

# ディープラーニング株価予測の アンサンブル学習による精度向上についての分析

池田 欽一

## 概要

本研究では、筆者が以前示したディープラーニングによる株式予測において、多数の予測器による多数決であるアンサンブル学習を導入することにより予測精度の向上が得られるかシミュレーションにより検証する。株価予測は、株価の1分足のローソク足画像、および出来高のグラフを入力とし、画像分析への応用が進められている畳み込みニューラルネットワークを用い、出力は株価が「上がる」、「下がる」という2値とする。アンサンブル学習では、多数の予測器の出力に共通性があると効果が十分に表れないが、すべての予測器を畳み込みニューラルネットワークにより作成するため、多様性を確保するためにバギングによるサブデータセット作成や、入力画像の株価、出来高の期間を変更するなどの手法を用いている。また、単純な多数決によるアンサンブルと集計方法を学習するスタッキングによる方法の予測精度について比較検討も行っている。

## 1 はじめに

本研究で用いるディープラーニングは、視覚の映像処理をモデル化した畳み込みニューラルネットワーク(CNN, Convolutional Neural Network)で、画像の部分的な特徴をフィルタの学習により抽出し、さらにその特徴量からフィードフォワードニューラルネットワークにより予測に有効な組み合わせを学習していく手法である。入力情報の空間的な配置を利用しており、特に画像、動画認識に優れた性能を示している、進近年のAI技術の大幅な進展に寄与している。

以前筆者が示した、畳み込みニューラルネットワークの株価予測への応用[2]では、株価の四本値とよばれる1分、1日、1週間などの一定時間の最初の取引価格「始値」、最後の価格「終値」、最大値「高値」、最小値「安値」を用いて作成されるローソク足チャート画像を入力とし、1から5分後の株価の変動「上がる」「下がる」の予測の可能をシミュレーションを元に示した。さらに、入力情報にローソク足チャートだけではなく、一定期間の間に取引された株式の数量の出来高のグラフによる予測の可能性、また、ローソク足チャートや出来高グラフを組み合わせることで入力に用いることにより予測精度が向上することを示した[1]。本論文では、まずローソク足チャートや出来高グラフに用いる時間間隔の差による予測精度についてまとめ、さらに、アンサンブル学習と呼ばれる多数の予測モデルの多数決により最終的な予測値を決定する手法の導入による予測精度の変化をシミュレーションにより検証している。

畳み込みニューラルネットワークなどディープラーニングは、大量の入力データ、いわゆるビッグデータを用いることにより、入力情報の微細な差が結果に与える影響を学習したり、様々な特徴量に柔軟に対応できる入出力モデルを学習できるが、学習するデータ量が足りないと、その柔軟性により学習データのみで特化し、多くのケースで学習に用いていない評価用データについての予測精度が悪化する過学習と言われる現象が発生することとなる。経済データは、月次、四半期や年単位でしか得られないデータも多く、サンプル数不足によりディープラーニングの適用が困難な場合が多いが、

株価データは、各取引ごとのティックデータも入手でき、取引の多い企業の株価（銘柄）によっては1日に数万件のサンプルを得ることも可能である。本論文では、1分ごとの四本値を用いるが、1日の営業日に300サンプル、1年間で約6万サンプルが入手可能である。ディープラーニングの学習にこの大量のデータを用いることにより、株価や出来高の変化パターンとその後の株価変化の汎用的な関係性を学習させることが可能となる。

以下、2.で畳み込みニューラルネットワークの概要について述べ、3.ではアンサンブル学習の概要を示す。4.では畳み込みニューラルネットワークを株価予測へ応用するための実装方法や用いる株価データ、パラメータ設定について述べる。4.ではローソク足チャートや出来高の画像に用いる時間数を変更した場合予測精度、および、アンサンブル学習を導入した場合の予測精度の改善効果についてのシミュレーション結果とその考察を示し、6.でまとめとする。

## 2 ディープラーニングとは

ディープラーニング（深層学習）は、様々なモデル、応用方法が研究されているが、脳神経細胞の処理を数理モデル化したユニットを用いネットワーク化したフィードフォワードニューラルネットワークを多層化したシステムや、視覚野での情報の2次元処理を模倣した畳み込みニューラルネットワークが代表的なモデルである。フィードフォワードネットワークは、データの入力側の1つ隣の層の各ユニット（神経細胞を数理モデル化したもの）からの出力を加重集計し、バイアス値を足し、活性化関数といわれる非線形関数により変換して出力側に順次送られていくシステムであるが、この層を複数化したものがディープラーニングと呼ばれる。ただ単に層を増やすだけでは入力側の学習が進まない問題が存在していたが、各種事前学習手法や活性化関数の工夫により、今日の機械翻訳やリコメンド等の応用の発展に寄与している。畳み込みニューラルネットワークは平面的な情報の広がり存在する特徴量を抽出することができ、画像やその応用分野において、近年は顕著な発展が見られ、従来の手法よりも良好な結果が得られている [1, 2, 5]。

本研究では畳み込みニューラルネットワークを用い、入力情報には株価や株式の取引数量である出来高の画像を用いる。分類、予測の出力は株式の将来価格が上がるか、下がるか（変わらない含む）の2値のカテゴリデータを用いる。入力された画像データは複数のフィルタと畳み込み（convolution）処理が行われ、各フィルタとの類似度として数値化される。畳み込みに用いられるフィルタは、出力の実際の値とネットワークの出力値の誤差が減少するように学習が進められるが、十分なデータ量で学習されたフィルタは、出力が実際の値に近くなるように有効な特徴を画像から抽出するパターンとなっていく。フィルタは入力画像よりもサイズが小さいものが用いられるので部分的な特徴を得ることができる。畳み込み計算では、フィルタ位置をずらしながら処理され、画像全体にフィルタに対応したパターンがどのように分布しているかという情報を得ることができる。この工程を実施する部分を畳み込み層と呼ぶ。さらに、畳み込みの後に、周辺の数値の平均や最大値を求めるPooling層を準備することにより、類似パターンの分布位置にあいまいさを持たせることができるようになっていく。一般的には、畳み込み層とPooling層を複数回実施し、入力画像に分布するフィルタに類似パターン、さらにその類似パターンの分布の状況を数値化することができるこれらフィルタに類似したパターン分布の2次元情報は、フィードフォワードネットワークへの入力に用いられ、出力層での実際値と出力値の誤差を最小化するように情報利用の組合せが最適化される。[1, 2, 5]。

これら、畳み込み層、Pooling層、フィードフォワードネットワークにより、画像を入力として、何らかの数値やカテゴリを予測する畳み込みニューラルネットワーク全体が構成されることとなる。以下では、文献 [1, 2] を元に、畳み込みニューラルネットワークについて説明する（模式図による説明など詳細は、文献 [2, 5] を参照されたい）。

## 2.1 畳み込み層

本研究では、すでに述べたように、株価の4本値を画像化したローソク足チャートや株式の取引数量を表す出来高グラフをシステムへの入力とする。どちらもグレースケールの画像形式で入力し、以下のように各入力画像とフィルタとの畳み込みを計算する。

ローソク足チャートや出来高グラフの画像の画素サイズを  $W \times W$  とし、画像データを  $x(i, j), i = 0, 1, \dots, W-1, j = 0, 1, \dots, W-1$  で表すこととする。この画像の部分的な特徴を抽出するために、入力画像よりも画素数の小さい画像をフィルタとして用いるが、このフィルタのサイズを  $H \times H$  とし、 $h(p, q), p = 0, 1, \dots, H-1, q = 0, 1, \dots, H-1$  で表す。このフィルタ  $h(p, q)$  を用いた画像  $x(i, j)$  の畳み込み  $u(i, j)$  は以下の式のように表される [1, 2, 4, 5]。

$$u(i, j) = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} h(p, q)x(i+p, j+q). \quad (1)$$

この畳み込みは前述のように、画像データ  $x(i, j)$  をフィルタと同サイズに切り取った部分とフィルタ  $h(p, q)$  の類似度を数値的に表すものとなる。類似度が高い場合には数値が大きくなり、類似度が低い場合には0に近くなり、写真のネガのように色を反転させた画像になる場合には負の絶対値の大きい数値となる。

