

# 深層学習によるファイナンスの新展開

## ディープ・ヘッジングの紹介

ファイナンス・ワークショップ  
2022/11/11

日本銀行 金融研究所 経済ファイナンス研究課  
篠崎 裕司

本セミナーで示される意見は、筆者個人に属し、日本銀行の公式見解を示すものではない。  
また、ありうべき誤りはすべて筆者個人に属する。本資料はファイナンスワークショップ参加者限り。  
(照会先：[yuuji.shinozaki@boj.or.jp](mailto:yuuji.shinozaki@boj.or.jp))

## 背景

- ✓ 深層学習の活用：画像・自然言語処理から、強化学習と組み合わせた制御問題へ
- ✓ ファイナンス、特に金融商品の価格付け・リスク管理における活用  
→ 直近5年程度で、実務家・研究者の双方から注目度高まる

## 本日の話題：ディープ・ヘッジングを中心とした、先端研究の紹介

- ✓ 「損失リスク最小化問題」を数値的に解く枠組み
- ✓ データに基づき、現実制約を加味したヘッジ戦略を導出  
→ 定量化困難だった取引コスト等の影響を、分析可能に  
ヘッジの適用範囲拡大・自動化へ
- *Deep hedging*. Buehler, Gonon, Teichmann, Wood著 (QF, 2019, 現在被引用数219)  
— Risk.netのQuant of The year 2022受賞  
外資系金融機関ではデリバティブ実務で活用も  
学会/主要学術雑誌においても当該テーマに関するものが急増

## § 1. 金融実務におけるヘッジの問題

「損失リスク最小化問題」を定式化、当該問題を解決することの意義を整理

## § 2. ディープ・ヘッジングの枠組み

深層学習の適用方法を概説、実務上の利点・留意点や現行手法との違いを整理

## § 3. 研究の潮流

深層学習のファイナンスへの応用研究を概説

## お伝えしたいこと

ファイナンスのどんな問題を、深層学習によりどう解決可能で、どんな応用が展望されるか



# 損失リスク最小化問題の定式化

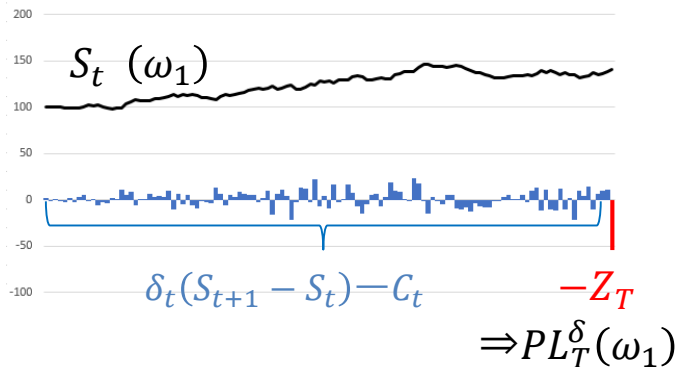
$$\underbrace{PL_T^\delta}_{\text{満期}T\text{までの合計損益}} = \underbrace{-Z_T}_{\text{ヘッジ対象}} + \sum_{t=0}^{T-1} \left( \underbrace{\delta_t}_{\text{保有量}} \underbrace{(S_{t+1} - S_t)}_{\text{価格変化}} - \underbrace{C(\delta_{t+1} - \delta_t)}_{\text{取引コスト}} \right)$$

ヘッジ手段から生じる損益

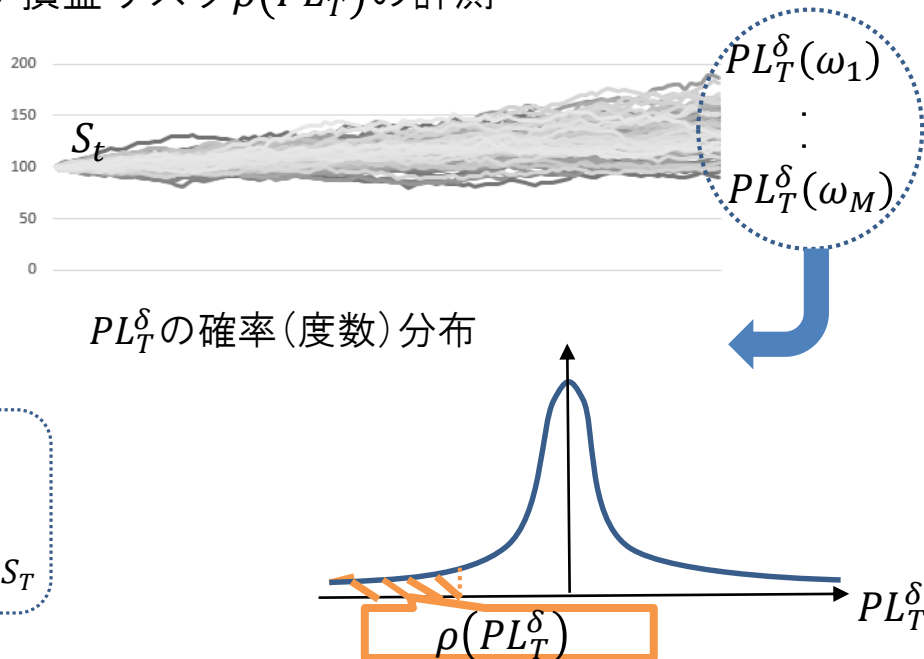
損失リスク  $\rho(PL_T^\delta)$  を最小化するように、ヘッジ戦略  $\delta_t (t = 1, \dots, T-1)$  を決定する問題

期待ショートフォール等

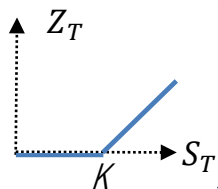
## ① 損益 $PL_T^\delta(\omega_1)$ の計測



## ② 損益リスク $\rho(PL_T^\delta)$ の計測



例.  $Z_T = \max(S_T - K, 0)$  ;  
 $C(\delta_{t+1} - \delta_t) = A \times |\delta_{t+1} - \delta_t|$   
 $\rho(PL) = E[PL \mid PL \leq \alpha]$





## リスク中立評価に基づくヘッジの問題点

6

現実の制約をモデル化できず、誤差が生じる

→ 完全には複製を達成できず、損失発生リスクあり

- ✓ 理想的な金融市場（複製可能であるための条件）  
取引コスト等の市場摩擦や貸出/借入金利の差がなく、任意の時刻に任意の量を取引可能
- ✓ リスク中立評価に基づくヘッジ：実行方法（付録3, 4）
  - (1) ヘッジ対象の現在価値計算  
複製に必要なコストを、現在価値とする（複製/無裁定の原理、リスク中立評価）  
— 市場で流動性の高い商品の価格に、理論モデルをキャリブレーションし、  
自社が保有する複雑な商品の価格を計算（補間）
  - (2) ヘッジ手段(原資産)等の価格変動に応じて、保有量を調整
- 理論の仮定が不成立 → 完全には、複製を達成できない  
⇒ リスク中立価格に、追加でコスト・チャージするのが通例  
(XVAと呼ばれる様々な価格調整も)

## 深層学習がもたらした進展

### 損失リスク最小化問題を直接解く数値的手法

- ✓ 損失リスク最小化問題：「非完備市場の理論」の研究 [Leland, 1985] [Hodges, Neuberger, 1989]  
→ その数値計算は困難で、実用化が容易ではなかった
- 漸近挙動や特殊な設定下での数値計算が議論されてきた
- ✓ 深層学習により、数値計算を実行（ヘッジ戦略を算出）し、その妥当性が確認された  
— 様々な設定で、理論的妥当性、及び、実務的有用性が多数報告されている
  - ✓ 損失リスク最小化問題が、数値的に解ける  
→ デリバティブ分野：リスク中立評価を経由しないヘッジが可能、前スライドの問題点解消へ  
（リスク尺度を導入し、複製できないコストを見積もる計算を可能に）  
— ALM等制約条件がより複雑な問題への適用も



§ 1. 金融実務におけるヘッジの問題

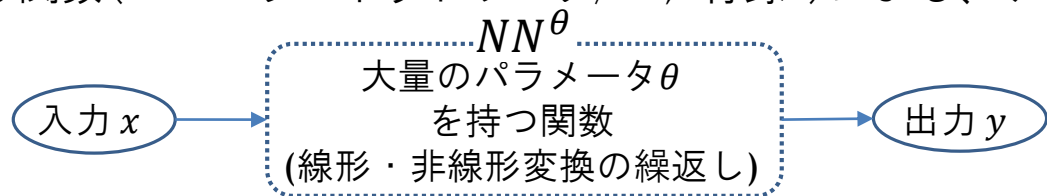
§ 2. ディープ・ヘッジングの枠組み

§ 3. 研究の潮流

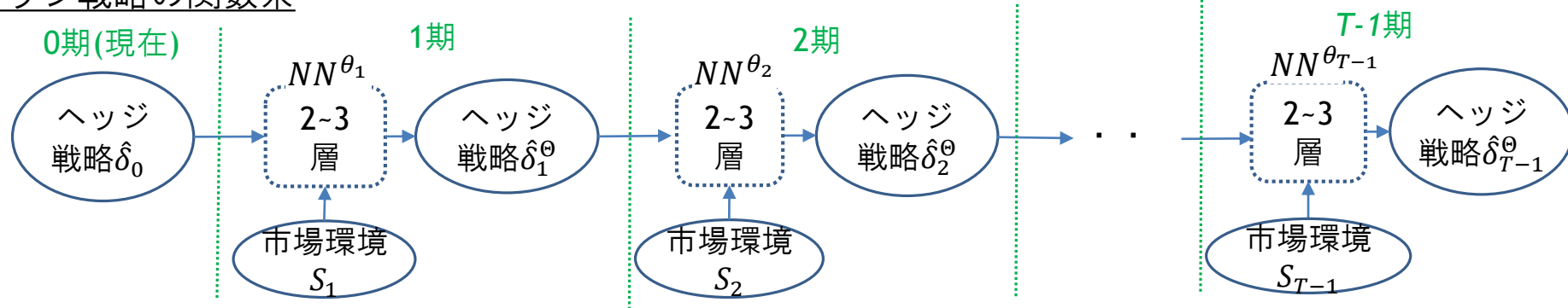
# ディープ・ヘッジングの枠組み

ヘッジ戦略を、大量のパラメータを持つ関数 $\hat{\delta}_t^\theta$ だと仮定し、損失リスクを最小化  
 するようパラメータ $\theta^*$ を決定する枠組み

- ・ 深層学習：膨大な関数(ニューラルネットワーク/NN; 付録8)による、データ・マッピング



## ヘッジ戦略の関数系



- ・  $\hat{\delta}_{i+1}^\theta = NN^{\theta_{i+1}}(\hat{\delta}_i^\theta, S_{i+1})$ : 前期のヘッジ戦略とその時点の市場環境から、ヘッジ戦略を決定
- ・ 例.  $NN^{\theta_i}$ のパラ数:300個,  $T=30$  程度

- ✓ 学習：市場環境の将来シナリオを仮定し、損失リスクを最小化するよう、  
 パラメータ $\theta^* = \{\theta_1^*, \theta_2^*, \dots, \theta_{T-1}^*\}$  を探索
- 活用：実現した市場環境 $s_1, s_2, \dots, s_T$ に応じて、 $\hat{\delta}_{i+1}^{\theta^*} = NN^{\theta_{i+1}^*}(\hat{\delta}_i^{\theta^*}, s_{i+1})$ と逐次的に代入し、  
 ヘッジ戦略 $\hat{\delta}_1^{\theta^*}, \hat{\delta}_2^{\theta^*}, \dots, \hat{\delta}_{T-1}^{\theta^*}$ を決定していく

## 損失リスク最小化問題が数値的に解ける仕組み・理論背景

10

損失リスク最小化問題を、パラメータに関する最小化問題で近似  
— 近似の理論的な妥当性（収束）も保証される

- ✓ ヘッジ戦略  $\delta_t$  の探索問題を、パラメータ  $\Theta = \theta_1, \theta_2, \dots, \theta_{T-1}$  の探索問題に置き換える  
— 無限次元の探索問題を有限次元に射影

全てのヘッジ戦略の中で  
リスク量を最小化するものを探索

$$\min_{\{\delta_t\}} \rho(PL_T^{\delta}) \approx -Z + \sum_{t=0}^{T-1} \delta_t (S_{t+1} - S_t) - C(\delta_t)$$

