Cristopher McIntyre Garcia

MNIST Classification

In this section we explore the differences between soft-max regression, MLP, and CNN classification. We explore how these models perform when dropout and batch normalization are incorporated. Finally we also look at how changing hyper parameters such as depth, width and kernel size affect these models. All throughout the tests, learning rate, epochs, and batch size are all kept consistent.

learning_rate = 0.001
num_epochs = 10
batch_zize = 64

We use Cross Entropy Loss as the loss function and Stochastic Gradient Descent as the optimizer.

First we can compare a basic version of each model. For the MLP, we use a width of 28 and a depth of 2 hidden layer. For the CNN, we use a kernel size of 3 and a depth of 3:

SoftMax_Regression - dropout=False bn=False
Epoch 1: 59.34
Epoch 3: 72.66
Epoch 5: 73.74
Epoch 7: 74.31
Epoch 9: 81.39

MLP - width=28 depth=2 dropout=False bn=False
Epoch 1: 18.46
Epoch 3: 41.89
Epoch 5: 23.88
Epoch 7: 39.22
Epoch 9: 78.19

CNN - k=3 layers=2 dropout=False bn=False
Epoch 1: 81.94
Epoch 3: 84.46
Epoch 5: 84.9
Epoch 7: 84.86
Epoch 9: 85.28

End

We can see that the CNN is out performing the other two models, though not by a lot. This is due to the inductive bias of the model. The convolutions make it possible for the positional properties of the data to be incorporated into the models understanding. The second best model is the soft-max regression model, which might be because the simplicity of the dataset (Only 28 by 28 pixels) is well suited for its simple architecture. In last place is the MLP, this is probably due to the hyper parameters we chose. Changing these hypermetropias will hopefully improve the results significantly.

Next we look at many different configurations of the CNN and MLP models. We start with MLP:

MLP - width=28 depth=2 dropout=False bn=False
Epoch 1: 18.46
Epoch 3: 41.89
Epoch 5: 23.88
Epoch 7: 39.22
Epoch 9: 78.19

MLP - width=28 depth=1 dropout=False bn=False
Epoch 1: 24.67
Epoch 3: 28.99
Epoch 5: 64.31
Epoch 7: 80.89
Epoch 9: 82.88

MLP - width=28 depth=3 dropout=False bn=False
Epoch 1: 10.32
Epoch 3: 25.38
Epoch 5: 27.37
Epoch 7: 27.67
Epoch 9: 26.81

MLP - width=56 depth=1 dropout=False bn=False
Epoch 1: 13.29
Epoch 3: 65.31
Epoch 5: 81.65
Epoch 7: 83.42
Epoch 9: 83.96

MLP - width=56 depth=3 dropout=False bn=False
Epoch 1: 34.82
Epoch 3: 30.07
Epoch 5: 18.29
Epoch 7: 32.76
Epoch 9: 69.33

MLP - width=112 depth=2 dropout=False bn=False
Epoch 1: 11.93
Epoch 3: 66.07
Epoch 5: 82.93
Epoch 7: 84.15
Epoch 9: 84.88

MLP - width=112 depth=1 dropout=False bn=False
Epoch 1: 54.42
Epoch 3: 82.5
Epoch 5: 84.04
Epoch 7: 84.62
Epoch 9: 85.12

End

According to the results above, using a smaller amount of hidden layer makes the MLP perform much better. The results are also much better when a wider configuration is used as well. The widest and shallowest model seems to be performing the best, almost reaching the same performance as the CNN we saw previously. That said, many of the configurations perform very bad, meaning that it is very sensitive to the architecture chosen. Also though some may get close, none of the model configurations were able to surpass the CNN. Next we take a look at different CNN networks:

CNN - k=3 layers=2 dropout=False bn=False
Epoch 1: 81.94
Epoch 3: 84.46
Epoch 5: 84.9
Epoch 7: 84.86
Epoch 9: 85.28

CNN - k=3 layers=1 dropout=False bn=False
Epoch 1: 74.02
Epoch 3: 84.2
Epoch 5: 84.71
Epoch 7: 85.11
Epoch 9: 85.1

CNN - k=3 layers=3 dropout=False bn=False
Epoch 1: 11.35
Epoch 3: 83.65
Epoch 5: 84.47
Epoch 7: 84.76
Epoch 9: 84.72

CNN - k=5 layers=3 dropout=False bn=False
Epoch 1: 72.79
Epoch 3: 75.81
Epoch 5: 76.64
Epoch 7: 77.28
Epoch 9: 78.17

CNN - k=5 layers=1 dropout=False bn=False
Epoch 1: 82.68
Epoch 3: 91.32
Epoch 5: 91.74
Epoch 7: 92.17
Epoch 9: 92.59

