Neural network

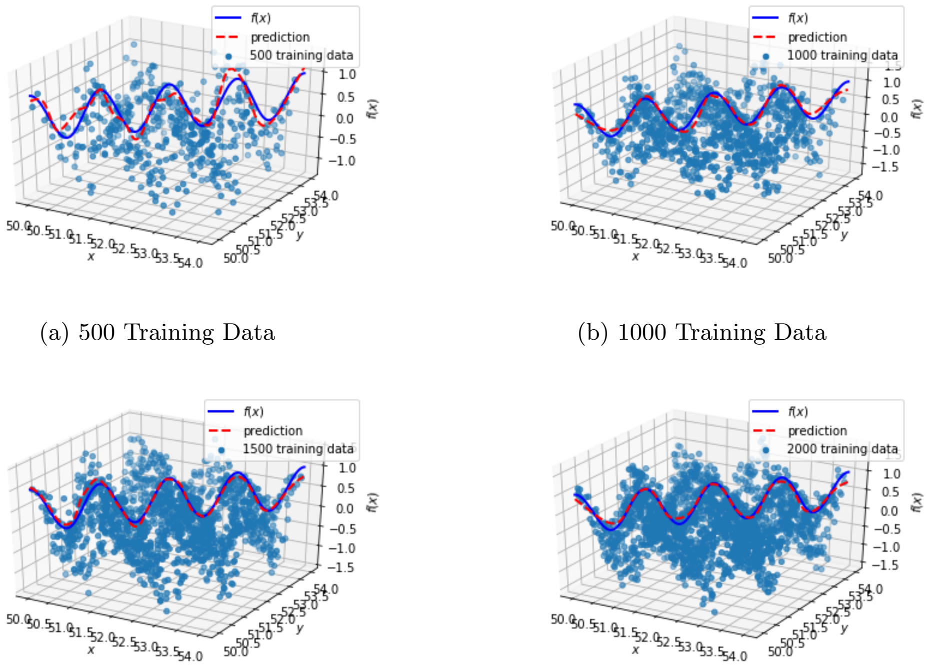
Xiaoguai Li

1 Goal

Generate a set of N observations using a Latin Hypercube Sampling of $f(x,y) = cos(\pi x)cos(\pi y), x, y \in [50, 54]$ and train a deep neural network with two hidden layers, 50 neurons per layer, and a hyperbolic tangent activation function to approximate $f(x, y)$. Plot the approximation error in the L2 norm as the number of training points is increased.

2 Results

The following figures in Fig.1 are directly generated by functionApproximatioin_NN, which is the main script. It imports NeuralNetwork from models numpy.py. At the beginning of functionApproximatioin_NN.py, number of training data can be defined. 1000 test data is used. 40000 iteration is used to train the model. Batch size is full. Fig.2 shows the error of prediction with respect to different numbers of training data. This plot shows convergence, which is not guaranteed theoretically.