NNの構造を仮定したヘッジ戦略の中でリス  
ク量を最小化するものを探索

$$\min_{\Theta = \theta_1, \theta_2, \dots, \theta_{t-1}} \rho(PL_T^{\hat{\delta}^{\Theta}}) \approx -Z + \sum_{t=0}^{T-1} \hat{\delta}_t^{\Theta} (S_{t+1} - S_t) - C(\hat{\delta}_t^{\Theta})$$

(\*)  $\rho$  : 期待ショートフォール等

- 上記近似の妥当性が、理論的に示された ([Buehler, et. al, 2019])  
— ニューラルネットワークの普遍近似性能 (付録9-3) により、大きなニューラルネットワークにより、右辺は左辺をいくらでも良く近似可能 (付録12)
- 右辺のパラメータの最適化問題を、深層学習分野で発展した技術を駆使して解く

# ヘッジ戦略の学習の実装

## 超高次元空間のパラメータ探索

→ 「リスク尺度の性質(期待値による表現) + 深層学習の技術」 で対処

### 探索アルゴリズムの基本

①  $\theta$ の初期値 $\theta^{(0)}$ を設定

②  $\rho(PL_T^{\hat{\delta}^\theta})$ が最も小さくなる勾配方向( $\nabla\rho(PL_T^{\hat{\delta}^\theta})$ )に $\theta$ を動かし、最適解 $\theta^*$ を探索

$$\theta^{(k+1)} = \theta^{(k)} + \nabla\rho(PL_T^{\hat{\delta}^{\theta^{(k)}}})$$

差分近似

✓  $\rho(PL_T^{\hat{\delta}^{\theta^{(k)}}})$ の計算(分布の裾)を、期待値として計算

— 期待ショートフォール等に特有の性質 (VaRは期待値として表現できず、使いにくい)

— 予め生成した市場環境 $S_t$ 等の将来シナリオを使って、 $\rho(PL_T^{\hat{\delta}^{\theta^{(k)}}})$ を計算

自社のポートフォリオ価値に影響すると考えられる情報を加味可能

✓ アルゴリズムの工夫

$\theta$ は、超高次元(取引可能日×ヘッジ手段の数×NNのパラメータ数)

— [Buehler, et. al, 2019]の数値例だと、23,280次元

→  $\nabla\rho(PL_T^{\hat{\delta}^{\theta^{(k)}}})$ の計算が非常に負荷が大きく、深層学習技術(誤差逆伝播法、確率勾配法)を活用

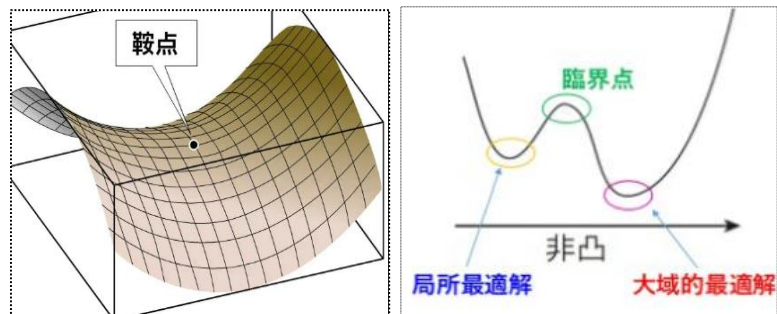
— 将来にわたる取引戦略のうち、合計の損益のリスクに影響の大きなところを特定し、調整する仕組み(適応的近似)がNNに備わっている(付録9-1)

# 深層学習の現在

実用での大成功＋周辺技術の発展＋理論研究 ⇒ 既存機械学習手法と一線を画す手法に  
— ただし、モンテカルロ法等の金融工学の既存の基盤技術に比べ、  
理論（特に実用上の設定の根拠や収束精度の保証等）は、必ずしも整っていない

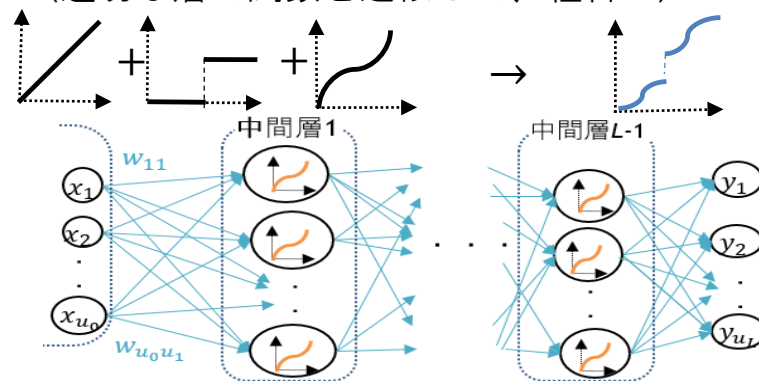
- ✓ 多層ニューラルネットの学習の成功を契機に、深層学習は第3次AIブームの中心に。AlphaGoの成功等、活用範囲が、回帰・予測から制御に広がりつつある（付録6）
- ✓ 深層学習の活用が進んだ理由は複合的、インフラや経験則の蓄積等も重要な要素（付録7）
- ✓ 学習や近似の技術、自然性能があることが徐々に明らかに（付録8, 9, 10）  
例：最適化で問題が起きにくいパラメータ空間の形状  
モデルの複雑さを自動調整し（近似対象の複雑さに合わせ）、過剰適合（過学習）を防ぐ仕組み

最適化で問題となる例  
(パラ更新できない)



適応型近似

(適切な層で関数を近似して、組合せ)



# 学習させたヘッジ戦略の妥当性

数理ファイナンスの既存の技術・理論と比較することで、妥当性を確認  
— 損失リスク最小化問題は、解の妥当性確認も容易ではない

## 1. 単純な問題での確認

取引コストなしの設定で、リスク中立評価に基づくヘッジと比較

→ ヘッジ誤差（実現損益）やグリークスがほぼ同じ挙動をすることを確認

— その後の研究でも、多様な設定で、同趣旨の挙動確認が行われている

## 2. 理論的に挙動が分かっている問題での確認

コストを定数倍 ( $C_k \rightarrow \varepsilon C_k$ ) した場合に、ヘッジ対象の商品の効用無差別価格  $p_\varepsilon$  (\*) が、理論通り変化することを確認

— 原資産価格が特定の条件を満たす1次元の確率微分方程式に従う場合、

$$p_\varepsilon - p_0 \rightarrow \varepsilon^{2/3} \quad (\varepsilon \downarrow 0)$$

が成立することが知られている [Rogers, 2004] [Muhle-Karbe et, 2017]

- ▶ ブラックショールズ・モデルに関し、上記が成り立つことが初めて数値的に示され、さらに、理論的に示されていなかったヘストンモデルでも同様の挙動が観測された  
→ ファイナンス論で理論的にしか解析できなかった問題に、深層学習が数値的手法を提供

(\*) ヘッジ対象  $Z_T$  を保有していてもいなくても、リスクが等しくなるよう価格を設定

—  $\pi(Z) = \inf_{\delta} \rho \left( PL_T^\delta(Z) \right)$  として、 $\pi(-Z + p) = \pi(0)$  を満たす価格  $p$

## デリバティブ実務上の利点・留意点

14

**利点** : 定量化困難だった取引コスト等の影響が分析可能

→ ヘッジの精緻化・自動化に繋がらう

1. 取引コストや取引制限を加味したヘッジコスト・戦略を算出可能で、現実の条件を加味したポートフォリオ管理やチャージコストの精緻化が可能
2. グリークス計算に基づく、トレーダーによるチャージの判断等、手作業が減る
3. 商品群を横断したヘッジ

**留意点** : 収束確認・結果の解釈の困難さ、設定の煩雑さ

→ リスク中立評価との比較が必要（特に価格付け）

1. 深層学習技術に起因するリスク(学習が収束しない、過剰適合等)
  - 精度保証、設定は難しく、計算負荷は非常に大きくなりうる
2. 訓練データ(ヘッジ手段等の将来シナリオ)が適切でないリスク
3. リスク尺度やそのパラメータ、損益の定式化が不適切なリスク
  - 特に、一時点のリスクだけでなく、保有期間中の最大損益リスクの計測が必要

⇒ 新たな分析手段となりうるが、課題は多く既存技術を置換するまでは至っていない

# ヘッジ手法の発展

大規模データに基づく、現実制約を加味したヘッジが可能に  
→ ヘッジ自動化や適用範囲拡大に向けて

## ディープ・ヘッジング

一般的な設定下で、ヘッジを可能とする枠組み

- ・ **主な元データ**：① 学習に用いる将来のシナリオ
  - 自社のポートフォリオ価値に影響する全ての情報を加味可能；  
(予測が必要)
- ② 自らのリスク選好の設定
- ・ **原理**：損失リスク最小化（非完備市場の理論）

大規模データを用いた最適化計算→深層学習技術で

## リスク中立評価に基づくヘッジ

理想的な金融市場の設定下で、複製（完全なヘッジ）を試みる枠組み

- ・ **主な元データ**：現時点の市場情報
- ・ **原理**：複製/無裁定（完備市場の理論）

確率微分方程式の期待値計算



§ 1. 金融実務におけるヘッジの問題

§ 2. ディープ・ヘッジングの枠組み

§ 3. 研究の潮流

# ディープ・ヘッジング：将来シナリオ生成(重要論点)

17

外生所与のシナリオでヘッジ戦略を学習させることの限界  
→シナリオ生成にも、深層学習を用いる

✓ 実確率の下での将来シナリオの生成(予測)：困難多数

➤ 深層学習によるデータ生成

[Wiese et al, 2019]: 株式オプションのシナリオ生成をGAN (Generative adversarial network) により行う手法

[Buehler, et al, 2020]: 少数の過去の市場データから、仮想的なシナリオを生成する方法

[Buehler, et al, 2021]: パスのシグナチャー (確率微分方程式の高階の反復積分の情報) を用いて、VAE (Variational Auto Encoder) により生成する手法

→シナリオ生成とヘッジ戦略の学習を、逐次的に繰り返す学習 (強化学習)、リアルタイムで情報を反映させる、完全にモデル依らない自動ヘッジが、実現しうるか

・この他、将来シナリオが従う確率分布をより現実的な設定とする試みとして、[Horvath, 2021]ではラフ・ボラティリティモデル、[Lütkebohmert, 2021]では不確実性な係数に従う確率微分方程式が、それぞれ用いられている

➤ ドリフトを推定する必要がある

[Buehler, et al, 2022]: オプションのシナリオ間で裁定が起きないシナリオを生成する手法

➤ ある一断面のポートフォリオ情報で学習するため、次営業日にポートフォリオが変わると毎回学習し直す必要がある

[Buehler, et al, 2022]: 一断面での学習結果を使いまわす(ディープ・ベルマン・ヘッジ)

# ディープ・ヘッジング：デリバ実務での活用に向けた研究

18

デリバティブ実務への適用を主眼に、様々な拡張・技術開発が進んでいる(付録13)

## 拡張研究

- ① より複雑な金融商品のヘッジ問題での数値計算  
[Carbonneau, 2021]:超長期のデリバティブ等
- ② 損失リスクの計測方法を変更した場合の技術開発、数値計算  
[Cao, Hull, et al, 2020]:実現損益ではなく、評価損益を用いる
- ③ NNによる取引戦略の別表現・学習方法工夫  
[Zhang, Huang, 2021]:短期記憶リカレントニューラルネットワーク (LSTM-RNN)
- ④ 解釈可能性が高い機械構造 → リスク中立評価に基づくヘッジとの組み合わせ  
[Tawada, Sugimura, 2020], [Imaki et al, 2021]

## 実務上の要請(今後の進展に期待)

- (1) 学習の高速化・安定化
- (2) 解釈可能性 → 既存リスク中立評価との整合性・並行参照;学習/シナリオ見直しタイミング
- (3) 保有期間中の最大損失リスクのコントロール  
→ **動的リスク尺度**[Klöppel, Schweizer, 2007]下でのヘッジ ファイナンス論的にも興味深い
- (4) キャリブレーション対象/Totem時価参照商品との整合性(「補間性能」の維持)  
→ 訓練データ(将来シナリオ)に制約条件として組入れ
- (5) XVA(多種の価格調整)を加味したヘッジ戦略

# ディープ・ヘッジング：リスク管理への応用可能性

19

損失リスク最小化問題の枠組みは、ALMやストレステストにも応用可能(付録14)

