Hw4 m11015q11 柯鐘奇

Problem 1.

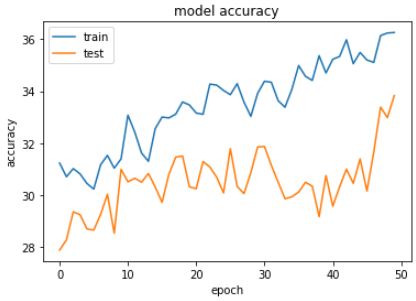
1. Model architecture

```
Convnet(
  (encoder): Sequential(
    (0): Sequential(
      (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (1): Sequential(
      (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (2): Sequential(
      (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (3): Sequential(
      (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
  )
)
```

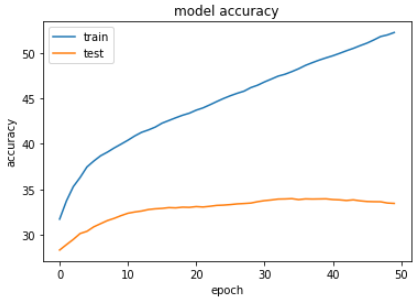
Details

training episodes	600*200 = 120000
distance function	Euclidean distance
learning rate	1e-4
data augmentation	None
optimizer	Adam
Meta-train	10 way-1 shot
Accuracy	43.35 +- 0.78 %

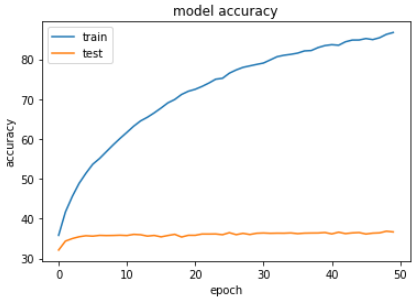
2.



Euclidean distance



Cosine similarity



Parametric function

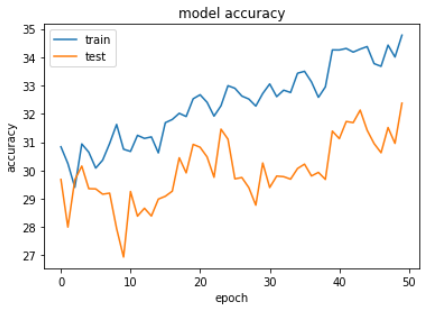
Parametric：將提取出來的特徵相乘，再將結果經過 fully-connected 學習。Loss 採用原本的 Cross Entropy Loss。

```
MLP(
  (fc): Sequential(
    (0): Linear(in_features=1600, out_features=512, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=512, out_features=1, bias=True)
  )
)
```

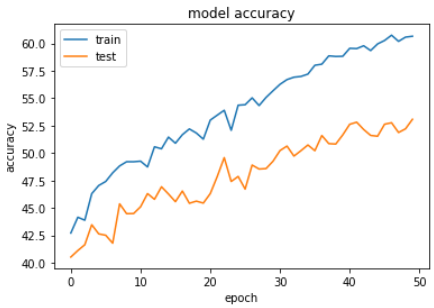
分析：

在相同設置下(Learning rate,)，Euclidean distance 收斂的速度相較於其他兩個慢了許多，epoch50 的情況下，Cosine similarity 及 Parametric function 的結果約在 35 上下，Euclidean distance 則在 32 學習較慢，但成長空間看起來比較大。

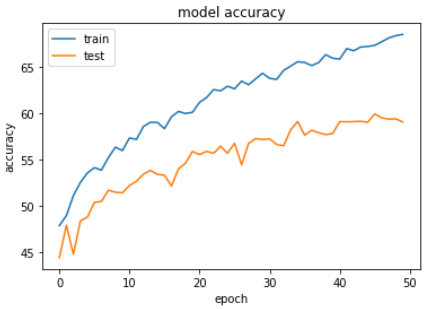
3.



K=1



K=5



K=10

分析：參考的照片樣本更多，更能夠找出適合的 prototype，因此 accuracy 更高是合理的表現。

Problem2.

