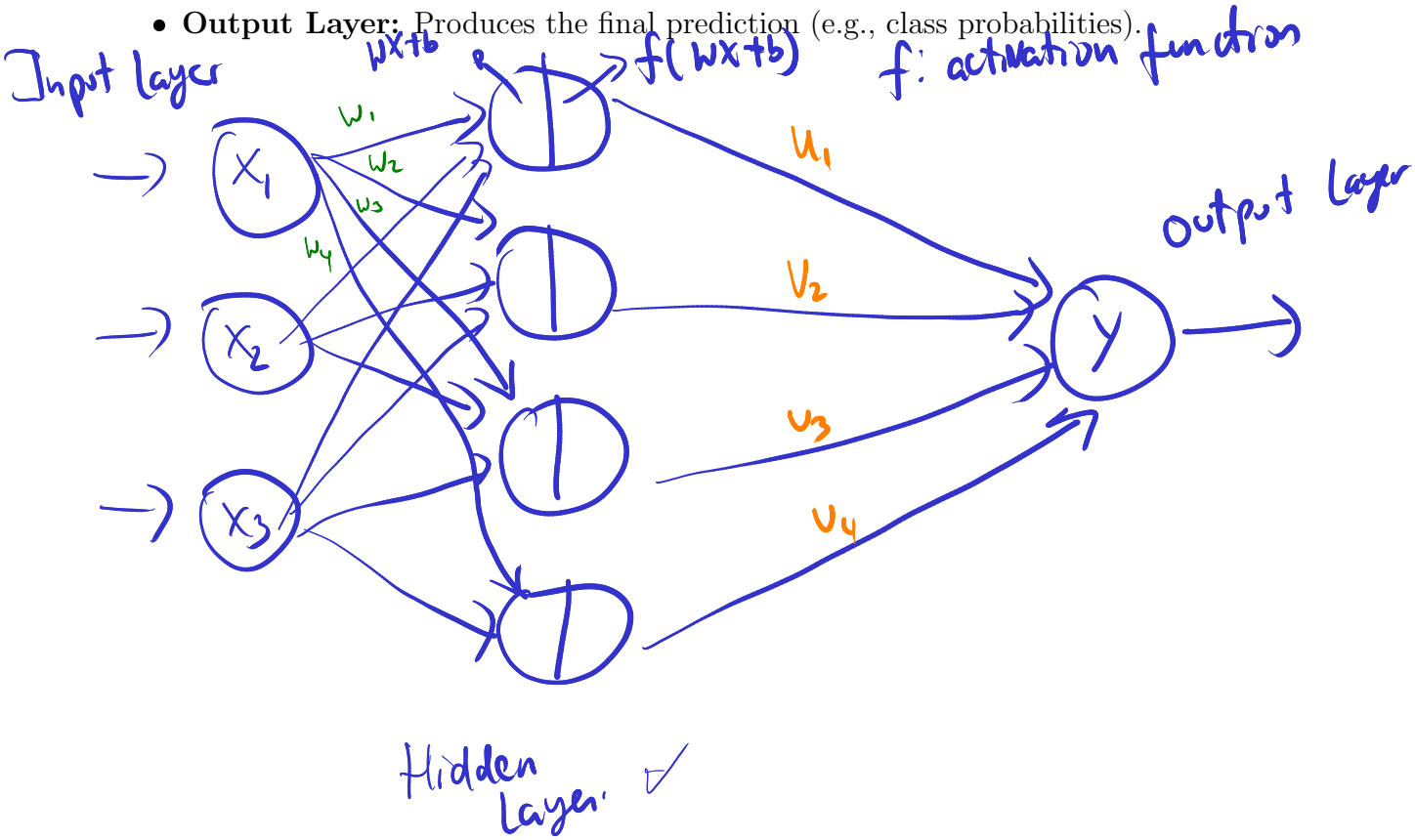
Introduction to Multi-Layer Perceptrons (MLPs)

1. Introduction

Multi-Layer Perceptrons (MLPs) are one of the most widely used types of feedforward artificial neural networks. An MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer is made up of nodes (neurons) that use activation functions to learn non-linear patterns.

2. MLP Architecture

- **Input Layer:** Receives raw input features.
- **Hidden Layers:** Perform non-linear transformations using activation functions.
- **Output Layer:** Produces the final prediction (e.g., class probabilities).



Each neuron computes:

$$z = w^T x + b, \quad a = f(z)$$

where w are the weights, b is the bias, x is the input, and f is the activation function (e.g., ReLU, sigmoid).

3. Applications of MLPs

MLPs are general-purpose models suitable for both regression and classification tasks. Some common applications include:

- **Image Classification:** Handwritten digit recognition (e.g., MNIST dataset).
- **Medical Diagnosis:** Predicting disease from patient data.
- **Fraud Detection:** Identifying suspicious transactions.
- **Credit Scoring:** Evaluating loan applications.
- **Forecasting:** Stock price prediction or demand forecasting.

4. Pros of MLPs

- **Universal Function Approximator:** Can model complex, non-linear relationships.
- **Versatile:** Can be applied to a wide range of supervised learning tasks.
- **Feature Learning:** Capable of automatically learning relevant features.
- **Scikit-learn Support:** Easy to implement using tools like `MLPClassifier`.

5. Cons of MLPs

- **Black Box:** Difficult to interpret internal mechanisms.
- **Overfitting:** Prone to overfitting, especially on small datasets.
- **Computational Cost:** Training can be slow and resource-intensive.
- **Hyperparameter Tuning:** Sensitive to parameters like learning rate and architecture.

6. Conclusion

MLPs provide a powerful foundation for deep learning models and are widely applicable across various domains. Understanding their structure and trade-offs is essential for any data scientist working on predictive modeling tasks.