

Deep Belief Network を用いた 日経平均株価の予測に関する研究

Forecasting the Nikkei Stock Average using Deep Belief Network

小牧 昇平^{1*} 白山 晋¹
Shohei Komaki¹ Susumu Shirayama¹

¹ 東京大学大学院 工学系研究科
¹ School of Engineering, The University of Tokyo

Abstract: In this study, we propose a new forecasting method of a financial time series based on Deep Belief Network (DBN) by enhancing the approach of the Chao *et al.* First, a new topology for a regression training is proposed. Second, we forecast a Nikkei Stock Average renewing a training term. Third, Self-Organized-Map (SOM) is introduced for reducing the computational time in DBN. It is shown by some experiments that some improved performance indexes can be obtained, and reduction of the computation time is achieved.

1 はじめに

2013年暮れに日経平均株価が6年ぶりの水準に回復したことや、少額非課税制度（NISA）が2014年1月から導入されたことなどから、個人の株式などへの投資に対する関心が高まっている。以前から経済時系列を予測するために様々なアプローチが提案されてきたが[11]、このような状況が相まって、経済時系列予測の研究が加速している。しかしながら、株価の変動など、経済時系列の変化は非定常かつ非線形性が強く、予測は非常に困難である。

一方、ニューラルネットワーク（NN）による予測を発展させた Deep Learning(DL) と呼ばれる手法が時系列データの予測に対する有力な手段として注目されている。Deep Learning は画像認識や音声認識などの分野で、そのパターン認識性能の高さが示されているが[7, 3, 6]、それを時系列データに展開しようという試みである[5]。

例えば、Chao ら[5]は、DLを用いることで経済時系列のパターンも学習できると考え、DLの一手法である Deep Belief Network (DBN) [8, 9] を用いて数日間の為替レートの変動をパターン認識し、翌日の為替レートを予測している。従来のフィードフォワードニューラルネットワーク (FFNN) と比較し、良い結果となることを示している。彼らの研究では、ある訓練期間で DBN を学習し、その学習された DBN を訓練期間に続く予測期間の全てに対し、常に予測しようとする日の

直前までのデータを使用して予測を行っている。このため、訓練期間と予測期間の隔たりが問題になる。また、DBN に入力する各データは訓練期間全体の最大値と最小値に対して各データがほぼ横ばいのデータになるため、パターン抽出が難しくなるという可能性が考えられる。さらに、訓練に非常に多くの計算時間を要する。

以上から、DLを用いた方法は有力であるが、適切な訓練期間の選択、計算時間の低減という課題があることがわかる。また、複数のデータの組み合わせなどによる予測精度の向上が望まれている。例えば、Chao らのネットワークトポロジーでは複数のデータ入力に対応していないが、他の研究では、日経平均株価と FX を組み合わせることにより、日経平均株価の予測精度が向上することが示されている [2]。

本研究は、Chao らの研究を発展させるものである。上述した課題の解決を試み、予測精度の向上と計算時間の低減を目指す。

具体的には、以下の3つの方法を提案する。

- (a) 日経平均株価の予測を行うが、日経平均株価の時系列データに別のデータを組み合わせることで複数の時系列データを入力できるようにする
- (b) 適切な訓練期間の選択として、訓練期間を常に最新のものに更新し続ける
- (c) 自己組織化マップ (SOM) を用いて訓練期間全体のデータから部分的なデータを取り出し、この部分的なデータのみで学習を行うことで効率的な学習を可能とする

*連絡先：東京大学大学院工学系研究科システム創成学専攻
〒113-8656 東京都文京区本郷 7-3-1
E-mail: sirayama-lab@nakl.t.u-tokyo.ac.jp

2 関連研究

2.1 RBM と Continuous RBM

はじめに, Restricted Boltzmann Machine(RBM)[10] について説明する.

