CLASSIFICATION BASED FINANCIAL MARKET PREDICTIONS

Deep Neural Networks

Diego Klabjan
Professor, Industrial Engineering and Management Sciences
Founding Director, Master of Science in Analytics

Northwestern McCORMICK SCHOOL OF ENGINEERING
Team

• Research in
  ▪ Deep learning (models, algorithms, analysis)
  ▪ Natural language processing (ontologies, knowledge management)
  ▪ Machine learning (robust PCA, large-scale optimization)

• Director, Master of Science in Analytics
  ▪ Full-time single cohort program
  ▪ All aspects of analytics, data science, artificial intelligence

• Joint work with Mark Harmon
  ▪ Ph.D candidate
  ▪ Lida Zhang, Research Assistant
MS in Analytics at Northwestern

- Full-time
  - On-campus in Evanston
- Fifteen months to complete
  - Sept 2016 to Dec 2017
- Small cohort
  - Maximum 40 students
- From SQL to machine learning to business

- Practical
  - Two company sponsored project
  - Summer internship
- Rigorous
  - Design of algorithms in models
  - Implementations
  - Artificial Intelligence
Motivation
Long-term Goal

- The trader bot
  - Automatically trade
- Precursor
  - Predict movement/direction of prices
- Is the stock price
  - Going up/down by more than
    - 10% or one standard deviation
    - In the next short period of time
- Challenge
  - Correlation of assets
Strategy Engineering

How can we engineer a strategy producing buy / sell decisions?

Are securities going
- up by 10%
- down by 10%
- unchanged
in the next few minutes?
Are securities going

- one and two standard deviations
- more than two standard deviations
- unchanged

in the next few minutes?
Traditional Pipeline

Raw features 5 minute mid-prices

- Lagged price differences from 1 to 100
- Moving price averages with window size from 5 to 100
- Pair-wise correlation of returns

Engineered features

Normalized features

Feature selection

Labeled feature set

Feature set consists of 9,895 features

Select 100 features

Labels
- Positive
- Neutral
- Negative market returns
Old vs New

- Traditional
  - 100 features

- Deep learning
  - Use all features
Concepts

- Feature vector at time timestamp $t$
  - Prices
- Model has to capture temporal dimension
  - Interactions
    - In time and among assets modeled
- Temporal aspects
  - Recurrent neural networks
- Feature selection
  - 1 dimensional convolutional neural networks
Goal

- Predict security price given
  - Past prices
  - Prices of other securities
- Five labels for each security
  - No difference in price
  - Small increase/decrease (with a standard deviation)
  - Large increase/decrease (more than a standard deviation)
- Varying prediction time horizon
Models
Characteristics

- Difficulties of financial time series
  - Correlations
  - Drift within the series
  - Infinite time series
    - Standard approaches assume finite sequences
  - Sequence length undetermined
- Our approach
  - Vary sequence length on output
  - Overlap sequences
  - Train on one year and predict on one week
    - Walk forward (validation is last week of the year)
Convolutional Neural Networks

- Learns filters
  - Short sliding windows
  - Weighted sums
    - One for each filter
- Advantageous for
  - Observing groups of data close in time
    - Model captures moving averages, etc
  - Observes similarly to humans on images
## Input Features

<table>
<thead>
<tr>
<th>Stock</th>
<th>$\delta_1$</th>
<th>$\delta_2$</th>
<th>$\delta_3$</th>
<th>$\delta_4$</th>
<th>$\delta_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stock 1</td>
<td>0</td>
<td>-0.27</td>
<td>0</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>Stock 2</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>-0.52</td>
<td>0</td>
</tr>
<tr>
<td>Stock 3</td>
<td>0</td>
<td>0.25</td>
<td>0.5</td>
<td>0</td>
<td>-0.25</td>
</tr>
<tr>
<td>Stock 4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0</td>
</tr>
<tr>
<td>Stock 5</td>
<td>-0.25</td>
<td>-0.28</td>
<td>0.5</td>
<td>0</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Recurrent Neural Network

