**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ**

**ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

**ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ**

**(ΕΡΓΑΣΤΗΡΙΟ)**

**ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2022-2023**

**4η Σειρά ασκήσεων**

**Ονομ/νυμα και αρ.μητρώου**

**Αθανασίου Ελένη 19387004**

**Βαβαΐτη Κωνσταντίνα 18387257**

**Εργαστηριακή ομάδα**

**Ομάδα Γ**

**Αιγάλεω 23/05/2023**

# 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 R2 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 R2 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 οχι. 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.

## 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 εφαπτομένη.

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 1ης 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 R2 χρησιμοποιούνται οι παρακάτω τύποι:

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 2ης 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)

## 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,hl)

%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(hl);

%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);

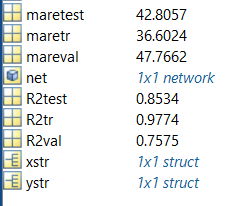
R2val = 1 - (SSEval/SSTval);

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



Το παραπάνω function μπαίνει σε έναν βρόχο επανάληψης μεταβάλοντας 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

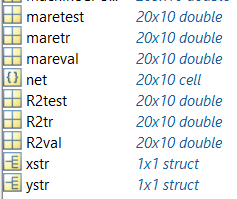
hl = [i j];

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

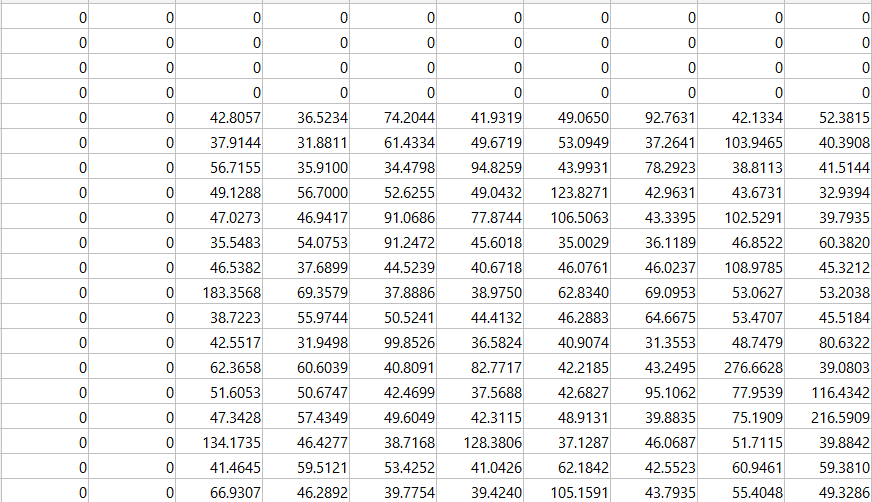
end

end

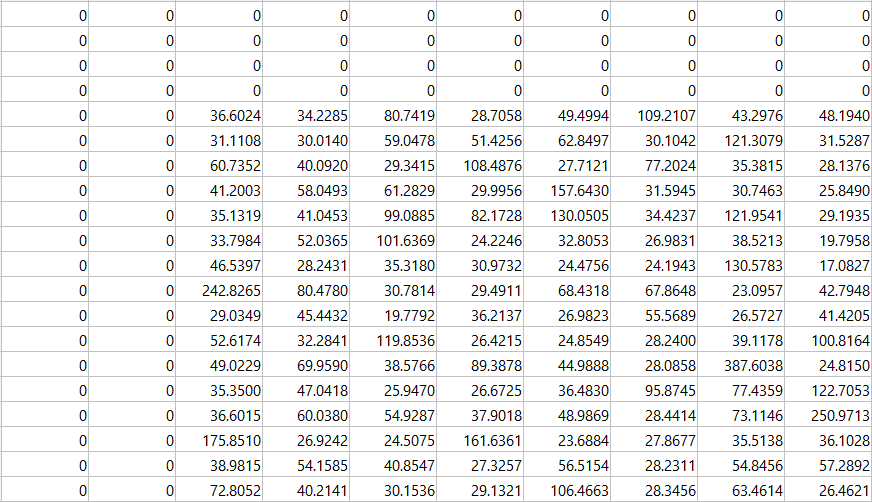
Οι πίνακες που δημιουργήθηκαν παρουσιάζονται παρακάτω



