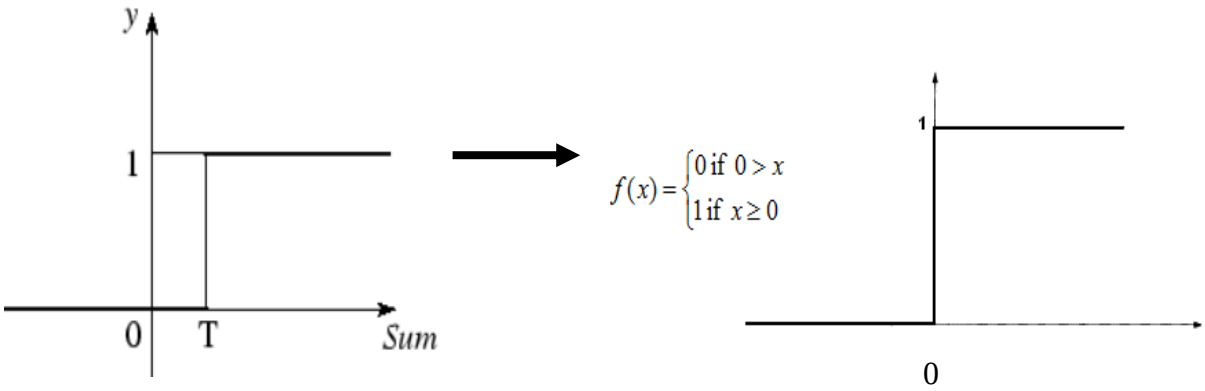
Στη συνέχεια το άθροισμα αυτό (Sum) δίνεται σαν είσοδος στην συνάρτηση ενεργοποίησης $y = f(S)$ όπου S = Sum η οποία θα καθορίσει την τελική έξοδο του νευρώνα. Η πρώτη συνάρτηση ενεργοποίησης ήταν η βηματική συνάρτηση ή συνάρτηση σκαλί (Σχήμα 2.13), η οποία όμως είχε το μειονέκτημα ότι δεν μας έδινε αρκετή πληροφορία στο πόσο κοντά ή μακριά ήταν το πραγματικό μας αποτέλεσμα σε σύγκριση με την επιθυμητή έξοδο του νευρώνα.



Σχήμα 2.12: Τεχνητός νευρώνας McCulloh και Pitts

Πάρθηκε από: http://aibook.csd.auth.gr/include/slides/Chap19.pdf

Πιο συγκεκριμένα όταν το άθροισμα sum ξεπεράσει μια συγκεκριμένη τιμή που καθορίστηκε (τιμή κατωφλίου) τότε μας δίνει το αποτέλεσμα 1 , διαφορετικά την τιμή 0. Για να εξαφανίσουμε την τιμή του κατωφλίου, ή πιο συγκεκριμένα να μετακινήσουμε την τιμή του κατωφλίου στον άξονα y (ίσο με 0), προστέθηκε ακόμη μια τιμή στο διάνυσμα των δεδομένων εισόδων με θετική τιμή 1 ή οποία ονομάζεται πόλωση (bias) όπως φαίνεται και στο Σχήμα 2.13.



Σχήμα 2.13: Βηματική συνάρτηση κατωφλίου. Αριστερά: Βηματική συνάρτηση ενεργοποίησης με τιμή κατωφλίου = T, Δεξιά: Βηματική συνάρτηση ενεργοποίησης με τιμή κατωφλίου = 0

Πάρθηκε από: http://wwwold.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node12.html

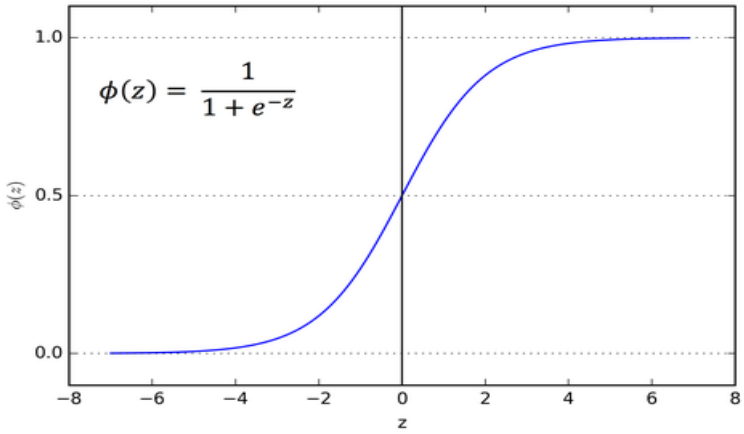
2.2.6 Συναρτήσεις ενεργοποίησης

Βηματική Συνάρτηση

Όπως έχει αναφερθεί πιο πάνω η βηματική συνάρτηση, ή αλλιώς συνάρτηση σκαλί, ήταν η αρχική συνάρτηση ενεργοποίησης που χρησιμοποιήθηκε σε ένα τεχνητό νευρώνα McCulloch και Pitts. Αφού υπολογιστεί το συνολικό άθροισμα των γινομένων των δεδομένων εισόδου με των αντιστοίχων βαρών, συγκρίνεται με μία προκαθορισμένη τιμή (τιμή κατωφλίου T) και αν τότε η τιμή του συνολικού αθροίσματος είναι μεγαλύτερη από την τιμή κατωφλίου τότε λέμε ο νευρώνας πυροδοτεί, δηλαδή παίρνει την τιμή 1, αλλιώς παίρνει την τιμή 0. Η συνάρτηση αυτή γραμμική με αποτέλεσμα να μην είναι αποδοτική αφού μόνο μια τιμή μπορεί να μας δώσει (0 ή 1) και αυτό έχει σαν συνέπεια να μην γνωρίζουμε πόσο κοντά ή μακριά βρίσκεται το τελικό αποτέλεσμα του νευρώνα από το κατώφλι. Επίσης ένα σημαντικό μειονέκτημα που έχει η συγκεκριμένη συνάρτηση ενεργοποίησης είναι ότι δεν είναι παραγωγίσιμη σε όλο το πεδίο ορισμού που αυτό θα επηρεάσει την λειτουργία της μεθόδου κατάβασης κλίση που θα αναλυθεί στο υποκεφάλαιο 2.2.7.2)

Σιγμοειδής Συνάρτηση

Η σιγμοειδής συνάρτηση είναι μια μη γραμμική συνάρτηση με μορφή «S» και με πεδίο τιμών μεταξύ του 0 και 1 (Σχήμα 2.14). Σε αντίθεση με την βηματική συνάρτηση, η σιγμοειδής μπορεί να επιλύσει μη γραμμικά διαχωρίσιμα προβλήματα όπως η συνάρτηση XOR και γι' αυτό είναι η πιο συνηθισμένη μορφή συνάρτησης ενεργοποίησης που χρησιμοποιείται στην κατασκευή τεχνητών νευρωνικών δικτύων. Επίσης δίνει περισσότερή πληροφορία στο πόσο κοντά ή μακριά βρισκόμαστε στην επιθυμητή έξοδο και είναι παραγωγίσιμη. Όμως ένα αρνητικό της συγκεκριμένης συνάρτησης είναι ότι προκαλείται το πρόβλημα του vanishing gradient (υποκεφάλαιο 2.2.10). Όσο αυξάνεται η παράγωγος της σιγμοειδούς συνάρτησης έχει σαν αποτέλεσμα να μειώνεται η κλίση της συνάρτησής και αυτό οδηγεί σε πολύ μικρές αλλαγές στις τιμές εξόδου. Αυτό το πρόβλημα επηρεάζει αρκετά την εκπαίδευση ενός δικτύου με το να μην μαθαίνει ή να μαθαίνει πάρα πολύ αργά.

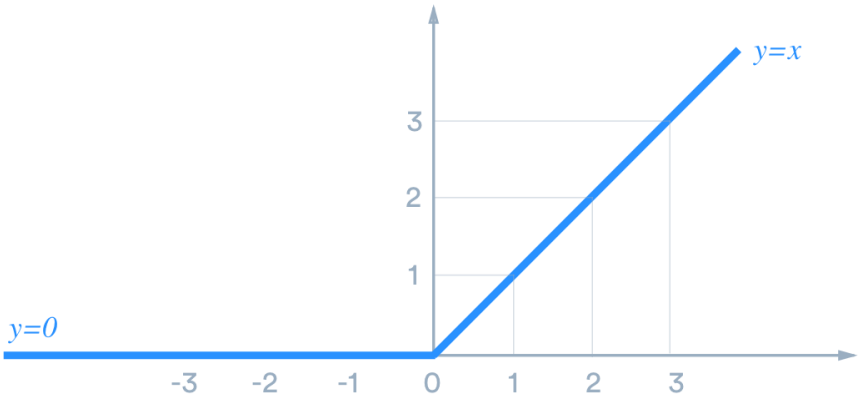


Σχήμα 2.14: Σιγμοειδής συνάρτηση ενεργοποίησης.

Πάρθηκε από: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Συνάρτηση Rectifier

Η rectifier συνάρτηση χρησιμοποιείται κυρίως σε βαθιά τεχνητά νευρωνικά δίκτυα όπου βοηθά στο πρόβλημα που αντιμετωπίζει η σιγμοειδής συνάρτηση (vanishing gradient problem). Αναλυτικότερα η rectifier συνάρτηση, πιο συγκεκριμένα η συνάρτηση ενεργοποίησης RELU είναι η πιο απλή μη γραμμική συνάρτηση ενεργοποίησης όπου θέτει σαν κάτω όριο το 0 ενώ σαν πάνω όριο το άπειρο (Εξίσωση 2.2). Όπως φαίνεται στο πιο κάτω σχήμα η συνάρτηση RELU είναι γραμμική για όλες τις θετικές τιμές και μηδέν σε όλες τις αρνητικές τιμές. Η γραμμικότητα που έχει αυτή η συνάρτηση μειώνει το πρόβλημα μείωση της κλίσης όπως αναφέρθηκε στην σιγμοειδή συνάρτηση.



Σχήμα 2.15: Rectifier συνάρτηση ενεργοποίησης.

Πάρθηκε από: https://www.tinymind.com/learn/terms/relu

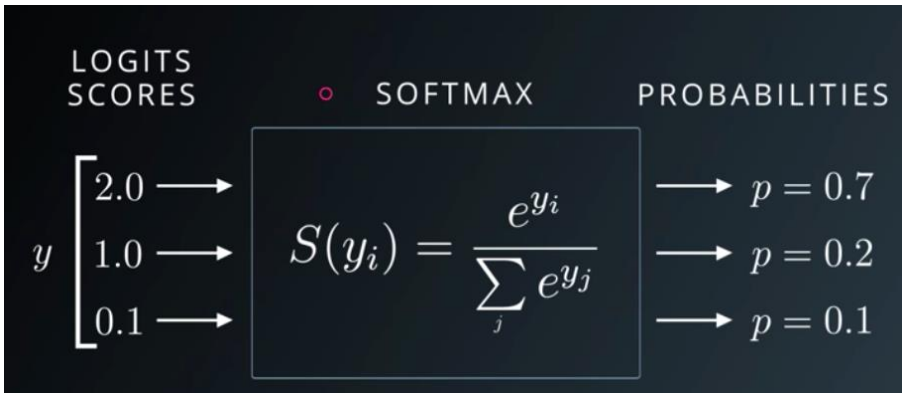
$$f(x) = x^+ = \max(0, x)$$

Εξίσωση 2.2: Εξίσωση συνάρτησης rectifier όπου x μια θετική τιμή.

Βάση των πιο πάνω, δεν υπάρχει αρκετή πολυπλοκότητα στον υπολογισμό τιμών σε αυτή την συνάρτηση ενεργοποίησης, αφού θα υπάρχουν αρκετές τιμές που θα παίρνουν την τιμή 0 και αυτό οδηγεί στην πιο γρήγορη εκπαίδευση. Όμως αυτό το γεγονός είναι επικίνδυνο, ότι ένα μεγάλο μέρος των νευρώνων θα πάρουν την τιμή 0, και μπορεί να προκαλέσει λανθασμένη εκπαίδευση.

Συνάρτηση Softmax

Η συνάρτηση Softmax υπολογίζει την κατανομή πιθανοτήτων του συμβάντος σε διάφορα γεγονότα. Γενικά, η λειτουργία αυτή θα υπολογίσει τις πιθανότητες κάθε κατηγορίας εξόδου σε όλες τις πιθανές ομάδες. Συγκεκριμένα κανονικοιποιεί τα αποτελέσματα του κάθε νευρώνα ούτως ώστε να βρίσκονται στο εύρος του 0 έως του 1 κα το άθροισμα όλων των πιθανοτήτων των αποτελεσμάτων εξόδου για ένα νευρώνα θα είναι ίσο με ένα. Κάθε αποτέλεσμα της softmax συνάρτησης ενεργοποίησης αντιστοιχεί σε μια πιθανότητα κατηγορίας, δηλαδή εκφράζει κατά πόσο το συγκεκριμένο αποτέλεσμα ανήκει στην συγκεκριμένη κατηγορία.



Σχήμα 2.16: Κανονικοποίηση αποτελεσμάτων εξόδου με την χρήση της softmax συνάρτησης. Αριστερά: Αποτελέσματα εξόδου, δεδομένα του τελευταίου επιπέδου του δικτύου, Στη μέση: Η μαθηματική πράξη της softmax συνάρτησης, ο τύπος υπολογίζει

την εκθετική της δεδομένης τιμής εξόδου και το άθροισμα των εκθετικών τιμών όλων των τιμών στις εξόδους. Η αναλογία της εκθετικής τιμής εξόδου και του αθροίσματος των εκθετικών τιμών είναι η έξοδος της συνάρτησης softmax., Δεξιά: Πιθανότητα κατηγορίας, του κάθε αποτελέσματος εξόδου.

Πάρθηκε από: https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d

Η συνάρτηση softmax που χρησιμοποιείται για πολλαπλή ταξινόμηση επιστρέφει τις πιθανότητες κάθε κατηγορίας, και η κατηγορία που έχει την υψηλότερη πιθανότητα τότε είναι και η επιθυμητή κατηγορία. Για παράδειγμα στο Σχήμα 2.16 για το συγκεκριμένο αποτέλεσμα εξόδου, η κατηγορία στην οποία θα το τοποθετήσουμε είναι αυτή με την μεγαλύτερη πιθανότητα, άρα αυτή που αντιστοιχεί στην πιθανότητα ίση με 0.7.

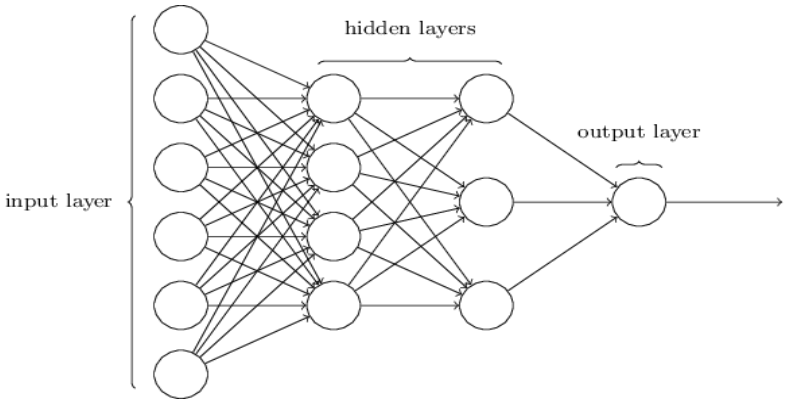
2.2.7 Πολυστρωματικά Δίκτυα Perceptron

Η πιο θεμελιώδης μονάδα ενός νευρικού δικτύου ονομάζεται τεχνητός νευρώνας, ο οποίος παίρνει μια είσοδο, την επεξεργάζεται, περνάει μέσα από μια συνάρτηση ενεργοποίησης η οποία επιστρέφει την ενεργοποιημένη έξοδο. Ένας τεχνητός νευρώνας που έχει αναφερθεί είναι ο νευρώνας McCulloh και Pitts στο Υποκεφάλαιο 2.2.5. Ο Frank Rosenblatt, ένας Αμερικανός ψυχολόγος, πρότεινε το κλασικό μοντέλο perceptron το 1958. Μετά από περισσότερη ανάλυση του νευρώνα του Rosenblatt, από τους Minsky και Papert (1969) το μοντέλο τους αναφέρεται πλέον ως μοντέλο perceptron. Το μοντέλο perceptron είναι ένα γενικότερο υπολογιστικό μοντέλο από τον νευρώνα McCulloch-Pitts. Καταργεί ορισμένους από τους περιορισμούς του νευρώνα McCulloh και Pitts, εισάγοντας την έννοια των αριθμητικών βαρών για τις εισροές (inputs) και έναν μηχανισμό για την εκμάθηση αυτών των βαρών. Οι είσοδοι δεν περιορίζονται πλέον σε τιμές boolean, όπως στην περίπτωση ενός νευρώνα McCulloh και Pitts, αλλά υποστηρίζει επίσης πραγματικές εισροές, γεγονός που καθιστά αυτή τη διαφορά πιο χρήσιμη και γενικευμένη.

Αν συνδέσουμε αρκετούς νευρώνες perceptron δημιουργείται ένα δίκτυο πολλαπλών στρωμάτων perceptron. Για να υπάρχει μια οργάνωση των πολλών νευρώνων σε ένα

δίκτυο πολλαπλών στρωμάτων perceptron κατατάσσονται σε συγκεκριμένα επίπεδα. Αναλυτικότερα υπάρχει ένα επίπεδο εισόδου όπου παρουσιάζονται όλα τα δεδομένα εισόδου σαν διάνυσμα στο δίκτυο, ένα τουλάχιστο κρυφό επίπεδο στο οποίο επεξεργάζονται τα δεδομένα βάση των διάφορων αλλαγών τιμών των βαρών ανάλογα με το πρόβλημα που προσπαθεί το δίκτυο να επιλύσει και ένα επίπεδο εξόδου στο οποίο υπάρχει το τελικό αποτέλεσμα εξόδου (Σχήμα 2.17). Στα δίκτυα αυτής της μορφής, οι νευρώνες ενός επιπέδου είναι συνήθως πλήρως διασυνδεδεμένοι με τους νευρώνες του επόμενου επιπέδου. Με την έννοια πλήρους διασύνδεσης εννοούμε ότι κάθε νευρώνας συνδέεται με την έξοδο όλων των νευρώνων του προηγούμενου επιπέδου.

Προσθέτοντας ένα ή περισσότερα επίπεδα κρυφών νευρώνων σε ένα νευρωνικό δίκτυο, το δίκτυο αποκτά τη δυνατότητα να αφομοιώνει περισσότερες πληροφορίες για τα δεδομένα εισόδου, μέσω των περισσότερων συνάψεων που έχει στη διάθεσή του και των μεγαλύτερης πολυπλοκότητας αλληλεπιδράσεων, μεταξύ των νευρώνων, που δημιουργούνται. Η αξία της δυνατότητας του δικτύου να επεξεργάζεται μεγαλύτερης πολυπλοκότητας δεδομένα, γίνεται φανερή όταν η διάσταση του διανύσματος εισόδου είναι μεγάλη. Η διαδικασία υπολογισμού της εξόδου του δικτύου είναι παρόμοια με των δικτύων ενός επιπέδου. Οι κόμβοι του επιπέδου εισόδου του δικτύου παρέχουν το διάνυσμά εισόδου του δικτύου στους κόμβους του 1ου κρυφού επιπέδου. Εκτελείται ο υπολογισμός στους κόμβους του 1ου επιπέδου και στη συνέχεια η έξοδος του 1ου κρυφού επιπέδου παρέχεται σαν είσοδος στους νευρώνες του 2ου κρυφού επιπέδου. Αφού εκτελεστεί ο υπολογισμός, η έξοδος μεταφέρεται στο επόμενο επίπεδο νευρώνων, και η διαδικασία συνεχίζεται, με το ένα επίπεδο να χρησιμοποιεί σαν είσοδο την έξοδο του προηγούμενου, μέχρι να γίνει ο υπολογισμός της τελικής εξόδου του δικτύου από το τελευταίο επίπεδο. Αυτή όλη η διαδικασία χαρακτηρίζεται μια προς τα εμπρός τροφοδότηση. Με τον όρο εμπρός τροφοδότησης εννοούμε ότι η φορά μεταφοράς της πληροφορίας σε ένα δίκτυο είναι μόνο προς τα μπροστά. Τα νευρωνικά δίκτυα προς τα εμπρός τροφοδότησης πολλαπλών επιπέδων είναι το συχνότερα χρησιμοποιούμενο και περισσότερο μελετημένο είδος νευρωνικών δικτύων. Γιατί; Ένας μόνο νευρώνας perceptron χρησιμοποιείται μόνο για την επίλυση γραμμικά διαχωρίσιμων προβλημάτων όπως ένας νευρώνας McCulloh-Pitts. Αντίθετα τα δίκτυα πολλαπλών επιπέδων perceptron εμπρόσθιου περάσματος μπορούν να επιλύσουν και μη γραμμικά διαχωρίσιμα προβλήματα όπως για παράδειγμα η γνωστή συνάρτηση XOR.



Σχήμα 2.17: Πολυστρωματικό δίκτυο Perceptron εμπρόσθιου περάσματος με 2 κρυφά επίπεδα. Πάρθηκε από: https://github.com/rcassani/mlp-example

2.2.7.1 Αλγόριθμος μάθησης Perceptron

Ο αλγόριθμος μάθησης Perceptron εκπαιδεύει δίκτυα από τεχνητούς νευρώνες McCulloh και Pitts και συγκεκριμένα η εκπαίδευση που εκτελείται είναι επιβλεπόμενης μάθησης. Αρχικά, γίνεται μια τυχαία αρχικοποίηση των κατωφλιών και των βαρών και έπειτα παρουσιάζουμε στο δίκτυο τα δεδομένα εισόδου και τα αντίστοιχα επιθυμητά δεδομένα εξόδου. Αφού το δίκτυο υπολογίσει την πραγματική έξοδο του δικτύου γίνεται μια σύγκριση του πραγματικού και του επιθυμητού αποτελέσματος και ανάλογα προσαρμόζονται τα βάρη και τα κατώφλια (Πίνακας 2.3)

- Ελέγχετε εάν η y που δίνει ο νευρώνας για το δείγμα x είναι η αναμενόμενη.
  - Εάν είναι, η διαδικασία εκπαίδευσης προχωρά στο επόμενο δείγμα.
  - Εάν όχι, τότε:
    - εάν η σωστή έξοδος είναι μεγαλύτερη από αυτήν που υπολόγισε ο νευρώνας, ο κανόνας εκμάθησης αυξάνει τα βάρη των εισόδων που είναι θετικές και μειώνει τα βάρη των εισόδων που είναι αρνητικές.
    - εάν η σωστή έξοδος είναι μικρότερη από αυτήν που υπολόγισε ο νευρώνας, ο κανόνας εκμάθησης μειώνει τα βάρη των εισόδων που είναι θετικές και αυξάνει τα βάρη των εισόδων που είναι αρνητικές.
- Η διαδικασία αυτή εκτελείται, μέχρις ότου ο νευρώνας να απαντά σωστά σε όλα τα δείγματα ή να μη βελτιώνει πλέον σημαντικά την απόδοσή του.

Πίνακας 2.3: Αλγόριθμος μάθησης Perceptron, όπου y: πραγματική έξοδος, x: είσοδος, κανόνας εκμάθησης: κανόνας Δέλτα.

Πάρθηκε από: http://repfiles.kallipos.gr/html_books/93/04a-main.html

Αυτή η αλλαγή των βαρών γίνεται βάση ενός κανόνα εκμάθησης και ένα από τους γνωστότερους είναι ο κανόνας Δέλτα (Εξίσωση 2.3) (Window and Hoff, 1960) .

$$Wi(t+1) = Wi(t) + n\Delta Xi(t)$$

Εξίσωση 2.3: Κανόνας Δέλτα, όπου Wi(t+1) είναι το νέο βάρος εισόδου από τον νευρώνα i, Wi(t) είναι το προηγούμενο βάρος εισόδου από τον νευρώνα i ,n είναι ο ρυθμός εκμάθησης, Xi(t) η είσοδος του νευρώνα i και Δ είναι το σφάλμα (Εξίσωση 2.4).

Ο ρυθμός εκμάθησης n καθορίζει το πόσο γρήγορα συγκλίνει η μάθηση. Μεγάλος ρυθμός μάθησης μπορεί να οδηγήσει σε γρηγορότερη σύγκλιση και σε ταλάντωση γύρω από τις βέλτιστες τιμών βαρών. Μικρός ρυθμός μάθησης έχει ως αποτέλεσμα πιο αργή σύγκλιση, ενώ μπορεί να οδηγήσει σε παγίδευση σε τοπικά ακρότατα.

$$\Delta = d(t) - y(t)$$

Εξίσωση 2.4: Σφάλμα Δ, όπου d(t) είναι η επιθυμητή έξοδος και y(t) η πραγματική έξοδος.

2.2.7.2 Μέθοδος κατάβασης κλίσης

Η μέθοδος κατάβασης κλίσης είναι ένας από τους πιο δημοφιλείς και ευρέως χρησιμοποιημένους αλγόριθμους βελτιστοποίησης για την ελαχιστοποίηση κάποιας συνάρτησης. Συγκεκριμένα η μέθοδος κατάβασης κλίσης προσπαθεί να αλλάξει τις τιμές των βαρών του δικτύου για να εντοπίσει το ολικό ελάχιστο της συνάρτησης που χρησιμοποιείται. Για να γινεί αυτό πρέπει η αλλαγή των βαρών να είναι ανάλογη ως προς το αρνητικό της παραγώγου της συνάρτησης σφάλματος ως προς τα βάρη (Εξίσωση 2.3).

$$\Delta wij = -\mathrm{n}\,\frac{\mathrm{dE}}{\mathrm{dwij}}$$

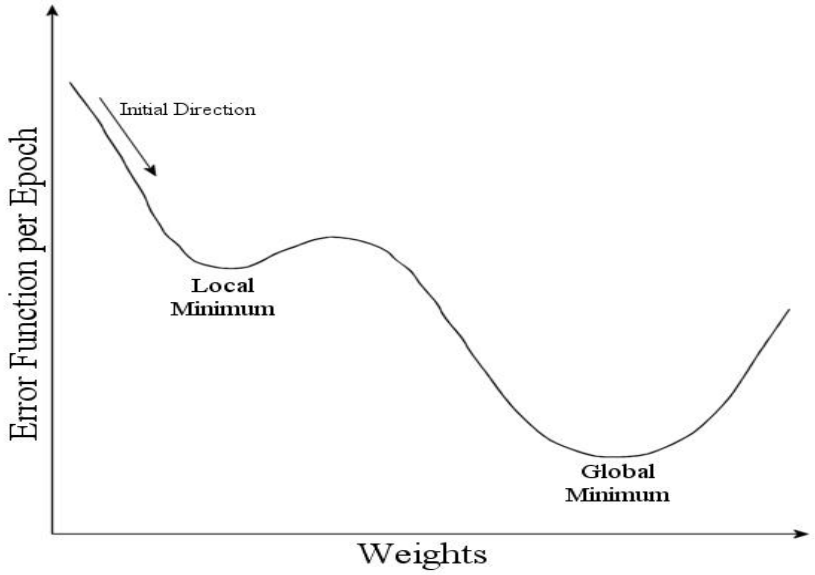
Εξίσωση 2.5: Αλλαγή των βαρών να είναι ανάλογη, ως προς το αρνητικό της παραγώγου της συνάρτησης σφάλματος, ως προς τα βάρη. Όπου n είναι ο ρυθμός μάθησης και E το σφάλμα εκμάθησης (Εξίσωση 2.4)

Έπειτα υπολογίζεται το σφάλμα του δικτύου μέσω του αλγορίθμου ελάχιστων τετραγώνων (Least Mean Square Algorithm).

$$E = \frac{1}{2}\sum_{j=1}^{m}\left(t_{pj} + o_{pj}\right)^2$$

Εξίσωση 2.6: Συνάρτηση σφάλματος, όπου t_{pj} είναι το επιθυμητό αποτέλεσμα και o_{pj} το πραγματικό αποτέλεσμα.

Ουσιαστικά όταν λέμε ότι η μέθοδος κατάβασης κλίσης προσπαθεί να αλλάξει τα βάρη του δικτύου, ούτως ώστε το σφάλμα να ελαχιστοποιηθεί όσο το δυνατό περισσότερο εννοούμε ότι πολύ απλά προσπαθεί να φέρει όσο πιο κοντά τις τιμές των πραγματικών αποτελεσμάτων, που υπολογίζονται στο δίκτυο, στις τιμές των επιθυμητών αποτελεσμάτων. Δυστυχώς, κατά την διάρκεια εύρεσης του ολικού ελάχιστου της συνάρτησης που χρησιμοποιείται μπορεί να προκύψει κάποιος εγκλωβισμός στις τιμές των βαρών σε ένα τοπικό ελάχιστο (Σχήμα 2.18) με αποτέλεσμα να θεωρηθεί το τοπικό ελάχιστο που εντοπίστηκε ως ολικό ελάχιστο και να μην παραχθούν τα βέλτιστα αποτελέσματα. Ένας τρόπος αποφυγής αυτού του προβλήματος είναι η ορμή η οποία αυξάνει το μέγεθος βημάτων που γίνονται κατά την εύρεση του ολικού ελαχίστου ούτως ώστε να μην μπορέσουν οι τιμές των βαρών να ‘κολλήσουν’ σε ένα τοπικό ελάχιστο. Επίσης κάποιοι άλλοι τρόποι που μπορούν να βοηθήσουν στην αντιμετώπιση αυτού του προβλήματος είναι η αύξηση επιπλέον εσωτερικών νευρώνων και η μείωση ρυθμού μάθησης (n) .



Σχήμα 2.18: Γραφική παράσταση, που απεικονίζει την συνάρτηση των σφαλμάτων ως προς τα βάρη, στην οποία θα εγκλωβιστεί η τιμή των βαρών στο τοπικό ελάχιστο (local minimum) αφού θα είναι και το πρώτο ελάχιστο που θα εντοπιστεί.
Πάρθηκε από: https://www.cse.unsw.edu.au/~cs9417ml/MLP2/BackPropagation.html

2.2.7.3 Αλγόριθμος ανάστροφης μετάδοσης σφάλματος

Ο αλγόριθμος ανάστροφης μετάδοσης σφάλματος χρησιμοποιείται ως αλγόριθμος επιβλεπόμενης μάθησης στα πολυεπίπεδα τεχνητά νευρωνικά δίκτυα. Όπως είχε αναλυθεί στα προηγούμενα υποκεφάλαια υπολογίζεται μια αλλαγή των βαρών κατά την εκπαίδευση του δικτύου ούτως ώστε να ελαχιστοποιηθεί όσο το δυνατό το σφάλμα μεταξύ των πραγματικών και επιθυμητών εξόδων. Ο αλγόριθμος ανάστροφης μετάδοσης σφάλματος στην ουσία μεταφέρει το σφάλμα, που υπολογίζεται στο τελευταίο επίπεδο, προς τα πίσω για να προσαρμοστούν σωστά τα βάρη των νευρώνων (Σχήμα 2.19). Ο αλγόριθμος αυτός χωρίζεται σε δυο φάσεις, το εμπρόσθιο πέρασμα και το προς τα πίσω πέρασμα. Αρχικά εκτελείται η πρώτη φάση η οποία είναι το εμπρόσθιο πέρασμα, που στην ουσία είναι η διαδικασία υπολογισμού της πραγματικής εξόδου όπως αναλύθηκε στο υποκεφάλαιο 2.2.7 Αφού υπολογιστεί η πραγματική έξοδος υπολογίζεται το σφάλμα για τους νευρώνες του επιπέδου εξόδου με την Εξίσωση 2.7. Έπειτα εκτελείται η δεύτερη φάση στην οποία γίνεται ένα πέρασμα προς τα πίσω. Συγκεκριμένα το σφάλμα που υπολογίστηκε στο εμπρόσθιο πέρασμα μεταφέρεται προς

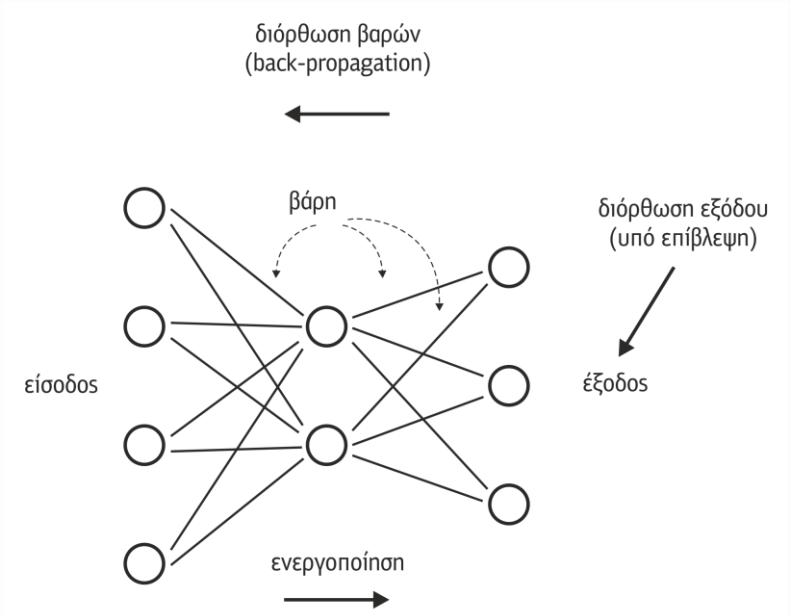
τα πίσω, δηλαδή από το επίπεδο εξόδου μεταφέρεται στα κρυφά επίπεδα και από τα κρυφά επίπεδα φτάνει στο επίπεδο εισόδου. Ο υπολογισμός του σφάλματος στα κρυφά επίπεδα διαφέρει από αυτόν στο επίπεδο εξόδου (Εξίσωση 2.8).

$$\delta_{pj} = o_{pj}\,(1 - o_{pj})(t_{pj} - o_{pj})$$

Εξίσωση 2.7: Υπολογισμός σφάλματος για τους νευρώνες του επιπέδου εξόδου. Όπου tpj είναι το επιθυμητό αποτέλεσμα και opj το πραγματικό αποτέλεσμα.

$$\delta_{pj} = o_{pj}\,(1 - o_{pj})\sum_{i=1}^{m}(\delta_{pi} * w_{pi})$$

Εξίσωση 2.8: Υπολογισμός σφάλματος για τους νευρώνες του κρυφού επιπέδου. Όπου δpi είναι το σφάλμα που υπολογίζεται από την Εξίσωση 2.7, δηλαδή το σφάλμα στο επίπεδο εξόδου, τώρα το δpj είναι το σφάλμα στο κρυφό επίπεδο και m είναι ο αριθμός νευρώνων στο επόμενο επίπεδο με τους οποίους συνδέεται ο νευρώνας j.



Σχήμα 2.19: Εκπαίδευση δικτύου με τον αλγόριθμος ανάστροφης μετάδοσης σφάλματος. Πάρθηκε από: http://repfiles.kallipos.gr/html_books/93/04a-main.html

2.2.8 Συνελικτικά Νευρωνικά Δίκτυα

2.2.8.1 Γενικά

Η έμπνευση για τα νευρωνικά δίκτυα, όπως έχει ήδη αναφερθεί, προήλθε από τη δομή και τη λειτουργία του ανθρώπινου εγκεφάλου. Γενικά, όπως είχε παρατηρήσει από νωρίς η επιστήμη της νευροβιολογίας, το νευρικό σύστημα αποτελείται από πολλαπλά επίπεδα επεξεργασίας της πληροφορίας που λαμβάνει από τους αισθητήρες του. Εξαιτίας αυτού είναι σε θέση να "ξετυλίγει" τη σύνθετη πληροφορία και να δημιουργεί μια πλούσια και λεπτομερή αναπαράστασή της για την καλύτερη κατανόηση και αξιοποίησή της. Ήταν, συνεπώς, εύλογο να εφαρμοστεί μια παρόμοια τεχνική και στα τεχνητά νευρωνικά δίκτυα με αποτέλεσμα να προταθούν πολλά νέα μοντέλα και είδη από το 1980 μέχρι και σήμερα.

Η ταξινόμηση εικόνας είναι στην ουσία η λήψη μιας εικόνας εισόδου και από αυτή η αναγνώριση μιας κλάσης (μια γάτα, σκύλος, κ.λπ.) ή μια πιθανότητα τάξεων που περιγράφουν καλύτερα την εικόνα. Για τον άνθρωπο, αυτό το καθήκον αναγνώρισης είναι μία από τις πρώτες δεξιότητες που μαθαίνει από τη στιγμή που γεννιέται και είναι αυτή που έρχεται φυσικά και αβίαστα ως ενήλικες. Χωρίς να σκεφτεί δύο φορές, είναι σε θέση να αναγνωρίσει γρήγορα και αδιάλειπτα το περιβάλλον στο οποίο βρίσκεται καθώς και τα αντικείμενα που τον περιβάλλουν. Όταν βλέπει μια εικόνα ή μόλις κοιτάξει τον κόσμο γύρω του, τις περισσότερες φορές χαρακτηρίζει αμέσως τη σκηνή, δηλαδή δίνει σε κάθε αντικείμενο μια ετικέτα, χωρίς καν να συνειδητοποιήσει ότι κατηγοριοποίησε τα αντικείμενα που είχε δει (Deshpande, 2017). Με αυτό το φαινόμενο οι επιστήμονες προσπάθησαν να το εφαρμόσουν σε τεχνητά μοντέλα. Ένα μοντέλο που είχε δημιουργηθεί λοιπόν, κυρίως για την υπολογιστική όραση, για την αναγνώριση ομιλίας και για την επεξεργασία και ανάλυση εικόνων είναι τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks-CNN).

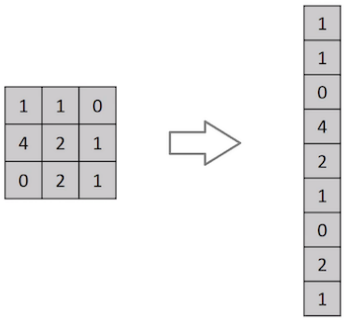
Τα CNNs λαμβάνουν μια βιολογική έμπνευση από τον οπτικό φλοιό[3]. Ο οπτικός φλοιός έχει μικρές περιοχές κυττάρων που είναι ευαίσθητες σε συγκεκριμένες περιοχές του

[3] Οπτικός φλοιός: Η κύρια φλοιώδης περιοχή του εγκεφάλου που λαμβάνει, ενσωματώνει και επεξεργάζεται οπτικές πληροφορίες που μεταδίδονται από τους αμφιβληστροειδείς.

οπτικού πεδίου. Αυτή η ιδέα επεκτάθηκε από ένα συναρπαστικό πείραμα των Hubel και Wiesel το 1962, όπου έδειξαν ότι μερικά μεμονωμένα νευρωνικά κύτταρα στον εγκέφαλο ενός γάτου ανταποκρίθηκαν (ή πυροδότησαν) μόνο με την παρουσία ακμών ενός συγκεκριμένου προσανατολισμού. Για παράδειγμα, μερικοί νευρώνες πυροδοτούν όταν εκτίθενται σε κατακόρυφες ακμές και μερικοί όταν απεικονίζονται οριζόντιες ή διαγώνιες ακμές. Ο Hubel και ο Wiesel διαπίστωσαν ότι όλοι αυτοί οι νευρώνες οργανώθηκαν σε μια αρχιτεκτονική και ότι μαζί μπορούσαν να παράγουν οπτική αντίληψη. Από αυτή την ιδέα κατασκευαστήκαν τα CNN.

Αυτά τα δίκτυα είναι πολύ παρόμοια με τα συνηθισμένα τεχνητά νευρωνικά δίκτυα που έχουν αναλυθεί προηγουμένως αλλά εκπαιδεύονται με έναν αλγόριθμο βαθιάς μάθησης. Με τον όρο βαθιά μάθηση εννοούμε ότι ένα δίκτυο αποτελείται από πολλά περισσότερα επίπεδα/στρώματα. Η άνθιση της βαθιάς μάθησης (Deep Learning) οφείλεται επίσης σε μεγάλο βαθμό στη σημερινή διαθεσιμότητα σε δεδομένα και σε πιο ισχυρούς υπολογιστικούς πόρους. Έχοντας πλέον πληθώρα προεπεξεργασμένων δεδομένων από το διαδίκτυο (big data) και εκμεταλλευόμενοι τον παραλληλισμό σε κάρτες γραφικών για μείωση της ταχύτητας της εκπαίδευσης, έχουμε καταφέρει να κατασκευάσουμε μοντέλα που μπορούν να εξάγουν συμπεράσματα και να ανακαλύψουν υψηλού επιπέδου πρότυπα. Τα Συνελικτικά νευρωνικά δίκτυα δέχονται σαν είσοδο δεδομένα σε μορφή πλέγματος, όπως οι εικόνες, με αποτέλεσμα να μπορούν να κωδικοποιούν διάφορα χαρακτηριστικά τους. Επίσης τα συνελικτικά νευρωνικά δίκτυα βοηθούν στην καλύτερη απόδοση εκπαίδευσης σε συγκεκριμένα προβλήματα όπως η ταξινόμηση μιας εικόνας. Το ερώτημα όμως που προκύπτει είναι το εξής: Μια εικόνα δεν είναι παρά τίποτα άλλο μερικοί πίνακες που περιέχουν πολλούς αριθμούς (pixels). Γιατί λοιπόν να μην μετατραπούν αυτοί οι πίνακες σε ένα μεγάλο διάνυσμα από τιμές και να εισαχθεί σε ένα δίκτυο πολλών στρωμάτων perceptron; (Σχήμα 2.20) Τα τυπικά νευρωνικά δίκτυα δεν μπορούν να χρησιμοποιηθούν σε έτσι είδους προβλήματα (εικόνες) διότι για παράδειγμα ας υποθέσουμε ότι έχουμε μια εικόνα μεγέθους 200x200x3 ως είσοδο (σχετικά μικρή εικόνα), σε ένα πλήρες συνδεδεμένο δίκτυο, ο κάθε νευρώνας στο δεύτερο επίπεδο θα είχε (200x200x3=) 120000 βάρη. Επιπλέον, σε ένα κρυφό επίπεδο δεν θα είχαμε σίγουρα μόνο έναν νευρώνα, και σίγουρα θα χρειαζόμασταν περισσότερα από ένα επίπεδα. Επομένως, οι υπό-μάθηση παράμετροι θα αυξάνονταν δραματικά δημιουργώντας αχανή δίκτυα, μη πρακτικά και

επιρρεπή στην υπέρεκπαίδευση [4]. Αυτό επίσης θα έχει σαν συνέπεια ότι τα συγκεκριμένα δίκτυα θα αγνοούσαν της αυξημένη συσχέτιση που υπάρχει μεταξύ των γειτονικών εικονοστοιχείων[5] σε σχέση με πιο απόμακρα, κάτι το οποίο αποτελεί κύριο χαρακτηριστικό γνώρισμα των φυσικών εικόνων. Γι' αυτό το λόγο, ένα διαφορετικό μοτίβο σχεδίασης έπρεπε να χρησιμοποιηθεί, τα CNN (Saha, 2018). Τα Συνελικτικά Νευρωνικά Δίκτυα λοιπόν μπορούν να μην έχουν πλήρη συνδεσιμότητα με αποτέλεσμα να μειώνονται οι συνδέσεις (βάρη) μεταξύ των νευρώνων σε κάθε επίπεδο και να μην έχουν να αντιμετωπίσουν το πρόβλημα που έχουν τα παραδοσιακά νευρωνικά δίκτυα πολλαπλών στρώσεων. Με άλλα λόγια, η λειτουργία συνέλιξης φέρνει μια λύση στο πρόβλημα αυτό, καθώς μειώνει τον αριθμό των ελεύθερων παραμέτρων, επιτρέποντας στο δίκτυο να είναι βαθύτερο με λιγότερες παραμέτρους.



