

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。先週のモデルでは、モデルの構築の時点で間違っており、`model_AtoB=model_BtoA` となってしまうため、全データが訓練データと同じになってしまい、高い識別率を示していたので、モデルを改めた。使っていないデータ（40000 件）で accuracy と loss を訓練後のモデルで計測すると AtoB, BtoA でどちらも同じ値をはじき出したことにより判明。

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 が低く、BtoA が高くなると予想される。

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: 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

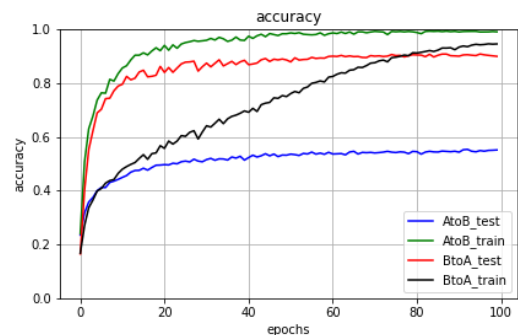


図 1: random=0, lr=0.0001, batch=32, epoch=1, change=10000, repe=100

重みをリセットしていないモデルを図 1, 2 に示した。図 1 では、test の結果が、AtoB では、57 % ほどだったのに対し、BtoA が 90 % 近くを示した。ただ、BtoA の訓練データの精度ばかりが上がり、過剰適合しているといえる。

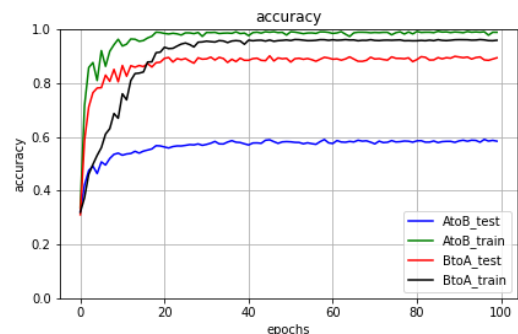


図 2: random=0, lr=0.001, batch=16, epoch=1, change=10000, repe=100

杉山君とパラメータを揃えたモデルを図 2 に示した。また、未知の 40000 件のデータをテストデータとした際には、AtoB モデルでは 51.1 %、BtoA モデルでは 56 % の accuracy を示した。

重みをリセットしたモデルを図 3 に示した。特に、AtoB と BtoA で差は見られなかった。交換する数が 1 になってました…

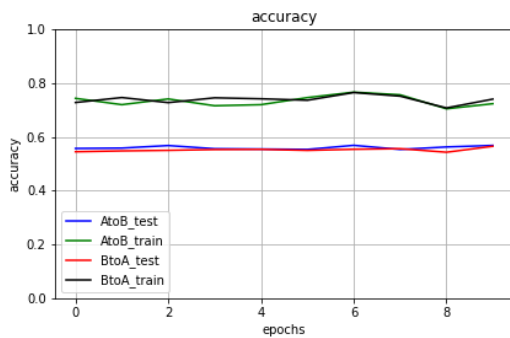


図 3: random=0,lr=0.0001,batch=32,epoch=50,change=1, repe=10

モデルのコンパイルは下記のように行い、
`model_AtoB.compile(loss='categorical_crossentropy',
optimizer=Adam(lr=0.001), metrics=['accuracy'])`

モデルの訓練は下記である。
`model_AtoB.fit(x_a,y_a, batch_size=32, epochs=1,
verbose=1, validation_split=0.1)`
また、モデルの構築は下記である。

```
model=Sequential()
1 層目
model.add(Conv2D(32,kernel_size=(3,3),activation='relu',padding='same',input_shape=(32,32,3)))
model.add(Conv2D(32,kernel_size=(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))
2 層目
model.add(Conv2D(32,kernel_size=(3,3),activation='relu',padding='same'))
model.add(Conv2D(32,kernel_size=(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))
出力層
model.add(Flatten())      model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))    model.add(Dense(10))
model.add(Activation('softmax'))
model_AtoB=model model_BtoA=model
```

2 考察

重みをリセットしないままで行うと、精度に大きな開きが見られたがこれでは目的の「スプリットの

仕方によって Accuracy の値が変わるか」どうかはわからない。そのため、交換するたびに十分といえるほど訓練していくような実装を行わなければならないと思った。

3 次回行うこと

- 一回の訓練のエポック数を伸ばし、交換数を最大にして再実験する。
- 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/0508/cifar10.ipynb>