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

It is generally not advisable to initialize all weights to the same value, even if that value is chosen using He initialization. When all weights are the same, neurons in the same layer will have the same gradients and update in the same way during training, which can lead to symmetrical weights and slow convergence. Weight diversity is important for the network's capacity to learn different features.

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

Initializing bias terms to 0 is a common practice and is generally fine. However, if you encounter issues with training or convergence, you can try initializing bias terms to small positive values to break the symmetry between neurons. But in most cases, initializing them to 0 is acceptable.

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

Advantages of the ELU (Exponential Linear Unit) activation function over ReLU: a. ELU avoids the "dying ReLU" problem by having negative values for certain inputs. This helps prevent neurons from becoming inactive during training. b. ELU is differentiable for all values, which simplifies the training process and can lead to faster convergence. c. ELU provides better performance for deeper neural networks in some cases, as it mitigates the vanishing gradient problem.

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?

Use of activation functions:

* ELU: Use ELU as a general-purpose activation function for hidden layers in deep neural networks. It can help mitigate the vanishing gradient problem.
* Leaky ReLU (and variants): Use Leaky ReLU when you want to prevent some neurons from being completely inactive. It helps with the dying ReLU problem.
* ReLU: ReLU is commonly used as a default activation function for hidden layers. It's simple and computationally efficient.
* Tanh: Use tanh when you need a centered activation function. It's often used in recurrent neural networks.
* Logistic (Sigmoid): Use logistic activation in binary classification problems as the output layer.
* Softmax: Use softmax as the output activation function for multi-class classification problems to obtain class probabilities.

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) when using a MomentumOptimizer, it can lead to oscillations and instability during training. This is because the optimizer relies too heavily on previous gradients, making it difficult to reach convergence. It's best to set momentum to a reasonable value between 0 and 1 (e.g., 0.9).

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

Three ways to produce a sparse model: a. Weight Pruning: Remove small-weight connections, making the model's parameters sparse. b. Sparse Activations: Use activation functions like ReLU or Leaky ReLU that can produce sparse activations by setting negative values to zero. c. L1 Regularization: Apply L1 regularization to the model's weights, encouraging some weights to become exactly zero.

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

Dropout can slow down training because it temporarily deactivates a random subset of neurons during each training iteration. This effectively reduces the model's capacity and requires more training iterations for the same level of convergence. However, dropout is not typically used during inference, so it does not slow down inference. Dropout is applied only during training to improve model generalization.