ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ & ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ

ΕΡΓΑΣΙΑ 3: ΑΡΧΙΤΕΚΤΟΝΙΚΗ CNN

ΚΑΓΙΑΣ ΑΝΤΩΝΙΟΣ - aid23003

aid23003@uom.edu.gr

Πίνακας περιεχομένων

Εισαγωγή

Το αντικείμενο της εργασίας είναι η δημιουργία ενός CNN μοντέλου που θα χρησιμοποιηθεί πάνω στα δεδομένα του dataset CIFAR-10. Το dataset αυτό περιέχει συνολικά 60.000 εικόνες διαστάσεως 32x32 εκ των οποίων οι 50.000 αποτελούν εικόνες training και οι υπόλοιπες 10.000 είναι για testing. Σε αυτές τα δύο sets οι εικόνες είναι εξ ολοκλήρου διαφορετικές, δηλαδή δεν υπάρχει εικόνα που να ανήκει και στα δύο sets. Επίσης, οι εικόνες ανήκουν σε μία από τις παρακάτω 10 κατηγορίες:

Class	Category
0	Airplane
1	Automobile
2	Bird
3	Cat
4	Deer
5	Dog
6	Frog
7	Horse
8	Ship
9	Truck

Θα δημιουργηθούν δύο αρχεία python. Το πρώτο αρχείο θα περιέχει την δημιουργία και την εκπαίδευση του μοντέλου και στο δεύτερο αρχείο θα χρησιμοποιείται το μοντέλο πάνω στα test data.

Στο πρώτο αρχείο θα γίνουν οι εξής ενέργειες:

1. Φόρτωση του dataset
2. Χωρισμός σε train, validation, test sets
3. Εκτύπωση 4 εικόνων από κάθε κατηγορία (με βάση την πραγματική κατηγορία στην οποία ανήκουν)
4. Δημιουργία και καθορισμός της αρχιτεκτονικής του μοντέλου CNN
5. Εκπαίδευση του μοντέλου
6. Αξιολόγηση του εκπαιδευμένου μοντέλου
7. Αποθήκευση του εκπαιδευμένου μοντέλου

Στο δεύτερο αρχείο θα γίνουν τα παρακάτω:

1. Φόρτωση του αποθηκευμένου μοντέλου
2. Χρήση μοντέλου πάνω στο test set
3. Υπολογισμός accuracy, precision, recall και F1 score
4. Εκτύπωση του confusion matrix
5. Εκτύπωση 4 τυχαίων εικόνων από κάθε κατηγορία (με βάση την πρόβλεψη του μοντέλου)

Τα παραπάνω θα γίνουν άλλες δύο φορές με τις εξής παραλλαγές:

- Το μοντέλο θα έχει την ίδια αρχιτεκτονική αλλά θα χρησιμοποιείται διαφορετικό loss function κατά την εκπαίδευσή του
- Το μοντέλο θα έχει διαφορετική, πιο περίπλοκη αρχιτεκτονική

Τα δύο αρχεία python αποτελούν μέρος των παραδοτέων όπως και το παρόν report. Επίσης, θα παραδοθεί και το Google Colaboratory αρχείο με κατάληξη .ipynb που περιέχει συνολικά όλο τον κώδικα μαζί με τα απαραίτητα σχόλια.

Μέθοδοι που εφαρμόστηκαν

Model 1.0

Αφού κατεβάσουμε και φορτώσουμε το dataset θα κάνουμε κανονικοποίηση των pixel values ώστε να βρίσκονται στο διάστημα [0, 1]. Με την φόρτωση του dataset γίνεται απευθείας ο διαχωρισμός σε train και test sets. Έτσι, στην συνέχεια δημιουργούμε και το validation set από το train set με αναλογία 75/25. Στο τέλος έχουμε 37.500 train εικόνες, 12.500 validation εικόνες και 10.000 test εικόνες.

