



**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

**ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ
(ΕΡΓΑΣΤΗΡΙΟ)**

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2022-2023

4^η Σειρά ασκήσεων

Όνομ/νυμα και αρ.μητρώου

Αθανασίου Ελένη 19387004

Βαβαΐτη Κωνσταντίνα 18387257

Εργαστηριακή ομάδα

Ομάδα Γ

Αιγάλεω 23/05/2023

1. Σκοπός και περίληψη της άσκησης

Σκοπός του πρώτου μέρους της άσκησης αυτής είναι να κατασκευαστεί ένα function βασισμένο σε νευρωνικά δίκτυα MLP που να μπορεί να προβλέπει την σχετική απόδοση ενός επεξεργαστή με βάση τα χαρακτηριστικά του συστήματος. Συγκεκριμένα, θα παίρνει σαν εισόδους το σύνολο των διαθέσιμων δεδομένων και θα επιστρέφει το εκπαιδευμένο δίκτυο, καθώς και τους στατιστικούς δείκτες MARE% και R^2 στα σύνολα αξιολόγησης και ελέγχου. Αρχικά, το νευρωνικό δίκτυο θα πρέπει να έχει 2 κρυφές στοιβάδες με 5 νευρώνες στην πρώτη και 3 νευρώνες στη δεύτερη στοιβάδα, και στην συνέχεια ο αριθμός νευρώνων στη πρώτη κρυφή στοιβάδα κυμαίνεται από 5 έως 20 και στη δεύτερη στοιβάδα από 3 έως 10. Από τα νευρωνικά δίκτυα που δημιουργήθηκαν επιλέγετε αυτό που έδωσε τον καλύτερο συντελεστή R^2 στο σύνολο δεδομένων ελέγχου και στη συνέχεια κατασκευάζονται δύο τρισδιάστατες γραφικές παραστάσεις που απεικονίζουν τις προβλέψεις του καλύτερου νευρωνικού δικτύου για την απόδοση ενός δεδομένου επεξεργαστή, συναρτήσει της μέγιστης κεντρικής μνήμης και της μνήμης Cache για δύο διαφορετικά εύρη τιμών της μέγιστης κεντρικής μνήμης και της μνήμης Cache αντίστοιχα.

Σκοπός του δεύτερου μέρους της άσκησης είναι η κατασκευή ενός function, βασισμένο επίσης σε νευρωνικό δίκτυο MLP, το οποίο θα έχει την ικανότητα να αποφασίζει αν ένα email είναι spam ή όχι. Συγκεκριμένα, θα παίρνει σαν εισόδους το σύνολο των διαθέσιμων δεδομένων και σαν έξοδο το εκπαιδευμένο δίκτυο, το συνολικό ποσοστό emails που κατηγοριοποιήθηκαν σωστά σε οποιαδήποτε από τις δύο κατηγορίες, που είναι spam και κατηγοριοποιήθηκαν σωστά ως spam, που δεν είναι spam και κατηγοριοποιήθηκαν σωστά ως μη spam, που είναι spam και κατηγοριοποιήθηκαν λάθος ως μη spam, που δεν είναι spam και κατηγοριοποιήθηκαν λάθος ως spam (ξεχωριστά για τα στα σύνολα αξιολόγησης και ελέγχου). Το νευρωνικό δίκτυο θα εκπαιδευτεί με δύο κρυφές στοιβάδες (20 νευρώνες στην πρώτη κρυφή στοιβάδα και 10 νευρώνες στη δεύτερη). Στην συνέχεια παρουσιάζεται ο πίνακας σύγχυσης (confusion matrix) για τα τρία σύνολα δεδομένων. Η παραπάνω διαδικασία επαναλαμβάνεται κρατώντας σταθερό το διαχωρισμό στα τρία υποσύνολα και αφού εκτελεστεί για έναν αριθμό επαναλήψεων επιλέγεται το μοντέλο που ανταποκρίνεται καλύτερα στην συγκεκριμένη εφαρμογή.

2. Θεωρητικό Υπόβαθρο

✓ Νευρωνικά δίκτυα MLP

Τα νευρωνικά δίκτυα MLP είναι ένα από τα πιο διαδεδομένα μοντέλα νευρωνικών δικτύων και χρησιμοποιείται για προβλήματα αναγνώρισης προτύπων, παλινδρόμησης και ταξινόμησης.

Ένα MLP αποτελείται από έναν ή περισσότερα κρυφά στρώματα νευρώνων, ένα στρώμα εισόδου και ένα στρώμα εξόδου. Οι νευρώνες σε κάθε στρώμα συνδέονται πλήρως με τους νευρώνες του επόμενου στρώματος. Κάθε σύνδεση έχει ένα συντελεστή βάρους που υποδηλώνει τη συνεισφορά του νευρώνα στην εξόδο του επόμενου στρώματος. Οι νευρώνες του MLP χρησιμοποιούν μη γραμμικές συναρτήσεις ενεργοποίησης, όπως η σιγμοειδής συνάρτηση ή η υπερβολική εφαιτομένη.

Η εκπαίδευση ενός MLP γίνεται μέσω αλγορίθμων backpropagation, κατά τους οποίους προσαρμόζονται τα βάρη του δικτύου ώστε να ελαχιστοποιηθεί ένα κριτήριο σφάλματος. Συνήθως χρησιμοποιείται η μέθοδος gradient descent για την ενημέρωση των βαρών.

