博士論文

東京大学大学院工学系研究科

システム創成学専攻

人工市場と深層学習の活用による
安全な金融市場予測の実現

前田 巌

学籍番号 37-187052

指導教員　和泉 潔 教授

2020 年 12 月 1 日

目 次

図一覧

表一覧

第1章　緒言

1.1　研究背景

1.1.1　金融市場予測と予測可能性への批判

金融市場予測は長きに渡り熱心な研究の対象であり，これまでに数多くの手法が提案されてきた [1, 2, 3, 4, 5]．古典的にはブラウン運動 [6] やマルチフラクタルモデル [7] を用いた市場価格変動のモデル化に始まり，より発展的な手法としてべき分布を用いた金融市場変動のモデル化 [8]，multi-agent model による金融市場のモデル化 [9] といったモデルベースの手法が多く研究，高度な複雑系である金融市場 [10, 11] のメカニズム解明に活用されてきた．中でも投資家を模した複数のエージェントにより市場を構成する agent-based models は金融市場のモデル化に適するとされ [12]，rebalancers and portfolio insurers model [13]，経済物理学の手法を取り入れた Levy-Levy-Solomon model [14] や Solomon-Levy-Huang model [15, 16]，percolation theory [17, 18] を金融市場に適用した Cont-Bouchaud model [19] ，および social percolation model [20, 21] といった手法が提案され，成果を挙げてきた．これらのモデルベースの手法は人工市場シミュレーション [22] の発展にも大きく寄与している．近年では計算機の発達とともに目覚ましい進歩を遂げている機械学習・深層学習を活用し，データドリブンに金融市場のメカニズムを学習する手法が多く提案されている [23, 2, 24]．こうした機械学習・深層学習を用いた手法は金融データの時系列性や様々な社会事象との関係性を取り入れ，古典的なモデルベース手法を大きく上回る精度での予測を可能としている．

金融市場予測はこうした多くの研究や成果に支えられている一方，その予測可能性には批判的な意見が存在する．Silver [25] によると，金融市場を含む経済の正確な予測が困難なことには以下の 3 つの要因があるとされる．

1. 因果関係を特定することの難しさ

経済・金融市場は社会のあらゆる事象に影響されており，因果関係や影響を及ぼす要素も無数に存在する．したがって，力学モデルや機械学習・深層学習モデルを用いて予測を行う場合，予測を行うために十分な説明変数を収集することや，金融市場の因果関係を表現するのに十分なモデルを作成することは非現実的である．

2. 内部構造の複雑性・非定常性
   金融市場は非常に高い複雑性を有している [26] 上にその内部構造は日々変化しており，影響を与える要素やその因果関係も常に変動している [27]．

3. データに含まれる多くのノイズ
   経済を予測する際には，もとになるデータが不正確であることが多く，そのようなデータを用いて学習した予測モデルは必ず不確かさを含んでしまう．

Silver の指摘した内容以外にも，投資家の行動により市場の内部状態が変化してしまうマーケットインパクト [28, 29, 30, 31] 等の要因が金融市場予測を困難とする要因として考えられる．また，金融市場予測は多くの場合投資判断に活用されるため，不確かな予測・誤った予測に基づいて取引を行うことは多大な損失や金融市場全体の不安定化を導く危険性がある．金融市場予測はその性質上，予測の不確実性も誤った予測を行うリスクも非常に大きいタスクなのである．深層学習を用いた画像解析・言語処理の分野においては特定のデータセットの予測精度をベンチマークとして手法の性能を競うことにより手法の優劣を比較することが広く行われているが，金融市場予測においては限定的な状況における予測精度以上に予測の安全性・信頼性を高めるような取り組みが必要である．

1.1.2 予測の安全性とは

予測の安全性については様々な解釈が考えられる．予測精度と類似した指標として解釈することや，投資実績の安定性を以て定義することも考えられる．本研究では以下の2つの状況下において危険な行動，すなわち大きな損失を招いたり，金融市場全体の不安定化の原因となるような行動を抑止する性能の度合いを安全性と考え研究を進める．

1. 現実市場では観測されていない市場状況

2. 説明変数の不足やデータに含まれるノイズにより予測が不確かになる状況

1. 現実市場では観測されていない市場状況 はモデルの学習データに含まれないため，モデルが適切な予測を行うことができない．このような状況のデータは一般に外挿 (out-of-distribution) [32, 33] データと呼ばれ，金融市場予測に限らず機械学習・深層学習研究の分野で広く知られている．特に金融市場においては，前述の内部構造の複雑性・非定常性により外挿データに直面する可能性が大きいと考えられる．

2. 説明変数の不足やデータに含まれるノイズにより予測が不確かになる状況 については，データの外挿・内挿 (in-distribution) に関わらず問題となる．モデルの予測が常に完璧でないことを理解し，予測が外れるリスクを考慮した行動選択を行う必要がある．

金融市場予測の安全性の評価については，定量的には行動選択における利益の平均や，大きな損失の頻度および程度を比較することにより行う．加えて，個別のケーススタディにより行動パターンを定性的に解析し，学習戦略の安全性を評価する．

1.1.3 本研究の方針

1.1.2 章で予測の安全性，および危険な予測を行いうる状況について定義した．これらの定義に基づき，以降では本論文のアプローチについて述べる．

はじめに，現実市場では観測されていない市場状況への対処について．モデルが適切に予測を行うことができない外挿データの存在は，機械学習・深層学習を用いたモデル化においてはデータの不足と言い換えることができる．一般的に機械学習・深層学習モデルは学習に大量のデータを必要とし [34, 35]，必要なデータサイズは予測タスクの複雑性に応じて上昇する．金融市場は高度に複雑であり，日々その内部構造が変化するため過去のデータは有効性が著しく下落してしまうため，適切な学習に必要なデータを用意できない．データサイズの不足は予測精度の低下だけでなく，市場の突発的な変動 [36, 37] に代表される頻度の低い市場変動を学習する機会が減り，予測の安全性を大きく悪化させる原因となる．したがって，何らかの手段によりデータサイズを拡大させることができれば，予測の安全性を向上させることができると考えられる．

機械学習・深層学習分野においてデータサイズの拡大を行う手法はデータ拡張 (data augmentation) [38, 39, 40, 41] と呼ばれ，金融予測分野に限らず盛んに提案されている．画像解析分野においては画像の回転や拡大縮小，上下左右シフトや上下左右反転により学習データを拡張することにより，学習効率が向上することが知られている．しかし金融

市場データ・金融時系列は複雑な生成分布や制約条件を持ち，画像データのように単純なデータ拡張を行うことができない．

そこで本研究では人工市場 (artificial market) [22, 42, 43, 44] を用いたデータ拡張を目指す，人工市場はは計算機上に仮想的な金融市場を作成し，取引を行う枠組みであり人工市場を用いて作成したデータを擬似的な実市場データとして機械学習・深層学習モデルの学習に利用することにより，学習データの不足を補うことができる．加えて，人工市場の導入は実市場データを用いた学習では実現不可能な マーケットインパクト の考慮も可能とする．人工市場シミュレーション上で同時にモデルによる予測を行い，予測に応じた行動選択をシミュレーションに反映することで予測者の行動を市場が織り込むことができ，学習時にも予測者の マーケットインパクト を考慮することができる．

以上より，人工市場を用いた金融市場データの拡張は金融市場予測の安全性を大きく向上させると期待される．しかし現在までにそのような研究例はごく少数しか報告されておらず，その成果も限定的である [45]．原因としては，以下のような課題の存在が考えられる．

課題 1. 人工市場環境で深層学習モデルを学習させるための理論面・技術面における知見不足

人工市場は単体で研究分野を構成するほどのテーマであり，金融市場の構成，エージェント (投資家) の構成，ハイパーパラメータ設定等考慮すべき事象が非常に多い．加えて人工市場環境上で深層学習モデルを学習させる場合，学習に適した人工市場環境の構築，および深層学習モデルの学習を同時進行で行う実装が必要となる．適切な学習，実験を行うためには既存研究に加えて大きな理論面・技術面における進展が必要である．

課題 2. 人工市場と実市場の乖離

本研究では人工市場を実市場の代替として用いることを目指している．しかし，実市場と同じような挙動や市場間相互作用，外的要因の反映を人工市場シミュレーションにおいて再現することは困難である．Stylized facts [46, 47, 48] と呼ばれる，実市場が持つ一般的な統計的性質を部分的に再現することは現在の人工市場で可能とされているものの，実際に現実市場の代替として用いることは難しいことが知られている．

本研究で目指す，人工市場を用いた実市場データの拡張を達成するためには，上記の

問題を解決する必要がある．したがって本論文では，それぞれの問題に対応し，2 章では人工市場と深層強化学習の活用による取引戦略学習研究を，3 章では模倣学習を用いた注文行動モデリング研究を行う．ここで，人工市場環境で学習させる深層学習モデルとしては深層強化学習 (deep reinforcement learning, DRL) [49, 50, 51] モデルを想定する．先述の通り マーケットインパクト を考慮した学習を行う場合，予測モデル自身がエージェントの行動を決定することが必要となる，一般的な分類 (classification) あるいは回帰 (regression) モデル単体ではこのような行動決定は実現できないため，行動法則を直接深層学習モデルで近似する深層強化学習を採用した．

次に，説明変数の不足やデータに含まれるノイズにより予測が不確かになる状況への対処について．このような予測の不確かさは現実世界のデータを用いる以上回避が難しく，一般的にはデータ拡張によっても解決しない．予測の不確かさに対処する手段としては，モデルが不確かさを見積もった予測を行い，見積もられた不確かさにより行動選択を変更することが考えられる．本論文では不確かさを見積もる方法としてベイズ深層学習 (Bayesian neural networks, BNNs) を採用し，ベイズ深層学習モデルを用いた金融市場予測手法，および予測を用いた投資意思決定アルゴリズムを提案，実験を行う．以上のベイズ深層学習を用いた不確かさの考慮研究は 4 章にて詳しく述べている．

1.2　論文構成

本研究の構成を図 1.1 に示す．本研究では大きな目標である人工市場を用いた実市場の補完を目指し，1.1 章で述べた 3 つの課題に対しそれぞれ手法提案を行っている．本論文の 2 章，3 章，4 章がそれぞれ課題 1，2，3 に対応しており，2 章では人工市場と深層強化学習の活用による取引戦略学習フレームワークの提案，3 章では模倣学習を用いたデータドリブンなシミュレータ学習，4 章ではベイズ深層学習を用いた金融市場予測の不確かさ考慮について詳説している．

2 章では人工市場シミュレーション上で深層強化学習モデルを学習させるためのフレームワーク，金融市場およびエージェント構成，深層強化学習モデル構成，報酬関数設計を行っている．2.1 章では関連分野についての先行研究を分野ごとに (2.1.1 章では取引戦略学習，2.1.2 章では深層強化学習，2.1.3 章では人工市場について) 概説している．2.2 章では提案フレームワークの詳細について述べている．2.2.1 章ではフレームワークの全体像について述べ，2.2.2 章，2.2.3 章，2.2.4 章ではそれぞれ提案フレームワークを構成する markes，stylized agents，DRL agents の構成について解説している．2.2.5 章では使用した深層強化学習モデルの手法，説明変数，ネットワーク構成について詳説し，2.2.6 章では深層強化学習モデルの学習に不可欠な報酬関数の設計を行っている．2.2.7 章にはこれらのフレームワークの実装方法が記載されている．提案フレームワークを用いた実験も行っており，2.2.8 章で実験条件を，2.2.9 章で実験結果を記述し，加えて 2.2.10 章で学習戦略の妥当性検証を行っている．さらに，提案フレームワークを用いた複数の深層強化学習手法の比較検討を 2.3 章で行っており，2.3.1 章で実験環境を，2.3.2 章で深層強化学習モデルの構成を，2.3.4 章で実験条件を説明している．2.3.5 章では比較実験で得られた結果について解説している．2.4 章で研究の結論を述べている．

3 章では模倣学習 (imitation learning, IL) を用いた注文行動モデリングについて研究を行っている．3.1 章では関連研究の先行研究を分野別に (3.1.1 章では模倣学習，3.1.2 章では latent segmentation について) 概説している．3.2 章では latent segmentation と multi-modal imitation learning を用いたラベルなしデータからの複数取引戦略学習手法を提案している．3.2.1 章では手法の概要を，3.2.2 章では説明変数，ネットワーク構成を含むモデル構成を解説している．提案手法を用いた実験はシミュレーションデータ，実市場データ両者に対し行っており，3.2.3 章で実験概要を，3.2.4 章でシミュレーションデータを用いた実験結果を，3.2.5 章で実市場データを用いた実験結果を示している．3.3 章では本

章の結論と課題を述べている．

4 章ではベイズ深層学習 (Bayesian neaurl networks, BNN) を用いた金融市場予測における不確かさ考慮研究を行っている．4.1 章ではベイズ深層学習の理論について，概要 (4.1.1 章)，学習 (4.1.2 章)，variational dropout(4.1.3 章) にわけて解説している．ベイズ深層学習の有効性確認のために本研究では 3 つの実験を行っている．4.2 章ではベイズ深層学習を含む深層学習における不確かさ考慮手法の比較検討を行っており，4.2.1 章で予測タスクおよびベースラインとなる手法について，4.2.2 章でベースラインモデルの予測の問題点について，4.2.3 章で提案手法であるベイズ深層学習モデルの構成について，4.2.4 章で比較手法について解説している．4.2.5 章ではベンチマークとなる不確かさ評価の定量化手法を示しており，4.2.6 章で実験結果について詳説している．4.3 章では金融市場予測の代表的なケースである株価動向予測についてベイズ深層学習を適用しており，4.3.1 章で解析に用いた実データセットについて，4.3.2 章でベイズ深層学習を適用する予測タスクについて，4.3.3 章でモデルの入力となる説明変数の設計について，4.3.4 章で提案手法であるベイズ深層学習モデルについて，4.3.5 章で比較手法について述べている．実験結果は 4.3.6 章で解説している．最後に，ベイズ深層学習モデルによる予測を投資アルゴリズムと組み合わせる実験を 4.4 章で行っており，4.4.1 章でベイズ深層学習により特異的に得られる予測結果を活用した投資判断アルゴリズムの提案を，4.4.2 章で実験に用いたデータセットの説明を，4.4.3 章で使用した説明変数の解説を，4.4.4 章で価格予測に用いたベイズ深層学習モデルの記述を行っている．実験結果は 4.4.5 章で示されている．ベイズ深層学習の活用についての結論と課題は 4.5 章で述べられている．

以上の内容を元に，5 章で本研究がまとめられている．

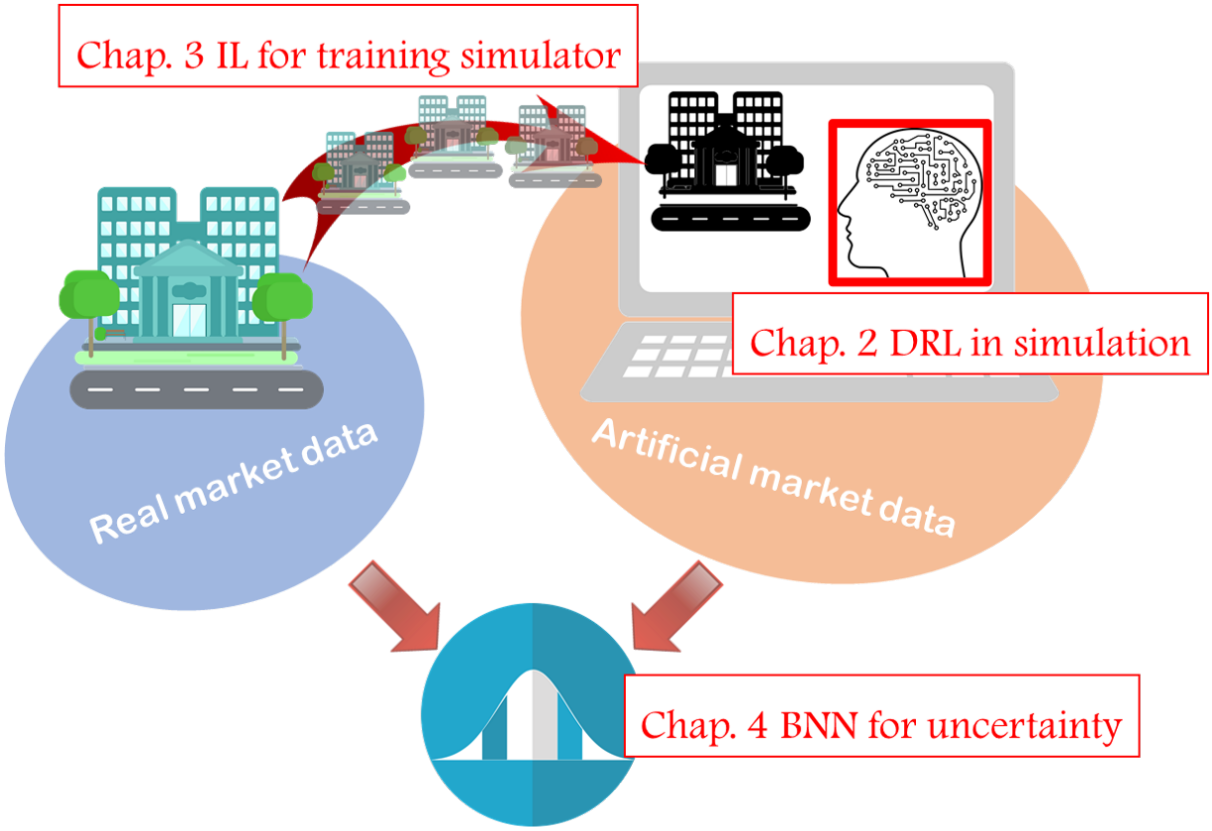


図 1.1: 本研究の構成．本研究では目的である人工市場を用いた実市場の補完に付随する 3 つの問題を解決するため，2 章，3 章，4 章でそれぞれ手法提案を行っている．2 章では人工市場と深層強化学習の活用による取引戦略学習フレームワークの提案，3 章では模倣学習を用いたデータドリブンなシミュレータ学習，4 章ではベイズ深層学習を用いた金融市場予測の不確かさ考慮について詳説している．

第2章　人工市場と深層強化学習の活用による取引戦略学習

2.1　関連研究

2.1.1　投資戦略学習

投資戦略学習は長い歴史を持つ [52]．古典的には，dynamic model-based (力学モデルベース) の手法により投資家行動を記述する手法が多く提案されてきた．例としては snipping strategy [53]，Zero intelligence strategy，risk-based bidding strategy などが挙げられる．近年ではそれまでのモデルベースのものと異なり，データドリブンで戦略を学習させる手法が台頭している．特に近年は深層学習を用いて強化学習 [54, 55, 56] を行う深層強化学習 (deep reinforcement learning, DRL) [49, 50, 51] の進歩がめざましく，金融市場応用も進んでいる [57, 58]．本研究と似た株式市場への適用としては，DRL for financial portfolio management [59]，market making via reinforcement learning [60]，DRL for price trailing [61] などが存在する．

2.1.2　深層強化学習

深層強化学習は大まかに価値ベースの手法と方策ベースの手法に分類される．価値ベース (value based) の深層強化学習手法 [62, 63] では，行動価値関数 (Q-function) を深層学習により近似し，予測時は Q-function が最大となる行動を選択する．このような手法は deep q-network (DQN) と呼ばれる． 深層強化学習は一般的に学習効率が悪く，様々な学習効率化手法が提案されている．Rainbow [64] では，double DQN [65] や dueling network [66] に代表される主要な深層強化学習の方法論を組み合わせることで劇的に性能が向上することが報告されている．General reinforcement learning architecture (Gorila) [67] により学習の高速化に寄与する並列学習が提案されている．Ape-X [68] では過去の行動履歴を学習対象として再現する experience replay [69] を用いた分散学習が提案されている．

方策ベース (policy based) の深層学習手法 [70] では価値ベースと異なり，行動戦略 (方策) を直接深層学習で近似する．中でも，方策関数 (actor) と行動価値関数 (critic) を同時に学習する actor-critic は盛んに研究されており，並列非同期と advantage の導入を施した asynchronous advantage actor-critic [71]，DQN の拡張として方策学習を行う deep deterministic policy gradient (DDPG) [72]，trust region policy optimization (TRPO) [73] 等が提案されている．

深層強化学習の応用先は幅広く，特に囲碁 [74] やビデオゲーム [49, 75, 76]，自動運転 [77] 等への応用は広く知られている．一方，深層強化学習モデルの学習にはモデルの行動に合わせて変化する環境シミュレータが必要であり，巨大で複雑な環境系における深層強化学習は発展途上である [78]．

金融市場における深層強化学習応用は主に過去取引データを用いて行われている．例としては DRL for financial portfolio management [59]，cryptocurrency portfolio management [58]，deep direct reinforcement learning [79] などが存在する．

2.1.3 人工市場

人工市場および人工市場シミュレーションは，マーケットマイクロストラクチャー (market microstructure) [80, 81, 82] や金融市場の規制 [83, 84] の研究に用いられてきた．特に投資家を模した複数のエージェントを用いてシミュレーションを行う agent based financial market simulation [85, 86, 87] は人工市場研究領域において広く用いられている．rebalancers and portfolio insurers model [13] や経済物理学による手法 [14] に代表される初期の agent based simulation は，stylized facts [46, 47, 48] と呼ばれる，現実の金融時系列が持つ基本的な統計的性質を再現できなかった．しかしその後の研究により，percolation theory の応用 [20, 18] や fundamentalist and chartist model [9] といった手法が開発され，徐々に stylized facts の観測が可能となった．

現代においては高頻度に大量の注文を行う高頻度取引 (high frequency trading, HFT) [88, 89, 90] の発達に伴い，人工市場の構造もそのような取引形態の反映が求められている．人工市場における high frequency trader の適用例としては，Hanson らによる研究 [91, 92]，McGroarty らによる研究 [93]，Leal らによる研究 [94] といった先行例が存在する．

2.2　人工市場における深層強化学習の実現

2.1.2 章で述べたとおり，複雑な系における深層強化学習は技術的な課題が大きく実現が難しい．本研究の対象である金融市場も高度な複雑系であり，その人工的な再現は人工市場として一大研究分野となっている．本研究では金融市場の環境シミュレータとして人工市場を採用し，人工市場シミュレーション上で深層強化学習モデルを学習するフレームワークの提案を行った．

本研究の一部は Journal of Risk and Financial Management で発表している [95].

2.2.1　フレームワークの概要

本研究のフレームワークの概要を図 2.1 に示す．シミュレータは金融市場およびエージェント (投資家) で構成されている．金融市場は本シミュレーションにおける環境であり，エージェント の行動に応じてその内部状況が変化していく．本シミュレーションにおいてはランダムで選ばれた エージェント の行動 および エージェント の行動に合わせた金融市場の変化を 1 step とし，一定ステップの行動により 1 回のシミュレーション (現実市場における引けまで) を構成している．エージェントの目的は，以下の式で計算される個々の資産 (capital) c を最大化することである．

$$c = m + \sum_{i}^{\text{Market}} p_i x_i \tag{2.1}$$

ここで，m はエージェントの現金を，p_i および x_i は 金融市場 i に対応した 仲値および エージェントの株式保有量を表し，シミュレータに登録されている市場全てについての総和を計算している．エージェント はルールに基づき行動選択を行う stylized agent と深層強化学習により行動選択を行う DRL agent が存在する．DRL agent は一定間隔ごとにネットワークの更新を行う．

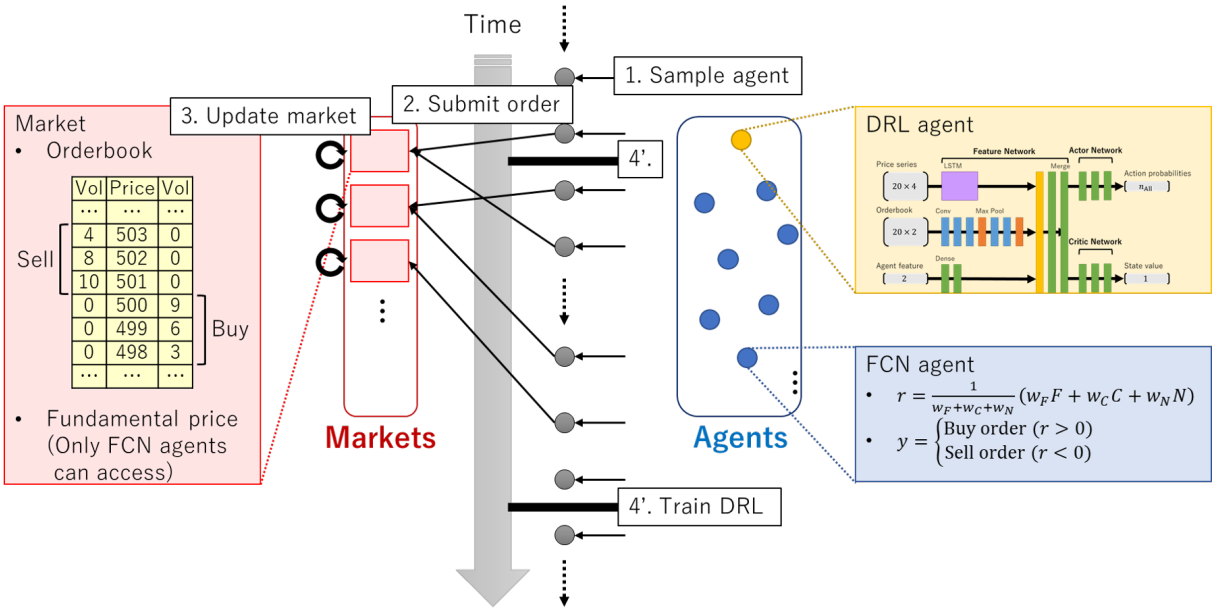


図 2.1: 人工市場シミュレーション上で深層強化学習を行うフレームワーク．シミュレータは金融市場およびエージェント (投資家) で構成されており，ランダムで選ばれたエージェント の行動およびエージェントの行動に合わせた金融市場の内部状態の変化によりシミュレーションが進行していく．エージェントは stylized (FCN) agent および DRL agent で構成されている．DRL agent の構成については図 2.3 で詳説している．

2.2.2　金融市場の構成

金融市場はそれぞれ，エージェントの注文を格納する orderbook (注文板) および fundamental price (理論価格) を持つ．エージェントが出すことができる注文は以下の 3 種類が存在する．

- Limit order (指値注文，LMT)
- Market order (成行注文，MKT)
- Cancel order (キャンセル注文，CXL)

LMT は price (価格)，quantity (数量)，side (buy または sell) を要素として持つ．MKT は price を持たないため，quantity および side を要素として持つ．CXL は取り消しする注文の ID で構成される．注文の ID は金融市場により振られ，その注文を出した エージェント のみが CXL を出すことができる．また，一定時間約定しない注文は自動的に取り消される．本来の金融市場では注文の一部を取り消す数量変更注文も存在するが，本研究では実装上注文すべてを取り消す CXL のみ可能とした．

市場の価格決定は continuous double auction (連続双方向オークション, CDA) [96, 97, 98] に従う．LMT については注文と逆の side で価格が合致する注文が存在する場合，MKT については反対側の side の注文が存在する場合約定が行われる．時間優先の原則に則り，同一価格の注文が複数存在する場合は先に出された注文から順に約定する．約定数量 v は以下の式で決定される．

$$v = \min(v_{\text{buy}}, v_{\text{sell}}) \tag{2.2}$$

ここで，v_{buy} および v_{sell} が買い注文および売り注文の数量である．買い数量および売り数量 決定後は，株式と現金のやり取りが行われる．以上の作業を約定可能な注文がなくなるまで続ける．ただし，シミュレーション開始後 1000 行動は寄せ前として注文行動を行い，1000 行動目板寄せ処理 (約定可能な注文すべての約定) を行う．

Fundamental price は市場の現時点での公正な取引価格を表す仮想的な価格であり，人工市場シミュレーションにおいては外的要因による価格変動を再現するために採用される．Fundamental price は stylized agent のみから観測でき，DRL agent は orderbook の

情報のみから予測を行う．Fundamental price は geometric Brownian motion [99, 100] (幾何ブラウン運動, GBM) に従い各ステップ変動する．

シミュレーションを行う際には，以下のハイパーパラメータを定める必要がある．

- Tick size: 価格の最小単位
- Initial fundamental price: Fundamental price の初期値
- Fundamental volatility: GBM のボラティリティ項

2.2.3 Stylized Agent の構成

Stylized agent [101] は，agent based simulation で広く用いられている エージェントであり，投資による対数リターン r を以下の式で計算する．

$$r = \frac{1}{w_F + w_C + w_N}(w_F F + w_C C + w_N N) \tag{2.3}$$

すなわち，r を 3 つの要素 F，C，N の重量平均で計算する．F は fundamental term と呼ばれ，現在の市場価格 p_t と fundamental price p_t^* の乖離度を表す．F は p_t，p_t^*，および time scale T を用いて以下のように計算される．

$$F = \frac{1}{T}\log\left(\frac{p_t^*}{p_t}\right) \tag{2.4}$$

Fundamental price p_t^* の観測値については金融市場に紐付いた真の値に正規分布ノイズを加えて生成する場合も多い．しかしながら本実験においては，後述のノイズ項で正規分布の測定誤差を内包できるため真の p_t^* の値を用いて計算を行った．

C は chart term と呼ばれ，価格のトレンドを表す．C は p_t，T および T ステップ前の市場価格 p_{t-T} を用いて以下のように計算される．

$$C = \frac{1}{T}\log\left(\frac{p_t}{p_{t-T}}\right) \tag{2.5}$$

N は noise term と呼ばれ，以下の正規分布からサンプリングされる．

$$N \sim \mathcal{N}(\mu, \sigma^2) \tag{2.6}$$

2.3 式の重み w_F，w_C，w_N は各 エージェント について指数分布からサンプリングを行い，エージェント の行動ルールに違いを作る．w_F，w_C，w_N のサンプリングについては一様分布等を利用することも考えられるが，対数リターンの予測値に対し 1 つの指標が支配的にならないよう，指数分布を利用している．指数分布の scale parameter σ_F，σ_C，σ_N は各シミュレーションごとに以下の式で決定する．

$$\sigma_F = 1 \tag{2.7}$$

$$\sigma_C \sim \mathcal{N}(\mu_C, \sigma_C^2) \tag{2.8}$$

$$\sigma_N \sim \mathcal{N}(\mu_N, \sigma_N^2) \tag{2.9}$$

ここで，w_F，w_C，w_N は重量平均に用いる相対値のため，3 値のうち 1 つの σ_F を固定している．また，w_F，w_C，w_N は対数の

対数リターン r を予測後，stylized agent は以下の式で将来価格を計算する．

$$p_{t+T} = p_t \exp(rT) \tag{2.10}$$

その後，$p_{t+T}(1-k) > p_t$ であれば buy LMT (買い指値注文) を，$p_{t+T}(1+k) < p_t$ であれば sell LMT (売り指値注文) を金融市場に出し，それ以外であれば注文行動を行わない．ここで，k は order margin と呼ばれ，取引による見込み利益の大きさを表す．Stylized agent の注文数量 v は離散一様分布 $u\{1, 5\}$ からサンプリングを行う．

Stylized agent はその利用により，金融市場の時系列が持つ基本的な統計的性質，すなわち stylized facts [46, 47, 48] 再現が可能となることが知られている．ただ，本研究のように適応的な戦略で学習する DRL agent が存在する場合，stylized facts の再現が可能であるかは不明確である．

2.2.4 DRL agent の構成

DRL agent はその名の通り，深層強化学習により注文行動の選択を行い，金融市場に投稿を行う．注文行動の選択方法としては，方策関数の予測およびルーレット選択を採用した．すなわち，有限個の行動選択肢それぞれについて選択確率を DRL モデルで予測し，予測確率を用いてルーレット選択で行動を決定する．

前述のような行動選択を行うためには，選択可能な注文行動の離散化を行う必要がある．ここで，価格，数量といった特徴で代表される注文行動の選択肢は非常に多く存在し，それら全てを選択可能とした場合学習に膨大な計算時間が必要となってしまう．そこで本研究では，一般的に株式市場においては仲値中心に取引が行われ，仲値から離れた価格に注文が出される場合は稀であることを考慮し，行動選択肢を少数に限定し学習効率の向上を図った．

本研究で提案する注文行動の離散化方法を図 2.2 に示す．．図 2.2 では，注文行動を以下の 5 カテゴリに離散化している．

1. Action: [Buy, Sell, Do-nothing]
2. Market: n_m 件
3. Type: [LMT, MKT, CXL]
4. Price: n_p 件
5. Volume: n_a 件

まず，action としては Buy (買い注文)，Sell (売り注文)，Do-nothing (注文行動なし) の 3 カテゴリが存在する．Buy または Sell の場合は，注文を行う金融市場を n_m (マーケット数) 件から，注文の Type を LMT, MKT, CXL の中から選択する，LMT の場合は Price を n_p (価格の選択可能数) 件から，Volume を n_a (数量の選択可能数) 件から選択する．MKT の場合は価格は不要ななため Volume のみを n_a (数量の選択可能数) 件から選択する．CXL については予測時にエージェントが出している注文の情報が必要なため離散化が難しい．したがって本研究では注文分類を単純化し，現在市場に登録されているエージェント の注文のうち最も高い価格の注文，最も低い注文の 2 カテゴリに離散化を行った．なお，詳しくは後述するが，取り消しできる注文がない状態で CXL が選ばれた

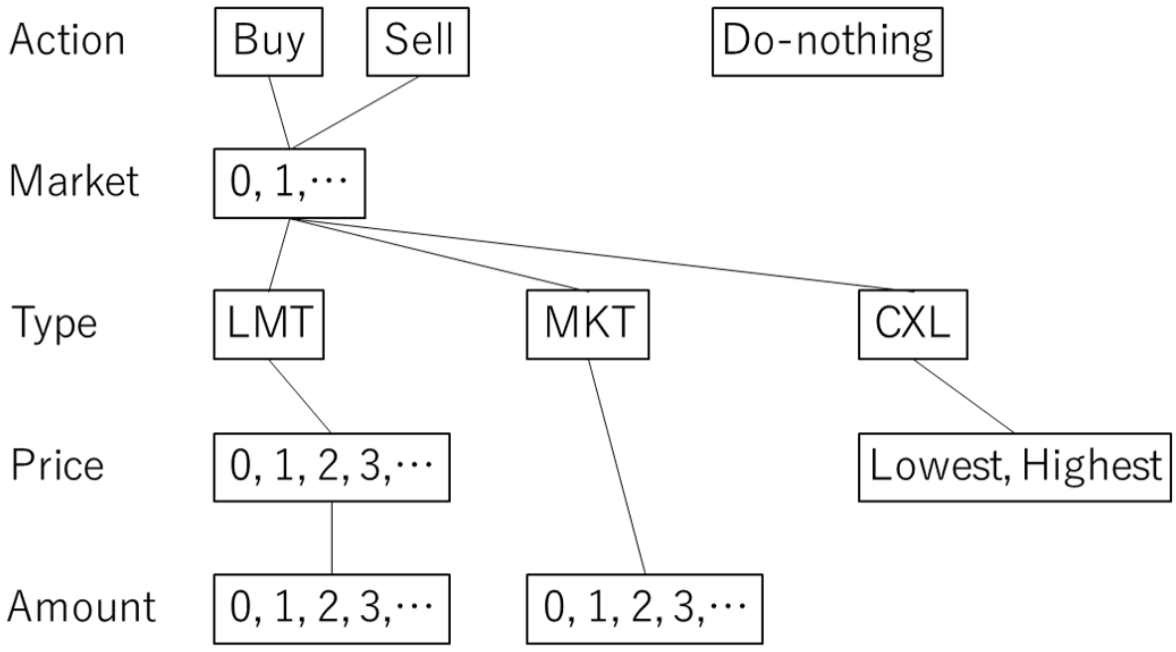
場合，行動は Do-nothing に変更された上で学習時に報酬にペナルティが与えられる．以上をまとめると，選択可能な全行動数 n_{all} は以下のように計算される．

$$n_{\text{all}} = 1 + 2n_m(n_p n_a + n_a + 2) \tag{2.11}$$

本実験では単一マーケットの場合について実験を行うため $n_m = 1$ となる．注文価格 (Price) としては前述のように，最良気配値からの差を $\{0, 2, 4, 6, 8, 10\}$ の 6 カテゴリにパラメータ化することにより，仲値前後の価格を選択可能とした．選択された Price の値を用いて，実際の注文価格 p は以下のように計算できる．

$$p = \begin{cases} p_{\text{bestask}} - \text{Price} & (\text{Action} = \text{Buy}) \\ p_{\text{bestbid}} + \text{Price} & (\text{Action} = \text{Sell}) \end{cases} \tag{2.12}$$

Price $= 0$ の場合，買い注文ならば best ask の価格で，売り注文ならば best bid の価格で注文を行うため，実質的に MKT に近い行動選択を行っていることになる．Price が大きい場合，より自身に有利な価格で注文を行うことになる．Volume については最小取引単位を 1 とし，1 あるいは 5 の 2 カテゴリにパラメータ化した．以上より，選択可能な前行動数は $1 + 2 \times 1 \times (6 \times 2 + 2 + 2) = 33$ となる．



$$n_{\text{all}} = 1 + 2n_m(n_p n_a + n_a + 2)$$

図 2.2: A2C モデルで予測する注文行動の離散化方法．本実験では，行動を Action，Market，Type，Price，Amount の 5 カテゴリに分類し，それらの組み合わせにより選択可能な行動の集合を定義した．

2.2.5 DRL モデルの構成

本実験で用いる DRL モデルとしては，2.1.2 章でも解説した policy based 手法の一種として advantage actor-critic (A2C) を用いた．A2C はその名の通り，

1. Advantage
2. Actor-critic

の 2 つの要素を組み合わせた深層強化学習手法である．Actor-critic [102] は，方策関数を予測する actor network，および状態価値関数を予測する critic network の 2 つのネットワークを同時に学習する手法である．

Actor network の勾配 ∂_θ および critic network の勾配 ∂_{θ_v} は以下の式により計算される．

$$\partial_\theta = \nabla_\theta \log(\pi(a_t|s_t;\theta))(R_t - V(s_t;\theta_v)) + \beta\nabla_\theta H(\pi(s_t;\theta)) \tag{2.13}$$

$$\partial_{\theta_v} = \nabla_{\theta_v}(R_t - V(s_t;\theta_v))^2 \tag{2.14}$$

$\pi(a_t|s_t;\theta)$ は actor network により予測された方策関数 $\pi(s_t;\theta)$ の，選択された行動 a_t に対する尤度関数を表す．本研究のように行動選択確率を予測する場合，a_t は実際に選択された行動のラベルで記述される．s_t は市場およびエージェントの状態を表し，特徴量として actor network に入力されることで方策関数の予測に反映される．θ は actor network のパラメータである．R_t が報酬関数を，$V(s_t;\theta_v)$ は critic network により予測される状態価値関数を表している．θ_v は critic network のパラメータである．

2.13 式の第一項は方策勾配と呼ばれ，報酬が大きい行動の選択確率が大きくなるようにネットワークが更新される．方策勾配の計算の際，通常の獲得報酬ではなく，advantage と呼ばれる，状態価値関数の予測値を用いた補正報酬 $R_t - V(s_t;\theta_v)$ を用いることで学習の効率化が達成されることが報告されている．2.13 式の第二項に見られる $H(\pi(s_t;\theta)$ は方策関数のエントロピーを表しており，エントロピー項の導入により決定的な行動選択になることが抑制される．β はエントロピー項の損失関数全体に対する重みを表す係数であり，通常 0.01 程度の値が用いられる．2.14 式は状態価値関数についての予測誤差を表し，

二乗誤差が小さくなるようにネットワーク更新を行うことで，正確な状態価値観数の予測が可能となる．

A2C モデルの学習に際しては，actor network と critic network がネットワークの一部を共有することで，学習効率が向上することが知られている [71]．加えて，特定の行動に選択が集中すると学習が進まなくなってしまうため，一定確率でランダム行動を行うことで行動選択を分散させる ϵ -greedy [103] の導入も推奨されている．

A2C へのインプットとする特徴量としては，以下の 3 種類を用いた．

1. Price series
2. Orderbook features
3. Agent features

Price series は 50 ステップごとに抽出した 20 件の市場の価格の代表値の行列である．価格の代表値としては，market price (直近の約定価格)，best ask (売り最良気配値)，best bid (買い最良気配値)，mid price (仲値) の 4 種類を使用した．水準の影響を取り除くため，それぞれの価格は最初の値で正規化している．Price series の次元は 20×4 である．

Orderbook features は仲値中心上下 10 価格に対応する数量のベクトルである．買い売りの数量を区別しやすくするため，買い注文の数量は -1 倍している．加えて，板状の全数量と エージェント の出した数量それぞれについてベクトルを作成し，20×2 の入力を作成した．

Agent features は エージェント の現金および各金融市場に対応する保有株式量で構成される $1 + n_{\text{market}}$ 次元のベクトルであり，本実験においては $n_{\text{market}} = 1$ のため 2 次元のベクトルとなる．空売りしている場合保有株式量は負の数となる．

本実験で用いた A2C モデルのネットワーク構成を図 2.3 に示す．本実験で用いた A2C では，前述の Price series，Orderbook features，Agent features からそれぞれ LSTM 層，Convolutional (Conv) および Maximum pooling (MaxPool) 層，Fully-connected (Dense) 層で特徴抽出を行い，特徴量を結合後 Fully-connected (Dense) 層で方策関数 (Action probabilities) および状態価値関数 (State value) を予測する．Price series を用いた金融市場動向予測は近年多く行われており，その多くが Long short-term memory (LSTM) [104, 105] を用いて特徴抽出がなされている [106, 107, 24, 108, 109]．したがって本実験でも Price

series からの特徴抽出には LSTM を用いた．また，位置情報が存在する Orderbook features からの特徴抽出には多くの研究で CNN が用いられており [110, 111, 112]，本研究でも Conv 層 および MaxPool 層を用いて特徴抽出を行った．LSTM 層の活性化関数には tangent hyperbolic 関数を，Conv および Dense 層の活性化関数には ReLU 関数を用いた．

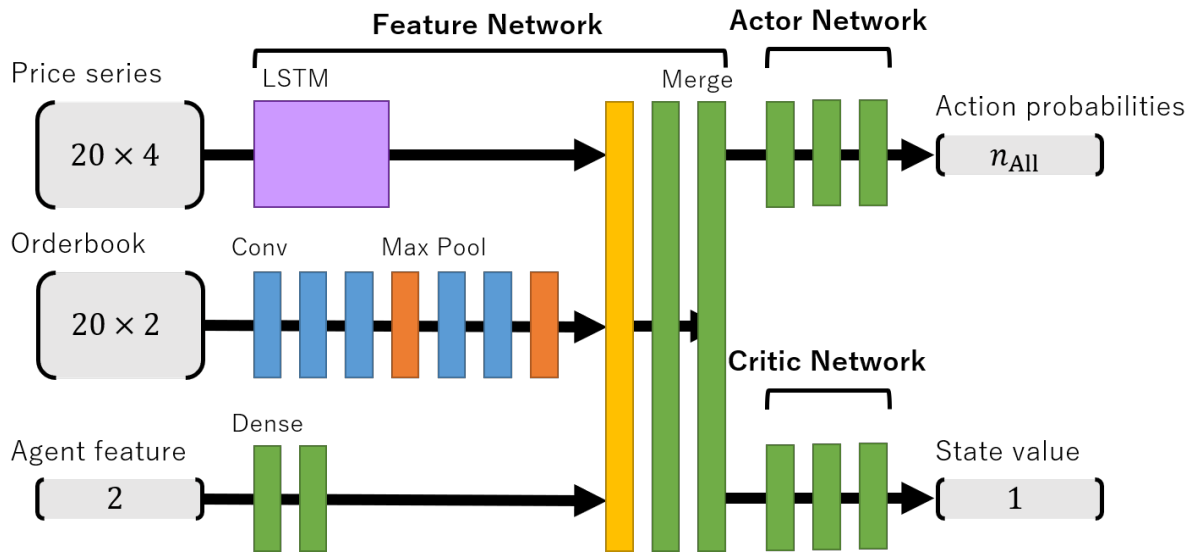


図 2.3: A2C モデルのネットワーク構成．本実験で用いた 3 つの特徴量 Price series，Orderbook features，Agent features からそれぞれ LSTM 層，Convolutional (Conv) および Maximum pooling (MaxPool) 層，Fully-connected (Dense) 層で特徴抽出を行い，特徴量を結合後 Fully-connected (Dense) 層で方策関数 (Action probabilities) および状態価値関数 (State value) を予測する．LSTM 層の活性化関数には tangent hyperbolic 関数を，Conv および Dense 層の活性化関数には ReLU 関数を用いた．

