情報処理学会論文誌　データベース　Vol.14　No.4　61–69 (Oct. 2021)

# 分布型強化学習を用いたポートフォリオマネジメントにおける低リスク投資行動の学習

佐藤 葉介[1]　張 建偉[1,a)]

受付日 2021年3月10日, 採録日 2021年7月6日

**概要**：近年，金融市場における投資行動を深層学習により獲得する研究がさかんである．金融市場は景気や政局など多くの複雑な要因により変動するため，確実な取引戦略の構築が困難である．一方，分布型強化学習（DRL）は強化学習における行動価値関数を離散分布に拡張した手法で，とりうる行動により期待される Q 値を分布で表すことで単一の Q 値よりも高い表現力を持つ．本研究では，ポートフォリオマネジメントにおいて保有する資産価値が低下するリスクを防ぎつつ利益を最大化させるような投資行動を DRL を用いて学習する手法を提案する．10 年分の日経 225 に含まれる銘柄のヒストリカルデータを用いて実験を行い，DRL を用いた提案手法の方が比較手法の DQN より評価値の標準偏差について優れていたため，低リスクな投資行動を学習できたといえる．

**キーワード**：時系列データ，分布型強化学習，低リスク投資行動，ポートフォリオマネジメント

# Learning Low-risk Investment Actions Using Distributional Reinforcement Learning for Portfolio Management Problem

Yosuke Sato[1]　Jianwei Zhang[1,a)]

Received: March 10, 2021, Accepted: July 6, 2021

***Abstract:*** In recent years, investment strategies on the financial market using deep learning have attracted a significant amount of research attention. Since the financial market is influenced by complex factors (e.g., economy, politics), it is difficult to construct a certain investment strategy. On the other hand, Distributional Reinforcement Learning (DRL) expands the action-value function to a discrete distribution in reinforcement learning, which expresses expected Q values for all actions as a distribution and thus has higher representation power than single Q values. In this study, we focus on the portfolio management problem and apply DRL to construct an investment trading model that is low-risk and maximizes profit. This model has been backtested on Nikkei 225 dataset over ten years and compared with Deep Q Network (DQN). The experimental results show that the proposed DRL-based method can learn low-risk actions outperforming the compared DQN-based method in terms of the standard deviations of evaluation values.

***Keywords:*** time-series data, distributional reinforcement learning, low-risk investment action, portfolio management

## 1. はじめに

近年，金融市場を対象として深層学習を用いた研究がさかんである [1], [2], [3]．金融市場は景気や為替などの経済的要因や，政局などの経済外的要因などが関わり変動するため，確実な状態予測や取引戦略の構築が困難である．そのため，投資行動学習の研究がされている [4], [5], [6], [7]．近年は高い特徴表現力を持つ深層モデルを用いた取引モデルが研究されている [3], [8], [9]．ほとんどの研究では利益を増加させつつ保有資産価値が減少するリスクへ対処する 2 つの課題に対して様々な手法を提案している [10], [11], [12], [13]．深層強化学習を用いた多くの先行研究では Deep Q Network（DQN）[14] が提案手法のベースや比較手法として用いら

[1] 岩手大学
Iwate University, Morioka, Iwate 020–8551, Japan
[a)] zhang@iwate-u.ac.jp

れている．これらの研究で用いられる評価値は，利益の増大を測るため，テスト期間に得られた資産の多さを利益率で表し，その平均や標準偏差を利益の多さやどれだけ安定して利益を得られるかを調べるために利用している．また，これら 2 つの評価値を統合した，とったリスクに対するリターンの大きさを示すシャープレシオ [15] も用いられる．

一方，分布型強化学習（Distributional Reinforcement Learning，DRL）[16] は Q 学習における行動価値関数の各行動の評価値を分布に拡張した手法であり，ベンチマークにおいて DQN，Double DQN（DDQN），Dueling Network より優れた結果を残している．DRL における行動価値関数は，ある行動で得られる報酬の期待値だけでなく，定義した報酬の値の範囲で各報酬が得られる期待値を離散分布で学習することができる．単一の期待リターンのみを各行動に対して保持する DQN に比べて，DRL は期待リターンを分布で保持して行動決定のための計算に利用する．将来の予測が困難な投資行動においては期待リターン分布が有用な情報となる可能性があり，DRL を用いることで安定化や低リスク化について改善したという先行研究も存在する [17]．また，モデル出力を人が観察しリスク操作をすることが可能な利点もあり，行動価値関数の出力と手動で設定した歪度の要素積を計算することで，学習の対象ごとの性質に合わせた行動の設計ができる手法も提案されている [18]．

著者らの先行研究では，日経 225 を構成するそれぞれの銘柄に対して投資行動を学習した [19]．DQN と DRL の 2 種類の手法を用いて実験を行った結果，多くの評価項目で DRL が DQN に比べて優位となった．現実の投資に応用するためには，同時に複数の金融資産を対象として学習を行うことで，より良い投資行動を学習することが必要であると考える．本研究では複数の金融資産に対して同時に投資行動を学習し，かつ DRL を利用することで利益を最大化しつつ低リスクな投資行動を獲得することを目的とする．

想定するユースケースは，日本円などの無リスク資産と複数種類の金融資産の売買における投資行動の選択または推薦である．また，学習に十分な過去のデータや，データを入手可能な環境がある必要がある．本手法は学習には多くの時間を費やすものの，少ない時間で状況に対する投資行動の選択ができる．そのため，デイトレードなどの短期的な投資に対してあらかじめモデルを学習しておき，投資行動を得るような応用が可能である．

