ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ

«ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ – ΕΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ»

ΣΑΡΑΝΤΙΔΗΣ ΡΩΜΑΝΟΣ ΜΠΠΛ2327

ΠΕΙΡΑΙΑΣ 2024

Περιεχόμενα

Εξώφυλλο 1
Περιεχόμενα 2
Εισαγωγή 4
Πρόβλημα 5
Θεωρία 7
Νευρωνικά Δίκτυα 7
Δομή 7
Λειτουργία 9
Συναρτήσεις Ενεργοποίησης 10
Υπολογισμός Σφάλματος 11
Οπισθόδρομη διάδοση 11
Ανάπτυξη - Σχεδίαση 14
Προετοιμασία Δεδομένων 14
Ανάγνωση 15
Αρχιτεκτονική του Νευρωνικού Δικτύου 16
Εκτέλεση 18
Νευρωνικό δίκτυο 18
Συμπεράσματα 23
Βιβλιογραφία 24

Εισαγωγή

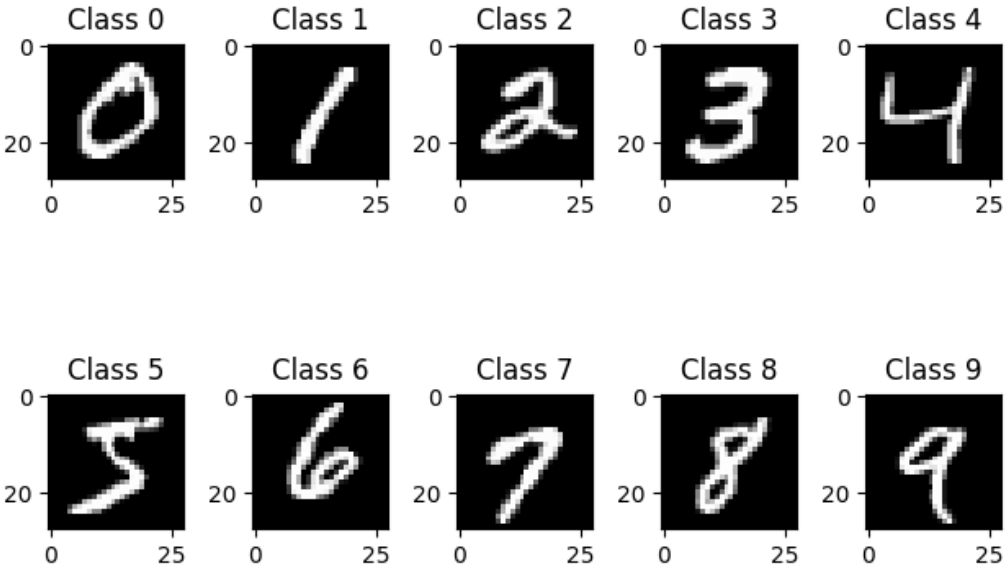
Τα τελευταία χρόνια ο κλάδος της τεχνητής νοημοσύνης έχει σημειώσει ραγδαία τεχνολογική ανάπτυξη. Τα μεγάλα γλωσσικά μοντέλα (Chat-GPT, Gemini) (*Briganti, 2024*), προσωπικοί βοηθοί με τη δυνατότητα αναγνώρισης φωνητικών εντολών (Siri, Google assistant), οχήματα αυτόνομης οδήγησης (Tesla) (*Kumari & Bhat., 2021*), τεχνητοί παίκτες πολύπλοκων παιχνιδιών (Go, Σκάκι) σε τέτοιο βαθμό ώστε να υπερνικούν ακόμα και παγκόσμιους πρωταθλητές σε αυτά (AlphaGo, Deep Blue) (*Silver et al., 2017; Campbell et al., 2002*), είναι μόνο μερικά από τα τεχνολογικά άλματα που βιώνουμε με τη βοήθεια της τεχνητής νοημοσύνης. Ενώ η δυνατότητα της παραγωγής κώδικα ως προϊόν τεχνητής νοημοσύνης προμηνύει ακόμα μεγαλύτερες μελλοντικές εξελίξεις στον παγκόσμιο τεχνολογικό κλάδο (*Nejjar et al., 2024*).

Τα τεχνητά νευρωνικά δίκτυα αποτελούν θεμελιώδες κομμάτι της μηχανικής μαθήσεως. Πρόκειται για ένα υπολογιστικό μοντέλο που είναι εμπνευσμένο από τη δομή και λειτουργία του ανθρώπινου εγκεφάλου. Είναι σχεδιασμένο έτσι ώστε να αναγνωρίζει μοτίβα και να εκτελεί συγκεκριμένες διεργασίες, μαθαίνοντας από μόνο του χωρίς σαφείς κανόνες και οδηγίες για όλα τα πιθανά ενδεχόμενα. Βρίσκει χρήση σε τομείς όπως αναγνώριση σημάτων εικόνας, ήχου και επεξεργασίας φυσικής γλώσσας, αυτόνομα οχήματα, πρόβλεψη οικονομικών δεδομένων, διάγνωση ασθενειών και άλλα.

Η παρούσα εργασία μελετά την λειτουργία ενός νευρωνικού δικτύου κατασκευασμένου για την αναγνώριση χειρόγραφων αριθμητικών ψηφίων (MNIST). Το MNIST παρόλο που έχει αντικατασταθεί από πολλά άλλα σύνολα δεδομένων, υψηλότερης δυσκολίας, αποτελεί κλασσικό σημείο εκκίνησης για μηχανική μάθηση σε νευρωνικά δίκτυα.

Πρόβλημα

Το σύνολο δεδομένων MNIST (Modified National Institute of Standards and Technology Dataset) αποτελείται από χειρόγραφα αριθμητικά ψηφία από το 0 έως το 9. Πρόκειται για 60.000 εικόνες ψηφίων ως πακέτου εκπαίδευσης και 10.000 εικόνες ψηφίων ως πακέτο ελέγχου, που έχει συλλέξει το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας της Αμερικής, από μαθητές και υπαλλήλους (*LeCun et al., 1998*). Το σύνολο των 70.000 εικόνων έχουν τη μορφή ασπρόμαυρων φωτογραφιών, διαστάσεων 28x28 pixel με τιμές από 0-255 για κάθε pixel (8 bit per pixel), προσδιορίζοντας το χρώμα από μαύρο έως λευκό. Τα ψηφία εμφανίζονται με αποχρώσεις του λευκού (255), ενώ το υπόλοιπο πλαίσιο είναι μαύρο (0). Οι εικόνες είναι επισημασμένες με βάση το ψηφίο που απεικονίζουν σε μια από τις 10 κατηγορίες (0-9).



Εικόνα 1. MNIST Dataset δείγμα εικόνων

Ο διαχωρισμός των 70.000 εικόνων σχετίζεται με τον βαθμό δυσκολίας αναγνώρισης των απεικονιζόμενων ψηφίων. Οι 60.000 πρώτες εικόνες είναι περισσότερο ευανάγνωστες και έχουν ως στόχο να εκπαιδεύσουν το μηχανικό μοντέλο, ενώ οι τελευταίες 10.000 είναι περισσότερο δυσανάγνωστες και έχουν ως στόχο να εξετάσουν τις αναγνωριστικές ικανότητες του μοντέλου. Ενώ οι εικόνες είναι ισοδύναμα διαμοιρασμένες στις κλάσεις όπως φαίνεται και από τον ακόλουθο πίνακα (Πίνακας 1).

Πίνακας 1. Περιεχόμενα MNIST Dataset

Κλάση (ψηφίο)	Πακέτο εκπαίδευσης	Πακέτο ελέγχου
0	5.923	980
1	6.742	1.135
2	5.958	1.032
3	6.131	1.010
4	5.842	982
5	5.421	892
6	5.918	958
7	6.265	1.028
8	5.851	974
9	5.949	1.009
Σύνολο (εικόνες):	**60.000**	**10.000**

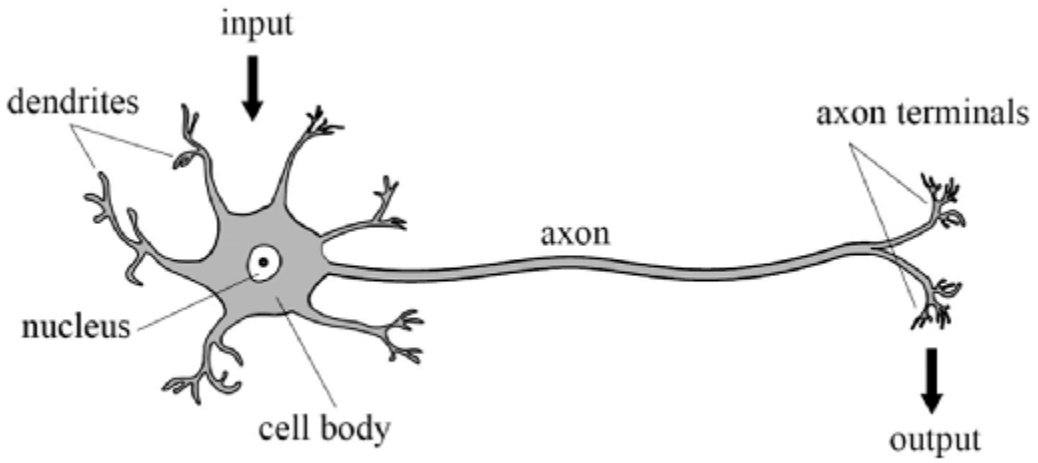
Πλέον είναι διαθέσιμα και άλλα σύνολα δεδομένων όπως το CIFAR-10 (αναγνώριση ζώων και αντικειμένων), το ImageNet (αναγνώριση ζώων και αντικειμένων), fashion-MNIST (αναγνώριση τύπου ρούχων) και λοιπά, ωστόσο είναι βασισμένα στη λογική του MNIST σε πολυπλοκότερο επίπεδο.