フィルタをストライド  $s$  と呼ばれる数値ずつ左右方向、上下方向それぞれずらしながら、この畳み込みを画像全体に適用することにより、フィルタに類似のパターンが画像全体にどのように分布しているかの2次元配列を得ることができる。この2次元配列もフィルタに類似のパターンがどのように配置しているかの画像として考えることができる。通常、フィルタは複数枚用意され、すべてのフィルタに同様の処理が実施され、出力された2次元配列へ後述する活性化関数を適用する一連の操作を実施する部分を、畳み込み層と呼ぶ。さらに準備した別フィルタとの畳み込みを計算して第1段階の特徴量がどのように分布しているかという広域的な特徴パターンの分布を数値化したものを得ることができる。本研究ではストライド  $s = 1$  とし、フィルタを1ずつずらしながら画像全体に適用することとする。畳み込み計算の際に画像の端ではフィルタとの畳み込みを計算するための周辺の画素情報が不足し、周辺画素の平均値、あるいは画像を反転させた値などで補うパディングを実施しないと、元の画像サイズ  $x(i, j)$  に比べ、畳み込み  $u(i, j)$  はフィルタサイズが奇数の場合  $2\lfloor H/2 \rfloor$ 、偶数の場合  $2\lfloor H/2 \rfloor - 1$  だけ小さいサイズの2次元配列となる。[ ] は床関数である。本研究ではパディングは用いので、畳み込み計算をするたびに出力値はフィルタサイズに応じて小さくなることとなる。

フィルタはランダムな数値で初期化されるが、ネットワークの出力と望ましい出力との誤差関数の勾配を各パラメータで偏微分することにより更新数量を求め、フィルタの数値、およびネットワークの接続ウェイトを修正（学習）していくことにより、入力画像とそれに対応した出力を学習していくこととなる [4]。畳み込み層のフィルタは学習が進むにつれ、望ましい出力を得るために有効なある種の画像の特徴抽出するパターンを表すようになっていく。本研究ではグレースケール画像を用いるが、カラー画像の場合には、R,G,Bの3原色に分解し、3枚の画像を3チャンネルとして入力し、R,G,B画像それぞれにフィルタを適用し、結果の畳み込みの同じ位置の値の総和を取ることにより同様に処理することができる。

式 (1) による畳み込みは  $m = 1, 2, \dots, M$  の  $M$  枚のすべてのフィルタについて計算され、内部状態  $u(i, j, m)$  に以下のように活性化関数  $f(\cdot)$  を適用することにより、畳み込み層の出力  $z(i, j, m)$  が求められる。

$$z(i, j, m) = f(u(i, j, m)). \quad (2)$$

活性化関数には、以下のようなシグモイド関数、双曲線正接関数、Relu (Rectified Linear Unit, Rectifier) 関数 (正規化線形関数) などが用いられ [1, 2, 5]、本研究では、シミュレーションの結果、最も平均的予測精度が最も高くなった下記のシグモイド関数を用いることとする。

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (3)$$

## 2.2 Pooling 層

Pooling 層は、通常畳み込み層 (複数の畳み込み層を連続して用いることもある) の出力に対して用い、畳み込み層のフィルタにより表される特徴パターンの位置に柔軟性を持たせることができるのである。畳み込み層で得られたフィルタと類似のパターンの分布の2次元配列を画像として考え、この画像への Pooling の適用は、画像の正方形の部分画像領域  $S \times S$  を考え、その中の画素値の最大値、あるいは平均値などを求める演算である。正方領域の一边  $S$  を大きくするとより大域的に特徴量をとらえ、小さくすると局所的にとらえることとなる。この Pooling の出力も2次元配列となるので、この出力も画像と考えることもできる。Pooling 演算の領域は、通常重複のないように適用されるので、出力画像の一边は入力画像の一边の  $1/S$  となり、画素数  $1/S \times 1/S$  となり、その後の計算量を大幅に減らすことにも貢献している。入力が多チャンネルの場合にはチャンネルごとに適用されるので、チャンネル数を  $K$  とすると、 $(W, W, K)$  のサイズの入力からは、 $(W/S, W/S, K)$  の出力が得られる [1, 2]。Pooling 層の後に出力値を標準化する正規化層を用いることもある [8]。

## 2.3 全結合層

畳み込み層、Pooling 層を経て (複数回適用の場合もある)、システムへの入力画像は、出力の最適化に有効な特徴量抽出分布を表す2次元配列が最後の畳み込み層のフィルタ枚数分出力される。次に、この2次元配列の各値を1行目の後に2行目、そのあとに3行目...のように1次元配列に引き伸ばし、全フィルタ分1列に繋ぎ、全結合層とも呼ばれるフィードフォワードネットワークへの入力として用いる。最終的にこのフィードフォワードネットワークの出力層において、入力データの所属カテゴリ (本研究では、株価が「上がる」「下がる」) の分類、予測出力が得られる。全結合層とは、以下のように神経細胞を模式化したニューロンを層状に並べたネットワークである。これは従来のフィードフォワードニューラルネットワークと同様に計算される [1, 2, 5]。

ニューロンとは神経細胞を次のようにモデル化したものである。

$$u_j = \sum_{i=1}^I w_{ji} x_i + b_j, \quad (4)$$

$$z_j = f(u_j). \quad (5)$$

ここで、 $u_j$  は第  $j$  ニューロンの内部状態を表し、次のフィードフォワードニューラルネットワークで説明するように、層としてグループ化され、前の層のニューロンからの出力にウェイトをかけて集計したもので計算される。 $x_i (i = 1, 2, \dots, I)$  は1つ前の層のニューロンの出力値、 $w_{ji}$  は1つ前の層の第  $i$  ニューロンから現在の計算対象の層の第  $j$  ニューロンへの接続の強さを表すウェイト、 $b_j$  は一定の値を表すバイアスである。内部状態  $u_j$  に畳み込み層で示したのと同様な活性化関数  $f()$  を適用し、そのニューロンの出力  $z_j$  を求める。



上記のようなニューロンを一定数グループ化し層としてまとめ、入力側から出力へ、入力層、任意の数の中間層、出力層と並べ、各層は1つ前の層の全ニューロンからのみ入力を受け取る。このようなネットワークをフィードフォワードニューラルネットワークと呼ぶ。第  $l$  ( $l = 1, 2, \dots, L$ ) 層の内部状態  $\mathbf{u}^l = (u_1^l, u_2^l, \dots, u_j^l, \dots, u_{j_l}^l)'$  は次のように表される。

$$\mathbf{u}^l = W^l \mathbf{z}^{l-1} + \mathbf{b}^l. \quad (6)$$

ここで、 $\mathbf{z}^{l-1} = (z_1^{l-1}, z_2^{l-1}, \dots, z_i^{l-1}, \dots, z_{j_l}^{l-1})'$  は1つ入力層側の第  $l-1$  層のニューロンの出力からなる第  $l$  層への入力ベクトルで、 $\mathbf{b}^l$  は第  $l$  層の各ニューロンのバイアス値を列ベクトルで表したものである。 $W$  は次のような第  $l-1$  層と  $l$  層のニューロン間のウェイト行列である。

$$W^l = \begin{pmatrix} w_{11}^l & w_{12}^l & \dots & w_{1j_l}^l \\ w_{21}^l & w_{22}^l & \dots & w_{2j_l}^l \\ \vdots & \vdots & \ddots & \vdots \\ w_{j_l 1}^l & w_{j_l 2}^l & \dots & w_{j_l j_l}^l \end{pmatrix}. \quad (7)$$

第  $l$  層の出力  $\mathbf{z}^l = (z_1^l, z_2^l, \dots, z_j^l, \dots, z_{j_l}^l)'$  は次のように表される。

$$\mathbf{z}^l = \mathbf{f}(\mathbf{u}^l). \quad (8)$$

ここで、 $\mathbf{f}(\mathbf{u})$  は活性化関数ベクトルで次のようなものである。

$$\mathbf{f}(\mathbf{u}^l) = (f(u_1^l), f(u_2^l), \dots, f(u_{j_l}^l))'. \quad (9)$$

ただし、予測対象がカテゴリである場合、出力層のユニット数はカテゴリと同数とし、活性化関数ではなく、以下の softmax 関数により、各ユニットの出力値が  $0 \sim 1$  の値、総計が  $1$  となるように変形される。これは各カテゴリに所属と判定する確率を表すと考えることができる。

$$z_j^L = \frac{\exp(u_j^L)}{\sum_{k=1}^{J^L} \exp(u_k^L)}. \quad (10)$$

本論文で扱うような出力対象がカテゴリデータ、つまりクラス分類の場合、出力層のユニット数をグループ（クラス）の数と同一とし、各ユニットを各グループに割り当て、学習後に入力データを与え、最も出力の大きいユニットに割り当てられたグループに所属すると予測することが一般的である [1, 2, 5]。具体的に言うと、本論文では予測対象は数分後の株価が「上がる」「下がる（変わらない）含む」の2値のカテゴリデータであるので、最終的な出力層は2つのニューロンユニットを準備し、どちらの出力が大きいかでシステムの分類、予測出力を決定することとなる。

