# Εισαγωγή στα Νευρωνικά Δίκτυα

Η ιδέα των τεχνητών νευρωνικών δικτύων (Artificial Neural Networks) άρχισε να αναπτύσσεται την δεκαετία του ‘50 από τον Frank Rosenblatt, με επιρροές από προηγούμενες έρευνες των Warren McCulloch και Walter Pitts, και αναφέρονταν ως perceptrons. Τα νευρωνικά δίκτυα είναι υπολογιστικά συστήματα εμπνευσμένα από τα βιολογικά νευρωνικά δίκτυα από τα οποία αποτελείται ο εγκέφαλος, και αποτελούν μια αρκετά απλουστευμένη εκδοχή αυτού. Αυτά τα συστήματα “μαθαίνουν” να εκτελούν ενέργειες για τις οποίες δεν έχουν άμεσα προγραμματιστεί, βασιζόμενα σε ένα πλήθος παραδειγμάτων που έχει αρχικά δοθεί.

Τα ANNs διαφοροποιούνται ανάλογα με την δομή και την λογική με την οποία υλοποιούνται. Στην παρούσα εργασία αναλύονται και χρησιμοποιούνται δυο διαφορετικές αρχιτεκτονικές νευρωνικών δικτύων, το dense (fully connected) neural network και το convolutional neural network.

# Dense Neural Networks

Τα dense neural network αποτελλούν την βασικότερη εκδοχή των νευρωνικών δικτύων. Η αρχιτεκτονική τους είναι fully connected, δηλαδή κάθε νευρώνας ενός layer συνδέεται με όλους τους νευρώνες των γειτονικων layers.



Η γενική ιδέα της λειτουργίας ενός τέτοιου δικτύο αποτελείται από:

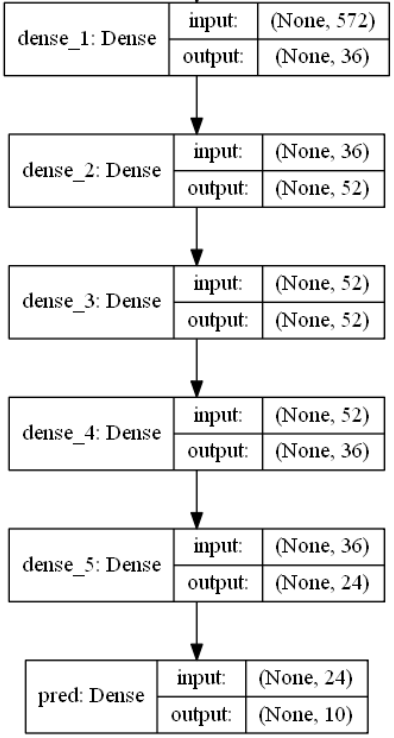
* Feed-forward: Το δίκτυο δέχεται σαν είσοδο το dataset με το οποίο θα εκπαιδευτεί και υπολογίζει την έξοδο του μέσω των παραμέτρων του δικτύου. Η είσοδος του δικτύου αρχικά υπόκειται σε μια προ-επεξεργασία ώστε η μορφή των δεδομένων να ικανοποιεί τις ανάγκες του νευρωνικού. Οι παράμετροι αποτελλούνται από τις ακμές (weights) και τους νευρώνες (biases) ανα layer. Πάνω σε αυτές θα εφαρμοστεί η εκπαίδευση, και μαζί με την συνάρτηση ενεργοποίησης σε κάθε εξόδο του κάθε layer, καταλήγουμε στην έξοδο του δικτύου. Η έξοδος είναι και η εκτίμηση (predict) της αρχικής εισόδου.
* Cost function: Στην έξοδο του δικτύου, κατά την φάση της εκπαίδευσης, εφαρμόζεται μια συνάρτηση κόστους η οποία υπολογίζει το σφάλμα της εκτίμησης του δικτύου για όλα τα δεδομένα με τα οποία εκπαιδεύεται.
* Back propagation: Με την τεχνική της οπισθοδιάδοσης, και βάση του σφάλματος το οποίο βρέθηκε στο προηγούμενο στάδιο, οι παράμετροι “ρυθμίζονται” και η όλη διαδικασία επαναλαμβάνεται έως το σφάλμα να ελαχιστοποιηθεί. Κάθε τέτοια επανάληψη ορίζεται ως epoch. Στην οπισθοδιάδοση (όπως και στην συνάρτηση σφάλματος) μπορούν να υλοποιηθούν αρκετές εναλλακτικές τεχνικές, όπως Gradient Descent, Adam, RMSprop κτλ.

