



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Σχεδιασμός και Υλοποίηση βάσης δεδομένων με παράγοντες αλληλεπίδρασης απώλειας ακοής και διαβήτη

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΑΝΑΣΤΑΣΙΑΣ ΤΣΙΤΟΥΡΑ

Επιβλέπων : Δημήτριος Κουτσούρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάης 2020



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**Σχεδιασμός και Υλοποίηση βάσης δεδομένων με
παράγοντες αλληλεπίδρασης απώλειας ακοής και
διαβήτη**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΑΝΑΣΤΑΣΙΑΣ ΤΣΙΤΟΥΡΑ

Επιβλέπων : Δημήτριος Κουτσούρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την: 5 Μαΐου 2020

Επιβλέπων

Μέλος

Μέλος

.....
Δ. ΚΟΥΤΣΟΥΡΗΣ
Καθηγητής Ε.Μ.Π.

.....
Π. ΤΣΑΝΑΚΑΣ
Καθηγητής Ε.Μ.Π.

.....
Γ. ΜΑΤΣΟΠΟΥΛΟΣ
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάης 2020

(Υπογραφή)

.....

ΑΝΑΣΤΑΣΙΑ ΤΣΙΤΟΥΡΑ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2020 – All rights reserved

Περίληψη

Ο συσχετισμός διαφόρων βιομετρικών μεγεθών έχει ιδιαίτερα μεγάλο ενδιαφέρον. Όπως για παράδειγμα η απώλεια ακοής σε αλληλεπίδραση με τον διαβήτη. Στην παρούσα διπλωματική καταπιανόμαστε με σχετικά θέματα αναπτύσσοντας προσεγγίσεις μέσω νευρωνικών δικτύων που βελτιώνουν την αντίληψή μας για το αντικείμενο. Ειδικότερα προτείνουμε ένα φθηνό υπολογισμό συνάρτησης μεταφοράς (της $\tanh x$) μέσω κυβικών splines, ο οποίος αποδεικνύεται ιδιαίτερα αποτελεσματικός.

Λέξεις Κλειδιά: Κυβικές splines, Νευρωνικά δίκτυα, Συναρτήσεις μεταφοράς, Υπερβολική εφαπτομένη

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

The correlation of different biometric quantities is of particular interest. For example, hearing loss in interaction with diabetes. In this dissertation we deal with relevant issues by developing approaches through neural networks that improve our perception of the subject. In particular, we propose a cheap calculation of transfer function (of $\tanh x$) via cubic splines, which proves to be very effective.

Keywords: Cubic splines, Neural networks, Transfer functions, Hyperbolic tangent

Η σελίδα αυτή είναι σκόπιμα λευκή

Πίνακας περιεχομένων

Κεφάλαιο	σελίδα
1. Εισαγωγή	1
2. Splines - Θεωρητικό υπόβαθρο	2
3. Νευρωνικά δίκτυα	16
4. Η συνάρτηση $\tanh x$	27
5. Υλοποίηση της $\tanh x$ σε νευρωνικά δίκτυα	47
6. Εκπαίδευση στο αρχείο δεδομένων <code>diabetes.csv</code>	77
7. Εκπαίδευση στο αρχείο δεδομένων <code>EV0real_time_synth.csv</code>	86
Σύνοψη - Συμπεράσματα - Μελλοντικές επεκτάσεις	98
Βιβλιογραφία	99

Η σελίδα αυτή είναι σκόπιμα λευκή

1

Εισαγωγή

Τα τελευταία χρόνια υπάρχει ένα συνεχές ενδιαφέρον για την βιοϊατρική τεχνολογία [GKK16, KGK15, KMK18, KTGK17, LBKKIK16, MKPAI18, PKK17, PKGK15, VPCMK17]. Ειδικότερα όσον αφορά την ακοή [KKGK16, KMK19, KPBK20, KTMBK17, KK18, PKGKB16, SKBKK17, TKLPKS17].

Εμείς εδώ εφαρμόσαμε τα τη νέα μέθοδο που μας απασχόλησε σε δεδομένα σχετικά με χοληστερίνη [Pulu92], σωματικό βάρος [UG19], διαβήτη [PIMA] και απώλεια ακοής [CPRAB19].

Ειδικότερα προτείνουμε μια νέα φθινή προσέγγιση για τον υπολογισμό της υπερβολικής εφαπτομένης ($\tanh x$), η οποία χρησιμοποιείται ως συνάρτηση μεταφοράς στα νευρωνικά δίκτυα [TT19]. Η νέα μέθοδος βασίζεται στην προσέγγιση με κυβικές splines. Έτσι ο υπολογισμός περιορίζεται σε ένα πολώνυμο 3ου βαθμού που απαιτεί 3 πολλαπλασιασμούς ενώ η κλασσική προσέγγιση (π.χ. στο MATLAB) απαιτεί 7 πολλαπλασιασμούς και 1 διαίρεση.

Η διατήρηση βασικών ιδιοτήτων της συνάρτησης (μονοτονία, συνέχεια, συμμετρία, διατήρηση εντός διαστήματος $[-1,1]$) επιτυγχάνεται όταν η προσέγγιση βασισθεί σε ειδικό σετ σημείων. Δυο τέτοια σετ υλοποιήθηκαν εδώ.

Η εφαρμογή της προσέγγισης σε βιομετρικά δεδομένα έδειξε ότι η νέα φθινή και γρήγορη προσέγγιση στάθηκε πολύ ικανοποιητικά κοντά στα αποτελέσματα που δίνει ο κλασσικός υπολογισμός της εν λόγω συνάρτησης μεταφοράς στα νευρωνικά δίκτυα.

2

Splines: Θεωρητικό υπόβαθρο

Εδώ γράφουμε σύντομα τις τεχνικές/μεθοδολογίες/μοντέλα που πιθανά θα χρησιμοποιήσει η διπλωματική και είναι αναγκαία η κατανόησή τους από τον αναγνώστη πριν από την παρουσίαση της ανάλυσης και σχεδίασης του συστήματος.

Πρόκειται για τεχνικές/μεθοδολογίες/μοντέλα που έχουν προταθεί από άλλους και δεν είναι πρωτότυπη δουλειά της διπλωματικής.

Μετά βάζουμε 1 ενότητα για κάθε τεχνική/μεθοδολογία/μοντέλο, όπου και δίνουμε λεπτομερή περιγραφή.

2.1 Παρεμβολή με συναρτήσεις splines

Στην κλασσική πολυωνυμική παρεμβολή αναπτύσσονται μέθοδοι για την προσέγγιση μίας συνάρτησης $f(x)$ σε κλειστό διάστημα με την χρήση πολυωνύμων. Αν και αυτός ο τρόπος προσέγγισης είναι κατάλληλος σε αρκετές περιπτώσεις, υπάρχουν πολλά παραδείγματα πραγματικών προβλημάτων για τα οποία η πολυωνυμική προσέγγιση δεν δίνει καλά αποτελέσματα.

Μία άλλη διαδικασία η οποία μπορεί να εφαρμοστεί για την κατασκευή παρεμβολικών συναρτήσεων, στηρίζεται στο να χωρίσουμε το διάστημα $[a, \beta]$ σε υποδιαστήματα, και σε κάθε υποδιάστημα να κατασκευάσουμε ένα διαφορετικό εν γένει προσεγγιστικό πολυώνυμο. Η προσέγγιση με συναρτήσεις αυτού του είδους ονομάζεται προσέγγιση με **τιμηματικά πολυώνυμα** (piecewise polynomials) αφού συνήθως αυτές οι συναρτήσεις είναι πολυώνυμα.

Ο απλούστερος τύπος προσέγγισης με τμηματικά πολυώνυμα είναι η *τμηματική γραμμική παρεμβολή*, όπου ένα σύνολο δεδομένων της μορφής:

$$\left\{ \left(x_0, f(x_0) \right), \left(x_1, f(x_1) \right), \dots, \left(x_n, f(x_n) \right) \right\}$$

συνδέονται με ευθύγραμμα τμήματα. Το μειονέκτημα της προσέγγισης με αυτού του είδους συναρτήσεις είναι ότι στα άκρα των υποδιαστημάτων δεν είναι σίγουρο ότι υπάρχει η παράγωγος. Γεωμετρικά αυτό σημαίνει ότι η παρεμβολική συνάρτηση δεν είναι λεία (smooth). Όμως συχνά απαιτείται, από φυσικές συνθήκες, η παρεμβολική συνάρτηση να είναι λεία και επομένως διαφορίσιμη.

Ένας άλλος τρόπος τον οποίον θα μπορούσαμε να ακολουθήσουμε, είναι να χρησιμοποιήσουμε τμηματικά πολυώνυμα τύπου Hermite. Για παράδειγμα, αν οι τιμές της $f(x)$ και $f'(x)$ είναι γνωστές στα σημεία:

$$x_0 < x_1 < x_2 < \dots < x_n$$

τότε ένα πολυώνυμο τύπου Hermite βαθμού 3 θα μπορούσε να χρησιμοποιηθεί σε κάθε υποδιάστημα:

$$\left[x_0, x_1 \right], \left[x_1, x_2 \right], \dots, \left[x_{n-1}, x_n \right]$$

για την κατασκευή μιας συνάρτησης η οποία είναι συνεχώς διαφορίσιμη στο διάστημα $\left[x_0, x_n \right]$. Όμως η δυσκολία με την χρήση τμηματικών πολυωνύμων τύπου Hermite για γενικά προβλήματα παρεμβολής είναι ότι πρέπει να γνωρίζουμε την παράγωγο της συνάρτησης την οποίαν θέλουμε να προσεγγίσουμε.

Ορισμός της κυβικής συνάρτησης spline

Το πλέον γνωστό και ικανό τμηματικό προσεγγιστικό πολυώνυμο, το οποίο σε κάθε υποδιάστημα χρησιμοποιεί πολυώνυμο τρίτου βαθμού, ονομάζεται κυβική συνάρτηση **spline** (cubic spline function). Η συνάρτηση spline θα συμβολίζεται με S , και επιλέγεται έτσι ώστε να είναι συνεχής και να έχει τις δύο πρώτες παραγώγους στα κομβικά σημεία. Είναι γνωστόν ότι κάθε κυβικό πολυώνυμο ορίζεται από τέσσερις συντελεστές και επομένως απαιτούνται ανάλογες συνθήκες για την κατασκευή τους. Ο παρακάτω ορισμός καταγράφει αυτές τις συνθήκες.

Ορισμός: Έστω η $f\left(\left[\alpha, \beta\right]\right)$ και το σύνολο των σημείων:

$$\alpha = x_0 < x_1 < x_2 < \dots < x_n = \beta.$$

Η spline τρίτου βαθμού S , που παρεμβάλλει την $f(x)$ είναι μία συνάρτηση η οποία ικανοποιεί τις συνθήκες:

I. Η συνάρτηση S είναι ένα κυβικό πολυώνυμο, στο υποδιάστημα $[x_i, x_{i+1}]$ συμβολίζεται με S_i , για κάθε $i = 0, 1, \dots, n-1$

II. $S(x_i) = f(x_i), i = 0, 1, \dots, n$

III. $S_{i+1}(x_{i+1}) = S_i(x_{i+1}), i = 0, 1, \dots, n-2$

IV. $S'_{i+1}(x_{i+1}) = S'_i(x_{i+1}), i = 0, 1, \dots, n-2$

V. $S''_{i+1}(x_{i+1}) = S''_i(x_{i+1}), i = 0, 1, \dots, n-2$

VI. στα άκρα του διαστήματος $[\alpha, \beta]$ ισχύει μία από τις επόμενες συνοριακές συνθήκες:

a) $S''(x_0) = S''(x_n) = 0$

b) $S'(x_0) = f'(x_0)$ και $S'(x_n) = f'(x_n)$. \square

Όταν ισχύουν οι συνοριακές συνθήκες (a), η κυβική spline ονομάζεται *φυσική spline* (natural spline).

2.2 Κατασκευή κυβικής spline

Για να κατασκευάσουμε την παρεμβολική κυβική spline για μία συνάρτηση $f(x)$, εφαρμόζουμε τις συνθήκες του ορισμού της στο κυβικό πολυώνυμο της μορφής:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad (2.1)$$

$$i = 0, 1, \dots, n - 1.$$

Προφανώς ισχύουν:

$$S_i(x_i) = a_i = f(x_i), \quad i = 0, 1, \dots, n - 1$$

και από την συνθήκη (iii) του ορισμού προκύπτει:

$$a_{i+1} = S_{i+1}(x_{i+1}) = S_i(x_{i+1})$$

$$a_i + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 + d_i(x_{i+1} - x_i)^3$$

για κάθε $i = 0, 1, \dots, n - 2$. Ορίζουμε τις ποσότητες

$$h_i = x_{i+1} - x_i \text{ και } a_n = f(x_n).$$

Τότε ισχύουν οι σχέσεις:

$$a_{i+1} = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3, \quad (2.2)$$

$$i = 0, 1, \dots, n - 1$$

Ορίζουμε τώρα ότι $b_n = S'(x_n)$ και από την σχέση

$$S'_i(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2$$

συνεπάγεται ότι $b_i = S'(x_i)$, $i = 0, 1, \dots, n - 1$.

Αν εφαρμόσουμε την συνθήκη (iv) του ορισμού, τότε προκύπτουν οι σχέσεις:

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2, \quad i = 0, 1, \dots, n - 1 \quad (2.3)$$

Τέλος, αν ορίσουμε την ποσότητα $c_n = S''(x_n)/2$, και εφαρμόσουμε την συνθήκη (v) του ορισμού θα προκύψουν με ανάλογο τρόπο οι σχέσεις:

$$c_{i+1} = c_i + 3d_i h_i, \quad i = 0, 1, \dots, n-1 \quad (2.4)$$

Η διαδικασία που ακολουθούμε τώρα είναι η εξής. Από την (2.4) υπολογίζουμε τον συντελεστή d_i τον οποίον αντικαθιστούμε στις σχέσεις (2.2) και (2.3) οπότε θα προκύψουν οι σχέσεις:

$$a_{i+1} = a_i + b_i h_i + \frac{h_i^2}{3} (2c_i + c_{i+1}), \quad (2.5)$$

$$i = 0, 1, \dots, n-1$$

και

$$b_{i+1} = b_i + h_i (c_i + c_{i+1}), \quad i = 0, 1, \dots, n-1 \quad (2.6)$$

Από την (2.5) υπολογίζουμε τον συντελεστή b_i με την σχέση:

$$b_i = \frac{1}{h_i} (a_{i+1} - a_i) - \frac{h_i}{3} (2c_i + c_{i+1}) \quad (2.7)$$

οπότε θα ισχύει και η σχέση:

$$b_{i-1} = \frac{1}{h_{i-1}} (a_i - a_{i-1}) - \frac{h_{i-1}}{3} (2c_{i-1} + c_i) \quad (2.8)$$

Θεωρούμε την σχέση (2.6) με τους δείκτες ελαττωμένους κατά ένα και αντικαθιστούμε τις εκφράσεις των b_i και b_{i-1} από τους τύπους (2.7) και (2.8) αντίστοιχα. Τότε προκύπτουν οι εξισώσεις:

$$\begin{aligned} & h_{i-1} c_{i-1} + 2(h_{i-1} + h_i) c_i + h_i c_{i+1} = \\ & = \frac{3}{h_i} (a_{i+1} - a_i) - \frac{3}{h_{i-1}} (a_i - a_{i-1}) \end{aligned} \quad (2.9)$$

$$i = 1, 2, \dots, n-1$$

οι οποίες ορίζουν ένα τριδιαγώνιο γραμμικό σύστημα με αγνώστους τα στοιχεία του διανύσματος $\mathbf{c} = [c_0, c_1, \dots, c_n]^T$. Οι τιμές h_i , $i = 0, 1, \dots, n-1$ είναι γνωστές από την απόσταση των σημείων, και οι τιμές των a_i δίνονται από τις σχέσεις $a_i = f(x_i)$, $i = 1, 2, \dots, n$.

Είναι φανερό ότι όταν υπολογιστεί το διάνυσμα c , τότε υπολογίζονται οι τιμές $b_i, i = 0, 1, \dots, n - 1$ από την σχέση (2.7), και οι τιμές $d_i, i = 0, 1, \dots, n - 1$ από την σχέση (2.4). Μετά από τους υπολογισμούς αυτούς τα κυβικά πολυώνυμα $S_i, i = 0, 1, \dots, n - 1$ εύκολα κατασκευάζονται.

Αν θεωρήσουμε την περίπτωση όπου οι συνοριακές συνθήκες είναι του τύπου

$$S''(\alpha) = S''(\beta),$$

τότε το γραμμικό σύστημα (2.9) θα έχει την μορφή:

$$Ac = q \tag{2.10}$$

όπου ο τριδιαγώνιος πίνακας A γράφεται:

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ h_0 2(h_0 + h_1) & h_1 & 0 & \cdot & \cdot & \cdot & \cdot \\ 0 & h_1 & 2(h_1 + h_2) h_2 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & \cdot \\ \cdot & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & 0 & h_{n-2} 2(h_{n-2} + h_{n-1}) h_{n-1} & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 & 0 & 1 \end{bmatrix}$$

και

$$c = \begin{bmatrix} c_0 \\ c_1 \\ \cdot \\ \cdot \\ \cdot \\ c_{n-1} \\ c_n \end{bmatrix}, q = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \cdot \\ \cdot \\ \cdot \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_0}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix},$$

με $c_0 = c_n = 0$. Ανάλογη μορφή για το γραμμικό σύστημα θα προκύψει όταν οι συνοριακές συνθήκες είναι της μορφής:

$$S'(x_0) = f'(x_0) \text{ και } S'(x_n) = f'(x_n).$$

Το παρακάτω θεώρημα εξασφαλίζει την ύπαρξη και μοναδικότητα της λύσης του συστήματος (2.10) όταν πρόκειται για την φυσική κυβική spline. Ανάλογο θεώρημα ισχύει και για splines που αντιστοιχούν σε άλλου τύπου συνοριακές συνθήκες.

Θεώρημα: Αν $f \in C^2[\alpha, \beta]$, τότε η $f(x)$ έχει μία μόνο παρεμβολική φυσική spline που ικανοποιεί τις συνοριακές συνθήκες $S''(\alpha) = S''(\beta) = 0$. \square

Η απόδειξη του θεωρήματος βασίζεται στο γεγονός ότι ο πίνακας A του συστήματος (2.10) έχει αυστηρά διαγώνια υπεροχή, και επομένως το γραμμικό σύστημα έχει μοναδική λύση.

Παράδειγμα: Να υπολογιστεί η συνάρτηση spline $S(x)$ η οποία παρεμβάλλει την συνάρτηση $f(x) = 1/x$ στα σημεία $x_0 = 1, x_1 = 2, x_2 = 3, x_3 = 4$ και ικανοποιεί τις συνθήκες $S''(1) = S''(4) = 0$.

Η συνάρτηση $S(x)$ είναι ένα τμηματικό κυβικό πολυώνυμο της μορφής:

$$S(x) = \begin{cases} S_0(x), & x \in [1, 2] \\ S_1(x), & x \in [2, 3] \\ S_2(x), & x \in [3, 4] \end{cases}$$

όπου για $i = 0, 1, 2$ έχουμε τις σχέσεις:

$$\begin{aligned} S_i(x) &= a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \\ S'_i(x) &= b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2 \\ S''_i(x) &= 2c_i + 6d_i(x - x_i) \end{aligned}$$

Έχουμε

$$a_i = f(x_i) = 1/x_i$$

οπότε

$$a_0 = 1, a_1 = 1/2, a_2 = 1/3.$$

Προφανώς ισχύει ότι $h_i = x_{i+1} - x_i = 1, i = 0, 1, 2$ οπότε από την συνθήκη (iii) του ορισμού προκύπτει:

$$a_{i+1} = a_i + b_i + c_i + d_i, a_3 = f(x_3) = 1/4, i = 0, 1, 2.$$

Επίσης αν ορίσουμε το $b_4 = S'(4)$, και εφαρμόσουμε την συνθήκη (iv) του ορισμού προκύπτει η σχέση:

$$b_{i+1} = b_i + 2c_i + 3d_i, i = 0, 1, 2.$$

Τέλος αν ορίσουμε το $c_4 = S''(4)/2$, και εφαρμόσουμε την συνθήκη (v) του ορισμού προκύπτει η σχέση:

$$c_{i+1} = c_i + 3d_i, i = 0, 1, 2.$$

Η τελευταία γράφεται:

$$d_i = 1/3(c_{i+1} - c_i), i = 0, 1, 2$$

οπότε οι δύο προηγούμενες για $i = 0, 1, 2$ γράφονται αντίστοιχα:

$$a_{i+1} = a_i + b_i + 1/3(2c_i - c_{i+1})$$

$$b_{i+1} = b_i + (c_i + c_{i+1})$$

Από την 1^η υπολογίζουμε το b_i με τη σχέση

$$b_i = a_{i+1} - a_i - 1/3(2c_i + c_{i+1})$$

Προφανώς θα ισχύει και η σχέση

$$b_{i-1} = (a_i - a_{i-1}) - 1/3(2c_{i-1} + c_i)$$

Θεωρώ την 2^η με τους δείκτες ελαττωμένους κατά 1 οπότε θα προκύψει η σχέση:

$$b_i = b_{i-1} + (c_{i-1} + c_i).$$

Αν στην τελευταία αντικαταστήσουμε τις εκφράσεις των δύο προηγούμενων θα προκύψουν οι εξισώσεις:

$$c_{i-1} + 4c_i + c_{i+1} = 3(a_{i+1} - a_i) - 3(a_i - a_{i-1})$$

με $c_0 = c_3 = 0$ και $i = 1, 2$. Από την παραπάνω εξίσωση προκύπτει το γραμμικό σύστημα

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1/4 \\ 0 \end{bmatrix}$$

Υπολογιζόμενα τα c_i εύκολα υπολογίζονται τα b_i και d_i και επομένως η $S(x)$.

Ο επόμενος αλγόριθμος περιγράφει τα βασικά βήματα για την κατασκευή της κυβικής spline S για την προσέγγιση της συνάρτησης $f(x)$ με συνοριακές συνθήκες της μορφής $S''(x_0) = S''(x_n) = 0$.

Αλγόριθμος φυσικής κυβικής spline

Εδώ θα περιγράψουμε τον αλγόριθμο που υπολογίζει την κυβική spline, S , για την προσέγγιση της $f(x)$ ορισμένης στα σημεία $\alpha = x_0 < x_1 < x_2 < \dots < x_n = \beta$ της μορφής (2.2), όταν ικανοποιεί τις συνοριακές συνθήκες $S''(x_0) = S''(x_n) = 0$.

INPUT: Τα σημεία x_0, x_1, \dots, x_n , ο αριθμός n , και

οι τιμές $a_i = f(x_i)$, $i = 0, 1, \dots, n$

OUTPUT: Οι συντελεστές a_i, b_i, c_i, d_i για

$i = 0, 1, \dots, n - 1$

Βήμα 1: Για $i = 0, 1, \dots, n - 1$

Θέσε $h_i = x_{i+1} - x_i$

Βήμα 2: Για $i = 0, 1, \dots, n - 1$

Θέσε:

$$q_i = \frac{3 \left[a_{i+1} h_{i-1} - a_i (x_{i+1} - x_{i-1}) + a_{i-1} h_i \right]}{h_{i-1} h_i}$$

Επίλυση του συστήματος (2.10) με την μέθοδο

Crout και υπολογισμός των συντελεστών

b_i, d_i , $i = 0, 1, \dots, n - 1$

Βήμα 3: Θέσε $l_0 = 1, u_0 = 0, z_0 = 0$

Βήμα 4: Για $i = 1, 2, \dots, n - 1$, Θέσε:

$$\begin{aligned} l_i &= 2(x_{i+1} - x_{i-1}) - h_{i-1} u_{i-1} \\ u_i &= h_i / l_i \\ z_i &= (q_i - h_{i-1} z_{i-1}) / l_i \end{aligned}$$

Βήμα 5: Θέσε $l_n = 1, z_n = 0, c_n = 0$

Βήμα 6: Για $i = n - 1, n - 2, \dots, 0$, Θέσε:

$$\begin{aligned}c_i &= z_i - u_i c_{i+1} \\b_i &= \left(a_{i+1} - a_i \right) / h_i - h_i \left(c_{i+1} + 2c_i \right) / 3 \\d_i &= \left(c_{i+1} - c_i \right) / 3h_i\end{aligned}$$

Βήμα 7: Output (a_i, b_i, c_i, d_i για $i = 0, 1, \dots, n - 1$)

STOP.

Το πρόγραμμα Matlab που ακολουθεί υπολογίζει την προσέγγιση με φυσική κυβική spline σε ένα σημείο. Στην είσοδο δίνουμε τα διανύσματα x, y που περιέχουν τις διαθέσιμες τιμές. Επιστρέφονται οι συντελεστές των πολυωνύμων.

Λογισμικό Matlab για την προσέγγιση με φυσική κυβική Spline

Το **Matlab** διαθέτει την συνάρτηση **spline** για εύρεση των συντελεστών φυσικής κυβικής spline. Έστω ότι έχουμε τα δεδομένα $x = 1, 2, 3, 5, 7$ και 8 , με $y = 1, 1/2, 1/3, 1/5, 1/7, 1/8$ αντίστοιχα (βασίζόμενοι στην συνάρτηση $y(x) = \frac{1}{x}$) και θα ζητήσουμε την εύρεση των συντελεστών της προσέγγισης με κυβική spline.

Παράδειγμα: Στο παράδειγμα έχουμε τα δεδομένα $x = 1, 2, 3, 5, 7$ και 8 , με $y = 1, 1/2, 1/3, 1/5, 1/7, 1/8$ αντίστοιχα και θα ζητήσουμε την εύρεση των συντελεστών απαιτώντας παράλληλα $y'(1) = -\frac{1}{1^2} = -1, y'(8) = -\frac{1}{64}$. Γράφουμε απλά στο command window του MATLAB.

```
>> cs=spline([1 2 3 5 7 8],[-1 1./[1 2 3 5 7 8] -1/64]);cs.coefs
ans =
-0.219818835210433    0.719818835210433   -1.000000000000000    1.000000000000000
-0.007210161035368    0.060362329579135   -0.219818835210433    0.500000000000000
-0.005851425113614    0.038731846473029   -0.120724659158269    0.333333333333333
 0.000049088618850    0.003623295791346   -0.036014374629520    0.200000000000000
-0.000842842323651    0.003917827504446   -0.020932128037937    0.142857142857143
```

Δηλαδή τα πολυώνυμα είναι

$$S_1 = -0.219818835210433(x - 1)^3 + 0.719818835210433(x - 1)^2 - (x - 1) + 1,$$

στο διάστημα [1,2].

$$S_2 = -0.007210161035368(x - 2)^3 + 0.060362329579135(x - 2)^2 - 0.219818835210433(x - 2) + 0.5,$$

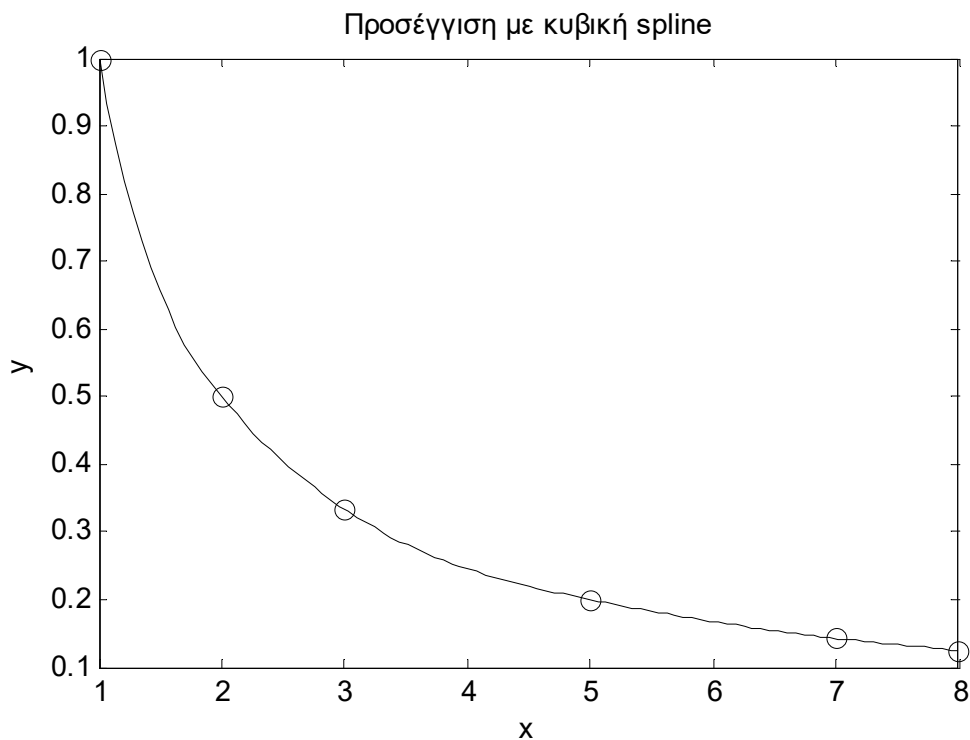
στο διάστημα [2,3].

$$S_3 = -0.005851425113614(x - 3)^3 + 0.038731846473029(x - 3)^2 - 0.120724659158269(x - 3) + 1/3,$$

στο διάστημα [3,5].

κ.ο.κ.

Στο επόμενο διάγραμμα βλέπουμε την προσέγγιση των σημείων (που δίνονται με κύκλους) από την κυβική spline που βρήκαμε παραπάνω.



