



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Δημιουργία με χρήση Python εργαλείου στο οποίο ενσωματώνονται μέθοδοι που αναπτύσσονται στο βιβλίο “Deep learning with python”.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΙΑΚΟΥΜΠ ΓΙΑΝ ΛΑΠΙΝΣΚΙ

A.M. 71345524

**Επιβλέπων
καθηγητής:**

**Πάρις Μαστοροκώστας
Καθηγητής ΠΑ.Δ.Α**

Αθήνα, Ιούλιος 2021



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Δημιουργία με χρήση Python εργαλείου στο οποίο ενσωματώνονται μέθοδοι που αναπτύσσονται στο βιβλίο “Deep learning with python”.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΙΑΚΟΥΜΠ ΓΙΑΝ ΛΑΠΙΝΣΚΙ

A.M. 71345524

**Επιβλέπων
καθηγητής:**

**Πάρις Μαστοροκώστας
Καθηγητής ΠΑ.Δ.Α**

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 13^η Ιουλίου του 2021.

.....
Ιωάννης Αμοργίνος
Λέκτορας ΠΑ.Δ.Α.

.....
Κωνσταντίνος Κουκουλέτσος
Καθηγητής ΠΑ.Δ.Α.

.....
Πάρις Μαστοροκώστας
Καθηγητής ΠΑ.Δ.Α.

Αθήνα, Ιούλιος 2021

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η ΓΙΑΚΟΥΜΠ ΓΙΑΝ ΛΑΠΙΝΣΚΙ του ΒΑΛΝΤΕΜΑΡ, με αριθμό μητρώου 71345524 φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο/Η Δηλών/ούσα



Ευχαριστίες

Θέλω να ευχαριστήσω τον καθηγητή κ. Πάρι Μαστοροκώστα για την βοήθεια του στην εκπόνηση της διπλωματικής μου εργασίας καθώς και για τον πολύτιμο χρόνο του. Εκτός από την πληθώρα γνώσεων που μας πρόσφερε όλα αυτά τα χρόνια, πάντα καλλιεργούσε μέσα μας την μηχανική σκέψη.

Επίσης θέλω να ευχαριστήσω θερμά την οικογένεια μου και ιδιαίτερα τους γονείς μου οι οποίοι πάντα έβαζαν την μόρφωση και το καλό των παιδιών τους πάνω από τους ίδιους.

Περίληψη

Στόχος της εργασίας είναι η μελέτη μεθόδων βαθιάς μάθησης που αναπτύσσονται στο βιβλίο “Deep learning with Python” του Francois Chollet και στην συνέχεια η πρακτική εφαρμογή των μεθόδων αυτών πάνω σε πραγματικά δεδομένα με τελικό σκοπό την εκμάθηση νευρωνικών δικτύων.

Abstract

The purpose of this thesis is to study deep learning methods from the book “Deep learning with Python” by Francis Chollet and applying these methods in order to train neural networks on real life datasets.

Περιεχόμενα

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΩΝ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ευχαριστίες

Περίληψη

Abstract

Περιεχόμενα

Κεφάλαιο 1: Θεωρία

1.1 Τεχνητή Νοημοσύνη

1.1.1 Τι είναι

1.1.2 Ιστορία

1.1.3 Turing's test

1.2 Machine Learning

1.2.1 Τι είναι

1.2.2 Κατηγορίες

1.3 Deep Learning

1.3.1 Τι είναι

1.3.2 Gradient descent

1.3.3 Νευρωνικά δίκτυα RNN

1.3.3.1 Τι είναι

1.3.3.2 Long short-term memory (LSTM)

1.3.3.3 Gated recurrent unit

1.3.4 Νευρωνικά δίκτυα CNN

Κεφάλαιο 2: Γλώσσα προγραμματισμού Python

2.1 Γενικά

2.2 Βιβλιοθήκες

2.2.1 Γενικά

2.2.2 TensorFlow

2.2.2.1 Τι είναι

2.2.2.2 Ιστορία

2.2.2.3 Εφαρμογές

2.2.3 Keras

2.2.4 Numpy

2.2.5 Matplotlib

Κεφάλαιο 3: Πρακτικές εφαρμογές

3.1 Εισαγωγή

3.1.1 Γενικά

3.1.2 Principal Component Analysis

3.1.3 F-score

3.1.3.1 Τι είναι

3.1.3.2 Δημιουργία συνάρτησης fscore

3.2 Mammography dataset

3.2.1 Γενικά

3.2.2 Εφαρμογή RNN δικτύου

3.2.2.1 Δημιουργία RNN δικτύου

3.2.2.2 Αποτελέσματα μη συμπιεσμένων δεδομένων

3.2.2.3 Αποτελέσματα συμπιεσμένων εισόδων

3.2.2.4 Σύγκριση αποτελεσμάτων

3.2.3 Εφαρμογή CNN δικτύου

3.2.3.1 Γενικά

3.2.3.2 Αποτελέσματα

3.3 Pima dataset

3.3.1 Γενικά

3.3.2 Εφαρμογή RNN δίκτυο

3.3.2.1 Αποτελέσματα μη συμπιεσμένων δεδομένων

3.3.2.2 Αποτελέσματα συμπιεσμένων δεδομένων

3.3.2.3 Σύγκριση αποτελεσμάτων

3.3.3 Εφαρμογή CNN δίκτυου

3.3.3.1 Γενικά

3.3.3.2 Αποτελέσματα

Κεφάλαιο 4

4.1 Συμπεράσματα

4.2 Βιβλιογραφία

4.3 Δικτυογραφία

Κεφάλαιο 1: Θεωρία

1.1 Τεχνητή Νοημοσύνη

1.1.1 Τι είναι

Τεχνητή Νοημοσύνη είναι ο κλάδος της επιστήμης της πληροφορικής ο οποίος ασχολείται με την ανάπτυξη υπολογιστικών συστημάτων οι οποίοι θα παρουσιάζουν στοιχειώδη ευφυΐα. Στόχος είναι τα συστήματα να μπορούν να παίρνουν αποφάσεις βάση των εισόδων που δέχονται χωρίς αυτές οι αποφάσεις να ελέγχονται ή να επιτηρούνται από ανθρώπινο παράγοντα.

Η τεχνητή νοημοσύνη είναι ένας πολύ γενικός όρος και περιλαμβάνει ένα μεγάλο εύρος τεχνικών, τεχνολογιών και αλγορίθμων που χρησιμοποιούνται για την επίτευξη του στόχου. Τα προβλήματα που προσπαθεί να λύσει η τεχνητή νοημοσύνη χωρίζονται σε πολλά παρακλάδια. Μερικά από αυτά είναι:

- **Logical AI:** Προγράμματα στα οποία το περιβάλλον, η τωρινή κατάσταση, ο στόχος προς επίτευξη καθώς και οι τρόποι αντίδρασης του συστήματος είναι προκαθορισμένοι με κάποια μαθηματική λογική. Το πρόγραμμα αποφασίζει βάση της κατάστασης που βρίσκεται και ανατρέχοντας στη λίστα των λειτουργιών που μπορεί να εκτελέσει. Παράδειγμα τέτοιου προγράμματος είναι αλγόριθμος ο οποίος “παίζει” σκάκι.
- **Pattern recognition:** Προγράμματα τεχνητής νοημοσύνης των οποίων ο σκοπός είναι να κάνει παρατηρήσεις κάποιου είδους. Συνήθως προσπαθεί να αναγνωρίσει κάποιο μοτίβο στα δεδομένα που επεξεργάζεται. Παράδειγμα τέτοιου προγράμματος είναι ένας αλγόριθμος ο οποίος προσπαθεί να αναγνωρίσει πρόσωπα σε εικόνες χρησιμοποιώντας σαν μοτίβα τα γενικά χαρακτηριστικά ενός προσώπου.
- **Planning:** Προγράμματα των οποίων ο στόχος είναι η δημιουργία μιας στρατηγικής για την επίτευξη του τελικού σκοπού. Τα προγράμματα αυτά έχουν σαν δεδομένα την αρχική κατάσταση του προβλήματος, τον τελικό σκοπό και τις επιλογές τις οποίες μπορούν να πάρουν μαζί με το πως αυτές επηρεάζουν το σύστημα. Έχοντας αυτά υπόψιν προσπαθούν να

οργανώσουν μια αλληλουχία από αυτές τις εντολές ώστε να επιτευχθεί ο στόχος με τον βέλτιστο τρόπο. Παράδειγμα τέτοιου προγράμματος είναι ένας αλγόριθμος ο οποίος οργανώνει την διαδρομή που θα ακολουθήσει ο courier για να εξυπηρετήσει όλους τους πελάτες όσο πιο γρήγορα γίνεται.

1.1.2 Ιστορία

Οι πρώτες έρευνες όσο αφορά τις σκεπτόμενες μηχανές ξεκίνησαν τις δεκαετίες 1930 έως 1950 εμπνευσμένες από τις νέες έρευνες στον τομέα της νευρολογίας οι οποίες έδειχναν ότι ο εγκέφαλος αποτελείται από δίκτυα κόμβων οι οποίοι επικοινωνούν μεταξύ τους με ηλεκτρικούς παλμούς. Μαζί με την θεωρία υπολογισμού του Alan Turing η οποία έδειξε ότι οποιαδήποτε μορφή υπολογιστικού συστήματος μπορεί να περιγραφεί ψηφιακά δημιουργήθηκαν οι πρώτες σκέψεις για το εάν μπορεί να δημιουργηθεί ψηφιακός εγκέφαλος.

Έτσι η τεχνητή νοημοσύνη γεννήθηκε την δεκαετία του 1950 όταν κάποιοι πρωτοπόροι της επιστήμης των υπολογιστών αναρωτήθηκαν εάν είναι δυνατόν να δημιουργηθούν υπολογιστικά συστήματα τα οποία θα σκέφτονται. Μια πρώτη προσπάθεια δημιουργίας προγραμμάτων τα οποία θα παίρνουν αποφάσεις χρησιμοποιούσε προγραμματιστικά προκαθορισμένους κανόνες βάση των οποίων το σύστημα έπαιρνε αποφάσεις. Αυτή η τεχνική δημιουργίας αλγορίθμων αργότερα ονομάστηκε **Symbolic AI** και αν έλυε απλά και ξεκάθαρα προκαθορισμένα προβλήματα όπως η δημιουργία αλγορίθμου ο οποίος θα παίζει σκάκι, δεν μπορούσε να εφαρμοστεί σε πιο περίπλοκα προβλήματα όπως η ταξινόμηση εικόνων και η αναγνώριση ομιλίας.

1.1.3 Turing's test

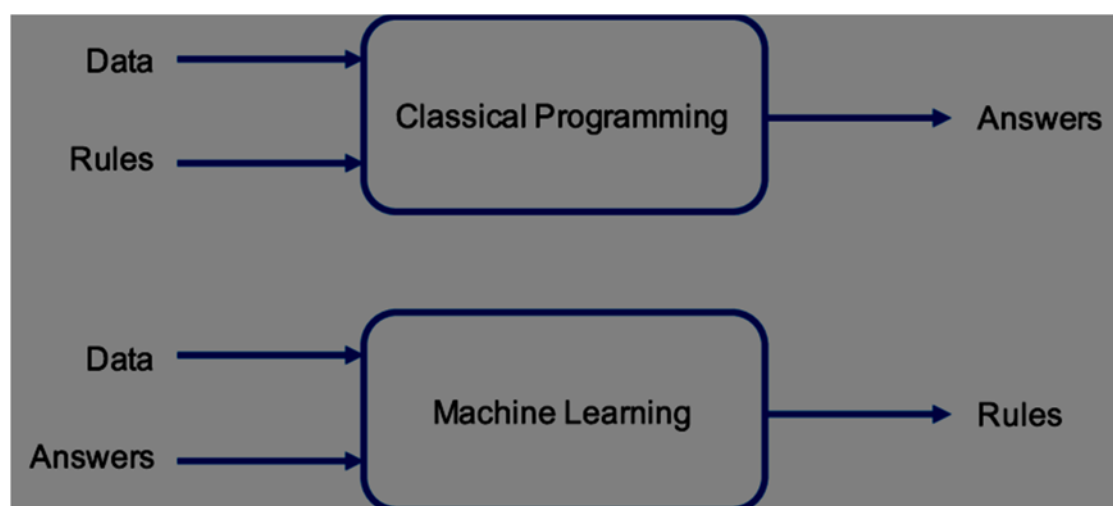
Το 1950 ο Alan Turing προσπαθώντας να απαντήσει στο ερώτημα εάν είναι δυνατόν οι υπολογιστές να σκέφτονται δημοσίευσε το διάσημο άρθρο "**Computing Machinery and Intelligence**". Στο άρθρο αυτό περιγράφει πως είναι δύσκολο να οριστεί πότε κάτι μπορεί να θεωρηθεί ότι σκεφτεται. Με στόχο την απάντηση του ερωτήματος αυτού δημιούργησε το Imitation game γνωστό και ως Turing's test. Στο test αυτό ένα άτομο θα αντάλλαζε μηνύματα κειμένου

με έναν άλλο άνθρωπο καθώς και με έναν αλγόριθμο φτιαγμένο για να παράγει μηνύματα που να μοιάζουν με ανθρώπινες απαντήσεις. Εφόσον ο αλγόριθμος κατάφερε να ξεγελάσει το άτομο που θα δεχόταν τις απαντήσεις τότε θα θεωρούταν ότι πέρασε το test. Στο άρθρο αυτό ο Alan Turing διαφωνούσε με τις περισσότερες αντιρρήσεις της εποχής για την δυνατότητα δημιουργίας “μηχανών που σκέφτονται”. Το άρθρο αυτό παραμένει έως σήμερα ένα από τα σημαντικότερα έγγραφα στην φιλοσοφία της τεχνητής νοημοσύνης.

1.2 Machine Learning

1.2.1 Τι είναι

Machine learning (Μηχανική Μάθηση) είναι ένα υποπεδίο της τεχνητής νοημοσύνης. Η μηχανική μάθηση μελετά την κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν προβλέψεις. Σκοπός τους είναι να αναγνωρίσουν πως τα δεδομένα τα οποία τους δίνονται επηρεάζουν το τελικό αποτέλεσμα. Σε αντίθεση με το Symbolic AI το οποίο χρησιμοποιεί προκαθορισμένους κανόνες για την λήψη αποφάσεων, η Μηχανική Μάθηση στοχεύει στην δημιουργία αλγορίθμων οι οποίοι θα μαθαίνουν μόνοι τους κανόνες βάση των οποίων θα “σκεφτονται”.

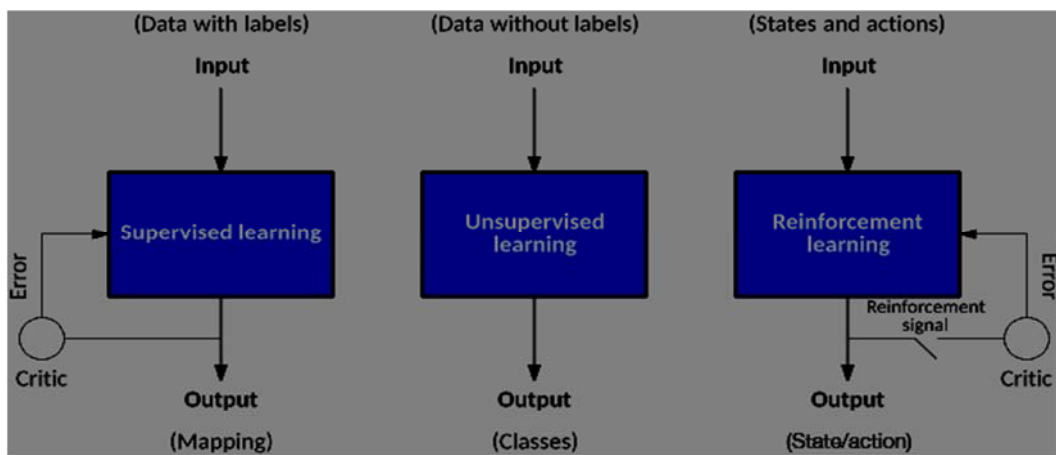


1.2.2 Κατηγορίες

Οι εργασίες μηχανικής μάθησης συνήθως ταξινομούνται σε τρεις

κατηγορίες. Αυτές είναι:

- **Επιτηρούμενη μάθηση:** Το υπολογιστικό πρόγραμμα δέχεται τις παραδειγματικές εισόδους και προσπαθεί να μάθει πως να αντιστοιχίζει τα δεδομένα εισόδου με τα δεδομένα εξόδου.
- **Μη επιτηρούμενη μάθηση:** Χωρίς να δέχεται παραδειγματικές εισόδους ο αλγόριθμος μάθησης, πρέπει να βρεί την δομή των δεδομένων εισόδου.
- **Ενισχυτική μάθηση:** Ένα πρόγραμμα υπολογιστή προσπαθεί να επιτύχει έναν συγκεκριμένο στόχο χωρίς κάποιος να του λέει ρητά πως να το κάνει. Ανα τακτά χρονικά διαστήματα οι αλγόριθμοι αυτοί δέχονται απο το κάποιον επιτηρητή είσοδο για το πόσο κοντά είναι στον στόχο.



Απεικόνιση των κατηγοριών Μηχανικής Μάθησης.

Συνοψίζοντας για να μπορέσουμε να αξιοποιήσουμε την Μηχανική Μάθηση χρειαζόμαστε 3 πράγματα:

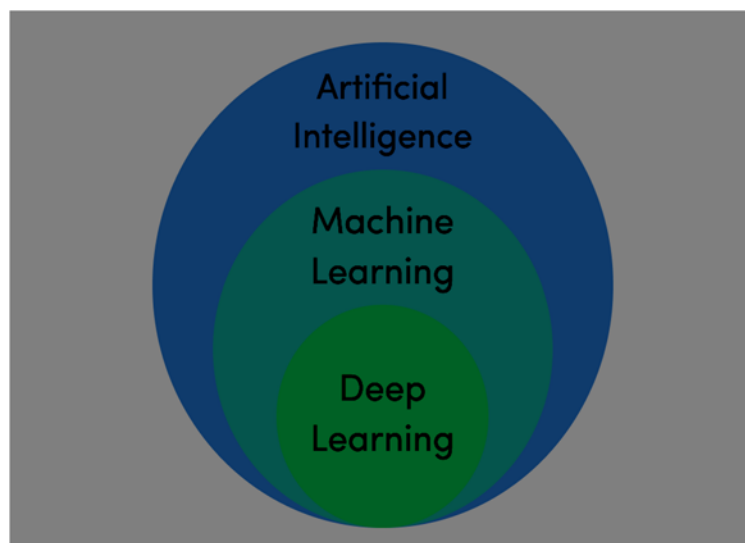
- **Δεδομένα εισόδου:** Παράδειγμα εισόδων μπορεί να είναι βαρομετρικά στοιχεία για την πρόγνωση καιρού ή φωτογραφίες προς κατάταξη.
- **Δεδομένα εξόδου:** Αυτά είναι τα αποτελέσματα τα οποία συνδέονται με τα αντίστοιχα δεδομένα εισόδου. Σε ένα πρόβλημα ταξινόμησης εικόνων μια εξόδος θα μπορούσε να αφορά το εάν η εικόνα περιλαμβάνει κάποιο είδος ζώου.
- **Τρόπος μέτρησης της αποτελεσματικότητας του αλγορίθμου:** Είναι

αναγκαίο να μπορούμε να καθορίσουμε πόσο κοντά ήταν η πρόβλεψη του μοντέλου στο πραγματικό θεμιτό αποτέλεσμα. Η μέτρηση αυτή χρησιμοποιείται σαν ανατροφοδότηση του συστήματος ώστε θα ρυθμιστεί ο τρόπος που λειτουργεί ο αλγόριθμος.

1.3 Deep Learning

1.3.1 Τι είναι

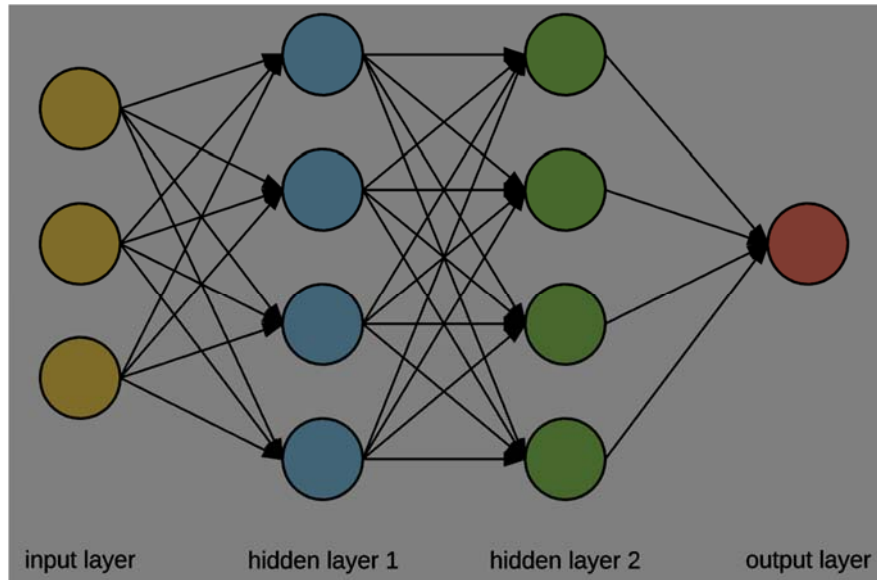
Deep Learning ή Βαθιά Μάθηση είναι ένα συγκεκριμένο υποπεδίο της Μηχανικής Μάθησης το οποίο βασίζεται στην δημιουργία διαδοχικών στρώσεων τα οποία ενισχύουν την διαδικασία της εκμάθησης του μοντέλου. Ο όρος “Βαθιά Μάθηση” δεν προέκυψε από το γεγονός ότι η τεχνολογία αυτή επιτυγχάνει πιο “βαθιά” εκμάθηση ενός μοντέλου σε σχέση με άλλες τεχνολογίες της τεχνητής νοημοσύνης. Αντιθέτως ο όρος αυτός δημιουργήθηκε λόγω των πολλαπλών συνδεδεμένων σε σειρά επεξεργαστικών κόμβων.



Απεικόνιση της συσχέτισης μεταξύ Τεχνητής Νοημοσύνης, Μηχανικής και Βαθιάς Μάθησης.

Η βαθιά μάθηση αξιοποιεί μοντέλα τα οποία ονομάζονται νευρωνικά δίκτυα συνδεδεμένα διαδοχικά. Παρά το γεγονός ότι τα Νευρωνικά δίκτυα πήραν το όνομα τους από την νευροβιολογία δεν έχει αποδειχθεί πως

λειτουργούν με τον ίδιο τρόπο που λειτουργούν τα νευρωνικά δίκτυα του ανθρώπινου εγκεφάλου. Τα νευρωνικά δίκτυα στον τομέα της τεχνητής νοημοσύνης είναι δίκτυα τα οποία αποτελούνται από διασυνδεδεμένους κόμβους οι οποίοι χρησιμοποιούν μαθηματικά μοντέλα για την επεξεργασία της πληροφορίας.



Απεικόνιση ενός νευρωνικού δικτύου

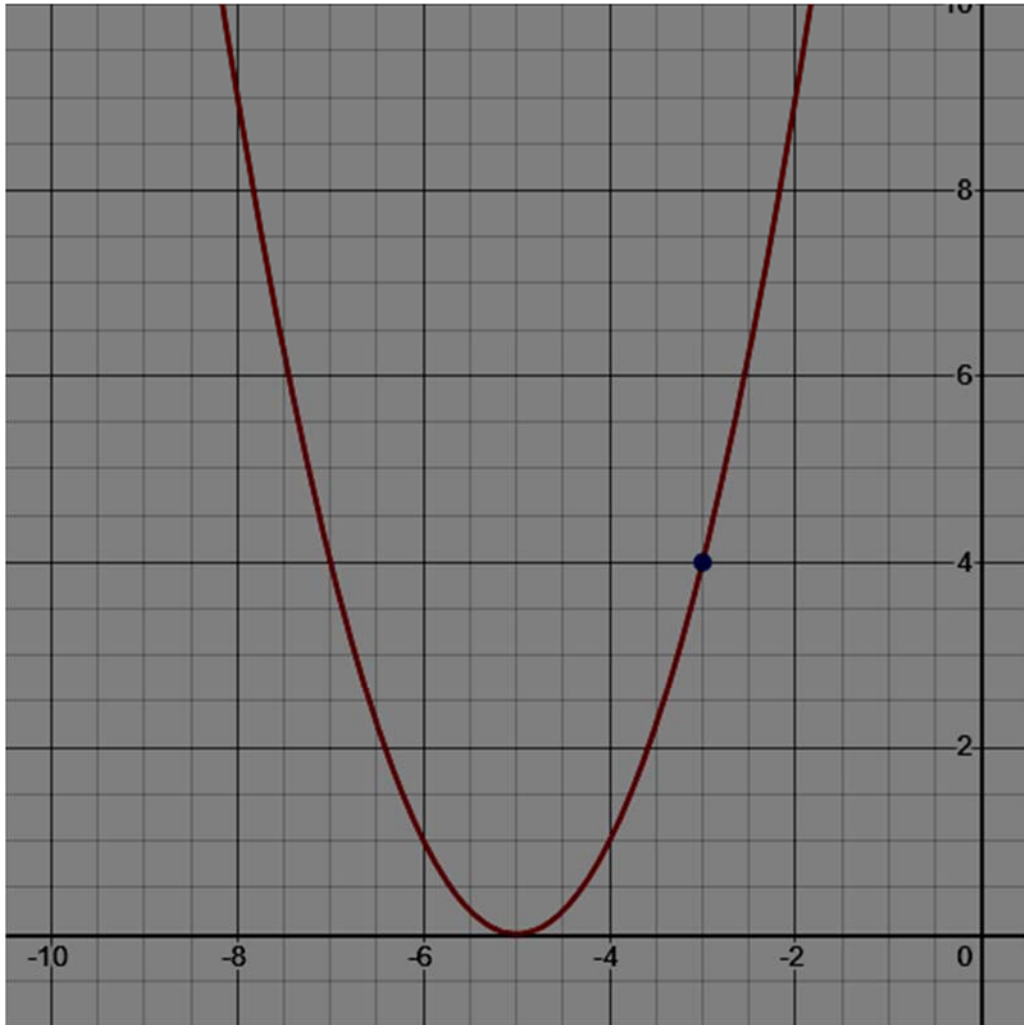
1.3.2 Gradient descent

Όπως αναφέραμε ένα νευρωνικό δίκτυο προσπαθεί συνεχώς να βρει πως τα δεδομένα της εισόδου σχετίζονται με τα δεδομένα εξόδου μέσω μαθηματικών πράξεων. Κατα την διαδικασία εκμάθησης το μοντέλο κάνει προβλέψεις βάση των οποίων αργότερα υπολογίζει την ακρίβεια του. Βάση της αστοχίας του, το σύστημα παραμετροποιεί τα βάρη του αλγορίθμου που χρησιμοποιεί με στόχο να πετύχει καλύτερα αποτελέσματα την επόμενη φορά.

Πώς όμως ένα μοντέλο γνωρίζει πως πρέπει να παραμετροποιήσει τα βάρη έτσι ώστε η αστοχία να μειωθεί; Ας θεωρήσουμε πως η αστοχία του συστήματος μας υπολογίζεται από μία συνάρτηση f η οποία δέχεται σαν παραμέτρους τα βάρη του συστήματος μας. Στόχος του μοντέλου είναι ο υπολογισμός του συνδυασμού των βαρών τα οποία θα επιφέρουν την

μικρότερη δυνατή τιμή της συνάρτησης f . Ιδανικά θέλουμε να βρεθεί ο συνδυασμός των βαρών που μηδενίζει την συνάρτηση. Αυτό επιτυγχάνεται με την παραγοντοποίηση της συνάρτησης και την κίνηση αντίθετα με την τιμή της παραγώγου του συγκεκριμένου σημείου.

Ας δούμε ένα απλό παράδειγμα. Έστω η συνάρτηση $y = (x + 5)^2$ με παράγωγο $y' = 2(x + 5)$ και αρχική τιμή παραμέτρου $x_0 = -3$.



