サーベイ論文

機械学習におけるハイパパラメータ最適化手法：概要と特徴

尾崎　嘉彦[†,††]　　野村　将寛[†,†††]　　大西　正輝[†a)]

Hyperparameter Optimization Methods: Overview and Characteristics

Yoshihiko OZAKI[†,††], Masahiro NOMURA[†,†††], and Masaki ONISHI[†a)]

あらまし　ハイパパラメータ最適化は，機械学習モデルのチューニングを自動化するための実用的な技術である．本論文は，ハイパパラメータ最適化に関心のある周辺分野の研究者及び，それを実務に活用しようとするエンジニアに向けた，ハイパパラメータ最適化手法の実用に焦点を当てたサーベイである．本サーベイの目的は，ハイパパラメータ最適化手法を概説し，各手法のもつ特徴や適切な使い分けについて整理し，見通しの良い形で読者に知識を共有することである．本サーベイは，序章と終章を除いて四つの節から構成される．まずはじめに，**2.** においてハイパパラメータ最適化の基礎知識について解説する．その後，**3.** でハイパパラメータ最適化において標準的であるブラックボックス最適化，**4.** で近年のトレンドであるグレーボックス最適化について順に解説する．最後に，**5.** では逐次評価回数の上限値，並列計算リソース，ハイパパラメータの種類の観点から，状況ごとに個別に議論を行い，適切な最適化手法選択のガイドラインを与える．

キーワード　ハイパパラメータ最適化，ブラックボックス最適化，グレーボックス最適化，機械学習

1. まえがき

一般物体認識の精度を競い合うコンペティションである ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 での AlexNet [1] の優勝を皮切りに，機械学習モデル，特に深層ニューラルネットワークの性能は改善の一途を辿り，先に挙げた一般物体認識をはじめ，画像生成 [2]，ゲーム AI [3] など極めて幅広い分野で次々とブレークスルーを巻き起こした．性能改善と歩調を合わせるようにして，インセプションモジュールと呼ばれる構造をもつモデル [4] や Residual Units と呼ばれる構造をもつ非常に深いモデル [5] が続々と出現するなど，モデル構造も複雑化し，それらのチューニングや設計は従来以上に高度な知識と経験が要求される作業へと変貌した．

一般に，機械学習モデルの性能を十分に引き出すためには，モデルのハイパパラメータを適切に設定する必要がある [6]．最近では，ACM Conference on Recommender Systems (RecSys) 2019 のベストペーパーに選出された文献 [7] において，十分にチューニングされた古典的な手法が近年提案された多くの深層ニューラルネットワークをベースとした手法を上回る性能を達成したことで，チューニングの重要性が再認識されることとなった．

このような背景から，近年，手作業に代わり，最適化アルゴリズムを用いて機械学習モデルをチューニングするハイパパラメータ最適化 [6] が盛んに研究されている．ハイパパラメータ最適化は，既にチューニング能力において専門家を上回っており [8], [9]，先の文献 [7] でも利用されている．また，計算機に作業を肩代わりさせることで，人の労力や苦痛を軽減する．更に，均質化された水準のチューニングを行えるため，研究の公平性改善にも貢献する [6] など，数多くの利点をもつ．

本論文は，ハイパパラメータ最適化の実用に焦点を当てたサーベイである．想定読者は，ハイパパラメータ最適化に関心のある周辺分野の研究者，及び，それを実務に活用したいエンジニアである．本論文では，1) 重要なハイパパラメータ最適化手法であるブラック

† 産業技術総合研究所人工知能研究センター，東京都
Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Koto-ku, Tokyo, 135–0064 Japan
†† グリー株式会社，東京都
GREE, Inc., Minato-ku, Tokyo, 106–0032 Japan
††† 株式会社サイバーエージェント，東京都
CyberAgent, Inc., Shibuya-ku, Tokyo, 150–0042 Japan
a) E-mail: onishi@ni.aist.go.jp
DOI:10.14923/transinfj.2019JDR0003

表 1　ハイパパラメータの分類
Table 1　The taxonomy of hyperparameters.

分類	概要	例
連続	連続値を取るハイパパラメータ	勾配降下法の学習率，ドロップアウト率
離散	離散値を取るハイパパラメータ	全結合層のユニット数
カテゴリー	量的でない値を取るハイパパラメータ	使用する活性化関数，使用するカーネル
条件	特定条件下でのみ有効な値となる上記 3 種類のハイパパラメータ	特定のカーネルのみがもつハイパパラメータ

ボックス最適化手法とグレーボックス最適化手法について概説し，2) 各手法がもつ特徴を見通しの良い形で整理し，3) 適切なハイパパラメータ最適化手法を選択するための実用的なガイドラインを与える．本論文の執筆時点において，筆者らの知る限りでは，最新のグレーボックス最適化手法を含むハイパパラメータ最適化手法に関する網羅的な日本語情報は存在しない．また，ハイパパラメータ最適化手法選択の実用的なガイドラインも，新しい試みである．

一方，本論文では，既存のサーベイ論文 [6], [10] で扱われている研究の歴史，適用事例や，各種文献 [11]〜[14] で解説されている個別の最適化アルゴリズムの詳細は扱わない．各節において，関連する参考文献を示すので，必要に応じて参照して欲しい．

本論文の構成は以下のとおりである．まず，**2.** においてハイパパラメータ最適化問題の定式化，問題特徴，ハイパパラメータ最適化手法に望まれる性質，手法の分類について述べる．続く **3.**，**4.** では，ハイパパラメータ最適化において標準的であるブラックボックス最適化，近年のハイパパラメータ最適化手法研究におけるトレンドであるグレーボックス最適化について，代表的な手法を概説した上，特徴を整理する．最後に **5.** で，逐次評価回数の上限値，並列計算リソース，ハイパパラメータの種類の観点から，状況ごとに個別に議論を行い，最適化手法選択のガイドラインを与える．

2. ハイパパラメータ最適化

いま，ある機械学習モデルが与えられており，このモデルのチューニングを行うとする．具体的なモデルとしては，サポートベクトルマシン (Support Vector Machine, SVM)，深層ニューラルネットワーク，k-近傍法 (k-Nearest Neighbor, k-NN) などが想定される．モデルがもつチューニング可能な n 個のハイパパラメータの定義域を X_i $(i = 1, \ldots, n)$ とする．各ハイパパラメータは，典型的には表 1 に示すような，連続，離散，カテゴリー，条件パラメータの 4 種類に分類でき，一般に，ハイパパラメータ設定の探索空間 $\mathbf{X} = X_1 \times \cdots \times X_n$ は，様々な種類のハイパパラメータを含む．更に，$f : \mathbf{X} \to \mathbb{R}$ をモデルの性能を示す損失関数とする．具体的な損失関数としては，検証データセットに対するモデルの誤識別率や Root Mean Square Error (RMSE) などが考えられる．このとき，ハイパパラメータ最適化は，モデルが最良性能を達成するハイパパラメータ設定を見つけ出す，次のような最小化問題として定式化できる：

$$\begin{aligned} &\text{minimize} \quad f(\boldsymbol{x}), \\ &\boldsymbol{x} \in \mathbf{X}. \end{aligned} \tag{1}$$

例として，一般物体認識を行う深層ニューラルネットワークのチューニングとすれば，問題 (1) はより具体的に，以下のように記述できる：

$$\begin{aligned} &\text{minimize} \quad f(\boldsymbol{x}; \boldsymbol{w}^*, D_{\text{valid}}), \\ &\text{subject to} \quad \boldsymbol{w}^* = \text{argmin}_{\boldsymbol{w}}\, g(\boldsymbol{x}, \boldsymbol{w}; D_{\text{train}}), \\ &\boldsymbol{x} \in \mathbf{X}. \end{aligned} \tag{2}$$

ここで，D_{valid} は検証データセット，f はモデルの性能を示す損失関数（すなわち，検証データセットに対する誤識別率），$\boldsymbol{w}$ はモデルの重み，D_{train} は学習データセット，g は学習に用いる損失関数（典型的にはクロスエントロピー誤差）である．このとき，問題 (2) の最適解は，モデルが検証データセットに対して最小の誤識別率を達成するようなハイパパラメータ設定となる．

2.1 問題特徴と最適化手法に望まれる性質

ハイパパラメータ最適化問題は以下の四つの性質をもつ．

一つめは，目的関数の評価コストの高さである．ハイパパラメータ最適化問題の目的関数は，先ほど述べたとおり，機械学習モデルの性能を示す損失関数である．よって，目的関数を評価するために，モデルの学習が必要となる．しかし，複雑な深層ニューラルネットワークなどのモデルは，学習に数時間から数週間を必要とすることも少なくなく，極めて計算コストが高

表 2　ハイパパラメータ最適化手法の分類
Table 2　The taxonomy of hyperparameter optimization methods.