Σχήμα 2.20: Μετατροπή 2D πίνακα μεγέθους 3x3 σε μονοδιάστατο πίνακα (διάνυσμα) μεγέθους 9x1 σφάλματος.

Πάρθηκε από: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Ένα CN είναι σε θέση να καταγράψει με επιτυχία τις χωρικές και προσωρινές εξαρτήσεις σε μια εικόνα μέσω της εφαρμογής σχετικών φίλτρων. Η αρχιτεκτονική έχει καλύτερη προσαρμογή στο σύνολο δεδομένων λόγω της μείωσης του αριθμού των παραμέτρων που εμπλέκονται και της επαναχρησιμοποίησης των βαρών. Με άλλα λόγια, το δίκτυο μπορεί να εκπαιδευτεί ώστε να κατανοήσει καλύτερα την

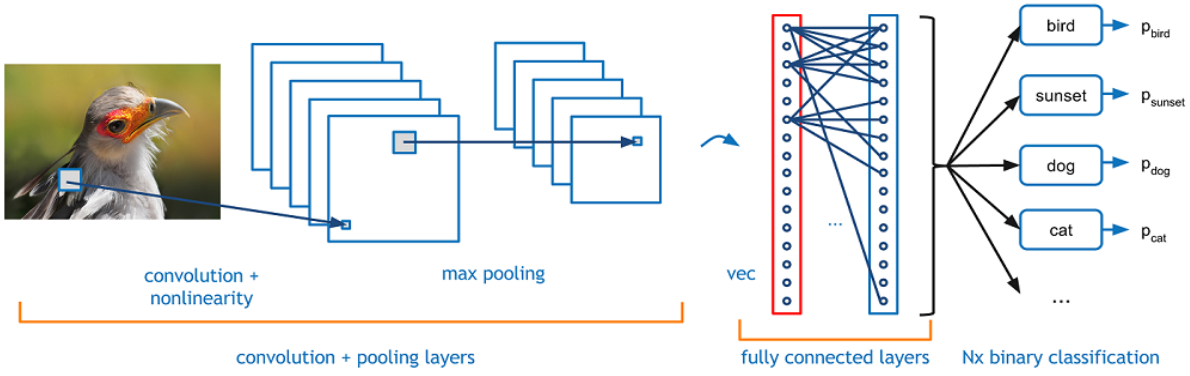
[4] Υπέρεκπαίδευση (overtraining- overfitting) : Συμβαίνει όταν ένα μοντέλο μαθαίνει τις λεπτομέρειες και το θόρυβο στα δεδομένα εκπαίδευσης, στο βαθμό που επηρεάζει αρνητικά την απόδοση του μοντέλου σε νέα δεδομένα. Προκύπτει όταν το σφάλμα εκπαίδευσης μειώνεται ενώ αντίθετα το σφάλμα επαλήθευσης αυξάνεται.

[5] Εικονοστοιχείο (pixel) : ένα σημείο μιας εικόνας που παρουσιάζονται στην οθόνη κάποιου υπολογιστή – το μικρότερο πλήρες δείγμα μιας εικόνα.

εκλεπτυσμένη εικόνα πράγμα πολύ δύσκολο για ένα δίκτυο πολλαπλών στρωμάτων Perceptron.

2.2.8.2 Αρχιτεκτονική

Τα Συνελικτικά νευρωνικά δίκτυα αποτελούν μια κατηγορία εξέλιξης των τυπικών νευρωνικών δικτύων λαμβάνοντας σαν είσοδο μια εικόνα. Γενικά, τα δεδομένα εισόδου σε ένα CNN συσχετίζονται μεταξύ τους όπως μια 3D εικόνα εστιάζοντας σε αυτά και εξάγοντας συγκεκριμένα χαρακτηριστικά τους. Αφού η είσοδος είναι ένα τρισδιάστατο διάνυσμα (πλάτος-ύψος-κανάλια), μπορούμε να οργανώσουμε τους νευρώνες σε 3 διαστάσεις. Τα βασικά επίπεδα που τα απαρτίζουν είναι το συνελικτικό επίπεδο, που εφαρμόζει τη μαθηματική πράξη της συνέλιξης από την οποία προκύπτει και το όνομά του, το επίπεδο συγκέντρωσης (pooling) και ένα πλήρως διασυνδεδεμένο επίπεδο (δίκτυο πολυστρωμάτων perceptron). Το κάθε επίπεδο από τα προαναφερόμενα έχει μία συγκεκριμένη λειτουργία και επομένως η σειρά τοποθέτησης και οργάνωσής τους πρέπει να ακολουθεί συγκεκριμένα μοτίβα (layer patterns). Η βάση κάθε συνελικτικού νευρωνικού δικτύου είναι μία αλληλουχία από συνελικτικά και πλήρως συνδεδεμένα επίπεδα, όπου τα συνελικτικά τοποθετούνται στην αρχή λόγω της τοπικότητας των ιδιοτήτων τους, ενώ τα πλήρως συνδεδεμένα τοποθετούνται συνήθως στο τέλος, καθώς δεν «αντιλαμβάνονται» κάποια τοπικότητα. Και τα δύο αυτά επίπεδα μπορεί να ακολουθούνται από επίπεδα συγκέντρωσης, τα οποία συνοψίζουν και μειώνουν τις χωρικές διαστάσεις των χαρακτηριστικών. Γενικά ένα συνελικτικό δίκτυο αποτελείται από πολλά επίπεδα τα οποία θα αναλυθούν ξεχωριστά στην συνέχεια.



Σχήμα 2.21: Η αρχιτεκτονική ενός συνελικτικού δικτύου για την αναγνώριση εικόνων που περιέχουν ένα ζώο. Αποτελείται από το επίπεδο εισόδου το οποίο είναι η

τρισδιάστατη μορφή της εικόνας, συνελικτικά στρώματα και επίπεδα συγκέντρωσης (convolution + max pooling) και στο τέλος ένα πλήρες συνδεδεμένο δίκτυο (Fully-Connected Neural Network) με ένα επίπεδο εξόδου το οποίο αποτελείται από νευρώνες όπου ο κάθε νευρώνας αντιστοιχεί σε μια κατηγορία ζώου.

Πάρθηκε από: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Επίπεδο Εισόδου

Όταν εισάγουμε σε ένα υπολογιστή μια εικόνα, αυτός θα δεί μια σειρά από τιμές εικονοστοιχείων οι οποίες αποτελούν την συγκεκριμένη εικόνα. Ανάλογα με την ανάλυση και το μέγεθος της εικόνας, θα δει ένα πίνακα από αριθμούς με μέγεθος πλάτος x ύψος x αριθμός καναλιών. Συγκεκριμένα ο αριθμός καναλιών μπορεί να είναι ίσος με το 3 το οποίο αντιπροσωπεύει τις έγχρωμες εικόνες (3 διότι οι έγχρωμές εικόνες αποτελούνται από 3 χρώματα – RGB-red, green, blue - κόκκινο, πράσινο και μπλε). Κάθε ένας από αυτούς τους αριθμούς (πλάτος, ύψος) έχει μια τιμή από 0 έως 255 που περιγράφει την ένταση[6] των εικονοστοιχείων στην συγκεκριμένη εικόνα. Αυτοί οι αριθμοί, αν και δεν έχουν νόημα για εμάς όταν εκτελούμε ταξινόμηση εικόνων, είναι οι μόνες είσοδοι που είναι διαθέσιμες και κατανοητές από τον υπολογιστή. Η ιδέα είναι ότι δίνετε στον υπολογιστή αυτή τη σειρά αριθμών και θα εξάγει αριθμούς που περιγράφουν την πιθανότητα της εικόνας να είναι μια ορισμένη κλάση (0,80 για γάτα, 0,15 για σκύλο, 0,05 για πουλί, κ.λπ.).

Τώρα που γνωρίζουμε τις εισροές καθώς και τα αποτελέσματα, ας σκεφτούμε πώς να το προσεγγίσουμε. Αυτό που θέλουμε να κάνει ο υπολογιστής είναι να είναι σε θέση να διαφοροποιήσει όλες τις εικόνες που του δίνονται και να καταλάβει τα μοναδικά χαρακτηριστικά που κάνουν ένα σκυλί να είναι ένας σκύλος ή που κάνουν μια γάτα να είναι μια γάτα. Αυτή είναι η διαδικασία που συμβαίνει στο μυαλό μας και υποσυνείδητα. Όταν κοιτάζουμε μια εικόνα ενός σκύλου, μπορούμε να το ταξινομήσουμε ως σκύλο, αν η εικόνα έχει αναγνωρίσιμα χαρακτηριστικά όπως αυτιά ή 4 πόδια. Με παρόμοιο τρόπο, ο υπολογιστής είναι σε θέση να εκτελέσει ταξινόμηση

[6] Ένταση: πόσες φορές υπάρχει η συγκεκριμένη τιμή του εικονοστοιχείου στην εικόνα (συχνότητα εμφάνισης)

εικόνων αναζητώντας χαρακτηριστικά χαμηλού επιπέδου, όπως άκρες και καμπύλες, και στη συνέχεια να δημιουργήσει πιο αφηρημένες έννοιες μέσω μιας σειράς συνθετικών στρώσεων. Αυτή είναι μια γενική επισκόπηση του τι κάνει το CNN και συγκεκριμένα στα συνελικτικά επίπεδα.

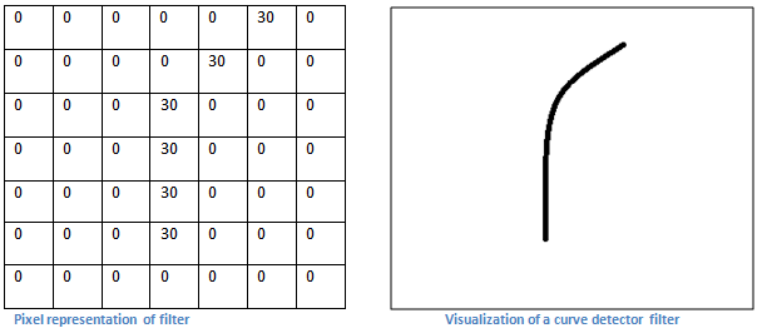
Συνελικτικό Επίπεδο

Ο στόχος της λειτουργίας της συνέλιξης είναι η εξαγωγή των χαρακτηριστικών υψηλού επιπέδου, όπως οι άκρες, από την εικόνα εισόδου. Τα ConvNets δεν χρειάζεται να περιορίζονται σε μόνο ένα συνελικτικό επίπεδο. Το πρώτο επίπεδο (ConvLayer) είναι υπεύθυνο για τη λήψη των χαρακτηριστικών χαμηλού επιπέδου, όπως οι άκρες, το χρώμα, ο προσανατολισμός κλίσης κλπ. Με επιπρόσθετα επίπεδα, η αρχιτεκτονική προσαρμόζεται επίσης σε χαρακτηριστικά υψηλού επιπέδου, δίνοντάς στο δίκτυο την δυνατότητα να κατανόει μια εικόνα όπως ένας άνθρωπος. Τώρα, ο καλύτερος τρόπος για να εξηγήσουμε τι γίνεται σε ένα συνελικτικό στρώμα είναι να φανταστούμε έναν φακό που λάμπει πάνω από την πάνω αριστερή πλευρά της εικόνας. Ας πούμε ότι ο φακός αυτός καλύπτει με το φως του μια περιοχή 5 x 5. Και τώρα, ας φανταστούμε ότι ο φακός αυτός ολισθαίνει σε όλες τις περιοχές της εικόνας εισόδου. Σε όρους μηχανικής μάθησης, αυτός ο φακός ονομάζεται φίλτρο (ή μερικές φορές αναφέρεται ως νευρώνας ή πυρήνας) και η περιοχή που λάμπει πάνω καλείται πεδίο υποδοχής ή οπτικό πεδίο. Σημαντικό να τονιστεί ότι το μέγεθος του φίλτρου πρέπει να είναι μικρότερο από το μέγεθος της εικόνας εισόδου. Τώρα αυτό το φίλτρο είναι επίσης μια σειρά αριθμών οι οποίοι αντιστοιχούν στα γνωστά βάρη. Μια πολύ σημαντική σημείωση είναι ότι το βάθος αυτού του φίλτρου πρέπει να είναι ίδιο με το βάθος της εισόδου, έτσι ώστε οι διαστάσεις αυτού του φίλτρου να είναι 5 x 5 x 3 και να συνάδει με τις διαστάσεις της εισόδου. Η πρώτη θέση που θα έχει το φίλτρο είναι η πάνω αριστερή γωνία της εικόνας. Καθώς το φίλτρο ολισθαίνει ή περιστρέφεται γύρω από την εικόνα εισόδου, πολλαπλασιάζει τις τιμές του φίλτρου με τις αρχικές τιμές εικονοστοιχείων της εικόνας. Αυτοί οι πολλαπλασιασμοί (dot product) συνοψίζονται σε έναν αριθμό για την συγκεκριμένη περιοχή που κάλυψε το φίλτρο. Τώρα επαναλαμβάνουμε αυτή τη διαδικασία για κάθε θέση παράγοντας έναν αριθμό για κάθε θέση. Αφού το φίλτρο περάσει από όλες τις θέσεις, θα διαπιστώσετε ότι θα παραχθεί ένας πίνακας αριθμών με

διαστάσεις 28 x 28 x 1, τον οποίο ονομάζουμε χάρτη ενεργοποίησης ή χάρτη χαρακτηριστικών (activation/feature map). Ο λόγος που ο πίνακας έχει διαστάσεις 28 x 28 είναι ότι υπάρχουν 784 διαφορετικές θέσεις που ένα φίλτρο 5 x 5 μπορεί να χωρέσει σε μια εικόνα εισόδου 32 x 32 (τύπος =μέγεθος εικόνας – μέγεθος φίλτρου+1-32-5+1=28) . Αυτοί οι αριθμοί 784 χαρτογραφούνται σε μια διάταξη 28 x 28 (28*28=784). Σε αυτό όλο το παράδειγμα υποθέσαμε ότι υπάρχει μόνο ένα φίλτρο το οποίο μετακινείται στην εικόνα. Ένα συνελικτικό επίπεδο όμως μπορεί να υποστηρίζει περισσότερα από 1 φίλτρα. Για παράδειγμα αν είχαμε 10 φίλτρα να κινούνται παράλληλα σε μια εικόνα θα παράγονταν 10 διαφορετικοί χάρτες χαρακτηριστικών με διαστάσεις 28 x 28.

Με αυτό τον τρόπο δουλεύουν τα συνελικτικά επίπεδα. Ουσιαστικά είναι αυτά που εντοπίζουν διάφορα σχέδια (patterns) και συγκεκριμένα τα φίλτρα τα όποια χρησιμοποιούνται για να γινεί η συνέλιξη. Πριν προχωρήσουμε σε πιο αναλυτική επεξήγηση σημαντικό να τονίσουμε την έννοια της τοπικής συνδεσιμότητας. Όταν ασχολούμαστε με εισόδους πολλών διαστάσεων δεν είναι πρακτικό να συνδεθούν όλοι οι νευρώνες του επιπέδου με όλους του προηγούμενου, αφού θα προκύψει πολύ μεγάλος αριθμός παραμέτρων. Για το λόγο αυτό συνδέουμε κάθε νευρώνα με μια περιοχή της εισόδου κι έτσι, προκύπτει μια νέα παράμετρος που ονομάζεται οπτικό πεδίο νευρώνων F και καθορίζει πόσα εικονοστοιχεία βλέπει ο κάθε νευρώνας. Είναι, δηλαδή, η διάσταση των φίλτρων του δικτύου και όσο μεγαλύτερη είναι η είσοδος, τόσο μεγαλύτερο είναι συνήθως το οπτικό πεδίο. Για να κατανοήσουμε περισσότερα αυτή την πράξη ας υποθέσουμε ότι έχουμε μια 32 x 32 x 3 εικόνα εισόδου που απεικονίζεται ένα ποντίκι και ένα φίλτρο μεγέθους 7 x 7 το οποίο είναι ανιχνευτής καμπύλης. Ας υποθέσουμε επίσης ότι δεν υπάρχει μια τρίτη διάσταση στα μέγεθη του φίλτρου και της εικόνας εισόδου, για καλύτερη κατανόηση, και ότι το φίλτρο μετακινείται στην εικόνα ανά ένα εικονοστοιχείο. Ως ανιχνευτής καμπύλης, το φίλτρο θα έχει μια δομή όπως φαίνεται στο σχήμα πιο κάτω όπου υπάρχουν ψηλές αριθμητικές τιμές κατά μήκος της περιοχής που είναι το σχήμα μιας καμπύλης.

Ξεκινώντας την διαδικασία της συνέλιξης πρώτα απ’ όλα τοποθετούμε το φίλτρο στην αριστερή πάνω γωνία της εικόνας που θέλουμε να ανιχνεύσουμε καμπύλες.

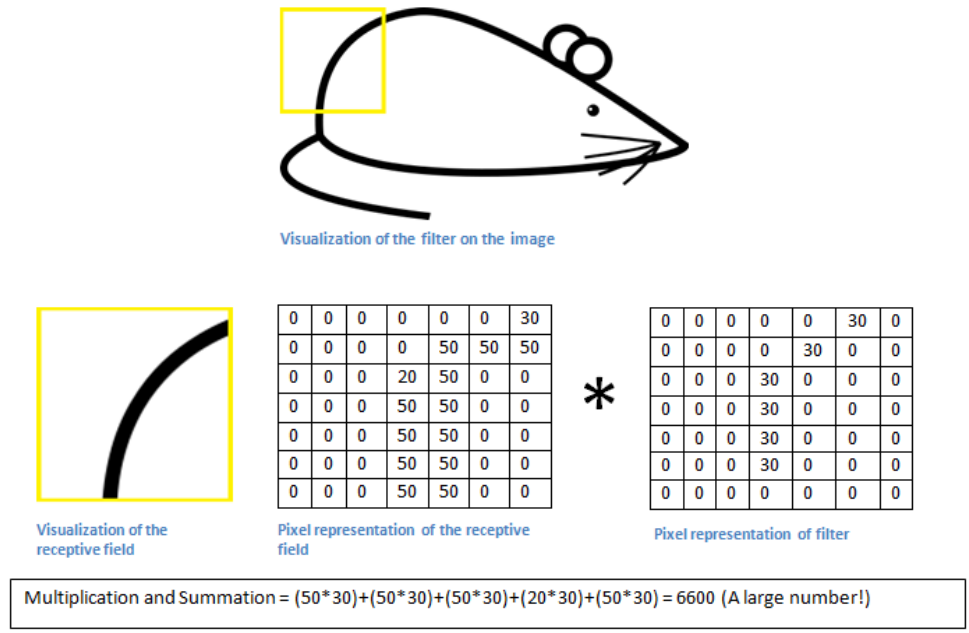


Σχήμα 2.22 Φίλτρο το οποίο είναι ανιχνευτής καμπύλης.

Αριστερά: Αναπαράσταση φίλτρου σε εικονοστοιχεία-τιμές, Δεξιά: Αναπαράσταση φίλτρου σε εικόνα

Πάρθηκε από: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Υπολογίζεται ο πολλαπλασιασμός μεταξύ των τιμών του φίλτρου με τις τιμές, ή αλλιώς τα λεγόμενα εικονοστοιχεία, της εικόνας παράγοντας μια τιμή.

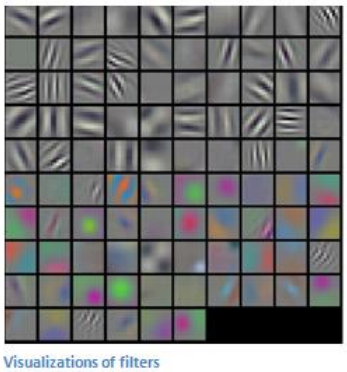


Σχήμα 2.23 Πράξη συνέλιξης στην αριστερή πάνω περιοχή της εικόνας εισόδου. Πάνω: Εικόνα εισόδου για ανίχνευση καμπύλων, Κάτω Αριστερά: Αριστερή πάνω περιοχή της εικόνας που έχει επιλεχθεί για συνέλιξη, Κάτω Δεξιά: Πολλαπλασιασμός της συγκεκριμένης περιοχής (αναπαράσταση σε εικονοστοιχεία) με το φίλτρο ανιχνευτή καμπύλων. Κάτω: Η τιμή μετά την πράξη της συνέλιξης,

Πάρθηκε από: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Όπως φαίνεται στο σχήμα πιο πάνω υπολογίζεται ο πολλαπλασιασμός του κάθε εικονοστοιχείου με την κάθε τιμή του φίλτρου και συναθροίζονται όλα τα αποτελέσματα όλων αυτών των γινομένων και παράγεται μια μεγάλη αριθμητική τιμή η οποία θα εισαχθεί στον χάρτη χαρακτηριστικών για την συγκεκριμένη θέση και για το συγκεκριμένο φίλτρο. Όλη αυτή διαδικασία επαναλαμβάνεται σε κάθε περιοχή της εικόνας γεμίζοντας στον χάρτη χαρακτηριστικών, ο οποίος δεν είναι τίποτα άλλο από ένας πίνακας, με ψηλές και χαμηλές τιμές όπου οι ψηλές αντιπροσωπεύουνε μια ψηλή πιθανότητα εντοπισμού καμπύλης ενώ οι χαμηλές αντιπροσωπεύουν μια χαμηλή πιθανότητα. Με άλλα λόγια, ο χάρτης χαρακτηριστικών θα δείξει τις περιοχές στις οποίες υπάρχουν κατά πάσα πιθανότητα καμπύλες στην εικόνα.

Σε αυτό το παράδειγμα, η άνω αριστερή τιμή του 26 × 26 x 1 χάρτη χαρακτηριστικών (26 λόγω του φίλτρου 7x7 → 32-7+1=26) θα είναι 6600. Αυτή η υψηλή τιμή σημαίνει ότι είναι πιθανό ότι υπάρχει κάποιο είδος καμπύλης στην περιοχή εισόδου που κάλυψε το συγκεκριμένο φίλτρο. Να τονιστεί ότι όλη αυτή η διαδικασία του συγκεκριμένου παραδείγματος είναι για μόνο ένα φίλτρο. Αυτό είναι μόνο ένα φίλτρο που πρόκειται να ανιχνεύσει γραμμές που καμπυλώνουν προς τα δεξιά (Σχήμα 2.22). Μπορούμε να έχουμε άλλα φίλτρα για γραμμές που καμπυλώνουν προς τα αριστερά ή για ευθεία άκρα. Όσο περισσότερα φίλτρα, τόσο μεγαλύτερο είναι το βάθος του χάρτη χαρακτηριστικών και οι περισσότερες πληροφορίες που θα έχουμε για την εικόνα εισόδου.



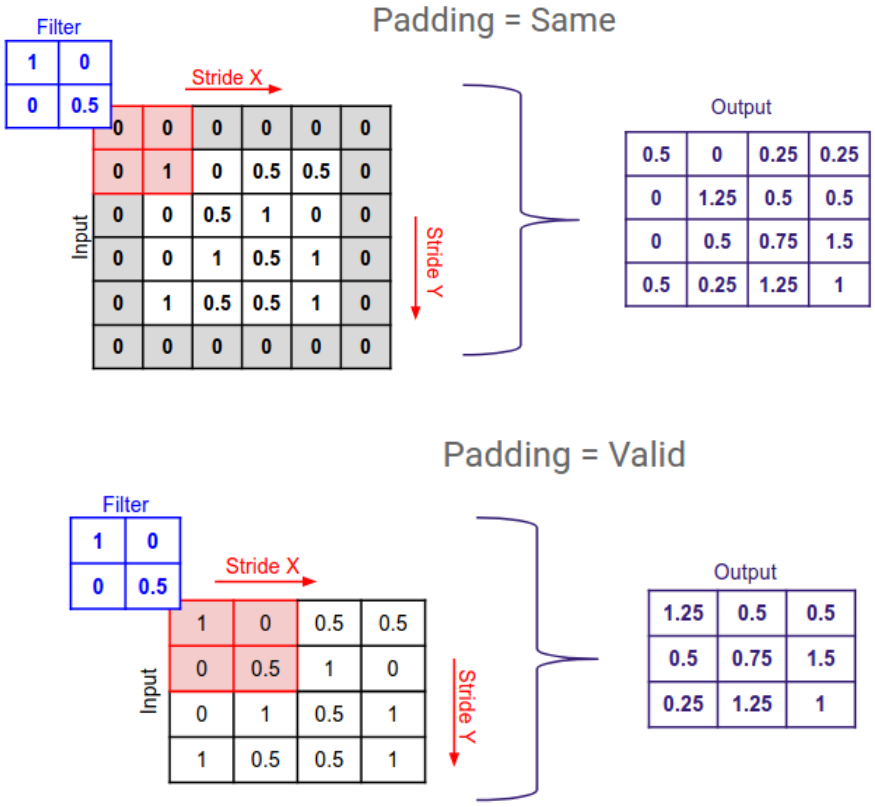
Σχήμα 2.24 Διάφορα πραγματικά φίλτρα μετά το πρώτο συνελικτικό επίπεδο από το Stanford's CS 231N course από τους Andrej Karpathy και Justin Johnson. Στα πάνω

φίλτρα παρατηρούνται ακμές σε διάφορες κατευθύνσεις και στη μέση διάφορα χρώματα σε σχήμα κύκλου.

Πάρθηκε από: http://cs231n.stanford.edu/

Όταν περάσουμε από ένα άλλο συνελικτικό στρώμα, η έξοδος του πρώτου συνελικτικού στρώματος γίνεται η είσοδος του 2ου συνελικτικού στρώματος και συγκεκριμένα η είσοδος είναι οι χάρτες ενεργοποίησης που προκύπτουν από το πρώτο στρώμα. Έτσι, κάθε στρώμα της εισόδου περιγράφει βασικά τις θέσεις στην αρχική εικόνα για το πού εμφανίζονται ορισμένα χαρακτηριστικά χαμηλού επιπέδου. Όταν εφαρμόζονται περισσότερα φίλτρα προκύπτουν χαρακτηριστικά υψηλότερου επιπέδου. Οι τύποι αυτών των χαρακτηριστικών θα μπορούσαν να είναι ημικυκλικά (συνδυασμός καμπύλης και ευθύγραμμης ακμής) ή τετράγωνα (συνδυασμός διαφόρων ευθύγραμμων άκρων). Καθώς περνάμε από περισσότερα συνελικτικά επίπεδα, όλο και πιο περίπλοκα χαρακτηριστικά εντοπίζονται.

Εκτός από το μέγεθος του φίλτρου και πόσα παράλληλα φίλτρα θα υπάρχουν σε ένα συνελικτικό επίπεδο, 2 ακόμα υπέρ-παράμετροι χρειάζεται να οριστούν στο συνελικτικό επίπεδο προκειμένου να καθοριστεί η διαδικασία σάρωσης (πράξη συνέλιξης) της εισόδου. Οι παράμετροι αυτές είναι το βήμα (stride) και το γέμισμα (padding). Το stride είναι η υπέρ-παράμετρος που καθορίζει πόσο πυκνή θα είναι η δειγματοληψία της εισόδου. Με άλλα λόγια, το stride καθορίζει πόσα εικονοστοιχεία θα μετακινείται το φίλτρο πάνω στην εικόνα οριζοντίως και καθέτως. Θέτοντας το stride ίσο με 1, το φίλτρο θα μετακινείται κατά ένα εικονοστοιχείο οδηγώντας σε πυκνή σάρωση της εισόδου ενώ μεγαλύτερα stride θα προκαλούν πιο αραιή σάρωση της εισόδου. Με την χρήση των strides δημιουργούνται πιο μικροί πίνακες όπως οι πίνακες χαρτογράφησης (χάρτης χαρακτηριστικών). Το Padding είναι η υπέρ-παράμετρος που χρησιμοποιείται για να γεμίσει με μηδενικά το περίγραμμα της εισόδου. Υπάρχουν 2 είδη padding: το Same padding (zero padding) και το Valid Padding (without padding). Το Same padding ή αλλιώς, το γέμισμα με μηδενικά (zero- padding) μπορεί να βοηθήσει σημαντικά εάν θέλουμε να διατηρήσουμε τις χωρικές διαστάσεις της εισόδου και να εφαρμόσουμε ένα συγκεκριμένο μέγεθος φίλτρου για την ομαλή σάρωση των χαρακτηριστικών. Το Valid padding δεν προσθέτει ή γεμίζει τίποτα στην αρχική εικόνα με αποτέλεσμα να έχουμε ένα πίνακα με μικρότερες διαστάσεις (Σχήμα 2.25).



Σχήμα 2.25 Διαδικασία συνέλιξης με valid και same padding. Πάνω: Same Padding - παρατηρείται η εισαγωγή μηδενικών τιμών γύρω από την εικόνα εισόδου με πλαίσιο χρώματος γκρι (από 4x4 → 6x6 διαστάσεις της εικόνας εισόδου) με stride=1 και η έξοδος θα έχει τις ίδιες διαστάσεις με τις πραγματικές διαστάσεις της εικόνας εισόδου, δηλαδή 4x4. Κάτω: Valid Padding - παρατηρείται η μείωση διαστάσεων μετά την διαδικασία συνέλιξης με stride=1 και χωρίς padding, δηλαδή η έξοδος θα έχει διαστάσεις μεγέθους 3x3.

Πάρθηκε από: https://labs.bawi.io/deep-learning-convolutional-neural-networks-7992985c9c7b

Συνοπτικά το επίπεδο συνέλιξης (convolutional layer) χαρακτηρίζεται από τα εξής στοιχεία:

- Είσοδο: ένας πίνακας διαστάσεων πλάτος (W_1) x ύψος (H_1) x αριθμός καναλιών (D_1)
- Παράμετροι:
  - Αριθμός φίλτρων (K)
  - Μέγεθος του κάθε φίλτρου (F)
  - Το βήμα μετατόπισης των φίλτρων - stride (S)

- Το ποσό γεμίσματος με μηδενικά κατά το ύψος και πλάτος της εικόνας εισόδου - padding (P)

- Έξοδος:
  - Χάρτη χαρακτηριστικών του κάθε φίλτρου W_2 x H_2 όπου:

$$W_2 = \left(\frac{W_1 - F + 2P}{S} \right) + 1$$

$$H_2 = \left(\frac{H_1 - F + 2P}{S} \right) + 1$$

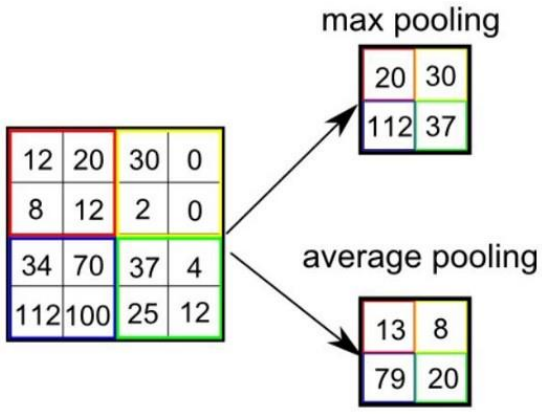
  - Συνολικοί χάρτες χαρακτηριστικών: W_2 x H_2 x D_2 όπου D_2 : K (K: αριθμός φίλτρων)
  - Βάρη του κάθε φίλτρου με διαστάσεις F x F x D_1 +1
  - Συνολικά βάρη: F x F x D_1 x K.

Επίπεδο Συγκέντρωσης

Ανάμεσα από τα συνελικτικά επίπεδα πραγματοποιείται μια διαδικασία χωρικής υποδειγματολειψίας η οποία δεν είναι πάντα αναγκαία. Συγκεκριμένα σε κάθε χάρτη χαρακτηριστικών εφαρμόζεται μια υποδειγματοληψία η οποία είναι υπεύθυνη στην μείωση των διαστάσεων αυτών των παραγόμενων χαρτών. Αυτό έχει σαν αποτέλεσμα την μείωση των αριθμών των παραμέτρων και των υπολογισμών στο δίκτυο, αφού μειώνεται το μέγεθος των feature maps και συνεπώς μειώνεται και η πιθανότητα εμφάνισης του προβλήματος της υπέρεκπαίδευσης. Η λειτουργία των επίπεδων χωρικής υποδειγματολειψίας βασίζεται στις ίδιες αρχές που διέπουν και τα συνελικτικά επίπεδα. Αυτό σημαίνει πως έχουν υπέρ-παραμέτρους για να δηλώσουν το μέγεθος του φίλτρου τους (F1 x F2), του βήματος σάρωσης -stride (S), και του γεμίσματος με μηδενικά - padding (P). Τα επίπεδα αυτά λειτουργούν ανεξάρτητα σε κάθε βαθμίδα του βάθους της εισόδου και την μειώνουν χωρικά, κάνοντας χρήση μιας συγκεκριμένης συνάρτησης χωρικής υποδειγματολειψίας. Ιστορικά ήταν κοινώς αποδεκτό να χρησιμοποιούνται αυτά τα επίπεδα μεταξύ διαδοχικών συνελικτικών επιπέδων με υπέρ-παραμέτρους της μορφής: μέγεθος πυρήνα 2x2 και βήματος 2, οι οποίες οδηγούν στην αφαίρεση του 75% του συνολικού διανυσματικού χώρου των χαρτών των χαρακτηριστικών των

δειγμάτων που εμφανίζονται στην είσοδο. Αυτή η επιθετική μείωση είναι χρήσιμη επειδή μειώνει το overfit στην ταξινόμηση μικρών σετ δεδομένων με έντονες διαφορές μεταξύ των κατηγοριών τους. Σε προβλήματα όμως, όπου οι κατηγορίες είναι αρκετά περισσότερες, η χρήση των επιπέδων χωρικής υπό-δειγματοληψίας μπορεί να καταστήσει αδύνατη την διακριτοποίηση των κατηγοριών στα τελευταία επίπεδα και έτσι συστήνεται να γινεί χρήση μεγαλύτερων μεγεθών stride για να μειωθούν οι διαστάσεις των χαρτών χαρακτηριστικών.

Πολλές συναρτήσεις έχουν δημιουργηθεί για την επίτευξη της χωρικής υπό-δειγματοληψίας όπως το max pooling και το average pooling. Το Max Pooling επιστρέφει τη μέγιστη τιμή από την περιοχή της εικόνας που καλύπτεται από το φίλτρο, ενώ η μέση (average) συγκέντρωση επιστρέφει τον μέσο όρο όλων των τιμών από την περιοχή της εικόνας που καλύπτεται από το φίλτρο. Η πιο συνηθισμένη και αποδοτικότερη συνάρτηση είναι η max pooling διότι επιλέγει την πιο μεγάλη τιμή από κάθε τμήμα που περνά από την εικόνα που αυτό σημαίνει ότι επιλέγει στην ουσία την πιο δυνατή ενεργοποίηση και συνεπώς δεν λαμβάνει υπόψη οποιοδήποτε θόρυβο και περιττές πληροφορίες που υπάρχουν την εικόνα εισόδου, αλλά μόνο τα πιο δυνατά χαρακτηριστικά.



Σχήμα 2.26 Παράδειγμα εφαρμογής max και average pooling σε μια εικόνα διαστάσεων 4x4. Πάνω πίνακας: εφαρμογή max pooling για παράδειγμα το αριστερό πάνω τμήμα (με χρώμα πλαισίου κόκκινο) η μέγιστη τιμή είναι 20. Κάτω πίνακας: εφαρμογή average pooling για παράδειγμα το αριστερό πάνω τμήμα (με χρώμα πλαισίου κόκκινο) η μέση τιμή είναι (12+20+8+12) / 4 = 13.

Πάρθηκε από: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

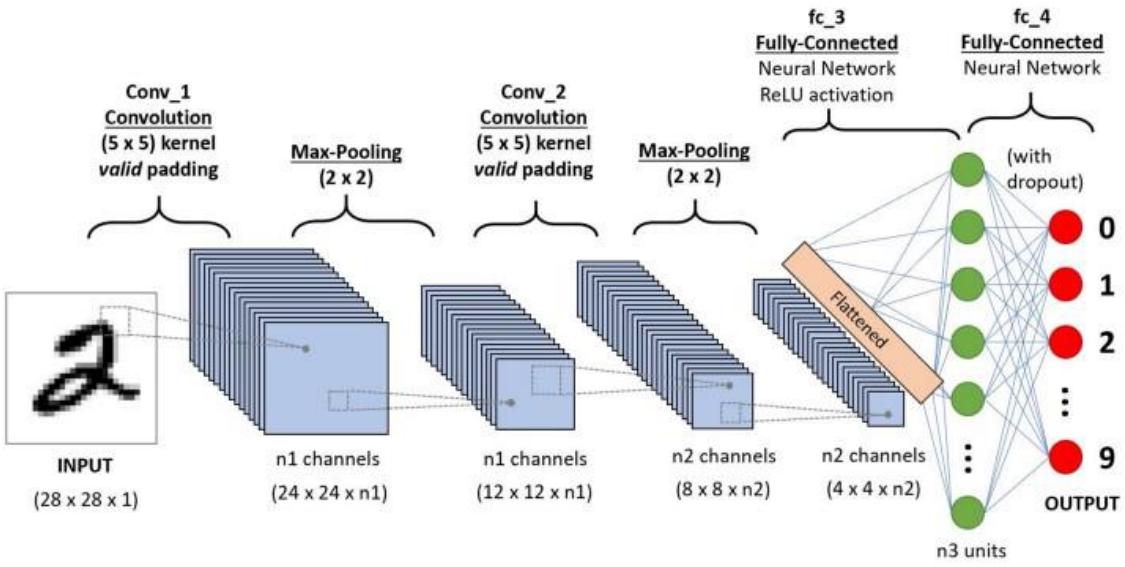
Πλήρως συνδεδεμένο επίπεδο

Το συνελικτικό επίπεδο με το επίπεδο συγκέντρωσης αποτελούν τα πιο σημαντικά κομμάτια σε ένα συνελικτικό νευρωνικό δίκτυο αφού αυτά εξάγουν τα χαρακτηριστικά μιας εικόνας. Όμως για να ολοκληρωθεί η αρχιτεκτονική αυτού του δικτύου και όπως προαναφέρθηκε, πρέπει να υπάρχει ένα επίπεδο εξόδου το οποίο μας δίνει διάφορες πιθανότητες σε ποια κατηγορία ανήκει η συγκεκριμένη εικόνα ή γενικότερα τις κατηγορίες που δηλώσαμε εμείς. Άρα πρέπει να συνδεθεί το τελευταίο επίπεδο των πιο πάνω επίπεδων και συγκεκριμένα το τελευταίο επίπεδο συγκέντρωσης με ένα πλήρες τυπικό νευρωνικό επίπεδο.

Συγκεκριμένα υπάρχει ένα πλήρες συνδεδεμένο στρώμα το οποίο παίρνει σαν είσοδο την έξοδο του τελευταίου επιπέδου συγκέντρωσης και εξάγει ένα διάνυσμα μεγέθους N όπου N είναι ο αριθμός κατηγοριών (classes) που αντιστοιχούν σε ορισμένα χαρακτηριστικά. Για παράδειγμα, εάν το πρόγραμμα προβλέπει ότι κάποια εικόνα είναι πουλί, θα έχει υψηλές τιμές στους χάρτες χαρακτηριστικών που αντιπροσωπεύουν χαρακτηριστικά υψηλού επιπέδου όπως φτερά ή ράμφος κλπ. Σημαντικό είναι ότι, για να γίνει αυτό πρέπει η έξοδος του τελευταίου επιπέδου συγκέντρωσης να μετατραπεί σε ένα διάνυσμα για να μπορεί να παρουσιαστεί σε ένα γνωστό δίκτυο Perceptron πολλαπλών επιπέδων (flattened output). Ο τρόπος με τον οποίο λειτουργεί αυτό το πλήρες συνδεδεμένο στρώμα είναι ως εξής: εξετάζει την έξοδο του προηγούμενου στρώματος, το οποίο αναπαριστά τους χάρτες χαρακτηριστικών υψηλού επιπέδου, και καθορίζει ποια χαρακτηριστικά σχετίζονται περισσότερο με μια συγκεκριμένη κατηγορία. Να τονίσουμε ότι αυτό το δίκτυο λειτουργεί με τον γνωστό τρόπο το προς τα εμπρός και το πίσω πέρασμα, για να εκπαιδευτεί ούτως ώστε να διακρίνει και να επεξεργαστεί τα χαρακτηριστικά που εντοπιστήκαν από τα προηγούμενα επίπεδα.