Ουσιαστικά είναι ένα πρόβλημα κατηγοριοποίησης εικόνων, που έχει ως στόχο να εκπαιδεύσει και να αξιολογήσει αλγόριθμους αναγνώρισης προτύπων. Αυτό επιτυγχάνεται με τη βοήθεια νευρωνικών δικτύων, τα οποία λαμβάνουν τα δεδομένα περιοδικά τα επεξεργάζονται και αποκτούν ικανότητα αναγνώρισης μοτίβων. Η συγκεκριμένη διαδικασία προϋποθέτει ένα στάδιο εκπαίδευσης και ένα ακόλουθο ελέγχου για την αξιολόγηση της ικανότητας αναγνώρισης. Με αυτό τον σκοπό είναι κατασκευασμένο και αυτή την διαδικασία ακολουθεί με την οργάνωσή του και το σύνολο δεδομένων MNIST. Είναι ένα πρόβλημα με το οποίο έχει ασχοληθεί η επιστημονική κοινότητα έχουν αναπτυχθεί πολλές τεχνικές γι' αυτό και έχει τελειοποιηθεί σε τέτοιο βαθμό, ώστε να επιτυγχάνεται ποσοστό λάθους 0.13% στο στάδιο ελέγχου. Κάτι που ισοδυναμεί με 13 λανθασμένες προβλέψεις από το σύνολο των 10.000 εικόνων του συνόλου ελέγχου (*Byerly et al., 2021*).

Θεωρία

Νευρωνικά Δίκτυα

Τα νευρωνικά δίκτυα έχουν κατασκευαστεί καθ' ομοίωση του ανθρώπινου εγκεφάλου. Δηλαδή ένα πυκνά διασυνδεδεμένο σύνολο νευρωνικών κυττάρων – τους νευρώνες, που αποτελούν την στοιχειώδη επεξεργαστική μονάδα του εγκεφάλου και των νευρωνικών δικτύων. Ο ανθρώπινος εγκέφαλος αποτελείται από 10 δισεκατομμύρια νευρώνες και 60 τρισεκατομμύρια συνδέσεις μεταξύ τους, τις ονομαζόμενες συνάψεις (*Shepherd, 2004*). Κάθε νευρώνας λαμβάνει πληροφορία, με τη μορφή ηλεκτρικού δυναμικού στις συνάψεις του (δενδρίτες), την επεξεργάζεται και αντίστοιχα παράγει ή όχι ένα ηλεκτρικό παλμό που μεταδίδει ως πληροφορία στους επόμενους νευρώνες μέσω του άξονα του (Εικόνα 2). Η βασική διαφορά των βιολογικών και των αλγοριθμικών νευρωνικών δικτύων είναι ο "τοπολογικός" τους χαρακτήρας, με τον ανθρώπινο εγκέφαλο να έχει τοπικό καθώς συγκεκριμένα τμήματα του εγκεφάλου επεξεργάζονται κάθε φορά την πληροφορία. Ενώ τα αλγοριθμικά νευρωνικά δίκτυα έχουν καθολικό χαρακτήρα καθώς το σύνολο των νευρώνων επεξεργάζεται την πληροφορία.

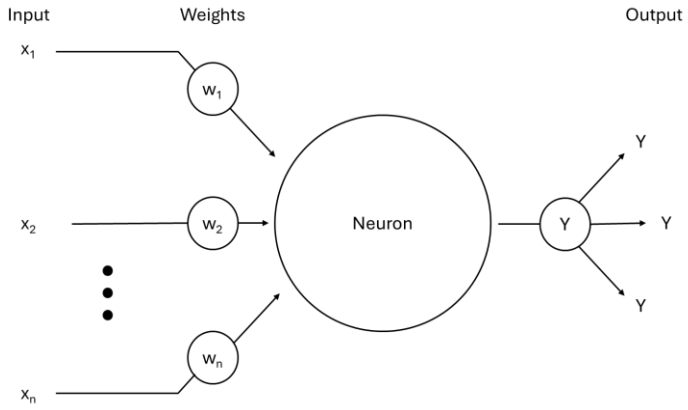


Εικόνα 2. Βιολογικό νευρωνικό δίκτυο (Neves et al., 2018)

Δομή

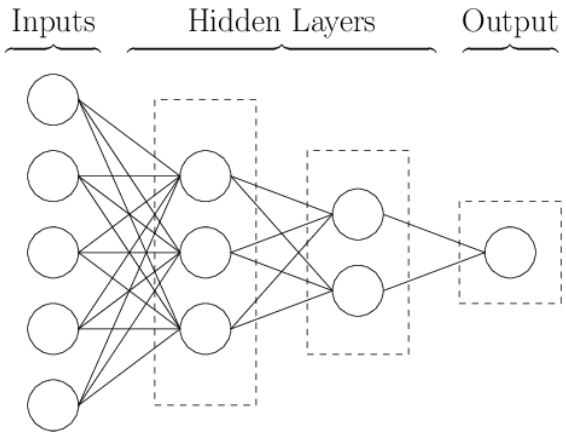
Η βασική δομική μονάδα των νευρωνικών δικτύων είναι οι νευρώνες. Όπως και ο βιολογικός νευρώνας, έτσι και ο τεχνητός αποτελεί τη μονάδα επεξεργασίας δεδομένων, που

λαμβάνει μέσω συνδέσεων με τους προηγούμενους απ' αυτόν νευρώνες. Όλα τα δεδομένα που λαμβάνει δεν έχουν την ίδια βαρύτητα, όπως και όλες οι συνδέσεις του δεν έχουν την ίδια σημασία. Οι συνδέσεις των νευρώνων πραγματοποιούνται με σταθμισμένους με βάρος συνδέσμους, που δίνουν ένα συντελεστή βαρύτητας στην πληροφορία που μεταφέρουν. Με το σύνολο των πληροφοριών που λαμβάνει ο νευρώνας, παράγει μια έξοδο που μεταφέρει και διαμοιράζει, μέσω του άξονα του, στους επόμενους νευρώνες (Εικόνα 3) (*Negnevitsky, 2001*).



Εικόνα 3. Αναπαράσταση νευρώνα

Οι νευρώνες συνθέτουν πολλαπλά στρώματα επεξεργασίας, μέσω συνδέσεων, απ' τα οποία διέρχεται η πληροφορία. Αυτά διαχωρίζονται σε 3 διαφορετικές κατηγορίες: στρώμα εισόδου, ενδιάμεσα/"κρυφά" στρώματα, στρώμα εξόδου. Το πρώτο στρώμα είναι αυτό της εισόδου, αυτό παρέχει στο νευρωνικό δίκτυο τα δεδομένα εισόδου (εικόνες, κείμενο, αριθμητικά δεδομένα κλπ). Τα κρυφά στρώματα αποκαλούνται έτσι καθώς δεν επικοινωνούν με το "εξωτερικό περιβάλλον", παρά μόνο πραγματοποιούν υπολογισμούς στα δεδομένα που δέχονται. Τα κρυφά στρώματα μπορεί να είναι πολλαπλά είτε ένα. Τέλος το στρώμα εξόδου παραλαμβάνει τα επεξεργασμένη δεδομένα, με τη μορφή εκτίμησης ή πρόβλεψης (Εικόνα 4).



Εικόνα 4. Στρώματα νευρωνικού δικτύου (Cinelli et al., 2018)

Λειτουργία

Η λειτουργία ενός νευρωνικού δικτύου είναι μια επαναλαμβανόμενη περιοδική διαδικασία, που αποτελείται από 4 βασικά στάδια: την **αρχικοποίηση**, την **ενεργοποίηση**, την **εκπαίδευση** και την **επανάληψη**.

