

# アメリカンオプションに対する Deep Hedging の学習方法に関する考察\*

佐久間達也<sup>1</sup>

<sup>1</sup> 東京都立大学大学院 経営学研究科 経営学専攻 博士前期課程

初回提出日：2023年1月10日

## 1 はじめに

本稿ではニューラルネットワーク (NN) を用いたデリバティブのヘッジ手法である Deep Hedging をアメリカンオプションに対して適用する方法について議論する。

数理ファイナンスの分野における重要な問題の一つとしてデリバティブの最適ヘッジ問題が挙げられる。この問題は最も基本的な市場モデルである Black-Scholes モデルにおいて仮定されている完備市場という設定だけでなく、様々な制約が課された現実の市場に近い設定である非完備市場という設定においても研究が進んでいる。非完備市場におけるデリバティブの価値評価や最適ヘッジの問題は、数学的には確率制御問題の枠組に位置しており、明示的表現の導出や数値計算方法について多くの研究がなされてきた。しかし既存の手法では原資産のモデルがジャンプを含むモデルや多次元になった場合に解を陽に表現することが難しく、数値計算において計算負荷が非常に大きくなってしまふことが課題とされている。

このようなファイナンスにおける高次元な確率制御問題の数値計算に対しても有効な手法として深層学習の応用が盛んに研究されている。Buehler et al. [2019] では深層強化学習の応用として、普遍近似定理 (Hornik [1991]) に基づき、最適ヘッジ戦略の制御変数であるヘッジポジションを NN によって近似する Deep Hedging と呼ばれるフレームワークを提案した。この手法は原資産のモデルや次元数によらず計算が可能であり、Semi-recurrent NN と呼ばれる結合関係を持つ NN で計算を行うことで、リバランスに伴い発生する取引コストなどの制約も取り込むこともできることからゲームチェンジャーとして期待されており、現在まで応用的な研究がなされている。

Deep Hedging をアメリカンオプションに応用した先行研究には、Becker et al. [2020] がある。この論文ではアメリカンオプションの最適停止時刻を決定する関数を NN により近似する手法と、ヘッジポジションを近似する NN の学習方法を示し、シミュレーションにより精度の検証を行っている。しかし彼らの論文で提案されている NN は Semi-recurrent NN ではなく結合関係を持たない Feed Forward NN である。

以上をふまえ本稿では、アメリカンオプションにおける再帰構造のある NN の学習方法の課題点に言及し、その課題点に対応した新しい NN の学習方法 (Two-Steps Learning Model) を提案する。今回新しく提案する Two-Steps Learning Model とは、直接アメリカンオプションをヘッジする NN を学習するのではなく、

---

\* 本論文は、東京都立大学大学院経営学研究科に提出した修士論文を加筆・修正したものである。

ヨーロッパオプションをヘッジする NN とアメリカンオプションの最適なヘッジとの差を NN で学習することにより、アメリカンオプションをヘッジする NN を構成する学習方法である。

検証では、シミュレーションを用いて各手法におけるアメリカンオプションのヘッジ精度の比較を行った。その結果、アメリカンオプションに対する Deep Hedging においても NN に再帰構造を組み込むことで、ヘッジ精度が向上することが示された。加えて Two-Steps Learning Model で学習することにより、多くの場合において、ヘッジ精度がさらに向上することを確認した。

最後に本稿の構成について述べる。2 章では本稿で扱うアメリカンオプションやヘッジ評価の設定について説明する。3 章と 4 章では Deep Hedging について説明する。3 章では基本的な Deep Hedging のフレームワークの説明を行い、4 章ではアメリカンオプションに対して従来の Deep Hedging の手法を直接適用する One-Step Learning Model と、提案手法である Two-Steps Learning Model について説明する。5 章では、シミュレーションにより、新しく提案したモデルの性能について分析を行い、6 章では本稿のまとめと今後の展望について述べる。

## 記号の定義

$\mathbb{N}$  を自然数全体の集合（ただし、0 は含まない）、 $\mathbb{Z}$  を整数全体の集合、 $\mathbb{R}$  を実数全体の集合とする。また、 $\mathbb{Z}_+$  を非負な整数全体の集合（すなわち、 $\mathbb{Z}_+ = \mathbb{N} \cup \{0\}$ ）、 $\mathbb{R}_+$  を非負な実数全体の集合（すなわち、 $\mathbb{R}_+ = [0, \infty)$ ）とする。

$\mathbb{R}^d$  を  $d$  次元ベクトル空間と同一視する。すなわち、 $\mathbb{R}^d$  の要素を縦ベクトルとして扱うことがある。また、 $x \in \mathbb{R}^d$  に対して、 $x$  の  $i$  番目の要素を  $x^i$  とし、 $x$  の要素を明示する際は  $x = (x^1, \dots, x^d)$  と表記する。

## 2 アメリカンオプションとヘッジについて

### 2.1 アメリカンオプションと最小二乗モンテカルロ法

アメリカンオプションとは、満期までの任意の時点で権利行使可能なオプションである。また、バミューダンオプションとは、事前に定められた複数の時点から任意の時点を選択し、権利行使できるオプションである。離散時間市場モデルにおいてはアメリカンオプションはバミューダンオプションの特別な場合となる。本 2.1 章では、離散時間市場モデルにおけるアメリカンオプションを考える。

$T > 0$  を満期とし、 $N \in \mathbb{N}$  を閉区間  $[0, T]$  の分割数、 $\Delta t := T/N$  を分割幅、 $t_n := n\Delta t$ ,  $n \in \{0, 1, \dots, N\}$  を時点とする。すなわち、等間隔な離散時間  $0 = t_0 < t_1 < \dots < t_N = T$  を想定しており、この離散時間上で権利行使可能なアメリカンオプションを考える。以降、右下の添え字  $n$  は時点  $t_n$  に依存していることを意味する。

$(\Omega, \mathcal{F}, \mathbb{F} := (\mathcal{F}_n)_{n=0}^N, \mathbb{P})$  をフィルター付き確率空間とする。 $d$  種類のリスク資産は  $\mathbb{R}^d$ -値マルコフ過程<sup>\*1</sup>  $S = (S_n)_{n=0}^N$  に従うと仮定する。また、無リスク金利は定数  $r \in \mathbb{R}$  とする。

$\mathcal{T}$  を  $\mathbb{F}$ -停止時刻<sup>\*2</sup>全体の集合とし、 $n \in \{0, 1, \dots, N\}$  に対して、 $\mathcal{T}_n$  を  $n$  以上である  $\mathbb{F}$ -停止時刻全体の集合とする。すなわち、 $\mathcal{T} := \{\tau : \Omega \rightarrow \{1, \dots, N\}; \mathbb{F}\text{-停止時刻}\}$ 、 $\mathcal{T}_n := \{\tau \in \mathcal{T}; n \leq \tau\}$  である。このとき、 $(\mathcal{T}_n)_{n=0}^N$  は単調減少である。すなわち、包含関係  $\{N\} = \mathcal{T}_N \subset \mathcal{T}_{N-1} \subset \dots \subset \mathcal{T}_0 = \mathcal{T}$  が成り立っている。

<sup>\*1</sup> 確率過程  $S = (S_n)_{n=0}^N$  が  $\mathbb{F}$ -適合であり、次の性質を満たすとき、 $S$  をマルコフ過程と呼ぶ：任意の  $n \in \{0, 1, \dots, N-1\}$ 、 $k \in \{1, 2, \dots, N-n\}$ 、 $B \in \mathcal{B}(\mathbb{R}^d)$  に対して、 $\mathbb{Q}(S_{n+k} \in B | \mathcal{F}_n) = \mathbb{Q}(S_{n+k} \in B | S_n)$  が成り立つ。

<sup>\*2</sup>  $\mathbb{F}$  に関する停止時刻を  $\mathbb{F}$ -停止時刻と呼ぶ。

**Definition 1** (アメリカンオプションの価値).  $g: \mathbb{R}^d \rightarrow \mathbb{R}$  をペイオフ関数、 $n \in \{0, 1, \dots, N\}$  とする。原資産<sup>\*3</sup>を  $S$  とするアメリカンオプションの時点  $t_n$  における価値  $V_n$  は次で与えられる：

$$V_n := \operatorname{ess\,sup}_{\tau \in \mathcal{T}_n} \mathbb{E}^{\mathbb{Q}} \left[ e^{-r\Delta t(\tau-n)} g(S_\tau) \mid \mathcal{F}_n \right]. \quad (2.1.1)$$

ただし、 $\mathbb{E}^{\mathbb{Q}}$  は  $\mathbb{P}$  の同値マルチンゲール測度  $\mathbb{Q}$  に関する期待値である。また、 $N \in \mathcal{T}_n$  より、 $\{\mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t(\tau-n)} g(S_\tau) \mid \mathcal{F}_n]; \tau \in \mathcal{T}_n\} \neq \emptyset$  であることに注意する。

式 (2.1.1) で定義した  $\mathbb{F}$ -適合過程  $(V_n)_{n=0}^N$  は Snell envelope であり、次のように後ろの時点から帰納的に決定できる：

$$V_n = \begin{cases} g(S_N) & n = N, \\ \max\{g(S_n), \mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{n+1} \mid \mathcal{F}_n]\} & n \in \{N-1, N-2, \dots, 1\}, \\ \mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_1] & n = 0. \end{cases} \quad (2.1.2)$$

また、式 (2.1.2) を用いると、式 (2.1.1) の上限を達成する  $\mathbb{F}$ -停止時刻の列  $(\tau_n^*)_{n=0}^N$  も次のように後ろの時点から帰納的に決定できる： $\mathcal{T}_N = \{N\}$  より、 $\tau_N^* = N$  であり、 $n \in \{N-1, N-2, \dots, 1\}$  に対して、式 (2.1.2) より、

$$\tau_n^* = \begin{cases} n & \text{if } g(S_n) \geq \mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{n+1} \mid \mathcal{F}_n], \\ \tau_{n+1}^* & \text{if } g(S_n) < \mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{n+1} \mid \mathcal{F}_n]. \end{cases} \quad (2.1.3)$$

ここで、式 (2.1.1) より、条件内における  $\mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{n+1} \mid \mathcal{F}_n]$  は  $\mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t(\tau_{n+1}^*-n)} g(S_{\tau_{n+1}^*}) \mid \mathcal{F}_n]$  と等しいことに注意する。なお、 $\tau_0^* = \tau_1^*$  であることに注意する。さらに、 $n \in \{1, 2, \dots, N-1\}$  に対して、式 (2.1.3) より、

$$\tau_n^* = \begin{cases} \min\{k \in \{n, n+1, \dots, N-1\}; g(S_k) \geq \mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{k+1} \mid \mathcal{F}_k]\} & \text{if } \{\dots\} \neq \emptyset, \\ N & \text{if } \{\dots\} = \emptyset \end{cases}$$

が成り立つ。ただし、 $\{\dots\} = \{k \in \{n, n+1, \dots, N-1\}; g(S_k) \geq \mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{k+1} \mid \mathcal{F}_k]\}$  である。なお、 $\tau_0^* = \tau_1^*$  だったので、

$$\tau_0^* = \begin{cases} \min\{k \in \{1, 2, \dots, N-1\}; g(S_k) \geq \mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{k+1} \mid \mathcal{F}_k]\} & \text{if } \{\dots\} \neq \emptyset, \\ N & \text{if } \{\dots\} = \emptyset \end{cases} \quad (2.1.4)$$

が成り立つ。ただし、 $\{\dots\} = \{k \in \{1, 2, \dots, N-1\}; g(S_k) \geq \mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{k+1} \mid \mathcal{F}_k]\}$  である。次に、アメリカンオプションの時点 0 における価値  $V_0$  の計算手法を考える。ここでは、式 (2.1.2), (2.1.4), (2.1.1) を経由して  $V_0$  を数値計算する手法を紹介する。

$n \in \{1, 2, \dots, N-1\}$  に対して、 $\mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{n+1} \mid \mathcal{F}_n]$  を時点  $t_n$  における継続価値と呼ぶ。式 (2.1.4) からわかるように、すべての時点における継続価値（条件付き期待値）を計算することができれば  $\tau_0^*$  の値が決まり、式 (2.1.1) の期待値を計算することで  $V_0$  の値を得ることができる。しかしながら一般に、前者の条件付き期待値、後者の期待値は明示的な陽の関数で表現することは難しいため、数値計算に頼る必要がある。これまでに様々な数値計算手法が提案されているが、ここでは、前者の条件付き期待値においては Longstaff and Schwartz [2001] が提案した最小二乗モンテカルロ（Least Square Monte Carlo, LSM）法を採用し、後者の期待値においては標準的なモンテカルロ法を採用する。以下で、LSM 法を簡単に解説する。

<sup>\*3</sup> 原資産はリスク資産の一つであることに注意する。本稿では、オプションの原資産以外のリスク資産は取り扱わない。

$X, Y \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathbb{P})$  に対して、条件付き期待値  $\mathbb{E}[Y | X]$  は  $\mathbb{E}[(Y - f(X))^2]$  を最小にする可測関数  $f \in \mathcal{L}^2(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d), dx)$  を用いて、 $\mathbb{E}[Y | X] \equiv f(X)$  と定義される。LSM 法とは、 $f$  を回帰によりシミュレーションすることで、 $\mathbb{E}[Y | X]$  の近似値を求める手法である。では、LSM 法を今回の場合に適用する。

はじめに、 $S$  のマルコフ性より、 $\mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{n+1} | \mathcal{F}_n] = \mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{n+1} | S_n]$  が成り立つことに注意する。<sup>\*4</sup>今、 $S_n, V_{n+1} \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathbb{Q})$  と仮定すると、条件付き期待値  $\mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{n+1} | \mathcal{F}_n]$  は  $\mathbb{E}^{\mathbb{Q}}[(e^{-r\Delta t} V_{n+1} - C_n(S_n))^2]$  を最小にする可測関数  $C_n \in \mathcal{L}^2(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d), dx)$  を用いて、 $\mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{n+1} | \mathcal{F}_n] \equiv C_n(S_n)$  と定義される。このとき、 $C_n$  を時点  $t_n$  における継続価値関数と呼び、 $\mathcal{L}^2(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d), dx)$  における基底関数  $(\phi_u)_{u=0}^{\infty}$  を用いて、 $C_n$  をフーリエ展開する。すなわち、 $a_{n,u} := \int_{\mathbb{R}^d} C_n(x) \phi_{n,u}(x) dx, u \in \mathbb{Z}_+$  とすると、

$$C_n = \sum_{u=0}^{\infty} a_{n,u} \phi_u$$

が成り立つ。言うまでもなく、右辺の無限級数は  $\mathcal{L}^2$ -収束している。このとき、 $C_n$  を  $U \in \mathbb{Z}_+$  まで級数展開し、 $L \in \mathbb{N}$  本のサンプルパスを用いて係数列  $(a_{n,u})_{u=0}^U$  を推定する。すなわち、継続価値関数  $C_n$  は次のように近似される： $(\hat{a}_{n,u}^L)_{u=0}^U$  を  $(a_{n,u})_{u=0}^U$  の推定値とすると、

$$C_n \approx \sum_{u=0}^U a_{n,u} \phi_u \approx \sum_{u=0}^U \hat{a}_{n,u}^L \phi_u =: \hat{C}_n^L.$$

実際、次のようにして実装する。

---

**アルゴリズム 1** 式 (2.1.2) を用いた継続価値関数の係数の推定

---

- 1:  $(S_n)_{n=0}^N$  の  $L$  本のサンプルパス  $(s_n^{(l)})_{n=0}^N, l \in \{1, 2, \dots, L\}$  を生成する。
- 2:  $\hat{V}_N^{(l)} \leftarrow g(s_N^{(l)}), l \in \{1, 2, \dots, L\}$
- 3: **for**  $n \in \{N-1, N-2, \dots, 2\}$  **do**
- 4: 次の二乗誤差を最小にするように  $(a_{n,u})_{u=0}^U$  の推定値  $(\hat{a}_{n,u}^L)_{u=0}^U$  を決める：

$$\inf_{a_{n,0}, a_{n,1}, \dots, a_{n,U}} \frac{1}{L} \sum_{l=1}^L \left( e^{-r\Delta t} \hat{V}_{n+1}^{(l)} - \sum_{u=0}^U a_{n,u} \phi_u(s_n^{(l)}) \right)^2.$$