✓ **Δεδομένα 1^{ης} Άσκησης**

Α) Θα πρέπει να χρησιμοποιηθεί ο παρακάτω χωρισμός δεδομένων σε υποσύνολα:

- **Εκπαίδευση** Δεδομένα 1 μέχρι 90
- **Αξιολόγηση** Δεδομένα 91 μέχρι 150
- **Έλεγχος** Δεδομένα 151 μέχρι 209

Για τον υπολογισμό των MARE% και R^2 χρησιμοποιούνται οι παρακάτω τύποι:

$$MARE\% = 100 \frac{\sum_{j=1}^P \frac{|y_j - \hat{y}_j|}{y_j}}{P}$$

$$R^2 = 1 - \frac{SSE}{SST}, \text{ όπου } SSE = \sum_{j=1}^P (y_j - \hat{y}_j)^2 \text{ και } SST = \sum_{j=1}^P (y_j - \bar{y})^2$$

Γ) Τα δεδομένα που δίνονται για τον υποθετικό επεξεργαστή που θα αξιοποιηθεί είναι:

Χρόνος κύκλου: 200ns
 Ελάχιστη κεντρική μνήμη: 3000KB
 Μέγιστη κεντρική μνήμη: 8000KB-16000KB
 Μνήμη Cache: 32KB-128KB
 Ελάχιστος αριθμός καναλιών: 6
 Μέγιστος αριθμός καναλιών: 16

Δ) Τα δεδομένα που δίνονται για τον υποθετικό επεξεργαστή που θα αξιοποιηθεί είναι:

Χρόνος κύκλου: 200ns
 Ελάχιστη κεντρική μνήμη: 3000KB
 Μέγιστη κεντρική μνήμη: 8000KB-64000KB
 Μνήμη Cache: 32KB-512KB
 Ελάχιστος αριθμός καναλιών: 6
 Μέγιστος αριθμός καναλιών: 16

✓ **Δεδομένα 2^{ης} Άσκησης**

Α) Θα πρέπει να χρησιμοποιηθεί ο παρακάτω χωρισμός δεδομένων σε υποσύνολα:

- Υποσύνολο εκπαίδευσης (40% των δεδομένων)
- Υποσύνολο αξιολόγησης (30% των δεδομένων)
- Υποσύνολο ελέγχου (30% των δεδομένων)

3. Πορεία Εργασίας

Ο κώδικας του function που είναι βασισμένο σε νευρωνικά δίκτυα MLP και προβλέπει την σχετική απόδοση ενός επεξεργαστή με βάση τα χαρακτηριστικά του συστήματος παρουσιάζεται παρακάτω

%Πρόβλεψη απόδοσης CPU

```
function [net, maretr, mareval, maretest, R2tr, R2val, R2test, xstr, ystr]=mlp(data, h1)
```

%Διαχωρισμός των δεδομένων σε 3 σύνολα(Εκπαίδευση, Αξιολόγηση, Έλεγχος)

```
xtr=data(1:90,4:9);
xval=data(91:150,4:9);
xtest=data(150:209,4:9);
```

```
ytr=data(1:90,10);
yval=data(91:150,10);
ytest=data(150:209,10);
```

%Κανονικοποίηση των δεδομένων

```
[xtrsc, xstr]=mapminmax(xtr');
xvalsc=mapminmax('apply', xval', xstr);
xtestsc=mapminmax('apply', xtest', xstr);

[ytrsc, ystr]=mapminmax(ytr');
yvalsc=mapminmax('apply', yval', ystr);
ytestsc=mapminmax('apply', ytest', ystr);
```

%Γεννήτρια τυχαίων αριθμών
rand('seed', 10);

%Δημιουργία δικτύου
net=feedforwardnet(h1);

%Ρύθμιση παραμέτρων δικτύου
net.divideFcn='divideind';
net.divideParam.trainInd=[1:90];
net.divideParam.valInd=[91:150];
net.divideParam.testInd=[151:209];

%Δημιουργία κανονικοποιημένων πινάκων
xsc=[xtrsc xvalsc xtestsc];
ysc=[ytrsc yvalsc ytestsc];

%Εκπαίδευση δικτύου
[net, tr]=train(net, xsc, ysc);

%Προβλέψεις
yscpr=net(xsc);

%Αποκανονικοποίηση προβλέψεων
ypr=mapminmax('reverse', yscpr, ystr);

```

%Εξαγωγή δεικτών
ptr=length(xtr);
pval=length(xval);
ptest=length(xtest);

maretr = 100*(sum(abs(ytr-ypr(1:90)'))./ytr)/ptr;
mareval = 100*(sum(abs(yval-ypr(91:150)'))./yval)/pval;
maretest = 100*(sum(abs(ytest-ypr(151:210)'))./ytest)/ptest;

ybartr = sum(ytr)/ptr;
ybarval = sum(yval)/pval;
ybartest = sum(ytest)/ptest;

SSEtr = sum((ytr - ypr(1:90)').^2);
SSTtr = sum((ytr-ybartr).^2);

R2tr = 1 - (SSEtr/SSTtr);

SSEval = sum((yval - ypr(91:150)').^2);
SSTval = sum((yval-ybarval).^2);










R2val = 1 - (SSEval/SSTval);

SSEtest = sum((ytest - ypr(151:210)').^2);
SSTtest = sum((ytest-ybartest).^2);

R2test = 1 - (SSEtest/SSTtest);

```

Φορτώνεται το αρχείο «Machine_CPU_NN.xlsx» που περιέχει τα δεδομένα στο Matlab και έπειτα εκτελείται το παραπάνω function με είσοδο τα δεδομένα αυτού του αρχείου και hl=[5,3]. Οι πίνακες που δημιουργήθηκαν παρουσιάζονται παρακάτω

	maretest	42.8057
	maretr	36.6024
	mareval	47.7662
	net	1x1 network
	R2test	0.8534
	R2tr	0.9774
	R2val	0.7575
	xstr	1x1 struct
	ystr	1x1 struct

