Deep Learning:

# 1. Introduction to Deep Learning

Deep Learning is a subset of Machine Learning inspired by the structure and function of the human brain - called Artificial Neural Networks (ANNs). It is especially powerful for tasks involving large, unstructured data like images, speech, and text.

# 2. Architecture of an Artificial Neural Network (ANN)

An ANN consists of layers of neurons:  
- Input layer: Receives the raw data (e.g., pixels of an image)  
- Hidden layers: Perform computations using weights, biases, and activation functions  
- Output layer: Gives the final prediction (e.g., class label)  
Each connection has a weight and a bias, and passes through an activation function.

These could be single layer perceptron or multi-layer perceptrons depending on the number of hidden layers.

# 3. Weights and Biases

Weights determine how strongly an input feature influences the output. Biases allow shifting the activation function left or right. These are learnable parameters optimized during training.  
Formula: z = 〖I=1wixi + b

# 4. Dimensionality and Tensors

Input data may be in the form of:  
- 1D (vectors)  
- 2D (matrices)  
- 3D/4D (tensors for images, batches)  
Tensors generalize matrices to higher dimensions and are the native format used in frameworks like TensorFlow and PyTorch.

# 5. Activation Functions

Activation functions introduce non-linearity to learn complex patterns.  
- Sigmoid: 1 / (1 + e^-z)  
- Tanh: (e^z- e^-z) / (e^z + e^-z) or 2/(1+e^-2z) +1  
- ReLU: max(0, z)  
- Softmax: Converts outputs to probabilities. In the range of 0 to 1

# 6. How Neural Networks Learn

The learning process involves:  
1. **Forward Propagation  
2. Loss Calculation  
3. Backward Propagation (gradient calculation)  
4. Weight Update (using optimizers).**

# 7. Optimizer Algorithms

- SGD: Stochastic Gradient Descent can work only on one sample or a small batch of samples.  
- Momentum: Adds velocity  
- RMSprop: Adapts learning rate  
- Adam: Combines momentum and RMSprop (widely used) and an extended version of SGD

# 8. Hyperparameter Tuning

Hyperparameters include:  
- Learning rate-**controls **how much the model's weights are updated** in response to the estimated error (loss) each time the model learns.**

**For the model handwriting detection, I just used 5 epochs but still managed to get an accuracy of 97percent whereas even after using 100 epochs in next word prediction model, I could get only 65 percent accuracy that could be due to low learning rate .**  
- Batch size-Number of training samples processed before theweights are updated once  
- Epochs-number of times a model learns from the training set.  
- Number of layers and neurons  
Tuning methods: Grid Search, Random Search, Bayesian Optimization, or manual tuning using validation set.

# 9. Training and Evaluation

Split data into training, validation, and test sets. Use accuracy, precision, recall, and loss to monitor performance. Use dropout, early stopping, and regularization to prevent overfitting.

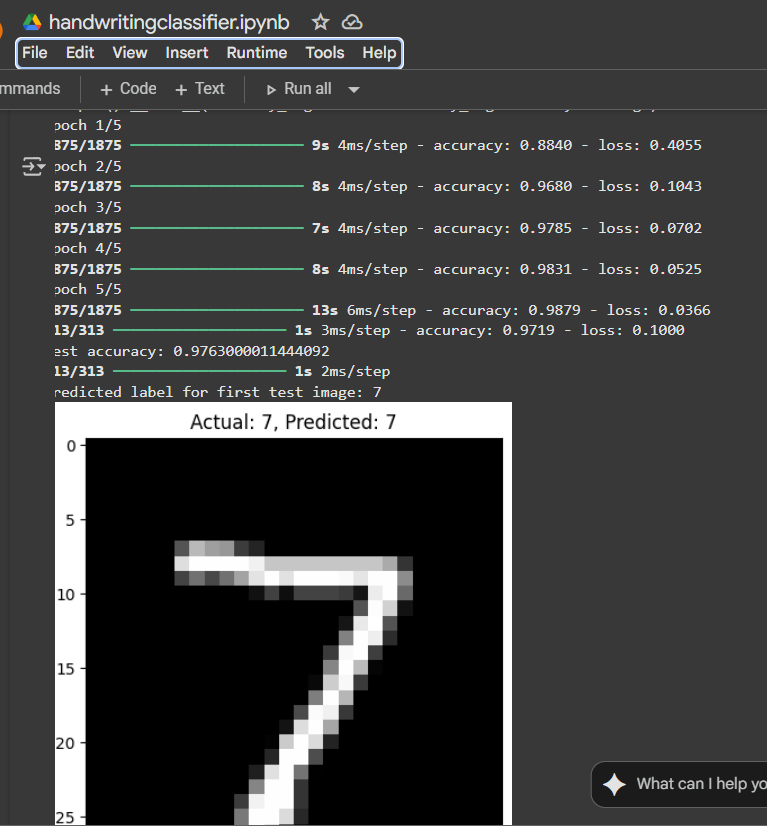
# 10. Applications of Deep Learning

Deep learning is used in:  
- Vision: Image classification, object detection  
- NLP: Chatbots, sentiment analysis  
- Healthcare: Disease detection  
- Finance: Fraud detection, stock forecasting.

# Conclusion

Deep learning enables machines to learn from raw data through layers of abstraction using weights, biases, activation functions, and optimizers. It-s transforming fields like vision, language, and medicine.

Hand writing detection model



Next word Prediction;

