THE UNIVERSITY OF CALGARY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ENEL 525 Machine Learning for Engineers

Lab 4 – Time Series Prediction Using Backpropagation

18 November (B02) & 21 November (B01), 2025

Objective: Design a predictor based on a feed forward neural network with multiple layers using the backpropagation learning algorithm.

Part 1: Training a Neural Network using the Backpropagation Algorithm

1. In this lab you will be exploring the predictive capability of a neural network based on backpropagation training. Design a predictor network with 5 neurons in a hidden layer and 1 output neuron. Use the log sigmoid activation function for the hidden layer neurons, and a linear activation function for the output neuron. Generate a data sequence of 200 points using the following chaotic dynamic system:

$$p[n] = 4p[n-1](1-p[n-1]) \text{ , with } p[0]=0.35.$$

For each point to be predicted, use the previous two data points in the sequence to predict the desired data point.

2. Implement and apply the backpropagation algorithm to find the weights and biases of the predictor for this chaotic data sequence (use the first 190 data points to train). Initialize the weight matrices and biases with random values using `numpy.random.normal()`. Set the learning rate to 0.1 and the error threshold to 0.015.

3. Keep track of the error resulting from each input pattern in a vector. At the end of an iteration (one iteration meaning a pass through the training points 1-190), calculate the mean squared error (MSE) using all the errors obtained from the current iteration. This MSE value will be compared to the error threshold, and it will be used for plotting the learning error curve; keep track of the MSE value from every iteration in another vector for plotting.

4. Plot the learning error curve (MSE vs iterations).

Testing the Neural Network:

Use your predictor to predict data points from 190 to 200. Compare the predicted values and the true values (the true values are the pre-generated chaotic data points from 190 to 200). Display the true values as well as the predicted values for points 190-200 in the same plot.

Part 2:

Data from a real-world time series will be provided in the lab. Use the `numpy.load()` function to read the .npy file. The size of the data sequence is 180 points.

Implement and apply the backpropagation learning algorithm to train the network weights and biases (use the same architecture as in part 1). Use only the first 170 data points for training. Set the learning rate to 0.05 and the error threshold to 0.00002. After training, plot the learning error curve. Predict the next 10 points using your predictor (points 170 to 180) and create a figure with both the true points and the predicted points.

Marking:

Show the following to the TA during the lab period:

Part 1:

● Show the learning curve plot (i.e. MSE vs iteration number).

● Display a figure showing the true values for points 190-200 as well as the predicted values for points 190-200.

Part 2:

● Show the learning curve plot (i.e. MSE vs iteration number).

● Display a figure showing the true values for points 170-180 as well as the predicted values for points 170-180.

The TA may apply different/additional inputs to your code.