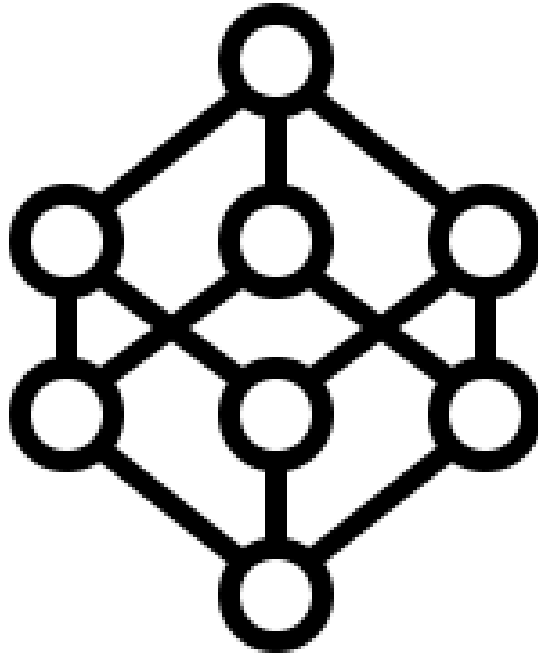


Εργασία CIFAR100 - Team 4



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Διατμηματικό ΠΜΣ στα Προηγμένα Συστήματα Υπολογιστών και
Επικοινωνιών
Υπολογιστή Νοημοσύνη - Συστήματα Εμπνευσμένα από τη Βιολογία

Βαγενάς Θεόδωρος - Παναγιώτης, AM 410
Πετρόπουλος Αλέξανδρος, AM 440
Τσίντζηρα Αγγελική - Αγαθή, AM 436

Θεσσαλονίκη Φεβρουάριος 2019

Ερώτημα Α

Για την υλοποίηση του πρώτου ερωτήματος χρησιμοποιήσαμε δύο από τα ήδη εκπαιδευμένα δίκτυα. Το πρώτο είναι το VGG16 και το δεύτερο το ResNet. Τα δύο αυτά μοντέλα είναι ήδη εκπαιδευμένα με βάρη για το σύνολο δεδομένων ImageNet. Και τα δύο μοντέλα χρησιμοποιούνται με παρόμοιο τρόπο και ο σκοπός μας είναι να τα παρατηρήσουμε και να εξασκηθούμε στο πως δουλεύουν τα νευρωνικά δίκτυα. Για την εκτέλεση τους χρησιμοποιήσαμε 56 τυχαίες φωτογραφίες από το διαγωνισμό ImageNet, μεγέθους 224x224.

Αρχικά, κατεβάσαμε τις εικόνες σε έναν κοινό φάκελο και στη συνέχεια αφού τις φορτώσαμε στην μνήμη, κάναμε μία μικρή προεπεξεργασία προκειμένου να είναι στην κατάλληλη μορφή για την εκτέλεση των μοντέλων. Ακολούθως, διαβάσαμε τα βάρη για τα μοντέλα και για κάθε ένα εκτελέσαμε τη συνάρτηση πρόβλεψης (predict()). Η συνάρτηση predict μας επιστρέφει διάφορα χαρακτηριστικά τα οποία ορίζουμε ως labels (ετικέτες) και εκτυπώνουμε για κάθε εικόνα δύο ετικέτες, αυτές που προβλέφθηκαν με το μεγαλύτερο ποσοστό. Στην Εικόνα 1 φαίνεται ένα στιγμιότυπο της παραπάνω εκτύπωσης.

```
for j in range(len(label)):
    temp = label[j][0]
    print('Picture %i is ' % (j))
    for i in range(2):
        print('%s (%.2f%%)' % (temp[i][1], temp[i][2]*100), end = '')
        if(i==1):
            print("\n")
```

```
Picture 0 is
swimming_trunks (27.85%) wool (20.77%)

Picture 1 is
pot (38.83%) coral_fungus (18.08%)

Picture 2 is
greenhouse (62.13%) maze (9.25%)

Picture 3 is
barn (64.68%) worm_fence (12.61%)

Picture 4 is
picket_fence (15.64%) bicycle-built-for-two (14.45%)

Picture 5 is
sea_slug (41.45%) rock_beauty (19.64%)

Picture 6 is
sliding_door (72.55%) window_shade (11.18%)

Picture 7 is
scale (71.67%) combination_lock (5.10%)
```

Εικόνα 1 - Ενδεικτικά αποτελέσματα εκτέλεσης VGG16 ImageNet

Στην Εικόνα 2 φαίνονται ενδεικτικά οι προβλέψεις δύο εικόνων, όπου παρατηρούμε ότι το δίκτυο VGG16 πρόβλεψε σωστά τα αντικείμενα της πρώτης εικόνας ενώ στη δεύτερη δεν τα πήγε καλά. Η ίδια ακριβώς διαδικασία υλοποιήθηκε και για το νευρωνικό δίκτυο ResNet. Στην Εικόνα 3 φαίνονται ενδεικτικά τα αποτελέσματα πρόβλεψης των δύο παραπάνω εικόνων για το δίκτυο ResNet το οποίο δεν τα πήγε εξίσου καλά με το πρώτο, ωστόσο δεν μπορούμε να βγάλουμε συμπεράσματα ελέγχοντας την απόδοση μόνο δύο εικόνων.



