

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών

**« 3η Εργαστηριακή Άσκηση για το μάθημα:**

**Νευρωνικά Δίκτυα και Ευφυή Υπολογιστικά Συστήματα -**

**Deep Learning »**

του 9ου Εξαμήνου

του φοιτητή:

**Καρυπιάδη Ευστρατίου, Α.Μ.: 03114911, Ομάδα: Α11**

**Πίνακας Περιεχομένων**

[**Σύνολο Δεδομένων** 3](#_Toc35810583)

[**Μοντέλα Δικτύων** 3](#_Toc35810584)

[**Συνελικτικό Νευρωνικό Δίκτυο *from-scratch*** 3](#_Toc35810585)

[**Αρχιτεκτονική** 3](#_Toc35810586)

[**Αλγόριθμος Βελτιστοποίησης - Optimizer** 4](#_Toc35810587)

[**Συνάρτηση Ενεργοποίησης – Activation Function** 5](#_Toc35810588)

[**Batch Size** 6](#_Toc35810589)

[**Data Augmentation** 6](#_Toc35810590)

[**Αποτελέσματα – Τελικές Σκέψεις και Παρατηρήσεις** 7](#_Toc35810591)

[**Μεταφορά Μάθησης – Transfer Learning** 7](#_Toc35810592)

[**Trainable vs Non-Trainable** 7](#_Toc35810593)

[**Αποτελέσματα – Τελικές Σκέψεις και Παρατηρήσεις** 8](#_Toc35810594)

[**Αναφορές** 9](#_Toc35810595)

# **Σύνολο Δεδομένων**

Στα πλαίσια της άσκησης χρησιμοποιήσαμε το σύνολο δεδομένων CIFAR-100, το οποίο συντί-θεται από 100 κλάσεις, κάθε μία από τις οποίες διαθέτει 500 εικόνες προς εκπαίδευση και 100 προς έλεγχο. Οι διαστάσεις των εικόνων είναι 32x32, ενώ σύμφωνα με το υποσύνολο που αντιστοιχεί στην ομάδα μας εκμεταλλευτήκαμε τελικά 80 κλάσεις με αποτέλεσμα να έ-χουμε στη διάθεσή μας 40.000 εικόνες. Σημειώνουμε ότι το 15% αυτών το ξεχωρίσαμε προς επαλήθευση κατά τη διαδικασία εκπαίδευσης των μοντέλων μας, με αποτέλεσμα το τελικό πλήθος των συνόλων μας να διαμορφώνεται ως εξής:

* 34.000 εικόνες για εκπαίδευση
* 6.000 εικόνες για επαλήθευση κατά την εκπαίδευση
* 8.000 εικόνες για τον τελικό έλεγχο

Τέλος, σημειώνουμε ότι το σύνολο του ελέγχου δεν τροφοδοτήθηκε ποτέ στα μοντέλα μας και χρησιμοποιήθηκε μονάχα στο τέλος κάθε κύκλου εκπαίδευσης προς εκτίμησης της από-δοσης των αρχιτεκτονικών που εξετάστηκαν.

# **Μοντέλα Δικτύων**

Σχετικά με τα μοντέλα δικτύων ακολουθήσαμε δύο μεθόδους εκπαίδευσης. Κατά τον πρώτο υλοποιήσαμε *from-scratch* ένα σχετικά απλό συνελικτικό δίκτυο, ενώ κατά το δεύτερο ακο-λουθήσαμε τη μέθοδο του *transfer-learning*, κατά την οποία εκμεταλλευτήκαμε ένα αρκετά πιο πολύπλοκο δίκτυο που είχε προ-εκπαιδευτεί σε μεγαλύτερο όγκο δεδομένων από το δικό μας. Η παραπάνω προσέγγιση αποδείχθηκε χρήσιμη, αφενός επειδή είχαμε τη δυνατότητα να συγκρίνουμε την απόδοση δικτύων διαφορετικής πολυπλοκότητας και αφετέρου, επειδή μας δόθηκε η ευκαιρία να πειραματιστούμε με τις διαφορετικές μεθόδους εκπαίδευσης που ακολουθούνται σήμερα στην εκπαίδευση συνελικτικών νευρωνικών δικτύων.

