

Elektrotehnički fakultet u Beogradu



# IZVEŠTAJ PROJEKTOG ZADATKA IZ NEURALNIH MREŽA

**VARIJANTA P=8**

Studenti:

Petković Uroš 2016/0186

Arandelović Katarina 2016/0180

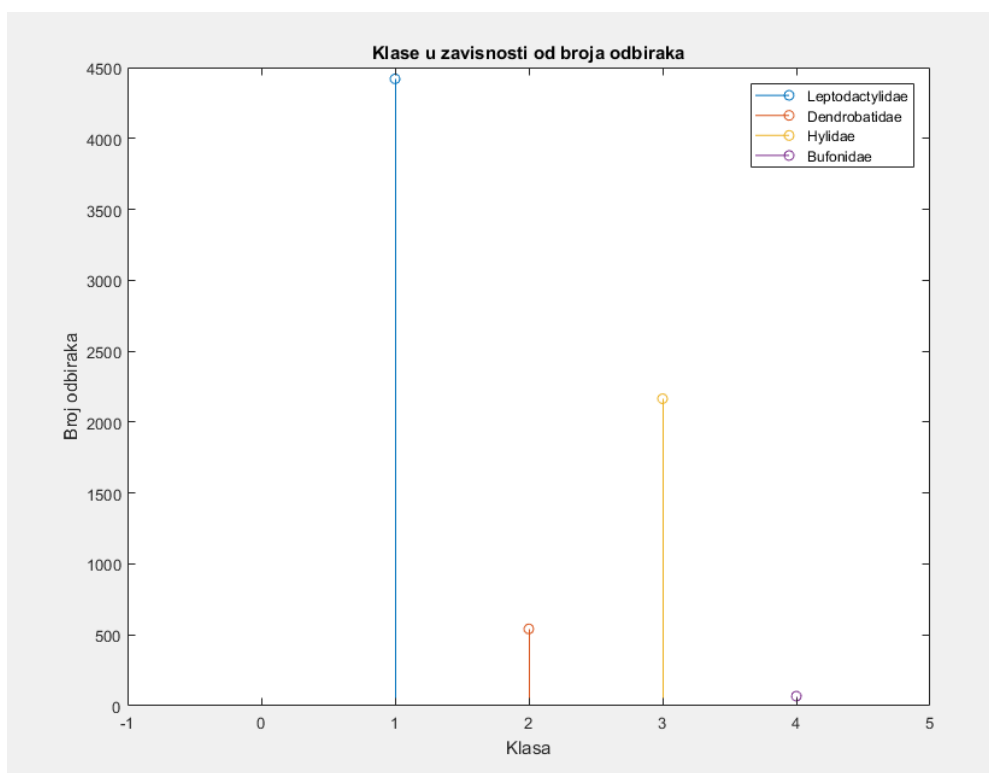
Mentor:

Novičić Marija

Beograd, januar 2020.

## Projektni zadatak- klasifikacija populacije žaba po familijama

Cilj ovog projekta je klasifikacija populacije žaba na familije. Izmereni su audio zvukovi koji predstavljaju karakteristične zvukove koji određene žabe ispuštaju, nakon čega su oni normalizovani od 0 do 1 i pretvoreni u odgovarajuća obeležja pogodna za dalju analizu. Takvih obeležja ima 22 i na osnovu njih se vrši klasifikacija žaba na 4 familije: Leptodactylidae, Hylidae, Dendrobatidae i Bufonidae. Iz razloga što za četvrtu klasu ima dosta manje podataka nego za ostale 3, možemo datu klasu izbaciti iz optičaja klasifikacije i vršiti klasifikaciju samo prve 3 klase. Na raspolaganju nam je Excel dokument sa podacima koji učitavamo. Dati podaci predstavljaju MFCC koeficijente govora žaba. Nakon učitavanja podataka izvršili smo podelu na klase i ta podela biće prikazana na sledećem grafiku.