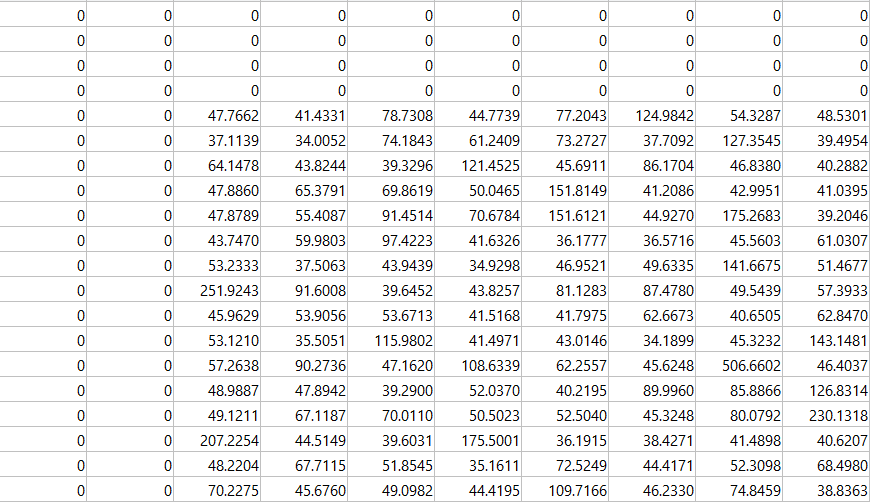
Πίνακας 1. Maretest



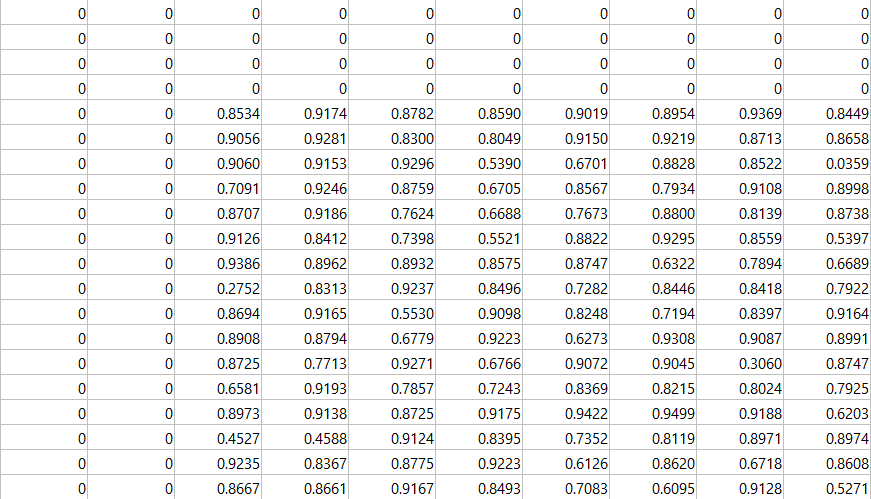
Πίνακας 2. Maretr



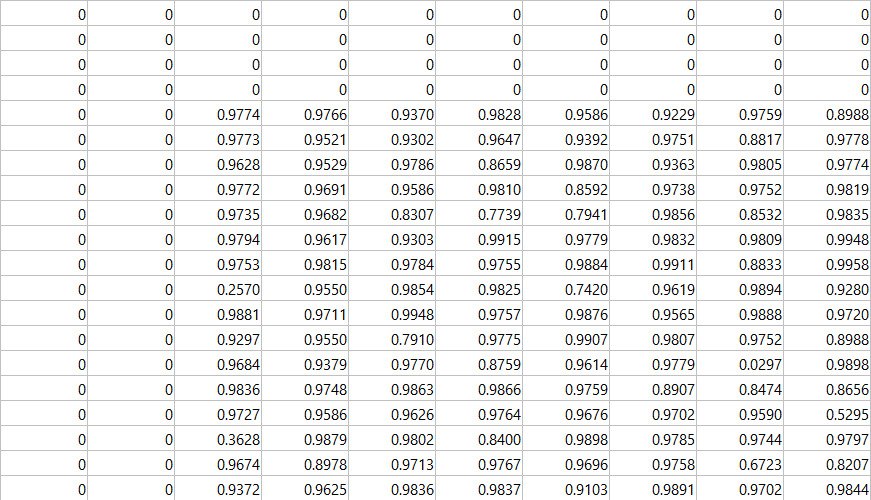
Πίνακας 3. Mareval



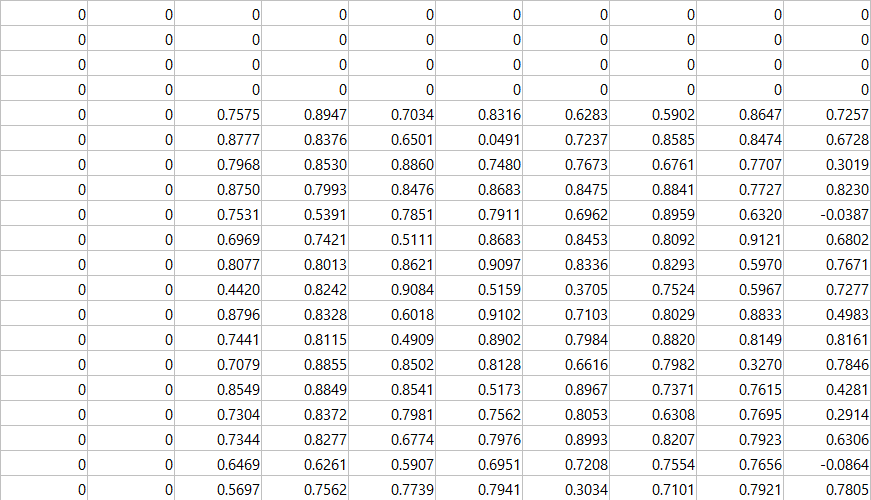
Πίνακας 4. R2test



Πίνακας 5. R2tr



Πίνακας 6. R2val



Για να βρεθεί το νευρωνικό δίκτυο που έδωσε τον καλύτερο συντελεστή R2 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 