

Online non-linear prediction of financial time-series patterns

Mr Joel da Costa
Supervisor: Prof Tim Gebbie

University of Cape Town
MSc. Advanced Analytics (STA5004W) Research Proposal

This is a technical document, which serves as an accompaniment to the project proposal at:

The purposes for this document are twofold: firstly, to provide a more detailed overview and explanation of the project implementation, and secondly, to provide an updated state of affairs for the project.

Online non-linear prediction of financial time-series patterns	1
Current Situation	3
Future Situation	3
Current Issues/Concerns	3
Figures and Tables	4
Current Implementation Features / Algorithms	5
Milestones and Project Deliverables	6
Implementation Workflow	7
Process Flow	7
Data Flow	7
Model Training and Configurations (FFN/LSTM)	8
Research Workflow	9
Data	10
Requirements Specification	11
Meeting Minutes & Correspondence	12
16th June 2018	12

Current Situation

- Finished library implementations for FFN, OGD, SGD, SAE etc.

Future Situation

- Descent optimisations (e.g. Momentum, Dropout, learning rate schedule)
- Surrogate Data generation
- Data handling library to effectively parse stock data

Current Issues/Concerns

- Need to consider if and how changing constituents are accounted for
- Aspect of if or how often SDAEs are updated in the online learning process

Figures and Tables

Pre-training vs Activation Test

MNIST Classification accuracy was used to perform a cursory test of weight initialisation techniques (the different configurations were not optimised for each one in terms of learning rate etc., so this is more an indication rather than a notable result).

Hinton weight initialisation is a naive approach without taking the layer sizes or activations into account. By contrast, Xavier & He activations consider both, and are appropriate for Sigmoid and ReLU activations respectively - they are designed such that the variance of the input and output nodes should be roughly the same when the initialisation is done (prior to any form of training).

All configurations had

- Network layers with sizes [784, 250, 250, 500, 10]*
- Minibatch Size: 30
- Learning Rate: 0.01
- Log likelihood cost function
- SGD training was run for 10 epochs

Any RBM pre-training was run for 1 SGD epoch per layer, using MSE as the cost function.

Hidden Activations	Output Activation	Weight Initialization	RBM Pre-training	Classification Accuracy
ReLU	Softmax	He	No	97,97%
Sigmoid	Sigmoid	Hinton	Yes	97,31%
Sigmoid	Softmax	Hinton	Yes	97,08%
Sigmoid	Softmax	Xavier	Yes	96,90%
Sigmoid	Softmax	Xavier	No	95,29%
Sigmoid	Sigmoid	Hinton	No	59,54%
Sigmoid	Softmax	Hinton	No	52,41%

Clearly, the RBM pre-training makes a huge difference when the naive Hinton weight initialisation approach is used. However, when the Xavier & He initialisations are used, it does not appear to be necessary. Considering it's a time consuming process, it may not be the most effective configuration.

* A scaled-down version of the configuration from Hinton's initial paper, "Reducing the Dimensionality of Data with Neural Networks"

Current Implementation Features / Algorithms

Network Types:

- Feedforward Network
- Restricted Boltzmann Machine
- Stacked Auto Encoder

Learning Algorithms:

- Contrastive Divergence-1 for RBM pre-training
- Online Gradient Descent
- Stochastic Gradient Descent
 - With L1 & L2 regularisation

Neural Network Configurations:

- Activations: Sigmoid, ReLU, Softmax
- Initialisations: Normal, Hinton, Xavier, He (Normal & Uniform variations)
- Cost Functions: MSE, CE, Loglikelihood

Output Library:

- Effective graphing of training logs (epoch based performance, weight update rates, output reconstructions etc.)

Testing:

- Basic hyperparameter search method & outputs
- Automated Unit Tests for CD1, OGD and SGD with various network configurations (mostly to check after refactoring has been done)

Milestones and Project Deliverables

Finish Date	Item	Deliverable	Dependency
Done - subject to changes	1 Literature Review	Literature Review document	-
Done	2 Data Collection	Full dataset file (CSV)	-
No longer required	3 Wavelet Transformation	Wavelet dataset file	2
Done	4 SDAE Implementation	SDAE Julia Library	-
Done	5 Online FFN/LSTM Implementation	NN Julia Library	-
15/09	6 CSCV Implementation	CSCV Julia Library	4, 5
30/09	7 Synthetic Data Generation	Synthetic Datasets	
15/10	8 Test full implementation on synthetic data	Test results and full implementation library	2-7
15/11	9 Test full implementation on actual dataset	Test results	2-8
31/11	10 Statistical Analysis (incl. profitability of trading strategy)	Analysis Results	8, 9
31/12	11 Write up & Revisions	Full thesis document	9, 10

Implementation Workflow

Process Flow

The model process flow used to predict $T+1$ fluctuations will be as such:

1. Data Preprocessing - e.g. Wavelet Transformation
2. Dimension Reduction through SDAE
3. Online ANN - either LSTM or FFN

Data Flow

- Split data into training & hold-out sets (80/20)
- Use the training set for the various configurations under the CSCV framework (the N trials)
 - The trials don't need to all be planned up front - as long as the same C data combinations are used & the OOS/IS are captured, then the configuration space can be explored as a process.
- CSCV framework will provide the best performing strategies and their relative PBO
- A selection of these (at a low enough PBO and high enough performance) could be used to test the holdout data, and the best would be the reported model performance.