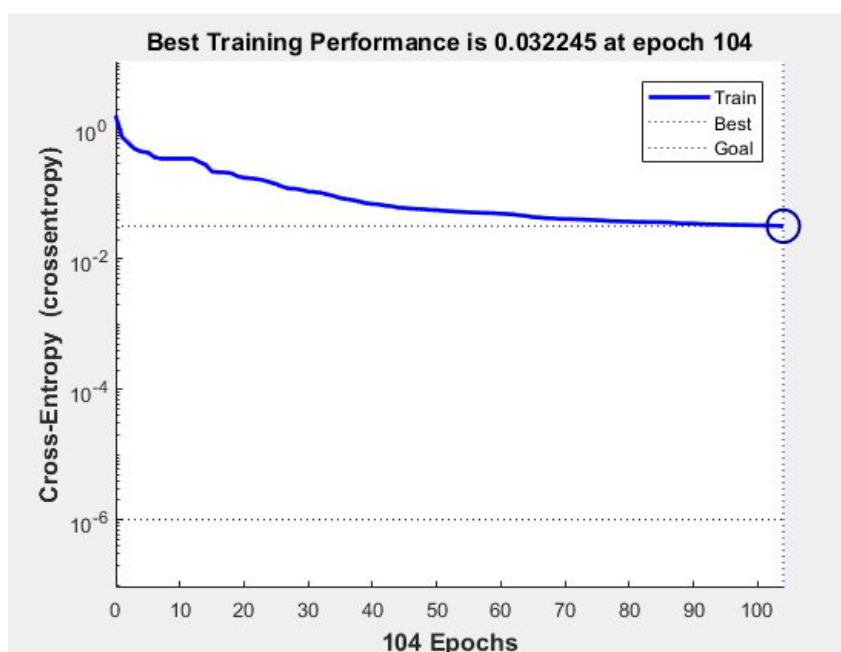


Na ovom grafiku se može videti da imamo 4 klase koje su nebalansirane. Naime, najviše ima odbiraka u klasi 1, odnosno klasi Leptodactylidae, a ekstremno manje u klasi Bufonidae koja se izostavlja iz dalje analize. Pošto su date klase nebalansirane, koristićemo odgovarajuću kriterijumsku funkciju F1 Score i to tako što ćemo za svaku klasu izračunati njen F1 Score i nakon toga naći srednju vrednost svih F1 Score-ova. Takva funkcija biće merilo tačnosti naše neuralne mreže. Pored toga, algoritam je pokrenut i za različite oblike strukture, koeficijenta regularizacije, transfer funkcije, kao i odgovarajućih težinskih matrica za drugu i treću klasu. U tom slučaju imamo dve petlje po težinskim faktorima za svaku klasu, kako bismo našli odgovarajući težinski faktor za obe klase. Jasno je da neće biti isti jer i broj odbiraka klasa nije

