Assignment 2 Report

ECE 4438B

Prof. Yimin Yang

JD Herlehy || Jacky Li

# Organizing the Dataset

We have chosen to work with the CIFAR10 [1] dataset. The images are packed in six different files. The first five are training and the final sixth is the testing data set. There is also a file that helps convert the label numbers to label words.

# Design of Deep Convolutional Neural Network

### Network 1: 5 Convolution Layers with no maxpooling with descending learning rate

options = trainingOptions('sgdm', ...

'MiniBatchSize', 64, ...

'MaxEpochs',20,...

'InitialLearnRate',1e-4, ...

'LearnRateSchedule','piecewise', ...

'LearnRateDropFactor', 0.2000, ...

'LearnRateDropPeriod', 5, ...

'Verbose',false, ...

'Plots','training-progress');

layers = [

imageInputLayer([32 32 3])

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

batchNormalizationLayer

reluLayer

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

batchNormalizationLayer

reluLayer

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

batchNormalizationLayer

reluLayer

convolution2dLayer(3,64, 'Padding','same')

batchNormalizationLayer

reluLayer

convolution2dLayer(3,128, 'Padding','same')

batchNormalizationLayer

reluLayer

fullyConnectedLayer(10)

softmaxLayer

classificationLayer];

### Network 2:

### 5 Convolution Layers with maxpooling with descending learning rate

%%options

options = trainingOptions('sgdm', ...

'MiniBatchSize', 64, ...

'MaxEpochs',20,...

'InitialLearnRate',1e-4, ...

'LearnRateSchedule','piecewise', ...

'LearnRateDropFactor', 0.2000, ...

'LearnRateDropPeriod', 5, ...

'Verbose',false, ...

'Plots','training-progress');

%%layers

layers = [

imageInputLayer([32 32 3])

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

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,64, 'Padding','same')

batchNormalizationLayer

reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,128, 'Padding','same')

batchNormalizationLayer

reluLayer

fullyConnectedLayer(10)

softmaxLayer

classificationLayer];

### Network 3: 5 Convolution Layers with no maxpooling with constant learning rate

%%options

options = trainingOptions('sgdm', ...

'MiniBatchSize', 64, ...

'MaxEpochs',20,...

'InitialLearnRate',1e-4, ...

'Verbose',false, ...

'Plots','training-progress');

%%layers

layers = [

imageInputLayer([32 32 3])

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

batchNormalizationLayer

reluLayer

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

batchNormalizationLayer

reluLayer

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

batchNormalizationLayer

reluLayer

convolution2dLayer(3,64, 'Padding','same')

batchNormalizationLayer

reluLayer

convolution2dLayer(3,128, 'Padding','same')

batchNormalizationLayer

reluLayer

fullyConnectedLayer(10)

softmaxLayer

classificationLayer];

# Performances

## Network 1:

A screen shot of a computer

Description automatically generatedA close-up of a sign

Description automatically generated

Figure 1 Training Results and Accuracy of Network 1

## Network 2:

A screenshot of a computer

Description automatically generatedA close-up of a number

Description automatically generated

Figure 2 Training Results and Accuracy of Network 2

## Network 3:

A graph showing a blue and red line

Description automatically generated  
A close-up of a number

Description automatically generated

Figure 3 Training Results and Accuracy of Network 3

# Discussion

In Network 1, we chose a network that consists of five convolution layers, with no maxpooling, and with a descending learning rate every 5 epochs. We chose Network 1 as the benchmark for the following network’s performance. For Network 1, we chose to pad the convolution filters to keep the same image size to reduce the amount of information lost between the filters. In Figure 1, we can see that the training accuracy reaches 93.75% with a training time of 6 minutes, but the testing accuracy is significantly lower at 61.45%.

In Network 2, we decided to experiment with maxpooling with Network 1 and its effects on the training and testing accuracy. The result, as shown in Figure 2, is that the training accuracy suffered drastically at 59.38%, along with the testing accuracy at 53.00%. This is most likely attributed to the fact that the maxpooling operation purposely compresses the image to improve training speed, with the trade-off of losing image information. In our case, the training speed improved from 6.0 minutes to 4.5 minutes.

Finally, in Network 3, we removed the descending learning rate from Network 1. The results are very similar to Network 1, as shown in Figure 3. The training accuracy slightly improved to 100.00%, but the testing accuracy remained the same at 61.00%. Switching from a descending learning rate to a constant learning rate showed little impact on the training and testing accuracy. The descending learning rate is used for when the training accuracy plateaus with a constant learning rate. However, in the constant learning rate network, the training accuracy never plateaued below 100%. Finally, keeping a constant learning rate decreased the training time slightly, to 5.75 minutes.

# Dataset

[1] Alex. Kirzhevsky, “Learning Multiple Layers of Features from Tiny Images,” 2009.