1) How many boolean functions can be designed for 3 inputs?

- 8
- 16
- 256
- 64

Yes, the answer is correct.
Score: 1

Accepted Answers:
256

For a Boolean function with n inputs, the number of possible functions is given by $2^{(2^n)}$. This is because each combination of inputs can have either a true or false output.

For 3 inputs:

- The number of possible input combinations is $2^3 = 8$.
- For each of these combinations, there are 2 possible output values (true or false).

Thus, the number of possible Boolean functions is:

$$2^{(2^3)} = 2^8 = 256$$

For a Boolean functi
because each combi
For 3 inputs:
- The number of

For 2 inputs:

- The number of possible input combinations is $2^2 = 4$.
- Therefore, the number of possible Boolean functions is:
$$2^{(2^2)} = 2^4 = 16$$

So, for **2 inputs**, there are **16** possible Boolean functions.

For 4 inputs:

- The number of possible input combinations is $2^4 = 16$.
- Therefore, the number of possible Boolean functions is:
$$2^{(2^4)} = 2^{16} = 65,536$$

2) Which of the following statements is(are) true about the following function?

$\sigma(z) = \frac{1}{1+e^{-(z)}}$

- [x] The function is bounded between 0 and 1
- [x] The function attains its maximum when $z \to \infty$
- [x] The function is continuously differentiable
- [x] The function is monotonic

Yes, the answer is correct.
Score: 1

Accepted Answers:
The function is bounded between 0 and 1
The function attains its maximum when $z \to \infty$
The function is continuously differentiable
The function is monotonic

1. **The function is bounded between 0 and 1:**
   - This is true. As $z \to -\infty, \sigma(z) \to 0$; as $z \to \infty, \sigma(z) \to 1$.
   - Therefore, $\sigma(z)$ is bounded between 0 and 1.
2. **The function attains its maximum when $z \to \infty$:**
   - This is also true. As $z \to \infty, \sigma(z) \to 1$, which is the maximum value of the function.
3. **The function is continuously differentiable:**
   - This is true. The sigmoid function is smooth and differentiable everywhere, with a continuous derivative given by:
     $$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$
4. **The function is monotonic:**
   - This is true. The derivative $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ is always positive for all z, meaning $\sigma(z)$ is monotonically increasing.

3) A function $f(x)$ is approximated using 250 tower functions. What is the minimum number of neurons required to construct the network that approximates the function?

- ( ) 250
- ( ) 249
- (•) 251
- ( ) 500
- ( ) 750
- ( ) 501

No, the answer is incorrect.
Score: 0

Accepted Answers:
501

For 250 tower functions, you need 250 neurons to represent each function. In a typical neural network setup, you'll also need an additional layer to combine these outputs, which typically requires another set of neurons. Therefore, you have 250 neurons for the tower functions plus an additional 250 neurons to process the outputs, plus 1 output neuron.

So the calculation is:

$$250(towerfunctions) + 250(processinglayer) + 1(output) = 501$$

4) You are training a model using the gradient descent algorithm and notice that the loss decreases and then increases after each successive epoch (pass through the data). Which of the following techniques would you employ to enhance the likelihood of the gradient descent algorithm converging? (Here, η refers to the step size.) ***1 point***

- Decrease the value of η
- Increase the value of η
- Set $\eta = 1$
- Set $\eta = 0$

Yes, the answer is correct.
Score: 1

Accepted Answers:
Decrease the value of η

1. **Decrease the value of η:**
   - This is a good approach. Lowering the learning rate can help the algorithm make smaller, more controlled updates to the model weights, which can stabilize training and help achieve convergence.
2. **Increase the value of η:**
   - This is not advisable. Increasing the learning rate may cause the model to diverge even more, as it would lead to even larger updates to the weights.
3. **Set $\eta = 1$:**
   - This is also likely to cause instability in training, as a learning rate of 1 is generally too high for most models and can lead to divergence.
4. **Set $\eta = 0$:**
   - Setting the learning rate to 0 means the model will not learn at all since there would be no weight updates.

5) Which of the following statements about the sigmoid function is NOT true?

- (selected) The derivative of the sigmoid function can be negative.
- The sigmoid function is continuous and differentiable.
- The sigmoid function maps any input value to a value between 0 and 1.
- The sigmoid function can be used as an activation function in neural networks.

Yes, the answer is correct.
Score: 1

Accepted Answers:
The derivative of the sigmoid function can be negative.

6) We have a function that we want to approximate using 150 rectangles (towers). How many neurons are required to construct the required network?

