A Survey on Few-Shot Learning

Xianda Xu

Yingcai Honors College

Problem Definition

Consider a supervised learning task T, Few-Shot Learning deals with a dataset $D = \{D^{train}, D^{test}\}$ which consists of training set $D^{\text{train}} = \left\{ \left(x^{(i)}, y^{(i)}\right) \right\}_{i=1}^{I}$ where I is small and test set $D^{\text{test}} = \{x^{\text{test}}\}$.

Usually, people consider the N-way-K-shot Classification task where D^{train} contains I=KN examples from N classes each with K examples.

Few-Shot Learning learns to discover $\hat{h}$, the optimal hypothesis from the input x to the output y.

To approximate $\hat{h}$, model determines a hypothesis space H. Algorithm is to search through H in order to find the θ that parameterizes the optimal h for D^{train}.

The performance is measured by a loss function defined over the prediction and the real output.

Typical Scenarios

- Learn for rare cases.

  Through Few-Shot Learning, one can learn suitable models for rare cases of limited supervised data.

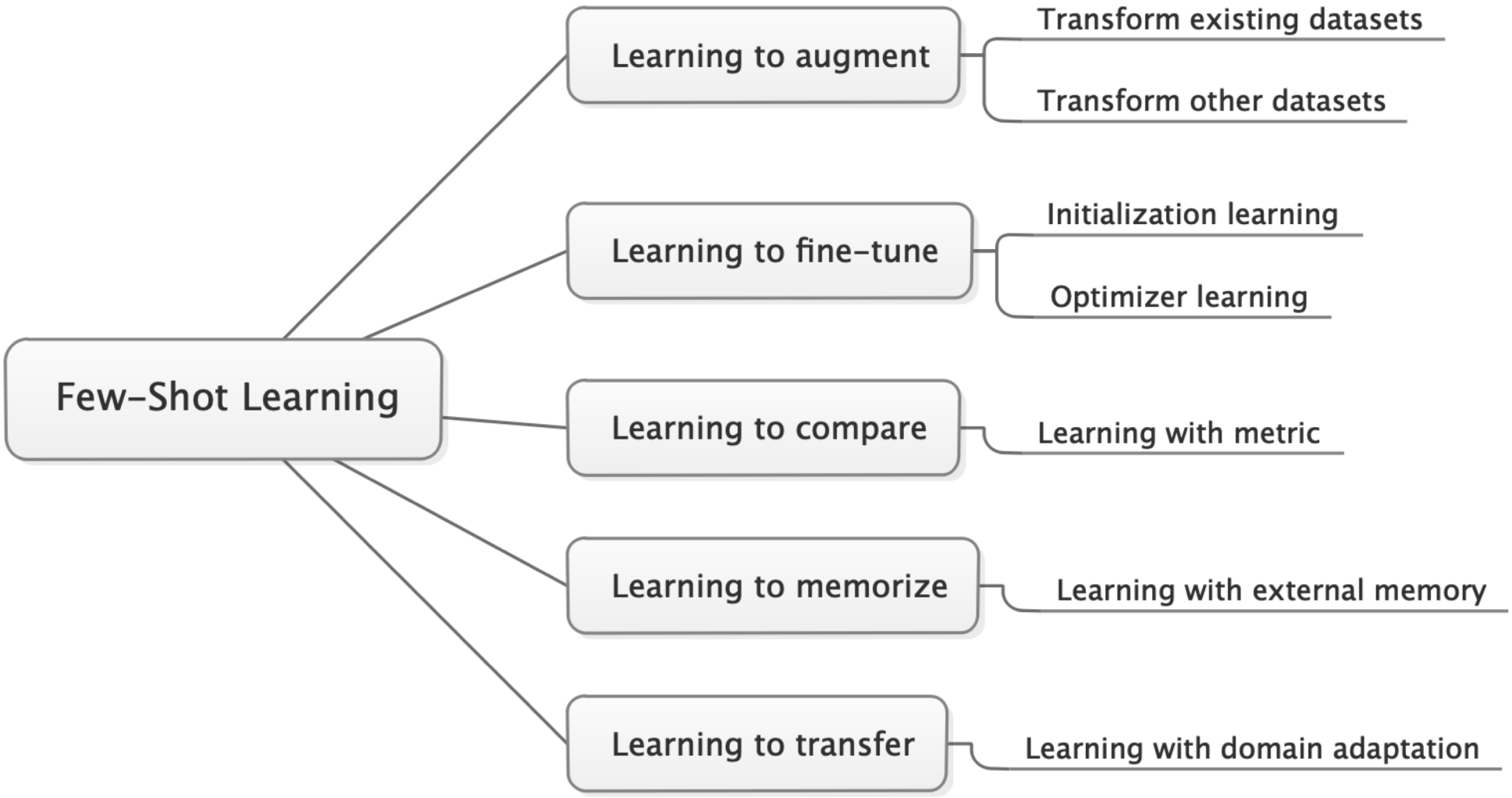
- Reduce data gathering effort.

