

Documentatie Proiect IA

Dumitrascu Lucian-Cosmin

Grupa 232

Pentru a antrena retea am folosit un *patternnet* (un singur strat) cu 100 de perceptroni pe stratul ascuns si 5 perceptroni pe stratul de iesire.

Patternnet foloseste pe stratul ascuns functia de activare *tansig* si pe stratul de iesire functia *softmax*.

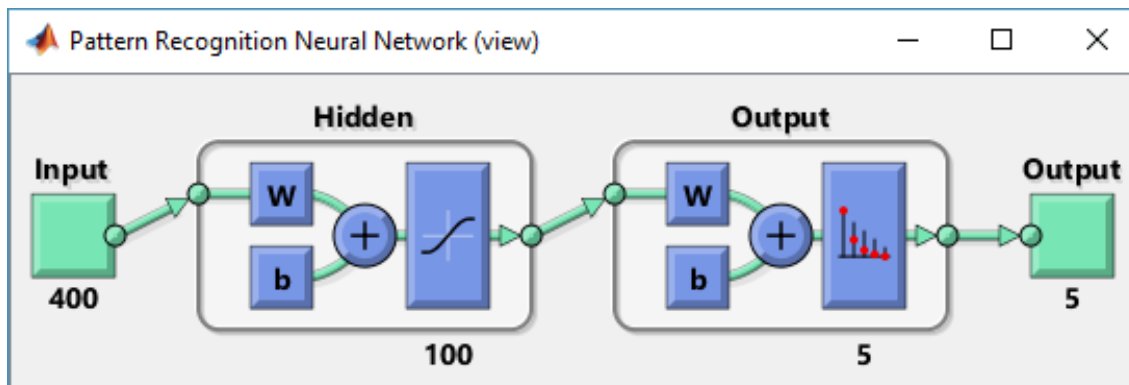
Pentru antrenare am folosit functia *traincgp*, iar pentru calcul performanta am folosit *crossentropy*. De asemenea, la antrenare am folosit si *regularizare* de 25%.

In prima faza am despartit multimea de antrenare in doua parti : una in proportie de 90%(antrenare) si cealalta in proportie de 10%(testare).

Am facut acest lucru deoarece voiam sa vad cum se comporta retea pe un set de date pe care nu l-a mai intalnit.

Pentru antrenare **nu am folosit** datele de intrare furnizate pe site. In schimb, le-am alterat (acel 90%) cu un zgomot repartizat normal pe intervalul $[0,0.2]$; practic am generat o matrice de zgomot pe care am sczut-o si adunat-o la datele de antrenare de pe site. Am facut asta de 4 ori, deci am generat 4 zgomote pe care le-am sczut si adunat la datele initiale. **Am folosit rezultatele acestor operatii pentru a-mi antrena retea** (71080 de observatii in total).

Antrenarea dura intre 200 si 400 de epoci. De obicei urmaream ca gradientul meu sa fie in jur de 0.006. Performanta pe datele de test este destul de buna (in general 91%+).



Rezultate 10-fold cross validation

i = 1

totalAccuracy = 0.8703

i = 2

totalAccuracy = 1.7236

i = 3

totalAccuracy = 2.5839

i = 4

totalAccuracy = 3.4634

i = 5

totalAccuracy = 4.3358

i = 6

totalAccuracy = 5.1949

i = 7

totalAccuracy = 6.0571

i = 8

totalAccuracy = 6.9254

i = 9

totalAccuracy = 7.7654

i = 10

totalAccuracy = 8.6488

crossValidationAccuracy = 0.8649

Matricile de confuzie

	1	2	3	4	5	
1	201 20.3%	2 0.2%	10 1.0%	7 0.7%	8 0.8%	88.2% 11.8%
2	3 0.3%	182 18.4%	17 1.7%	6 0.6%	8 0.8%	84.3% 15.7%
3	18 1.8%	12 1.2%	156 15.8%	8 0.8%	13 1.3%	75.4% 24.6%
4	7 0.7%	6 0.6%	7 0.7%	119 12.0%	1 0.1%	85.0% 15.0%
5	3 0.3%	3 0.3%	4 0.4%	2 0.2%	185 18.7%	93.9% 6.1%
	86.6% 13.4%	88.8% 11.2%	80.4% 19.6%	83.8% 16.2%	86.0% 14.0%	85.3% 14.7%
	1	2	3	4	5	
Output Class	Target Class					

Confusion Matrix						
Output Class	1	2	3	4	5	
	192 19.5%	3 0.3%	13 1.3%	11 1.1%	2 0.2%	86.9% 13.1%
	1 0.1%	182 18.4%	7 0.7%	4 0.4%	4 0.4%	91.9% 8.1%
	10 1.0%	7 0.7%	173 17.5%	12 1.2%	9 0.9%	82.0% 18.0%
	8 0.8%	2 0.2%	11 1.1%	119 12.1%	2 0.2%	83.8% 16.2%
	1 0.1%	9 0.9%	8 0.8%	4 0.4%	193 19.6%	89.8% 10.2%
	1	2	3	4	5	
Target Class						

