

進捗報告

1 今週やったこと

- GA の実験

2 FixMatch の変換操作の探索

2.1 設定

画像変換は {AutoContrast, Brightness, Color, Contrast, Equalize, Identity, Posterize, Rotate, Sharpness, ShearX/Y, Solarize, TranslateX/Y} の 14 種 (一つは恒等変換) を用い, 各遺伝子はその遺伝子座に対する変換を RandAugment に用いるかいないかについて 0/1 の値を持つビットエンコーディングとした. また, FixMatch における弱及び強変換に対し持つので遺伝子長は 28 とした.

選択はサイズ 2 のトーナメント選択, 交叉には一様交叉, 突然変異は遺伝子座ごとに対立遺伝子に置換されるように設定した.

表 1, 2 に実験の設定を示す.

表 1: 実験 1 の GA の設定

個体数	40
交叉率	1.0
突然変異率	0.03

また得られた個体について最終世代における上位 10 個体を用いて学習した.

2.2 結果

図 1 に適応度の推移を, 図 2 最終個体における各 transform の累積を示す.

また表 3 テストに対する識別率を示す. 前回と同様探索されたものは過学習を起こし, 従来の RandAugment を超えることはできなかった. 変換操作の種類が従来のもので 14 種なのに対し, 探索されたものは平均 5 種類程度であり訓練させる画像の多様性が非常に少ないものとなってしまっていることが原因であると考えらるが, バグでないか再度調査が必要な気がする. また, 今回は個体数が大きくしたためか前回に比べ収束は少ない. 一方で, 前回よ

表 2: 実験 1 の設定

model	WideResNet16-2	
data set	cifar10	
train	labeled	50
	unlabeled	49750
valid	200	
test	10000	
事前学習		
batch size	labeled	32
	unlabeled	32 * 7
augment	labeled	RandAugment
	weak	Identity
	strong	Identity
optimizer	SGD(lr=0.05,momntum=0.9)	
num_iterations	5000	
GA の評価		
batch size	labeled	16
	unlabeled	16
augment	labeled	RandAugment
num_iterations	3000	

く選ばれていた AutoContrast が少ないといったことがみられる.

表 3: テスト識別率

iteration	従来	探索個体
5000	0.268	0.357
10000	0.460	0.287

3 Contrastive Learning の実験

3.1 設定

Contrastive Learning の実装及び GA をからめた実験をした. 個体について, 遺伝子は 0-9 までの整数値のラベルをもち, search する枚数である 50 が遺伝子長である. 選択はトーナメント選択 (k=3), 交叉は 2 点交叉, 突然変異はランダムな別の値に置換

表 4: 実験 2 の GA の設定

個体数	40
交叉率	1.0
突然変異率	0.1

表 5: 実験 2 の設定

model	Encoder	WideResNet50
	Projection head	2 層 MLP(shape:2048to512)
	classifier	MLP(shape:2048to10)
data set	cifar10	
train	labeled	250
	unlabeled	49700
	serach	50
事前学習		
batch size	64	
optimizer	Adam(lr=1.0 * 10 ⁻³ ,momntum=1.0 * 10 ⁻⁶)	
epochs	75	
GA の評価		
batch size	16	
loss	Cross Entropy Loss	
optimizer	Adam(lr=5.0 * 10 ⁻⁴ ,momntum=1.0 * 10 ⁻⁶)	
epochs	25	
valid	250(:labeled)	

では元論文に比べ Encoder の事前学習は十分でない
といえ、また初期個体が今回ランダムであるため、
もう一度実験を行う予定である。

4 RandAugment に scheduler を導入した

4.1 設定

GA の画像変換手法の探索において探索された
ものが iteration が少ない範囲で識別率がよかったこ
とから DataAugment でカリキュラム学習のような
ことができないかと考えた。