  Few-Shot Learning can relieve the burden of collecting large-scale supervised data.

- Serve as a test bed for human-like learning.

  Few-Shot Learning is more like how children learn with limited resources.

Taxonomy of Previous Works



Learning to augment

Based on prior knowledge, it learns how to augment the training dataset.

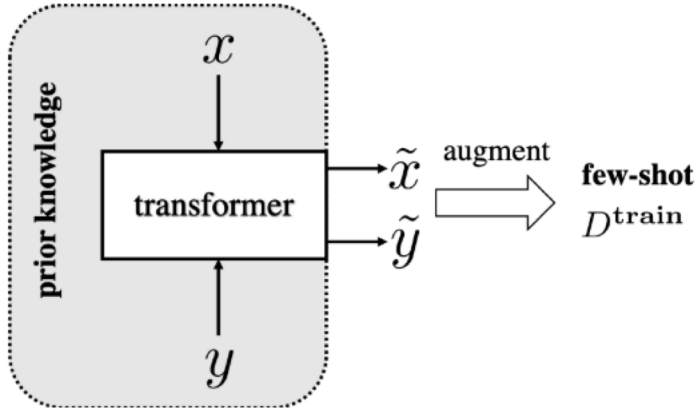
The idea of augmenting the training dataset is straightforward. With more samples (more supervised info), the data is sufficient to meet the sample complexity needed by subsequent machine learning models to obtain to more reliable hypothesis.

Methods can be classified into two categories.

One is to transform existing datasets, namely the training set.

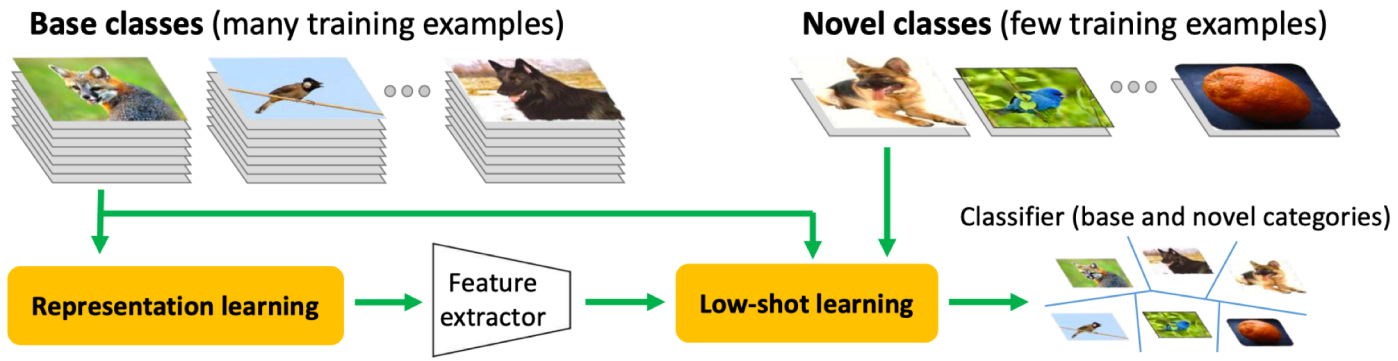
The other one is to transform other datasets, namely the external datasets.

In this section, we would discuss how these two ideas work and their advantages & disadvantages.



Learning to augment – Transform existing datasets

- One idea is to transform original examples in the training set using handcrafted rules including flipping, shearing, scaling, reflecting, cropping, rotating, etc.
- The other idea is to duplicate original examples into several samples which are then modified by learned transformation. For example, assuming all categories share general transformable variability across samples, a single transformation function is learned in Hariharan et al.'s work to transfer variation between sample pairs learned from other classes by analogy.



Hariharan, B., Girshick, R., & Ai, F. (n.d.). *Low-shot Visual Recognition by Shrinking and Hallucinating Features*. 3018–3027.

Learning to augment – Transform other datasets

- The idea is to augment the training set by aggregating samples pairs from other similar but larger data sets. For example, a data set of tigers is similar to another data set of cats. The assumption is that the underlying optimal hypothesis $\hat{h}$ applies to all classes, and the similarity between x of classes can be transferred to y of classes.

  Therefore, Gao et al. have designed a method based on generative adversarial network (GAN) to generate indiscriminate synthetic $\tilde{x}$ aggregated from data set of many samples.

Gao, H., Shou, Z., Zareian, A., Zhang, H., & Chang, S. (2018). *Low-shot Learning via Covariance-Preserving Adversarial Augmentation Networks*. (NeurIPS), 1–11.

Learning to augment – Summary

- To augment the training set, the idea of transforming the existing training set is very simple. But the constructed new samples will not be far away enough from the training set so generating samples in this way is restricted.

