LEARNING TO SEMI-SUPERVISED FEW-SHOT LEARN

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent progress made in few-shot learning featured meta-learning, in which training consists of mini-episodes that resemble testing. In this work, we advance the meta-learning paradigm by augmenting it with a semi-supervised learning procedure. We extended the prototypical networks with capability of making representation refinement on unlabeled data, and we formulate new end-to-end models that improve the robustness towards unrelated noisy examples. Our proposed methods are evaluated on Omniglot, mini-ImageNet, and a larger split of ImageNet, and observe consistent gain in performance with our proposed meta semi-supervised learning model.

1 INTRODUCTION

2 BACKGROUND

2.1 FEW-SHOT LEARNING

2.2 PROTOTYPICAL NETWORKS

3 SEMI-SUPERVISED PROTOTYPICAL NETWORKS

We denote our training set as a tuple of labeled and unlabeled examples: $(\mathcal{S}, \mathcal{R})$. The labeled portion is also called the support set in few-shot learning literature, containing a list of tuples of inputs and targets: $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_N, y_N)\}$. Each $\boldsymbol{x}_i \in \mathbb{R}^D$ is some input vector of dimension D, and $y_i \in \{1, 2, ..., K\}$ is a class label. In addition to classic few-shot learning, we introduce an unlabeled set $\mathcal{R}$ containing only inputs: $\mathcal{R} = \{\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, ..., \tilde{\boldsymbol{x}}_M\}$. In this paper, we refer the unlabeled portion "refinement set". To evaluate the performance of the model, there is a query set $\mathcal{Q}$ (i.e. test set), also contraining tuples of inputs and targets. Figure shows the structure of our training episode.

[[Here include a figure of our training episode]]

3.1 PROTOTYPICAL NETWORKS WITH SOFT K-MEANS

One way to refine the prototypes is to model it as a semi-supervised clustering problem. Adapted from the classical k-means algorithm, we can initialize the cluster centers at the prototypes, and infer the latent class assignment for unlabeled examples. First, we start with the standard prototypical networks formulation, initializing the prototypes at the mean representation of the embeddings of different classes:

$$z_{i,k} = \mathbb{1}[y_i = k] \tag{1}$$

$$\boldsymbol{p}_k = \frac{\sum_i \boldsymbol{x}_i z_{i,k}}{\sum_i z_{i,k}} \tag{2}$$

Then, the unlabeled examples get some soft assignment ($z_{j,k}$) to the clusters based on their Euclidean distance to the cluster centers. The assignment is normalized across all clusters, and the prototypes

gets refined by incorporating these unlabeled examples.

$$z_{j,k} = \frac{\exp\left(-||\tilde{\boldsymbol{x}}_j - \boldsymbol{p}_k||_2^2\right)}{\sum_{k'} \exp\left(-||\tilde{\boldsymbol{x}}_j - \boldsymbol{p}_{k'}||_2^2\right)} \tag{3}$$

$$\tilde{\boldsymbol{p}}_k = \frac{\sum_i \boldsymbol{x}_i z_{i,k} + \sum_j \tilde{\boldsymbol{x}}_j z_{j,k}}{\sum_i z_{i,k} + \sum_j z_{j,k}} \tag{4}$$

Lastly, given an unseen input $\mathbf{x}$, we can classify it based on our refined prototypes.

$$\hat{y} = \underset{k}{\operatorname{argmax}} \left\{ \frac{\exp\left(-||\boldsymbol{x} - \tilde{\boldsymbol{p}}_k||_2^2\right)}{\sum_{k'} \exp\left(-||\boldsymbol{x} - \tilde{\boldsymbol{p}}_{k'}||_2^2\right)} \right\} \tag{5}$$

3.2 Extra high variance cluster model

In practical usages of few-shot classification, there may be other unlabeled items belonging to some unknown class. For example, if we would like our model to distinguish between a unicycle and a scooter, it is probably not realistic to assume we have many other examples exclusively of these two classes; however, we are able to find some similar images on the web that may or may not belong to classes of interest. As regular K-means try to assign all unlabeled images to one of the clusters, unrelated noisy items may be harmful to the final performance. One simple model to exclude the cluster is to create another extra cluster to capture the outliers.

