Assignment 4 Report:

Done By : Ashwath Raghav

Course: IST 597 - Deep Learning

Github Link:

https://github.com/ashwathraghav/IST597-Assignment4.git

Assignment 4

- Ashwath Raghav

Introduction:

In this assignment, we were asked to explore using Batch Normalization as a Regularization Technique. I employed Convolutional Neural Network and Multi Layer Perceptron models to test this regularization technique. The models were trained with Fashion-MNIST. Experiments were conducted to observe how Batch Normalization behaves during Training and Testing. In Training, Batch Normalization takes into account the batch's mean and variance whereas in testing, Batch Normalization takes into account the entire population's mean and variance to find the scaled value of x or scaled value of g(x).

We will look at how Batch Normalization behaves differently when it is performed before performing Activation function on our inputs and after performing Activation function on our inputs.

From previous experiments by other scientists, it was noted that, BN may be more appropriate after the activation function if for s-shaped functions like the hyperbolic tangent and logistic function.

Scientists also noted that, BN may be appropriate before the activation function for activations that may result in non-Gaussian distributions like the rectified linear activation function, the modern default for most network types. Graphs comparing the test accuracy of the different models are compared.

Core Experiment 1 - CNN:

Fashion-Mnist dataset comprises 70,000 [28 x 28] pictures of items like shirts, shoes,etc.

Data Preprocessing/Feature Engineering:

1. The dataset Fashion-MNIST is retrieved using fetch_openml. Since the dataset is in the form of pixels, we need to convert it into float data and the categorical labels should be converted to one-hot vectors. We use to_categorical() for this.

2. Since we have 28 x 28 pixels, each picture will be of size 784. Thus, we need to initialise the size of the input layer as 784.

3. There are 10 output labels. Thus, we need to initialise the size of the output layer as 10.

Experiments performed on this dataset:

1. Perform Batch Normalization after performing Activation Function(ReLU) [Post Activation]:

Assignment 4

- Ashwath Raghav

In this experiment, we perform BN(x), then pass these normalized inputs to our ReLU. We run this experiment thrice to check how the validation and test accuracy changes. The results are present in the following screenshots:

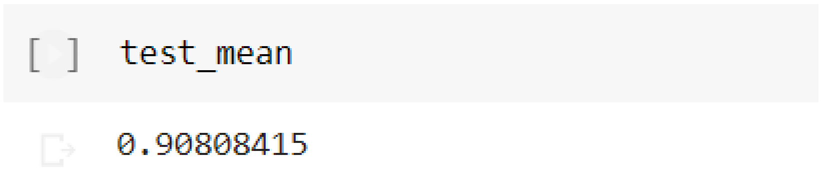
First Run:

```
Train Acc at epoch 13:  0.90880996
Eval loss at epoch 13:  0.24392311
Eval Acc at epoch 13:  0.91035426
```

The model stops converging at Epoch 13. [Early Stopping comes into play here]

