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: 29.92
Epoch 3: 42.47
Epoch 5: 20.55
Epoch 7: 52.05
Epoch 9: 71.81

CNN - k=3 layers=2 dropout=False bn=False
Epoch 1: 47.35
Epoch 3: 84.58
Epoch 5: 85.09
Epoch 7: 85.45
Epoch 9: 85.71

End

We can see that the CNN is out performing the other two models, though not by a lot. The second best model is the soft-max regression model, which might be because of the simplicity of the dataset (Only 28 by 28 pixels). In last place is the MLP, this is probably due to the hyper parameters being used and changing them might 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: 18.83
Epoch 3: 51.68
Epoch 5: 27.5

Epoch 7: 48.79
Epoch 9: 69.01

MLP - width=28 depth=1 dropout=False bn=False
Epoch 1: 45.48
Epoch 3: 35.0
Epoch 5: 67.27
Epoch 7: 81.35
Epoch 9: 83.01

MLP - width=28 depth=3 dropout=False bn=False
Epoch 1: 10.28
Epoch 3: 10.28
Epoch 5: 11.11
Epoch 7: 28.31
Epoch 9: 30.36

MLP - width=56 depth=1 dropout=False bn=False
Epoch 1: 33.39
Epoch 3: 64.65
Epoch 5: 74.51
Epoch 7: 82.96
Epoch 9: 83.79

MLP - width=56 depth=3 dropout=False bn=False
Epoch 1: 17.02
Epoch 3: 29.7
Epoch 5: 23.63
Epoch 7: 26.73
Epoch 9: 59.3

MLP - width=112 depth=2 dropout=False bn=False
Epoch 1: 12.75
Epoch 3: 66.43
Epoch 5: 82.81
Epoch 7: 84.28
Epoch 9: 84.83

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. Some MLP configurations out perform that of the soft -max regression classifier. However, the CNN still seems to be performing better than any of these configurations, Next we take a look at different CNN networks:

CNN - k=3 layers=2 dropout=False bn=False
Epoch 1: 83.29
Epoch 3: 84.67
Epoch 5: 85.13

Epoch 7: 85.15
Epoch 9: 85.21

CNN - k=3 layers=1 dropout=False bn=False
Epoch 1: 87.39
Epoch 3: 91.02
Epoch 5: 91.81
Epoch 7: 92.01
Epoch 9: 92.01

CNN - k=3 layers=3 dropout=False bn=False
Epoch 1: 82.47
Epoch 3: 84.49
Epoch 5: 84.9
Epoch 7: 84.82
Epoch 9: 84.84

CNN - k=5 layers=3 dropout=False bn=False
Epoch 1: 32.41
Epoch 3: 82.88
Epoch 5: 83.68
Epoch 7: 84.35
Epoch 9: 85.82

CNN - k=5 layers=1 dropout=False bn=False
Epoch 1: 82.12
Epoch 3: 84.46
Epoch 5: 91.52
Epoch 7: 91.82
Epoch 9: 92.28

CNN - k=7 layers=1 dropout=False bn=False
Epoch 1: 82.45
Epoch 3: 84.09
Epoch 5: 84.64
Epoch 7: 84.8
Epoch 9: 91.76

End

The above results show that the CNN benefits from a smaller amount of layers. The kernel size does not have a large impact on the accuracy of the CNN, sometimes a smaller kernel size is better, and sometimes worse. The results are very good, in comparison to the other models, sometimes reaching over 90 percent accuracy.