RBMにおいて, 入力データベクトルを入力する層を可視層 \mathbf{V} , 特徴ベクトルが抽出される層を隠れ層 \mathbf{H} とする. 各層のノード数をそれぞれ l, m とし, k 番目の入力データの各層の状態ベクトルを $\mathbf{v}^k, \mathbf{h}^k$ とする. \mathbf{V} に属するノード v_j と \mathbf{H} に属するノード h_i の間はすべての組み合わせにおいて重み w_{ij} のリンクがあるが, 同じ層に属するノード間ではリンクを持たない. また, すべてのノードはバイアスを持ち, v_j のノードのバイアスを b_j , h_i のノードのバイアスを c_i とする. さらに, 各ノードはシグモイド関数

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad (1)$$

により入力値を変換し出力値を得る. すなわち,

$$P(h_i | \mathbf{v}^k) = \text{sigmoid}(c_i + \sum_j v_j w_{ij}) \quad (2)$$

$$P(v_j | \mathbf{h}^k) = \text{sigmoid}(b_j + \sum_i h_i w_{ij}) \quad (3)$$

である.

RBM の学習は対数尤度の最大化を目的とした勾配法により行われる. 具体的には η_w, η_b, η_c を学習率として以下の更新式により行われる.

$$\Delta w_{ij} = \eta_w (\langle v_j h_i \rangle_0 - \langle v_j h_i \rangle_A) \quad (4)$$

$$\Delta b_j = \eta_b (\langle v_j \rangle_0 - \langle v_j \rangle_A) \quad (5)$$

$$\Delta c_i = \eta_c (\langle h_i \rangle_0 - \langle h_i \rangle_A) \quad (6)$$

ここで, $\langle \cdot \rangle_A$ は式 3 と式 2 を交互に A 回ずつ用いた際の可視層と隠れ層の値を用いたノードごとの平均を表す. 多くの場合 $A = 1$ でも十分な学習成果を実現することができる. 学習された RBM に対し, データベクトルを入力すると, 隠れ層にはそのデータの特徴ベクトルが出力される.

RBM の入出力は一般的には 0 か 1 のバイナリである. 経済時系列に対してはクラス分類に用いられる. 次節で示す Chao らの方法では, 経済時系列などの連続値に対応させるために, Hsin ら [4] によって提案された Continuous RBM(CRBM) を用いている. 以下において CRBM について説明する.

状態ベクトルの成分を $\{s_j\}$ とする入力層に対し出力層のニューロン i の出力値を s_i とする. このとき以下のように s_i の値を定める.

$$s_i = \phi_i \left(\sum_j w_{ij} s_j + \sigma \cdot \mathcal{N}_i(0, 1) \right) \quad (7)$$

$$\phi_i(x_i) = \theta_L + (\theta_H - \theta_L) \cdot \frac{1}{1 + \exp(-a_i x_i)} \quad (8)$$

ここで, $\mathcal{N}_i(0, 1)$ は標準正規分布に従う乱数とし, σ は定数とする. $y = \phi_i(x)$ は漸近線が $y = \theta_L$ と $y = \theta_H$ であるシグモイド関数である. a_i はノイズコントロールパラメータと呼ばれ, シグモイド関数の勾配 (俊度) を制御するパラメータである. w_{ij} と a_j は, 学習率 η_w, η_a を用いて

$$\Delta w_{ij} = \eta_w (\langle s_i s_j \rangle_0 - \langle s_i s_j \rangle_1) \quad (9)$$

$$\Delta a_j = \frac{\eta_a}{a_j^2} (\langle s_i^2 \rangle_0 - \langle s_i^2 \rangle_1) \quad (10)$$

と表すことができる.

RBM の学習で更新されるパラメータは重み行列と可視層バイアスベクトル, 隠れ層バイアスベクトルであった. 一方, CRBM の学習では重み行列と, 各ノードの俊度を学習することになる. 俊度がニューロンの敏感さを表すと考えれば, バイアス項を学習しない代わりにニューロンの敏感さを各ニューロンで学習することでニューロンごとに特性を与えていることになる.

