Data Driven Optimization
Tutorial 6

A.K. Cherukuri

April 25, 2023

1 Hands on Regression with Neural Networks

In this tutorial, we'll construct a neural network which is learning a simple function from the corrupted data.
Steps:

1. Create an equally spaced (with step size 0.01) input data between 0 and 2π. As the output, first generate random values ξ_i, distributed uniformly between $[0, 0.1]$, with the same size of the input vector and then calculate the to-be-regressed output via the equation

$$y_i = \sin(x_i) + \xi_i.$$

2. Split the dataset into a train and test sets, x_{train}, y_{train}, x_{test} and y_{test}, using a 80/20 % split. To do this, you can use the function

   `cvpartition().`

3. Now we define the neural network parameters; construct a neural network with one output and one input layer, with 25 neurons, for the learning rate make use of $\alpha = 0.03$, and lastly train the neurons for 1000 epochs (iterations).

4. During each epoch, by using each input-output pair, calculate the prediction error of the network by using the current neuron weights (*forward run*) and update the weights based on the prediction error (*back-propagation*). Store the training RMS error of each epoch.

5. Manipulate the NN parameters (the learning rate, number and (topological) structure of neurons, total number of epochs, cost function) and repeat the training algorithm by making use of the **same** training data.

6. Visualize the true data and the predictions via the obtained NNs for the training and test datasets. Create histogram plots of the test errors (of different NNs) using the function

   `plotterrhist().`

7. Another important metric to compare the NN models is the out-of-sample prediction error, the prediction error of the model for the input data that is considerably ‘outside’ of the training samples. Generate new input data between $[3\pi, 5\pi]$ with steps of 0.01 and calculate the ‘true’ output by $y_i = \sin(x_i)$ (there is no need to corrupt this data with random values.). Now compare the models obtained from the steps above with respect to their out-of-sample prediction errors. Can you regularize the cost function such that the sinusoidal behaviour is embedded into the NN?

8. Lastly, we will repeat all of these steps with the Neural Network toolbox of Matlab. For initiating an instance of neural network in Matlab using the toolbox, make use of the function

   ```
   fitnet().
   ```

   Make use of the documentation to ‘configure’ the neural network parameters as desired and repeat again the analysis requested above.