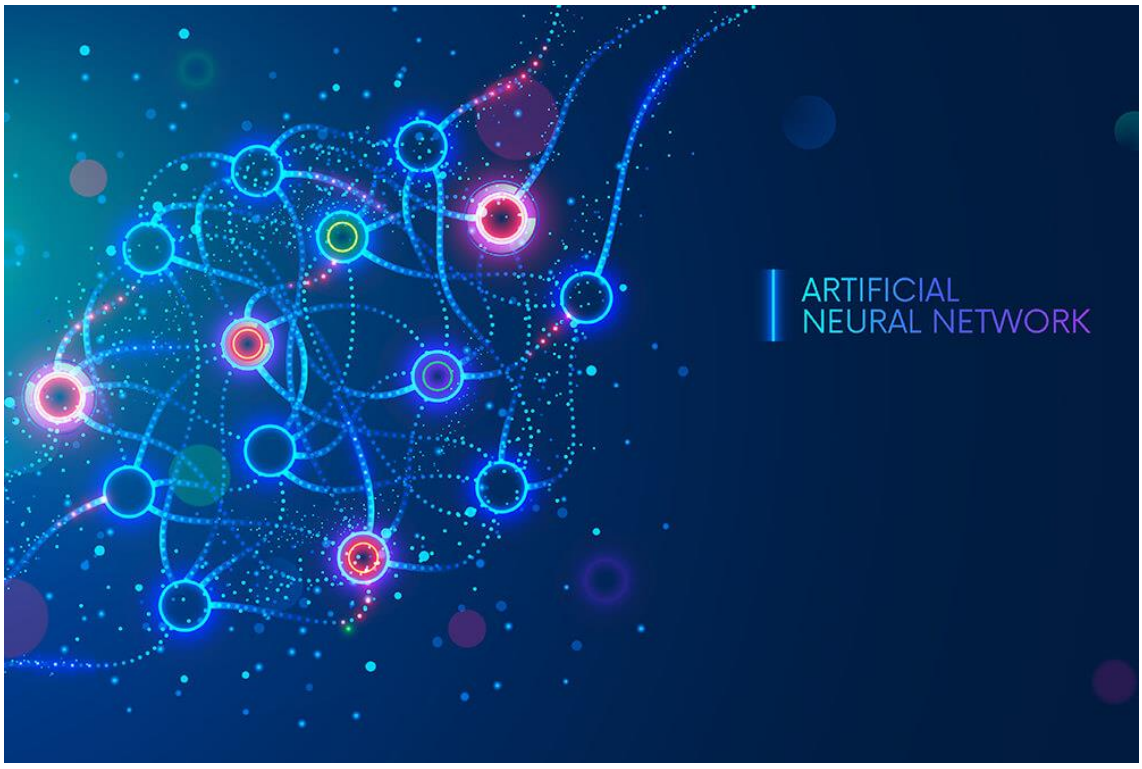
Νευρωνικά δίκτυα

Δημιουργία ενός Multi-Layer Perceptron (MLP) σε Python

Χρήστος Γάλλος

ΑΜ:220140



Μέρος Α: Θεωρητικό Μέρος

1) Ποιος είναι ο ρόλος της συνάρτησης ενεργοποίησης σε ένα MLP;

Ο ρόλος της συνάρτησης ενεργοποίησης είναι να εισάγει μη γραμμικότητα στο μοντέλο με σκοπό να μάθει πολύπλοκα μοτίβα όπως πχ. εικόνες. Η συνάρτηση ενεργοποίησης είναι επίσης υπεύθυνη να περιορίζει (κανονικοποιεί) την έξοδο ενός νευρώνα σε ένα συγκεκριμένο εύρος. Η επιλογή της σωστής συνάρτησης ενεργοποίησης επηρεάζει την επηρεάζει την συμπεριφορά της μάθησης, για αυτό είναι σημαντικό να επιλέξουμε την σωστή συνάρτηση ανάλογα με την χρήση.

Συναρτήσεις ενεργοποίησης:

Συνάρτηση	Τύπος συνάρτησης	Ιδιότητες
Sigmoid	σ(x)=1/(1+e^-x)	Κανονικοποίηση σε (0,1), καλή για πιθανότητες
Relu	ReLU(x)=max(0,x)	Απλή και γρήγορη, δεν κορεστεί
Tahn	tanh(x)=(e^x-e^-x)/(e^x+e^-x)	Κανονικοποίηση σε (-1,1), πιο κεντροποιημένη
Elu	ELU(x)=x if x>0,α(ex−1)	Μειώνει την προκατάληψη προς τα θετικά, διατηρεί μη γραμμικότητα
Softmax	Softmax(x)=e^xi/(Σj*e^xj)	Χρήσιμη για ταξινόμηση κατηγοριών, εξασφαλίζει άθροισμα 1

Επιπτώσεις στην απόδοση του Δικτύου:

Οι συναρτήσεις Sigmoid και Tanh μπορούν να προκαλέσουν εξαφάνιση του gradient (vanishing gradient) ιδιαίτερα σε μεγάλα δίκτυα, κάνοντας τη μάθηση αργή. Επίσης η Relu παρουσιάζει προβήματα όταν η είσοδος της είναι αρνητική, η έξοδος είναι πάντα 0. Αν κατά την διάρκεια της εκπαίδευσης η έξοδος είναι 0 τότε ο νευρώνας ¨πεθαίνει και δεν ξαναενεργοποιείται. Ορισμένες συναρτήσεις, όπως η **Softmax**, μπορούν να προκαλέσουν αριθμητική αστάθεια (overflow) αν οι τιμές είναι πολύ μεγάλες ή πολύ μικρές.

