



3^η Εργασία

Αρχιτεκτονική CNN

Μηχανική Μάθηση και Επεξεργασία Φυσικής Γλώσσας

Χριστακάκης Παναγιώτης

A.M.: aid23004



**Πρόγραμμα Μεταπτυχιακών Σπουδών «Τεχνητή Νοημοσύνη & Αναλυτική Δεδομένων» ,
Πανεπιστήμιο Μακεδονίας**

Ημερομηνία: 13/11/2022

Αρχιτεκτονική CNN

Πίνακας Περιεχομένων

1	Εισαγωγή	3
2	Μέθοδοι	4
3	Συμπεράσματα	11

Πίνακες

Πίνακας 2.1: Classification Report του 1 ^{ου} Μοντέλου	6
Πίνακας 2.2: Classification Report του 2 ^{ου} Μοντέλου	8
Πίνακας 2.3: Classification Report του 3 ^{ου} Μοντέλου	10
Πίνακας 2.4: Σύγκριση train-test accuracy & loss	10

Εικόνες

Εικόνα 2.1: Γράφημα τιμών train & validation loss του 1 ^{ου} Μοντέλου.....	4
Εικόνα 2.2: Γράφημα τιμών train & validation accuracy του 1 ^{ου} Μοντέλου	5
Εικόνα 2.3: Γράφημα τιμών train & validation loss του 2 ^{ου} Μοντέλου.....	7
Εικόνα 2.4: Γράφημα τιμών train & validation accuracy του 2 ^{ου} Μοντέλου.....	7
Εικόνα 2.5: Γράφημα τιμών train & validation loss του 3 ^{ου} Μοντέλου.....	9
Εικόνα 2.6: Γράφημα τιμών train & validation accuracy του 3 ^{ου} Μοντέλου.....	9

1 Εισαγωγή

Σκοπός του προβλήματος του οποίου καλούμαστε να αντιμετωπίσουμε είναι η δημιουργία του καλύτερου δυνατού συνελικτικού νευρωνικού δικτύου για το CIFAR-10 dataset.

Το συγκεκριμένο dataset περιέχει 60.000 εικόνες μεγέθους 32 x 32 σε RGB χρωματισμό και κάθε μία από αυτές αντιστοιχεί σε μία από τις δέκα διαθέσιμες κατηγορίες. Οι κατηγορίες είναι οι εξής:

1. Αεροπλάνο
2. Αυτοκίνητο
3. Πτηνό
4. Γάτα
5. Ελάφι
6. Σκύλος
7. Βάτραχος
8. Άλογο
9. Πλοίο
10. Φορητό

Στα πειράματα που θα εκτελέσουμε θα δοκιμαστούν αρχιτεκτονικές συνελικτικών νευρωνικών δικτύων με διαφορετικές παραμέτρους. Θα ξεκινήσουμε με πιο απλές δομές αρχιτεκτονικών, στη συνέχεια θα αλλάξουμε τη συνάρτηση απώλειας κατά τη διάρκεια του model training και θα καταλήξουμε σε κάτι πιο σύνθετο και που ελπίζουμε να μας δώσει το επιθυμητό αποτέλεσμα.

Φυσικά, στην προσπάθεια μας να αυξήσουμε την αποδοτικότητα του μοντέλου μας χρησιμοποιώντας μια πιο περίπλοκη δομή, ενδέχεται να συναντήσουμε προβλήματα υπερπροσαρμογής που καλούμαστε να λύσουμε στον μέγιστο δυνατό βαθμό με διάφορες τεχνικές.