分類	概説
ブラックボックス最適化手法	目的関数値のみを利用して最適化する手法．モデルの詳細や勾配情報に依存しないため適用範囲が広い．現在の主流．
グレーボックス最適化手法	対象問題の特徴から得られる最適化に有益な補助情報を活用し，ブラックボックス最適化手法を高速化した手法．ブラックボックス最適化手法の次のトレンドとして近年研究が盛んであり，発展が著しい．
その他	勾配法や強化学習 [15]〜[17] などの適用事例がある．普及はしていない．

い．このため，ハイパパラメータ最適化においては，多くの場合，目的関数の評価が実行時間におけるボトルネックとなり，限られた回数の直列評価しか実用上許容できない．ゆえに，ハイパパラメータ最適化手法には，限られた目的関数評価から得られる情報を最大限に活用すること，及び評価の並列化に適することが望まれる．

二つめは，探索空間の複雑性である．SVM や深層ニューラルネットワークなど比較的よく用いられる機械学習モデルは，数個から数十個ほどのハイパパラメータをもつため，ハイパパラメータ最適化問題は典型的には数十次元程度までの探索空間を扱う．そして，探索空間は表 1 に示したような，異なる種類のハイパパラメータが組み合わさったものである．ゆえに，ハイパパラメータ最適化手法は，一般的な連続最適化や組合せ最適化を対象とした手法に比べ，はるかに複雑な探索空間を扱えることが望まれる．

三つめは，目的関数の実効的な次元数の低さである．Bergstra と Bengio は文献 [18] において，機械学習モデルがもつハイパパラメータ全体のうち，性能に関して重要なものがごく一部しかないことを計算実験により発見し，そのような性質を Low Effective Dimensionality (LED) と呼んでいる．また，Hutter らも functional ANOVA を用いてハイパパラメータ最適化の結果について分析を行い，ごく一部の重要なハイパパラメータの影響によって，性能の変化の大部分を説明できたことを報告している [19]．更に，van Rijn らも functional ANOVA を用いて 100 種類のデータセットに対して分析を行い，SVM，ランダムフォレスト，AdaBoost の 3 種類のモデルについて，重要なハイパパラメータが多くのデータセット間で共通していたことを報告している [20]．このように，ハイパパラメータ最適化問題における各ハイパパラメータの重要度は，多くの場合，偏っていることが知られている．ゆえに，ハイパパラメータ最適化手法は，LED に強いことが望まれる．LED の影響については，後ほど **3.1** と **3.2** において具体例を与え，詳しく説明する．

四つめは，目的関数が確率的なことである．モデルの学習は，その過程に乱数による重みの初期化，学習バッチのランダムサンプリング，結合のドロップアウトなど確率的な処理を含みうる．このため，ハイパパラメータ設定の良し悪しについて，確率的に過大評価や過小評価が起こるリスクが存在する．ゆえに，ハイパパラメータ最適化手法は，目的関数評価における不確実性を扱えることが望まれる．

上記四つの性質に関する議論から導かれた，ハイパパラメータ最適化手法に望まれる性質は以下である．

（1）　目的関数評価から得られる情報を活用する．
（2）　並列化に適する．
（3）　複雑な探索空間を扱える．
（4）　LED に強い．
（5）　目的関数評価の不確実性を扱える．

これらの性質を全て高い水準で満たすことは難しいため，異なる性質を備えた多くの手法が提案されている．

2.2　ハイパパラメータ最適化手法の分類

機械学習モデルの学習では，勾配法を用いて損失関数を最適化することが一般的である．一方，ハイパパラメータ最適化では，ハイパパラメータに対する勾配計算が大変なことや，微分を不可能とする非連続な損失関数，離散，カテゴリー，条件パラメータの存在性から，勾配法の適用は限定的である [21]〜[23]．代わりに，実用上成功を収めているのは，幅広く適用が可能なブラックボックス最適化手法や，対象問題の特徴から得られる最適化に有益な補助情報を活用して高速なチューニングを実現するグレーボックス最適化手法である [6]．ハイパパラメータ最適化手法の分類を，表 2 に示す．

本論文では，続く二つの節で，ハイパパラメータ最適化の実用において重要である，ブラックボックス最適化，及びグレーボックス最適化の代表的な手法を紹介する．また，各手法の特徴を **2.1** で挙げた，手法に

望まれる性質の観点から整理する．

3. ブラックボックス最適化

ブラックボックス最適化問題は，目的関数や制約がブラックボックスとして与えられるような最適化問題である [14]．より具体的には，ブラックボックス最適化問題では，目的関数や制約の関数形が不明であり，勾配情報などの目的関数値以外の最適化に有用な情報が利用できない．このような問題を解くためのブラックボックス最適化手法は，実行に目的関数値以外の情報を必要とせず，適用範囲が広い．ハイパパラメータ最適化問題も損失関数をブラックボックスと考えれば，ブラックボックス最適化問題とみなすことができる．そして現在，ハイパパラメータ最適化を解くための最も標準的な方針は，ブラックボックス最適化手法を用いることである．

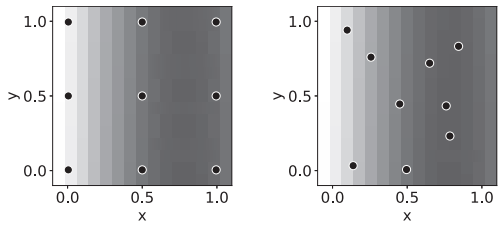
本節では，代表的なブラックボックス最適化手法について述べる．

3.1 グリッドサーチ

グリッドサーチは，各ハイパパラメータに対して幾つかの代表値を選択し，ハイパパラメータ設定の探索空間をその直積としその空間を全探索する．この手法は，人間にとって直感的であり，機械学習コミュニティにおいて広く利用される．

グリッドサーチは，あらかじめ選択した代表値に基づき評価されるハイパパラメータ設定が確定するため，全ての目的関数評価を非同期に並列化できる．また，離散，カテゴリー，条件パラメータを扱える．一方，探索の途中で目的関数評価から得られる情報は一切活用できない．

図 1 に示した関数 $f(x,y) = (x-\frac{3}{4})^2 + \frac{y}{100}$ は探索空間 $[0,1]^2$ において LED をもち，変数 x が目的関数値に対して支配的である．すなわち $g(x) = (x-\frac{3}{4})^2$, $h(y) = \frac{y}{100}$ として，$f(x,y) = g(x)+h(y) \approx g(x)$ が成り立つ．このような状況では，実用上，x の値が異なる評価だけが探索に役立つ．ところが，グリッドサーチは，x が同様な設定の評価を繰り返してしまう．例えば，図 1 (a) では，$f(0.0,0.0) \approx f(0.0,0.5) \approx f(0.0,1.0)$, $f(0.5,0.0) \approx f(0.5,0.5) \approx f(0.5,1.0)$, $f(1.0,0.0) \approx f(1.0,0.5) \approx f(1.0,1.0)$ となるため，9 回の評価のうち 6 回は無駄である．また，一般には実効的でないパラメータの数に対して，無駄な評価の数は指数オーダーで増加する．これは，グリッドサーチが LED に弱い [18] ことを示している．



(a) グリッドサーチ　(b) ランダムサーチ

図 1 グリッドサーチとランダムサーチを用いて LED をもつ関数 $f(x,y) = (x-\frac{3}{4})^2 + \frac{y}{100}$ を最適化した例

Fig. 1 An example of optimization of $f(x,y) = (x-\frac{3}{4})^2 + \frac{y}{100}$ using grid search and random search.

3.2 ランダムサーチ

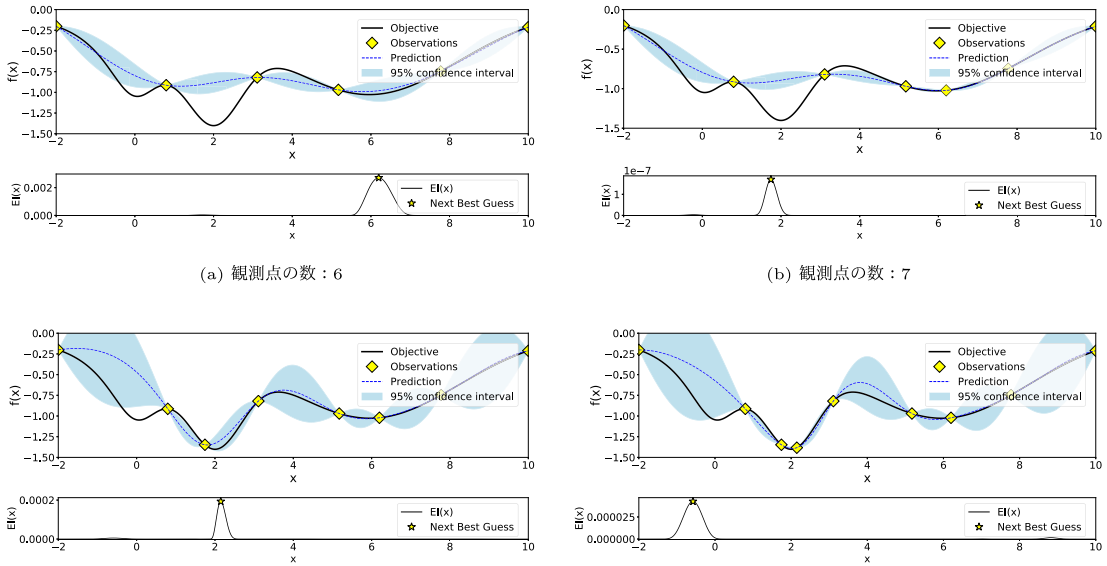
ランダムサーチは，乱数生成器を用いてハイパパラメータ設定を生成する．

ランダムサーチは，乱数に基づいて評価されるハイパパラメータ設定が確定するため，全ての評価を非同期に並列化可能である．また，連続，離散，カテゴリー，条件パラメータを全て含む探索空間を扱える．一方，探索の途中で目的関数評価から得られる情報は一切活用できない．

ランダムサーチの重要な特徴は，グリッドーサーチと比較して，LED に強いことである [18]．さきほど，図 1 (a) を例に，グリッドサーチは x が同様な設定の探索を繰り返すため，LED に弱いことを述べた．一方，ランダムサーチは常に全てのハイパパラメータの値をランダムに決定する．このため，図 1 (b) では，9 回の評価全てにおいて x が変化している．よって，ランダムサーチは，実効的でないパラメータの存在によって，無駄な目的関数評価を生じない．

また，ランダムサーチは，任意の時点で実行を打ち切ったり，独立に実行した複数のランダムサーチの結果を混ぜ合わせても，ランダムサーチとして成立する，といったグリッドサーチに対しては成り立たない，実用上優れた性質ももつ [6]．

