ディープオプティマル ストッピング

ヴァネッサ・ピザンテ

トロント大学

2019年3月26日（金

最適停止問題

- $X = (X_n)_{n=0}^N$ はRのマルコフ過程である ^{d} $(\Omega, F, (F)_{n=0}^N, P)$. 上で
- T はすべての X-停止時間の集合である。

思い出してほしい：τは$\{\tau = n\} \in F_n, \forall n \in \{0, 1, ..., N\}$のとき、$X$-停止時間である。

- $g : \{0, 1, ..., N\}$ とする。$\times R^d \to R$ は測定可能で可積分な関数である。

目的

$V = \sup_{\tau \in T} E g(\tau, X_\tau)$を求めます。

はじめに

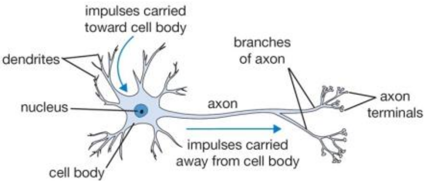
一次資料

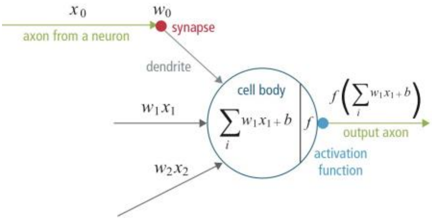
Becker, S., Cheridito, P., and Jentzen, A.：*ディープ・オプティマル・ストッピング*.(2018).

- 有限個の停止時間を持つ最適停止問題は、正確に解くことができる。
  - Snell Envelopeで与えられる最適なV。
- 高次元で数値的に近似することが困難。
- [Becker et al., 2018]は、τを$N+1$の0-1停止判断のシーケンスに分解することを提案している。
- 各判断は、ディープ・フィードフォワード・ニューラルネットワークによって学習されます。

ニューラルネットワークとは？

- ニューラルネットは、人間の脳を大まかにモデル化した計算システムである。ニューロンは、他のニューロンや外部ソースから入力を受け取り、それを使って出力を生成する。
- 各入力は、その相対的な重要性を定量化する重みを持つ。ニ
- ューロンの出力は、その相対的な発火率と考えることができます。



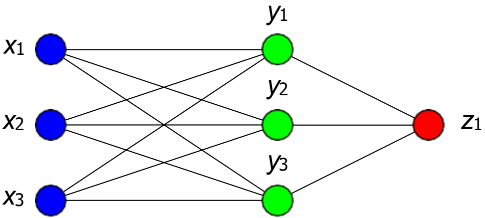


ニューラルネットワークとは？

出典はこちら[ヒジャジら、2015年］。

フィードフォワードニューラルネットワーク

- 入力層：$x = (x_1, x_2, x_3)$.
- **隠れ層(Hidden Layer)**。x を受け取り、$y = (y_1, y_2, y_3)$ を出力する。ここで $y_i = f_y(w_i x + b_i)$ と $w_i = (w_{i1}, w_{i2}, w_{i3})$ で $i = 1, 2, 3$ とする。
- 出力層。y を取り、$z_1 = f_z(w_z y + b_z)$ を出力する。
  $w_z = (w_{z1}, w_{z2}, w_{z3})$



- f_y と f_z は**活性化関数**と呼ばれる。

ニューラルネットワークの学習

- **学習データ**は、入力とターゲットのペア $(x^m, z^m)_{m=1}^{M}$ で構成するだけでなくことができます。
  inputs $(x^m)_{m=1}^{M}$
- ネットワークの学習とは、学習データを用いて**重み** w_i と**バイアス** b_i の項を設定することである。
  - $\theta = \{w1, w2, w3, wz, b1, b2, b3, bz\}$ をモデル**パラメータ**と呼ぶ。
- 学習は、**損失関数** C をθで最小化することによって行われる。
  - 例：平均二乗エラー損失関数

$$C(\theta) = \frac{1}{2M} \sum_{m=1}^{M} \| z^{\theta}(x^m) - z^m \|^2$$

ニューラルネットワークの学習

- 最適なパラメータは、勾配降下アルゴリズムによって損失関数を最小化することによって求められる。
  - 例：学習率ηの更新ステップを持つバニラ勾配降下法。

$$\theta_{t+1} \leftarrow \theta^t - \eta - \nabla_\theta C(\theta^t)$$

- $\nabla_\theta C(\theta^t)$はバックプロパゲーションを用いて計算される。
  - ニューラルネットワークのグラフを効率的に逆行させる方法をそれぞれ計算する。$\frac{\partial C}{\partial \theta_i}$ を連鎖法則で実行します。
  - リバースモード自己微分の特殊なケース。

- 学習後、新たなデータセット（これが**テストデータ**）を用いてアルゴリズムを実行する。

- 注：勾配降下法で損失関数を最小化することは、勾配上昇法で報酬関数を最大化することと等価である。

強化学習(オプション)