R2 =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 παρουσιάζονται 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();

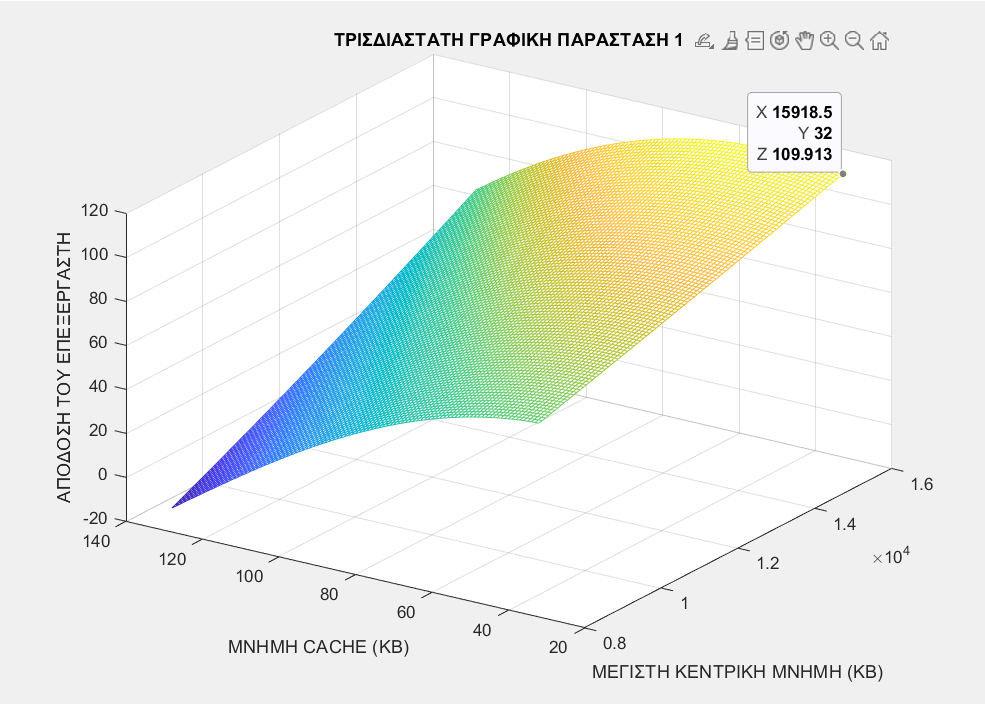
mesh(XX, YY, z')

title("ΤΡΙΣΔΙΑΣΤΑΤΗ ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ 1");

xlabel("ΜΕΓΙΣΤΗ ΚΕΝΤΡΙΚΗ ΜΝΗΜΗ (KB)");

ylabel("ΜΝΗΜΗ CACHE (KB)");

zlabel("ΑΠΟΔΟΣΗ ΤΟΥ ΕΠΕΞΕΡΓΑΣΤΗ");



