Vol.2021-MBL-98 No.10
Vol.2021-UBI-69 No.10
2021/3/1

ベイズ最適化と蒸留を用いた最適な圧縮モデル探索手法の提案

別宮 広朗[1,a)]　小澤 遼[2,b)]　大越 匡[2,c)]　中澤 仁[1,d)]

概要：近年, DeepLearning 技術が急激に発達し高精度なニューラルネットワークモデルが多数出現しており, IoT デバイス等の様々なデバイスへ搭載することが期待される. ニューラルネットワークモデルは層や重みパラメータが多いほど精度が向上する傾向があり, 高精度なモデルは推論時間が長くなる場合が多い. 計算資源の限られた IoT デバイス等の小型端末に搭載するためには, 限られた計算資源でも軽快に動作するモデルの構築が求められる. 推論時間を削減する手法の一つとして, 蒸留という手法が存在する. 蒸留は高精度な教師モデルの知識を小さい生徒モデルに学習させてニューラルネットワークの圧縮を行う技術である. しかし, 生徒モデルの任意性は高くトレードオフな関係にある推論速度と精度を両立するようなモデルの発見は困難である. また, 実際に小型端末に搭載する上で, アプリケーションの目的に応じて推論速度や精度の重要度も変わるため, 目的に応じた圧縮を行える必要がある. そのため本研究では, 推論速度と精度に最低値を設定した探索や推論速度と精度に比重を置いた探索を行うことを可能とする評価関数を定義し, ベイズ最適化を用いて, 推論速度を重視, または精度を重視等の目的に応じた圧縮のための最適な生徒モデルの探索手法を提案し, 検証実験を行う. 更に, 圧縮モデルの推論速度と精度の探索パレート最適解によって得られた, 圧縮モデルが達成できる限界値の曲線を可視化する.

1. はじめに

近年, DeepLearning 技術が急激に発達している. 高精度なニューラルネットワークモデルは, 画像分類や自然言語などの様々な分野において多数出現しており, IoT デバイス等へ実際に搭載することが期待される. 一般的に, ニューラルネットワークモデルは, 層や重みパラメータが多いほど精度が向上する傾向がある. 更に, その大規模なモデルをアンサンブルさせて汎化性を高めた精度の高いモデルを利用する場合もあり, 推論時間が非常に長くなる場合が多い. 計算資源の限られた IoT デバイス等の小型端末に搭載するためには, 限られた計算資源でも軽快に動作するモデルの構築が求められる.

ニューラルネットワークモデルは, モデル構造に注目して, 精度を向上させるための技術は盛んに研究が行われている. ハイパーパラメータの最適化や Neural Architecture Search（以下, NAS と呼ぶ）などの技術は, 精度の向上を主な目的として, パラメータやモデル構造の探索に適した手法を用いて, 最適化する分野である. しかし, モデル圧縮のための推論時間やモデルの大きさを考慮した探索を行っている研究は少なく, 始まったばかりであると考えられる.

ニューラルネットワークモデルの推論時間を削減する手法では, モデル圧縮技術が有効である. モデル圧縮技術は, 枝刈り, 量子化, 蒸留が有名であり, パラメータ数の削減による圧縮が主となっている. しかし, 上記の圧縮技術はパラメータを削減するため, 軽量化や高速化が望める一方で, 精度が下がる欠点がある. 大量のパラメータの削減を行った場合, 高速化が望まれるものの, 精度が著しく欠落し, 少量のパラメータ削減では, 精度は維持されるものの, 十分な高速化を望むことができない. これはニューラルネットワークモデルの推論時間と精度がトレードオフの関係であることが要因となると考えられる. そのため, それぞれを両立できるようなモデルを探索するためには, 同時に評価できるような評価関数を用いる必要がある. その評価関数は, タスクによっては必ずしも推論時間と精度が同様に重視されているとは限らず, 精度の方が重要性が高いといった状況等に適応する必要がある.

本研究では, 蒸留を用いたモデル圧縮における推論時間

[1] 慶應義塾大学環境情報学部
Faculty of Environment and Information Studies, Keio University, Fujisawa, Kanagawa 252-0882, Japan
[2] 慶應義塾大学大学院政策・メディア研究科
Graduate School of Media and Governance, Keio University, Fujisawa, Kanagawa 252-0882, Japan
[a)] t18727hb@sfc.keio.ac.jp
[b)] oza15015@.keio.jp
[c)] slash@sfc.keio.ac.jp
[d)] jin@sfc.keio.ac.jp

と精度のトレードオフ関係による, 最適な圧縮モデル構築の曖昧性を問題視する. そして, 圧縮先となる生徒モデルの推論時間と精度を両立するために, 正規化したそれぞれの評価値の調和平均を用いた評価関数による評価値が最大となるモデルをベイズ最適化で探索することによって, 最適なモデルを発見する手法を提案する. 推論時間と精度の重要性や条件は状況やタスクによって異なるため, 本研究では, どちらかに重視するようなモデルの探索を速度重視パラメータを調節することで可能にする評価関数の改良を行った. また, 圧縮モデルの推論時間と精度の探索によって得られたモデルのパレート曲線を可視化することで, 圧縮のための生徒モデルの達成し得る性能を可視化できる. 達成できる限界値の曲線を可視化することで, 生徒モデルの選択における, 情報量が増加し, 圧縮モデルの高い任意性という問題を改善させる.