2 Μέθοδοι

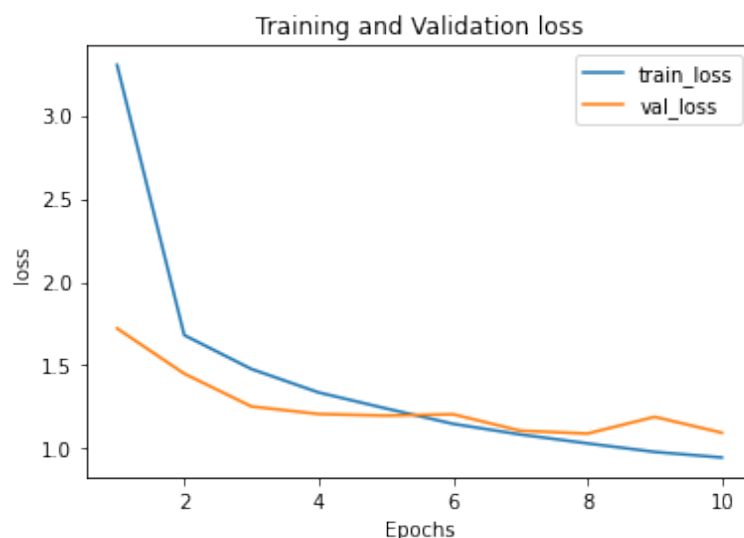
Για την υλοποίηση των μεθόδων μας χρησιμοποιήσαμε την γλώσσα python και τις κατάλληλες βιβλιοθήκες. Ενδεικτικά αναφέρονται οι pandas, numpy, sklearn, keras, matplotlib, tensorflow, random και άλλες.

Από τη βιβλιοθήκη του keras με τα datasets, προσφέρεται η δυνατότητα να φορτωθούν τα δεδομένα απευθείας χωρισμένα σε train και test. Πρώτο και κύριο μέλημα είναι να κρατήσουμε ένα κομμάτι από τα δεδομένα εκπαίδευσης για να δημιουργηθούν τα validation data. Τα δεδομένα αυτά επικυρώνουν το μοντέλο μας και διασφαλίζουν την ακρίβειά του. Συνεχίζοντας, στον κώδικά μας εκτυπώνονται σχετικές πληροφορίες με το μέγεθος και σχηματισμό των δεδομένων, ενώ παράλληλα εμφανίζεται ένα δείγμα εικόνων από κάθε κλάση.

Πριν ξεκινήσουμε να δημιουργούμε το μοντέλο μας, μετατρέπουμε τα δεδομένα σε μορφή που είναι κατάλληλη για να τα δεχτεί το συνελκτικό νευρωνικό μας δίκτυο. Αρχικά, θα δοκιμάσουμε ένα μοντέλο με τα δύο πρώτα επίπεδα να είναι συνελκτικά. Με συνάρτηση ενεργοποίησης την 'Relu' και μέγεθος 3 x 3 του παραθύρου συνέλιξης, γίνεται η πρώτη εξαγωγή χαρακτηριστικών των εικόνων. Το επόμενο επίπεδο MaxPooling μειώνει το μέγεθος της εικόνας, ενώ στη συνέχεια προσθέτουμε και ένα Dropout layer ώστε να αφήνει τυχαία κάποιους νευρώνες έκτος εκπαίδευσης, για την αποφυγή της υπερπροσαρμογής των δεδομένων μας. Τέλος, το συγκεκριμένο μοντέλο καταλήγει με δύο απλά νευρωνικά δίκτυα 128 και 10 νευρώνων που δίνουν και το τελικό αποτέλεσμα.

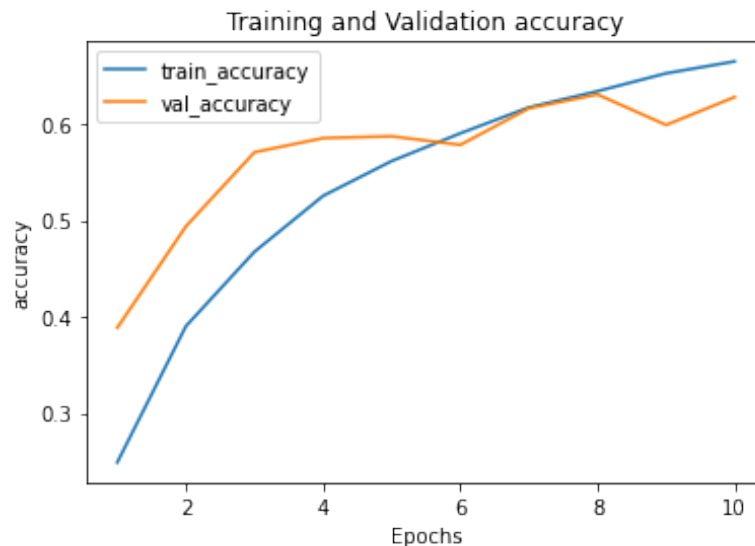
Κατά την εκπαίδευση επιλέξαμε το μέγεθος των batches να είναι 64, ενώ οι εποχές εκπαίδευσης ορίστηκαν στις 10. Εκτυπώνοντας κάποιες πρώτες αξιολογήσεις για το μοντέλο μας βλέπουμε ότι η ακρίβεια φτάνει στο 62%. Πρέπει όμως να δούμε τα αποτελέσματα αναλυτικότερα.

