

# FXにおける日足・4時間・15分のチャートのCNNによる最適化 Optimization of Daily, Weekly, and 15 Minute FX Chart by Using CNN

河合 継<sup>1\*</sup> 小澤 昂<sup>1,2</sup> 大川 堯郁<sup>1,3</sup>  
Kei Kawai<sup>1</sup>, Takashi Ozawa<sup>1,2</sup>, Takafumi Okawa<sup>1,3</sup>

<sup>1</sup> クリスタルメソッド株式会社

<sup>1</sup> Crystal method co.ltd

<sup>2</sup> 東京工業大学理学院数学系数学コース数理ファイナンス専攻

<sup>2</sup> Department of Mathematics, Tokyo Institute of Technology

<sup>3</sup> 東京大学工学系研究科物理工学専攻

<sup>3</sup> Department of Applied Physics, Graduate School of Engineering, The University of Tokyo

**Abstract:** In recent years, many people attempt to apply machine learning to a broad range of fields. At the same time, thanks to improvements of the information technology, we can use the enormous number of the past financial data as a learning data. In this paper, we describe prediction results of the foreign exchange price. The prediction model was made by using Convolutional Neural Network(CNN). Then, we compare the results of the model which learned numeric data with the model which learned image data.

## 1 はじめに

近年では、人工知能分野の技術を金融市場の分析や予測に応用する動きが活発になっている。同時に、情報技術の発達により、これまでの膨大な金融市場のデータをインターネットを通じ、学習データとして用いることが出来るようになった。

前回の研究では、為替レート(USD/JPY)のチャート画像による15分後の値動き予測を行った。15分足のチャート画像を用いて2006年のデータで学習を行い、2007年から2016年のデータで検証を行った。検証では、年率2%のリターンとなった。今回の研究では、より予測の精度を向上させるため、日足/4時間足/15分足のチャート画像を入力データとした。また、為替レートの時系列データ(数値データ)を入力データとしたCNNによる学習・検証も行い、画像データによる結果との比較・考察を行った。

## 2 手法

### 2.1 前提

実際にトレードを行う際には、日足、4時間足、15分足を情報として用いる事が多い。そのため、今回の研究では、それらの三つの情報を一つのデータにまとめ、学習・検証を行った。過去データの取り込みにおいては、学習データに未来のデータが混入しないように、日足は前日分、4時間足は予測する15分足が含まれない時刻までの4時間足を学習データとして取り込んだ。また、現在の日付の途中までの未確定足の情報は、実際のトレードでは重要なシグナルとなるが、今回は4時間、15分足にその情報が含まれていると考え、前日の日足を用いてデータ構築を行った。

今回の研究で用いたデータなどについては、下記の通り。

通貨ペア: USD/JPY, 日足データ数: 3115, 4時間足データ数: 18956, 15分足データ数: 304567, 画像サイズ: 892 × 500, チャート画像一つあたりの足本数: 211, チャート数値データファイル数: 199, 利用フレームワーク: chainer, 使用言語: python.

\*連絡先: クリスタルメソッド株式会社  
〒162-0844 東京都新宿区市谷八幡町1 新高ビル 904  
E-mail: kawai@crystal-method.com

## 2.2 準備

### 2.2.1 画像データ・数値データ

今回の検証においては、画像データ量が 30 万枚程度となるため、機械学習時のマシンパワーが必要になると考えられた。そのため、GPU を用いた学習環境として Nvidia GTX 1080 Ti, Nvidia GTX 1060 Ti を使用して計算を行った。学習用ハードウェアの構成は、それぞれ数値学習マシンでは GTX 1080 Ti 1 枚, GTX 1060 Ti 1 枚, 画像学習マシンでは GTX 1080 Ti 2 または 3 枚とした。<sup>1</sup>

また、画像データ、数値データの共通の準備として、価格データを SQL データベース化し、検証時に問い合わせを行い、売買タイミングでの利益などのデータを抽出し、データベースには 15 分足のデータを入力した。また、当該 15 分足の終値で約定し、次足終値でクローズする仕組みとした。値幅についても同様に、当該足終値と次足の終値の差を用いて検証を行った。

