Chapter 11

Textbook Exercises

What is the problem that Glorot/Xavier initialization and He initialization aim to fix?

What is the problem that Glorot/Xavier initialization and He initialization aim to fix?

Glorot/Xavier initialization and He initialization were designed to make the output standard deviation as close as possible to the input standard deviation, at least at the beginning of training. This reduces the vanishing/exploding gradients problem.

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

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

No, all weights should be sampled independently; they should not all have the same initial value. One important goal of sampling weights randomly is to break symmetry. Concretely, this means that all the neurons in any given layer will always have the same weights. It's like having just one neuron per layer, and much slower. It is virtually impossible for such a configuration to converge to a good solution.

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

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. Then it will slow down and come back, accelerate again, overshoot again, and so on. It may oscillate this way many times before converging, so overall it will take much longer to converge than with a smaller `momentum` value.

Does dropout slow down training? Does it slow down inference (i.e., making predictions on new instances)?

Does dropout slow down training? Does it slow down inference (i.e., making predictions on new instances)?

Yes, dropout does slow down training, in general roughly by a factor of two. However, it has no impact on inference speed since it is only turned on during training.

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

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

The SELU activation function is a good default. If you need the neural network to be as fast as possible, you can use the leaky ReLU. The simplicity of the ReLU activation function makes it many people's preferred option. The hyperbolic tangent (tanh) can be useful in the output layer if you need to output a number between –1 and 1. The logistic activation function is also useful in the output layer when you need to estimate a probability for binary classification. Finally, the softmax activation function is useful in the output layer to output probabilities for mutually exclusive classes.