1) Which of the following activation functions is not zero-centered?

- [x] Sigmoid
- [ ] Tanh
- [ ] ReLU
- [x] Softmax

Partially Correct.
Score: 0.67

Accepted Answers:

Sigmoid

ReLU

Softmax

- **Sigmoid:** Outputs values in the range of (0, 1), so it is not zero-centered.
- **ReLU:** Outputs values in the range of [0, ∞), thus it is definitely not zero-centered.
- **Softmax:** Outputs a probability distribution across classes, where all outputs are between (0, 1), making it not zero-centered as well.

Considering this, the activation functions that are not zero-centered are indeed **Sigmoid, ReLU,** and **Softmax.** Thank you for providing the accepted answers!

2) We have observed that the sigmoid neuron has become saturated. What might be the possible output values at this neuron?

- [x] 0.02
- [ ] 0.5
- [ ] 1
- [x] 0.97

Partially Correct.
Score: 0.67

Accepted Answers:

0.02

1

0.97

If a sigmoid neuron is saturated, it typically means that the inputs to the neuron are either very high or very low, resulting in outputs that are close to the extremes of the sigmoid function (0 and 1).

Accepted Values for a Saturated Sigmoid Neuron:

1. **0.02** - Indicates that the neuron is likely saturated and producing an output close to 0 due to a very negative input.
2. **1** - Indicates that the neuron is strongly saturated and producing an output near 1 due to a very positive input.
3. **0.97** - Indicates that the neuron is nearly saturated on the positive side, producing an output very close to 1.

Summary:

So, the final accepted outputs for a saturated sigmoid neuron are **0.02**, **1**, and **0.97**. Thank you for your patience!

3) What is the gradient of the sigmoid function at saturation?

0

Yes, the answer is correct.
Score: 1

Accepted Answers:
(Type: Numeric) 0

Gradient of the Sigmoid Function

The derivative (gradient) of the sigmoid function can be calculated as follows:

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$$

Behavior at Saturation

When the sigmoid function is saturated, it means that the output values are close to 0 or 1, often corresponding to very negative or very positive input values respectively.

- If $\sigma(x)$ is close to **0** (when x is a very large negative value), then:
  - $\sigma'(x) \approx 0 \cdot (1 - 0) = 0$
- If $\sigma(x)$ is close to **1** (when x is a very large positive value), then:
  - $\sigma'(x) \approx 1 \cdot (1 - 1) = 0$

Conclusion

At saturation (when the outputs are very close to either 0 or 1), the gradient of the sigmoid function approaches:

- **0**

1. Tanh (Hyperbolic Tangent)

The tanh function is defined as:

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

or it can similarly be expressed in terms of the sigmoid function:

$$\tanh(x) = 2\sigma(2x) - 1$$

The derivative of the tanh function is given by:

$$\tanh'(x) = 1 - \tanh^2(x)$$

Behavior at Saturation

- When x is a very large positive value, $\tanh(x) \approx 1$:

$$\tanh'(x) \approx 1 - 1^2 = 0$$

- When x is a very large negative value, $\tanh(x) \approx -1$:

$$\tanh'(x) \approx 1 - (-1)^2 = 0$$

Conclusion for Tanh

At saturation, the gradient of **tanh** approximates **0**.

2. ReLU (Rectified Linear Unit)

The ReLU function is defined as:

$$\text{ReLU}(x) = \max(0, x)$$

The derivative of the ReLU function is given by:

$$\text{ReLU}'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

Behavior at Saturation

- For $x < 0$ (saturation at 0):

$$\text{ReLU}'(x) = 0$$

- For $x \geq 0$ (not fully saturated, gradient is 1):

$$\text{ReLU}'(x) = 1$$

Conclusion for ReLU

At saturation (for $x < 0$), the gradient of **ReLU** is **0**. For $x \geq 0$, the gradient is **1**, meaning it is not saturated in the positive region.

---

3. Softmax

The softmax function is defined for a vector $\mathbf{z}$ as:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

The gradients for softmax are given by:

$$\text{softmax}'(z_i) = \text{softmax}(z_i) \cdot (1 - \text{softmax}(z_i)) \quad \text{for } i = j$$

$$\text{softmax}'(z_i) = -\text{softmax}(z_i) \cdot \text{softmax}(z_j) \quad \text{for } i \neq j$$

Behavior at Saturation

- When z_i is the largest value in the vector (indicating perfect confidence, output approaching 1 for that class), then:

$$\text{softmax}(z_i) \approx 1 \quad \Rightarrow \quad \text{softmax}'(z_i) \approx 1 \cdot (1 - 1) = 0$$

- When z_i is not the largest, it outputs values close to 0.

Conclusion for Softmax

At saturation (when one output is very close to 1), the gradient of **Softmax** for that output approximates **0**.

5) How does pre-training prevent overfitting in deep networks?

