

LSTM を用いた株価変動予測

Stock Market Prediction Using LSTMs

松井 藤五郎 *1*2 汐月 智也 *2
Tohgoroh Matsui Tomoya Shiotsuki

*1 中部大学 生命健康科学部 臨床工学科
Department of Clinical Engineering, College of Life and Health Sciences, Chubu University

*2 中部大学 工学部 情報工学科
Department of Computer Science, College of Engineering, Chubu University

This paper proposes a method for stock market prediction using LSTM (Long Short-Term Memory) networks. LSTMs are a special kind of RNNs (Recurrent Neural Networks) that have loops in them and allow information to persist. We also show an experimental result to model a time series of the rate of return after one minute.

1. はじめに

株価は時々刻々と変化しており、その動きを予測することは難しい問題である。中長期的には業績や財務状況等のファンダメンタルズによって株価が動くと考えられるが、短期的な株価の動きはファンダメンタルズでは説明することができず、より難しい問題である。

本研究では、1分ごとの株価の動きを予測する問題を対象とする。ある銘柄について、実際の1分ごとの株価の動きを図1に示す。株式市場は09:00に始まり、11:30から12:30まで1時間の昼休みを挟んで15:00に終わる。09:00から11:30までを前場、12:30から15:00までを後場といい、前場と後場で最初に成立した取引を寄り付き、最後に成立した取引を引けという。この図のように、昼休みまたは終了後から翌営業日の開始時刻までの間も時間外取引や株式市場を介さずに当事者間で売買が行われる相対取引などで株価は動いており、寄り付きの価格はその前の引けの価格に比べると大きく異なる場合もある。

近年、深層学習 (deep learning) の発達に伴って、ニューラル・ネットワークを用いた手法が有力になっている。そこで本論文では、時系列データを学習できるニューラル・ネットワークである LSTM (Long Short-Term Memory) を用いて、株価の動きを予測する方法を提案する。

2. LSTM

LSTM [Hochreiter 97, Olah 15] は、時系列性の有するデータを学習できるニューラル・ネットワークであるリカレント・ニューラル・ネットワーク (RNN) の一種である。

RNN は、前回の出力を現在の入力に追加するものであり、時間方向に展開すると静的なニューラル・ネットワークと見ることができる。RNN は、この時間方向に展開した静的なネットワーク上で誤差逆伝播を用いて学習を行うものであるが、系列が長くなると重みが掛けられる回数が多くなり、勾配が消失してしまう。これによって、長期依存を学習できなくなるという問題が生じる。これに対し、LSTM では、重みを掛けずに誤差を逆伝播させることによって、長期依存を学習できなくなる問題を解消している。

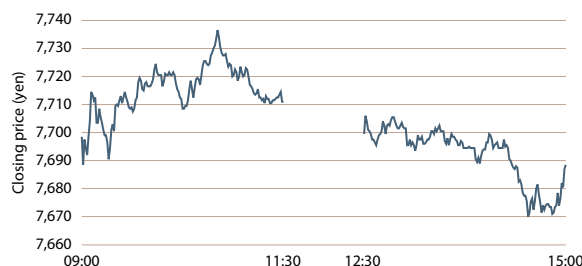


図1 1分ごとの株価の動き

時刻 t における LSTM への入力値を x_t 、セルの状態を C_t 、出力値を h_t とすると、まずはじめに、 h_{t-1} と x_t から忘却ゲート (forget gate) の値 f_t を求める。

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

ここで、 W_f は f のための重み、 b_f は f のための切片、 $\sigma(\cdot)$ はシグモイド関数を表す。

次に、入力調整ゲート (input modulation gate) の値 i_t とセルの状態の候補値 \tilde{C}_t を求める。

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3)$$

これらの値を用いて、セルの状態 C_t を求める。

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

最後に、出力ゲート (output gate) の値 o_t とセルの状態 h_t を求める。

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

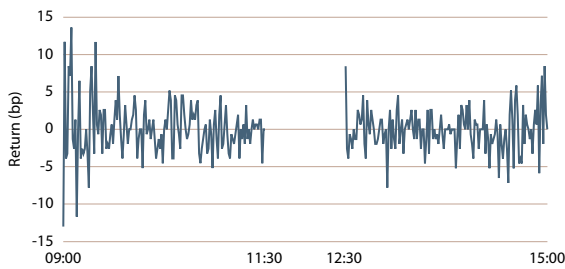


図2 1分ごとの利益率の動き

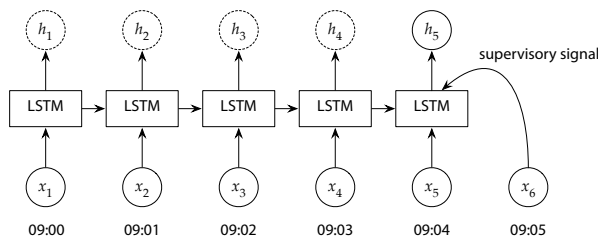


図3 LSTM への入力、出力、教師信号

3. 提案手法

本論文では、1分ごとの株価の動きを予測するために、株価ではなく、利益率の時系列データを LSTM を用いてモデル化し、予測することを提案する。株価そのものを予測する方法では、上場来高値（または安値）のような過去に経験したことのない株価になると正しく予測できなくなってしまう。そこで、株価の代わりに利益率を用いる。図1と同じ株価を利益率に変換したものを図2に示す。

