Differential Machine Learning を用いたラフ・ボラティリティモデルのキャリブレーション

CALIBRATION OF ROUGH VOLATILITY MODELS USING DIFFERENTIAL MACHINE LEARNING

黒崎地大
Chihiro KUROSAKI

指導教員 劉慶豊

法政大学大学院理工学研究科システム理工学専攻修士課程

In this study, we propose an integrated deep calibration method incorporating Differential Machine Learning (DML) and apply it to the calibration of the rBergomi rough volatility model. While traditional deep calibration suffered from issues such as unstable training and overfitting, we demonstrate that by incorporating derivative information along with option prices into the neural network training, both the calibration accuracy and the stability of the training process can be significantly improved.

Key Words *: Rough volatility, Neural Network, Option pricing, Calibration, Differential Machine Learning*

1. はじめに

近年の金融市場では，短期間におけるボラティリティの変動が従来の予想を大きく上回る激しさを見せ，その複雑な構造を従来のマルコフ型モデルでは十分に捉えきれないとの指摘がある [1]．こうした状況を背景に，ラフボラティリティモデルへの関心が急速に高まっている．一例である rBergomi モデル [2] は，フラクショナルブラウン運動により資産価格とのレバレッジ効果や非マルコフ特性を再現し得るが，そのキャリブレーションには大きな計算負荷が伴う．そのような問題に際し，近年はニューラルネットワーク（以下 NN）を活用したディープキャリブレーション手法が提案され [3]，高速なキャリブレーションを行う研究が進んでいる．

一方，ラフボラティリティモデルのように多くのパラメータを含むモデルでは，ディープキャリブレーション手法における推定精度の低下や学習の不安定性といった課題が指摘されており [4, 5]，さらなる高効率化・高精度化を求めた研究が進んでいる．そこで，本研究では Differential Machine Learning（以下 DML）[7] という新しいアプローチを用いたキャリブレーション手法を提案する．DML を導入することによって，価格に加えてモデルのパラメータに対するオプション価格の微分情報を同時に学習することで，パラメータ空間全域における高精度な近似と学習の安定化を目指す．実証研究として，実際の市場データを用いてキャリブレーションを行い，既存の方法と比較して本研究で開発した手法が高速かつ高精度であることを確認した．

2. ラフボラティリティモデル

(1) フラクショナルブラウン運動 (fBm)

フラクショナルブラウン運動 (fBm) は，従来のブラウン運動を拡張した確率過程であり，その特徴は「時間の自己相関構造」を持つ点にある．通常のブラウン運動では，各時点の増分（変化量）は互いに独立であるが，fBm では過去と現在が関係づけられている．これにより，fBm は過去の変動が未来の動きにも影響するような効果を表現可能となる．

fBm は，Hurst 指数と呼ばれるパラメータ $H \in (0,1)$ を用いて定義されるガウス過程であり，以下のような性質を持つ．

- $H = \frac{1}{2}$ の場合：fBm は標準ブラウン運動に一致する．このとき，各増分は独立である．
- $H > \frac{1}{2}$ の場合：増分が正であった後に再び正になりやすいなど，過去の増分と未来の増分に正の相関を持つ．
- $H < \frac{1}{2}$ の場合：増分が正であった後に負に反転しやすいなど，過去の増分と未来の増分に負の相関をもつ．非常に不規則で「ラフ」な動きを示す．

このように，Hurst 指数 H はプロセスの「粗さ」を定量的に調整する鍵となる．$H < 1/2$ ではパスがより不規則で鋭い変動を示し，$H > 1/2$ ではパスがより平滑で一方向的な傾向を持つ．fBm の共分散関数は，時刻 $s, t \geq 0$ に対して

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

と表せる．ここで，W_t^H は Hurst 指数 H を持つ fBm である．この共分散構造により，fBm は過去全体の軌跡が将来に影響する非マルコフ的特性を有している．

(2) rBergomi モデル

rBergomi モデルは，Bergomi モデルを非マルコフ的かつより一般に拡張したものであり，市場で観測される「ラフ」なボラティリティ構造を再現する上で有用なモデルである．このモデルでは，分散過程 v_t をフラクショナルブラウン運動 (fBm) 由来のカーネルを用いて定義し，短期スケールでの急峻な変動や資産価格との逆相関（レバレッジ効果）を表現する．

定義　rBergomi における株価過程 $\{S_t\}_{t\geq 0}$ と分散過程 $\{v_t\}_{t\geq 0}$ を以下のように与える．

$$\frac{dS_t}{S_t} = \sqrt{v_t}\left(\rho\, dW_t + \sqrt{1-\rho^2}\, dW_t^{\perp}\right), \quad S_0 > 0 \tag{2}$$

$$v_t = \xi_0(t)\exp\left(\eta W_t^H - \frac{\eta^2}{2}t^{2H}\right), \quad v_0 := \xi_0(0) > 0 \tag{3}$$