## 2.4 畳み込みニューラルネットワーク

上記のような畳み込み層、Pooling 層を通常は数度経て、全結合層を数層経過したのち出力層にてシステム全体の出力を得るシステム全体は畳み込みニューラルネットワークと呼ばれる。出力層のユニットの出力は式 (10) の softmax 関数により求められ、これらの出力値と実際のデータ（教師データ）との間のクロスエントロピー誤差を計算し、この誤差を減ずるように誤差を入力側へ伝搬しながら、全結合層のウェイト、バイアス、畳み込み層のフィルタ値、バイアスを更新して学習を進める

[4]. クロスエントロピーは、学習サンプル  $n, n = 1, 2, \dots, N$  の実際の所属カテゴリが  $C_n$  である場合、以下の式により求められる [5]。

$$E(\theta) = - \sum_{n=1}^N \log z_{C_n}. \quad (11)$$

ここで、 $z_{C_n}$  は学習サンプルの実際の所属カテゴリに対応する出力層のユニットの softmax 関数適用後の出力値であり、 $\theta$  はネットワークのパラメータ、つまり、全結合層のウェイト、バイアス、畳み込み層のフィルタ値、バイアスである。

本論文では、畳み込み層のフィルタ、バイアス、および、全結合層のウェイト、バイアスの学習には、Adam(Adaptive moment estimation)を用いる。Adam とは、少量のメモリで一次勾配のみにより適用可能な効率的な確率的最適化手法で、パラメータごとに勾配の平均、分散の推定値を利用する手法で、多くのケースにおいて良好な学習が得られることが示されている [11]。各パラメータの勾配の平均、分散を推定したものを利用し、勾配が疎になる場合や非正常に強い性質を持っている [12]。

## 2.5 ドロップアウト

ドロップアウト (dropout) とは、通常畳み込み層の出力の 2 次元配列や全結合層のニューロンの出力値に利用され、出力値の一部をある割合で強制的に 0 にすることにより実装されることが多い。このドロップアウトの効果としては、フィルタとの畳み込みの出力にノイズを付加する効果や全結合層の使用ユニット学習の際に使用ユニットを制限することと同様な効果があり、これら効果によりフィルタやウェイトの学習の際に学習データに特化したシステムとなり、学習に利用していない評価用データの予測の性能が低下する、いわゆる過学習を防ぐことができる。また、ドロップアウトをどの程度の出力に適用するかのドロップアウト率を変化させることにより、最終的に収束する畳み込みニューラルネットワークのパラメータに影響を与えることとなる。

## 3 アンサンブル学習

本論文では上記のような、ローソク足チャートや出来高画像を入力とし、画像の作成に用いた株価や出来高の最終時刻から数分後の株価の「上がり」「下がり」の予測を出力をするシステムを 1 つの予測器とし、この予測器を多数準備し、アンサンブル学習により最終的な予測を決定することとする。

アンサンブル学習とは、多様性を持たせた多数の予測器による多数決で予測精度を向上させる技術である [7]。参加者の所有する情報に偏りがあり個々の参加者が誤った判断をする可能性が高い状態での判断では、多数決が全体の良い意思決定とならない例も多々あるが、個々が正しい情報に従い、ある程度正しい判断をしていると一人の専門家よりも有効なことがある。本論文の予測対象である株式の価格は、企業の経営状態など、いわゆるファンダメンタルにより決まる部分と、ファンダメンタルに変化はなくても市場参加者の個々の様々な思惑によりその株式に対する需要と供給が変化し、その需給関係により決まる部分とに大まかに分けて考えることができる。本論文ではファンダメンタルに関する入力情報は用いていないが、ファンダメンタルな情報が価格に織り込まれると考えると価格変化のみで様々な情報を得ると考えることもできる。また、株式市場は個々の意志、判断で取引を行い需要と供給により価格が決まるので、通常物理現象よりも予測は困難であると思われるが、株式市場と同様に個々の予測器の多数決で最終的な予測を決定するアンサンブル学習は予測精度の

向上に貢献することが期待される。アンサンブル学習では、多数の種類の子測モデルを用いることもあるが、本論文では、畳み込みニューラルネットワークのみを予測器に用い、学習データの部分集合(サブデータセット)の違いや、ドロップアウト率の変更などにより多様性を確保することとする。

一概にアンサンブル学習と言っても、いくつかの方法があるが、最も簡単な方法は上記のように多様性を持った予測器の各予測の多数決を実施する方法である。株価の「上り」「下がり」の予測では、各予測器が「上がる」か「下がる」かを予測するので、どちらの投票が多いかにより最終的な判断を実施する方法である。この方法は同じ入力画像に対して各予測器が注目する特徴量に違いがあるなどにより、同様な予測でないことが精度向上のためには期待される。単純な集計では、2値(0か1)の予測の多数決には各予測器が0と1のどちらと予測するかの投票数の多い方に決定するhard voteと各予測器が2値のどちらかであることを0から1の数値で出力し、その数値の集計が予測器の数の半数を超えるかどうかで決定するsoft voteの方法があるが、本論文の単純多数決ではsoft voteを用いる。

以下ではアンサンブル学習における主要な手法についてまとめておく[6, 7]。

### バギング

バギングとは、bootstrap aggregatingの略称で、復元抽出を用いたサブデータセットを多数作成し、それらサブデータセットで学習させた個々の予測器の多数決で最終的な意思決定をする手法である。サブデータセットの作成も容易で並列計算も実施でき、学習用データの持つモデルへの不適合性(非線形関係であるのに線形モデルを当てはめるなど)は解決できないものの、学習データへの過剰な適応、いわゆる過学習は減らすことができる。株式の価格変動予測では、学習データに対しては学習率がほぼ100%となるが、学習に用いていない評価データに対しては60数パーセントの予測精度[1, 2]であり、バギングは有効であると考えられる。

### ブースティング

ブースティングとは、hypothesis boostingの略称で、新しい予測器に、前の予測器の間違いを正すように連続的に複数の予測器を作成する手法である。主流はAdaBoost(Adaptive Boosting)で、最初は各学習データの重要度のウェイトを均一として、出力が誤りであるデータのウェイトの合計が最小となるようにベースの予測器を作成し、その予測器が適切に予測できなかった学習データのウェイトを増加し、次の予測器の作成を繰り返していく。これにより前の予測器が苦手とするデータを次の予測器が集中的に学習することができるようになっていく。各予測器の誤り率から計算された信頼度を掛け各予測器の出力を集計し、最終的な予測を決定する方法である。

### スタッキング

スタッキングは、簡単に言うと多数決の集計方法を単純な合計ではなく、学習により最適化する手法であると言える。最終予測は学習データを分けることにより実施する方法が一般的である。2段階のシステムの場合、第1段階目に各予測器を準備し、それらの出力を入力として扱い、最終出力を求めるための最適な組み合わせを学習する最終段階の予測器(blender)を用いる。2段階のシステムの場合、学習データを半分に分け、それぞれで第1、第2段階を学習させることとなる。同様にして3段、4段のシステムの学習も同様に3つ、4つにデータを分割することにより実行可能である。

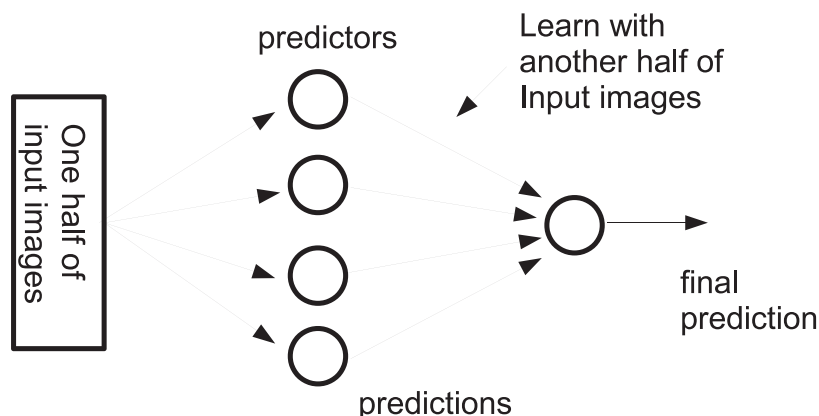


図 1: 2段のスタッキングの例

#### 4 アンサンブル学習を用いた畳み込みニューラルネットワークによる株価変動予測

筆者が以前示した株価予測への畳み込みニューラルネットワークの応用 [1, 2] で、日経平均株価の1分間隔の四本値（始値、安値、高値、終値）より作成した20分間のローソク足チャート、および出来高を入力とし、画像作成に用いた最後の1分から1～5分後に平均株価が「上がる」か、「下がる」（変わらないを含む）かを予測する方法を示した。結果として、ローソク足チャートや出来高のグラフのいずれも2分後から5分後については、60%前後の予測精度となり、ランダムは予測の50%を超える精度が得られることを示した。さらに、ローソク足チャートと出来高のグラフを組合せて入力を多重化することにより、それぞれ単独で用いる場合よりも予測精度が向上することも示した。いずれの場合も、畳み込みニューラルネットワークは単独で用いている。

