1. Is it okay to initialize all the weights to the same value as long as that value is selected randomly using He initialization?

🡪Yes, it is okay to initialize all the weights to the same value as long as that value is selected randomly using He initialization. He initialization is a specific type of random initialization technique that ensures that the weights are initialized with a good balance of small and large values, which helps to optimize the learning process.

1. Is it okay to initialize the bias terms to 0?

🡪Yes, it is okay to initialize the bias terms to zero. Initializing the bias terms to zero is a popular approach as it allows the model to learn the bias more quickly during training.

1. Name three advantages of the ELU activation function over ReLU.

🡪1. The ELU activation function has a non-zero mean, which helps to reduce the vanishing gradients problem.

2. The ELU activation function has a smoother transition from negative to positive values, which can help to improve the training process.

3. The ELU activation function has a higher maximum output than ReLU, which can help to improve the model's performance.

1. In which cases would you want to use each of the following activation functions: ELU, leaky ReLU (and its variants), ReLU, tanh, logistic, and softmax?

🡪ELU: ELU is best used in deep learning networks where the vanishing gradient problem is a concern.

Leaky ReLU (and its variants): Leaky ReLU and its variants are best used in networks where the vanishing gradient problem is a concern, but where the non-zero mean of ELU is not desired.

ReLU: ReLU is best used in networks where the vanishing gradient problem is not a concern and where a non-zero mean is not desired.

Tanh: Tanh is best used in networks where a symmetric activation is desired and where the vanishing gradient problem is not a concern.

Logistic: Logistic is best used in networks where a sigmoid activation is desired and where the vanishing gradient problem is not a concern.

Softmax: Softmax is best used in networks where a probability distribution across multiple classes is desired.

1. What may happen if you set the momentum hyperparameter too close to 1 (e.g., 0.99999) when using a MomentumOptimizer?

🡪If the momentum hyperparameter is set too close to 1 (e.g., 0.99999), the model may not be able to make any updates to the weights during the training process as the gradients will be too small. This can lead to poor model performance and longer training times.

1. Name three ways you can produce a sparse model.

🡪1. Use a weight regularization technique such as L1 regularization, which encourages the model to set the weights for less important features to zero.

2. Use a pruning technique such as magnitude pruning, which removes weights with small magnitudes from the model.

3. Use a model architecture that produces sparse models such as a convolutional neural network, which has fewer parameters due to the shared weights across different convolutional layers.

1. Does dropout slow down training? Does it slow down inference (i.e., making predictions on new instances)?

🡪Yes, dropout can slow down training. This is due to the fact that the model has to recompute the forward and backward passes for each iteration, which takes more time than if dropout was not used.

No, dropout does not slow down inference. This is because dropout is only used during the training process and is not used during inference.