## **Συνελικτικό Νευρωνικό Δίκτυο *from-scratch***

Στη συνέχεια θα αναλύσουμε τις παραμέτρους του δικτύου που λάβαμε υπόψιν και πιο συ-γκεκριμένα την αρχιτεκτονική, τον *optimizer*, τη συνάρτηση ενεργοποίησης, το *batch-size*, καθώς και την επίδραση της τεχνικής *data-augmentation* στο μοντέλο μας.

### **Αρχιτεκτονική**

Η αρχιτεκτονική του μοντέλου μας είναι έντονα επηρεασμένη από εκείνη του δικτύου *VGG-16*. Πιο συγκεκριμένα, αποτελείται από τρία κεντρικά επίπεδα, καθ’ ένα από τα οποία συντί-θεται με τη σειρά του από δύο σειριακά συνελικτικά επίπεδα(*Conv2D*) και ένα επίπεδο μείω-σης της διαστατικότητας(*MaxPooling2D*). Οι διαστάσεις του συνελικτικού επιπέδου είναι 3x3 και εκείνες του φίλτρου 2x2. Στο τέλος των 3 κεντρικών επιπέδων προσθέτουμε ένα επίπεδο *Flatten*, ενώ στη συνέχεια τοποθετούμε δύο πλήρως διασυνδεδεμένα επίπεδα, προκειμένου να τροφοδοτήσουμε στο δίκτυο τις συνολικές εικόνες εκπαίδευσης, όπως συνηθίζεται. Ση-μειώνουμε ότι το τελευταίο επίπεδο έχει διαστάσεις που συμπίπτουν με το πλήθος των κλά-σεων του προβλήματος που επιχειρούμε να επιλύσουμε, στη συγκεκριμένη, δηλαδή, περί-πτωση, 100.