Το παραπάνω function μπαίνει σε έναν βρόχο επανάληψης μεταβάλλοντας το hl έτσι ώστε να ληφθούν όλοι οι πιθανοί συνδυασμοί νευρωνικών δικτύων με τους στατιστικούς τους δείκτες


```

for i=5:20
    for j = 3:10
        hl = [i j];

        [net{i,j},maretr(i,j),mareval(i,j),maretest(i,j),R2tr(i,j),R2val(i,j),R2test(i,j),xs
tr,ystr]=mlp(MachineCPUNN,hl);
    end
end

```

Οι πίνακες που δημιουργήθηκαν παρουσιάζονται παρακάτω

 maretest	20x10 double
 maretr	20x10 double
 mareval	20x10 double
 net	20x10 cell
 R2test	20x10 double
 R2tr	20x10 double
 R2val	20x10 double
 xstr	1x1 struct
 ystr	1x1 struct

Πίνακας 1. Maretest

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	42.8057	36.5234	74.2044	41.9319	49.0650	92.7631	42.1334	52.3815
0	0	37.9144	31.8811	61.4334	49.6719	53.0949	37.2641	103.9465	40.3908
0	0	56.7155	35.9100	34.4798	94.8259	43.9931	78.2923	38.8113	41.5144
0	0	49.1288	56.7000	52.6255	49.0432	123.8271	42.9631	43.6731	32.9394
0	0	47.0273	46.9417	91.0686	77.8744	106.5063	43.3395	102.5291	39.7935
0	0	35.5483	54.0753	91.2472	45.6018	35.0029	36.1189	46.8522	60.3820
0	0	46.5382	37.6899	44.5239	40.6718	46.0761	46.0237	108.9785	45.3212
0	0	183.3568	69.3579	37.8886	38.9750	62.8340	69.0953	53.0627	53.2038
0	0	38.7223	55.9744	50.5241	44.4132	46.2883	64.6675	53.4707	45.5184
0	0	42.5517	31.9498	99.8526	36.5824	40.9074	31.3553	48.7479	80.6322
0	0	62.3658	60.6039	40.8091	82.7717	42.2185	43.2495	276.6628	39.0803
0	0	51.6053	50.6747	42.4699	37.5688	42.6827	95.1062	77.9539	116.4342
0	0	47.3428	57.4349	49.6049	42.3115	48.9131	39.8835	75.1909	216.5909
0	0	134.1735	46.4277	38.7168	128.3806	37.1287	46.0687	51.7115	39.8842
0	0	41.4645	59.5121	53.4252	41.0426	62.1842	42.5523	60.9461	59.3810
0	0	66.9307	46.2892	39.7754	39.4240	105.1591	43.7935	55.4048	49.3286

Πίνακας 2. Maretr

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	36.6024	34.2285	80.7419	28.7058	49.4994	109.2107	43.2976	48.1940
0	0	31.1108	30.0140	59.0478	51.4256	62.8497	30.1042	121.3079	31.5287
0	0	60.7352	40.0920	29.3415	108.4876	27.7121	77.2024	35.3815	28.1376
0	0	41.2003	58.0493	61.2829	29.9956	157.6430	31.5945	30.7463	25.8490
0	0	35.1319	41.0453	99.0885	82.1728	130.0505	34.4237	121.9541	29.1935
0	0	33.7984	52.0365	101.6369	24.2246	32.8053	26.9831	38.5213	19.7958
0	0	46.5397	28.2431	35.3180	30.9732	24.4756	24.1943	130.5783	17.0827
0	0	242.8265	80.4780	30.7814	29.4911	68.4318	67.8648	23.0957	42.7948
0	0	29.0349	45.4432	19.7792	36.2137	26.9823	55.5689	26.5727	41.4205
0	0	52.6174	32.2841	119.8536	26.4215	24.8549	28.2400	39.1178	100.8164
0	0	49.0229	69.9590	38.5766	89.3878	44.9888	28.0858	387.6038	24.8150
0	0	35.3500	47.0418	25.9470	26.6725	36.4830	95.8745	77.4359	122.7053
0	0	36.6015	60.0380	54.9287	37.9018	48.9869	28.4414	73.1146	250.9713
0	0	175.8510	26.9242	24.5075	161.6361	23.6884	27.8677	35.5138	36.1028
0	0	38.9815	54.1585	40.8547	27.3257	56.5154	28.2311	54.8456	57.2892
0	0	72.8052	40.2141	30.1536	29.1321	106.4663	28.3456	63.4614	26.4621

