Q1: Is it OK to initialize all the weights to the same value as long as that value is selected randomly using He initialization?

A: It is generally not recommended to initialize all weights to the same value, even if it's a random value selected using He initialization. Initializing all weights to the same value can result in symmetrical gradients, which can lead to the model getting stuck in a suboptimal solution.

Q2: Is it OK to initialize the bias terms to 0?

A: It is generally OK to initialize the bias terms to 0. This is because the bias terms are typically updated during training, so their initial values don't have a significant impact on the final performance of the model.

Q3: Name three advantages of the SELU activation function over ReLU.

A: Three advantages of the SELU activation function over ReLU are: 1) it can achieve better performance on deep neural networks, 2) it is self-normalizing, which means it can maintain mean and variance of activations as they propagate through the network, and 3) it can prevent vanishing gradients and exploding gradients.

Q4: In which cases would you want to use each of the following activation functions: SELU, leaky ReLU (and its variants), ReLU, tanh, logistic, and softmax?

A: The SELU activation function is typically used in deep neural networks with many layers. Leaky ReLU and its variants are useful when ReLU leads to dead neurons. ReLU is a popular choice for most neural networks. Tanh and logistic functions are used in the output layer for binary classification problems. Softmax is used in the output layer for multiclass classification problems.

Q5: What may happen if you set the momentum hyperparameter too close to 1 (e.g., 0.99999) when using an SGD optimizer?

A: Setting the momentum hyperparameter too close to 1 can cause the optimizer to overshoot the minimum of the cost function, resulting in unstable training and slower convergence.

Q6: Name three ways you can produce a sparse model.

A: Three ways to produce a sparse model are: 1) L1 regularization, which encourages weights to be zero, 2) dropout, which randomly sets activations to zero during training, and 3) pruning, which removes connections with low weight magnitudes.

Q7: Does dropout slow down training? Does it slow down inference (i.e., making predictions on new instances)? What about MC Dropout?

A: Dropout can slow down training because it adds noise to the activations and requires the model to be trained for longer. However, dropout does not slow down inference because it is only applied during training. MC Dropout, which is a variation of dropout that is used during inference, can slow down inference because it requires the model to be run multiple times to obtain a probabilistic estimate of the output.