ここで，$(W, W^{\perp}) = (W_t, W_t^{\perp})_{t\in[0,T]}$ は二つの独立したブラウン運動であり，$\rho \in (-1, 1)$ は株価とボラティリティの間の相関パラメータである．$\eta > 0$ はボラティリティのスケールに関するパラメータ，$\xi_0(t)$ は forward variance curve と呼ばれるもので，さまざまな表現方法があるが本研究では定数で扱う．W_t^H は Hurst パラメータ $H \in (0, 1)$ のフラクショナルブラウン運動であり，

$$W_t^H = \sqrt{2H}\int_0^t (t-s)^{H-\frac{1}{2}} dW_s \tag{4}$$

という Riemann–Liouville 型の確率積分で表される．

(3)　rBergomi モデルによるヨーロピアンコールオプションのシミュレーション

rBergomi モデル下でヨーロピアンコールオプション（行使価格 $K > 0$，満期 $T > 0$）の価格はリスク中立測度下で

$$C(S(0), K, T) = \mathbb{E}[(S(T) - K)^+] \tag{5}$$

により定義される．ここで $(x)^+ = \max(x, 0)$ である．この期待値は，モンテカルロ（以下 MC）シミュレーションによって

$$C(S(0), K, T) \approx \frac{1}{N}\sum_{i=1}^{N}(S^{(i)}(T) - K)^+ \tag{6}$$

と近似する．ラフボラティリティモデルは過去のパスが影響するため通常のオイラー法を直接適用すると負荷が大きいが，ハイブリッドスキーム [6] を導入することで計算を効率化できる．

3.　ファイナンスにおけるキャリブレーション

キャリブレーションを定式化するため，パラメータ $\theta \in \Theta \subseteq \mathbb{R}^n$, $n \in \mathbb{N}$ によって定まるモデル $\mathcal{M}(\theta)$ を以下のように定義する．

$$\mathcal{M} := \{\mathcal{M}(\theta) \mid \theta \in \Theta\} \tag{7}$$

さらに，プライシングマップを以下のように定義する．

$$P : (\mathcal{M}(\theta), \zeta) \to \mathbb{R}^m \tag{8}$$

ここで，$m \in \mathbb{N}$ は金融商品の個数であり，$\zeta : C(\mathbb{R}) \to \mathbb{R}^m$ は与えられた満期・ストライク条件に対するオプションの集合を表し，その観測市場データを $\mathcal{P}^{MKT}(\zeta) \in \mathbb{R}^m$ とする．

しかし，実際のオプション価格付けモデルにおけるキャリブレーションでは，ほとんどの場合オプション価格 $P(\mathcal{M}(\theta), \zeta)$ に対する解析解は存在せず，MC シミュレーションなどの数値近似が不可欠となる．これにより，キャリブレーションは以下の近似最適化問題として表される．

近似キャリブレーション問題

$$\hat{\theta} = \underset{\theta\in\Theta}{\arg\min}\; d(\widetilde{P}(\mathcal{M}(\theta), \zeta), \mathcal{P}^{MKT}(\zeta)) \tag{9}$$

ここで，$\widetilde{P}$ は数値シミュレーションで得た価格近似値，$d(\cdot,\cdot)$ は平均二乗誤差 (MSE) などの誤差を表す指標である．

この近似キャリブレーション問題は実務上，大きな課題を伴う．前述したように，ラフボラティリティモデルのような複雑で非マルコフ的な確率構造を持つモデルでは，標準的な有限差分法や解析的アプローチが適用困難であり，キャリブレーションは莫大な計算負荷を伴うためである．ゆえに，高頻度でモデル更新が求められる実務（マーケットメイキングや日次・リアルタイムでのリスク報告など）において，キャリブレーションが重大なボトルネックとなる懸念があった．

こうした背景から，本研究では近年注目されている「ディープキャリブレーション」アプローチを採用する．これは MC シミュレーションによって生成した豊富なトレーニングデータ（パラメータと対応する，シミュレーションによって得たオプション価格）を用い NN を学習し，プライシングマップ P を高精度に近似する手法である．これにより，キャリブレーション段階では学習済み NN を用いて，市場データに合致するパラメータを勾配法などで迅速に算出できるようになり，複雑なモデルでもリアルタイムに近い速度でのキャリブレーションが可能となる．