		Source Class					
		1	2	3	4	5	
Output Class	1	192 19.4%	4 0.4%	16 1.6%	7 0.7%	2 0.2%	86.9% 13.1%
	2	3 0.3%	184 18.6%	12 1.2%	10 1.0%	3 0.3%	86.8% 13.2%
	3	6 0.6%	7 0.7%	158 16.0%	5 0.5%	5 0.5%	87.3% 12.7%
	4	7 0.7%	3 0.3%	13 1.3%	118 11.9%	2 0.2%	82.5% 17.5%
	5	7 0.7%	12 1.2%	13 1.3%	1 0.1%	198 20.0%	85.7% 14.3%
		89.3% 10.7%	87.6% 12.4%	74.5% 25.5%	83.7% 16.3%	94.3% 5.7%	86.0% 14.0%
		1	2	3	4	5	

Output Class	1	2	3	4	5	
1	200 20.2%	4 0.4%	14 1.4%	14 1.4%	3 0.3%	85.1% 14.9%
2	1 0.1%	182 18.4%	15 1.5%	2 0.2%	2 0.2%	90.1% 9.9%
3	8 0.8%	9 0.9%	163 16.5%	9 0.9%	4 0.4%	84.5% 15.5%
4	4 0.4%	4 0.4%	6 0.6%	124 12.6%	1 0.1%	89.2% 10.8%
5	3 0.3%	4 0.4%	11 1.1%	1 0.1%	200 20.2%	91.3% 8.7%
	92.6% 7.4%	89.7% 10.3%	78.0% 22.0%	82.7% 17.3%	95.2% 4.8%	88.0% 12.0%
	1	2	3	4	5	

Output Class	1	2	3	4	5	
1	200 20.3%	1 0.1%	11 1.1%	13 1.3%	2 0.2%	88.1% 11.9%
2	5 0.5%	174 17.6%	10 1.0%	6 0.6%	4 0.4%	87.4% 12.6%
3	13 1.3%	9 0.9%	165 16.7%	7 0.7%	7 0.7%	82.1% 17.9%
4	9 0.9%	6 0.6%	8 0.8%	135 13.7%	0 0.0%	85.4% 14.6%
5	3 0.3%	4 0.4%	7 0.7%	1 0.1%	187 18.9%	92.6% 7.4%
	87.0% 13.0%	89.7% 10.3%	82.1% 17.9%	83.3% 16.7%	93.5% 6.5%	87.2% 12.8%
	1	2	3	4	5	

Output Class	1	2	3	4	5	
1	184 18.6%	6 0.6%	11 1.1%	6 0.6%	3 0.3%	87.6% 12.4%
2	2 0.2%	162 16.4%	9 0.9%	5 0.5%	9 0.9%	86.6% 13.4%
3	13 1.3%	17 1.7%	172 17.4%	15 1.5%	7 0.7%	76.8% 23.2%
4	5 0.5%	5 0.5%	5 0.5%	137 13.9%	1 0.1%	89.5% 10.5%
5	4 0.4%	4 0.4%	12 1.2%	0 0.0%	193 19.6%	90.6% 9.4%
	88.5% 11.5%	83.5% 16.5%	82.3% 17.7%	84.0% 16.0%	90.6% 9.4%	85.9% 14.1%
	1	2	3	4	5	

Output Class \ Target Class	1	2	3	4	5	
1	206 20.9%	4 0.4%	8 0.8%	14 1.4%	4 0.4%	87.3% 12.7%
2	3 0.3%	157 15.9%	12 1.2%	3 0.3%	4 0.4%	87.7% 12.3%
3	11 1.1%	14 1.4%	160 16.2%	12 1.2%	15 1.5%	75.5% 24.5%
4	1 0.1%	6 0.6%	5 0.5%	129 13.1%	2 0.2%	90.2% 9.8%
5	5 0.5%	4 0.4%	9 0.9%	0 0.0%	199 20.2%	91.7% 8.3%
	91.2% 8.8%	84.9% 15.1%	82.5% 17.5%	81.6% 18.4%	88.8% 11.2%	86.2% 13.8%

Output Class	1	2	3	4	5	
1	196 19.9%	5 0.5%	10 1.0%	5 0.5%	2 0.2%	89.9% 10.1%
2	1 0.1%	136 13.8%	11 1.1%	8 0.8%	3 0.3%	85.5% 14.5%
3	7 0.7%	10 1.0%	183 18.5%	6 0.6%	6 0.6%	86.3% 13.7%
4	13 1.3%	8 0.8%	11 1.1%	141 14.3%	1 0.1%	81.0% 19.0%
5	5 0.5%	8 0.8%	8 0.8%	2 0.2%	201 20.4%	89.7% 10.3%
	88.3% 11.7%	81.4% 18.6%	82.1% 17.9%	87.0% 13.0%	94.4% 5.6%	86.8% 13.2%
	1	2	3	4	5	