Εικόνα 2 - Αποτελέσματα εκτέλεσης VGG16 ImageNet



Εικόνα 3 - Αποτελέσματα εκτέλεσης ResNet ImageNet

Ερώτημα Β

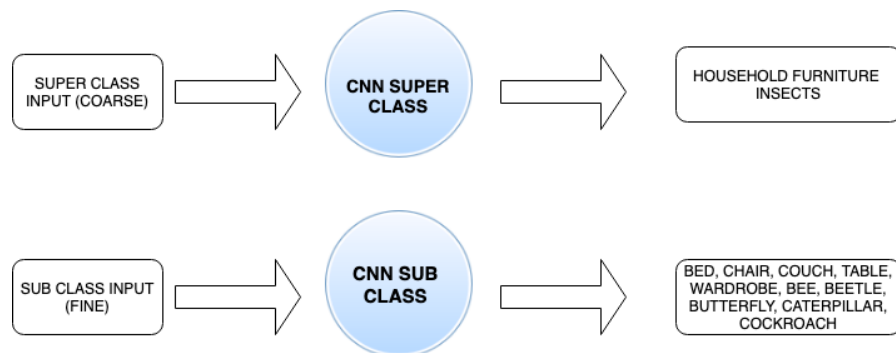
Περίληψη

Το νευρωνικό δίκτυο που κατασκευάσαμε έχει ως σκοπό να ταξινομήσει φωτογραφίες 10 κλάσεων οι οποίες ανά πέντε ανατίθενται σε δύο υπερ κλάσεις. Οι δύο υπερ κλάσεις ανήκουν στην ομάδα 4 δηλαδή, είναι τα έπιπλα και τα έντομα. Οι υποκλάσεις των επίπλων είναι το κρεβάτι, η καρέκλα, ο καναπές, το τραπέζι και η ντουλάπα ενώ οι υποκλάσεις των εντόμων είναι η μέλισσα, το σκαθάρι, η πεταλούδα, η κάμπια και η κατσαρίδα. Η δομή του συνόλου δεδομένων που μας δόθηκε αποτελείται από τις διαστάσεις των έγχρωμων εικόνων (32x32x3) και τις κλάσεις στις οποίες ανήκουν. Κάθε φωτογραφία ανήκει σε μία υπο κλάση και σε μία υπερ κλάση.

Τα στοιχεία που χαρακτηρίζουν τα έντομα από τα έπιπλα είναι ιδιαίτερα διακριτά οπότε, το συνελικτικό δίκτυο που θα υλοποιήσουμε αναμένουμε να έχει επίδοση άνω του 90% ως προς τον διαχωρισμό μεταξύ των υπερ κλάσεων. Οι υποκλάσεις όμως είναι ιδιαίτερα όμοιες, παραδείγματος χάριν ένα τραπέζι με ένα κρεβάτι έχουν πολλά κοινά χαρακτηριστικά (4 πόδια, επίπεδο που ενώνει τα 4 πόδια) ή το σκαθάρι με την κατσαρίδα έχουν επίσης παρόμοια δομή. Επομένως, όσον αφορά το διαχωρισμό μεταξύ των υποκλάσεων περιμένουμε χαμηλότερη επίδοση λόγω της δυσκολίας διαχωρισμού των

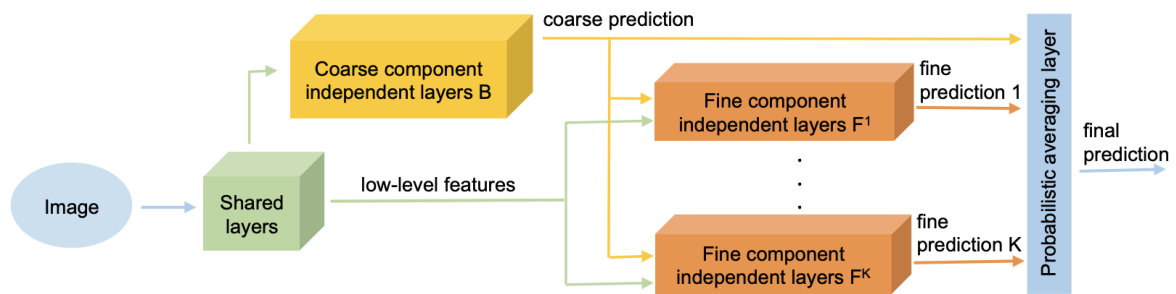
κοινών χαρακτηριστικών. Προκειμένου να πετύχουμε μία καλύτερη γενίκευση και το μοντέλο μας να πετύχει καλύτερη απόδοση χρησιμοποιήσαμε τεχνικές Data Augmentation οι οποίες μας βοηθούν να παράγουμε νέες εικόνες από τις ήδη υπάρχουσες εφαρμόζοντας διάφορους μετασχηματισμούς όπως περιστροφή ή αντανάκλαση. Με αυτή την τεχνική καταφέρνουμε να αυξήσουμε το μέγεθος του συνόλου δεδομένων και την ποικιλομορφία του με αποτέλεσμα το μοντέλο μας να γενικεύει καλύτερα και να βελτιώνεται η απόδοσή του. Επίσης, εφαρμόσαμε κανονικοποίηση των δεδομένων καθώς ο τρόπος που ανανεώνονται τα βάρη επηρεάζεται από τις αρχικές τιμές. Αν το εύρος είναι διαφορετικό, λανθασμένα θα δοθεί υψηλότερη βαρύτητα σε χαρακτηριστικά με μεγαλύτερες τιμές, οπότε κανονικοποιώντας όλες τις τιμές στο ίδιο εύρος αποφεύγουμε την λανθασμένη αρχικοποίηση - παραγωγή βαρών.