Model Training and Configurations (FFN/LSTM)

Training Process:

- Model Training through batch learning
- Model Updating through online learning
- Testing & Validation through online learning
- Hold-out assessment through online learning

Data Inputs:

- FFN: Fluctuations for OHLCV through K time periods (e.g. last day, week, month, 6 months).
- LSTM: Fluctuations for OHLCV for full index from T-1

Model Training and Parameter Choices (FFN vs LSTM)

FFN & LSTM:

- Batch training can be used for the training data (BP/BPTT)
 - Parameter choices of number of epochs, batch size and learning rate
- Network structures: layers & nodes
- Will need to decide whether or when to update SDAEs

FFN

- Inputs: Will need to decide on the K time window periods
- Online Learning: Variations on SGD as in Lit Review

LSTM

- Online learning method: BPTT variation - NoBackTrack, Unbiased Online Recurrent Learning, ADAM optimisation (notably - RTRL suffers from scalability issues)
- Training will require choices around truncation for BPTT & delays (though this is probably less arbitrary than window periods closed for FNN in a non-analytical manner)

Research Workflow

Each of the implementation steps will be achieved by following the process steps detailed below:

1. Problem Definition

The problem, and solution are defined here. In context of the above steps, there would be clear definitions of the input data to be used, output data to be expected, and the details of the technique used to achieve this.

2. Data Processing

At this step the data is prepared to be in an appropriate input format, as defined in step 1.

Data collection will be done in R, though post processing is expected to be done in Julia.

3. Data Exploration

Data exploration will be conducted in order to make sure the understanding of the data is correct - summary statistics may be used to this end (e.g. in the case of synthetic data sets, there will be clear expectations here). Visualisation will also be used as a technique to check this. Either of these may be done in R, Python or Julia, as considered most convenient for the task at hand.

4. Baseline Modelling

In the case where it is a modelling task, a baseline model will be used as a reference point going forward. For the Autoencoder, this will be a simple configuration that could be compared to PCA, or for the neural network it could be a single later configuration to measure improvements against.

5. Secondary Modelling

This stage is the more experimental stage of modelling. It is expected that numerous configurations for a model will be considered before arriving at the final model. In line with this, the modelling may go through it's own iterative machine learning process of parameter exploration, training and validation.

6. Testing

In some instance, more generalised statistical testing will be performed, e.g. statistical arbitrage tests.

7. Process Preparation

Once the above steps have been completed, the current stage will be prepared as an input to the following one - e.g. a consumable model, or library which performs a task.

Data

Data will be collected from the Bloomberg terminal using R. The full data requirements and specifications, as set out in the project proposal, are below:

- A full dataset will need to be collected from Bloomberg
- The dataset will comprise of OHLCV daily data for 20 years of the JSE ALSI
- Synthetic data cases (e.g. Monte Carlo simulations) will also be considered in order to discuss issues encountered with in-sample versus out-of-sample backtesting
 - Examples of such data cases would be where stocks are all increasing/decreasing over time, or both for a combination of stocks.
- The Daily TRI (Total Return Index) OHLCV for 20 years will be considered for all stocks.

Requirements Specification

Hardware

A Macbook pro will be used for the crux of the development, with the following specs:

- 2.8 Ghz Intel Core i7
- 16 GB 2133 Mhz LPDDR3
- SSD

Software & Packages

Julia will be the primary programming language used to develop the project, though Python and R will also be used interchangeably for Exploratory Data Analysis, and as needed otherwise.

The list of external libraries within these languages will be added here as the project progresses.

Meeting Minutes & Correspondence

16th June 2018

Literature Review

- Include:
 - Wilcox & Debbie Hierarchical Paper
 - Crutchfield & Farmer: Geometry of Time Series
 - F Takens: Detecting strange attractors in turbulence
 - Reference Riaz

Implementation

- Key implementation point was the data segmentation choices
- LSTM network would resolve this point for us (CNN convolution could also potentially work)
 - Non LSTM will run risk of correlating times with stocks
- Representation needs a mixture of time scales as well as the features
- Will be considering data for the entire index at any time point t
 - Will need to consider how to address the changing constituents
- Aspect of how often the feature selection is updated (e.g. SDAE in online model)
 - Does it get updated in the online model ?
 - SDAE result should be similar to edge detection which will give set of predictions essentially
- Use known test cases for CSCV method development

Data

- Use simple noise with Gaussian distribution & long trends to develop network models (6 variations of increasing, decreasing and flat with low or high variability)
- Build backdated tests module with surrogate data afterwards
- Not going to try intraday data due to different nature of problem
- Implementation of either a flat file or MongoDB
- Important to include Volume data
 - Volume time -> might be far more predictable; volume and variance linked

Admin

- Single document - table of contents (nice to have) with authors; sections etc. ; full doc with versioning; PDF + latex (always send both)

Further Correspondence:

- * Iterate over FFN/LSTM implementations rather than starting with LSTM
- * For the meantime, assume the models do not differ fundamentally enough to be considered as separate for the PBO calculations