

進捗報告

表 1: 実験の設定

dataset	cifar10
data N	2,000 / model
task	5, 10 クラス識別
input	image(3x32x32)
output	class(5 or 10)
model	CNN(16 層)
optim	SDG (lr=0.001, moment=0.9)
loss	Cross Entropy Loss
batch size	16
epoch	100

1 今週やったこと

- 5 クラス識別を組み合わせたモデルの構築

2 モデルの構築

5 クラス識別を組み合わせたモデルの構築.

cifar10 に含まれる 10 クラスを 5 クラスずつに分割した. 今回はインデックスの前半 (airplane, mobile, bird, cat, deer) と後半 (dog, frog, horse, ship, truck) で分けることにした. 生成した部分データセットを、それぞれモデル A, B で学習した. 5 クラス分類ができるモデル A, B を持つ結合モデル A + B を作成し、10 クラス分類の精度を計測した. このモデルの学習と相互関係を図 1 に示した.

結合モデル A + B は 10 クラスのデータセットを A, B に入力し、得られた出力をクラスインデックス順に結合して、出力とする. 結合では特別な処理を行わず、そのままのデータを連結した.

2.1 結果

実験の結果は図 2 にテスト精度として示した. 5 クラス識別ではモデル A, B ともにベースラインから始まり、A は 80%, B は 70%程度という精度になった. 1 回の実験かつデータ数が 2000 であったため、同様の条件でも差が出たことが考えられる. モデル B を再度学

習してみると、80%程度になることを後で確認した. 10 クラス識別では、ベースライン 10%に対して、50%という結果となった. さらに精度を高めるには、結合方法の改良が必要と考えられる.

3 考察

クラスの分割パターンを変えて、実験
モデルの結合方法の調査

4 今後の予定

- モデルの改良

5 ソースコード

結合モデルの Pytorch での実装をコード 1 に示す.

Listing 1: cnn

```

1 class catCNN(nn.Module):
2     def __init__(self, x, y):
3         super(catCNN, self).__init__()
4         self.model = [x, y]
5
6     def forward(self, i):
7         x = self.model[0](i)
8         y = self.model[1](i)
9
10        return torch.cat((x, y), 1)

```

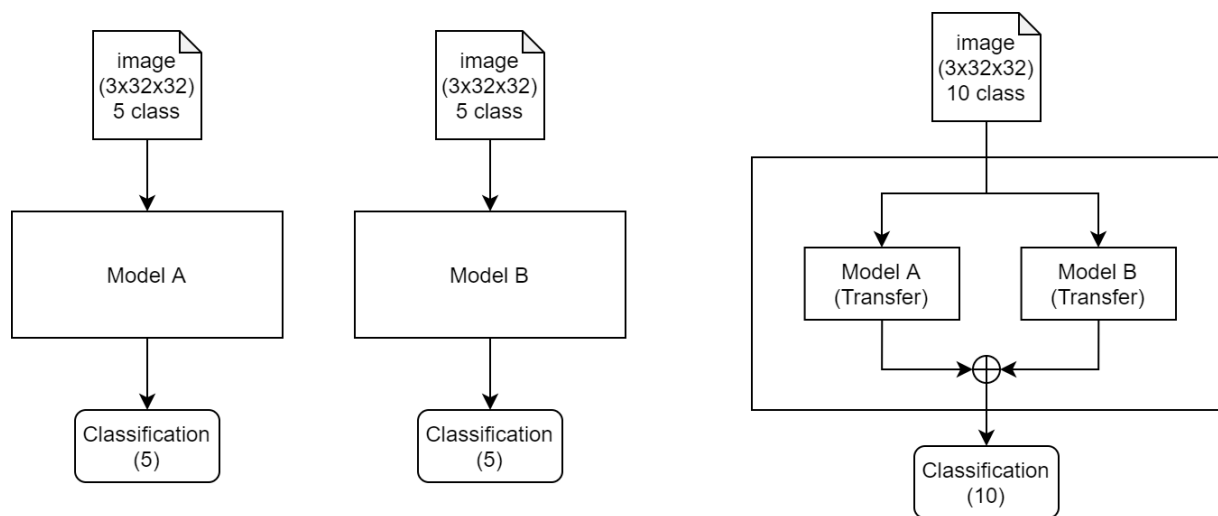


図 1: モデルの簡略図 () 内はデータの次元数

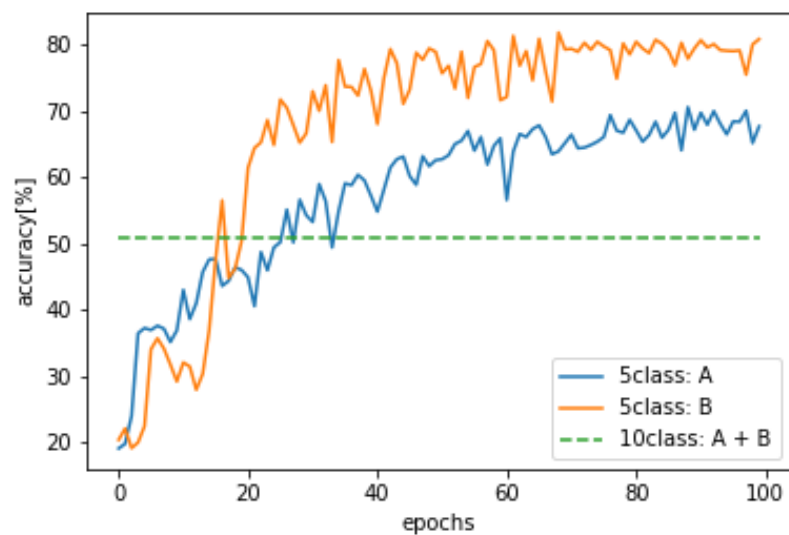


図 2: テストの精度の比較