(1)　ディープキャリブレーションの二段階アプローチ

ディープキャリブレーション手法に関し，Horvath ら [5] は二段階アプローチという一般的なフレームワークを提唱した．本研究では，それを参考に以下のステップでキャリブレーションスキームを構築した．

ステップ 1: モデルの学習

まず，確率モデル $\mathcal{M}(\theta)$ のパラメータ θ と対応するオプション価格 $P(\mathcal{M}(\theta), \zeta)$ を多数のサンプルとして生成する．これらは MC シミュレーションなどの数値手法によって得られる．続いて，θ を入力，価格 $P(\mathcal{M}(\theta), \zeta)$ を出力とする NN $F(\theta)$ を学習する．学習は以下の平均二乗誤差 (MSE) による損失関数を最小化する．

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\|F(\theta_i) - P(\mathcal{M}(\theta_i), \zeta_i)\|^2 \tag{10}$$

ここで，N はトレーニングサンプル数である．学習が完了すると，NN $F(\theta)$ は入力パラメータ θ に対する価格関数を高精度に近似することが可能となる．

ステップ 2: 市場データへのキャリブレーション

次に，市場から観測されたオプション価格データ $\mathcal{P}^{MKT}(\zeta)$ を用いて，学習済みの NN $F(\theta)$ によるキャリブレーションを行う．すなわち，以下の最適化問題を解く．

$$\hat{\theta} = \underset{\theta\in\Theta}{\arg\min}\; d(F(\theta), \mathcal{P}^{MKT}(\zeta)) \tag{11}$$

勾配に基づく最適化手法を用いることで，この問題は高速に解くことができる．さらには，最適化したパラメータを NN の入力とすることで，高速な価格評価を行うことができる．

4.　Differential Machine Learning（DML）

(1)　DML の概要

従来のディープラーニング手法では，入力のパラメータと出力（オプション価格）を対応付ける近似関数を学習する際，オプション価格データのみを用いて損失関数を最小化していた．

一方，Huge[7] らが開発した DML では，オプション価格データに加えて，各パラメータに関する勾配情報を同時に学習する．この勾配情報は，モデル近似関数にとって価格曲面の傾きを直接的に捉えるものであり，学習プロセスにおいて強力な補助情報となる．具体的には，パラメータベクトル $\theta \in \Theta \subseteq \mathbb{R}^n$ に対して，モデルを介して得られる価格関数 $P(\mathcal{M}(\theta), \zeta)$ のパラメータに関する勾配 $\frac{\partial P}{\partial \theta}$ を追加のデータとして用いて学習を行う．これにより，以下の利点が得られる．

- データ効率の向上: 各パラメータセットごとに微分ラベルを利用することで，事実上データセットの情報量が増大する．結果的に，オプション価格のみを学習する場合と比較して，同程度の近似精度に達するために必要なサンプル数が減少することが知られている [7, 10]．
- 形状認識能力の強化: ネットワークが価格曲面の勾配情報を直接取得するため，値から傾きへと豊富な特徴量が学習可能になる．これにより，より安定的かつ頑健な関数近似が可能となり，高次元・複雑なモデルに対してもロバストな性能を発揮すると考えられる．
- 正則化効果: 微分情報を利用することは，価格近似関数に対する有効な正則化手段として機能すると考えられる．これにより，過学習を抑制し，実務上求められる近似精度と汎用性のトレードオフを改善することが可能であると考えられる．

(2)　AAD による勾配の取得

DML 手法では，価格関数に関する勾配を効率的かつ正確に計算する必要がある．そこで，近年金融分野で注目されている AAD を用いる．AAD は複雑な関数の微分を，計算コストをほとんど増やさずに求める手法である．金融工学領域でも「Smoking Adjoints」[8] として知られ，利用が進んでいる．

特に，MC シミュレーションを用いてオプション価格を評価する際，パスワイズ微分（各シミュレーションパスごとの勾配情報）を AAD によって低コストで取得できる．これにより，DML ではペイオフ+勾配の拡張データセットを構築可能であり，NN がパラメータ空間における価格曲面を高精度で近似できるようになる．本研究では，AAD のライブラリとして，Pytorch を用いた実装を行った．

(3)　Twin-Network モデルによる実装

DML を実装する上で有用なアーキテクチャとして，Twin-Network モデルが提案されている [9]．Twin-Network は，オプション価格を予測するメインネットワークと，価格勾配を予測する補助的なネットワークを統合した構造を持ち，ペイオフと勾配情報の学習を同時に行う．

