Fashion MNIST Basic Neural Network Architectures **Danial Zohourian**

In this report I will be showcasing the procedure I took to run different networks I ran on the Fashion MNIST dataset. First off, the hyperparameters used are as follows:

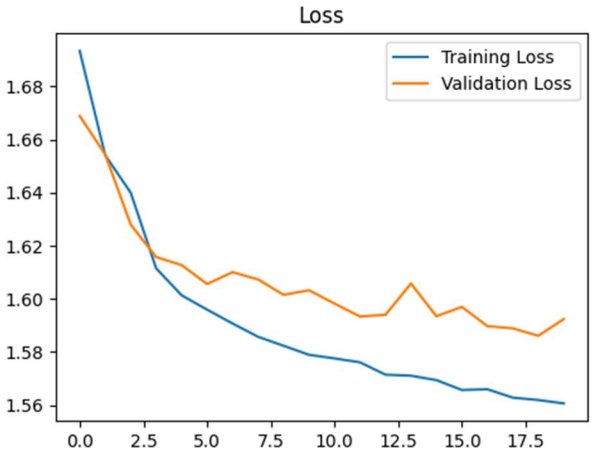
```
# Hyperparameters
input_size = 28 * 28
num_classes = 10
learning_rate = 0.001
batch_size = 64
num_epochs = 20
dropout_prob = 0.5
l1_strength = 0.001
l2_strength = 0.001
```

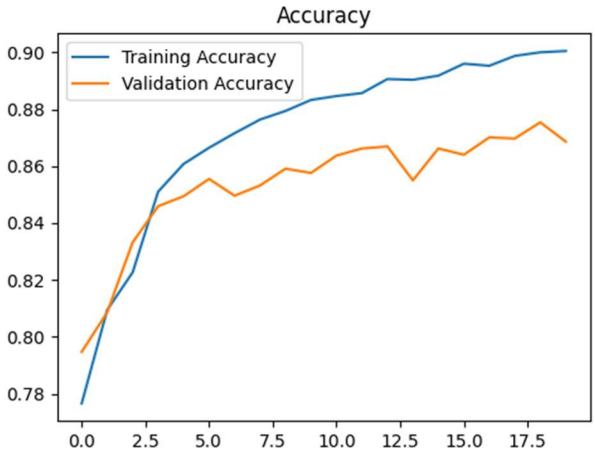
Also, for each layer, the activation function is ReLU, except for the last layer, which is softmax.

Network #1

Hidden Layer 1: 128 neurons

This is a very simple perceptron with just a single layer. The train/val loss and accuracy scores are shown below:





The validation accuracy reaches close to 87 percent and fluctuates onward.

Network #2

Hidden Layer 1: 128 neurons

Hidden Layer 2: 64 neurons

In the second network, I added a new hidden layer with 64 neurons. The train/val loss and accuracy scores are shown below:

As we can see, the performance of the model didn't improve much. This can be of the model being too complex. We can also see in both networks that the trends of both validation and train loss and accuracies are somewhat the same, so with more epochs, the validation scores may improve, but the gap between them may suggest overfitting.

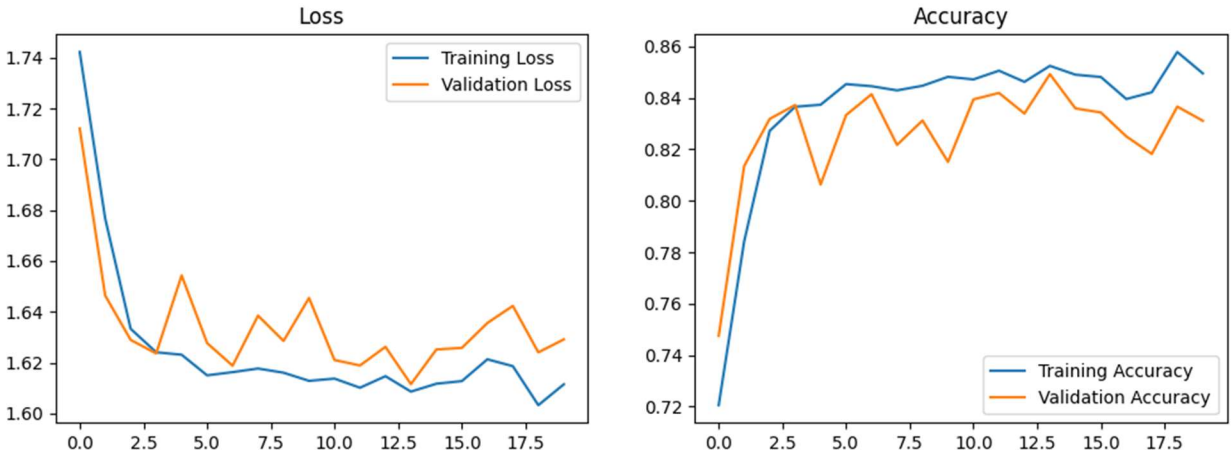
Network #3

Hidden Layer 1: 512 neurons

Hidden Layer 2: 128 neurons

Hidden Layer 3: 64 neurons

In this network, I added a somewhat large hidden layer at the beginning. The train/val loss and accuracy scores are shown in the next page.



