UNIVERZITET U BIHAĆU
TEHNIČKI FAKULTET
BIHAĆ

Auditorne vježbe iz predmeta

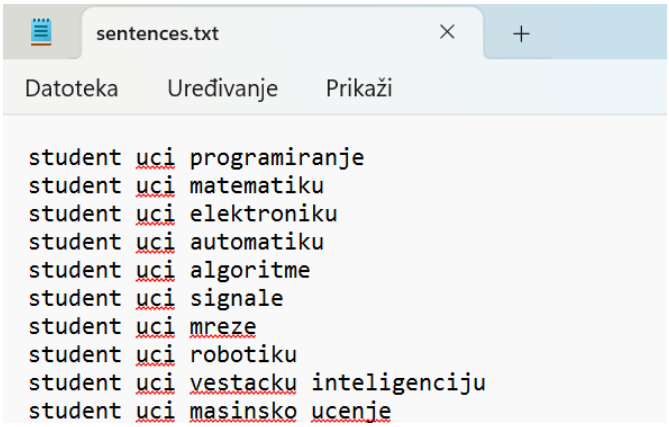
VJEŠTAČKA INTELIGENCIJA I EKSPERTNI SISTEMI

Una Drakulić, MA elektrotehnike
Viši asistent

Primjer 1

Razviti jednostavnu neuronsku mrežu koja na osnovu niza tekstualnih rečenica uči redoslijed i obrazac pojavljivanja riječi, te na osnovu naučenog modela generiše novu rečenicu koja logički slijedi dati niz. Ulaz u sistem je niz kratkih rečenica zapisanih u .txt fajlu. Svaka rečenica ima sličnu strukturu i pripada istom tematskom kontekstu. Neuronska mreža treba da nauči koja riječ najčešće slijedi nakon prethodne, te da na osnovu toga predvidi naredni tekstualni element.

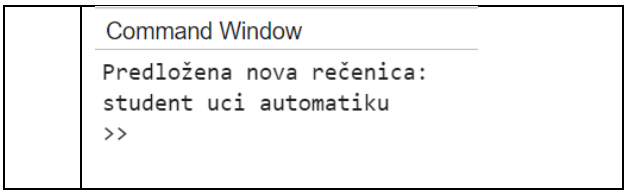
Ulazni podaci




```
clc; clear; close all;

%% 1. Učitavanje teksta
fid = fopen('sentences.txt','r');
C = textscan(fid,'%s','Delimiter','\n');
fclose(fid);
sentences = C{1};

%% 2. Ekstrakcija riječi
allWords = {};
for i = 1:length(sentences)
    words = split(lower(sentences{i}));
    allWords = [allWords; words];
end

uniqueWords = unique(allWords);
numWords = length(uniqueWords);

%% 3. Mapiranje riječi u indekse
word2idx = containers.Map(uniqueWords,1:numWords);
idx2word = uniqueWords;

%% 4. Kreiranje ulaza i izlaza
% Ulaz: [student uci] → Izlaz: oblast
X = [];
Y = [];

for i = 1:length(sentences)
    words = split(lower(sentences{i}));
    inputVec = zeros(numWords,1);
    inputVec(word2idx(words{1})) = 1;
    inputVec(word2idx(words{2})) = 1;

    outputVec = zeros(numWords,1);
    outputVec(word2idx(words{3})) = 1;

    X = [X inputVec];
    Y = [Y outputVec];
end
```

```
%% 5. Neuronska mreža
net = feedforwardnet(15);
net.trainParam.epochs = 500;
net = train(net,X,Y);

%% 6. Predikcija nove rečenice
testInput = zeros(numWords,1);
testInput(word2idx('student')) = 1;
testInput(word2idx('uci')) = 1;

prediction = net(testInput);
[~,idx] = max(prediction);

fprintf('Predložena nova rečenica:\n');
fprintf('student uci %s\n', idx2word{idx});
```

