ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ

Εργασία μαθήματος

«Τεχνητή Νοημοσύνη και Εφαρμογές στο Διαδίκτυο των Πραγμάτων»

Αναγνώριση Σημάτων Οδικής Κυκλοφορίας για Αυτόνομα Οχήματα

Τελικό Παραδοτέο

Τζιόγκας Ιωάννης A.M:itp20135
Χατζηθεοδούλου Μαριλένα A.M: itp20146

Περιγραφή του προβλήματος

Το μεγαλύτερο ποσοστό των αυτοκινητιστικών ατυχημάτων οφείλεται στον ανθρώπινο παράγοντα.

Το να αναγνωρίζει ένα όχημα μια σήμανση και να προσαρμόζεται στον ΚΟΚ ή να δρα εγκαίρως έτσι ώστε να αποφευχθεί ένα ατύχημα, θα μειώσει τον αριθμό αυτό και γιατί όχι να τον εξαλείψει.

Αυτό που χρειάζεται είναι να αναγνωρίζει , να δρα και το κυριότερο άμεσα. Αρα ο αλγόριθμος αναγνώρισης πρέπει να είναι γρήγορος.

Στο κομμάτι της εργασίας ασχολούμαστε με την αναγνώριση σημάτων οδικής κυκλοφορίας. Προσπαθήσαμε να φτάσουμε στο καλύτερο δυνατό αποτέλεσμα πρόβλεψης αυξάνοντας το βάθος του μοντέλου μας. Να αντιμετωπίσουμε τυχόν προβλήματα που θα παρουσιαστούν πχ overfitting. Και να δούμε την συμπεριφορά των μοντέλων μας με διαφορετικές παραμέτρους.

Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη αναφέρεται στην ικανότητα μιας μηχανής να αναπαράγει τις γνωστικές λειτουργίες ενός ανθρώπου, όπως είναι η μάθηση, ο σχεδιασμός και η δημιουργικότητα.

Η τεχνητή νοημοσύνη καθιστά τις μηχανές ικανές να 'κατανοούν' το περιβάλλον τους, να επιλύουν προβλήματα και να δρουν προς την επίτευξη ενός συγκεκριμένου στόχου. Ο υπολογιστής λαμβάνει δεδομένα (ήδη έτοιμα ή συλλεγμένα μέσω αισθητήρων, π.χ. κάμερας), τα επεξεργάζεται και ανταποκρίνεται βάσει αυτών.

Τα συστήματα τεχνητής νοημοσύνης είναι ικανά να προσαρμόζουν τη συμπεριφορά τους, σε ένα ορισμένα βαθμό, αναλύοντας τις συνέπειες προηγούμενων δράσων και επιλύοντας προβλήματα με αυτονομία.

Τι μπορεί να κάνει σήμερα η ΤΝ;

- Παιχνίδια
- Ρομποτική
- Αυτόνομη οδήγηση
- Όραση με υπολογιστές
- Αναγνώριση αντικειμένων

- Ανάκτηση πληροφοριών
- Recommender systems
- Αυτόματη μετάφραση
- Σύνθεση κειμένου
- Ερωτήσεις και απαντήσεις
- Φίλτρα ανεπιθύμητης αλληλογραφίας
- Αναγνώριση χειρόγραφων χαρακτήρων
- Πρόβλεψη ζήτησης (demand forecasting)
- Μοντέλα προβλέψεων καιρικών, πολιτικών.

Μάθηση - Μηχανική μάθηση

Η μάθηση είναι από τα σημαντικότερα εργαλεία του ανθρώπου. Του επιτρέπει να απομνημονεύει και να χρησιμοποιεί δεδομένα που λαμβάνει από το περιβάλλον ώστε να βελτιώνεται σταδιακά κατα την εκτέλεση μιας λειτουργίας.

Αποτελεί ένα από τα σημαντικότερα χαρακτηριστικα της ανθρώπινης νοημοσύνης αν αναλογιστούμε την συμβολή της σε καθημερινές ανθρώπινες λειτουργίες όπως: ομιλία, κατανόηση κειμένων, αναγνώριση προτύπων, αναγνώριση εικόνων κ.α.

Όσο είναι αναγκαία η μάθηση για την ανθρώπινη νοημοσύνη άλλο τόσο σπουδαία είναι η μηχανική μάθηση για την τεχνητή νοημοσύνη.

Η μηχανική μάθηση είναι το φαινόμενο κατα το οποιο ενα συστημα βελτιωνει την αποδοση του κατα την εκτελεση μιας εργασιας χωρις να

υπαρχει αναγκη να προγραμματιστεί εκ νέου.

Κατηγορίες μηχανικής μάθησης

Επιβλεπόμενη μαθηση(Supervised Learning)

Είναι η διαδικασία όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους (σύνολο εκπαίδευσης) σε γνωστές επιθυμητές εξόδους, με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο. Χρησιμοποιείται σε προβλήματα:

- ταξινόμησης
- προγνωσης
- διερμηνείας

Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)

Είναι η διαδικασία όπου ο αλγόριθμος κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων υπό μορφή παρατηρήσεων χωρίς να γνωρίζει τις επιθυμητές εξόδους.

Χρησιμοποιείται σε προβλήματα:

- Ανάλυσης Συσχετισμών (Association Analysis)
- Ομαδοποίησης (Clustering)

Ενισχυτική Μάθηση (Reinforcement Learning)

Είναι η διαδικασία όπου ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσα από άμεση αλληλεπίδραση με το περιβάλλον. Χρησιμοποιείται κυρίως σε προβλήματα:

- Σχεδιασμού (Planning), όπως για παράδειγμα ο έλεγχος κίνησης ρομπότ και η βελτιστοποίηση εργασιών σε εργοστασιακούς χώρους

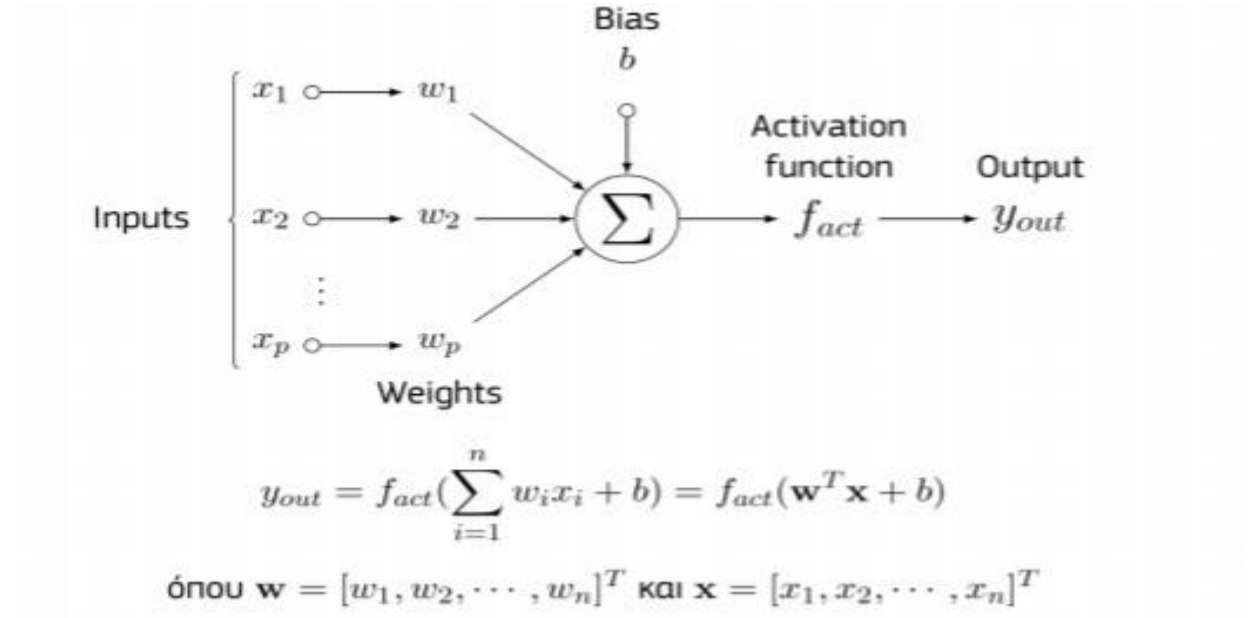
Νευρωνικά δίκτυα

Τα Νευρωνικά Δίκτυα (Neural Networks) παρέχουν ένα πρακτικό τρόπο για την εκμάθηση αριθμητικών και διανυσματικών συναρτήσεων ορισμένων σε συνεχή ή διακριτά μεγέθη. Χρησιμοποιούνται τόσο για παλινδρόμηση(γραμμική και μη γραμμική) όσο και για ταξινόμηση. Έχουν το μεγάλο πλεονέκτημα της ανοχής που παρουσιάζουν σε δεδομένα εκπαίδευσης με θόρυβο, δηλαδή δεδομένα που περιστασιακά έχουν λανθασμένες τιμές. Αδυνατούν όμως να εξηγήσουν ποιοτικά τη γνώση που μοντελοποιούν.

Μοντέλο Τεχνητού Νευρώνα

Ο θεμελιώδης λίθος των νευρωνικών δικτύων είναι ο Νευρώνας. Ένας νευρώνας αποτελείται από τις μεταβλητές εισόδου του, τον κορμό του και μία μεταβλητή εξόδου. Πολλοί νευρώνες παράλληλοι αποτελούν ένα επίπεδο του νευρωνικού δικτύου. Οι νευρώνες του ίδιου επιπέδου δεν

συνδέονται μεταξύ τους, αλλά συνδέονται ένας προς έναν με όλους τους νευρώνες του επόμενου επιπέδου. Κάθε νευρωνικό δίκτυο έχει τουλάχιστον δύο επίπεδα, το επίπεδο εισόδου και το επίπεδο εξόδου (Perceptron).



Βαθιά Μάθηση (Deep Learning)

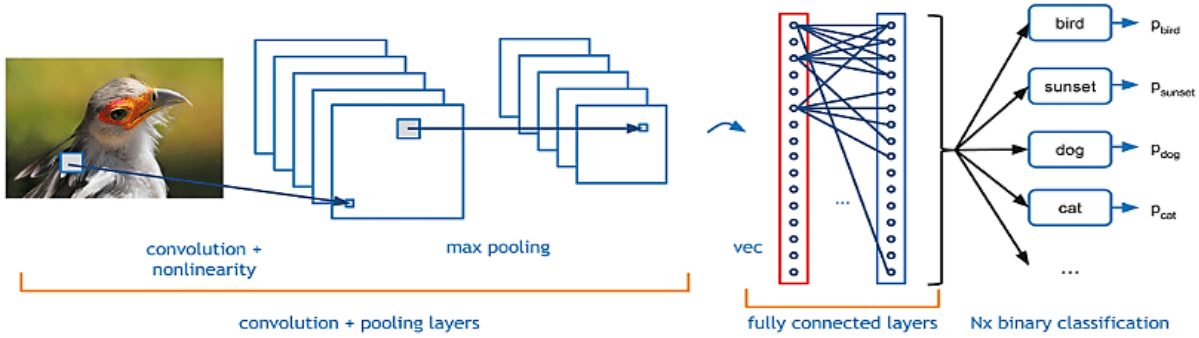
Αποτελεί μια τεχνική μάθησης, η οποία αποσκοπεί να διδάξει τους υπολογιστές πως μπορούν να μαθαίνουν αυτόνομα μέσα από παραδείγματα, όπως ακριβώς και ο άνθρωπος – χωρίς την ανάγκη παροχής προτυποποιημένων και καθορισμένων μοντέλων γνώσης. Στη Βαθιά Μάθηση, ένα υπολογιστικό σύστημα μαθαίνει να εκτελεί εργασίες ταξινόμησης απευθείας από τα εισερχόμενα δεδομένα (εικόνες, κείμενο, ήχος κτλ). Τα μοντέλα Βαθιάς Μάθησης μπορούν να επιτύχουν ακρίβεια που πολλές φορές υπερβαίνει την ανθρώπινη απόδοση(ResNet 2015). Ο

όρος "Βαθιά", συνήθως αναφέρεται στον αριθμό των κρυφών επιπέδων (hidden layers).