2.2.6 報酬関数の設計

学習に用いる報酬関数 (reward function) は，学習効率や学習モデルの性能に対し非常に大きく影響する．再掲するが，各 エージェント の目的は以下に示す資産 cap の最大化である．

$$cap = cash + \sum_i p_{\mathrm{Mid},i} pos_i \tag{2.15}$$

したがって，資産 cap の最大化を目指す報酬関数 R_{CO} (capital optimization reward) は以下のように計算される．

$$R_{\mathrm{CO}} = cap_{t+\tau} - cap_t \tag{2.16}$$

$$cap_{t+\tau} = cash_{t+\tau} + \sum_i p_{\mathrm{Mid},i,t+\tau} pos_{i,t+\tau} \tag{2.17}$$

$$cap_t = cash_t + \sum_i p_{\mathrm{Mid},i,t+\tau} pos_{i,t} \tag{2.18}$$

ここで，t および τ は行動時刻および時定数である．

以上のようにして計算した R_{CO} は直感的であるが，問題点も存在する．はじめに，注文が約定しなかった場合 $cash$ および pos が変化しないため，報酬関数の正負が仲値のみに依存してしまう．加えて，R_{CO} の計算で用いる資産 $cap_{t+\tau}$ および cap_t は株式保有量の絶対値を考慮しないため，大きなポジションをとることを許容してしまう．例として，100 円の株を 100 株，現金を 10,000 円持っている状態と，現金を 20,000 円持っている状態では cap の値は同一 $(10000 + 100 \times 100 = 20000)$ となる．しかし，一般的に株取引には取引コスト (あるいはスプレッド) が存在するため，その時点においては後者のほうが望ましい状態と言える．したがって cap の計算に取引コストを考慮することでより妥当な報酬が算出できると考えられる．取引コストの計算については多くの研究がなされているが [113, 114, 115]，本実験を行うシミュレーション環境において一意に計算することは難しい．そこで本研究では orderbook の情報のみから計算できる方法として，現在のポジションをすべて成行注文で解消した場合の資産 liquidation value LV_b を導入した．

Liquidation value を用いた報酬関数 R_{LV} は以下のように計算される．

$$R_{\mathrm{LV}} = LV_{t+\tau} - LV_t \tag{2.19}$$

ここで，もしすべてのポジションを解消できない場合，penalty price で解消することとする．すなわち，正のポジションについては価格 0 で，負のポジションについては市場価格の 2 倍で取引を行う．ただし，現実的にはそのような場合は殆ど起こらない．Liquidation value はポジションに対して大きなペナルティが発生するため，R_{LV} もリスク回避的な行動を推奨する報酬関数であると考えられる．

学習時においては，R_{CO} および R_{LV} は学習バッチ内の報酬を用いて以下のように正規化される．

$$R'_i = \frac{R_i}{\sqrt{E[R^2]}} \tag{2.20}$$

また，実現不可能な行動 (自身の注文が orderbook 上に存在しない状態での CXL 等) が選択された場合には，正規化された報酬に penalty reward R_p を加算する．本実験においては $R_p = -0.5$ とした．

2.2.7 シミュレータの実装

シミュレータの実装の概要を図 2.4 に示す．本実験で用いたシミュレータは，人工市場シミュレーションを行う client と，深層強化学習モデルを保存する server に二分される．Client においては Java [1] で simulator が記述されており，シミュレーションの進行が行われる．Simulator からは 1 ステップごとに iterativeDataJson.txt に市場の状態が出力される．深層強化学習モデルは python [2] のウェブアプリケーションフレームワーク Flask [3] で作成したウェブアプリケーション上に保存されている．

予測時の処理手順を図 2.5 に示す．Java 上で進行する人工市場シミュレーションにおいて DRL エージェントに行動選択権が与えられた場合，http 通信を用いて flask server 上の予測関数が呼び出される．DRL モデルの予測関数を呼び出す URL は flask-url/predict のように定められている．http 通信を用いて呼び出された flask server は iterativeDataJson.txt からシミュレーション履歴を読み取り説明変数を作成，DRL モデルに入力し予測結果から選択行動を決定する．選択行動は client 側からの http 通信のレスポンスとして送信され，DRL agent の行動からシミュレーションが再開される．DRL モデルが予測を行う際 client 側は返信待ち状態となるため，計算時間の関係上 DRL エージェントの行動処理がシミュレーションの律速段階となる．また，学習は行動選択時点から一定期間経過後に得られる報酬関数を用いて行う必要があり，それに伴い予測時の市場状態および行動選択結果を DRL モデルが保持している必要がある．DRL モデルの予測結果は flask server および predictionDataJson.txt に保存される．

学習時の処理手順を図 2.6 に示す．DRL モデルの学習は人工市場シミュレーションにおける一定ステップごとに行われる．人工市場シミュレーションにおいて DRL モデルの学習ステップに到達した場合，http 通信を用いて flask server 上の学習関数が呼び出される．DRL モデルの学習関数を呼び出す URL は flask-url/train のように定められている．http 通信を用いて呼び出された flask server は iterativeDataJson.txt および predictionDataJson.txt からシミュレーション履歴および行動履歴を取得する．取得された履歴から説明変数，予測結果および報酬関数を抽出し，勾配を計算，DRL モデルのパラメータ更新を行い，http 通信のレスポンスを client に返す．DRL モデルの学習はシミュレーションと並行して行うことができるが，DRL モデルの学習はシミュレーション履歴を大きく過去に遡り参照

[1] https://java.com/
[2] https://www.python.jp/
[3] https://flask.palletsprojects.com/

するため計算量が大きく，次に予測行動でDRLモデルが呼び出されるまでに学習行動が完了しない恐れがあるため学習中においてシミュレーションは返信待ち状態としている．

本章のフレームワークにおいて，シミュレーションを並列に実行することは想定されていない．並列非同期学習については，2.3.3 章で実装を含め詳説している．

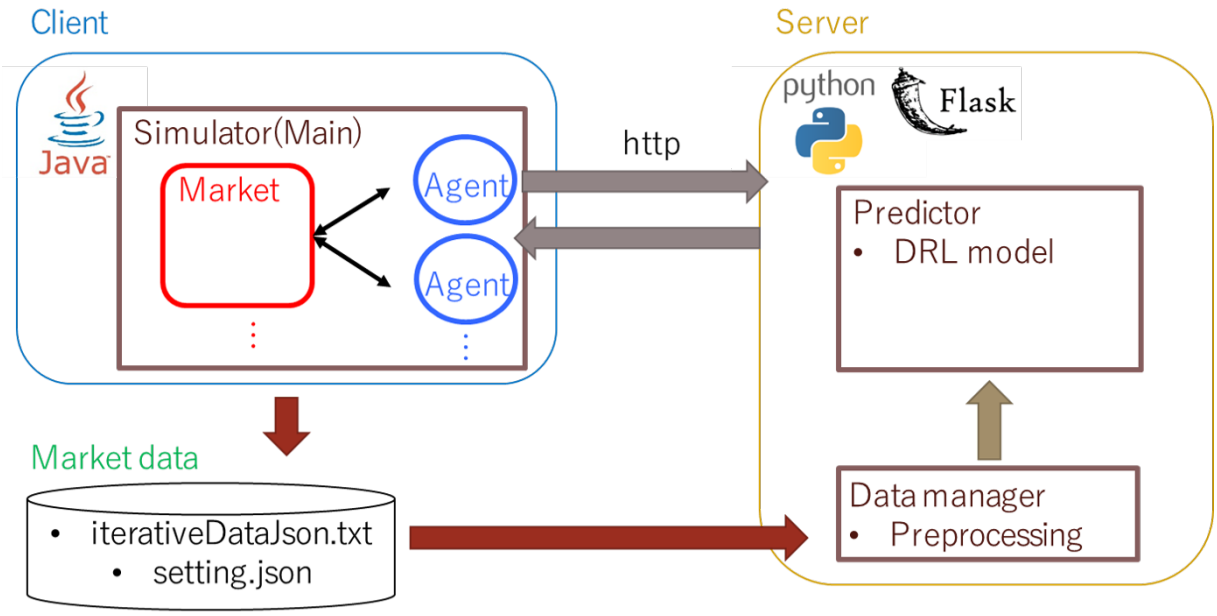


図 2.4: シミュレータ実装の概要図．本実験で用いたシミュレータは，人工市場シミュレーションを行う client と，深層強化学習モデルを保存する server に二分される．Client においては Java で simulator が記述されており，シミュレーションの進行が行われる．深層強化学習モデルは python のウェブアプリケーションフレームワーク Flask で作成したウェブアプリケーション上に保存されいる．

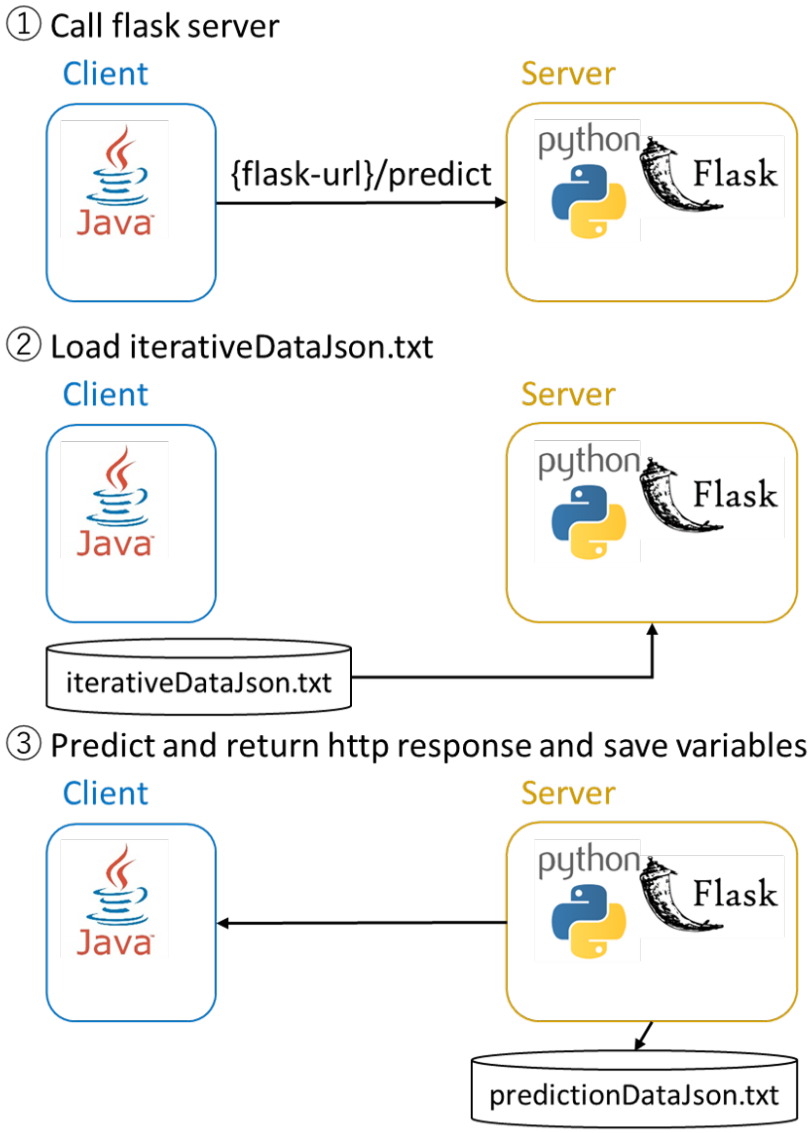


図 2.5: シミュレータにおける予測時の処理手順．Java 上で進行する人工市場シミュレーションにおいて DRL agent に行動選択権が与えられた場合，http 通信を用いて flask server 上の予測関数が呼び出される．http 通信を用いて呼び出された flask server は iterativeDataJson.txt からシミュレーション履歴を読み取り説明変数を作成，DRL モデルに入力し予測結果から選択行動を決定する．DRL モデルの予測結果は flask server および predictionDataJson.txt に保存される．

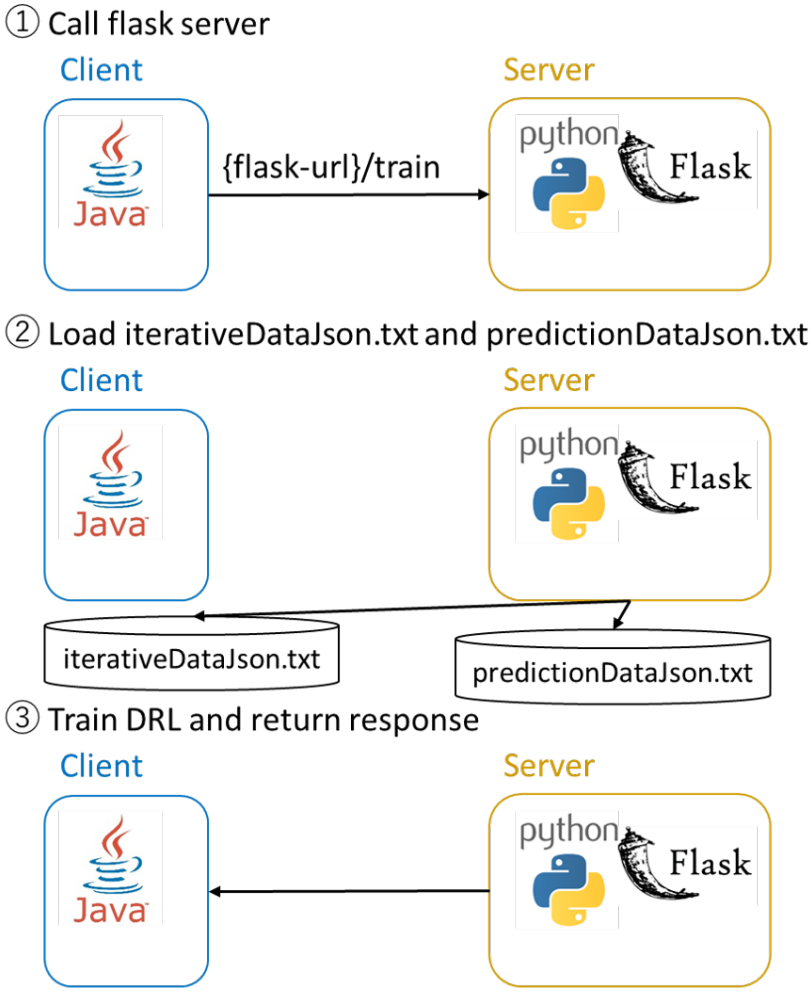


図 2.6: シミュレータにおける学習時の処理手順．DRL モデルの学習時，http 通信を用いて flask server 上の学習関数が呼び出される．http 通信を用いて呼び出された flask server は iterativeDataJson.txt および predictionDataJson.txt からシミュレーション履歴および行動履歴を取得する．取得された履歴から説明変数，予測結果および報酬関数を抽出し，勾配を計算，DRL モデルのパラメータ更新を行い，http 通信のレスポンスを client に返す．

2.2.8　実験条件

前章までで解説したフレームワークを用いて実験を行った．本実験においては以下の 2 点について確認することを目指す．

1. 設計したシミュレータで深層強化学習モデルの学習が行えるか
2. 学習した取引戦略が有効かどうか

2.2.5 章の DRL モデルを 2.2.6 章で解説した capital-only (CO) reward および liquidation-value (LV) reward を用いて学習し，その後学習済みモデルの検証のため DRL モデルの学習を伴わないシミュレーションを行った．DRL の学習を行うシミュレーションを training simulation，検証を行うシミュレーションを validation simulation と呼ぶ．

実験のパラメータ設定を表 2.1 に示す．Validation simulation においては，epsilon-greedy を用いたランダムな行動選択は行わない．Training simulation におけるランダム行動確率 ϵ は 0.2 とした． Training simulation および validation simulation はそれぞれ 100 回の sub simulation で構成され，各 sub simulation は 1,000,000 ステップの エージェント 行動および market の更新で構成される．したがって，training simulation および validation simulation はそれぞれ合計 100,000,000 ステップで構成される．各 sub simulation はそれぞれ半日間の市場の値動きを想定している．

各シミュレーションごとに，シミュレーションのハイパーパラメータおよび 1000 体の stylized agent のパラメータをサンプリングし実験を行う．Stylized agent においては，各 agent において対数リターン計算式 2.3 式の重量パラメータ w_F，w_C，w_N をサンプリングする必要がある．重量パラメータは指数分布からサンプリングされ，指数分布のスケールパラメータ σ_F，σ_C，σ_N は各 sub simulation ごとに正規分布からサンプリングされる．加えて，時定数 T および order margin k は一様分布からサンプリングを行った．

指数分布のスケールパラメータ σ_F，σ_C，σ_N

のパラメータは正規分布および一様分布からサンプリングを行った．σ_F，σ_C，σ_N は stylized agent の対数リターン計算式 2.3 式の重量パラメータ w_F，w_C，w_N をサンプリングする指数分布のスケールパラメータであり，サンプリングされた σ_F，σ_C，σ_N を用いて 1000 体の stylized agent の w_F，w_C，w_N のサンプリングを行った．Stylized agent の各パラメータにおいては以下の分布からサンプリングを行った．また，stylized agent

と DRL agent の行動確率を揃えるため，各ステップでランダム選択された stylized agent の行動確率を 0.5 とした．すなわち，選択された stylized agent は 0.5 の確率で行動し，0.5 の確率で行動せずに次の候補 エージェント が選択される．

加えて，DRL agent の各 sub simulation 開始時における保有株式量 x_0 を正規分布からサンプリングし，初期資産が一定となるよう初期現金を設定することで，初期条件のランダム性を導入した．

および 現金 $cash_0$ は以下のように計算し，初期資産一定の条件下でランダム性を導入した．

表 2.1: 人工市場における深層強化学習モデル学習の実験条件．なお，$\mathcal{N}$ は正規分布を，$\mathcal{U}$ は一様分布を表している．また，p_0 は市場価格の初期値である．

Category	Parameter	Value or distribution
Simulation	# sub simulation (training)	$= 100$
	# sub simulation (validation)	$= 100$
	# step in sub simulation	$= 1,000,000$
	Random action prob $\epsilon_{\text{training}}$	$= 0.2$
	Random action prob $\epsilon_{\text{validation}}$	$= 0$
Market	Initial fundamental price	$\sim \mathcal{N}(500, 50^2)$
	Fundamental volatility	$\sim \mathcal{N}(10^{-4}, (10^{-5})^2)$
Stylized agent	σ_F	$= 1$
	σ_C	$\sim \mathcal{N}(0.3, 0.03^2)$
	σ_N	$\sim \mathcal{N}(0.3, 0.03^2)$
	Time scale T	$\sim \mathcal{U}(500, 1000)$
	Order margin k	$\sim \mathcal{U}(0, 0.05)$
	Action probability	0.5
	Time scale τ	1000
DRL agent	Initial position x_0	$\sim \mathcal{N}(0, 10^2)$
	Initial cash m_0	$= 35000 - p_0 x_0$
	Time scale T	$= 1000$

2.2.9 実験結果

実験結果を以下に示す．本実験では比較として，DRL agent と同様の行動候補からランダムで行動選択を行うモデルについても実験を行った．モデルの性能および投資実績は average reward $\overline{R}$, Sharpe ratio S_p, maximum drawdown MDD を用いて評価を行う．Average reward は sub simulation 内の全行動に対する報酬関数の平均値である．Sharpe ratio S_p は以下のように計算される．

$$S_p = \frac{E[R_a - R_b]}{\sqrt{V[R_a - R_b]}} \tag{2.21}$$

ここで，R_a および R_b は投資およびベンチマークのリターンを，E および V は期待値および分散を表す．本実験ではベンチマークのリターン R_b は 0 とする．Sharpe ratio はどれだけ安定的にリターンを出せるかを評価する指標と解釈できる．最後に，maximum drawdown MDD [116, 117] は以下のように計算される．

$$MDD = \frac{P - L}{P} \tag{2.22}$$

ここで，P および L はシミュレーション期間で最も大きい損失を出した前後における資産の値である．MDD は損失の大きさを評価する指標として使用され，小さいほど安全な投資が行えていることがわかる．

Training simulation および validation simulation における average reward の推移を図 2.7 および図 2.8 に示す．Random が比較手法のランダム行動選択モデル，CO-DRL が capital-only reward R_{CO} を用いて学習した DRL モデル，LV-DRL が liquidation-value reward R_{LV} を用いて学習した DRL モデルを表す．ランダム行動選択モデルの報酬関数には R_{CO} を用いた．

図 2.7 を見ると，学習初期において Random および CO-DRL の averege reward は 0 付近であるのに対し，LV-DRL では負の値となっている．2.2.6 章で述べたとおり，liquidation-value reward R_{LV} はエージェントのポジションに対し厳しく資産が計算されるため，学習が不十分な状態での報酬の平均値は低くなると考えられる．しかし sub simulation の回数が増えるにれ，LV-DRL モデルの average reward は大きく上昇し，20 sub simulations 程度進行後には 0.5 程度と，報酬関数を正規化していることを考慮すれば非常に良好な成

績を残していることがわかる．一方，CO-DRL の average reward は学習の進行につれわずかに上昇しているものの，LV-DRL と比べると非常に低い値で推移しており，学習が効率的に行えていないことがわかる．以上より，本研究で提案した liquidation-value reward R_{LV} が一般的な capital-only reward R_{CO} に比べ学習効率を飛躍的に高めたことがわかる．

一方，図 2.8 に示す validation simulation では，DRL モデルの予測時に epsilon-greedy を適用していないため，training simulation に比べ CO-DRL および LV-DRL モデルの average reward の値がわずかに改善している．Validation simulation においても，sub simulation ごとに上下動は存在するものの全体に LV-DRL が良好な成績を残しており，sub simulation ごとに設定した多様な市場条件下でも学習した戦略が有効であることが確認された．

Training simulation および validation simulation における評価指標の計算結果を表 2.2 に示す．モデル性能は前述の average reward $\overline{R}$, Sharpe ratio S_p, maximum drawdown MDD で評価している．各評価指標は sub simulation ごとに計算されており，training simulation，validation simulation それぞれ 100 回の sub simulation の平均および標準偏差が記載されている．表 2.2 を見ると，3 つの指標全てにおいて LV-DRL モデルが非常に良い成績を残している．LV-DRL モデルは Sharpe ratio が高いのに加え Maximum drawdown が小さく，損失を回避しながら安定的に利益を上げていることがわかる．なお，training simulation において LV-DRL モデルの Maximum drawdown が大きくなっているのは，学習初期に非常に大きな損失を出しているためであると考えれられる．

各モデルの投資例を図 2.9 および図 2.10 に示す．図 2.9 は validation simulation における 9 回目の sub simulation の結果を，図 2.10 は validation simulation における 4 回目の sub simulation の結果を示しており，図には仲値を元に計算した資産 (Capital)，保有株式量 (Position)，仲値 (mid price) の変動がプロットされている．

図 2.9 の例は LV-DRL が非常に優秀な成績を示している例であり，緑色の線で表される LV-DRL は安定して capital が上昇している．また，LV-DRL は序盤以降は position が 0 付近で推移しており，リスクをとらずに資産を増やすことに成功している．対して Random および CO-DRL は position が大きく上下動しており，CO-DRL については資産は現状維持，Random については取引コストにより徐々に資産が下降している．また，capital の振れ幅についても position の絶対値が大きい分大きくなっており，リスクの高い行動を選択していることがわかる．なお，sub simulation 序盤について LV-DRL の position が小さくなっていることについては，DRL モデルの学習が不十分であることが原因であると

考えられる．

一方，図 2.10 では CO-DRL が優秀な成績を残している．LV-DRL が 図 2.9 同様安定して利益を挙げているのに対し，CO-DRL は仲値の上昇にあわせ急速に資産を増やしている．しかし，CO-DRL は sub simulation 全体で終始 position を増加させており，capital の上下動の大きさからも分かる通り，非常にリスクの高い投資行動を行っていることがわかる．このように，実際の投資行動の様子を見ても LV-DRL が優れた戦略を学習できていることがわかる．

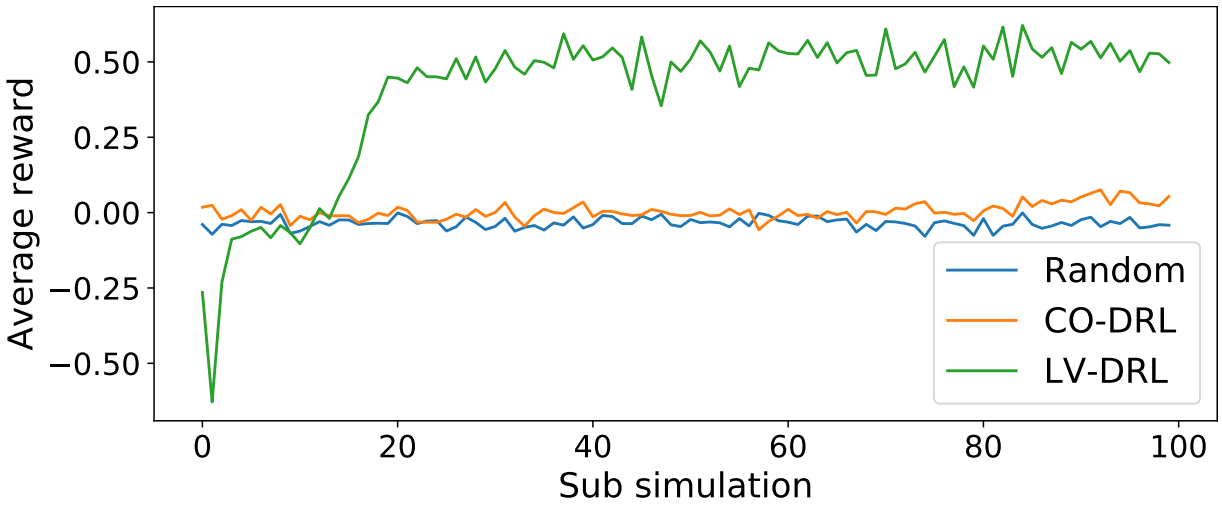


図 2.7: Training simulation における average reawrd $\overline{R}$ の推移．Random が比較手法のランダム行動選択モデル，CO-DRL が capital-only reward R_{CO} を用いて学習した DRL モデル，LV-DRL が liquidation-value reward R_{LV} を用いて学習した DRL モデルを表す．

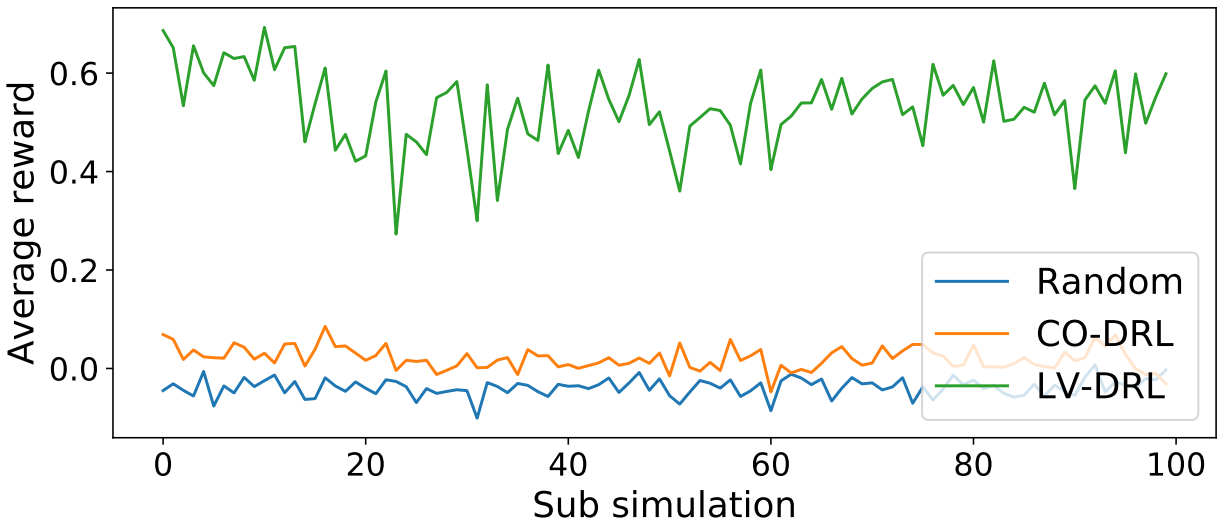


図 2.8: Validation simulation における average reawrd $\overline{R}$ の推移．Random が比較手法のランダム行動選択モデル，CO-DRL が capital-only reward R_{CO} を用いて学習した DRL モデル，LV-DRL が liquidation-value reward R_{LV} を用いて学習した DRL モデルを表す．

表 2.2: ランダム行動選択モデル，CO-DRL モデル，LV-DRL モデルのモデル性能. 各手法の性能は $\overline{R}$，S_p，MDD で評価される．各指標は sub simulation ごとに計算され，平均 ± 標準偏差の形で記載されている．

Simulation	Model	$\overline{R}\uparrow$	$S_p\uparrow$	$MDD\downarrow$
Training	Random	-0.035 ± 0.017	0.005 ± 0.005	0.111 ± 0.143
	CO-DRL	0.004 ± 0.025	0.009 ± 0.006	0.144 ± 0.196
	LV-DRL	0.402 ± 0.236	0.043 ± 0.023	0.049 ± 0.137
Validation	Random	-0.038 ± 0.018	0.005 ± 0.007	0.110 ± 0.104
	CO-DRL	0.020 ± 0.023	0.009 ± 0.006	0.124 ± 0.178
	LV-DRL	0.530 ± 0.079	0.053 ± 0.019	0.012 ± 0.004

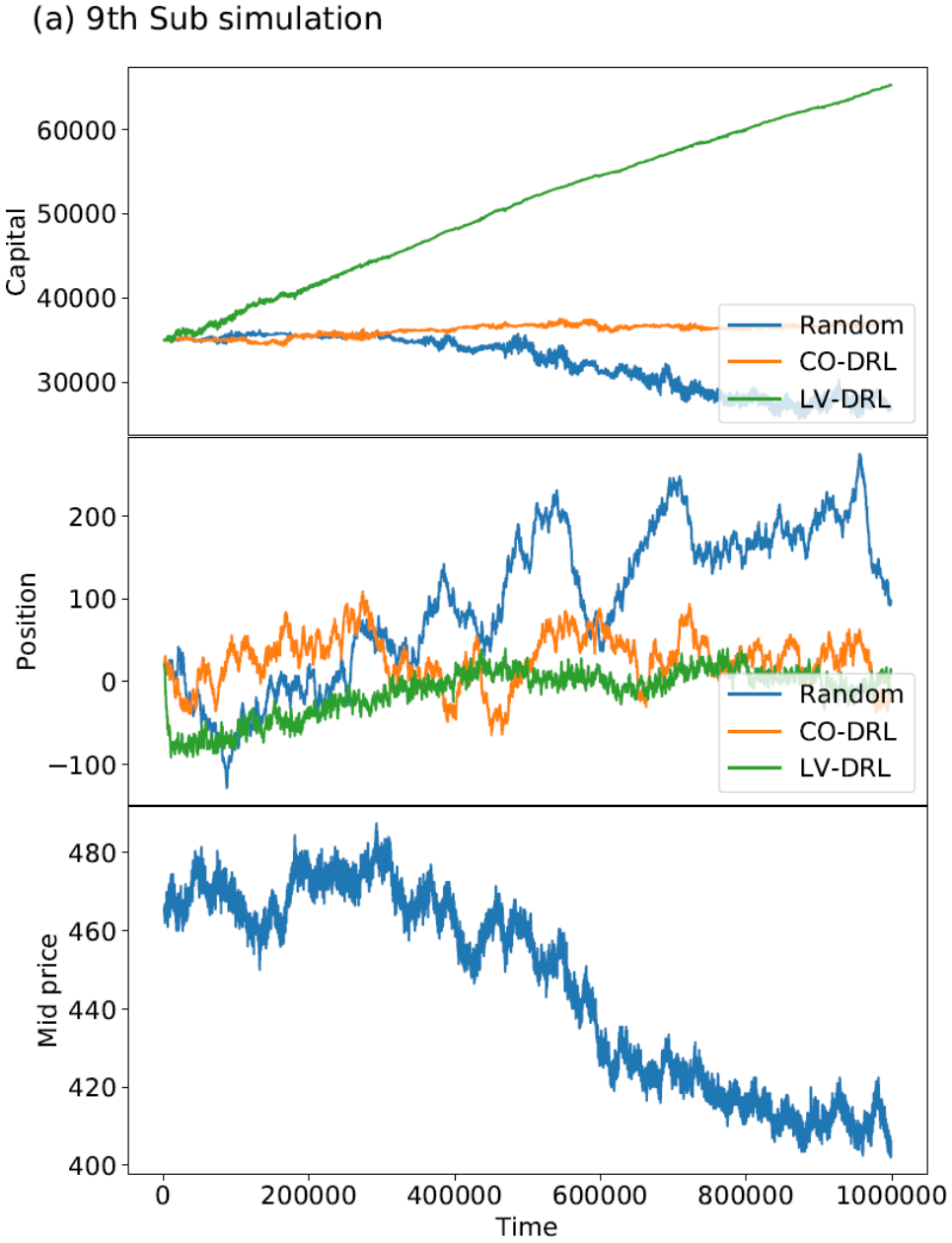


図 2.9: ランダム行動選択モデル，CO-DRL モデル，LV-DRL モデルの行動例．Validation simulation おける 9 回目の sub simulation の結果を示しており，上から順に資産 (Capital)，保有株式量 (Position)，仲値 (Mid price) の推移をプロットしている．

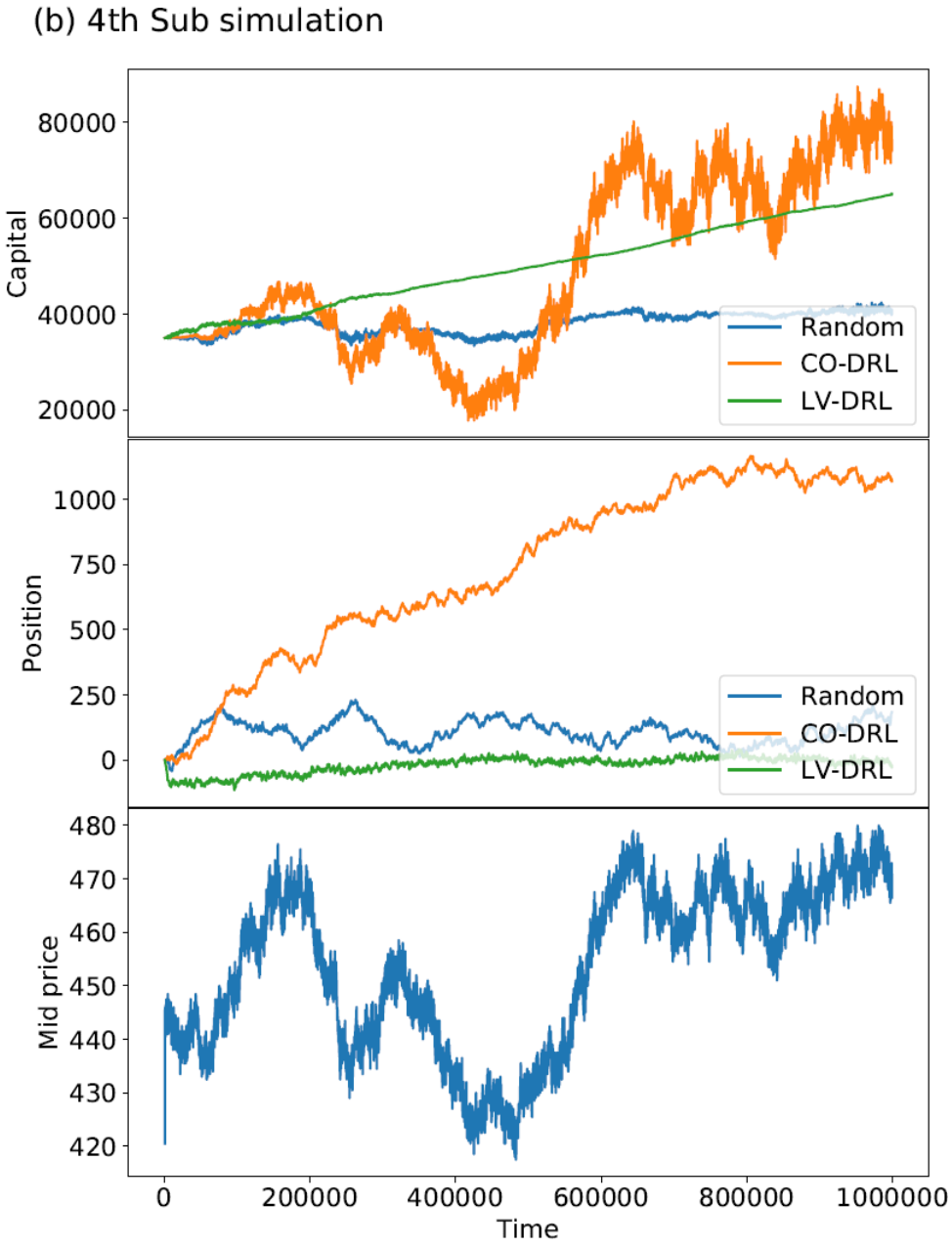


図 2.10: ランダム行動選択モデル，CO-DRL モデル，LV-DRL モデルの行動例．Validation simulation おける 4 回目の sub simulation の結果を示しており，上から順に資産 (Capital)，保有株式量 (Position)，仲値 (Mid price) の推移をプロットしている．

2.2.10　学習戦略の妥当性

LV-DRL が学習した投資戦略についてより詳細な考察を行うため，選択可能な行動の選択頻度分布を比較した．結果を図 2.11 に示す．行動は”Action-Type-Price-Volume”の形式で表示されている．例として，”Stay”は何もしない行動 (Do-nothing)，”Buy-LMT-2.0-1.0”は買い指値注文で best ask より 2 価格分安い価格，1 数量の注文行動を，”Sell-CXL-Low-”は売り注文板の最小価格注文に対する取り消し注文を表している．

図 2.11 を見ると，Random についてはその設計通り全行動をほぼ同確率で選択している．CO-DRL については CXL を行う確率が小さく学習されているものの，その他の行動については大差なく，有効な行動戦略が学習できていないことがわかる．対して LV-DRL では行動選択に顕著な傾向が現れている．LV-DRL で特に選択確率が低くなっている”Buy(Sell)-LMT-0.0-5.0”および”Buy(Sell)-MKT–5.0”は大きな数量で即約定を行うような注文行動であり，一般的に自身に不利な行動であるため選択頻度が低くなっていることが考察される．対して LV-DRL モデルが多く選択している”Buy(Sell)-4.0(6.0)-5.0”は，比較的に自身に有利な価格 (買い注文の場合は買い最良気配，売り注文の場合は売り最良気配付近) での大きな数量の注文であり，利益を見込める注文行動が優先的に選ばれていることがわかる．概して LV-DRL は「自身に有利な価格で大きな数量の注文を，自身に不利な価格で小さな数量の注文を行う」ような，人間の目で見ても戦略を学習できていることがわかる．

さらに詳細な行動パターンとして，連続する 2 行動および 3 行動について頻度分布を作成し比較した．結果を図 2.12 および図 2.13 に示す．図の横軸が行動系列を表し，行動に対応する数字は図 2.11 の横軸に示される行動に左から 0, 1, と振った場合の番号を表している．例として．図中に多く見られる行動は以下である．

- 17: Buy-LMT-4.0-5.0 (売り最良気配値から 4 単位安い価格への数量 5 の買い注文)
- 18: Sell-LMT-4.0-5.0 (買い最良気配値から 4 単位高い価格への数量 5 の売り注文)
- 19: Buy-LMT-6.0-5.0 (売り最良気配値から 6 単位安い価格への数量 5 の買い注文)
- 20: Sell-LMT-6.0-5.0 (買い最良気配値から 6 単位高い価格への数量 5 の売り注文)

図 2.12 および図 2.13 を見ると，特徴的な行動パターンは見られず，単純に図 2.11 の頻度分布のかけ算に近いように見える．本実験においては一度の行動で 1 注文のみしか出

すことができないため，実市場の金融市場 making [118, 119] のような戦略を行うには連続する行動パターンを確立する必要があるが，本実験においてはそのような学習は見られなかった．これは，本実験で用いた報酬関数が非常に短期の利益を元に計算されており，かつ行動 エージェント がランダムに選択されるため短時間に複数の注文行動を連続して行うことが難しい環境であることが原因だと考えられる．

DRL agent の学習が予想される基本的な投資戦略として,，trend follower [120] が挙げられる．Trend follower は価格のトレンドに追随した投資行動を行う．DRL agent が trend follower の行動戦略を行う場合，価格トレンドが売り買い，注文板の情報が注文数量に影響を与えると予想される．しかし本実験では，価格トレンドや注文板のスプレッドと注文行動の明確な相関関係は見られなかった．本研究で提案した liquidation value reward は非常にリスク回避的な報酬関数であり，DRL agent の行動には自身の株式保有量に基づくリスク回避の観点が多分に含まれていると考えられる．また，実験条件や説明変数，報酬関数の設計，異なるアルゴリズムで行動する環境エージェントの導入により，より高度な注文戦略を学習することが可能になると期待される．

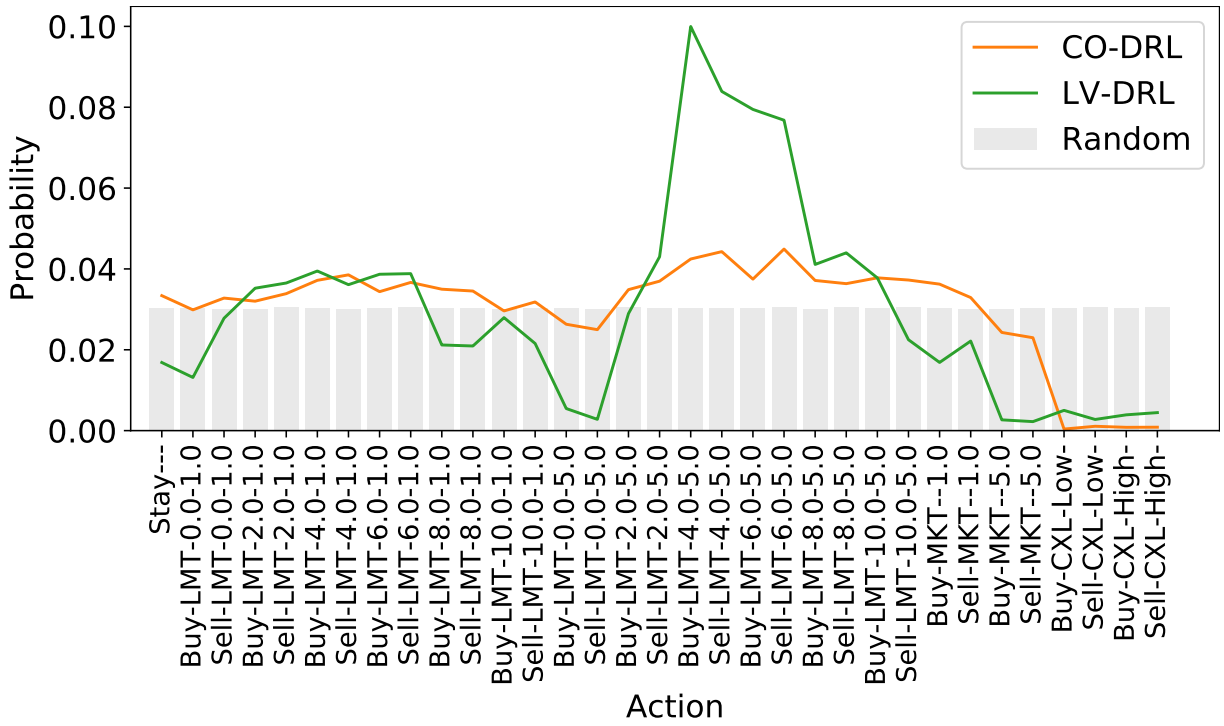


図 2.11: ランダム行動選択モデル，CO-DRL モデル，LV-DRL モデルの行動選択頻度分布．行動は”Action-Type-Price-Volume”の形式で表示されており，validation simualtion 全体における各モデルの選択頻度を表示している．

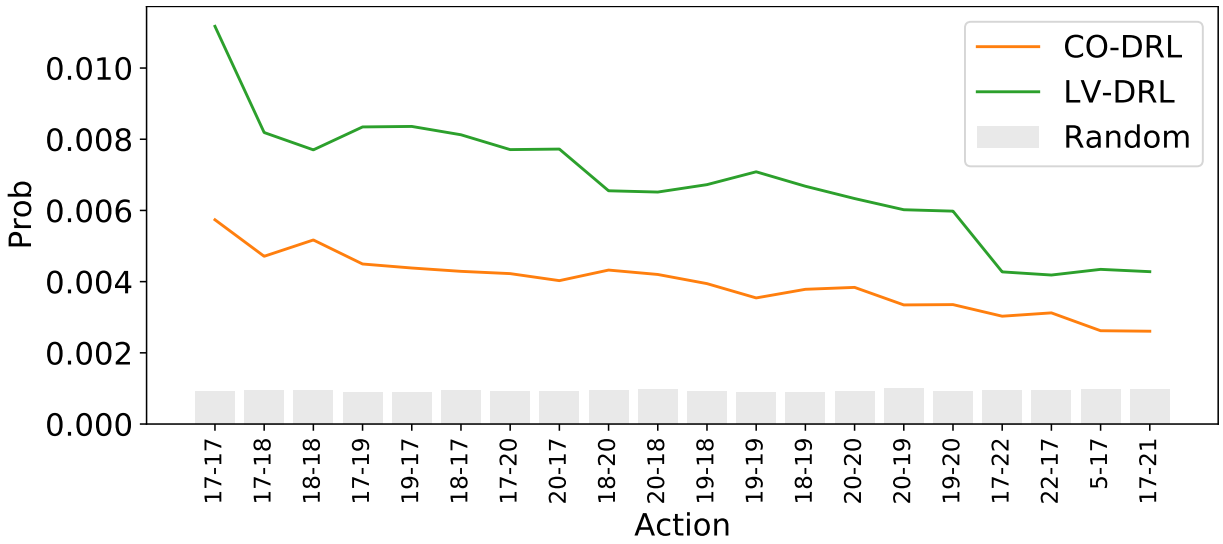


図 2.12: ランダム行動選択モデル，CO-DRL モデル，LV-DRL モデルについての，連続する 2 行動の頻度分布．横軸が行動系列を表し，行動系列の数字は図 2.11 の横軸に示される行動に左から 0, 1, と振った番号を表す．

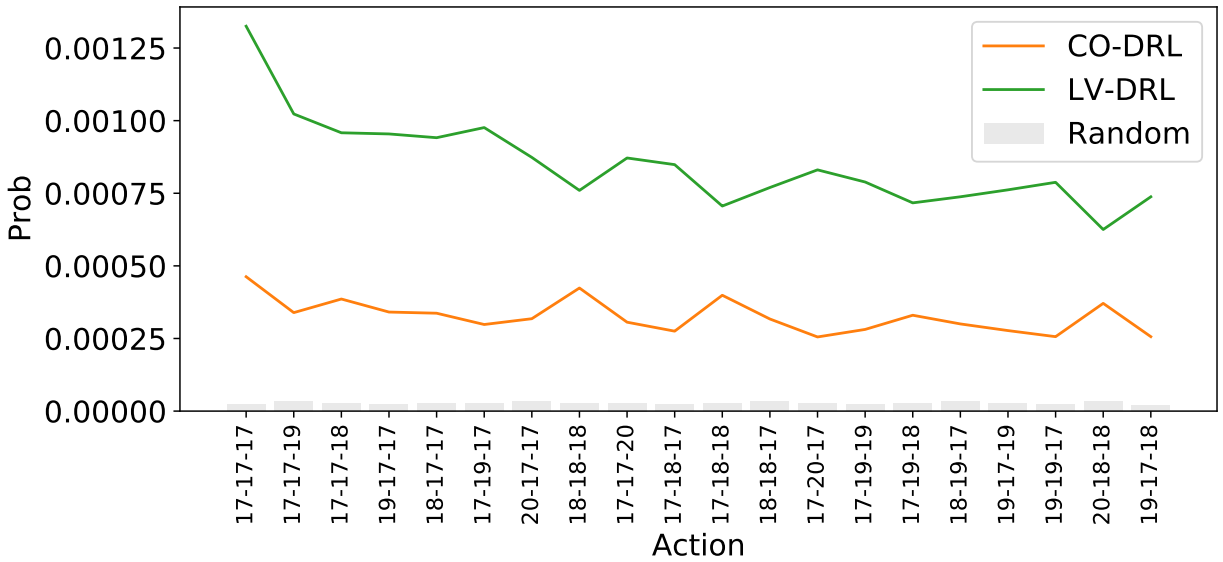


図 2.13: ランダム行動選択モデル，CO-DRL モデル，LV-DRL モデルについての，連続する 3 行動の頻度分布．横軸が行動系列を表し，行動系列の数字は図 2.11 の横軸に示される行動に左から 0, 1, と振った番号を表す..