Ένα χαρακτηριστικό παράδειγμα είναι η κατηγοριοποίηση χειρόγραφων ψηφίων (digit classification), το N θα είναι 10 αφού υπάρχουν 10 ψηφία. Κάθε αριθμός σε αυτό το διάνυσμα N αντιπροσωπεύει την πιθανότητα μιας συγκεκριμένης κατηγορίας. Για παράδειγμα, αν το διάνυσμα που προκύπτει είναι [0,1,1,75 0 0 0 0 0 0,05], τότε αυτό αντιπροσωπεύει πιθανότητα 10% ότι η εικόνα είναι 1, 10% πιθανότητα ότι η εικόνα

είναι 2, 75% πιθανότητα ότι η εικόνα είναι 3 και 5% πιθανότητα η εικόνα να είναι 9 (softmax classification technique)



Σχήμα 2.27: Η αρχιτεκτονική ενός συνελικτικού δικτύου για την αναγνώριση χειρόγραφων αριθμών. Αποτελείται από το επίπεδο εισόδου το οποίο είναι η τρισδιάστατη μορφή μιας μαυρόασπρης εικόνας (INPUT), 2 συνελικτικά στρώματα (Conv_1, Conv_2) όπου ενδιάμεσα από αυτά υπάρχουν 2 επίπεδα συγκέντρωσης (Max pooling) και στο τέλος δυο πλήρες συνδεδεμένα τυπικά επίπεδα (Fully-Connected Neural Network) με ένα επίπεδο εξόδου το οποίο αποτελείται από 10 νευρώνες όπου ο κάθε νευρώνας αντιστοιχεί σε ένα αριθμό (κατηγοριοποίηση).
Πάρθηκε από: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Αφού το πλήρες συνδεδεμένο επίπεδο διακρίνει τα χαρακτηριστικά μιας εικόνας, το τελευταίο πράγμα που πρέπει να κάνει για να ολοκληρώσει το δίκτυο είναι να τα ταξινομήσει στις διάφορες κλάσεις που υπάρχουν στο τελευταίο επίπεδο. Όπως είχαμε δει στο προηγούμενο παράδειγμα η ταξινόμηση των ψηφίων είχε γινεί με ένα συγκεκριμένο τρόπο. Ο τρόπος αυτός ονομάζεται Softmax Classification technique. Αυτή η τεχνική είναι μια μορφή λογικής παλινδρόμησης που ομαλοποιεί μια τιμή εισόδου σε ένα διάνυσμα τιμών που ακολουθεί μια κατανομή πιθανότητας της οποίας το συνολικό άθροισμα είναι μέχρι 1 και έχουν σαν πεδίο τιμών [0,1].

Έχουν εφευρεθεί διάφορες αρχιτεκτονικές των συνελικτικών νευρωνικών δικτύων οι οποίες είναι οι εξής:

- LeNet-5 (1998): Πρώτη επιτυχημένη εφαρμογή από τον Yann LeCun. Διάσημη για την ανάγνωση ταχυδρομικών κωδικών και ψηφίων.
- AlexNet (2012): Πρώτο έργο που προώθησε τα CNN στο Computer Vision από τους Alex Krizhevsky, Ilya Sutskever και Geoff Hinton.
- ZFNet(2013): Μια εξέλιξη των AlexNet από τους Matthew Zeiler και Rob Fergus.
- GoogleNet/Inception(2014): Ανάπτυξη μιας Μονάδας Έναρξης που μείωσε δραματικά τον αριθμό των παραμέτρων στο δίκτυο.
- VGGNet (2014): Η κύρια συμβολή της Karen Simonyan και του Andrew Zisserman ήταν να δείξουν ότι το βάθος του δικτύου είναι ένα κρίσιμο στοιχείο για καλές επιδόσεις.
- ResNet(2015): Ειδικές skip συνδέσεις και βαριά χρήση της κανονικοποίησης παρτίδων από τον Kaiming He et al. Χάρη σε αυτή την τεχνική ήταν σε θέση να εκπαιδεύσουν ένα NN με 152 στρώματα ενώ εξακολουθούν να έχουν μικρότερη πολυπλοκότητα από το VGGNet. Επιτυγχάνει ένα ποσοστό σφάλματος 5,57% που υπερβαίνει την απόδοση σε επίπεδο ανθρώπου.

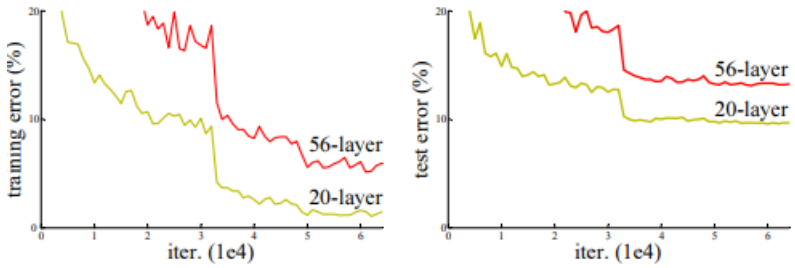
Κάθε μία από αυτές τις αρχιτεκτονικές δικτύου έχει μοναδική προσέγγιση σε διάφορα προβλήματα. Για παράδειγμα, το AlexNet έχει παράλληλες δύο σειρές CNN που εκπαιδεύονται σε δύο GPUs με διασυνδέσεις, το GoogleNet έχει αρχικές ενότητες, το ResNet έχει υπολειπόμενες συνδέσεις οι οποίες αντιμετωπίζουν προβλήματα βαθιάς μάθησης, η οποία αρχιτεκτονική θα αναλυθεί στην επόμενη υπό ενότητα.

2.2.9 Υπολειπόμενα Νευρωνικά Δίκτυα

2.2.9.1 Γενικά

Τα τελευταία χρόνια, οι τομείς εφαρμογής των βαθιών νευρωνικών δικτύων επεκτάθηκαν πολύ γρήγορα. Δεν είναι καθόλου παράξενο το γεγονός ότι κάθε χρόνο εμφανίζεται μια σημαντική βελτίωση στον τομέα αυτό. Επιπλέον, οι τομείς εφαρμογής

της βαθιάς μάθησης γίνονται όλο και πιο ευρείς λόγω της διαθεσιμότητας μεγάλων συνόλων δεδομένων και ισχυρών μονάδων GPU που έχουν επιτρέψει την εκπαίδευση πολύ βαθιών αρχιτεκτονικών. Αυτό έχει ως αποτέλεσμα ότι μεγάλοι παίκτες όπως το Google, το Facebook, η Microsoft διοργάνωσαν ομάδες για να μελετήσουν την βαθιά μάθηση. Συγκεκριμένα ο Simonyan και οι συγγραφείς του VGG απέδειξαν ότι με την απλή στοίβαση περισσότερων στρωμάτων μπορούσαν να βελτιώσουν την ακρίβεια, αφού και πριν από αυτό το 2009, ο Yoshua Bengio στην μονογραφία του, "Learning Deep Architectures for AI", έδωσε μια δυνατή θεωρητική ανάλυση της αποτελεσματικότητας που είχαν τα βαθιά νευρωνικά δίκτυα. Άρα βάση αυτή της θεωρίας που έχει αναλυθεί, είμαστε σε θέση να πούμε ότι μπορούμε να χτίσουμε πιο ακριβή συστήματα απλά στοιβάζοντας όλο και περισσότερα στρώματα; Σε κάποιο σημείο, η ακρίβεια ναι θα βελτιωνόταν, αλλά πέρα από τα περίπου 25+ στρώματα, η ακρίβεια δυστυχώς μειωνόταν, είχαν τονίσει δραματικά οι ερευνητές της ομάδας Microsfort (Kaiming He et al., 2016a) (Σχήμα 2.28).



Σχήμα 2.28: Σύγκριση σφαλμάτων εκπαίδευσης και δοκιμής δυο δικτύων με 20 επίπεδα και 56 στρώματα με το CIFAR-10 dataset. Αριστερά : σφάλμα εκπαίδευσης (training error) , Δεξιά: σφάλμα δοκιμής (testing error). Παρατηρείται ότι το βαθύτερο δίκτυο (56-layer) έχει υψηλότερο σφάλμα εκπαίδευσης και, ως εκ τούτου, σφάλμα δοκιμής από το δίκτυο 20-layer (He et al., 2016a).

Στο τέλος του έτους 2015, η Microsoft Research Asia κυκλοφόρησε ένα έγγραφο με τίτλο "Βαθιά Υπολειπόμμενη Μάθηση για Αναγνώριση Εικόνας", από τους Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. Το έγγραφο αυτό πέτυχε κορυφαία αποτελέσματα στην ταξινόμηση και ανίχνευση εικόνων, κερδίζοντας την 1η θέση στον διαγωνισμό κατάταξης ILSVRC 2015 με ποσοστό σφάλματος 3,57% χρησιμοποιώντας

ένα δίκτυο που είχε 152 στρώματα, δηλαδή ένα 8 φορές μεγαλύτερο δίκτυο από ένα δίκτυο VGG. Επίσης έλαβε την πρώτη θέση στο διαγωνισμό ILSVRC και COCO 2015 στην ανίχνευση ImageNet, εντοπισμό ImageNet, ανίχνευση Coco και τμηματοποίηση Coco.

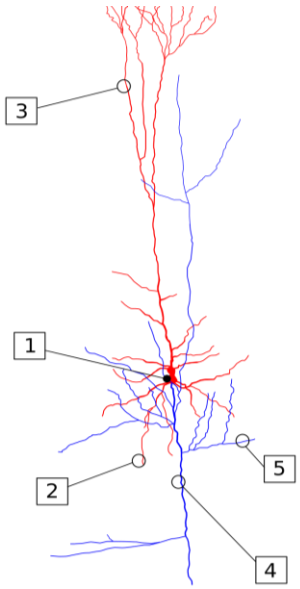
Συγκεκριμένα ο He et al. το 2016 πρότεινε μια αξιοσημείωτη λύση η οποία από τότε επέτρεψε την εκπαίδευση μέχρι και 1000 στρώματα με αυξανόμενη ακρίβεια. Αξιοποιώντας τις ισχυρές δυνατότητες αναπαράστασης αυτής της λύσης, έχει επιταθεί η απόδοση πολλών εφαρμογών ηλεκτρονικής όρασης εκτός από την ταξινόμηση εικόνων, όπως η ανίχνευση αντικειμένων και η αναγνώριση προσώπου. Δεδομένου ότι η Βαθιά Υπολειπόμμενη Μάθηση εξέπληξε αρκετούς ανθρώπους το 2015, πολλοί στην ερευνητική κοινότητα έχουν βυθιστεί στα μυστικά της επιτυχίας της με αποτέλεσμα να υπάρξουν πολλές βελτιώσεις στην αρχιτεκτονική της. Ο ενθουσιασμός και η έκπληξη που υπήρξε δεν ήταν μόνο στους ανθρώπους που ενημερωθήκαν για την καινούργια αυτή λύση αλλά εκπλαγήκαν και οι ίδιοι οι εφευρέτες της. Συγκεκριμένα ένα μέλος της ομάδας της Microsoft και συγγραφής του εγγράφου που κυκλοφόρησε είχε πει το εξής:

"We even didn't believe this single idea could be so significant."
-Jian Sun from Microsoft Research Team

Η κεντρική ιδέα του ίδιου του χαρτιού ήταν απλή και σαφής. Οι ερευνητές της Microsoft δημιούργησαν τα λεγόμενα βαθιά υπολειπόμενα δίκτυα (Deep Residual Networks – ResNets) τα οποία παίρνουν τα συνελικτικά νευρωνικά δίκτυα ένα βήμα παραπέρα με "μικρές" αλλά "σημαντικές" αλλαγές στην αρχιτεκτονική τους. Αναλυτικότερα τα ResNets έχουν παρόμοια αρχιτεκτονική με τα CNNs, αφού και αυτά ανήκουν στην οικογένεια των βαθιών δικτύων, με την διαφορά ότι υπάρχουν επιπρόσθετες συνδέσεις στο δίκτυο. Αυτές οι συνδέσεις ονομάζονται συνδέσεις συντόμευσης, υπολειμματικές συνδέσεις ή συνδέσεις παράκαμψης (skip connections) οι οποίες βοηθούν στο να αντιμετωπίσουν το πρόβλημα vanishing problem (υποκεφάλαιο 2.2.10) με αποτέλεσμα την αύξηση της ακρίβειας του δίκτυού.

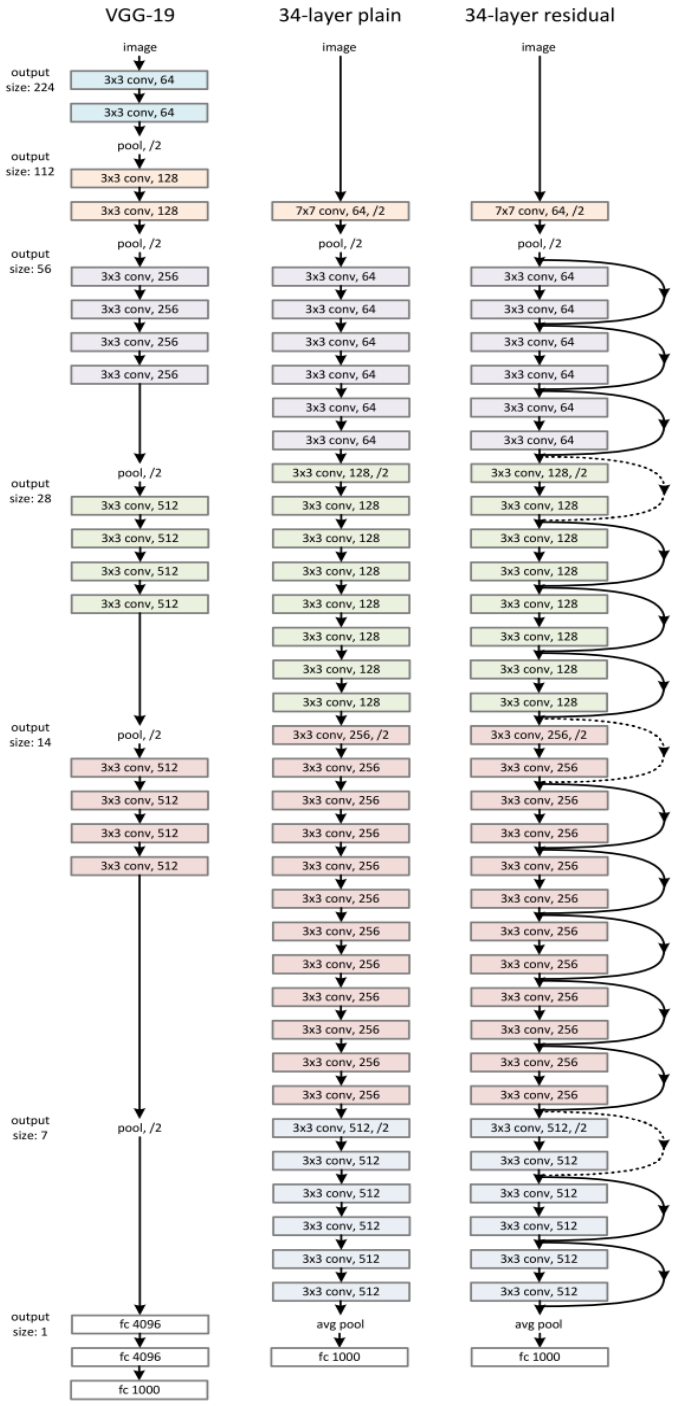
2.2.9.2 Αρχιτεκτονική

Ένα υπολειπόμενό νευρωνικό δίκτυο είναι ένα τεχνητό νευρωνικό δίκτυο που βασίζεται σε γνωστά κατασκευάσματα, συγκεκριμένα είναι μια ανακατασκευή των γνωστών πυραμιδικών κυττάρων του εγκεφαλικού φλοιού. Ορισμένοι νευρώνες του φλοιώδους στρώματος VI εισάγονται από το στρώμα I, παρακάμπτοντας τα ενδιάμεσα στρώματα. Στην εικόνα πιο κάτω συγκρίνονται τα σήματα από τον κορυφαίο δενδρίτη (3) που υπερπηδούν τα στρώματα και από ένα άλλο δενδρίτη (2) οποίος συλλέγει σήματα από το προηγούμενο ή / και το ίδιο στρώμα.



Σχήμα 2.29: Μια ανακατασκευή ενός πυραμιδικού κυττάρου. Το Σώμα και οι δενδρίτες επισημαίνονται σε κόκκινο ενώ άξονες σε μπλε χρώμα. (1) Soma, (2) Βασικός δενδρίτης, (3) Ακδικός δενδρίτης, (4) Axon, (5) Εξάρτημα άξονας.
Πάρθηκε από: https://en.wikipedia.org/wiki/Residual_neural_network

Βάση αυτής της ιδέας τα ResNets χρησιμοποιούν τις λεγόμενες συνδέσεις παράκαμψης η οποίες δίνουν την δυνατότητα της μετάβασης σε επόμενα ή προηγούμενα επίπεδα. Η παράληψη ενός μόνο στρώματος γίνεται στα δίκτυα με όνομα HighwayNets ενώ πολλές παράλληλες παραλήψεις γίνονται στα DenseNets (Huang et al.,2016). Για να μπορούν να υπάρξουν αυτές οι παραλήψεις, ή αλλιώς τα πηδήματα στο δίκτυο πρέπει να υπάρχει ένα μεγαλύτερο βάθος απ' ότι υπάρχει στα τυπικά βαθιά νευρωνικά δίκτυα (Σχήμα 2.30).



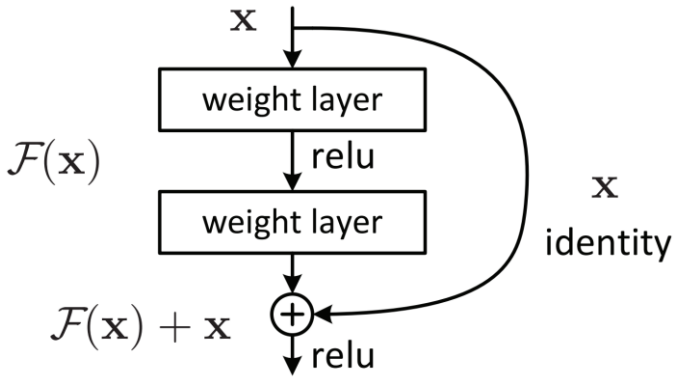
Σχήμα 2.30: Σύγκριση επιπέδων 3 διαφορετικών αρχιτεκτονικών συνελικτικών νευρωνικών δικτύων (κυρίως με διαστάσεις 3x3 του φίλτρου). Παρατηρείται δραματική αύξηση επιπέδων (βάθους) στα ResNets (δεξιά) αφού περιέχει 8 φορές περισσότερα επίπεδα από τα VGG δίκτια (αριστερά) (He et al., 2016a).

Κάθε παράκαμψη δημιουργεί ένα υπολειπόμενο μπλοκ που περιέχει κάποια επίπεδα συνέλιξης τα οποία προβλέπουν ένα υπόλοιπό που προστίθεται στην είσοδο του επόμενου μπλοκ. Για να κατανοήσουμε καλύτερα το τι γίνεται σε ένα ResNet ας αναλύσουμε πρώτα ένα υπολειπόμενο μπλοκ. Η κατανόηση ενός υπολειπόμενου μπλοκ είναι αρκετά εύκολη. Στα παραδοσιακά νευρωνικά δίκτυα, κάθε στρώμα τροφοδοτεί το επόμενο στρώμα. Σε ένα δίκτυο με υπολειπόμενα μπλοκς, κάθε στρώμα τροφοδοτεί την επόμενη στρώση αλλά και τα μεθεπόμενα στρώματα απευθείας και μπορεί και τα επόμενα στρώματα των μεθεπόμενών. Με άλλα λόγια υπάρχει σύνδεση όχι μόνο στα ακριβώς επόμενα επίπεδα αλλά υπάρχει σύνδεση και στα στρώματα που είναι 2-3 επίπεδα πιο μακριά. Όπως έχει προαναφερθεί, η ακρίβεια αυξάνεται με τον αυξανόμενο αριθμό των στρώσεων, αλλά υπάρχει ένα όριο πρόσθεσης στον αριθμό των επιπέδων. Δυστυχώς όμως εξαιτίας κάποιων προβλημάτων, όπως το πρόβλημα του vanishing gradient, εάν έχουμε αρκετά βαθιά δίκτυα, μπορεί να μην είναι σε θέση να μάθει το δίκτυο απλές λειτουργίες όπως μια συνάρτηση-λειτουργία ταυτότητας (identity fuction). Μαθηματικά, μια συνάρτηση ταυτότητας που ονομάζεται επίσης και χαρτογράφηση ταυτότητας (identity mapping) είναι μια συνάρτηση που πάντα επιστρέφει την ίδια τιμή που χρησιμοποιήθηκε ως παράμετρο της. Το ίδιο ισχύει και στην μηχανική μάθηση. Δηλαδή η χαρτογράφηση ταυτότητας διασφαλίζει ότι η έξοδος κάποιου νευρωνικού δικτύου είναι ίση με την είσοδο του. Στις εξισώσεις, η συνάρτηση δίνεται από το f (x) = x, όπου x είναι μια τιμή εισόδου και f μια συνάρτηση η οποία παίρνει την τιμή εισόδου και ως έξοδο έχει πάλι την τιμή εισόδου. Μπορεί να ακούγεται άσκοπο να υπάρχει μια χαρτογράφηση ταυτότητας, που στην ουσία θα επαναφέρει αυτό που του δώσαμε, αλλά αυτή είναι η χρήση της. Η ανάγκη για αυτή προκύπτει από τη χρήση μιας αρχιτεκτονικής που αναμένει μια χαρτογράφηση ταυτότητας σε ένα βήμα που στην πραγματικότητα δεν χρειάζεται. Αντί να κατασκευαστεί μια νέα αρχιτεκτονική για αυτήν την περίπτωση, χρησιμοποιούμε μια χαρτογράφηση ταυτότητας που μας επιτρέπει να ξαναχρησιμοποιούμε την παλιά αρχιτεκτονική.

Όπως είχε αναφερθεί προηγουμένως, εάν συνεχιστεί η αύξηση του αριθμού των στρώσεων, θα διαπιστώσουμε ότι η ακρίβεια θα αρχίσει να κορεύεται (degrade) σε ένα σημείο και τελικά θα υποβαθμιστεί. Και, αυτό συνήθως δεν προκαλείται λόγω υπερφόρτωσης, αφού αν παρατηρήσουμε το σχήμα 2.28 η μείωση του σφάλματος

υπάρχει και κατά την εκπαίδευση και όχι μόνο κατά την δοκιμή. Αυτό το πρόβλημα είναι γνωστό ως πρόβλημα της υποβάθμισης (degradation problem). Με απλά λόγια το πρόβλημα που υπάρχει είναι ότι με την προσθήκη περισσότερών στρωμάτων στο δίκτυο η ακρίβεια εκπαίδευσης είναι χαμηλή με αποτέλεσμα να επαναφερόμαστε στα τυπικά νευρωνικά δίκτυα, και συγκεκριμένα τα shallow δίκτυα τα οποία έχουν μόνο ένα κρυφό δίκτυο. Γενικά, ερευνητές των συνελικτικών νευρωνικών δικτύων όπως ο Yann Lecun et al. το 2015 ανάφερε ότι η αιτία αυτού του προβλήματος δεν ήταν η υπερφόρτωση αλλά ούτε το φαινόμενο του vanishing gradien, αλλά η δύσκολη μάθηση της χαρτογράφησης ταυτότητας σε ορισμένα επίπεδα του δικτύου. Δηλαδή τους είναι δύσκολο να μάθουν με την συνάρτηση f(x)=x αλλά θεωρούν πιο εύκολη την μηδενική χαρτογράφηση (zero mapping) f(x)=0. Όλο αυτό οδήγησε στους ερευνητές να τροποποιήσουν την χαρτογράφηση που έπρεπε να μάθει το κάθε επίπεδο, αναλυτικότερα χρησιμοποίησαν την μηδενική χαρτογράφηση σε συνδυασμό να μαθαίνουν υπόλοιπα (residuals) προσθέτοντας τα στην είσοδο του επομένου επιπέδου. Αυτή η διαδικασία θα αναλυθεί περισσότερο στη συνέχεια.

Σύμφωνα με τα πιο πάνω υπήρξαν κάποιες σκέψεις από επιστήμονες ώστε να λύσουν τα προβλήματα που αντιμετώπιζε βαθιά μάθηση. Είχαν σκεφτεί, ότι αφού το πρόβλημα της υποβάθμισης δεν υπήρχε σε δίκτυα με πολλά επίπεδα, να κτίσουν ένα βαθύ νευρωνικό δίκτυο αλλά να υπάρχουν κάποια πηδήματα-παραλήψεις (skip connections). Για να το πετύχουν αυτό χρησιμοποιήσαν τις συνδέσεις παράκαμψης όπως έχει αναφερθεί προηγουμένως.



Σχήμα 2.31: Ένα υπολειπόμενο μπλοκ, όπου x η είσοδος/ταυτότητα και F(x) η χαρτογράφηση/λειτουργία ταυτότητας (He et al., 2016a).

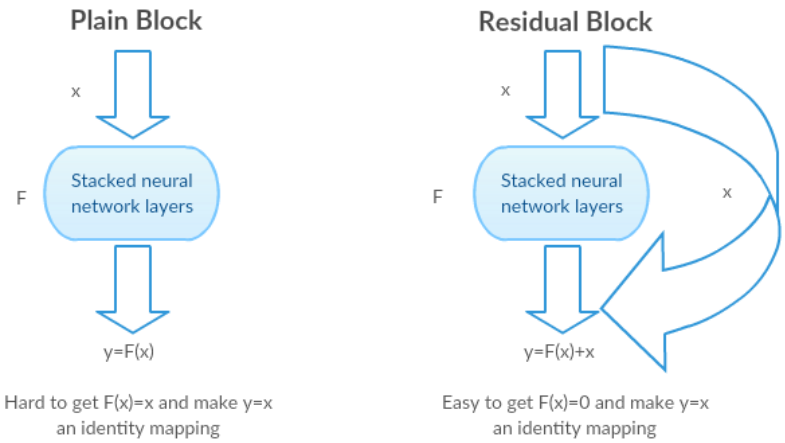
Παρατηρώντας το σχήμα πιο πάνω μπορούμε να μάθουμε την ταυτότητα x απευθείας από την σύνδεση παράκαμψης και γι' αυτό το λόγο ονομάζεται και σύνδεση συντόμευσης ταυτότητας (identity shortcut connection). Αλλά το ερώτημα είναι γιατί ονομάζεται αυτό το δίκτυο υπολειπόμενο; Γιατί υπάρχει η έννοια του υπόλοιπου; Ας υποθέσουμε ότι έχουμε ένα υπολειπόμενο μπλοκ το οποίο παίρνει σαν είσοδο μια τιμή x η οποία είναι και η επιθυμητή έξοδος και θέλει να μάθει την πραγματική έξοδο y. Άρα η διαφορά (το υπόλοιπο) που θα προκύψει είναι :

$$F(x): \text{Output} - \text{Input} \rightarrow F(x): y - x$$

Άρα η πραγματική έξοδος του συγκεκριμένου μπλοκ θα είναι:

$$y: F(x) + x$$

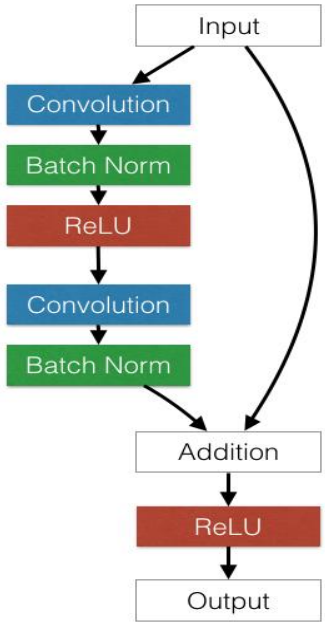
Το υπολειπόμενο μπλοκ προσπαθεί να μάθει την πραγματική έξοδο y και εφόσον από τις συνδέσεις συντόμευσης έχουμε απευθείας το x το μόνο που του απομένει είναι να μάθει το υπόλοιπο F(x). Συνοπτικά τα παραδοσιακά επίπεδα προσπαθούν να μάθουν την πραγματική έξοδο του δίκτυού ενώ τα υπολειπόμενα επίπεδα προσπαθούν να μάθουν το υπόλοιπο και γι' αυτό το λόγο ονομάζονται και υπολειπόμενα (Σχήμα 2.31). Με άλλα λόγια αν η χαρτογράφηση ταυτότητας είναι η πιο βέλτιστη, τότε είναι πιο εύκολο για τα επίπεδα να μάθουν ένα υπόλοιπο ίσο με 0 (F(x)=0 → y=x) αντί να πρέπει να μάθουν μια χαρτογράφηση ταυτότητας (identity mapping).



Σχήμα 2.32: Σύγκριση ενός τυπικού νευρωνικού μπλοκ και ενός υπολειπόμμενου μπλοκ. Αριστερά: Παραδοσιακό νευρωνικό μπλοκ με είσοδο x όπου τα επίπεδα προσπαθούν να βρουν την λειτουργία ταυτότητας F(x)=x, Δεξιά: Υπολειπόμμενο νευρωνικό μπλοκ με είσοδο x όπου τα επίπεδα προσπαθούν να μάθουν μια μηδενική ταυτότητα.

Πάρθηκε από: https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624

Ας εμβαθύνουμε περισσότερο στο υπολειπόμμενο μπλοκ για να κατανοήσουμε περισσότερο τα πλεονεκτήματα που προσφέρει αυτή η αρχιτεκτονική. Στην πιο κάτω εικόνα (Σχήμα 2.33) φαίνονται τα διάφορα στρώματα που υπάρχουν σε ένα υπολειπόμενο μπλοκ και κατά ακρίβεια υπάρχει ένα συνελικτικό επίπεδο (convolutional layers - CN), ακολουθεί ένα επίπεδο κανονικοποίησης (batch normalization -BN), μια συνάρτηση ενεργοποίησης η οποία επιλέχθηκε η RELU, η οποία αναλύθηκε στο υποκεφάλαιο 2.2.6 τονίζοντας τα πλεονεκτήματα που μπορεί να προσφέρει στην βαθιά μάθηση. Έπειτα υπάρχουν ακόμη ένα συνελικτικό επίπεδο και ακόμη ένα επίπεδο κανονικοποίησης. Υπολογίζεται η έξοδος από όλη αυτή την στοίβα επιπέδων προσθέτοντας την αρχική τιμή εισόδου (addition). Όμως αυτή η τιμή που υπολογίζεται δεν είναι η τελική μας έξοδος αλλά περνά ακόμη και από την συνάρτηση ενεργοποίησης RELU και έπειτα είναι η τελική μας έξοδος.



Σχήμα 2.33: Πρώτη αρχιτεκτονική ενός υπολειπόμενου μπλοκ.
Πάρθηκε από: http://torch.ch/blog/2016/02/04/resnets.html

Αν υποθέσουμε ότι η είσοδος (Input) είναι X_L και η έξοδος είναι η X_{L+1}, βάσει των πιο πάνω θα έχουμε τα εξής:

$$Y_{L=}F(X_L) + X_L$$

$$X_{L+1} = \mathrm{RELU}(Y_L)$$

Και αν υποθέσουμε ότι η συνάρτηση ενεργοποίησης RELU είναι συνάρτηση ταυτότητας (παίρνει σαν παράμετρο την Y_L και παράγει σαν έξοδο την Y_L, τότε θα έχουμε τα εξής:

$$Y_{L=}F(X_L) + X_L$$

$$X_{L+1} = Y_L$$

Και επομένως

$$X_{L+1} = F(X_L) + X_L$$

Αν προχωρήσουμε και στο επόμενο μπλοκ θα έχουμε τα εξής:

$$X_{L+2} = F(X_{L+1}) + X_{L+1}$$

Και βάσει της εξίσωσης

$$X_{L+1} = F(X_L) + X_L$$

Θα έχουμε

$$X_{L+2} = F(X_{L+1}) + X_{L+1=}\, F(X_{L+1}) + F(X_L) + X_L$$

$$\rightarrow X_{L+2} = X_L + F(X_L) + F(X_{L+1})$$

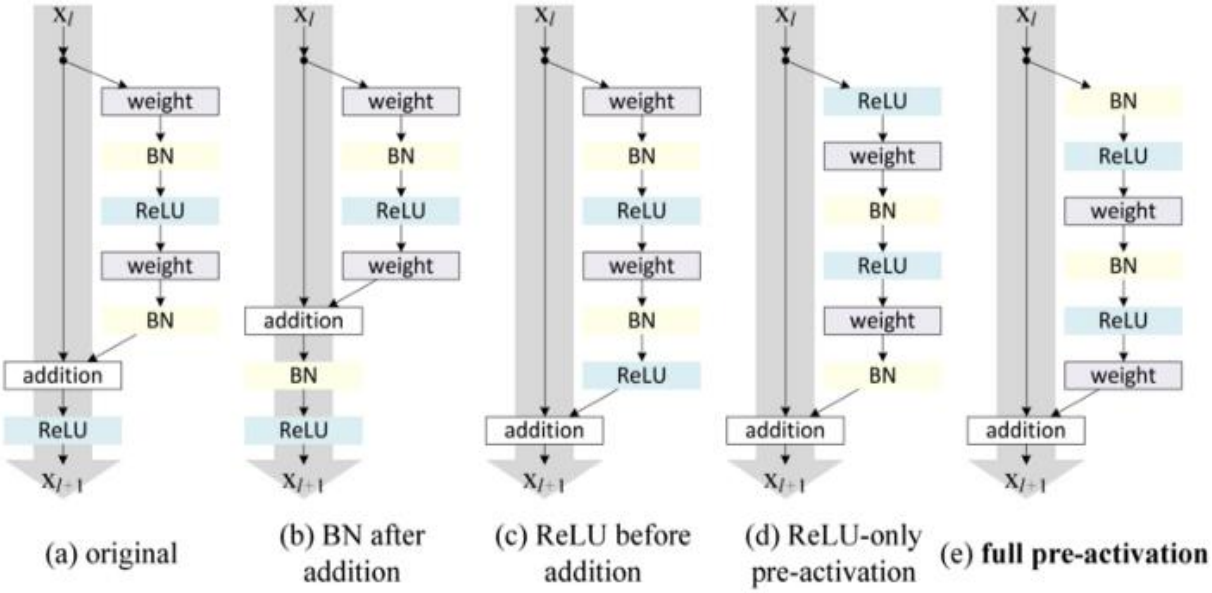
Αν υποθέσουμε ότι το τελευταίο μπλοκ είναι το L+N τότε η έξοδος του θα είναι:

$$X_{L+N} = X_L + F(X_L) + F(X_{L+1}) + F(X_{L+2}) + \ldots + F(X_{L+N-1})$$

Όλη αυτή η μαθηματική διαδικασία έχει γινεί για ένα σημαντικό σκοπό. Όπως θα παρατηρήσετέ η έξοδος κάθε επιπέδου μπορεί να υπολογιστεί βάσει της εξόδου του προηγούμενου επιπέδου και πάει λέγοντας. Βάσει αυτού, μπορούμε να συμπεράνουμε ότι οι πληροφορίες που ρέουν στα επίπεδα δεν έχουν κάποια εμπόδια και οι κλίσεις δεν εξαφανίζονται ποτέ, ανεξαρτήτως από το πόσο βάθος έχουν (Nat Roth from Microsoft, 2016). Συνεπώς με την πρόσθεση των συνδέσεων συντόμευσης αντιμετωπίζεται το πρόβλημα της εξαφάνισης της κλίσης (vanishing gradient problem) αφού με αυτή την αρχιτεκτονική οι κλίσεις δεν μειώνονται κατά την μεταφορά τους προς τα πίσω

(backpropagation) και συνεπώς θα έχουμε μεγάλες κλίσεις στα αρχικά στρώματα δίνοντας την δυνατότητα στο δίκτυο να εκπαιδεύεται βαθύτερα.

Αφού αναλύθηκε πλήρως το υπολειπόμενο μπλοκ να αναφέρουμε ότι οι ερευνητές της Microsoft είχαν τροποποιήσει την σειρά των επιπέδων στο μπλοκ για να εξετάσουν αν μπορούσαν να υπάρξουν καλύτερα αποτελέσματα (Σχήμα 2.29).



Σχήμα 2.34: Διάφοροι τύποι υπολειπόμμενων μπλοκ, όπου weights είναι ένα συνελικτικό επίπεδο αφού εκεί αλλάζουν οι τιμές των βαρών, BN είναι η κανονικοποίησης πατρίδων – batch normalization, RELU η συνάρτηση ενεργοποίησης και X_l και X_{l+1} η είσοδος και έξοδος αντίστοιχα (He et al., 2016b).

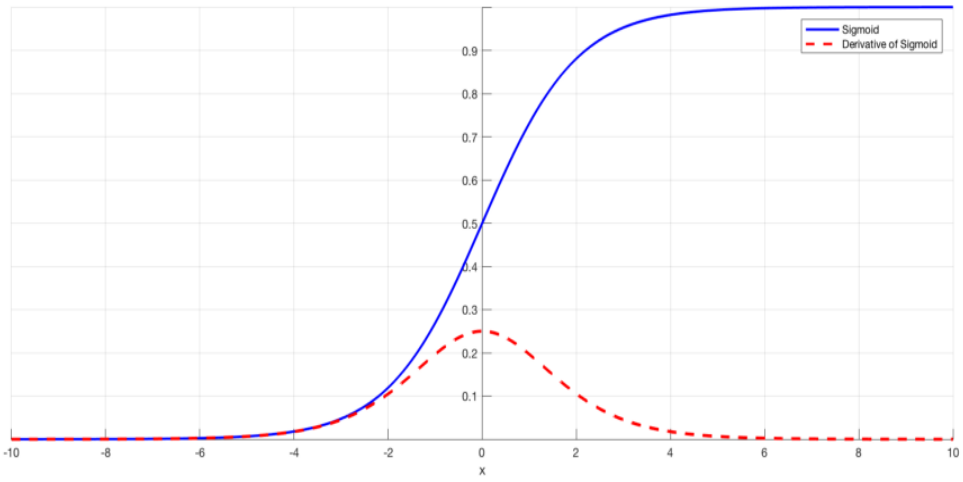
Μια αλλαγή που τέθηκε ήταν να μετακινήσουν την κανονικοποίηση παρτίδων (BN) μετά την προσθήκη (addition) (Δεύτερο σχήμα στο Σχήμα 2.34 – (b)). Αυτό όμως είχε βλαβερές συνέπειες διότι το στρώμα κανονικοποίηση εφαρμόζει την δική του ξεχωριστή παραμόρφωση της εισόδου που παίρνει, και παράγει την δική του έξοδο με αποτέλεσμα να υπάρχουν ορισμένες αλλαγές. Αυτό είχε σαν αποτέλεσμα να διαπιστώσουν οι He et al., το 2016 ότι το σφάλμα δοκιμής είχε μεγαλώσει σε σύγκριση με το σφάλμα που υπήρχε στην πρώτη τους αρχιτεκτονική (original – (a)). Συνεπώς οι συγγραφείς προσπαθήσαν να μετακινήσουν όλα τα στρώματα πριν την προσθήκη για

να μην υπάρξουν ανεπιθύμητες παραμορφώσεις. Δεδομένου τώρα ότι το αποτέλεσμα της προσθήκης μεταφέρεται απευθείας στο επόμενο στρώμα υπάρχει μεγάλη πιθανότητα η είσοδος να διατηρηθεί. Η αρχιτεκτονική που περιγράφτηκε τώρα ονομάζεται προ-ενεργοποίηση (δεξιά αρχιτεκτονική- full pre-activation). Και τελικά οι συγγραφείς δηλώσαν ότι η προ-ενεργοποίηση είχε διπλή επίδραση τόσο ως προς την ευκολία βελτιστοποίησης λόγω χαρτογραφήσεων ταυτότητας όσο και ως προς τη βελτιωμένη ρύθμιση και γενίκευση λόγω της ομαλοποίησης των εισροών από τα στρώματα BN. Υπήρξαν διάφορα πειράματα όχι μόνο με την βάση δεδομένων CIFAR-10 αλλά και με πιο μεγάλη βάση δεδομένων και συγκεκριμένα την ImageNet (He et al, 2016a). Τα αποτελέσματα που παραχθήκαν επιβεβαιώνουν ότι η προ-ενεργοποίηση παράγει το χαμηλότερο σφάλμα σε σχέση με τις υπόλοιπες αλλαγές που είχαν γινεί στην συγκεκριμένη αρχιτεκτονική.

2.2.10 Πρόβλημα Vanishing Gradient

Το πρόβλημα vanishing gradient προκύπτει κατά την εκπαίδευση ενός βαθιού τεχνητού νευρικού δικτύου με την χρήση μεθόδων εκμάθησης με βάση την κλίση (backpropagation). Σε τέτοιες μεθόδους, κάθε ένα από τα βάρη του νευρικού δικτύου λαμβάνει μια ενημέρωση αναλογική προς το αρνητικό της παράγωγου της συνάρτησης σφάλματος σε σχέση με το τρέχον βάρος σε κάθε επανάληψη της εκπαίδευσης. Το πρόβλημα είναι ότι σε ορισμένες περιπτώσεις, η κλίση θα είναι απαρατήρητη μικρή, εμποδίζοντας αποτελεσματικά το βάρος να αλλάξει την αξία του. Στη χειρότερη περίπτωση, αυτό μπορεί να σταματήσει τελείως το νευρικό δίκτυο από περαιτέρω εκπαίδευση. Με κάθε επόμενο στρώμα το μέγεθος των κλίσεων παίρνει εκθετικά μικρότερη τιμή (εξαφανίζεται) κάνοντας τα βήματα επίσης πολύ μικρά, πράγμα που οδηγεί σε πολύ αργή και δύσκολη εκπαίδευση (Chi-Feng Wang, 2018).

Ορισμένες συναρτήσεις ενεργοποίησης, όπως η σιγμοειδής συνάρτηση, συμπιέζει ένα μεγάλο αριθμό δεδομένων εισόδου σε ένα πολύ μικρό πεδίο ορισμού (μεταξύ 0 και 1) με αποτέλεσμα μια μεγάλη αλλαγή που έχει στην είσοδο της σιγμοειδούς συνάρτησης (η παράγωγος γίνεται αρκετά μικρή ως συνέπεια οι κλίσεις να εξαφανίζονται αρκετά γρήγορα) θα προκαλέσει μια πολύ μικρή αλλαγή στην έξοδο (Σχήμα 2.35).



Σχήμα 2.35: Η σιγμοειδής συνάρτηση ενεργοποίησης και η παράγωγος της.
Πάρθηκε από: https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

Υπάρχουν όμως και συναρτήσεις ενεργοποίησής που δεν επηρεάζονται από αυτό το πρόβλημα για παράδειγμα η συνάρτηση Rectified στην οποία η κλίση είναι 0 για αρνητικές (και μηδενικές) εισόδους και 1 για θετικές εισόδους. Συνεπώς είναι και η απλούστερη λύση που μπορεί να χρησιμοποιηθεί για να επιλύσει το πρόβλημα vanishing gradient. Μια άλλη λύση που μπορεί να αντιμετωπίσει αυτό το πρόβλημα είναι το batch normalization το οποίο κανονικοποιεί τις εισόδους ούτως ώστε να βοηθά στο να μην μειωθεί αρκετά η κλίση της συνάρτησης που χρησιμοποιείται. Επίσης σημαντική λύση είναι τα υπολειπόμενα δίκτυα στα οποία υπάρχουν επιπρόσθετες συνδέσεις οι οποίες δεν περνούν από κάποια συνάρτηση ενεργοποίησης αλλά κατευθύνονται απευθείας στα επόμενα επίπεδα όπως έχει αναφερθεί στο προηγούμενο υποκεφάλαιο.