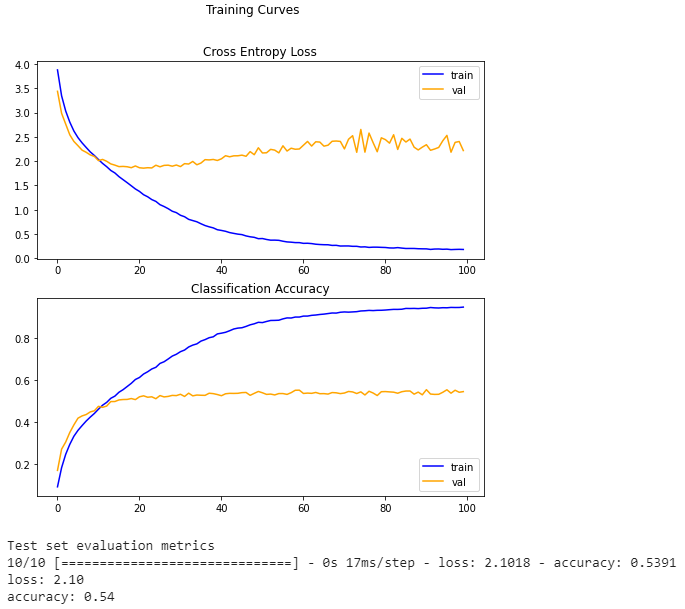
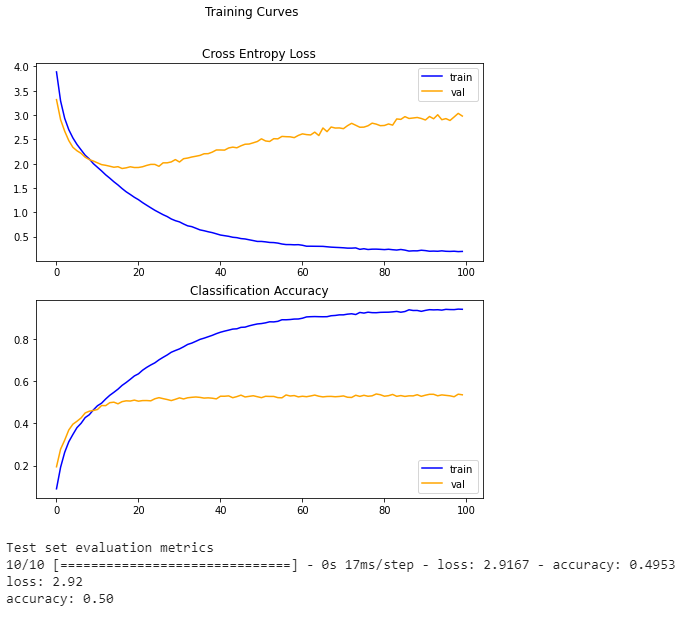
Τέλος, αξίζει να σημειώσουμε ότι στο πέρας κάθε κεντρικού επιπέδου τοποθετούμε ένα επίπεδο *Dropout*, σκοπός του οποίου είναι η πρόληψη της υπερ-εκπαίδευσης του μοντέλου μας, μέσω της τυχαίας ρήξης νευρώνων των κρυμμένων επιπέδων, καθώς και η επιτάχυνση της εκπαίδευσης. Οι τιμές που περνάμε σαν παραμέτρους στα αντίστοιχα επίπεδα αντιστοι-χούν στο ποσοστό των νευρώνων που πρόκειται να αγνοηθούν προς το επόμενο επίπεδο. Σύμφωνα με την επιστημονική βιβλιογραφία[1], τυπικές τιμές *dropout* ανήκουν στο διάστημα [0.5, 0.8]. Για αυτό το σκοπό επιλέξαμε τη σταδιακή αύξηση της συχνότητας ρήξης, αφού στο συγκεκριμένο πρόβλημα ταξινόμησης εικόνων, μεγαλύτερες τιμές θα ήταν αντιπαραγωγικές για τη γενίκευση του μοντέλου.

Στη συνέχεια παρουσιάζεται κομμάτι του κώδικα που αντιστοιχεί στην τελική αρχιτεκτονική του δικτύου μας:

Εικόνα 1: Αρχιτεκτονική Συνελικτικού Δικτύου from-scratch

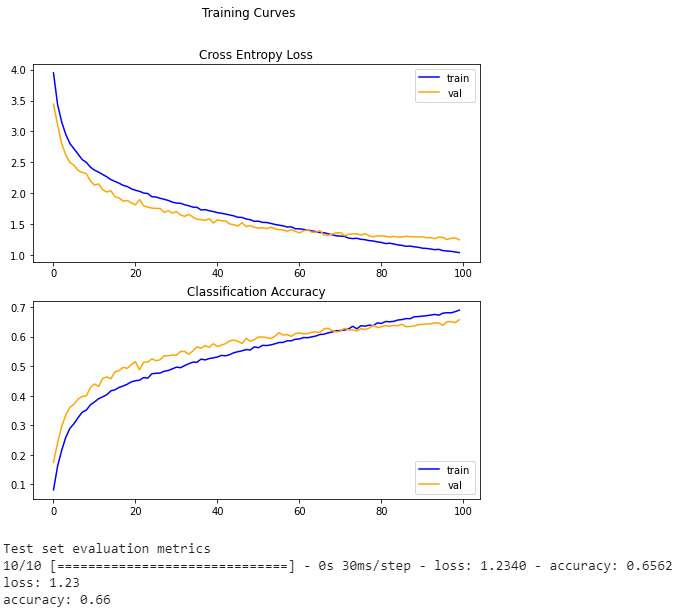
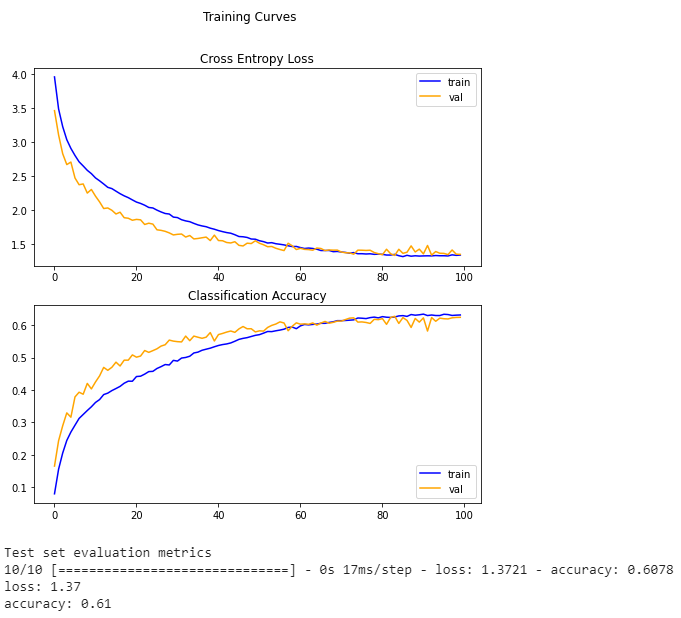
### **Αλγόριθμος Βελτιστοποίησης - Optimizer**

Σχετικά με τον αλγόριθμο βελτιστοποίησης εξετάσαμε εκείνους του *Adam* και του *RMSprop*. Αν και αναμέναμε ο πρώτος αλγόριθμος να είναι αποδοτικότερος[2],παρατηρήσαμε ότι κατά την απλή εκπαίδευση του δικτύου μας για 100 εποχές, ο αλγόριθμος *RMSprop* εμφάνισε ελα-φρώς καλύτερα αποτελέσματα, γεγονός που απεικονίζεται στα παρακάτω διαγράμματα:



Εικόνα 2: Σύγκριση αλγορίθμων RMSprop(αριστερά) και Adam(δεξιά) σε απλή εκπαίδευση

Ωστόσο αξίζει να παρατηρήσουμε, ότι κατά την εκπαίδευση του δικτύου μας με τεχνική da-ta-augmentation για 100 εποχές, ο αλγόριθμος Adam εμφάνισε, όχι μόνο καλύτερη απόδο-ση, αλλά και την τάση περαιτέρω βελτίωσης.

****

Εικόνα 3: Σύγκριση αλγορίθμων RMSprop(αριστερά) και Adam(δεξιά) σε Data-Augmented εκπαίδευση

Ορμώμενοι από την παραπάνω παρατήρηση επιλέξαμε τον αλγόριθμο *Adam* για το τελικό μας μοντέλο.