Πίνακας 3. Mareval

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	47.7662	41.4331	78.7308	44.7739	77.2043	124.9842	54.3287	48.5301
0	0	37.1139	34.0052	74.1843	61.2409	73.2727	37.7092	127.3545	39.4954
0	0	64.1478	43.8244	39.3296	121.4525	45.6911	86.1704	46.8380	40.2882
0	0	47.8860	65.3791	69.8619	50.0465	151.8149	41.2086	42.9951	41.0395
0	0	47.8789	55.4087	91.4514	70.6784	151.6121	44.9270	175.2683	39.2046
0	0	43.7470	59.9803	97.4223	41.6326	36.1777	36.5716	45.5603	61.0307
0	0	53.2333	37.5063	43.9439	34.9298	46.9521	49.6335	141.6675	51.4677
0	0	251.9243	91.6008	39.6452	43.8257	81.1283	87.4780	49.5439	57.3933
0	0	45.9629	53.9056	53.6713	41.5168	41.7975	62.6673	40.6505	62.8470
0	0	53.1210	35.5051	115.9802	41.4971	43.0146	34.1899	45.3232	143.1481
0	0	57.2638	90.2736	47.1620	108.6339	62.2557	45.6248	506.6602	46.4037
0	0	48.9887	47.8942	39.2900	52.0370	40.2195	89.9960	85.8866	126.8314
0	0	49.1211	67.1187	70.0110	50.5023	52.5040	45.3248	80.0792	230.1318
0	0	207.2254	44.5149	39.6031	175.5001	36.1915	38.4271	41.4898	40.6207
0	0	48.2204	67.7115	51.8545	35.1611	72.5249	44.4171	52.3098	68.4980
0	0	70.2275	45.6760	49.0982	44.4195	109.7166	46.2330	74.8459	38.8363

Πίνακας 4. $R2_{test}$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0.8534	0.9174	0.8782	0.8590	0.9019	0.8954	0.9369	0.8449
0	0	0.9056	0.9281	0.8300	0.8049	0.9150	0.9219	0.8713	0.8658
0	0	0.9060	0.9153	0.9296	0.5390	0.6701	0.8828	0.8522	0.0359
0	0	0.7091	0.9246	0.8759	0.6705	0.8567	0.7934	0.9108	0.8998
0	0	0.8707	0.9186	0.7624	0.6688	0.7673	0.8800	0.8139	0.8738
0	0	0.9126	0.8412	0.7398	0.5521	0.8822	0.9295	0.8559	0.5397
0	0	0.9386	0.8962	0.8932	0.8575	0.8747	0.6322	0.7894	0.6689
0	0	0.2752	0.8313	0.9237	0.8496	0.7282	0.8446	0.8418	0.7922
0	0	0.8694	0.9165	0.5530	0.9098	0.8248	0.7194	0.8397	0.9164
0	0	0.8908	0.8794	0.6779	0.9223	0.6273	0.9308	0.9087	0.8991
0	0	0.8725	0.7713	0.9271	0.6766	0.9072	0.9045	0.3060	0.8747
0	0	0.6581	0.9193	0.7857	0.7243	0.8369	0.8215	0.8024	0.7925
0	0	0.8973	0.9138	0.8725	0.9175	0.9422	0.9499	0.9188	0.6203
0	0	0.4527	0.4588	0.9124	0.8395	0.7352	0.8119	0.8971	0.8974
0	0	0.9235	0.8367	0.8775	0.9223	0.6126	0.8620	0.6718	0.8608
0	0	0.8667	0.8661	0.9167	0.8493	0.7083	0.6095	0.9128	0.5271

Πίνακας 5. $R2_{tr}$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0.9774	0.9766	0.9370	0.9828	0.9586	0.9229	0.9759	0.8988
0	0	0.9773	0.9521	0.9302	0.9647	0.9392	0.9751	0.8817	0.9778
0	0	0.9628	0.9529	0.9786	0.8659	0.9870	0.9363	0.9805	0.9774
0	0	0.9772	0.9691	0.9586	0.9810	0.8592	0.9738	0.9752	0.9819
0	0	0.9735	0.9682	0.8307	0.7739	0.7941	0.9856	0.8532	0.9835
0	0	0.9794	0.9617	0.9303	0.9915	0.9779	0.9832	0.9809	0.9948
0	0	0.9753	0.9815	0.9784	0.9755	0.9884	0.9911	0.8833	0.9958
0	0	0.2570	0.9550	0.9854	0.9825	0.7420	0.9619	0.9894	0.9280
0	0	0.9881	0.9711	0.9948	0.9757	0.9876	0.9565	0.9888	0.9720
0	0	0.9297	0.9550	0.7910	0.9775	0.9907	0.9807	0.9752	0.8988
0	0	0.9684	0.9379	0.9770	0.8759	0.9614	0.9779	0.0297	0.9898
0	0	0.9836	0.9748	0.9863	0.9866	0.9759	0.8907	0.8474	0.8656
0	0	0.9727	0.9586	0.9626	0.9764	0.9676	0.9702	0.9590	0.5295
0	0	0.3628	0.9879	0.9802	0.8400	0.9898	0.9785	0.9744	0.9797
0	0	0.9674	0.8978	0.9713	0.9767	0.9696	0.9758	0.6723	0.8207
0	0	0.9372	0.9625	0.9836	0.9837	0.9103	0.9891	0.9702	0.9844

Πίνακας 6. $R2val$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0.7575	0.8947	0.7034	0.8316	0.6283	0.5902	0.8647	0.7257
0	0	0.8777	0.8376	0.6501	0.0491	0.7237	0.8585	0.8474	0.6728
0	0	0.7968	0.8530	0.8860	0.7480	0.7673	0.6761	0.7707	0.3019
0	0	0.8750	0.7993	0.8476	0.8683	0.8475	0.8841	0.7727	0.8230
0	0	0.7531	0.5391	0.7851	0.7911	0.6962	0.8959	0.6320	-0.0387
0	0	0.6969	0.7421	0.5111	0.8683	0.8453	0.8092	0.9121	0.6802
0	0	0.8077	0.8013	0.8621	0.9097	0.8336	0.8293	0.5970	0.7671
0	0	0.4420	0.8242	0.9084	0.5159	0.3705	0.7524	0.5967	0.7277
0	0	0.8796	0.8328	0.6018	0.9102	0.7103	0.8029	0.8833	0.4983
0	0	0.7441	0.8115	0.4909	0.8902	0.7984	0.8820	0.8149	0.8161
0	0	0.7079	0.8855	0.8502	0.8128	0.6616	0.7982	0.3270	0.7846
0	0	0.8549	0.8849	0.8541	0.5173	0.8967	0.7371	0.7615	0.4281
0	0	0.7304	0.8372	0.7981	0.7562	0.8053	0.6308	0.7695	0.2914
0	0	0.7344	0.8277	0.6774	0.7976	0.8993	0.8207	0.7923	0.6306
0	0	0.6469	0.6261	0.5907	0.6951	0.7208	0.7554	0.7656	-0.0864
0	0	0.5697	0.7562	0.7739	0.7941	0.3034	0.7101	0.7921	0.7805

