

## 1 2022年6月24日

运行demo code, private score为0.46235。

### 1.1 尝试不同的frames数目

1. 将concat\_nframes设定为5, accuracy为0.567。
2. 将concat\_nframes设定为11, private score为0.63391。
3. 将concat\_nframes设定为21, 会导致RAM爆掉。

### 1.2 尝试不同的hidden layers的数目

1. 将hidden layers从1改为2, private score为0.63567。没有提升。
2. 加深到3层, 将学习率从 $10^{-4}$ 提高到 $10^{-3}$ , private score为0.66980, 有明显提升。
3. 继续加深到4层, Train Acc: 0.689845 Loss: 0.966410 — Val Acc: 0.667523 loss: 1.066279。现在已经开始overfitting了, 之后使用别的技术来解决。
4. 继续加深到5层, Train Acc: 0.688220 Loss: 0.976690 — Val Acc: 0.665157 loss: 1.069590。现在继续加深已经没有用了。

## 2 尝试batch normalisation

首先将激活函数从ReLU改为sigmoid, 结果明显变差。Train Acc: 0.620066 Loss: 1.236119 — Val Acc: 0.616463 loss: 1.251375。

仍然使用ReLU, 在ReLU之前加入BatchNorm1d。Train Acc: 0.706134 Loss: 0.906614 — Val Acc: 0.680002 loss: 1.011591。结果变好。说明batch normalisation在ReLU上就有效果。Private score为0.68313。

尝试组合Sigmoid和BatchNorm1d。Train Acc: 0.674194 Loss: 1.026898 — Val Acc: 0.662758。结果是没有ReLU和BatchNorm1d好的。

## 3 继续尝试不同的frames数目

参考作业提示, 当前应在concat\_nframes这步尝试得不够。将RAM调大, 然后继续5 layers + BatchNorm1D + ReLU。

1. (control)将concat\_nframes设定为11, 之前的结果: Private score为0.68313。
2. 将concat\_nframes设定为21。Private score为0.71052。
3. 将concat\_nframes设定为41。Train Acc: 0.760101 Loss: 0.737564 — Val Acc: 0.709739 loss: 0.937353。Private score为0.71573。有进一步提高。

## 4 尝试加宽神经网络

将`hidden_dim`从256加宽到1024, 尝试能不能让training loss足够小。Train Acc: 0.937122 Loss: 0.174209 — Val Acc: 0.711997 loss: 1.452417。Private score: 0.72765。进一步提高。

可以看到, 这时候已经有明显的 overfitting 了。可以尝试一些 dropout。

## 5 尝试dropout

迭代10轮。

dropout $p = 0.5$ 。Train Acc: 0.665569 Loss: 1.090370 — Val Acc: 0.713079 loss: 0.913127。可以感觉是dropout加太强了。

dropout $p = 0.2$ 。Train Acc: 0.784093 Loss: 0.658822 — Val Acc: 0.750746 loss: 0.807976。Private score: 0.75365。进一步提高。这里dropout起了作用。这里的经验是dropout要起作用, 需要神经网络表达能力, 并且产生了明显的overfitting。dropout的目标是抑制overfitting, 尽量让training loss和validation loss接近。