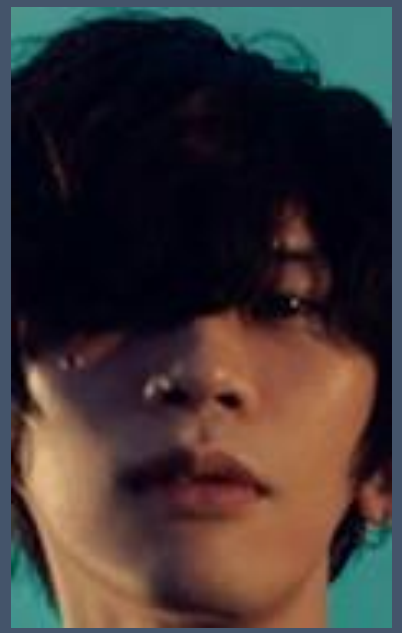
認識システム特論
最終課題
バンドマン識別モデルの作成

情報科学研究科
システム工学専攻
1年　中司　泰誠






皆さんはこの人達を
見分けることができますか？

川上洋平さん　[Alexandros]

代表曲：閃光

川谷絵音さん　ゲスの極み乙女，indigo la End
代表曲：私以外私じゃないの

米津玄師さん
代表曲：lemon

背景

見分けられないと気まずい会話になる

複数回，間違えられると訂正するのが面倒

友達「髪切った？」

中司「切った！この人の髪型ぽくしてみた〜」

友達「いいね！この人…ゲス極の川谷絵音だよね？」

中司「[Alexandros]の川上洋平やね」

友達「そうなんだ！なんか最近のバンドマンて似たような人多いよね〜」

中司「確かに多いよね．簡単に見分けられるシステムとかがあればいいんやけどね」

バンドマン識別モデルの作成を決意

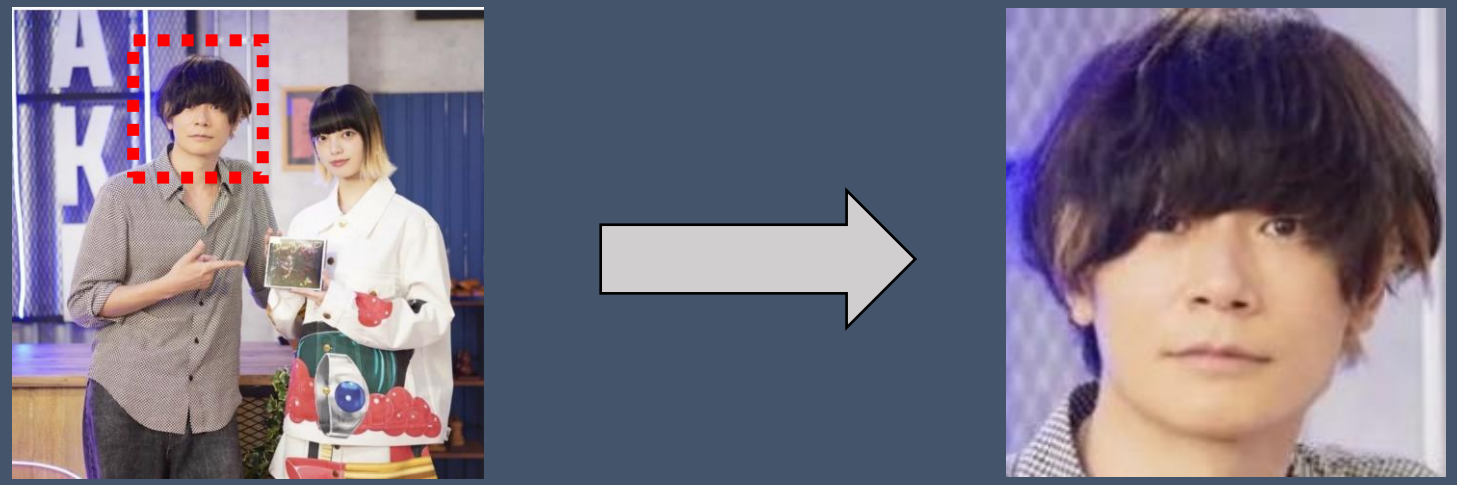
データセット作成方法

Icrawlerを用いてGoogleから指定したキーワードを含む画像を自動収集
キーワードは「川上洋平」「川谷絵音」「米津玄師」それぞれ100枚

```python
from icrawler.builtin import GoogleImageCrawler

def download_images(keyword, max_num):
    google_crawler = GoogleImageCrawler(storage={"root_dir": keyword})
