**Introduction to Deep Learning**

**1. What is Deep Learning?**

* **Artificial Intelligence (AI):** Machines that can act smart.
* **Machine Learning (ML):** A way to make machines learn from data.
* **Deep Learning (DL):** A part of ML that uses **Neural Networks with many layers** to learn from data automatically.

In simple words:  
Deep Learning is like **teaching a computer brain** to recognize patterns, just like humans do.

**2. Why is it called "Deep"?**

* "Deep" means **many layers** in a neural network.
* A simple network = few layers (Shallow Learning).
* A deep network = many hidden layers → more learning power.

**3. Where is Deep Learning used?**

* Face Recognition (Facebook, iPhone Face ID)
* Voice Assistants (Siri, Alexa, Google Assistant)
* Self-driving cars (detect pedestrians, traffic lights)
* Google Translate (language translation)
* Healthcare (detecting diseases from X-rays)

**4. How does it work? (Very Simple)**

Imagine teaching a child to recognize **cats vs dogs**:

1. Show many pictures of cats and dogs.
2. The child starts noticing features (ears, tails, fur).
3. With practice, the child gets better.

Neural networks work the same way → they **learn from lots of data**.

**5. Key Terms (Simplified)**

* **Neuron:** Small unit that takes input, processes it, and gives output (like brain cell).
* **Layers:** Collection of neurons (input layer → hidden layers → output layer).
* **Weights:** Numbers that tell how important each input is.
* **Activation function:** Decides if the neuron should "fire" (like a switch).
* **Training:** Process of learning by adjusting weights using data.

**6. A Tiny Example (Python Code )**

Here’s a **very simple neural network** using TensorFlow/Keras:

import tensorflow as tf

from tensorflow.keras import layers, models

# Sample dataset: MNIST digits (0-9 images)

(x\_train, y\_train), (x\_test, y\_test) = tf.keras.datasets.mnist.load\_data()

x\_train, x\_test = x\_train / 255.0, x\_test / 255.0 # normalize (0-1)

# Build a simple neural network

model = models.Sequential([

layers.Flatten(input\_shape=(28, 28)), # Input layer (images 28x28)

layers.Dense(64, activation='relu'), # Hidden layer

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

])

# Compile the model

model.compile(optimizer='adam',

loss='sparse\_categorical\_crossentropy',

metrics=['accuracy'])

# Train the model

model.fit(x\_train, y\_train, epochs=3, validation\_data=(x\_test, y\_test))