Ο μαθηματικός τύπος που θα εφαρμόσουμε για να βρούμε την τιμή του x για να ελαχιστοποιήσουμε την αστοχία του συστήματος είναι ο εξής:

$$x_{n+1} = x_n - (\text{learningrate}) \times \left(\frac{dy}{dx}\right)$$

Όπου:

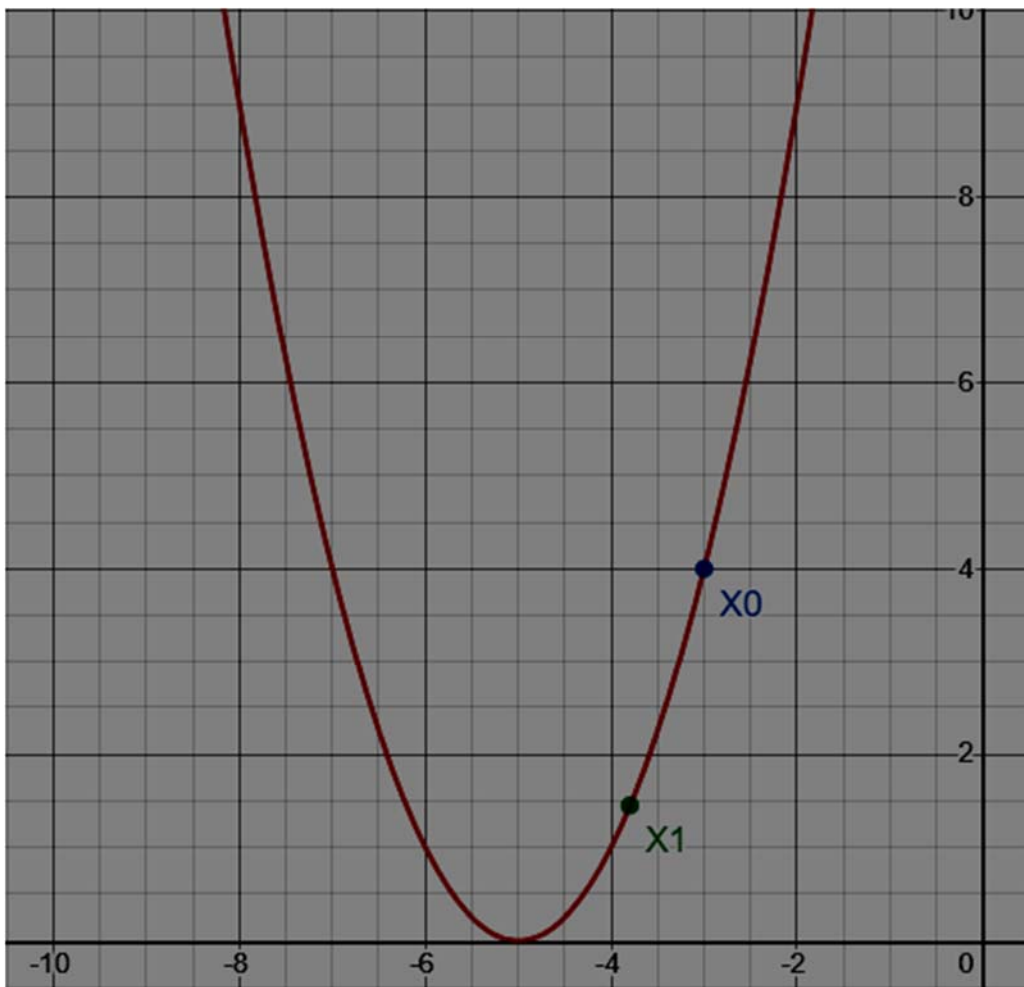
- x_{n+1} : Το βάρος που θα εφαρμόσουμε στην επόμενη επανάληψη της

εκμάθησης του νευρωνικού δικτύου με στόχο την μείωση της αστοχίας.

- x_n : Το τωρινό βάρος του συστήματος.
- (*learningrate*): Μεταβλητή που ορίζει την ταχύτητα εκμάθησης του μοντέλου. Όσο πιο μεγάλη η τιμή τόσο πιο απότομη η αλλαγή του βάρους πράγμα που μπορεί να δυσκολέψει την εύρεση της ελάχιστης τιμής. Οι πολύ μικρές τιμές θα αυξήσουν αρκετά τον χρόνο εύρεσης της ελάχιστης τιμής.
- $\frac{dy}{dx}$: Η παράγωγος της συνάρτησης.

Εφαρμόζοντας την συνάρτηση για *learningrate* = 0.2θα πάρουμε $x_1 = -3 - (0.2 \times (2 \times (-3 + 5)))$

Επομένως για $x_1 = -3.8$



Συνεχίζοντας με αυτό τον τρόπο σε επόμενες επαναλήψεις του

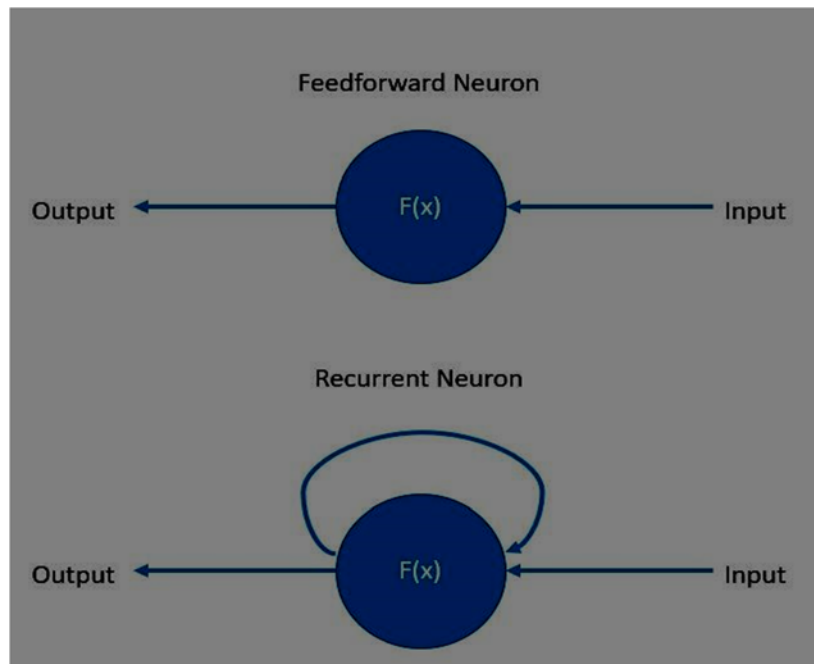
συστήματος μπορούμε να μειώσουμε ακόμα περισσότερο την αστοχία του συστήματος.

Η διαδικασία που ακολουθήσαμε ονομάζεται **Gradient descent** και χρησιμοποιείται συνεχώς στον τομέα της βαθιάς μάθησης για τον υπολογισμό των βαρών με στόχο την εκμηδένιση της αστοχίας. Το παράδειγμα που δείξαμε παραπάνω αποτελεί ένα πολύ απλό παράδειγμα. Στην πραγματικότητα ένα νευρωνικό δίκτυο χρησιμοποιεί πολλαπλά βάρη πράγμα που κάνει το πρόβλημα πιο σύνθετο.

1.3.3 Νευρωνικά δίκτυα RNN

1.3.3.1 Τι είναι

RNN ή αλλιώς Recurrent Neural Network είναι μια κατηγορία νευρωνικών δικτύων που παρουσιάζει μια σημαντική διαφορά σε σχέση με άλλα νευρωνικά δίκτυα. Τα δίκτυα αυτά κρατάνε συνεχώς παλιότερες καταστάσεις του ίδιου του συστήματος. Σε αντίθεση δηλαδή με τα υπόλοιπα feedforward δίκτυα τα οποία επεξεργάζονται όλο το σύνολο των δεδομένων μαζί, τα δίκτυα RNN επεξεργάζονται τα δεδομένα με την σειρά κρατώντας στην μνήμη παλιότερες καταστάσεις του συστήματος. Αυτή η ιδιότητα κάνει τα RNN δίκτυα ιδανικά για προβλήματα όπου η σειρά των δεδομένων είναι εξίσου σημαντική όσο τα ίδια τα δεδομένα. Παράδειγμα τέτοιου προβλήματος είναι η πρόβλεψη καιρού όπου δεν μας νοιάζουν μονάχα τα βαρομετρικά στοιχεία της σημερινής μέρας αλλά και των προηγούμενων.



1.3.3.2 Long short-term memory (LSTM)

Τα Long short-term memory είναι ένα από τα πολλά είδη νευρωνικών δικτύων που ανήκουν στην κατηγορία των RNN. Τα LSTM μπορούν να επεξεργαστούν μεμονωμένα στιγμιότυπα δεδομένων καθώς και ολόκληρες αλληλουχίες δεδομένων.

Μια συνήθης LSTM μονάδα αποτελείται από τα ένα πυρήνα, μια πύλη εισόδου, μια πύλη εξόδου καθώς και μια πύλη που αφορά την επαναφορά της μνήμης του πυρήνα. Ο πυρήνας της μονάδας κρατάει στην μνήμη του στιγμιότυπα για αυθαίρετα χρονικά διαστήματα και οι 3 πύλες ρυθμίζουν τις ροές εισόδου και εξόδου του πυρήνα.

Η ιδιότητα να κρατάνε στην μνήμη προηγούμενες καταστάσεις καθιστά τα δίκτυα αυτά χρήσιμα στην επεξεργασία, κατηγοριοποίηση και πρόβλεψη αποτελεσμάτων σε δεδομένα που ακολουθούν χρονική σειρά. Τα LSTM μοντέλα παρέχουν μεγαλύτερη μνήμη από τα συνηθισμένα δίκτυα RNN καθώς και πύλη εκκαθάρισης της μνήμης η οποία χρησιμοποιείται όταν τα δεδομένα αυτά δεν είναι πια χρήσιμα.

1.3.3.3 Gated recurrent unit

Τα gated recurrent units δημιουργήθηκαν το 2014 από τον Kyunghyun Cho. Τα GRU βασίζονται στην ίδια λογική με τα LSTM με την διαφορά ότι χρησιμοποιούν λιγότερες παραμέτρους και δεν παρέχουν πύλη εξόδου. Η ακρίβεια των GRU RNN δικτύων φαίνεται να είναι πολύ κοντά με αυτή των LSTM δικτύων σε πολλά προβλήματα. Επίσης σε συγκεκριμένες περιπτώσεις έχει παρατηρηθεί τα GRU δίκτυα να επιτυγχάνουν καλύτερους χρόνους σε σχέση με τα αντίστοιχα LSTM.

1.3.4 Νευρωνικά δίκτυα CNN

Convolutional Network ή αλλιώς CNN είναι ένα είδος νευρωνικού δικτύου. Σε αντίθεση με τα RNN δίκτυα που αναφέρθηκαν νωρίτερα πρόκειται για ένα feed forward δίκτυο. Αυτό σημαίνει ότι όλα τα δεδομένα εισόδου επεξεργάζονται μαζί και το σύστημα δεν κρατάει στην μνήμη του παλαιότερες καταστάσεις. Η ονομασία προέρχεται από την λέξη Convolution ή Συνέλιξη στα Ελληνικά λόγω του ότι τα δίκτυα αυτά χρησιμοποιούν την μαθηματική πράξη της συνέλιξης για την επεξεργασία των δεδομένων.

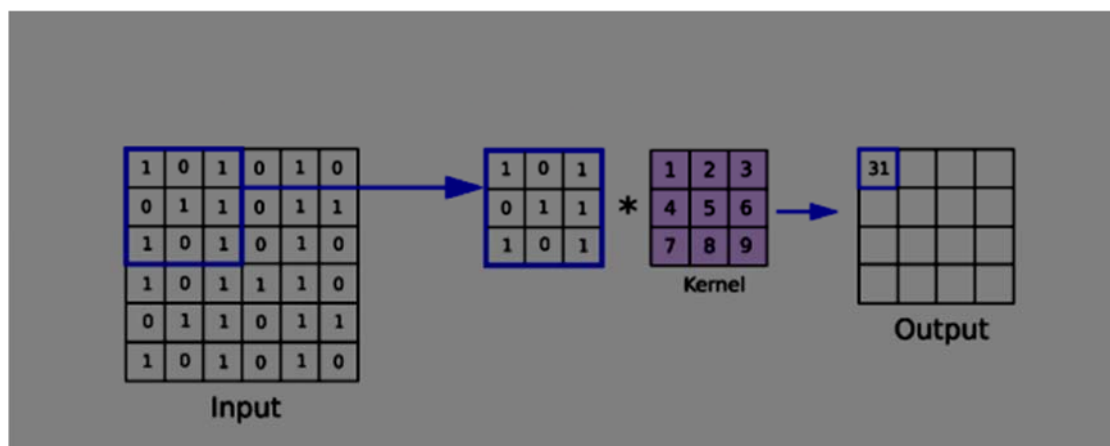
Το πρώτο κομμάτι ενός CNN νευρωνικού δικτύου είναι ένα συνελικτικό δίκτυο και αποτελείται από πολλαπλά επίπεδα κόμβων. Συγκεκριμένα αποτελείται από τουλάχιστον 3 επίπεδα. Το πρώτο επίπεδο αφορά την είσοδο και συνδέεται με τα κρυφά επίπεδα τα οποία αφορούν την επεξεργασία των δεδομένων (συνήθως είναι περισσότερα από ένα) και στην συνέχεια συνδέονται με το επίπεδο εξόδου του δικτύου. Το κάθε κρυφό επίπεδο επεξεργάζεται τα δεδομένα μέσω μαθηματικών πράξεων και στέλνει τα αποτελέσματα στο ακριβώς επόμενο επίπεδο.

Στην συνέχεια με σκοπό να κρατηθούν μόνο τα σημαντικότερα αποτελέσματα της εξόδου του συνελικτικού δικτύου, τα αποτελέσματα δίνονται σαν είσοδοι σε ένα **pooling layer**. Οι στρώσεις αυτές δρουν πάνω σε συμπλέγματα κόμβων με σκοπό την εύρεση ενός αριθμού καλύτερων

χαρακτηριστικών και αφαίρεση των υπολοίπων. Υπάρχουν δύο είδη ομαδοποίησης που μπορεί να εφαρμόσει ένα pooling layer.

- **Max pooling:** χρησιμοποιεί τις μέγιστες τιμές για κάθε σύμπλεγμα κόμβων.
- **Average pooling:** χρησιμοποιεί την μέση τιμή των τιμών κάθε συμπλέγματος.

Πολλές φορές ώστε να επιτευχθεί ο στόχος εφαρμόζονται πολλά δίκτυα συνδεδεμένα σε σειρά. Τα αποτελέσματα δηλαδή ενός pooling layer μπορούν να συνδεθούν και με την είσοδο ενός άλλου Convolution layer και η διαδικασία να συνεχιστεί. Τέτοιο παράδειγμα φαίνεται στην προηγούμενη εικόνα.



Στην παραπάνω εικόνα φαίνεται η διαδικασία που ακολουθούν τα συνελκτικά δίκτυα για την επεξεργασία των δεδομένων. Πιο συγκεκριμένα δημιουργείται μια μήτρα (kernel) η οποία με την πράξη της συνέλιξη πάνω στα δεδομένα εισόδου έχει ως αποτέλεσμα μια μήτρα εξόδου. Η μήτρα εξόδου αφορά ένα βρόγχο εξόδου του συστήματος. Αυτό σημαίνει πως εάν έχουμε ένα συνελκτικό δίκτυο με αριθμό εξόδων 32 τότε θα δημιουργηθούν 32 μήτρες εξόδου. Η κάθε τέτοια μήτρα θα έχει διαφορετικά αποτελέσματα μιας και τα βάρη επιλέγονται αυθαίρετα από το σύστημα και μετά παραμετροποιούνται κατά την εκπαίδευση του.

Κεφάλαιο 2: Γλώσσα προγραμματισμού Python

2.1 Γενικά

Python είναι μια γλώσσα υψηλού επιπέδου γενικής χρήσης. Δίνει μεγάλη έμφαση στο να είναι ευανάγνωστος ο κώδικας της και για αυτόν τον λόγο πολλές φορές προτείνεται σαν πρώτη γλώσσα σε άτομα που θέλουν να μάθουν να προγραμματίζουν. Πρόκειται για μια δυναμική και αντικειμενοστραφής γλώσσα προγραμματισμού η οποία χρησιμοποιείται σε μικρές καθώς και μεγάλες εφαρμογές. Μεταξύ πολλών άλλων λειτουργιών η Python διαχειρίζεται την μνήμη απελευθερώνοντας τις θέσεις αυτές οι οποίες δεν χρησιμοποιούνται πλέον απο το πρόγραμμα (garbage collection). Ένα μεγάλο της χαρακτηριστικό είναι ότι σε αντίθεση με άλλες γλώσσες προγραμματισμού δεν χρησιμοποιεί αγκύλες για την δήλωση πεδίου δράσης μιας συνάρτησης η ενός βρόγχου ελέγχου. Αντι για αγκύλες χρησιμοποιεί εσοχές για την δήλωση του πεδίου δράσης.



2.2 Βιβλιοθήκες

2.2.1 Γενικά

Ένα από τα μεγάλα πλεονεκτήματα της Python είναι ο τεράστιος αριθμός βιβλιοθηκών οι οποίες προσφέρουν πληθώρα εργαλείων κατάλληλων για πολλές εφαρμογές. Τον Μάρτιου του 2021 η επίσημη αποθήκη βιβλιοθηκών της Python (PyPi) περιέχει πάνω από 290,000 βιβλιοθήκες. Μεταξύ άλλων οι λειτουργίες που υλοποιούν πολλές από τις εφαρμογές αφορούν τομείς της πληροφορικής όπως:

- Αυτοματοποίηση

- Ανάλυση δεδομένων
- Επεξεργασία εικόνας
- Βάσεις δεδομένων
- Γραφικά περιβάλλοντα διεπαφής χρήστη
- Μηχανική Μάθηση
- Επεξεργασία κειμένων

Μια σημαντική βιβλιοθήκη την οποία θα χρησιμοποιήσουμε είναι η TensorFlow. Ανεπτυγμένη από την Google προσφέρει σημαντικά εργαλεία για την υλοποίηση νευρωνικών δικτύων.

2.2.2 TensorFlow

2.2.2.1 Τι είναι

Ιστότοπος βιβλιοθήκης: <https://www.tensorflow.org/>

TensorFlow είναι βιβλιοθήκη ανοιχτού λογισμικού η οποία δημιουργήθηκε με σκοπό την διευκόλυνση διαδικασιών όπως ο προγραμματισμός ροής δεδομένων, ενώ βρήκε χρήση και στην μηχανική μάθηση έχοντας ως παράδειγμα τα νευρωνικά δίκτυα. Χρησιμοποιείται από την Google στις παραγωγικές τους μονάδες καθώς και σαν ερευνητικό εργαλείο.

2.2.2.2 Ιστορία

Το 2011 δημιουργήθηκε από την Google το DistBelief, ένα σύστημα μηχανικής μάθησης βασισμένο στην βαθιά μάθηση και στα νευρωνικά δίκτυα. Έγινε γρήγορα διάσημο καθώς χρησιμοποιούνταν από πολλές εταιρείες της Alphabet για ερευνητικούς αλλά και εμπορικούς λόγους. Η βιβλιοθήκη αυτή χρησιμοποιούσε μόνο τους πυρήνες του επεξεργαστή για την δημιουργία νευρωνικών δικτύων. Αναθέτοντας σε πολλούς επιστήμονες του τομέα της πληροφορικής να απλοποιήσουν και να κάνουν την βιβλιοθήκη πιο γρήγορη, η Google δημιούργησε την βιβλιοθήκη tensorflow. Έτσι το 2017 ανακοινώνεται η πρώτη έκδοση του TensorFlow. Η βιβλιοθήκη tensorflow μπορεί να εκτελεστεί και με CPU καθώς και με GPU. Λόγω των συνεχόμενων βελτιώσεων των καρτών γραφικών η χρόνοι εκπαίδευσης νευρωνικών δικτύων έχουν μειωθεί

ραγδαία.

2.2.2.3 Εφαρμογές

Το TensorFlow χρησιμοποιείται σε ένα μεγάλο εύρος εφαρμογών μεγάλης κλίμακας. Παραδείγματα τέτοιων εφαρμογών καθώς και εταιρειών που αξιοποιούν την βιβλιοθήκη είναι:

- **Mozilla:** Ένα σύστημα φωνητικής αναγνώρισης.
- **Google:** Αλγόριθμος τεχνητής νοημοσύνης που αποτελεί σημαντικό κομμάτι της μηχανής αναζήτησης της.
- **AirBnB:** Χρησιμοποιεί νευρωνικά δίκτυα μεταξύ άλλων για την κατηγοριοποίηση φωτογραφιών ενός διαμερίσματος στον αντίστοιχο χώρο του σπιτιού.
- **Coca Cola:** Νευρωνικό δίκτυο για την αναγνώριση των κωδικών που κρύβονται κάτω από τα καπάκια των προϊόντων.
- **Paypal:** Νευρωνικά δίκτυα για την αναγνώριση μοτίβων που μπορούν να οδηγήσουν στην προφύλαξη από απάτες.
- **Spotify:** Νευρωνικά δίκτυα για την βελτιστοποίηση των εξατομικευμένων συστάσεων προς του χρήστες.

2.2.3 Keras

Ιστότοπος βιβλιοθήκης: <https://keras.io/>

Το Keras είναι άλλη μια από τις πολύ χρήσιμες βιβλιοθήκες της Python η οποία προσφέρει διεπαφές για άλλες βιβλιοθήκες που αφορούν την μηχανική μάθηση όπως το TensorFlow. Η βιβλιοθήκη αυτή διευκολύνει αρκετά την δημιουργία ολοκληρωμένων νευρωνικών δικτύων. Μεταξύ άλλων προσφέρει αρκετά έτοιμα είδη νευρωνικών δικτύων δίνοντας παράλληλα την δυνατότητα παραμετροποιήσεις. Από την έκδοση 2.4 υποστηρίζει μόνο την βιβλιοθήκη TensorFlow όμως παλιότερες εκδόσεις υποστήριζαν και άλλες βιβλιοθήκες όπως:

- Microsoft Cognitive Toolkit

- Theano
- PlaidML

2.2.4 Numpy

Ιστότοπος βιβλιοθήκης: <https://numpy.org/>

Το Numpy είναι μια βιβλιοθήκη για την γλώσσα Python η οποία έχει δημιουργηθεί με σκοπό να διευκολύνει αρκετές μαθηματικές διαδικασίες. Συγκεκριμένα προσφέρει πληθώρα μαθηματικών και μη συναρτήσεων για τον ευκολότερο χειρισμό διανυσμάτων και πινάκων. Στα παραδείγματα που θα ακολουθήσουν θα χρησιμοποιήσουμε την συγκεκριμένη βιβλιοθήκη μεταξύ άλλων για τον ευκολότερο χειρισμό διανυσμάτων και την επεξεργασία των δεδομένων πριν την εισαγωγή τους στα νευρωνικά δίκτυα.

2.2.5 Matplotlib

Ιστότοπος βιβλιοθήκης: <https://matplotlib.org/>

Πρόκειται για μια βιβλιοθήκη της Python δημιουργημένη για την δημιουργία διαγραμμάτων. Έχει αναπτυχθεί με στόχο να μοιάζει με τα διαγράμματα του Matlab. Η Matplotlib προσφέρει αντικειμενοστραφή τρόπο για την δημιουργία πολλών ειδών διαγραμμάτων καθώς και κινούμενων διαγραμμάτων.

Κεφάλαιο 3: Πρακτικές εφαρμογές

3.1 Εισαγωγή

3.1.1 Γενικά

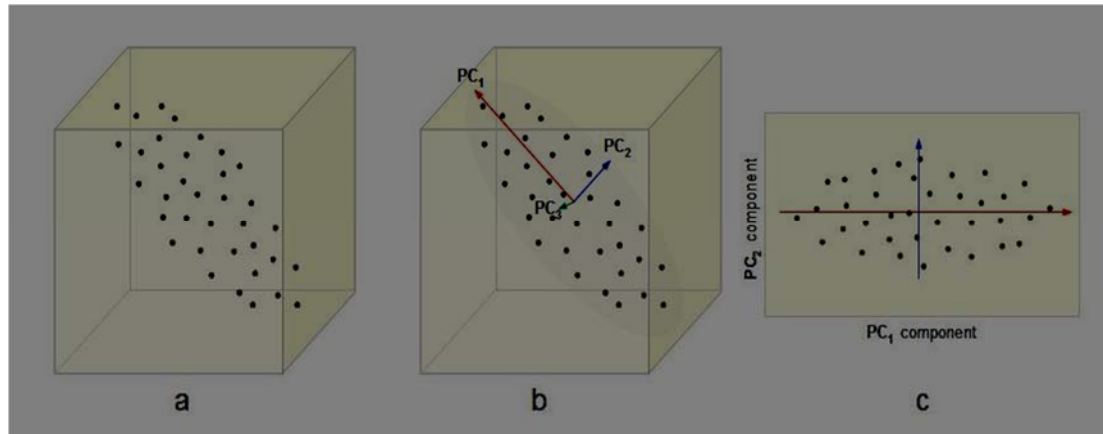
Σε αυτό το Κεφάλαιο θα εφαρμοστούν τα νευρωνικά δίκτυα που έχουν προαναφερθεί σε προηγούμενα κεφάλαια αξιοποιώντας μεθόδους και τεχνικές από το βιβλίο *Deep Learning with Python* του Francois Chollet. Μεταξύ άλλων θα εφαρμοστεί ένα νευρωνικό δίκτυο RNN καθώς και ένα CNN σε δύο διαφορετικά σύνολα δεδομένων (datasets). Το κάθε ένα από τα dataset αυτά αποτελείται από δύο διαφορετικές εκδόσεις. Η πρώτη έκδοση των δεδομένων είναι τα ατόφια δεδομένα ενώ η δεύτερη απαρτίζεται από τα ίδια δεδομένα άλλα συμπιεσμένα με την μαθηματική μέθοδο PCA που εξηγείται στην συνέχεια. Επομένως εκτός της σύγκρισης των αποτελεσμάτων των δυο νευρωνικών δικτύων θα μας δοθεί και η ευκαιρία να δούμε πως η συμπίεση των δεδομένων εισόδου μπορεί να επηρεάζει τα τελικά αποτελέσματα των νευρωνικών δικτύων.

3.1.2 Principal Component Analysis

Η **Ανάλυση Κυρίων Συνιστωσών** είναι μια στατιστική μέθοδος ανάλυσης παραγόντων. Εφευρέθηκε το 1901 από τον Karl Pearson ενώ ονομάστηκε και αναπτύχθηκε ανεξάρτητα από τον Harold Hotelling το 1930. Με την μέθοδο αυτή ένα σύνολο δεδομένων το οποίο αποτελείται από N μετρήσεις όπου η κάθε μέτρηση αποτελείται από K μεταβλητές, μπορεί να απεικονιστεί σε ένα “σύννεφο” το οποίο αποτελείται από N σημεία σε έναν χώρο K διαστάσεων. Συχνά χρησιμοποιείται για την συμπίεση των δεδομένων μειώνοντας τις διαστάσεις της απεικόνισης. Αυτό επιτυγχάνεται με την εύρεση των διαστάσεων με την μεγαλύτερη διασπορά (variance) και αφαίρεση των υπολοίπων.

Έστω ένα σημείο το οποίο αρχικά βρίσκεται στις συντεταγμένες (x,y,z) , θεωρώντας ότι οι διαστάσεις με τη μεγαλύτερη διασπορά είναι οι X και Y τότε μετά την συμπίεση το σημείο αυτό θα χαρακτηρίζεται μόνο από τις τιμές των

δύο αξόνων αυτών (x,y). Σαφώς κατα την συμπίεση των δεδομένων πάντα θα χάνεται ένα μέρος της πληροφορίας. Για αυτόν τον λόγο διατηρούνται πάντα οι διαστάσεις με την μεγαλύτερη διασπορά μιας και θεωρείται ότι αυτοι κατέχουν την μεγαλύτερη πληροφορία. Ένα τέτοιο παράδειγμα φαίνεται στην παρακάτω εικόνα.



Παράδειγμα αφαίρεσης διαστάσεων.

3.1.3 F-score

3.1.3.1 Τι είναι

Το **F-score** είναι μια στατιστική μέθοδος μέτρησης ακρίβειας ενός μοντέλου πάνω σε ένα σύνολο δεδομένων. Υπολογίζεται απο το **precision**(ακρίβεια) και το **recall**(ανάκληση), όπου το precision ορίζεται ως η διαίρεση των σωστά αναγνωρισμένων θετικών αποτελεσμάτων προς το σύνολο των λανθασμένα αναγνωρισμένων θετικών αποτελεσμάτων συν το σύνολο των σωστά αναγνωρισμένων θετικών.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Απεικόνιση υπολογισμού του precision.

Το recall ορίζεται ως το σύνολο των σωστά αναγνωρισμένων θετικών αποτελεσμάτων προς το σύνολο όλων θετικών αποτελεσμάτων.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Απεικόνιση υπολογισμού του recall.

Έχοντας υπολογίσει το **precision** και το **recall** μπορούμε να υπολογίσουμε το **f-score** με την παρακάτω σχέση.

$$F_1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Εξίσωση υπολογισμού του f-score.

Η μέγιστη και ιδανική τιμή του f-score είναι το 1.0 το οποίο υποδεικνύει τέλεια ακρίβεια και ανάκληση, ενώ η ελάχιστη τιμή είναι το 0.

3.1.3.2 Δημιουργία συνάρτησης fscore

Όστε να μπορέσουμε να εφαρμόσουμε την μέθοδο fscore πάνω στα αποτελέσματα των νευρωνικών μα δικτύων θα πρέπει να δημιουργήσουμε την συνάρτηση υπολογισμού του fscore με της χρήση της Python. Παρακάτω φαίνεται η υλοποίηση της συνάρτησης fscore.

```
def fscore(true,pred):
    true_positive = 0
    false_positive = 0
    true_negative = 0
    false_negative = 0
    precision = 0.0
    recall = 0.0
```

```

fscore = 0.0
for i, t in enumerate(true):
    if(pred[i] == 1):
        if(pred[i] == true[i]):
            true_positive += 1
        else:
            false_positive += 1

    elif(pred[i] == 0):
        if(pred[i] == true[i]):
            true_negative += 1
        else:
            false_negative += 1

precision = true_positive / ( true_positive + false_positive + 0.0000001)
recall = true_positive / ( true_positive + false_negative + 0.0000001)
fscore = 2*(( precision * recall ) / ( precision + recall + 0.0000001))
return precision, recall, fscore

```

Η συγκεκριμένη συνάρτηση δέχεται δύο παραμέτρους. Η πρώτη παράμετρος **true** αφορά να σωστά αποτελέσματα, ενώ η παράμετρος **pred** δέχεται τις προβλέψεις του μοντέλου. Και οι δύο αυτοί παράμετροι παίρνουν σαν τιμές αλληλουχίες απο τιμές μιας και το fscore υπολογίζεται πάνω σε όλο το σύνολο των προβλέψεων και κατ' επεκταση το σύνολο των δεδομένων. Κατα τον υπολογισμό των **precision**, **recall** και **fscore** προσθέτουμε στους παρονομαστές μια πολύ μικρή τιμή ώστε αυτοί να μην μηδενίζονται.

