

## Problem 1: Prototypical Network (70%)

### Report (50%)

#### 1. (20%) Describe the architecture & implementation details of your model.

Ans:

照著助教所提供的 conv-4 架構來當作 feature extractor，每一張圖片經過 feature extractor 後會 flatten 成 1600 長的一維向量，之後再透過 distance function 算出 support data 和 query data 的距離當作是 logits 之後並且跟預測的 label 計算 loss 來訓練，而詳細架構如下：

Training episode : 200

Distant function : Euclidean distance

Optimizer : Adam

Learning rate : 0.001

Data augmentation : Normalize

Meta train : 30-way 1-shot, 15 query data per way

Meta test : 5-way 1-shot, 15 query data per way

**Accuracy on validation set : 47.34 +- 0.87 %**

#### 2. (20%) When meta-train and meta-test under the same 5-way 1-shot setting, please report and discuss the accuracy of the prototypical network using 3 different distance function (i.e., Euclidean distance, cosine similarity and parametric function). You should also describe how you design your parametric function.

Ans:

當 meta-train 和 meta-test 都在 5-way 1-shot 的設定下：

distance function : Euclidean distance → Accuracy: 40.00 +- 0.75 %

distance function : cosine similarity → Accuracy: 38.89 +- 0.75 %

distance function : parametric function → Accuracy: 44.64 +- 0.82%

可以發現在使用 Euclidean distance 的效果會比 cosine similarity 的效果還要好，因此我在設計我的 parametric function(MLP model)時，使用的是六層的 linear+Relu 的架構所構成的，而輸入的 model 時會將 support vector 以及 query vector 進行 Euclidean distance 的計算後再輸入到 model 裡面，可以看到最終結果是比單純用 Euclidean distance 還要好上許多。

3. (10%) When meta-train and meta-test under the same 5-way K-shot setting, please report and compare the accuracy with different shots. (K=1, 5, 10)

Ans:

本題在使用 distance function 為 Euclidean distance 的情況下根據不同的 k 做訓練後，所 test 出來的結果如下：

5 ways 1 shots : accuracy → 40.00%

5 ways 5 shots : accuracy → 61.08%

5 ways 10 shots : accuracy → 68.54%

可以發現如同預期，越多 shots 代表著同一類別的 data 越多，所以當我們得到的資訊越多，理所當然 accuracy 也會跟著上升

## Problem 2: Self-Supervised Pre-training for Image

### Classification (50%)

1. (10%) Describe the implementation details of your SSL method for pre-training the ResNet50 backbone.

Ans:

本題所使用的 SSL pre-trained 的方法為 BYOL，先將同一張圖片、會將同一張圖片分別輸入兩個網路，其中一個網路為 online network 另一個則為 target network，在丟進 target network 時，會將輸入的圖片 augmentation，本題使用的 augmentation 有 RandomGrayscale、RandomHorizontalFlip、GaussianBlur、RandomResizedCrop 等等，這樣形成一個 learner 的架構。

而在 pretrain 的時候則是輸入 Mini-ImageNet 的圖片，將其 resize 成 128\*128 的大小且進行標準化，batch size 為 64，最後再丟入 learner 裡面，使用的 optimizer 為 Adam，learning rate 為 0.0003，通過 100 epochs 訓練了 35.2 小時後，將最後一個 epochs 的 model 的 weights 儲存起來作為我們的 pretrain weight。

2. (10%) Following Problem 2-1, please conduct the Image classification on Office-Home dataset as the downstream task for your SSL method.

Ans:

Setting	Pre-training(Mini-ImageNet)	Fine-tuning(Office-Home dataset)	Classification accuracy on valid set
A	-	Train full model(backbone + classifier)	19.21%
B	w/ label	Train full model(backbone + classifier)	31.53%
C	w/o label (SSL)	Train full model(backbone + classifier)	38.92%
D	w/ label	Fix the backbone. Train classifier only	23.89%
E	w/o label (SSL)	Fix the backbone. Train classifier only	28.82%

最終結果為，在 performance 上(accuracy)， $C > B > E > D > A$ 。

3. (10%) Discuss or analyze the results in Problem 2-2

Ans:

可以看到根據結果，有使用 pre-train weight 的效果會好很多相對於完全沒有使用 pre-train 的 setting A，而使用 supervise learning 在不同的 dataset 上做 pre-train 的效果會比 SSL 還要差；訓練時如果只訓練 classifier 的部分所得到的結果也會比訓練全部的 model 還要差。最終我們得到的最佳結果是 setting C，透過 SSL pre-train 在 Mini-ImageNet 上面後，再 fine-tune 在 Office-Home dataset 上得到的結果為 38.92%。

Reference:

<https://github.com/yinboc/prototypical-network-pytorch>

[https://github.com/lucidrains/byol-](https://github.com/lucidrains/byol-pytorch/tree/caa65d7c19a80a3611f4049c638ecd8401f8f3c4)

[pytorch/tree/caa65d7c19a80a3611f4049c638ecd8401f8f3c4](https://github.com/lucidrains/byol-pytorch/tree/caa65d7c19a80a3611f4049c638ecd8401f8f3c4)