2. 関連研究

最適なモデル構造を探索する手法は, ハイパーパラメータ最適化と NAS の 2 つの分野が有名である. これらの分野では, 進化アルゴリズムである遺伝的アルゴリズム, ベイズ最適化による探索が代表的である.

2.1 遺伝的アルゴリズム

遺伝的アルゴリズムは, 複数の個体を世代を通して, 個体同士を組み合わせるなどの操作を行う交叉や個体の要素にランダム性を付与する突然変異を繰り返し, 前の世代よりも評価値が良い, 新しい個体を生成することで最適化を行う手法である [1].

2.2 ベイズ最適化

ベイズ最適化では, ハイパーパラメータ最適化や NAS のモデル構造の組み合わせ問題をブラックボックス最適化問題とみなして, 観測データからガウス過程やランダムフォレストなどの予測モデルを代理モデルとして, 代理モデルの値の更新の期待値が高い有望な点を計算する獲得関数をもとに, 繰り返し探索と値の更新を行うことで, 関数の最適値を探索する手法である.

2.3 ハイパーパラメータ最適化

ハイパーパラメータは, 学習や推論によって変動することのないパラメータであり, モデル構造の重要な要素となる場合もある. ハイパーパラメータ最適化では, ハイパーパラメータの最適な組み合わせを発見する分野である. 代表的なものでは, ガウス過程を代理モデルとし, Expected Improvement (EI) を獲得関数とした, ハイパーパラメータ最適化探索がある [2]. 遺伝的アルゴリズムでの, ハイパーパラメータ最適化探索は [3] がある. 本研究が目的としているような, モデル圧縮のためのトレードオフ関係にある推論時間やサイズと精度を両立するハイパーパラメータ探索を含むモデル構造の探索は, [4] があり, 探索のためのモデルの評価関数は, 値のスケールの異なる圧縮率と損失関数の加重平均となっており, どちらかに比重が偏ってしまうため, 圧縮のための両立したモデルの探索やスケールの差の解析なしに各評価値に対して重視した探索を十分に行えるとは言い難い. また, この論文では, 圧縮のためのハイパーパラメータ探索は, 始まったばかりであると述べられている.

2.4 NAS

NAS は, ハイパーパラメータだけでなく, convolution や pooling などの層のオペレーションやモデル構造をグラフ問題としてオペレーションの繋がり方も考慮したモデル構造の探索を行う分野である.[5] や [6] は, ベイズ最適化によって, モデル構造を探索した研究であり, [7] では, 遺伝的アルゴリズムと類似した働きをする進化的アルゴリズムによる探索手法を提案されている. [8] では, NAS は, 離散空間上の探索であるため, 一般的に勾配法は使えないことを問題視し, 各オペレーションの繋がりを有効巡回グラフ（DAG）として考慮したモデル構造という離散空間上の探索において, 選択を確率的に優劣を表現し, 連続化させることで, 勾配法を適用させることを可能とし, 高速な探索を成功している. また, NAS は, 精度を重視してモデル構造を探索するものが一般的であるが, [9] や [10], [11] のように推論時間を重視するような研究も存在する. 例えば, [11] では, 実際のデバイス上の推論時間を損失関数に掛けたものを損失関数として, 推論時間と損失が適したモデルの探索を行っている. しかし, 損失関数が積である式の特性上, 0 に近づきやすい要素の影響を強く受けてしまうため, 双方が両立したモデルの探索は行いにくい.[12] や [13],[14] は, 圧縮のための, NAS であり, 本研究のモチベーションに近い. [13][14] は蒸留での圧縮ではなく, 枝刈りでの圧縮であり, [12] は [11] 同様に, 推論時間と精度の双方が両立した探索を行いにくい.

3. 提案手法

3.1 生徒モデルの生成

f_i は, 各 convolution 層 i のフィルター数とし, 0, 16, 32, 64, 128, 256, 512 のいずれかの値を取る. 本研究では, リスト $[f_1, f_2, f_3, , , , f_N]$ を入力として, フィルター数が設定された各 convolution 層を積み重ねてモデルを構築する. このとき, f_i が 0 の場合は, 層としてみなさない。また, フィルター数以外の構造やパラメータに関しては, VGG16 や ResNet の形式に従う. VGG16 形式では, 13 層の convolution 層のフィルター数を 13 個のパラメータによって設定し, モデルを生成する. ResNet 形式では, skip connection を行う 2 層は, 同数のフィルター数として, 入力層含む 33

層の convolution 層のフィルター数を 17 個のパラメータによって設定し, モデルを生成する.

3.2 蒸留による圧縮

蒸留とは, 圧縮先となる生徒モデルが訓練データの正解ラベルから学習するのではなく, 圧縮元となる精度の良い教師モデルの出力を正解ラベルとして, 学習する方法である. 教師モデルの誤りも含めた出力を正解ラベルとして学習するため, 生徒モデルは, 教師モデルと似た出力をするようになり, 訓練データの正解ラベルのみで学習した同等のモデルよりも精度が向上する傾向がある. [15] より, ディープニューラルネットワークへの有効性も示されている.

