Appendix

ReLU, softmax, early stopping 2019

Network 1

```
clear;clc;
%%
[xTrain, tTrain, xValid, tValid, xTest, tTest] = LoadCIFAR(3);

layers = [imageInputLayer([32 32 3])
fullyConnectedLayer(50)
reluLayer
fullyConnectedLayer(50)
reluLayer
fullyConnectedLayer(10)
softmaxLayer
classificationLayer];

options =  trainingOptions('sgdm',...
'MiniBatchSize', 8192,...
'ValidationData', {xValid, tValid},...
'ValidationFrequency', 30,...
'MaxEpochs',400,...
'Plots', 'Training-Progress',...
'L2Regularization', 0, ...
'Momentum', 0.9, ...
'ValidationPatience', 3, ...
'Shuffle', 'every-epoch', ...
'InitialLearnRate', 0.001);

net = trainNetwork(xTrain, tTrain, layers, options);
classify_Train = net.classify(xTrain);
classify_Valid = net.classify(xValid);
classify_Test = net.classify(xTest);

count1 = 0;
for i = 1:size(tTrain,1)
   if (classify_Train(i,:)~=tTrain(i,:))
      count1 = count1+1;
   end
end
C_Train = count1/size(tTrain,1)

count2 = 0;
for i = 1:size(tValid,1)
   if (classify_Valid(i,:)~=tValid(i,:))
      count2 = count2+1;
   end
end
```

```
C_Valid = count2/size(tValid,1)

count3 = 0;
for i = 1:size(tTest,1)
   if (classify_Test(i,:)~=tTest(i,:))
      count3 = count3+1;
   end
end
C_Test = count3/size(tTest,1)

%C_Train = 0.4368


%C_Valid = 0.5140


%C_Test = 0.5123


%Epoch 203 of 400
```

Network 2

```
clear;clc;
%%
[xTrain, tTrain, xValid, tValid, xTest, tTest] = LoadCIFAR(3);

layers = [imageInputLayer([32 32 3])
fullyConnectedLayer(50)
reluLayer
fullyConnectedLayer(50)
reluLayer
fullyConnectedLayer(50)
reluLayer
fullyConnectedLayer(10)
softmaxLayer
classificationLayer];

options =  trainingOptions('sgdm',...
'MiniBatchSize', 8192,...
'ValidationData', {xValid, tValid},...
'ValidationFrequency', 30,...
'MaxEpochs',400,...
'Plots', 'Training-Progress',...
'L2Regularization', 0, ...
'Momentum', 0.9, ...
'ValidationPatience', 3, ...
'Shuffle', 'every-epoch', ...
'InitialLearnRate', 0.003);

net = trainNetwork(xTrain, tTrain, layers, options);
classify_Train = net.classify(xTrain);
classify_Valid = net.classify(xValid);
classify_Test = net.classify(xTest);

count1 = 0;
for i = 1:size(tTrain,1)
   if (classify_Train(i,:)~=tTrain(i,:))
      count1 = count1+1;
   end
end
C_Train = count1/size(tTrain,1)

count2 = 0;
for i = 1:size(tValid,1)
   if (classify_Valid(i,:)~=tValid(i,:))
      count2 = count2+1;
   end
end
C_Valid = count2/size(tValid,1)
```

```
count3 = 0;
for i = 1:size(tTest,1)
    if (classify_Test(i,:)~=tTest(i,:))
        count3 = count3+1;
    end
end
C_Test = count3/size(tTest,1)


%C_Train = 0.4474

%C_Valid = 0.5296

%C_Test = 0.5247


%Epoch 165 of 400
```

Network 3

```
clear;clc;
%%
[xTrain, tTrain, xValid, tValid, xTest, tTest] = LoadCIFAR(3);

layers = [imageInputLayer([32 32 3])
fullyConnectedLayer(50)
reluLayer
fullyConnectedLayer(50)
reluLayer
fullyConnectedLayer(10)
softmaxLayer
classificationLayer];

options =  trainingOptions('sgdm',...
'MiniBatchSize', 8192,...
'ValidationData', {xValid, tValid},...
'ValidationFrequency', 30,...
'MaxEpochs',400,...
'Plots', 'Training-Progress',...
'L2Regularization', 0.2, ...
'Momentum', 0.9, ...
'ValidationPatience', 3, ...
'Shuffle', 'every-epoch', ...
'InitialLearnRate', 0.001);

net = trainNetwork(xTrain, tTrain, layers, options);
classify_Train = net.classify(xTrain);
classify_Valid = net.classify(xValid);
classify_Test = net.classify(xTest);

count1 = 0;
for i = 1:size(tTrain,1)
   if (classify_Train(i,:)~=tTrain(i,:))
      count1 = count1+1;
   end
end
C_Train = count1/size(tTrain,1)

count2 = 0;
for i = 1:size(tValid,1)
   if (classify_Valid(i,:)~=tValid(i,:))
      count2 = count2+1;
   end
end
C_Valid = count2/size(tValid,1)
```

```
count3 = 0;
for i = 1:size(tTest,1)
   if (classify_Test(i,:)~=tTest(i,:))
      count3 = count3+1;
   end
end
C_Test = count3/size(tTest,1)

%C_Train = 0.3920


%C_Valid = 0.4984


%C_Test = 0.4975


%Epoch 368 of 400
```

