AMMAI HW-1 Report

ID: r08922a20
Name: 洪筱慈

1. Environment setting

a. Create conda environment
`conda create -n r08922a20 python=3.7.10`
b. Activate environment
`conda activate r08922a20`
c. Install requirements
`pip install -r requirements.txt`
d. Install PyTorch
If your CUDA verison is 10.1:
`conda install pytorch==1.7.1 torchvision==0.8.2 torchaudio==0.7.2 cudatoolkit=10.1 -c pytorch`
Otherwise please refer to https://pytorch.org/get-started/previous-versions/ to find the version compatible for your CUDA.

2. Usage

- Download the pretrained models from here: https://drive.google.com/drive/folders/1yxxLpmg4x5ecDopkhMUgQ3uJoM3n7n30?usp=sharing

a. Test for closed-set: `python main.py --mode closed`
b. Test for open-set: `python main.py --mode open`

The verification results will be saved to `MODE_MODEL-ID_predict.txt`.
Note: If it was not the first time you run the testing script, you can skip the testing data processing by: `python main.py --mode open --data_process 0`

c. Training:
   i. Change the **config.output** in `config.py`. The trained-model will be saved to this folder.
   ii. Run script: `python main.py --mode train`

3. Experiments

Baseline:

- Model and loss: ArcFace, modified from [1].
- Parameter: using original setting.
- Train-val split: 90% for training, 10% for validation.

Variant 1: Use grayscale image as input

- In the open set, I found that a large part of false-negative error samples have the issue of light. The color temperature is different between two face images. So I wonder if that's the reason why the model cannot verify them. For example:

	
Pair 1	Pair 2

I think there are two possible approach to solve it:

(1) Using grayscale: transfer the input image to grayscale, so that the color temperature will not have impact.

(2) Doing data augmentation: generate different kinds of color temperature to each identity.

I first tried on the first approach. Due to the time limitation, I don't implement the second approach.

Variant 2: margin m=0.8

- In ArcFace loss, the parameter $margin\ m$ is the key to separate classes apart from the other. According to the discussion [4]. In the original setting, $m=0.5$. In the class we know that when the m is bigger, the classes are separated more. So I decided to try on $m=0.8$.

Variant 3: margin m=0.2

- Following Variant 2, the discussion[4] mentioned that m should be <<1, but the author chose 0.5. So I'm curious about what happens if I use a smaller margin?

4. Results and Discussion

a. Accuracy:

Model	Closed-set Accuracy	Open-set Accuracy
Baseline	0.601	0.745
Variant 1 (grayscale)	**0.609**	0.740
Variant 2 (m=0.8)	0.586	**0.765**
Variant 3 (m=0.2)	0.584	0.754

b. Discussion

Baseline

It's odd for me that open-set accuracy is higher than closed-set accuracy..... My guess is that there might be some mislabeled pairs in the APD closed-set.

Variant 1: Use grayscale image as input

The following table shows the number of false-positive and false-positive samples for two models:

	# of false-positive	# of false-negative
Baseline	105	289
Variant 1	208	183

From the table above, we can observe that using grayscale input can decrease the false-negative rate. This might imply that without the variance of color temperature, the model can verify a person more easily. However, at the same time the false-positive rate increased. This means that the threshold of distinguishing different people is decreased, so the model tends to verify two people as the same person. According to this result, adding grayscale images to the original data and training them together might help.

Variant 2 & 3:

By increasing the margin from 0.5 to 0.8, there is a 2 % improvement on open-set accuracy, but a 1.5% drop on closed-set. With the assumption that there is mislabeled data in APD, I think that the improvement of open-set is aligned with the conclusion we get on

course: bigger margin helps classes separate apart from each other, and the margin penalty makes the model learn worse when mislabeled data exists.

5. Murmurs

一開始決定用 ArcFace, 找到 deepinsight[1] 的 implement version, 因為有8.9k 顆星星, 因此就採用了, 但是這個版本的 data processing 寫得很不清楚, 並且因為是一個大套件, 且為了要做 distributed training 而將 model 包的比較複雜, 因此我花了很多時間在準備 data , 包成 code 所需要的格式, 才開始做 training。後來才看到另外一個 repo [5]相較之下簡單許多, 也更好做更改, 但是相見恨晚了。從這次經驗裡面學到的教訓是, 雖然遇到問題/bug 就要面對它解決它沒錯, 但是有時解決辦法不一定是技術本身 , 也可以是暫停下來看看有沒有更好的辦法可以用, 可能可以省下更多時間。在 survey 過程中發現了好東西 : [6] , 一個包含各種 PyTorch metric learning loss 的 implementation, 之後有時間的話會想要試試各種 loss。
剛開始作實驗的時候, 看到準確率不高覺得很慌, 亂試了一些實驗, 也嘗試過要不要換 framework。不知道自己在做什麼的時候特別容易覺得壓力超大。某次問 Remy 問題的時候, 他提到老師其實重點是要讓我們有沒有在過程中學到東西, 驗證上課說的內容是不是真的是這樣, 而不只是看準確率。這提醒了我, 想從這堂課獲得什麼？亂換 framework 亂試實驗沒有學到東西, 那不是浪費掉這些時間了嗎？於是才終於靜下心來重新回顧上課學的東西, 從裡面挑一個自己有興趣的小 topic 來嘗試。太晚覺醒了, 這次沒有太多時間可以做嘗試, 但要把這個學習記在心裡, 別讓一開始的結果不好亂了自己的陣腳 , 仔細想想自己想從中學到什麼、好奇什麼 , 專注在這些 , 會讓過程更有趣快樂一點。

6. Reference

[1] ArcFace: https://github.com/deepinsight/insightface/tree/master/recognition/arcface_torch
[2] How to prepare data: https://github.com/deepinsight/insightface/issues/791
[3] Build-Your-Own-Face-Model: https://github.com/siriusdemon/Build-Your-Own-Face-Model
[4] How to choose margin and scale parameters correctly in arcface loss?: https://github.com/KevinMusgrave/pytorch-metric-learning/issues/186
[5] ronghuaiyang/Arcface-pytorch: https://github.com/ronghuaiyang/arcface-pytorch
[6] PyTorch Metric Learning: https://kevinmusgrave.github.io/pytorch-metric-learning/