

Αναφορά Παράδοσης Άσκησης 1 στην Υπολογιστική Νοημοσύνη

Ομάδα:

- Γεωργούλας Δημοσθένης AM 4039
- Στεργίου Βασίλειος AM 4300

Άσκηση 1

Στην παρούσα άσκηση ζητείται η υλοποίηση ενός MLP δικτύου, αρχικά με δύο κρυμμένα επίπεδα και έπειτα με τρία.

Το MLP δομείται από τρεις κλάσεις:

- i. Την κλάση Main, στην οποία στηρίζεται η αρχιτεκτονική του MLP και από την οποία εξάγονται αποτελέσματα που αφορούν το δίκτυο
- ii. Την κλάση Layer, η οποία χρησιμοποιείται για την αναπαράσταση των επιπέδων του δικτύου με τη χρήση νευρώνων που υλοποιούνται στην κλάση Neuron.
- iii. Την κλάση Neuron η οποία αντιπροσωπεύει του νευρώνες του δικτύου.

Στο σημείο αυτό θα γίνει σταδιακή αναφορά στις επιμέρους φάσεις κατασκευής του MLP με την σειρά που ζητούνται από την άσκηση:

1. Στην κλάση Main γίνεται ο καθορισμός των παραμέτρων που απαιτούνται για την λειτουργία του δικτύου, όπως
 - αριθμός εισόδων (d),
 - αριθμός κατηγοριών (k),
 - αριθμός νευρώνων στο πρώτο κρυμμένο επίπεδο (h1),
 - αριθμός νευρώνων στο δεύτερο κρυμμένο επίπεδο (h2),
 - αριθμός νευρώνων στο τρίτο κρυμμένο (h3)
 - είδος συνάρτησης ενεργοποίησης (activation_function) για τα κρυμμένα επίπεδα.

Επίσης έχουν προστεθεί και επιμέρους παράμετροι, όπως:

- αριθμός εποχών (epoch)
 - αριθμός ομάδων (batches_number)
 - κατώφλι ελέγχου της διαφοράς του σφάλματος (threshold)
 - ρυθμός μάθησης (l_r)
2. Στην συνέχεια φορτώνονται τα δεδομένα των συνόλων εκπαίδευσης και ελέγχου, η δημιουργία των οποίων έχει πραγματοποιηθεί σε γλώσσα Python.
 3. Σχετικά με την αρχιτεκτονική του MLP, έχουν δημιουργηθεί τα επίπεδα που ζητούνται, τα οποία διακρίνονται από τα εξής χαρακτηριστικά:
 - Το επίπεδο εισόδου (Input_Layer) έχει d εισόδους και d εξόδους.
 - Το πρώτο κρυμμένο επίπεδο έχει d εισόδους, όσες δηλαδή και οι εξοδοι του επιπέδου εισόδου και h1 εξόδους.
 - Το δεύτερο κρυμμένο επίπεδο έχει h1 εισόδους, όσες δηλαδή και οι εξοδοι του πρώτου κρυμμένου επιπέδου και h2 εξόδους.
 - Το τρίτο κρυμμένο επίπεδο έχει h2 εισόδους, όσες δηλαδή και οι εξοδοι του δεύτερου κρυμμένου επιπέδου και h3 εξόδους.
 - Το επίπεδο εξόδου έχει h3 εισόδους, όσες δηλαδή και οι εξοδοι του τρίτου κρυμμένου επιπέδου και k εξόδους.
 - Κατά την δημιουργία των επιπέδων, αρχικοποιούνται οι νευρώνες κάθε επιπέδου και τα βάρη/πολώσεις του κάθε νευρώνα με την χρήση της συνάρτησης `init_weights()` στην κλάση Neuron.

4. Η συνάρτηση `forward_pass()` χρησιμοποιείται σε κάθε επίπεδο, εκτός του αρχικού, για την μεταφορά των εξόδων κάθε νευρώνα του προηγούμενου επιπέδου στο τρέχον επίπεδο. Ο υπολογισμός της εξόδου κάθε νευρώνα γίνεται ακολουθώντας την εξής μέθοδο.

$$u_i^{(h)} = \sum_{j=0}^{d_{h-1}} w_{ij}^{(h)} y_j^{(h-1)} \iff u_i^{(h)} = \sum_{j=1}^{d_{h-1}} w_{ij}^{(h)} y_j^{(h-1)} + w_{i0}^{(h)}, \quad i=1, \dots, d_h$$

$$y_i^{(h)} = g_h(u_i^{(h)}) \quad i=1, \dots, d_h, \quad y_0^{(h)} = 1$$

Στο επίπεδο εξόδου, η συνάρτηση `forward_pass()` υπολογίζει την έξοδο του δικτύου MLP.

5. Ο υπολογισμός του back propagation αποτελείται από δύο μέρη:
- Υπολογισμός δ στα επίπεδα του MLP
 - Υπολογισμός των μερικών παραγώγων για τα βάρη κάθε νευρώνα

Για τον υπολογισμό του δ , έχουν δημιουργηθεί οι συναρτήσεις `back_prop_final()` για τον υπολογισμό του δ στο επίπεδο εξόδου και `back_prop()` για τα κρυμμένα επίπεδα.

– Νευρώνες εξόδου (επίπεδο $H+1$) (συν. ενεργοποίησης g_{H+1})

$$\delta_i^{(H+1)} = g'_{H+1}(u_i^{(H+1)})(o_i - t_{ni}), \quad i=1, \dots, p$$

$\delta_i^{(H+1)} = (o_i - t_{ni}), \quad i=1, \dots, p$ (γραμμική συν. ενεργοποίησης)

$\delta_i^{(H+1)} = o_i(1-o_i)(o_i - t_{ni}), \quad i=1, \dots, p$ (λογιστική συν. ενεργοποίησης)

– Νευρώνες κρυμμένων επιπέδων: για επίπεδο $h=H, \dots, 1$ (συν. ενεργοποίησης g_h)

$$\delta_i^{(h)} = g'_h(u_i^{(h)}) \sum_{j=1}^{d_{h+1}} w_{ji}^{(h+1)} \delta_j^{(h+1)}, \quad i=1, \dots, d_h$$

