

tdgaCNN における適応度評価手法の検討

1 はじめに

近年、機械学習を用いた画像認識が注目を集めている。その一つに畳み込みニューラルネットワーク (Convolutional Neural Network: CNN) がある。現代の CNN のアーキテクチャは、問題の高度化に伴い複雑になる一方である。数あるアーキテクチャの中から最適なものを探し出すことは困難な組み合わせ最適化問題であり、更にそれを人手で行うことは難しい。そこで、様々な問題を最適化するための効果的な手法である遺伝的アルゴリズム (Genetic Algorithm: GA) を CNN のアーキテクチャ探索に用いる、gaCNN [1] が提案されている。しかしこの手法においては、GA の選択ルールに対する十分な検討がなされていない。適切な選択ルールを用いないと、探索初期に個体の多様性が失われる初期収束が問題となる。そこで、GA の選択ルールに熱力学的選択ルールを採用した熱力学的遺伝アルゴリズム (Thermodynamical Genetic Algorithm: TDGA) を用いて CNN アーキテクチャを最適化する tdgaCNN が提案され、従来手法との比較を行なった結果、その優位性が確認された。

先行研究では、tdgaCNN の探索フェーズにおける適応度を、各個体を 1 エポックだけ学習させた場合の精度としていた。本実験では、最終的により良い個体を得ることを目的とし、様々なエポック数で個体を学習することで、適応度の評価方法を検討する。

2 要素技術

2.1 畳み込みニューラルネットワーク

画像認識分野ではさまざまな深層学習手法が提案されてきているが、畳み込みニューラルネットワーク (Convolutional Neural Network: CNN) はその中でも特に顕著な成功を収めている手法である。CNN のアーキテクチャは畳み込み層、プーリング層、全結合層の 3 種類の層と、それに伴う活性化関数から構成され、それらの組み合わせ、および各種パラメータが識別精度を左右する。

2.2 遺伝的アルゴリズム

遺伝的アルゴリズム (Genetic Algorithm: GA) は、生物の進化からヒントを得た最適化手法である。問題の解を個体とみなし、その遺伝子を表現する配列に交叉、突然変異、選択といった操作を繰り返し適用する。そして、ある個体がどの程度優れているかの指標である適応度を各個体について計算し、高いものを次世代に残し低いものを淘汰する。これを複数世代繰り返すことによって、最終的に良い解を得ようとする。

2.3 gaCNN

gaCNN は、GA によって CNN のアーキテクチャを自動最適化する手法である。FashionMNIST・MNIST で検証した結果、競合 16 手法のうち 12 手法の精度を上回ったこと、従来は数日以上かかっていた最適化を 1 日以内に圧縮することに成功したことが報告されている。

2.4 熱力学的遺伝アルゴリズム

熱力学的遺伝アルゴリズム (Thermodynamical Genetic Algorithm: TDGA) は、個体の多様性維持を目的として、GA の選択ルールに熱力学的選択ルールを採用した手法である。先行研究において、従来手法よりも優れた CNN アーキテクチャの獲得が報告されている。

2.5 可変長遺伝子型熱力学的選択ルール

可変長遺伝子型熱力学的選択ルールは、tdgaCNN で提案された選択ルールである。温度 T において熱平衡状態にあるシステムでは、状態の定常分布は自由エネルギー

$$F = \langle E \rangle - HT \quad (1)$$

を最小にする分布になることが知られている。ここで $\langle E \rangle$ はシステムの平均エネルギー、 H はエントロピーである。(1) 式の右辺第 1 項は、系がエネル

ギー最小化という本来の目的を追求する項，第 2 項は系の状態の多様性を維持する項と解釈することができる．つまり，同一個体が多い状態よりも様々な個体が存在している状態のほうが乱雑であり，エントロピーが高いと考える．そして，これらを温度 T で調和させた状態が実現されているものとする．この考え方のもと温度 T を適切に設定することで，GA における探索初期に個体の多様性が失われるという初期収束問題を解決することができる．

また，(1) 式中の H は以下の式で表される．

$$H = H_D, \quad H_D = \frac{\sum_{s \in S \setminus p} L(p, s)}{S_S} \quad (2)$$

ここで p は新たに選択する個体， S は選択済みの個体集合に p を加えた集合， $L(x, y)$ は個体 x と個体 y における遺伝子配列の層に対する Levenshtein 距離， S_S は集合 S の要素数である． H_D の値が大きいほど，個体群内の各個体間の類似度が低いということになる．つまり，個体には多様性があるということを示す．