本論文では、上記で概要を示したアンサンブル学習をシステムに応用し、シミュレーション分析により、単独の予測器による予測と予測精度の比較分析を実施する。1つの予測器のみでの予測精度が高い場合には単独の予測器で十分である場合もあるが、2値予測の場合で、個々の予測性能がランダムな予測の50%を超えがそれほど高くない場合で、かつ予測結果にバラつき、つまり多様性がある場合には、多数決が有効であることが示されている [6, 7]。通常アンサンブル手法は様々なモデルによる予測器を多数用いることにより多様性を担保しているが、本論文では、予測モデルは畳み込みニューラルネットワークのみを用いることとする。そこで、多様性を確保するためにはすでに述べたように、入力情報をローソク足チャート、出来高グラフそれぞれ作成に用いるデータ期間を10分、20分、30分の計6種類、さらにドロップアウトによるユニット使用割合を0.3, 0.7, 1.0の3設定とし、これらの組合せにバギングによる学習データのサブセットをそれぞれ3つずつ作成し、学習に用いていない評価用データの予測の正確性は維持しつつ、正解、不正解が評価データの中でばらつき、予測結果に多様性が出るような工夫を行うこととする。ただし、入力画像種とサブセットデータ、ドロップアウトの組合せのうち、安定的に50%を超えるものを選別して用いることとする。

図2には、モンテカルロシミュレーションにより求めたアンサンブルに用いる2値予測器の数と予測精度の関係を示している。各予測器の予測精度55%とし、ランダムに10000サンプル分の正解、不正解を決定することとする。各予測器の多数決（hard vote）により集団の予測を決定することとす

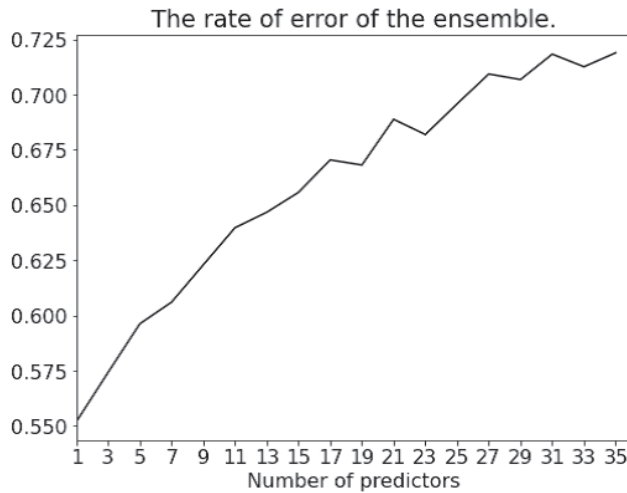


図 2: 用いる学習器の数と予測精度

る。アンサンブルに用いる予測器の数は集計が同数となることがないように1から35の奇数として計算している。この図を見ると、個々の予測器の予測精度は55%であるが、予測器の数を増やすにしたがって、予測精度は増加傾向にあることが読み取れ、35の予測器を用いた場合には予測精度が約72%に達している。ただし、このシミュレーションでは個々の予測器の正解、不正解はそれぞれランダムに決定し、各予測器の予測の間には相関はないことを前提とした結果となっている。本論文では、すでに述べたように学習用データのサブセット作成や入力情報、ドロップアウト率の違いを用いて予測結果に多様性を持たせるよう工夫するが、すべての予測器が同じ畳み込みニューラルネットワークであることと、過去の株価や出来高の変動から決まった分数後の株価の上がり、下がり予測するというように、予測の対象が同じであるため、予測に完全な独立性を持たせることは不可能である。これにより、各予測器の予測に共通性が現れ、図2の結果よりも予測精度は低い結果となる。すべての予測器が同一の予測をすると単独の予測器の予測精度55%に等しくなるので、アンサンブル学習による予測結果は、個々の予測器の予測の相関の強さにより、単独の予測器の予測精度～図2に示される予測精度の間の精度となると考えられる。

#### 4.1 入力に用いるデータと画像

ディープラーニングの学習では、学習データが十分準備できない場合には、その学習データにシステムのパラメータが追従しすぎ、学習に用いていない評価用データ（トレーニングデータ）の予測性能が低下する過学習が発生することが知られている。この過学習を防ぐためには、複雑すぎないモデルにする、ドロップアウトを導入する、パラメータに制限をかける目的で誤差関数に正則化項を追加する、あるいは、学習、評価データとは別に検証用のデータセットを準備し、この検証データはパラメータの最適化には利用せず、パラメータ最適化に用いないデータの予測精度のモニタリングに利用し、この検証データの予測精度が低下する前で学習を終了させる Early stopping などの方法が利用されている。この中で、過学習を防ぐ最も単純な方法は学習用データを大量に準備する方法であり、さらに学習データ間のわずかな相違をシステムに取り込むことができ、汎化性能の向上が期待される。株価や出来高データは、ティックデータ（取引発生ごとの価格と数量の記録）は銘柄によっては



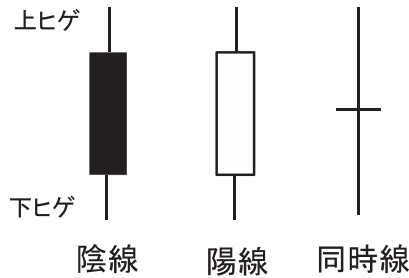


図 3: ローソク足 (文献 [2] より引用)

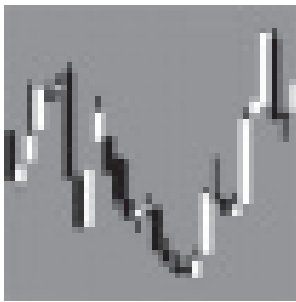


図 4: ローソク足チャートの例

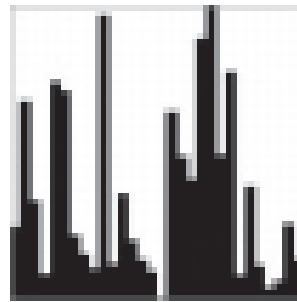


図 5: 出来高グラフの例

1日に数万件以上得ることもでき、日足（1日ごと）、週足（週ごと）、月足（月ごと）などの期間の株価を用いるよりも、ビッグデータを確保する目的では最も適しているということになる。しかし、ティックデータは取引が発生する間隔が決まっておらず、入力画像の横軸、つまり入力される画像の時間間隔が一定とはならないこととなる、よって本研究では1分ごとの株価や取引数量である1分足データを用いることとする。また、個別銘柄への適用も可能であるが、1分間に1度も取引がないとデータが観測されず欠損値となり、また個別企業の業績などにより過去の株価や出来高と関連のない変動が発生することがあるので、本研究では、安定的にデータが観測され、個別企業の影響がない日経平均株価を予測の対象としてシミュレーションすることとする [1, 2]。

図3にはローソク足を示している。一定の時間間隔、本論文では1分間の最初の取引価格（始値）、最後の価格（終値）、最も高い価格（高値）、最も低い価格（安値）の情報が1つの図で表され、黒や白の長方形や直線の組合せにより表されている。始値より終値が高い1分間は始値と終値がそれぞれ白い長方形の下底と上底の位置で表され（陽線と呼ばれる）、その上底、下底から高値、安値まで直線が引かれている。始値より終値が低い1分間は上底が始値、下底が終値とする黒い長方形で表され（陰線と呼ばれる）、高値、安値は陽線と同様に表される。始値と終値が同じ場合には同時線により示される。