- On the other hand, the idea of transforming other data sets seems more effective since other data sets are in large-scale, providing tremendous samples of large variation for transformation. However, how to adapt those samples to be like the existing samples in the training set can be hard.

Learning to fine-tune

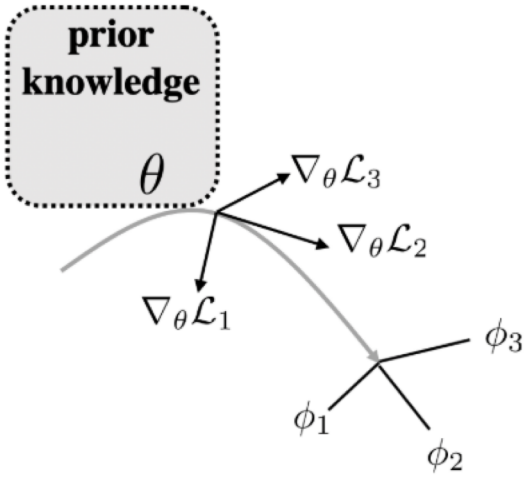
Based on prior knowledge, it learns how to alter the search for the optimal hypothesis.

Algorithm is used to search in the hypothesis space H for the parameter θ of the best hypothesis $\hat{h}$.

The idea of learning to fine-tune takes advantage of prior knowledge to alter the search for θ that parameterizes the $\hat{h}$ in the hypothesis space H. It does not restrict the shape of H.

There are two main methods.
One is to provide a good initialization to begin the search.
The other one is to provide a good optimizer to conduct the search.

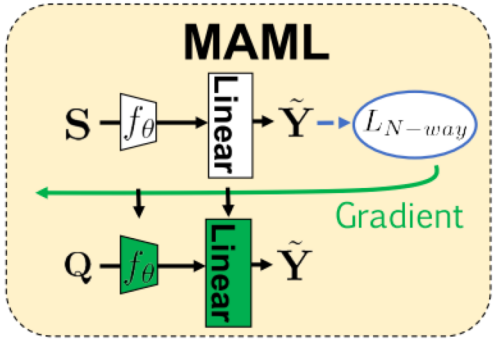
In this section, we would discuss how these two ideas work and their advantages & disadvantages.

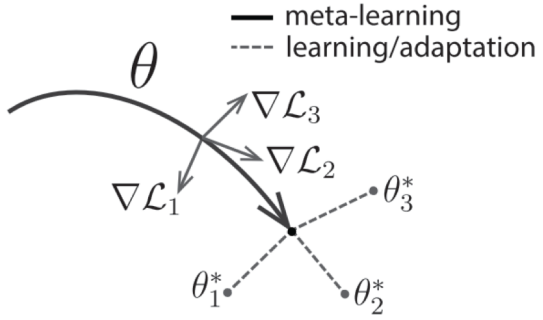


Learning to fine-tune – Initialization learning

- The idea is to learn a good initialization so that the classifiers for novel classes can be learned with a limited number labeled examples and a small number of gradient update steps. The assumption is that θ^0 captures general structures by learning from large-scale data, therefore it can be adapted using a few iterations to work well on D^{train}.

One typical work can be Model-Agnostic Meta-Learning (MAML) from Finn et al.



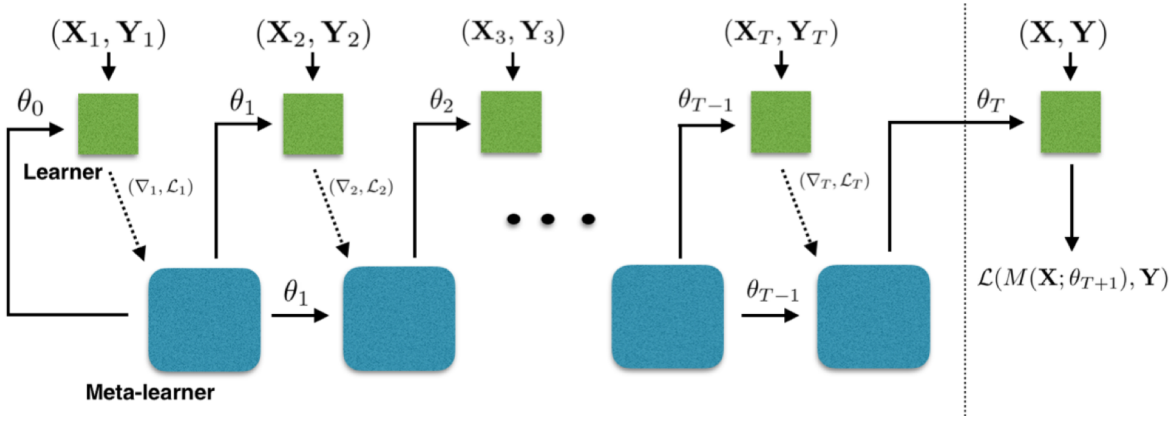


Finn, C., Abbeel, P., & Levine, S. (2017). *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks*.