Εικόνα 2.1: Γράφημα τιμών train & validation loss του 1^{ου} Μοντέλου



Σχόλια: Στην παραπάνω εικόνα βλέπουμε τη σύγκριση τιμών απώλειας για τα δεδομένα εκπαίδευσης και επικύρωσης. Και οι δύο δείκτες έχουν μια φυσιολογική πτωτική πορεία και ειδικά μετά την 5^η εποχή εκπαίδευσης σχεδόν ταυτίζονται. Φτάνοντας στην 8^η εποχή και έπειτα το validation loss αρχίζει να αυξάνεται και να ξεπερνά το training loss.

Εικόνα 2.2: Γράφημα τιμών train & validation accuracy του 1^{ου} Μοντέλου



Σχόλια: Στο γράφημα αυτό μπορούμε να διακρίνουμε την ανοδική πορεία και των δύο δεικτών, που δείχνει ότι το μοντέλο αποδίδει όπως θέλουμε. Το γεγονός ότι δεν φτάνει υψηλά η ακρίβεια μας δείχνει ότι πρέπει να χρησιμοποιήσουμε διαφορετική αρχιτεκτονική. Αναλύοντας περαιτέρω την εικόνα, βλέπουμε ότι στις τελευταίες εποχές εκπαίδευσης το train accuracy αρχίζει και ξεπερνά το αντίστοιχο validation accuracy. Εδώ, ξεκινάει το πρόβλημα της υπερπροσαρμογής. Το μοντέλο μας γίνεται πιο εξειδικευμένο στα δεδομένα εκπαίδευσης και αρχίζει να μην μπορεί να γενικευτεί σε νέα δεδομένα, με αποτέλεσμα να βλέπουμε το validation accuracy σε χαμηλότερες τιμές.

Πίνακας 2.1: Classification Report του 1^{ου} Μοντέλου

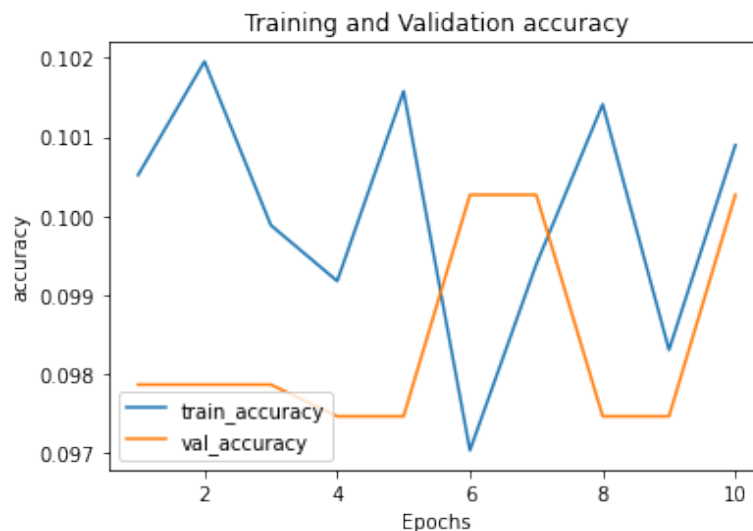
Classification Report				
	precision	recall	f1-score	support
0	0.64	0.69	0.66	1000
1	0.71	0.83	0.76	1000
2	0.55	0.45	0.49	1000
3	0.41	0.38	0.39	1000
4	0.51	0.57	0.54	1000
5	0.61	0.42	0.50	1000
6	0.53	0.88	0.66	1000
7	0.80	0.62	0.70	1000
8	0.75	0.72	0.73	1000
9	0.78	0.65	0.71	1000
accuracy			0.62	10000
macro avg	0.63	0.62	0.62	10000
weighted avg	0.63	0.62	0.62	10000