図4と5には、30分間のローソク足チャートと出来高グラフの例を示している。ローソク足チャートでは、3のローソク足が1分ごとに並んでいて、右端が最も新しい1分間のデータとなる。画像の縦軸の最大値、最小値は用いたデータ区間での最小株価、最大株価に応じて決定し、背景はグレースケールの中間色のグレーとしている。出来高グラフは1分ごとの出来高数量を単純に棒グラフで表している。ローソク足チャートと同様に右端が最も新しい1分の出来高である。出来高グラフの最大値

は用いたデータ区間の最大値、最小値は0で固定している。本論文では、10分間、20分間と30分間のローソク足チャート、出来高グラフ、6種を入力として用いることとする。10分、20分と30分は画像に使う最も新しい値を揃え、さかのぼる時間をそれぞれ9分、19分と29分とし、画像の最も新しい値から1~5分後の予測のための入力情報に用いることとする。

株価は、株式を取引している市場参加者の中で、株価が上がるとする参加者の買い注文の株式数量が下がると考えた参加者の売り注文の数量より多くなると株価は上がり、下がると考える参加者が多く売り注文が増えると株価は下がる、つまり、市場の参加者の予測の大勢が変化の結果へ影響を与えるものとなっている。また、市場参加者の中には企業の経営状況、つまりファンダメンタルに応じて株式投資の意思決定をするケースや、ローソク足チャートや出来高グラフなど、過去の株価、出来高の情報により意思決定をする参加者もいる。本論文のケースではファンダメンタルな要素は直接的には用いることはせず、過去の株価、出来高情報の画像のみであるので、ファンダメンタルに大きな変動がない状況での予測に有効であると考えられる。ただし、ファンダメンタルの変化が与えた株価や出来高への影響はそれぞれの画像にある程度は含まれているとも考えることができる。これらのことから、過去の株価変動や出来高の情報はその後の株価変動へ一定の関連性があり、予測も可能と言える。ただし、ファンダメンタルや過去の変動にとらわれないノイズトレーダーと言われる参加者の存在も指摘されており、予測は簡単ではないことは容易に予想できる。

本研究で用いるデータは、東京証券取引所では、朝9時から11時30分の前場と12時30分から15時までの後場があるが、1分足であれば、前場、後場それぞれで、150分間のデータを観測することができる。前場、後場、あるいは後場と翌日の前場を連続と考えられる方法もあるが、本論文ではそれぞれ不連続であると考えられる。また、文献[1, 2]と同様な1つの画像に20分のデータを用いる場合と、アンサンブルに用いる予測器に多様性を持たせるためにデータ使用期間を10分間と30分としたものも追加した3つの時間間隔を用い、最大5分先までの予測を実施するので、最初の29分間と最後の5分間を除いて、前場、後場それぞれ、10分、20分、30分のローソク足チャート、出来高グラフが各116画像と教師データ（1から5分後に上がるか下がるか）を作成することができる。

用いる株価と出来高の期間は文献[1, 2]と同様に2017年8月1日から2018年1月26日の間の東京株式市場営業日の120日間分とし、27,840個のローソク足チャートと出来高グラフ、それぞれ10分間、20分間、30分間の6種で、画像総数は167,040となる。ただし、各予測器の作成の際は、6種の画像のいずれかを利用するため、利用するデータ数は27,840となる。なお文献[1, 2]では20分間の画像のみを用いていたため、前場、後場にそれぞれ126画像が作成でき、ローソク足チャート、出来高グラフ、それぞれ30,240個の画像が作成でき、本論文と評価サンプル数が違うため、比較する際には調整した値を用いる必要がある。データ総数の90%を畳み込みニューラルネットワークのパラメータ最適化に用いる学習データとし、残り10%学習に用いず評価用に用い、学習していないデータに対する予測性能評価に用いる。評価に用いる10%のテストデータは最初にランダムに選択し、複数の予測器作成においては、学習用データ、評価用データは固定して用いている。

入力データであるローソク足チャートは日経平均の1分足の四本値（始値、終値、高値、安値）から、10分、20分、および30分間のデータを切り出し、Excelの株価チャートグラフで52ピクセル×52ピクセル画像に加工した。用いる区間を1分ずつずらしながら同様に複数枚作成した。出来高グラフは、日経平均の1分足の出来高から10分、20分、および30分間分を切り出し、Excelの縦棒グラフを用い、ローソク足チャートと同様に1分ずつずらしながら52ピクセル×52ピクセルの画像を作成した。ローソク足チャートの入力画像は図4に示しているように、陽線を白、背景色をグレー、陰線と上下のヒゲを黒で作成しているが、畳み込みニューラルネットワークへの入力の際、白が1、背景が0、黒が1となるように標準化している[1, 2]。

教師データは、画像作成に用いた最終時刻から $T_{pred}$ 分後に、1分足終値基準で「上がる」(1で表

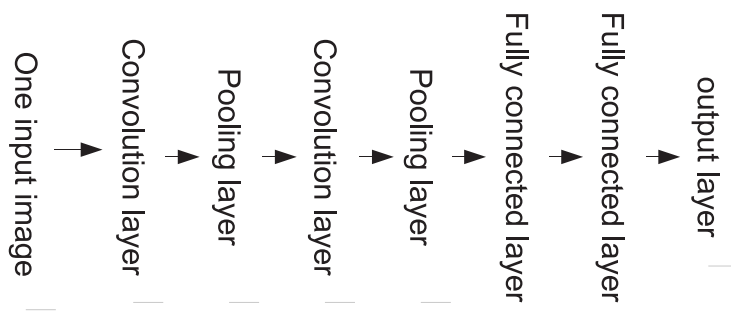


図 6: 1つの予測器の全体像

す)か、または「下がる(変わらない含む)」(0で表す)のカテゴリデータを用いることとする。これにより、入力した画像の  $T_{pred}$  分後に株価が上がるか下がるか学習させることができることとなる。シミュレーションにおいては、ローソク足チャートに用いた最後の時刻の1分足終値に比較し、 $T_{step} = 1, 2, 3, 4, 5$ 分後の1分足終値が「上がる」か、「下がる」かを予測の対象とした。

## 4.2 畳み込みニューラルネットワークの設定

畳み込みニューラルネットワークでは入力されるデータは2次元画像で、本研究では、4.1で示したローソク足チャートと出来高グラフ画像それぞれ10分間、20分間、30分間分のデータを用いた画像のいずれか1つのみを用いることとする。文献[1]では、入力に画像の組合せを用いることにより単独で用いるよりも予測精度が向上することを示したが、本論文では、最終的には多数の畳み込みニューラルネットワークによる予測器のアンサンブルにより最終的な予測を決定するため、複数画像を用いたことと同様の効果があると考えられるため、1つの予測器には1つの画像のみを入力することとした。図6に示すように、それぞれの予測器では、入力画像は畳み込み層、Pooling層、再度畳み込み層、Pooling層を経て、画像に含まれる特徴とその分布、さらにその特徴的な分布を抽出し、その出力を全結合層であるフィードフォワードネットへの入力とする。全結合層での中間層では、画像の特徴分布を引き延ばし1次元ベクトル化された情報を非線形処理し、その後ニューロンユニットを2つ準備した出力層で、2つのユニットのどちらの出力が大きいかにより、の  $T_{pred}$  分後に株価が「上がる」か「下がる」か決定し、各予測器の予測が出力される。

入力画像は、1つ目の畳み込み層ではパディングは用いず  $5 \times 5$ サイズのフィルタ20枚を用い、入力画像と各フィルタの類似度の分布を出力し、Pooling層で  $2 \times 2$ の範囲の最大値プーリングを用いる、2つ目の畳み込み層では、パディングなしで  $5 \times 5$ サイズのフィルタ50枚により畳み込み計算を行い、最大値プーリングにより出力を得る。この1つ目のPooling層の出力の2次元配列を、1行目の数値の後に2行目の数値、3行目、...と一列に並べた1次元配列得て、これを全チャンネル分並べた1次元配列を全結合層の入力とする。全結合層は1次元配列の要素数に対応した5,000ユニット、第2層2,000、出力層は「上がる」「下がる(変わらない含む)」の2カテゴリに対応した2ユニットとした。畳み込み層、および、全結合層の活性化関数には式(3)のシグモイド関数を用いた。また、畳み込み層、全結合層では、学習データに特化しすぎる過学習防止とアンサンブルの際の多様性確保のためにドロップアウト[5, 14]を用いている。全結合層で使用するユニットの比率、および畳み込みでの出力値を用いる比率を  $p, 0 \leq p < 1$ と表すこととする。ドロップアウトを用いると、畳み込みの類似度出力値や全結合層のユニット出力値の合計が平均  $1/p$ 倍となるので、出力値を  $p$ 倍して、出力