Εικόνα 1.Τριστιάστατη γραφική παράσταση της απόδοσης του επεξεργαστη, 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();

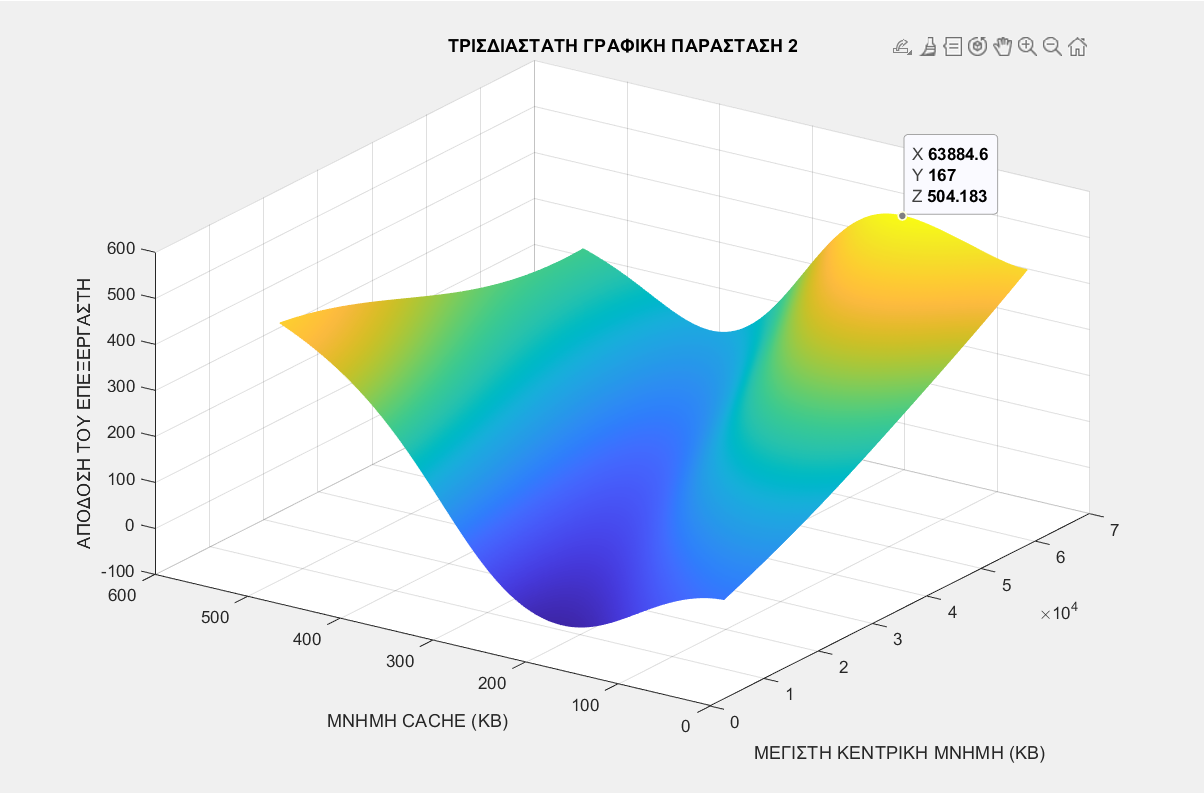
mesh(XX, YY, z')

title("ΤΡΙΣΔΙΑΣΤΑΤΗ ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ 2");

xlabel("ΜΕΓΙΣΤΗ ΚΕΝΤΡΙΚΗ ΜΝΗΜΗ (KB)");

ylabel("ΜΝΗΜΗ CACHE (KB)");

zlabel("ΑΠΟΔΟΣΗ ΤΟΥ ΕΠΕΞΕΡΓΑΣΤΗ");



Εικόνα 2.Τριστιάστατη γραφική παράσταση της απόδοσης του επεξεργαστη, 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);

%Κανονικοποιήση των X

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

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

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

hl = [20 10];

%Δημηουργία δικτύου

net=patternnet(hl);

