DLCV HW 3

B08902134, 曾揚哲

1 Prototypical Network

1. Implementation Detail

   I simply applies the scenario TA required for the baseline model, i.e. the 5-way 1-shot setting. I uses the parametric function with an MLP containing `Linear`, activation function `ReLU` and `Dropout`. For training, the MLP for after the provided `ConvNet` consist of `Linear`, activation function `RReLU` and `Dropout`. Moreover, I trained by `AdamW` optimizer with initial learning rate of 1×10^{-4} and `StepLR` scheduler. Moreover, I use `(0.9, 0.98)` as beta for smoother loss decrease.

   The accuracy on validation set is approximately 47.98 ± 0.91 %.

2. Comparison on different distance function

   The accuracy with different distance is as below.

Distance	Accuracy
Euclidean distance	24.9%
Cosine similarity	42.13
Parametric function	46 %

   My parametric function is a simple MLP with the following structure.

   ```
   Linear(800, 400),
   ReLU(),
   Dropout(),
   Linear(400, 1)
   ```

   Discussion: Apparently, cosine similarity performs far better than euclidean distance. Cosine similarity is calculated using only the dot product and affected only by the terms the two vectors have in common, whereas Euclidean has a term for every dimension which is non-zero in either vector. Therefore, usually cosine similarity is better to measure the semantics of inputs. [1]

3. Comparison on different different setting

   The accuracy with different settings is as below.

Setting	Accuracy
5-way 1-shot	45.97
5-way 5-shot	45.07
5-way 10-shot	43.07

2 Self-Supervised Pre-training

1. Implementation detail

   I pretrain the `ResNet50` backbone with BYOL. I replace the fully connected layer with the output dimension given as a parameter for customization. I train for 200 epochs using AdamW as optimizer with learning rate 5×10^{-3} with `ReduceLRonPlateau` scheduler and a batch size of 128.

2. Image Classification on **Office-Home** dataset

Setting	Pretraining	Fine-tuning	Accuracy
A	-	Full model	0.27
B	Supervised	Full model	0.38
C	SSL	Full model	0.35 ~ 0.41 (individual) 0.47 (ensemble)
D	Supervised	Backbone fixed	0.19
E	SSL	Backbone fixed	0.14

3. Discussion

   During fine-tuning, the performance are far worse if we fixed the backbone, even worse than training from scratch, indicating that we still have to transfer the backbone from the mini-Image dataset to Office-Home dataset. The two fixed backbone scenarios indicate the robustness of the pretrained backbones, and my experiment result did not meet the expectation where SSL pretraining should prevail over the Supervised method. Therefore, I would like to try to fine tune the pretrain process if time permits. Supervised pretrained backbone may be easily fit into the original domain, while the result SSL pretrained may be better because the backbone should be more general.

References

[1] https://www.quora.com/Why-is-cosine-similarity-better-in-measuring-similarity-between-vectors-than-Euclidean-in-Vector-Space-model