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が、削除されたデータではどのように識別率が変わっていくのかを実験した。前回の結果より、二回の削除でおよそ半分にデータが減ることがわかったので、その削除された半分のデータで識別率及びクラス数がどうなるかを実験する。

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、失敗したデータをランダムに M 個ずつ削除（自分で指定）
7. 3.-6. を二回繰り返す
8. 二回の繰り返しで得られた難しいデータ (削除された側のデータ) に対して 3.-6. を繰り返す

上記のアルゴリズムより、前回の結果と比較して、データ数はもっと少な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）
- delete_num: 1 回の削除で削除するデータの最大数
- iteration: 3.4.5.6. の繰り返し数

パラメータは random=0,lr=0.001,batch=128,epoch=25,delete_num=10000,iteration=40 とした。

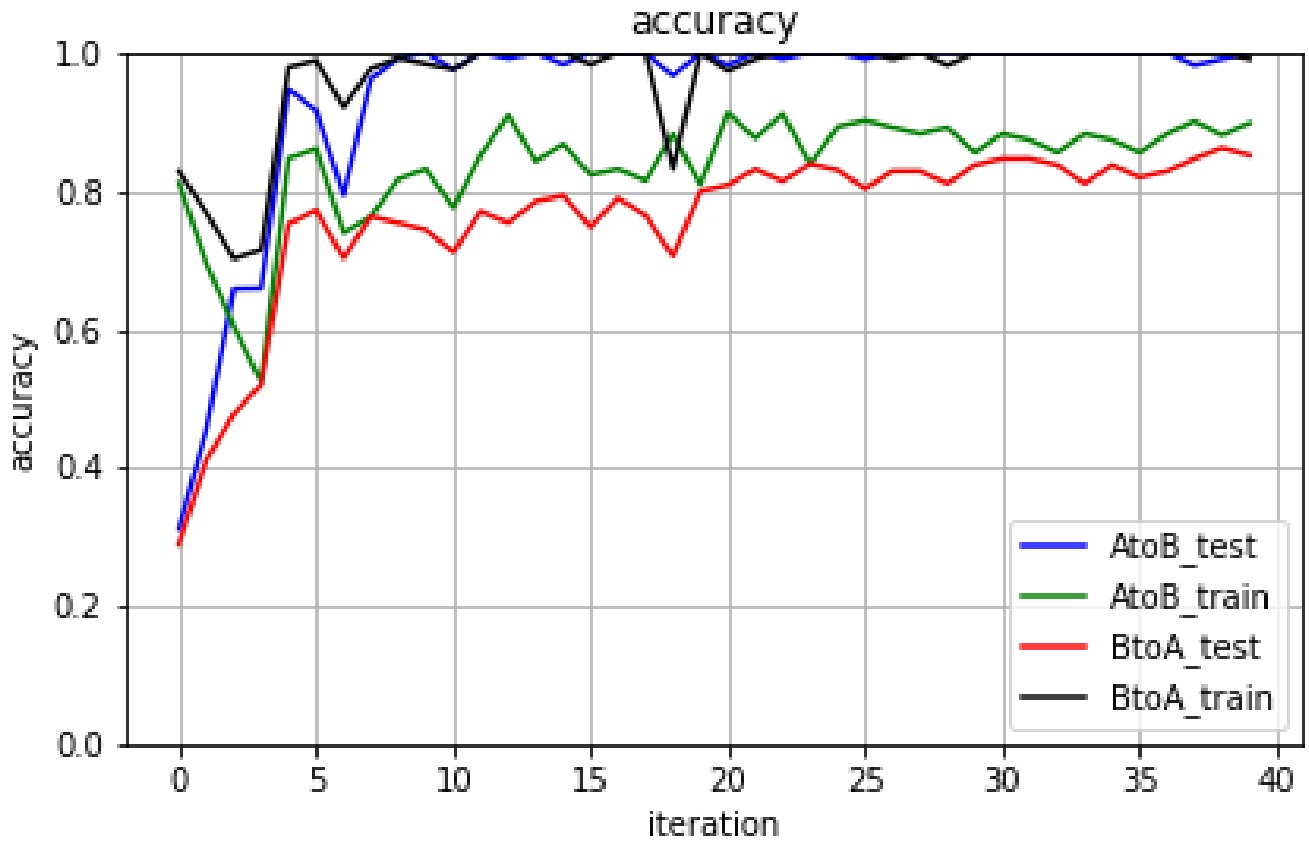

図 1: テスト精度と訓練精度。横軸が繰り返す回数、縦軸が Accuracy.