2.2 DBN による時系列予測

DBN を用いた時系列予測では, Hinton ら [9] により提案された貪欲学習により学習を行う. 貪欲学習はプレトレーニングとファインチューニングという 2 段階で行う.

プレトレーニングでは最初の RBM を学習し, 隠れ層のベクトル, すなわち特徴ベクトルを算出し, それを 2 番目の RBM の入力ベクトルにする. 2 番目の RBM を学習し, 特徴ベクトルを算出して 3 番目の RBM の入力ベクトルとする. これを繰り返すことにより RBM が積み重なった stacked-RBM ができる. 最後に, 最後の最も深い RBM の特徴ベクトルを入力ベクトルとしてサポートベクターマシンや FFNN などで教師あり学習を行い, プレトレーニングを終了する.

ファインチューニングでは, すべての RBM と教師ありのネットワーク部分をひとつの FFNN とみなして全体で BP を行い, パラメータの微調整を行う. 層が多い場合の単純な誤差逆伝播学習 (バックプロパゲーション, BP) による学習の問題点を前述したが, プレトレーニングにより学習された各 RBM の重みやバイアスはファインチューニング時の準最適初期値となり, 学習が正しく行われるとされている.

2.3 Chao らの研究

Chao らは経済時系列からスライディングウィンドウ形式でデータを作成し, そのデータから翌日のものを

DBNに学習させている。その後、学習済みのDBNに予測用入力データを入力し、その出力値を予測値としている。DBNは多くの隠れ層（中間層）を持つFFNNである。FFNNは隠れ層の数が増えるとBPにおいて誤差のパラメータに対する勾配が小さくなるため、入力層に近い層のパラメータの更新が十分に行われないという問題点が生じる。また、過学習の問題もある。Chaoらは、Hintonら[9]によって提案されたDBNのパラメータの初期値を複数のRBMにより準最適にすることで深層なネットワークの学習を可能にするという方法を採用し、この問題を解決している。ただし、RBMの入出力は一般的には0か1のバイナリであり、経済時系列などの連続値に対応させるための方法が必要になる。Chaoらは、RBMの代わりに、Hsinら[4]によって提案されたCRBMを用いている。

彼らは、数日間の為替レートの変動をパターン認識し、翌日の為替レートを予測している。従来のフィードフォワードニューラルネットワーク（FFNN）に比べ、平均二乗誤差のルート（RMSE）、方向一致率（DA）などの指標について予測精度が向上することを示した。

3 提案手法

本研究では、現在世界的に注目されている日本市場の平均的な変動を表すと考えられる日経平均株価を対象とする。はじめに、データの作成法を示し、次に、Chaoらの研究の問題点を改善するためのネットワークトポロジーを提案する。また、自己組織化マップ（SOM）を用いた学習の効率化手法を提案する。

3.1 データの作成

Chaoらと同様に経済時系列からスライディングウィンドウ形式でデータを作成する。

時系列データを $p(t)$ ($1 \leq t \leq T$) とする。この時系列データからスライディングウィンドウ形式により入力データと正解データを作成する。

k 番目の入力データベクトル \mathbf{p}^k を $t = k$ から始まる n 日間の次式で示す経済時系列データとする。

$$\mathbf{p}^k = [p(k), p(k+1), \dots, p(k+n-1)] \quad (11)$$

ここで、 $1 \leq k \leq T - n$ とし、訓練データ数を $N_{all} = T - n$ とする。また、正解データを $p(k+n)$ とする。

訓練データを $1 \leq k \leq N_{tr}$ を満たすデータとし、予測データを $N_{tr} + 1 \leq k \leq N_{all}$ を満たすデータとする。よって $N_{pr} = N_{all} - N_{tr}$ とすると、訓練データ数は N_{tr} 、予測データ数は N_{pr} となる。

