

4η εργασία στην Τεχνητή Νοημοσύνη

Άγγελος Εξαφτόπουλος

AEM: 4635 email: aexaf@csd.auth.gr

Hyper-parameter Tuning

Όπως αναφέρεται και στο [Google Colab Notebook](#), θα μεταβληθούν οι παράμετροι που αναφέρονται στην εκφώνηση: αριθμός επιπέδων, πλήθος Νευρώνων ανά επίπεδο, ρυθμός μάθησης, εποχές και συνάρτηση ενεργοποίησης. Στο τέλος αυτής της ενότητας υπάρχει πίνακας στον οποίο συνοψίζονται τα αποτελέσματα, καθώς και αναλυτική περιγραφή του τελικού μοντέλου.

Αριθμός Επιπέδων

Η επιλογή επιπέδων χρειάζεται ακρίβεια, καθώς αν βάλουμε λίγα επίπεδα, μπορεί το Τεχνητό Νευρωνικό Δίκτυο (ΤΝΔ) να μην μάθει σύνθετες σχέσεις, και έτσι να έχει μικρότερη ακρίβεια. Από την άλλη, αν βάλουμε πολλά επίπεδα έχουμε μεγάλο υπολογιστικό κόστος, και αυξημένη πιθανότητα για overfitting. Δοκιμάζουμε την ακρίβεια για 1 και 4 κρυφά επίπεδα, αλλά και στις δύο περιπτώσεις το αποτέλεσμα δεν είναι θετικό. Η default επιλογή των 2 επιπέδων λειτουργεί καλύτερα.

Αριθμός Νευρώνων

Όπως και πριν, μεγαλύτερος αριθμός νευρώνων ανά επίπεδο αναμένεται να επιφέρει μεγαλύτερη ακρίβεια, αλλά «κρύβει» μεγαλύτερο υπολογιστικό κόστος και κίνδυνο overfitting. Συνδυαστικά, δοκιμάστηκαν 3 επίπεδα των 128 νευρώνων και 2 επίπεδα των 512 νευρώνων. Η δεύτερη επιλογή εμφανίζει σημαντική βελτίωση στην ακρίβεια. Μπορούμε να συνάγουμε ότι λίγα επίπεδα με

μεγάλο αριθμό νευρώνων λειτουργούν καλύτερα για το δοθέν σύνολο δεδομένων.

Ρυθμός Μάθησης

Κρατάμε τον συνδυασμό 2 επιπέδων των 512 νευρώνων και πειραματιζόμαστε με τον ρυθμό μάθησης. Περιμένουμε ότι μεγαλύτερες τιμές (π.χ. 0.1) εκπαιδεύουν το ΤΝΔ πιο γρήγορα (αλλά με μικρότερη ακρίβεια), ενώ μικρότερες τιμές (π.χ. 0.001) «καθυστερούν» περισσότερο (αλλά είναι πιο ακριβείς). Για ρυθμό μάθησης 0.1, η ακρίβεια μειώνεται αισθητά, ενώ δεν κερδίζουμε καν πολύ χρόνο. Βέβαια, ούτε ο ρυθμός μάθησης 0.0001 αποφέρει βελτίωση: για μόλις 10 εποχές, το ΤΝΔ δεν «προλαβαίνει» να προσεγγίσει καλά τα δεδομένα.

Εποχές

Θέλοντας τώρα να επωφεληθώ από τον μικρό ρυθμό μάθησης (0.0001), δοκίμασα να αυξήσω τον αριθμό των εποχών σε 20. Όμως, όσο περνούσαν οι εποχές, η βελτίωση ήταν όλο και μικρότερη, και η τελική ακρίβεια δεν ήταν ικανοποιητική.

Συνάρτηση Ενεργοποίησης

Με βάση τα πειράματα μέχρι τώρα επιλέγουμε 2 κρυφά επίπεδα των 512 νευρώνων, ρυθμό μάθησης 0.001 (default τιμή) και 15 εποχές. Δοκιμάζουμε τη συνάρτηση ανόρθωσης (ReLU νευρώνες). Τα αποτελέσματα είναι εντυπωσιακά: πετυχαίνουμε την υψηλότερη μέχρι τώρα ακρίβεια (96.22% στο σύνολο επαλήθευσης).

Τελικό Μοντέλο και Σύνοψη Αποτελεσμάτων

Όπως αναφέρθηκε, το τελικό μοντέλο χαρακτηρίζεται από 2 κρυφά επίπεδα των 512 νευρώνων, ρυθμό μάθησης 0.001, 15 εποχές και ReLU συνάρτηση ενεργοποίησης. Τα αποτελέσματα όλων των πειραμάτων συνοψίζονται στον πίνακα:

Αριθμός Κρυφών Επιπέδων	1	4	3	2
Νευρώνες ανά Επίπεδο	256	256	128	512
Συνάρτηση Ενεργοποίησης	tanh	tanh	tanh	tanh
Ρυθμός Μάθησης	0.001	0.001	0.001	0.001
Εποχές	10	10	10	10
Train Accuracy	0.9385	0.9379	0.9216	0.9500
Train Loss	0.2161	0.2111	0.2685	0.1784
Validation Accuracy	0.9362	0.9357	0.9235	0.9435
Validation Loss	0.2197	0.2116	0.2606	0.1929
Αριθμός Κρυφών Επιπέδων	2	2	2	2
Νευρώνες ανά Επίπεδο	512	512	512	512
Συνάρτηση Ενεργοποίησης	tanh	tanh	tanh	relu
Ρυθμός Μάθησης	0.1	0.0001	0.0001	0.001
Εποχές	10	10	20	15
Train Accuracy	0.7212	0.8122	0.9124	1.0000
Train Loss	0.8136	0.6335	0.3260	0.0012
Validation Accuracy	0.7991	0.8278	0.8971	0.9622
Validation Loss	0.6068	0.6696	0.3561	0.2296

Ερωτήσεις Κατανόησης

Σε αυτή την ενότητα δίνονται συνοπτικές απαντήσεις στις ερωτήσεις κατανόησης:

- **Θεωρείτε πως τα δεδομένα της MNIST είναι καλά για την εκπαίδευση ενός μοντέλου; Αιτιολογείστε.** Το σύνολο δεδομένων είναι ιδανικό για την εκπαίδευση ενός μοντέλου. Οι εικόνες είναι αρκετές σε αριθμό, καλύπτουν πολλά στυλ γραφής και δεν υπάρχει σημαντικός θόρυβος. Επίσης, όλα τα δείγματα έχουν τις ίδιες διαστάσεις και βρίσκονται στην ίδια χρωματική κλίμακα. Κάθε εικόνα αποτελείται από λίγα σχετικά pixel, οπότε μπορεί εύκολα να εκπαιδευτεί ένα ΤΝΔ με μικρό (σχετικά) υπολογιστικό κόστος. Προφανώς, το σύνολο δεδομένων δεν ενδείκνυται για μεγάλης κλίμακας εφαρμογή.
- **Θεωρείτε πως όλα τα pixel είναι σημαντικά για την πρόβλεψη της κλάσης ενός ψηφίου;** Μερικά pixels δεν είναι τόσο σημαντικά για την πρόβλεψη της κλάσης ενός ψηφίου. Για παράδειγμα, πολλά pixels - ειδικά στην άκρη της οθόνης - είναι πάντα μαύρα και δεν έχουν

χρήσιμη πληροφορία. Πιο σημαντικά είναι τα άσπρα και γκρι pixels. Σε αυτή την αρχή βασίζεται και η τεχνική των autoencoders.

- **Σε ποιες περιπτώσεις είναι καλή ιδέα να χρησιμοποιηθούν Βαθιά Νευρωνικά Δίκτυα;** Σύμφωνα με τις διαφάνειες, τα Βαθιά Νευρωνικά Δίκτυα (BNΔ) λύνουν με εκπληκτική αποτελεσματικότητα προβλήματα στα οποία η είσοδος και η έξοδος έχουν πολύπλοκες μη γραμμικές σχέσεις. Αυτό συμβαίνει για παράδειγμα όταν τα δεδομένα είναι εικόνες, ήχος, φυσική γλώσσα ή ολόκληρα κείμενα. Τα BNΔ δουλεύουν καλύτερα με μεγάλο όγκο δεδομένων και λύνουν προβλήματα ταξινόμησης, αναγνώρισης, αποτίμησης και πρόβλεψης. Γενικότερα, τα TNΔ είναι αρκετά ισχυρά, αλλά λόγω του υψηλού κόστους, χρησιμοποιούνται όταν όλες οι υπόλοιπες τεχνικές στο πεδίο της Τεχνητής Νοημοσύνης αποτυγχάνουν.
- **Η Βαθιά Μάθηση μπορεί να χρησιμοποιηθεί και στους 3 κλάδους της Μηχανικής Μάθησης; (Supervised Learning, Unsupervised Learning, Reinforcement Learning)** Η Βαθιά Μάθηση βρίσκει εφαρμογές και στους 3 κλάδους της Μηχανικής Μάθησης:
 1. Στην επιβλεπόμενη μάθηση, μπορούν να εκπαιδευτούν TNΔ που λύνουν προβλήματα πρόβλεψης και ταξινόμησης. Για παράδειγμα, τα δεδομένα της MNIST, τα οποία περιέχουν ετικέτες, βοηθούν στην εκπαίδευση ενός μοντέλου που ταξινομεί εικόνες.
 2. Στη μη επιβλεπόμενη μάθηση, μοντέλα όπως οι Autoencoders και τα Generative Adversarial Networks (GANs) μπορούν να ανακαλύψουν πρότυπα σε data sets ή να δημιουργήσουν νέα δεδομένα αντίστοιχα.
 3. Στην ενισχυτική μάθηση, μπορούμε να προσεγγίσουμε την τιμή της Q-function χρησιμοποιώντας ένα BNΔ το οποίο εκτιμά την αναμενόμενη συνολική ανταμοιβή του πράκτορα για κάθε ενέργεια σε δοθείσα κατάσταση. Τα Deep Q-Networks (DQNs) χρησιμοποιούνται δηλαδή για να επεκτείνουν τον αλγόριθμο Q-Learning.

Χρήσιμοι Σύνδεσμοι

[Github Repository](#)

[Google Colab Notebook](#)