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交換というアルゴリズムから、間違えたデータを削除するという手順をとった。

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. に戻る

上記のアルゴリズムを繰り返すと、難しいデータが除かれるのでデータの精度が高くなっていくと考えられる。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

アルゴリズムに沿って、削除を行っていくと、精度は図 1 のようになった。15 回ほどからテスト精度は 98 %ほどから 100 %付近をうろうろしている。パラメータは random=0,lr=0.001,batch=128,epoch=25,delete_num=10000,iteration=40 とした。

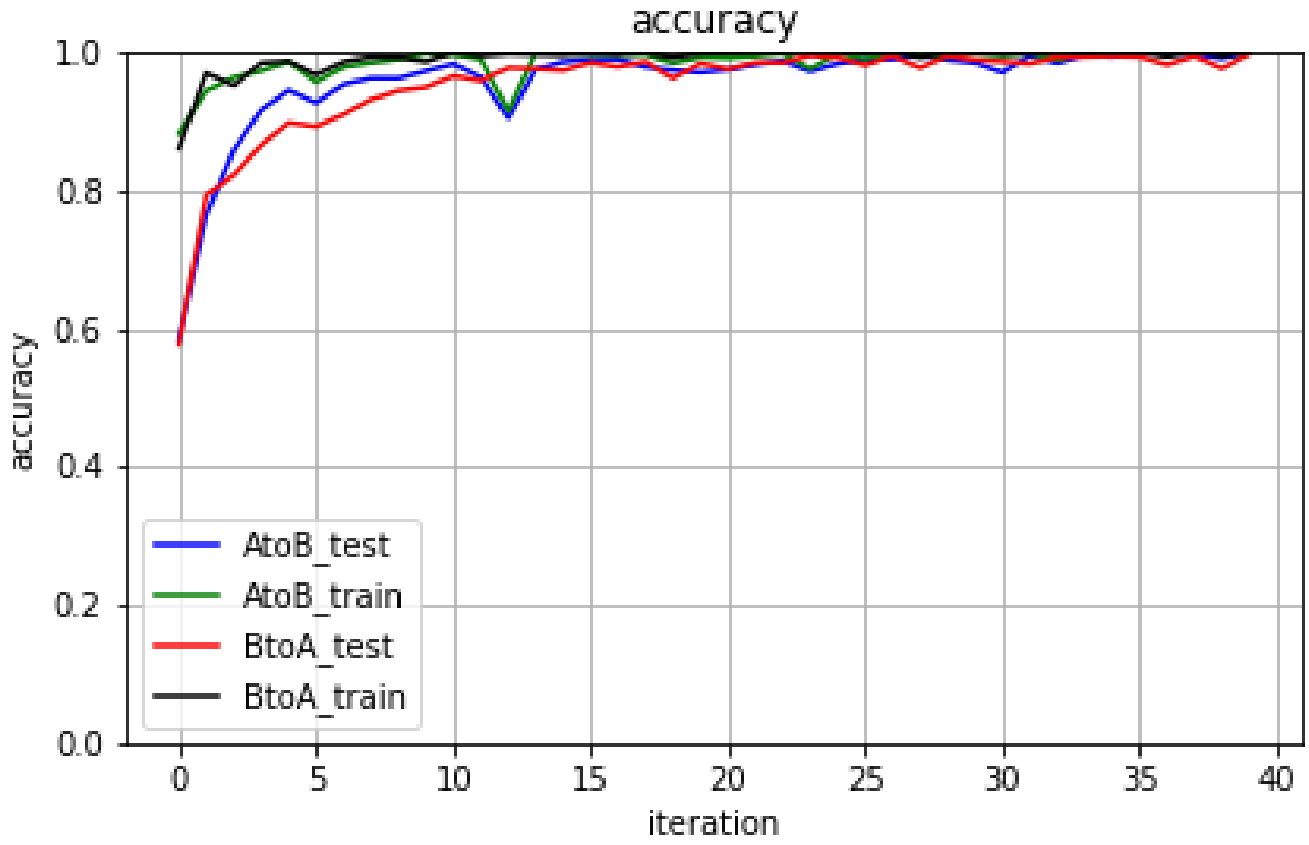

図 1: テスト精度と訓練精度。横軸が繰り返す回数、縦軸が Accuracy.

