

TN: Εργασία στη Βαθιά Μάθηση

Ονοματεπώνυμο: Οκλαλιώτης Παντελής-Παναγιώτης
ΑΕΜ: 4626

Hyper-parameter Tuning

Παραλλαγή	Layers	Νευρώνες/Layer	Activation	Learning Rate	Epochs	Accuracy	Loss	Validation accuracy	Validation loss
A	2	512,256	relu	0.001	10	0.9956	0.0195	0.9561	0.2127
B	3	512,256,128	relu	0.001	15	0.9992	0.0080	0.9600	0.1730
C	2	512,256	relu	0.01	10	0.0989	NaN	0.0980	NaN
D	2	512,256	sigmoid	0.001	10	0.9010	0.4226	0.9034	0.4010
E	2	128,64	relu	0.001	10	0.9563	0.1470	0.9471	0.2030
F	3	256,128,64	relu	0.001	20	0.9950	0.0237	0.9573	0.2018

Δοκιμάστηκαν πολλές διαφορετικές παράμετροι, ώστε να επιλέξουμε τις καλύτερες για το δικό μας δίκτυο. Η βέλτιστη προσέγγιση με βάση τα στατιστικά που έχουμε δίνεται από την παραλλαγή B:

- **Layers:** 3 (αυξάνει τη δυνατότητα μάθησης πιο σύνθετων προτύπων)
- **Νευρώνες:** 512,256,128 (σταδιακή μείωση βοηθάει σε regularization)
- **Activation:** relu (καλύτερη απόδοση από tanh και sigmoid)
- **Learning Rate:** 0.001 (σταθερή μάθηση χωρίς ταλαντώσεις)
- **Epochs:** 15 (επιτρέπει τη σύγκλιση του loss)

Improvements

Για τη βελτίωση της απόδοσης του νευρωνικού δικτύου εκτός από τη **Hyper-parameter Tuning**, εφαρμόστηκαν κάποιες βελτιώσεις στην αρχιτεκτονική και τα δεδομένα του δικτύου:

1. **Κανονικοποίηση των εισόδων στο εύρος [0, 1]** : Τα δεδομένα MNIST έχουν τιμές pixel στο εύρος [0,255]. Κανονικοποιώντας τα ώστε να βρίσκονται στο [0,1], βοηθάμε το δίκτυο να μάθει πιο γρήγορα και σταθερά.
2. **Προσθέτουμε momentum=0.9 στο SGD, ώστε να επιταχύνουμε τη διαδικασία μάθησης.**
3. **Αυξάνουμε τις εποχές σε 30 για να αυξήσουμε τη σύγκλιση του loss.**
4. **Προσθέτουμε Dropout layers (0.2), ώστε να μειωθεί το overfitting απενεργοποιώντας τυχαία νευρώνες κατά την εκπαίδευση.**

Ερωτήσεις κατανόησης

- a) Τα δεδομένα MNIST θεωρούνται καλά για την εκπαίδευση ενός μοντέλου, ειδικά για εκπαιδευτικούς σκοπούς και απλές εφαρμογές ταξινόμησης εικόνων. Περιέχει 10 κατηγορίες (0–9) με περίπου **ίσο αριθμό δειγμάτων ανά κατηγορία** και οι εικόνες είναι μικρές (28×28 px, grayscale), ιδανικές για ταχύ πειραματισμό. Ωστόσο, έχουν και ορισμένους περιορισμούς. Το dataset είναι αρκετά απλοϊκό, καθώς οι εικόνες είναι κεντραρισμένες, καθαρές και δεν αντικατοπτρίζουν τον "θόρυβο" του πραγματικού κόσμου. Επίσης, δεν καλύπτουν σύνθετη μεταβλητότητα, όπως παραμορφώσεις και περιστροφές.
- b) Όχι, **δεν είναι όλα τα pixels εξίσου σημαντικά** για την πρόβλεψη της κλάσης ενός ψηφίου στο MNIST. Ορισμένα pixels περιέχουν **χρήσιμη πληροφορία**, ενώ άλλα είναι **σχεδόν πάντα "κενά" ή μηδενικά** και δεν συμβάλλουν ουσιαστικά στη μάθηση του μοντέλου.
- c) Η χρήση **Βαθιών Νευρωνικών Δικτύων** είναι καλή ιδέα **όταν το πρόβλημα έχει μεγάλη πολυπλοκότητα και απαιτεί την κατανόηση σύνθετων ή αφηρημένων προτύπων**. Κάποιες περιπτώσεις όπου τα DNNs υπερέχουν είναι η επεξεργασία εικόνων, αναγνώριση ομιλίας, επεξεργασία φυσικής γλώσσας, ενισχυτική μάθηση, κτλ.
- d) **Η Βαθιά Μάθηση** μπορεί να χρησιμοποιηθεί **και στους τρεις βασικούς κλάδους της Μηχανικής Μάθησης**:
- **Επιβλεπόμενη μάθηση**: Πρόβλεψη αν μια εικόνα περιέχει γάτα ή σκύλο. Μπορούμε να χρησιμοποιήσουμε CNNs για ταξινόμηση εικόνων και Deep regression models για πρόβλεψη συνεχών τιμών.
 - **Μη επιβλεπόμενη μάθηση**: Ομαδοποίηση χειρόγραφων ψηφίων με clustering. Μπορούμε να χρησιμοποιήσουμε autoencoders για εξαγωγή χαρακτηριστικών και GANs για παραγωγή δεδομένων.
 - **Ενισχυτική μάθηση**: Ένας πράκτορας παίζει σκάκι και βελτιώνεται με εμπειρία. Μπορούμε να χρησιμοποιήσουμε DQNs για βελτιστοποίηση αποφάσεων και CNNs για είσοδο από εικόνες.

GitHub Repository URL: <https://github.com/pokla6/mnist-digits-classification.git>

Notebook URL: <https://colab.research.google.com/drive/1Nmj732D-K51QMKNXitgOGoflv-rsCYP?usp=sharing>