Κατά την **αρχικοποίηση** γίνεται ανάθεση αυθαίρετων τιμών, στις συνδέσεις μεταξύ των νευρώνων, για τα βάρη w_1, w_2, w_n με εύρος τιμών [-0.5, 0.5]. Όπως επίσης και για το θ και το b, όπου το b είναι η μεροληψία που προστίθεται σε κάθε νευρώνα για να τον ενεργοποιήσει ακόμα και αν η είσοδός του είναι 0 και το θ θα αποτελέσει το κατώφλι ενεργοποίησης του νευρώνα.

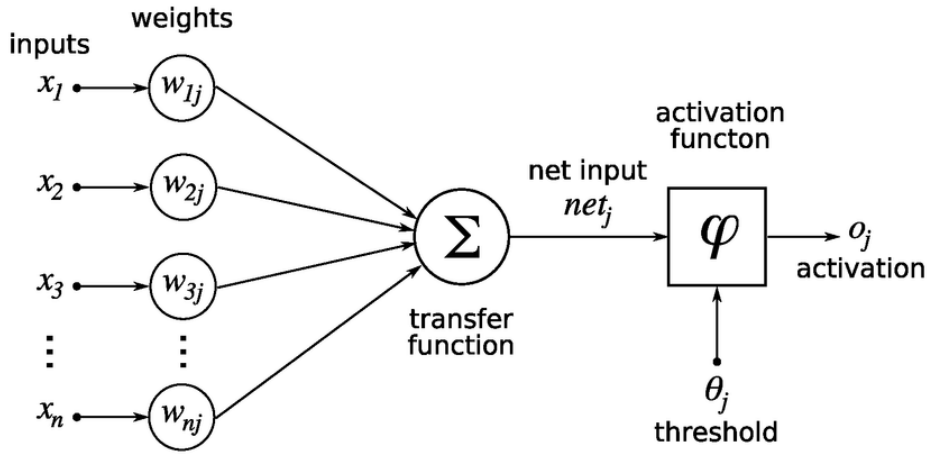
Στο στάδιο της **ενεργοποίησης** κάθε νευρώνας με βάση την επεξεργασία των εισόδων του "πυροδοτείται" ή όχι και παράγει έξοδο 0 για αδρανή κατάσταση και 1 για ενεργή. Ο νευρώνας αρχικά επεξεργάζεται τα δεδομένα που δέχεται από προηγούμενα στρώματα:

$$z = (w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_n \cdot x_n) + b$$

Και παράγει αποτέλεσμα συγκρίνοντας το σταθμισμένο άθροισμα εισόδου με το κατώφλι ενεργοποίησης του, με την εφαρμογή μιας συνάρτησης απότομου περιορισμού (βηματική/προσήμου) (*McCulloch & Pitts, 1943*):

$$Y = \begin{cases} 1 \text{ εάν } z \geq \vartheta \\ 0 \text{ εάν } z < \vartheta \end{cases}$$

Όπου Y είναι η έξοδος του νευρώνα μετά την επεξεργασία. Πλέον το κατώφλι δεν χρησιμοποιείται τόσο συχνά και έχει αντικατασταθεί από συναρτήσεις ενεργοποίησης (*sigmoid, ReLU, softmax*) (Εικόνα 5).



Εικόνα 5. Λειτουργία νευρώνα (Ren et al., 2018)

Ακολουθεί το στάδιο της **εκπαίδευσης** κατά το οποίο το νευρωνικό δίκτυο ενημερώνει τα βάρη των συνδέσμων του. Η διαδικασία αυτή είναι διμερής, αρχικά υπολογίζεται το σφάλμα (*loss*) και εν συνεχεία πραγματοποιείται οπισθοδιάδοση (*backpropagation*) για την ενημέρωση των βαρών των συνδέσμων.

Ο κύκλος του νευρωνικού δικτύου ολοκληρώνεται με επιστροφή στο πρώτο στάδιο για **επανάληψη** όλων των σταδίων από την αρχή. Η εκπαίδευση επαναλαμβάνεται για όλο το σύνολο των δεδομένων εισόδου πολλές φορές επιτρέποντας στο νευρωνικό δίκτυο να βελτιώνει τα βάρη των συσχετίσεων του οδηγώντας σε καλύτερες προβλέψεις. Ο πλήρης κύκλος εκπαίδευσης ονομάζεται εποχή, ένα νευρωνικό δίκτυο συνήθως λειτουργεί για αρκετές εποχές μέχρι να ολοκληρώσει την εκπαίδευση του και να ξεκινήσει τον έλεγχο.

Συναρτήσεις Ενεργοποίησης

Πλέον αντί για τη χρήση των συναρτήσεων απότομου περιορισμού/κατωφλίου (*hard-limit functions/threshold*), υπάρχει πληθώρα διαθέσιμων συναρτήσεων ενεργοποίησης (*activation functions*) για τα νευρωνικά δίκτυα. Οι συνηθέστερες αλλά και πιο σχετικές με το πρόβλημα του MNIST Dataset είναι οι: ***sigmoid***, ***rectified linear unit***, ***leaky rectified linear unit***, ***softmax***.

Η **σιγμοειδής** (*sigmoid*) συνάρτηση είναι μια από τις πιο γνωστές μη-γραμμικές συναρτήσεις που χρησιμοποιούνται. Παράγει έξοδο με τιμές (0,1) και χρησιμοποιείται σε προβλήματα δυαδικής ταξινόμησης (*Narayan, 1997*). Ο τύπος της είναι ο ακόλουθος:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Η ***rectified linear unit*** (*ReLU*) είναι ίσως η πιο συχνά χρησιμοποιούμενη στα νευρωνικά δίκτυα ιδιαιτέρως στα κρυφά επίπεδα. Ενεργοποιεί τον νευρώνα μόνο για θετικές εισόδους και επιστρέφει γραμμική έξοδο και 0 για αρνητικές τιμές. Ακολουθεί ο τύπος της:

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

Η ***leaky rectified linear unit*** (*leaky ReLU*) είναι μια παραλλαγή της *ReLU* που επιτρέπει και αρνητικές τιμές. Επιστρέφει έξοδο γραμμική για θετικές τιμές και πολύ μικρές αρνητικές τιμές για είσοδο μικρότερη του 0:

$$f(x) = max(0.01 \cdot x, x)$$

Τέλος η ***softmax*** χρησιμοποιείται συνήθως σε επίπεδα εξόδου προβλημάτων πολλαπλής ταξινόμησης, όπως το MNIST. Επιστρέφει κατανομή πιθανοτήτων για τις κλάσεις. Η πιθανότητα της κάθε κλάσης υπολογίζεται από τον ακόλουθο τύπο:

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Υπολογισμός Σφάλματος

Ένα απ' τα σημαντικότερα στοιχεία ενός νευρωνικού δικτύου, αφότου οριστεί η αρχιτεκτονική του, είναι η συνάρτηση απώλειας (*loss function*). Ο σωστός υπολογισμός της απόκλισης της πρόβλεψης από την πραγματική τιμή είναι καίριος για την επιτυχή εκπαίδευση του δικτύου. Η συνάρτηση απώλειας δείχνει πόσο επιτυχημένες είναι οι προβλέψεις του δικτύου, ενώ ο στόχος ολόκληρου του δικτύου είναι να ελαχιστοποιήσει τις τιμές αυτής της συνάρτησης. Υπάρχουν διάφορες συναρτήσεις απώλειας (*Mean Squared Error*, *Binary Cross-Entropy*), ωστόσο για προβλήματα πολλαπλής ταξινόμησης χρησιμοποιείται η ***Categorical Cross-Entropy*** (*Chollet, 2021*). Εφαρμόζεται σε σύνολα δεδομένων με ετικέτες της μορφής *one-hot encoding* δηλαδή από το σύνολο των κλάσεων (πιθανών προβλέψεων) μόνο μια είναι η σωστή (1) και οι υπόλοιπες λάθος (0). Οι εκτιμήσεις του μοντέλου για ένα σύνολο κλάσεων *k*, συγκροτούν ένα μοναδιαίο διάνυσμα μήκους *k*. Η μαθηματική της έκφραση έχει ως ακολούθως:

$$L(y, \hat{y}) = -\sum_{i=1}^{k} y_i \log(\hat{y}_i)$$

Όπου y_i είναι η πραγματική ετικέτα (0 ή 1) και $\hat{y}_i$ η εκτιμώμενη πιθανότητα απ' το μοντέλο για την κάθε κλάση *i*. Ενώ η συνάρτηση υπολογίζει το άθροισμα των σφαλμάτων για όλες της κλάσεις ταξινόμησης, για να αξιολογηθεί η "ευστοχία" του μοντέλου (*Zhang et al., 2023*). Η *Categorical Cross-Entropy* αποτελεί ειδική κατηγορία της *Cross-Entropy* συνάρτησης σφάλματος, που αφορά την πολλαπλή ταξινόμηση. Σε αντίθετη περίπτωση για δυαδική ταξινόμηση χρησιμοποιείται η *Binary Cross-Entropy*.