%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==yptest)/length(ytest);

spamcval=100\*sum((yval==1)&(ypval==1))/sum(yval==1);

spamctest=100\*sum((ytest==1)&(yptest==1))/sum(ytest==1);

nspamcval=100\*sum((yval==0)&(ypval==0))/sum(yval==0);

nspamctest=100\*sum((ytest==0)&(yptest==0))/sum(ytest==0);

spamfval=100\*sum((yval==1)&(ypval==0))/sum(yval==1);

spamftest=100\*sum((ytest==1)&(yptest==0))/sum(ytest==1);

nspamfval=100\*sum((yval==0)&(ypval==1))/sum(yval==0);

nspamftest=100\*sum((ytest==0)&(yptest==1))/sum(ytest==0);

%Δημηουργία 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',yptest');

title('Confusion Matrix test');

xlabel('ytest');

ylabel('yptest');

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

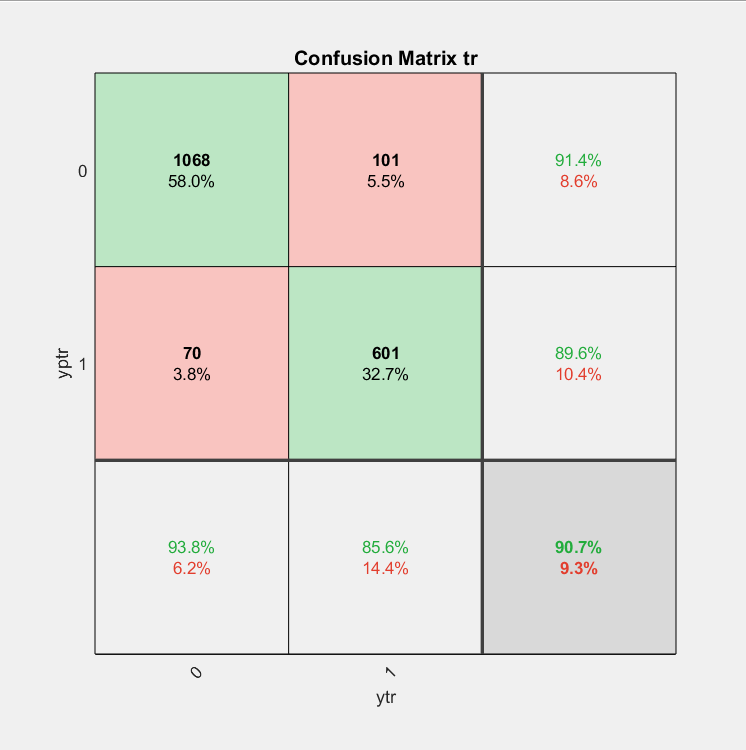
spamfval = 15.0268

spamftest = 15.7609

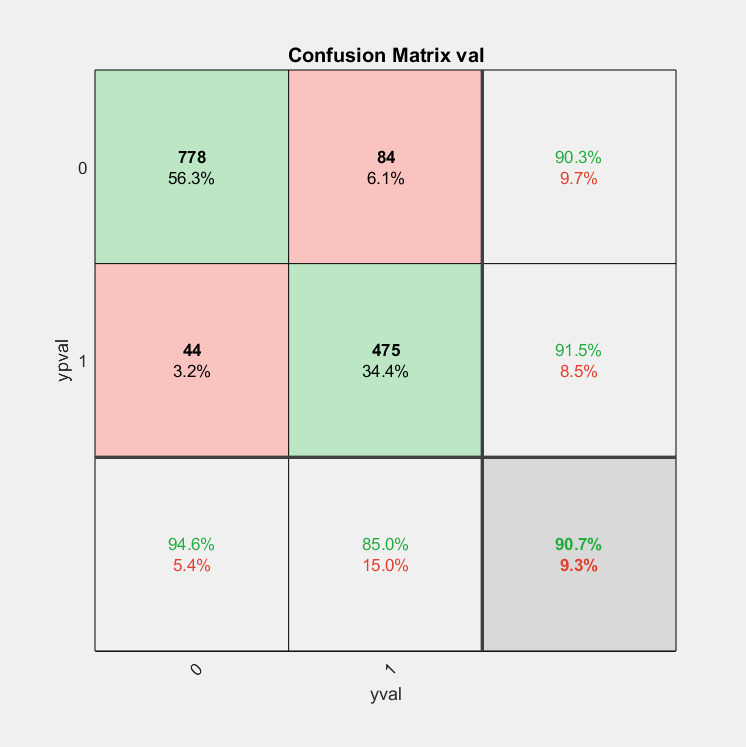
