APPLIED PHYSICS 186 – ACTIVITY 17

MACHINE LEARNING:

Neural Networks

Professor: Dr. Maricor Soriano
Instructor: Jayson Cubero

Rene L. Principe Jr.
2015-04622

dataset

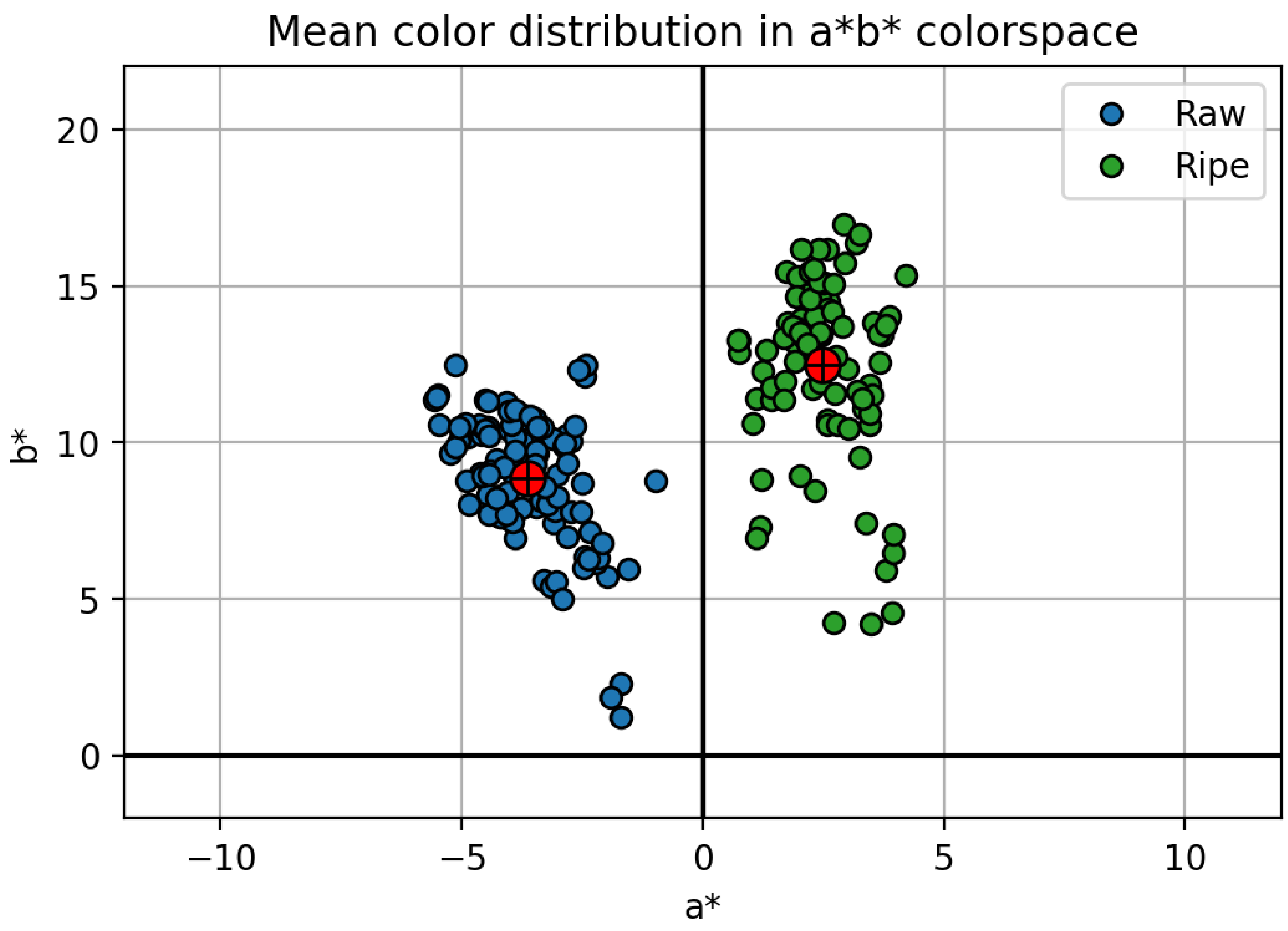

Figure 1. Color features a* and b* of raw and ripe bananas shall be used as feature inputs to a Neural Network.

