Few shot learning by features adaptation with Graph Neural Networks

Armand Nicolicioiu
University Politehnica of Bucharest
armand.nicolicioiu@gmail.com

Andrei Nicolicioiu
Bitdefender
andrei.nicolicioiu@gmail.com

1. Introduction

Deep learning methods are known for requiring large amounts of data and high computing power. The few-shot learning regime tries to overcome these drawbacks by bringing innovations to the training method and optimization objective, often extending to the field of meta-learning. In classification tasks, there is an inherent structure between the classes, that is implicitly learned in current methods while in this work we model the relations between classes more explicitly. We study how to adapt embeddings of a new sample to the task context, given by centroids of the samples in a support set, by modeling the interactions between them using Graph Neural Networks with attention mechanisms.

2. Related Work

In few-shot setting, we have multiple classification tasks with N classes, from a base set, each having only K (1 to 5) annotated examples. Using them we meta-learn how to solve a new task with N novel, unseen classes. For a task, the annotated samples are denoted as the support set and are used to learn a classifier that is evaluated on a query set. The difficulty is to transfer knowledge between tasks and learn how to learn on the small support set in order to promote generalization and reduce overfitting.

Non-parametric methods [Vinyals et al., 2016] and [Snell et al., 2017] show good performance on the classification task by comparing the representation of a query sample with the representations of the samples in the support set using a metric. Optimization based methods often define an *inner loop* and an *out loop* as in [Finn et al., 2017]. In the inner loop an initial set of model parameters θ are fine-tunned on the support set while in the outer loop it makes predictions on the query set and uses the gradient of the loss on the query set to backpropagate through the inner loop optimization steps, to learn the original parameters θ. Further, [Zintgraf et al., 2018] shows that fine-tuning all the parameters in the inner-loop is prone to overfitting and propose adapting only a special set of *context parameters* to describe the current task.

The early work [Vinyals et al., 2016] shows the importance of forming a context of a task by using the support set. Follow-up work included the support set more tightly in the query embedding function using attention mechanisms [Mishra et al., 2018] and graph neural networks [Satorras and Estrach, 2018]. Using the context given by the support set [Gidaris and Komodakis, 2018] generates final classification weights, while [Shi et al., 2019] use known relations between the classes to produce better graph networks embeddings. Similar to us, other works modulate the features of a sample [Jiang et al., 2019, Kang et al., 2019] but they do not consider the relations between the support set and current sample.

3. Out Method

Given a new task, it is hard to optimize the whole model towards representations specific for every novel class. Given the few data points, this would lead towards non-robust features, as they could easily overfit the new samples. This could be avoided by learning only the final classification parameters W and methods like [Rusu et al., 2019, Gidaris and Komodakis, 2018] have been proposed to generate them given the labelled data in the support set.

Differently, we adapt the activations of the model for a new task by incorporating information received as input from the labelled data in the support set of the current task. We could modify the intermediate features $f_\theta^l(\mathbf{x})$, at different layers l of a model f_θ accordingly to the current task by modifying the activations statistics. It has been shown in the works of [Dumoulin et al., 2017, Huang and Belongie, 2017] that you could transfer the style between two images by matching the statistics of the features at multiple levels, and similarly we make the features of a new sample more distinctively for the current task.

We modulate the features of a sample $\mathbf{x}$ by using a scale γ and shift β:

$$\hat{f}_\theta(\mathbf{x}) = f_\theta(\mathbf{x}) \odot \gamma + \beta. \tag{1}$$

If γ and β are determined by optimizable parameters, we would arrive at a method similar to [Zintgraf et al., 2018] where the context is learned by optimization. Orthogonally to this approach, we estimate these parameters from the labelled samples in the support set. This way, we take advantage of the correlations between the current sample and the support set in an explicit way.

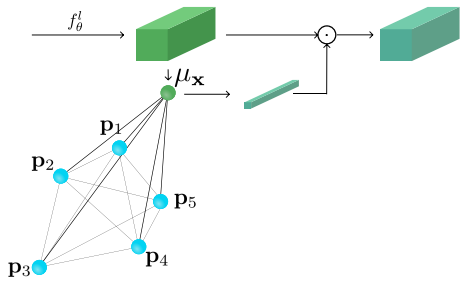

Figure 1. The GNN adapts the features $f_\theta(\mathbf{x})$ to the context given by the prototype statistics $\{\mathbf{p}_n\}$.

The current activations $f_\theta(\mathbf{x})$ are represented by their spatial mean, and the context is represented by the prototypes $\mathbf{p}_n$ formed by averaging the spatial mean for all the samples of each class S_n.

$$\mu_{\mathbf{x}} = \mu(f_\theta(\mathbf{x})) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} f_\theta(\mathbf{x})_{h,w} \tag{2}$$

$$\mathbf{p}_n = \frac{1}{K}\sum_{\mathbf{s}\in S_n} \mu(f_\theta(s)) \tag{3}$$

We need to put the statistics of the query sample in the context of the task, thus it is helpful to model the interactions between them and the statistics of the prototypes. Graph neural networks have a strong inductive bias towards relational reasoning [Battaglia et al., 2018] thus they are perfectly suited to achieve this, especially with an attention mechanism.