Learning to fine-tune – Optimizer learning

- The idea is to learn a good optimizer so that the the classifiers for novel classes can be learned with advice on search speeds or search directions. The assumption is that by meta-learning from a set of tasks, the learned optimizer captures the common search strategy for this kind of tasks, therefore it can fine-tune the step size or find the best descent direction in Few-Shot Learning tasks automatically.

One typical work can be an LSTM-based meta-learner from Ravi et al which learns the exact optimization algorithm used to train another learner neural network classifier in the few-shot regime.



Ravi, S., & Larochelle, H. (2017). *O PTIMIZATION AS A M ODEL FOR F EW -S HOT L EARNING*. 1–11.

Learning to fine-tune – Summary

- These two methods, namely initialization learning and optimizer learning, are both meta-learned. The former meta-learns a good initialization θ^0 for the search of the optimal hypothesis $\hat{h}$ .The latter meta-learns a good optimizer that guides the search of the optimal hypothesis $\hat{h}$.

However, both methods are affected by previous tasks without checking the relatedness of the consecutive tasks. Once the tasks are negatively related, negative transfer incurs easily.

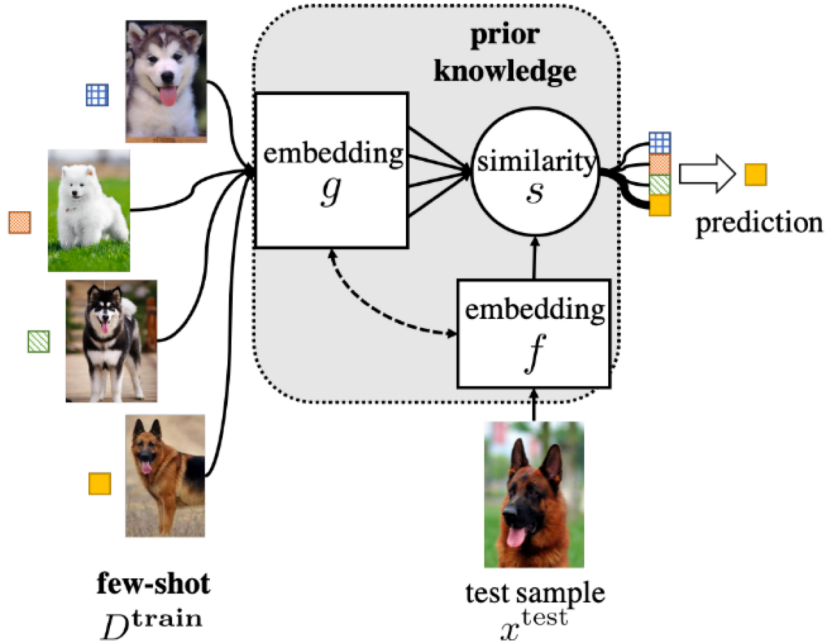
Learning to compare

Based on prior knowledge, it learns how to embed samples into a embedding space where the similar and dissimilar pairs can be easily identified using a metric so that the hypothesis space is constrained.

The idea contains three key components. The embedding function g embeds the supporting samples while the embedding function f embeds the querying samples. The similarity function s measures their distances.

The embedding function g and f are normally the same. They are mainly learnt by prior knowledge and can additionally use the training set to bring in task-specific information. The similarity function can either manually chosen or learnt.

In this section, we would discuss three typical models, talking about their characteristics and differences among them.



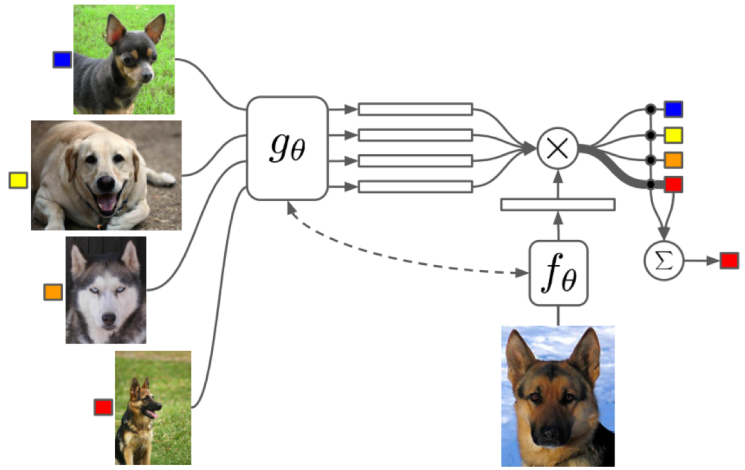
Learning to compare – Matching Networks

- Matching Networks assign x^{test} to the most similar $x^{(i)}$ in the samples, where x^{test} and $x^{(i)}$ are embedded differently by f and g.