Κεφάλαιο 3

Σχεδίαση και Υλοποίηση

3.1 Δεδομένα Εισόδου 69

3.1.1 Δεδομένα Εκπαίδευσης και Επαλήθευσης 69

3.1.2 Αναπαράσταση δεδομένων στα υπολειπόμμενα δίκτυα 71

3.1.2.1 Πρόβλημα αναπαράστασης δεδομένων στα CNN και η επίλυση του 72

3.1.2.2 DSSP Κωδικοποίηση 74

3.1.2.3 Αρχεία πολλαπλής στοίχισης (MSA) 74

3.1.2.4 Τεχνική Σημαντικότερων Γειτονικών Αμινοξέων 76

3.2 Βιβλιοθήκη Tensorflow 77

3.2.1 Γενικά 78

3.2.2 Προσαρμογή Βιβλιοθήκης στο PSSP Πρόβλημα 78

3.2.2.1 Βάση MNIST 79

3.2.2.2 Δεδομένα Εισόδου 81

3.2.2.3 Δεδομένα Εξόδου 81

3.2.2.4 Εκπαίδευση Δικτύου 82

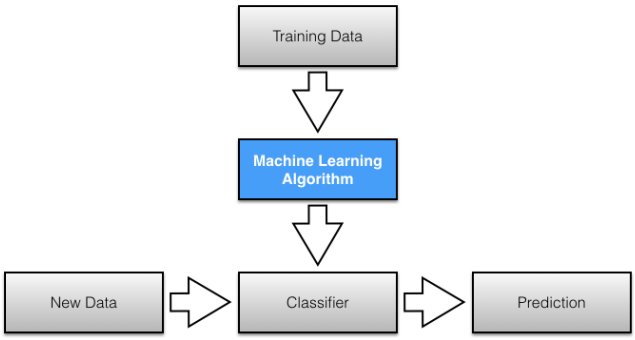
3.1 Δεδομένα Εισόδου

3.1.1 Δεδομένα Εκπαίδευσης και Επαλήθευσης

Στη μηχανική μάθηση γνωρίζουμε ότι είναι απαραίτητη η διαδικασία εκπαίδευσης ενός τεχνητού δικτύου ούτως ώστε να μπορεί να μάθει και να γενικεύσει ορισμένα χαρακτηριστικά που μαθαίνει. Σημαντικός και απαραίτητος παράγοντας κατά την εκπαίδευση ενός NN είναι τα δεδομένα που παρουσιάζονται σε αυτό. Η εκπαίδευση (training) παίζει σημαντικό ρόλο κατά την διαδικασία μάθησης ενός τεχνητού νευρωνικού δικτύου αλλά η διαδικασία που εγγυάται και ελέγχει ότι όντως το δίκτυο έχει μάθει και είναι σε θέση να γενικεύει όσο το δυνατό μπορεί είναι η διαδικασία επαλήθευσης/ δοκιμής (testing). Συνοπτικά, υπάρχουν 2 διαφορετικές διαδικασίες που γίνονται για την κατασκευή του τελικού δικτύου οι οποίες είναι η διαδικασία εκπαίδευσης και η διαδικασία επαλήθευσης. Σε κάθε διαδικασία υπάρχουν και τα αντίστοιχα δεδομένα που πρέπει να παρουσιαστούν στο δίκτυο. Τα δεδομένα αυτά ονομάζονται δεδομένα εκπαίδευσης (training dataset) και δεδομένα επαλήθευσης (testing dataset) στην αντίστοιχη διαδικασία. Ο διαχωρισμός των δεδομένων σε ένα σύνολο εκπαίδευσης και επαλήθευσης είναι αρκετά σημαντικός για την αξιολόγηση του νευρωνικού δικτύου. Συνήθως όταν διαχωρίζουμε ένα σύνολο δεδομένων σε ένα σετ εκπαίδευσης και σετ δοκιμών, τα περισσότερα δεδομένα χρησιμοποιούνται για την εκπαίδευση, περίπου το 80%,και ένα μικρότερο μέρος των δεδομένων, περίπου 20%, χρησιμοποιείται για έλεγχο/ επαλήθευση. Σημαντικό να τονιστεί ότι τα δεδομένα εκπαίδευσης και τα δεδομένα επαλήθευσης δεν είναι ακριβώς τα ίδια αλλά έχουν παρόμοια χαρακτηριστικά. Αυτό θα βοηθήσει το δίκτυο να εντοπίσει καλύτερα τα επιθυμητά χαρακτηριστικά των δεδομένων κατά την διαδικασία επαλήθευσης αφού από πριν, στην διαδικασία εκπαίδευσης θα είχε συγκεντρώσει ορισμένα χαρακτηριστικά. Με άλλα λόγια το σύνολο δοκίμων περιέχει ήδη γνωστές τιμές για το επιθυμητό αποτέλεσμα που θέλει να προβλέψει το δίκτυο, αφού είχε ‘δει’ και μάθει από πριν παρόμοιες τιμές από την διαδικασία εκπαίδευσης, με αποτέλεσμα να του είναι εύκολο να προσδιορίσει σωστές προβλέψεις.

Τα δεδομένα εκπαίδευσης παρουσιάζονται πρώτα στο δίκτυο για να το βοηθήσουν να μάθει και να μπορεί να γενικεύει. Συγκεκριμένα τα δεδομένα εκπαίδευσης είναι ένα σύνολο παραδειγμάτων (συμβολοσειρές, αριθμοί -vectors) τα οποία χρησιμοποιούνται για την προσαρμογή των παραμέτρων. Μια σημαντική παράμετρος είναι οι συνδέσεις των νευρώνων, τα βάρη, τα οποία αλλάζουν μόνο κατά την εκπαίδευση του δικτύου. Ουσιαστικά τα δεδομένα εκπαίδευσης χρησιμοποιούνται κατά την διαδικασία εκπαίδευσης στην οποία το δίκτυο εκπαιδεύεται λαμβάνοντας υπόψη αυτά τα δεδομένα σαν εισόδους του δικτύου και χρησιμοποιώντας ένα είδος αλγορίθμου μάθησης που θα τροποποιήσει τις μεταβαλλόμενές παραμέτρους του δικτύου, την ώρα της προς τα πίσω διαδρομής, για να μπορέσει το δίκτυο να γενικεύσει αυτή την γνώση που αποκτά. Για παράδειγμα στην συγκεκριμένη έρευνα θα χρησιμοποιηθεί ένας επιβλεπόμενος αλγόριθμος μάθησής (supervised learning algorithm) και βάσει αυτού το σύνολο δεδομένων χωρίζεται σε ζεύγη τα οποία αποτελούνται από την είσοδο και την επιθυμητή έξοδο. Το τρέχον δίκτυο παίρνει τα ζεύγη αυτά και παράγει ένα αποτέλεσμα το οποίο στην μηχανική μάθηση καλείται ως πραγματική έξοδος. Έπειτα το δίκτυο συγκρίνει την πραγματική έξοδος με την επιθυμητή έξοδο και βάση αυτής της σύγκρισης και του αλγορίθμου μάθησης ρυθμίζονται ανάλογα οι παράμετροι του μοντέλου αυτού.

Μετά από αυτή την διαδικασία εκπαίδευσης το δίκτυο περνά στην διαδικασία επαλήθευσης η οποία αυτή θα φανερώσει αν πράγματι το δίκτυο έχει εκπαιδευτεί καλά. Πιο συγκεκριμένα το δίκτυο παίρνει τα δεδομένα επαλήθευσης τα οποία μοιάζουν με τα δεδομένα εκπαίδευσης και με τις γνώσεις και τους γενικούς κανόνες που έχει δημιουργήσει κατά την ώρα της εκπαίδευσης του, παράγει το πραγματικό αποτέλεσμα το οποίο θεωρητικά αν έχει εκπαιδευτεί σωστά, δεν θα έχει μεγάλη διαφορά με το επιθυμητό αποτέλεσμα. Να τονιστεί εδώ ότι κατά την διαδικασία επαλήθευσης δεν υπάρχει οποιαδήποτε αλλαγή παραμέτρων που γίνεται κατά την προς τα πίσω διαδρομή του δικτύου, αλλά απλά ένα πέρασμα των δεδομένων επαλήθευσης. Επομένως ένα σετ δοκιμών είναι ένα σύνολο παραδειγμάτων που χρησιμοποιείται μόνο για την εκτίμηση της απόδοσης (γενίκευση) ενός τεχνητού νευρωνικού δικτύου.



Σχήμα 3.1: Διαδικασία κατασκευής τελικού τεχνητού νευρωνικού δικτύου το οποίο χρησιμοποίει επιβλεπόμενο αλγόριθμο μάθησης κατά την εκπαίδευση του (training data + machine learning algorithm) με αποτέλεσμα να μετατρέπεται σε ένα μοντέλο κατηγοριοποίησης (classifier).Και έπειτα με νέα δεδομένα εισόδου (testing data / new data) ελέγχει κατά πόσο είναι γίνεται σωστή κατηγοριοποίηση.
Πάρθηκε από: https://sebastianraschka.com/images/blog/2014/intro_supervised_learning/

3.1.2 Αναπαράσταση δεδομένων στα υπολειπόμενα δίκτυα

Στη συγκεκριμένη έρευνα θα χρησιμοποιηθούν τα υπολειπόμενα δίκτυα για να λυθεί το πρόβλημα της πρόβλεψης της δευτεροταγούς δομής της πρωτεΐνης. Όπως έχει αναφερθεί, τα υπολειπόμενα νευρωνικά δίκτυα αποτελούνται από συνελικτικά δίκτυα με την διαφορά ότι υπάρχουν επιπρόσθετες συνδέσεις στην αρχιτεκτονική τους. Αφού η κύρια μας δομή είναι ένα συνελικτικό δίκτυο και αφού ένα συνελικτικό δίκτυο παίρνει σαν είσοδο εικόνες και συγκεκριμένα ένα τρισδιάστατο πίνακα, τα δεδομένα εισόδου μας κατά την διαδικασία εκπαίδευσης και επαλήθευσης πρέπει να έχουν αυτή την δομή.

Αρχικά, πριν να προχωρήσουμε στην ανάλυση της αναπαράστασης των δεδομένων στα υπολειπόμενα δίκτυα να τονίσουμε ποια είναι τα δεδομένα εισόδου και εξόδου για την συγκεκριμένη έρευνα. Η συγκεκριμένη έρευνα ασχολείται με την επίλυση του PSSP προβλήματος. Δηλαδή, προσπαθούμε να βρούμε μια λύση για την πρόβλεψή της δευτεροταγούς δομής της πρωτεΐνης. Άρα η επιθυμητή μας έξοδος θα είναι η δευτεροταγής δομή της πρωτεΐνης. Όπως είχαμε πει, η δευτεροταγής δομή προκύπτει από αντιδράσεις των αμινοξέων που βρίσκονται μέσα σε μια πρωτεΐνη (υποκεφάλαιο

2.1). Βάσει αυτού, για να μπορέσουμε να προβλέψουμε αυτή την δομή θα πρέπει να λάβουμε υπόψη την αλληλουχία των αμινοξέων η οποία είναι η πρωτοταγής δομή της πρωτεΐνης. Άρα η είσοδος που πρέπει να παρουσιαστεί στο δίκτυο μας είναι η πρωτοταγής δομή της πρωτεΐνης. Συνεπώς στόχος αυτής της έρευνας είναι η πρόβλεψη της δευτεροταγούς δομής της πρωτεΐνης εισάγοντας στο δίκτυο την πρωτοταγή της δομή. Να τονίσουμε ότι θα χρησιμοποιηθεί ένας επιβλεπόμενος αλγόριθμος μάθησης και επομένως τα δεδομένα εκπαίδευσης και επαλήθευσης θα περιέχουν τα ζεύγη (είσοδος – επιθυμητή έξοδος) που έχουμε αναφέρει πιο πάνω. Αφού έχουμε εντοπίσει τώρα ποια είναι η είσοδος και ποια είναι η έξοδος του νευρωνικού δικτύου, πρέπει να προσαρμοστούν αυτά τα στοιχεία σε ένα υπολειπόμενο δίκτυο, που στην ουσία είναι ένα CNN.

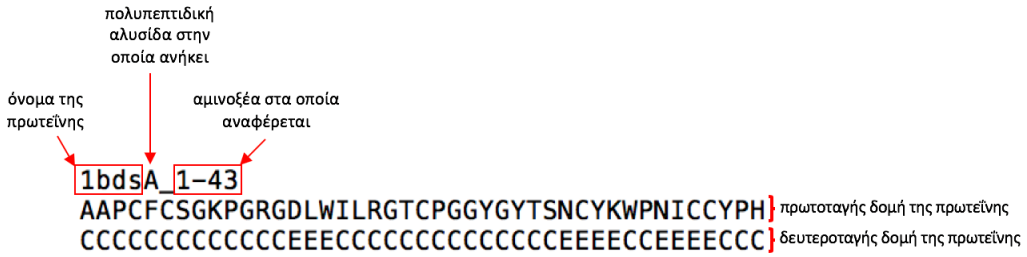
3.1.2.1 Πρόβλημα αναπαράστασης δεδομένων στα CNN και η επίλυση του

Τα δεδομένα που είχαμε στην διάθεση μας γι’ αυτή την διπλωματική ήταν 10 επεξεργαζόμενα αρχεία τα οποία ο Διονυσίου Ανδρέας τα είχε δανειστεί από την γνωστή βάση δεδομένων CB513 (Cuff and Barton, 1999), το οποίο περιέχει 513 πρωτεϊνες, σε συνδυασμό με τα αρχεία πολλαπλής στοίχισης (Multiple Sequence Alignment (MSA) profiles). Χρησιμοποιήθηκε αυτό το dataset λόγω του ότι είναι μια από τις γνωστές βάσεις για το πρόβλημα PSSP ούτως ώστε να υπάρχει μια εφικτή σύγκριση μεταξύ των διάφορων μεθόδων ως προς την απόδοση, αλλά το πιο σημαντικό είναι ότι υπήρξε σημαντική βελτιστοποίηση στα συγκεκριμένα αρχεία από τον Διονυσίου με αποτέλεσμα να υπάρχει μια καλύτερη αναπαράσταση των δεδομένων στο νευρωνικό δίκτυο. Σημαντικό να τονιστεί ότι το κύριο πρόβλημα που αντιμετώπισαν αρκετοί φοιτητές και ερευνητές κατά την προσπάθεια τους να επιλύσουν το PSSP πρόβλημα ήταν η δύσκολη αναπαράσταση των δεδομένων στο δίκτυο. Όπως είχε αναφερθεί και προηγουμένως, είναι αρκετά σημαντικό το πως θα παρουσιαστούν τα δεδομένα σε ένα δίκτυο και στην συγκεκριμένη περίπτωση, για ένα συνελικτικό δίκτυο το οποίο είχε σαν στόχο την επίλυση ενός σύνθετου προβλήματος ταξινόμησης διαδοχικών δεδομένων. Ο Διονυσίου λοιπόν είχε σαν στόχο να δημιουργήσει μια εξαιρετική αναπαράσταση δεδομένων ούτως ώστε να παρουσιαστούν σωστά τα δεδομένα στο συνελικτικό του δίκτυο με αποτέλεσμα να παραχθούν ψηλά ποσοστά

επιτυχίας. Συγκεκριμένα λήφθηκε υπόψη ποια δεδομένα εισόδου έπρεπε να δώσει στο δίκτυο και τόνισε ότι είχε επιλέξει πρωτεΐνες οι οποίες να περιέχουν στοιχεία τα οποία να δίνουν μια γενική εικόνα και να αντιπροσωπεύουν όλο το σύνολο των πρωτεϊνών και όχι ένα συγκεκριμένο με αποτέλεσμα την κατασκευή ενός αποδοτικού συνόλου και συνεπώς και μια αποδοτική εκπαίδευση του συνελικτικού δικτύου. Όμως το ερώτημα που προκύπτει από όλο αυτό είναι το εξής: Γιατί θεωρείται πρόβλημα η αναπαράσταση δεδομένων σε ένα συνελικτικό δίκτυο;

Όπως αναφέρθηκε παραπάνω, τα CNNs είναι σε θέση να αναλύουν εισόδους τύπου εικόνας. Το κύριο εμπόδιο στην προσπάθεια επίλυσης ενός σύνθετου προβλήματος ταξινόμησης διαδοχικών δεδομένων με CNNs είναι η αναπαράσταση των δεδομένων, με τέτοιο τρόπο ώστε το δίκτυο να είναι σε θέση όχι μόνο να κατανοήσει το σχήμα της εισόδου, αλλά και να μπορεί να κατανοήσει τις σύνθετες αλληλουχίες που έχουν μεταξύ τους. Άρα για να μπορεί ένα συνελικτικό δίκτυο να προβλέψει την δευτεροταγή δομή της συγκεκριμένης πρωτεΐνης πρέπει η πρωτοταγής δομή να μετατραπεί σε μια μορφή η οποία να αντιπροσωπεύει μια δομή εικόνας για να μπορεί να εντοπίσει κάποιες συνθέτες συσχετίσεις, και συγκεκριμένα τις αλληλεπιδράσεις των κοντινών και απομακρυσμένων αμινοξέων που περιέχει η πρωτεΐνη, και συνεπώς να μπορεί να προβλέψει σωστά την δευτεροταγή δομή της πρωτεΐνης.

Τα αρχεία που περιέχουν το συγκεκριμένο σύνολο πρωτεϊνών έχουν συγκεκριμένη μορφή η οποία φαίνεται στο σχήμα πιο κάτω (Σχήμα 3.2). Οι πρώτοι τέσσερις χαρακτήρες της πρώτης γραμμής καθορίζουν το όνομα της πρωτεΐνης, ο πέμπτος χαρακτήρας αντιπροσωπεύει την πολυπεπτιδική αλυσίδα στην οποία ανήκει, ενώ οι αριθμοί μετά τον χαρακτήρα "_" αναφέρονται στα συγκεκριμένα αμινοξέα της πρωτεΐνης.



Σχήμα 3.2: Αναπαράσταση μίας πρωτεΐνης που υπάρχει στο σύνολο δεδομένων (Δημητρίου, 2018).

Στη δεύτερη γραμμή βρίσκεται η πρωτοταγής δομή της συγκεκριμένης πρωτεΐνης (είσοδος) και στην τρίτη γραμμή η δευτεροταγής της δομή (έξοδος).

3.1.2.2 DSSP Κωδικοποίηση

Όπως φαίνεται και στο σχήμα πιο πάνω η κωδικοποίηση της δευτεροταγούς δομής της πρωτεΐνης έχει γινεί με την γνωστή τυποποίηση DSSP (Dictionary for Secondary Structure of Proteins). Ο αλγόριθμος DSSP προσδιορίζει την δευτεροταγή δομής μιας πρωτεΐνης με κωδικοποίηση ενός γράμματος. Υπάρχουν οκτώ είδη δευτερεύουσας δομής. Οι 3_{10} έλικες, ο α έλικας και π έλικας τα οποία συμβολίζονται G, H και I αντίστοιχα και αναγνωρίζονται από την επαναλαμβανόμενη αλληλουχία δεσμών υδρογόνου στην οποία τα υπολείμματα είναι τρία, τέσσερα ή πέντε κατάλοιπα αντίστοιχα. Υπάρχουν επίσης τρία είδη beta, μια β-bridge η οποία έχει το σύμβολο B, τα β-strand τα οποία έχουν μακρύτερα σύνολα δεσμών υδρογόνου και συμβολίζονται με το γράμμα E, και το T συμβολίζει τις πρωτεΐνες με στροφές που μετατρέπονται από δεσμούς υδρογόνου των ελίκων (β-turn). Επίσης υπάρχουν και τα bend (S) και coil (C) τα οποία δεν ανήκουν στις πιο πάνω κατηγορίες. Η δυαδική αναπαράσταση αυτών των οκτώ κατηγοριών είναι η εξής: G – 100, H – 011, I – 101, T – 111, E – 010, B – 000, S – 110, C – 001.

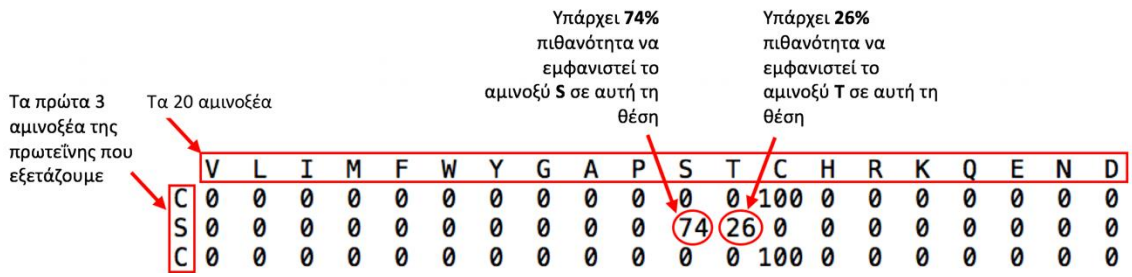
Αυτές οι οκτώ κατηγορίες ομαδοποιούνται σε τρεις μεγαλύτερες κατηγορίες: H (Helix) η οποία περιλαμβάνει τις ομάδες G, H και I, E (Extended Strand) η οποία περιλαμβάνει τις ομάδες E και B, και C η οποία περιλαμβάνει τις υπόλοιπες ομάδες. Η δυαδική αναπαράσταση αυτών των τριών κατηγοριών είναι η εξής: C – 001, E – 010, H-100

3.1.2.3 Αρχεία πολλαπλής στοίχισης (MSA)

Στο προηγούμενο υποκεφάλαιο αναφερθήκαν τα αρχεία πολλαπλής στοίχισης (MSA) τα οποία έχουν σημαντικό ρόλο στην αναπαράσταση των δεδομένων στο δίκτυο μας. Τα πολλαπλής στοίχισης αρχεία, ή αλλιώς αρχεία ευθυγράμμισης πολλαπλών αλληλουχιών είναι αρχεία τα οποία ευθυγραμμίζουν 3 ή περισσότερες αλληλουχίες.

Αυτά τα αρχεία χρησιμοποιούνται για να διατηρείται η αλληλουχία πρωτεϊνικών περιοχών. Η ευθυγράμμιση των αλληλουχιών σχετικού μήκους μπορεί να είναι δύσκολη και σχεδόν πάντα χρονοβόρα και ειδικά με το χέρι, χρησιμοποιούνται υπολογιστικοί αλγόριθμοι για την παραγωγή και την ανάλυση των ευθυγραμμίσεων αυτών. Σε αυτή την έρευνα τα αρχεία αυτά βοήθησαν σε μεγάλο βαθμό να λύσουν το πρόβλημα της αναπαράστασης των δεδομένων σε ένα συνελικτικό δίκτυο. Συγκεκριμένα ο Διονυσίου, αναδιοργανωσε τα MSA αρχεία με τέτοιο τρόπο ούτως ώστε να εντοπιστούν ομοιότητές μεταξύ των αλληλουχιών των αμινοξέων για να μπορούμε να μελετήσουμε τις σχέσεις που έχουν μεταξύ τους και συγκεκριμένα τις αλληλεπιδράσεις του, ούτως ώστε το δίκτυο να μπορεί παράξει γενικούς κανόνες και να εκπαιδευτεί βάσει αυτών των σχέσεων. Δηλαδή βάσει των κοινών χαρακτηριστικών που θα μάθει το δίκτυο θα μπορεί να κατηγοριοποιήσει τις πρωτεΐνες που του παρουσιάζονται σε διάφορες ομάδες με αποτέλεσμα να υπάρχει μια ποικιλομορφία δεδομένων. Γενικά τα αρχεία πολλαπλής στοίχισης αλληλουχιών έχουν μια μορφή κωδικοποίησης η οποία μπορεί εύκολα ένα τεχνητό νευρωνικό δίκτυο να την κατανοήσει.

Τα αρχεία αυτά είναι στην ουσία CSV αρχεία, διαχωρισμός κάθε τιμής με κόμμα, τα οποία αντιπροσωπεύουν μια πρωτεΐνη το καθένα. Αναλυτικότερα ένα αρχείο περιέχει N γραμμές, όπου κάθε γραμμή είναι ένα αμινοξύ που περιέχει η συγκεκριμένη πρωτεΐνη, και 20 στήλες, όπου κάθε στήλη είναι η πιθανότητα εμφάνισης του αμινξέους στην συγκεκριμένη θέση. Να τονίσουμε ότι αφού το N αντιπροσωπεύει όλα τα αμινοξέα μιας πρωτεΐνης, οι γραμμές δεν θα έχουν σταθερό αριθμό, διότι κάθε πρωτεΐνη δεν αποτελείται από τα ίδια αμινοξέα. Με αυτό τον τρόπο, δηλαδή η τοποθέτηση αμινοξέων το ένα κάτω από το άλλο, δημιουργείται μια είσοδος τύπου εικόνας που θα βοηθήσει ένα CNN δίκτυο να κατανοήσει τα δεδομένα και τις σχέσεις μεταξύ τους. Άρα το δίκτυο μας θα παίρνει σαν είσοδο ένα 2D πίνακα Nx20 αντί ένα τρισδιάστατο όπως είχε σημειωθεί προηγουμένως με αποτέλεσμα μια πιο εύκολή και σωστή αναπαράσταση των δεδομένων (Σχήμα 3.3).



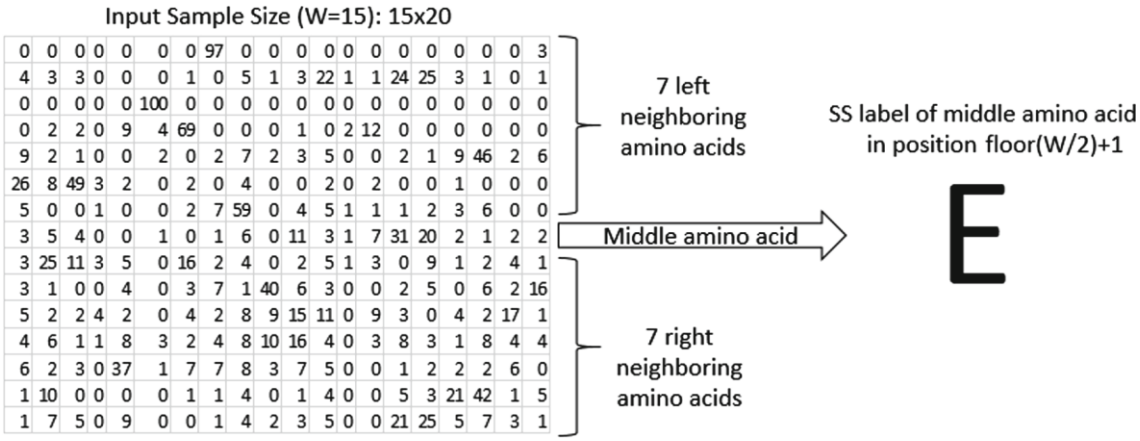
Σχήμα 3.3: Κομμάτι του MSA αρχείου που αναπαριστά την πρωτεΐνη 1ednA_1-21 (Δημητρίου, 2018).

3.1.2.4 Τεχνική Σημαντικότερων Γειτονικών Αμινοξέων

Σύμφωνα με την έρευνα του Wang et al. (2016) σημαντικό είναι να λάβουμε υπόψη την τεχνική των γειτονικών αμινοξέων. Αυτή η τεχνική τροποποιεί τα δεδομένα εισόδου με στόχο την καλύτερη πρόβλεψη της δευτεροταγούς δομής της πρωτεΐνης. Η έρευνα του Wang et al. (2016) τόνισε ότι η δευτεροταγής δομή ενός αμινοξέος επηρεάζεται αρκετά από τα γειτονικά αμινοξέα του. Όπως είδαμε στο κεφάλαιο που αναλύθηκε το βιολογικό υπόβαθρο, τονίστηκε ότι η σειρά με την οποία βρίσκονται τοποθετημένα τα αμινοξέα παίζει σημαντικό ρόλο για τον καθορισμό της δευτεροταγής δομής. Συνεπώς λοιπόν λάβαμε υπόψη έναν αριθμό γειτονικών αμινοξέων. Ο Διονυσίου πρόσθεσε στην ουσία σε κάθε αμινοξύ, εγγραφή-γραμμή, και τα στοιχεία των γειτονικών αμινοξέων του συγκεκριμένου αμινοξύ. Δηλαδή σε κάθε γραμμή περιείχε τις 20 τιμές του κάθε προηγούμενου γειτονικού αμινοξέος, τις 20 τιμές του συγκεκριμένου αμινοξέος που μελετάτε και τις 20 τιμές του κάθε επόμενου γειτονικού αμινοξέος. Όσο αφορά τον αριθμό των γειτονικών αμινοξέων που έπρεπε να προστεθεί στα δεδομένα εισόδου, υπήρξαν ορισμένα πειράματα από τον Διονυσίου με διάφορα μεγέθη παραθύρων. Συμπέρανε ότι ο αριθμός 15 ήταν ο πιο αποδοτικός αφού έδινε και τα καλύτερα αποτελέσματα.

Συμπερασματικά, το δίκτυο θα παίρνει κάθε εγγραφή η οποία θα αποτελείται από έναν 2D πίνακα διαστάσεων 15x20 όπου θα λαμβάνονται υπόψη τα 7 προηγούμενα

γειτονικά αμινοξέα , το μεσαίο που θα είναι το αμινοξύ που μελετάμε και τα επόμενα 7 γειτονικά αμινοξέα (Σχήμα 3.4).



Σχήμα 3.4: Παράδειγμα απεικόνισης δεδομένων ενός δείγματος εισόδου χρησιμοποιώντας ένα μέγεθος παραθύρου 15 αμινοξέων. Κάθε γραμμή αντιπροσωπεύει την MSA εγγραφή για το συγκεκριμένο αμινοξύ. Το E είναι η ετικέτα, η έξοδος του δικτύου, για το μεσαίο αμινοξύ.

Πάρθηκε από: Dionysiou et al. (2018).

Η είσοδος του δίκτυού μας λοιπόν είναι οι δυσδιάστατοι πίνακες που έχουν αναλυθεί. Όσο αφορά την επιθυμητή έξοδος, η οποία πρέπει να εισαχθεί και αυτή στο δίκτυο είναι τοποθετημένη στην τελευταία στήλη στα MSA αρχεία στο το κάθε αμινοξύ (για κάθε γραμμή). Με άλλα λόγια το δίκτυο θα διαβάζει το κάθε αμινοξύ και παράλληλα θα έχει αποθηκευμένή την επιθυμητή κατηγορία που ανήκει το κάθε αμινοξύ. Συγκεκριμένα η επιθυμητή έξοδος του κάθε αμινοξύ, ή αλλιώς η ετικέτα της κάθε εγγραφής χωρίζεται στις 3 κατηγορίες που αναφέραμε πιο πάνω. Η ετικέτα που αντιπροσωπεύει την κατηγορία C είναι ο αριθμός 0, η ετικέτα για την κατηγορία E είναι ο αριθμός 1 και η ετικέτα για την κατηγορία H είναι ο αριθμός 2.

3.2 Βιβλιοθήκη Tensorflow

3.2.1 Γενικά

Η βιβλιοθήκη Tensorflow είναι μια πολύ ισχυρή πλατφόρμα ανοιχτού κώδικα που χρησιμοποιείται για την υλοποίηση και την ανάπτυξη μεγάλης κλίμακας μοντέλων μηχανικής μάθησης. Η βιβλιοθήκη αυτή αναπτύχθηκε από την ομάδα της Google Brain για εσωτερική χρήση της Google και έχει διαδοθεί ελεύθερα στις 9 Νοεμβρίου το 2015. Περιέχει ένα ολοκληρωμένο και ευέλικτο οικοσύστημα εργαλείων με ενσωματωμένες άλλες βιβλιοθήκες έτσι ώστε να επιτρέπει στους ερευνητές εύκολα να αναπτύξουν μοντέλα μηχανικής μάθησης. Αξίζει να τονιστεί ότι με τα χρόνια η βιβλιοθήκη αυτή έχει γινεί μια από τις πιο δημοφιλείς βιβλιοθήκες για βαθιά εκμάθηση.

Η βιβλιοθήκη Tensorflow δίνει την δυνατότητα στο χρήστη να εκπαιδεύσει και να ελέγξει διάφορα μοντέλα εκμάθησης. Αυτό γίνεται διότι περιέχει πολλές έτοιμές συναρτήσεις οι οποίες μπορούν εύκολα να ενσωματωθούν στο δίκτυο με αποτέλεσμα να μην χρειάζεται ο χρήστης να υλοποιήσει από μόνος του ένα μεγάλο σε κλίμακα κώδικα. Άρα λόγω της διαθεσιμότητας των πολλών βοηθητικών βιβλιοθηκών και συναρτήσεων που προσφέρει η βιβλιοθήκη αυτή, ειδικά για την βαθιά μάθηση, την κάνει κατάλληλη για να λύσουμε το πρόβλημα της συγκεκριμένης έρευνας. Επίσης αυτή η βιβλιοθήκη είναι βασισμένη στην γνωστή γλώσσα Python που και αυτό είναι ένα αρκετά μεγάλο πλεονέκτημα για την υλοποίηση του δικτύου μας, αφού η Python είναι μια απλή γλώσσα υψηλού επιπέδου η οποία μπορεί να χρησιμοποιηθεί εύκολα για πολλούς σκοπούς.

3.2.2 Προσαρμογή Βιβλιοθήκης στο PSSP Πρόβλημα

Για την επίλυση του προβλήματος PSSP θα χρησιμοποιηθεί η βιβλιοθήκη Tensorflow για την δημιουργία ενός υπολειπόμενου δικτύου. Επιλέχθηκε το συγκεκριμένο δίκτυο γιατί μπορεί να αντιμετωπίσει ορισμένα προβλήματα τα οποία μέχρις στιγμής άλλα βαθιά νευρωνικά δίκτυα δεν μπορούσαν να τα αντιμετωπίσουν. Επίσης είναι μια πρόσφατη εφεύρεση η οποία έχει γινεί διάσημη για τα πόσο υψηλά ποσοστά επιτυχίας

παράγει δεδομένου ότι δεν υπήρξαν δραματικές αλλαγές στην αρχιτεκτονική του δικτύου σε σχέση με τα προηγούμενα βαθιά δίκτυα. Επομένως η διπλωματική αυτή μελετά αν όντως αυτού του είδους δίκτυα μπορούν να παράξουν υψηλότερα ποσοστά επιτυχίας για το πρόβλημα πρόβλεψής δευτεροταγούς δομής της πρωτεΐνης.

Η υλοποίηση του υπολειπόμενου δικτύου έχει γινεί στην γλώσσα Python με την χρήση της βιβλιοθήκης Tensorflow στο περιβάλλον του Jupyter Notebook. Το Jupyter Notebook έδωσε την δυνατότητα στο χρήστη να δουλέψει σε ένα πολύ οικείο περιβάλλον, δηλαδή το περιβάλλον είχε την μορφή ενός τετραδίου όπου μπορούσε να εισάγει οποιοδήποτε κώδικα σε διάφορα κουτιά - μπλοκς και να τα τρέξει ξεχωριστά. Αυτό μας έχει βοηθήσει στο να προσθέσουμε συγκεκριμένες λειτουργίες στο κώδικα και να τις ελέγξουμε χωρίς να επηρεάσουμε οποιοδήποτε άλλο κομμάτι του κώδικα, αφού είχαμε στην αρχή ξεχωριστά μπλοκς τα οποία ήταν ανεξάρτητα το ένα από το άλλο. Σημαντικό να τονιστεί ότι αυτό το περιβάλλον βοήθησε αρκετά στην επεξεργασία των δεδομένων που θα δούμε στην συνέχεια.

3.2.2.1 Βάση MNIST

Έχουν γινεί αρκετές αλλαγές και προσθέσεις στον ανοικτό πηγαίο κώδικα της βιβλιοθήκης Tensorflow. Ο πηγαίος κώδικας που είχαμε στην διάθεση μας που ήταν η υλοποίηση ενός υπολειπόμενου δικτύου ακολουθούσε την ίδια αρχιτεκτονική που είχαμε αναλύσει στο υποκεφάλαιο 2.2.9 για τα υπολειπόμενα βαθιά δίκτυα. (https://github.com/lucko515/residual-network). Συγκεκριμένα το υπολειπόμενο δίκτυο είχε σαν είσοδο τα γνωστά αρχεία δεδομένων από την βάση MNIST (Modified National Institute of Standards and Technology database) τα οποία τα επεξεργάστηκε μέσω της υπολειπόμμενης μάθησης παράγοντας αρκετά ψηλά ποσοστά επιτυχίας.

Η αρχική υλοποίηση που είχαμε είναι η γνωστή κατηγοριοποίηση των χειρόγραφων αριθμών της βάσης MNIST. Η βάση δεδομένων MNIST είναι μια μεγάλη βάση δεδομένων εικόνων που απεικονίζουν τα χειρόγραφα ψηφία 0 ως 9, έχει συλλεχθεί από την αμερικάνικη υπηρεσία απογραφής και χρησιμοποιείται ευρέως για εκπαίδευση και αξιολόγηση διάφορων συστημάτων επεξεργασίας εικόνας. Αποτελείται από δύο σύνολα

εικόνων ταξινομημένων σε 10 κλάσεις, το σύνολο εκπαίδευσης με 60.000 εικόνες και τις ετικέτες τους, και το σύνολο αξιολόγησης με 10.000. Οι εικόνες έχουν διάσταση 28x28, αποτελούνται από ένα κανάλι (grayscale), έχουν κανονικοποιηθεί και κεντραριστεί με βάση το κέντρο μάζας των εικονοστοιχείων. Πρόκειται για μια βάση με πάρα πολλές χρήσεις σε δοκιμές συστημάτων μηχανικής μάθησης, ιδιαίτερα συνελικτικών νευρωνικών δικτύων και έχει αναφερθεί σε πληθώρα επιστημονικών εργασιών. Στην παρούσα εργασία επιλέχθηκε ως πρωταρχικό σύνολο εκπαίδευσης, επειδή απαιτεί ελάχιστη προεπεξεργασία, περιέχει εικόνες με σχετικά μικρή διάσταση και ένα μόνο κανάλι χρώματος (ιστοσελίδα βάσης MNIST: http://yann.lecun.com/exdb/mnist/).

Χρησιμοποιώντας λοιπόν την πιο πάνω βάση δεδομένων υλοποιήθηκε ένα υπολειπόμενο δίκτυο με βάση την προ-ενεργοποίηση που τονίστηκε γιατί είναι σημαντική στο υποκεφάλαιο 2.2.9.2. Δημιουργήθηκε ένα δίκτυο με 20 επίπεδα στα οποία γίνεται χρήση φίλτρου μεγέθους 3x3 με strides 1,1. Κινούνται 32 φίλτρα παράλληλα στα συνελικτικά επίπεδα και η συνάρτηση ενεργοποίησης που χρησιμοποιείται είναι η συνάρτηση ReLU και στο τέλος η συνάρτηση softmax. Η χρήση του max pool γίνεται μόνο ενδιάμεσα στα συνελικτικά στρώματα, ενώ δεν υπάρχει καμία χρήση του ανάμεσα στα υπολειπόμενα μπλοκς. Γίνεται χρήση και του όρου padding με αρχική τιμή ‘VALID’. Αυτές ήταν οι πρώτες δεδομένες-προκαθορισμένες (by default) τιμές των παραμέτρων στην αρχιτεκτονική του ResNet. Το προσαρμοσμένο αρχείο κώδικα φαίνεται στο Παράρτημα Β και συγκεκριμένα με το χρώμα μοβ απεικονίζονται οι αλλαγές που έχουν ούτως ώστε να προσαρμοστεί βάση των δεδομένων εισόδου για να επιλυθεί το PSSP. Επίσης στο Παράρτημα Β υπάρχει ένας πίνακας για καλύτερη κατανόηση ορισμένων συναρτήσεων και όρων που υπάρχουν στον τροποποιημένο κώδικα και για την εγκατάσταση και εκτέλεση του κώδικα, βρίσκονται βοηθητικές σημειώσεις στο Παράρτημα Α.. Να αναφέρουμε ότι η δομή της βιβλιοθήκης Tensorflow έχει την μορφή ενός γράφου όπου βάση αυτού υπάρχει η έννοια της κληρονομικότητας, προτεραιότητάς και διάφορα άλλα χαρακτηριστικά μιας object- oriented γλώσσας. Γενικά ένας γράφος δεν υπολογίζει τίποτα, απλά ορίζει τις λειτουργίες που έχουν καθοριστεί στον κώδικα.

3.2.2.2 Δεδομένα Εισόδου

Όσο αφορά τα δεδομένα εισόδου έχουμε στην διάθεση μας 10 αρχεία τα οποία εισάγονται για εκπαίδευση και το κάθε αρχείο εκπαίδευσης έχει και το αντίστοιχο αρχείο επαλήθευσης. Αρχικά έχει γινεί διαχωρισμός των τιμών βάση του κόμμα (,). Κάθε γραμμή περιείχε 301 τιμές (15*20+1=301). Για κάθε γραμμή έχει δημιουργηθεί ένας πίνακας 15x20 για πίνακας εισόδου και αντίστοιχα η ετικέτα που αντιπροσωπεύει την συγκεκριμένη γραμμή όπως αναφέρθηκε αναλυτικά στο υποκεφάλαιο 3.1. Για να προσαρμόσουμε αυτό τον 2D πίνακα στα σημεία που η υλοποίηση μας έπαιρνε τα δεδομένα εισόδου από την MNIST έπρεπε να μετατρέψουμε τον 2D πίνακα σε 4D αφού αυτή η συγκεκριμένη μορφή πίνακα υποστήριζε η βιβλιοθήκη και η συναρτήσεις που υπήρχαν μέσα. Επομένως απλά εισάγαμε ακόμη 2 διαστάσεις με την τιμή 1 όπου με αυτό τον τρόπο δεν έχει επηρεαστεί ο 2D πίνακας μας και συνεπώς είχαμε ένα 4D πίνακα να εισάγουμε στο δίκτυο.

3.2.2.3 Δεδομένα Εξόδου

Όσο αφορά τα δεδομένα εξόδου και συγκεκριμένα οι επιθυμητές εξόδου/ ετικέτες έχουν τροποποιηθεί και αυτές αφού ο κώδικας μας χρησιμοποιεί τον αλγόριθμο one hot encoding το οποίο βοηθά στην κατηγοριοποίηση των εξόδων. Συγκεκριμένα υπάρχουν 2 τιμές που μπορεί να πάρει η κατηγορία στο τελευταίο επίπεδο. Το 0 αν υπάρχει χαμηλή πιθανότητα ταύτισής και 1 αν είναι υψηλή. Για παράδειγμα με την χρήση της βάσης MNIST η έξοδος ήταν 10 στοιχεία που ένα από αυτά είχε την τιμή 1 και τα υπόλοιπα την τιμή 0. Με αυτό τον τρόπο το δίκτυο όταν δει την τιμή 1 θα συμπεράνει και την κατηγορία που βρίσκεται το συγκεκριμένο δεδομένο εισόδου (Πίνακας 3.2).