具体的には，θ を入力とし，オプション価格 $p = P(\mathcal{M}(\theta), \zeta)$ を出力するメインネットワークに加え，その勾配 $\frac{\partial p}{\partial \theta}$ を出力する補助ネットワークを併設する．訓練は，以下のような拡張損失関数によって行われる．

$$C = \alpha \underbrace{\frac{1}{m}\sum_{i=1}^{m}(p^{(i)} - \hat{p}^{(i)})^2}_{\text{MSE(価格誤差)}} + \beta \underbrace{\frac{1}{m}\sum_{i=1}^{m}\left\| \frac{\partial p^{(i)}}{\partial \theta} - \frac{\partial \hat{p}^{(i)}}{\partial \theta} \right\|^2}_{\text{MSE(勾配誤差)}} \tag{12}$$

α および β は価格誤差と勾配誤差の相対的な重みを調整するハイパーパラメータであり，λ とパラメータ空間の次元 n を用いて

$$\alpha = \frac{1}{1+\lambda n}, \quad \beta = \frac{\lambda n}{1+\lambda n} \tag{13}$$

と定義される [7]．λ は価格と勾配を組み合わせる際の基本的なスケーリング因子であり，例えば $\lambda = 1$ の場合，パラメータ数 n に応じて価格・勾配誤差のバランスが決まる．

ここで，$\hat{y}$ はモデルが予測するオプション価格，$\frac{\partial \hat{y}}{\partial \theta}$ はモデルが予測する勾配，λ は勾配誤差項に対する重み付けである．λ を大きく設定すれば，より微分情報に重きを置いた学習となり，反対に小さく設定すれば，従来の価格情報のみの学習に近づく．このようにすることで，モデルはオプション価格と勾配の両方を整合的に近似し，結果としてパラメータ空間全域で優れた汎化性能を獲得できる．

(4)　DML のラフボラティリティモデルへの応用

本研究では，DML を以下のフローで rBergomi モデルへのキャリブレーションに応用する．

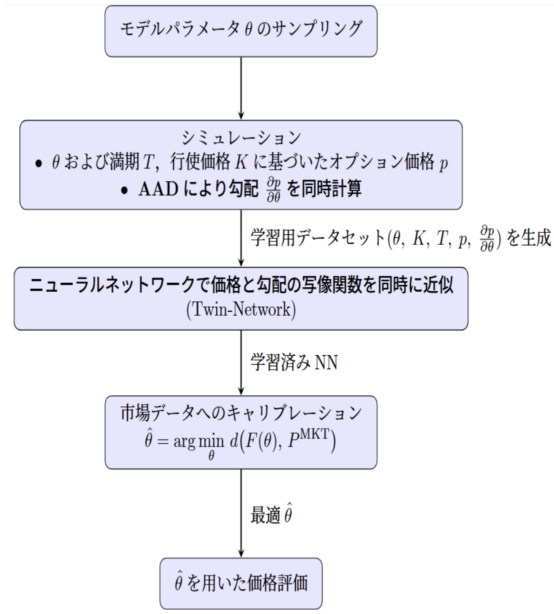

図 1 DML を用いたキャリブレーションのフロー

最初のステップでは，ラフボラティリティモデル（rBergomi）のパラメータ θ を広い範囲からサンプリングする．次に，MC シミュレーションによってオプション価格 p を計算する．同時に，AAD によってパラメータ θ に対する価格の勾配 $\frac{\partial p}{\partial \theta}$ を取得する．続いて，得られた $(\theta, K, T, p, \frac{\partial p}{\partial \theta})$ の組を学習用データセットとし，Twin-Network モデルを訓練する．価格と勾配の両方をラベルとして与えることで，価格曲面およびその勾配構造を同時に学習し，高い汎化性能を獲得することが期待される．学習が完了した段階で，市場から観測されるオプション価格データ $P_{MKT}^{K,T}$ に対して学習済みネットワークを用いたキャリブレーションを実施し，最適パラメータ $\hat{\theta}$ を高速に同定する．最後に，同定した $\hat{\theta}$ を用いて価格評価を行う．

5. 実証分析

rBergomi モデルのシミュレーションを用いて生成した学習データに対して DML を適用し，オプション価格のみを用いた標準的なディープニューラルネットワーク（DNN）モデルと，オプション価格＋勾配情報を用いた DML モデルを比較する．これにより，勾配情報の活用がキャリブレーション精度や計算効率にどの程度寄与するかを明らかにする．

(1) データ生成

まず，表 1に示すパラメータ範囲を設定し，1,000 組のパラメータ θ をランダムサンプリングした．各 θ について，ハイブリッドスキーム [6] を用いて MC シミュレーションを行い，ヨーロピアンコールオプション価格を計算した．さらに AAD により，H を除くパラメータに関して価格に対するパラメータ勾配 $\partial p/\partial\theta$ を取得した．