問題設定。

- エージェントは、その行動 a_t と状態に応じて、報酬 $r_t = r(s_t, a_t)$ を受け取る。
  s_t .
- 環境は、*未知の*遷移確率 $p(s_{t+1} | s_t, a_t)$ を持つマルコフ決定過程である。

強化する。

- エージェントは政策 $\pi_\theta(s_t, a_t)$ を学習することを目的とし、環境と相互作用して、ある累積報酬関数を最大化するようにする。
- 異なる出目 $\rho = (s_1, a_1, ..., s_T, a_T)$ をサンプリングし、高い報酬に対応する出目がより選ばれやすい方針を得ることができます。

- 政策反復のためのモデルフリーアプローチ。

Q-Learning(オプション)

- Q関数(a.k.a. action-value function)。行動aをとってから方針に従った場合の期待報酬（係数γで割り引かれる）。

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^{i} r_{t+i} \;\middle|\; s_t = s, a_t = a\right]$$

- Q学習は、Qの再帰式に基づく反復アルゴリズムです。は、ベルマンの方程式によって導かれる。

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(r(s_t, a_t) + \gamma \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t)\right)$$

- 注 $r(s_t, a_t) + \gamma \max_a Q(s_t, a) - Q(s_t, a_t)$ はベルマンエラーである。
  =⇒ Q-Learningは、ベルマンエラーを最小化することと等価である。

最適停止とQ-Learning(オプション)

Yu, H.とBertsekas, D.P.。*最小二乗法に基づく最適停止のためのQ-learningアルゴリズム*.(2007).

- 現在の状態 xt が与えられると、2つの選択肢
  - がある。停止してコスト $f(xt)$ を発生させる。
  - 続けて、コスト$h(xt, xt{+}1)$を発生させる。
- $Q^* (xt) = \text{mina } Q(xt, a)$ ここでaは止めるか続けるかのどちらかである。
- 最適な政策は、(Xt)が集合に入ると同時に停止することである。$D = \{xt | f(xt) \leq Q^* (xt)\}$とする。

最適停止とQ-Learning(オプション)

- 変換されたxの線形射影を用いたQの近似値を考えてみましょう。t [Yu and Bertsekas, 2007]。

$$Q^*(x_t) \equiv \varphi(X_t)^t \beta_t$$

- そして、対応する投影ベルマン誤差は、以下のようにQ-learningアルゴリズムに基づく最小二乗近似によってβ$_t$を最小化する。

$$\beta_{t+1} = \beta_t + \alpha(\hat{\beta}_{t+1} - \beta_t)$$

どこ

$$\hat{\beta}_t = \arg\min_{\beta} \sum_{k=0}^{t} \left(\varphi(x_k)^t \beta - h(x_k, x_{k+1}) - \gamma \min\{f(x_{k+1}), \varphi(x_k)^t \beta_t\}\right)^2$$

最小二乗モンテカルロ法(LSMC)

Longstaff, F. and Schwartz E.: *Valuing American Options by Simulation: シンプルな最小二乗法によるアプローチ*, (2001).

- M個の株価経路をシミュレートする $(x_n^m)_{n=0}^{N}$.
- $V_N^m = g(N, x_N^m)$, $\forall m$ とし、$n = N - 1, ..., 0$ とする。
  - $\hat{q}_n(x) = \sum_{k=0}^{K} \beta_k x^k$ による近似的な継続値 q_n。ここで、βは最小化する。

  最小二乗誤差

$$\sum_{m=1}^{M} \left(e^{-rt_n} V_{n+1}^m - q_n(x_n^m)\right)^2$$

  - Set $V_n^m = \begin{cases} g(n, x_n^m) & \text{if } q_n(x_n^m) \leq g(n, x_n^m) \text{ である。} \\ e^{-r\frac{T}{N}} V_{n+1}^m & \text{それ以外の場合} \end{cases}$

f

最小二乗ニューラルネットワークによるアプローチ

Kohler, Krzyzak and Todorovic: *Pricing of High-Dimensional American Options by Neural Networks*, (2006).

- Longstaff-Schwartzアルゴリズムにニューラルネットワークを適用し、より高次元のアメリカンオプションの価格を算出。
- 最小二乗回帰をニューラルネットワークに置き換えた形式。

$$q^{\theta n} = a_2^{\theta n} \circ \sigma \circ a_1^{\theta n}$$

  どこ

  - $a_1 : \mathbb{R}^d \to \mathbb{R}^K$, $a_2 : \mathbb{R}^K \to \mathbb{R}$: $a_i(x) = W_i x + b_i$ and $\sum_{k=0}^{K} |W_{2,k}| \le C_n$.
  - $\sigma : \mathbb{R}^k \to [0,1]^K : \sigma_k(y_k) = 1/(1+e^{-y_k})$.
- アルゴリズムの後方では、学習データ（$x_n^m, e^{-rtn} - V_{n+1}$）を再帰的に) を有し作成する。

  平均二乗損失関数

エクスタンションより複雑なニューラルネットワーク

Hu, R.: *Deep Learning for Ranking Response Surfaces with Applications to Optimal Stopping Problems*, (2019).