Βάσει της δυαδικής κωδικοποίησης λοιπόν έχουμε μετατρέψει τα επιθυμητά αποτελέσματα από 0 για την κατηγορία C σε 001, από 1 για την κατηγορία E σε 010 και από 2 για την κατηγορία H σε 100.

digit	0	1	2	3	4	5	6	7	8	9
4	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1
3	0	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0

Πίνακας 3.2: Παράδειγμα δυαδικής κωδικοποίησης (one hot encoding) εξόδου με 4 διαφορετικά χειρόγραφα ψηφία βάση της MNIST.

3.2.2.4 Εκπαίδευση Δικτύου

Όσο αφορά την εκπαίδευση του δικτύου χρησιμοποιήθηκε ένας σταθερός αριθμός μεγέθους στα σύνολα δεδομένων εκπαίδευσης. Αυτά τα σύνολα λέγονται batches και βοηθήσαν αρκετά στην πρόβλεψη της δευτεροταγούς δομής της πρωτεΐνης. Τι γίνεται λοιπόν με τα batches; Γνωρίζουμε ότι κατά την εκπαίδευσή ενός τεχνητού νευρωνικού δικτύου πραγματοποιείται αρχικά ένα προς τα εμπρός πέρασμα, ούτως ώστε να υπολογιστεί η πραγματική έξοδος και έπειτα με τον αλγόριθμο ανάστροφης μετάδοσης λάθους μεταφέρεται το λάθος προς τα πίσω, δηλαδή τα βάρη αλλάζουν βάση του σφάλματος που υπολογίστηκε ανάμεσα στην πραγματική και επιθυμητή έξοδο. Με την χρήση των mini-batches υπάρχει μια μικρή διαφορά κατά την εκπαίδευση του δικτύου. Η διαφορά είναι η εξής: εκτελείται ο αλγόριθμος ανάστροφής μετάδοσης σφάλματος για ένα συγκεκριμένο αριθμό δεδομένων. Ουσιαστικά τα batches είναι μια τεχνική που χωρίζει τα δεδομένα εισόδου σε μικρότερα κομμάτια, εισάγοντας το καθένα ξεχωριστά στο δίκτυο, ούτως ώστε, να υπολογίζεται ο μέσος όρος των σφαλμάτων των πραγματικών αποτελεσμάτων για το κάθε batch ξεχωριστά και βάση αυτού θα γίνουν οι αλλαγές των βαρών για το κάθε batch κατά το προς τα πίσω πέρασμα του δικτύου. Η χρήση των batches για την εκπαίδευση ενός νευρωνικού δικτύου βοήθα αρκετά παράγοντας και πιο υψηλά ποσοστά επιτυχίας. Συγκεκριμένα εκτελείται λιγότερες φορές ο αλγόριθμος ανάστροφης μετάδοσης σφάλματος αφού λαμβάνεται υπόψη ο μέσος όρος και έπειτα εκτελείται. Επίσης λόγω του ότι λαμβάνεται υπόψη ο μέσος όρος

όλων των δεδομένων που υπάρχουν σε ένα batch δίνει στο δίκτυο περισσότερη πληροφορία με αποτέλεσμα να το δίνει και την δυνατότητα να μαθαίνει πιο αποδοτικά.

Κεφάλαιο 4

Πειράματα και Αποτελέσματα

4.1 Αποτελέσματα βάσης δεδομένων MNIST 85

4.2 10-fold Cross-validation 87

4.3 Πειράματα βελτιστοποίησης υπερπαραμέτρων του δικτύου 89

4.3.1 Χρήση Διαφορετικού Αριθμού Κρυφών Επιπέδων 89

4.3.2 Χρήση Διαφορετικών Μεγεθών Φίλτρου 93

4.3.3 Χρήση Διαφορετικού Αριθμού Παράλληλων Φίλτρων 96

4.3.4 Χρήση Διαφορετικής Τιμής Γεμίσματος 99

4.3.5 Χρήση Διαφορετικού Stride 102

4.3.6 Χρήση Διαφορετικού ρυθμού εκμάθησης 105

4.3.7 Χρήση αρχείου δεδομένου ονομασίας ‘fold8’ 108

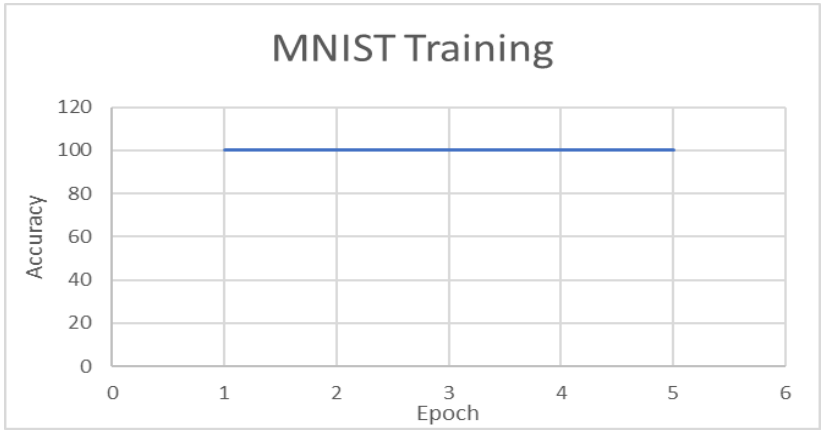
4.3.8 Χρήση max pooling layers σε συνδυασμό το αρχείο δεδομένων ονομασίας ‘fold8’ 111

4.1 Αποτελέσματα βάσης δεδομένων MNIST

Όπως έχει αναφερθεί στο προηγούμενο κεφάλαιο έχει γίνει χρήστη της βάσης δεδομένων MNIST αρχικά για να δούμε ότι όντως η βιβλιοθήκη, ειδικότερα το υπολειπόμενο δίκτυο, παράγει ικανοποιητικά ποσοστά για να μπορεί να χρησιμοποιηθεί για το πρόβλημα που προσπαθούμε να επιλύσουμε σε αυτή την έρευνα, το πρόβλημα πρόβλεψης δευτεροταγούς δομής της πρωτεΐνης.

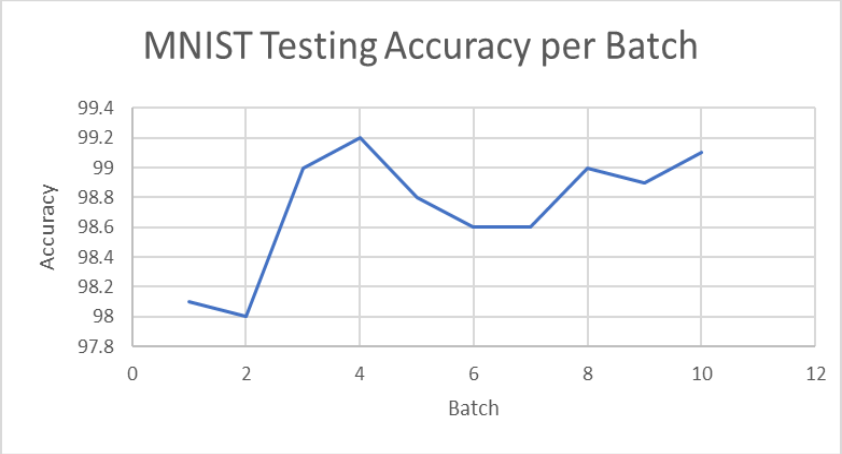
Η βάση δεδομένων MNIST επιλέχθηκε ως σημείο αναφοράς σχετικά με την διαδικασία εισαγωγής δεδομένων εισόδου, διότι χρησιμοποιείται ευρέως για εκπαίδευση και δοκιμές στον τομέα της μηχανικής μάθησης και συγκεκριμένα για την εκπαίδευση συστημάτων επεξεργασίας εικόνας. Συνεπώς αφού ασχολείται με επεξεργασία εικόνας, αυτό την καθιστά κατάλληλη βάση δεδομένων, αφού στην ουσία χρησιμοποιούμε ένα συνελικτικό δίκτυο το οποίο επεξεργάζεται συγκεκριμένες εικόνες.

Σύμφωνά με τις προκαθορισμένες τιμές των υπερπαραμέτρων του δικτύου που καταγραφτήκαν στο υποκεφάλαιο 3.2.2.1 με εποχές ίσο με 5 υπήρξαν τα πιο κάτω αποτελέσματα. Όταν λέμε εποχές ίσο με 5 εννοούμε τον αριθμό φορών που το δίκτυο κατά την εκπαίδευση του θα περάσει από ολόκληρό το αρχείο δεδομένων εισόδου εκπαίδευσης. Στην συγκεκριμένη περίπτωση θα γινούν 5 πλήρη περάσματα των δεδομένων εκπαίδευσης στο υπολειπόμενο δίκτυο.



Γραφική Παράσταση 4.1: Ακρίβεια Εκπαίδευσης - Εποχή με τις προκαθορισμένες τιμές του νευρωνικού δικτύου.

Γραφική Παράσταση 4.2: Σφάλμα Εκπαίδευσης - Εποχή με τις προκαθορισμένες τιμές του νευρωνικού δικτύου.

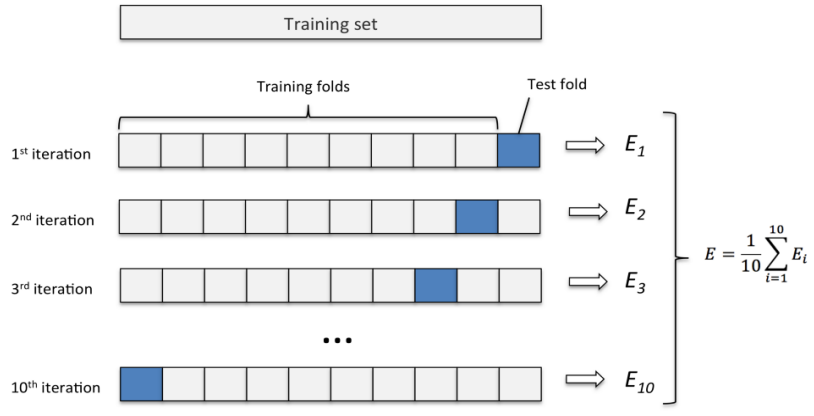


Γραφική Παράσταση 4.3: Ακρίβεια Επαλήθευσης – Εποχή ανά batch με τις προκαθορισμένες τιμές του νευρωνικού δικτύου.

Από τις πιο πάνω γραφικές, παρατηρούμε ότι το δίκτυο μαθαίνει αφού το σφάλμα εκπαίδευσης μειώνεται αρκετά και η ακρίβεια εκπαίδευσης είναι 100%, αλλά το πιο σημαντικό είναι ότι η ακρίβεια επαλήθευσης, που βάση αυτής θα συμπεράνουμε ότι το δίκτυο μαθαίνει, κυμαίνεται στο 98-99.2%. Αυτό το ποσοστό ακρίβειας είναι αρκετά ψηλό και μπορούμε να πούμε ότι το δίκτυο μαθαίνει με την εισαγωγή των συγκεκριμένων εικόνων της βάσης δεδομένων MNIST και μπορούμε να προχωρήσουμε στην εισαγωγή των δικών μας δεδομένων εισόδου.

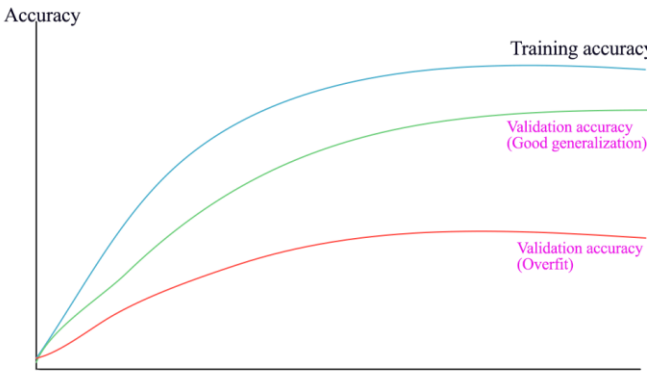
4.2 10-fold Cross-validation

Πριν προχωρήσουμε στην ανάλυση των αποτελεσμάτων από τα πειράματα που είχαν γινεί να τονίσουμε ότι χρησιμοποιήθηκε η τεχνική της διασταυρωμένης επικύρωσης (cross validation). Η διασταυρωμένη επικύρωση είναι μια από τις τεχνικές επικύρωσης που αξιολογούν τον τρόπο γενίκευσης των αποτελεσμάτων που θα έχει το δίκτυο όταν χρησιμοποιήσει σαν είσοδο ένα άγνωστο σύνολο δεδομένων. Συγκεκριμένα στην k-fold cross-validation χωρίζει το αρχικό αρχείο δεδομένων σε τυχαία k υποδείγματα ίσου μεγέθους. Από αυτά τα k υποδείγματα, επιλέγεται 1 ως αρχείο εισαγωγής στην διαδικασία επαλήθευσης, άρα ονομάζεται και σετ επαλήθευση (testing set) και τα υπόλοιπα k-1 υποδείγματα θα χρησιμοποιηθούν για την διαδικασία εκπαίδευσης και αποκαλούνται δεδομένα εκπαίδευσης (training data). Η διαδικασία αυτή επαναλαμβάνεται k φορές, με αποτέλεσμα το κάθε υπόδειγμα να χρησιμοποιείται μια φορά ως δεδομένο επαλήθευσης. Για παράδειγμα η πιο συνηθισμένη τεχνική στην διαδικασία αυτή είναι η 10-fold cross-validation όπου k=10. Συνεπώς το αρχικό αρχείο μας θα διαχωριστεί σε 10 μικρά αρχεία τα οποία τα 9 θα χρησιμοποιηθούν ως δεδομένα εκπαίδευσης και το υπόλοιπο 1 θα χρησιμοποιηθεί ως δεδομένο επαλήθευσης/επικύρωσης. Όταν όλα τα μικρά αρχεία/ υποδείγματα περάσουν από την διαδικασία επαλήθευσης θα παράξει το καθένα ένα ποσοστό κατά πόσο έχει γινεί σωστή η πρόβλεψη βάση τα δεδομένα. Άρα θα έχουμε 10 διαφορετικά ποσοστά. Όταν ολοκληρωθεί όλη η διαδικασία αυτή το συνολικό ποσοστό θα είναι ο μέσος όρος των ποσοστών που έχει παράξει το κάθε υπόδειγμα (Σχήμα 4.1).



Σχήμα 4.1: Η τεχνική διασταυρωμένη επικύρωση (cross validation) με κ=10 και το μπλε κουτί αντιπροσωπεύει το υπόδειγμα στην συγκεκριμένη φορά ως δεδομένα επαλήθευσης. Πάρθηκε από: http://karlrosaen.com/ml/learning-log/2016-06-20/

Με αυτό τον τρόπο λοιπόν όλα τα υποδείγματα περνούν και από την διαδικασία εκπαίδευσης και από την διαδικασία επαλήθευσης με συνέπεια να αντιμετωπίζεται το πρόβλημα της υπερφόρτωσης (overfitting). Δηλαδή το δίκτυο θα μάθει, στην διαδικασία εκπαίδευσης αλλά και στην διαδικασία επαλήθευσης αφού τα δεδομένα επαλήθευσης δεν θα του είναι εντελώς άγνωστα με αποτέλεσμα να μην υπάρχει κάποια πτώση στα ποσοστά ακρίβειας στην διαδικασία επαλήθευσης αλλά να αυξάνονται όπως στην διαδικασία εκπαίδευσης (Σχήμα 4.2). Αυτός ο τρόπος λοιπόν θα βοηθήσει αρκετά κατά την αξιολόγηση των διαφορετικών υπερπαραμέτρων εξάγοντας πιο ακριβέστερη εκτίμηση της απόδοσης πρόβλεψης του δίκτυού.



Σχήμα 4.2: Σύγκριση ακρίβειας εκπαίδευσης με 2 διαφορετικές ακρίβειες επαλήθευσης. Η ακρίβεια εκπαίδευσης είναι με μπλε χρώμα και η ακρίβεια επαλήθευσης στην πρώτη περίπτωση είναι με πράσινο χρώμα που βάσει της γραφικής παράστασης δεν έχει απότομη πτώση με αποτέλεσμα το δίκτυο να έχει γενικεύσει σωστά βάση τα άγνωστα δεδομένα που του παρουσιαστήκαν. Η άλλη περίπτωση είναι με κόκκινο χρώμα η ακρίβεια επαλήθευσης η οποία έχει μειωθεί αρκετά με αποτέλεσμα να εμφανιστεί το πρόβλημα της υπερφόρτωσης.
Πάρθηκε από: https://medium.com/@jonathan_hui/improve-deep-learning-models-performance-network-tuning-part-6-29bf90df6d2d

Αυτή η τεχνική επικύρωσης θα χρησιμοποιηθεί στο σύνολο δεδομένων CB513 για την επίλυση του συγκεκριμένου προβλήματος που μελετούμε σε αυτή την έρευνα. Πιο συγκεκριμένα ο Διονυσίου, έχει χωρίσει αυτό το σύνολο δεδομένων σε 10 υποδείγματα (folds) όπου τα 9 τα έχει καταχωρήσει σε ένα αρχείο με ονομασία protein_csv_train_all_15neighbors_technique_fold**0** και το υπόλοιπό 1 υπόδειγμα στο αρχείο protein_csv_test_all_15neighbors_technique_fold0. Αυτή η διαδικασία έχει γινεί

ακόμη 9 φορές τοποθετώντας το κάθε υπόδειγμα στο αρχείο επαλήθευσής μια φορά, έχοντας στην διάθεση μας τώρα 10 αρχεία protein_csv_ train_all_15neighbors_technique_fold**k** όπου k=0-9 με το αντίστοιχο protein_csv_test_all_15neighbors_technique_fold**k** όπου k=0-9.

Άρα είχαμε στην διάθεση μας 10 αρχεία για εκπαίδευση και τα 10 αντίστοιχα τους αρχεία για επαλήθευση. Πριν όμως εφαρμοστεί η 10-fold cross-validation υπήρξαν αρκετά πειράματα με διάφορες αλλαγές στις τιμές των υπερπαραμέτρων του δικτύου με στόχο όσο το δυνατό καλύτερη πρόβλεψη της δευτεροταγούς δομής της πρωτεΐνης, δηλαδή όσο το δυνατό βέλτιστα και υψηλότερα ποσοστά ακρίβειας και συγκεκριμένα Q3 ποσοστά επιτυχίας.

4.3 Πειράματα βελτιστοποίησης υπερπαραμέτρων του δικτύου

Η βελτιστοποίηση των υπερπαραμέτρων έχει γινεί σε ένα συγκεκριμένο αρχείο με ονομασία fold1 και έπειτα αφού βρεθούν οι βέλτιστες τιμές των υπερπαραμέτρων θα εφαρμοστούν και στα υπόλοιπα αρχεία. Κάποιες παράμετροι που επιλέξαμε για να βελτιστοποιήσουμε τα αποτελέσματα ήταν ο αριθμός των κρυφών νευρώνων, το μέγεθος του φίλτρου που εφαρμόστηκε κατά την πράξη συνέλιξης στα συνελικτικά στρώματα, ο αριθμός των παράλληλων φίλτρων που θα περνούσαν όλα από την εικόνα εισόδου, η τιμή του γεμίσματος (valid or same padding), πόσα εικονοστοιχεία θα μετακινείται το φίλτρο πάνω στην εικόνα οριζοντίως και καθέτως (η τιμή του stride) και η τιμή του ρυθμού εκμάθησης που έχει αναλυθεί στο υποκεφάλαιο 2.2.7.1 (learning rate).

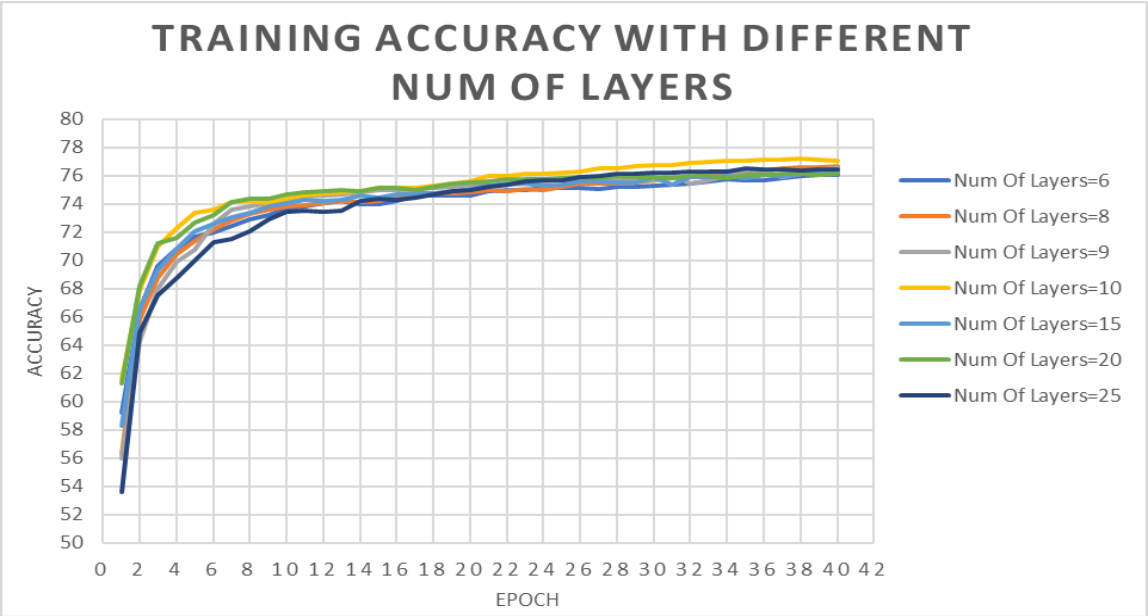
4.3.1 Χρήση Διαφορετικού Αριθμού Συνελικτικών Επιπέδων

Τα πρώτα πειράματα που είχαν γινεί ήταν με την αλλαγή τιμής της πιο σημαντικής παραμέτρους για βελτιστοποίηση των ποσοστών επιτυχίας. Η παράμετρός αυτή είναι ο αριθμός των συνελικτικών επιπέδων του δικτύου. Γνωρίζουμε από τα πιο πάνω κεφάλαια ότι χρησιμοποιήθηκε η βαθιά μάθηση για την επίλυση του προβλήματος

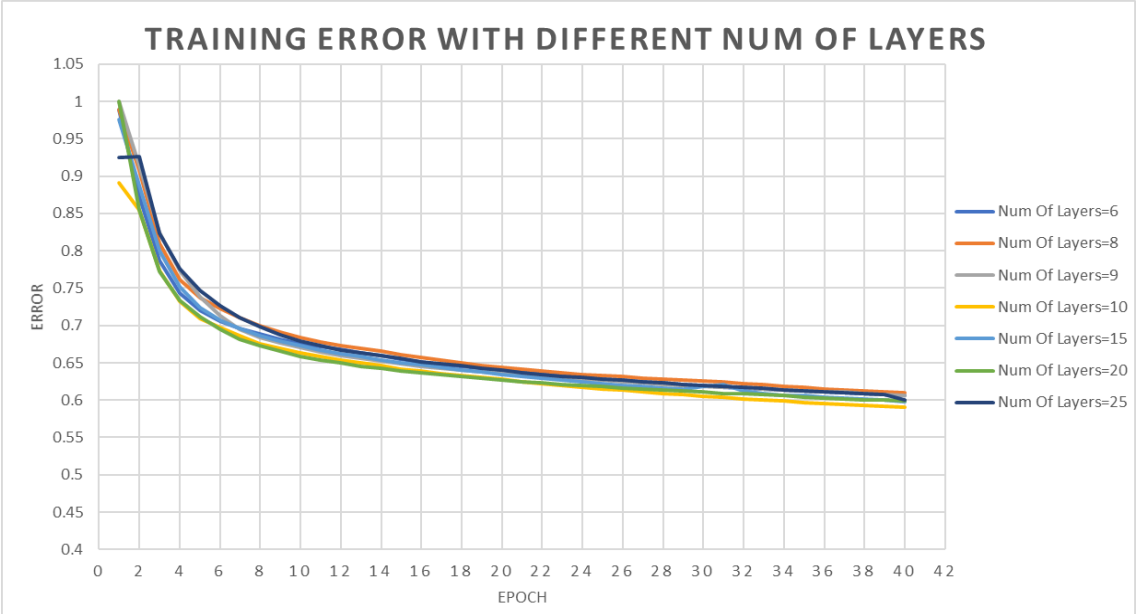
PSSP, συνεπώς θα υπάρχει περισσότερο από ένα επίπεδο στο δίκτυο μας. Συγκεκριμένα χρησιμοποιήσαμε πολύ περισσότερα επίπεδα λόγω του ότι θέλουμε να επεξεργαστούμε περισσότερα περίπλοκα χαρακτηριστικά των δεδομένων εισόδου αλλά και στο ότι χρησιμοπείται ένα υπολειπόμενο δίκτυο, που αυτό σημαίνει ότι χρειάζεται περισσότερο βάθος για να υπάρξουν οι επιπρόσθετες συνδέσεις παραλήψεις στο δίκτυο μας και στο ότι αυτό το δίκτυο μπορεί να έχει ένα μεγάλο βάθος συντηρώντας την ακρίβεια πρόβλεψης του. Για να εξετάσουμε ότι όντως ισχύουν οι θεωρίες του ResNet είχαμε αφήσει σταθερές όλες τις προκαθορισμένες τιμές των παραμέτρων, οι οποίες φαίνονται στον πίνακα πιο κάτω (Πίνακας 4.1), και αλλάζαμε μόνο την τιμή του αριθμού που αντιπροσώπευε τα συνελικτικά επίπεδα. Οι τιμές που δώσαμε στην συγκεκριμένη παράμετρο ήταν από το 6 μέχρι το 25. Χρησιμοποιηθήκαν αρχικά μικρές τιμές λόγω μεγάλου χρόνου εκπαίδευσης που απαιτούσε το δίκτυο και στην μικρή διαθέσιμη μνήμη του υπολογιστή που τρέχαμε τα πειράματα, αλλά μετά λόγω πρόσθεσης μνήμης στον υπολογιστή μπορέσαμε να εισάγουμε μεγαλύτερες τιμές στην παράμετρο με λιγότερο χρόνο εκπαίδευσης και εξάγοντας περισσότερα αποτελέσματα για σύγκριση.

Παράμετροί ρύθμισης	Τιμή
Εποχές	40
Μέγεθος batch δεδομένων εκπαίδευσης	7000
Μέγεθος batch δεδομένων επαλήθευσης	7289
Αριθμός batches δεδομένων εκπαίδευσης	12
Αριθμός batches δεδομένων επαλήθευσης	1
Συνάρτηση ενεργοποίησης	ReLU
Ρυθμός μάθησης	0.002
Μέγεθος φίλτρου	3
Αριθμός συνελικτικών επιπέδων	20 (μεταβαλλόμενη)
Γέμισμα	Valid
Αριθμός Παράλληλων φίλτρων	32
Αριθμός βημάτων φίλτρου (stride)	1,1
Μέθοδος Ενημέρωσης του Ρυθμού Μάθησης (Updater)	Adam Optimizer
Max Pooling layers	True

Πίνακας 4.1: Οι προκαθορισμένες τιμές των παραμέτρων που έχει το υπολειπόμενο δίκτυο.

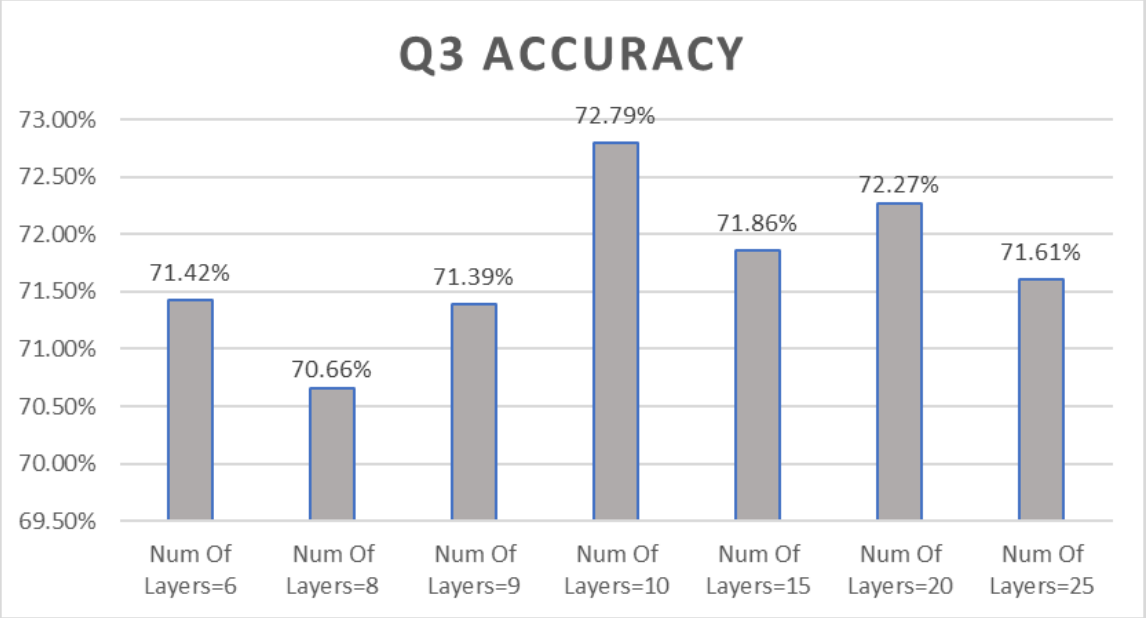


Γραφική Παράσταση 4.4: Ακρίβεια Εκπαίδευσης - Εποχή για κάθε νευρωνικό δίκτυο με διαφορετικό αριθμό συνελικτικών επιπέδων.



Γραφική Παράσταση 4.5: Σφάλμα εκπαίδευσης – εποχής για κάθε νευρωνικό δίκτυο με διαφορετικό αριθμό συνελικτικών επιπέδων.

Από τις πιο πάνω γραφικές παραστάσεις (Γραφική Παράσταση 4.5 και 4.5) συμπεραίνουμε ότι το δίκτυο εκπαιδεύεται σωστά αφού η ακρίβεια αυξάνεται και το σφάλμα μειώνεται ανά εποχή. Παρατηρείται ότι το δίκτυο με αριθμό συνελικτικών επιπέδων ίσο με 10 έχει την υψηλότερη ακρίβεια και το χαμηλότερο σφάλμα σε σύγκριση με τα υπόλοιπα δίκτυα. Όμως για να συμπεράνουμε ότι όντως το δίκτυο μαθαίνει και μπορεί να προβλέψει την δευτεροταγούς δομή της πρωτεΐνης έπρεπε να συγκρίνουμε τα ποσοστά επιτυχίας κατά την διαδικασία επαλήθευσης.



Γραφική Παράσταση 4.6: Ποσοστά επιτυχίας Q3 για κάθε νευρωνικό δίκτυο με διαφορετικό αριθμό συνελικτικών επιπέδων.

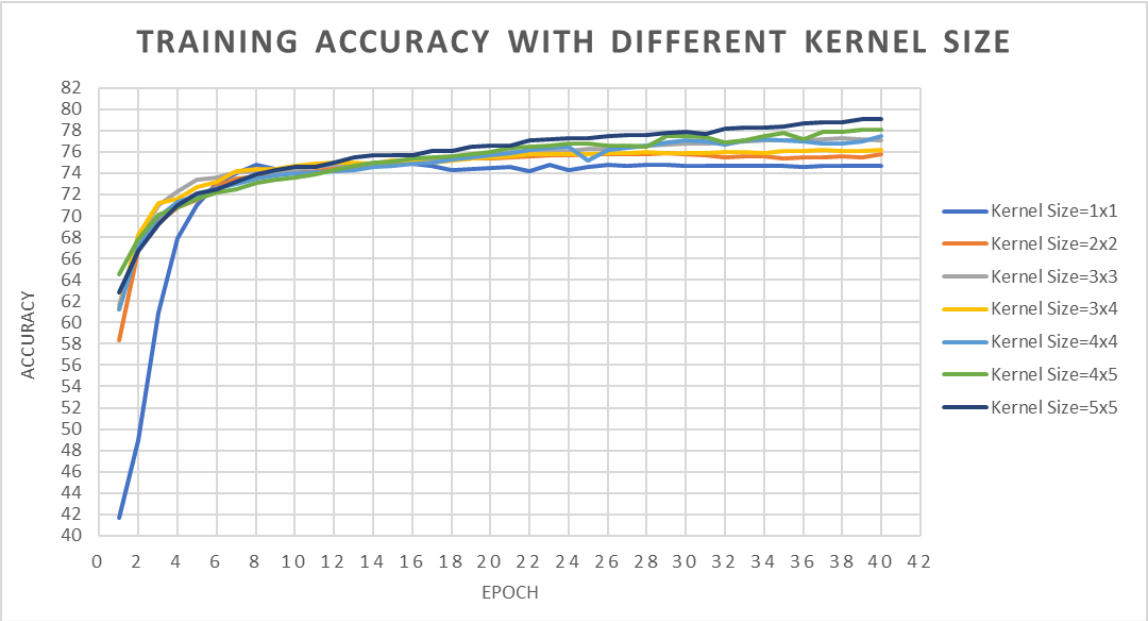
Σύμφωνα με την πιο πάνω γραφική παράσταση φαίνεται ότι όντος το δίκτυο έχει εκπαιδευτεί και μπορεί να προβλέψει την δευτεροταγή δομή της πρωτεΐνης βάση αγνώστων δεδομένων. Το πιο ψηλό ποσοστό ακρίβειας Q3 είναι του δικτύου με αριθμό συνελικτικών επιπέδων ίσο με 10 το οποίο είναι λογικό αφού είχε και το χαμηλότερο σφάλμα εκπαίδευσης. Επομένως λάβαμε υπόψη ότι ο αριθμός των συνελικτικών επιπέδων θα είναι ίσος με 10 για να προχωρήσουμε έπειτα στις αλλαγές των υπόλοιπων παραμέτρων που έχει το δίκτυο.

4.3.2 Χρήση Διαφορετικών Μεγεθών Φίλτρου

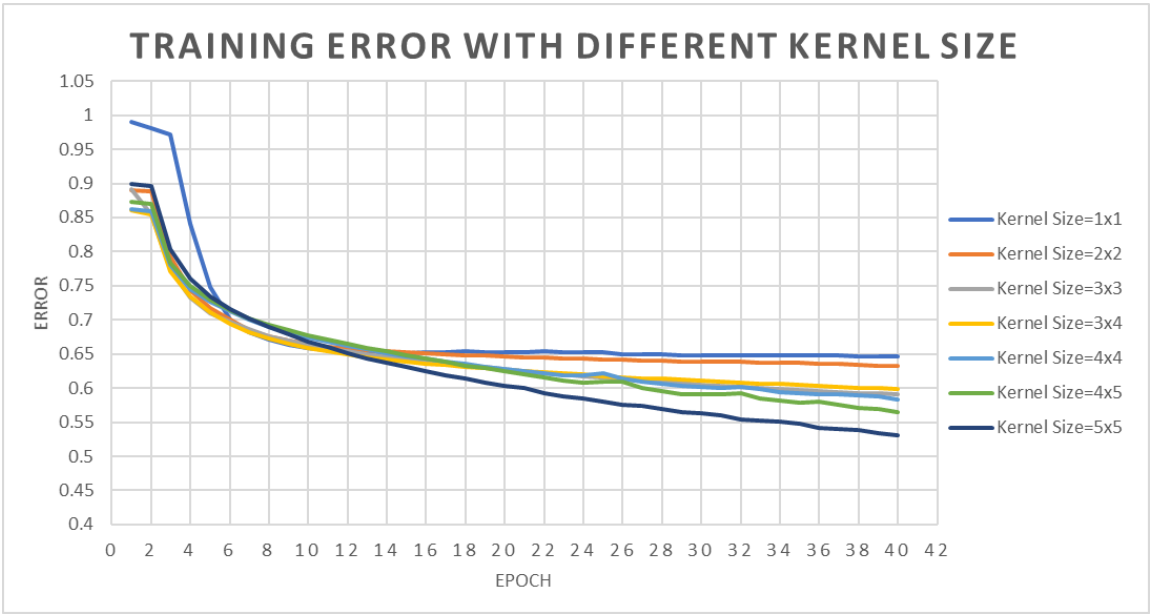
Μια άλλη παράμετρος που έχει σημαντικό ρόλο στην μάθηση του δίκτυού είναι το μέγεθος του φίλτρου που θα χρησιμοποιηθεί στα συνελικτικά επίπεδα για την πράξη της συνέλιξης και συνεπώς την εξαγωγή σημαντικών χαρακτηριστικών που έχουν τα δεδομένα εισόδου. Αναλόγως με το μέγεθος του φίλτρου λαμβάνονται υπόψη οι τιμές τις εισόδου. Δηλαδή για ένα δίκτυο με φίλτρο μεγέθους ίσο με 4 θα παίρνει 4-4 τιμές των δεδομένων εισόδου και θα υπολογίζεται η πράξη συνέλιξης. Λαμβάνοντας υπόψη λοιπόν ότι ο αριθμός των συνελικτικών επιπέδων είναι ίσος με 10 και τις προκαθορισμένες τιμές για τις υπόλοιπες παραμέτρους, αλλάξαμε την τιμή του μεγέθους του φίλτρου σε πολλές διαφορετικές τιμές για να αποφασίσουμε ποια από αυτές έχει το μεγαλύτερο ποσοστό επιτυχίας Q3.

Παράμετροί ρύθμισης	Τιμή
Εποχές	40
Μέγεθος batch δεδομένων εκπαίδευσης	7000
Μέγεθος batch δεδομένων επαλήθευσης	7289
Αριθμός batches δεδομένων εκπαίδευσης	12
Αριθμός batches δεδομένων επαλήθευσης	1
Συνάρτηση ενεργοποίησης	ReLU
Ρυθμός μάθησης	0.002
Μέγεθος φίλτρου	3 (μεταβαλλόμενη)
Αριθμός συνελικτικών επιπέδων	10
Γέμισμα	Valid
Αριθμός Παράλληλων φίλτρων	32
Αριθμός βημάτων φίλτρου (stride)	1,1
Μέθοδος Ενημέρωσης του Ρυθμού Μάθησης (Updater)	Adam Optimizer
Max Pooling layers	True

Πίνακας 4.2: Οι προκαθορισμένες τιμές των παραμέτρων που έχει το υπολειπόμενο δίκτυο με την νέα τιμή του αριθμού των κρυφών επιπέδων.

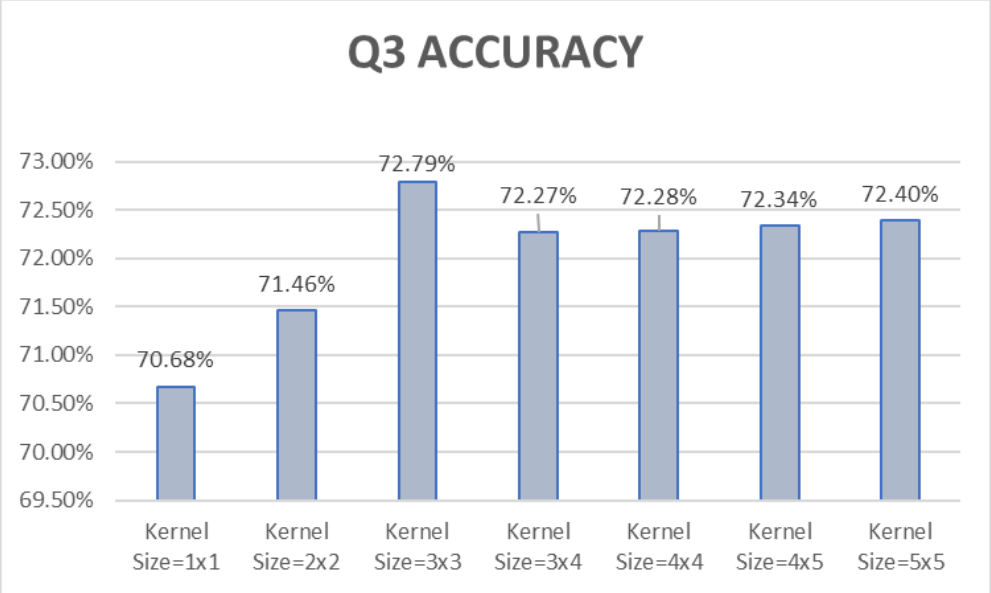


Γραφική Παράσταση 4.7: Ακρίβεια Εκπαίδευσης - Εποχή για κάθε νευρωνικό δίκτυο με διαφορετικό μέγεθος φίλτρου.



Γραφική Παράσταση 4.8: Σφάλμα εκπαίδευσης – εποχής για κάθε νευρωνικό δίκτυο με διαφορετικό μέγεθος φίλτρου.

Από τις πιο πάνω γραφικές παραστάσεις (Γραφική Παράσταση 4.7 και 4.8) συμπεραίνουμε ότι το δίκτυο εκπαιδεύεται σωστά αφού η ακρίβεια αυξάνεται και το σφάλμα μειώνεται ανά εποχή. Παρατηρείται ότι το δίκτυο με μέγεθος φίλτρου ίσο με 5 έχει την υψηλότερη ακρίβεια και το χαμηλότερο σφάλμα σε σύγκριση με τα υπόλοιπα δίκτυα, πράγμα λογικό αφού παίρνει και την περισσότερη πληροφορία από τα δεδομένα εισόδου κα την επεξεργάζεται. Όμως σύμφωνα με την πιο κάτω γραφική παράσταση (Γραφική Παράσταση 4.9) φαίνεται ότι το πιο ψηλό ποσοστό ακρίβειας Q3 είναι του δικτύου με μέγεθος φίλτρου ίσο με 3 το οποίο σχετικά είχε ένα από τα χαμηλότερο σφάλμα εκπαίδευσης. Συγκεκριμένα το δίκτυο που περιείχε το φίλτρο με μέγεθος ίσο με 3 είχε 77.04286 για ακρίβεια και 0.5908 ως σφάλμα στην διαδικασία εκπαίδευσης ενώ το δίκτυο με φίλτρο μεγέθους ίσο με 5 είχε 79.07143 ακρίβεια και 0.53082 σφάλμα κατά την εκπαίδευση του. Υπάρχει μια διαφορά αλλά βάση της διαδικασίας επαλήθευσης το δίκτυο μαθαίνει καλύτερα με φίλτρο μεγέθους ίσο με 3. Όσο αφορά τα υπόλοιπα αποτελέσματα δεν είχαν μεγάλη διαφορά τιμής εκτός από το δίκτυο με φίλτρο μεγέθους ίσο με 1. Αυτό οφείλεται στο γεγονός ότι το φίλτρο θα παίρνει μια-μια τιμή εισόδου για να υπολογιστεί η πράξη συνέλιξης που αυτό θα έχει ως συνέπεια να μην ληφθούν υπόψη τα γειτονικά αμινοξέα και οι αλληλεπιδράσεις που θα έχουν μεταξύ τους και συνεπώς δεν θα προβλέψει σωστά την κατηγορία του συγκεκριμένου αμινοξύ που είχε μελετήσει.