3.2 Mammography dataset

3.2.1 Γενικά

Το Mammography dataset αποτελείται από έξι δεδομένα εισόδου τα οποία αφορούν κάποιες τιμές μετρήσεων μαστογραφιών καθώς και από ένα δεδομένο εξόδου το οποίο ορίζει το αν ο ασθενής είχε κάποιο όγκο. Η ύπαρξη όγκου στον ασθενή ορίζεται με την τιμή 1 και 0 αντίστοιχα εφόσον δεν έχει κάποιον όγκο. Σκοπός είναι η δημιουργία μοντέλου το οποίο θα μπορεί να

προβλέπει την πιθανότητα του άν ο ασθενής μπορεί να έχει κάποιο όγκο. Πρόκειται για πρόβλημα δυαδικής ταξινόμησης μιας και τα πιθανά αποτελέσματα μιας μέτρησης μπορούν να είναι μόνο δύο.

Το σύνολο των δεδομένων του Mammography dataset έχει χωριστεί σε training dataset και testing dataset. Επίσης όπως και στην περίπτωση του Pima dataset που θα δούμε αργότερα έχει δημιουργηθεί ένα διπλότυπο του συνόλου δεδομένων με την διαφορά ότι αυτό είναι συμπιεσμένο με την μέθοδο PCA. Με αυτόν τον τρόπο τα δεδομένα εισόδου έχουν μειωθεί από έξι σε τέσσερα. Στο τέλος θα μας δοθεί η ευκαιρία να συγκρίνουμε τα αποτελέσματα των μοντέλων με και χωρίς συμπίεση δεδομένων και να δουμε κατα πόσο αυτή μπορεί να επηρεάσει το τελικό αποτέλεσμα.

3.2.2 Εφαρμογή RNN δικτύου

3.2.2.1 Δημιουργία RNN δικτύου

Το νευρωνικό μας δίκτυο θα απαρτίζεται συνολικά από ένα GRU RNN μοντέλο καθώς και από ένα Dense μοντέλου συνδεδεμένο σε σειρά.

```
Model = Sequential()  
model.add(layers.GRU(layers_num, input_shape=(1, num_inputs),  
return_sequences=True, dropout=drop, recurrent_dropout=rec_dropout ))  
model.add(layers.Dense(1))
```

- **layers_num:** Παράμετρος που αφορά των αριθμό των κόμβων εξόδου του συστήματος.
- **input_shape:** Παράμετρος που αφορά την μορφή των δεδομένων εισόδων. Σε αυτήν την περίπτωση θα ορίσουμε στο num_inputs την τιμή 6 αφού θα χρησιμοποιήσουμε τα μη συμπιεσμένα δεδομένα.
- **return_sequence:** Παράμετρος που αφορά τι θα επιστρέφει το σύστημα. Για τιμή True το σύστημα θα επιστρέψει όλη την αλληλουχία των εξόδων (είναι αναγκαία η τιμή True όταν έχουν πολλά δίκτυα RNN συνδεδεμένα σε σειρά), ενώ για τιμή False θα επιστραφεί μόνο το τελευταίο διάνυσμα σαν έξοδος του συστήματος.

- **dropout:** Ποσοστό τυχαίας αφαίρεσης δεδομένων εισόδου (τιμές από 0 έως 1). Χρησιμοποιείται σε περιπτώσεις όπου το σύστημα παρουσιάζει πρόβλημα *overfitting*, δηλαδή μαθαίνει να αναγνωρίζει μονάχα τα δεδομένα τα οποία του δίνονται κατά την εκπαίδευση λόγω υπερέκθεσης σε αυτά και έτσι παρουσιάζει προβλήματα σε δεδομένα που δεν έχει ξαναδεί.
- **reccurent_dropout:** Πρακτικά έχει την ίδια λειτουργία με το *dropout*. Η διαφορά είναι ότι σε κάθε επανάληψη δεν αφαιρεί δεδομένα τυχαία άλλα έχει μια σταθερή μάσκα βάση της οποίας τα αφαιρεί. Χρησιμοποιείται σε RNN μοντέλα μιας και η τυχαία αφαίρεση δεδομένων σε κάθε βήμα μπορεί να κάνει ζημιά στα μοντέλα αυτά αφού βασίζονται και σε παλαιότερες καταστάσεις.

Στην συνέχεια θα πρέπει να εισάγουμε τα δεδομένα στο σύστημα.

```
model.compile(optimizer='RMSprop', loss='mse', metrics=['acc'])
history = model.fit(x=samples_expanded, y=train_targets, batch_size=batch,
steps_per_epoch=steps, epochs=50, shuffle=True, validation_split=validation_split,
verbose=1)
```

Συνάρτηση *compile*:

- **optimizer:** Αφορά τον αλγόριθμο που χρησιμοποιείται για την εκμάθηση του μοντέλου.
- **loss:** Παράμετρος που αφορά τον αλγόριθμο υπολογισμού του λάθους του συστήματος.

Συνάρτηση *fit*:

- **x:** Τα δεδομένα εισόδου του συστήματος.
- **y:** Τα δεδομένα εξόδου του συστήματος.
- **batch_size:** Αριθμός δειγμάτων πριν από κάθε ενημέρωση του *gradient*.

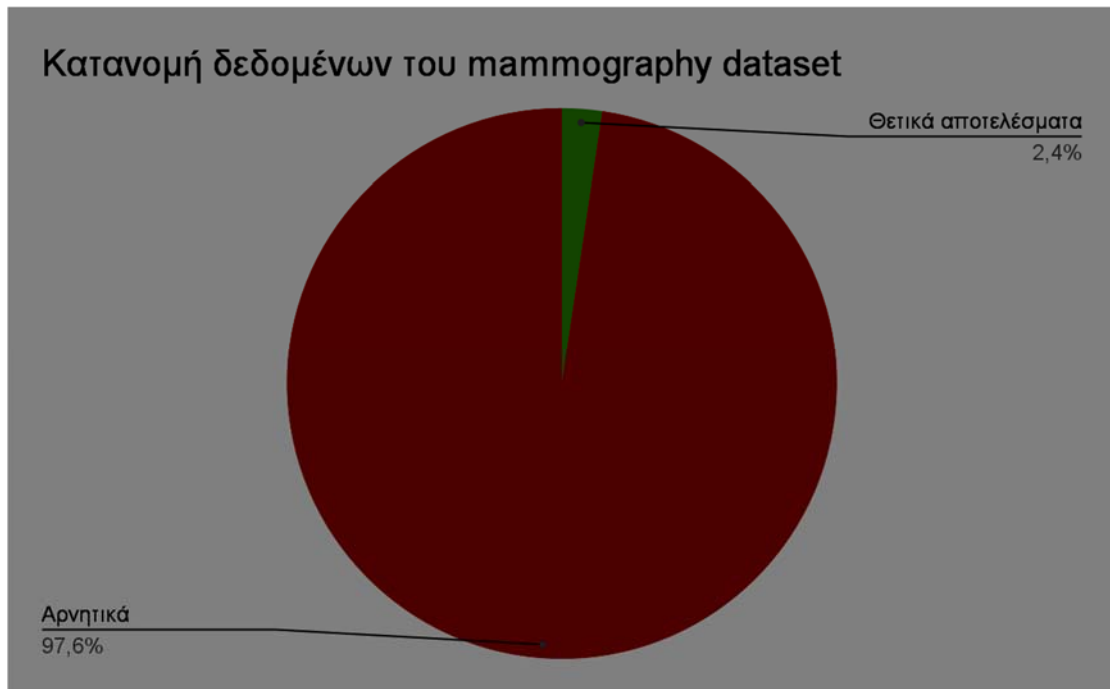
- **steps_per_epoch:** Αριθμός batch_size προς επεξεργασία ανα εποχή.
- **epochs:** Σύνολο εποχών εκμάθησης του μοντέλου.

3.2.2.2 Αποτελέσματα μη συμπιεσμένων δεδομένων

Τα πρώτα αποτελέσματα του μοντέλου φαίνονται παρακάτω

Layers	Batch	Zeros_prediction_acc	Ones_prediction_acc
32	32	0.9975579975579976	0.4810126582278481
32	64	0.9978632478632479	0.4240506329113924
32	128	0.998015873015873	0.41139240506329117
64	32	0.9975579975579976	0.4620253164556962
64	64	0.9983211233211233	0.41139240506329117
64	128	0.9981684981684982	0.3987341772151899
128	32	0.9975579975579976	0.44936708860759494
128	64	0.9981684981684982	0.4050632911392405
128	128	0.9983211233211233	0.3987341772151899

Μιάς και η αρχική ακρίβεια του συστήματος ήταν γύρω στα 98% πράγμα που φαίνεται αδύνατο για πρώτη προσπάθεια, η ακρίβεια χωρίστηκε σε ακρίβεια πρόβλεψης άσπων και ακρίβεια πρόβλεψης μηδενικών. Όπως φαίνεται το σύστημα έχει μάθει να αναγνωρίζει πολύ καλύτερα τα μηδενικά παρά τους άσπους. Μελετώντας το mammography dataset παρατηρήθηκε ότι το 98% όλου του συνόλου των αποτελεσμάτων αποτελείται από μηδενικά. Αυτό εξηγεί γιατί το μοντέλο δυσκολεύεται με την αναγνώριση των άσπων. Η υπερέκθεση του μοντέλου στα μηδενικά το καθιστά αδύνατο να αναγνωρίσει τότε πρόκειται για θετικό αποτέλεσμα.



Σε αυτό το σημείο πρέπει να ορίσουμε ποιοι είναι οι στόχοι πριν ξεκινήσουμε να εκπαιδεύουμε περαιτέρω το μοντέλο. Εφόσον το 98% του συνόλου των δεδομένων αποτελείται από μηδενικά, μια προσέγγιση η οποία θα είχε μεγάλη ακρίβεια θα ήταν να θεωρήσουμε πως όλα τα αποτελέσματα ανεξάρτητα από την είσοδο είναι μηδεν. Με αυτόν τον τρόπο θα πετυχαίναμε 98% ακρίβεια. Στην πραγματικότητα όμως δεν θα χρησιμοποιούσαμε τεχνητή νοημοσύνη και το σημαντικότερο θα είχαμε 0% ακρίβεια όσο αφορά τους άσους. Για αυτόν τον λόγο ο στόχος που τέθηκε είναι να δημιουργηθεί ένα μοντέλο το οποίο θα έχει περίπου ίσα ποσοστά ακρίβειας και για τα δυο αποτελέσματα πράγμα το οποίο μπορεί να μειώσει την συνολική ακρίβεια των προβλέψεων.

Σε αυτή την περίπτωση πρέπει να παραμετροποιήσουμε το μοντέλο έτσι ώστε να δίνει πιο πολύ σημασία στους άσους παρα στα μηδενικά. Αυτό γίνεται με την προσθήκη επιπλέον βάρους εκμάθησης. Το βάρος αυτό θα είναι διαφορετικό για τους άσους και διαφορετικό για τα μηδενικά. Μιάς και το 98% των αποτελεσμάτων είναι μηδενικά διαλέχθηκαν τα βάρη 0,02 για τα μηδενικά και 1,0 για τους άσους. Υπάρχει μια ειδική παράμετρος την οποία πρέπει να προσθέσουμε στην συνάρτηση fit.

Με την χρήση της παραμέτρου `class_weight` μπορούμε να προσθέσουμε βάρη εκμάθησης για όλες τις πιθανές τιμές εξόδου (στην περίπτωση μας είναι δυο).

```
History = model.fit(x=samples_expanded, y=train_targets, batch_size=batch,
steps_per_epoch=steps, epochs=50, class_weight={0:0.02, 1:1.0}, shuffle=True,
validation_split=validation_split, verbose=1)
```

Αυτή την φορά θα εκτελέσουμε το μοντέλο για πολλούς συνδυασμούς παραμέτρων. Τα αποτελέσματα φαίνονται παρακάτω.

Units	Batch	Rec Drop	Dropout	Zeros_prediction_acc	Ones_prediction_acc
16	16	0	0	0.9105616605616605	0.879746835443038
16	16	0.2	0	0.8841575091575091	0.8734177215189873
16	16	0.2	0.2	0.9441391941391941	0.7848101265822784
16	16	0.4	0	0.9343711843711844	0.8481012658227848
16	16	0.4	0.2	0.8702686202686203	0.879746835443038
16	16	0.4	0.4	0.8836996336996337	0.879746835443038
16	32	0	0	0.9091880341880342	0.8607594936708861
16	32	0.2	0	0.8920940170940171	0.8860759493670886
16	32	0.2	0.2	0.902014652014652	0.8607594936708861
16	32	0.4	0	0.932997557997558	0.8607594936708861
16	32	0.4	0.2	0.9033882783882784	0.8544303797468354
16	32	0.4	0.4	0.8365384615384616	0.8924050632911392
16	64	0	0	0.8884310134310134	0.8924050632911392
16	64	0.2	0	0.8814102564102564	0.9050632911392406
16	64	0.2	0.2	0.8862942612942613	0.8734177215189873
16	64	0.4	0	0.8972832722832723	0.8734177215189873
16	64	0.4	0.2	0.8797313797313797	0.8860759493670886
16	64	0.4	0.4	0.8791208791208791	0.879746835443038
16	128	0	0	0.8901098901098901	0.8860759493670886
16	128	0.2	0	0.8756105006105006	0.8924050632911392
16	128	0.2	0.2	0.8815628815628815	0.8860759493670886
16	128	0.4	0	0.8862942612942613	0.8924050632911392
16	128	0.4	0.2	0.8661477411477412	0.879746835443038
16	128	0.4	0.4	0.8449328449328449	0.8860759493670886
16	256	0	0	0.8956043956043956	0.8734177215189873
16	256	0.2	0	0.8750000000000000	0.8544303797468354

16	256	0.2	0.2	0.8667582417582418	0.8607594936708861
16	256	0.4	0	0.8728632478632479	0.8924050632911392
16	256	0.4	0.2	0.8937728937728938	0.8734177215189873
16	256	0.4	0.4	0.8183760683760684	0.8987341772151899
32	16	0	0	0.940018315018315	0.8291139240506329
32	16	0.2	0	0.9482600732600732	0.8037974683544303
32	16	0.2	0.2	0.9155982905982906	0.8417721518987342
32	16	0.4	0	0.9244505494505495	0.8481012658227848
32	16	0.4	0.2	0.8435592185592186	0.8987341772151899
32	16	0.4	0.4	0.8791208791208791	0.8734177215189873
32	32	0	0	0.9108669108669109	0.8734177215189873
32	32	0.2	0	0.914072039072039	0.879746835443038
32	32	0.2	0.2	0.9175824175824175	0.8544303797468354
32	32	0.4	0	0.9120879120879121	0.8734177215189873
32	32	0.4	0.2	0.8592796092796092	0.8924050632911392
32	32	0.4	0.4	0.8572954822954822	0.8924050632911392
32	64	0	0	0.8760683760683761	0.8987341772151899
32	64	0.2	0	0.8864468864468864	0.8924050632911392
32	64	0.2	0.2	0.8975885225885226	0.8734177215189873
32	64	0.4	0	0.8812576312576312	0.8987341772151899
32	64	0.4	0.2	0.8566849816849816	0.8987341772151899
32	64	0.4	0.4	0.8849206349206349	0.8734177215189873
32	128	0	0	0.8963675213675214	0.879746835443038
32	128	0.2	0	0.8431013431013431	0.9050632911392406
32	128	0.2	0.2	0.8705738705738706	0.8860759493670886
32	128	0.4	0	0.8835470085470085	0.8860759493670886
32	128	0.4	0.2	0.8887362637362637	0.879746835443038
32	128	0.4	0.4	0.8321123321123322	0.8987341772151899
32	256	0	0	0.9065934065934066	0.8734177215189873
32	256	0.2	0	0.8803418803418803	0.8860759493670886
32	256	0.2	0.2	0.8589743589743589	0.8860759493670886
32	256	0.4	0	0.8940781440781441	0.8924050632911392
32	256	0.4	0.2	0.8174603174603174	0.8987341772151899
32	256	0.4	0.4	0.8325702075702076	0.8924050632911392
64	16	0	0	0.9278083028083028	0.8291139240506329
64	16	0.2	0	0.9572649572649573	0.810126582278481
64	16	0.2	0.2	0.9084249084249084	0.8481012658227848
64	16	0.4	0	0.9458180708180708	0.8227848101265823
64	16	0.4	0.2	0.8470695970695971	0.8987341772151899
64	16	0.4	0.4	0.8856837606837606	0.8544303797468354

64	32	0	0	0.9043040293040293	0.8734177215189873
64	32	0.2	0	0.8692002442002442	0.8924050632911392
64	32	0.2	0.2	0.8862942612942613	0.879746835443038
64	32	0.4	0	0.913003663003663	0.8734177215189873
64	32	0.4	0.2	0.8531746031746031	0.8987341772151899
64	32	0.4	0.4	0.8539377289377289	0.8924050632911392
64	64	0	0	0.898962148962149	0.879746835443038
64	64	0.2	0	0.8231074481074481	0.9113924050632911
64	64	0.2	0.2	0.8746947496947497	0.8860759493670886
64	64	0.4	0	0.9116300366300366	0.879746835443038
64	64	0.4	0.2	0.8554639804639804	0.8860759493670886
64	64	0.4	0.4	0.8663003663003663	0.8860759493670886
64	128	0	0	0.8753052503052503	0.8987341772151899
64	128	0.2	0	0.8972832722832723	0.8924050632911392
64	128	0.2	0.2	0.8894993894993894	0.8670886075949367
64	128	0.4	0	0.8803418803418803	0.8860759493670886
64	128	0.4	0.2	0.827075702075702	0.8987341772151899
64	128	0.4	0.4	0.8388278388278388	0.8860759493670886
64	256	0	0	0.8782051282051282	0.8860759493670886
64	256	0.2	0	0.86492673992674	0.9050632911392406
64	256	0.2	0.2	0.8579059829059829	0.8860759493670886
64	256	0.4	0	0.8936202686202687	0.8860759493670886
64	256	0.4	0.2	0.8615689865689866	0.879746835443038
64	256	0.4	0.4	0.831959706959707	0.8924050632911392
128	16	0	0	0.884004884004884	0.879746835443038
128	16	0.2	0	0.92002442002442	0.8607594936708861
128	16	0.2	0.2	0.9084249084249084	0.8417721518987342
128	16	0.4	0	0.931013431013431	0.8544303797468354
128	16	0.4	0.2	0.9581807081807082	0.7848101265822784
128	16	0.4	0.4	0.8914835164835165	0.8544303797468354
128	32	0	0	0.8728632478632479	0.8924050632911392
128	32	0.2	0	0.8524114774114774	0.8924050632911392
128	32	0.2	0.2	0.8785103785103785	0.879746835443038
128	32	0.4	0	0.9232295482295483	0.8734177215189873
128	32	0.4	0.2	0.8855311355311355	0.8734177215189873
128	32	0.4	0.4	0.8485958485958486	0.8987341772151899
128	64	0	0	0.8826312576312576	0.879746835443038
128	64	0.2	0	0.8827838827838828	0.8860759493670886
128	64	0.2	0.2	0.9192612942612942	0.8354430379746836
128	64	0.4	0	0.86492673992674	0.8987341772151899

128	64	0.4	0.2	0.8804945054945055	0.879746835443038
128	64	0.4	0.4	0.8513431013431013	0.8924050632911392
128	128	0	0	0.8783577533577533	0.8987341772151899
128	128	0.2	0	0.8772893772893773	0.8924050632911392
128	128	0.2	0.2	0.8817155067155067	0.8670886075949367
128	128	0.4	0	0.8426434676434676	0.9113924050632911
128	128	0.4	0.2	0.8899572649572649	0.8734177215189873
128	128	0.4	0.4	0.8305860805860806	0.8924050632911392
128	256	0	0	0.8473748473748474	0.8924050632911392
128	256	0.2	0	0.8298229548229549	0.9113924050632911
128	256	0.2	0.2	0.8586691086691086	0.8860759493670886
128	256	0.4	0	0.887057387057387	0.8860759493670886
128	256	0.4	0.2	0.8698107448107448	0.879746835443038
128	256	0.4	0.4	0.8295177045177046	0.8924050632911392
256	16	0	0	0.938949938949939	0.8164556962025317
256	16	0.2	0	0.9078144078144078	0.8607594936708861
256	16	0.2	0.2	0.989010989010989	0.6518987341772152
256	16	0.4	0	0.9305555555555556	0.8544303797468354
256	16	0.4	0.2	0.8353174603174603	0.8987341772151899
256	16	0.4	0.4	0.8724053724053724	0.8670886075949367
256	32	0	0	0.9191086691086691	0.8670886075949367
256	32	0.2	0	0.8725579975579976	0.8860759493670886
256	32	0.2	0.2	0.8844627594627594	0.8734177215189873
256	32	0.4	0	0.9218559218559218	0.8670886075949367
256	32	0.4	0.2	0.8368437118437119	0.8987341772151899
256	32	0.4	0.4	0.8255494505494505	0.8987341772151899
256	64	0	0	0.8495115995115995	0.8924050632911392
256	64	0.2	0	0.9053724053724054	0.8860759493670886
256	64	0.2	0.2	0.8765262515262515	0.8924050632911392
256	64	0.4	0	0.8765262515262515	0.9050632911392406
256	64	0.4	0.2	0.8785103785103785	0.8860759493670886
256	64	0.4	0.4	0.8724053724053724	0.8860759493670886
256	128	0	0	0.8223443223443223	0.8987341772151899
256	128	0.2	0	0.9036935286935287	0.8860759493670886
256	128	0.2	0.2	0.9026251526251526	0.8670886075949367
256	128	0.4	0	0.8559218559218559	0.9113924050632911
256	128	0.4	0.2	0.8400488400488401	0.8924050632911392
256	128	0.4	0.4	0.8394383394383395	0.8987341772151899
256	256	0	0	0.8774420024420024	0.8987341772151899
256	256	0.2	0	0.8310439560439561	0.8987341772151899

256	256	0.2	0.2	0.8911782661782662	0.8734177215189873
256	256	0.4	0	0.8841575091575091	0.8860759493670886
256	256	0.4	0.2	0.8598901098901099	0.8860759493670886
256	256	0.4	0.4	0.8856837606837606	0.8670886075949367

Όπως φαίνεται τα αποτελέσματα βελτιώθηκαν πολύ για τις περιπτώσεις όπου το αποτέλεσμα ήταν θετικό. Παράλληλα όμως μικρυνε η ακρίβεια όσο αφορά τα μηδενικά. Τα καλύτερα αποτελέσματα σημειώθηκαν για τις εξής παραμέτρους:

- units 128
- batch 32
- recurrent dropout 0,4
- dropout 0

Με αυτές τις παραμέτρους φτάσαμε σε ακρίβεια 92,3% για μηδενικά και 87,3% για τους άσσους αντίστοιχα.

Το νευρωνικό δίκτυο δεν επιστρέφει απλές τιμές 0 και 1 αλλά επιστρέφει αριθμούς ανάμεσα στο 0 και το 1. Όσο πιο κοντα η τιμή βρίσκεται σε ένα από τα άκρα τόσο πιο σίγουρο είναι το δίκτυο για την πρόβλεψη του. Όλες οι παραπάνω προβλέψεις του συστήματος γίναν με threshold 0,5. Αυτό σημαίνει ότι εφόσον η πρόβλεψη του συστήματος είχε τιμή μεγαλύτερη του 0,5 τότε αυτή θεωρείται σαν πρόβλεψη άσσου, ενώ σε περίπτωση που η πρόβλεψη είχε τιμή μικρότερη ή ίση του 0,5 τότε αυτή θεωρείται ως πρόβλεψη μηδενικού. Αυτό μπορεί να φανεί και στο παρακάτω τμήμα κώδικα που υλοποιεί ακριβώς αυτήν την λειτουργία.

```
Preds = model.predict(test_samples_expanded, verbose=1)
preds = preds.reshape(len(preds),1)
predsnormalized = (preds > 0.5).astype(int)
```

Η συνάρτηση predict καλείται πάνω στα test data του mammography dataset και επιστρέφει τις προβλέψεις του εκπαιδευμένου συστήματος . Στην συνέχεια οι προβλέψεις μετατρέπονται σε ένα δισδιάστατο πίνακα. Στην

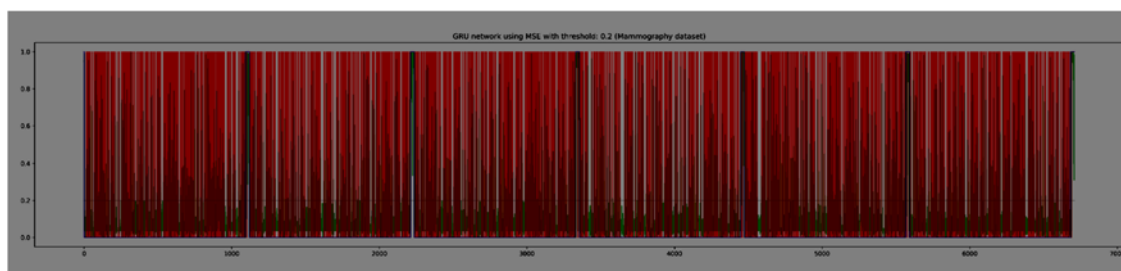
τελευταία γραμμή φαίνεται η μετατροπή των προβλέψεων του συστήματος σε μηδέν και ένα βάση του threshold το οποίο στην συγκεκριμένη περίπτωση έχει την τιμή 0,5.

Πέρα των αποτελεσμάτων του συστήματος σε μορφή πίνακα, πολλές φορές είναι χρήσιμο τα αποτελέσματα να απεικονιστούν με σκοπό την καλύτερη κατανόηση των ίδιων των αποτελεσμάτων. Παρακάτω φαίνονται οι απεικονίσεις των προβλέψεων του συστήματος για διαφορετικές τιμές threshold.

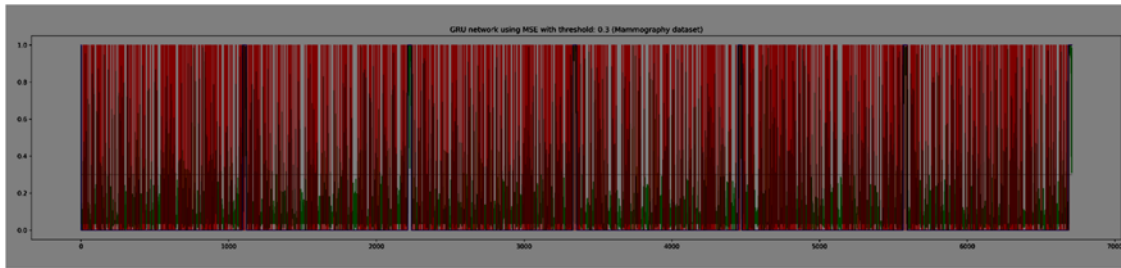
Εξήγηση των γράφων:

- Γκρι οριζόντια γραμμή: Απεικονίζει το threshold πάνω στον γράφο.
- Πράσινος παλμός: Απεικονίζει την έξοδο του νευρωνικού δικτύου.
- Κόκκινος παλμός: Απεικονίζει το αποτέλεσμα της πρόβλεψης του συστήματος μετά την εφαρμογή του threshold.
- Μπλε παλμός: Απεικονίζει τα πραγματικά σωστά αποτελέσματα.

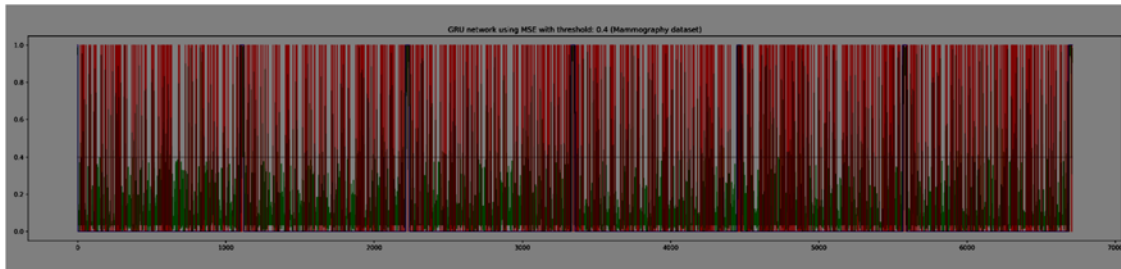
Βάση των παραπάνω σκοπός είναι οι κόκκινοι παλμοί μοιάζουν όσο το περισσότερο δυνατόν με τους μπλε παλμούς.



