CHAPTER 29

Training a Neural Network

This chapter gives an overview of the techniques for training a deep neural network. Here, we briefly discuss

- How learned information flows through a neural network
- The role of the cost function at the output layer of the network
- One-hot encoding and the softmax activation function for determining class membership at the output layer of a classification problem
- The backpropagation algorithm for improving the learned parameters of the network
- Activation functions that enable the neural network to learn non-linear patterns

In this chapter, as we discuss the methods involved in training a neural network, we will use the example of a classification problem with two possible outputs. In designing a neural network, the number of neurons in the input layer is typically the number of features of the dataset, while the number of neurons in the output layer is the number of classes in the target variable that the neural network is learning to classify.

As illustrated in Figure 29-1, the dataset features are the inputs to the neural network, while the classes in the target variable determine the number of output neurons. In this example, the network learns two classes, 0 and 1.

E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*,
https://doi.org/10.1007/978-1-4842-4470-8_29

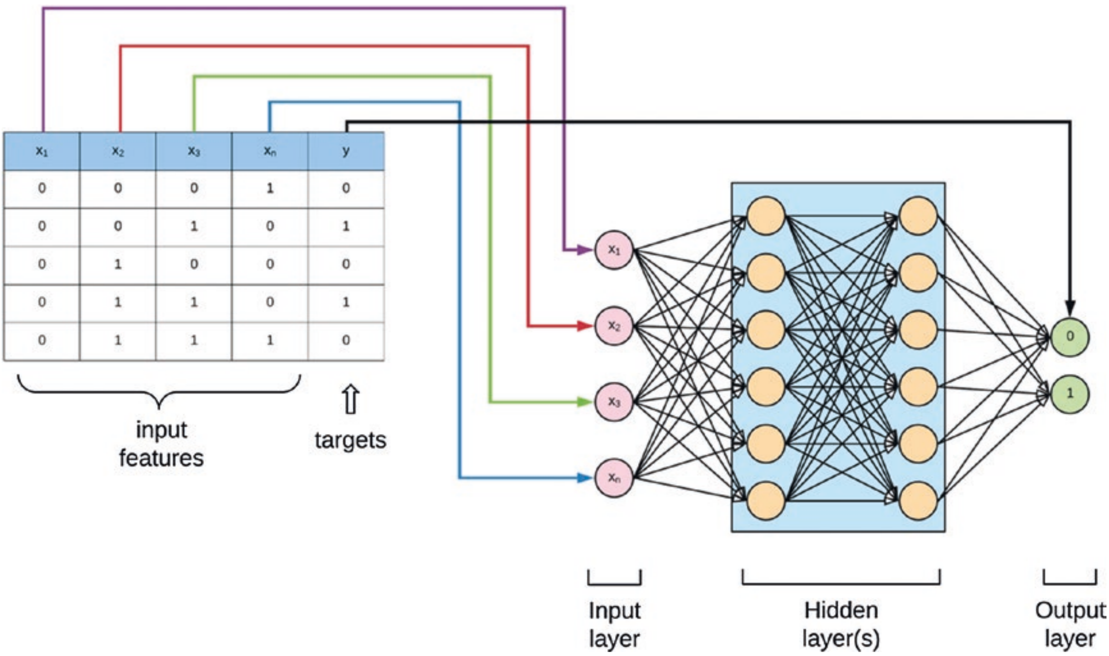

Figure 29-1. *Defining a neural network from a dataset*

A weight (also called parameter) is assigned to every neuron. The weights of neurons in a neural layer are multiplied by their inputs and then passed through an activation function (to be discussed in this chapter) for which the outputs are the inputs to the neurons in the next neural layer of the network (see Figure 29-2). This procedure is repeated as information of what the neural network is trying to learn moves from one layer of the network to another. Every neuron layer also has a bias neuron (typically set to 1) that controls the weighted sum. This is similar to the bias term in the logistic regression model.

