

進捗報告

1 今週行ったこと

先週と同様に、Googlecolab を用いて、Cifar10 の CNN の以下の実装を行った。また、プログラムが間違っており、A と B でのデータの交換がうまく行えていなかったため修正した。今回は、各訓練を十分に行ってから交換という手順を踏んだ。（以下のアルゴリズムの 3. 4. での実験の訓練回数を増加）

1. データを 10000 個取り出す
2. データを A と B にランダムに二分割する
3. A→train,B→test で実験（A→train,B→test としたとき AtoB と表記する）
4. B→train,A→test で実験（B→train,A→test としたとき BtoA と表記する）
5. 3. と 4. でミスしたデータを調べる
6. A をテストした時、失敗したデータ B をテストした時、成功したデータをランダムに M 個交換（自分で指定）
7. 3. に戻る

上記のアルゴリズムを繰り返すと A → 簡単、B → 難しいデータが集まると考えられる。つまり、accuracy は AtoB_test が低く、BtoA_test が高くなると予想される。モデルの概要は以下である。

クラス	10 クラス分類 (Cifar10)
データ数	10000
input	image(32 × 32 × 3)
output	class(10)
モデル	CNN
optimizer	Adam
損失関数	categorical_crossentropy

以下に、変数を定義する

- random: A,B のシャッフルの random_state
- lr: 学習率
- batch: バッチサイズ
- epoch: 3.4 での訓練回数（エポック数）
- change: 1 回の交換で交換するデータの数
- repe: 3.4.5.6. の繰り返し数

まず、どれくらいのエポック数が必要か判断するため、全く交換を行わないで、訓練を行っていくと図 1, 図 2 のようになった。2つの図より、25 回程度でテスト精度は横ばいになっているので各訓練のエポック数を 25 回程度にすることに決定した。

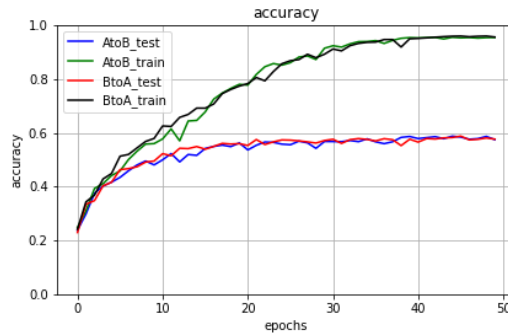


図 1: random=0,lr=0.001,batch=128,epoch=1,change=0, repe=50

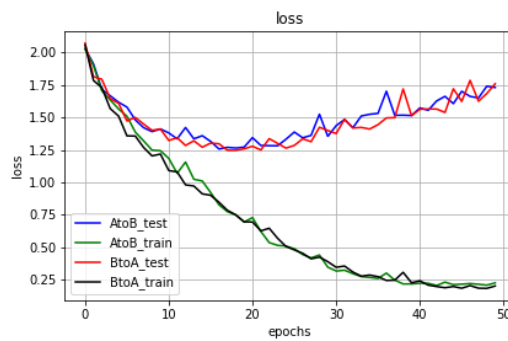


図 2: random=0,lr=0.001,batch=128,epoch=1,change=0, repe=50

次に、一回の交換で間違ったもの全てを交換すると、図 3 のようになった。図 3 のテスト精度を見ると、epoch=0 では、交換を行わないのでほぼ一致しているが、epoch \geq 1 では、AtoB_test と BtoA_test で 25 % から 35 % ほど離れており、予想通り AtoB_test が低く、BtoA_test が高いという結果になった。

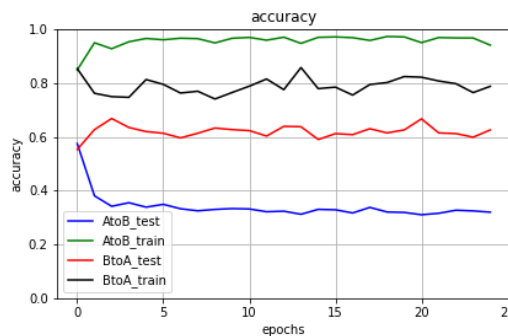


図 3: random=0,lr=0.001,batch=128,epoch=25,change=10000, repe=25

モデルの層は下記である。
実験に使った CNN のモデルを 1 に示す。

Listing 1: model

```
1
2 model=Sequential()
3
4 #1st layer
5 model.add(Conv2D(32, kernel_size=(3,3), activation='relu', padding='same', input_shape=(32,32,3)))
6 model.add(Conv2D(32, kernel_size=(3,3), activation='relu'))
7 model.add(MaxPooling2D(pool_size=(2,2)))
8 model.add(Dropout(0.25))
9
10 #2nd layer
11 model.add(Conv2D(32, kernel_size=(3,3), activation='relu', padding='same'))
12 model.add(Conv2D(32, kernel_size=(3,3), activation='relu'))
13 model.add(MaxPooling2D(pool_size=(2,2)))
14 model.add(Dropout(0.25))
15
16 #Output layer
17 model.add(Flatten())
18 model.add(Dense(512))
19 model.add(Activation('relu'))
20 model.add(Dropout(0.5))
21 model.add(Dense(10))
22 model.add(Activation('softmax'))
```

2 考察

確かに、A と B で有意差は見られたが、試行回数を重ねても精度の向上は見られなかった。これはモデルのテスト精度が低く、交換する個数が多すぎるためであると思われる。そのためモデルのテスト精度を高めたり、難しいデータかの判定を間違えた回数によって変えていくとより、A と B でうまく差異をつけられそうである。

3 次回行うこと

- モデルを改良して、テスト精度を上げる
- 訓練と交換をしていく間で間違えた回数をカウントして、間違えた回数の多いもの、少ないものに 2 分割して accuracy の差を見る
- 初期の A,B を振り分ける random.state を変えながら訓練していき、同様に間違えた回数をカウントして、間違えた回数の多いもの、少ないものに 2 分割して accuracy の差を見る

現時点でのコードを以下に示す。

https://github.com/KeitaTakami/WeeklyReport/blob/master/0515/cifar10_sameclass.ipynb