Για να βρεθεί το νευρωνικό δίκτυο που έδωσε τον καλύτερο συντελεστή R^2 στο σύνολο δεδομένων ελέγχου, εκτελείται ο παρακάτω κώδικας

```
r=0;
```

```
for i=5:20
```

```
    for j = 3:10
```

```
        if (r<R2test(i,j))
```

```
            r=R2test(i,j);
```

```
            k1=i;
```

```
            k2=j;
```

```
        end
```

```
    end
```

```
end
```

Το νευρωνικό δίκτυο είναι αυτό με συντεταγμένες (17,8) και συντελεστή $R^2 = 0,9499$

Κατασκευάζεται τρισδιάστατη γραφική παράσταση που απεικονίζει τις προβλέψεις αυτού του νευρωνικού δικτύου για την απόδοση ενός υποτιθέμενου επεξεργαστή, του οποίου οι προδιαγραφές αναφέρονται στο θεωρητικό υπόβαθρο, συναρτήσει της μέγιστης κεντρικής μνήμης και της μνήμης Cache. Ο κώδικας υλοποίησης όπως και η γραφική παρουσίαζονται παρακάτω

```
x3 = 8000:8001/97:16000;
```

```
x4 = 32:1:128;
```

```
[XX,YY] = meshgrid(x3,x4);
```

```
for i=1:97
```

```
    for j=1:97
```

```
        X = [200 3000 x3(i) x4(j) 6 16]';
```

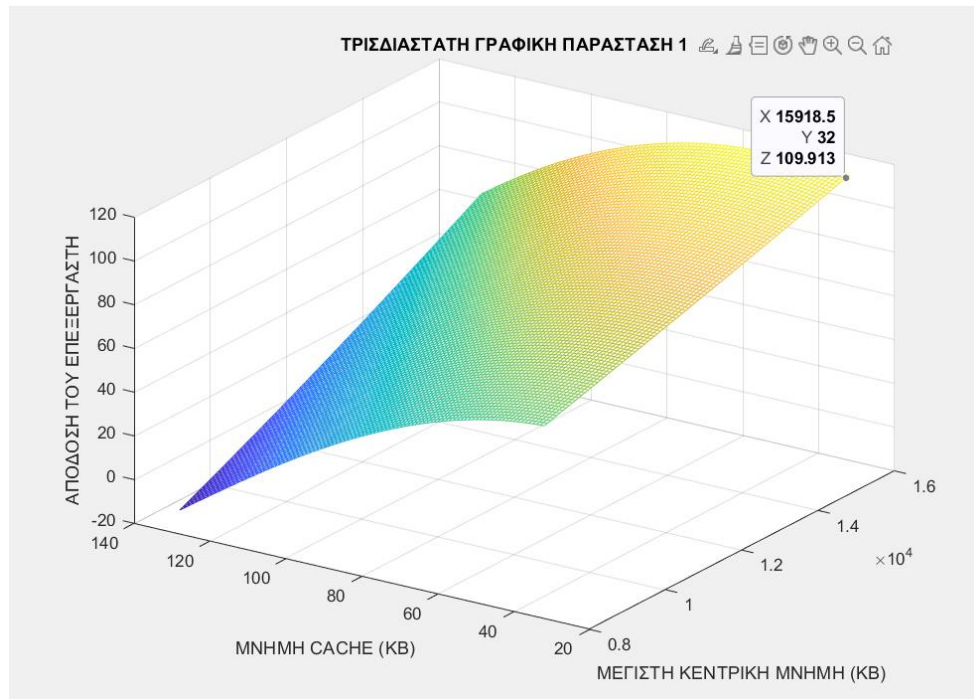
```
        [xsc] = mapminmax('apply', X, xstr);
```

```
        zsc(i,j) = net{k1,k2}(xsc);
```

```

end
end
z = mapminmax('reverse',zsc, ystr);
figure();
mesh(XX, YY, z')
title("ΤΡΙΣΔΙΑΣΤΑΤΗ ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ 1");
xlabel("ΜΕΓΙΣΤΗ ΚΕΝΤΡΙΚΗ ΜΝΗΜΗ (KB)");
ylabel("ΜΝΗΜΗ CACHE (KB)");
zlabel("ΑΠΟΔΟΣΗ ΤΟΥ ΕΠΕΞΕΡΓΑΣΤΗ");

```



Εικόνα 1. Τρισδιάστατη γραφική παράσταση της απόδοσης του επεξεργαστή, της μέγιστης κεντρικής μνήμης και της μνήμης Cache