    google_crawler.crawl(keyword=keyword, max_num=max_num)

search = '川上洋平'
download_images(search, 100)
```

データセット作成方法

手作業でCrop



モデル

MLP(多層パーセプトロン)

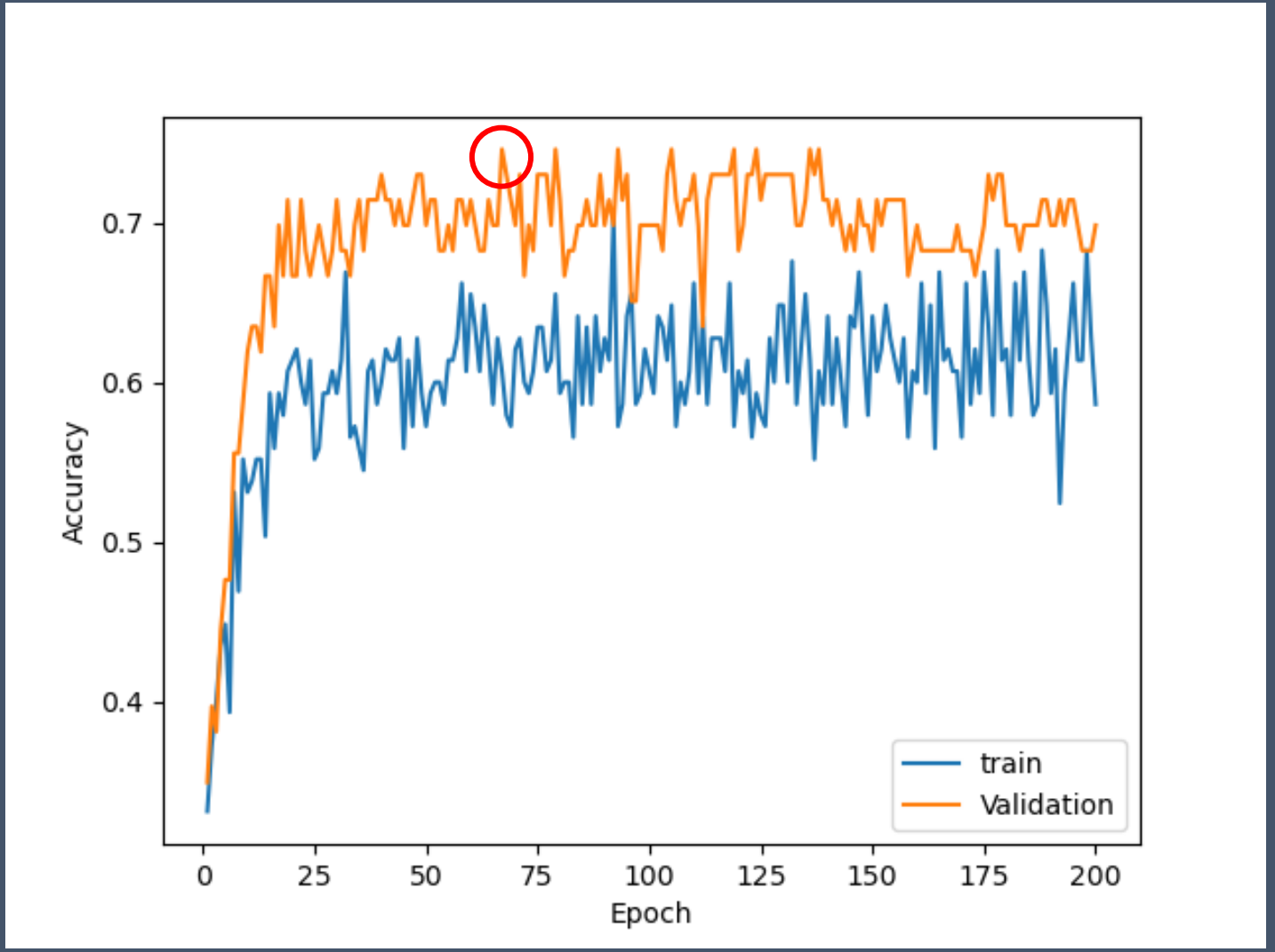
```
class MLP(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(MLP, self).__init__()
        self.flatten = nn.Flatten()
        self.linear_relu_stack = nn.Sequential(
            nn.Linear(input_size, hidden_size),
            nn.ReLU(),
            nn.Linear(hidden_size, output_size),
            nn.ReLU(),
            nn.Dropout(0.5)
        )

    def forward(self, x):
        x = self.flatten(x)
        logits = self.linear_relu_stack(x)
        return logits
```