- ALMへの応用(付録14-1)
  - [Krabichler, Teichmann, 2020]：銀行のバンキング部門
  - [Fernandez-Arjona, Filipović, 2021]：保険会社の議論
  
- ストレステストへの応用(付録14-2)
  - [清水, 1998]で論じられた枠組みに適用することも可能
  
- ✓ 問題の定式化の難しさ、制約条件の複雑性・時間変動性等（規制や経営課題等）
  - **デリバティブのヘッジに活用する以上の難しさ**を伴う
  - ① デリバティブ分野で、ディープ・ヘッジング技術を成熟
  - ② リスク管理の現場で、議論・チューニング、経営とのコミュニケーション等

↓

将来的に、大規模データに基づく意思決定の一助として、**金融市場以外**でも活用しうるか

# 深層学習によるファイナンスの新展開

強化学習が台頭、予測から**制御・生成**等のタスクへ深層学習の活用が広がる  
 → ファイナンス分野：ヘッジの問題を始め、強化学習と相性が良い

	資産運用	市場分析	リスク管理	デリバティブ
予測	資産価格の将来予測 [Patel et al, 2015]		リスク管理全般 [Mashrur et al, 2020]	デリバティブの時価 評価モデルの近似 [Hutchinson et al, 1994]
制御	ポートフォリオ最適化 [Jiang et al. 2017]			ディープ・キャリブレーション [Hernandez, 2017]
	最適執行問題 [Ning et al, 2021]	マーケットメイキング [Spooner, 2018]		ディープ・ヘッジング [Buehler et al, 2019]
生成	アルゴトレード [Deng, 2016]		前進後退SDEの近似→ CVA/IM計算 [Weinan E et al, 2017]	
	金融時系列データの生成 [Wiese et al, 2019]			

# ディープ・キャリブレーション

21

現在の最適化技術を深層学習で置き換え、高速化

→ 新しい確率モデル（ラフ・ボラティリティ）の実用化へ（付録15）

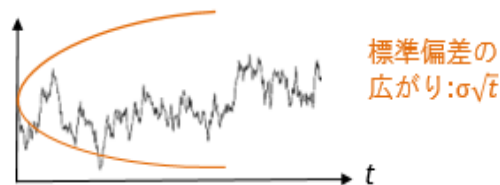
- ✓ キャリブレーション：市場で価格が出ている高流動性商品に関し、モデル価格が市場価格に一致するよう、モデル・パラメータを推計する（最適化）
  - ニュートン法等により日次でモデル・パラメータを推計するのが主流  
計算コストが高く収束の問題が起こりがち（特にモンテカルロ法が必要な場合は困難）  
→現在のデリバティブのモデル開発・運用の中心的問題

## 深層学習による解決

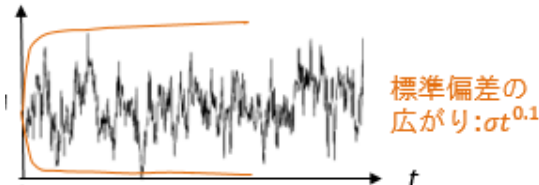
予めモデル・パラメータとモデル価格の関係をNNで学習し、日次の最適化計算を置換

- 現状の計算ロジックを変更することなく、計算の高速化・安定化  
ー キャリブレーション・エラーの原因特定もしやすい
- NNの結果検証しやすく(テストデータを生成可能な教師あり学習)、計算規模も小さい
- “Model calibration with neural networks” (Hernandez著, Risk.net, 2017)で数値例が提示された後、様々な実務上の計算が、容易に高速化・安定化されることが報告され、特に、市場変動に対する表現力が高いラフ・ボラティリティモデル[Gatheral, 2018]の実用化に繋がりをう

既存モデルのボラ過程（マルコフ）



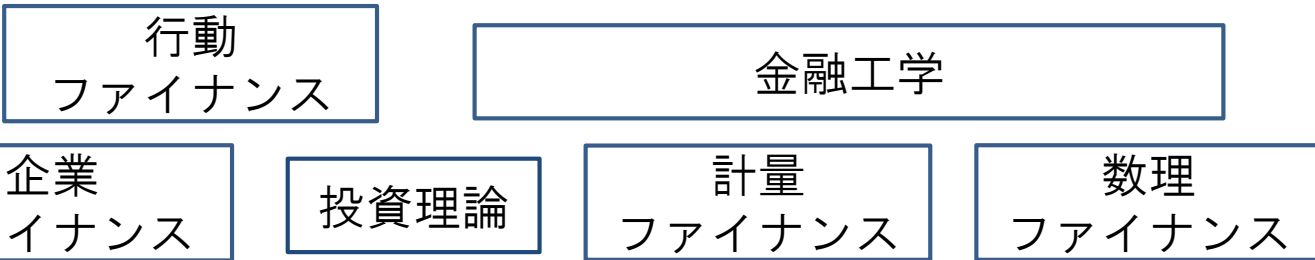
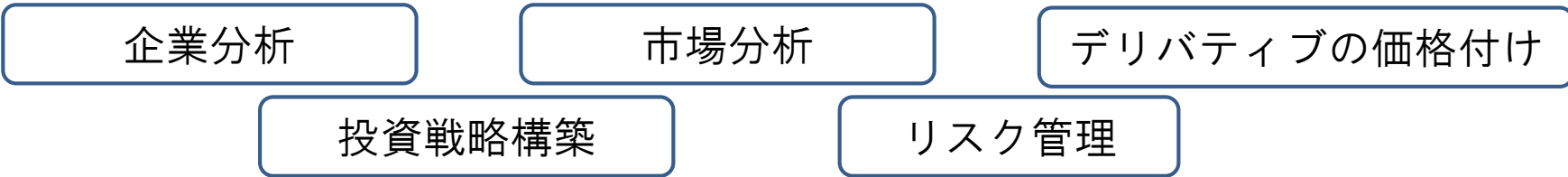
ラフ・ボラ過程（非マルコフ）



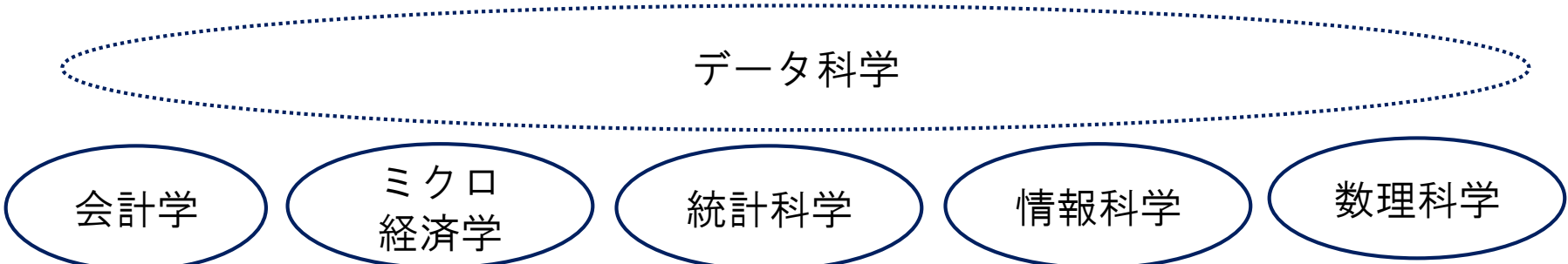
## おわりに

1. 深層学習は、短期的・長期的に多様な角度から、ファイナンスに新展開をもたらす可能性を孕んでいる
2. もっとも、深層学習の特性上、現場のチューニングが非常に重要であり、様々な試行錯誤をし、さらにその理論的根拠も確立していく必要がある
3. 本日紹介したディープ・ヘッジングを中心に、ディープ・キャリブレーション等も含め、サーベイ論文を執筆中

(実務分野)



(学術分野)





$S = \{S_k\}_{k=0,1,\dots,n}$  を一般に  $d$  個の資産とする

- ✓  $0 = t_0 < \dots < t_k < t_{k+1} < \dots < t_n = T$ : 取引可能 (リバランス) 日
- ✓  $(\Omega, F, P)$ : 確率空間,  $\Omega = \{\omega_1, \dots, \omega_N\}$ ,  $\{F_k\}_{k=1,\dots,n}$ : フィルトレーションの設定の下、各変数の定式上の条件は、
- ✓  $S = \{S_k\}_{k=0,1,\dots,n}$ ,  $\delta = \{\delta_k\}_{k=0,1,\dots,n}: \{t_0, \dots, t_n\} \times \Omega \rightarrow R^d$  :  $F_k$ -適合
- ✓  $Z_T: \Omega \rightarrow R$  :  $F_n$ -可測,
- ✓  $C(\delta_t) := \sum_{k=1}^t c_k (\delta_k - \delta_{k-1})$ ;  $c_k: \Omega \times R^d \rightarrow R$  で、任意の  $\omega \in \Omega$  に対し  $c_k(\cdot)$  が右側半連続として、 $PL_T^\delta$  を  $F_n$ -可測な確率変数として定義する

損失リスク最小化問題のニューラルネットによる近似の収束を保証するために必要な条件

$\chi$  を  $\Omega$  上の確率変数とする。

凸リスク尺度の定義

$\rho: \chi \rightarrow R$  が凸リスク尺度 :  $\Leftrightarrow$

- (1) 単調減少性:  $X_1, X_2 \in \chi$  に対し、 $X_1 \leq X_2 \Rightarrow \rho(X_1) \leq \rho(X_2)$
- (2) 凸性:  $\alpha \in [0,1]$ ,  $X_1, X_2 \in \chi$  に対し、 $\rho(\alpha X_1 + (1 - \alpha)X_2) \leq \alpha \rho(X_1) + (1 - \alpha)\rho(X_2)$
- (3) 定数不変性:  $c \in R$ ,  $X \in \chi$  に対し、 $\rho(X + c) = \rho(X) - c$

- ・  $\{S_t^{(i)}\}_{i=0,1,\dots,d}$  : ヘッジ手段（安全資産  $S_t^{(0)}$  と危険資産  $S_t^{(1)}, \dots, S_t^{(d)}$ ）の価格過程  
 $0 \leq t \leq T$   
 → ブラウン運動で駆動される確率微分方程式に従うと仮定  
 （その確率空間を  $(\Omega, \{F_t\}, P)$  とする）
- ・ 自己調達の戦略（外部資金の流入・流出はない）

1. 任意の  $Z_T \in L^2(\Omega)$  に対し、ある  $p_0 \in R$  と  $\{\delta_t^{(i)}\}_{i=0,1,\dots,d}$  ( $F_t$ -可測; 2乗可積分な確率過程) が存在して

$$p_0 - Z_T + \sum_{i=0}^d \int_0^T \delta_t^{(i)} dS_t^{(i)} = 0 \quad \text{を満たす}$$

2. 特に、危険資産が一つで、その価格過程が対数正規分布に従う ( $dS_t^{(1)} = rS_t^{(1)} dt + \sigma S_t^{(1)} dB_t$ ) 時、  
 $\left( \partial_t + \frac{1}{2} \sigma^2 x^2 \partial_{xx} + rx \partial_x - r \right) u(t, x) = 0, \quad u(T, x) = Z_T$  を満たす  $u$  を用いて

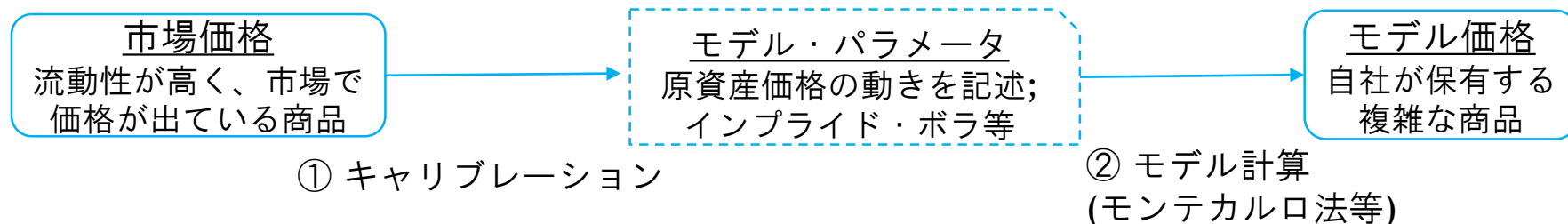
$$p_0 = u(0, S_0^{(1)}), \quad \delta_t^{(0)} = \partial_t u(t, S_t^{(1)}) \quad \text{と表現できる}$$

