1. Is it okay to initialize all the weights to the same value as long as that value is selected

randomly using He initialization?

**Initializing all weights to the same value, even when using He initialization, is not recommended. He initialization is designed to provide a good initial spread of weights to prevent the vanishing gradient problem. Initializing all weights to the same value would defeat this purpose.**

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

**Initializing bias terms to 0 is generally acceptable. Bias terms are often initialized to 0, and the network can learn appropriate bias values during training. However, in some cases, you may choose to initialize biases differently based on the problem or network architecture.**

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

**The advantages of the Exponential Linear Unit (ELU) activation function over Rectified Linear Unit (ReLU) include:**

**Handles the dying ReLU problem better: ELU has a non-zero gradient for negative inputs, reducing the likelihood of neurons becoming inactive during training.**

**Smoother gradients: ELU provides smooth gradients everywhere, which can lead to more stable training.**

**Improved model performance: ELU has been observed to yield better results in some cases compared to ReLU.**

4. 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 ELU or variants:**

**When you want to mitigate the vanishing gradient problem.**

**When you want to benefit from smooth gradients and potentially faster convergence.**

**Use Leaky ReLU or variants:**

**When you want a simple activation function that addresses the dying ReLU problem.**

**When you have limited resources for experimentation.**

**Use ReLU:**

**When you want a computationally efficient activation function that works well in many cases.**

**As the default choice for many architectures.**

**Use tanh:**

**When you need outputs in the range of [-1, 1].**

**In recurrent neural networks (RNNs) and some hidden layers.**

**Use logistic (sigmoid):**

**In binary classification output layers.**

**Use softmax:**

**In multi-class classification output layers.**

5. What may happen if you set the momentum hyperparameter too close to 1 (e.g., 0.99999)

when using a MomentumOptimizer?

**Setting the momentum hyperparameter too close to 1 (e.g., 0.99999) in a MomentumOptimizer can lead to excessively large and unstable updates in gradient descent. It can result in overshooting the optimal solution and difficulty in converging to a good model.**

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

**Three ways to produce a sparse model are:**

**Weight Pruning: Identifying and setting small weights in the neural network to zero.**

**Sparse Activation Functions: Using activation functions like the sparse version of ReLU (e.g., Leaky ReLU) to encourage sparsity in activations.**

**Regularization Techniques: Applying regularization techniques like L1 regularization, which encourages the model to have many zero-valued weights**

7. Does dropout slow down training? Does it slow down inference (i.e., making predictions on

new instances)?

**Dropout can slow down training because it randomly deactivates a fraction of neurons during each training step, requiring more iterations for convergence.**

**During inference (making predictions on new instances), dropout is typically turned off, so it does not affect inference speed. However, you may use a variant called Monte Carlo Dropout (MC Dropout) during inference for uncertainty estimation, which can be slower due to multiple forward passes.**