ハイパーパラメータ

Test：Validation=7:3(145:63)
Input：256×256×3
Lr=0.00001
Epoch：200
Batch size：8
Normalize：[0.485, 0.456, 0.406],[0.229, 0.224, 0.225]
Loss：CrossEntropyLoss
Optimizer: Adam
Activation function：ReLU
Dropout：0.5

結果

Validation Acc：0.746031

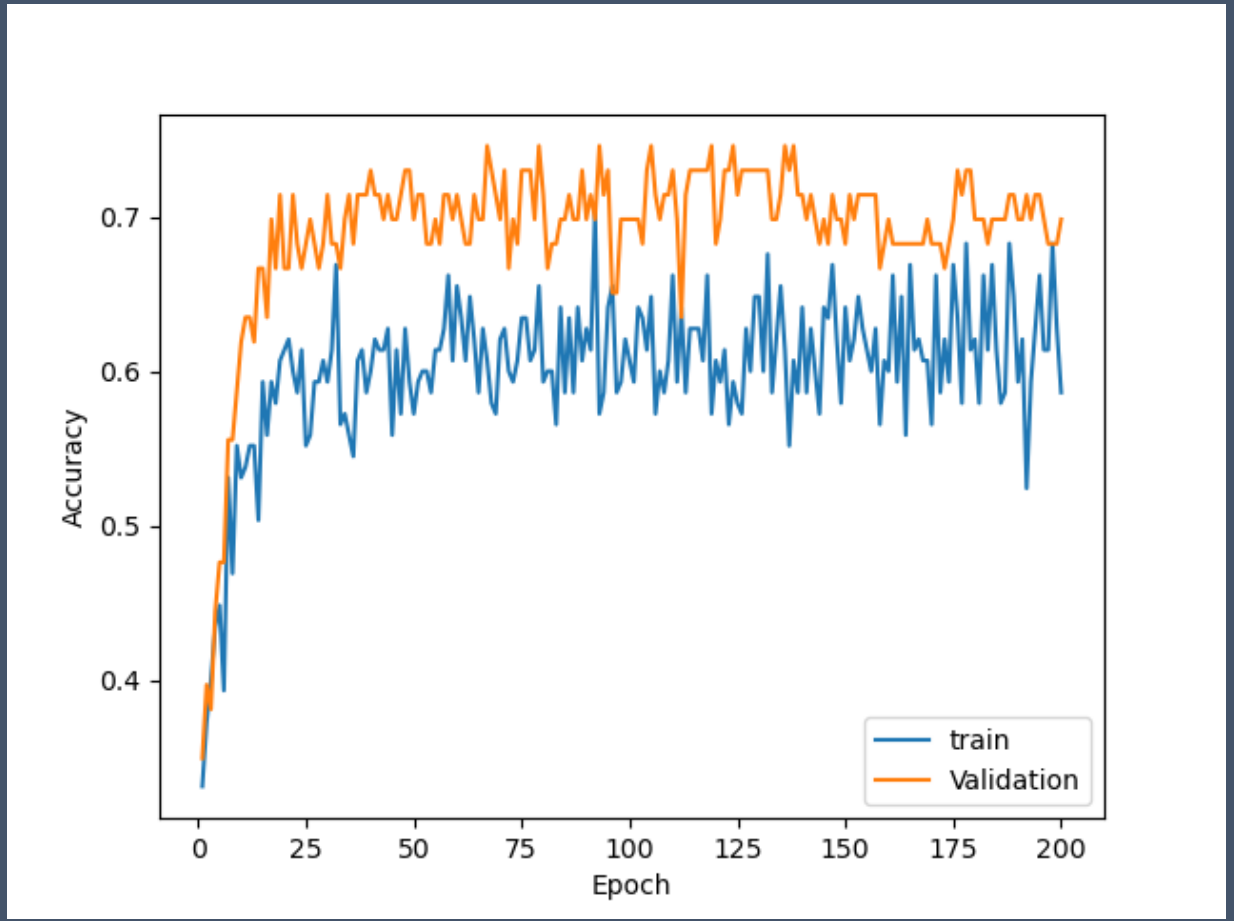


問題点・改善点

Training Acc < Validation Acc

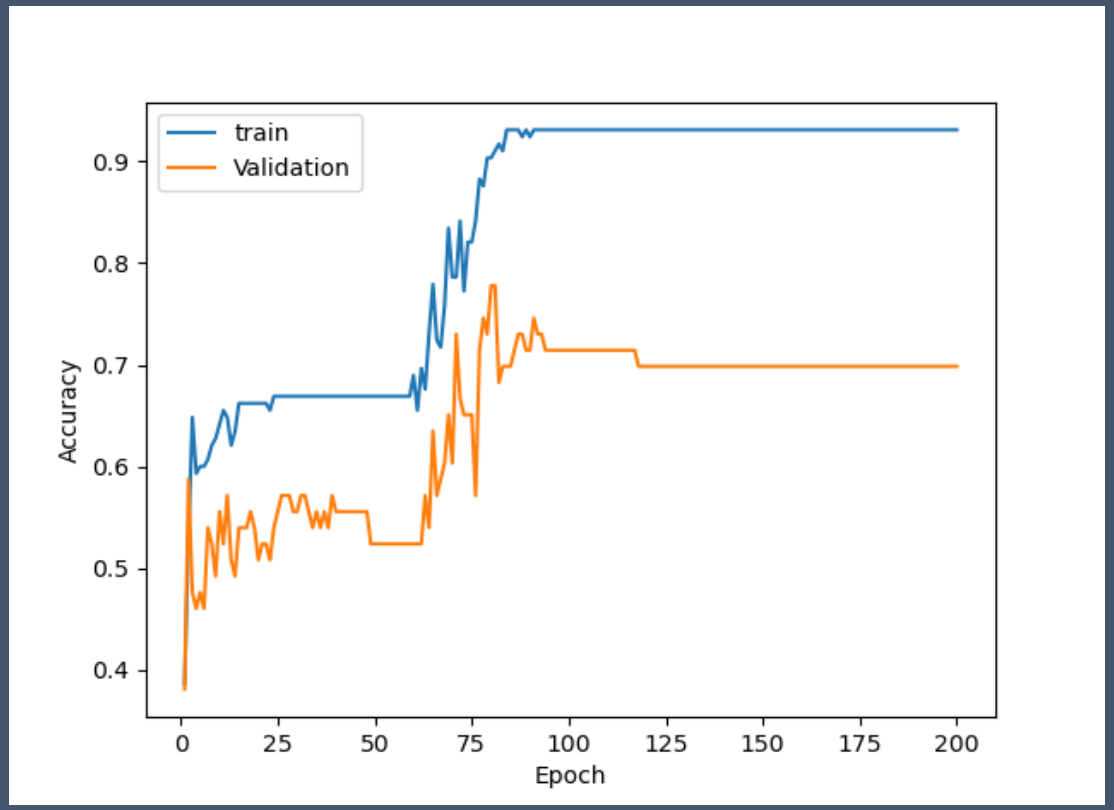
考えられる要因