日経 225 を構成するすべての銘柄に対してどのような投資行動を行うかという，ポートフォリオマネジメント問題に対して DRL を用いて行動を学習し実験を行った．バックテストを行い，得られた評価値を DQN と比較した．実験結果として，特に評価値の標準偏差について DQN よりも DRL を用いた提案手法の方が優位な結果を得られた．評価値の標準偏差が小さいほど安定した投資行動を期待できると考えられ，DQN に比べて低リスクな行動を学習できているといえる．実験結果に対して両側検定を行ったところ，投資行動のリスクに関する評価値であるシャープレシオの結果について有意差が得られた．

## 2. 関連研究

### 2.1 Deep Q Network

Q 学習は強化学習手法の 1 つであり，行動と状態に対する Q 値を出力する行動価値関数を学習する．エージェントの方策によって Q 値を利用し，次の行動を判断する．Q 学習では状態と行動に対する Q 値を保存する Q テーブルを用いるが，問題によってはすべての状態と行動の組合せを学習や保存することは困難となる．Deep Q Network（DQN）[14] は畳み込みニューラルネットワークによる深層モデルを用いて Q テーブルを近似した手法であり，Atari というゲーム AI を用いたベンチマークでは人のスコアを超えた手法である．ほかにもいくつかの学習手法を取り入れている．experience replay は一定頻度でバッチ学習を行い，ステップごとに学習する場合に比べて学習の効率を高めた．reward clipping は報酬値を閾値で上限と下限を設けることで学習を安定させた．また，損失関数に Huber 関数を用いていることで二乗誤差に比べて誤差が大きいときに過大な値にならないようにすることで，学習を安定させた．DQN は多くの研究で応用や発展がされており，深層強化学習の手法として最も知られたものであると考えられる．本研究で利用する DRL も DQN を拡張した手法である．

### 2.2 金融分野における強化学習

ポートフォリオマネジメントは利益の最大化やリスクの最小化が課題であり，強化学習による適切な金融資産の分配を行う投資行動の学習が試みられている．Almahdi らは Recurrent Neural Network（RNN）を利用した投資行動を学習する強化学習手法を開発した [20]．学習段階でシャープレシオとカルマーレシオを目的関数に用いて risk-return portfolio を最適化する低リスクな手法を提案した．価値ベースの強化学習を用いた手法として Shin らは DQN を用いて 8 種類の暗号通貨と USD のポートフォリオマネジメントを学習させており，最も資産価値の減少を抑えた行動を学習できている [10]．方策ベースの強化学習手法として Xiong らは Deep Deterministic Policy Gradient（DDPG）を用いて株価，保有株数，残高の状態からそれぞれの株に対する購入，売却，維持の行動を学習している．約 10 年分の Dow Jones 30 stocks についてバックテストを行い比較手法より利益とシャープレシオについて優位な結果を残した [11]．また，Ye らも DDPG を用いており，ニュース記事と株価の推移を前処理したデータからポートフォリオの割当てを学習している．ベンチマークにおいては比較手法より優位な利益を出し，シャープレシオもおおむね優位

な結果を残した [13]．CNN ベースの DDPG を構築し 12 種類の暗号通貨のポートフォリオマネジメントを行った研究も存在する [21]．Jiang らは暗号通貨のポートフォリオマネジメントを行う EIIE フレームワークを開発し UBAH などの比較手法より portfolio value やシャープレシオについて優位な結果を得ている．EIIE フレームワークは方策ベースの手法であり，資産の潜在的な成長性から直近の予測を行う IIE のアンサンブル学習である [22]．

本研究に最も近い手法として，Harnpadungkij らは DRL を投資行動の学習に初めて応用している [17]．本手法と同様に低リスクな投資行動を学習できており，金融分野における DRL の有用性を示した．しかし，対象とする金融資産はリスク指標であるベータ値をもとに選出した 10 種類の株式ペアであり，それぞれのペアについて投資行動を学習している．現実には多くの金融資産を視野に入れて投資行動を決定するため，実験内容が現実の問題に合っているとはいえず，提案されたモデルも複数の金融資産に対して同時に投資行動を決定できるものではない．本研究は，先行研究と同様に低リスクな投資行動を学習することを目的としているが，複数の金融資産に対して同時に投資行動を学習する点で異なる．

## 3. ポートフォリオマネジメントにおける DRL

本手法では価値が変動する複数の金融資産と日本円などの無リスク資産を対象としたポートフォリオマネジメントを行う．ここでの投資行動の対象資産は東京証券取引所第一部に上場しているうち，日経 225 に含まれる企業の株式と日本円による無リスク資産とする．また，本手法では日足データを元に取引行動を行う．日足には始値（open price），高値（high price），低値（low price），終値（close price）の 4 つの変数が含まれ，エージェントが購入または売却するときは終値を用いる．

本研究では金融市場における投資行動をマルコフ決定過程 $M(\mathcal{S}, \mathcal{A}, R, P)$ とする．$\mathcal{A}$ は行動空間を表す．行動 $a \in \mathcal{A}$ は各銘柄に対し，保持している無リスク資産で株式を “購入”，保持している株式の “売却”，資産の売買を行わない “維持” の 3 つをとりうる．$\boldsymbol{a}$ は 225 銘柄それぞれに対する行動の集合を表す．

$$\boldsymbol{a} = (a_1, a_2, \ldots, a_{225}) \tag{1}$$

$\mathcal{S}$ は状態空間を表す．状態 $s \in \mathcal{S}$ には整形した日足データが含まれる．あるステップ $t$ における $n$ 日分の時系列データは以下のようになる．

$$v_t^{diff} = v_t - v_{t-1} \tag{2}$$

$$\boldsymbol{v_t} = (f(v_t^{open\ \ diff}), f(v_t^{high\ \ diff}), f(v_t^{low\ \ diff}), f(v_t^{close\ \ diff})) \tag{3}$$

$$\boldsymbol{s_t} = (\boldsymbol{v_t}, \boldsymbol{v_{t-1}}, \ldots, \boldsymbol{v_{t-n+1}}) \tag{4}$$