Σχόλια: Ο Πίνακας 2.1, περιέχει δεδομένα μετρικών για κάθε διαθέσιμη κλάση. Μια απλή προσέγγιση για να αντιληφθούμε τον συγκεκριμένο πίνακα, είναι να δούμε το f1-score που υπολογίζεται από τον αρμονικό μέσο όρο των precision και recall. Οι μεγαλύτερη τιμή σε αυτόν τον δείκτη εμφανίζεται στην 2^η κλάση. Το μοντέλο καταφέρνει να ξεχωρίσει με μεγαλύτερη επιτυχία φωτογραφίες με αυτοκίνητα.

Στο Πείραμα 2, μας ζητείται να δημιουργήσουμε ένα μοντέλο με την ίδια αρχιτεκτονική αλλά με διαφορετική συνάρτηση απώλειας κατά το training. Επιλέξαμε να χρησιμοποιήσουμε τη συνάρτηση του μέσου τετραγωνικού σφάλματος (MSE), η οποία μετρά τη μέση τετραγωνική διαφορά μεταξύ των εκτιμώμενων τιμών και της πραγματικής τιμής. Η συγκεκριμένη συνάρτηση απώλειας δεν χρησιμοποιείται σε προβλήματα classification αλλά regression. Οπότε περιμένουμε τα αποτελέσματα να είναι χειρότερα από το 1^ο μοντέλο. Με μια πρώτη ματιά, το μοντέλο μας φαίνεται να έχει πολύ κακά αποτελέσματα, τα οποία δεν ξεπερνούν το 10% accuracy.

Εικόνα 2.3: Γράφημα τιμών train & validation loss του 2^{ου} Μοντέλου

Σχόλια: Με μια πρώτη ματιά παρατηρούμε αστάθεια στις δύο μετρικές των δεδομένων εκπαίδευσης και επαλήθευσης. Παρόλα αυτά αν κοιτάξουμε τον κάθετο άξονα Υ, βλέπουμε ότι οι τιμές του είναι κοντά και οι δύο δείκτες στην πραγματικότητα πάντα κυμαίνονται κοντά στο 0.18. Εάν αφήναμε το μοντέλο να εκπαιδευτεί για περισσότερες εποχές, το παραπάνω γράφημα ενδέχεται να φαινόταν πιο σταθερό και για τις δύο αυτές τιμές των μετρικών.

Εικόνα 2.4: Γράφημα τιμών train & validation accuracy του 2^{ου} Μοντέλου

Σχόλια: Στο παραπάνω γράφημα έχουμε περίπου το ίδιο αποτέλεσμα με την Εικόνα 2.1. Training και validation accuracy φαίνεται να έχουν έντονες ανακατατάξεις στις τιμές τους, ενώ στην πραγματικότητα οι τιμές είναι πάντα κοντά στο 0.10 ή 10% accuracy. Φυσικά, μπορούμε να διακρίνουμε εμφανώς ότι η συγκεκριμένη loss function που εφαρμόσαμε είναι αναποτελεσματική.

Πίνακας 2.2: Classification Report του 2^{ου} Μοντέλου

Classification Report				
	precision	recall	f1-score	support
0	0.00	0.00	0.00	1000
1	0.00	0.00	0.00	1000
2	0.00	0.00	0.00	1000
3	0.00	0.00	0.00	1000
4	0.00	0.00	0.00	1000
5	0.10	1.00	0.18	1000
6	0.00	0.00	0.00	1000
7	0.00	0.00	0.00	1000
8	0.00	0.00	0.00	1000
9	0.00	0.00	0.00	1000
accuracy			0.10	10000
macro avg	0.01	0.10	0.02	10000
weighted avg	0.01	0.10	0.02	10000