Όπως αναφέρθηκε παραπάνω για κάθε κατηγοριοποίηση (μεταξύ υπερ κλάσεων και υπό κλάσεων) αναμένουμε άλλα αποτελέσματα. Επομένως, επιλέξαμε να δοκιμάσουμε διάφορα μοντέλα για να μελετήσουμε αυτή μας την παρατήρηση. Πιο συγκεκριμένα, υλοποιήσαμε 2 μοντέλα. Το πρώτο μοντέλο φαίνεται στην Εικόνα 4. Δημιουργήσαμε δύο ξεχωριστά συνελκτικά μοντέλα, ένα για τις υπερ κλάσεις και ένα για τις υπό κλάσεις. Τα μοντέλα αυτά είναι αυτόνομα και ανεξάρτητα μεταξύ τους. Δέχονται ως είσοδο το καθένα τα κατάλληλα δεδομένα αναλόγως αν είναι υπο κλάση ή υπερ κλάση και κάνουν τις προβλέψεις τους αντίστοιχα. Παρατηρήσαμε ότι η ακρίβεια του παραπάνω μοντέλου για τις υπερ κλάσεις ήταν ιδιαίτερα καλή ενώ για τις υπό κλάσεις ήταν εμφανώς μειωμένη. Οπότε, σκεφτήκαμε να ελέγξουμε τι επιρροή έχει η ιεραρχική κατηγοριοποίηση των δεδομένων και αν θα βελτιώνει τα αποτελέσματά μας. Το δεύτερο μοντέλο φαίνεται στην Εικόνα 5 .



Εικόνα 4 : Δύο ανεξάρτητα συνελκτικά νευρωνικά δίκτυα

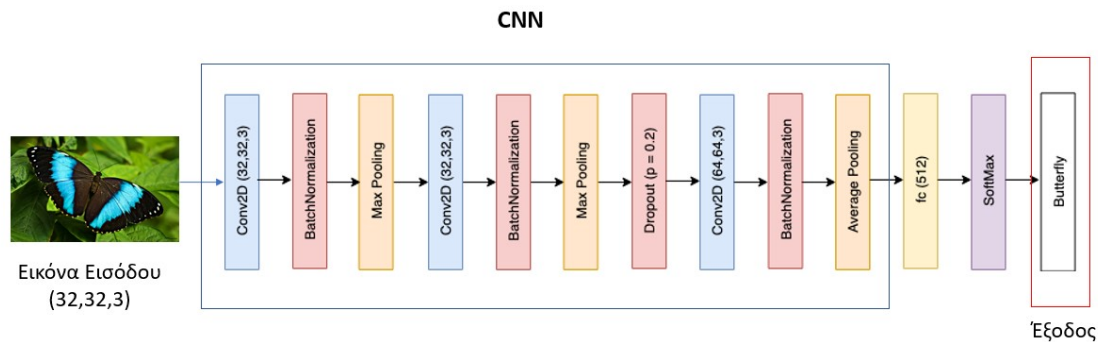
Το ιεραρχικό μοντέλο [1] προσπαθεί να αποσυνθέσει τη διαδικασία της ταξινόμησης σε δύο βήματα. Αρχικά χρησιμοποιείται ένα Hierarchical Deep Convolutional Neural Network (HD-CNN) με σκοπό να κάνει διαχωρισμό μεταξύ των υπερ κλάσεων. Στη συνέχεια, τα δεδομένα δρομολογούνται σε επόμενο επίπεδο δικτύων τα οποία προσπαθούν να κάνουν διαχωρισμό σε επίπεδο υπό κλάσεων. Υιοθετήθηκε μία σχεδιαστική δομή και κάθε CNN χτίστηκε πάνω σε ένα “δομικό” CNN (shared layers). Το “δομικό” CNN είναι μία απλοποίηση ενός από τα top ranked single CNNs. Τα ιεραρχικά CNN μπορούν να επωφεληθούν από την πρόοδο του “δομικού” CNN (Transfer Learning). Το HD-CNN ακολουθεί την κατηγοριοποίηση υπερ κλάσης σε υπό κλάση και ενσωματώνει τις προβλέψεις αυτές για να καταλήξει στην τελική του πρόβλεψη. Ο στόχος μας, από τη δημιουργία μιας ιεραρχικής δομής είναι να ομαδοποιήσουμε τις μη εύκολα διαχωρίσιμες υπό κατηγορίες σε υπερ κατηγορίες για τις οποίες ένα ταξινομητής θα εκπαιδευτεί. Χρησιμοποιούμε μια προσέγγιση από την κορυφή προς τα κάτω για να μάθουμε την ιεραρχία από τα δεδομένα εκπαίδευσης.



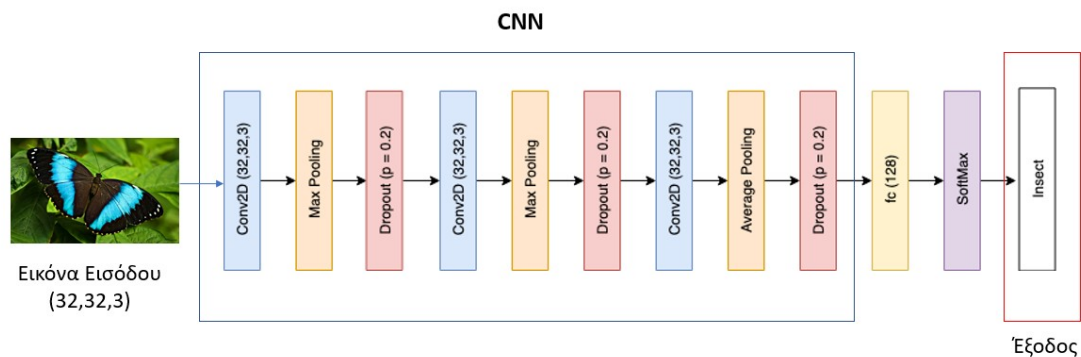
Εικόνα 5 : Ιεραρχική αρχιτεκτονική συνελληνικών νευρωνικών δικτύων (HD-CNN)

Υλοποίηση

Για την υλοποίηση του πρώτου μοντέλου (Εικόνα 4) δημιουργήσαμε δύο ανεξάρτητα συνελκτικά δίκτυα, ένα για τις υπό κλάσεις και ένα για τις υπέρ κλάσεις τα οποία φαίνονται στις Εικόνες 6 και 7 αντίστοιχα.