Έπειτα γίνεται το plot 4 εικόνων από κάθε κατηγορία και ακολουθεί η δημιουργία του μοντέλου CNN η οποία είναι ουσιαστικά στοίβα από στρώματα Conv2D και MaxPooling2D. Στο πρώτο στρώμα θα περαστούν ως `input_shape` οι διαστάσεις των εικόνων που είναι 32x32x3. Αφού προσθέσουμε όλα τα στρώματα που θέλουμε, θα τροφοδοτήσουμε το τελευταίο output tensor του convolutional base σε ένα ή περισσότερα στρώματα Dense για να πραγματοποιήσουμε ταξινόμηση. Τα στρώματα Dense λαμβάνουν ως είσοδο διανύσματα ενώ η έξοδος είναι 3D. Έτσι, πρώτα θα «ισιώσουμε» (flatten) την 3D έξοδο σε έξοδο 1D (διάνυσμα) και μετά θα προσθέσουμε ένα στρώμα Dense. Μάλιστα, επειδή το dataset μας έχει 10 classes, το τελευταίο στρώμα Dense θα έχει 10 εξόδους.

Στη συνέχεια θα κάνουμε compile το μοντέλο μας με arguments `optimizer='adam',loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),metrics =['accuracy']` και θα κάνουμε fit στα train data μας χρησιμοποιώντας και τα validation data. Ακολουθεί μια αξιολόγηση του μοντέλου όπου εκτυπώνουμε σε γράφημα το train accuracy και το validation accuracy που πέτυχε το μοντέλο κατά την εκπαίδευσή του.

Αφού αποθηκεύσουμε το μοντέλο μας (`CIFAR10_CNN.h5`) θα το χρησιμοποιήσουμε πάνω στα test data και θα υπολογίσουμε τα απαραίτητα metric scores, ενώ θα εκτυπώσουμε και τα true positives, true negatives, false positives και false negatives με βάση το confusion matrix. Τέλος, θα εκτυπώσουμε 4 εικόνες από κάθε κατηγορία με βάση το classification που έκανε το μοντέλο μας.

Model 2.0

Για την δεύτερη έκδοση του μοντέλου το μόνο που θα διαφοροποιήσουμε είναι η συνάρτηση loss που θα χρησιμοποιηθεί κατά το compiling του μοντέλου. Στην περίπτωσή μας θα χρησιμοποιηθεί η συνάρτηση CategoricalHinge() η οποία χρησιμοποιείται για categorical data, ωστόσο απαιτεί τα δεδομένα να είναι σε μορφή one-hot-encoding. Η συγκεκριμένη μορφή χρησιμοποιείται για categorical δεδομένα προκειμένου να χρησιμοποιούνται σε μοντέλα μηχανικής μάθησης που απαιτούν ως είσοδο αριθμητικά δεδομένα. Εδώ στην πραγματικότητα έχουμε ήδη αριθμητικά δεδομένα ως είσοδο καθώς τα classes δίνονται με αριθμούς από 0 έως 9, ωστόσο το one-hot-encoding απαιτεί αριθμούς 0 και 1. Έτσι, ο μετασχηματισμός θα γίνει ως εξής:

Class labels	One-hot-encoding class labels
0	**1** 0 0 0 0 0 0 0 0 0
1	0 **1** 0 0 0 0 0 0 0 0
2	0 0 **1** 0 0 0 0 0 0 0
3	0 0 0 **1** 0 0 0 0 0 0
…	…

Για παράδειγμα, αν μια εικόνα πριν ανήκε στο class 7 δηλαδή στο horse, τώρα θα ανήκει στο class 0 0 0 0 0 0 0 1 0 0.

Στα υπόλοιπα δεν αλλάζει κάτι σε σχέση με πριν. Θα γίνει η εκπαίδευση του μοντέλου, η αξιολόγηση, η αποθήκευση (`CIFAR10_CNN_2.h5`), η χρήση στα test data, ο υπολογισμός των μετρικών, η εκτύπωση του confusion matrix και η εκτύπωση 4 εικόνων από κάθε κατηγορία με βάση τις προβλέψεις του μοντέλου.

Model 3.0

Η τρίτη έκδοση του μοντέλου θα περιλαμβάνει διαφορετική αρχιτεκτονική. Στην πρώτη έκδοση του μοντέλου θα προσθέσουμε άλλο ένα στρώμα Conv2D με 64 filters και kernel size = (3,3) και άλλο ένα στρώμα MaxPooling2D με pool size = (2,2). Ακολουθεί προφανώς η εκπαίδευση, η αξιολόγηση και η αποθήκευση του μοντέλου (`CIFAR10_CNN_3.h5`), η χρήση πάνω στα test data, ο υπολογισμός των μετρικών, η εκτύπωση του confusion matrix και η εκτύπωση 4 εικόνων από κάθε κατηγορία με βάση τις προβλέψεις του.