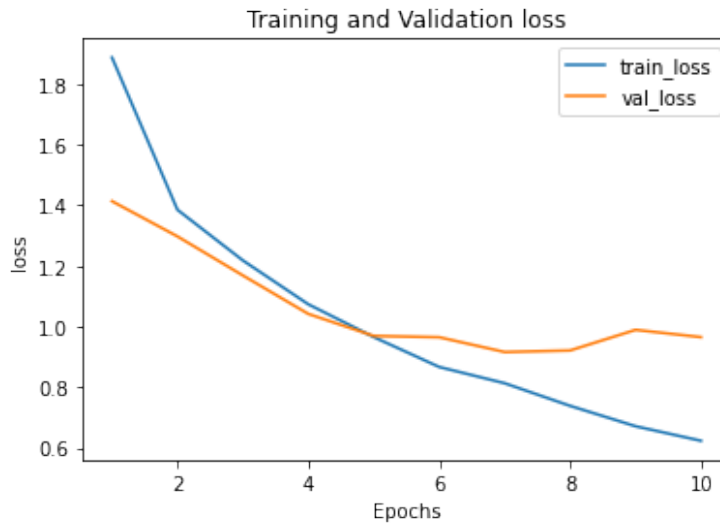
Σχόλια: Ο Πίνακας 2.1, με τα δεδομένα μετρικών για τις κλάσεις μάς, φαίνεται να είναι σχεδόν παντού 0. Το μοντέλο με συνάρτηση απώλειας Mean Squared Error κατάφερε να προβλέψει σωστά μόνο κάποιες φωτογραφίες της κλάσης 5, αυτή με τους σκύλους. Συνεπώς, όλα τα υπόλοιπα αποτελέσματα είναι μηδενικά, αφού δεν υπήρχε και η κατάλληλη πρόβλεψη από το μοντέλο του Πειράματος 2.

Συνεχίζοντας με το 3^ο πείραμα, καλούμαστε να χρησιμοποιήσουμε μια πιο περίπλοκη δομή με extra convolutional layers ή kernels. Έτσι, η αρχιτεκτονική που επιλέξαμε για το μοντέλο είναι:

- Αρχικό Convolutional layer με 64 νευρώνες, συνάρτηση ενεργοποίησης την relu, kernel 3 x 3 και είσοδο εικόνων 32 x 32 x 3.
- Convolutional layer με 64 νευρώνες, συνάρτηση ενεργοποίησης την relu και kernel 3 x 3.
- MaxPooling2D layer με μέγεθος pooling 2 x 2.
- Dropout στρώμα με 30% αποκλεισμό νευρώνων για την αποφυγή υπεπροσαρμογής.
- Convolutional layer με 32 νευρώνες, συνάρτηση ενεργοποίησης την relu και kernel 3 x 3.
- Convolutional layer με 64 νευρώνες, συνάρτηση ενεργοποίησης την relu και kernel 3 x 3.
- MaxPooling2D layer με μέγεθος pooling 2 x 2.
- Flatten layer για την προετοιμασία εισόδου σε απλό νευρωνικό δίκτυο.
- Layer 1024 απλών νευρώνων με συνάρτηση ενεργοποίησης την relu.
- Layer 512 απλών νευρώνων με συνάρτηση ενεργοποίησης την relu.
- Dropout στρώμα με 20% αποκλεισμό νευρώνων για την αποφυγή υπεπροσαρμογής.
- Τελικό layer 10 νευρώνων που θα δώσουν και το αποτέλεσμα κατηγοριοποίησης της εικόνας, με συνάρτηση ενεργοποίησης την softmax.

