

COMP 5313- ARTIFICIAL INTELLIGENCE

ASSIGNMENT 1

MLP for Image Classification: Connectionist AI

Methodology:

I created one .py file in order to create the multilayer perceptron modelling to develop and test a neural net that are able to classify images like the handwritten digits with high accuracy.

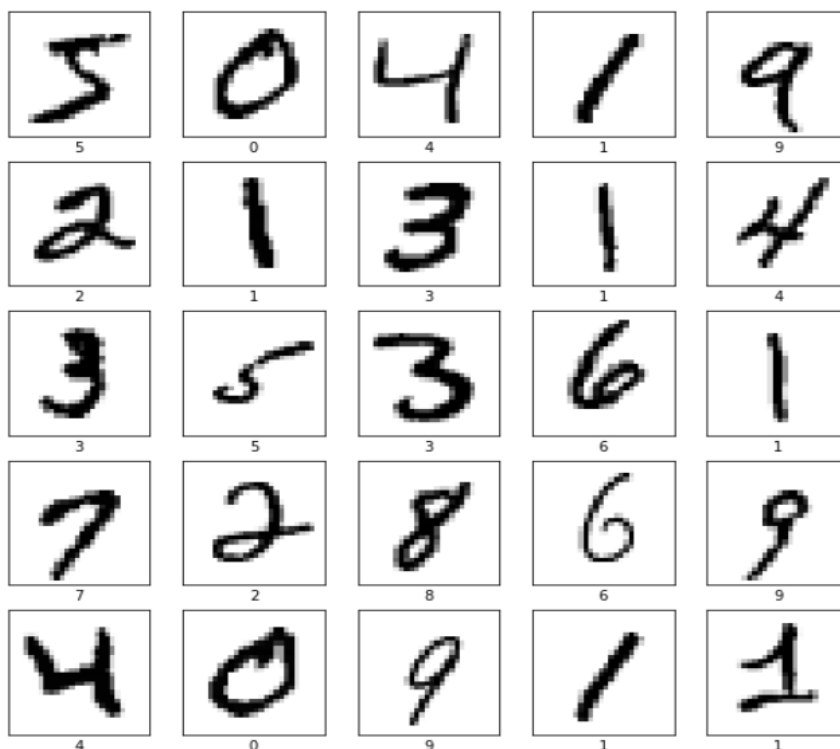
The dataset that I used to train the models is the MNIST provided by the tensorflow API.

I used many ways to define the MLP models to classify the handwritten digits like Simple three-layer MLP, MLP-Mixer model, FNet model, gMLP model and ResMLP.

Dataset:

I loaded and preprocessed the MNIST dataset. Then reshaped the data to have a single channel.

```
plt.show()
```



Simple Three Layer MLP

Multi-layer perceptron (MLP) is a supplement of feed forward neural network. It consists of three types of layers—the input layer, output layer and hidden layer.

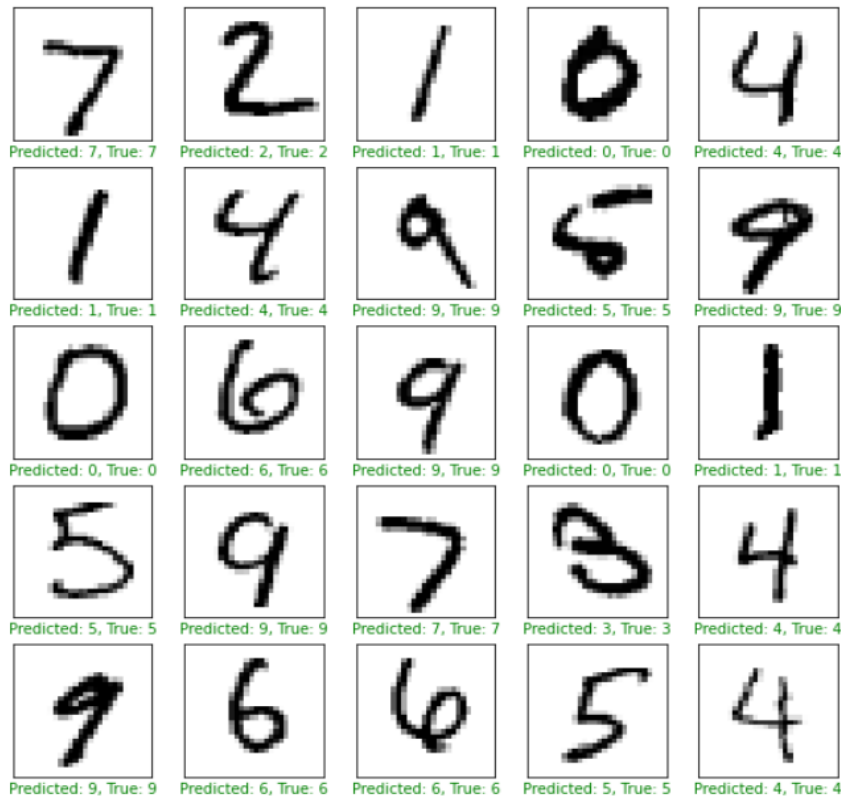
It got an accuracy rate of 97.63.

```
# Evaluate the model on the test set
test_loss, test_accuracy = model.evaluate(x_test, y_test)
print("Test Accuracy of Simple Three Layer MLP:", (test_accuracy*100))
```

```
313/313 [=====] - 1s 3ms/step - loss: 0.0942 - accuracy: 0.9763
Test Accuracy of Simple Three Layer MLP: 97.63000011444092
```