連続した1分後利益率を $n+1$ 個並べ、最初の n 個を LSTM への入力、最後の1個を教師信号として用いる。図3に、 $n=5$ のときの LSTM への入力、出力、教師信号を示す。LSTM は時間方向に展開し、静的なネットワークとして表現されている。先頭の利益率が順に LSTM に入力し、 n 個目の利益率を入力したときに LSTM から出力された値が $n+1$ 個目の利益率の予測値となるように、 $n+1$ 個目の利益率を教師信号として LSTM を学習させる。このとき、11:30、12:30 のような不連続な系列はデータに含めないようにする。

図の例 ($n=5$) では、09:00 から 09:05 までの6個の連続した1分後利益率を用意し、09:00における1分後利益率から順に入力する。5個目の値である09:04における1分後利益率を入力したときの出力が09:05における1分後利益率となるように、6個目の値を教師信号として LSTM を学習させる。

4. 実験結果

提案手法の有効性を確認するため、実験を行なった。LSTM は TensorFlow [Abadi 16] の BasicLSTMCell を用いて実装した。TensorFlow の BasicLSTMCell では、忘却ゲートの値 f_t に忘却バイアスが加えられる。

対象銘柄をトヨタ自動車とし、2015年10月20日から12月4日までの1分ごとの1分後利益率を用いて、25日間を訓練期間として LSTM モデルを学習した。また、訓練期間の直後の1日をテスト期間とした。訓練データとテストデータのそれぞれ

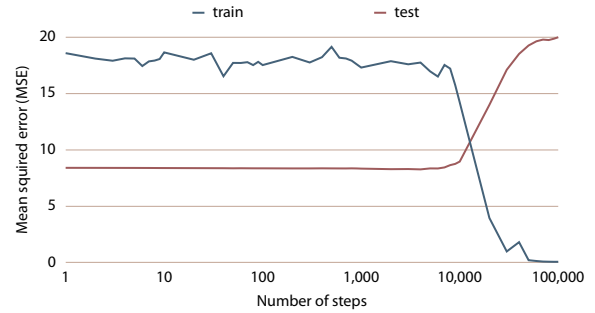


図4 平均二乗誤差

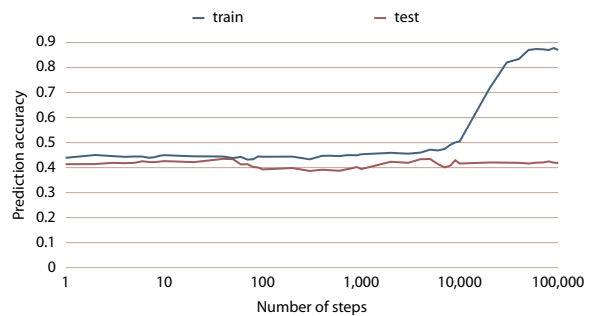


図5 符号一致率

に対して、モデルが出力した値と実際の値の平均二乗誤差と符号一致率を求めた。隠れユニットの数を128個とし、学習率を0.01、訓練データのサンプリング率を0.2、忘却バイアスを0.2とした。

結果を図4と図5に示す。図4は LSTM の出力値と真の値との平均二乗誤差を、図5は符号一致率を表している。訓練データに対しては、平均二乗誤差はほぼゼロとなり、符号一致率は87.0%になった。符号一致率は、正、ゼロ、負の3通りで求めており、この期間には1分後利益率がゼロになっているものが9.8%含まれている。したがって、正または負のデータについてはほとんど正解している。しかしながら、テストデータに対しては平均二乗誤差が増加し、符号一致率の上昇は見られなかった。つまり、過学習が生じていると考えられる。

5. まとめ

本論文では、1分ごとの株価の動きを予測するために、LSTM を用いて1分ごとの1分後利益率をモデル化する手法を提案した。実験の結果、LSTM は1分後利益率の時系列をモデル化できることが確認された。ただし、過学習が生じ、テストデータに対する予測ができない点が課題である。

参考文献

- [Abadi 16] Abadi, M. et al.: TensorFlow: A system for large-scale machine learning, *OSDI 2016*, pp. 265–283 (2016)
- [Hochreiter 97] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780 (1997)
- [Olah 15] Olah, C.: Understanding LSTM Networks, colah's blog (2015)