表 1 rBergomi モデルにおけるパラメータ範囲

パラメータ	範囲
ξ	[0.01, 0.16]
η	[0.5, 4.0]
ρ	[-0.95, -0.1]
H	[0.05, 0.5]
S_0	[10, 6000]

また，オプション価格はストライク K と満期 T の組み合わせを複数設定した．結果として合計 14 万超のオプションデータポイントを生成し，うち 20%をバリデーションデータとして用いた．

(2) 生成データの前処理

シミュレーション後の学習データには，(価格+勾配) の拡張ラベルが含まれる．前処理として，すべてのデータに関して標準化（平均 0，標準偏差 1）の処理を行った．

(3) ネットワークの構造

NN は 3 層の全結合ネットワークを用い，各層 64 ニューロン，活性化関数に Softplus を使用した．最適化アルゴリズムは Adam，学習率 0.0005，最大エポック 4000，バッチサイズ 1024 とし，アーリーストップを 100 で設定した．価格のみを予測する標準的な DNN モデルに加え，(1)DML_ALL（全パラメータ勾配を使用），(2) デルタ（S0 に関する勾配）のみを用いた DML_S0 の 2 種類を学習し比較を行った．

(4) キャリブレーション設定

学習後は，Yahoo Finance から取得したシカゴ・オプション取引所の S&P500 指数オプション（ヨーロピアンコール）データ[1]についてキャリブレーションを実施した．マルチスタート最適化を採用し，NN を用いた式 (11) の最適化問題を Adam で解く．最適化したパラメータで価格評価を行い，相対誤差 Mean Relative Error（MRE）と最適化から評価までにかかった計算時間を記録し，キャリブレーション性能を評価した．

(5) 学習結果および性能評価

(a) 学習曲線

図 2–4に，DNN，DML_ALL，DML_S0 の学習曲線（トレーニング損失とバリデーション損失）例を示す．

[1]2024 年 11 月 27 日に取得．行使価格は 20 から 1200，満期は 0.002 年から 2.054795 年の間に分布．

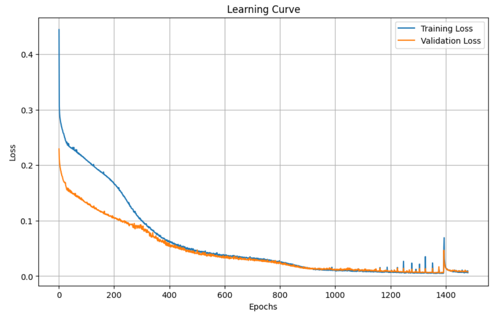

図 2 標準的な DNN モデル (勾配情報不使用) の学習曲線

DNN（図 2）は，Validation loss（Val loss）がある程度順調に下がるが，途中で過学習が生じて損失関数が跳ねている．最終的には Val Loss が 0.005 前後で収束した．

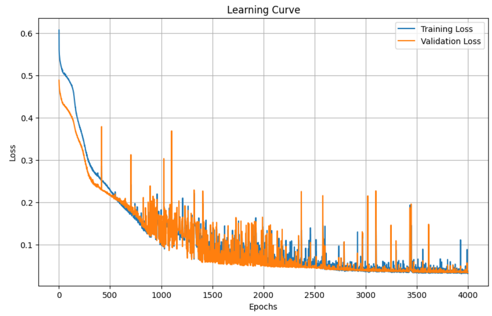

図 3 DML_ALL モデル (全パラメータ勾配利用) の学習曲線

DML_ALL（図 3）は価格＋全パラメータ勾配の同時フィットである．学習の挙動として，時折 Val loss が大きく変動しており，不安定性が見られる．最終的に Val Loss 約 0.033 となった．

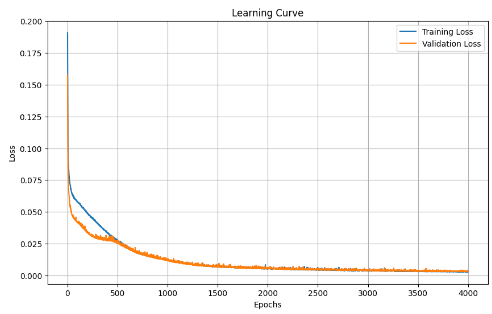

図 4 DML_S0 モデル (デルタのみ利用) の学習曲線

DML_S0（図 4）はデルタのみ勾配情報を用いているが，学習中の Val Loss の挙動は最も安定している．過学習は見られず，最終的には Val Loss が 3 モデル中最も低い値 (約 0.003 程度) で学習が終了した．