Για τον υπολογισμό των μερικών παραγώγων για τα βάρη κάθε νευρώνα χρησιμοποιείται η συνάρτηση `calculate_de()` σε κάθε επίπεδο, η οποία χρησιμοποιεί τη συνάρτηση `calculate_grad()` για τον υπολογισμό των μερικών παραγώγων σε επίπεδο νευρώνα ως εξής

- Μερική παράγωγος βάρους σύνδεσης:
σφάλμα προορισμού \times έξοδος πηγής
- $$\frac{\partial E^n}{\partial w_{ij}^{(h)}} = \delta_i^{(h)} y_j^{(h-1)}$$
- Μερική παράγωγος πόλωσης = σφάλμα του νευρώνα

$$\frac{\partial E^n}{\partial w_{i0}^{(h)}} = \delta_i^{(h)}$$

6. Ο αλγόριθμος που υλοποιεί το MLP τρέχει για τουλάχιστον ένα δεδομένο αριθμό εποχών που ορίζεται από τον χρήστη στην αρχή του προγράμματος και συγκλίνει όταν έχει ξεπεραστεί ο αριθμός των εποχών και η διαφορά σφάλματος μεταξύ δύο εποχών είναι μικρότερη από κάποιο κατώφλι που επίσης έχει οριστεί από τον χρήστη.

Η ενημέρωση των βαρών του δικτύου πραγματοποιείται ανά ομάδες (batches) και στο τέλος εποχής υπολογίζεται το σφάλμα και γίνεται έλεγχος τερματισμού του αλγορίθμου.

7. Μόλις τερματίσει ο αλγόριθμος, καλείται η συνάρτηση `generalization()`, η οποία τυπώνει την γενικευτική ικανότητα του δικτύου, με είσοδο ένα σύνολο παραδειγμάτων ελέγχου.

Σημείωση: Για πιο λεπτομερή περιγραφή των λειτουργιών του αλγορίθμου, έχουν προστεθεί αντίστοιχα σχόλια στον κώδικα της κάθε κλάσης.

Για την εκτέλεση του προγράμματος θα χρειαστούν οι εντολές:

- `javac *.java`
- `java Main1` (για 2 κρυμμένα επίπεδα) ή `java Main2` (για 3 κρυμμένα επίπεδα)

Με χρήση των παραπάνω δύο εντολών τυπώνονται αποτελέσματα δηλαδή το σφάλμα ανά εποχή, και η αντίστοιχη εποχή και στο τέλος τυπώνεται η γενικευτική ικανότητα του MLP.

Συμπεριλαμβάνεται αρχείο `python` το οποίο αυτοματοποιεί όλη την διαδικασία δηλαδή κάνει `compile` τα αρχεία, τρέχει την `Main1` ή `Main2` και τυπώνει τα αποτελέσματα του terminal καθώς και τα `plots`.

- `python` ή `python3 plot_data1.py` (Για τα 2 κρυμμένα επίπεδα)
- `python` ή `python3 plot_data2.py` (Για τα 3 κρυμμένα επίπεδα)

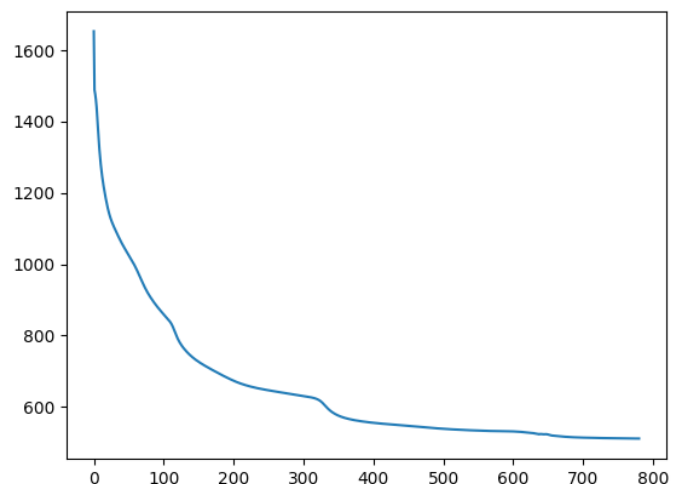
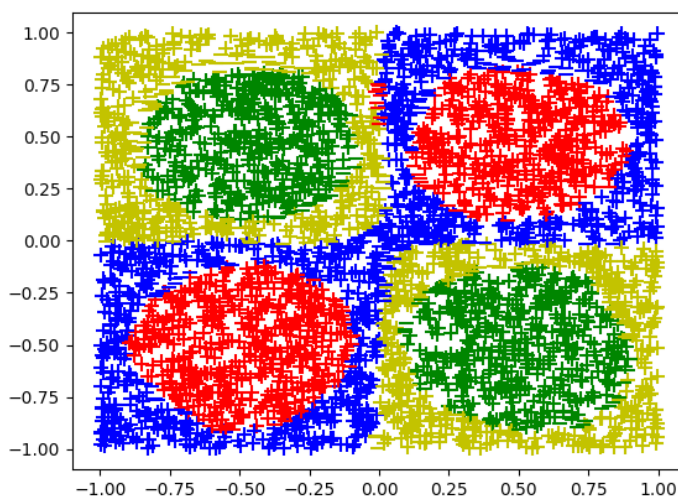
Στις παρακάτω εικόνες έχει εκτελεστεί ο αλγόριθμος για διαφορετικές τιμές των παραμέτρων:

- `h1`
- `h2`
- `batches_number`
- `l_r`
- `activation_function`

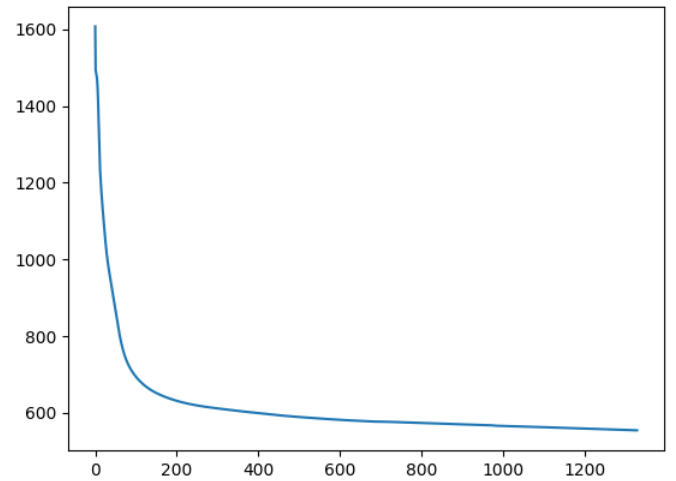
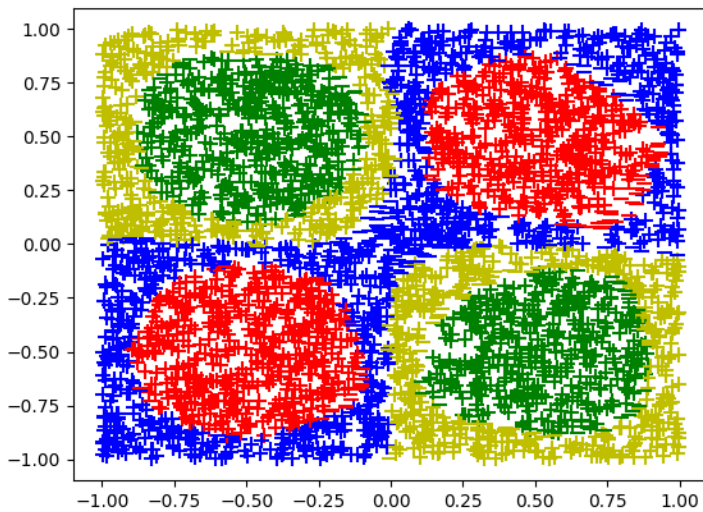
καθώς και για διαφορετικό αριθμό κρυμμένων επιπέδων.

Αρχικά παρατίθενται τα αποτελέσματα του δικτύου με εφαρμογή της Υπερβολική Εφαπτομένης (Tanh) ως συνάρτηση ενεργοποίησης με σταθερό ρυθμό μάθησης $\rightarrow l_r = 0.005$, διαφορά σφάλματος μεταξύ δύο εποχών (threshold) $\rightarrow threshold = 0.01$ και 2 κρυμμένα επίπεδα.

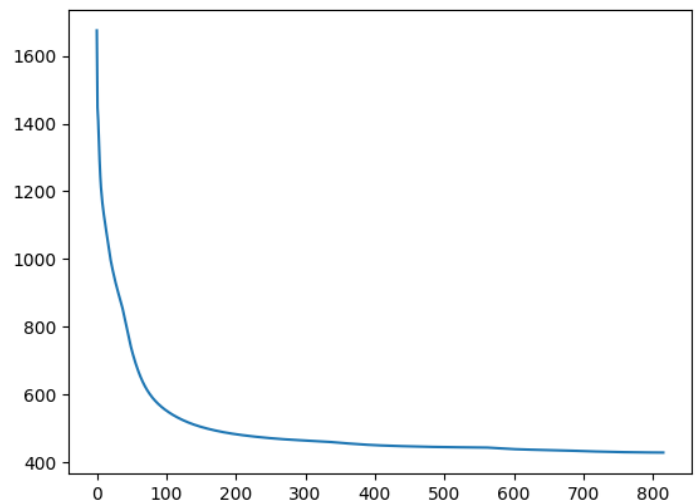
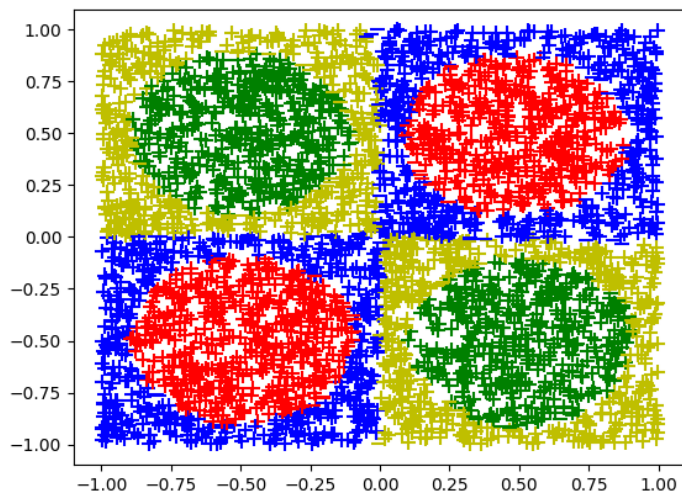
Με 7 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 40` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.



Με 12 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 40` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.

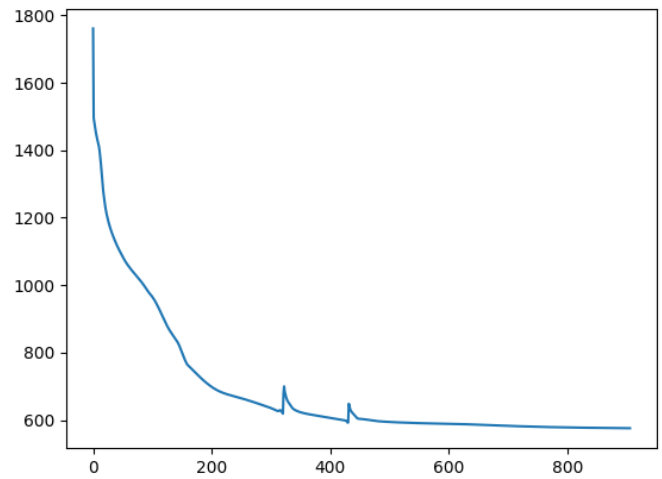
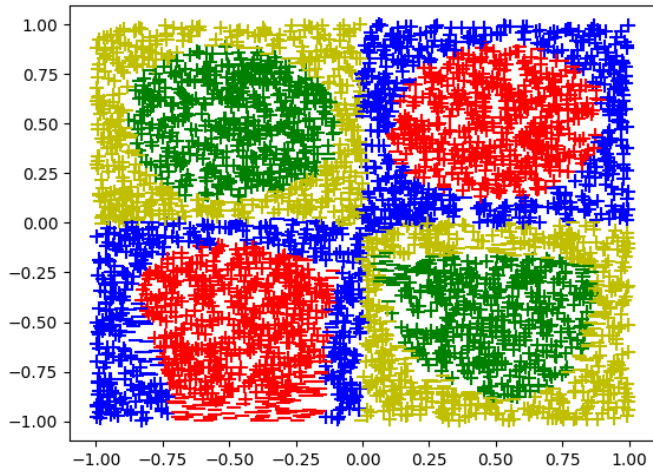


Με 17 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 40` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.