Confusion Matrix						
Output Class	1	2	3	4	5	
	174 17.6%	6 0.6%	17 1.7%	18 1.8%	4 0.4%	79.5% 20.5%
	1 0.1%	179 18.1%	12 1.2%	7 0.7%	4 0.4%	88.2% 11.8%
	6 0.6%	10 1.0%	151 15.3%	11 1.1%	9 0.9%	80.7% 19.3%
	10 1.0%	3 0.3%	9 0.9%	141 14.3%	1 0.1%	86.0% 14.0%
	6 0.6%	7 0.7%	13 1.3%	4 0.4%	184 18.6%	86.0% 14.0%
						Target Class
						1 2 3 4 5

Confusion Matrix						
Output Class	1	2	3	4	5	
	216 21.9%	5 0.5%	17 1.7%	10 1.0%	4 0.4%	85.7% 14.3%
	1 0.1%	182 18.4%	8 0.8%	2 0.2%	4 0.4%	92.4% 7.6%
	7 0.7%	7 0.7%	141 14.3%	9 0.9%	11 1.1%	80.6% 19.4%
	6 0.6%	4 0.4%	4 0.4%	135 13.7%	1 0.1%	90.0% 10.0%
	2 0.2%	7 0.7%	4 0.4%	2 0.2%	198 20.1%	93.0% 7.0%
						Target Class
						1 2 3 4 5

Codul

```
clear all
clc
load('trainData.mat')
load('testData.mat')
trainVectors=trainVectors';
trainLabels=trainLabels';
trainVectors1=trainVectors1';
trainLabels1=trainLabels1';
total = size(trainVectors, 2);
antrenare = floor(total * 0.90); %rotunjeste rezultatul
verificare = total - antrenare;
ind = randperm(total);%amsteca datele de intrare
ind1 = ind(1:antrenare);%genereaza indicii pentru antrenare
ind2 = ind(antrenare + (1:verificare)); %genereaza indicii pentru testare
ind1=ind1';
x1 = trainVectors(:, ind1);%genereaza date pentru antrenare
t1 = trainLabels(:, ind1);% genereaza etichete pentru antrenare
x2 = trainVectors(:, ind2);%genereaza date pentru testare
t2 = trainLabels(:, ind2);% genereaza etichete pentru testare
t = 0.2*randn(size(x1));%genereaza zgomot random in [0,0.2]
xgen=x1+t;%genereaza date 'alterate'
xgen2=x1-t;%genereaza date 'alterate'
t = 0.2*randn(size(x1));%genereaza zgomot random in [0,0.2]
xgen3=x1+t;%genereaza date 'alterate'
xgen4=x1-t;%genereaza date 'alterate'
t = 0.2*randn(size(x1));%genereaza zgomot random in [0,0.2]
xgen5=x1+t;%genereaza date 'alterate'
xgen6=x1-t;%genereaza date 'alterate'
t = 0.2*randn(size(x1));%genereaza zgomot random in [0,0.2]
xgen7=x1+t;%genereaza date 'alterate'
xgen8=x1-t;%genereaza date 'alterate'
xtrain=[xgen xgen2 xgen3 xgen4 xgen5 xgen6 xgen7 xgen8];%concateneaza datele
de antrenare alterate
ttrain=[t1 t1 t1 t1 t1 t1 t1 t1];%concateneaza etichetele
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
classes=zeros(5,size(xtrain,2));
for i=1:size(xtrain,2)
if(ttrain(1,i)==1)
classes(1,i)=1;
end
end
for i=1:size(xtrain,2)
if(ttrain(1,i)==2)
classes(2,i)=1;
end
end
for i=1:size(xtrain,2)
if(ttrain(1,i)==3)
classes(3,i)=1;
end
end
for i=1:size(xtrain,2)
```

% am incercat sa folosesc ind2vec,
%dar parea ceva in neregula cu datele ,
%asa ca am facut asta manual

[illegible]

```

partitions = cvpartition(length(trainVectors),'kfold',10);
totalAccuracy = 0;
for i=1:10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
net = patternnet(100);
net.trainFcn='traincgp'; %initializare retea
net.performFcn = 'crossentropy';
net.performParam.regularization =0.25 ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
trainIndex{i} = find(partitions.training(i));%gasesc partitia de antrenare cu
indicele i
testIndex{i} = find(partitions.test(i));%gasesc partitia de test cu indicele
i
testV{i} = trainVectors(:, testIndex{i});%generez partitie pentru testare
testL{i} = labels(:, testIndex{i});%generez etichete pentru testare
trainV{i} = trainVectors(:, trainIndex{i});%generez partitie pentru antrenare
trainL{i} = labels(:, trainIndex{i});%generez etichete pentru antrenare
i
[net,tr] = train(net,trainV{i},trainL{i});%antrenez
crossValLabel=sim(net,testV{i});%prezic etichete
figure, plotconfusion(testL{i},crossValLabel);%plotez matricea de confuzie
crossValLabel=vec2ind(crossValLabel);
checkLabel=vec2ind(testL{i});
matchedLabels = sum(crossValLabel == checkLabel);%verific cate etichete sunt
prezise corect
accuracy = matchedLabels / length(testIndex{i});%calculez performanta
totalAccuracy = totalAccuracy + accuracy%retin performantele
end
crossValidationAccuracy = totalAccuracy / 10 %calculez performanta medie

*****

```