$v_t^{diff}$ はステップ $t$ における前ステップとの差分を表す．$\boldsymbol{v_t}$ は差分に対して $f$ による正規化を行った値をまとめたベクトルである．$\boldsymbol{s_t}$ はステップ $t$ における環境から観測される状態とする．次状態 $\boldsymbol{s_{t+1}}$ はステップ $t+1$ において同様に計算したものである．DQN と同様に複数ステップの情報をまとめて 1 つの状態とし，225 銘柄分ベクトル化したものを $s$ とする．

$R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ は報酬関数を表しており時刻 $t$ において得られた報酬を $r_t$ とする．報酬値には，初期状態または株式の購入時から売却時までの資産額の増減を利用している．売却時のみ即時報酬を与えると過程が評価されないため，報酬取得時に売却時までの各状態 $s$ に対し遅延報酬を与える．各報酬値に対して DQN と同様に reward clipping を適用する．

遷移関数 $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ は状態 $s$ のとき行動 $\boldsymbol{a}$ によって状態 $s'$ へ遷移する状態遷移確率 $p(s, a, s')$ を表す．方策 $\pi(\boldsymbol{a} \mid s)$ は状態 $s$ のときの行動 $\boldsymbol{a}$ をとる確率を表す関数である．行動価値関数 $Q^\pi(s, \boldsymbol{a})$ は状態 $s$ のとき方策 $\pi$ に従い行動 $\boldsymbol{a}$ をとったときに得られる期待報酬値を定義する．

$$Q^\pi(s, \boldsymbol{a}) = \mathbf{E}[R(s, \boldsymbol{a})] + \gamma \mathbf{E}_{P,s}[Q^\pi(s', \boldsymbol{a}')] \tag{5}$$

最適方策 $\pi^*$ を学習し最適行動価値関数 $\mathbf{E}[Q^*(s, \boldsymbol{a})]$ の戻り値を最大化するような行動を学習することが目的となる．最適方策とは任意の初期状態 $s \in \mathcal{S}$ から期待報酬を最大化する方策である．DRL において利用する Q 学習では以下の更新式により最適行動価値関数を学習する．

$$Q_\theta(s, \boldsymbol{a}) \leftarrow \mathbf{E}[R(s, \boldsymbol{a})] + \gamma \mathbf{E}_P[\underset{\boldsymbol{a}'}{max} Q_\theta(s', \boldsymbol{a}')] \tag{6}$$

森村らはベルマン期待方程式のリターンを分布に拡張した分布ベルマン方程式を定義している [23]．分布ベルマン方程式を解くことでリターン分布を推定できるが分布ベルマン方程式は汎関数の自由度を持つため一般に推定は困難であるため近似が必要となる．

Bellemare ら [16] はリターン分布を多項分布で近似した DRL を提案している．リターン分布は，直感的には bin という 1 つの報酬値が得られる期待値を表すものが複数個連続している．近似リターン分布の bin 数 $M \geq 2$ と，近似リターン分布の上限 $Q_{max}$ と下限 $Q_{min}$ をハイパーパラメータとして定め，bin 間隔 $\Delta_z$ を

$$\Delta_z := \frac{Q_{max} - Q_{min}}{M - 1} \tag{7}$$

とし，各 bin に対するリターン代表値を $z_m, m \in \{1, \ldots, M\}$

$$z_m := Q_{min} + (m - 1)\Delta_z \tag{8}$$

とする．状態 $s$ と行動 $a$ の入力に対するリターン分布を表現する $M$ 次元ベクトル $(q_1(s, a), \ldots, q_M(s, a))$ を出力する

深層モデル $\mathcal{S} \times \mathcal{A} \to \mathbb{R}^M$ を用いて，推定リターン分布 $\hat{P}$ を

$$\hat{P}(C = z_m|s,a) := \frac{\exp(q_m(s,a))}{\sum_{m'=1}^{M} \exp(q_{m'}(s,a))}, \quad \forall m \in \{1,\ldots,M\} \quad (9)$$

として求める．このとき DRL の行動価値の推定値 $\hat{Q}$ は

$$\hat{Q}(s,a) \triangleq \sum_{m=1}^{M} z_m \hat{P}(C = z_m|s,a) \quad (10)$$

となる．$\hat{Q}$ は DQN と同様に近似分布ベルマン行動最適作用素 $\hat{D}$ [16] を適用して現在の推定リターン分布 $\hat{P}$ から目的分布 $\hat{P}_n^{target}$ を求める．experience replay [14] により得た経験 $n$ を用いて目的分布 $\hat{P}_n^{target}$ と現在の推定分布 $\hat{P}(\cdot|s_n,a_n)$ との差異が小さくなるように深層モデルの重みを更新する．

experience replay では，学習において 1 ステップ更新されるごとに，replay memory に現在の状態，次の状態，評価値を保存する．$n$ ステップに 1 度，ハイパーパラメータとして指定したバッチサイズだけ replay memory からデータを取り出し Target-Network の重みを学習する．