- 5:  $\hat{V}_n^{(l)} \leftarrow \max\{g(s_n^{(l)}), \hat{C}_n^L(s_n^{(l)})\}, l \in \{1, 2, \dots, L\}$
- 6: 次の二乗誤差を最小にするように  $(a_{1,u})_{u=0}^U$  の推定値  $(\hat{a}_{1,u}^L)_{u=0}^U$  を決める：

$$\inf_{a_{1,0}, a_{1,1}, \dots, a_{1,U}} \frac{1}{L} \sum_{l=1}^L \left( e^{-r\Delta t} \hat{V}_2^{(l)} - \sum_{u=0}^U a_{1,u} \phi_u(s_1^{(l)}) \right)^2.$$


---

アルゴリズム 1 より、 $(a_{n,u})_{u=0}^U$  の推定値  $(\hat{a}_{n,u}^L)_{u=0}^U$  を得る。このとき、式 (2.1.2) をみると、一見、次のようにして  $V_0$  の推定値  $\hat{V}_0$  を数値計算することができるように思える。

---

<sup>\*4</sup> このことから、オプション価値も  $V_n = V(S_n)$  と記述できる。

---

アルゴリズム 2 式 (2.1.2) を用いた  $\hat{V}_0$  の数値計算

---

- 1:  $\hat{V}_1^{(l)} \leftarrow \max\{g(s_1^{(l)}), \hat{C}_1^L(s_1^{(l)})\}, l \in \{1, 2, \dots, L\}$
  - 2: **return**  $\hat{V}_0 = \frac{1}{L} \sum_{l=1}^L e^{-r\Delta t} \hat{V}_1^{(l)}$
- 

しかしこの推定した値は上方向と下方向のバイアスが混ざってしまっており、制御不可能なバイアスを発生させてしまう。(上下方向へのバイアスの性質や原因などの詳細については Glasserman [2004] を参照)

そこで、継続価値関数の係数の推定に用いたサンプルパスと独立なサンプルパスを新たに生成し、式 (2.1.4), (2.1.1) を用いて  $V_0$  の推定値  $\hat{V}_0$  を数値計算することを考える。独立なパスを使用して計算することで、 $\hat{V}_0$  の期待値が真の値  $V_0$  以下になるという下方向のバイアスを使用して、推定結果を制御する。(詳しくは Glasserman [2004] を参照)。実際、次のようにして実装する。

---

アルゴリズム 3 式 (2.1.4), (2.1.1) を用いた  $\hat{V}_0$  の数値計算

---

- 1: アルゴリズム 1 より、 $(C_n)_{n=1}^N$  の近似値  $(\hat{C}_n)_{n=1}^N$  を得る。
  - 2: アルゴリズム 1 で使用した  $(S_n)_{n=0}^N$  のサンプルパスと独立な  $L' \in \mathbb{N}$  本のサンプルパス  $(s_n^{(l)})_{n=0}^N, l \in \{1, 2, \dots, L'\}$  を新たに生成する。
  - 3: **for**  $l' \in \{1, 2, \dots, L'\}$  **do**
  - 4:   **if**  $\{n \in \{1, 2, \dots, N-1\}; g(s_n^{(l)}) \geq \hat{C}_n(s_n^{(l)})\} \neq \emptyset$  **then**
  - 5:      $\hat{\tau}_0^{*,(l)} \leftarrow \min\{n \in \{1, 2, \dots, N-1\}; g(s_n^{(l)}) \geq \hat{C}_n(s_n^{(l)})\}$
  - 6:   **else if**  $\{n \in \{1, 2, \dots, N-1\}; g(s_n^{(l)}) \geq \hat{C}_n(s_n^{(l)})\} = \emptyset$  **then**
  - 7:      $\hat{\tau}_0^{*,(l)} \leftarrow N$
  - 8: **return**  $\hat{V}_0 = \frac{1}{L'} \sum_{l=1}^{L'} e^{-r\Delta t \hat{\tau}_0^{*,(l)}} g(s_{\hat{\tau}_0^{*,(l)}}^{(l)})$
- 

このようにして得られた下方バイアスを持つ推定価格を Low price と呼ぶ。

## 2.2 ダイナミックヘッジと Greeks

本稿ではトレーダーがオプションを顧客に売却し、顧客が権利行使するリスクをヘッジする設定で議論を行う。オプションをヘッジするとは、オプションを売却することによる損失リスクを軽減することである。具体的には、オプションの権利行使により発生する損失を打ち消すようなポートフォリオ (ヘッジポートフォリオ) を、原資産やその他の金融資産を使用して構築することにより、全体的な損失を抑えることである。そして満期までの間にヘッジポートフォリオ内の金融資産の保有量を何度も変える (リバランスする) ヘッジ手法をダイナミックヘッジという。Black-Scholes モデルでは、原資産と無リスク資産で構成したポートフォリオを連続的にリバランスすることで、ヨーロピアンオプションのリスクを完全にヘッジ可能であり、このようなポートフォリオを複製ポートフォリオという。完備市場では、任意のオプションに対して複製ポートフォリオが必ず存在する。しかし非完備な市場では複製ポートフォリオの存在が保証されていないため、何らかの意味で「最適な」ヘッジ手法を考える必要がある。そこで本稿では Buehler et al. [2019] などの先行研究にない、オプションとヘッジポートフォリオの売買によって最終的に発生する損益である Profit & Loss (PL) を評価指標として採用する。具体的に PL は売却価格とヘッジポートフォリオの収益の和からオプションのペイオフと累積取引コストを引いたものである。

2.1 章では、離散時間市場モデルにおけるアメリカンオプションに焦点を当てたが、以降では、離散時間市場モデルにおけるバミューダンオプションに焦点を当てる。特に、2.1 章で議論した数値計算手法はバミューダンオプションでもそのまま適用できることに注意されたい。はじめに、バミューダンオプションを考えるにあたり、2.1 章で扱った離散時間市場モデルを拡張する。

2.1 章で扱った等間隔な離散時間  $0 = t_0 < t_1 < \dots < t_N = T$  に対して、各々の時点間をさらに  $M \in \mathbb{N}$  個に等分割する。このとき、分割幅  $\Delta t := T/(NM)$  を再定義し、 $t_k := k\Delta t$ ,  $k \in \{0, 1, \dots, NM\}$  を時点とする。すなわち、等間隔な離散時間  $0 = t_0 < t_1 < \dots < t_M < \dots < t_{2M} < \dots < t_{NM} = T$  を想定しており、元々の時点  $0 = t_0, t_M, t_{2M}, \dots, t_{NM} = T$  上で権利可能なバミューダンオプションを考える。言うまでもなく、実際に権利を行使するまでのすべての時点でリバランスすることができる。また、 $M = 1$  のときはアメリカンオプションである。以降、右下の添え字  $k$  は時点  $t_k$  に依存していることを意味する。

フィルトレーション  $\mathbb{F} := (\mathcal{F}_k)_{k=0}^{NM}$  を再定義し、2.1 章と同様に、 $d$  種類のリスク資産は  $\mathbb{R}^d$ -値マルコフ過程  $S = (S_k)_{k=0}^{NM}$  に従うと仮定する。また、無リスク金利は定数  $r \in \mathbb{R}$  とする。

複雑な式を避けるため、時点 0 に割引いた記号を導入する。 $h(S)_k := e^{-r\Delta t k} S_k$ ,  $k \in \{0, 1, \dots, NM\}$  を時点 0 に割引いたリスク資産とする。また、ペイオフ関数  $g: \mathbb{R}^d \rightarrow \mathbb{R}$  に対して、 $\tilde{g}_n(x) := e^{-r\Delta t n M} g(x)$ ,  $x \in \mathbb{R}^d$ ,  $n \in \{0, 1, \dots, N\}$  を時点 0 に割引いたペイオフ関数とする。

ヘッジポートフォリオを構成する資産はオプションの原資産と無リスク資産とする。リスク資産の保有量であるヘッジ戦略は  $\mathbb{R}^d$ -値  $\mathbb{F}$ -適応な確率過程  $\delta = (\delta_k)_{k=0}^{NM}$  に従うと仮定する。ただし、 $\delta_{-1} = \delta_{NM} \equiv 0$  である。 $\lambda: \mathbb{R}_+ \times \mathbb{R} \rightarrow \mathbb{R}_+$  をコスト関数とする。例えば、 $j \in \{1, 2, \dots, d\}$ ,  $k \in \{1, 2, \dots, NM\}$  に対して、 $\lambda(|\delta_k^j - \delta_{k-1}^j|, S_k^j)$  は第  $j$  番目の原資産  $S_k^j$  の保有量を  $\delta_k^j - \delta_{k-1}^j$  だけ変化させた際に発生するコストを意味する。 $n \in \{0, 1, \dots, N\}$ 、オプションの時点 0 における価値  $v \in \mathbb{R}$  に対して、時点  $t_{nM}$  で行使した場合における損益の一つの評価指標 Profit & Loss (PL)  $PL_n(v, \delta, S)$  を次のように定義する：

$$PL_n(v, \delta, S) := v - \tilde{g}_n(S_{nM}) + (\delta \cdot S)_n - \Lambda(\delta, S)_n,$$

$$(\delta \cdot S)_n := \sum_{j=1}^d \sum_{k=0}^{nM-1} \delta_k^j (h(S)_{k+1}^j - h(S)_k^j), \quad (2.2.1)$$

$$\Lambda(\delta, S)_n := \sum_{j=1}^d \sum_{k=0}^{nM} e^{-r\Delta t k} \lambda(|\delta_k^j - \delta_{k-1}^j|, S_k^j). \quad (2.2.2)$$

ここで、 $(\delta \cdot S)_n$  は時点  $t_{nM}$  までの原資産の価格変化とリバランスに伴う割引済み累積収益を意味する。また、 $\Lambda(\delta, S)_n$  は時点  $t_{nM}$  までのポートフォリオ戦略のリバランスに伴う割引済み累積コストを意味する。

ヘッジ資産の保有量に関する重要な概念として Greeks がある。Greeks とは、オプション価格を決定する各種パラメーターの変動に対してオプションの価格がどれだけ変動するかを表す感応度指標である。Greeks には原資産価格に対する価格感応度であるデルタ  $\Delta$ 、 $\Delta$  の原資産価格に対する感応度ガンマ  $\Gamma$  などがある。これらの感応度指標は数学的にはオプション価格を目的のパラメーターで偏微分したものとして定義される。例えば時点  $t_n$  におけるデルタは以下のように定義される： $j \in \{1, \dots, d\}$  に対して、

$$\Delta_n^j := \frac{\partial}{\partial S_n^j} V_n(S_n). \quad (2.2.3)$$

ただし、 $V_n$  は脚注\*4 で言及した関数である。

## 2.3 シミュレーションによるデルタの計算

最もよく知られたヘッジ手法の一つとして原資産の保有量  $\delta$  をデルタにより決定するデルタヘッジが挙げられる。すなわちデルタヘッジでは、式 (2.2.1) の任意の  $\delta_n$  について、

$$\delta_n = \Delta_n \quad (2.3.1)$$

となる。そのためアメリカンオプションのデルタヘッジを行う上で、デルタを計算することは重要である。しかしアメリカンオプションのデルタは計算に最適停止時刻を考慮する必要があることから、解の陽な表現は一般に知られていない。そのため偏微分方程式の差分近似を後退的に計算する有限差分法などを用いて求められる。この手法は計算過程が明瞭であり、実装も簡単であるものの、計算結果がバイアスをもつことや、ペイオフ関数が状態変数である原資産に対してなめらかでないことから、計算がうまくいかない危険性を抱えている。また不偏推定量を計算できる方法として pathwise derivative method や Likelihood ratio method などの方法がある。これらの手法は確率過程に関する情報を用いて、数値微分を厳密微分に置き換えるものであり、ロバストな結果が得られるものの、それぞれペイオフ関数と確率密度の微分が計算可能であることという制約がある。これらの計算手法及び問題点については Glasserman [2004] を参照されたい。

本稿ではシミュレーションにより PL を評価するため、サンプルの全時点でデルタを計算する必要がある。しかし、上記の計算手法では、計算負荷や計算時間の観点から実行は現実的ではない。そこでデルタの計算には LSM 法で計算した継続価値関数の近似  $\hat{C}_n^L(S_n)$  を微分した値を用いた。厳密にはアメリカンオプションのデルタを  $\hat{C}_n^L(S_n)$  の微分により求めることはできない。しかし行使されるまではアメリカンオプションの価値は継続価値と等しく、行使された時点以降ではヘッジを行わないため、デルタを計算する必要がない。そのためアメリカンオプションが行使されるまでのオプション価値を表す継続価値を微分した値をアメリカンオプションのデルタと考え、さらにその推定値として継続価値の近似  $\hat{C}_n^L(S_n)$  を微分した値を使用することとした。

また権利行使価格間のリバランスにおいてもデルタを計算する必要があるため、拡張された市場において LSM 法により継続価値関数の近似  $\hat{C}_k^L(S_k) = \sum_{u=0}^U \hat{a}_{k,u}^L \phi_u(S_k)$  を推定した。行使可能時点においては即時行使価格と継続価値を比較した値に対して最小二乗法推定を行うが、権利行使可能時点間では、直後の権利行使可能時点のオプション価値を割引いた値に対して最小二乗推定を行い、継続価値関数を得る。具体的な計算方法はアルゴリズム 4 である。

---

**アルゴリズム 4** 拡張された市場における LSM 法による継続価値関数の推定
 

---

- 1:  $(S_k)_{k=0}^{NM}$  の  $L$  本のサンプルパス  $(s_k^{(l)})_{k=0}^{NM}$ ,  $l = 1, 2, \dots, L$  を生成する。
- 2:  $\hat{V}_{NM}^{(l)} \leftarrow g(s_{NM}^{(l)})$ ,  $l \in \{1, 2, \dots, L\}$
- 3: **for**  $n \in \{N, N-1, \dots, 2\}$  **do**
- 4:   **for**  $m \in \{1, 2, \dots, M-1\}$  **do**
- 5:      $k \leftarrow nM - m$
- 6:     以下の二乗誤差を最小化するようにパラメータ  $(\hat{a}_{k,u}^L)_{u=0}^U$  を決める:

$$\inf_{\hat{a}_{k,0}^L, \hat{a}_{k,1}^L, \dots, \hat{a}_{k,U}^L} \frac{1}{L} \sum_{l=1}^L \left( e^{-r\Delta t m} \hat{V}_{nM}^{(l)} - \sum_{u=0}^U \hat{a}_{k,u}^L \phi_u(s_k^{(l)}) \right)^2. \quad (2.3.2)$$

- 7:   以下の二乗誤差を最小化するようにパラメータ  $(\hat{a}_{(n-1)M,u}^L)_{u=0}^U$  を決める:

$$\inf_{\hat{a}_{(n-1)M,0}^L, \hat{a}_{(n-1)M,1}^L, \dots, \hat{a}_{(n-1)M,U}^L} \frac{1}{L} \sum_{l=1}^L \left( e^{-r\Delta t M} \hat{V}_{nM}^{(l)} - \sum_{u=0}^U \hat{a}_{(n-1)M,u}^L \phi_u(s_{(n-1)M}^{(l)}) \right)^2. \quad (2.3.3)$$

- 8:    $\hat{V}_{(n-1)M}^{(l)} \leftarrow \max\{g(s_{(n-1)M}^{(l)}), \hat{C}_{(n-1)M}^L(s_{(n-1)M}^{(l)})\}$
- 9: **for**  $m \in \{M-1, M-2, \dots, 1\}$  **do**
- 10:   以下の二乗誤差を最小化するようにパラメータ  $(\hat{a}_{m,u}^L)_{u=0}^U$  を決める:

$$\inf_{\hat{a}_{m,0}^L, \hat{a}_{m,1}^L, \dots, \hat{a}_{m,U}^L} \frac{1}{L} \sum_{l=1}^L \left( e^{-r\Delta t m} \hat{V}_N^{(l)} - \sum_{u=0}^U \hat{a}_{m,u}^L \phi_u(s_m^{(l)}) \right)^2. \quad (2.3.4)$$


---

アルゴリズム 4 では、時点 0 でのアメリカンオプションの価値を具体的な形式で表現できず、時点 0 では原資産価格が全てのパスで同じであり、クロスセクションでの回帰を行うこともできないため、デルタを得ることができない。

そこで LSM 法をベースとして時点 0 のデルタを計算する Modified LSM (MLSM) 法が Wang and Calfisch [2009] により提案された。MLSM 法ではパスの初期値を分散させることで、初期時点においてもオプション価値を近似する関数をクロスセクション回帰で推定可能にしている。MLSM 法の計算アルゴリズムはアルゴリズム 5 である。