Συμπεράσματα

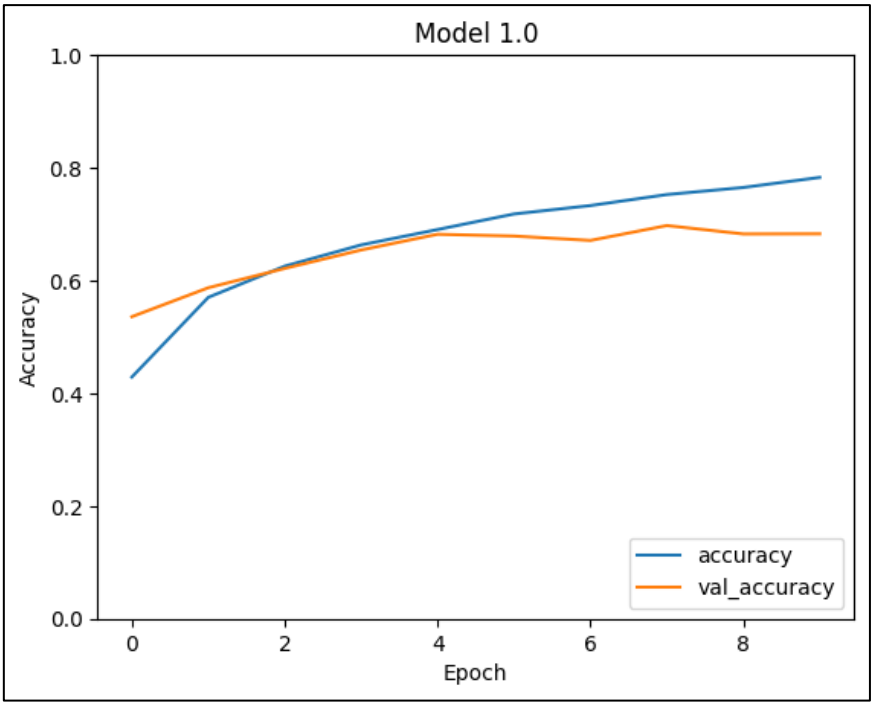
Αρχιτεκτονικές - training και validation accuracy - confusion matrices

