Homework #4

Deep Learning for Computer Vision

Due: 111/01/04 (Tue.) 02:00 AM due

電機所 R09921102 曾翊維

Problem 1: Few-Shot Learning - Prototypical Network

1. (20%) Describe the architecture & implementation details of your model. (Include but not limited to the number of training episodes, distance function, learning rate schedule, data augmentation, optimizer, and N-way K-shot setting for meta-train and meta-test phase) Please report the accuracy on validation set under 5-way 1-shot setting (during inference).

   Acc on validation set under 5-way 1-shot meta-test = 43.31 +- 0.79 %

   Training epochs: 200
   Training episodes: 100
   Distance function: euclidean distance
   Learning rate schedule: StepLR(optimizer, step_size=20, gamma=0.5)
   Optimizer: Adam
   10-way 1-shot for meta-train and 5-way 1-shot for meta-test

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.

   Parametric function : nn.Linear(1600*5, 5)
   Input: query - proto (1600*1)
   Since the setting is 5-way, we have five protos. The size of function input is thus 1600*5 after concatenation.

Distance function	Acc
Euclidean	37.08%
Cosine similarity	37.42%
Parametric function	20.00%

I have tried another deeper model architecture for parametric function as follows.

```
self.layer = nn.Sequential(
        nn.Linear(1600*5, 512),
        nn.BatchNorm1d(512),
        nn.ReLU(),
        nn.Linear(512, 5),
 )
```

However, both of them don't work well (acc ≈ 20%), which may results from low training epochs (80 for problem 2-2). That is, It hasn't been trained long enough.
Another possible factor may be the model architecture. A conv1d with 1600-kernel-size and 1600-stride might be better than these two linear models.
The other possible factor may be the input. abs(query - proto) may outperform (query - proto).

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)

   As expected, the accuracy increases since the protos (mean vector of K images) for each type become more representative when larger K being used.

K	Acc
1	35.96%
5	59.72%
10	65.64%

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

1. (10%) Describe the implementation details of your SSL method for pre-training the ResNet50 backbone. (Include but not limited to the name of the SSL method you used, data augmentation for SSL, learning rate schedule, optimizer, and batch size setting for this pre-training phase)

   Method: BYOL
   Data augmentation: default in LINK
   Epochs: 100
   Learning rate: 3e-4
   optimizer: Adam
   Batch size: 32

2. (10%) 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, which contains different image classification setting, and compare the results.

Setting	Pre-training (Mini-ImageNet)	Fine-tuning (Office-Home dataset)	Classification accuracy on valid set (Office-Home dataset)
A	-	Train full model (backbone + classifier)	27.83%
B	w/ label (TAs have provided this backbone)	Train full model (backbone + classifier)	31.03%
C	w/o label (Your SSL pre-trained backbone)	Train full model (backbone + classifier)	39.90
D	w/ label (TAs have provided this backbone)	Fix the backbone. Train classifier only	22.91
E	w/o label (Your SSL pre-trained backbone)	Fix the backbone. Train classifier only	35.47

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

   A: Without per-training on large dataset, it makes sense that directly training on insufficient number of images leads to poor result.

   B: With supervised learning pre-training backbone on large dataset followed by full model training on small dataset, the performance is better than A.

   C: Self-supervised learning pre-training backbone on large dataset followed by full model training on small dataset yields the best result because the SSL model are trained without labels and thus has the ability to extract features from images of all classes rather than certain classes from certain datasets.

   D: Compared to B, fixing the backbone leads to worse accuracy because the backbone is trained on large dataset whose classes (e.g. airplane, bird, cat) are different from the small office-home dataset (e.g. printer, chair, glasses)

   E: Compared to C, the accuracy drops due to the same reason as B and D.

Ref:

1. https://github.com/orobix/Prototypical-Networks-for-Few-shot-Learning-PyTorch
2. https://github.com/yinboc/prototypical-network-pytorch
3. https://www.cnblogs.com/marsggbo/p/11308889.html