A more sophisticated and complex architecture can be made by interconnecting multiple neurons to create a neural network [1][2]. Unlike the Perceptron algorithm, a hidden layer shall be added in this activity with the same goal, to classify the raw and ripe banana fruit dataset as shown in Figure 1. Here we use the color features a* and b* implying two inputs to the network, and a single percentage output from 0 to 1 classifying whether it is a ripe or raw one. Model parameters were varied and the corresponding training and validation loss were plotted to see how the model learns the appropriate weights and bias. There shall be 196 labeled data where 50% was used for training and 50% were being validated simultaneously.

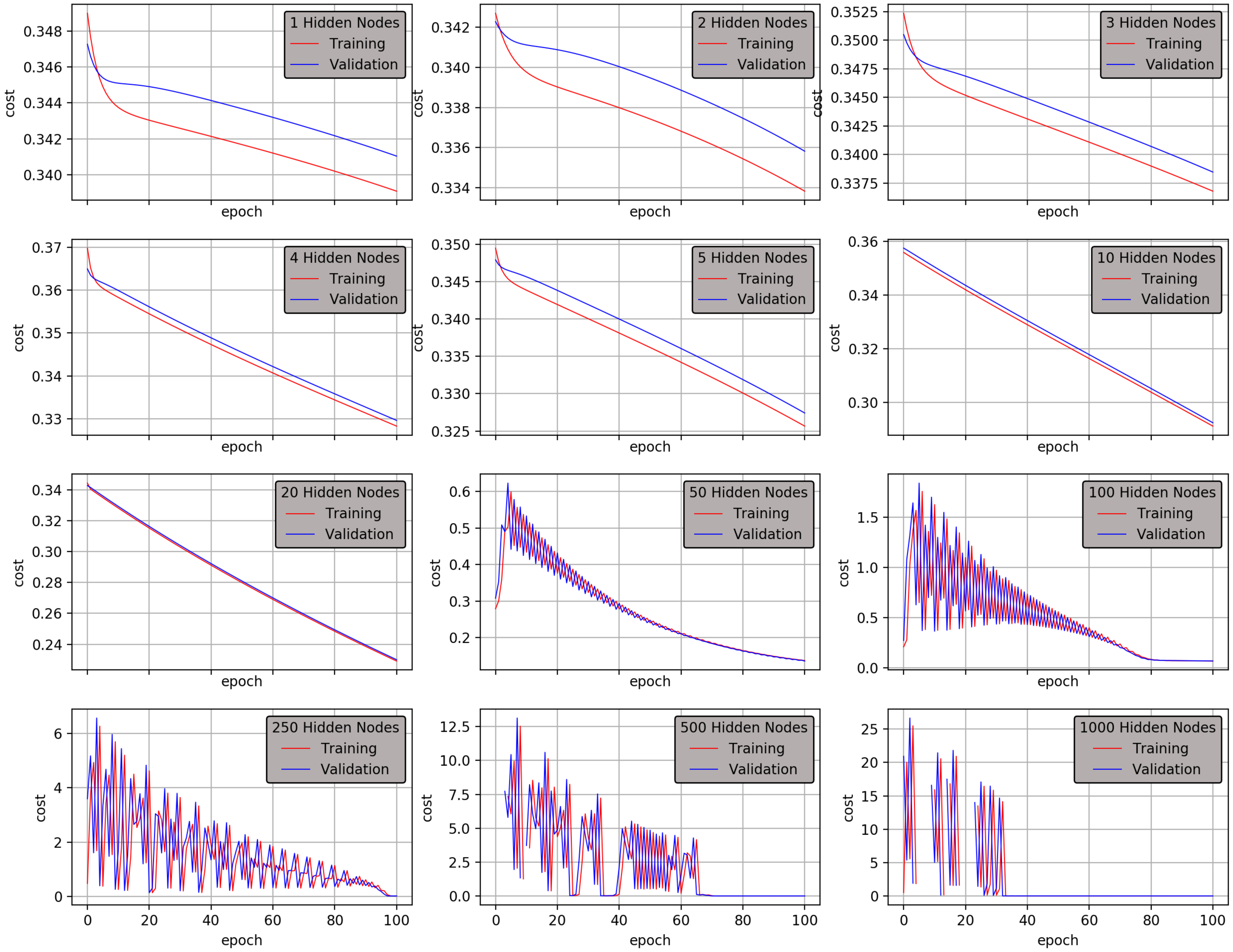

Figure 2. Trained at 100 epochs and learning rate of 0.1, the model's number of nodes in the hidden layer was varied and the plot shows its performance in terms of the cost function.