- 301
- 451
- 150
- 500

No, the answer is incorrect.
Score: 0

Accepted Answers:
301

To approximate a function using **150 rectangles (or towers)** in a neural network, we typically consider that each rectangle can be represented by a pair of neurons (for the two parts of the function defining the rectangle). Therefore, if we use 150 rectangles, the minimum number of neurons required is:

$$\text{Total neurons} = 150\ (\text{rectangles}) \times 2\ (\text{neurons per rectangle}) + 1\ (\text{output neuron}) = 301$$

Conclusion

The total number of neurons required to construct the network is **301**.

7) What happens to the output of the sigmoid function as $|x|$ becomes very large for input x?Select all relevant operations

- [ ] The output approaches 0.5
- [x] The output approaches 1.
- [ ] The output oscillates between 0 and 1.
- [x] The output approaches 0.

Yes, the answer is correct.
Score: 1

Accepted Answers:
The output approaches 1.
The output approaches 0.

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Let's analyze what happens to the output as $|x|$ becomes very large:

1. **As $x \to \infty$:**
   - When x becomes very large (positive), e^{-x} approaches 0. Therefore:

   $$\sigma(x) = \frac{1}{1+e^{-x}} \to \frac{1}{1+0} = 1$$

   - So, the output approaches **1**.
2. **As $x \to -\infty$:**
   - When x becomes very large (negative), e^{-x} approaches infinity. Thus:

   $$\sigma(x) = \frac{1}{1+e^{-x}} \to \frac{1}{1+\infty} = 0$$

   - So, the output approaches **0**.

Summary of Operations:

- The output **approaches 1** as $x \to \infty$.
- The output **approaches 0** as $x \to -\infty$.
- The output does not oscillate between 0 and 1; rather, it smoothly transitions between these values.

Relevant Options:

- **The output approaches 1.**
- **The output approaches 0.**

8) Suppose we have a function $f(x_1, x_2) = x_1^2 + 3x_2 + 25$ which we want to minimize the given function using the gradient descent algorithm. We initialize $(x_1, x_2) = (0, 0)$. What will be the value of x_1 after ten updates in the gradient descent process?(Let η be 1)

- 0
- -3
- −4.5
- −3

Yes, the answer is correct.
Score: 1
Accepted Answers:
0

Step 1: Compute the gradient

1. **Partial derivative with respect to x_1:**

$$\frac{\partial f}{\partial x_1} = 2x_1$$

2. **Partial derivative with respect to x_2:**

$$\frac{\partial f}{\partial x_2} = 3$$

Step 2: Initialize the values

We start with the initial values:

$$(x_1, x_2) = (0, 0)$$

Step 3: Update the values using gradient descent

With a learning rate $\eta = 1$, the updates are as follows:

- **Update Rule:**

$$x_1 = x_1 - \eta \cdot \frac{\partial f}{\partial x_1}$$

$$x_2 = x_2 - \eta \cdot \frac{\partial f}{\partial x_2}$$

Step 4: Iteratively apply updates

Update for x_1 and x_2:

1. **Iteration 1:**
   - Compute gradient:
   $$\frac{\partial f}{\partial x_1} = 2(0) = 0 \quad \text{and} \quad \frac{\partial f}{\partial x_2} = 3$$
   - Update:
   $$x_1 = 0 - 1 \cdot 0 = 0$$
   $$x_2 = 0 - 1 \cdot 3 = -3$$
2. **Iteration 2:**
   - Compute gradient:
   $$\frac{\partial f}{\partial x_1} = 2(0) = 0 \quad \text{and} \quad \frac{\partial f}{\partial x_2} = 3$$
   - Update:
   $$x_1 = 0 - 1 \cdot 0 = 0$$
   $$x_2 = -3 - 1 \cdot 3 = -6$$
3. **Iteration 3:**
   - Compute gradient:
   $$\frac{\partial f}{\partial x_1} = 2(0) = 0 \quad \text{and} \quad \frac{\partial f}{\partial x_2} = 3$$
   - Update:
   $$x_1 = 0 - 1 \cdot 0 = 0$$
   $$x_2 = -6 - 1 \cdot 3 = -9$$

This pattern will continue, as x_1 does not change since its gradient is always zero, while x_2 will keep decreasing by 3 with each update.

Following this pattern, after **10 iterations:**

$$x_2 = 0 - 3 \times 10 = -30$$

Final Result for x_1:

The value of x_1 remains **0** throughout the updates.

Answer:

The value of x_1 after ten updates in the gradient descent process is **0**.