2.3　実験:深層強化学習手法の比較

2.2 章では，人工市場シミュレーション上で深層強化学習モデルを学習させるフレームワークの提案，実装，および検証を行った．本実験では，同フレームワークを用いた実験を通し，複数の深層強化学習手法，DQN および A3C の比較検討を行った．A3C の学習に際し，モデル学習プロセスの並列化も行っている．

本研究の成果は 2020 年度人工知能学会全国大会で発表している [121].

2.3.1　実験環境

実験で用いたシミュレータ，金融市場，エージェントは 2.2.1 章，2.2.2 章，2.2.3 章，2.2.4 章で解説したものと同様の構成のものを用いた．金融市場数は 1 とした. 各エージェントにはランダムな順番で行動機会が与えられ，市場状態を元に注文の有無および内容を決定する． エージェント の注文は market の orderbook (注文板) に登録され，連続双方向オークション (continuous double auction) の価格決定プロセスを用いて取引が行われる．シミュレータには environmental agent (環境エージェント) として stylized agent が 1000 体，学習エージェントとして DRL agent が 1 体登録される．DRL agent は 2.2.5 章同様，価格系列 (price series)，注文板状態 (orderbook features)，エージェント情報 (agent features) を入力とし，図 2.2 でも解説した以下の 5 つの項目を出力とする注文行動の離散化を行った．

1. 注文行動 (Action): Buy, Sell, Do-nothing
2. 注文の種類 (Type): LMT, MKT, CXL
3. 価格 (Price, LMT のみ): 仲値 0, 2, 4, 6, 8, 10 単位離れた価格
4. 数量 (Volume, LMT および MKT のみ) : 1 単位あるいは 5 単位
5. 取り消しする注文 (CXLOrder, CXL のみ): best あるは worst

以上の条件を図 2.2 の全行動数の計算式に代入すると，DRL の出力次元は以下のように計算される．

$$1 + 2 \times 1 \times (6 \times 2 + 2 + 2) = 33 \tag{2.23}$$

DRL モデルの学習に用いる報酬関数としては，2.2 章の実験結果を踏まえ，liquidation value reward R_{LV} (2.2.6 章) を用いた．

2.3.2 DRL モデルの構成

DRL モデルとしては，2.2 章で用いた A2C に並列非同期 (Asynchronous) 学習を取り入れた Asynchronous Advantage Actor-Critic (A3C) [71] を用いた．並列非同期学習の実装については 2.3.3 章で詳説している．

深層強化学習手法としては，前述の A3C に加え，Deep-Q Network (DQN) を採用し実験，比較を行った．DQN は方策ベースの A3C と異なり価値関数ベースの手法であり，各行動の行動価値関数 (期待報酬) をニューラルネットワークで近似する．行動選択時には行動価値関数の予測値が最も大きい行動が選択される．A3C および DQN のネットワーク構成を図 2.14 に示す．両モデルは出力層を除き同一のネットワーク構成を持ち，価格系列，注文板状態，エージェント情報の 3 つの説明変数から抽出された特徴量をもとに，DQN では行動価値関数，A3C では方策関数および状態価値関数が予測される．それぞれの説明変数から特徴抽出を行うネットワークは 2.2.5 章と同様に選択した．価格系列に対しては LSTM を，注文板情報に対しては Conv および MaxPool を，エージェント情報については Dense を用いて特徴抽出を行った．

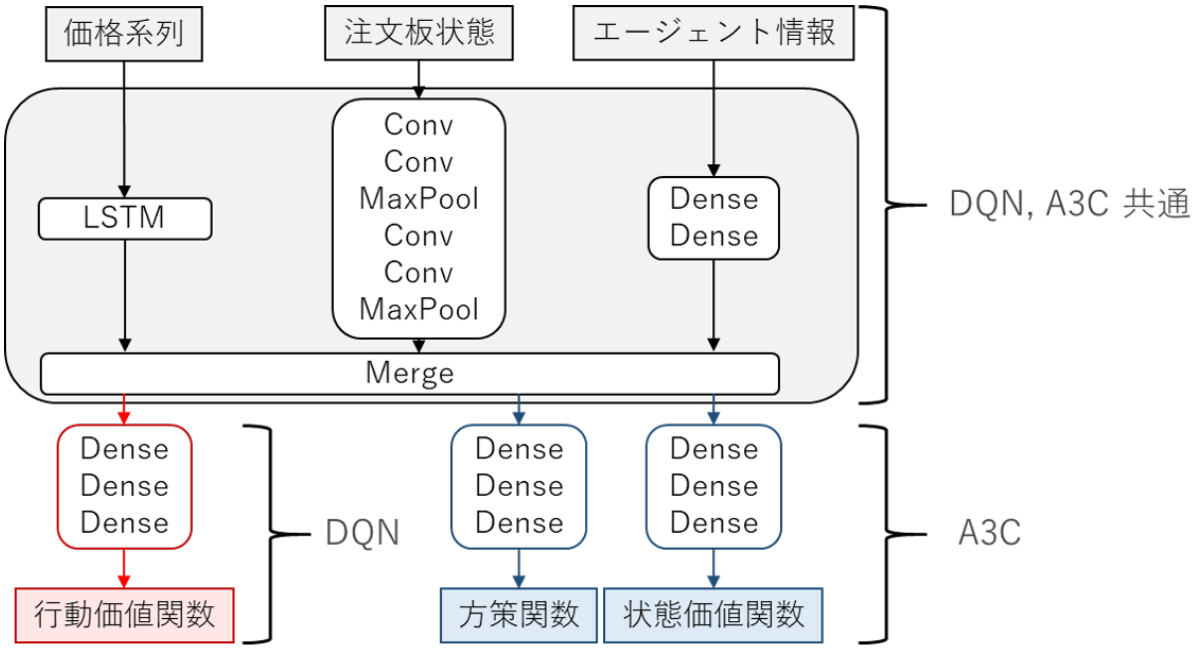


図 2.14: 深層強化学習手法比較実験で用いた A3C および DQN モデルのネットワーク構成．両モデルは出力層を除き同一のネットワーク構成を持つ．出力層においては，DQN では行動価値関数が， A3C では方策関数および状態価値観数が予測される．

2.3.3 並列非同期学習の実装

並列非同期学習の概要を図 2.15 に示す．並列非同期学習 [122, 123] においては，DRL モデルの最終的なパラメータを格納する DRL server および実際のシミュレーションで呼び出しを行う DRL worker を用いる．DRL sever および各 DRL worker は同一のニューラルネットワークを保持しており，各 server および worker のパラメータは別個の値となっている．人工市場シミュレーションを行う simulator はそれぞれ独立して設定しており，simulator と DRL worker は一対一対応している．

DRL モデル予測時については，各 simulator および worker で 2.2.7 章と同様の処理を行う．DRL モデル学習時における処理を図 2.16 に示す．学習メソッドは各シミュレーションにから呼び出されている，例として Simulator #1 から DRL worker #1 の学習メソッドが呼び出された場合，2.2.7 章の処理同様，iterativeDataJson.txt および predictionDataJson.txt から履歴を参照，損失関数を算出し，DRL worker #1 のパラメータ θ_1 についての勾配 $\partial\theta_1$ を計算する．その後，勾配 $\partial\theta_1$ を用いて DRL server のパラメータ θ を更新する．

$$\theta \leftarrow \theta - \alpha\partial\theta_1 \tag{2.24}$$

ここで，α は係数である．Adam [124] や RMSprop [125] 等の optimizer [126, 127] を用いる場合，この更新式は異なるものとなる．DRL server のパラメータを更新した後，DRL server のパラメータ θ を DRL worker #1 のパラメータ θ_1 にコピーする．以上のような処理を各 simulator および worker について実行することで，並列非同期な学習が可能となる．並列非同期学習においては，勾配計算時の worker のパラメータと server のパラメータが異なるため，適切な勾配がパラメータ勾配が算出されない危険性がある．しかし，並列にシミュレーションを実行し，異なる環境下における勾配を並列にパラメータ更新に用いることで学習の偏りが軽減されるという報告もあり [122, 128]，一般的には効率的な学習過程として知られている．本実験では 10 並列で学習を行っている．

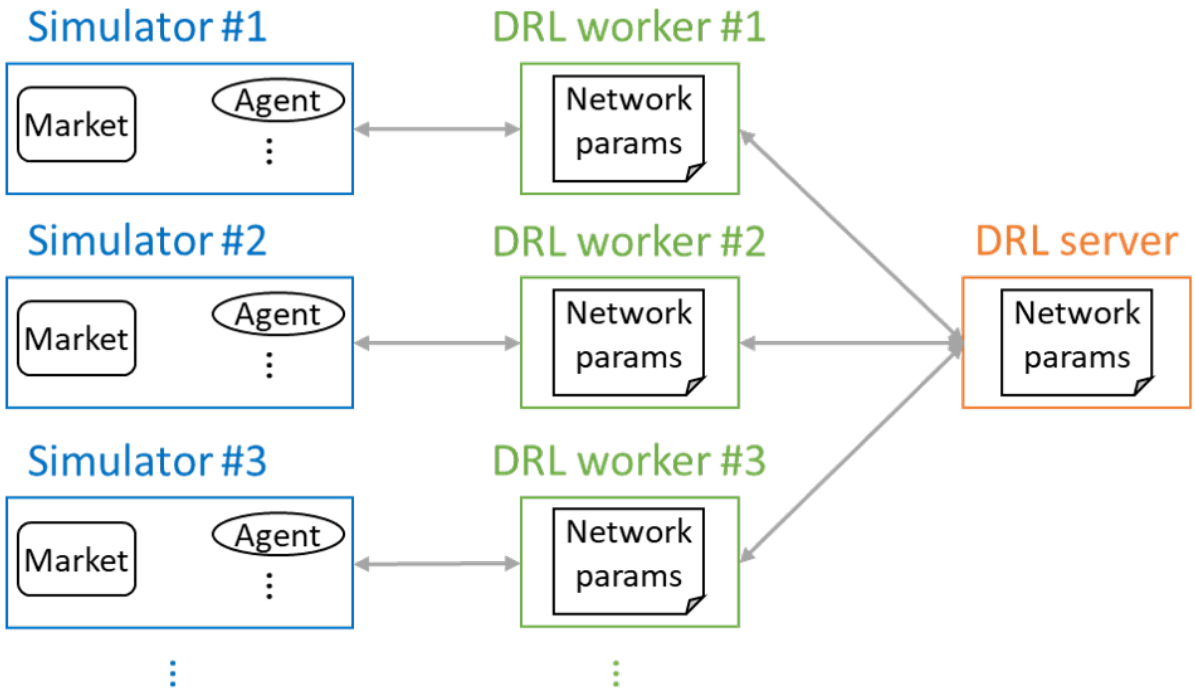


図 2.15: 深層強化学習手法比較実験で用いた並列非同期 (asynchronous) 学習の概要．並列非同期学習においては，DRL モデルの最終的なパラメータを格納する DRL server および実際のシミュレーションで呼び出しを行う DRL worker を用いる．DRL sever および各 DRL worker は同一のニューラルネットワークを保持しており，各 server および worker のパラメータは別個の値となっている．

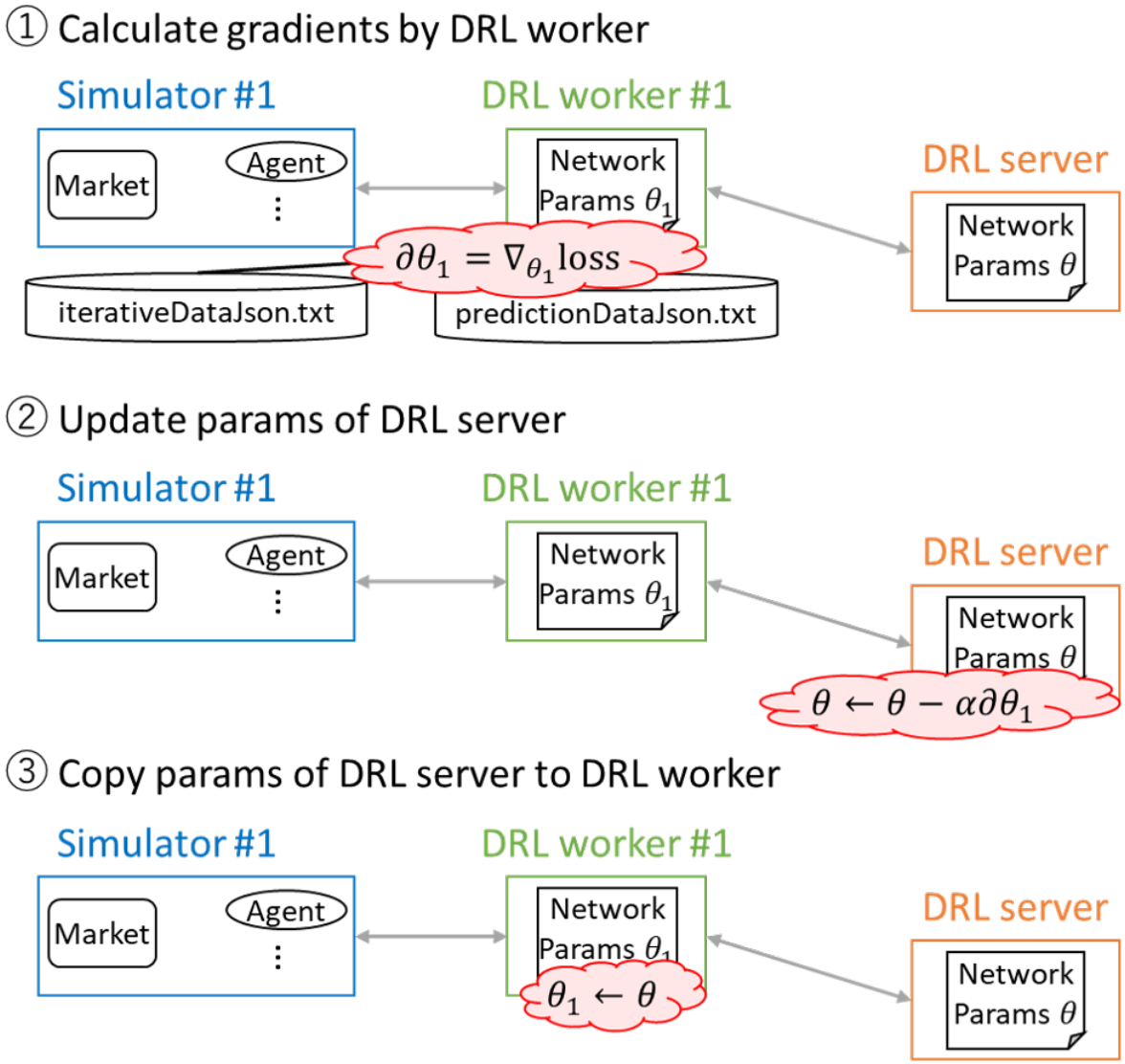


図 2.16: DRL モデル学習時における並列非同期処理の概要．simulator #1 から DRL worker #1 の学習メソッドが呼び出された場合，2.2.7 章の処理同様，iterativeDataJson.txt および predictionDataJson.txt から履歴を参照，損失関数を算出し，DRL worker #1 のパラメータ θ_1 についての勾配 $\partial\theta_1$ を計算する．その後，勾配 $\partial\theta_1$ を用いて DRL server のパラメータ θ を更新する．DRL server のパラメータ更新後，DRL server のパラメータ θ を DRL worker #1 のパラメータ θ_1 にコピーする．

2.3.4　実験条件

提案するシミュレータを用いて，DQN および A3C の学習および検証を行った．シミュレーションの 1 試行は エージェント の行動 100,000 ステップで構成され，training では 300 試行，validation では 100 試行のシミュレーションを行った．

設定するパラメータについては 2.2.8 章と同様である．パラメータ設定を 表 2.3 に示す．A3C および DQN モデルの学習および検証は前述の通り並列非同期で行う．本実験の並列ワーカー数は 10 とした．本実験では比較対象として，DQN および A3C と同様の行動候補からランダムに行動選択を行う zero-intelligence agent (ZI agent) [129, 130, 131, 132] についても実験を行った．

表 2.3: 人工市場における深層強化学習モデル学習の実験条件．なお，$\mathcal{N}$ は正規分布を，$\mathcal{U}$ は一様分布を表している．また，p_0 は市場価格の初期値である．

Category	Parameter	Value or distribution
Simulation	# sub simulation (training)	$= 300$
	# sub simulation (validation)	$= 100$
	# step in sub simulation	$= 1,00,000$
	Random action prob $\epsilon_{\text{training}}$	$= 0.2$
	Random action prob $\epsilon_{\text{validation}}$	$= 0$
Market	Initial fundamental price	$\sim \mathcal{N}(500, 50^2)$
	Fundamental volatility	$\sim \mathcal{N}(10^{-4}, (10^{-5})^2)$
Stylized agent	σ_F	$= 1$
	σ_C	$\sim \mathcal{N}(0.1, 0.01^2)$
	σ_N	$\sim \mathcal{N}(0.1, 0.01^2)$
	Time scale T	$\sim \mathcal{U}(500, 1000)$
	Order margin k	$\sim \mathcal{U}(0, 0.05)$
	Action probability	0.5
	Time scale τ	1000
DRL agent	Initial position x_0	$\sim \mathcal{N}(0, 10^2)$
	Initial cash m_0	$= 35000 - p_0 x_0$
	Time scale T	$= 1000$

2.3.5　実験結果

各手法の average reward $E[r]$ の変化をを図 2.17 および図 2.18 に示す．図には各 sub simulation 毎の average reward の変動が記載されており,，A3C は Asynchronous Advantage Actor-Critic を，DQN は Deep-Q Network を，ZI は zero-intelligence agent の結果を示している．ここで，本実験は並列非同期で学習および検証を行っており，基本的に sub simulation は番号の浅い順に実行しているものの，sub simulation の速度 (注文板上の注文量等で変動) によりその終了順に若干の誤差が存在する．そのような影響もあり前後の sub simulation と比較すると average reward の値が上下している場合が多いものの，傾向としては A3C および DQN モデルは図 2.17 に示す学習段階において sub simulation の進行とともに average reward の値が向上している．ここから両深層強化学習手法が適切に学習を行えていることがわかるが，両手法を比較すると DQN がより高速に学習を行えていることがわかる．対して A3C は緩やかに average reward の値が向上しており，学習を続けることでさらに実績が向上する可能性も考えられる．一方，図 2.18 に示す検証段階においては，ネットワークの学習を行っていないため，average reward の上昇傾向は存在しない．しかし，各 sub simulation の条件 (乱数) の違いにより実験結果が大きく上下している．検証段階については多くのケースにおいて DQN が A3C を上回る結果を残しているが，DQN は一部大きく成績を落としている区間があり，投資行動の安全性という側面では不安が残る結果となっている．両手法の差異については後に詳細に検証を行う．

実験結果を表 2.4 に示す．表は各手法の投資実績を示し，A3C は Asynchronous Advantage Actor-Critic を，DQN は Deep-Q Network を，ZI は zero-intelligence agent を表している．投資実績は average reward $E[r]$, Sharpe ratio S_p [133], maximum drawdown MDD [116, 117] で評価され，validation simulations 100 試行の平均および標準偏差が示されている．

表 2.4 を見ると，average reward および Sharpe ratio については DQN が非常に優秀な成績を示している．A3C も同様に優秀な average reward および Sharpe ratio の値を示しているが，DQN と比較するといずれも低い値となっている．ZI については average reward および Sharpe ratio が負の値となっており，行動により損失を出している．

一方，maximum drawdown については，A3C の 0.0015 ± 0.0007 に対し DQN は 0.0052 ± 0.0093 と，DQN が A3C の 3 倍近く大きな損失を出している．この結果から，DQN は平均としては大きな利益を挙げる一方，大きな損失を出すリスク伴う行動選択を行っている

ことがわかる．

より詳細な手法間の比較を行うため，validation simulations における投資結果例を図 2.19 および図 2.20 に示す．図 2.19 は validation simulation 5 試行目，図 2.20 は validation simulation 59 試行目の結果であり，A3C (Asynchronous Advantage Actor-Critic)，DQN (Deep-Q Network)，ZI (zero-intelligence agent) それぞれの capital (liquidation value) および inventory (証券保有量) の時間変化をプロットしている．

図 2.19 は A3C，DQN 両者ともに効果的な投資行動を行っている例である．両手法とも inventory が 0 付近で推移しており，大きなリスクを回避しながら順調に capital を増加させていっている．また，capital の増加率を比較すると DQN が A3C を上回る結果を残している．この結果は表 2.4 の average reward および Sharpe ratio の優劣と一致しており，効果的な投資が行えている場合については DQN の学習戦略の方が優れていることがわかる．

一方，図 2.20 の結果では，A3C が効果的な投資行動を行っているのに対し，DQN は inventory の値が乱高下し，capital も増加させることができていない．この試行でサンプリングされたシミュレーション環境，stylized agent 環境では DQN agent は非常に大きなポジションをとり，加えてそれらが全く有効ではないことが見て取れる．

このような行動を DQN agent がとった原因としては，DQN が決定的なアルゴリズムであることが考えれられる．すなわち，DQN は常に行動価値関数の予測値が最大となる行動を選択するため，状態によっては同一の行動を連続で選択し続け，それによりポジションが非常に大きくなってしまうことが想定される．このような状況は深層強化学習モデルの「暴走」とも解釈でき，大きな損失や金融市場の不安定化を招くリスクになると考えられる．

検証段階 100 試行における各種法の Sharpe ratio および maximum drawdown の頻度分布を図 2.21 および図 2.22 に示す．両指標に共通する特徴として，A3C は指標のばらつきが小さくなっている．特に maximum drawdown については 9 割近いケースで 0 最小カテゴリで推移しており，損失のリスクを抑えた行動選択が行えていることがわかる．一方で Sharpe ratio については平均すると DQN より低い値に推移しており，DQN で効率的な行動選択が行えた場合と比べると劣る結果となっている．

図 2.22 に示すとおり，DQN では 1 割程度のケースで大きな損失を出している一方，A3C ではそのような状況は観測されていない．この要因としては，A3C が確率的な行動選択

を行う手法であることが考えられる．各行動の選択確率を予測する A3C では DQN のように連続して同一の行動を選択する可能性が低い．したがって行動選択も決定的な手法に比べ安定し，大きな損失を避けた投資が行えていると考察される．

以上の結果より，価値関数ベース，方策ベース両手法のトレードオフ関係が明らかとなった．先述のように深層強化学習手法は DQN や A3C 以外にも数多く提案されており，手法選択により行動パターンや投資実績，行動の安定性が大きく変動することが示唆されている．今後，異なる条件化での実験や異なる観点からの考察により，投資戦略学習の効率や有効性が向上することが期待される．

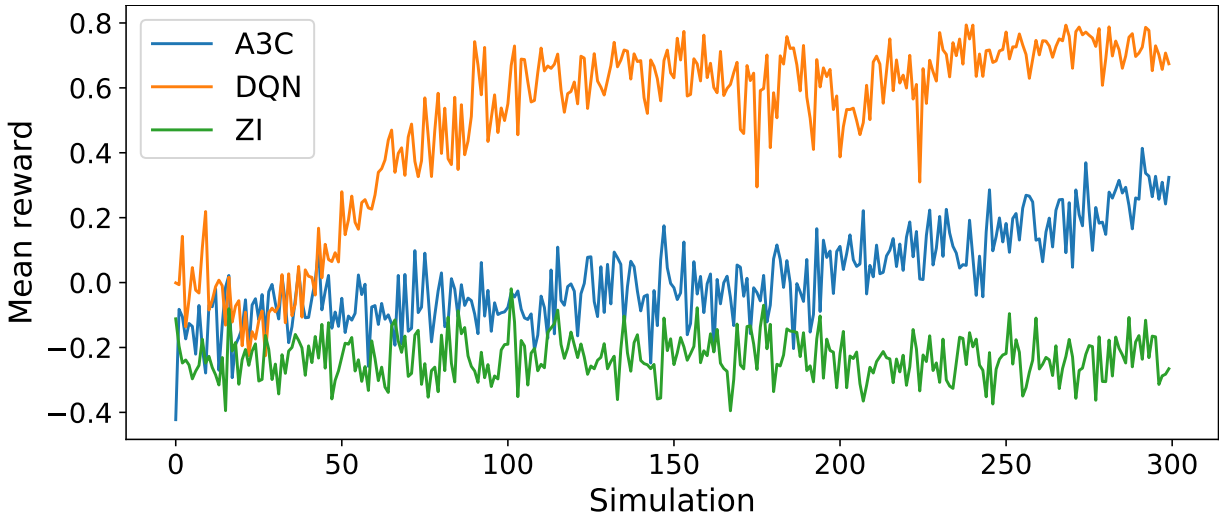


図 2.17: 学習段階における Asynchronous Advantage Actor-Critic (A3C)，Deep-Q Network (DQN)，zero-intelligence agent (ZI) の average reward の推移，横軸が sub simulation を，縦軸が sub simulation 内における average reward の値を示している．sub simulation は並列に進行するため，学習順は多少前後している．

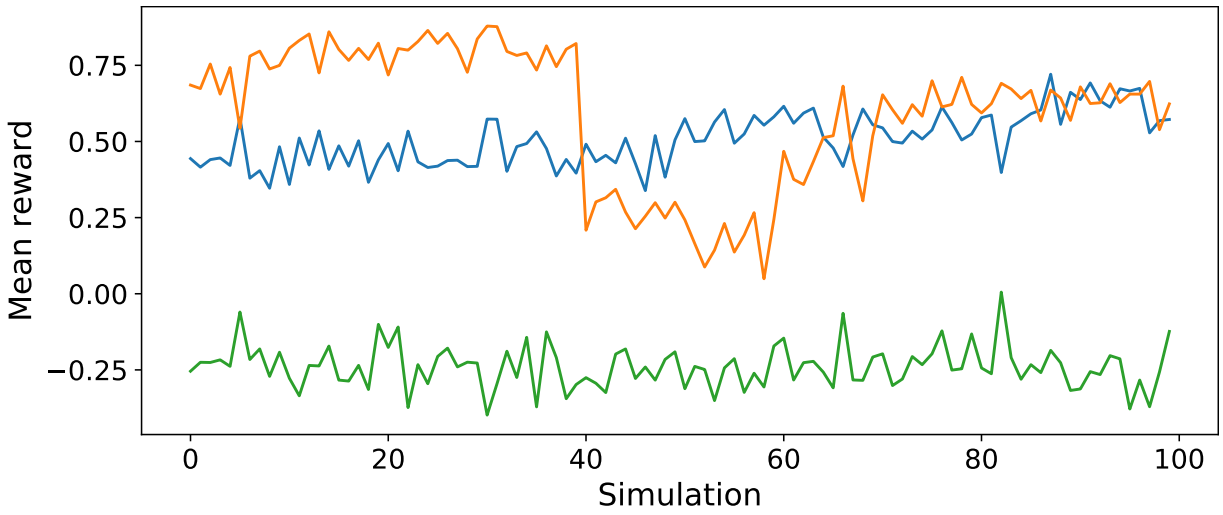


図 2.18: 検証段階における Asynchronous Advantage Actor-Critic (A3C)，Deep-Q Network (DQN)，zero-intelligence agent (ZI) の average reward の推移，横軸が sub simulation を，縦軸が sub simulation 内における average reward の値を示している．sub simulation は並列に進行するため，学習順は多少前後している．

表 2.4: 深層強化学習エージェントの投資実績．A3C は Asynchronous Advantage Actor-Critic を，DQN は Deep-Q Network を，ZI は zero-intelligence agent を表している． 投資実績は $E[r]$，S_p，MDD で評価され，validation simulations 100 回のシミュレーションの平均および標準偏差を示す．

Agent	$E[r] \uparrow$	$S_p \uparrow$	$MDD \downarrow$
A3C	0.5081 ± 0.0850	0.1609 ± 0.0516	$\mathbf{0.0015 \pm 0.0007}$
DQN	$\mathbf{0.5943 \pm 0.2190}$	$\mathbf{0.2087 \pm 0.1271}$	0.0052 ± 0.0093
ZI	-0.2395 ± 0.0706	-0.0126 ± 0.0140	0.0120 ± 0.0076

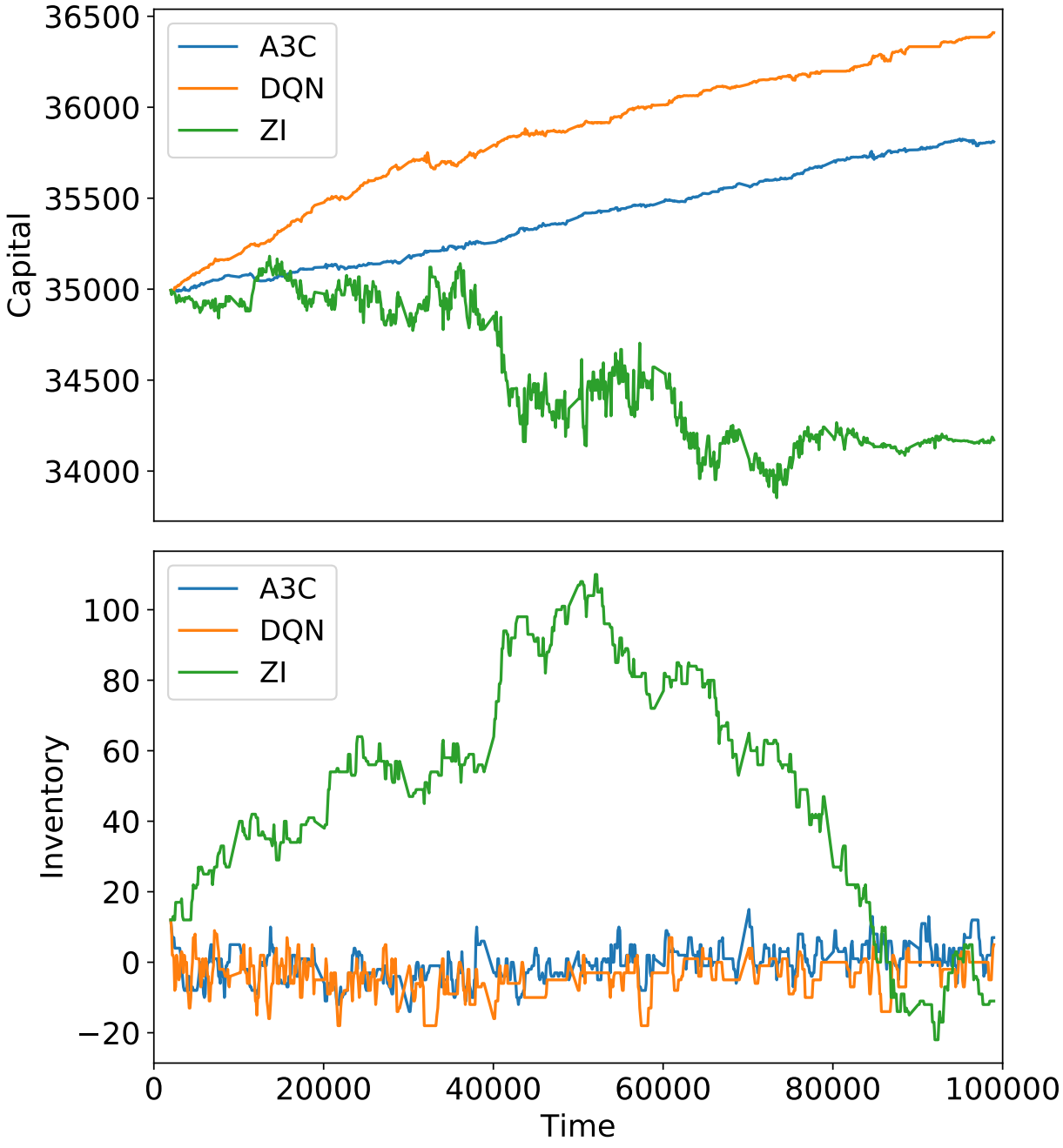


図 2.19: Validation simulation 5 試行目における投資結果．A3C (Asynchronous Advantage Actor-Critic)，DQN (Deep-Q Network)，ZI (zero-intelligence agent) それぞれの capital (liquidation value) および inventory (証券保有量) の時間変化をプロットしている．

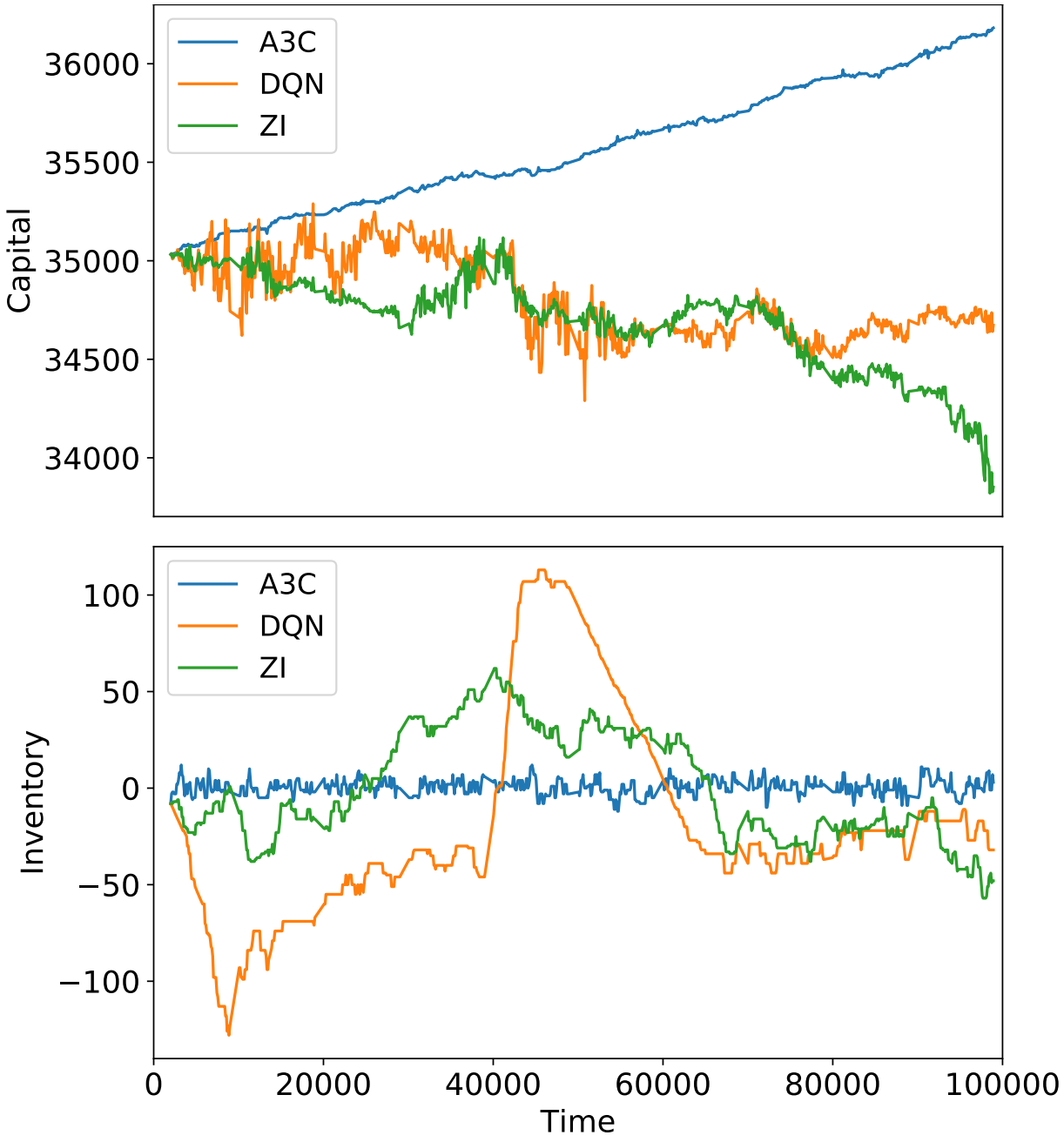


図 2.20: Validation simulation 59 試行目における投資結果．A3C (Asynchronous Advantage Actor-Critic)，DQN (Deep-Q Network)，ZI (zero-intelligence agent) それぞれの capital (liquidation value) および inventory (証券保有量) の時間変化をプロットしている．