Μεταβάλλονται τα εύρη τιμών της μέγιστης κεντρικής μνήμης και της μνήμης Cache, όπως παρουσιάζονται στο θεωρητικό υπόβαθρο, και κατασκευάζεται εκ νέου η τρισδιάστατη γραφική παράσταση. Ο κώδικας υλοποίησης όπως και η γραφική παράσταση παρουσιάζονται παρακάτω

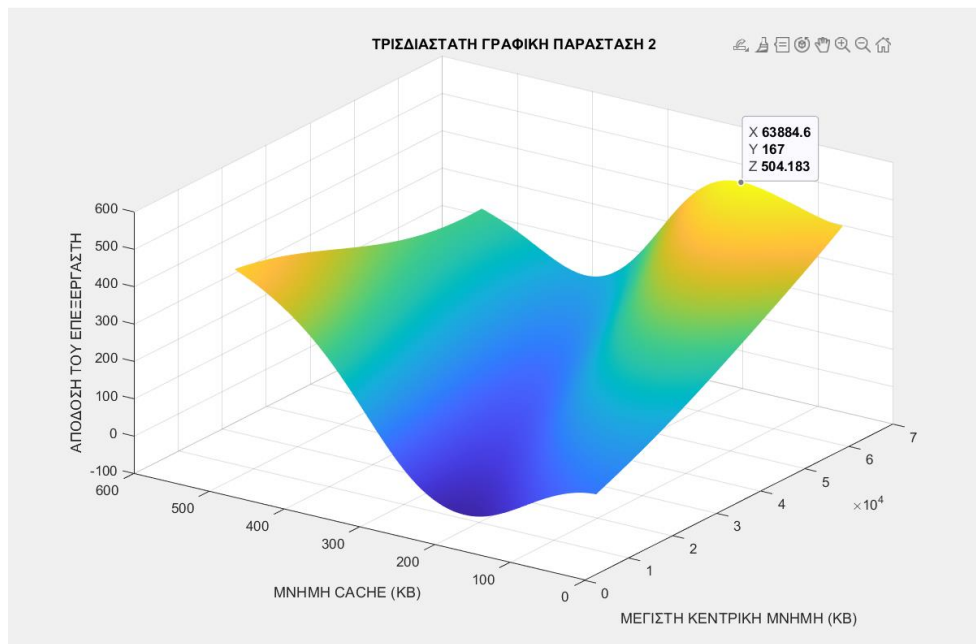
```

x3 = 8000:56001/481:64000;
x4 = 32:1:512;
[XX,YY] = meshgrid(x3,x4);

for i=1:481
    for j=1:481
        X = [200 3000 x3(i) x4(j) 6 16]';
        [xsc] = mapminmax('apply', X, xstr);
        zsc(i,j) = net{k1,k2}(xsc);
    end
end
z = mapminmax('reverse',zsc, ystr);
figure();
mesh(XX, YY, z')
title("ΤΡΙΣΔΙΑΣΤΑΤΗ ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ 2");
xlabel("ΜΕΓΙΣΤΗ ΚΕΝΤΡΙΚΗ ΜΝΗΜΗ (KB)");
ylabel("ΜΝΗΜΗ CACHE (KB)");

```

```
zlabel("ΑΠΟΔΟΣΗ ΤΟΥ ΕΠΕΞΕΡΓΑΣΤΗ");
```



Εικόνα 2. Τρισδιάστατη γραφική παράσταση της απόδοσης του επεξεργαστή, της μέγιστης κεντρικής μνήμης και της μνήμης Cache