また、各クラスの数は図 2、図 3 のようになった。クラスがなくなったものは排除された。B のほうからは、bird と automobile のみ残った。同じ数だけ削除という手順をとっているので、airplane と cat は A にのみ残っているが、AtoB の Accuracy が高すぎて、B から削除するものがなく、結果として A の airplane と cat は判別できてないはずであるが残り続けている。

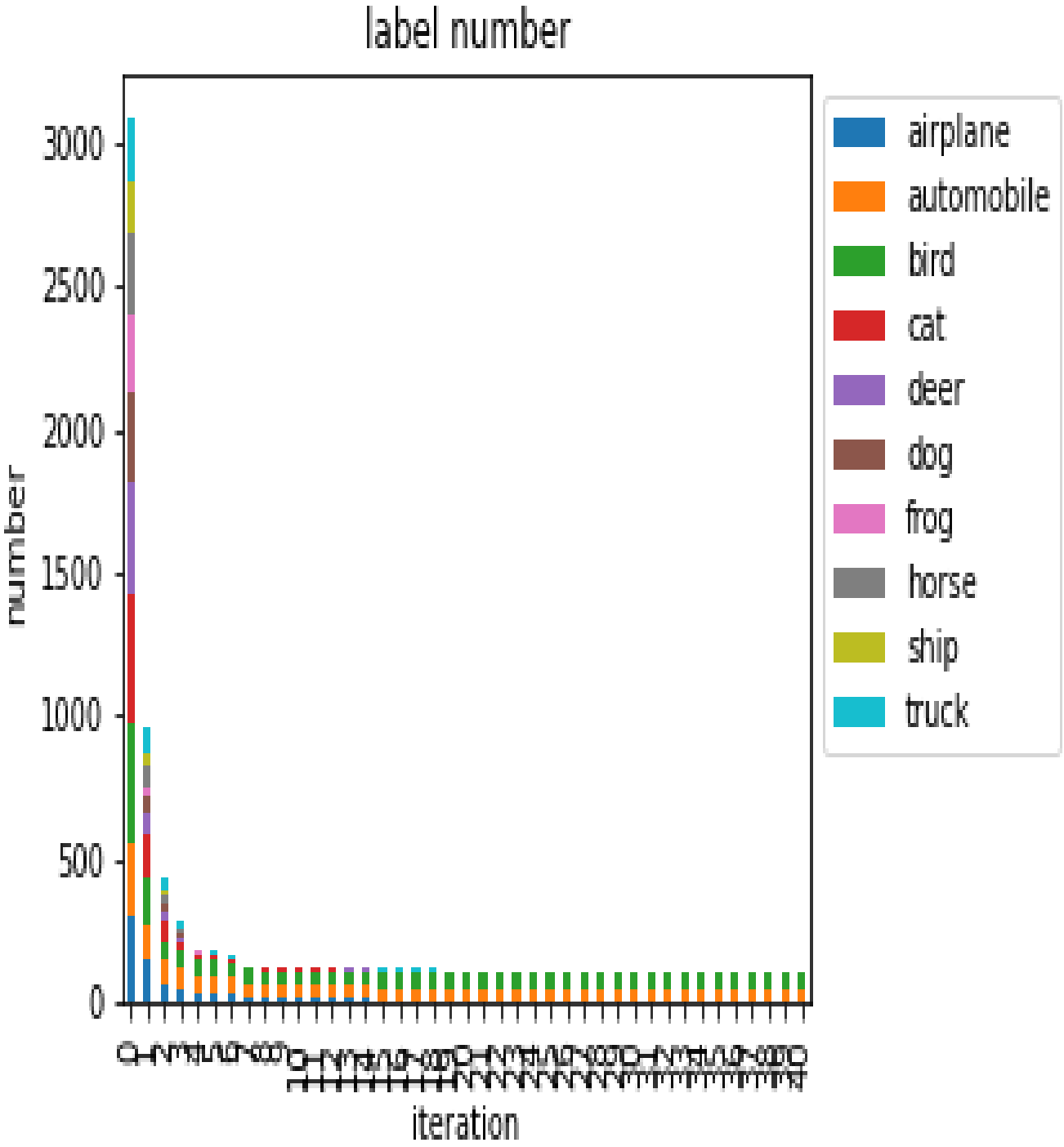