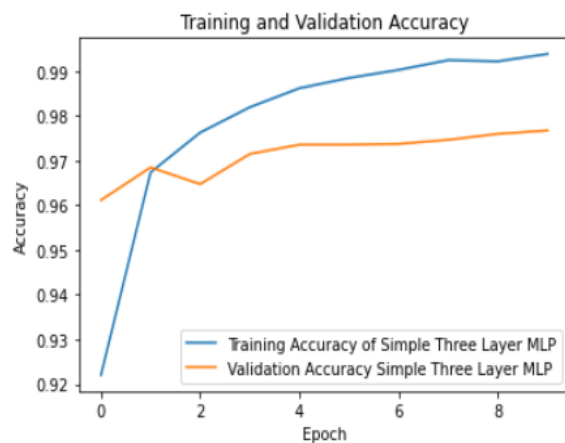
Visualization of the prediction by Simple three-layer MLP.

```
plt.show()
```



Visualization of training and validation accuracy.

```
plt.show()
```



gMLP model:

gMLP is an MLP-based alternative to Transformers without self-attention, which simply consists of channel projections and spatial projections with static parameterization.

It got an accuracy rate of 94.16.

```
# Evaluate the model on the test set
test_loss, test_accuracy = model.evaluate(x_test, y_test)
print("Test Accuracy of gMLP model:", (test_accuracy*100))
```

313/313 [=====] - 7s 14ms/step - loss: 0.2031 - accuracy: 0.9416
Test Accuracy of gMLP model: 94.16000247001648

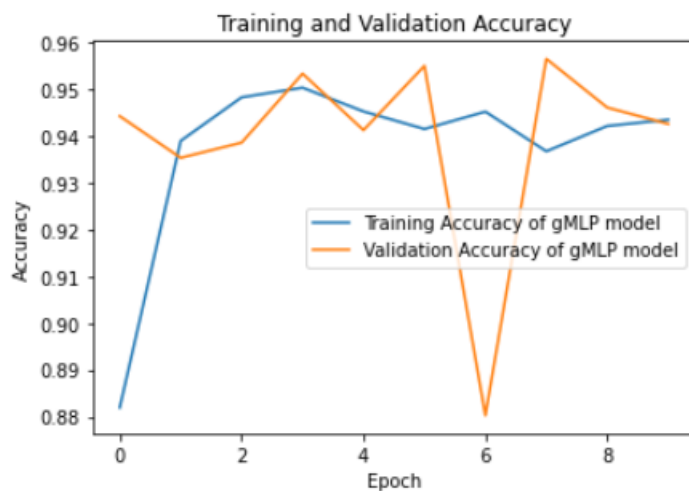
Visualization of the prediction by gMLP model.

```
plt.show()
```



Visualization of training and validation accuracy.

```
plt.show()
```



ResMLP:

ResMLP is proposed where it is built entirely upon multi-layer perceptrons (MLP). It is a simple residual network that alternates:

1. A linear layer in which image patches interact, independently and identically across channels, and
2. A two-layer feed-forward network in which channels interact independently per patch.