### **Συνάρτηση Ενεργοποίησης – Activation Function**

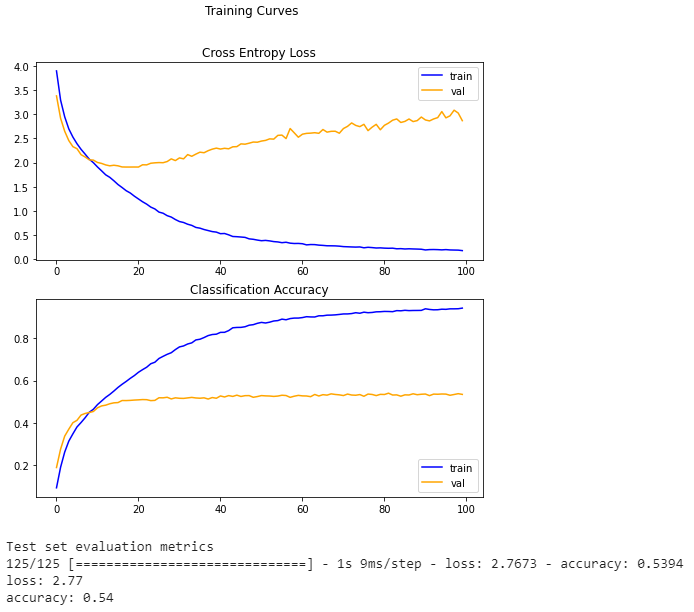
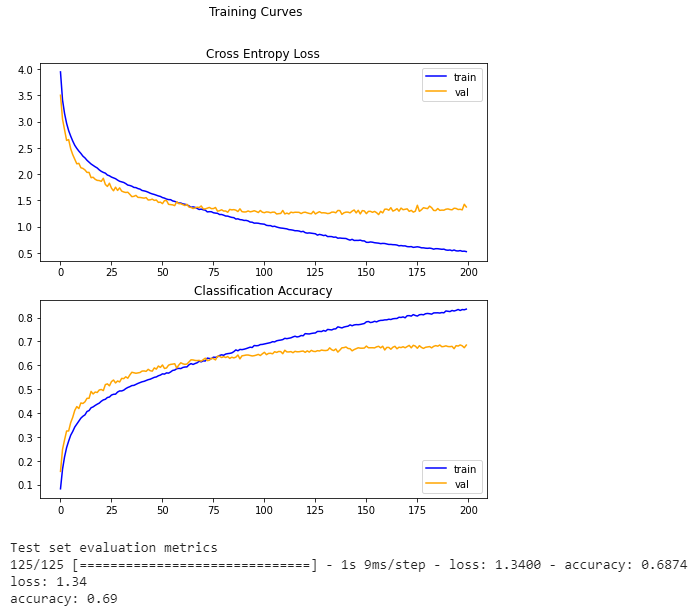
Ως συνάρτηση ενεργοποίησης χρησιμοποιήσαμε την *Exponential Linear Unit(ELU)*. Επιλέξαμε αυτήν την επιλογή, καθώς έπειτα από πρόχειρες δοκιμές του μοντέλου μας με συνάρτηση ε-νεργοποίησης τη *Rectified Linear Unit(ReLU)* παρατηρήσαμε έντονες ταλαντώσεις στη γραφι-κή παράσταση ακρίβειας του μοντέλου μας. Μελετώντας, λοιπόν, τις διάφορες διαθέσιμες εναλλακτικές και λαμβάνοντας υπόψιν τις αντίστοιχες επιλογές της επιστημονικής κοινότη-τας σήμερα, καταλήξαμε στην ELU. Σε κάθε περίπτωση σημειώνουμε, ωστόσο, ότι δεν πειρα-ματιστήκαμε εξαντλητικά με τη συνάρτηση ενεργοποίησης, όπως λόχου χάρη με τον αλγό-ριθμο βελτιστοποίησης.

### **Batch Size**

To μέγεθος του batch το διατηρήσαμε σταθερό καθ’ όλη τη διάρκεια της εκπαίδευσης, επι-λέγοντας την τιμή του 64. Σε αυτή την επιλογή μας συνέβαλαν επιστημονικές δημοσιεύσεις[3] οι οποίες προτείνουν μικρό μέγεθος batch, στοιχείο το οποίο εξασφαλίζει σταθερότητα κατά τη διαδικασία της εκπαίδευσης και προωθεί την τελική γενίκευση του μοντέλου.

### **Data Augmentation**

Το μοντέλο μας επιχειρήσαμε να το εκπαιδεύσουμε τόσο στο σύνολο δεδομένων μας, όσο και στο επαυξημένο σύνολο δεδομένων, τροφοδοτώντας το τυχαία με επεξεργασμένες ει-κόνες εισόδου. Πιο συγκεκριμένα, οι μετασχηματισμοί που επιλέξαμε περιέχουν την οριζό-ντια περιστροφή καθώς και κατακόρυφη ή οριζόντια μετατόπιση κατά το 10% του μεγέθους της εικόνας.

