

進化的な深層学習の構築に関する研究

1 はじめに

機械学習の成功は目覚ましく、広い分野に適用され多くの研究がなされている。例えばディープニューラルネットワーク (deep neural network; DNN) は、音声・画像・自然言語を対象とした問題で高い性能を示し、強い関心を集めた。DNN は言葉通り層を深く重ねるため、層の種類、パラメータ、数、層同士の接続など多数のハイパーパラメータを持つ。

問題に合わせて適切なパラメータを選択しなければ良好な精度が出ない一方で、そのネットワークやアーキテクチャの設計に明確な指針はなく非常に困難である。従来では、専門知識を持つ人が手作業で設計し、学習した結果を参照して繰り返し修正していた。

自動化された機械学習 (Automated Machine Learning; AutoML) と呼ばれる分野は、アーキテクチャ全体を対象とした最適化を目指す。本研究では AutoML の開発を行う前段階として、論文の調査を行った。まず AutoML の分類を見た後、その例である Neural Architecture Search(NAS) と Auto Augment について紹介する。

2 要素技術

2.1 AutoML

自動化された機械学習 (Automated Machine Learning; AutoML) は特定の技術ではなく、機械学習モデルの設計を自動化する全般的な手法、または概念を指す。

問題ごとにパラメータを適切に設定する必要があるというパラメータ設定問題の解決を目的とする。確率的なブラックボックス最適化であるこの問題の解決によって、

- 時間コストの削減・性能の向上
- アルゴリズムの評価、比較のためパラメータの最適性の影響を緩和
- パラメータが与える影響の知識の必要性を排除

の 3 つの利益が得られる。

AutoML はそのアルゴリズムによって、以下の 3 つの主要カテゴリと 6 つのサブカテゴリに分類される [1]。

主要カテゴリ 構造や構成による分類

1. 単純な生成・評価法
生成段階で候補となる設定を生成、評価段階で評価して最適な設定を見つける手法。
2. 反復的生成・評価法
少数の設定を生成して、最も優れたものをみつけ、反復的に生成する新しい設定の指針とする手法。
3. 高レベルの生成・評価方法
高レベル生成機構として既存の自動パラメータチューニング手法や探索手法を持ち、少数精鋭の設定を生成し評価段階では慎重に評価する手法。

サブカテゴリ 評価方法による分類

1. 繰り返し評価法
複数回の評価を、平均することなどで評価する手法。
2. F-Racing
統計的に劣る評価の設定を段階的に排除し、有望な候補に計算を集中する手法。繰り返し評価より効率的になる。
3. インテンシフィケーション (強化)
問題のリストで候補設定と暫定設定の評価を次々比較し、劣る場合は途中で排除し、そうでなければ候補が暫定設定となる手法。
4. シャープニング
少ないテスト数で評価を始め、将来性のある設定のテスト数を 2 倍にすることで素早く探索できる手法。
5. アダプティブキャッピング
有望でない設定の実行を中断して計算量を削減できる手法。

2.2 NAS

従来の機械学習では手作業によって設計されたモデルをデータセットで学習して重みの最適化を行うが、アーキテクチャの設計には、高度な専門知識と手作業による構築が必要である。Neural Architecture

Search(NAS)[2] は、ニューラルネットワークのアーキテクチャ自体を最適化する。

アーキテクチャの探索は 3 つの段階で行う。まず最初にコントローラと呼ばれる再帰型ニューラルネットワーク (Recurrent Neural Network; RNN) で、アーキテクチャのハイパーパラメータを生成する。例えば畳み込み層を利用するネットワークでは、レイヤーごとにフィルタの高さ・幅、ストライドの高さ・幅、フィルタ数が必要となる。次にハイパーパラメータから子ネットワークを構築し、通常のように重みを訓練して検証データセットの精度を得る。最後に得られた精度で報酬を計算し、方策勾配法 (Policy gradient method) によってコントローラのネットワークを更新する。これらの手順を繰り返すことで、子ネットワークのアーキテクチャが最適化される。

NAS では子ネットワークとして、畳み込みネットワークと RNN の探索を行い従来のネットワークより高い精度と少ないパラメータ数を達成したが、数百台の GPU と 1 ヶ月の時間がかかり計算量に問題があった。以降の研究では計算量の削減を目的とした手法が提案された。

佐藤 怜ら [3] の研究では、NAS を改良した先行研究と比較して、より高速なアーキテクチャ探索アルゴリズムを開発した。

特徴としてコントローラで生成していたカテゴリカルなアーキテクチャパラメータから、アーキテクチャを確率分布で連続的に表現し、分布推定を行う手法に変えることで、微分可能となった。これによって貢献度と呼ぶアーキテクチャの構成要素ごとの評価を導入して、一様な更新ではなく構成要素ごとの勾配で更新をすることができ、高速化ができた。

