AI Final Project

Mark Cohen and Jonah Caro

Goal

Our goal is to test if it is possible to predict the 30-year Fixed-Rate Mortgage (FRM) rates a few days into the future using a Temporal network or an Elman Recurrent Neural Network (RNN).

Approach

Both networks are trained and tested using a csv file which contains about 2700 data samples obtained from Fed Economic Research. Each network is trained with about 85% of the data, and the remaining 15% is used for testing. Both networks print out the actual rate, a predicted rate, and an error function we can use to evaluate accuracy.

The Temporal network reads the data into a TemporalMLDataSet used for training. An MLRegression model is built and used to train a feedforward network with sigmoid activation.

For the RNN we read in the data and normalize it. We then create an Elman RNN BasicNetwork using sigmoid activation. An MLDataSet is then generated and used for training.

Challenges

We set out to predict mortgage rates 3 - 5 days out. This was not within the scope of what we could accomplish so we stuck to a one day prediction. There are many variables: neuron count, network structure, window size, target training error, training set size. While we experimented with many of these variables, we decided to focus on window size and tested each network using sizes of 1, 12 and 60 (in weeks).

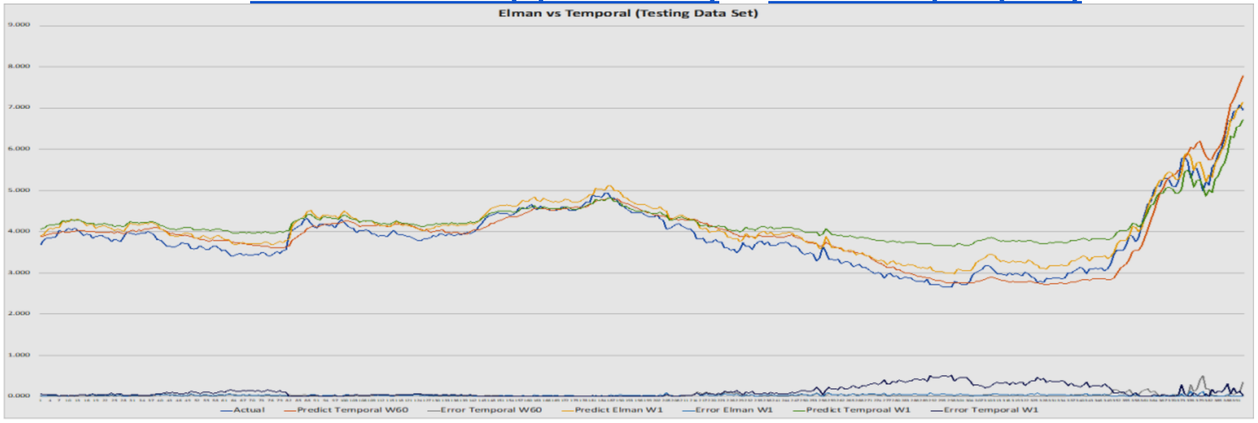
Results

In all cases our predictions lagged behind the actual data. We guessed this problem might represent a martingale, and theorized that a window size of 1 may yield the best results. This tended to be true, but surprisingly the Temporal network with a window size of 60 gave some of the best results with the lowest error, possibly implying that the data has some yearly patterns. For the Elman RNN, increasing the window size consistently gave worse results. We are unsure why this is, but speculate that it may be our network topology as we had difficulty converging with such a large window size even when adjusting hidden layer neuron count. We graphed all detailed output in excel for visual analysis.

Code Instructions:

- Clone the FRM-Neural repository from GitHub onto personal computer
- Run IntelliJ and open the FRM-Neural project
- Navigate the file path: FRM-Neural > src > main > java > frm
- Run both ElmanFRM and TimeDomainFRM to view network outputs

GitHub Links to **Detailed Results (spreadsheet)** or **Result Graphs (PDF)**


Elman vs Temporal (Testing Data Set)
Actual
Predict Temporal W60
Error Temporal W60
Predict Elman W1
Error Elman W1
Predict Temproal W1
Error Temporal W1