Στη συνέχεια παρουσιάζονται διαγράμματα που αντικατοπτρίζουν την επίδραση της παρα-πάνω τεχνικής, όταν το μοντέλο μας εκπαιδεύτηκε με ή χωρίς αυτή, για 100 και 200 εποχές αντίστοιχα. Όπως σημειώσαμε και παραπάνω, επιλέξαμε το διάστημα των 200 εποχών, κα-θώς παρατηρήσαμε την τάση περαιτέρω βελτίωσης του μοντέλου. Παρατηρούμε ότι, από τη μία πλευρά βελτιώθηκε η απόδοση του δικτύου, ενώ από την άλλη αντιμετωπίσαμε επιτυ-χώς και το πρόβλημα υπερ-εκπαίδευσης, γεγονός το οποίο γίνεται αντιληπτό από τη μεγάλη διαφορά μεταξύ της καμπύλης του *training* και του *validation*.

Εικόνα 4: Εκπαίδευση δικτύου χωρίς(αριστερά) και με Data-Augmentation(δεξιά)

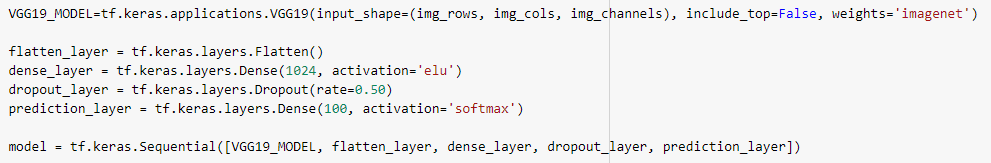
### **Αποτελέσματα – Τελικές Σκέψεις και Παρατηρήσεις**

Ακολουθώντας την αρχιτεκτονική που παρουσιάστηκε παραπάνω και κάνοντας χρήση των ήδη αναφερθέντων μεθόδων, η αρχική απόδοση του μοντέλου μας στα δεδομένα ελέγχου ανέρχεται σε 54% ακρίβεια και με το δείκτη απώλειας ίσο με 2.27. Εμπλουτίζοντας τα δεδο-μένα μας με *augmented-images*, επιτυγχάνουμε μία βελτίωση της ακρίβειας του μοντέλου της τάξης του 15% περίπου. Πιο συγκεκριμένα η τελική ακρίβεια του δικτύου στα δεδομένα ελέγχου ανέρχεται στο 69%, ενώ ο δείκτης απώλειας μειώθηκε στο 1.34.

Η πρώτη εκπαίδευση του μοντέλου διήρκησε περίπου 20 λεπτά, ενώ η δεύτερη 3 φορές πε-ρισσότερο, αγγίζοντας τα 60 λεπτά. Δεδομένης της βελτίωσης της ακρίβειας του δικτύου μας, καθώς και της περαιτέρω πτώσης του δείκτη απώλειας, βρίσκουμε την αύξηση της χρονικής διάρκειας λογική και η όλη διαδικασία μας αφήνει ικανοποιημένους.

Αξίζει να παρατηρήσουμε, ότι τα δεδομένα μας χαρακτηρίζονται από 80 συνολικά εικόνες. Με μικρότερο αριθμό κλάσεων αναμένουμε, αφενός ταχύτερους χρόνους εκπαίδευσης, λό-γω του μικρότερου όγκου δειγμάτων και αφετέρου την αισθητή βελτίωση της ακρίβειας του μοντέλου.