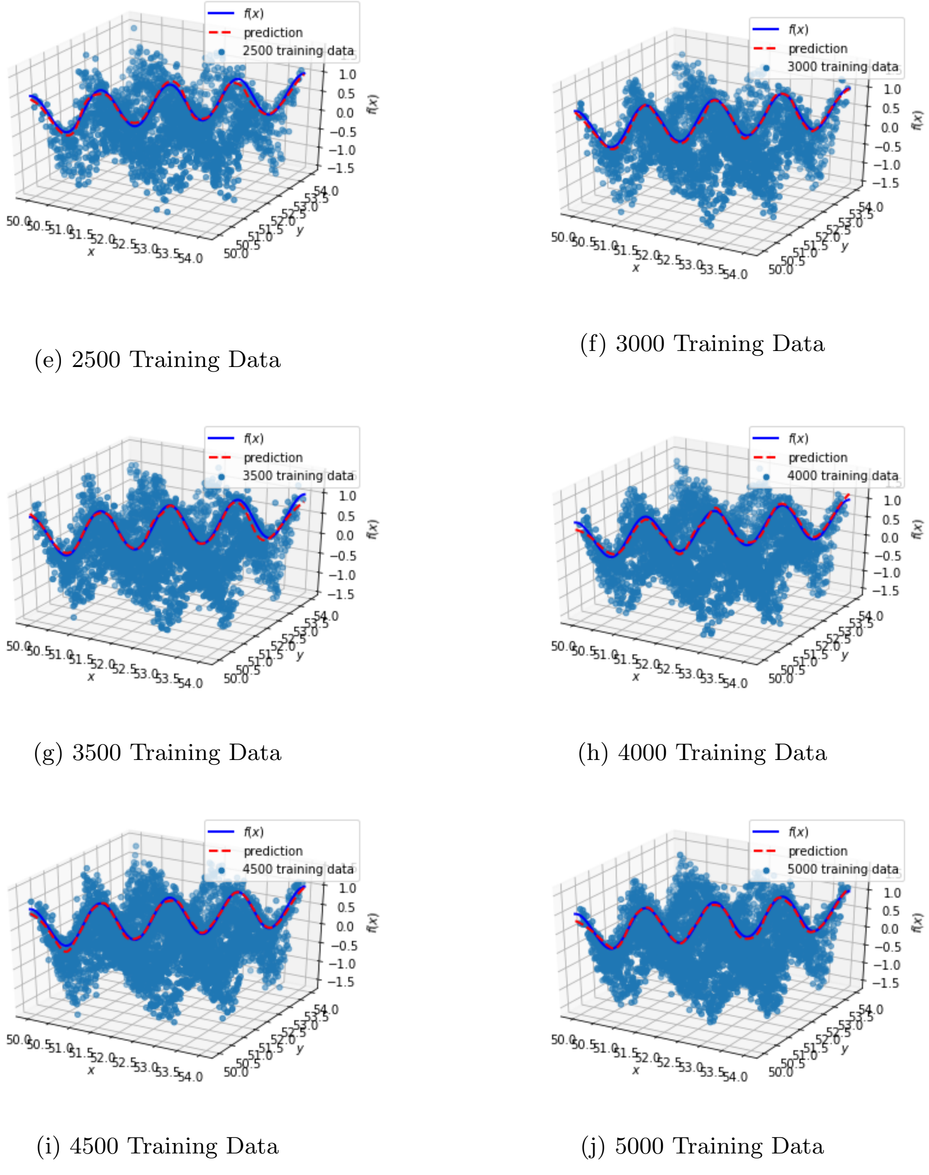


(a) 500 Training Data

(b) 1000 Training Data

(c) 1500 Training Data

(d) 2000 Training Data



(e) 2500 Training Data

(f) 3000 Training Data

(g) 3500 Training Data

(h) 4000 Training Data

(i) 4500 Training Data

(j) 5000 Training Data

Figure 1: Prediction results with different numbers of training data

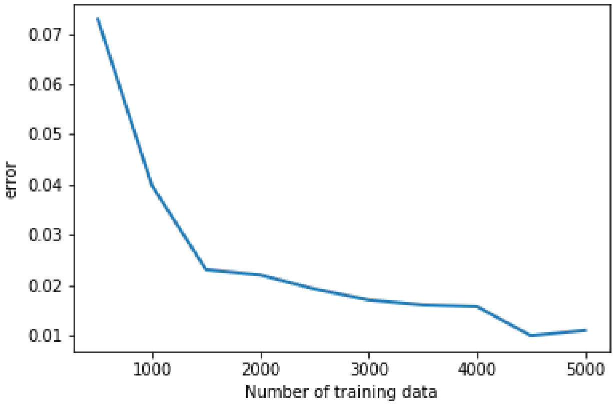


Figure 2: Prediction error decreases with increasing number of training data