

Chapter 15

Dropout

Another option for neural network regularization is adding a **dropout layer**. This type of layer disables some neurons, while the others pass through unchanged. The idea here, similarly to regularization, is to prevent a neural network from becoming too dependent on any neuron or for any neuron to be relied upon entirely in a specific instance (which can be common if a model overfits the training data). Another problem dropout can help with is **co-adoption**, which happens when neurons depend on the output values of other neurons and do not learn the underlying function on their own. Dropout can also help with **noise** and other perturbations in the training data as more neurons working together mean that the model can learn more complex functions.

The Dropout function works by randomly disabling neurons at a given rate during every forward pass, forcing the network to learn how to make accurate predictions with only a random part of neurons remaining. Dropout forces the model to use more neurons for the same purpose, resulting in a higher chance of learning the underlying function that describes the data. For example, if we disable one half of the neurons during the current step, and the other half during the next step, we are forcing more neurons to learn the data, as only a part of them “sees” the data and gets updates in a given pass. These alternating halves of neurons are an example, and in reality, we’ll use a hyperparameter to inform the dropout layer of the number of neurons to disable randomly.