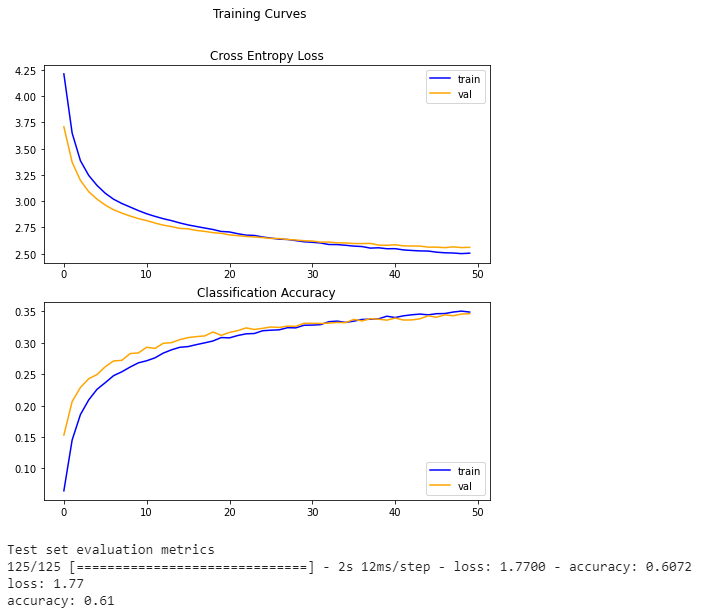
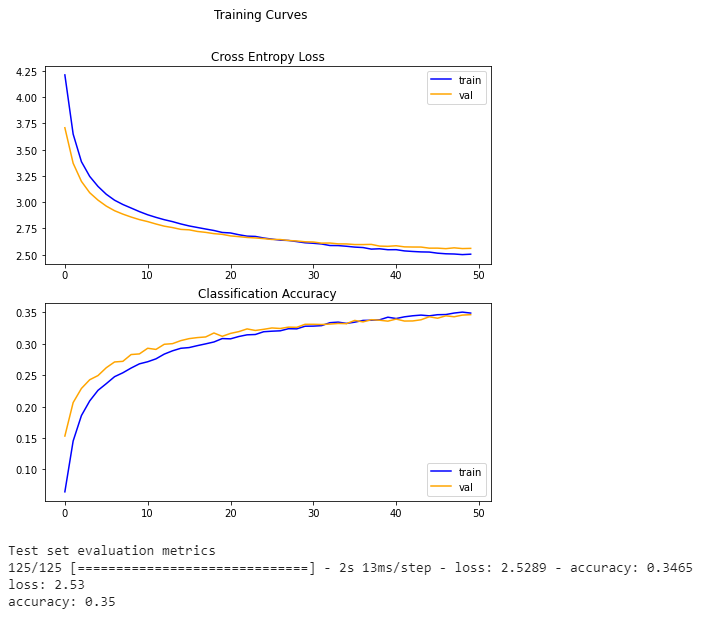
## **Μεταφορά Μάθησης – Transfer Learning**

Για τη μέθοδο της μεταφοράς μάθησης θα χρησιμοποιήσουμε το προ-εκπαιδευμένο δίκτυο VGG19. Σημειώνουμε ότι θα αγνοήσουμε το στρώμα πρόβλεψης του δικτύου, προκειμένου να το προσαρμόσουμε στο συγκεκριμένο πρόβλημα που θέλουμε να λύσουμε. Πιο συγκεκρι-μένα, θα προσθέσουμε ένα πλήρως συνδεδεμένο στρώμα νευρώνων, από το οποίο θα αγνο-ήσουμε στη συνέχεια, τυχαία, τους μισούς. Ολοκληρώνουμε την αρχιτεκτονική με ακόμα ένα πλήρως συνδεδεμένο στρώμα, με πλήθος νευρώνων ίσο με το πλήθος των κλάσεων του προ-βλήματος. Ο αντίστοιχος κώδικας παρουσιάζεται παρακάτω:

Εικόνα 5: Συμπληρωματική αρχιτεκτονική προ-εκπαιδευμένου VGG19

Στη συνέχεια θα εξετάσουμε πως επιδρά η πλήρης ή μερική εκπαίδευση του δικτύου στην απόδοση της ταξινόμησης για το σύνολο δεδομένων μας.

