

## 進捗報告

### 1 今週やったこと

GA を用いた DataAugmentaion

### 2 実験 1

前回に引き続き G A を用いた DataAugmentation の実験を行った。

#### 2.1 実験データ

実験データは cifar10 を用いて、事前学習では epoch 数 300, train\_data を各ラベル 5000 枚の計 50000 枚使用し、G A で学習する際は epoch 数 100, train\_data は各ラベル 200 枚のオリジナルとそれらすべてを DataAugmentaion したものとを合わせ計 4000 枚とし、test\_data は共に 10000 枚とした。また事前学習での accuracy は 0.8475 である。

#### 2.2 遺伝的アルゴリズム

##### 2.2.1 探索空間

探索する水増し操作として画素値操作 (Sharpness, Posterize, Brightness, Autocontrast, Equalize, Solarize, Invert, Contrast, ColorBalance), 変形操作 (Mirror, Flip, Translate X/Y, Shear X/Y, Rotate) の 16 種類の操作であり、今回はそれらすべてを個別にどの程度強くかけるかおよびどの順序でかけるかということを探索する。各操作についての強度の最大最小を設定し、それを -100% から 100% まで 25% ずつ 11 段階の度合いとする。ただし、Autocontrast, Equalize, Invert, Mirror については適用するか否かであるためパラメータが 0 以上で適用するとした。強度は 0 から 5 の整数値を持つ 15 個の遺伝子を実数値コーディングによって表現する。また、適用順序に関しては同様に 15 個の遺伝子を持つ順列コーディングによって表現する。確率は 10% ごと 11 段階の実数地コーディングによって表現する。つまり、探索空間は  $2^5 * 11^{11} * 15! * 11^{16}$  となる。

##### 2.2.2 選択

選択について、エリート選出によって最も適応度の高い 2 つの個体を選択する。なお、この二つは後述する交叉、突然変異は受けずに次の世代に追加する。残りの選出にはトーナメント選出を用いた。トーナメント選出は集団の中から任意の数 (トーナメントサイズ) の個体のうち最も適応度の高い個体を選出し次の世代に追加する。今回トーナメントサイズは 2 とした。

##### 2.2.3 交叉

強度、確率を表す染色体については 2 点交叉、順序を表す染色体については部分写像交叉を用いた。2 点交叉は一对の親染色体をそれぞれ同じ場所で三分割し中央の染色体を入れ替えて交叉を行う。部分写像交叉は親遺伝子を二分割し入れ替える際重複をなくす交叉法で、重複のあった遺伝子について、それに該当した重複する遺伝子座を見つけ、それに対してとなっているもう一方の親の遺伝子を参照する。

##### 2.2.4 突然変異

強度、確率を表す染色体について、対象となる遺伝子の値を各 50% の確率に 1 増減させ、順序を表す染色体について、染色体の一部を逆順にする操作か、染色体を二つに分け前後を入れ替える操作のいずれかを行うものとした。

### 2.3 パラメータ

表 1 に学習パラメータを示す。表 2 に GA の設定

表 1: 学習パラメータ

optimizer	Adam
learning rate	0.001
loss function	categorical_crossentropy
batch size	128

を示す。

表 2: 実験パラメータ

個体数	20
世代数	50
交叉率	0.9
突然変異率	
強度, 確率 (遺伝子ごと)	0.06
順序 (染色体ごと)	0.1

### 2.3.1 結果

図 1 に accuracy の最良値及び平均値の推移を示す. 最終的な最良値は 0.8691 となった. 適用確率を追加

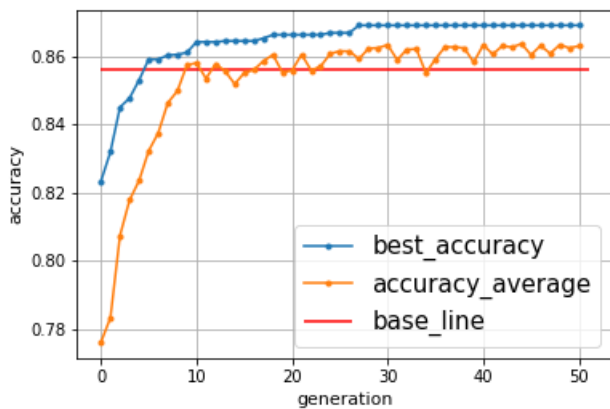


図 1: accuracy の推移

前の最良値は 0.8676 であり, あまり改善があったとはいえない.

## 3 実験 2

実験 1 で得られた個体を用いてアンサンブル学習を行うことで精度を高めることを試みた.

### 3.1 実験 2.1

表 3 に学習パラメータを示す.

元のデータ数は 50000 とした. まず, 上位 3 個体を用いて, 2 倍に水増して学習させ, 予測したものの平均をとり, 最終的な予測値とした. また, 対照実験は上位 3 個体のいずれかを適用する操作で 6 倍に水増して学習, 予測を行った. 表 4 に実験結果を示す.

アンサンブル学習は他のものと同様に 1% ほどの改

表 3: 学習パラメータ

optimizer	Adam
learning rate	0.001
loss function	categorical_crossentropy
batch size	128
epoch	100

表 4: 実験結果

1st	0.9044
2nd	0.9006
3rd	0.9037
ensemble	0.9141
control experiment	0.8963

善が見られ, アンサンブル学習の効果が表れていることが分かる.

次に表 5 に上位 10 個体を用いてアンサンブル学習を行った結果を示す. ただし, 10 個体のうちどの個体を用いるかを全探索によって調べ上げ最良のものとした. また, 各個体の学習についての accuracy について最良のものではなく最終のもので行った. 色を付けたものがアンサンブルに使われた個体で

表 5: 実験結果

1st	0.9044
2nd	0.9006
3rd	0.9037
4th	0.8386
5th	0.8541
6th	0.8624
7th	0.8608
8th	0.8565
9th	0.8740
10th	0.8596
ensemble	0.9205

ある. このことから上位個体だけよりもさらに複数の個体を組み合わせることで多少の改善が行われることが分かる. また, 上位 3 個体について GA で非常に良い個体が選ばれていたことが伺える. 一方でそれ以外の個体は順位通りとはいかないような

accuracy となっており，これは個体の学習時に学習データをランダムにとってきているのでそれに合わせた個体になっていると考えられる．このことから，個体の適応度を算出するのにデータ数を多くする，あるいは複数回の学習の平均をとらなければならないことが考えられる．

### 3.2 実験 2.2

さらに表 6 に上位 20 個体を用いてアンサンブル学習を行った結果を示す．ただし，20 個体のうちのどの個体を用いるかを全探索によって調べ上げ最良のものを示す．また，今回は元データ 5000 枚を 2 倍に水増しして行った．色を付けたものがアンサンブルに使われた個体である．

表 6: 実験結果

1st	0.7982	2nd	0.8149
3rd	0.8055	4th	0.8107
5th	0.8075	6th	0.8048
7th	0.7997	8th	0.8123
9th	0.8101	10th	0.8064
11st	0.8041	12nd	0.7966
13rd	0.8091	14th	0.8087
15th	0.7980	16th	0.8066
17th	0.8072	18th	0.8094
19th	0.7977	20th	0.8028
ensemble		0.8568	

アンサンブル学習の効果自体は非常によく見て取れるが，バグったのか accuracy が全体を通して小さくなってしまった．

## 4 今後の課題

- 適応度の計算の際に複数回学習を行うのかデータ数を増やしたほうがいいのか
- アンサンブル学習を最終的に行うことを前提とした個体選択ができないのか