Activity 17 - Neural Networks

de Castro, Crizzia Mielle | 2015-08076

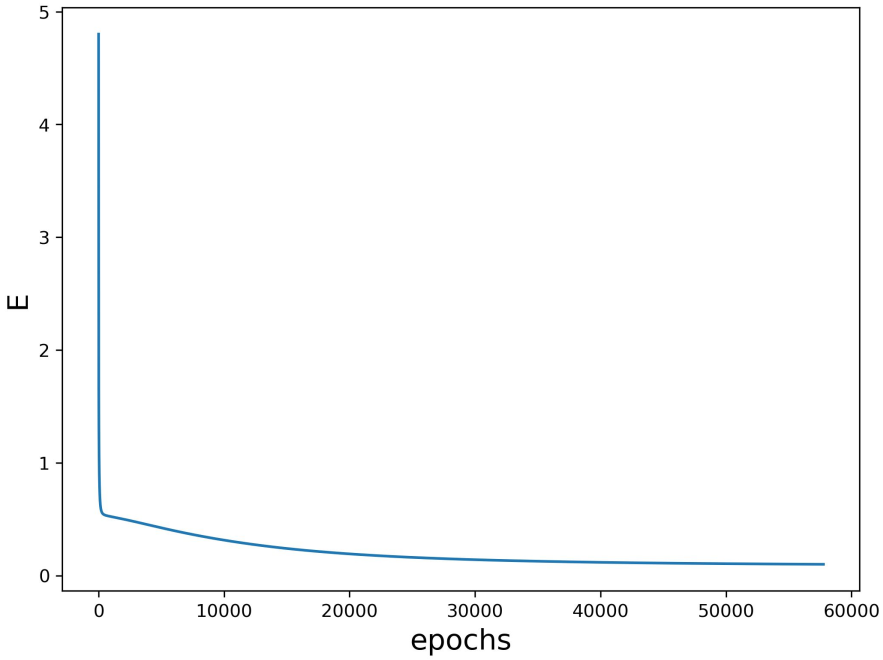
Parameters for my curve fitting neural network

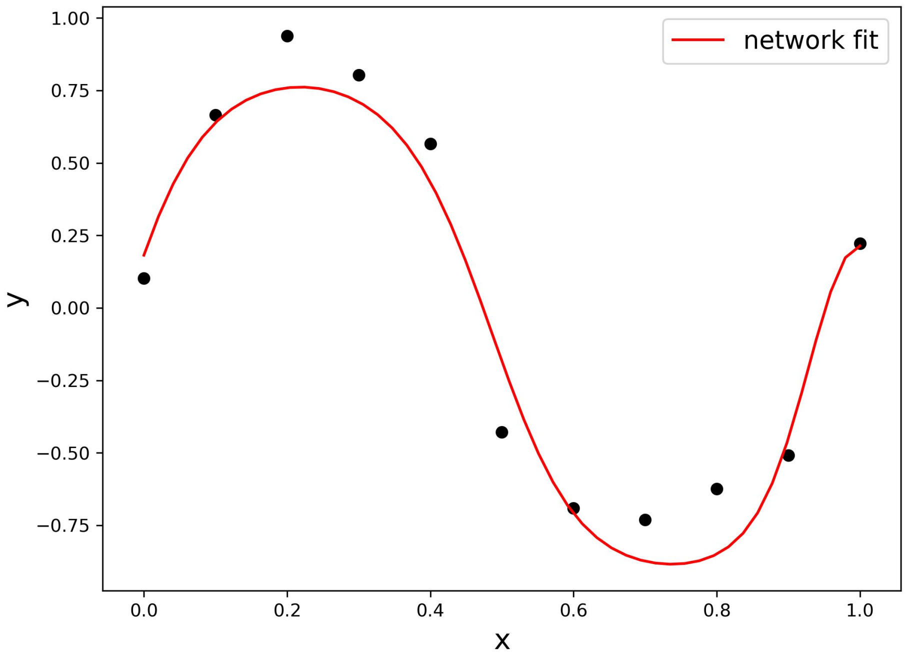
I tried to fit a sine wave using my neural network. I removed the hidden node, so my neural network only had the input and output node. I generated data points similar to the ones shown in Fig. 5 of [1]. I also tried to fit a pure sine wave without the Gaussian noise that has 30 data points.