本研究では, [15] 同様に出力層への入力 z_i を温度 T で割り, 出力層の softmax 関数を適用させることで, 各出力の確率である q_i をソフト化する. ソフト化された教師モデルの出力 $\tilde{q}_i^t$ と生徒モデルの出力 $\tilde{q}_i^s$ をクロスエントロピー誤差と温度の二乗 T^2 の積によって, 蒸留損失 L_{KD} を計算する. 上記の softmax による, 勾配は $1/T^2$ 倍されるため, 蒸留損失 L_{KD} の計算時には, T^2 を乗算している.

$$q_i = \frac{\exp(\frac{(z_i)}{T})}{\sum_j \exp(\frac{(z_j)}{T})} \tag{1}$$

$$L_{KD} = -T^2 \sum_k \tilde{q}_k^t \log \tilde{q}_k^s \tag{2}$$

最終的な生徒モデルの損失関数 L は, [16] で蒸留学習において, 一般的であるとされている, 学習ラベルに対するクロスエントロピー誤差 L_{cls} と, 教師モデルの出力との誤差である蒸留損失 L_{KD} の組み合わせたものと同様のものである. パラメータも同様の $T = 4$, $\alpha = 0.9$ を用いる.

$$L = \alpha L_{cls} + (1 - \alpha) L_{KD} \tag{3}$$

3.3 正規化

本研究では, 各モデルのテストデータに対する推論時間と精度に対して, それぞれ正規化した値を評価値と定義する. 正規化は, 異なる値のスケールを統一させて, 同等の評価を行うことを可能とする技術である. 計算環境によって, 推論時間と精度の値のスケールは異なり, 最終的な評価が片方の影響を大きく受け, 公平に評価することができなくなる場合がある. 推論時間と精度を両立するモデルを探索することを目標としているため, 正規化は必須だと考えられる. 提案手法では, 教師モデルの圧縮であるため, 最大値を教師モデルの推論時間や精度と設定できるため, min-max normalization を実施する。最小値は, 標準的な探索では 0 として設定する. 各評価の最低値を設定する場合は, 最低値を最小値と設定し, 最低値以下の評価を 0 として出力するような処理を行うことで最低値を設定した探索も実現する.

3.3.1 推論時間の評価値

0 から 1 の範囲を取る関数に, 速度重視パラメータ r_t をかけた $j_t(t)$ を推論時間の評価値と定義する. 推論時間の評価関数は, 教師モデルの推論時間を t_{max}, ある条件での生徒モデルの最小の推論時間 t_{min}, 評価する生徒モデルの推論時間 t を引数とし min-max normalization を行い速度重視パラメータをかけることで, 生徒モデルの推論時間を 0 から r_t の範囲で評価し評価値を出力する. 推論時間の評価関数は, 評価する生徒モデルの推論時間が設定した最小値に近づくほど r_t に近づき, 設定した最大値に近づくほど 0 に近づく. また, 設定した最大の推論時間 t_{max} を超えた場合, 評価関数 $j_t(t)$ は 0 を出力し, 設定した最小の推論時間 t_{min} を下回った場合は r_t を出力する。

$$j_t(t) = \begin{cases} r_t & t < t_{min} \\ \frac{r_t(t_{max} - t)}{t_{max} - t_{min}} & t_{min} \leq t \leq t_{max} \\ 0 & t > t_{max} \end{cases} \tag{4}$$

3.3.2 精度の評価値

0 から 1 の範囲を取る関数に, 速度重視パラメータの逆数 $\frac{1}{r_t}$ をかけた $j_a(a)$ を精度の評価値と定義する. 精度の評価関数は, 教師モデルの精度 (accuracy) を a_{max}, ある条件での生徒モデルの最小の精度 a_{min}, 評価する生徒モデルの精度 a を引数とし min-max normalization を行い速度重視パラメータの逆数をかけることで, 生徒モデルの精度を 0 から $\frac{1}{r_t}$ の範囲で評価し評価値を出力する. 精度の評価関数は, 評価する生徒モデルの精度が設定した最小値に近づくほど 0 に近づき, 設定した最大値に近づくほど $\frac{1}{r_t}$ に近づく. また, 設定した最大の推論時間 a_{max} を超えた場合, 評価関数 $j_t(t)$ は $\frac{1}{r_t}$ を出力し, 設定した最小の推論時間 a_{min} を下回った場合は 0 を出力する。

$$j_a(a) = \begin{cases} 0 & a < a_{min} \\ \frac{a - a_{min}}{r_t(a_{max} - a_{min})} & a_{min} \leq a \leq a_{max} \\ \frac{1}{r_t} & a > a_{max} \end{cases} \tag{5}$$

3.4 調和平均

図 1 のように, 調和平均は, 式の特性から各要素の値が大きいほど値が大きく, 各要素の少なくとも片方の値が 0 に近いほど 0 に近づくような算術方法である. 調和平均の値を大きくするためには, 両要素の値を大きくする必要がある. また, 図 2 のような様々なトレードオフ関係に対して, 両方の値が大きくなるような点を最大値と出力することから, トレードオフ関係の双方を同時に評価することができる手法の一つだと考えられる. 図 2 より, 相加平均の式は, (1) の場合, 全ての値が同等となるためトレードオフ関係にあるものに対して, 比較するための評価をすることができない. (2) と (3) の場合, 片方の要素の値が小さくなったとしても, 片方の要素の値が大きくなるような点で, 相加平均の値も大きい値を示す. そのため, 相加平均は, 双方が優れたような点を最大値としないため, 両立を目的とした場合, 探索には適していないと考えられる. 実際に, トレー

ドオフ関係にある適合率と再現率の双方を同時に評価するために, それぞれの調和平均である F 値が一般的に用いられる.