また NAS ではアーキテクチャ候補ごとに重みを学習し直していたが、ネットワークの重みを再利用することで大幅な学習時間の削減が達成できた。

重みの再利用

1. 冗長にネットワーク構造を決めておく。(あるノードはそれ以前のノード全てに接続可能とする)
2. 重みを学習して、各エッジ、各演算子ごとに重みを保存
3. アーキテクチャ (接続するか? + 演算子) を探索

2.3 AutoAugment

画像識別問題で利用されるデータセットでは、画像データの左右反転や回転、色相や輝度の操作すること

表 1: 実験の設定

Optimizer	SDG
momentum	0.9
lr	0.001
Loss	Cross Entropy Loss
batch size	64
epoch	100

でデータ数を増やすデータ拡張 (Data Augmentation; DA) が精度の向上に効果的であることが分かっている。しかし操作の種類と強度、操作の順序の組み合わせは膨大であり、現在データ拡張の実装は手動で設計されているため試行錯誤に時間がかかる。AutoML の一つである Auto Augment[4] はデータ拡張の空間を探索し、最高の検証精度となる最適な方策を見つける。

3 実験

Auto Augment はデータセットを対象とした操作を最適化するが、データセットの持つ特徴を解析する実験として、以下の予備実験を練習的に行った。

画像データの分類問題を考える。異なるクラスラベルを持つサブデータセット集合と、それらを 1 つに結合したデータセットを作成する。集合内の各サブデータセットで学習した各分類器を用いて、結合したデータセット全体のクラス分類を行う。

関連手法として、複数のネットワークの多数決をとり 1 つのモデルとするアンサンブル学習がある

3.1 データセット

データセットは、10 クラスの画像データが含まれる Cifar-10[5] を利用した。サブデータセットは、Cifar-10 から 10 クラスの内いくつかを抜き出し作成した。

3.2 モデルの構築

分類器には、16 層の畳み込みニューラルネットワーク (Convolutional Neural Network; CNN) を用いた。実験 1 では作成したサブデータセットで、5 クラス分類を学習した後、同様に学習した他のネットワークと組み合わせで 10 クラス分類を行う。この関係を図 1 に示した。

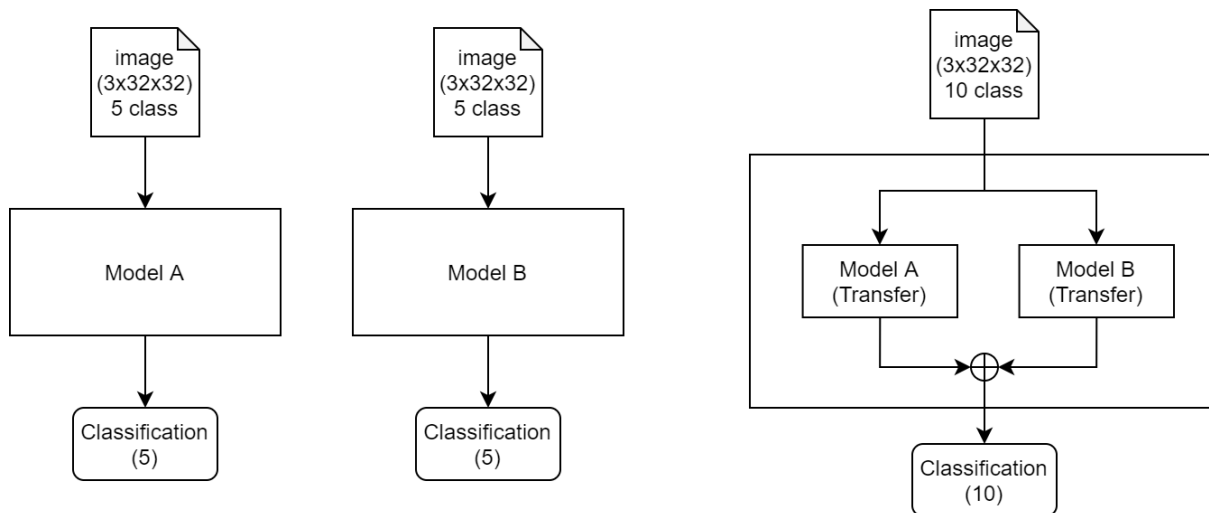


図 1: モデルの簡略図 () 内はデータの次元数

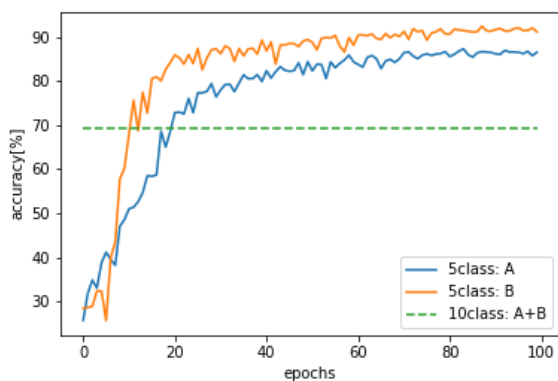


図 2: 実験 1 の 5 クラス分類器の正答率

3.3 実験 1

実験 1 ではデータセットを 2 つの異なる 5 クラスサブデータセットに分割した。このデータセットはそれぞれ 16,000 枚の画像データが含まれる。クラスを表す番号の前半 (airplane, mobile, bird, cat, deer) で学習したモデルを A, 後半 (dog, frog, horse, ship, truck) で学習したモデルを B とする。