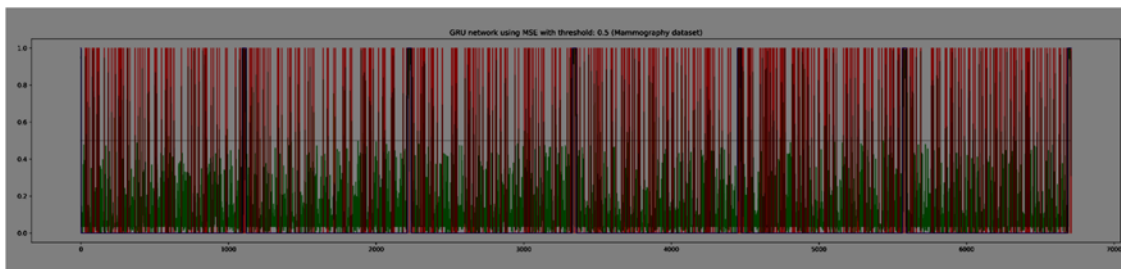
Απεικόνιση αποτελεσμάτων για threshold 0,2



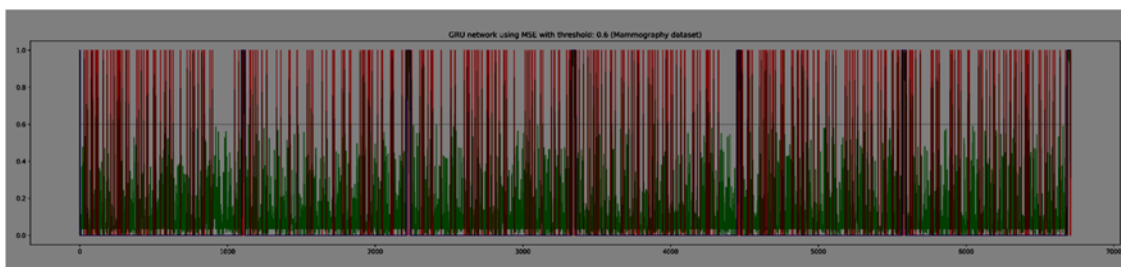
Απεικόνιση αποτελεσμάτων για threshold 0,3



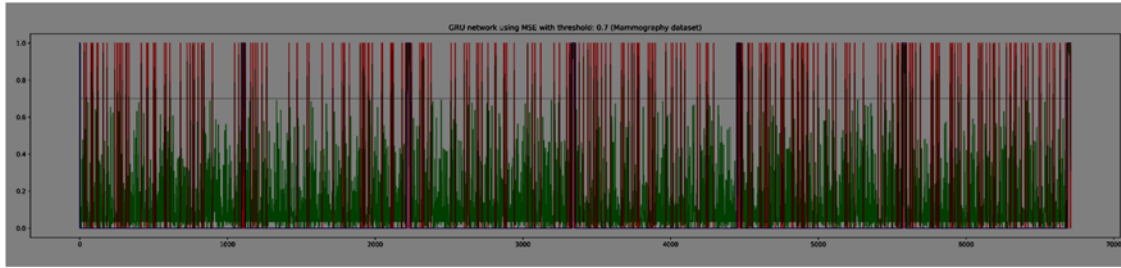
Απεικόνιση αποτελεσμάτων για threshold 0,4



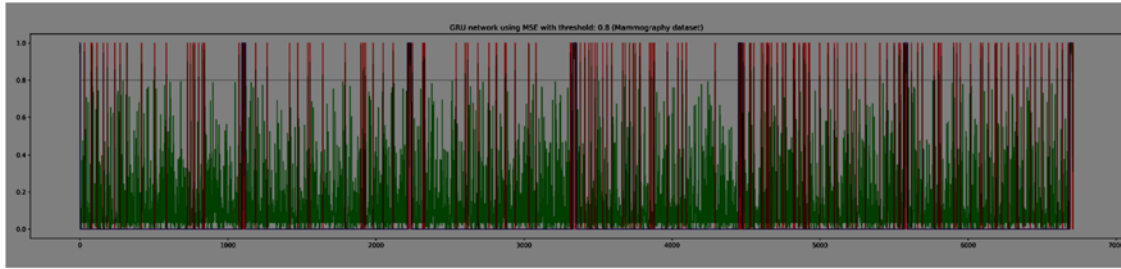
Απεικόνιση αποτελεσμάτων για threshold 0,5



Απεικόνιση αποτελεσμάτων για threshold 0,6



Απεικόνιση αποτελεσμάτων για threshold 0,7



Απεικόνιση αποτελεσμάτων για threshold 0,8

Αν και τα καλύτερα αποτελέσματα φαίνονται να είναι αυτά στην τελευταία εικόνα, στην πραγματικότητα η μεγαλύτερη ισορροπία όσον αφορά την πρόβλεψη άσων και μηδενικών υπάρχει στην περίπτωση όπου το threshold έχει τιμή 0,5. Αυτό για άλλη μια φορά συμβαίνει λόγω του πολύ μεγάλου αριθμού μηδενικών σε σχέση με τους άσους. Αν και για threshold 0,8 το σύστημα μπορεί να αγγίζει πολύ μεγάλη ακρίβεια στην πραγματικότητα αυτό ισχύει μόνο για τα μηδενικά ενώ η πρόβλεψη των άσων θα βρίσκεται γύρω στο 50%.

Units	Batch	Rec Drop	Dropout	Zeros_prediction_acc	Ones_prediction_acc
16	16	0	0	0.9935897435897436	0.5949367088607594
16	16	0.2	0	0.9914529914529915	0.6139240506329114
16	16	0.2	0.2	0.9906898656898657	0.6582278481012658
16	16	0.4	0	0.9891636141636142	0.6708860759493671
16	16	0.4	0.2	0.9905372405372406	0.6582278481012658
16	16	0.4	0.4	0.9913003663003663	0.5822784810126582
16	32	0	0	0.9818376068376068	0.7151898734177216
16	32	0.2	0	0.9880952380952381	0.7151898734177216
16	32	0.2	0.2	0.9841269841269841	0.740506329113924
16	32	0.4	0	0.9853479853479854	0.7341772151898734
16	32	0.4	0.2	0.987942612942613	0.7088607594936709

16	32	0.4	0.4	0.9894688644688645	0.6329113924050633
16	64	0	0	0.9891636141636142	0.7278481012658228
16	64	0.2	0	0.9938949938949939	0.6265822784810127
16	64	0.2	0.2	0.9803113553113553	0.740506329113924
16	64	0.4	0	0.9917582417582418	0.6772151898734177
16	64	0.4	0.2	0.9824481074481074	0.740506329113924
16	64	0.4	0.4	0.9691697191697192	0.7468354430379747
16	128	0	0	0.9942002442002442	0.620253164556962
16	128	0.2	0	0.9891636141636142	0.7341772151898734
16	128	0.2	0.2	0.9862637362637363	0.7151898734177216
16	128	0.4	0	0.9853479853479854	0.740506329113924
16	128	0.4	0.2	0.9784798534798534	0.7215189873417721
16	128	0.4	0.4	0.9851953601953602	0.7088607594936709
16	256	0	0	0.985958485958486	0.7215189873417721
16	256	0.2	0	0.9870268620268621	0.7215189873417721
16	256	0.2	0.2	0.9803113553113553	0.7215189873417721
16	256	0.4	0	0.9856532356532357	0.7341772151898734
16	256	0.4	0.2	0.9755799755799756	0.740506329113924
16	256	0.4	0.4	0.9905372405372406	0.6518987341772152

Παράδειγμα μετρήσεων για threshold 0,8. Η ακρίβεια πρόβλεψης άσων μειώθηκε αρκετά.

Μέχρι στιγμής σαν ακρίβεια χρησιμοποιούσαμε το ποσοστό των σωστών προβλέψεων του συστήματος. Στην συνέχεια χρησιμοποιήθηκε και η μέθοδος f-score. Τα αποτελέσματα με χρήση f-score φαίνονται στον παρακάτω πίνακα.

Unit s	Batc h	Rec drop	Dropo ut	Zeros acc	Ones acc	Precisi on	Recall	f1-score
16	16	0	0	0.930250 30525030 53	0.822784 8101265 823	0.22146 507666 098808	0.82278 481012 65823	0.34899 3288590 604
16	16	0.2	0	0.916514 04151404 15	0.848101 2658227 848	0.19676 945668 135096	0.84810 126582 27848	0.31942 7890345 64956
16	16	0.2	0.2	0.926129 42612942 62	0.822784 8101265 823	0.21172 638436 482086	0.82278 481012 65823	0.33678 7564766 8394
16	16	0.4	0	0.924297 92429792 43	0.854430 3797468 354	0.21394 611727 416799	0.85443 037974 68354	0.34220 5323193 91633
16	16	0.4	0.2	0.915140 41514041 51	0.860759 4936708 861	0.19653 179190 751446	0.86075 949367 08861	0.32
16	16	0.4	0.4	0.907356 53235653 24	0.822784 8101265 823	0.17639 077340 569878	0.82278 481012 65823	0.29050 2793296 0894

16	32	0	0	0.893772 89377289 38	0.886075 9493670 886	0.16746 411483 25359	0.88607 594936 70886	0.28169 0140845 0705
16	32	0.2	0	0.908882 78388278 39	0.854430 3797468 354	0.18442 622950 819673	0.85443 037974 68354	0.30337 0786516 85395
16	32	0.2	0.2	0.836385 83638583 64	0.898734 1772151 899	0.11696 869851 729819	0.89873 417721 51899	0.20699 7084548 10498
16	32	0.4	0	0.920482 29548229 54	0.873417 7215189 873	0.20940 819423 36874	0.87341 772151 89873	0.33782 1297429 6205
16	32	0.4	0.2	0.870573 87057387 06	0.879746 8354430 38	0.14083 080040 52685	0.87974 683544 3038	0.24279 4759825 3275
16	32	0.4	0.4	0.881562 88156288 15	0.873417 7215189 873	0.15098 468271 334792	0.87341 772151 89873	0.25746 2686567 1642
16	64	0	0	0.882631 25763125 76	0.898734 1772151 899	0.15587 266739 846323	0.89873 417721 51899	0.26566 8849391 9551
16	64	0.2	0	0.887057 38705738 7	0.898734 1772151 899	0.16099 773242 630386	0.89873 417721 51899	0.27307 6923076 9231
16	64	0.2	0.2	0.878815 62881562 88	0.886075 9493670 886	0.14989 293361 884368	0.88607 594936 70886	0.25641 0256410 25644
16	64	0.4	0	0.893925 51892551 9	0.898734 1772151 899	0.16965 352449 223417	0.89873 417721 51899	0.28542 7135678 39195
16	64	0.4	0.2	0.885073 26007326 38	0.879746 8354430 255606	0.15582 959641 3038	0.87974 683544 3038	0.26476 1904761 9048
16	64	0.4	0.4	0.860653 23565323 57	0.873417 7215189 873	0.13130 352045 67079	0.87341 772151 89873	0.22828 7841191 06701
16	128	0	0	0.872252 74725274 73	0.905063 2911392 406	0.14591 836734 693878	0.90506 329113 92406	0.25131 8101933 2162
16	128	0.2	0	0.888583 63858363 85	0.892405 0632911 392	0.16188 289322 61768	0.89240 506329 11392	0.27405 2478134 11077
16	128	0.2	0.2	0.853937 72893772 89	0.886075 9493670 886	0.12762 078395 62443	0.88607 594936 70886	0.22310 7569721 1155
16	128	0.4	0	0.880952 38095238 09	0.898734 1772151 899	0.15401 301518 43818	0.89873 417721 51899	0.26296 2962962 96295
16	128	0.4	0.2	0.853327 22832722 83	0.892405 0632911 392	0.12794 918330 30853	0.89240 506329 11392	0.22380 9523809 52382
16	128	0.4	0.4	0.810439 56043956 04	0.898734 1772151 899	0.10260 115606 936417	0.89873 417721 51899	0.18417 6394293 12582
16	256	0	0	0.870573 87057387 06	0.879746 8354430 38	0.14083 080040 52685	0.87974 683544 3038	0.24279 4759825 3275
16	256	0.2	0	0.871642 24664224 67	0.886075 9493670 886	0.14271 151885 830785	0.88607 594936 70886	0.24582 9675153 64354

16	256	0.2	0.2	0.880647 13064713 06	0.879746 8354430 38	0.15092 290988 056462	0.87974 683544 3038	0.25764 5968489 342
16	256	0.4	0	0.864010 98901098 9	0.886075 9493670 886	0.13579 049466 537343	0.88607 594936 70886	0.23549 2010092 5147
16	256	0.4	0.2	0.858821 73382173 38	0.886075 9493670 886	0.13145 539906 103287	0.88607 594936 70886	0.22894 5216680 29436
16	256	0.4	0.4	0.829212 45421245 43	0.886075 9493670 886	0.11119 936457 505956	0.88607 594936 70886	0.19760 0564573 04162
32	16	0	0	0.927655 67765567 77	0.829113 9240506 329	0.21652 892561 983472	0.82911 392405 06329	0.34338 1389252 94887
32	16	0.2	0	0.903388 27838827 84	0.873417 7215189 873	0.17898 832684 824903	0.87341 772151 89873	0.29709 3649085 03766
32	16	0.2	0.2	0.916208 79120879 12	0.854430 3797468 354	0.19736 842105 263158	0.85443 037974 68354	0.32066 5083135 39194
32	16	0.4	0	0.912240 53724053 72	0.854430 3797468 354	0.19014 084507 042253	0.85443 037974 68354	0.31105 9907834 1013
32	16	0.4	0.2	0.956959 70695970 7	0.810126 5822784 81	0.31219 512195 12195	0.81012 658227 8481	0.45070 4225352 11263
32	16	0.4	0.4	0.931166 05616605 62	0.803797 4683544 303	0.21972 318339 100347	0.80379 746835 44303	0.34510 8695652 1739
32	32	0	0	0.884920 63492063 49	0.898734 1772151 899	0.15848 214285 714285	0.89873 417721 51899	0.26944 9715370 01894
32	32	0.2	0	0.920329 67032967 03	0.867088 6075949 367	0.20789 074355 08346	0.86708 860759 49367	0.33537 3317013 46385
32	32	0.2	0.2	0.926282 05128205 13	0.848101 2658227 848	0.21717 990275 52674	0.84810 126582 27848	0.34580 6451612 90324
32	32	0.4	0	0.893162 39316239 32	0.886075 9493670 886	0.16666 666666 666666	0.88607 594936 70886	0.28056 1122244 4889
32	32	0.4	0.2	0.904456 65445665 45	0.860759 4936708 861	0.17847 769028 87139	0.86075 949367 08861	0.29565 2173913 0435
32	32	0.4	0.4	0.882478 63247863 25	0.867088 6075949 367	0.15104 740904 079383	0.86708 860759 49367	0.25727 6995305 16435
32	64	0	0	0.889041 51404151 4	0.886075 9493670 886	0.16147 635524 798154	0.88607 594936 70886	0.27317 0731707 3171
32	64	0.2	0	0.871794 87179487 18	0.905063 2911392 406	0.14547 304170 905392	0.90506 329113 92406	0.25065 7318141 9807
32	64	0.2	0.2	0.866758 24175824 18	0.886075 9493670 886	0.13820 335636 722605	0.88607 594936 70886	0.23911 1870196 4133
32	64	0.4	0	0.914377 28937728 93	0.886075 9493670 886	0.19971 469329 529243	0.88607 594936 70886	0.32596 0419091 96743

32	64	0.4	0.2	0.914529 91452991 45	0.854430 3797468 354	0.19424 460431 654678	0.85443 037974 68354	0.31652 9894490 0352
32	64	0.4	0.4	0.844322 34432234 43	0.898734 1772151 899	0.12220 309810 671257	0.89873 417721 51899	0.21515 1515151 51518
32	128	0	0	0.866147 74114774 12	0.886075 9493670 886	0.13765 978367 74828	0.88607 594936 70886	0.23829 7872340 42554
32	128	0.2	0	0.892246 64224664 23	0.892405 0632911 392	0.16646 989374 262103	0.89240 506329 11392	0.28059 7014925 37314
32	128	0.2	0.2	0.885989 01098901 1	0.867088 6075949 367	0.15497 737556 561086	0.86708 860759 49367	0.26295 5854126 6795
32	128	0.4	0	0.890262 51526251 52	0.898734 1772151 899	0.16492 450638 792103	0.89873 417721 51899	0.27870 4612365 0638
32	128	0.4	0.2	0.872557 99755799 76	0.886075 9493670 886	0.14358 974358 974358	0.88607 594936 70886	0.24713 1509267 43155
32	128	0.4	0.4	0.831349 20634920 64	0.892405 0632911 392	0.11316 211878 009631	0.89240 506329 11392	0.20085 4700854 7009
32	256	0	0	0.871642 24664224 67	0.898734 1772151 899	0.14445 574771 108852	0.89873 417721 51899	0.24890 4469763 36547
32	256	0.2	0	0.878663 00366300 36	0.892405 0632911 392	0.15064 102564 102563	0.89240 506329 11392	0.25776 9652650 82263
32	256	0.2	0.2	0.871947 49694749 7	0.879746 8354430 38	0.14212 678936 605316	0.87974 683544 3038	0.24471 8309859 1549
32	256	0.4	0	0.876984 12698412 7	0.886075 9493670 886	0.14799 154334 038056	0.88607 594936 70886	0.25362 3188405 7971
32	256	0.4	0.2	0.882326 00732600 73	0.873417 7215189 873	0.15181 518151 815182	0.87341 772151 89873	0.25866 9165885 66075
32	256	0.4	0.4	0.826465 20146520 14	0.892405 0632911 392	0.11032 863849 765258	0.89240 506329 11392	0.19637 8830083 56546
64	16	0	0	0.929639 80463980 47	0.841772 1518987 342	0.22390 572390 57239	0.84177 215189 87342	0.35372 3404255 3192
64	16	0.2	0	0.894078 14407814 41	0.867088 6075949 367	0.16486 161251 50421	0.86708 860759 49367	0.27704 7522750 25274
64	16	0.2	0.2	0.911630 03663003 66	0.854430 3797468 354	0.18907 563025 210083	0.85443 037974 68354	0.30963 3027522 9358
64	16	0.4	0	0.921245 42124542 12	0.854430 3797468 354	0.20737 327188 94009	0.85443 037974 68354	0.33374 5364647 7132
64	16	0.4	0.2	0.899267 39926739 93	0.860759 4936708 861	0.17085 427135 67839	0.86075 949367 08861	0.28511 5303983 22854
64	16	0.4	0.4	0.917429 79242979 24	0.829113 9240506 329	0.19494 047619 04762	0.82911 392405 06329	0.31566 2650602 4096

64	32	0	0	0.888736 26373626 37	0.892405 0632911 392	0.16206 896551 724137	0.89240 506329 11392	0.27431 9066147 85995
64	32	0.2	0	0.909645 90964590 96	0.873417 7215189 873	0.18904 109589 041096	0.87341 772151 89873	0.31081 0810810 8108
64	32	0.2	0.2	0.870879 12087912 09	0.886075 9493670 886	0.14198 782961 460446	0.88607 594936 70886	0.24475 5244755 24474
64	32	0.4	0	0.892551 89255189 26	0.886075 9493670 886	0.16587 677725 118483	0.88607 594936 70886	0.27944 1117764 47106
64	32	0.4	0.2	0.858974 35897435 89	0.886075 9493670 886	0.13157 894736 842105	0.88607 594936 70886	0.22913 2569558 10143
64	32	0.4	0.4	0.881105 00610500 61	0.879746 8354430 38	0.15141 612200 43573	0.87974 683544 3038	0.25836 4312267 658
64	64	0	0	0.881257 63125763 12	0.898734 1772151 899	0.15434 782608 695652	0.89873 417721 51899	0.26345 0834879 4063
64	64	0.2	0	0.854853 47985347 98	0.898734 1772151 899	0.12991 765782 250686	0.89873 417721 51899	0.22701 8385291 7666
64	64	0.2	0.2	0.891788 76678876 68	0.854430 3797468 354	0.15995 260663 507108	0.85443 037974 68354	0.26946 1077844 3114
64	64	0.4	0	0.876221 00122100 12	0.879746 8354430 38	0.14631 578947 36842	0.87974 683544 3038	0.25090 2527075 8123
64	64	0.4	0.2	0.913766 78876678 87	0.841772 1518987 342	0.19054 441260 744986	0.84177 215189 87342	0.31074 7663551 4019
64	64	0.4	0.4	0.845238 09523809 52	0.892405 0632911 392	0.12207 792207 792208	0.89240 506329 11392	0.21477 5323686 21477
64	128	0	0	0.861111 11111111 12	0.892405 0632911 392	0.13415 794481 446242	0.89240 506329 11392	0.23325 0620347 39452
64	128	0.2	0	0.885683 76068376 06	0.892405 0632911 392	0.15842 696629 213482	0.89240 506329 11392	0.26908 3969465 64883
64	128	0.2	0.2	0.873931 62393162 39	0.879746 8354430 38	0.14404 145077 720207	0.87974 683544 3038	0.24755 1202137 13264
64	128	0.4	0	0.871642 24664224 67	0.892405 0632911 392	0.14358 452138 492872	0.89240 506329 11392	0.24736 8421052 6316
64	128	0.4	0.2	0.854700 85470085 47	0.886075 9493670 886	0.12820 512820 51282	0.88607 594936 70886	224
64	128	0.4	0.4	0.834859 58485958 49	0.898734 1772151 899	0.11601 307189 542484	0.89873 417721 51899	0.20549 9276410 99858
64	256	0	0	0.893315 01831501 84	0.892405 0632911 392	0.16785 714285 714284	0.89240 506329 11392	0.28256 5130260 52105
64	256	0.2	0	0.877747 25274725 27	0.905063 2911392 406	0.15148 305084 745764	0.90506 329113 92406	0.25952 8130671 5063

64	256	0.2	0.2	0.899114 77411477 41	0.860759 4936708 861	0.17063 989962 358847	0.86075 949367 08861	0.28481 6753926 70164
64	256	0.4	0	0.869505 49450549 45	0.886075 9493670 886	0.14070 351758 79397	0.88607 594936 70886	0.24284 4752818 73374
64	256	0.4	0.2	0.880952 38095238 09	0.873417 7215189 873	0.15032 679738 56209	0.87341 772151 89873	0.25650 5576208 17843
64	256	0.4	0.4	0.879731 37973137 97	0.867088 6075949 367	0.14810 810810 810812	0.86708 860759 49367	0.25300 0923361 0342
128	16	0	0	0.862179 48717948 72	0.898734 1772151 899	0.13588 516746 411483	0.89873 417721 51899	0.23607 6475477 97173
128	16	0.2	0	0.853174 60317460 31	0.898734 1772151 899	0.12862 318840 57971	0.89873 417721 51899	0.22503 9619651 34706
128	16	0.2	0.2	0.858974 35897435 89	0.892405 0632911 392	0.13239 436619 71831	0.89240 506329 11392	0.23058 0539656 58218
128	16	0.4	0	0.906593 40659340 66	0.860759 4936708 861	0.18181 818181 818182	0.86075 949367 08861	0.30022 0750551 8764
128	16	0.4	0.2	0.913308 91330891 33	0.848101 2658227 848	0.19088 319088 31909	0.84810 126582 27848	0.31162 7906976 74425
128	16	0.4	0.4	0.821886 44688644 69	0.892405 0632911 392	0.10779 816513 761468	0.89240 506329 11392	0.19236 0163710 77766
128	32	0	0	0.894536 01953601 96	0.886075 9493670 886	0.16847 172081 82912	0.88607 594936 70886	0.28311 4256825 0758
128	32	0.2	0	0.935286 93528693 53	0.835443 0379746 836	0.23741 007194 244604	0.83544 303797 46836	0.36974 7899159 66383
128	32	0.2	0.2	0.886446 88644688 64	0.867088 6075949 367	0.15550 510783 200908	0.86708 860759 49367	0.26371 5110683 34935
128	32	0.4	0	0.883547 00854700 85	0.892405 0632911 392	0.15597 345132 743362	0.89240 506329 11392	0.26553 6723163 84175
128	32	0.4	0.2	0.918650 79365079 36	0.835443 0379746 836	0.19849 624060 150375	0.83544 303797 46836	0.32077 7642770 3523
128	32	0.4	0.4	0.787851 03785103 78	0.898734 1772151 899	0.09268 929503 91645	0.89873 417721 51899	0.16804 7337278 10653
128	64	0	0	0.866605 61660561 66	0.892405 0632911 392	0.13891 625615 763548	0.89240 506329 11392	0.24040 9207161 12532
128	64	0.2	0	0.903235 65323565 32	0.898734 1772151 899	0.18298 969072 164947	0.89873 417721 51899	0.30406 8522483 94006
128	64	0.2	0.2	0.889346 76434676 43	0.867088 6075949 367	0.15893 271461 716937	0.86708 860759 49367	0.26862 7450980 3922
128	64	0.4	0	0.907203 90720390 72	0.886075 9493670 886	0.18716 577540 106952	0.88607 594936 70886	0.30905 0772626 93154

128	64	0.4	0.2	0.852411 47741147 74	0.898734 1772151 899	0.12804 328223 624886	0.89873 417721 51899	0.22415 1539068 66614
128	64	0.4	0.4	0.851190 47619047 62	0.886075 9493670 886	0.12556 053811 659193	0.88607 594936 70886	0.21995 2867242 7337
128	128	0	0	0.894841 26984126 99	0.892405 0632911 392	0.16987 951807 228915	0.89240 506329 11392	0.28542 5101214 57493
128	128	0.2	0	0.890720 39072039 08	0.886075 9493670 886	0.16355 140186 915887	0.88607 594936 70886	0.27613 4122287 9684
128	128	0.2	0.2	0.857448 10744810 74	0.886075 9493670 886	0.13035 381750 46555	0.88607 594936 70886	0.22727 2727272 72727
128	128	0.4	0	0.862179 48717948 72	0.911392 4050632 911	0.13753 581661 891118	0.91139 240506 32911	0.23900 4149377 59337
128	128	0.4	0.2	0.858821 73382173 38	0.886075 9493670 886	0.13145 539906 103287	0.88607 594936 70886	0.22894 5216680 29436
128	128	0.4	0.4	0.879578 75457875 46	0.860759 4936708 861	0.14702 702702 702702	0.86075 949367 08861	0.25115 4201292 70544
128	256	0	0	0.884310 13431013 43	0.886075 9493670 886	0.15590 200445 4343	0.88607 594936 70886	0.26515 1515151 5152
128	256	0.2	0	0.895299 14529914 53	0.886075 9493670 886	0.16949 152542 37288	0.88607 594936 70886	0.28455 2845528 4553
128	256	0.2	0.2	0.873168 49816849 82	0.879746 8354430 38	0.14329 896907 216494	0.87974 683544 3038	0.24645 3900709 21983
128	256	0.4	0	0.885531 13553113 55	0.898734 1772151 899	0.15919 282511 21076	0.89873 417721 51899	0.27047 6190476 1905
128	256	0.4	0.2	0.859279 60927960 92	0.886075 9493670 886	0.13182 674199 62335	0.88607 594936 70886	0.22950 8196721 31145
128	256	0.4	0.4	0.819291 81929181 93	0.898734 1772151 899	0.10708 898944 193061	0.89873 417721 51899	0.19137 4663072 77629
256	16	0	0	0.866300 36630036 63	0.898734 1772151 899	0.13948 919449 901767	0.89873 417721 51899	0.24149 6598639 45577
256	16	0.2	0	0.906898 65689865 69	0.873417 7215189 873	0.18449 197860 962566	0.87341 772151 89873	0.30463 5761589 4039
256	16	0.2	0.2	0.876984 12698412 7	0.879746 8354430 38	0.14708 994708 994708	0.87974 683544 3038	0.25203 9891205 80234
256	16	0.4	0	0.877747 25274725 27	0.886075 9493670 886	0.14877 789585 54729	0.88607 594936 70886	0.25477 7070063 69427
256	16	0.4	0.2	0.938949 93894993 9	0.803797 4683544 303	0.24098 671726 75522	0.80379 746835 44303	0.37080 2919708 02914
256	16	0.4	0.4	0.828144 07814407 82	0.898734 1772151 899	0.11198 738170 347003	0.89873 417721 51899	0.19915 8485273 49225

256	32	0	0	0.894383 39438339 44	0.879746 8354430 38	0.16726 835138 387486	0.87974 683544 3038	0.28109 2012133 4682
256	32	0.2	0	0.907203 90720390 72	0.873417 7215189 873	0.18498 659517 426275	0.87341 772151 89873	0.30530 9734513 2743
256	32	0.2	0.2	0.899267 39926739 93	0.873417 7215189 873	0.17293 233082 706766	0.87341 772151 89873	0.28870 2928870 2929
256	32	0.4	0	0.861416 36141636 15	0.892405 0632911 392	0.13441 372735 93899	0.89240 506329 11392	0.23363 7116818 5584
256	32	0.4	0.2	0.889041 51404151 4	0.879746 8354430 38	0.16050 808314 08776	0.87974 683544 3038	0.27148 4375000 00006
256	32	0.4	0.4	0.912393 16239316 24	0.835443 0379746 836	0.18696 883852 691218	0.83544 303797 46836	0.30555 5555555 5555
256	64	0	0	0.908424 90842490 84	0.879746 8354430 38	0.18809 201623 81597	0.87974 683544 3038	0.30992 1962095 8752
256	64	0.2	0	0.905219 78021978 02	0.879746 8354430 38	0.18289 473684 210528	0.87974 683544 3038	0.30283 2244008 7146
256	64	0.2	0.2	0.903998 77899877 9	0.841772 1518987 342	0.17454 068241 469817	0.84177 215189 87342	0.28913 0434782 6087
256	64	0.4	0	0.878357 75335775 33	0.905063 2911392 406	0.15212 765957 446808	0.90506 329113 92406	0.26047 3588342 4408
256	64	0.4	0.2	0.899420 02442002 44	0.873417 7215189 873	0.17314 930991 217065	0.87341 772151 89873	0.28900 5235602 0943
256	64	0.4	0.4	0.835927 96092796 1	0.898734 1772151 899	0.11668 036154 478226	0.89873 417721 51899	0.20654 5454545 45456
256	128	0	0	0.890415 14041514 04	0.886075 9493670 886	0.16317 016317 016317	0.88607 594936 70886	0.27559 051181 1024
256	128	0.2	0	0.848595 84859584 86	0.892405 0632911 392	0.12444 836716 681378	0.89240 506329 11392	0.21843 5321456 23545
256	128	0.2	0.2	0.893315 01831501 84	0.886075 9493670 886	0.16686 531585 220502	0.88607 594936 70886	0.28084 2527582 74825
256	128	0.4	0	0.888736 26373626 37	0.905063 2911392 406	0.16399 082568 807338	0.90506 329113 92406	0.27766 9902912 62134
256	128	0.4	0.2	0.865079 36507936 51	0.886075 9493670 886	0.13671 875 875	0.88607 594936 70886	0.23688 6632825 71915
256	128	0.4	0.4	0.908119 65811965 81	0.835443 0379746 836	0.17983 651226 158037	0.83544 303797 46836	0.29596 4125560 5381
256	256	0	0	0.869200 24420024 42	0.886075 9493670 886	0.14042 126379 137412	0.88607 594936 70886	0.24242 4242424 24243
256	256	0.2	0	0.902472 52747252 75	0.879746 8354430 38	0.17866 323907 455012	0.87974 683544 3038	0.29700 8547008 54695

256	256	0.2	0.2	0.854853 47985347 98	0.886075 9493670 886	0.12832 263978 001834	0.88607 594936 70886	0.22417 9343474 77983
256	256	0.4	0	0.880952 38095238 09	0.905063 2911392 406	0.15492 957746 478872	0.90506 329113 92406	0.26456 9842738 20533
256	256	0.4	0.2	0.863553 11355311 36	0.886075 9493670 886	0.13539 651837 524178	0.88607 594936 70886	0.23489 9328859 06037
256	256	0.4	0.4	0.867368 74236874 24	0.873417 7215189 873	0.13704 071499 503476	0.87341 772151 89873	0.23690 9871244 6352

Ιδανικά το f-score έχει την τιμή 1 ενώ στην χειρότερη το 0. Όπως φαίνεται παραπάνω το f-score έχει τιμές γύρω από το 0,2 με 0,3, ας δούμε όμως γιατί συμβαίνει αυτό.