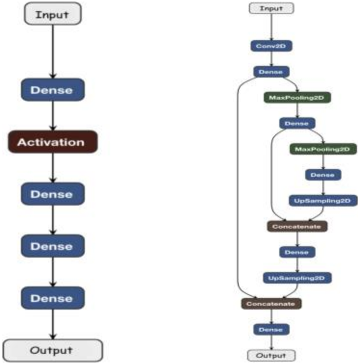


図1：左：フィードフォワードNN、[Becker et al., 2018]で適用されたNNと同様、右：フィードフォワードNN、[Becker et al:UNET、[Hu, 2019]で適用されている。

出典はこちら[フー、2019年】。]

レスポンスサーフェスランキング

- 最適停止問題は、表面ランク付けによる画像分類問題として再解釈することができる。
- 曲面ランキング問題：各入力xに対して最小曲面のインデックスを抽出する。
を作成し、クラスラベルとして扱います。
  - 最適な停止ラベルは、（各タイムステップで）停止または継続するだけです。
- より計算効率の良い**コンボリューションNN**（UNETなど）を**使用**できるようになった[Hu, 2019].
- デメリット完全畳み込み型NNの収束理論はまだ完全には開発されていない。

停止時間を0-1の連続した判断で表現すること

目標

$(X_n)_{n=0}^{N}$、$\{f_n(X_n)\}_{n=0}^{N}$ に従って最適な停止判断を示すことができる。

ここで、f_n。$\mathbb{R}^d \to \{0, 1\}$とする。

- 時間Nの停止時間τ_n を$\{f_k(X_k)\}_{k=n}^{N}$の関数として書けばよい。
- T_n をすべての X -停止時間 τ s.t. $n \leqq \tau \leqq N$ の集合とする。
- 明らかに $T_N = \{N\}$ なので、$\tau_N = N - f_N(X_N)$ ここで $f_N \equiv 1$ とする。
- さて、$n = N - 1, ..., 0$ に対して、次のように定義することができる。

$$\tau_n = \sum_{k=n}^{N} k f_k(X_k) \prod_{j=n}^{k-1} (1 - f_j(X_j)) \in T_n \tag{1}$$

定理1

$n \in \{0, \ldots, N-1\}$ に対して、$\tau_{n+1} \in T_{n+1}$ は、次のような形式とする。

$$\tau_{n+1} = \sum_{k=n+1}^{N} k f_k(X_k) \prod_{j=n+1}^{k-1} (1 - f_j(X_j)) \tag{2}$$

そのとき、測定可能な関数 f_n が存在する。$\mathbb{R}^d \to \{0, 1\}$ このような $\tau_n \in T_n$ 満たす

$$\mathrm{E}g(\tau_n, X_{\tau_N}) \geqq V_n - (V_{n+1} - \mathrm{E}g(\tau_{n+1}, X_{\tau_{n+1}})) \tag{3}$$

ここで、V_n と V_{n+1} は以下の条件を満たす。

$$V_n = \sup_{\tau \in T_n} \mathrm{E}g(\tau, X_\tau) \tag{4}$$

定理1の証明のスケッチ

- 停止時間τ$\in T_n$ を固定し、$E = V_{n+1}$ - Eg (τ$_{n+1}$, $X_{\tau_{n+1}}$) とする
- 。g (τ$_{n+1}$, $X_{\tau_{n+1}}$)は(2)を用いて測定可能であることを示せ。
  $\Longrightarrow$ $\exists$ h_n meas. とマルコフ的な s.t. $h_n(X_n)$ = E[g (τ$_{n+1}$, $X_{\tau_{n+1}}$)|X_n]である
  。
- $D = \{g(n, X_n) \geqq h_n(X_n)\}$、$E = \{\tau = n\} \in F_n$ と定義する。
  - $\tau_n = nI_D +$ τ$_{n+1}$ $I_{D^c} \in T_n$
  - $\tau\sim =$ τ$_{n+1}$ $I_E +$ τ $I_{E^c} \in T_{n+1}$
- したがって、Eg (τ, $X_{\tau_{n+1}}$) ≧ Eg (τ˜, $X_{\tilde{\tau}}$) - E $\Longrightarrow$ E[g (τ, $X_{\tau_{n+1}}$)I_{E^c}] ≧ E[g (τ, X_τ)I_{E^c}] - E となる。
- *練習問題です。*Eg (τ$_n$, X_{τ_n}) ≧ Eg (τ, X_τ) - E を示せ。
- τは任意であるから、Eg (τ$_n$, X_{τ_n}) ≧ V_n - E となり、(3)を満たす
-

