**Getting Started with TensorFlow: Building Your First Neural Network**

**1. Introduction**

TensorFlow is an open-source machine learning framework developed by Google that enables developers to build and train deep learning models efficiently. In this guide, we'll walk through the process of building a simple **neural network** using TensorFlow and Keras.

**2. Prerequisites**

Before we begin, ensure you have the following installed:

* **Python 3.x**
* **TensorFlow** (pip install tensorflow)
* **NumPy, Matplotlib** (pip install numpy matplotlib)
* Basic understanding of Python and neural networks

**3. Understanding Neural Networks**

A **neural network** consists of:

* **Input Layer** – Receives data for training.
* **Hidden Layers** – Processes data using weighted connections and activation functions.
* **Output Layer** – Produces the final prediction.

**4. Building Your First Neural Network in TensorFlow**

**Step 1: Import Dependencies**

**import tensorflow as tf**

**from tensorflow import keras**

**import numpy as np**

**import matplotlib.pyplot as plt**

**Step 2: Load and Prepare the Dataset**

For this example, we'll use the **MNIST dataset**, which consists of handwritten digits.

# Load dataset

**mnist = keras.datasets.mnist**

**(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()**

# Normalize pixel values to [0,1] range

**x\_train, x\_test = x\_train / 255.0, x\_test / 255.0**

**Step 3: Define the Neural Network Model**

**model = keras.Sequential([**

**keras.layers.Flatten(input\_shape=(28, 28)), # Flatten 2D image into 1D**

**keras.layers.Dense(128, activation='relu'), # Hidden layer with 128 neurons**

**keras.layers.Dense(10, activation='softmax') # Output layer with 10 classes**

**])**

**Step 4: Compile the Model**

**model.compile(optimizer='adam',**

**loss='sparse\_categorical\_crossentropy',**

**metrics=['accuracy'])**

**Step 5: Train the Model**

**model.fit(x\_train, y\_train, epochs=5)**

**Step 6: Evaluate the Model**

**test\_loss, test\_acc = model.evaluate(x\_test, y\_test, verbose=2)**

**print(f'\nTest accuracy: {test\_acc}')**

**Step 7: Make Predictions**

**predictions = model.predict(x\_test)**

**print(f'Predicted class for first test image: {np.argmax(predictions[0])}')**

**5. Conclusion**

Congratulations! You’ve built your first neural network using TensorFlow. This simple model recognizes handwritten digits with decent accuracy.

**Next Steps:**

1. Experiment with different architectures and activation functions.
2. Use **convolutional neural networks (CNNs)** for improved accuracy.
3. Train on custom datasets for real-world applications.

By mastering TensorFlow, you unlock the potential to build powerful AI models.