9) Consider a function $f(x) = x^3 - 3x^2 + 2$. What is the updated value of x after 2nd iteration of the gradient descent update, if the learning rate is 0.1 and the initial value of x is 4?

1.792

Yes, the answer is correct.
Score: 1

Accepted Answers:
(Type: Range) 1.76,1.82

To find the updated value of x after the 2nd iteration of the gradient descent update for the function

$$f(x) = x^3 - 3x^2 + 2,$$

we need to follow these steps:

Step 1: Compute the derivative of $f(x)$

First, we calculate the derivative $f'(x)$:

$$f'(x) = \frac{d}{dx}(x^3 - 3x^2 + 2) = 3x^2 - 6x.$$

Step 2: Initialize values

- Initial value: $x_0 = 4$
- Learning rate: $\eta = 0.1$

Step 3: Perform the gradient descent updates

Iteration 1:

1. Compute $f'(x_0)$:

$$f'(4) = 3(4^2) - 6(4) = 3(16) - 24 = 48 - 24 = 24.$$

2. Update x:

$$x_1 = x_0 - \eta \cdot f'(x_0) = 4 - 0.1 \cdot 24 = 4 - 2.4 = 1.6.$$

Iteration 2:

1. Compute $f'(x_1)$:

$$f'(1.6) = 3(1.6^2) - 6(1.6) = 3(2.56) - 9.6 = 7.68 - 9.6 = -1.92.$$

2. Update x:

$$x_2 = x_1 - \eta \cdot f'(x_1) = 1.6 - 0.1 \cdot (-1.92) = 1.6 + 0.192 = 1.792.$$

Final Result

The updated value of x after the 2nd iteration of the gradient descent update is **1.792**.

10) What is the purpose of the gradient descent algorithm in machine learning?

- (selected) To minimize the loss function
- To maximize the loss function
- To minimize the output function
- To maximize the output function

Yes, the answer is correct.
Score: 1

Accepted Answers:
To minimize the loss function

Question 1:

1. How many weights does a neural network have if it consists of an input layer with 2 neurons, three hidden layers each with 4 neurons, and an output layer with 2 neurons? Assume there are no bias terms in the network.

- **Input Layer:** 2 neurons
- **Hidden Layers:** 3 hidden layers, each with 4 neurons
- **Output Layer:** 2 neurons

Step-by-Step Calculation of Weights

1. **Weights between Input Layer and First Hidden Layer:**
   - Number of weights = (Number of neurons in Input Layer) × (Number of neurons in First Hidden Layer)
   - Weights from Input to First Hidden Layer = $2 \times 4 = 8$
2. **Weights between First Hidden Layer and Second Hidden Layer:**
   - Weights from First Hidden to Second Hidden Layer = $4 \times 4 = 16$
3. **Weights between Second Hidden Layer and Third Hidden Layer:**
   - Weights from Second Hidden to Third Hidden Layer = $4 \times 4 = 16$
4. **Weights between Third Hidden Layer and Output Layer:**
   - Weights from Third Hidden to Output Layer = $4 \times 2 = 8$

Total Weights Calculation

Now, we sum all the weights calculated from each layer:

$$\text{Total Weights} = (8) + (16) + (16) + (8) = 48$$

Question 2:

Suppose we have a Multi-layer Perceptron with an input layer, one hidden layer, and an output layer. The hidden layer contains 64 perceptrons. The output layer contains one perceptron. Choose the statement(s) that are true about the network.

Options:

- The network is capable of implementing 2^64 Boolean functions
- The network is capable of implementing 2^6 Boolean functions
- Each perceptron in the hidden layer can take in only 64 Boolean inputs
- Each perceptron in the hidden layer can take in only 6 Boolean inputs

1. **The network is capable of implementing 2^{64} Boolean functions.**
   - **True:** A network with n neurons in the hidden layer can implement up to 2^n Boolean functions, assuming each neuron can be activated independently. Since there are 64 perceptrons in the hidden layer, the network can represent 2^{64} Boolean functions.

Question 3:

Consider a function f(x) = x^3 - 4x^2 + 7. What is the updated value of x after the 2nd iteration of the gradient descent update, if the learning rate is 0.1 and the initial value of x is 5?

