

## 進化的な深層学習の構築に関する研究

### 1 はじめに（研究背景）

人工知能の成功が目覚ましい人の手が必要専門知識が必要困難である深層学習の広範囲な（メタな）技術のサーベイ

### 2 要素技術

#### 2.1 AutoML

サーベイ [1]

##### 2.1.1 パラメータ設定問題

問題ごとにパラメータを適切に設定する必要がある

#### 2.2 自動パラメータ調整

開発動機

- 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

主要カテゴリ

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
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。
3. 高レベルの生成・評価方法  
高レベル生成機構では既存のチューナーや探索手法から少数精鋭の設定を生成. 評価段階では慎重に評価する

サブカテゴリ

1. 評価の繰り返し  
確率的目的関数最適化問題では複数回評価した平均値が最も分かりやすい方法
2. F-Racing  
評価した統計的に劣る設定を段階的に排除し, 有望な候補に計算を集中. 繰り返し評価より効率的
3. インテンシフィケーション（強化）  
評価途中で候補が暫定設定より劣るなら排除し, そうでなければ次の問題で評価する. 全ての問題で評価されると新しい暫定設定となる
4. シャープニング  
少ないテスト数で評価が始まり, 将来性のあるパラメータはテスト数が2倍になる. 素早く探索できる
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

#### 2.3 NAS

AutoML の一種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.

NAS では, まずコントローラと呼ばれる Recurrent Neural Network(RNN) で, レイヤーの種類やその接続を表すアーキテクチャのハイパーパラメータを生成する. そしてパラメータからビルダーによって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. 得られた精度を報酬として, 方策勾配法によってコントローラのネットワークを更新して, アーキテクチャの探索を行う. 最良の子ネットワークがコントローラが見つけたアーキテクチャとなる. 例えば畳み込み層を利用するネットワークでは, RNN でフィルタの高さ, 幅, スライドの高さ, 幅, フィルタ数をレイヤーごとに予測する.

スキップ接続や他のレイヤを用いたアーキテクチャの複雑性の向上

- アーキテクチャをカテゴリカルから連続的な確率分布に
- ネットワークの重みを再利用して学習を削減
- 見つけたアーキテクチャの重みを再学習する (acc 6 割 → 9 割)

#### 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

得られたセルでは、直前のセルからの入力を恒等写像でそのまま出力していた。

畳み込み層は全て separable convolution を利用している。パラメータ数を減らしたかった？

#### アーキテクチャの表現法

- RNN でパラメータを生成 (カテゴリカル)
- GA で個体表現 (カテゴリカル)
- 演算子の確率分布集合として表現

## 2.4 AutoAugment

[3] AutoML の一種? (編集しよう!) 画像認識の分野では、画像データを少し回転させたり左右反転させたりなどの操作をすることで画像データ数を増やす Data Augmentation (以下、DA) が広く使われています。ただ、どの操作を行うのかというのは試行錯誤で見つけるしかなく時間がかかります。そこで強化学習を使うことで自動的に DA を選択してくれるという AutoAugment が提案されました。

## 3 実験

～に関する実験の前段階として、以下の予備実験を練習的に行った。アンサンブル学習に関連？

表 1: 実験の設定

dataset	cifar10
n data	16,000 / model
task	5, 7, 10 クラス識別
input	image(3x32x32)
output	class(5, 7, 10)
model	CNN(16 層)
optim	SDG (lr=0.001, moment=0.9)
loss	Cross Entropy Loss
batch size	64
epoch	100

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

### 3.1 モデルの構築

前回、5 クラス識別器を 2 つ利用したモデルの構築をしたが、10 クラスの識別が 50% であったため、より適したパラメータを探索し、再び実験して精度の向上を目指した。

#### 3.1.1 5 クラス x2

cifar10 に含まれる 10 クラスを 5 クラスずつに分割した。インデックスの前半 (airplane, mobile, bird, cat, deer) と後半 (dog, frog, horse, ship, truck) で分けることにした。生成した部分データセットを、それぞれモデル A, B で学習した。5 クラス分類ができるモデル A, B を持つ結合モデル A + B を作成し、10 クラス分類の精度を計測した。このモデルの学習と相互関係を図 1 に示した。

結合モデル A + B は 10 クラスのデータセットを A, B に入力し、得られた出力をクラスインデックス順に結合して、出力とする。結合では特別な処理を行わず、そのままのデータを連結した。