Συνελικτικά νευρωνικά δίκτυα(CNN)

Ένα συνελικτικό δίκτυο είναι μία νευρωνική αρχιτεκτονική πολλών επιπέδων ειδικά σχεδιασμένη ώστε να αναγνωρίζει δισδιάστατα σχήματα με υψηλό βαθμό μη ευαισθησίας στη μετατόπιση, την κλιμάκωση, την στρέβλωση και άλλες μορφές παραμόρφωσης.

Ο πιο σύγχρονος τρόπος ανάλυσης των φωτογραφιών είναι ένα συγκεκριμένο είδος νευρωνικών δικτύων, τα συνελικτικά δίκτυα. Η ιδιαιτερότητα τους, οφείλεται στην ικανότητά τους να αναγνωρίζουν συγκεκριμένα (πολλά και διαφορετικά) στοιχεία εντός των εικόνων πετυχαίνοντας με αυτό τον τρόπο μεγάλες ακρίβειες στην κατηγοριοποίηση του απεικονιζόμενου στην εικόνα.



Στο παραπάνω σχήμα παρουσιάζονται όλα τα βασικά επίπεδα ενός συνελικτικού δικτύου.

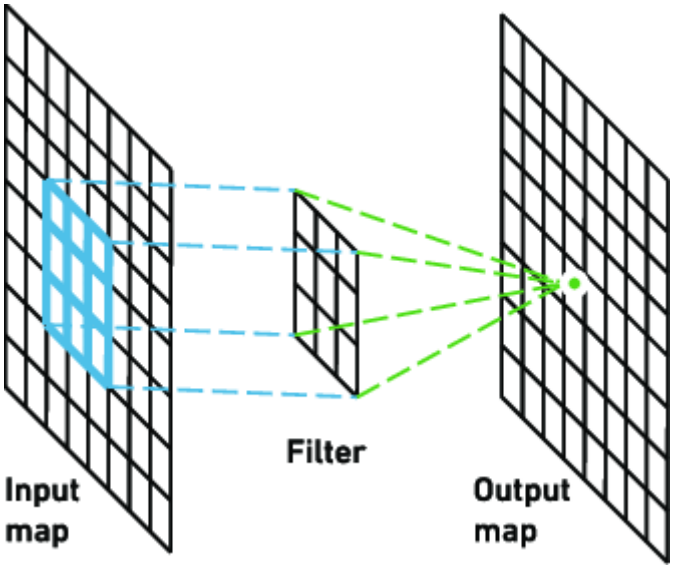
Επίπεδο εισόδου (Input layer):

Είναι το αρχικό επίπεδο του δικτύου. Όπου φορτώνονται τα δεδομένα μέσα στο δίκτυο. Τα δεδομένα εισόδου στην παρούσα εργασία είναι εικόνες 30X230 με 3 κανάλια (RGB).

Συνελικτικό επίπεδο:

Δέχεται τα δεδομένα και τα μετασχηματίζει, χρησιμοποιώντας ένα σύνολο συνδεδεμένων νευρώνων του προηγούμενου επιπέδου. Είναι το πιο βασικό δομικό μέρος στην αρχιτεκτονική του συνελικτικού νευρωνικού δικτύου. Ένα συνελικτικό επίπεδο, συνελίσσει την είσοδο του με μία σειρά από φίλτρα ή πυρήνες (kernels), παράγοντας έτσι χάρτες χαρακτηριστικών. Στα συνελικτικά δίκτυα η συνέλιξη είναι γνωστή και ως ανιχνευτής χαρακτηριστικών (feature detector). Η είσοδος της μπορεί να είναι ακατέργαστα δεδομένα ή ένας χάρτης χαρακτηριστικών (feature map) που προήλθε από προηγούμενου επίπεδο του δικτύου.

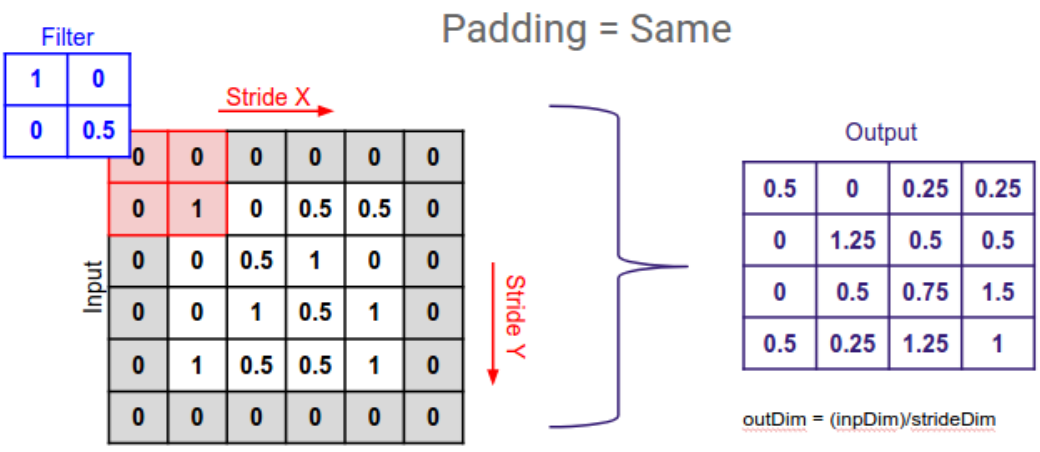
Παράδειγμα συνελικτικού επιπέδου



Στην εργασία μας θα χρησιμοποιηθεί σε όλα τα μοντέλα με παραμέτρους:

- padding=same (ουσιαστικά μετά την συνέλιξη δεν αλλάζει το μέγεθος του πίνακα-προσθέτουμε μηδενικά στον αρχικό πίνακα)

παράδειγμα padding=same



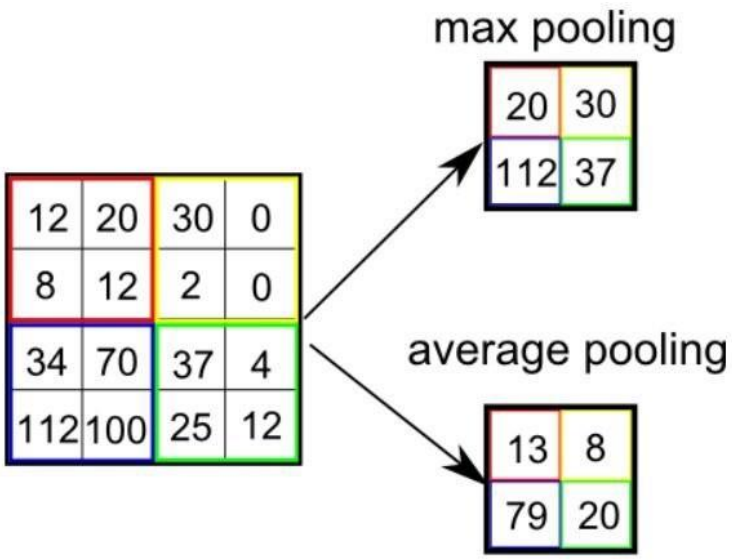
- Activation function = ReLu (Περιγράφεται παρακάτω)

Συγκεντρωτικό επίπεδο (Pooling layer)

Τα συγκεντρωτικά επίπεδα συνήθως εισάγονται μεταξύ διαδοχικών συνελικτικών επιπέδων. Στόχος είναι η προοδευτική μείωση του χωρικού μεγέθους (πλάτος και ύψος) της αναπαράστασης δεδομένων και των παραμέτρων του δικτύου, έτσι ώστε να ελεγχθεί η υπερπροσαρμογή (overfitting) του μοντέλου. Επομένως τα συγκεντρωτικά επίπεδα μειώνουν τις διαστάσεις κάθε χάρτη χαρακτηριστικών αλλά διατηρεί τις πιο σημαντικές πληροφορίες. Είναι σημαντικό να σημειωθεί ότι αυτή η λειτουργία δεν επηρεάζει τη διάσταση του βάθους. Ορίζεται ένας πίνακας πχ 2x2 που μετακινείται πάνω στην εικόνα και ανάλογα και διατηρεί το μεγαλύτερο στοιχείο (max pooling) ή τον μέσο όρο των στοιχείων του 2X2

(average pooling)

Παράδειγμα pooling 2x2



Συναρτήσεις ενεργοποίησης (Activation Functions)

Πρακτικά αποτελούν έναν κόμβο που τοποθετείται στο τέλος ή ανάμεσα σε ΝΔ. Βοηθά στο να παρθεί η απόφαση για το αν θα πυροδοτηθεί ένας νευρώνας ή όχι. “Η συνάρτηση ενεργοποίησης είναι ο μη-γραμμικός μετασχηματισμός που εφαρμόζουμε στο σήμα εισόδου. Η μετασχηματισμένη έξοδος στέλνεται στο επόμενο επίπεδο, για το οποίο και αποτελεί σήμα εισόδου”

ReLU Activation Function

Η συνάρτηση ReLU (Rectified Linear Unit) είναι μία μη – γραμμική συνάρτηση ενεργοποίησης η οποία είναι ίσως η πιο δημοφιλής στον τομέα της βαθιάς μάθησης και των Συνελικτικών Νευρωνικών Δικτύων τα τελευταία χρόνια. Στην πράξη δέχεται ως είσοδο έναν αριθμό και σε περίπτωση που είναι αρνητικός η έξοδος ισούται με 0, διαφορετικά ισούται με τον ίδιο τον αριθμό. Ο λόγος για τον οποίο χρησιμοποιείται πολύ τα τελευταία χρόνια είναι επειδή επιταχύνει την εύρεση των βέλτιστων τιμών από τον αλγόριθμο gradient descent συγκριτικά με τις συναρτήσεις ενεργοποίησης που αναφέρθηκαν προηγουμένως. Ιδανικά χρησιμοποιείται σε βαθιά νευρωνικά δίκτυα με πολλά επίπεδα, καθώς έχει την ιδιότητα να περιορίζει κατά πολύ τον αριθμό των νευρώνων που ενεργοποιούνται, με αποτέλεσμα το νευρωνικό δίκτυο να γίνεται ελαφρύτερο και να επιταχύνονται ιδιαίτερα οι υπολογισμοί. Η σχέση η οποία περιγράφει την ReLU φαίνεται παρακάτω:

$$RELU(x) = \begin{cases} 0 \ if \ x < 0 \\ x \ if \ x >= 0 \end{cases}$$

Παράδειγμα εφαρμογής της Relu σε τιμές [-10,10](python)