CNN - k=7 layers=1 dropout=False bn=False
Epoch 1: 75.16
Epoch 3: 84.29
Epoch 5: 84.55

Epoch 7: 84.9
Epoch 9: 85.07

CNN - k=7 layers=3 dropout=False bn=False
Epoch 1: 22.19
Epoch 3: 73.1
Epoch 5: 75.59
Epoch 7: 76.94
Epoch 9: 92.73

End

The above results show that the CNN benefits from a smaller amount of convolution layers. The kernel size has an odd influence on the network. Looking at the results, when using three convolution layers both kernel sizes of three and seven perform better than a kernel size of five. However a kernel size of five outperforms a kernel size of three and seven when a single convolution layer is used. The best results obtained on this run were by the model with a kernel size of seven, and three convolution layers. In comparison to the other models, the CNN is clearly out performing them with and accuracy over 90 percent.

Next we compare models with different configurations of dropout and batch norm (bn). We will compare the models without any of the these added techniques, with a single added technique, and then with both techniques implemented. When in use, we apply these techniques right before the final full connected layer of each network.

Only one configuration for each model is used for comparisons in this report, the best performer from our previous tests. More results can be found in the results.txt files in the results folder. First we start with the soft-max regression classifier :

SoftMax_Regression - dropout=False bn=False
Epoch 1: 52.94
Epoch 3: 77.9
Epoch 5: 80.57
Epoch 7: 81.45
Epoch 9: 82.1

SoftMax_Regression - dropout=True bn=False
Epoch 1: 67.86
Epoch 3: 72.65
Epoch 5: 80.45
Epoch 7: 81.37
Epoch 9: 81.84

SoftMax_Regression - dropout=False bn=True
Epoch 1: 85.63
Epoch 3: 89.43
Epoch 5: 90.06
Epoch 7: 90.56
Epoch 9: 90.98

SoftMax_Regression - dropout=True bn=True
Epoch 1: 83.86
Epoch 3: 88.68
Epoch 5: 89.66
Epoch 7: 90.15
Epoch 9: 90.43

End

The above results show that adding dropout to the soft-max regression model lowers its accuracy. On the other hand, the addition of batch normalization improves the accuracy of the model significantly. With batch normalization, the soft-max regression classifier is able to out perform certain configuration of CNN. When both batch normalization and dropout are in use, the accuracy diminishes from when only batch normalization is used. This makes sense, because drop out is removing random nodes form the network. The network in this case has no hidden layers, and so is negatively impacted when nodes are removed. Next we look at MLP:

MLP - width=112 depth=1 dropout=False bn=False
Epoch 1: 54.42
Epoch 3: 82.5
Epoch 5: 84.04
Epoch 7: 84.62
Epoch 9: 85.12

MLP - width=112 depth=1 dropout=True bn=False
Epoch 1: 51.18
Epoch 3: 82.28
Epoch 5: 84.1
Epoch 7: 84.57
Epoch 9: 84.95

MLP - width=112 depth=1 dropout=False bn=True
Epoch 1: 96.43
Epoch 3: 97.61
Epoch 5: 98.07
Epoch 7: 98.16
Epoch 9: 98.36

MLP - width=112 depth=1 dropout=True bn=True
Epoch 1: 96.1
Epoch 3: 97.62
Epoch 5: 98.09
Epoch 7: 98.18
Epoch 9: 98.33

End

The above results show that including dropout reduces the results seen originally. The difference is not very significant however, and different results might be seen in a different run. Also because we are analyzing a model with a depth of one, this could be the reason why the results worsened. And so we take a look at the results of a network with multiple layers bellow:

MLP - width=112 depth=2 dropout=False bn=False
Epoch 1: 11.93
Epoch 3: 66.07
Epoch 5: 82.93
Epoch 7: 84.15
Epoch 9: 84.88

MLP - width=112 depth=2 dropout=True bn=False
Epoch 1: 9.8
Epoch 3: 66.61
Epoch 5: 82.93
Epoch 7: 84.16
Epoch 9: 84.89

Here we see that the model performs essential the same. These results were all found with dropout on the final layer, and so if we add it to a hidden layer maybe the results would be different. We explore this idea bellow:

MLP - width=28 depth=3 dropout=False bn=False
Epoch 1: 10.32
Epoch 3: 25.38
Epoch 5: 27.37
Epoch 7: 27.67
Epoch 9: 26.81

MLP - width=56 depth=3 dropout=False bn=False
Epoch 1: 34.82
Epoch 3: 30.07
Epoch 5: 18.29
Epoch 7: 32.76
Epoch 9: 69.33

MLP - width=28 depth=3 dropout=True bn=False
Epoch 1: 17.08
Epoch 3: 13.25
Epoch 5: 14.51
Epoch 7: 18.68
Epoch 9: 21.62

MLP - width=56 depth=3 dropout=True bn=False
Epoch 1: 19.03
Epoch 3: 18.65
Epoch 5: 11.33
Epoch 7: 14.28

Epoch 9: 32.46

End

Sadly, dropout still seems to perform poorly even when used in the central hidden layer. On the other hand, batch normalization has a positive affect on the accuracy of the model. We see when batch normalization is implemented, the results increase significantly, reaching almost 99 percent. These results are better than any that have been presented so far. Also, using a combination of batch normalization and dropout also provide good results, performing better than when batch normalization is used alone. This is odd because it is the opposite of what we saw when dropout was used alone.