Εικόνα 6 : Αρχιτεκτονική συνελκτικού δικτύου ταξινόμησης υπό κλάσεων



Εικόνα 7 : Αρχιτεκτονική συνελκτικού δικτύου ταξινόμησης υπέρ κλάσεων

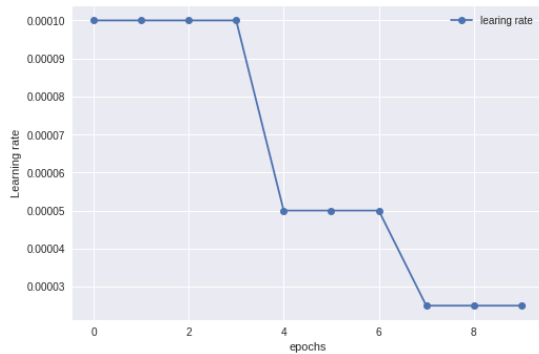
Όπως βλέπουμε στην Εικόνα 6, ως είσοδο στο δίκτυο δίνονται εικόνες μεγέθους 32x32x3. Αυτές περνούν αρχικά από ένα 2D συνελκτικό layer με 32 φίλτρα καθένα από τα οποία έχει μέγεθος 3x3 (το βάθος των φίλτρων είναι ίδιο με τον αριθμό των channels) το οποίο ορίζει και το receptive field (το κομμάτι της εικόνας που είναι ορατό στο φίλτρο). Με την εφαρμογή του κάθε φίλτρου ολισθαίνοντας το πάνω στην εικόνα (συνέλιξη εικόνας

φίλτρου) προκύπτει ένας πίνακας $30 \times 30 \times 1$ (feature map - έξοδος) διότι έχουμε $\text{stride}=1$ και $\text{padding}=0$. Επειδή έχουμε 32 φίλτρα έχουμε 32 feature maps. Έπειτα, εφαρμόζεται Batch Normalization (κανονικοποιούνται οι έξοδοι του προηγούμενου layer σε κάθε batch για $\text{mean}=0$ και $\text{std}=1$) καθώς έτσι γίνεται πιο σταθερή η εκπαίδευση και μειώνεται ο χρόνος εκπαίδευσης. Ως συνάρτηση ενεργοποίησης έχει χρησιμοποιηθεί η “elu” η οποία είναι ίδια με την Relu με μόνη διαφορά τις αρνητικές εισόδους. Η elu συγκλίνει πιο γρήγορα και έχει μικρότερο σφάλμα ακρίβειας. Έπειτα, εφαρμόζεται ένα layer Max-pooling 2D (2×2) για τη μείωση των διαστάσεων των feature maps και ακολούθως των παραμέτρων ώστε να αποφευχθεί το overfitting και να γίνει πιο απλό το δίκτυο (μια μικρή αλλαγή στην εικόνα δεν θα επηρεάσει το δίκτυο καθώς διατηρείται το μέγιστο της τοπικής περιοχής). Η έξοδος του επιπέδου αυτού είναι $(15 \times 15 \times 32)$. Ακολουθεί το 2ο συνελικτικό layer με τις ίδιες παραμέτρους όπως και το 1ο. Μετά τη συνέλιξη η έξοδος μειώνεται λίγο ακόμη $(13 \times 13 \times 32)$. Εφαρμόζεται batch normalization με συνάρτηση ενεργοποίησης “elu”. Μετά τη συνέλιξη γίνεται και πάλι Max-pooling 2D (2×2) για την περαιτέρω μείωση των διαστάσεων δίνοντας έξοδο $(6 \times 6 \times 32)$. Αυτή την φορά προστίθεται ένα layer για dropout με το οποίο τυχαία επιλεγμένοι νευρώνες αγνοούνται κατά τη διάρκεια της εκπαίδευσης με πιθανότητα 0.2. Έτσι το μοντέλο γενικεύει καλύτερα και αποφεύγεται η υπερεκπαίδευση. Ακολουθεί το 3ο συνελικτικό layer αυτή την φορά με 64 φίλτρα, μεγέθους 3×3 και $\text{stride}=1$ χωρίς padding, μειώνοντας έτσι το μέγεθος του feature map και δίνοντας ως έξοδο $(4 \times 4 \times 64)$. Σε αυτό το συνελικτικό αυξήθηκε ο αριθμός των φίλτρων με σκοπό να μη χαθεί πολύ μεγάλη πληροφορία από την αρχική εικόνα προκειμένου να υπάρξει μια εξισορρόπηση στους υπολογισμούς καθώς το μέγεθος από κάθε feature map έχει μειωθεί αισθητά σε σχέση με το αρχικό. Ακολούθησε η εφαρμογή των batch normalization και ενός Averaging-pooling (κρατείται ο μέσος όρος για μικρή βελτίωση του αποτελέσματος). Με το layer flatten μετατρέπονται οι 3 διαστάσεις σε ένα μονοδιάστατο διάνυσμα για να εισαχθεί στο fully-connected layer που ακολουθεί. Για το fully-connected layer έχουμε 512 νευρώνες (αριθμό μεγαλύτερο από τα χαρακτηριστικά των προηγούμενων layer). Τέλος, ακολουθεί ένα fully-connected layer με 10 εξόδους (όσες και οι υπό κλάσεις) και η συνάρτηση ενεργοποίησης softmax η οποία κανονικοποιεί τις εξόδους στο διάστημα 0-1 για να προκύψουν οι πιθανότητες για κάθε κλάση.