Model 1.0

Η αρχιτεκτονική του μοντέλου έκδοσης 1.0 είναι:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
max_pooling2d (MaxPooling2D)  (None, 15, 15, 32)       0
conv2d_1 (Conv2D)            (None, 13, 13, 64)        18496
max_pooling2d_1 (MaxPooling2D)  (None, 6, 6, 64)         0
conv2d_2 (Conv2D)            (None, 4, 4, 64)          36928
flatten (Flatten)            (None, 1024)              0
dense (Dense)                (None, 64)                65600
dense_1 (Dense)              (None, 10)                650
=================================================================
Total params: 122,570
Trainable params: 122,570
Non-trainable params: 0
_________________________________________________________________
```

Μετά την εκπαίδευση του μοντέλου δημιουργούμε το παρακάτω γράφημα όπου αποτυπώνεται η εξέλιξη του training accuracy (με μπλε) και του validation accuracy (με πορτοκαλί):



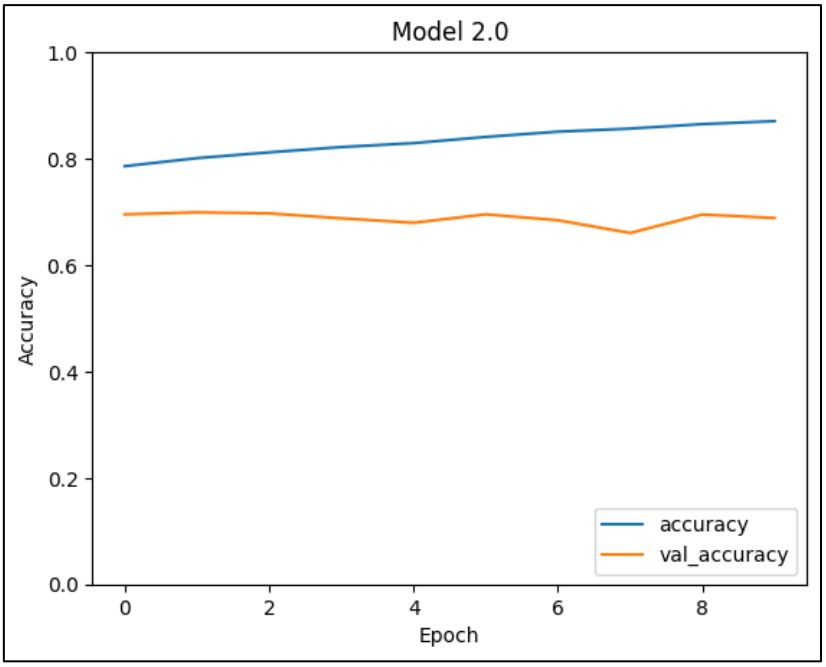
Εικόνα 1. Training accuracy vs. validation accuracy στο model 1.0

Παρακάτω αντλούμε τις περιπτώσεις TP, TN, FP και FN από τον confusion matrix μετά την εφαρμογή του μοντέλου πάνω στα test data μας. Ο πίνακας διαβάζεται ως εξής. Για την κατηγορία 0 το μοντέλο προέβλεψε σωστά 790 εικόνες που πράγματι ανήκουν σε αυτή (true positives) και 8603 εικόνες που πράγματι δεν ανήκουν σε αυτή (true negatives). Αντίθετα, προέβλεψε ότι 397 εικόνες ανήκουν στην κατηγορία 0 ενώ δεν ανήκουν στην πραγματικότητα σε αυτή (false positives) και πως 210 εικόνες δεν ανήκουν σε αυτή την κατηγορία ενώ στην πραγματικότητα ανήκουν (false negatives). Κάθε κατηγορία περιέχει 1.000 εικόνες (TP+FN) και κάθε γραμμή αθροίζει στο 10.000 (συνολικά οι testing εικόνες).

Class	TP	TN	FP	FN
0	790	8603	397	210
1	831	8826	174	169
2	530	8740	260	470
3	584	8217	783	416
4	510	8825	175	490
5	708	8292	708	354
6	646	8882	118	354
7	713	8763	237	287
8	785	8831	169	215
9	731	8849	151	269

Model 2.0

Η αρχιτεκτονική του μοντέλου έκδοσης 2.0 είναι η ίδια με την έκδοση 1.0 (αφού αλλάζει μόνο το loss function). Επομένως, η σύγκριση μεταξύ training και validation accuracy παρουσιάζεται παρακάτω:



Εικόνα 2. Training accuracy vs. validation accuracy στο model 2.0

Όπως φαίνεται, το validation accuracy είναι σε παρόμοιες τιμές με το μοντέλο 2.0, όμως το training accuracy κυμαίνεται σε υψηλότερα επίπεδα.

Παρόμοια με πριν αντλούμε τις περιπτώσεις TP, TN, FP και FN από τον confusion matrix για το μοντέλο 2.0:

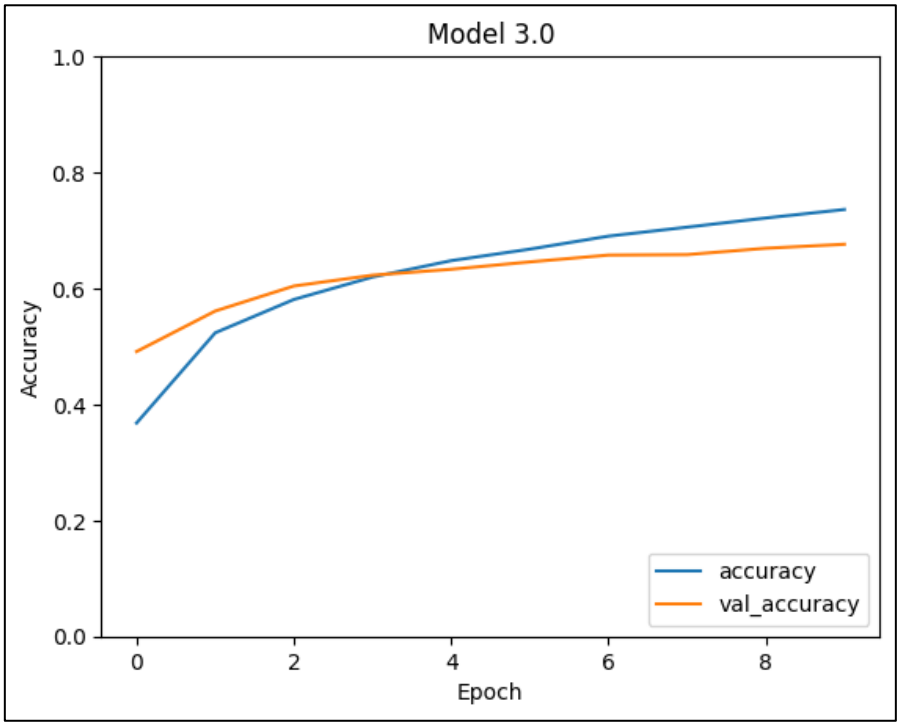
Class	TP	TN	FP	FN
0	711	8671	329	289
1	806	8842	158	194
2	611	8469	531	389
3	517	8421	579	483
4	647	8619	381	353
5	554	8600	400	446
6	635	8881	119	365
7	743	8708	292	257
8	807	8782	218	193
9	753	8791	209	247

Model 3.0

Η αρχιτεκτονική του μοντέλου έκδοσης 3.0 είναι:

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 30, 30, 32)        896
max_pooling2d_2 (MaxPooling2D)  (None, 15, 15, 32)       0
conv2d_4 (Conv2D)            (None, 13, 13, 64)        18496
max_pooling2d_3 (MaxPooling2D)  (None, 6, 6, 64)         0
conv2d_5 (Conv2D)            (None, 4, 4, 64)          36928
conv2d_6 (Conv2D)            (None, 2, 2, 64)          36928
max_pooling2d_4 (MaxPooling2D)  (None, 1, 1, 64)         0
flatten_1 (Flatten)          (None, 64)                0
dense_2 (Dense)              (None, 64)                4160
dense_3 (Dense)              (None, 10)                650
=================================================================
Total params: 98,058
Trainable params: 98,058
Non-trainable params: 0
_________________________________________________________________
```