Στην εργασία υλοποιήθηκαν αρκετές παραλλαγές νευρωνικών δικτύων μέχρι να καταλήξουμε στην κατάλληλη αρχιτεκτονική. Το κάθε ένα από αυτά δέχεται σαν είσοδο το κάθε αρχείο ήχου από ένα dataset με πάνω από 23.000 audio clips. Τα δεδομένα αυτά του κάθε αρχείου, πριν την είσοδο τους στο νευρωνικό δίκτυο, πέρασαν από μια διαδικασία preprocessing, η οποία διαφοροποιείται ανάλογα με την αρχιτεκτονική του δικτύου. Στη συγκεκριμένη υλοποίηση εφαρμόστηκαν διάφοροι αλγόριθμοι για την προεπεξεργασία των δεδομένων όπως Fourier, MFCC (Mel-Frequency Cepstral Coefficients) και sampling, με την μεγαλύτερη επιτυχία εκπαίδευσης να επιτυγχάνεται μέσω του αλγορίθμου MFCC.

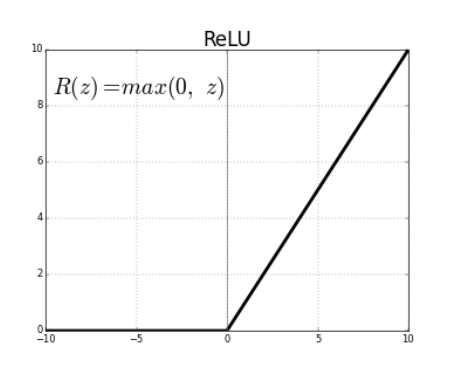
Οι δομές οι οποίες ερευνήθηκαν ποικίλουν στον αριθμό των νευρώνων ανά layer, στον αριθμό των layer, στην συνάρτηση κόστους αλλά και στον αλγόριθμο του back propagation.

Αναλυτικότερα, η τελική μορφή του νευρωνικού δικτύου αποτελλείται από:

* Input layer 572 νευρώνων, 5 hidden layers (36, 52, 52, 36 και 24 νευρώνες αντίστοιχα) και output layer 10 νευρώνων.

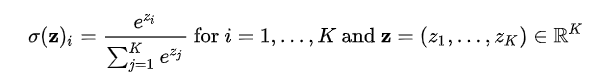


* Για την συνάρτηση ενεργοποίησης σε κάθε hidden layer κάναμε χρήση της ReLU, ενώ του output layer την softmax, καθώς η τελική έξοδος αποτελείται από 10 εξόδους (μία για κάθε ψηφίο).

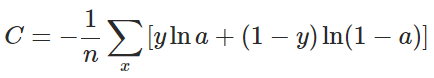


Γραφική απεικόνιση της συνάρτησης ενεργοποίησης ReLU

Η ReLU είναι μια υπολογιστικά εύκολη συνάρτηση ενεργοποίησης όπου είναι γραμμική για όλες τις θετικές τιμές εξόδου και μηδέν για όλες τις αρνητικές. Στο τελικό layer, εφαρμόστηκε η συνάρτηση ενεργοποίησης softmax, όπου δέχεται ένα vector K αριθμών (10 στην δική μας περίπτωση) και το κάνει normalize σε μια κατανομή πιθανοτήτων.