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。今回は cat だけであった。また、deer,dog,bird も残っているデータの数が少ないことから比較的判別するのが難しいデータといえる。

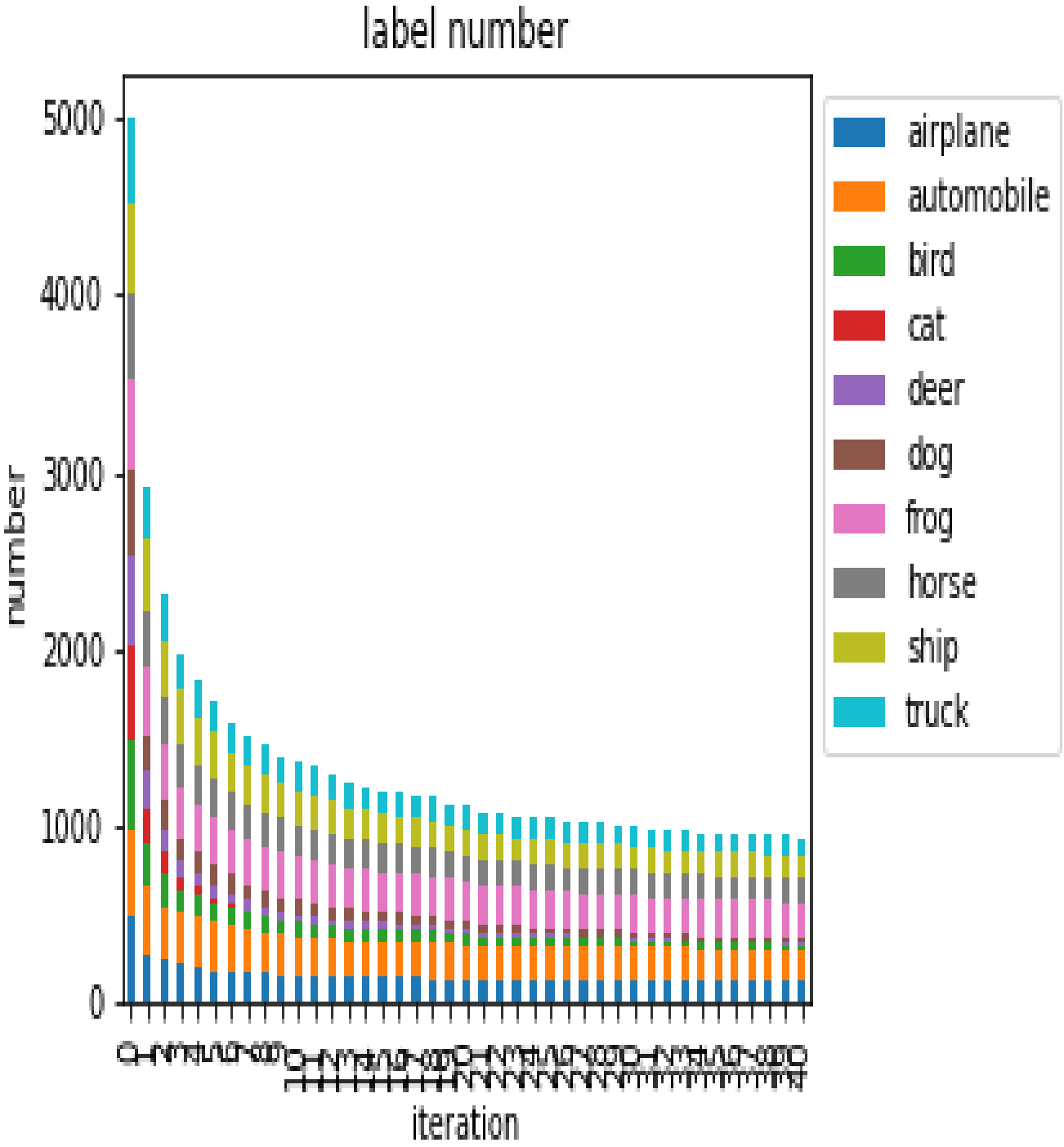

図 2: A のラベルの数の推移。横軸が繰り返す回数、縦軸が A のラベルの数。

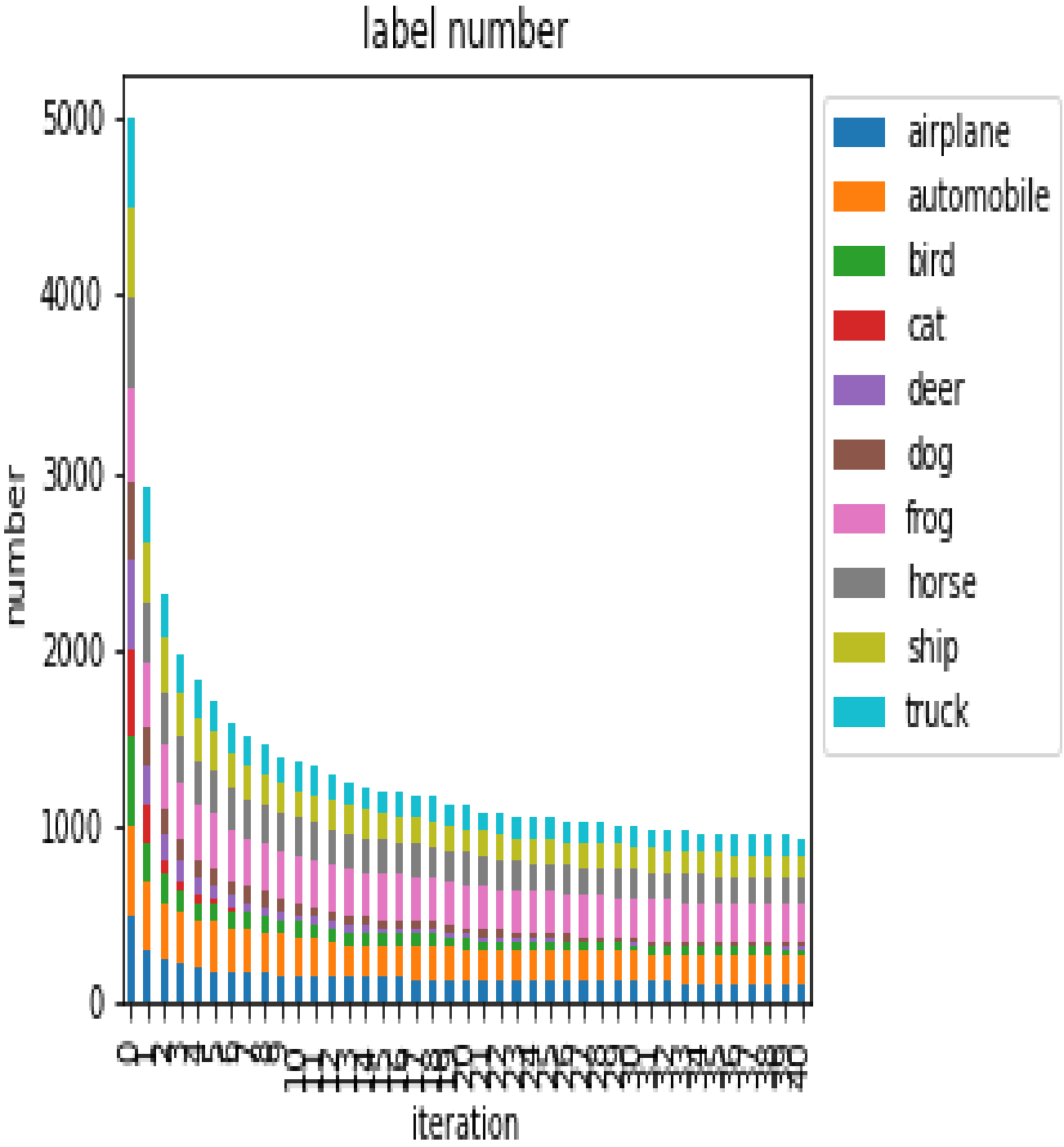

図 3: B のラベルの数の推移。横軸が繰り返す回数、縦軸が B のラベルの数。

また、A、B のラベルの CSV データをそれぞれ図 4, 図 5 に示す。

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
0	499	484	516	523	508	493	503	490	497	487
1	288	382	243	186	228	190	386	316	399	302
2	247	304	179	120	137	160	310	266	332	258
3	227	296	123	77	96	122	290	239	301	214
4	207	285	113	50	83	109	276	230	265	199
5	187	273	109	32	74	107	272	214	259	189
6	186	260	99	16	65	105	255	205	222	176
7	182	243	90	11	59	89	251	199	221	169
8	173	233	87	4	53	89	250	196	208	163
9	167	226	84	1	46	80	248	191	197	160
10	156	224	82	1	43	78	239	184	197	157
11	151	219	80	1	42	78	237	183	193	153
12	151	214	72	0	40	76	229	178	180	148
13	151	211	71	0	37	65	227	176	173	147
14	150	211	71	0	36	60	227	165	171	136
15	148	210	68	0	36	58	227	158	168	136
16	145	208	68	0	34	57	227	157	165	135
17	145	206	67	0	31	54	227	157	163	133
18	142	203	66	0	31	53	227	157	156	131
19	142	198	60	0	31	51	225	148	155	125
20	142	195	54	0	30	48	224	146	153	124
21	140	195	46	0	26	39	223	144	151	124
22	140	191	46	0	23	38	223	143	151	116
23	140	189	46	0	23	37	222	142	142	116
24	137	189	46	0	23	35	221	142	142	116