3.2 ネットワークトポロジー

本研究では、連続値に対応させるために、Chaoらと同様にCRBMを用いる。しかし、Chaoらの手法では、各入力データの変化が訓練データの最大値と最小値に対して小さくほぼ横ばいの入力データとなり、十分に変動パターンを抽出できない可能性がある。図1において n を6とした場合、すなわち6日間のウィンドウで得たデータの訓練データの1つを赤線で示した。青線は訓練データ全体の最大値と最小値である。このようにChaoらの手法では訓練データの各々の入力データは横ばいになる傾向が非常に強い。そこで、 k 番目の

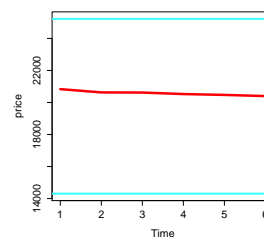


図1: Chaoらの手法での入力ベクトルの例

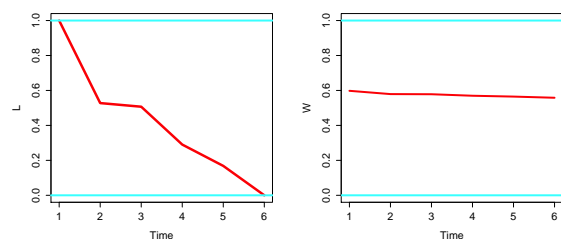


図2: 本研究の手法での入力ベクトルの例

入力データ \mathbf{p}^k を各データごとに正規化した \mathbf{L}^k と、訓練期間全体で正規化した \mathbf{W}^k の両方を用いたDBNにより予測を行うことにする。図2において、ある k での \mathbf{L}^k を左図に、 \mathbf{W}^k を右図に示す。 \mathbf{L}^k を併用することでより \mathbf{p}^k のパターンが明確となり、より良いパターンの抽出が可能となると考えられる。

提案手法のネットワークトポロジーとChaoらのネットワークトポロジーを図3に示す。図3の左図はChaoらの提案するネットワークトポロジーであり、右図は本研究で提案するネットワークトポロジーである。各矩形はNNの層を表す。赤色の矢印の部分だけが教師あり学習である。提案手法では、Chaoらと同様にCRBMを積み上げたものを用いて特徴抽出を行うが、これに入力するのは \mathbf{L}^k である点が異なる。さらに、最後のCRBMの特徴ベクトルと \mathbf{W}^k を結合したベクトルを

教師あり部分の入力ベクトルとする。これは、 \mathbf{L}^k だけでは回帰学習を行うことができないためである。

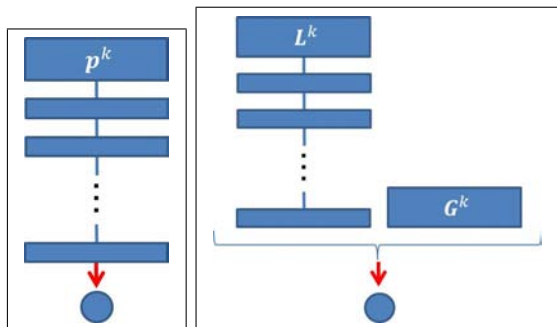


図 3: ネットワークトポロジー

3.3 訓練期間の更新

Chao らは翌日の値を予測する際、過去の離れた訓練期間により学習された DBN を用いている。しかし、翌日の値を予測する場面では、直近までのデータを用いて予測しようとする方が自然であると考えられる。よって、本研究では予測するたびに訓練データと予測データを 1 日分更新していくことを考える。すなわち、 N_{tr} 個の訓練データ \mathbf{p}^k ($1 \leq k \leq N_{tr}$) と各々の正解 $p(k+n)$ により DBN を学習し、学習済みの DBN に $\mathbf{p}^{N_{tr}+1}$ を入力して $p(N_{tr}+n+1)$ を予測する。