⇒ ヘッジ戦略を  $\delta_t^{(0)} = \partial_t u(t, S_t^{(1)})$  とするのが、デルタ・ヘッジ

(\*) 詳細の条件等は、[関根, 2005] 3.4節参照

## 複雑なデリバティブの時価評価/リスク管理の例

1. 時価評価：対象商品の原資産価格の動き(モデル)を仮定して、リスク中立価格付け
  - ① **キャリブレーション**:市場で価格が付いている金融商品に関し、モデル価格が市場価格に一致するよう最適化して、モデル・パラメータを推定(逆算)
    - 仮のモデル・パラメータで何度もモデル計算して、市場価格に近付けていく(最適化)
  - ② 対象商品のモデル計算(無裁定を保った価格付け；期待値計算)



## 2. **ヘッジ**操作

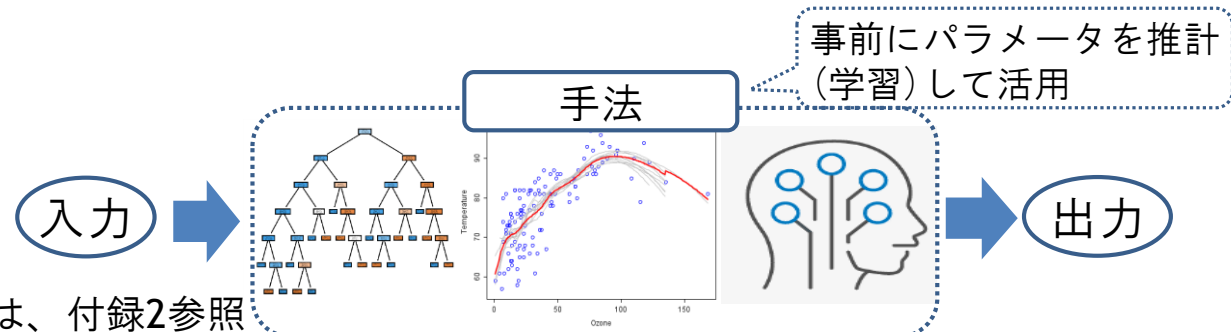
将来のリスクを減らすため、保有する商品と将来キャッシュフローが同じになるポートフォリオを構築する(複製)ことを目指す

- 例えば、グリークス(市場の指標等が1単位動いた時のモデル価格の変化)に基づき取引

- ✓ 時価評価では、多くの商品にキャリブレーションすることが望ましく(それらの商品と裁定が発生しないため)、**計算負荷が問題に**
- ✓ 上記のリスク中立に基づくヘッジ操作では、取引コスト等の現実制約を加味するのは、**数値的に困難**で、実際に複製を達成するのは難しい

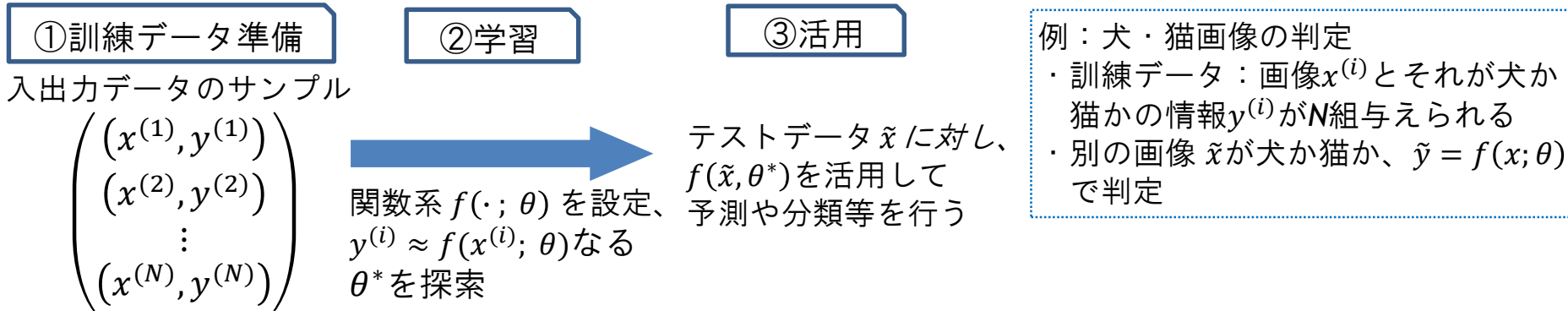
深層学習とは：多層のニューラルネットワーク（NN）等を用いた、機械学習の一手法

- ✓ 機械学習とは  
コンピュータに、データからルールや知識を学習させ、その結果を用いて問題解決する手法  
— 人間の判断による演繹的な解法 → データによる帰納的な解法
- ✓ 学習とは：パラメータ付けられた関数系を仮定して、そのパラメータを推定すること
- ✓ 機械学習の分類：2軸
  - ① 問題設定：データの与えられ方、学習の目的・成果の違い  
代表例：教師あり学習、教師なし学習、強化学習（次ページ）
  - ② 手法：推定に使用する、パラメータ付けられた関数系 深層学習：②手法による分類  
例：ロジスティック回帰、ランダムフォレスト、サポートベクターマシン、k近傍法  
NN（多層パーセプトロン）  
→ 脳の構造を模した関数系：線形・非線形変換の繰返しで、多次元の複雑な関数を模倣；パラメータ数が極めて多い



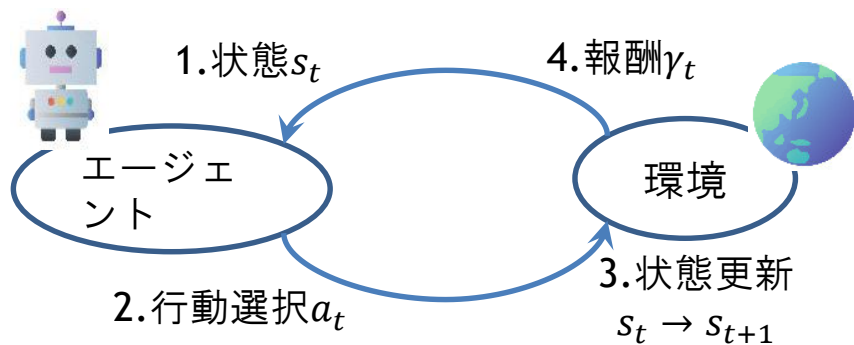
※機械学習の全体像・金融への応用は、付録2参照

## 教師あり学習：与えられた入出力データの関係性を推定する問題



※教師なし学習：訓練データが入力  $x^{(i)}$  のみで、何らかの特長量の関数  $f$  のパラメータを探索

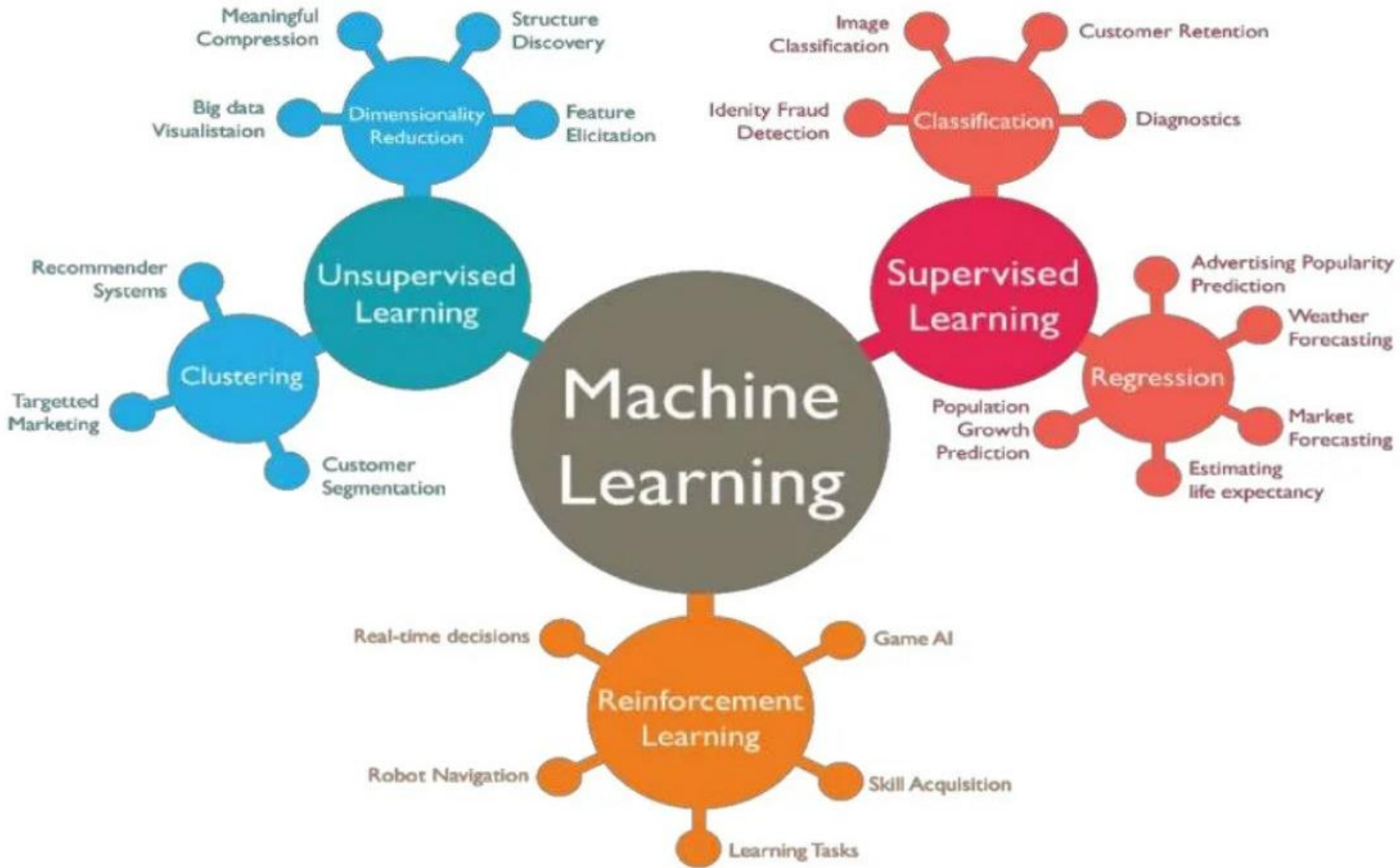
## 強化学習：逐次的に変わる環境中で、将来にわたり得られる報酬を最大化する問題 — 与えられる情報（訓練データ）に応じて、うまく関数系を設定して推定



「確率的に状態更新され、それに応じて報酬が決まる問題」では、例えば、

- ・ 報酬の和  $\sum_t \gamma_t$  の期待値を、 $s_t$  や  $a_t$  の関数として推定し、最大化する  $a_t$  を求める
- ・ 行動  $a_t$  を  $s_t$  の関数と仮定して、報酬をシミュレーションして、報酬を最大化する  $a_t$  を求める

※強化学習の詳細（枠組み・一般的手法）は、付録3参照



## 付録6. 深層学習の歴史

AI・機械学習（深層学習）は、2度のブームを経て、現在第3次ブーム

— それぞれ、契機となった技術革新、及び、ブームが去った限界・問題があった

第1次ブーム  
(1950-60年代)

AI誕生(1956)

パーセプトロン(1957)

・視神経をモデル化

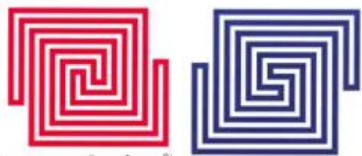
線形SVM(1963)



簡単な図形の判別すら  
できない例が発見

[Minsky, 1969]

線形性の限界



線形パーセプトロン  
では区別できず,,

第2次ブーム  
(1980-90年代)

誤差逆伝播法

NNの学習(1986)

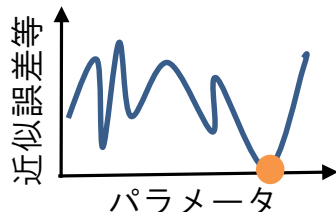
→CNNによる画像処理  
にも成功



2層程度のNNでしか学習  
がうまくいかない  
(収束しない/過剰適合)

非凸性の問題

付録10の一焦点



学習(関数系のパラ  
メータ推計)が困難,,

第3次ブーム  
(2006年-)

多層NNの学習に成功:事前学習による(Hinton, 2006)

Alex-Netが画像認識コンペで優勝(2012)