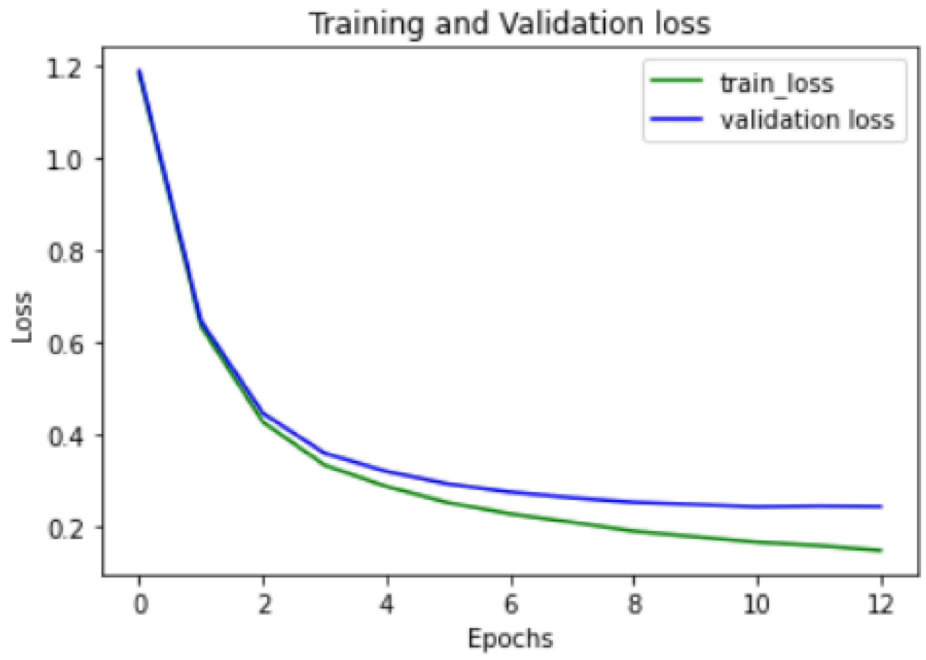
Final Validation Acc: 0.9103

Test Acc:



```
[ ]  test_mean

     0.90808415
```



Assignment 4

- Ashwath Raghav

Second Run:

```
[illegible]
Train Acc at epoch 15:  0.91403466
Eval loss at epoch 15:  0.23799595
Eval Acc at epoch 15:  0.9153332
```

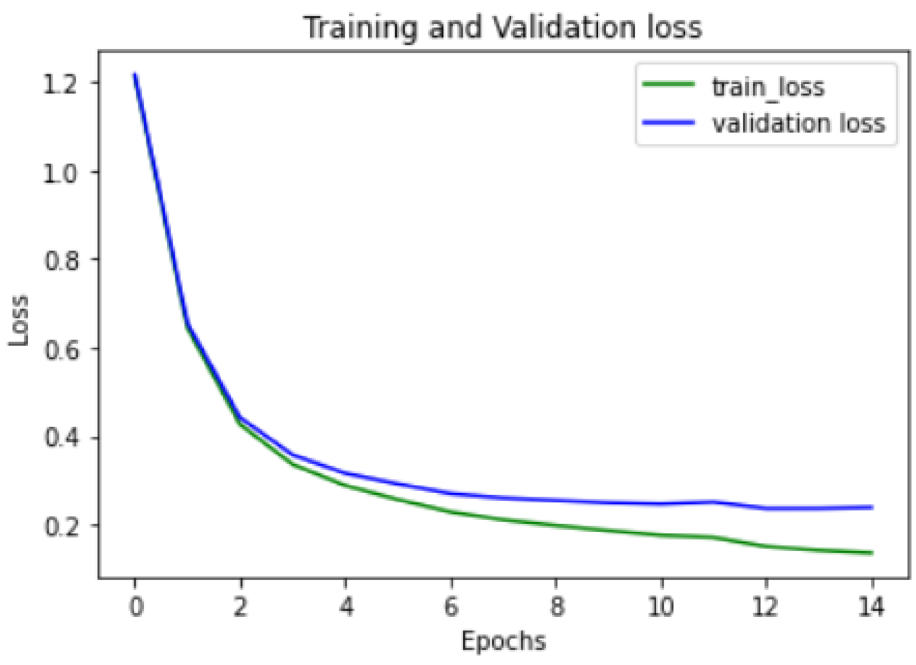
The model stops converging at Epoch 15. [Early Stopping comes into play here]

Final Validation Acc: 0.9155

Test Acc:

```
test_mean
```

```
0.9130197
```



Third Run:

```
Train loss at epoch 13:  0.13962936
Train Acc at epoch 13:  0.92237335
Eval loss at epoch 13:  0.24039266
Eval Acc at epoch 13:  0.92341673
```