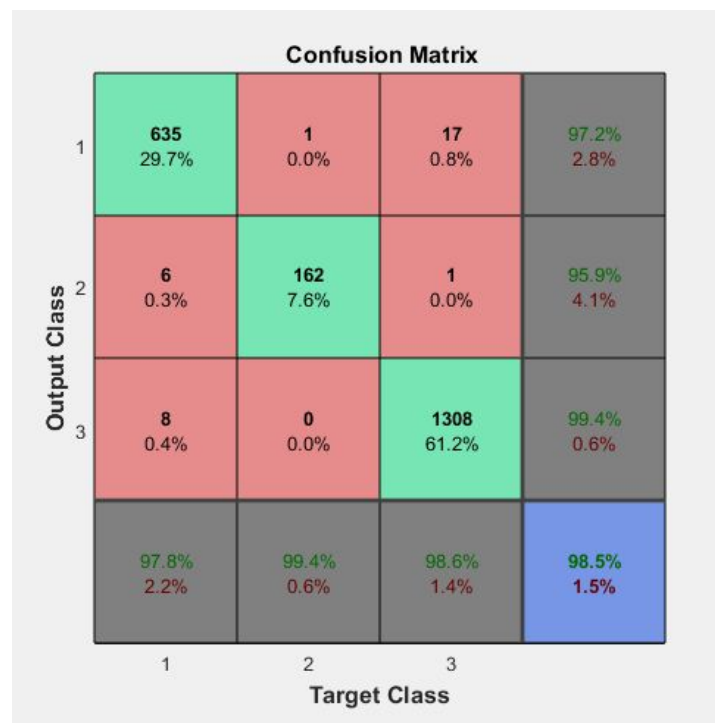
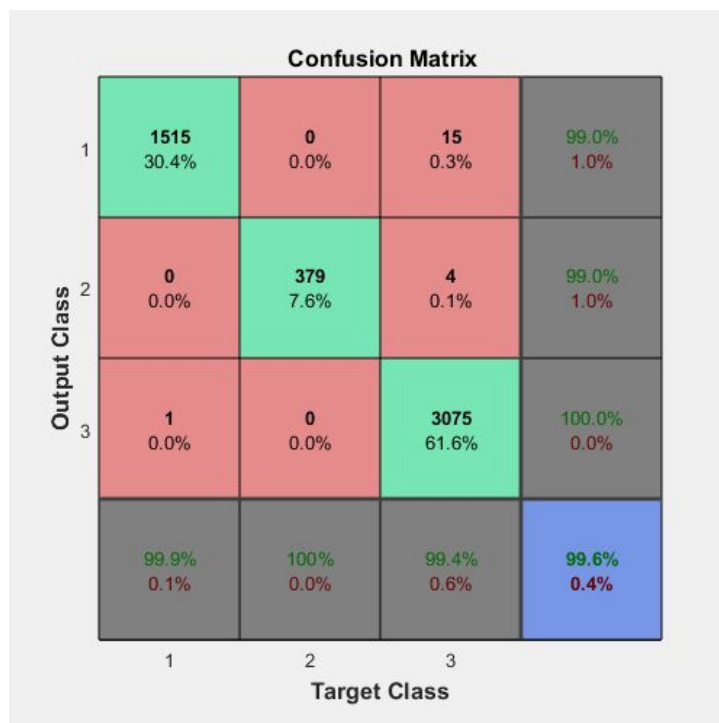
isti, već je treća klasa dosta manja od druge, a samim tim još više i od prve. Najpre razvrstane klase izmešamo koristeći funkciju `randperm()` kako ne bismo uzimali prvih 70% za trening skup, a ostalih 30% za test skup, već neke random odbirke. Na osnovu test odbiraka biće izvršeno testiranje najbolje dobijene mreže dok se za trening skup vrši krosvalidacija. Određeni broj odbiraka odvojen je za krosvalidaciju u kojoj računamo kriterijumsku funkciju i odlučujemo da li je naša celokupna trenutna struktura neuralne mreže najbolja. Pošto vršimo klasifikaciju nad 3 klase, naš izlaz projektovan je kao trobitna vrednost i to tako da je za prvu klasu izlaz jednak 100, za drugu klasu 010 i za treću klasu 001. Kriterijumska funkcija koja je korišćena jeste `crossentropy` jer je ona dobar izbor za probleme klasifikacije, dok naprimera `mse` predstavlja dobar izbor kriterijumske funkcije ukoliko radimo neku regresiju. Za pokrenute hiperparametre, dobili smo najbolje rezultate:

- najbolja struktura: [18,10,11]
- najbolja aktivaciona funkcija 'tansig'
- najbolji koeficijent regularizacije 0.21
- najbolji F1 Score 0.9930
- najbolji broj epoha 104
- najbolji  $w_1=8$  i best  $w_2=2$

Na slikama ispod biće prikazani struktura neuralne mreže kao i kriva performansi:



Nakon nalaženja najboljih parametara, istrenirana je neuralna mreža za najbolje dobijene parametre, pri čemu smo u ovom slučaju istrenirali mrežu na celokupnom trening skupu, ne deleći ga na trening i validation set. Za tako dobijenu neuralnu mrežu testirani su odbirci na osnovu test skupa, takođe prethodno izmešanog. Na slikama ispod biće prikazani rezultati dobijeni za trening i test konfuzione matrice.



Odakle vidimo da je greška skoro nula na trening skupu, a na test skupu oko 1% što znači da je naša neuralna mreža dobro obučena samim tim što daje podjednako dobre rezultate i na trening i na test skup. Moguće da je za neke druge parametre moglo biti još bolje rešenje, ali i ovi rezultati su sasvim zadovoljavajući.

U nastavku će biti prikazan kod projektnog zadatka.