3 提案手法

3.1 tdgaCNN

TDGA を用いて CNN のアーキテクチャを探索する手法が tdgaCNN である．tdgaCNN の流れを以下に示す．

1. 遺伝子初期化戦略に基づき個体を生成し，初期母集団とする．
2. 母集団の個体の適応度を評価する．
3. 母集団から個体を母集団の一定割合選択し，集合 S を形成する．
4. S からランダムに選択した 2 個体に交叉操作を適用し，集合 C に加えるという操作を C の大きさが母集団の一定割合になるまで繰り返す．
5. S と C からランダムに選択した個体に突然変異操作を適用し集合 M に加えるという操作を M の大きさが一定割合になるまで繰り返す．
6. $S + C + M$ を次世代の母集団とする．
7. 2 から 6 を世代回数だけ反復する．
8. 最終世代で最も適応度が高い個体を本学習する．

3.2 遺伝子符号化

個体の遺伝子配列は CNN の層の順番に対応し，遺伝子座は層と活性化関数の組を保持する．層の候補は畳み込み層，プーリング層，全結合層の 3 つで，活性化関数の候補は ReLU, tanh, Sigmoid 関数の 3 つである．また，層の種類に応じた複数のパラメータ値を持つ．

3.3 遺伝子初期化戦略

層数，全結合層の最小数及び最大数を事前に定め，以下のように遺伝子を初期化する．

1. 層数の大きさを持つ空の遺伝子配列を用意する．また，全結合層の数を最大数と最小数の範囲からランダムに決定する．
2. 先頭に畳み込み層とランダムな活性化関数のペアを追加する．
3. ランダムな層とランダムな活性化関数のペアを追加する．ただし，前の層が全結合層である場合は，追加する層の種類は全結合層とする．
4. 遺伝子長が最大層数，あるいは全結合層の数が所定の数に達するまで 3. を繰り返す．

3.4 適応度評価

本学習フェーズの学習データを全て探索フェーズの学習データとして使用し，学習データに対する所定のエポック数だけの分類精度を適応度とする．先行研究では，エポック数を 1 として適応度を評価していた．また生成されたモデルが無効である場合は，致死個体として適応度を 0 とする．

3.5 選択

可変長遺伝子型熱力学的選択ルールを採用する．

3.6 交叉

選択された 2 個体について，同じ種類の層を持つ遺伝子座を前から順に交換することで新たな 2 個体を生成する．このとき，活性化関数も同時に交換する．対になるものがない遺伝子座はそのまま維持する．

3.7 突然変異

以下の4種類の操作から1つを等確率で適用する.

- 層と活性化関数のランダムなペアをランダムな位置に追加
- ランダムな位置にある遺伝子座を削除
- ランダムな遺伝子座の層パラメータすべてと活性化関数をランダムに変更
- ランダムな位置に畳み込み層, ReLU, プーリング層のブロックを追加

4 実験概要

tdgaCNN の先行研究では探索フェーズにおいて, CNN を1エポックのみ学習させて適応度を算出していた. 本実験では適応度算出のためのエポック数が n_i である世代の数を g_i , c を定数としたとき,

$$\sum_i n_i g_i = c \quad (3)$$

となるように, 様々な世代数とエポック数の組み合わせで tdgaCNN を実行した場合, 最終的な最良個体に及ぼす影響を検証した. (3) 式に従うことで, 各試行において計算量をほとんど一定に保って実験することができる. 本実験では $c = 80$ として実験した.

4.1 実験用初期個体群の作成

1度の tdgaCNN 実行にかかる時間を短縮するために, 後の実験の初期個体群として, tdgaCNN によってある程度適応度の平均を高めた個体群を作成した. 作成の際はランダムな100個体を, 1エポックの学習での精度を適応度として20世代探索をした.

4.2 データセット

データセットには FashionMNIST を用いた. 図1にデータセットの分割方法を示す. 7万枚の画像を, 初期母集団作成時の探索フェーズでの訓練データ3万枚, 実験1, 2の探索フェーズでの訓練データ3万枚, そしてテストデータ1万枚に分割して使用した.

4.3 実験1

探索フェーズにおいてエポック数を固定して実験をした. 具体的には (3) 式において, $i = 1$ のみとして, 80世代1エポック, 40世代2エポック, 20世代4エポック, 16世代5エポック, 10世代8エポック, 8世代10エポックの計6パターンで tdgaCNN を実行した.

