Applied Mathematics Project

Deep Learning

Prediction of Time Series

Shay Malkin
Tal Ladijensky
Nir Titelbom

Contents

Introduction

Time series is a sequence of data points in chronological sequence, most often gathered in regular intervals.

Time series adds an explicit order dependence between observations: a time dimension.

In addition, time series forecasting is an important area of machine learning.

Forecasting involves making models fit on historical data and using them to predict future observations.

An important distinction in forecasting is that the future is completely unavailable and must only be estimated from what has already happened.

Examples for time series: average daily temperatures, stock value at the end of each business day, hourly electricity demand etc.

Time series analysis provides a body of techniques to better understand a dataset. Perhaps the most useful of these is the decomposition of a time series into 4 constituent parts: trend, seasonality, noise or randomness, and the cyclic movements. not all time series data will include every one of these time series components. For instance, audio files that are taken in sequence are examples of time series data, however they won't contain a seasonal component. On the other hand, most business data will likely contain seasonality.

There are different time series forecasting methods. For example: autoregressive moving average (ARMA) and artificial neural network (ANN). In this project we focus on artificial neural network.

Artificial Neural Network (ANN)

An ANN is a mathematical model for computation inspired by the structure of a biological neural network (the brain). The model is able to "learn" to perform tasks from given examples (training data), without being "hard coded" with specific instructions. This approach enables us to solve complex problems that otherwise would be impossible to solve (with direct instructions).

Activation functions are an extremely important feature of the artificial neural networks. They basically decide whether a neuron should be activated or not. Whether the information that the neuron is receiving is relevant for the given information. The activation function is often a nonlinear transformation that we do over the input signal. This transformed output is then sent to the next layer as input.

When we do not have the activation function the weights and bias would simply do a linear transformation.
Types of activation functions are: Sigmoid function, Tanh function, ReLU function etc.

A neural network propagates the signal of the input data forward through its parameters towards the moment of decision, and then back-propagates information about the error, in reverse through the network, so that it can change the parameters.
Backpropagation takes the error associated with a wrong guess by a neural network, and uses that error to adjust the neural network's parameters in the direction of less error.

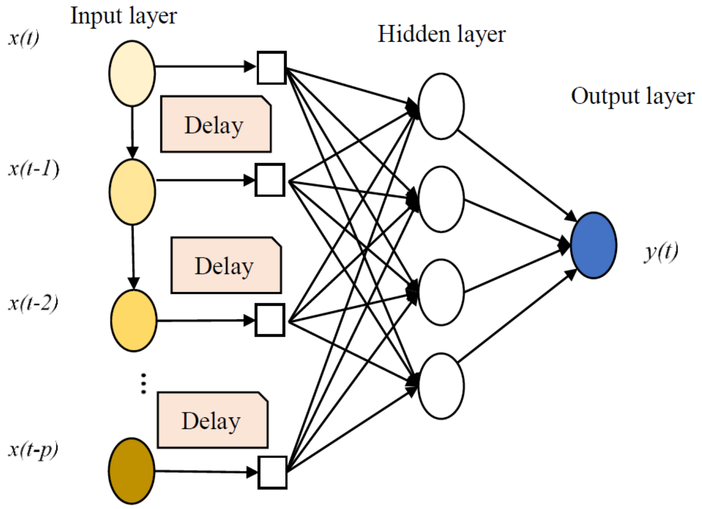
We will present two models for predicting a time series of the Euro-Dollar rate ratio.

Time Lagged Neural Networks (TLNN)

ANNs can be truly referred as model free structures because they do not need any prior knowledge about the intrinsic data generating process. ANNs are also favored due to their distinctive ability of nonlinear modeling with remarkable accuracies.