- [x] It adds regularization
- [x] It initializes the weights near local minima
- [ ] It constrains the weights to a certain region
- [ ] It eliminates the need for fine-tuning

No, the answer is incorrect.
Score: 0

Accepted Answers:

It adds regularization

It constrains the weights to a certain region

1. It adds regularization:

Pre-training can be viewed as a form of implicit regularization. By training on a larger dataset first, the model may learn more generalized features that are less likely to overfit to the smaller task-specific dataset. This broader understanding can make the model more robust and less sensitive to noise in the smaller dataset.

2. It constrains the weights to a certain region:

During pre-training, weights are initialized and adjusted based on a large dataset. This process can effectively constrain the weights to a region that is more likely to yield good generalization performance. When the model is fine-tuned on a smaller dataset, it starts from this well-constrained region rather than arbitrary or random initialization, which can help avoid extreme weight configurations that could lead to overfitting.

6) We train a feed-forward neural network and notice that all the weights for a particular neuron are equal. What could be the possible causes of this issue?

- [ ] Weights were initialized randomly
- [ ] Weights were initialized to high values
- [x] Weights were initialized to equal values
- [ ] Weights were initialized to zero

Partially Correct.
Score: 0.5

Accepted Answers:
Weights were initialized to equal values
Weights were initialized to zero

Which of the following methods can help to avoid saturation in deep learning?

Using a different activation function.
Increasing the learning rate.
Increasing the model complexity
All of the above.

Yes, the answer is correct.
Score: 1

Accepted Answers:
Using a different activation function.

1. Using a different activation function:

Some activation functions, like ReLU and its variants (Leaky ReLU, Parametric ReLU), are less prone to saturation compared to sigmoid and tanh. Therefore, using a different activation function can help mitigate saturation issues.

8) Which of the following is true about the role of unsupervised pre-training in deep learning?

- ○ It is used to replace the need for labeled data
- ◉ It is used to initialize the weights of a deep neural network
- ○ It is used to fine-tune a pre-trained model
- ○ It is only useful for small datasets

Yes, the answer is correct.
Score: 1

Accepted Answers:
It is used to initialize the weights of a deep neural network

9) Which of the following is an advantage of unsupervised pre-training in deep learning?

- [x] It helps in reducing overfitting
- [x] Pre-trained models converge faster
- [x] It improves the accuracy of the model
- [ ] It requires fewer computational resources

Yes, the answer is correct.
Score: 1

Accepted Answers:
It helps in reducing overfitting
Pre-trained models converge faster
It improves the accuracy of the model

1. **Compute the Weighted Sum (Net Input)**:

$$z = w_1 \cdot x_1 + w_2 \cdot x_2$$

Plugging in the given values:

$$z = (1.5 \cdot 0.2) + (0.5 \cdot -0.5)$$

$$z = 0.3 - 0.25 = 0.05$$

2. **Apply the ReLU Activation Function**:
   The ReLU function is defined as:

$$\text{ReLU}(z) = \max(0, z)$$

So, applying ReLU to our computed net input:

$$\text{output} = \text{ReLU}(0.05) = \max(0, 0.05) = 0.05$$

Conclusion:

The output of the ReLU neuron is **0.05**.

Question:

Which of the following best describes the concept of saturation in deep learning?

Options:

- When the activation function output approaches either 0 or 1 and the gradient is close to zero.
- When the activation function output is very small and the gradient is close to zero.
- When the activation function output is very large and the gradient is close to zero.
- None of the above.

The correct answer is:
When the activation function output approaches either 0 or 1 and the gradient is close to zero.

In Batch Normalization, which parameter is learned during training?

Options:

- Mean
- Variance
- γ

- ε

During the training of Batch Normalization, the parameters **γ (gamma)** and **β (beta)** are learned.

- **Mean** and **Variance** are calculated on the fly for each mini-batch.
- **ε (epsilon)** is a small constant added for numerical stability.

So, the parameter that is actually learned during training is **γ**. How are you finding Batch Normalization so far?

Question 5:

What is the main cause of the Dead ReLU problem in deep learning?

- High variance
- **High negative bias**
- Overfitting
- Underfitting

High negative bias.

Explanation:

- The Dead ReLU problem occurs when a ReLU neuron output becomes stuck at zero. This typically happens when the weights are initialized in such a way that a significant portion of inputs to the neuron are negative, leading to consistently zero outputs. This is often associated with high negative bias.

Question 6:

How can you tell if your network is suffering from the Dead ReLU problem?

- The loss function is not decreasing during training
- The accuracy of the network is not improving
- **A large number of neurons have zero output**
- The network is overfitting to the training data

A large number of neurons have zero output.

Explanation:

- If a significant number of neurons are outputting zero, it indicates that these neurons are effectively "dead" and not contributing to the learning process. This is a direct sign of the Dead ReLU problem and suggests that these neurons are not being activated due to the specific input distributions and weight configurations.