The results for varying the number of nodes in the hidden layer is shown in Figure 2. Notice that increasing the number of nodes doesn't necessarily imply a better performance. At 50+ hidden nodes, the model becomes unstable on starting epochs. Although it stabilized later on, precautions must still be taken to ensure a good model performance. Another tradeoff for having more nodes is more weight parameters, hence, more computation time.

Now, we set 10 nodes in the hidden layer and train at 500 epochs, the plots on Figure 3 show the model performance on increasing learning rates. Learning rates can be increased to decrease the training time as it dictates how much the parameter is changed, but then again, caution must be taken because high learning rate has a tendency to over-compute.

We can say that a model performs good if the cost function is minimized. For different problems, the model parameters and architecture is novel. In short, there is no staple choice of number of nodes and learning rate.

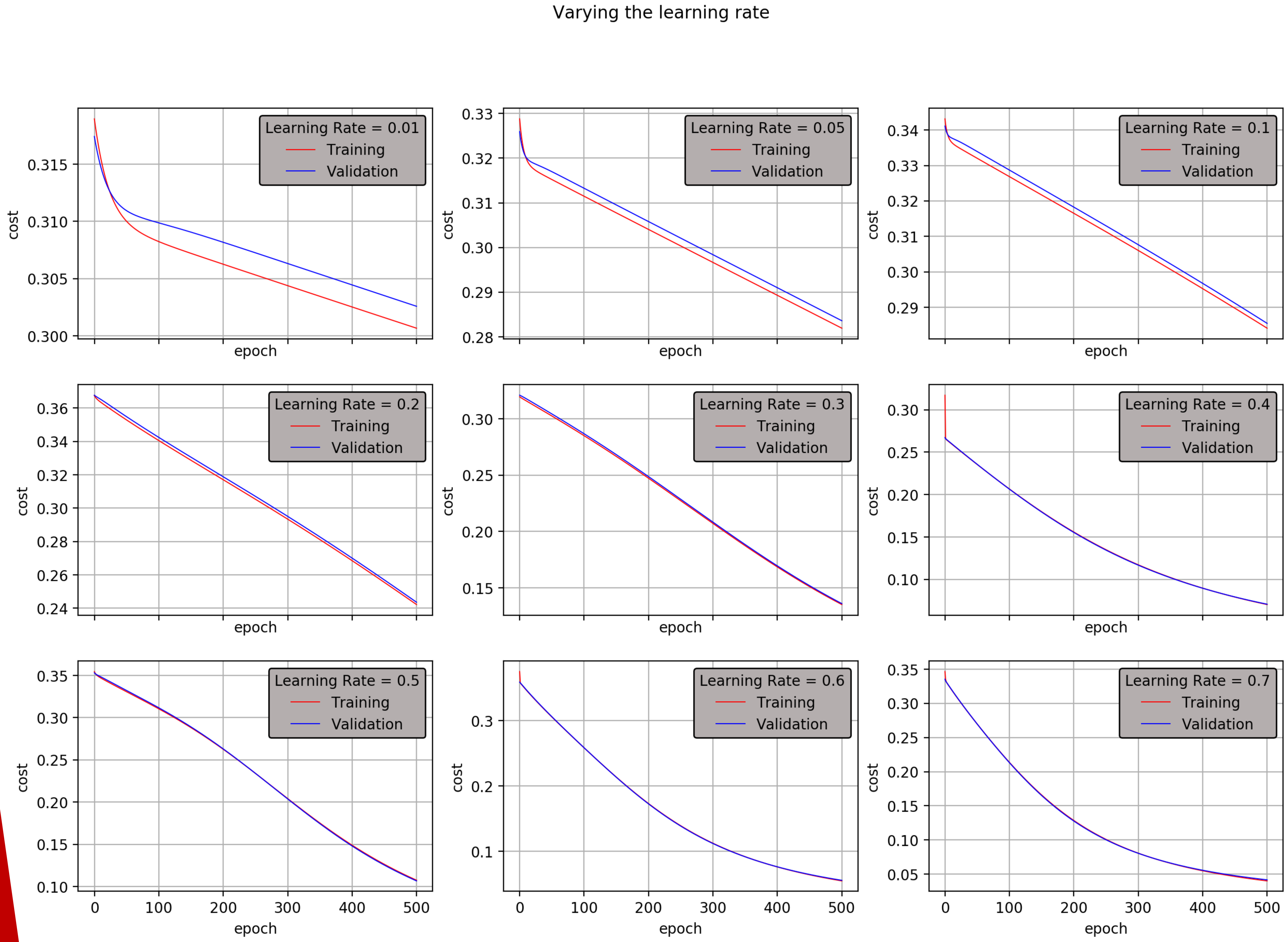

Figure 3. Different learning rates would have different slopes of performance. Higher learning rates decreases the cost on fewer epochs.

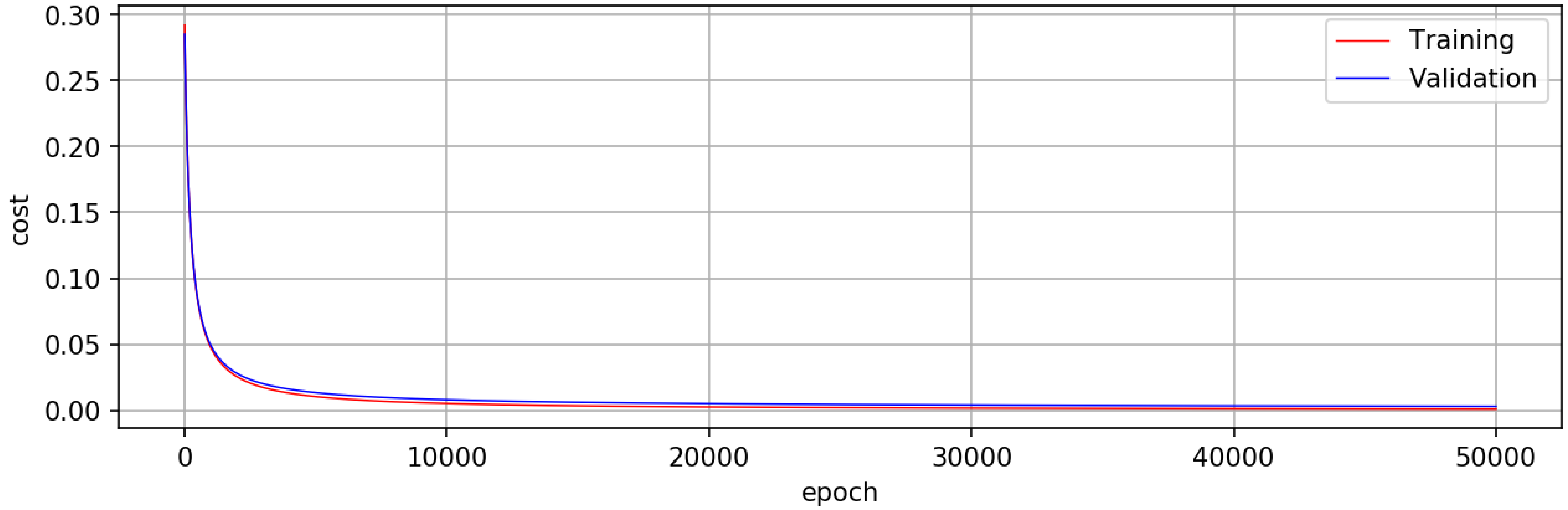

Figure 4. Trained at 100 nodes in the hidden layer and al earning rate of 0.1, the model shows a stable performance both for training and validation.