Λογισμικό Mathematica για την προσέγγιση με φυσική κυβική Spline

Ακολουθεί ένα Mathematica Package με το οποίο μπορούμε να βρούμε τους συντελεστές της κυβικής spline. Υλοποιώντας τη σχετική θεωρία που αναπτύξαμε παραπάνω έχουμε,

```
CubSpline[Sigma_,bd_,ed_,a_,b_,c_,d_,m_,tau_] :=
Module[{g,aa,n,z,K,h},
  m=Length[Sigma];
  g=Table[Sigma[[j,2]],{j,1,m}];
  tau=Table[Sigma[[j,1]],{j,1,m}];
  a=Table[0,{j,1,m}];
  b=Table[0,{j,1,m}];
  c=Table[0,{j,1,m}];
  d=Table[0,{j,1,m}];
  aa=Table[0,{j,1,m}];
  n=Table[0,{j,1,m}];
  z=Table[0,{j,1,m}];
  K=Table[0,{j,1,m}];
  dif=Table[0,{j,1,m}];
```

```

dif[[1]]=tau[[2]]-tau[[1]];
aa[[1]]=3(g[[2]]-g[[1]])/dif[[1]]-3bd;
K[[1]]=2dif[[1]];
n[[1]]=1/2;
z[[1]]=aa[[1]]/K[[1]];
b[[1]]=bd;
For[j=2,j<m,j++,
    dif[[j]]=tau[[j+1]]-tau[[j]];
    aa[[j]]=(3/(dif[[j]]*dif[[j-1]]))*(g[[j+1]]*dif[[j-1]]-
        g[[j]]*(dif[[j]]+dif[[j-1]])+g[[j-1]]*dif[[j]]);
    K[[j]]=2*(dif[[j]]+dif[[j-1]])-dif[[j-1]]*n[[j-1]];
    n[[j]]=dif[[j]]/K[[j]];
    z[[j]]=(aa[[j]]-dif[[j-1]]*z[[j-1]])/K[[j]]
];
aa[[m]]=3ed-3(g[[m]]-g[[m-1]])/dif[[m-1]];
K[[m]]=dif[[m-1]]*(2-n[[m-1]]);
z[[m]]=(aa[[m]]-dif[[m-1]]z[[m-1]])/K[[m]];
c[[m]]=z[[m]];
For[j=m-1,j>0,j--,
    c[[j]]=z[[j]]-n[[j]]*c[[j+1]];
    b[[j]]=(g[[j+1]]-g[[j]])/dif[[j]]-dif[[j]]*
        (c[[j+1]]+2*c[[j]])/3;
    d[[j]]=(c[[j+1]]-c[[j]])/(3*dif[[j]]);
    a[[j]]=g[[j]]
]]

```


Οι παράμετροι στην είσοδο έχουν τις παρακάτω έννοιες

bd : y' (αριστερά)

ed : y' (δεξιά)

a, b, c, d : Οι συντελεστές των πολυωνύμων

m : το πλήθος των διαστημάτων

tau : τα σημεία στήριξης

Επαναλαμβάνουμε το προηγούμενο παράδειγμα.

```
In[2]:= data={1,1},{2,1/2},{3,1/3},{5,1/5},{7,1/7},{8,1/8};
In[3]:= CubSpline[data,-1,-1/64,a,b,c,d,m,tau];
```

Κατασκευάζουμε τα πολυώνυμα

```
In[4]:= For[i=1,i<m,i++,Poly[i,x_]=a[[i]]+b[[i]]*(x-tau[[i]])+c[[i]]
*(x-tau[[i]])^2+d[[i]]*(x-tau[[i]])^3]
```

Οπότε τα 5 πολυώνυμα είναι:

```
In[7]:= N[Poly[1,x],16]
Out[7]= 2.0000000000000000+0.7198188352104327 (-1.0000000000000000+x)^2-
0.2198188352104327 (-1.0000000000000000+x)^3-1.0000000000000000 x

In[8]:= N[Poly[2,x],16]
Out[8]= 0.5000000000000000-0.2198188352104327 (-2.0000000000000000+x)
+0.06036232957913456 (-2.0000000000000000+x)^2
-0.007210161035368504 (-2.0000000000000000+x)^3

In[9]:= N[Poly[3,x],16]
Out[9]= 0.3333333333333333-0.1207246591582691 (-3.0000000000000000+x)
+0.03873184647302905 (-3.0000000000000000+x)^2
-0.005851425113613910 (-3.0000000000000000+x)^3

In[10]:= N[Poly[4,x],16]
Out[10]= 0.2000000000000000-0.03601437462951986 (-5.0000000000000000+x)
+0.003623295791345584 (-5.0000000000000000+x)^2
+0.00004908861885002964 (-5.0000000000000000+x)^3

In[11]:= N[Poly[5,x],16]
Out[11]= 0.1428571428571429-0.02093212803793717 (-7.0000000000000000+x)
+0.003917827504445762 (-7.0000000000000000+x)^2
-0.0008428423236514523 (-7.0000000000000000+x)^3
```

Σε απόλυτη συμφωνία με ότι βρήκαμε στο MATLAB.

3

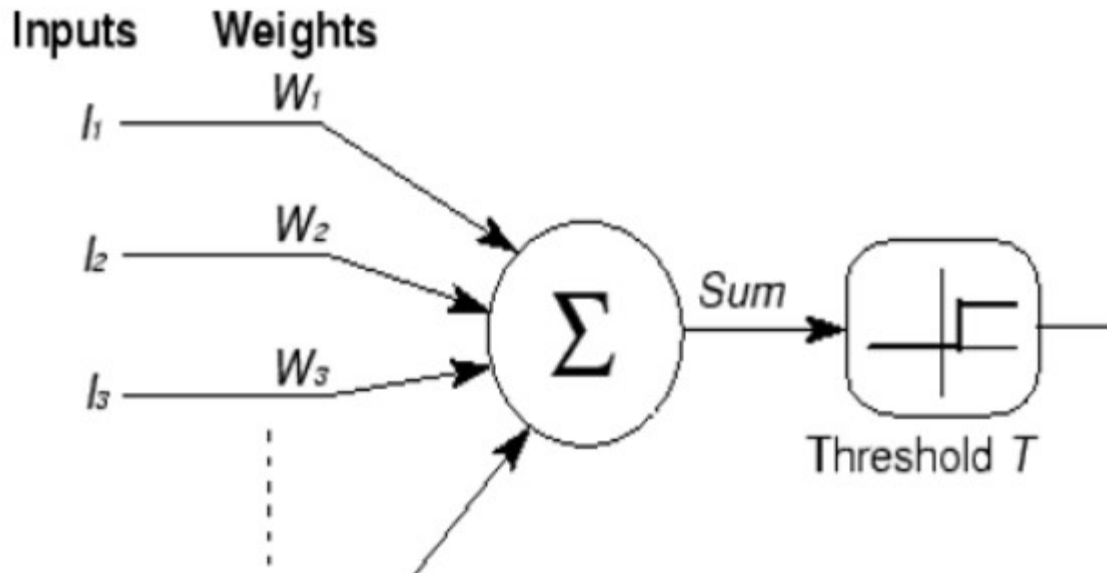
Νευρωνικά Δίκτυα

Τα νευρωνικά δίκτυα έχουν γίνει γνωστά και έχουν αναπτυχθεί μόνο κατά τα τελευταία σαράντα περίπου χρόνια. Εν τούτοις, η περιοχή αυτή έχει δει μια μεγάλη άνθηση, κρίνοντας από την μεγάλη ανάπτυξη που έχει παρατηρηθεί, από τον αριθμό των επιστημόνων που ασχολούνται με αυτά τα θέματα, και βέβαια από τα πολύ σημαντικά επιτεύγματα, που έχουν συμβάλει στο να γίνουν γνωστά σε ένα ευρύτερο κύκλο. Αποτελούν επομένως ένα θέμα με μεγάλο ενδιαφέρον στις τεχνολογικές επιστήμες. Το κύριο χαρακτηριστικό τους είναι ότι οι πρώτες αρχές και λειτουργίες τους βασίζονται στο νευρικό σύστημα των ζώντων οργανισμών (και φυσικά του ανθρώπου), αλλά η μελέτη και η χρήση τους έχει προχωρήσει πολύ πέρα από τους βιολογικούς οργανισμούς, και σήμερα τα νευρωνικά δίκτυα χρησιμοποιούνται για να λύσουν κάθε είδους προβλήματα με ηλεκτρονικό υπολογιστή. Η φιλοσοφία τους όμως είναι διαφορετική από τον τρόπο με τον οποίο δουλεύουν οι κλασικοί υπολογιστές. Η λειτουργία τους προσπαθεί να συνδυάσει τον τρόπο σκέψης του ανθρώπινου εγκεφάλου με τον αφηρημένο μαθηματικό τρόπο σκέψης. Έτσι στα νευρωνικά δίκτυα χρησιμοποιούμε ιδέες όπως, π.χ. ένα δίκτυο μαθαίνει και εκπαιδεύεται, θυμάται ή ξεχνά μια αριθμητική τιμή, κλπ. πράγματα που μέχρι τώρα τα αποδίδουμε μόνο στην ανθρώπινη σκέψη. Αλλά βέβαια μπορούν και χρησιμοποιούν επί πλέον και περίπλοκες μαθηματικές συναρτήσεις και κάθε είδους εργαλεία από την μαθηματική ανάλυση.

Τη δεκαετία του 1940 οι McCulloch και Pitts περιέγραψαν ένα απλό μοντέλο της δραστηριότητας του νευρώνα. Η κατάσταση του περιγράφεται από ένα δυαδικό αριθμό y .

$y = 0 \rightarrow$ ο νευρώνας είναι αδρανής

$y = 1 \rightarrow$ ο νευρώνας πυροβολεί στη μέγιστη συχνότητα



\leq

Οι συνάψεις περιγράφονται από τα **συναπτικά βάρη** w_i που είναι πραγματικοί αριθμοί, θετικοί για τις ενισχυτικές συνάψεις και αρνητικοί για τις ανασταλτικές. Αν i_1, i_2, \dots, i_N οι είσοδοι του νευρώνα τότε το άθροισμα του φορτίου που δέχεται ο άξονας είναι

$$Sum = \sum_{i=1}^n w_i x_i$$

Αν το άθροισμα είναι μεγαλύτερο από το κατώφλι (threshold) θ τότε ο νευρώνας πυροβολεί, διαφορετικά παραμένει αδρανής.

$$y = f(u - \theta)$$

Όπου $f(\cdot)$ η βηματική συνάρτηση:

Βηματική συνάρτηση 0/1 :

$$f(u) = \begin{cases} 0, & \text{αν } u \leq 0 \\ 1, & \text{αν } u > 0 \end{cases}$$

Υπάρχουν πολλές διαφορετικές μοντελοποιήσεις του νευρώνα που αποκλίνουν από το μοντέλο McCulloch-Pitts. Η πιο σημαντική διαφορά είναι στη μορφή της μη γραμμικής συνάρτησης $f()$ που χρησιμοποιείται στην έξοδο.

Η συνάρτηση αυτή που καλείται και συνάρτηση ενεργοποίησης του νευρώνα μπορεί να πάρει εναλλακτικά τις παρακάτω μορφές:

Βηματική συνάρτηση -1/1

$$f(u) = \begin{cases} -1, & \text{if } u \leq 0 \\ 1, & \text{if } u > 0 \end{cases}$$

Η συνάρτηση αυτή συμβολίζεται και ως sgn

Σιγμοειδής συνάρτηση

$$f(u) = 1/(1 + e^{-u})$$

Υπερβολική εφαπτομένη

$$f(u) = \tanh(u) = (1 - e^{-u})/(1 + e^{-u})$$

Συνάρτηση κατωφλίου

$$f(u) = 0, \text{ if } u \leq 0$$

$$f(u) = u, \text{ if } 0 < u < 1$$

$$f(u) = 1, \text{ if } u \geq 1$$

Συνάρτηση ράμπας

$$f(u) = 0, \text{ if } u \leq 0$$

$$f(u) = u, \text{ if } u \geq 0$$

Γραμμική

$$f(u) = u$$

3.1 Δίκτυο Perceptron

Ο Perceptron είναι ένας δυαδικός ταξινομητής, δηλαδή μία συνάρτηση η οποία απεικονίζει την είσοδο x (ένα διάνυσμα με πραγματικές τιμές) σε μία τιμή εξόδου $f(x)$ (μία και μοναδική δυαδική τιμή).

$$f(u) = \begin{cases} 1, & \text{if } w * u + b > 0 \\ 0, & \text{else} \end{cases}$$

όπου w είναι ένα διάνυσμα από βάρη με πραγματικές τιμές και $w * u$ είναι το εσωτερικό γινόμενο μεταξύ των διανυσμάτων w, u . Το b είναι το 'bias', ένας σταθερός όρος ο οποίος δεν εξαρτάται από καμία τιμή εισόδου.

Η τιμή της $f(u)$ (0 ή 1) χρησιμοποιείται για να ταξινομήσει το u είτε ως θετικό ή αρνητικό στιγμιότυπο, στην περίπτωση ενός δυαδικού προβλήματος ταξινόμησης. Το bias χρησιμοποιείται για την μετατόπιση της συνάρτησης ενεργοποίησης ή για να δώσει στον νευρώνα εξόδου ένα βασικό επίπεδο δραστηριότητας. Αν το b είναι αρνητικό τότε ο βεβαρημένος συνδυασμός των εισόδων πρέπει να παραγάγει μία θετική τιμή μεγαλύτερη του $-b$ έτσι ώστε να αναγκάσει τον νευρώνα που ταξινομεί να έχει τιμή άνω του κατωφλίου 0. Χωρικά, το bias μεταβάλλει την θέση (αλλά όχι τον προσανατολισμό) του συνόρου απόφασης.

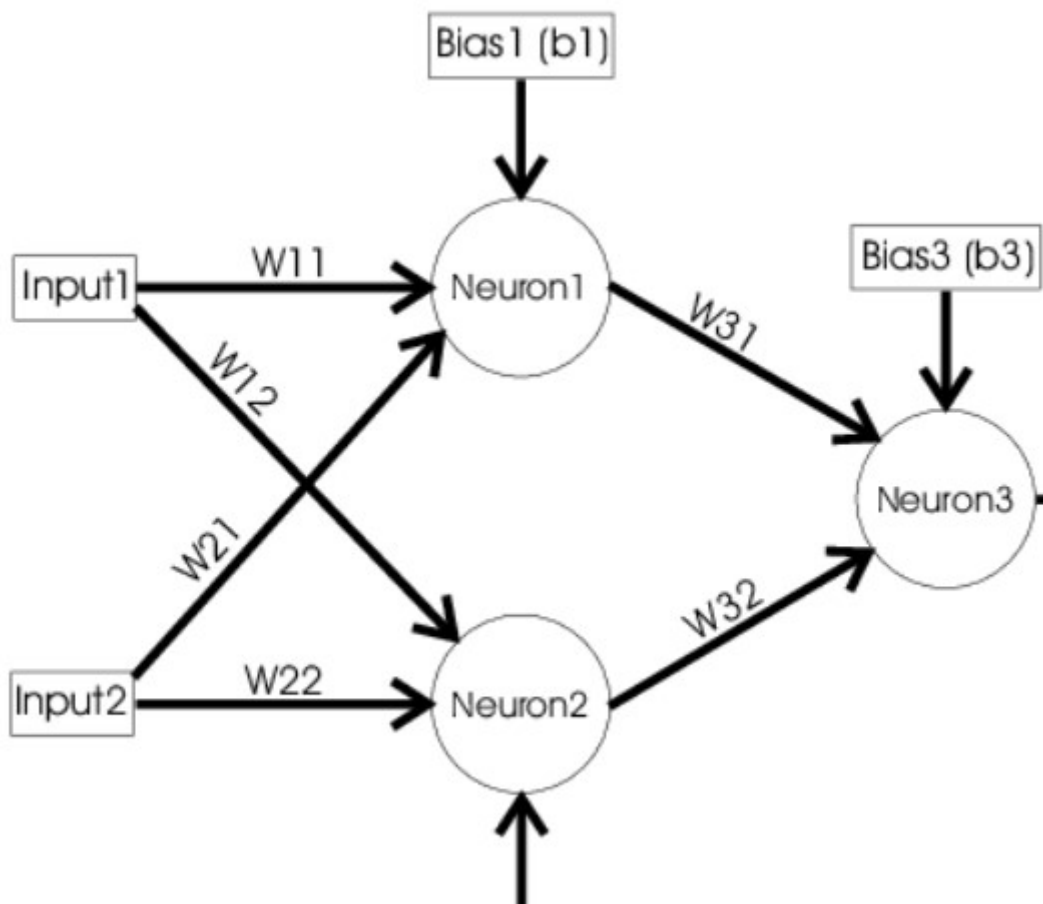
Εφόσον οι εισοδοί τροφοδοτούνται στο δίκτυο άμεσα μέσω των βεβαρημένων συνδέσεων, ο νευρώνας μπορεί να θεωρηθεί ως ένα απλό είδος νευρωνικού δικτύου εμπρός τροφοδότησης.

3.2 Δίκτυο MLP

Οι δυνατότητες αναπαράστασης διαχωριστικών επιφανειών είναι περιορισμένες στο δίκτυο Perceptron καθώς με ένα μόνο νευρώνα το δίκτυο μπορεί να αναπαραστήσει μόνο επίπεδες επιφάνειες. Ο περιορισμός αυτός αίρεται με τη χρήση περισσότερων νευρώνων. Για παράδειγμα, ένα δίκτυο 3 νευρώνων που χρησιμοποιούν την κλασική βηματική συνάρτηση $f(u) = 0, \text{ if } u < 0$ και $f(u) = 1, \text{ if } u \geq 0$

Το δίκτυο αποτελείται από 2 στρώματα:

1. Το στρώμα εξόδου που αποτελείται από το νευρώνα 3
2. Το κρυφό στρώμα που αποτελείται από τους νευρώνες 1 με έξοδο w_{31} και 2 με έξοδο w_{32} .



Το στρώμα των εισόδων 1 και 2 δε λογίζεται ως στρώμα νευρώνων καθώς οι κόμβοι της εισόδου δε λειτουργούν ως νευρώνες αλλά απλώς μεταδίδουν τις τιμές w_1 , w_2 στο επόμενο στρώμα.

Στο παρακάτω σχήμα φαίνονται οι τιμές των συναπτικών βαρών καθώς και επίσης οι τιμές των κατωφλίων των νευρώνων. Με τις συγκεκριμένες τιμές το δίκτυο υλοποιεί την πύλη XOR.

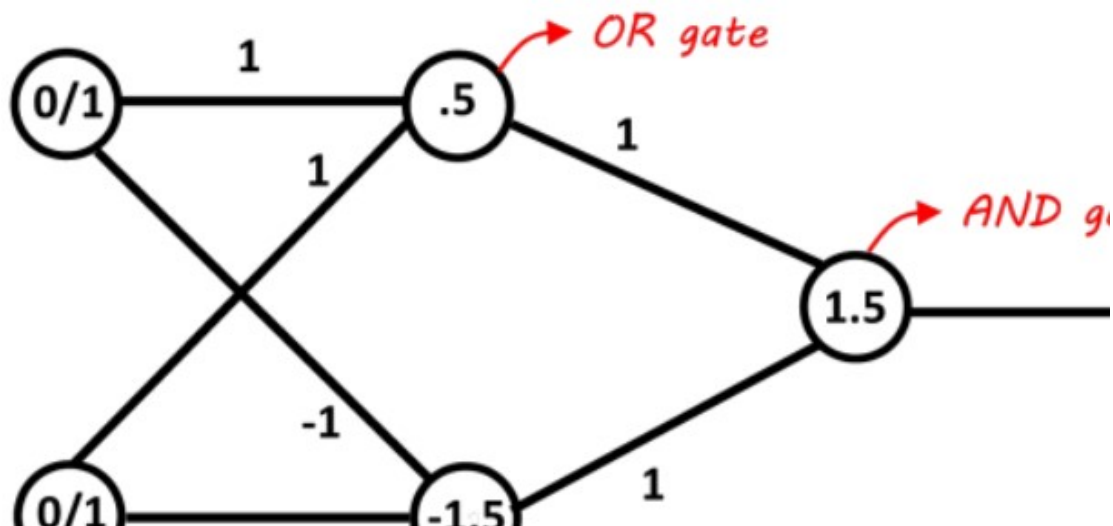
OUTPUT = XOR (w_1 , w_2), όπου w_1 , w_2 οι είσοδοι.

Ο πρώτος νευρώνας λειτουργεί ως πύλη OR και ο δεύτερος ως πύλη NOT AND.

Έπειτα, οι έξοδοί τους εισέρχονται σε πύλη AND.

Οι είσοδοι της πύλης NOT AND πρέπει να είναι αρνητικές για τις εισόδους 0/1.

Οι τιμές στις συνδέσεις είναι τα βάρη και οι τιμές στους νευρώνες είναι τα bias.



Αυτό γίνεται φανερό στον πίνακα όπου υπολογίζονται οι τιμές των νευρώνων για τα 4 πρότυπα εισόδου [0,0],[0,1],[1,0],[1,1]. Η συνάρτηση XOR είναι ίσως η πιο απλή περίπτωση συνάρτησης που δεν μπορεί να υλοποιηθεί από το δίκτυο Perceptron.

w_1	w_2	$f(w_1 + w_2 + 0.5)$	$f(-w_1 - w_2 - 1.5)$	$f(w_{31} + w_{32} + 1.5)$
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

3.3 Εκπαίδευση δικτύων MLP

Ο αλγόριθμος οπισθοδιάδοσης (Back-Propagation) είναι μια οικογένεια μεθόδων που χρησιμοποιούνται για την αποτελεσματική εκπαίδευση των τεχνητών νευρωνικών δικτύων, ακολουθώντας έναν αλγόριθμο βελτιστοποίησης βασισμένο στην κλίση που εκμεταλλεύεται τον κανόνα της αλυσίδας. Το κύριο χαρακτηριστικό της οπισθοδιάδοσης είναι η επαναληπτική, αναδρομική και αποδοτική μέθοδος για τον υπολογισμό των ανανεώσεων των βαρών για τη βελτίωση του δικτύου έως ότου είναι σε θέση να εκτελέσει το έργο για το οποίο εκπαιδεύεται. Είναι στενά συνδεδεμένη με τον αλγόριθμο Gauss-Newton .

Η οπισθοδιάδοση απαιτεί ότι οι παράγωγοι των συναρτήσεων ενεργοποίησης κατά το σχεδιασμό του δικτύου είναι γνωστοί. Η αυτόματη διαφοροποίηση είναι μια τεχνική που

παρέχει αυτόματα και αναλυτικά τις παραγώγους των συναρτήσεων στον αλγόριθμο εκπαίδευσης. Στο πλαίσιο της μάθησης, η οπισθοδιάδοση χρησιμοποιείται συνήθως από τον αλγόριθμο βελτιστοποίησης μείωσης κλίσης ώστε να ρυθμίσει το βάρος των νευρώνων υπολογίζοντας την κλίση της συνάρτησης απώλειας. Η οπίσθοδιάδοση υπολογίζει τις κλίσεις, ενώ η (στοχαστική) μείωση της κλίσης χρησιμοποιεί τις κλίσεις για την εκπαίδευση του μοντέλου (μέσω βελτιστοποίησης).

Ας θεωρήσουμε ένα δίκτυο MLP με

- L στρώματα
- $n = N(0)$ εισόδους
- $m = N(L)$ εξόδους

Ονομάζουμε τις εισόδους x_1, \dots, x_n και τις εξόδους y_1, \dots, y_m . Για μια σειρά από P διανύσματα εισόδου επιθυμούμε οι εξόδοι να επιτύχουν τιμές που δίνονται από αντίστοιχα P διανύσματα στόχων. Ονομάζουμε:

- $x^{(p)} = [x_1^{(p)}, \dots, x_n^{(p)}]^T$, το p -οστό διάνυσμα εισόδου
- $y^{(p)} = [y_1^{(p)}, \dots, y_m^{(p)}]^T$, το p -οστό διάνυσμα εξόδου
- $d^{(p)} = [d_1^{(p)}, \dots, d_m^{(p)}]^T$, το p -οστό διάνυσμα στόχων

Τα δεδομένα που απαιτούνται για να εκπαιδευτεί το δίκτυο είναι P ζεύγη διανυσμάτων εισόδων-στόχων $\{x^{(1)}, d^{(1)}\}, \{x^{(2)}, d^{(2)}\}, \dots, \{x^{(P)}, d^{(P)}\}$. Επιζητούμε τη βέλτιστη προσέγγιση της επιθυμητής κατάστασης χρησιμοποιώντας ένα κριτήριο κόστους. Το μέσο τετραγωνικό σφάλμα είναι ένα κλασικό κριτήριο κόστους

$$J = \frac{1}{P} \sum_{p=1}^P |d^{(p)} - y^{(p)}|^2 = \frac{1}{P} \sum_{p=1}^P \sum_{i=1}^m [d_i^{(p)} - y_i^{(p)}]^2$$

3.4 Αλγόριθμος εκπαίδευσης *Back-Propagation*

Τα συναπτικά βάρη w_{ij} είναι οι παράμετροι που πρέπει να διορθωθούν ώστε να ελαχιστοποιηθεί το σφάλμα J , καθώς τόσο οι εισόδοι $x^{(p)}$ όσο και οι στόχοι $d^{(p)}$ είναι δεδομένοι και σταθεροί. Σύμφωνα με τη μέθοδο κατάβασης δυναμικού, η μεταβολή της παραμέτρου w_{ij} ως προς το χρόνο t γίνεται χρησιμοποιώντας την παράγωγο του J ως προς w_{ij} .

$$\frac{d w_{ij}}{dt} = - \frac{\partial J}{\partial w_{ij}}$$

Η οποία ονομάζεται κλίση του J ως προς w_{ij} .

Χρησιμοποιώντας διακριτό χρόνο k , ονομάζουμε:

$w_{ij}(l, k)$ το συναπτικό βάρος που συνδέει το νευρώνα j του στρώματος $l-1$ με το νευρώνα i του στρώματος l κατά τη χρονική στιγμή k .

Χρησιμοποιώντας τον παραπάνω συμβολισμό, η μέθοδος κατάβασης δυναμικού εξομοιώνεται με μια διακριτή εξίσωση

$$w_{ij}(l, k + 1) - w_{ij}(l, k) = -\beta \frac{\partial J}{\partial w_{ij}(l, k)}$$

Η έξοδος του νευρώνα i δίνεται από τη σχέση:

$$a_i^{(k)}(l) = f(u_i^{(k)}(l))$$

Για να διευκολύνουμε στον υπολογισμό της παραγώγου $\frac{\partial J}{\partial w_{ij}(l, k)}$ θα ονομάσουμε

$$\delta_i^{(k)}(l) = -\frac{\partial J}{\partial w_{ij}(l, k)}$$

Την παράγωγο του κόστους ως προς τη δικτυακή διέγερση του νευρώνα i .

Μετά τις απαραίτητες απλοποιήσεις έχουμε:

$$\frac{\partial J}{\partial w_{ij}(l, k)} = -\delta_i^{(k)}(l) a_j^{(k)}(l-1)$$

Για $j = 0, 1, 2, \dots, N(l-1)$ και $l = 1, 2, \dots, L$

Απομένει ο υπολογισμός των σφαλμάτων $\delta_i^{(k)}(l)$ για όλους τους νευρώνες i όλων των στρωμάτων l . Ο υπολογισμός θα γίνει με μια φορά αντίστροφη σε σχέση με την ανάκληση, δηλαδή ξεκινώντας από το στρώμα εξόδου και πηγαίνοντας προς το πρώτο στρώμα.

1. Για το στρώμα εξόδου L :

Η παράγωγος του κόστους J ως προς τη διέγερση είναι:

$$\delta_i^{(k)}(L) = -\frac{dJ}{da_i^{(k)}(L)} \frac{\partial a_i^{(k)}(L)}{\partial u_i^{(k)}(L)} = -\frac{\partial J}{\partial a_i^{(k)}(L)} \frac{\partial f(u_i^{(k)}(L))}{\partial u_i^{(k)}(L)}$$

$$\delta_i^{(k)}(L) = d_i^{(k)} - y_i^{(k)} f'(u_i^{(k)}(L)).$$

Η τιμή του σφάλματος $\delta_i^{(k)}(L)$ που παίρνουμε είναι η διαφορά του στόχου από την έξοδο επί την παράγωγο f' της συνάρτησης ενεργοποίησης f .

2. Για οποιοδήποτε στρώμα $l = 1, \dots, L - 1$

Έχουμε

$$\delta_i^{(k)}(l) = \sum_{\mu=1}^{N^{(l+1)}} \delta_{\mu}^{(k)}(l+1) w_{\mu i}^{(k)}(l+1) f'(\mathbf{u}_i^{(k)}(l))$$

Ο παραπάνω τύπος έδωσε το όνομα “Back-propagation” σε όλη τη μέθοδο. Πράγματι από τον παραπάνω τύπο φαίνεται ότι το σφάλμα δ σε οποιοδήποτε νευρώνα του στρώματος l είναι συνάρτηση των σφαλμάτων δ του στρώματος $l + 1$.

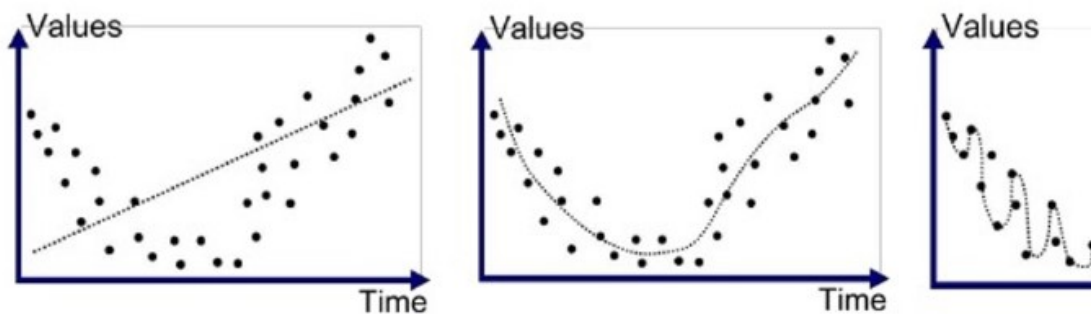
3.5 Μάθηση και Γενίκευση

Μάθηση είναι η διαδικασία αυτοπροσαρμογής ενός συστήματος με στόχο να βελτιστοποιήσει κάποιο κριτήριο καταλληλότητας για την επίλυση ενός συγκεκριμένου προβλήματος. Ένα νευρωνικό δίκτυο μαθαίνει ή εκπαιδεύεται όταν αλλάζει τις εσωτερικές τους παραμέτρους (συνήθως τα συναπτικά του βάρη) με στόχο να ελαχιστοποιήσει ένα κριτήριο. Η έννοια της μάθησης χαρακτηρίζει τα λεγόμενα ευφυή συστήματα. Τα νευρωνικά δίκτυα είναι τυπικές περιπτώσεις μηχανών μάθησης.

Η χρησιμότητα της μάθησης είναι περιορισμένη αν απλώς προσφέρει τη δυνατότητα καλής προσέγγισης των στόχων μόνο για τα συγκεκριμένα πρότυπα που χρησιμοποιήθηκαν στην εκπαίδευση και δεν προσφέρει στο μοντέλο την ικανότητα να εκτιμά με επιτυχία τους στόχους για τα πρότυπα που δεν έχει ξαναδεί. Η ικανότητα αυτή καλείται γενίκευση και είναι αυτή που διακρίνει τα ευφυή συστήματα από τα απλά συστήματα απομνημόνευσης, πχ μια μνήμη RAM.

Μπορούμε να κάνουμε 2 σφάλματα κατά τη μάθηση:

- 1) Να επιλέξουμε συνάρτηση ιδιαίτερα πολύπλοκη για να προσεγγίσουμε τα δεδομένα που διαθέτουμε. Το σφάλμα αυτό καλείται υπερ-μοντελοποίηση (over-modeling). Στην περίπτωση αυτή θα έχουμε κακή γενίκευση αν και μπορεί να έχουμε τέλεια προσέγγιση των διαθέσιμων δεδομένων.
- 2) Να επιλέξουμε μια συνάρτηση πολύ απλή. Το σφάλμα αυτό καλείται υπο-μοντελοποίηση (under-modeling). Στην περίπτωση αυτή θα έχουμε φτωχή προσέγγιση των διαθέσιμων δεδομένων και μέτρια εκτίμηση των μη διαθέσιμων.