ランダムサーチの亜種として，ラテン超方格サンプリング (Latin hypercube sampling, LHS) [24] や Sobol 列 [25] などの低食い違い量列（low discrepancy sequence）を用いたサンプリングが提案されている [18]．一様乱数を用いたサンプリングでは，しばしばよく似たハイパパラメータ設定が複数回サンプルされる．対して，低食い違い量列を用いると，おおまかに言えば，互いに似過ぎていないハイパパラメータ設定をサンプルできる（図 2）．Bergstra と Bengio は，Sobol 列が一

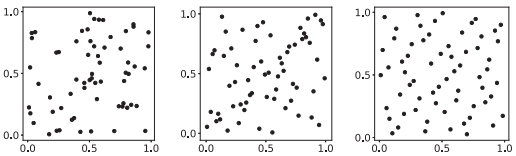


(a) 観測点の数：6　(b) 観測点の数：7

(c) 観測点の数：8　(d) 観測点の数：9

図 3　ベイズ最適化で $f(x) = -\mathrm{e}^{-(x-2)^2} - \mathrm{e}^{-\frac{(x-6)^2}{10}} - \frac{1}{(x^2+1)}$ を最小化した例．反復的に，観測データから代理モデルを構築し獲得関数値が最大となる点を評価する．

Fig. 3　An example of optimization of $f(x) = -\mathrm{e}^{-(x-2)^2} - \mathrm{e}^{-\frac{(x-6)^2}{10}} - \frac{1}{(x^2+1)}$ using Bayesian optimization. The algorithm iteratively constructs a surrogate and evaluates the maximizer of an acquisition function.



(a) 一様ランダム列　(b) LHS　(c) Sobol 列

図 2　一様ランダム列と低食い違い量列からのサンプル

Fig. 2　Samples from uniform random and low discrepancy sequences.

様乱数よりも高い探索性能を示すことを実験的に明らかにしている [18]．ただし，ラテン超方格サンプリングや Sobol 列は，連続パラメータしか扱えないことに注意する．

ランダムサーチに関する参考文献として，ランダムサーチが機械学習モデルのチューニングにおいて専門家を上回ることを示した文献 [18] がある．また，LED に関する議論や低食い違い量列の利用提案も，この文献で行われたものである．

3.3　ベイズ最適化

ベイズ最適化は，ハイパパラメータ最適化において，最も成功を納めているブラックボックス最適化手法である．この手法は，確率的な代理モデルと獲得関数の二つの要素からなる．代理モデルは，観測データ（ハイパパラメータ設定と目的関数値のペアデータ）から，評価コストが高い目的関数や有望なハイパパラメータ設定の分布を近似するために用いられる．獲得関数は，効率的に有望なハイパパラメータをサンプルするための指標である．ベイズ最適化は，次の手順を繰り返す．

（1）　観測データから代理モデルを構築，更新する．
（2）　構築した代理モデルから獲得関数を計算し，その獲得関数を最大化することで，次に評価するハイパパラメータ設定を選択する．
（3）　選択したハイパパラメータ設定を評価する．

通常，最初に幾つかの観測データを集めるために，ランダムサーチを行う．図 3 に，ベイズ最適化の反復の実行例を示した．

ベイズ最適化は，目的関数評価から得られる情報を代理モデルを構築するために活用している．代理モデルの構築と次に評価するハイパパラメータ設定の選択

は交互に行われるため，目的関数評価は逐次的となる．このため，この手法は原理的に並列化に適さない．ただし，実用上重要であるため，並列化の試みは数多く行われている [8], [9], [26]〜[32].

3.3.1 代理モデル

ハイパパラメータ最適化において最も代表的な代理モデルは，ガウス過程 (Gaussian Process, GP) [33] である．ガウス過程は，連続関数に対する確率過程であり，出力値の平均を定める平均関数と，2 点間の距離を定めるカーネルによって定義される．ハイパパラメータ最適化では，ガウス過程を用いて，観測データから目的関数をモデル化する．この代理モデルは，ベイズ的に観測ノイズをモデル化することで，目的関数評価の不確実性を扱うことができる．図 3 の各グラフの上側のサブプロットは，各状態における真の目的関数，観測済みの点，ガウス過程による目的関数の予測と 95% 信頼区間を可視化したものである．観測点の付近では予測はおおむね正確であり，反対に周辺が未探索の領域では情報が不足しているため 95% 信頼区間が広がっていることが確認できる．ガウス過程の予測性能や，扱うことができるパラメータの種類は，カーネルに依存する．文献 [9] では，ハイパパラメータ最適化に適しているとして，Matérn カーネルの利用が推奨されている．一方，カテゴリーパラメータや条件パラメータを扱うためには，それらに対して適切に距離を定めるカーネルが必要である [34]〜[36]．ガウス過程は優れた性能をもつ代理モデルであるが，計算量が大きい（観測点の数の 3 乗オーダー）という欠点があり，観測データが多い場合には，実行時間が問題となる．そのような場合，ガウス過程の近似 [37], [38] や他の代理モデルが利用される．

観測データから目的関数をモデル化するための代理モデルとして，ランダムフォレストが用いられることもある [39]．ランダムフォレストは，カテゴリーパラメータを自然に扱える．また，ガウス過程と比較して計算量が小さいため，比較的観測点の数が多い場合も扱える．

Tree-structured Parzen Estimator (TPE) [8], [40] と呼ばれる，カーネル密度推定に基づく代理モデルが用いられることもある．この代理モデルは，観測データから目的関数をモデル化するのではなく，有望なハイパパラメータ設定の分布を近似する．また，連続，離散，カテゴリーパラメータを容易に扱える．ガウス過程と比較して計算量も小さく，観測点の数が多い場合も問題がない．

このほかに，代理モデルとして，深層ニューラルネットワーク [41] や Neural Process が利用されることもある [42].

3.3.2 獲 得 関 数

ハイパパラメータ最適化において最も代表的な獲得関数は，Expected Improvement (EI) [43] である．EI はあるしきい値（典型的には既知の最良の目的関数値）に対する，ハイパパラメータ設定 $\boldsymbol{x}$ における $f(\boldsymbol{x})$ の改善量の期待値を表す指標である．EI の具体的な計算方法は，ベイズ最適化に用いる確率的な代理モデルに依存する [8], [9]．図 3 の各グラフの下側のサブプロットは，各状態における EI を可視化したものである．既に良い目的関数値を達成している点の付近（有望な点である可能性が高いと考えられる領域）や周辺が未探索の領域（情報が少ないため積極的に探索すべきであると考えられる領域）で EI が大きな値を取っていることが分かる．ベイズ最適化では，各反復において獲得関数を最大化する点を，次の評価点として選択する．ここで，獲得関数の評価は，ハイパパラメータ最適化問題の目的関数の評価に比べ，評価コストが極めて低いことに注意する．

現在，ハイパパラメータ最適化において最も主要なベイズ最適化のアルゴリズム（**3.3.3**）は，いずれも獲得関数として EI を採用しているものの，EI 以外にも，Probability of Improvement (PI) [44]，Upper Confidence Bound (UCB) [45]，Predictive Entropy Search (PES) [46] など，幾つかの獲得関数が提案されている．

3.3.3 主要なベイズ最適化のアルゴリズム

ベイズ最適化には，用いる代理モデルと獲得関数の組み合わせによって，幾つかの種類が存在する．ここでは，ハイパパラメータ最適化において主要なものを取り上げる．

ハイパパラメータ最適化に用いられるベイズ最適化として，最も標準的なものは，代理モデルとしてガウス過程，獲得関数として EI を用いる手法 [8], [9]（便宜上 GP-EI と呼ぶ）である．GP-EI は，代理モデルとしてガウス過程を用いるため，目的関数評価の不確実性を加味した関数の予測を行い，次の評価点を選択できる．一方，観測データが多い場合には，実行時間が問題となる．また，内部で行う獲得関数の最大化のために，毎反復非凸大域的最適化を行う必要がある [8], [11].

Sequential Model-based Algorithm Configuration (SMAC) [39] は，代理モデルとしてランダムフォレ

スト，獲得関数として EI を用いる．SMAC と GP-EI は基本的に代理モデルが異なるだけで，仕組みに大きな違いはない．SMAC の特徴は，カテゴリーパラメータを自然に扱えることと，観測データが多い場合の実行時間が GP-EI と比較して小さいことであり，これらはランダムフォレストの特徴に由来する．

TPE（アルゴリズム名）は，代理モデルとして TPE，獲得関数として EI を用いる [8], [40]．この手法は，GP-EI や SMAC とは仕組みが大きく異なる．GP-EI や SMAC は，目的関数に対する代理モデルを構築する．一方，TPE は，各ハイパパラメータについて良質なハイパパラメータ設定の分布 $l(\boldsymbol{x})$ と，それ以外の分布 $g(\boldsymbol{x})$ をそれぞれカーネル密度推定する．その後，次の評価点の候補として，$l(\boldsymbol{x})$ から幾つかのハイパパラメータ設定をサンプルし，$l(\boldsymbol{x})/g(\boldsymbol{x})$ が最大となるものを，次の評価点として選択する（TPE において，$l(\boldsymbol{x})/g(\boldsymbol{x})$ が最大となる $\boldsymbol{x}$ は EI を最大化することが示されている [8]）．TPE は，カーネル密度推定によって連続，離散，カテゴリーパラメータ，ハイパパラメータの階層的なサンプルによって条件パラメータを自然に扱える．また，観測データが多い場合であっても，問題なく実行できる．