次に $p(N_{tr}+n+2)$ を予測するため、 \mathbf{p}^k ($2 \leq k \leq N_{tr}+1$) と各々の正解 $p(k+n)$ により学習し、 $\mathbf{p}^{N_{tr}+2}$ を入力して $p(N_{tr}+n+2)$ を予測する。これを N_{pr} 回繰り返すことで、 N_{pr} 個の予測値を得ることができる。

3.4 SOM による学習の効率化

前節で述べた訓練期間の更新により予測毎に訓練データが新しくなり、再度ネットワークを学習しなおす必要が生じ、非常に多くの計算時間を要することになる。この問題に対し、自己組織化マップ (SOM) により訓練データの一部のデータのみを用いた学習により、学習の効率化を試みる。

自己組織化マップによって、教師なし学習により入力データをクラスタリング (セグメント化) する。SOM は NN のひとつであり、教師なし競合強化学習と近傍学習によって、特徴が類似する入力データを 2 次元平面、トーラス面、球面にマッピングしていくものである。類似データは面上の距離の近い位置にマッピングされるという性質を持つ。入力データの各データの次元を d とすると、出力層のすべてのノードにも d 次元のベクトルが格納されておりこれをコードブックベクトルと呼ぶ。各クラス (セグメント) にはそのクラス

の特徴ベクトルを表すコードブックベクトルがあり、近傍のセグメントのコードブックベクトルが類似するように学習が行われる。

τ 回目の予測において、訓練データ: \mathbf{p}^k ($\tau \leq k \leq N_{tr} + \tau - 1$) と、予測のために入力するデータ: $\mathbf{p}^{N_{tr} + \tau}$ を用いて $p(N_{tr} + n + \tau)$ を予測する場合で説明する。

はじめに、 \mathbf{L}^k の集合 \mathbf{L} を SOM によってクラスタリングする。

\mathbf{L} の各々のデータ \mathbf{L}^k は n 日間の株価変化を n 日間の最大値と最小値を用いて正規化されたものであるため、 n 次元のベクトルである。本稿では \mathbf{L} を 2 次元の $P \times Q$ の格子状の平面に写像する。結果として、 $C (= P \times Q)$ 個の領域に分けられる。また、 i ($i = 1, 2, \dots, P$) と j ($j = 1, 2, \dots, Q$) は格子状の平面での位置を表し、 (i, j) をユニットと呼ぶ。ユニット (i, j) のコードブックベクトルを \mathbf{m}_{ij} とする。また、SOM の更新回数を T_{SOM} とし、 t_{SOM} を SOM の更新における時刻とする。

時刻 t_{SOM} におけるユニット (i, j) を中心にした近傍領域 $N_c(t_{SOM})$ に含まれるコードブックベクトル全てが入力データベクトルに近づくように学習される。

以下に具体的なアルゴリズムで示す。

(1) 初期化

\mathbf{m}_{ij} をランダムに発生させる。

(2) 繰り返し処理

以下を $t_{SOM} = 1, 2, \dots, T_{SOM}; k = 1, 2, \dots, N_{tr}$ について繰り返す。

(2-1) \mathbf{L}^k に最も類似したコードブックベクトル \mathbf{m}_{ij} を探す。このコードブックベクトルに対応するユニットを勝利ユニット (i^*, j^*) とし、これを以下のようにして求める。

$$(i^*, j^*) = \arg \min_{i,j} \|\mathbf{m}_{ij} - \mathbf{L}^k\| \quad (12)$$

ここでノルムはユークリッドノルムである。

(2-2) 勝利ユニットと近傍領域内のユニットの更新

$(i, j) \in (i^*, j^*) + N_c(t_{SOM})$ のとき、すなわち、勝利ユニットと近傍領域内のユニットであるとき、

$$\mathbf{m}_{ij} \leftarrow \mathbf{m}_{ij} + \alpha(t)[\mathbf{L}^k - \mathbf{m}_{ij}]$$

