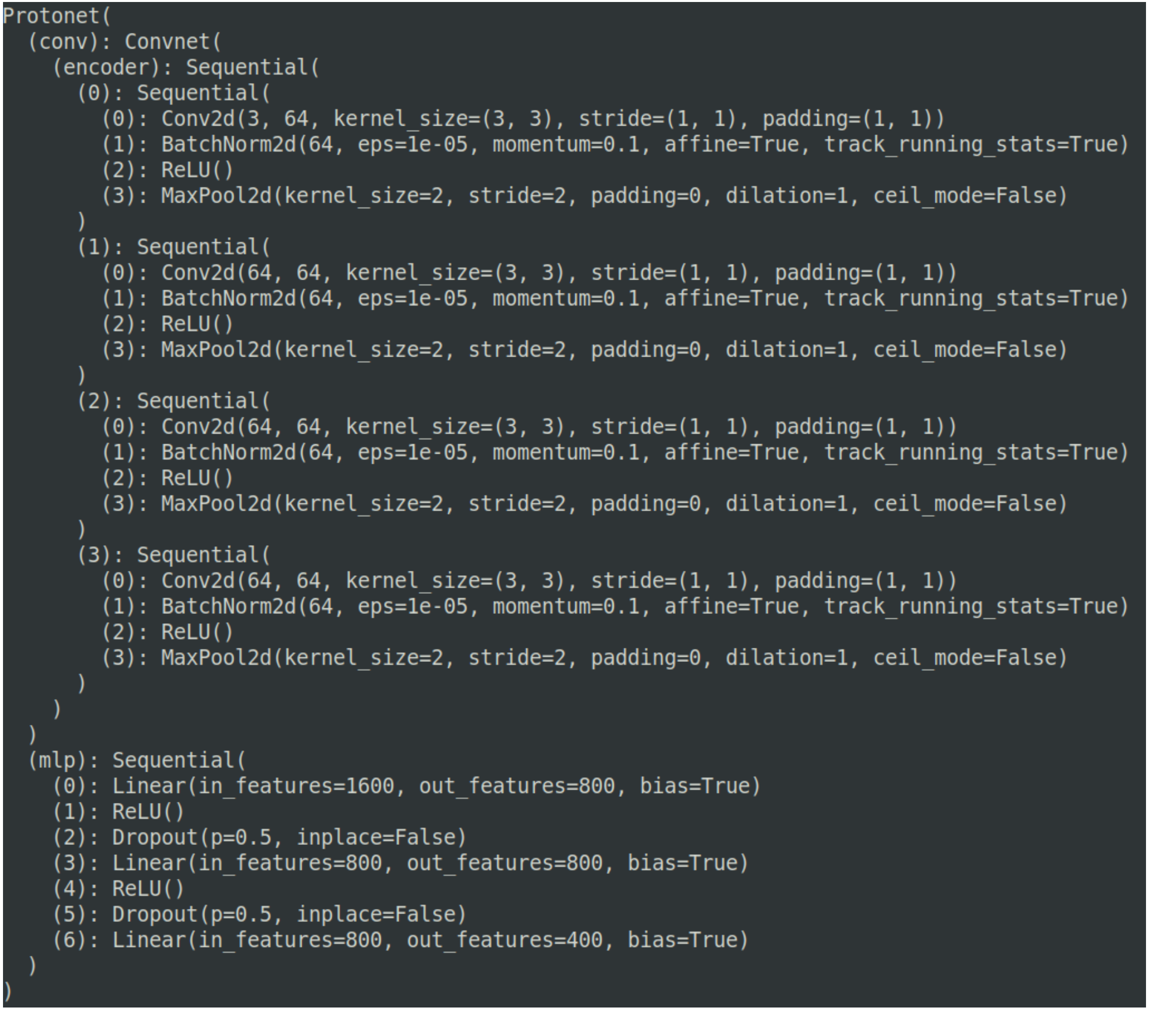
DLCV HW4

學號：B08502141　　姓名：石旻翰　　系級：電機三

1 Problem1

1.1 Model Architecture

the number of training episodes: 600
distance function: Euclidean distance
learning rate schedule: lr * 0.9 for every 40 epochs
data augmentation: None
optimizer: Adam(lr=1e-4, betas=(0.9, 0.999), weight_decay=1e-6)
5-way 1-shot setting for meta-train and 5-way 1-shot for meta-test phase

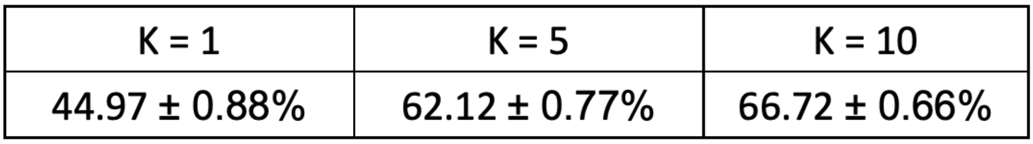


```
Protonet(
  (conv): Convnet(
    (encoder): Sequential(
      (0): Sequential(
        (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
        (3): MaxPool2d(kernel_size=2, 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)
      )
    )
  )
  (mlp): Sequential(
    (0): Linear(in_features=1600, out_features=800, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=800, out_features=800, bias=True)
    (4): ReLU()
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=800, out_features=400, bias=True)
  )
)
```

1.2 Accuracy

Euclidean: 44.97 ± 0.88%
Parametric: 42.46 ± 0.83%
Cosine : 40.17 ± 0.83%

1.3 Compare



K = 1	K = 5	K = 10
44.97 ± 0.88%	62.12 ± 0.77%	66.72 ± 0.66%

The parameters used in this part are the same with those in p1-1. I think the reason why it progresses as the k increase is it sees more images, so its accuracy also become higher.

2 Problem2

2.1 Implementation of Backbone

SSL method: BYOL
Data augmentation for SSL: No change from TA provided one
learning rate schedule: get_cosine
optimizer:Adam(learner.parameter(), lr=3e-4)
batch size: 64

2.2 Classification Accuracy on ValidSet

A: 35%
B: 38%
C: 41%
D: 22%
E: 28%
Because time is limited, so setting A, B, D, E, I only run them for a few epochs.

2.3 Analysis

It seems that methods which backbone fixed but training classifier, like D, E, have worse performances than methods that training backbone + classifier, like B, C.

References

[1] https://github.com/kai860115/DLCV2020-FALL/tree/main/hw4

[2] https://towardsdatascience.com/train-without-labeling-data-using-self-supervised-learning-by-relational-reasoning-b0298ad818f9