Meta & Multi-task Learning Assignment 2

AIGS Lee hyo jeong 20215539

1. Prototypical Networks for Few-shot Learning

Few-shot classification is a task in which a classifier must be adapted to accommodate new classes not seen in training, given only a few examples of each of these classes. A naïve approach to this task could be re-training the model on the new data which often ends up overfitting.

This paper propose prototypical networks for the problem of few-shot classification. To alleviate upper mentioned key issue of overfitting, prototypical networks learn a metric space in which classification can be performed by computing distances to prototype representation of each class. This approach based on the idea that there exists an embedding in which points cluster around a single prototype representation for each class. Overall algorithm can be summarized as 1) Learn a non-linear mapping of the input into an embedding space using a neural network 2) Take a class's prototype to be the mean of its support set in the embedding space 3) Classification is then performed for an embedded query point by simply finding the nearest class prototype. For the case of zero-shot learning; here each class comes with meta-data giving a high-level description of the class rather than a small number of labeled examples. Therefore learn an embedding of the meta-data into a shared space to serve as the prototype for each class.

More concrete basic prototypical networks algorithm as follows. First, we get M-dimensional representation $c_k \in R^D \rightarrow R^M$ with learnable parameters ϕ. Each prototype is the mean vector of the embedded support points belonging to its class. Given a distance function $d : R^M \times R^M \rightarrow [0, +\infty]$, prototypical networks produce a distribution over classes for a query point x based on a softmax over distances to the prototypes in the embedding space. For a particular class of distance functions, known as regular Bregman divergences, the prototypical networks algorithm is equivalent to performing mixture density estimation on the support set with an exponential family density.

Prototypical networks can be seen as similar networks compare to a matching networks but there are differences. 1) Matching networks produce a weighted nearest neighbor classifier given the support set, while prototypical networks produce a linear classifier when squared Euclidean distance is used. 2) For the case of using multiple prototypes per class, prototypical networks differ from previously proposed methods given that prototypical networks could be trained with ordinary gradient descent methods.

From the experiment results, We can see the prototypical networks with some simply design decisions yield substantial improvements over recent approaches involving complicated architectural choices and meta learning.

2. TapNet: Neural Network Augmented with Task-Adaptive Projection for Few-Shot Learning

Few-shot learning promises to allow machines to carry out tasks that are previously unencountered, using only a small number of relevant examples. Unfortunately, despite immense interest and active research in recent years, reported results on few-shot learning still fall well below the levels that would be considered reliable in crucial real world settings

Previous works can be categorized by meta-learning perspective coupled with episodic training and metric based learning. For the meta-learning methods, the repetitive exposure to previously unseen tasks, each time with low samples, during this initial learning or meta-training stage seems to provide a viable option for preparing the learner for quick adaptation to new data. For the metric-based learners, methods that fall into this category incorporate non-parametric, distance-based learning, where embedding space is trained to minimize a relevant distance metric across episodes before stabilizing to perform actual few-shot classification. Relative to prior work, the unique characteristic of TapNet is in explicit task-dependent conditioning via linear projection of embedded features. Class-representing vectors in TapNet are not the outputs of an embedding fuction. Rather, the references in TapNet are a simple set of standalone vectors not directly coupled to input images, although they are updadted for each episode based on distances from the embedded query images projected in the classification space.

TapNet consists of three key elements : an embedding network f_θ, a set of per-class reference vectors Φ, and the task-dependent adaptive projection or mapping M of embedded features to a new classification space. Given the new support set, as well as f_θ and Φ learned through the last episode stage, a projection space M is first constructed such that the embedded feature vectors $f_\theta(x_k)$'s and the class reference vectors ϕ_k's with matching labels align closely when projected into M. The network f_θ and the reference set Φ are in turn updated according to softmax based on Euclidean distance between the mapped query image and reference vectors. The updated f_θ and Φ are passes to the next episode processing stage. In summary, the embedder f_θ and the per-class reference vectors Φ are learned across varying tasks while the projection space M is built specific to the given task, providing a quick task-dependent conditioning.

Finding the mapping fuction or projection space M is based on removing misalignment between the task-embedded features and the references. We wish to find a mapper M such that c_k and the matching reference vector ϕ_k are highly aligned in the mapped space. It turns out that a simple linear projection that does not require any learning offers an effective solution.

$$(1)\ \tilde{\phi}_k = \phi_k - \frac{1}{N_C - 1} \qquad (2)\ \sum_{l \neq k} \phi_l \epsilon_k = \frac{\tilde{\phi}_k}{\left|\left|\tilde{\phi}_k\right|\right|} - \frac{c_k}{||c_k||} \qquad (3) M = null_D([\epsilon_1; \cdots; \epsilon_{N_C}])$$

3. Experiment Results

(1) Reproduce Prototypical Nets on the miniImageNet benchmark

	5-way 1-shot	5-way 5-shot
Prototypical Nets	42.5	62.1

	5-way 1-shot	5-way 5-shot
Prototypical Nets	49.42	68.20

(Left : Reproduced, Right : Reported) For the reproduced results, I scaled the outputs of the euclidean distance [1]. Both of reproduced 1-shot and 5-shot results are showing worsen performance than reported results about 6%.

(2) Train/evaluate Prototypical Nets with a larger backbone

	miniImageNet		tieredImageNet	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
Resnet-12 backbone prototypical nets	32.34	48.07	34.21	50.10

Prototypical nets with resnet-12 backbone showed unsatisfactory performance both miniImageNet and tiered ImageNet. I assume that the cause of the worsen performance root from not adjusting the hyperparameters when training. Since resnet-12 model is bigger than the conv4 which is used in original paper, giving more epoch to training would enhance the performance. And inspecting the code is required.

(3) Implement TapNet, which is a variant of ProtoNets

	miniImageNet		tieredImageNet	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
TapNet	59.86	74.01	60.01	77.52

Reproduced results

	miniImageNet		tieredImageNet	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
TapNet	61.65	76.36	63.08	80.26

Reported results

Both of reproduced 1-shot and 5-shot results are showing worsen performance than reported results about 2% in miniImageNet and 3% in tieredImageNet.

[1] https://github.com/jakesnell/prototypical-networks/issues/5