As we can see, the gap between the plots have plummeted but the overall validation scores have dropped. This can be of the model complexity being too high. We can see that there was no improvement after the 6th epoch on validation scores.

By analyzing the last three networks, we can see that we need to use techniques like dropout, L1/L2 regularization to prevent the model from becoming too complex.

Network #4

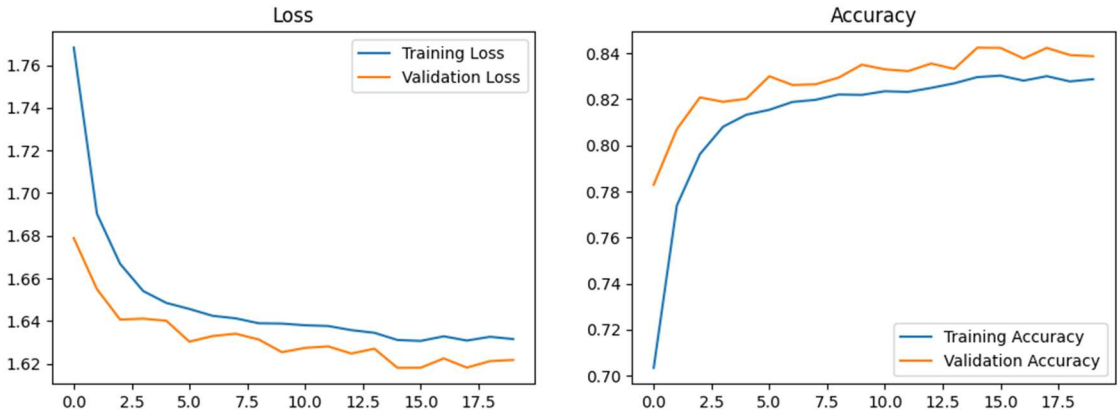
Hidden Layer 1: 128 neurons

0.5 Probability Dropout

Hidden Layer 2: 64 neurons

0.5 Probability Dropout

In the next network we will be adding a 50% dropout to network #2. The train/val loss and accuracy scores are shown below:



We can see that the validation scores are constantly higher than the train's. If there is randomness in the training process, like using dropout, it could lead to variations in training accuracy. Running multiple training sessions and observing the average behavior might provide more insights. We can also decrease the dropout probability.