$$\boldsymbol{p}_k = \begin{cases} \frac{\sum_i \boldsymbol{x}_i z_{i,k}}{\sum_i z_{i,k}} & \text{for } k = 1...K \\ \mathbf{0} & \text{for } k = K+1 \end{cases} \tag{6}$$

We use Gaussian distribution with different radius to represent the high variance distractor cluster. For simplicity reason, we set $r_{1...K}$ to 1 in our experiments, and only learning the radius of the extra cluster.

$$z_{j,k} = \frac{\exp\left(-\frac{1}{r_k^2}||\tilde{\boldsymbol{x}}_j - \boldsymbol{p}_k||_2^2 - A(r_k)\right)}{\sum_{k'} \exp\left(-\frac{1}{r_k^2}||\tilde{\boldsymbol{x}}_j - \boldsymbol{p}_{k'}||_2^2 - A(r_k)\right)} \tag{7}$$

$$A(r) = \frac{D}{2}\log(2\pi) + D\log(r) \tag{8}$$

3.3 Soft mean-shift model

Modeling the noisy examples using an extra cluster maybe over-simplistic. Since the extra cluster may have many modes. For example, when searching for unicycles or scooter, there will probably be a cluster of bicycles that enters into the set of unlabeled images and its distribution may not be well represented by a high variance uni-modal distribution. In our experiments, we manually constructed the noisy examples that form a multi-modal distribution, and we observe that although the extra cluster can alleviate the problem of drifting prototypes towards noises, this can be

3.4 Loss function

The loss function for prototypical network is defined as the classification cross entropy loss for query example tuple $(\boldsymbol{x}, y)$:

$$\mathcal{L}(\boldsymbol{x}, y, \{\boldsymbol{p}_k\}) = -\sum_k \mathbb{1}[y == k] \log\left(\frac{\exp(-||\boldsymbol{x} - \boldsymbol{p}_k||_2^2)}{\sum_{k'} \exp(-||\boldsymbol{x} - \boldsymbol{p}_{k'}||_2^2)}\right) \tag{9}$$

In practice, we found that the average loss before and after prototype refinement yields better performance.

$$\bar{\mathcal{L}} = \frac{1}{2}\mathcal{L}(\boldsymbol{x}, y, \{\boldsymbol{p}_k\}) + \frac{1}{2}\mathcal{L}(\boldsymbol{x}, y, \{\tilde{\boldsymbol{p}}_k\}) \tag{10}$$

4 Related Work

5 Experiments

5.1 Datasets

We evaluate the performance of our model on three datasets: two benchmark few-shot classification datasets and a novel large-scale dataset that we hope will be useful for future few-shot learning work.

Omniglot (Lake et al., 2011) is a dataset of 1,623 handwritten characters from 50 alphabets. Each character was drawn by 20 human subjects. We follow the few-shot setting proposed by Vinyals et al. (2016), in which the images are resized to 28×28 pixels and rotations in multiples of 90° are applied, yielding 6,492 classes in total which are split into 4,800 training classes and 1,692 classes for test. The training alphabets are distinct from test alphabets except for the Gurmukhi alphabet, for which 41 characters plus rotations are in the training split and 4 characters plus rotations are in test.

***mini*ImageNet** (Vinyals et al., 2016) is a modified version of the ILSVRC-12 dataset (Russakovsky et al., 2015), in which 600 images for each of 100 classes were randomly chosen. We use the splits introduced by Ravi & Larochelle (2017) in order to be able to compare with the latest work in few-shot classification. These splits use 64 classes as training, 16 for validation, and 20 for test. All images are of size 84×84 pixels.

- mini-ImageNet Vinyals et al. (2016); Ravi & Larochelle (2017)
- hierImagenet

evaluated on Omniglot Lake et al. (2011),

5.2 Results

6 Conclusion

References

Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33th Annual Meeting of the Cognitive Science Society, CogSci 2011, Boston, Massachusetts, USA, July 20-23, 2011*, 2011.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations*, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3630–3638, 2016.