Γραφική Παράσταση 4.9: Ποσοστά επιτυχίας Q3 για κάθε νευρωνικό δίκτυο με διαφορετικό μέγεθος φίλτρου.

Επομένως λάβαμε υπόψη ότι το μέγεθος του φίλτρου θα είναι ίσος με 3, αφού αυτό δίνει και το πιο ψηλό ποσοστό επιτυχίας Q3, για να προχωρήσουμε έπειτα στις αλλαγές των υπόλοιπων παραμέτρων που έχει το δίκτυο.

4.3.3 Χρήση Διαφορετικού Αριθμού Παράλληλων Φίλτρων

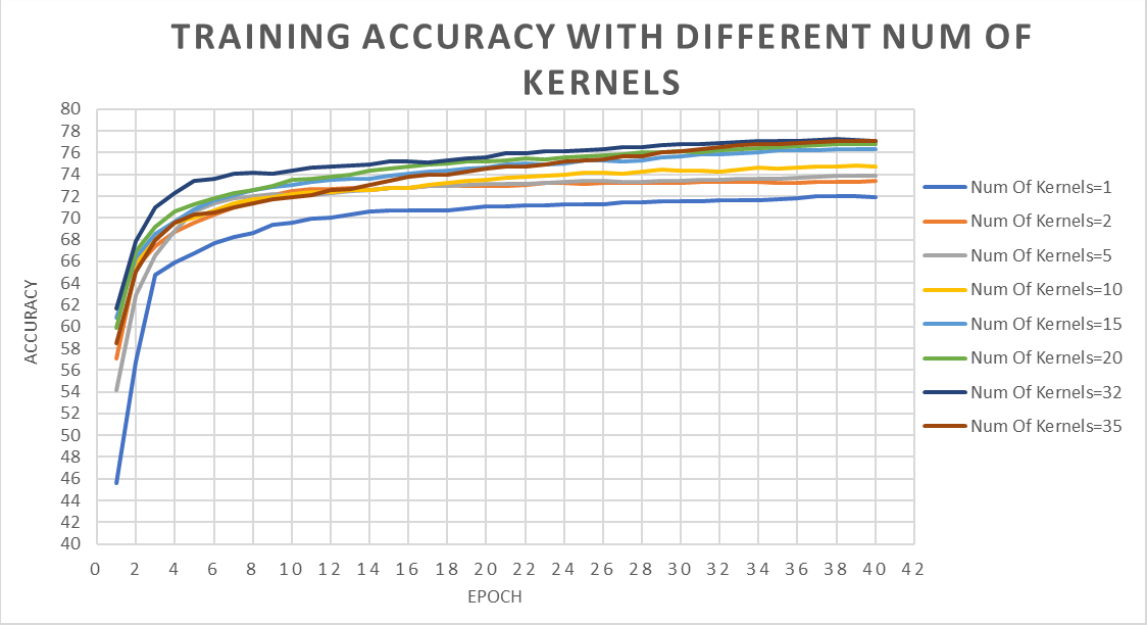
Όσο αφορά τον αριθμό των παράλληλων φίλτρων σε ένα συνελικτικό δίκτυο γνωρίζουμε από το υποκεφάλαιο 2.2.8 ότι έχει σημαντικό ρόλο στην πρόβλεψη ενός προβλήματος. Συγκεκριμένα ένα φίλτρο με την πράξη της συνέλιξης εντοπίζει στην ουσία ένα συγκεκριμένο χαρακτηριστικό στα δεδομένα εισόδου. Όσο περισσότερα φίλτρα έχουμε τόσο πιο πολλά διαφορετικά χαρακτηριστικά θα εντοπίσει το δίκτυο μας με αποτέλεσμα μια πιο αποδοτική πρόβλεψη. Όμως όσο αυξάνουμε τα φίλτρα τόσο περισσότερος χρόνος και μνήμη απαιτείται για την εκπαίδευση του δικτύου. Αφού είχαμε το κατάλληλο εξοπλισμό δώσαμε όσο πιο μεγάλες τιμές σε αυτή την παράμετρο ούτως ώστε να μπορεί το δίκτυο μας να επεξεργαστεί όσο το δυνατό περισσότερες αλληλεπιδράσεις των αμινοξέων με αποτέλεσμα την πιο βέλτιστή κατηγοριοποίηση των αμινοξέων.

Επομένως όπως και στα προηγούμενα πειράματα κρατήσαμε σταθερές τις γνωστές παραμέτρους που έχουν προκαθοριστεί στο δίκτυο μας, και ορισμένες που έχουν αλλάξει τιμή λόγω πιο αποδοτικών αποτελεσμάτων από πειράματα (Πίνακας 4.3), αλλάζοντας σε αυτή την περίπτωση μόνο την τιμή των παράλληλων φίλτρων εξάγοντας τις πιο κάτω γραφικές παραστάσεις.

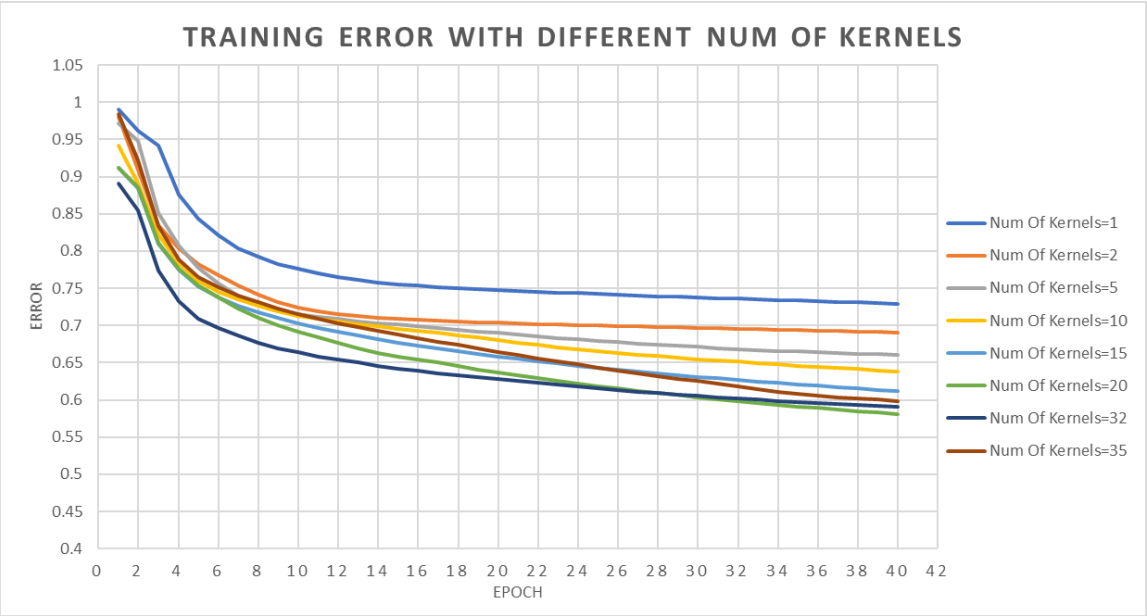
Παράμετροί ρύθμισης	Τιμή
Εποχές	40
Μέγεθος batch δεδομένων εκπαίδευσης	7000
Μέγεθος batch δεδομένων επαλήθευσης	7289
Αριθμός batches δεδομένων εκπαίδευσης	12
Αριθμός batches δεδομένων επαλήθευσης	1
Συνάρτηση ενεργοποίησης	ReLU

Ρυθμός μάθησης	0.002
Μέγεθος φίλτρου	3
Αριθμός συνελικτικών επιπέδων	10
Γέμισμα	Valid
Αριθμός Παράλληλων φίλτρων	32 (μεταβαλλόμενη)
Αριθμός βημάτων φίλτρου (stride)	1,1
Μέθοδος Ενημέρωσης του Ρυθμού Μάθησης (Updater)	Adam Optimizer
Max Pooling layers	True

Πίνακας 4.3: Οι προκαθορισμένες τιμές των παραμέτρων που έχει το υπολειπόμενο δίκτυο με την νέα τιμή του μεγέθους του φίλτρου.

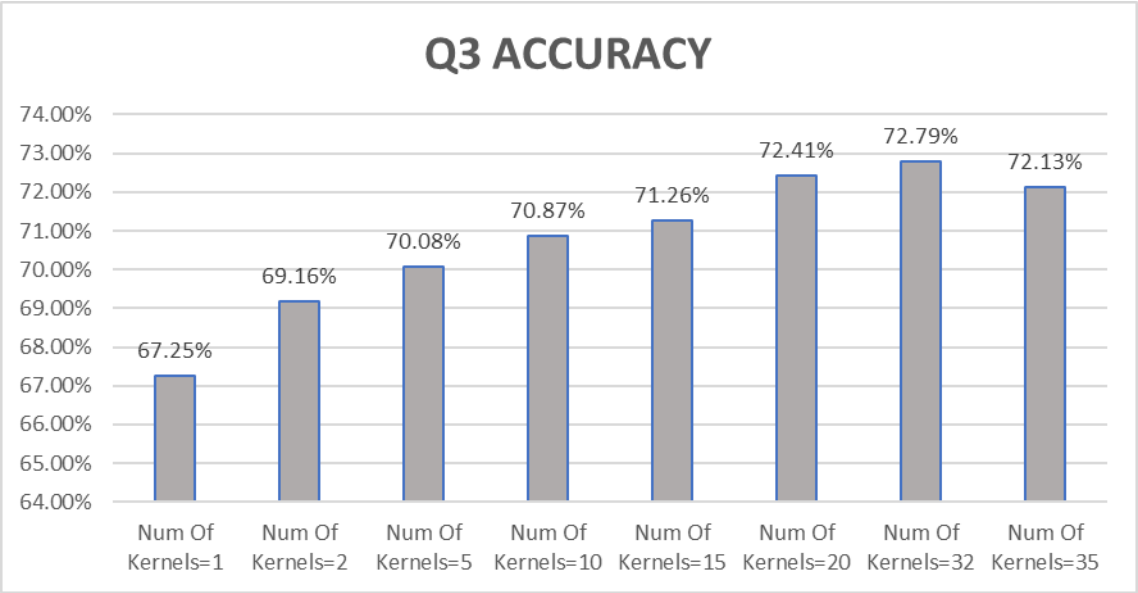


Γραφική Παράσταση 4.10: Ακρίβεια Εκπαίδευσης - Εποχή για κάθε νευρωνικό δίκτυο με διαφορετικό αριθμό παράλληλων φίλτρων.



Γραφική Παράσταση 4.11: Σφάλμα εκπαίδευσης – εποχής για κάθε νευρωνικό δίκτυο με διαφορετικό αριθμό παράλληλων φίλτρων.

Από τις πιο πάνω γραφικές παραστάσεις (Γραφική Παράσταση 4.10 και 4.11) συμπεραίνουμε και εδώ ότι το δίκτυο εκπαιδεύεται σωστά αφού η ακρίβεια εκπαίδευσης αυξάνεται και το σφάλμα εκπαίδευσης μειώνεται ανά εποχή. Παρατηρείται ότι τα δίκτυα με αριθμό παράλληλων φίλτρων ίσο με 20, 32 και 35 έχουν τις πιο υψηλές τιμές στην ακρίβεια και τις πιο χαμηλές τιμές στο σφάλμα εκπαίδευσης σε σύγκριση με τα υπόλοιπα δίκτυα. Όμως για να συμπεράνουμε ότι όντως το δίκτυο μαθαίνει και μπορεί να προβλέψει την δευτεροταγή δομή της πρωτεΐνης έπρεπε να συγκρίνουμε τα ποσοστά επιτυχίας κατά την διαδικασία επαλήθευσης. Σύμφωνα με την πιο κάτω γραφική παράσταση (Γραφική Παράσταση 4.12) παρατηρείται ότι τα δίκτυα με αριθμό παράλληλων φίλτρων ίσο με 20, 32 και 35 κατάφεραν να πετύχουν τα υψηλότερα ποσοστά το οποίο ήταν αναμενόμενο αφού είχαν τις υψηλότερες τιμές στην γραφική ακρίβειας εκπαίδευσης-εποχής και τις χαμηλότερες τιμές στην γραφική σφάλμα εκπαίδευσης-εποχή. Το υψηλότερο ποσοστό ακρίβειας Q3 από τα 3 αυτά δίκτυα είναι του δικτύου με αριθμό παράλληλο φίλτρων ίσο με 32. Και βάση αυτού ορίσαμε ως τελική τιμή του αριθμού των παράλληλων φίλτρων ίση με 32 για να προχωρήσουμε στις επόμενες αλλαγές των υπόλοιπων παραμέτρων που έχει το δίκτυο.



Γραφική Παράσταση 4.12: Ποσοστά επιτυχίας Q3 για κάθε νευρωνικό δίκτυο με διαφορετικό αριθμό παράλληλων φίλτρων.

4.3.4 Χρήση Διαφορετικής Τιμής Γεμίσματος

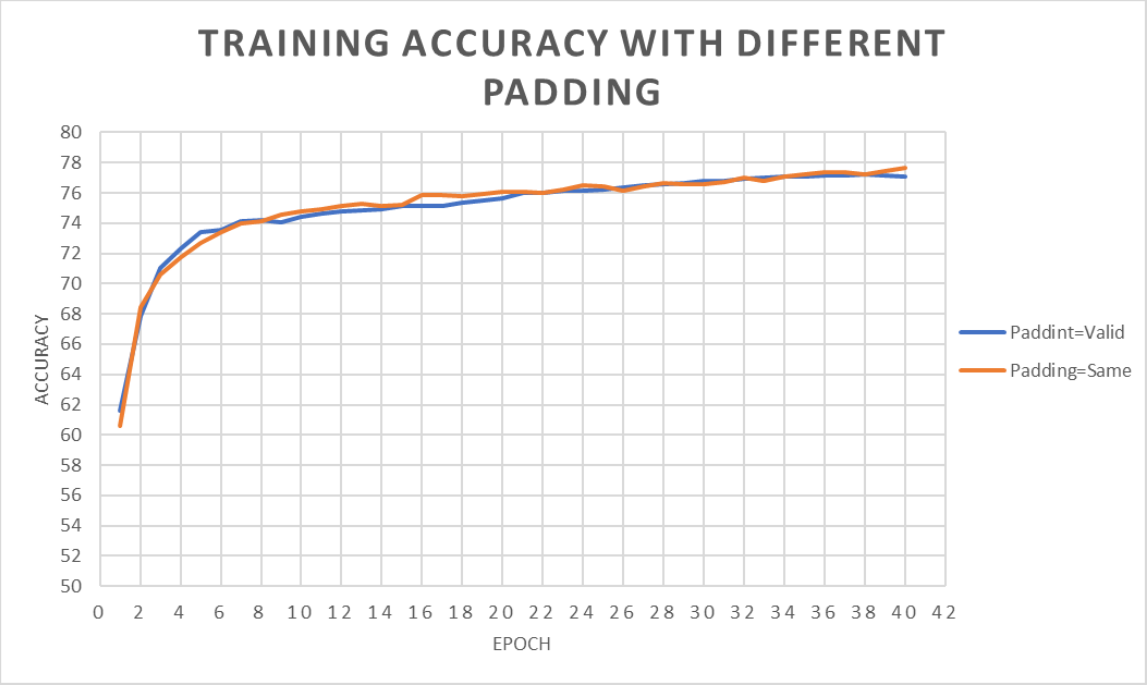
Το γέμισμα όπως έχει αναλυθεί στο υποκεφάλαιο 2.2.8.2 χρησιμοποιείται στην αρχιτεκτονική του συνελικτικού επιπέδου. Το Padding λοιπόν είναι η υπέρ-παράμετρος που χρησιμοποιείται για να γεμίσει με μηδενικά το περίγραμμα της εισόδου. Στην διάθεση μας έχουμε 2 τιμές που πρέπει να πάρει αυτή η παράμετρος οι οποίες είναι οι εξής:

- ‘Same’. Το ‘Same’ padding (zero padding) γεμίζει στην ουσία το περίγραμμά της εικόνας εισόδου με μηδενικά για να διατηρηθούν χωρικές διαστάσεις της εικόνας αυτής. Να σημειώσουμε εδώ όμως ότι με την πρόσθεση μηδενικών στα δεδομένα εισόδου μπορεί να θεωρηθεί ως θόρυβος με αποτέλεσμα όχι και τόσο καλά αποτελέσματα.
- ‘Valid’. Το ‘Valid’ Padding (without padding) δεν προσθέτει τίποτα στην εικόνα εισόδου με αποτέλεσμα να μειώνονται οι διαστάσεις της.

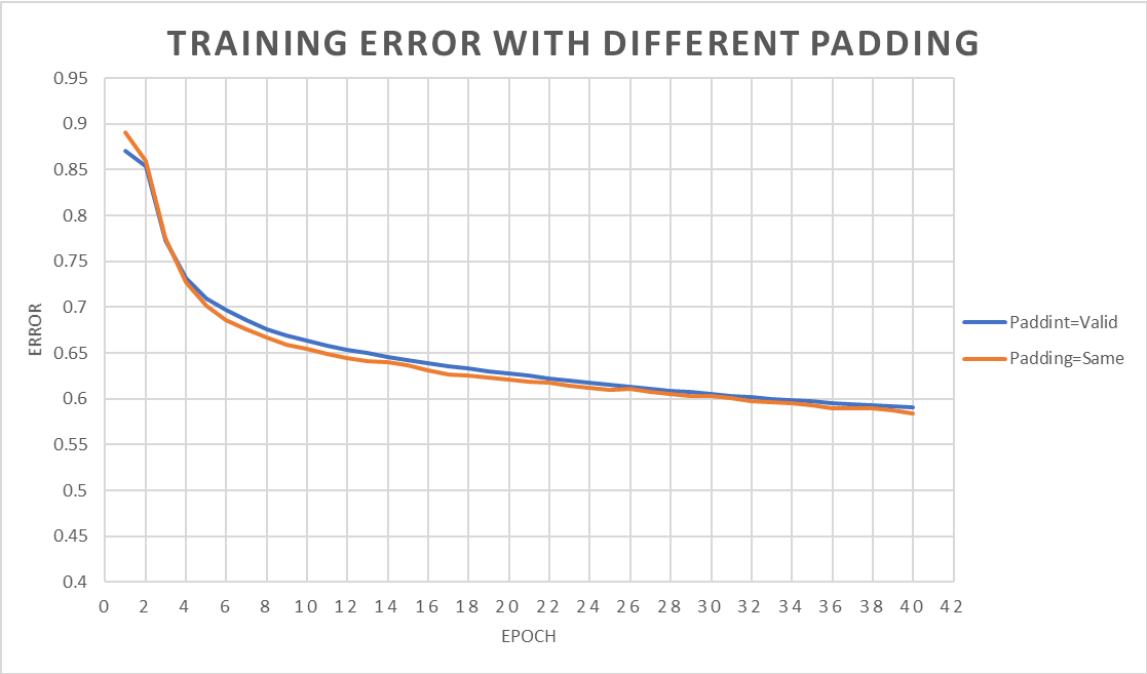
Κρατώντας λοιπόν σταθερές τις υπόλοιπες παραμέτρους δημιουργηθήκαν οι πιο κάτω γραφικές οι οποίες συγκρίνουν τα 2 δίκτυα με διαφορετική τιμή γεμίσματος.

Παράμετροί ρύθμισης	Τιμή
Εποχές	40
Μέγεθος batch δεδομένων εκπαίδευσης	7000
Μέγεθος batch δεδομένων επαλήθευσης	7289
Αριθμός batches δεδομένων εκπαίδευσης	12
Αριθμός batches δεδομένων επαλήθευσης	1
Συνάρτηση ενεργοποίησης	ReLU
Ρυθμός μάθησης	0.002
Μέγεθος φίλτρου	3
Αριθμός συνελικτικών επιπέδων	10
Γέμισμα	Valid (μεταβαλλόμενη)
Αριθμός Παράλληλων φίλτρων	32
Αριθμός βημάτων φίλτρου (stride)	1,1
Μέθοδος Ενημέρωσης του Ρυθμού Μάθησης (Updater)	Adam Optimizer
Max Pooling layers	True

Πίνακας 4.3: Οι προκαθορισμένες τιμές των παραμέτρων που έχει το υπολειπόμενο δίκτυο με την νέα τιμή του αριθμού των παράλληλων φίλτρων.

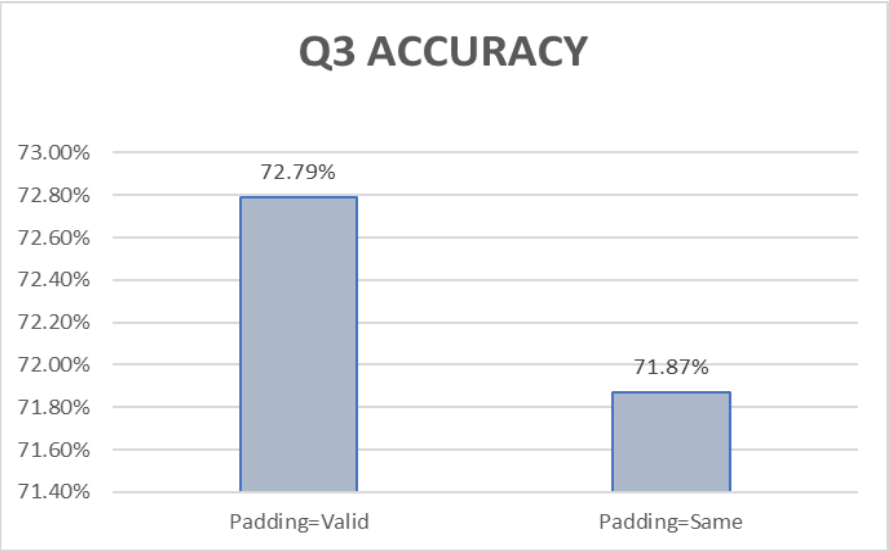


Γραφική Παράσταση 4.13: Ακρίβεια Εκπαίδευσης - Εποχή για κάθε νευρωνικό δίκτυο με τιμή γεμίσματος Valid και Same.



Γραφική Παράσταση 4.14: Σφάλμα εκπαίδευσης – εποχής για κάθε νευρωνικό δίκτυο με τιμή γεμίσματος Valid και Same.

Από τις πιο πάνω γραφικές παραστάσεις (Γραφική Παράσταση 4.13 και 4.14) συμπεραίνουμε και εδώ ότι το δίκτυο εκπαιδεύεται σωστά αφού η ακρίβεια εκπαίδευσης αυξάνεται και το σφάλμα εκπαίδευσης μειώνεται ανά εποχή. Παρατηρείται ότι τα δυο αυτά δίκτυα δεν έχουν μεγάλη διαφορά στα αποτελέσματα, συγκεκριμένα το δίκτυο με τιμή γεμίσματος Valid έχει 77.04286 τιμή ακρίβειας εκπαίδευσης και 0.59080 τιμή σφάλματος εκπαίδευσης ενώ το δίκτυο με τιμή γεμίσματος Same έχει 77.65714 τιμή ακρίβειας εκπαίδευσης και 0.5841 τιμή σφάλματος εκπαίδευσης. Όμως για να συμπεράνουμε ποιο από τα 2 δίκτυα όντως μαθαίνει και μπορεί να προβλέψει την δευτεροταγούς δομή της πρωτεΐνης καλύτερα συγκρίναμε τα ποσοστά επιτυχίας κατά την διαδικασία επαλήθευσης. Σύμφωνα με την πιο κάτω γραφική παράσταση (Γραφική Παράσταση 4.15) παρατηρείται ότι το δίκτυο με τιμή γεμίσματος Valid είχε το υψηλότερο ποσοστό ακρίβειας Q3. Η διαφορά ανάμεσα στα ποσοστά ακρίβειας ήταν αρκετά μικρή αλλά η τελική τιμή που λήφθηκε υπόψη για την τιμή του γεμίσματος ήταν το ‘Valid’.



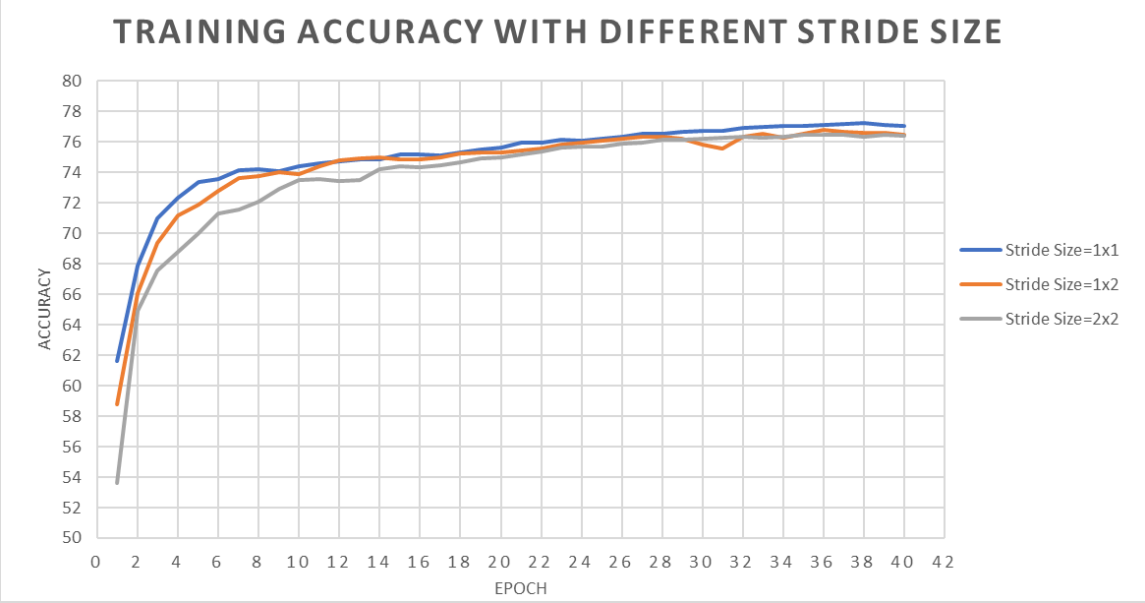
Γραφική Παράσταση 4.15: Ποσοστά επιτυχίας Q3 για κάθε νευρωνικό δίκτυο με τιμή γεμίσματος Valid και Same.

4.3.5 Χρήση Διαφορετικού Stride

Όπως έχουμε εξηγήσει σε προηγούμενα κεφάλαια, το βήμα (stride) είναι κατά πόσο θα μετακινηθεί το φίλτρο στην εικόνα εισόδου για να εκτελέσει την πράξη συνέλιξης με την αντίστοιχη περιοχή που ‘σκαρώνει’. Αναλόγως με το πόσο μικρή θα είναι η τιμή της παραμέτρου αυτής τόσο πυκνή θα είναι η σάρωση του φίλτρου στην εισόδου. Δηλαδή αν η τιμή είναι ίση με 3 τότε το φίλτρο θα κινείται ανά 3 εικονοστοιχεία στον οριζόντιο και κάθετο άξονα. Ενώ για τιμή ίση με 5 το φίλτρο θα κινείται αν 5 εικονοστοιχεία. Στην περίπτωση μικρής τιμής της παραμέτρους να τονίσουμε ότι δεν θα παραλείψει πολλές τιμές της εισόδου ενώ στην περίπτωση μεγάλης τιμής της παραμέτρους αυτής θα προσπεράσει αρκετά εικονοστοιχεία με αποτέλεσμα να μην λάβει υπόψη αρκετές περιοχές της εικόνας εισόδου που αυτό θα οδηγήσει το δίκτυο σε πιο χαμηλά αποτελέσματα. Κρατώντας λοιπόν σταθερό τον κατάλληλο αριθμό της κάθε παραμέτρου που έχει εξεταστεί και τις προκαθορισμένες τιμές για τις υπόλοιπες παραμέτρους, αλλάξαμε την τιμή της τιμής του βήματος σε πολλές διαφορετικές τιμές για να αποφασίσουμε ποια από αυτές έχει το μεγαλύτερο ποσοστό επιτυχίας Q3.

Παράμετροί ρύθμισης	Τιμή
Εποχές	40
Μέγεθος batch δεδομένων εκπαίδευσης	7000
Μέγεθος batch δεδομένων επαλήθευσης	7289
Αριθμός batches δεδομένων εκπαίδευσης	12
Αριθμός batches δεδομένων επαλήθευσης	1
Συνάρτηση ενεργοποίησης	ReLU
Ρυθμός μάθησης	0.002
Μέγεθος φίλτρου	3
Αριθμός συνελικτικών επιπέδων	10
Γέμισμα	Valid
Αριθμός Παράλληλων φίλτρων	32
Αριθμός βημάτων φίλτρου (stride)	1,1 (μεταβαλλόμενη)
Μέθοδος Ενημέρωσης του Ρυθμού Μάθησης (Updater)	Adam Optimizer
Max Pooling layers	True

Πίνακας 4.4: Οι προκαθορισμένες τιμές των παραμέτρων που έχει το υπολειπόμενο δίκτυο με την νέα τιμή του γεμίσματος.

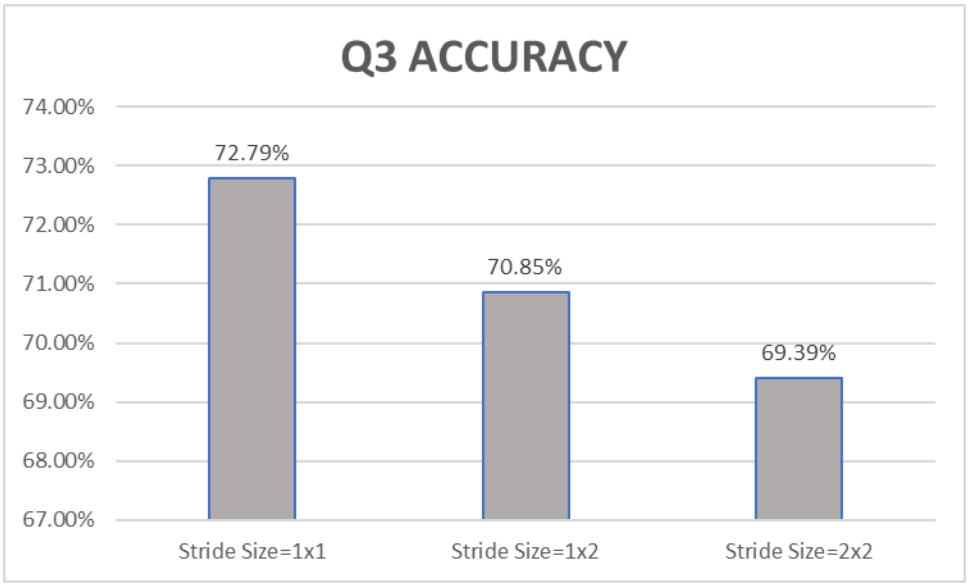


Γραφική Παράσταση 4.16: Ακρίβεια Εκπαίδευσης - Εποχή για κάθε νευρωνικό δίκτυο με διαφορετική τιμή βήματος.

Γραφική Παράσταση 4.17: Σφάλμα εκπαίδευσης – εποχής για κάθε νευρωνικό δίκτυο με διαφορετική τιμή βήματος.

Από τις πιο πάνω γραφικές παραστάσεις (Γραφική Παράσταση 4.16 και 4.17) συμπεραίνουμε ότι το δίκτυο εκπαιδεύεται σωστά αφού η ακρίβεια αυξάνεται και το σφάλμα μειώνεται ανά εποχή. Παρατηρείται ότι το δίκτυο με τιμή βήματος ίση με 1 έχει την υψηλότερη ακρίβεια και το χαμηλότερο σφάλμα σε σύγκριση με τα υπόλοιπα δίκτυα, πράγμα λογικό αφού παίρνει και την περισσότερη πληροφορία από τα δεδομένα εισόδου κα την επεξεργάζεται. Με άλλα λόγια το δίκτυο αυτό δεν παραλείπει κανένα στοιχείο εισόδου με αποτέλεσμα να εξάγει και περισσότερα διαφορετικά χαρακτηριστικά με τα οποία το δίκτυο θα μάθει και θα προβλέψει καλύτερα τις κατηγορίες του κάθε αμινοξύ.

Όπως και στην πιο κάτω γραφική παράσταση φαίνεται ότι το πιο ψηλό ποσοστό ακρίβειας Q3 είναι του δικτύου με βήμα ίσο με 1x1 ενώ το δίκτυο με βήμα ίσο 2x2 έχει το χαμηλότερο ποσοστό Q3 πράγμα αναμενόμενο. Επομένως λάβαμε υπόψη ότι το μέγεθος του βήματος θα είναι ίσο με 1, αφού αυτό δίνει και το πιο ψηλό ποσοστό επιτυχίας Q3, για να προχωρήσουμε έπειτα στις αλλαγές των υπόλοιπων παραμέτρων που έχει το δίκτυο.



Γραφική Παράσταση 4.18: Ποσοστά επιτυχίας Q3 για κάθε νευρωνικό δίκτυο με διαφορετική τιμή βήματος.

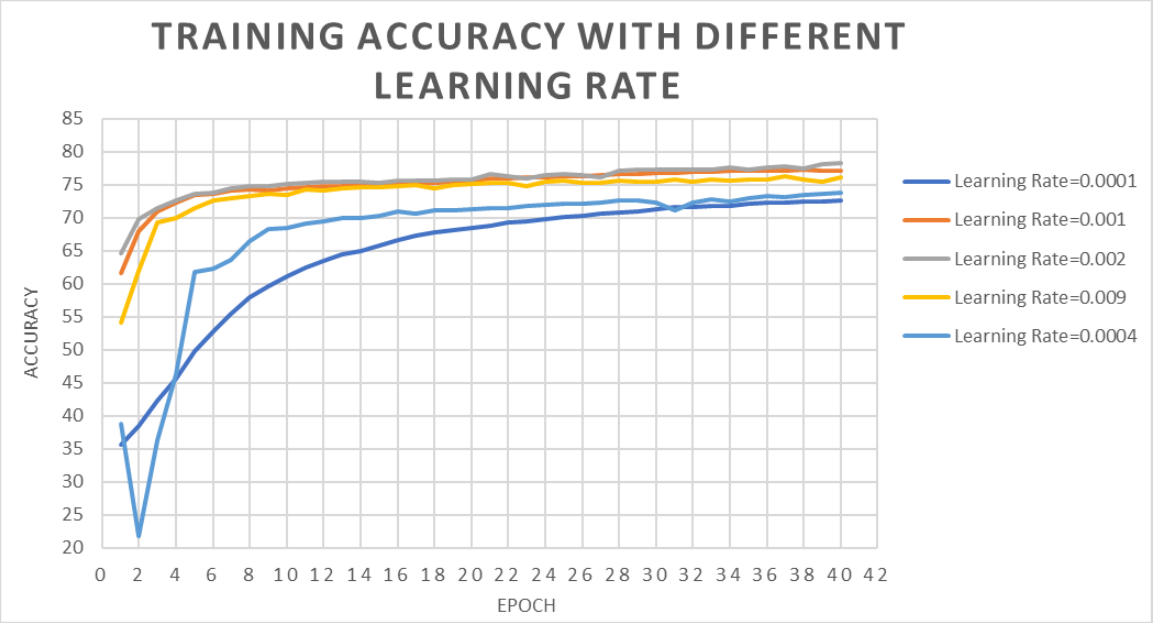
4.3.6 Χρήση Διαφορετικού ρυθμού εκμάθησης

Η παράμετρος αυτή έχει αναφερθεί στο υποκεφάλαιο 2.2.7.1 με το σύμβολο n. Ο ρυθμός εκμάθησης n καθορίζει το πόσο γρήγορα συγκλίνει η μάθηση. Μεγάλος ρυθμός μάθησης μπορεί να οδηγήσει σε γρηγορότερη σύγκλιση και σε ταλάντωση γύρω από τις βέλτιστες τιμών βαρών. Μικρός ρυθμός μάθησης έχει ως αποτέλεσμα πιο αργή σύγκλιση, ενώ μπορεί να οδηγήσει σε παγίδευση σε τοπικά ακρότατα. Γενικά μια μεγάλη τιμή της παραμέτρου αυτής θα βρει μια γρήγορη και σωστή λύση ενώ μια μικρή τιμή του ρυθμού εκμάθησης θα δώσει όχι και τόσο καλά αποτελέσματα με μεγαλύτερο χρόνο εκπαίδευσης. Η βιβλιοθήκη που έχουμε στην διάθεση μας είχε ως προκαθορισμένη μεταβλητή για την διαδικασία εκπαίδευσης τον αλγόριθμο Adam με ρυθμό εκμάθησης ίσο με 0.001. Ο Adam είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται για να ενημερώσει τα βάρη του δικτύου με βάση τα δεδομένα εκπαίδευσης. Χρησιμοποιήσαμε και εμείς αυτό τον αλγόριθμο επειδή είναι ένας από τους πιο δημοφιλείς και υπολογιστικά αποδοτικούς αλγορίθμους ενημέρωσης ρυθμού εκμάθησης, απαιτεί λίγο χώρο στη μνήμη και στο ότι λειτουργεί καλά με μεγάλα σύνολα δεδομένων και πολλές παραμέτρους. Άρα θεωρήσαμε ότι είναι ο ιδανικός αλγόριθμος ενημέρωσης ρυθμού εκμάθησης και το μόνο που αλλάζαμε σε αυτά τα πειράματα ήταν η τιμή του ρυθμού εκμάθησης. Εισάγαμε αρχικά μικρές τιμές για να

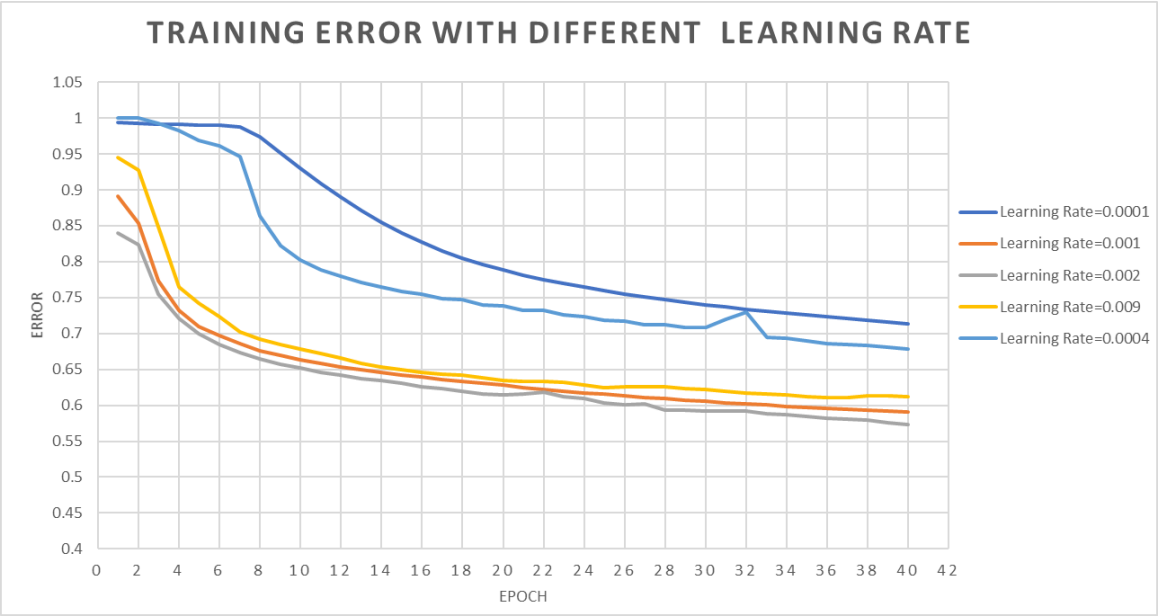
δούμε ότι όντος χρειάζεται περισσότερο χρόνο το δίκτυο να εκπαιδευτεί και επίσης αν εξάγει καλά αποτελέσματα και έπειτα προχωρήσαμε σε λίγο πιο μεγάλες τιμές.

Παράμετροί ρύθμισης	Τιμή
Εποχές	40
Μέγεθος batch δεδομένων εκπαίδευσης	7000
Μέγεθος batch δεδομένων επαλήθευσης	7289
Αριθμός batches δεδομένων εκπαίδευσης	12
Αριθμός batches δεδομένων επαλήθευσης	1
Συνάρτηση ενεργοποίησης	ReLU
Ρυθμός μάθησης	0.002 (μεταβαλλόμενη)
Μέγεθος φίλτρου	3
Αριθμός συνελικτικών επιπέδων	10
Γέμισμα	Valid
Αριθμός Παράλληλων φίλτρων	32
Αριθμός βημάτων φίλτρου (stride)	1,1
Μέθοδος Ενημέρωσης του Ρυθμού Μάθησης (Updater)	Adam Optimizer
Max Pooling layers	True

Πίνακας 4.5: Οι προκαθορισμένες τιμές των παραμέτρων που έχει το υπολειπόμενο δίκτυο με την νέα τιμή του γεμίσματος.