Ο κώδικας του function, βασισμένο επίσης σε νευρωνικό δίκτυο MLP, το οποίο θα έχει την ικανότητα να αποφασίζει αν ένα email είναι spam ή όχι παρουσιάζεται παρακάτω

```
%Αυτόματος ανιχνευτής spam emails
```

```
function
```

```
[net,ecval,ectest,spamcval,spamctest,nspamcval,nspamctest,spamfval,spamftest,nspamfval,nspamftest]=spam(data)
```

```
%Αναδιάταξη
```

```
p=length(data);
```

```
r=randperm(p);
```

```
dat=data(r,:);
```

```
%Διαχωρισμός των δεδομένων σε 3 σύνολα(Εκπαίδευση, Αξιολόγηση, Έλεγχος)
```

```
xtr=dat(1:round(p*0.4),1:end-1);
```

```
xval=dat(round(p*0.4)+1:round(p*0.7),1:end-1);
```

```
xtest=dat(round(p*0.7)+1:p,1:end-1);
```

```
ytr=dat(1:round(p*0.4),58);
```

```
yval=dat(round(p*0.4)+1:round(p*0.7),58);
```

```
ytest=dat(round(p*0.7)+1:p,58);
```

```
%Κανονικοποίηση των X
```

```
[xtrsc,xstr]=mapminmax(xtr');
```

```
xvalsc=mapminmax('apply',xval',xstr);
```

```
xtestsc=mapminmax('apply',xtest',xstr);
```

```
h1 = [20 10];
```

```

%Δημιουργία δικτύου
net=patternnet(h1);

%Ρύθμιση παραμέτρων δικτύου
net.divideFcn='divideind';
net.divideParam.trainInd=[1:round(p*0.4)];
net.divideParam.valInd=[round(p*0.4)+1:round(p*0.7)];
net.divideParam.testInd=[round(p*0.7)+1:p];

xsc=[xtrsc xvalsc xtestsc];
y=[ytr;yval;ytest];

y2=[~y y];

%Εκπαίδευση δικτύου
[net,tr]=train(net,xsc,y2');

%Προβλέψεις
yp2=net(xsc);
yp2=yp2';

yp=yp2(:,1)<yp2(:,2);

%Εξαγωγή ποσοστών
yptr=yp(1:length(ytr));
ypval=yp(length(ytr)+1:length(yval)+length(ytr));
yptest=yp(length(yval)+length(ytr)+1:length(y));

ecval=100*sum(yval==ypval)/length(yval);
ectest=100*sum(ytest==ypval)/length(ytest);
spamcval=100*sum((yval==1)&(ypval==1))/sum(yval==1);
spamctest=100*sum((ytest==1)&(ypval==1))/sum(ytest==1);
nspamcval=100*sum((yval==0)&(ypval==0))/sum(yval==0);
nspamctest=100*sum((ytest==0)&(ypval==0))/sum(ytest==0);
spamfval=100*sum((yval==1)&(ypval==0))/sum(yval==1);
spamftest=100*sum((ytest==1)&(ypval==0))/sum(ytest==1);
nspamfval=100*sum((yval==0)&(ypval==1))/sum(yval==0);
nspamftest=100*sum((ytest==0)&(ypval==1))/sum(ytest==0);

%Δημιουργία πινάκων σύγχυσης
figure;
plotconfusion(ytr',yptr');
title('Confusion Matrix tr');
xlabel('ytr');
ylabel('yptr');

figure;
plotconfusion(yval',ypval');
title('Confusion Matrix val');
xlabel('yval');
ylabel('ypval');

figure;
plotconfusion(ytest',ypval');
title('Confusion Matrix test');
xlabel('ytest');
ylabel('ypval');

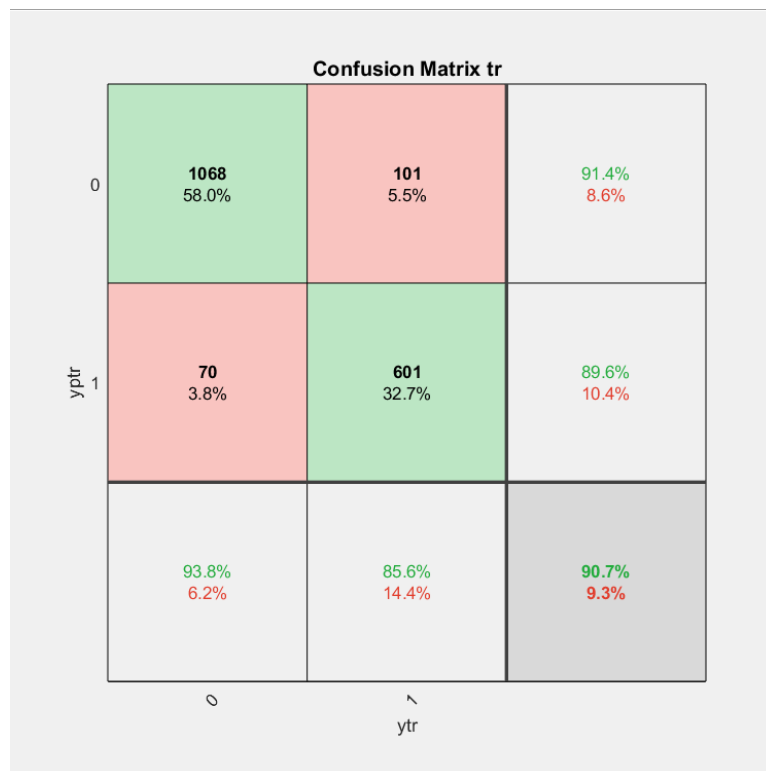
```

```
figure;
plotconfusion(y',yp');
title('Confusion Matrix all');
xlabel('y');
ylabel('yp');
```

Φορτώνεται το αρχείο «spamdata.mat» που περιέχει τα δεδομένα στο Matlab και έπειτα εκτελείται το παραπάνω function με είσοδο τα δεδομένα αυτού του αρχείου. Τα αποτελέσματα παρουσιάζονται παρακάτω

```
ecval = 90.7314
ectest = 90.5797
spamcval = 84.9732
spamctest = 84.2391
nspamcval = 94.6472
nspamctest = 94.8068
spamfval = 15.0268
spamfctest = 15.7609
nspamfval = 5.3528
nspamfctest = 5.1932
```

Πίνακας 7. Πίνακας σύγχυσης για τα δεδομένα εκπαίδευσης



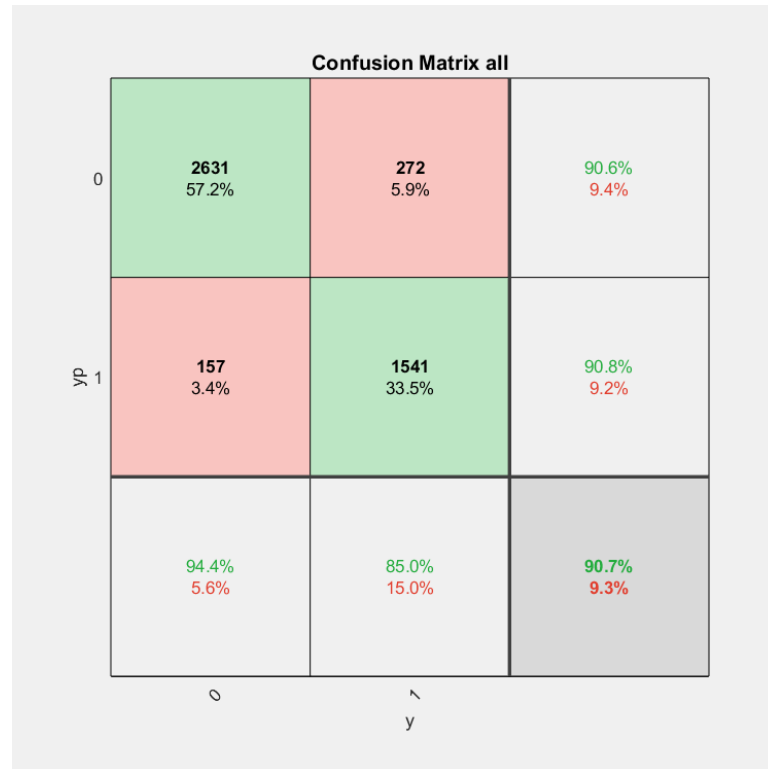
Πίνακας 8. Πίνακας σύγκρισης για τα δεδομένα αξιολόγησης

