Regression with neural networks

1. Introduction

The task is to implement a simple neural network that achieves an adequate performance on a real data set for predicting a real valued parameter. The dataset consists of chemical properties (representation) of superconductors, and the parameter value to be predicted is their critical temperature in Kelvins (°K).

Further information on the dataset is available at:

http://archive.ics.uci.edu/ml/datasets/Superconductivty+Data

The task is utilize the training dataset consisting of the properties of superconductors and their critical temperatures in order to learn a regression model. Then this model should be applied on the test dataset to predict the critical temperature of the superconductors in that dataset.

2. Assignment

Implement a simple neural network and the backpropagation algorithm in Java or Python! Use the trained model to predict the critical temperature for each test sample.

2.1. Java

The code must contain a *Main* class, and within this, a *main()* function. It will receive all inputs on the standard input, and should output the solution to the standard output. Upload the zipped source code files of your application to the BME MIT HomeWork portal. (https://hf.mit.bme.hu).

2.2. Python

The code must be a single python file, that will be run and receives all inputs onto the standard input, and it should write the solution to the standard output. Upload the zipped single python file to the BME MIT Homework portal. Use Python3.x, and only standard libraries are available (e.g. no numpy!) (https://hf.mit.bme.hu).

2.3. Input

The program receives all inputs via the standard input. The input consists of the representation of training samples, the corresponding critical temperatures, and also the representation of test samples. The character ’\n’ is used as a line separator. The input is structured according to the following:

1. The first 17011 lines each contain a representation of a chemical compound, that is 81 parameters as real numbers separated by the ’\t’ character. These are the training samples.

2. These are followed by 17011 temperature values, i.e. the target value to be learned (i.e. a single temperature value in each row).

3. Lastly, 4252 test samples (chemical compound representations) for which the critical temperature has to be predicted. These are the test samples.

The solution should implement the backpropagation algorithm. The scaling/normalization of data is recommended before learning. Note that the available CPU time for the code is approximately 120 CPU secs.

2.4. Output

The output contains the predictions for the test samples, i.e. a predicted temperature for each sample. The output should be formatted such that each row contains only one prediction, the order corresponds to the order of test samples. Rows should be separated by the \n character, the output should be written to standard output.

3. Evaluation

The evaluation is based on RMSE (root mean squared error):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2},$$

where y_i is the real value, and $\widehat{y_1}$ is the predicted value. A solution reaching a RMSE lower than 17.0 gets 12 points, however a solution above 23.0 gets 0 points. Between these two endpoints the evaluation is linear (the score is rounded to the nearest integer).