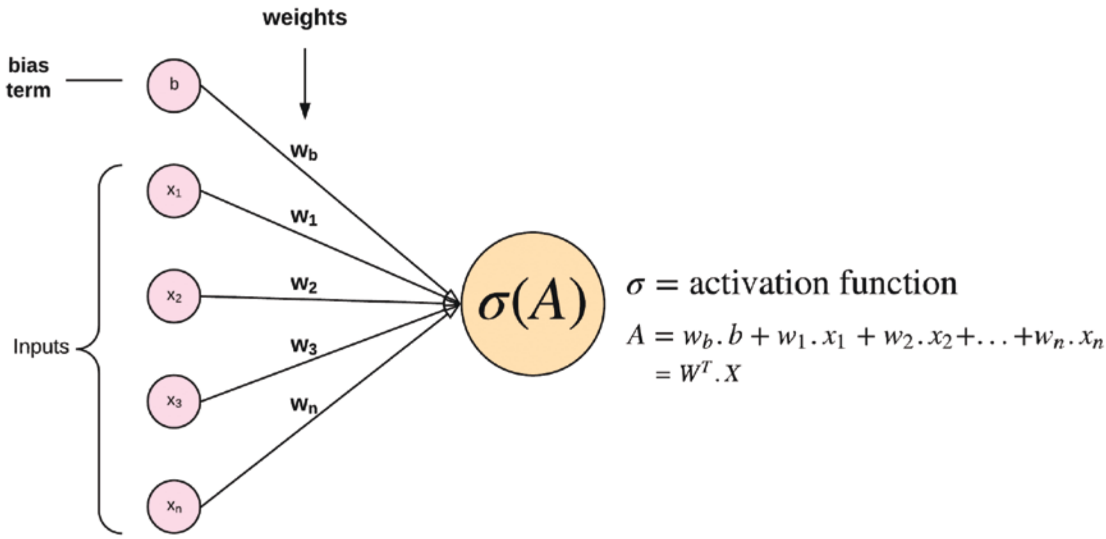

Figure 29-2. *Information flowing from a previous neural layer to a neuron in the next layer*

The weights are initialized as random values that are later adjusted as the network begins to learn using the backpropagation algorithm (to be discussed in this chapter). In summary, the outputs (or activations) of the neurons in the neural network layers are determined by the sum of the weight times the outputs plus the bias term of the neurons in the previous layer acted upon by a non-linear *activation function* (see Figure 29-2). This move is called the feedforward learning algorithm.

However, the output of the feedforward pass through the network may most likely result in an incorrect classification. The errors made from the feedforward procedure are later adjusted using the backpropagation algorithm (to be discussed). To evaluate the performance of the neural network, we define a cost function or loss function (similar to other machine learning algorithms) that captures the quality of the prediction made by the network.

The goal of the neural network is to minimize the cost function. Two commonly used cost functions are the squared error cost function for regression problems and the softmax cross-entropy cost function for classification problems.

Cost Function or Loss Function

The squared error cost function (also known as the mean squared error) finds the sum of the squared difference between the estimated target and the actual target for a real-valued problem, while the cross-entropy cost function finds the difference between the predicted class from the probability estimates of the actual class label in a classification problem.

Regardless of the cost function used, when the error loss is small, we say that the cost is minimized. In Figure 29-3, the correct output of the example passed into the network is **2.3**. After the output values are evaluated from the feedforward training, the squared error cost function is used to assess the quality of the network's output.

Remember that the MSE finds the average cost over all the data samples in the training dataset. In the example illustrated in Figure 29-3, we used just one data sample to demonstrate how the cost function works.

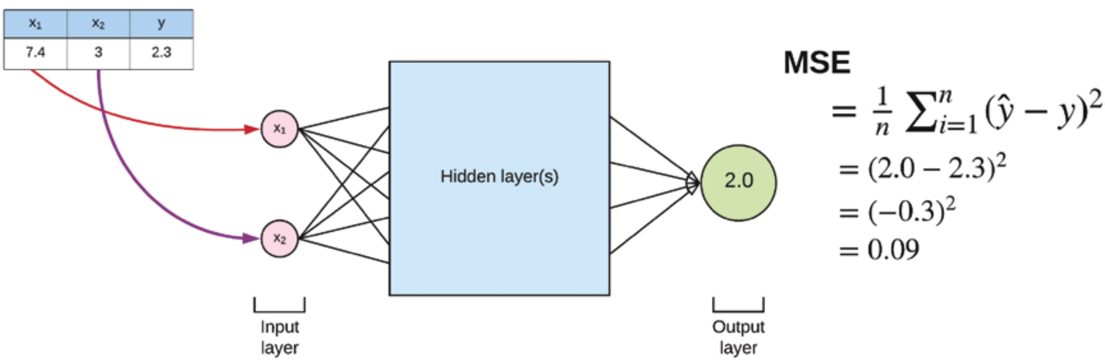

Figure 29-3. *MSE estimate of the neural network*

One-Hot Encoding

In a classification problem, one-hot encoding is the process of transforming the class labels of the target variable into a matrix of binary variables. The one-hot encoder assigns 1 when the output belongs to a particular class and 0 otherwise. An illustration of one-hot encoding is shown in Figure 29-4.