定理1の証明のスケッチ

。ここでτが$_n$(1)を満たすことを示せば証明は完了する。

私たちは何を証明したのでしょうか？

- 定理1により、(1)からのτ_n は、V_n を計算するのに十分使用できることが示される。
  - τnは時間n最適停止時間であることを意味する。
  - τnは後方再帰的に計算されるため、(3)に従います。

- $V = \sup_{\tau \in T} \mathrm{E}g(\tau, X_\tau)$に対応する最適な停止時間は

$$\tau = \overset{N}{\underset{n=1}{}} n f_n(X_n) \overset{n-1}{\underset{k=0}{}} (1 - f_k(X_k))$$

- ...しかし、どうやって数列 $\{f_n\}_{n=0}^{N}$ を見つけるのだろうか？

ニューラルネットワークの導入

- $f_N \equiv 1$ を用いて、$n = N - 1, ..., 0$ について f_n を近似する一連のニューラルネットワーク $f^{\theta_n}: \mathrm{R}^d \rightarrow \{0, 1\}$ を構築する。
- そして、T_{n+1} を経由して近似することができる。

$$\underset{k=n}{\overset{N}{}} k - f^{\theta_k}(X_k)\ \underset{j=n}{\overset{k-1}{}} (1 - f^{\theta_j}(X_j)) \qquad (5)$$

問題です。

- f^{θ_n} は2値出力 $\Longrightarrow$ θに対して連続でない。
- θ_n を勾配に基づく最適化アルゴリズムで学習するための連続出力が必要。

連続出力を持つニューラルネットワーク

$F^\theta : \mathbb{R}^d \to (0,1)$という形の2層式フィードフォワードニューラルネットワークを導入する。

$$F^\theta = \psi \circ a_3^\theta \circ \varphi_{q_2} \circ a_2^\theta \circ \varphi_{q_1} \circ a_1^\theta \tag{6}$$

どこ

- q_1 と q_2 は隠れ層のノード数である。
- $a_1^\theta : \mathbb{R}^d \to \mathbb{R}^{q_1}$, $a_2^\theta : \mathbb{R}^{q_1} \to \mathbb{R}^{q_2}$, $a_3^\theta : \mathbb{R}^{q_2} \to \mathbb{R}$ を満たす。

$$a_i^\theta(x) = W_i x + b_i$$

- $\varphi_{q_i} : \mathbb{R}^{q_i} \to \mathbb{R}^{q_i}$ はReLU活性化関数、$\varphi_q(x_1, ..., x_q) = (x_1^+, ..., x_{q_i}^+)$ である。
- $\psi : \mathbb{R} \to \mathbb{R}$ はロジスティック・シグモイド関数で、$\psi(x) = 1/(1 + e^{-x})$ です。

バイナリーデシジョンに戻る

- パラメータは、$\theta = \{(A_i, b_i)\}_{i=1}^{3} \in \mathbb{R}^q$ である。, ここで $q = q_1(d + q_2 + 1) + 2q_2 + 1$.
- F^{θ} を使って、$n = N - 1, ..., 0$ に対して勾配降下法で最適 θ_n を求めることができる。その後、$f^{\theta_n} : \mathbb{R}^d \to \{0, 1\}$を使って計算できる。

$$f^{\theta_n} = I_{[0,\infty)} \circ a^{\theta_n}_3 \circ \varphi_{q_2} \circ a^{\theta_n}_2 \circ \varphi_{q_1} \circ a^{\theta_n}_1 \qquad (7)$$

- 注 [Becker et al., 2018] は、F^{θ} を使ってθを最適化すると、f^{θ} の最適なパラメータが得られるという正式な証明を提供していない。
- しかし、$F^{\theta_n}(X_n)$ を、時間 n において停止が最適な決定である確率（X_n が与えられた場合）と考えると、直感的に理解できるようになる。

リワード機能の選択

主な内容

将来期待される報酬を最大化するような、時間nにおける停止判定をもたらす報酬関数を定義したい。

もし、時刻nに私たちが

- 停止 $\Longrightarrow f_n(X_n) = 1$ でペイオフ $g(n, X_n)$ 続き、最適に進んだ
- 後 $\Longrightarrow f_n(X_n) = 0$ で最終的にペイオフ $g(\tau_{n+1}, X_{\tau_{n+1}})$ を受け取ることになります。

そこで、時間nにおける報酬関数が、以下のように近似されるようにしたい。

$$\sup_{f \in D} \mathrm{E}[g(n, X_n) f(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1) f(X_n))] \text{ とする。} \tag{8}$$