アルゴリズム 5 で推定した関数  $\hat{C}_0^Q(S_0, j)$  を微分することでデルタの推定値  $\hat{\Delta}_0^j$  を得る:  $j \in \{1, \dots, d\}$  に対して、

$$\hat{\Delta}_0^j := \frac{\partial}{\partial S_0^j} \hat{C}_0^Q(S_0, j) = \sum_{u=0}^{U_0} \hat{a}_{u,j}^Q \frac{\partial}{\partial S_0^j} \psi_{u,j}(S_0). \quad (2.3.10)$$



---

**アルゴリズム 5** MLSM 法のアルゴリズム
 

---

仮定: 時点  $t_1$  から  $t_{NM-1}$  までの各権利行使可能時点における継続価値関数が推定済み

目的: 各  $j \in 1, \dots, d$  に対して、初期時点における原資産価格が  $S_0$  のオプション価値  $V_0(S_0)$  を  $U_0$  個の基底関数  $(\psi_{u,j})_{u=0}^{U_0}$  の線形和で近似する:

$$V_0(S_0) \approx \sum_{u=0}^{U_0} \hat{a}_{u,j}^Q \psi_{u,j}(S_0) =: \hat{C}_0^Q(S_0, j). \quad (2.3.5)$$


---

- 1: **for**  $j \in \{1, \dots, d\}$  **do**
- 2:  $S_0$  の  $Q$  個のサンプルとして  $(S_0^{(q)})_{q=1}^Q$  を生成する
- 3: **for**  $q \in \{1, \dots, Q\}$  **do**
- 4:  $j$  番目の原資産  $S_0^{j,(q)}$  の値を、それぞれ独立な確率変数  $(\xi^{j,(q)})_{q=1}^Q$  \*5を用いて分散させる。

$$S_0^{j,(q)} \leftarrow S_0^{j,(q)} \cdot \xi^{j,(q)} \quad (2.3.6)$$

- 5:  $q \in \{1, \dots, Q\}$  に対して、 $S_0^{(q)}$  を初期値とするサンプルパス  $(S_k^{(q)})_{k=0}^{NM}$  を生成する。
- 6: 全パスの権利行使時点で即時行使価値と継続価値を計算し、各パスでの最適停止時刻の推定値  $\hat{\tau}_0^{*,(q)}$  を求める:

- 7: **if**  $\{n \in \{1, 2, \dots, N\}; g(S_{nM}^{(q)}) \geq \hat{C}_{nM}^L(S_{nM}^{(q)})\} \neq \emptyset$  **then**

$$\hat{\tau}_0^{*,(q)} \leftarrow \min \{n \in \{1, 2, \dots, N\}; g(S_{nM}^{(q)}) \geq \hat{C}_{nM}^L(S_{nM}^{(q)})\}. \quad (2.3.7)$$

- 8: **else**  $\{n \in \{1, 2, \dots, N\}; g(S_{nM}^{(q)}) \geq \hat{C}_{nM}^L(S_{nM}^{(q)})\} = \emptyset$

$$\hat{\tau}_0^{*,(q)} \leftarrow N. \quad (2.3.8)$$

- 9: 以下の二乗誤差を最小化するようにパラメーター  $(\hat{a}_{u,j}^Q)_{u=0}^{U_0}$  を決める:

$$\inf_{\hat{a}_{0,j}^Q, \dots, \hat{a}_{U_0,j}^Q} \sum_{q=1}^Q \left( g(\hat{\tau}_0^{*,(q)}, S_{\hat{\tau}_0^{*,(q)}}^{(q)}) - \sum_{u=0}^{U_0} \hat{a}_{u,j}^Q \psi_{u,j}(S_0^{(q)}) \right)^2. \quad (2.3.9)$$


---

### 3 Deep Hedging

Deep Hedging とは Buehler et al. [2019] で提案された深層強化学習を用いたヘッジ手法である。Deep Hedging の特徴は、既存の手法ではモデルの解析解や数値計算により導出していたヘッジポジションを、その時点で取得可能な市場情報をインプットとする NN を用いて導出する点である。既存手法に対して Deep Hedging が優れている点は、その自由度の高さである。Deep Hedging では原資産やヘッジ資産のシミュレーションが生成出来れば NN のトレーニングが可能であるため、Black-Scholes モデルや Heston モデルなどの確率微分方程式で記述されるモデルを用いる必要がなく、敵対的生成ネットワーク (GAN) などのマーケット

シミュレーターを利用できる。そして取引コストの存在、リスク保有量や流動性によるトレーディング制約など、完備市場の理論では仮定されていないが、現実のトレーダー達が対応しなければならない仮定を設定した上での最適ヘッジ戦略を計算することが可能である。

本章では、Deep Hedging の設定で重要な概念である凸リスク測度や NN の構造、トレーニングのアルゴリズムなどについて述べる。

### 3.1 凸リスク測度と無差別効用価格

前章で述べたように、非完備市場でのヘッジを考える上では最適なヘッジ戦略を決める指針が必要である。Deep Hedging では PL を評価する指針として、凸リスク測度を採用している。凸リスク測度は以下のように定義される。

**Definition 2** (凸リスク測度).  $\mathcal{X}$  を  $\mathbb{R}$ -値確率変数全体の集合とする。写像  $\rho: \mathcal{X} \rightarrow \mathbb{R}$  が次の条件 (i)–(iii) を満たすとき、 $\rho$  を凸リスク測度と呼ぶ。

- (i) 単調減少:  $X_1 \geq X_2$  a.s. を満たす任意の  $X_1, X_2 \in \mathcal{X}$  に対して、 $\rho(X_1) \leq \rho(X_2)$ 。
- (ii) 凸性: 任意の  $X_1, X_2 \in \mathcal{X}$ 、 $a \in [0, 1]$  に対して、 $\rho(aX_1 + (1-a)X_2) \leq a\rho(X_1) + (1-a)\rho(X_2)$ 。
- (iii) Cash-Invariant: 任意の  $X \in \mathcal{X}$ 、 $c \in \mathbb{R}$  に対して、 $\rho(X+c) = \rho(X) - c$ 。

また、 $X \in \mathcal{X}$  が  $\rho(X) \leq 0$  を満たすとき、 $X \in \mathcal{X}$  は *acceptable* という。

**Remark 3.**  $\rho: \mathcal{X} \rightarrow \mathbb{R}$  を凸リスク測度、 $X \in \mathcal{X}$  とする。 $c \geq \rho(X)$  を満たす任意の  $c \in \mathbb{R}$  に対して、Definition 2 (iii) Cash-Invariant, (i) 単調減少より、 $\rho(X+c) = \rho(X) - c \leq 0$  が成り立つ。特に、 $\rho(X)$  は  $X$  に定数  $c$  を加えた  $X+c$  を *acceptable* にする最小の値である。すなわち、

$$\rho(X) = \inf\{c \in \mathbb{R}; X+c \text{ は } \textit{acceptable}\}.$$

凸リスク測度の例として次が挙げられる。

**Example 4.** 1. CVaR (Conditional Value at Risk, 期待ショートフォール):  $\alpha \in [0, 1]$  をパーセント点とする。

$$\text{CVaR}_\alpha(X) := \frac{1}{1-\alpha} \int_0^{1-\alpha} \text{VaR}_\gamma(X) d\gamma, \quad X \in \mathcal{X}.$$

ただし、

$$\text{VaR}_\gamma(X) := \inf\{m \in \mathbb{R}; \mathbb{P}(X < -m) \leq \gamma\}, \quad X \in \mathcal{X}.$$

リスク測度  $\text{CVaR}_\alpha$  は  $\alpha$  を下回るときの平均損失を意味している。

2. Entropic risk measure :

$$\rho(X) = \frac{1}{\lambda} \log \mathbb{E}[\exp(-\lambda X)], \quad X \in \mathcal{X}.$$

ただし、 $\lambda > 0$  はリスク回避度を表すパラメーターである。

ある条件 (有界性や可積分性など) を有する  $\mathbb{R}^d$ -値  $\mathbb{F}$ -適度な確率過程  $\delta = (\delta_k)_{k=-1}^{NM}$  をヘッジ戦略と呼ぶ。ただし、 $\delta_{-1} = \delta_{NM} \equiv 0$  とする。また、制約のないヘッジ戦略の集合を  $\mathcal{H}$  とする。さらに、 $\mathcal{H}^r$  をある制約が設けられたヘッジ戦略の集合とする (具体的な定義は Buehler et al. [2019] を参照)。非完備市場の設定は

$\delta \in \mathcal{H}$  に対して、取引制約を表す関数を与えることで表現できる。本稿では取引制約がない設定、すなわち  $\mathcal{H}^r = \mathcal{H}$  とする。

**Definition 5** (最適化問題).  $\rho : \mathcal{X} \rightarrow \mathbb{R}$  を凸リスク測度、 $\tau$  を停止時刻とする。このとき、 $\tau$  に関する最適化問題  $\pi_\tau$  を次で定義する：

$$\pi_\tau(X) := \inf_{\delta \in \mathcal{H}^r} \rho(X + (\delta \cdot S)_\tau - \Lambda(\delta, S)_\tau), \quad X \in \mathcal{X}. \quad (3.1.1)$$

ただし、 $(\delta \cdot S)_\tau, \Lambda(\delta, S)_\tau$  は各々、式 (2.2.1), (2.2.2) で定義した。

**Proposition 6** (cf. Proposition 3.1 in Buehler et al.(2019)).  $\rho : \mathcal{X} \rightarrow \mathbb{R}$  を凸リスク測度、 $\tau$  を停止時刻とする。最適化問題  $\pi_\tau : \mathcal{X} \rightarrow \mathbb{R}$  は *Definition 2* (i) 単調減少, (iii) *Cash-invariant* を満たす。また、 $\Lambda(\cdot, S)_0^r, \mathcal{H}^r$  が各々、凸ならば、 $\pi_\tau$  は *Definition 2* (ii) 凸性を満たし、凸リスク測度となる。

(3.1.1) 式を最小化するような  $\delta \in \mathcal{H}^r$  を凸リスク測度  $\rho$  に関する最適なヘッジ戦略と呼ぶ。以降では  $\Lambda(\cdot, S)_\tau$  と  $\mathcal{H}^r$  は凸であると仮定する。本稿の設定では、(3.1.1) 式の  $X$  はオプションのペイオフである。

Black–Scholes モデルや LSM 法で提示されるオプションの初期価格は、リスク中立測度（同値マルチンゲール測度）下での「フェアな」価格である。凸リスク測度という基準においても、以下の条件を満たすように定義される無差別価格（indifference price）として初期価格を求めることができる。

**Definition 7** (無差別価格). ペイオフ  $X$  を持つオプションに対する無差別価格  $p(X) \in \mathbb{R}$  は、 $-X + p_0$  のポジションを取った場合と、何もポジションを取らなかった場合が無差別となる価格、すなわち、 $\pi(-X + p_0) = \pi(0)$  を満たすような価格  $p_0$  と定義される。ここで凸リスク測度  $\pi$  の *cash-Invariant* 性を用いると、 $\pi(-X + p(X)) = \pi(-X) - p(X)$  となるため無差別価格は、

$$p(X) = \pi(-X) - \pi(0), \quad (3.1.2)$$

として計算できる。

また無差別価格ではなく、リスク中立価格  $q_0 \in \mathbb{R}$  があたえられたとき、損失関数  $\ell : \mathbb{R} \rightarrow [0, \infty)$  を用いて以下のように停止時刻  $\tau$  を与えたときの最適化問題を考えることができる：

$$\pi_\tau^\ell(-X) := \inf_{\delta \in \mathcal{H}^r} \mathbb{E}[\ell(-X + q_0 + (\delta \cdot S)_\tau - \Lambda(\delta, S)_\tau)]. \quad (3.1.3)$$

**Example 8.** 式 (3.1.3) の最適化問題でよく使用される損失関数は  $\ell(x) := x^2$  であり、平均二乗誤差（*Mean Square Error, MSE*）という。

## 3.2 ニューラルネットワークの構造

ニューラルネットワーク (NN) とはパラメーター付けられた関数  $\{f^\theta : \theta \in \Theta\}$  である。ここで  $\Theta$  はパラメーター全体の集合である。特に複数の層で線形変換と非線形変換を繰り返す構造を持ったものを Multilayer feed-forward NN (FNN) といい、以下のように定義される：

**Definition 9** (Multilayer feed-forward NN).  $\mathcal{L} \geq 2$  をレイヤー数、 $N_0, N_1, \dots, N_{\mathcal{L}} \in \mathbb{N}$  とする。また、 $\Sigma : \mathbb{R} \rightarrow \mathbb{R}$  を活性化関数とし、 $\ell = 1, \dots, \mathcal{L}$  に対して、 $W_\ell : \mathbb{R}^{N_{\ell-1}} \rightarrow \mathbb{R}^{N_\ell}$  をアフィン関数とする。このと

き、**Multilayer feed-forward NN (FNN)**  $f: \mathbb{R}^{N_0} \rightarrow \mathbb{R}^{N_L}$  を次で定義する：

$$f^\theta(x) := W_L \circ (\Sigma \circ W_{L-1}) \circ \cdots \circ (\Sigma \circ W_1)(x). \quad (3.2.1)$$

ただし、 $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}, b_\ell \in \mathbb{R}^{N_\ell}$  に対して  $W_\ell(x) = A_\ell x + b_\ell$  であり、活性化関数  $\Sigma$  は成分ごとに適用している。従って FNN のパラメーター  $\theta$  は

$$\theta = (A_1, A_2, \dots, A_L, b_1, b_2, \dots, b_L), \quad (3.2.2)$$

である。

NN は Hornik [1991] で示された普遍近似定理 (Universal approximation theorem) により、層の数とノード数を十分に増やすことで、任意の可積分関数を任意の精度で近似できることが保証されており、Deep Hedging ではこの性質を用いて  $\delta$  を NN により近似する。このとき (3.1.1) 式の最適化問題  $\pi(X)$  は、最適な  $\delta$  を見つける問題から、その近似  $\delta^\theta$  を出力する NN のパラメーター  $\theta \in \Theta$  を求める問題となり、以下のように新たに定義できる：

$$\pi_\tau^\theta(X) := \inf_{\theta \in \Theta} \rho(X + (\delta^\theta \cdot S)_\tau - \Lambda(\delta^\theta, S)_\tau). \quad (3.2.3)$$

時点  $t_0$  から  $t_{NM-1}$  の各時点の  $\delta^\theta$  を近似する NN をそれぞれ  $F^{\theta_0}, \dots, F^{\theta_{NM-1}}$  とする。このときこれらの NN は市場情報  $I_k$  (原資産価格:  $S_k$ , ボラティリティ:  $\sigma$  など) を共通のインプットとしてその構造で以下のように分類される。

- **Simple Feed-Forward NN**

Simple Feed-Forward NN は各時点ごとに対応する FNN を用いてヘッジ戦略を計算する。 $\delta_k^\theta$  を計算するには、市場情報  $I_k$  をインプットする：

$$\delta_k^\theta = F^{\theta_k}(I_k). \quad (3.2.4)$$

- **Full-recurrent NN**

Full-recurrent NN は Simple Feed-Forward NN とは異なり、一つの NN を用いて全ての時点におけるヘッジ戦略の計算を行う。すなわち  $\forall k \in \{1, 2, \dots, NM-1\}, \theta_k = \theta_0$  である。またひとつ前の出力結果をインプットデータとして用いることで、NN が循環構造を持つことが特徴である。ヘッジ戦略  $\delta_k^\theta$  を計算するには、市場情報  $I_k$  と満期までの残存期間  $T - t_k$  に加えて、一期前のヘッジ戦略をインプットする：

$$\delta_k^\theta = F^{\theta_0}(I_k, T - t_k, \delta_{k-1}). \quad (3.2.5)$$

- **Semi-recurrent NN**

Semi-recurrent NN は、Simple Feed-Forward NN と同様にヘッジ戦略を計算する NN が時点ごとに異なるが、Full-recurrent NN と同様にインプットに一期前のヘッジ戦略を持つことで、NN 全体で結合関係を持つことが特徴である。ヘッジ戦略  $\delta_k^\theta$  を計算するには、市場情報  $I_k$  と一期前のヘッジ戦略をインプットする：

$$\delta_k^\theta = F^{\theta_k}(I_k, \delta_{k-1}). \quad (3.2.6)$$

Buehler et al. [2019] で提案されているのが、Semi-recurrent NN である。彼らの論文によると、Simple Feed-Forward NN は、取引コストのないケースでは Semi-recurrent NN と同程度の結果を残しているが、取引コストが発生する場合には、Semi-recurrent NN に比べて悪い結果となっており、NN が recurrent 構造を持つことの重要性が示されている。