Izlaz:



Primjer 2

Razviti MLP neuronsku mrežu koja predviđa sentiment:

- 1 = pozitivna recenzija
- 0 = negativna recenzija

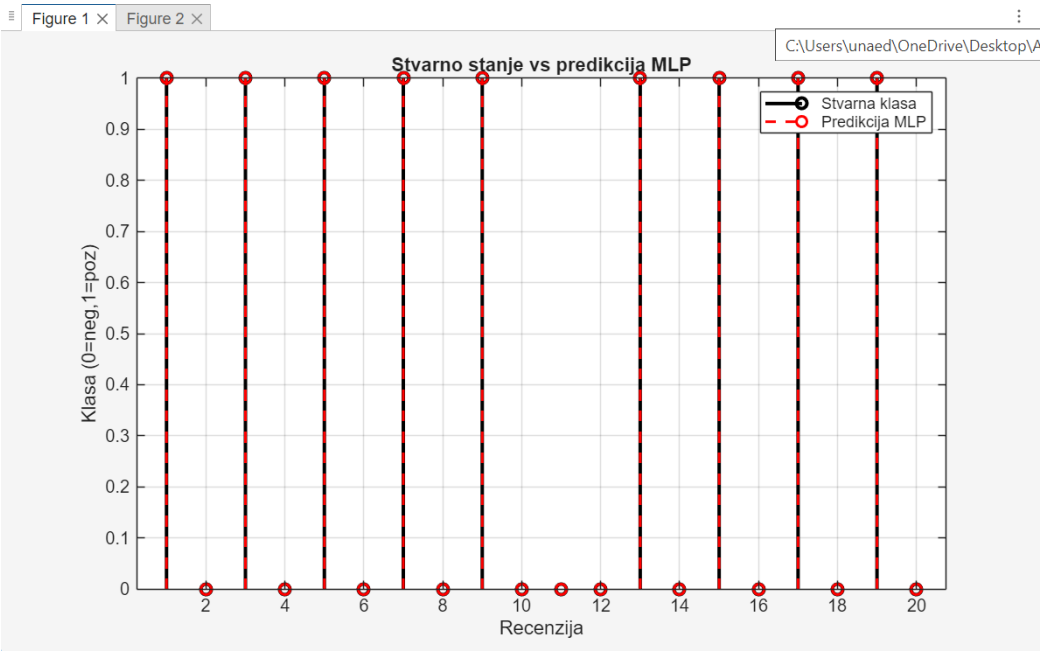
Tekst datoteka **reviews.txt** (20 recenzija)