25	137	187	46	0	19	34	220	142	142	115
26	137	187	46	0	18	32	212	142	141	115
27	137	186	46	0	15	32	212	141	140	115
28	137	184	45	0	14	32	212	141	138	115
29	137	184	45	0	14	31	211	139	138	113
30	137	182	43	0	11	31	208	139	137	112
31	136	181	43	0	8	28	207	139	132	112
32	135	181	41	0	8	28	207	139	129	112
33	135	181	38	0	8	27	206	137	128	112
34	132	181	38	0	7	26	206	137	128	112
35	131	178	38	0	7	25	206	136	128	112
36	131	178	38	0	7	25	206	136	128	110
37	131	178	38	0	7	25	205	135	128	110
38	131	177	37	0	6	25	205	134	128	110
39	131	173	33	0	5	25	205	134	128	110
40	131	173	33	0	5	24	205	134	128	110

図 4: A のラベルの数の CSV データ。

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
0	506	490	516	493	491	444	527	511	528	494
1	313	370	234	201	226	219	369	325	337	326
2	242	338	149	82	153	145	348	298	309	249
3	226	302	116	47	116	132	312	266	241	227
4	195	281	102	29	99	112	306	255	232	206
5	188	274	99	24	83	104	300	242	215	187
6	182	243	96	13	71	95	282	229	203	175
7	175	238	95	7	60	90	276	215	196	162
8	172	238	89	5	49	83	274	213	178	155
9	164	234	79	2	44	76	267	205	175	154
10	159	227	77	0	42	71	262	205	166	152
11	156	221	76	0	39	64	262	205	162	152
12	151	199	75	0	37	59	259	201	162	145
13	147	189	74	0	37	57	259	199	157	139
14	145	188	71	0	32	54	257	192	151	137
15	145	186	70	0	31	50	255	188	149	135
16	145	185	70	0	31	48	254	186	145	132