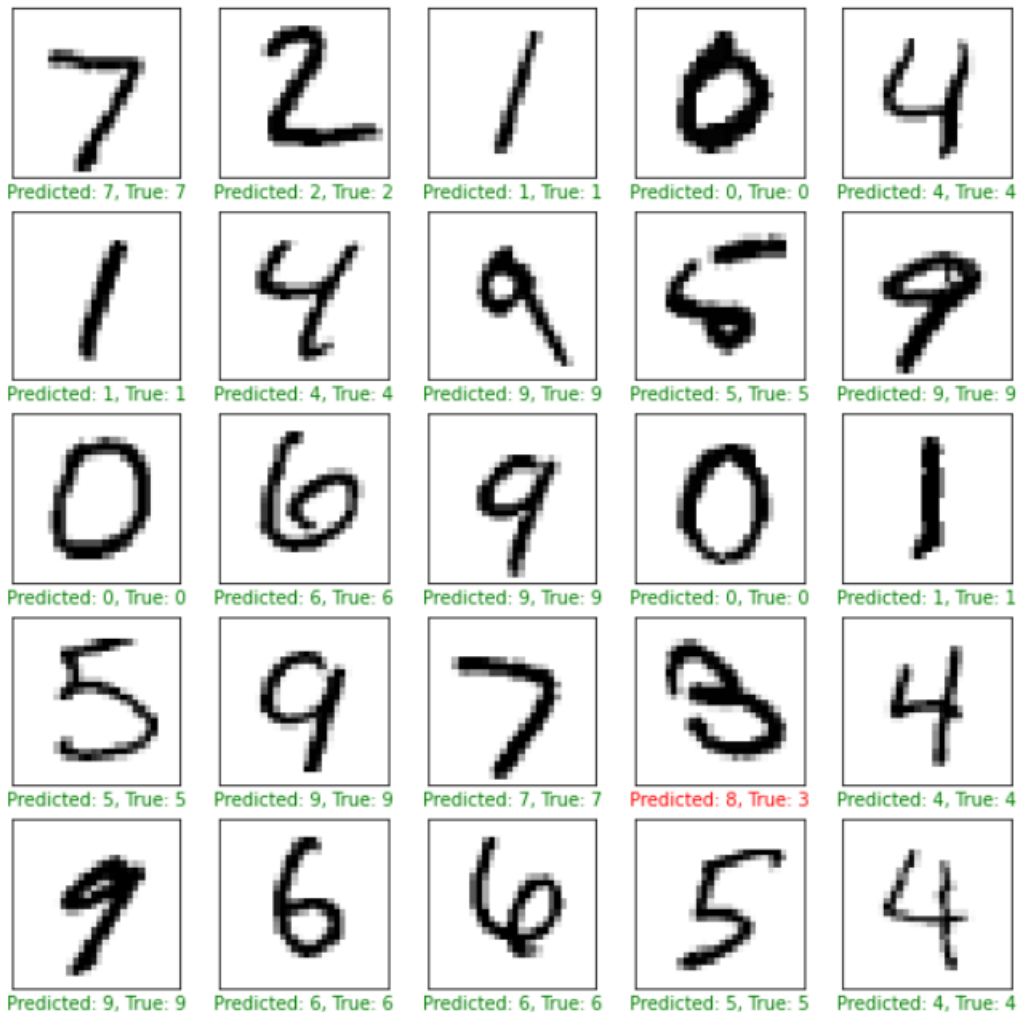
It got an accuracy rate of 97.72.

```
# Evaluate the model on the test set
test_loss, test_accuracy = model.evaluate(x_test, y_test)
print("Test Accuracy of ResMLP:", (test_accuracy*100))
```

```
313/313 [=====] - 4s 8ms/step - loss: 0.0870 - accuracy: 0.9772
Test Accuracy of ResMLP: 97.71999716758728
```

Visualization of the prediction by ResMLP model.

```
plt.show()
```



Visualization of training and validation accuracy.

```
plt.show()
```



Conclusion:

As you can see from the above results the simple three-layer MLP has an accuracy rate of 97.63 then gMLP model has an accuracy of 94.16 finally ResMLP has an accuracy of 97.72. From this we can decide ResMLP is better.

References:

- [1] [https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron#:~:text=Multi%20layer%20perceptron%20\(MLP\)%20is,input%20signal%20to%20be%20processed.](https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron#:~:text=Multi%20layer%20perceptron%20(MLP)%20is,input%20signal%20to%20be%20processed.)
- [2] <https://paperswithcode.com/method/mlp-mixer>
- [3] https://keras.io/examples/vision/mlp_image_classification/
- [4] <https://sh-tsang.medium.com/review-resmlp-feedforward-networks-for-image-classification-with-data-efficient-training-4eeb1eb5efa6>