1.A perceptron has 3 inputs (x1, x2, x3) with weights (0.3, -0.1, 0.2) and bias -0.2.

Calculate the output for input vectors:

a) (1, 0, 1)

b) (0, 1, 1)

Show all steps including the activation function.

1. Perceptron Output Calculation

A perceptron computes its output using the following formula:
output = activation_function(w1 * x1 + w2 * x2 + w3 * x3 + bias)
where w1, w2, w3 are the weights, x1, x2, x3 are the inputs, and the bias is a constant.

Weights and Bias:

- Weights: w1 = 0.3, w2 = -0.1, w3 = 0.2
- Bias: b = -0.2

Activation Function: For simplicity, we will use the step function as the activation function:

activation_function(x) =

- 1 if x >= 0
- 0 if x < 0

a) Input Vector: (1, 0, 1)

1. Calculate the weighted sum: z = (0.3 * 1) + (-0.1 * 0) + (0.2 * 1) - 0.2 z = 0.3 + 0 + 0.2 - 0.2 = 0.3
2. Apply the activation function: output = activation_function(0.3) = 1

b) Input Vector: (0, 1, 1)

1. Calculate the weighted sum: z = (0.3 * 0) + (-0.1 * 1) + (0.2 * 1) - 0.2 z = 0 + (-0.1) + 0.2 - 0.2 = -0.1
2. Apply the activation function: output = activation_function(-0.1) = 0

2.Design a genetic algorithm to find the maximum value of the function $f(x) = x^2$ in the range [-10, 10]. Explain:

- ❖ Chromosome representation
- ❖ Fitness function
- ❖ Crossover and mutation operators
- ❖ Selection method

2. Genetic Algorithm Design for Maximizing $f(x) = x^2$

Chromosome Representation:

- Each chromosome can be represented as a floating-point number within the range [-10, 10]. For example, a chromosome could be represented as x.

Fitness Function:

- The fitness function evaluates how well a chromosome performs. In this case, the fitness function is simply the value of the function: $\text{fitness}(x) = f(x) = x^2$
- The goal is to maximize this value.

Crossover and Mutation Operators:

- **Crossover:** A simple one-point crossover can be used. Two parent chromosomes are selected, and a random point is chosen to exchange genetic material. For example, if parent 1 is x1 and parent 2 is x2, the offspring could be: offspring = (x1 + x2) / 2
- **Mutation:** A small random value can be added to a chromosome to introduce variability. For example: mutated_chromosome = x + random(-0.5, 0.5)

Selection Method:

- **Tournament Selection:** Randomly select a few chromosomes and choose the one with the highest fitness. This method balances exploration and exploitation.

3. For a given multilayer perceptron with one hidden layer, calculate the backpropagation updates for one training example. Show the forward pass and backward pass calculations.

3. Backpropagation in a Multilayer Perceptron

Forward Pass:

1. **Input Layer:** Assume inputs x1, x2.
2. **Hidden Layer:** Let weights be w11, w12, w21, w22 and biases b1, b2.
   - Calculate hidden layer outputs: h1 = activation(w11 * x1 + w12 * x2 + b1) h2 = activation(w21 * x1 + w22 * x2 + b2)
3. **Output Layer:** Let weights be wo1, wo2 and bias bo.
   - Calculate output: output = activation(wo1 * h1 + wo2 * h2 + bo)

Backward Pass:

1. Calculate the error at the output layer: error = target - output
2. Compute gradients for output layer weights: Δwo1 = learning_rate * error * h1 Δwo2 = learning_rate * error * h2
3. Calculate error for hidden layer: hidden_error_1 = error * wo1 * activation_derivative(h1) hidden_error_2 = error * wo2 * activation_derivative(h2)
4. Update hidden layer weights: Δw11 = learning_rate * hidden_error_1 * x1 Δw12 = learning_rate * hidden_error_1 * x2 (similar for w21, w22)

4. Compare and contrast genetic programming and genetic algorithms using a practical example from time series forecasting.

4. Genetic Programming vs. Genetic Algorithms

Genetic Algorithms (GA):

- Focus on optimization problems.
- Chromosomes represent solutions (e.g., numbers).
- Example: Finding the maximum of a function.

Genetic Programming (GP):

- Focus on evolving programs or expressions.
- Chromosomes represent code or expressions (e.g., trees).
- Example: Evolving a mathematical expression that predicts time series data.

Practical Example:

- **GA:** Optimize parameters for a forecasting model (e.g., ARIMA).
- **GP:** Evolve a mathematical expression that predicts future values based on historical data.

5. Design a single Perceptron architecture to represent the Boolean AND and OR function

5. Perceptron Architecture for Boolean AND and OR Functions

Boolean AND Function:

- Inputs: x1, x2
- Weights: w1 = 1, w2 = 1
- Bias: b = -1.5
- Output: output = activation(1 * x1 + 1 * x2 - 1.5)

Boolean OR Function:

- Inputs: x1, x2
- Weights: w1 = 1, w2 = 1
- Bias: b = -0.5
- Output: output = activation(1 * x1 + 1 * x2 - 0.5)