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

**Ans: No, all weights should be sampled independently; they should not all have the same initial value. An important purpose of random sampling weights is to break symmetry:**

**If all weights have the same initial value, even if the value is not zero, then the symmetry will not be broken (that is, all neurons in a given layer are equivalent), and backpropagation will not be able to break it .**

**Specifically, this means that all neurons in any given layer will always have the same weight. It's like only one neuron per layer, and it's much slower. It is almost impossible for this configuration to converge to a good solution.**

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

**Ans : Initializing the bias term to zero is completely correct. Some people like to initialize them like weights, which is okay; it doesn't make much difference.**

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

**Ans : After the SELU activation function, the sample distribution is automatically normalized to 0 mean and unit variance (self-normalization, to ensure that the gradient will not explode or disappear during the training process, and the effect is better than Batch Normalization)**

**The relu activation function has a gentle slope on the negative semi-axis, so that when the activation variance is too large, it can be reduced to prevent gradient explosion, but the positive semi-axis slope is simply set to 1.**

**The positive semi-axis of selu is greater than 1, which can be increased when the variance is too small, while preventing the gradient from disappearing. In this way, the activation function has a fixed point. After the network is deep, the output of each layer has a mean value of 0 and a variance of 1.**

1. 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?

**Ans : After SELU passes the activation function, the sample distribution is automatically normalized to 0 mean and unit variance.**

**If you need to use neural networks as fast as possible, you can use one of the variants leaky ReLU**

**(For example, simple leaky ReLU with default hyperparameter values).**

**The simplicity of the ReLU activation function makes it the preferred option for many people, although they are generally superior to ELU and leaky ReLU. However, in some cases, the ability of the ReLU activation function to output exactly zero may be useful.**

**If you need to output a number between -1 and 1, the hyperbolic tangent (tanh) is useful in the output layer, but now it is not used much in the hidden layer.**

**When you need to estimate the probability, the logical activation function is also useful in the output layer (for example, for binary classification), but it is also rarely used in the hidden layer.**

**The softmax activation function is used in the output layer to output the probability of mutually exclusive classes, but other than that, it is rarely (if ever) used in the hidden layer.**

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

**Ans : The algorithm may gain a lot of speed, hoping to roughly reach the global minimum, but due to its momentum, it will overshoot after the minimum. Then it will slow down and come back, speed up again, overshoot again, etc.**

**It may oscillate in this way many times before converging, so overall the time required to converge is much longer than using a smaller momentum value.**

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

**Ans : One method is to train the model normally and then zero the tiny weights.**

**You can apply l1 regularization during training to push the optimizer to sparsity.**

**Use TensorFlow's FTRLOptimizer class to combine l1 regularization with dual averaging.**

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

**Ans : Dropout will slow down the training speed, generally speaking, about twice. However, it has no effect on the prediction because it is only turned on during training.**

**The number of Monte Carlo samples is a hyperparameter that can be adjusted. The higher the number, the higher the estimate of prediction and inaccuracy. However, if the number of samples doubles, the inference time also doubles.**

**In addition, if the number of samples exceeds a certain number, the improvement is not large. Therefore, it depends on the task itself, making a trade-off between delay and accuracy.**