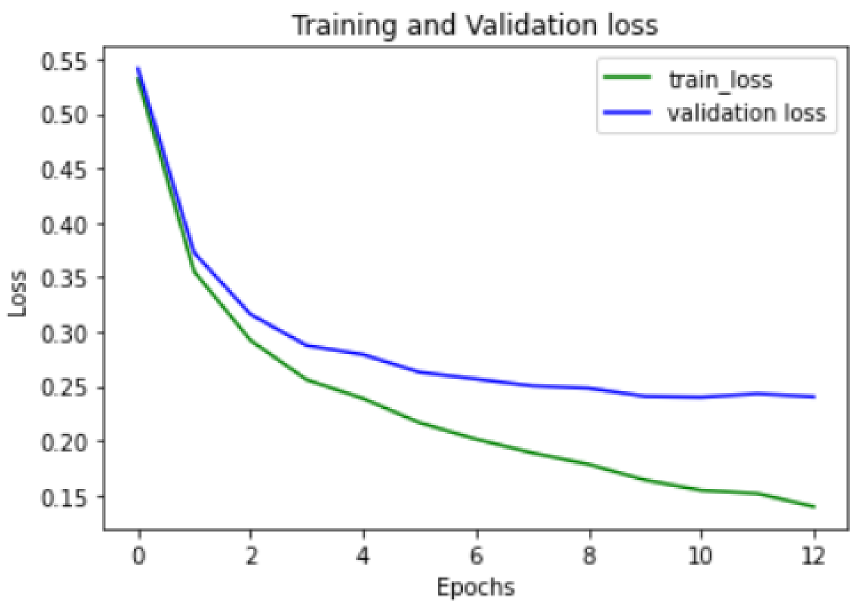
The model stops converging at Epoch 13. [Early Stopping comes into play here]

Final Validation Acc: 0.9234

Test Acc:

```
[ ]  test_mean

     0.9205788
```



2. Perform Batch Normalization before performing Activation Function(ReLU) [Pre Activation]:

   In this experiment, we perform Activation Function ReLU, ie., g(x), then pass g(x) to BN, which will give us BN(g(x)), then pass these as inputs to our next layer. We run this experiment thrice to check how the validation and test accuracy changes. The results are present in the following screenshots:

Assignment 4

- Ashwath Raghav

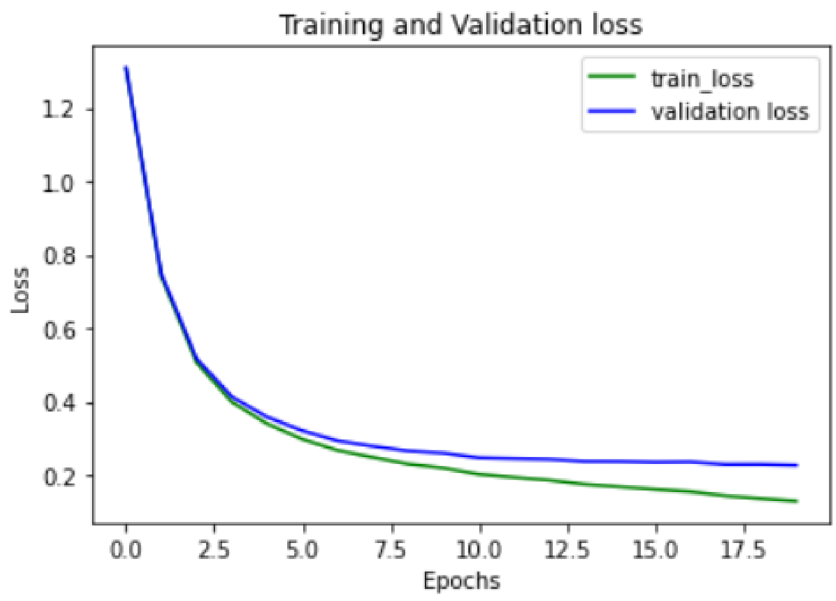
First Run:

```
Train loss at epoch 20:  0.12889472
Train Acc at epoch 20:  0.9117026
Eval loss at epoch 20:  0.2263381
Eval Acc at epoch 20:  0.91277593
```

Test Acc:

```
[ ]  test_mean

     0.9110058
```



Second Run:

```
Train loss at epoch 20:  0.12196393
Train Acc at epoch 20:  0.91336936
Eval loss at epoch 20:  0.22172657
Eval Acc at epoch 20:  0.91444916
```