Το Recall στην ουσία είναι το ποσοστό σωστών προβλέψεων των άσπων. Είναι δηλαδή το ίδιο πράγμα με το Ones accuracy. Όπως φαίνεται το Recall είναι αρκετά ικανοποιητικό και κυμαίνεται γύρω στο 0,85 με 0,90.

Το Precision αφορά το ποσοστό των σωστών προβλέψεων άσπων σε σχέση με το σύνολο όλων των αποτελεσμάτων που η πρόβλεψη ήταν άσπος. Απαντά επομένως στο ερώτημα “Από τα αποτελέσματα όπου η πρόβλεψη είχε τιμή 1 πόσες προβλέψεις ήταν σωστες;”. Ο μικρός αριθμός του precision μας δείχνει ότι το σύστημα πλεόν κάνει πολύ περισσότερες θετικές προβλέψεις κάνοντας όμως αρκετά λάθη.

Συνοψίζοντας, με τις αλλαγές που κάναμε το νευρωνικό δίκτυο έχει βελτιώσει αρκετά την ακρίβεια του για τις θετικές προβλέψεις άλλα παράλληλα έχουν αυξηθεί αρκετά οι άστοχες θετικές προβλέψεις μειώνοντας πολύ όμως την συνολική ακρίβεια του συστήματος.

3.2.2.3 Αποτελέσματα συμπιεσμένων εισόδων

Στην συνέχεια εφαρμόζοντας την ίδια λογική για τα συμπιεσμένα με την μέθοδο PCA δεδομένα με στόχο την σύγκριση τους με τα αποτελέσματα των μη συμπιεσμένων δεδομένων πήραμε τα παρακάτω αποτελέσματα.

Un	Bat	Rec	Dropou	Zeros	Ones	Precision	Recall	f1-score
----	-----	-----	--------	-------	------	-----------	--------	----------

its	ch	drop	t	acc	acc			
16	16	0	0	0.8640 109890 10989	0.841772 1518987 342	0.12988281 25	0.84177 2151898 7342	0.225042 3011844 3316
16	16	0.2	0	0.8605 006105 006106	0.854430 3797468 354	0.12869399 428026693	0.85443 0379746 8354	0.223695 1118475 5592
16	16	0.2	0.2	0.8794 261294 261294	0.810126 5822784 81	0.13943355 11982571	0.81012 6582278 481	0.237918 2156133 8292
16	16	0.4	0	0.8957 570207 570208	0.797468 3544303 798	0.15574783 68355995	0.79746 8354430 3798	0.260599 7931747 6734
16	16	0.4	0.2	0.8733 211233 211233	0.835443 0379746 836	0.13721413 721413722	0.83544 3037974 6836	0.235714 2857142 857
16	16	0.4	0.4	0.8981 990231 990232	0.803797 4683544 303	0.15994962 216624686	0.80379 7468354 4303	0.266806 7226890 756
16	32	0	0	0.8780 525030 52503	0.841772 1518987 342	0.14270386 26609442	0.84177 2151898 7342	0.244036 6972477 064
16	32	0.2	0	0.8485 958485 958486	0.873417 7215189 873	0.12212389 380530973	0.87341 7721518 9873	0.214285 7142857 143
16	32	0.2	0.2	0.8618 742368 742369	0.841772 1518987 342	0.12813102 1194605	0.84177 2151898 7342	0.222408 0267558 5286
16	32	0.4	0	0.8490 537240 53724	0.873417 7215189 873	0.12244897 959183673	0.87341 7721518 9873	0.214785 9922178 9882
16	32	0.4	0.2	0.8482 905982 905983	0.879746 8354430 38	0.12268314 210061783	0.87974 6835443 038	0.215336 9481022 4632
16	32	0.4	0.4	0.8725 579975 579976	0.816455 6962025 317	0.13381742 73858921	0.81645 5696202 5317	0.229946 5240641 7113
16	64	0	0	0.8412 698412 698413	0.886075 9493670 886	0.11864406 779661017	0.88607 5949367 0886	0.209267 5635276 5322
16	64	0.2	0	0.8659 951159 95116	0.867088 6075949 367	0.13497536 945812807	0.86708 8607594 9367	0.233589 0878090 3664
16	64	0.2	0.2	0.8484 432234 432234	0.873417 7215189 873	0.12201591 51193634	0.87341 7721518 9873	0.214119 4724592 7074
16	64	0.4	0	0.8325 702075 702076	0.892405 0632911 392	0.11389337 641357028	0.89240 5063291 1392	0.202005 7306590 2583
16	64	0.4	0.2	0.8821 733821 733821	0.822784 8101265 823	0.14412416 851441243	0.82278 4810126 5823	0.245283 0188679 2453
16	64	0.4	0.4	0.8640 109890 10989	0.816455 6962025 317	0.12647058 82352941	0.81645 5696202 5317	0.219015 2801358 2341
16	128	0	0	0.8499 694749 69475	0.879746 8354430 38	0.12388591 800356506	0.87974 6835443 038	0.217187 5
16	128	0.2	0	0.8438	0.879746	0.11962134	0.87974	0.210606

				644688	8354430	251290878	6835443	0606060
				644689	38		038	606
				0.8637	0.848101	0.13047711	0.84810	0.226160
16	128	0.2	0.2	057387	2658227	781888996	1265822	3375527
				057387	848		7848	426
				0.8630	0.879746	0.13416988	0.87974	0.232830
16	128	0.4	0	952380	8354430	416988418	6835443	8207705
				952381	38		038	1928
				0.8437	0.879746	0.11951848	0.87974	0.210446
16	128	0.4	0.2	118437	8354430	667239896	6835443	6313398
				118437	38		038	9402
				0.8438	0.854430	0.11658031	0.85443	0.205167
16	128	0.4	0.4	644688	3797468	088082901	0379746	1732522
				644689	354		8354	7962
				0.8450	0.886075	0.12121212	0.88607	0.213252
16	256	0	0	854700	9493670	121212122	5949367	0944402
				854701	886		0886	1323
				0.8414	0.892405	0.11949152	0.89240	0.210762
16	256	0.2	0	224664	0632911	542372882	5063291	3318385
				224664	392		1392	6502
				0.8443	0.873417	0.11917098	0.87341	0.209726
16	256	0.2	0.2	223443	7215189	445595854	7721518	4437689
				223443	873		9873	9696
				0.8592	0.879746	0.13100848	0.87974	0.228055
16	256	0.4	0	796092	8354430	256361922	6835443	7834290
				796092	38		038	4015
				0.8484	0.879746	0.12279151	0.87974	0.215503
16	256	0.4	0.2	432234	8354430	943462897	6835443	8759689
				432234	38		038	9223
				0.8594	0.816455	0.12285714	0.81645	0.213576
16	256	0.4	0.4	322344	6962025	285714286	5696202	1589403
				322345	317		5317	9733
				0.8745	0.841772	0.13926701	0.84177	0.238993
32	16	0	0	421245	1518987	570680627	2151898	7106918
				421245	342		7342	2387
				0.8711	0.841772	0.13613101	0.84177	0.234361
32	16	0.2	0	843711	1518987	330603888	2151898	2334801
				843712	342		7342	7618
				0.8597	0.848101	0.12725546	0.84810	0.221304
32	16	0.2	0.2	374847	2658227	058879392	1265822	7068538
				374848	848		7848	398
				0.8887	0.822784	0.15133876	0.82278	0.255653
32	16	0.4	0	362637	8101265	600698487	4810126	8839724
				362637	823		5823	681
				0.8763	0.835443	0.14012738	0.83544	0.240000
32	16	0.4	0.2	736263	0379746	853503184	3037974	0000000
				736264	836		6836	0002
				0.8980	0.810126	0.16080402	0.81012	0.268343
32	16	0.4	0.4	463980	5822784	010050251	6582278	8155136
				463981	81		481	268
				0.8672	0.860759	0.13518886	0.86075	0.233676
32	32	0	0	161172	4936708	679920478	9493670	9759450
				161172	861		8861	1716
				0.8278	0.898734	0.11181102	0.89873	0.198879
32	32	0.2	0	388278	1772151	362204724	4177215	5518207
				388278	899		1899	2828
				0.8543	0.860759	0.12477064	0.86075	0.217948
32	32	0.2	0.2	956043	4936708	220183487	9493670	7179487
				956044	861		8861	1798

32	32	0.4	0	0.8794 261294 261294	0.848101 2658227 848	0.14502164 502164502	0.84810 1265822 7848	0.247689 4639556 377
32	32	0.4	0.2	0.8652 319902 319903	0.841772 1518987 342	0.13090551 18110236	0.84177 2151898 7342	0.226575 8091993 1855
32	32	0.4	0.4	0.8867 521367 521367	0.803797 4683544 303	0.14614499 424626007	0.80379 7468354 4303	0.247322 2979552 0935
32	64	0	0	0.8305 860805 860806	0.892405 0632911 392	0.11270983 213429256	0.89240 5063291 1392	0.200141 9446415 8977
32	64	0.2	0	0.8405 067155 067155	0.886075 9493670 886	0.11814345 991561181	0.88607 5949367 0886	0.208488 4586746 0908
32	64	0.2	0.2	0.8663 003663 003663	0.841772 1518987 342	0.13181367 690782952	0.84177 2151898 7342	0.227934 8757497 8577
32	64	0.4	0	0.8424 908424 908425	0.892405 0632911 392	0.12020460 358056266	0.89240 5063291 1392	0.211870 7738542 449
32	64	0.4	0.2	0.8618 742368 742369	0.841772 1518987 342	0.12813102 1194605	0.84177 2151898 7342	0.222408 0267558 5286
32	64	0.4	0.4	0.8496 642246 642246	0.848101 2658227 848	0.11974977 658623771	0.84810 1265822 7848	0.209866 8754894 2834
32	128	0	0	0.8400 488400 488401	0.892405 0632911 392	0.11858704 793944491	0.89240 5063291 1392	0.209354 1202672 6058
32	128	0.2	0	0.8496 642246 642246	0.886075 9493670 886	0.12444444 444444444	0.88607 5949367 0886	0.218238 5035074 045
32	128	0.2	0.2	0.8275 335775 335775	0.892405 0632911 392	0.11093627 06530291	0.89240 5063291 1392	0.197340 7977606 718
32	128	0.4	0	0.8452 380952 380952	0.892405 0632911 392	0.12207792 207792208	0.89240 5063291 1392	0.214775 3236862 1477
32	128	0.4	0.2	0.8269 230769 230769	0.892405 0632911 392	0.11058823 529411765	0.89240 5063291 1392	0.196789 9511514 3055
32	128	0.4	0.4	0.8579 059829 059829	0.841772 1518987 342	125	0.84177 2151898 7342	0.217675 9410801 964
32	256	0	0	0.8659 951159 95116	0.867088 6075949 367	0.13497536 945812807	0.86708 8607594 9367	0.233589 0878090 3664
32	256	0.2	0	0.8519 536019 536019	0.879746 8354430 38	0.12533814 247069433	0.87974 6835443 038	0.219415 9431728 4926
32	256	0.2	0.2	0.8510 378510 37851	0.854430 3797468 354	0.12151215 121512152	0.85443 0379746 8354	0.212765 9574468 085
32	256	0.4	0	0.8496 642246 642246	0.886075 9493670 886	0.12444444 444444444	0.88607 5949367 0886	0.218238 5035074 045
32	256	0.4	0.2	0.8661 477411 477412	0.841772 1518987 342	0.13168316 83168317	0.84177 2151898 7342	0.227739 7260273 9725

32	256	0.4	0.4	0.8263 125763 125763	0.892405 0632911 392	0.11024237 685691947	0.89240 5063291 1392	0.196242 1711899 791
64	16	0	0	0.8722 527472 527473	0.835443 0379746 836	0.13622291 021671826	0.83544 3037974 6836	0.234250 2218278 6154
64	16	0.2	0	0.8400 488400 488401	0.867088 6075949 367	0.11561181 434599156	0.86708 8607594 9367	0.204020 8488458 6746
64	16	0.2	0.2	0.8431 013431 013431	0.860759 4936708 861	0.11683848 797250859	0.86075 9493670 8861	0.205748 8653555 2195
64	16	0.4	0	0.8524 114774 114774	0.848101 2658227 848	0.12170753 860127158	0.84810 1265822 7848	0.212867 3550436 8548
64	16	0.4	0.2	0.8476 800976 800977	0.829113 9240506 329	0.11603188 662533215	0.82911 3924050 6329	0.203574 2035742 0355
64	16	0.4	0.4	0.8865 995115 995116	0.810126 5822784 81	0.14695752 009184845	0.81012 6582278 481	0.248785 2283770 6513
64	32	0	0	0.8659 951159 95116	0.854430 3797468 354	0.13326752 22112537	0.85443 0379746 8354	0.230572 1605465 4142
64	32	0.2	0	0.8620 268620 268621	0.867088 6075949 367	0.13160422 670509125	0.86708 8607594 9367	0.228523 7698081 735
64	32	0.2	0.2	0.8556 166056 166056	0.848101 2658227 848	0.12407407 407407407	0.84810 1265822 7848	0.216478 1906300 4844
64	32	0.4	0	0.8609 584859 58486	0.860759 4936708 861	0.12989493 791786055	0.86075 9493670 8861	0.225726 1410788 3818
64	32	0.4	0.2	0.8331 807081 807082	0.873417 7215189 873	0.11210398 050365557	0.87341 7721518 9873	0.198704 1036717 0627
64	32	0.4	0.4	0.8811 050061 050061	0.797468 3544303 798	0.13922651 933701657	0.79746 8354430 3798	0.237064 9106302 9161
64	64	0	0	0.8467 643467 643468	0.886075 9493670 886	0.12237762 237762238	0.88607 5949367 0886	0.215053 7634408 6022
64	64	0.2	0	0.8229 548229 548229	0.898734 1772151 899	0.10906298 003072197	0.89873 4177215 1899	0.194520 5479452 0549
64	64	0.2	0.2	0.8754 578754 578755	0.829113 9240506 329	0.13833157 338965152	0.82911 3924050 6329	0.237104 0723981 9004
64	64	0.4	0	0.8321 123321 123322	0.892405 0632911 392	0.11361804 995970991	0.89240 5063291 1392	0.201572 5518227 3054
64	64	0.4	0.2	0.8536 324786 324786	0.867088 6075949 367	125	0.86708 8607594 9367	0.218500 7974481 6586
64	64	0.4	0.4	0.8650 793650 793651	0.854430 3797468 354	0.13248282 63002944	0.85443 0379746 8354	0.229396 7714528 462
64	128	0	0	0.8634 004884 004884	0.860759 4936708 861	0.13191076 624636275	0.86075 9493670 8861	0.228763 6669470 1428

64	128	0.2	0	0.8635 531135 531136	0.867088 6075949 367	0.13288069 835111543	0.86708 8607594 9367	0.230445 7527333 894
64	128	0.2	0.2	0.8615 689865 689866	0.841772 1518987 342	0.12788461 538461537	0.84177 2151898 7342	0.222036 7278797 9964
64	128	0.4	0	0.8220 390720 39072	0.898734 1772151 899	0.10856269 113149847	0.89873 4177215 1899	0.193724 4201909 9592
64	128	0.4	0.2	0.8402 014652 014652	0.879746 8354430 38	0.11720067 453625632	0.87974 6835443 038	0.206845 2380952 3808
64	128	0.4	0.4	0.8226 495726 495726	0.886075 9493670 886	0.10752688 172043011	0.88607 5949367 0886	0.191780 8219178 082
64	256	0	0	0.8211 233211 233211	0.892405 0632911 392	0.10738766 184310738	0.89240 5063291 1392	0.191706 3222297 7566
64	256	0.2	0	0.8513 431013 431013	0.879746 8354430 38	0.12488769 092542677	0.87974 6835443 038	0.218725 4130605 8219
64	256	0.2	0.2	0.8699 633699 6337	0.841772 1518987 342	0.13502538 07106599	0.84177 2151898 7342	0.232720 9098862 6423
64	256	0.4	0	0.8452 380952 380952	0.886075 9493670 886	0.12131715 771230503	0.88607 5949367 0886	0.213414 6341463 415
64	256	0.4	0.2	0.8627 899877 899878	0.848101 2658227 848	0.12971926 427879962	0.84810 1265822 7848	0.225020 9907640 6382
64	256	0.4	0.4	0.8197 496947 496947	0.892405 0632911 392	0.10665658 093797277	0.89240 5063291 1392	0.190540 5405405 4054
12 8	16	0	0	875	0.803797 4683544 303	0.13424947 145877378	0.80379 7468354 4303	0.230072 4637681 1594
12 8	16	0.2	0	0.8614 163614 163615	0.848101 2658227 848	0.12859884 836852206	0.84810 1265822 7848	0.223333 3333333 333
12 8	16	0.2	0.2	0.8389 804639 80464	0.879746 8354430 38	0.11641541 038525963	0.87974 6835443 038	0.205621 3017751 4792
12 8	16	0.4	0	0.8711 843711 843712	0.816455 6962025 317	0.13257965 056526208	0.81645 5696202 5317	0.228116 7108753 3156
12 8	16	0.4	0.2	0.9087 301587 301587	0.810126 5822784 81	0.17630853 99449036	0.81012 6582278 481	0.289592 7601809 9545
12 8	16	0.4	0.4	0.8782 051282 051282	0.810126 5822784 81	0.13822894 168466524	0.81012 6582278 481	0.236162 3616236 1626
12 8	32	0	0	0.8534 798534 798534	0.879746 8354430 38	0.12647861 69244768	0.87974 6835443 038	0.221161 4956245 028
12 8	32	0.2	0	0.9191 086691 086691	0.791139 2405063 291	0.19083969 465648856	0.79113 9240506 3291	0.307503 0750307 503
12 8	32	0.2	0.2	0.8499 694749 69475	0.860759 4936708 861	0.12153708 668453976	0.86075 9493670 8861	0.212999 2169146 4369

12 8	32	0.4	0	0.8708 791208 791209	0.854430 3797468 354	0.13761467 889908258	0.85443 0379746 8354	0.237050 0438981 5632
12 8	32	0.4	0.2	0.8731 684981 684982	0.816455 6962025 317	0.134375	0.81645 5696202 5317	0.230769 2307692 3075
12 8	32	0.4	0.4	0.8606 532356 532357	0.835443 0379746 836	0.12631578 947368421	0.83544 3037974 6836	0.219451 3715710 7233
12 8	64	0	0	0.8566 849816 849816	0.879746 8354430 38	0.12894248 608534323	0.87974 6835443 038	0.224919 0938511 327
12 8	64	0.2	0	0.8577 533577 533577	0.867088 6075949 367	0.12815715 622076707	0.86708 8607594 9367	0.223308 8834555 8272
12 8	64	0.2	0.2	0.8768 315018 315018	0.816455 6962025 317	0.13782051 282051283	0.81645 5696202 5317	0.235831 8098720 2926
12 8	64	0.4	0	0.8400 488400 488401	0.892405 0632911 392	0.11858704 793944491	0.89240 5063291 1392	0.209354 1202672 6058
12 8	64	0.4	0.2	0.8435 592185 592186	0.860759 4936708 861	0.11714039 621016366	0.86075 9493670 8861	0.206216 8309325 2463
12 8	64	0.4	0.4	0.8537 851037 851037	0.822784 8101265 823	0.11948529 411764706	0.82278 4810126 5823	0.208667 7367576 244
12 8	128	0	0	0.8305 860805 860806	0.892405 0632911 392	0.11270983 213429256	0.89240 5063291 1392	0.200141 9446415 8977
12 8	128	0.2	0	0.8498 168498 168498	0.886075 9493670 886	0.12455516 014234876	0.88607 5949367 0886	0.218408 7363494 5396
12 8	128	0.2	0.2	0.8337 912087 912088	0.886075 9493670 886	0.11391375 101708706	0.88607 5949367 0886	0.201874 5493871 6656
12 8	128	0.4	0	0.8460 012210 01221	0.867088 6075949 367	0.11954624 781849912	0.86708 8607594 9367	0.210122 6993865 0308
12 8	128	0.4	0.2	0.8502 747252 747253	0.848101 2658227 848	0.12017937 219730941	0.84810 1265822 7848	0.210526 3157894 7367
12 8	128	0.4	0.4	0.8324 175824 175825	0.879746 8354430 38	0.11236863 379143087	0.87974 6835443 038	0.199283 1541218 638
12 8	256	0	0	0.8487 484737 484737	0.879746 8354430 38	0.12300884 955752213	0.87974 6835443 038	0.215838 5093167 7018
12 8	256	0.2	0	0.8373 015873 015873	0.892405 0632911 392	0.11681855 840927921	0.89240 5063291 1392	0.206593 4065934 066
12 8	256	0.2	0.2	0.8392 857142 857143	0.879746 8354430 38	0.11661073 825503356	0.87974 6835443 038	0.205925 9259259 2593
12 8	256	0.4	0	0.8707 264957 264957	0.860759 4936708 861	0.13835198 372329605	0.86075 9493670 8861	0.238387 3794916 74
12 8	256	0.4	0.2	0.8455 433455 433455	0.860759 4936708 861	0.11846689 895470383	0.86075 9493670 8861	0.208269 5252679 9386

12				0.8432	0.835443		0.83544	0.200455
8	256	0.4	0.4	539682	0379746	0.11389128	3037974	5808656
				539683	836	559102675	6836	0364
				0.7603	0.898734		0.89873	0.151871
25	16	0	0	785103	1772151	0.08294392	4177215	6577540
6				785103	899	523364486	1899	1068
				0.8826	0.810126		0.81012	0.242654
25	16	0.2	0	312576	5822784	0.14269788	6582278	0284360
6				312576	81	18283166	481	1896
				0.9152	0.791139		0.79113	0.298329
25	16	0.2	0.2	930402	2405063	0.18382352	9240506	3556085
6				930403	291	941176472	3291	919
				0.8641	0.854430		0.85443	0.228233
25	16	0.4	0	636141	3797468	0.13170731	0379746	3051563
6				636142	354	707317074	8354	821
				0.7970	0.898734		0.89873	0.174233
25	16	0.4	0.2	085470	1772151	0.09646739	4177215	1288343
6				085471	899	130434782	1899	5582
				0.8856	0.791139		0.79113	0.242248
25	16	0.4	0.4	837606	2405063	0.14302059	9240506	0620155
6				837606	291	496567507	3291	0386
				0.8521	0.860759		0.86075	0.215360
25	32	0	0	062271	4936708	0.12307692	9493670	2533650
6				062271	861	307692308	8861	0397
				0.8147	0.898734		0.89873	0.187582
25	32	0.2	0	130647	1772151	0.10471976	4177215	5627476
6				130647	899	401179942	1899	8825
				0.8261	0.886075		0.88607	0.194850
25	32	0.2	0.2	599511	9493670	0.10946051	5949367	3827418
6				599511	886	602814699	0886	2323
				0.8859	0.816455		0.81645	0.249516
25	32	0.4	0	890109	6962025	0.14726027	5696202	4410058
6				89011	317	397260275	5317	027
				0.7934	0.898734		0.89873	0.171808
25	32	0.4	0.2	981684	1772151	0.09498327	4177215	8324258
6				981685	899	759197324	1899	9232
				0.8510	0.816455		0.81645	0.204275
25	32	0.4	0.4	378510	6962025	0.11674208	5696202	5344418
6				37851	317	14479638	5317	052
				0.8460	0.886075		0.88607	0.214231
25	64	0	0	012210	9493670	0.12184508	5949367	0635042
6				01221	886	268059181	0886	081
				0.8421	0.879746		0.87974	0.208865
25	64	0.2	0	855921	8354430	0.11849957	6835443	5146506
6				855922	38	37425405	038	3863
				0.8273	0.892405		0.89240	0.197202
25	64	0.2	0.2	809523	0632911	0.11084905	5063291	7972027
6				809523	392	660377359	1392	972
				0.8571	0.848101		0.84810	0.218241
25	64	0.4	0	428571	2658227	0.12523364	1265822	0423452
6				428571	848	48598131	7848	769
				0.8402	0.860759		0.86075	0.202833
25	64	0.4	0.2	014652	4936708	0.11496196	9493670	7061894
6				014652	861	111580727	8861	109
				0.8768	0.797468		0.79746	0.230980
25	64	0.4	0.4	315018	3544303	0.13504823	8354430	7516040
6				315018	798	151125403	3798	3303
				0.8440	0.879746		0.87974	0.210765
25	128	0	0	170940	8354430	0.11972437	6835443	7316148
6				17094	38	553832903	038	5976

25 6	128	0.2	0	0.8122 710622 710623	0.905063 2911392 406	0.10415149 308084487	0.90506 3291139 2406	0.186806 0091443 5014
25 6	128	0.2	0.2	0.8177 655677 655677	0.892405 0632911 392	0.10561797 752808989	0.89240 5063291 1392	0.188881 4467515 0706
25 6	128	0.4	0	0.8655 372405 372406	0.854430 3797468 354	0.13287401 57480315	0.85443 0379746 8354	0.229982 9642248 7227
25 6	128	0.4	0.2	0.8855 311355 311355	0.816455 6962025 317	0.14675767 918088736	0.81645 5696202 5317	0.248794 5998071 3597
25 6	128	0.4	0.4	0.8356 227106 227107	0.879746 8354430 38	0.11430921 052631579	0.87974 6835443 038	0.202328 9665211 0624
25 6	256	0	0	0.8569 902319 902319	0.841772 1518987 342	0.12429906 542056075	0.84177 2151898 7342	0.216612 3778501 6285
25 6	256	0.2	0	0.8463 064713 064713	0.886075 9493670 886	0.12205754 141238012	0.88607 5949367 0886	0.214559 3869731 801
25 6	256	0.2	0.2	0.8560 744810 74481	0.841772 1518987 342	0.12360594 795539033	0.84177 2151898 7342	0.215559 1572123 1766
25 6	256	0.4	0	0.8624 847374 847375	0.873417 7215189 873	0.13282001 924927817	0.87341 7721518 9873	0.230576 4411027 5693
25 6	256	0.4	0.2	0.8440 170940 17094	0.879746 8354430 38	0.11972437 553832903	0.87974 6835443 038	0.210765 7316148 5976
25 6	256	0.4	0.4	0.8576 007326 007326	0.816455 6962025 317	0.12146892 655367232	0.81645 5696202 5317	0.211475 4098360 6558

Όπως φαίνεται από τα αποτελέσματα υπήρξε μείωση της ακρίβειας και για άσσους και για μηδενικά. Η πτώση αυτή οφείλεται στην συμπίεση των δεδομένων εισόδου.