Η επιλογή της κατάλληλης τάξης (πολυπλοκότητας) του μοντέλου είναι κεφαλαιώδους σημασίας για την επιτυχημένη γενίκευση. Συνήθως, η τάξη του μοντέλου είναι ανάλογη του πλήθους των ελεύθερων παραμέτρων. Σε ένα νευρωνικό δίκτυο η τάξη είναι ανάλογη του πλήθους των συναπτικών βαρών κι αυτό με τη σειρά του είναι ανάλογο του πλήθους των νευρώνων.

3.6 Νευρωνικά δίκτυα στο MATLAB

Ως γνωστόν το MATLAB [Matlab] είναι ένα πακέτο επιστημονικού/τεχνικού υπολογισμού. Μας δίνει τη δυνατότητα γρήγορου και εύκολου προγραμματισμού σχετικών θεμάτων. Για παράδειγμα αν θέλουμε να βρούμε τη ρίζα της συνάρτησης

$$f(x) = x^3 e^x - \sin \ln x - 2,$$

γράφουμε απλά στο command window του MATLAB:

```
>> fzero(@ (x) x^3*exp(x)-sin(log(x))-2,1)
ans =
0.914803129388098
```

Μπορούμε επίσης να βρούμε το ολοκλήρωμα

$$\int_1^2 e^{-x} \sin x^4 dx,$$

γράφοντας στο command window του MATLAB:

```
>> quadl(@ (x) exp(x).*sin(x.^4),1,2)
ans =
0.642278362045191
```

Για ειδικότερα προβλήματα έχουν δημιουργηθεί σχετικά toolboxes. Έτσι για θέματα βελτιστοποίησης παρέχεται το σχετικό optimization toolbox. Εκεί μπορούμε να λύσουμε πιο σύνθετα προβλήματα. Π.χ.

$$\min_{x_1, x_2} \{ \sin^2(x_1 + x_2) + (\cos(x_2 - 3) - 1)^2 \},$$

επιλύεται γράφοντας στο command window του MATLAB:

```
>> x=lsqnonlin(@(x) [sin(x(1)+x(2))-2],cos(x(2)-3)+1]', [2 5]')
Optimization terminated: relative function value
changing by less than OPTIONS.TolFun.
x =
    5.298360684610861
    6.126417276409533
```

Εμείς εδώ θα κάνουμε χρήση του Deep Learning Toolbox™ [UG19],[RG19].

Με αυτό είναι δυνατό να ορίζουμε νευρωνικά δίκτυα απευθείας. Για παράδειγμα θα προσπαθήσουμε να προσεγγίσουμε τη συνάρτηση $f(x) = x^2$, σε 20 σημεία, με ένα νευρωνικό δίκτυο με ένα κρυφό layer με 2 νευρώνες.

```
>> x = 1:20
x =
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
>> t = x.^2
      1      4      9     16     25     36     49     64     81
100    121    144    169    196    225    256    289    324    361
400

>> net = feedforwardnet(2,'trainlm');
>> net.trainParam.epochs = 1000;
>> net.trainParam.show = 10;
>> net.trainParam.goal = 0;
>> net = train(net,x,t);
>> a = net(x)

0.9786     4.0333     8.9960    15.9560    24.9469    35.9680
49.0030    64.0331    81.0451   100.0349   121.0071   143.9732
168.9477   195.9445   224.9714  256.0267  289.0942   324.1395
361.1070   399.9164
```

Παρατηρούμε ότι η μεταβλητή a έχει προσεγγίσει αρκετά καλά την t .

4

Η συνάρτηση $\tanh x$

Η συνάρτηση $\tanh x$ είναι μέλος μιας γενικότερης κατηγορίας συναρτήσεων που λέγονται υπερβολικές συναρτήσεις.

4.1 Υπερβολικές συναρτήσεις

Οι υπερβολικές συναρτήσεις ορίζονται όπως παρακάτω:

$$\sinh z = \frac{e^z - e^{-z}}{2}, \quad (z = x + iy)$$

$$\cosh z = \frac{e^z + e^{-z}}{2},$$

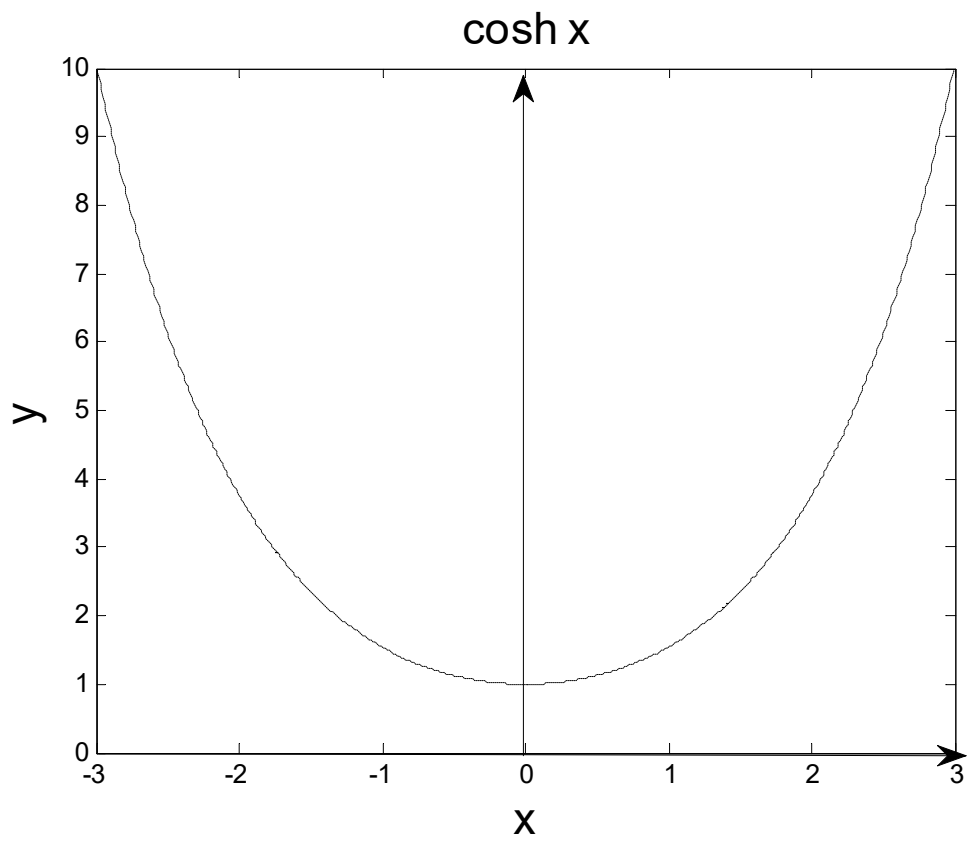
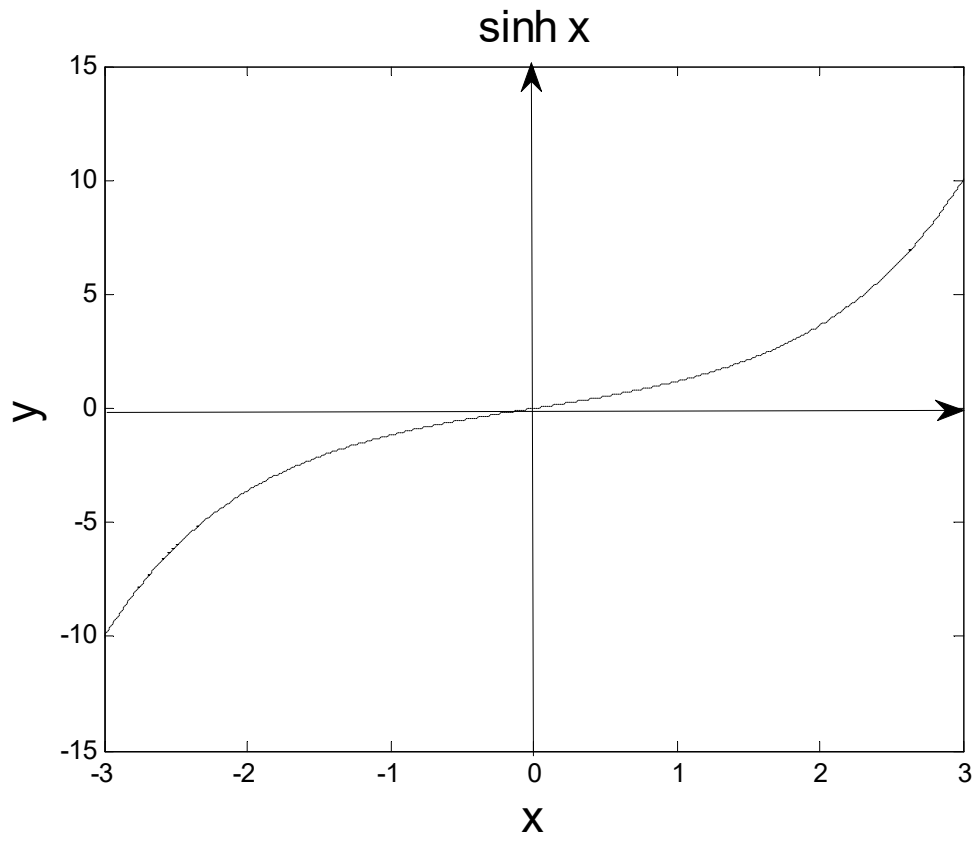
$$\tanh z = \frac{\sinh z}{\cosh z},$$

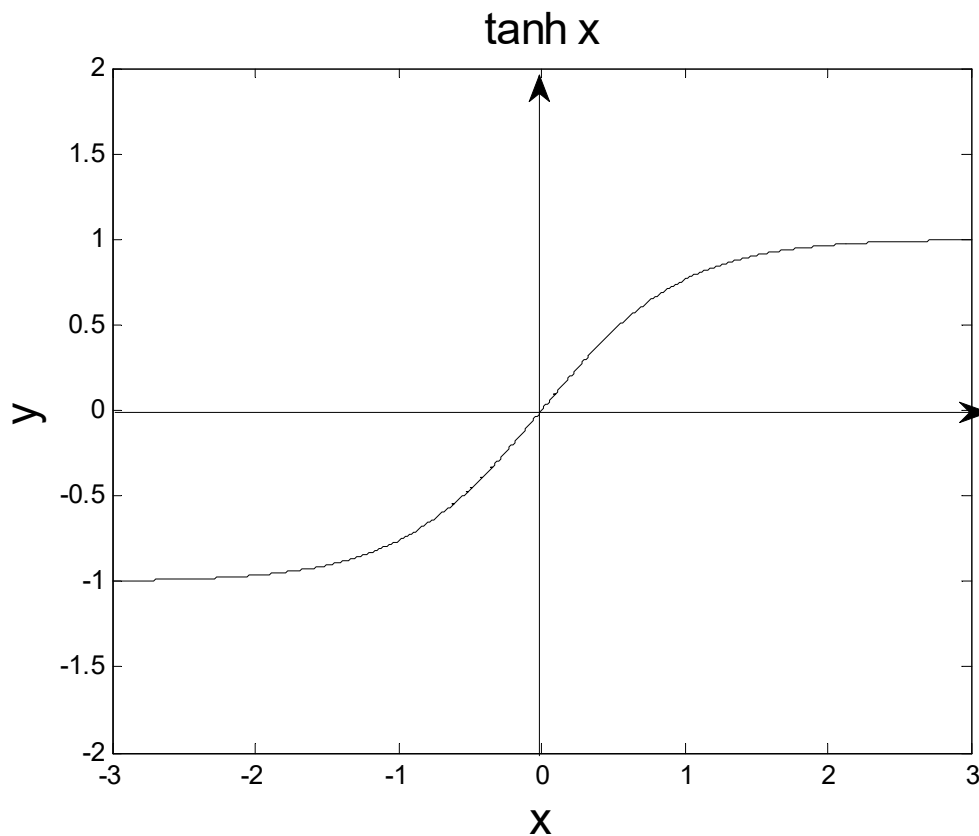
$$\coth z = \frac{1}{\tanh z},$$

$$\operatorname{sech} z = \frac{1}{\cosh z},$$

$$\operatorname{csch} z = \frac{1}{\sinh z}.$$

Οι γραφικές τους παραστάσεις είναι:





Μερικές βασικές σχέσεις που διέπουν τις υπερβολικές συναρτήσεις είναι:

$$\cosh^2 z - \sinh^2 z = 1$$

$$\tanh^2 z + \operatorname{sech}^2 z = 1$$

$$\operatorname{coth}^2 z - \operatorname{csch}^2 z = 1$$

$$\cosh z + \sinh z = e^z$$

$$\cosh z - \sinh z = e^{-z}$$

$$\sinh(-z) = -\sinh z$$

$$\cosh(-z) = \cosh z$$

$$\tanh(-z) = -\tanh z$$

$$\sinh \frac{z}{2} = \sqrt{\frac{\cosh z - 1}{2}}$$

$$\cosh \frac{z}{2} = \sqrt{\frac{\cosh z + 1}{2}}$$

καθώς και οι τύποι αθροισμάτων:

$$\sinh(z + u) = \cosh u \sinh z + \cosh z \sinh u$$

$$\cosh(z + u) = \cosh u \cosh z + \sinh u \sinh z$$

$$\tanh(z + u) = \frac{\tanh z + \tanh u}{1 + \tanh z \tanh u}$$

$$\operatorname{coth}(z + u) = \frac{\operatorname{coth} z \operatorname{coth} u + 1}{\operatorname{coth} z + \operatorname{coth} u}$$

Οι συναρτήσεις αυτές αναπτύσσονται σε σειρά κατά Taylor:

$$\sinh x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots \quad (|x| < \infty)$$

$$\cosh x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots \quad (|x| < \infty)$$

$$\tanh x = x - \frac{x^3}{3} + \frac{2x^5}{15} - \frac{17x^7}{315} + \frac{62x^9}{2835} - \frac{1382x^{11}}{155925} + \dots \quad (|x| < \frac{\pi}{2})$$

$$\coth x = \frac{1}{x} + \frac{x}{3} - \frac{x^3}{45} + \frac{2x^5}{945} - \frac{x^7}{4725} + \frac{2x^9}{93555} - \frac{1382x^{11}}{638512875} + \dots \quad (|x| < \pi)$$

Άλλες πιο ειδικές σχέσεις που ισχύουν για την συνάρτηση $\tanh x$, είναι

$$\tanh(x) = \coth(2x) - \operatorname{csch}(2x)$$

$$\tanh(x) = (-1 + \cosh(2x)) \operatorname{csch}(2x)$$

$$\tanh(x) = i \tan(m\pi - ix) \text{ για } m \in \mathbb{Z}$$

$$\tanh(x) = \frac{\sinh(2x)}{1 + \cosh(2x)}$$

$$\tanh(x) = \frac{\sqrt{x^2 - 1} \sqrt{-\tanh^2(x)}}{x}$$

$$\tanh(x) = \frac{2 \tanh\left(\frac{x}{2}\right)}{1 + \tanh^2\left(\frac{x}{2}\right)}$$

$$\tanh(x) = \frac{\operatorname{sech}\left(\frac{x}{3}\right) \sinh(x)}{-1 + 2 \cosh\left(\frac{2x}{3}\right)}$$

$$\tanh(x) = \frac{x \sqrt{-\tanh^2(x)}}{\sqrt{-x^2}} \text{ για } 2 |\operatorname{Im}(x)| < \pi$$

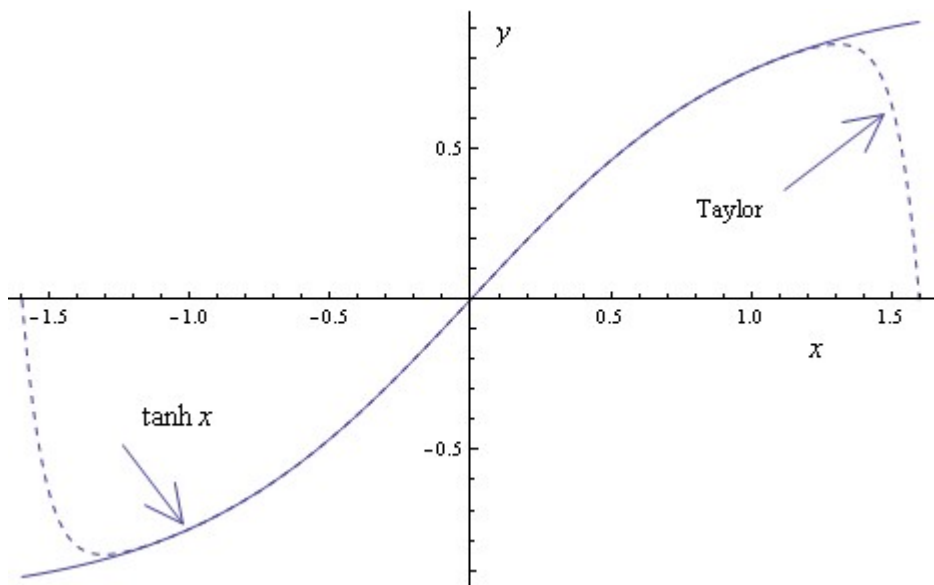
Επίσης παρατηρούμε ότι

$$\lim_{x \rightarrow \infty} \tanh x = 1 = - \lim_{x \rightarrow -\infty} \tanh x.$$

Όλες οι απλές ή ειδικές συναρτήσεις (π.χ. $\sin x$, $\cos x$, $\tan x$, e^x , $\log x$, $\tanh x$, κ.λ.π.) υπολογίζονται συνήθως με κάποια πολυωνμική τους προσέγγιση. Για την $\tanh x$ μια ακόμα πιο ακριβής πολυωνμική προσέγγιση είναι:

$$\begin{aligned} \tanh x \approx x - \frac{x^3}{3} + \frac{2x^5}{15} - \frac{17x^7}{315} + \frac{62x^9}{2835} - \frac{1382x^{11}}{155925} + \frac{21844x^{13}}{6081075} - \frac{929569x^{15}}{638512875} \\ + \frac{6404582x^{17}}{10854718875} - \frac{443861162x^{19}}{1856156927625}. \end{aligned}$$

Όμως όπως βλέπουμε στο παρακάτω γράφημα, αυτή η προσέγγιση είναι αποκλίνουσα για $|x| > \frac{\pi}{2}$.

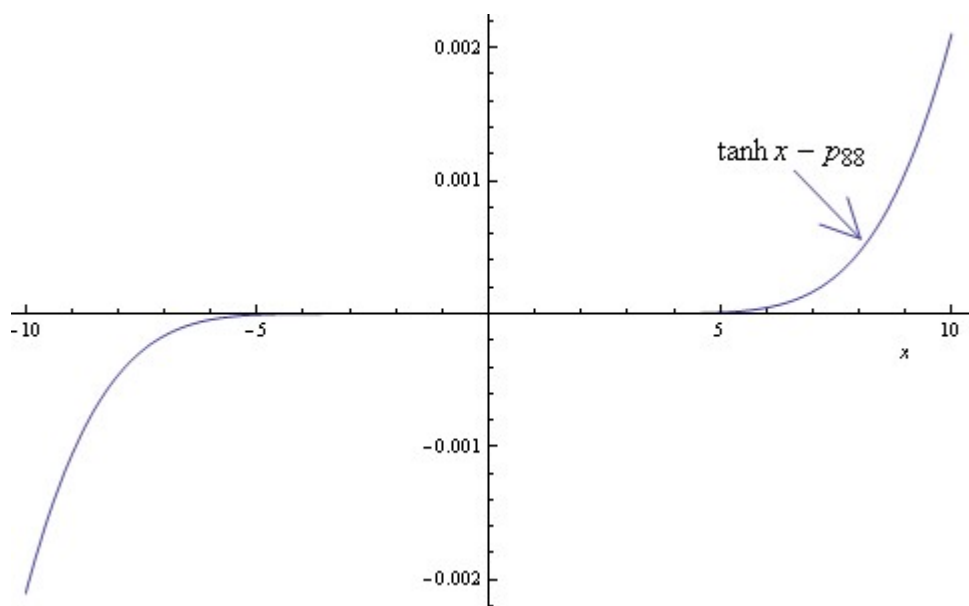


Γράφημα: Απόκλιση της $\tanh x$ από την προσέγγισή της κατά Taylor (19^{ου} βαθμού).

Το πρόβλημα αυτό μπορεί να αποφευχθεί χρησιμοποιώντας ρητές συναρτήσεις. Δηλαδή κλασματικές μορφές όπου ο αριθμητής και ο παρονομαστής είναι πολυώνυμα. Έτσι για παράδειγμα λαμβάνουμε την (8,8) ρητή προσέγγιση Pade:

$$p_{88} = \frac{x + \frac{2x^3}{15} + \frac{2x^5}{585} + \frac{4x^7}{225225}}{1 + \frac{7x^2}{15} + \frac{x^4}{39} + \frac{2x^6}{6435} + \frac{x^8}{2027025}}$$

Όμως και αυτές οι προσεγγίσεις (Pade) παρουσιάζουν μια σχετική απόκλιση μακριά από την αρχή των αξόνων όπως φαίνεται στο παρακάτω σχήμα.

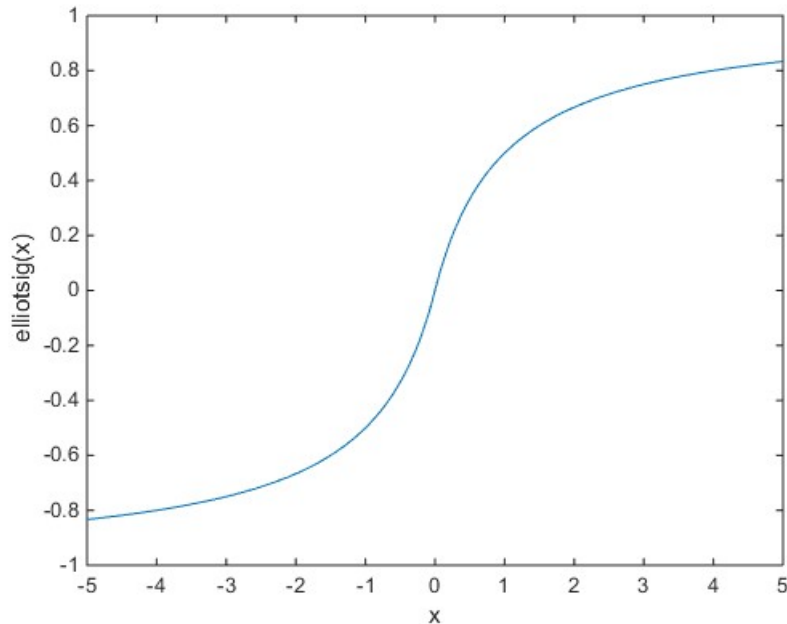


Γράφημα: Απόκλιση της $\tanh x$ από την προσέγγισή της κατά Pade (8^{ου} βαθμού).

Μια άλλη πρόταση δόθηκε στο [Ell93]. Εκεί χρησιμοποιήθηκε η συνάρτηση

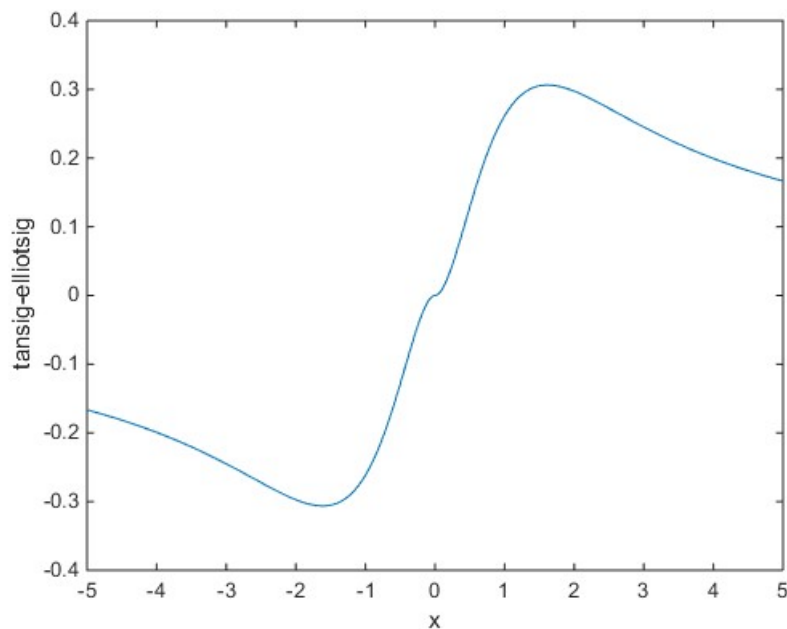
$$\frac{x}{1 + |x|}$$

η οποία υλοποιείται στο Matlab ως `elliotsig`. Ακολουθεί η γραφική της παράσταση.



Γράφημα: Απόκλιση της $\tanh x$ από την `elliotsig`.

Στη συνέχεια δίνουμε την γραφική παράσταση της απόκλισής της από την $\tanh x$.



Γράφημα: Απόκλιση της $\tanh x$ από την `elliotsig`.

Πρόκειται όπως φαίνεται για μια διαφορετική συνάρτηση μεταφοράς. Η απόσταση από την $\tanh x$ είναι μεγάλη. Ο σκοπός μας εδώ είναι όχι μια όποια προσέγγιση της $\tanh x$, αλλά μια φτηνή μεν προσέγγιση που να μιμείται όμως όσο το δυνατόν πιο πιστά την συμπεριφορά της όταν χρησιμοποιείται στα νευρωνικά δίκτυα.

Η λύση είναι τελικά η L_∞ τύπου προσεγγίσεις. Έτσι λαμβάνουμε :

$$\tanh x \approx x + x \cdot \frac{x^2((-0.96437492777225470x^2 - 99.225929672236083)x^2 - 1613.4119023996228)}{((x^2 + 112.74474380534949)x^2 + 2233.7720718962313)x^2 + 4840.2357071988689}$$

για $x \in [\sqrt{\varepsilon}, 0.5 \cdot \ln 1.5]$ με ε την ακρίβεια αριθμητικής του υπολογιστή (double precision). Για το Matlab $\varepsilon = 2.22 \cdot 10^{-16}$ περίπου.

Για $x < \sqrt{\varepsilon}$ αρκεί να λάβουμε $\tanh x \approx x$.

Όταν $x > 0.5 \cdot \ln 1.5$, ο υπολογισμός γίνεται μέσω του τύπου

$$\tanh x \approx \frac{2}{e^{-2x} + 1} - 1.$$

Ο παραπάνω τύπος υπολογίζεται με τη σειρά του μέσω κάποιας ρητής συνάρτησης – προσέγγισης του e^x . Όμως και αυτή η προσέγγιση απαιτεί τον υπολογισμό του $\exp(-2x)$ και επιβαρύνεται με μια διαίρεση. Άλλωστε και το $\exp x$ για να υπολογισθεί απαιτούνται πάλι 7 πολλαπλασιασμοί και μια διαίρεση. Έτσι έχουμε

$$\exp x \approx 1 - \frac{\psi x}{\psi - 2} + x,$$

με $t = x^2$,

$$\psi = x - t \cdot \left(p_1 + t \cdot \left(p_2 + t \cdot \left(p_3 + t \cdot \left(p_4 + t \cdot p_5 \right) \right) \right) \right),$$

και

$$\begin{aligned} p_1 &= 0.166666666666666602, p_2 = -0.00277777777777015593, \\ p_3 &= 0.000066137563214379344, p_4 = -0.0000016533902205465251, \\ p_5 &= 4.138136797057238 \times 10^{-8}. \end{aligned}$$

Επίσης τα ακριβώς παραπάνω (για την e^x) ισχύουν όταν $0 \leq x \leq \ln 2$. Αλλιώς βρίσκουμε

$$|\tilde{x}| \leq 0.5 \ln 2,$$

έτσι ώστε $\exp x = \exp \tilde{x} \cdot 2^N$. Προσεγγίζουμε το $\exp \tilde{x}$ και μετά το $\exp x$.

Για περισσότερες πληροφορίες δείτε [Netlib].

Συνάρτηση Matlab για τον υπολογισμό της $\tanh x$

Παρακάτω δίνουμε τη συνάρτηση `tanh` στο Matlab για τον υπολογισμό της $\tanh x$ που βασίζεται στον αμέσως παραπάνω αλγόριθμο και υλοποιείται στο εν λόγω λογισμικό τη δεκαετία το 90.

```

function y = tanhh(xin)
%TANH Hyperbolic tangent.
%   TANH(X) is the hyperbolic tangent of the elements of X.

%   C. Moler, 6-17-92
%   Copyright (c) 1984-94 by The MathWorks, Inc.

%   21 Oct 92 (jmh) -- fixed for matrix inputs
%   2 Sep 93 (jmh) -- fixed for null matrix input

% Complex argument
if any(any(imag(xin))) ~= 0           % 2 Sep 93 (jmh)
    tr = tanh(real(xin));
    ti = i*tan(imag(xin));
    y = (tr + ti)./(1 + tr.*ti);

% Real argument
else

    % Reference: W. J. Cody and W. Waite, "Software Manual
    % for the Elementary Functions", 1980, chapter 13.

    xbig = 18.715;                    % 0.5*log(4/eps)
    xmid = 0.5493061443340549;        % log(3)/2
    xsmall = 1.49e-8;                % sqrt(eps)
    x = abs(xin);
    y = zeros(size(x));

    % tanh(x) == 1 to working precision
    k = xbig <= x;
    if any(any(k))
        y(k) = ones(size(find(k)));
    end

    % 1/2 < tanh(x) < 1
    k = (xmid < x) & (x < xbig);
    if any(any(k))                   % 21 Oct 92 (jmh)
        y(k) = 1 - 2./(exp(2*x(k))+1);
    end

    % x < tanh(x) <= 1/2
    k = (xsmall < x) & (x <= xmid);
    if any(any(k))
        p = [-0.16134119023996228e4 -0.99225929672236083e2 ...
              -0.96437492777225470];
        q = [0.48402357071988689e4 0.22337720718962313e4 ...
              0.11274474380534949e3];
        xx = x(k).^2;
        y(k) = x(k) + x(k).*(xx.*((p(3)*xx + p(2)).*xx + p(1))) ./ ...
              (((xx + q(3)).*xx + q(2)).*xx + q(1)));
    end

    % tanh(x) == x to working precision
    k = (x <= xsmall) | isnan(x);
    if any(any(k))
        y(k) = x(k);
    end

    y = sign(xin).*y;
end

```

4.2 Αποτελεσματικά “ανακριβής” υπολογισμός της $\tanh x$

Όπως παρατηρούμε για τον υπολογισμό της $\tanh x$ απαιτούνται 7 πολλαπλασιασμοί και μια διαίρεση. Έτσι είναι δυνατή η επίτευξη ακρίβειας 16 δεκαδικών ψηφίων.

