金融取引戦略獲得のための複利型深層強化学習

Compound Deep Reinforcement Learning to Acquire Trading Strategies

松井 藤五郎 [1,2*] 片桐 雅浩 [2]

[1] 中部大学 生命健康科学部 臨床工学科 [2] 中部大学 工学部 情報工学科

Abstract: 本論文では、深層強化学習を複利型に拡張した複利型深層強化学習を提案する。複利型深層強化学習は、報酬の替わりに利益率を観測し、利益率の複利効果を最大化する行動規則を学習する複利型強化学習において、行動価値関数をニューラル・ネットワークで表し、深層学習を用いて行動規則を学習する。また、金融商品（日本国債）の取引戦略の獲得に複利型深層強化学習を用いた例を示す。

1 はじめに

我々は、強化学習 (RL: reinforcement learning) における報酬 (reward) の替わりに利益率 (rate of return) を観測し、利益率の複利効果 (compound effect) を最大化する行動規則を学習する複利型強化学習を提案してきた [松井 13a, 松井 11b, Matsui 12, 松井 11a]。複利型強化学習 (compound RL) では、将来の利益率を二重指数関数を用いて割り引き、複利利益率の対数を取ることによって、行動価値関数を従来の強化学習と同様の Bellman 方程式の形で表し、従来の強化学習アルゴリズムを容易に複利型強化学習アルゴリズムに拡張することを可能としている。

我々は、これまでに、強化学習と複利型強化学習を金融商品の取引戦略の獲得に応用してきた [松井 13a, 松井 11b, Matsui 09]。これらの研究では、状態変数として終値と移動標準偏差を用いており、状態変数を相対化することによって終値や移動標準偏差が大きく変動する市場でも有効な取引戦略を獲得できるようにしている。従来手法では、行動価値関数と投資比率を放射基底関数 (radial basis function) を用いて線形近似していた。しかしながら、放射基底関数を用いた線形関数近似は、状態変数を増やすと指数的に状態が複雑になり、学習が難しくなってしまうという問題があった。

深層学習 (deep learning) は、大規模で複雑なニューラル・ネットワーク (artificial neural network) を GPU の計算能力を用いて学習するものであり、画像認識の分野で特に顕著な成果を上げている。2013 年に、Mnih らによって、深層学習を用いた強化学習の手法である深層強化学習 (deep RL) [Mnih 13] が提案された。深層強化学習アルゴリズムの Deep *Q*-Network は、コンピューター・ゲームの画面（画像）を入力、ゲームの得点を報酬として、高得点を取る操作方法を学習させたところ、人間よりも高得点を取ることができるようになった。これをコンピューター囲碁に応用したのが AlphaGo [Silver 16] である。2016 年 3 月、世界中にネット中継されて多くの人が注目する中、世界で最も強いプロ棋士の一人であるイ・セドルと対戦して 4 勝 1 敗で勝利したことは、1997 年に Deep Blue がチェスの世界王者であったゲーリー・カスパロフに勝利したこと [Pandolfini 97] とともに人工知能研究の歴史に残る大きな出来事となった。Deep *Q*-Network の大きな特徴が、コンピューター・シミュレーションによる膨大な数の反復学習を用いることによって人間以上の速さで学習することと、行動価値関数を複雑なニューラル・ネットワークで表すことによって多数の状態変数を扱えることである。

本論文では、深層強化学習を複利型に拡張した複利型深層強化学習を提案する。複利型深層強化学習は、報酬の替わりに利益率を観測し、利益率の複利効果を最大化する行動規則を学習する複利型強化学習において、行動価値関数と投資比率をニューラル・ネットワークで表し、深層学習を用いて行動規則の学習と投資比率の最適化を行う。複利型強化学習の行動価値関数と投資比率を複雑なニューラル・ネットワークで表すことによって、状態を多数の状態変数で表すことができ、より複雑な問題に適用できるようになることが期待できる。

本論文では、まず、複利型強化学習と深層強化学習について説明する。そして、深層強化学習を複利型に拡張した複利型強化学習を提案する。また、金融商品（日本国債）の取引戦略の獲得に複利型深層強化学習を用いた例を示し、最後に考察を述べる。