## 4. 投資行動学習とエージェントの行動アルゴリズム

本手法では日経 225 に含まれる 225 銘柄に対するポートフォリオマネジメントを行う．使用する深層モデルを**図 1** に示す．入力次元数は状態 $s$ として $n$ 日分の 225 銘柄の日足データを用いるため $n \times 225 \times 4$ となる．出力次元数は DRL における離散分布を構成する bin 数 $M$ として 3 種類の行動を 225 種類の銘柄ごとに学習するため $M \times 3 \times 225$ となる．提案手法は一般的な強化学習を踏襲しており，アルゴリズムを Algorithm 1 に示す．1 行目から 7 行目にかけて変数の初期化を行う．$T$ はタイムステップであり，1 ステップで 1 日だけ時間を進める．*done* はエポックの終了判定に，*episodes* はエピソード数のカウントに使用する．*stack_memory* は遅延報酬を与えるまでデータを保存するスタックである．*env*，*agent* はそれぞれ環境とエージェントである．*mem* は replay memory に利用する．9 行目から 13 行目にかけて初期化処理を行っている．1 エポックが終了したときやプログラム開始時に初期化する．14 行目から 16 行目にかけてはエージェントの行動と環境の相互作用が働き，報酬の獲得が行われる．17 行目から 25 行目にかけて 3 章に従い Algorithm 1 のように遅延報酬をエージェントに与える．26 行目に示すように 1 エポックほどデータ取得期間を得てから，27 行目から 29 行目で，指定した頻度でエージェントの学習を行う．31 行目に示すようにエポックの終了ごとにモデルの評価を実施する．34，35 行目で次のステップへの変数の更新を行う．

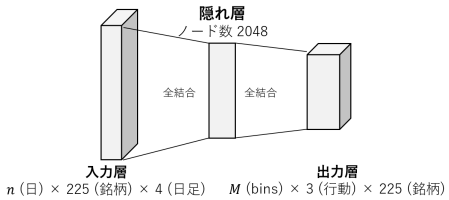



**図 1**　実験で使用するモデル形状
**Fig. 1**　Model for experiments.

提案手法におけるデータフローを**図 2** に示す．ステップごとに得られる時系列の株価差分データやエージェントの行動などをタプルに整形してスタックし保存する．エージェントが売却の行動を行ったとき，遅延報酬をスタックに適用して replay memory に保存する．モデルは replay frequency として設定した頻度で学習を行う．行動選択時にはモデルを用いることで分布行動価値が行動ごとに出力され，3 章に従い行動を決定する．

エージェントの行動のアルゴリズムを Algorithm 2 に示す．DRL の出力を元にそれぞれの銘柄に対して行動を行

**Algorithm 1** 投資行動学習アルゴリズム

```
 1: T ⇐ 0
 2: done ⇐ True
 3: episodes ⇐ 0
 4: stack_memory ⇐ [ ]
 5: env ⇐ InitializeEnvironment()
 6: agent ⇐ InitializeAgent()
 7: mem ⇐ InitializeReplayMemory()
 8: while episodes < MAX_EPISODES do
 9:     if done is True then
10:         state ⇐ env.reset()
11:         done ⇐ False
12:         episodes ⇐ episodes + 1
13:     end if
14:     action ⇐ agent.act_epsilon_greedy(state)
15:     next_state, reward, done ⇐ env.step(action)
16:     reward ⇐ reward_clipping(reward)
17:     stack_memory.append([state, action, reward, done])
18:     if action is sell then
19:         for i = 0 to len(stack_memory) do
20:             Apply discount reward to each stacked data
21:         end for
22:         for item in stack_memory do
23:             mem.append(item)
24:         end for
25:     end if
26:     if episodes > 1 then
27:         if T % REPLAY_FREQUENCY = 0 then
28:             agent.learn(mem)
29:         end if
30:         if done is True then
31:             evaluate_model(agent)
32:         end if
33:     end if
34:     T ⇐ T + 1
35:     state ⇐ next_state
36: end while
```

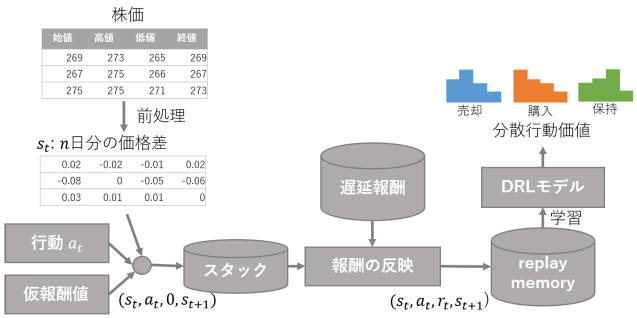



図 2　提案手法におけるデータフロー
**Fig. 2**　Dataflow of proposed method.

**Algorithm 2** エージェントの行動アルゴリズム

```
ActionValues ⇐ Model(s_t)
BuyRatio ⇐ [ ]
Sum ⇐ 0
for i = 1 to 225 do
    Action = GetMaxValAction(ActionValues[i])
    if Action is sell then
        sell stock i
    end if
end for
for i = 1 to 225 do
    Action ⇐ GetMaxValAction(ActionValues[i])
    if Action is buy then
        BuyRatio[i] ⇐ Max(ActionValues[i])
        Sum+ = BuyRatio[i]
    else
        BuyRatio[i] ⇐ 0
    end if
end for
for i = 1 to 225 do
    BuyRatio[i] / = Sum
    SubNoRiskAsset ⇐ NoRiskAsset × BuyRatio[i]
    buy stock i using SubNoRiskAsset
end for
```

う．1 行目に示す ActionValues には，すべての銘柄に対する各行動の価値が格納される．2，3 行目の変数は複数銘柄を購入するときの比率の計算に使用する．はじめに，4 行目から 9 行目に示すように，売却の行動を選択した銘柄についてすべての持ち株を売却する．次に，10 行目から 18 行目に示すように，購入を選択した銘柄について手持ちの無リスク資産で購入を行う．14 行目に示すように購入を選択したすべての銘柄の期待報酬値の和をとり，19 行目から 24 行目にかけてそれぞれの銘柄について期待報酬値が占める割合を計算し，その割合だけ無リスク資産を割り当てて購入する．維持を選択した銘柄については何も行動しない．