- Created specifically for time series
- Model flow
  - Input to output
  - Previous cell passes encoding to next time step
  - Chooses information to retain
Sequence to Sequence Network

- Variable size input and output
  - Output prediction length flexible
- Two parts
  - Encoder
  - Decoder
Combined Model

INPUT

FEATURES

CNN

CNN

CNN

SEQUENCE LEARNING

LSTM

LSTM

LSTM

OUTPUT

$y_1$

$y_2$

$y_T$
Adaptive Prediction Length

- Function that captures ‘reliability’ of prediction
  - Max probability value
    \[ g_t = \max_i y_{i,t} \]
  - Entropy
    \[ g_{i,t} = \sum_i y_{i,t} \log y_{i,t} \]
  - Total Variation
    \[ g_t = \max_i |y_{i,t} - y_{i,t-1}| \]
  - Wasserstein
    \[ \min_{w \geq 0} \sum_j w_j (y_j t - y_{j,t-1}) \]
    \[ |w_p - w_q| \leq |p - q| \quad \forall p, q \]
Loss Functions

• Contains two parts
  - Accuracy loss (log loss)
  - Time prediction penalty

• Balance loss to
  - Increase F1 performance
  - Predict as far in time as possible

\[
\sum_{g_t(X, \theta) \geq \tau} KL(Y_t \| p_t(X, \theta)) + \lambda \sum_{t} \max(\tau - g_t(X, \theta), 0)
\]
Improving Robustness

- Normal prior on weights
  - Compute expected value of each weight
- Standard
  - Compute expected value
  - Compute standard deviation for each weight

\[
\min_{\bar{w}, \bar{\sigma}} E_{X,Y} E_{Z \sim N(\bar{w}, \bar{\sigma})} KL(Y | p(X, z))
\]

\[
\min_{\bar{w}, \bar{\sigma}} E_{X,Y} E_{Z \sim N(0, 1)} KL(Y | p(X, \bar{w} + z\bar{\sigma}))
\]
Adaptive Computational Time

- Alex Graves (2017): Adaptive Computation Time for Recurrent Neural Networks
- Developed for RNN
  - Extended to sequence-to-sequence
- Embed attention
  - In decoder add

\[ \text{att}_t = \sum_i \alpha_{it} \text{hidden}_i^{enc} \]

\[ \alpha_{it} = f(\text{hidden}_{t-1}^{dec}, \text{hidden}_i^{enc}) \]
Evaluation
Data

• Two datasets
  - Commodity securities
  - ETF’s
• Data contains
  - 5 minute tick data
  - Roughly 14 years
  - No volume information
• Data imbalanced
  - Few samples outside of two standard deviations
Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Average F1 Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvLSTM</td>
<td>0.324</td>
</tr>
<tr>
<td>10 ticks</td>
<td>0.250</td>
</tr>
<tr>
<td>20 ticks</td>
<td>0.279</td>
</tr>
<tr>
<td>40 ticks</td>
<td>0.271</td>
</tr>
</tbody>
</table>

After first few weeks of training convolutional LSTM works much better.
Enhancements
Prediction Reliability

- Maximum prediction probability above 0.35
  - Challenge how to set up training set
  - Specific loss function
- Stock price does not change
  - Reliable ‘longer term predictions’

![Bar chart showing average stopping time for various stocks (Percentage Class Neutral)](chart.png)
• 'Easier’ to predict stocks with no large swings
Conclusions on Deep Networks

• Pros
  ▪ Works for complex models
  ▪ No need for feature selection
  ▪ Improved accuracy

• Cons
  ▪ Long time to train
  ▪ Hyperparameter nightmare
  ▪ Expertise
    • Models
    • Implementation
    • Tricks of trade

• Convolution with sequence to sequence works best
• Challenge with unbalanced data
Ongoing Work

- Further enhancement to ACT
  - Use for varying prediction sequences
- Drift detection
  - Autoencoders
  - ACT
- Deep reinforcement learning
  - Place orders
  - Or perhaps contextual bandit?
Thank you very much!

@dklabjan
d-klabjan@northwestern.edu