* http://とうごろう.jp, TohgorohMatsui@tohgoroh.jp

2 複利型強化学習

2.1 複利型強化学習の枠組み

強化学習 (RL: reinforcement learning) は、試行錯誤に基づいて適切な行動規則を学習する機械学習 (machine learning) の枠組みである。エージェントは、時刻 t において状態 s_t を観測すると、行動 a_t をして実行し、その結果として報酬 r_{t+1} を得て状態 s_{t+1} に遷移する。これを繰り返しながら、割引収益 (discounted profit)

$$\begin{aligned} &r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \\ &= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \end{aligned} \tag{1}$$

を最大化するような行動規則を学習する。ここで、γ は将来の報酬を割り引く割引率パラメーター (discount rate) を表す。エージェントは γ が 1 に近いほど遠い将来の報酬を考慮し、γ が 0 に近いほど近い将来の報酬しか考慮しないようになる。

複利型強化学習 (compound RL) [松井 13a, 松井 11b, Matsui 12, 松井 11a] は、報酬 r_{t+1} の替わりに利益率 R_{t+1} を観測し、割引複利利益率 (discounted compound return)

$$\begin{aligned} &(1 + R_{t+1}f)(1 + R_{t+2}f)^{\gamma}(1 + R_{t+3}f)^{\gamma^2} \dots \\ &= \prod_{k=0}^{\infty} (1 + R_{t+k+1}f)^{\gamma^k} \end{aligned} \tag{2}$$

の期待値を最大化するような行動規則を学習する。ここで、R_{t+1} は時刻 t の取引の結果として時刻 $t+1$ に求めた利益率、f は投資比率パラメーター (bet fraction) を表す。

割引複利利益率は、対数を取ることで、従来の強化学習と同じように再帰的な形で表すことができる。すなわち、行動規則 π の下での状態 s の価値 $V^{\pi}(s)$ と行動規則 π の下での状態 s における行動 a の価値 $Q^{\pi}(s,a)$ は次のように表される。

$$\begin{aligned} V^{\pi}(s) =& \mathrm{E}_{\pi}\left[\log \prod_{k=0}^{\infty} (1 + R_{t+k+1}f)^{\gamma^k} \middle| s_t = s \right] \\ =& \mathrm{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k \log(1 + R_{t+k+1}f) \middle| s_t = s \right] \\ =& \sum_{a \in \mathcal{A}} \pi(s,a) \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathrm{R}^a_{ss'} + \gamma V^{\pi}(s') \right) \end{aligned} \tag{3}$$

$$\begin{aligned} Q^{\pi}(s,a) =& \mathrm{E}_{\pi}\left[\log \prod_{k=0}^{\infty} (1 + R_{t+k+1}f)^{\gamma^k} \middle| s_t = s, a_t = a \right] \\ =& \mathrm{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k \log(1 + R_{t+k+1}f) \middle| s_t = s, a_t = a \right] \\ =& \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathrm{R}^a_{ss'} + \gamma V^{\pi}(s') \right) \end{aligned} \tag{4}$$

ここで、$\pi(s,a)$ は行動規則 π の下で状態 s において行動 a が選択される確率 (行動選択確率)、$\mathcal{P}^a_{ss'}$ は状態 s において行動 a を行ったときに次の状態が s' になる確率 (状態遷移確率)、$\mathrm{R}^a_{ss'}$ は状態 s において行動 a を行って次の状態が s' になったときに得られるグロス利益率 (利益率に 1 を加えたもの) の対数 $\log(1 + Rf)$ の期待値を表す。複利型強化学習では、すべての s, a に対してこの $Q^{\pi}(s,a)$ を最大化するような行動規則 π を学習する。

2.2 投資比率の最適化

複利型強化学習では、総資産のうち投資する資産の割合を表す投資比率パラメーター (bet fraction) f が導入されており、投資比率によって獲得できる複利利益率が決まる。