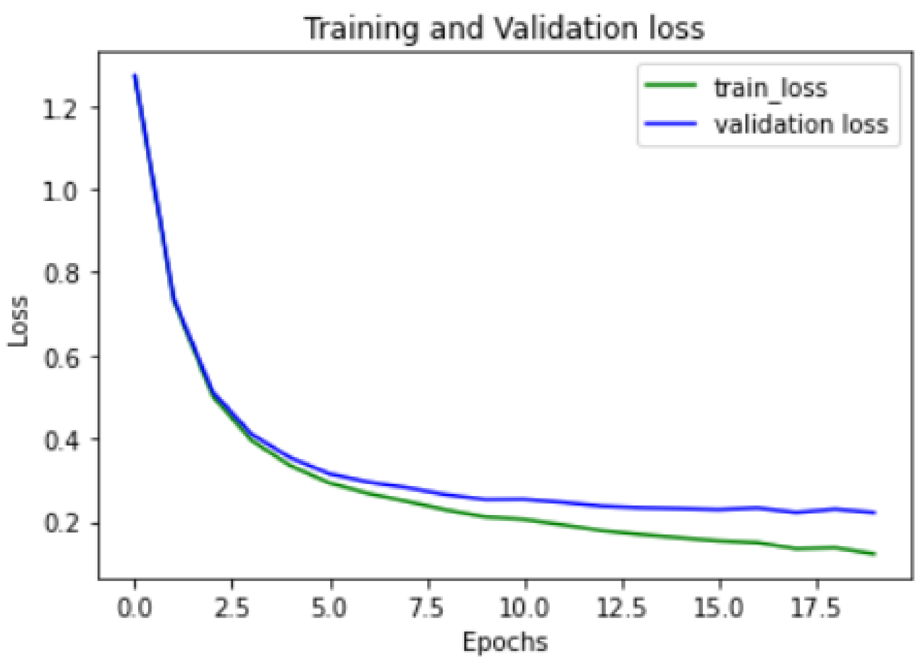
Test Acc:

Assignment 4

- Ashwath Raghav

```
[ ]  test_mean

     0.91278327
```



Third Run:

```
Train loss at epoch 20:  0.1093617
Train Acc at epoch 20:  0.92981756
Eval loss at epoch 20:  0.23703425
Eval Acc at epoch 20:  0.9305446
```

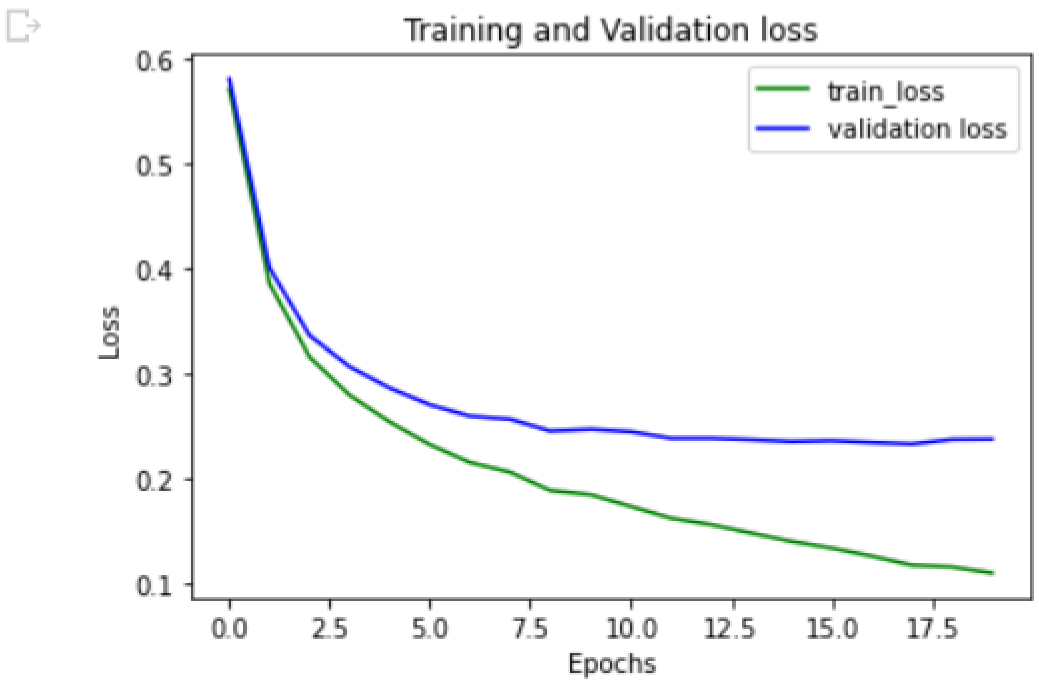
Test Acc:

```
[ ]  test_mean

     0.9288417
```

Assignment 4

- Ashwath Raghav



Core Experiment 2 - MLP:

Fashion-Mnist dataset comprises 70,000 [28 x 28] pictures of items like shirts, shoes,etc.