The activation function I used was a s**igmoid function, shifted from range 0-1 to range -1-1.** I shifted the sigmoid function since the range of a sine wave is from -1 to 1. I also calculated the corresponding derivative of the shifted sigmoid function. This was suggested to me by Jom Macalintal. I set the **order of my polynomial to 11** and my **error threshold to 0.1**.

Determining accuracy of my curve fitting neural network:

generated noisy data points

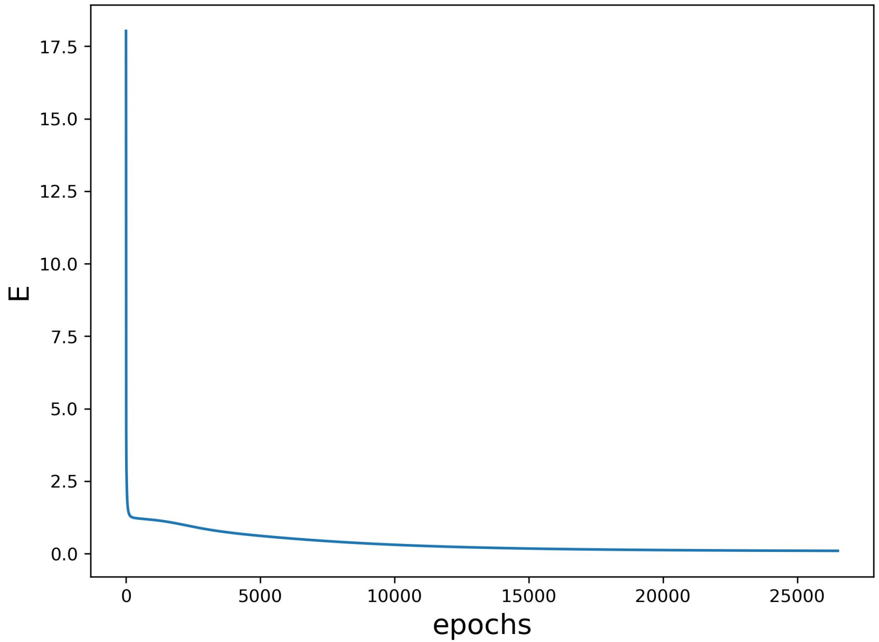


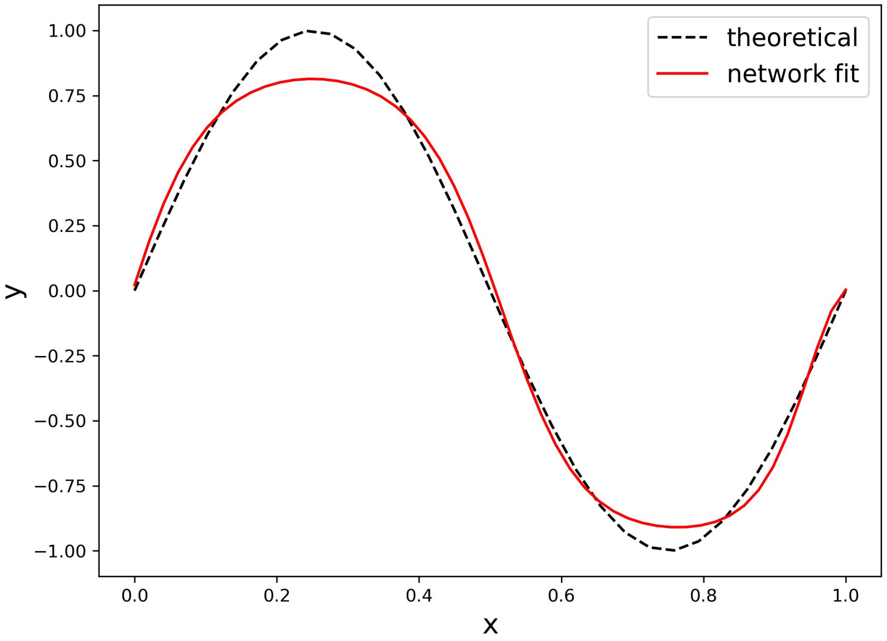


My neural network was able to converge after around 60000 epochs. The plot on the right shows how my neural network compares to the generated noisy data points. Just like in [1], my neural network was able to obtain the underlying trend in the data. The curve fit resembles a sine wave.