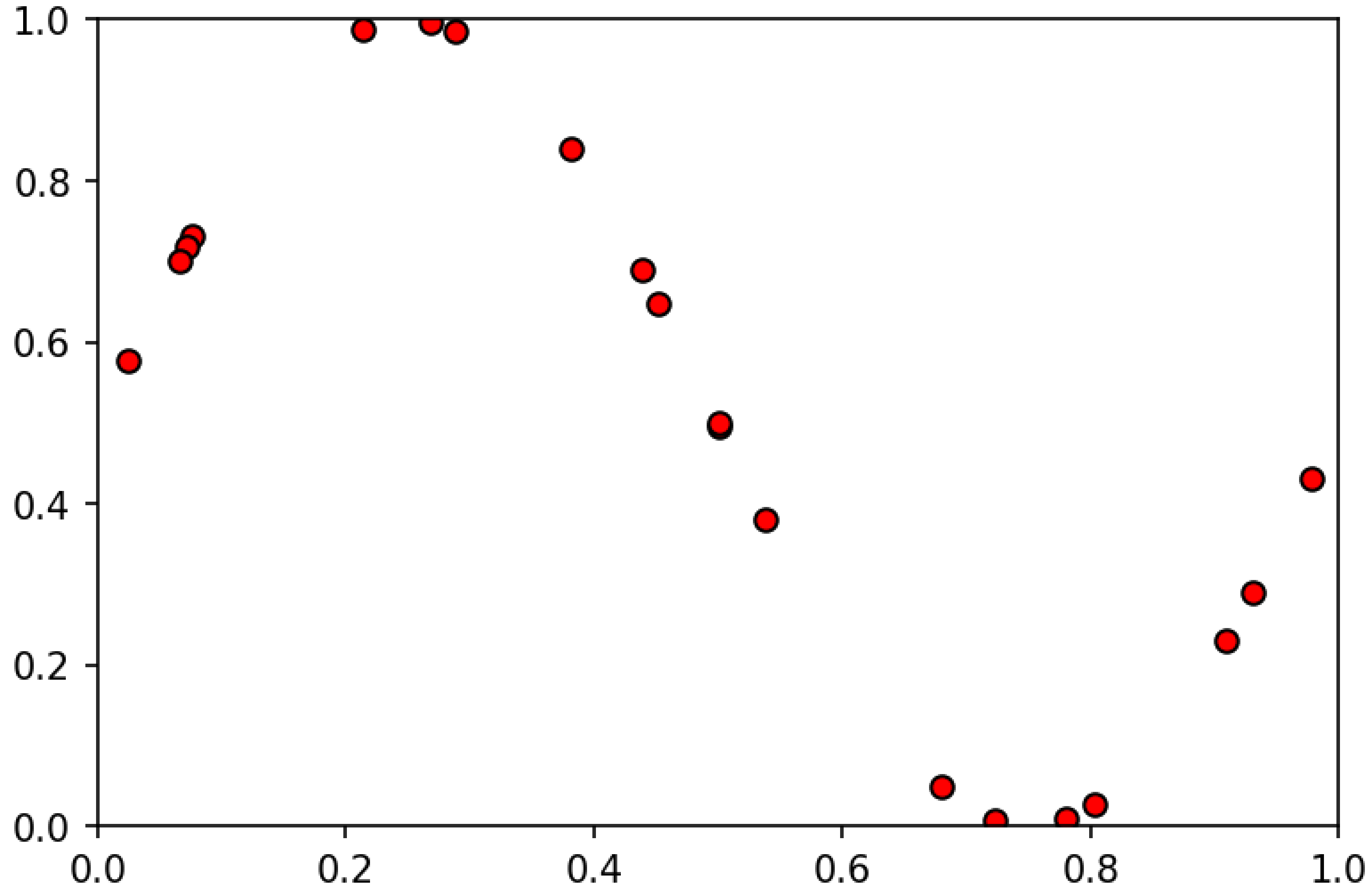

Figure 5. A synthetic 20 point dataset was generated showing the coordinates of a sinusoid. These values were normalized on both axes.

From the variations we've done earlier, a stable performing model is shown in Figure 4.

Aside from classification, a neural network can be a curve fitting tool. Here we use the concept we've learned from the logistic regression activity. Discretely sampled points of a sinusoid as shown in Figure 5 shall serve as the training data and model parameters were varied to check which performs the curve fitting the best. Results of the trial and error are shown in Figure 6.