図 2: A のラベルの数の推移。横軸が繰り返す回数、縦軸が A のラベルの数。

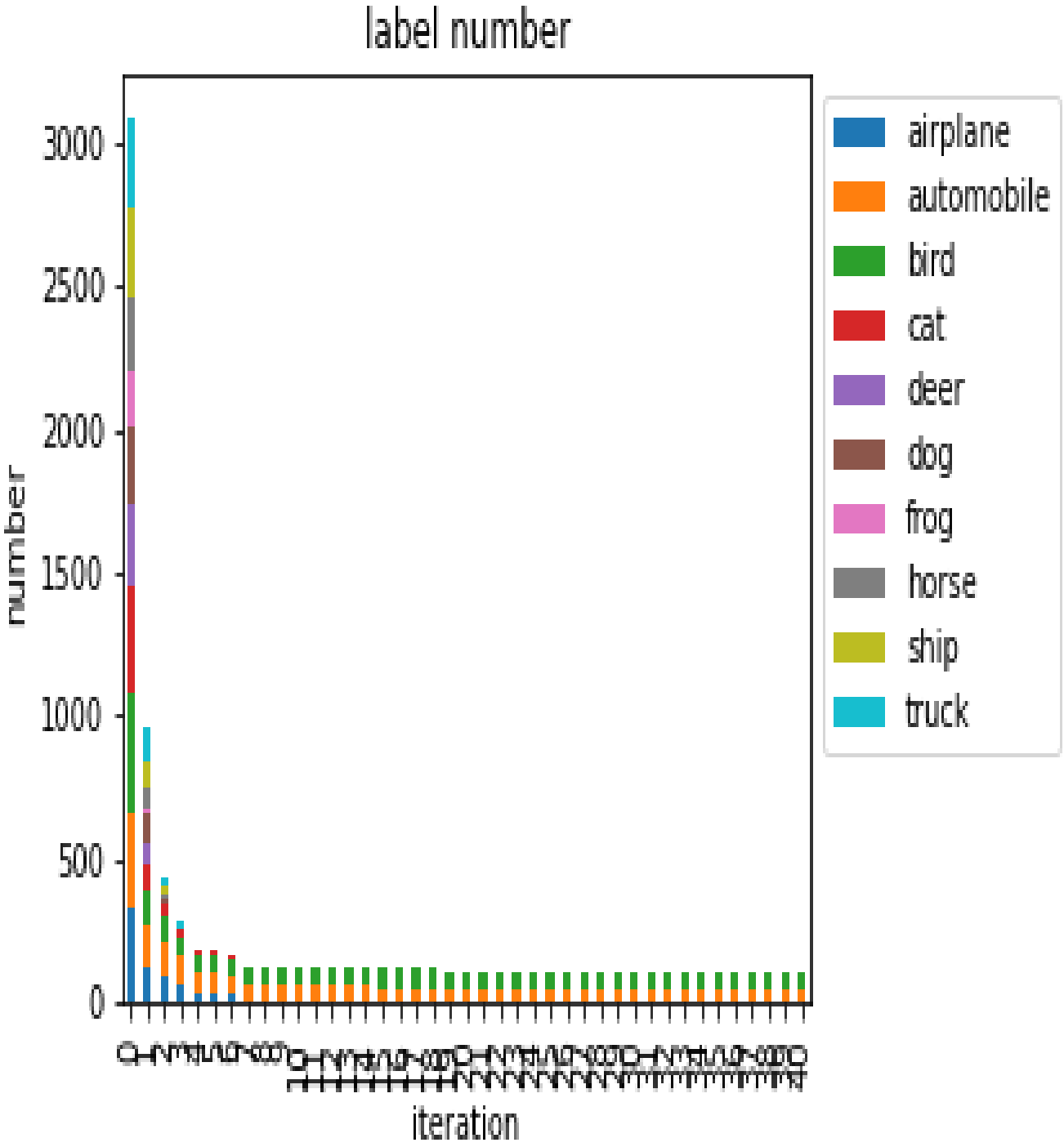

図 3: B のラベルの数の推移。横軸が繰り返す回数、縦軸が B のラベルの数。

比較として、前回のデータ（10000 件のデータから徐々に減らしていくもの）を載せる。前回はデータ数が 5000 → 1000 と 1/5 程度であったが、今回は 3000 → 100 と 1/30 とデータ数が大まかに減って、クラス数も A で 4 個、B で 2 個しか残らなかった。