Specially, f is conditioned on D^{train}, and g aggregates information from all samples in D^{train} by a bi-directional LSTM (biLSTM).

The mathematics expression of the network is: $\hat{y} = \sum_{i=1}^{k} a\left(\hat{x}, x_i\right) y_i$

The metric used in the network is Cosine Distance.



Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 3637–3645.

Learning to compare – Prototypical Networks

- Prototypical Networks perform only one comparison between x^{test} and prototype of each class in D^{train} .

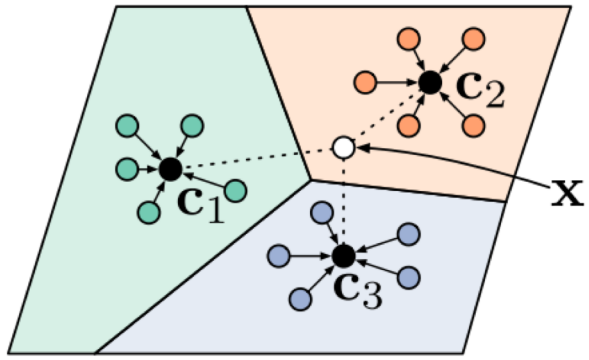
The class n's prototype is defined as the mean of embeddings of that class: $c_n = \frac{1}{K} \sum_{k=1}^{K} g(x^{(i)})$

where $x^{(i)}$ is one of the K samples of the class n in D^{train}.

Notice that it embeds both x^{test} and $x^{(i)}$ using the same neural network.

The metric used in the network is Euclidean Distance.

It is different from Matching Networks in the Few-Shot case while equivalent in the One-Shot case.

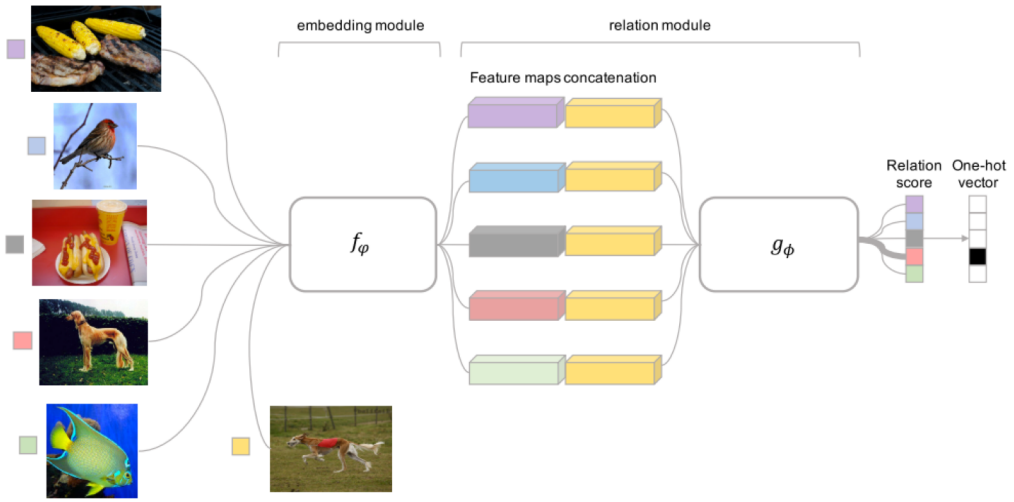


Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, *2017-Decem*, 4078–4088.

Learning to compare – Relation Networks

- Relation Networks further embed the embedding of x^{test} and each c_n calculated from D^{train} jointly, which is then directly mapped to the similarity score.

It can be seen as both learning a deep embedding and learning a deep non-linear metric (similarity function).



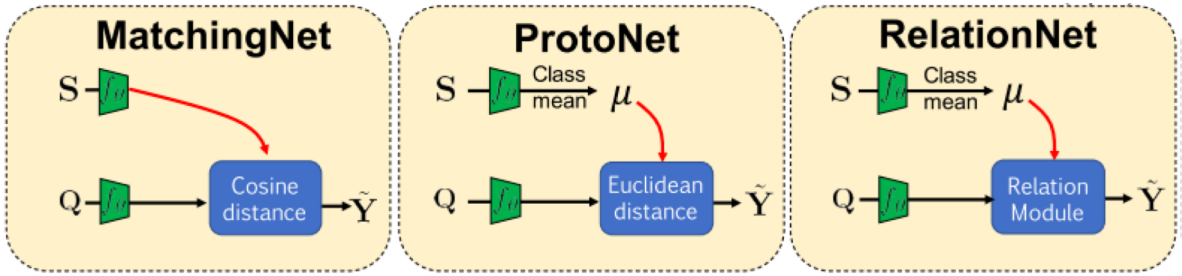
Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., & Hospedales, T. M. (2018). Learning to Compare: Relation Network for Few-Shot Learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1199–1208. https://doi.org/10.1109/CVPR.2018.00131

Learning to compare – Summary

Chen et al. have conducted a comparative analysis of these methods and got some interesting findings.

