ANN

Introduction to Artificial Neural Networks (ANN)

Title: Basics of Artificial Neural Networks (ANN)

Content:

- **Definition:** Artificial Neural Networks are computational models inspired by the human brain, used for tasks like classification, regression, and pattern recognition.
- **Components:**
  - **Neurons (Nodes):** Basic units of computation.
  - **Weights:** Parameters that modify input importance.
  - **Biases:** Additional parameters to adjust the output.
- **Architecture:** Consists of Input Layer, Hidden Layers, and Output Layer.

Inputs and Biases

Title: Role of Inputs and Bias in ANN

Content:

- **Inputs:** Features or data points fed into the neural network.
  - Represented as a vector: ($X = [x_1, x_2, ..., x_n]$).
- **Weights:** Multiply the inputs to assign importance.
  - (Weighted : Input = $w_1x_1 + w_2x_2 + ... + w_nx_n$)
- **Bias:** Shifts the activation function to make the model flexible.
  - Equation: ($Y = \sum w_i x_i + b$).

Activation Functions

Title: Introduction to Activation Functions

Content:

- Purpose: Determines the output of a neuron by applying a non-linear transformation.
- Common Types:
  1. **Linear:** \($f(x) = x$\)

2. **Sigmoid:** $f(x) = \frac{1}{1 + e^{-x}}$
3. **Tanh:** $f(x) = \tanh(x)$
4. **ReLU:** $f(x) = \max(0, x)$

Graphs of Activation Functions
Title: Visualization of Activation Functions

Content:

- **Linear:** Outputs a straight line.
- **Sigmoid:** S-shaped curve, outputs between 0 and 1.
- **Tanh:** Outputs between -1 and 1.
- **ReLU:** Outputs zero for negative inputs, linear for positive inputs.

Forward Propagation in ANN

Title: Understanding Forward Propagation

Content:

- **Step 1:** Calculate the weighted sum of inputs: $Z = \sum w_i x_i + b$.
- **Step 2:** Apply activation function: $A = f(Z)$.
- **Step 3:** Output is passed to the next layer.

Equation Summary:

$$
A = f\left(\sum_{i=1}^n w_i x_i + b\right)
$$

Summary

Title: Key Takeaways

Content:

- Artificial Neural Networks mimic biological neural systems.
- Inputs, weights, and biases are essential for computation.
- Activation functions introduce non-linearity for better learning.
- ANN can solve complex problems using layered computations.

In the **McCulloch-Pitts (M-P) neuron**, the weights play a critical role in determining how inputs influence the output. The weights can be classified into two main types based on their nature:

---

1. Excitatory Weights

- **Nature**: Positive weights that encourage the neuron to "fire" (produce an output of 1).
- **Effect**: If an input has an excitatory weight, it contributes positively to the weighted sum, increasing the chances of the neuron reaching or exceeding the threshold.
- **Biological Analogy**: Excitatory synapses in biological neurons increase the likelihood of a neuron firing.

Example:

- For an **AND Gate**, if inputs x1x_1 and x2x_2 both have excitatory weights w1=1,w2=1w_1 = 1, w_2 = 1, the total sum will increase when both inputs are active.

---

2. Inhibitory Weights

- **Nature**: Negative weights that prevent the neuron from firing (output = 0).
- **Effect**: If any input with an inhibitory weight is activated (value = 1), it dominates the output, forcing it to **0**, regardless of other excitatory inputs.
- **Biological Analogy**: Inhibitory synapses in biological neurons suppress or stop the firing of neurons.

Special Rule in M-P Neuron:

- The inhibitory input has **absolute dominance** over excitatory inputs.
- If an inhibitory input is active (1), the output is immediately set to 0, even if other excitatory inputs try to make the neuron fire.

Example:

- If an M-P neuron has an inhibitory weight w3=−∞w_3 = -\infty assigned to x3x_3:
  - If x3=1x_3 = 1, the output is forced to **0**, no matter the values of other inputs.

Role in Logic Gates

- **Excitatory Weights**: Used to simulate logical AND and OR operations.

  - **AND Gate**: Both inputs must contribute positively to reach the threshold.
  - **OR Gate**: One or more inputs contribute positively to meet the threshold.
- **Inhibitory Weights**: Used for NOT operations or to ensure specific inputs suppress the output.

Summary

- **Excitatory Weights**: Positive weights that encourage neuron firing.
- **Inhibitory Weights**: Dominant negative weights that suppress the output, forcing it to 0 when active.
  The combination of excitatory and inhibitory weights allows the **M-P neuron** to simulate **binary logic gates** and perform simple logical operations.