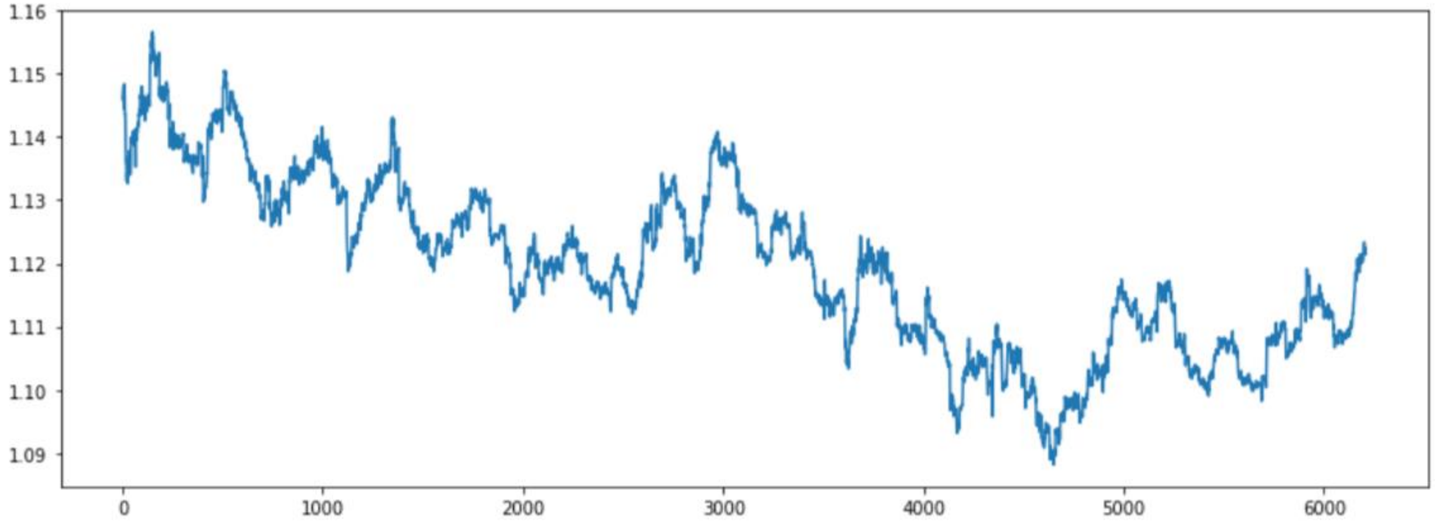
In the feed forward network (FNN), the input nodes are the successive observations of the time series, i.e. the target y_t is a function of the values $y_{t-i}, i = 1,2, \ldots, p$ where p is the number of input nodes. Another variation of FNN is the TLNN architecture that also widely used. In TLNN, the input nodes are the time series values at some particular lags. For example, a typical TLNN for a time series, with seasonal period $s = 12$ can contain the input nodes as the lagged values at time $t-1$, $t-2$ and $t-12$. The value at time t is to be forecasted using the values at lags 1, 2 and 12.

The next figure shows the architecture of a TDNN. The structure of the TDNN includes an input layer, one or more hidden layers, and an output layer. Each layer contains one or more nodes determined through a trial and error process of the given data, as there is no theoretical basis. In here, the figure present one hidden layer in the TDNN's structure.