### 3.3 トレーニングのアルゴリズム

本節では Buehler et al. [2019] に沿ってヨーロッパオプションをヘッジ対象とした場合の Deep Hedging のトレーニングアルゴリズムについて示す。原資産は  $S = (S_n)_{n=0}^N$  であり、オプションの割引済みペイオフは  $g(n, S_n)$  とし、オプションの時点 0 での価格を  $V_0$  とする。また簡単のため、取引コストは発生しないものとする。トレーニングでは  $d$  次元原資産の  $L$  本のサンプルパス  $S^L = ((S_n^1)_{n=0}^N, (S_n^2)_{n=0}^N, \dots, (S_n^L)_{n=0}^N)$  を用いる。ここで任意の  $\ell \in \{1, \dots, L\}, n \in \{0, \dots, N\}$  について  $S_n^\ell = (S_{n,1}^\ell, \dots, S_{n,d}^\ell)$  である。NN は初期パラメータが  $\theta_0^{(0)}, \dots, \theta_{N-1}^{(0)}$  の Semi recurrent NN ( $F^{\theta_0^{(0)}}, \dots, F^{\theta_{N-1}^{(0)}}$ ) を使用し、インプットに用いる市場情報はその時点での割引済み原資産価格を対数変換した値  $\ln(S_n)$  とする。ただし  $\ln(S_n) := (\ln(S_{n,1}), \dots, \ln(S_{n,d}))$  である。そして  $\ell(X) = X^2$  の場合での (3.1.3) 式の最適化問題に対するパラメータの最適化を行う。

具体的な計算アルゴリズムはアルゴリズム 6 である。

## 4 Deep Hedging のアメリカンオプションへの応用

### 4.1 Becker et al. [2020]

Deep Hedging をアメリカンオプションに応用した論文としては Becker et al. [2020] がある。この論文では、アメリカンオプションの価格付けとヘッジの問題に対してそれぞれ NN を応用する手法を提案している。NN を用いた価格付けでは、LSM 法のアルゴリズムにおいて、従来は基底関数の線形和で近似されていた継続価値関数を NN で近似し、この継続価値関数を用いて Low price と High price を求めている。ヘッジの問題においては、Buehler et al. [2019] らと同様にヘッジ戦略を NN で近似している。彼らはアメリカンオプション用の NN のトレーニングの際に、初期時点から満期までの PL を最適化する一つの問題を、継続価値関数を用いることによりアメリカンオプションの権利行使可能時点間ごとのポートフォリオの PL を最適化する複数の問題に分解している。論文の最後にはシミュレーションによって価格付けの精度や、コストがない場合での多次元原資産の Max コールオプションのヘッジ精度を検証している。

彼らの論文におけるヘッジ戦略を計算する NN ( $F^{\theta_0}, \dots, F^{\theta_{NM-1}}$ ) のトレーニング手法はアルゴリズム 7 である。アルゴリズム 7 でヘッジポジションを計算するために使用されている NN は Simple Feed-Forward

NN である。Semi-recurrent NN を使用する場合には  $F^{\theta^{nM}}$  のインプットとして  $F^{\theta^{nM-1}}$  が必要であり、逆伝播によりパラメータを調整するために同時に学習する必要がある。しかし、アルゴリズム 7 に示したように  $F^{\theta^{nM}}$  と  $F^{\theta^{nM-1}}$  のトレーニングは別に行われるために、Semi-recurrent NN を使用することができない。そのため、彼らの提案している手法では取引コストがある場合にヘッジ精度が悪化する可能性がある。そこで、本章では recurrent NN を用いた Deep Hedging をアメリカンオプションに応用する手法について議論する。

---

**アルゴリズム 6** ヨーロピアンオプションに対する Deep Hedging の学習アルゴリズム
 

---

仮定:  $E$ : エポック数

- 1:  $L$  本の原資産のサンプルパスを生成する。
- 2: 時点  $-1$  でのヘッジ戦略  $\delta_{-1} = (\delta_{-1,1}, \dots, \delta_{-1,d})$  を零ベクトル  $\vec{0} \in \mathbb{R}^d$ 、累積ヘッジゲインを  $(\delta^\theta \cdot S)_{-1} = 0$  とする。
- 3: **for**  $e \in \{1, \dots, E\}$  **do**
- 4:   **for**  $n \in \{0, \dots, N-1\}$  **do**
- 5:     時点  $t_n$  におけるヘッジ戦略を NN  $F_n^{\theta^{(e)}}$  を用いて計算する:

$$\begin{bmatrix} \delta_n^{1, \theta^{(e)}} \\ \vdots \\ \delta_n^{L, \theta^{(e)}} \end{bmatrix} = \begin{bmatrix} F_n^{\theta^{(e)}}(\ln(h(S)_n^1), \delta_{n-1}^{1, \theta^{(e)}}) \\ \vdots \\ F_n^{\theta^{(e)}}(\ln(h(S)_n^L), \delta_{n-1}^{L, \theta^{(e)}}) \end{bmatrix}. \quad (3.3.1)$$

- 6:     時点  $n+1$  での原資産価格  $S_{n+1}$  を用いて、ヘッジゲインを計算する:

$$\begin{bmatrix} (\delta^\theta \cdot S)_n^1 \\ \vdots \\ (\delta^\theta \cdot S)_n^L \end{bmatrix} = \begin{bmatrix} (\delta_n^{1, \theta^{(e)}})^\top (h(S)_{n+1}^1 - h(S)_n^1) + (\delta^\theta \cdot S)_{n-1}^1 \\ \vdots \\ (\delta_n^{L, \theta^{(e)}})^\top (h(S)_{n+1}^L - h(S)_n^L) + (\delta^\theta \cdot S)_{n-1}^L \end{bmatrix}. \quad (3.3.2)$$

- 7:     満期でのオプションのペイオフと Profit and Loss  $PL_N$  を計算する:

$$\begin{bmatrix} PL_N^{1, \theta} \\ \vdots \\ PL_N^{L, \theta} \end{bmatrix} = \begin{bmatrix} V_0 - \tilde{g}_N(S_N^1) + (\delta^\theta \cdot S)_{N-1}^1 \\ \vdots \\ V_0 - \tilde{g}_N(S_N^L) + (\delta^\theta \cdot S)_{N-1}^L \end{bmatrix}. \quad (3.3.3)$$

- 8:     損失関数の計算を行う:

$$J(\theta_0^{(e)}, \dots, \theta_{N-1}^{(e)}) := \frac{1}{L} \sum_{l=1}^L (PL_N^{l, \theta})^2. \quad (3.3.4)$$

- 9:     損失関数の勾配  $\nabla J(\theta_0^{(e)}, \dots, \theta_{N-1}^{(e)})$  を計算し、学習率  $\eta_e$  の勾配降下法によりパラメータの更新を行う:

$$\theta^{(e+1)} = \theta^{(e)} - \eta_e \nabla J(\theta_0^{(e)}, \dots, \theta_{N-1}^{(e)}). \quad (3.3.5)$$


---

---

**アルゴリズム 7** Becker et al. [2020] におけるヘッジポジションを計算する NN の学習アルゴリズム
 

---

仮定: 初期時点での推定価格を  $\hat{V}_0$ 、NN で近似した  $n$  回目の権利行使可能時点での継続価値関数を  $c^\theta(n, S_{nM})$  とする。

$n$  回目の権利行使可能時点におけるオプション価値を表す関数を以下で定義する:

$$v^\theta(n, S_{nM}) := \max(\tilde{g}(S_{nM}), c^\theta(n, S_{nM})). \quad (4.1.1)$$

- 1: **for**  $n \in \{1, \dots, N\}$  **do**
- 2:   **if**  $n=1$  **then**
- 3:     以下を満たすパラメータ  $\theta^0, \dots, \theta^{N-1}$  を求める:

$$\min_{\theta^0, \dots, \theta^{N-1} \in \Theta} \mathbb{E} \left[ \left( \hat{V}_0 - v^\theta(1, S_N) + \sum_{k=0}^{N-1} F^{\theta^k}(S_k)(h(S)_{k+1} - h(S)_k) \right)^2 \right]. \quad (4.1.2)$$

- 4:   **else**
- 5:      $C^\theta(n, S_{nM}) := \max(c^\theta(n, S_{nM}), 0)$  とする。
- 6:     以下を満たすパラメータ  $\theta^{(n-1)M}, \dots, \theta^{nM-1}$  を求める:

## 4.2 One-Step Learning Model

本節では One-Step Learning Model について説明する。One-Step Learning Model(1SL Model) とは、従来の Deep Hedging の学習方法を直接アメリカンオプションに適用する方法である。この方法では外生的に与えられた最適停止戦略に基づき、各サンプルパスにおいて最適停止時刻  $\tau^*$  を決定し、 $\tau^*$  での PL を損失関数で評価、パラメーターの更新を行う。すなわち NN  $(F^{\theta_0^A}, \dots, F^{\theta_{N-1}^A})$  を以下の最適化問題でトレーニングする:

$$\inf_{\theta_0^A, \dots, \theta_{N-1}^A \in \Theta} \rho \left( PL_{\tau^*}(V_0^A, \delta^{\theta^A}, S) \right). \quad (4.2.1)$$

このトレーニングにおいては PL を最適停止時刻で評価するため、アルゴリズム 1 で計算した継続価値関数を使用し、最適停止時刻と以下で定義される指示関数  $\mathbb{C}_n$  を計算する:

$$\hat{\tau}^* := \min\{n \in \{1, \dots, N-1\} : \tilde{g}(S_{nM}) \geq \hat{C}_{nM}^L(S_{nM})\}, \quad (4.2.2)$$

$$\mathbb{C}_n := \begin{cases} 0 & \text{if } n > \hat{\tau}^*, \\ 1 & \text{else.} \end{cases} \quad (4.2.3)$$

$$(4.2.4)$$

ただし  $\{n \in \{1, \dots, N-1\} : \tilde{g}(S_{nM}) \geq \hat{C}_{nM}^L(S_{nM})\} = \emptyset$  ならば  $\hat{\tau}^* = N$  である。具体的な学習アルゴリズムはアルゴリズム 8 である。この例では簡単のために取引コストをなしにしている。

仮定:  $\mathbb{E}$ : エポック数

- 1:  $L$  本の原資産のサンプルパスを生成する。
- 2: 時点  $-1$  でのヘッジ戦略  $\delta_{-1} = (\delta_{-1,1}, \dots, \delta_{-1,d})$  を零ベクトル  $\vec{0} \in \mathbb{R}^d$ 、累積ヘッジゲインを  $(\delta^\theta \cdot S)_{-1} = 0$  と定義する。
- 3: **for**  $e \in \{1, \dots, \mathbb{E}\}$  **do**
- 4:      $\mathbb{C}_0 = 1$  とする。
- 5:     **for**  $n \in \{0, \dots, N-1\}$  **do**
- 6:         **for**  $m \in \{0, \dots, M-1\}$  **do**
- 7:              $k = nM + m$
- 8:             時点  $k$  におけるヘッジ戦略を NN  $F^{\theta_k^{(e)}}$  を用いて計算する:

$$\begin{bmatrix} \delta_k^{1, \theta_k^{(e), A}} \\ \vdots \\ \delta_k^{L, \theta_k^{(e), A}} \end{bmatrix} = \begin{bmatrix} F^{\theta_k^{(e), A}}(\ln(h(S)_k^1), \delta_{k-1}^{1, \theta_k^{(e)}}) \\ \vdots \\ F^{\theta_k^{(e), A}}(\ln(h(S)_k^L), \delta_{k-1}^{L, \theta_k^{(e)}}) \end{bmatrix}. \quad (4.2.5)$$

- 9:             時点  $k+1$  での原資産価格  $S_{k+1}$  を用いて、ヘッジゲインを計算する:

$$\begin{bmatrix} (\delta^{\theta, A} \cdot S)_k^1 \\ \vdots \\ (\delta^{\theta, A} \cdot S)_k^L \end{bmatrix} = \begin{bmatrix} (\delta_k^{1, \theta_k^{(e), A}})^\top (h(S)_{k+1}^1 - h(S)_k^1) \mathbb{C}_n^1 + (\delta^{\theta, A} \cdot S)_{k-1}^1 \\ \vdots \\ (\delta_k^{L, \theta_k^{(e), A}})^\top (h(S)_{k+1}^L - h(S)_k^L) \mathbb{C}_n^L + (\delta^{\theta, A} \cdot S)_{k-1}^L \end{bmatrix}. \quad (4.2.6)$$

- 10:             継続価値関数を用いて、 $\mathbb{C}_n$  の値を更新する。
- 11:             各パスでの最適停止時刻  $\tau^*$  でのオプションのペイオフと Profit and Loss  $PL_{\tau^* M}$  を計算する:

$$\begin{bmatrix} PL_{\tau^*, 1M}^{1, \theta^A} \\ \vdots \\ PL_{\tau^*, LM}^{L, \theta^A} \end{bmatrix} = \begin{bmatrix} V_0^A - \tilde{g}_{\tau^*, 1}(S_{\tau^*, 1M}^1) + (\delta^{\theta, A} \cdot S)_{NM-1}^1 \\ \vdots \\ V_0^A - \tilde{g}_{\tau^*, L}(S_{\tau^*, LM}^L) + (\delta^{\theta, A} \cdot S)_{NM-1}^L \end{bmatrix}. \quad (4.2.7)$$

- 12:             損失関数を計算する:

$$J(\theta_0^{(e)}, \dots, \theta_{NM-1}^{(e)}) := \frac{1}{L} \sum_{l=1}^L (PL_{\tau^*, lM}^{l, \theta^A})^2. \quad (4.2.8)$$

- 13:             損失関数の勾配  $\nabla J(\theta_0^{(e)}, \dots, \theta_{NM-1}^{(e)})$  を計算し、学習率  $\eta_e$  の勾配降下法によりパラメータの更新を行う:

$$\theta^{(r+1)} = \theta^{(r)} - \eta_e \nabla J(\theta_0^{(e)}, \dots, \theta_{NM-1}^{(e)}). \quad (4.2.9)$$


---

### 4.3 Two-Steps Learning Model

前節で示したように、アメリカンオプションとヨーロピアンオプションに対する Deep Hedging のトレーニングにおいて大きく異なる点は、権利行使が満期前に行われる可能性がある点である。ヨーロピアンオプション



ンの Deep Hedging では、全てのパスにおいて権利行使機会が固定されているため、損失関数の計算で平均値を取る際に、全ての時点の NN が同じ比率で計算される。しかし、1SL Model の場合にはパスごとに権利行使と PL の評価時点が異なるために、PL の平均値を取ると、初期時点での NN に比べて満期に近いパスの比率が小さくなってしまう。この偏りにより逆伝播でのパラメータの変化量が NN ごとに変わり、効果的なトレーニングができない可能性がある。

そこで、アメリカンオプションの最適なヘッジ戦略が同じペイオフのヨーロッパンオプションでの最適なヘッジ戦略とその補正分の和で表し、前者を従来の Deep Hedging でヨーロッパンオプションに対して最適化された NN、 $NN^E$  で近似して、後者を  $NN^E$  をインプットとする NN、 $NN^{A-E}$  で近似する手法を Two-Steps Learning Model (2SL Model) とする。このとき最適化問題は所与のパラメーター  $\tilde{\theta}^E$  の下での最適なパラメーター  $\theta^{A-E}$  を求める問題となる：

$$\inf_{\theta^{A-E} \in \Theta} \rho \left( PL_{\tau^*}(V_0^A, \delta^{\tilde{\theta}^E} + \delta^{\theta^{A-E}}, S) \right). \quad (4.3.1)$$

$$\tilde{\theta}^E = \arg \inf_{\theta^E \in \Theta} \rho \left( PL_N(V_0^E, \delta^{\theta^E}, S) \right). \quad (4.3.2)$$

