

Παράδειγμα Κώδικα για Ταξινόμηση Προτύπων

Το παράδειγμα κώδικα που δίνεται στη συνέχεια είναι ένα πρόγραμμα σε MATLAB (script) το οποίο χρησιμεύει για την εκπαίδευση ενός δικτύου Multi-Layer Perceptron σε ένα πρόβλημα ταξινόμησης. Σε σχόλια δίνονται όλες οι εξηγήσεις καθώς και όσες συναρτήσεις θα χρειαστεί να γράψετε για να ολοκληρωθεί το εν λόγω πρόγραμμα.

```
clear;
clc;
InitParameters;
% Στη συνάρτηση αυτή ορίζονται όλες οι παράμετροι του δικτύου,
% όπως πλήθος εισόδων, πλήθος εξόδων, πλήθος επιπέδων του δικτύου,
% νευρώνων σε κάθε επίπεδο κλπ.
% Για παράδειγμα: inputnodes, outputnodes, hiddenLlnodes, κλπ.
,

MakeData;
% Στη συνάρτηση αυτή προετοιμάζονται τα δεδομένα για χρήση από το
% δίκτυο. Από τα δεδομένα διαγράφονται πρότυπα στα οποία λείπουν τιμές
% κάποιων χαρακτηριστικών (attributs), κανονικοποιούνται,
% ανακατεύονται (shuffling) και χωρίζονται σε σύνολα εκπαίδευσης και
% ελέγχου.

% Παράδειγμα εντολής φόρτωσης των δεδομένων προβλήματος από αρχείο που
% προέκυψε από τη συνάρτηση MakeData
load('dataset', 'train_set', 'test_set');

% Οι επόμενες εντολές ενημερώνουν μεταβλητές για τοπική χρήση στο
% πρόγραμμα με τα δεδομένα εκπαίδευσης και τα δεδομένα ελέγχου καθώς
% και με τους αντίστοιχους στόχους
p = train_set(:, 1:inputnodes)';
t = train_set(:, inputnodes+1:end)';
tp = test_set(:, 1:inputnodes)';
tt = test_set(:, inputnodes+1:end)';

% Στην επόμενη εντολή δίνεται ένα παράδειγμα ορισμού ενός δικτύου MLP
% με ένα κρυμμένο επίπεδο και πλήθος κρυμμένων νευρώνων hiddenLlnodes.
% Η συνάρτηση ενεργοποίησης των νευρώνων του κρυμμένου επιπέδου είναι
% η υπερβολική εφαπτομένη δηλ. tansig στο MATLAB και
% η συνάρτηση ενεργοποίησης των νευρώνων του επιπέδου εξόδου είναι η
% λογιστική σιγμοειδής δηλ. logsig στο MATLAB
% Ο αλγόριθμος gradient descent που χρησιμοποιείται για batch-training
% στο back-propagation είναι ο αλγόριθμος Levenberg-Marquard, trainlm
% στο MATLAB
net = newff(p, t, hiddenLlnodes, {'tansig','logsig'},'trainlm');

% Αρχικοποίηση παραμέτρων του MATLAB για χρήση κατά την εκπαίδευση.
% Συμβουλευτείτε το MATLAB Documentation για κάθε μία από αυτές τις
% παραμέτρους
```

```

net.inputs{1,1}.processFcns = {};
net.outputs{2}.processFcns = {};

net.initFcn = 'initlay';
net.layers{1}.initFcn = 'initnw';
net.layers{2}.initFcn = 'initnw';

tElapsed = 0;

% Στην επόμενη εντολή αρχικοποιούνται τα βάρη του νευρωνικού δικτύου
% σε τυχαίες τιμές σύμφωνα με τον αλγόριθμο Nguyen-Widrow.
% Ο αλγόριθμος χρησιμοποιεί τη γεννήτρια ψευδοτυχαίων αριθμών της
% συνάρτησης rand του MATLAB η οποία θα πρέπει να αρχικοποιηθεί πριν
% από κάθε «τρέξιμο» (run).
% Συμβουλευτείτε το MATLAB Documentation για περισσότερες πληροφορίες.
net = init(net);

% Αρχικοποίηση άλλων χρήσιμων παραμέτρων του MATLAB για χρήση κατά την
% εκπαίδευση.
% Συμβουλευτείτε το MATLAB Documentation για τις παραμέτρους αυτές
net.divideFcn = '';
net.trainParam.show = 5000;
net.trainParam.epochs = 5000;
net.trainParam.goal = 1e-8;
net.trainParam.lr = 0.05;
net.trainParam.min_grad = (2*1e-006)/(3*size(p,2)); %1e-006;
tStart = tic;

% Κλήση της συνάρτησης train για την εκπαίδευση του δικτύου
% Οι παράμετροι σε σχόλιο χρησιμοποιούνται με validation set
% Συμβουλευτείτε το MATLAB Documentation για περισσότερες πληροφορίες.
[net,tr] = train(net, p, t); % , [], [], val); validation dataset

% Για τον υπολογισμό του χρόνου εκπαίδευσης
tElapsed = toc(tStart);
tStart = 0;

% Υπολογισμός της επίδοσης του αλγορίθμου εκπαίδευσης σε εποχές
[perfr, perfc] = size(tr.perf);
convergence = perfc-1;

% Έλεγχος του δικτύου με τα δεδομένα ελέγχου μετά την ολοκλήρωση της
% εκπαίδευσης και υπολογισμός του πίνακα σύγχυσης
% Συμβουλευτείτε το MATLAB Documentation για περισσότερες πληροφορίες.
y = net(tp);
[c,cm,ind,per] = confusion(tt,y);

```