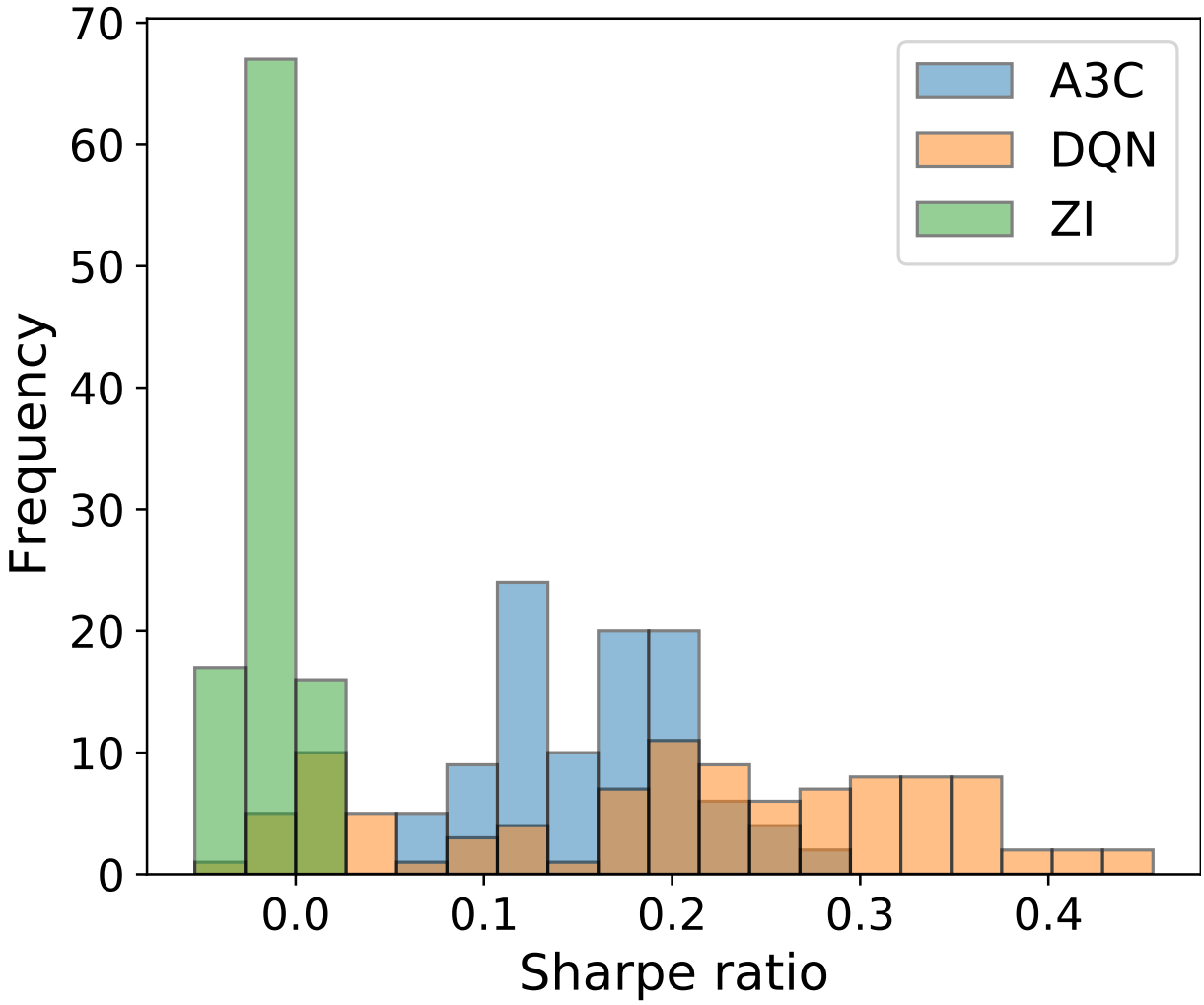


図 2.21: Validation simulation100 試行における Sharpe ratio のヒストグラム，A3C が Asynchronous Advantage Actor-Critic，DQN が Deep-Q Network，ZI が zero-intelligence agent の結果を示している．

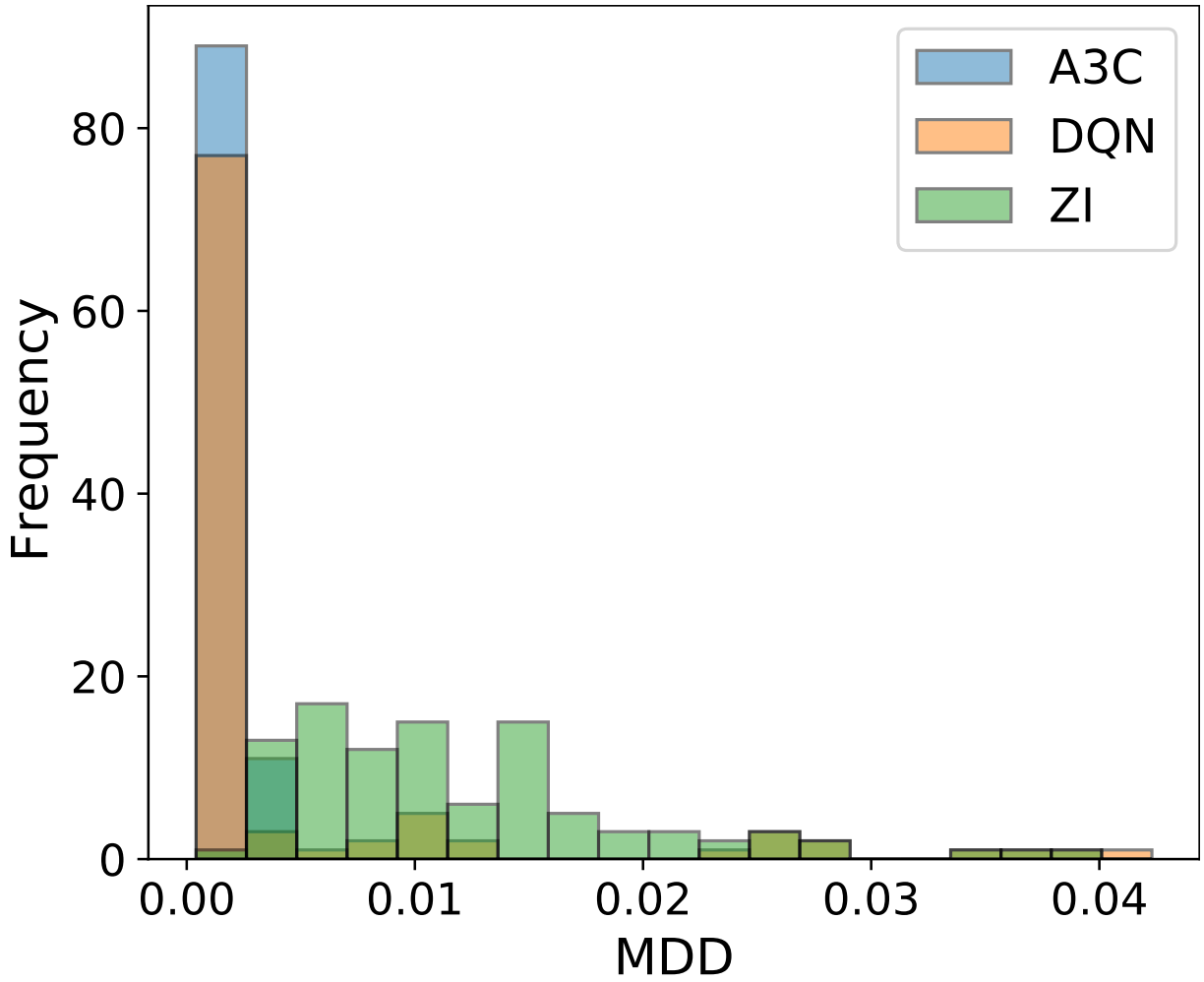


図 2.22: Validation simulation100 試行における maximum drawdown のヒストグラム，A3C が Asynchronous Advantage Actor-Critic，DQN が Deep-Q Network，ZI が zero-intelligence agent の結果を示している．

2.4 結論と課題

本研究により，これまで実現が困難であった金融市場における深層強化学習が人工市場の導入により可能となり，人工市場シミュレーションを用いた学習データの拡張も現実的なものとなった．人工市場シミュレーション上で深層強化学習モデルを学習させるためのシミュレータ，金融市場，エージェント 設計，報酬関数の提案を含む理論的な発展，および学習の実現に付随した技術的な進展は有用なものであると考えている．本研究で提案したフレームワークを用いて学習した投資戦略はシミュレーション環境においては有効に働き，詳細な考察により学習戦略の定性的な妥当性も確認された．加えて，複数の深層強化学習手法，deep Q network (DQN) および asynchronous advantage actor-critic (A3C) を用いた比較実験を行い，決定的な予測を行う価値関数ベース手法，確率的な予測を行う方策ベース手法のトレードオフ関係を発見した．強化学習手法の行動規範の比較はシミュレーション環境のみでなく，実市場における投資戦略学習にも応用でき，非常に有用な知見抽出が行えたと考えている．

現状における本手法の課題としては，以下のような点が考えられる．

1. 異なる人工市場環境への適用
2. 複数市場環境への適用
3. 深層強化学習エージェント間の相互作用

以下でこれらの課題について概説する．

2.4.1 異なる人工市場環境への適用

本章では，stylized agent で構成される人工市場を対象にとり実験を行った．しかしながら，人工市場はその目的に応じ多種多様な形態が提案されている [134]．特に，高頻度取引市場を模した人工市場 [135, 136] への適用は，現実の株式市場との類似性を考慮しても非常に重要なタスクであると考えられる．高頻度取引市場に対し本手法を適用する場合，注文時間スケールで実行しているシミュレーションに実時間スケールを導入する，単一の注文ではなく複数の注文行動を考慮したより現実的な行動戦略の parameterize といった手法の改良が必要である．実時間スケールの導入には，取引開始 (寄り前および寄

り) および取引終了 (引け) 前後における環境エージェントの行動変化や各エージェントの行動間隔，行動までの時間遅れの導入と言った課題が付随する．現実的な行動戦略の parameterize を行う場合，単純な one-hot 形式における離散化では行動候補数が膨大になり，深層強化学習モデルの学習が現実的な計算時間で行えなくなってしまうため，より学習効率を高めるような parameterize 手法が必要となる．

また，現状の stylized agent を用いた人工市場環境においても，市場設定に改良を加えることで，有効な知見抽出が行えると考えている．例として，通常は幾何ブラウン運動に従い変動する fundamental price に突発的な変動を加える fundamental shock [137] が考えられる．Fundamental shock は現実市場における突発的なイベント (株価に影響を与えるニュース等) に対応しており，非定常な市場状態における DRL エージェント行動の効率性や安全性について検証が行えると考えている．

2.4.2　複数市場環境への適用

本章では単純化のため単一市場環境についてのみ実験を行った．しかし，現実の金融市場は多くの市場とそれらの複雑な相互作用により成り立っており，そのような複数市場環境においても実験を行う必要がある．複数市場間の相互作用については，単純には fundamental price が従う幾何ブラウン運動に相関をもたせることが考えられる．しかしながら，単純な相関関係のみの導入の場合，複数市場を利用した投資戦略が必要なく，一市場のみを用いた取引戦略が学習される可能性が高い，むしろ説明変数および行動候補数の増加により学習効率の低下のみが現れると考えられる．複数市場環境において有用な実験を行うためには，複数市場に対する取引を活用してポートフォリオを組むような市場設定および相関関係の設定が必要である．

また，上記の目的に加え，複数市場環境を用いることで，学習済みモデルの転移学習 (transfer learning) [138, 139, 140] についても検討が可能であると考えている．特定の環境で学習した DRL モデルを活用し，他の環境においても効率的あるいは安全に行動選択が行えるモデルを構築することは，学習効率のみならず，異なる市場間で普遍的に有用な知見抽出につながると考えている．

2.4.3 深層強化学習エージェント間の相互作用

本章では環境エージェント複数体，DRL エージェント 1 体の市場環境において実験を行った．環境エージェントは特定のルールベースで行動選択を行うため，一度 DRL エージェントが効率的な行動戦略を学習した場合，以後は安定的に利益を挙げることができる．今後の研究で複数の DRL エージェントが存在する市場環境，あるいは DRL エージェントのみの市場環境で実験することができれば，他の適応的エージェントの行動を予測したより実践的な行動戦略学習が可能となると考えられる．加えて，DRL エージェントのみの市場環境で学習を行うことは，Alpha Go Zero [74] に代表されるような，先天的な知識に一切頼らない学習の実現につながり，market microstructure における有用な知見抽出が可能となると期待される．

しかしながら，複数 DRL エージェントを活用した実験にはエージェント数分の深層学習モデルおよび GPU 環境が必要となり，膨大な計算リソースおよび計算環境構築が必要となり現状では実現は難しい，現実的な環境での実現も含め，フレームワーク自体の改良が必要である．

第3章　模倣学習を用いた注文行動モデリング

2 章ではシミュレータ，金融市場，エージェント，DRL モデル，報酬関数の設計により人工市場環境における深層強化学習モデルの学習を実現した．しかしながら，1 章でも解説した通り，人工市場は現実市場と様々な面で異なる設計・性質を持つことが知られている．例として，2 章で設計した人工市場はその設計上，以下のような点の考慮が難しい．

- 実時間スケールにおけるシミュレーション，情報取得の時間遅れ
- 数量変更注文，価格変更注文を始めとする複雑な注文行動
- 取引開始前後および取引終了直前の市場変化
- 突発的な fundamental price の変化および市場変動

これらはすべて現実の金融市場では考慮が必要な内容である．実現可能な注文の多様性や突発的な fundamental price の変化についてはモデルの改良によりシミュレーションに組み込むことが可能であるものの，環境エージェントの注文行動の選択アルゴリズムや fundamental price の変動方法等，調整すべき内容が多く実市場に合わせた導入は非常に困難である．また，シミュレーションの動向についても，大局的な stylized facts の観測と言った観点における実市場への近似は進んでいるものの，実市場に存在する多種多様なダイナミクスをモデル化できいるとは言い難い．実市場にどれだけ漸近したかを示す評価指標の開発や，それに基づくシミュレータの改良は今後の大きな課題である．

さて，ここまで人工市場はルールベースのモデルを前提としてきたが，必ずしもシミュレータはルールベースのものである必要はなく，過去取引データからデータドリブンに学習するものも考えられる．直前で述べた実市場の挙動への近似は，機械学習・深層学習を用いて実現することも可能である．そこで本章では，模倣学習を用いた過去取引データか

らの投資家の注文行動モデリングを行う．市場状態を入力とし，次に観測される注文行動を予測するモデルを過去取引データから学習することで，データドリブンなシミュレータを作成可能であると期待される．本研究を行う段階では深層学習モデルの学習および予測にかかる計算時間の関係上シミュレータとしての運用は難しいが，計算機の高速化や学習済み深層学習モデルの予測プロセス高速化が成されれば現状の人工市場の大体として利用可能であると考えている．

模倣学習を用いた注文行動モデリングを行う際に重要なのは，金融市場におけるミクロな挙動，すなわちマーケットマイクロストラクチャーの考慮である．内部構造が複雑であり，市場状態および可能な注文行動の多さに対し圧倒的に学習データが少ない金融市場に対し効率的な学習を行うためには，金融市場が持つミクロな特徴を考慮したモデル設計を行う必要がある．例として本章では過去取引データが多様な投資判断基準を持った複数の投資家の行動の集合であり，かつ取引データ上では投資家に関する情報が匿名化されている事象に注目し，投資家の情報が匿名化されたデータからでも適切に複数の投資戦略を抽出できる模倣学習手法の提案を行っている．

3.1　関連研究

3.1.1　模倣学習

模倣学習 (imitation learning, IL) [141, 142, 143, 144] は，対象の行動規則を主に機械学習を用いて近似する手法であり，自動運転 [145, 146] やロボット工学 [147]，ゲーム操作 [148, 144] 等の分野で活用されてきた．

他の機械学習分野同様，模倣学習についても近年は深層学習疑似術の適用が進んでいる．例としては one-shot IL [149, 150]， third-person IL [151]，generative adversarial IL (GAIL) [152, 153, 154, 155, 156] 等が挙げられる．また，熟練者の (expert behavior) 行動を強化学習の枠組みで学習する inverse reinforcement learning (inverse RL) [157, 158, 159, 160] も，模倣学習の一種と解釈できる．

本研究の対象である金融市場データは複数の異なる行動規則を持つ投資家の行動の集合であり，学習する行動規則も複数仮定することが妥当である．このような複数の方策分布の近似を行う手法は multi-modal IL と呼ばれている [161, 162]．中でも，行動者のラベルなしデータから generative adversarial networks (GAN) [163] を用いて複数の方策分布を近似する Hausman らの研究 [164] は本研究でも参考にしている．しかし，複数の参加者・行動者に由来するような混合戦略の学習については現在に至るまで研究されていない．

金融データに対する模倣学習の応用例は存在するが，いずれも限定的な例にとどまっている [165, 166]．

3.1.2　Latent segmentation

Latent segmentation [167] は，統計的情報を元にデータの分割を行う手法であり，古くは消費者の segment を用いたマーケティング手法の提案等に用いられてきた [168, 169]．近年では，ニューラルネットワークや深層学習を用いた latent segmentation も提案されている [170]．

3.2 Latent segmentation imitation learning (LSIL)

本章では解析対象として株式市場の過去取引データを用い，市場状態を入力として次の注文行動・行動確率を予測するモデルの学習を目指す．ここで問題となるのは，プライバシーの観点から，解析用に入手可能な取引履歴データは注文者の情報が匿名化されている [171, 172] ことである．個々の投資家がどのような注文行動を行うかは非常に重要な情報であり，それらは当然公開される取引履歴データには記載されない．しかし一方で取引履歴データは多数の異なる投資家の行動履歴の集合であり，個々の投資家の行動戦略が異なると仮定するならば，匿名化されたデータを用いて単純な模倣学習を行うことは不適切であると考えられる．

複数の戦略を同時に学習する模倣学習は multi-modal imitation learning と呼ばれる．一般的な multi-modal imitation learning は，1 体のエージェントに複数の行動モードを仮定し，行動モード毎に模倣学習を行うことで複数モードへの対応を行う．しかし本章のタスクではモードがステップごとに変化するため，市場状態をもとにしたモードの予測と各モードに対応する模倣学習を同時に行う必要がある．

本研究で対象とする投資者のラベルなしデータを用いた複数取引戦略の学習は，単純な模倣学習モデルを用いる場合，学習の自由度が高すぎる．戦略の数，どの行動がどの戦略に属するか等，予想すべき要素があまりに多いため，正常に学習が進むとは考えにくい．そこで本研究では，マーケットマイクロストラクチャーの観点からモード (あるいは投資家) 変化に対する仮定を置き，モードに紐付けられた報酬関数の最大化問題を設定，モードの予測を可能とした．投資家の数や種類は観測不可能であるため，市場を構成する投資戦略に有限個のクラスタを仮定，クラスタそれぞれに報酬関数を設定し，クラスタ全体での期待報酬の最大化と行動の尤度最大化のマルチタスクとして問題設定を行うことで，ラベルなしデータからの戦略の抽出を可能とした．本手法を Latent segmentation imitation learning (LSIL) と呼ぶ．

本研究の成果の一部は Journal of Risk and Financial Management，Special Issue ”AI and Financial Markets II” で発表している [173].

3.2.1　手法概要

提案手法では，投資家の行動規則は有限個の latent segment に分類されると仮定する．投資家は各時点においていずれかの latent segment に属し，市場状態の変化に応じ異なる segment に移動することも可能であるが，同時に複数の segment に属することはできないとする．

Latent segment $s_i(i = 1, 2, \cdots, N)$ がそれぞれ固有な投資戦略を代表すると仮定し，それぞれの投資戦略の方策関数を $\pi_{s_i}(o|X;\theta_i)$ とする．ここで，X は市場状態，θ_i はパラメータを表す．また，方策関数としては各注文行動の選択確率を想定している．

いま，市場状態 X において latent segment s_i に属する投資家が行動を行う確率を $p(s_i|X)$ と定義する．本研究ではこの確率を segment probability と呼び，各 segment についての segment probability の和は 1 となる．すると，各 latent segment の方策関数および segment probability を用いて，市場全体で観測される方策関数 $\pi(o|X)$ は以下のように表すことができる．

$$\pi(o|X) = \sum_{i=1}^{N} p(s_i|X)\pi_{s_i}(o|X;\theta_i) \tag{3.1}$$

以上のような仮定により，latent segment を仮定した市場全体の方策関数の規定が可能となる．したがって，segment probability $p(s_i|X)$ および各 segment の方策関数 $\pi_{s_i}(o|X;\theta_i)$ をニューラルネットワークで近似することで取引データから各関数を学習することができる．Segment probability $p(s_i|X)$ を近似するニューラルネットワークを segment network，各 segment の方策関数 $\pi_{s_i}(o|X;\theta_i)$ を近似するニューラルネットワークを segment level order network と呼ぶ．

学習において最大の問題となるのは，segment network の学習である．前述の通り各注文に segment を特定できるような情報は存在せず，そのままでは segment network についての勾配を計算することができない．そこで本手法では，segment ごとに報酬関数 $r_{s_i}(o)$ を導入する．報酬関数は各 segment で特異であり，同一の注文に対しても各 segment で報酬が異なる．すると，以下のように報酬期待値 $E[r(o)$ を計算することができる．

$$E[r(o)] = \sum_{i=1}^{N} p(s_i|X;\theta_s)r_{s_i}(o) \tag{3.2}$$

仮に市場が効率的であるならば，より多くの報酬が見込まれる行動を各 segment に属する投資家が行っているはずであり，$E[r(o)]$ 最大化により segment probability $p(s_i|X)$ を学習することができる．しかし，これまでの研究で市場は完全に効率的でないことが主張されており [174, 175, 176, 177] このような仮定は非現実的である．そこで本手法では，非効率的な行動を行う segment s_* を導入することでこの問題の解決を図った．s_* は一様な報酬関数を持ち，その導入により segment probability の和についての制約は $\sum_{i=1}^{N} p(s_i|X;\theta_s) + p(s_*|X;\theta_s) = 1$ となる．s_* は人工市場等でも導入される noise trader [178, 179] を代表すると解釈することもできる．以上より，segment network のパラメータ θ_s についての勾配は以下のように計算できる．

$$-\nabla_{\theta_s}\left(\sum_{i=1}^{N} p(s_i|X;\theta_s) r_{s_i}(o) + p(s_*|X;\theta_s) r_*\right) \tag{3.3}$$

ここで，r_* は s_* の報酬を表す．r_* の値によって $p(s_*|X;\theta_s)$ が常に大きく，あるいは小さくなる危険性は存在するが，適切な報酬関数の scaling によりこの問題はある程度解決できると考えられる．

効率的な学習のためには，報酬関数は同程度なスケールを持つことが望ましい．報酬関数としては様々なパターンが考えられるが，本研究では手法の有効性確認のため，もっとも基本的なケースとして，以下に示す注文の profit and loss (P&L) を用いた報酬関数を使用した．

$$r(o) = p_{\mathrm{mid},\tau}\Delta I + \Delta c \tag{3.4}$$

ここで，$p_{mid,\tau}$ は注文時から時間 τ だけ進んだ時点における仲値，ΔI および Δc は注文行動により生じた株式保有量および現金保有量の変化量であり，それぞれ市場の注文板状態から計算することができる．ΔI および Δc は注文の種類により扱いが異なり，LMT(指値注文) および MKT(成行注文) については，ΔI が約定数量，Δc が約定による現金変化である．Buy LMT および Buy MKT では $\Delta I \geq 0$，$\Delta c \leq 0$ となり，Sell LMT および Sell MKT では $\Delta I \leq 0$，$\Delta c \geq 0$ となる．注文が約定しなかった場合は両者はいずれも 0 となる．対して CXL(取り消し注文) では，その注文によって避けることができた約定量である．すなわち，その CXL の取り消し先の注文を取り消さなかった場合の ΔI および Δc を計算し，それを-1 倍することで計算される．したがって，buy CXL で

は $\Delta I \leq 0$，$\Delta c \geq 0$ となり，sell CXL では $\Delta I \geq 0$，$\Delta c \leq 0$ となる．

3.4 式で導入した報酬関数において，時間スケールを表す τ は $p_{mid,\tau}$，ΔI，Δc すべてに影響を与えるパラメータである．したがって，各 segment の違いを τ の値の違いで表現することができる．各 segment ごとに異なる τ を設定することで segment network が学習可能となり，それに伴い各 segment level order network も学習可能となる．

各 segment level order network は一般的な cross entropy 最小化問題により学習することができる．各 segment level order network のパラメータ θ_i についての勾配は以下のように計算できる．

$$\nabla_{\theta_i} CE(o_t, \sum_{i=1}^{N} p(s_i|X_t)\pi_{s_i}(o|X_t;\theta_i)) \tag{3.5}$$

ここで，CE は cross entropy 関数，o_t および X_t は観測された注文および市場状態を表す．また，上式で計算した損失関数は segment network のパラメータ θ_s で微分することで segment network のパラメータ更新にも用いることができる．Cross entropy を含む θ_s の勾配は以下のように計算される．

$$-\nabla_{\theta_s}\left(\sum_{i=1}^{N} p(s_i|X;\theta_s)r_{s_i}(o) + p(s_*|X;\theta_s)r_* + CE(o_t, \sum_{i=1}^{N} p(s_i|X_t)\pi_{s_i}(o|X_t;\theta_i))\right) \tag{3.6}$$

本研究においては cross entropy を含む θ_s の勾配でパラメータ更新を行う手法を LSIL1，cross entropy を含まない θ_s の勾配でパラメータ更新を行う手法を LSIL2 と呼び，その両者について実験を行った．

3.2.2 モデル構成

本研究で用いた説明変数としては，2 章でも用いた 2 つの特徴量，price series および orderbook features を用いた．Price series は一定のインターバルごとに取得した 10 件の市場価格の系列である．インターバルは各 segment の time scale τ_{c_i} を 10 で割った値とした．Orderbook features は直近の注文板における仲値中心上下 10 価格の数量であり，買い注文の数量は-1 倍することで売り注文との区別を容易にしている．

注文行動の離散化の概要を図 3.1 に示す．注文行動の離散化は注文の種類 (LMT, MKT, CXL)，注文の価格，および数量を用いてなされた．ここで，価格変更注文や数量変更注文については，変更後の注文のみを注文行動とした．注文の価格については仲値中心で上下 10 価格分に加え，成行注文を表す価格を考慮し合計 $10 \times 2 + 1 = 21$ 価格を選択可能な候補とした．仲値から 10 価格分以上離れた注文については，市場状態への影響が少ないと考え考慮しない．数量に関しては，LMT および MKT は正の値，CXL は負の値とし，それぞれ最小取引単位の 5 倍までの 5 数量を選択可能とした．最小取引単位の 5 倍以上の数量の注文については，数量を 5 にクリッピングして解析に用いた．以上より，$21 \times 10 = 210$ カテゴリの行動が選択可能となる．

解析に用いた LSIL モデルの構造を図 3.2 に示す．3.2.1 章で解説したとおりモデルは segment network および segment level order networks で構成され，両者が予測した segment probability および segment level order probabilities を組み合わせて市場全体の注文確率 overall order probability が予測される．モデルの学習時には，segment 毎の報酬関数および segment probability を用いて計算した reward loss，overall order probability と観測注文を元に計算した order probability loss の両者から勾配が計算される．

各 segment network および segment level order networks のネットワーク構成としては，2 章でも活用した LSTM 層および convolutional 層中心のものを用いた．すなわち，price series からの特徴抽出には同様の特徴量を用いた研究 [23, 24] を元に LSTM [104] を，注文板情報からの特徴抽出には注文板の位置情報 [110, 180] を考慮できる convolutional および maximum pooling 層 [181] を用いた．両説明変数から抽出された特徴量は結合され，全結合層を用いて各 network の出力が得られる．

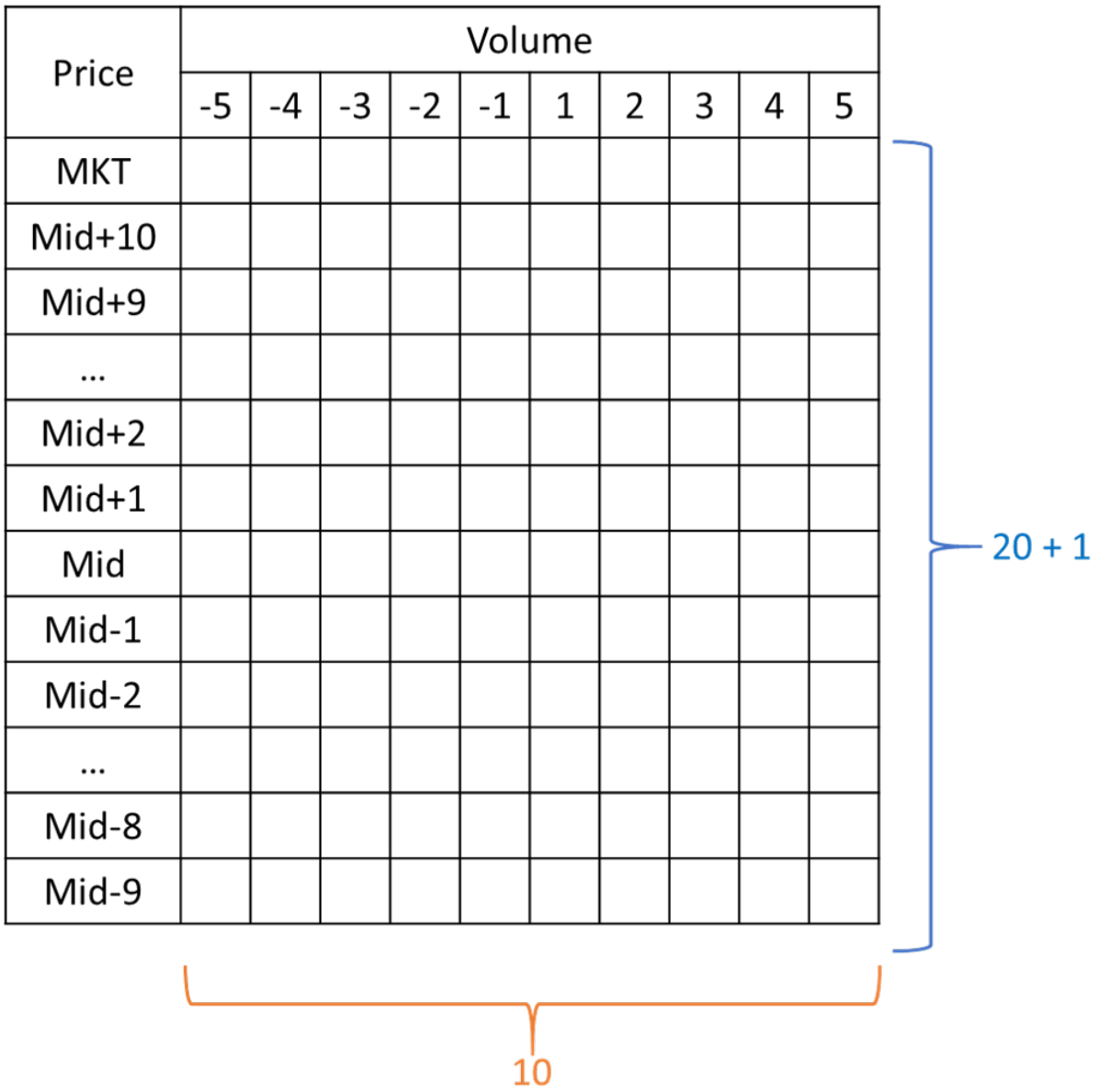


図 3.1: LSIL 実験で用いた注文行動の離散化の概要．注文は価格および板状における変動数量を用いて離散化された．価格については仲値中心で上下 10 価格分に加え，成行注文を表す価格を考慮し合計 $10 \times 2 + 1 = 21$ 価格を選択可能な候補とした．数量に関しては，LMT および MKT は正の値，CXL は負の値とし，それぞれ最小取引単位の 5 倍までの 5 数量を選択可能とした．

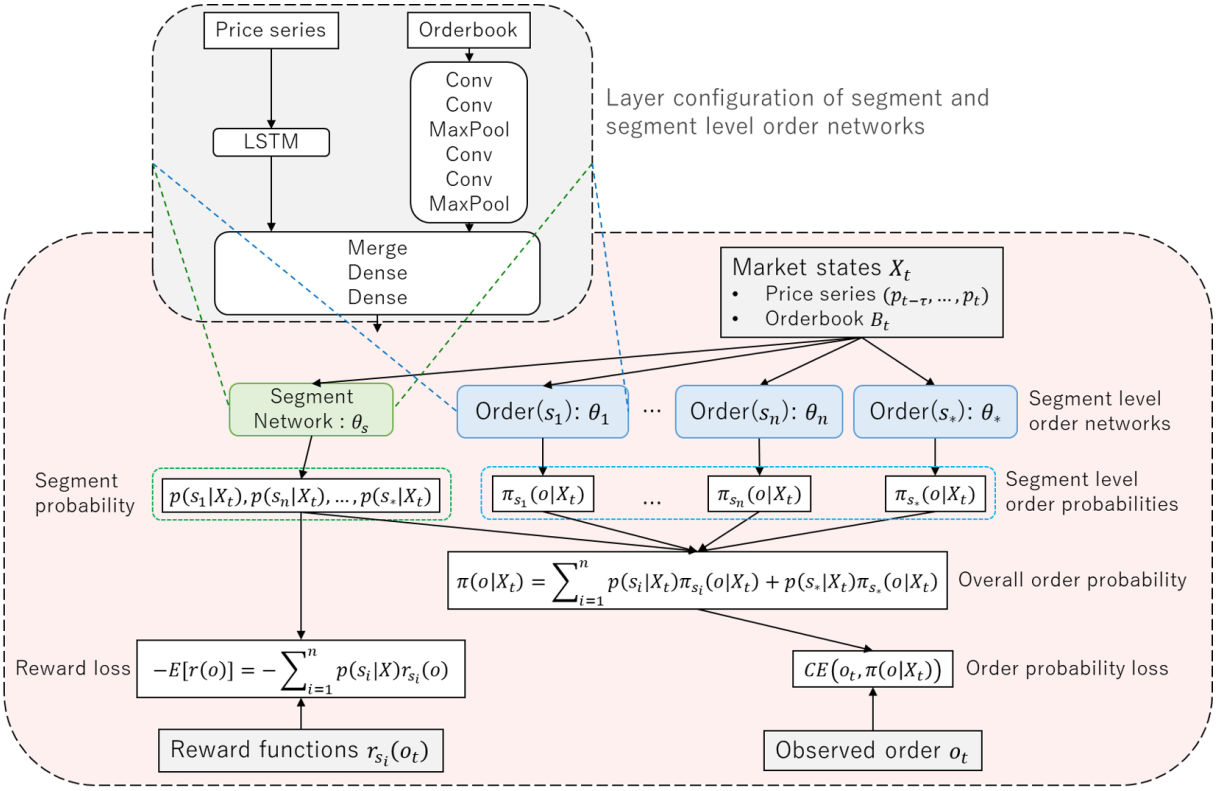


図 3.2: 本実験で用いた latent segmentation imitation learning (LSIL) モデルの構造．LSIL モデルは segment probability を予測する segment network と segment 毎の方策関数を予測する segment level order networks で構成されており，両者の予測値を組み合わせることで市場全体の注文行動が予測される．モデルの学習は報酬関数についての損失関数，および注文の cross entropy についての損失関数を用いたマルチタスクで行われる．各 segment network および segment level order networks は LSTM 層および convolutional 層を中心とした同一のネットワーク構成をしている．

3.2.3　実験概要

提案手法の有効性を確認するため，人工市場によるシミュレーションデータ，および実市場の取引データを用いてモデルの学習および検証を行った．シミュレーションデータを用いた実験においては，提案手法の segment 分けと対応するようなモデルでシミュレーションを行い，提案手法の学習が適切に行われるか確認した．実市場データを用いた実験においては，当然ながら実際の市場参加者の segment は不明であるが，予測精度やケーススタディを通し，提案手法の有効性を検証した．

3.2.1 章で解説したとおり，提案手法の LSIL は order probability loss も活用し segment network のパラメータを更新する場合 (LSIL1)，reward loss のみから更新する場合 (LSIL2) の 2 パターンについて実験を行った．比較手法としては，一般的な imitation learning (IL) モデル，generative adversarial imitation learning (GAIL) [152] モデル，segment imitation learning (SIL) モデルを採用した．IL モデルは LSIL モデルの 1 つの segment level order network と同様のネットワーク構成を持ち，単一の方策で注文行動を近似する．GAIL モデルは sequence generative adversarial nets (Seq-GAN) [182, 183] を元に作成しており，市場状態を元に注文系列を生成する．SIL モデルは LSIL モデルと同様の構造を持つが，reward loss を用いた segment network の学習を行わず，order probability loss の最小化のみを目指してパラメータ更新を行う．

モデルの予測性能は以下の指標を用いて比較する．

- Precision at k (P@k)
- Area under receiver operating characteristic (AUROC)
- Expected reward $E[r(o, X)]$

Precision at k はモデルの予測確率上位 k 件に正解が含まれる割合であり，本実験では $k = 1, 5, 10$ の場合について指標を計算した．Expected reward $E[r(o, X)]$ は LSIL の予測する segment probability の妥当性を検証するために計算され，正の大きい値であるほど適切な segment probability の予測が行えていることになる．

3.2.4 シミュレーションデータを用いた実験

提案手法の有効性を確認するため，人工市場によるシミュレーションデータを用いてモデルの学習および検証を行った．人工市場シミュレーションについては，2.2.2 章および 2.2.3 章 で解説したような，stylized agent の行動で進行するシミュレーションを選択した．2.2 章の実験と異なり，シミュレータに DRL agent は登録されていない．シミュレーションは training，validation それぞれ 10 回の sub simulation で構成され，各 sub simulation では stylized agent がサンプリングされた上で 100,000 ステップの stylized エージェントの行動，および行動に伴う金融市場の変動が行われる．本シミュレーションにおいては提案手法の有効性確認のため，以下に示す 6 種類の stylized agent を使用した．

1. Stylized-agent-20
2. Stylized-agent-40
3. Stylized-agent-80
4. Stylized-agent-160
5. Stylized-agent-320
6. Noise-agent

各 stylized agent の後の数字は エージェント の time scale を表し，提案手法の式 3.4 における τ に対応している．Noise エージェント はランダムな注文を行う エージェント であり，提案手法の s_* に対応している．Noise agent は stylized agent の fundamental weight w_F および chart weight w_C を 0 とすることで実装した．各 エージェント のハイパーパラメータを表 3.1 に示す．各 stylized agent は 100 体ずつ，noise agent は 400 体サンプリングされる．各 stylized agent は time scale T の平均が名前末尾の数字と一致するように設定し，それをもとに板の仲値付近に注文を出すよう他の値を調整している．

その他のシミュレーションのパラメータについては，fundamental price が従う geometric Brownian motion (GBM) の volatility は 5×10^{-6}，注文の数量は常に 1 とした．

提案手法の segment については，stylized agent の構成をもとに以下のように設定した．

1. s_1: $\tau = 20$

2. s_2: $\tau = 40$
3. s_3: $\tau = 80$
4. s_4: $\tau = 160$
5. s_5: $\tau = 320$
6. s_6: $\tau = 640$
7. s_*: Exceptional

実験結果を表 3.2 に示す．表 3.2 には各モデルの validation data に対する precision at $k = 1, 5, 10$，AUROC が記載されている．表を見ると，order probability loss を用いて segment network の更新を行わない LSIL2 がすべての指標について非常に良好な成績を残している．同一なネットワーク構造を持つ SIL と比べても LSIL2 が良い結果を残していることから，提案手法で仮定した latent segmentation が有効に働いていることがわかる．一方，order probability loss を用いた segment network の更新は逆効果となっている．加えて，LSIL1 および LSIL2 の expected reward $E[r(o, X)]$ はそれぞれ 0.1127，0.0721 と共に正の値となっており，この結果からも segment probability の予測が適切に行われていることが示唆される．

表 3.1: Stylized agent および noise agent のハイパーパラメータ．各エージェントについて，エージェント数 num agent，fundamental scale σ_F，chart scale σ_C，noise scale σ_N，time scale T，order margin k の値を設定している．Stylized agent の後の数字は エージェント の time scale の平均を表し，その設定値に合わせ他のパラメータを調整している．Noise agent はノイズ情報のみから行動選択を行う．

Agent	Num agent	σ_F	σ_C	σ_N	T	k
Stylized-agent-20	100	1.0	0.1	0.0	$\sim \mathcal{U}(15, 25)$	$\sim \mathcal{U}(0, 0.01)$
Stylized-agent-40	100	1.0	0.1	0.0	$\sim \mathcal{U}(30, 50)$	$\sim \mathcal{U}(0, 0.01)$
Stylized-agent-80	100	1.0	0.1	0.0	$\sim \mathcal{U}(60, 100)$	$\sim \mathcal{U}(0, 0.01)$
Stylized-agent-160	100	1.0	0.1	0.0	$\sim \mathcal{U}(120, 200)$	$\sim \mathcal{U}(0, 0.01)$
Stylized-agent-320	100	1.0	0.01	0.0	$\sim \mathcal{U}(240, 400)$	$\sim \mathcal{U}(0, 0.01)$
Stylized-agent-640	100	1.0	0.01	0.0	$\sim \mathcal{U}(480, 800)$	$\sim \mathcal{U}(0, 0.001)$
Noise-agent	400	0.0	0.0	0.1	$\sim \mathcal{U}(100, 200)$	$\sim \mathcal{U}(0, 0.001)$

表 3.2: 人工市場シミュレーションデータに対する予測精度.Standard imitation learning (IL) モデル，generative adversarial imitation learning (GAIL) モデル，segment imitation learning (SIL) モデル，latent segmentation imitation learning (LSIL1, LSIL2) モデルについて実験を行っており，予測精度は validation data に対する precision at $k = 1, 5, 10$，AUROC で評価される．

Model	P@1 ↑	P@5 ↑	P@10 ↑	AUROC ↑
IL	0.0436	0.1764	0.2889	0.8416
GAIL	0.0166	0.0713	0.1391	0.8263
SIL	0.0546	0.2181	0.3796	0.9071
LSIL1	0.0518	0.1959	0.3446	0.9016
LSIL2	**0.0606**	**0.2439**	**0.4094**	**0.9120**

3.2.5　実市場データを用いた実験

次に，実市場データを用いて同様の実験を行った．実市場データとしては，FLEX FULL historical data を利用した．FLEX FULL historical dataset は 東京証券取引所から提供されている注文および約定系列データである [1] であり，ミリ秒単位で記録された一日あたり数百万件の注文情報が保存されている．本実験では，ID 9022 (東海旅客鉄道株式会社，Central Japan Railway Company) についての 2018 年 1 月 1 日から同年 12 月 31 日までのデータを training data，2019 年 1 月 1 日から 8 月 31 日までのデータを validation data とした．Training, validation 両データセットは利用可能な全データから 10 サンプルごとに 1 件抽出することで作成している．説明変数には 3.2.4 章同様 price series および orderbook features を使用し，segment の設定も同様とした．

実験結果を表 3.3 に示す．表 3.3 を見ると，今回は SIL, LSIL1, LSIL2 がほぼ同程度の精度となっている．また，LSIL1 および LSIL2 の expected reward $E[r(o, X)]$ はそれぞれ 0.0371，0.0179 となっており，正の値ではあるがシミュレーションデータの結果と比べると小さな値となっている．ここから，実市場データデータに対しては latent segmentation が良好には働いていないことがわかる．これは，本実験で設定した segemnt および segemnt の報酬関数があまりにも単純であり，実市場の戦略と対応していなかったことが原因として考えられる．ただ，適切な segment の設定さえなされていれば LSIL が有効に働くことは 3.2.4 章で確認されており，今後，ことなる segment 設定で実験を行うことにより，結果が改善すると期待される．

Validation data 全体を通した segment probability の平均は以下のようになっている．

1. $p(s_1) = 0.4362$
2. $p(s_2) = 0.0517$
3. $p(s_3) = 0.0500$
4. $p(s_4) = 0.0893$
5. $p(s_5) = 0.1348$
6. $p(s_6) = 0.1835$

[1] https://www.jpx.co.jp/english/markets/paid-infoequities/ realtime/index.html

7. $p(s_*) = 0.0546$

これを見ると，最短期の利益を見込む s_1 や，長期の利益を見込む s_6 の割合が高くなっており，逆に非効率的な行動を行う s_* の割合が低くなっている．現実市場でも多くの注文は短期の取引を行うマーケットメーカー [184, 185, 186] によりなされていることから，比較的妥当な結果であると考察できる．