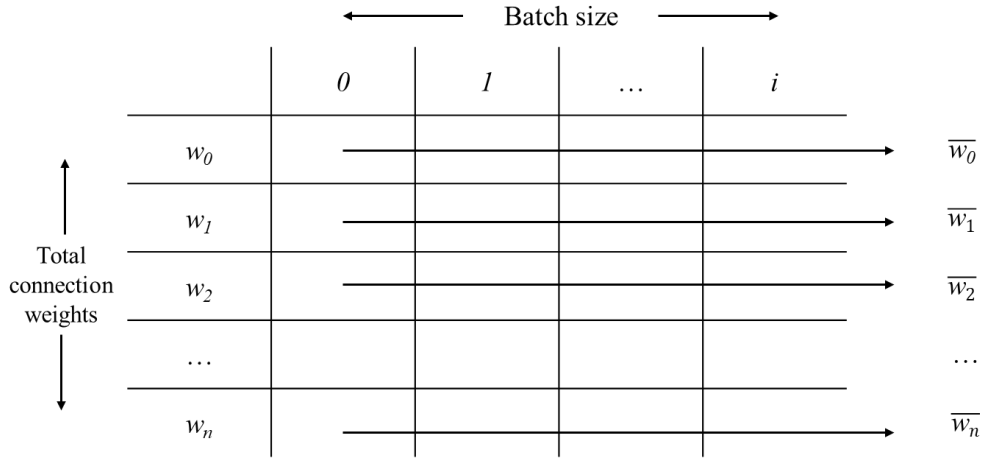
Οπισθόδρομη διάδοση

Η οπισθόδρομη διάδοση ή οπισθοδιάδοση (*backpropagation*) αποτελεί τον πυρήνα της μηχανικής μάθησης. Είναι η διαδικασία η οποία αξιοποιεί την υπολογισμένη απώλεια, για να επαναπροσδιορίσει τα βάρη των δεσμών των νευρώνων. Ο σκοπός είναι στο συνολικό άθροισμα απώλειας να βρεθούν οι δεσμοί οι οποίοι έχουν συμβάλλει περισσότερο, αποκλείοντας από την

επιθυμητή πρόβλεψη. Για να επιτευχθεί αυτό χρησιμοποιείται η βαθμίδα καθόδου (gradient descent), η οποία υπολογίζει τα ελάχιστα της συνάρτησης απώλειας και κατευθύνει τα βάρη προς τις τιμές που θα οδηγήσουν στη βέλτιστη απώλεια. Η μαθηματική έκφραση της βαθμίδας καθόδου είναι η ακόλουθη:

$$w_{new} = w_{old} - a \cdot \nabla J(w_{old})$$

Όπου w_{new} είναι το νέο βάρος ενός δεσμού, w_{old} το προηγούμενο βάρος του δεσμού, a ο ρυθμός μάθησης του δικτύου και $\nabla J(w_{old})$ η κλίση της συνάρτησης απώλειας ως προς τα βάρη (*Ruder, 2016*). Ουσιαστικά αυξάνει ή μειώνει το βάρος ενός δεσμού με βάση την κλίση της συνάρτησης κόστους κατευθύνοντας την τιμή της προς τα ελάχιστα. Ωστόσο σαν διαδικασία δεν θα ήταν φερέγγυα ή αμερόληπτη αν κάθε μοναδιαίο σύνολο δεδομένων επέβαλε αλλαγές στα βάρη του δικτύου (*stochastic gradient descent*) (Amari, 1993). Η λύση σε αυτό το πρόβλημα είναι η ενημέρωση των βαρών μετά από εξέταση ομάδων συνόλων δεδομένων (*batch gradient descent*), όπου υπολογίζονται οι εκτιμώμενες αλλαγές ανά μοναδιαίο σύνολο δεδομένων και εφαρμόζονται οι μέσοι όροι της ομάδας για τις εκτιμώμενες αλλαγές σε κάθε βάρος (Εικόνα 6).



Εικόνα 6. Batch gradient descent

Επίσης σημαντικός για την αποφυγή είναι και ο επιλεγόμενος ρυθμός μάθησης του μοντέλου καθώς μπορεί να οδηγήσει σε μη αποδοτική επεξεργασία ή αντίθετα αποστήθιση. Είναι επιθυμητό να λαμβάνει μια μέση τιμή ή κυμαινόμενη.

Εναλλακτικά της βαθμίδας καθόδου υπάρχουν πολλοί βελτιστοποιητές που χρησιμοποιούνται σήμερα κατά την οπισθοδιάδοση, ο πιο συχνός είναι ο *Adam* (*Adaptive Moment Estimation*). Αποτελεί μια εξελιγμένη μέθοδο της βαθμίδας καθόδου, που χρησιμοποιεί τους

βελτιωτές *Momentum* και *RMSProp*. Πρόκειται για πιο σταθερή μέθοδο με ταχύτερη σύγκλιση προς το ελάχιστο της συνάρτησης απώλειας. Ακολουθεί ο μαθηματικός της τύπος:

$$w_{t+1} = w_t - a \cdot \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} - \epsilon}$$

Όπου w_{t+1} και w_t είναι τα βάρη ενημερωμένο και παλιό αντίστοιχα, a είναι ο ρυθμός εκμάθησης, $\widehat{m_t}$ η πρώτη ροπή (υπολογίζεται από το *Momentum*), $\widehat{v_t}$ η δεύτερη ροπή (υπολογίζεται από το *RMSProp*) και ϵ μια απειροελάχιστη θετική ποσότητα (10^{-8}) (*Kingma & Ba, 2014; Ruder, 2016*).

Η διαδικασία αυτές πραγματοποιούνται πολλές φορές μέσα στο πέρας της επεξεργασίας των εκπαιδευτικών δεδομένων (στο τέλος κάθε ομάδας δεδομένων), μέχρι την επεξεργασία όλου του συνόλου των δεδομένων. Ωστόσο η επεξεργασία του συνόλου των εκπαιδευτικών δεδομένων μπορεί να επαναλαμβάνεται και αυτή. Σε αυτή την περίπτωση κάθε πλήρης κύκλος επεξεργασίας όλων των δεδομένων ονομάζεται εποχή. Ένα νευρωνικό δίκτυο μπορεί να εκπαιδεύεται για πολλαπλές εποχές πριν την ολοκλήρωση της εκπαίδευσής του και τον έλεγχο του.

Ανάπτυξη - Σχεδίαση

Προετοιμασία Δεδομένων

Το MNIST Dataset, όπως προαναφέρθηκε έχει συλλεχθεί από τον *Yann LeCun* το 1998, απαρτίζεται από 4 αρχεία IDX μορφής. Πρόκειται για δυαδικά αρχεία που αποθηκεύουν διανύσματα και πολυδιάστατες μήτρες πολλαπλών αριθμητικών τύπων. Το σύνολο των δεδομένων συνοδεύεται από τον απαιτούμενο οδηγό διαδικασιών και τεκμηρίωσης, σύμφωνα με τον οποίο τα δεδομένα διαχωρίζονται σε 4 αρχεία. Δυο αρχεία που περιέχουν τις εικόνες των χειρόγραφων ψηφίων (εκπαίδευσης και ελέγχου) και δυο αρχεία που περιέχουν τις ετικέτες των εικόνων που συνοδεύουν (εκπαίδευσης και ελέγχου αντίστοιχα). Η δομή και η οργάνωση των αρχείων περιγράφεται αναλυτικά στον οδηγό του MNIST, έτσι ώστε να μπορεί να πραγματοποιηθεί προετοιμασία των αρχείων για επεξεργασία, ακολουθεί ο κώδικας επεξεργασίας και μετατροπής των αρχείων σε csv (*comma separated values*):

```
import os

def convert(imgf, labelf, outf, n):
    with open(imgf, "rb") as file , open(outf, "w") as out, open(labelf, "rb") as label:

        file.read(16)          #Skip first 16 characters for images (additional info)
        label.read(8)           #Skip first 8 characters for labels (additional info)
        images = []

        for i in range(n):
            image = [ord(label.read(1))]
            for j in range(28*28):
                image.append(ord(file.read(1)))
            images.append(image)
        for image in images:
            out.write(",".join(str(pix) for pix in image) + "\n")

destTrain = 'Images\\train.csv'
destTest = 'Images\\test.csv'


if not os.path.exists(destTrain):
    convert('Dataset\\train-images-idx3-ubyte', 'Dataset\\train-labels-idx1-ubyte', destTrain,
60000)
else:
    print("Train dataset already converted")

if not os.path.exists(destTest):
```

```
    convert('Dataset\\t10k-images-idx3-ubyte', 'Dataset\\t10k-labels-idx1-ubyte', destTest,
10000)
else:
    print("Test dataset already converted")
```

Η διαδικασία έχει τα εξής βήματα:

- Προσπερνώνται τα πρώτα 8 byte για αρχείο ετικέτας και τα πρώτα 16 byte για αρχείο εικόνας, καθώς περιέχουν πληροφορίες για το αρχείο (*magic number MSB first,* σύνολο εικόνων/ετικετών, σύνολο στηλών και γραμμών ανά εικόνα).
- Συνδυάζονται οι ετικέτες με τις εικόνες στις οποίες απευθύνονται σε αρχεία csv που περιέχουν ουσιαστικά ένα δισδιάστατο πίνακα αλφαριθμητικών συμβολοσειρών. Το μέγεθος των αρχείων είναι 60.000 ή 10.000 γραμμές, για την εκπαίδευση και για τον έλεγχο αντίστοιχα, μεγέθους 785 χαρακτήρων (ετικέτα πρώτος χαρακτήρας κάθε γραμμής + 28x28 pixel values) συνυπολογίζοντας τα κόμματα ως διαχωριστικά.