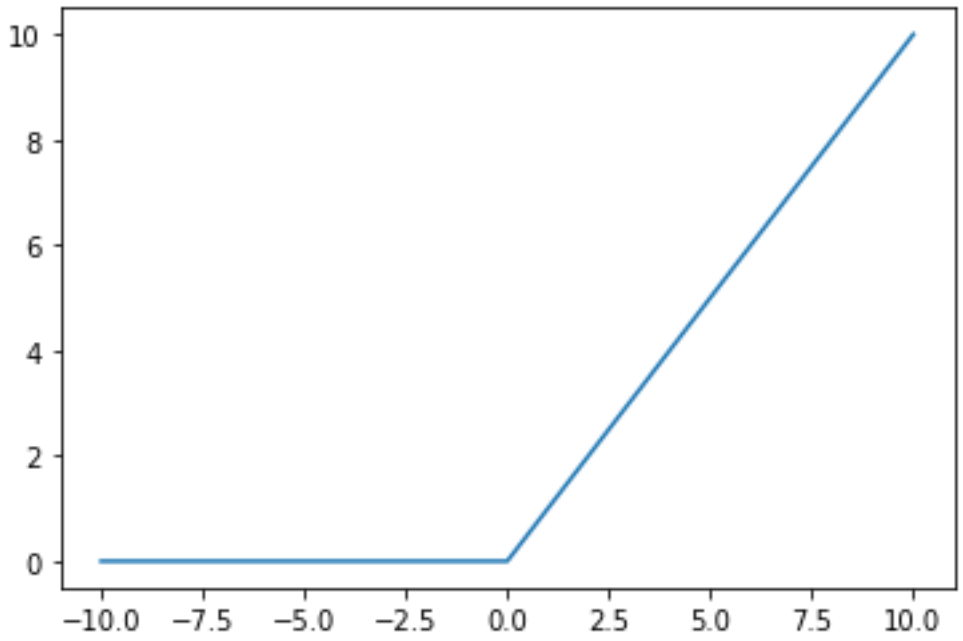
```python
from matplotlib import pyplot as plt

def relu(x):
  return max(0.0,x)

x=[]
y=[]
for i in range (-10,11):
  x.append(i)
  y.append(relu(i))

plt.plot(x,y)
plt.show()
```

Η αναπαράσταση της



Softmax Activation Function

Η συνάρτηση softmax χρησιμοποιείται σε προβλήματα ταξινόμησης, καθώς μετατρέπει την έξοδο του νευρωνικού δικτύου σε τιμές ανάμεσα σε 0 και 1 , όπως η σιγμοειδής συνάρτηση, αλλά επιπλέον το συνολικό άθροισμα όλων των εξόδων είναι 1. Η έξοδός της δείχνει δηλαδή την πιθανότητα του κάθε παραδείγματος να ανήκει σε κάθε κλάση, είναι ιδανική επομένως για προβλήματα ταξινόμησης σε πολλές κλάσεις. Η σχέση που εκφράζει τη συνάρτηση ενεργοποίησης softmax φαίνεται παρακάτω:

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_i}}$$

Πλήρως συνδεδεμένο επίπεδο (fully connected layer)

Το πλήρως συνδεδεμένο επίπεδο είναι μία παραδοσιακή αρχιτεκτονική πολλών επιπέδων με νευρώνες, η οποία χρησιμοποιεί μία συνάρτηση ενεργοποίησης (συνήθως τη softmax) στην έξοδό της. Κάθε επίπεδο που

ανήκει σε αυτή την αρχιτεκτονική έχει την ιδιότητα πως κάθε νευρώνας που περιλαμβάνει συνδέεται με όλους τους νευρώνες του προηγούμενου επιπέδου. Στις αρχιτεκτονικές μας χρησιμοποιούμε την softmax με έξοδο τις πιθανότητες να ανήκει σε κάποια από τις 43 κλάσεις.

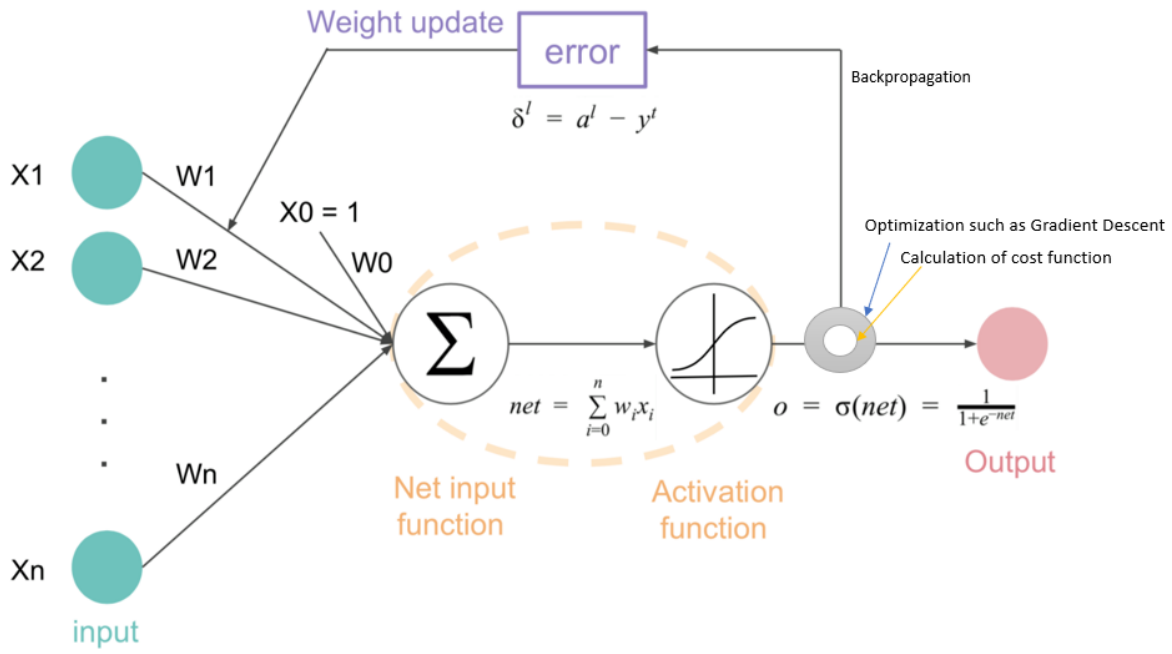
Συνάρτηση κόστους (loss function)

Οι συναρτήσεις κόστους ποσοτικοποιούν πόσο αποδοτικά έχει εκπαιδευτεί το νευρωνικό δίκτυο, χρησιμοποιώντας τα δεδομένα εκπαίδευσης. Οι συναρτήσεις κόστους εκφράζουν μία μέτρηση με βάση το σφάλμα που παρατηρείται στις προβλέψεις του δικτύου. Ο μέσος όρος των σφαλμάτων σε ολόκληρο το σύνολο δεδομένων εκφράζει πόσο κοντά είναι το μοντέλο σε ένα ιδανικό μοντέλο που δεν κάνει ποτέ λάθος. Η αναζήτηση αυτής της ιδανικής κατάστασης ισοδυναμεί με την εύρεση των παραμέτρων που θα ελαχιστοποιήσουν τη συνάρτηση κόστους. Με βάση αυτή τη λογική, οι συναρτήσεις κόστους χρησιμοποιούνται έτσι ώστε η αποδοτική εκπαίδευση του νευρωνικού δικτύου να ανάγεται σε ένα πρόβλημα βελτιστοποίησης. Υπάρχουν πολλές συναρτήσεις κόστους που χρησιμοποιούνται στα προβλήματα βελτιστοποίησης. Για προβλήματα παλινδρόμησης (regression) η πιο κοινή συνάρτηση κόστους είναι αυτή του μέσου τετραγωνικού σφάλματος (mean squared error – mse), ενώ σε προβλήματα κατηγοριοποίησης (classification) συνήθως χρησιμοποιείται η συνάρτηση εντροπίας (cross entropy). Στις αρχιτεκτονικές μας σαν συνάρτηση κόστους χρησιμοποιούμε την categorical cross entropy.

Backpropagation Algorithm

Σημαντικό μέρος της διαδικασίας εκπαίδευσης αποτελεί ο αλγόριθμος Back Propagation, ο οποίος χρησιμοποιείται για τη μείωση του σφάλματος του νευρωνικού δικτύου. Το κλειδί είναι να βρεθούν τα βάρη που είναι περισσότερο υπεύθυνα για το εξαγόμενο σφάλμα και στη συνέχεια να αλλάξουν οι τιμές αυτών αναλόγως. Δηλαδή, αν ένα βάρος επηρεάζει περισσότερο την τιμή του σφάλματος τότε πρέπει να δεχθεί μεγαλύτερη προσαρμογή. Ο αλγόριθμος Back Propagation είναι μία ρεαλιστική προσέγγιση για τη διαίρεση της συμβολής σφάλματος για κάθε βάρος του νευρωνικού δικτύου

Backpropagation



Αλγόριθμοι βελτιστοποίησης

Gradient Descent

Είναι ένας επαναληπτικός αλγόριθμος ελαχιστοποίησης ο οποίος σε κάθε βήμα μεταβάλει τις τιμές των μεταβλητών αντίθετα από την παράγωγο της συνάρτησης. Η επανάληψη σε κάθε βήμα δίνεται από την σχέση:

$\boldsymbol{\theta}t+1 = \boldsymbol{\theta}t - \eta\nabla\boldsymbol{\theta}J(\boldsymbol{\theta})$ όπου:

η: ρυθμός εκμάθησης (learning rate)

$\nabla\boldsymbol{\theta}J(\boldsymbol{\theta})$: Διάνυσμα κλίσης που υπολογίζεται στο σημείο $\boldsymbol{\theta}t$

Stochastic Gradient Descent

Η διαφορά του με τον Gradient Descent είναι ότι ο SGD εφαρμόζεται σε ένα ή περισσότερα μικρά δείγματα εκπαίδευσης και όχι σε όλο το δείγμα εκπαίδευσης όπως κάνει ο Gradient Descent. Αυτό στην πράξη επιταχύνει την εκπαίδευση του νευρωνικού δικτύου.

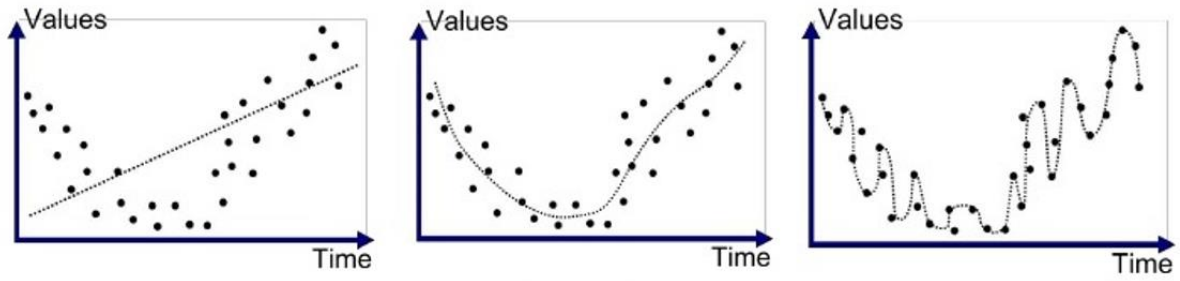
Adam

Ειναι μια επέκταση του SDG, είναι ευρέως διαδεδομένος στον τομέα της βαθιάς μηχανικής μάθησης γιατί επιτυγχάνει καλά αποτελέσματα γρήγορα. Είναι και ο αλγόριθμός βελτιστοποίησης που θα χρησιμοποιήσουμε στις αρχιτεκτονικές μας με διαφορετικά Learning rate για κάθε αρχιτεκτονική: `l=[0.01,0.005,0.001,0.0005,0.0001]` και μικρά δείγματα των 32 φωτογραφιών batch_size = 32.