Αντίστοιχα, στην Εικόνα 7 παρουσιάζεται το συνελικτικό δίκτυο για την αναγνώριση των 2 υπέρ κλάσεων. Και σε αυτό το μοντέλο έχουμε 3 συνελικτικά επίπεδα. Αρχικά, εφαρμόζεται ένα layer συνέλιξης με 32 φίλτρα μεγέθους (3×3) και σε αυτό η συνάρτηση ενεργοποίησης είναι η “elu”. Ακολουθεί ένα layer Max-pooling 2d μεγέθους (2×2) και ένα layer dropout από το πρώτο κιάλας επίπεδο για να μειωθεί η υπερεκπαίδευση του μοντέλου η οποία παρατηρήθηκε από τις πρώτες εκτελέσεις. Το νευρωνικό αποτελείται από άλλα 2 συνελικτικά επίπεδα ίδιων παραμέτρων με του 1ου. Το 2ο επίπεδο είναι ακριβώς ίδιο με το 1ο ενώ η μόνη διαφοροποίηση του 3ου είναι η χρήση Average-pooling. Έπειτα, έχουμε τη μετατροπή σε μονοδιάστατο διάνυσμα μέσω του Flatten και 2 fully-connected layers, το 1ο με 128 νευρώνες και το τελευταίο με συνάρτηση ενεργοποίησης softmax και 2 εξόδους για τις υπό κλάσεις.

Η επιλογή των παραμέτρων έγινε σύμφωνα με το πρόβλημα που έχουμε να επιλύσουμε, τη θεωρία και φυσικά με δοκιμές. Για το **Kernel Shape** δοκιμάστηκαν μεγέθη 3×3 και 5×5 με καλύτερα αποτελέσματα το 3×3 . Σκοπός είναι να επιτευχθεί το κατάλληλο επίπεδο granularity ώστε τα αφαιρετικά χαρακτηριστικά να είναι στη σωστή κλίμακα. Αφού οι εικόνες μας είναι αρκετά μικρές (32×32) επιλέχθηκε και μικρό μέγεθος kernel. Για το CNN μας έχουμε υλοποιήσει adaptive learning rate (για υπερ κλάσεις ξεκινά από 0.0001 ενώ για υπο

κλάσεις από 0.001), όπου όταν δεν παρατηρηθεί βελτίωση στην απώλεια του συνόλου ελέγχου μεγαλύτερη από 0.05 για 3 εποχές, τότε μειώνεται στο μισό το **Learning Rate**. Χαρακτηριστικά δίνεται το εξής διάγραμμα παρακάτω. Οι τιμές που επιλέξαμε ως αρχικές για

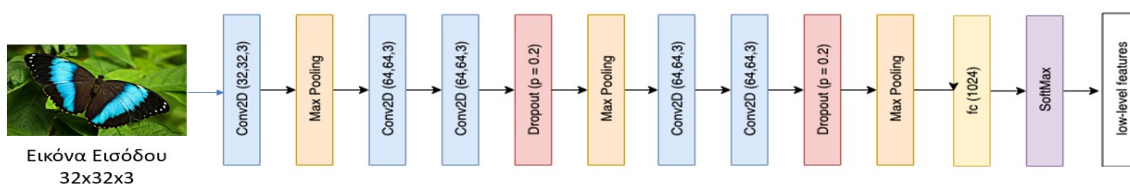


learning rate είναι αυτές που προτείνονται από το documentation του optimizer Adam και επιλέχθηκε μικρότερη τιμή για τις υπό κλάσεις καθώς είναι πιο δύσκολος ο διαχωρισμός τους.

Δοκιμάστηκαν αρκετοί optimizers και επιλέχθηκε ο **Adam (Adaptive moment estimation)** ο οποίος είναι απλός στη χρήση και χρησιμοποιεί momentum δηλαδή, λαμβάνει υπόψη του προηγούμενα gradients για να ομαλοποιήσει τα βήματα του gradient descent. Όσον αφορά την **προεπεξεργασία των δεδομένων** έγινε