Given Function

$$f(x) = x^3 - 4x^2 + 7.$$

Step 1: Compute the Derivative of $f(x)$

First, we calculate the derivative $f'(x)$:

$$f'(x) = \frac{d}{dx}(x^3 - 4x^2 + 7) = 3x^2 - 8x.$$

Step 2: Initialize Values

- **Initial value:** $x_0 = 5$
- **Learning rate:** $\eta = 0.1$

Step 3: Perform the Gradient Descent Updates

Iteration 1:

1. Compute $f'(x_0)$:

$$f'(5) = 3(5^2) - 8(5) = 3(25) - 40 = 75 - 40 = 35.$$

2. Update x:

$$x_1 = x_0 - \eta \cdot f'(x_0) = 5 - 0.1 \cdot 35 = 5 - 3.5 = 1.5.$$

Iteration 2:

1. Compute $f'(x_1)$:

$$f'(1.5) = 3(1.5^2) - 8(1.5) = 3(2.25) - 12 = 6.75 - 12 = -5.25.$$

2. Update x:

$$x_2 = x_1 - \eta \cdot f'(x_1) = 1.5 - 0.1 \cdot (-5.25) = 1.5 + 0.525 = 2.025.$$

To predict the labels for these input values using a sigmoid neuron, we need to calculate the sigmoid function for each input and then determine if the result is closer to 0 or 1. Usually, if the output is greater than or equal to 0.5, it's rounded to 1, otherwise, it's 0.

For simplicity, let's assume the threshold of 0.5. Given the inputs, here's how they would map:

- $x_1 = 0.72$ → Sigmoid function value will be greater than 0.5 → **1**
- $x_2 = 0.49$ → Sigmoid function value will be less than 0.5 → **0**
- $x_3 = 0.08$ → Sigmoid function value will be less than 0.5 → **0**
- $x_4 = 0.53$ → Sigmoid function value will be greater than 0.5 → **1**
- $x_5 = 0.27$ → Sigmoid function value will be less than 0.5 → **0**

So, the predicted labels in sequence are **[1, 0, 0, 1, 0]**.

$$\sigma(z) = \frac{1}{1 + e^{-(wx+b)}}$$

where w is the weight and b is the bias.

1. **Increasing the value of w decreases the slope of the sigmoid function.**
   - **False.** Increasing w increases the slope of the sigmoid function. A higher value of w makes the function steeper at the midpoint.
2. **Increasing the value of w increases the slope of the sigmoid function.**
   - **True.** As w increases, the transition from values close to 0 to values close to 1 happens more rapidly, resulting in a steeper slope.

The sigmoid function is expressed as:

$$y = \frac{1}{1 + e^{-(w_0 + w^T \cdot x)}}$$

Here:

- w_0 is the bias term.
- $w^T \cdot x$ is the dot product of weights and input features x.

Information from the Diagram

From the description:

- The blue line represents $y = 0.5$.
- The line $w^T \cdot x = 14$ indicates that the sigmoid function output will be 0.5 when $w_0 + w^T \cdot x = 0$.

Setting the Equations

The output $y = 0.5$ occurs when:

$$1 + e^{-(w_0 + w^T \cdot x)} = 2$$

This simplifies to:

$$w_0 + w^T \cdot x = 0$$

Given that $w^T \cdot x = 14$, we substitute this value:

$$w_0 + 14 = 0 \quad \Rightarrow \quad w_0 = -14$$

Evaluating the Options

1. **w0 = 14**: **False**. We found that $w_0 = -14$.
2. **w0 = -14**: **True**. This matches our calculation.
3. **w > 0**: This statement cannot be definitively determined without additional information about the weights. The value of w could be positive or negative, and we cannot conclude its sign based solely on the information given.

2. Determining the Condition for $y = 0.5$

The sigmoid function outputs $y = 0.5$ when the argument of the exponential function becomes zero. This is because:

$$y = \frac{1}{1 + e^0} = \frac{1}{1 + 1} = \frac{1}{2} = 0.5$$

3. Setting Up the Equation

For y to equal 0.5:

$$w_0 + w^T \cdot x = 0$$

This implies that the expression inside the exponent must equal zero:

$$e^{-(w_0 + w^T \cdot x)} = e^0 = 1$$

4. Given Line Condition

According to the diagram:

- The line $w^T \cdot x = 14$ indicates a specific value of $w^T \cdot x$.

Understanding Boolean Functions

For n binary inputs, the number of possible Boolean functions is given by 2^{2^n}. In the case of 4 inputs:

$$n = 4 \quad \Rightarrow \quad \text{Number of Boolean functions} = 2^{2^4} = 2^{16} = 65,536$$

Calculating Neurons Needed

For 4 inputs:

$$2^4 = 16$$

Thus, **16 neurons** in the hidden layer are sufficient to represent any Boolean function with 4 inputs.