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

No.Because Initializing all the weights with zeros leads the neurons to learn the same features during training. In fact, any constant initialization scheme will perform very poorly.

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

It is possible and common to initialize the biases to be zero, since the asymmetry breaking is provided by the small random numbers in the weights

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

Like ReLU, **SELU does not have vanishing gradient problem** and hence, is used in deep neural networks. Compared to ReLUs, SELUs cannot die. SELUs learn faster and better than other activation functions without needing further procession.

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?**

Some of the activation functions are never used for solving real-world problems. They are just for knowledge. Most of the time for hidden layers we use relu and its variants (**SELU, leaky ReLU ...**)and for the final layer, we use softmax or linear function depending upon the type of problem.

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

If you set the momentum hyperparameter too close to 1 (e.g., 0.99999) when using an SGD optimizer, then **the algorithm will likely pick up a lot of speed, hopefully moving roughly toward the global minimum, but its momentum will carry it right past the minimum**.

1. **Name three ways you can produce a sparse model**.
2. **Does dropout slow down training? Does it slow down inference (i.e., making predictions on new instances)? What about MC Dropout?**

* Logically, by omitting at each iteration neurons with a dropout, those omitted on an iteration are not updated during the backpropagation. They do not exist. So **the training phase is slowed down**.
* Concisely, MC-dropout is a method of performing multiple stochastic forward passes with the means of activated dropout in a neural network during the testing process to provide ensemble of predictions that could reflect uncertainty estimations.

1. Practice training a deep neural network on the CIFAR10 image dataset:
   1. Build a DNN with 20 hidden layers of 100 neurons each (that’s too many, but it’s the point of this exercise). Use He initialization and the ELU activation function.
   2. Using Nadam optimization and early stopping, train the network on the CIFAR10 dataset. You can load it with keras.datasets.cifar10.load\_​data(). The dataset is composed of 60,000 32 × 32–pixel color images (50,000 for training, 10,000 for testing) with 10 classes, so you’ll need a softmax output layer with 10 neurons. Remember to search for the right learning rate each time you change the model’s architecture or hyperparameters.
   3. Now try adding Batch Normalization and compare the learning curves: Is it converging faster than before? Does it produce a better model? How does it affect training speed?
   4. Try replacing Batch Normalization with SELU, and make the necessary adjustements to ensure the network self-normalizes (i.e., standardize the input features, use LeCun normal initialization, make sure the DNN contains only a sequence of dense layers, etc.).
   5. Try regularizing the model with alpha dropout. Then, without retraining your model, see if you can achieve better accuracy using MC Dropout.