宝くじなど、利益率の確率分布が既知の場合には、複利利益率の期待値を最大化する投資比率を解析的に求めることができる。これをケリー基準 [Kelly, Jr. 56] という。ファイナンスの分野では、投資比率を求める方法として Vince が提案したオプティマル f (optimal f) [Vince 90a, Vince 90b] と呼ばれる手法が知られているが、オプティマル f はケリー基準、つまり複利利益率の期待値を最大化する最適な投資比率を求めることができないことを Vince 自身が認めている [Vince 11]。

そこで、複利型強化学習では、最適化手法の一つであるオンライン勾配法 (online gradient method) を用いて、投資比率を最適化する [松井 13b, 羽根田 16]。リターン R_{t+1} を観測すると、投資比率 f を次のように更新する。

$$f(s_t, a_t) \leftarrow f(s_t, a_t) + \eta_t \frac{R_{t+1}}{1 + R_{t+1} f(s_t, a_t)} \tag{5}$$

$$\eta_t = \left(\frac{\eta_0}{\sqrt{t}} \right)^k \tag{6}$$

ここで、$0 \leq \eta_t \leq 1$ は投資比率の学習率パラメーター (learning rate)、η_0 は初期学習率 (initial learning rate)、k は $0 < f_{t+1}(s_t, a_t)$ となる最小の正の整数である。

利益率が $R_{t+1} = -1$ になると、投資比率が $f = 1$ だと $\log(1 + R_{t+1}f) = \log 0 = -\infty$ となり、割引複利利益率の期待値を求めることができない。そこで、ギャン

ブルなど、$R_{t+1} = -1$ となる可能性がゼロでないときは $f < 1$ とする。

その一方で、利益率の絶対値が小さいとき（利益率がゼロに近いとき）は、投資比率を大きくしてもこの問題は生じない。外国為替証拠金取引 (FX) では、レバレッジ (leverage) をかけて預け入れた証拠金の一定倍数まで取引することができる。複利型強化学習においても、このような状況では、レバレッジをかけて取引を行っても問題はなく、また、オンライン勾配法を用いてレバレッジをかけた最大投資可能額に対する最適な投資比率を求めることができる [塚本 16]。

2.3 複利型 Q 学習

従来の Q 学習では、時刻 t での状態 s_t において行動 a_t を取り、その結果として報酬 r_{t+1} を得て状態 s_{t+1} に遷移したとき、以下の式によって Q 値を更新する。

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right) \quad (7)$$

ここで、α は Q の学習率を表すパラメーターである。

複利型 Q 学習では、s_t で a_t を取り、その結果として利益率 R_{t+1} をえて s_{t+1} に遷移したとき、Q 値を次のように更新する。

$$r \leftarrow \log\left(1 + R_{t+1} f(s_t, a_t)\right) \quad (8)$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right) \quad (9)$$

複利型 Q 学習は、Q 学習の報酬 r_{t+1} をグロス利益率の対数 $\log\left(1 + R_{t+1} f(s_t, a_t)\right)$ に置き換え、オンライン勾配法による投資比率の最適化を加えたものである。Algorithm 1 に、複利型 Q 学習のアルゴリズムを示す。

2.4 取引戦略の獲得

株価など金融商品の価格は大きく変動するため、そのまま状態変数として用いると未知の状態に陥りやすい。たとえば、ある株式を対象としているときに、その株価が上場来高値を更新している状態では、株価をそのまま状態変数として用いていると、未だかつて体験したことのない状態であるため、学習が行われていない状況で行動しなければならなくなってしまう。

そこで、状態変数を、直近の値と比較した相対的な値として正規化することによって、状態変数の値が大きく異なる場合でも学習した行動規則を利用できるようにする。具体的には、移動平均 (moving average) および移動標準偏差 (moving standard deviation) の算出期間を k とし、以下のようにして相対化 (relativization) する [Matsui 09]。

$$x_{i,t} = \frac{v_{i,t} - \mu_{t,k}}{4\sigma_{t,k}} \quad (10)$$

ここで、$v_{i,t}$ は時刻 t における i 番目の状態変数の値、$\mu_{t,k}$ は時刻 t の直近 k 個のデータから求めた移動平均、$\sigma_{t,k}$ は同じく移動標準偏差を表す。