- Using a deep backbone shrinks the performance gap between these models in the setting of limited domain differences between base and novel classes.
- These models fail to address the classification problem where base and novel classes are sampled from different domains (domain shifts).

Their work draws our attention to the challenge of domain differences and domain shifts in the context of Few-Shot Learning.



Chen, W. Y., Wang, Y. C. F., Liu, Y. C., Kira, Z., & Huang, J. Bin. (2019). A closer look at few-shot classification. *7th International Conference on Learning Representations, ICLR 2019*, (2018), 1–17.

Learning to memorize

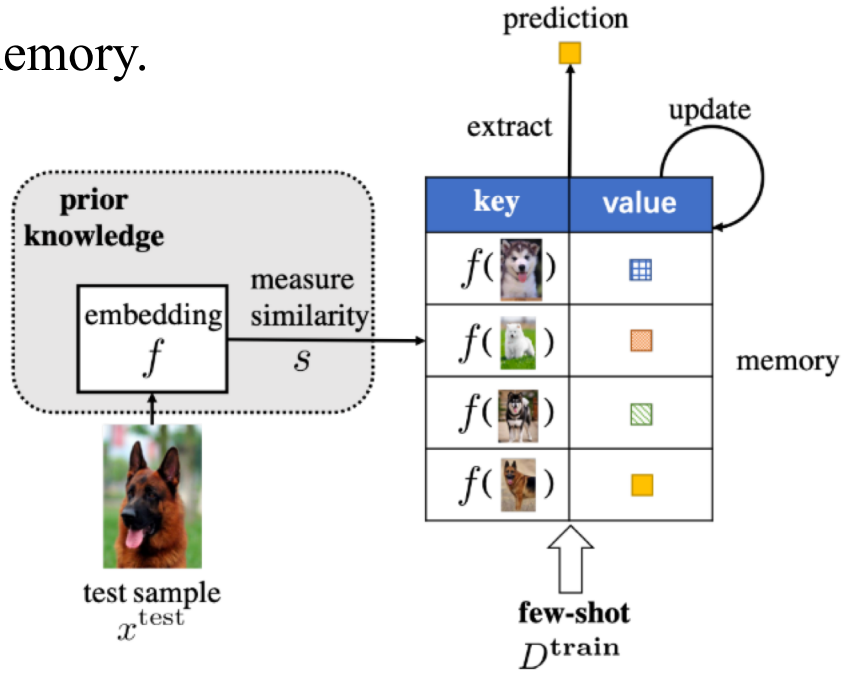
Based on prior knowledge, it learns how to store the reinterpretation of samples in its external memory so that the hypothesis space is constrained.

The idea is based on the fact that some networks like NTM or RNN allow shot-term memorization.

They iterate over examples of a given problem and accumulates the knowledge required to solve that problem in its hidden activations, or external memory.

When new examples come, classification can be done by comparing these new examples to historic information stored in the memory.

In this section, we would show how the reinterpretation of existing data can be stored and used in memory networks.

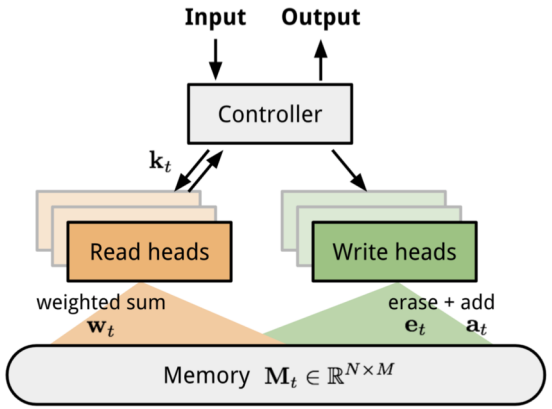


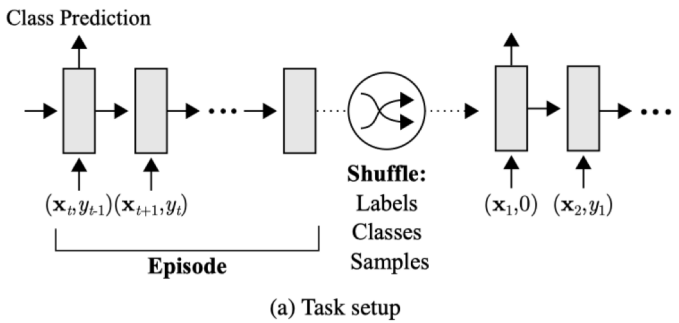
Learning to memorize

Santoro et al adapt Neural Turing Machine (NTM) to memorize learnt knowledge.