2) Ποια είναι τα βασικά στάδια του αλγορίθμου backpropagation;

Ο αλγόριθμος **backpropagation** είναι μια βασική μέθοδος εκπαίδευσης MLPs .Ο σκοπός του είναι η ενημέρωση των βαρών κάθε συνδέσμου στο δίκτυο ώστε να ελαχιστοποιηθεί το σφάλμα (Loss Function) ανάμεσα στην προβλεπόμενη έξοδο και την πραγματική τιμή. Αφού γίνει το forward propagation και error calculation το back propagation υπολογίζεται με βάση το error. Το error που υπολογίστηκε διαδίδεται προς τα πίσω μέσα στο δίκτυο, στρώμα προς στρώμα. Χρησιμοποιείτε ο κανόνας chain rule για να υπολογιστούν οι μερικές παράγωγοι της συνάρτησης κόστους ως προς τις παραμέτρους του δικτύου. Δηλαδή για κάθε βάρος και σταθερά όρο στο δίκτυο , υπολογίζεται πόσο επηρεάζει το συνολικό σφάλμα. Μετά από αυτή την διαδικασία γίνεται η ενημέρωση των Βαρών και σταθερών όρων με βάση τις υπολογίσιμες παραγώγους(gradients). Ο κανόνας ενημέρωσης είναι συνήθως:

Νέα παράμετρος=παλιά παράμετρο-ρυθμό μάθησης*gradient

Όπου ο ρυθμός μάθησης (learning rate) είναι μια παράμετρος που ελέγχει το μέγεθος του βήματος κατά την ενημέρωση. Αυτά τα βήματα επαναλαμβάνονται ανάλογα με το πόσες εποχές υπάρχουν στο δίκτυο.

3) Ποια είναι η διαφορά μεταξύ overfitting και underfitting;

Το overfitting συμβαίνει όταν ένα μοντέλο MLP μαθαίνει τα δεδομένα εκπαίδευσης υπερβολικά καλά. Το μοντέλο ουσιαστικά ¨απομνημονεύει¨ τα δεδομένα εκπαίδευσης αντί να τα μαθαίνει, με αποτέλεσμα να έχει εξαιρετική απόδοση στα δεδομένα εκπαίδευσης αλλά κακή απόδοση σε νέα δεδομένα που δεν έχει εκπαιδευτεί. Για να αποφύγουμε αυτό το φαινόμενο πρέπει το μοντέλο που δημιουργούμε να μην έχει υπερβολικά πολλά κρυφά στρώματα και πολλούς νευρώνες. Αντίθετα το underfitting συμβαίνει όταν το μοντέλο είναι πολύ απλό και δεν μπορεί να αναγνωρίσει και να καταγράψει τις σχέσεις στα δεδομένα εκπαίδευσης. Δηλαδή το μοντέλο έχει κακή απόδοση τόσο στα δεδομένα εκπαίδευσης όσο και στα νέα δεδομένα. Συνήθως ο λόγος που υπάρχει underfitting στο μοντέλο είναι γιατί έχουμε υπερβολικά λίγα στρώματα και λίγους νευρώνες, έτσι δεν μπορεί να μάθει πολύπλοκες συναρτήσεις.

4) Ποια είναι τα πλεονεκτήματα και τα μειονεκτήματα του MLP σε σχέση με άλλα μοντέλα όπως τα RBF δίκτυα;

Το θετικό με τα MLP είναι ότι μπορούν να μάθουν πολύπλοκες μη γραμμικές σχέσεις μεταξύ δεδομένων εισόδου και εξόδου, αυτό το καταφέρνουν χάρη στην συνάρτηση ενεργοποίησης που εισάγει μη γραμμικότητα στο δίκτυο. Ακόμα τα MLP χρησιμοποιούν τη μέθοδο backpropagation που επιτρέπει την αποδοτική ενημέρωση των βαρών σε όλα τα επίπεδα, βελτιώνοντας την απόδοση. Επίσης είναι αποδοτικά όταν εκπαιδεύονται με μεγάλα σύνολα δεδομένων, ιδίως όταν συνδυάζονται με τεχνικές κανονικοποίησης. Το γεγονός ότι εκπαιδεύονται σε μεγάλα δεδομένα είναι ταυτόχρονα και αρνητικό, γιατί άμα δεν έχουμε μεγάλα σύνολα δεδομένων δεν είναι αποδοτική δυνατή η εκπαίδευση τους. Τα MLP επειδή χρησιμοποιούν backpropagation χρειάζονται περισσότερο χρόνο επεξεργασίας καθώς υπάρχει και κίνδυνος overfitting. Το MLP δεν είναι πάντα η καλύτερη επιλογή μοντέλου, καθώς υπάρχουν πολλά προβλήματα που άλλα μοντέλα υπερτερούν. Για παράδειγμα, το πρόβλημα κυκλικής ταξινόμησης. Έχουμε δύο κατηγορίες σημείων, στην κατηγορία Α τα σημεία είναι συγκεντρωμένα σε έναν κύκλο στο κέντρο. Κατηγορία Β τα σημεία σχηματίζουν ένα δακτύλιο γύρω από τα σημεία της κατηγορίας Α. Ο στόχος του μοντέλου είναι να μάθει τον διαχωρισμό, δηλαδή να