今回の検証では、ラベルの振り方を 3 種類用意し、それぞれの振り分け方に対して学習と検証を行った。ラベルの振り方は、表 1 の様である。

表 1: ラベルの振り分け

| ラベル     | 1            | 2        | 3         |
|---------|--------------|----------|-----------|
| 振り分け方 1 | -10pip 10pip | 10pip 以上 | -10pip 以下 |
| 振り分け方 2 | -15pip 15pip | 15pip 以上 | -15pip 以下 |
| 振り分け方 3 | -20pip 20pip | 20pip 以上 | -20pip 以下 |

### 2.2.2 画像データ

- チャート画像の取得・加工プログラム

学習データに用いるためのチャート画像の取得には、MQL4, C++ によって作成したプログラムを用い、MetaTrader<sup>2</sup> のチャートをコマ送りにし、USD/JPY の 15 分足チャート画像をスクリーンショットとして 30 万枚分を png 形式で保存した。また、同様にして 15 分足の日付、時間に対応する日足、4 時間足のチャート画像を読み込んだ。その後、png 形式で保存した画像データに対し、ビット深度やサイズ変更を行う加工プログラムを作成、実行した。

日足、4 時間足、15 分足の取得データのサンプルは、それぞれ以下の図 1, 2, 3 に示す通りである。

また、画像はそれぞれ 8bit 画像として生成し、同時刻の日足、4 時間足、15 分足のサンプルを RGB 画像として統合する事で 1 枚の画像にまとめ、その画像を CNN で学習させた。(図 4)



図 1: 日足



図 2: 4 時間



図 3: 15 分足

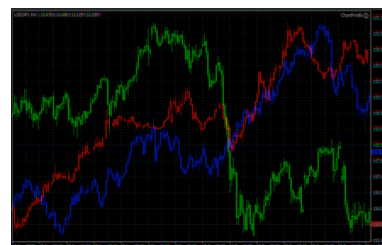


図 4: 日足(青)・4 時間足(緑)・15 分足(赤) 統合データ

<sup>1</sup>近年の GPU においては、PCI バスが 16 レーンの帯域となっているが、現実的には 8 レーン × 4 のマザーボードと速度があまり変わらないため、安価な 8 レーン × 4 のボードを選択した。この学習については CPU 性能もかかってきており、コア数の多いものの学習がより早く進んだ。2008 年の 8 コアマシンと 2017 年の 8 コアマシンでは、計算時間に大きな差は見られなかった。

<sup>2</sup>外国為替証拠金取引プラットフォーム  
<https://www.metatrader4.com/ja>

- Chainer CNN プログラム畳み込みニューラルネットワークを利用し、上記画像を学習するプログラムを Python により作成した。
- Chainer 学習結果返却プログラム  
上記で学習した学習済みデータセットに、学習させていないデータを入力として与えることで売り、買い、または何もしないかを返却するプログラムを Python により作成した。
- チャート価格取得プログラム  
あるチャート画像を対象として、チャートの開始時刻(チャート左端)、チャートの現在時刻(チャートの右端)、次の足の時刻(チャートの右端+1)、及び右端の終値、右端+1の終値、両者の価格差を全ての15分足で取得した。
- 結果検証プログラム  
学習済みデータセットが返却した売却情報と上記のデータベースに対してSQLを発行し、Excelにまとめるプログラムを Python, Postgres により作成した。

### 2.2.3 数値データ

数値データに関しては、USD/JPYのFXチャートにおいて、日足・4時間足・15分足データをメタトレーダーによりCSV形式で取得した。その後、CSVデータをプログラムを用いて対象とする15分足、4時間足、日足データをそれぞれ200本分をファイルにまとめた。出力データは、高安/始終ごとに差分と平均の計算を行った。数値データの例は、表2に示す通りである。

表 2: 数値データの例

|      | 日付         | 時刻          | 終値 - 始値     |
|------|------------|-------------|-------------|
| 日足   | 2016.03.21 | 00:00       | 40.756      |
| 4時間足 | 2016.11.25 | 04:00       | 0.608       |
| 15分足 | 2016.12.30 | 22:15       | 0.095       |
|      | 高値 - 安値    | (始値 + 終値)/2 | (高値 + 安値)/2 |
| 日足   | 0.437      | 111.587     | 111.7165    |
| 4時間足 | 0.398      | 113.555     | 113.555     |
| 15分足 | 0.065      | 116.7725    | 116.7835    |