LSIL2 モデルにより予測された segment probability の変遷の例を図 3.3 に示す．図 3.3 を見ると，時間経過による市場状態および market price の変動に伴い，LSIL2 モデルの予測する segment probabilities が変動していることがわかる．注目すべきは 9 時 7 分前後の変動であり，market price が上昇する直前に，青色で示されている s_1，すなわち短期の利益を見込む投資家の行動確率が大きく上昇している．これは価格変動を予知した短期の投資家が行動を行ったと解釈でき，実市場の挙動と照らし合わせても非常に合理的な確率変動をしていると考えられる．s_1 の割合は 9 時 6 分すぎにも上昇しているが，ここでは価格変動は起きていない．予測した segment probability を真とすれば，この場面では投資家が市場変動を読み違えたと解釈することもできる．

以上のように，データの性質上不明瞭な点も残るが，現実市場の取引戦略とも一部合致するような結果を提案手法により得ることができた．

表 3.3: FLEX FULL historical dataset に対する予測精度.Standard imitation learning (IL) モデル，generative adversarial imitation learning (GAIL) モデル，segment imitation learning (SIL) モデル，latent segmentation imitation learning (LSIL1, LSIL2) モデルについて実験を行っており，予測精度は validation data に対する precision at $k = 1, 5, 10$, AUROC で評価される．

Model	P@1 ↑	P@5 ↑	P@10 ↑	AUROC ↑
IL	0.1344	0.4347	0.6266	0.9371
GAIL	0.0456	0.2388	0.4155	0.9091
SIL	0.1392	**0.4469**	**0.6381**	**0.9407**
LSIL1	**0.1401**	0.4466	0.6376	0.9406
LSIL2	0.1398	0.4463	0.6371	0.9404

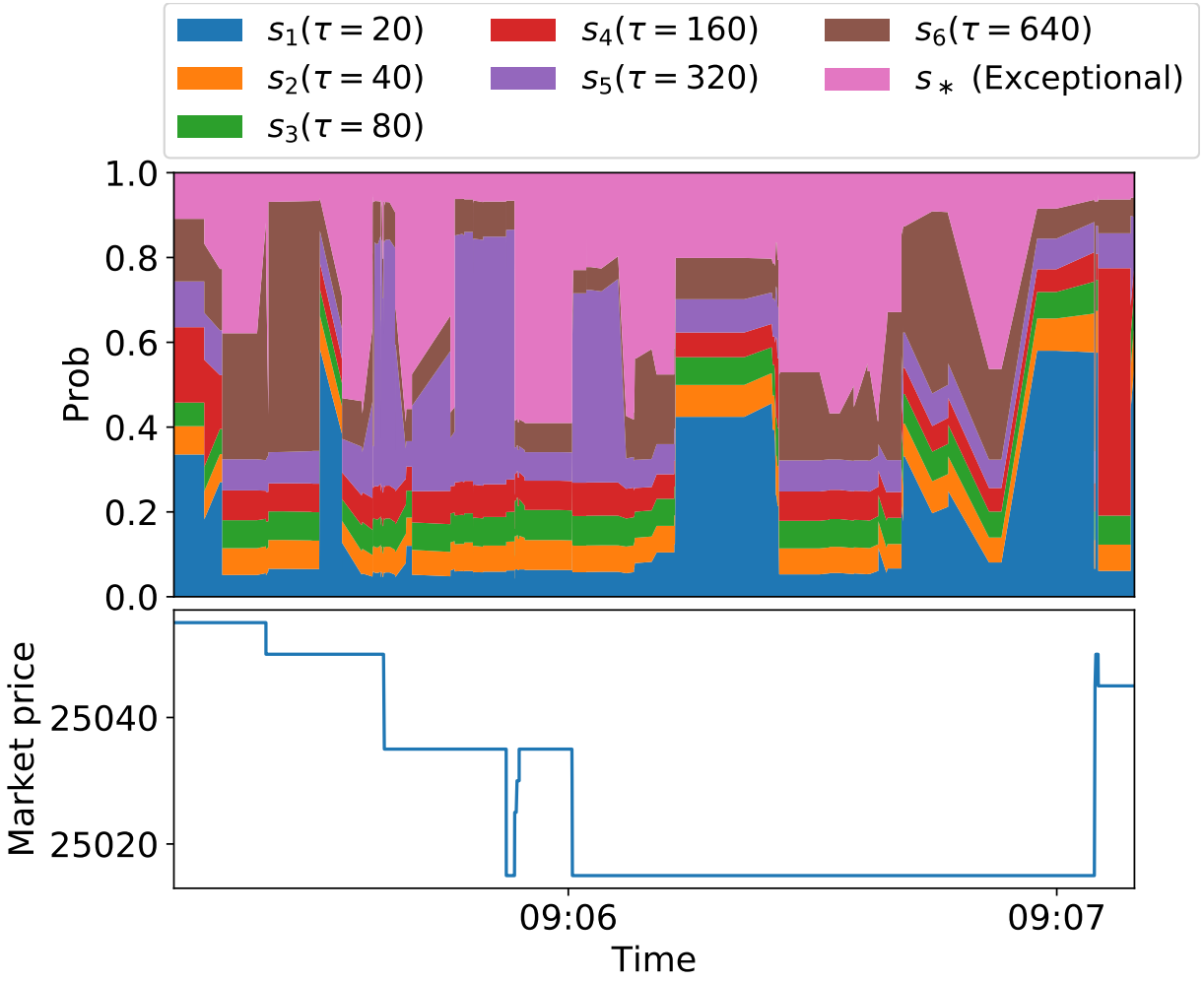


図 3.3: LSIL2 モデルにより予測された segment probability の変遷の例 (2019 年 4 月 4 日)．時間経過による市場状態および market price の変動に伴い，LSIL2 モデルの予測する segment probabilities が変動していることがわかる．

3.3 結論と課題

本章では latent segmentation と multi-modal imitation learning を組み合わせた模倣学習手法を提案し，シミュレーションデータおよび実市場データを用いた検証を通し，ラベルなしデータからの複数行動戦略学習タスクについての有効性を確認した．提案手法は深層学習分野において盛んに研究されている模倣学習とマーケットマイクロストラクチャーについての知見から導かれる報酬関数理論を内蔵し，データドリブンとモデルベースのハイブリッド手法としても有用なものであると考えている．

提案手法を人工市場のシミュレータとして用いる場合の大きな課題は，そのカスタマイズ性の低さである．通常のモデルベースの人工市場はそのパラメータ調整により様々な市場状態を再現でき，本研究の目的に合わせれば深層強化学習モデルの予測妥当性や安全性を高めることが可能である．しかし本章で提案した LSIL は特定の市場状態に対し常に一定な確率分布を予測するため，深層強化学習モデルの学習データ拡張に用いるにはシミュレーション可能な市場状態の多様性が低いと考えられる．

人工市場として用いるためには，内生的あるいは外生的にシミュレータの内部状態を変動させ，シミュレーション結果に多様性を生み出す必要がある．以下に考えられるモデルの改良案および今後の課題を述べる．

3.3.1 多様な行動戦略への対応

本実験では投資家の行動戦略として非常に単純な報酬関数を設定しており，それらは現実の多様かつ複雑な投資戦略とはかけ離れたものとなっている．複数の複雑な報酬関数を用いる場合，その正規化が課題となる．複数の報酬関数を用いた深層学習モデルの最適化を行う場合，少しでも正規化が不十分だと一つまたは少数の報酬関数が支配的になってしまい，学習が適切に行われない．したがって既存の報酬関数ではなく，多様な要素をバランス良く組み合わせた報酬関数の設計が必要となる．

3.3.2 本物らしさの考慮

本章で提案した GAN モデルは常に「学習データと近い」データの生成を目的とする．しかし，生成データを深層学習モデルの学習にもちいる目的に照らすと，一定程度実デー

タから異なる特徴を持つシミュレーションデータの生成がむしろ学習効率の向上に有効であると考えられる．そこで今後の課題として，生成データの「本物らしさ (authenticity)」の考慮を検討している．最も簡単な本物らしさの導入方法は，GAN の discriminator の出力値を本物らしさとして用いることである．GAN の generator にも入力値として本物らしさの目標値を入力し，例えば discriminator の出力値が 0.5 となることを目標にシミュレーションを行うことで，普段観測できない市場状態を擬似的に作成し，学習対象とすることが可能である．

上述のような GAN モデルを考える際，GAN の discriminator の出力値を本物らしさとして用いることの妥当性は問題となる．本物らしさは本章で目的とするような実市場からの乖離度の他に，実現不可能な注文行動かどうかの側面も存在し，両者を明確に切り離す事ができなければ，生成データの妥当性も低下してしまう．注文行動の実現可能性の数理的なモデル化含め，今後の一つの方針であると考えている．

3.3.3　学習したシミュレータを用いた深層強化学習

実市場を的確に再現するデータドリブンなシミュレータの学習が実現した後には，シミュレータを用いた深層強化学習モデルの学習を検討している．この際，シミュレータは実市場の大体として用いるため，以下の実験が必要となる．

1. 実市場データのみを用いた強化学習
2. シミュレータのみを用いた強化学習
3. 実市場・シミュレータ両者を用いた強化学習

上記の実験で実市場・シミュレータ両者を用いた強化学習により学習戦略の有効性が向上すれば，本章の目的達成となる．また，学習した深層強化学習モデルを用いて実際に実市場で取引を行う検証も有効だろう．本研究ではシミュレーション環境における深層強化学習モデルの学習は実現したが，実市場の代替として用いるシミュレータの学習は実現しなかったため，上記実験についても今後の課題となる．

第4章　ベイズ深層学習を用いた不確かさの考慮

1.1 章でも述べたとおり，金融市場予測を不確かにする要因には，

1. 因果関係を特定することの難しさ
2. 内部構造の複雑性・非定常性
3. データに含まれる多くのノイズ
4. マーケットインパクト

といったものが考えられる．この内，内部構造の複雑性・非定常性およびマーケットインパクトの考慮については，2 章および 3 章で一定の成果を挙げた．しかし，依然として他の要素に由来する予測の不確かさは残っている．特に因果関係を特定することの難しさについては，金融市場が社会のあらゆる要素の影響を受けている非常に複雑な系であることを考えると，現状の計算機上で実現できるモデルでは予測の不確かさは完全には回避できないと考えられる．したがって，本研究の目的である安全な金融市場予測および投資意思決定の実現には，金融市場予測における予測の不確かさについて，その軽減および考慮を行う研究が不可欠であると考えている．

Kendall ら [187, 188] によると，予測に伴う不確かさのうち，特に以下の 2 種類の考慮が重要であるとされる．

1. Epistemic uncertainty (認識における不確かさ)
2. Aleatoric uncertainty (偶発的な不確かさ)

Epistemic uncertainty は，学習データに含まれないようなデータ，すなわち外挿データに対する予測の不確かさを表し．Aleatoric uncertainty はデータが説明できない情報に対

する不確かさを表す．このうち epistemic uncertainty についてはデータ拡張により軽減が期待でき，2 章の人工市場および 3 章の模倣学習の適用は有効な手段であると考えられる．一方，aleatoric uncertainty に対しては，予測タスク自体が持つ予測の不確実性でありデータ拡張によって解決することは難しい．

Aleatoric uncertainty の考慮においては，不確かさのモデル化，すなわち損失関数の設計が必要となる．深層学習・深層強化学習においてこのような不確かさのモデル化研究は数多く，金融市場予測タスクに限らず盛んに研究が行われている [189, 190, 191]．中でも本章ではベイズ深層学習 (Bayesian neural networks, BNNs) に着目し実験を行うが，ベイズ深層学習以外にも，Lakshminarayanan らによるアンサンブル手法 [192]，Kuleshov らの手法 [193]，Sensoy らによる evidencial deep learning [194]，Nair らによる手法 [195] といった多くの手法が提案されている．本章で採用したベイズ深層学習はこれらの手法の中でも，同一の入力に対し幅のある予測が行えることが特徴である．深層学習における不確かさ評価手法を金融市場予測に応用する場合，予測結果を用いた投資意思決定アルゴリズムの提案が必要不可欠である．このような課題に対しベイズ深層学習により得られる幅のある予測は，ベイズ最適化 (Bayesian optimization) [196, 197] との類似で意思決定プロセスを提案することがの可能であり，非常に有用であると考えられる．そこで本章では多くの不確かさ考慮手法の中でも特にベイズ深層学習に着目し，手法提案および実験を行う．

本章ではベイズ深層学習の金融市場予測に対する有効性確認のため，以下の 3 つの実験を行う．

1. 深層学習における不確かさ考慮手法の比較検討
2. ベイズ深層学習を用いた株価動向予測
3. 投資シミュレーションを用いた不確かさ評価の妥当性検証

1. 深層学習における不確かさ考慮手法の比較検討では金融市場予測に限らない予測タスク全般について，ベイズ深層学習および他の代表的な不確かさ評価手法を比較し，予測の不確かさ考慮に対する有効性の確認を行った．2. ベイズ深層学習を用いた株価動向予測では短期の市場動向予測タスクに対しベイズ深層学習を適用し，予測の不確かさを考慮しない通常の深層学習モデルとの比較から，予測不確実性の妥当性検討を行った．3. 投資シミュレーションを用いた不確かさ評価の妥当性検証では，ベイズ深層学習により得られる

幅のある予測を活用した投資意思決定アルゴリズムを提案し，不確かさを考慮しない手法に対する有効性を確認した．

4.1　ベイズ深層学習

4.1.1　ベイズ深層学習の概要

ベイズ深層学習 (Bayesian Neural Networks, BNNs) [198, 199] は，深層学習の枠組みでベイズ推論 (Bayesian inference) [200] を行う手法である．通常の深層学習では，一つの入力 X に対し決定論的に出力 y が得られる．

$$y = f_\theta(X) \tag{4.1}$$

ここで，$f_\theta(\cdot)$ は深層学習モデル，θ は $f_\theta(\cdot)$ のパラメータを表す．

一方 BNNs では，モデルのパラメータ (多くの場合線形変換の重みおよび定数項) を確率分布で定義することにより，モデルの出力 y の事後分布 $p(y|X,\theta)$ を近似する．

$$p(y|X,\theta) = f_\theta(X) \tag{4.2}$$

多くの BNN モデルは事後分布の sampler として定義される．最も単純な BNN の例として，以下の式で表現される単入力単出力 BNN を考える．

$$y = \tanh(wx + b) \tag{4.3}$$

上式はニューラルネットワークとして解釈すると，一次元の入力 x に対し線形変換を行い．活性化関数として双曲線正接 (hyperbolic tangent) 関数を採用したものである．モデルのパラメータとしては線形変換の重み w および係数 b が存在し，これらが以下に示す正規分布に従うと仮定する．

$$w \sim N(1, 0.1^2) \tag{4.4}$$

$$b \sim N(0, 0.1^2) \tag{4.5}$$

一般的に，BNN モデルの出力値分布は解析的に求められない．したがって，予測時に

はそれぞれパラメータ分布から値をサンプリングし，得られた値をもとに出力値の予測を行う．例として，同一の入力 x に対し 10 回の予測を行った結果を図 4.1 に示す．図 4.1 を見ると，同一の入力 x に対しても幅を持った出力 y の値が得られている事がわかる．本例では正規分布ノイズに双曲線正接関数による変換を施した出力分布が得られるが，より多層のニューラルネットワークを用いることにより，複雑な出力分布を表現することができる．

BNN モデルの学習方法については 4.1.2 章で解説するが， 大まかには出力分布の教師データと実際のデータ分布の間の Kullback-Leibler divergence を最小化するように学習を行うことで，出力分布の最適化が行われる．

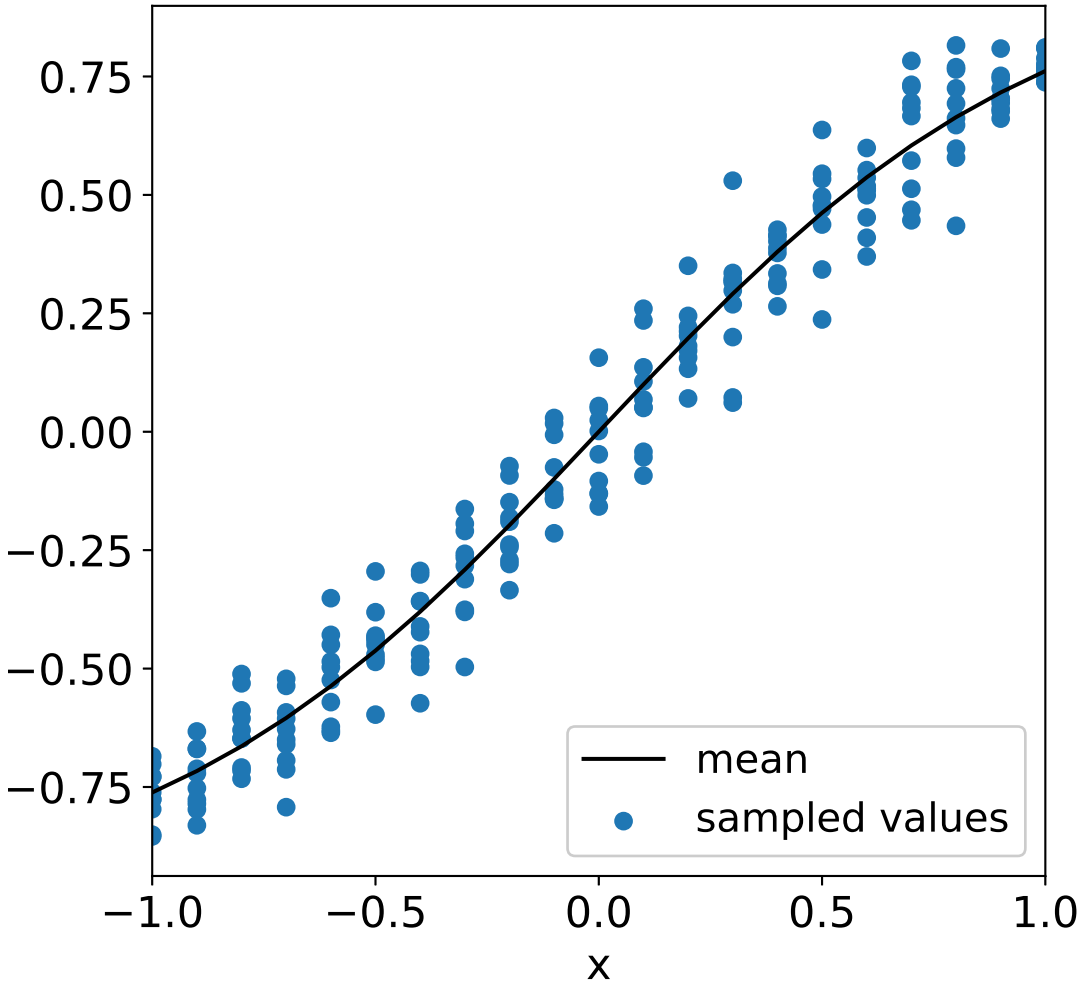


図 4.1: BNN を用いた予測の例．パラメータ w, b が正規分布に従う 単入力単出力 BNN $y = \tanh(wx + b)$ $(w \sim N(1, 0.1^2), b \sim N(0, 0.1^2))$ について，同一の入力に対し 10 回の予測を行った結果を示している．BNN を用いることで，幅を持った予測が行えていることがわかる．

4.1.2 ベイズ深層学習モデルの学習

BNN モデルの学習方法については複数の手法が提案されている．マルコフ連鎖モンテカルロ法 (Markov chain Monte Carlo methods, MCMC) [201] や ラプラス近似 [202] による学習は計算量や表現力に課題があるため，多くの手法では変分推論を用いた最適化を行っている．変分推論による最適化では，BNN による近似分布 $q(w|\theta)$ と目標の事後分布 $q(w|D)$ の Kullback-Leibler (KL) divergence [203] を最小化することを目的とし，問題を KL divergence の evidence lower bound (ELBO) [204] 最大化に置き換えてパラメータ更新が行われる．

BNN のパラメータ更新を効率的に行うためには，ELBO のパラメータに対する勾配を求める必要がある．実用的な勾配の計算手法としては reparameterization trick を用いたものが知られている [205, 206]．この手法では，パラメータの事前分布 (多くの場合 reparameterization が容易な正規分布が用いられる) からサンプリングを行い，以下に基づき勾配を計算する．

$$\frac{\partial}{\partial\theta}\mathcal{L}(D;\theta) = \frac{\partial}{\partial\theta}E_{q(w|\theta)}[f(w,\theta)] = \frac{\partial}{\partial\theta}E_{q(\epsilon)}\left[\frac{\partial f(w,\theta)}{\partial\theta}\frac{\partial w}{\partial\theta} + \frac{\partial f(w,\theta)}{\partial\theta}\right] \tag{4.6}$$

このようにパラメータ事後分布からのサンプリングを用いて下限推定量を計算する手法を stochastic gradient variational bayes (SGVB) [205, 207] と呼び，勾配計算時のミニバッチ数を大きくすることで，サンプリング回数を 1 回にしても安定に学習が進行することが知られている．

4.1.3 Variational dropout

変分推論を行うもう一つの方法として，dropout [208] を変分推論として解釈する variational dropout [209] が存在する．Variational dropout は， Bayesian ではないという批判も存在する [210] ものの，広く BNN 研究で活用されている．Variational dropout では，入力に平均 1 の正規分布ノイズを乗算する Gaussian dropout をネットワークの学習時だけでなく予測時にも用いる．例として，全結合層 $B = AW$ において，Gaussian dropout は以下のように表すことができる．

$$B = (A \odot \Xi)W \tag{4.7}$$

ここで，$A \odot \Xi$ は入力 A とノイズ Ξ のドット積を表す．ノイズ Ξ は A と同数のパラメータ $\xi_{m,i}$ からなり，各 $\xi_{m,i}$ は以下の正規分布からサンプリングされる．

$$\xi_{m,i} \sim \mathcal{N}(1, \alpha = \frac{p}{1-p}) \tag{4.8}$$

Gaussian dropout 前後で平均を維持するため，正規分布の平均は 1 である．また，正規分布の分散 α は，一般的にニューラルネットワークで用いられる binary dropout において，dropout を行う確率である dropout rate p を用いて表すことができることが知られている [208]．また，重み W に正規分布からサンプリングされたノイズを乗算することと，以下の正規分布から重みをサンプリングすることと等価である．

$$w_{i,j} = \theta_{i,j}\xi_{i,j} = \theta_{i,j}(1 + \sqrt{\alpha}\epsilon_{i,j}) \sim \mathcal{N}(\theta_{i,j}, \alpha\theta_{i,j}^2) \tag{4.9}$$

ここで，$\epsilon_{i,j}$ は標準正規分布ノイズである．また，上記のパラメータ化には α が大きくなると下界の勾配のノイズが大きくなるという問題がある．$w_{i,j}$ の $\theta_{i,j}$ に関する偏微分は以下のように表される．

$$\frac{\partial w_{i,j}}{\partial \theta_{i,j}} = 1 + \sqrt{\alpha_{i,j}}\epsilon_{i,j} \tag{4.10}$$

上式の勾配は α の増加に伴い，標準正規分布からサンプリングされたノイズ $\epsilon_{i,j}$ の影響が大きくなってしまう．そこで，以下のような式変形をすることで下界の勾配を減らし，

α を無限大まで大きくできる．

$$w_{i,j} = \theta_{i,j}(1 + \sqrt{\alpha}\epsilon_{i,j}) = \theta_{i,j} + \sigma_{i,j}\epsilon_{ij} \tag{4.11}$$

ただし，$\sigma_{i,j}^2 = \alpha_{i,j}\theta_{i,j}^2$ である．以上のような変形は Molchanov らにより提案されている [211]．実装時は正規分布の平均 θ および 標準偏差 σ をパラメータとすれば良い．なお，σ には非負の条件があるため，$\log(\sigma^2)$ としてパラメータ化することも行われる．

さて，variational dropout モデルを学習する場合，学習時の共分散を 0 とするためには，入力バッチ数だけ重みをサンプリングする必要がある．入力 A の次元を $M \times n_{\text{in}}$，重み W の次元を $n_{\text{in}} \times n_{\text{out}}$ とすると，サンプリング数は $M \times n_{\text{in}} \times n_{\text{out}}$ となる．ここで，サンプリングを各重みについてではなく，出力 B について行うことを考えると，サンプリング数は $M \times n_{\text{out}}$ で済み，サンプリング数を大幅に抑えることが可能となる．以下のような手法を local reparametarization trick [205] と呼ぶ．Local reparametarization trick におけるサンプリングの式は以下のように書ける．

$$b_{m,j} = \gamma_{m,j} + \sqrt{\delta_{m,j}}\epsilon_{m,j} \tag{4.12}$$

ここで，$\epsilon_{m,j}$ は正規分布ノイズであり，$\gamma_{m,j}$ および $\delta_{m,j}$ は以下のように計算される．

$$\gamma_{m,j} = \sum_{i=1}^{n_{\text{in}}} a_{m,i}\theta_{i,j} \tag{4.13}$$

$$\delta_{m,j} = \sum_{i=1}^{n_{\text{in}}} a_{m,i}^2\sigma_{i,j}^2 \tag{4.14}$$

また，local reparametarization trick は畳み込み層にも適用可能である [211]．畳み込み層の場合の $\gamma_{m,j}$ および $\delta_{m,j}$ の計算は以下のように行われる．

$$\gamma_{m,j} = \text{vec}(A_m * \theta_k) \tag{4.15}$$

$$\delta_{m,j} = \text{diagvec}(A_m^2 * \sigma_k^2) \tag{4.16}$$

ここで，vec は行列のベクトル化処理を，diagvec は行列の対角成分を，$A_m * \theta_k$ および $A_m^2 * \sigma_k^2$ は畳み込み処理を表す．畳み込み処理は数式では以下のように計算される．

$$(X * W)_{i,j} = b + \sum_{l=0}^{s} \sum_{m=0}^{s} w_{l,m} x_{i+l,j+m} \tag{4.17}$$

ここで，$(X * W)_{i,j}$ は入力 X に対し $s \times s$ 行列 W を用いて畳み込み処理を行った後の行列の i, j 成分を，b は定数項を，$w_{l,m}$ は W の l, m 成分を，$x_{i+l,j+m}$ は X の $i+l, j+m$ 成分を表す．

以上により variational dropout のパラメータ化に成功したが，本ネットワークの学習には Kullback-Leibler (KL) divergence の近似を行う必要がある．KL divergence の近似関数としては，Kingma らにより提案されたもの [212] もあるが，α が 1 以上では良い近似にはなっていない．前述の Molchanov らによって，以下のような 1 以上の α について実験的に良い近似となる式が提案されている [211].

$$-D_{\mathrm{KL}}(q(w_{i,j}|\theta_{i,j}, \alpha_{i,j})||p(w_{i,j})) \approx k_1 \sigma(k_2 + k_3 \log(\alpha_{i,j})) - 0.5 \log(1 + \alpha_{i,j}^{-1}) + C \tag{4.18}$$

ここで，$k_1 = 0.63576$，$k_2 = 1.87320$，$k_3 = 1.48695$ である．以上のような手法を sparse variational dropout (SVD) と呼ぶ．

Variational dropout の発展研究は sparse variational dropout 以降も多く提案されている．例としては，Kharitonov らの経験ベイズを用いた手法 [213]，Liu らの variational Bayesian dropout [214]，Wang らの Si-vdnas [215] などが存在する．Variational dropout により強化学習の性能を向上させる研究も存在する [216].

4.2 実験:深層学習における不確かさ考慮手法の比較検討

前述の通り，深層学習における不確かさ評価手法はBNN以外にも数多く提案されており，様々なタスクで成果を挙げている．本章ではそれらの手法の有効性を比較するため，手書き数字画像データを用いた簡単な画像識別タスクを用意し，手法間の比較を行った．その際，画像の識別精度だけでなく，予測スコアの妥当性やデータセットに存在しない特徴を持つ外挿データに対する予測結果を比較し，不確かさ予測の妥当性を検証している．

本実験において金融時系列ではなく画像データを用いている理由としては，予測タスクとして広く用いられている点が挙げられる．1.1 章でも解説したとおり，金融時系列は非常に予測が困難なタスクであり，そのような予測が困難な対象で手法間比較を行うのは適切ではない．一方画像識別タスクは広く研究されており，高精度で予測を行う手法が多く提案されている [217]．加えて，画像データに対しては本研究で対象とするような外挿データの作成も容易である．一方で，画像識別タスクで利用した深層学習モデルを金融市場予測に流用することはできず，特徴量設計やネットーワーク構成等，効果的な予測を行うためのモデル設計が必要となる．

本研究の成果は2019年度人工知能学会全国大会で発表している [218]．

4.2.1 予測タスクとベースライン

予測タスクとしては，手書き数字画像データ MNIST [219] を用いて学習したモデルに数字以外の画像データを入力し，モデルの予測結果を調べた．予測モデルの基本構造としては，画像識別タスクで広く用いられている畳み込みニューラルネットワーク (Convolutional neural networks, CNN) [220, 181, 221] を用いた．CNN は予測が入力データに対し位置普遍性を有するため特に画像認識や系列データの予測に優れ，pooling や batch normalization [222] といった手法と組み合わせることで高い予測精度を実現できる．

手法比較のベースラインとして用いるCNNのネットワーク構造を図 4.2 に示す．ベースラインは，Conv(畳み込み) 層を，MaxPool(最大値プーリング) 層を，Flatten(変形) 層を，Dense(全結合) 層で構成されている．各 Conv 層は 3×3 のフィルター64個で構成され，活性化関数には ReLU [223] を使用した．各 MaxPool 層はフィルターサイズ 3×3，ストライド2であり，入力画像の半分のサイズの出力が得られる．Dense 層からは10次元の出力が得られ，そのそれぞれが0から9までの数字に対応した logit(論理推定量) を

表す．以上のネットワークを，一般的な cross entropy loss [224] を損失関数とし，Adam optimizer [124] を用いて学習を行った．

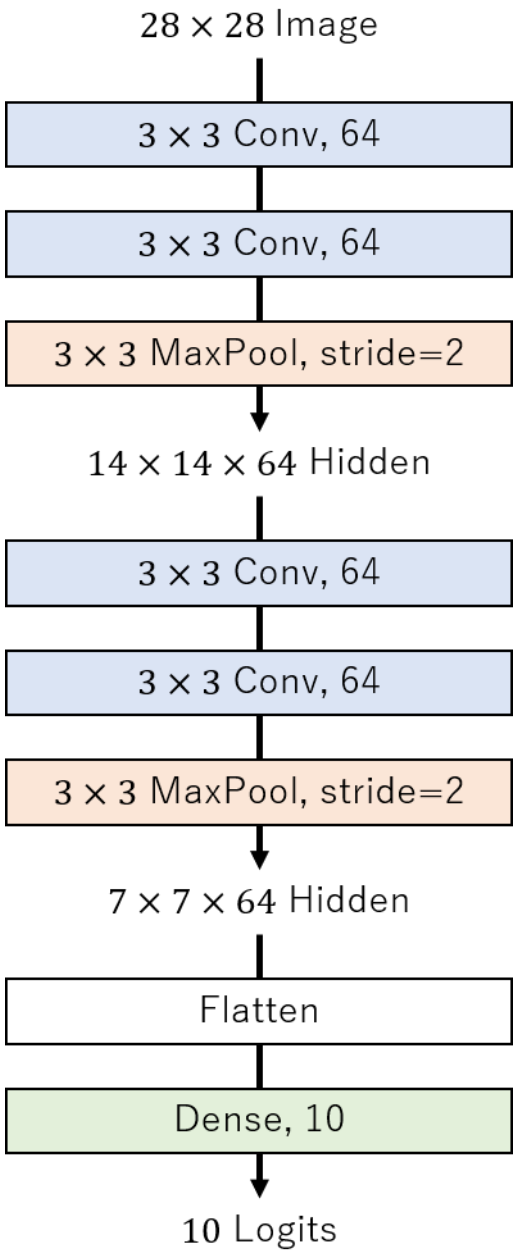


図 4.2: MNIST 予測タスクでベースとして用いた CNN のネットワーク構造．ネットワークを構成する層 (layer) を長方形の枠で表しており，Conv が畳み込み (convolutional) 層を，MaxPool が最大値プーリング (maximum pooling) 層を，Flatten が入力の変形層を，Dense が全結合 (fully connected) 層を示している．モデルは 28×28 次元の画像入力を受け取り，0 から 9 までのそれぞれの数字に対応した 10 次元の logit が得られる．

4.2.2　ベースラインモデルの問題点

ベースラインとして用いるモデルは入力画像に対し 10 次元の logit を計算し，logit が最大となるクラスを予測結果とする．ここで，予測の信頼性あるいは不確かさを評価するために 以下に示す softmax 関数による logit の正則化が行われる．

$$p_i = \frac{\exp l_i}{\sum_j \exp l_j} \tag{4.19}$$

ここで，l_i が i 番目のクラスに対する logit の値を表す．Softmax 関数は損失関数と親和性があり，その適用により，logit の和を 1 に揃え，各クラスに対応する擬似的な確率として扱うことができる．

学習したモデルは当然，学習に用いたデータに類似した手書き数字データに対しては正確な予測を行うことができる．しかし，数字以外の文字やノイズ画像等，学習データと異なる特徴を持つ画像に対しては誤った予測を行ってしまう．このような学習データ，対象データと異なる特徴を持つデータを外挿データ (out-of-distribution data) [225, 33] と呼ぶ．例として 3 種類の外挿データに対し，ベースラインモデルで予測を行った結果を図 4.3 に示す．ここでは外挿データとしてアルファベットの A，ギリシア文字の π に対応する手書き文字，及び一様分布に基づきノイズ画像を生成する white noise の 3 つを入力しており，Softmax 関数適用後の logit の値を confidence (予測信頼性を示す値) として使用している．図 4.3 を見ると，これらの外挿データに対しても，ベースラインモデルの予測結果では confidence が 1 に近い．すなわち，高い確信を持って誤った予測を行ってしまったことがわかる．このように，通常のニューラルネットワークモデルは予測結果の信頼性についての議論が不十分であり,「間違っているかもしれない」予測を見抜くことが困難であることがわかる．

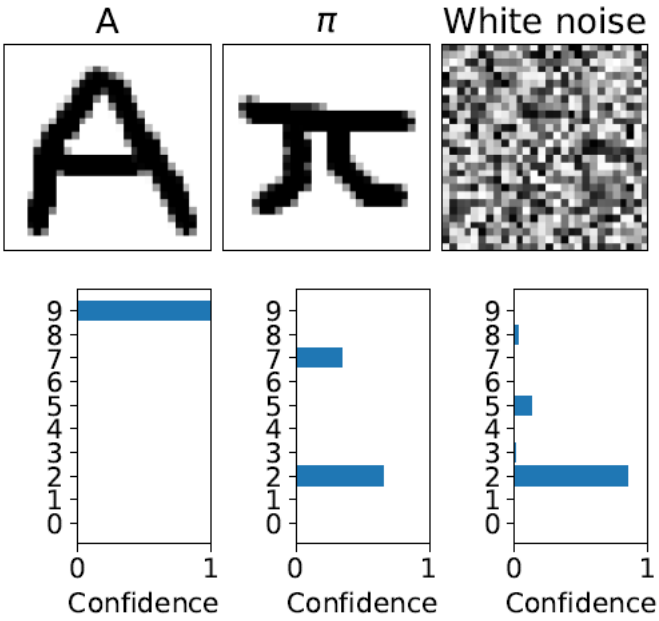


図 4.3: MNIST 予測タスクでベースとして用いた CNN の外挿データに対する予測例．外挿データとしてアルファベットの A，ギリシア文字の π に対応する手書き文字，white noise の 3 種類の画像を使用している．Confidence としては softmax 関数適用後の logit の値を使用している．予測結果より，ベースラインモデルは外挿データに対し高い確信を持って誤った予測を行ってしまったことがわかる．

4.2.3　BNN モデルの構成

4.2.2 章の問題点を解決するため，BNN の導入を行った．BNN としては 4.1.3 章で解説した Sparse variational dropout を採用し，baseline モデルと同じ構成のネットワークに Gaussian dropout を導入することでモデルの Bayesian 化を行った．

実験に使用した　BNN モデルのネットワーク構成を 図 4.4 に示す．ベースラインモデルの Conv および Dense 層に sparse variational dropout (SVD) を導入することで，ネットワークの BNN 化を行っている．入力は 28×28 のモノクロ画像であり，はじめの sparse variational dropout convolutional (SVDConv) 層入力時には $28 \times 28 \times 1$ 次元に変形される．各 SVDConv 層は 3×3 のフィルター 64 個で構成され，入出力の形状を変更しないように画像のパディングを行っている．SVDConv 層を 2 回通した後にストライド 2 の maximum pooling (MaxPool) 層を適用し，画像のサイズ縮小を行っている．以上の処理を 2 回行い $7 \times 7 \times 64$ の特徴量を得た後特徴を 3136 次元のベクトルに形状変更，最終的に sparse variational dropout fully connected (SVDDense) 層にて 10 次元の出力が得られる．

BNN モデルの予測には，通常の深層学習モデルと同様の順伝播に加え，標準正規分布からのノイズのサンプリングが必要となる．ノイズのサンプリングは各 SVD 層の出力次元分必要となるため，必要なサンプリング数は

$$28 \times 28 \times 64 \times 2 + 14 \times 14 \times 64 \times 2 + 10 = 125450 \tag{4.20}$$

となる．非常に多くのサンプリングが必要となるが，local reparametarization trick の適用により，深層学習モデルのパラメータ数自体は大きく削減できている．BNN モデルの学習は 4.1.3 章で解説した損失関数により行われる．

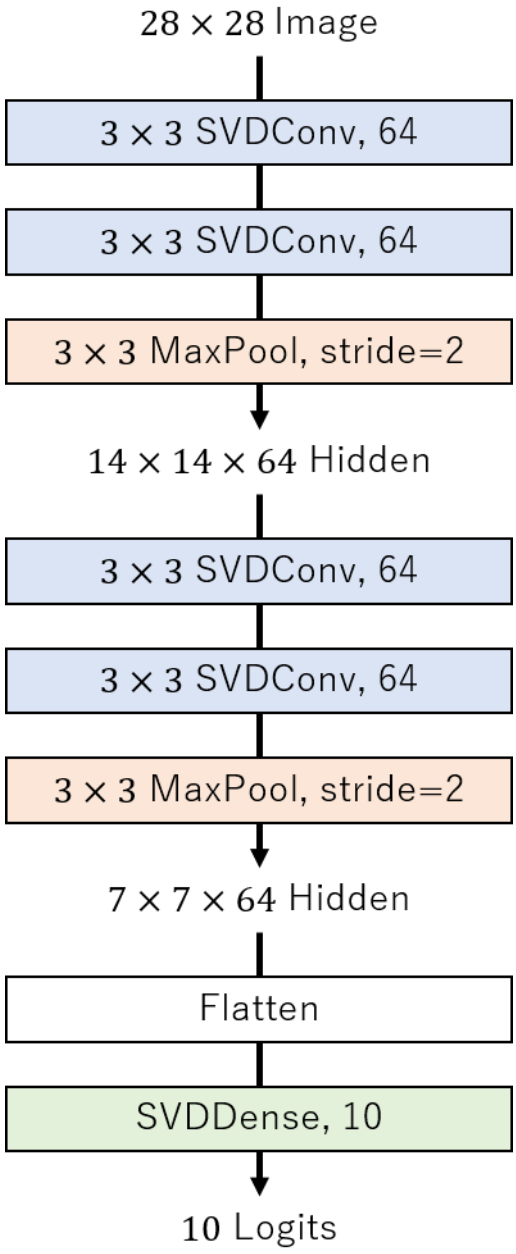


図 4.4: MNIST 予測タスクで用いた BNN ネットワーク構造．ネットワークを構成する層 (layer) を長方形の枠で表しており，SVDConv が sparse variational dropout convolutional 層を，MaxPool が最大値プーリング (maximum pooling) 層を，Flatten が入力の変形層を，SVDDense が sparse variational dropout fully connected 層を示している．モデルは 28×28 次元の画像入力を受け取り，0 から 9 までのそれぞれの数字に対応した 10 次元の logit が得られる．予測の度に SVDConv および SVDDense 層からサンプリングが行われるため，同一の入力に対して幅のある予測を行うことが可能である．

4.2.4　比較手法

本実験では BNN モデルに加え，近年提案されている不確かさ評価手法を比較手法として選択した．選択した手法を以下に示す．

1. Sigmoid activation

   ベースラインモデルで用いた softmax activation は，logit の値の相対的な大きさを評価する． それに対し，logistic regression [226, 227, 228] 等で用いられる sigmoid activation $\frac{1}{1+\exp -l}$ は，予測サンプルが各クラスに属する確率を絶対的に評価する．したがって，外挿データに対しても適切な評価 (全クラスに対し confidence の値が 0) を行うことが理論上可能である．しかし，各クラスに対して独立に学習を行うため学習効率が低くなる，confidence の和が 1 を大きく超えることが起こりうる，といった問題点も存在する．

2. Temperature scaling

   Guo [229] らは，ニューラルネットワークの出力値に calibration (較正) を行うことにより，不確かさ評価の妥当性が向上することを報告している．中でも，softmax あるいは sigmoid 関数を適用する前に logit を定数倍する temperature scaling が最も良好な成績を示している．Temperature scaling はボルツマン分布の温度 (あるいは逆温度) に由来している．

3. Label smoothing

   Label smoothing [230] は，識別モデル学習時に正解ラベル $t \in 0, 1$ に線形変換 $t' = (1-\alpha)t + \frac{\alpha}{K}$ を施すことにより，過学習 [231] が抑制する手法である．通常の正解ラベルを用いて，cross entropy $t\log(p) + (1-t)\log(1-p)$ の最小化を目指す場合，softmax activation あるいは sigmoid activation 後の予測確率 p は 0 または 1 に近づく，すなわち logit は無限に大きな (あるいは小さな) 数になってしまう．Label smoothing の適用により，logit の目標値に上限・下限値が設けられ，過剰なパラメータの学習が抑制されるため，予測信頼性が改善されることが報告されている．

4. Evidential deep learning

   Evidential deep learning [194] は，Sensoy らにより提案された手法であり，通常の識別モデルと異なりデータの Bayesian evidence $e_i \geq 0$ [200, 232] をニューラルネットワークで近似する．非負値を予測するため，出力値の活性化関数には ReLU ある

いは softplus 関数を用いる．予測された Bayesian evidence を元に，各クラスに対応する confidence c_i および uncertainty(どのクラスにも属さない確率) u は以下のように計算される．

$$c_i = \frac{e_i}{\sum_j^K (e_j + 1)} \tag{4.21}$$

$$u = \frac{K}{\sum_j^K (e_j + 1)} \tag{4.22}$$

サンプルに対する evidence e_i が大きいほど確信を持った予測を行う．学習は出力分布に Dirichlet distribution [233] を仮定し損失関数が定義されている．

各比較手法のネットワーク構造としては，図 4.2 で示したベースラインモデルと同じもの使用した．Sigmoid activation， temperature scaling および label smoothing は学習後のベースラインおよび BNN モデルに適用した．Evidential deep learning については，出力層の活性化関数に softplus 関数を採用し学習を行った．

4.2.5　不確かさ評価の定量化

深層学習モデルの不確かさ評価について検証するためには，4.2.2 章のような視覚的な比較だけでなく，定量的な指標による評価が必要である [229, 234]．本実験では様々な手法間比較が可能な指標として,expected calibration error (ECE) [235] を用いた．ECE は以下の式で計算される．

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)| \tag{4.23}$$

ここで，M が confidence の値で分けられた batch 数，B_m が m 番目の batch，$|B_m|$ が B_m のサンプル数，n が全サンプル数，$\mathrm{acc}(B_m)$ が B_m の 正解率 (accuracy)，$\mathrm{conf}(B_m)$ が B_m の予測信頼性の平均値である．ECE は confidence と accuracy の乖離度を表し，confidence と accuracy が比例するほど小さな値をとる．ECE を用いることで，定量的にモデルの不確かさ評価の妥当性を評価することができる．

4.2.6 実験結果

BNN および比較手法モデルをを用いて MNIST 識別モデルを学習し，検証用データに対し精度および ECE を計算した．精度指標としては，識別タスクの評価に広く用いられている F1 score [236] を使用した．結果を表 4.1 に示す．ここで，temperature scaling については baseline モデルと精度が変化しないため記載していない．

表 4.1 を見ると，F1 score については各手法で大差ないことがわかる．MNIST が予測タスクとしては単純であったため，すべての手法で良好な予測が行えていることがわかる．ただし，evidential deep learning のみ他の手法より若干精度が低い．これは同手法の学習効率の低さが原因となっていることがわかる．

対して ECE については，label smoothing を施さないベースラインおよび BNN モデルが良好な成績を残している．活性化関数については，label smoothing を施した場合のみ sigmoid activation が良い値を出している．しかし，これらは学習データと似た特徴を持つ内挿データに対しての値であり，外挿データに対する不確かさ評価の妥当性は検証できていない．

外挿データに対する不確かさ評価を検証するため，数字画像を回転させ擬似的な外挿データを作成し，各モデルに入力して予測結果を比較した．入力として使用した画像を図 4.5 に示す．本実験では，数字の 6 に対応した手書き画像を一定角度ずつ回転させることで，内挿から外挿に向け変化する画像系列を作成した．図 4.5 に表示される画像以外にも様々な角度の回転画像を作成し入力データとしている．元画像および，数字の 9 に近い 180 度回転画像は内挿データだと考えられ，モデルが小さな不確かさ (大きな confidence) を予測することが期待される．対して，それらから離れた画像になるほど外挿データに近づくと考えられ，モデルが大きな不確かさ (小さな confidence) を予測することが期待される．

各モデルの予測結果を図 4.6，4.7，4.8 に示す．ベースラインモデル，BNN モデルについては softmax activation および sigmoid activation，temperature scaling の有無，label smoothing の有無を変化させ実験を行った．Evidential deep learning モデルについては前述の手法は適用が困難なため通常のモデルの予測結果のみを示している．

予測結果を見ると，外挿データに対する各モデルの不確かさ評価が大きく異なることがわかる．図 4.6 を見ると正則化を施さない softmax activation モデルは全角度通して大きな confidence の値を予測している．Softmax activation は実装上外挿データを仮定しない

ため，学習データと異なる特徴を持つデータに対しても大きな confidence を予測してしまう場合が多いと考えられる．これらの問題は temperature scaling および label smoothing といった正則化の導入によりある程度改善され，元画像および 180 度回転画像から角度が離れるにつれ confidence の値が小さくなっている．しかし，後に示す手法に比べると confidence が角度の変化に対し大きく上下動しており，予測が不安定であることが考察される．

次に，図 4.7 に示す sigmoid activation モデルでは，正則化を施さない場合でも良好な予測が行えている．正則化を施した場合は confidence の予測がなめらかになっている．図 4.6 と比べると confidence の上下動も比較的少なく，安定した予測ができていることがわかる．

最後に，図 4.8 に示す evidential deep learning モデルは，sigmoid activation モデルより更に良好な予測が行えている．大きな confidence が期待される画像に対しては 1 に近い confidence が予測され，逆に小さい confidence が期待される画像に対しては 0 に近い confidence が予測されている．さらに，180 度前後の回転画像については，0 度付近に比べ若干 confidence の値が低くなっており，数字の 9 の画像との小さな違いを識別できていることがわかる．

図 4.6，4.7，4.8 の実験で用いた画像は 28×28 の解像度の低い画像であり，回転処理により画像が大きく変化したため，予測される confidence の値が大きく上下動している．解像度の大きい画像を用いて実験を行うことで，より滑らかな confidence の変化が観測でき，モデルの画像知覚に関する有用な知見の抽出が期待される．

表 4.1: 不確かさ評価の比較結果. MNIST データセットを用いて各モデルを学習し，MNIST の検証データについて F1 score および ECE を計算した．CNN がベースラインモデルを， VDCNN が sparse variational dropout BNN モデルを，Evidential CNN が evidential deep learning モデルを表す.