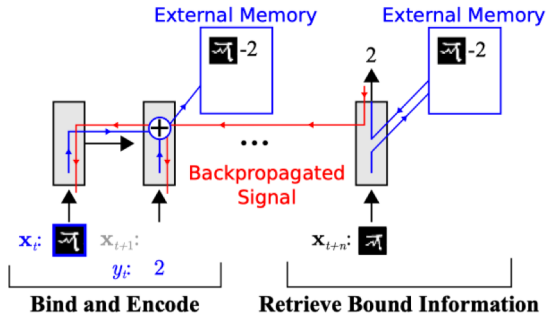
The reading strategy is similar to NTM: $\mathbf{r}_i = \sum_{i=1}^{N} w_t^r(i)\mathbf{M}_t(i),\ \text{where } w_t^r(i) = \text{softmax}\left(\frac{\mathbf{k}_t \cdot \mathbf{M}_t(i)}{\|\mathbf{k}_t\| \cdot \|\mathbf{M}_t(i)\|}\right)$

The writing strategy adopts Least Recently Used Access (LRUA).





(a) Task setup



(b) Network strategy

Santoro, A., Botvinick, M., Lillicrap, T., Deepmind, G., & Com, W. G. (2016). *Meta-Learning with Memory-Augmented Neural Networks*. *48*.

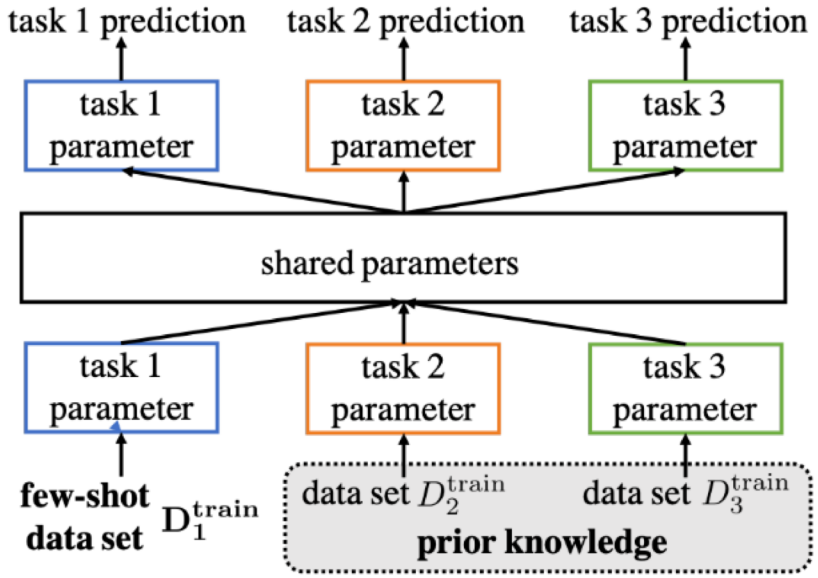
Learning to transfer

It learns how to apply prior knowledge in tasks with different domains for Few-shot Learning problems and reduce the domain shifts.

We can define Few-shot Learning tasks as target tasks while other tasks as source tasks.

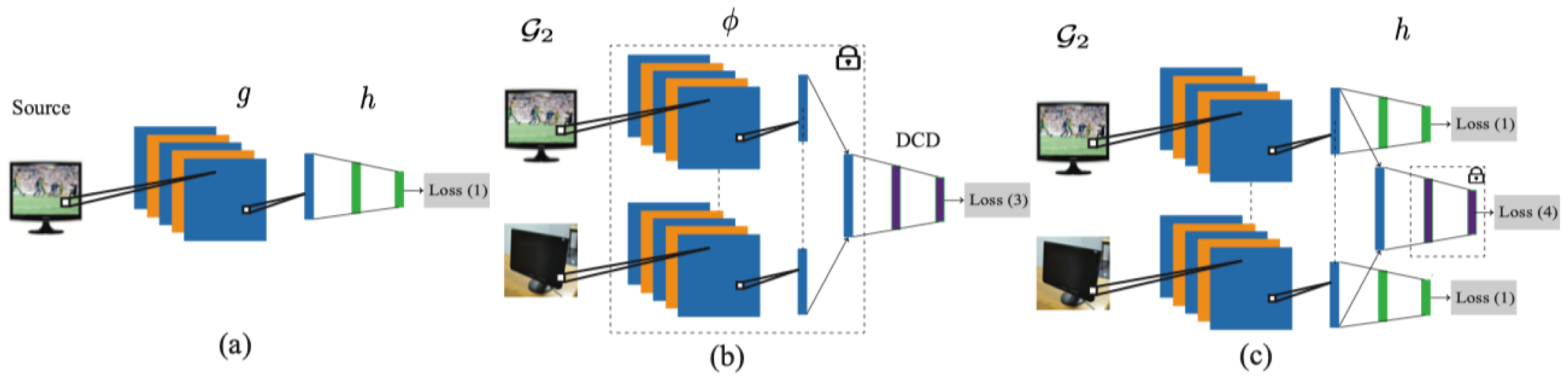
The idea is based on the assumption that the target tasks and the source tasks have similar or overlapping hypothesis space. So, it explicitly shares parameters among tasks to promote overlapping hypothesis space and additionally learn task-specific parameters for each task to account for tasks specialties.