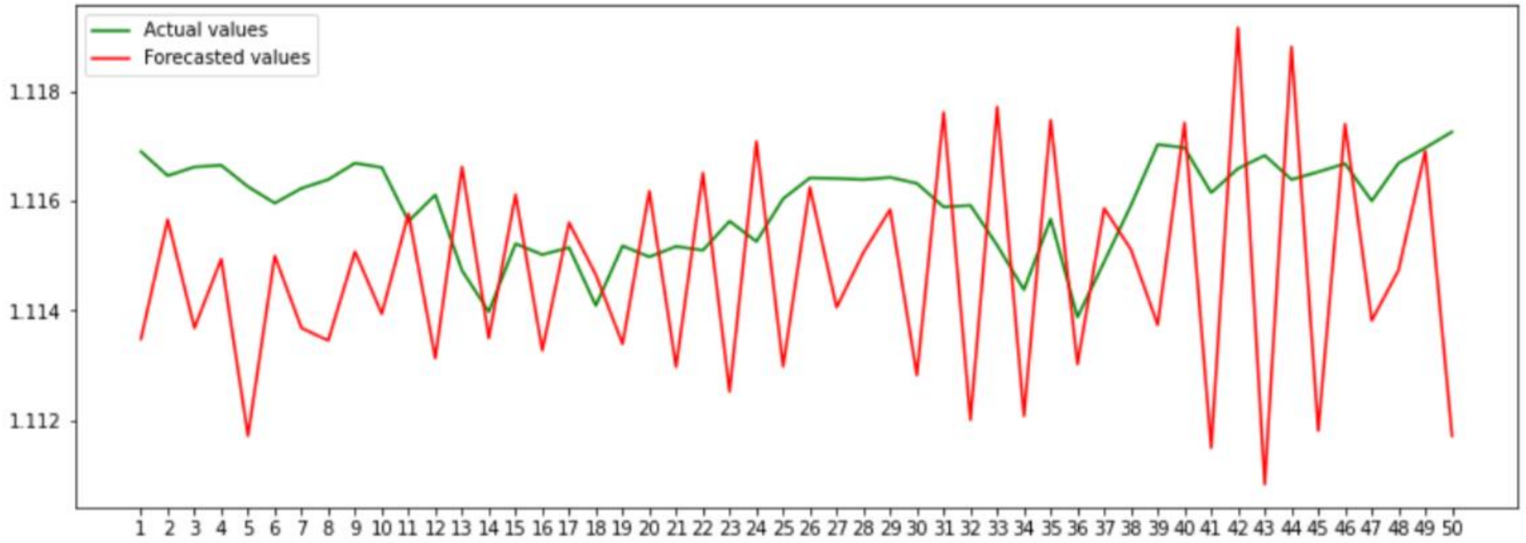


TLNN – Implementation in python

The input:



The prediction on the test data:



Long Short Time Memory (LSTM)

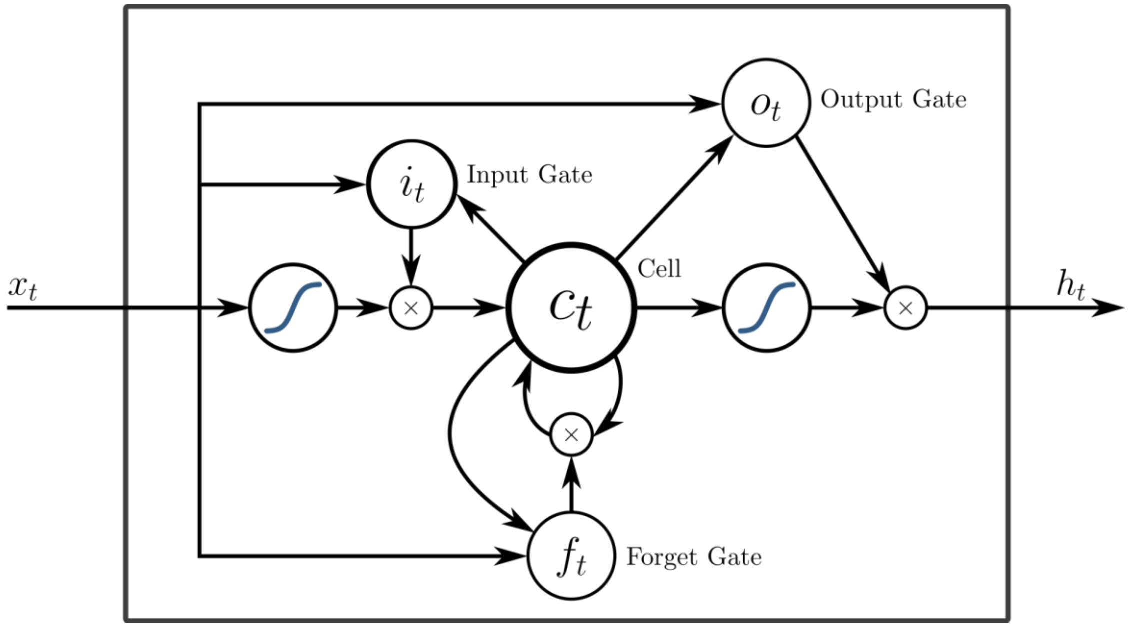
Recurrent neural network is essentially a fully connected neural network that contains a refactoring of some of its layers into a loop. That loop is typically an iteration over the addition or concatenation of two inputs, a matrix multiplication and a nonlinear function.