Method	Activation function	F1 score	ECE
CNN	Softmax	0.9947	0.0042
CNN	Sigmoid	0.9947	0.0042
CNN + Label smoothing	Softmax	0.9946	0.0625
CNN + Label smoothing	Sigmoid	0.9948	0.0374
VDCNN	Softmax	0.9948	0.0035
VDCNN	Sigmoid	0.9951	0.0043
VDCNN + Label smoothing	Softmax	0.9957	0.0656
VDCNN + Label smoothing	Sigmoid	0.9954	0.0422
Evidential CNN		0.9642	0.0208

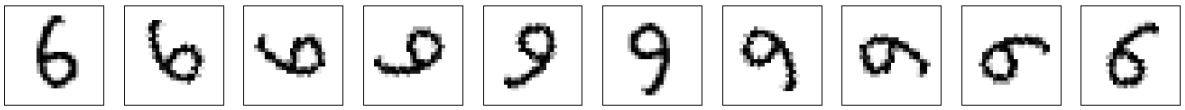


図 4.5: 外挿データに対する不確かさ評価の妥当性検証に使用した入力画像データ．数字の 6 に対応した手書き画像を一定角度ずつ回転させることで，内挿から外挿に向け変化する画像系列を作成した．元画像および数字の 9 に近い 180 度回転画像に対しは小さな不確かさ (大きな confidence) が期待され，それらから離れた角度になるほど大きな不確かさが期待される．sparse variational dropout BNN

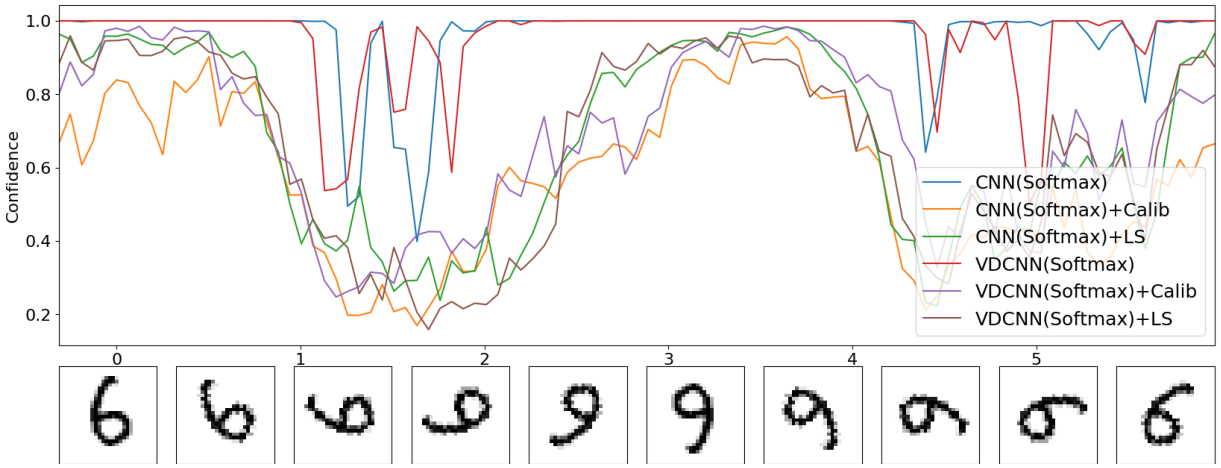


図 4.6: Softmax activation モデルの回転画像に対する予測結果．CNN がベースラインモデルを， VDCNN が sparse variational dropout BNN モデルを，calibration が temperature scaling を，LS が label smoothing を表す．

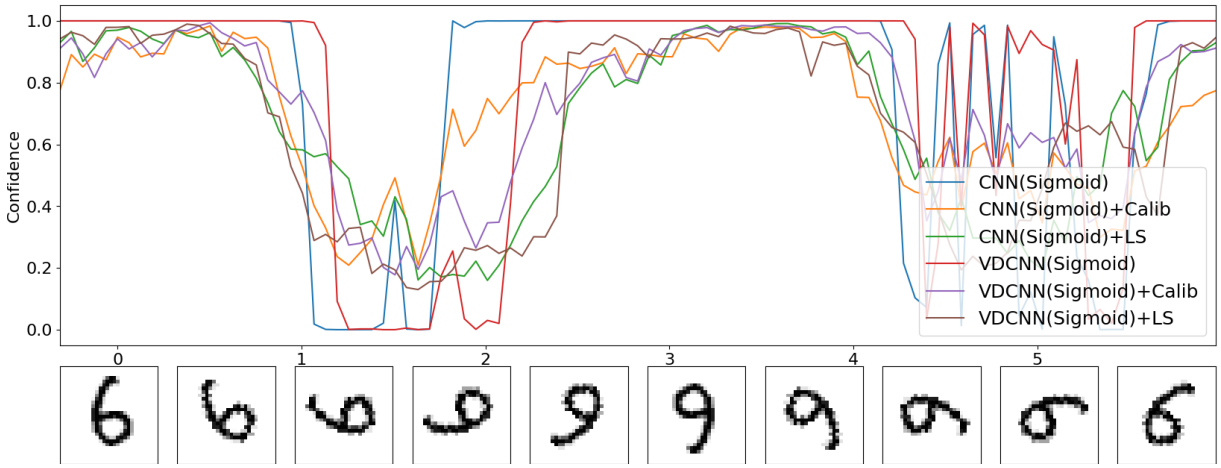


図 4.7: Sigmoid activation モデルの回転画像に対する予測結果．CNN がベースラインモデルを， VDCNN が sparse variational dropout BNN モデルを，calibration が temperature scaling を，LS が label smoothing を表す．

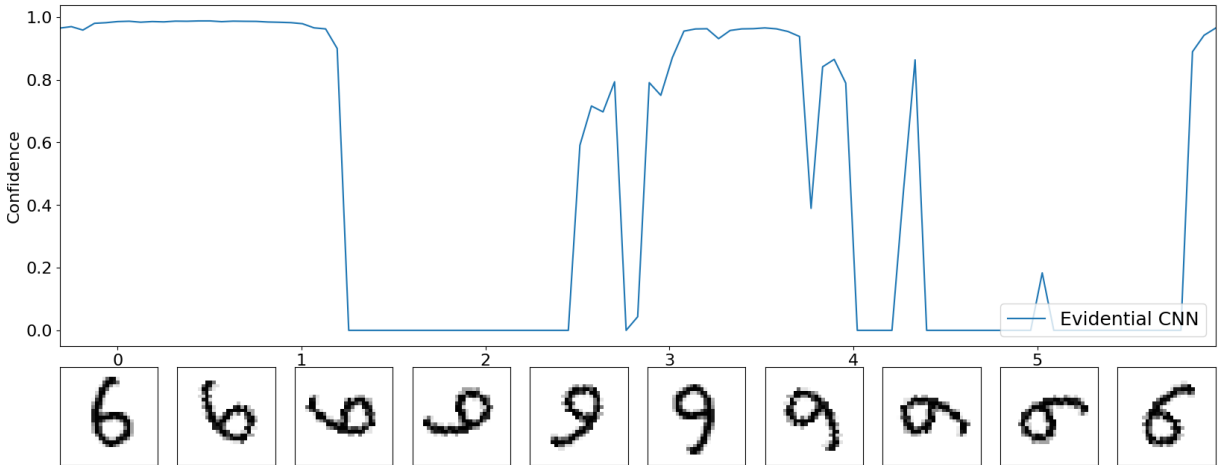


図 4.8: Evidential deep learning モデルの回転画像に対する予測結果．

4.3　実験:BNN を用いた株価動向予測

4.2 章では画像識別タスクに対し BNN の有効性を確認した．続いて本章では，金融市場の過去取引データを用いた価格動向予測タスクに対し BNN を適用し，予測精度および予測信頼性の比較検討を行う．

本研究の成果は International Conference on Decision Economics (DECON) (2019) で発表している [237].

4.3.1　データセット

本実験では FLEX FULL historical data をデータセットとして利用した．FLEX FULL historical dataset は 東京証券取引所から提供されている注文および約定系列データである [1]．高頻度取引 (high frequency trading, HFT) [88, 89, 90] が発達した現在では，ミリ秒単位で大量の注文が市場に登録されている．FLEX FULL についても，大きな銘柄では一日に 10 万件以上の注文・約定データが存在する．FLEX FULL historical data は高頻度取引データであり，本実験の予測タスクとしても短期 (1 秒未満から数分先) の市場予測を行う．短期の市場動向はある程度現在の市場状態から予測できる仮定のもと，モデルの説明変数には FLEX FULL データから取得できる市場状態のみを用いた．

FLEX FULL の注文および約定データの例を図 4.9 に示す．図 4.9 のようなデータが各注文，約定ごとに存在している．各データには日付 (time)，銘柄 (code) といった基本情報に加え，注文・約定情報 (message)，市場価格 (market_price)，最良気配値 (best_ask, best_bid)，仲値 (mid_price)，板情報 (buy_book，sell_book) が含まれている．図 4.9 のデータには以下の 3 つの注文・約定情報が存在している．

1. 'tag': '1P', 'price': '20520': 20520 円での約定
2. 'tag': 'VL', 'volume': '200': 200 株の約定
3. 'tag': 'QB', 'price': '20520', 'qty': '200-¿0': buy_book の 20520 円の数量が 200 から 0 に減少

[1] https://www.jpx.co.jp/english/markets/paid-infoequities/ realtime/index.html

すなわち，数量 200 の売り成行注文があり，買いの最良気配注文が約定したことがわかる．これらの情報から説明変数および目的変数を作成し解析を行った．

{'Data': {'time': '09:54:18.711868', 'code': '9022', 'status': '20', 'message': [{'tag': 'QS', 'price': '19960', 'qty': '900->800'}], 'market_price': '19950', 'best_ask': '19955', 'best_bid': '19945', 'mid_price': '19950', 'buy_book': {'19945': '200', '19940': '400', '19935': '800', '19930': '600', '19925': '700', '19920': '600', '19915': '700', '19910': '1400', '19905': '1300', '19900': '500', '19895': '400', '19890': '500', '19885': '200', '19880': '100', '19875': '300', '19870': '300', '19865': '300', '19860': '100', '19855': '400', '19850': '300', '19845': '200', '19840': '200', '19835': '300', '19820': '100', '19815': '100', '19800': '200', '19795': '100', '19790': '100', '19785': '300', '19780': '200', '19775': '400', '19770': '200', '19765': '200', '19760': '500', '19755': '100', '19745': '200', '19740': '100', '19735': '1900', '19730': '100', '19725': '100', '19720': '100', '19715': '100', '19710': '200', '19705': '100', '19700': '100', '19695': '200', '19640': '500', '19555': '100', '19550': '500', '19535': '100', '19520': '400', '19500': '300', '19490': '200', '19480': '100', '19460': '200', '19450': '400', '19430': '100', '19400': '1100', '19385': '100', '19380': '100', '19350': '100', '19325': '100', '19320': '100', '19310': '300', '19300': '400', '19290': '200', '19280': '400', '19270': '200', '19260': '200', '19200': '1400', '19160': '400', '19155': '100', '19150': '100', '19130': '300', '19100': '600', '19080': '100', '19040': '500', '19030': '100', '19000': '2100', '18970': '100', '18950': '100', '18920': '400', '18910': '100', '18900': '500', '18880': '100', '18800': '600', '18700': '100', '18600': '100', '18500': '1400', '18460': '100', '18400': '200', '18300': '100', '18200': '200', '18100': '300', '18005': '100', '18000': '900', '17900': '100', '17820': '100', '17800': '200', '17700': '1500', '17655': '100', '17525': '200', '17500': '100', '17430': '100', '17150': '100', '17000': '400', '16200': '100', '16000': '1100', '15800': '100', '15615': '200'}, 'sell_book': {'19955': '100', '19960': '800', '19965': '800', '19970': '900', '19975': '900', '19980': '1600', '19985': '2100', '19990': '600', '19995': '700', '20000': '3300', '20005': '200', '20010': '500', '20015': '100', '20025': '100', '20030': '300', '20035': '200', '20045': '100', '20050': '400', '20065': '100', '20080': '100', '20100': '1300', '20105': '100', '20120': '400', '20130': '100', '20150': '100', '20155': '100', '20160': '1800', '20180': '100', '20195': '100', '20200': '1500', '20205': '700', '20210': '100', '20220': '300', '20240': '400', '20270': '200', '20290': '100', '20295': '100', '20300': '800', '20335': '100', '20350': '2200', '20355': '100', '20360': '500', '20390': '200', '20400': '400', '20420': '100', '20425': '200', '20450': '300', '20480': '500', '20490': '100', '20495': '100', '20500': '4500', '20505': '300', '20545': '100', '20580': '100', '20600': '400', '20620': '300', '20650': '200', '20680': '100', '20695': '100', '20700': '300', '20720': '300', '20750': '500', '20780': '100', '20800': '100', '20820': '100', '20830': '100', '20840': '100', '20880': '100', '20900': '700', '20945': '100', '20950': '300', '20990': '100', '21000': '4500', '21100': '200', '21110': '100', '21195': '100', '21200': '500', '21270': '100', '21280': '200', '21295': '100', '21300': '1200', '21350': '100', '21380': '100', '21400': '400', '21445': '100', '21450': '100', '21495': '100', '21500': '1800', '21550': '300', '21600': '100', '21700': '200', '21755': '100', '21770': '100', '21780': '100', '21850': '200', '21880': '100', '21900': '200', '22000': '1900', '22020': '100', '22050': '300', '22100': '100', '22225': '100', '22300': '200', '22330': '100', '22500': '500', '22555': '100', '22900': '100', '23000': '1500', '23185': '100', '23190': '100', '23195': '100', '23275': '100', '23285': '100', '23500': '100', '23600': '100'}}}

図 4.9: FLEX FULL historical data の例．注文，約定ごとに本図のような情報が存在している．

4.3.2 予測タスク

本実験では将来の仲値の上昇下降を予測することを目指す．深層学習モデルは回帰タスクに比べ識別タスクの方が得意な場合が多いため，以下のように将来の価格変動を 3 クラスに分類し予測を行う．

1. Up: $p_{\mathrm{th}} < p_{\mathrm{mid},t+\Delta t} - p_{\mathrm{mid},t}$
2. Stay: $-p_{\mathrm{th}} \leq p_{\mathrm{mid},t+\Delta t} - p_{\mathrm{mid},t} \leq p_{\mathrm{th}}$
3. Down: $p_{\mathrm{mid},t+\Delta t} - p_{\mathrm{mid},t} < -p_{\mathrm{th}}$

ここで，$p_{\mathrm{mid},t}$ が時刻 t における仲値 (midprice)，p_{th} がしきい値を表す．t および Δt としては，実時間スケールおよび注文スケールが考えられる．実時間スケールの場合は 5 秒後や 1 分後，注文スケールの場合は 20 注文後や 100 注文後といった Δt の値が考えられる．本実験では注文が出された時間帯や市場状態ではクラスタリングは行わないため，状況に応じ時間あたりの注文数や市場の変動速度は大きく異なると考えられる．したがって本実験では比較的市場変化速度が一定な注文スケールを採用した．また，Δt を大きく取りすぎるとこのデータセットのみからでは観測できない外的要因 (市場に影響を及ぼすイベント) に大きく左右されるため，$\Delta t = 20$ とし，非常に短い時間先の市場変動を予測対象とした．ここで，本実験で用いる銘柄については，20 注文後は平均して 15 秒程度先である．

4.3.3 Feature engineering

本実験では説明変数として，予測時点から直近 100 注文を以下の 3 次元に parameterize し，100×3 次元の系列入力を作成した．

1. $x_{\text{price}} = p - p_{\text{mid},t}$
2. $x_{\text{amount}} = \log(a)$
3. $x_{\text{midprice}} = p_{\text{mid},t} - p_{\text{mid},t-1}$

ここで，p が注文価格，$p_{\text{mid},t}$ が注文時 (時刻 t) における仲値，a が注文数量，$p_{\text{mid},t-1}$ が直前の仲値である．買い注文の場合は $x_{\text{price}} < 0$ に，売り注文の場合は $x_{\text{price}} > 0$ となる．取り消し注文および成行注文は考慮しない．注文価格および仲値は正規化した上で入力とすることにより，深層学習モデルの学習効率の向上を目指した．

4.3.4 BNN モデル

本実験で入力として用いる説明変数には時系列情報が含まれている．このような系列情報を含む入力からの特徴抽出には，convolutional neural network (CNN) [238, 239] あるいは recurrent neural network (RNN) [240, 241] が用いられる場合が多い．ここで重要なのは，注文系列は高頻度注文の注文系列は「ゆるやかな系列性を持っている」ことである．高頻度で大量の注文が投稿される現代の金融市場では，注文は必ずしも直近の注文を参照しているとは限らず，また隣接する注文は順序が入れ替わることも起こりうる．このような高頻度注文系列の解析に対し，Tsantekidis [110] らや田代 [180] らは CNN を用いた学習を行い成果を挙げている．したがって本研究でも CNN ベースの BNN モデルを用いて学習を行う．

解析を行う BNN モデルとしては，4.1.3 章で解説した variational dropout BNN を採用した．本実験で使用した BNN モデルの構造を図 4.10 に示す．比較対象とした通常の CNN モデルと dropout の回数を揃えるため，SVD Conv 層と Conv 層を併用している．また，特徴次元は位置情報を持たないため，一番はじめの Conv 層で特徴次元を 1 次元に折りたたみ，残りの Conv, SVDConv および MaxPool 層では注文次元方向に畳み込みをしている．最後の全結合層で 3 次元の logit (それぞれ Up, Stay, Down に対応している) が出力される．予測の度に SVDConv 層からサンプリングが行われるため，幅を持った予測が可能である．

学習済み BNN モデルを用いて予測する際は，各入力に対し出力値のサンプリングをそれぞれ 100 回行う．得られた 100 件の confidence (softmax(logits)) を平均し，予測 confidence および予測ラベルとした．このような予測値の算出方法は Bayesian prediction [242] および ensemble methods [243] を参考にしている．

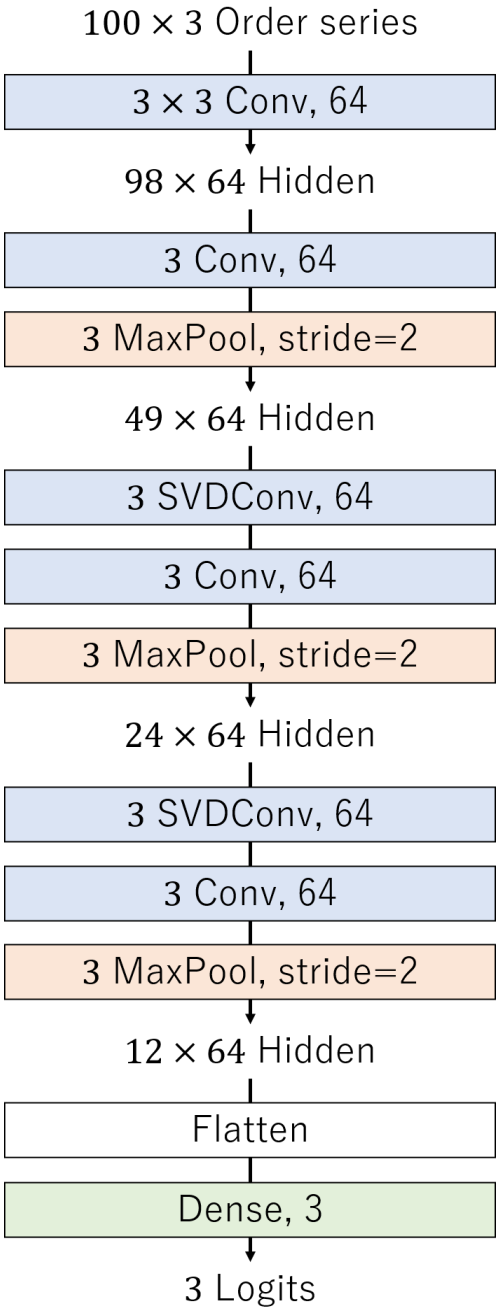


図 4.10: 株価予測タスクで用いた BNN ネットワーク構造．ネットワークを構成する層 (layer) を長方形の枠で表しており，Conv が convolutional 層を， SVDConv が sparse variational dropout convolutional 層を，MaxPool が最大値プーリング (maximum pooling) 層を，Flatten が入力の変形層を，Dense が fully connected 層を示している．モデルは 100×3 次元の注文系列入力を受け取り，Up, Stay, Down それぞれに対応した 3 次元の logit が得られる．予測の度に SVDConv 層からサンプリングが行われる．

4.3.5 比較手法

比較手法としては，variational dropout を施さない通常の CNN，および logistic regression (LR) [227, 228] を用いた．BNN とネットワーク構造を揃えるため，CNN では SVD Conv 層の代わりに Dropout + Conv 層を配置している． BNN 同様，CNN および LR の出力値を confidence として指標を計算した．

4.3.6 実験結果

FLEX FULL の主要な 5 銘柄 (3407: 旭化成，4188: 三菱ケミカルホールディングス，4568: 第一三共，5020: ENEOS ホールディングス，6502: 東芝) について解析を行った．モデル検証データに対する予測精度を表 4.2 および 表 4.3 に表す．表 4.2 では Accuracy (正確度) および F1 Score を，表 4.3 では Down (下降) および Up (上昇) クラスに対する Precision (正解率) を示している．本実験で用いたデータは Stay (変動なし) クラスの割合が大きいため，Accuracy および F1 Score に比べ Precision の値が小さくなってしまう．各種法を比較すると，提案手法である BNN が CNN や LR に比べ高い精度を残していることがわかる．

次に予測信頼性の妥当性を検証するため，confidence の値にしきい値を設定し，しきい値以上の confidence を持つサンプルのみで Precision を計算した．結果を図 4.11，4.12，4.13，4.14，4.15 に表す．実験で使用した全 5 名柄について，横軸にしきい値を，縦軸にしきい値以上の confidence のサンプルに対する precision の値をプロットしており，プロットが高い位置を通っているほど妥当な予測信頼性評価ができていることを表す．図を見ると，全銘柄について提案手法である BNN が良好な結果を残しており，Bayesian の導入により confidence 予測の妥当性が向上していることがわかる．また，図 4.13 や 図 4.15 のようにしきい値に対して順調に precision が向上している銘柄もある一方，図 4.12 のようにしきい値に対し precision が急落するような例も観測される．これはしきい値を高めたことによりしきい値以上の confidence が予測されたサンプルが非常に少なくなり，precision の値が不安定になってしまったためであると考えられる．説明変数やネットーワーク構造の改良により学習効率を改善することで，より妥当な confidence の予測が可能になることが期待される．加えて，BNN の予測は複数回のサンプリングを必要とするため，通常の CNN に比べサンプリング回数倍の計算時間が必要なことも課題の一つである．

表 4.2: FLEX FULL データに対する各手法の Accuracy および F1 Score．TYO ID は東京証券取引所の銘柄 ID を表し，3407: 旭化成，4188: 三菱ケミカルホールディングス，4568: 第一三共，5020: ENEOS ホールディングス，6502: 東芝 である．

	BNN		CNN		Logistic Regression	
TYO ID	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score
3407	0.875	0.855	0.873	0.853	0.699	0.754
4188	0.845	0.843	0.835	0.837	0.681	0.752
4568	0.828	0.792	0.827	0.791	0.445	0.549
5020	0.908	0.901	0.907	0.900	0.810	0.847
6502	0.905	0.887	0.908	0.889	0.785	0.823

表 4.3: FLEX FULL データに対する各手法の Precision．TYO ID は東京証券取引所の銘柄 ID を表し，3407: 旭化成，4188: 三菱ケミカルホールディングス，4568: 第一三共，5020: ENEOS ホールディングス，6502: 東芝 である．

	BNN		CNN		Logistic Regression	
TYO ID	Down	Up	Down	Up	Down	Up
3407	0.238	0.319	0.233	0.273	0.107	0.115
4188	0.195	0.248	0.173	0.243	0.135	0.143
4568	0.383	0.319	0.370	0.309	0.105	0.110
5020	0.279	0.245	0.303	0.220	0.130	0.133
6502	0.343	0.371	0.385	0.378	0.162	0.162

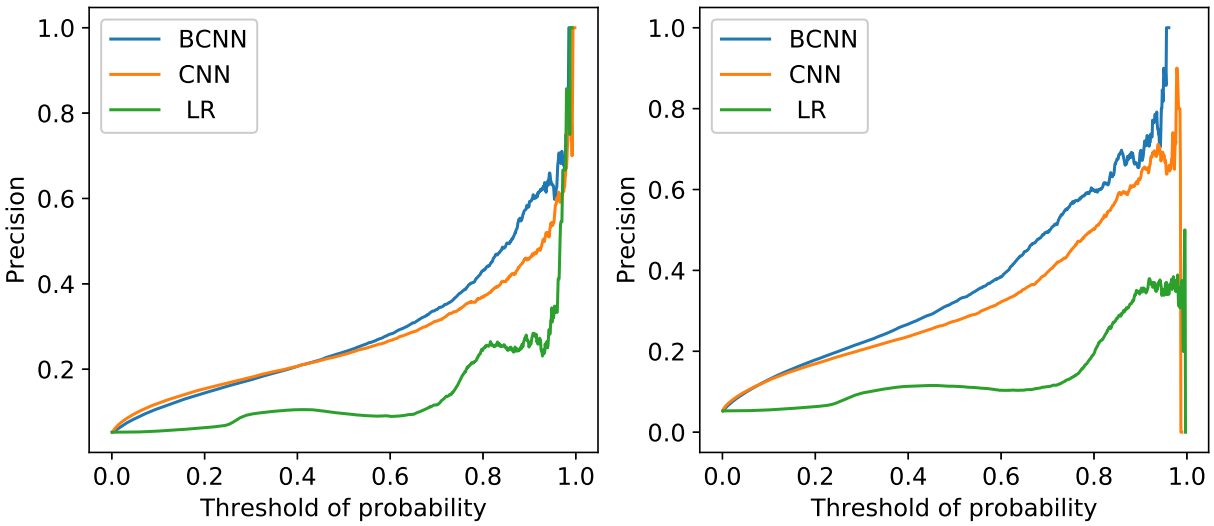


図 4.11: TYO ID 3407(旭化成) に対する予測信頼性の比較結果．横軸が confidence のしきい値を，縦軸がしきい値以上の confidence のサンプルに対する Precision(正解率の値を表す．図の青色，オレンジ色，緑色の曲線がそれぞれ BNN，CNN，logistic regression についての結果を表す．)

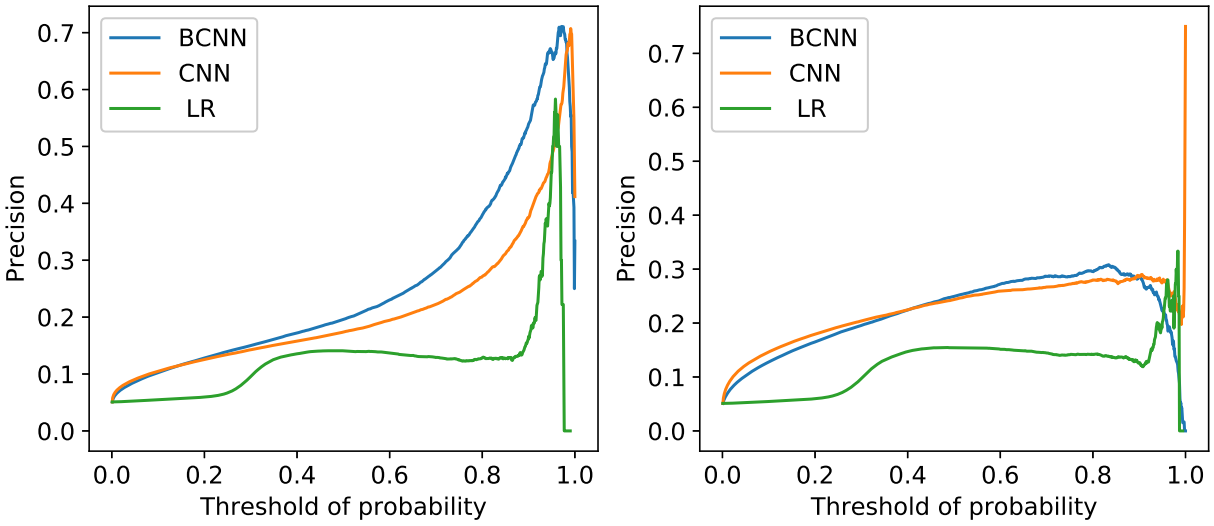


図 4.12: TYO ID 4188(三菱ケミカルホールディングス) に対する予測信頼性の比較結果．横軸が confidence のしきい値を，縦軸がしきい値以上の confidence のサンプルに対する Precision(正解率の値を表す．図の青色，オレンジ色，緑色の曲線がそれぞれ BNN，CNN，logistic regression についての結果を表す．)

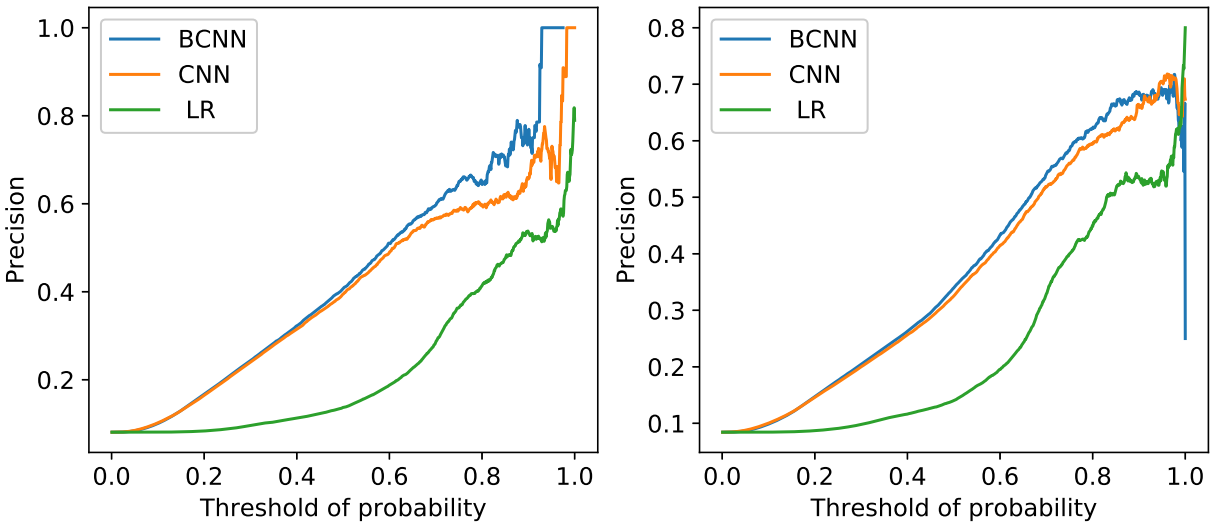


図 4.13: TYO ID 4568(第一三共) に対する予測信頼性の比較結果．横軸が confidence のしきい値を，縦軸がしきい値以上の confidence のサンプルに対する Precision(正解率の値を表す．図の青色，オレンジ色，緑色の曲線がそれぞれ BNN，CNN，logistic regression についての結果を表す．)

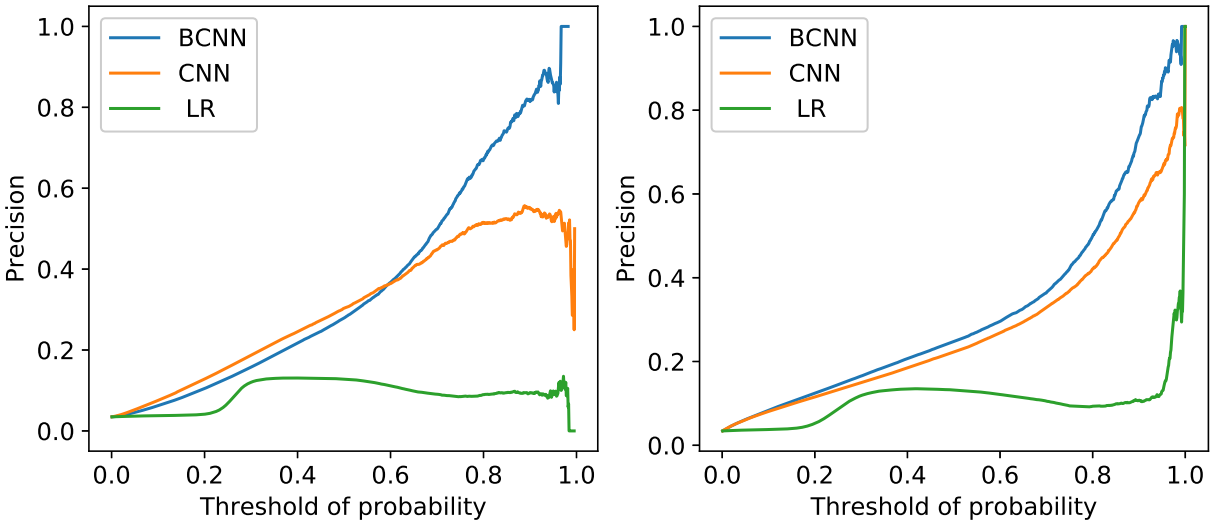


図 4.14: TYO ID 5020(ENEOS ホールディングス) に対する予測信頼性の比較結果．横軸が confidence のしきい値を，縦軸がしきい値以上の confidence のサンプルに対する Precision(正解率の値を表す．図の青色，オレンジ色，緑色の曲線がそれぞれ BNN，CNN，logistic regression についての結果を表す．)

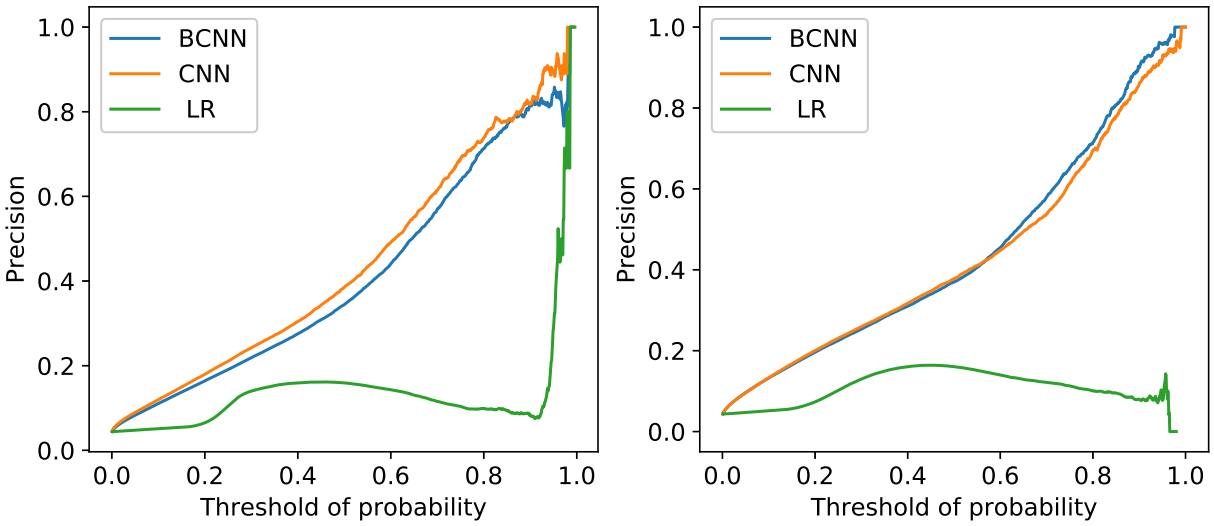


図 4.15: TYO ID 6502(東芝) に対する予測信頼性の比較結果．横軸が confidence のしきい値を，縦軸がしきい値以上の confidence のサンプルに対する Precision(正解率の値を表す．図の青色，オレンジ色，緑色の曲線がそれぞれ BNN，CNN，logistic regression についての結果を表す．)

4.4　実験:投資シミュレーションを用いた不確かさ評価の妥当性検証

4.3 章では価格動向予測タスクに対し BNN を適用し，予測信頼性の予測が妥当に行えることを確認した．本章ではさらに過去取引データを用いた投資シミュレーション (back-testing [244, 245, 246]) を行い，深層学習モデルの不確かさ評価により安全かつ効率的な投資判断が行えるかどうか確認した．実験に際し，不確かさ評価を活用した投資アルゴリズムの提案も行った．

本研究の一部は，2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI) および International Journal of Smart Computing and Artificial Intelligence で発表している [247, 248].