ここで、D はすべての f の集合である。$\mathrm{R}^d \to \{0, 1\}$の可測集合である。

- ...しかし、(8)のfをニューラルネットワークf^{θ}に置き換えることもできるのだろうか？

定理2

$n \in \{0, ..., N-1\}$とし、停止時間$\theta_{n+1} \in T_{n+1}$を固定すると、$\forall E > 0$、以下のような$q_1, q_2 \in \mathbb{N}^+$が存在する。

$$\sup_{\theta \in \mathbb{R}^q} \mathrm{E}[g(n, X_n) f^{\theta}(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - f^{\theta}(X_n))]$$ となります。

$$\geq \sup_{f \in D} \mathrm{E}[g(n, X_n) f(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - f(X_n))] - E \tag{9}$$

ここで、D はすべての測定可能な関数 f の集合である。$\mathbb{R}^d \to \{0, 1\}$である。

コラール1

任意の$E > 0$に対して、$q_1, q_2 \in \mathbb{N}^+$ といった形のニューラルネットワーク関数が存在する。

(7) $f^{\theta_N} \equiv 1$ となるような、停止時間

$$\hat{\tau} = \sum_{n=1}^{N} n f^{\theta_n}(X_n) \prod_{k=0}^{n-1} (1 - f^{\theta_k}(X_k))\text{である。} \tag{10}$$

は、$\mathrm{Eg}(\hat{\tau}, X_{\hat{\tau}}) \geq \sup_{\tau \in T} \mathrm{Eg}(\tau, X_\tau) - E$ を満たす。

- つまり、最適な停止判断のシーケンスを近似的に求めることができるのです $\{f_n\}_{n=0}^N$ を、最適化されたニューラルネットワークのシーケンス $\{f^{\theta_n}\}_{n=0}^N$ で表現する。

定理2の証明のスケッチ（オプション）

- gの可積分性により、$\exists \tilde{f} : \mathbb{R}^d \to \{0, 1\}$ meas.t.となる。

$$E[g(n, X_n)\tilde{f}\theta(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - \tilde{f}\theta(X_n))] \text{ とする。}$$
$$\geq \sup_{f \in D} E[g(n, X_n)f(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1)f(X_n))] - E/4 \quad (11)$$

- $\tilde{f} = I_A$, $A = \{x \in \mathbb{R}^d \mid \tilde{f} = 1\}$とし、以下のことに注意する。

$B \to E[|g(n, X_n)| I_B(X_n)]$ および $B \to E[|g(\tau_{n+1}, X_{\tau_{n+1}})| I_B(X_n)]$は$\mathbb{R}^d$ 上の有限ボアレス測度である。

- だから$\exists K \subseteq A$ compact s.t.である。

$$E[g(n, X_n)I_K(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_K(X_n))]\text{である。}$$
$$\geqq E[g(n, X_n)\tilde{f}\theta(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - \tilde{f}\theta(X_n))] \text{ とする。} - E/4 \quad (12)$$

定理2のスケッチ 証明の続き(オプション)

- $\rho_K(x) = \inf_{y \in K} ¥x - y\|_2$ とし、$k_j(x) = \max\{1 - j \cdot \rho_K(x), -1\}$ に注意すること。
  $j \in \mathbb{N}$ は点状に $I_K - I_{K^c}$ に収束する。だからDCTによって

$$\mathrm{E}[g(n, X_n) I_{k_j(X_n) \geq 0} + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_{k_j(X_n) \geq 0})] \text{である。}$$
$$\geqq \mathrm{E}[g(n, X_n) I_K(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_K(X_n))] - E/4 \qquad (13)$$

- k_j はコンパクトセット上で一様に近似できるため、**[Leshno et al., 1993]**による。
  by $h(x) = \sum_{i=1}^{r} (v_i^T x + c_i)^+ - \sum_{i=1}^{s} (w_i^T x + d_i)^+$ となり，このようになります。

$$\mathrm{E}[g(n, X_n) I_{h(x_n) \geq 0} + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_{h(x_n) \geq 0})] \text{。}$$
$$\geqq \mathrm{E}[g(n, X_n) I_{k_j(X_n) \geq 0} + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_{k_j(X_n) \geq 0})] - E/4 \qquad (14)$$

- $I_{[0,\infty)} \circ h$ をf^θ の形のNNと考えると、eq(14)は次のようになる。

$$\mathrm{E}[g(n, X_n) f^\theta(X_n) + g(\tau_{n+1}, X_\tau)$$

定理2のスケッチ 証明の続き(オプション)

$n+1$

$)(1 - f^{\theta}(X_n))]$ となります。

$$\geqq \mathrm{E}[g(n, X_n) I_{k_j(X_n)\geqq 0} + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_{k_j(X_n)\geqq 0})] - E/4 \qquad (15)$$