17	144	184	67	0	30	46	251	186	143	132
18	143	182	66	0	30	45	249	177	143	131
19	139	182	54	0	23	45	248	176	139	129
20	133	180	53	0	22	41	245	175	138	129
21	132	174	53	0	22	40	241	170	138	118
22	131	173	52	0	20	36	236	169	137	117
23	131	173	49	0	14	36	234	166	137	117
24	131	173	49	0	14	33	231	166	137	117
25	131	172	49	0	14	33	230	160	136	117
26	130	170	49	0	11	31	226	160	136	117
27	128	170	49	0	11	28	226	159	136	117
28	128	166	49	0	10	27	226	159	136	117
29	127	166	46	0	9	27	226	159	136	116
30	125	165	45	0	8	25	226	159	133	114
31	122	164	39	0	8	25	224	159	131	114
32	122	163	39	0	8	25	224	155	130	114
33	119	163	39	0	8	23	224	153	130	113
34	119	163	38	0	7	23	224	153	127	113
35	119	162	36	0	7	23	223	151	127	113
36	119	162	36	0	7	23	222	151	126	113
37	119	162	36	0	6	23	222	151	126	112
38	119	162	36	0	6	23	222	150	126	109
39	119	158	36	0	6	22	221	150	124	108
40	119	158	36	0	6	22	221	150	124	107

図 5: B のラベルの数の CSV データ。

2　考察

予想通り、だんだん精度は高くなっていった。動物のほうが識別が難しいというのは予想していたが、ほとんどのクラスが最終まで残ったのは意外であった。iteration 数を増やしたら、データは残り続けるのかを調べてみたいと思った。また、初期データ数を増減させると最終的に残るクラスの数はどうなるのだろうか。

3　次回行うこと

- 初期のデータ数を変えて実験を行ってみる。
- iteration 数をさらに増やして、データがなくなるか識別率が 100 %になるかを実験する。

現時点でのコードを以下に示す。
https://github.com/KeitaTakami/WeeklyReport/blob/master/0522/cifar10_delete.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'))
```