Data Preprocessing/Feature Engineering:

1. The dataset Fashion-MNIST is retrieved using fetch_openml. Since the dataset is in the form of pixels, we need to convert it into float data and the categorical labels should be converted to one-hot vectors. We use to_categorical() for this.

2. Since we have 28 x 28 pixels, each picture will be of size 784. Thus, we need to initialise the size of the input layer as 784.

3. There are 10 output labels. Thus, we need to initialise the size of the output layer as 10.

Experiments performed on this dataset:

1. Perform Batch Normalization after performing Activation Function(ReLU) [Post Activation]:

Assignment 4

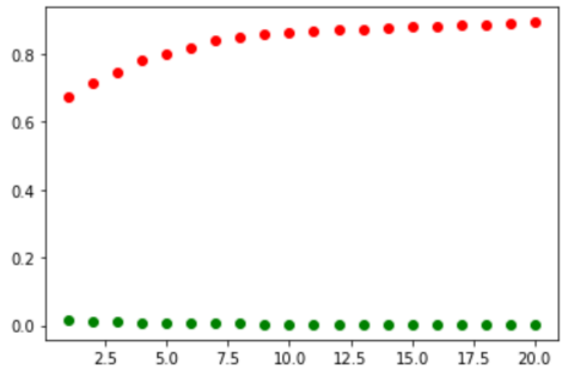
- Ashwath Raghav

In this experiment, we perform BN(x), then pass these normalized inputs to our ReLU. We run this experiment thrice to check how the validation and test accuracy changes. The results are present in the following screenshots:

After 3 runs, final Train, Validation and Test Accuracy:

```
Train Accuracy: 0.8942
Number of Epoch = 20 - Average Cross Entropy:= 0.002807323913574219

Validation Accuracy: 0.8718
```



```
Test Accuracy = 0.8619

Total time taken (in seconds): 266.19
Test Accuracy [0.862, 0.8677, 0.8619]
the mean of 3 runs is 0.8638666666666667, and the variance is 7.348888888888889725e-06
```

2. Perform Batch Normalization before performing Activation Function(ReLU) [Pre Activation]:

In this experiment, we perform Activation Function ReLU, ie., g(x), then pass g(x) to BN, which will give us BN(g(x)), then pass these as inputs to our next layer. We run this experiment thrice to check how the validation and test accuracy changes. The results are present in the following screenshots:

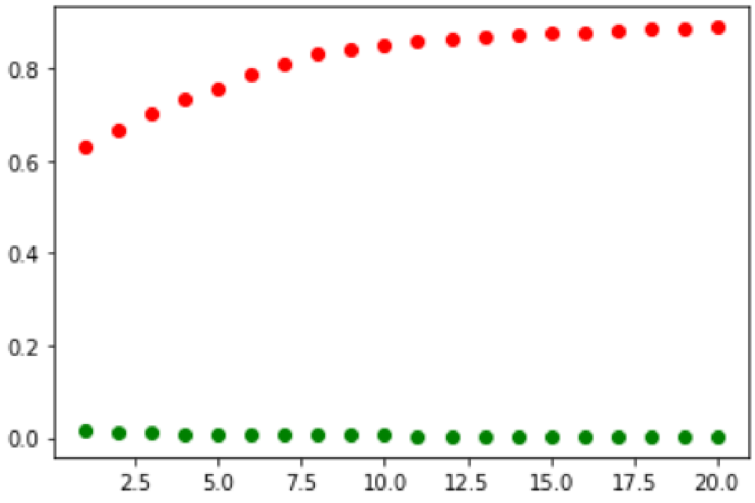
After 3 runs, final Train, Validation and Test Accuracy:

Assignment 4

- Ashwath Raghav

```
Train Accuracy: 0.8891
Number of Epoch = 20 - Average Cross Entropy:= 0.0032840737915039065

Validation Accuracy: 0.8693
```



```
Test Accuracy = 0.8648

Total time taken (in seconds): 268.76
Test Accuracy [0.8625, 0.8638, 0.8648]
the mean of 3 runs is 0.8637, and the variance is 8.866666666666416e-07
```

Conclusion:

From all these experiments, we can observe that the test accuracy when we use CNN was far better than when we use the MLP model. We can also observe that Pre Activation works better in most of our cases than Post Activation.