以下の命題で (4.3.1) がアメリカンオプションの最適なヘッジ戦略を近似できることを示す。

**Propositon 10.** 任意の  $\tilde{\theta} \in \Theta$  に対して、このパラメーターを持つ  $NN F^{\tilde{\theta}}$  により決定される  $\delta^{\tilde{\theta}}$  を所与とし、

$$\inf_{\theta^{A-E} \in \Theta} \rho \left( PL_{\tau^*}(V_0^A, \delta^{\tilde{\theta}} + \delta^{\theta^{A-E}}, S) \right), \quad (4.3.3)$$

$$(4.3.4)$$

を達成するパラメーターを  $\tilde{\theta}^{A-E}(\tilde{\theta}) \in \Theta$  とする。このとき

$$\inf_{\theta^A \in \Theta} \rho \left( PL_{\tau^*}(V_0^A, \delta^{\theta^A}, S) \right), \quad (4.3.5)$$

を達成するパラメーター  $\tilde{\theta}^A \in \Theta$  について、 $\delta_k^{\tilde{\theta}^A} = \delta_k^{\tilde{\theta}} + \delta_k^{\tilde{\theta}^{A-E}(\tilde{\theta})}$  が成り立つ。

(証明)

$\tilde{\theta} \in \Theta$  を固定する。明らかに

$$\inf_{\theta^A \in \Theta} \rho \left( PL_{\tau^*}(V_0^A, \delta^{\theta^A}, S) \right) \leq \rho \left( PL_{\tau^*}(V_0^A, \delta^{\tilde{\theta}} + \delta^{\tilde{\theta}^{A-E}(\tilde{\theta})}, S) \right),$$

である。一方、任意の  $\hat{\theta} \in \Theta$  に対して、

$$\begin{aligned} \rho \left( PL_{\tau^*}(V_0^A, \delta^{\tilde{\theta}} + \delta^{\tilde{\theta}^{A-E}(\tilde{\theta})}, S) \right) &= \inf_{\theta \in \Theta} \rho \left( PL_{\tau^*}(V_0^A, \delta^{\tilde{\theta}} + \delta^{\theta}, S) \right), \\ &\leq \rho \left( PL_{\tau^*}(V_0^A, \delta^{\tilde{\theta}} - \delta^{\hat{\theta}} + \delta^{\hat{\theta}}, S) \right) = \rho \left( PL_{\tau^*}(V_0^A, \delta^{\hat{\theta}}, S) \right). \end{aligned}$$

よって、 $\hat{\theta}$  は任意だったので、

$$\rho \left( PL_{\tau^*}(V_0^A, \delta^{\tilde{\theta}} + \delta^{\tilde{\theta}^{A-E}(\tilde{\theta})}, S) \right) \leq \inf_{\hat{\theta} \in \Theta} \rho \left( PL_{\tau^*}(V_0^A, \delta^{\hat{\theta}}, S) \right)$$

□

次に 2SL Model の実装について示す。ペイオフが  $g(nM, S_{nM})$  で表わされるアメリカンオプションのヘッジを考える。サンプルパスは  $N$  回の権利行使可能時点と、各権利行使可能時点間での  $M$  回のリバランス時点

において値を取るとする。すなわち  $0 = t_0 < t_1, \dots, t_{NM} = T$  である。まずヨーロッパンオプションの Deep Hedging の NN、 $F^{\theta_0}, F^{\theta_1}, \dots, F^{\theta_{NM-1}}$  を以下の最適化問題についてアルゴリズム 6 でトレーニングする：

$$\inf_{\theta_0, \dots, \theta_{NM-1} \in \Theta} \rho \left( V_0^E - \tilde{g}_N(S_{NM}) + \sum_{j=1}^d \sum_{k=0}^{NM-1} \delta_k^{j, \theta_k} (h(S)_{k+1}^j - h(S)_k^j) \right). \quad (4.3.6)$$

このトレーニングで最適化された NN のパラメーター  $\theta_0 \dots, \theta_{NM-1}$  を  $\tilde{\theta}_0^E, \dots, \tilde{\theta}_{NM-1}^E$  とする。

次にアメリカンオプションの補正分を計算する NN  $F^{\theta^{A-E}}$  を以下の最適化問題でトレーニングする。

$$\inf_{\theta^{A-E} \in \Theta} \rho \left( V_0^A - \tilde{g}_{\tau^*}(S_{\tau^*M}) + \sum_{j=1}^d \sum_{k=0}^{\tau^*M-1} (\delta_k^{j, \tilde{\theta}_k^E} + \delta_k^{j, \theta^{A-E}}) (h(S)_{k+1}^j - h(S)_k^j) \right) \quad (4.3.7)$$

以上をまとめるとアルゴリズム 9 となる。

仮定:  $\mathbb{E}$ : エポック数,

- 1:  $L$  本 の 原資産 の サンプルパス を 生成 する。
- 2: アルゴリズム 6 を 行い、 $\tilde{\theta}_0^E, \dots, \tilde{\theta}_{NM-1}^E$  を 得る。
- 3: 時点  $-1$  で の ヘッジ戦略  $\delta_{-1} = (\delta_{-1,1}, \dots, \delta_{-1,d})$  を 零ベクトル  $\vec{0} \in \mathbb{R}^d$ 、累積ヘッジゲインを  $(\delta^\theta \cdot S)_{-1} = 0$  と 定義 する。
- 4: **for**  $e \in \{1, \dots, \mathbb{E}\}$  **do**
- 5:      $\mathbb{C}_0 = 1$  と する。
- 6:     **for**  $n \in \{0, \dots, N-1\}$  **do**
- 7:         **for**  $m \in \{0, \dots, M-1\}$  **do**
- 8:              $k = nM + m$
- 9:             時点  $k$  に おける ヘッジ戦略 を NN  $F^{\tilde{\theta}_k^E}$  を 用いて 計算 する:

$$\begin{bmatrix} \delta_k^{1, \tilde{\theta}_k^E} \\ \vdots \\ \delta_k^{L, \tilde{\theta}_k^E} \end{bmatrix} = \begin{bmatrix} F^{\tilde{\theta}_k^E}(\ln(h(S_k^1)), \delta_{k-1}^{1, \tilde{\theta}_{k-1}^E}) \\ \vdots \\ F^{\tilde{\theta}_k^E}(\ln(h(S_k^L)), \delta_{k-1}^{L, \tilde{\theta}_{k-1}^E}) \end{bmatrix}. \quad (4.3.8)$$

- 10:      $\delta_{k-1}^{\theta^{(e)}, A-E}, \delta_k^{\tilde{\theta}_k^E}$ 、原資産価格、残存期間をインプットとして  $F^{\theta^{(e)}, A-E}$  で  $\delta_k^{\theta^{(e)}, A-E}$  を 計算 する:

$$\begin{bmatrix} \delta_k^{1, \theta^{(e)}, A-E} \\ \vdots \\ \delta_k^{L, \theta^{(e)}, A-E} \end{bmatrix} = \begin{bmatrix} F^{\theta^{(e)}, A-E}(\delta_{k-1}^{1, \theta^{(e)}, A-E}, \delta_k^{1, \tilde{\theta}_k^E}, \ln(h(S_k^1)), T - t_k) \\ \vdots \\ F^{\theta^{(e)}, A-E}(\delta_{k-1}^{L, \theta^{(e)}, A-E}, \delta_k^{L, \tilde{\theta}_k^E}, \ln(h(S_k^L)), T - t_k) \end{bmatrix}. \quad (4.3.9)$$

- 11:     時点  $k+1$  で の 原資産価格  $S_{k+1}$  を 用いて、ヘッジゲインを 計算 する:

$$\begin{bmatrix} (\delta^{\theta, A} \cdot S)_k^1 \\ \vdots \\ (\delta^{\theta, A} \cdot S)_k^L \end{bmatrix} = \begin{bmatrix} (\delta_k^{1, \tilde{\theta}_k^E} + \delta_k^{1, \theta^{(e)}, A-E})^\top (h(S_{k+1}^1) - h(S_k^1)) \mathbb{C}_n^1 + (\delta^{\theta, A} \cdot S)_{k-1}^1 \\ \vdots \\ (\delta_k^{L, \tilde{\theta}_k^E} + \delta_k^{L, \theta^{(e)}, A-E})^\top (h(S_{k+1}^L) - h(S_k^L)) \mathbb{C}_n^L + (\delta^{\theta, A} \cdot S)_{k-1}^L \end{bmatrix}. \quad (4.3.10)$$

- 12:     継続価値関数を用いて、 $\mathbb{C}_n$  の 値 を 更新 する。
- 13:     各パスでの最適停止時刻  $\tau^*$  で の オプションのペイオフと Profit and Loss  $PL_{\tau^*M}$  を 計算 する:

$$\begin{bmatrix} PL_{\tau^*,1M}^{1, \theta^A} \\ \vdots \\ PL_{\tau^*,LM}^{L, \theta^A} \end{bmatrix} = \begin{bmatrix} V_0^A - \tilde{g}_{\tau^*,1}(S_{\tau^*,1M}^1) + (\delta^{\theta, A} \cdot S)_{NM-1}^1 \\ \vdots \\ V_0^A - \tilde{g}_{\tau^*,L}(S_{\tau^*,LM}^L) + (\delta^{\theta, A} \cdot S)_{NM-1}^L \end{bmatrix}. \quad (4.3.11)$$

- 14:     損失関数を 計算 する:

$$J(\theta^{(e)}, A-E) := \frac{1}{L} \sum_{l=1}^L (PL_{\tau^*,lM}^{l, \theta^A})^2. \quad (4.3.12)$$

- 15:     損失関数の勾配  $\nabla J(\theta^{(e)}, A-E)$  を 計算 し、学習率  $\eta_e^A$  の勾配降下法によりパラメーターの更新を行う:

$$\theta^{(e+1), A-E} = \theta^{(e), A-E} - \eta_e^A \nabla J(\theta^{(e)}, A-E). \quad (4.3.13)$$


---

## 5 検証

### 5.1 シミュレーションの設定

本章では満期までの決められた時点において権利行使が可能なバミューダンオプションを用いてシミュレーションを行った。オプションのペイオフ関数は  $\tilde{g}_n(S_{nM}) = e^{-r\Delta tnM} g(S_{nM})$  であり、 $g: \mathbb{R}^d \rightarrow \mathbb{R}$  は原資産が 1 次元の場合はコールオプションとし、2 次元以上の場合には Max コールオプションとした:

$$d = 1 \text{ の場合 } g(x) := (x - K)^+ \quad (5.1.1)$$

$$d \geq 2 \text{ の場合 } g(x) := \left( \max_{i=1, \dots, d} x_i - K \right)^+, \quad x = (x_1, \dots, x_d) \in \mathbb{R}^d. \quad (5.1.2)$$

原資産は  $d$  次元の連続的な配当のある Black-Scholes モデルとし、実確率測度  $\mathbb{P}$  の同値マルチンゲール測度  $\mathbb{Q}$  の下で確率微分方程式に従うとする:

$$dS_t^i = (r - q)S_t^i dt + \sigma S_t^i dB_t^i, \quad S_0^i = s_0^i, \quad \forall i \in (1, \dots, d), \quad (5.1.3)$$

$$dB_t^i dB_t^j = 0, \quad \forall i, j \in (1, \dots, d). \quad (5.1.4)$$

ここで、 $s_0^i$  は初期時点での原資産価格、 $r$  はリスクフリーレート、 $q$  は配当率を表し、 $B_t^i$  は  $\mathbb{Q}$  の下での  $i$  次元目のブラウン運動である。トレーニング及びテストで使用する原資産のパスはオイラー・丸山法で以下のように離散化し計算した:

$$S_k^i = S_{k-1}^i \exp \left( \left( r - q - \frac{\sigma^2}{2} \right) \Delta t + \sigma z_k^i \sqrt{\Delta t} \right), \quad (5.1.5)$$

$$z_k \stackrel{\text{iid}}{\sim} N(\vec{0}, I), \quad \Delta t = \frac{T}{NM}. \quad (5.1.6)$$

バミューダンオプションは満期  $T$  までの区間  $[0, T]$  で  $T/N$  時点ごとに権利行使を選択できることとし、ヘッジポートフォリオのリバランスはこれらの権利行使可能時点間で  $M$  回のリバランスを行うとする。ヘッジゲインの計算の際には、原資産は時点 0 まで割り引いて計算される。また配当収入は即座に株式に再投資されると仮定する。この場合、割引済みリスク資産過程  $h(S_k)$  は以下のように計算される:

$$h(S_k) = e^{-(r-q)\Delta tk} S_k. \quad (5.1.7)$$

またコストがある場合に、リバランスに伴い発生するコスト関数  $\lambda(|\delta_k^j - \delta_{k-1}^j|, S_k^j)$  は  $\epsilon > 0$  について以下のように設定した:

$$\lambda(|\delta_k^j - \delta_{k-1}^j|, S_k^j) := \epsilon |\delta_k^j - \delta_{k-1}^j| S_k^j. \quad (5.1.8)$$

$\epsilon$  の決め方については 0.01 から 0.001 の幅で、先行研究の論文ごとに様々であった。そこで本研究では  $\epsilon = 0.005$  としている。

最適停止時刻の決定に必要な継続価値関数は  $L = 2 \times 10^5$  とし、LSM 法により推定した。

原資産が 1 次元の場合の継続価値関数の近似は、関数の形式を定数項と 6 次までのベキ級数とした:

$$\hat{C}_k^L(S_k) = \sum_{u=0}^6 a_u(t_k) S_k^u. \quad (5.1.9)$$

2次元以上の場合の継続価値関数の近似は定数項と  $S_i$  の4次までのベキ級数、最大値と2番目に大きな値の積、2番目と3番目の積、...、2番目に小さい値と1番小さい値の積、そして全ての  $S_i$  値の積を基底関数としている。具体的に原資産が3次元 ( $S_k^1, S_k^2, S_k^3$ ) の場合の基底関数は以下ようになる:

$$\begin{aligned} \hat{C}_k^L(S_k) = & a_0(t_k) + \sum_{d=1}^3 \sum_{n=1}^4 a_{4(d-1)+n}(t_k)(S_k^d)^n + a_{13}(t_k)L(S_k, 1) \cdot L(S_k, 2) \\ & + a_{14}(t_k)L(S_k, 2) \cdot L(S_k, 3) + a_{15}(t_k)S_k^1 S_k^2 S_k^3. \end{aligned} \quad (5.1.10)$$

ここで  $L(x, n)$  はベクトル  $x$  の  $n$  番目に大きな値を表す。

また時点0のデルタの導出には  $Q = 1 \times 10^5$  のMLSM法を使用した。MLSM法において、初期値を分散させる  $(\xi^{(q)})_{q=1}^Q$  は Wang and Caffisch [2009] に習い以下のようにした:

$$\xi^{(q)} = e^{0.5\sigma\sqrt{T}\omega^{(q)}}, \quad \omega^{(q)} \sim N(0, 1). \quad (5.1.11)$$

ここで  $\omega^{(q)}$  はそれぞれ独立であり、 $\sigma, T$  はそれぞれ原資産のボラティリティと満期である。時点0でMLSM法により推定する関数の近似も関数の形式を定数項と6次までのベキ級数とした:

$$\hat{C}_0^Q(S_k, j) = \sum_{u=0}^6 a_{u,j}(S_k^j)^u. \quad (5.1.12)$$

Deep Hedging のモデルでは1SL Model と2SL Model を使用した。1SL Model で使用するNNは、Simple Feed Forward NN(FF)、Semi-recurrent NN(Semi)、Full-recurrent NN(Full) の3種類を使用した。NNのインプットには、計算時点での割引済みの原資産価格を対数化した値と、原資産価格の最大値を権利行使価格で割った値を共通で使用し、Semi-recurrent NN と Full-recurrent NN には1期前のヘッジポジションもインプットとして加えている。また Full-recurrent NN には、残存期間の全期間に対する比率もインプットとしている。すなわち、各NNを使用する際に  $\delta_k$  は以下のように計算される。:

$$\delta_k^{FF} = F^{\theta_k}(\ln(h(S_k)), \max_{i=1, \dots, d} (S_k)/K), \quad (5.1.13)$$

$$\delta_k^{Semi} = F^{\theta_k}(\ln(h(S_k)), \max_{i=1, \dots, d} (S_k)/K, \delta_{k-1}^{Semi}), \quad (5.1.14)$$

$$\delta_k^{Full} = F^{\theta}(\ln(h(S_k)), \max_{i=1, \dots, d} (S_k)/K, \delta_{k-1}^{Full}, (T - t_k)/T). \quad (5.1.15)$$

1SL Model のトレーニングにおけるパラメーター更新には Adam(Kingma and Ba [2015]) を用いており学習率は  $\eta$  で固定している。

