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:

It makes the hidden units symmetric and this problem is known as the symmetry problem. Hence to break this symmetry the weights connected to the same neuron should not be initialized to the same value. Never initialize all the weights to zero. Never initialize all the weights to the same value

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

ANS:

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.

ANS:

Advantages of SELU

* 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?

ANS:

Advantages of SELU  
  
Compared to ReLUs, SELUs cannot die. SELUs learn faster and better than other activation functions without needing further procession. Moreover, other activation function combined with batch normalization cannot compete with SELUs.

Leaky ReLU is a modification of the ReLU activation function. It has the same form as the ReLU, but it will leak some positive values to 0 if they are close enough to zero. it is a variant of the ReLU activation function.

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

ANS:

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.

ANS:

* ` regression.
* machine-learning.
* lasso.
* regularization.
* ridge-regression.

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

ANS:

In general by a factor of two. Does dropout slow down inference (. i.e, making predictions on new instances)? No, since it is only turned on during training.

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.

ANS:

shallow neural network

As described in the prior Q&A, a shallow neural network has only one (or just a few) hidden layers between the input and output layers.

* 1. 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.

ANS:

First, we will import torch.

1. import torch. Then we will import torchvision.
2. import torchvision. Torchvision is a package in the PyTorch library containing computer-vision models, datasets, and image transformations. ...
3. import torchvision.datasets as datasets.
   1. 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?

ANS:

BatchNorm impacts network training in a fundamental way: it makes the landscape of the corresponding optimization problem be significantly more smooth. This ensures, in particular, that the gradients are more predictive and thus allow for use of larger range of learning rates and faster network convergence

* 1. 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.).

ANS:

The batch normalization can be applied before and after the activation function. However, research shows its best when applied before the activation function

* 1. Try regularizing the model with alpha dropout. Then, without retraining your model, see if you can achieve better accuracy using MC Dropout.

ANS:

With dropout (dropout rate less than some small value), the accuracy will gradually increase and loss will gradually decrease first(That is what is happening in your case). When you increase dropout beyond a certain threshold, it results in the model not being able to fit properly.