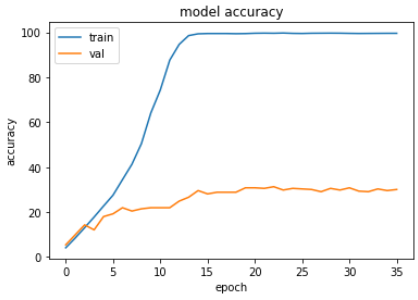
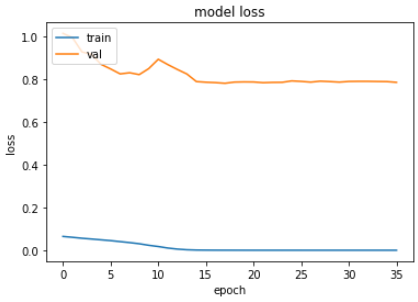
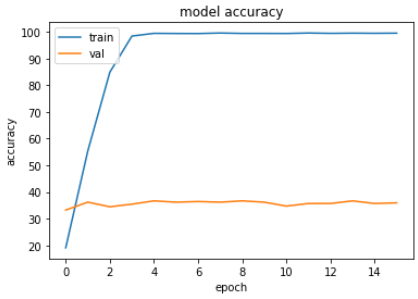
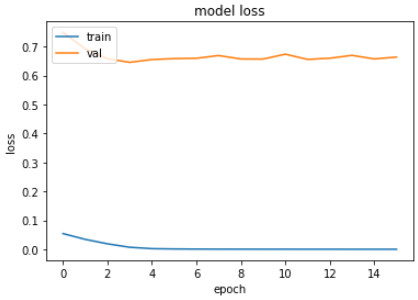
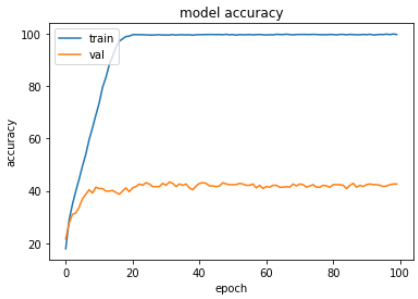
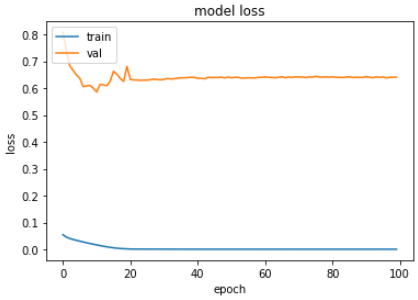
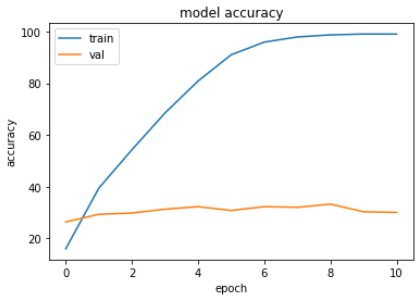
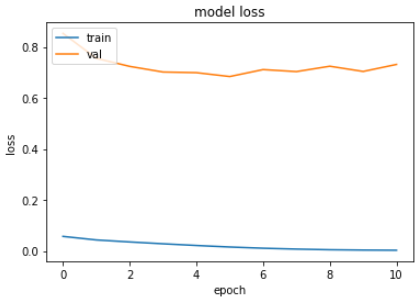
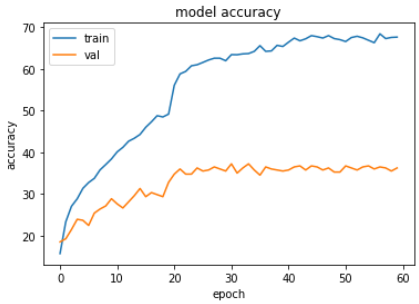
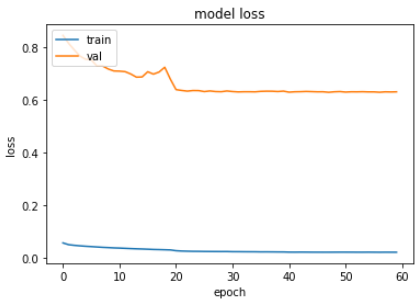
1. (Pre-training/Fine-tuning)

SSL method	BYOL	
data augmentation	BYOL default	None
learning rate	3e-4	3e-4 w/ stepLR
optimizer	Adam	
Batch-size	20	64
Accuracy	41%	

2.

Setting	**Pre-training (Mini-ImageNet)**	**Fine-tuning (Office-Home dataset)**	**Classification accuracy on valid set (Office-Home dataset)**
A	-	Train full model (backbone + classifier)	30%
B	w/ label (TAs have provided this backbone)	Train full model (backbone + classifier)	36%
C	w/o label (Your SSL pre-trained backbone)	Train full model (backbone + classifier)	41%
D	w/ label (TAs have provided this backbone)	Fix the backbone. Train classifier only	33%
E	w/o label (Your SSL pre-trained backbone)	Fix the backbone. Train classifier only	36%

3.

Setting	acc	loss	analyze
A 30%			dataset 太少 Train from scratch 沒有 freeze 任何層， ep:15 即 overfitting。
B 36%			因為用 supervise learning pretrain on Mini-ImageNet 所以此 resnet 只 fix 在 Mini-ImageNet ，因此用 office dataset 做 finetune 效果不好。
C 41%			因為用 SSL 方式做 pretrain，因此 resnet50 學出來的 feature extractor 不會只 fit 在 Mini-ImageNet，相較於 B 效果更好
D 33%			因為把 backbone fix 住，更加 fit 在 Mini-ImageNet 上，相較於 B 有 fine-tune backbone 的情況下結果更差。
E 36%			SSL 學習的是彼此關係的差距，因此就算把 backbone fix 住，僅 finetune 後面的 fc 層效果仍然顯著。

Reference

同學	M11015Q02 柯元豪 M11015Q12 黃柏翰
計算距離	Calculate Dist
Flatten	How does the list comprehension to flatten a python list work?
COSINE_SIMILARITY	TORCH.NN.FUNCTIONAL.COSINE_SIMILARITY
Sampler	Pytorch Sampler 详解
Sampler	TORCH.UTILS.DATA
Sampler	一文弄懂 Pytorch 的 DataLoader, DataSet, Sampler 之间的关系