これまでの研究では、終値を相対化した相対終値 (RCP: relative closing price) と移動標準偏差を相対化した相対移動標準偏差 (RMSD: relative moving standard deviation) を状態変数として用いている。株式を対象とした取引の場合、RCP が正のときは現在の株価が移動平均株価より大きい、すなわち、株価が上昇していることを表している。RMSD が正のときは現在の標準偏差が移動平均標準偏差より大きい、すなわち、株価の変動が大きくなっていることを表している。

エージェントの行動は買い (buy) と売り (sell) の 2 種類である。金融商品を購入している状態をロング・ポジション、金融商品を信用売りしている状態をショート・ポジションという。エージェントは、複利型強化学習によって学習された取引戦略によって行動を選択し、オンライン勾配法によって学習された投資比率 f によってポジションの大きさを調整する。

なお、本研究では、金融市場に関する状態変数はエージェントの行動に依存しないと仮定している。たとえば、株取引の場合には、エージェントがどちらの行動を選んでも、株価には影響しない。

3 深層強化学習

深層強化学習 (deep RL) [Mnih 13, Silver 16] は、強化学習における価値関数 (value function) Q をニューラル・ネットワーク (artificial neural network) で表し、深層学習 (deep learning) を用いて Q を学習するものである。

図 1 に、Deep Q-Network のニューラル・ネットワークを示す。状態 s が m 次元の状態変数ベクトル $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ で表されるとき、入力層には $m+1$ 個のユニットが配置され、x_1 から x_m までの状態変数と行動 a が入力される。出力層にはユニットが一つだけ配置され、状態 s における行動 a の価値 $Q(\mathbf{x}, a)$ が出力される。

Deep Q-Network の学習アルゴリズムを Algorithm 2 に示す。従来の Q 学習では 1 ステップごとに

Algorithm 1 複利型 Q 学習アルゴリズム

入力: 割引率 γ, 行動価値学習率 α, 初期投資比率 f_0, 初期投資比率学習率 η_0
for all s, a **do**
  $Q(s,a)$ を任意に初期化
  $f(s,a) \leftarrow f_0$
end for
loop (各エピソードに対して繰り返し)
  s を初期化
  repeat (エピソードの各ステップに対して繰り返し)
    Q から導かれる行動規則（行動選択確率）に従って s での行動 a を選択
    行動 a を実行し、利益率 R と次の状態 s' を観測
    $Q(s,a) \leftarrow Q(s,a) + \alpha \left(\log(1 + Rf(s,a)) + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$
    $\eta \leftarrow 1, f \leftarrow f(s,a), \Delta f \leftarrow \dfrac{R}{1+Rf}$
    repeat
      $\eta \leftarrow \dfrac{\eta_0}{\sqrt{t}}\eta, f' \leftarrow f + \eta \Delta f$
    until $f' > 0$
    $f(s,a) \leftarrow f', s \leftarrow s'$
  until s が終端状態ならば繰り返しを終了
end loop

Algorithm 2 Deep Q-Network 学習アルゴリズム

入力: 割引率 γ, 行動価値学習率 α
Q を表すニューラル・ネットワークを任意に初期化
for $i = 1$ to N **do**
  Q から導かれる行動規則に従ってしばらくの間行動し、状態変数ベクトル $\mathbf{x}$, 行動 a, 報酬 r, 次の状態の状態変数ベクトル $\mathbf{x}'$ の組を収集する
  収集した $\langle \mathbf{x}, a, r, \mathbf{x}' \rangle$ の集合からランダム・サンプリングによって M 個を取り出す
  for $j = 1$ to M **do**
    $q_j \leftarrow Q(\mathbf{x}_j, a_j) + \alpha \left(r_j + \gamma \max_{a'} Q(\mathbf{x}'_j, a') - Q(\mathbf{x}_j, a_j) \right)$
  end for
  $\mathbf{x}_j$ と a_j を入力、q_j を出力とした M 個（$j = 1, \ldots, M$）の訓練データを用いて Q を表すニューラル・ネットワークを更新する
end for