### 2.2.4 検証実施時のモデル

検証時のモデルには、数値・画像ともにCNN(畳み込みニューラルネット)を用いた。画像畳み込みニューラルネットに関しては、日足、4時間足、15分足をそれぞれRGBにマッピングし、入力層の大きさは使用する画像データ数によって変更を加えた。数値データ、及び画像データを学習させた際のモデルは、それぞれ図5、図6の通りである。<sup>3</sup> 図5において、①を4回実行し、②を2回実行した。また、図6においては、①を3回実行し、②を2回実行した。

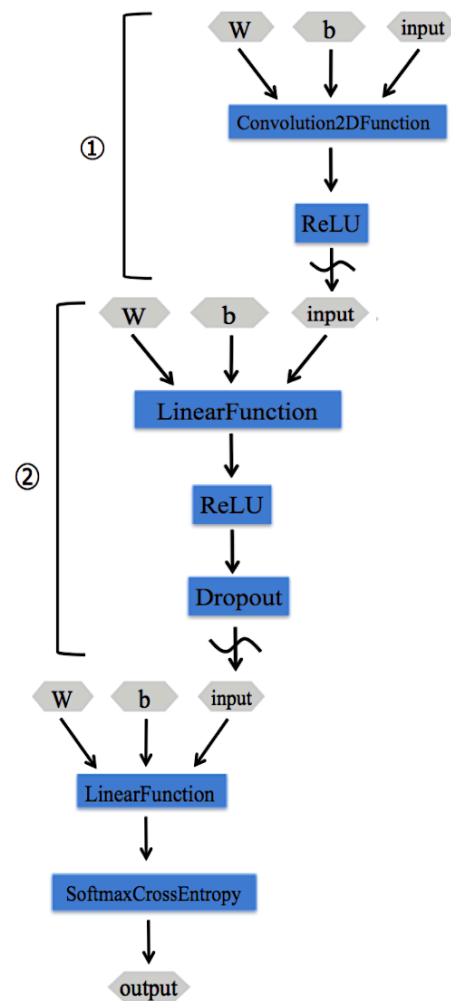


図 5: 数値データ学習モデル

<sup>3</sup>学習結果に偏りが生じてしまう現象が起こってしまった際には、学習する時の一つ目の中間層のフィルター枚数を入力する画像データの量に比例して変化させることによって、軽減することが出来た。

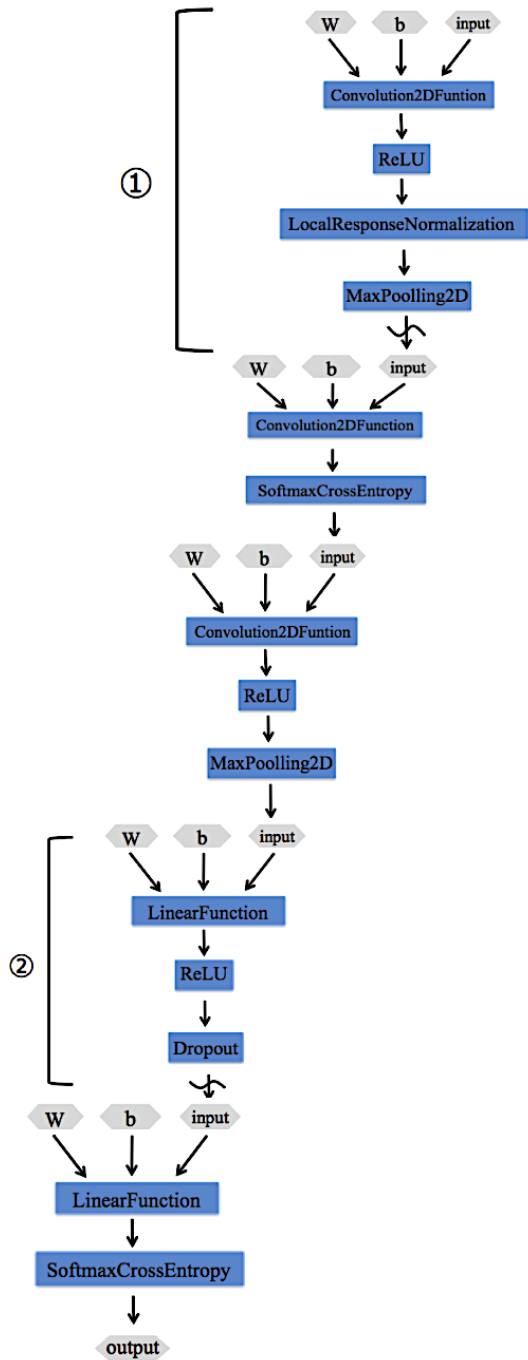


図 6: 画像データ学習モデル

## 2.3 検証方法

検証方法は、下表 3 に示すように、2006 年～2008 年を学習させたモデルについて学習させた年度以降の検証を行った。また、値動きの幅に応じて、3 つのパターンに対して検証を行った。