Ανάγνωση

Για να τροφοδοτήσουν τα δεδομένα το νευρωνικό δίκτυο είναι πιο πρακτικό να μορφοποιηθούν ως πίνακες, κάτι που διευκολύνεται με την αποθήκευσή τους σε csv μορφή. Η μορφοποίηση επιτυγχάνεται με τη βοήθεια της βιβλιοθήκης numpy, ενώ η ανάγνωση csv αρχείων με την ομώνυμη βιβλιοθήκη.

```
import csv
import numpy as np
from keras.utils import to_categorical


def loadData(destination):
    with open(destination, "r") as csv_file:
        reader = csv.reader(csv_file)
        data = np.array(list(reader)).astype(float)

    X = data[:, 1:]     #Pixel values
    Y = data[:, 0]      #Labels
    X = X / 255.0       #From grayscale values of 0-255 to absolute values 0-1
    Y = to_categorical(Y, num_classes=10)
    return X, Y

dest = 'Images\\train.csv'
dest2 = 'Images\\test.csv'

X_train, Y_train = loadData(dest)
X_test, Y_test = loadData(dest2)
```

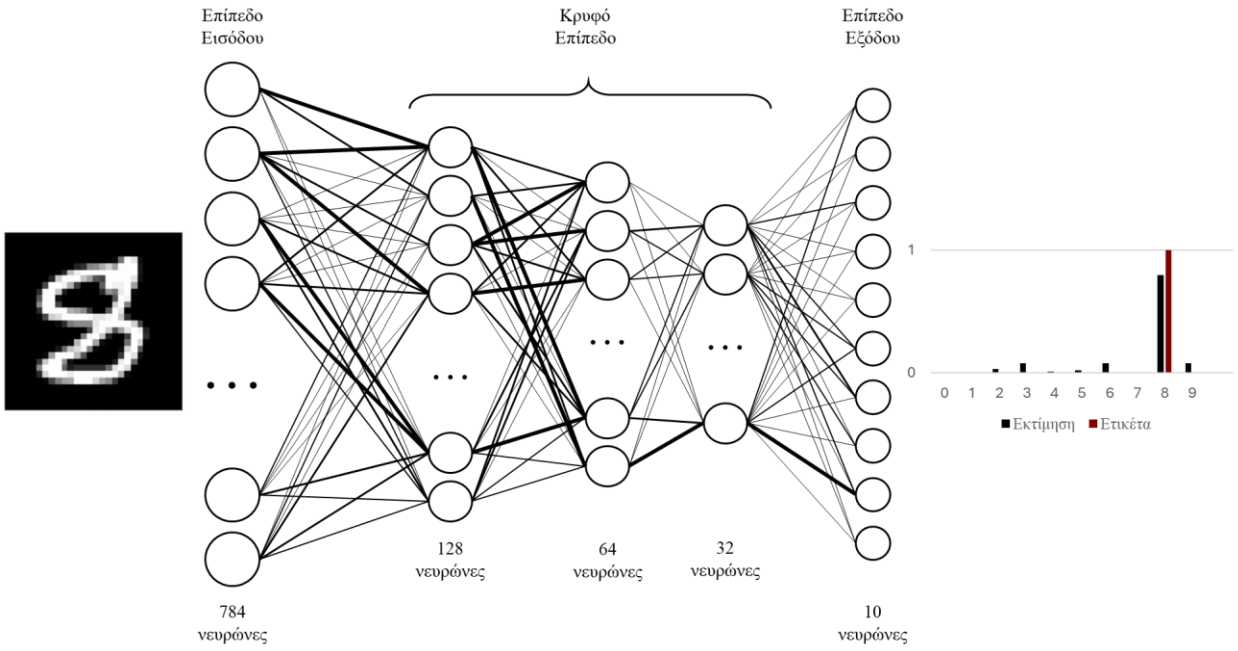
Η διαδικασία έχει τα εξής βήματα:

- Αρχικά ανοίγονται τα αρχεία και φορτώνονται στη μεταβλητή data ως πίνακας float στοιχείων.
- Τα δεδομένα διαχωρίζονται σε X,Y όπου X είναι οι τιμές σε κλίμακα ασπρόμαυρου (0-255) ανά pixel και Y οι ετικέτες των εικόνων και απαρτίζονται από τις δέκα κλάσεις που συνθέτουν τα αριθμητικά ψηφία (0-9)
- Ακολούθως τα δεδομένα κανονικοποιούνται για τις εικόνες οι αποχρώσεις του ασπρόμαυρου διαιρούνται με 255, ώστε να εκφράζονται ποσοστιαίες μονάδες. Ενώ οι ετικέτες μετατρέπονται σε *one-hot encoding*, δηλαδή μοναδιαίο διάνυσμα μήκους 10 στοιχείων όπου όλα τα στοιχεία είναι 0 εκτός από το στοιχείο που αντιπροσωπεύει την κλάση, που είναι 1 (*πχ* η ετικέτα 2 γίνεται [0,0,1,0,0,0,0,0,0,0] και ούτω καθεξής). Η μετατροπή σε *one-hot encoding* γίνεται με τη βοήθεια της συνάρτησης `to_categorical` από τη βιβλιοθήκη *tensorflow*.
- Τέλος τα στοιχεία αποθηκεύονται στις μεταβλητές `X_train` (εικόνα εκπαίδευσης), `Y_train` (ετικέτα εκπαίδευσης), `X_test` (εικόνα ελέγχου), `Y_test` (ετικέτα ελέγχου).

Σημειωτέον ότι επειδή το πρόβλημα του MNIST είναι ιδιαίτερα γνωστό, υπάρχουν στο διαδίκτυο ήδη επεξεργασμένα σε μορφή csv τα δεδομένα, ακόμα και μέσω της βιβλιοθήκης *tensorflow*. Ωστόσο για την πιο πιστή διεκπεραίωση του πειράματος προτιμήθηκε η αρχική μορφή τους, όπως δόθηκε στην επιστημονική κοινότητα από τους *LeCun*, *Cortes* και *Burges* (1998).

Αρχιτεκτονική του Νευρωνικού Δικτύου

Η αρχιτεκτονική που επιλέχθηκε είναι αυτή ενός απλού νευρωνικού δικτύου εμπρόσθιας τροφοδότησης (*Feed-forward Neural Networks, FNN*). Το επίπεδο εισόδου υποχρεωτικά αποτελείται από 784 χαρακτηριστικά εισόδου (νευρώνες) που συντελούνται από το σύνολο των pixel της κάθε εικόνας (28x28). Το κρυφό επίπεδο μπορεί να αποτελείται από ένα επίπεδο, χωρίς να στερείται σημαντικά υπολογιστικής δύναμης (*Cybenko, 1989*). Ωστόσο πολλές φορές προτιμώνται πολλαπλά επίπεδα επεξεργασίας. Αυτό δεν συνεπάγεται ότι η ποσότητα οδηγεί στην ποιότητα πάντα. Έρευνες έχουν δείξει ότι η υπέρβαση ενός ωφέλιμου ορίου κρυφών επιπέδων μπορεί να δημιουργήσει προβλήματα όπως χρονική πολυπλοκότητα και αποστήθιση (*Uzair & Jamil, 2021*). Ερευνητικές εκτιμήσεις συγκλίνουν προς τα 3 κρυφά επίπεδα και αυτή είναι και η γραμμή που ακολούθησε και η παρούσα εργασία (*Shen, Yang & Zhang, 2021; Uzair & Jamil, 2021*). Τέλος το επίπεδο εξόδου καθορίζεται από τον τύπο του προβλήματος ξανά. Το ζητούμενο είναι κατηγοριοποίηση σε κλάσεις ψηφίων από το 0 έως το 9, συνεπώς και το επίπεδο εξόδου αποτελείται από 10 νευρώνες (Εικόνα 7).