- (11)、(12)、(13)、(15)を組み合わせると、必要な(9)が得られる。

ニューラルネットワークによる推定値の算出

- マルコフ過程のM個の独立した経路をシミュレートする、$(x_n^m)_{n=0}^N$.
- θ_N は、$f^{\theta_N} \equiv 1$ となり、$n = N - 1, ..., 0$ となるように選ばれます。

$\theta_{n+1}, ..., \theta_N$ を使って、m 個の経路のそれぞれに沿った τ_{n+1}^m を、以下の方法で計算する。

$$\tau_{n+1}^m = \sum_{k=n+1}^{N} k f^{\theta_k}(x_n^m) \prod_{j=n+1}^{k-1} (1 - f^{\theta_j}(x_j^m))$$

経路mに沿った時間nにおいて、$F^{\theta_n}(x^m)$ を停止し、その後、以下を遵守する場合 $\{f^{\theta_k}(x_k^m)\}_{k=n+1}^N$ であり、実現報酬は

$$r_n^m(\theta_n) = g(n, x_n^m)F^{\theta_m}(x_n) + g(\tau_{n+1}^m, x_{\tau_{n+1}^m}^m)$$

- 十分に大きなMに対して

$$\frac{1}{M}\sum_{m=1}^{M} r_n^m(\theta_n) \quad (16)$$

ニューラルネットワークによる推定値の算出

は、$\mathrm{E}[g(n, X_n)F^{\theta_n}(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - F^{\theta_n}(X_n))]$ に近似します

。

ニューラルネットワークによる推定値の算出

- (16)は勾配アルゴリズムによってθnを最大化したい報酬関数として働くことに注意。
- 次に、$F^{\theta_n}(x^m_n)$ を0-1停止判定、$f^{\theta_n}(x^m_n)$ に変換する。

- それでは、テスト用サンプルパスを生成してみましょう $(y^m_n)^N_{n=0}$ を$m=1, \ldots, M$とする。
- $n = N - 1, \ldots, 0$ について、学習で見つけた $\{\theta_n\}$ を使って、次のように計算する。

$$\tilde{\tau}^{mn}_{+1} = \sum_{k=n+1}^{N} k f^{\theta_k}(y^m_k) \prod_{j=n+1}^{k-1} (1 - f^{\theta_j}(y^m_j))$$

m個のサンプルパスのそれぞれに沿って

ニューラルネットワークによる推定値の算出

- 全体最適停止時間（(10)から$\hat{\tau}$を推定）。

$$\tilde{\tau}^m = \sum_{n=1}^{N} n f^{\theta_n}(y_n^m) \prod_{k=1}^{n-1} (1 - f^{\theta_k}(y_k^m))$$

- $V = g(\hat{\tau}, X_{\hat{\tau}})$ のモンテカルロ推定値に対応する。

$$\hat{V} = \frac{1}{M} \sum_{m=1}^{M} g(\tilde{\tau}^m, y^m_{\tilde{\tau}^m}) \tag{17}$$

- CLTにより、Vの$1-\alpha$, $\alpha \in (0, 1)$の信頼区間は次のようになる。

$$\left[\hat{V} - z_{\alpha/2} \frac{\hat{\sigma}}{\sqrt{M}},\ \hat{V} + z_{\alpha/2} \frac{\hat{\sigma}}{\sqrt{M}} \right]$$

ここで、$z_{\alpha/2}$ は $N(0, 1)$ の $1 - \alpha/2$ 分位数であり

$$\hat{\sigma} = \sqrt{\frac{1}{M-1} \sum_{m=1}^{M} \left(g(\tilde{\tau}^m, y^m_{\tilde{\tau}^m}) - \hat{V} \right)^2}$$

アプリケーションバミューダンマックスコールオプション

例

権利行使価格Kで時刻Tに満期を迎えるバミューダンマックスコールオプションをdに書き込んだ。
資産、X^1、...、X^d、$N+1$の等距離行使時刻を持つ。
$t_n = nT/N, n = 0, 1, ..., N.$

- ペイオフ関数時間tは：$\left(\max_{i\in\{1,...,d\}} X^i_t - K\right)^+$.
- したがって、時刻tにおけるその価格はsupτである。 E（e-rτ（$\left(\max_{i\in\{1,...,d\}} X^i_\tau - K\right)^+$.
- これを最適停止問題 supτ∈T Eg (τ, X_τ) としてとらえ、ここで

$g(n, x) = e^{-rt}$

アプリケーションバミューダンマックスコールオプション

$$\text{マックス}_{i\in\{1,\cdots,d\}} \left(x^i - K \right)^+ \quad (18)$$

$i\in\{1,\cdots,d\}$とする。

アプリケーションバミューダンマックスコールオプション

- 相関のない*d個の*資産からなるブラック・ショールズ市場モデルを仮定する。

- $i = 1, \ldots, d$ の場合、設定します。

$$x_0^i = 90, \quad K = 100, \quad \sigma_i = 0.2, \quad \delta_i = 0.1, \quad r = 0.05, \quad T = 3, \quad N = 9$$

- 資産価格パスは、以下の方法でシミュレーションできます。

$$x_{n,i}^m = x_{0,i} \cdot \exp\left(\sum_{k=0}^{n} (r - \delta_i - \sigma_i^2/2)\Delta t + \sigma_i \sqrt{\Delta t} \cdot Z_{k,i}^m\right) \tag{19}$$

ここで、$\Delta t = T/N$ および $Z_{k,i}^m \sim N(0, 1)$.