表 3: 学習・検証年度

| 学習年度      | 検証年度      | 勝敗の値幅 | 比較する点             |
|-----------|-----------|-------|-------------------|
| 2006～2008 | 2009～2015 | 10pip | 勝敗の値幅<br>数値と画像の比較 |
| 2006～2008 | 2009～2015 | 15pip | 勝敗の値幅<br>数値と画像の比較 |
| 2006～2008 | 2009～2015 | 20pip | 勝敗の値幅<br>数値と画像の比較 |

## 3 結果

### 3.1 数値データによる結果

数値データを用いて、2006 年～2008 年までを学習させたモデルの結果を表 4 に示す。また、このモデルにおける損益グラフを、図 7～9 に示した。(縦軸：円)

表 4: 検証結果 (2006～2008 年を学習)

| 指標 / 2009-2016 を検証 | 10pip | 15pip  | 20pip |
|--------------------|-------|--------|-------|
| Retuen(単利)         | 1.63% | -0.54% | 1.84% |
| Retuen(複利)         | 1.56% | -0.60% | 1.84% |
| Sharp Ratio(単利)    | 0.391 | -0.164 | 0.872 |
| Sharp Ratio(複利)    | 0.373 | -0.18  | 0.869 |
| 勝率                 | 50.5% | 50.2%  | 50.8% |
| 取引回数               | 26941 | 15507  | 4276  |

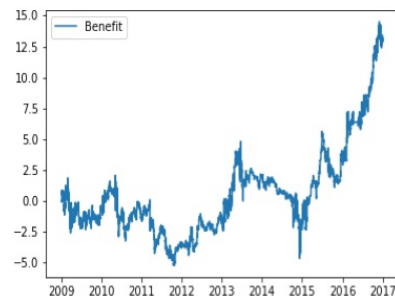


図 7: 数値データ損益 (10pips)

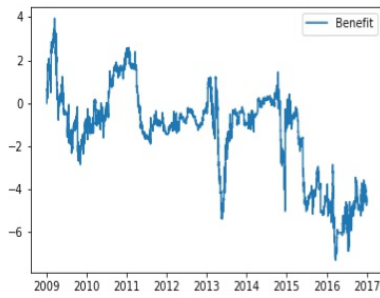


図 8: 数値データ損益 (15pips)

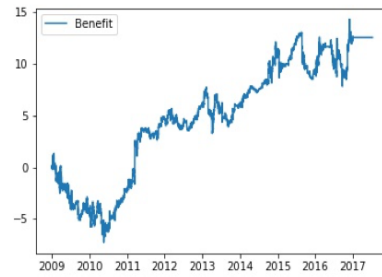


図 10: 画像データ損益 (10pips)

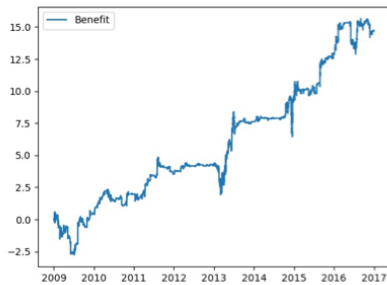


図 9: 数値データ損益 (20pips)

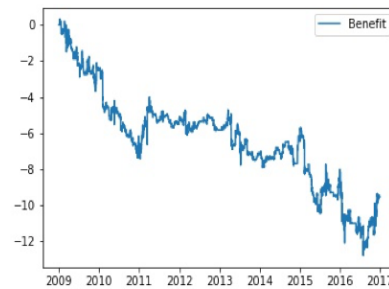


図 11: 画像データ損益 (15pips)

### 3.2 画像データによる結果

また、画像データを用いて、2006年～2008年までを学習させたモデルの結果を表5に示す。この時の損益グラフは図10～12の様であった。(縦軸：円)

画像データを用いた学習においては、2006年度の15分足のみを学習させた前回研究と比較するため、2006年の15分足、4時間足、日足を学習させた結果を表6、図13に示す。(縦軸：円)