- ・教師あり学習、CNNの成功
- ・以降、画像・自然言語処理分野で研究が盛り上がり、各種アルゴリズム・ハード等の整備が進み、毎年コンペの記録を塗り替えるような急発展

AlphaGoが、囲碁のプロ棋士に勝利(2016)

- ・強化学習、モンテカルロ探索の成功
- ・深層学習が、予測/回帰だけでなく「制御」でも有用に

ファイナンスの新展開の  
大きな契機に

以降、多様な分野で、高品質化/自動化/高速化に貢献

- ・BERT:文脈を読める自然言語処理(事前学習/注意機構)
- ・AlphaFold:アミノ酸からタンパク質の構造予測
- ・自動運転、物理や生化学の計算高速化等にも応用を展望

## 付録7. 深層学習の現在：成功の3つの背景、特徴

### ① 数理的技術・性質の深化

NN（線形・非線形変換を何度も繰り返した関数）は、様々な関数を良く近似でき、関数を表現するための学習も、うまくいく

### ② インフラの急発展

- ✓ ハードウェア：GPUによる並列計算、深層専用チップ（TPU、MN-Core等）
  - ✓ データ量の爆発的増加：ストレージ費用の低下、クラウドの活用
  - ✓ フレームワーク等の整備：TensorFlow、PyTorch(Chainer)等
- ⇒ オープン・サイエンス化

### ③ 問題解決の経験則の蓄積

- ✓ 問題設定/アルゴリズムの体系整備  
強化学習との融合（適用範囲が回帰・分類等から制御に）、GAN、半教師あり学習等
- ✓ データ・問題の特性に応じた、関数構造等の決定ノウハウ（帰納バイアス）



他の機械学習の手法比、**大規模**で、**入力データの特性も含め学習**

- ただし、データが少ない問題には不向き

データ数 $N$ :10億(画像処理SEER)  
5000億(言語処理GPT-3)

パラメータ $\theta$ :1.6兆個(言語処理Switch Transformer)  
層の数:1,500(タンパク質の構造予測AlphaFold)

学習時間：  
数か月..



全結合型NN：パラメータ付けられた、線形・非線形変換の繰り返し（合成関数）

✓ 1回変換（1層）の定義

入力  $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}$ 、重みパラメータ  $\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1d} \\ w_{21} & \ddots & & \vdots \\ \vdots & & & \vdots \\ w_{d'1} & & & w_{d'd} \end{pmatrix}$ 、 $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{d'} \end{pmatrix}$ 、活性化関数  $\eta$  とし、

$$\eta(\mathbf{W}\mathbf{x} + \mathbf{b}) = \begin{pmatrix} \eta(\sum_{j=1}^d w_{1j}x_j + b_1) \\ \eta(\sum_{j=1}^d w_{2j}x_j + b_2) \\ \vdots \\ \eta(\sum_{j=1}^d w_{d'j}x_j + b_{d'}) \end{pmatrix} \text{で定義}$$

色々な非線形関数  $\eta$  が使われる

シグモイド    ReLu    階段関数

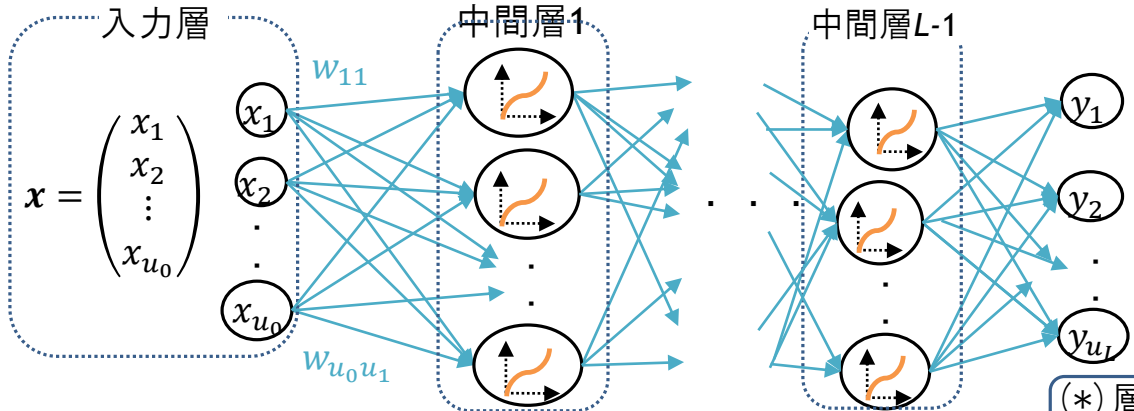
-パラメータは持たない場合が多い

✓ 重みパラメータ  $\mathbf{W}_1, \dots, \mathbf{W}_L, \mathbf{b}_1, \dots, \mathbf{b}_L$  を考え（和集合を  $\theta$  と表記）、 $F_i(\mathbf{x}) := \eta(\mathbf{W}_i\mathbf{x} + \mathbf{b}_i)$  とする

$$NN^\theta(\mathbf{x}) := \mathbf{W}_L(F_{L-1} \circ \dots \circ F_2 \circ F_1(\mathbf{x})) + \mathbf{b}_L \quad (= \mathbf{W}_L(\eta(\mathbf{W}_{L-1}(\eta \dots) + \mathbf{b}_{L-1})) + \mathbf{b}_L)$$

入力  $\mathbf{x}$  に対し、線形・非線形変換を  $L$  回変換を繰り返したものが、出力  $\mathbf{y} = NN^\theta(\mathbf{x})$

$$\mathbf{x} \rightarrow \mathbf{W}_1\mathbf{x} + \mathbf{b}_1 \rightarrow \eta(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \rightarrow \mathbf{W}_2(\eta(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)) + \mathbf{b}_2 \dots \rightarrow \mathbf{W}_L(\eta(\mathbf{W}_{L-1}(\eta \dots) + \mathbf{b}_{L-1})) + \mathbf{b}_L$$



出力層

$$NN^\theta(\mathbf{x}) = \mathbf{y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(u_L)} \end{pmatrix}$$

- ・  $L - 1$ : 中間層の数
- 変換回数(左図横方向)
- ・  $u_1, \dots, u_{L-1}$ : ユニット数
- 変換の次元数

(左図縦方向)  
重みパラメータ  $\theta$  の総数  $\sum_{i=1}^L u_{i-1} \times u_i + u_i$  個

線形変換 非線形変換 線形変換

(\*) 層/ユニット数、活性化関数の種類等は、ハイパーパラメータと呼ばれ、通常決めて固定

## 付録9-1. NNの関数近似性能

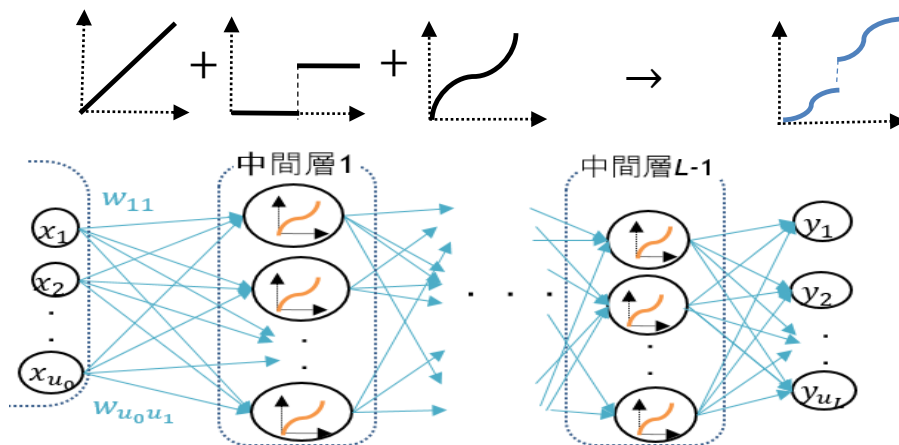
NNは、①どんな関数も近似可能で、②近似したい関数の複雑さに合わせ適切な関数の重ね合わせて近似しており、③他手法比少ないパラメータで高精度に近似可能(付録4)

- ① NNの層数・ユニット数を増やせば、不連続関数も含めいくらかでも精度良く近似可能  
 — NNにより、任意の可積分関数は、任意の精度で近似可能[Hornik, 1991]  
 ・特に、任意の連続関数は、中間層1層のNNで近似可能[Cybenko, 1989]

万能近似能力

→では、なぜ多層にする必要があるか？

- ② 多層NNは、層毎に関数を表現して、それらを重ね合わせることで、**表現したい関数の局所的な滑らかさに応じて解像度を調整**:適応的近似



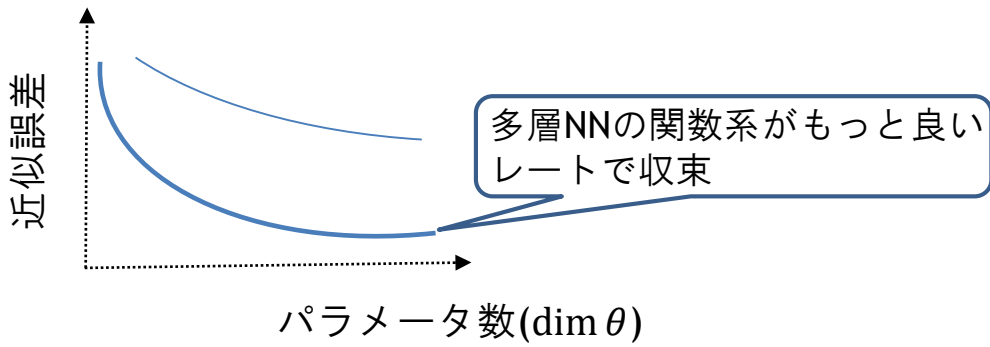
特徴抽出能力/  
 スパース推定  
 (c. f., Lasso)

- ③ さらに、多層NNは、他のどの機械学習手法よりも、パラメータ数に関し近似誤差の収束が早い：最適近似レートの達成  
 — 適応的近似の性質

- [Suzuki, 2019]: 多層NNは、ベゾフ空間の関数の元に対し、適応的近似精度を達成する
- ・ ベゾフ空間  $B_{p,q}^s([0,1]^d)$ : 関数の局所的な微分可能性/可積分性を記述するための関数クラス
    - 定義は、複数があがるが、ヘルダー空間(関数の一様連続性を記述する空間)、ソボレフ空間(関数の大域的な微分可能性/可積分性)を一般化した関数クラス
    - パラメータ  $s$  が微分可能性、 $p, q$  が可積分性を記述

$$\forall f \in B_{p,q}^s([0,1]^d), \exists NN^\theta \text{ s.t., } \|f - NN^\theta\|_{L^r([0,1]^d)} < O\left(\frac{1}{(\dim \theta)^{s/d}}\right), \text{ for } r \in \mathbb{R}$$

- ✓ 適応的近似手法は、他のどんな関数系でも、上記の  $\dim \theta$  に関する近似レートを達成できないことが知られている（最適近似レートを達成）



[Cybenko, 1989]：中間層1層（ $L=2$ ）のNNは、ユニット数を増やすことで、連続関数をいくらでもよく近似できる

活性化関数 $\eta(x) \rightarrow \begin{cases} 1 & (x \rightarrow \infty) \\ 0 & (x \rightarrow -\infty) \end{cases}$ , 連続と仮定する。このとき、以下が成立

$\forall f \in C([0, 1]^d), \forall \varepsilon > 0, \exists u_1 \in N, \{w_{i,j}^{(0)}\}_{\substack{i=1,\dots,d \\ j=1,\dots,u_1}}, \{b_j\}_{j=1,\dots,u_1}, \{w_j^{(1)}\}_{j=1,\dots,u_1} \subset R \text{ s.t.},$

$$\sup_{x \in [0,1]^d} \left| f(x) - \sum_{j=1}^{u_1} w_j^{(1)} \eta \left( \sum_{i=1}^d w_{i,j}^{(0)} x_i + b_j \right) \right| < \varepsilon$$

[Hornik, 1991]：NNにより、任意の可積分関数は、任意の精度で近似可能

活性化関数 $\eta$ が、有界、かつ、定数でない

$\forall f \in L^1([0, 1]^d), \forall \varepsilon > 0, \exists L \in N, \exists L \in N, \{u_i\}_{i=1,\dots,L} \subset N, \text{重みパラメータ } \mathbf{W}_i, \mathbf{b}_i, \text{ s.t.},$

