Machine Learning

Introduction

LeNet-5:

One of the original convolutional neural networks, LeNet-5 helped to advance deep learning. LeNet-5 has been the name of the ground-breaking work since 1988, following years of study and several successful iterations.

LeNet-5, a prototype of the first convolutional neural network, possesses the fundamental components of a convolutional neural network, including the convolutional layer, pooling layer, and complete connection layer, providing the groundwork for its future advancement. As shown in the graphic (32*32 pixels of raw picture data): Seven layers make up LeNet-5. Every other layer can train parameters in addition to input. The layers in the diagram are labeled Cx for convolution, Sx for sub-sampling, Fx for full connection, and x for layer index.

Pytorch:

The Torch library-based machine learning framework PyTorch was created by Meta AI and is now a member of the Linux Foundation. It is used for applications like computer vision and natural language processing. It is software that is available for free and open source under a modified BSD license.

MNIST:

The MNIST dataset is an acronym that stands for the Modified National Institute of Standards and Technology dataset**.** It has handwritten numbers in 28x28 (or 32x32) grayscale pictures, each with an integer between 0 and 9. MNIST's size, which enables deep learning practitioners to quickly assess, train, and publish their algorithms, is one of the reasons it is so well-liked.

CNN is mostly used for MNIST dataset because using CNN (Convolutional Neural Network) and a functional model, we can get 99.06% accuracy. The use of a functional model is necessary to connect the layers with ease.

Results on Dataset:

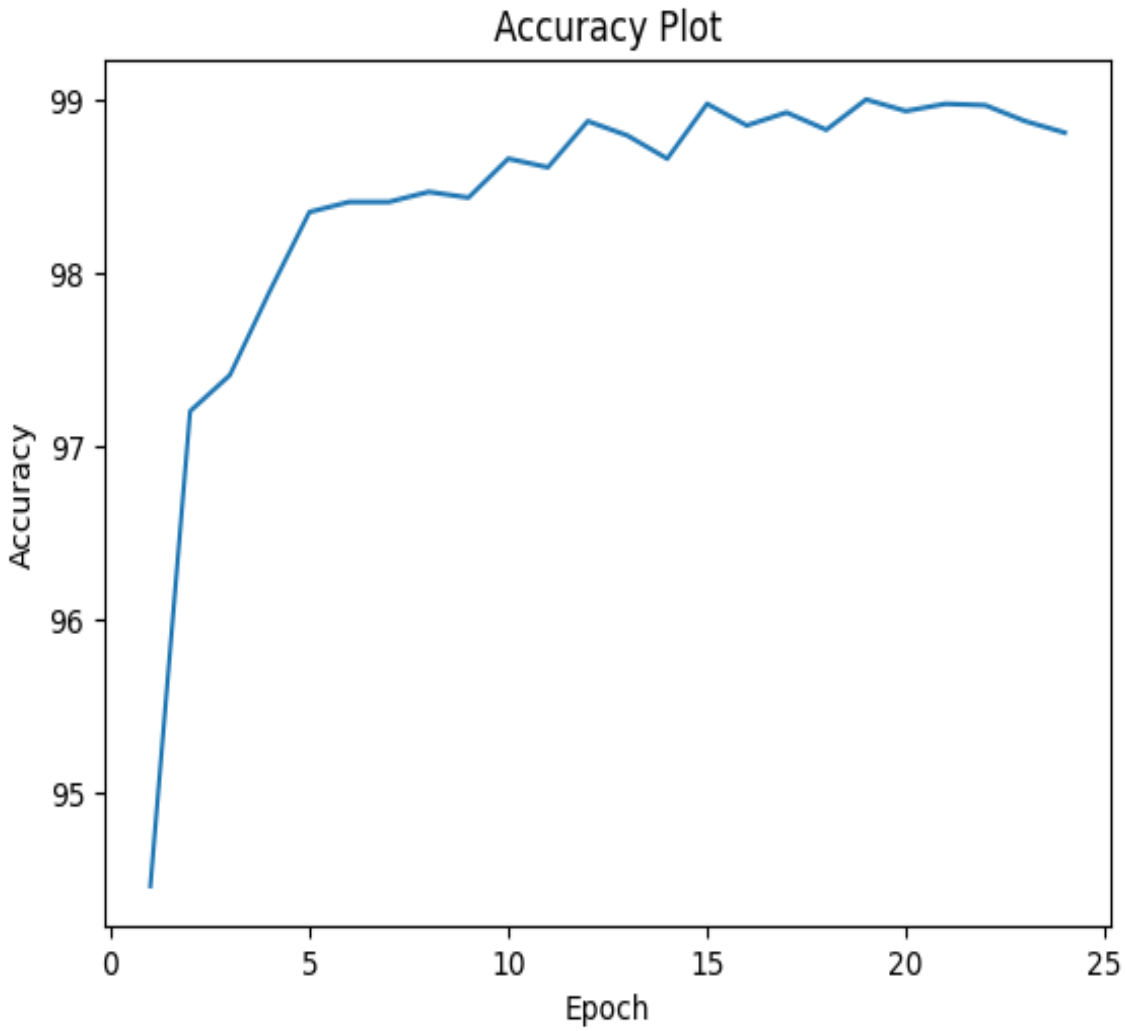
Epochs:

The Pytorch model was trained for 30 epochs and got the following results.

```
Epoch: 1/24 Train loss:0.4344 Test loss:0.1859 Test acc:94.4583
Epoch: 2/24 Train loss:0.1304 Test loss:0.0956 Test acc:97.2000
Epoch: 3/24 Train loss:0.0891 Test loss:0.0846 Test acc:97.4083
Epoch: 4/24 Train loss:0.0707 Test loss:0.0719 Test acc:97.8917
Epoch: 5/24 Train loss:0.0606 Test loss:0.0569 Test acc:98.3500
Epoch: 6/24 Train loss:0.0503 Test loss:0.0526 Test acc:98.4083
Epoch: 7/24 Train loss:0.0440 Test loss:0.0537 Test acc:98.4083
Epoch: 8/24 Train loss:0.0387 Test loss:0.0537 Test acc:98.4667
Epoch: 9/24 Train loss:0.0342 Test loss:0.0482 Test acc:98.4333
Epoch: 10/24 Train loss:0.0311 Test loss:0.0428 Test acc:98.6583
Epoch: 11/24 Train loss:0.0276 Test loss:0.0477 Test acc:98.6083
Epoch: 12/24 Train loss:0.0255 Test loss:0.0408 Test acc:98.8750
Epoch: 13/24 Train loss:0.0228 Test loss:0.0420 Test acc:98.7917
Epoch: 14/24 Train loss:0.0192 Test loss:0.0478 Test acc:98.6583
Epoch: 15/24 Train loss:0.0191 Test loss:0.0412 Test acc:98.9750
Epoch: 16/24 Train loss:0.0167 Test loss:0.0452 Test acc:98.8500
Epoch: 17/24 Train loss:0.0148 Test loss:0.0390 Test acc:98.9250
Epoch: 18/24 Train loss:0.0135 Test loss:0.0416 Test acc:98.8250
Epoch: 19/24 Train loss:0.0126 Test loss:0.0425 Test acc:99.0000
Epoch: 20/24 Train loss:0.0130 Test loss:0.0420 Test acc:98.9333
Epoch: 21/24 Train loss:0.0086 Test loss:0.0416 Test acc:98.9750
Epoch: 22/24 Train loss:0.0121 Test loss:0.0426 Test acc:98.9667
Epoch: 23/24 Train loss:0.0098 Test loss:0.0459 Test acc:98.8750
Epoch: 24/24 Train loss:0.0095 Test loss:0.0505 Test acc:98.8083
Best acc:99.0000
```

Accuracy:

The best accuracy achieved was 99.000. We iterated through the number of epochs and did forward and backward passes to make predictions using our model and update weights of our model respectively. We then calculated the gradients and updated the weights to achieve the following results.



Analysis:

On test we get an accuracy of 98.7% which is good even on a basic dataset such as MNIST.

We can observe that the loss keeps decreasing with every epoch, hence we can say that the model is learning. It can also be an indication

of overfitting. It can be solved using different methods which are not discussed here as they are out of scope for the following assignment.