3.2.2.4 Συγκριση αποτελεσμάτων

Παρακάτω φαίνεται ο πίνακας συγκρίσεων των αποτελεσμάτων για τα συμπίεσιμα και μη δεδομένα.

	Μη συμπίεσιμα	Συμπίεσιμα
ΜΟ ακρίβειας άσων	0,8778902954	0,8582278481

MO ακρίβειας μηδενικών	0,8830453805	0,8542002442
MO Precision	0,1593902649	0,1262220601
MO Recall	0,8778902954	0,8582278481
MO Fscore	0,268169344	0,2195689

Όπως ήταν και αναμενόμενο τα αποτελέσματα είναι καλύτερα για τα μη συμπιεσμένα δεδομένα. Η διαφορά όμως αυτή είναι αρκετά μικρή στην συγκεκριμένη περίπτωση. Έτσι σε συστήματα με μικρή υπολογιστική ισχύ η συμπίεση των δεδομένων ίσως επιτάχυνε την εκπαίδευση τους μοντέλου χωρίς μεγάλες απώλειες όσον αφορά τα αποτελέσματα.

3.2.3 Εφαρμογή CNN δικτύου

3.2.3.1 Γενικά

Έχοντας εφαρμόσει ένα δίκτυο RNN στην συνέχεια θα δημιουργήσουμε ένα δίκτυο CNN με σκοπό να συγκρίνουμε ποιο από τα δυο αυτά μοντέλα λειτουργεί καλύτερα στα συγκεκριμένα datasets. Στο δίκτυο αυτό θα εφαρμόσουμε τα ίδια βάρη που εφαρμόσαμε και προηγουμένως. Ο κώδικας φαίνεται παρακάτω

```

model = Sequential()
model.add(layers.Conv1D(num_layers*4, 6, padding='same', activation='relu',
input_shape=(num_inputs, 1)))
model.add(layers.Conv1D(num_layers*2, 3, padding='same', activation='relu'))
model.add(layers.Conv1D(num_layers, 2, padding='same', activation='relu'))
model.add(layers.GlobalMaxPooling1D())
model.add(layers.Dense(1))
model.compile(optimizer='RMSprop', loss='binary_crossentropy', metrics=['acc'])
model.summary()
history = model.fit(samples_expanded, traintargets, shuffle=False, batch_size=32,
epochs=50, validation_split=0.2, class_weight={0:0.02, 1:1.0})

```

Ας εξηγήσουμε τις παραμέτρους που δίνουμε σε κάθε ένα από τα Conv1D επίπεδα.

`Conv1D(filters, kernel, padding, activation, input_shape)`

- **filters:** Αφορά των αριθμό κόμβων εξόδου του συγκεκριμένου επιπέδου.
- **kernel:** Μέγεθος πίνακα βαρών.
- **activation:** Συνάρτηση ενεργοποίησης των εξόδων των κόμβων του συστήματος.
- **padding:** Αφορά την συμπεριφορά του συστήματος κατά την την συνέλιξη των δεδομένων με τις μήτρες των βαρών. Όστε ο αριθμός των δεδομένων εξόδου να μείνει ίδιος θα πρέπει να εφαρμόσουμε `padding="same"` το οποίο σημαίνει ότι κατά την συνέλιξη ο πίνακας εισόδου θα περιέχει μηδενικά πριν τα δεδομένα έτσι ώστε ο αριθμός εξόδων να είναι ίδιος με τον αριθμός εισόδων.
- **input_shape:** Το σχήμα των δεδομένων εισόδου. Στην περίπτωση μας θα είναι πίνακας (8,1) για τα μη συμπιεσμένα δεδομένα και (3,1) για τα συμπιεσμένα δεδομένα.

Επίσης η μεταβλητή `num_layers` θα πάρει τιμες απο 8 έως 64 δημιουργώντας έτσι τα εξής μοντέλα και τα αποτελέσματα τους:

3.2.3.2 Αποτελέσματα

1st	2nd	3rd	Zeros acc	Ones acc	Precision	Recall	fscore
32	16	8	0.9829059 82905982 9	0.74050632 9113924	0.510917030 3445777	0.74050 632864 52492	0.60465111 41611454
64	32	16	1.0	0.0	0.0	0.0	0.0
128	64	32	1.0	0.0	0.0	0.0	0.0
256	128	64	1.0	0.0	0.0	0.0	0.0

Όπως φαίνεται από τα αποτελέσματα των δικτύων όσο μεγαλώνουμε τα filters του συστήματος τόσο αυτό δυσκολεύεται να προβλέψει του άσους παρά το γεγονός ότι έχουμε προσθέσει βάρη. Αυτό πιθανότατα συμβαίνει λόγω του μικρού όγκου θετικών αποτελεσμάτων του συνόλου δεδομένων μιας και πρόκειται για feedforward δίκτυο.

3.3 Pima dataset

3.3.1 Γενικά

Το Pima είναι ένα σύνολο δεδομένων που αποτελείται από κάποιες τιμές αιματολογικών εξετάσεων καθώς και από το αποτέλεσμα που αφορά το εάν ο ασθενής έχει διαβήτη. Όπως και στην περίπτωση του mammography dataset πρόκειται για ένα πρόβλημα κατάταξης με δύο πιθανά αποτελέσματα. Το dataset αποτελείται από 8 τιμές εισόδου και 1 τιμή εξόδου ενώ η συμπιεσμένη έκδοση αποτελείται από 3 τιμές εισόδου και 1 τιμή εξόδου αντίστοιχα. Όπως και προηγουμένως θα εκπαιδευτούν νευρωνικά δίκτυα και για τα δυο dataset και τα αποτελέσματα θα συγκριθούν.

3.3.2 Εφαρμογή RNN δίκτυο

3.3.2.1 Αποτελέσματα μη συμπιεσμένων δεδομένων

Η διαδικασία που ακολουθήθηκε είναι η ίδια και για το συγκεκριμένο dataset. Ας δούμε τις αρχικές μετρήσεις.

Units	Batch	Rec Drop	Dropout	Zeros acc	Ones acc
16	8	0	0	0.8520900321543409	0.6133333333333333
16	8	0.2	0	0.8713826366559485	0.5933333333333334
16	8	0.2	0.2	0.8553054662379421	0.64
16	8	0.4	0	0.8231511254019293	0.66
16	8	0.4	0.2	0.8681672025723473	0.62
16	8	0.4	0.4	0.8263665594855305	0.64
16	16	0	0	0.8617363344051447	0.6133333333333333
16	16	0.2	0	0.8295819935691319	0.6466666666666666
16	16	0.2	0.2	0.842443729903537	0.6533333333333333
16	16	0.4	0	0.819935691318328	0.6533333333333333

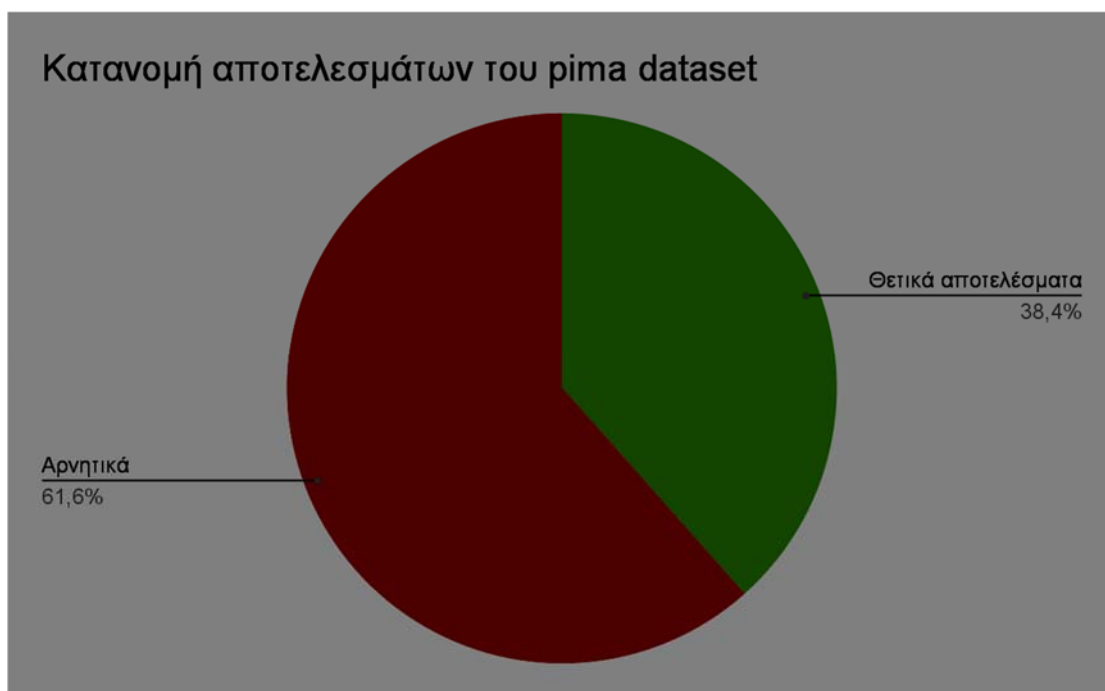
16	16	0.4	0.2	0.8520900321543409	0.6533333333333333
16	16	0.4	0.4	0.8263665594855305	0.6866666666666666
16	32	0	0	0.8488745980707395	0.6266666666666667
16	32	0.2	0	0.8681672025723473	0.6133333333333333
16	32	0.2	0.2	0.8392282958199357	0.64
16	32	0.4	0	0.8488745980707395	0.6266666666666667
16	32	0.4	0.2	0.8360128617363344	0.66
16	32	0.4	0.4	0.8231511254019293	0.6666666666666666
16	64	0	0	0.8360128617363344	0.66
16	64	0.2	0	0.8617363344051447	0.64
16	64	0.2	0.2	0.842443729903537	0.6266666666666667
16	64	0.4	0	0.8488745980707395	0.64
16	64	0.4	0.2	0.8231511254019293	0.6666666666666666
16	64	0.4	0.4	0.8360128617363344	0.6533333333333333
16	128	0	0	0.9196141479099679	0.32
16	128	0.2	0	0.9421221864951769	0.3
16	128	0.2	0.2	0.9517684887459807	0.2933333333333333 3
16	128	0.4	0	0.9292604501607717	0.4066666666666667
16	128	0.4	0.2	0.8778135048231511	0.48
16	128	0.4	0.4	0.9614147909967846	0.1933333333333333 3
32	8	0	0	0.8456591639871383	0.64
32	8	0.2	0	0.8488745980707395	0.6333333333333333
32	8	0.2	0.2	0.8681672025723473	0.6
32	8	0.4	0	0.864951768488746	0.5866666666666667
32	8	0.4	0.2	0.8488745980707395	0.6466666666666666
32	8	0.4	0.4	0.8585209003215434	0.6533333333333333
32	16	0	0	0.8553054662379421	0.6
32	16	0.2	0	0.8585209003215434	0.5933333333333334
32	16	0.2	0.2	0.8295819935691319	0.6666666666666666
32	16	0.4	0	0.8488745980707395	0.6333333333333333
32	16	0.4	0.2	0.8488745980707395	0.6533333333333333
32	16	0.4	0.4	0.842443729903537	0.64
32	32	0	0	0.842443729903537	0.6533333333333333
32	32	0.2	0	0.8360128617363344	0.6733333333333333
32	32	0.2	0.2	0.8488745980707395	0.6266666666666667
32	32	0.4	0	0.8617363344051447	0.6333333333333333
32	32	0.4	0.2	0.8488745980707395	0.6733333333333333
32	32	0.4	0.4	0.8263665594855305	0.6933333333333334
32	64	0	0	0.8553054662379421	0.6133333333333333
32	64	0.2	0	0.8585209003215434	0.6466666666666666
32	64	0.2	0.2	0.8456591639871383	0.6133333333333333
32	64	0.4	0	0.8553054662379421	0.6333333333333333
32	64	0.4	0.2	0.8520900321543409	0.6533333333333333

32	64	0.4	0.4	0.8520900321543409	0.62
32	128	0	0	0.9131832797427653	0.5333333333333333
32	128	0.2	0	0.8842443729903537	0.6066666666666667
32	128	0.2	0.2	0.909967845659164	0.5
32	128	0.4	0	0.8971061093247589	0.5
32	128	0.4	0.2	0.8842443729903537	0.5133333333333333
32	128	0.4	0.4	0.8938906752411575	0.36
64	8	0	0	0.864951768488746	0.5933333333333334
64	8	0.2	0	0.8585209003215434	0.6133333333333333
64	8	0.2	0.2	0.8392282958199357	0.6466666666666666
64	8	0.4	0	0.8520900321543409	0.6
64	8	0.4	0.2	0.8263665594855305	0.6466666666666666
64	8	0.4	0.4	0.8327974276527331	0.6866666666666666
64	16	0	0	0.8617363344051447	0.64
64	16	0.2	0	0.8617363344051447	0.6333333333333333
64	16	0.2	0.2	0.8553054662379421	0.6533333333333333
64	16	0.4	0	0.8488745980707395	0.6266666666666667
64	16	0.4	0.2	0.819935691318328	0.68
64	16	0.4	0.4	0.8488745980707395	0.62
64	32	0	0	0.8520900321543409	0.6266666666666667
64	32	0.2	0	0.8488745980707395	0.6266666666666667
64	32	0.2	0.2	0.8488745980707395	0.64
64	32	0.4	0	0.8617363344051447	0.6133333333333333
64	32	0.4	0.2	0.8392282958199357	0.6333333333333333
64	32	0.4	0.4	0.8456591639871383	0.66
64	64	0	0	0.8520900321543409	0.6266666666666667
64	64	0.2	0	0.8681672025723473	0.6333333333333333
64	64	0.2	0.2	0.8488745980707395	0.66
64	64	0.4	0	0.8681672025723473	0.62
64	64	0.4	0.2	0.8713826366559485	0.6133333333333333
64	64	0.4	0.4	0.8778135048231511	0.6
64	128	0	0	0.8617363344051447	0.64
64	128	0.2	0	0.8681672025723473	0.6533333333333333
64	128	0.2	0.2	0.8778135048231511	0.6333333333333333
64	128	0.4	0	0.864951768488746	0.64
64	128	0.4	0.2	0.8713826366559485	0.62
64	128	0.4	0.4	0.8617363344051447	0.62
128	8	0	0	0.8360128617363344	0.6266666666666667
128	8	0.2	0	0.8681672025723473	0.5933333333333334
128	8	0.2	0.2	0.8745980707395499	0.6133333333333333
128	8	0.4	0	0.8713826366559485	0.6133333333333333
128	8	0.4	0.2	0.8681672025723473	0.6133333333333333
128	8	0.4	0.4	0.8585209003215434	0.6133333333333333
128	16	0	0	0.864951768488746	0.6

128	16	0.2	0	0.864951768488746	0.62
128	16	0.2	0.2	0.8520900321543409	0.6533333333333333
128	16	0.4	0	0.8745980707395499	0.5666666666666667
128	16	0.4	0.2	0.8617363344051447	0.6066666666666667
128	16	0.4	0.4	0.8520900321543409	0.6466666666666666
128	32	0	0	0.8617363344051447	0.62
128	32	0.2	0	0.8681672025723473	0.6133333333333333
128	32	0.2	0.2	0.819935691318328	0.6933333333333334
128	32	0.4	0	0.8713826366559485	0.6133333333333333
128	32	0.4	0.2	0.842443729903537	0.64
128	32	0.4	0.4	0.8360128617363344	0.6066666666666667
128	64	0	0	0.8745980707395499	0.6266666666666667
128	64	0.2	0	0.8488745980707395	0.6133333333333333
128	64	0.2	0.2	0.819935691318328	0.66
128	64	0.4	0	0.8456591639871383	0.64
128	64	0.4	0.2	0.8553054662379421	0.6266666666666667
128	64	0.4	0.4	0.864951768488746	0.6133333333333333
128	128	0	0	0.8681672025723473	0.64
128	128	0.2	0	0.8778135048231511	0.6466666666666666
128	128	0.2	0.2	0.8585209003215434	0.6533333333333333
128	128	0.4	0	0.8585209003215434	0.66
128	128	0.4	0.2	0.8456591639871383	0.6266666666666667
128	128	0.4	0.4	0.8778135048231511	0.6
256	8	0	0	0.7877813504823151	0.7
256	8	0.2	0	0.9131832797427653	0.5466666666666666
256	8	0.2	0.2	0.8745980707395499	0.6
256	8	0.4	0	0.8938906752411575	0.5533333333333333
256	8	0.4	0.2	0.842443729903537	0.6733333333333333
256	8	0.4	0.4	0.8938906752411575	0.6
256	16	0	0	0.8456591639871383	0.62
256	16	0.2	0	0.8681672025723473	0.5933333333333334
256	16	0.2	0.2	0.864951768488746	0.6066666666666667
256	16	0.4	0	0.8392282958199357	0.66
256	16	0.4	0.2	0.9260450160771704	0.4666666666666667
256	16	0.4	0.4	0.8681672025723473	0.62
256	32	0	0	0.8617363344051447	0.64
256	32	0.2	0	0.8713826366559485	0.6066666666666667
256	32	0.2	0.2	0.8456591639871383	0.62
256	32	0.4	0	0.8617363344051447	0.6266666666666667
256	32	0.4	0.2	0.8745980707395499	0.58
256	32	0.4	0.4	0.8392282958199357	0.6533333333333333
256	64	0	0	0.8585209003215434	0.64
256	64	0.2	0	0.8585209003215434	0.64
256	64	0.2	0.2	0.8585209003215434	0.6333333333333333

256	64	0.4	0	0.8585209003215434	0.6266666666666667
256	64	0.4	0.2	0.864951768488746	0.6266666666666667
256	64	0.4	0.4	0.8778135048231511	0.6133333333333333
256	128	0	0	0.8713826366559485	0.6466666666666666
256	128	0.2	0	0.8617363344051447	0.6333333333333333
256	128	0.2	0.2	0.864951768488746	0.6266666666666667
256	128	0.4	0	0.8553054662379421	0.6533333333333333
256	128	0.4	0.2	0.864951768488746	0.62
256	128	0.4	0.4	0.887459807073955	0.5733333333333334

Όπως φαίνεται και πάλι το μοντέλο έχει μάθει να αναγνωρίζει καλύτερα τα μηδενικά απο τους άσσους. Ελέγχοντας το σύνολο των δεδομένων παρατηρήθηκε ότι το training pima dataset αποτελείται απο περίπου 38,5% θετικά αποτελέσματα και 61,5 αρνητικά αποτελέσματα.



Η κατανομή είναι σαφώς πολύ καλύτερη από ότι στην περίπτωση του Mamnography αλλα και πάλι θα εφαρμόσουμε βάρη. Αποτελέσματα με βάρη και fscore φαίνονται στον από κάτω πίνακα

Unit	Batc	Rec	Dropou	Zeros	Ones	Precisio	Recall	Fscore
------	------	-----	--------	-------	------	----------	--------	--------

s	h	drop	t	acc	acc	n		
				0.77491	0.75333	0.61748	0.75333	0.67867
16	8	0	0	9614147	333333	6338460	3332831	8628762
				91	33333	39	1111	0992
				0.78135	0.77333	0.63043	0.77333	0.69461
16	8	0.2	0	0482315	333333	4782266	3332817	0728545
				1125	33333	0681	7778	3081
				0.72668	0.81333	0.58937	0.81333	0.68347
16	8	0.2	0.2	8102893	333333	1980391	3332791	3340247
				8906	33334	6078	1112	4751
				0.77491		0.60674		0.65853
16	8	0.4	0	9614147	0.72	1572692	0.71999	6535328
				91		8418	999952	6772
				0.70739		0.57476	0.81999	0.67582
16	8	0.4	0.2	5498392	0.82	6354871	9999453	4126998
				283		6045	3334	5543
				0.68810	0.83333	0.56306	0.83333	0.67204
16	8	0.4	0.4	2893890	333333	3062809	3332777	2962264
				6752	33334	4311	7778	427
				0.74919		0.59999		0.67826
16	16	0	0	6141479	0.78	9999692	0.77999	0820022
				0996		3077	999948	6877
				0.75884		0.60937		0.68421
16	16	0.2	0	2443729	0.78	4999682	0.77999	0476669
				9035		6172	999948	7481
				0.71704		0.58878		0.69230
16	16	0.2	0.2	1800643	0.84	5046453	0.83999	7643473
				0869		8388	999944	0138
				0.77170	0.74666	0.61202	0.74666	0.67267
16	16	0.4	0	4180064	666666	1857589	6666168	2622759
				3086	66667	0591	8889	7003
				0.74598		0.59067	0.75999	0.66472
16	16	0.4	0.2	0707395	0.76	3574823	9999493	2982468
				4984		4853	3334	1931
				0.70739	0.81333	0.57276	0.81333	0.67217
16	16	0.4	0.4	5498392	333333	9952782	3332791	6259675
				283	33334	7371	1112	648
				0.76848	0.74666	0.60869	0.74666	0.67065
16	32	0	0	8745980	666666	5651843	6666168	8632751
				7074	66667	1003	8889	2676
				0.75562	0.73333	0.59139	0.73333	0.65476
16	32	0.2	0	7009646	333333	7849144	3332844	1854946
				3023	33333	4097	4445	1489
				0.72990	0.79333	0.58620	0.79333	0.67422
16	32	0.2	0.2	3536977	333333	6896262	3332804	0913917
				492	33333	9523	4445	9389
				0.77491		0.61956	0.75999	0.68263
16	32	0.4	0	9614147	0.76	5217054	9999493	4680648
				91		5841	3334	2879
				0.72990	0.78666	0.58415	0.78666	0.67045
16	32	0.4	0.2	3536977	666666	8415552	6666142	4496164
				492	66666	3969	2222	7764
				0.68488	0.85333	0.56637	0.85333	0.68085
16	32	0.4	0.4	7459807	333333	1681165	3332764	1015510
				074	33334	3223	4445	4154
				0.71704	0.72666	0.55329	0.72666	0.62824
16	64	0	0	1800643	666666	9492104	6666182	2025483
				0869	66667	9241	2222	1487
16	64	0.2	0	0.72347	0.77333	0.57425	0.77333	0.65909

				2668810	333333	7425458	3332817	0859807
				2894	33333	2884	7778	5965
				0.72347		0.57635		0.66288
16	64	0.2	0.2	2668810	0.78	4679519	0.77999	9469165
				2894		0371	999948	1521
				0.74919	0.78666	0.60204	0.78666	0.68208
16	64	0.4	0	6141479	666666	0816019	6666142	0875344
				0996	66666	367	2222	9866
				0.72990	0.75333	0.57360	0.75333	0.65129
16	64	0.4	0.2	3536977	333333	4060622	3332831	6780513
				492	33333	536	1111	0884
				0.68488	0.80666	0.55251	0.80666	0.65582
16	64	0.4	0.4	7459807	666666	1415272	6666128	6509658
				074	66666	8259	8889	4227
				0.92926	0.33333	0.69444	0.33333	0.45045
16	128	0	0	0450160	333333	4443479	3333111	0406217
				7717	33333	9383	11113	0321
				0.90996		0.61643	0.29999	0.40358
16	128	0.2	0	7845659	0.3	8355319	9999800	7399545
				164		9475	00003	5417
				0.93247	0.21333	0.60377	0.21333	0.31527
16	128	0.2	0.2	5884244	333333	3583766	3333191	0897066
				373	333335	465	1111	17956
				0.90353		0.65517	0.37999	0.48101
16	128	0.4	0	6977491	0.38	2413040	9999746	2611355
				9614		0317	6667	0224
				0.93247	0.30666	0.68656	0.30666	0.42396
16	128	0.4	0.2	5884244	666666	7163154	6666462	3090564
				373	666664	3774	22224	6797
				0.90353	0.41333	0.67391	0.41333	0.51239
16	128	0.4	0.4	6977491	333333	3042745	3333057	6646663
				9614	333333	7467	7778	4836
				0.76205	0.75333	0.60427	0.75333	0.67062
32	8	0	0	7877813	333333	8074543	3332831	3095605
				5049	33333	1668	1111	3185
				0.75562	0.75333	0.59788	0.75333	0.66666
32	8	0.2	0	7009646	333333	3597567	3332831	6616935
				3023	33333	2574	1111	1156
				0.74276	0.79333	0.59798	0.79333	0.68194
32	8	0.2	0.2	5273311	333333	9949448	3332804	8374663
				8971	33333	2462	4445	594
				0.76848	0.72666	0.60220	0.72666	0.65861
32	8	0.4	0	8745980	666666	9944418	6666182	0221943
				7074	66667	6686	2222	9436
				0.75562		0.61224	0.79999	0.69364
32	8	0.4	0.2	7009646	0.8	4897646	9999466	1568979
				3023		8139	6667	9226
				0.68488	0.85333	0.56637	0.85333	0.68085
32	8	0.4	0.4	7459807	333333	1681165	3332764	1015510
				074	33334	3223	4445	4154
				0.80385	0.73333	0.64327	0.73333	0.68535
32	16	0	0	8520900	333333	4853424	3332844	8205238
				3215	33333	9855	4445	695
				0.73633	0.78666	0.58999	0.78666	0.67428
32	16	0.2	0	4405144	666666	9999705	6666142	5664920
				6945	66666	0001	2222	8199
				0.74919		0.59999		0.67826
32	16	0.2	0.2	6141479	0.78	9999692	0.77999	0820022
				0996		3077	999948	6877

32	16	0.4	0	0.79099 6784565 9164	0.74	0.63068 1817823 4762	0.73999 9999506 6667	0.68098 1544992 2879
32	16	0.4	0.2	0.68810 2893890 6752	0.81333 333333 33334	0.55707 7625316 4029	0.81333 3332791 1112	0.66124 6563856 0269
32	16	0.4	0.4	0.68167 2025723 4726	0.84	0.55999 9999751 1111	0.83999 999944	0.67199 9951641 6034
32	32	0	0	0.76848 8745980 7074	0.76666 666666 66667	0.61497 3261703 2229	0.76666 6666155 5556	0.68249 2531800 0546
32	32	0.2	0	0.74919 6141479 0996	0.78	0.59999 9999692 3077	0.77999 999948	0.67826 0820022 6877
32	32	0.2	0.2	0.73633 4405144 6945	0.79333 333333 33333	0.59203 9800700 4777	0.79333 3332804 4445	0.67806 2628731 9131
32	32	0.4	0	0.76527 3311897 1061	0.73333 333333 33333	0.60109 2895846 3974	0.73333 3332844 4445	0.66066 0610754 9027
32	32	0.4	0.2	0.72990 3536977 492	0.78	0.58208 9551949 2092	0.77999 999948	0.66666 6617342 3953
32	32	0.4	0.4	0.67524 1157556 2701	0.80666 666666 66666	0.54504 5044799 5293	0.80666 6666128 8889	0.65053 7585931 9031
32	64	0	0	0.75884 2443729 9035	0.75333 333333 33333	0.60106 3829467 5192	0.75333 3332831 1111	0.66863 9003490 7778
32	64	0.2	0	0.74598 0707395 4984	0.75333 333333 33333	0.58854 1666360 1346	0.75333 3332831 1111	0.66081 8663817 9306
32	64	0.2	0.2	0.71061 0932475 8842	0.78	0.56521 7391031 296	0.77999 999948	0.65546 2135781 3749
32	64	0.4	0	0.76527 3311897 1061	0.74	0.60326 0869237 3582	0.73999 9999506 6667	0.66467 0608802 7574
32	64	0.4	0.2	0.71382 6366559 4855	0.78666 666666 66666	0.57004 8308903 3583	0.78666 6666142 2222	0.66106 4376674 5951
32	64	0.4	0.4	0.71382 6366559 4855	0.78	0.56796 1164772 8344	0.77999 999948	0.65730 3321654 4665
32	128	0	0	0.88745 9807073 955	0.5	0.68181 8181198 3471	0.49999 9999666 6667	0.57692 3027662 726
32	128	0.2	0	0.84244 3729903 537	0.6	0.64748 2013922 6748	0.59999 9999600 0001	0.62283 7319883 6261
32	128	0.2	0.2	0.92926 0450160 7717	0.38	0.72151 8986428 457	0.37999 9999746 6667	0.49781 6548258 04644
32	128	0.4	0	0.88745 9807073 955	0.60666 666666 66667	0.72222 2221649 03	0.60666 6666262 2222	0.65942 0239755 3073
32	128	0.4	0.2	0.87138 2636655 9485	0.43333 333333 333335	0.61904 7618458 0499	0.43333 3333044 44444	0.50980 3872725 8792