## 5.　日経 225 を用いた投資行動実験

### 5.1　実験設定

データは東京証券取引所第一部に上場しており，日経 225 に含まれる 225 銘柄を利用した．期間は 2010 年 1 月 4 日から 2019 年 12 月 30 日までの 10 年間の日足データを利用し，10 年分のデータが存在しない銘柄については，データが存在する年月日から 2019 年 12 月 30 日まで利用する．そのうちより新しい 1 年間のデータを評価に用いて残りのデータを学習に用いる．市場は過去の状態の影響を受けて将来の状態が決定していると考えられるため，未来の情報を学習しないようにする．

3 章で述べたように，行動 $a$ は各銘柄に対し保持している無リスク資産で株式を “購入”，保持している株式の “売却”，資産の売買を行わない “維持” の 3 つをとりうる．状態 $s$ には整形した日足データが含まれる．$r_t$ は報酬が得られてから与えられるため，それまで $(\boldsymbol{s_t}, \boldsymbol{a_t}, r_t, \boldsymbol{s_{t+1}})$ の組をスタックする．報酬が得られてからスタックしたデータに割引率を適用した報酬を与え，replay memory に保存される．

本実験で使用した計算機の CPU は AMD 3020e 1.20 GHz，RAM 容量は 4 GB である．実験環境は Open AI Gym をもとに構築した．初期状態として投資エージェントは無リスク資産である ¥1,000,000 を所有する．学習データを用いて 1 エポックだけ replay memory を構築し，2 エポック目から設定した頻度で experience replay による学習を行う．次に学習したモデルとリセットした初期資産を用いて投資行動を行う．資産額は無リスク資産と保持している株式の時価額の和で計算する．エージェントが購入したときの資産額を $Asset\_buy$，売却をしたときの資産額を $Asset\_sell$ としたとき，売却時の利益率は $Asset\_sell/Asset\_buy$ により計算される．1 回の実験では最終的な資産額 $m_i$，評価期間における利益率の平均値 $p_i$，利益率標準偏差 $\sigma_i$，シャープレシオ $s_i$ による 4 つの評価値を求める．

最終的な資産額は端的にどれほど資産が増減したかを評価する．利益率の平均値は，1 回の実験のうちに行った取引における利益率の平均値であり，1 度の取引で生まれる利益率の期待値を表す．利益率標準偏差は，平均値からどれだけ利益率のばらつきが生まれるかを表す．シャープレシオはとったリスクに対するリターンの大きさを示す．シャープレシオ [15] は評価期間における利益率の平均値 $p$ から無リスク資産の利子率 $R_f$ を引き，利益率標準偏差 $\sigma$ で割った値とする．実験では問題の簡単化のために無リスク資産の利子率 $R_f$ は 0 とする．

$$S = \frac{p - R_f}{\sigma} \tag{11}$$

報酬値に資産額の増減を用いているため，最終的な資産額は増加することを期待している．また，先行研究と同様にシャープレシオを用いており，DRL を用いることで低リスクであり資産額を増加させるような行動をとることでシャープレシオが増加することを期待している．利益率の平均値が増加，利益率標準偏差が減少すると，シャープレシオが増加する．評価値の変化が期待どおりで DQN より

優位な結果を示せれば，複数の銘柄に対して同時に投資行動を学習するためにDRLを用いることが有用であると検証できる．

DQNやDRLを用いた学習ではランダムな要素を含んでいるため，本実験では100回の投資行動実験を行い評価値の平均値と標準偏差を計算した．最終的な資産額，利益率の平均値，シャープレシオそれぞれの平均値は値が大きいほど良い結果を表し，利益率標準偏差の平均値は値が小さいほど良い結果を表す．各評価値の標準偏差は，ばらつきであるため値が小さいほど結果が良い．利益率は任意の銘柄で売却を行ったときに直前の購入から変化した総資産額を利用する．評価は5エポック学習してから実施した．DRLとDQNの共通パラメータとして，予備実験により，モデルは3層全結合とし隠れ層のノード数を2,048，Q学習の割引率を0.9，replay frequencyを4，experience replayにおけるバッチサイズを16，Adam-$\varepsilon$を$1.5\times10^{-2}$，1状態に含める日数を5日とした．DRLのパラメータは，モデルの最終層のbin数を71，Vmaxを10，Vminを$-10$とした．

### 5.2　各評価値の平均値と標準偏差

本実験ではDQNとDRLについて100回の投資行動の学習をする実験を行い，各評価値の平均値と標準偏差を計算した．最終的な資産額，評価期間における利益率の平均値，利益率標準偏差，シャープレシオによる評価値で，DRLを用いた提案手法とDQNを用いた比較手法について比較する．各評価値の平均値における結果を**表 1**に，各評価値の標準偏差における結果を**表 2**に示す．

各評価値の平均値（表1）の結果のうち(a)最終資産額，(b)利益率の平均値，(d)シャープレシオは値が大きいほど優れており，各評価値の平均値のうち(c)利益率標準偏差と，各評価値の標準偏差の結果（表2）については値が小さいほどばらつきが少なく優れている．各評価値の平均値では，(c)と(d)に示すようにDRLの方が利益率標準偏差とシャープレシオ，各評価値の標準偏差では(e)から(h)に示す結果についてDRLが上回る結果となった．特に各評価値の標準偏差の結果については大きな差があり，(e)最終資産額ではDRLはDQNの57.7%，(f)利益率の平均値は57.6%，(g)利益率標準偏差は61.6%，(h)シャープレシオは74.2%となっている．各評価値の平均値については多少の優劣はあるが大きな差はなかった．しかし，各評価値の標準偏差についてDQNよりDRLの方が小さな結果となった．