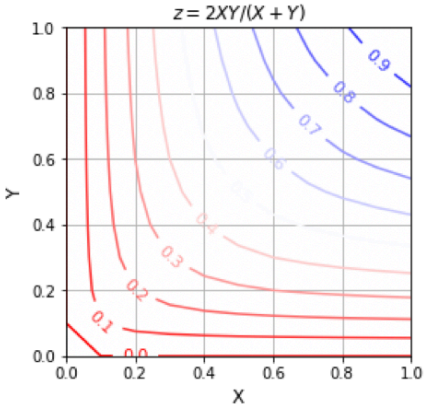

図 1 調和平均のグラフ化

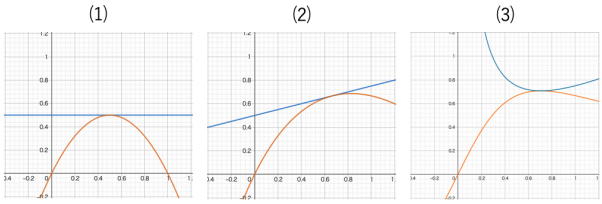

図 2 (1) $x_2 = 1 - x_1$, (2) $x_2 = 1 - 0.5x_1$, (3) $x_2 = 1/x_1$ で表される様々な, トレードオフ関係に対する, 相加平均 ($y = \frac{x_1+x_2}{2}$) と調和平均 ($y = \frac{2x_1x_2}{x_1+x_2}$) をグラフ化している. 青線が相加平均, 赤線が調和平均によるグラフを表す.

以上より, 3.3 で定義した推論時間と精度の評価値の調和平均を推論時間と精度の両立評価関数とする.

$$J(a,t) = \frac{2j_t(t)j_a(a)}{j_t(t) + j_a(a)} \tag{6}$$

また, 調和平均によって, 得られた推論時間と精度の両立評価関数は, 速度重視パラメータ r_t を大きくすることで, 図 3 のように推論時間の評価値の影響力を大きく, 精度の評価値の影響力を小さくなる. 同様に, 速度重視パラメータ r_t を 1 より小さくすることで, 推論時間の評価値の影響力を小さく, 精度の評価値の影響力を大きくなる. 各評価値の影響を受けた両立評価関数の最大となるモデルを探索するため, 推論時間重視, 精度重視のような探索を可能とする.

3.5 ベイズ最適化

ベイズ最適化は, ブラックボックス関数に対して, 最大値や最小値を効率よく探索する手法であることから, ハイパーパラメータ最適化や NAS の分野で活用される. 本研究では, ガウス過程によって予測されたモデルに対して, 獲得関数は EI とした, 改善量の期待値が最大となるような空

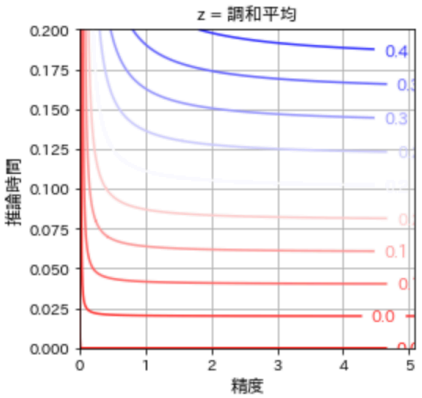

図 3 $r_t = 5$ によって, 精度軸のスケールが大きく, 推論時間軸のスケールが小さくなった調和平均のグラフ

間に対して積極的に探索を行う手法を用いて, 両立評価値が最大化するような蒸留学習された生徒モデルを効率的に探索する. 獲得関数 EI の改善量は, 現在の最小値 $f_{best}(x)$ よりも小さい未知の評価値である $f_{t+1}(x)$ に対して, 0 以上の値を持つ.

$$I(x) = \max\Big\{0, f_{best}(x) - f_{t+1}(x)\Big\} \tag{7}$$

一般的な改善量は, 最小値を求める探索を行う. 本研究は, 生徒モデルの両立評価値のマイナス値を出力することで, 最大値の探索を行う.

最適化する両立評価値は, 実行によって値が固定とは限らない推論時間を含み, 予測値に確信度として, 誤差を許容する不確実性の表現を行えることから, 適していると考えられるため採用した.

4. 実験

本章で, 提案手法の有効性について検証する. 実験用のデータセットは, cifar10 を使用する. cifar10 は, 飛行機, 自動車, 鳥, 猫, 鹿, 犬, カエル, 馬, 船, トラックの 10 種類からなる画像データセットである. 全ての実験を通して, 圧縮元となる教師モデルは, 学習データの正解ラベルによって学習させた VGG16 を使用する. モデル構造の探索の指標となる評価値の推論時間と精度 (accuracy) は, cifar10 の半分のテストデータによって計算される. また, モデル構造の探索時の生徒モデルの蒸留学習では, ミニバッチ数は 128, epoch 数は 10 とする. 最終的な性能評価は, 最適化された生徒モデルに対して, 教師モデルと同様の学習環境と同 epoch 数である 200epoch で学習を行った生徒モデルをモデル構造の探索に使用しなかった半分のテストデータで評価し, 比較する. ベイズ最適化では, 初期値の探索数 150, 探索数の上限 150 の合計 300 回を探索数とする. また, 比較手法として, 進化アルゴリズムの一つである, 遺伝的アルゴリズムを用いる. 個体数 20, 世代数 20 とし, 探索数が約 300 回ほどとなるように, 突然変異の確率と交差の確率などを SEED 値の元設定する. 遺伝的アルゴリ

ズムの実験では, DEAP のライブラリ, ベイズ最適化の実験では, Gpyopt のライブラリを使用して実装する. 実験環境での GPU は, Tesla V100-SXM2 を使用し, CPU は, MacBook Pro (13-inch, 2020, Four Thunderbolt 3 ports) を使用する.