2SL Model では、 $\delta^E$  の計算と  $\delta^{A-E}$  の計算で2つのNNを使用する。検証において、1つ目のNNはSemi-recurrent NN と Full-recurrent NN の2種類を使用する。このNNは割引済みの原資産価格を対数化した値と1期前のヘッジポジションを共通のインプットとしており、Full-recurrent NN には残存期間の全期間に対する比率もインプットとした:

$$\delta_k^{Semi,E} = F^{\theta_k^E}(\ln(h(S_k)), \delta_{k-1}^{Semi}), \quad (5.1.16)$$

$$\delta_k^{Full,E} = F^{\theta^E}(\ln(h(S_k)), \delta_{k-1}^{Full}, (T - t_k)/T). \quad (5.1.17)$$

2つ目のNNはFull-recurrent NNである。インプットには1期前の  $\delta^{A-E}$  と同時期の  $\delta^E$ 、割引済みの原資産価格を対数化した値、原資産価格の最大値を権利行使価格で割った値、残存期間の全期間に対する比率と

した:

$$\delta_k^{A-E} = F^{\theta^{A-E}}(\delta_{k-1}^{A-E}, \delta_k^E, \ln(h(S_k)), \max_{i=1, \dots, d} (S_k)/K, (T - t_k)/T). \quad (5.1.18)$$

2SL Model のトレーニングにおけるパラメーター更新にも Adam(Kingma and Ba [2015]) を用いており、1 段階目の学習率は  $\eta^E$  で固定し、2 段階の学習では  $\eta^{A-E}$  で固定している。また両学習モデルで活性化関数には  $\mathbf{Relu}(x) := \max(x, 0)$  を使用し、効率的な学習のためにバッチ学習を行っている。

シミュレーションの実装は Python を使用し、Deep Hedging の実装は Python の深層学習ライブラリーである TensorFlow(Abadi et al. [2015]) を使用した。

## 5.2 アメリカンコールオプションでの検証

最初の検証では、原資産が 1 次元のコールオプションで検証を行った。オプションの満期は  $T = 1/4$  とし、 $N = 10, M = 3$  とした。またシミュレーションにおける各種設定は以下の通りである。

- 原資産のパラメーター:  $s_0^i = 100, r = 0.05, q = 0.1, \sigma = 0.2, d = 1$
- 権利行使価格:  $K = 100$
- コスト率:  $\epsilon = 0.005$
- LSM 法及び MLSM 法においてパラメーター推定に使用するパスの数:  $L = Q = 2 \times 10^5$
- $\hat{V}_0$  の low price の推定に使用するパスの数:  $10^5$
- NN のトレーニングに使用するパスの数:  $10^5$
- 評価に使用するパスの数:  $5 \times 10^4$
- 学習率:  $\eta = \eta^E = 0.005, \eta^A = 0.001$ , バッチサイズ: 500
- NN のレイヤー数: 3, NN のニューロン数: 20

表 1 と図 1 は取引コストがない場合での検証結果である。

表 1: 各種統計量 アメリカンコールオプション 取引コストなし

損失関数	MSE						
NN_type	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full	MLSM 法	Unhedged
平均	-0.034	-0.033	-0.034	-0.035	-0.036	0.001	0.000
標準偏差	0.648	0.637	0.636	0.634	0.625	1.182	4.687
-CVaR <sub>0.95</sub>	-1.511	-1.487	-1.426	-1.521	-1.455	-2.808	-10.609
MSE	0.421	0.407	0.406	0.403	0.392	1.398	21.972

損失関数	CVaR <sub>0.95</sub>						
NN_type	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full	MLSM 法	Unhedged
平均	-0.032	-0.036	-0.034	-0.036	-0.037	0.001	0.000
標準偏差	0.763	0.775	0.721	0.703	0.714	1.182	4.687
-CVaR <sub>0.95</sub>	-1.384	-1.447	-1.292	-1.293	-1.295	-2.808	-10.609
MSE	0.583	0.602	0.521	0.495	0.511	1.398	21.972

原資産が 1 次元 Black-Scholes モデルに従うコールオプションのヘッジ結果。各種統計量は各パスでの最適停止時刻における PL を用いて導出。1S は 1SL Model で、2S は 2SL Model で学習した結果。FF は Feed Forward NN、Semi は Semi-recurrent NN、Full は Full-recurrent NN を NN として用いた Deep Hedging の結果。MLSM 法は初期時点でのヘッジポジションを MLSM 法で計算し、それ以外の時点でのヘッジポジションを継続価値関数の微分により求めたモデルでの結果。Unhedged はヘッジをしないモデルであり、初期価格と最適停止時刻での割引済みペイオフの差。損失関数はそれぞれの深層学習モデルで平均二乗誤差 (MSE) と CVaR<sub>0.95</sub> (下 5% 平均損失) を使用。評価における CVaR<sub>0.95</sub> の値は下 5% 平均損失であるため負の値を取っている。

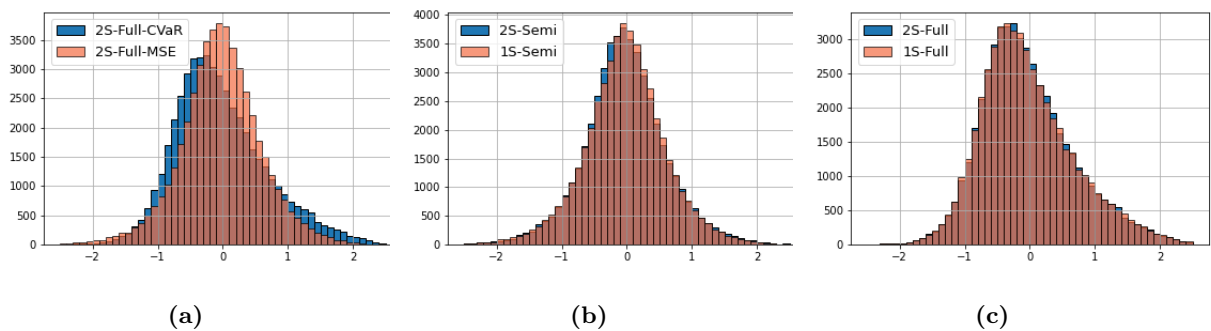


図 1: 取引コストのないアメリカンコールオプションのヘッジ PL。(a) は Full recurrent NN の 2SL Model で損失関数をそれぞれ MSE と CVaR<sub>0.95</sub> にした場合での比較。(b) は損失関数が MSE の場合で Semi recurrent NN の 2SL Model と 1SL Model での比較。(c) は損失関数が CVaR<sub>0.95</sub> の場合で Full recurrent NN の 2SL Model と 1SL Model での比較。

まず表 1 で Deep Hedging と MLSM 法の結果を、ヘッジの精度を評価する各種統計量で比較すると、Deep Hedging のモデルは標準偏差及び MSE の値を小さく抑え、CVaR<sub>0.05</sub> の値が高くなっており、MLSM 法より優れたパフォーマンスを示している。

次に図 1 の (a) から、同じモデルでも損失関数を変えることで PL の分布が明らかに変わっていることが確認できる。MSE を損失関数としているモデルでは、PL の分布が 0 付近を軸に左右対称な形状となっているのに対して、CVaR<sub>0.95</sub> を損失関数としているモデルでの PL の分布は非対称でプラス方向の裾が厚い分布となっている。これらの結果は二乗平均を抑える目的や下 5% の損失を抑える目的など、損失関数の特徴を反映した学習を行えていることを示している。

また表 1 で 1SL Model で異なる NN を用いた結果を、各損失関数で評価した値で比較すると、両方の損失関数のケースで Full recurrent NN を用いたケースが最も優れた結果を残している。そして 2SL Model は 1S-Full に比べて同等、もしくは優れた結果を残している。

次にリバランスに取引コストが発生するケースを検証した。検証結果は表 2 と図 2 である。

表 2: 各種統計量 アメリカンコールオプション 取引コストあり

損失関数	MSE						
	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full	MLSM 法	Unhedged
平均	-0.846	-0.732	-0.722	-0.781	-0.728	-1.258	0.000
標準偏差	0.772	0.781	0.771	0.762	0.779	1.567	4.687
-CVaR <sub>0.95</sub>	-2.563	-2.419	-2.403	-2.430	-2.417	-4.983	-10.609
MSE	1.313	1.147	1.115	1.191	1.137	4.036	21.972

損失関数	CVaR <sub>0.95</sub>						
	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full	MLSM 法	Unhedged
平均	-0.943	-0.841	-0.790	-0.829	-0.818	-1.258	0.000
標準偏差	0.816	0.820	0.832	0.797	0.803	1.567	4.687
-CVaR <sub>0.95</sub>	-2.387	-2.299	-2.231	-2.249	-2.226	-4.983	-10.609
MSE	1.557	1.379	1.317	1.323	1.312	4.036	21.972

原資産が 1 次元 Black-Scholes モデルに従う Call オプションのヘッジ結果。



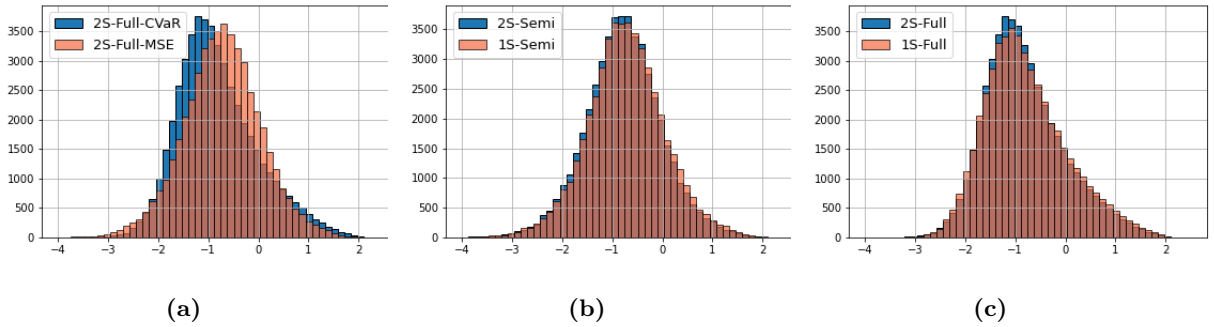


図 2: 取引コストのある場合でのアメリカンコールオプションのヘッジ PL。(a) は Full recurrent NN の 2SL Model で損失関数をそれぞれ MSE と  $\text{CVaR}_{0.05}$  にした場合での比較。(b) は損失関数が MSE の場合で Semi recurrent NN の 2SL Model と 1SL Model での比較。(c) は損失関数が  $\text{CVaR}_{0.05}$  の場合で Full recurrent NN の 2SL Model と 1SL Model での比較。

結果を見ると、累積の取引コストの影響で全てのモデルで PL の分布が負の方向へずれていることが確認できる。Deep Hedging のモデルは MLSM 法に比べて平均が 0 に近く、コストの与える影響を抑えることができています。

また 1SL Model では Feed Forward NN を用いたモデルの結果は他のモデルの結果に比べて、平均値の負の方向への影響が大きく、コストがある場合には再帰構造を持つモデルに比べてパフォーマンスが劣るという予想と一致する結果となった。

このケースでも 2SL Model は 1SL Model と同等以上の結果を残している。

### 5.3 多次元アメリカン MaxCall オプションでの検証

次に原資産の次元数を  $d = 3$  としたケースと  $d = 5$  としたケースでの検証を行った。アメリカン Max コールオプションの満期は  $T = 1/4$  とし、 $N = 10, M = 3$  とした。またシミュレーションにおける各種設定は以下の通りである。

- 原資産のパラメーター:  $s_0^i = 100, r = 0.05, q = 0.1, \sigma = 0.2, i = 1, 2, \dots, d$
- 権利行使価格:  $K = 100$
- コスト率:  $\epsilon = 0.005$
- LSM 法及び MLSM 法においてパラメーター推定に使用するパスの数:  $L = Q = 2 \times 10^5$
- $\hat{V}_0$  の low price の推定に使用するパスの数:  $10^5$
- NN のトレーニングに使用するパスの数:  $10^5$
- 評価に使用するパスの数:  $5 \times 10^4$
- 学習率:  $\eta = 0.005, \eta^A = 0.001$ , バッチサイズ: 500
- NN のレイヤー数: 3, NN のニューロン数:  $12d$

まず  $d = 3$  のケースで検証を行った。表 3 と図 3 は  $d = 3$  で取引コストがないケース、表 4 と図 4 は  $d = 3$  で取引コストがあるケースの検証結果である。

表 3: 各種統計量 3次元 MaxCall オプション 取引コストなし

損失関数	MSE						
NN_type	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full	MLSM 法	Unhedged
平均	-0.014	-0.018	-0.015	-0.007	-0.005	0.083	0.000
標準偏差	1.265	1.289	1.033	1.112	1.027	1.419	6.348
-CVaR <sub>0.95</sub>	-2.772	-2.851	-2.341	-2.442	-2.321	-3.491	-14.220
MSE	1.601	1.663	1.068	1.237	1.054	2.021	40.297

損失関数	CVaR <sub>0.95</sub>						
NN_type	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full	MLSM 法	Unhedged
平均	-0.010	-0.014	-0.018	-0.007	-0.006	0.083	0.000
標準偏差	1.884	1.901	1.326	1.232	1.119	1.419	6.348
-CVaR <sub>0.95</sub>	-3.740	-3.893	-2.592	-2.359	-2.159	-3.491	-14.220
MSE	3.550	3.613	1.760	1.519	1.252	2.021	40.297

原資産が3次元 Black-Scholes モデルに従う MaxCall オプションのヘッジ結果。

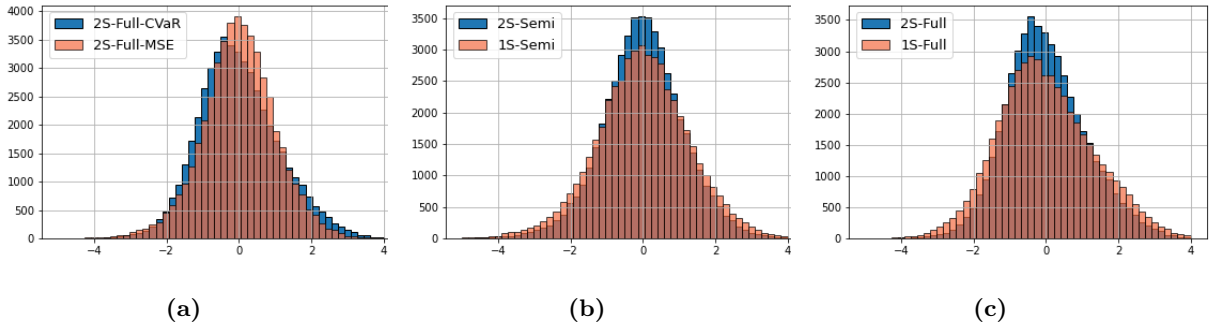


図 3: 取引コストのないアメリカン Max コールオプションのヘッジ PL。(a) は Full recurrent NN の 2SL Model で損失関数をそれぞれ MSE と CVaR<sub>0.95</sub> にした場合での比較。(b) は損失関数が MSE の場合で Semi recurrent NN の 2SL Model と 1SL Model での比較。(c) は損失関数が CVaR<sub>0.95</sub> の場合で Full recurrent NN の 2SL Model と 1SL Model での比較。

表 4: 各種統計量 3次元 MaxCall オプション 取引コストあり

損失関数	MSE					MLSM 法	Unhedged
	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full		
平均	-1.762	-1.440	-1.382	-1.415	-1.380	-2.643	0.000
標準偏差	1.658	1.646	1.616	1.526	1.490	1.953	6.348
$-CVaR_{0.95}$	-5.577	-5.072	-4.977	-4.820	-4.677	-8.099	-14.220
MSE	5.853	4.784	4.521	4.332	4.124	10.796	40.297

NN_type	CVaR <sub>0.95</sub>					損失関数	
	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full	MLSM 法	Unhedged
平均	-2.321	-1.866	-1.718	-1.761	-1.720	-2.643	0.000
標準偏差	1.700	1.636	1.404	1.445	1.329	1.953	6.348
$-CVaR_{0.95}$	-5.691	-5.176	-4.442	-4.513	-4.288	-8.099	-14.220
MSE	8.276	6.160	4.921	5.191	4.726	10.796	40.297