nspamfval = 5.3528

nspamftest = 5.1932

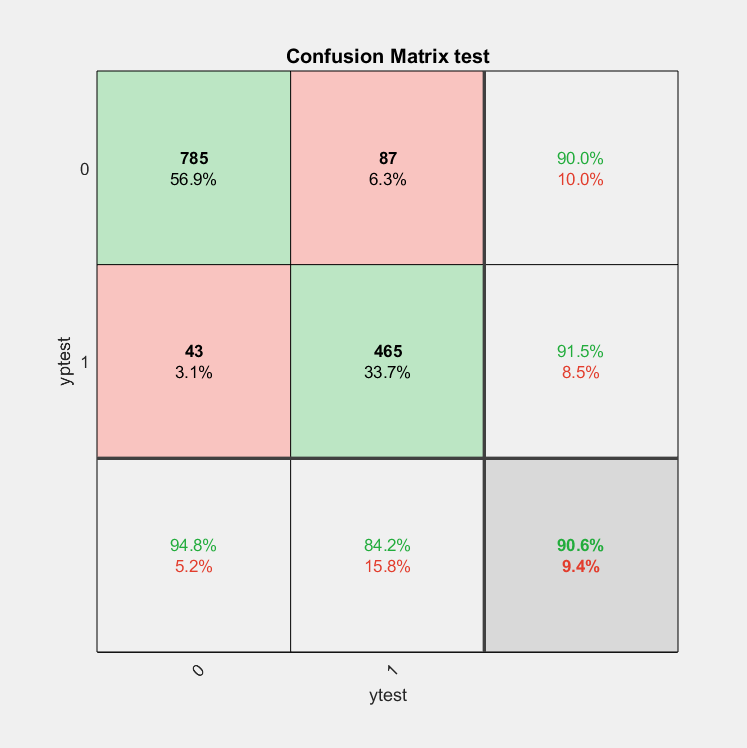
Πίνακας 7. Πίνακας σύγχυσης για τα δεδομένα εκπαίδευσης



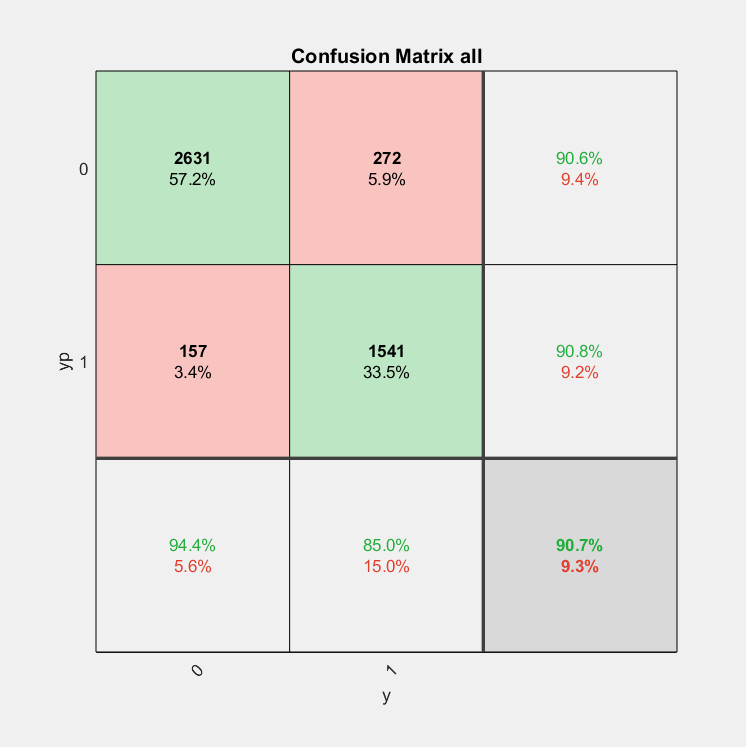
Πίνακας 8.Πίνακας σύγχυσης για τα δεδομένα αξιολόγησης



Πίνακας 9.Πίνακας σύγχυσης για τα δεδομένα ελέγχου



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),spamftest(i,j),nspamfval(i,j),nspamftest(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

nspamftest =

4.9528 4.6931 5.9666

4.9941 6.4634 8.0982

4.3317 5.7387 7.7295

## Συμπεράσματα

Στην 1η 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 1η 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 2η 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 1η 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 2η 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.

## Βιβλιογραφία

* *Σημειώσεις του μαθήματος*