Image to use:10 minute candle chart					
$T_{pred}$	1	2	3	4	5
acc	0.503	0.507	0.516	0.509	0.504
profit	-395	12	-156	1,162	356
Rate of profit	-38.8%	0.6%	-5.1%	28.6%	7.0%
Image to use:20 minute candle chart					
$T_{pred}$	1	2	3	4	5
acc	0.513	0.552	0.562	0.607	0.604
profit	-23	1,719	3,324	6,398	5,391
Rate of profit	-2.3%	84.5%	108.9%	157.2%	106.0%
Image to use:30 minute candle chart					
$T_{pred}$	1	2	3	4	5
acc	0.522	0.589	0.633	0.613	0.660
profit	497	3,392	6,407	6,915	8,892
Rate of profit	48.8%	166.7%	209.9%	169.9%	174.8%

表 1: ローソク足チャートによる予測の  $T_{pred}$  とチャートに用いる株価分数の影響

値の合計の平均値が同程度となるような調整が必要となる。学習後の評価、つまりテストデータを用いた予測では、すべてのユニットを用いて出力を計算する。

### 4.3 単独の入力画像による株価予測シミュレーション結果

#### 4.3.1 入力画像と予測先タイムステップ $T_{pred}$ による影響

ここでは、まずアンサンブル学習の前の予備的なシミュレーションとして、10分、20分、30分のローソク足チャート、および出来高グラフの6種類の画像のうち、1つの画像を入力とした場合の予測精度について、予測のタイムステップ  $T_{pred}$  の違いによる影響を、文献 [1, 2] と同様にとまとめておく。ドロップアウトも用いるが、ユニット使用率  $p$  は 0.3 に固定しておくこととする。また、学習には利用可能なデータの 90% の学習データすべてを利用することとする。

予測の正確性を検証するため、文献 [1, 2] と同様以下の予測精度 (Accuracy、以降 acc と表す) を用いる。

$$\text{acc} = (\text{上がると予想して正解した数} + \text{下がると予測して正解した数}) / \text{全データ数}. \quad (12)$$

acc を学習データについて求めると、システムがどの程度学習できているかの目安となり、評価用データについて求めると、未学習データへの適用可能性、つまり汎化性能を確認することができる。全データ数とは学習データ、あるいはテストデータに用いたサンプル画像の枚数である。

また、acc 以外にテストデータについての総利益 (Profit) と年間推定利益率 (Rate of profit) を予測の能力の評価指標に用いることとする。総利益は学習に用いていないテストデータに対して学習済み畳み込みニューラルネットワークを適用し、予測を求め、その予測結果に従って投資をした場合の利益 (マイナスの場合は損失) の合計値である。具体的には、テストデータの1つの画像を入力し、「上がる」とシステムが予測した場合には、入力グラフに用いた最終終値で1株購入し、何分

Image to use:10 minute Volume image					
$T_{pred}$	1	2	3	4	5
acc	0.509	0.497	0.515	0.511	0.520
profit	302	649	1,365	1,004	1,783
Rate of profit	29.7%	31.9%	44.7%	24.7%	35.0%
Image to use:20 minute Volume image					
$T_{pred}$	1	2	3	4	5
acc	0.492	0.541	0.575	0.596	0.602
profit	68	1,555	3,421	4,800	6,693
Rate of profit	6.6%	76.4%	112.1%	117.9%	131.6%
Image to use:30 minute Volume image					
$T_{pred}$	1	2	3	4	5
acc	0.506	0.585	0.622	0.662	0.664
profit	441	3,557	6,242	8,904	10,712
Rate of profit	40.4%	174.8%	204.5%	218.8%	210.5%

表 2: 出来高グラフによる予測の  $T_{pred}$  とグラフに用いる出来高分数の影響

後の予測を行ったかの  $T_{pred}$  分後の終値で売却し、売却した価格から購入した価格を引いた値が利益（マイナスの場合は損失）となる。同様に「下がる」と予測した場合には、画像に用いた最終終値で空売り（株式を借りて売る）し、 $T_{pred}$  分後に株式を買い戻し、返却し、空売り価格から買い戻しの価格を引いた値を利益（マイナスの場合損失）とする。この個々利益（損失）をすべてのテストデータ（利用可能データ 27,840 個の 10% をテストデータとするため、2,784 サンプル）について合計した値が総利益（Profit）である。年間推定利益率（Rate of profit）は年間の推定利益を、投資に必要な最大資金額で割った値である。、年間の推定利益は、総利益（Profit）をテストデータのサンプル数で割り 1 回当たりの平均利益を求め、1 日の取引可能回数（作成可能画像数）236 とおおよその株式市場営業日 250 をかけてを計算した値である。投資に必要な最大資金額は、上がるか下がるか予測して売買した後、タイムステップ  $T_{pred}$  分間株価を保有し続けるので、 $T_{pred}$  に、用いる全データ区間の日経平均株価の平均額 21,557 をかけた金額としている。総利益と年間利益率は予測精度 acc と当然相関は高いが、より変動の大きい時の予測の誤りが多いと必ずしも高い値とならない場合も出てくる。

ただし、本研究では株式市場では直接取引のできない日経平均株価という指標を予測対象とし、ローソク足チャートや出来高のグラフに用いている最後の 1 分間の終了時において予測、取引が瞬時にできると仮定しており、さらには株式購入や空売りの手数料を考慮していないので、総利益や年間推定利益率は、あくまで仮想的なシミュレーションでの値となる。

表 1 には、ローソク足チャートの 10 分、20 分、30 分のいずれかの画像 1 つを入力とし、 $T_{pred}$  の違いによる予測精度の結果をまとめている。10 分間のローソク足チャートを入力とした結果では、予測精度 acc はランダムな予測の 50% をわずかに超えてはいるものの、総利益や年間推定力率が 20 分や 30 分に比べ低い値となっており、予測の精度や予測による投資の有効性が低いことが分かる。20 分、30 分のローソク足チャートによる予測では、 $T_{pred} = 1$  では 10 分間の結果と同様に予測精度や投資の有効性は他と比べ低くなっているが、 $T_{pred} = 2, 3, 4, 5$  では予測精度は 55% を超えて、20 分では  $T_{pred} = 4$ 、30 分では  $T_{pred} = 5$  で最大となっており、1 から 5 分の間では比較的先の時点の予測の方が精度が高いことを示している。予測に従った投資の総利益と年間利益率は、30 分ローソク足チャー



Image to use	20 minute candle chart			30 minute candle chart			
	$p$	0.3	0.7	1.00	0.3	0.7	1.0
acc		0.588	0.572	0.554	0.645	0.609	0.582
profit		4,947	3,694	2,639	8,598	6,582	4,633
Rate of profit		97.2%	72.6%	51.9%	169.0%	129.4%	91.1%

表 3: ローソク足チャートによる予測器の予測能力

トの  $T_{pred} = 5$  で最大となっている。20分と30分を比較すると予測精度も予測による投資の結果もすべての  $T_{pred}$  で30分の方が高い値となっている。これは、30分間のチャートの方がより多くの情報を持ち、その情報の特徴量を畳み込みニューラルネットワークが有効に活用していることが理由であると考えられる。

表2には、出来高グラフの10分、20分、30分間の画像のいずれかを入力とし、ローソク足チャートと同様に、 $T_{pred}$  の違いによる予測精度と総利益、年間推定利益率を示している。ローソク足チャートと同様に、10分間のグラフを用いた場合は、20分、30分の出来高グラフを用いたケースよりも予測精度は低くなっているが、ローソク足チャートよりは高い精度となり、それに伴って、総利益や年間利益率もローソク足チャートによる予測よりも高い値となっている。10分間のデータを用いて予測を行う場合、ローソク足チャートによる予測よりも、出来高グラフによる予測の方が適していることを示している。株価変動の直前（ここでは10分間）の株価変化はその後の変動の予測にあまり有効ではないが、直前の出来高の変動はその後の株価の変動と関連性があることを示唆している。出来高が株価の変動に先導して変化するケースが観測されることが知られているが、この結果はこのことを裏付ける結果となっていると言える。ただし、ローソク足チャートと同様に  $T_{pred} = 2, 3, 4, 5$  では、10分間よりも20分間、さらに30分間の出来高情報を用いた方が予測精度と総利益、年間利益率が高い結果となっている。