Tasks that RNNs are effective at solving are time series predictions or other sequence predictions that aren't image or tabular based.
RNNs effectively have an internal memory that allows the previous inputs to affect the subsequent predictions.
For example, the swift key keyboard software uses RNNs to predict what you are typing.
The Recurrent Neural Network consists of multiple fixed activation function units, one for each time step.

Each unit has an internal state which is called the hidden state of the unit. This hidden state signifies the past knowledge that the network currently holds at a given time step. An RNN has short term memory. When used in combination with Long Short Term Memory (LSTM) Gates, the network can have long term memory.
Instead of the recurring section of an RNN, an LTSM is a small neural network consisting of four neural network layers.
These are the recurring layer from the RNN with three networks acting as gates:

1. An Input gate, this controls the information input at each time step.
2. An Output gate, this controls how much information is outputted to the next cell or upward layer.
3. A Forget gate, this controls how much data to lose at each time step.

We can see here an example of LSTM architecture:

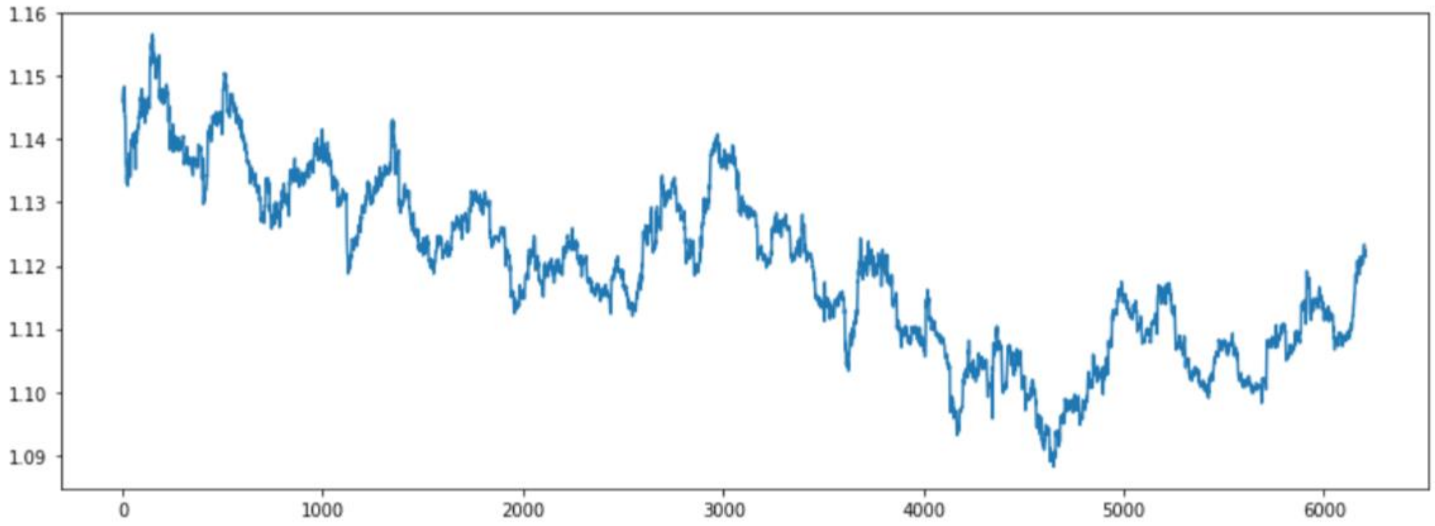


We implement the input data x_t in a regular feed-forward and then we feed it forward to our gates or compute on it an activation function (sigmoid, ReLU, etc.) and feed it to our memory cell c_t.

