

Problem1-Few-Shot Learning – Prototypical Network

- [Problem 1-1] Implement the prototypical network as your baseline model to perform 5-way 1-shot classification.

- a. Describe the architecture & implementation details of your model.

Ans: 在本次實驗的模型設定，使用投影片上的 Conv-4 架構，並使用優化器 Adam，其 learning rate 設為 0.001，在 distance function 上選擇 Euclidean distance 進行分類。在模型訓練設定上，在 meta-training 上是設定 5-way 1-shot，訓練 200 個 epoch，每個 epoch 有 100 個 training tasks；在 meta-test 也是設定 5-way 1-shot。

- b. Clearly mark out a single final result on the validation set.

Ans: 在 validation set 上的 mean accuracy 為 **45.22%**，其 95%信賴區間為 $45.22 \pm 0.82 \%$ 。

```
(dlcv-hw4) wayne@wayne:~/Desktop/DLCV/hw4$ python eval.py output1.csv hw4_data/mini/val_testcase_gt.csv
Accuracy: 45.22 +- 0.82 %
```

- [Problem 1-2] 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 functions.

Ans: 對於 3 種不同的 distance function，其中 parametric function 的設計為 learnable model，程式架構如下圖，將兩個想要關注的 feature embeddings (其中為一 prototype representation feature 和 query feature) 作 concatenation，其 concatenation 後的 feature embedding 維度變為 3200，經過 parametric function 後會輸出一個值，這個值代表這兩個 feature embeddings 的 distance。

對於評估這 3 種 distance function，皆訓練 150 個 epochs 後，從 validation set 中隨機選取並使用 600 個 episodes，mean accuracy 結果如下表所示，可以發現 Cosine similarity 效果沒有其他兩個好。

```
class Parametric(nn.Module):
    def __init__(self, input_dim=3200, out_dim=1):
        super().__init__()
        self.mlp = nn.Sequential(
            nn.Linear(input_dim, 512),
            nn.ReLU(),
            nn.Linear(512, 128),
            nn.ReLU(),
            nn.Linear(128, out_dim)
        )
    def forward(self, x):
        return self.mlp(x)
```

distance	Euclidean distance	Cosine similarity	Parametric function
Mean accuracy (%)	45.98	37.07	44.62

- [Problem 1-3] When meta-train and meta-test under the same 5-ways K-shot setting, please report and compare the accuracy with different shots.

Ans: 在本次實驗的設定下，使用 Euclidean distance 這個 distance function，皆訓練 150 個 epochs 後，從 validation set 中隨機選取並使用 600 個 episodes 計算 mean accuracy，validation 結果如下表所示，可以發現當 K 值越大 mean accuracy 相對提升，代表在計算 prototype feature representation 時每個 class 有較多的 sample 可以做參考，進而有較高的可信度。另一個不同的地方在於訓練階段的時候，越大的 K 值訓練 mean accuracy 也普遍比較高。

shots	1	5	10
Mean accuracy (%)	45.28	63.32	70.46

Problem2-Self-Supervised Pre-training for Image Classification

- [Problem 2-1] Describe the implementation details of your SSL method for pre-training the ResNet50 backbone.

Ans: 在本次 self-supervised learning 實驗中，使用的方法為投影片上建議的 BYOL，套用的 backbone 為 torchvision package 的 models 中 resnet50，BYOL 設定的參數如下: BYOL(resnet, image_size=128, hidden_layer='avgpool')，其他訓練參數如優化器 Adam (learning rate=0.0003)、batch size 為 128、epoch 為 100，在 Mini-ImageNet 訓練資料上進行 self-supervised 訓練。

- [Problem 2-2] Following Problem 2-1, please conduct the Image classification on Office-Home dataset as the downstream task for your SSL method. Also, please complete the following table.

Ans: 在所有 setting 下的 classifier layer，均使用一層的 nn.Linear(2048, 65)進行 downstream task 的 fine-tuning(訓練 50 個 epoch)，各個 setting 在 validation set 上的表現如下表所示，其中在 setting C 的模型表現為 **47.0%**。

setting	Pre-training(Mini-ImageNet)	Fine-tuning (Office-Home dataset)	Accuracy (%)
A	-	Train full model	32.3
B	w/ label (TA backbone)	Train full model	37.4
C	w/o label (SSL pre-training backbone)	Train full model	47.0
D	w/ label (TA backbone)	Fix the backbone. Train classifier only.	32.8
E	w/o label (SSL pre-training backbone)	Fix the backbone. Train classifier only.	31.8

- [Problem 2-3] Discuss or analyze the results in Problem 2-2.

Ans: 觀察 Problem 2-2 的 table 裡的 5 個 setting，由於 Office-Home dataset 的訓練資料量相對較少，直接 train from scratch 的表現較其他 4 個 setting 表現較差。比較 setting B 和 C，發現使用 self-supervised learning 可以讓 backbone 學習比較好的模型參數，使得在 downstream task 下再 fine-tuning 後表現較好；比較 setting D 和 E，可以發現兩個的表現差不多，可能的原因在於使用簡單的一層 classifier layer。最後比較更新整個模型參數和只更新 classifier layer 參數(B v.s. D / C v.s. E)，可以發現前者設定(更新整個模型參數)普遍表現較好，其可能的原因在於 downstream task 需要有別於其他 dataset 訓練的 backbone，搭配對應的 classifier layer，才可以表現較好。

Reference

- [1] Prototypical-network-pytorch: <https://github.com/yinboc/prototypical-network-pytorch>
- [2] byol-pytorch : <https://github.com/lucidrains/byol-pytorch>