Εικόνα 7. Αρχιτεκτονική νευρωνικού δικτύου του πειράματος

Ακόμη συνηθίζεται ως αρχιτεκτονική επιλογή του δικτύου σχετικά με τα κρυφά επίπεδα, ο πληθυσμός των νευρώνων ανά επίπεδο να φθίνει σε δυνάμεις του 2, ενώ κυμαίνεται μεταξύ του πλήθους των νευρώνων των επιπέδων εισόδου και εξόδου. Όπως γίνεται αντιληπτό και από το διάγραμμα της αρχιτεκτονικής του δικτύου μας. Για την αρχιτεκτονική του κρυφού επιπέδου βάσει των προαναφερθέντων, επιλέχθηκε δομή 3 επιπέδων. Το πρώτο κρυφό επίπεδο αποτελείται από 128 νευρώνες, το δεύτερο κρυφό επίπεδο από 64 νευρώνες και το τρίτο από 32.

Εκτέλεση

Νευρωνικό δίκτυο

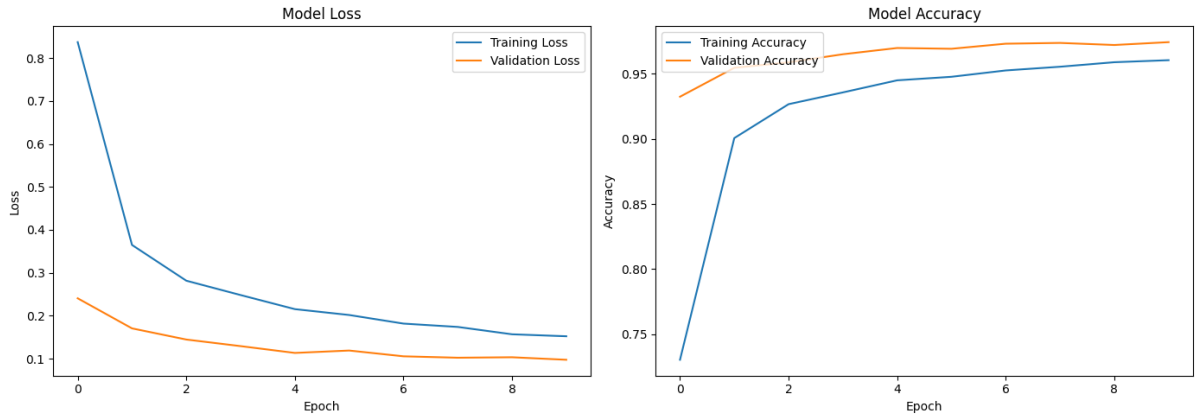
Το νευρωνικό δίκτυο (*network.py*) που κατασκευάστηκε επεξεργάζεται μέσω του ακόλουθου κώδικα τα δεδομένα εικόνων του MNIST Dataset (*train.csv & test.csv*):

```
import tensorflow as tf
from tensorflow import keras
from keras import Sequential
from keras.layers import Dense, Activation, Dropout
from matplotlib import pyplot as plt

model = Sequential([
    Dense(128, activation='relu', input_shape=(784,)), Dropout(0.3),
    Dense(64, activation='relu'), Dropout(0.3),
    Dense(32, activation='relu'),  Dropout(0.3),
    Dense(10, activation='softmax')
])
lr_schedule = keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=1e-2,
    decay_steps=7500,
    decay_rate=0.9)
optimizer = tf.keras.optimizers.Adam(learning_rate=lr_schedule)
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
history = model.fit(X_train, Y_train, validation_data=(X_test, Y_test), batch_size= 128,
epochs= 10)

test_loss, test_acc = model.evaluate(X_test, Y_test)
print('Test accuracy:', test_acc)


fig, axs = plt.subplots(1, 2, figsize=(14, 5))
axs[0].plot(history.history['loss'], label='Training Loss')
axs[0].plot(history.history['val_loss'], label='Validation Loss')
axs[0].set_title('Model Loss')
axs[0].set_ylabel('Loss')
axs[0].set_xlabel('Epoch')
axs[0].legend(loc='upper right')
axs[1].plot(history.history['accuracy'], label='Training Accuracy')
axs[1].plot(history.history['val_accuracy'], label='Validation Accuracy')
axs[1].set_title('Model Accuracy')
axs[1].set_ylabel('Accuracy')
axs[1].set_xlabel('Epoch')
axs[1].legend(loc='upper left')
plt.tight_layout()
plt.show()
```

Όπου τα στοιχεία του δικτύου, πέραν των προαναφερθέντων αρχιτεκτονικών επιλογών, είναι τα ακόλουθα:

- *Activation Function,* τα τρία κρυφά επίπεδα που χρησιμοποιήθηκαν ενεργοποιούνταν με ReLU. Ενώ στο επίπεδο εξόδου εφαρμοζόταν softmax επιστρέφοντας κατανομή πιθανοτήτων για την κάθε κλάση, ώστε να συγκριθεί με την one-hot κωδικοποίηση που χρησιμοποιήθηκε για τις ετικέτες των εικόνων.
- *Dropout,* στα κρυφά επίπεδα εφαρμόστηκε η τεχνική κανονικοποίησης dropout που αντιμετωπίζει το overfitting (αποστήθιση). Πρόκειται για το πρόβλημα που προκύπτει όταν το δίκτυο εκπαιδεύεται αρκετά καλά στο σύνολο εκπαίδευσης, αξιοποιώντας ως πληροφορία και τον θόρυβο, καταλήγοντας να "αποστηθίζει" τα δεδομένα που λαμβάνει. Με αποτέλεσμα όταν δέχεται τα δεδομένα ελέγχου να μην μπορεί να αποδώσει (*Srivastava et al., 2014*). Το dropout απενεργοποιεί νευρώνες σε επιλεγόμενο ποσοστό ώστε να αποφεύγεται η στήριξη σε μεμονωμένες σχέσεις μεταξύ νευρώνων, εισάγοντας τυχαιότητα. Οι νευρώνες των κρυφών επιπέδων του δικτύου είχαν dropout 30% (0.3).
- *Optimizer,* έγινε χρήση του Adam βελτιωτή και ως συνάρτηση σφάλματος αξιοποιήθηκε η categorical Cross-Entropy. Ο ρυθμός μάθησης ήταν μεταβλητός (lr_schedule). Ένας υψηλός ρυθμός μάθησης στα πρώτα στάδια της εκπαίδευσης επιτρέπει στο δίκτυο τις κατάλληλες προσαρμογές. Όμως στα προχωρημένα στάδια της εκπαίδευσης ο ίδιος ρυθμός μάθησης μπορεί να οδηγήσει σε υπερπροσαρμογές (overcorrections), με αποτέλεσμα την απομάκρυνση από τη βέλτιστη παραμετροποίηση. Ο ρυθμός μάθησης (learning rate) αρχικά ήταν 0.01 (1e-2) και μειωνόταν εκθετικά με βήμα 7500 εικόνες και μείωση 10% (rate = 0.9) ανά ρύθμιση.
- *Batch size,* το δίκτυο οργανωνόταν σε δέσμες των 128 συνόλων δεδομένων για την πραγματοποίηση της οπισθοδιάδοσης.
- *Epochs*, 10 πλήρεις κύκλους εκπαίδευσης του δικτύου στο εκπαιδευτικό σύνολο (60.000 εικόνων).

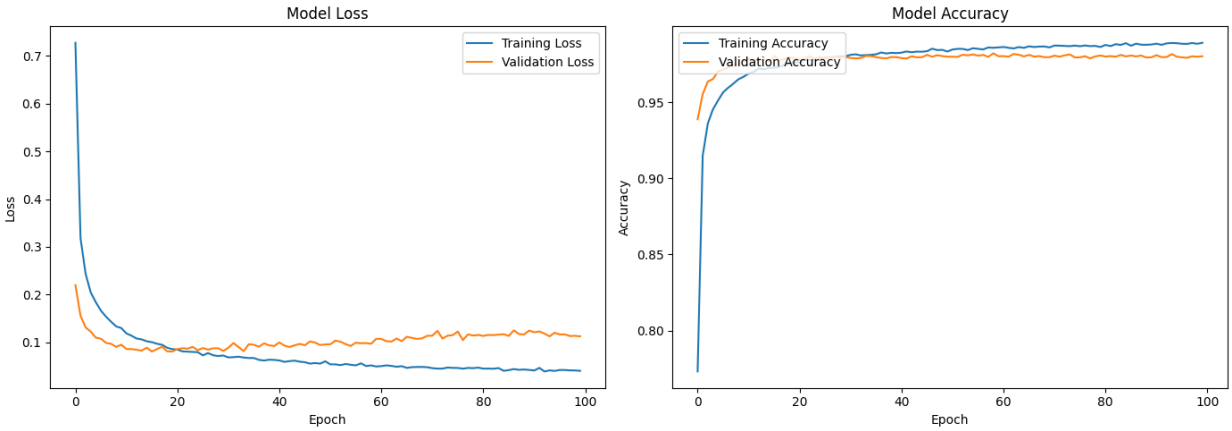
Η κατασκευή και υλοποίηση του δικτύου γίνεται με την βοήθεια της βιβλιοθήκης *tensorflow*, όπως γίνεται αντιληπτό και από το δείγμα κώδικα. Τα αποτελέσματα εκφράζονται σε διαγράμματα με τη βοήθεια της βιβλιοθήκης *matplotlib*. Ενώ το παραγόμενο διάγραμμα είναι το ακόλουθο:



Εικόνα 8. Απώλεια (αριστερά) και ευστοχία εκτιμήσεων (δεξιά) σε σύνολα εκπαίδευσης (μπλε) και ελέγχου (πορτοκαλί) σε νευρωνικό δίκτυο 10 εποχών

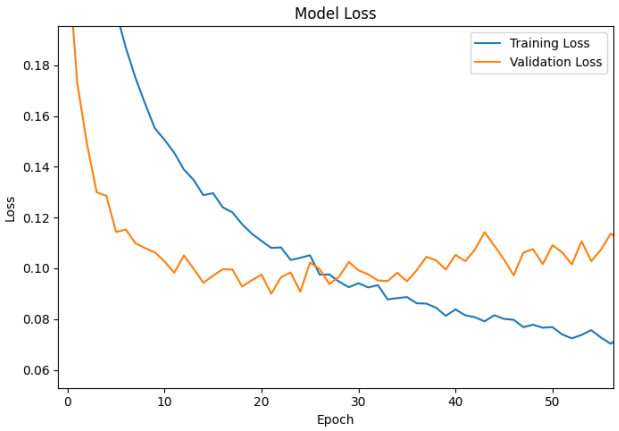
Το δίκτυο από την πρώτη κιόλας εποχή επιτυγχάνει ποσοστό ευστοχίας 93% στο σύνολο ελέγχου, παρόλο που η ευστοχία του στο σύνολο εκπαίδευσης ήταν ιδιαίτερα χαμηλή, μόλις 55%. Σε ότι αφορά την απώλεια, η πρώτη εποχή είχε αντίστοιχη εικόνα ξεκινώντας από 1.3 για το σύνολο εκπαίδευσης και πέφτοντας ραγδαία στο σύνολο εξέτασης με 0.2. Η εικόνα του δικτύου είναι η αναμενόμενη, ραγδαία αύξηση της "γνώσης" στις πρώτες εποχές και σταδιακή βελτίωση μεν, αλλά σημαντικά βραδύτερη δε στις τελευταίες εποχές. Όλα τα παραπάνω οδηγούν το δίκτυο στην επίτευξη ποσοστού ευστοχίας, μετά το πέρας της εκπαίδευσης του, 97.4% στο σύνολο ελέγχου. Γεγονός που υποδεικνύει σημαντική βελτίωση σε μόλις 10 εποχές με 4% αύξηση στην επίδοση του δικτύου.

Περαιτέρω με αλλαγή της λειτουργίας του νευρωνικού δικτύου από 10 εποχές σε 100 λαμβάνουμε το ακόλουθο διάγραμμα:



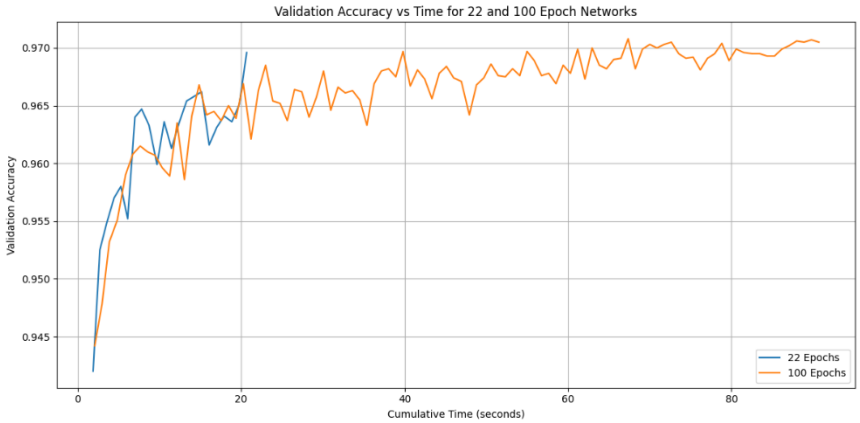
Εικόνα 8. Απώλεια (αριστερά) και ευστοχία εκτιμήσεων (δεξιά) σε σύνολα εκπαίδευσης (μπλε) και ελέγχου (πορτοκαλί) σε νευρωνικό δίκτυο 100 εποχών

Αναλύοντας τα δεδομένα του γραφήματος, λαμβάνουμε πανομοιότυπες καταστάσεις εκκίνησης του δικτύου στην 1η εποχή 57% ευστοχία και απώλεια 1.3 για το σύνολο εκπαίδευσης, ενώ για τον έλεγχο 93% ευστοχία και 0.2 απώλεια. Μετά το πέρας των 100 εποχών και του συνόλου εκπαίδευσης το δίκτυο εξετάζεται και αποκρίνεται με βαθμό ευστοχίας 97.9%. Δηλαδή παρά το γεγονός ότι η διαδικασία εκπαίδευσης δεκαπλασιάστηκε σε εποχές επιτεύχθηκε αύξηση 0.4%. Ακόμη σημαντικά δείγματα δίνει το διάγραμμα απώλειας το οποίο παρά την πρώτη εκθετική μείωση (*validation set, 0-5 epochs*) παρουσιάζει χαμηλή αλλά αύξουσα πορεία μετά της 20 πρώτες εποχές. Ειδικότερα η χαμηλότερη τιμή του υπολογισμού σφάλματος εντοπίζεται στην 22η εποχή με 0.09, ενώ το δίκτυο τερματίζει στην 100η εποχή με 0.12 απώλεια συνόλου ελέγχου (Εικόνα 9).



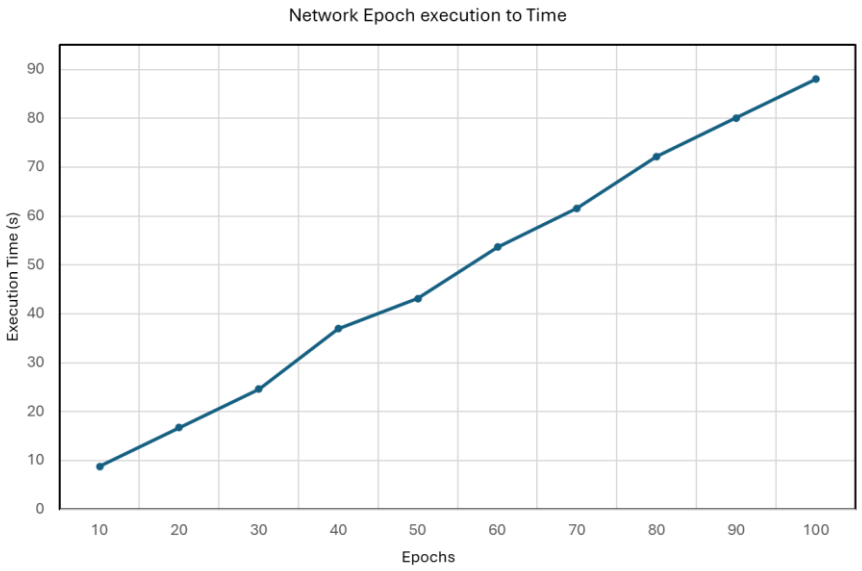
Εικόνα 9. Ελάχιστο απώλειας (0.09) 22ης εποχής δικτύου 100 εποχών

Για να εκτιμήσουμε καλύτερα την αποδοτικότητα της αύξησης εποχών μπορούμε να αντιστοιχήσουμε την ακρίβεια του μοντέλου στον χρόνο λειτουργίας του. Έτσι μπορούμε να λάβουμε μια εικόνα σχετικά με την απόδοση της διαχείριση των πόρων που πραγματοποιεί το σύστημά μας. Λαμβάνοντας υπόψη ότι η ελάχιστη απώλεια εντοπίστηκε στις 22 εποχές μπορούμε να την συγκρίνουμε με το δίκτυο πλήρους εκπαίδευσης (100 εποχών). Σε διάγραμμα που εκφράζει την ακρίβεια ελέγχου για δίκτυο 22 και 100 εποχών ως προς το χρόνο λαμβάνουμε την ακόλουθη εικόνα (Εικόνα 10):



Εικόνα 10. Ακρίβεια συστήματος ως προς τον χρόνο για δίκτυα 22 και 100 εποχών

Ο χρόνος λειτουργίας του συστήματος εν αντιστοιχία με τις εκτελούμενες εποχές παρουσιάζει γραμμική αύξηση (Εικόνα 11). Η ολοκλήρωση της εκπαίδευσης και του ελέγχου του δικτύου πραγματοποιείται σε 88 δευτερόλεπτα (για 100 εποχές), δίνοντας μας μια αναλογία 0.88 δευτερόλεπτα ανά εποχή. Προφανώς η αναλογία χρόνου προς εποχή διαφέρει από περιβάλλον σε περιβάλλον, δεδομένων των υπολογιστικών δυνατοτήτων. Ωστόσο τα εν λόγω στοιχεία έχουν προκύψει από το ίδιο περιβάλλον, ούτως ώστε να είναι εφικτή η σύγκρισή τους.