Είναι όμως χρήσιμη μια τέτοια απαίτηση όταν η $\tanh x$ υλοποιείται σε ένα νευρωνικό δίκτυο; Μήπως κάποια γρηγορότερη προσέγγιση με μικρότερη ακρίβεια είναι αρκετή;

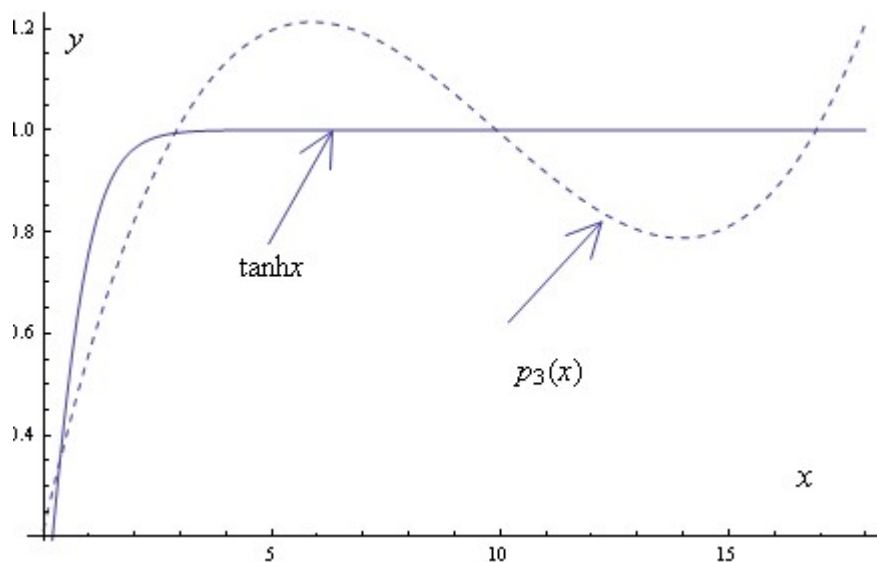
Εδώ θα επικεντρωθούμε σε κυβικά πολυώνυμα της μορφής

$$p(x) = dx^3 + cx^2 + bx + a.$$

Αν θεωρήσουμε ότι $\tanh x = 1$ για $|x| > 18$, τότε το καλύτερο που μπορεί να πετύχουμε στο διάστημα $[0,18]$ είναι:

$$p_3(x) = 0.212358 + 0.395907x - 0.0478683x^2 + 0.00160888x^3.$$

Παρακάτω δίνουμε την γραφική παράσταση της p_3 σε σχέση με την $\tanh x$.



Γράφημα: Απόκλιση της $\tanh x$ από την προσέγγιση p_3 .

Η μέγιστη απόκλιση της p_3 είναι περίπου 0.212708 κάτι που είναι μάλλον απαράδεκτο.

Παρατηρούμε επίσης ότι υπάρχουν διαστήματα όπου $p_3 > 1$.

Σύμφωνα με τον [Mu16] όταν κατασκευάζουμε μια προσέγγιση συνάρτησης θα πρέπει να λαμβάνουμε υπόψη:

- Την ταχύτητα υπολογισμού (το κύριό μας ενδιαφέρον εδώ)
- Την ακρίβεια (με μικρότερο ενδιαφέρον εδώ)
- Λογική χρήση των πόρων (π.χ. λογική χρήση της μνήμης)
- Διατήρηση των βασικών μαθηματικών ιδιοτήτων (π.χ. μονοτονία, συνέχεια, συμμετρία)
- Διατήρηση της φοράς της στρογγυλοποίησης
- Διατήρηση των ορίων: π.χ. $\tanh x > 1$, είναι μη αποδεκτό έστω και σε απειροελάχιστο ποσό.

Αφού δεν μπορούμε να χρησιμοποιήσουμε ένα πολυώνυμο τρίτου βαθμού σε όλο το διάστημα μπορούμε να χρησιμοποιήσουμε τμηματικά πολυώνυμα 3^{ου} βαθμού σε ένα πλήθος υποδιαστημάτων. Οι κυβικές splines είναι μια προφανής λύση τότε.

Όμως σε ποια σημεία πρέπει να στηριχθούν οι προσεγγίσεις με splines:

Θα ξεκινήσουμε με στήριξη σε 4 σημεία το $n_0 = 0, n_1, n_2$ και το $n_3 = 18$. Για $x > 18$ λαμβάνουμε $\tanh x = 1$, και αφού $\tanh(-x) = -\tanh x$ περιοριζόμαστε στο διάστημα $[0,18]$. Το ερώτημα είναι $n_1 = ?, n_2 = ?$, έτσι ώστε να ικανοποιούνται τα κριτήρια που θέσαμε παραπάνω.

Για να επιλεγούν τα σημεία αυτά θα παραγάγουμε τυχαία ζεύγη στο $[0,18]$ και θα ελέγχουμε την μονοτονία και την διατήρηση της προσέγγισης στο $[0,1]$. Η συνέχεια και η συμμετρία διατηρούνται εκ κατασκευής. Η ταχύτητα είναι δεδομένη. Η ακρίβεια είναι δευτερευούσης σημασίας αλλά εντός συγκεκριμένων πλαισίων. Η χρήση μνήμης είναι ίδια με το πρόγραμμα του Moller.

Στο παρακάτω πρόγραμμα κάνουμε τον έλεγχο σε 100000 τυχαία ζεύγη. Ξεκινάμε με ένα γρήγορο έλεγχο 180 σημείων στο $[0,1]$. Όσα ζεύγη προκριθούν ελέγχονται πλέον ενδελεχώς ως προς τα δυο κριτήρια.

```
for j1=1:100000,
    n=sort(rand(1,2))*18;
    x=[0 n 18];
    y=tanh(x);
    n=sort(rand(1,2))*18;
    cs4=spline(x,[1 y 0]);
    yy=ppval(cs4,0:1e-2:18);
    if max(abs(yy))<=1,
        yy=ppval(cs4,0:1e-5:18);
        if max(abs(yy))<=1 && min(yy(2:end)-yy(1:end-1))>=0,
            fprintf('%26.15f %26.15f \n',n(1),n(2));
        end;
    end;
end;
```

Αφού $\tanh' 0 = 1$ και $\tanh' 18 \approx 0$, χρησιμοποιούμε τη εντολή με

```
cs4=spline(x,[1 y 0]);
```

Παρήχθησαν έτσι δεκάδες ζεύγη.

1.932471244114961	2.798547845316649
0.309289950841608	2.617040870663818
1.724186567337780	2.661921260715057
2.183625852423822	3.007470358057990
2.067234722026224	2.906468532186573
1.264003763097397	2.443819860432835
2.716768984443669	3.796159128731798
0.924405878254698	2.387898616542549
2.475727777142676	3.339662371510555
1.015070317490389	2.399050892763512
2.243551796260792	3.065017972607904
2.277068298206905	3.092511965193564
2.484125881766001	3.346291949517931
1.497784780176345	2.535742787395212

κ.λ.π.

Από αυτά επιλέξαμε ένα ζεύγος που δίνει τη σχετικά μικρότερη απόκλιση. Δηλαδή,

$$n_1 = 0.371025186672900, n_2 = 2.572153900248530,$$

που μας δίνει τα πολυώνυμα:

$$S_0 = x + 0.019872193438979585x^2 - 0.36950760861254955x^3,$$

στο διάστημα $[0, 0.371025186672900]$.

$$S_1 = 0.35488810724962294 + 0.862147260944915(-0.3710251866729 + x) \\ - 0.3914176949486047(-0.3710251866729 + x)^2 \\ + 0.05928356367224765(-0.3710251866729 + x)^3$$

στο διάστημα $[0.371025186672900, 2.572153900248530]$. Και

$$S_2 = 0.9884026213740197 + 0.0007066442941002771(-2.57215390024853 + x) \\ + 0.000054567777615618335(-2.57215390024853 + x)^2 \\ - 0.000003347599023062611(-2.57215390024853 + x)^3$$

στο διάστημα $[2.572153900248530, 18]$.

Ονομάζουμε πλέον μόνο την προσέγγιση αυτή ως CS_4 .

Η εύρεση των πολυωνύμων μπορεί να γίνει είτε με Matlab γράφοντας στο command window:

```
>> x=[0 0.371025186672900,2.572153900248530 18];
>> y=tanh(x);
```

```
>> cs4=spline(x,[1 y 0]);
>> cs4.coefs
ans =
-0.369507608612549  0.019872193438979  1.000000000000000  0
 0.059283563672248 -0.391417694948604  0.862147260944915  0.354888107249623
-0.000003347599023  0.000054567777616  0.000706644294101  0.988402621374020
```

Είτε με Mathematica γράφοντας

```
In[2]:= data=Transpose[{{0,0.371025186672900`,2.572153900248530`,18},
      Tanh[{0,0.371025186672900`,2.572153900248530`,18}]]]
In[3]:= CubSpline[data,1,0,a,b,c,d,m,tau];
In[4]:= For[i=1,i<m,i++,Poly[i,x_]=a[[i]]+b[[i]]*(x-tau[[i]])
      +c[[i]]*(x-tau[[i]])^2+d[[i]]*(x-tau[[i]])^3]
In[5]:= N[Poly[1,x],16]
Out[5]= 1. x+0.0198722 x^2-0.369508 x^3
In[6]:= N[Poly[2,x],16]
Out[6]= 0.354888 +0.862147 (-0.371025+x)-0.391418 (-0.371025+x)^2
      +0.0592836 (-0.371025+x)^3
In[7]:= N[Poly[3,x],16]
Out[7]= 0.988403 +0.000706644 (-2.57215+x)+0.0000545678 (-2.57215+x)^2
      -3.3476*10^-6 (-2.57215+x)^3
```

Η μονοτονία μπορεί να ελεγχθεί μέσω των παραγώγων. Όντως

$$s'_0 = 1 + 0.03974438687795917x - 1.1085228258376487x^2,$$

με ρίζες

$$x = -0.9320326202140093 || x = 0.9678860875054262,$$

και κατά συνέπεια $s'_0 > 0$ εντός των ριζών (ο συντελεστής του x στο s_0 είναι αρνητικός), άρα και στο διάστημα $[0, n_1]$.

Όμοια

$$s'_1 = 0.862147260944915 - 0.7828353898972094(-0.3710251866729 + x) \\ + 0.17785069101674295(-0.3710251866729 + x)^2,$$

που έχει μιγαδικές ρίζες και είναι μόνιμα θετικό. Τέλος

$$s'_2 = 0.0007066442941002771$$

$$+0.00010913555523123667 \cdot (-2.57215390024853 + x) \\ - 0.000010042797069187833(-2.57215390024853 + x)^2,$$

με ρίζες

$$x = -1.9886444557947718 \quad || \quad x = 18,$$

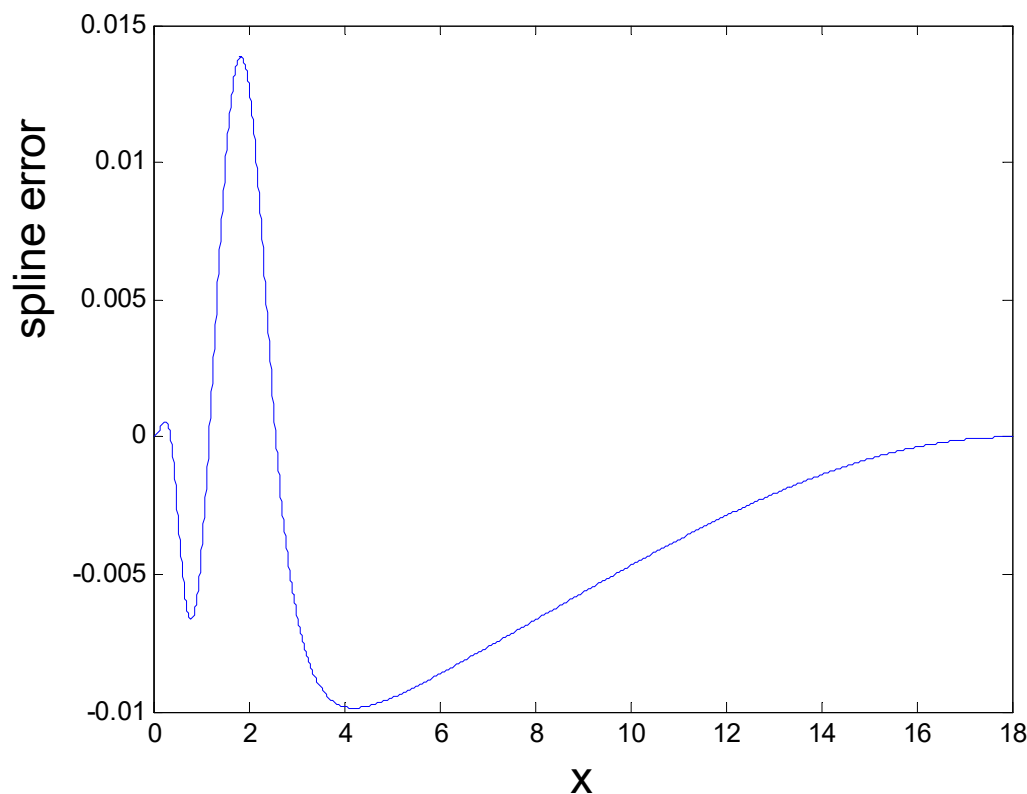
και κατά συνέπεια $s'_2 > 0$ εντός των ριζών (ο συντελεστής του x στο s_2 είναι αρνητικός), άρα και στο διάστημα $[n_2, n_3]$.

Επίσης έχουμε $s_2(18) = 1$. Δηλαδή σε μια αύξουσα συνάρτηση η δεξιότερη τιμή είναι 1. Αυτή είναι και η μεγαλύτερη τιμή που επιτυγχάνει η cs_4 .

Το μέγιστο σφάλμα που παρατηρήθηκε ήταν.

$$|cs_4 - \tanh x| \approx 0.013852304438455,$$

ενώ το γράφημα του σφάλματος δίνεται στο επόμενο σχήμα. Είναι μια συζητήσιμη ακρίβεια η οποία θα ελεγχθεί στην εφαρμογή της στα νευρωνικά δίκτυα.



Γράφημα: Απόκλιση της $\tanh x$ από την προσέγγιση με splines σε 4 σημεία (cs_4).

Στο παρακάτω πρόγραμμα περιγράφουμε τον σχετικό αλγόριθμο υπολογισμού της υπερβολικής εφαπτομένης μέσω των κυβικών spline σε 4 υποδιαστήματα.

```

function y = cs4(xin,param);
% TANH      Hyperbolic tangent.
% using splines

x=abs(xin);
y = zeros(size(x));

k = 18 <= x;
if any(any(k)),
    y(k) = ones(size(find(k)));
end

k = x <= 0.371025186672900;
if any(any(k)),
    xx = x(k);
    y(k) = (((-3.695076086125492e-001*xx ...
              +1.987219343897867e-002).*xx+1).*xx);
end

k = (0.371025186672900 < x) & (x <= 2.572153900248530);
if any(any(k)),
    xx = x(k)-0.371025186672900;
    y(k) = (((5.928356367224758e-002*xx
              -3.914176949486042e-001).*xx+ ...
              8.621472609449146e-001).*xx+ ...
              3.548881072496229e-001);
end

k = (2.572153900248530 < x) & (x <= 18);
if any(any(k)),
    xx = x(k)-2.572153900248530;
    y(k) = (((-3.347599023061577e-006*xx+ ...
              5.456777761558641e-005).*xx+ ...
              7.066442941005233e-004).*xx+ ...
              9.884026213740197e-001);
end

y = sign(xin).*y;
return

```

Οι συγκρίσεις με χρόνους είναι συνήθως παραπλανητικές διότι βασίζονται πολύ στον τρόπο προγραμματισμού και όχι στον καθαυτό αλγόριθμο. Συγκρίσεις με build in συναρτήσεις (π.χ. `sin`, `cos`, `exp`, κ.λπ.) του MATLAB δεν έχουν νόημα καθότι αυτές είναι μεταφρασμένες από κώδικες σε C ή FORTRAN και είναι σε εκτελέσιμη μορφή. Έτσι το να

επιχειρήσει να τις ανταγωνιστεί κανείς με m-αρχεία τα οποία εκτελούνται με interpreter είναι άσκοπο. Γι' αυτό συγκρίνουμε χρόνους με την tanhh που είναι γραμμένη σε m-αρχείο.

Δυο τυχαία διανύσματα καταδεικνύουν ένα υποδιπλασιασμό του χρόνου εκτέλεσης.

```
>> r=rand(2000,2000)*log(3)/2;
>> tic;tanh(r);toc
Elapsed time is 1.588258 seconds.
>> tic;cs4(r);toc
Elapsed time is 0.805776 seconds.
```

```
>> r=randn(2000000,1)*log(3)/2;tic;tanh(r);toc
Elapsed time is 0.968602 seconds.
>> tic;tanhsp13(r);toc
Elapsed time is 0.467059 seconds.
```

Σημειώνουμε ότι περιορίσαμε τις εισόδους στο $x \leq 0.5 \cdot \ln 2$, για να αποφύγουμε τον υπολογισμό της tanhh μέσω της build-in συνάρτησης exp.

4.3 Προσέγγιση της tanh x σε 36 ισομήκη υποδιαστήματα

Ένα ενδιαφέρον αποτέλεσμα προέκυψε από την επιλογή των σημείων

$$n_0 = 0, n_1 = 0.5, n_2 = 1, n_3 = 1.5, \dots, n_{34} = 17, n_{35} = 17.5, n_{36} = 18.$$

Έτσι το κάθε s_ξ ορισμένο στο διάστημα $[0.5\xi, 0.5\xi + 0.5]$, είναι τα πολυώνυμα:

$$s_0 = 1.0x - 0.02798073870433028x^2 - 0.2471012645112613x^3,$$

$$s_1 = 0.4621171572600097 + 0.7866933129122237(-0.5 + x) \\ - 0.3986326354712223(-0.5 + x)^2 \\ + 0.0463080088595909(-0.5 + x)^3,$$

$$s_2 = 0.7615941559557649 + 0.4227916840856945(-1.0 + x) \\ - 0.3291706221818359(-1.0 + x)^2 \\ + 0.1156072895337061(-1.0 + x)^3,$$

$$s_3 = 0.9051482536448665 + 0.1803265290541381(-1.5 + x) \\ - 0.1557596878812767(-1.5 + x)^2 \\ + 0.06124787099360439(-1.5 + x)^3,$$

$$s_4 = 0.9640275800758169 + 0.0705027444180647(-2.0 + x) \\ - 0.06388788139087019(-2.0 + x)^2 \\ + 0.02645852971438882(-2.0 + x)^3,$$

$$s_5 = 0.9866142981514303 + 0.02645876031298612(-2.5 + x) \\ - 0.02420008681928695(-2.5 + x)^2 \\ + 0.01008877666903069(-2.5 + x)^3,$$

$$\begin{aligned}
s_6 &= 0.9950547536867305 + 0.009825255995472195(-3.0 + x) \\
&\quad - 0.009066921815740916(-3.0 + x)^2 \\
&\quad + 0.003817971045339117(-3.0 + x)^3, \\
s_7 &= 0.9981778976111987 + 0.003621812463735616(-3.5 + x) \\
&\quad - 0.003339965247732239(-3.5 + x)^2 \\
&\quad + 0.001403897663468688(-3.5 + x)^3, \\
s_8 &= 0.9993292997390671 + 0.001334770463604893(-4.0 + x) \\
&\quad - 0.001234118752529207(-4.0 + x)^2 \\
&\quad + 0.0005204445223227824(-4.0 + x)^3, \\
s_9 &= 0.9997532108480276 + 0.0004909851028177732(-4.5 + x) \\
&\quad - 0.0004534519690450334(-4.5 + x)^2 \\
&\quad + 0.0001909108433597315(-4.5 + x)^3, \\
s_{10} &= 0.9999092042625952 + 0.0001807162662925383(-5.0 + x) \\
&\quad - 0.0001670857040054361(-5.0 + x)^2 \\
&\quad + 0.00007044949251014629(-5.0 + x)^3, \\
s_{11} &= 0.9999665971563038 + 0.00006646768166971191(-5.5 + x) \\
&\quad - 0.00006141146524021675(-5.5 + x)^2 \\
&\quad + 0.00002586815973567327(-5.5 + x)^3, \\
s_{12} &= 0.9999877116507956 + 0.00002445733623125011(-6.0 + x) \\
&\quad - 0.00002260922563670683(-6.0 + x)^2 \\
&\quad + 0.000009530711217391504(-6.0 + x)^3, \\
s_{13} &= 0.9999954793514042 + 0.000008996144007586908(-6.5 + x) \\
&\quad - 0.00000831315881061958(-6.5 + x)^2 \\
&\quad + 0.000003502481914722605(-6.5 + x)^3, \\
s_{14} &= 0.9999983369439447 + 0.000003309846633009282(-7.0 + x) \\
&\quad - 0.000003059435938535671(-7.0 + x)^2 \\
&\quad + 0.000001289498156850333(-7.0 + x)^3, \\
s_{15} &= 0.9999993881955462 + 0.00000121753431211136(-7.5 + x) \\
&\quad - 0.000001125188703260171(-7.5 + x)^2 \\
&\quad + 4.741131960033753 \times 10^{-7}(-7.5 + x)^3, \\
s_{16} &= 0.9999997749296759 + 4.47930505853721 \times 10^{-7}(-8.0 + x) \\
&\quad - 4.140189092551082 \times 10^{-7}(-8.0 + x)^2 \\
&\quad + 1.744883803937592 \times 10^{-7}(-8.0 + x)^3,
\end{aligned}$$

$$s_{17} = 0.9999999172012491 + 1.647778818939322 \times 10^{-7}(-8.5 + x) \\ - 1.522863386644693 \times 10^{-7}(-8.5 + x)^2 \\ + 6.417148513213684 \times 10^{-8}(-8.5 + x)^3,$$

$$s_{18} = 0.999999969540041 + 6.06201570785656 \times 10^{-8}(-9.0 + x) \\ - 5.602911096626404 \times 10^{-8}(-9.0 + x)^2 \\ + 2.361252332409624 \times 10^{-8}(-9.0 + x)^3,$$

$$s_{19} = 0.9999999887944072 + 2.230043860537374 \times 10^{-8}(-9.5 + x) \\ - 2.061032598011967 \times 10^{-8}(-9.5 + x)^2 \\ + 8.685182146036171 \times 10^{-9}(-9.5 + x)^3,$$

$$s_{20} = 0.9999999958776928 + 8.203999234781193 \times 10^{-9}(-10.0 + x) \\ - 7.582552761065417 \times 10^{-9}(-10.0 + x)^2 \\ + 3.1954697985895 \times 10^{-9}(-10.0 + x)^3,$$

$$s_{21} = 0.9999999984834879 + 3.018048822657902 \times 10^{-9}(-10.5 + x) \\ - 2.789348063181166 \times 10^{-9}(-10.5 + x)^2 \\ + 1.17544856384524 \times 10^{-9}(-10.5 + x)^3,$$

$$s_{22} = 0.9999999994421064 + 1.110287182360666 \times 10^{-9}(-11.0 + x) \\ - 1.026175217413305 \times 10^{-9}(-11.0 + x)^2 \\ + 4.324499118254322 \times 10^{-10}(-11.0 + x)^3,$$

$$s_{23} = 0.9999999997947624 + 4.084493988164348 \times 10^{-10}(-11.5 + x) \\ - 3.775003496751572 \times 10^{-10}(-11.5 + x)^2 \\ + 1.590823179265347 \times 10^{-10}(-11.5 + x)^3,$$

$$s_{24} = 0.999999999244973 + 1.502607875861786 \times 10^{-10}(-12.0 + x) \\ - 1.388768727853551 \times 10^{-10}(-12.0 + x)^2 \\ + 5.85250204515104 \times 10^{-11}(-12.0 + x)^3,$$

$$s_{25} = 0.999999999722241 + 5.52776801394563 \times 10^{-11}(-12.5 + x) \\ - 5.108934210808953 \times 10^{-11}(-12.5 + x)^2 \\ + 2.152964104609662 \times 10^{-11}(-12.5 + x)^3,$$

$$s_{26} = 0.999999999897818 + 2.033556881593924 \times 10^{-11}(-13.0 + x) \\ - 1.879488053894459 \times 10^{-11}(-13.0 + x)^2 \\ + 7.920449198158969 \times 10^{-11}(-13.0 + x)^3,$$

$$s_{27} = 0.99999999996241 + 7.481025175613872 \times 10^{-12}(-13.5 + x) \\ - 6.914206741706142 \times 10^{-12}(-13.5 + x)^2 \\ + 2.913733674428932 \times 10^{-12}(-13.5 + x)^3,$$

$$s_{28} = 0.9999999999986171 + 2.752118689729429 \times 10^{-1} (-14.0 + x) \\ - 2.543606230062743 \times 10^{-1} (-14.0 + x)^2 \\ + 1.071912836502405 \times 10^{-12} (-14.0 + x)^3,$$

$$s_{29} = 0.9999999999994913 + 1.012447087043489 \times 10^{-1} (-14.5 + x) \\ - 9.357369753091357 \times 10^{-1} (-14.5 + x)^2 \\ + 3.943309632320671 \times 10^{-1} (-14.5 + x)^3,$$

$$s_{30} = 0.9999999999998129 + 3.724583341584042 \times 10^{-13} (-15.0 + x) \\ - 3.442405304610348 \times 10^{-13} (-15.0 + x)^2 \\ + 1.450710619475633 \times 10^{-13} (-15.0 + x)^3,$$

$$s_{31} = 0.999999999999312 + 1.370211001580419 \times 10^{-13} (-15.5 + x) \\ - 1.266339375396898 \times 10^{-1} (-15.5 + x)^2 \\ + 5.3353163016885 \times 10^{-14} (-15.5 + x)^3,$$

$$s_{32} = 0.999999999999747 + 5.040203488101585 \times 10^{-1} (-16.0 + x) \\ - 4.66041930143623 \times 10^{-14} (-16.0 + x)^2 \\ + 1.968471696851141 \times 10^{-14} (-16.0 + x)^3,$$

$$s_{33} = 0.999999999999907 + 1.856137959303711 \times 10^{-14} (-16.5 + x) \\ - 1.707711756159518 \times 10^{-14} (-16.5 + x)^2 \\ + 7.028360168023748 \times 10^{-15} (-16.5 + x)^3,$$

$$s_{34} = 0.999999999999966 + 6.755532157459747 \times 10^{-15} (-17.0 + x) \\ - 6.534577309559557 \times 10^{-15} (-17.0 + x)^2 \\ + 3.38137407771711 \times 10^{-15} (-17.0 + x)^3,$$

$$s_{35} = 0.999999999999988 + 2.756985406188022 \times 10^{-15} (-17.5 + x) \\ - 1.462516192983891 \times 10^{-15} (-17.5 + x)^2 \\ - 1.725958950938842 \times 10^{-15} (-17.5 + x)^3.$$

Ονομάζουμε αυτή την προσέγγιση `cs36` και ακολουθεί η υλοποίησή της σε Matlab.

```
function y = cs36(xin,param);
%TANH Hyperbolic tangent.
% using splines

[n1,n2]=size(xin);
xin=reshape(xin,n1*n2,1);
s=sign(xin);
xin=abs(xin);
ln=length(xin);
```

```

pp= [
-0.247101264511261357 -0.02798073870433029 1.0000000000000000 0.
0.04630800885959091 -0.39863263547122232 0.786693312912223695 0.46211715726000976
0.11560728953370613 -0.32917062218183596 0.42279168408569455 0.76159415595576489
0.06124787099360439 -0.155759687881276775 0.180326529054138182 0.90514825364486644
0.02645852971438883 -0.06388788139087019 0.07050274441806470 0.96402758007581688
0.01008877666903069 -0.024200086819286951 0.02645876031298613 0.98661429815143029
0.00381797104533912 -0.00906692181574092 0.00982525599547219 0.99505475368673045
0.00140389766346869 -0.00333996524773224 0.00362181246373562 0.99817789761119871
0.00052044452232278 -0.001234118752529207 0.00133477046360489 0.99932929973906704
0.00019091084335973 -0.000453451969045034 0.00049098510281777 0.99975321084802754
0.00007044949251015 -0.00016708570400544 0.00018071626629254 0.999909204262595130
0.000025868159735673 -0.000061411465240216 0.00006646768166971 0.999966597156303810
9.53071121739e-6 -0.00002260922563671 0.000024457336231251 0.999987711650795570
3.50248191472e-6 -8.313158810619e-6 8.99614400759e-6 0.99999547935140419
1.28949815685e-6 -3.059435938535e-6 3.30984663301e-6 0.99999833694394467
4.7411319600e-7 -1.12518870326e-6 1.21753431211e-6 0.999999388195546148
1.7448838039e-7 -4.1401890926e-7 4.4793050585e-7 0.99999977492967589
6.417148513e-8 -1.5228633866e-7 1.6477788189e-7 0.99999991720124905
2.361252332e-8 -5.602911097e-8 6.062015708e-8 0.99999996954004097
8.68518215e-9 -2.061032598e-8 2.230043861e-8 0.99999998879440719
3.19546980e-9 -7.58255276e-9 8.20399923e-9 0.99999999587769276
1.17544856e-9 -2.789348063e-9 3.01804882e-9 0.99999999848348792
4.3244991e-10 -1.02617522e-9 1.11028718e-9 0.999999999442106381
1.5908232e-10 -3.7750035e-10 4.0844940e-10 0.999999999794762406
5.852502e-11 -1.3887687e-10 1.5026079e-10 0.999999999924497308
2.152964e-11 -5.108934e-11 5.527768e-11 0.999999999972224111
7.92045e-12 -1.879488e-11 2.033557e-11 0.999999999989781822
2.91373e-12 -6.91421e-12 7.48103e-12 0.999999999996240942
1.07191e-12 -2.54361e-12 2.75212e-12 0.999999999998617120
3.9433e-13 -9.3574e-13 1.01245e-12 0.999999999999491267
1.4507e-13 -3.4424e-13 3.7246e-13 0.999999999999812848
5.335e-14 -1.2663e-13 1.3702e-13 0.999999999999931150
1.968e-14 -4.660e-14 5.040e-14 0.999999999999974672
7.03e-15 -1.708e-14 1.856e-14 0.999999999999990682
3.38e-15 -6.53e-15 6.76e-15 0.99999999999999657
-1.73e-15 -1.46e-15 2.76e-15 0.99999999999999874
0 0 0 1];

seira=fix(xin*2)+1;
seira=seira.*(seira<=36)+(seira>36)*37;

```

```

pps=pp(seira, :);
xin=xin-seira/2+1/2;
y=s.*((pps(:,1).*xin+pps(:,2)).*xin+pps(:,3)).*xin+pps(:,4).*ones(ln,1));
y=reshape(y,n1,n2);
return

```

Μπορούμε να αποδείξουμε την μονοτονία σε κάθε υποδιάστημα. Πράγματι στο 1^ο υποδιάστημα έχουμε

$$s'_0 = 1.0 - 0.055961477408660574x - 0.74130379353378407x^2,$$

με ρίζες

$$x = -1.1998121390193982 || x = 1.1243215265379848,$$

και $s'_0 > 0$ εντός των ριζών (ο συντελεστής του είναι αρνητικός) και άρα και του διαστήματος $[0,0.5]$. Δηλαδή s_0 αύξουσα στο διάστημα ενδιαφέροντος.