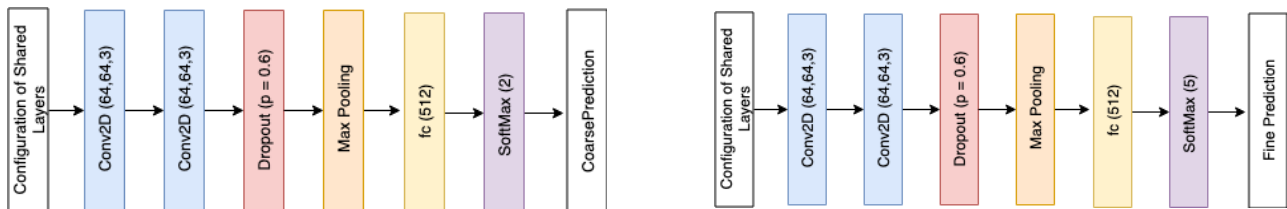
κανονικοποίηση των εικόνων στο διάστημα 0-1 με τη διαίρεση /255 η οποία είναι και η μέγιστη τιμή που μπορούν να πάρουν. Επίσης, υλοποιήσαμε **Data augmentation** το οποίο βοήθησε στη βελτίωση της ακρίβειας αλλά και της ικανότητας γενίκευσης του νευρωνικού. Εφαρμόστηκε horizontal_flip καθώς για τις εικόνες των επιπλων και εντόμων που είχαμε να ταξινομήσουμε είχε νόημα. Επίσης, εφαρμόστηκαν τυχαίες περιστροφές σε εύρος 10 μοιρών, τυχαίες μεγεθύνσεις (zoom) και μετατόπιση κατά πλάτος και ύψος. Τέλος, από τα **Callbacks** πιο βασικά είναι το ModelCheckpoint το οποίο έχει ρυθμιστεί ώστε να αποθηκεύει το μοντέλο με την μικρότερη val_loss, το EarlyStopping για τη διακοπή της εκπαίδευσης όταν η μετρική που ορίζεται σταματάει να βελτιώνεται (accuracy ορίσαμε εμείς) και το ReduceLROnPlateau το οποίο μειώνει το learning rate όταν το MSE (στο validation) σταματάει να μειώνεται.

Για την υλοποίηση του δεύτερου μοντέλου (Εικόνα 5) αρχικά δημιουργήσαμε ένα δομικό CNN το οποίο φαίνεται στην Εικόνα 8. Αποτελεί μία απλοποίηση ενός ήδη υπάρχων CNN καθώς οι κλάσεις και τα χαρακτηριστικά του προβλήματος μας προς ταξινόμηση είναι πολύ λιγότερα από αυτά για τα οποία φτιάχτηκε το συγκεκριμένο CNN [1]. Το μοντέλο αυτό είναι υπεύθυνο για την εξαγωγή χαρακτηριστικών χαμηλού επιπέδου και οι ρυθμίσεις του χρησιμοποιούνται ως είσοδος για τα υπόλοιπα CNN (Transfer Learning). Με αυτή την αρχιτεκτονική, στο επίπεδο του δομικού CNN εξάγονται χαρακτηριστικά χαμηλού επιπέδου (ακμές και γραμμές) και έτσι τα υπόλοιπα CNN είναι σε θέση να εξάγουν πιο αφαιρετικά χαρακτηριστικά (όπως πρόσωπο ζώου). Καθώς τα χαρακτηριστικά χαμηλού επιπέδου είναι σημαντικά και για τις υπερ και για τις υπό κλάσεις μειώνεται ο χρόνος και η υπολογιστική ισχύς διότι αυτά υπολογίζονται μία φορά στην αρχή και τα υπόλοιπα μοντέλα χρησιμοποιούν ένα προ-εκπαιδευμένο μοντέλο ως είσοδο. Επίσης, με αυτή την αρχιτεκτονική πετυχαίνουμε μείωση των παραμέτρων των HD-CNN, χαρακτηριστικό σημαντικό για την επιτυχία και το χρόνο της εκπαίδευσης.



Εικόνα 8 : Αρχιτεκτονική δομικού συνελκτικού δικτύου - Shared Layers

Στη συνέχεια οι ρυθμίσεις αυτές χρησιμοποιούνται ως είσοδοι για τα τρία επόμενα συνελκτικά δίκτυα, ένα για τις υπερ κλάσεις και δύο για τις υπό κλάσεις των δύο υπερ κλάσεων. Τα τρία αυτά CNN έχουν ίδια δομή η οποία φαίνεται στην Εικόνα 9. Τονίζουμε ότι τόσο οι υπό κλάσεις όσο και οι υπερ κλάσεις επαναχρησιμοποιούν τις διαμορφώσεις από το δομικό στοιχείο CNN (shared layer). Στο αριστερό τμήμα της Εικόνας 9 απεικονίζεται το συνελκτικό δίκτυο για την πρόβλεψη υπερ κλάσεων. Στο δεξιό τμήμα της Εικόνας 9 απεικονίζεται το συνελκτικό δίκτυο για την πρόβλεψη υπό κλάσεων. Η δομή τους είναι η ίδια, αλλάζει μόνο η SoftMax η οποία σε κάθε περίπτωση έχει ως τιμή το πλήθος των εξόδων (κλάσεων).



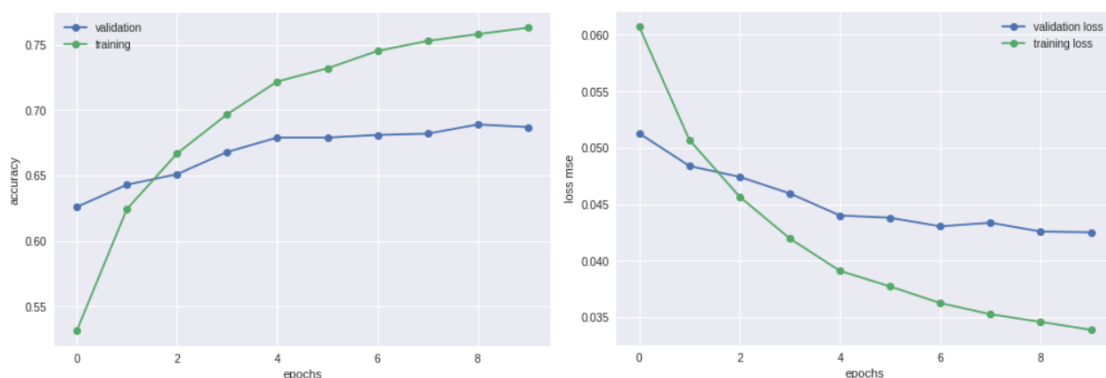
Εικόνα 9 : Αριστερά CNN για υπερ κλάσεις και δεξιά CNN για υπό κλάσεις

Το τελευταίο επίπεδο (Probabilistic Average Layer) που φαίνεται στην Εικόνα 5 συνδυάζει τα αποτελέσματα των προβλέψεων των προηγούμενων CNN και υπολογίζει την σταθμισμένη πρόβλεψη για τις υπό κλάσεις. Η πρόβλεψη των υπερ κλάσεων προέρχεται από το συνελκτικό αριστερά της Εικόνας 9.

