Class #3:

📌 Formula and Notations:

✒️ The general formula to calculate the result of a single neuron in a layer is:

z = (w1*x1 + w2*x2 + ... + wn*xn) + b

Where:

z: The weighted sum of the inputs plus the bias. This is the value that will be passed to the activation function.

w1, w2, ..., wn: The weights connecting the neuron to the inputs from the previous layer.

x1, x2, ..., xn: The inputs from the previous layer.
b: The bias term.

In matrix form:

Z = W * X + B

Where:

Z = Weighted sum (before activation)

W = Weight matrix

X = Input vector

B = Bias vector

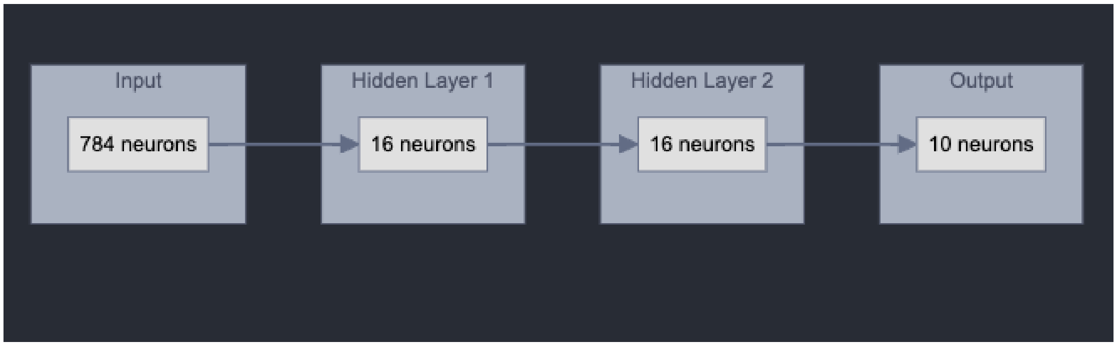
✒️ Calculation of Number of Parameters:

Consider a layer with:

1 input layer with 784 neurons (from an image of 28x28 pixels)

2 hidden layers with 16 neurons each

1 output layer with 10 neurons



Weights: (neurons of input layer x neurons of hidden layer 1 + neurons of hidden layer 1 x neurons of hidden layer 2 + neurons of hidden layer 2 x neurons of output layer)

= (784 x 16 + 16 x 16 + 16 x 10) = 12544 + 256 + 160 = 12960

Bias: (16 + 16 + 10) = 42

Total number of parameters : 12960 + 42 = 13002

📌 Keywords

✒️ Overfitting - The model learns too much from the training data, including noise and irrelevant patterns. It performs well on training data but poorly on new (test) data.

✒️ Underfitting - The model is too simple to learn from the data and fails to capture key patterns. It performs poorly on both training and test data.

📌 **Outliers**: Outliers are data points that are significantly different from the rest of the data.

In a Dataset:

Data: [10, 12, 11, 13, 1000]

Here, **1000** is an outlier because it is much larger than the other values.

📌 Normalization

- Normalization is a technique used to scale or transform data into a specific range.
- It helps in making different features (variables) comparable and improves the performance of machine learning algorithms.

✒️ Types of Normalization:

1. **Min-Max Normalization**: Scales data to a fixed range, (usually 0 to 1 or -1 to 1). Sensitive to outliers.
   x' = (x - x min)/(x max - x min)
2. **Z-Score Standardization**: Transforms data to have a mean of 0 and a standard deviation of 1.
   x'=(x - mean)/std

✒️ Why do we need normalization for multiple features? :

- **Avoids Dominance:** Ensures no feature disproportionately influences the model.
- **Speeds up Convergence:** Helps gradient descent reach the minimum faster.
- **Improves Accuracy:** Makes distance-based models more reliable.
- **Prevents Numerical Instability:** Avoids calculation errors due to large values.

✒️ Normalization with single features :

- Helps with extreme values, speeding up learning.
- Makes hyperparameter tuning easier.