4.1 では, ベイズ最適化によって, 正規化された各評価値の調和平均である両立評価関数が最大となるような生徒モデルの探索を行う. 圧縮先となる生徒モデルには, VGG 形式と ResNet 形式のモデルを使用する. 提案手法により最適化された VGG 形式の生徒モデルは, VGG16 の convolution 層のフィルター数が 1/2 となるモデルと遺伝的アルゴリズムによる, 同程度の約 300 回の探索を行い, 最適化されたモデルとの性能比較を行う. 提案手法により最適化された ResNet 形式の生徒モデルは, ResNet18 との性能比較を行う.4.2 では, 遺伝的アルゴリズムの手法との比較を行う. 探索モデルは, 各 convolution 層のフィルター数である離散パラメータの非常に複雑な探索空間であることもあり, 進化アルゴリズムの中から遺伝的アルゴリズムを採用した. 今回, 提案手法におけるベイズ最適化の有効性を検証するため, 遺伝的アルゴリズムを用いて, 同様の探索を行い性能と探索過程の比較からベイズ最適化の有効性を示す.4.3 では, 推論時間や精度を重視するため, 速度重視パラメータ r_t を変更して, 推論時間が短く, 精度が高くなるようなモデルを探索できているか検証を行う. また, 探索する過程においても, 提案手法の重視する評価値が高くなる空間の探索を積極的に行えているかの検証も行う. 4.4 では, 各評価値の重視探索によって, 最適化された生徒モデルの精度を特徴量として, 推論時間をガウス過程回帰によって予測して, グラフ化する。これは, 推論時間と精度の限界値となるような, 探索によるパレート最適の予測線と見なすことができるため, 生徒モデルの選択における情報量を増加させる.

4.1 BO による探索の有効性

推論時間も精度も偏った重視をせず, $r_t = 1$ による, 両立評価関数を利用する. VGG 形式の生徒モデルをベイズ最適化によって探索したモデルの性能を表 1 で示す. 表 1 の 1/2 モデルは, VGG16 の convolution 層がフィルター数が 1/2 となるモデルであり, GAVGG は, 遺伝的アルゴリズムによって探索されたモデルであり, BOVGG は, ベイズ最適化によって探索したモデルである. また, 評価値は両立評価値であり, 精度は accuracy, GPU 時間は GPU 上の半分のテストデータの推論時間, CPU 時間は CPU 上の半分のテストデータの推論時間である. 両立評価値は, BOVGG が最も優れた値を得た. 教師モデルと比較すると, 精度が 1 %劣るものの, GPU 上では 3.223 倍、CPU 上では 9.525 倍の高速化に成功している.

ResNet 形式の生徒モデルをベイズ最適化によって探索したモデルの性能を表 2 で示す. ResNet18 は, Pytorch などでも学習済みモデルが提供されている有名なモデルであり, 多様な ResNet の中でも比較的高速である. 両立評価値は, VGG での探索同様に, BOResNet が最も優れた値を得た. GPU 上では 1.789 倍、CPU 上では 4.396 倍の高速化に成功している. 今回の実験環境においては, ResNet18 よりも推論時間と精度を両立する優れた圧縮のための生徒モデルだと言える.

表 1 生徒モデルが VGG 形式の性能比較

モデル	評価値	精度	GPU 時間	CPU 時間
教師モデル	x	0.883	0.390	46.874
1/2 モデル	0.649	0.881	0.202	13.465
GAVGG モデル	0.736	0.897	0.163	12.986
BOVGG モデル	**0.812**	0.872	**0.121**	**4.921**

表 2 生徒モデルが ResNet 形式の BO 性能比較

モデル	評価値	精度	GPU 時間	CPU 時間
教師モデル	x	0.883	0.390	46.874
ResNet18	0.505	0.853	0.249	24.920
BOResnet	**0.625**	0.845	**0.218**	**10.662**

4.2 遺伝的アルゴリズムとの比較

SEED 値を変えて, ベイズ最適化と遺伝的アルゴリズムによる生徒モデルの探索を 9 回行う. 図 4 は, Harmonic mean が探索されたモデルの調和平均による両立評価値を意味しており, 9 回探索された最適な生徒モデルの両立評価値を示す. 直感的にも, BO で表されるベイズ最適化は, GA で表される遺伝的アルゴリズムによって, 最適化された生徒モデルの両立評価値よりも優れている割合が多いことがわかる. 各手法による最適化モデルの平均両立評価値を表 3 に示す. 表 3 より, 最適化モデルの平均両立評価値でも, ベイズ最適化が優れている.