Όμοια

$$s'_1 = 0.7866933129122237 - 0.79726527094244465(-0.5 + x) + 0.13892402657877272(-0.5 + x)^2,$$

με ρίζες

$$x == 1.766037703426636 || x == 4.972820365810071,$$

και $s'_1 > 0$ εκτός των ριζών (ο συντελεστής του είναι θετικός) και άρα και του διαστήματος $[0.5,1]$. Συνεχίζοντας έτσι μπορούμε να αποδείξουμε ότι έχουμε μονοτονία, δηλαδή αύξουσα συνάρτηση.

Αλλά και πρακτικά μπορούμε να υπολογίσουμε την $cs4$ στα 180001 σημεία

$$0, 10^{-5}, 2 \cdot 10^{-5}, 3 \cdot 10^{-5}, 4 \cdot 10^{-5}, 5 \cdot 10^{-5}, 6 \cdot 10^{-5}, \dots, 18 - 2 \cdot 10^{-5}, 18 - 10^{-5}, 18$$

και να διαπιστώσουμε μόνιμη αύξηση της τιμής της. Πράγματι η μικρότερη αύξηση ανάμεσα σε δυο διαδοχικά σημεία είναι θετική.

```

>> r=cs4(0:1e-5:18);
>> min(r(2:end)-r(1:end-1))
ans =
    9.9920e-015

```

Επίσης $s_{35}(18) = 1$, (τουλάχιστον με ακρίβεια 16 δεκαδικών). Οπότε η μέγιστη τιμή που επιτυγχάνεται από τα πολυώνυμα $s_1, s_2, s_3, \dots, s_{34}, s_{35}$, είναι το 1.

5

Υλοποίηση της $\tanh x$ σε νευρωνικά δίκτυα

Σκοπός μας τώρα είναι η υλοποίηση των συναρτήσεων `cs4` & `cs36` ως αντικαταστάτριες της $\tanh x$. Θα χρησιμοποιήσουμε το Matlab και το συνοδευτικό του λογισμικό Deep Learning Toolbox.

5.1 Το λογισμικό Deep Learning Toolbox™

Στην ουσία πρόκειται για αναβάθμιση του λίγο παλαιότερου Neural Networks Toolbox. Έτσι μπορούμε να επεξεργαστούμε και δημιουργήσουμε δίκτυα βαθιάς μάθησης.

Η εφαρμογή Deep Network Designer σας επιτρέπει να δημιουργείτε, να απεικονίζετε και να επεξεργάζεστε δίκτυα βαθιάς μάθησης. Χρησιμοποιώντας αυτήν την εφαρμογή, μπορείτε να:

- Εισαγάγετε προρυθμισμένα δίκτυα και να τα επεξεργαστείτε τα για εκμάθηση μεταφοράς.
- Εισαγωγή και επεξεργασία δικτύων και δημιουργία νέων δικτύων.
- Σύρετε και αφήστε να προσθέσετε νέα στρώματα και να δημιουργήσετε νέες συνδέσεις.
- Προβολή και επεξεργασία ιδιοτήτων στρώματος (layers).
- Αναλύστε το δίκτυο για να βεβαιωθείτε ότι έχετε ορίσει σωστά την αρχιτεκτονική και ανίχνευση προβλημάτων πριν από την εκπαίδευση.
- Δημιουργήστε τον κώδικα MATLAB®.

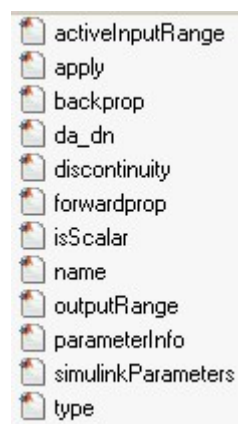
Αφού ολοκληρώσετε το σχεδιασμό ενός δικτύου, μπορείτε να το εξάγετε στο χώρο εργασίας, όπου μπορεί να γίνει αποθήκευση ή εκπαίδευση του δικτύου.

Περισσότερες πληροφορίες μπορεί να αντληθούν στο εγχειρίδιο αναφοράς [RG19] και το εγχειρίδιο χρήστη [UG19].

Πλην όμως εμείς εδώ μάλλον θα αρκεστούμε σε απλούστερες εφαρμογές καθότι δεν μας ενδιαφέρει να λύσουμε περίπλοκα προβλήματα αλλά να ελέγξουμε την αξιοπιστία των `cs4` & `cs36` σε σχέση με την κλασσική συνάρτηση μεταφοράς $\tanh x$. Έτσι θα εφαρμόσουμε τις εν λόγω συναρτήσεις μεταφοράς σε κλασσικά feed-forward νευρωνικά δίκτυα.

Με τη χρήση του συγκεκριμένου πακέτου (toolbox) αποφεύγουμε τον εξειδικευμένο προγραμματισμό και αρκούν απλές προσθήκες με τις νέες συναρτήσεις μεταφοράς `cs4` & `cs36` όπως αυτές δόθηκαν στο προηγούμενο κεφάλαιο.

Έτσι στον φάκελο με όνομα `nntransfer` που είναι εγκατεστημένοι οι φάκελοι για τις υπόλοιπες συναρτήσεις μεταφοράς (transfer functions) προσθέτουμε τον φάκελο `+cs4` ο οποίος περιέχει τα m-αρχεία



Το αρχείο `apply.m` είναι το `cs4.m` όπως το εμφανίσαμε στο προηγούμενο κεφάλαιο. Το `da_dn.m` περιέχει την παράγωγο της συνάρτησης `cs4`. Αμέσως πιο κάτω δίνουμε το σχετικό listing σε Matlab.

```
function d = dy_dn(xin,yin,param);
% TANH      Hyperbolic tangent.
% using splines

xin=abs(xin);
cs=struct('form','pp', ...
    'breaks',[0 0.371025186672900 2.572153900248530 18 1e10], ...
    'coefs',[-1.108522825837648 3.974438687795734e-2 1; ...
1.778506910167428e-1 -7.828353898972084e-1 8.621472609449146e-1; ...
-1.004279706918473e-5 1.09135552311728e-4 7.066442941005233e-4; ...
    0 0 0], ...
    'pieces',4, ...
    'order',3, ...
    'dim',1);
d=ppval(cs,xin);
return
```

Τα υπόλοιπα αρχεία τα αντιγράφουμε ως έχουν από τον φάκελο +tansig που υλοποιείται η συνάρτηση μεταφοράς $\tanh x$ για το nnet. Τέλος στη ρίζα nntransfer αντιγράφουμε ένα αρχείο cs4.m το οποίο είναι αντιγραφή του αντίστοιχου αρχείου tansig.m με τις αντίστοιχες τροποποιήσεις στα ονόματα όπου χρειάζεται.

Το ίδιο ακριβώς κάνουμε και με την συνάρτηση cs36. Γι' αυτή γράφουμε την αντίστοιχη function da_dn.m που υλοποιεί την παράγωγό της. Την αποθηκεύουμε και αυτή στο αρχείο μέσα στο φάκελο +cs36. Το listing της da_dn.m δίνεται παρακάτω.

```
function d = dy_dn(xin, param);
%TANH - derivative
%      using splines
% fprintf('2');
[n1,n2]=size(xin);
xin=reshape(xin,n1*n2,1);
xin=abs(xin);
ln=length(xin);
pp=[
-7.413037935337841e-001 -5.596147740866058e-002 1.000000000000000e+000
 1.389240265787727e-001 -7.972652709424446e-001 7.866933129122237e-001
 3.468218686011184e-001 -6.583412443636719e-001 4.227916840856946e-001
 1.837436129808132e-001 -3.115193757625536e-001 1.803265290541382e-001
 7.937558914316649e-002 -1.277757627817404e-001 7.050274441806470e-002
 3.026633000709207e-002 -4.840017363857390e-002 2.645876031298613e-002
 1.145391313601736e-002 -1.813384363148184e-002 9.825255995472191e-003
 4.211692990406070e-003 -6.679930495464480e-003 3.621812463735620e-003
 1.561333566968340e-003 -2.468237505058414e-003 1.334770463604890e-003
 5.727325300791900e-004 -9.069039380900680e-004 4.909851028177700e-004
 2.113484775304500e-004 -3.341714080108800e-004 1.807162662925400e-004
 7.760447920701899e-005 -1.228229304804320e-004 6.646768166971000e-005
 2.859213365217000e-005 -4.521845127342000e-005 2.445733623125100e-005
 1.050744574416000e-005 -1.662631762123800e-005 8.996144007590000e-006
 3.868494470550001e-006 -6.118871877070000e-006 3.309846633010000e-006
 1.422339588000000e-006 -2.250377406520000e-006 1.217534312110000e-006
 5.234651411700001e-007 -8.280378185200000e-007 4.479305058500000e-007
 1.925144553900000e-007 -3.045726773200000e-007 1.647778818900000e-007
 7.083756996000000e-008 -1.120582219400000e-007 6.062015708000000e-008
 2.605554645000000e-008 -4.122065196000000e-008 2.230043861000000e-008
 9.586409399999999e-009 -1.516510552000000e-008 8.203999230000000e-009
 3.526345680000000e-009 -5.578696126000000e-009 3.018048820000000e-009
 1.297349730000000e-009 -2.052350440000000e-009 1.110287180000000e-009
 4.772469600000000e-010 -7.550007000000000e-010 4.084494000000000e-010
```

```

1.7557506000000000e-010 -2.7775374000000000e-010 1.5026079000000000e-010
6.4588920000000000e-011 -1.0217868000000000e-010 5.5277680000000000e-011
2.3761350000000000e-011 -3.7589760000000000e-011 2.0335570000000000e-011
8.7411899999999999e-012 -1.3828420000000000e-011 7.4810300000000000e-012
3.2157300000000000e-012 -5.0872200000000000e-012 2.7521200000000000e-012
1.1829900000000000e-012 -1.8714800000000000e-012 1.0124500000000000e-012
4.3521000000000000e-013 -6.8848000000000000e-013 3.7246000000000000e-013
1.6005000000000000e-013 -2.5326000000000000e-013 1.3702000000000000e-013
5.9040000000000001e-014 -9.3200000000000000e-014 5.0400000000000000e-014
2.1090000000000000e-014 -3.4160000000000000e-014 1.8560000000000000e-014
1.0140000000000000e-014 -1.3060000000000000e-014 6.7600000000000000e-015
-5.1900000000000000e-015 -2.9200000000000000e-015 2.7600000000000000e-015
                                0                                0                                0];

seira=fix(xin*2)+1;
seira=seira.*(seira<=36)+(seira>36)*37;
pps=pp(seira,:);
xin=xin-seira/2+1/2;
d=(pps(:,1).*xin+pps(:,2)).*xin+pps(:,3)).*ones(ln,1);
d=reshape(d,n1,n2);
return

```

Μετά από αυτά έχουμε δυο νέες συναρτήσεις μεταφοράς για χρήση. Τις `cs4` & `cs36`.

5.2 Set με τα δεδομένα

Επιλέγουμε τέσσερα σετ με δεδομένα όπως δίνονται στο MATLAB. Ειδικότερα:

1° σετ δεδομένων: `simplefit`

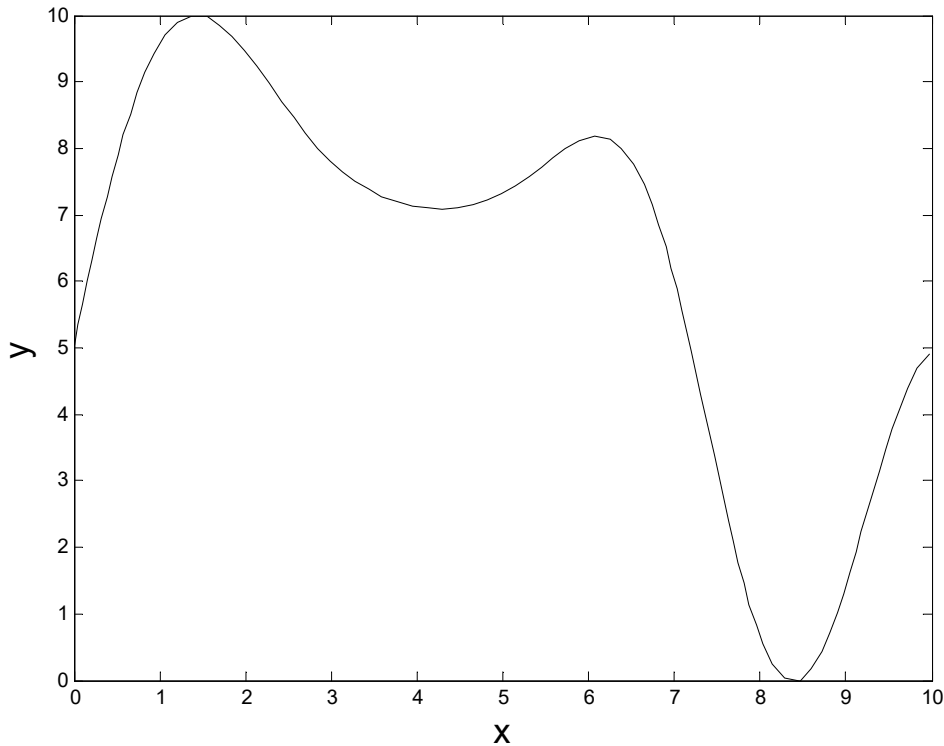
Πρόκειται για μια ημιτονοειδή συνάρτηση μιας μεταβλητής με 94 σημεία να δίνονται.

```

>> load simplefit_dataset
>> whos
  Name                Size                Bytes   Class
Attributes
  simplefitInputs     1x94                752    double
  simplefitTargets     1x94                752    double
>> plot(simplefitInputs,simplefitTargets,'-k');

```

Στην περίπτωση αυτή θα χρησιμοποιήσουμε ένα feed-forward δίκτυο 1-n-1. Δηλαδή με ένα κρυφό layer που αποτελείται από n το πλήθος νευρώνες. Στην είσοδο και στην έξοδο έχουμε 1 νευρώνα. Η συνάρτηση μεταφοράς `tansig` εφαρμόζεται στο κρυφό layer και η συνάρτηση μεταφοράς `purelin` στην έξοδο.



Γράφημα: Γραφική παράσταση της `simplefit`

2^ο σετ δεδομένων: `bodyfat`

Σε αυτό το παράδειγμα επεξηγείται πώς ένα feed-forward νευρωνικό δίκτυο μπορεί να υπολογίσει το ποσοστό σωματικού λίπους βάσει ανατομικών μετρήσεων.

Το πρόβλημα: Εκτίμηση του ποσοστού σωματικού λίπους

Σε αυτό το παράδειγμα προσπαθούμε να οικοδομήσουμε ένα νευρωνικό δίκτυο που μπορεί να εκτιμήσει το ποσοστό σωματικού λίπους ενός ατόμου που περιγράφεται από δεκατρία φυσικά χαρακτηριστικά:

Ηλικία (έτη)

Βάρος σε λίβρες)

Ύψος (ίντσες)

Περιφέρεια λαιμού (cm)

Περιφέρεια θώρακα (cm)

Περιφέρεια κοιλίας (cm)

Περιφέρεια ισχίου (cm)
 Περιφέρεια μηρού (cm)
 Περιφέρεια γόνατος (cm)
 Περιφέρεια αστραγάλου (cm)
 Περιφέρεια δικεφάλου (εκτεταμένη) (cm)
 Περιφέρεια βραχίονα (cm)
 Περιφέρεια καρπού (cm)

Αυτό είναι ένα παράδειγμα ενός προβλήματος προσαρμογής, όπου οι εισόδοι ταιριάζουν με τις αντίστοιχες εξόδους στόχου και θα θέλαμε να δημιουργήσουμε ένα νευρωνικό δίκτυο το οποίο όχι μόνο εκτιμά τους γνωστούς στόχους που δίνουν γνωστές εισόδους αλλά να μπορεί να γενικεύσει για να υπολογίσει με ακρίβεια τα αποτελέσματα για εισόδους που δεν ανήκουν στο σετ που δόθηκε. Περιέχει 252 εισόδους με τα 13 δεδομένα για κάθε περίπτωση. Επίσης 252 εξόδους με το αντίστοιχο ποσοστό λίπους. Φορτώνεται απευθείας στο Matlab [UG19, σελ. 6-14]. Ακολουθεί ένα παράδειγμα.

```
>> clear
>> load bodyfat_dataset
>> whos
  Name                Size          Bytes  Class
  bodyfatInputs       13x252        26208  double
  bodyfatTargets      1x252         2016   double
>> bodyfatInputs(:,1)
ans =
 23.0000
154.2500
 67.7500
 36.2000
 93.1000
 85.2000
 94.5000
 59.0000
 37.3000
 21.9000
 32.0000
 27.4000
 17.1000
>> bodyfatTargets(1)
ans =
 12.3000
```

Έτσι στην 1^η είσοδο παρατηρούμε Ηλικία 23έτη, Βάρος 154.25λίβρες, Ύψος 67.75ίντσες, περιφέρεια λαιμού 36.2cm, περιφέρεια θώρακα 93.1cm, περιφέρεια κοιλίας 85.2cm, περιφέρεια ισχίου 94.5cm, περιφέρεια μηρού 59.0cm, περιφέρεια γόνατου 37.3cm, περιφέρεια αστραγάλου 21.9cm, περιφέρεια δικεφάλου (εκτεταμένη) 32.0cm, περιφέρεια βραχίονα 27.4cm και περιφέρεια καρπού 17.1cm.

Στη συγκεκριμένη περίπτωση μετρήθηκε λίπος 12.3%.

Στην περίπτωση αυτή θα χρησιμοποιήσουμε ένα feedforward δίκτυο 13-n-1. Δηλαδή με ένα κρυφό layer που αποτελείται από n το πλήθος νευρώνες. Στην είσοδο έχουμε 13 και στην

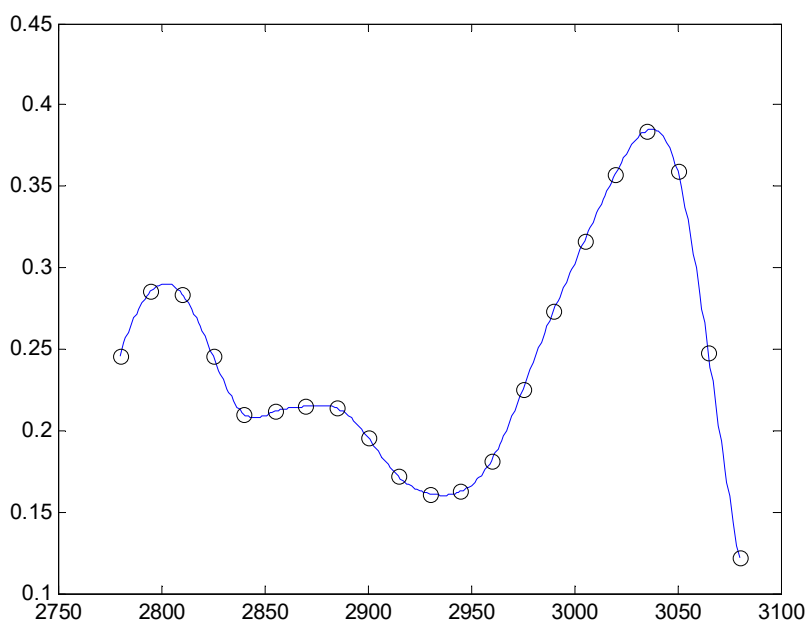
έξοδο έχουμε 1 νευρώνα. Η συνάρτηση μεταφοράς `tansig` εφαρμόζεται στο κρυφό layer και η συνάρτηση μεταφοράς `purelin` στην έξοδο.

3^ο σετ δεδομένων: Cholesterol

Το τρίτο πρόβλημα αναφοράς είναι ένα πρόβλημα προσέγγισης (ή μη γραμμικής παλινδρόμησης) της ρεαλιστικής λειτουργίας. Ο στόχος του δικτύου είναι η πρόβλεψη των επιπέδων χοληστερόλης (ldl, hdl και vldl) βάσει μετρήσεων 21 φασματικών συστατικών. Τα στοιχεία ελήφθησαν από τον Δρ. Neil Purdie, Τμήμα Χημείας, Πανεπιστήμιο της Οκλαχόμα [PuLu92]. Περιέχεται στο σχετικό Matlab toolbox [UG19, σελ. 12-26].

```
>> load cho_dataset
>> whos
  Name           Size           Bytes  Class      Attributes
  choInputs      21x264         44352  double
  choTargets     3x264          6336   double
>> choInputs(:,100) '
ans =
  Columns 1 through 7
  0.2461  0.2854  0.2842  0.2462  0.2102  0.2116  0.2148
  Columns 8 through 14
  0.2138  0.1959  0.1717  0.1613  0.1625  0.1817  0.2249
  Columns 15 through 21
  0.2731  0.3165  0.3577  0.3842  0.3592  0.2481  0.1224
>> choTargets(:,100) '
ans =
  67  87  27
```

Έτσι στο παράδειγμα από πάνω οι 21 μετρήσεις έδωσαν τα 3 αποτελέσματα.



Γράφημα: Δειγματοληψία των παραπάνω 21 τιμών από το φάσμα

Στην περίπτωση αυτή θα χρησιμοποιήσουμε ένα feed-forward δίκτυο 21-n-3. Δηλαδή με ένα κρυφό layer που αποτελείται από n το πλήθος νευρώνες. Στην είσοδο έχουμε 21 και στην έξοδο έχουμε 3 νευρώνες. Ομοίως και εδώ η συνάρτηση μεταφοράς `tansig` εφαρμόζεται στο κρυφό layer και η συνάρτηση μεταφοράς `purelin` στην έξοδο.

5.3 Αλγόριθμοι Εκπαίδευσης

Θα χρησιμοποιήσουμε εννέα διαφορετικούς αλγόριθμους εκμάθησης (Learning Algorithms) που προσφέρονται από το Deep Learning Toolbox του Matlab. Ειδικότερα αυτοί συνοψίζονται στον παρακάτω πίνακα.

Συντομογραφία	Learning Algorithm
LeMa	Levenberg-Marquardt
BrFGS	BFGS Quasi-Newton
ReBP	Resilient Backpropagation
ScCG	Scaled Conjugate Gradient
CG-PB	Conjugate Gradient with Powell/Beale Restarts
FP-CG	Fletcher-Powell Conjugate Gradient
PR-CG	Polak-Ribière Conjugate Gradient
OneSS	One Step Secant
VLRBP	Variable Learning Rate Backpropagation

Οι αλγόριθμοι αυτοί ενεργοποιούνται όταν φτιάχνουμε αρχικά ένα feed forward δίκτυο. Έτσι γράφουμε:

```
>> myNet = feedforwardnet(n, tt)
```

με n το πλήθος των layers και το `tt` μπορεί να λάβει μια από τις τιμές:

```
'trainlm', 'trainbfg', 'trainrp', 'trainscg', 'traincgb',  
'traincgf', 'traincgp', 'trainoss', 'traingdx'
```

που υλοποιούν τους αλγόριθμους μάθησης LeMa, BrFGS, ReBP, ScCG, CG-PB, FP-CG, PR-CG, OnesSS και VLRBR αντίστοιχα.

5.4 Αποτελέσματα

`simplefit` : Για να βρούμε τα αποτελέσματα σε αυτό το σετ τρέχουμε το παρακάτω:

```
disp('simplefit');
format compact;
clear;
load simplefit_dataset;
[x,t]=simplefit_dataset;
trainss={'trainlm','trainbfg','trainrp','trainscg','traincgb', ...
        'traincgf','traincgp','trainoss','traingdx'};
funfcn={'tansig','elliotsig','cs4','cs36'};
for j0=5:5:20,
    for j1=trainss,
        errorsum=zeros(4,1);epochsum=zeros(4,1);
        for j3=1:30,
            j22=0;
            for j2=funfcn,
                seedRandom=rng(1,'twister');
                myNet = feedforwardnet(j0,char(j1));
                myNet=configure(myNet,x,t);myNet=init(myNet);
                myNet.b{1}(1)=0.25*j3*(-1)^j3;
                myNet.b{1}(end)=-myNet.b{1}(1);
                myNet.trainParam.showWindow = false;
                myNet.layers{1}.transferFcn = char(j2);
                myNet.trainParam.goal=0.002;
                [net,tr] = train(myNet,x,t);
                j22=j22+1;
                epochsum(j22)=epochsum(j22)+length(tr.epoch);
                errorsum(j22)=errorsum(j22)+tr.best_vperf;
            end;
        end;

        fprintf('%6.0f %10.2e %10.2f %11s %11s ...
                \n',j0,errorsum(1)/30,epochsum(1)/30,j1{1},funfcn{1});
        fprintf('%6.0f %10.2e %10.2f %11s %11s ...
                \n',j0,errorsum(2)/30,epochsum(2)/30,j1{1},funfcn{2});
        fprintf('%6.0f %10.2e %10.2f %11s %11s ...
                \n',j0,errorsum(3)/30,epochsum(3)/30,j1{1},funfcn{3});
        fprintf('%6.0f %10.2e %10.2f %11s %11s ...
                \n',j0,errorsum(4)/30,epochsum(4)/30,j1{1},funfcn{4});
    end;
end;
```

Όλες οι παράμετροι αφήνονται στην προκαθορισμένη από το σύστημα τιμή [RG19, σελ. 535]. Εξάιρεση η `trainParam.goal=0.002`; Για κάθε μια από τις 36 περιπτώσεις (9 αλγόριθμοι X 4 τιμές του n) κάνουμε 30 διαφορετικές εκκινήσεις των αρχικών βαρών. Έπειτα εφαρμόζουμε τις συναρτήσεις μεταφοράς. Στους επόμενους πίνακες εμφανίζονται

Στην 1η στήλη: Το πλήθος των νευρώνων στο κρυφό layer

Στην 2η στήλη: Το μέσο σφάλμα της εκπαίδευσης στις 30 δοκιμές

Στην 3η στήλη: Ο μέσος αριθμός των epochs που χρειάστηκαν για την εκπαίδευση

Στην 4η στήλη: Ο αλγόριθμος εκπαίδευσης

Στην 5η στήλη: Η συνάρτηση μεταφοράς

Σκοπός είναι το μέσο πλήθος των epochs και του σφάλματος να πλησιάζει κατά το δυνατό την συνάρτηση αναφοράς `tansig` (δηλ. `tanh`) που είναι αξιόπιστη και δοκιμασμένη. Αποκλίσεις (και προς τα πάνω αλλά και προς τα κάτω) δεν είναι επιθυμητές.

n	σφάλμα	epochs	algorithm	tr. Function
5	4.41e-01	19.47	trainlm	tansig
5	1.88e-01	77.33	trainlm	elliotsig
5	6.03e-01	27.30	trainlm	cs4
5	5.00e-01	19.17	trainlm	cs36
5	6.54e-01	36.90	trainbfg	tansig
5	6.36e-01	30.90	trainbfg	elliotsig
5	6.95e-01	35.57	trainbfg	cs4
5	6.64e-01	35.40	trainbfg	cs36
5	2.26e+00	25.30	trainrp	tansig
5	2.13e+00	36.47	trainrp	elliotsig
5	2.03e+00	32.20	trainrp	cs4
5	2.26e+00	25.07	trainrp	cs36
5	1.17e+00	26.40	trainscg	tansig
5	1.30e+00	30.97	trainscg	elliotsig
5	1.11e+00	32.83	trainscg	cs4
5	1.17e+00	28.70	trainscg	cs36
5	8.61e-01	36.27	traincgb	tansig
5	1.18e+00	29.60	traincgb	elliotsig
5	9.05e-01	36.70	traincgb	cs4
5	8.81e-01	34.13	traincgb	cs36
5	1.02e+00	31.97	traincgf	tansig
5	1.29e+00	27.67	traincgf	elliotsig
5	1.10e+00	33.40	traincgf	cs4
5	1.04e+00	30.90	traincgf	cs36
5	9.72e-01	37.00	traincgp	tansig
5	1.25e+00	31.17	traincgp	elliotsig
5	1.09e+00	32.80	traincgp	cs4
5	9.58e-01	35.87	traincgp	cs36
5	1.23e+00	36.80	trainoss	tansig
5	1.52e+00	26.80	trainoss	elliotsig
5	1.24e+00	34.03	trainoss	cs4
5	1.27e+00	37.20	trainoss	cs36
5	3.28e+00	23.30	traingdx	tansig
5	2.92e+00	32.80	traingdx	elliotsig
5	3.27e+00	25.53	traingdx	cs4
5	3.28e+00	23.30	traingdx	cs36

simplefit, 1-5-1

n	σφάλμα	epochs	algorithm	tr. Function
10	3.69e-03	8.93	trainlm	tansig
10	1.49e-02	31.47	trainlm	elliotsig
10	3.21e-03	9.27	trainlm	cs4
10	3.41e-03	8.57	trainlm	cs36
10	4.19e-01	28.43	trainbfg	tansig
10	5.05e-02	34.13	trainbfg	elliotsig
10	3.01e-01	29.53	trainbfg	cs4
10	4.18e-01	28.60	trainbfg	cs36
10	1.40e-01	73.67	trainrp	tansig
10	3.42e-01	73.60	trainrp	elliotsig
10	1.14e-01	88.67	trainrp	cs4
10	1.35e-01	79.90	trainrp	cs36
10	3.86e-01	25.53	trainscg	tansig
10	1.93e-01	38.13	trainscg	elliotsig
10	3.67e-01	28.37	trainscg	cs4
10	4.03e-01	24.63	trainscg	cs36
10	3.81e-01	25.83	traincgb	tansig
10	7.18e-02	38.23	traincgb	elliotsig
10	3.26e-01	27.50	traincgb	cs4
10	3.91e-01	25.33	traincgb	cs36
10	4.39e-01	24.43	traincgf	tansig
10	2.55e-01	33.27	traincgf	elliotsig
10	5.23e-01	22.17	traincgf	cs4
10	4.38e-01	24.67	traincgf	cs36
10	4.09e-01	26.77	traincgp	tansig
10	1.85e-01	36.77	traincgp	elliotsig
10	4.06e-01	27.30	traincgp	cs4
10	4.08e-01	27.60	traincgp	cs36
10	5.12e-01	28.10	trainoss	tansig
10	3.19e-01	61.80	trainoss	elliotsig
10	5.19e-01	26.77	trainoss	cs4
10	5.49e-01	24.80	trainoss	cs36
10	3.90e+00	55.87	traingdx	tansig
10	1.18e+00	95.50	traingdx	elliotsig
10	4.05e+00	52.67	traingdx	cs4
10	3.90e+00	55.87	traingdx	cs36

