Βαθιά μάθηση στο CIFAR-100 με TF2

Αρχικά βελτιστοποιούμε ένα custom μοντέλο CNN και στη συνέχεια πειραματιζόμαστε με transfer learning χρησιμοποιώντας μοντέλα από το tf.Keras Applications. Η εκπαίδευση των δικτύων έγινε στο Kaggle με χρήση GPU.

CNN “from scratch”

Αρχιτεκτονική

Θα πειραματιστούμε με το να βελτιώσουμε ένα custom CNN εξετάζοντας την αρχιτεκτονική, optimizer, learning rate και batch size. Για τη βελτιστοποίηση χρησιμοποιούμε 20 κλάσεις, σε όλα τα πειράματα χρησιμοποιούμε early stopping με 5 εποχές patience και για τη βελτίωση της αρχιτεκτονικής χρησιμοποιούμε εμπειρικά “default” παραμέτρους με batch size=128, optimizer=Adam και learning rate=0.001. Παρακάτω περιγράφουμε τις αρχιτεκτονικές των δικτύων, όπως σταδιακά βελτιώθηκαν:

- **Simple CNN**: Το CNN που μας παρέχεται στον βοηθητικό κώδικα.
- **CNN (3 blocks)**: Οι αλλαγές που έγιναν σε σχέση με το Simple CNN είναι ότι το δίκτυο αποτελείται από τρία convolutional blocks όπου κάθε block περιέχει ένα convolutional layer και ένα max pooling layer με τα feature maps να είναι 64, 128 και 256 αντίστοιχα.
- **CNN (Batch Normalization)**: Σε κάθε block του CNN (3 blocks) προστέθηκε ένα Batch Normalization Layer μετά το convolutional layer και πριν τη ReLU.
- **CNN (Global Max Pooling)**: Το Flatten και Dense layer μετά τα Convolutional Blocks αντικαταστάθηκαν από ένα Global Max Pooling layer.
- **CNN (4 blocks)**: Προστέθηκε ένα 4^{ο} convolutional block.

Στον παρακάτω πίνακα αναφέρουμε τα αποτελέσματα σχετικά με το accuracy, χρόνο σε δευτερόλεπτα και εποχές εκπαίδευσης για τις παραλλαγές CNN.

Μοντέλο	Accuracy	Εποχές	Χρόνος
Simple CNN	0.48	22	13
CNN (3 blocks)	0.52	20	15
CNN (Batch Normalization)	0.60	16	15
CNN (Global Max Pooling)	0.65	23	21
CNN (4 blocks)	**0.68**	18	19

Παρατηρούμε πως το Batch Normalization αν και πολύ απλή αλλαγή είχε σημαντική βελτίωση στο accuracy.

Επίσης, αναφέρουμε μερικές προσεγγίσεις που δεν δούλεψαν είτε χειροτερεύοντας το accuracy είτε τον χρόνο εκπαίδευσης:

- Η χρήση 0.2 Dropout μετά από το Global Max Pooling χειροτέρεψε το accuracy.
- Η χρήση 0.2 SpatialDropout2D μετά τα convolutional layers δεν άλλαξε το accuracy, αλλά είχε κακή επίδραση στον χρόνο εκπαίδευσης.
- Η προσθήκη 5^{ου} block χειροτέρεψε το accuracy και τον χρόνο εκπαίδευσης.
- Η χρήση 2 convolutional layers σε κάθε block (όπως στο VGG16) βελτίωσε ελάχιστα το accuracy, αλλά προφανώς είχε σημαντική αρνητική επίδραση στον χρόνο εκπαίδευσης.
- Στην προηγούμενη περίπτωση η αντικατάσταση των Convolutional Layers με Separable Convolutional Layers στα τελευταία blocks βελτίωσε σημαντικά την ταχύτητα εκπαίδευσης, αλλά χειροτέρεψε το accuracy.

Optimizer, Learning Rate, Batch Size

Στους παρακάτω πίνακες φαίνονται μερικά πειράματα με optimizers, learning rate και batch size χρησιμοποιώντας την τελική αρχιτεκτονική (CNN 4 blocks με Batch Normalization και Global Max Pooling). Αρχικά, πειραματιζόμαστε με batch size χρησιμοποιώντας Adam(0.001).