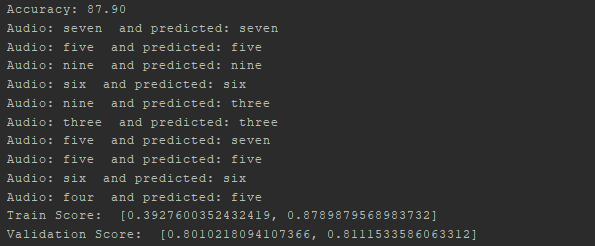
* Για την συνάρτηση κόστους χρησιμοποιήθηκε η cross-entropy.

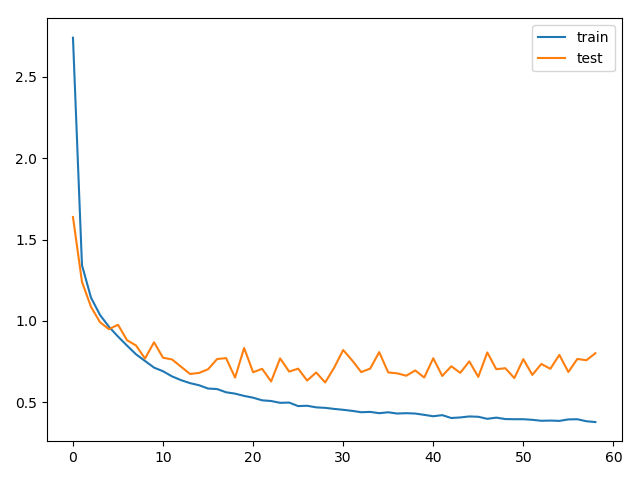


* Τέλος για την οπισθοδιάδοση κάναμε χρήση του αλγορίθμου RMSprop. Αντίστοιχα αποτελέσματα είχαμε και με την χρήση του αλγορίθμου Adam, που αποτελεί μια βελτιωμένη έκδοση του gradient descent.

Η διαδικασία μάθησης έχει οριστεί στα 100 epochs με δυνατότητα διακοπής εάν το accuracy δεν βελτιώνεται για πάνω από 10 epochs. Με αυτή την τεχνική διατηρούμε τις παραμέτρους της καλύτερης εποχής και μειώνουμε αισθητά τον χρόνο εκπαίδευσης του δικτύου.

Τα αποτελέσματα που είχαμε με είσοδο την διακριτοποίηση της συχνότητας των αρχείων ήχου στα 8000 δείγματα δεν ήταν τόσο καλά. Κάνοντας δοκιμές στην επεξεργασία των δεδομένων μέσω των αλγορίθμων Fourier και MFCC, είχαμε καλύτερα αποτελέσματα. Συγκεκριμένα, κάνοντας preprocessing με τον αλγόριθμο MFCC, πετύχαμε accuracy του train στο 87% και του test στο 81% με προοπτικές βελτίωσης.

