# Option#1 Building a tensor flow demo

This Jupyter notebook demonstrates the implementation of a neural network for image classification using a dataset of handwritten digits (MNIST). The goal is to train a model to accurately classify digits from 0 to 9.

# Overview

This Jupyter notebook demonstrates the implementation of a neural network for image classification using a dataset of handwritten digits (MNIST). The goal is to train a model to accurately classify digits from 0 to 9.

# Sections

1. Setup and Imports  
2. Data Preparation  
3. Model Building  
4. Model Training  
5. Model Evaluation  
6. Model Prediction

## 1. Setup and Imports

In this section, the necessary libraries and modules are imported, including TensorFlow and other dependencies for building and training the neural network.

```   
import tensorflow as tf  
from tensorflow.keras import datasets, layers, models  
import matplotlib.pyplot as plt  
import numpy as np  
```

## 2. Data Preparation

The MNIST dataset is loaded, and the data is preprocessed. The dataset is split into training and testing sets, and the images are normalized to improve the model's performance.

```   
(x\_train, y\_train), (x\_test, y\_test) = datasets.mnist.load\_data()  
x\_train, x\_test = x\_train / 255.0, x\_test / 255.0  
```

## 3. Model Building

A Convolutional Neural Network (CNN) is built using Keras' Sequential API. The model consists of convolutional layers, pooling layers, and dense layers.

```  
model = models.Sequential([  
 layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(28, 28, 1)),  
 layers.MaxPooling2D((2, 2)),  
 layers.Conv2D(64, (3, 3), activation='relu'),  
 layers.MaxPooling2D((2, 2)),  
 layers.Conv2D(64, (3, 3), activation='relu'),  
 layers.Flatten(),  
 layers.Dense(64, activation='relu'),  
 layers.Dense(10)  
])  
```

## 4. Model Training

The model is compiled with an appropriate optimizer, loss function, and metrics. It is then trained on the training dataset.

```  
model.compile(optimizer='adam',  
 loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),  
 metrics=['accuracy'])  
  
history = model.fit(x\_train, y\_train, epochs=5,   
 validation\_data=(x\_test, y\_test))  
```

## 5. Model Evaluation

The trained model is evaluated on the test dataset to determine its accuracy and loss.

```  
test\_loss, test\_acc = model.evaluate(x\_test, y\_test, verbose=2)  
print(f'\nTest accuracy: {test\_acc}')  
```

## 6. Model Prediction

The model makes predictions on a few samples from the test dataset. The predicted labels are displayed along with the corresponding images.

```  
n\_images = 5  
test\_images = x\_test[:n\_images]  
predictions = model.predict(test\_images)  
  
for i in range(n\_images):  
 plt.imshow(np.reshape(test\_images[i], [28, 28]), cmap='gray')  
 plt.show()  
 print(f'Model prediction: {np.argmax(predictions[i])}')  
```

## Results

The model's performance is summarized by its accuracy on the test set. Additionally, a few sample predictions are shown to demonstrate the model's ability to classify handwritten digits.

```   
Test accuracy: 0.98  
Model prediction: 7  
Model prediction: 2  
Model prediction: 1  
Model prediction: 0  
Model prediction: 4  
```

## Conclusion

This notebook provides a step-by-step guide to building, training, and evaluating a convolutional neural network for digit classification using the MNIST dataset. The model achieves high accuracy and demonstrates effective image classification capabilities.

## Additional Q&A

## What was notable during the installation of TensorFlow? Did you run into any problems?

The installation of TensorFlow was straightforward using pip, the Python package installer. Since I am using a normal intel cpu I installed the windows native version from the portal <https://www.tensorflow.org/install/pip#windows-native_1>

## Describe your chosen example. How did the example run? What did it do?

The chosen example demonstrates the implementation of a Convolutional Neural Network (CNN) for classifying images from the MNIST dataset, which consists of handwritten digits. The example runs by loading the dataset, preprocessing the images, building the CNN model, training it on the training dataset, evaluating its performance on the test dataset, and making predictions on a few sample images. The example successfully trains the model to achieve high accuracy in digit classification.

## What type of model does your example use? Do further research to describe the model type and its common applications.

The example uses an Convolutional Neural Network (CNN). ANNs are a class of deep neural networks that are particularly effective for image processing tasks. They use convolutional layers to automatically and adaptively learn spatial hierarchies of features from input images. Common applications of CNNs include image classification, object detection, facial recognition, and medical image analysis.

## How might you leverage techniques found in your example for your portfolio project?

The techniques demonstrated in this example, such as data preprocessing, model building, and evaluation, can be leveraged in my portfolio project involving climate and image data.

## What type of dataset did the example use? Is it possible to improve that dataset? What other ways might the model type used in the example be expanded, improved upon, or integrated into other techniques?

The example uses the MNIST dataset, which is a collection of 70,000 handwritten digit images. While the MNIST dataset is well-suited for benchmarking image classification models, it is relatively simple and may not represent more complex image recognition tasks. To improve the dataset, we could use more diverse and complex image datasets, such as CIFAR-10 or ImageNet. Additionally, the model can be expanded or improved by using more advanced architectures like ResNet or EfficientNet, incorporating techniques such as data augmentation, or integrating the ANN with other machine learning models or frameworks for enhanced performance.

## Is the model used an example of a supervised or unsupervised model?

The model used in this example is an example of a supervised learning model. Supervised learning involves training a model on a labeled dataset, where each input is paired with the correct output. In this case, the MNIST dataset provides labeled images of digits, allowing the ANN to learn to classify new images based on the patterns it has seen during training.

Ref - <https://github.com/ArunSaxena200/TensorFlow-Examples/blob/master/tensorflow_v2/notebooks/1_Introduction/helloworld.ipynb>

<https://www.tensorflow.org/install/pip#windows-native_1>