Πρόβλημα υπερπροσαρμογής (overfitting)

Οι αλγόριθμοι βελτιστοποίησης επιχειρούν πρώτα να λύσουν το πρόβλημα της υποπροσαρμογής (underfitting). Όταν το μοντέλο δεν έχει εκπαιδευτεί αποδοτικά τότε δεν προσεγγίζει καλά τα δεδομένα. Αντιθέτως, σε περίπτωση που το μοντέλο έχει εκπαιδευτεί πολύ καλά στις λεπτομέρειες και στο θόρυβο του εκπαιδευτικού συνόλου δεδομένων και αδυνατεί να γενικεύσει αποδοτικά σε δεδομένα που δεν έχουν χρησιμοποιηθεί στο στάδιο της εκπαίδευσής του, τότε παρουσιάζεται το πρόβλημα της υπερ προσαρμογής (overfitting).Στόχος των μοντέλων της μηχανικής μάθησης είναι η αποδοτική γενίκευση σε νέα δεδομένα, έτσι ώστε να εξάγονται σωστές προβλέψεις. Επομένως, ιδανικά, το μοντέλο πρέπει να εκπαιδευτεί πολύ καλά με χρήση των δεδομένων εκπαίδευσης, έτσι ώστε να λάβει τη διαθέσιμη πληροφορία και στη συνέχεια πρέπει να εξάγει πολύ καλές προβλέψεις. Υπάρχει λοιπόν μία ισορροπία ανάμεσα στην υποπροσαρμογή και την υπερπροσαρμογή του μοντέλου.

παραδείγματα Underfitted και overfitted μοντέλων



Underfitted Good Fit/Robust Overfitted

Τροποι αντιμετωπισης overfitting

- Πρόωρη διακοπή εκπαίδευσης (early stopping)
- Ομαλοποίηση (regularization) (Περιορισμός των δυνατών τιμών παραμέτρων που μπορεί να πάρει το μοντέλο)
- Dropout (Αποφυγή υπερεκπαίδευσης) στα πλήρως διασυνδεδεμένα επίπεδα

Dataset

Το dataset που χρησιμοποιήσαμε είναι το https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

- Οι εικονες εχουν μεγέθη απο 29 μεχρι 31 pixels.
- 43 κατηγορίες εικόνων.

- Σύνολο δεδομένων
- 39209 training set
    - 31367 training (80%)
    - 7842 validation (20%)
- 12630 test set

Preprocessing

- Μετατροπή όλων των φωτογραφιών σε 30X30 pixels
- Μετατροπή κάθε φωτογραφίας σε array και αποθήκευση της στη λίστα append
- Αντιστοίχιση της φωτογραφίας στην κατηγορία που ανήκει στην λίστα labels
- Μετατροπή των τιμών σε float32 και κανονικοποίηση από [0-255] σε [0,1]
- Μετατροπή των labels σε one-hot διανύσματα

Αρχιτεκτονικές

Σε κάθε μοντέλο δοκιμάσαμε διαφορετικά learning rates l=[0.01,0.005,0.001,0.0005,0.0001], σαν αλγόριθμο βελτιστοποίησης τον adam και σαν συνάρτηση απώλειας την categorical cross entropy.

Σε κάθε επίπεδο εξόδου χρησιμοποιήσαμε την softmax που είναι μια

γενίκευση της logistic για παραπάνω από δύο κατηγορίες με 43 εξόδους που είναι ο αριθμός των πινακίδων. Εκπαιδεύσαμε κάθε αρχιτεκτονική σε 150 εποχές και χρησιμοποιήσαμε σαν callback function την early stopping που ουσιαστικά όταν αρχίσει να αποκλίνει η απώλεια επικύρωσης (validation loss) με την απώλεια εκπαίδευσης(training loss) θα σταματά μετά από 20 εποχές(patience=20)

Αρχιτεκτονική 1

Επίπεδο εισόδου:30x30x3

2 συνελικτικά επίπεδα με : 3x3 φίλτρα στο καθένα , με 16, 32 φίλτρα αντίστοιχα ακολουθούμενα από ένα επίπεδο maxpooling 2:1.

Επίπεδο εξόδου: Στην είσοδο του εφαρμόζεται η flatten και έχουμε ένα επίπεδο εξόδου που έχει 43 εξόδους με activation function την softmax.

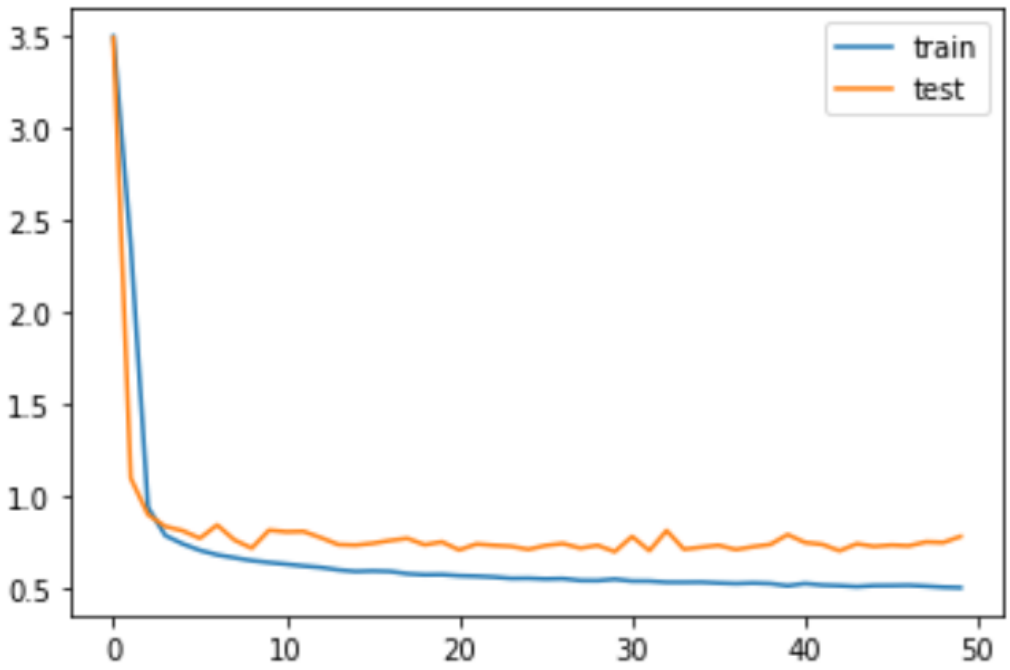
Αρχιτεκτονική 1

Layer (type)	Output Shape	Param #
conv2d_20 (Conv2D)	(None, 30, 30, 16)	448
max_pooling2d_20 (MaxPooling	(None, 15, 15, 16)	0
conv2d_21 (Conv2D)	(None, 15, 15, 32)	4640
max_pooling2d_21 (MaxPooling	(None, 7, 7, 32)	0
flatten_10 (Flatten)	(None, 1568)	0
dense_10 (Dense)	(None, 43)	67467

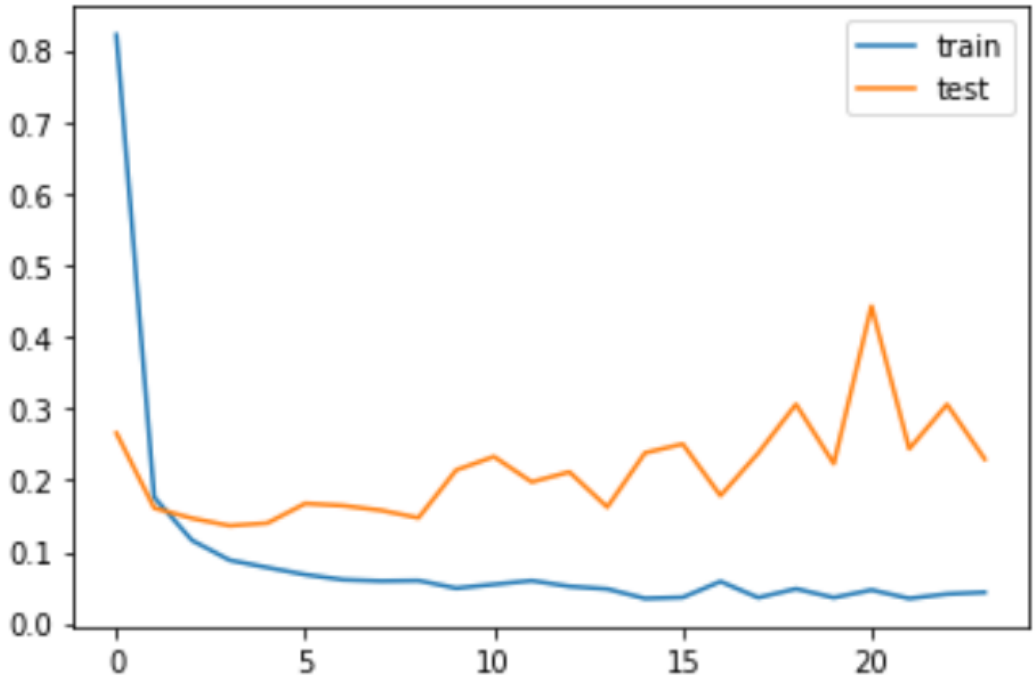
Total params: 72,555
Trainable params: 72,555
Non-trainable params: 0

Παρακάτω βλέπουμε τις διακυμάνσεις του training loss και του Validation loss στις εποχές που έτρεξε κάθε αρχιτεκτονική 1 με τα διαφορετικά Learning Rates:

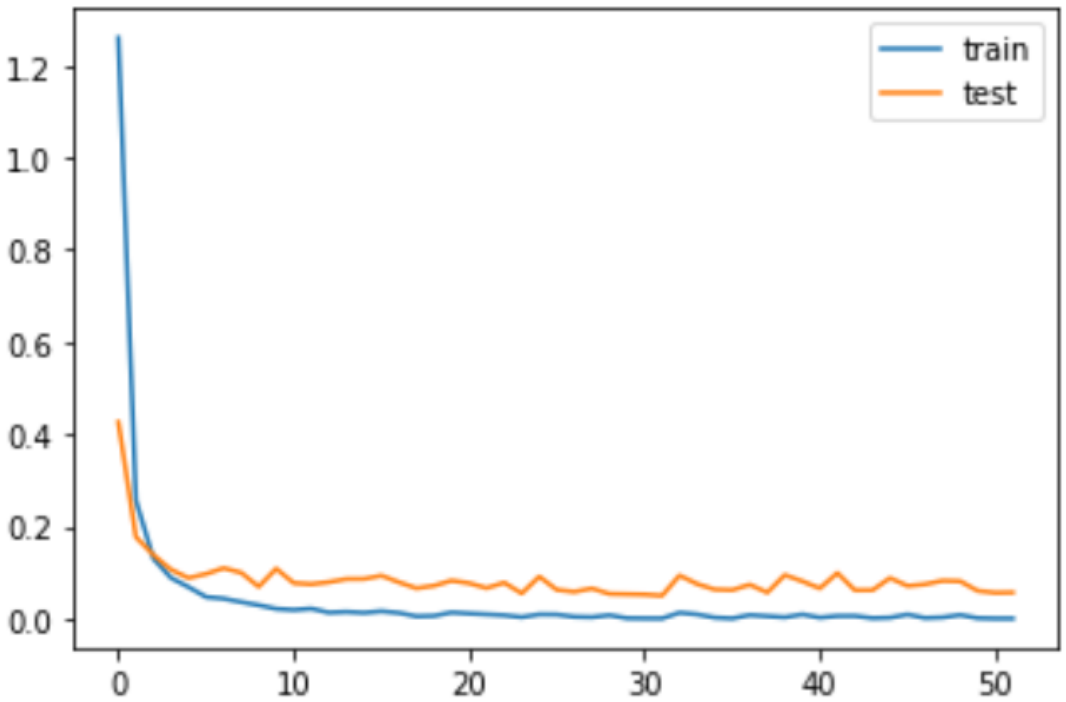
αποτελέσματα για learning rate = 0.01 σταμάτησε μετά από 50 εποχές



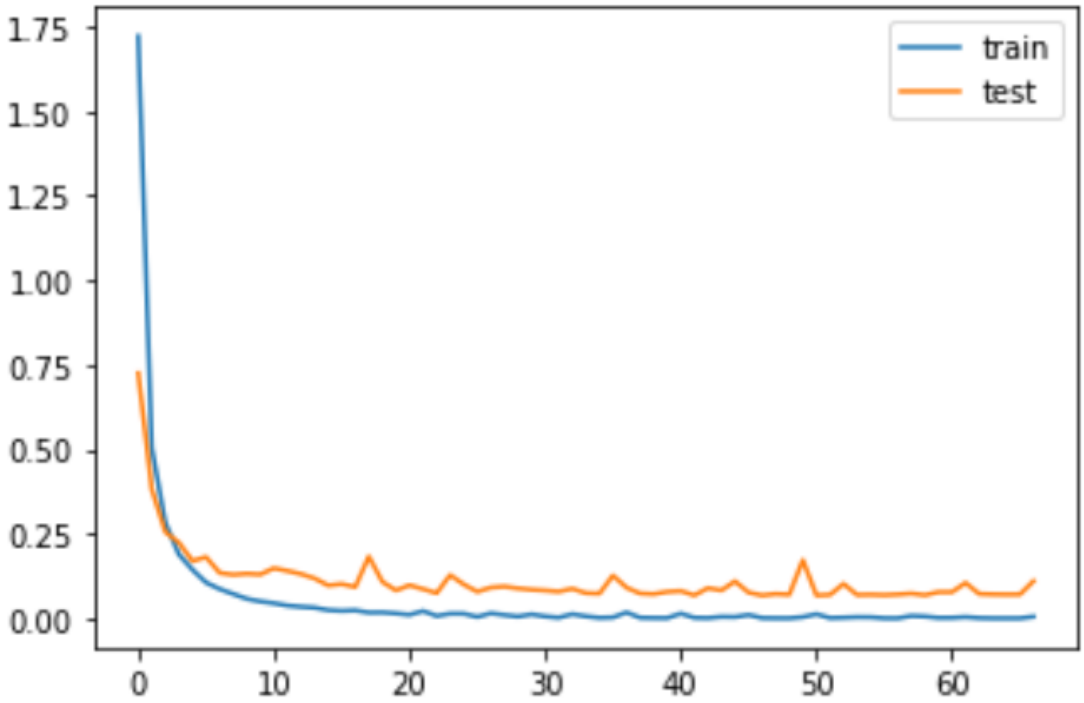
αποτελέσματα για learning rate = 0.005 σταμάτησε μετά από 24 εποχές



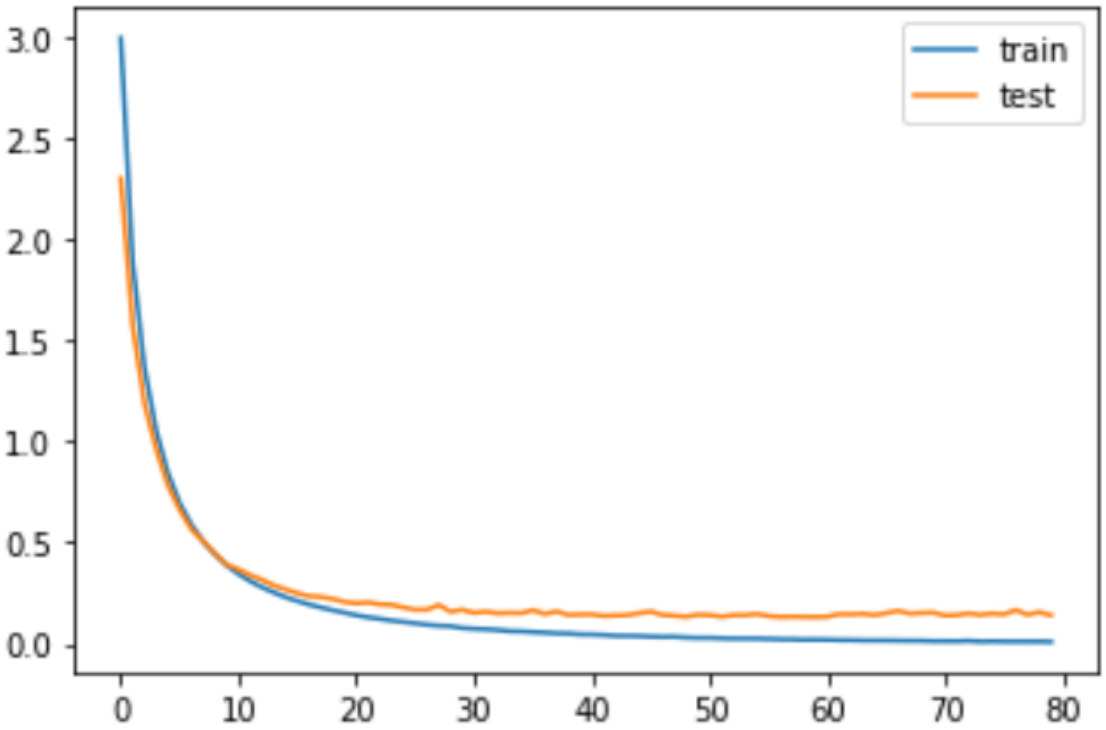
αποτελέσματα για learning rate = 0.001 σταμάτησε μετά από 52 εποχές



αποτελέσματα για learning rate = 0.0005 σταμάτησε μετά από 67 εποχές



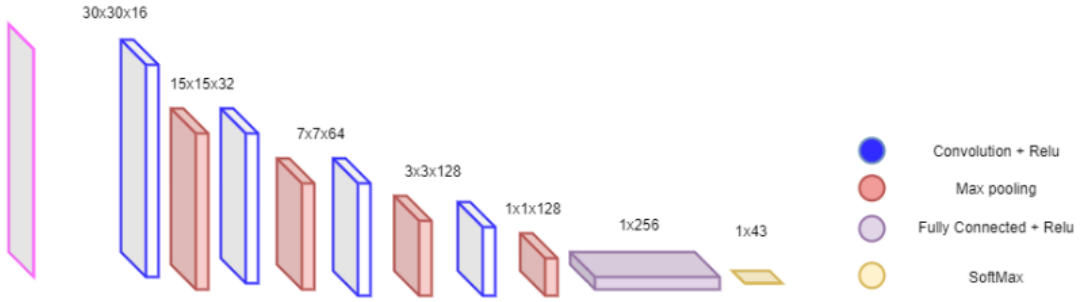
αποτελέσματα για learning rate = 0.0001 σταμάτησε μετά από 80 εποχές



Αρχιτεκτονική 2

Κρατήσαμε τα επίπεδα της αρχιτεκτονικής 1 και προσθέσαμε στα ενδιάμεσα επίπεδα ένα συνελικτιό με 2 συνελικτικά επιπλέον με 64 και 128 φίλτρα αντίστοιχα καθώς και ένα πλήρως διασυνδεδεμένο επίπεδο 1x256 με activation function την Relu πρίν το επίπεδο εξόδου της Softmax.

Αρχιτεκτονική 2



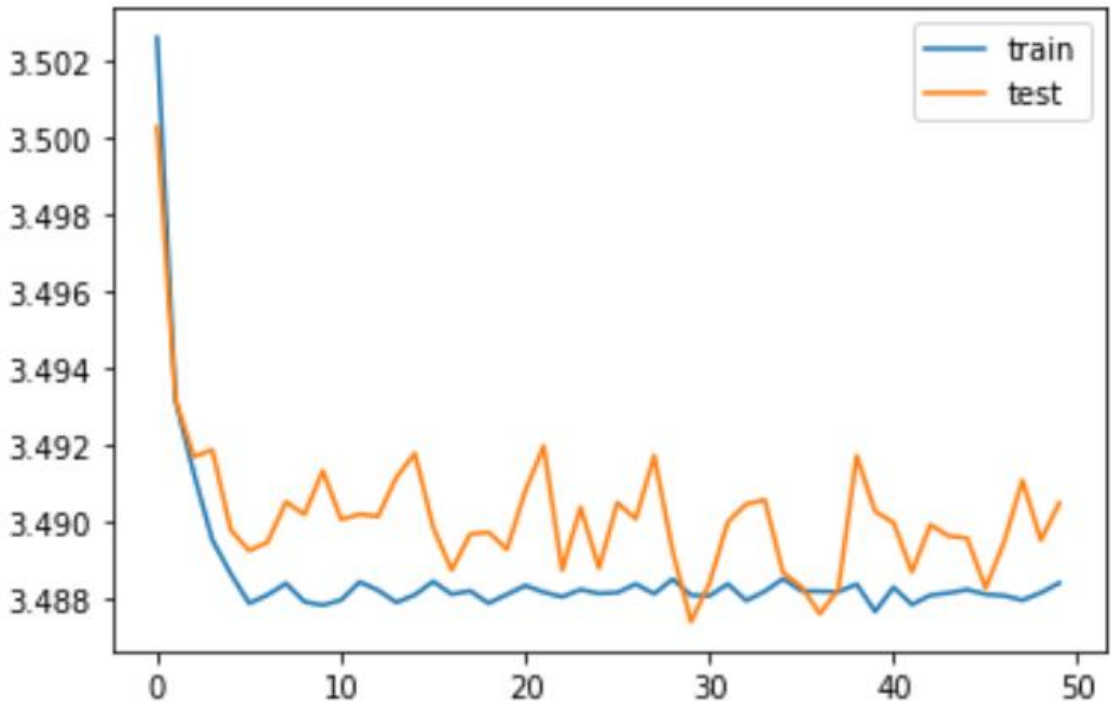
Layer (type)	Output Shape	Param #
conv2d_30 (Conv2D)	(None, 30, 30, 16)	448
max_pooling2d_30 (MaxPooling	(None, 15, 15, 16)	0
conv2d_31 (Conv2D)	(None, 15, 15, 32)	4640
max_pooling2d_31 (MaxPooling	(None, 7, 7, 32)	0
conv2d_32 (Conv2D)	(None, 7, 7, 64)	18496
max_pooling2d_32 (MaxPooling	(None, 3, 3, 64)	0
conv2d_33 (Conv2D)	(None, 3, 3, 128)	73856
max_pooling2d_33 (MaxPooling	(None, 1, 1, 128)	0
flatten_15 (Flatten)	(None, 128)	0
dense_15 (Dense)	(None, 256)	33024
dense_16 (Dense)	(None, 43)	11051