PyTorchによるニューラルネットの構築

(6)からF^{θ}という形のニューラルネットワークを構築するのは、Pytorchを使えば簡単です!

```
nnとしてtorch.nnをインポートする

class NeuralNet(torch. nn.
    Module): def   init (self, d,
    q1, q2):
        super(NeuralNet, self).      _()
        self. a1 = nn.Linear(d, q1)
        self. relu = nn.ReLU()
        self. a2 = nn. リニア(q1, q2)
        self. a3 = nn. リニア(q2, 1)
        self. sigmoid=nn.シグモイド()

    def forward(self, x):
        out = self. a1(x)
        out = self. relu(out)
        out = self. a2(out)
        out = self. relu(out)
        out = self. a3(out)
        out = self. sigmoid(out)

        引揚げる
```

Pytorchでのネットワークの学習

```
def loss(y_pred,s, x, n, tau):
    r_n=torch.zeros((s.M))
      for m in range(0,s.M):
         r_n[m]=-s.g(n,m,x)*y_pred[m] - s.g(tau[m],m,x)*(1-y_pred[m]) return(r_nの
     平均())

def NN(n,x,s, tau_n_plus_1): epochs=50
     model=NeuralNet(s.d,s.d+40,s.d+40)
     optimizer = torch.optim.Adam(model.parameters(), lr = 0.0001)

     for epoch in range(epochs):F
         =     model.forward(X[n])
         optimizer.zero_grad()
         criterion = loss(F,S,X,n,tau_n_plus_1) criterion.backward()
         。
         オプティマイザーのステップ()

     return F,model
```

- s.gはg、式(18)で定義される。

適応的モーメント推定（Adam）最適化

入力：$\boldsymbol{\alpha}$、r（θ）、θ^0

Set $\beta_1 = 0.9$; $\beta_2 = 0.99$; $E = 10^{-8}$

$m_0 = v_0 = 0$; $t = 0$

while $r(\theta_t)$ *not minimized* **do**

$t = t + 1$

$G_t = \nabla_\theta r(\theta_{t-1})$

$m_t = \beta_1 - m_{t-1} + (1 - \beta_1) - G_t$

$v_t = \beta_2 - v_{t-1} + \beta_2 - (G_t)^2$

$\hat{m}_t = m_t/(1 - (\beta_1)^t)$

$\hat{v}_t = v_t/(1 - (\beta_2)^t)$

$\theta_t = \theta_{t-1} - \alpha - \hat{m}_t/(\sqrt{\hat{v}_t} + E)$

出力：$\boldsymbol{\theta}^t$

- 適応型。頻出する特徴量に対して、より小さな更新を行う（学習率を下げる）。
- モーメント推定。勾配の平均 (m_t) と非心分散 (v_t) の指数関数的に減衰する平均を格納します。

結果

d	実績	見積もり	標準誤差	95%信頼区間
2	8.04	7.50	0.23	(7.05, 7.95)
5	16.64	16.80	0.32	(16.18,17.43)
10	26.20	23.83	0.29	(23.25, 24.40)

表 1: V の推定値を表にしてみた。$M = 5000$ サンプルパスで計算した結果。

注）d＝2の実測値は[Hu, 2019]、d＝5, 10は[Becker et al, 2018]による。

フラクショナルブラウン運動の最適な停止方法（オプション）

- ハーストパラメータHを持つフラクショナルブラウン運動は、平均0、共分散構造を持つガウス過程である

$$\mathrm{E}[W_t^H W_s^H] = \frac{1}{2}\left(t^{2H} + s^{2H} - |t-s|^{2H}\right) \tag{20}$$

- 標準ブラウン運動は、H=1/2の分数ブラウン運動である。

目的

$\sup_{0\leqq\tau\leqq 1} \mathrm{E}W_\tau^H$ を評価したい。

FBMの最適停止への挑戦（オプション）

任意停止定理 (OST)

$(X_t)_{t\geq0}$ をマルチンゲール、τをフィルトレーション $(F_t)_{t\geq0}$ に関する停止時間とすると、以下のいずれかが成立する場合である。

