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ＧＡを用いたアンサンブル学習のための 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，ＧＡ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，Autoconstrast，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した．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 点交叉は一対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．

2.2.5　多様性維持

多様性を維持するために，上記 3 つの操作 (選択，交叉，突然変異) を行った集団に対し，適用順序を表す染色体について一致するものが 3 つ以上あれば，それが 2 つになるように一部の個体を突然変異させたうえで次の世代の集団とした．

2.3　実験 1

2.3.1 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
epoch size	30

を示す．

表 2: 実験パラメータ

個体数	15
世代数	25
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

2.3.2 結果

図 1 に accuracy の最良値及び平均値の推移を示す．表 3 にアンサンブル学習の各世代の最良値を示す．

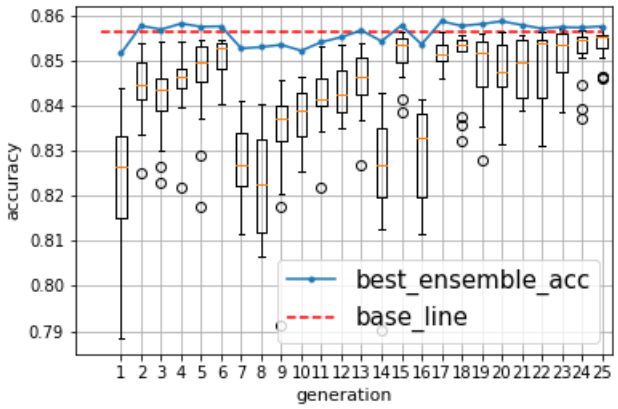

図 1: accuracy の推移

図 1 の箱ひげ図よりから程度精度は揃ってきていることが分かる．一方で各世代のアンサンブル学習についてあまり精度が上がっていない．また，前回の学習では精度が 0.8676 となっているので，全体的な精度自体は低い．記載はしていないが世代を追うごとに，適用順序については制限をかけることでバラバラではあるが，適用確率，強度は 8 割程度揃っていた．

表 3: 実験結果

1st	0.8576	2nd	0.8568
3rd	0.8581	4th	0.8574
5th	0.8575	6th	0.8526
7th	0.8529	8th	0.8534
9th	0.8521	10th	0.8540
11st	0.8551	12nd	0.8566
13rd	0.8542	14th	0.8577
15th	0.8535	16th	0.8586
17th	.8576	18th	0.8580
19th	0.8586	20th	0.8578
21th	0.8570	22th	0.8573
23th	.8572	24th	0.8575
25th	0.8578		

2.4 実験 2

実験 1 で得られた 1,25 世代を用いてデータ数を増やして実験を行った

2.4.1 パラメータ

表 4 に学習パラメータを示す．

表 4: 学習パラメータ

optimizer	Adam
learning rate	0.001
loss function	categorical_crossentropy
batch size	128
epoch	30

元のデータ数は 20000 とし，DataAugmentation で 2 倍にして学習を行う．

2.4.2 結果

表 5，表 6 に結果を示す．

アンサンブル学習について 25 世代の個体を使うよりも，1 世代の個体を使う方が精度が高くなった．

表 5: 1 世代

1st	0.7943	2nd	0.8232
3rd	0.7861	4th	0.4543
5th	0.5574	6th	0.6032
7th	0.7725	8th	0.7953
9th	0.8042	10th	0.7850
11st	0.7675	12nd	0.7498
13rd	0.8101	14th	0.8182
15th	0.8155		
ensemble		0.8770	

表 6: 25 世代

1st	0.8162	2nd	0.8434
3rd	0.8373	4th	0.8343
5th	0.8377	6th	0.8096
7th	0.8359	8th	0.8385
9th	0.8243	10th	0.8388
11st	0.8289	12nd	0.8394
13rd	0.8346	14th	0.8165
15th	0.8388		
ensemble		0.8732	

2.5 考察

今回の実験では，適用順序の染色体の重複数を制限することで順序の多様性は確保できた一方で，適用確率や強度についてはの多様性は保たれていない．そのため順序が逆でも同等になる変換がある可能性から，個体同士自体の多様性は低い可能性がある．そのため，実験 2 では多様性の高い 1 世代目のほうがアンサンブル学習がうまくいったのではないかと考えられる．表 5 のようにアンサンブル学習について個々の精度が高いほど良いのかもしれないが，より時に精度が低いものも混ぜる必要があると考えられる．そのため今回の実験において適応度は各個別の精度であり，次の世代に持ち越すのにアンサンブル学習について全く考慮していないものとなっていたので，個々の精度に加えてアンサンブル学習において良い精度を出した時に選出された個体に対しある程度の重みを付加する必要があると思う．

3 今後の課題

- 適応度の計算についてアンサンブル学習での結果を考慮したものを試してみる