結合したモデル A + B は, 10 クラス分類問題を解く。結合の際は 10 クラスのデータセットを A, B に入力し, 得られた出力の次元を 10 クラスに拡張して和を求めた。

表 2: モデルごとの 7 クラスの振り分け

model	0	1	2	3	4	5	6	7	8	9
A	o	o	o	o	o	o	o			
B	o	o	o	o				o	o	o
C	o				o	o	o	o	o	o

3.3.1 結果

図 2 にモデル A, B のテスト精度の結果を示した。モデル A が 84%, B が 90% となった。この精度はやや低いため, データ拡張による改善が望まれる。

結合モデル A + B の 10 クラス分類の結果は, 70% となった。片方のモデルのみで分類した場合, 40% 程度となるので, データセットのクラス拡張に対応していることが分かる。

3.4 実験 2

あるクラスの分類を複数のモデルで行うことで, 実験 1 より高い精度の獲得を目指した。7 クラスを持つサブデータセットを 3 つ作成し, 学習するモデルをそれぞれ A, B, C とした。各データセットのクラスは, 2 つ以上の分類器でクラスを推定できるように振り分けた。表 2 には, クラスの番号と共に選択したクラスを o で示した。

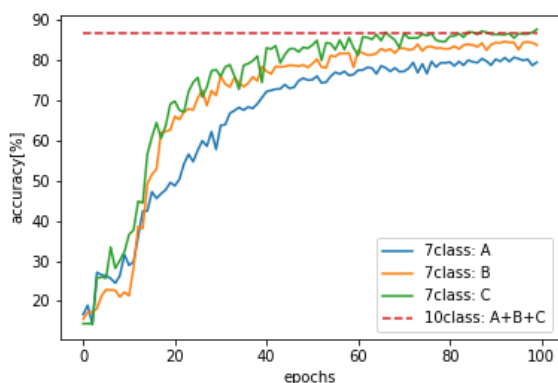


図 3: 実験 2 の 7 クラス分類器の正答率

3.4.1 結果

7 クラス分類の精度を図 3 に示した。

7 クラス分類では、正答率が平均 80%程度で、結合した結果 10 クラス分類では 86.7%となった。クラスが増え問題が困難になっているが、精度が高まる結果となった。

4 考察

実験 1 ではクラスを拡張したデータセットでの、モデルの結合の有効性を見た。実験 2 では分類の結果を重ね合わせることで、テスト精度が非常に高まることが分かった。

色情報の似た画像クラスを含めることで効果的に、形状的特徴を学習するように促すなど、データセットの統合におけるクラスの組み合わせに関する問題を考えたい。

図 2, 3 とともにインデックスが前半のクラスを持つモデルでは、正答率が低い傾向が見られた。再度学習を行っても変わらなかったため誤差の影響ではなく、困難なクラスの分類によって精度が下がっていると考えられる。これはクラスを単純に分割したことによる偏りに原因がある。

様々なクラスの組み合わせパターンで実験することで、クラスの潜在的な識別難度や、類似クラス間の識別における相互作用などを確かめたい。

5 まとめと今後の課題

ネットワークによって自動的にネットワークをメタ最適化する AutoML について調査した。従来では人が行っていた工程を機械学習によって自動化しようという試みがある？一方でネットワークを複数回学習するため計算時間が問題となったが、探索時間の削減を行う様々な工夫が提案された。

実験は～であった。

今後の展望として、らの論文中 [3] のモデルを再現する、ネットワークの構造を探索する NAS ファミリのシステムを実装し、簡単な問題で動作実験を行うことで、NAS の理解を深めたい。

参考文献

- [1] Changwu Huang, Yuanxiang Li, and Xin Yao. A survey of automatic parameter tuning methods for metaheuristics. *IEEE Transactions on Evolutionary Computation*, 24:201–216, 06 2019.
- [2] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. *CoRR*, abs/1611.01578, 2016.
- [3] 佐藤 怜, 秋本 洋平, and 佐久間 淳. 貢献度分配を導入した方策勾配による neural architecture search の高速化. *人工知能学会全国大会論文集*, JSAI2019:2P3J202–2P3J202, 2019.
- [4] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *CoRR*, abs/1805.09501, 2018.
- [5] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).