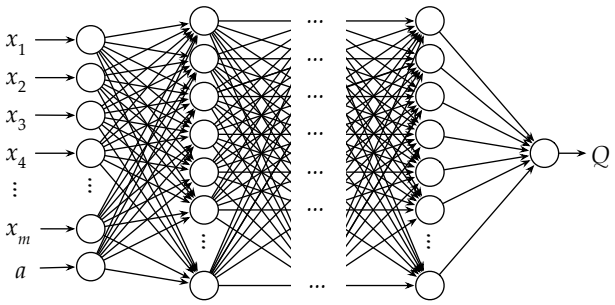

図 1　Deep Q-Network

Q 値を更新して学習するが、Deep Q-Network では、1 ステップごとの学習は行わず、M ステップ分の更新をまとめて行う。Deep Q-Network を固定して Q から導かれる行動規則に従って M ステップ以上行動し、状態 s_t を表す状態変数ベクトル $\mathbf{x}_t$, 行動 a_t, 報酬 r_{t+1}, 次の状態 s_{t+1} の状態変数ベクトル $\mathbf{x}_{t+1}$ の組を収集する。収集した $\langle \mathbf{x}_t, a_t, r_{t+1}, \mathbf{x}_{t+1} \rangle$ の集合から、ランダム・サンプリングによって M 個を取り出す。取り出した M 個のデータのそれぞれに対し、以下の式によって $\mathbf{x}_t$ と a_t を Deep Q-Network へ入力したときの望ましい出力値 q_t を求める。

$$q_t \leftarrow Q(\mathbf{x}_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_{a'} Q(\mathbf{x}_{t+1}, a') - Q(\mathbf{x}_t, a_t) \right) \quad (11)$$

これらの M 個のデータを訓練データとして Deep Q-Network を更新する。これを N 回繰り返す。

Deep Q-Network のランダム・サンプリングは、深層

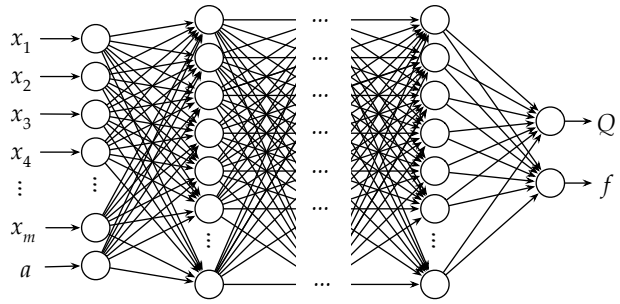

図 2　複利型 Deep Q-Network

学習で大きな問題となる過学習 (overfitting) への対策となっている。すべてのステップで学習せずに、ランダムに選択されたステップのみで学習することによって、特定の状況へ過度にフィットしないようになっている。これは、アンサンブル学習 (ensamble learning) のバギング (bagging) が過学習対策として用いられるのと同じ仕組みである。

4　複利型深層強化学習

4.1　仕組みとアルゴリズム

複利型深層強化学習 (compound deep RL) は、深層強化学習を複利型に拡張したものである。図 2 に、Deep Q-Network を複利型に拡張した複利型 Deep Q-Network のニューラル・ネットワークを示す。複利型 Deep Q-Network では、行動価値 Q に加えて、投資比率 f を出力する。

複利型 Deep Q-Network の学習アルゴリズムは、Deep Q-Network 学習アルゴリズムを複利型に拡張したものである。複利型 Deep Q-Network の学習アルゴリズムを Algorithm 3 に示す。

複利型強化学習は、従来の強化学習における報酬 r の替わりに利益率 R を受け取り、更新式の報酬 r をグロス利益率の対数 $\log(1+Rf)$ に置き換えたものである。複利型深層強化学習も、同様に、深層強化学習における報酬 r の替わりに利益率 R を受け取る。複利型 Deep Q-Network では、$\langle \mathbf{x}_t, a_t, R_{t+1}, \mathbf{x}_{t+1} \rangle$ を収集し、そこからランダム・サンプリングによって Deep Q-Network を更新するための訓練データを作成する。このとき、$\mathbf{x}_t$ と a_t を Deep Q-Network へ入力したときの望ましい出力値 q_t は、以下のように求める。

$$r \leftarrow \log(1 + R_{t+1} f) \tag{12}$$

$$q_t \leftarrow Q(\mathbf{x}_t, a_t) + \alpha \left(r + \gamma \max_{a'} Q(\mathbf{x}_{t+1}, a') - Q(\mathbf{x}_t, a_t) \right) \tag{13}$$