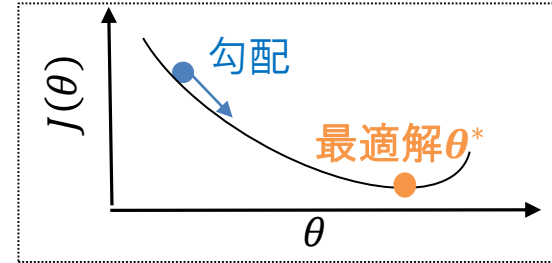
$$\sup_{x \in [0,1]^d} |f(x) - \mathbf{W}_L (\eta(\mathbf{W}_{L-1} (\eta \cdots) + \mathbf{b}_{L-1})) + \mathbf{b}_L| < \varepsilon$$

**学習**：問題毎に設定された損失関数 $J(\theta)$ を、重みパラメータ $\theta$ に関して最適化(最小化)  
 — 例えば、教師あり学習の場合、訓練データの誤差 $J(\theta) = \sum_{i=1}^N |y_i - NN^\theta(x_i)|$  等

- ①  $\theta$ の初期値を設定
- ②  $J(\theta)$ が最も小さくなる**勾配**方向( $\nabla J(\theta)$ )に $\theta$ を動かし、 $J(\theta)$ を更新、**最適解 $\theta^*$** を探索

学習率(定数)

$$J^{(k+1)}(\theta) = J^{(k)}(\theta) + \alpha \cdot \nabla J^{(k)}(\theta)$$

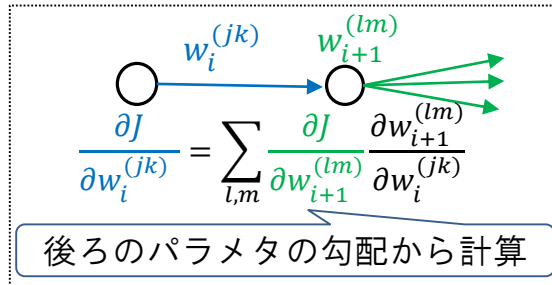


✓  $\nabla J^{(k)}(\theta)$ の効率計算

確率勾配法 + 誤差逆伝播法

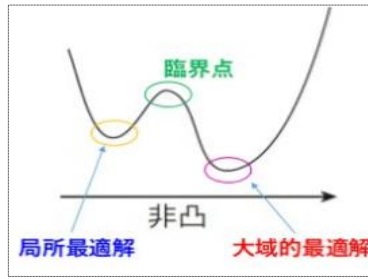
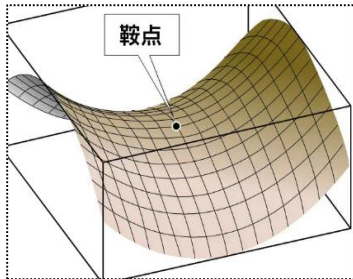
一部のデータのみで勾配を近似計算  
(エポック数：何データに分けるか)

合成関数の微分の法則を用いて効率計算(右図)

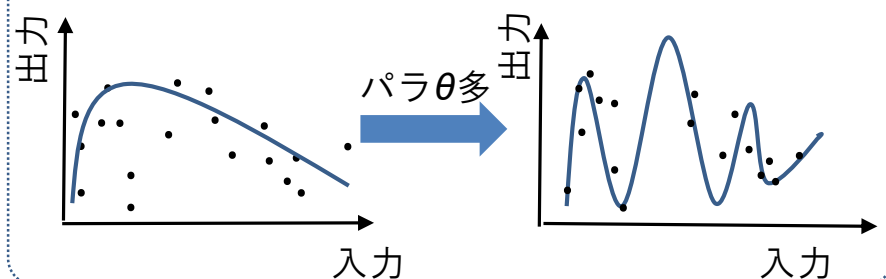


→しかし、様々な問題が起こる：

**問題①勾配消失/爆発**： $J(\theta)$ の非凸性  
鞍点や局所解にはまって、 $J$ を更新できない

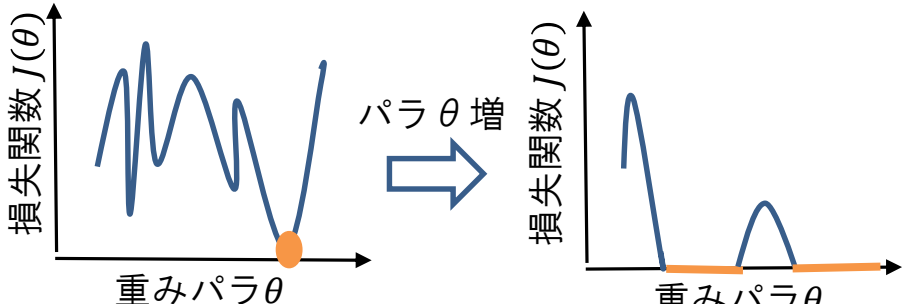
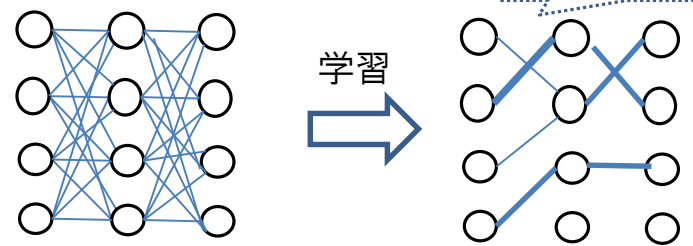


**問題②過剰適合(過学習)**：訓練データには適合するが、未知のデータでは予測等できない



(a) 学習の問題を解決する技術が開発されると共に、(b) NNが自然に問題を解決する性質・仕組みが備わっていることが理論的・実証的に示され始めた

	問題①：勾配消失/爆発	問題②：過剰適合
(a) 技術	スキップ接続：誤差伝播を層間でスムーズに モーメンタム法：勾配降下の履歴を加味して勢いを 学習率の自動調整 <span style="border: 1px dashed black; padding: 2px;">Adam [Kingma, 2015]: これらを組み合わせ</span>	正則化層：入力データの偏りを平滑化(正規化) データオーグメント：訓練データ擬似生成 ドロップアウト：一部パラを0に(モデル簡易化)
注意機構：データの重要な部分に着目するため、動的にデータの流れを変更 <span style="border: 1px dashed black; padding: 2px;">Bertでも重要</span>		

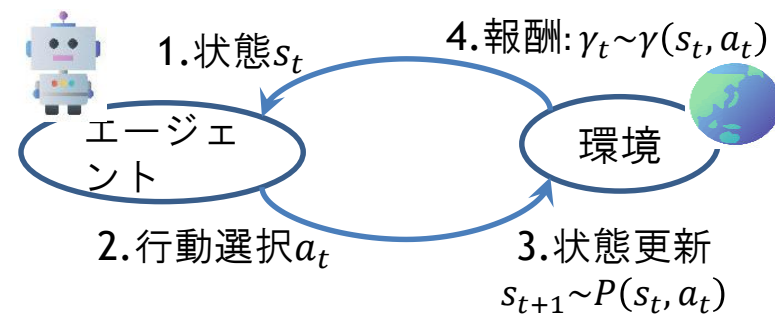
(b) NNの自然性質	<p>特定の場合、<b>損失関数<math>J(\theta)</math>の形状が非常に良い</b></p> <ul style="list-style-type: none"> <li>深層線形NNの局所解は、全て大域解 [Kawaguchi, 2016]</li> <li>ユニット数の多いNNの最適解は、パラメータ空間中で連結 [Garipov, 2018]</li> </ul> 	<p>近似したい関数の複雑さに応じて表現力を<b>自動調整</b>して過剰適合を防ぐ性能あり(多くのパラが0等)：陰的正則化、宝くじ仮説</p> <p>— 従来、表現力が高い(パラ数が多い)学習手法は、過剰適合が起こりやすいと考えられていた(オッカムのカミソリ)</p> <p style="text-align: right;"><span style="border: 1px dashed black; padding: 2px;">適応的近似にも</span></p> 
-------------	---	--

## 付録11. 強化学習の一般論①：問題設定

以下、数学的な定式化を省いて、離散時刻の強化学習の一般的な問題設定と解決法を紹介

✓ 問題設定：本文中の設定に加え、以下を仮定：

- ・  $s_{t+1} \sim P(s_t, a_t)$ 
  - $s_{t+1}$ が $P(s_t, a_t)$ の確率分布に従う
  - 状態更新が $s_t, a_t$ を引数とする確率 $P$ に従う
- ・  $\gamma_t \sim \gamma(s_t, a_t)$ 
  - 報酬が $s_t, a_t$ を引数とするある関数 $\gamma$ に従う



この時、将来にわたり得られる報酬の期待値

$$E \left[ \sum_{t=0}^{\infty} \gamma(s_t, a_t) \right]$$

期待値 $E$ は、確率 $P$ だけでなく、行動 $a_t$ にも依存する点注意

を最大化するように、 $\{a_t\}$ を決定する問題

(\*) 一般には、上記に加え、割引率を考える

(\*) 上記、 $E[\sum_{t=0}^{\infty} \gamma(s_t, a_t)]$ に関する方程式がベルマン方程式と呼ばれるもので、ファイナンス分野では、古くから使われてきたもの

## 付録11. 強化学習の一般論②：代表的解法

### ✓ 代表的な解決方法（関数系の設定と学習方法）

#### ① 価値ベースの方法

状態価値  $V(s) = E[\sum_{t=1}^{\infty} \gamma(s_t, a_t) | \{a_t\}_{t=0,1,\dots}, s_0 = s]$  か

行動価値  $Q(s, a) = E[\sum_{t=1}^{\infty} \gamma(s_t, a_t) | \{a_t\}_{t=1,\dots}, s_0 = s, a_0 = a]$  を関数としてNNで推定、その関数を最大化するように、 $\{a_t\}$ を決定

AlphaGoは、CNNを用いて、行動価値 $Q$ を推定(Q学習)

#### ② 方策ベースの方法

行動 $\{a_t\}$ を、方策関数 $\pi$ で $a_t = \pi(s_t)$ とモデル化し、 $E[\sum_{t=0}^{\infty} \gamma(s_t, \pi(s_t))]$ を最大化

方策ベースの方法では、方策に関する勾配を求める必要があり、通常方策勾配定理（うまい測度変換）を用いる



$$\lim_{M \rightarrow \infty} \min_{\hat{\delta} \in \mathcal{H}_M} \rho(PL_T^{\hat{\delta}}) = \min_{\delta \in \mathcal{H}} \rho(PL_T^{\delta})$$

$M$ 個の重みパラメータをもつNN  
で近似された取引戦略の集合

ヘッジ戦略全体

- ✓ [Buehler et al., 2019] proposition 2.4
- ✓ NNの普遍近似定理による証明
  - 必要な仮定は、取引コストが右側半連続である等、少ない

様々な最適化目標（損益の定式化 $PL_T$ やリスク尺度 $\rho$ ・期待効用）での数値例や複雑なデリバティブでの検証も報告されている

— ただし、どの最適化目標を採用すべきか、どれだけ複雑なデリバティブでディープ・ヘッジングが効力を発揮するか、いずれも明確ではない

— ブレークスルー論文では、実現損益 $PL_T(Z, \delta) = -Z + \sum_{k=1, \dots, n} (\delta_{t_k} (S_{t_{k+1}} - S_{t_k}) - C_{t_k})$ をC-VaR等の凸リスク尺度の元で最小化することを目標とし、コールオプションのヘッジを考えている

- ✓ [Cao, Hull, et al, 2020]: 実現損益の代わりに、評価損益（将来の支払いのリスク中立測度の元での期待値計算した損益）を適切に用いると、パフォーマンスが向上する
- ✓ その他、最適化目標の変更として、[Carbonneau, 2020]ではリスク均等価格付けが論じられ、[Kolm, Ritter 2019], [Carbonneau, 2021]等でも同様の議論がされている他、より古典的な平均・分散アプローチに基づく定式化も議論されている
- ✓ より複雑な支払い・超長期のエキゾチックデリバティブを扱っている文献も多数あり [Imaki, et al, 2021], [Carbonneau, 2021], [Carbonneau, 2020]等