In this section, we would explain how to transfer knowledge from other domains to the domain of Few-shot Learning problem.



Learning to transfer

Method in Motiian et al's work first pre-trains a variational auto-encoder from the source tasks in the source domain and clones it for the target tasks. Then it shares some layers to capture generic information and lets both tasks to have some task-specific layers. The target tasks can only update their task-specific layers, while the source tasks can update both shared and their specific layers.



Motiian, S., Jones, Q., Iranmanesh, S. M., & Doretto, G. (2017). Few-shot adversarial domain adaptation. *Advances in Neural Information Processing Systems*, *2017-December*(Nips), 6671–6681.

Evaluation on Prototypical Networks

Dataset: Omniglot

Training Set: 82240 images

Validation Set: 13760 images

Test Set: 33840 images

Model: Prototypical Networks

Network: 4 layers (1-64-64-64-64)

Metric: Euclidean Distance

Training Details:

Optimizer: Adam

Initialized Learning Rate: 0.001

Scheduler: StepLR (20, 0.5)

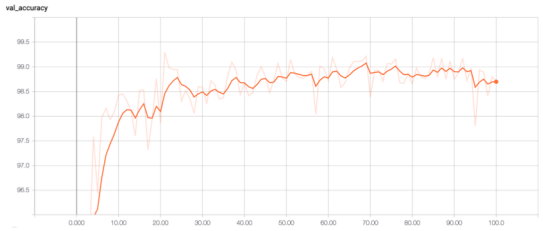
Evaluation Details:

Within 100 epochs, Prototypical Networks is trained and validated.

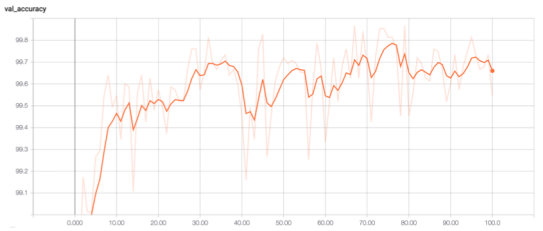
Then, the best model in validation is tested with 10 epochs.

Evaluation on Prototypical Networks

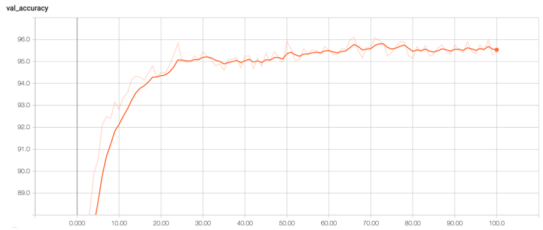
Validating



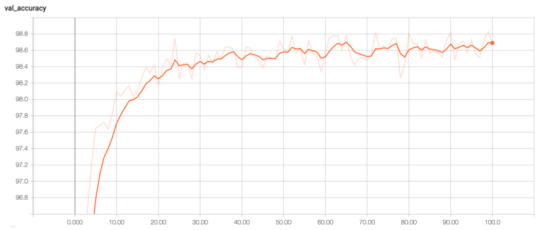
5-way 1-shot



5-way 5-shot



20-way 1-shot



20-way 5-shot

Testing

	5-way 1-shot(Acc)	5-way 5-shot(Acc)	20-way 1-shot(Acc)	20-way 5-shot(Acc)
Paper	98.8%	99.7%	96.0%	98.9%
Ours	98.4%	99.6%	94.7%	98.6%

Future Work

- Learning to augment

  Alfassy et al's work has given us an inspiration to augment data using label set manipulation. So, whether we could excavate more information from multi-label data is a possible direction.

- Learning to compare

  Settings in the current model like the hypothesis space H still rely on human design. So, one possible direction is to extend neural architecture search to this method. Another possible direction is to apply metric in Natural Language Processing tasks to Computer Vision tasks to see special characteristics like hierarchy in Nickel et al's work.

Alfassy, A., Karlinsky, L., Aides, A., Shtok, J., Harary, S., Feris, R., ··· Bronstein, A. M. (2019). *LaSO: Label-Set Operations networks for multi-label few-shot learning*. 6548–6557.

Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems*, *2017-December*(Nips), 6339–6348.

Future Work

- About datasets

  Available training data in Few-Shot Learning tasks is small-scale. It makes the training dataset more vulnerable to characteristics including imbalanced distribution, inter-class variance, intra-class variance, fineness, etc. So, one possible direction is to explore algorithms dealing with Few-Shot Learning tasks under more challenging datasets.

Thank You ！

Contact: bryce_xu@outlook.com