従来の RandAugment は各変換に対し最大最小
が決められており、その範囲で強度が一様分布になっ
ている。今回は強度の分布 Y について次のように設

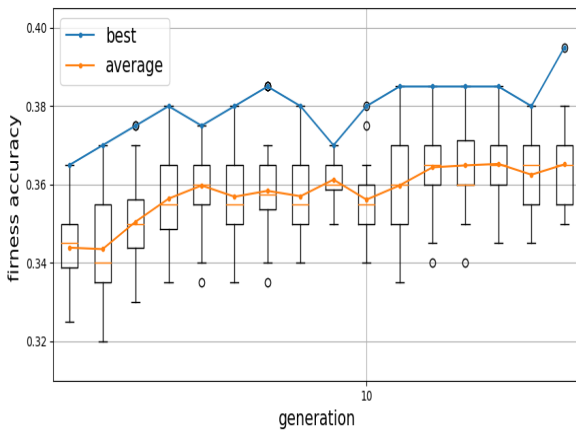


図 1: 適応度の推移

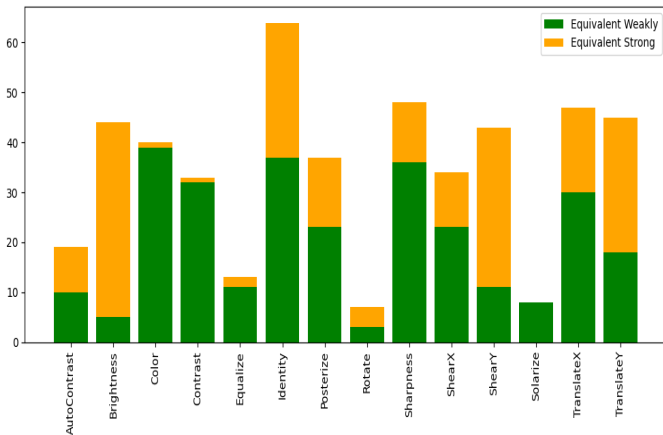


図 2: 最終個体

するものとした。表に事前学習の設定を、表 4, 5 に
設定を示す。

3.2 結果

時間がなかったため表などはありません。全て正
解ラベルを入れた時、識別率が 60% で、最終世代の
最良個体は識別率 47.2% でラベルの正答率は 42% で
あった。これまでの探索と同様に正解ラベルではな
いもので識別率を上げることがありやはり GA で
ラベルを探索することは難しいといえる。一方で、
ランダムなラベルからここまでの正答率を出せたの
は、一個体の学習速度が短く世代あたりの個体数を
確保できたからだと考えられる。また、今回の実験

定することとした．

$$Y = \begin{cases} (1 - 2t)x + t & (-1 \leq x \leq 0) \\ -(1 - 2t)x + t & (0 \leq x \leq 1) \\ 0 & (\text{上記以外}) \end{cases} \quad (1)$$

$$0 \leq t \leq 1 \quad (2)$$

t を学習のイテレーション数に応じて変化させた．現在のイテレーション数の最大イテレーション数に対する割合を $iter$ としたとき，以下のものを用意した

- $t = a$ (a は定数, $a = 0.5$ で一様分布)
- $t = iter$ (linear)
- $t = \tanh(4iter - 2)/2 + 0.5$ (tanh)
- $t = (2iter - 1)^3 + 0.5$ (cube)

4.2 結果

図 3, 4 に結果を示す．

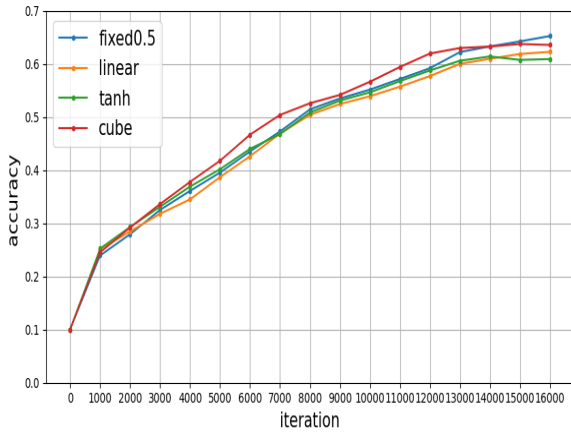


図 3: test_accuracy

最終的には従来のものに抜かされてはいるものの，cube は他のものよりも少し速い学習が行われていたと言えそう．またこのことから学習最初期に限った話では FixMatch の強変換は強度の低いものがよいと考えられ，GA における探索で時間上イテレーション数が小さくしていたため，それゆえの前の結果だったと思われる．

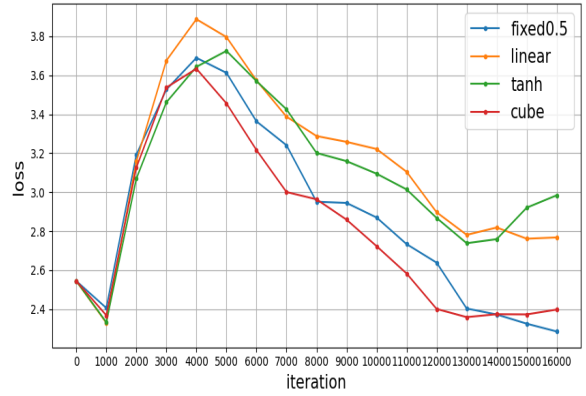


図 4: test_loss

5 来週の課題

- 実験設定の改良
- SimCLR を用いた実験