Determining accuracy of my curve fitting neural network:

pure sine wave





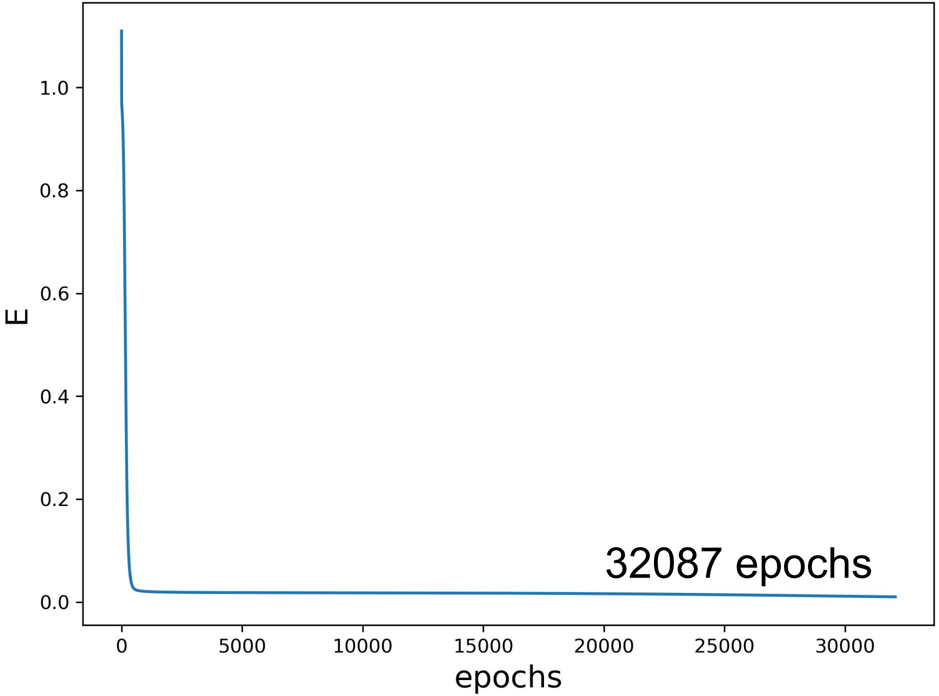
My neural network was able to converge after around 25000 epochs. The plot on the right shows how my neural network compares to the pure sine wave. The curve fit from the neural network looks similar whether it was fit with noisy data points or a pure sine wave.

Parameters for my classification neural network

I have three classes: roses, chrysanthemums, and butterfly peas. I set the expected output of my neural network to be **0.3, 0.6, and, 0.9**, respectively, for each class. Each class has four features: eccentricity, r, g, and b. Therefore, I have **five input nodes** (including bias)**, four hidden nodes** (including bias), and **one output node**.

I set the **learning rate at 0.1** and the **error threshold at 0.01**. I plotted the **E vs epochs** for different activation functions to determine which converges given my data set. I found that the **sigmoid function** works for my data set. I set aside half of my data set for training, and the other half for testing.

Determining accuracy of my curve fitting neural network



My neural network was able to converge after around 30000 epochs. To check the accuracy of my network, I applied the final weights (w_{ji} and w_{kj}) to the other half of my dataset. I calculated y_k for each. **All of my roses, chrysanthemums, and butterfly peas output 0.3, 0.6, and 0.9**, respectively, after rounding off. My network was able to correctly classify all of my test data set.

References

1. C. Bishop. Neural Networks and Their Applications. Rev. Sci. Instrum. 65(6), June 1994.