Eggensperger らは，GP-EI の代表的な実装である Spearmint [9]，SMAC，及び TPE について性能比較のための計算実験を行っている [47]．実験の結果，探索空間が低次元の問題では GP-EI，高次元，条件パラメータを含む問題では，SMAC や TPE が優れた性能を発揮したことが報告されている．

ベイズ最適化に関する参考文献を紹介する．文献 [11], [12] は標準的な GP-EI に関する解説を行っており，特に文献 [12] は，最新の発展的なトピックについても扱っている．ガウス過程については，文献 [33] が詳しい．SMAC については，提案者である Hutter の博士論文 [48] において，手法や計算実験の結果が極めて詳細に解説されている．TPE については，文献 [8], [40] が最も詳しい解説である．また，文献 [49] は，ベイズ最適化の網羅的なサーベイである．

3.4 進化計算

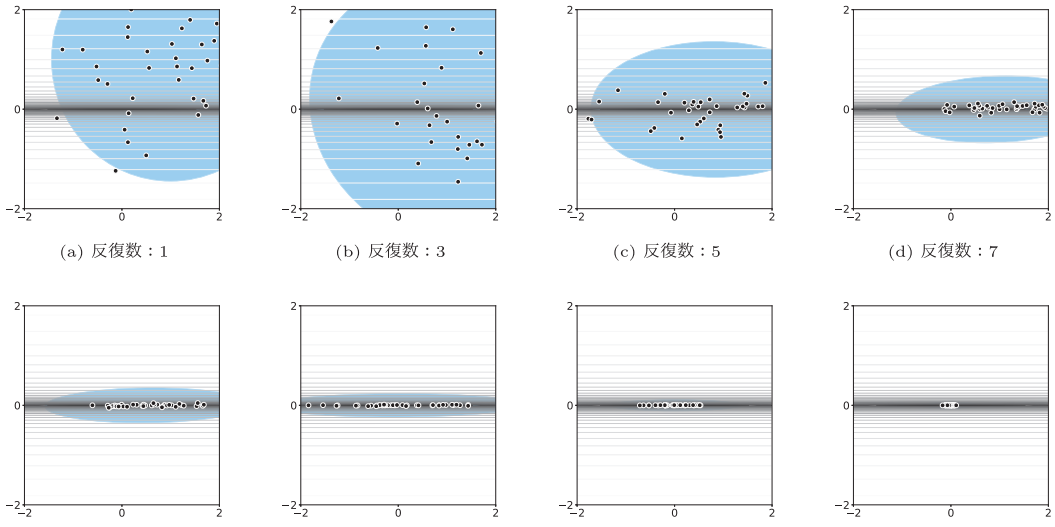
進化計算は，反復的に個体の生成と各個体の適応度評価を行う．ハイパパラメータ最適化において，個体はハイパパラメータ設定に対応し，個体の適応度は目的関数値に基づき計算される．この手法は，一般に各反復（世代と呼ぶ）において複数の個体を生成し評価を行う．このとき，各世代で評価した個体の適応度に基づいて次世代の個体を生成するため，世代単位で目的関数評価から得られる情報が探索に活用される．また，同一世代間の個体の評価は互いに依存しないため並列化できる．一方，異なる世代間の個体の評価は逐次的でなければならず，待ち合わせが必要である．個体の表現方法や生成方法は具体的な手法によって異なり，Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [50], [51] や Genetic Algorithm (GA) [52] が，ハイパパラメータ最適化で実績をもつ [53]〜[57]．

CMA-ES は，正規分布からの個体の生成，各個体の適応度評価，生成分布の更新を繰り返して最適化を行う．個体は正規分布からサンプルされるため，実数ベクトルで表現される．このため，連続パラメータのみからなる探索空間しか扱えない．図 4 に，個体群サイズを 40 とした CMA-ES を用いた最適化の実行例を示した．Loshchilov と Hutter が行った計算実験では，評価回数をそろえたとき，評価回数を大きくできるのであれば，CMA-ES がベイズ最適化よりも優れていることが報告されている [55]．

GA は，個体を表現するベクトル（個体の遺伝子と呼ばれる）に対する操作による個体の生成と，各個体の適応度評価を繰り返して最適化を行う．この手法において，個体の遺伝子の表現方法や生成方法は，極めて高い自由度をもつ．個体の遺伝子としては，実数ベクトルや 0–1 整数ベクトルが一般に用いられる．個体の生成方法としては，複数の個体の遺伝子を元に新たな個体を生成する交叉や，一個体の遺伝子をランダムに変化させることで新たな個体を生成する突然変異が一般に用いられる．この手法は，個体の遺伝子と生成方法を工夫することで，連続，離散，カテゴリー，条件パラメータを含む，複雑な探索空間を適切に扱える．反面，それらによって探索性能が大きく変わるため，問題ごとに適切な設計が必要である．

ハイパパラメータ最適化においては，複雑な探索空間を扱う必要がある場合のみ，適切な個体の遺伝子と生成方法を実装した GA を用い，そうでなければ，CMA-ES を用いればよい．

進化計算に関する参考文献を紹介する．文献 [13] は，CMA-ES の考案者である Hansen 自身による詳細なチュートリアルである．文献 [58] は，2018 年の GECCO で行われた Akimoto と Hansen による CMA-ES チュートリアルの講演資料である．文献 [14] は，最新の微分フリー/ブラックボックス最適化の教科書であり，GA が解説されている．

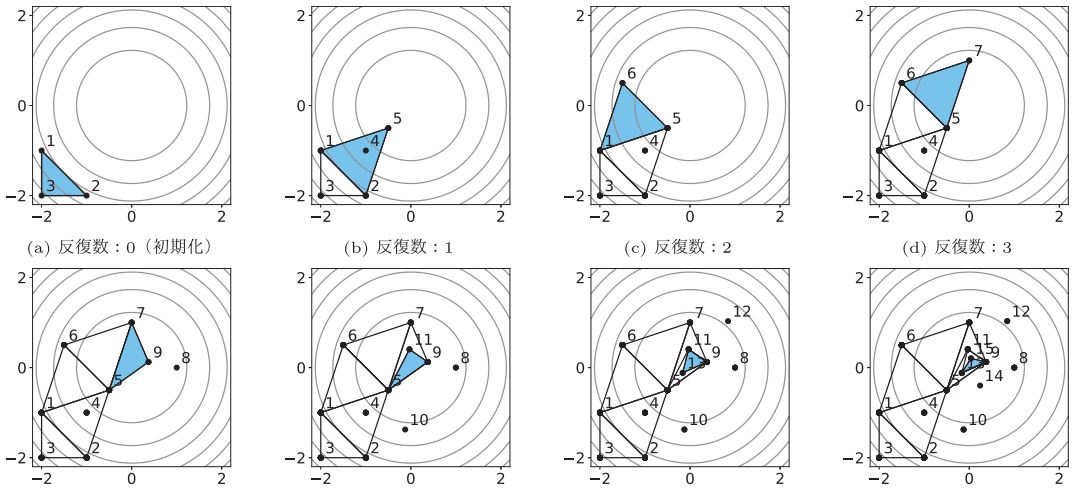


(a) 反復数：1　(b) 反復数：3　(c) 反復数：5　(d) 反復数：7

(e) 反復数：9　(f) 反復数：11　(g) 反復数：13　(h) 反復数：15

図 4　CMA-ES で $f(x, y) = x^2 + (100y)^2$ を最小化した例．点は正規分布から生成された個体，塗り潰しはその反復の時点での 95% 信頼だ円を表す．

Fig. 4　An example of minimizing $f(x, y) = x^2 + (100y)^2$ with CMA-ES. Each point represents an individual. The fill represents 95% confidence ellipse for the population of the iteration.



(a) 反復数：0（初期化）　(b) 反復数：1　(c) 反復数：2　(d) 反復数：3

(e) 反復数：4　(f) 反復数：5　(g) 反復数：6　(h) 反復数：7

図 5　Nelder–Mead 法で $f(x, y) = x^2 + y^2$ を最小化した例，点の添字は評価順序，塗り潰しはその反復の時点における単体を表す．

Fig. 5　An example of minimizing $f(x, y) = x^2 + y^2$ with the Nelder–Mead method. Each subscript represents the order of evaluation. The fill represents the simplex of the iteration.

3.5　Nelder–Mead 法

Nelder–Mead 法 [59] は，単体（n 次元空間内でアフィン独立な $n + 1$ 点がなす凸多面体）を用いたヒューリスティックである（図 5）．Nelder–Mead 法のおおまかな手順は以下である．まず，探索空間内にランダムに初期単体を生成し，単体の各頂点における目的関数値を求める．次に，それらの値の大小関係に基づく操作によって，次の評価点を生成する．そして，新たな評価点における目的関数値と，既存の各頂点における目的関数値の大小関係に基づく操作によって，単体を更新する．図 5 における水色の塗り潰しが，各反復時点における単体である．その後，終了条件（例えば，目

表 3　ブラックボックス最適化手法とその特徴
Table 3　Blackbox optimization methods and their characteristics.