καταλάβει αν ένα νέο σημείο ανήκει στην κατηγορία Α ή Β ανάλογα με την γεωμετρική του τοποθεσία και απόσταση. Για αυτό το παράδειγμα η καλύτερη επιλογή μοντέλου θα ήταν το RBF καθώς η μάθηση γίνεται με βάση την συνάρτηση Gaussian που υπολογίζει την απόσταση από τα κέντρα των δύο περιοχών. Το MLP μαθαίνει μη γραμμικούς διαχωρισμούς, αλλά είναι βασισμένο σε γραμμικούς συνδυασμούς των χαρακτηριστικών. Η προσπάθεια να δημιουργήσει έναν τέλειο κύκλο στον χώρο απαιτεί πολλές παραμορφώσεις, που συχνά οδηγούν σε πολύπλοκα και βαθιά δίκτυα με πολλές παραμέτρους που καθυστερούν και πολλές φορές είναι εσφαλμένα.

Μέρος Γ: Πειραματική Αξιολόγηση & Αναφορά

1) Διαδικασία υλοποίησης:

Η διαδικασία της υλοποίησης του MLP έγινε με βάση το dataset make_moons με 1000 δείγματα, 0,2 θόρυβο και random state που θα ξεκινάει κάθε φορά από 42.

Η αρχιτεκτονική σχεδιάστηκε με:

- 2 Εισόδους
- 2 Κρυφές στρώσεις που αποτελούνται από:
  $1^{η}$ στρώση με 10 νευρώνες
  $2^{η}$ στρώση με 5 νευρώνες
- 1 Έξοδο (binary)
- Learning rate 0.005
- 1001 epochs

Η επιλογή των διαστάσεων έγινε για να επιτραπεί η αναπαράσταση μη γραμμικών σχέσεων ανάμεσα στις δύο κλάσεις. Οι δύο κρυφές στρώσεις επιτρέπουν στο δίκτυο να μάθει πιο περίπλοκα χαρακτηριστικά του χώρου των δεδομένων. Η επιλογή των αριθμών νευρώνων (10, 5) έγιναν μετά από δοκιμές αφού δημιουργήθηκε ο κώδικας με σκοπό την βελτίωση του accuracy. Ο αριθμός ων εποχών (1001) επιλέχτηκε για να διασφαλιστεί ότι το MLP μας θα είχε την ευκαιρία να συγκλίνει προς το ολικό ελάχιστο της συνάρτησης απώλειας, αποφεύγοντας τον εγκλωβισμό σε τοπικά ελάχιστα. (όπως παρατηρείτε από τα διαγράμματα παρακάτω).

Συνάρτηση ενεργοποίησης:

Το δίκτυο χρησιμοποιεί 2 συναρτήσεις ενεργοποίησης:

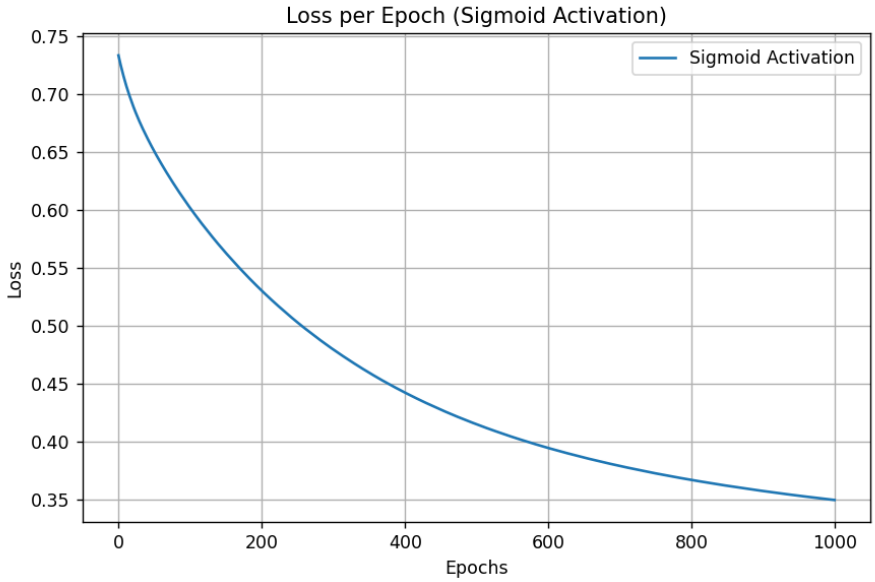
- Sigmoid(1/(1+e^-x)): Χρησιμοποιείται για τη μη γραμμική μετατροπή των τιμών σε κάθε νευρώνα και είναι κατάλληλη για προβλήματα δυαδικής ταξινόμησης.
- Rellu(max(0,x)): Προσφέρει ταχύτερη εκμάθηση σε πολλά προβλήματα και μειώνει το πρόβλημα της εξαφάνισης του gradient σε βάθη δίκτυα.