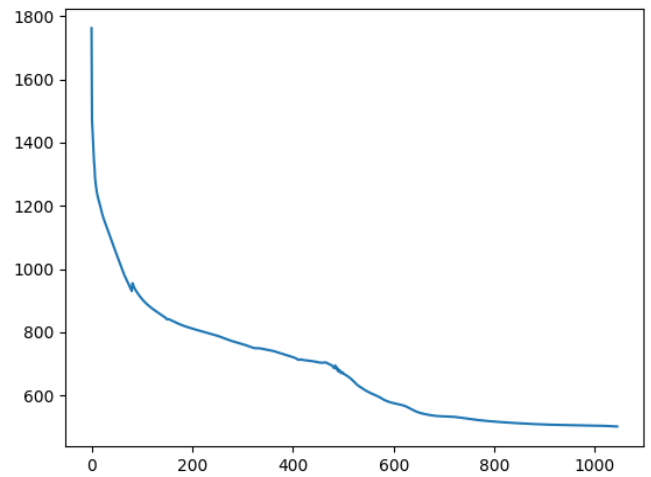
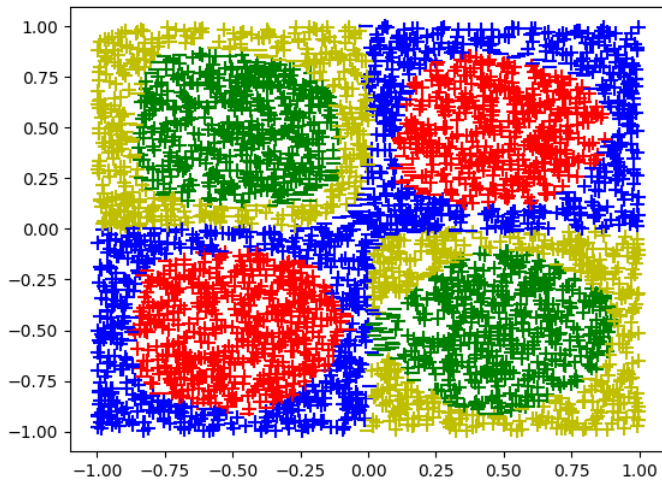


Με βάση τα αποτελέσματα αυτά, παρατηρούμε ότι το δίκτυο εκπαιδεύεται πιο αποτελεσματικά όσο αυξάνεται ο αριθμός των νευρώνων και επέρχεται σύγκλιση του αλγορίθμου σε μικρότερο αριθμό εποχών. Επίσης, το σφάλμα του δικτύου μειώνεται πιο “ομαλά”, καθώς αυξάνεται ο αριθμός των νευρώνων.

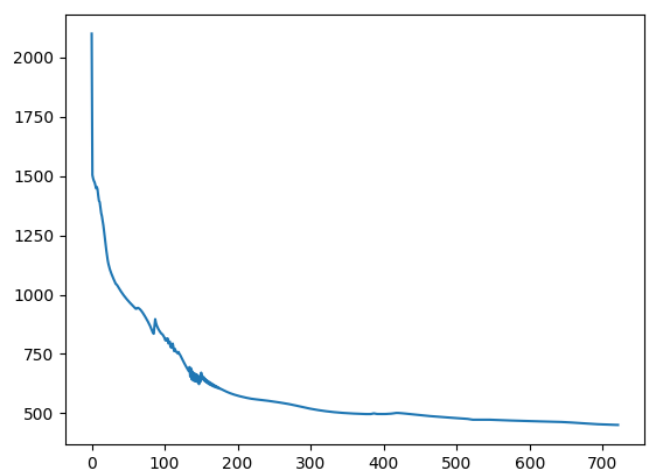
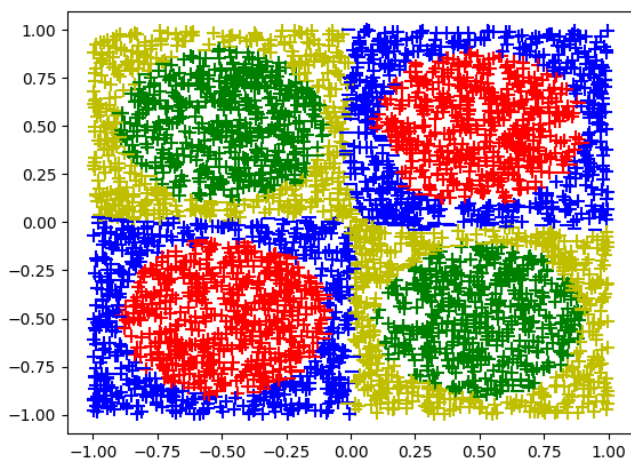
Με 7 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 400` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.



Με 12 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 400` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.



Με 17 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 400` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.

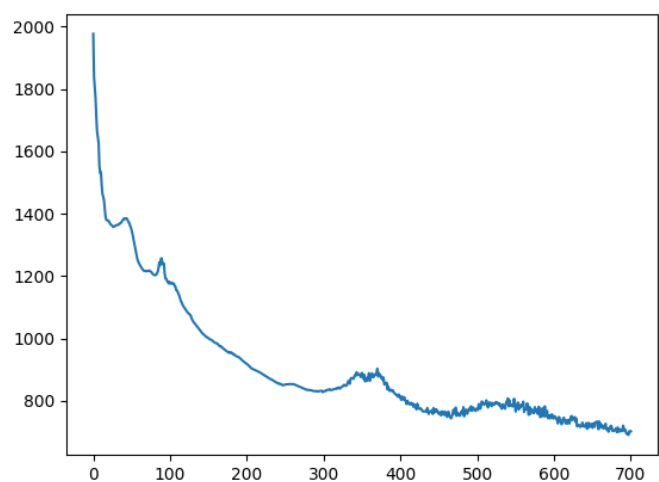
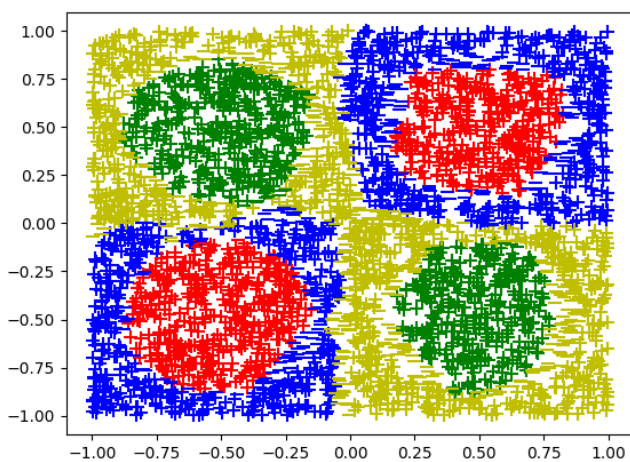