手法	評価情報の活用	評価の並列化	扱える探索空間	LED への強さ	評価の不確実性の扱い方	探索傾向
グリッドサーチ	活用不可	全て可	離散・カテゴリー・条件	脆弱	複数回評価の平均・再評価	大域的
ランダムサーチ	活用不可	全て可	全種類	頑強	複数回評価の平均・再評価	大域的
GP-EI	評価ごとに活用	不適	全種類		ベイズ的にモデル化	大域的
SMAC	評価ごとに活用	不適	全種類		複数回評価の平均・再評価	大域的
TPE	評価ごとに活用	不適	全種類		複数回評価の平均・再評価	大域的
CMA-ES	世代単位で活用	世代単位で可	連続・離散		複数回評価の平均・再評価	大域的
GA	世代単位で活用	世代単位で可	全種類		複数回評価の平均・再評価	大域的
Nelder–Mead 法	評価ごとに活用	不適	連続・離散		複数回評価の平均・再評価	局所的

的関数評価回数の上限値）に達するまで，上述の目的関数評価と単体の更新を繰り返し行う．この手法は，評価ごとに目的関数評価から得られる情報を活用する一方，評価は原則として逐次的であるため並列化に適さない．また，ハイパパラメータ設定が実数ベクトルで表現されるため，連続パラメータのみからなる探索空間しか扱えない．

Nelder–Mead 法の最大の特徴は，局所的な探索を行うことである．グリッドサーチやランダムサーチは，良いハイパパラメータ設定を見つけられたとき，その近傍のより良いハイパパラメータ設定を積極的に探索する能力をもたない．また，ベイズ最適化や進化計算も，探索空間の幅広い範囲を大域的に探索しようとするため，近傍の局所解へと速やかには収束しない．対して，Nelder–Mead 法は多くの場合，比較的速やかに近傍の良い解へと収束する．一方，局所探索の欠点として，初期値への依存が高く [60]，悪質な局所解に陥る可能性があることが挙げられる．ただし，この問題は複数の Nelder–Mead 法を異なる初期点から実行するマルチスタートなどで緩和できる．

Nelder–Mead 法は，SVM や畳込みニューラルネットワークのチューニングにおいて実績がある [61]〜[63]．文献 [62] では，畳み込みニューラルネットワークのチューニングを行う計算実験において，Nelder–Mead 法が，ランダムサーチ，GP-EI，CMA-ES より高速に良いハイパパラメータ設定を発見できたことが報告されている．また，複数回の最適化により得られたハイパパラメータ設定を分析した結果，モデルが高い性能を達成する良いハイパパラメータ設定は一つではなく，多数存在していたことも報告されている．

Nelder–Mead 法に関する参考文献として，微分フリー/ブラックボックス最適化を扱った教科書 [14], [64] があり，手法の解説のほか，挙動の理論的な解析，ヒューリスティックが局所解への収束に失敗する例などが紹介されている．

3.6　実用的な最適化のテクニック

ハイパパラメータ最適化手法に適用可能な実用的なテクニックを紹介する．

まず，CMA-ES や Nelder–Mead 法は，連続パラメータしか扱えず，適用範囲が狭い．しかし，整数への丸めを行えば，実用上は離散パラメータを扱える．

次に，TPE，CMA-ES，Nelder–Mead 法などは，サンプルされる値に対して制約を与えることができない．しかし，ハイパパラメータの中には取りうる値に制約をもつものがある（例えば，ドロップアウト率は確率を表すため $[0, 1]$ の範囲の値のみを取る）ため，範囲外の値がサンプルされてしまうと学習を実行できず困る．このような場合，値が範囲外の際に，サンプルをやり直すか，損失関数値を ∞ として扱う Extreme Barrier Function (EBF) [14] を用いることで対処できる．

最後に，本論文で紹介した手法の中では，GP-EI 以外の手法は，明示的に評価の不確実性を扱う仕組みをもたない．しかし，これらの手法でも，目的関数値をあらかじめ複数回評価し平均を取ることや，目的関数を後から再評価し直すことで，評価の不確実性を扱える [65]．

以降の節では，これらのテクニックの使用を前提として議論を行う．

3.7　ブラックボックス最適化手法のまとめ

表 3 に，各ブラックボックス最適化手法の特徴を整理した．

目的関数評価から得られる情報の活用と評価の並列化可能性はトレードオフの関係にあることが分かる．グリッドサーチやランダムサーチは，最も並列化に適するが，評価から得られる情報は一切活用しない．一方，ベイズ最適化や Nelder–Mead 法は，目的関数を評価するごとに得られた情報を活用し，代理モデルの更新や，次の評価点の決定を行う．このため，評価は原則

として逐次で行わなければならない．進化計算は，世代単位で評価情報を活用するため，同一世代間において評価を並列化でき，二つの性質をバランスしている．

探索空間や不確実性の扱いについては，**3.6** で紹介した丸め，サンプルのやり直し，EBF，複数回評価の平均や再評価などのテクニックを用いれば，多くの手法で大部分に対応できる．

LED への強さについては，グリッドサーチは弱く，ランダムサーチは強いことが示されている [18] が，他の手法については，どちらともいえない．

探索傾向については，Nelder–Mead 法だけが局所的である．

4. グレーボックス最適化

グレーボックス最適化は，ハイパパラメータ最適化研究における近年のトレンドである．現在，グレーボックス最適化は発展途上であるため，研究者の間においても厳密な定義について十分な合意は取れていない(注1)．しかしながら，グレーボックス最適化手法とみなされている手法に共通する特徴として，目的関数値に加えて，対象問題の特徴から得られる最適化に有益な補助情報を活用し，従来のブラックボックス最適化手法を高速化している点が挙げられる．例えば，**2.** で例として挙げた問題 (2) であれば，学習途中のモデルの誤識別率や，D_{train} の一部を用いてモデルを学習した場合の誤識別率などが，この補助情報に該当する．

本節では，代表的なグレーボックス最適化手法について述べる．

4.1 データセットのサブサンプリング

データセットのサブサンプリング [66]〜[69] は，目的関数の評価コストを下げることで，従来のブラックボックス最適化手法を高速化する．この手法は，対象問題の特徴として，最適化する損失関数が学習データの増減に対してある程度ロバストであることを仮定する．このような仮定が成り立つならば，学習データセットのサイズを小さくすれば，モデルの学習時間を削減できるため，サブセットで学習した機械学習モデルに対するハイパパラメータ最適化問題を解くことで，元のハイパパラメータ最適化問題を短時間で近似的に解ける．

文献 [67], [68] では，SVM における 400 通りのハイパパラメータ設定について，教師データセットから全体の 1/128，1/16，1/4，1/1 をサブサンプリングしたサブセットを用いて学習する計算実験が行われた．その結果，異なるサブセットサイズ間でハイパパラメータ設定の優劣に強い相関があること，サブセットに対するハイパパラメータ最適化問題の解が元問題のよい近似解であることが実験的に示された．同文献では，Fast Bayesian optimization for large datasets (FABOLAS) と呼ばれる，データセットサイズに依存する目的関数の計算コストを加味した獲得関数を採用した，高速なベイズ最適化手法も提案されている．

4.2 学習の早期打ち切り

学習の早期打ち切り [69]〜[75] は，データセットのサブサンプリングと同様に，目的関数の評価コストを下げることで，従来のブラックボックス最適化手法を高速化する．この手法は，対象問題の特徴として，1) 学習途中のモデルを利用できること，2) 最適化する損失関数が学習経過に対してある程度ロバストであることを仮定する．

深層ニューラルネットワークなどの機械学習モデルは，勾配法などによって損失関数を反復的に最小化することで学習する．学習の早期打ち切りは，この学習経過を監視する．そして，現在のハイパパラメータ設定における学習を続けたとしても，他のハイパパラメータ設定に比べて，性能が勝る見込みがないと思われる場合，早期に学習を停止する．このような学習の打ち切りにより，従来のブラックボックス最適化手法において，最良である見込みのないハイパパラメータ設定のもとでモデルの学習に費やされていた時間や計算リソースを削減できる．

Successive Halving [70], [71] は，学習の早期打ち切りにより，ランダムサーチを効率化した手法である．はじめに，この手法は複数のハイパパラメータ設定をサンプルする．次に，各設定について，定められたリソース（典型的にはモデルの学習に費やせるエポック数など）を割り当て，モデルの学習を実行する，割り当てたリソースのもとで学習が済んだ後，性能が下位半分のハイパパラメータ設定を捨て去る．そして，残る上位のハイパパラメータ設定に追加のリソースを割り当て直し，引き続き学習を継続する．この操作を反復的に繰り返すと，最良のハイパパラメータ設定候補を効率的に絞り込むことができる．図 6 に Successive

(注1)：ICML 2019 にて併催された AutoML Workshop においてもグレーボックス最適化の講演が行われた．ここでも，箱の中身を見るという喩えと，学習曲線の予測などの具体的な手法が紹介されるに留まっており，厳密な定義は与えられていない．https://slideslive.com/38917532/greybox-bayesian-optimization-for-automl.

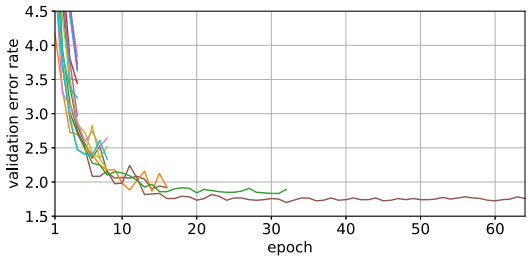


図 6　Successive Halving の実行結果例．性能の低いハイパパラメータ設定の学習は早期に打ち切られていることが見て取れる．

Fig. 6　An example run of Successive Halving. Poor configured trainings are terminated in its early stage.

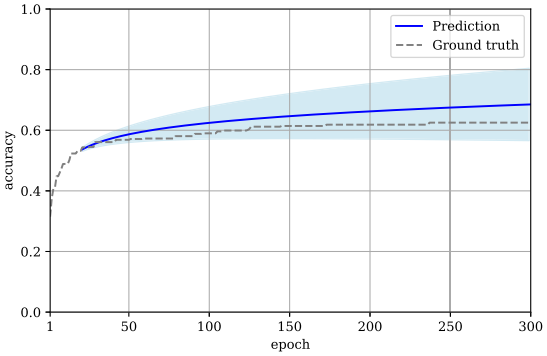


図 7　青線は，文献 [76] の手法を用いて機械学習モデルの学習が 20 エポック経過した時点で予測した学習曲線．点線は，真の学習曲線．塗り潰しは，95% 信頼区間．この文献では，ハイパパラメータ最適化を正答率の最大化として定式化しており，学習曲線予測モデルはチューニング対象としている機械学習モデルの正答率を予測する．

Fig. 7　The blue line is a predicted learning curve generated by the method proposed in [76]. The black line is the ground truth. The fill is 95% confidence interval. In [76], hyperparameter optimization is formalized as accuracy maximization and the learning curve prediction model predicts the accuracy of the target model.

