tags: `dlcv`

hw4

1.

1.

I applied autoaugment, colorjitter, randomaffine randomly.
Each epoch I train for 8000 episodes, total 50 epochs.
Train in a 10-way 1 shot 1 query setting. I used parametric function to evaluate distance. Optimizers for the Convnet and MLP are tuned by warmup schdulers.
Result on testcase.csv:
52.45 ± (0.90) %

2.

euclidean:0.4962
parametric:0.5245
cosine: 0.4398
Since the distance funciton is learned rather than hard-coded, parametric function outperforms euclidean and cosine similarity.
Parametric: implemented by concating two feature vectors. For example, given query tensor of shape (75, 1600), where 1600 is the feature dimension, and 5-way prototype tensor (5, 1600), first expand query to (75, 5, 1600) and proto to (75, 5, 1600), and concat to (75, 5, 3200), then pass to a MLP to have output (75, 5, 1)

3.

Using parametric function:
K=1: 0.5245
K=5: 0.68
K=10: 0.7036
With more shots, the accuracy gets higher.

2.

1.

I used BYOL to train my ssl backbone, I randonly apply transformation on my image to strike a balance between data diversity and the loss of the model. And my MLP has 3 linear layers. Pretrain batch size=32

2.

setting	acc
A	0.1895
B	0.3880
C	0.4187
D	0.3137
E	0.3545

3.

From the table we observe that without pretraining, the performance is sloppy. And fine-tuned backbone is better than fixed backbone. Self-pretrained backbone performs better than TA’s, I think this illustrates the powerfulness of BYOL.