Το **forward pass** είναι το στάδιο στο οποίο τα δεδομένα εισόδου περνούν από όλα τα επίπεδα του δικτύου. Δηλαδή οι είσοδοι πολλαπλασιάζονται με τα βάρη (weights) και προστίθεται το bias. Μετά Εφαρμόζεται η συνάρτηση ενεργοποίησης (Sigmoid ή ReLU) για να εισαχθεί μη γραμμικότητα. Η διαδικασία αυτή επαναλαμβάνεται μέχρι την τελική έξοδο. Έπειτα χρησιμοποιούμε το **back propagation** που είναι η διαδικασία με την οποία το μοντέλο μαθαίνει. Δηλαδή το μοντέλο υπολογίζει το σφάλμα μεταξύ της πρόβλεψης και της πραγματικής τιμής. Το σφάλμα αυτό μεταφέρεται προς τα πίσω ξεκινώντας από την έξοδο προς τις προηγούμενες στρώσεις (Backwards). Κατά την μετάδοση προς τα πίσω ενημερώνονται τα βάρη με βάση τη συνάρτηση απώλειας (Binary Cross Entropy) και τον ρυθμό εκμάθησης(learning rate). Ουσιαστικά όλη αυτή η διαδικασία γίνεται ώστε το MLP να διορθώσει τα βάρη του ώστε να μειώσει το σφάλμα στις επόμενες επαναλήψεις. Η εκπαίδευση (**training**) του δικτύου γίνεται με την μέθοδο SGD(hybrid, με mini-batches). Σκοπός της είναι να βελτιώσει την γενίκευση του μοντέλου. Αυτό το κατορθώνει χωρίζοντας τα δεδομένα σε μικρά “batches” των 32 δειγμάτων. Έπειτα καλεί και υπολογίζει τις συναρτήσεις Forward pass, το loss(binary_cross_entropy) και backpropagation για την ενημέρωση των βαρών. Τέλος, υπάρχει και η οπτικοποίηση των δεδομένων, που γίνεται με την χρήση 2 διαγραμμάτων για κάθε συνάρτηση ενεργοποίησης. Το πρώτο διάγραμμα μας δείχνει τα όρια απόφασης(decision boundaries) που έχει μάθει το μοντέλο για να διαχωρίζει τα δύο σύνολα δεδομένων (κόκκινο-μπλε) και το δεύτερο διάγραμμα μας δείχνει την εξέλιξη της συνάρτησης απώλειας ανα εποχή(loss per epoch). Εκτός από τα διαγράμματα υπάρχει και ένας πίνακας που εμφανίζει στο terminal κάθε 100 εποχές την απώλεια που υπάρχει και το accuracy.

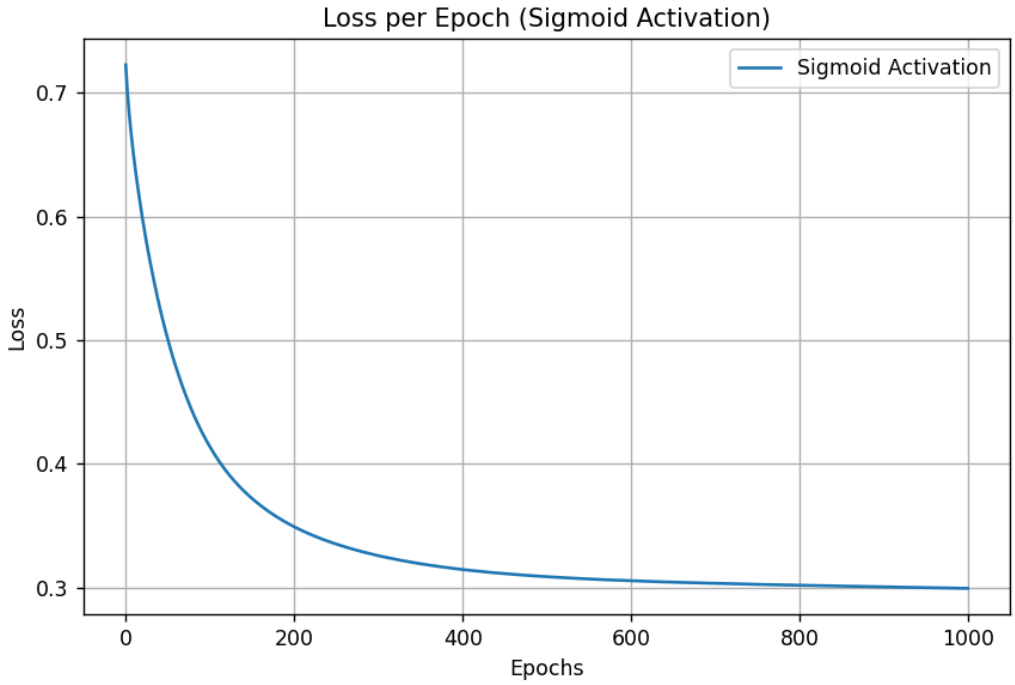
2) Τα πειράματα που εκτελέστηκαν:

Αφού δημιουργήθηκε ο κύριος κώδικας, για την βελτίωση του accuracy του μοντέλου υλοποιήθηκαν τα εξής πειράματα:

- Για να επιτευχθεί το μεγαλύτερο accuracy έγιναν πειραματισμοί με το learning rate. Αρχικά δόθηκε τιμή στο μοντέλο learning rate =0.001. Αυτό έφερε σαν αποτέλεσμα μετά την υλοποίηση του 86.10% accuracy στη Sigmoid συνάρτηση και 92.4% στη Relu συνάρτηση. Αυτά τα αποτελέσματα είναι καλά αλλά όταν παρατηρήσουμε το διάγραμμα loss per epoch (Sigmoid) το μοντέλο βελτιώνει το accuracy σχετικά ¨αργά¨.