Halving の実行結果例を示した．

Successive Halving には，幾つかの改良手法が提案されている．Hyperband [69] は，絞り込みの積極性を調整するパラメータを変化させながら，複数回の Successive Halving を実行する．この手法は，学習の序盤に性能が低かったハイパパラメータ設定が終盤に逆転する場合に，誤って早期に学習を打ち切ってしまう失敗が少なく，Successive Halving より高い探索性能をもつ．Asynchronous Successive Halving Algorithm [72] は，目的関数の評価を非同期化し，Successive Halving の並列化性能を向上させている．また，文献 [73], [74] では，ベイズ最適化と Hyperband を組み合わせた手法が提案されている．

Successive Halving や Hyperband は，探索にランダムサーチを用い，学習経過を監視して早期打ち切りを行う．しかし，グリッドサーチとランダムサーチを除き，本論文で紹介したブラックボックス最適化手法は，次に評価すべきハイパパラメータ設定を選択するために評価情報を活用する（表 3）．ところが，単純な学習の早期打ち切りは，この評価情報に大きな影響を及ぼしてしまう．そこで，評価情報を活用する最適化手法に対しても，学習の早期打ち切りによる高速化を実現するため，学習曲線を予測する（図 7）ことで機械学習モデルの学習完了時の性能を見積もる手法が数多く提案されている [76]〜[82]．これらの手法では，学習曲線予測モデルを，学習を早期に打ち切るか否か決定するアルゴリズム [76], [80] を介してブラックボックス最適化手法と繋ぎ込むことで，高速化を実現する．

4.3　ウォームスタート

ウォームスタート [83]〜[87] は，過去に解いたハイパパラメータ最適化問題の結果を利用することで，従来のブラックボックス最適化手法の初期化を改良し，最適化を高速化する．この手法は，対象問題の特徴として，最適化する損失関数と過去に解いたハイパパラメータ最適化問題の損失関数の間で解の優劣に相関があることを仮定する．

Multi-Task Bayesian Optimization [83] は，Multi-task Gaussian Process [88] をベイズ最適化に導入し，過去に異なるタスクに対して最適化を行った際のデータを現在の最適化に活用することで，最適化を高速化する．Sequential Model-based Bayesian Optimization approach with meta-learning-based initialization (MI-SMBO) [84], [85] は，データセットから得られるメタ特徴量に基づくタスク間の距離を用いて類似タスクを絞り込み，それらに対する過去の観測データを利用することで，最適化を高速化する．このほかにも，ベイジアンニューラルネットワークやベイズ線形回帰に基づく手法が存在する [86], [87].

4.4　グレーボックス最適化のまとめ

グレーボックス最適化手法とその特徴を，表 4 に整理した．これらの手法を用いれば，チューニングを高速化したり，限られた時間内でより多くの目的関数評価が可能となる．また，各手法は互いに排他的なものではないため，併用できる．ただし，一般に，高速化

表 4　グレーボックス最適化手法とその特徴
Table 4　The acceleration techniques in graybox optimization and their characteristics.

手法	対象問題の特徴に対する仮定	その他の要件
データセットのサブサンプリング	最適化する損失関数が学習データの増減に対してある程度ロバストであること.	
学習の早期打ち切り	学習が反復的であること，各反復において損失関数が評価可能でありモデルが利用可能であること，最適化する損失関数が学習経過に対してある程度ロバストであること.	高速化対象のブラックボックス最適化手法が評価情報を活用する場合には，学習曲線予測モデルとの繋ぎ込みが必要.
ウォームスタート	最適化する損失関数と過去に解いたハイパパラメータ最適化問題の損失関数の間で解の良し悪しに相関があること.	過去に解いたハイパパラメータ最適化の結果が必要，高速化対象のブラックボックス最適化手法に適したウォームスタート手法が個別に必要.

の程度と近似の誤差はトレードオフの関係にあることに注意が必要である.

データセットのサブサンプリングや学習の早期打ち切りのように，データセットのサイズや学習のエポック数などを調整することで多段階の目的関数の近似を利用する手法は，マルチフィデリティ最適化と呼ばれる [6]. **4.2** で紹介した Hyperband や文献 [89] で提案されている手法は，リソースとして学習エポック数だけでなく，データセットのサイズや特徴量の数なども扱える．また，文献 [90], [91] で提案されている手法など，複数のリソースを同時に扱えるものもある.

グレーボックス最適化に関する参考文献を紹介する．文献 [6] には，マルチフィデリティ最適化の解説がある．また，メタラーニングに関するサーベイである文献 [92] には，ウォームスタートに関する情報がある.

5. 最適化手法選択のガイドライン

本節では，ハイパパラメータ最適化を行う上で適切なブラックボックス/グレーボックス最適化手法の選択について，以下の観点からそれぞれ議論する.

（1） 逐次評価回数の上限値
　(1–a) 標準的（数十回以上）
　(1–b) 限定的（数回程度）
（2） 並列計算リソース
　(2–a) 少ない（一から十程度）
　(2–b) 多い（数十以上）
（3） ハイパパラメータの種類
　(3–a) カテゴリー・条件パラメータなし
　(3–b) カテゴリー・条件パラメータあり

これらの観点は，ハイパパラメータ最適化手法自体に関するものではなく，ハイパパラメータ最適化を用いる我々自身の状況に関するものである．逐次評価回数の上限値は，我々がチューニングに費やせる時間と機械学習モデルの学習にかかる時間の比によって定まる．並列計算リソースは，我々が費やせる予算や作業環境などに依存して定まる．ハイパパラメータの種類は，我々がモデルのどのハイパパラメータをチューニング対象とするか選ぶことで定まる.

5.1 逐次評価回数の上限値

一般に，目的関数の逐次評価回数の上限値が限定的な場合，数回の逐次評価で良質なハイパパラメータ設定を見つけ出すことは極めて難しい．この場合，対象問題の特徴がグレーボックス最適化手法の仮定を満たすならば，グレーボックス最適化手法を用いるべきである．データセットのサブサンプリングや学習の早期打ち切りによって目的関数の評価コストが下がれば，逐次回数評価の上限値を緩和できる．また，ウォームスタートも擬似的に逐次評価回数の上限値を増やすことに相当する.

並列計算リソースが多く利用可能な場合，進化計算やランダムサーチを用いることで，数回の逐次評価でも良質なハイパパラメータ設定を探索できる場合がある（**5.2** を参照せよ）.

それ以外の場合，現時点で有望な手段はないため，チューニングに費やせる時間や並列計算リソースを見直すべきである.

5.2 並列計算リソース

並列計算リソースが少ない場合，進化計算や局所探索の弱点を補うマルチスタートは効果的でない．そこで，目的関数評価から得られる情報を最大限に活用して大域的な探索を行うベイズ最適化及び，それらの小規模な並列化 [8], [9], [26] が最も有望な選択肢となる．GP-EI，SMAC，TPE の使い分けについては，文献 [47] における議論に従い，対象問題の探索空間が低次元かつ連続な探索空間をもつ場合には GP-EI，高次元であ

表 5　最適化手法選択のガイドライン
Table 5　Guidelines for selecting optimization methods.

逐次評価回数の上限値	並列計算リソース	ハイパパラメータの種類	適切なハイパパラメータ最適化手法
(1–a) 標準的	(2–a) 少ない	(3–a) カテゴリー・条件パラメータなし	対象問題の探索空間が低次元で連続な場合には GP-EI，高次元である場合や条件パラメータを含むような場合には SMAC や TPE を用いる．
		(3–b) カテゴリー・条件パラメータあり	同上．
	(2–b) 多い	(3–a) カテゴリー・条件パラメータなし	CMA-ES，ランダムサーチ，または Nelder–Mead 法のマルチスタートを用いる．
		(3–b) カテゴリー・条件パラメータあり	ランダムサーチ，または GA を用いる．
(1–b) 限定的	(2–a) 少ない	(3–a) カテゴリー・条件パラメータなし	グレーボックス最適化手法を適用できれば状況 (1–a) とみなせる．難しければ，チューニングに費やせる時間や並列計算リソースを見直して状況 (1–a) や (2–b) を目指す．
		(3–b) カテゴリー・条件パラメータあり	同上．
	(2–b) 多い	(3–a) カテゴリー・条件パラメータなし	CMA-ES，またはランダムサーチを用いる．
		(3–b) カテゴリー・条件パラメータあり	ランダムサーチ，または GA を用いる．

る場合や条件パラメータを含むような場合には SMAC や TPE を用いればよい．

並列計算リソースが多く，数十程度の並列化を行える場合，進化計算が最も有望な選択肢となる．文献 [55] の結果では，目的関数評価を 30 並列に行う CMA-ES が，200 回未満の目的関数評価回数で，同評価回数のベイズ最適化を上回っている．これは，逐次評価に換算して 6 回から 7 回程度の実行時間であるため，評価回数の上限値が限定的な場合でも十分対応できる．

逐次評価回数の上限値が標準的な場合には，局所探索を行う Nelder–Mead 法をマルチスタートする手も考えられる．

更に，数百以上の並列化を行える場合，ランダムサーチが最も有望な選択肢となる．ランダムサーチは，全ての評価を非同期に並列化できるため，待ち合わせによるオーバーヘッドが発生せず，並列数が非常に大きい場合，進化計算より計算リソースを無駄なく活用できるためである．