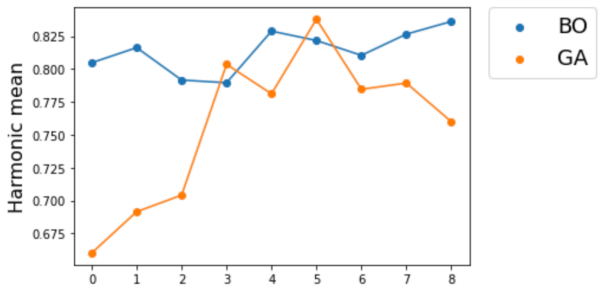

図 4 GA と BO による複数回の探索による両立評価値のグラフ化

図 5 は, ベイズ最適化と遺伝的アルゴリズムによって, 探索の過程で生成された各生徒モデルの性能を示すものである. time が推論時間, accuracy が精度を意味する. グラフの右下辺りに図示されるモデルは, 推論時間が短く, 精度の

表 3 BO と GA の探索モデルの平均両立評価値の比較

探索手法	平均両立評価値
GA	0.756
BO	**0.814**

高い, 優れた両立評価値を出すモデルであり, グラフの左上辺りに図示されるモデルは, 推論時間が長く, 精度の悪い, 劣った両立評価値を出すモデルである. 図 5 では, ベイズ最適化は, 遺伝的アルゴリズムの探索よりも両立評価値が高い空間での探索を重点的に行っているのがわかる. 本研究での実験設定上のような現実的な探索数の場合, ベイズ最適化が, 遺伝的アルゴリズムよりも比較的優れていると言える.

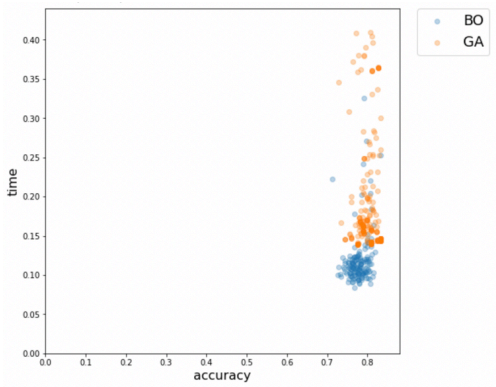

図 5 BO と GA の探索の過程で生成された各生徒モデルの推論時間と精度による性能の散布図

4.3 重視探索の有効性の検証

VGG 形式の精度を重視する $r_t = 1/100$ と $r_t = 1/10$ による探索と, 推論時間を重視する $r_t = 3$ と $r_t = 10$ による探索によって得られた生徒モデルの性能を表 4 で示す. 表 4 より, 精度重視による探索で得たモデルは, $r_t = 1$ で探索したモデルよりも精度が高く, 推論時間重視による探索で得たモデルも $r_t = 1$ で探索したモデルよりも推論時間が短く優れていた.

表 4 推論時間や精度に重視した探索による、最適生徒モデル

モデル	精度	GPU 時間	CPU 時間
教師モデル	0.883	0.390	46.874
BOVGG モデル	0.872	0.121	4.921
BOVGG(rt = 1/100)	**0.899**	0.277	26.0186
BOVGG(rt = 1/10)	0.882	0.124	10.565
BOVGG(rt = 3)	0.850	0.094	3.425
BOVGG(rt = 10)	0.664	**0.081**	**2.244**

また, 図 6, 図 7 は, 各評価値の重視探索の過程で生成された各生徒モデルの推論時間と精度による性能の散布図を示す. 赤枠が重視する評価値が高い空間であり, 赤矢印方向ほど評価値の値は, 高くなる. 散布図から分かる通り, 探索されるモデルは, 重視したモデルが存在する空間に対して, 探索回数が増加している.

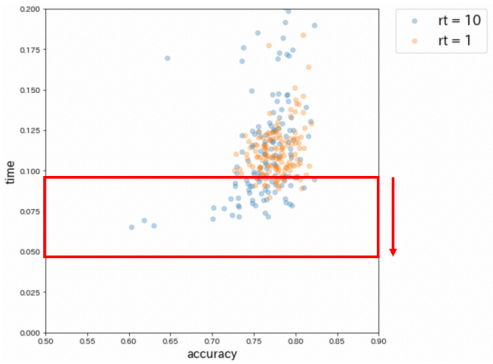

図 6 推論時間重視の探索の過程で生成された各生徒モデルの推論時間と精度による性能の散布図. 青が $r_t = 10$ での速度重視探索モデル, 橙が $r_t = 1$ での同重視探索モデル.

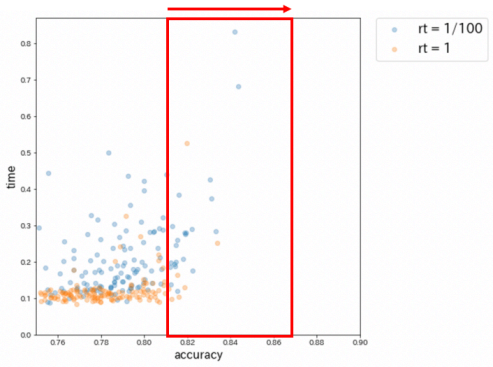

図 7 精度重視の探索過程で生成された各生徒モデルの推論時間と精度による性能の散布図. 青が $r_t = 1/100$ での精度重視探索モデル, 橙が $r_t = 1$ での同重視探索モデル.

