

FFR135, Artificial Neural Networks
Home Problem 2
Two-layer perceptron

6 oktober 2019

Ella Guiladi
930509-0822
guiladi@student.chalmers.se

1 Two-layer perceptron

```

clear
clc
trainingSet = csvread('training_set.csv');
validationSet = csvread('validation_set.csv');
learningRate=0.01;
numberOfUpdates=10^6;
patternsValidationSet=5000;
M1=15;
M2=25;

firstLayer = [];
secondLayer = [];
output=0;

outputError=[];
secondLayerError=[];
firstLayerError=[];

firstLayerValidation = zeros(M1);
secondLayerValidation = zeros(M2);
outputValidation=0;

%initializing thresholds
firstThreshold = normrnd(0,1,[1,M1]);
secondThreshold = normrnd(0,1,[1,M2]);
outputThreshold = normrnd(0,1,[1,1]);

%initializing weights
inputWeight= normrnd(0,1,[M1, 2]);
hiddenWeight= normrnd(0,1,[M2,M1]);
outputWeight= normrnd(0,1,[1,M2]);

%training
for i=1:numberOfUpdates
    my = randi([1, 10000]);

    %forward propagation
    firstLayerSum=[];
    for s = 1:M1
        firstLayerSum(s)=sum(inputWeight(s,:).*trainingSet(my,1:2));
        firstLayer(s)=tanh(-firstThreshold(s)+firstLayerSum(s));
    end

    secondLayerSum=[];
    for q=1:M2
        secondLayerSum(q)=sum(hiddenWeight(q,:).*firstLayer);
        secondLayer(q) = tanh(-secondThreshold(q)+secondLayerSum(q));
    end

    outputSum=sum(outputWeight.*secondLayer);
    output=tanh(-outputThreshold+outputSum);

    %Backpropagation
    %errors are updates from right to left
    outputError=(trainingSet(my,3)-output)*(1-(tanh(-outputThreshold+outputSum))^2);

    for c=1:M2
        secondLayerError(c)=outputError*outputWeight(c)*(1-(tanh(-secondThreshold(c)+secondLayerSum(c)))^2);
    end
end

```

```

for d=1:M1
    firstLayerError(d)=sum(secondLayerError.*hiddenWeight(:,d))*(1-(←
        tanh(-firstThreshold(d)+firstLayerSum(d))^2));
end

%weight update
%neurons are updated from left to right
updatedInputWeight= [];
updatedHiddenWeight = [];
updatedOutputWeight = [];

for u=1:M1
    for v=1:2
        updatedInputWeight(u,v)= inputWeight(u,v)+learningRate*(←
            firstLayerError(u)*trainingSet(my,v);
    end
end

for z=1:M2
    for y=1:M1
        updatedHiddenWeight(z,y)= hiddenWeight(z,y)+learningRate*(←
            secondLayerError(z)*firstLayer(y);
    end
end

for h=1:M2
    updatedOutputWeight(h)= outputWeight(h)+learningRate*outputError*(←
        secondLayer(h);
end

%updating thresholds
newFirstThreshold = [];
newSecondThreshold = [];

for bc=1:M1
    firstThreshold(bc)=firstThreshold(bc)-learningRate*firstLayerError(←
        bc);
end

for de=1:M2
    newSecondThreshold(de)=secondThreshold(de)-learningRate*(←
        secondLayerError(de);
end

newOutputThreshold = outputThreshold-learningRate*outputError;

%write over old weights and thresholds
inputWeight=updatedInputWeight;
hiddenWeight=updatedHiddenWeight;
outputWeight=updatedOutputWeight;

newFirstThreshold=firstThreshold;
secondThreshold=newSecondThreshold;
outputThreshold=newOutputThreshold;

end

%validation
sumOfValidation=0;
for j=1:patternsValidationSet

    theSumInTheOutput1Validation=[];
    for pq = 1:M1
        theSumInTheOutput1Validation=0;
        for rs=1:2

```

```

        summationOutput1Validation=inputWeight(pq,rs)*validationSet(j,rs);
        theSumInTheOutput1Validation=theSumInTheOutput1Validation+summationOutput1Validation;
    end
    theSumInTheOutput11Validation(pq)=theSumInTheOutput1Validation;
    firstLayerValidation(j,pq)=tanh(-firstThreshold(pq)+theSumInTheOutput11Validation(pq));
end

theSumInTheOutput22Validation=[];
for tu=1:M2
    theSumInTheOutput2Validation=0;
    for vx=1:M1
        summationOutput2Validation=hiddenWeight(tu,vx)*firstLayerValidation(j,vx);
        theSumInTheOutput2Validation=theSumInTheOutput2Validation+summationOutput2Validation;
    end
    theSumInTheOutput22Validation(tu)=theSumInTheOutput2Validation;
    secondLayerValidation(j,tu) = tanh(-secondThreshold(tu)+theSumInTheOutput22Validation(tu));
end

theSumInTheOutput3Validation=0;
for f=1:M2
    summationOutput3Validation=outputWeight(f)*secondLayerValidation(j,f);
    theSumInTheOutput3Validation=theSumInTheOutput3Validation+summationOutput3Validation;
end

outputValidation(j)=tanh(-outputThreshold+theSumInTheOutput3Validation);

partSumOfValidation=abs(sign(outputValidation(j))-validationSet(j,3));
sumOfValidation=sumOfValidation+partSumOfValidation;

end

C = ((1/(2*patternsValidationSet))*(sumOfValidation))

disp('The run is done!')

csvwrite('w1.csv',inputWeight);
csvwrite('w2.csv',hiddenWeight);
csvwrite('w3.csv',outputWeight);
csvwrite('t1.csv',firstThreshold);
csvwrite('t2.csv',secondThreshold);
csvwrite('t3.csv',outputThreshold);

```