複利型強化学習では、投資比率 f が導入されている。複利型深層強化学習では、ニューラル・ネットワークの出力層に投資比率 f を加え、行動価値 Q とともに出力する。$\mathbf{x}_t$ と a_t を複利型 Deep Q-Network へ入力したのときの望ましい出力値 f_t は、以下のように求める。

$$f_t \leftarrow f(\mathbf{x}_t, a_t) + \eta_t \frac{R_{t+1}}{1 + R_{t+1} f(\mathbf{x}_t, a_t)} \tag{14}$$

$$\eta_t = \left(\frac{\eta_0}{\sqrt{iM}} \right)^k \tag{15}$$

ここで、$0 \leq \eta_t \leq 1$ は投資比率の学習率パラメーター (learning rate)、η_0 は初期学習率 (initial learning rate)、i は繰り返し回数、M はランダム・サンプリングのサンプル数、k は $f_t > 0$ となる最小の正の整数である。

4.2　実装

本研究では、深層学習ライブラリーの Deeplearning4j (DL4J) に、複利型強化学習のモジュールを作成して組み込むことで、複利型 Deep Q-Network のプログラムを実装した。

本論文における金融商品取引では、状態変数の値を過去の価格データだけから求められるため、任意の時点から行動を始めることができる。そこで、しばらくの間行動して $\langle \mathbf{x}_t, a_t, R_{t+1}, \mathbf{x}_{t+1} \rangle$ を収集し、そこからランダム・サンプリングを行う代わりに、取引時刻をランダム・サンプリングによって選択し、取引を行うことによって $\langle \mathbf{x}_t, a_t, R_{t+1}, \mathbf{x}_{t+1} \rangle$ を収集する。

そこで、DL4J に以下の 5 つのモジュールを追加した。

1. ランダム・サンプリング
2. Q 値推定
3. 行動選択
4. 利益率計算
5. 更新後 Q 値計算

まず、(1) ランダム・サンプリングによって、全期間から M 個の取引時刻を選択する。選択したデータ $\langle \mathbf{x}_j, a_j, R_j, \mathbf{x}'_j \rangle$ のそれぞれに対して、(2) 行動（買いまたは売り）ごとに、学習中のニューラル・ネットワークに状態変数 $\mathbf{x}_j$ と行動を入力して Q 値と投資比率 f を推定し、(3) 推定した Q 値に基づいて、ϵ-グリーディー選択を用いて行動 a_j を選択、(4) a_j を取ったときの利益率 R_j を求め、(5) Q 値推定モジュールを用いて次の状態での行動価値を推定し、式 (12)–(15) を用いて Q 値と投資比

Algorithm 3 複利型 Deep Q-Network 学習アルゴリズム

入力: 割引率 γ, 強化学習率 α, 投資比率学習率 η_0
Q と f を表すニューラル・ネットワークを任意に初期化
for $i = 1$ to N **do**
  Q から導かれる行動規則に従ってしばらくの間行動し、状態変数ベクトル $\mathbf{x}$, 行動 a, 利益率 R, 次の状態の状態変数ベクトル $\mathbf{x}'$ の組を収集する
  収集した $\langle \mathbf{x}, a, R, \mathbf{x}' \rangle$ の集合からランダム・サンプリングによって M 個を取り出す
  for $j = 1$ to M **do**
    $$q_j \leftarrow Q(\mathbf{x}_j, a_j) + \alpha \left(\log(1 + R_j f) + \gamma \max_{a'} Q(\mathbf{x}'_j, a') - Q(\mathbf{x}_j, a_j) \right)$$
    $$\eta \leftarrow 1, f \leftarrow f(\mathbf{x}_j, a_j), \Delta f = \frac{R_j}{1 + R_j f}$$
    repeat
      $$\eta \leftarrow \frac{\eta_0}{\sqrt{iM}} \eta, f' \leftarrow f + \eta \Delta f$$
    until $f' > 0$
    $f_j \leftarrow f'$
  end for
  $\mathbf{x}_j$ と a_j を入力、q_j と f_j を出力とした M 個（$j = 1, \ldots, M$）の訓練データを用いて Q と f を表すニューラル・ネットワークを更新する
end for