4.4.1　BNN を用いた投資判断

投資シミュレーションを用いた手法比較を行うためには，モデルの予測結果を利用した投資アルゴリズムを設計する必要がある．本研究ではモデルの予測値，および BNN の予測により得られる予測標準偏差を用いた投資意思判断過程を設計した．以下にその詳細を示す．なお，価格予測モデルの出力としては 4.3 章で利用した {Up, Stay, Down} の 3 クラスを考える．

Score-Based Process

はじめにベースラインとして，決定的な予測手法でも利用可能な score (あるいは confidence) ベースの投資判断過程を示す．本手法を score-based process と呼ぶ．Score-based process では，投資判断を以下のように下す．

1. Buy ($y_{\text{up}} > \max(t, y_{\text{stay}}, y_{\text{down}})$)
2. Sell ($y_{\text{down}} > \max(t, y_{\text{up}}, y_{\text{stay}})$)
3. Stay (otherwise)

ここで，$y_{\text{up}}, y_{\text{stay}}, y_{\text{down}}$ はそれぞれ価格上昇，変化なし，下降に対応する予測 score ($y_{\text{up}} + y_{\text{stay}} + y_{\text{down}} = 1$)，$t$ は score のしきい値である．y_{up} あるいは y_{down} が他の

クラスに対応する score よりも大きく，かつしきい値よりも大きい場合，買いあるいは売りの判断が下される．Buy あるいは sell の行動が選択された場合，市場状態に応じ取引が行われる．ここで，株式市場の取引形態は多種存在し，主要なものでも 指値注文 (limit order, LMT) による取引や成行注文 (market order, MKT) による取引が挙げられる．本来は効率的な注文の約定を目指す場合，市場状態を元に臨機応変な注文行動を行う必要がある [249, 250]．しかし本研究においては，注文行動を行うアルゴリズムの優劣により評価指標が上下してしまうと手法間の比較が適切に行えなくなってしまう．加えて成行注文による取引のような取引コスト [251] が存在する取引方法の場合，取引を行う回数に比例して資産の期待値が小さくなってしまう．したがって本実験では，仲値 p_{mid} を約定価格として取引を行うことで，取引コストを無視できるようにするとともに，投資アルゴリズムによる結果への影響が存在しないようにした．仲値 p_{mid} を用いて数量 v の取引を行う場合，投資者の現金の変化量 Δc は以下のように表される．

$$\Delta c = -p_{\mathrm{mid}} v \tag{4.24}$$

ここで，取引数量 v は買い注文では正の値，売り注文では負の値となる．取引数量 v は以下のように決定する

$$v = \begin{cases} \frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}} & (\mathrm{Buy}) \\ -\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}} & (\mathrm{Sell}) \end{cases} \tag{4.25}$$

ここで，I は投資者の現在の株式保有量 (inventory) を，r は投資割合 (investment ratio) を表す．$c + p_{\mathrm{mid}}I$ は現在の資産を表し，投資者はその最大化を目指す．また，Inventory I の値には上限 I_{lim} および 下限 $-I_{\mathrm{lim}}$ を定める．I_{lim} は以下の式で計算される．

$$I_{\mathrm{lim}} = \frac{(c + p_{\mathrm{mid}}I)l}{p_{\mathrm{mid}}} \tag{4.26}$$

ここで，l はレバレッジ (leverage) [252, 253] である．以上をもとに，投資数量 v は以下のように計算される．

$$v = \begin{cases} \min\left(\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}}, \max(I_{\mathrm{lim}} - I, 0)\right) & \text{(Buy)} \\ -\min\left(\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}}, \max(I_{\mathrm{lim}} + I, 0)\right) & \text{(Sell)} \end{cases} \tag{4.27}$$

Score-based process のアルゴリズムを Algorithm 1 に示す．通常の決定的な深層学習モデルでは，ネットワークの予測値 y が score として用いられる．BNN においては，複数回の予測の平均値 $\overline{y_i}$ が score として用いられる．

Algorithm 1 Score-based process の予測および投資判断プロセス

Require: prediction NN (BNN) $f(\cdot)$
Parameter: score threshold t, investment ratio r, and leverage l
Get current cash c and inventory I
Get market states X and mid price p_{mid}
Predict scores $[y_{\mathrm{up}}, y_{\mathrm{stay}}, y_{\mathrm{down}}] = f(X)$
Get inventory limit $I_{\mathrm{lim}} = \frac{(c+p_{\mathrm{mid}}I)l}{p_{\mathrm{mid}}}$
if $y_{\mathrm{up}} > \max(t, y_{\mathrm{stay}}, y_{\mathrm{down}})$ **then**
  $v = \min(\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}}, \max(I_{\mathrm{lim}} - I, 0))$
else if $y_{\mathrm{down}} > \max(t, y_{\mathrm{up}}, y_{\mathrm{stay}})$ **then**
  $v = -\min(\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}}, \max(I_{\mathrm{lim}} + I, 0))$
else
  $v = 0$
end if
Update cash $c \leftarrow c - p_{\mathrm{mid}}v$
Update inventory $I \leftarrow I + v$

Std-Based Process

予測の不確かさを考慮することで投資意思判断過程を改善することができる．本研究では，BNN による複数回予測の標準偏差 σ_y を用いることで，以下のように score の値を補正する

$$y' = y - k\sigma_y \tag{4.28}$$

ここで k は係数である．式 4.28 の補正は lower confidence bound (LCB) [254, 255] を元にしている．Lower confidence bound または upper confidence bound (UCB) はベイズ最適化 (Bayesian optimization) [196, 197] で獲得関数 (acquisition function) として用いられている関数の 1 つである．ベイズ最適化は Gaussian process (GP) [256, 257] によって出力された正規分布に従う予測分布を用いて実験計画 [258, 259] を行う手法であり，探索対象を決定する獲得関数の選択は探索効率に大きく影響するが，LCB および UCB のは探索効率が高い手法として知られている．式 4.28 の係数 k は活用 (utilization) と探索 (search) のトレードオフを表す値であり，本実験においては，k を大きくすればするほど安定志向な行動選択を行うことになる．補正スコア y' を用いた std-based process の過程をアルゴリズム 2 に示す．

Algorithm 2 Std-based process の予測および投資判断プロセス

Require: prediction BNN $f(\cdot)$
Parameter: coefficient k, score threshold t, investment ratio r and leverage l
Get current cash c and inventory I
Get market states X and mid price p_{mid}
for $i = 1, \cdots, N$ **do**
  Sample scores $[y_{i,\mathrm{up}}, y_{i,\mathrm{stay}}, y_{i,\mathrm{down}}] \sim f(X)$
end for
Get modified sores $y'_{\mathrm{up}} = E[y_{i,\mathrm{up}}] - k\sqrt{V[y_{i,\mathrm{up}}]}, y'_{\mathrm{stay}} = E[y_{i,\mathrm{stay}}] - k\sqrt{V[y_{i,\mathrm{stay}}]}$, $y'_{\mathrm{down}} = E[y_{i,\mathrm{down}}] - k\sqrt{V[y_{i,\mathrm{down}}]}$
Get inventory limit $I_{\mathrm{lim}} = \frac{(c+p_{\mathrm{mid}}I)l}{p_{\mathrm{mid}}}$
if $y'_{\mathrm{up}} > \max(t, y'_{\mathrm{stay}}, y'_{\mathrm{down}})$ **then**
  $v = \min(\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}}, \max(I_{\mathrm{lim}} - I, 0))$
else if $y'_{\mathrm{down}} > \max(t, y'_{\mathrm{up}}, y'_{\mathrm{stay}})$ **then**
  $v = -\min(\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}}, \max(I_{\mathrm{lim}} + I, 0))$
else
  $v = 0$
end if
Update cash $c \leftarrow c - p_{\mathrm{mid}}v$
Update inventory $I \leftarrow I + v$

4.4.2　データセット

本実験では 4.3 章 と同様に FLEX FULL historical dataset をデータセットとして用いた．前述の通り FLEX FULL は東京証券取引所の板・注文および約定データであり，高頻度取引により構成された 1 日あたり数万件から数十万件程度のデータで成り立っている．FLEX FULL には数千の銘柄が存在しているが，今回は銘柄コード 9022 (東海旅客鉄道) を選択し，2019 年の 1 月 1 日から 6 月 30 日を学習データに，7 月 1 日から 8 月 31 日までを検証データとした．

4.4.3　Feature engineering

本実験では，深層学習モデルの入力として，以下の 2 つの特徴量を使用した．

1. Price series
2. Orderbook features

Price series は 10 件の market price (市場価格) の系列であり，注文時点から注文スケールで 10 時点ずつ遡って価格を収集している．すなわち，時刻 t における market price を p_t，注文時点を $t = 0$ とすれば price series は $\{p_{-90}, p_{-80}, p_{-70}, \cdots, p_0\}$ のように表せる．

Orderbook features は予測時点の板情報を一次元のベクトルに直した特徴量である．予測時点に板に登録されている注文の数量のうち，仲値中心に上下 10 価格のもののみを取り出し，価格順に並べることで作成される．ここで，買い注文および売り注文をモデルが区別しやすくするため，買い注文の数量は -1 倍している．Orderbook features の例を図 4.16 に示す．

Sell Volume	Price	Buy Volume
…	…	…
3	505	0
2	504	0
4	503	0
8	502	0
10	501	0
0	500	9
0	499	6
0	498	3
0	497	0
0	496	1
…	…	…

⇨

Orderbook feature
…
3
2
4
8
10
-9
-6
-3
0
-1
…

図 4.16: Orderbook features の作成例．予測時点に板に登録されている注文の数量のうち，仲値中心に上下 10 価格のもののみを取り出し，買い注文の数量は -1 倍した上で価格順に並べて特徴量を作成している．

4.4.4　予測モデル

本実験で用いた深層学習モデルのネットワーク構造を図 4.17 に示す．本実験で用いる Price series および Orderbook features から特徴量が抽出され，結合された上で全結合層 (Dense) により価格の変動が 3 クラス (Up, Stay, Down) で予測される．2.2.5 章と同様の理由で, Price series からの特徴抽出には LSTM を，Orderbook features からの特徴抽出には　CNN 用いた．以上のようなモデルをベースライン (Baseline NN) として用いる．

深層学習モデルの不確かさ考慮手法としては，4.3 章同様 sparse variational dropout BNN を用いる．現在 sparse variational dropout は Dense および Conv 層に適用可能なため，本実験では Baseline NN の Dense および Conv 層を SVDDense および SVDConv 層に置き換えて BNN モデルを作成した．

また，ベースラインおよび BNN に加え，正則化として L2 regularization [260, 261] を加えた場合についても学習を行い，結果を比較した．

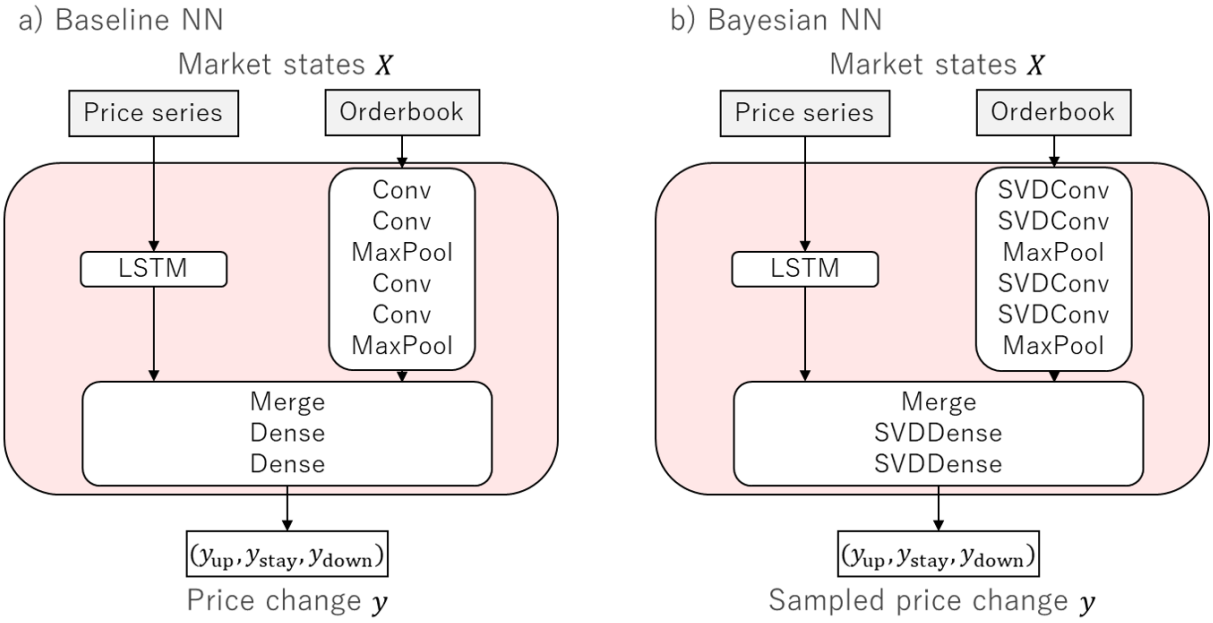


図 4.17: 投資シミュレーション実験で用いたネットワーク構造．Price series および Orderbook features から特徴量が抽出され，結合される．Price series の特徴抽出には LSTM が，Orderbook features の特徴抽出には畳み込み層が，結合された特徴量からの出力値の算出には全結合層が用いられ，価格の変動が 3 クラスで予測される．BNN においては，Conv の代わりに SVD Conv 層が，Dense 層の代わりに SVD Dense 層が使用され，幅のある出力が得られる．

4.4.5　実験結果

実験結果を以下に示す．はじめに本実験で用いた 4 つのモデルについて，モデル検証データに対する予測精度を比較した．精度指標としては，近年の識別タスク比較で広く用いられている area under a receiver operating characteristic (AUROC) [262]，false-postive rate 95% (FPR95) [263]，area under precision-recall curve (AUPR) [264, 265] を用いた．加えて，予測スコアの妥当性検証のため 4.2.5 章でも解説した expected calibration error (ECE) を用いた．結果を表 4.4 に示す．

表 4.4 を見ると，予測精度については SVDBNN + L2Regu が若干高い精度を残しているものの，ベースラインおよび BNN モデルが拮抗していることがわかる．対して ECE を比べると，SVDBNN がベースラインを大きく上回る値を残している．通常の識別タスクとして学習したベースラインモデルに比べ，出力値事後分布の近似を目的に学習する BNN では予測信頼性が劇的に改善したと考察される．

次に投資シミュレーションの結果を示す．投資シミュレーションはモデル検証データを用いて行い，前述の Algorithm 1 および Algorithm 2 に従って行動選択を行った．Score-based process においてはしきい値 t の値を $\{\frac{1}{3}, 0.4, 0.5, 0.6, 0.7, 0.8\}$ の中で変化させシミュレーションを行った．Std-based process においては係数 k を $\{0.5, 1, 2\}$ の中で，しきい値 t の値を $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ の中で変化させシミュレーションを行った．結果を表 4.5 および表 4.6 に示す．

投資実績は total return rate T_r, Sharpe ratio S_p, maximum drawdown MDD を用いて評価された．Total return rate は以下の式でに計算される．

$$T_r = \frac{cap_{\text{end}} - cap_{\text{start}}}{cap_{\text{start}}} \tag{4.29}$$

ここで，cap_{start} および cap_{end} はシミュレーションの開始時および終了時の資産 $cap = c + p_{\text{mid}} I$ を表す．total return rate は単純な資産の増加率として解釈できる．日次で計算された T_r は 250 倍され年次に換算される．また，Sharpe ratio S_p [133, 266, 267] は以下のように計算される．

$$S_p = \frac{E[R_a - R_b]}{\sqrt{V[R_a - R_b]}} \tag{4.30}$$

ここで，R_a および R_b は投資およびベンチマークのリターンを，E および V は期待値および分散を表す．本実験ではベンチマークのリターン R_b は 0 とする．Sharpe ratio はどれだけ安定的にリターンを出せるかを評価する指標と解釈できる．最後に，maximum drawdown MDD [116, 117] は以下のように計算される．

$$MDD = \frac{P - L}{P} \tag{4.31}$$

ここで，P および L はシミュレーション期間で最も大きい損失を出した前後における資産の値である．MDD は損失の大きさを評価する指標として使用され，小さいほど安全な投資が行えていることがわかる．

表 4.5 および表 4.6 を見ると，全体的にしきい値 t が小さければ良好な total return rate および Sharpe ratio の値が得られていることがわかる．一般的に，しきい値 t を大きくすればするほどリスクを避けた安定的な投資ができると考えられる．しかし投資コストを無視する本実験においては，t を大きくしすぎると投資機会が減り，逆に安定的に利益を挙げるのが困難になったと考えられる．

手法間の比較をすると，全体に SVDBNN が Baseline を上回る実績を出しているように見える一方，score based process と std based process の間では，最良な結果間を比較すると大きな差は存在しない様に見える．そこでモデル間，および投資アルゴリズム間の差が有意であるかを検証するため，日次の評価指標を用いて t 検定を行った．Baseline モデル ($t = \frac{1}{3}$) および SVDBNN モデル (score based process，$t = \frac{1}{3}$) の比較結果を表 4.7 および図 4.18 に，SVDBNN モデル (score based process，$t = \frac{1}{3}$) および SVDBNN モデル (std based process，$k = 0.5, t = 0$) の比較結果を表 4.8 および図 4.19 に示す．表 4.7 を見ると，Baseline モデルおよび SVDBNN モデルの間では MDD 比較の p 値が十分に小さく，図 4.18 からも SVDBNN モデルが有意に優れていることがわかる．一方，表 4.8 を見ると，L2 regularization を施さない場合の MDD，および L2 regularization を施した場合の Sp は若干 p 値が低くなっているものの，両者の差が有意だと言えるほど低い p 値は観測されていない．Score based process および std based process は学習済みモデルはネットワークのパラメータ (重み) が同一であり，予測結果も同一のため大きな差が生まれなかったと考えられる．提案手法である std based process の利点は予測の不確かさ考慮であり，今後取引コストを考慮した条件下でシミュレーションを行うことにより優れた実績を残すことができると期待される．

表 4.4: 投資シミュレーション実験で用いたモデルの予測精度．予測精度は FLEX FULL の検証データを用いて計算され，area under a receiver operating characteristic (AUROC)，false-postive rate 95% (FPR95)，area under precision-recall curve (AUPR)，expected calibration error (ECE) で評価した．表の L2Regu は L2 regularization を表す．

Model	AUROC ↑	FPR95 ↓	AUPR ↑	ECE ↓
Baseline	**0.5627**	0.9322	0.3767	0.1087
Baseline + L2Regu	0.5599	0.9336	0.3727	0.1145
SVDBNN	0.5504	0.9167	0.3725	**0.0068**
SVDBNN + L2Regu	0.5603	**0.9081**	**0.3819**	0.0089

投資シミュレーションの実行例を図 4.20 に示す．図 4.20 では 2019 年 8 月 8 日のデータについて，各手法で投資シミュレーションを行い，資産 (Capital)，株式保有料 (Inventory)，および仲値 (Mid price) の変化をプロットした．ベースラインモデルにおいては $t = \frac{1}{3}$ の場合を，BNN モデルにおいては $k = 0.5$，$t = 0$ の場合を例として選択した．図 4.20 を見ると全体的に capital の値が上下動しているが，SVDBNN はベースラインモデルに比べ上下動が少なく，安定的に資産を増加させていくことができている．また，良好な結果を残した SVDBNN においても全体的に Inventory の絶対値が大きくなっており，許容される範囲内で大きなリスクをとって投資を行っていることがわかる．このような行動選択については今後改善の余地があると考えている．

表 4.5: Score-based process の投資シミュレーション実績．投資シミュレーションは学習済みのベースラインおよびBNN モデルを用いて行い，モデル検証データに沿って進められ，Algorihtm 1 に従って行動選択がなされた．投資実績は total return rate T_r, Sharpe ratio S_p, maximum drawdown MDD を用いて評価された．書く指標は日毎に計算され，その平均および標準偏差が表に示されている．表記のない箇所については，しきい値が大きすぎて投資行動が全く行われなかった．

Model	t	$T_r \uparrow$	$S_p \uparrow$	$MDD \downarrow$
Baseline	1/3	0.9379 ± 1.3388	0.0138 ± 0.0197	0.0016 ± 0.0004
(Score based)	0.4	0.8338 ± 1.3769	0.0121 ± 0.0194	0.0016 ± 0.0004
	0.5	0.5159 ± 1.1397	0.0076 ± 0.0170	0.0017 ± 0.0004
	0.6	0.1867 ± 1.3677	0.0020 ± 0.0199	0.0018 ± 0.0004
	0.7	0.0655 ± 1.3935	0.0006 ± 0.0205	0.0015 ± 0.0006
	0.8	-0.0573 ± 1.1387	-0.0033 ± 0.0212	0.0012 ± 0.0006
Baseline	1/3	1.2575 ± 1.6847	0.0174 ± 0.0206	0.0014 ± 0.0002
+ L2Regu	0.4	1.1952 ± 1.7257	0.0162 ± 0.0210	0.0014 ± 0.0003
(Score based)	0.5	0.8636 ± 1.7033	0.0102 ± 0.0215	0.0016 ± 0.0003
	0.6	0.4135 ± 1.4812	0.0031 ± 0.0199	0.0017 ± 0.0005
	0.7	0.3654 ± 1.5046	0.0027 ± 0.0232	0.0015 ± 0.0006
	0.8	0.2087 ± 1.1344	0.0048 ± 0.0236	0.0011 ± 0.0006
SVDBNN	1/3	1.1951 ± 1.5409	0.0169 ± 0.0207	0.0013 ± 0.0003
(Score based)	0.4	0.6977 ± 1.6765	0.0080 ± 0.0228	0.0015 ± 0.0003
	0.5	-0.0024 ± 0.8229	0.0017 ± 0.0230	0.0009 ± 0.0010
	0.6	-0.1827 ± 0.7854	-0.0013 ± 0.0212	0.0007 ± 0.0014
	0.7	-0.0259 ± 0.1088	-0.0117 ± 0.0138	0.0001 ± 0.0003
	0.8	$-$	$-$	$-$
SVDBNN	1/3	1.3602 ± 1.3615	0.0208 ± 0.0197	0.0013 ± 0.0003
+ L2Regu	0.4	0.6145 ± 1.2419	0.0085 ± 0.0202	0.0015 ± 0.0003
(Score based)	0.5	-0.0487 ± 1.2490	-0.0006 ± 0.0215	0.0014 ± 0.0006
	0.6	-0.1248 ± 0.7760	-0.0018 ± 0.0261	0.0006 ± 0.0005
	0.7	0.0018 ± 0.0117	0.0100 ± 0.0000	0.0000 ± 0.0000
	0.8	$-$	$-$	$-$

表 4.6: Std-based process の投資シミュレーション実績.

Model	k	t	$T_r \uparrow$	$S_p \uparrow$	$MDD \downarrow$
SVDBNN	0.5	0	1.2450 ± 1.4619	0.0177 ± 0.0191	0.0013 ± 0.0003
(Std based)		0.1	1.3119 ± 1.4066	0.0192 ± 0.0190	0.0013 ± 0.0002
		0.2	1.2634 ± 1.4551	0.0180 ± 0.0194	0.0013 ± 0.0002
		0.3	1.2748 ± 1.6564	0.0175 ± 0.0208	0.0013 ± 0.0002
		0.4	0.5462 ± 1.7732	0.0052 ± 0.0222	0.0018 ± 0.0005
		0.5	-0.2973 ± 0.9135	-0.0053 ± 0.0194	0.0009 ± 0.0015
SVDBNN	1	0	1.2243 ± 1.5536	0.0169 ± 0.0199	0.0013 ± 0.0002
(Std based)		0.1	1.2976 ± 1.5561	0.0182 ± 0.0206	0.0013 ± 0.0002
		0.2	1.2374 ± 1.6341	0.0170 ± 0.0208	0.0013 ± 0.0002
		0.3	1.2111 ± 1.4363	0.0170 ± 0.0192	0.0014 ± 0.0003
		0.4	-0.4022 ± 1.1832	-0.0058 ± 0.0190	0.0013 ± 0.0015
		0.5	-0.2971 ± 0.8451	-0.0061 ± 0.0196	0.0008 ± 0.0015
SVDBNN	2	0	1.0663 ± 1.6193	0.0138 ± 0.0208	0.0015 ± 0.0006
(Std based)		0.1	1.2235 ± 1.6485	0.0163 ± 0.0212	0.0015 ± 0.0005
		0.2	1.0241 ± 1.5412	0.0136 ± 0.0206	0.0014 ± 0.0003
		0.3	0.7311 ± 1.5499	0.0091 ± 0.0212	0.0015 ± 0.0004
		0.4	-0.2767 ± 0.9115	-0.0052 ± 0.0190	0.0009 ± 0.0015
		0.5	-0.0097 ± 0.0796	-0.0020 ± 0.0200	0.0000 ± 0.0001
SVDBNN	0.5	0	1.4327 ± 1.2971	0.0224 ± 0.0196	0.0012 ± 0.0003
+ L2Regu		0.1	1.4289 ± 1.4009	0.0213 ± 0.0191	0.0013 ± 0.0003
(Std based)		0.2	1.3262 ± 1.4027	0.0201 ± 0.0198	0.0013 ± 0.0003
		0.3	1.4454 ± 1.3298	0.0224 ± 0.0195	0.0013 ± 0.0003
		0.4	-0.1249 ± 1.2795	-0.0020 ± 0.0205	0.0017 ± 0.0013
		0.5	-0.1191 ± 0.7834	-0.0042 ± 0.0221	0.0006 ± 0.0005
SVDBNN	1	0	1.3759 ± 1.3126	0.0212 ± 0.0175	0.0014 ± 0.0004
+ L2Regu		0.1	1.4013 ± 1.2687	0.0214 ± 0.0174	0.0013 ± 0.0003
(Std based)		0.2	1.4828 ± 1.5593	0.0223 ± 0.0209	0.0013 ± 0.0003
		0.3	1.4453 ± 1.3713	0.0216 ± 0.0182	0.0013 ± 0.0003
		0.4	-0.0479 ± 0.8866	-0.0092 ± 0.0224	0.0006 ± 0.0005
		0.5	–	–	–
SVDBNN	2	0	1.2509 ± 1.2408	0.0191 ± 0.0177	0.0013 ± 0.0003
+ L2Regu		0.1	1.1639 ± 1.3837	0.0173 ± 0.0193	0.0013 ± 0.0003
(Std based)		0.2	1.3471 ± 1.4619	0.0206 ± 0.0204	0.0013 ± 0.0003
		0.3	1.0053 ± 1.3677	0.0155 ± 0.0212	0.0015 ± 0.0004
		0.4	–	–	–
		0.5	–	–	–

表 4.7: Baseline($t = \frac{1}{3}$) モデルおよび SVDBNN モデル (Score based process，$t = \frac{1}{3}$) 二者間の，投資シミュレーション結果に対する t 検定結果．両モデルの日次の Sharpe ratio(Sp), maximum drawdown(MDD) について t 検定を行い，得られた p 値を比較している．表の Without L2 および With L2 は L2 regularization の有無を表している．

P value	Sp	MDD
Without L2	0.438	5.31×10^{-4}
With L2	0.353	8.85×10^{-4}

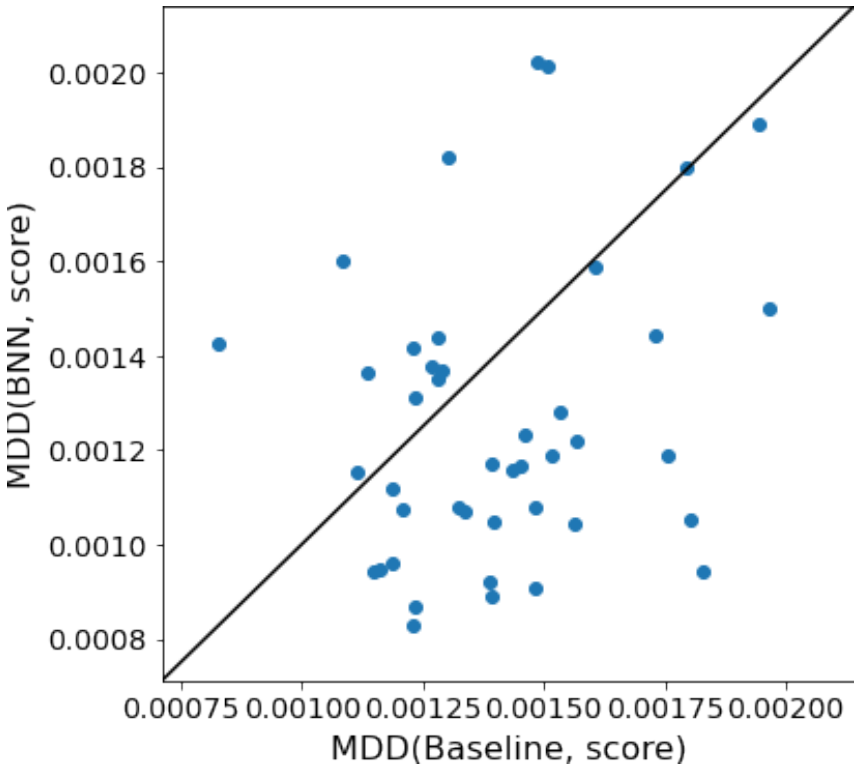


図 4.18: Baseline モデルおよび SVDBNN モデル (Score based process) の，投資シミュレーションにおける日次 Maximum drawdown の比較プロット．L2 regularization を施して学習したモデル間の結果を示している．

表 4.8: SVDBNN モデル (Score based process，$t = \frac{1}{3}$) および SVDBNN モデル (Std based process，$k = 0.5, t = 0$) 二者間の，投資シミュレーション結果に対する t 検定結果．両モデルの日次の Sharpe ratio(Sp), maximum drawdown(MDD) について t 検定を行い，得られた p 値を比較している．表の Without L2 および With L2 は L2 regularization の有無を表している．

P value	Sp	MDD
Without L2	0.561	0.134
With L2	0.146	0.605

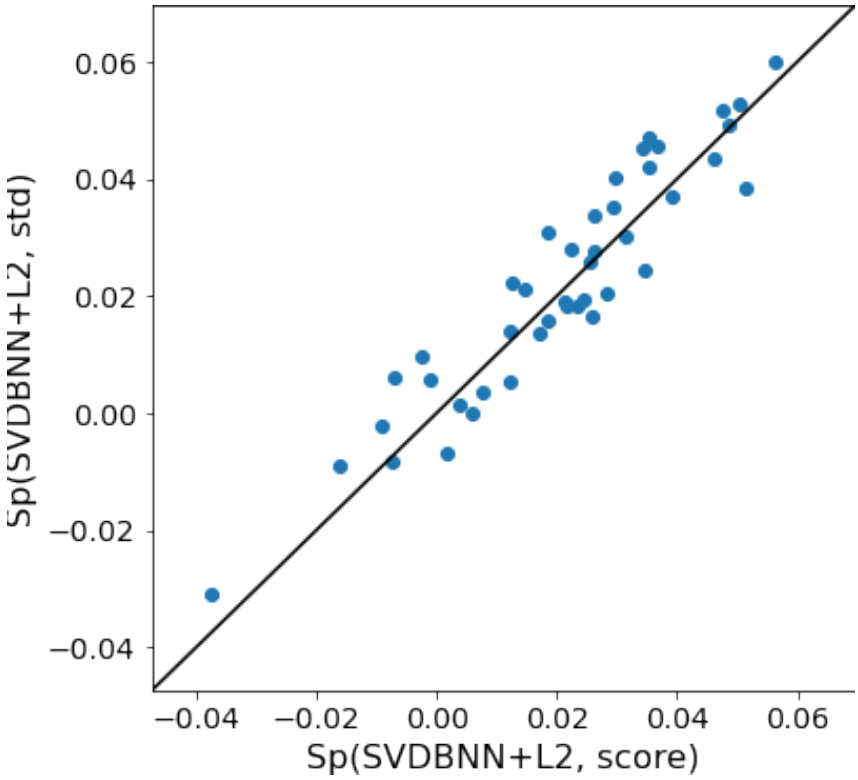


図 4.19: SVDBNN モデル (Score based process) および SVDBNN モデル (Std based process) の，投資シミュレーションにおける日次 Sharpe ratio の比較プロット．L2 regularization を施して学習したモデル間の結果を示している．

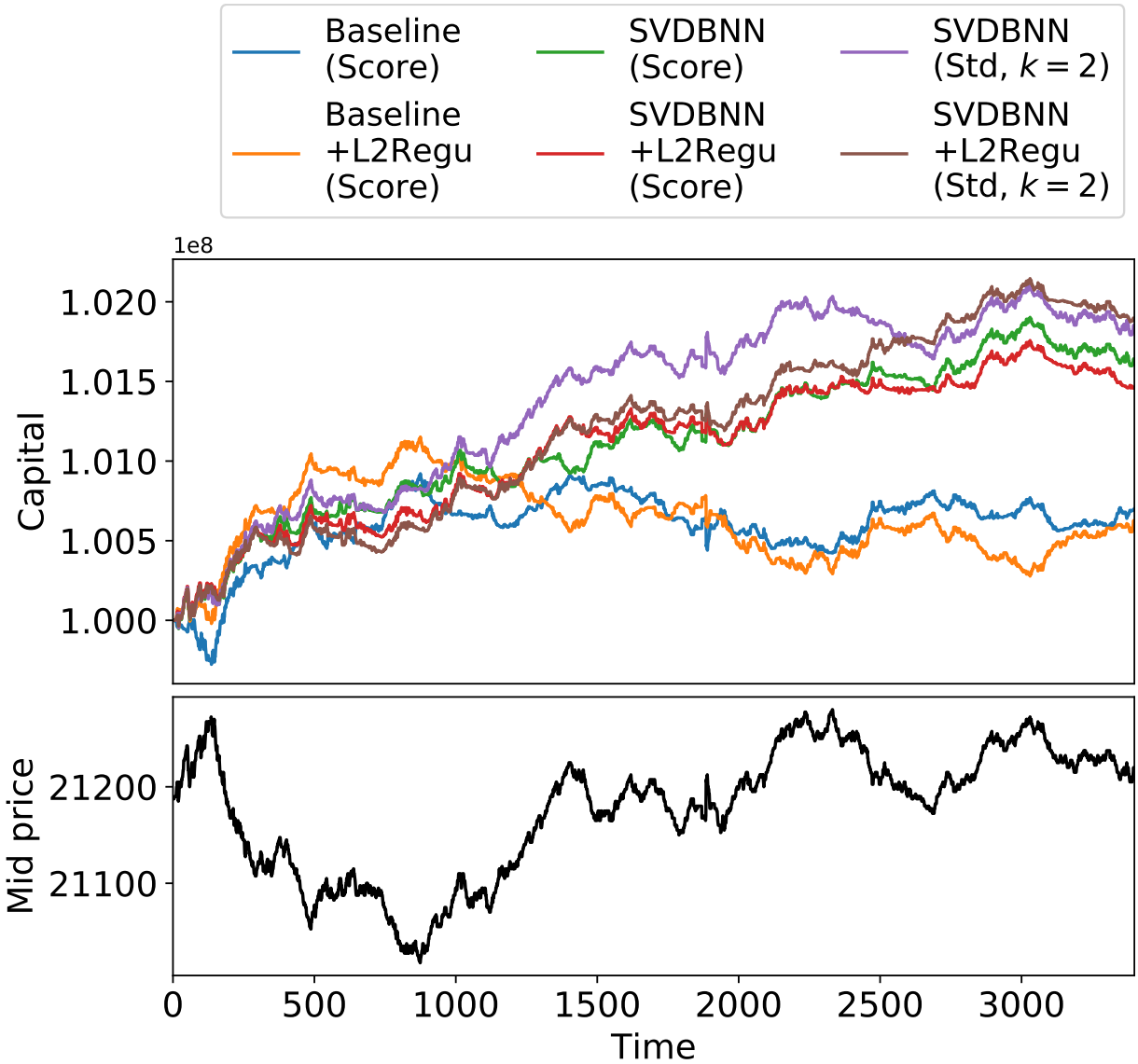


図 4.20: 投資シミュレーションの実行例．2019 年 8 月 8 日のデータについて，各手法で投資シミュレーションを行い，資産 (Capital)，および仲値 (Mid price) の変化をプロットした．ベースラインモデルにおいては $t = \frac{1}{3}$ の場合を，BNN モデルにおいては $k = 0.5$, $t = 0$ の場合を例として選択した．図の横軸は全行動数を表す．

4.5　結論と課題

本章では，ベイズ深層学習 (Bayesian neural networks, BNNs) を活用し，金融市場データに限らず予測タスク全般について，不確かさ考慮の妥当性を検討した．1.1 章でも述べたとおり金融市場予測にはデータ拡張では対応しきれない予測の不確実性が存在し，深層学習モデル自体の予測不確実性考慮を向上させなければ，安全な金融市場予測は実現できない．本章でははじめにベイズ深層学習含む複数の深層学習における不確かさ考慮手法を比較検討した．実験結果からはベイズ深層学習以外にも evidential deep learning のように不確かさ評価に有効な手法の存在が示唆されたが，不確かさを考慮した予測を行う際，ガウス過程との類似から投資意思決定プロセスを構築できるベイズ深層学習が最終的には選択された．金融市場予測の代表的なタスクである価格動向予測，および価格予測を活用した投資判断実験においてベイズ深層学習は既存手法を上回る実績を残し，金融市場予測タスクに対する不確かさ考慮の有効性を示した．

本章で行った実験は深層学習研究全般に適用できるものであり，逆に言えば本章の実験では十分に検証できていない事項も多い．例としては 4.2 賞で用いた以外の深層学習における不確かさ考慮手法の有効性検討，4.4 章で提案した以外の行動意思決定アルゴリズムにおける比較実験が挙げられる．中でも，本研究の目的である金融市場予測，投資意思決定に照らすと，以下に挙げる 2 点が大きな今後の課題として挙げられる

4.5.1　予測標準偏差以外の不確かさ考慮手法を用いた投資意思決定

本章ではベイズ深層学習を用いた予測により得られる予測標準偏差を活用し投資意思決定アルゴリズムを提案した．しかし，ベイズ深層学習を用いて得られる出力値の分布は正規分布である必然性はないため，標準偏差以外の特徴を用いたより発展的な意思決定アルゴリズムの開発が可能であると考えている [268, 269]．例として，サンプリングをもとに予測値の信頼区間を算出し，投資判断に用いる手法が考えられるが，必要なサンプリング数の多さが課題となる．加えて，本章では活性化関数として ReLU 関数を用いているが，より解析的に解きやすい活性化関数を用いることにより，出力値分布を明確な形状を持つ関数として取得し，少ないサンプリング数から予測区間等を求めることも考えられる．

また，ベイズ深層学習以外の不確かさ考慮手法を用いた投資意思決定アルゴリズムの開発も今後の方針として考えられる．4.2 章ではベイズ深層学習以外にも evidencial deep

learning が不確かさ評価として良好な成績を残しており，同手法の予測を用いた意思決定も有効であると予想されるが，ベイズ深層学習と異なり予測標準偏差が得られないため，Bayesian evidence の予測値を用いたしきい値等による意思決定しか行えない．同一の入力に対し幅のある予測を行うためには，ベイズ深層学習やアンサンブル学習の導入が有効であると考えている．

4.5.2 人工市場および深層強化学習との組み合わせ

本章の内容は現時点では 2 章や 3 章との関連が薄くなっている．今後の方針として，ベイズ深層学習技術を人工市場環境における深層強化学習に応用することを考えている．

2 章で提案した A3C および DQN モデルは現状予測の不確かさを考慮できておらず，学習データに存在しない市場状態等，予測が不確かになる状況に対しても確信を持った予測を行ってしまう．そこで深層強化学習モデルのベイズ化を行い，A3C であれば方策関数および状態価値関数，DQN であれば行動価値関数の平均および標準偏差を算出することで，本章で提案した std-based process を適用することができ，不確実性が大きい場面では投資行動を行わない判断が可能である．さらに進んだ方針としては，ベイズ深層学習を用いて得られる予測の不確かさを深層強化学習モデルが入力として受け取り，不確かさの大きい場面ではリスクの少ない行動を選択するように学習を行うことが考えられる．

3 章で提案した模倣学習モデルについても，ベイズ化を行うことにより幅のある確率分布の予測が可能となる．しかし 3 章の目的に照らした上で重要なのは，市場状態の動向や本物らしさをコントロールできるような手法であり，単純なベイズ的予測では目的は達成できないと考えている．そこで今後の方針として検討しているのは，ベイズ深層学習により得られる市場予測の不確かさをシミュレーションの指標として用いることである．学習効率を高めるために未知の市場状態を生成したいとき，生成したシミュレーションデータの中からベイズ深層学習モデルによる予測の不確かさが大きいものを選択する，あるいは予測の不確かさが大きくなるようにデータの生成を行うことにより，予測の安全性向上に有効なシミュレーションデータの生成が行えるのではないかと考えている．

第5章　結論

5.1　本研究の貢献

本研究の貢献を図 5.1 に示す.

人工市場と深層強化学習の活用による取引戦略学習は，これまで理論的，技術的な要因から行われてこなかった，人工市場環境における深層強化学習モデルの学習を実現するものである．深層強化学習エージェントは学習を行った人工市場環境に適応し，同環境下において安全かつ効率的な行動選択を行った．本研究の実験環境は現実市場に比べ非常に単純なものであるが，深層強化学習モデルが適切な行動戦略学習を行うことが確認され，人工市場を用いたデータ拡張が現実的なものとなった．加えて深層強化学習手法の比較実験より，実際にエージェントが予測の困難な状況に直面した際の挙動も確認でき，深層強化学習を用いた意思決定の安全性向上に対し有用な知見抽出がなされた.

模倣学習を用いた投資行動モデリング研究は，従来の模倣学習研究に金融市場が持つ行動戦略の特徴を導入することで，より妥当な投資行動のモデル化を実現するものである．金融市場の構造的特徴を織り込み学習された模倣学習モデルは良い市場のシミュレータとして機能し，現実市場では未観測の起こりうる市場シナリオを生成，生成データを予測モデルの学習対象として利用することで予測の安全性向上に貢献する．加えて，マーケットマイクロストラクチャーにおいて有用な知見抽出にもつながり，ミクロな視点，あるいはモデルベースな手法からの市場予測の発展につながると期待される.

ベイズ深層学習を用いた不確かさ考慮は，他の 2 研究では考慮できない，aleatoric uncertainty(4 章) の考慮を可能とするものである．データ拡張では解決できないノイズによる予測の不確実性を深層学習手法の改良により考慮し，モデルが見積もった不確かさをもとに行動戦略を変化させる．ベイズ深層学習により得られる幅のある予測を用いた行動アルゴリズムについては改善の余地があるもの，ベイズ深層学習の導入については有意に優れた結果が得られ，不確かさ考慮の安全性向上への寄与が明らかになった.

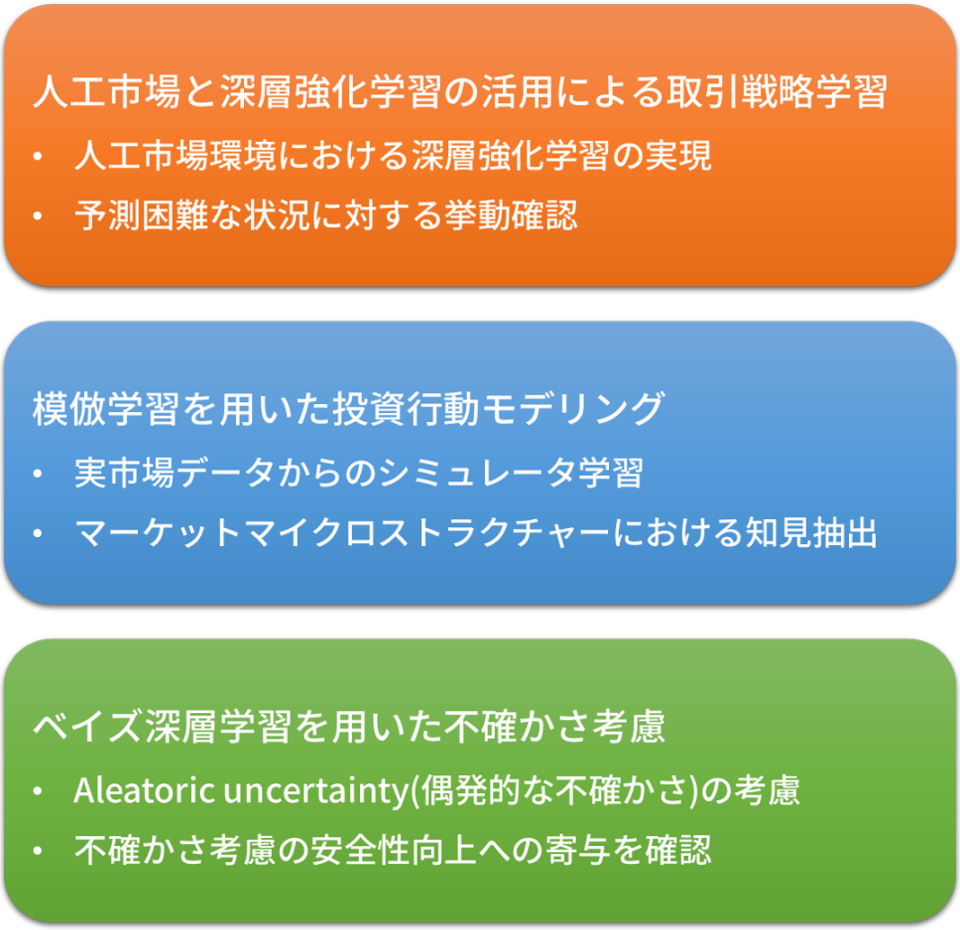


図 5.1: 本研究の貢献．人工市場と深層強化学習の活用による取引戦略学習，模倣学習を用いた投資行動モデリング，ベイズ深層学習を用いた不確かさ考慮それぞれが金融市場予測の安全性向上に寄与する成果を挙げている．

5.2　今後の展望

提案したフレームワークでは，単一市場かつ可能な注文の種類は限られ，環境エージェントは一種類，深層強化学習エージェントの行動候補は 33 通りで深層強化学習モデルも単純なものであった．人工市場と深層強化学習の結合，人工市場環境の構成，深層強化学習モデルの導入方法およびそれらの実装は常に手探りで，無数に課題が存在する．中でも人工市場環境の改良については，今後も進展が続くと考えている．加えて，人工市場環境における深層強化学習研究の活用は取引戦略学習にとどまらず，金融市場の制度設計や政策比較に用いる実験市場としての活用や，マーケットマイクロストラクチャーにおける有用な知見抽出にもつながると期待できる．

模倣学習を用いた投資行動モデリングは，マーケットマイクロストラクチャーに代表されるモデルベースなアプローチと，機械学習・深層学習を用いたデータドリブンなアプローチのちょうど中間に位置している．将来，投資家行動を完璧に表現できるモデルや，無限の学習能力を持つ人工知能が登場すれば本研究の存在価値は消滅するが，それは少なくとも当分先だろう．本研究で扱った株式市場含め，金融市場はあまりに巨大で複雑であり，単一の理論やせいぜい数十年分のデータで本質を掴める代物ではない．必要なアプローチはモデルベースおよびデータドリブンな手法の融合であると考えられる．本研究で提案した latent segmentation imitation learning はそのような融合の実現例であり，一定の成果は挙げたものの，まだまだ改善点は多い．特に投資家を分類する報酬関数の設計は非常に簡素なものであった．実市場データにおける投資家のクラスタリングは非常に困難であり，提案手法の有効性は示しにくい．こういった課題の解決により果たされるのはより正確な予測だけでなく，モデルベースの研究，データドリブンの研究両者にとって有用な知見抽出であり，今後のさらなる発展が期待される．

ベイズ深層学習を用いた不確かさ考慮は前述の 2 研究とは目的が異なり，金融市場･株式市場応用に限らない aleatoric uncertainty(偶発的な不確かさ) のモデル化を目的としている．このような目的の手法提案はベイズ深層学習の他にも数多く提案されており (4.2.4 章)，一般的な予測タスク，金融市場予測タスクについて比較実験を行うことで，さらなる発展が期待される．また，本研究では明確な優位性は示せなかったものの，予測された不確かさを活用した安全な意思決定アルゴリズムの開発も期待される．従来の決定的な予測に基づく意思決定プロセスは，特に安全性・危機管理の面で金融市場予測には不適切であることは本論文でも述べたとおりであり，データ拡張や計算時間の増加では対処できない，

予測可能性の問題への解決策として今後の発展が期待される．

本研究で提案した人工市場の活用はあくまで金融市場予測の安全性向上における１つの手段であり，目的達成のためには人工市場および人工市場以外の様々な分野における研究が進行し，それらが互いに相互作用する必要がある．投資家の行動原理をミクロな視点から解明するマーケットマイクロストラクチャーや，データからその関係性を正確に学習する機械学習・深層学習手法の発展はその最たる例である．今後も熱心に進められるであろう金融市場の研究に対し，本研究で提案した手法がその一助となると嬉しい．同分野の益々の発展を祈り，本論文の結びとする．

参考文献

[1] S.-H. Poon and C. W. Granger, "Forecasting volatility in financial markets: A review," *Journal of economic literature*, vol. 41, no. 2, pp. 478–539, 2003.