原資産が3次元 Black-Scholes モデルに従う MaxCall オプションのヘッジ結果。

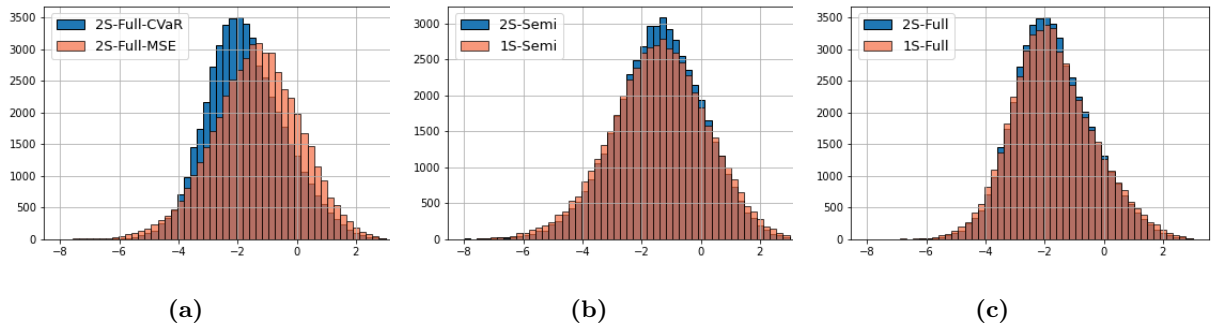


図 4: 取引コストのあるアメリカンコールオプションのヘッジ PL。(a) は 2SL Model で損失関数をそれぞれ MSE と CVaR<sub>0.95</sub> にした場合での比較。(b) は損失関数が MSE の場合で Semi recurrent NN の 2SL Model と 1SL Model での比較。(c) は損失関数が CVaR<sub>0.95</sub> の場合で Full recurrent NN の 2SL Model と 1SL Model での比較。

表 3、表 4 の結果を見ると原資産の次元が増えたことで全モデルで PL の標準偏差が上昇しており、図 3、図 4 を見ると分布の裾が広がっている。これはオプションのリスクの源泉が増えたため、ヘッジが難しくなっていることが原因と考えられる。またコストがある場合の負の方向へのずれも増えているが、これは動かす原資産の数が増えたことにより発生するコストが増えたことに原因があると考えられる。

このケースでも Deep Hedging のモデルは MLSM 法に比べて高いパフォーマンスを示している。また 2SL Model と 1SL Model との比較では、損失関数が MSE の場合には 2S-Full のモデルが、損失関数が CVaR<sub>0.95</sub> の場合には 2S-Semi、2S-Full の両モデルが、1SL Model の中で最もパフォーマンスの良い 1SL-Full のモデ

ルより優れた結果を残している。

最後に  $d = 5$  のケースで検証を行った。表 5 と図 5 は  $d = 5$  で取引コストがないケース、表 6 と図 6 は  $d = 5$  で取引コストがあるケースの検証結果である。

表 5: 各種統計量 5 次元 MaxCall オプション 取引コストなし

損失関数	MSE					MLSM 法	Unhedged
	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full		
平均	0.008	0.008	0.010	0.005	0.005	0.107	0.000
標準偏差	2.181	1.806	1.237	1.281	1.201	1.553	6.463
$-CVaR_{0.95}$	-4.884	-3.933	-2.676	-2.873	-2.711	-3.668	-14.025
MSE	4.755	3.263	1.531	1.641	1.441	2.423	41.767

損失関数	$CVaR_{0.95}$					MLSM 法	Unhedged
	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full		
平均	0.003	-0.002	0.006	-0.003	0.004	0.107	0.000
標準偏差	4.355	4.014	1.875	1.622	1.342	1.553	6.463
$-CVaR_{0.95}$	-8.626	-8.100	-3.708	-3.092	-2.626	-3.668	-14.025
MSE	18.962	16.114	3.514	2.630	1.800	2.423	41.767

原資産が 5 次元 Black-Scholes モデルに従う MaxCall オプションのヘッジ結果。

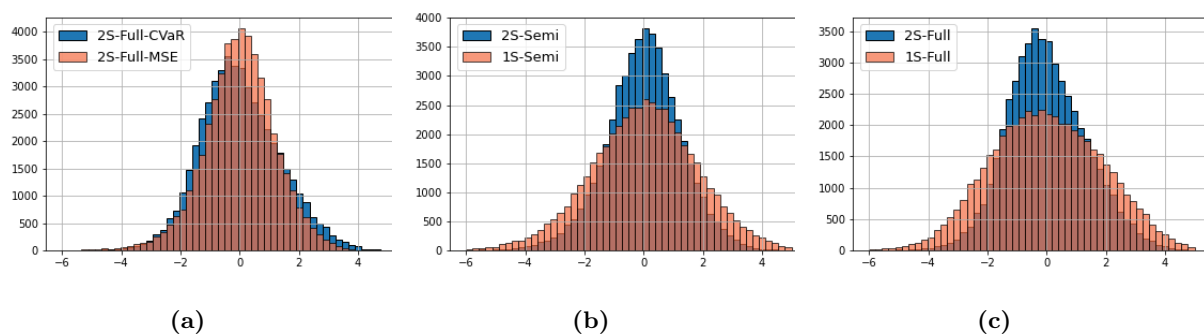


図 5: 取引コストのないアメリカンコールオプションのヘッジ PL。(a) は Full recurrent NN の 2SL Model で損失関数をそれぞれ MSE と  $CVaR_{0.05}$  にした場合での比較。(b) は損失関数が MSE の場合で Semi recurrent NN の 2SL Model と 1SL Model での比較。(c) は損失関数が  $CVaR_{0.05}$  の場合で Full recurrent NN の 2SL Model と 1SL Model での比較。

表 6: 各種統計量 5次元 MaxCall オプション 取引コストあり

損失関数	MSE						
NN_type	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full	MLSM 法	Unhedged
平均	-2.089	-1.710	-1.643	-1.709	-1.686	-3.737	0.000
標準偏差	2.316	2.184	1.946	1.878	1.815	2.111	6.463
-CVaR <sub>0.95</sub>	-7.335	-6.481	-5.847	-5.899	-5.746	-9.364	-14.025
MSE	9.727	7.692	6.487	6.447	6.135	18.423	41.767

損失関数	CVaR <sub>0.95</sub>						
NN_type	1S-FF	1S-Semi	1S-Full	2S-Semi	2S-Full	MLSM 法	Unhedged
平均	-2.605	-1.953	-2.077	-2.194	-2.137	-3.737	0.000
標準偏差	2.652	2.575	1.989	1.733	1.620	2.111	6.463
-CVaR <sub>0.95</sub>	-7.925	-7.285	-6.365	-5.676	-5.361	-9.364	-14.025
MSE	13.820	10.443	8.268	7.815	7.188	18.423	41.767

原資産が5次元 Black-Scholes モデルに従う MaxCall オプションのヘッジ結果。

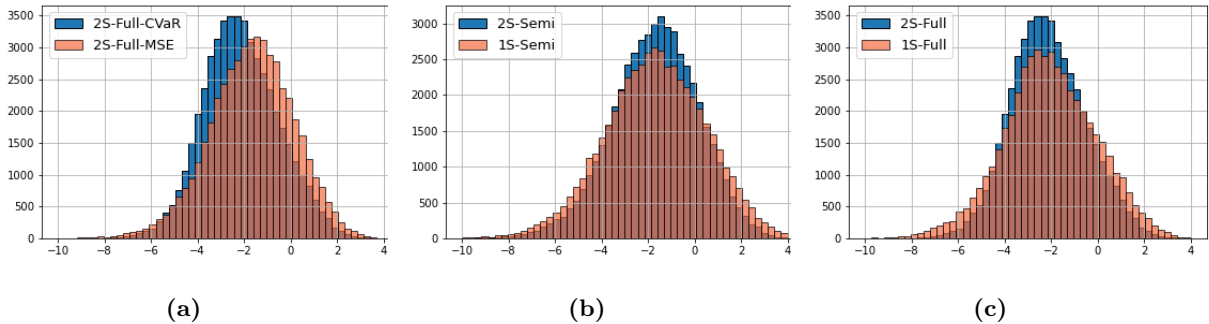


図 6: 取引コストのあるアメリカンコールオプションのヘッジ PL。(a) は Full recurrent NN の 2SL Model で損失関数をそれぞれ MSE と CVaR<sub>0.95</sub> にした場合での比較。(b) は損失関数が MSE の場合で Semi recurrent NN の 2SL Model と 1SL Model での比較。(c) は損失関数が CVaR<sub>0.95</sub> の場合で Full recurrent NN の 2SL Model と 1SL Model での比較。

表 5、表 6 の結果や図 5、図 6 を見ると、 $d = 3$  のケースに比べて、1SL Model と 2SL Model のパフォーマンス差がよりはっきりとしていことがわかる。表 5 をみると 1SL Model は MSE を損失関数としたケースでの CVaR<sub>0.95</sub> の値が、CVaR<sub>0.95</sub> を損失関数としたケースの CVaR<sub>0.95</sub> の値に比べて大きくなっており、特に Feed Forward NN と Semi recurrent NN を用いた 1SL Model のケースでは、MLSM 法にパフォーマンスで劣ってしまっている。この結果から  $d = 5$  のケースでは 1SL Model が十分な学習をできていない可能性が考えられる。2SL Model では、Full recurrent NN を使用したケースで、損失関数ごとに PL の分布の形状や統計量の値に明確に差が出ていることから学習が十分にできていると考えられる。

この検証全体を通して、2SL Model のパフォーマンスは 1SL Model と同等以上のパフォーマンスを達成し、特に原資産が増えたケースで 1SL Model より明らかに優れた結果を残しており、NN の学習法としての 2SL Model に優位性があることが示唆された。

#### 5.4 計算時間とパフォーマンス

次に 1SL Model と 2SL Model の両手法の学習における損失値の収束速度について検証する。2SL Model はヨーロピアンオプションとアメリカンオプションのそれぞれに対して最適化を行うため、同じエポック数であれば 1SL Model の 2 倍程度の計算時間がかかることが予想される。

そこで 2SL Model が 1SL Model に比べて時間効率的な意味でも優れているのか検証した。具体的には各モデルで設定したエポック数の学習を行い、終了までに要した時間と損失関数の減少を分析した。1SL Model ではエポック数を 1000 回、2SL Model ではヨーロピアンオプションとアメリカンオプションのそれぞれに対して 500 エポックずつ、合計で 1000 エポックの学習を行っている。またシミュレーションにおける各種設定は以下の通りである。

- オプションのパラメーター:  $s_0^i = 100$ ,  $r = 0.05$ ,  $q = 0.1$ ,  $\sigma = 0.2$ ,  $d = 3$ ,  $i = 1, 2, \dots, d$
- 権利行使価格:  $K = 100$
- コスト率:  $\epsilon = 0.005$
- $T = 1/4$ ,  $N = 10$ ,  $M = 3$
- LSM 法及び MLSM 法においてパラメーター推定に使用するパスの数:  $L = Q = 2 \times 10^5$
- $\hat{V}_0$  の low price の推定に使用するパスの数:  $10^5$
- NN のトレーニングに使用するパスの数:  $10^5$
- 評価に使用するパスの数:  $5 \times 10^4$
- 学習率:  $\eta = 0.005$ ,  $\eta^A = 0.001$ , バッチサイズ: 500
- NN のレイヤー数: 3, NN のニューロン数: 20
- 損失関数: CVaR<sub>0.05</sub>, 取引コストあり

計算時間については、Python の time パッケージを使用して計測した。結果は図 7 で示した。



図 7: 各モデルで設定したエポック数の学習に要した時間と  $-CVaR_{0.05}$  の値。1S-Semi,1S-Full は 1SL Model で NN が Semi-recurrent NN と Full-recurrent NN としたケースでの結果であり最適停止時刻での PL を損失関数で評価している。2S-full-A は 2SL Model でアメリカンオプションに対して最適化した 2 段階目の学習過程であり、数値は最適停止時刻での PL を損失関数で評価している。2S-full-A は 1 段階目の学習が終了した時点からプロットしている。

1S-Semi と 1S-Full を比較すると、1000 エポック終了に要する時間は 1S-Full のほうが短く、また下 5 % の平均損失も抑えることができている。この原因としては CVaR が損失の下 5 % のパスを評価することから、時点ごとに異なる NN を使用する Semi-recurrent NN ではパスの後半の NN が使われないことにより、学習が難しくなっていることが考えられる。また 2SL Model の結果を見ると、2 段階目の学習が開始してすぐに 1S-Full のパフォーマンスを上回っており、2 段階の学習の合計エポック数が 1SL Model が同じであるにも関わらず計算に要する時間も短く済んでいる。計算時間については、2SL Model の 1 段階目の学習では最適停止時刻の計算が不要であり、1SL Model に比べて同じエポック数の計算に要する時間が短くなることが理由である。以上の結果をふまえると、2SL Model は計算時間効率の観点でも 1SL Model より優れていると考えられる。

## 5.5 リバランスの回数と NN の関係性

最後に、アメリカンオプションの満期及びリバランス回数を変化させ検証を行った。具体的には 2SL Model のヨーロピアンオプションをヘッジする recurrent NN の種類について、リバランス回数が増えた場合でパフォーマンスがどのように変化するかを比較した。

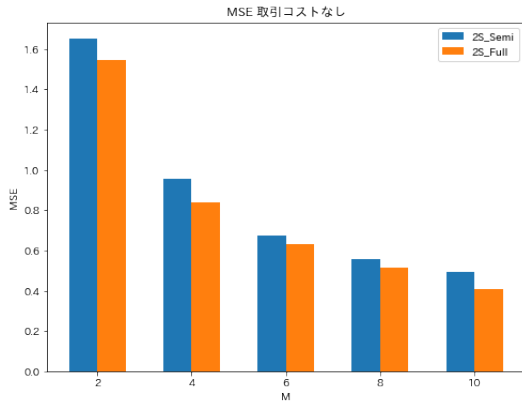
この検証を行う理由は、Semi-recurrent NN の構造上、パフォーマンスとリバランス回数の上昇に負の関係が考えられるためである。Semi-recurrent NN は異なる NN が結合関係を持つという構造から一つの大きな NN であると考えられることができる。そのためリバランスの日数が増えると、NN 全体で層の数が増えることとなる。NN は逆伝播でパラメーターを更新するという性質上、構造がネストするほど学習が難しくなるため、Semi-recurrent NN はリバランスの多い設定では、1 つの NN を使い回す Full-recurrent NN に比べて、パフォーマンスが悪化する可能性がある。

検証では満期を  $T = 1/4$ 、権利行使可能時点数を  $N = 10$  に固定し、権利行使可能時点間でのリバランス回数である  $M$  を変更してパフォーマンスを確認した。オプションの設定は原資産が 3 次元 Black-Scholes モデルに従うアメリカン Max コールオプションとした。またシミュレーションにおける各種設定は以下の通りである。

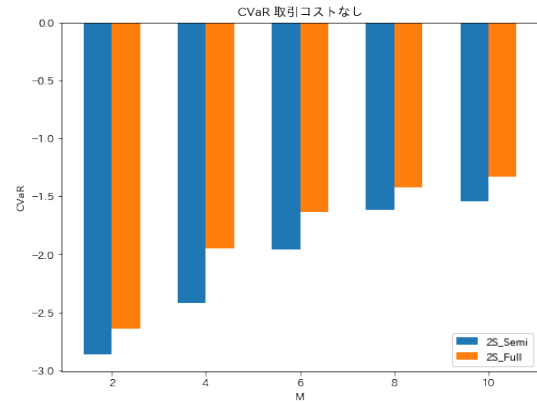
- オプションのパラメーター:  $s_0^i = 100$ ,  $r = 0.05$ ,  $q = 0.1$ ,  $\sigma = 0.2$ ,  $d = 3$ ,  $i = 1, 2, \dots, d$
- 権利行使価格:  $K = 100$
- コスト率:  $\epsilon = 0.005$
- LSM 法及び MLSM 法においてパラメーター推定に使用するパスの数:  $L = Q = 2 \times 10^5$
- $\hat{V}_0$  の low price の推定に使用するパスの数:  $10^5$
- NN のトレーニングに使用するパスの数:  $10^5$
- 評価に使用するパスの数:  $5 \times 10^4$
- 学習率:  $\eta = 0.005$ ,  $\eta^A = 0.001$ , バッチサイズ: 500
- NN のレイヤー数: 3, NN のニューロン数: 20, エポック数: 500
- 損失関数: CVaR<sub>0.05</sub>

