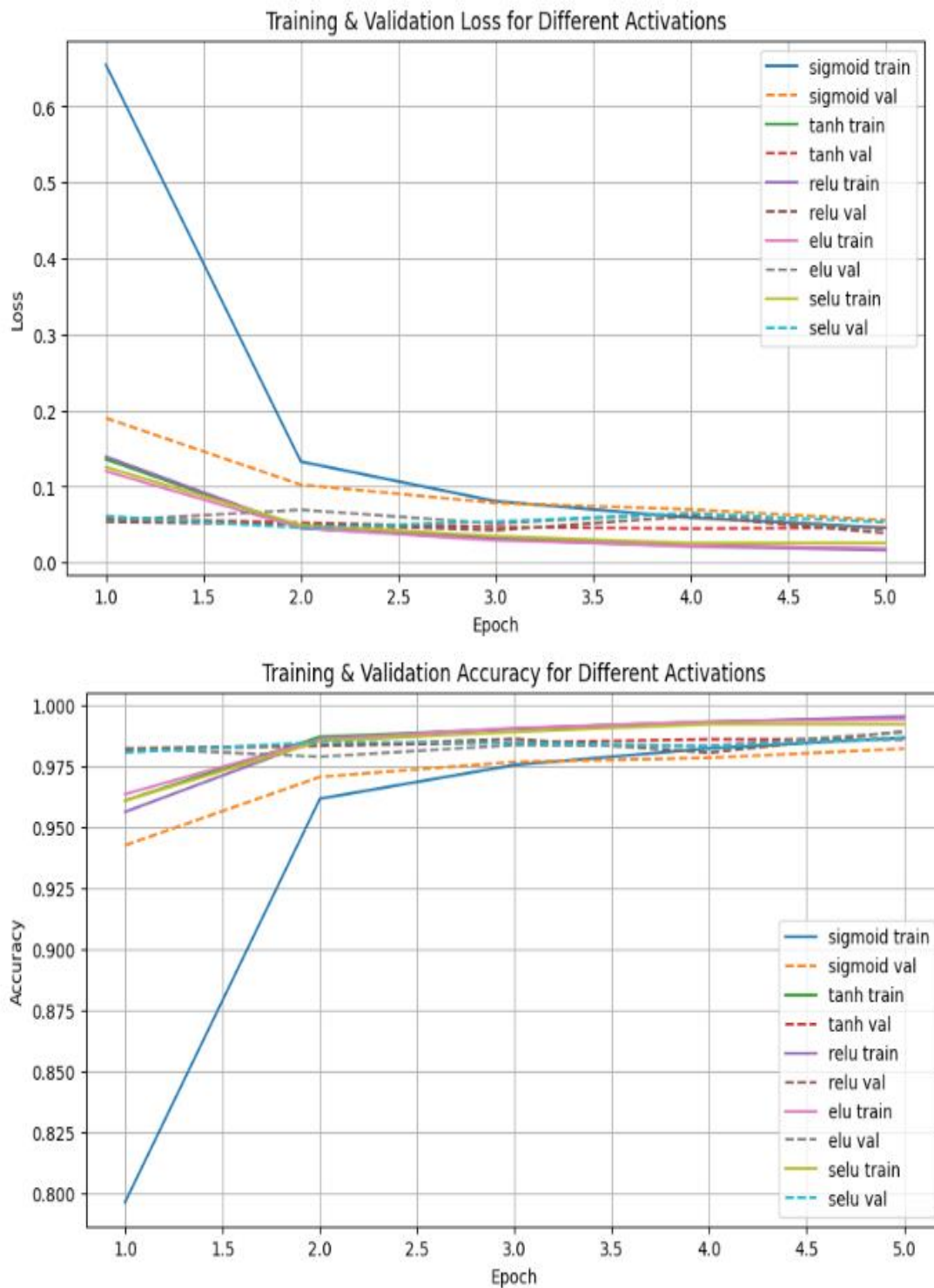


I trained a simple CNN model on the MNIST dataset using PyTorch, importing essential libraries such as torchvision, transforms, and NumPy. The dataset was loaded from the cloud through the torchvision library, and standard normalization was applied using transforms. I trained and evaluated the model across multiple activation functions and plotted accuracy and loss curves for each over both 10-epoch and 5-epoch runs. Based on the results, ReLU and SELU consistently delivered the best and nearly identical performance, showing faster convergence and higher accuracy compared to other activation functions. The screenshots clearly illustrate the training results, including epoch-wise loss and accuracy for the MNIST dataset. At the end there are the best accuracy values for all activation functions.

