1. Is it okay to initialize all the weights to the same value as long as that value is selected randomly using He initialization?
2. Is it okay to initialize the bias terms to 0?
3. Name three advantages of the ELU activation function over 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?
5. What may happen if you set the momentum hyperparameter too close to 1 (e.g., 0.99999) when using a MomentumOptimizer?
6. Name three ways you can produce a sparse model.
7. Does dropout slow down training? Does it slow down inference (i.e., making predictions on new instances)?

Answer:

1. Initializing all weights to the same value using He initialization is generally not recommended, even if the value is selected randomly. This is because the weights will have similar gradients and will update in a similar way during training, which can result in the network being stuck in a suboptimal solution. He initialization ensures that the initial weights have different values and appropriate variances, which helps prevent this problem.
2. Yes, it is generally okay to initialize the bias terms to 0. In fact, this is often the default initialization used in many deep learning libraries.
3. Three advantages of the ELU activation function over ReLU are:

* ELU can take negative values, which can help alleviate the vanishing gradient problem.
* ELU has a nonzero mean, which helps the network converge faster.
* ELU is smooth everywhere, which can help gradient-based optimization algorithms work better.

1. Here are some recommendations for when to use each activation function:

* ELU: Use ELU as a default activation function. It generally works well in most cases and has some advantages over ReLU.
* Leaky ReLU (and its variants): Use leaky ReLU when you want to avoid the "dying ReLU" problem, where ReLU units can become permanently inactive. Use PReLU, RReLU, or LReLU variants if you want to add more flexibility.
* ReLU: Use ReLU as a default activation function if you don't need the advantages of ELU or leaky ReLU and want a simpler function.
* Tanh: Use tanh if you need a symmetric activation function that can take both positive and negative values.
* Logistic: Use logistic if you need an activation function that can output probabilities between 0 and 1.
* Softmax: Use softmax in the output layer if you need to classify multiple classes and want to output class probabilities.

1. If the momentum hyperparameter is set too close to 1, the optimizer may oscillate around the minimum or diverge, especially if the learning rate is too high. This can slow down training or prevent the network from converging to a good solution.
2. Three ways to produce a sparse model are:

* L1 regularization: Add an L1 penalty to the cost function, which encourages the weights to be close to 0 and makes the model more sparse.
* Dropout: Randomly set some activations to 0 during training, which forces the network to learn redundant representations and can lead to a sparser model.
* Pruning: Remove some weights or neurons from the network after training, based on their importance or contribution to the network's performance.

1. Dropout can slow down training because the network needs to learn redundant representations and needs more iterations to converge. However, it can also prevent overfitting and improve the generalization performance of the network. Dropout does not slow down inference, because it is only used during training and the full network is used during inference.