と更新する。それ以外の場合は何もしない。ここで $\alpha(t)$ は時間とともに減少する関数である。

このようにして、 \mathbf{L} が C 個のセグメントに分けられる。

次に、訓練データを各セグメントに分割する。ここで、各セグメントのデータを \mathbf{S}_c ($1 \leq c \leq C$) とする。

次に、 $\mathbf{p}^{N_{tr} + \tau}$ とコードブックベクトルとの類似度を算出し、 $\mathbf{p}^{N_{tr} + \tau}$ がどのセグメントに属するかを求め、ここで、このセグメントを c_{pr} とする。

学習には、 $S_{c_{pr}}$ と各々の正解を用いる。このようにすることで、 $p^{N_{tr}+r}$ と類似した訓練データの部分集合のみで学習することができる。

4 数値実験

Chao らの提案したネットワークトポロジーを G_α 、本研究で提案するトポロジーを G_β とする。さらに、 G_β で SOM を用いるものを G_β^{SOM} とする。使用する時系列データは 1991 年 7 月 11 日から 1997 年 9 月 1 日の日経平均株価の終値とする (図 4)。訓練期間と予測期間に分割して性能評価を行う。

以下で示す実験では、 $T = 1520$ 、 $n = 6$ とする。この時系列から 3.1 節で述べたようにして訓練データと予測データを各々 $N_{tr} = 1500$ 、 $N_{pr} = 14$ だけ作成する。 n については Chao の研究の結果を参考にしている。また、本稿では SOM の格子を 2×2 とする。

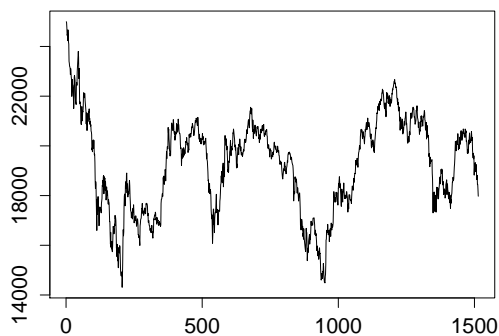


図 4: 使用した日経平均株価データ

本稿では、予測精度の指標として、 DA 、 $CORR$ 、 $RMSE$ 、 MAE 、 $MAPE$ を用いる。

以下、 $y(t)$ ($1 \leq t \leq N_{pr}$) を各予測日における正解値、 $y_p(t)$ ($1 \leq t \leq N_{pr}$) を各予測日における予測値とし、 \mathbf{Y} を正解値を順に並べたベクトル、 \mathbf{Y}_p を予測値を順に並べたベクトルであるとする。

DA は騰落の正確さを示す指標であり、値が高いほど騰落が予測できていることになる。以下のように定義される。

$$DA = \frac{100}{N_{pr} - 1} \sum_{k=1}^{N_{pr}-1} I(a_t) \quad (13)$$

$$I(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$

$$a_t = (y(t+1) - y(t))(y_p(t+1) - y(t)) \quad (14)$$

$RMSE$ 、 MAE 、 $MAPE$ は誤差の指標であり、それぞれ以下のように定義される。値が小さいほど予測期間

において平均的に正解に近い値を予測できていることになる。

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N_{pr}} (y_p(t) - y(t))^2}{N_{pr}}} \quad (15)$$

$$MAE = \frac{1}{N_{pr}} \sum_{k=1}^{N_{pr}} |y_p(t) - y(t)| \quad (16)$$

$$MAPE = \frac{100}{N_{pr}} \sum_{k=1}^{N_{pr}} \frac{|y_p(t) - y(t)|}{y(t)} \quad (17)$$

$CORR$ はピアソンの相関係数であり、予測値と正解の相関を調べるために使用する。

$$CORR = \frac{E(\mathbf{Y}\mathbf{Y}_p) - E(\mathbf{Y})E(\mathbf{Y}_p)}{\sqrt{E(\mathbf{Y}^2) - E^2(\mathbf{Y})}\sqrt{E(\mathbf{Y}_p^2) - E^2(\mathbf{Y}_p)}} \quad (18)$$