MNIST dataset-based experiment results:

```
***  
=== Training with activation: sigmoid ===  
Epoch [1/5] Train Loss: 0.6552, Train Acc: 0.7963, Val Loss: 0.1903, Val Acc: 0.9424  
Epoch [2/5] Train Loss: 0.1331, Train Acc: 0.9615, Val Loss: 0.1026, Val Acc: 0.9704  
Epoch [3/5] Train Loss: 0.0809, Train Acc: 0.9753, Val Loss: 0.0786, Val Acc: 0.9766  
Epoch [4/5] Train Loss: 0.0593, Train Acc: 0.9821, Val Loss: 0.0701, Val Acc: 0.9782  
Epoch [5/5] Train Loss: 0.0460, Train Acc: 0.9864, Val Loss: 0.0560, Val Acc: 0.9820  
Final Test Accuracy with sigmoid: 0.9853  
=== Training with activation: tanh ===  
Epoch [1/5] Train Loss: 0.1361, Train Acc: 0.9607, Val Loss: 0.0588, Val Acc: 0.9815  
Epoch [2/5] Train Loss: 0.0457, Train Acc: 0.9868, Val Loss: 0.0523, Val Acc: 0.9832  
Epoch [3/5] Train Loss: 0.0328, Train Acc: 0.9901, Val Loss: 0.0470, Val Acc: 0.9845  
Epoch [4/5] Train Loss: 0.0232, Train Acc: 0.9931, Val Loss: 0.0449, Val Acc: 0.9858  
Epoch [5/5] Train Loss: 0.0185, Train Acc: 0.9945, Val Loss: 0.0462, Val Acc: 0.9858  
Final Test Accuracy with tanh: 0.9880  
=== Training with activation: relu ===  
Epoch [1/5] Train Loss: 0.1396, Train Acc: 0.9561, Val Loss: 0.0538, Val Acc: 0.9822  
Epoch [2/5] Train Loss: 0.0456, Train Acc: 0.9859, Val Loss: 0.0508, Val Acc: 0.9832  
Epoch [3/5] Train Loss: 0.0310, Train Acc: 0.9902, Val Loss: 0.0427, Val Acc: 0.9860  
Epoch [4/5] Train Loss: 0.0214, Train Acc: 0.9929, Val Loss: 0.0612, Val Acc: 0.9803  
Epoch [5/5] Train Loss: 0.0165, Train Acc: 0.9951, Val Loss: 0.0390, Val Acc: 0.9891  
Final Test Accuracy with relu: 0.9901  
=== Training with activation: elu ===  
Epoch [1/5] Train Loss: 0.1205, Train Acc: 0.9634, Val Loss: 0.0553, Val Acc: 0.9822  
Epoch [2/5] Train Loss: 0.0457, Train Acc: 0.9858, Val Loss: 0.0696, Val Acc: 0.9787  
Epoch [3/5] Train Loss: 0.0293, Train Acc: 0.9905, Val Loss: 0.0516, Val Acc: 0.9835  
Epoch [4/5] Train Loss: 0.0225, Train Acc: 0.9930, Val Loss: 0.0654, Val Acc: 0.9831  
Epoch [5/5] Train Loss: 0.0190, Train Acc: 0.9940, Val Loss: 0.0433, Val Acc: 0.9888  
Final Test Accuracy with elu: 0.9901  
=== Training with activation: selu ===  
Epoch [1/5] Train Loss: 0.1258, Train Acc: 0.9607, Val Loss: 0.0613, Val Acc: 0.9804  
Epoch [2/5] Train Loss: 0.0495, Train Acc: 0.9850, Val Loss: 0.0458, Val Acc: 0.9847  
Epoch [3/5] Train Loss: 0.0349, Train Acc: 0.9889, Val Loss: 0.0543, Val Acc: 0.9844  
Epoch [4/5] Train Loss: 0.0259, Train Acc: 0.9921, Val Loss: 0.0648, Val Acc: 0.9829  
Epoch [5/5] Train Loss: 0.0259, Train Acc: 0.9920, Val Loss: 0.0532, Val Acc: 0.9863  
Final Test Accuracy with selu: 0.9877
```

Plot shows the graphical representation of Loss and Accuracy of model on MNIST



The Final values of all activation functions include sigmoid, tanh, Relu, Elu, Selu.

v ...

```
=====
In my Trainig and testing CNN both RELU and ELU perform well and have same accoracy value in testing for MNIST Dataset
```

```
=====
sigmoid: final val acc = 0.9820, test acc = 0.9853
tanh: final val acc = 0.9858, test acc = 0.9880
relu: final val acc = 0.9891, test acc = 0.9901
elu: final val acc = 0.9888, test acc = 0.9901
selu: final val acc = 0.9863, test acc = 0.9877
```

```
=====
🔥 Best activation function(s): relu, elu
🔥 Best final validation accuracy: 0.9901
```



## -----CIFAR10 Experiment-----

I made a small modification to the CNN model for the CIFAR-10 dataset, since its image size is **32×32** instead of 28×28 as in MNIST. I also made some additional improvements.