計算時間は5エポック学習して評価が終わった時点で，100回の実験においてDQNは平均22分52秒，DRLは平均67分47秒かかった．DRLはDQNの約3倍の計算時間がかかった．

**表 1**　各評価値の平均値
**Table 1**　Mean of each evaluation value.

| | DQN | DRL（提案手法） |
|---|---|---|
| (a) 最終資産額（円） | **1,085,988** | 1,075,999 |
| (b) 利益率の平均値 | **1.000387** | 1.000354 |
| (c) 利益率標準偏差 | 0.0106 | **0.01007** |
| (d) シャープレシオ | 96.175 | **100.153** |

**表 2**　各評価値の標準偏差
**Table 2**　Standard deviation of each evaluation value.

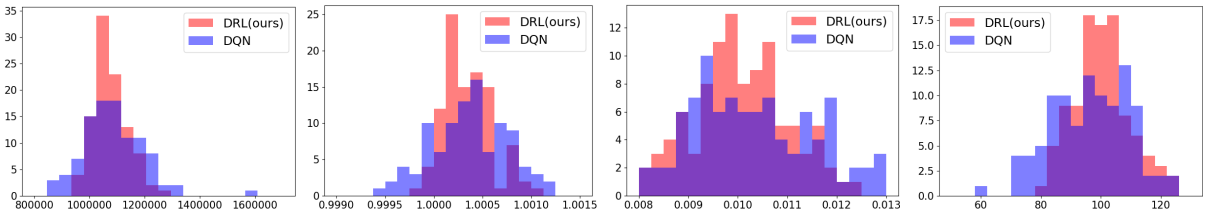
| | DQN | DRL（提案手法） |
|---|---|---|
| (e) 最終資産額（円） | 109,665 | **63,324** |
| (f) 利益率の平均値 | $4.24\times10^{-4}$ | $\mathbf{2.44\times10^{-4}}$ |
| (g) 利益率標準偏差 | $1.54\times10^{-3}$ | $\mathbf{9.48\times10^{-4}}$ |
| (h) シャープレシオ | 12.841 | **9.522** |

### 5.3　各評価値のヒストグラムの比較

DQNとDRLの結果について，各評価値のヒストグラムを比較した．最終資産額のヒストグラムを**図 3**(a)に示す．表1(a)より最終資産額の平均値ではDQNはDRLの1.0092倍となったが，ヒストグラムを観察して分かるようにDQNの方がばらつきが大きい．表2(e)よりDQNの標準偏差はDRLの1.73倍である．DRLは比較的最終的な資産額が小さかった一方，全体的には安定した投資行動を学習したと考えられる．DQNはDRLに比べるとより利益が大きかった反面，損失を出した結果も多かった．DRLの各評価値の標準偏差については，結果的にDQNよりも優位な結果となった．表1(b)が示す結果のように，図3(b)に示す利益率の平均値のヒストグラムではDRLとDQNの間にはそれほど差はない．一方で利益率の平均値の標準偏差はDRLの方が小さいことが分かる．利益率標準偏差のヒストグラムを図3(c)に示す．最終資産額や利益率の平均値ほど大きなばらつきの範囲の違いはないが，DRLの方が平均値付近に結果が分布している．シャープレシオのヒストグラムを図3(d)に示す．表1(d)が示すように，DRLの方がシャープレシオの平均値は大きく，優れていることを示しており，ヒストグラムを観察してもDRLの方がシャープレシオの値が大きい方に分布していることが観察できる．また，ばらつきに関してもDRLの方が小さいことが分かる．

### 5.4　有意性の検証

本研究では各評価値について平均値と標準偏差を計算しているが，結果の有意性について調べる．DRLとDQNにより得られた結果を2つの群として代表値の有意性を検定する．まず，DRLとDQNの平均値と標準偏差それぞれの評価値の結果について，シャピロ–ウィルク検定により標本に正規性があることを帰無仮説としテストした．有意水準を5%とすると**表 3**に示すp値のように，DQNの最終資産額，利益率の平均値，利益率標準偏差，DRLの最終資産額，利益率の平均値については帰無仮説が棄却され正規




(a) 最終資産額　(b) 利益率の平均値　(c) 利益率標準偏差　(d) シャープレシオ

図 3　DQN と DRL における実験結果のヒストグラム比較
**Fig. 3**　Histogram comparison between DQN and DRL results.

表 3　シャピロ–ウィルク検定による各評価値の p 値
**Table 3**　p values for each evaluation value by Shapiro-Wilk test.

| | DQN | DRL（提案手法） |
|---|---|---|
| 最終資産額（円） | $1.132 \times 10^{-3}$ | $7.327 \times 10^{-3}$ |
| 利益率の平均値 | $4.370 \times 10^{-2}$ | $4.293 \times 10^{-2}$ |
| 利益率標準偏差 | $9.350 \times 10^{-5}$ | 0.560 |
| シャープレシオ | 0.406 | 0.217 |

表 4　各評価値の有意性の検定結果
**Table 4**　Significance test results for each evaluation value.

| 評価値 | 検定 | p 値 |
|---|---|---|
| 最終資産額（円） | ウィルコクソンの順位和検定 | 0.596 |
| 利益率の平均値 | ウィルコクソンの順位和検定 | 0.614 |
| 利益率標準偏差 | ウィルコクソンの順位和検定 | 0.061 |
| シャープレシオ | Welch の t 検定 | 0.018 |

性を持たないことが分かった．一方，DQN のシャープレシオ，DRL の利益率標準偏差，シャープレシオでは帰無仮説は棄却されず，正規性を持たないとはいえない結果となった．