本稿では、以下に示す 2 つのシミュレーションを行う。

シミュレーション 1 Chao らの研究で行われていたように固定された訓練期間で学習を行い、各期間で G_α 、 G_β 、 $FFNN$ の予測性能を比較する。

シミュレーション 2 予測毎に訓練期間を最新のものに更新していく学習を行い、各期間で G_α 、 G_β 、 G_β^{SOM} 、 $FFNN$ の予測性能を比較する。

シミュレーション 2 では、 G_β^{SOM} において、先に現在の予測データがどのセグメントに属するかを判定し、そのセグメントの訓練データのみを用いて学習することで学習の効率化を行えることを示す。また、精度についても検証を行う。

4.1 シミュレーション 1

各トポロジーでのシミュレーションを 3 回ずつ行い、各指標の平均値を算出した。結果を表 1 に示す。計算時間については、シミュレーション開始から結果を出力するまでの計算時間 (秒) とした。 DA では、 G_α が他手法よりも高い数値となった。誤差の指標では、 G_α で高い数値が記録されている一方、 G_β で最も低く良い結果が得られている。

また、各シミュレーションで予測した数値の平均と正解の値をプロットしたものを図 5 に示す。図 5 より、Chao らのネットワークトポロジーではほぼ一定の数値を出力しており、誤差も大きいことがわかる。これは、特徴ベクトルが上手く求められていないことが原因と考えられる。

表 1: シミュレーション 1 の評価指標.

	T_α	T_β	FFNN
DA	59.0	53.8	53.8
CORR	0.846	0.834	0.833
RMSE	613.5	223.2	256.5
MAE	476.1	182.4	216.8
MAPE	2.58	0.97	1.16
計算時間	1839	1921	44

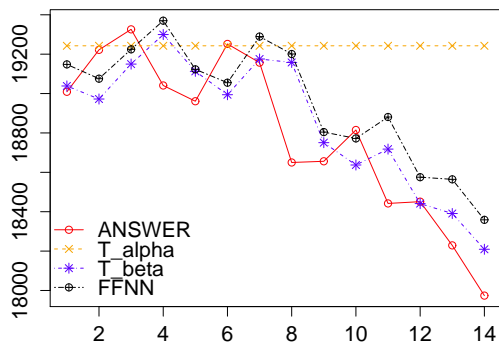


図 5: シミュレーション 1 の結果

4.2 シミュレーション 2

シミュレーション 2 の結果から得られた評価指標を表 2 に示す。また、各シミュレーションで予測した数値の平均と正解の値をプロットしたものを図 6 に示す。

表 2: シミュレーション 2 の評価指標.

	G_α	G_β	G_β^{SOM}	FFNN
DA	52.4	52.4	57.1	53.8
CORR	0.786	0.820	0.929	0.818
RMSE	728.1	230.0	231.0	253.0
MAE	507.3	189.9	184.6	205.4
MAPE	2.75	1.01	0.98	1.10
計算時間	12309	12868	6068	637

指標について、シミュレーション 2 では DA について G_β^{SOM} が最も良い結果となった。また、 G_β^{SOM} は、RMSE については最も良い結果であった G_β とほとんど違いが見られず、MAE と MAPE については最も良い結果となった。その上で、計算時間については G_β^{SOM} の計算時間が他の DBN の計算時間のほぼ半分の時間でシミュレーションすることができている。シミュレーション 2 でも G_α では適切な予測を行うことができなかった。