The **same approach** used for MNIST was applied to CIFAR-10, including testing different activation functions using the same PyTorch implementation. I trained the model with **10 epochs** and **5 epochs**, but the results shown here are from **5 epochs** (to match the MNIST comparison).

For CIFAR-10, CNN performed best with **ReLU**, achieving an accuracy of **71.34%**, while the worst performance came from **Sigmoid**, which only reached about **10%** accuracy. The figures show training accuracy, testing accuracy, training loss, and testing loss. At the end, the screen displays the **best accuracy value** among all the activation functions.

...

=== Activation: sigmoid ===

Epoch [1/5] Train Loss: 2.346, Val Loss: 2.308, Train Acc: 10.1%, Val Acc: 10.0%  
Epoch [2/5] Train Loss: 2.315, Val Loss: 2.307, Train Acc: 10.2%, Val Acc: 10.0%  
Epoch [3/5] Train Loss: 2.312, Val Loss: 2.307, Train Acc: 10.1%, Val Acc: 10.0%  
Epoch [4/5] Train Loss: 2.312, Val Loss: 2.307, Train Acc: 9.9%, Val Acc: 10.0%  
Epoch [5/5] Train Loss: 2.311, Val Loss: 2.307, Train Acc: 10.0%, Val Acc: 10.0%

=== Activation: tanh ===

Epoch [1/5] Train Loss: 1.379, Val Loss: 1.116, Train Acc: 51.5%, Val Acc: 60.8%  
Epoch [2/5] Train Loss: 1.053, Val Loss: 1.023, Train Acc: 63.2%, Val Acc: 63.9%  
Epoch [3/5] Train Loss: 0.955, Val Loss: 0.982, Train Acc: 66.7%, Val Acc: 65.8%  
Epoch [4/5] Train Loss: 0.887, Val Loss: 0.975, Train Acc: 69.0%, Val Acc: 66.6%  
Epoch [5/5] Train Loss: 0.826, Val Loss: 0.951, Train Acc: 71.1%, Val Acc: 67.3%

=== Activation: relu ===

Epoch [1/5] Train Loss: 1.371, Val Loss: 1.060, Train Acc: 50.7%, Val Acc: 62.7%  
Epoch [2/5] Train Loss: 0.986, Val Loss: 0.908, Train Acc: 65.4%, Val Acc: 68.0%  
Epoch [3/5] Train Loss: 0.819, Val Loss: 0.861, Train Acc: 71.4%, Val Acc: 69.8%  
Epoch [4/5] Train Loss: 0.692, Val Loss: 0.819, Train Acc: 75.7%, Val Acc: 71.1%  
Epoch [5/5] Train Loss: 0.569, Val Loss: 0.854, Train Acc: 80.1%, Val Acc: 71.3%

=== Activation: elu ===

Epoch [1/5] Train Loss: 1.340, Val Loss: 1.038, Train Acc: 52.6%, Val Acc: 63.8%  
Epoch [2/5] Train Loss: 1.003, Val Loss: 0.952, Train Acc: 64.8%, Val Acc: 66.3%  
Epoch [3/5] Train Loss: 0.860, Val Loss: 0.901, Train Acc: 69.9%, Val Acc: 68.7%  
Epoch [4/5] Train Loss: 0.744, Val Loss: 0.876, Train Acc: 73.8%, Val Acc: 70.2%  
Epoch [5/5] Train Loss: 0.642, Val Loss: 0.911, Train Acc: 77.5%, Val Acc: 69.7%

=== Activation: selu ===

Epoch [1/5] Train Loss: 1.424, Val Loss: 1.076, Train Acc: 51.0%, Val Acc: 63.0%  
Epoch [2/5] Train Loss: 1.067, Val Loss: 0.988, Train Acc: 62.7%, Val Acc: 64.9%  
Epoch [3/5] Train Loss: 0.933, Val Loss: 0.942, Train Acc: 67.6%, Val Acc: 67.3%  
Epoch [4/5] Train Loss: 0.827, Val Loss: 0.914, Train Acc: 70.8%, Val Acc: 67.9%  
Epoch [5/5] Train Loss: 0.738, Val Loss: 0.952, Train Acc: 74.0%, Val Acc: 68.5%

Best validation accuracy for each activation:

sigmoid: 10.00%

tanh : 67.35%

relu : 71.34%

elu : 70.23%

selu : 68.52%

..

