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.

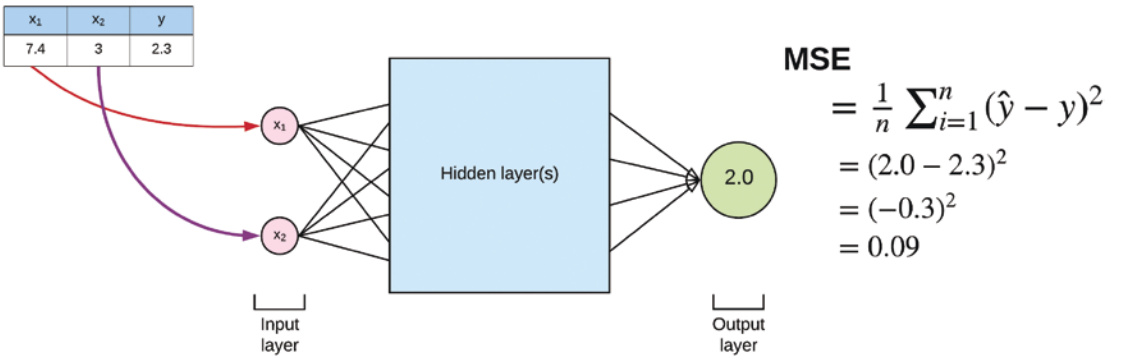
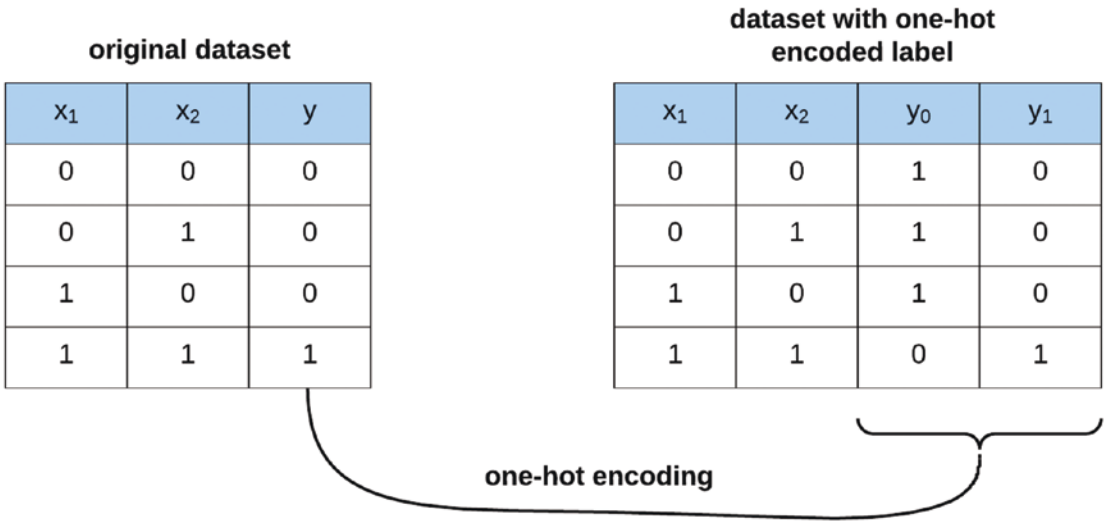


Figure 29-3. *MSE estimate of the neural network*

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.



x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

x_1	x_2	y_0	y_1
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure 29-4. *One-hot encoding*

In the final layer of the neural network, just before the output layer, an activation function called the softmax (same as discussed under "Logistic Regression") is applied to transform the activations to the probability that the example belongs to one of the output classes.

The purpose of applying one-hot encoding to the labels of the dataset is to represent the output as a vector of distinct classes with the probability that an example in the training dataset belongs to any one of the output categories.

The Backpropagation Algorithm

Backpropagation is the process by which we train the neural network to improve its prediction accuracy. To train the neural network, we need to find a mechanism for adjusting the weights of the network; this in turn affects the value of the activations within each neuron and consequently updates the value of the predicted output layer. The first time we run the feedforward algorithm, the activations at the output layer are most likely incorrect with a high error estimate or cost function.

The goal of backpropagation is to repeatedly go back and adjust the weights of each preceding neural layer and perform the feedforward algorithm again until we minimize the error made by the network at the output layer (see Figure 29-5).