Inverse Computation of the Thermostat Damping Parameter in NVT Simulation Using Artificial Neural Networks

Yuhai Li

1. Introduction

The purpose of this project is to perform a data-driven model and explore its possibility to predict the damping parameter in a NVT simulation.

There are four parts in this project: 1) Simulation. The raw data is generated by a set of thermostat simulations given in the homework 4 with damping parameters ranging from 10 to 2500. 2) Data encoding. The temperature changes of the system is preprocessed and eventually converted to a 101-dimension vector which is the input vector for neural network. 3) ANN model. The artificial neural network has 101 input neurons, two hidden layers, each of which has 51 neurons, and one output neuron which denotes to the damping parameter. 4) Evaluation. A couple of evaluations are performed to estimate the overall performance and the model's ability to extrapolate unseen data.

2. Simulation

Simulations are performed in the identical circumstance as homework 4 except that there are 40,000 steps instead of 20,000 steps performed in the process of rise in temperature. Therefore, there are 250 simulations performed in total, by assigning the damping parameter from 10 to 2,500, incremented by 1 per simulation.

3. Encoding

In a simulation, there are totally 4,000 temperature data returned, each of which denotes the result of every ten steps of simulations. If total 4,000 variable are directly fed into a neural network, one of the following two problem would be raised: 1) if a lightweight network is used, then most likely it will be overwhelmed by 4,000 input variables, 2) if a giant network with 4,000 variables is used, then 250 simulations is not enough for this network to converge. Therefore, it is necessary to reduce the number of variables with as little information loss as possible.
I found that a 101-dimension temperature vector with results of every 400 steps (starting point and ending point inclusive) is a reasonable balance between the number of variables and information loss. The Figure 1 shows the comparison between the original curve and the encoded curve of temperature vs. time.

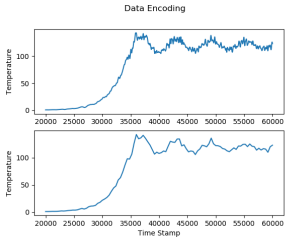


Figure 1: Data Encoding

The top curve is the original curve and the bottom curve is encoded curve. Figure 1 shows that the encoded curve is able to capture enough information and the trend of the original curve in a good precision with less time stamp variables.

4. Artificial Neural Network (ANN)

The ANN uses the encoded vector as the input vector. Therefore, there are 101 neurons in the input layer, each neuron is fed by a temperature at a certain time stamp in the encoded vector. The ANN has two hidden layers, each has 51 neurons. The ANN is fully connected. Each neuron in the hidden layers uses ReLU activation function and the output neuron used liner activation function. The loss function is mean squared error (MSE). The ANN is trained by back propagation with stochastic gradient descent to minimize the loss function.

5. Model Evaluation and Discussion

5.1. Interpolation

The ANN reveals an excellent performance in interpolation. With training and testing data random splitting ratio equals to 8:2, the ANN model has a r^2=0.9999 and root mean squared error (MSE) = 2.55. The graph of predicted vs. true value is shown in Figure 2.