I absolutely love this product, it works perfectly and exceeded my expectations
The item was terrible, I am very disappointed and would not recommend it
Fantastic quality, I am very happy with my purchase and everything works great
This is the worst experience I have ever had, very poor design and functionality
I am impressed, excellent performance and perfect for my needs
Awful, it broke after two uses and I hate it
Good value for money, really satisfied with the product
Terrible service, the item arrived damaged and support was awful
I highly recommend this, fantastic and awesome product
I dislike this product, very bad quality and poor performance
Perfect, just what I needed and works as described
I am very disappointed, it did not meet my expectations
Great item, excellent design and I love using it
Worst purchase ever, terrible quality and I hate it
Amazing product, very happy with the results and recommend it
Poor build, bad materials and disappointing overall
I am satisfied, good quality and works perfectly
Awful experience, would never buy again
Fantastic, excellent, and perfect, very happy
Terrible, worst item I have bought, do not recommend

```
clc; clear; close all;

%% 1. Ucitavanje fajla
fid = fopen('reviews.txt','r');
C = textscan(fid,'%s','Delimiter','\n');
fclose(fid);
reviews = C{1};
N = length(reviews);

%% 2. Lista pozitivnih i negativnih rijeci
positiveWords =
{'good','excellent','fantastic','love','great','happy','reco
mmend','awesome','perfect'};
negativeWords =
{'bad','poor','terrible','hate','awful','worst','disappointe
d'};

%% 3. Kreiranje ulaznih podataka
X = zeros(2,N); % 2 ulaza: broj pozitivnih i negativnih
rijeci
Y = zeros(1,N); % ciljni izlaz: 0=negativno, 1=pozitivno

for i = 1:N
    text = lower(reviews{i});
    words = split(text);
    posCount = sum(ismember(words,positiveWords));
    negCount = sum(ismember(words,negativeWords));
    X(:,i) = [posCount; negCount];

    % Pravilo za cilj
    if posCount > negCount
        Y(i) = 1;
    else
        Y(i) = 0;
    end
end

%% 4. Kreiranje i trening MLP
net = feedforwardnet(5); % skriveni sloj sa 5 neurona
net.trainParam.epochs = 500;
net = train(net,X,Y);

%% 5. Predikcija
Y_pred = net(X);
Y_class = double(Y_pred > 0.5);

%% 6. Tacnost
accuracy = sum(Y_class == Y)/N * 100;
fprintf('Tacnost MLP: %.2f %%\n', accuracy);

%% 7. Graf predikcija vs stvarno stanje
```

```
figure;
stem(1:N,Y,'k','LineWidth',2); hold on;
stem(1:N,Y_class,'r--','LineWidth',1.5);
xlabel('Recenzija'); ylabel('Klasa (0=neg,1=poz)');
title('Stvarno stanje vs predikcija MLP');
legend('Stvarna klasa','Predikcija MLP');
grid on;
```

Od ukupnog broja 20 recenzija:



Primjer 3

Razviti konvolucionu neuronsku mrežu (CNN) koja na osnovu ulazne slike prepoznaje da li se na slici nalazi horizontalna linija ili vertikalna linija. Ulaz u sistem su jednostavne binarne slike dimenzije 20×20 piksela. Izlaz neuronske mreže je klasa slike: 0 – vertikalna linija (crvena) i 1 – horizontalna linija (plava)

```
clc; clear; close all;

%% 1. Parametri
numImages = 200;
imgSize = 20;

%% 2. Generisanje RGB slika sa tankim linijama
X = zeros(imgSize,imgSize,3,numImages);
Y = categorical(zeros(numImages,1));

for i = 1:numImages
    img = zeros(imgSize,imgSize,3);

    if mod(i,2) == 0
        % Horizontalna linija - PLAVA
        img(10,:,3) = 1;          % B kanal
        Y(i) = categorical(1);    % Klasa 1
    else
        % Vertikalna linija - CRVENA
        img(:,10,1) = 1;          % R kanal
        Y(i) = categorical(0);    % Klasa 0
    end
```

```
    % Blagi šum (realističniji ulaz)
    img = img + 0.05*randn(size(img));
    img = max(min(img,1),0);

    X(:,:,:,i) = img;
end

%% 3. Podjela na trening i test skup
idx = randperm(numImages);
trainIdx = idx(1:150);
testIdx  = idx(151:end);

XTrain = X(:,:,:,trainIdx);
YTrain = Y(trainIdx);

XTest  = X(:,:,:,testIdx);
YTest  = Y(testIdx);

%% 4. CNN arhitektura
layers = [
    imageInputLayer([imgSize imgSize 3])

    convolution2dLayer(3,8,'Padding','same')
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,16,'Padding','same')
    reluLayer

    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer
];

%% 5. Opcije treninga
options = trainingOptions('adam', ...
    'MaxEpochs',10, ...
    'MiniBatchSize',16, ...
    'Verbose',false);

%% 6. Treniranje mreže
net = trainNetwork(XTrain,YTrain,layers,options);

%% 7. Testiranje
YPred = classify(net,XTest);
accuracy = sum(YPred == YTest)/numel(YTest)*100;

fprintf('Tačnost CNN-a: %.2f %%\n', accuracy);

%% 8. Vizualizacija test slika i predikcije
figure;
for i = 1:6
    subplot(2,3,i);
    imshow(XTest(:,:,:,i));
    title(['Predikcija: ' char(YPred(i))]);
end
```