Ακολουθεί και πάλι η σύγκριση training και validation accuracy:



Εικόνα 3. Training accuracy vs. validation accuracy στο model 3.0

Παρατηρούμε ότι και τα δύο κινούνται σε παρόμοια επίπεδα με το μοντέλο της έκδοσης 1.0, πράγμα που είναι λογικό αφού η αρχιτεκτονική είναι παρόμοια και δεν προσθέσαμε πολλά παραπάνω στρώματα. Τέλος, από τον confusion matrix έχουμε:

Class	TP	TN	FP	FN
0	803	8545	455	197
1	754	8896	104	246
2	573	8535	465	427
3	454	8597	403	546
4	622	8574	426	378
5	609	8506	494	391
6	824	8559	441	176
7	690	8774	226	310
8	678	8859	141	322
9	699	8861	139	301

Metric scores

Ο παρακάτω πίνακας περιλαμβάνει όλες τις μετρικές για κάθε μοντέλο:

Model	Accuracy	Precision	Recall	F1 score
Model 1.0	0.68	0.71	0.68	0.69
Model 2.0	0.68	0.69	0.68	0.68
Model 3.0	0.67	0.68	0.67	0.67

Όλα τα μοντέλα βρίσκονται πολύ κοντά από πλευράς αποδόσεων στις παραπάνω μετρικές και θα λέγαμε ότι παράγουν παρόμοια αποτελέσματα. Παρ' όλα αυτά, στα σημεία μπορούμε να διακρίνουμε ότι το μοντέλο 1.0 βρίσκεται στην $1^{η}$ θέση, το μοντέλο 2.0 στη $2^{η}$ θέση και το μοντέλο 3.0 στη $3^{η}$ θέση.