(b) 市場データへのキャリブレーション結果

次に，実際の S&P500 コールオプションに対するキャリブレーションを行った．表 2に，各モデルの相対誤差 MRE(%) とキャリブレーション時間 (秒) を示す．満期の使用数（Num

Maturities）を 1，5，10，全て（“all”）の 4 パターンで比較した．

表 2 キャリブレーション結果（MRE (%), Time (s)）

モデル	満期数	MRE (%)	時間 (s)
Model_DNN	1	279.962	4.042
Model_DML_ALL	1	519.577	7.405
Model_DML_S0	1	21.483	6.071
Model_DNN	5	78.607	8.224
Model_DML_ALL	5	501.913	4.950
Model_DML_S0	5	76.562	4.285
Model_DNN	10	56.261	6.107
Model_DML_ALL	10	477.422	5.191
Model_DML_S0	10	56.310	24.048
Model_DNN	all	216.870	3.291
Model_DML_ALL	all	459.190	7.207
Model_DML_S0	all	66.156	9.598

Num Maturities=1 では DML_S0 のみが 21.48%と低い誤差を記録した．また，Num Maturities=all の場合も DNN や DML_ALL は誤差が非常に大きい．満期を増やした場合でも DML_S0 は良好な精度を保っており，DML 手法の有効性が伺える．しかし，DML_ALL はいずれの場合も 400%超えの MRE を示し不安定であった．

キャリブレーション後の価格予測と残差分布

さらに，図 5–図 8に，キャリブレーション後の予測価格 対 実際価格（左図）および残差 (予測 – 実際) 対 ストライク（右図）を示す．

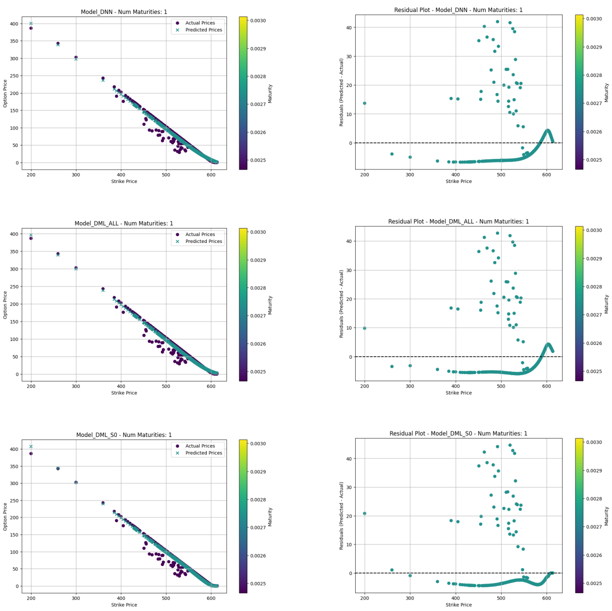

図 5 満期 1 種類を使用したキャリブレーション結果（上段：DNN，中段：DML_ALL，下段：DML_S0）

図 5（Num Maturities=1）の場合，DML_S0 の残差（下段）が最も小さく分布し，実際の価格とのフィットが良好である．一方，DNN（上段）と DML_ALL（中段）は 500〜600 付近のインザマネー領域での乖離が大きい．

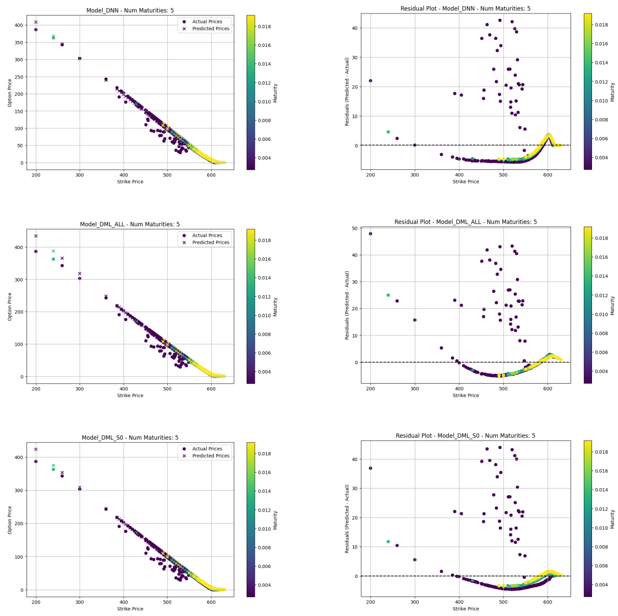

図 6 満期 5 種類を使用したキャリブレーション結果（上段：DNN，中段：DML_ALL，下段：DML_S0）

図 6（Num Maturities=5）および図 7（Num Maturities=10）において，DNN（上段）は全体的に良好なフィットを示した．DML_ALL（中段）は 400〜600 付近の領域で大きな残差が見られる．DML_ALL（下段）は全体的に良好なフィットを示し，特にアットザマネーの領域でのフィットが良好である．