32	128	0.4	0.4	0.90353 6977491 9614 0.73633	0.49333 333333 333335	0.71153 8460854 29	0.49333 3333004 4445	0.58267 7116535 437
64	8	0	0	4405144 6945 0.76848	666666 66666	9999705 0001	6666142 2222	5664920 8199
64	8	0.2	0	8745980 7074 0.75562	0.74	7376717 7282	9999506 6667	6616757 3015
64	8	0.2	0.2	7009646 3023 0.72025	0.76	9999684 2106	9999493 3334	8185591 6992
64	8	0.4	0	7234726 6881 0.70418	666666 66666	9755816 7758	6666142 2222	8683219 9996
64	8	0.4	0.2	0064308 6816 0.73311	0.8	7735582 0577	9999466 6667	3376514 7619
64	8	0.4	0.4	8971061 0932 0.73954	0.8	3004634 9099	9999466 6667	6636294 3322
64	16	0	0	9839228 2959 0.77170	0.79333 333333 33333	0.59499 9999702 5001	0.79333 3332804 4445	0.67999 9950631 8404
64	16	0.2	0	4180064 3086 0.73311	0.74	0109555 0054	9999506 6667	4648856 877
64	16	0.2	0.2	8971061 0932 0.74276	0.78	9999707 5	0.77999 999948	0.66857 1379209 7996
64	16	0.4	0	5273311 8971 0.73954	0.75333 333333 33333	0.58549 2227675 9108	0.75333 3332831 1111	0.65889 2078681 5055
64	16	0.4	0.2	9839228 2959 0.73633	0.8	4925076 112	9999466 6667	0.68376 0634426 672
64	16	0.4	0.4	4405144 6945 0.78778	0.77333 333333 33333	0.58585 8585562 6977	0.77333 3332817 7778	0.66666 6617234 777
64	32	0	0	1350482 3151 0.74598	0.73333 333333 33333	0.62499 9999644 8864	0.73333 3332844 4445	0.67484 6575670 8983
64	32	0.2	0	0707395 4984 0.69774	0.77333 333333 33333	0.59487 1794566 7324	0.77333 3332817 7778	0.67246 3718576 7732
64	32	0.2	0.2	9196141 4791 0.73311	666666 66666	0.55660 3773322 3568	0.78666 6666142 2222	0.65193 3652763 9609
64	32	0.4	0	8971061 0932 0.73633	0.76	0202751 939	9999493 3334	0.65706 0469270 5732
64	32	0.4	0.2	4405144 6945 0.66559	666666 66666	0.58999 9999705 0001	0.78666 6666142 2222	0.67428 5664920 8199
64	32	0.4	0.4	4855305 4662 0.74598	0.83333 333333 33334	0.54585 1528145 9164	0.83333 3332777 7778	0.65963 0558684 5017
64	64	0	0	0707395 4984	0.76666 666666	0.59278 3504849	0.76666 6666155	0.66860 4601592 0805

64	64	0.2	0	0.76848 8745980 7074	0.74666 666666 66667	0.60869 5651843 1003	0.74666 6666168 8889	0.67065 8632751 2676
64	64	0.2	0.2	0.73954 9839228 2959	0.74666 666666 66667	0.58031 0880528 3363	0.74666 6666168 8889	0.65306 1174894 818
64	64	0.4	0	0.76205 7877813 5049	0.78	0.61256 5444705 4631	0.77999 999948	0.68621 6959118 0022
64	64	0.4	0.2	0.75884 2443729 9035	0.78	0.60937 4999682 6172	0.77999 999948	0.68421 0476669 7481
64	64	0.4	0.4	0.71382 6366559 4855	0.78	0.56796 1164772 8344	0.77999 999948	0.65730 3321654 4665
64	128	0	0	0.80707 3954983 9229	0.68	0.62962 9629240 9694	0.67999 9999546 6667	0.65384 6103500 99
64	128	0.2	0	0.81350 4823151 1254	0.67333 333333 33333	0.63522 0125386 6541	0.67333 3332884 4445	0.65372 1632467 1963
64	128	0.2	0.2	0.84244 3729903 537	0.63333 333333 33333	0.65972 2221764 0818	0.63333 3332911 1112	0.64625 8452982 5574
64	128	0.4	0	0.81350 4823151 1254	0.69333 333333 33334	0.64197 5308245 6943	0.69333 3332871 1111	0.66666 6616313 2845
64	128	0.4	0.2	0.83601 2861736 3344	0.62666 666666 66667	0.64827 5861621 8788	0.62666 6666248 8889	0.63728 8085175 5282
64	128	0.4	0.4	0.80707 3954983 9229	0.66666 666666 66666	0.62499 9999609 375	0.66666 6666222 2223	0.64516 1239958 3806
128	8	0	0	0.78778 1350482 3151	0.75333 333333 33333	0.63128 4915848 4442	0.75333 3332831 1111	0.68693 0041156 3122
128	8	0.2	0	0.74276 5273311 8971	0.76666 666666 66667	0.58974 3589441 1571	0.76666 6666155 5556	0.66666 6617130 8585
128	8	0.2	0.2	0.72347 2668810 2894	0.78	0.57635 4679519 0371	0.77999 999948	0.66288 9469165 1521
128	8	0.4	0	0.71704 1800643 0869	0.8	0.57692 3076645 7101	0.79999 9999466 6667	0.67039 1012390 3783
128	8	0.4	0.2	0.74598 0707395 4984	0.78	0.59693 8775205 6435	0.77999 999948	0.67630 0528527 5187
128	8	0.4	0.4	0.73954 9839228 2959	0.75333 333333 33333	0.58247 4226503 8793	0.75333 3332831 1111	0.65697 6694622 0967
128	16	0	0	0.74598 0707395 4984	0.78	0.59693 8775205 6435	0.77999 999948	0.67630 0528527 5187
128	16	0.2	0	0.75241 1575562 701	0.76666 666666 66667	0.59895 8333021 3759	0.76666 6666155 5556	0.67251 4570243 8393
128	16	0.2	0.2	0.73633 4405144 6945	0.76666 666666 66667	0.58375 6344881 342	0.76666 6666155 5556	0.66282 4158028 058

128	16	0.4	0	0.75562 7009646 3023 0.72347	0.76666 666666 66667 0.79333	0.60209 4240522 4637 0.58048	0.76666 6666155 5556 0.79333	0.67448 6753846 2898 0.67042
128	16	0.4	0.2	2668810 2894 0.71061	333333 33333 0.82666	7804594 884 0.57943	3332804 4445 0.82666	2486033 727 0.68131
128	16	0.4	0.4	0932475 8842 0.79099	666666 66667 0.79099	9252065 6826 0.63068	6666115 5556 0.73999	8632490 0408 0.68098
128	32	0	0	6784565 9164 0.76527	0.74 0.74	1817823 4762 0.60962	9999506 6667 0.75999	1544992 2879 0.67655
128	32	0.2	0	3311897 1061 0.71382	0.76 0.76	5668123 1949 0.57004	9999493 3334 0.78666	7813702 6865 0.66106
128	32	0.2	0.2	6366559 4855 0.77491	666666 66666 0.76666	8308903 3583 0.62162	6666142 2222 0.76666	4376674 5951 0.68656
128	32	0.4	0	9614147 91 0.75562	666666 66667 0.79333	1621285 6099 0.61025	6666155 5556 0.79333	7114314 9957 0.68985
128	32	0.4	0.2	7009646 3023 0.73311	333333 33333 0.79333	6409943 4583 0.58910	3332804 4445 0.79333	5022914 5172 0.67613
128	32	0.4	0.4	8971061 0932 0.77170	333333 33333 0.74	8910599 451 0.60989	3332804 4445 0.73999	6314343 3662 0.66867
128	64	0	0	4180064 3086 0.76527	0.74 0.74	0109555 0054 0.60962	9999506 6667 0.75999	4648856 877 0.67655
128	64	0.2	0	3311897 1061 0.76205	0.76 0.76	5668123 1949 0.60846	9999493 3334 0.76666	7813702 6865 0.67846
128	64	0.2	0.2	7877813 5049 0.76848	666666 66667 0.76666	5608143 669 0.60655	6666155 5556 0.73999	6026957 6528 0.66666
128	64	0.4	0	8745980 7074 0.74919	0.74 0.74	7376717 7282 0.59374	9999506 6667 0.75999	6616757 3015 0.66666
128	64	0.4	0.2	6141479 0996 0.72668	0.76 0.76	9999690 7553 0.57711	9999493 3334 0.77333	6617030 8849 0.66096
128	64	0.4	0.4	8102893 8906 0.79099	333333 33333 0.72666	4427573 5749 0.62643	3332817 7778 0.72666	8611647 6363 0.67283
128	128	0	0	6784565 9164 0.78456	666666 66667 0.73333	6781249 1742 0.62146	6666182 2222 0.73333	9456031 8586 0.67278
128	128	0.2	0	5916398 7139 0.74919	333333 33333 0.74	8926202 56 0.58730	3332844 4445 0.73999	2824547 133 0.65486
128	128	0.2	0.2	6141479 0996 0.79099	0.74 0.74	1586990 8458 0.62643	9999506 6667 0.72666	7206912 5784 0.67283
128	128	0.4	0	6784565 9164 0.76205	666666 66667 0.72666	6781249 1742 0.59562	6666182 2222 0.72666	9456031 8586 0.65465
128	128	0.4	0.2	7877813 5049 0.75562	666666 66667 0.76666	8414975 0665 0.60209	6666182 2222 0.76666	4604752 5039 0.67448

128	128	0.4	0.4	0.76527		0.60326	0.73999	0.66467
				3311897	0.74	0869237	9999506	0608802
				1061		3582	6667	7574
				0.79421		0.62130	0.69999	0.65830
256	8	0	0	2218649	0.7	1774780	9999533	7159795
				5176		2948	3334	9963
				0.79421	0.71333	0.62573	0.71333	0.66666
256	8	0.2	0	2218649	333333	0993786	3332857	6616465
				5176	33334	1223	7778	2943
				0.75562	0.77333	0.60416	0.77333	0.67836
256	8	0.2	0.2	7009646	333333	6666351	3332817	2523456
				3023	33333	9966	7778	7938
				0.84887		0.66428	0.61999	0.64137
256	8	0.4	0	4598070	0.62	5713811	9999586	9259961
				7395		2245	6667	954
				0.77170		0.61621	0.75999	0.68059
256	8	0.4	0.2	4180064	0.76	6215883	9999493	6965064
				3086		1264	3334	8288
				0.73633		0.58163	0.75999	0.65895
256	8	0.4	0.4	4405144	0.76	2652764	9999493	9488075
				6945		4732	3334	1147
				0.72668	0.76666	0.57499	0.76666	0.65714
256	16	0	0	8102893	666666	9999712	6666155	2807787
				8906	66667	5	5556	7588
				0.74919	0.79333	0.60406	0.79333	0.68587
256	16	0.2	0	6141479	333333	0913398	3332804	8913057
				0996	33333	9538	4445	9975
				0.77170	0.74666	0.61202	0.74666	0.67267
256	16	0.2	0.2	4180064	666666	1857589	6666168	2622759
				3086	66667	0591	8889	7003
				0.76848	0.70666	0.59550	0.70666	0.64634
256	16	0.4	0	8745980	666666	5617642	6666195	1413384
				7074	66667	9744	5556	8937
				0.72668		0.58536	0.79999	0.67605
256	16	0.4	0.2	8102893	0.8	5853372	9999466	6288847
				8906		9923	6667	4544
				0.74598	0.82666	0.61083	0.82666	0.70254
256	16	0.4	0.4	0707395	666666	7438122	6666115	9525799
				4984	66667	7402	5556	9054
				0.80064	0.70666	0.63095	0.70666	0.66666
256	32	0	0	3086816	666666	2380576	6666195	6616407
				7203	66667	8141	5556	5826
				0.76848	0.79333	0.62303	0.79333	0.69794
256	32	0.2	0	8745980	333333	6648888	3332804	7164389
				7074	33333	4626	4445	7147
				0.72668	0.78666	0.58128	0.78666	0.66855
256	32	0.2	0.2	8102893	666666	0787890	6666142	5191541
				8906	66666	9947	2222	5455
				0.81028		0.64670		0.68138
256	32	0.4	0	9389067	0.72	6586439	0.71999	7962332
				5241		0978	999952	1992
				0.74276	0.75333	0.58549	0.75333	0.65889
256	32	0.4	0.2	5273311	333333	2227675	3332831	2078681
				8971	33333	9108	1111	5055
				0.69453	0.81333	0.56221	0.81333	0.66485
256	32	0.4	0.4	3762057	333333	1981307	3332791	0087543
				8779	33334	7364	1112	901
				0.76848	0.74666	0.60869	0.74666	0.67065
256	64	0	0	8745980	666666	5651843	6666168	8632751
				7074	66667	1003	8889	2676

256	64	0.2	0	0.77170 4180064 3086	0.76	0.61621 6215883 1264	0.75999 9999493 3334	0.68059 6965064 8288
256	64	0.2	0.2	0.73311 8971061 0932	0.78666 666666 66666	0.58706 4676324 8435	0.78666 6666142 2222	0.67236 4623037 1543
256	64	0.4	0	0.72347 2668810 2894	0.78666 666666 66666	0.57843 1372265 4748	0.78666 6666142 2222	0.66666 6617453 4811
256	64	0.4	0.2	0.72347 2668810 2894	0.78 666666 66666	0.57635 4679519 0371	0.77999 999948 0371	0.66288 9469165 1521
256	64	0.4	0.4	0.73954 9839228 2959	0.76666 666666 66667	0.58673 4693578 1966	0.76666 6666155 5556	0.66473 9834892 5827
256	128	0	0	0.74276 5273311 8971	0.74666 666666 66667	0.58333 3333029 514	0.74666 6666168 8889	0.65497 0710604 9761
256	128	0.2	0	0.76848 8745980 7074	0.74 666666 66666	0.60655 7376717 7282	0.73999 9999506 6667	0.66666 6616757 3015
256	128	0.2	0.2	0.74598 0707395 4984	0.76 666666 66666	0.59067 3574823 4853	0.75999 9999493 3334	0.66472 2982468 1931
256	128	0.4	0	0.76848 8745980 7074	0.75333 333333 33333	0.61081 0810480 6428	0.75333 3332831 1111	0.67462 6815814 6616
256	128	0.4	0.2	0.75562 7009646 3023	0.72666 666666 66667	0.58918 9188870 7086	0.72666 6666182 2222	0.65074 6218813 9935
256	128	0.4	0.4	0.77170 4180064 3086	0.74666 666666 66667	0.61202 1857589 0591	0.74666 6666168 8889	0.67267 2622759 7003

Όπως ήταν αναμενόμενο η ακρίβεια των αρνητικών αποτελεσμάτων μειώθηκε ενώ των θετικών αυξήθηκε.

3.2.2.2 Αποτελέσματα συμπίεσμένων δεδομένων

Εφαρμόζοντας το ίδιο δίκτυο στα συμπίεσμένα δεδομένα με την μέθοδο PCA με χρήση βαρών παίρνουμε τα παρακάτω αποτελέσματα.

Units	Batch	Rec drop	Dropout	Zeros acc	Ones acc	Precision	Recall	Fscore
16	8	0	0	0.7299035 36977492	0.6866 666666 666666	0.5508 021387 428865	0.6866 666662 088889	0.61127 591463 16377

16	8	0.2	0	0.7234726 688102894	0.72	0.5567 010306 408757	0.7199 999995 2	0.62790 692719 71375
16	8	0.2	0.2	0.7234726 688102894	0.7266 666666 666667	0.5589 743586 877055	0.7266 666661 822222	0.63188 400845 53703
16	8	0.4	0	0.7363344 051446945	0.7	0.5614 973259 029427	0.6999 999995 333334	0.62314 535082 63738
16	8	0.4	0.2	0.6784565 916398714	0.7466 666666 666667	0.5283 018865 432538	0.7466 666661 688889	0.61878 448151 15572
16	8	0.4	0.4	0.6720257 234726688	0.7666 666666 666667	0.5299 539168 064729	0.7666 666661 555556	0.62670 294860 01122
16	16	0	0	0.7202572 347266881	0.74	0.5606 060603 229263	0.7399 999995 066667	0.63793 098506 73839
16	16	0.2	0	0.7491961 414790996	0.6866 666666 666666	0.5690 607731 662648	0.6866 666662 088889	0.62235 644553 08042
16	16	0.2	0.2	0.6881028 938906752	0.76	0.5402 843599 335145	0.7599 999994 933334	0.63157 889844 61483
16	16	0.4	0	0.7491961 414790996	0.6933 333333 333334	0.5714 285711 145997	0.6933 333328 711111	0.62650 597418 34847
16	16	0.4	0.2	0.6881028 938906752	0.7333 333333 333333	0.5314 009659 268595	0.7333 333328 444445	0.61624 644952 8835
16	16	0.4	0.4	0.7106109 324758842	0.7466 666666 666667	0.5544 554452 700716	0.7466 666661 688889	0.63636 358709 32373
16	32	0	0	0.7556270 096463023	0.6866 666666 666666	0.5754 189940 919447	0.6866 666662 088889	0.62613 976763 70362
16	32	0.2	0	0.7299035 36977492	0.6933 333333 333334	0.5531 914890 674513	0.6933 333328 711111	0.61538 456565 2467
16	32	0.2	0.2	0.7073954 98392283	0.7266 666666	0.5449 999997	0.7266 666661	0.62285 709352

					666667	275	822222	16364
16	32	0.4	0	0.7459807 073954984	0.6866 666666 666666	0.5659 340656 231132	0.6866 666662 088889	0.62048 187780 15716
16	32	0.4	0.2	0.6913183 279742765	0.7533 333333 333333	0.5406 698562 006365	0.7533 333328 311111	0.62952 641339 53064
16	32	0.4	0.4	0.6559485 530546624	0.76	0.5158 371038 389877	0.7599 999994 933334	0.61455 520756 46104
16	64	0	0	0.7363344 051446945	0.6733 333333 333333	0.5519 125680 044193	0.6733 333328 844445	0.60660 655673 33139
16	64	0.2	0	0.7491961 414790996	0.6733 333333 333333	0.5642 458097 406449	0.6733 333328 844445	0.61398 171293 3181
16	64	0.2	0.2	0.7524115 75562701	0.6933 333333 333334	0.5745 856350 416654	0.6933 333328 711111	0.62839 874159 96607
16	64	0.4	0	0.7234726 688102894	0.72	0.5567 010306 408757	0.7199 999995 2	0.62790 692719 71375
16	64	0.4	0.2	0.7106109 324758842	0.7266 666666 666667	0.5477 386931 920911	0.7266 666661 822222	0.62464 178443 85552
16	64	0.4	0.4	0.6848874 59807074	0.7533 333333 333333	0.5355 450234 428697	0.7533 333328 311111	0.62603 873224 42315
16	128	0	0	0.9356913 183279743	0.3666 666666 666666 4	0.7333 333323 555556	0.3666 666664 222222 3	0.48888 884400 98806
16	128	0.2	0	0.9742765 273311897	0.22	0.8048 780468 173706	0.2199 999998 533333 5	0.34554 970414 188535
16	128	0.2	0.2	0.9710610 932475884	0.3133 333333 333333 5	0.8392 857127 869898	0.3133 333331 244444 5	0.45631 063957 96056

16	128	0.4	0	0.9228295 819935691	0.5	0.7575 757568 105296	0.4999 999996 666667	0.60240 959016 79043
16	128	0.4	0.2	0.9710610 932475884	0.16	0.7272 727250 688705	0.1599 999998 933333 3	0.26229 505211 86094
16	128	0.4	0.4	1.0	0.0	0.0	0.0	0.0
32	8	0	0	0.7395498 392282959	0.6933 333333 333334	0.5621 621618 582907	0.6933 333328 711111	0.62089 547256 31584
32	8	0.2	0	0.7363344 051446945	0.7	0.5614 973259 029427	0.6999 999995 333334	0.62314 535082 63738
32	8	0.2	0.2	0.6591639 871382636	0.76	0.5181 818179 46281	0.7599 999994 933334	0.61621 616767 27577
32	8	0.4	0	0.7652733 118971061	0.66	0.5755 813950 141969	0.6599 999995 6	0.61490 678214 96124
32	8	0.4	0.2	0.7106109 324758842	0.7133 333333 333334	0.5431 472078 461181	0.7133 333328 577778	0.61671 464796 8179
32	8	0.4	0.4	0.6881028 938906752	0.7466 666666 666667	0.5358 851672 077105	0.7466 666661 688889	0.62395 538275 77415
32	16	0	0	0.7395498 392282959	0.7066 666666 666667	0.5668 449194 829707	0.7066 666661 955556	0.62908 006892 37419
32	16	0.2	0	0.7363344 051446945	0.6866 666666 666666	0.5567 567564 558071	0.6866 666662 088889	0.61492 532331 29912
32	16	0.2	0.2	0.6913183 279742765	0.76	0.5428 571425 986395	0.7599 999994 933334	0.63333 328437 03742
32	16	0.4	0	0.7299035 36977492	0.7	0.5555 555552 616108	0.6999 999995 333334	0.61946 897684 49669
32	16	0.4	0.2	0.7459807 073954984	0.6866 666666	0.5659 340656	0.6866 666662	0.62048 187780

					666666	231132	088889	15716
32	16	0.4	0.4	0.7266881 028938906	0.72	0.5595 854919 380386	0.7199 999995 2	0.62973 755974 8068
32	32	0	0	0.7395498 392282959	0.7066 666666 666667	0.5668 449194 829707	0.7066 666661 955556	0.62908 006892 37419
32	32	0.2	0	0.7299035 36977492	0.7133 333333 333334	0.5602 094237 904663	0.7133 333328 577778	0.62756 593275 94402
32	32	0.2	0.2	0.7234726 688102894	0.7133 333333 333334	0.5544 041447 904642	0.7133 333328 577778	0.62390 665596 13805
32	32	0.4	0	0.7170418 006430869	0.72	0.5510 204078 821325	0.7199 999995 2	0.62427 740717 03069
32	32	0.4	0.2	0.7299035 36977492	0.7266 666666 666667	0.5647 668390 856131	0.7266 666661 822222	0.63556 846353 47555
32	32	0.4	0.4	0.7041800 643086816	0.7466 666666 666667	0.5490 196075 7401	0.7466 666661 688889	0.63276 831238 78872
32	64	0	0	0.7427652 733118971	0.6933 333333 333334	0.5652 173909 971645	0.6933 333328 711111	0.62275 444116 31866
32	64	0.2	0	0.7556270 096463023	0.7	0.5801 104969 170661	0.6999 999995 333334	0.63444 103766 85172
32	64	0.2	0.2	0.7331189 710610932	0.7066 666666 666667	0.5608 465605 498166	0.7066 666661 955556	0.62536 868185 62355
32	64	0.4	0	0.7331189 710610932	0.7133 333333 333334	0.5631 578944 404433	0.7133 333328 577778	0.62941 171502 76854
32	64	0.4	0.2	0.7202572 347266881	0.7266 666666 666667	0.5561 224486 958559	0.7266 666661 822222	0.63005 775398 77749
32	64	0.4	0.4	0.7106109 324758842	0.7266 666666 666667	0.5477 386931 920911	0.7266 666661 822222	0.62464 178443 85552

32	128	0	0	0.9003215 434083601	0.48	0.6990 291255 349232	0.4799 999996 8	0.56916 991174 99143
32	128	0.2	0	0.9324758 84244373	0.48	0.7741 935475 546307	0.4799 999996 8	0.59259 254485 59709
32	128	0.2	0.2	0.9228295 819935691	0.4933 333333 333333 5	0.7551 020400 458143	0.4933 333330 044445	0.59677 414526 53524
32	128	0.4	0	0.9163987 138263665	0.52	0.7499 999992 788462	0.5199 999996 533333	0.61417 317950 27628
32	128	0.4	0.2	0.9453376 205787781	0.3866 666666 666666 6	0.7733 333323 022222	0.3866 666664 088889	0.51555 551065 28433
32	128	0.4	0.4	0.9485530 546623794	0.4133 333333 333333 3	0.7948 717938 527285	0.4133 333330 577778	0.54385 960363 18906
64	8	0	0	0.7234726 688102894	0.7333 333333 333333	0.5612 244895 095794	0.7333 333328 444445	0.63583 810080 5243
64	8	0.2	0	0.7331189 710610932	0.7066 666666 666667	0.5608 465605 498166	0.7066 666661 955556	0.62536 868185 62355
64	8	0.2	0.2	0.7299035 36977492	0.7066 666666 666667	0.5578 947365 484764	0.7066 666661 955556	0.62352 936208 99693
64	8	0.4	0	0.7813504 823151125	0.66	0.5928 143709 025064	0.6599 999995 6	0.62460 562798 31663
64	8	0.4	0.2	0.7331189 710610932	0.7066 666666 666667	0.5608 465605 498166	0.7066 666661 955556	0.62536 868185 62355
64	8	0.4	0.4	0.6752411 575562701	0.7466 666666 666667	0.5258 215959 972669	0.7466 666661 688889	0.61707 984097 32221
64	16	0	0	0.7299035 36977492	0.72	0.5624 999997 070312	0.7199 999995 2	0.63157 889775 31586

64	16	0.2	0	0.7234726 688102894	0.72	0.5567 010306 408757	0.7199 999995 2	0.62790 692719 71375
64	16	0.2	0.2	0.7073954 98392283	0.7266 666666 666667	0.5449 999997 275	0.7266 666661 822222	0.62285 709352 16364
64	16	0.4	0	0.6977491 961414791	0.72	0.5346 534650 818547	0.7199 999995 2	0.61363 631437 88779
64	16	0.4	0.2	0.6752411 575562701	0.7733 333333 333333	0.5345 622117 352248	0.7733 333328 177778	0.63215 253987 77963
64	16	0.4	0.4	0.6591639 871382636	0.7466 666666 666667	0.5137 614676 542379	0.7466 666661 688889	0.60869 560355 03346
64	32	0	0	0.7652733 118971061	0.6466 666666 666666	0.5705 882349 584775	0.6466 666662 355556	0.60624 994981 64103
64	32	0.2	0	0.7459807 073954984	0.6733 333333 333333	0.5611 111107 993827	0.6733 333328 844445	0.61212 116216 34567
64	32	0.2	0.2	0.7556270 096463023	0.66	0.5657 142853 910204	0.6599 999995 6	0.60923 071915 172
64	32	0.4	0	0.7491961 414790996	0.6933 333333 333334	0.5714 285711 145997	0.6933 333328 711111	0.62650 597418 34847
64	32	0.4	0.2	0.7234726 688102894	0.7066 666666 666667	0.5520 833330 457899	0.7066 666661 955556	0.61988 299132 72499
64	32	0.4	0.4	0.6623794 21221865	0.7666 666666 666667	0.5227 272724 896694	0.7666 666661 555556	0.62162 157307 5241
64	64	0	0	0.7427652 733118971	0.6866 666666 666666	0.5628 415297 470811	0.6866 666662 088889	0.61861 856873 81114
64	64	0.2	0	0.7427652 733118971	0.6933 333333 333334	0.5652 173909 971645	0.6933 333328 711111	0.62275 444116 31866
64	64	0.2	0.2	0.7202572 347266881	0.7266 666666	0.5561 224486	0.7266 666661	0.63005 775398

					666667	958559	822222	77749
64	64	0.4	0	0.7299035 36977492	0.6933 333333 333334	0.5531 914890 674513	0.6933 333328 711111	0.61538 456565 2467
64	64	0.4	0.2	0.7106109 324758842	0.7533 333333 333333	0.5566 502460 312068	0.7533 333328 311111	0.64022 657965 95789
64	64	0.4	0.4	0.7202572 347266881	0.7133 333333 333334	0.5515 463914 682751	0.7133 333328 577778	0.62209 297371 21456
64	128	0	0	0.8553054 662379421	0.5733 333333 333334	0.6564 885491 171844	0.5733 333329 511111	0.61209 959392 10536
64	128	0.2	0	0.8360128 617363344	0.5866 666666 666667	0.6330 935247 243932	0.5866 666662 755555	0.60899 648944 33776
64	128	0.2	0.2	0.8778135 048231511	0.5533 333333 333333	0.6859 504126 562394	0.5533 333329 644444	0.61254 607558 17634
64	128	0.4	0	0.8745980 707395499	0.5533 333333 333333	0.6803 278682 948133	0.5533 333329 644444	0.61029 406772 81614
64	128	0.4	0.2	0.8520900 321543409	0.58	0.6541 353378 540337	0.5799 999996 133334	0.61484 093914 5205
64	128	0.4	0.4	0.8810289 389067524	0.5466 666666 666666	0.6890 756296 730457	0.5466 666663 022223	0.60966 537772 00466
128	8	0	0	0.7041800 643086816	0.74	0.5467 980292 872916	0.7399 999995 066667	0.62889 513490 67922
128	8	0.2	0	0.7491961 414790996	0.68	0.5666 666663 518519	0.6799 999995 466667	0.61818 176822 03896
128	8	0.2	0.2	0.7041800 643086816	0.7133 333333 333334	0.5376 884419 408601	0.7133 333328 577778	0.61318 046639 35475
128	8	0.4	0	0.6655948 553054662	0.7866 666666 666666	0.5315 315312 92103	0.7866 666661 422222	0.63440 855368 25104