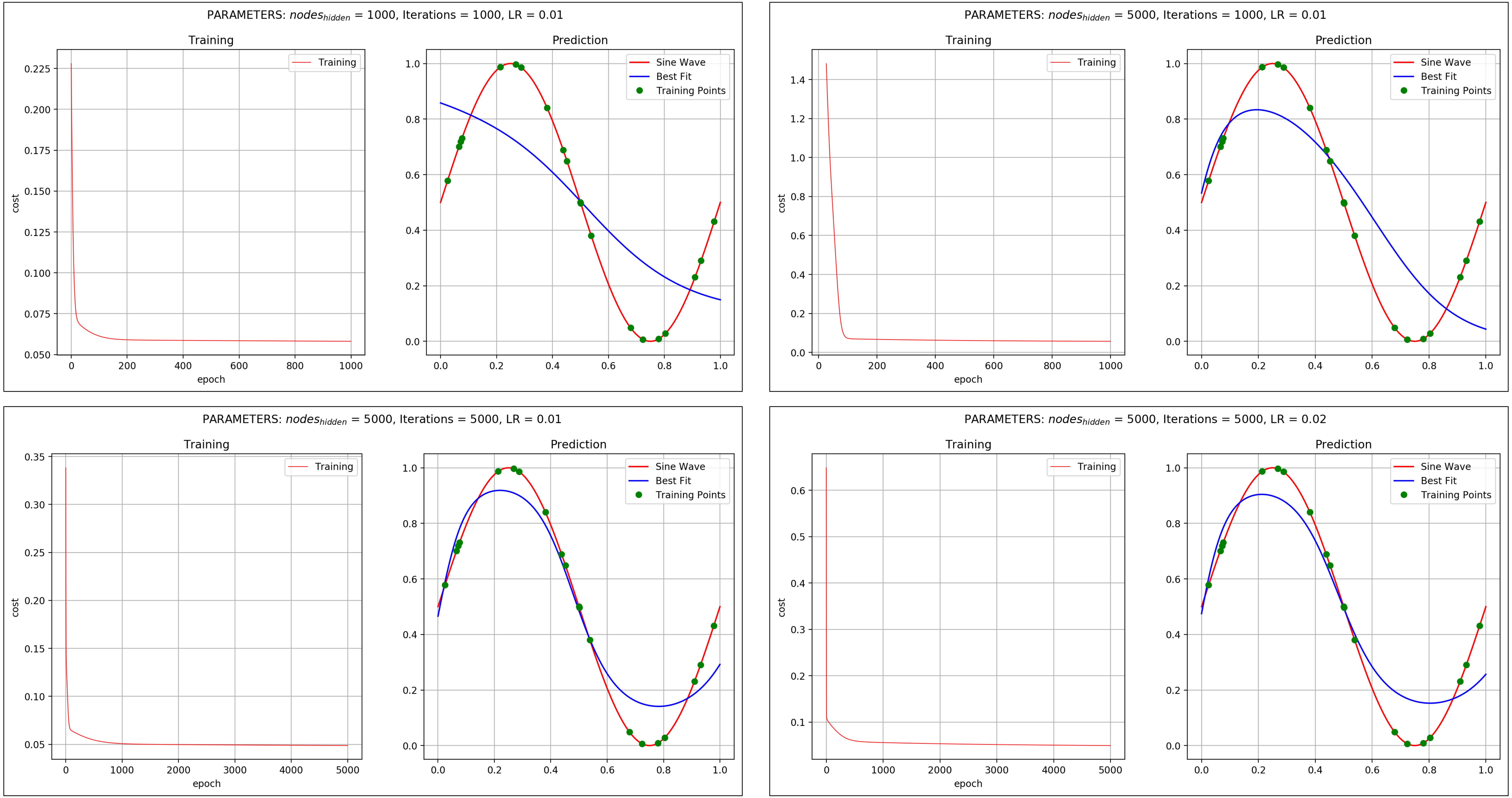

Figure 6. Different models were designed to perform the curve fitting and the plots above show its performance and fitting. More nodes and iterations has shown to improve the result.

The main advantage of Neural Network is that it incorporates the concept of gradient descent, ensuring that the weights are changed such that it moves closer to the actual value, thus, minimizing the cost [2].

Curve fitting and classification using an Neural Network was successfully implemented.

In this activity, I'd give myself a 12.