4.4 実験2

実験1では探索フェーズでのエポック数を固定して実験した. 実験2では, 探索フェーズの序盤から終盤にかけてエポック数 n を増加させた場合と, 減少させた場合の2パターンにおいて実験した. 表1に具体的な世代数とエポック数の推移を示す.

表1: 実験2の条件

(a) n を増加させる場合

i	g_i	n_i
1	16	1
2	8	2
3	4	4
4	2	8
5	1	16

(b) n を減少させる場合

i	g_i	n_i
1	1	16
2	2	8
3	4	4
4	8	2
5	16	1

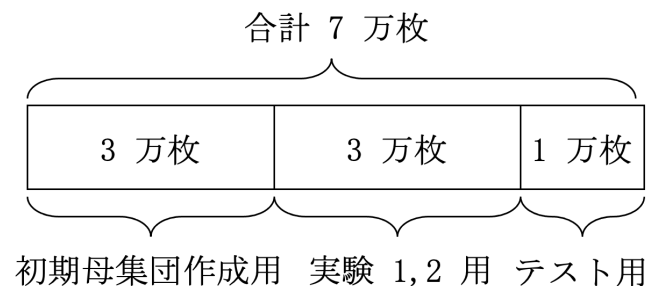


図1: データセットの分割

5 実験結果

5.1 実験用初期個体群の作成

探索前の個体群の適応度の平均値は 63.26 % で, 20世代探索後の個体群の適応度の平均値は 85.64 %

表 2: 実験 1, 2 実験条件

個体数	100
層数	10
最小全結合層数	1
最大全結合数	3
選択割合	40
交叉割合	40
突然変異割合	20
本学習エポック数	100
探索バッチサイズ	24
本学習バッチサイズ	16
学習率	1e-4
最適化手法	Adam
温度	0.04

となっており、上昇していることが確認できる。そのため当初の、後の実験の時間短縮という目的の達成が見込まれる。よって以下の実験 1, 2 では、ここで作成した個体群を初期個体群として採用した。

5.2 実験 1 の結果

表 3 に作成した初期母集団を用いて、 $c = 80$ として実験をした結果を示す。表 3 の値は、各試行の最終世代において最も適応度が高かったものを 100 エポック本学習させたときの最良識別精度である。表 3 より、エポック数 n と世代数 g の組 (n, g) が、 $(80, 1)$, $(40, 2)$, $(20, 4)$, と、 n が増えるにつれて識別精度が良くなり、 $(n, g) = (16, 5)$ の時にピークを迎え、その後は識別精度は悪化していったことが読み取れる。これはエポック数を適切に増やして学習すると、将来的な性能が予測しやすくなるためと考える。

表 3: 実験 1 の結果

世代数	エポック数	識別精度 [%]
80	1	92.04
40	2	92.13
20	4	92.26
16	5	93.09
10	8	90.00
8	10	92.29

5.3 実験 2 の結果

表 4 に実験 2 の結果を示す。表 4 から、探索フェーズ終盤にエポック数を大きくした方がより良い個体を得ることができたということが確認できる。探索終盤に本学習に相対的に近いエポック数で学習することで、本学習したときにより良い精度となる個体を選択できたと考えられる。

表 4: 実験 2 の結果

条件	1 回目 [%]	2 回目 [%]	平均値 [%]
n 増加	91.91	92.39	92.15
n 減少	91.34	91.96	91.65

6 まとめと今後の課題

本実験では、tdgaCNN における適応度評価を様々な方法で実験し、それが得られる CNN の精度にどのような影響を及ぼすかを調査した。その結果として、1 エポックのみ学習させた時の精度を適応度としたときよりも、適切なエポック数を定め、そのエポック数で学習させた時の精度を適応度とした時の方が良い個体得られる場合があるという結果が得られた。また、探索序盤よりも探索終盤に適応度評価のための学習エポック数を増やす方が、より良い個体得られるということも確かめられた。

今後の課題として、試行回数を増やすことで、適応度評価手法ごとの信頼区間を調査することが挙げられる。また本実験で、適応度評価のための学習エポック数を変化させると、得られる個体の良し悪しに影響を及ぼすということは確認できたが、そのエポック数をどのように最適化するかということについては解明できていない。様々な実験条件下で、エポック数を最適化する手法を提案することも、今後の課題である。

参考文献

- [1] Raphael de Lima Mendes, Alexandre Henrick da Silva Alves, Matheus de Souza Gomes, Pedro Luiz Lima Bertarini, and Laurence Rodrigues do Amaral. gacnn: Composing cnns and gas to build an optimized hybrid classification architecture. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pp. 79–86, 2021.