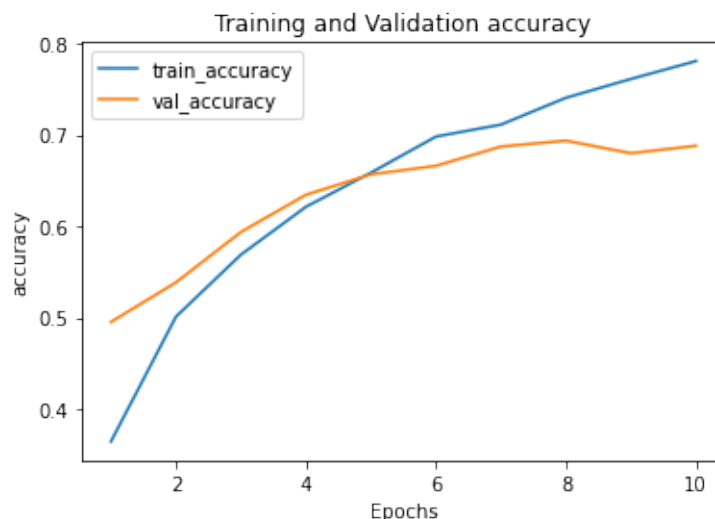
Ως optimizer επιλέχθηκε ο 'adam' καθώς με δοκιμές φαίνεται να έχει πιο καλά και γρήγορα αποτελέσματα και ως συνάρτηση απώλειας επιλέξαμε την categorical crossentropy. Ο αριθμός των batch size ήταν 64, ενώ το μοντέλο εκπαιδεύτηκε για 10 εποχές.

Εικόνα 2.5: Γράφημα τιμών train & validation loss του 3^{ου} Μοντέλου



Σχόλια: Κατά τις εποχές εκπαίδευσης της Εικόνας 2.5, εμφανίζεται το train loss να έχει φθίνουσα πορεία. Αντίθετα, η τιμή του validation loss εμφανίζεται να έχει αυξομειώσεις από την 6^η εποχή εκπαίδευσης και μετά πράγμα που μας δείχνει ενδεχόμενη υπερπροσαρμογή στα δεδομένα.

Εικόνα 2.6: Γράφημα τιμών train & validation accuracy του 3^{ου} Μοντέλου



Σχόλια: Αναμενόμενα, όπως και στην προηγούμενη εικόνα, οι δύο μετρικές training και validation accuracy μας δείχνουν ξεκάθαρα ότι το μοντέλο μας έχει υπερπροσαρμογή στα δεδομένα εκπαίδευσης. Στο συγκεκριμένο διάγραμμα αυτό διακρίνεται διότι το train accuracy έχει συνεχόμενη ανοδική πορεία, ενώ το validation accuracy μετά από κάποιες εποχές ξεκινάει και είναι σταθερή ή και λίγο πτωτική.

Πίνακας 2.3: Classification Report του 3^{ου} Μοντέλου

Classification Report				
	precision	recall	f1-score	support
0	0.74	0.69	0.71	1000
1	0.84	0.81	0.83	1000
2	0.56	0.58	0.57	1000
3	0.49	0.50	0.49	1000
4	0.57	0.71	0.63	1000
5	0.68	0.46	0.55	1000
6	0.65	0.85	0.74	1000
7	0.82	0.69	0.75	1000
8	0.77	0.82	0.79	1000
9	0.83	0.74	0.78	1000
accuracy			0.69	10000
macro avg	0.69	0.68	0.68	10000
weighted avg	0.69	0.69	0.68	10000

Σχόλια: Ο Πίνακας 2.3, μας προσφέρει κάποιες επιπλέον πληροφορίες συγκεκριμένα με τις εκτιμήσεις του μοντέλου. Πιο συγκεκριμένα, μπορούμε να δούμε ποιες κλάσεις προβλεφθήκαν με μεγαλύτερη επιτυχία. Οι κλάσεις 1, 8, 9 δηλαδή αυτές των αυτοκινήτων, των πλοίων και των φορτηγών ήταν ανάμεσα σε αυτές που το μοντέλο ξεχώριζε με επιτυχία σχεδόν 80%.

Από την σύγκριση αποτελεσμάτων των πειραμάτων μεταξύ train και test sets προκύπτει ο παρακάτω πίνακας.