## KOD ZADATKA

```
% Ucitavanje podataka i podela na trening i test skup

clc; clear all;

Podaci=readtable('Frogs_MFCCs.csv'); %Ucitavanje fajla
Podaci1=Podaci(:,1:22); %Prve 22 kolone su obelezja po 7195 odbiraka
Podaci1=table2array(Podaci1);%Vrsimo prebacivanje iz table u double
D=[Podaci1(673:1214,:)]; %Odvajamo podatke za klasu D
B=[Podaci1(6980:7047,:)]; %Odvajamo podatke za klasu B,L i H
L=[Podaci1(1:672,); Podaci1(1215:4692,); Podaci1(6596:6865,)]];
H=[Podaci1(4693:6595,); Podaci1(6866:6979,);Podaci1(7048:7195,:) ]];

XL=L; %Smestamo ih u ove vektore
XD=D;
XH=H;
XB=B;

%Nalazimo broj elemenata u svakoj klasi i mesamo ih radi boljeg obucavanja
NL=length(XL); indL=randperm(NL); XL=XL(indL,:);
ND=length(XD); indD=randperm(ND); XD=XD(indD,:);
NH=length(XH); indH=randperm(NH); XH=XH(indH,:);
NB=length(XB); %Ovu klasu ne koristimo

%Prikaz klasa u zavisnosti od broja podataka
figure(1); stem(1,NL); hold on;
stem(2,ND); hold on;
stem(3,NH); hold on;
stem(4,NB); hold off;
title('Klase u zavisnosti od broja odbiraka');
xlabel('Klasa');
ylabel('Broj odbiraka');
legend('Leptodactylidae','Dendrobatidae','Hylidae','Bufonidae');
xlim([-1 5]);

%Uzimamo 70% podataka za trening i 30% za test
%Zbog malog broja podataka klase 4,koristimo samo 3 klase pri cemu izlaz
%svake klase predstavlja trobitni broj 001,010 ili 100

XLtrening=XL(1:round(0.7*NL),:); %Trening
YLtrening=zeros(round(0.7*NL),3); %Trening izlaz 001
YLtrening(:,3)=1;
XLtest=XL(round(0.7*NL)+1:end,:); %Test
YLtest=zeros(round(0.3*NL),3); %Test izlaz 001
YLtest(:,3)=1;

XDtrening=XD(1:round(0.7*ND),:); %Trening
YDtrening=zeros(round(0.7*ND),3); %Trening izlaz 010
YDtrening(:,2)=1;
XDtest=XD(round(0.7*ND)+1:end,:); %Test
YDtest=zeros(round(0.3*ND),3); %Test izlaz 010
YDtest(:,2)=1;
```

```

XHtraining=XH(1:round(0.7*NH),:); %Trening
YHtraining=zeros(round(0.7*NH),3); %Trening izlaz 100
YHtraining(:,1)=1;
XHtest=XH(round(0.7*NH)+1:end,:); %Test
YHtest=zeros(round(0.3*NH),3); %Test izlaz 100
YHtest(:,1)=1;

Xtraining=[XLtraining;XDtraining;XHtraining]; %Pakujemo sve u jedan trening
Ytraining=[YLtraining;YDtraining;YHtraining];
Xtest=[XLtest;XDtest;XHtest]; %Pakujemo sve u jedan test
Ytest=[YLtest;YDtest;YHtest];

ind1=randperm(length(Xtraining)); %Mesamo smestene podatke
ind2=randperm(length(Xtest));

Xtraining=Xtraining(ind1,:); %Dobijamo konacni izgled treninga
Ytraining=Ytraining(ind1,:);

Xtest=Xtest(ind2,:); %Dobijamo konacni izgled testa
Ytest=Ytest(ind2,:);

Ntraining=length(Xtraining); %Broj odbiraka u trening i test skupu
Ntest=length(Xtest);
%% Krosvalidacija

%Inicijalizacija optimalnih hiperparametara
best_structure=[];
best_trainFcn = 'poslin';
best_reg= 0.2;
best_weight1 = 0;
best_weight2 = 0;
best_f1=0;
best_epoch=0;
% Pronalazenje optimalnih hiperparametara
for structure = {[7 10 7],[18 10 11], [5 5 4], [7 8 5]}
    for trainFcn = {'tansig','logsig','poslin'}
        for reg = {0.05, 0.1, 0.21}
            for weight1 = {2, 5, 8, 10}
                for weight2 = {2, 5, 8, 10}
                    net = patternnet(structure{1});
                    net.layers{1}.transferFcn = trainFcn{1};
                    net.layers{2}.transferFcn = trainFcn{1};
                    net.layers{3}.transferFcn = trainFcn{1};
                    net.performParam.regularization = reg{1};
                    net.divideFcn = 'divideind';
                    net.divideParam.trainInd = 1:round(0.8*Ntraining);
                    net.divideParam.valInd = round(0.8*Ntraining)+1:Ntraining;
                    net.divideParam.testInd = [];
                    Xval = Xtraining(:, net.divideParam.valInd);
                    Yval = Ytraining(:, net.divideParam.valInd);
                    net.trainParam.max_fail = 6;
                    net.trainParam.goal = 5e-6;
                    net.trainParam.epochs = 350;