Αντίθετα στο δεύτερο πείραμα δόθηκε τιμή στο μοντέλο learning rate =0.005 έφερε ως αποτέλεσμα 87.10% accuracy στη Sigmoid και 97.9% στη Relu. Εκτός από το καλύτερο accuracy παρατηρείτε και στο διάγραμμα loss per epoch (Sigmoid) το accuracy έχει μεγαλύτερη τιμή σε λιγότερες epoch από ότι ήταν πριν, αυτό σημαίνει ότι το μοντέλο είναι πιο ¨γρήγορο¨ και χρειάζεται λιγότερες εποχές για να είναι ακριβείς.



- Με βάση το προηγούμενο πείραμα παρατηρούμε ότι η συνάρτηση Relu έχει πολύ καλύτερο accuracy (97.9%) σε σχέση με την Sigmoid (87.1%). Αυτό οφείλεται γιατί η Sigmoid περιορίζει τις τιμές στο διάστημα (0,1) και όταν το input είναι μεγάλο ή πολύ μικρό, η παράγωγος πλησιάζει το 0. Ενώ η Relu από την άλλη επιτρέπει τις τιμές από 0 έως ∞ , διατηρώντας την παράγωγο στο 1 όσο η είσοδος είναι θετική, επιτρέποντας έτσι τη σωστή διάχυση της πληροφορίας προς τα πίσω.

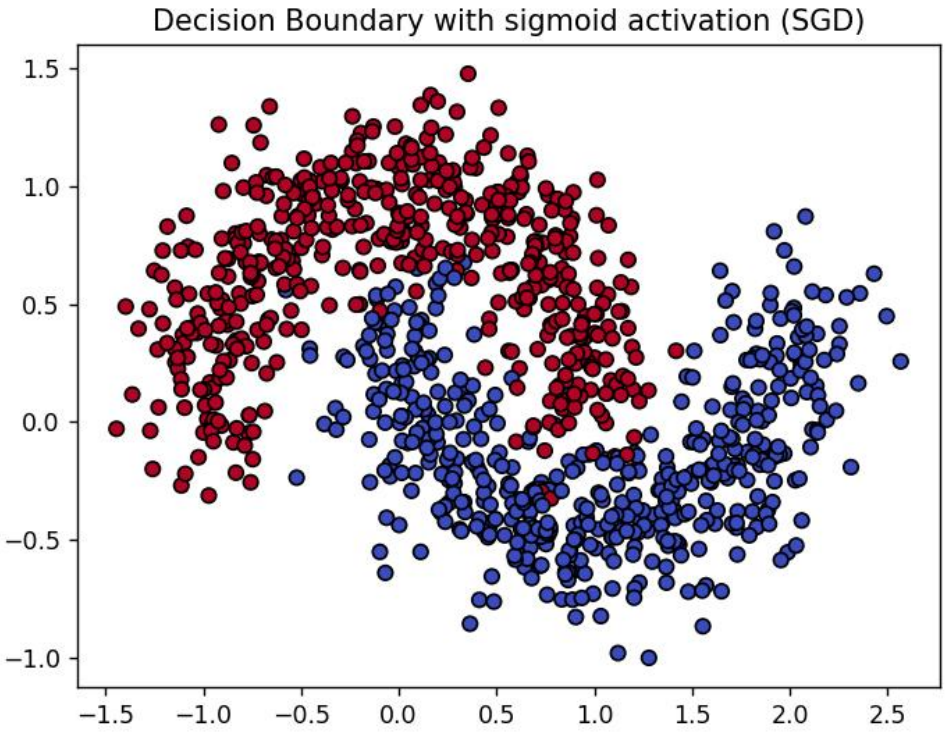
- Στο πλαίσιο των πειραματικών δοκιμών, εκτελέστηκαν δύο ξεχωριστά πειράματα για την αξιολόγηση της επίδραση της κανονικοποίησης στην απόδοση του μοντέλου. Στο πρώτο πείραμα τα δεδομένα εισόδου κανονικοποιήθηκαν με τη χρήση Min-Max Scaling μετατρέποντας όλες τις τιμές σε ένα εύρος μεταξύ 0 και 1. Αυτή η διαδικασία θα έπρεπε να επιτρέπει στα βάρη να προσαρμόζονται ομαλότερα. Όμως τα αποτελέσματα δείχνουν το ακριβώς αντίθετο. Με κανονικοποίηση η συνάρτηση Sigmoid έχει 86,6% accuracy και η συνάρτηση Relu έχει 97,3% accuracy, ενώ όταν εκτελεστεί το μοντέλο χωρίς κανονικοποίηση παρατηρείτε στη Sigmoid 87,10% και 97,8% accuracy. Τα αποτελέσματα αναδεικνύουν ότι παρόλο που η κανονικοποίηση συχνά ενισχύει την εκπαίδευση των μοντέλων, σε ορισμένες περιπτώσεις η απουσία της μπορεί να προσφέρει καλύτερη απόδοση.

