TrainingByGradientDecentMethodSigmoid.m

TestRawImages=csvread('MNIST/test.csv', 0, 0);

TestImages = zeros(10,784);

TestImages(1:10,1:784)=TestRawImages(1:10,2:785);

CorrectTestLabels= zeros(10,1);

CorrectTestLabels(1:10)=TestRawImages(1:10);

File=csvread('MNIST/train.csv', 0, 0);

Weights=zeros(10,785);

for i=1:10

for j=2:785

Weights(i,j)= rand(1,1) -0.5;

end

end

Weights(:,1)=1; %bias term initailize

noOfTrainingImages =7000;

TrainingImages = zeros(noOfTrainingImages,784);

TrainingImages(1:noOfTrainingImages,1:784)=File(1:noOfTrainingImages,2:785);

TrLab= zeros(noOfTrainingImages,1);

TrLab(1:noOfTrainingImages)=File(1:noOfTrainingImages);

CorrectLabels= zeros(10,noOfTrainingImages);

for i=1:10

k=1;

for j=1:noOfTrainingImages

if(TrLab(j,1)==(i-1))

CorrectLabels(i,k)=1;

k=k+1;

else

CorrectLabels(i,k)=-1;

k=k+1;

end

end

end

predictions=zeros(1,10);

noOfTestImages =10;

for k=1:noOfTestImages

for i=1:10

Weights(i,:)=GradientDecentMethodUsingSigmoid(Weights(i,:),TrainingImages,CorrectLabels(i,:));

TransposedEachTestImage=TestImages(k,:)';

weightsOfCurrentLabel=Weights(i,2:785);

weightedSum=weightsOfCurrentLabel\*TransposedEachTestImage;

weightedSum=weightedSum+Weights(i,1);

%bias term

predictions(1,i)=SigmoidActivationFunction(weightedSum);

end

CurrentCorrectLabel=CorrectTestLabels(k,1);

disp('Prediction');

disp(predictions);

disp('Correct Label');

disp(CurrentCorrectLabel);

end

function Weights = GradientDecentMethodUsingSigmoid(Weights,TrainingImages,CorrectLabels)

eta=0.05;

for epoch=1:100

for i=1:size(TrainingImages,1)

TransposedEachTrainingImage=TrainingImages(i,:)';

%We can view it as Oi of the mathematical equation

CurrentCorrectLabel=CorrectLabels(:,i);

%We can view it as yi of the mathematical equation

weightsOfCurrentLabel=Weights(1,2:785);

weightedSum=weightsOfCurrentLabel\*TransposedEachTrainingImage;

weightedSum=weightedSum+Weights(1,1);

%bias term

CalculatedLabel=SigmoidActivationFunction(weightedSum);

%We can view it as yi\* of the mathematical equation

error=CurrentCorrectLabel-CalculatedLabel;

delta=CalculatedLabel\*(1-CalculatedLabel)\*error; %d(x)=(x)(1-x).

Weights(1,1)=Weights(1,1)+(eta\*delta);

%If there is no error then the delta will be zero hence no change will

%be done to weight

dWeight=eta\*delta\*TransposedEachTrainingImage;

%Updating previous weights

for d=2:785

Weights(1,d)=Weights(1,d)+dWeight(d-1);

end

end

end

function g = SigmoidActivationFunction(x)

g = zeros(size(x));

g = 1.0 ./ ( 1.0 + exp(-x));

end

TrainingByGradientDecentMethodTanh.m

TestRawImages=csvread('MNIST/test.csv', 0, 0);

TestImages = zeros(10,784);

TestImages(1:10,1:784)=TestRawImages(1:10,2:785);

CorrectTestLabels= zeros(10,1);

CorrectTestLabels(1:10)=TestRawImages(1:10);

File=csvread('MNIST/train.csv', 0, 0);

Weights=zeros(10,785);

for i=1:10

for j=2:785

Weights(i,j)= rand(1,1) -0.5;

end

end

Weights(:,1)=1; %bias term initailize

noOfTrainingImages =7000;

TrainingImages = zeros(noOfTrainingImages,784);

TrainingImages(1:noOfTrainingImages,1:784)=File(1:noOfTrainingImages,2:785);

TrLab= zeros(noOfTrainingImages,1);

TrLab(1:noOfTrainingImages)=File(1:noOfTrainingImages);

CorrectLabels= zeros(10,noOfTrainingImages);

for i=1:10

k=1;

for j=1:noOfTrainingImages

if(TrLab(j,1)==(i-1))

CorrectLabels(i,k)=1;

k=k+1;

else

CorrectLabels(i,k)=-1;

k=k+1;

end

end

end

predictions=zeros(1,10);

noOfTestImages =10;

for k=1:noOfTestImages

for i=1:10

Weights(i,:)=GradientDecentMethodUsingTanh(Weights(i,:),TrainingImages,CorrectLabels(i,:));

TransposedEachTestImage=TestImages(k,:)';

weightsOfCurrentLabel=Weights(i,2:785);

weightedSum=weightsOfCurrentLabel\*TransposedEachTestImage;

weightedSum=weightedSum+Weights(i,1);

%bias term

predictions(1,i)=SigmoidActivationFunction(weightedSum);

end

CurrentCorrectLabel=CorrectTestLabels(k,1);

disp('Prediction');

disp(predictions);

disp('Correct Label');

disp(CurrentCorrectLabel);

end

function Weights = GradientDecentMethodUsingTanh(Weights,TrainingImages,CorrectLabels)

eta=0.05;

for epoch=1:100

for i=1:size(TrainingImages,1)

TransposedEachTrainingImage=TrainingImages(i,:)';

%We can view it as Oi of the mathematical equation

CurrentCorrectLabel=CorrectLabels(:,i);

%We can view it as yi of the mathematical equation

weightsOfCurrentLabel=Weights(1,2:785);

weightedSum=weightsOfCurrentLabel\*TransposedEachTrainingImage;

weightedSum=weightedSum+Weights(1,1);

%bias term

CalculatedLabel=tanh(weightedSum);

%We can view it as yi\* of the mathematical equation

error=CurrentCorrectLabel-CalculatedLabel;

delta=CalculatedLabel\*(1-(CalculatedLabel\*CalculatedLabel))\*error;

Weights(1,1)=Weights(1,1)+(eta\*delta);

%If there is no error then the delta will be zero hence no change will

%be done to weight

dWeight=eta\*delta\*TransposedEachTrainingImage;

%Updating previous weights

for d=2:785

Weights(1,d)=Weights(1,d)+dWeight(d-1);

end

end

end