5.3　ハイパパラメータの種類

チューニング対象の機械学習モデルがカテゴリー・条件パラメータをもたない場合，全ての手法が利用できる．このとき，進化計算については，**3.4** で述べたように CMA-ES を用いることを勧める．

一方，チューニング対象の機械学習モデルがカテゴリー・条件パラメータをもつ場合，これらを扱えない手法は利用できないため，CMA-ES と Nelder–Mead 法は選択肢から外れる．

いずれの場合においても，残された選択肢から，逐次評価回数の上限値及び並列計算リソースに基づき，最適化手法を選択すればよい．ただし，グリッドサーチとランダムサーチについては，**3.1** と **3.2** で述べた性質から，ランダムサーチを用いることを勧める．

5.4　最適化手法選択のガイドラインのまとめ

これまでの議論を元に，状況ごとの適切なハイパパラメータ最適化手法をまとめて表 5 に最適化手法選択のガイドラインとして示した．

ただし，このガイドラインはあくまでも一つの考え方に過ぎず，常に従う必要はないことに注意して欲しい．例えば，本論文で紹介した 3 種類以外にも多くのベイズ最適化手法が存在するが，ここでは検討されていない．また，チューニングに費やせる時間が多くある場合でも，高速化と誤差のトレードオフを考慮した上で，グレーボックス最適化手法を検討してもよい．

6.　む　す　び

本論文では，ハイパパラメータ最適化において標準的であるブラックボックス最適化手法について代表的なものを概説し，それらの特徴をハイパパラメータ最適化手法に望まれる性質に基づき整理した．更に，近年のトレンドであるグレーボックス最適化手法についても，代表的なものを概説，整理した．最後に，逐次評価回数の上限値，並列計算リソース，ハイパパラメータの種類の三つの観点に基づいて，適切なハイパパラメータ最適化手法を議論し，各状況に応じたガイドラ

インを与えた．

人の労力削減，機械学習モデルの性能向上，研究の公平性改善など複数の観点から，アカデミアと産業界の双方において，ハイパパラメータ最適化をはじめとする機械学習の自動化を支える技術の広がりは，今後も一層拡大していくと考えられる．

本論文が，機械学習に関わる研究，産業の現場で活躍する読者の一助となれば幸いである．

謝辞　この成果は，国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務の結果得られたものである．

文　献

[1] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, pp.1097–1105, 2012.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Advances in Neural Information Processing Systems, pp.2672–2680, 2014.

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

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1–9, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.770–778, 2016.

[6] M. Feurer and F. Hutter, "Hyperparameter optimization," Automated Machine Learning, pp.3–33, Springer, 2019.

[7] M.F. Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? a worrying analysis of recent neural recommendation approaches," Proc. 13th ACM Conference on Recommender SystemsACM, pp.101–109, 2019.

[8] J.S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-Parameter Optimization," Advances in Neural Information Processing Systems, pp.2546–2554, 2011.

[9] J. Snoek, H. Larochelle, and R.P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," Advances in Neural Information Processing Systems, pp.2951–2959, 2012.

[10] F. Hutter, J. Lücke, and L. Schmidt-Thieme, "Beyond manual tuning of hyperparameters," KI-Künstliche Intelligenz, vol.29, no.4, pp.329–337, 2015.

[11] E. Brochu, V.M. Cora, and N.D. Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," arXiv preprint arXiv:1012.2599, pp.1–49, 2010.

[12] P.I. Frazier, "A tutorial on Bayesian optimization," arXiv preprint arXiv:1807.02811, pp.1–22, 2018.

[13] N. Hansen, "The cma evolution strategy: A tutorial," arXiv preprint arXiv:1604.00772, pp.1–39, 2016.

[14] C. Audet and W. Hare, Derivative-Free and Blackbox Optimization, Springer Series in Operations Research and Financial Engineering, Springer International Publishing, 2017.

[15] S. Hansen, "Using deep q-learning to control optimization hyperparameters," arXiv preprint arXiv: 1602.04062, pp.1–14, 2016.

[16] I. Bello, B. Zoph, V. Vasudevan, and Q.V. Le, "Neural optimizer search with reinforcement learning," International Conference on Machine Learning, pp.459–468, 2017.

[17] X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao, and F. Porikli, "Hyperparameter optimization for tracking with continuous deep q-learning," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.518–527, 2018.

[18] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," J. Mach. Learn. Research, vol.13, no.Feb, pp.281–305, 2012.

[19] F. Hutter, H. Hoos, and K. Leyton-Brown, "An efficient approach for assessing hyperparameter importance," Proc. 31st International Conference on International Conference on Machine Learning-Volume 32, pp.754–762, 2014.

[20] J.N. vanRijn and F. Hutter, "Hyperparameter importance across datasets," Proc. 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp.2367–2376, 2018.

[21] Y. Bengio, "Gradient-based optimization of hyperparameters," Neural Computation, vol.12, no.8, pp.1889–1900, 2000.

[22] D. Maclaurin, D. Duvenaud, and R.P. Adams, "Gradient-based hyperparameter optimization through reversible learning," Proc. 32nd International Conference on International Conference on Machine Learning-Volume 37, pp.2113–2122, 2015.

[23] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil, "Forward and reverse gradient-based hyperparameter optimization," Proc. 34th International Conference on Machine Learning-Volume 70, pp.1165–1173, 2017.

[24] M.D. McKay, R.J. Beckman, and W.J. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," Technometrics, vol.21, no.2, pp.239–245, 1979.

[25] I.M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, vol.7, no.4, pp.784–802, 1967.

[26] F. Hutter, H.H. Hoos, and K. Leyton-Brown, "Parallel algorithm configuration," International Conference on Learning and Intelligent OptimizationSpringer, pp.55–70, 2012.

[27] E. Contal, D. Buffoni, A. Robicquet, and N. Vayatis, "Parallel Gaussian process optimization with upper confidence bound and pure exploration," Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp.225–240, 2013.

[28] T. Desautels, A. Krause, and J.W. Burdick, "Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization," J. Machine Learning Research, vol.15, no.1, pp.3873–3923, 2014.

[29] A. Shah and Z. Ghahramani, "Parallel predictive entropy search for batch global optimization of expensive objective functions," Advances in Neural Information Processing Systems, pp.3330–3338, 2015.

[30] T. Kathuria, A. Deshpande, and P. Kohli, “Batched gaussian process bandit optimization via determinantal point processes,” Advances in Neural Information Processing Systems, pp.4206–4214, 2016.

[31] J. Wu and P. Frazier, “The parallel knowledge gradient method for batch Bayesian optimization,” Advances in Neural Information Processing Systems, pp.3126–3134, 2016.

[32] K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Póczos, “Parallelised Bayesian optimisation via Thompson sampling,” International Conference on Artificial Intelligence and Statistics, pp.133–142, 2018.

[33] C.K. Williams and C.E. Rasmussen, Gaussian processes for machine learning, vol.2, MIT Press Cambridge, MA, 2006.

[34] F. Hutter, H.H. Hoos, and K. Leyton-Brown, “Sequential model-based optimization for general algorithm configuration (extended version),” Technical Report TR-2010-10, University of British Columbia, Department of Computer Science, 2010. Available online: http://www.cs.ubc.ca/˜hutter/papers/10-TR-SMAC.pdf.

[35] K. Swersky, D. Duvenaud, J. Snoek, F. Hutter, and M.A. Osborne, “Raiders of the lost architecture: Kernels for Bayesian optimization in conditional parameter spaces,” arXiv preprint arXiv:1409.4011, pp.1–5, 2014.

[36] J.-C. Lévesque, A. Durand, C. Gagné, and R. Sabourin, “Bayesian optimization for conditional hyperparameter spaces,” 2017 International Joint Conference on Neural Networks (IJCNN), pp.286–293, 2017.

[37] J. Quiñonero-Candela, C. Rasmussen, C. Williams, O. Chapelle, D. DeCoste, J. Weston, et al., “Approximation methods for Gaussian process regression,” Large-Scale Kernel Machines, pp.203–223, MIT Press, 2007.

[38] M. Titsias, “Variational learning of inducing variables in sparse Gaussian processes,” Proc. Twelth International Conference on Artificial Intelligence and Statistics, eds. by D. vanDyk and M. Welling, vol.5, pp.567–574, Proc. Machine Learning Research, PMLR, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 2009.

[39] F. Hutter, H.H. Hoos, and K. Leyton-Brown, “Sequential Model-Based Optimization for General Algorithm Configuration,” International Conference on Learning and Intelligent Optimization, pp.507–523, 2011.

[40] J. Bergstra, D. Yamins, and D. Cox, “Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures,” International Conference on Machine Learning, pp.115–123, 2013.

[41] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams, “Scalable Bayesian optimization using deep neural networks,” International conference on machine learning, pp.2171–2180, 2015.

[42] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D.J. Rezende, S. Eslami, and Y.W. Teh, “Neural processes,” arXiv preprint arXiv:1807.01622, 2018.

[43] D.R. Jones, M. Schonlau, and W.J. Welch, “Efficient Global Optimization of Expensive Black-Box Functions,” J. Global Optimization, vol.13, no.4, pp.455–492, 1998.

[44] H.J. Kushner, “A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise,” J. Basic Engineering, vol.86, no.1, pp.97–106, 1964.

[45] N. Srinivas, A. Krause, S.M. Kakade, and M.W. Seeger, “Information-theoretic regret bounds for Gaussian process optimization in the bandit setting,” IEEE Trans. Inf. Theory, vol.58, no.5, pp.3250–3265, 2012.