```

```

net.trainParam.min_grad = 1e-8;
net.trainParam.showWindow = false;
w = ones(1,length(Xtrening));
w(1,Ytrening(1,:)==1) = weight1{1};
w(1,Ytrening(2,:)==1) = weight2{1};
[net,tr] = train(net, Xtrening, Ytrening, [], [], w);
Yval_pred = sim(net,Xval);
[c,cm_val] = confusion(Yval,Yval_pred);
cm_val = cm_val';
recall1 = cm_val(1,1)/(cm_val(2,1)+cm_val(1,1)+cm_val(3,1));
recall2 = cm_val(2,2)/(cm_val(2,2)+cm_val(1,2)+cm_val(3,2));
recall3 = cm_val(3,3)/(cm_val(2,3)+cm_val(1,3)+cm_val(3,3));
precision1 =
cm_val(1,1)/(cm_val(1,2)+cm_val(1,1)+cm_val(1,3));
precision2 =
cm_val(2,2)/(cm_val(2,2)+cm_val(2,1)+cm_val(2,3));
precision3 =
cm_val(3,3)/(cm_val(3,2)+cm_val(3,1)+cm_val(3,3));
f11=2*precision1*recall1/(precision1+recall1);
f12=2*precision2*recall2/(precision2+recall2);
f13=2*precision3*recall3/(precision3+recall3);
f1=(f11+f12+f13)/3; %Koristmo f1 score kao srednju vrednost
sva tri f1 scora
    if (f1 > best_f1)
        best_structure=structure{1};
        best_trainFcn=trainFcn{1};
        best_reg=reg{1};
        best_epoch=tr.best_epoch;
        best_weight1 = weight1{1};
        best_weight2 = weight2{1};
        best_f1=f1;
    end
end
end
end
end
end
end

```

%% Realizacija neuralne mreze sa najboljim hiperparametrima

```

net = patternnet([best_structure]);
net.layers{1}.transferFcn = best_trainFcn;
net.layers{2}.transferFcn = best_trainFcn;
net.layers{3}.transferFcn = best_trainFcn;
net.performParam.regularization = best_reg;
net.divideFcn = 'divideind';
net.divideParam.trainInd = [1:length(Xtrening)];
net.divideParam.valInd = [];
net.divideParam.testInd = [];

```

```

% Ocitavanje validacionih podataka
Xval = Xtrening(:, net.divideParam.valInd);
Yval = Ytrening(:, net.divideParam.valInd);
net.trainParam.max_fail = 6;
net.trainParam.goal = 1e-6;

```

```

net.trainParam.epochs = best_epoch;
net.trainParam.min_grad = 1e-7;
net.trainParam.showWindow = true;
w = ones(1,length(Xtrening));
w(1,Ytrening(1,:)==1)=best_weight1;
w(1,Ytrening(2,:)==1)=best_weight2;
[net,tr] = train(net, Xtrening, Ytrening, [], [], w);

%% Testiranje mreze

Ytrain_pred = sim(net,Xtrening);
Ytest_pred = sim(net,Xtest);

% Matrice konfuzije
figure()
plotconfusion(Ytrening, Ytrain_pred);
figure()
plotconfusion(Ytest, Ytest_pred);
%% Za prikaz klasa

NB=length(B);
figure(1); stem(1,NL); hold on;
stem(2,ND); hold on;
stem(3,NH); hold on;
stem(4,NB); hold off;
title('Klase u zavisnosti od broja odbiraka');
xlabel('Klasa');
ylabel('Broj odbiraka');
legend('Leptodactylidae', 'Dendrobatidae', 'Hylidae', 'Bufonidae');
xlim([-1 5]);

```