Confusion Matrix val			
ypval	0	1	
	778 56.3%	84 6.1%	90.3% 9.7%
1	44 3.2%	475 34.4%	91.5% 8.5%
	94.6% 5.4%	85.0% 15.0%	90.7% 9.3%
	0	1	yval

Πίνακας 9. Πίνακας σύγκρισης για τα δεδομένα ελέγχου

Confusion Matrix test			
yp_test	0	1	
	0	1	ytest
0	785 56.9%	87 6.3%	90.0% 10.0%
1	43 3.1%	465 33.7%	91.5% 8.5%
	94.8% 5.2%	84.2% 15.8%	90.6% 9.4%

Πίνακας 10. Πίνακας σύγχυσης για όλα τα δεδομένα



Εκτελείται το παραπάνω function 9 φορές μέσω του παρακάτω κώδικα

```
for i=1:3
    for j = 1:3
        [net{i,j},ecval(i,j),ectest(i,j),spamcval(i,j),spamctest(i,j),nspamcval(i,j),nspamctest(i,j),spamfval(i,j),spamfctest(i,j),nspamfval(i,j),nspamfctest(i,j)]=spam(dat)
    end
end
```

Τα αποτελέσματα παρουσιάζονται παρακάτω

ecval =

```
89.9348  91.1658  91.0210
92.4692  92.1072  88.7038
90.3693  91.2382  90.2245
```

ectest =

```
91.5217  91.3768  91.0870
92.6087  90.8696  89.0580
92.4638  91.8841  87.4638
```

spamcval =

85.3097	86.2007	87.2180
89.1386	86.5942	86.3309
84.0237	87.1401	84.5574

spamctest =

85.9023	85.4281	86.5314
88.8683	86.9643	84.9558
87.9371	88.4135	80.2536

nspamcval =

93.1373	94.5322	93.4040
94.5691	95.7780	90.3030
94.0503	93.7209	93.7647

nspamctest =

95.0472	95.3069	94.0334
95.0059	93.5366	91.9018
95.6683	94.2613	92.2705

spamfval =

14.6903	13.7993	12.7820
10.8614	13.4058	13.6691
15.9763	12.8599	15.4426

spamftest =

14.0977	14.5719	13.4686
11.1317	13.0357	15.0442
12.0629	11.5865	19.7464

nspamfval =

6.8627	5.4678	6.5960
5.4309	4.2220	9.6970
5.9497	6.2791	6.2353

nspsamftest =

4.9528	4.6931	5.9666
4.9941	6.4634	8.0982
4.3317	5.7387	7.7295

4. Συμπεράσματα

Στην 1^η Άσκηση μπορεί κανείς να δει ότι υπάρχουν αρκετά μεγάλες αποκλείσεις στις τιμές των στατιστικών δεικτών, αυτό οφείλεται στο φαινόμενο του overfitting. Το overfitting είναι μια κατάσταση που συμβαίνει όταν ένα μοντέλο μάθησης μάθει να προσαρμόζεται πάρα πολύ στα δεδομένα εκπαίδευσης και χάνει την ικανότητά του να γενικεύει σε νέα δεδομένα. Στην 1^η 3D γραφική παράσταση παρατηρείται ότι όσο αυξάνεται η μέγιστη κεντρική μνήμη η απόδοση του επεξεργαστή αυξάνεται, αλλά όσο αυξάνεται η μνήμη Cache η απόδοση του επεξεργαστή μειώνεται. Στην αντίστοιχη γραφική παράσταση της δεύτερης εργαστηριακής άσκησης είχε παρατηρηθεί πως όσο αυξάνεται η μέγιστη κεντρική μνήμη η απόδοση του επεξεργαστή αυξάνεται όπως και όσο αυξάνεται η μνήμη Cache η απόδοση του επεξεργαστή αυξάνεται. Στην 2^η 3D γραφική παράσταση παρατηρείται πως η μέγιστη κεντρική μνήμη και η μνήμη Cache επιδρούν με τον ίδιο τρόπο στην απόδοση του επεξεργαστή όπως και στην 1^η γραφική, ωστόσο η απόδοση του επεξεργαστή παρουσιάζει μια περίεργη αυξομείωση στις τιμές της, αυτό οφείλεται στο φαινόμενο του extrapolation. Το extrapolation είναι η διαδικασία πρόβλεψης ή εκτίμησης των τιμών μιας μεταβλητής σε περιοχές εκτός του εύρους των διαθέσιμων δεδομένων. Στην 2^η Άσκηση το μοντέλο που επιλέγεται ως το καλύτερο για την συγκεκριμένη εφαρμογή είναι αυτό στη θέση (2,1), καθώς αυτό παρουσιάζει το μεγαλύτερο ποσοστό ectest.

5. Βιβλιογραφία

- ✓ Σημειώσεις του μαθήματος