より複雑な問題設定（エキゾチックデリバティブのヘッジ等）への対応や、学習の高速化・安定化を主眼に、NNの構造・学習方法の工夫が論じられている

— ただし、問題に応じて、NNの構造をどう設計すべきか、理論的な裏付けはなく、学習の高速化・安定化も実験的な議論に留まっている

- ブレークスルー論文では、 $\delta_{t_{i+1}}^{\varphi_{i+1}} = NN^{\theta_i}(\delta_{t_i}^{\varphi_i}, S_{t_i})$ として、方策勾配法を用いた  
なお、 $\delta_{t_{i+1}}^{\varphi_{i+1}} = NN^{\theta_i}(S_{t_i})$ とする（次期のヘッジ戦略の推定に、現在のヘッジポジションの情報を用いない）とすると、パフォーマンスが悪化することも数値的に示されている
- ✓ [Zhang and Huang, 2021]:流動性リスクを追加的に加味することを目的に、LSTM-RNN(Long Short Term Memory Recurrent Neural Network)を用いることを提案
- ✓ [Imaki et al, 2021](PFN):学習時間短縮を目的に、 $\delta_{t_{i+1}}^{\varphi_{i+1}} = NN^{\theta_i}(\delta_{t_i}^{\varphi_i}, S_{t_i})$ の構造を一部簡易化し、学習時間の短縮とパフォーマンスの維持を両立できることを示した
  - その他、[Son and Kim, 2021]では、全結合NNよりパラメータ数の少ないNNを用いても、パフォーマンスが維持されることを示している
- ✓ なお、ブレークスルー論文以前に、[Halperin 2017]で、同様の問題設定（ただし取引コストは加味していない）の元、取引戦略をNNにより学習させる方法を提案しているが、Q学習を用いているため、より現実的な設定に拡張することは難しいと考えられる

## 付録14-1. ALMへの応用

**ALM** (Asset Liability Management) : 執行コストを加味したポート管理方針の策定  
ディーブ・ヘッジングの枠組みは、デリバティブの複製 (リスク管理) を目的に提案されたものであるが、よりマクロな例として、金融機関のALMも同様に定式化可能

- ✓ 銀行のバンキング部門のALMの議論が[Krabichler, Teichmann, 2020]で、保険会社の議論が[Fernandez-Arjona, Filipović, 2021]でされている
  - ✓ ただし、ALMに関しては、その問題の定式化の難しさ、制約条件の複雑性・時間変動性等 (特に、規制や経営課題等定式化されていない制約が多い) から、デリバティブのヘッジに活用する以上の難しさを伴う
- ⇒将来的には、大規模データに基づく意思決定の一助となりうるが、学術的な議論と共に、リスク管理での現場での議論・チューニング、経営とのコミュニケーション等が必要

金研DP [清水, 1998]にて、NNのストレステストへの応用が論じられており、本稿で紹介したディープ・ヘッジングの技術も応用しうる

ー ただし、システム全体の因果等のモデル化が可能なわけではない

- ✓ [清水, 1998]: ストレステストにおいて、フィードバック効果（ストレスを受けた際の取引者の行動）の勘案する一手法として、市場環境を入力データ、取引行動を出力データとするNNにより推定する方法が提案されている。ただし、訓練データである「ストレス時の金融機関の行動」をエキスパートジャッジにより生成。

⇒リスク尺度を設定しディープ・ヘッジングの枠組みを用いることが有用になりうる。

[清水, 1998]

訓練データ

( (ストレス状態1, 行動1)  
 (ストレス状態2, 行動2)  
 ⋮  
 (ストレス状態N, 行動N) )



ある一時点のストレス下での行動を  $NN^\theta$  で推定

ディープ・ヘッジング

設定・訓練データ

・ 金融機関のリスク尺度  
 ・ ストレスシナリオ (N本)



一定期間のストレスシナリオに対する期間を通じた行動を  $NN^{\phi_n}$  で推定

(\*)その他、ストレスシナリオの生成等にQuant GANs [Wiese, et al, 2020]を応用することも展望される

# 付録15-1. 教師あり学習によるキャリブレーション：概要

## 金融実務における課題

**市場価格**  
市場で価格が出て  
いる高流動性商品  
(キャリブ対象)

**モデル・パラメータ**  
原資産価格の動き(モデル)を記述

保有する複  
雑な商品の  
モデル価格

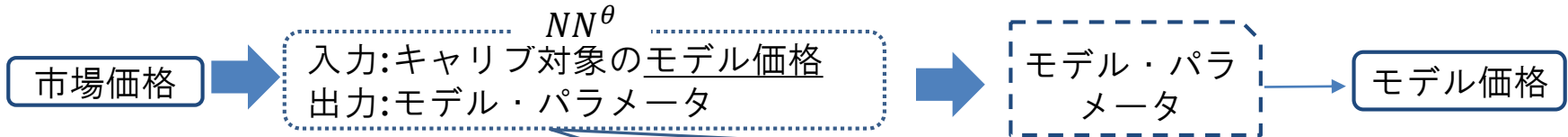
### ① キャリブレーション

キャリブ対象に関し、モデル価格が市場価格に一致するよう、モデル・パラメータを推計  
(1)モデル・パラメータの初期値を設定、(2)キャリブ対象のモデル価格を計算、(3)モデル価格と市場価格の差分に基づきモデル・パラメータを更新、(2),(3)を繰り返し収束させる:ニュートン法

- ✓ 日次の計算負荷が重く、日々の変動する市場価格に対し、ニュートン法が収束しない場合も
  - ・特に、キャリブ対象のモデル価格計算が解析的にできない場合、計算負荷大
  - ・ニュートン法が収束しない原因を明らかにすることは難しく(ニュートン法の設定の問題/与えられた市場価格に合うモデル・パラメータが存在しない等)、モデル・パラメータ仮置きで時価評価等することも

### 深層学習による解決

日次のキャリブレーション(最適化)を、**1度の学習で置き換えて**、高速化/安定化  
- 「モデル価格を入力/モデル・パラメータを出力」として学習させたNNに、市場価格を入力して、キャリブ対象に関し、モデル価格と市場価格に一致するモデル・パラメータを近似的に算出



教師あり学習で重みパラ $\theta$ を学習 (要定期検証)  
訓練データ: 「モデル・パラメータ」と「キャリブ対象のモデル価格」の組

### ブレークスルー 論文

Model calibration with neural networks. Hernandez著 (PwCコンサル, Risk.net, 2017, 現在被引用数59)

- Hull-Whiteモデルのキャリブレーションの数値例を提示

- ✓ その後、様々な実務上の計算が、容易に高速化・安定化されることが報告され、特に、市場変動に対する表現力が高いラフ・ボラティリティモデル[Gatheral, 2018]の実用化に繋がらう

## 付録15-2. キャリブレーションの問題設定

### キャリブレーション

市場価格に整合的になるよう、モデル・パラメータ（デリバティブの原資産価格が従う確率微分方程式のパラメータ）を決める

- ・モデル：パラメータ $\sigma$ を入力して、価格 $P^{(model)}(\sigma)$ を返す
- ・キャリブレーション対象商品の市場価格 $P^{(mkt)}$ が与えられたとき、  

$$P^{(model)}(\sigma) \approx P^{(mkt)}$$

となる、 $\sigma^*$ を探す問題

- ✓ ブラック・ショールズ( $dS_t = rS_t dt + \sigma S_t dW_t$ )では、例えば、モデル・パラメータはボラ $\sigma$ で、  

$$P^{(model)}(\sigma) = S_t N(d_1) - Ke^{-r(T-t)} N(d_2)$$
 (ただし $d_1 = \frac{\log(S_t/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}$ ,  $d_2 = d_1 - \sigma\sqrt{T-t}$ )
- ✓ デリバティブの価格付けでは、 $P^{(mkt)}$ 、 $\sigma$ とも高次元であり（モンテカルロ法等が必要な場合、特に計算負荷大）、近似公式やニュートン法等の最適化手法を駆使

① **キャリブレーション**：最適なモデル・パラメータ $\sigma^*$ を計算

- キャリブ対象：市場価格 $\approx$ モデル価格 $F(\sigma)$

$$\text{モデル: } dS_t = rS_t dt + \sum \sigma_i S_t dW_t$$

$$dS_t = rS_t dt + \sum \sigma_i^* S_t dW_t$$

キャリブ対象の市場価格  
(スワプションマトリックス等)

$$\begin{matrix} P_{T_1, K_1}^{(mkt)} & P_{T_1, K_m}^{(mkt)} & \dots & P_{T_1, K_n}^{(mkt)} \\ P_{T_2, K_1}^{(mkt)} & \ddots & & \\ \vdots & & & \\ P_{T_n, K_1}^{(mkt)} & & & P_{T_n, K_m}^{(mkt)} \end{matrix}$$

モデル価格  
(解約権付スワプション等)

② **モデル計算**： $\sigma^*$ をモデルに代入して、リスク中立確率測度の下で、期待値計算 $E^Q[f(S_t)]$

# 付録15-3. 深層学習による問題解決：アルゴリズムと活用方法 47

教師あり学習により、**モデル・パラメータとモデル価格の関係**を推定

→ある一断面で推定し、日々の計算で活用

① 訓練データ生成： $N$ 回、ランダムに $\hat{\sigma}^{(i)}$ を取り、 $P^{(model)}(\hat{\sigma}^{(i)})$ を計算(モンテカルロ法等)

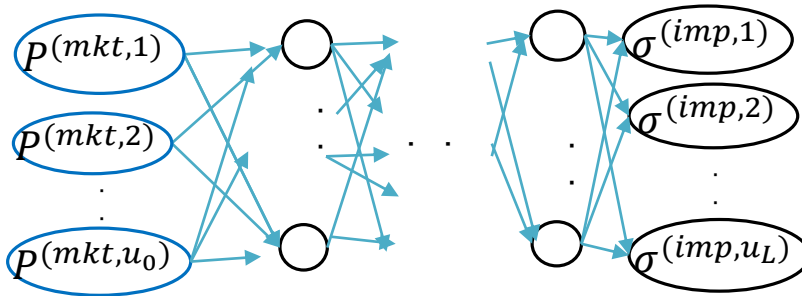
② 学習： $\hat{\sigma}^{(i)} \cong NN^\theta(P^{(model)}(\hat{\sigma}^{(i)}))$  ( $i = 1, \dots, N$ )なる $\theta$ を探索

— モデル価格関数  $P^{(model)}(\cdot)$ の逆関数をNNで学習させる

③ 活用：日々市場で観測される価格 $P^{(mkt)}$ に対し、 $NN^\theta(P^{(mkt)})$ をキャリブレートされたモデル・パラメータとして、使用



流動性の高い  
商品の市場価格



整合的に

$$dr_t = \alpha(\beta - r_t)dt + \sigma^{(imp)}\sqrt{r_t}dW_t$$



複雑で流動性の低い商品の  
価格計算・リスク管理で活用

- ✓  $NN^\theta(\cdot)$ の精度を調整可能：訓練データのサンプル数 $N$ は、自由に決められるため、 $NN^\theta(\cdot)$ の精度を検証しながら、サンプル数 $N$ を調整できる
  - ただし、非常に重い計算（モンテカルロ法等）をサンプル数 $N$ 回行う必要がある
- ✓  $NN^\theta(\cdot)$ の精度は検証可能： $P^{(mkt)}$ と $P^{(model)}(NN^\theta(P^{(mkt)}))$ を比較（交差検証等）
  - ③の日々の活用では、定期的に $NN^\theta(\cdot)$ を見直す必要あり



以上のアルゴリズムは簡易化して記載したが、デリバティブの価格付け問題の場合、問題設定はより複雑

1. 実際には、市場で出ているのは、流動性の高い商品のインプライド・ボラ等であることが多い
2. 満期・ストライク等が異なる商品群を、同時に少数のモデル・パラメータ(期間構造あり)でキャリブレーションする必要がある  
→ 入出力とも構造を保った、キャリブレーションが必須
3. モデル価格が、マーケットレート等、他の外生パラメータに依存  
→ 外生パラメータも、NNの引数として学習させる必要があり、  
訓練データとして、「整合性のとれた」外生パラメータを生成が必要  
— 詳細は、 [Hernandez, 2017]参照

## 付録15-5. 高速化に関する研究

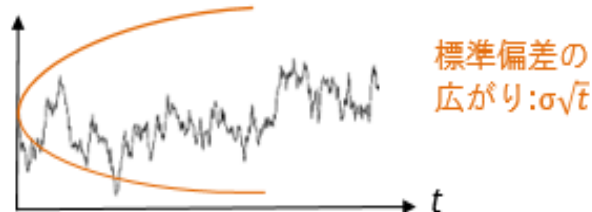
デリバティブの価格付けモデルでの高速化例(数百~数万倍)が多数報告されており、特に、最近研究が進むラフ・ボラティリティの実用化に繋がっている