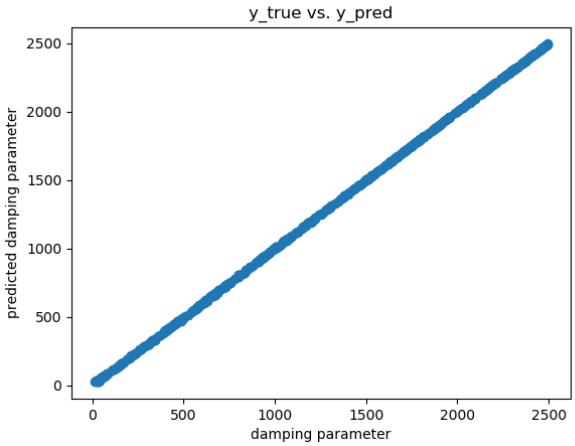


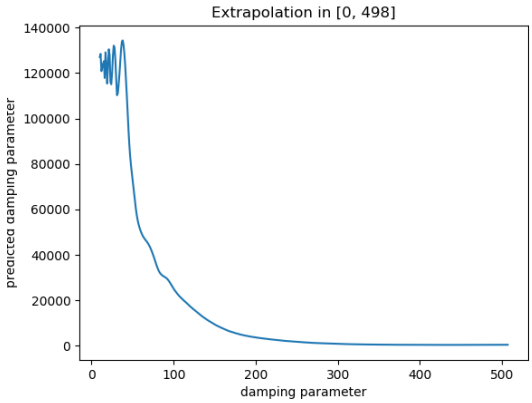
Figure 2: Data Encoding

5.2. Extrapolation

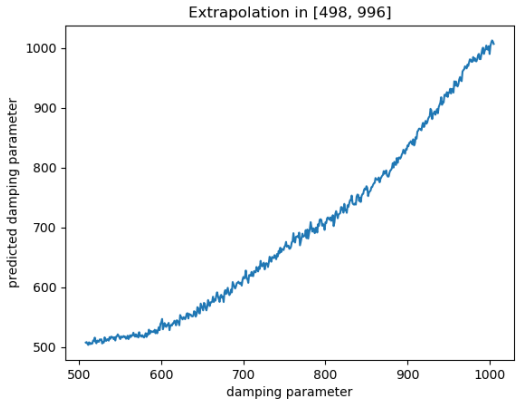
To measure the ANN's ability for extrapolation, I manually split the whole dataset into five equal-sized data chunks based on the value of their damping parameter. To estimate the performance in each range, there are five individual test performed with one chunk being testing set and the rest chunks being training set, alternately. The results are shown in the Figure 3. From the results, we can see that ANN fails to extrapolate the extreme values, like figure 3(a). When the damping parameter is small, the temperature reaches the final temperature very soon which results in a shape rise in temperature at the beginning and the rest temperatures are same. To handle this case, the ANN model must assign a large weight to its first input neuron because the rest of input variables don't make much contribution to the final output. If the ANN has never seen this type of data, it is hard to make this extreme decision.

6. Conclusion

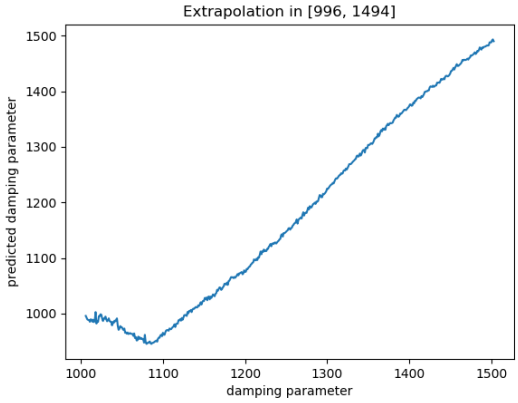
It is feasible to perform a data-driven model to inversely compute the damping parameter given simulated data. The ability of ANN model to interpolate is perfect while it still has limits to extrapolate unseen values, especially extreme values.



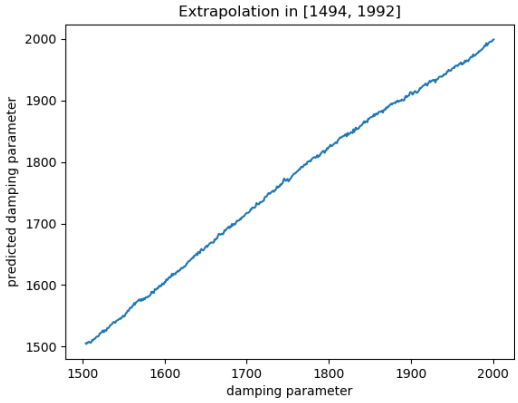
(a) R2 = -51576.98, RMSE = 32649.01



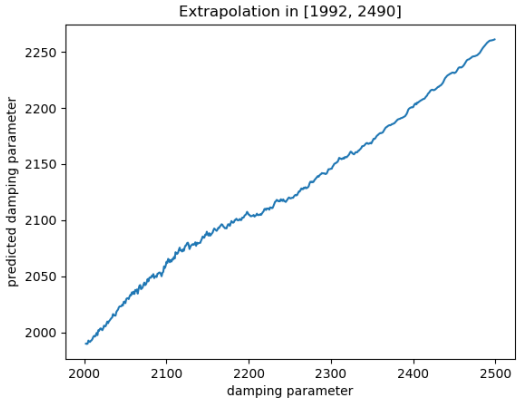
(b) R2 = 0.93, RMSE = 37.43



(c) R2 = 0.68, RMSE = 81.32



(d) R2 = 0.99, RMSE = 10.88



(e) R2 = 0.63, RMSE = 87.31

Figure 3: Performance of extrapolation in five ranges