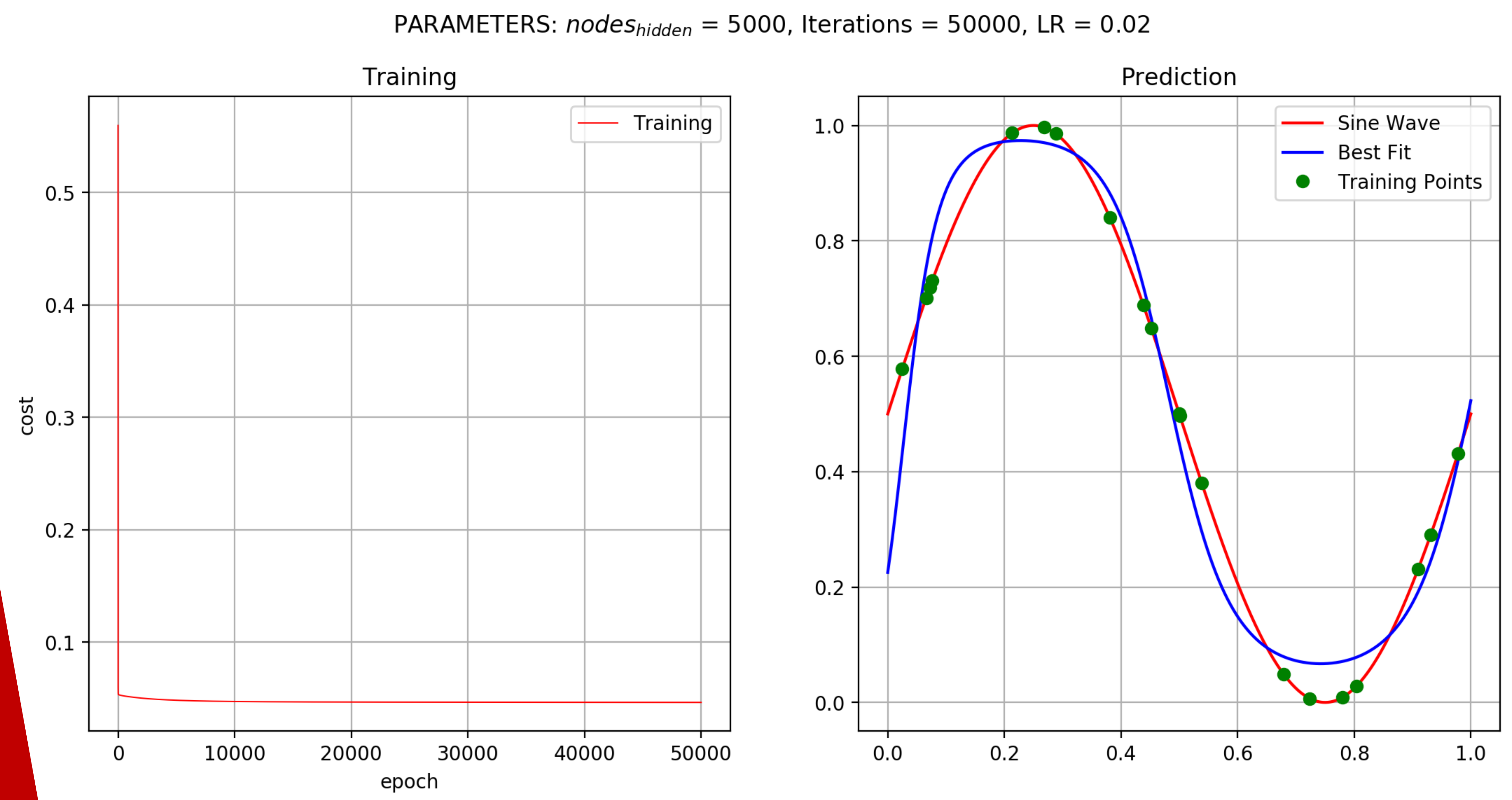

Figure 7. Executed in 3 m and 15 s, this model has performed best in fitting one period of a sine wave.

References:

[1] M. Soriano, "Neural Networks", 2019.

[2] C. Bishop, "Neural networks and their applications", *Rev. Sci. Instrum.*, Vol. 65, No. 6, June 1994.