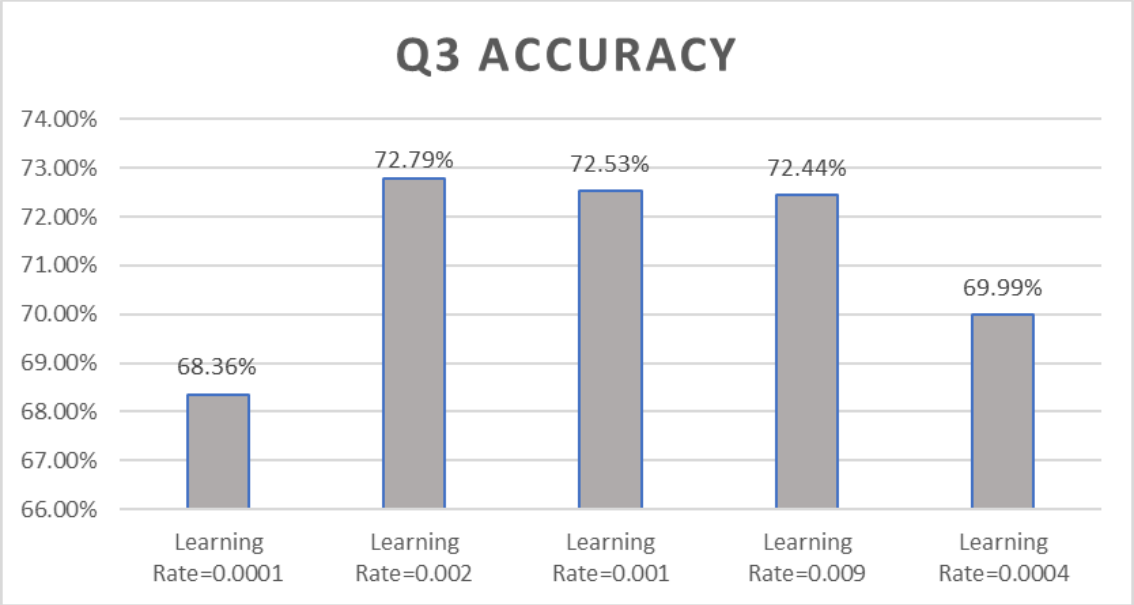
Γραφική Παράσταση 4.19: Ακρίβεια Εκπαίδευσης - Εποχή για κάθε νευρωνικό δίκτυο με διαφορετική τιμή ρυθμού εκμάθησης.



Γραφική Παράσταση 4.20: Σφάλμα εκπαίδευσης – εποχής για κάθε νευρωνικό δίκτυο με διαφορετική τιμή ρυθμού εκμάθησης.

Από τις πιο πάνω γραφικές παραστάσεις (Γραφική Παράσταση 4.19 και 4.20) συμπεραίνουμε ότι το δίκτυο εκπαιδεύεται σωστά αφού η ακρίβεια αυξάνεται και το σφάλμα μειώνεται ανά εποχή. Παρατηρείται ότι το δίκτυο με τιμή ρυθμού εκμάθησης ίση με 0.002 έχει την υψηλότερη ακρίβεια και το χαμηλότερο σφάλμα σε σύγκριση με τα υπόλοιπα δίκτυα. Πράγμα αναμενόμενο αφού όπως έχει αναφερθεί πιο πάνω, αν η τιμή του ρυθμού μάθησης είναι μεγάλη θα παραχθεί καλύτερο αποτέλεσμα σε σύγκριση με μια πιο μικρή τιμή του ρυθμού μάθησης. Αυτό φαίνεται στην Γραφική Παράσταση 4.19 όπου η ακρίβεια εκπαίδευσης στο δίκτυο με ρυθμό μάθησης ίσο με 0.002 είναι μεγαλύτερη από την ακρίβεια εκπαίδευσης στο δίκτυο με ρυθμό μάθησης ίσο με 0.0001.

Όπως και στην πιο κάτω γραφική παράσταση φαίνεται ότι το πιο ψηλό ποσοστό ακρίβειας Q3 είναι του δικτύου με ρυθμό εκμάθησης ίσο με 0.002 ενώ το δίκτυο με βήμα ίσο 0.0001 έχει το χαμηλότερο ποσοστό Q3, πράγμα αναμενόμενο. Επομένως λάβαμε υπόψη ότι η τιμή του ρυθμού εκμάθησης θα είναι ίσο με 0.002, αφού αυτό δίνει και το πιο ψηλό ποσοστό επιτυχίας Q3.



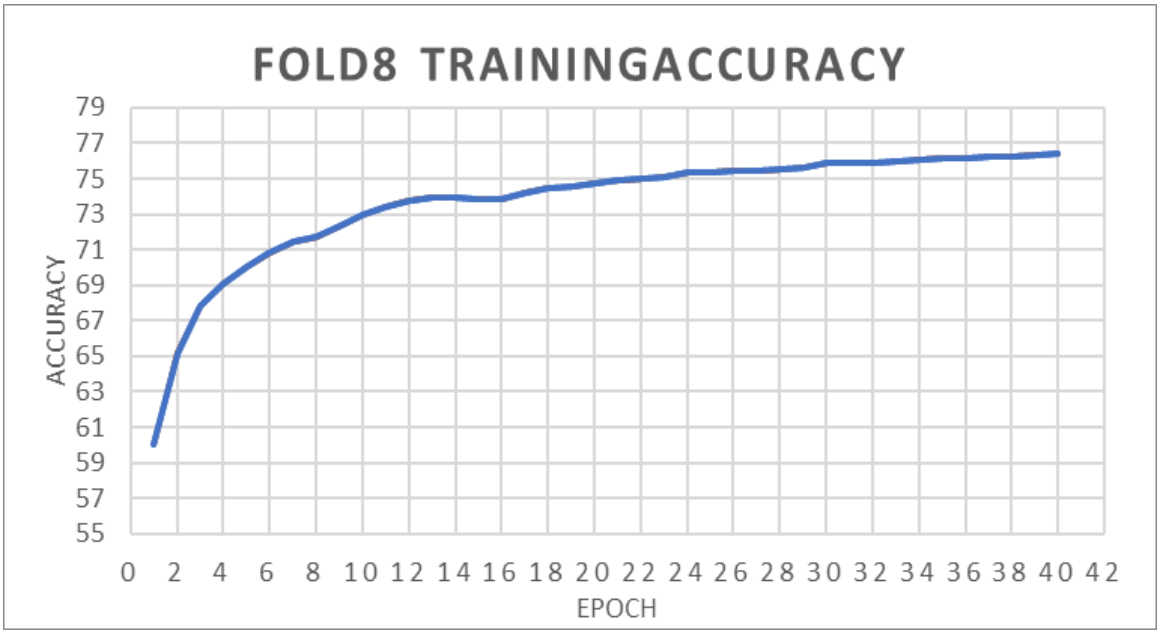
Γραφική Παράσταση 4.21: Ποσοστά επιτυχίας Q3 για κάθε νευρωνικό δίκτυο με διαφορετική τιμή ρυθμού εκμάθησης.

4.3.7 Χρήση αρχείου δεδομένων ονομασίας 'fold8'

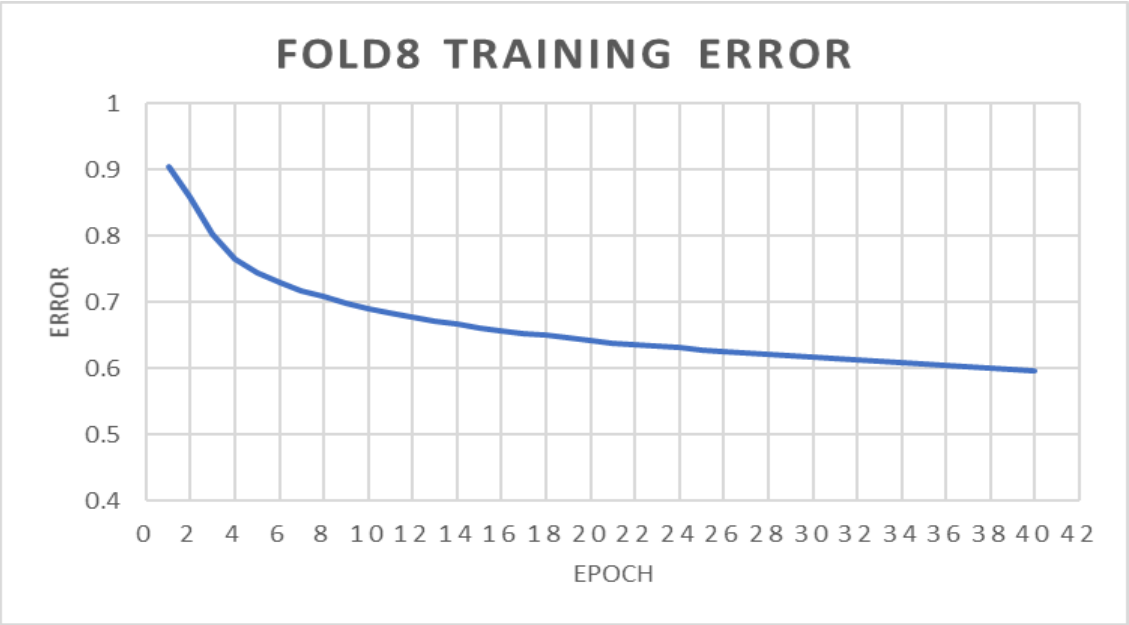
Σύμφωνα με τα πειράματα που έχουν γινεί με διαφορετικές τιμές των διάφορων παραμέτρων που επιλεχθήκαν για να βελτιστοποιηθούν έχουν γινεί σε ένα συγκεκριμένο αρχείο, το αρχείο με ονομασία fold1. Όμως παρατηρήθηκε ότι το ποσοστό επιτυχίας Q3, παρόλο που υπήρξαν αρκετές αλλαγές στις τιμές των υπερπαραμέτρων, παραμένει στην τιμή 72.79% από τα αρχικά πειράματα που υπήρξαν. Με βάση το γεγονός αυτό προσπαθήσαμε να τρέξουμε το δίκτυο με διαφορετικό αρχείο δεδομένων εισόδου, συγκεκριμένα το αρχείο με ονομασία fold8 για να δούμε αν υπάρχει αλλαγή στο ποσοστό επιτυχίας Q3 και συγκεκριμένα για τυχόν πιο ψηλό αποτέλεσμα. Επιλέχθηκε το συγκεκριμένο αρχείο βάση των ποσοστών που παραχθήκαν από την έρευνα του Διονυσίου (2018), όπου στο fold8 παράχθηκε και το ψηλότερο ποσοστό επιτυχίας Q3 λαμβάνοντας υπόψη ότι δεν χρησιμοποιείται οποιοδήποτε φίλτρο και ensembles. Επομένως εκτελέστηκε το πείραμα και συγκεκριμένα το δίκτυο παίρνει νέα δεδομένα εισόδου και εξόδου και λαμβάνοντας υπόψη τις τιμές των παραμέτρων που βελτιστοποιηθήκαν που φαίνονται και στο πιο κάτω πίνακα (Πίνακας 4.6) εξαχθήκαν οι πιο κάτω γραφικές παραστάσεις.

Παράμετροί ρύθμισης	Τιμή
Εποχές	40
Μέγεθος batch δεδομένων εκπαίδευσης	7000
Μέγεθος batch δεδομένων επαλήθευσης	7289
Αριθμός batches δεδομένων εκπαίδευσης	12
Αριθμός batches δεδομένων επαλήθευσης	1
Συνάρτηση ενεργοποίησης	ReLU
Ρυθμός μάθησης	0.002
Μέγεθος φίλτρου	3
Αριθμός συνελικτικών επιπέδων	10
Γέμισμα	Valid
Αριθμός Παράλληλων φίλτρων	32
Αριθμός βημάτων φίλτρου (stride)	1,1
Μέθοδος Ενημέρωσης του Ρυθμού Μάθησης (Updater)	Adam Optimizer
Max Pooling layers	True

Πίνακας 4.6: Οι προκαθορισμένες τιμές των παραμέτρων που έχει το υπολειπόμενο δίκτυο.

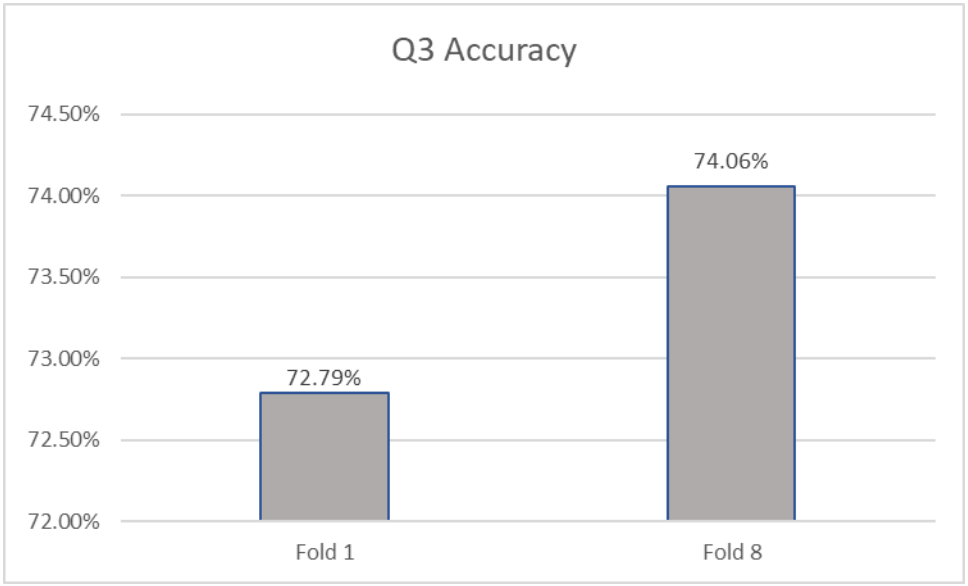


Γραφική Παράσταση 4.22: Ακρίβεια Εκπαίδευσης - Εποχή για το δίκτυο που χρησιμοποίησε τα δεδομένα του αρχείου fold 8.



Γραφική Παράσταση 4.23: Σφάλμα εκπαίδευσης – εποχής για το δίκτυο που χρησιμοποίησε τα δεδομένα του αρχείου fold 8.

Από την εκπαίδευση του συγκεκριμένου αρχείου παράχθηκε 76.41% ποσοστό ακρίβειας και 0.5962 σφάλμα εκπαίδευσης. Όμως για να συμπεράνουμε ότι όντως το δίκτυο μαθαίνει και μπορεί να προβλέψει την δευτεροταγή δομή της πρωτεΐνης έπρεπε να εξαχθούν και τα ποσοστά επιτυχίας κατά την διαδικασία επαλήθευσης.



Γραφική Παράσταση 4.25: Ποσοστά επιτυχίας Q3 για το δίκτυο που χρησιμοποίησε τα δεδομένα του αρχείου fold 8 και fold 1.

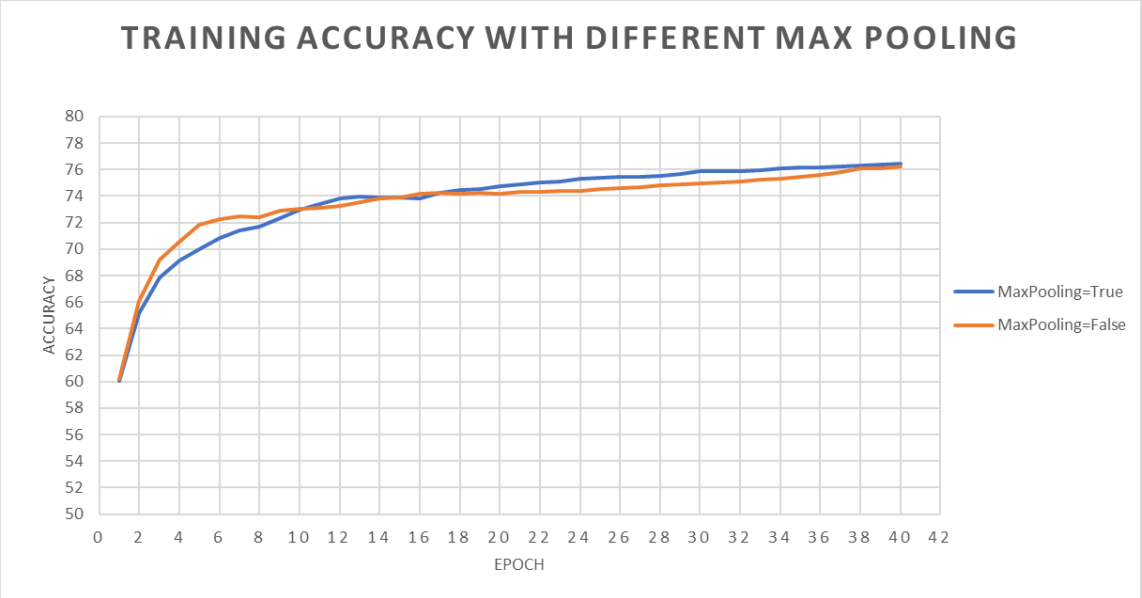
Σύμφωνα με την πιο πάνω γραφική παράσταση φαίνεται ότι όντως το δίκτυο έχει εκπαιδευτεί και μπορεί να προβλέψει την δευτεροταγή δομή της πρωτεΐνης βάσει αγνώστων δεδομένων και μάλιστα παράχθηκε και πιο ψηλό ποσοστό σε σύγκριση με την εκτέλεση του δικτύου που χρησιμοποίησε το αρχείο δεδομένων με ονομασία fold1. Το ποσοστό ακρίβειας Q3 που πέτυχε το δίκτυο αυτό είναι 74.06% - ποσοστό σχετικά πιο ψηλό από το ποσοστό που πέτυχε το δίκτυο με την χρήση του fold1. Συνεπώς η χρήση του κάθε αρχείο δεδομένων, που έχουμε στην διάθεση μας, από το υπολειπόμενο νευρωνικό δίκτυο παράγει ένα διαφορετικό ποσοστό ακρίβειας Q3 το οποίο πρέπει να ληφθεί υπόψη για την διαδικασία του 10-fold cross-validation.

4.3.8 Χρήση max pooling layers σε συνδυασμό το αρχείο δεδομένων ονομασίας 'fold8'

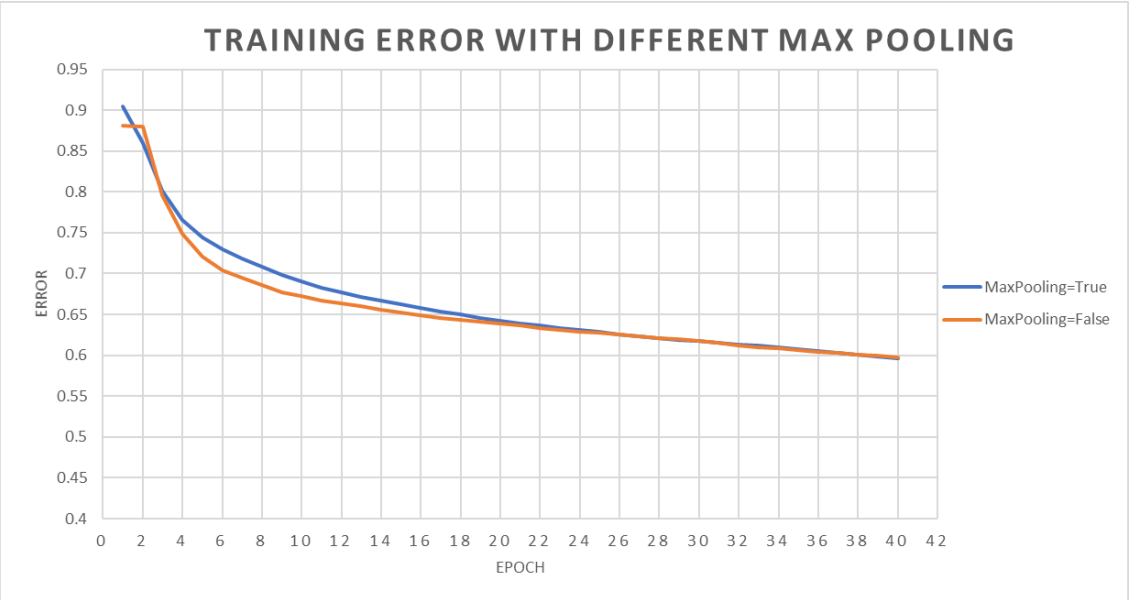
Όπως έχει αναφερθεί στο υποκεφάλαιο 2.2.8.2 ιστορικά ήταν κοινώς αποδεκτό να χρησιμοποιούνται τα επίπεδα υποδειγματοληψίας, ή αλλίως τα max pooling layers, μεταξύ διαδοχικών συνελικτικών επιπέδων τα οποία οδηγούν στην αφαίρεση του 75% του συνολικού διανυσματικού χώρου των χαρτών των χαρακτηριστικών των δειγμάτων που εμφανίζονται στην είσοδο. Αυτή η επιθετική μείωση είναι χρήσιμη επειδή μειώνει την υπέρεκπαίδευση στην ταξινόμηση μικρών σετ δεδομένων με έντονες διαφορές μεταξύ των κατηγοριών τους. Σε προβλήματα όμως, όπου οι κατηγορίες είναι αρκετά περισσότερες, η χρήση των επιπέδων χωρικής υπό-δειγματοληψίας μπορεί να καταστήσει αδύνατη την διακριτοποίηση των κατηγοριών στα τελευταία επίπεδα και έτσι συστήνεται να γινεί χρήση μεγαλύτερων μεγεθών stride για να μειωθούν οι διαστάσεις των χαρτών χαρακτηριστικών. Για να επιβεβαιώσουμε αυτό τον ισχυρισμό πραγματοποιήθηκαν κάποια πειράματα με, και χωρίς την χρήση pooling επιπέδων, λαμβάνοντας υπόψη τις τιμές των παραμέτρων που βελτιστοποιηθήκαν που φαίνονται και στο πιο κάτω πίνακα (Πίνακας 4.7) και την χρήση του αρχείου δεδομένων fold 8, έτσι ώστε να παραχθούν τα κατάλληλα αποτελέσματα και βάσει αυτών να αποφασίσουμε αν θα πρέπει να χρησιμοποιηθεί ή όχι pooling επίπεδα στο υπολειπόμενο νευρωνικό δίκτυο.

Παράμετροί ρύθμισης	Τιμή
Εποχές	40
Μέγεθος batch δεδομένων εκπαίδευσης	7000
Μέγεθος batch δεδομένων επαλήθευσης	7289
Αριθμός batches δεδομένων εκπαίδευσης	12
Αριθμός batches δεδομένων επαλήθευσης	1
Συνάρτηση ενεργοποίησης	ReLU
Ρυθμός μάθησης	0.002
Μέγεθος φίλτρου	3
Αριθμός συνελικτικών επιπέδων	10
Γέμισμα	Valid
Αριθμός Παράλληλων φίλτρων	32
Αριθμός βημάτων φίλτρου (stride)	1,1
Μέθοδος Ενημέρωσης του Ρυθμού Μάθησης (Updater)	Adam Optimizer
Max Pooling layers	True (μεταβαλλόμενη)

Πίνακας 4.7: Οι προκαθορισμένες τιμές των παραμέτρων που έχει το υπολειπόμενο δίκτυο μέχρις στιγμής.

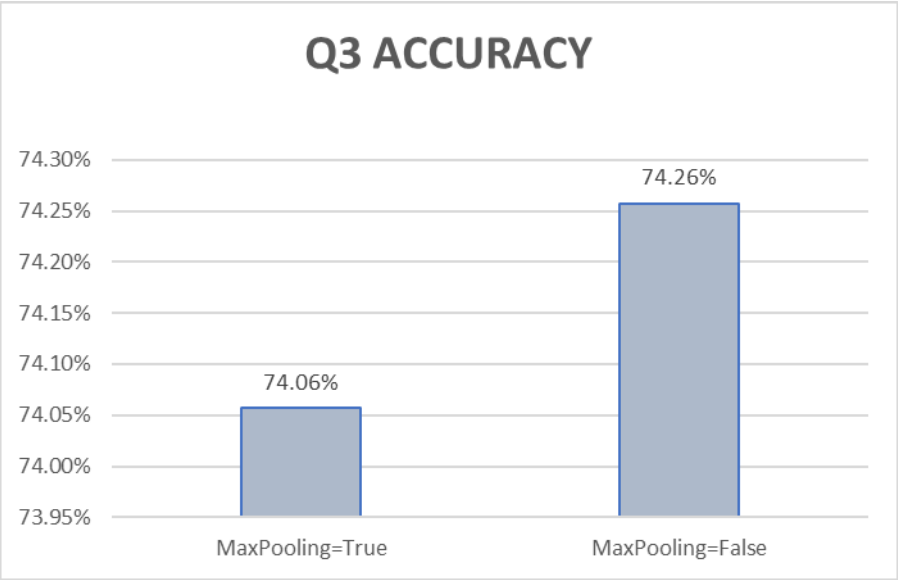


Γραφική Παράσταση 4.25: Ακρίβεια Εκπαίδευσης - Εποχή για κάθε νευρωνικό δίκτυο με και χωρίς max pooling layers.



Γραφική Παράσταση 4.26: Σφάλμα εκπαίδευσης – εποχής για κάθε νευρωνικό δίκτυο με και χωρίς max pooling layers.

Από τις πιο πάνω γραφικές παραστάσεις (Γραφική Παράσταση 4.25 και 4.26) συμπεραίνουμε και εδώ ότι το δίκτυο εκπαιδεύεται σωστά αφού η ακρίβεια εκπαίδευσης αυξάνεται και το σφάλμα εκπαίδευσης μειώνεται ανά εποχή. Παρατηρείται ότι τα δυο αυτά δίκτυα δεν έχουν μεγάλη διαφορά στα αποτελέσματα, συγκεκριμένα το δίκτυο με χρήση max pooling επιπέδων έχει 76.41428 τιμή ακρίβειας εκπαίδευσης και 0.596273 τιμή σφάλματος εκπαίδευσης ενώ το δίκτυο χωρίς την χρήση max pooling επιπέδων έχει 76.200005 τιμή ακρίβειας εκπαίδευσης και 0.59717 τιμή σφάλματος εκπαίδευσης. Όμως για να συμπεράνουμε ποιό από τα 2 δίκτυα όντως μαθαίνει και μπορεί να προβλέψει την δευτεροταγή δομή της πρωτεΐνης καλύτερα συγκρίναμε τα ποσοστά επιτυχίας κατά την διαδικασία επαλήθευσης. Σύμφωνα με την πιο κάτω γραφική παράσταση (Γραφική Παράσταση 4.27) παρατηρείται ότι το δίκτυο χωρίς την χρήση max pooling επιπέδων είχε το υψηλότερο ποσοστό ακρίβειας Q3 Συγκεκριμένα το δίκτυο με χρήση max pooling επιπέδων πέτυχε ποσοστό ακρίβειας Q3 74.05673% ενώ το δίκτυο χωρίς την χρήση max pooling επιπέδων πέτυχε ποσοστό ακρίβειας Q3 74.25743% χρησιμοποιώντας το αρχείο δεδομένων fold 8. Η διαφορά ανάμεσα στα ποσοστά ακρίβειας ήταν αρκετά μικρή αλλά η τελική τιμή που λήφθηκε υπόψη για την τιμή του max pooling ήταν το ‘False’, δηλαδή να μην χρησιμοποιηθούν τα pooling επίπεδα στο υπολειπόμενο νευρωνικό δίκτυο.



Γραφική Παράσταση 4.27: Ποσοστά επιτυχίας Q3 για κάθε νευρωνικό δίκτυο με και χωρίς max pooling layers.

Κεφάλαιο 5

Συμπεράσματα και Μελλοντική Εργασία

5.1 Συμπεράσματα 116
5.2 Μελλοντική Εργασία 118

5.1 Συμπεράσματα

Στο προηγούμενο κεφάλαιο αναλύθηκαν τα αποτελέσματα από ορισμένα πειράματα που εκτελέσαμε για τυχόν βελτιστοποιήσεις στις παραμέτρους του δικτύου. Τροποποιώντας τις υπερπαραμέτρους πετύχαμε ποσοστό επιτυχίας Q3 72.79% για το fold 1 και 74.06% για το fold 8. Το fold 1 και το fold 8 όπως έχει προαναφερθεί είναι υποδείγματα (κομμάτια) της επεξεργαζόμενης βάσης CB513. Αφού μετά την εισαγωγή του καινούργιου αρχείου δεδομένων στο δίκτυο (fold 8) παρατηρήσαμε πιο ψηλό ποσοστό ακρίβειας και επομένως τρέξαμε ακόμη ένα πείραμα αλλάζοντας την τιμή της παραμέτρου που αντιπροσωπεύει τα επίπεδα υποδειγματολειψίας. Από τους πίνακες στο προηγούμενο κεφάλαιο παρατηρείται ότι η τιμή αυτής της παραμέτρου είναι ίση με True, που αυτό υποδηλώνει ότι γίνεται χρήση των επιπέδων υποδειγματολειψίας. Επομένως αλλάξαμε αυτή την τιμή σε False, δεδομένου ότι γίνεται και χρήση του αρχείου fold 8, πετυχαίνοντας ένα πιο ψηλό ποσοστό ακρίβειας Q3 και συγκεκριμένα 74.26%. Το ποσοστό αυτό είναι σχετικά ικανοποιητικό αν λάβουμε υπόψη την πολυπλοκότητα που έχει το πρόβλημα της πρόβλεψης της δευτεροταγούς δομής της πρωτεΐνης. Όπως έχει προαναφερθεί η χρήση των επιπέδων υποδειγματολειψίας μειώνει τις διαστάσεις των χαρτών χαρακτηριστικών που παράγονται από τα συνελικτικά επίπεδα. Αυτό βοηθά αρκετά σε προβλήματα επεξεργασίας εικόνας. Όμως στην συγκεκριμένη έρευνα, η εικόνα που παίρνει σαν είσοδο το υπολειπόμενο δίκτυο είναι η αναπαράσταση που δημιούργησε ο Διονυσίου (2018) όπως αναλύθηκε προηγουμένως, που στην ουσία είναι η ευθυγράμμιση των αμινοξέων. Επομένως η εισαγωγή pooling επιπέδων θα προκαλέσει αρκετές αλλαγές στην σειρά των αμινοξέων με αποτέλεσμα να διαλυθεί αυτή η αναπαράσταση και να υπάρξουν καταστροφικές προβλέψεις. Συνεπώς έτσι εξηγείται και η διαφορά που υπήρξε αλλάζοντας την τιμή των επιπέδων υποδειγματολειψίας από True σε False. Οι τιμές που οριστήκαν στις παραμέτρους του δικτύου οι οποίες οδήγησαν στο υψηλότερο ποσοστό επιτυχίας Q3 χρησιμοποιώντας το αρχείο δεδομένων με ονομασία fold 8 φαίνονται στον πίνακα πιο κάτω (Πίνακας 5.1).

Παράμετροί ρύθμισης	Τιμή
Εποχές	40
Μέγεθος batch δεδομένων εκπαίδευσης	7000

Μέγεθος batch δεδομένων επαλήθευσης	7289
Αριθμός batches δεδομένων εκπαίδευσης	12
Αριθμός batches δεδομένων επαλήθευσης	1
Συνάρτηση ενεργοποίησης	ReLU
Ρυθμός μάθησης	0.002
Μέγεθος φίλτρου	3
Αριθμός συνελικτικών επιπέδων	10
Γέμισμα	Valid
Αριθμός Παράλληλων φίλτρων	32
Αριθμός βημάτων φίλτρου (stride)	1,1
Μέθοδος Ενημέρωσης του Ρυθμού Μάθησης (Updater)	Adam Optimizer
Max Pooling layers	False

Πίνακας 5.1: Οι τελικές τιμές των παραμέτρων που έχει το υπολειπόμενο δίκτυο οι οποίες δίνουν το πιο ψηλό ποσοστό επιτυχίας Q3.

Η έρευνα αυτή είχε ως κύριο στόχο την επίλυση της πρόβλεψης της δευτεροταγούς δομής των πρωτεϊνών με την χρήση των βαθιών υπολειπόμενων νευρωνικών δικτύων. Με βάση τα χαρακτηριστικά αυτών των βαθιών νευρωνικών δικτύων που έχουν αναλυθεί στα προηγούμενα κεφάλαια μπορούμε να πούμε ότι έχουν κατηγοριοποιήσει σε γενικές γραμμές αρκετά καλά τα αμινοξέα της κάθε πρωτεΐνης εξάγοντας και σχετικά ψηλά ποσοστά επιτυχίας. Σημαντικό να τονιστεί ότι για να προκύψουν αυτά τα αποτελέσματα απαραίτητη ήταν και η χρήση των MSA αρχείων, αφού βάση αυτών έχει βελτιωθεί η αναπαράσταση των δεδομένων εισόδου σε τέτοιου είδος δίκτυα, συνελικτικά δίκτυα, με αποτέλεσμα το δίκτυο να μπορεί να γενικεύσει καλύτερα.

Ένα αρκετά εντυπωσιακό αλλά και σημαντικό πλεονέκτημα που υπήρξε κατά την έρευνα αυτή ήταν στο ότι μπορέσαμε να επεκτείνουμε το βάθος του δίκτυού για το συγκεκριμένο πρόβλημα που ερευνάτε τα τελευταία 10 περίπου χρόνια. Μπορέσαμε να εκπαιδεύσουμε μέχρι και 25 συνελικτικά επίπεδα, όμως το δίκτυο που είχε τα πιο ψηλά ποσοστά επιτυχίας κατά την επαλήθευση ήταν το δίκτυο με 10 συνελικτικά επίπεδα, που αναλόγως σε αυτό το δίκτυο υπάρχει αρκετό βάθος. Βάσει αυτού, συμπεραίνουμε ότι όντως η υπολειπόμενη μάθηση μπορεί να εκπαιδεύσει σε μεγάλο βάθος νευρωνικά

δίκτυα και να παράξει και βέλτιστα αποτελέσματα με αρκετά μεγάλο βάθος στο δίκτυο. Αυτό είχε μεγάλο πλεονέκτημα αφού μέχρι στιγμής τα βαθιά νευρωνικά δίκτυα που υλοποιηθήκαν για την επίλυση του προβλήματος της πρόβλεψης της δευτεροταγούς δομής της πρωτεΐνης αντιμετώπιζαν το πρόβλημα εξαφανιζόμενης κλίσης και γι' αυτό τον λόγο το μεγαλύτερο βάθος που είχαν ήταν μέχρι τα 5 κρυφά επίπεδα ούτως ώστε να παραχθεί ένα ικανοποιητικό ποσοστό επιτυχίας.

Παρόλα αυτά, σύμφωνα με την έρευνα του Διονυσίου (2018), όπου υλοποιήθηκε ένα κανονικό βαθύ νευρωνικό δίκτυο παραχθήκαν πιο ψηλά αποτελέσματα. Σύμφωνα με αυτό η ιδέα στο ότι τα υπολειπόμενα νευρωνικά δίκτυα παράγουν πιο ψηλά ποσοστά επιτυχίας από τα παραδοσιακά βαθιά νευρωνικά δίκτυα δεν ισχύει. Συγκεκριμένα το δίκτυο που ανάλυσε ο Διονυσίου (2018) χωρίς την χρήση κάποιου φίλτρου ή ensembles και δεδομένου ότι έχει χρησιμοποιηθεί η ίδια βάση δεδομένων (CB513), πιο συγκεκριμένα στο αρχείο με ονομασία fold 1, έχει πετύχει ποσοστό μέχρι και 75.16% και στο αρχείο με ονομασία fold 8, έχει πετύχει ποσοστό μέχρι και 76.35%. Ωστόσο τα υπολειπόμενα δίκτυα πέτυχαν ποσοστό ακρίβειας 72.79% για το αρχείο με ονομασία fold1 και 74.26% για το αρχείο με ονομασία fold 8 (χωρίς max pooling). Συνεπώς τα ποσοστά ακρίβειας Q3 είναι πιο μεγάλα από τα ποσοστά που πετύχαν τα υπολειπόμενα δίκτυα, μετά από αρκετή βελτιστοποίηση των παραμέτρων του δικτύου, πράγμα που υποδεικνύει ότι όντως ο ισχυρισμός των He et al. (2016a) στο ότι τα ResNets παράγουν υψηλότερα ποσοστά επιτυχίας ίσως δεν ισχύει για το συγκεκριμένα πρόβλημα.

5.2 Μελλοντική Εργασία

Λόγω του ότι αυτά τα υπολειπόμενα νευρωνικά δίκτυα έχουν εφευρεθεί αρκετά πρόσφατα υπάρχουν περιθώρια να τα τροποποιήσουμε για να εξάγουν πιο βέλτιστα αποτελέσματα και να δούμε αν όντως ο ισχυρισμός των He et al. (2016), ότι τα υπολειπόμενα δίκτυα μπορούν να πετύχουν ψηλότερα ποσοστά από τα συνηθισμένα συνελικτικά δίκτυα, ισχύει.

Είναι σημαντικό να ληφθεί υπόψη ο χρόνος και η χωρητικότητα που χρειάστηκε για το κάθε πείραμα για να τυχόν περισσότερα πειράματα ή οποιεσδήποτε αλλαγές στο

μέλλον. Πιο συγκεκριμένα ο χρόνος που χρειάστηκε περίπου για να εκτελεστεί το κάθε πείραμα, δεδομένου ότι η παράμετρος που αντιπροσωπεύει τον αριθμό των συνελικτικών στρωμάτων που υπήρχαν στο υφιστάμενο υπολειπόμενο δίκτυο ήταν ίση με την τιμή 10, ήταν περίπου 518400 δευτερόλεπτα, δηλαδή περίπου 6 μέρες. Να τονιστεί ότι αρχικά τα πειράματα εκτελεστήκαν σε φορητό υπολογιστή με χωρητικότητα (RAM) ίση με 8 GB και υπήρξαν αρκετές αποτυχημένες εκτελέσεις λόγω του ότι το δίκτυο χρειαζόταν περισσότερη χωρητικότητα μνήμης για να εκπαιδευτεί. Επομένως αναβαθμίστηκε η χωρητικότητα σε 16GB και το δίκτυο με αριθμό στρωμάτων ίσο με 10 μπορούσε να εκπαιδευτεί στις 3 ημέρες και χρειαζόταν 13 GB μνήμη. Για τα πειράματα με αριθμό στρωμάτων μεγαλύτερο από 10 χρειάστηκαν περισσότερες μέρες και περισσότερη χωρητικότητα μνήμης στον υπολογιστή αφού προσθέταμε περισσότερη πολυπλοκότητα στο δίκτυο. Συγκεκριμένα η εκτέλεση του δικτύου με αριθμό επιπέδων ίσο με 25 χρειάστηκε 15GB μνήμη και περίπου 5 ημέρες μέχρι να ολοκληρώσει την εκπαίδευση του. Επομένως πρέπει να ληφθεί υπόψη ότι για να εκπαιδευτεί αυτό το υφιστάμενο δίκτυο πρέπει να υπάρχει διαθέσιμή ηλεκτρονική συσκευή η οποία να υποστηρίζει το λιγότερο 16GB μνήμη RAM και όσο αφορά τον χρόνο εκτέλεσης εξαρτάται πλήρως από τις παραμέτρους του υπολειπόμενου δικτύου, αλλά γενικά θα χρειαστεί περισσότερο από μια εβδομάδα αν αυξήσουμε τα συνελικτικά επίπεδα του και τα παράλληλα φίλτρα τα οποία εκτελούν την πράξη συνέλιξης μέσα στα συνελικτικά αυτά επίπεδα. Να αναφέρουμε ότι τα δεδομένα εισόδου μας τα οποία ήταν χωρισμένα σε 10 αρχεία βάση της τεχνικής επικύρωσης (10 fold-cross validation) είχαν περίπου το ίδιο μέγεθος αρχείου και επομένως χρειαστήκαν περίπου το ίδιο χρόνο εκτέλεσης και την ίδια χωρητικότητα μνήμης.

Ένα σημαντικό πρόβλημα που υπήρξε στην συγκεκριμένη διπλωματική ήταν η έλλειψη χρόνου για περισσότερα πειράματα στο υφιστάμενο νευρωνικό δίκτυο. Είναι απαραίτητη η εκτέλεση διάφορων πειραμάτων όπου βάσει αυτών βελτιστοποιούνται οι παράμετροι του δικτύου, οι οποίες παράμετροι καθορίζουν το μέγιστο της απόδοσης αυτού του νευρωνικού δικτύου. Συγκεκριμένα, θα μπορούσαμε να εκτελέσουμε πειράματα με διαφορετική τιμή της ορμής και της μεθόδου ενημέρωσης ρυθμού μάθησης. Πριν όμως εκτελέσουμε πειράματα με αυτές τις επιπρόσθετες παραμέτρους θα ήταν καλό να εκτελέσουμε περισσότερα πειράματα χρησιμοποιώντας κι άλλες διαφορετικές τιμές στην παράμετρο που αντιπροσωπεύει τον αριθμό των συνελικτικών

επιπέδων που υπάρχουν στο υπολειπόμενο δίκτυο. Η παράμετρος αυτή είναι αρκετά σημαντική λόγω του ότι καθορίζει και πόσα περισσότερα ή λιγότερα διαφορετικά χαρακτηριστικά των εικόνων εισόδου θα εντοπιστούν. Το πιο σημαντικό όμως είναι ότι χρησιμοποιείται ένα υπολειπόμενο δίκτυο, που αυτό σημαίνει ότι χρειάζεται περισσότερο βάθος για να υπάρξουν οι επιπρόσθετες συνδέσεις παραλήψεις στο δίκτυο μας. Επομένως είναι καλό να πραγματοποιηθούν περισσότερα πειράματα μεταβάλλοντας τον αριθμό των στρώσεων, και συγκεκριμένα ένα προς ένα, για τυχόν εντοπισμό δραματικής αύξησης στο ποσοστό ακρίβειας Q3. Συγκεκριμένα θα μπορούσαμε αρχικά να δώσουμε τιμές σε αυτή την παράμετρο τις οποίες δεν λάβαμε υπόψη στο πείραμα, για παράδειγμα τις τιμές 11 μέχρι 14 και 21 μέχρι 24 και έπειτα να δώσουμε πιο ψηλότερες τιμές (μεγαλύτερες του 25) αφού υπάρχει και ο ισχυρισμός ότι το υπολειπόμενο δίκτυο μπορεί να έχει ένα εξαιρετικά μεγάλο βάθος συντηρώντας την ακρίβεια πρόβλεψης του.

Έχοντας λοιπόν όσο δυνατόν περισσότερα πειράματα, τόσο βέλτιστη θα είναι η ανάθεση των παραμέτρων του δικτύου οι οποίες καθορίζουν κατά πόσο το δίκτυο προβλέπει την κατηγορία του κάθε αμινοξέος σωστά. Επίσης θα μπορούσαν να χρησιμοποιηθούν κάποιες τεχνικές ensembles και κάποιες μεθόδοι filtering με χρήση πειραμάτων, ούτως ώστε να διορθώνουν ορισμένες προβλέψεις που έχουν γινεί και συνεπώς να παραχθούν πιο ψηλά ποσοστά επιτυχίας.