Next we look at different configurations with the CNN:

CNN - k=7 layers=3 dropout=False bn=False
Epoch 1: 22.19
Epoch 3: 73.1
Epoch 5: 75.59
Epoch 7: 76.94
Epoch 9: 92.73

CNN - k=7 layers=3 dropout=True bn=False
Epoch 1: 10.57
Epoch 3: 24.04
Epoch 5: 80.16
Epoch 7: 91.38
Epoch 9: 93.87

CNN - k=7 layers=3 dropout=True bn=False (**Used in the conv layers**)
Epoch 1: 61.5
Epoch 3: 75.52
Epoch 5: 76.97
Epoch 7: 86.24
Epoch 9: 87.13

CNN - k=7 layers=3 dropout=False bn=True
Epoch 1: 95.26
Epoch 3: 97.34
Epoch 5: 97.92
Epoch 7: 98.11
Epoch 9: 98.28

CNN - k=7 layers=3 dropout=True bn=True
Epoch 1: 93.36
Epoch 3: 96.44
Epoch 5: 97.56
Epoch 7: 98.23
Epoch 9: 98.23

End

With the results above, we see definite benefits in implementing both dropout and batch normalization into the CNN. Dropout seems to impact the accuracy less than bath normalization, however they both have a positive impact. It also works better on the final layer than in the convolution layers. When they are used in combination, we see the best results out of all. That said, these results are not better than what we saw with the best MLP seen so far. Because of this I decided to find all the best results for each model architecture to compare their best configurations.

We have already seen the best Soft-Max regression model. It was the model that used dropout without batch normalization to obtain a final accuracy of 90.98.

Next we look at the best MLPs:

Best without dropout or batch normalization
MLP - width=112 depth=1 dropout=False bn=False
Epoch 1: 54.42
Epoch 3: 82.5
Epoch 5: 84.04
Epoch 7: 84.62
Epoch 9: 85.12

Best with dropout
MLP - width=112 depth=1 dropout=True bn=False
Epoch 1: 51.18
Epoch 3: 82.28
Epoch 5: 84.1
Epoch 7: 84.57
Epoch 9: 84.95

Best with batch normalization
MLP - width=112 depth=2 dropout=False bn=True
Epoch 1: 96.58
Epoch 3: 97.91
Epoch 5: 98.11
Epoch 7: 98.29
Epoch 9: 98.43

Best with dropout & batch normalization
MLP - width=112 depth=2 dropout=True bn=True
Epoch 1: 96.56
Epoch 3: 97.81
Epoch 5: 98.19
Epoch 7: 98.42
Epoch 9: 98.41

End

We see the best performer with an MLP architecture is the one using a width of 112 with a depth of 2, without dropout, and batch normalization enabled. The is the best performing MLP up to date, and beats any CNN we have seen so far.

Now we look at the best CNN configurations:

Best without dropout or batch normalization
CNN - k=7 layers=3 dropout=False bn=False
Epoch 1: 22.19
Epoch 3: 73.1
Epoch 5: 75.59
Epoch 7: 76.94
Epoch 9: 92.73

Best with dropout
CNN - k=7 layers=3 dropout=True bn=False
Epoch 1: 10.57
Epoch 3: 24.04
Epoch 5: 80.16
Epoch 7: 91.38
Epoch 9: 93.87

Best with batch normalization
CNN - k=5 layers=3 dropout=False bn=True
Epoch 1: 95.39
Epoch 3: 97.71
Epoch 5: 98.39
Epoch 7: 98.56
Epoch 9: 98.75

Best with dropout & batch normalization
CNN - k=5 layers=3 dropout=True bn=True
Epoch 1: 94.16
Epoch 3: 97.25
Epoch 5: 97.93
Epoch 7: 98.31
Epoch 9: 98.53

End

It is interesting to see how different configuration of the CNN and MLP affect the accuracy of the models. As seen above, the best CNNs always outperform the best MLP regardless of the configuration. More layers is desirable when using CNNs, however this is not the case when using MLP. Also, using a larger kernel size seems to provide better results. All the best MLPs used the largest width experimented with. Finally, the CNN and MLP with the best performance only used batch normalization without using dropout.