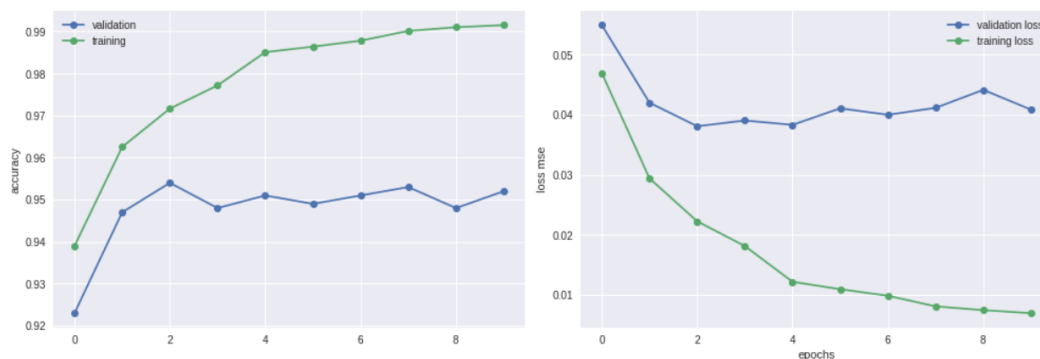
Αποτελέσματα - Συμπεράσματα

Οι εκτελέσεις και δοκιμές των νευρωνικών δικτύων έγιναν στο *Google Colaboratory* καθώς παρείχε GPU Runtime για την επιτάχυνση των εκτελέσεων.

Παρακάτω παρουσιάζονται τα αποτελέσματα του πρώτου μοντέλου (Εικόνα 4). Στην Εικόνα 10 βλέπουμε την ακρίβεια και το MSE κατά τη διάρκεια της εκπαίδευσης ως προς τις εποχές για τις υπό κλάσεις. Το πλήθος των εποχών είναι 10 καθώς έπειτα διακόπηκε με τη διαδικασία *early stopping* αφού δεν άλλαξε αισθητά το αποτέλεσμα στην εκπαίδευση για 7 συνεχόμενες εποχές. Στο σύνολο ελέγχου η ακρίβεια ήταν **71.2%-75.1%**. Στην Εικόνα 11 τα διαγράμματα δείχνουν τα αντίστοιχα αποτελέσματα για τις 2 υπερ κλάσεις όπου η ακρίβεια στο σύνολο ελέγχου είναι **92%-95%**.



Εικόνα 10 : Διαγράμματα ακρίβειας - σφάλματος σε σχέση με τις εποχές για υπό κλάσεις



Εικόνα 11 : Διαγράμματα ακρίβειας - σφάλματος σε σχέση με τις εποχές για υπέρ κλάσεις

Η απόδοσή του μοντέλου για κάθε υποκλάση ξεχωριστά φαίνεται στον Πίνακα 1 και για κάθε υπέρ κλάση στον Πίνακα 2.

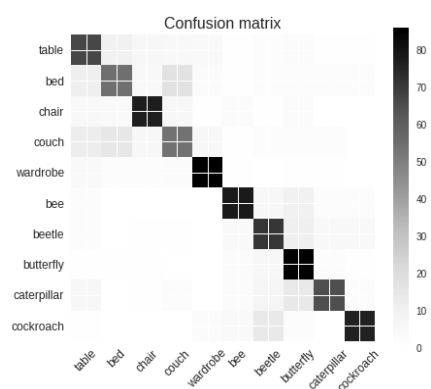
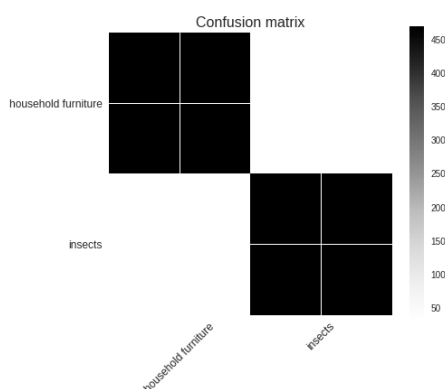
| | table | bed | chair | couch | wardrobe | bee | beetle | butterfly | caterpillar | cockroach |
|-----|-------|------|-------|-------|----------|------|--------|-----------|-------------|-----------|
| Acc | 0.67 | 0.55 | 0.77 | 0.54 | 0.86 | 0.78 | 0.71 | 0.85 | 0.65 | 0.76 |

Πίνακας 1 : Απόδοση μοντέλου ανά κλάση για υπό κλάσεις

| | Household furniture | insects |
|-----|---------------------|---------|
| Acc | 0.94 | 0.94 |

Πίνακας 2 : Απόδοση μοντέλου ανά κλάση για υπέρ κλάσεις

Η εκπαίδευση με τη χρήση του early stopping για τις υπο κλάσεις διήρκεσε ~17 λεπτά για 8 εποχές και θεωρείται αρκετά ικανοποιητικός χρόνος σε σχέση με την περίπτωση των 30 εποχών και ~2 ωρών (που ορίστηκαν αρχικά) χωρίς να μειώνεται η απόδοση του δικτύου. Για τις υπέρ κλάσεις αντίστοιχα η εκτέλεση διήρκεσε ~4 λεπτά για 9 εποχές, αποτέλεσμα αναμενόμενο καθώς απαιτούνται λιγότεροι υπολογισμοί για μικρότερο-ευκολότερο πρόβλημα. Παρακάτω δίνονται οι πίνακες σύγχυσης(confusion matrix) για τις υπέρ κλάσεις αριστερά και για τις υπό κλάσεις δεξιά.