### **Trainable vs Non-Trainable**

Στην πρώτο μέθοδο που ακολουθήσαμε “παγώσαμε” τα προ-εκπαιδευμένα επίπεδα στο ImageNet και εκπαιδεύσαμε μονάχα τα τελευταία στρώματα που προσθέσαμε, εκείνα δηλαδή που εξειδικεύ-ονται στο πρόβλημά μας. Αντίθετα, κατά τη δεύτερη μέθοδο εκπαιδεύσαμε όλο το δίκτυο, επιλέγο-ντας συντελεστή μάθησης αρκετά μικρό, έτσι ώστε το σύνολο των δεδομένων μας να μην υπερισχύ-σει του πολύ μεγαλύτερο ImageNet. Σημειώνουμε ότι και για τις δύο μεθόδους συμπεριλάβαμε τεχνι-κή data-augmentation, αυτούσια όπως παρουσιάστηκε παραπάνω. Ακολουθούν τα αντίστοιχα δια-γράμματα απόδοσης:

Εικόνα 6: Trainable vs Non-trainable

### **Αποτελέσματα – Τελικές Σκέψεις και Παρατηρήσεις**

Τα τελικά αποτελέσματα των μοντέλων μας διαμορφώνονται ως εξής: 35% ακρίβεια και 2.53 δείκτη απώλειας για τη μερική εκπαίδευση του VGG19, έναντι 61% ακρίβειας και 1.77 δείκτη απώλειας για την πλήρη εκπαίδευσή του.

Λαμβάνοντας υπόψιν ότι η εκπαίδευση της πρώτης μεθόδου διήρκησε 50 εποχές, ενώ εκεί-νη της δεύτερης 25, τα αποτελέσματα τα βρίσκουμε λογικά και ικανοποιητικά. Αξίζει να ση-μειώσουμε ότι η εκπαίδευση με “παγωμένα” τα συνελικτικά επίπεδα του VGG19 χρειάστηκε κάτι περισσότερο από 15 λεπτά, δηλαδή υποδιπλάσιος χρόνος από εκείνη που θα χρειαζόταν η πλήρης εκπαίδευση του δικτύου για το ίδιο πλήθος εποχών. Κάτι τέτοιο είναι αναμενόμε-νο, εάν λάβουμε υπόψιν το πλήθος των μεταβλητών προς εκπαίδευση για τα δύο μοντέλα, τιμές οι οποίες ανέρχονται σε 627.812 και 20.652.196, αντίστοιχα.

Αξίζει να παρατηρήσουμε ότι το πρώτο μοντέλο εμφανίζει την τάση για περαιτέρω βελτίωση, σε αντίθεση με το δεύτερο, στο οποίο μπορούμε να διακρίνουμε το φαινόμενο της υπερ-εκπαίδευσης του δικτύου, παρά το μικρό πλήθος εποχών. Κάτι τέτοιο είναι αναμενόμενο, καθώς το πρώτο δίκτυο έχει πολύ λιγότερα περιθώρια να υπερ-προσαρμοστεί στις εικόνες που του τροφοδοτούμε λόγω των “παγωμένων” επιπέδων. Συνοπτικά, λοιπόν, παρά τη σχε-τικά κακή απόδοση το πρώτο μοντέλο εμφανίζει ενθαρρυντικά αποτελέσματα για περισσό-τερο *fine-tuning*.

# **Αναφορές**

[1]. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov - *Dropout: A Simple Way to Prevent Neural Networks from Overfitting* – 2014

[2]. Sebastian Ruder - *An Overview of Gradient Descent Optimization Algorithms* - 2016

[3]. Dominic Masters, Carlo Luschi - *Revisiting Small Batch Training for Deep Neural Networks* - 2018