表 5: 検証結果 (2006～2008年を学習)

| 指標 / 2009-2016 を検証 | 10pip | 15pip  | 20pip |
|--------------------|-------|--------|-------|
| Retuen(単利)         | 1.63% | -0.54% | 1.84% |
| Retuen(複利)         | 1.56% | -0.60% | 1.84% |
| Sharp Ratio(単利)    | 0.391 | -0.164 | 0.872 |
| Sharp Ratio(複利)    | 0.373 | -0.18  | 0.869 |
| 勝率                 | 0.505 | 0.502  | 0.508 |
| 取引回数               | 26941 | 15507  | 4276  |

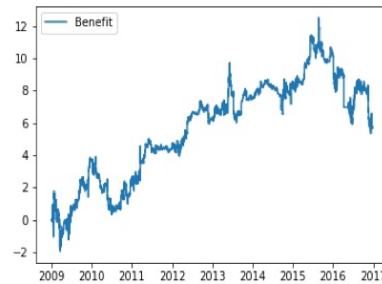


図 12: 画像データ損益 (20pips)

表 6: 検証結果 (2006 年を学習)

| 指標 / 検証年度       | 2007-2016 |
|-----------------|-----------|
| Retuen(単利)      | 1.016%    |
| Retuen(複利)      | 1.076%    |
| Sharp Ratio(単利) | 0.283     |
| Sharp Ratio(複利) | 0.299     |
| 勝率              | 51.0%     |
| 取引回数            | 16879     |

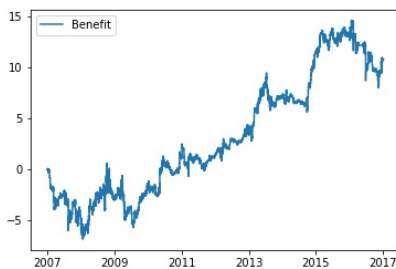


図 13: 画像データ損益 (2006 年を学習)

## 4 考察

前回の研究発表後に取得した, 2006 年の 15 分足のみを学習データとして用いたモデルの損益は 1000pip, 年利 1%ほどの利益であったが, 今回の日足情報と 4 時間足を 15 分足に加えて学習を行ったモデルの損益は, 1500pip, 1.5%程度であり, 改善が見られた.

また, 画像データでの結果と数値データでの比較では, 若干数値の結果が上回った. 数値データ・画像データ共に, 15pip のラベル付けにおいてマイナスとなった. 原因は特定できておらず, どの場合においても勝率の有意性は見られなかった. これらを踏まえ, 画像データについて, 今後改善すべき点としては日足, 4 時間足, 15 分足を一つの画像としたが, それぞれのコンテキストを踏まえた学習となっていない可能性があるため, 別画像として実施したい. また, 数値データについては, 一部複数回実行した場合, 結果が全く同じにならなかったことがあったため, これらの原因特定はこれから実施したいと考えている. 更に, 経済指標をモデルを用いることによって, より良いモデルを構築出来る可能性があるため, 今後研究を進めたいと考えている.

また, データ量などは数値データの方が少なく学習を行うことができた. また, それぞれの学習データを用いた際のメリット・デメリットは以下のようなものが考えられた.

### 数値データを使用するメリット

- より正確な情報を用いることができる.
- 計算量が格段に減る.

### 数値データを使用するデメリット

- CNN は画像データについて解析を行うために開発されている歴史がありの本来の目的を果たせない.

### 画像データを使用するメリット

- 人間が目で見たま取引を行うという目的を果たすためには CNN モデルが必要で, 画像データは CNN モデルの目的に合致している.

### 画像データを使用するデメリット

- 入力の次元が多次元になり, 計算に負担がかかる.

## 謝辞

本研究は人工知能学会金融情報学研究会の先生方に多大なご指導, ご鞭撻を頂き完成したもので, 心より感謝申し上げます. また, 共同研究をしている学生のお二人をご指導されている先生方, 大学関係者の方々もこのように優秀な人材を育てられていることに対して, 非常に強い感謝の念を抱いております.

## 参考文献

- [1] ジョン・コーツ, 訳 小野木明恵: トレーダーの生理学, 早川書房, (2013)
- [2] 新納 浩幸: Chainer による実践深層学習, オーム社, (2016)
- [3] ラリー・ペサベント: フィボナッチ逆張り売買法, パンローリング, (2008)
- [4] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner: Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE*, 86(11):2278-2324(1998)