Total params: 141,515

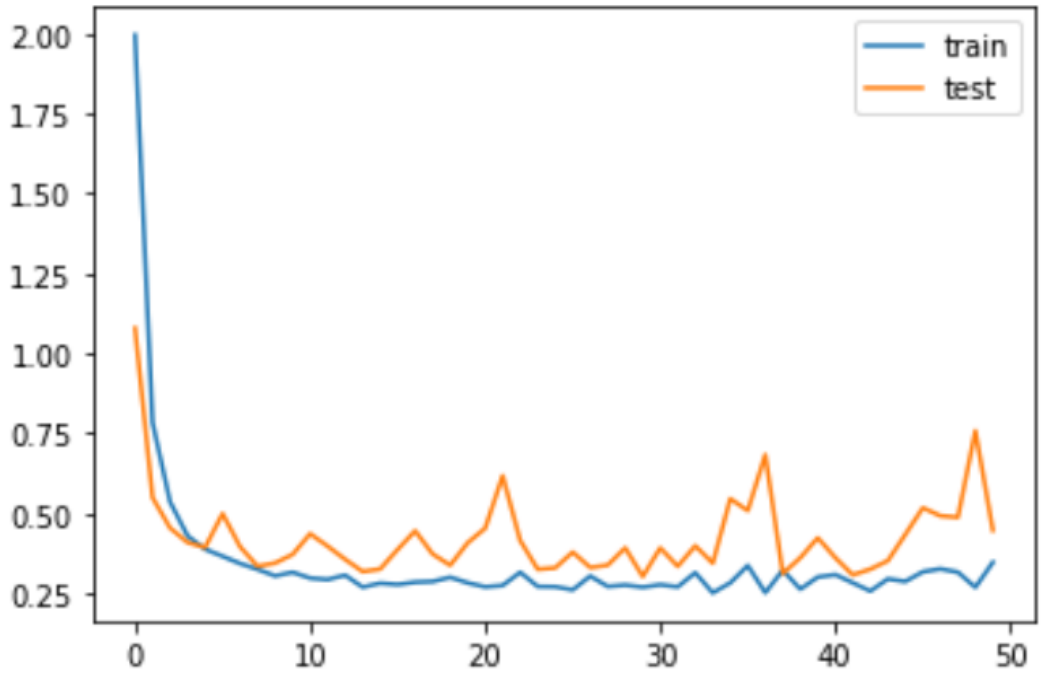
```
Trainable params: 141,515
Non-trainable params: 0
```

Παρακάτω βλέπουμε τις διακυμάνσεις του training loss και του Validation loss στις εποχές που έτρεξε κάθε αρχιτεκτονική 2 με τα διαφορετικά Learning Rates:

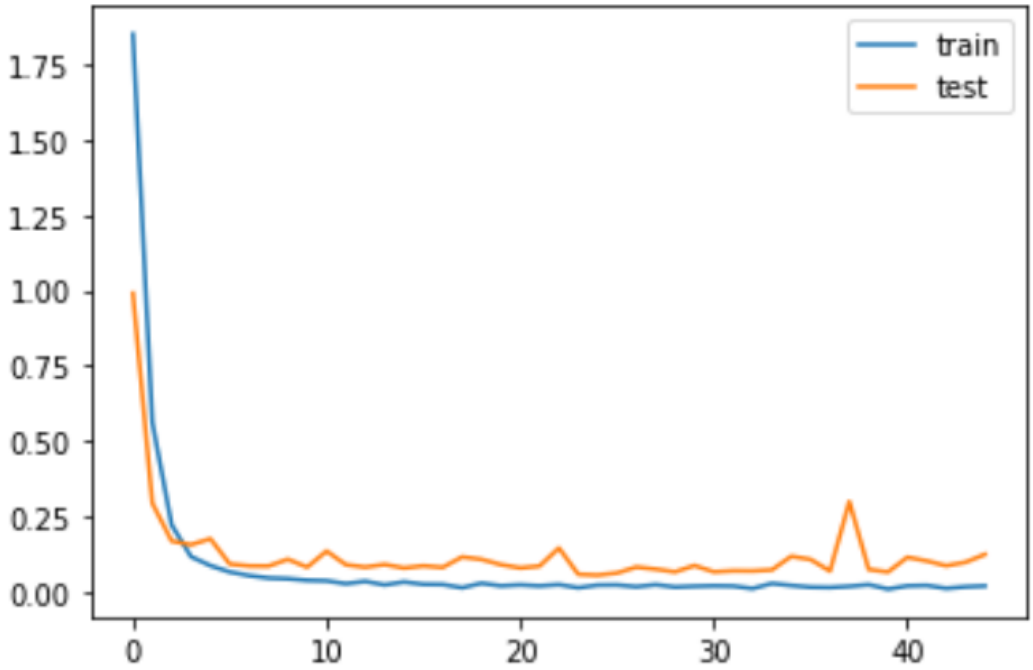
αποτελέσματα για learning rate = 0.01 σταμάτησε μετά από 50 εποχές



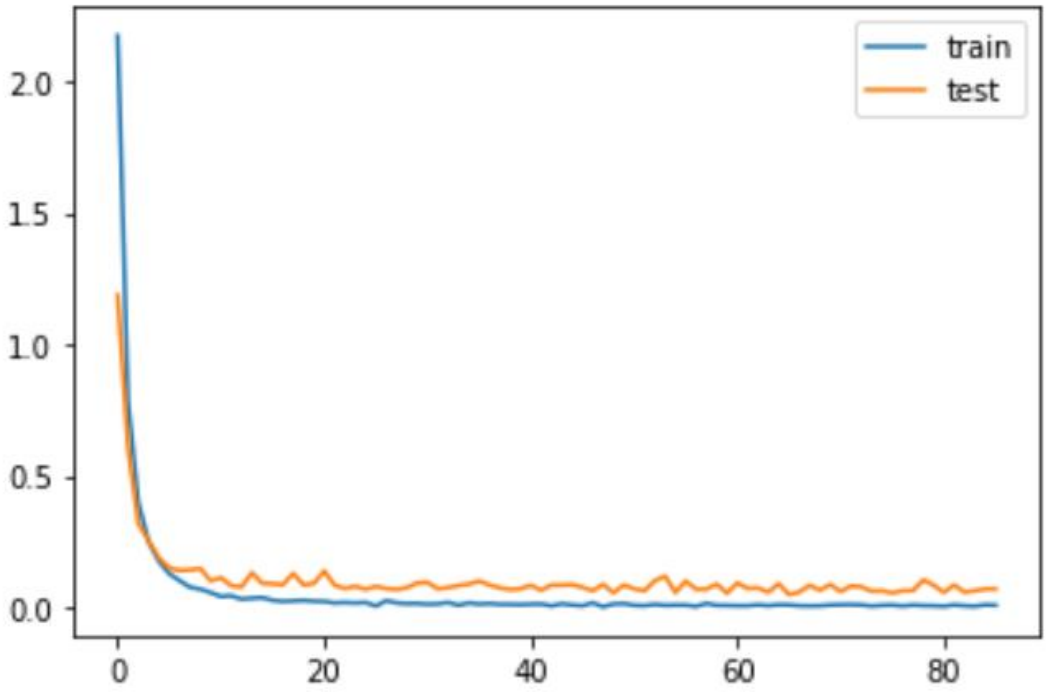
αποτελέσματα για learning rate = 0.005 σταμάτησε μετά από 50 εποχές



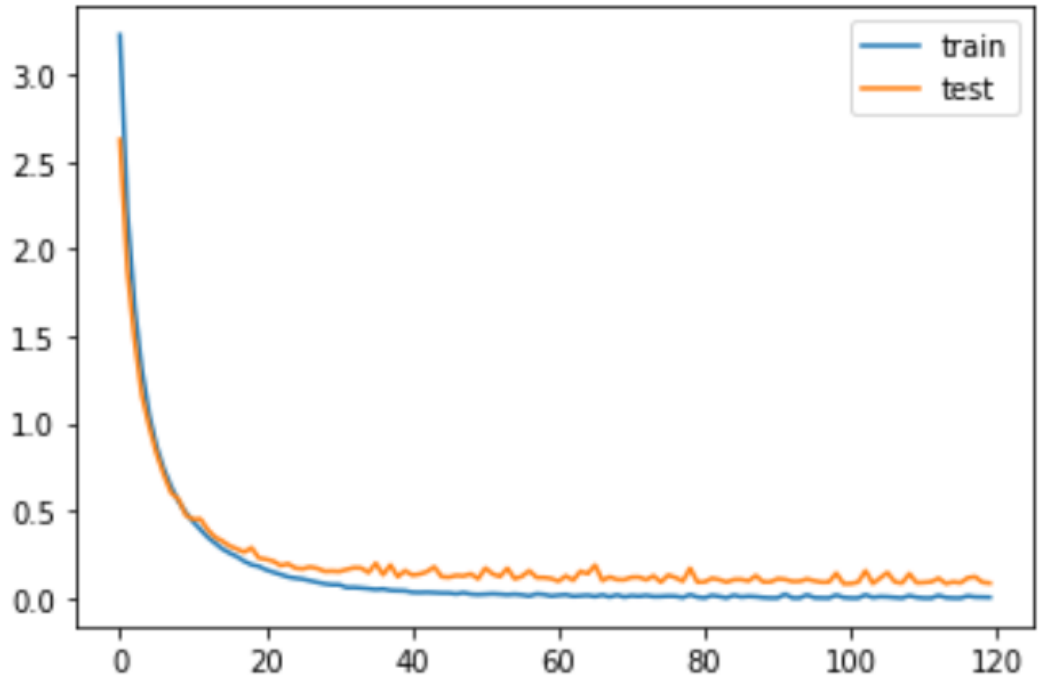
αποτελέσματα για learning rate = 0.001 σταμάτησε μετά από 45 εποχές