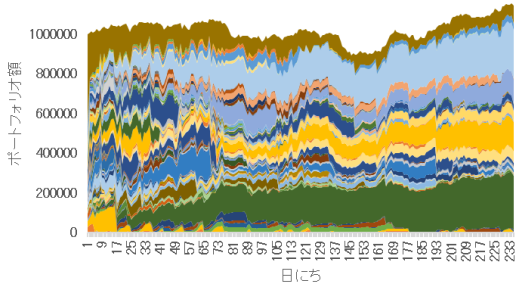
DQN と DRL の結果にはデータ間の対応がない．表 3 の結果から最終資産額，利益率の平均値，利益率標準偏差の結果についてはウィルコクソンの順位和検定を行った．シャープレシオについて F 検定を行ったところ 2 群間は不等分散であることが示されたため，Welch の t 検定を行った．これらの検定は両側検定で実施した．各検定結果を**表 4** に示す．

Welch の t 検定の結果，シャープレシオの結果に関しては有意水準 5%で統計的有意差があることが分かった．その他の結果については統計的有意差がないことが分かった．ただし利益率標準偏差に関しては p 値が 6.1%であり，有意水準に近い数値となった．よって 5.2 節の結果のうち，シャープレシオの結果に関しては統計的有意差が存在する．

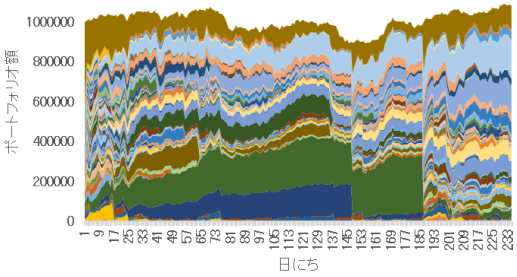
## 6.　考察

### 6.1　学習した投資行動の一例

本実験ではポートフォリオマネジメントを学習している．本節ではモデルが学習した行動の一例を示すとともに




(a) 学習 1 エポック




(b) 学習 5 エポック

図 4　学習エポック数によるポートフォリオ推移
**Fig. 4**　Portfolio transition for different epochs.

評価値や銘柄を比較し，学習の傾向を考察する．

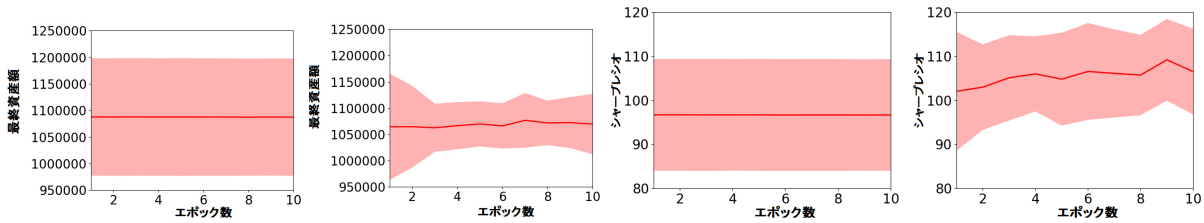
実験結果のうち，学習 1 エポック目と 5 エポック目のポートフォリオ推移を**図 4** に示す．積み上げ面グラフであり，銘柄ごとの資産額と無リスク資産を統合している．図 4 に示した色は銘柄ごとに異なり，最も上に積み上げられた面グラフは無リスク資産（茶色）を表す．

この例は 1 エポック目が最も最終資産額は大きく，学習エポックが進むにつれて最終資産額が減少している．ある程度保持し続ける銘柄に傾向はあるものの，学習するエポック数によってポートフォリオを占める銘柄は変化している．特定の銘柄に固着しておらず，エポックごとに別々の銘柄が利益に影響していることが観察できる[*1]．

### 6.2　学習エポック数による評価値の変化

前実験として，10 エポック分の学習を行いそれぞれのエポック終了時の評価値を観察し，実験でどのエポック数を

[*1]　一見すると特定の銘柄がポートフォリオを占める割合が増加しているが，株価は変化せず購入し続けていて増加しているのかなど，要因は判別できない．




(a) DQN:最終資産額　(b) DRL:最終資産額　(c) DQN:シャープレシオ　(d) DRL:シャープレシオ

**図 5**　学習エポック数による評価値の推移

**Fig. 5**　Transition of evaluation values for each epoch.

評価するか決定した．

**図 5** では赤い線が平均値，薄い赤が標準偏差 $\pm 1\sigma$ を示している．最終的な資産額では，DRL（図 5 (b)）は 3 エポックで大幅な標準偏差の減少がみられ，平均値は微小な上昇傾向が続いた．7 エポック付近で標準偏差が大きくなる傾向がみられた．シャープレシオについて観察すると，DRL（図 5 (d)）は全体的に上昇傾向がみられた．また，標準偏差は学習が進むにつれて減少傾向がみられた．DQN の評価値はいずれも大きな変化はなかった．DQN の学習が早く収束した原因として，モデルのノード数が関与していると考えられる．DQN は 1 つの行動に対して 1 つの行動価値を持つ一方，DRL は今回の場合 71 個の分布行動価値を持っており，さらにモデルが出力する行動の数は 675 個であり，モデルの大きさに違いがある．DRL の最終的な資産額の結果を優先し，5 章の実験では 5 エポック学習したモデルを利用して評価を行った．

## 7.　結論

DRL を用いた日経 225 を構成する銘柄における金融市場において投資行動手法の提案し，実験を行い DQN と比較した．実験では DQN に比べて DRL を用いた提案手法は 8 つのうち 6 つの評価項目で優位な結果を得ており，特に各評価値の標準偏差については DQN に対して大きく優れている．標準偏差が小さいほど安定した評価値を得られる投資行動が学習できているため，DRL を用いた提案手法は DQN より低リスクな投資行動を学習できているといえる．リスクに対するリターンの大きさを評価する指標であるシャープレシオの評価値に関して，DQN と DRL の結果の比較について統計的有意差が確認できた．

今後の課題として，結果の分析の観点では学習した投資行動を確かめる余地があると考えている．より定量的に評価するには局面や行動選択の時系列データについて時系列クラスタリングを取り入れて分析できると考えられる．また，比較手法が少ないため類似研究が採用している DDPG [24] との比較などが可能であると考えている．本研究は多くの先行研究 [9], [11], [13], [17], [21] などと同様にシャープレシオを補正せずに用いているが，文献 [25], [26] の研究によると観測したリターンによってはシャープレシオは過大評価され，本来の値は補正して算出するべきものであるとしている．今後はシャープレシオの補正についても検討していきたい．

## 参考文献


[1] Chan, N.T. and Shelton, C.: An electronic market-maker, *AI Memo 2001-005* (2001).