128	8	0.4	0.2	0.7170418 006430869	0.7066 666666 666667	0.5463 917522 956744	0.7066 666661 955556	0.61627 902022 71538
128	8	0.4	0.4	0.6495176 848874598	0.76	0.5112 107621 025961	0.7599 999994 933334	0.61126 000520 66826
128	16	0	0	0.7266881 028938906	0.6866 666666 666666	0.5478 723401 341105	0.6866 666662 088889	0.60946 740589 26548
128	16	0.2	0	0.7363344 051446945	0.7133 333333 333334	0.5661 375658 380224	0.7133 333328 577778	0.63126 838686 75041
128	16	0.2	0.2	0.7170418 006430869	0.72	0.5510 204078 821325	0.7199 999995 2	0.62427 740717 03069
128	16	0.4	0	0.7781350 482315113	0.6266 666666 666667	0.5766 871162 106214	0.6266 666662 488889	0.60063 892733 82438
128	16	0.4	0.2	0.7427652 733118971	0.6933 333333 333334	0.5652 173909 971645	0.6933 333328 711111	0.62275 444116 31866
128	16	0.4	0.4	0.7524115 75562701	0.7333 333333 333333	0.5882 352938 030827	0.7333 333328 444445	0.65281 894131 32142
128	32	0	0	0.7395498 392282959	0.6933 333333 333334	0.5621 621618 582907	0.6933 333328 711111	0.62089 547256 31584
128	32	0.2	0	0.7331189 710610932	0.6933 333333 333334	0.5561 497323 229146	0.6933 333328 711111	0.61721 063272 90057
128	32	0.2	0.2	0.6881028 938906752	0.74	0.5336 538458 972818	0.7399 999995 066667	0.62011 168280 95291
128	32	0.4	0	0.7620578 778135049	0.6533 333333 333333	0.5697 674415 29205	0.6533 333328 977778	0.60869 560202 92468
128	32	0.4	0.2	0.7652733 118971061	0.6533 333333 333333	0.5730 994148 695325	0.6533 333328 977778	0.61059 185014 5092
128	32	0.4	0.4	0.6366559 485530546	0.7666 666666	0.5043 859646	0.7666 666661	0.60846 556027

					666667	910588	555556	26725
128	64	0	0	0.7395498	0.7133	0.5691	0.7133	0.63313
				392282959	333333	489358	333328	604493
					333334	67474	577778	19039
128	64	0.2	0	0.7427652	0.6933	0.5652	0.6933	0.62275
				733118971	333333	173909	333328	444116
					333334	971645	711111	31866
128	64	0.2	0.2	0.7363344	0.7266	0.5706	0.7266	0.63929
				051446945	666666	806279	666661	613803
					666667	734656	822222	11525
128	64	0.4	0	0.7331189	0.7066	0.5608	0.7066	0.62536
				710610932	666666	465605	666661	868185
					666667	498166	955556	62355
128	64	0.4	0.2	0.7620578	0.68	0.5795	0.6799	0.62576
				778135049		454542	999995	682109
						161673	466667	97818
128	64	0.4	0.4	0.6591639	0.7466	0.5137	0.7466	0.60869
				871382636	666666	614676	666661	560355
					666667	542379	688889	03346
128	128	0	0	0.7684887	0.6533	0.5764	0.6533	0.61249
				459807074	333333	705878	333328	994981
					333333	961938	977778	2504
128	128	0.2	0	0.7556270	0.66	0.5657	0.6599	0.60923
				096463023		142853	999995	071915
						910204	6	172
128	128	0.2	0.2	0.7588424	0.6666	0.5714	0.6666	0.61538
				437299035	666666	285711	666662	456530
					666666	020408	222223	17792
128	128	0.4	0	0.7556270	0.6733	0.5706	0.6733	0.61773
				096463023	333333	214686	333328	695302
					333333	041687	844445	11676
128	128	0.4	0.2	0.7749196	0.6666	0.5882	0.6666	0.62499
				1414791	666666	352937	666662	994980
					666666	716263	222223	46914
128	128	0.4	0.4	0.7717041	0.6466	0.5773	0.6466	0.61006
				800643086	666666	809520	666662	284285
					666666	372732	355556	8277
256	8	0	0	0.7588424	0.68	0.5762	0.6799	0.62385
				437299035		711861	999995	316096
						151011	466667	84971

256	8	0.2	0	0.6141479 099678456	0.8133 333333 333334	0.5041 322311 966396	0.8133 333327 911112	0.62244 893202 83252
256	8	0.2	0.2	0.8456591 639871383	0.6	0.6521 739125 708885	0.5999 999996 000001	0.62499 994965 27818
256	8	0.4	0	0.7234726 688102894	0.6933 333333 333334	0.5473 684207 64543	0.6933 333328 711111	0.61176 465621 45368
256	8	0.4	0.2	0.6720257 234726688	0.7733 333333 333333	0.5321 100914 990321	0.7733 333328 177778	0.63043 473397 33024
256	8	0.4	0.4	0.6366559 485530546	0.7866 666666 666666	0.5108 225106 013755	0.7866 666661 422222	0.61942 252411 32297
256	16	0	0	0.6977491 961414791	0.76	0.5480 769228 134246	0.7599 999994 933334	0.63687 145933 64789
256	16	0.2	0	0.7266881 028938906	0.7	0.5526 315786 565097	0.6999 999995 333334	0.61764 700915 2253
256	16	0.2	0.2	0.7524115 75562701	0.72	0.5837 837834 68225	0.7199 999995 2	0.64477 606956 38265
256	16	0.4	0	0.7588424 437299035	0.6933 333333 333334	0.5810 055862 675947	0.6933 333328 711111	0.63221 879498 89637
256	16	0.4	0.2	0.8038585 209003215	0.62	0.6038 961035 039636	0.6199 999995 866667	0.61184 205486 92908
256	16	0.4	0.4	0.7395498 392282959	0.6933 333333 333334	0.5621 621618 582907	0.6933 333328 711111	0.62089 547256 31584
256	32	0	0	0.6623794 21221865	0.78	0.5270 270267 896274	0.7799 999994 8	0.62903 220959 93792
256	32	0.2	0	0.7427652 733118971	0.72	0.5744 680848 008149	0.7199 999995 2	0.63905 320469 17163
256	32	0.2	0.2	0.7845659 163987139	0.66	0.5963 855418	0.6599 999995	0.62658 222821

						094063	6	26301
256	32	0.4	0	0.7491961 414790996	0.6666 666666 666666	0.5617 977524 93372	0.6666 666662 222223	0.60975 604755 35436
256	32	0.4	0.2	0.7524115 75562701	0.6866 666666 666666	0.5722 222219 04321	0.6866 666662 088889	0.62424 237427 73227
256	32	0.4	0.4	0.7491961 414790996	0.7133 333333 333334	0.5783 783780 657414	0.7133 333328 577778	0.63880 592031 36595
256	64	0	0	0.7138263 665594855	0.72	0.5482 233499 755211	0.7199 999995 2	0.62247 833672 56638
256	64	0.2	0	0.7491961 414790996	0.6866 666666 666666	0.5690 607731 662648	0.6866 666662 088889	0.62235 644553 08042
256	64	0.2	0.2	0.7491961 414790996	0.7	0.5737 704914 897429	0.6999 999995 333334	0.63063 058074 2909
256	64	0.4	0	0.7620578 778135049	0.6533 333333 333333	0.5697 674415 29205	0.6533 333328 977778	0.60869 560202 92468
256	64	0.4	0.2	0.6848874 59807074	0.7666 666666 666667	0.5399 061030 32908	0.7666 666661 555556	0.63360 876658 39498
256	64	0.4	0.4	0.7909967 845659164	0.6333 333333 333333	0.5937 499996 289063	0.6333 333329 111112	0.61290 317546 30634
256	128	0	0	0.7524115 75562701	0.7	0.5769 230766 060862	0.6999 999995 333334	0.63253 007056 53979
256	128	0.2	0	0.7363344 051446945	0.7066 666666 666667	0.5638 297869 341331	0.7066 666661 955556	0.62721 888517 20917
256	128	0.2	0.2	0.7459807 073954984	0.7	0.5706 521736 029064	0.6999 999995 333334	0.62874 246511 16967
256	128	0.4	0	0.7524115 75562701	0.68	0.5698 324019 162948	0.6799 999995 466667	0.62006 074028 51084

256	128	0.4	0.2	0.7363344 051446945	0.7266 666666 666667	0.5706 806279 734656	0.7266 666661 822222	0.63929 613803 11525
256	128	0.4	0.4	0.7491961 414790996	0.72 451609 781478	0.5806 999995 2	0.7199 709304 84732	0.64285

3.2.2.3 Σύγκριση αποτελεσμάτων

Συγκρίνοντας τα αποτελέσματα για τα συμπιεσμένα και μη δεδομένα βρίσκουμε τις εξής διαφορές (με χρήση βαρών):

	Μη συμπιεσμένα	Συμπιεσμένα
M.O ακρίβεια άσπων	0,7361777778	0,6755555556
M.O ακρίβεια μηδενικών	0,7617577706	0,7499678457
M.O Precision	0,6028394312	0,5736198187
M.O Recall	0,7361777773	0,6755555551
M.O Fscore	0,6559343489	0,6102643318

Όπως φαίνεται τα μη συμπιεσμένα δεδομένα παράγουν καλύτερα αποτελέσματα. Αυτό συμβαίνει καθώς κατά την συμπίεση των δεδομένων όπως είναι αναμενόμενο χάνουμε πάντα κάποια πληροφορία. Συγκρίνοντας τα αποτελέσματα με αυτά του mammography dataset βλέπουμε μεγαλύτερες διαφορές μεταξύ των συμπιεσμένων και μη δεδομένων. Αυτό βεβαίως είναι αναμενόμενο καθώς το συγκεκριμένο dataset δέχτηκε συμπίεση από 8 εισόδους σε 3 σε αντίθεση με το mammography dataset που δέχτηκε συμπίεση από 6 εισόδους σε 4.

3.3.3 Εφαρμογή CNN δίκτυου

3.3.3.1 Γενικά

Έχοντας εφαρμόσει ένα RNN δίκτυο, στην συνέχεια θα δημιουργήσουμε ένα δίκτυο CNN και θα συγκρίνουμε τα αποτελέσματα των δυο. Ο κώδικας του

CNN δικτύου φαίνεται παρακάτω.

```

model = Sequential()
model.add(layers.Conv1D(num_layers*4, 2, padding='same', activation='relu',
input_shape=(num_inputs, 1)))
model.add(layers.Conv1D(num_layers*2, 3, padding='same', activation='relu'))
model.add(layers.Conv1D(num_layers, 4, padding='same', activation='relu'))
model.add(layers.GlobalMaxPooling1D())
model.add(layers.Dense(1))
model.compile(optimizer='RMSprop', loss='binary_crossentropy', metrics=['acc'])
model.summary()

history = model.fit(samples_expanded, traintargets, shuffle=False, batch_size=32,
epochs=50, validation_split=0.2, class_weight={0:0.38, 1:0.61})

```

Όπως φαίνεται εφαρμόζουμε 3 συνελκτικά επίπεδα στην σειρά τα οποία στην συνέχεια συνδέονται με ένα maxpooling επίπεδο που εξάγει μόνο τις μεγαλύτερες τιμές.

3.3.3.2 Αποτελέσματα

Αποτελέσματα μη συμπιεσμένων δεδομένων (8 είσοδοι)

layers output size							
1st	2nd	3rd	Zeros accuracy	Ones accuracy	precision	recall	fscore
32	16	8	0.7073954 98392283	0.786666 66666666 66	0.56459330 11652664	0.786666 6661422 222	0.657381566 5831307
64	32	16	0.7106109 32475884 2	0.733333 33333333 33	0.54999999 9725	0.733333 3328444 445	0.628571379 2326568
128	64	32	0.7588424 43729903 5	0.693333 33333333 34	0.58100558 62675947	0.693333 3328711 111	0.632218794 9889637

256	128	64	0.7459807 07395498 4	0.66	0.55617977 49684384	0.659999 99956	0.603658486 5816519
-----	-----	----	----------------------------	------	------------------------	-------------------	------------------------

Αποτελέσματα συμπιεσμένων δεδομένων (3 είσοδοι)

layers output size			Zeros accuracy	Ones accuracy	precision	recall	fscore
1st	2nd	3rd					
32	16	8	0.6688102 89389067 5	0.726666 66666666 67	0.51415094 31537024	0.726666 66618222 22	0.60220989 58853553
64	32	16	0.6655948 55305466 2	0.773333 33333333 33	0.52727272 70330578	0.773333 33281777 78	0.62702697 84777247
128	64	32	0.5594855 30546623 8	0.853333 33333333 34	0.48301886 7742257	0.853333 33276444 45	0.61686742 34216901
256	128	64	0.5273311 89710610 9	0.833333 33333333 34	0.45955882 33604563	0.833333 33277777 78	0.59241701 55095383

Τα αποτελέσματα του CNN νευρωνικού δικτύου είναι σαφώς καλύτερα σε σχέση με τα αντίστοιχα CNN δίκτυα του mammography dataset. Αν και υστερούν λίγο σε σχέση με τα RNN δίκτυα, τα αποτελέσματα είναι αρκετά ικανοποιητικά. Τα αποτελέσματα των συμπιεσμένων δεδομένων είναι επίσης λίγο χειρότερα από τα αντίστοιχα μη συμπιεσμένα.

Τα δίκτυα CNN εφαρμόζονται καλύτερα σε προβλήματα όπως η κατηγοριοποίηση εικόνων. Στην δική μας περίπτωση τα δεδομένα αποτελούνται από έναν μονοδιάστατο πίνακα λίγων τιμών (8 τιμές στην περίπτωση του pima dataset και 3 για τα συμπιεσμένα δεδομένα). Σε αυτήν την περίπτωση τα δίκτυα τύπου RNN έχουν σαφώς πλεονέκτημα μιας και κατέχουν

στην μνήμη του παλαιότερες καταστάσεις του συστήματος.

Κεφάλαιο 4

4.1 Συμπεράσματα

Η τεχνητή νοημοσύνη και ειδικότερα ο τομέας της βαθιάς μάθησης έχει εξελιχθεί ραγδαία τα τελευταία χρόνια και όλα δείχνουν πως ο κλάδος αυτός θα επιφέρει σημαντικές βελτιώσεις στην ζωή του ανθρώπου τα επόμενα χρόνια.

Έχοντας μελετήσει μεθόδους δημιουργίας νευρωνικών δικτύων με την βοήθεια της γλώσσας προγραμματισμού `python` και των βιβλιοθηκών `Keras` και `Tensorflow` καταφέραμε να εφαρμόσουμε τα δίκτυα αυτά σε πραγματικά δεδομένα με σκοπό την πρόβλεψη μελλοντικών καταστάσεων. Επίσης είδαμε πως η μη ισορροπημένη κατανομή των δεδομένων μπορεί να επηρεάσει αρνητικά την ανταπόκριση και την ακρίβεια του νευρωνικού δικτύου. Στην συνέχεια είδαμε κατα πόσο η συμπίεση των δεδομένων εισόδων μπορεί να επηρεάσει το τελικό αποτέλεσμα ενός συστήματος. Παρα τις μη ισορροπημένες κατανομές των `datasets` και ειδικά του `mammography dataset` καταφέραμε να έχουμε ίση ακρίβεια μεταξύ των προβλέψεων θετικών και αρνητικών αποτελεσμάτων. Αυτό βέβαια μείωσε αρκετά την ακρίβεια του `fscore` μιας και μεγάλωσαν οι αστοχίες για τα αρνητικά αποτελέσματα τα οποία αποτελούσαν το μεγαλύτερο κομμάτι των δεδομένων.

Όπως φάνηκε και από τα αποτελέσματα που λάβαμε από τη εκπαίδευση των μοντέλων, τα δίκτυα `RNN` λειτούργησαν καλύτερα στα συγκεκριμένα δεδομένα από τα δίκτυα `CNN`. Αυτό συνέβη λόγω του ότι τα δίκτυα `RNN` συγκρατούν στην μνήμη παλαιότερες καταστάσεις του συστήματος και αυτό τα καθιστά πολύ πιο αποτελεσματικά στην περίπτωση των δεδομένων μας.

Επίσης το σύνολο των δεδομένων αποτελείται από στιγμιότυπα με μικρό αριθμό εισόδων (8 στην περίπτωση του `rima` και 6 στην περίπτωση του `mammography`). Αυτό καθιστά δύσκολο την εύρεση μοτίβων από τα δίκτυα `CNN` που θα μπορούσε να βοηθήσει στην κατηγοριοποίηση των δεδομένων. Τα δίκτυα αυτά βρίσκουν καλύτερες εφαρμογές σε προβλήματα κατηγοριοποίησης εικόνων. Αυτός είναι ο λόγος για τον οποίο τα δίκτυα `CNN` εφαρμόζονται όλο και πιο συχνά μιας και πλέον ζούμε στην εποχή της εικόνας.

Πιθανές βελτιώσεις για εκπαίδευση με μη ισορροπημένα δεδομένα σε προβλήματα δυαδικής κατηγοριοποίησης:

- **Undersampling:** Εισαγωγή λιγότερων δεδομένων των οποίων η κατηγορία αποτελεί την πλειοψηφία των δεδομένων.
- **Oversampling:** Εισαγωγή περισσότερων (διπλότυπων) δεδομένων των οποίων η κατηγορία αποτελεί την μειοψηφία των δεδομένων.
- **Δημιουργία συνθετικών δεδομένων** των οποίων η κατηγορία αποτελεί την μειοψηφία των δεδομένων.
- **Χρήση συνδυασμού νευρικών δικτύων.**

4.4 Κώδικας

Κώδικας δημιουργίας RNN δικτύου με χρήση βαρών (κεφάλαιο 3.2.2.1).

```
import os
import numpy as np
import tensorflow as tf
from matplotlib import pyplot as plt
from keras import backend as K
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras import activations
from tensorflow.keras import metrics
from tensorflow.keras.optimizers import RMSprop
import random

#Φάκελος όπου βρίσκεται το αρχείο
data_dir = r'C:\Users\Jake\Desktop\mammography'

#Αριθμός εισόδων (αναγκαίο για την συνάρτηση που διαβάζει τα αρχεία,
εξαρτάται από το dataset)
num_inputs = 6

#Συνάρτηση για διαβάσμα δεδομένων από αρχείο και αποθήκευση σε πίνακα
def readdatafromfile(file):
    fname = os.path.join(data_dir, file)
    f = open(fname)
```

```

data = f.read()
f.close()
lines = data.split('\n')
samples = np.zeros((len(lines), num_inputs))
targets = np.zeros(len(lines), )

for i,line in enumerate(lines):
    line_data = line.split(' ')
    samples[i,:] = [float(x) for x in line_data[0:num_inputs]]
    targets[i] = int(line_data[num_inputs])

return samples, targets

#συνάρτηση υπολογισμού fscore
def fscore(true,pred):
    true_positive = 0
    false_positive = 0
    true_negative = 0
    false_negative = 0
    precision = 0.0
    recall = 0.0
    fscore = 0.0
    for i, t in enumerate(true):
        if(pred[i] == 1):
            if(pred[i] == true[i]):
                true_positive += 1
            else:
                false_positive += 1

        elif(pred[i] == 0):
            if(pred[i] == true[i]):
                true_negative += 1
            else:
                false_negative += 1

    precision = true_positive / ( true_positive + false_positive +
0.0000001 )
    recall = true_positive / ( true_positive + false_negative +
0.0000001 )
    fscore = 2*(( precision * recall ) / ( precision + recall +
0.0000001 ))

```

```

    return precision, recall, fscore

#Συμπιεσμένο dataset
# trainsamples, traintargets = readdatafromfile('PCA transformed
datasets_4 inputs\Training
dataset\pca_transformed_m_mammo_trn_binary01.dat')
# testsamples, testtargets = readdatafromfile('PCA transformed
datasets_4 inputs\Testing
dataset\pca_transformed_m_mammo_tst_binary01.dat')

#Μη συμπιεσμένο dataset
trainsamples, traintargets = readdatafromfile('Initial datasets_6
inputs\Training dataset\mammo_trn_binary01.dat')
testsamples, testtargets = readdatafromfile('Initial datasets_6
inputs\Testing dataset\mammo_tst_binary01.dat')

negativeres = 0
positiveres = 0

#Υπολογισμός θετικών και αρνητικών αποτελεσμάτων με σκοπό τον
υπολογισμό της ακρίβεια στο τέλος
for i,sample in enumerate(testtargets):
    if sample == 0:
        negativeres+=1
    else:
        positiveres+=1

#κανονικοποίηση δεδομένων
mean = trainsamples.mean(axis=0)
trainsamples -= mean
std = trainsamples.std(axis=0)
trainsamples /= std

mean = testsamples.mean(axis=0)
testsamples -= mean
std = testsamples.std(axis=0)
testsamples /= std

#Επέκταση δεδομένων σε 3 διαστάσεις (αναγκαίο για τα RNN μοντέλα)
samples_expanded = np.expand_dims(trainsamples, axis=1)
test_samples_expanded = np.expand_dims(testsamples, axis=1)

```

```

layers_num=16
batch = 16
rec_dropout=0
drop=0
validationsplit=0.2

#αρχείο όπου θα αποθηκεύονται τα αποτελέσματα
csvfile = open("output.csv","w")
csvfile.write("GRU_output_units;Batch;Reccurent_Dropout;Dropout;Zeros
_prediction_acc;Ones_prediction_acc;Precision;Recall;f1-score\n")
csvfile.close()

#Βρόγχος επανάληψης για την εκπαίδευση δικτύων με διαφορετικές
παραμέτρους
while layers_num<=256:
    batch = 16
    while batch<=256:
        rec_dropout=0
        while rec_dropout<=0.6:
            drop=0
            while drop<=rec_dropout:
                steps=(len(trainsamples) -
validationsplit*len(trainsamples)) // batch

                #Δημιουργία του δικτύου
                model = Sequential()
                model.add(layers.GRU(layers_num, input_shape=(1,
num_inputs), return_sequences=True, dropout=drop,
recurrent_dropout=rec_dropout ))
                model.add(layers.Dense(1))
                model.compile(optimizer='RMSprop', loss='mse',
metrics=['acc'])
                history = model.fit(x=samples_expanded,
y=traintargets, batch_size=batch, steps_per_epoch=steps, epochs=50,
class_weight={0:0.02, 1:1.0}, shuffle=True,
validation_split=validationsplit, verbose=1)

                preds = model.predict(test_samples_expanded,
verbose=1)
                preds = preds.reshape(len(preds),1)
                #threshold
                predsnormalized = (preds > 0.5).astype(int)
                sumpred = np.array([0,0])

```

```

        for i,pred in enumerate(predsnormalized):
            if(pred == int(testtargets[i])):
                sumpred[int(testtargets[i])]+=1

        #Υπολογισμός ακρίβειας για θετικά και αρνητικά
αποτελέσματα
        prediction_acc_zeros = sumpred[0]/negativeres
        prediction_acc_ones = sumpred[1]/positiveres
        pred_acc_avg =
(prediction_acc_ones+prediction_acc_zeros)/2

        #Κλήση συνάρτησης υπολογισμού fscore
        prec, rec, score = fscore(np.array(testtargets,
dtype="int"), predsnormalized)
        csvfile = open("output.csv","a")

csvfile.write(str(layers_num)+" "+str(batch)+" "+str(rec_dropout)+" "+
+str(drop)+" "+str(prediction_acc_zeros)+" "+str(prediction_acc_ones)
+" "+str(prec)+" "+str(rec)+" "+str(score)+"\n")

        csvfile.close()

        drop+=0.2

        rec_dropout+=0.2

        batch*=2

        layers_num*=2

```

Κώδικας δημιουργίας CNN δικτύου με χρήση βαρών (κεφάλαιο 3.3.3).

```
import os
```

```

import numpy as np
import tensorflow as tf
from matplotlib import pyplot as plt
from keras import backend as K
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras import activations
from tensorflow.keras import metrics
from tensorflow.keras.optimizers import RMSprop
import random
from tensorflow.python.keras.backend import dropout

#Συνάρτηση υπολογισμού fscore
def fscore(true,pred):
    true_positive = 0
    false_positive = 0
    true_negative = 0
    false_negative = 0
    precision = 0.0
    recall = 0.0
    fscore = 0.0
    for i, t in enumerate(true):
        if(pred[i] == 1):
            if(pred[i] == true[i]):
                true_positive += 1
            else:
                false_positive += 1

        elif(pred[i] == 0):
            if(pred[i] == true[i]):
                true_negative += 1
            else:
                false_negative += 1

    precision = true_positive / ( true_positive + false_positive +
0.0000001)
    recall = true_positive / ( true_positive + false_negative +
0.0000001)
    fscore = 2*(( precision * recall ) / ( precision + recall +
0.0000001))
    return precision, recall, fscore

```

```

data_dir = r'C:\Users\Jake\Desktop\pima'
#Αριθμός εισόδων. Εξαρτάται απο το dataset
num_inputs = 3

#Συνάρτηση για διάβασμα δεδομένων και αποθήκευση σε πίνακα
def readdatafromfile(file):
    fname = os.path.join(data_dir, file)
    f = open(fname)
    data = f.read()
    f.close()
    lines = data.split('\n')
    samples = np.zeros((len(lines), num_inputs))
    targets = np.zeros(len(lines), )

    for i,line in enumerate(lines):
        line_data = line.split(' ')
        samples[i,:] = [float(x) for x in line_data[0:num_inputs]]
        targets[i] = int(line_data[num_inputs])

    return samples, targets

#Συμπίεσμένα δεδομένα
trainsamples, traintargets = readdatafromfile('PCA transformed_3
inputs\Training dataset\pca_transformed_m_pima_trn_binary01.dat')
testsamples, testtargets = readdatafromfile('PCA transformed_3
inputs\Testing dataset\pca_transformed_m_pima_tst_binary01.dat')

#Μή συμπίεσμένα δεδομένα
# trainsamples, traintargets = readdatafromfile('Initial datasets_8
inputs\Training dataset\pima_trn_binary01.dat')
# testsamples, testtargets = readdatafromfile('Initial datasets_8
inputs\Testing dataset\pima_tst_binary01.dat')

negativeres = 0
positiveres = 0

#Υπολογισμός συνόλου θετικών και αρνητικών αποτελεσμάτων
for i,sample in enumerate(testtargets):
    if sample == 0:
        negativeres+=1

```



```

else:
    positiveres+=1

#Κανονικοποίηση δεδομένων
mean = trainsamples.mean(axis=0)
trainsamples-=mean
std = trainsamples.std(axis=0)
trainsamples/=std

mean = testsamples.mean(axis=0)
testsamples -= mean
std = testsamples.std(axis=0)
testsamples /= std

#επέκταση σε 3 διαστάσεις των δεδομένων
samples_expanded = np.expand_dims(trainsamples, axis=2)
test_samples_expanded = np.expand_dims(testsamples, axis=2)

validationsplit=0.2

num_layers = 8
csvfile = open("output.csv","w")
csvfile.write("first layer;second layer;third layer;batch size;Zeros
accuracy;Ones accuracy;preccision;recall;fscore\n")
csvfile.close()
while(num_layers <= 64):

    model = Sequential()
    model.add(layers.Conv1D(num_layers*4, 2, padding='same',
activation='relu', input_shape=(num_inputs, 1)))
    model.add(layers.Conv1D(num_layers*2, 3, padding='same',
activation='relu'))
    model.add(layers.Conv1D(num_layers, 4, padding='same',
activation='relu'))
    model.add(layers.GlobalMaxPooling1D())
    model.add(layers.Dense(1))
    model.compile(optimizer='RMSprop', loss='binary_crossentropy',
metrics=['acc'])
    model.summary()
    history = model.fit(samples_expanded, traintargets,
shuffle=False, batch_size=32, epochs=50, validation_split=0.2,

```

```

class_weight={0:0.38, 1:0.61})

preds = model.predict(test_samples_expanded, verbose=1)

predsnormalized = (preds > 0.5).astype(int)
sumpred = np.array([0,0])

for i,pred in enumerate(predsnormalized):
    if(pred == int(testtargets[i])):
        sumpred[int(testtargets[i])] +=1

#Ακρίβεια μηδενικών και άσων
prediction_acc_zeros = sumpred[0]/negativeres
prediction_acc_ones = sumpred[1]/positiveres
pred_acc_avg = (prediction_acc_ones+prediction_acc_zeros)/2

prec, rec, score = fscore(np.array(testtargets, dtype="int"),
predsnormalized)

csvfile = open("output.csv", "a")

csvfile.write(str(num_layers*4)+";" +str(num_layers*2)+";" +str(num_layers)+";" +str(prediction_acc_zeros)+";" +str(prediction_acc_ones)+";" +str(prec)+";" +str(rec)+";" +str(score)+"\n")

csvfile.close()
num_layers*=2

```

4.3 Βιβλιογραφία

1. Francois Chollet (2018). Deep Learning with Python.
2. Robert Layton (2015). Learning Data Mining with Python.

4.4 Δικτυογραφία

1. <https://deepai.org/machine-learning-glossary-and-terms/f-score>
2. <https://en.wikipedia.org/wiki/F-score>
3. https://en.wikipedia.org/wiki/Precision_and_recall
4. <https://keras.io/api/>
5. <https://datascience.stackexchange.com/questions/12964/what-is-the-meaning-of-the-number-of-units-in-the-lstm-cell>
6. https://en.wikipedia.org/wiki/Mean_squared_error
7. https://homes.di.unimi.it/borghese/Teaching/AdvancedIntelligentSystems/Old/IntelligentSystems_2008_2009/Old/IntelligentSystems_2005_2006/Documents/Symbolic/04_McCarthy_whatissai.pdf
8. https://en.wikipedia.org/wiki/Computing_Machinery_and_Intelligence
9. https://en.wikipedia.org/wiki/Turing_test
10. https://el.wikipedia.org/wiki/Tensor_Flow
11. <https://el.wikipedia.org/wiki/%CE%9F%CF%80%CE%B9%CF%83%CE%B8%CE%BF%CE%B4%CE%B9%CE%AC%CE%B4%CE%BF%CF%83%CE%B7>
12. <https://storage.googleapis.com/pub-tools-public-publication-data/pdf/40565.pdf>
13. https://en.wikipedia.org/wiki/Gradient_descent
14. <https://flatironschool.com/blog/deep-learning-vs-machine-learning>
15. https://en.wikipedia.org/wiki/Convolutional_neural_network
16. <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>