

```

irisData = readmatrix('iris-data.csv');
irisLabels = readmatrix('iris-labels.csv');

irisData = irisData / max(max(irisData));

dataInput = zeros(1,4);
dataOutput = zeros(40,40);
weightMatrix = zeros(40,40,4);

for i = 1:40
    for j = 1:40
        for k = 1:4
            weightMatrix(i,j,k) = rand();
        end
    end
end

initialLearningRate = 0.1;
learningRateDecayRate = 0.01;
initialwidth = 10;
widthDecayRate = 0.05;

for epoch = 1:10
    trainingOrder = randperm(150);
    learningRate = initialLearningRate * exp(-learningRateDecayRate * epoch);
    width = initialwidth * exp(-widthDecayRate * epoch);
    for p = 1:150
        dataInput = irisData(trainingOrder(p),:);
        winningNeuron = FindWinningNeuron(weightMatrix, dataInput);
        for i = 1:40
            for j = 1:40
                for k = 1:4
                    deltaWeight = FindDeltaWeight(learningRate, width,
winningNeuron, [i j], dataInput, weightMatrix(i, j, k));
                    weightMatrix(i, j, k) = weightMatrix(i, j, k) + deltaWeight(k);
                end
            end
        end
    end
end

finalWinningNeuron = zeros(150,2);
for p = 1:150
    dataInput = irisData(p,:);
    finalWinningNeuron(p,:) = FindWinningNeuron(weightMatrix, dataInput);
end

scatter(finalWinningNeuron(1:50,1), finalWinningNeuron(1:50,2),"red");
hold on;
scatter(finalWinningNeuron(51:100,1), finalWinningNeuron(51:100,2),"green");

```

```

hold on;
scatter(finalWinningNeuron(101:150,1), finalWinningNeuron(101:150,2),"blue");
hold on;

```

```

function vectorAngle = CalculateVectorAngle(weight, dataInput)

vectorAngle = acos(dot(weight, dataInput)/(norm(weight) * norm(dataInput)));

end

function winningNeuron = FindWinningNeuron(weightMatrix,dataInput)

winningNeuronValue = transpose(squeeze(weightMatrix(1,1,:)));
winningNeuron = [1 1];

for i = 1:40
    for j = 1:40
        if CalculateVectorAngle(winningNeuronValue,dataInput) >
CalculateVectorAngle(transpose(squeeze(weightMatrix(i,j,:))),dataInput)
            winningNeuronValue = transpose(squeeze(weightMatrix(i,j,:)));
            winningNeuron = [i j];
        end
    end
end

end

function deltaWeight = FindDeltaWeight(learningRate, width, winningNeuron,
targetOutput, dataInput, weight)

neighbourhoodFunction = exp ( (- norm(winningNeuron - targetOutput) ^ 2) / (2 *
width ^ 2));
deltaWeight = learningRate * neighbourhoodFunction * (dataInput - weight);

end

```