#### 4.3.2 アンサンブルに用いる予測器

次にアンサンブル学習を用いた場合予測結果をシミュレーションにより検証していくが、アンサンブルに利用する個々の予測器について分析をしておく。ここで用いる畳み込みニューラルネットワークの予測器は、後のアンサンブル学習に利用する。そのため、すでに述べたように各予測器の予測に共通性が強い場合、多様性がなくアンサンブルによる予測精度向上効果が低くなるため、十分な多様性を持たせるよう設定を様々に変更した上で学習をさせている。また、ランダムな予測精度約50%を安定的に超える予測器を用いないと予測精度が低下する場合も考えられるため、4.3.1の入力画像による結果から、用いる入力画像はローソク足チャート、および出来高グラフ、それぞれについて20分間、30分間の4種類のいずれかとすることとする。10分間の入力による予測器は利用しないこととする。また、多様性を持たせるためにドロップアウトのユニット使用率0.3、0.7、1.0の3種類を用いることとする。入力画像の4種類とドロップアウト3設定を組合せ、12パターンの畳み込みニューラルネットワークシステムを準備し、さらに学習データにバギングの復元抽出により12パターンそれぞれに3回ずつ予測器を作成し直し、総数36個の畳み込みニューラルネットワークによる予測器を作成することとした。また、4.3.1における予備実験の最も予測精度、総利益、利益率の高い30分間の出来高画像の5分先予測と比較するため、アンサンブルによる予測の予測先ステップ数  $T_{pred}$  は5分後を用いることとした。

表3には、ローソク足チャートの20分間と30分間の画像を入力とした場合に、ドロップアウトの

Image to use	20 minute Volume image			30 minute volume image			
	$p$	0.3	0.7	1.00	0.3	0.7	1.0
acc		0.549	0.545	0.551	0.609	0.601	0.588
profit		2,737	3,139	2,990	7,596	6,689	5,801
Rate of profit		53.8%	62.8%	58.8%	149.3%	131.5%	114.0%

表 4: 出来高グラフによる予測器の予測能力

ユニット使用率  $p$  を 0.3, 0.7, 1.0 と変更した場合のテストデータについての予測精度、総利益、年間予想利益率を示している。ただし、4.3.1 の結果と違い、学習データをすべては用いず、バギングの復元抽出により作成されたサブデータセットとなっている。各設定でバギングによるサブデータセットで 3 個の予測器を作成し、表の数値はその 3 個の予測器の結果の平均値となっている。20 分間、30 分間のローソク足チャートの場合のいずれもユニット使用率  $p$  が小さい方がテストデータについての予測精度、総利益、年間利益率は高くなっている。これは、ドロップアウトにより学習時のユニット使用率が低い方が学習データに特化しすぎる過学習が起こりにくく、汎用性が高い予測器となり、学習に用いていないテストデータの予測性能が高いことが理由と考えられる。4.3.1 の予備実験の結果と比べると、 $p = 0.3$  のケースで 20 分も 30 分も予測性能が低下しているが、これはバギングにより全学習データ中の使用データの割合が少なくなり、学習データのサブデータセットに過学習が発生しやすくなったことが原因であると考えられる。ただし、アンサンブル学習においては個々の予測器の精度よりもアンサンブルの結果の予測の精度が重要であるので、個々の予測器の予測性能の低下は大きな問題とはならない。

表 4 には、同様に、20 分間、30 分間の出来高グラフを入力としたときのドロップアウトのユニット使用率  $p = 0.3, 0.7, 1.0$  のテストデータについての予測精度、総利益、年間予想利益率の影響を示している。出来高のケースも学習データをすべては用いず、バギングの復元抽出により作成されたサブデータセットにより各種設定で 3 個ずつ予測器を作成し、その平均値となっている。20 分間の出来高グラフを用いた結果は、 $p$  を変更しても他の画像に比べ大きな変化はなく、予測精度 acc は 55% 前後の値となっている。この数値は、全学習データによる予測の結果である表 2 の結果からの低下の原因はローソク足チャートと同様に過学習が原因であると考えられるが、出来高グラフの方がローソク足チャートよりも単純な図であるので、学習に用いるサンプルデータの減少による汎化能力低下が大きいことが理由であると考えられる。30 分間の出来高グラフの場合は、ユニット使用率の値が小さいほどやや予測精度が良くなる傾向があるが、やはり過学習が原因と思われる予測精度の低下がみられる。しかし、予測精度はおおよそ 60% 前後となっており、アンサンブルに用いることを前提とすれば、十分な性能であると考えられる。

すでに述べたようにアンサンブルにより予測精度の向上効果を得るには、用いる予測器の予測に共通性が高くない必要がある。つまり正解率は似たような値であっても、個々のデータに対する予測出力の相関が低く、個々のデータの正解、不正解にバラつきがあることがアンサンブル学習の有効性を高めることになる。ここでは個々の予測器の学習に用いていないテストデータに対する予測出力の多様性について検証を行う。

図 7 には、用いる画像の種類 4 種、ドロップアウトユニット使用率、バギングによるサブデータセットの 36 個の組合せの畳み込みニューラルネットワークの入力したテストデータ画像に対する出力値のヒートマップを示している。横軸は 2,784 のテストデータ画像のサンプル番号、縦軸は 36 の予測器の番号 (0 番から 35 番) に対応している。各予測器の出力を (10) 式のソフトマックス関数により、0 から 1 に調整した出力値が 1 に近い、つまり、5 分後の株価が上がり強く予測するほど黒

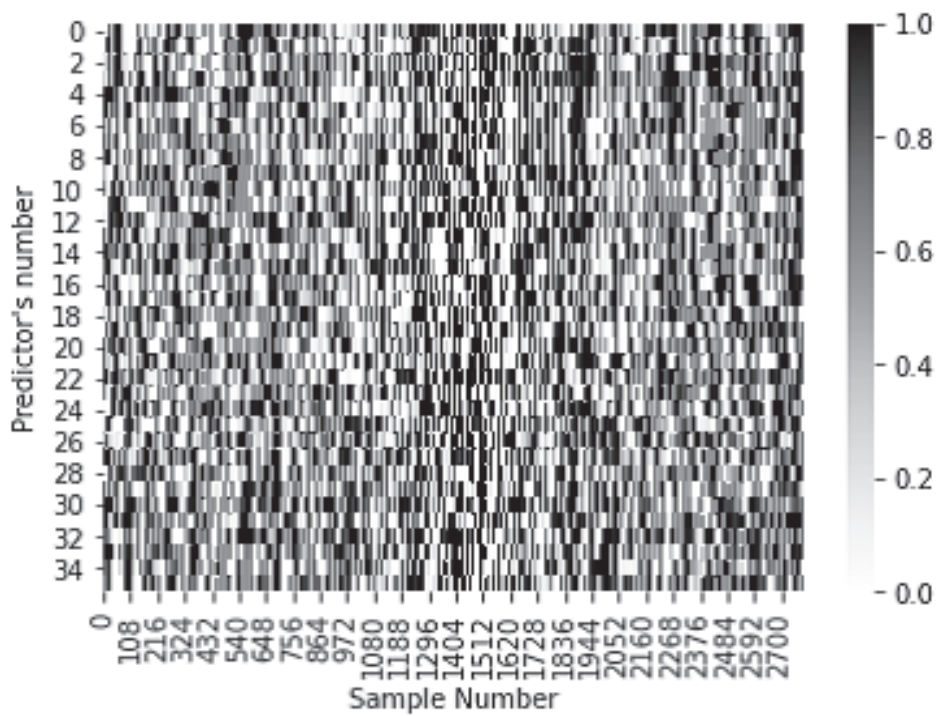


図 7: 予測器の出力の多様性

test acc	profit	Rate of profit
0.707	12,462	306.2%

表 5: soft vote による予測能力

で表示され、株価が下がると強く予測するほど白で表示されている。この図の見方としては、縦方向に同じ入力画像に対する 36 個の予測器の出力が並んでいるので、縦方向に黒や白の直線が現れると、各予測器による予測に共通性が高いことを示している。予測器番号の付け方は、20 分のローソク足チャートを入力とする 12 個の予測器 ( $p = 0.3$  の 3 つのサブセットによる予測器が 0,1,2、 $p = 0.7$  の 3 つのサブセットによる予測器が 3,4,5、 $p = 1.0$  の 3 つのサブセットによる予測器の順番 6,7,8)、その後 30 分のローソク足チャート、20 分の出来高グラフ、30 分の出来高グラフの順で、20 分のローソク足チャートの  $p$ 、サブセットと同様な順番付けとしている。7 を見ると、白や黒の予測が数個続いている様子、つまり、同じ入力画像種の同じ  $p$  など、似たような設定の予測器は予測に共通性があることを示しているが、縦方向に黒や白の直線がはっきりと現れることはないため、36 の予測器全体としてはある程度の多様性が確保されていることが確認できる。なお、36 個の予測器の出力のすべての 2 組の組合せについて、積率相関を計算すると、最大 0.434、最小 -0.021、平均 0.128 と相関から判断しても、高すぎる共通性は持っていないことが確認できる。