Επίσης, θα μπορούσαμε να δημιουργήσουμε μια παραλλαγή συνελικτικών δικτύων συνδυάζοντας τα συνελικτικά επίπεδα των δικτύων με κάποιο άλλο είδος νευρωνικό δίκτυο επιβλεπόμενης μάθησης, το οποίο θα λαμβάνει σαν είσοδο τους γνωστούς χάρτες χαρακτηριστικών τους οποίους μπορούμε να λάβουμε από το τελευταίο επίπεδο υποδειγματολειψίας ή από το τελευταίο συνελικτικό επίπεδο δεδομένου ότι δεν υπάρχει επίπεδο υποδειγματολειψίας. Συγκεκριμένα θα μπορούσαμε να αντικαταστήσουμε το πλήρες συνδεδεμένο δίκτυο MLP που βρίσκεται τοποθετημένο στο τέλος του υπολειπόμενου δικτύου με ένα άλλο δίκτυο που να εκπαιδεύεται με επιβλεπόμενη μάθηση αφού έχουμε στην διάθεση μας τα επιθυμητά αποτελέσματα . Μια τέτοια αλλαγή θα μπορούσε να βελτιώσει τα ποσοστά επιτυχίας αλλά και να μειώσει τον χρόνο εκτέλεσης που απαιτείται. Ένα συγκεκριμένο δίκτυο που μπορεί να χρησιμοποιηθεί είναι το δίκτυο συναρτήσεων αξονικών βάσεων (Radial Basis Function-

RBF) το οποίο είναι γνωστό ότι μπορεί να λύσει δύσκολά προβλήματα, και συγκεκριμένα προβλήματα παλινδρόμησης, επομένως και προβλήματα πρόβλεψης, με λιγότερο χρόνο εκτέλεσης και λιγότερη χωρητικότητα μνήμης σε σύγκριση με το πλήρες συνδεδεμένο δίκτυο MLP. Συνεπώς συνδυάζοντας αυτές τις αρχιτεκτονικές νευρωνικών δικτύων θα μπορούσαμε να πετύχουμε ψηλότερα αποτελέσματα σε λιγότερο χρονικό διάστημα με μικρότερες χωρητικές απαιτήσεις.

Ακόμη ένα σημείο το οποίο είναι απαραίτητο να γινεί είναι η διαδικασία 10-fold cross-validation. Αυτή η διαδικασία της διασταυρωμένης επικύρωσης είναι αναγκαία να γινεί διότι όπως έχει αναφερθεί στο υποκεφάλαιο 4.2 είναι μια από τις τεχνικές επικύρωσης που αξιολογούν τον τρόπο γενίκευσης των αποτελεσμάτων που θα έχει το δίκτυο όταν χρησιμοποιήσει σαν είσοδο ένα άγνωστο σύνολο δεδομένων. Βάσει αυτής της τεχνικής ξεκινήσαμε την διαδικασία των πειραμάτων για την βελτιστοποίηση και θα ήταν καλό να ολοκληρωθεί. Δυστυχώς λόγω περιορισμένου χρόνου και του γεγονότος ότι η εκπαίδευση του υπολειπόμενου δικτύου ήταν αρκετά χρονοβόρα, λόγω αρκετού βάθους που είχε, έχουν εκτελεστεί πειράματα μόνο σε δύο υποδείγματα του συνόλου CB513. Συγκεκριμένα στην αρχή χρησιμοποιήσαμε το αρχείο fold1 παράγοντας ποσοστό ακρίβειας Q3 72.79% και έπειτα χρησιμοποιήθηκε και το αρχείο fold8 παράγοντας ποσοστό ακρίβειας Q3 74.06%. Επομένως θα πρέπει να χρησιμοποιηθούν όλα τα αρχεία δεδομένων για να παραχθεί το γενικό και σωστό ποσοστό ακρίβειας αφού το κάθε αρχείο παράγει διαφορετικό αποτέλεσμα. Σωστό θα ήταν λοιπόν να εκτελεστούν και τα υπόλοιπα αρχεία με τις τελικές τιμές των παραμέτρων που ορίσαμε βάσει των πειραμάτων που έχουν γινεί ούτως ώστε να παραχθεί ένα δίκαιο αποτέλεσμα. Δίκαιο με την έννοια ότι δεν επιλέξαμε εμείς τον διαχωρισμό των δεδομένων για να παραχθούν κάποια βέλτιστα αποτελέσματα, πιο συγκεκριμένα να έχουμε επιλέξει τις δύσκολες ακολουθίες αμινοξέων για εκπαίδευση και τις εύκολες για επαλήθευσης. Με το 10-fold cross-validation αναγκαστικά θα γινεί ένα τυχαίο ανακάτεμα έτσι ώστε όλα τα δεδομένα να περάσουν και από την διαδικασία εκπαίδευσης και από την διαδικασία επαλήθευσης και να παραχθεί ένα σωστό αποτέλεσμα.

Επίσης, θα μπορούσαμε να εκτελέσουμε επιπρόσθετα πειράματα στα οποία να αλλάζαμε την τιμή των επιπέδων που θα παρακάμπτονταν. Συγκεκριμένα στο υφιστάμενο δίκτυο υπήρξαν 10 υπολειπόμενα μπλόκς στα οποία περιείχαν 2

συνελικτικά στρώματα. Επομένως θα ήταν καλό να υπάρξουν πειράματα με διαφορετικές τιμές των επιπέδων που βρίσκονται στα υπολειπόμενα μπλοκς. Με άλλα λόγια, να τροποποιήσουμε την κατεύθυνση των συνδέσεων παράκαμψης. Για παράδειγμα η τιμή εισόδου του πρώτου επιπέδου να μην πηγαίνει απευθείας στο επόμενο επίπεδο αλλά να πηγαίνει απευθείας στο 5^{ο} επίπεδο ή στο προτελευταίο επίπεδο του δικτύου για να δούμε τυχόν αλλαγές στο ποσοστό ακρίβειας Q3.

Ένα άλλο σημείο το οποίο θα μπορούσε να δοκιμαστεί είναι να αντικατασταθούν τα δεδομένα εισόδου με άλλα. Συγκεκριμένα χρησιμοποιηθήκαν τα αρχεία στα οποία χρησιμοποιήθηκε μέγεθος παραθύρου ίσο με 15. Δηλαδή λαμβάναμε υπόψη μαζί με το αμινοξύ που εξετάζαμε τα 15 γειτονικά αμινοξέα του όπως αναφέραμε στο υποκεφάλαιο 3.1.2.4. Θα ήταν καλό να χρησιμοποιηθούν αρχεία στα οποία να έχουν διαφορετικό αριθμό γειτονικών αμινοξέων για τυχόν αύξηση του ποσοστού ακρίβειας Q3.

Επιπρόσθετα, θα μπορούσαμε να αλλάξουμε τον τρόπο αναπαράστασης των δεδομένων εισόδων του δικτύου. Όπως αναφέρθηκε στο υποκεφάλαιο 3.1.2.1, τα CNNs είναι σε θέση να αναλύουν εισόδους τύπου εικόνας. Το κύριο εμπόδιο στην προσπάθεια επίλυσης ενός σύνθετου προβλήματος ταξινόμησης διαδοχικών δεδομένων με CNNs είναι η αναπαράσταση των δεδομένων, με τέτοιο τρόπο ώστε το δίκτυο να είναι σε θέση όχι μόνο να κατανοήσει το σχήμα της εισόδου, αλλά και να μπορεί να κατανοήσει τις σύνθετες αλληλουχίες που έχουν μεταξύ τους. Άρα για να μπορεί ένα συνελικτικό δίκτυο να προβλέψει την δευτεροταγή δομή της συγκεκριμένης πρωτεΐνης πρέπει η πρωτοταγής δομή να μετατραπεί σε μια μορφή η οποία να αντιπροσωπεύει μια δομή εικόνας. Όπως αναφέρθηκε προηγουμένως ο Διονυσίου (2018) δημιούργησε ένα τρόπο αναπαράστασης ούτως ώστε να δημιουργηθεί μια είσοδος τύπου εικόνας για να μπορέσει το υφιστάμενο δίκτυο να επεξεργαστεί αυτές τις αλληλουχίες. Όμως θα μπορούσαμε να αλλάξουμε αυτό τον τρόπο αναπαράστασης σε κάποιο άλλο καλύτερο, αφού έχει τεράστιο ρόλο ο τρόπος αναπαράστασης των δεδομένων σε ένα συνελικτικό δίκτυο, για να δούμε αν μπορεί το ποσοστό ακρίβειας Q3 να πάρει μια πιο ψηλή τιμή.

Θα ήταν καλό να υπολογιστεί και η μετρική SOV (Segment OVerlap) (Rost et al., 1994; Zelma et al., 1999) η οποία αυτή δίνει μια πιο σωστή ακρίβεια πρόβλεψης της

δευτεροταγούς δομής μιας πρωτεΐνης. Η μετρική αυτή είναι βασισμένη στο μέσο όρο επικάλυψης των ακολουθιών της πραγματικής και της επιθυμητής δευτεροταγούς δομής της πρωτεΐνης. Πιο συγκεκριμένα, η μέθοδος SOV δεν συγκρίνει αμινοξικά κατάλοιπα ένα προς ένα ανάμεσα στην πραγματική και στην επιθυμητή δευτεροταγή δομή, αλλά συγκρίνει διαστήματα από αμινοξικά κατάλοιπα τα οποία επικαλύπτονται στις ακολουθίες της επιθυμητής και της πραγματικής ακολουθίας αντίστοιχα.

Σημαντικό θα ήταν επίσης να εκτελεστούν πειράματα με μεγαλύτερα σύνολα δεδομένων πρωτεϊνών, κάτι το οποίο είναι εξαιρετικά χρονοβόρο αλλά απαραίτητο. Χρησιμοποιώντας λοιπόν μεγαλύτερα σύνολα δεδομένων, για παράδειγμα το σύνολο PISCES το οποίο περιέχει 8500 πρωτεϊνικές ακολουθίες, θα μπορούσε το δίκτυο να δώσει καλύτερες προβλέψεις για το πρόβλημα PSSP.

Τέλος, όσο αφορά το υφιστάμενο υπολειπόμενο νευρωνικό δίκτυο, θα μπορούσε να δοκιμαστεί να αφαιρεθούν οι 2 περιττές διαστάσεις στην είσοδο του δικτύου (από 4D σε 2D πίνακα) αφού η είσοδος βάσει των MSA είναι ένας δυσδιάστατος πίνακας. Με την αφαίρεση αυτών των διαστάσεων μπορεί να ευκολυνθεί το δίκτυο να μάθει αλλά και να μειωθεί και ο χρόνος εκπαίδευσης αφού οι πράξεις συνέλιξης που θα γίνονται στα φίλτρα και στις συγκεκριμένες περιοχές εισόδου θα είναι σε πιο μικρή διάσταση, δηλαδή από 4D πίνακα σε 2D πίνακα. Επίσης θα μπορούσαν να δοκιμαστούν περισσότερα πειράματα αυξάνοντας περισσότερο το βάθος του δικτύου αφού το υπολειπόμενο δίκτυο είναι δημοφιλές στο ότι μπορεί να έχει ένα εξαιρετικό μεγάλο βάθος χωρίς να επηρεάσει την απόδοση του εξάγοντας και αρκετά ψηλά αποτελέσματα. Γενικότερα στο μέλλον θα ήταν καλά να αντικατασταθεί το υφιστάμενο υπολειπόμενο δίκτυο από κάποιο άλλο υπολειπόμενο δίκτυο το οποίο να έχει την δυνατότητα εύκολης εισαγωγής των διαθέσιμων δεδομένων εισόδου για να παραχθούν πιο βέλτιστα ποσοστά επιτυχίας για το συγκεκριμένο πρόβλημα που ερευνήθηκε σε αυτή την διπλωματική.

Αναφορές

Αγαθοκλέους Μ. (2009) , Πρόβλεψη Δευτεροταγούς Δομής Πρωτεϊνών με την χρήση Νευρωνικών Δικτύων Αμφίδρομης Ανάδρασης, Προπτυχιακή διπλωματική, Πανεπιστήμιο Κύπρου, Λευκωσία, 2009.

Δημητρίου Π. (2018) , Πρόβλεψη δευτεροταγούς δομής πρωτεϊνών με τη χρήση clockwork νευρωνικών δικτύων αμφίδρομής ανάδρασης. Προπτυχιακή διπλωματική, Τμήμα Πληροφορικής Πανεπιστήμιο Κύπρου.

Διονυσίου Α. (2018) , Πρόβλεψη δευτεροταγούς δομής πρωτεϊνών με τη χρήση convolutional neural networks σε συνδυασμό με gabor filters, Προπτυχιακή διπλωματική, Τμήμα Πληροφορικής Πανεπιστήμιο Κύπρου.

Παυλίδης Π. (2016), Πρόβλεψη Δευτεροταγούς Δομής Των Πρωτεϊνών Με Τη Χρήση Των Convolutional Neural Networks Για Οπτική Αναγνώριση Αντικειμένων, Προπτυχιακή διπλωματική, Πανεπιστήμιο Κύπρου.

Χριστοδούλου Γ, (2010) , Διερεύνηση Μεθόδων Εκπαίδευσης Νευρωνικών Δικτύων Αμφίδρομης Ανάδρασης για Πρόβλεψη Δευτεροταγούς Δομής Πρωτεϊνών, Προπτυχιακή διπλωματική, Τμήμα Πληροφορικής Πανεπιστήμιο Κύπρου, Λευκωσία.

Adit Deshpande (2017) , A Beginner's Guide To Understanding Convolutional Neural Networks, , Retrieved February 9, 2019, URL: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

P. Baldi, P. Brunak, S. Frasconi, P., Soda, G. and Polastri, G, (1999) , Exploiting the past and the future in protein secondary structure prediction. Bioinformatics, 15, 937–946.

Bengio, Y. (2009) , Learning deep architectures for AI. Foundations and trends in Machine Learning, 2(1), 1-127.

Chen J. and Chaudhari N. S. (2007) , Cascaded Bidirectional Recurrent Neural Networks for Protein Secondary Structure Prediction, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 14(4), 572-582.

Dionysiou, A., Agathocleous, M., Christodoulou, C. and Promponas, V. (2018). Convolutional Neural Networks in combination with Support Vector Machines for complex sequential data classification. Artificial Neural Networks and Machine Learning - ICANN 2018, Lecture Notes in Computer Science, ed. by V. Kurkova, Y. Manolopoulos, B. Hammer, L. Iliadis, I. Maglogiannis, Cham: Springer, 11140, Part II, 444-455.

Eldan, R., & Shamir, O. (2016) , The power of depth for feedforward neural networks. Columbia University, New-York City, USA, 907-940.

He, K., Zhang, X., Ren, S., & Sun, J. (2016a) , Deep residual learning for image recognition , IEEE- computer vision and pattern recognition, Jun 26, 2016 – Jul 1, 2016, Las Vegas, Nevada, United States, 770-778.

He, K., Zhang, X., Ren, S., & Sun, J. (2016b) , Identity mappings in deep residual networks, ECCV2016 - Computer Vision, Amsterdam, The Netherlands, 630-645.

He, K. (2016) , Deep Residual Networks-Deep Learning Gets Way Deeper.

Huang, G., Liu Z., van der Maaten, L. and Weinberger ,K. Q. (2016) , Densely Connected Convolutional Networks. arXiv preprint, arXiv:1608.06993

Hubel, D. H., & Wiesel, T. N. (1962) , Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology, 160(1), 106-154.

King RD and Sternberg MJ. (1996) , Identification and application of the concepts important for accurate and reliable protein secondary structure prediction, Protein Sci, 5(11), 298-310.

Koutnik, J., Greff, K., Gomez, F., & Schmidhuber, J. (2014) , A clockwork rnn. arXiv preprint arXiv:1402.3511.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25: Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada. F. Pereira, C.J.C. Burges, L. Bottou and K.Q. Weinberger, Eds. Red Hook, NY: Curran Associates, pp. 1097–1105.

LeCun, Y., Bengio, Y., & Hinton, G. (2015) , Deep Learning. Nature, 521(7553), 436.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998) , Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

McCulloch, W. S., & Pitts, W. (1943) , A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.

Minsky, M. & Papert, S. (1969). Perceptrons: An Introduction to Computational Geometry, The MIT Press, Cambridge, MA

Nat Roth , (2016) Quora Question “What exactly is the degradation problem that Deep Residual Networks try to alleviate?”, Microsoft, https://www.quora.com/What-exactly-is-the-degradation-problem-that-Deep-Residual-Networks-try-to-alleviate.

Orhan, A. E., & Pitkow, X. (2017) , Skip connections eliminate singularities. arXiv preprint arXiv:1701.09175.

Qian, N. and Sejnowski, TJ. (1988) , Predicting the secondary structure of globular proteins using neural network models, Journal of Molecular Biology, 202(4), 865-884.

Rosenblatt, F. (1958) , The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386-392.

Rost, B. and Sander, C. (1993) , Improved prediction of protein secondary structure by use of sequence profiles and neural networks, Proceedings of the National Academy of Sciences of the United States of America - PNAS, 90, 7558-7562.

Rost, B., Sander, C., and Schneider, R. (1994). Redefining the goals of protein secondary structure prediction, Journal of Molecular Biology, 235, 13–26.

Simonyan, K., & Zisserman, A. (2014) , Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Saha S., (2018) , A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way, Towards Data Science, , Retrieved February 3, 2019, URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Wang Chi-Feng, (2018) , The Vanishing Gradient Problem, The Problem, Its Causes, Its Significance, and Its Solutions, Toward Data Science, Retrieved February 16, 2019, URL: https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Van Gool, L. (2016). Temporal segment networks: Towards good practices for deep action recognition. In Proceedings of the European conference on Computer Vision, Computer Vision - ECCV 2016, Part VIII, Lecture Notes in Computer Science, 9912, Ed. by B. Leibe, J. Matas, N. Sebe and M. Welling, Springer, Cham, 20-36.

Wang, S., Peng, J., Ma, J. and Xu, J. (2016). Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields, Scientific Reports, 6, 18962.

Yeung, D. S., Cloete, I., Shi, D., & Ng, W. W. (2009). Principles of Sensitivity Analysis. In: Sensitivity Analysis for Neural Networks. Natural Computing Series. Springer, Berlin, Heidelberg, (pp. 17-24).

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In Proceedings of the European conference on computer vision, Computer Vision – ECCV 2014. ECCV 2014, Part I, Lecture Notes in Computer Science, 8689, Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds), Springer, Cham, 818-833.

Zemla, A.,Venclovas, C., Fidelis, K. and Rost, B. (1999). A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment, Proteins, 34, 220–223.

Βιβλιογραφία

Allan Handan, Deep Learning: Convolutional Neural Networks, Apr 11 2018, Retrieved January 13, 2019, https://labs.bawi.io/deep-learning-convolutional-neural-networks-7992985c9c7b

Baki Er, Microsoft Presents : Deep Residual Networks, Aug 10, 2016 , Retrieved January 13, 2019 https://medium.com/@bakiiii/microsoft-presents-deep-residual-networks-d0ebd3fe5887

BioNinja, Proteins, Retrieved March 20, 2019, Retrieved January 13, 2019 http://www.vce.bioninja.com.au/aos-1-molecules-of-life/biomolecules/proteins.html

Carlos E. Perez, Author - The Deep Learning Playbook and Artificial Intuition, Jun 3 2016, Retrieved January 13, 2019 https://www.quora.com/How-does-deep-residual-learning-work

Coursera, ResNets,Convolutional Neural Networks, Retrieved January 5, 2019 https://www.coursera.org/lecture/convolutional-neural-networks/resnets-HAhz9

Coursera, ResNets, Retrieved January 5, 2019 https://www.coursera.org/lecture/convolutional-neural-networks/resnets-HAhz9

Deep Residual Networks, 19 June 2016, Retrieved January 5, 2019 https://icml.cc/2016/tutorials/icml2016_tutorial_deep_residual_networks_kaiminghe.pdf

Diadrastika Sxolika Vivllia, ebooks.edu.gr, Makromoria, 2013, Retrieved February 3, 2019, http://ebooks.edu.gr/modules/ebook/show.php/DSGL-B106/726/4800,21687/

Dietz Michael (2017), Understand Deep Residual Networks—a simple, modular learning framework that has redefined state-of-the- art, 24/12/2018,

https://blog.waya.ai/deep-residual-learning-9610bb62c355

Kallipos, Μηχανική Μάθηση, Retrieved February 3, 2019, http://repfiles.kallipos.gr/html_books/93/04a-main.html

Kaiming He , Deep Residual Networks, Deep Learning Gets Way Deeper, June 19 2016, Retrieved January 5, 2019 https://icml.cc/2016/tutorials/icml2016_tutorial_deep_residual_networks_kaiminghe.pdf

Meta-Learning Mania, Pre-activation in Neural Networks, January 4,2017, https://learningstracker.wordpress.com/2017/01/04/pre-activation-in-neural-networks/

Prakash Jay,Understanding and Implementing Architectures of ResNet and ResNeXt for state-of-the-art Image Classification: From Microsoft to Facebook [Part 1], Feb 7, 2019, https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624

Siddharth Das, CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more…, Nov 2016,2017, Retrieved January 5, 2019 https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

Spanoy Dimitra, Η σημασία των βιομορίων στον ανθρώπινο οργανισμό. Μέρος Ένατο. Πρωτείνες και ορισμένες ιδιότητές τους, 2014, Retrieved March 20, 2019 http://m.dimitra-spanoy.webnode.gr/products/i-simasia-ton-viomorion-ston-anthropino-organismo-oi-proteines/

Sumit Saha, A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way, Dec 15 2018, Towards Data Science, Retrieved March 20, 2019 https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Understand Deep Residual Networks—a simple, modular learning framework that has redefined state-of-the-art , May 2, 2017, Retrieved January 5, 2019 https://blog.waya.ai/deep-residual-learning-9610bb62c355

Παράρτημα Α

Εγκατάσταση και εκτέλεση υπολειπόμενου νευρωνικού δικτύου:

Εγκατάσταση στον υπολογιστή-Install

Εξαρτήσεις:

1. Python version
    a. Python version used in this project: 3.5+
2. TensorFlow 1.2.0
3. Numpy 1.10.4

Ο κώδικας έχει 2 μορφές οι οποίες έχουν τις εξής ονομασίες:

1. ResNet_python.ipynb
2. ResNet_python.py

Εκτέλεση-How to run

Για να τρέξετε το αρχείο με ονομασία ResNet_python.py τρέχετε την εξής εντολή στο terminal/cmd:
python ResNet_python.py

Για να τρέξετε το αρχείο με ονομασία ResNet_python.ipynb τρέχετε την εξής εντολή στο anaconda terminal/cmd:

ipython notebook ResNet_python.ipynb
ή
jupyter notebook ResNet_python.ipynb

Μπορείτε επίσης να τρέξετε το αρχείο αυτό σε πιο φιλικό περιβάλλον και κομμάτι-κομμάτι του κώδικα με τα εξής βήματα:

1. Κατεβάστε την κονσόλα Anaconda terminal
2. Ανοίξτε το anaconda terminal και δημιουργήστε ένα δικό σας περιβάλλον σε αυτή την κονσόλα με την εξής εντολή: conda create –name myenv

3. Για να βρεθείτε μέσα στο περιβάλλον που δημιουργήσατε γράψτε την εντολή: activate myenv
4. Σε αυτό το περιβάλλον πρέπει να έχετε κατεβάσει τις βιβλιοθήκες που αναφερθήκαν πιο πάνω.
5. Τώρα που βρίσκεστε μέσα στο περιβάλλον πληκτρολογήστε την εντολή: jupyter notebook
6. Αυτόματα ένας local server εμφανίζεται στον by default browser σας και μια λίστα από τα αρχεία που έχετε στον υπολογιστή
7. Κατευθυντήτε στο αρχείο που έχετε αποθηκέυσει το αρχείο με την ονομασία ResNet_python.ipynb .
8. Θα εμφανιστεί μια άλλη σελίδα όπου θα βρίσκεται ο κώδικας σε διάφορα μπλοκς. Μπόρετε να τρέξετε κάθε μπλοκ με το κουμπί > ή πατώντας το Ctr + Enter ή Shift+ Enter.

Παράρτημα Β

Υλοποίηση βαθιού υπολειπόμενου νευρωνικού δικτύου βάση σε γλώσσα Python και βάσει της γνωστής βιβλιοθήκης Tensorflow.

Οι αλλαγές που υπήρξαν στον κώδικα απεικονίζονται με το χρώμα μοβ.

Ο πιο κάτω πίνακας περιέχει σημαντικούς όρους που βρίσκονται στο κώδικα υλοποίησης για καλύτερη κατανόηση.

Ονομασία	Περιγραφή
Matrix	Ένα πλέγμα (grid).Π.χ. ένας δυσδιάστατος ή τρισδιάστατος πίνακας.
Vector	Ένας πίνακας με μόνο μια γραμμή ή στήλη (διάνυσμα)
Tensor	Γενική δομή αποθήκευσης δεδομένων με μεγαλύτερες διαστάσεις. Π.χ. ένας πίνακας 4D ή 5D.
Variables	Οι μεταβλητές χρησιμοποιούνται για την αποθήκευση της κατάστασης ενός γραφήματος. Οι μεταβλητές πρέπει να αρχικοποιηθούν με μία τιμή κατά την δήλωση τους.
Placeholders	Χρησιμοποιούνται για την τροφοδοσία εξωτερικών δεδομένων στον γράφο. Επιτρέπει την εκχώρηση μιας τιμής αργότερα, δηλ. Μιας θέσης στη μνήμη, όπου αργότερα θα αποθηκεύουμε μια τιμή.
None ή -1	Δυναμικός καθορισμός διάστασης. Η διάσταση που έχει τιμή None ή -1 μπορεί να πάρει οποιαδήποτε τιμή (δυναμική) βάση της άλλες διαστάσεις που έχουν οριστεί.
Conv2d	Πράξη συνέλιξης ενός 4D πίνακα εισόδου με 4D φίλτρου.
Dropout	Τεχνική κανονικοποίησης που απορρίπτει αχρείαστα δεδομένα, βάση μιας πιθανότητας, για να μην υπάρξει υπερφόρτωση στο δίκτυο.
Flatten	Μετατροπή διαστάσεων από 4D πίνακα σε 2D πίνακα και έπειτα σε 1D για να εισαχθεί στο πλήρες συνδεδεμένο επίπεδο.
filter_size	Μέγεθος φίλτρου που θα διασχίσει τα δεδομένα εισόδου
num_of_channels	Αριθμός καναλιών που περιέχει η εικόνα (=1- είναι μαυρόασπρη)

num_of_filters	Αριθμός παράλληλων φίλτρων
Dense_custom	Ένα πλήρες συνδεδεμένο επίπεδο (Fully connected layer). Μια γραμμική λειτουργία στην οποία κάθε είσοδος συνδέεται σε κάθε έξοδο με ένα βάρος. Γενικά ακολουθείται από μια μη γραμμική λειτουργία ενεργοποίησης
matmul	Πράξη πολλαπλασιασμού μεταξύ 2 πινάκων.
Batch normalization	Η κανονικοποίηση παρτίδας είναι μαθηματικές πράξει που βοηθούν στην πιο γρήγορη εκμάθηση παράγοντας υψηλότερα ποσοστά επιτυχίας. Επίσης, η κανονικοποίηση παρτίδων επιτρέπει σε κάθε στρώμα ενός δικτύου να μάθει από μόνο του λίγο πιο ανεξάρτητα από τα άλλα επίπεδα.
argmax	Επιστρέφει τον δείκτη με τη μεγαλύτερη τιμή που έχει ένας πίνακας/tensor.
AdamOptimizer	Μέθοδος Ενημέρωσης του Ρυθμού Μάθησης (Updater) – Αλλαγές τιμών των βαρών βάση του αλγόριθμου Αδάμ.
Dropout	Απόρριψη (απομάκρυνση - drop) ορισμένων νευρώνων ανάλογα με μια πιθανότητα για να μειωθεί η υπερφόρτωση και να βελτιωθεί το σφάλμα γενίκευσης.
Session	Περίοδος λειτουργίας που επιτρέπει την εκτέλεση γράφων. (εκκίνηση εκτέλεσης κώδικα- όπως την συνάρτηση main στην Java)

```
1.  #!/usr/bin/env python
2.  # coding: utf-8
3.
4.  #
5.  #
6.  # Network depth is of crucial importance in neural network architectures, but
    deeper networks are more difficult to train. The residual learning framework e
    ases the training of these networks, and enables them to be substantially deep
    er – leading to improved performance in both visual and non-
    visual tasks. These residual networks are much deeper than their ‘plain’ count
    erparts, yet they require a similar number of parameters (weights).
7.  #
8.  # Materials:
9.  #
10. # [Deep Residual Learning for Image Recognition](https://arxiv.org/pdf/1512.03
    385.pdf)
11. #
12. # [Identity Mappings in Deep Residual Networks](https://arxiv.org/pdf/1603.050
    27.pdf)
```

```
#
# This [Blog post](https://blog.waya.ai/deep-residual-learning-9610bb62c355) is great for intuition behind ResNet.
#
# ![](paper_net.png)

# In[1]:


import tensorflow as tf
import numpy as np
import time


# ### Step 1. Define helper functions

# In[2]:


def weights_init(shape):
    '''''
    Weights initialization helper function.

    Input(s): shape - Type: int list, Example: [5, 5, 32, 32], This parameter is used to define dimensions of weights tensor

    Output: tensor of weights in shape defined with the input to this function
    '''
    return tf.Variable(tf.truncated_normal(shape, stddev=0.05))


# In[3]:


def bias_init(shape, bias_value=0.01):
    '''''
    Bias initialization helper function.

    Input(s): shape - Type: int list, Example: [32], This parameter is used to define dimensions of bias tensor.
              bias_value - Type: float number, Example: 0.01, This parameter is set to be value of bias tensor.

    Output: tensor of biases in shape defined with the input to this function
    '''
    return tf.Variable(tf.constant(bias_value, shape=shape))


# In[4]:


def conv2d_custom(input, filter_size, num_of_channels, num_of_filters, activation=tf.nn.relu, dropout=None,
                  padding='VALID', max_pool=True, strides=(1, 1)):
    '''''
    This function is used to define a convolutional layer for a network,

    Input(s): input - this is input into convolutional layer (Previous layer or an image)
```

```
66.                 filter_size -
     also called kernel size, kernel is moved (convolved) across an image. Example
    : 3
67.                 number_of_channels - how many channels the input tensor has
68.                 number_of_filters -
     this is hyperparameter, and this will set one of dimensions of the output ten
    sor from
69.                                     this layer. Note: this number will be number
    _of_channels for the layer after this one
70.                 max_pool -
     if this is True, output tensor will be 2x smaller in size. Max pool is there
    to decrease spartial
71.                           dimensions of our output tensor, so computation is les
    s expensive.
72.                 padding -
     the way that we pad input tensor with zeros ("SAME" or "VALID")
73.                 activation - the non-linear function used at this layer.
74.
75.
76.     Output: Convolutional layer with input parameters.
77.     '''
78.     weights = weights_init([filter_size, filter_size, num_of_channels, num_of_
    filters])
79.     bias = bias_init([num_of_filters])
80.
81.     layer = tf.nn.conv2d(input, filter=weights, strides=[1, 1, 1, 1], padding=
    padding) + bias #strides=[1,1,1,1]
82.
83.     if activation != None:
84.         layer = activation(layer)
85.
86.     if max_pool:
87.         layer = tf.nn.max_pool(layer, ksize=[1, 1, 1, 1], strides=[1, 1, 1, 1]
    , padding='VALID')
88.
89.     if dropout != None:
90.         layer = tf.nn.dropout(layer, dropout)
91.
92.     return layer
93.
94.
95. # In[5]:
96.
97.
98. def flatten(layer):
99.     '''''
100.           This method is used to convert convolutional output (4 dimensional
    tensor) into 2 dimensional tensor.
101.
102.           Input(s): layer -
     the output from last conv layer in your network (4d tensor)
103.
104.           Output(s): reshaped - reshaped layer, 2 dimensional matrix
105.                      elements_num - number of features for this layer
106.           '''
107.           shape = layer.get_shape()
108.
109.           num_elements_ = shape[1:4].num_elements()#what the?
110.
111.           flattened_layer = tf.reshape(layer, [-1, num_elements_])
112.           return flattened_layer, num_elements_
113.
114.
115.       # In[6]:
116.
```

```
117.
118.       def dense_custom(input, input_size, output_size, activation=tf.nn.relu,
    dropout=None):
119.           '''''
120.           This function is used to define a fully connected layer for a netwo
    rk,
121.
122.           Input(s): input -
    this is input into fully connected (Dense) layer (Previous layer or an image)

123.                     input_size -
    how many neurons/features the input tensor has. Example: input.shape[1]
124.                     output_shape - how many neurons this layer will have
125.                     activation - the non-
    linear function used at this layer.
126.                     dropout -
    the regularization method used to prevent overfitting. The way it works, we r
    andomly turn off
127.                                 some neurons in this layer
128.
129.           Output: fully connected layer with input parameters.
130.           '''
131.           weights = weights_init([input_size, output_size])
132.           bias = bias_init([output_size])
133.
134.           layer = tf.matmul(input, weights) + bias
135.
136.           if activation != None:
137.               layer = activation(layer)
138.
139.           if dropout != None:
140.               layer = tf.nn.dropout(layer, dropout)
141.
142.           return layer
143.
144.
145.       # The resunit implemented in this notebook is explained in this [paper]
    (https://arxiv.org/pdf/1603.05027.pdf).
146.       #
147.       # This is a picutre of the resunit used:
148.       #
149.       # ![](resunit.jpeg?raw=true)
150.       #
151.       # Note: implemented version is B
152.
153.       # In[7]:
154.
155.
156.       def residual_unit(layer):
157.           '''''
158.           Input(s): layer - conv layer before this res unit
159.
160.           Output(s): ResUnit layer - implemented as described in the paper
161.           '''
162.           step1 = tf.layers.batch_normalization(layer)
163.           step2 = tf.nn.relu(step1)
164.           step3 = conv2d_custom(step2, 1, 32, 32, activation=None, max_pool=F
    alse) #32 number of feautres is hyperparam
165.           step4 = tf.layers.batch_normalization(step3)
166.           step5 = tf.nn.relu(step4)
167.           step6 = conv2d_custom(step5, 1, 32, 32, activation=None, max_pool=F
    alse)
168.           return layer + step6
169.
170.
```

```
171.        # ### Step 2. Residual Network (ResNet)
172.
173.        # In[8]:
174.
175.
176.        inputs = tf.placeholder(tf.float32, [None,15,20,1], name='inputs')# for
     mnist the inputs have the shape[None, 28, 28, 1]
177.        targets = tf.placeholder(tf.float32, [None,3], name='targets') #for mni
    st the targets were [None, 10]
178.
179.
180.        # In[9]:
181.
182.
183.        num_of_layers = 20
184.        between_strides = num_of_layers/5
185.
186.
187.        # #### This is our network
188.
189.        # In[10]:
190.
191.
192.        prev1 = conv2d_custom(inputs, 3, 1, 32, activation=None, max_pool=False
    )
193.        prev1 = tf.layers.batch_normalization(prev1)
194.        for i in range(5): # this number * between_strides = number_of_layers
195.            for j in range(int(between_strides)):
196.                prev1 = residual_unit(prev1)
197.
198.            prev1 = tf.layers.batch_normalization(prev1)
199.        #after all resunits we have last conv layer, than flattening and output
     layer
200.        last_conv = conv2d_custom(prev1, 3, 32, 3, activation=None, max_pool=Fa
    lse)
201.        flat, features = flatten(last_conv)
202.        output = dense_custom(flat, features, 3, activation=None)
203.
204.
205.        # In[11]:
206.
207.
208.        #This part is for computing the accuracy of this model
209.        pred_y = tf.nn.softmax(output)
210.        pred_y_true = tf.argmax(pred_y, 1)
211.        y_true = tf.argmax(targets, 1)
212.        correct_prediction = tf.equal(pred_y_true, y_true)
213.        accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))*100

214.
215.
216.        # In[12]:
217.
218.
219.        # loss function and optimizer
220.        cost = tf.reduce_mean((tf.nn.softmax_cross_entropy_with_logits(logits=o
    utput, labels=targets)))
221.        optimizer = tf.train.AdamOptimizer(0.001).minimize(cost)
222.
223.
224.        # ### Step 3. Training and testing helper functions
225.
226.        # In[13]:
227.
228.
```

```
229.        # READ THE TRAINING DATA
230.        inputFile=[]
231.        f = open("protein_csv_train_all_15neighbors_technique_fold1.txt", "r")

232.        for x in f:
233.            inputFile.append(x)
234.
235.        results=[]
236.        splitInputFile=[]
237.        #split string to small string according to semicolon's position
238.        splitInputFile=[i.split(",") for i in inputFile]
239.        #converting the string values to integer values
240.        intInputs=[[int(i) for i in j] for j in splitInputFile]
241.        #desired outputs in 1D list
242.        labels=[]
243.        labels=[intInputs[i][300]for i in range(len(intInputs))]#300 the last e
    lement
244.
245.        #one hot encoding labels list
246.        encodingLabels=[]
247.        for i in range(len(labels)):
248.            if labels[i]==0:
249.                encodingLabels.append(0)
250.                encodingLabels.append(0)
251.                encodingLabels.append(1)
252.            elif labels[i]==1:
253.                encodingLabels.append(0)
254.                encodingLabels.append(1)
255.                encodingLabels.append(0)
256.            else:
257.                encodingLabels.append(1)
258.                encodingLabels.append(0)
259.                encodingLabels.append(0)
260.        #3 cols for each tensor_labels row
261.        final_labels=tf.reshape(encodingLabels,[-1,3])
262.
263.        #copy the list to delete the last col where the labels are
264.        finalInput=intInputs.copy()
265.        #axis y-->cols
266.        axis=1
267.        #delete the last element from each road(desired output)#we don't need t
    hem because we save them in the labels list
268.        #we have 301 elements in each row. That means we have 301 cols in each
    row
269.        #We have to delete the last col which is the 300th(cols start with 0 (0
    -300=301 cols))
270.        finalInput=np.delete(finalInput,300,axis)
271.        #because of numpy fuction delete we have to convert the array to a list
    (not necessary)
272.        finalInput=finalInput.tolist()
273.        #convert final inputs list to a tensor
274.        tensor_list=tf.convert_to_tensor(finalInput)
275.        #convert final_labels list to a tensor
276.        tensor_labels=tf.convert_to_tensor(final_labels)
277.
278.        batch_size = 7000 #####32
279.        epochs = 40 ###5
280.        train_data_size=tensor_list.shape[0] #77092
281.        period=train_data_size//batch_size #11.01
282.
283.
284.        def optmizer():
285.
286.            for j in (range(epochs)):
287.                epoch_loss = []
```

```
                start_epoch = time.time()
                for i in range(period):
                    batch=tensor_list[i*batch_size:(i+1)*batch_size]
                    batch1=tensor_labels[i*batch_size:(i+1)*batch_size]

                    imgs=tf.reshape(batch,(-1,15,20,1))
                    imgsToFloat=tf.to_float(imgs)
                    imgsF=session.run(imgsToFloat)

                    labels=tf.reshape(batch1,(-1,3))
                    labelsToFloat=tf.to_float(labels)

                    labelsF=session.run(labelsToFloat)

                    dict_input = {inputs:imgsF, targets:labelsF}

                    c, _ = session.run([cost, optimizer], feed_dict=dict_input)
     # cost= training= change weights
                    epoch_loss.append(c)
                print(" {}".format(j+1), " ", (session.run(accuracy, feed_dict=
    dict_input)),
                      " {} ".format(np.mean(epoch_loss)))



        # READ THE TESNING DATA

        inputFileTest=[]
        f = open("protein_csv_test_all_15neighbors_technique_fold1.txt", "r")
        for x in f:
            inputFileTest.append(x)

        resultsTest=[]
        splitInputFileTest=[]
        #split string to small string according to semicolon's position
        splitInputFileTest=[i.split(",") for i in inputFileTest]
        #converting the string values to integer values
        intInputsTest=[[int(i) for i in j] for j in splitInputFileTest]
        #desired outputs in 1D list
        labelsTest=[]
        labelsTest=[intInputsTest[i][300]for i in range(len(intInputsTest))]#30
    0 the last element
        #one hot encoding labels list
        encodingLabelsTest=[]
        for i in range(len(labelsTest)):
            if labelsTest[i]==0:
                encodingLabelsTest.append(0)
                encodingLabelsTest.append(0)
                encodingLabelsTest.append(1)
            elif labelsTest[i]==1:
                encodingLabelsTest.append(0)
                encodingLabelsTest.append(1)
                encodingLabelsTest.append(0)
            else:
                encodingLabelsTest.append(1)
                encodingLabelsTest.append(0)
                encodingLabelsTest.append(0)
        #3 cols for each tensor_labels row
        final_labelsTest=tf.reshape(encodingLabelsTest,[-1,3])

        #copy the list to delete the last col where the labels are
        finalInputTest=intInputsTest.copy()
        #axis y-->cols
        axis=1
```

```
350.        #delete the last element from each road(desired output)#we don't need t
    hem because we save them in the labels list
351.        #we have 301 elements in each row. That means we have 301 cols in each
    row
352.        #We have to delete the last col which is the 300th(cols start with 0 (0
    -300=301 cols))
353.        finalInputTest=np.delete(finalInputTest,300,axis)
354.        #because of numpy fuction delete we have to convert the array to a list
    (not necessary)
355.        finalInputTest=finalInputTest.tolist()
356.        #convert final inputs list to a tensor
357.        tensor_list_test=tf.convert_to_tensor(finalInputTest)
358.        #convert final_labels list to a tensor
359.        tensor_labels_test=tf.convert_to_tensor(final_labelsTest)
360.
361.
362.        test_data_size=tensor_list_test.shape[0] #7289
363.
364.
365.
366.        def test_model():
367.            accuracy_per_batch = []
368.
369.            imgs_test=tf.reshape(tensor_list_test,(-1,15,20,1))
370.            imgsToFloat_test=tf.to_float(imgs_test)
371.            imgsF_test=session.run(imgsToFloat_test)
372.
373.            labels_test=tf.reshape(tensor_labels_test,(-1,3))
374.            labelsToFloat_test=tf.to_float(labels_test)
375.            labelsF_test=session.run(labelsToFloat_test)
376.
377.            accuracy_per_batch.append(session.run(accuracy, feed_dict={inputs:i
    mgsF_test, targets:labelsF_test}))
378.            print("Accuracy {}".format(accuracy_per_batch))
379.
380.
381.        # ### Step 4. Train/Test the network
382.
383.        # In[16]:
384.
385.
386.        session = tf.Session()
387.        session.run(tf.global_variables_initializer())
388.
389.
390.        # In[ ]:
391.
392.
393.        optmizer()
394.
395.
396.        # In[ ]:
397.
398.
399.        test_model()
400.
401.
402.        # In[ ]:
403.
404.
405.        #validate_model()
406.
407.
408.        # In[25]:
409.
```

```
410.
411.        session.close()
412.        #close the session after testing the model
413.
414.
415.        # In[ ]:
```