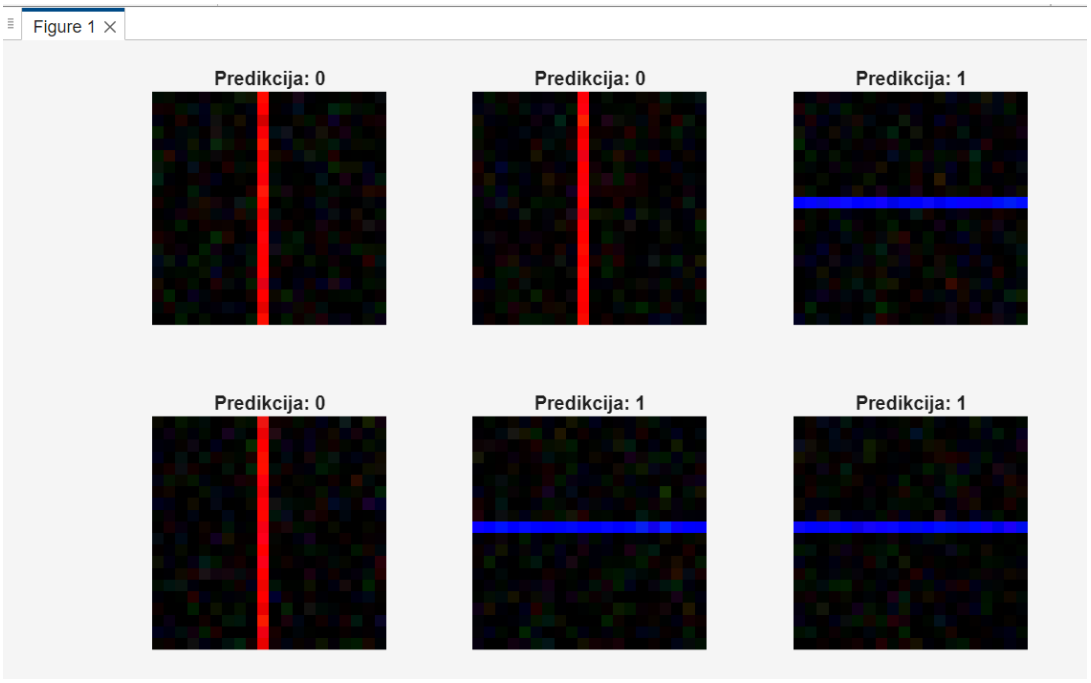


Figure 1
Predikcija: 0
Predikcija: 0
Predikcija: 1
Predikcija: 0
Predikcija: 1
Predikcija: 1