αποτελέσματα για learning rate = 0.0005 σταμάτησε μετά από 86 εποχές



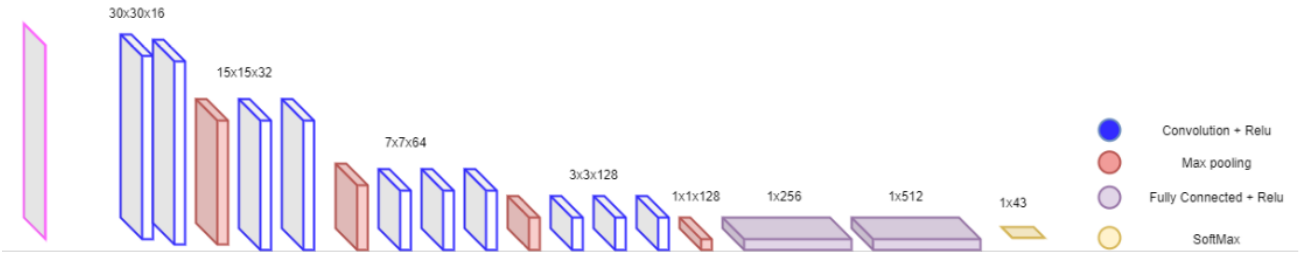
αποτελέσματα για learning rate = 0.0001 σταμάτησε μετά από 120 εποχές



Αρχιτεκτονική 3

Προσπαθήσαμε να προσομοιώσουμε την αρχιτεκτονική VGG προσθέτοντας και άλλο βάθος στο δίκτυο μας στα ενδιάμεσα επίπεδα.
2 διαδοχικά συνελικτικά με 16 φίλτρα 3x3 ακολουθούμενα από ένα maxpooling επίπεδο 2:1
2 διαδοχικά συνελικτικά με 32 φίλτρα 3x3 ακολουθούμενα από ένα maxpooling επίπεδο 2:1
3 διαδοχικά συνελικτικά με 64 φίλτρα 3x3 ακολουθούμενα από ένα maxpooling επίπεδο 2:1
3 διαδοχικά συνελικτικά με 128 φίλτρα 3x3 ακολουθούμενα από ένα maxpooling επίπεδο 2:1
Το επίπεδο flatten και 2 διαδοχικά πλήρως διασυνδεδεμένα επίπεδα (Dense) 1x256 και 1x512 αντίστοιχα.

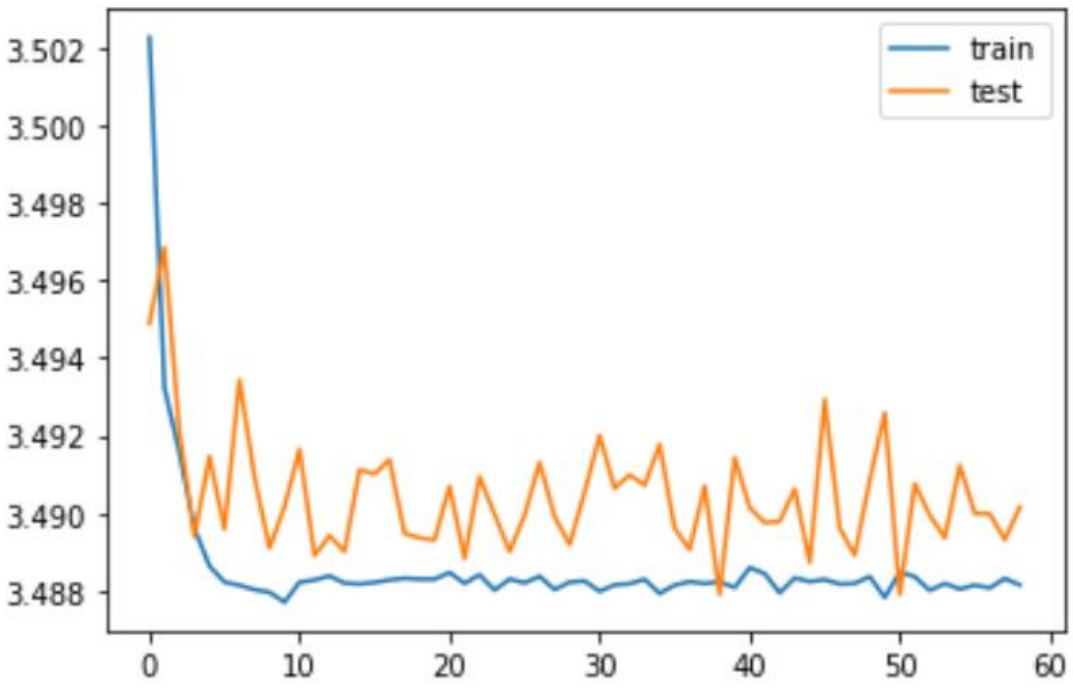
Αρχιτεκτονική 3



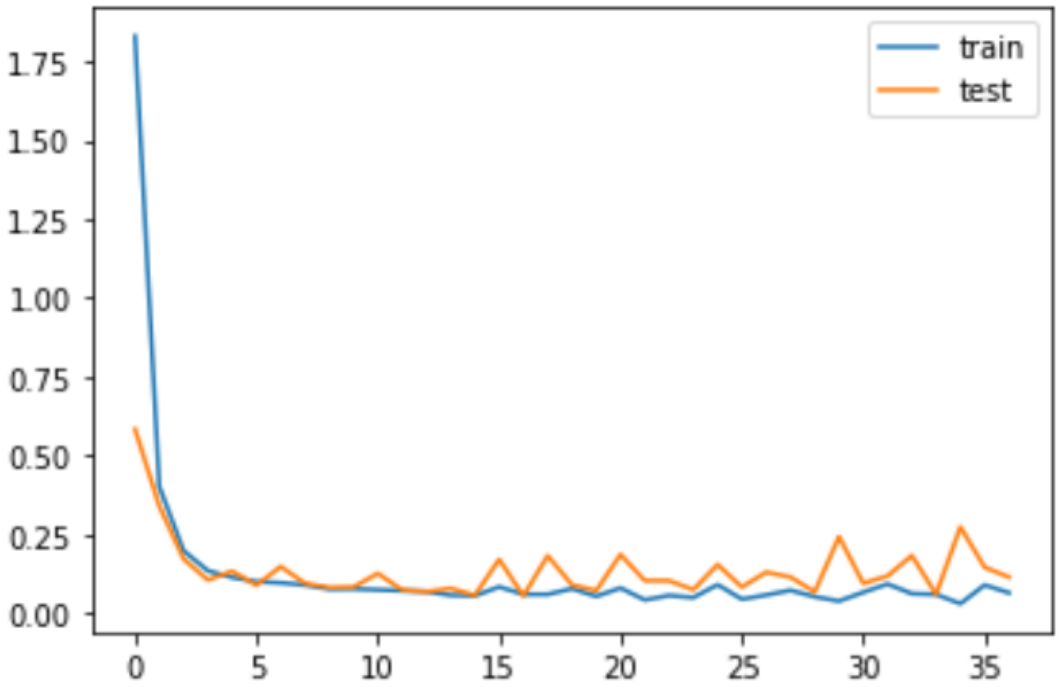
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 16)	448
conv2d_1 (Conv2D)	(None, 30, 30, 16)	2320
max_pooling2d (MaxPooling2D)	(None, 15, 15, 16)	0
conv2d_2 (Conv2D)	(None, 15, 15, 32)	4640
conv2d_3 (Conv2D)	(None, 15, 15, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 7, 7, 32)	0

```
conv2d_4 (Conv2D)            (None, 7, 7, 64)          18496
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 7, 7, 64)          36928
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 7, 7, 64)          36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 3, 3, 64)          0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 3, 3, 128)         73856
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 3, 3, 128)         147584
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 3, 3, 128)         147584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 1, 1, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 256)               33024
_________________________________________________________________
dense_1 (Dense)              (None, 43)                11051
=================================================================
Total params: 522,107
Trainable params: 522,107
Non-trainable params: 0
```

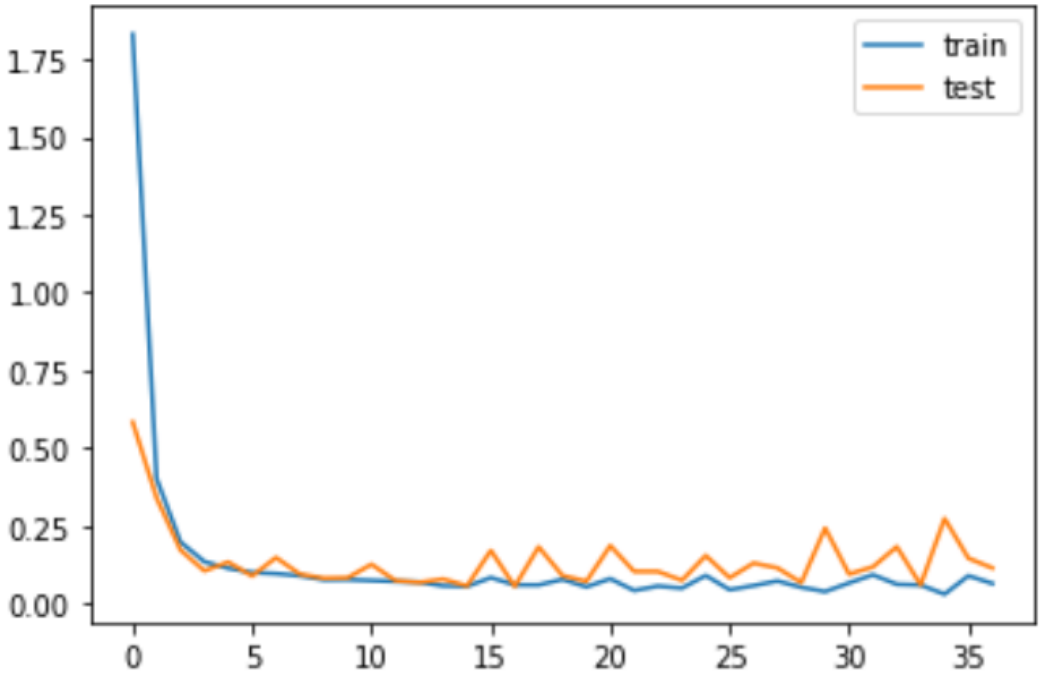
αποτελέσματα για learning rate = 0.01 σταμάτησε μετά από 48 εποχές