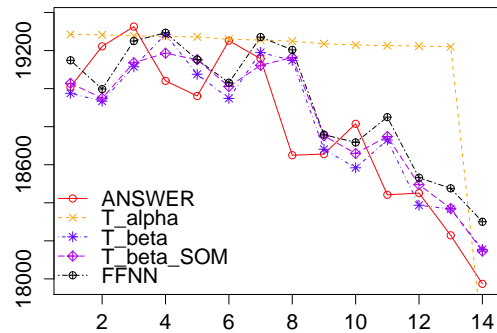


図 6: シミュレーション 2 の結果

5 むすび

本研究では、Deep Belief Network に基づく Chao らの経済時系列予測手法を拡張することにより、日経平均株価の短期予測を行った。

はじめに、Chao らの手法では、各々の入力データが横ばいになりパターンの抽出が上手くいかない可能性があることを指摘した。また、これに対し、各データごとに正規化した L と訓練期間全体で正規化した W の両方を入力することでその問題が解消できることを示した。

次に、学習期間を更新し続ける学習では、SOM を用いることで訓練データの一部分のみで学習できることを提案し、数値実験によって予測性能を低下させずに計算時間が削減できることを示した。

一方、訓練期間をずらしていく予測では、訓練期間を固定した予測よりも全体として評価指標の値は悪くなることがわかった。これは、固定していた時に予測に貢献していたデータのパターンが、更新によって崩れてしまったためと考えられる。よって、訓練期間をずらすのではなく、新しいデータを追加し続けていくような学習形式での予測も試み、予測精度を比較した方が良いと考える。この際にも提案手法である SOM での効率化を行うことができると考えられる。

また、本稿では、特徴抽出用の入力 L としてその時の株価を正規化したもののみを用いているが、他の時系列データやテクニカル指標、およびテキストデータなどを入力することも可能である。これは今後の課題である。

参考文献

- [1] 岡谷貴之, 齋藤真樹: ディープラーニング, 情報処理学会研究報告, Vol.2013-CVIM-185, No.19, (2013)

- [2] 中田貴之, 古関隆章: Support Vector Machine による値動きの方向性に注目した短期間の経済時系列予測, (D-12. パターン認識・メディア理解, 一般セッション), 電子情報通信学会総合大会講演論文集 2008, p. 264(2008)
- [3] Le, Q.V., Ranzato, M.A., Monga, R., Devin, M. Chen, K., Corrado, G. S., Dean, J. and Ng, A. Y.: Building High-level Features Using Large Scale Un-supervised Learning, In proceedings of *ICML 2012*, (2012)
- [4] Hsin, C. and Murray, A.F. : Continuous restricted Boltzmann machine with an implementable training algorithm, *Vision, Image and Signal Processing*. Vol. 150, No. 3. ,pp. 153–158 , (2003)
- [5] Chao, J., Shen, F. and Zhao, J.: Forecasting exchange rate with deep belief networks, *2011 International Joint Conference on Neural Networks (IJCNN)*, pp. 1259–1266(2011)
- [6] Merck Competition Results - Deep NN and GPUs come out to play;
<http://blog.kaggle.com/2012/10/31/merck-competition-results-deep-nn-and-gpus-come-out-to-play/>, 2013/12/15 アクセス.
- [7] Dahl, G. E. , Yu, D. , Deng, L. , and Acero, A. : Context-dependent pre-trained deep neural networks for large vocabulary speech recognition, *IEEE Trans. Audio. Speech. Lang. Process.*, Vol. 20, No. 1, pp.30–42(2012)
- [8] Bengio, Y.: Learning deep architectures for AI. *Foundations and trends in Machine Learning*, Vol. 2, No. 1, pp. 1-127(2009)
- [9] Hinton, G. , Osindero, S. and Teh, Y. W. : A fast learning algorithm for deep belief nets. *Neural Computation*, Vol. 18, No. 7 , pp. 1527–1554(2006).
- [10] Tieleman, T.: Training restricted Boltzmann machines using approximations to the likelihood gradient. In *International Conference on Machine Learning 2008*, (2008)
- [11] Box, G., Jenkins, G.M. and Reinsel, G.: Time Series Analysis: Forecasting and Control, Prentice-Hall, (1994)