Introduction to Feedforward Neural Network

Overview

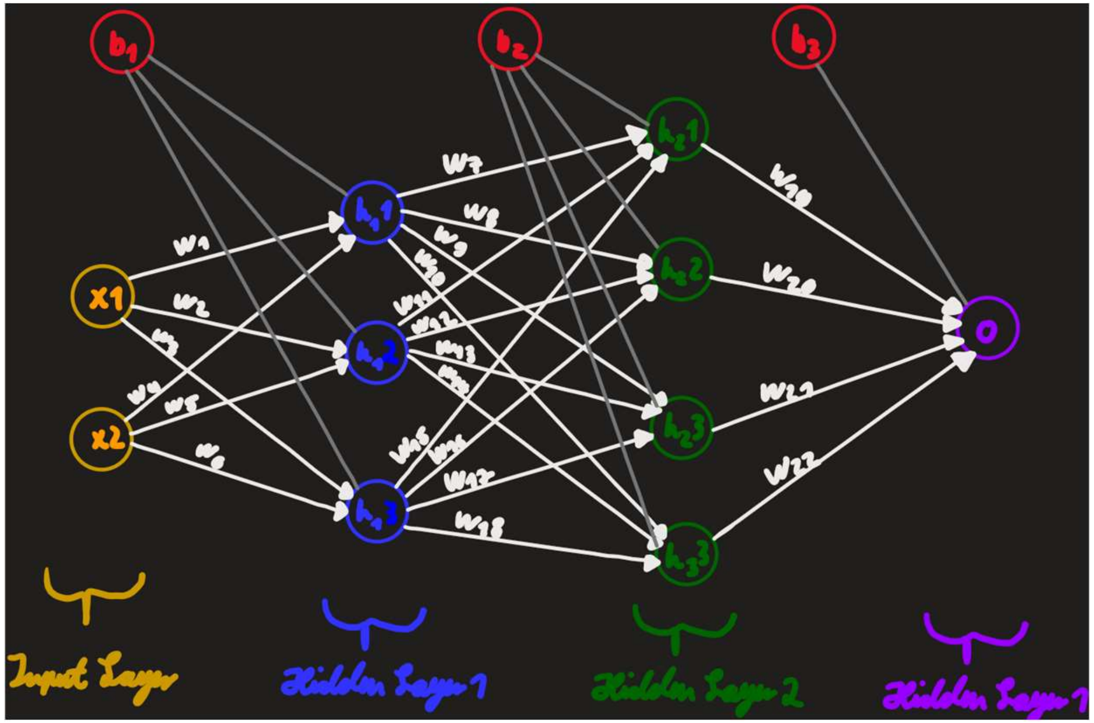
- A feedforward neural network is a type of artificial neural network where connections between nodes do not form a cycle.
- This neural network consists of an input layer, hidden layers, and an output layer.

Architecture (2, 3, 4, 1)

- **Input Layer:** 2 inputs.
- **Hidden Layers**:
  - ➢ First hidden layer with 3 neurons.
  - ➢ Second hidden layer with 4 neurons.
- **Output Layer:** 1 output neuron.

Objective

- Calculate the output of the network and compare it to a fixed target.
- Calculate the error.



The image illustrates a Neutral Network with a (2,3,4,1) Architecture

Implementing the Feed Forward Neural Network

Activation Function:

- Sigmoid Function: $\sigma(x) = \frac{1}{1+e^{x+1}}$
- Used to introduce non-linearity into the network.

Neuron Output Calculation:

- Calculate the dot Produkt of the weighted sum of inputs and biases.
- Apply the sigmoid function to the weighted sum.

Forward Propagation:

- Pass through each layer of the network.
- For each neuron in a layer:
  - Compute the dot product of input values and neuron weights.
  - Apply the sigmoid activation function.
  - Store the result to be used as input for the next layer.

Forward Propagation Example:

$x_1 = 9,\ x_2 = 3$

$w_1 = 0{,}25,\ w_2 = 0{,}35,\ w_3 = 0{,}28$

$w_4 = 0{,}15,\ w_5 = 0{,}2,\ w_6 = 0{,}4$

$w_7 = 0{,}65,\ w_{11} = 1{,}47,\ w_{15} = 1{,}72,\ w_{19} = 0{,}54$

$b_1 = 0{,}45,\ b_2 = 0{,}3,\ b_3 = 0{,}6$

$h_1 1 = w_1 \cdot x_1 + w_4 \cdot x_2 + b_1$

$h_1 1 = 0{,}25 \cdot 9 + 0{,}15 \cdot 3 + 0{,}45 = 3{,}15$

$\sigma(h_1 1) = \frac{1}{1+e^{-3{,}15}} \approx 0{,}9589$

$h_1 2 = w_2 \cdot x_1 + w_5 \cdot x_2 + b_1$

$h_1 2 = 0{,}35 \cdot 9 + 0{,}2 \cdot 3 + 0{,}45 = 4{,}2$

$\sigma(h_1 2) = \frac{1}{1+e^{-4{,}2}} \approx 0{,}9852$

$h_1 3 = w_3 \cdot x_1 + w_6 \cdot x_2 + b_1$

$h_1 3 = 0{,}28 \cdot 9 + 0{,}4 \cdot 3 + 0{,}45 = 4{,}17$

$\sigma(h_1 3) = \frac{1}{1+e^{-4{,}17}} \approx 0{,}9848$

$h_2 1 = w_7 \cdot \sigma(h_1 1) + w_{11} \cdot \sigma(h_1 2) + w_{15} \cdot \sigma(h_1 3) + b_2$

$h_2 1 = 0{,}65 \cdot 0{,}9589 + 1{,}47 \cdot 0{,}9852 + 1{,}72 \cdot 0{,}9848 + 0{,}3 \approx 5{,}0723$

$\sigma(h_2 1) = \frac{1}{1+e^{-5{,}0723}} \approx 0{,}9938$

For Simularity: $h_2 1 = h_2 2 = h_2 3 = h_2 4$

$w_{19} = w_{20} = w_{21} = w_{22}$

$o = w_{19} \cdot \sigma(h_2 1) + w_{20} \cdot \sigma(h_2 2) + w_{21} \cdot \sigma(h_2 3) + w_{22} \cdot \sigma(h_2 4) + b_3$

$o = 0{,}54 \cdot 0{,}9938 + 0{,}54 \cdot 0{,}9938 + 0{,}54 \cdot 0{,}9938 + 0{,}54 \cdot 0{,}9938 + 0{,}6$

$o = 2{,}7466$

Network Components and Forward Propagation

Random Initialization

- Weights and Biases: Initialize randomly between 0 and 1 for each neuron.

Network Implementation

- Define the architecture: [2, 3, 4, 1].
- Use nested loops to create layers and neurons with random weights and biases.

Error Calculation

- Target Output: Set a fixed target value.
- Actual Output: Obtain from forward propagation.
- Formula: I(Actual Output – Target Output)I

Algorithm Steps

- Initialize the network with random weights and biases.
- Perform forward propagation to get the network output.
- Calculate the error by comparing the output with the target.
- Repeat the process to find the set of weights and biases with the lowest error.

ANN - algorithm description