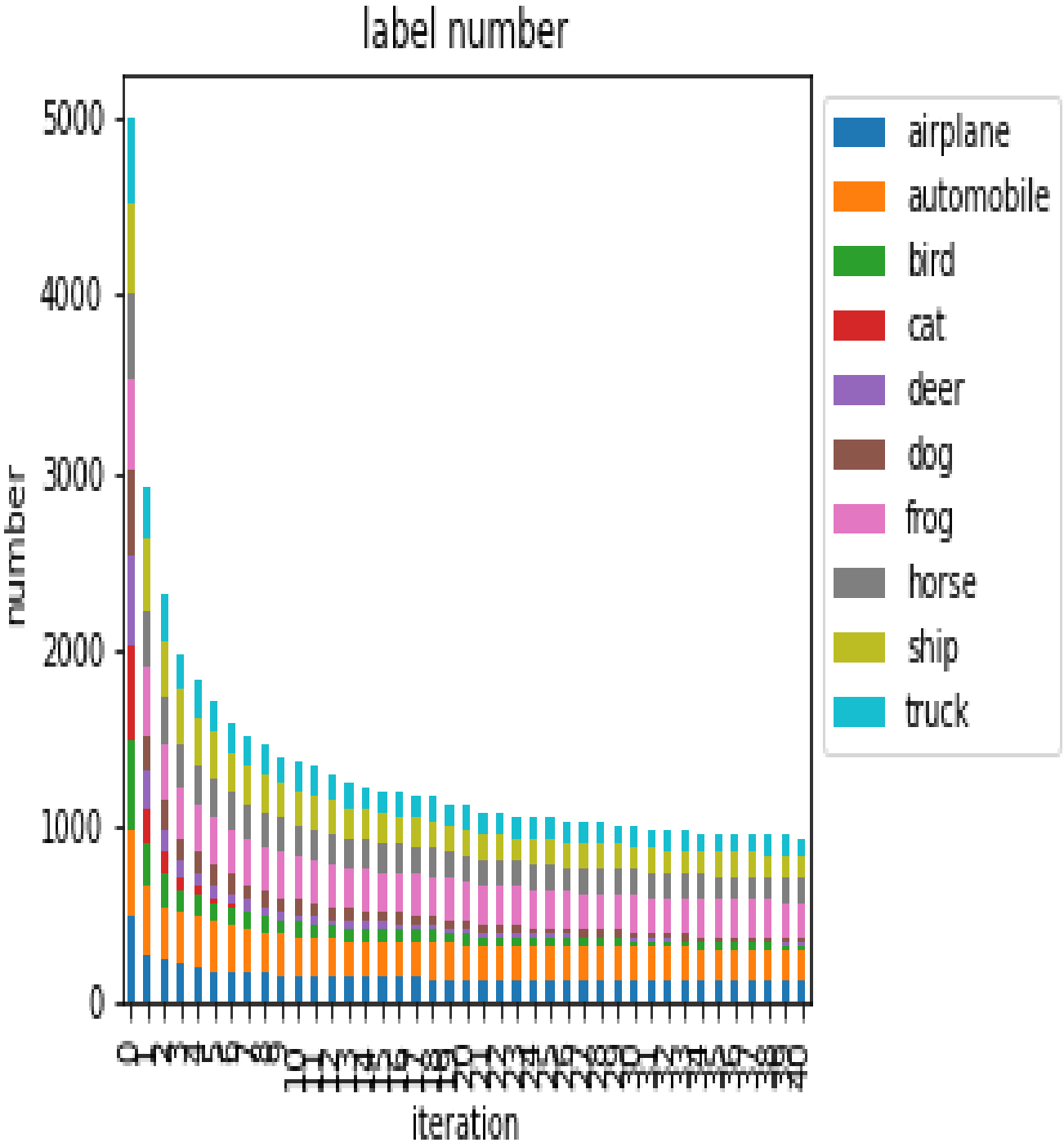

図 4: 前回の結果。A のラベルの数の推移。横軸が繰り返す回数、縦軸が A のラベルの数。

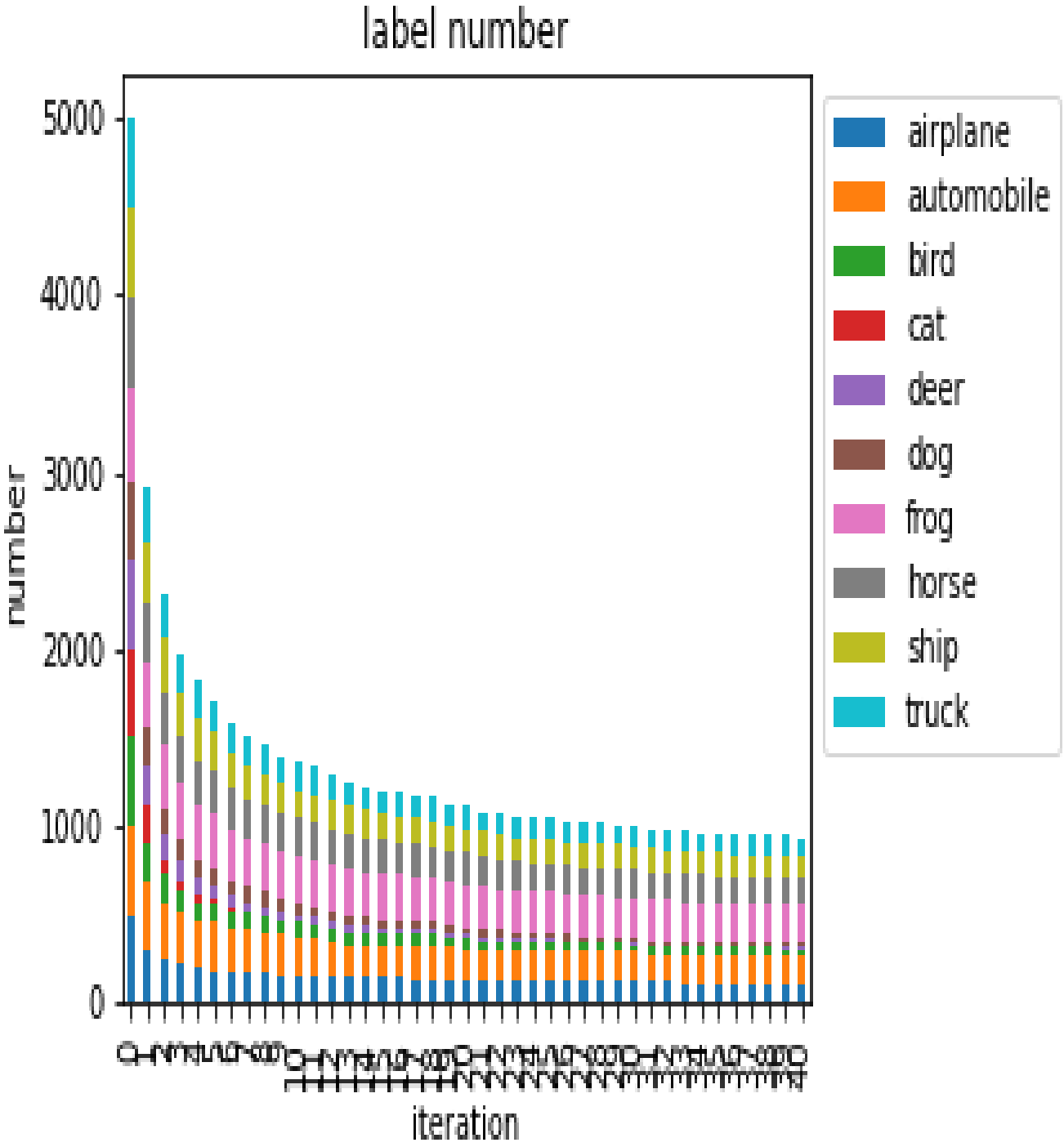

図 5: 前回の結果。B のラベルの数の推移。横軸が繰り返す回数、縦軸が B のラベルの数。

初期データの偏りと間違えたデータであることから、最終的には 108 個のデータしか残っていない。また、A、B のラベルの CSV データをそれぞれ図 6, 図 7 に示す。