取引コストのないケースでの検証の結果は図 8 に示した。





(a) 損失関数が MSE のケースの結果

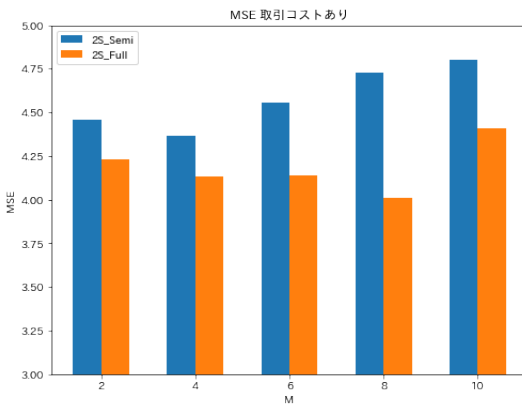


(b) 損失関数が CVaR<sub>0.95</sub> のケースの結果

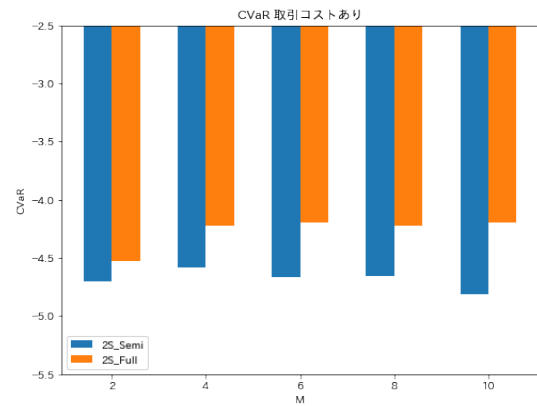
図 8: 取引コストがなくリバランス回数  $M$  を変化させた各ケースにおける損失関数の値。比較したモデルは 2SL Model で Semi-recurrent NN を使用したモデルと Full-recurrent NN を使用したモデルの計 2 パターン。

一般にオプションの満期が同じであれば、期間内でのリバランスの回数が増えるほどヘッジエラー (MSE) は小さくなる。そのため Deep Hedging のモデルにおいても、学習の精度が十分であれば  $M$  の上昇に伴い MSE の値が小さくなるはずである。MSE の検証結果 (a) を見ると、2SL Model の両ケースで  $M$  の値が増えるほどに値が小さくなっていることが確認できる。CVaR<sub>0.95</sub> を損失関数としたケース (b) においても、 $M$  の値が増えるほどに値が 0 に近づいており、下 5% 平均損失を圧縮できていることがわかる。

次に取引コストのあるケースでの検証を行った。結果は図 9 に示した。



(a) 損失関数が MSE のケースの結果



(b) 損失関数が CVaR<sub>0.95</sub> のケースの結果

図 9: 取引コストがありリバランス回数  $M$  を変化させた各ケースにおける損失関数の値。比較したモデルは 2SL Model で Semi-recurrent NN を使用したモデルと Full-recurrent NN を使用したモデルの計 2 パターン。

MSE の検証結果 (a) を見ると、2SL Model で Full-recurrent NN を使用したケースは、 $M = 8$  まで  $M$  の値が増えるほどに値が小さくなっていることが確認できる。 $M = 10$  が悪化している要因としては、第一段階の学習の際に、図 10 のように損失関数の値が途中で大きく上昇してしまったことが原因であると考えられる。

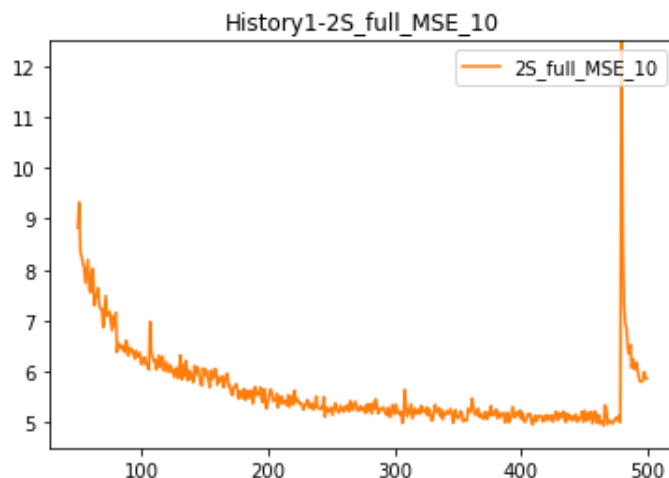


図 10: 2SL Model の Full-recurrent NN で損失関数を MSE とし、 $M = 10$  としたケースでのヨーロッパンオプションに対する学習過程。

しかし Semi-recurrent NN を使用したモデルでは  $M$  が増えるほどに数値が悪化している。また  $M$  を変化した各ケースでのヨーロッパンオプションに対する学習の学習過程である図 11 を見ると、 $M$  が増加しても最終的な収束点が小さくなることなく、逆に上昇しており Semi-recurrent NN の学習精度が十分でない。2SL Model は第一段階の学習精度が第二段階目の学習精度に影響を与えるため、図 8 の結果になっている可能性がある。コストがある場合にヨーロッパンオプションに対するヘッジ精度が悪くなっている理由としては、ヘッジポジションが影響を与える対象が累積ヘッジゲインと累積取引コストに増えているため、複数の NN のパラメーターを同時に動かして最適化することが困難となっているためと考えられる。

また CVaR の検証結果 (b) においても、MSE のケースと同様に Semi-recurrent NN のモデルが  $M$  の値の上昇に伴い、パフォーマンスが下がっている。

結果をまとめると、リバランス回数が増えるほどに Semi-recurrent NN の学習が難しくなるという予想に反して、取引コストがないケースでの検証ではリバランス回数の多いケースでも Semi-recurrent NN は十分な学習精度を達成した。しかしコストがある場合には Semi-recurrent NN は学習が困難となっており、ヘッジポジションの影響を与える対象が増えることが学習精度の向上を妨げている可能性が示唆された。そして両方のケースで十分な学習精度を達成していることから Full-recurrent NN が優れていることが示唆された。

## 6 結論

本稿ではアメリカンオプションに対する Deep Hedging の効果的な学習手法について検討した。

満期の固定されたヨーロッパンオプションを対象とした従来の Deep Hedging の学習方法では、満期前に権利行使可能なアメリカンオプションの特性を学習することができない。そのため、本稿ではまず従来の Deep Hedging の学習方法を直接適用し、最適停止時刻での PL を評価することで、直接アメリカンオプションをヘッジする NN を学習させる One-Step Learning Model (1SL Model) の提案を行った。

しかし 1SL Model では異なる停止時刻の PL を平均値で評価するため、NN の学習に偏りが出てしまう可能性がある。そこで新しい NN の学習方法として、1SL Model のように直接アメリカンオプションをヘッジする NN を学習するのではなく、ヨーロッパンオプションをヘッジする NN とアメリカンオプションの最適

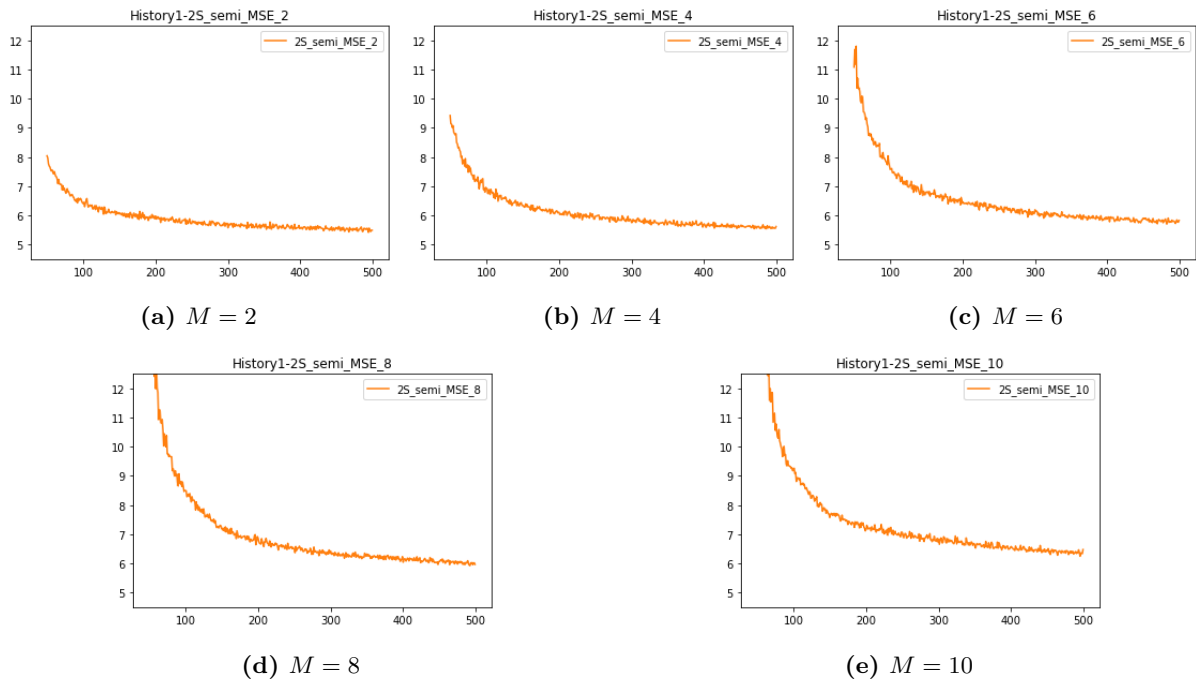


図 11: 2SL Model の Semi-recurrent NN で損失関数を MSE とし、 $M$  を変化させた各ケースでのヨーロッパオプションに対する学習の学習過程。

なヘッジとの差を NN で学習することにより、アメリカンオプションをヘッジする NN を構成する学習方法である Two-Steps Learning Model(2SL Model) の提案を行った。

検証では最初にアメリカンオプションとアメリカン Max コールオプションを対象として、1SL Model と 2SL Model のヘッジパフォーマンスをアメリカンオプションのデルタを計算する既存手法である Modified Least Square Monte Carlo(MLSM 法) のヘッジパフォーマンスと比較した。その結果、提案した 2 つのモデルは多くの場合で MLSM 法に比べて高いパフォーマンスを示した。2 つのモデル間での比較では、2SL Model が 1SL Model と同等以上のパフォーマンスを示した。特にアメリカン Max コールオプションにおける原資産の次元数を増やした場合に 1SL Model が十分に学習ができなかったのに対して、2SL Model はうまく学習をできており、学習方法として 2SL Model の優位性が示された。さらに 2 つのモデルで学習にかかる時間と収束速度の観点でも検証を行い、2SL Model が時間効率の観点でも優れていることを確認した。最後にリバランス回数の増加と取引コストの追加に伴い、Semi-recurrent NN の学習精度が不安定になることを確認した。以上の検証を通じて、2SL Model は 1SL Model に比べて優れた学習手法である可能性が示唆された。

本稿ではヘッジ資産としてアメリカンオプションの原資産のみを扱ったが、実務では市場で取引されているヨーロッパオプションもヘッジ資産とすることで、ガンマやベガといったデルタ以外のリスクをヘッジする手法が存在する。時間的制約上、検証することができなかったが、ヨーロッパオプションのヘッジポジションを学習する 2SL Model はオプションも用いてヘッジする手法でも有効な可能性があるため、今後の研究課題としたい。

## 謝辞

本研究に際しては、東京都立大学大学院経営学研究科 足立高德教授、同竹原浩太准教授、同湯浅智意助教より丁寧かつ熱心なご指導を賜り、深く感謝いたします。また、多くのご指導を賜りました当コースの先生方、多くのディスカッションをさせて頂いた学生の皆様に深く感謝の意を表します。本検証では金融工学研究センターの所有するワークステーションである Turing を数値計算に使用しました。

## 参考文献

- Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng (2015) “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” URL: <https://www.tensorflow.org/>, Software available from tensorflow.org.
- Becker, Sebastian, Patrick Cheridito, and Arnulf Jentzen (2020) “Pricing and Hedging American-Style Options with Deep Learning,” *Journal of Risk and Financial Management*, Vol. 13, No. 7, URL: <https://www.mdpi.com/1911-8074/13/7/158>, DOI: <http://dx.doi.org/10.3390/jrfm13070158>.
- Buehler, Hans, Lukas Gonon, Josef Teichmann, and Ben Wood (2019) “Deep hedging,” *Quantitative Finance*, Vol. 19, No. 8, pp. 1271-1291, URL: <https://doi.org/10.1080/14697688.2019.1571683>, DOI: <http://dx.doi.org/10.1080/14697688.2019.1571683>.
- Detemple, Jrme (2006) *American-style derivatives : valuation and computation / Jerome Detemple.*, Chapman & Hall/CRC financial mathematics series: Taylor& Francis.
- Glasserman, Paul (2004) *Monte Carlo Methods in Financial Engineering*, New York, NY, USA: Springer.
- Hornik, Kurt (1991) “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, Vol. 4, pp. 251-257.
- Kingma, Diederik P. and Jimmy Ba (2015) “Adam: A Method for Stochastic Optimization,” in Bengio, Yoshua and Yann LeCun eds. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, URL: <http://arxiv.org/abs/1412.6980>.
- Longstaff, Francis A. and Eduardo S. Schwartz (2001) “Valuing American options by simulation: a simple least-squares approach,” *The Review of Financial Studies*, Vol. 14, No. 1, pp. 113-147.
- Wang, Yang and Russel E. Caflisch (2009) “Pricing and Hedging American-Style Options: A Simple Simulation-Based Approach,” *Derivatives*.

## 付録 A 一次元アメリカンオプションとデルタのプレミアム分解

アメリカンオプションの価値は、同じペイオフをもつヨーロッパンオプションの価値に早期行使プレミアムを上乗せした形で表現することができる。

**Theorem 11** (Detemple [2006] **Theorem 37**). 原資産が固定された利子と配当のある幾何ブラウンに従うとする:

$$dS_t = (r - q)S_t dt + \sigma S_t dW_t \quad (\text{A.0.1})$$

$t \in [0, T]$  についてアメリカンコールオプション  $V^A(t, S_t)$  は以下の早期行使境界表現を持つ:

$$V^A(t, S_t) = V^E(t, S_t) + \pi(t, S_t, B(\cdot)) \quad (\text{A.0.2})$$

ここで、 $V^E(t, S_t)$  は *Black-Scholes* モデルのヨーロッパンコールオプション価格である。また  $\pi(t, S_t, B(\cdot))$  は以下で与えられる早期行使プレミアムである。

$$\pi(t, S_t, B(\cdot)) = \int_t^T \left( q \cdot S_n e^{-q(s-t)} N(d_+(S_t, B_s, s-t)) - rK e^{-r(s-t)} N(d_-(S_t, B_s, s-t)) \right) ds \quad (\text{A.0.3})$$

$$d_+(S_t, B_s, s-t) = \frac{\log(S_t/B_s) + (r-q + \frac{1}{2}\sigma^2)(s-t)}{\sigma\sqrt{s-t}} \quad (\text{A.0.4})$$

$$d_-(S_t, B_s, s-t) = d_+(S_t, B_s, s-t) - \sigma\sqrt{s-t} \quad (\text{A.0.5})$$

即時行使境界  $B$  は  $t \in [0, T]$  について以下の再帰的非線形積分方程式を満たす:

$$B_t - K = V^E(t, B_t) + \pi(B_t, t, B(\cdot)) \quad (\text{A.0.6})$$

$$B_T = K < B_{T-} \quad (\text{A.0.7})$$

$$B_{T-} = \max(K, (r/q)K) \quad (\text{A.0.8})$$

$$(\text{A.0.9})$$

また早期行使境界表現を用いるとアメリカンオプションのデルタは以下のように表すことができる。

**Proposition 12** (Detemple [2006] **Proposition 38**). *Theorem 11* において、アメリカンオプションのデルタ  $\Delta^A(t, S_t) := \partial V^A(t, S_t) / \partial S$  は  $\mathbb{R}_+ \times [0, T]$  で連続である。そして  $\Delta(t, S_t)$  は以下のように与えられる。

$$\begin{aligned} \Delta^A(t, S_t) &= \Delta^E(t, S_t) + \int_t^T \text{div} \cdot e^{-q(s-t)} N(d_+(S_t, B_s, s-t)) ds \\ &\quad + \int_t^T e^{q(s-t)} n(d_+(S_t, B_s, s-t)) \left( \frac{q \cdot B_s - rK}{B_s} \right) \frac{1}{\sigma\sqrt{s-t}} ds \end{aligned} \quad (\text{A.0.10})$$

ここで  $\Delta(t, S_t) = N(d_+)$  で与えられるヨーロッパンオプションのデルタである。