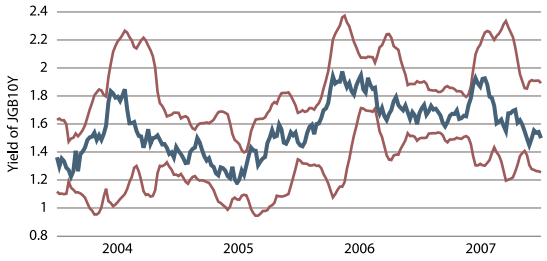

図 3　残存期間 10 年日本国債金利と $\pm 4\sigma$ のボリンジャー・バンド

率の望ましい値 q_j, f_j を求める。最後に、DL4J の機能を用いて、M 個の訓練データを与えてニューラル・ネットワークを更新する。これを N 回繰り返す。

5　実験結果

提案手法の有効性を確認するために、残存期間 10 年の日本国債の週次取引を対象として実験を行った。取引の期間は、2004 年から 2007 年の 4 年間とした。図 3 に、対象期間の金利の推移と $\pm 4\sigma$ のボリンジャー・バンドを示す。この期間には計 208 週あり、利益率を計算するため、最後の週を除いた 207 週を対象とした。移動平均と移動標準偏差の算出期間は、従来研究と同じ 14 週とした。

複利型 Deep Q-Network と従来の Deep Q-Network で用いるニューラル・ネットワークは、中間層を 1 つ、ユニット数は 20 とした。重みの初期値は Xavier を用いて確率的に決定し、入力層から中間層への活性化関数はランプ関数 (ReLU)、中間層から出力層への活性化関数は線形結合（単純パーセプトロン）とした。

強化学習の行動選択には $\epsilon = 0.2$ の ϵ-グリーディー選択を用い、行動価値の学習率は $\alpha = 0.2$、割引率は $\gamma = 0$ とした。ランダム・サンプリングのサンプル数は $M = 100$ とし、繰り返し回数を $N = 10,000$ とした。また、今回の実験では、複利型 Deep Q-Network の投資比率 f の最適化は行わず、$f = 1$ に固定した。

繰り返し 1 回ごとに、すなわち、通常の強化学習 100 ステップ分の学習を行うごとに、学習した Q を用いて全期間で取引を行い、その年平均利益率を求め、繰り返し 100 回ごとに年平均利益率の平均を求めた。結果を図 4 に示す。最後の 100 回の年平均利益率の平均値は複利型 Deep Q-Network が 2.12%、Deep Q-Network が 2.08% であった。t 検定によって平均の差を検定したところ、差の平均値は 0.365、標準偏差は 0.371 であり、差の 95% 信頼区間の下限が 0.0291、上限が 0.0439 であることから、有意水準 5% で優位な差があることが確認された。また、自由度 99 で t 統計量が 9.83 であったことから、有意水準 0.1% でも有意な差があった。

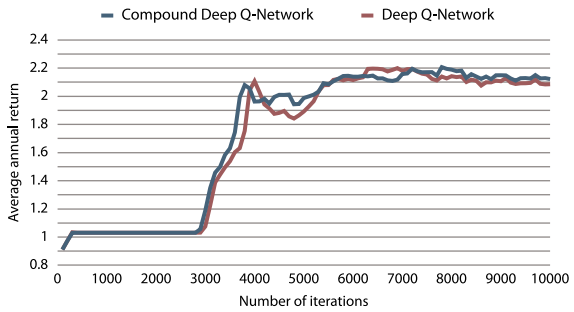

図 4　実験結果

6　考察とまとめ

本論文では、深層強化学習を複利型に拡張した複利型深層強化学習を提案した。深層強化学習は、囲碁に応用した AlphaGo が世界で最も強い棋士の一人に勝利するなど、従来は複雑で難しいと考えられていた問題にも適用可能であることが示されており、これを複利型に拡張した複利型深層強化学習は、複雑な環境でも複利利益率を最大化することが期待できる。本論文では、深層学習ライブラリーの Deeplearning4j (DL4J) に複利型強化学習のモジュールを作成して追加することで、複利型 Deep Q-Network を実装した。また、日本国債の週次取引を対象とした実験により、複利型深層強化学習が取引戦略を学習できることを確認した。