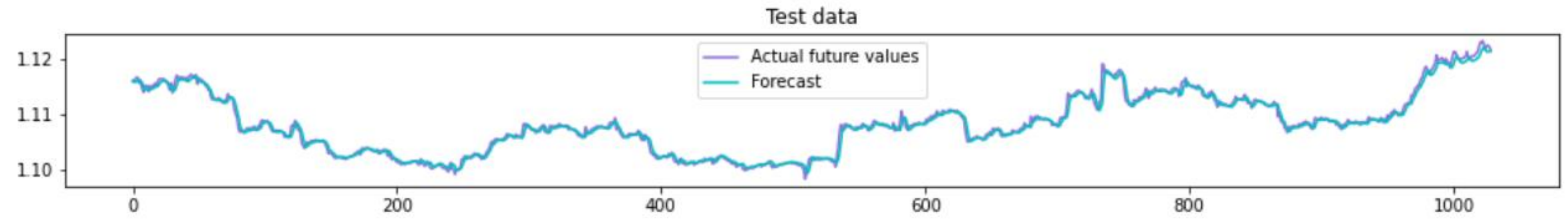
The arrows from the memory cell represent the contributions of the activation of the memory cell c_{t-1} and not c_t. In other words, the gates calculate their functions at time step t considering the memory cell at time $t-1$ and in that we achieve better handling of the long-term dependencies within the time series data.

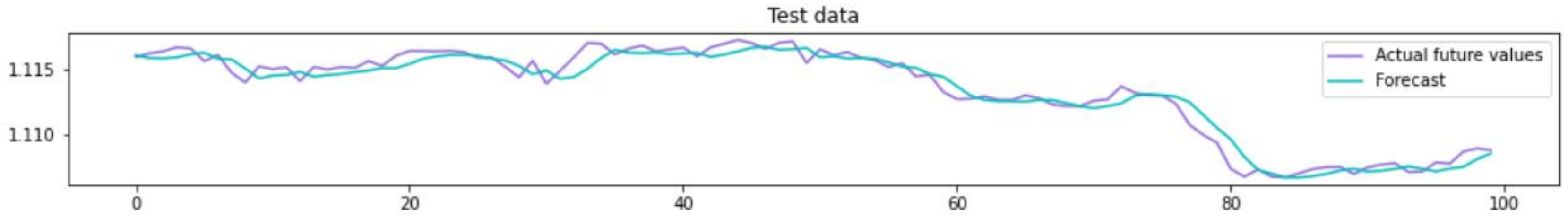
LSTM – Implementation in python

The input:



The prediction on the test data:





Conclusions

ANN has the ability to work with incomplete knowledge. After we finish the training process, our data may predict a 'good' output despite the incomplete information. The total loss of performance depends on the significance of the missing data.

The network can to be fault tolerant. If one or more cells of the network are corrupted, it will not prevent it from generating an output.

In addition, ANN has a distributed memory. The network's success is a derivative of the examples that are given to the network. Furthermore, ANN has the capability of parallel processing. The network has enough numerical strength, so that he can perform more than one process at the same time.

Finally, ANN has the ability to make machine learning. The network can learn from examples and make a decision based on similar data.

After trying many different configurations, we’ve come to the conclusion that the TDNN model doesn’t provide a good enough estimation for this particular time series. This does not imply that TDNNs are useless, for they have proven to be very effective in some cases.

In our case, we’ve found that an LSTM model works much better, providing a very accurate estimation with minimal error. After testing many different configurations, we’ve found that the best estimation is achieved by using 15 LSTM units with a 5 time step look-back. We were able to achieve a future prediction of a 1000 time steps with a Root Mean Square Error (with respect to the actual test data) of approximately 7×10^{-4}.