Machine Learning
Jacobs University Bremen
Mohit Shrestha
Seongjin Bien

May 26, 2018

Homework 11

For the task of learning the XOR function, we implemented a Multi-Layer Perceptron with the network structure that was recommended in the assignment sheet - with 1 hidden layer of 2 neurons, and bias units in both the input and the hidden layer. The weight matrices are initialized using NumPy's `random.random` function, which populate the matrices with floating point values between 0 and 1. We first implemented our backpropagation algorithm, and once we could verify that our error rate was converging, we varied our step size to observe the convergence rate and optimize it. To do this, we wrote a loop with varying step sizes, which broke whenever the error fell below a certain threshold, and noted down the number of epochs required. The top graph Figure 1 below pictures the convergence of the absolute error rate, whereas the bottom graph pictures the convergence of the misclassification rate, by postprocessing the data as mentioned in the assignment sheet.

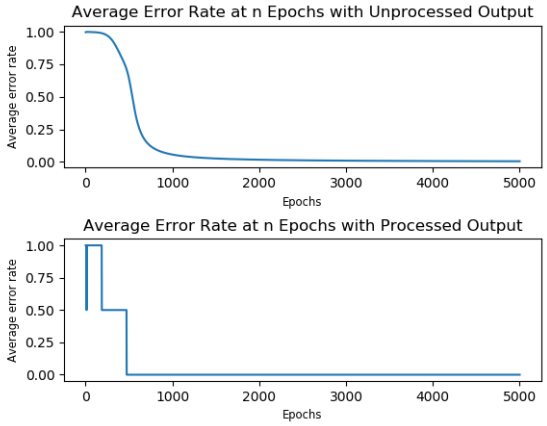

Figure 1