

# 進捗報告

M1 高山 裕成

2021 年 6 月 30 日

## 1 進捗

- なし

## 2 transformers

新しいバージョンの transformers では, 仕様が結構変わっており苦戦. 覚え直し中... ついでに [lg-hub/bert\\_practice](#) も作り直した方がいいかもしれない.

## 3 今やっていること

- 前期発表分で, 5 分割交差検証する実装
- 今後の研究方針に関する資料作成とその実装

---

```

1 def train(self):
2     for epoch in range(self.n_epoch):
3         time_start = time.time()
4
5         for phase in ['train', 'val']:
6
7             if phase == 'train':
8                 self.net.train()
9             else:
10                self.net.eval()
11
12            self.reset_count()
13
14            for x, y in self.study_loaders[phase]:
15                x = x.to(device)
16                y = y.to(device)
17                self.optimizer.zero_grad()
18                with torch.set_grad_enabled(phase == 'train'):
19                    y_pred = self.net(x)
20                    _, predicted = torch.max(y_pred.data, 1)
21                    self.total += y.size(0)
22                    # loss 計算・加算
23                    loss = self.criterion(y_pred, y.argmax(1))
24                    self.total_loss += loss.item()
25                    # 正解数加算
26                    self.correct += (predicted == y.argmax(1)).sum().item()
27                    # 混合行列更新
28                    for i in range(len(predicted)):
29                        self.c_mat[torch.max(y.data, 1)[1][i]][predicted[i]] += 1
30
31                    # 訓練時のみバックプロパゲーション
32                    if phase == 'train':
33                        loss.backward()
34                        self.optimizer.step()
35
36                # ロスの合計をlen(data_loader)で割る
37                mean_loss = self.total_loss / len(self.study_loaders[phase])
38                acc = (self.correct / self.total) * . . .

```

---