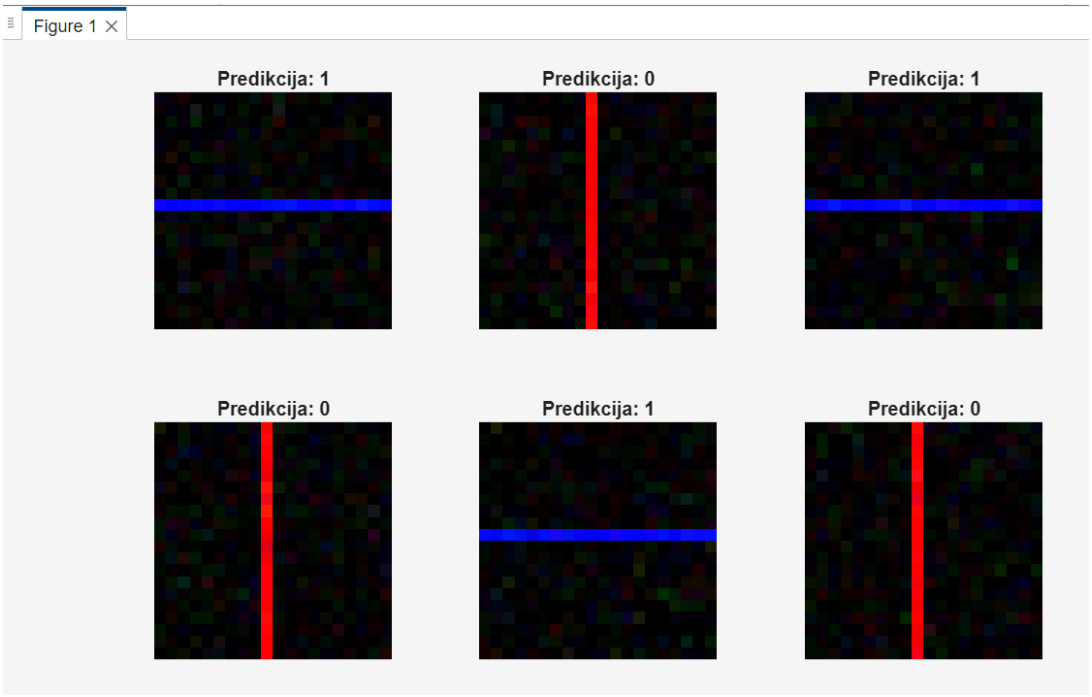


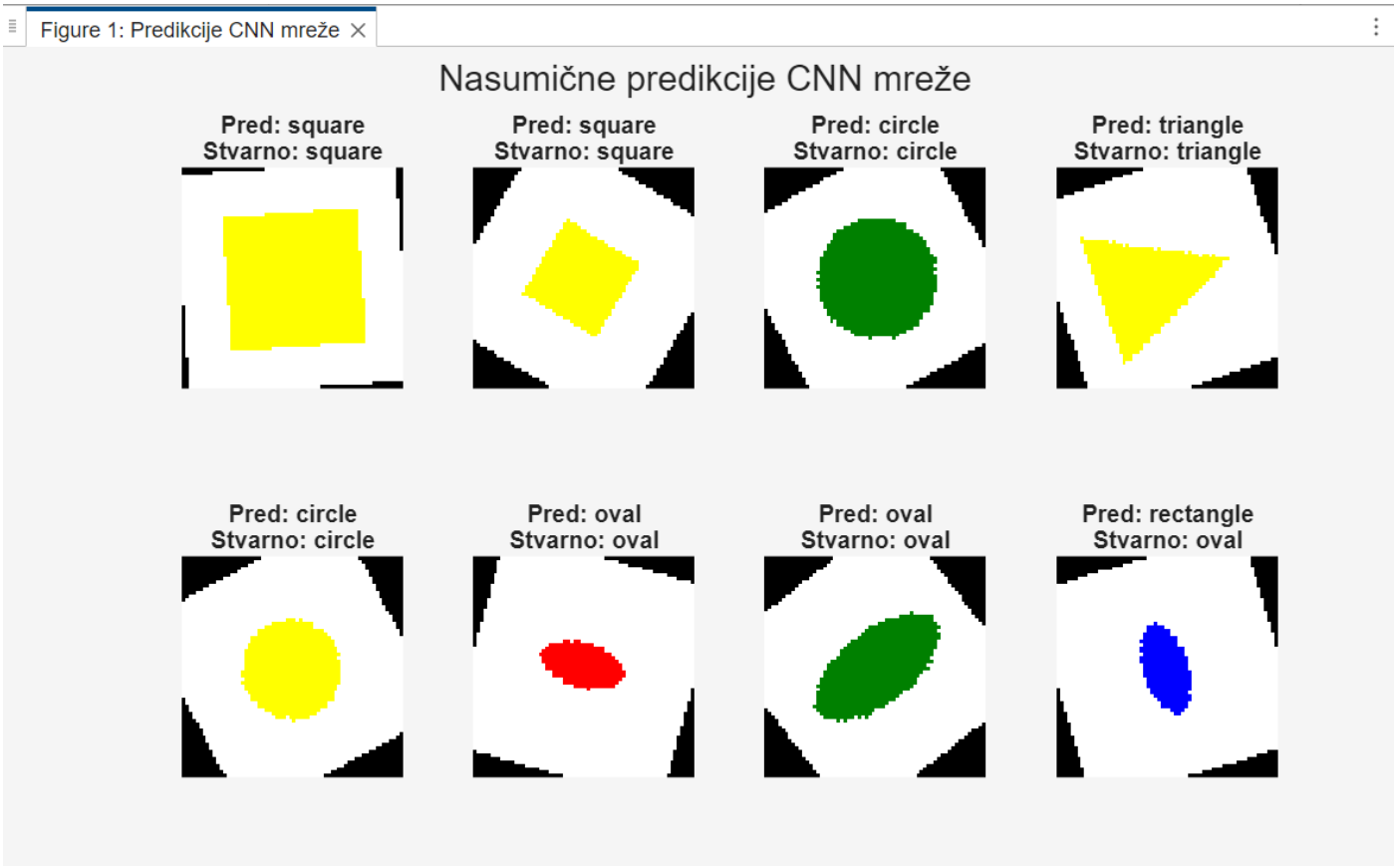
Figure 1
Predikcija: 1
Predikcija: 0
Predikcija: 1
Predikcija: 0
Predikcija: 1
Predikcija: 0