Στις περιπτώσεις αυτές, παρατηρούμε ότι, με την αύξηση του `batches_number` από 40 σε 400, το σφάλμα αυξομειώνεται στο πέρας των εποχών και σε σύγκριση με τα αποτελέσματα για `batches_number` = 40, το τελικό σφάλμα φαίνεται να είναι μεγαλύτερο για τον αντίστοιχο αριθμό νευρώνων. Η πορεία της εκπαίδευσης του δικτύου είναι πιο αισθητή για `batches_number` = 400, καθώς όσο αυξάνουμε τον αριθμό των νευρώνων, τόσο πιο εμφανής είναι η διαφορά στην εκπαίδευση μεταξύ δύο πειραμάτων. Για `batches_number` = 40, η διαφορά στην εκπαίδευση δεν είναι το ίδιο αισθητή.

Στο σημείο αυτό παρατίθενται τα αποτελέσματα του δικτύου με εφαρμογή της ReLu ως συνάρτηση ενεργοποίησης με σταθερό τις εξής τροποποιήσεις στις παραμέτρους:

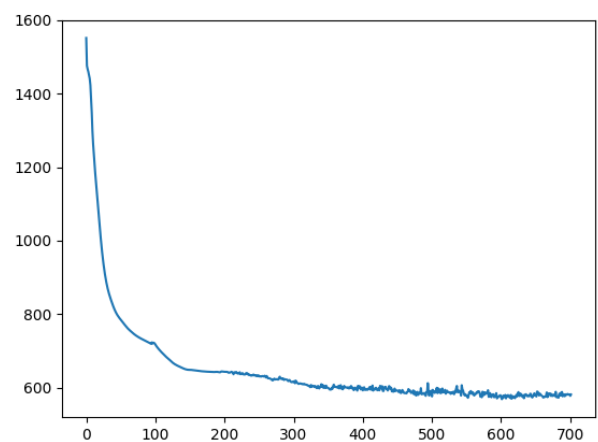
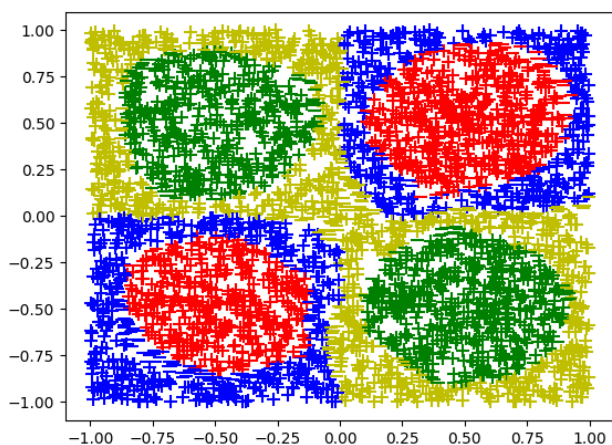
- ρυθμό μάθησης $\rightarrow l_r = 0.0005$
- διαφορά σφάλματος μεταξύ δύο εποχών (threshold) $\rightarrow \text{threshold} = 0.01$
- νευρώνες στα κρυμμένα επίπεδα ($h1, h2$) $\rightarrow 17$
- `batches_number` $\rightarrow 40$

Όσον αφορά τα αποτελέσματα της ReLu, δεν έχουν προστεθεί περισσότερα αποτελέσματα λόγω της δυσκολίας εύρεσης παραμέτρων που να δίνουν ικανοποιητικά αποτελέσματα.

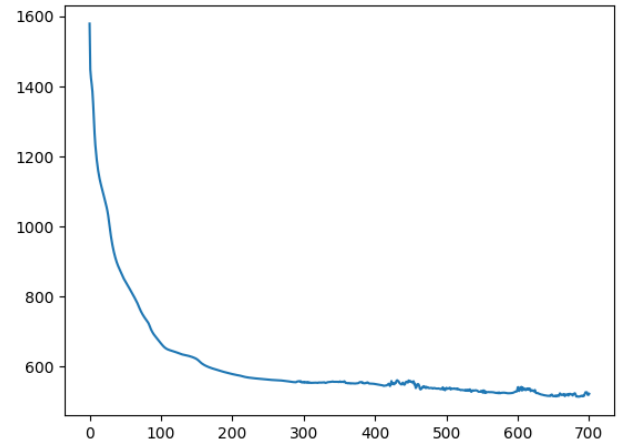
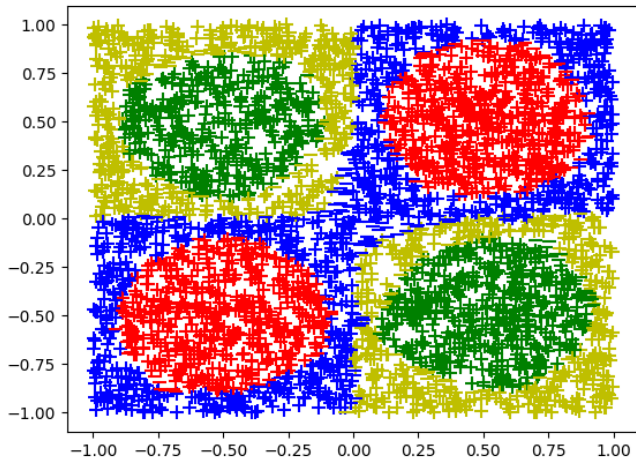


Τα παρακάτω αποτελέσματα του δικτύου έχουν προκύψει με εφαρμογή της Υπερβολικής Εφαπτομένης (Tanh) ως συνάρτηση ενεργοποίησης με σταθερό ρυθμό μάθησης $\rightarrow l_r = 0.005$, διαφορά σφάλματος μεταξύ δύο εποχών (threshold) $\rightarrow \text{threshold} = 0.01$ και 3 κρυμμένα επίπεδα.