3) Οπτικοποίηση αποτελεσμάτων:

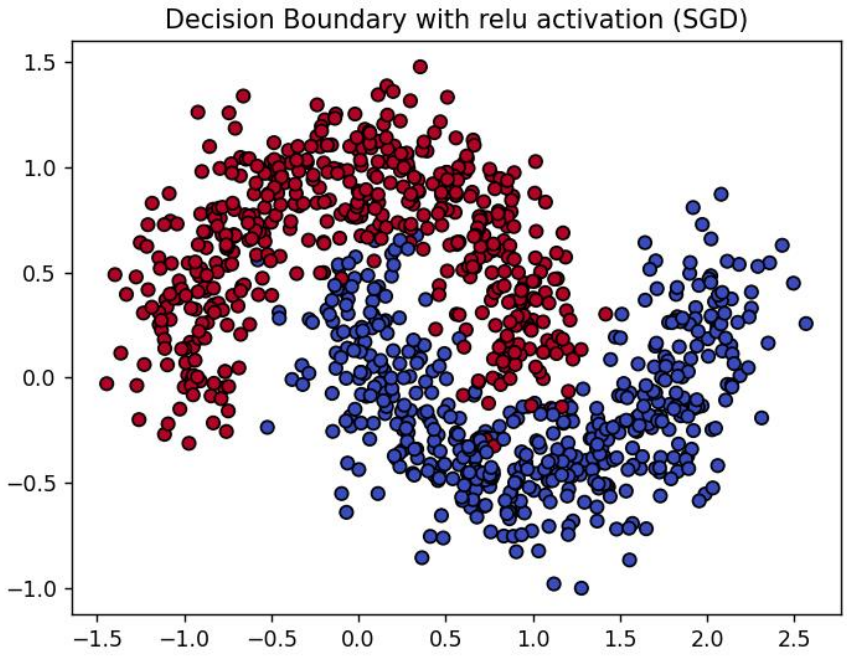
Στο μοντέλο που δημιουργήθηκε υπάρχουν 3 είδη διαγραμμάτων για κάθε συνάρτηση ενεργοποίησης:

- Το πρώτο διάγραμμα είναι το Decision Boundary. Αυτό το γράφημα εμφανίζει τα σύνορα απόφασης που προκύπτουν από το νευρωνικό δίκτυο κατά την ταξινόμηση των δεδομένων. Ουσιαστικά, το δίκτυο προσπαθεί να "χωρίσει" τα δύο διαφορετικά classes (κατηγορίες) των δεδομένων που βρίσκονται στο dataset.

Το διάγραμμα που ακολουθεί χρησιμοποιεί την Sigmoid:

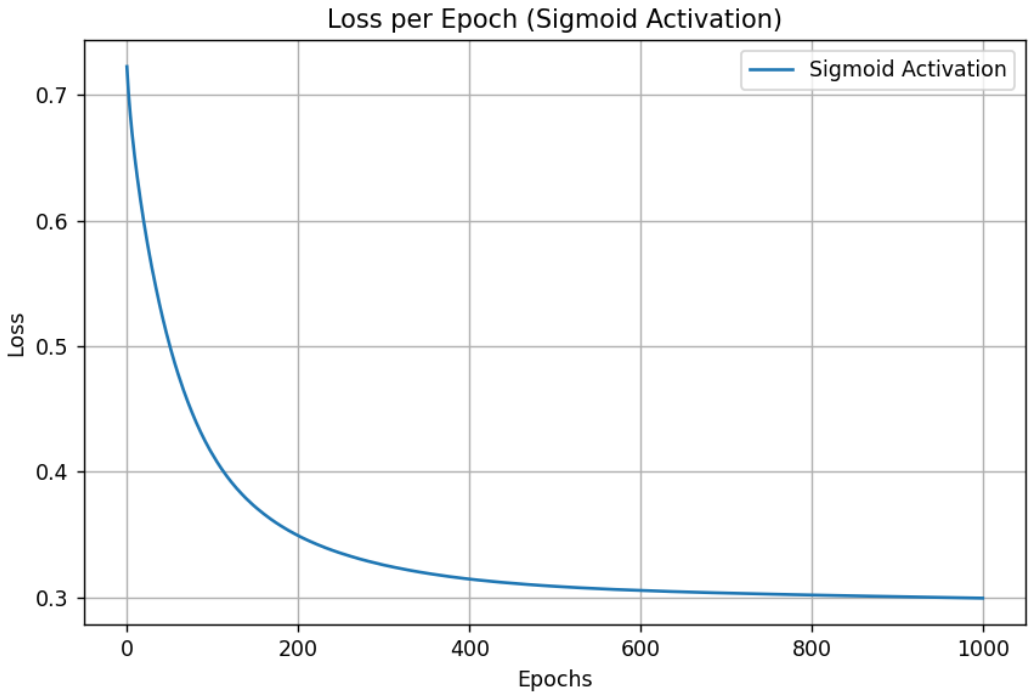


Το διάγραμμα που ακολουθεί χρησιμοποιεί την Relu:

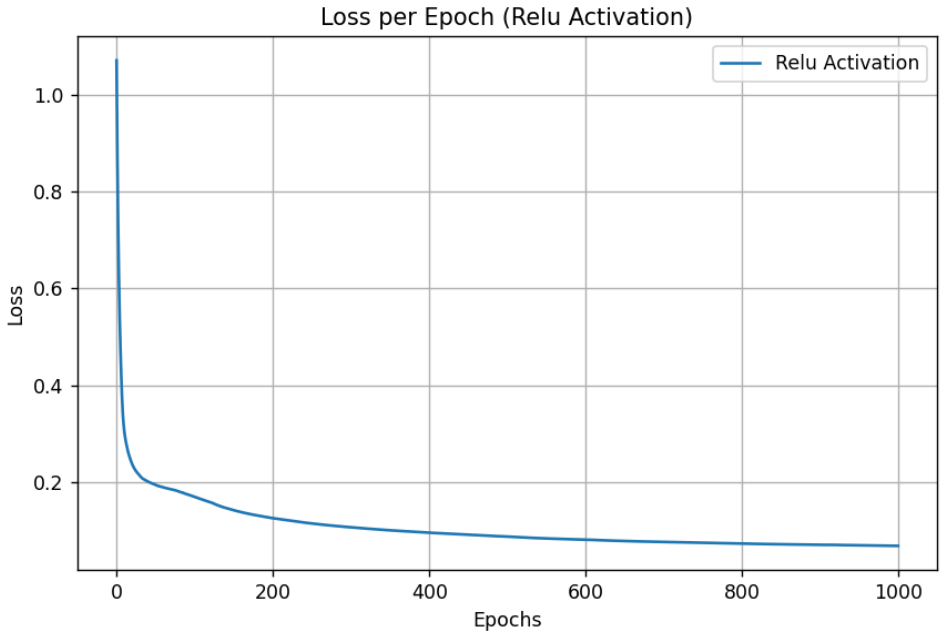


- Το δεύτερο διάγραμμα, εμφανίζει το loss ανά epoch. Δηλαδη, στο διάγραμμα που προκύπτει από τον κώδικα, παρατηρούμε την επίδραση της εκπαίδευσης με SGD και πως αυτή βελτιώνει τη λειτουργία του νευρωνικού δικτύου.

  Το διάγραμμα που ακολουθεί χρησιμοποιεί την Sigmoid:

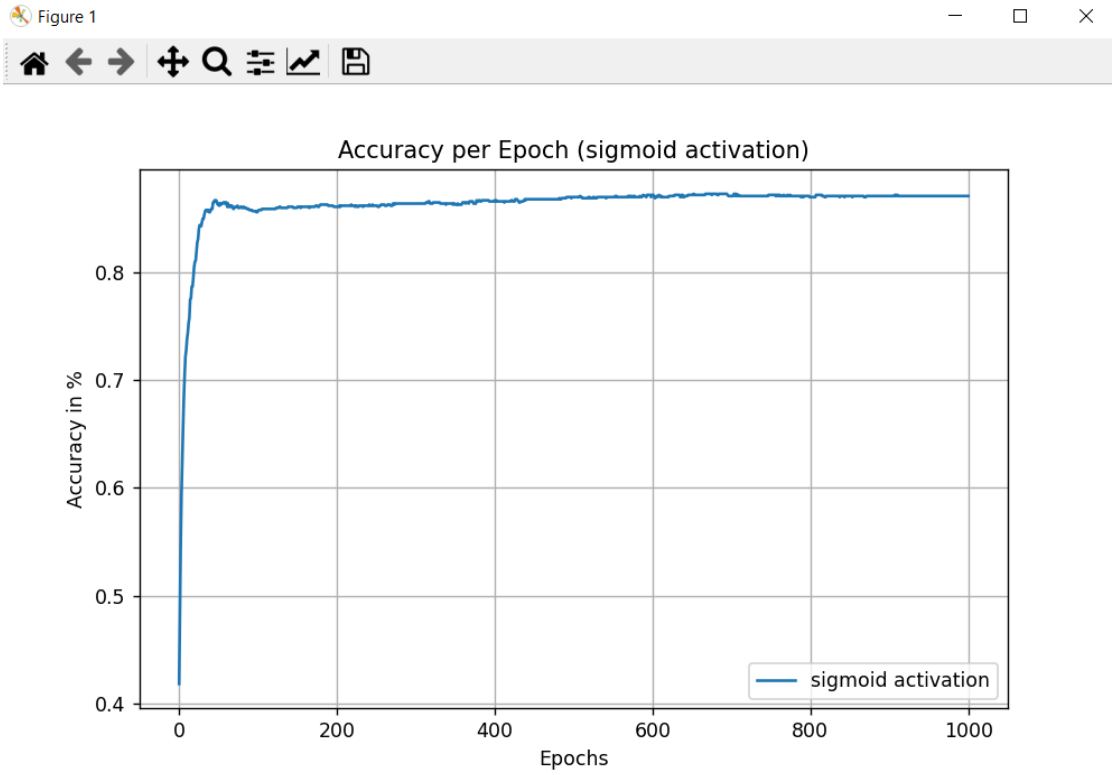


  Το διάγραμμα που ακολουθεί χρησιμοποιεί την Relu:

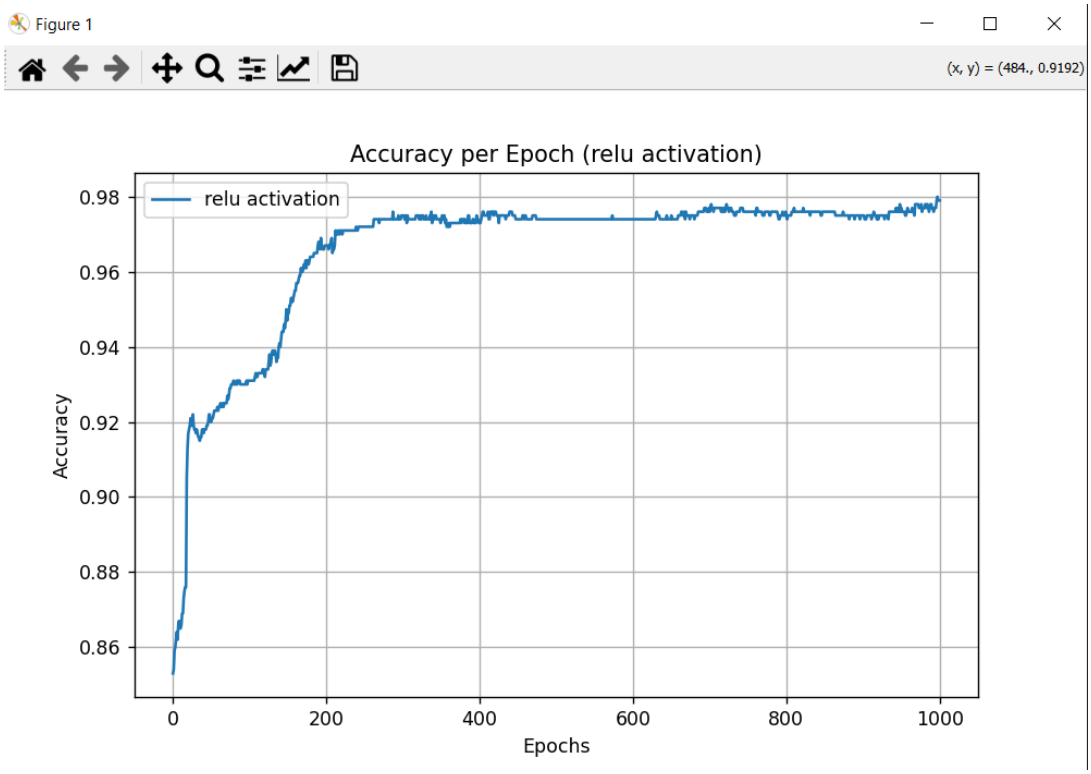


- Το τρίτο γράφημα παρουσιάζει την ακρίβεια του MLP μοντέλου ανά εποχή όταν χρησιμοποιείται η συνάρτηση ενεργοποίησης Sigmoid ή Relu. Παρατηρείται η εξέλιξη της απόδοσης του μοντέλου καθώς εκπαιδεύεται, γεγονός που μας βοηθά να αξιολογήσουμε την αποδοτικότητα της εκπαίδευσης και να συγκρίνουμε διαφορετικές συναρτήσεις ενεργοποίησης. Το γράφημα είναι ιδιαίτερα χρήσιμο για να εντοπίσουμε σημεία σταθεροποίησης ή ενδεχόμενο overfitting. Στο διάγραμμα της Sigmoid παρατηρούμε ότι κατά τις πρώτες 50 εποχές, η ακρίβεια αυξάνεται ραγδαία, γεγονός που δείχνει ότι το μοντέλο αρχίζει να μαθαίνει βασικά μοτίβα από τα δεδομένα. Από εκεί και πέρα, η ακρίβεια σταθεροποιείται προοδευτικά και μετά τις 300 εποχές παρουσιάζει πολύ μικρές αυξομειώσεις, χωρίς σημαντική βελτίωση, το μοντέλο φτάνει σε ένα επίπεδο κορεσμού ακρίβειας, το οποίο παραμένει σταθερό μέχρι και την 1000 εποχή. Οπότε η περεταίρω εκπαίδευση μετά την 300 εποχή δεν προσφέρει ουσιαστικά οφέλη και απλώς αυξάνει το χρόνο υπολογισμού. Τα ίδια φαινόμενα παρατηρούμε και στην 400 εποχή από την συνάρτηση Relu.

Το διάγραμμα που ακολουθεί χρησιμοποιεί την Sigmoid:



Το διάγραμμα που ακολουθεί χρησιμοποιεί την Relu:



4) Συμπεράσματα – Τεκμηρίωση Ορθότητας:

Η εκπαίδευση του MLP πάνω στο dataset make moons ανέδειξε τη δύναμη και την ευελιξία των (MLP) στην επίλυση προβλημάτων ταξινόμησης με μη γραμμικές σχέσεις. Με μια αρχιτεκτονική που περιλάμβανε δύο κρυφές στρώσεις, η πρώτη με 10 νευρώνες και η δεύτερη με 5 νευρώνες, το δίκτυο κατόρθωσε να μάθει τα πολύπλοκα μοτίβα που υπάρχουν στο σύνολο δεδομένων, επιτυγχάνοντας πολύ υψηλή ακρίβεια τόσο με τη συνάρτηση ενεργοποίησης **ReLU** (97.9%) όσο και με τη **Sigmoid** (87.1%). Επιπλέων παρατηρήθηκε ότι η κανονικοποίηση των δεδομένων μειώνει το accuracy και για αυτό τον λόγο δεν χρησιμοποιείτε σε αυτό το μοντέλο. Αν μου δινόταν η ευκαιρία να βελτιώσω κάτι μελλοντικά θα ήταν η κανονικοποίηση των δεδομένων. Γιατί υπό φυσιολογικές συνθήκες θα έπρεπε με την κανονικοποίηση των δεδομένων να βελτιώνεται το accuracy και όχι να χειροτερεύει.