Convolutional networks 2019

Network 1

```
clear;clc;
%%
[xTrain, tTrain, xValid, tValid, xTest, tTest] = LoadCIFAR(4);

layers = [imageInputLayer([32 32 3])
convolution2dLayer(5, 20, 'Padding',1,'Stride',1)
reluLayer
maxPooling2dLayer(2, 'Padding',0,'Stride',2)
fullyConnectedLayer(50)
reluLayer
fullyConnectedLayer(10)
softmaxLayer
classificationLayer];

options =  trainingOptions('sgdm',...
'MiniBatchSize', 8192,...
'ValidationData', {xValid, tValid},...
'ValidationFrequency', 30,...
'MaxEpochs',120,...
'Plots', 'Training-Progress',...
'L2Regularization', 0, ...
'Momentum', 0.9, ...
'ValidationPatience', 3, ...
'Shuffle', 'every-epoch', ...
'InitialLearnRate', 0.001);

net = trainNetwork(xTrain, tTrain, layers, options);

classify_Train = net.classify(xTrain);
classify_Valid = net.classify(xValid);
classify_Test = net.classify(xTest);

count1 = 0;
for i = 1:size(tTrain,1)
   if (classify_Train(i,:)~=tTrain(i,:))
      count1 = count1+1;
   end
end
C_Train = count1/size(tTrain,1)

count2 = 0;
for i = 1:size(tValid,1)
   if (classify_Valid(i,:)~=tValid(i,:))
      count2 = count2+1;
   end
```

```
end
C_Valid = count2/size(tValid,1)

count3 = 0;
for i = 1:size(tTest,1)
   if (classify_Test(i,:)~=tTest(i,:))
      count3 = count3+1;
   end
end
C_Test = count3/size(tTest,1)

%C_Train = 0.3318

%C_Valid = 0.3978

%C_Test = 0.3992

%epoch 120 of 120
```

Network 2

```
clear;clc;
%%
[xTrain, tTrain, xValid, tValid, xTest, tTest] = LoadCIFAR(4);

layers = [imageInputLayer([32 32 3])
convolution2dLayer(3, 20, 'Padding',1,'Stride',1)
batchNormalizationLayer
reluLayer
maxPooling2dLayer(2, 'Padding',0,'Stride',2)
convolution2dLayer(3, 30, 'Padding',1,'Stride',1)
batchNormalizationLayer
reluLayer
maxPooling2dLayer(2, 'Padding',0,'Stride',2)
convolution2dLayer(3, 50, 'Padding',1,'Stride',1)
batchNormalizationLayer
reluLayer
fullyConnectedLayer(10)
softmaxLayer
classificationLayer];

options =  trainingOptions('sgdm',...
'MiniBatchSize', 8192,...
'ValidationData', {xValid, tValid},...
'ValidationFrequency', 30,...
'MaxEpochs',120,...
'Plots', 'Training-Progress',...
'L2Regularization', 0, ...
'Momentum', 0.9, ...
'ValidationPatience', 3, ...
'Shuffle', 'every-epoch', ...
'InitialLearnRate', 0.001);

net = trainNetwork(xTrain, tTrain, layers, options);

classify_Train = net.classify(xTrain);
classify_Valid = net.classify(xValid);
classify_Test = net.classify(xTest);

count1 = 0;
for i = 1:size(tTrain,1)
   if (classify_Train(i,:)~=tTrain(i,:))
      count1 = count1+1;
   end
end
C_Train = count1/size(tTrain,1)

count2 = 0;
```

```
for i = 1:size(tValid,1)
   if (classify_Valid(i,:)~=tValid(i,:))
      count2 = count2+1;
   end
end
C_Valid = count2/size(tValid,1)

count3 = 0;
for i = 1:size(tTest,1)
   if (classify_Test(i,:)~=tTest(i,:))
      count3 = count3+1;
   end
end
C_Test = count3/size(tTest,1)

%C_Train = 0.2611

%C_Valid = 0.2981

%C_Test = 0.2956

%epoch 120 of 120
```