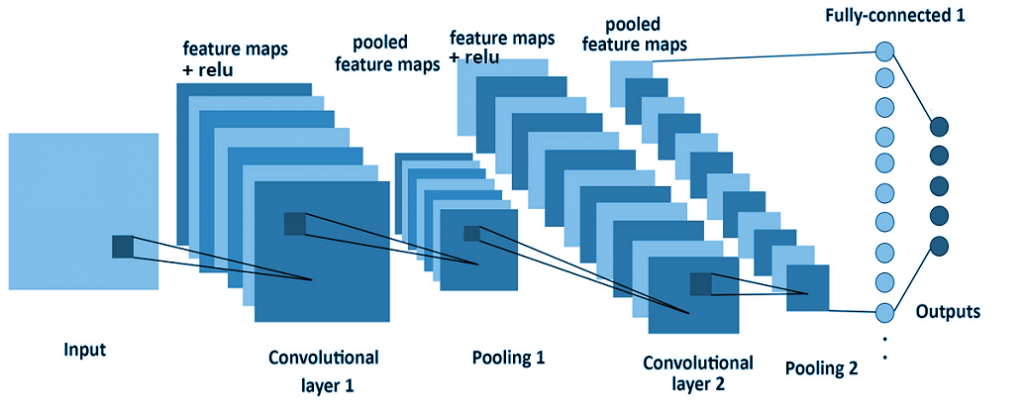




Γραφική απεικόνιση σφάλματος κατά την διάρκεια της εκπαίδευσης σε train και test data

# Convolutional Neural Network

Μετά από έρευνα πάνω στην ανάλυση ήχου καταλήξαμε ότι, χωρίς το κατάλληλο preprocessing, η επιλογή ενός dense νευρωνικού δικτύου για αναγνώριση ψηφίων δεν είναι αρκετά αποτελεσματική. Έτσι, δοκιμάσαμε το convolutional neural network, μια διαφορετική αρχιτεκτονική δικτύου που χρησιμοποιείται ιδιαίτερα στην ανάλυση και αναγνώριση εικόνας.

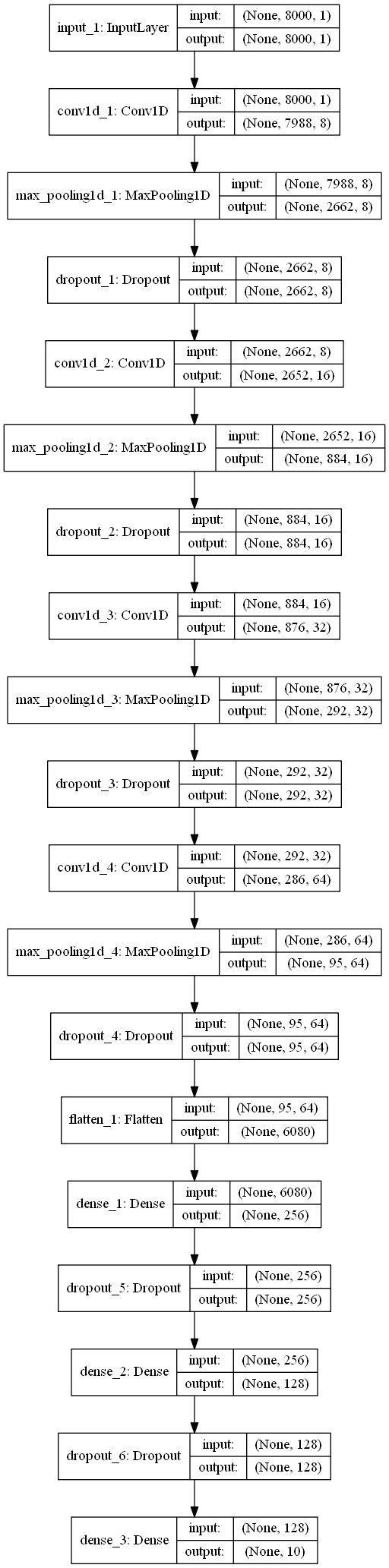


Μια βασική διαφορά των convolutional neural network από τα κλασσικά feed forward νευρωνικά είναι η ανοχή τους στα μη προ-επεξεργασμένα δεδομένα. Χωρίς δηλαδή το κατάλληλο preprocessing, τα convolutional καταφέρνουν και ανιχνεύουν τα χαρακτηριστικά των δεδομένων. Αυτό γίνεται τροποποιώντας τα δεδομένα σε μορφή πολύ πιο εύκολη προς επεξεργασία χωρίς να χάνουν σημαντικά features των δεδομένων, κάτι που είναι πολύ σημαντικό για την πρόβλεψη. Μετά την επεξεργασία αυτών των δεδομένων, καταλήγουμε σε ένα dense νευρωνικό που κάνει και την τελική πρόβλεψη των δεδομένων.