文献	モデル・対象	NNの構造（全結合層）	学習法/NNのパラメタ	効果
[Hernandez, 2017]	Hull-White スワプション	$L=5$ （中間層4つ） $u_0 = 200, u_l = 64 (l = 1, 2, 3, 4),$ $u_5 = 2$ 活性化関数：ELU → 重みパラメータ数: 29, 634	Adam サンプル数 $N=150,000$ 学習率 $\alpha = 0.001$ エポック数500 ドロップアウト率0.2	1度学習すれば、 日々の計算は 約 <b>250倍高速</b> に
[Horvath, 2021]	Bergomi/rough- Bergomi(*) ボラ・サー フェス	$L=5$ （中間層4つ） $u_0 = 8, u_l = 30 (l = 1, 2, 3, 4),$ $u_5 = 88$ 活性化関数：ELU → 重みパラメータ数: 6,718	Levenberg-Marquart? サンプル数 $N=68,000$ エポック数200 + 正則化等の技術の組合せ	1度学習すれば、 日々の計算は 約 <b>9,000-16,000</b> <b>倍高速</b> に (モンテ最適化比)

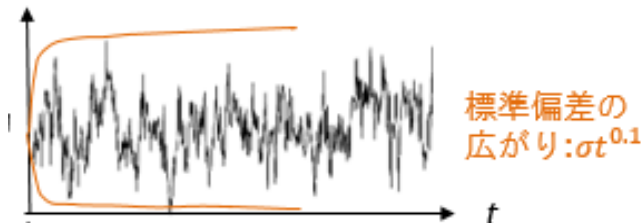
その他、Hestonモデル[Brigo, 2021]、HJMモデル[Benth, 2021]、Gaussian金利モデル[Sabbioni, 2020]、Trolle-Schwartzモデル[Büchel, 2021]等の数値例が報告されている

(\*) ラフ・ボラティリティのイメージ

既存モデルのボラ過程（マルコフ）



ラフ・ボラ過程（非マルコフ）



安定性・妥当性も盛んに議論されており、工夫や検証方法は提案されている  
— 既存の最適化手法でも、安定性は常に問題になる

✓ 安定性向上

1. 2step法 ([Horvath, 2021])

モデル価格関数  $P^{(model)}(\cdot)$  の関数自体をNNで学習させ、その後、通常最適化手法でモデル・パラメータを推定する

- モデル価格関数  $P^{(model)}(\cdot)$  のNNによる近似の研究は古く、1990年代から議論されている(付録5)
- [Liu, 2019], [Büchel, et al, 2021]等でも、同様のアプローチをとっているが、[Itkin, 2017]では、この手法が不安定になるケースがあると指摘されている

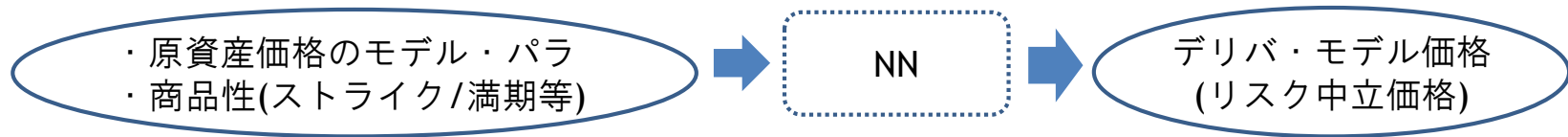
2. Grid base Algorithm ([Horvath, 2021]等)

最適化の際、複数の商品価格の誤差をまとめて損失関数として、重みパラメータを最適化

✓ 学習結果の妥当性の議論

[Liu, 2019]等では、他の最適化手法と比較している他、[Brigo, 2021]ではSHAP値を用いた検証方法を示している

- ✓ デリバティブの理論価格等をNNを用いて近似することを主眼とした研究は、[Hutchinson et al, 1994]に始まり、その後、様々な拡張・改良が提案されてきた。



- これらの研究の問題意識は、学習対象をインプライド・ボラティリティとしたり、様々な価格付けモデルを対象としたもの、学習させたNNを用いてヘッジを試みたもの、学習のインプットを工夫することでより望ましい性質を保証するもの等多岐に渡る。
- これらの研究は、100本を超える論文で論じられており、それらの一覧が[Ruf and Wang, 2020]にまとめられている。代表的な論文以下の通り。

## 付録15-8. 実務上の利点と留意点

### 利点：少ない導入コストで、高速/安定化

- ✓ 教師あり学習で訓練データを生成可能なため、精度の調整/検証が可能
- ✓ 既存ロジックの一部置換で、時価等は不変。また、通常の深層学習比、計算規模が小さい (GPU不要) ため、導入ハードルが低い
- ✓ ラフ・ボラティリティ等、計算負荷が重いが表現力の高いモデルへの対応可能性
- ✓ キャリブレーション可能な市場価格の範囲が事前にわかるため、モデル・パラメータを仮置きする等の対応が減り、時価評価/リスク管理の頑健性向上の観点からも望ましい

### 留意点：検証・更新の必要性

- NNの精度を定期的に検証して、新しい市場データが手にまとまって入手できたタイミングで更新する必要あり（更新の負荷は重いが、検証の負荷は軽い）
  - 一 特に、市場急変時には、モデル価格関数の形状が大きく変わりうる
- 既存の最適化手法同様、不安定になることもあり、安定性確認（特に学習されたNNの形状の検証）が必要

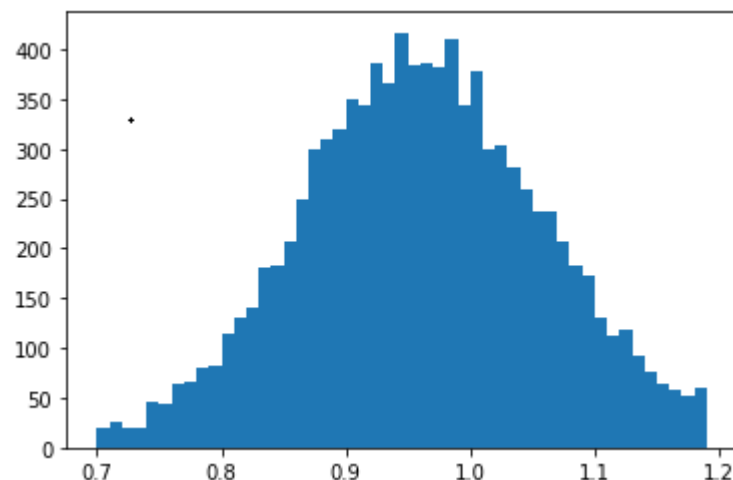
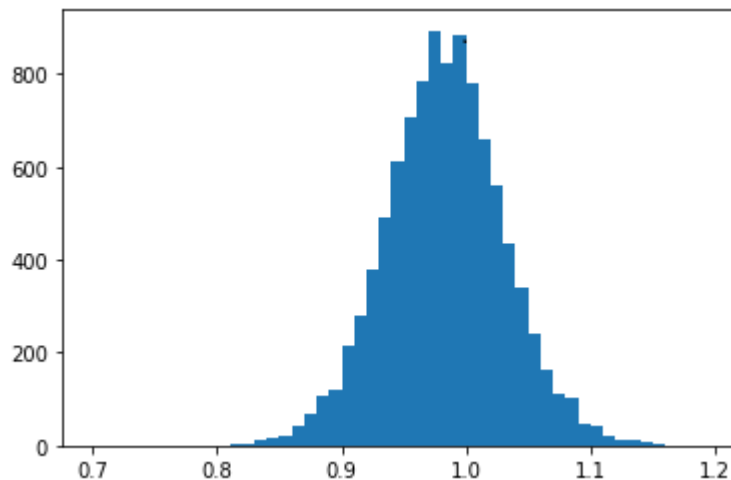
本稿と趣旨が近いサーベイ論文は、以下の通り

- － 本稿で解説する研究にも触れてはいるが、それを中心としたサーベイ論文はない

文献	概要
[Hambly et al., 2021]	強化学習の基礎をまとめた上で、金融への応用を幅広く解説。具体的には、デリバティブのヘッジの他、最適実行問題、ポートフォリオ最適化、ロボ・アドバイザー、電子取引における市場構造分析・価格形成等。
[Ruf and Wang, 2019]	深層学習のデリバティブ分野への応用をまとめたもの。特に、価格関数の近似に関する1990年代からの研究がリスト化されている。
[Mashrur et al., 2020]	機械学習の金融リスク管理への応用を網羅的にまとめたもの。

# 付録17. 将来シナリオと実現シナリオが異なる場合の ディープ・ヘッジングの有効性

- ・ヘッジ手段の価格過程： $dS_t = S_t(\mu dt + \sigma dB_t)$
- ・効用関数  
指数効用： $U(x) = E[e^{-\alpha x}]$ ,
- ・1期のヘッジ戦略のNN：2層・3ユニット
- ・学習のサンプル数：10000、バッチ数：1000
- ✓ 学習用の訓練データに対し、実現したボラティリティが大きくなった場合のヘッジ誤差の度数分布は以下の通り。



→実現したボラティリティが大きくなるにつれ、ヘッジ誤差は急速に増大。

設定 ・ヘストンモデル下でのヨーロピアンコールオプション

(満期  $T=30/365$ 、ヘッジ取引タイミング：日次、ATMストライク)

・ニューラルネットワークの設定：

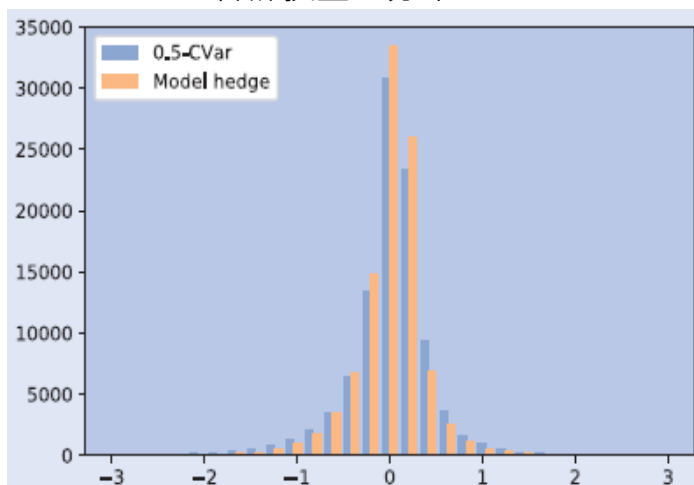
層数  $L=3$ ，ニューロン数  $u_0 = 4, u_1 = u_2 = 17, u_3 = 2$ ，活性化関数  $\eta(x) = \max(0, x)$

・学習の設定

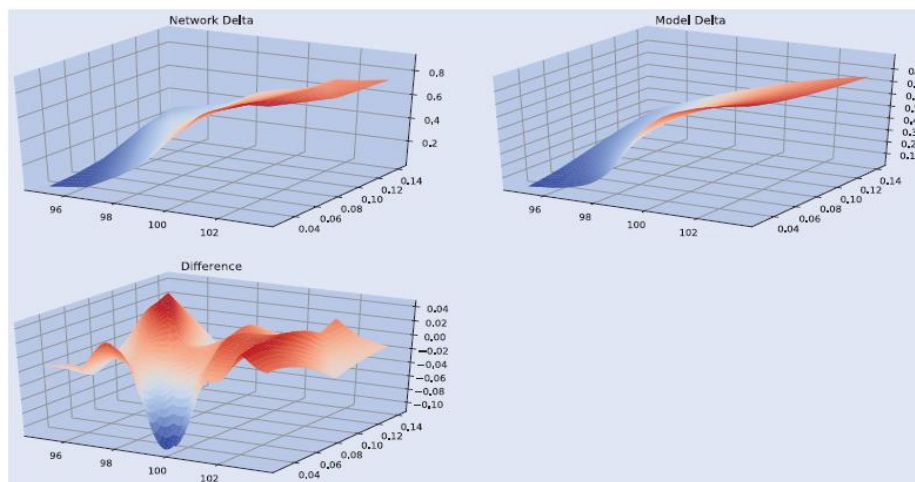
サンプルシナリオ：ヘストンモデル（リスク中立確率下），バッチ数256，学習率0.005

結果：（50%タイルES）リスク中立評価に基づくヘッジとかなり近い挙動

合計損益の分布



デルタの分布



(\*) その他、リスク尺度のパラメータ、デリバティブのペイオフ等を変えた数値実験を実施