Πίνακας 2.4: Σύγκριση train-test accuracy & loss

	Πείραμα 1 ^ο	Πείραμα 2 ^ο	Πείραμα 3 ^ο
train accuracy	0.6653	0.1009	0.7809
test accuracy	0.6200	0.1798	0.6850
train loss	0.9461	0.1000	0.6237
test loss	1.1070	0.1800	0.9782

Σχόλια: Συγκρίνοντας τα παραπάνω αποτελέσματα παρατηρούμε ότι στο 1^ο και στο 3^ο πείραμα η ακρίβεια στα training δεδομένα ήταν υψηλότερη απ' ότι στα test. Αυτό, σε συνδυασμό του ότι τα αντίστοιχα test losses και των δύο πειραμάτων ήταν υψηλότερα από τα train, μας δίνει μια εικόνα υπερπροσαρμογής των μοντέλων. Το 2^ο πείραμα μας δείχνει ότι το μοντέλο δεν ήταν καθόλου καλό, αφού οι τιμές είναι εξίσου χαμηλές σε όλες τις μετρικές.

3 Συμπεράσματα

Συμπεραίνοντας, το βέλτιστο μοντέλο βάση αποτελεσμάτων είναι αυτό που παρουσιάστηκε στο 3^ο πείραμα. Παρόλα αυτά, όπως φάνηκε εμφανίστηκε overfitting της τάξεως του 10%, που μπορεί να μην είναι αρκετά μεγάλο αλλά σε ενδεχόμενη συνέχιση της εκπαίδευσης με περισσότερες εποχές, θα φαινόταν περισσότερο. Ο εκάστοτε ερευνητής ανάλογα με τα αποτελέσματα που θέλει να πάρει, μπορεί να αποφασίσει ποιο θα είναι το ανεκτό όριο υπερπροσαρμογής του μοντέλου του. Εδώ, θα μπορούσαμε να καταπολεμήσουμε αυτό το πρόβλημα αρχικά δοκιμάζοντας να αυξήσουμε τα ποσοστά dropout στους νευρώνες του μοντέλου.

Το CIFAR – 10 dataset παρόλο που ακούγεται επαρκές επειδή έχει 6.000 φωτογραφίες για κάθε κλάση, στην πραγματικότητα δεν είναι. Εάν ήταν μεγαλύτερο, η ύπαρξη περισσότερων εικόνων για κάθε μία από τις δέκα κλάσεις θα βοηθούσε αρκετά την εκπαίδευση του μοντέλου βαθιάς μάθησης και φυσικά στην βελτίωση της ακρίβειας του μοντέλου. Μία τεχνική που χρησιμοποιείται για την καταπολέμηση των μικρότερων dataset είναι η ‘Image Data Augmentation’. Με αυτήν, επεκτείνουμε τεχνητά το σύνολο δεδομένων εκπαίδευσης δημιουργώντας τροποποιημένες εκδόσεις όλων των εικόνων που υπάρχουν στα δεδομένα. Η τυχαία περιστροφή, το ζουμ, η αλλαγή χρωμάτων στα κανάλια RGB, η θόλωση και η διαγραφή μέρους της εικόνας συμβαίνουν κατά την εφαρμογή αυτής της τεχνικής και μας προσφέρουν νέες εικόνες από τις οποίες καμία δεν είναι ίδια με τις προηγούμενες. Αυτό επίσης βοηθάει στην βελτίωση της γενίκευσής του μοντέλου.

Τέλος, δύο από τις δύο συνηθέστερες συναρτήσεις απώλειας είναι οι Sparse Categorical Cross-Entropy και Categorical Cross-Entropy. Η πρώτη χρησιμοποιείται όταν οι κλάσεις είναι αμοιβαία αποκλειόμενες και δεν υπάρχει περίπτωση να έχουμε πολλαπλές κλάσεις σε ένα παράδειγμα – φωτογραφία. Η δεύτερη χρησιμεύει όταν ένα παράδειγμα μπορεί να έχει πολλές κλάσεις. Στα πειράματά μας χρησιμοποιήθηκε η δεύτερη, επομένως μια πρόταση για βελτίωση ενός μοντέλου για το CIFAR – 10 dataset θα ήταν η επιλογή της Sparse Categorical Cross-Entropy ως loss function.

Συμπερασματικά, θα ήταν ενδιαφέρον να δοκιμαζόταν το συγκεκριμένο dataset μετά από αύξηση δεδομένων με τη τεχνική που προηγουμένως περιγράψαμε, να είχε χρησιμοποιηθεί διαφορετική loss function και οι εποχές εκπαίδευσης με το batch size να ήταν αρκετά μεγαλύτερα νούμερα.