Batch Size	Accuracy	Εποχές	Χρόνος
128	**0.68**	18	19
64	0.66	17	24
256	0.67	31	27

Βλέπουμε πως το καλύτερο batch size είναι 128 όπως είχαμε χρησιμοποιήσει αρχικά. Χρησιμοποιώντας άλλο batch size δεν υπάρχει σημαντική επίπτωση στο accuracy, αλλά επηρεάζεται αρνητικά ο χρόνος εκπαίδευσης. Στην συνέχεια εξετάζουμε διαφορετικούς optimizers και learning rates χρησιμοποιώντας batch size 128.

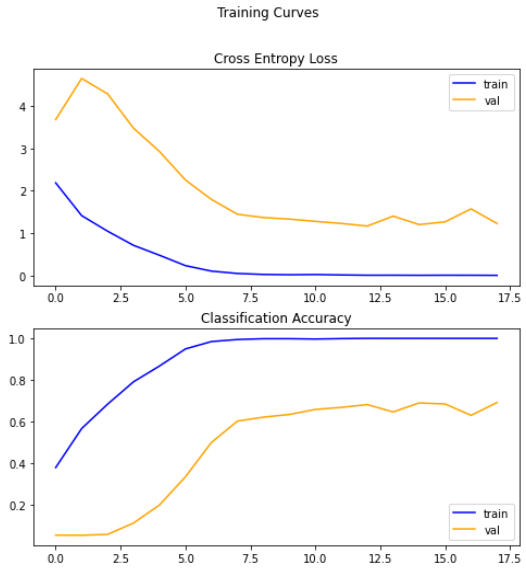
Optimizer(LR)	Accuracy	Εποχές	Χρόνος
Adam(0.001	**0.68**	18	19
Adam(0.0001)	0.60	41	43
Adam(0.01)	0.64	21	22
Nadam(0.001)	0.67	16	20
Nadam(0.0001)	0.59	21	26
Nadam(0.01)	0.66	16	20
Adamax(0.001)	0.63	23	25
Adamax(0.0001)	0.56	28	30
Adamax(0.01)	0.66	20	22

Παρατηρούμε πως τελικά η εμπειρική προσέγγιση με Adam(0.001) ήταν η καλύτερη.

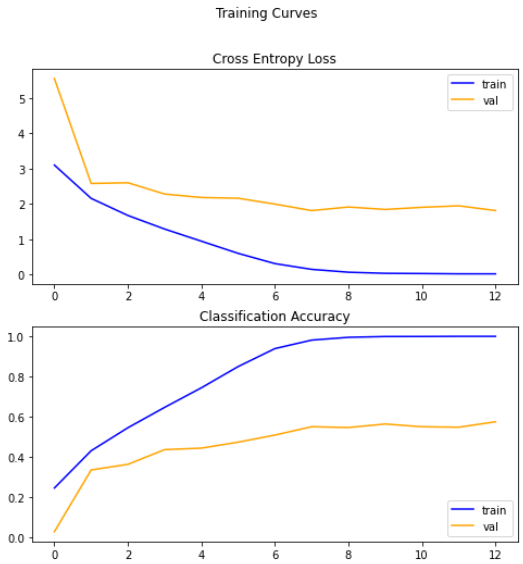
Επίδοση τελικού μοντέλου σε 20 vs 80 κλάσεις

Στον παρακάτω πίνακα και διαγράμματα φαίνεται η επίδοση του τελικού μοντέλου σε 20 και σε 80 κλάσεις.

Κλάσεις	Accuracy	Εποχές	Χρόνος
20	**0.68**	18	19
80	0.57	13	53



Εικόνα 1: Καμπύλες εκμάθησης για 20 κλάσεις



Εικόνα 2: Καμπύλες εκμάθησης για 80 κλάσεις

Παρατηρούμε πως υπήρχε σημαντική αύξηση στον χρόνο εκπαίδευσης με 80 κλάσεις, όπως και ήταν αναμενόμενο, αλλά τα πήγε σχετικά καλά από άποψη accuracy αν σκεφτούμε ότι προστέθηκαν 60 κλάσεις στο πρόβλημα.

