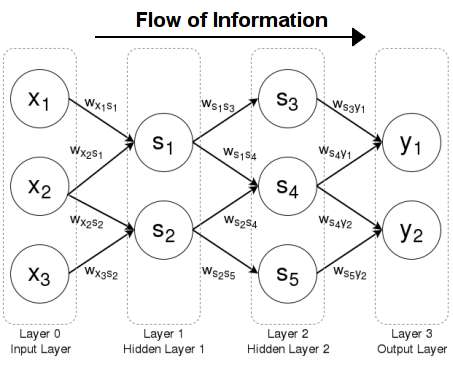
**Assignment 2**

Subrat Kumar Swaim (2021QIZ8247)

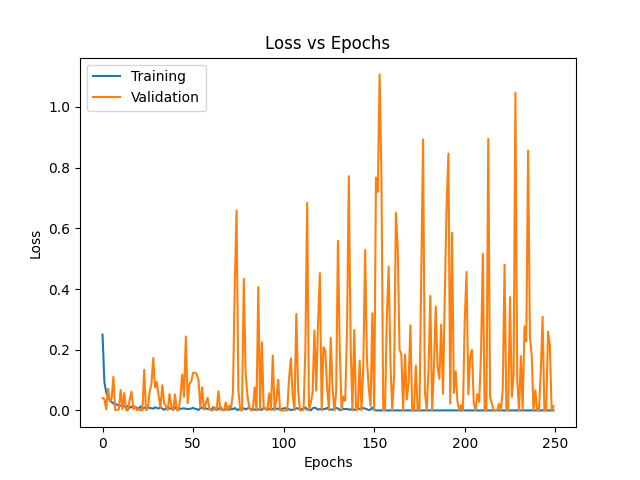
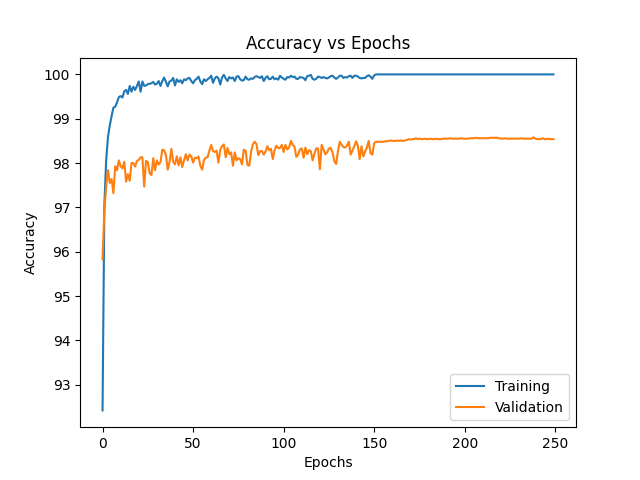
Chandan (2021EEZ8527)

**Problem 1 (a):**

In this assignment, we have used PyTorch Framework for data training models and data validations. As suggested in the assignment, we are using the Feed Forward neural network which is also known as the Artificial Neural network where the connections between units do not form a cycle. Feedforward neural networks were the first type of artificial neural network invented and are simpler than their counterpart, recurrent neural networks. They are called *feedforward* because information only travels forward in the network (no loops), first through the input nodes, then through the hidden nodes (in our case, we have 2 hidden layers), and finally through the output nodes. The information/features of text in images flows from input layer to first hidden layer and then to second layer and finally goes to output layer.



In this training model, we are using Sequential classifier and Adam optimizer as it requires less memory and is efficient in large datasets and a lot of parameters present in the data sets. First, we have downloaded MNIST Raw datasets and then created a python file to load data in our scripts, then created a python file named mnist.py in which we have written all the code to train the model based on the data loaded from dataloader.py file, we have used batch size of 64, epochs size is 250, we are using cross-entropy loss in this algorithm.We are using the cross entropy loss built-in function of the pytorch module, as it is used where more than two classes have been used, like in our example, we have classification of digits (0-9). After training and checking the losses, accuracy on training data and validation/test data sets and being able to compute these losses. The following graphs are attached below.

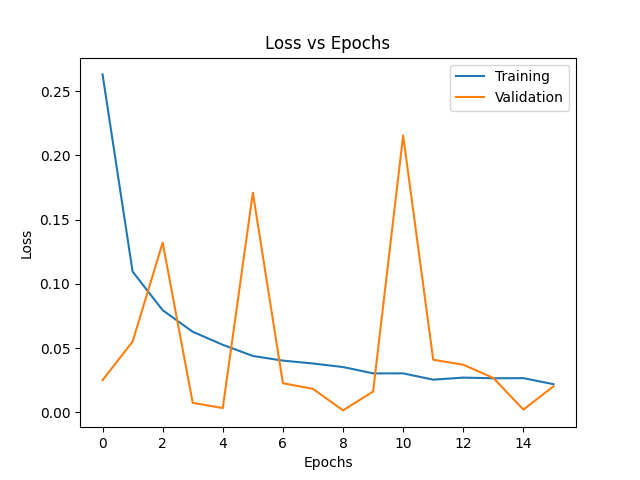
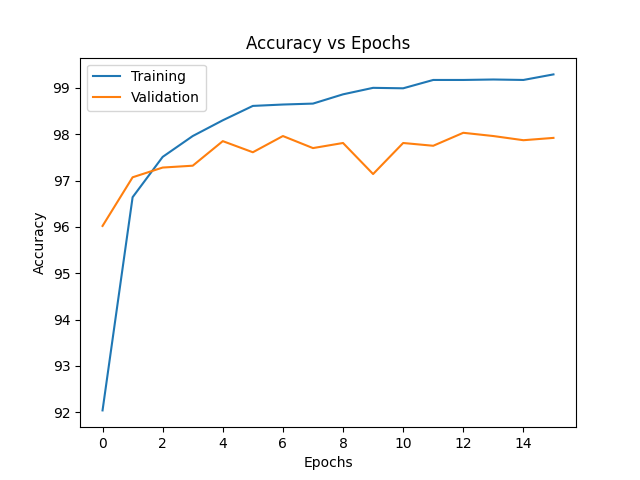


In this observation, we noted that at training images, our model has 0.25 loss but at epochs increases, our losses nearly becomes zero and our model overfits on training images and under performs on validation images. The accuracy of our model on training images is at 100% as epochs reach to 250, and on validation images it gets settled at around 98% as epochs close to 250. This under performance of validation data is mainly due to overfitting of the model on training model.

**Part 1 (b):**

In this case, we have included 12 regularisations, Dropouts with (p =0.2 or 20%) in our training model so that our model doesn't overfit on the training images because of the large dataset and huge number of epochs.

The graphs of Losses and accuracy versus epochs are shown below:



In this model, the accuracy of the model on the training images reaches to 99% and on validation images it reaches to close to 98% and losses on the training images nearly goes to 0.02 and on validation image is in between 0.20 to 0.02.

**Problem 2 (a):**

In this case, we constructed a CNN with a depth of 18 layers, i.e, ResNet -18 model.ResNet is the most popular architecture for classifiers. The users were able to build a very deep, powerful network without running into the problem of vanishing gradients. After training the model, the performance of the model is near to 75-77%. The graphs are shown below:

**Part 2 (b) :**

In this case, we initialised our model with pre-trained datasets, then we freezed initial layers and ran the model on a few last layers to record the performance of our model. For the last 4 layers, our model performance is 79%, and for the last 5 layers, the performance is 80% and for the last 6 layers, performance goes down to 78%. The graphs are shown below:

**Part 2 (c) :**

**Part 2 (d) :**