Network #5

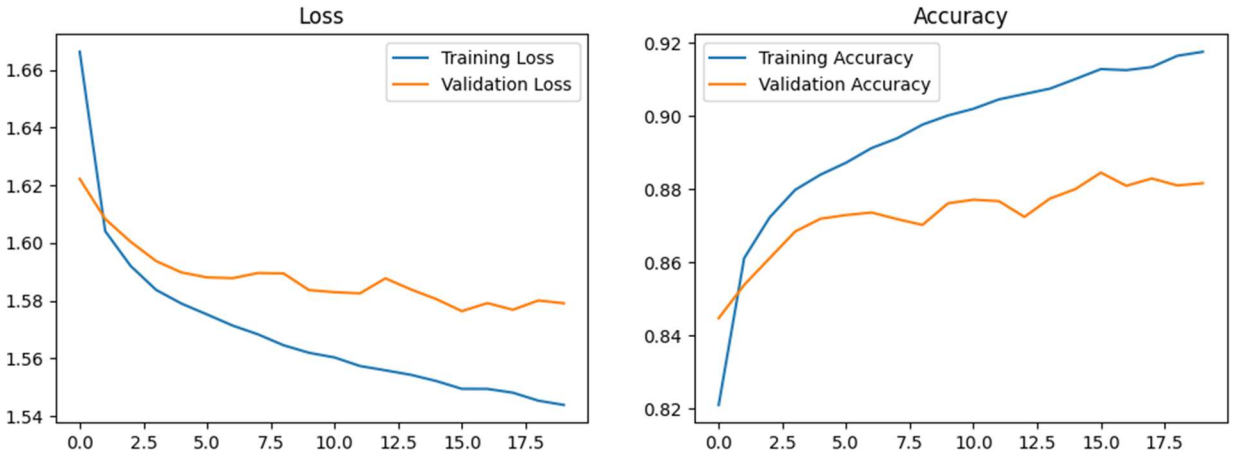
Hidden Layer 1: 128 neurons

Batch Normalization

Hidden Layer 2: 64 neurons

Batch Normalization

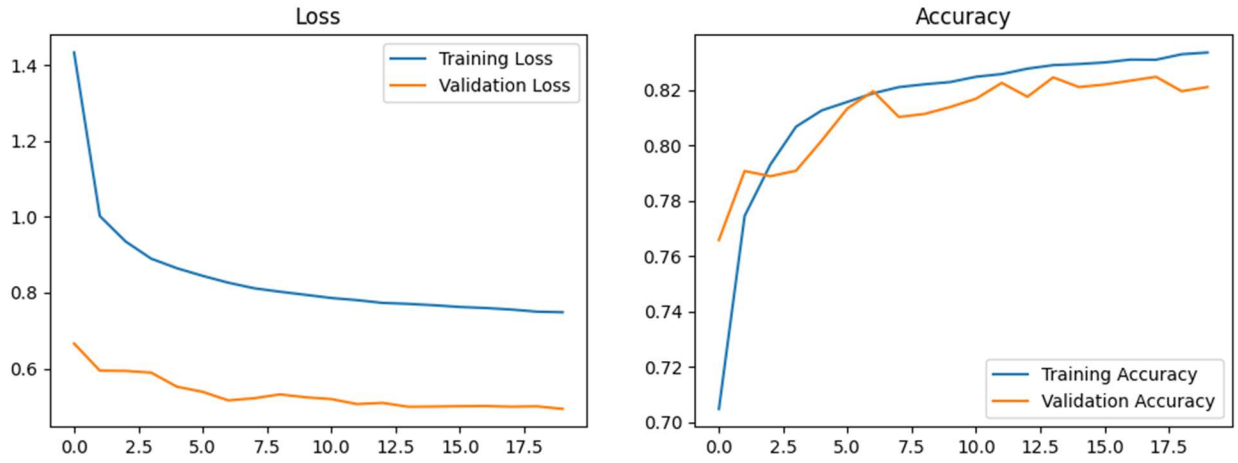
In this network we use batch normalization instead of dropout. The train/val loss and accuracy scores are shown below:



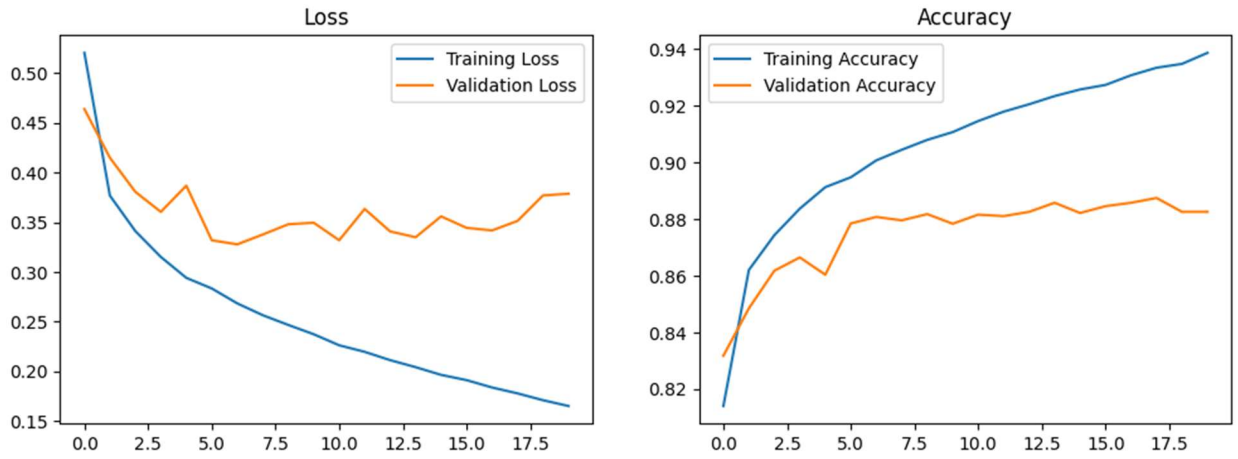
In this network, we have the highest validation accuracy (0.88) and lowest validation loss (1.54).

Network #6 and #7

This is the same as network #2, but with L1 or L2 regularization. The train/val loss and accuracy scores for the network with L1 regularization are shown below:



The train/val loss and accuracy scores for the network with L2 regularization are shown below:



We can see that the L1 regularization does not perform very well.

When working with image data, such as the Fashion MNIST dataset, L2 regularization is often preferred over L1 regularization. Here are a few reasons:

In image data, pixels are typically continuous values, and L2 regularization is well-suited for continuous variables. It penalizes large weights but doesn't encourage exactly zero weights. This can be beneficial when dealing with the smooth gradients in pixel values.

L2 regularization is less likely to lead to sparse solutions (weights set exactly to zero), which can be important in image processing tasks where each pixel may contribute to the overall understanding of the image.

L2 regularization tends to produce smoother weight solutions compared to L1 regularization. In image-related tasks, this smoothness can be advantageous.

Network #8

Convolution 1: 32 filters, kernel size 3x3, 1 stride, 1 padding

Convolution 2: 32 filters, kernel size 3x3, 1 stride, 1 padding

Hidden Layer 1: 128 neurons

Hidden Layer 2: 64 neurons

The train/val loss and accuracy scores are shown below:

We can see that in the convolutional network, the scores are much better. We can further increase the score by using batch normalization and dropout. We can also fine-tune other hyper parameters to see the effects of each one.