simplefit, 1-10-1

n	σφάλμα	epochs	algorithm	tr. Function
15	6.98e-03	7.73	trainlm	tansig
15	8.06e-03	22.83	trainlm	elliotsig
15	5.13e-03	7.90	trainlm	cs4
15	6.67e-03	7.83	trainlm	cs36
15	5.21e-02	27.67	trainbfg	tansig
15	4.65e-02	19.70	trainbfg	elliotsig
15	4.86e-02	28.40	trainbfg	cs4
15	5.04e-02	28.00	trainbfg	cs36
15	6.61e-01	71.13	trainrp	tansig
15	1.25e+00	71.97	trainrp	elliotsig
15	1.06e+00	57.53	trainrp	cs4
15	6.68e-01	69.03	trainrp	cs36
15	8.32e-02	32.87	trainscg	tansig
15	7.03e-02	38.93	trainscg	elliotsig
15	8.90e-02	32.43	trainscg	cs4
15	8.31e-02	32.60	trainscg	cs36
15	8.31e-02	32.20	traincgb	tansig
15	6.66e-02	35.47	traincgb	elliotsig
15	5.86e-02	33.17	traincgb	cs4
15	8.02e-02	32.30	traincgb	cs36
15	7.21e-02	34.93	traincgf	tansig
15	5.44e-02	40.63	traincgf	elliotsig
15	6.21e-02	35.87	traincgf	cs4
15	8.63e-02	34.20	traincgf	cs36
15	1.05e-01	33.57	traincgp	tansig
15	7.81e-02	38.40	traincgp	elliotsig
15	7.87e-02	34.33	traincgp	cs4
15	1.05e-01	32.90	traincgp	cs36
15	1.20e-01	44.17	trainoss	tansig
15	1.37e-01	46.97	trainoss	elliotsig
15	1.23e-01	45.17	trainoss	cs4
15	1.23e-01	44.27	trainoss	cs36
15	5.09e+00	67.17	traingdx	tansig
15	1.08e+01	16.70	traingdx	elliotsig
15	5.06e+00	66.33	traingdx	cs4
15	5.09e+00	67.17	traingdx	cs36

simplefit, 1-15-1

n	σφάλμα	epochs	algorithm	tr. Function
20	1.20e-02	11.07	trainlm	tansig
20	1.62e-02	16.50	trainlm	elliotsig
20	9.92e-03	11.23	trainlm	cs4
20	1.19e-02	11.23	trainlm	cs36
20	4.93e-02	30.33	trainbfg	tansig
20	3.39e-02	34.87	trainbfg	elliotsig
20	4.73e-02	30.20	trainbfg	cs4
20	4.91e-02	30.03	trainbfg	cs36
20	6.70e-01	67.77	trainrp	tansig
20	4.79e-01	67.97	trainrp	elliotsig
20	7.26e-01	77.87	trainrp	cs4
20	6.78e-01	64.40	trainrp	cs36
20	2.30e+00	40.20	trainscg	tansig
20	5.53e-01	62.80	trainscg	elliotsig
20	2.36e+00	41.27	trainscg	cs4
20	2.27e+00	44.23	trainscg	cs36
20	4.25e-01	85.33	traincgb	tansig
20	8.02e-01	54.07	traincgb	elliotsig
20	2.70e-01	84.20	traincgb	cs4
20	2.22e-01	85.27	traincgb	cs36
20	4.01e-01	101.63	traincgf	tansig
20	7.68e-01	63.93	traincgf	elliotsig
20	5.52e-01	94.03	traincgf	cs4
20	3.95e-01	100.83	traincgf	cs36
20	1.14e+00	76.70	traincgp	tansig
20	1.13e+00	33.47	traincgp	elliotsig
20	1.18e+00	61.60	traincgp	cs4
20	8.71e-01	82.43	traincgp	cs36
20	2.92e+00	20.03	trainoss	tansig
20	1.56e+00	19.70	trainoss	elliotsig
20	2.91e+00	19.87	trainoss	cs4
20	2.96e+00	18.80	trainoss	cs36
20	5.45e+00	59.53	traingdx	tansig
20	2.28e+00	69.23	traingdx	elliotsig
20	5.31e+00	61.93	traingdx	cs4
20	5.45e+00	59.53	traingdx	cs36

simplefit, 1-20-1

bodyfat : Για να βρούμε τα αποτελέσματα σε αυτό το σετ τρέχουμε το παρακάτω:

```
disp('bodyfat');
format compact;
clear;
load bodyfat_dataset;
[x,t]=bodyfat_dataset;
trainss={'trainlm','trainbfg','trainrp','trainscg','traincgb', ...
        'traincgf','traincgp','trainoss','traingdx'};
funfcn={'tansig','elliotsig','cs4','cs36'};
for j0=5:5:20,
    for j1=trainss,
        errorsum=zeros(4,1);epochsum=zeros(4,1);
        for j3=1:30,
            j22=0;
            for j2=funfcn,
                seedRandom=rng(1,'twister');
                myNet = feedforwardnet(j0,char(j1));
                myNet=configure(myNet,x,t);myNet=init(myNet);
                myNet.b{1}(1)=0.25*j3*(-1)^j3;
                myNet.b{1}(end)=-myNet.b{1}(1);
                myNet.trainParam.showWindow = false;
                myNet.layers{1}.transferFcn = char(j2);
                myNet.trainParam.goal=0.01;
                [net,tr] = train(myNet,x,t);
                j22=j22+1;
                epochsum(j22)=epochsum(j22)+length(tr.epoch);
                errorsum(j22)=errorsum(j22)+tr.best_vperf;
            end;
        end;
    end;

    fprintf('%6.0f %10.2e %10.2f %11s %11s ...
            \n',j0,errorsum(1)/30,epochsum(1)/30,j1{1},funfcn{1});
    fprintf('%6.0f %10.2e %10.2f %11s %11s ...
            \n',j0,errorsum(2)/30,epochsum(2)/30,j1{1},funfcn{2});
    fprintf('%6.0f %10.2e %10.2f %11s %11s ...
            \n',j0,errorsum(3)/30,epochsum(3)/30,j1{1},funfcn{3});
    fprintf('%6.0f %10.2e %10.2f %11s %11s ...
            \n',j0,errorsum(4)/30,epochsum(4)/30,j1{1},funfcn{4});
end;
end;
```

Όλες οι παράμετροι αφήνονται στην προκαθορισμένη από το σύστημα τιμή [RG19, σελ. 535]. Εξάιρεση η `trainParam.goal=0.01`; Για κάθε μια από τις 36 περιπτώσεις (9 αλγόριθμοι X 4 τιμές του n) κάνουμε 30 διαφορετικές εκκινήσεις των αρχικών βαρών. Έπειτα εφαρμόζουμε τις συναρτήσεις μεταφοράς. Στους επόμενους πίνακες εμφανίζονται

Στην 1η στήλη: Το πλήθος των νευρώνων στο κρυφό layer

Στην 2η στήλη: Το μέσο σφάλμα της εκπαίδευσης στις 30 δοκιμές

Στην 3η στήλη: Ο μέσος αριθμός των epochs που χρειάστηκαν για την εκπαίδευση

Στην 4η στήλη: Ο αλγόριθμος εκπαίδευσης

Στην 5η στήλη: Η συνάρτηση μεταφοράς

Σκοπός είναι το μέσο πλήθος των epochs και του σφάλματος να πλησιάζει κατά το δυνατό την συνάρτηση αναφοράς `tansig` (δηλ. `tanh`) που είναι αξιόπιστη και δοκιμασμένη. Αποκλίσεις (και προς τα πάνω αλλά και προς τα κάτω) δεν είναι επιθυμητές.

n	σφάλμα	epochs	algorithm	tr. Function
10	2.43e+01	13.27	trainlm	tansig
10	2.34e+01	13.50	trainlm	elliotsig
10	2.22e+01	13.50	trainlm	tanhsp14
10	2.36e+01	14.23	trainlm	tanhsp11
10	1.77e+01	347.57	trainbfg	tansig
10	1.95e+01	21.17	trainbfg	elliotsig
10	1.74e+01	83.07	trainbfg	tanhsp14
10	1.82e+01	300.47	trainbfg	tanhsp11
10	2.02e+01	31.50	trainrp	tansig
10	2.29e+01	33.67	trainrp	elliotsig
10	2.14e+01	28.27	trainrp	tanhsp14
10	2.02e+01	32.07	trainrp	tanhsp11
10	1.89e+01	26.60	trainscg	tansig
10	2.07e+01	24.60	trainscg	elliotsig
10	1.77e+01	31.67	trainscg	tanhsp14
10	1.87e+01	28.50	trainscg	tanhsp11
10	1.75e+01	28.33	traincgb	tansig
10	1.80e+01	29.83	traincgb	elliotsig
10	1.75e+01	26.77	traincgb	tanhsp14
10	1.73e+01	29.23	traincgb	tanhsp11
10	1.87e+01	27.17	traincgf	tansig
10	1.89e+01	31.63	traincgf	elliotsig
10	1.83e+01	29.60	traincgf	tanhsp14
10	1.88e+01	27.77	traincgf	tanhsp11
10	1.91e+01	29.40	traincgp	tansig
10	2.66e+01	19.63	traincgp	elliotsig
10	1.95e+01	29.17	traincgp	tanhsp14
10	1.92e+01	28.73	traincgp	tanhsp11
10	1.96e+01	30.10	trainoss	tansig
10	2.01e+01	29.23	trainoss	elliotsig
10	1.96e+01	28.33	trainoss	tanhsp14
10	1.96e+01	30.60	trainoss	tanhsp11
10	2.30e+01	110.67	traingdx	tansig
10	2.64e+01	95.77	traingdx	elliotsig
10	2.31e+01	109.17	traingdx	tanhsp14
10	2.30e+01	110.60	traingdx	tanhsp11

bodyfat, 1-10-1

n	σφάλμα	epochs	algorithm	tr. Function
15	2.59e+01	11.00	trainlm	tansig
15	2.12e+01	11.30	trainlm	elliotsig
15	2.15e+01	10.67	trainlm	tanhsp14
15	3.86e+01	11.33	trainlm	tanhsp11
15	1.64e+01	211.73	trainbfg	tansig
15	1.76e+01	560.13	trainbfg	elliotsig
15	1.61e+01	93.27	trainbfg	tanhsp14
15	1.64e+01	198.27	trainbfg	tanhsp11
15	2.82e+01	37.30	trainrp	tansig
15	2.21e+01	40.07	trainrp	elliotsig
15	1.86e+01	44.67	trainrp	tanhsp14
15	2.62e+01	40.67	trainrp	tanhsp11
15	1.82e+01	24.70	trainscg	tansig
15	2.00e+01	27.77	trainscg	elliotsig
15	1.77e+01	25.43	trainscg	tanhsp14
15	1.78e+01	26.00	trainscg	tanhsp11
15	1.69e+01	29.10	traincgb	tansig
15	1.90e+01	22.53	traincgb	elliotsig
15	1.67e+01	30.00	traincgb	tanhsp14
15	1.69e+01	28.60	traincgb	tanhsp11
15	1.64e+01	53.77	traincgf	tansig
15	2.16e+01	22.80	traincgf	elliotsig
15	1.64e+01	55.40	traincgf	tanhsp14
15	1.64e+01	53.23	traincgf	tanhsp11
15	1.87e+01	32.70	traincgp	tansig
15	2.37e+01	14.27	traincgp	elliotsig
15	2.22e+01	25.63	traincgp	tanhsp14
15	1.86e+01	31.20	traincgp	tanhsp11
15	1.81e+01	28.80	trainoss	tansig
15	2.00e+01	20.73	trainoss	elliotsig
15	1.86e+01	25.20	trainoss	tanhsp14
15	1.76e+01	32.57	trainoss	tanhsp11
15	1.90e+01	106.00	traingdx	tansig
15	3.41e+01	66.93	traingdx	elliotsig
15	1.89e+01	105.07	traingdx	tanhsp14
15	1.89e+01	105.97	traingdx	tanhsp11

bodyfat, 1-15-1

n	σφάλμα	epochs	algorithm	tr. Function
20	3.04e+01	12.30	trainlm	tansig
20	2.41e+01	12.00	trainlm	elliotsig
20	3.46e+01	11.77	trainlm	tanhsp14
20	2.87e+01	13.00	trainlm	tanhsp11
20	1.75e+01	1001.00	trainbfg	tansig
20	1.73e+01	391.20	trainbfg	elliotsig
20	1.70e+01	837.13	trainbfg	tanhsp14
20	1.75e+01	1001.00	trainbfg	tanhsp11
20	1.96e+01	84.00	trainrp	tansig
20	1.86e+01	67.40	trainrp	elliotsig
20	1.89e+01	82.50	trainrp	tanhsp14
20	1.98e+01	76.70	trainrp	tanhsp11
20	1.84e+01	32.37	trainscg	tansig
20	1.66e+01	35.10	trainscg	elliotsig
20	1.65e+01	38.47	trainscg	tanhsp14
20	1.80e+01	32.93	trainscg	tanhsp11
20	1.61e+01	28.97	traincgb	tansig
20	1.57e+01	32.27	traincgb	elliotsig
20	1.64e+01	31.83	traincgb	tanhsp14
20	1.64e+01	28.47	traincgb	tanhsp11
20	1.62e+01	38.53	traincgf	tansig
20	1.56e+01	36.87	traincgf	elliotsig
20	3.96e+01	28.10	traincgf	tanhsp14
20	1.62e+01	38.53	traincgf	tanhsp11
20	1.68e+01	29.97	traincgp	tansig
20	1.57e+01	36.60	traincgp	elliotsig
20	1.67e+01	29.27	traincgp	tanhsp14
20	1.67e+01	32.07	traincgp	tanhsp11
20	1.69e+01	40.10	trainoss	tansig
20	1.81e+01	32.70	trainoss	elliotsig
20	1.85e+01	33.50	trainoss	tanhsp14
20	1.72e+01	37.60	trainoss	tanhsp11
20	4.65e+01	79.77	traingdx	tansig
20	4.12e+01	73.07	traingdx	elliotsig
20	4.87e+01	72.93	traingdx	tanhsp14
20	4.65e+01	80.23	traingdx	tanhsp11

bodyfat, 1-20-1

n	σφάλμα	epochs	algorithm	tr. Function
25	2.03e+01	10.00	trainlm	tansig
25	2.16e+01	10.87	trainlm	elliotsig
25	1.96e+01	10.00	trainlm	tanhsp14
25	2.10e+01	10.13	trainlm	tanhsp11
25	2.26e+01	28.60	trainbfg	tansig
25	2.01e+01	340.63	trainbfg	elliotsig
25	2.03e+01	42.03	trainbfg	tanhsp14
25	2.22e+01	28.90	trainbfg	tanhsp11
25	2.25e+01	31.30	trainrp	tansig
25	2.16e+01	27.33	trainrp	elliotsig
25	2.30e+01	31.90	trainrp	tanhsp14
25	2.20e+01	32.03	trainrp	tanhsp11
25	2.24e+01	22.57	trainscg	tansig
25	2.06e+01	16.03	trainscg	elliotsig
25	2.22e+01	20.80	trainscg	tanhsp14
25	2.27e+01	19.90	trainscg	tanhsp11
25	2.07e+01	19.10	traincgb	tansig
25	1.92e+01	20.27	traincgb	elliotsig
25	2.02e+01	19.17	traincgb	tanhsp14
25	2.08e+01	17.53	traincgb	tanhsp11
25	2.09e+01	21.30	traincgf	tansig
25	1.91e+01	23.53	traincgf	elliotsig
25	2.10e+01	22.07	traincgf	tanhsp14
25	2.08e+01	20.60	traincgf	tanhsp11
25	2.12e+01	21.77	traincgp	tansig
25	1.96e+01	24.97	traincgp	elliotsig
25	2.13e+01	23.43	traincgp	tanhsp14
25	2.15e+01	21.67	traincgp	tanhsp11
25	2.17e+01	22.33	trainoss	tansig
25	2.05e+01	19.47	trainoss	elliotsig
25	2.13e+01	32.00	trainoss	tanhsp14
25	2.15e+01	24.20	trainoss	tanhsp11
25	8.87e+01	30.73	traingdx	tansig
25	5.28e+01	41.77	traingdx	elliotsig
25	8.86e+01	30.97	traingdx	tanhsp14
25	8.87e+01	30.70	traingdx	tanhsp11

bodyfat, 1-25-1

cholesterol : Για να βρούμε τα αποτελέσματα σε αυτό το σετ τρέχουμε το παρακάτω:

```

disp('cholesterol');
format compact;
clear;
load cho_dataset;
[x,t]=cho_dataset;
trainss={'trainlm','trainbfg','trainrp','trainscg','traincgb', ...
        'traincgf','traincgp','trainoss','trainingdx'};
funfcn={'tansig','elliotsig','cs4','cs36'};
for j0=5:5:20,
    for j1=trainss,
        errorsum=zeros(4,1);epochsum=zeros(4,1);
        for j3=1:30,
            j22=0;
            for j2=funfcn,
                seedRandom=rng(1,'twister');
                myNet = feedforwardnet(j0,char(j1));
                myNet=configure(myNet,x,t);myNet=init(myNet);
                myNet.b{1}(1)=0.25*j3*(-1)^j3;
                myNet.b{1}(end)=-myNet.b{1}(1);
                myNet.trainParam.showWindow = false;
                myNet.layers{1}.transferFcn = char(j2);
                myNet.trainParam.goal=0.027;
                [net,tr] = train(myNet,x,t);
                j22=j22+1;
                epochsum(j22)=epochsum(j22)+length(tr.epoch);
                errorsum(j22)=errorsum(j22)+tr.best_vperf;
            end;
        end;
    end;

    fprintf('%6.0f %10.2e %10.2f %11s %11s ...
            \n',j0,errorsum(1)/30,epochsum(1)/30,j1{1},funfcn{1});
    fprintf('%6.0f %10.2e %10.2f %11s %11s ...
            \n',j0,errorsum(2)/30,epochsum(2)/30,j1{1},funfcn{2});
    fprintf('%6.0f %10.2e %10.2f %11s %11s ...
            \n',j0,errorsum(3)/30,epochsum(3)/30,j1{1},funfcn{3});
    fprintf('%6.0f %10.2e %10.2f %11s %11s ...
            \n',j0,errorsum(4)/30,epochsum(4)/30,j1{1},funfcn{4});
end;
end;

```

Όλες οι παράμετροι αφήνονται στην προκαθορισμένη από το σύστημα τιμή [RG19, σελ. 535]. Εξαιρέση η `myNet.trainParam.goal=0.027`; Για κάθε μια από τις 36 περιπτώσεις (9 αλγόριθμοι X 4 τιμές του n) κάνουμε 30 διαφορετικές εκκινήσεις των αρχικών βαρών. Έπειτα εφαρμόζουμε τις συναρτήσεις μεταφοράς. Στους επόμενους πίνακες εμφανίζονται

Στην 1η στήλη: Το πλήθος των νευρώνων στο κρυφό layer

Στην 2η στήλη: Το μέσο σφάλμα της εκπαίδευσης στις 30 δοκιμές

Στην 3η στήλη: Ο μέσος αριθμός των epochs που χρειάστηκαν για την εκπαίδευση

Στην 4η στήλη: Ο αλγόριθμος εκπαίδευσης

Στην 5η στήλη: Η συνάρτηση μεταφοράς

Σκοπός είναι το μέσο πλήθος των epochs και του σφάλματος να πλησιάζει κατά το δυνατό την συνάρτηση αναφοράς `tansig` (δηλ. `tanh`) που είναι αξιόπιστη και δοκιμασμένη. Αποκλίσεις (και προς τα πάνω αλλά και προς τα κάτω) δεν είναι επιθυμητές.

n	σφάλμα	epochs	algorithm	tr. Function
5	1.77e+03	21.40	trainlm	tansig
5	5.37e+02	29.20	trainlm	elliotsig
5	5.32e+02	23.93	trainlm	cs4
5	5.39e+02	25.70	trainlm	cs36
5	7.90e+02	843.03	trainbfg	tansig
5	7.72e+02	649.03	trainbfg	elliotsig
5	8.04e+02	843.27	trainbfg	cs4
5	8.14e+02	666.20	trainbfg	cs36
5	1.01e+03	46.37	trainrp	tansig
5	8.27e+02	49.93	trainrp	elliotsig
5	1.17e+03	35.77	trainrp	cs4
5	1.07e+03	45.97	trainrp	cs36
5	7.56e+02	37.40	trainscg	tansig
5	7.66e+02	30.97	trainscg	elliotsig
5	8.02e+02	31.60	trainscg	cs4
5	7.58e+02	36.53	trainscg	cs36
5	7.73e+02	35.57	traincgb	tansig
5	7.81e+02	26.07	traincgb	elliotsig
5	7.88e+02	33.20	traincgb	cs4
5	7.74e+02	34.63	traincgb	cs36
5	8.20e+02	27.40	traincgf	tansig
5	8.39e+02	24.27	traincgf	elliotsig
5	8.00e+02	27.67	traincgf	cs4
5	8.59e+02	24.30	traincgf	cs36
5	8.19e+02	24.97	traincgp	tansig
5	7.77e+02	28.07	traincgp	elliotsig
5	8.10e+02	26.80	traincgp	cs4
5	8.16e+02	25.70	traincgp	cs36
5	8.28e+02	29.00	trainoss	tansig
5	7.89e+02	31.33	trainoss	elliotsig
5	8.49e+02	30.97	trainoss	cs4
5	8.37e+02	31.57	trainoss	cs36
5	8.73e+02	91.10	traingdx	tansig
5	1.13e+03	83.13	traingdx	elliotsig
5	8.73e+02	91.07	traingdx	cs4
5	8.73e+02	91.10	traingdx	cs36

cholesterol, 1-5-1

n	σφάλμα	epochs	algorithm	tr. Function
10	4.00e+02	14.00	trainlm	tansig
10	3.42e+02	14.40	trainlm	elliotsig
10	4.27e+02	13.67	trainlm	cs4
10	3.92e+02	13.93	trainlm	cs36
10	4.69e+02	761.27	trainbfg	tansig
10	5.17e+02	1001.00	trainbfg	elliotsig
10	5.13e+02	216.43	trainbfg	cs4
10	4.88e+02	382.07	trainbfg	cs36
10	5.36e+02	61.17	trainrp	tansig
10	5.61e+02	43.00	trainrp	elliotsig
10	5.11e+02	61.30	trainrp	cs4
10	5.27e+02	72.80	trainrp	cs36
10	4.70e+02	39.47	trainscg	tansig
10	5.33e+02	42.30	trainscg	elliotsig
10	4.60e+02	40.27	trainscg	cs4
10	4.44e+02	44.63	trainscg	cs36
10	4.33e+02	44.53	traincgb	tansig
10	5.31e+02	38.63	traincgb	elliotsig
10	4.17e+02	48.60	traincgb	cs4
10	4.39e+02	42.47	traincgb	cs36
10	4.58e+02	41.83	traincgf	tansig
10	5.64e+02	31.63	traincgf	elliotsig
10	4.22e+02	48.90	traincgf	cs4
10	4.32e+02	43.60	traincgf	cs36
10	4.57e+02	46.40	traincgp	tansig
10	5.54e+02	36.87	traincgp	elliotsig
10	4.84e+02	41.57	traincgp	cs4
10	4.89e+02	41.63	traincgp	cs36
10	5.64e+02	42.80	trainoss	tansig
10	5.77e+02	31.60	trainoss	elliotsig
10	5.78e+02	47.13	trainoss	cs4
10	5.43e+02	46.03	trainoss	cs36
10	6.63e+02	91.27	traingdx	tansig
10	6.65e+02	92.73	traingdx	elliotsig
10	6.55e+02	91.53	traingdx	cs4
10	6.63e+02	91.27	traingdx	cs36

cholesterol, 1-10-1

n	σφάλμα	epochs	algorithm	tr. Function
15	7.11e+02	12.80	trainlm	tansig
15	6.23e+02	20.40	trainlm	elliotsig
15	6.32e+02	11.73	trainlm	cs4
15	6.69e+02	14.33	trainlm	cs36
15	8.05e+02	46.73	trainbfg	tansig
15	7.60e+02	111.13	trainbfg	elliotsig
15	8.00e+02	49.93	trainbfg	cs4
15	8.04e+02	46.07	trainbfg	cs36
15	9.07e+02	38.00	trainrp	tansig
15	8.36e+02	31.37	trainrp	elliotsig
15	9.33e+02	33.33	trainrp	cs4
15	8.74e+02	39.87	trainrp	cs36
15	8.96e+02	19.97	trainscg	tansig
15	7.88e+02	18.80	trainscg	elliotsig
15	8.93e+02	20.30	trainscg	cs4
15	8.97e+02	20.00	trainscg	cs36
15	8.82e+02	17.13	traincgb	tansig
15	8.42e+02	16.47	traincgb	elliotsig
15	8.62e+02	17.87	traincgb	cs4
15	8.63e+02	18.13	traincgb	cs36
15	8.59e+02	18.13	traincgf	tansig
15	8.41e+02	18.23	traincgf	elliotsig
15	8.65e+02	17.50	traincgf	cs4
15	8.69e+02	18.07	traincgf	cs36
15	8.91e+02	17.37	traincgp	tansig
15	8.16e+02	16.93	traincgp	elliotsig
15	8.51e+02	18.47	traincgp	cs4
15	8.90e+02	18.30	traincgp	cs36
15	8.33e+02	22.07	trainoss	tansig
15	8.30e+02	24.20	trainoss	elliotsig
15	8.72e+02	21.70	trainoss	cs4
15	8.34e+02	22.23	trainoss	cs36
15	8.12e+02	98.10	traingdx	tansig
15	7.70e+02	87.43	traingdx	elliotsig
15	8.07e+02	98.30	traingdx	cs4
15	8.13e+02	98.10	traingdx	cs36

cholesterol, 1-15-1

n	σφάλμα	epochs	algorithm	tr. Function
20	7.55e+02	15.00	trainlm	tansig
20	6.22e+02	16.07	trainlm	elliotsig
20	6.58e+02	15.73	trainlm	cs4
20	9.13e+02	14.73	trainlm	cs36
20	7.25e+02	216.50	trainbfg	tansig
20	6.51e+02	309.93	trainbfg	elliotsig
20	7.22e+02	223.60	trainbfg	cs4
20	7.25e+02	210.50	trainbfg	cs36
20	8.09e+02	35.27	trainrp	tansig
20	7.46e+02	30.70	trainrp	elliotsig
20	8.02e+02	38.00	trainrp	cs4
20	8.08e+02	35.83	trainrp	cs36
20	7.82e+02	28.43	trainscg	tansig
20	6.80e+02	24.50	trainscg	elliotsig
20	8.04e+02	22.60	trainscg	cs4
20	8.10e+02	23.00	trainscg	cs36
20	7.28e+02	30.73	traincgb	tansig
20	6.91e+02	24.50	traincgb	elliotsig
20	7.59e+02	24.73	traincgb	cs4
20	7.44e+02	27.90	traincgb	cs36
20	7.73e+02	21.83	traincgf	tansig
20	7.03e+02	21.23	traincgf	elliotsig
20	7.69e+02	22.53	traincgf	cs4
20	7.76e+02	21.13	traincgf	cs36
20	7.68e+02	24.03	traincgp	tansig
20	6.99e+02	23.93	traincgp	elliotsig
20	7.69e+02	24.17	traincgp	cs4
20	7.66e+02	25.37	traincgp	cs36
20	7.64e+02	27.93	trainoss	tansig
20	7.24e+02	31.07	trainoss	elliotsig
20	7.30e+02	30.77	trainoss	cs4
20	7.37e+02	29.83	trainoss	cs36
20	9.57e+02	89.53	traingdx	tansig
20	1.18e+03	70.20	traingdx	elliotsig
20	9.53e+02	89.73	traingdx	cs4
20	9.57e+02	89.53	traingdx	cs36

cholesterol, 1-20-1

Παρακάτω δίνουμε πίνακες που συμπυκνώνουν την πληροφορία από τις προηγούμενες 15 σελίδες. Δηλαδή για κάθε υλοποίηση (δηλαδή πλήθους νευρώνων στον κρυφό layer) παρουσιάζουμε το μέσο αριθμό εποχών (epochs) που απαιτήθηκαν για τη σύγκλιση.

	tanh	cs4	cs36
LeMa	19.47	27.30	19.17
BrFGS	36.90	35.57	35.40
ReBP	25.30	32.20	25.07
ScCG	26.40	32.83	28.70
CG-PB	36.27	36.70	34.13
FP-CG	31.97	33.40	30.90
PR-CG	37.00	32.80	35.87
OneSS	36.80	34.03	37.20
VLRBP	23.30	25.53	23.30

simplefit: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 1-5-1.

	tanh	cs4	cs36
LeMa	8.93	9.27	8.57
BrFGS	28.43	29.53	28.60
ReBP	73.67	88.67	79.90
ScCG	25.53	28.37	24.63
CG-PB	25.83	27.50	25.33
FP-CG	24.43	22.17	24.67
PR-CG	26.77	27.30	27.60
OneSS	28.10	26.77	24.80
VLRBP	55.87	52.67	55.87

simplefit: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 1-10-1.

	tanh	cs4	cs36
LeMa	7.73	7.90	7.83
BrFGS	27.67	28.40	28.00
ReBR	71.13	57.53	69.03
ScCG	32.87	32.43	32.60
CG-PB	32.20	33.17	32.30
FP-CG	34.93	35.87	34.20
PR-CG	33.57	34.33	32.90
OneSS	44.17	45.17	44.27
VLRBR	67.17	66.33	67.17

simplefit: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 1-15-1.

	tanh	cs4	cs36
LeMa	11.07	11.23	11.23
BrFGS	30.33	30.20	30.03
ReBR	67.77	77.87	64.40
ScCG	40.20	41.27	44.23
CG-PB	85.33	84.20	85.27
FP-CG	101.63	94.03	100.83
PR-CG	76.70	61.60	82.43
OneSS	20.03	19.87	18.80
VLRBR	59.53	61.93	59.53

simplefit: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 1-20-1.

	tanh	cs4	cs36
LeMa	13.27	13.50	14.23
BrFGS	347.57	83.07	300.47
ReBR	31.50	28.27	32.07
ScCG	26.60	31.67	28.50
CG-PB	28.33	26.77	29.23
FP-CG	27.17	29.60	27.77
PR-CG	29.40	29.17	28.73
OneSS	30.10	28.33	30.60
VLRBR	110.67	109.17	110.60

bodyfat: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 13-10-1.

	tanh	cs4	cs36
LeMa	11.00	10.67	11.33
BrFGS	211.73	93.27	198.27
ReBR	37.30	44.67	40.67
ScCG	24.20	25.43	26.00
CG-PB	29.10	30.00	28.60
FP-CG	53.77	55.40	53.23
PR-CG	32.70	25.63	31.20
OneSS	28.80	25.20	32.57
VLRBR	106.00	105.07	105.97

bodyfat: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 13-15-1.

	tanh	cs4	cs36
LeMa	12.30	11.77	13.00
BrFGS	1001.00	837.13	1001.00
ReBR	84.00	82.50	76.70
ScCG	32.37	38.47	32.93
CG-PB	28.97	31.83	28.47
FP-CG	38.53	28.10	38.53
PR-CG	29.97	29.27	32.07
OneSS	40.10	33.50	37.60
VLRBR	79.77	72.93	80.23

bodyfat: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 13-20-1.

	tanh	cs4	cs36
LeMa	10.00	10.00	10.13
BrFGS	28.60	42.03	28.90
ReBR	31.30	31.90	32.03
ScCG	22.57	20.80	19.90
CG-PB	19.10	19.17	17.53
FP-CG	21.30	22.07	20.60
PR-CG	21.77	23.43	21.67
OneSS	22.33	32.00	24.20
VLRBR	30.73	30.97	30.70

bodyfat: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 13-25-1.

	tanh	cs4	cs36
LeMa	21.4	23.93	25.70
BrFGS	843.03	843.27	666.2
ReBP	46.37	35.77	45.97
ScCG	37.40	31.60	36.53
CG-PB	35.57	33.2	34.63
FP-CG	27.40	27.67	24.30
PR-CG	24.97	26.80	25.70
OneSS	29.00	30.97	31.57
VLRBP	91.10	91.07	91.10

cholesterol: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 21-5-3.

	tanh	cs4	cs36
LeMa	14.00	13.67	13.93
BrFGS	761.27	716.43	782.07
ReBP	61.17	61.30	72.80
ScCG	39.47	40.27	44.63
CG-PB	44.53	48.60	42.47
FP-CG	41.83	48.90	43.60
PR-CG	46.40	41.57	41.63
OneSS	42.80	47.13	46.03
VLRBP	91.27	91.53	91.27

cholesterol: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 21-10-3.

	tanh	cs4	cs36
LeMa	12.80	11.73	14.33
BrFGS	46.73	49.93	46.07
ReBP	38.00	33.33	39.87
ScCG	19.97	20.30	20.00
CG-PB	17.13	17.87	18.13
FP-CG	18.13	17.50	18.07
PR-CG	17.37	18.47	18.30
OneSS	22.07	21.70	22.23
VLRBP	98.10	98.30	98.10

cholesterol: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 21-15-3.

	tanh	cs4	cs36
LeMa	15.00	15.73	14.73
BrFGS	216.5	223.6	210.5
ReBP	35.27	38.00	35.83
ScCG	28.43	22.60	23.00
CG-PB	30.73	24.73	27.90
FP-CG	21.83	22.53	21.13
PR-CG	24.03	24.17	25.37
OneSS	27.93	30.77	29.83
VLRBP	89.53	89.73	89.53

cholesterol: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 21-20-3.