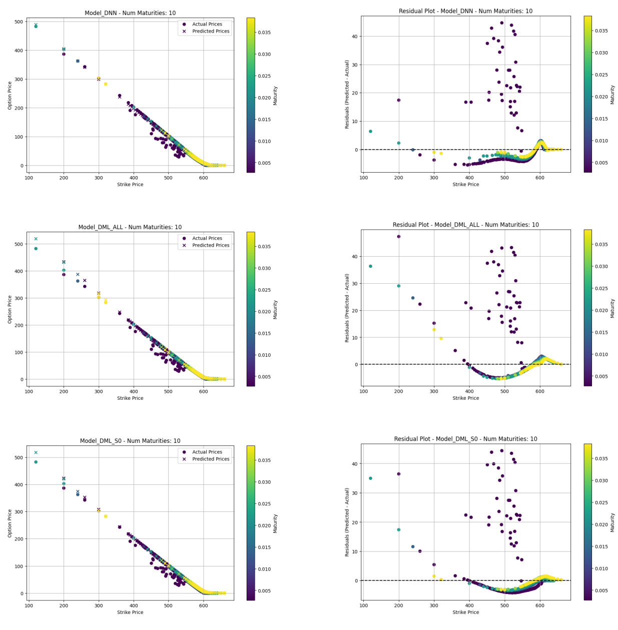

図 7 満期 10 種類を使用したキャリブレーション結果（上段：DNN，中段：DML_ALL，下段：DML_S0）

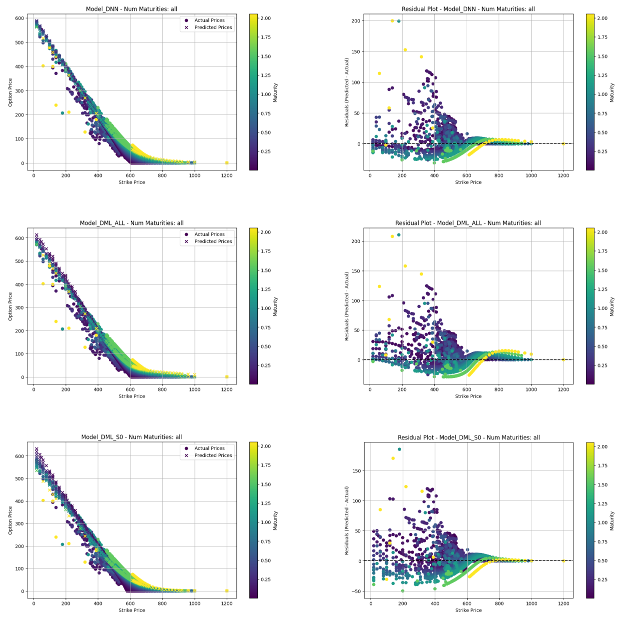

図 8 利用可能な満期全てを使用したキャリブレーション結果（上段：DNN，中段：DML_ALL，下段：DML_S0）

図 8（Num Maturities=all）では，アウトオブザマネーのオプション価格が低い領域で DML_ALL の残差がゼロ付近に集中している．結果として，DML_S0 が 3 モデル中最も低い相対誤差を達成している．

最適パラメータ

表 3に，キャリブレーションで得られた最適パラメータを列挙した（$\theta = (\xi, \eta, \rho, H)$ の順）．数値には学習上のランダム初期値や局所解の影響があるが，DML_S0 は負の ρ を大きめにとる傾向があり，レバレッジ効果や短期ボラティリティ変動を強く評価する形で市場価格にフィットしている様子がうかがえる．

表 3 キャリブレーションによる最適パラメータ

モデル	満期数	パラメータ
Model_DNN	1	[0.039, 1.764, -0.100, 0.187]
Model_DML_ALL	1	[0.010, 2.706, -0.343, 0.310]
Model_DML_S0	1	[0.143, 2.545, -0.723, 0.272]
Model_DNN	5	[0.010, 1.712, -0.100, 0.204]
Model_DML_ALL	5	[0.010, 3.420, -0.100, 0.063]
Model_DML_S0	5	[0.034, 4.000, -0.950, 0.114]
Model_DNN	10	[0.026, 2.486, -0.266, 0.053]
Model_DML_ALL	10	[0.010, 3.401, -0.100, 0.065]
Model_DML_S0	10	[0.038, 4.000, -0.950, 0.101]
Model_DNN	all	[0.089, 1.207, -0.514, 0.043]
Model_DML_ALL	all	[0.010, 2.646, -0.160, 0.117]
Model_DML_S0	all	[0.053, 0.500, -0.876, 0.500]