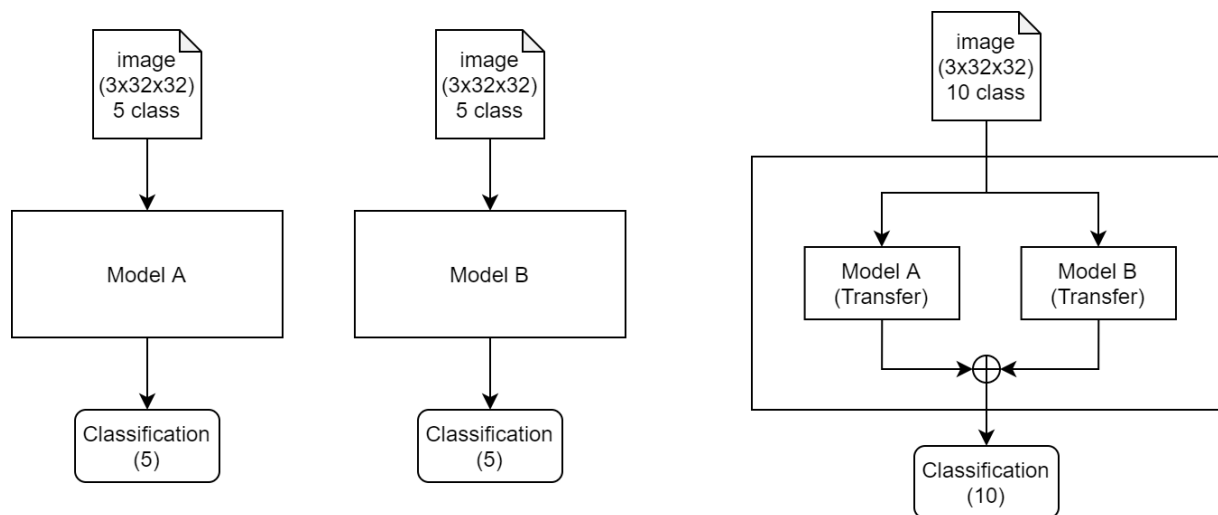


図 1: モデルの簡略図 ( ) 内はデータの次元数

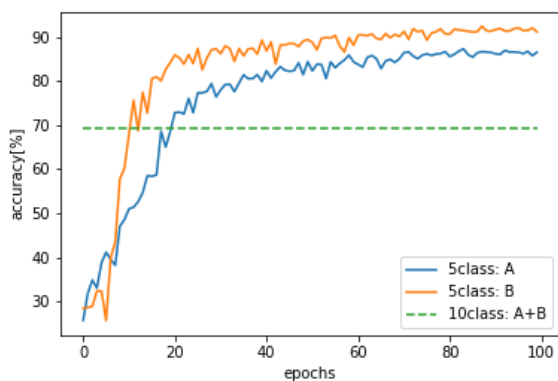


図 2: 5 クラス分類の正答率

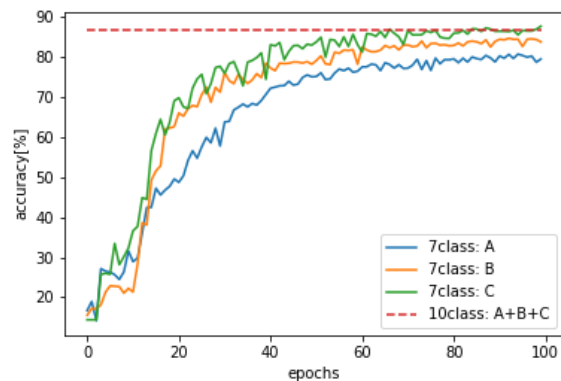


図 3: 7 クラス分類の正答率

### 3.1.2 7 クラス x3

さらに今回は、細川君に指摘してもらったアイデアでも実験を行った。7 クラス分類器を 3 つ組み合わせて、10 クラスの分類を行った。表 2 のように、2 つ以上の分類器で各クラスを推定するように、クラスを振り分けた。

## 3.2 結果

5 クラス分類の精度を図 2 に、7 クラス分類の精度を図 3 に示した。

5 クラス分類の場合、80% ~ 90% の正答率で前回よりも 10% 程向上した。前回のデータ数 2000 よりもデータ数を増やしたことで精度がよかった。結合した結果も、70% (前回+20%) となった。

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。

## 3.3 考察

データ数とバッチサイズを増やして、精度の向上と学習の安定化ができた。特に 5 クラス分類ではあるが、正答率 9 割を超えることができた。データオーギュメントはしていないので、さらに精度を上げることはできると思われる。

図 2, 3 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。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。

分割する組み合わせを自由に替える実装ができたので、様々な組み合わせパターンで実験して、その差異を見たい。特にモデル間の正答率の差が目立ったため、これを埋めるような組み合わせを探したい。

## 4 考察

## 5 まとめと今後の課題

論文の調査をした AutoML とその関連技術についてまとめた

NAS について確率的なアーキテクチャ表現したシステムを構築し簡単な問題を解く小さなネットワークを探索する実験をしたい

## 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] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. CoRR, abs/1805.09501, 2018.