Primjer 4

Dat je skup slika geometrijskih oblika organizovan u direktorije po klasama (krug, kvadrat, trougao, itd.). Razviti konvolucionu neuronsku mrežu (CNN) u MATLAB-u koja na osnovu ulazne slike prepoznaje kojem geometrijskom obliku slika pripada.

```
%% CNN za prepoznavanje geometrijskih oblika i prikaz predikcija
clc; clear; close all;

% Folder dataset-a
datasetFolder = fullfile(pwd, 'Shapes_Dataset');

% Učitavanje slika
imds = imageDatastore(datasetFolder, ...
    'IncludeSubfolders',true, ...
    'LabelSource','foldernames');

% Promjena veličine slika na 64x64
inputSize = [64 64 3];
imds.ReadFcn = @(x) imresize(imread(x), inputSize(1:2));

% Podjela na trening i test skup
[imdsTrain, imdsTest] = splitEachLabel(imds, 0.8, 'randomized');

%% CNN arhitektura
layers = [
    imageInputLayer(inputSize)

    convolution2dLayer(3,8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,32,'Padding','same')
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(numel(unique(imds.Labels)))
    softmaxLayer
    classificationLayer];

%% Opcije treniranja
options = trainingOptions('adam', ...
    'MaxEpochs',5, ... % povećati po potrebi
    'MiniBatchSize',64, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsTest, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');

%% Treniranje mreže
net = trainNetwork(imdsTrain, layers, options);

%% Testiranje i tačnost
YPred = classify(net, imdsTest);
YTest = imdsTest.Labels;
accuracy = sum(YPred == YTest)/numel(YTest);
disp(['Tačnost na test skupu: ', num2str(accuracy*100), '%']);

%% Prikaz nasumičnih 8 predikcija
numExamples = 8;
```

```
idx = randperm(numel(imdsTest.Files), numExamples);

figure('Name','Predikcije CNN mreže');
for i = 1:numExamples
    img = readimage(imdsTest, idx(i));
    subplot(2,4,i);
    imshow(img);
    title({['Pred: ', char(YPred(idx(i)))], ['Stvarno: ', char(YTest(idx(i)))]}, ...
        'FontSize',10);
end
sgtitle('Nasumične predikcije CNN mreže');
```



Command Window

```
Tačnost na test skupu: 80.2%
```