Από τον confusion matrix παρατηρούμε ότι ως προς τις ετικέτες coarse το αποτέλεσμα ήταν πολύ καλό ως προς τον διαχωρισμό των κλάσεων ενώ για τις ετικέτες fine οπτικοποιούνται οι αδυναμίες καθώς, μπορούμε να πούμε ότι τα παρακάτω ζευγάρια αναμείχθηκαν στο αποτέλεσμα: caterpillar - butterfly , cockroach-beetle και couch-bed-table. Γενικά, το μοντέλο μας κατάφερε ιδιαίτερα ικανοποιητικά αποτελέσματα ως προς το διαχωρισμό των υπέρ κλάσεων και με διάφορες βελτιστοποιήσεις των δικτύων καταφέραμε να επιτύχουμε καλύτερη απόδοση και στην ταξινόμηση των υπό κλάσεων (με μεγαλύτερη δυσκολία σαφώς). Εν κατακλείδι, το μοντέλο μας είχε τα αναμενόμενα αποτελέσματα που περιγράψαμε στην *Εισαγωγή*.

Για το δεύτερο μοντέλο (Εικόνα 5) η απόδοσή του για κάθε υποκλάση ξεχωριστά φαίνεται στον Πίνακα 3 και για κάθε υπέρ κλάση στον Πίνακα 4.

| Συνολικό Acc = 0.416 | | | | | | Συνολικό Acc = 0.584 | | | | | |
|----------------------|-------|------|-------|-------|----------|----------------------|------|--------|-----------|-------------|-----------|
| Furniture | | | | | | Insects | | | | | |
| | table | bed | chair | couch | wardrobe | | bee | beetle | butterfly | caterpillar | cockroach |
| Acc | 0.24 | 0.28 | 0.3 | 0.8 | 0.46 | Acc | 0.53 | 0.7 | 0.46 | 0.67 | 0.56 |

Πίνακας 3 : Απόδοση μοντέλου ανά κλάση για υπό κλάσεις

| | Household furniture | insects |
|-----|---------------------|---------|
| Acc | 0.766 | 0.978 |

Πίνακας 4 : Απόδοση μοντέλου ανά κλάση για υπέρ κλάσεις

Για το μοντέλο αυτό, παρατηρούμε ότι η ταξινόμηση των υπέρ κλάσεων πέτυχε επίσης καλή ακρίβεια (για την κλάση έντομα καλύτερη από το μοντέλο 1) αλλά για τις υπό κλάσεις η ακρίβεια είναι αισθητά μειωμένη. Πιστεύουμε, υπάρχουν περιθώρια βελτίωσης του συγκεκριμένου μοντέλου αλλά λόγω ελλειπούς χρόνου δεν προλάβαμε να εφαρμόσουμε όλες τις βελτιστοποιήσεις. **Η συνολική ακρίβεια του ιεραρχικού μοντέλου για τις υπέρ κλάσεις είναι 87.2%, για τις υπό κλάσεις των επίπλων η ακρίβεια είναι 41,6% ενώ για τις υπό κλάσεις των εντόμων είναι 58,4%.**

Τέλος, συγκρίνοντας τα αποτελέσματα των δύο μοντέλων, καταλήγουμε ότι το πρώτο μοντέλο με τα ξεχωριστά CNN πέτυχε καλύτερα αποτελέσματα. Επίσης, ποιοτικά τα αποτελέσματα των δύο μοντέλων συμφωνούν, δηλαδή στις υπέρ κλάσεις πέτυχαν μεγαλύτερη ακρίβεια σε σύγκριση με τις υπό κλάσεις και οι υπο κλάσεις της υπέρ κλάσης “έντομα” ήταν πιο εύκολα διαχωρίσιμες σε σχέση με τις υπό κλάσεις της υπέρ κλάσης “έπιπλα”.

Βιβλιογραφία

- [1] Zhicheng,Y.,Zhang,H.,Piramuthu,R., Jagadeesh,V.,DeCoste,D., Wei,D., Yizhou,Y.,"*HD-CNN: Hierarchical Deep Convolutional Neural Network for Large Scale Visual Recognition*",IEEE International Conference on Computer Vision (ICCV 2015)
- [2] Lecun,Y.,Bottou,L.,Bengio,Y.,Haffner,P.,"*Gradient-based learning applied to document recognition*",Proceedings of the IEEE (Volume: 86 , Issue: 11), (Nov 1998)
- [3] Wang,L.,Sohmshetty,A.,Learning Image Representations to Understand and Predict Semantic Hierarchies,Stanford University
- [4] Deshpande,A.,"A Beginner's Guide To Understanding Convolutional Neural Networks",CS Undergrad at UCLA ('19),Blog About GitHub Projects Resume, (2016)
- [5]https://github.com/justinessert/hierarchical-deep-cnn/blob/master/hdcnn.ipynb?fbclid=IwAR1xgj5DaajG2p3aG3uhnkhkCR6WRH02cPnuny7Xqivgefu43a9n8hT_jTIU