[2] A.-S. Chen, M. T. Leung, and H. Daouk, "Application of neural networks to an emerging financial market: forecasting and trading the taiwan stock index," *Computers & Operations Research*, vol. 30, no. 6, pp. 901–923, 2003.

[3] M. R. Hassan, B. Nath, and M. Kirley, "A fusion model of hmm, ann and ga for stock market forecasting," *Expert systems with Applications*, vol. 33, no. 1, pp. 171–180, 2007.

[4] M.-W. Hsu, S. Lessmann, M.-C. Sung, T. Ma, and J. E. Johnson, "Bridging the divide in financial market forecasting: machine learners vs. financial economists," *Expert Systems with Applications*, vol. 61, pp. 215–234, 2016.

[5] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: Machine learning techniques applied to financial market prediction," *Expert Systems with Applications*, vol. 124, pp. 226–251, 2019.

[6] B. G. Malkiel and E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[7] B. Mandelbrot, "New methods in statistical economics," *Journal of political economy*, vol. 71, no. 5, pp. 421–440, 1963.

[8] X. Gabaix, P. Gopikrishnan, V. Plerou, and H. E. Stanley, "A theory of power-law distributions in financial market fluctuations," *Nature*, vol. 423, no. 6937, pp. 267–270, 2003.

[9] T. Lux and M. Marchesi, "Scaling and criticality in a stochastic multi-agent model of a financial market," *Nature*, vol. 397, no. 6719, pp. 498–500, 1999.

[10] P. W. Anderson, *The economy as an evolving complex system*. CRC Press, 2018.

[11] W. B. Arthur, *The economy as an evolving complex system II*. CRC Press, 2018.

[12] E. Samanidou, E. Zschischang, D. Stauffer, and T. Lux, "Agent-based models of financial markets," *Reports on Progress in Physics*, vol. 70, no. 3, p. 409, 2007.

[13] G.-r. Kim and H. M. Markowitz, "Investment rules, margin, and market volatility," *Journal of Portfolio Management*, vol. 16, no. 1, p. 45, 1989.

[14] M. Levy, H. Levy, and S. Solomon, "A microscopic model of the stock market: cycles, booms, and crashes," *Economics Letters*, vol. 45, no. 1, pp. 103–111, 1994.

[15] M. Levy and S. Solomon, "Power laws are logarithmic boltzmann laws," *International Journal of Modern Physics C*, vol. 7, no. 04, pp. 595–601, 1996.

[16] S. Solomon and M. Levy, "Spontaneous scaling emergence in generic stochastic systems," *International Journal of Modern Physics C*, vol. 7, no. 05, pp. 745–751, 1996.

[17] V. K. Shante and S. Kirkpatrick, "An introduction to percolation theory," *Advances in Physics*, vol. 20, no. 85, pp. 325–357, 1971.

[18] D. Stauffer, "Percolation models of financial market dynamics," *Advances in Complex Systems*, vol. 4, no. 01, pp. 19–27, 2001.

[19] R. Cont and J.-P. Bouchaud, "Herd behavior and aggregate fluctuations in financial markets," *Macroeconomic Dynamics*, vol. 4, no. 2, p. 170–196, 2000.

[20] S. Solomon, G. Weisbuch, L. de Arcangelis, N. Jan, and D. Stauffer, "Social percolation models," *Physica A: Statistical Mechanics and its Applications*, vol. 277, no. 1-2, pp. 239–247, 2000.

[21] J. Goldenberg, B. Libai, S. Solomon, N. Jan, and D. Stauffer, "Marketing percolation," *Physica A: statistical mechanics and its applications*, vol. 284, no. 1-4, pp. 335–347, 2000.

[22] B. LeBaron, W. B. Arthur, and R. Palmer, "Time series properties of an artificial stock market," *Journal of Economic Dynamics and control*, vol. 23, no. 9-10, pp. 1487–1516, 1999.

[23] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS one*, vol. 12, no. 7, p. e0180944, 2017.

[24] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[25] N. Silver, *The Signal and the Noise: Why So Many Predictions Fail-but Some Don't*. Penguin Publishing Group, 2012.

[26] M. Brunnermeier and M. Oehmke, "Complexity in financial markets," *Princeton University*, vol. 8, 2009.

[27] A. Christer and W. Goodbody, "Equipment replacement in an unsteady economy," *Journal of the Operational Research Society*, vol. 31, no. 6, pp. 497–506, 1980.

[28] L. K. Chan and J. Lakonishok, "The behavior of stock prices around institutional trades," *The Journal of Finance*, vol. 50, no. 4, pp. 1147–1174, 1995.

[29] R. Almgren, C. Thum, E. Hauptmann, and H. Li, "Direct estimation of equity market impact," *Risk*, vol. 18, no. 7, pp. 58–62, 2005.

[30] E. Moro, J. Vicente, L. G. Moyano, A. Gerig, J. D. Farmer, G. Vaglica, F. Lillo, and R. N. Mantegna, "Market impact and trading profile of hidden orders in stock markets," *Physical Review E*, vol. 80, no. 6, p. 066102, 2009.

[31] R. Almgren and T. M. Li, "Option hedging with smooth market impact," *Market microstructure and liquidity*, vol. 2, no. 01, p. 1650002, 2016.

[32] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *Proceedings of International Conference on Learning Representations*, 2016.

[33] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," in *Advances in Neural Information Processing Systems*, pp. 14707–14718, 2019.

[34] D. R. Stockwell and A. T. Peterson, "Effects of sample size on accuracy of species distribution models," *Ecological modelling*, vol. 148, no. 1, pp. 1–13, 2002.

[35] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[36] L. S. Bamber, "Unexpected earnings, firm size, and trading volume around quarterly earnings announcements," *Accounting review*, pp. 510–532, 1987.

[37] H. Bessembinder and P. J. Seguin, "Price volatility, trading volume, and market depth: Evidence from futures markets," *Journal of financial and Quantitative Analysis*, pp. 21–39, 1993.

[38] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *Journal of the American statistical Association*, vol. 82, no. 398, pp. 528–540, 1987.

[39] S. Frühwirth-Schnatter, "Data augmentation and dynamic linear models," *Journal of time series analysis*, vol. 15, no. 2, pp. 183–202, 1994.

[40] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[41] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[42] M. Raberto, S. Cincotti, S. M. Focardi, and M. Marchesi, "Agent-based simulation of a financial market," *Physica A: Statistical Mechanics and its Applications*, vol. 299, no. 1-2, pp. 319–327, 2001.

[43] T. Zhang and D. Zhang, "Agent-based simulation of consumer purchase decision-making and the decoy effect," *Journal of business research*, vol. 60, no. 8, pp. 912–922, 2007.

[44] B. Zenobia, C. Weber, and T. Daim, "Artificial markets: A review and assessment of a new venue for innovation research," *Technovation*, vol. 29, no. 5, pp. 338–350, 2009.

[45] N. Raman and J. L. Leidner, "Financial market data simulation using deep intelligence agents," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pp. 200–211, Springer, 2019.

[46] A. C. Harvey and A. Jaeger, "Detrending, stylized facts and the business cycle," *Journal of applied econometrics*, vol. 8, no. 3, pp. 231–247, 1993.

[47] R. Levine and A. Demirgüç-Kunt, *Stock market development and financial intermediaries: stylized facts*. The World Bank, 1999.

[48] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," 2001.

[49] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013. NIPS Deep Learning Workshop 2013.

[50] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning.," in *ICLR (Poster)*, 2016.

[51] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[52] B. LeBaron, "Building the santa fe artificial stock market," *Physica A*, pp. 1–20, 2002.

[53] J. Rust, R. Palmer, and J. H. Miller, "Behaviour of trading automata in a computerized double auction market," Santa Fe Institute, 1992.

[54] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[55] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*, vol. 135. MIT press Cambridge, 1998.

[56] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[57] Y. Nevmyvaka, Y. Feng, and M. Kearns, "Reinforcement learning for optimized trade execution," in *Proceedings of the 23rd international conference on Machine learning*, pp. 673–680, 2006.

[58] Z. Jiang and J. Liang, "Cryptocurrency portfolio management with deep reinforcement learning," in *2017 Intelligent Systems Conference (IntelliSys)*, pp. 905–913, IEEE, 2017.

[59] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," *arXiv preprint arXiv:1706.10059*, 2017.

[60] T. Spooner, J. Fearnley, R. Savani, and A. Koukorinis, "Market making via reinforcement learning," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, (Richland, SC), p. 434–442, International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[61] K. S. Zarkias, N. Passalis, A. Tsantekidis, and A. Tefas, "Deep reinforcement learning for financial trading using price trailing," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3067–3071, IEEE, 2019.

[62] T. G. Dietterich, "Hierarchical reinforcement learning with the maxq value function decomposition," *Journal of artificial intelligence research*, vol. 13, pp. 227–303, 2000.

[63] M. L. Littman, "Value-function reinforcement learning in markov games," *Cognitive systems research*, vol. 2, no. 1, pp. 55–66, 2001.

[64] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep

reinforcement learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[65] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 2094–2100, AAAI Press, 2016.

[66] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*, pp. 1995–2003, 2016.

[67] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[68] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, and D. Silver, "Distributed prioritized experience replay," in *International Conference on Learning Representations*, 2018.

[69] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Advances in neural information processing systems*, pp. 5048–5058, 2017.

[70] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, pp. 1057–1063, 2000.

[71] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, pp. 1928–1937, 2016.

[72] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," 2014.

[73] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, pp. 1889–1897, 2015.

[74] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[75] G. Lample and D. S. Chaplot, "Playing fps games with deep reinforcement learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, p. 2140–2146, AAAI Press, 2017.

[76] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[77] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.

[78] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, (Red Hook, NY, USA), p. 2455–2467, Curran Associates Inc., 2018.

[79] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 653–664, 2016.

[80] M. B. Garman, "Market microstructure," *Journal of financial Economics*, vol. 3, no. 3, pp. 257–275, 1976.

[81] D. F. Spulber, "Market microstructure and intermediation," *Journal of Economic perspectives*, vol. 10, no. 3, pp. 135–152, 1996.

[82] A. Madhavan, "Market microstructure: A survey," *Journal of financial markets*, vol. 3, no. 3, pp. 205–258, 2000.

[83] A. Demirguc-Kunt, L. Laeven, and R. Levine, "Regulations, market structure, institutions, and the cost of financial intermediation," tech. rep., National Bureau of Economic Research, 2003.

[84] S. G. Hanson, A. K. Kashyap, and J. C. Stein, "A macroprudential approach to financial regulation," *Journal of economic Perspectives*, vol. 25, no. 1, pp. 3–28, 2011.

[85] B. LeBaron, "Agent-based computational finance: Suggested readings and early research," *Journal of Economic Dynamics and Control*, vol. 24, no. 5-7, pp. 679–702, 2000.

[86] L. Tesfatsion and K. L. Judd, *Handbook of computational economics: agent-based computational economics*. Elsevier, 2006.

[87] J. D. Farmer and D. Foley, "The economy needs agent-based modelling," *Nature*, vol. 460, no. 7256, pp. 685–686, 2009.

[88] I. Aldridge, *High-frequency trading: a practical guide to algorithmic strategies and trading systems*, vol. 604. John Wiley & Sons, 2013.

[89] J. Brogaard, T. Hendershott, and R. Riordan, "High-frequency trading and price discovery," *The Review of Financial Studies*, vol. 27, no. 8, pp. 2267–2306, 2014.

[90] A. Kirilenko, A. S. Kyle, M. Samadi, and T. Tuzun, "The flash crash: High-frequency trading in an electronic market," *The Journal of Finance*, vol. 72, no. 3, pp. 967–998, 2017.

[91] T. A. Hanson, "The effects of high frequency traders in a simulated market," in *Midwest Finance Association 2012 Annual Meetings Paper*, 2011.

[92] T. A. Hanson, "High frequency traders in a simulated market," *Review of Accounting and Finance*, 2016.

[93] F. McGroarty, A. Booth, E. Gerding, and V. R. Chinthalapati, "High frequency trading strategies, market fragility and price spikes: an agent based model perspective," *Annals of Operations Research*, vol. 282, no. 1-2, pp. 217–244, 2019.

[94] S. J. Leal and M. Napoletano, "Market stability vs. market resilience: Regulatory policies experiments in an agent-based model with low-and high-frequency trading," *Journal of Economic Behavior & Organization*, vol. 157, pp. 15–41, 2019.

[95] I. Maeda, D. deGraw, M. Kitano, H. Matsushima, H. Sakaji, K. Izumi, and A. Kato, "Deep reinforcement learning in agent based financial market simulation," *Journal of Risk and Financial Management*, vol. 13, no. 4, p. 71, 2020.

[96] D. Friedman, "The double auction market institution: A survey," *The double auction market: Institutions, theories, and evidence*, vol. 14, pp. 3–25, 1993.

[97] R. Das, J. E. Hanson, J. O. Kephart, and G. Tesauro, "Agent-human interactions in the continuous double auction," in *International joint conference on artificial intelligence*, vol. 17, pp. 1169–1178, Lawrence Erlbaum Associates Ltd, 2001.

[98] E. Smith, J. D. Farmer, L. s. Gillemot, S. Krishnamurthy, *et al.*, "Statistical theory of the continuous double auction," *Quantitative finance*, vol. 3, no. 6, pp. 481–514, 2003.

[99] I. Karatzas and S. E. Shreve, "Brownian motion," in *Brownian Motion and Stochastic Calculus*, pp. 47–127, Springer, 1998.

[100] R. R. Marathe and S. M. Ryan, "On the validity of the geometric brownian motion assumption," *The Engineering Economist*, vol. 50, no. 2, pp. 159–192, 2005.

[101] C. Chiarella, G. Iori, *et al.*, "A simulation analysis of the microstructure of double auction markets," *Quantitative finance*, vol. 2, no. 5, pp. 346–353, 2002.

[102] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, pp. 1008–1014, 2000.

[103] M. Wunder, M. L. Littman, and M. Babes, "Classes of multiagent q-learning dynamics with epsilon-greedy exploration," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 1167–1174, Citeseer, 2010.

[104] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[105] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Beijing, China), pp. 1556–1566, Association for Computational Linguistics, July 2015.

[106] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," in *2015 IEEE international conference on big data (big data)*, pp. 2823–2824, IEEE, 2015.

[107] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with lstm neural networks," in *2017 International joint conference on neural networks (IJCNN)*, pp. 1419–1426, IEEE, 2017.

[108] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.

[109] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on ceemdan and lstm," *Physica A: Statistical Mechanics and its Applications*, vol. 519, pp. 127–139, 2019.

[110] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Forecasting stock prices from the limit order book using convolutional neural networks," in *2017 IEEE 19th Conference on Business Informatics (CBI)*, vol. 1, pp. 7–12, IEEE, 2017.

[111] J. Niño, A. Arévalo, D. Leon, G. Hernandez, and J. Sandoval, "Price prediction with cnn and limit order book data," in *Workshop on Engineering Applications*, pp. 124–135, Springer, 2018.

[112] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Using deep learning for price prediction by exploiting stationary limit order book features," *Applied Soft Computing*, p. 106401, 2020.

[113] H. Demsetz, "The cost of transacting," *The quarterly journal of economics*, vol. 82, no. 1, pp. 33–53, 1968.

[114] Y. Amihud and H. Mendelson, "Liquidity and stock returns," *Financial Analysts Journal*, vol. 42, no. 3, pp. 43–48, 1986.

[115] J. Hasbrouck, "Assessing the quality of a security market: A new approach to transaction-cost measurement," *The Review of Financial Studies*, vol. 6, no. 1, pp. 191–212, 1993.

[116] M. Magdon-Ismail, A. Atiya, A. Pratap, and Y. Abu-Mostafa, "The maximum drawdown of the brownian motion," in *2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings.*, pp. 243–247, IEEE, 2003.

[117] M. Magdon-Ismail and A. F. Atiya, "Maximum drawdown," *Risk Magazine*, vol. 17, no. 10, pp. 99–102, 2004.

[118] Y. Amihud and H. Mendelson, "Dealership market: Market-making with inventory," *Journal of financial economics*, vol. 8, no. 1, pp. 31–53, 1980.

[119] U. A. Müller and R. B. Olsen, "Method for market making," Dec. 16 2008. US Patent 7,467,110.

[120] W. Fung and D. A. Hsieh, "The risk in hedge fund strategies: Theory and evidence from trend followers," *The review of financial studies*, vol. 14, no. 2, pp. 313–341, 2001.

[121] 前田巌, 松島裕康, 坂地泰紀, 和泉潔, ディグローデビット, , 加藤惇雄, and 北野道春, "人工市場と深層強化学習の融合による株式投資戦略学習," in *2020*年度 人工知能学会全国大会（第 *34* 回*)*, 2020.

[122] S. R. Hiltz and R. Goldman, *Learning together online: Research on asynchronous learning networks*. Routledge, 2004.

[123] B. Offir, Y. Lev, and R. Bezalel, "Surface and deep learning processes in distance education: Synchronous versus asynchronous systems," *Computers & Education*, vol. 51, no. 3, pp. 1172–1183, 2008.

[124] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[125] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Slides of Lecture Neural Networks for Machine Learning*, vol. 14, no. 8, 2012.

[126] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On optimization methods for deep learning," in *ICML*, 2011.

[127] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[128] C. De Sa, K. Olukotun, and C. Ré, "Ensuring rapid mixing and low bias for asynchronous gibbs sampling," in *JMLR workshop and conference proceedings*, vol. 48, p. 1567, NIH Public Access, 2016.

[129] D. K. Gode and S. Sunder, "Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality," *Journal of political economy*, vol. 101, no. 1, pp. 119–137, 1993.

[130] J. D. Farmer, P. Patelli, and I. I. Zovko, "The predictive power of zero intelligence in financial markets," *Proceedings of the National Academy of Sciences*, vol. 102, no. 6, pp. 2254–2259, 2005.

[131] A. Othman, "Zero-intelligence agents in prediction markets," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pp. 879–886, Citeseer, 2008.

[132] D. Ladley, "Zero intelligence in economics and finance," *Knowledge Engineering Review*, vol. 27, no. 2, pp. 273–286, 2012.

[133] W. F. Sharpe, "The sharpe ratio," *Journal of portfolio management*, vol. 21, no. 1, pp. 49–58, 1994.

[134] H. Kita, K. Taniguchi, and Y. Nakajima, *Realistic Simulation of Financial Markets: Analyzing Market Behaviors by the Third Mode of Science*, vol. 4. Springer, 2016.

[135] I. Veryzhenko, L. Arena, E. Harb, and N. Oriol, "A reexamination of high frequency trading regulation effectiveness in an artificial market framework," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pp. 15–25, Springer, 2016.

[136] I. Veryzhenko, L. Arena, E. Harb, and N. Oriol, "Time to slow down for high-frequency trading? lessons from artificial markets," *Intelligent Systems in Accounting, Finance and Management*, vol. 24, no. 2-3, pp. 73–79, 2017.

[137] T. Torii, K. Izumi, and K. Yamada, "Shock transfer by arbitrage trading: analysis using multi-asset artificial market," *Evolutionary and Institutional Economics Review*, vol. 12, no. 2, pp. 395–412, 2015.

[138] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[139] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242–264, IGI global, 2010.

[140] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[141] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[142] V. S. Ramachandran, "Mirror neurons and imitation learning as the driving force behind " the great leap forward " in human evolution," 2000.

[143] G. Buccino, S. Vogt, A. Ritzl, G. R. Fink, K. Zilles, H.-J. Freund, and G. Rizzolatti, "Neural circuits underlying imitation learning of hand actions: an event-related fmri study," *Neuron*, vol. 42, no. 2, pp. 323–334, 2004.

[144] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668, 2010.

[145] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Boots, "Agile autonomous driving using end-to-end deep imitation learning," in *Proceedings of Robotics: Science and Systems*, (Pittsburgh, Pennsylvania), June 2018.

[146] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–9, IEEE, 2018.

[147] S. Schaal and C. G. Atkeson, "Learning control in robotics," *IEEE Robotics & Automation Magazine*, vol. 17, no. 2, pp. 20–29, 2010.

[148] C. Thurau, C. Bauckhage, and G. Sagerer, "Imitation learning at all levels of game-ai," in *Proceedings of the international conference on computer games, artificial intelligence, design and education*, vol. 5, 2004.

[149] Y. Duan, M. Andrychowicz, B. Stadie, O. J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-shot imitation learning," in *Advances in neural information processing systems*, pp. 1087–1098, 2017.

[150] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *Proceedings of the 1st Annual Conference on Robot Learning* (S. Levine, V. Vanhoucke, and K. Goldberg, eds.), vol. 78 of *Proceedings of Machine Learning Research*, pp. 357–368, PMLR, 13–15 Nov 2017.

[151] B. C. Stadie, P. Abbeel, and I. Sutskever, "Third-person imitation learning," *Proceedings of International Conference on Learning Representations*, 2017.

[152] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in neural information processing systems*, pp. 4565–4573, 2016.

[153] J. Song, H. Ren, D. Sadigh, and S. Ermon, "Multi-agent generative adversarial imitation learning," in *Advances in neural information processing systems*, pp. 7461–7472, 2018.

[154] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially compliant navigation through raw depth inputs with generative adversarial imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1111–1117, IEEE, 2018.

[155] N. Baram, O. Anschel, I. Caspi, and S. Mannor, "End-to-end differentiable adversarial imitation learning," in *International Conference on Machine Learning*, pp. 390–399, 2017.

[156] F. Torabi, G. Warnell, and P. Stone, "Generative adversarial imitation from observation," June 2019.

[157] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Icml*, vol. 1, p. 2, 2000.

[158] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, p. 1, 2004.

[159] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning.," in *IJCAI*, vol. 7, pp. 2586–2591, 2007.

[160] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning.," in *Aaai*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.

[161] A. Kuefler and M. J. Kochenderfer, "Burn-in demonstrations for multi-modal imitation learning," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, (Richland, SC), p. 1071–1078, International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[162] L. Ke, M. Barnes, W. Sun, G. Lee, S. Choudhury, and S. Srinivasa, "Imitation learning as f-divergence minimization," *arXiv preprint arXiv:1905.12888*, 2019.

[163] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[164] K. Hausman, Y. Chebotar, S. Schaal, G. Sukhatme, and J. J. Lim, "Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 1235–1245, 2017.

[165] L. Zeng, X. Qi, and Y.-c. CHEN, "Bidding strategy for generating firm based on imitation learning [j]," *Proceedings of the CSEE*, vol. 31, 2008.

[166] M. F. Dixon, I. Halperin, and P. Bilokon, *Machine Learning in Finance*. Springer, 2020.

[167] S. H. Cohen and V. Ramaswamy, "Latent segmentation models," *Marketing Research*, vol. 10, no. 2, p. 14, 1998.

[168] J. Swait, "A structural equation model of latent segmentation and product choice for cross-sectional revealed preference choice data," *Journal of retailing and consumer services*, vol. 1, no. 2, pp. 77–89, 1994.

[169] A. Bhatnagar and S. Ghose, "A latent class segmentation analysis of e-shoppers," *Journal of business research*, vol. 57, no. 7, pp. 758–767, 2004.

[170] J. Ezeobiejesi and B. Bhanu, "Latent fingerprint image segmentation using deep neural network," in *Deep Learning for Biometrics*, pp. 83–107, Springer, 2017.

[171] T. Foucault, S. Moinas, and E. Theissen, "Does anonymity matter in electronic limit order markets?," *The Review of Financial Studies*, vol. 20, no. 5, pp. 1707–1747, 2007.

[172] C. Comerton-Forde and K. M. Tang, "Anonymity, liquidity and fragmentation," *Journal of Financial Markets*, vol. 12, no. 3, pp. 337–367, 2009.

[173] I. Maeda, D. deGraw, M. Kitano, H. Matsushima, K. Izumi, H. Sakaji, and A. Kato, "Latent segmentation of stock trading strategies using multi-modal imitation learning," *Journal of Risk and Financial Management*, vol. 13, no. 11, p. 250, 2020.

[174] J. C. Stein, "Efficient capital markets, inefficient firms: A model of myopic corporate behavior," *The quarterly journal of economics*, vol. 104, no. 4, pp. 655–669, 1989.

[175] A. Shleifer, *Inefficient markets: An introduction to behavioural finance*. OUP Oxford, 2000.

[176] B. Mandelbrot and R. L. Hudson, *The Misbehavior of Markets: A fractal view of financial turbulence*. Basic books, 2007.

[177] N. N. Taleb, *The black swan: The impact of the highly improbable*, vol. 2. Random house, 2007.

[178] J. B. De Long, A. Shleifer, L. H. Summers, and R. J. Waldmann, "Noise trader risk in financial markets," *Journal of political Economy*, vol. 98, no. 4, pp. 703–738, 1990.

[179] A. Shleifer and L. H. Summers, "The noise trader approach to finance," *Journal of Economic perspectives*, vol. 4, no. 2, pp. 19–33, 1990.

[180] D. Tashiro, H. Matsushima, K. Izumi, and H. Sakaji, "Encoding of high-frequency order information and prediction of short-term stock price by deep learning," *Quantitative Finance*, vol. 19, no. 9, pp. 1499–1506, 2019.

[181] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[182] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[183] Z. Yang, W. Chen, F. Wang, and B. Xu, "Improving neural machine translation with conditional sequence generative adversarial nets," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 1346–1355, Association for Computational Linguistics, June 2018.

[184] S. J. Grossman and M. H. Miller, "Liquidity and market structure," *the Journal of Finance*, vol. 43, no. 3, pp. 617–633, 1988.

[185] A. J. Menkveld, "High frequency trading and the new market makers," *Journal of financial Markets*, vol. 16, no. 4, pp. 712–740, 2013.

[186] N. J. Renton and M. Sweeting, "Managing the execution of trades between market makers," June 19 2018. US Patent 10,002,385.

[187] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *Advances in neural information processing systems*, pp. 5574–5584, 2017.

[188] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

[189] A. Clark, *Surfing uncertainty: Prediction, action, and the embodied mind*. Oxford University Press, 2015.

[190] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.

[191] Y. Gal, "Uncertainty in deep learning," *University of Cambridge*, vol. 1, no. 3, 2016.

[192] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems*, pp. 6402–6413, 2017.

[193] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, (Stockholmsmässan, Stockholm Sweden), pp. 2796–2804, PMLR, Jul 2018.

[194] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Advances in Neural Information Processing Systems*, pp. 3179–3189, 2018.

[195] T. Nair, D. Precup, D. L. Arnold, and T. Arbel, "Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation," *Medical image analysis*, vol. 59, p. 101557, 2020.

[196] M. Pelikan, D. E. Goldberg, E. Cantú-Paz, *et al.*, "Boa: The bayesian optimization algorithm," in *Proceedings of the genetic and evolutionary computation conference GECCO-99*, vol. 1, pp. 525–532, 1999.

[197] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, pp. 2951–2959, 2012.

[198] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate bayesian a posteriori probabilities," *Neural computation*, vol. 3, no. 4, pp. 461–483, 1991.

[199] A. Bate, M. Lindquist, I. R. Edwards, S. Olsson, R. Orre, A. Lansner, and R. M. De Freitas, "A bayesian neural network method for adverse drug reaction signal generation," *European journal of clinical pharmacology*, vol. 54, no. 4, pp. 315–321, 1998.

[200] A. P. Dempster, "A generalization of bayesian inference," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 30, no. 2, pp. 205–232, 1968.

[201] C. J. Geyer, "Practical markov chain monte carlo," *Statistical science*, pp. 473–483, 1992.

[202] Z. Shun and P. McCullagh, "Laplace approximation of high dimensional integrals," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 4, pp. 749–760, 1995.

[203] S. Kullback, *Information theory and statistics*. Courier Corporation, 1997.

[204] X. Yang, "Understanding the variational lower bound," 2017.

[205] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2014.

[206] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1613–1622, PMLR, 07–09 Jul 2015.

[207] D. A. Knowles, "Stochastic gradient variational bayes for gamma approximating distributions," *arXiv preprint arXiv:1509.01631*, 2015.

[208] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[209] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, pp. 1050–1059, 2016.

[210] J. Hron, A. G. d. G. Matthews, and Z. Ghahramani, "Variational gaussian dropout is not bayesian," *arXiv preprint arXiv:1711.02989*, 2017.

[211] D. Molchanov, A. Ashukha, and D. Vetrov, "Variational dropout sparsifies deep neural networks," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 2498–2507, PMLR, 06–11 Aug 2017.

[212] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Advances in neural information processing systems*, pp. 2575–2583, 2015.

[213] V. Kharitonov, D. Molchanov, and D. Vetrov, "Variational dropout via empirical bayes," *arXiv preprint arXiv:1811.00596*, 2018.

[214] Y. Liu, W. Dong, L. Zhang, D. Gong, and Q. Shi, "Variational bayesian dropout with a hierarchical prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7124–7133, 2019.

[215] Y. Wang, W. Dai, C. Li, J. Zou, and H. Xiong, "Si-vdnas: Semiimplicit variational dropout for hierarchical one-shot neural architecture search," in *International Joint Conference on Artificial Intelligence*, 2020.

[216] T. Blau, L. Ott, and F. Ramos, "Improving reinforcement learning pre-training with variational dropout," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4115–4122, IEEE, 2018.

[217] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[218] 前田巌, 松島裕康, 坂地泰紀, 和泉潔, ディグローデビット, 富岡博和, 加藤惇雄, and 北野道春, "深層学習における不確かさ評価の重要性," in *2019*年度人工知能学会全国大会（第 *33*回, 2019.

[219] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[220] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.

[221] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Baltimore, Maryland), pp. 655–665, Association for Computational Linguistics, June 2014.

[222] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 448–456, PMLR, Jul 2015.

[223] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[224] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.

[225] Y. Bengio, F. Bastien, A. Bergeron, N. Boulanger–Lewandowski, T. Breuel, Y. Chherawala, M. Cisse, M. Côté, D. Erhan, J. Eustache, X. Glorot, X. Muller, S. P. Lebeuf, R. Pascanu, S. Rifai, F. Savard, and G. Sicard, "Deep learners benefit more from out-of-distribution examples," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudík, eds.), vol. 15 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 164–172, JMLR Workshop and Conference Proceedings, 11–13 Apr 2011.

[226] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.

[227] S. Menard, *Applied logistic regression analysis*, vol. 106. Sage, 2002.

[228] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*, vol. 398. John Wiley & Sons, 2013.

[229] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, p. 1321–1330, JMLR.org, 2017.

[230] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[231] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[232] P. Marshall, N. Rajguru, and A. Slosar, "Bayesian evidence as a tool for comparing datasets," *Physical Review D*, vol. 73, no. 6, p. 067302, 2006.

[233] T. Minka, "Estimating a dirichlet distribution," 2000.

[234] A. Kumar, S. Sarawagi, and U. Jain, "Trainable calibration measures for neural networks from kernel mean embeddings," in *International Conference on Machine Learning*, pp. 2805–2814, 2018.

[235] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, vol. 2015, p. 2901, NIH Public Access, 2015.

[236] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[237] I. Maeda, H. Matsushima, H. Sakaji, K. Izumi, D. deGraw, H. Tomioka, A. Kato, and M. Kitano, "Learning uncertainty in market trend forecast using bayesian neural networks," in *Decision Economics: Complexity of Decisions and Decisions for Complexity* (E. Bucciarelli, S.-H. Chen, and J. M. Corchado, eds.), (Cham), pp. 210–218, Springer International Publishing, 2020.

[238] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition.," in *Ijcai*, vol. 15, pp. 3995–4001, Citeseer, 2015.

[239] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[240] T. Mikolov, S. Kombrink, L. Burget, J. Černockỳ, and S. Khudanpur, "Extensions of recurrent neural network language model," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5528–5531, IEEE, 2011.

[241] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[242] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker, "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *Journal of the American Statistical Association*, vol. 86, no. 416, pp. 953–963, 1991.

[243] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, pp. 1–15, Springer, 2000.

[244] G. J. Alexander, "On back-testing " zero-investment " strategies," *The Journal of Business*, vol. 73, no. 2, pp. 255–278, 2000.

[245] C. R. Harvey and Y. Liu, "Backtesting," *The Journal of Portfolio Management*, vol. 42, no. 1, pp. 13–28, 2015.

[246] D. H. Bailey, J. Borwein, M. Lopez de Prado, and Q. J. Zhu, "The probability of backtest overfitting," *Journal of Computational Finance, forthcoming*, 2016.

[247] I. Maeda, H. Matsushima, H. Sakaji, K. Izumi, D. deGraw, A. Kato, and M. Kitano, "Effectiveness of uncertainty consideration in neural-network-based financial forecasting," in *2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pp. 673–678, IEEE, 2019.

[248] I. Maeda, H. Matsushima, H. Sakaji, K. Izumi, D. deGraw, A. Kato, and M. Kitano, "Predictive uncertainty in neural network-based financial market forecasting," *International Journal of Smart Computing and Artificial Intelligence*, 2020 in press.

[249] T. Hendershott, R. Riordan, *et al.*, "Algorithmic trading and information," *Manuscript, University of California, Berkeley*, 2009.

[250] A. P. Chaboud, B. Chiquoine, E. Hjalmarsson, and C. Vega, "Rise of the machines: Algorithmic trading in the foreign exchange market," *The Journal of Finance*, vol. 69, no. 5, pp. 2045–2084, 2014.

[251] M. J. Brennan and T. E. Copeland, "Stock splits, stock prices, and transaction costs," *Journal of financial economics*, vol. 22, no. 1, pp. 83–101, 1988.

[252] L. Lang, E. Ofek, and R. Stulz, "Leverage, investment, and firm growth," *Journal of financial Economics*, vol. 40, no. 1, pp. 3–29, 1996.

[253] T. Adrian and H. S. Shin, "Liquidity and leverage," *Journal of financial intermediation*, vol. 19, no. 3, pp. 418–437, 2010.

[254] Z. Govindarajulu, "Distribution-free confidence bounds for p (x¡ y)," *Annals of the institute of statistical mathematics*, vol. 20, no. 2, pp. 229–38, 1968.

[255] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.

[256] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, pp. 63–71, Springer, 2003.

[257] N. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal of machine learning research*, vol. 6, no. Nov, pp. 1783–1816, 2005.

[258] B. J. Winer, "Statistical principles in experimental design.," 1962.

[259] R. E. Kirk, "Experimental design," *Handbook of Psychology, Second Edition*, vol. 2, 2012.

[260] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, (Arlington, Virginia, USA), p. 109–116, AUAI Press, 2009.

[261] T. Van Laarhoven, "L2 regularization versus batch and weight normalization," *arXiv preprint arXiv:1706.05350*, 2017.

[262] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve.," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[263] F. Fressin, G. Torres, D. Charbonneau, S. T. Bryson, J. Christiansen, C. D. Dressing, J. M. Jenkins, L. M. Walkowicz, and N. M. Batalha, "The false positive rate of kepler and the occurrence of planets," *The Astrophysical Journal*, vol. 766, no. 2, p. 81, 2013.

[264] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, 2006.

[265] K. Boyd, K. H. Eng, and C. D. Page, "Area under the precision-recall curve: point estimates and confidence intervals," in *Joint European conference on machine learning and knowledge discovery in databases*, pp. 451–466, Springer, 2013.

[266] C. Memmel, "Performance hypothesis testing with the sharpe ratio," *Finance Letters*, vol. 1, no. 1, 2003.

[267] O. Ledoit and M. Wolf, "Robust performance hypothesis testing with the sharpe ratio," *Journal of Empirical Finance*, vol. 15, no. 5, pp. 850–859, 2008.

[268] J. M. Beck, W. J. Ma, R. Kiani, T. Hanks, A. K. Churchland, J. Roitman, M. N. Shadlen, P. E. Latham, and A. Pouget, "Probabilistic population codes for bayesian decision making," *Neuron*, vol. 60, no. 6, pp. 1142–1152, 2008.

[269] N. Dey, A. S. Ashour, and S. Borra, *Classification in BioApps: automation of decision making*, vol. 26. Springer, 2017.

謝辞

本論文の作成にあたり，多くの方々にお力添えを頂きました．適切な助言や温かな励ましをくださった皆様に深く感謝いたします．指導教員の和泉潔教授には，博士課程在籍中の 3 年間，多くの熱心なご指導，多くの有益なご助言を頂きました．金融市場・人工市場については素人同然であった私が同分野で研究成果を挙げることができたのはひとえに和泉先生のおかげです．心より感謝いたします．3 年間隣の席から多くのご助言を頂いた坂地泰紀特任講師には，私の成長を一番近くから見守って頂きました．釣りやゲームといった趣味の会話も含め，未熟な私の背中を押し続けて頂いたことに心より感謝申し上げます．博士 2 年終了時までご指導いただいた松島裕康特任助教 (当時) には，研究面での多くのご指導・ご助言に加え共同研究のマネジメントや研究環境の整備でもご助力頂きました．心より感謝いたします．短い期間ですがご指導いただきました特任研究員の向井大誠博士，佐野仁美博士にも感謝申し上げます．研究室秘書の山本由香さんには，研究室生活において数え切れないほどのサポートを頂きました．怠惰な私が不足なく博士後期課程を終えることができたのも山本さんのご指導あってです．心より感謝いたします．

大変ご多様な中，博士論文の審査にあたり副査を引き受けて下さった先生方には幾重にもお礼申し上げます．特に研究室ゼミはじめ普段から熱心にご指導頂き，トロントで行われた Rotman International Trading Competition でもご助力いただいた島田尚准教授には感謝してもしきれません．富山でもお話頂いた八木勲教授，予備審査で多くの有益なご意見を頂いた古田一夫教授，米倉一男講師に心より感謝申し上げます．

私の研究はその全てが株式会社大和証券，および大和総研株式会社のご助力の元行われました．大変お忙しい立場にありながら頻繁にディスカッションに来てくださり，稚拙な研究を暖かく支えて下さった David deGraw 様，短い期間ではありましたが多くの研究にご参加くださった富岡博和様，共同研究全般のマネジメントにご尽力頂き，さらに私の研究内容についても数多くの有益なご意見をくださった加藤惇雄様，研究のコンセプトから詳細まで，ありとあらゆる点について熱心にご指導頂いた北野道春様，私の研究を全面的に支えてくださった皆様に心より御礼申し上げます．

研究室では多くの素晴らしい仲間に囲まれ，楽しく充実した時間を過ごせました．数々の輝かしい研究実績を出し，常に私の目標であり続けた伊藤友貴博士，お仕事と両立しながら博士号を取得し，スペインの学会等多くの機会でご一緒いただいた余野京登博士，高い社会的地位と大変なお忙しさにも関わらず何度もお酒の場にご一緒頂き，進路や将来の悩みについても多くのご助言を頂いた水門善之さん，現在は休学中ですが，注文履歴データや人工市場設計等様々な面でサポートいただいた小林弘幸さん，研究面・研究室インフラ面含めありとあらゆる研究室の活動にご尽力いただき，事ある毎に有益なディスカッションをしてくださった平野正徳さん，先行研究として多くを参考にさせていただいた 2018 年修士卒の田代大悟さん，2019 年修士卒の王洲浩さん，堅木聖也さん，蔵本涼太さん，曽根泰平さん，寺尾仁志さん，2020 年修士卒の松浦出さん，五十嵐光秋さん，高嶺航さん，濱脇諒さん， 現在研究室に所属している許タオさん，蘭盈盈さん，仁木裕太さん，松原冬樹さん，Cong Liu さん，鈴木雅弘さん，若杉亮さん，王在舸さん，陳思哲さん，任浪さん，晴海勝さん，小村一紘さん，藤江林旭さん，小野潔さん皆様に感謝しております．慣れない地で慣れない英語で必死に戦った Rotman International Trading Competition，前日に誘われ堅木さん曽根さんと楽しんだ USJ，語るべき思い出には際限ありませんが，それら一つひとつが心の支えになっています．

研究室の外においても，本当に多くの方々に支えられて博士課程を歩んできました．学部 3 年時から現在に至るまで様々な経験をさせていただき，アルバイトの枠を大きく，大きく超えてお世話になった高木塾の皆様，無愛想な私を暖かく受け入れ，常に元気と笑顔を届けていただいたバー，Abbot’s Choice，ma.YU.ge，AMUSEMENT の皆様，毎晩のようにオンラインゲームで遊んだ高校の部活動の友人たち，すべての方々に感謝しています．

自身の定めた目標に向けて未開の地を進み，目に見える成果を求められる．博士課程は想像を絶する困難の道でした．私が苦しみながら前に進むことができたのは，ひとえに皆様のおかげです．皆様に支えられ多くの挑戦と失敗を繰り返した分，現在の私は進学前から何倍にも成長している，はずです．重ねてになりますが関わっていただいた皆様への感謝と...

最後に，常に私の一番の味方であり続けてくれた家族への万謝を表し，本論文の締めと致します．

2020 年 12 月 1 日　　前田 巌