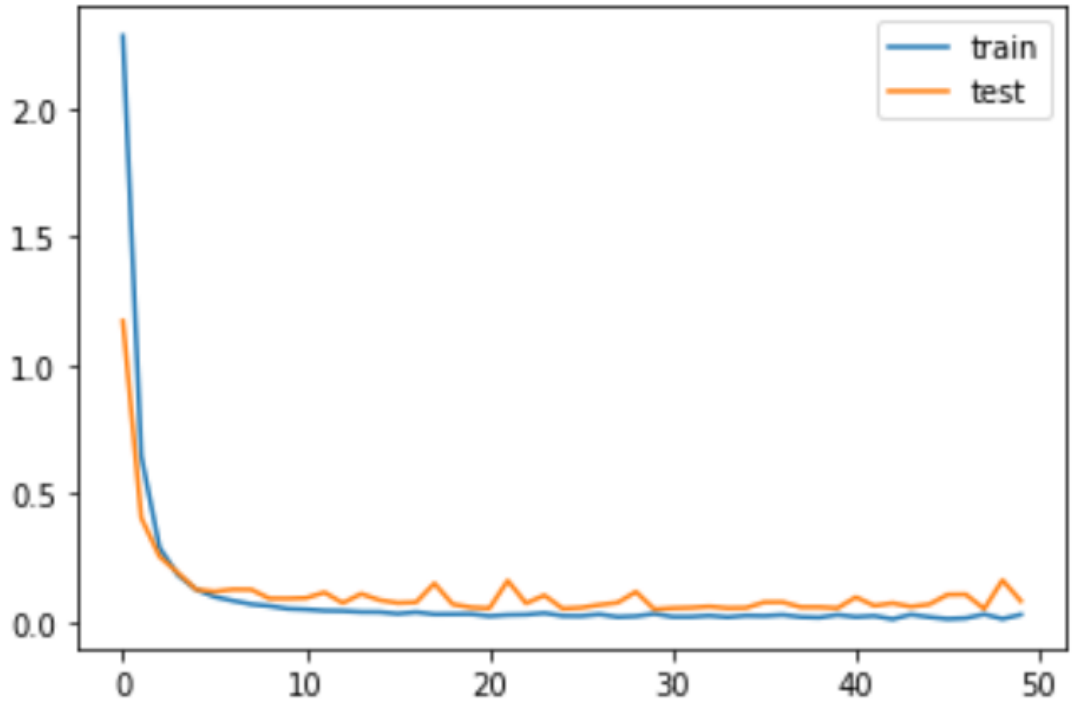
αποτελέσματα για learning rate = 0.005 σταμάτησε μετά από 37 εποχές



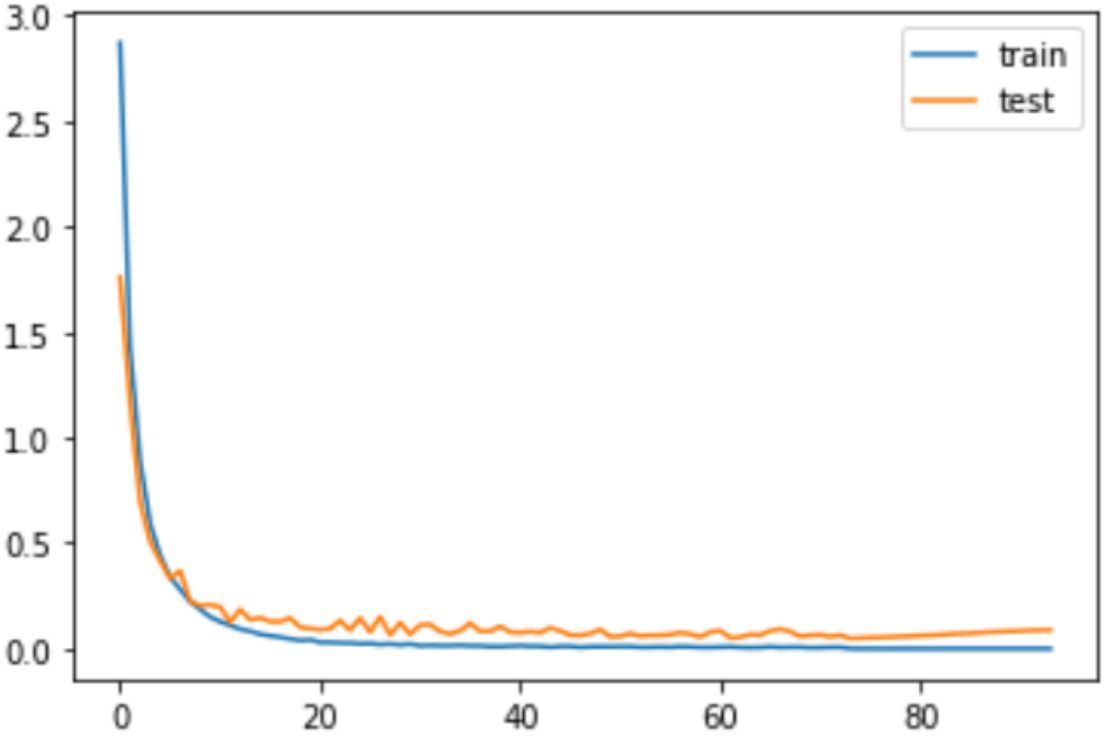
αποτελέσματα για learning rate = 0.001 σταμάτησε μετά από 37 εποχές



αποτελέσματα για learning rate = 0.0005 σταμάτησε μετά από 50 εποχές



αποτελέσματα για learning rate = 0.0001 σταμάτησε μετά από 94 εποχές



Αποτελέσματα εκτιμήσεων (Evaluation)

Σαν μετρική χρησιμοποιούμε το accuracy για την αξιολόγηση των αρχιτεκτονικών μας. Τα αποτελέσματα φαίνονται στον παρακάτω πίνακα

L/R	Αρχιτεκτονικη 1	Αρχιτεκτονικη 2	Αρχιτεκτονικη 3
0.01	71,99%	5.93%	5.16%
0.005	91.39%	83.31%	5.70%
0.001	93.15%	91.35%	93.99%
0.0005	90.16%	91.87%	**94.63%**
0.0001	88.54%	87.07%	94.06%

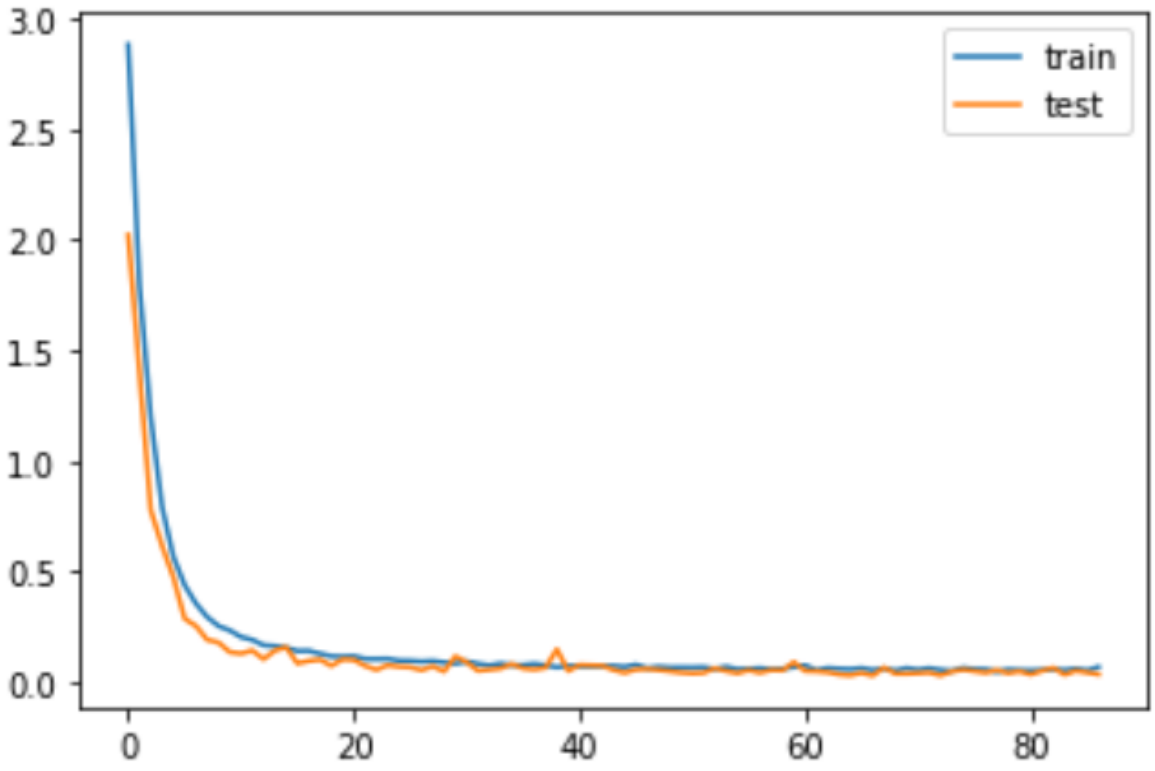
Παρατηρούμε ότι όσο μεγαλώνει το βάθος αν χρησιμοποιήσουμε μεγάλο ρυθμό εκμάθησης βγάζουμε δεν έχουμε καλά αποτελέσματα. Το μεγαλύτερο accuracy το έδωσε η αρχιτεκτονική 3 με ρυθμό εκμάθησης 0,0005.

Τελική αρχιτεκτονική

Τέλος πήραμε την αρχιτεκτονική με το μεγαλύτερο accuracy και κάναμε επαύξηση δεδομένων. Έτσι ώστε να εκπαιδευτεί το ΣΝΔ σε διαφορετικές εικόνες τροποποιημένες ώστε να βγάλει καλύτερα αποτελέσματα.

```
aug = ImageDataGenerator(
    rotation_range=10,
    zoom_range=0.15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    vertical_flip=True
    )
```

αποτελέσματα για learning rate = 0.0005 σταμάτησε μετά από 87 εποχές



Το αποτέλεσμα που έβγαλε είναι **96.03%** accuracy που ήταν χωρίς την επαύξηση στο 94.63%.

Δοκιμή σε πραγματικά δεδομένα

Τέλος δοκιμάσαμε το ΣΝΔ σε 13 πραγματικές φωτογραφίες από δρόμους της Γερμανίας εκτός του Dataset που χρησιμοποιήσαμε για training και testing για να δούμε κατά πόσο καλά μπορεί να κατηγοριοποιήσει η αρχιτεκτονική μας και άλλου είδους εικόνες.

παράδειγμα φωτογραφίας που χρησιμοποιήσαμε

Τα αποτελέσματα που βγάλαμε:

Προέβλεψε στις 12/13 εικόνες σωστά την πραγματική κατηγορία την οποία ανήκουν 92.85% accuracy.Το οδικό σήμα που δεν βρήκε σωστά ήταν αυτό της υποχρεωτικής πορείας δεξιά που το κατηγοριοποίησε στην κατηγορία υποχρεωτική πορεία ευθεία. Βέβαια δεν σημαίνει κάτι αυτό καθώς το δείγμα μας ήταν πολύ μικρό. Το θετικό είναι ότι βρήκε τα οδικά σήματα και ας είχαν καλύτερη ανάλυση από τις φωτογραφίες του dataset μας , ήταν και από διαφορετικές γωνίες λήψης και μερικές είχαν και θόρυβο όπως το χιονισμένο STOP.

Επεκτάσεις της εργασίας:

Θα μπορούσαμε να χρησιμοποιήσουμε και άλλες αρχιτεκτονικές όπως το ResNet. Καθώς και να κάνουμε ανίχνευση των πινακίδων και να τις κατηγοριοποιούμε χρησιμοποιόντας αλγορίθμους για detection (πχ YOLO)

Κώδικας

https://github.com/tziojo/tn_ergasia

Αναφορές

1. https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/11551/Diamantis_MPPL13016.pdf?sequence=1&isAllowed=y
2. https://dspace.lib.uom.gr/bitstream/2159/24683/1/KafaliEfthimiaMsc2020.pdf
3. V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning” ArXiv160307285 Cs Stat, Mar. 2016.
4. https://users.ece.cmu.edu/~koopman/pubs/wagner15_philosphy_self_driving_cars.pdf

5. http://www.transport.ntua.gr/wp-content/uploads/dtd949-GoliasKonstantinos-1.pdf