1. 学習不足点
2. ハイパーパラメータの設定
3. dropout値が大きすぎて重要なlayerの重みを無視



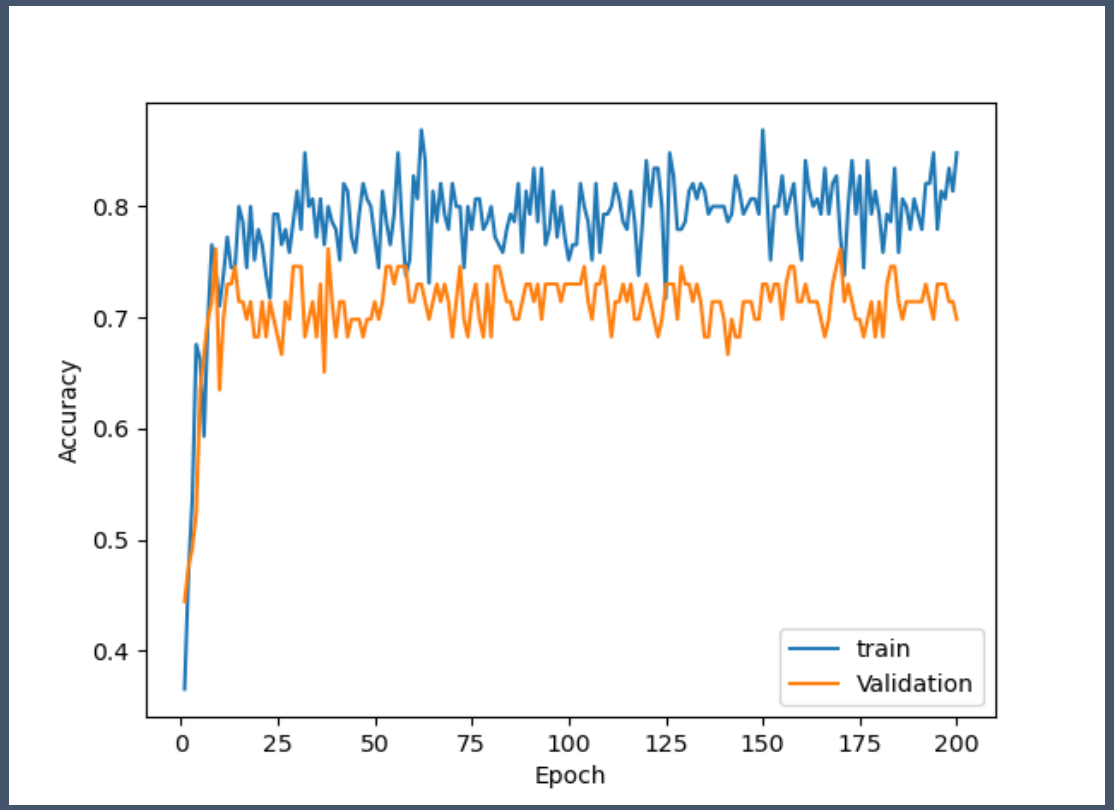
問題点・改善点

Dropoutなし



Validation Acc：0.777778

Dropout 0.25



Validation Acc：0.761904

Dropoutが無い方が良い学習・識別を行えた…

工夫した点

Mixed Precision Training（混合精度トレーニング）[1]
従来ニューラルネットワークのモデル
32 ビットの単精度浮動小数点数 (FP32)
↓
半精度浮動小数点数 (FP16)
モデルの正確度をほぼ落とすことなくトレーニングを高速化

```
scaler = torch.cuda.amp.GradScaler()
```

[1]Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., ... & Wu, H. (2017). Mixed precision training. arXiv preprint arXiv:1710.03740.

感想

- データセットを作成する大変さを痛感
- 研究や他の授業でclassificationはしたことがなかったためやりがいがあった
- 先入観でdropout＝良いものと認識していたため再認識
- 時間があればData AugmentationとしてGANやDiffusion Modelを使ってみたかった

コード

コードはGit上に記載しました.
以下のリンクから確認お願いします.

https://github.com/taipain/final.git