今後の課題はまだ多く残っている。まず、深層強化学習の特徴の一つは、複雑なニューラル・ネットワークを用いて Q 関数を表すことによって、多数の状態変数を扱うことができることである。本論文では、従来手法と同じ、相対終値と相対移動標準偏差の二つしか状態変数を用いていない。多数の状態変数を用いることによってより優れた取引戦略の獲得ができるか、検証が必要である。

また、本論文の実験では、投資比率を $f = 1$ にして実験を行った。本論文で提案した投資比率 f をニューラル・ネットワークで表して投資比率を最適化することについての有効性はまだ確認されていない。これについても、検証が必要である。

今後、複数の金融商品を対象とした実験を行い、これらの有効性を検証していきたい。

参考文献

[Kelly, Jr. 56] Kelly, Jr., J. L.: A new interpretation of information rate, *Bell System Technical Journal*, 35:917–926 (1956)

[Matsui 09] Matsui, T., Goto, T., and Izumi, K.: Acquiring a government bond trading strategy using reinforcement learning, *JACIII*, 13(6):691–696 (2009)

[Matsui 12] Matsui, T., Goto, T., *et al.*: Compound Reinforcement Learning: Theory and An Application to Finance, in *European Workshop on Reinforcement Learning 9 (EWRL 2011)*, *LNCS* 7188:321–332 (2012)

[Mnih 13] Mnih, V., Kavukcuoglu, K., *et al.*: Playing Atari with Deep Reinforcement Learning, in *NIPS Deep Learning Workshop* (2013)

[Pandolfini 97] Pandolfini, B.: *KASPAROV AND DEEP BLUE*, Simon & Schuster (1997), 鈴木知道 訳, ディープブルー vs. カスパロフ, 河出書房新社 (1998)

[Silver 16] Silver, D., Huang, A., *et al.*: Mastering the game of Go with deep neural networks and tree search, *Nature*, 529:484–489 (2016)

[Vince 90a] Vince, R.: Find your optimal f, *Technical Analysis of Stock & Commodities*, 8(12):476–477 (1990)

[Vince 90b] Vince, R.: *Portfolio management formulas: Mathematical trading methods for the futures, options, and stock markets*, Wiley (1990), 長尾 慎太郎 訳, 投資家のためのマネーマネジメント—資産を最大限に増やすオプティマル f, パン・ローリング (2005)

[Vince 11] Vince, R.: Optimal f and the Kelly Criterion, *IFTA Journal*, 21–28 (2011)

[塚本 16] 塚本 智大, 松井 藤五郎：レバレッジを用いた複利型強化学習, 第 78 回情報処理学会全国大会 (IPSJ 78), 3P3-05 (2016)

[羽根田 16] 羽根田 卓哉, 松井 藤五郎：複利型強化学習における投資比率学習法の改善, 第 78 回情報処理学会全国大会 (IPSJ 78), 3P3-06 (2016)

[松井 11a] 松井 藤五郎：複利型強化学習, 人工知能学会論文誌, 26(2):330–334 (2011)

[松井 11b] 松井 藤五郎, 後藤 卓, 和泉 潔, 陳 ユ：複利型強化学習の枠組みと応用, 情報処理学会論文誌, 52(12):3300–3308 (2011)

[松井 13a] 松井 藤五郎：複利型強化学習—強化学習のファイナンスへの応用—, 計測と制御, 52(11):1022–1027 (2013)

[松井 13b] 松井 藤五郎, 後藤 卓, 和泉 潔, 陳 ユ：複利型強化学習における投資比率の最適化, 人工知能学会論文誌, 28(3):267–272 (2013)