4.4 限界値を示す, 探索パレート最適曲線の可視化

図 8 のように, 各評価値の重視探索によって最適化された生徒モデルの探索パレート最適となるモデルの精度を特徴量として, 推論時間の予測線を引くことで, 探索パレート最適曲線の可視化を行うことができる. 探索パレート最適曲線の可視化により, 予測線であるため誤差はあるものの探索を行えていない空間に対しても, 情報量が増加する. 本研究では, 生徒モデルの任意性を改善する上で, 生徒モデルの情報量を増加させることは重要であるため, 予測への確信度も同時に推定するガウス過程回帰を用いて, 曲線を予測する. 探索パレート最適曲線の予測線によって, 精度が 0.89 を超える生徒モデルは推論時間を大幅に要することがわかる. 上記のように, 生徒モデルが実現できる限界や, トレードオフの関係性が強くなる点も認識でき, 最終的な生徒モデルの選択するための情報量が増加していることから, 生徒モデルの高い任意性という問題が改善される.

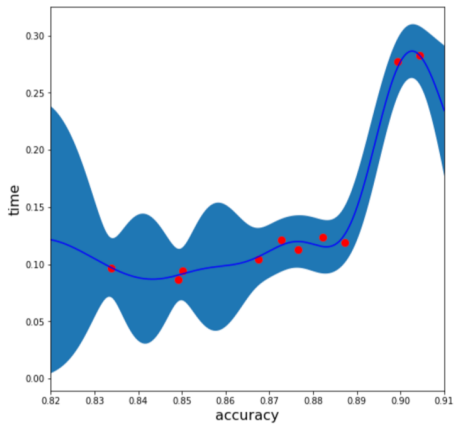

図 8 ガウス過程回帰による, 探索パレート最適曲線

5. 結論

本研究では, 蒸留による生徒モデルの任意性を問題視し, 提案手法により, 推論時間と精度の双方を両立するような生徒モデルの探索と推論時間と精度の片方に重視するような状況に適した生徒モデルの探索に成功した. 各評価値に重視して最適化されたモデルとガウス過程回帰による予測線を用いて, 生徒モデルの限界値を認識することができる, 探索によるパレート最適曲線の可視化を行った. 生徒モデルの選択における情報量が増加し, 本研究で問題視していた生徒モデルの任意性を改善させた. また, 検証として, ベイズ最適化と遺伝的アルゴリズムによる手法の比較も行い, 本研究の実験設定上のような現実的な探索数の場合, ベイズ最適化の方が優れていると結論づけた.

6. 考察

本研究による, 実験や結果の考察をまとめる. 本研究の実験では, GPU を用いて, モデルの探索を行っている. しかし, ニューラルネットワークモデルは, GPU の推論時間と CPU の推論時間の大小関係が必ずしも一致するとは言えない. 実際に, 計算資源に限りがある IoT デバイス等の小型端末に適応させる場合, それぞれの計算資源による推論時間を指標にする必要が生じる可能性がある.

本実験において, 探索の有効性の検証のため蒸留学習の epoch 数を 10 としている. 実用化する場合は, 多量の epoch 数での探索を行う必要がある. 提案手法による, 多量の epoch 数での生徒モデル探索は、膨大な学習時間となるため, Successive Halving[17] のような優れた評価値が望めないようなモデルの探索の打ち切りを行う技術を用いて, 探索効率を向上させる必要がある.

本研究では, ベイズ最適化の探索数を 300 としているが, ResNet 形式の最適生徒モデルは, VGG 形式の最適生徒モデルと比べると両立評価値を大きく下回っている. VGG 形式は, 設定する convolution 層が 13 層, 7 種のフィルター数通りあるため, 7^{13} 通りである一方, ResNet 形式は, 設定する convolution 層が 17 層, 7 種のフィルター数通りあるため, 7^{17} 通りであることから, ResNet 形式の方が探索の難易度が高いと言える. そのため, 生徒モデルの組み合わせの数によって, 適した探索数が存在すると考えられるため, 本研究の実験は, ResNet 形式よりも VGG 形式の生徒モデルが優れているということを示すものではない.

4.3 では, 評価関数に調和平均を採用したことにより, 重視していない評価値が極端に小さくなることを避けていると考えられる. 相加平均の式の推論時間の評価値に速度重視パラメータを掛けて, 2 ではなく重みの和で割った加重平均を評価関数とした場合, 各評価値の影響力を大きくして重視探索を行うと, 重視していない評価値が極端に小さいモデルを最適とする場合がある. 表 5 は, 調和平均と加重平均の推論時間を重視した探索によって得られた生徒モデルの性能である. 本研究での圧縮モデルとして適していない生徒モデルが選出されるため問題である. 推論時間が減少していない場合, 圧縮ができておらず, 精度が極端に低下する生徒モデルは, 双方が両立したモデルを探索しているとは言い難いため, 本研究での圧縮モデルとして適していない生徒モデルとは, 全く推論時間が減少していないモデルや, 精度が極端に低下した生徒モデルである.

