

## 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。

変更点は各 transform の適応確率を 0% から 100% までは 10% ずつ計 11 段階を導入した。

また、強度の遺伝子について前回までは 0 から 5 をとり、その強度に  $[-1, 1]$  のランダムな値をかけて適用強度としていたが、transform のランダム性を取り除きました、一部において負の変換 (例えば rotate であれば 90 の回転にいたいし時計回りと反時計回りがある) を考慮し -5 から 5 をとるものとした。

#### 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% ずつ分 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 段階の実数値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、それに對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 の設定を示す。

表 1: 学習パラメータ

optimizer	Adam
learning rate	0.001
loss function	categorical_crossentropy
batch size	128

表 2: 実験パラメータ

個体数	20
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

## 2.4 実験 1.1

### 2.4.1 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 0.8671 となった。適用確率を追加

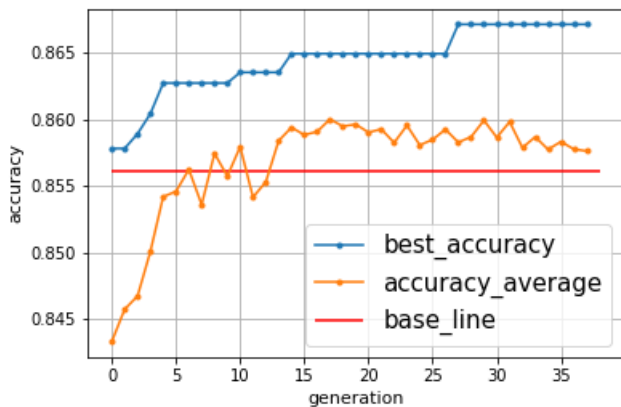


図 1: accuracy の推移

したとはいえど、1 回の学習に使われる枚数自体は 4000 枚で固定となってしまうので、前回と大差がないのではないかと考えられる。

## 2.5 実験 1.2

実験 1.1 は前回同様水増ししたものを学習したのに対し、実験 1.2 では generator を用いてミニバッチごとに変換を行った。またこの時元データが 2000 枚をミニバッチ 128 枚、31 ステップで 1epoch のうち 3968 枚を学習させる。

### 2.5.1 結果

図 2 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 0.8692 となった。実験 1.1 に比べ

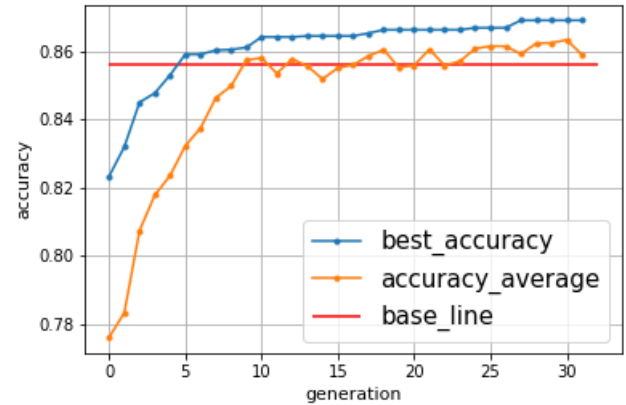


図 2: accuracy の推移

改善が見られたかどうかデータの違いが影響したとも考えられるので改善したとは言い難い。

また、実験時間に関して、1 個体の学習時間は実験 1.1 は 90 秒、実験 1.2 は 100 秒程度であった。また generator を使用しているため、メモリは実験 1.2 のほうが実験 1.1 の半分程度に抑えられていた。

表 3: 対照実験

transform	accuracy
なし (同じ画像二枚ずつ)	0.86043
shearX	0.8586
shearY	0.8592
translateX	0.8628
translateY	0.8589
rotate	0.8610
color	0.8605
posterize	0.8577
solarize	0.8313
contrast	0.8601
sharpness	0.8606
brightness	0.8630
autocontrast	0.8594
equalize	0.8570
invert	0.8333
mirror	0.8624
flip	0.8366

### 3 実験 2

前回の DataAugmentation が有効であることを示す対照実験として、表 tb:taisho 各 transform に対し変換を行いその結果 10 回の平均をいかに示す。

だいたい 0.8600 前後であり、solarize, invert, flip はあまり良い結果ではないことが分かる。前回の実験では最良値は 0.8676 であり、あまり大差がないことが分かる。

### 4 今後の課題

モデルや実験パラメータの改良