6. 考察

実証分析の結果を基に，学習の安定性やキャリブレーション精度，実務適用性などの観点から考察を行う．とりわけ，本研究で導入した Differential Machine Learning（DML）の有効性と課題点を整理し、今後の発展可能性について論じる．

(1) 勾配情報活用によるキャリブレーション精度の向上

DML_S0 モデルでは，デルタという 1 種類の勾配情報しか利用していないにもかかわらず，満期数が増えても安定的に低い誤差で市場価格にフィットできることが分かった．学習曲線も DNN モデルに比べ安定し，バリデーション損失の低下も顕著である．これは，原資産価格に対する感応度情報が価格曲面の勾配を捉える上で強力に機能しているためであると考えられる．

(2) 全パラメータ勾配の不安定性

理論上はパラメータ勾配を多く使うほど近似が強化される可能性があるが，本研究での DML_ALL は満期数が増えるほど誤差が大きく，かつ学習が不安定であった．これは，勾配損失と価格損失の重み付けやスケーリングが難しいことが要因と推察される．勾配情報の過度な制約が，かえって局所解へ導いたり，学習収束を阻害する可能性があると考えられる．

(3) 実務的意義と今後の展望

デルタのみの勾配を併用する DML_S0 が高精度かつ計算負荷も比較的低く抑えられることは，実務におけるラフボラティリティモデル導入を容易にする．実際のオプション市場では，市場データの質や流動性が銘柄・ストライク帯によって異なるため，全領域での高精度フィットが必要な場合や特定ストライク帯のみ重視する場合など状況は様々だが，DML_S0 は十分汎用的に適用可能と考えられる．

一方，DML_ALL は，改良により本来の優位性（価格曲面をあらゆる方向から補足する）を活かすことができれば，さらに安定的で高精度なキャリブレーションを実現できる可能性がある．今後は正則化やネットワーク構造の見直しによる安定化が展望として挙げられる．

7. 結論

本研究では，ラフボラティリティモデルの一種である rBergomi モデルに対し，DML 手法を適用した高速かつ高精度なキャリブレーション手法を構築した．

実験の結果，原資産価格に対する勾配（デルタ）のみを併用する DML_S0 が安定かつ高精度に市場オプション価格へフィットし，既存の DNN モデルやすべての勾配を使う DML_ALL モデルよりも優れた性能を示した．これはデルタ情報が価格曲面の形状学習を強力に補助することを意味し，実務的に計算コストと精度のバランスを最適化する上で有用である．

今後は，DML_ALL を安定的に運用するための正則化やネットワーク設計の改善，市場流動性に応じた重み付けなどの応用的課題が残る．それでもなお，DML はラフボラティリティモデル下のキャリブレーションを大幅に効率化し，短期変動を強く意識したリスク管理・オプション取引におけるリスクを低減できる有力手段であると考えられる．

参考文献

[1] Gatheral, J., Jaisson, T., & Rosenbaum, M. (2018). Volatility is rough. *Quantitative finance*, 18(6), 933-949.

[2] Bayer, C., Friz, P., & Gatheral, J. (2016). Pricing under rough volatility. *Quantitative Finance*, 16(6),

887-904.

[3] Hernandez, A. (2017). Model calibration with neural networks. *Risk*.

[4] Bayer, C., Horvath, B., Muguruza, A., Stemper, B., & Tomas, M. (2019). On deep calibration of (rough) stochastic volatility models. arXiv preprint arXiv:1908.08806.

[5] Horvath, B., Muguruza, A., & Tomas, M. (2021). Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models. *Quantitative Finance*, 21(1), 11-27.

[6] Bennedsen, M., Lunde, A., & Pakkanen, M. S. (2017). Hybrid scheme for Brownian semistationary processes. *Finance and Stochastics*, 21, 931-965.

[7] Huge, B., & Savine, A. (2020). Differential machine learning. arXiv preprint arXiv:2005.02347.

[8] Giles, M., & Glasserman, P. (2006). Smoking adjoints: Fast monte carlo greeks. *Risk*, 19(1), 88-92.

[9] Huge, B. N., & Savine, A. (n.d.). Differential machine learning notebooks. Retrieved from `https://github.com/differential-machine-learning/notebooks/blob/master/DifferentialML.ipynb`

[10] Sridi, A., & Bilokon, P. (2023). Applying Deep Learning to Calibrate Stochastic Volatility Models. arXiv preprint arXiv:2309.07843.