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
0	311	246	413	450	395	314	263	295	174	220
1	149	129	164	150	78	51	35	68	49	89
2	63	88	71	62	35	25	4	30	12	47
3	47	82	62	32	10	10	2	19	4	20
4	44	55	54	17	5	3	1	2	1	8
5	42	54	54	16	3	1	1	2	1	6
6	38	54	54	12	2	1	0	1	1	2
7	21	45	54	8	1	1	0	0	0	1
8	18	44	54	7	1	1	0	0	0	1
9	17	44	54	7	1	1	0	0	0	1
10	17	44	54	7	1	1	0	0	0	1
11	15	44	54	6	1	1	0	0	0	1
12	15	44	54	6	1	1	0	0	0	1
13	15	44	54	5	1	1	0	0	0	1
14	15	44	54	5	1	1	0	0	0	1
15	14	44	54	5	0	1	0	0	0	1
16	14	44	54	5	0	1	0	0	0	1
17	14	44	54	5	0	1	0	0	0	1
18	14	44	54	5	0	1	0	0	0	1
19	13	42	54	4	0	1	0	0	0	1
20	13	42	54	4	0	1	0	0	0	1
21	11	42	54	4	0	1	0	0	0	1
22	11	42	54	4	0	1	0	0	0	1
23	11	42	54	4	0	0	0	0	0	1
24	11	42	54	4	0	0	0	0	0	1
25	11	42	54	4	0	0	0	0	0	1
26	11	42	53	4	0	0	0	0	0	1
27	11	42	53	4	0	0	0	0	0	1
28	11	42	53	4	0	0	0	0	0	1
29	11	42	53	4	0	0	0	0	0	1
30	11	42	53	4	0	0	0	0	0	1
31	11	42	53	4	0	0	0	0	0	1
32	11	42	53	4	0	0	0	0	0	1
33	11	42	53	4	0	0	0	0	0	1
34	11	42	53	4	0	0	0	0	0	1
35	11	42	53	4	0	0	0	0	0	1
36	11	42	53	4	0	0	0	0	0	1
37	11	42	53	4	0	0	0	0	0	1
38	9	42	53	4	0	0	0	0	0	1
39	9	42	53	4	0	0	0	0	0	0
40	9	42	53	4	0	0	0	0	0	0

図 6: A のラベルの数の CSV データ。

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
0	334	327	423	371	281	267	197	251	315	315
1	129	150	116	89	74	102	25	65	87	125
2	89	129	85	41	10	9	2	21	17	34
3	67	105	65	22	1	1	0	5	0	22
4	32	81	63	14	0	0	0	0	0	0
5	31	75	63	11	0	0	0	0	0	0
6	31	62	63	9	0	0	0	0	0	0
7	6	62	63	0	0	0	0	0	0	0
8	1	62	63	0	0	0	0	0	0	0
9	1	61	63	0	0	0	0	0	0	0
10	1	61	63	0	0	0	0	0	0	0
11	0	60	62	0	0	0	0	0	0	0
12	0	60	62	0	0	0	0	0	0	0
13	0	59	62	0	0	0	0	0	0	0
14	0	59	62	0	0	0	0	0	0	0
15	0	57	62	0	0	0	0	0	0	0
16	0	57	62	0	0	0	0	0	0	0
17	0	57	62	0	0	0	0	0	0	0
18	0	57	62	0	0	0	0	0	0	0
19	0	53	62	0	0	0	0	0	0	0
20	0	53	62	0	0	0	0	0	0	0
21	0	51	62	0	0	0	0	0	0	0
22	0	51	62	0	0	0	0	0	0	0
23	0	50	62	0	0	0	0	0	0	0
24	0	50	62	0	0	0	0	0	0	0
25	0	50	62	0	0	0	0	0	0	0
26	0	49	62	0	0	0	0	0	0	0
27	0	49	62	0	0	0	0	0	0	0
28	0	49	62	0	0	0	0	0	0	0
29	0	49	62	0	0	0	0	0	0	0
30	0	49	62	0	0	0	0	0	0	0
31	0	49	62	0	0	0	0	0	0	0
32	0	49	62	0	0	0	0	0	0	0
33	0	49	62	0	0	0	0	0	0	0
34	0	49	62	0	0	0	0	0	0	0
35	0	49	62	0	0	0	0	0	0	0
36	0	49	62	0	0	0	0	0	0	0
37	0	49	62	0	0	0	0	0	0	0
38	0	47	62	0	0	0	0	0	0	0
39	0	46	62	0	0	0	0	0	0	0
40	0	46	62	0	0	0	0	0	0	0

図 7: B のラベルの数の CSV データ。

2　考察

初期データの偏りからか前回は bird が難しいクラスである結果であったが、今回は残り続けた。これは bird の初期データ数が多く、他のクラスが先になくなってしまったため、二値分類に近い形となってタスクが簡単になってしまったためだと思われる。

3　次回行うこと

- 難しいデータ、簡単なデータとして対照実験するため、データとして二分する際にそれぞれのクラス数をすべてそろえてから同様に実験する。

現時点でのコードを以下に示す。
https://github.com/KeitaTakami/WeeklyReport/blob/master/0529/cifar10_missdata.ipynb

実験に使った CNN のモデルを下記に示す.

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'))
```