[46] J.M. Hernández-Lobato, M.W. Hoffman, and Z. Ghahramani, “Predictive entropy search for efficient global optimization of black-box functions,” Advances in Neural Information Processing Systems, pp.918–926, 2014.

[47] K. Eggensperger, M. Feurer, F. Hutter, J. Bergstra, J. Snoek, H. Hoos, and K. Leyton-Brown, “Towards an empirical foundation for assessing Bayesian optimization of hyperparameters,” NIPS workshop on Bayesian Optimization in Theory and Practice, 2013.

[48] F. Hutter, “Automated configuration of algorithms for solving hard computational problems,” PhD thesis, University of British Columbia, 2009.

[49] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, and N.D. Freitas, “Taking the human out of the loop: A review of Bayesian optimization,” Proc. IEEE, vol.104, no.1, pp.148–175, 2016.

[50] N. Hansen and A. Ostermeier, “Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation,” Proc. IEEE International Conference on Evolutionary Computation, pp.312–317, 1996.

[51] N. Hansen and A. Ostermeier, “Completely Derandomized Self-Adaptation in Evolution Strategies,” Evolutionary computation, vol.9, no.2, pp.159–195, 2001.

[52] J.H. Holland, et al., Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT press, 1992.

[53] F. Friedrichs and C. Igel, “Evolutionary Tuning of Multiple SVM Parameters,” Neurocomputing, vol.64, pp.107–117, 2005.

[54] S. Watanabe and J. Le Roux, “Black box optimization for automatic speech recognition,” 2014 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), pp.3256–3260, 2014.

[55] I. Loshchilov and F. Hutter, “CMA-ES for Hyperparameter Optimization of Deep Neural Networks,” ICLR Workshop, 2016.

[56] F. Leung, H. Lam, S. Ling, and P. Tam, “Tuning of the structure and parameters of a neural network using an improved genetic algorithm,” IEEE Trans. Neural Netw., pp.79–88, 2003.

[57] S.R. Young, D.C. Rose, T.P. Karnowski, S.-H. Lim, and R.M. Patton, “Optimizing deep learning hyper-parameters through an evolutionary algorithm,” Proc. Workshop on Machine Learning in High-Performance Computing Environments, p.4, 2015.

[58] Y. Akimoto and N. Hansen, “Cma-es and advanced adaptation mechanisms,” Proc. Genetic and Evolutionary Computation Conference CompanionACM, pp.720–744, 2018.

[59] J.A. Nelder and R. Mead, “A simplex method for function minimization,” Comput. J., vol.7, no.4, pp.308–313, 1965.

[60] S. Wessing, “Proper initialization is crucial for the Nelder–Mead simplex search,” Opt. Lett., pp.1–10, 2018.

[61] G. Cohen, P. Ruch, and M. Hilario, “Model Selection for Support Vector Classifiers via Direct Simplex Search,” Proc. Eighteenth International Florida Artificial Intelligence Research Society Conference, Clearwater Beach, Florida, USA, pp.431–435, 2005.

[62] Y. Ozaki, M. Yano, and M. Onishi, "Effective hyperparameter optimization using Nelder-Mead method in deep learning," IPSJ Transactions on Computer Vision and Applications, vol.9, no.1, p.20, 2017.
[63] Y. Ozaki, S. Watanabe, and M. Onishi, "Accelerating the Nelder–Mead method with predictive parallel evaluation," 6th ICML Workshop on Automated Machine Learning, pp.1–9, 2019.
[64] A.R. Conn, K. Scheinberg, and L.N. Vicente, Introduction to derivative-free optimization, vol.8, Siam, 2009.
[65] H. Wang, H. Qian, and Y. Yu, "Noisy derivative-free optimization with value suppression," Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[66] L. Bottou, "Stochastic gradient descent tricks," Neural Netw.: Tricks of the trade, pp.421–436, Springer, 2012.
[67] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast Bayesian optimization of machine learning hyperparameters on large datasets," International Conference on Artificial Intelligence and Statistics (AISTATS 2017), pp.528–536, 2017.
[68] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast Bayesian hyperparameter optimization on large datasets," Electronic J. Statistics, vol.11, 2017.
[69] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization," J. Machine Learning Research, vol.18, no.185, pp.1–52, 2018.
[70] Z. Karnin, T. Koren, and O. Somekh, "Almost optimal exploration in multi-armed bandits," International Conference on Machine Learning, pp.1238–1246, 2013.
[71] K. Jamieson and A. Talwalkar, "Non-stochastic best arm identification and hyperparameter optimization," Artificial Intelligence and Statistics, pp.240–248, 2016.
[72] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, and A. Talwalkar, "Massively parallel hyperparameter tuning," arXiv preprint arXiv:1810.05934, pp.1–16, 2018.
[73] J. Wang, J. Xu, and X. Wang, "Combination of hyperband and Bayesian optimization for hyperparameter optimization in deep learning," arXiv preprint arXiv:1801.01596, pp.1–10, 2018.
[74] S. Falkner, A. Klein, and F. Hutter, "Bohb: Robust and efficient hyperparameter optimization at scale," International Conference on Machine Learning, pp.1436–1445, 2018.
[75] Z. Dai, H. Yu, B.K.H. Low, and P. Jaillet, "Bayesian optimization meets Bayesian optimal stopping," International Conference on Machine Learning, pp.1496–1506, 2019.
[76] T. Domhan, J.T. Springenberg, and F. Hutter, "Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves," Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.
[77] K. Swersky, J. Snoek, and R.P. Adams, "Freeze-thaw Bayesian optimization," arXiv preprint arXiv:1406. 3896, pp.1–12, 2014.
[78] A. Klein, S. Falkner, J.T. Springenberg, and F. Hutter, "Learning Curve Prediction with Bayesian Neural Networks," International Conference on Learning Representations (ICLR) 2017 Conference Track, 2017.
[79] L. Faivishevsky and A. Armon, "Deep structured modeling of deep learning training convergence with application to hyperparameter optimization," Proc. ICML 17 Workshop on Deep Structured Prediction, 2017.
[80] A. Chandrashekaran and I.R. Lane, "Speeding up Hyper-parameter Optimization by Extrapolation of Learning Curves Using Previous Builds," Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp.477–492, 2017.
[81] B. Baker, O. Gupta, R. Raskar, and N. Naik, "Accelerating neural architecture search using performance prediction," International Conference on Learning Representations (ICLR) 2018 Workshop Track, 2018.
[82] M. Gargiani, A. Klein, S. Falkner, and F. Hutter, "Probabilistic rollouts for learning curve extrapolation across hyperparameter settings," 6th ICML Workshop on Automated Machine Learning, pp.1–31, 2019.
[83] K. Swersky, J. Snoek, and R.P. Adams, "Multi-task bayesian optimization," Advances in Neural Information Processing Systems, pp.2004–2012, 2013.
[84] M. Feurer, J.T. Springenberg, and F. Hutter, "Using meta-learning to initialize Bayesian optimization of hyperparameters," Proc. 2014 International Conference on Meta-learning and Algorithm Selection-Volume 1201, pp.3–10, 2014.
[85] M. Feurer, J.T. Springenberg, and F. Hutter, "Initializing Bayesian hyperparameter optimization via meta-learning," Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[86] J.T. Springenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian optimization with robust Bayesian neural networks," Advances in Neural Information Processing Systems, pp.4134–4142, 2016.
[87] V. Perrone, R. Jenatton, M.W. Seeger, and C. Archambeau, "Scalable hyperparameter transfer learning," Advances in Neural Information Processing Systems, pp.6845–6855, 2018.
[88] E.V. Bonilla, K.M. Chai, and C. Williams, "Multi-task Gaussian process prediction," Advances in neural information processing systems, pp.153–160, 2008.
[89] K. Kandasamy, G. Dasarathy, J.B. Oliva, J.G. Schneider, and B. Póczos, "Multi-fidelity Gaussian process bandit optimisation,"J. Artif. Intell. Res., vol.66, pp.151–196, 2019.
[90] K. Kandasamy, G. Dasarathy, J. Schneider, and B. Poczos, "Multi-fidelity Bayesian optimisation with continuous approximations," Proc. 34th International Conference on Machine Learning-Volume 70, pp.1799–1808, 2017.
[91] J. Wu, S. Toscano-Palmerin, P.I. Frazier, and A.G. Wilson, "Practical multi-fidelity Bayesian optimization for hyperparameter tuning," Proc. Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, p.284, 2019.
[92] J. Vanschoren, "Meta-learning," Automated Machine Learning, pp.35–61, Springer, 2019.

（2019 年 10 月 30 日受付，2020 年 3 月 5 日再受付，
5 月 14 日早期公開）

尾崎　嘉彦

2017 筑波大学大学院システム情報工学研究科博士前期課程了．同年グリー株式会社入社．2018 より産業技術総合研究所人工知能研究センター特定集中研究専門員（兼務）．ハイパパラメータ最適化に関する研究で電子情報通信学会 PRMU 研究会研究奨励賞（2017）受賞．

野村　将寛

2017 東京工業大学大学院総合理工学研究科知能システム科学専攻博士前期課程了．同年株式会社サイバーエージェント入社．株式会社サイバーエージェントにて機械学習手法のハイパパラメータ最適化に関する研究に従事．2019 より産業技術総合研究所人工知能研究センター特定集中研究専門員（兼務）．

大西　正輝　（正員）

1997 大阪府立大学工学部情報工学科卒業．2002 同大大学院博士後期課程了．同年理化学研究所バイオ・ミメティックコントロール研究センター研究員を経て，2006 産業技術総合研究所情報技術研究部門研究員，現在人工知能研究センター社会知能研究チーム長．画像認識による人の検出とその応用に関する研究に従事．博士（工学）．