Support-Attention. We generate the γ and β parameters by sending messages between the prototypes of the support set and the current sample.

$$\gamma, \beta = \left[\mu_{\mathbf{x}} | \text{softmax}\left(\frac{(\mu_{\mathbf{x}} W_q)(PW_k)^T}{\sqrt{C}}\right)(PW_v)\right] W \tag{4}$$

where $\mu_{\mathbf{x}}, \gamma, \beta \in \mathbb{R}^{1\times C}$, $P \in \mathbb{R}^{N\times C}$, $W_q, W_k, W_v \in \mathbb{R}^{C\times C}$ and $W \in \mathbb{R}^{C\times 2C}$.

Support-Graph-Attention. We also experiment with a message passing scheme in four steps: send messages from support set to $\mu_{\mathbf{x}}$ with the previous mechanism, the resulting node sends back linear message to all elements of the support set then update the resulting nodes by self-attention, and finally generate γ and β with the previous mechanism. Different parameters are used in all the steps.

4. Experiments

To assess the performance of our model for few-shot learning, we experiment on MiniImagenet dataset and present some preliminary results. The proposed method to adapt the intermediate features of a model is flexible and could be used with any convolutional model and trained in multiple ways. Similar to other works, we experiment with two backbones having 4 convolutional layers, each with 32 or 128 channels. Similar to [Zintgraf et al., 2018], we train the model using an inner loop but optimize only a subset of the parameters. In the presented results, we trained only the final classification layer in the inner loop. Following [Snell et al., 2017], we also train the model by using the cosine similarity between final layer prototypes. We compare our method with our re-implementations of MAML, CAVIA and Proto-Nets and show in Table 1 that the adaptation is able to achieve improvements. We also compare with recent methods in Table 2 and show competitive results with methods using similar backbones.

Table 1. Results of our model and our re-implementations of MAML, CAVIA and ProtoNets on miniImagenet for 5-way 1-shot.

Model	ConvNet-4-32	ConvNet-4-128
MAML	47.41	48.29
Cavia	46.01	49.44
Proto-Nets	49.09	51.33
Our Inner Att	48.04	49.81
Our Inner Graph	46.72	49.2
Our Proto Graph	**50.23**	**52.38**

Table 2. Results on the test set of miniImageNet for 5-way 1-shot.

Model	Backbone	1-shot
Matching Nets [Vinyals et al., 2016]	ConvNet-4-32	43.56 ± 0.84
Proto Nets [Snell et al., 2017]	ConvNet-4-32	48.70 ± 1.84
MAML [Finn et al., 2017]	ConvNet-4-32	48.07 ± 1.75
Cavia [Zintgraf et al., 2018]	ConvNet-4-128	49.84 ± 0.68
GNN [Satorras and Estrach, 2018]	64-96-128-256	50.33 ± 0.36
LEO [Rusu et al., 2019]	WRN-28-10	61.76 ± 0.08
SNAIL [Mishra et al., 2018]	ResNet-12	55.71 ± 0.99
MetaOptNet [Lee et al., 2019]	ResNet-12	62.64 ± 0.61
Ours	ConvNet-4-32	50.23
Ours	ConvNet-4-128	52.38

5. Conclusion

We propose a method for learning in the few-shot setting by considering architectures that are constrained to model relations between the classes with graph neural networks. In preliminary experiments we show that modulating the features with graph models improves multiple backbones trained in different ways.

References

[Battaglia et al., 2018] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Ma-

linowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

[Dumoulin et al., 2017] Dumoulin, V., Shlens, J., and Kudlur, M. (2017). A learned representation for artistic style. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[Finn et al., 2017] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML 2017*, pages 1126–1135. JMLR. org.

[Gidaris and Komodakis, 2018] Gidaris, S. and Komodakis, N. (2018). Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375.

[Huang and Belongie, 2017] Huang, X. and Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510.

[Jiang et al., 2019] Jiang, X., Havaei, M., Varno, F., Chartrand, G., Chapados, N., and Matwin, S. (2019). Learning to learn with conditional class dependencies. In *International Conference on Learning Representations*.

[Kang et al., 2019] Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., and Darrell, T. (2019). Few-shot object detection via feature reweighting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8420–8429.

[Lee et al., 2019] Lee, K., Maji, S., Ravichandran, A., and Soatto, S. (2019). Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665.

[Mishra et al., 2018] Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. (2018). A simple neural attentive meta-learner. In *International Conference on Learning Representations*.

[Rusu et al., 2019] Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. (2019). Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*.

[Satorras and Estrach, 2018] Satorras, V. G. and Estrach, J. B. (2018). Few-shot learning with graph neural networks. In *International Conference on Learning Representations*.

[Shi et al., 2019] Shi, X., Salewski, L., Schiegg, M., Akata, Z., and Welling, M. (2019). Relational generalized few-shot learning. *arXiv preprint arXiv:1907.09557*.

[Snell et al., 2017] Snell, J., Swersky, K., and Zemel, R. (2017). Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087.

[Vinyals et al., 2016] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638.

[Zintgraf et al., 2018] Zintgraf, L. M., Shiarlis, K., Kurin, V., Hofmann, K., and Whiteson, S. (2018). Fast context adaptation via meta-learning. *arXiv preprint arXiv:1810.03642*.