

深層学習を用いた日経平均株価の予測手法の提案

Proposal for the Forecasting Method of Nikkei Stock Average by Using Deep Learning

1741010 石毛 康太郎

Koutarou ISHIGE

指導教員 秋葉 知昭

Deep Learning techniques are rapidly being developed in many fields including stock price forecast. In this study, I considered Nikkei Stock Average forecast using deep learning. I proposed forecast idea of the Nikkei Stock Average forecast.

1. 緒言

日経平均株価は現在の日本経済においてにおいて日本の景気における良し悪しを判断する上で重要な指標の1つである。2020年春に起きたコロナショックと呼ばれている新型コロナウイルス感染症(COVID-19)の感染拡大は、日本経済に大きな影響を与えた。日経平均株価は中国が武漢市を閉鎖した2020年1月23日の終値である23750円から3月9日の安値である16552円まで値下がりした[1]。このように日本を含めた世界情勢は日経平均株価に色濃く反映されている。特定の上場銘柄を予測するのではなく、将来の日本経済の良し悪しを判断するために東京証券取引所第一部に上場する225銘柄から算出される日経平均株価を予測したいと考えた。

近年、AIの進化によりコンピューターによる物事の部類分けや予測などといった技術が大きく発展した。その技術は上述した新型コロナウイルス感染症の感染者数のシミュレーションや、企業の顧客分析など多岐にわたる。その中でAIによる株価予測は昔から多くの投資家が考えている夢のようなものであると言える。しかし近年のAIの進化によってAIによる日経平均株価の予測が夢のようなものではなく現実味を帯びてきている[2]。

本研究では、ここ近年の日経平均株価の予測精度が大きく向上した要因である、LSTMを用いた予測手法を提案する。さらにディープラーニングの学習時の大きな役割を果たす最適化関数について、昔から用いられているものからここ数ヶ月で知名度の上がってきた最新のものを含めた4つを用いて予測を行い、結果を比較する。

現在ディープラーニングを使った日経平均株価予測はLSTMが大半であり[3]、このLSTMが一般的となる以前はCNNを使用していた。CNNとは画像処理を得意とするディープラーニングの一種であり、時系列データのグラフを画像に変換し、画

像を読み込ませることで予測を行っていた[4]。

2. 予測手法の提案

2.1 予測手法の概要

本研究では時系列データの扱いに長けているLSTMを使用する。1日の終値50日分のデータを入力し、翌日の51日目の終値を予測する。最適化関数はSGD,RSMprop,Adam,RAAdamを用いた結果を比較し、考察する。フレームワークはPyTorchを使用する。バージョンは1.6.0である。プログラミング言語はpythonを使用する。

なお実際のソースコードは50個の1連の時系列データを入力して51個目の1つの値を予測するため、sin波の予測を行うソースコードに近い。日経平均株価をsin波と似た波であるというのは厳密に無理があるかもしれないが、一定の値(底または天井)で反発を繰り返し、反発時以外には値の上昇と下降のトレンドがあるという特徴から予測可能だと判断した。

2.2 収集データ

本研究ではデータ収集にSBI証券が無料で提供している取引ツールHYPER SBIを使用した。HYPER SBIでダウンロードできる、1991年から2019年まで29年間のデータ7123日分のデータを収集した。ダウンロードしたデータは日付、始値、高値、安値、終値、5日平均、25日平均、75日平均、VWAP、出来高であるが、本研究では終値のみを使用する。1回の学習で50のデータを入力する。

2.3 パラメータ及びモジュールの設定

最適化関数はディープラーニングにおいて一般的なSGD,RSMprop,Adam,RAAdamの4つを用いた結果を比較する。誤差関数はMSELossを使用した。学習率は分散を抑えるためにSGDのみ0.1、他の3つは0.01を初期値とした。エポック数は2000と

3000 のどちらかを過学習にならないように使用した。バッチサイズは 10 とした。なお計算環境は Anaconda で PyTorch を動かす。統合環境は Jupyter Notebook であり、バージョンは 1.0.0 である。OS は windows 10, CPU は Intel(R) Core(TM) i7-7700 3.60GHz, メモリは 16GB である。

3. 結果及び考察

3.1 出力データ

出力データはエポック数, loss, 正解率, training_accuracy, test_accuracy を出力した。loss のグラフにおいて SGD は学習曲線が見られなかった。逆に他の 3 つの最適化関数は学習曲線が確認できた。