Εικόνα 11. Αναλογία εποχών ως προς χρόνο εκτέλεσης

Καθώς η διάρκεια της λειτουργίας αυξάνεται γραμμικά με τις εκπληρωμένες εποχές, αντιλαμβανόμαστε ότι δεν ακολουθεί την ακρίβεια ελέγχου. Και τα δυο μεγέθη έχουν αύξουσα πορεία, σε διαφορετικό ρυθμό ωστόσο. Κάτι που έχει ως αποτέλεσμα για δίκτυο 22 εποχών ένα μέσο ρυθμό βελτίωσης 0.1% ανά εποχή, έναντι του 0.03% για το δίκτυο των 100 εποχών.

Συμπεράσματα

Ένα τυπικό νευρωνικό δίκτυο εμπρόσθιας διάδοσης είναι ένα ικανοποιητικό μέσο αναγνώρισης προτύπων. Η αρχιτεκτονική του μπορεί να διαφέρει αναλόγως του εξεταζόμενου προβλήματος. Για την περίπτωση του συνόλου δεδομένων MNIST μπορεί να εκτιμήσει ικανοποιητικά μέχρι 97.9% το είδος του ψηφίου που τροφοδοτείται. Σε ότι αφορά την απόδοση του από άποψη πόρων, εκτιμάται ότι ένας υψηλός αριθμός επαναλήψεων της εκπαίδευσης – εποχών – δεν προσδίδει υποχρεωτικά αξιοπιστία. Τα δεδομένα οδηγούν στο συμπέρασμα ότι υπάρχει ένας βέλτιστος αριθμός εποχών για την απόδοση του δικτύου. Μια αρχιτεκτονική δικτύου που ξεπερνά το κατώφλι αυτού του βέλτιστου αριθμού εποχών, κινδυνεύει από αποστήθιση του συνόλου εκπαίδευσης και πτώση της αποδοτικότητας του στο σύνολο ελέγχου. Συμπεράσματα τα οποία βρίσκουν σύμφωνες και παλαιότερες έρευνες (*Wu & Liu, 2009; Afaq & Rao, 2020*) οι οποίες εισαγάγουν την τεχνική της πρώιμης διακοπής (*early stopping*). Αξιοποιώντας ως κριτήριο την πορεία της συνάρτησης απώλειας μπορούμε να εκτιμήσουμε την αποδοτικότητα του δικτύου και να διακόψουμε την εκπαίδευση με την εμφάνιση αύξησης στην απώλεια κατά την πτωτική πορεία της.

Βάσει των προαναφερθέντων θεωρείται ασφαλής εκτίμηση η επιλογή της διακοπής της εκπαίδευσης του υπό εξέταση δικτύου, καθώς αποτελούσε κατώτατο για συνάρτηση απώλειάς του. Επιτυγχάνοντας έτσι την αύξηση της αποδοτικότητας του και την αποφυγή του κινδύνου αποστήθισης των δεδομένων που του παρήχθησαν.

Βιβλιογραφία

Afaq, S., & Rao, S. (2020). Significance of epochs on training a neural network. *International Journal of Scientific & Technology Research*, *9*, 485–488. Retrieved from https://api.semanticscholar.org/CorpusID:225647672

Amari, S. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, *5*(4–5), 185–196. Retrieved September 30, 2024, from https://linkinghub.elsevier.com/retrieve/pii/092523129390006O

Briganti, G. (2024). How ChatGPT works: A mini review. *European Archives of Oto-Rhino-Laryngology*, *281*(3), 1565–1569. Retrieved September 24, 2024, from https://link.springer.com/10.1007/s00405-023-08337-7

Byerly, A., Kalganova, T., & Dear, I. (2021). No routing needed between capsules. *Neurocomputing*, *463*, 545–553. Retrieved September 25, 2024, from https://linkinghub.elsevier.com/retrieve/pii/S0925231221012546

Campbell, M., Hoane, A. J., & Hsu, F. (2002). Deep blue. *Artificial Intelligence*, *134*(1–2), 57–83. Retrieved September 25, 2024, from https://linkinghub.elsevier.com/retrieve/pii/S0004370201001291

Cinelli, L., Chaves, G., & Lima, M. (2018). Vessel classification through convolutional neural networks using passive sonar spectrogram images. *Anais de XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*. Presented at the XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, Sociedade Brasileira de Telecomunicações. Retrieved September 26, 2024, from http://biblioteca.sbrt.org.br/articles/1800

Chollet, F. (2021). *Deep learning with Python* (2nd ed.). Manning Publications Co. Retrieved September 27, 2024, from https://www.manning.com/books/deep-learning-with-python-second-edition

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303–314. Retrieved October 7, 2024, from http://link.springer.com/10.1007/BF02551274

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv. Retrieved September 30, 2024, from https://arxiv.org/abs/1412.6980

Kumari, D., & Bhat, S. (2021). Application of artificial intelligence in tesla- a case study. Retrieved September 24, 2024, from https://zenodo.org/record/5775457

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. Retrieved September 26, 2024, from http://ieeexplore.ieee.org/document/726791/

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. Retrieved September 25, 2024, from http://link.springer.com/10.1007/BF02478259

Narayan, S. (1997). The generalized sigmoid activation function: Competitive supervised learning. *Information Sciences*, *99*(1–2), 69–82. Retrieved September 26, 2024, from https://linkinghub.elsevier.com/retrieve/pii/S0020025596002009

Negnevitsky, M. (2001). *Artificial intelligence: A guide to intelligent systems* (1st Edition.). USA: Addison-Wesley Longman Publishing Co.

Nejjar, M., Zacharias, L., Stiehle, F., & Weber, I. (2024). LLMs for science: Usage for code generation and data analysis. *Journal of Software: Evolution and Process*, e2723. Retrieved September 24, 2024, from https://onlinelibrary.wiley.com/doi/10.1002/smr.2723

Neves, A. C., González, I., Leander, J., & Karoumi, R. (2018). A new approach to damage detection in bridges using machine learning. In J. P. Conte, R. Astroza, G. Benzoni, G. Feltrin, K. J. Loh, & B. Moaveni (Eds.), *Experimental Vibration Analysis for Civil Structures* (Vol. 5, pp. 73–84). Cham: Springer International Publishing. Retrieved September 26, 2024, from http://link.springer.com/10.1007/978-3-319-67443-8_5

Ren, S., Chen, G., Li, T., Chen, Q., & Li, S. (2018). A deep learning-based computational algorithm for identifying damage load condition: An artificial intelligence inverse problem solution for failure analysis. *Computer Modeling in Engineering & Sciences*, *117*(3), 287–307. Retrieved September 26, 2024, from http://www.techscience.com/CMES/v117n3/33820

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv. Retrieved September 30, 2024, from https://arxiv.org/abs/1609.04747

Shen, Z., Yang, H., & Zhang, S. (2021). Neural network approximation: Three hidden layers are enough. *Neural Networks*, *141*, 160–173. Retrieved October 9, 2024, from https://linkinghub.elsevier.com/retrieve/pii/S0893608021001465

Shepherd, G. M. (2004). Introduction to synaptic circuits. In G. M. Shepherd (Ed.), *The Synaptic Organization of the Brain* (pp. 1–38). Oxford University Press. Retrieved September 26, 2024, from https://academic.oup.com/book/25657/chapter/193090888

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., et al. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*(7676), 354–359. Retrieved September 24, 2024, from https://www.nature.com/articles/nature24270

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. Retrieved October 12, 2024, from http://jmlr.org/papers/v15/srivastava14a.html

Uzair, M., & Jamil, N. (2020). Effects of hidden layers on the efficiency of neural networks. *2020 IEEE 23rd International Multitopic Conference (INMIC)* (pp. 1–6). Presented at the 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan: IEEE. Retrieved October 9, 2024, from https://ieeexplore.ieee.org/document/9318195/

Wu, X., & Liu, J. (2009). A new early stopping algorithm for improving neural network generalization. *2009 Second International Conference on Intelligent Computation Technology and Automation* (Vol. 1, pp. 15–18). Presented at the 2009 Second International Conference on Intelligent Computation Technology and Automation. Retrieved October 20, 2024, from https://ieeexplore.ieee.org/document/5287721

Zhang, A., Lipton C., Z., Li, M., & Smola J., A. (2023). *Dive into Deep Learning*. Cambridge University Press. Retrieved from https://d2l.ai/index.html