[2] Heaton, J.B., Polson, N.G. and Witte, J.H.: Deep learning in finance, arXiv:1602.06561 (2018).

[3] 常井祥太，穴田　一：NT 倍率取引における深層強化学習を用いた投資戦略の構築，第 22 回人工知能学会金融情報学研究会 (2019).

[4] 石原龍太：多層ニューラルネットワークと GA を用いた TOPIX 運用 AI，第 19 回人工知能学会金融情報学研究会 (2017).

[5] 加藤旺樹，穴田　一：遺伝的プログラミングを用いたテクニカル指標による金融取引の戦略木構築，第 24 回人工知能学会金融情報学研究会 (2020).

[6] 上田　翼，東出卓朗：人工知能を用いた金融政策予想と市場予測分布に基づく為替の投資戦略，第 18 回人工知能学会金融情報学研究会 (2017).

[7] 宮坂純也，穴田　一：心理的要素を考慮した投資行動モデル，第 18 回人工知能学会金融情報学研究会 (2017).

[8] Wu, J., Wang, C., Xiong, L. and Sun, H.: Quantitative trading on stock market based on deep reinforcement learning, *IJCNN 2019* (2019).

[9] 小林弘幸，和泉　潔，松島裕康，坂地泰紀，島田　尚：強化学習による高頻度取引戦略の構築，第 24 回人工知能学会金融情報学研究会 (2020).

[10] Shin, W., Bu, S.-J. and Cho, S.-B.: Automatic financial trading agent for low-risk portfolio management using deep reinforcement learning, arXiv:1909.03278 (2019).

[11] Xiong, Z., Liu, X.-Y., Zhong, S., Yang, H. and Walid, A.: Practical deep reinforcement learning approach for stock trading, *NIPS 2018 Workshop on Challenges and Opportunities for AI in Financial Services* (2018).

[12] Zhang, Y., Zhao, P., Wu, Q., Li, B., Huang, J. and Tan, M.: Cost-sensitive portfolio selection via deep reinforcement learning, arXiv:2003.03051 (2020).

[13] Ye, Y., Pei, H., Wang, B., Chen, P.-Y., Zhu, Y., Li, B. and Xiao, J.: Reinforcement-learning based portfolio management with augmented asset movement prediction states, *AAAI 2020* (2020).

[14] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D.: Human-level control through deep reinforcement learning, *Nature*, Vol.518, pp.529–533 (2015).

[15] Sharpe, W.F.: The sharpe ratio, *The Journal of Portfolio Management*, pp.49–58 (1994).
[16] Bellemare, M.G., Dabney, W. and Munos, R.: A distributional perspective on reinforcement learning, *ICML 2017* (2017).
[17] Harnpadungkij, T., Chaisangmongkon, W. and Phunchongharn, P.: Risk-sensitive portfolio management by using distributional reinforcement learning, *iCAST 2019* (2019).
[18] Dabney, W., Ostrovski, G., Silver, D. and Munos, R.: Implicit quantile networks for distributional reinforcement learning, *ICML 2018* (2018).
[19] Sato, Y. and Zhang, J.: Modeling low-risk actions from multilative time series data using distributional reinforcement learning, *iCAST 2020* (2020).
[20] Almahdi, S. and Yang, S.: An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown, *Expert Systems with Applications* (2017).
[21] Jiang, Z. and Liang, J.: Cryptocurrency portfolio management with deep reinforcement learning, *IntelliSys 2017* (2017).
[22] Jiang, Z., Xu, D. and Liang, J.: A deep reinforcement learning framework for the financial portfolio management problem, arXiv:1706.10059 (2017).
[23] Morimura, T., Sugiyama, M., Kashima, H., Hachiya, H. and Tanaka, T.: Parametric return density estimation for reinforcement learning, *UAI 2010* (2010).
[24] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. and Wierstra, D.: Continuous control with deep reinforcement learning, arXiv:1509.02971 (2015).
[25] Benhamou, E., Saltiel, D., Guez, B., and Paris, N.: Testing sharpe ratio: Luck or skill?, *SSRN Electronic Journal* (2019).
[26] Bailey, D.H. and de Prado, M.L.: The sharpe ratio efficient frontier, *Journal of Risk* (2012).

**佐藤 葉介**

2021 年岩手大学大学院総合科学研究科博士前期課程修了．機械学習の応用に関する研究に従事．

**張 建偉**　（正会員）

2005 年筑波大学大学院システム情報工学研究科博士前期課程修了．2008 年同大学院同研究科博士後期課程修了．博士（工学）．埼玉大学情報メディア基盤センター産学官連携研究員，京都産業大学コンピュータ理工学部特定研究員，筑波技術大学産業技術学部助教を経て，2017 年より岩手大学理工学部准教授．Web マイニング，Web 情報システム，情報保障の研究に従事．ACM，IEEE Computer Society，日本データベース学会，人工知能学会各会員．

（担当編集委員　田中 剛）