- τは有界
- $X_{\tau\wedge t}$ は有界

ということは、$EX_\tau = EX_0$ となる。

- したがって、OSTによって、任意の停止時間τ s.t. $W^{1/2}_{\tau\wedge t}$ に対してEW$^{1/2}_{\tau}$ =0 は上界されます。
を定数で指定します。
- しかし、$H \neq 1/2$の場合、W^H 、マーチンゲールではないので、OSTは適用されない。

フラクショナルブラウン運動の最適な停止方法 (オプション)

- 時間間隔 [0, 1] を時間ステップに離散化し、$t_n = \frac{n}{100}$.
- 100次元のマルコフ過程、$(X_n)_{n=0}^{100}$ を作成し、$(W^H_{t_n})_{n=0}^{100}$ を記述します。

$$x_0 = (0, 0, ..., 0)$$

$$x_1 = (w^H_{t_1}, 0, ..., 0)$$

$$x_2 = (w^H_{t_1}, w^H_{t_2}, ..., 0)$$

$$\vdots$$

$$x_{100} = (w^H_{t_{100}}, w^H_{t_{99}}, ..., w^H_{t_1})$$

- g とする。$\mathrm{R}^{100} \to \mathrm{R}$ を $g(x_1, ..., x_{100}) = x_1$ とすると、次のように計算することを目指します。

$$\sup_{\tau \in T} \mathrm{E}g(X_\tau) \tag{21}$$

ここで、τは全てのX停止時刻の集合である。

フラクショナルブラウン運動の最適な停止方法 (オプション)

- [Becker et al, 2018] $Y^H = W^H - W$ を定義することによって、$(X_n$ ティーエヌワン にとって $)^{100}_{n=0}$ をシミュレートした。H を持つ定常ガウス過程を形成する。
自己共分散

$$E[Y_n^H Y_{n+k}^H] = \frac{|k+1|^{2H} - |k|^{2H} + |k-1|^{2H}}{2(100^{2H})}$$
=(エ)

- 次に、(X_n) のサンプル経路は、$W^H = \sum_{k=1}^{n}$ を介して計算される。Y_k^H.

- 次に、$d = 100$、$q_1 = 110$、(7)の形のニューラルネットを学習します。$q_2 = 55$ とし、モンテカルロ法で(21)を近似する。

フラクショナルブラウン運動の最適な停止方法 (オプション)

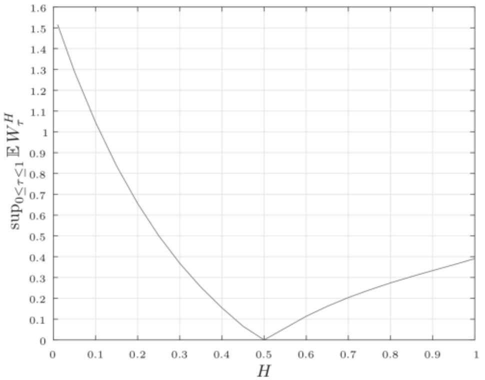


図2：【Becker et al., 2018】の結果。最適に停止したFBMの期待値そのハーストパラメーターHをw.r.t.

出典はこちら[ベッカーら、2018年]。

まとめ

- ニューラルネットワーク（および機械学習アルゴリズム全般）は、高次元での最適停止問題の解法を支援することができる。
- 新しい研究領域です。
  - Becker et al., 2018]が採用したNeural Networkは、最もシンプルなアーキテクチャの一つである。
  - より複雑なアーキテクチャでは、完全に開発された収束理論がありません[Hu, 2019]。

さらなる研究のための質問。

- このアルゴリズムでは、隠れ層あたりのノード数やエポック数を戦略的
- に選択することができますか？
- 他に試す価値のありそうなニューラルネットワークアーキテクチャは？

数理ファイナンスの問題で、ニューラルネットワークを応用して高次元で近似できるものは他にあるのだろうか？

参考文献 48

Becker, S., Cheridito, P., and Jentzen, A. (2018).ディープ・オプティマム・ストッピング
テクニカルレポートです。

Hijazi, S., Kumar, R., and Rowen, C. (2015).
画像認識のための畳み込みニューラルネットワークの使用。

Hu, R. (2019).
応答曲面をランク付けするためのディープラーニングと最適停止問題への応用。
arXiv e-prints、ページarXiv:1901.03478。

キングマ，D.P.，バ，J.（2014）.
アダム：確率的最適化のための手法。
CoRR。

Kohler, M., Krzyz˙ak, A., and Todorovic, N. (2010).
ニューラルネットワークによる高次元アメリカン・オプションの価格決定.
数理ファイナンス, 20(3):383-410.

参考文献 49

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993).
非多項式活性化関数を持つ多層フィードフォワードネットワークは、あらゆる関数を近似することができる。
Neural Networks, 6:861-867.

Yu, H. and Bertsekas, D. P. (2007).
最小二乗法に基づく最適停止のためのQ-learningアルゴリズム。
2007 European Control Conference (ECC), pages 2368-2375.