#### 4.4 アンサンブル (soft vote) による予測精度の分析

次にアンサンブル学習による予測を単独の予測器による予測と比較することとする。まずは、アンサンブルでも単純な個々の予測器の多数決により最終的な予測を決定する方法を用いることとする。多数決においては、個々の予測器は「上がる」か「下がる」かの 2 値の投票ではなく、(10) 式のソフトマックス関数により計算された出力値を投票することとする。つまり、投票は 0.0 ~ 1.0 の間の値であり、0.5 を超える場合には「上がる」と予測するが、1.0 に近づくにつれその予測の正しさに対する自信が強いことを表している。反対に、0.5 以下である場合は、0 に近いほど「下がる」という予想の自信が強いこととなる。4.3.2 に示した 36 個の予測器の出力の平均値を計算し、0.5 を超えると予測器集団の多数決、つまりアンサンブルによる予測は「上がる」、0.5 以下で「下がる (変わらないも含む)」と決定することになる。

表 5 には soft vote による予測の精度と予測に従った投資行動の総利益、年間推定利益率を示している。学習に用いていないテストデータの予測精度 acc は 0.707 となっている。全学習データを用いた単独の畳み込みニューラルネットワークによる予測の中で最も精度が高い、30 分間の出来高グラフの画像により 5 分先の株価変動を予測した表 2 の結果の予測精度は 0.664 であり、4.3% の予測精度向上が確認できる。テストデータに対して予測に従った投資行動をした場合の総利益も 10,712 から 12,462 と 16.3% 増加し、年間推定利益率も 210.5% から 306.2% へ増加している。総利益の増加は予測精度が向上し、投資に失敗する回数が減り、成功する回数が増えたということも理由であるが、増加の割合が精度の上昇率  $4.3/66.4 = 6.5\%$  よりも高いため、単独の予測よりもアンサンブルによる予測の方が、より値動きが大きい時の正解率が向上しているということが示唆されている。ただし、図 2 に示した正解率 55% で独立な予測を行う予測器によるアンサンブルによる hard vote の結果 0.719 よりも低い値となっている。つまり、36 個の予測器のそれぞれの予測精度の平均は 58.3% で、図 2 の予測器よりも高いものとなっているが、図 2 の結果よりもアンサンブルの効果が低い結果となっている。これは、36 個の予測器の予測に共通性 (同じサンプルに対して同じように正解したり間違ったりする傾向) があることが原因だと考えられる。

test acc	profit	Rate of profit
0.716	13,116	322.2%

表 6: stacking による予測能力

#### 4.5 スタッキングによる予測性の分析

4.4では単純な予測出力の集計により集団の予測を決定している。ここでは、36の予測器の出力（それぞれ上がると予測する確率である0.0から1.0で表される1つの値）を組合せてベクトルとし、これを2.3で示した全結合フードフォワードニューラルネットワークと同様な最終段階の予測器（blender）の入力として、各予測器の出力の重要度を学習させるスタッキングを用いた予測の予測精度についてまとめる。単純な集計に比べて、より予測精度の高い予測器を重視するようなシステムになり、予測精度が向上することが期待される。

表6には、スタッキングを用いたアンサンブル学習の予測精度、総利益、年間推定利益率をまとめている。表2にある単独の予測器でもっとの高い予測精度の30分の出来高グラフによる5分後の予測の予測精度0.664より5.2%の予測精度の向上が見られる。表5のsoft voteと比較すると、0.9%予測精度が向上しているが、単独予測器による予測精度に対するsoft voteの予測精度の向上率に比べるとわずかな上昇にとどまっている。総利益は13,116となり、soft voteから5.2%上昇し、年間推定利益率は322.2%となった。これらのことから、単純な集計から大幅な予測精度向上は見られなかったものの、値動きの小さなケースよりも、より値動きの大きなケースの予測の正解率が向上しているということが読み取れる。これは各予測器の予測に対する自信度をblenderが調整効果が表れていることを表している。

## 5 むすび

本研究では、1分足のローソク足チャートと出来高グラフを入力とする畳み込みニューラルネットワークを用いた株価変動予測について、アンサンブル学習を導入した際の予測精度の向上について、シミュレーションによる検証により示した。筆者が以前示した単独の予測器による予測と比較し、アンサンブルによる予測では予測精度、株価収益が向上することを示した。今後は、さらなる予測精度の向上や、多数の予測器を参加者とした仮想市場での政策分析について検討を進めたいと考えている。

## 謝辞

本研究はJSPS 科研費 JP17K01267の助成を受けたものであり、ここに感謝の意を表します。

## 参考文献

- [1] 池田欽一, ディープラーニングによる株価予測の入力情報の多重化による予測精度向上についての分析, 北九州市立大学商経論集, 55(1,2,3,4), 1-17, 2020, [https://kitakyu.repo.nii.ac.jp/?action=pages\\_view\\_main&active\\_action=repository\\_view\\_main\\_item\\_detail&item\\_id=730&item\\_no=1&page\\_id=13&block\\_id=294](https://kitakyu.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=730&item_no=1&page_id=13&block_id=294).



- [2] 池田欽一, 株価ローソク足チャート画像を用いた畳み込みニューラルネットワークによる株価変動予測, 北九州市立大学商経論集,54(1,2,3,4), 1-18, 2019, [https://kitakyu.repo.nii.ac.jp/?action=pages\\_view\\_main&active\\_action=repository\\_view\\_main\\_item\\_detail&item\\_id=678&item\\_no=1&page\\_id=13&block\\_id=294](https://kitakyu.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=678&item_no=1&page_id=13&block_id=294).
- [3] 池田欽一, 林田実, ディープラーニングの株価予測への応用, 北九州市立大学商経論集,52(1,2,3,4), 13-26, 2017, [https://kitakyu.repo.nii.ac.jp/?action=pages\\_view\\_main&active\\_action=repository\\_view\\_main\\_item\\_detail&item\\_id=552&item\\_no=1&page\\_id=13&block\\_id=294](https://kitakyu.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=552&item_no=1&page_id=13&block_id=294).
- [4] Yann Lecun and Léon Bottou and Yoshua Bengio and Patrick Haffner, “Gradient-based learning applied to document recognition”, Proceedings of the IEEE, 2278–2324,1998.
- [5] 岡谷貴之, 深層学習, 講談社,2015.
- [6] Zhi-Hua Zhou, Ensemble Methods: Foundations and Algorithms, CRC press,2012.
- [7] オーレリアン ジュロン (著), 下田 倫大 (監訳), 長尾 高弘 (訳), scikit-learn、Keras、TensorFlow による実践機械学習 第2版, オライリージャパン, 2020.
- [8] Rosenblatt F., “The Perceptron: A probabilistic model for information storage and organization in the brain”, Psychological Review, 65, 386–408, 1958.
- [9] Minsky M. and Papert S., Perceptrons: An Introduction to Computational Geometry. MIT Press, MA,1969.
- [10] Rumelhart D. E. , Hinton G. E. and Williams R. J. , “Learning internal representations by error propagation. Parallel Distributed Processing”, 1, MIT Press, MA, 318-362, 1986.
- [11] Diederik P. K. , Jimmy L. B., “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION”, Conference paper at ICLR 2015,1-15,2015.
- [12] “A Powerful, Flexible, and Intuitive Framework of Neural Networks”,<http://chainer.org/>.
- [13] 神鷲敏弘 (編) , 麻生英樹, 安田宗樹 他 (著) , 深層学習, 近代科学社, 2015.
- [14] Srivastava N. , Hinton G. E. , Krizhevsky A. , Sutskever I. , and Salakhutdinov R, “Dropout: A simple way to prevent neural network from overfitting”, Journal of Machine Learning Research, 15(Jun), 1929–1958, 2014.