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. Only one configuration for each model is used, the best performing from our previous tests. First we start with the soft-max regression classifier :

SoftMax_Regression - dropout=False bn=False
Epoch 1: 61.11
Epoch 3: 76.62
Epoch 5: 80.59
Epoch 7: 81.5
Epoch 9: 81.94

SoftMax_Regression - dropout=True bn=False
Epoch 1: 53.76
Epoch 3: 78.51
Epoch 5: 80.53
Epoch 7: 81.34
Epoch 9: 81.84

SoftMax_Regression - dropout=False bn=True
Epoch 1: 85.51
Epoch 3: 89.54
Epoch 5: 90.26
Epoch 7: 90.89
Epoch 9: 90.85

SoftMax_Regression - dropout=True bn=True
Epoch 1: 84.01
Epoch 3: 88.78
Epoch 5: 89.8
Epoch 7: 90.14
Epoch 9: 90.54

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. Next we look at MLP:

MLP - width=112 depth=2 dropout=False bn=False
Epoch 1: 9.8
Epoch 3: 71.79
Epoch 5: 83.37
Epoch 7: 84.39
Epoch 9: 84.93

MLP - width=112 depth=2 dropout=True bn=False
Epoch 1: 9.81
Epoch 3: 71.81
Epoch 5: 83.41
Epoch 7: 84.44
Epoch 9: 85.09

MLP - width=112 depth=2 dropout=False bn=True
Epoch 1: 10.46
Epoch 3: 71.41
Epoch 5: 82.98
Epoch 7: 84.38
Epoch 9: 84.86

MLP - width=112 depth=2 dropout=True bn=True
Epoch 1: 9.8
Epoch 3: 68.35
Epoch 5: 83.23
Epoch 7: 84.23
Epoch 9: 84.97

End

The above results show that including dropout and batch normalization, in any configuration, does not drastically improve the results of the MLP. Dropout does seem to improve the results, while batch normalization seems to lower them a little. This is in direct contrast to the results seen with the soft-max regression model, leading me to believe that a deeper model will benefit more from dropout. Next we look at different configurations with the CNN:

CNN - k=5 layers=1 dropout=False bn=False
Epoch 1: 82.81
Epoch 3: 84.48
Epoch 5: 84.63
Epoch 7: 84.93
Epoch 9: 91.65

CNN - k=5 layers=1 dropout=True bn=False
Epoch 1: 93.82
Epoch 3: 95.88
Epoch 5: 96.75
Epoch 7: 97.17
Epoch 9: 97.62

CNN - k=5 layers=1 dropout=False bn=True
Epoch 1: 82.45
Epoch 3: 90.74
Epoch 5: 91.45
Epoch 7: 91.84
Epoch 9: 92.02

CNN - k=5 layers=1 dropout=True bn=True
Epoch 1: 94.6
Epoch 3: 96.51
Epoch 5: 97.15
Epoch 7: 97.6

Epoch 9: 97.83

End

With the results above, we see definite benefits in implementing both dropout and batch normalization into the CNN. Drop out seems impact the accuracy more than bath normalization, however they both impact it positively. When they are used in combination, we see the best results out of all the models.

For curiosity, I ran tests on the CNN model to find the best configurations, here are some of the results of the best performers.

Best without dropout or batch normalization
CNN - k=7 layers=1 dropout=False bn=False
Epoch 1: 82.53
Epoch 3: 84.47
Epoch 5: 86.55
Epoch 7: 91.85
Epoch 9: 92.29

Best with dropout
CNN - k=3 layers=3 dropout=True bn=False
Epoch 1: 94.28
Epoch 3: 96.55
Epoch 5: 97.42
Epoch 7: 97.73
Epoch 9: 97.87

Best with batch normalization
CNN - k=5 layers=1 dropout=False bn=True
Epoch 1: 74.77
Epoch 3: 90.86
Epoch 5: 91.59
Epoch 7: 91.81
Epoch 9: 92.05

CNN - k=7 layers=3 dropout=True bn=True
Epoch 1: 94.69
Epoch 3: 97.4
Epoch 5: 97.99
Epoch 7: 98.05
Epoch 9: 98.41

It is interesting to see how different configuration of the CNN will be affected differently from the addition of dropout and/or batch normalization. From what I see, having a larger kernel and more layers is better when using dropout and batch normalization, but may not be the case when using only batch normalization. When using only dropout, having a smaller kernel seems to work best, with a larger amount of layers. Finally neither dropout nor batch normalization are used, it is good to use a large kernel with a smaller amount of layers.