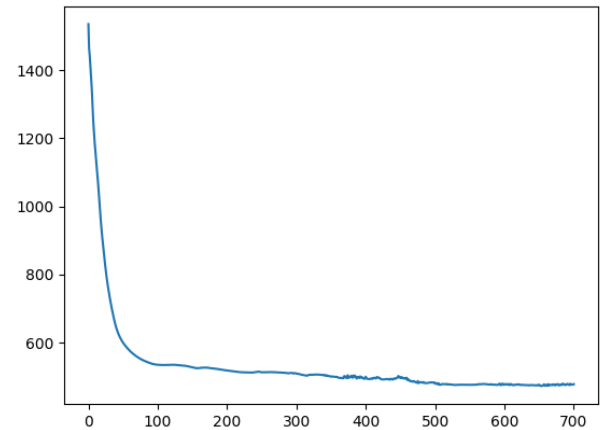
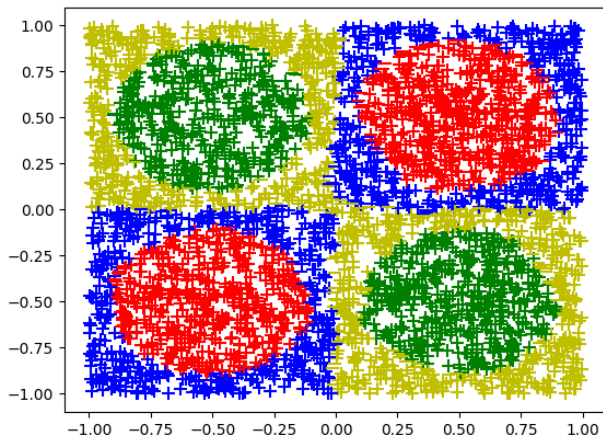
Με 7 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number` = 40 το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.



Με 12 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 40` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.

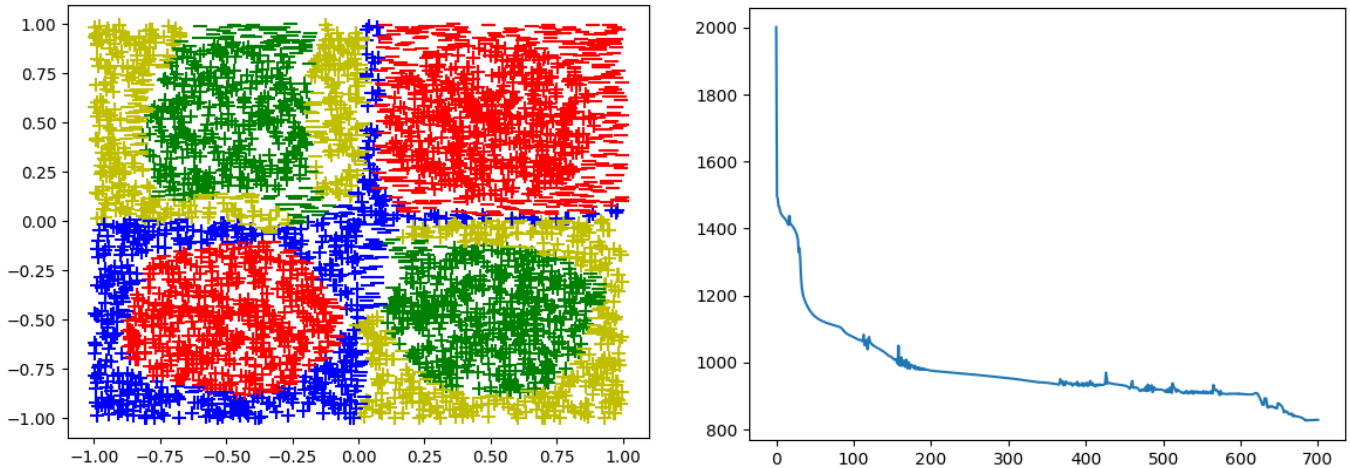


Με 17 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 40` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.

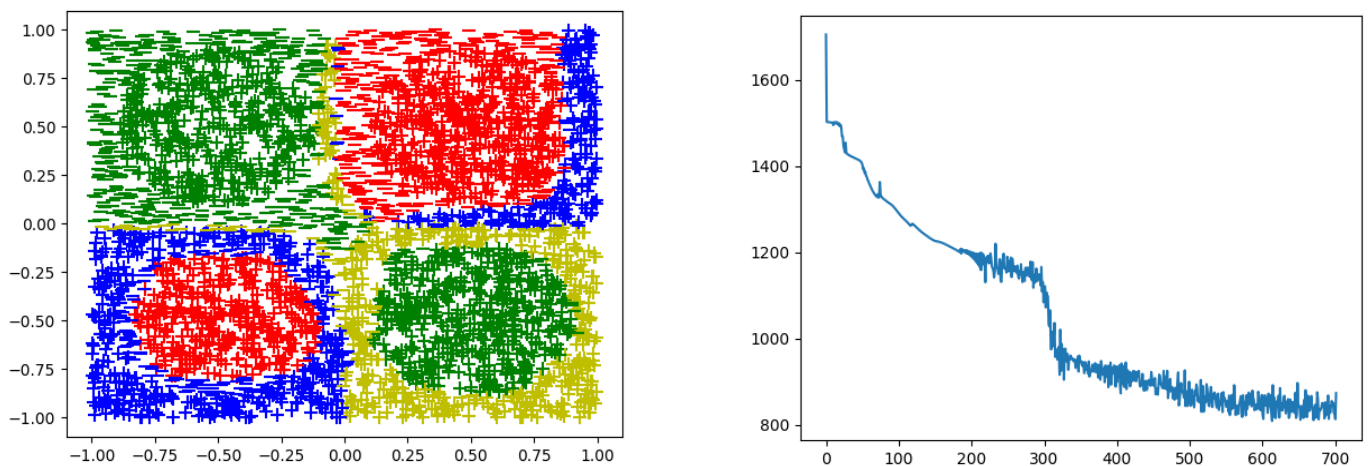


Με την προσθήκη ενός ακόμα κρυμμένου κρυμμένου επιπέδου, το δίκτυο επέρχεται πιο γρήγορα σε σύγκλιση, παρατηρούνται λιγότερες αυξομειώσεις (spikes) στο σφάλμα και το ίδιο το σφάλμα μειώνεται, καθώς αυξάνονται οι νευρώνες στα κρυμμένα επίπεδα.