Τέλος δίνουμε μια υπερσυμπύκνωση της πληροφορίας που παρατέθηκε εκτενώς παραπάνω. Έτσι στους παρακάτω πίνακες και για κάθε υλοποίηση (αριθμός νευρώνων στον κρυφό layer) εμφανίζεται η μέση ποσοστιαία απόκλιση από την $\tanh x$ στον αριθμό των εποχών (epochs) για τις διάφορες συναρτήσεις μεταφοράς στις 36 περιπτώσεις.

simplefit	cs4	cs36	elliott
1 – 5 – 1	14.4	3.2	54.5
1 – 10 – 1	7.5	3.8	70.5
1 – 15 – 1	4.1	1.2	40.7
1 – 20 – 1	5.9	3.5	29.8

simplefit: Μέση ποσοστιαία απόκλιση στον αριθμό των εποχών (epochs) για τις διάφορες συναρτήσεις μεταφοράς στις 36 περιπτώσεις για κάθε υλοποίηση (πλήθος κρυφών νευρώνων)

bodyfat	cs4	cs36	elliott
13 – 10 – 1	14.4	4.3	20.2
13 – 15 – 1	13.6	4.9	43.2
13 – 20 – 1	11.7	3.5	17.4
13 – 25 – 1	12.1	4.1	135.7

bodyfat: Μέση ποσοστιαία απόκλιση στον αριθμό των εποχών (epochs) για τις διάφορες συναρτήσεις μεταφοράς στις 36 περιπτώσεις για κάθε υλοποίηση (πλήθος κρυφών νευρώνων)

cholesterol	cs4	cs36	elliott
21 – 5 – 3	8.00	7.78	16.85
21 – 10 – 3	6.37	8.35	17.46
21 – 15 – 3	5.02	3.41	27.55
21 – 20 – 3	7.79	5.56	14.89

cholesterol: Μέση ποσοστιαία απόκλιση στον αριθμό των εποχών (epochs) για τις διάφορες συναρτήσεις μεταφοράς στις 36 περιπτώσεις για κάθε υλοποίηση (πλήθος κρυφών νευρώνων)

ΣΥΜΠΕΡΑΣΜΑ: Γενικότερα η cs36 ακολουθεί πιο πιστά την $\tanh x$ απ' ό τι η cs4. Η elliotsig εμφανίζει σχετικά μεγάλες αποκλίσεις.

6

Εκπαίδευση στο αρχείο δεδομένων diabetes.csv

Εδώ έχουμε ένα πρόβλημα αναγνώρισης προτύπων [UG19, σελ.12-28]. Ο στόχος του δικτύου είναι να αποφασίσει εάν ένα άτομο έχει διαβήτη, με βάση τα προσωπικά δεδομένα (ηλικία, εγκυμοσύνες) και τα αποτελέσματα ιατρικών εξετάσεων (π.χ. αρτηριακή πίεση, δείκτης σωματικής μάζας, αποτέλεσμα δοκιμής ανοχής γλυκόζης κλπ.). Τα στοιχεία ελήφθησαν από

<https://www.kaggle.com/uciml/pima-indians-diabetes-database#diabetes.csv>

Το δίκτυο που χρησιμοποιείται για αυτό το πρόβλημα είναι ένα δίκτυο 8-*n*-15-2 με νευρώνες $\tanh x$ σε όλα τα επίπεδα. Τα δεδομένα για τις 768 περιπτώσεις αναλύονται ως εξής:

έγκυος	Πόσες φορές είναι έγκυος
γλυκόζη	συγκέντρωση γλυκόζης στο πλάσμα (δοκιμή ανοχής γλυκόζης)
πίεση	Διαστολική αρτηριακή πίεση (mm Hg)
triceps	πάχος πτυχώσεων δέρματος Triceps (mm)
ινσουλίνη	2-ωρη ινσουλίνη στον ορό ($\mu\text{U} / \text{ml}$)
Μάζα	Δείκτης μάζας σώματος (βάρος σε kg / (ύψος σε m) ²)
Γενεαλογία	γενεαλογική λειτουργία του σακχαρώδη διαβήτη
Ηλικία	Ηλικία (έτη)
Διαβήτης	έλεγχος για διαβήτη +1/-1

Τα πρώτα 8 δεδομένα δίνονται στην είσοδο και το ένατο στην έξοδο

Εδώ όλες οι συναρτήσεις (και η έξοδος) ακολουθούν την $\tanh x$. Όλες οι παράμετροι αφήνονται στην προκαθορισμένη από το σύστημα τιμή [RG19, σελ. 535]. Εξαίρεση η `myNet.trainParam.goal=0.05`; Για κάθε μια από τις 36 περιπτώσεις (9 αλγόριθμοι X 4

τιμές του η κάνουμε 10 διαφορετικές εκκινήσεις των αρχικών βαρών. Έπειτα εφαρμόζουμε τις συναρτήσεις μεταφοράς. Στους επόμενους πίνακες εμφανίζονται

Στην 1η στήλη: Το πλήθος των νευρώνων στο κρυφό layer

Στην 2η στήλη: Το μέσο σφάλμα της εκπαίδευσης στις 10 δοκιμές

Στην 3η στήλη: Ο μέσος αριθμός των epochs που χρειάστηκαν για την εκπαίδευση

Στην 4η στήλη: Ο αλγόριθμος εκπαίδευσης

Στην 5η στήλη: Η συνάρτηση μεταφοράς

Σκοπός είναι το μέσο πλήθος των epochs και του σφάλματος να πλησιάζει κατά το δυνατό την συνάρτηση αναφοράς \tansig (δηλ. \tanh) που είναι αξιόπιστη και δοκιμασμένη.

Αποκλίσεις (και προς τα πάνω αλλά και προς τα κάτω) δεν είναι επιθυμητές.

Χρησιμοποιήσαμε το παρακάτω πρόγραμμα.

```
disp('diabetes');
format compact;
load diabetes.txt;
x=diabetes(:,1:8)';
t=diabetes(:,9)';
t=(t<.5)*(-1)+(t>.5)*1;
trainss={'trainlm','trainbfg','trainrp','trainscg','traincgb', ...
        'traincgf','traincgp','trainoss','traingdx'};
funfcn={'tansig','elliotsig','cs4','cs36'};
for j0=10:5:25,
    for j1=trainss,
        errorsum=zeros(4,1);epochsum=zeros(4,1);
        for j3=1:30,
            j22=0;
            for j2=funfcn,
                seedRandom=rng(1,'twister');
                myNet = feedforwardnet([j0 15],char(j1));
                myNet=configure(myNet,x,t);myNet=init(myNet);
                myNet.b{1}(1)=0.25*j3*(-1)^j3;
                myNet.b{1}(end)=-myNet.b{1}(1);
                myNet.trainParam.showWindow = false;
                myNet.layers{1}.transferFcn = char(j2);
                myNet.layers{2}.transferFcn = char(j2);
                myNet.layers{3}.transferFcn = char(j2);
                myNet.trainParam.goal=0.05;
                [net,tr] = train(myNet,x,t);
                j22=j22+1;
                epochsum(j22)=epochsum(j22)+length(tr.epoch);
                errorsum(j22)=errorsum(j22)+tr.best_vperf;
            end;
        end;
    end;

fprintf('%6.0f %10.2e %10.2f %11s %11s ...
        \n',j0,errorsum(1)/30,epochsum(1)/30,j1{1},funfcn{1});
fprintf('%6.0f %10.2e %10.2f %11s %11s ...
        \n',j0,errorsum(2)/30,epochsum(2)/30,j1{1},funfcn{2});
fprintf('%6.0f %10.2e %10.2f %11s %11s ...
        \n',j0,errorsum(3)/30,epochsum(3)/30,j1{1},funfcn{3});
fprintf('%6.0f %10.2e %10.2f %11s %11s ...
        \n',j0,errorsum(4)/30,epochsum(4)/30,j1{1},funfcn{4});
end;
end;
```

Τα αποτελέσματα:

n	σφάλμα	epochs	algorithm	tr. Function
10	2.27e-01	3.67	trainlm	tansig
10	2.33e-01	4.40	trainlm	elliotsig
10	2.29e-01	3.37	trainlm	cs4
10	2.30e-01	3.50	trainlm	cs36
10	2.28e-01	6.20	trainbfg	tansig
10	2.33e-01	8.23	trainbfg	elliotsig
10	2.27e-01	6.53	trainbfg	cs4
10	2.28e-01	6.20	trainbfg	cs36
10	2.33e-01	8.90	trainrp	tansig
10	2.42e-01	6.37	trainrp	elliotsig
10	2.28e-01	9.67	trainrp	cs4
10	2.33e-01	9.33	trainrp	cs36
10	2.29e-01	8.37	trainscg	tansig
10	2.46e-01	6.47	trainscg	elliotsig
10	2.28e-01	8.67	trainscg	cs4
10	2.29e-01	8.83	trainscg	cs36
10	2.27e-01	6.43	traincgb	tansig
10	2.34e-01	6.43	traincgb	elliotsig
10	2.27e-01	6.43	traincgb	cs4
10	2.27e-01	6.70	traincgb	cs36
10	2.27e-01	7.47	traincgf	tansig
10	2.35e-01	7.63	traincgf	elliotsig
10	2.28e-01	7.33	traincgf	cs4
10	2.27e-01	7.47	traincgf	cs36
10	2.27e-01	7.30	traincgp	tansig
10	2.34e-01	6.27	traincgp	elliotsig
10	2.27e-01	7.10	traincgp	cs4
10	2.28e-01	7.37	traincgp	cs36
10	2.29e-01	7.30	trainoss	tansig
10	2.34e-01	7.07	trainoss	elliotsig
10	2.27e-01	7.50	trainoss	cs4
10	2.28e-01	7.33	trainoss	cs36
10	2.52e-01	25.60	traingdx	tansig
10	2.57e-01	27.30	traingdx	elliotsig
10	2.52e-01	25.27	traingdx	cs4
10	2.52e-01	25.60	traingdx	cs36

diabetes, 8-10-15-2

n	σφάλμα	epochs	algorithm	tr. Function
15	1.99e-01	3.77	trainlm	tansig
15	1.98e-01	4.03	trainlm	elliotsig
15	2.02e-01	3.97	trainlm	cs4
15	2.00e-01	3.93	trainlm	cs36
15	1.92e-01	6.30	trainbfg	tansig
15	1.91e-01	6.73	trainbfg	elliotsig
15	1.92e-01	6.63	trainbfg	cs4
15	1.93e-01	6.23	trainbfg	cs36
15	1.94e-01	11.03	trainrp	tansig
15	2.01e-01	10.83	trainrp	elliotsig
15	2.02e-01	9.27	trainrp	cs4
15	1.98e-01	9.97	trainrp	cs36
15	1.94e-01	9.60	trainscg	tansig
15	1.88e-01	9.03	trainscg	elliotsig
15	2.04e-01	10.10	trainscg	cs4
15	1.95e-01	9.60	trainscg	cs36
15	1.98e-01	6.87	traincgb	tansig
15	1.91e-01	5.77	traincgb	elliotsig
15	1.95e-01	6.70	traincgb	cs4
15	1.96e-01	6.73	traincgb	cs36
15	1.92e-01	8.37	traincgf	tansig
15	1.92e-01	6.63	traincgf	elliotsig
15	1.94e-01	8.17	traincgf	cs4
15	1.92e-01	8.53	traincgf	cs36
15	1.93e-01	8.13	traincgp	tansig
15	1.90e-01	5.50	traincgp	elliotsig
15	1.94e-01	7.73	traincgp	cs4
15	1.96e-01	7.27	traincgp	cs36
15	1.98e-01	7.57	trainoss	tansig
15	1.91e-01	7.67	trainoss	elliotsig
15	2.03e-01	7.17	trainoss	cs4
15	2.02e-01	8.00	trainoss	cs36
15	2.56e-01	25.83	traingdx	tansig
15	2.06e-01	40.00	traingdx	elliotsig
15	2.36e-01	30.30	traingdx	cs4
15	2.46e-01	28.30	traingdx	cs36

diabetes, 8-15-15-2

n	σφάλμα	epochs	algorithm	tr. Function
20	4.01e-01	3.30	trainlm	tansig
20	2.63e-01	3.73	trainlm	elliotsig
20	2.70e-01	5.77	trainlm	cs4
20	3.98e-01	3.73	trainlm	cs36
20	4.23e-01	4.97	trainbfg	tansig
20	2.58e-01	7.40	trainbfg	elliotsig
20	2.67e-01	6.33	trainbfg	cs4
20	4.26e-01	4.03	trainbfg	cs36
20	2.67e-01	9.97	trainrp	tansig
20	2.59e-01	7.13	trainrp	elliotsig
20	2.70e-01	9.43	trainrp	cs4
20	2.68e-01	9.40	trainrp	cs36
20	2.87e-01	6.70	trainscg	tansig
20	3.64e-01	2.73	trainscg	elliotsig
20	3.16e-01	5.23	trainscg	cs4
20	2.88e-01	6.53	trainscg	cs36
20	3.32e-01	5.67	traincgb	tansig
20	2.81e-01	6.50	traincgb	elliotsig
20	2.76e-01	7.87	traincgb	cs4
20	3.22e-01	6.80	traincgb	cs36
20	3.31e-01	5.63	traincgf	tansig
20	2.84e-01	7.23	traincgf	elliotsig
20	3.19e-01	7.53	traincgf	cs4
20	2.84e-01	6.60	traincgf	cs36
20	4.07e-01	3.63	traincgp	tansig
20	2.94e-01	4.83	traincgp	elliotsig
20	2.92e-01	6.23	traincgp	cs4
20	3.61e-01	4.83	traincgp	cs36
20	3.33e-01	5.60	trainoss	tansig
20	2.58e-01	7.17	trainoss	elliotsig
20	2.82e-01	7.17	trainoss	cs4
20	3.25e-01	6.63	trainoss	cs36
20	3.23e-01	16.93	traingdx	tansig
20	3.19e-01	12.07	traingdx	elliotsig
20	3.33e-01	14.20	traingdx	cs4
20	3.23e-01	16.83	traingdx	cs36

diabetes, 8-20-15-2

n	σφάλμα	epochs	algorithm	tr. Function
25	2.40e-01	4.03	trainlm	tansig
25	2.31e-01	3.83	trainlm	elliotsig
25	2.39e-01	3.87	trainlm	cs4
25	2.40e-01	4.10	trainlm	cs36
25	3.10e-01	3.37	trainbfg	tansig
25	2.25e-01	5.63	trainbfg	elliotsig
25	2.27e-01	5.07	trainbfg	cs4
25	3.11e-01	3.20	trainbfg	cs36
25	2.52e-01	6.17	trainrp	tansig
25	2.39e-01	8.67	trainrp	elliotsig
25	2.56e-01	8.50	trainrp	cs4
25	2.52e-01	6.17	trainrp	cs36
25	2.54e-01	9.40	trainscg	tansig
25	2.29e-01	6.53	trainscg	elliotsig
25	2.52e-01	42.67	trainscg	cs4
25	2.54e-01	8.50	trainscg	cs36
25	2.33e-01	5.70	traincgb	tansig
25	2.28e-01	5.00	traincgb	elliotsig
25	2.32e-01	6.27	traincgb	cs4
25	2.33e-01	5.90	traincgb	cs36
25	2.33e-01	5.23	traincgf	tansig
25	2.23e-01	5.27	traincgf	elliotsig
25	2.30e-01	5.33	traincgf	cs4
25	2.32e-01	5.33	traincgf	cs36
25	2.34e-01	5.07	traincgp	tansig
25	2.26e-01	4.70	traincgp	elliotsig
25	2.35e-01	4.37	traincgp	cs4
25	2.33e-01	5.53	traincgp	cs36
25	3.10e-01	3.70	trainoss	tansig
25	2.23e-01	5.60	trainoss	elliotsig
25	2.27e-01	4.87	trainoss	cs4
25	3.09e-01	3.33	trainoss	cs36
25	3.53e-01	4.47	traingdx	tansig
25	2.35e-01	38.17	traingdx	elliotsig
25	3.54e-01	4.53	traingdx	cs4
25	3.53e-01	4.47	traingdx	cs36

diabetes, 8-25-15-2

Παρακάτω δίνουμε πίνακες που συμπυκνώνουν την πληροφορία από τις προηγούμενες 4 σελίδες. Δηλαδή για κάθε υλοποίηση (δηλαδή πλήθους νευρώνων στον 1^ο κρυφό layer) παρουσιάζουμε το μέσο αριθμό εποχών (epochs) που απαιτήθηκαν για τη σύγκλιση.

	tanh	cs4	cs36	elliott
LeMa	3.67	3.37	3.50	4.40
BrFGS	6.20	6.53	6.20	8.23
ReBP	8.90	9.67	9.33	6.37
ScCG	8.37	8.67	8.83	6.47
CG-PB	6.43	6.43	6.70	6.43
FP-CG	7.47	7.33	7.47	7.63
PR-CG	7.30	7.10	7.37	6.27
OneSS	7.30	7.50	7.33	7.07
VLRBP	25.60	25.27	25.60	27.30

diabetes: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 8-10-15-2.

	tanh	cs4	cs36	elliott
LeMa	3.77	3.97	3.93	4.03
BrFGS	6.30	6.63	6.23	6.73
ReBP	11.03	9.27	9.97	10.83
ScCG	9.60	10.10	9.60	9.03
CG-PB	6.87	6.70	6.73	5.77
FP-CG	8.37	8.17	8.53	6.63
PR-CG	8.13	7.73	7.27	5.50
OneSS	7.57	7.17	8.00	7.67
VLRBP	25.83	30.30	28.30	40.00

diabetes: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 8-15-15-2.

	tanh	cs4	cs36	elliott
LeMa	3.30	3.77	3.73	3.73
BrFGS	4.97	6.33	4.03	7.40
ReBP	9.97	9.43	9.40	7.13
ScCG	6.70	5.23	6.53	2.73
CG-PB	5.67	7.87	6.80	6.50
FP-CG	5.63	7.53	6.60	7.23
PR-CG	3.63	5.23	4.83	4.83
OneSS	5.60	7.17	6.63	7.17
VLRBP	16.93	14.20	16.83	12.07

diabetes: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 8-20-15-2.

	tanh	cs4	cs36	elliott
LeMa	4.03	3.87	4.10	3.83
BrFGS	3.37	5.07	3.20	5.63
ReBP	6.17	8.50	6.17	8.67
ScCG	9.40	12.67	8.50	6.53
CG-PB	5.70	6.27	5.90	5.00
FP-CG	5.23	5.33	5.33	5.27
PR-CG	5.07	4.37	5.53	4.70
OneSS	3.70	4.87	3.33	5.60
VLRBP	4.47	4.53	4.47	38.17

diabetes: Μέσος όρος πλήθους εποχών (epochs) για τους 9 αλγόριθμους εκπαίδευσης και τις διάφορες συναρτήσεις μεταφοράς. Υλοποίηση 8-25-15-2.

Τέλος δίνουμε μια υπερσυμπύκνωση της πληροφορίας που παρατέθηκε εκτενώς παραπάνω. Έτσι στον παρακάτω πίνακα και για κάθε υλοποίηση (αριθμός νευρώνων στον πρώτο κρυφό layer) εμφανίζεται η μέση ποσοστιαία απόκλιση από την $\tanh x$ στον αριθμό των εποχών (epochs) για τις διάφορες συναρτήσεις μεταφοράς στις 36 περιπτώσεις.

diabetes	cs4	cs36	elliott
8 – 10 – 15 – 2	3,82	2,28	14,42
18 – 15 – 15 – 2	7,12	4,97	16,31
18 – 20 – 15 – 2	25,53	14,38	31,39
18 – 25 – 15 – 2	20,63	4,54	107,63

diabetes: Μέση ποσοστιαία απόκλιση στον αριθμό των εποχών (epochs) για τις διάφορες συναρτήσεις μεταφοράς στις 36 περιπτώσεις για κάθε υλοποίηση (πλήθος κρυφών νευρώνων)

Το Συμπέρασμα και σε αυτή την περίπτωση είναι υπεροχή της cs4 και μεγάλη απόκλιση για την elliotsig.

7

Εκπαίδευση στο αρχείο δεδομένων

`EV0real_time_synth.csv`

Εδώ έχουμε ένα αρχείο όπου έχουν καταγραφεί διάφορες μετρήσεις όπως φαίνονται στον πίνακα της επόμενης σελίδας [CPRAB19]. Το πλήθος των μετρήσεων είναι εξαιρετικά μεγάλο, 400000.

Εμείς εδώ επικεντρωθήκαμε σε ένα μικρό υποσύνολο με 324 μετρήσεις.

Έτσι μας ενδιαφέρουν οι ασθενείς που

- Τους χορηγείται το ΜΣΑΦ aspirin
- Βρίσκονται σε ήσυχο περιβάλλον
- Το πρόγραμμα ακουστικής βοήθειας είναι μέτριο

Από τις καταγραφές αυτές που σχετίζονται με τις στήλες SNR & MI προκύπτουν τετριμμένα (αφαίρεση) από τις άλλες και δεν τις χρησιμοποιούμε. Το γεωγραφικό στίγμα και η ώρα της μέτρησης απορρίπτονται επίσης.

Στο παράδειγμά μας επιλέγουμε τις τιμές από τα PTA4, lowSPL, midSPL, highSPL και θα προσεγγίσουμε τις τιμές fbSPL.

μεταβλητή	περιγραφή	τιμές
ID	Αναγνωριστικό	Ακέραιος 1:53
SoundClass	Μια τιμή που περιγράφει το περιβάλλον ήχου	QUIET, SPEECH, SPEECH-IN- NOISE, NOISE
hProg	Το επίπεδο ακουστικής βοήθειας	MEDIUM, LOW, HIGH, HIGH+
hVol	Μια τιμή που περιγράφει την ενεργή ένταση του ακουστικού	Ακέραιος -9:4
LonRel	Γεωγραφικό μήκος	Θέση GPS
LatRel	Γεωγραφικό πλάτος	Θέση GPS
LowSPL	Επίπεδο πίεσης ήχου (SPL) σε χαμηλές συχνότητες	Τιμή σε dB
MidSPL	SPL σε μεσαίες συχνότητες	Τιμή σε dB
HighSPL	SPL σε υψηλές συχνότητες	Τιμή σε dB
fbSPL	SPL σε όλο το εύρος	Τιμή σε dB
LowNf	The noise floor (Nf) σε χαμηλές συχνότητες	Τιμή σε dB
MidNF	Nf σε μεσαίες συχνότητες	Τιμή σε dB
HighNF	Nf σε υψηλές συχνότητες	Τιμή σε dB
fbNF	Nf σε όλο το εύρος	Τιμή σε dB
LowME	Το εύρος της διαμόρφωσης (ME) σε χαμηλές συχνότητες	Τιμή σε dB
MidME	ME σε μεσαίες συχνότητες	Τιμή σε dB
HighME	ME σε υψηλές συχνότητες	Τιμή σε dB
fbME	ME σε όλο το εύρος	Τιμή σε dB
TimeStab	Τοπική ώρα καταγραφής	Τιμή σε dB
LowSNR	$SNR = SPL - N$ σε χαμηλές συχνότητες	Τιμή σε dB
MidSNR	$SNR = SPL - N$ σε μεσαίες συχνότητες	Τιμή σε dB
HighSNR	$SNR = SPL - N$ σε υψηλές συχνότητες	Τιμή σε dB
fbSNR	$SNR = SPL - N$ σε όλο το εύρος	Τιμή σε dB
LowMI	$MI = ME - Nf$ σε χαμηλές συχνότητες	Τιμή σε dB
MidMI	$MI = ME - Nf$ σε μεσαίες συχνότητες	Τιμή σε dB
HighMI	$MI = ME - Nf$ σε υψηλές συχνότητες	Τιμή σε dB
fbMI	$MI = ME - Nf$ σε όλο το εύρος	Τιμή σε dB
PTA4	Μέσος καθαρός τόνος (PTA) διαμέσου 4 συχνοτήτων (0.5, 1, 2, and 4kHz) στο καλύτερο αυτί	Τιμή σε dB
Drug	Το φάρμακο που χρησιμοποιήθηκε	

Πίνακας περιγραφής των δεδομένων του EVOreal_time_synth.csv

Ακολουθεί ο πίνακας με τα δεδομένα.

PTA4	lowSPL	midSPL	highSPL	fbSPL
0	33.2352036	23.45155276	18.98592268	28.65594954
0	24.63545426	18.51825837	13.56495234	31.26062272
0	28.44132402	24.57958772	25.49695655	30.72674411
0	32.61356358	20.76651781	14.12574919	30.64497191
0	39.77132682	29.60501359	24.68978522	38.96398506
0	39.46566074	25.57462549	17.9357269	39.31384287
0	25.06484654	17.01273848	17.66288876	28.79067454
0	31.11222134	24.84241802	24.00142836	36.05819791
0	42.96758171	37.12096169	31.55368493	44.83969595
0	50.57629196	40.86208925	34.07094104	52.19302853
0	34.19574757	29.00514031	31.12052593	33.63772561
0	39.00930274	31.37700667	15.79841933	39.92532286
0	26.13640737	23.52844103	21.2760734	31.34186477
0	44.422814	27.05711681	13.98912041	44.37668256
0	43.23752136	16.69948111	13.71367196	41.83352075
0	51.85014991	35.1816338	19.35776055	50.6656946
0	28.17584144	19.25289214	21.73497364	22.92427772
0	45.27182463	41.22182983	23.23289688	48.39863195
0	38.01880993	21.12610567	21.53799818	37.62528929
0	26.7969098	19.91511157	16.08908545	29.84306343
0	31.04610594	26.16522511	20.04128571	33.48883825
0	31.23264991	22.20678343	20.03222227	27.41750442
0	30.92472794	23.01634842	17.34225785	33.59938978
0	26.9696579	21.18886856	24.07313669	30.52347726
0	76.06303624	46.57381306	39.24569129	69.12651417
0	30.0220309	22.13461384	17.62439595	28.7062128
0	52.16413695	42.3494005	38.33316847	50.75453495
12.5	36.3645641	15.83895841	22.30950553	33.2207522
12.5	66.72907628	41.45518988	30.09005772	61.99977197
12.5	47.20498358	26.68715971	19.43174989	46.95347007
12.5	37.30554554	18.53828077	20.4777053	40.70752276
12.5	41.27580877	26.89363956	19.75459215	37.08662828
12.5	25.6822546	21.15429481	16.47357813	23.68277784
13.75	32.43651349	23.52064722	14.53976674	31.32861177
13.75	40.94361675	32.20593885	28.04832044	42.13377221
13.75	67.22157761	47.40466241	41.9046049	64.17180697
13.75	32.66062459	21.84183072	19.17389612	34.83146562
13.75	53.00602198	42.55061364	31.08592011	50.81442649
15	42.31002718	28.138554	20.18684064	42.55426651
15	44.93468204	39.55601896	27.75294714	48.61453886
15	50.32098569	48.44986729	47.1448442	52.21101364
15	24.78889174	22.02538298	15.81184197	29.80672548
15	45.46035898	28.71204939	19.30396717	48.24663382
16.25	39.4988796	39.56383556	34.95981189	44.60130204
16.25	36.82717501	27.00887551	18.51980211	40.89669005
18.75	39.67362946	37.08782525	17.94141087	44.05302493
18.75	72.70765216	65.20666605	49.80469791	70.34357491
18.75	52.59598453	40.13626121	30.61448567	48.68438995
21.25	39.81625226	28.2620212	24.88099476	41.67113403
22.5	31.86542878	18.13684101	17.5288013	33.77107091
22.5	23.38749951	16.80698855	15.57234145	25.22926092
22.5	31.61940987	15.94223787	14.5209566	31.37874834
22.5	21.74757267	16.39940756	17.51224701	26.24775478
25	41.03160802	30.96675796	32.43690769	41.41574849
25	62.83623282	37.47886306	29.77586623	66.95421875