表 5 生徒モデルを ResNet 形式の BO 探索

モデル	精度	GPU 時間
教師モデル	0.883	0.390
調和平均 (速度重視)VGG	0.664	0.081
加重平均 (速度重視)VGG	**0.413**	0.050

図 8 は, 推論時間を超えるような推論時間の評価値が 0 となる, 生徒モデルに対して, 調和平均での探索は一回のみ行っているが, 加重平均での探索は頻繁に行われている. 図 8 が示すように, 調和平均は, 加重平均よりも圧縮モデルとして適していない生徒モデルが存在する空間に対して積極的に探索を行わないという経験的知見を得た. 相加平均の場合, 片方の評価値が低いような圧縮モデルとして適していない生徒モデルに対して, 他方の評価値が高い場合, 最終的な両立評価値は低い値を出力しないため, 獲得関数 EI による, 探索の改善量の期待値が低い値を取らない. 圧縮モデルとして適していない生徒モデルの空間に対しても探索を続けてしまうのは, 改善量の期待値が低くないからだと考えられる. 一方, 調和平均の場合, 片方の評価値が低いような圧縮モデルとして適していない生徒モデルは, 他方の評価値が高くても, 最終的な両立評価値は低い値を出力するため, 改善量の期待値も低いことで頻繁に探索することを避けていると考えられる. そのため, 探索においても調和平均が優れている可能性がある.

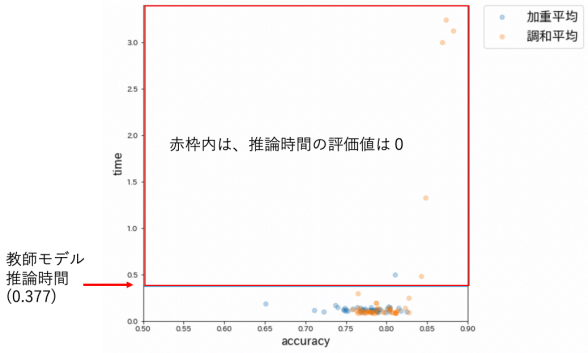

図 9 相加平均と調和平均の探索への有効性の比較

参考文献

[1] Singh, Nanua: *Systems approach to computer-integrated design and manufacturing*, (1996): 775-775.

[2] Snoek, Jasper and Larochelle, Hugo and Adams, Ryan P: *Practical bayesian optimization of machine learning algorithms*, arXiv preprint arXiv:1206.2944 (2012).

[3] Leung, Frank Hung-Fat and Lam, Hak-Keung and Ling, Sai-Ho and Tam, Peter Kwong-Shun: *Tuning of the structure and parameters of a neural network using an improved genetic algorithm*, IEEE Transactions on Neural networks 14.1 (2003): 79-88.

[4] Ma, Xingchen and Triki, Amal Rannen and Berman, Maxim and Sagonas, Christos and Cali, Jacques and Blaschko, Matthew B: *A bayesian optimization framework for neural network compression*, Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[5] Mendoza, Hector and Klein, Aaron and Feurer, Matthias and Springenberg, Jost Tobias and Hutter, Frank: *Towards automatically-tuned neural networks*, PMLR, 2016.

[6] White, Colin and Neiswanger, Willie and Savani, Yash: *Bananas: Bayesian optimization with neural architectures for neural architecture search*, arXiv preprint arXiv:1910.11858 (2019).

[7] Jozefowicz, Rafal and Zaremba, Wojciech and Sutskever, Ilya: *An empirical exploration of recurrent network architectures*, International conference on machine learning. PMLR, 2015.

[8] Liu, Hanxiao and Simonyan, Karen and Yang, Yiming: *Darts: Differentiable architecture search*, arXiv preprint arXiv:1806.09055 (2018).

[9] Yang, Tien-Ju and Howard, Andrew and Chen, Bo and Zhang, Xiao and Go, Alec and Sandler, Mark and Sze, Vivienne and Adam, Hartwig, and H. Adam.: *Netadapt: Platform-aware neural network adaptation for mobile applications*, Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[10] Tan, Mingxing and Chen, Bo and Pang, Ruoming and Vasudevan, Vijay and Sandler, Mark and Howard, Andrew and Le, Quoc V: *MnasNet: Platform-aware neural architecture search for mobile*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[11] Wu, Bichen and Dai, Xiaoliang and Zhang, Peizhao and Wang, Yanghan and Sun, Fei and Wu, Yiming and Tian, Yuandong and Vajda, Peter and Jia, Yangqing and Keutzer, Kurt: *Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[12] He, Yihui and Lin, Ji and Liu, Zhijian and Wang, Hanrui and Li, Li-Jia and Han, Song: *Automl for model compression and acceleration on mobile devices*, Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[13] He, Yihui and Lin, Ji and Liu, Zhijian and Wang, Hanrui and Li, Li-Jia and Han, Song: *Automl for model compression and acceleration on mobile devices*, Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[14] Dong, Xuanyi and Yang, Yi: *Network pruning via transformable architecture search*, arXiv preprint arXiv:1905.09717 (2019).

[15] Hinton, Geoffrey and Vinyals, Oriol and Dean, Jeff: *Distilling the knowledge in a neural network*, arXiv preprint arXiv:1503.02531 (2015).

[16] Carlsson, Sven A: *Developing information systems design knowledge: a critical realist perspective*, The Electronic Journal of Business Research Methodology 3.2 (2005): 93-102.

[17] Jamieson, Kevin and Talwalkar, Ameet: *Non-stochastic best arm identification and hyperparameter optimization*, Artificial Intelligence and Statistics. PMLR, 2016.