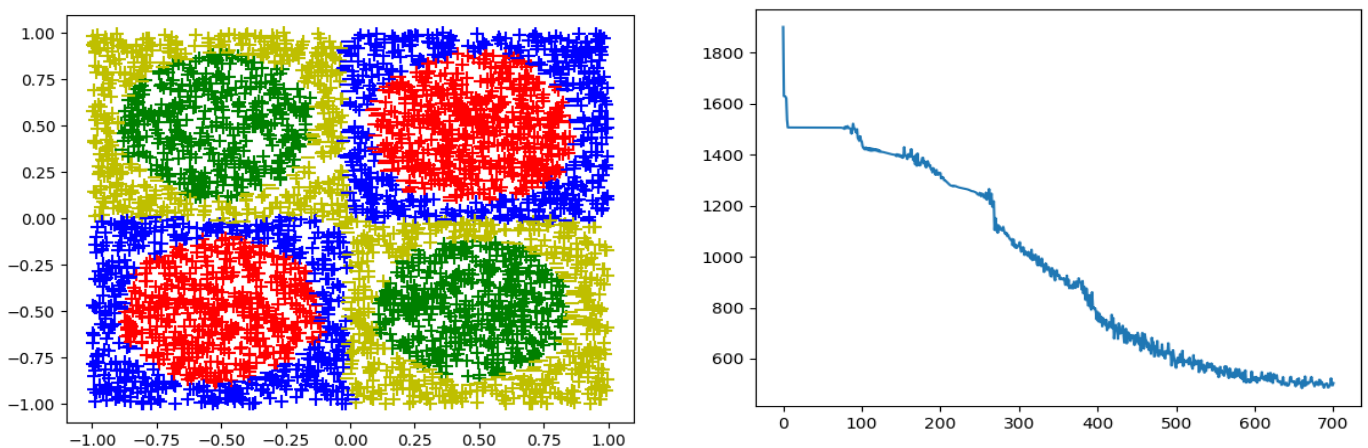
Με 7 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 400` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.



Με 12 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 400` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.

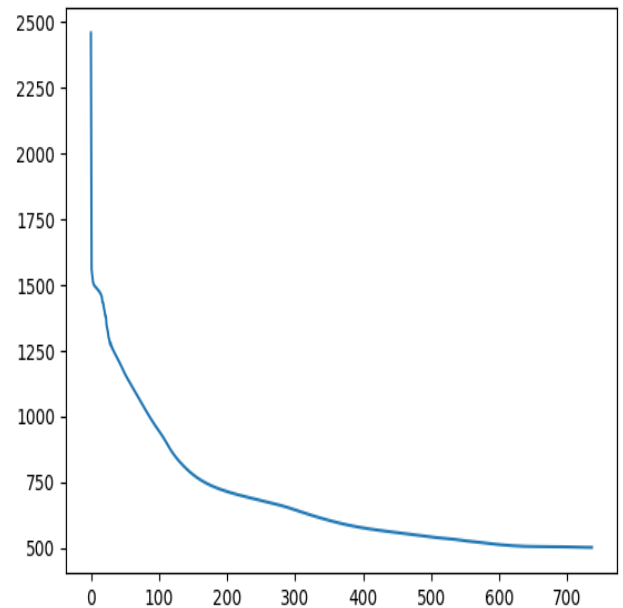
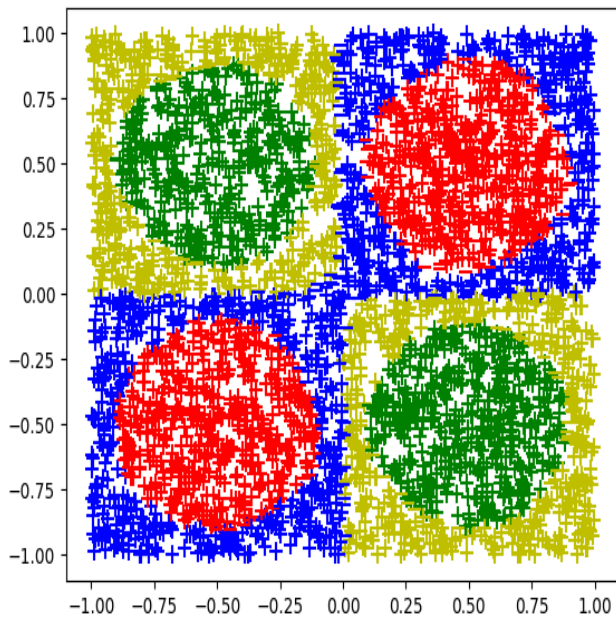


Με 17 νευρώνες σε κάθε κρυμμένο επίπεδο και `batches_number = 400` το δίκτυο δίνει τα εξής αποτελέσματα όσον αφορά την εκπαίδευσή του και την πορεία του σφάλματος.



Με την αύξηση του `batches_number` από 40 σε 400, παρατηρούνται περισσότερες αυξομειώσεις (spikes) στο σφάλμα και η εκπαίδευση του δικτύου γίνεται με πιο αργό ρυθμό.

Μειώνοντας τον ρυθμό μάθησης από 0.005 σε 0.001 στην περίπτωση των 17 νευρώνων ανά κρυμμένο επίπεδο, παρατηρείται μείωση του σφάλματος χωρίς αυξομειώσεις (spikes) και τα αποτελέσματα είναι παρόμοια με την περίπτωση των δύο κρυμμένων επιπέδων.