Η βασική αρχιτεκτονική των συνελικτικών (convolutional) νευρωνικών δικτύων αναλύεται σε:

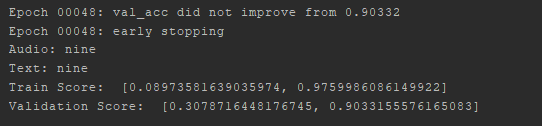
* Convolutional layer: Γίνεται συνέλιξη της εισόδου με φίλτρα τα οποία έχουν προκύψει από την διαδικασία εκπαίδευσης.
* Activation layer: Γίνεται η γραμμικοποίηση της εξόδου του convolutional layer (ReLU).
* Pooling layer: Σε αυτό το στάδιο γίνεται υπο-δειγματοληψία των δεδομένων συνοψίζοντας τις εξόδους γειτονικών γκρουπ νευρώνων εντός ενός παραθύρου με μια αντιπροσωπευτική τιμή, μειώνοντας έτσι το μέγεθος των δεδομένων. Οι επικρατέστερες κατηγορίες του pooling είναι το max, sum και average. Στην παρούσα εργασία χρησιμοποιείται η max.
* Fully connected layer: Κάθε νευρώνας του προηγούμενου layer ενώνεται με το επόμενο και στην συνέχεια παράγονται οι έξοδοι του νευρωνικού.

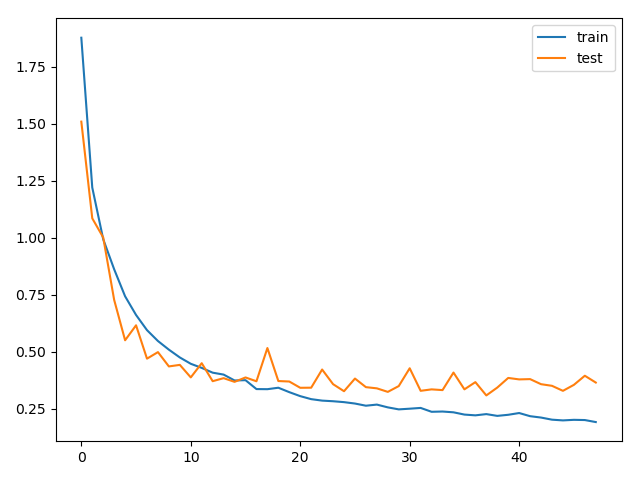
Η αρχιτεκτονική που χρησιμοποιήθηκε για το convolutional νευρωνικό δίκτυο είναι η ακόλουθη:



Η επιλογή αυτής της αρχιτεκτονικής βοήθησε πολύ και στην αντιμετώπιση του overfitting. Κατά την υλοποίηση του fully-connected νευρωνικού, το πρόβλημα του overfitting παρουσιάστηκε αρκετές φορές λόγω της μη επιθυμητής αναλογίας των features των δεδομένων (8000) με τον αριθμό των δεδομένων για εκπαίδευση (23000). Κάνοντας και την χρήση του preprocessing που προαναφέρθηκε, το overfitting υπήρχε ακόμα και με τεχνικές αντιμετώπισης του όπως regularization και dropouts σε κάθε layer. Τα convolutional και pooling layers ανιχνεύουν τα high-level features της εισόδου ώστε αυτό το πρόβλημα να αποφεύγεται με επιτυχία.

Τα αποτελέσματα του σε σχέση με το dense είναι αρκετά ικανοποιητικά, καθώς φτάνουμε accuracy στο train της τάξης του 97% και 90% στο test. Η διαδικασία μαθησης είχε οριστεί στα 100 epochs, παρ’όλα αυτα έφτασε το μεγαλύτερο accuracy στα 48 epochs.





Γραφική απεικόνιση σφάλματος κατά την διάρκεια της εκπαίδευσης σε train και test data