Transfer Learning

Σε όλα τα πειράματα χρησιμοποιούμε τον Adam optimizer, για 20 κλάσεις με early stopping.

Μοντέλο και μέγεθος εικόνων

Αρχικά, κρατώντας την default αρχιτεκτονική που μας παρέχεται στον βοηθητικό κώδικα δοκιμάζουμε διαφορετικά προ-εκπαιδευμένα μοντέλα, κάνοντας fine-tune όλα τα layers. Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα με διαφορετικά μοντέλα και μεγέθη εικόνων σε παρένθεση. Ορισμένα μοντέλα δεν δέχονταν εικόνες με μέγεθος 32x32.

Μοντέλο (μέγεθος εικόνας)	Accuracy	Εποχές	Χρόνος
VGG16 (32x32)	0.72	11	32
ResNet50 (32x32)	0.58	21	93
ResNet101 (32x32)	0.62	28	211
ResNet50V2 (32x32)	0.51	20	82
MobileNet (32x32)	0.57	31	49
DenseNet121 (32x32)	0.67	16	75
DenseNet169 (32x32)	0.69	14	95
VGG16 (75X75)	0.81	10	93
MobileNet (75x75)	0.78	37	194
DenseNet121 (75x75)	0.81	18	168
Xception (75x75)	0.84	12	179
InceptionV3 (75x75)	0.73	12	95
InceptionResNetV2 (75x75)	0.74	9	168
VGG16 (64X64)	0.79	9	58
VGG16 (96X96)	0.82	11	143

Παρατηρούμε πως το VGG16 είχε τα καλύτερα αποτελέσματα και για αυτό πειραματιστήκαμε σε αυτό με περισσότερα μεγέθη εικόνων. Βλέπουμε γενικά πως οι εικόνες χρειάζονται up-scale και το καλύτερο μέγεθος φαίνεται να είναι το 75x75 λαμβάνοντας υπόψιν την ταχύτητα σε σχέση με το accuracy.

Blocks για εκπαίδευση

Στη συνέχεια πειραματιζόμαστε με το ποια blocks θα κάνουμε train. Παρακάτω φαίνονται τα αποτελέσματα για διαφορετικά blocks που γίνονται train, ενώ όλα τα προηγούμενα είναι frozen.

Trainable Blocks	Accuracy	Εποχές	Χρόνος
Block 5	0.74	12	44
Blocks 4-5	0.79	10	51
Blocks 3-5	0.79	11	70
Blocks 2-5	0.80	11	83
All layers	**0.81**	10	93

Τελικά, το καλύτερο είναι να κάνουμε fine-tune όλα τα layers, αλλά εάν θέλαμε να εξοικονομήσουμε χρόνο τα δύο τελευταία blocks φαίνεται να είναι αρκετά.

Data augmentation

Στη συνέχεια δοκιμάζουμε data augmentation τεχνικές και στον παρακάτω πίνακα φαίνονται τα αποτελέσματα.

Τεχνική	Accuracy	Εποχές	Χρόνος
Flip	0.80	14	132
Rotation	0.82	14	133
Contrast	0.83	9	85
Translation	0.82	11	104
Zoom	0.80	10	95
Translation + Contrast	0.83	13	124

Με βάση τα αποτελέσματα θα χρησιμοποιήσουμε τελικά data augmentation με random translation και contrast των εικόνων.

Αριθμός κλάσεων

Τέλος, παρουσιάζουμε τα αποτελέσματα στην πρόβλεψη διαφορετικού αριθμού κλάσεων.

Κλάσεις	Accuracy	Εποχές	Χρόνος
20	0.83	13	124
40	0.80	12	227
60	0.76	15	424
80	0.72	14	536

Βλέπουμε πως όπως και με το CNN “from scratch” υπάρχει περίπου 10% διαφορά μεταξύ του accuracy πρόβλεψης 20 με 80 κλάσεων , αλλά το transfer learning γενικά έχει πολύ καλύτερα αποτελέσματα.