Συμπερασματικά, η αύξηση των κρυμμένων επιπέδων δεν συνεπάγεται απαραίτητα καλύτερη εκπαίδευση του δικτύου.

Στην περίπτωση που χρησιμοποιείται η Υπερβολική Εφαπτομένη (Tanh)

ως συνάρτηση ενεργοποίησης, τα αποτελέσματα από την εκπαίδευση με δύο κρυμμένα επίπεδα είναι παρόμοια με αυτά που λαμβάνουμε με την χρήση τριών κρυμμένων επιπέδων, μεταβάλλοντας κατάλληλα παραμέτρους όπως ο ρυθμός μάθησης.

Σε αντίθεση με την Υπερβολική Εφαπτομένη (Tanh), η ReLu, λόγω της ευαισθησίας της στην τυχαιότητα των βαρών, καθώς και στην μεταβολή αυτών, δεν επιφέρει ίδια αποτελέσματα με χρήση σταθερών παραμέτρων και διαφορετικού πλήθους νευρώνων και κρυμμένων επιπέδων.

Για το δοθέν πρόβλημα, η χρήση της ReLu δεν επιφέρει καλύτερα αποτελέσματα από αυτά της Υπερβολικής Εφαπτομένης (Tanh). Ομοίως, η χρήση επιπλέον επιπέδων κρυμμένων επιπέδων δεν επιφέρει καλύτερα αποτελέσματα από αυτά των δύο κρυμμένων επιπέδων.

Γενικότερα, η χρήση όλο και περισσότερων νευρώνων στα κρυμμένα επίπεδα επιφέρει γρηγορότερη σύγκλιση του αλγορίθμου.

Η εμφάνιση των αυξομειώσεων (spikes) παρατηρείται όταν αυξάνονται τα batches με χρήση της Tanh. Με τη χρήση της ReLu, παρατηρείται εξίσου η εμφάνισή τους, με την διαφορά ότι γίνονται πιο έντονα καθώς αυξάνονται τα batches.

Στον ακόλουθο πίνακα παρουσιάζεται η γενικευτική ικανότητα του δικτύου για διαφορετικές παραμέτρους του για δύο κρυμμένα επίπεδα.
 Στην περίπτωση της Υπερβολικής Εφαπτομένης, η γενικευτική ικανότητα του δικτύου είναι μεγαλύτερη όταν οι νευρώνες του πρώτου κρυμμένου επιπέδου είναι περισσότεροι από του δεύτερου.
 Επίσης, μεγαλύτερη γενικευτική ικανότητα παρουσιάζεται όταν αυξάνεται το πλήθος των νευρώνων και στα δύο επίπεδα.
 Όσον αφορά την ReLu, παρουσιάζει μικρότερη γενικευτική ικανότητα συγκριτικά με την Tanh, ανεξάρτητα από τις παραμέτρους

F	LR	H1	H2	H3	B		G
Tanh	0.005	7	7		40		0.934
Tanh	0.005	7	15		40		0.94775
Tanh	0.005	15	7		40		0.97475
Tanh	0.005	20	5		40		0.97825
Tanh	0.005	20	20		40		0.97025
Tanh	0.005	7	7		400		0.802
Tanh	0.005	7	15		400		0.889
Tanh	0.005	15	7		400		0.9385
Tanh	0.005	20	5		400		0.802
Tanh	0.005	20	20		400		0.97675
Relu	0.0005	7	7		40		0.80325
Relu	0.0005	7	15		40		0.886
Relu	0.0005	15	7		40		0.69825
Relu	0.0005	20	5		40		0.63925
Relu	0.0005	20	20		40		0.69825
Relu	0.0005	7	7		400		0.38825
Relu	0.0005	7	15		400		0.347
Relu	0.0005	15	7		400		0.24225
Relu	0.0005	20	5		400		0.336
Relu	0.0005	20	20		400		0.2485

Στον ακόλουθο πίνακα παρουσιάζεται η γενικευτική ικανότητα του δικτύου για διαφορετικές παραμέτρους του για τρία κρυμμένα επίπεδα.

Όπως και στην περίπτωση των δύο κρυμμένων επιπέδων, η Tanh παρουσιάζει μεγαλύτερη γενικευτική ικανότητα σε σχέση με την ReLu για τις διάφορες τιμές των παραμέτρων.

Όταν το πλήθος των νευρώνων στα H1,H2 είναι μεγαλύτερο από αυτό του H3 παρατηρείται βελτίωση στην γενικευτική ικανότητα του δικτύου.

Με την αύξηση του αριθμού των batches, παρατηρείται μεγάλη μείωση στην γενικευτική ικανότητα του δικτύου και αυτή είναι πιο ευαίσθητη στις αλλαγές του πλήθους των νευρώνων.

F	LR	H1	H2	H3	B		G
Tanh	0.005	7	7	7	40		0.88475
Tanh	0.005	7	15	7	40		0.93275
Tanh	0.005	15	15	7	40		0.9625
Tanh	0.005	15	7	15	40		0.94225
Tanh	0.005	7	7	15	40		0.91525
Tanh	0.005	15	7	7	40		0.95
Tanh	0.005	15	15	15	40		0.9495
Tanh	0.005	7	7	7	400		0.247
Tanh	0.005	7	15	7	400		0.3635
Tanh	0.005	15	15	7	400		0.91775
Tanh	0.005	15	7	15	400		0.874
Tanh	0.005	7	7	15	400		0.247
Tanh	0.005	15	7	7	400		0.68475
Tanh	0.005	15	15	15	400		0.89925
Relu	0.000005	7	7	7	40		0.26225
Relu	0.000005	7	15	7	40		0.34075
Relu	0.000005	15	15	7	40		0.284
Relu	0.000005	15	7	15	40		0.37125
Relu	0.000005	7	7	15	40		0.45925
Relu	0.000005	15	7	7	40		0.3135
Relu	0.000005	15	15	15	40		0.4765
Relu	0.000005	7	7	7	400		0.26225
Relu	0.000005	7	15	7	400		0.26225
Relu	0.000005	15	15	7	400		0.2485
Relu	0.000005	15	7	15	400		0.2485
Relu	0.000005	7	7	15	400		0.247
Relu	0.000005	15	7	7	400		0.2485
Relu	0.000005	15	15	15	400		0.247