26.25	24.85472022	22.26452916	16.07796117	30.51134915
26.25	50.1553728	45.61173774	24.58300493	53.95873418
26.25	64.81916925	57.78109029	45.29376694	65.80638479
26.25	48.05908035	34.11988044	29.54555793	51.45488186
26.25	41.83250295	35.95586637	22.07443087	39.33725516
26.25	42.03658755	30.05572433	30.3094627	40.68901837
26.25	42.19500993	24.61291794	14.19485744	44.11302191
26.25	51.62230691	17.64773153	25.5984887	50.23895651
26.25	50.5374644	40.56106599	21.65745219	51.86803571
26.25	37.22465993	27.99748951	13.79854809	39.70304419
26.25	43.39381987	34.98745279	21.88134769	44.25157152
26.25	39.81488003	30.37773624	24.56618339	47.50610191
26.25	41.19684349	23.94973971	16.0672229	41.50714865
26.25	40.66076859	36.82914122	23.67275572	44.28876722
26.25	50.05983704	40.68975591	32.5298392	50.71250164
26.25	39.0810826	30.20687351	29.24032626	44.55907734
26.25	41.23908492	29.63554945	24.46607415	42.51852378
26.25	46.83309714	48.00160651	20.04742344	50.47503871
26.25	36.9279332	25.34424604	25.70984386	33.15236862
26.25	55.23471151	55.37660551	64.42159041	56.46360904
26.25	46.54539438	38.757275	29.40957247	51.28162448
26.25	30.596692	39.73334192	28.40766703	47.124317
26.25	34.20150882	26.9950792	20.1979805	38.08938438
26.25	36.64559194	28.98712456	30.3729747	37.98748939
26.25	47.93443658	46.56269307	30.1980435	59.66506543
26.25	38.77912894	30.32599615	22.11610426	43.1477854
26.25	59.93696964	50.67120585	43.14385225	60.04049606
26.25	56.18300407	45.10558812	40.11000278	55.08384932
26.25	32.33705616	28.15658921	23.0682852	34.21240314
26.25	40.8510217	37.77435226	38.46099499	41.98696426
28.75	40.05166713	35.6395524	28.44513137	49.05238191
28.75	48.52792198	40.71950167	33.39695164	53.01860442
28.75	47.04876667	41.60275743	35.02862958	49.20148966
28.75	55.42067478	44.45909295	27.84302051	56.20927221
28.75	45.93446431	39.97849722	34.22978584	55.15384298
30	36.5220127	25.03127741	21.52879428	40.76913549
30	42.65976228	34.43521672	29.71660648	41.7607738
30	30.22011698	40.87559293	32.08490007	50.92169735
30	45.40850239	29.35156329	14.70873016	49.63042175
30	43.73553731	29.52452487	19.50833549	46.85867323
30	35.42012727	25.52368536	28.34947455	37.35118747
30	47.56517743	30.32631057	19.64598589	47.95521068
30	25.941275	15.12988565	21.87775069	30.25519602
30	41.86605264	36.1755877	21.72526045	42.02564427
30	53.90116366	43.10716464	28.40592319	55.59076355
30	43.6600868	45.47892802	56.63002958	49.1205608
30	33.35871415	23.00217694	22.26067809	34.45367311
36.25	47.78019598	41.30510416	23.62735975	46.09507347
36.25	41.6849803	21.13989841	15.58775562	40.87415364
36.25	25.3866339	17.61877797	16.81021849	24.6058616
36.25	41.82240243	36.3306719	33.0766955	41.30754433
36.25	29.84383374	28.99578273	23.52238364	33.63078751
36.25	41.52318022	28.37760499	18.20357324	47.14855325
37.5	50.30046037	29.87181114	25.86674268	51.47122886
40	43.36459516	35.13922981	30.01898555	41.49660402
40	28.80340486	16.46607191	13.93342202	26.8365901
40	34.96935786	27.02928125	20.74474651	32.30864186
40	29.33033084	19.47836868	16.93740441	27.42621161

40	41.41362006	24.37191188	20.6239786	42.11889274
40	27.52104888	22.68427199	18.60874803	29.15773998
40	20.86312349	66.4721374	44.63156648	83.00314014
40	32.78576459	23.22872478	21.31830712	28.12839279
40	44.08208235	33.78224243	31.84194823	42.88331044
40	48.62678169	34.97363193	31.92633344	52.38109286
40	31.67567406	18.39824296	20.49467394	27.92063855
40	44.81525836	37.22312208	37.89099306	45.82491522
40	44.52757725	26.62035756	18.76503151	42.58007445
40	42.42934309	32.88995303	29.12439481	44.62987824
40	74.23372402	45.9120778	36.54236288	76.26143633
40	36.49361468	24.08556696	22.94872997	39.96088411
40	57.71620577	36.49692413	23.23459782	62.15939001
40	30.85456188	24.14611057	21.48125825	34.80289794
40	25.40220389	27.24131931	18.1723996	29.09373509
40	36.63586444	31.72750196	22.72661864	39.75921377
40	47.26845587	32.32331247	21.38891336	45.61551761
40	40.61202218	34.41139408	32.26497734	43.18756592
40	26.27731326	21.77744828	21.59840885	27.65231186
40	62.10040536	48.1123649	17.56514718	60.90905634
40	32.45836067	27.38918268	34.83490934	36.26800659
40	36.88281701	21.71357034	24.99094123	39.32872622
40	38.41352947	26.02345062	18.19827897	40.92740273
40	25.65058703	15.45775902	19.91579767	27.91287004
40	45.99709772	33.05178025	27.06076406	48.17132186
40	25.37574429	16.30682463	14.46911038	30.69613577
40	22.46178744	17.08691425	24.35668427	26.93930206
40	29.43176254	27.35490741	18.56080462	34.13295394
40	27.97974971	15.89796433	31.37270113	26.73966571
40	27.68392304	20.63442076	21.18756736	30.02173005
40	39.82563398	34.3675144	29.77045677	43.77193301
40	43.22028445	25.92887238	25.31896882	40.35539023
40	36.49293581	39.75472103	36.68699158	44.49543039
40	49.12089463	40.74254117	28.88167222	47.85458442
40	54.180307	22.29653201	15.31364931	58.61653593
40	44.8297471	31.04586496	25.90879827	42.02028698
40	32.92661234	26.40114315	24.40128332	32.98197764
41.25	30.17428471	26.55678746	25.94239342	30.71693739
41.25	28.25952101	19.86787186	18.90438937	31.12269334
41.25	36.28911715	31.26466491	28.92776171	36.49389705
41.25	29.77063377	33.44511399	23.15369679	36.51954622
41.25	26.51275098	19.06400335	16.91824435	30.44369221
41.25	23.41880191	16.40938121	14.04522614	25.96815022
42.5	38.08961392	27.71087616	21.41482225	33.50258681
46.25	42.77058369	36.90229717	24.61202816	44.18148992
46.25	39.68550657	36.64801421	33.81695669	42.66083898
46.25	34.84492198	21.27741958	16.14729713	36.85996446
46.25	25.81491916	22.28845809	14.82469415	29.73263604
47.5	60.20067345	53.53520297	38.80850353	57.8148005
50	52.84652421	50.89686015	34.84647146	58.80654447
50	48.92392997	44.20559197	32.25996611	51.72380533
50	42.64245773	34.99993177	28.48504269	44.8143486
50	24.81316966	16.05416293	15.07345016	25.39826768
50	50.78338869	40.57869401	43.57225601	50.18610155
50	47.49183578	26.77187033	24.1907235	49.57751463
50	51.67335565	29.36313468	25.78877813	49.74186598
53.75	39.54456132	34.40984968	31.86712102	41.85091755
53.75	27.64077531	25.74533054	16.67086617	27.43591411

53.75	49.7652252	40.91607311	25.26271345	48.78045415
53.75	59.31092711	48.26844813	31.38146507	63.48365733
53.75	40.45826672	34.42135791	23.19673366	37.27642075
53.75	36.31358402	21.95108529	14.14166516	36.52217951
53.75	55.3768758	46.12296374	33.03477672	58.86784814
53.75	32.9955056	23.75472706	29.35978677	29.44777198
53.75	38.63252638	26.60369418	19.84126712	39.35191734
56.25	30.4615465	19.62282364	14.52801474	27.52756015
56.25	38.69749899	26.90024091	29.61572057	39.08255135
56.25	47.96810914	36.29777402	20.21512055	46.68629614
56.25	42.22335074	33.39301548	37.84936339	44.26458105
56.25	45.5948271	43.45212415	32.77986467	46.44563292
57.5	55.95766611	43.1483759	32.61264671	53.32536298
57.5	35.25038292	37.79933975	27.53463732	47.62626184
57.5	31.05888858	24.75804614	21.27154262	35.32024729
57.5	43.51184003	34.77277187	24.69442931	45.84651743
57.5	37.04825815	19.72442197	21.44991514	40.5736377
57.5	39.80587599	32.16856398	22.22530028	45.46165388
57.5	49.44527107	40.05418324	39.23583493	47.77298942
57.5	47.73644127	29.8704313	26.85264471	44.60103122
57.5	39.23631863	26.508628	17.96666415	38.95441524
57.5	56.02904996	46.24109175	38.67276317	55.06790663

Στο επόμενο πρόγραμμα υλοποιούμε διάφορα δίκτυα και ελέγχουμε την αποτελεσματικότητα των συναρτήσεων σε αυτά.

```
x=aspirin(:,1:4)';
t=aspirin(:,5)';
trainss={'trainlm','trainbfg','trainrp','trainscg','traincgb', ...
         'traincgf','traincgp','trainoss','traingdx'}; ...
funfcn={'tansig','elliotsig','cs4','cs36'};
for j0=5:5:20,
    for j1=trainss,
        errorsum=zeros(4,1);epochsum=zeros(4,1);
        for j3=1:10,
            j22=0;
            for j2=funfcn,
                seedRandom=rng(1,'twister');
                myNet = feedforwardnet(j0,char(j1));
                myNet=configure(myNet,x,t);myNet=init(myNet);
                myNet.b{1}(1)=0.25*j3*(-1)^j3;
                myNet.b{1}(end)=-myNet.b{1}(1);
                myNet.trainParam.showWindow = false;
                myNet.layers{1}.transferFcn = char(j2);
                myNet.trainParam.goal=0.0;
                [net,tr] = train(myNet,x,t);
                j22=j22+1;
                epochsum(j22)=epochsum(j22)+length(tr.epoch);
                errorsum(j22)=errorsum(j22)+tr.best_vperf;
            end;
        end;
    end;
    fprintf('%6.0f %10.2e %10.2f %11s %11s\n', ...
           j0,errorsum(1)/30,epochsum(1)/30,j1{1},funfcn{1});
    fprintf('%6.0f %10.2e %10.2f %11s %11s \n', ...
           j0,errorsum(2)/30,epochsum(2)/30,j1{1},funfcn{2});
    fprintf('%6.0f %10.2e %10.2f %11s %11s \n', ...
           j0,errorsum(3)/30,epochsum(3)/30,j1{1},funfcn{3});
    fprintf('%6.0f %10.2e %10.2f %11s %11s \n', ...
           j0,errorsum(4)/30,epochsum(4)/30,j1{1},funfcn{4});
end;
end;
```

n	σφάλμα	epochs	algorithm	tr. Function
3	2.92E+00	7.4	trainlm	tansig
3	2.58E+00	13.77	trainlm	elliotsig
3	2.60E+00	6.9	trainlm	cs4
3	2.90E+00	6.47	trainlm	cs36
3	2.74E+00	11.43	trainbfg	tansig
3	3.16E+00	11.63	trainbfg	elliotsig
3	2.43E+00	14.07	trainbfg	cs4
3	2.67E+00	12.67	trainbfg	cs36
3	3.14E+00	16.87	trainrp	tansig
3	3.89E+00	13.67	trainrp	elliotsig
3	3.24E+00	13.2	trainrp	cs4
3	3.04E+00	18.13	trainrp	cs36
3	4.61E+00	7.4	trainscg	tansig
3	4.27E+00	10.03	trainscg	elliotsig
3	4.37E+00	8.2	trainscg	cs4
3	4.19E+00	9.2	trainscg	cs36
3	3.77E+00	9.27	traincgb	tansig
3	4.69E+00	6.93	traincgb	elliotsig
3	3.56E+00	9.53	traincgb	cs4
3	3.70E+00	10.63	traincgb	cs36
3	4.54E+00	7.57	traincgf	tansig
3	6.08E+00	7.07	traincgf	elliotsig
3	6.50E+00	6.9	traincgf	cs4
3	4.62E+00	7.9	traincgf	cs36
3	4.04E+00	9.57	traincgp	tansig
3	4.33E+00	9.53	traincgp	elliotsig
3	4.33E+00	7.07	traincgp	cs4
3	4.22E+00	7.5	traincgp	cs36
3	4.39E+00	8.5	trainoss	tansig
3	4.72E+00	8.4	trainoss	elliotsig
3	4.56E+00	7.33	trainoss	cs4
3	4.40E+00	8.77	trainoss	cs36
3	7.26E+00	26.5	traingdx	tansig
3	1.02E+01	21.63	traingdx	elliotsig
3	1.62E+01	20.63	traingdx	cs4
3	7.26E+00	26.5	traingdx	cs36

Υλοποίηση 3-3-1

n	σφάλμα	epochs	algorithm	tr. Function
4	2.90E+00	5.33	trainlm	tansig
4	2.49E+00	4.8	trainlm	elliotsig
4	2.29E+00	6.73	trainlm	cs4
4	2.45E+00	6.37	trainlm	cs36
4	5.04E+00	72.13	trainbfg	tansig
4	3.35E+00	5.97	trainbfg	elliotsig
4	6.40E+00	7.47	trainbfg	cs4
4	4.76E+00	51.17	trainbfg	cs36
4	4.25E+00	11	trainrp	tansig
4	4.26E+00	11.77	trainrp	elliotsig
4	3.69E+00	14.17	trainrp	cs4
4	4.30E+00	10.47	trainrp	cs36
4	4.07E+00	6.97	trainscg	tansig
4	4.63E+00	5.9	trainscg	elliotsig
4	4.76E+00	6.2	trainscg	cs4
4	4.09E+00	6.57	trainscg	cs36
4	3.93E+00	6.07	traincgb	tansig
4	4.01E+00	5.53	traincgb	elliotsig
4	4.30E+00	4.97	traincgb	cs4
4	3.89E+00	6.07	traincgb	cs36
4	3.89E+00	7.13	traincgf	tansig
4	4.17E+00	6.2	traincgf	elliotsig
4	4.37E+00	5.73	traincgf	cs4
4	3.93E+00	7	traincgf	cs36
4	4.23E+00	6.17	traincgp	tansig
4	4.27E+00	5.87	traincgp	elliotsig
4	4.18E+00	6.23	traincgp	cs4
4	4.35E+00	5.63	traincgp	cs36
4	4.27E+00	5.87	trainoss	tansig
4	4.78E+00	5.77	trainoss	elliotsig
4	4.53E+00	5.8	trainoss	cs4
4	4.06E+00	6.13	trainoss	cs36
4	6.86E+00	24.13	traingdx	tansig
4	6.95E+00	22.17	traingdx	elliotsig
4	6.89E+00	24.13	traingdx	cs4
4	6.86E+00	24.13	traingdx	cs36

Υλοποίηση 3-4-1

n	σφάλμα	epochs	algorithm	tr. Function
5	2.56E+00	5.47	trainlm	tansig
5	2.69E+00	5.6	trainlm	elliotsig
5	2.59E+00	5.4	trainlm	cs4
5	2.41E+00	4.4	trainlm	cs36
5	2.96E+00	42.1	trainbfg	tansig
5	2.35E+00	4.9	trainbfg	elliotsig
5	4.73E+00	8.63	trainbfg	cs4
5	3.44E+00	19.4	trainbfg	cs36
5	2.82E+00	14.77	trainrp	tansig
5	2.80E+00	13.63	trainrp	elliotsig
5	2.73E+00	15	trainrp	cs4
5	2.71E+00	14.87	trainrp	cs36
5	2.48E+00	6.6	trainscg	tansig
5	2.26E+00	5.43	trainscg	elliotsig
5	2.44E+00	6.27	trainscg	cs4
5	2.38E+00	6.7	trainscg	cs36
5	2.18E+00	5.5	traincgb	tansig
5	2.18E+00	5.83	traincgb	elliotsig
5	2.19E+00	5.7	traincgb	cs4
5	2.18E+00	5.63	traincgb	cs36
5	2.10E+00	6.5	traincgf	tansig
5	2.23E+00	6.13	traincgf	elliotsig
5	2.15E+00	6.6	traincgf	cs4
5	2.13E+00	6.33	traincgf	cs36
5	2.10E+00	5.97	traincgp	tansig
5	2.72E+00	5.2	traincgp	elliotsig
5	2.29E+00	5.6	traincgp	cs4
5	2.12E+00	5.67	traincgp	cs36
5	2.53E+00	6.47	trainoss	tansig
5	2.50E+00	6.67	trainoss	elliotsig
5	2.34E+00	6.8	trainoss	cs4
5	2.55E+00	6.17	trainoss	cs36
5	3.66E+00	29.1	traingdx	tansig
5	2.73E+00	32.83	traingdx	elliotsig
5	3.46E+00	27.93	traingdx	cs4
5	3.66E+00	29.1	traingdx	cs36

Υλοποίηση 3-5-1

n	σφάλμα	epochs	algorithm	tr. Function
6	4.80E+00	5.77	trainlm	tansig
6	3.95E+00	4.83	trainlm	elliotsig
6	7.66E+00	6.07	trainlm	cs4
6	4.43E+00	5.93	trainlm	cs36
6	4.65E+00	31.8	trainbfg	tansig
6	4.96E+00	4.9	trainbfg	elliotsig
6	4.50E+00	39.53	trainbfg	cs4
6	4.51E+00	32.67	trainbfg	cs36
6	4.82E+00	10.33	trainrp	tansig
6	5.27E+00	8.57	trainrp	elliotsig
6	4.63E+00	9.8	trainrp	cs4
6	4.73E+00	10.93	trainrp	cs36
6	4.58E+00	4.77	trainscg	tansig
6	5.43E+00	3.67	trainscg	elliotsig
6	4.54E+00	4.7	trainscg	cs4
6	4.58E+00	4.77	trainscg	cs36
6	4.10E+00	4.4	traincgb	tansig
6	5.15E+00	3.47	traincgb	elliotsig
6	4.08E+00	4.67	traincgb	cs4
6	4.12E+00	4.2	traincgb	cs36
6	4.32E+00	4.43	traincgf	tansig
6	5.18E+00	3.83	traincgf	elliotsig
6	4.26E+00	4.6	traincgf	cs4
6	4.32E+00	4.53	traincgf	cs36
6	4.44E+00	4.13	traincgp	tansig
6	5.19E+00	3.7	traincgp	elliotsig
6	4.17E+00	5.17	traincgp	cs4
6	4.43E+00	4.23	traincgp	cs36
6	4.73E+00	6.17	trainoss	tansig
6	1.32E+01	4.7	trainoss	elliotsig
6	5.51E+00	4.97	trainoss	cs4
6	5.28E+00	5.8	trainoss	cs36
6	1.80E+02	17.13	traingdx	tansig
6	5.27E+00	19.53	traingdx	elliotsig
6	7.43E+00	24.37	traingdx	cs4
6	1.80E+02	17.13	traingdx	cs36

Υλοποίηση 3-6-1

n	σφάλμα	epochs	algorithm	tr. Function
7	2.92E+00	5.37	trainlm	tansig
7	3.52E+00	4.97	trainlm	elliotsig
7	2.84E+00	5.07	trainlm	cs4
7	2.75E+00	4.7	trainlm	cs36
7	3.44E+00	278.4	trainbfg	tansig
7	3.55E+00	144.97	trainbfg	elliotsig
7	4.06E+00	184	trainbfg	cs4
7	3.26E+00	261.07	trainbfg	cs36
7	3.00E+00	14.17	trainrp	tansig
7	2.84E+00	13.13	trainrp	elliotsig
7	3.42E+00	12.2	trainrp	cs4
7	2.89E+00	15.23	trainrp	cs36
7	3.43E+00	13.53	trainscg	tansig
7	2.66E+00	13	trainscg	elliotsig
7	3.33E+00	12.87	trainscg	cs4
7	3.71E+00	11.6	trainscg	cs36
7	3.77E+00	8.6	traincgb	tansig
7	2.90E+00	8.5	traincgb	elliotsig
7	2.99E+00	12.37	traincgb	cs4
7	3.60E+00	10.2	traincgb	cs36
7	3.15E+00	13.1	traincgf	tansig
7	2.83E+00	11.37	traincgf	elliotsig
7	3.05E+00	15.87	traincgf	cs4
7	3.05E+00	14.3	traincgf	cs36
7	3.19E+00	13.6	traincgp	tansig
7	3.24E+00	9.73	traincgp	elliotsig
7	3.26E+00	12.43	traincgp	cs4
7	3.30E+00	12.87	traincgp	cs36
7	3.61E+00	13.9	trainoss	tansig
7	3.22E+00	10.53	trainoss	elliotsig
7	3.14E+00	13.8	trainoss	cs4
7	3.26E+00	14.57	trainoss	cs36
7	1.09E+01	22.53	traingdx	tansig
7	8.98E+00	19.13	traingdx	elliotsig
7	9.79E+00	24.07	traingdx	cs4
7	1.03E+01	25.43	traingdx	cs36

Υλοποίηση 3-7-1

Κατ' αντιστοιχία των προηγούμενων δίνουμε μια υπερσυμπύκνωση της πληροφορίας που παρατέθηκε εκτενώς παραπάνω. Έτσι στον παρακάτω πίνακα και για κάθε υλοποίηση (αριθμός νευρώνων στον πρώτο κρυφό layer) εμφανίζεται η μέση ποσοστιαία απόκλιση από την $\tanh x$ στον αριθμό των εποχών (epochs) για τις διάφορες συναρτήσεις μεταφοράς στις 36 περιπτώσεις.

υλοποίηση	cs4	cs36	elliott
4 – 3 – 1	15.12	10.01	21.57
4 – 4 – 1	21.74	8.24	17.85
4 – 5 – 1	11.98	10.03	17.41
4 – 6 – 1	14.77	2.95	24.88
4 – 7 – 1	15.49	10.14	16.53

Μέση ποσοστιαία απόκλιση στον αριθμό των εποχών (epochs) για τις διάφορες συναρτήσεις μεταφοράς στις 5 περιπτώσεις για κάθε υλοποίηση (πλήθος κρυφών νευρώνων)

Διαπιστώνουμε ότι η προσέγγιση cs36 υπερέχει σαφώς.

Σύνοψη και συμπεράσματα

Το βασικό συμπέρασμα είναι ότι μπορούμε να προσεγγίσουμε τις συναρτήσεις μεταφοράς σε ένα νευρωνικό δίκτυο με φθηνότερο τρόπο. Αντί να χρησιμοποιούμε αριθμητικές προσεγγίσεις με μεγάλη ακρίβεια που απαιτούν πολλές πράξεις (πολλαπλασιασμούς & διαιρέσεις) μπορούμε να κάνουμε την προσέγγιση με κυβικές splines που χρησιμοποιούν κυβικά πολώνυμα μόνο.

Στην παρούσα διπλωματική βρήκαμε τρόπους ανάπτυξης σχετικών προσεγγίσεων αποφεύγοντας τα προβλήματα που ανακύπτουν (π.χ. συνέχειας, μονοτονίας, κ.λπ.). Τα πρώτα αποτελέσματα ήταν άκρως ενθαρυντικά.

Μελλοντικές επεκτάσεις

Η έρευνα μπορεί να επεκταθεί σε δυο άξονες.

- I. Εύρεση βέλτιστων σημείων της spline για προσέγγιση της $\tanh x$ με μεγαλύτερη ακρίβεια.
- II. Εφαρμογή σε άλλες συναρτήσεις μεταφοράς όπως logsig , κλπ.

Βιβλιογραφία

- [UG19] M. H. Beale, M. T. Hagan and H. B. Demuth, Deep Learning Toolbox™ Users Guide, version 2019b, The Mathworks, Inc., Natick, MA.
- [RG19] M. H. Beale, M. T. Hagan and H. B. Demuth, Deep Learning Toolbox™ Reference Guide, version 2019b, The Mathworks, Inc., Natick, MA.
- [CPRAB19] Jeppe H. Christensen, Niels H. Pontoppidan, Rikke Rossing, Marco Anisetti, Doris-Eva Bamiou, George Spanoudakis, Louisa Murdin, Thanos Bibas, Dimitris Kikidiks, Nikos Dimakopoulos, Giorgos Giotis and Apostolos Ecomomou, Fully Synthetic Longitudinal Real-World Data From Hearing Aid Wearers for Public Health Policy Modeling, *Front. Neurosci.*, 2019, <https://doi.org/10.3389/fnins.2019.00850>
- [CW80] W. J. Cody and W. Waite, Software manual for the elementary functions, Prentice--Hall inc., Englewood cliffs, N.J., 1980.
- [EII93] D.L. Elliott, A better Activation Function for Artificial Neural Networks, ISR Technical Report TR 93-8, University of Maryland, College Park, MD 20742, 1993.
- [GKK16] Giokas, K., Katrakazas, P., Koutsouris, D. “A Collaborative m-Health Platform for Evidence-Based Self-Management and Detection of Chronic Multi morbidity Development and Progression” In: *M - Health Innovations for Patient - Centered Care*, chapter 3, pp. 52-71, A. Moutzoglou (Editor), IGI Eds, 2016.
- [KKGK16] Kalogeropoulou, M.; Katrakazas, P., Giokas, K.; Koutsouris, D. “A Nanobots-based Methodology for Assessing the Acoustic Signal's Attenuation in the Auditory System's Pathway” In: *SAN2016 Meeting*, Corfu, Greece. October, 2016

- [KGK15] Katrakazas, P., Giokas, K., Koutsouris D. “Mobile-Cloud Platform for Raising Hearing Loss Awareness”, ACM - International Conference on Pervasive Technologies Related to Assistive Environments - PETRA15, Proceedings, art. no. a91, Corfu, Greece, July 2015.
- [KMK19] P. Katrakazas, O. Manta, and D. Koutsouris, “A Big-Data Informed Model Approach to Hearing Health Policy Decision Making,” in 2018 14th International Conference on Signal - Image Technology Internet - Based Systems (SITIS), 2018, pp. 725–729, doi: 10.1109/SITIS.2018.00116.
- [KMK18] P. Katrakazas, O. Manta and D. Koutsouris, "A Big-Data Informed Model Approach to Hearing Health Policy Decision Making," 2018 14th International Conference on Signal Image Technology & Internet - Based Systems (SITIS), Las Palmas de Gran Canaria, Spain, 2018, pp. 725-729. doi: 10.1109/SITIS.2018.00116
- [KPBK20] Katrakazas, P., Pasiadis, K., Bibas A., Koutsouris, D., (2019) A General Systems Theory Approach in Public Hearing Health: Lessons learned from a Systematic Review of General Systems Theory in Healthcare. *IEEE Access*, vol. 8, pp. 53018-53033, 2020
- [KTMBK17] Katrakazas, P., Trenkova, L., Milas, J., Brdaric, D., Koutsouris, D. “The EVOTION decision support system: Utilizing it for public health policy-making in hearing loss”, *Studies in Health Technology and Informatics*, Househ M.S.,Mantas J.,Hasman A.,Gallos P (Eds), vol. 238, pp. 88 – 91, Elsevier, IOS Press, 2017.
- [KK18] P. Katrakazas and D. Koutsouris, "A (Lack of) Review on Cyber-Security and Privacy Concerns in Hearing Aids," 2018 IEEE 31st International Symposium on Computer - Based Medical Systems (CBMS), Karlstad, 2018, pp. 223-226. doi: 10.1109/CBMS.2018.00046
- [KTGK17] Katrakazas, P., Tarousi, M., Giokas, K., Koutsouris, D. “Analysis of Cervical cancer: an Evidence-based treatments” In: 30 th IEEE International Symposium on Computer - Based Medical Systems - IEEE CBMS 2017, Thessaloniki, Greece. June 2017.
- [LBKKIK16] Lambrou, G.I; Braoudaki, M; Katrakazas, P.; Kouris, I.; Iliopoulou, D.; Koutsouri, Tz.; Petropoulou, O.; Koutsouris, D. “Complex Dynamics in Tumor Gene Regulatory Networks: Oncogenesis Dynamics Driven by “Genes Gone Crazy” XIV Mediterranean Conference on Medical and

- Biological Engineering and Computing 2016, MEDICON 2016, vol. 57 of the series IFMBE Proceedings, pp, 501-505, Springer International Publishing, Cyprus, March-April 2016.
- [Matlab] MATLAB version R2019b, The Mathworks, Inc., Natick, MA.
- [Mol94] C. Moler, Available from:
https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/347/versions/2/previews/tanh_v5_2.m/index.html
- [MKPAI18] S. Michael, P. Katrakazas, O. Petronoulou, A. Anastasiou, D. Iliopoulou and D. Dionisios Koutsouris, "Nutrition Buddy: a Childhood Obesity Serious Game," 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, 2018, pp. 5-8. doi: 10.1109/WorldS4.2018.8611473
- [Mu16] J-M. Muller, Elementary Functions. Algorithms and implementation, Birkhauser, 3rd edition, NY, 2016.
- [Netlib] Netlib, http://www.netlib.org/fdlibm/s_tanh.c
- [PIMA] PIMA: <https://www.kaggle.com/uciml/pima-indians-diabetes-database#diabetes.csv>
- [PKK17] Pitsaris, M. Katrakazas, P. Koutsouris D. "Design and Development of a Low-Budget Infrasonic Detector", In: Joint conference of the European Medical and Biological Engineering Conference (EMBEC 17) and the Nordic - Baltic cinference in Biomedical Engineering and Medical Physics(NBC 17),Tampere, Finland, June, 2017.
- [PKGK15] K. Plakas, P. Katrakazas, K. Giokas, D. Koutsouris, "Inner-ear modelling, with the use of FEM", 6th Panhellenic Conference on Biomedical Technology, p. 41, Athens, May 2015
- [PKGKB16] Plakas, K., Katrakazas, P., Giokas, K., Kouris, I., Bibas, A., Koutsouris, D.D. "Otosclerosis and Tympanosclerosis Modeling Using the Finite Element Method", XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016, MEDICON 2016, vol. 57 of the series IFMBE Proceedings, pp, 720-724, Springer International Publishing, Cyprus, March-April 2016.
- [Pulu92] Purdie, N., E.A. Lucas, and M.B. Talley, "Direct measure of total cholesterol and its distribution among major serum lipoproteins," *Clinical*

- Chemistry*, Vol. 38, No. 9, 1992, pp. 1645–1647.
- [SKBKK17] Spanoudakis, G., Kikidis, D., Bibas, A., Katrakazas, P., Koutsouris, D. Pontopidan, N. H. “Public health policy for management of hearing impairments based on big data analytics: EVOTION at Genesis”. In: 17th IEEE International Bio - Informatic s and Bio Engineering Conference, BIBE 2017, October, Washington D.C., USA, 2017.
- [TKLPKS17] Tietz, L.H.B., Katrakazas, P., Laplante-Lévesque, A., Pontoppidan, N.H., Koloutsou, N., Spanoudakis, G., Koutsouris, D. “Associations between hearing performance and physiological measures - An overview and outlook”, *Studies in Health Technology and Informatics*, Househ M.S.,Mantas J.,Hasman A.,Gallos P (Eds), vol. 238, pp. 100 – 103, Elsevier, IOS Press, 2017.
- [TT19] A. Ch. Tsitoura and Ch. Tsitouras, Cubic Spline Approximation of Transfer Functions for Speeding Neural Networks Performances, 17th International Conference of Numerical Analysis and Applied Mathematics, Rhodes, September 2019 (AIP Conf. Proc., to appear).
- [TK14] Ch. Tsitouras and V. N. Katsikis, Bounds for Variable Degree Rational L_∞ Approximations to the Matrix Cosine, *Comput. Phys. Comm.* 185 (2014) 2834–2840.
- [TF18] Ch. Tsitouras and I. Th. Famelis, Bounds for Variable Degree Rational L_∞ Approximations to the Matrix Exponential, *Appl. Maths. Comput.*, 338 (2018) 376--386.
- [VPCMK17] Voutyrakou, D.A., Papanastasis, A., Chatsikian, M., Katrakazas, P., Koutsouris, D. “Transoral Robotic Surgery (TORS) Advantages and Disadvantages: A Narrative Review”. *The Journal of Engineering*, IET Digital Library, DOI: 10.1049/joe.2017.0409, 2017.