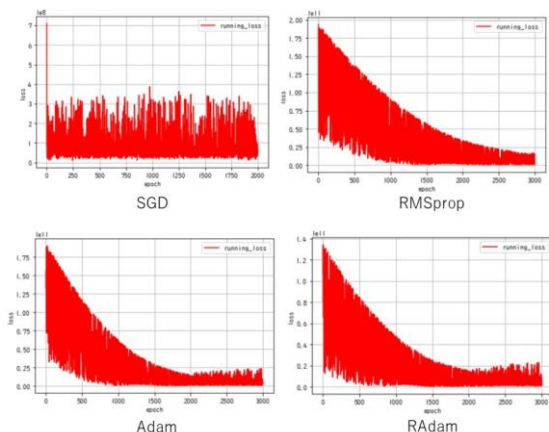


図1 各最適化関数における loss の推移

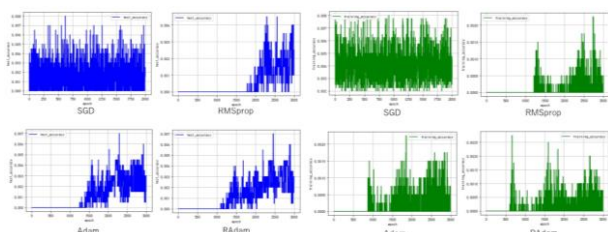


図2 各最適化関数における training_accuracy, test_accuracy の推移

3.2 最適化関数の比較

まず loss のグラフを見る。SGD では学習曲線が確認できず、学習ができていないと見える。他の 3 つは学習曲線が確認でき、大まかには似た形をしているが RMSprop は同じエポック数なのにに対し学習状況が RAdam, Adam より遅いように見える。RAdam は学習の初期段階での適応学習自体の分散を抑えた loss 値から学習を始めるが、最終的な loss の値に Adam と大きな違いは見られなかった。

次に training accuracy と test accuracy を見る。この 2 つは学習時とテスト時に予測と正解ラベルの

誤差が 10 円以下を正解としてその正解数の推移をグラフで表したものである RMSprop の正解数が Adam, RAdam と比べて低く見えるが loss のグラフから学習速度が遅いことを考慮すると学習状況に対しての正解数に大きな違いは見られない。

3.3 考察

本研究の手法では SGD を使用した場合うまく学習ができず, Adam, RAdam, RMSprop を使用した場合は学習ができた。SGD は極小値から抜け出せていない状態にあると考えられる。

RMSprop は学習が確認できた 3 つの中では学習速度が遅かったがこれは Adam と RAdam が RMSprop に移動平均線で振動を抑制した仕組みのものであるからだと考えられる。

本研究で最も高い正解率は Adam と RAdam の 7% となった。最も高い正解率を出す前後の推移はほとんど同じであり、学習の後半では 1.5% から 4% の間で落ち着いている。

4. 結言

本報告では PyTorch を用いた日経平均株価の予測を行い、結果を比較、考察をした。最も良い正解率は Adam と RAdam の 7% となった。学習と予測している過程は確認できたが、正解率 7% で十分に予測できているとは言えない。よって入力データの変更の改善によって十分な精度を得る必要がある。

これらの結果から LSTM で Adam, RAdam, RMSprop を用いた場合に日経平均株価の予測はできていることが確認できた。RMSprop の学習速度が Adam, RAdam より遅いことを考慮すると、株価などの指標の予測には LSTM と Adam, RAdam を用いた手法が有効であると言える。本研究では RAdam の有用性が十分に確認できなかったため、Adam と RAdam の優劣をつけることができなかった。今後は更新によってさらに精度の高い予測が可能となると考える。

文献

- [1] SBI 証券関東財務局長 (金商) 第 44 号: HYPER SBI
- [2] 奥村順哉: ディープラーニングによる経済記事テキストデータを用いた株価予測, 高知工科大学, 平成 26 年度学資学位論文 (2014)
- [3] 小松隆行: 深層学習による株価予測に関する実験考察, 北海道大学研究紀要第 46 号 (2018)
- [4] 宮崎邦洋, 松尾豊: 深層学習を用いた株価予測の分析, 東京工業大学系研究科技術経営戦略専攻 (2017)