Transductive Mutual Information Encoder Network for Few Shot Learning

Wei Huang* and Caijun Ren
Department of Computer Science and Technology, University of Science and Technology of China, Hefei, China
*Corresponding Author’s Email: hwei1023@gmail.com

Abstract. Deep learning has made breakthroughs in recent decades and has been widely used in many domains. However, most of those methods heavily rely on large labeled datasets, which results in poor performance when provided with limited labeled data. Few-shot learning (FSL), which aims at learning a novel task with limited samples, has attracted a lot of research recently. The previous metric-based methods ignore the internal bias between the training and testing data sets since the categories of the testing dataset are completely different from the training set. Transfer learning methods also suffer from few labeled data and tends to be overfitting in this situation. This paper proposes Transductive Mutual Information Encoder Network (TMIN) for few-shot learning problems. TMIN typically trains a convolutional neural network with a mutual information maximization module in an unsupervised manner. The trained network maps images to a high dimensional embedding space. Then the embeddings are exploited to measure the similarity between samples by a distance metric. Experiments indicate that the proposed model achieves competitive performance compared with the counterparts.

1. Introduction
With the development of big data technology, deep learning has made great progress recently in many domains such as natural language processing [1, 2], computer vision [3, 4], and time-series data [5, 6]. However, most of the deep learning-based methods rely on large-scale datasets, while humans can learn from few samples. Few-shot learning, which aims at learning from limited labeled data like humans, remains a key challenge in machine learning and has attracted numerous studies.

Many works propose to utilize meta-learning, which means learning to learn, to handle the aforementioned problem. Recently, metric-based meta-learning methods make great progress in solving the FSL problem. These methods firstly train an embedding model that maps samples to a high dimensional embedding space. The model aims at clustering samples with the same label together while separating samples with different labels far away in this space. Then a classifier with a distance metric is applied to recognize new samples in this learned embedding space. The key idea is to find a discriminative embedding space and a proper distance metric as in [7, 8].

In the meta-learning settings, the testing set contains different classes from the training set, which indicates that there exists an internal bias between the distributions of these two sets. This may result in disorder when mapping the testing samples to the high dimensional embedding space learned on the training set, which, unfortunately, is neglected by those methods mentioned above. On the other hand, for the FSL problem, each task only contains a few labeled data. This greatly increases the difficulty of generating robust embedding representations for samples with limited labeled data. Optimization-based methods as well as transfer learning may result in overfitting.
To alleviate the problem caused by the insufficient samples and the inconsistent distribution between the training and testing datasets, we propose a novel method denoted as Transductive Mutual Information Encoder Network (TMIN). TMIN trains an embedding model that maximizes the mutual information between the samples in an FSL task. As there are only a few labeled samples in a task, each task could be roughly viewed as an unsupervised learning problem, therefore TMIN is trained unsupervised. Finally, a distance metric is applied to classify the unlabelled data in the embedding space.

2. Related Works

Many approaches propose to use meta-learning to deal with the FSL scenarios. With the strategy of meta-learning, models learn transferable knowledge from auxiliary data or tasks via episodic training. Each task contains two sets, denoted as support set $S$ and query set $Q$, which will be detailed later. These models utilize the learned prior knowledge to process the few-shot classification for new classes with a few examples.

2.1. Metric Based Methods

Metric-based is one of the efficient meta-learning strategies. Figure 1 illustrates the framework of metric-based methods. Matching networks [9] mapped samples into an embedding space and an attention mechanism was used on this learned embedding space to predict a class for the unlabelled data. Then cosine distance was used to calculate the similarity score between the embeddings of the samples in $S$ and $Q$ for classification. Prototypical networks proposed in [10] exploited the mean embedding of samples from the same class as the representation of the class. And Euclidean distance was used for classification between the prototype of each class and the embedding of query sample. Relation network [11] comprised two parts, relation module, and embedding module. The embedding module was similar to the aforementioned metric-based algorithms, while the relation module used a learnable distance metric to measure the similarity of the embeddings by obtaining relation scores. Although the targets of the ground-truth are the values of 0 or 1, the relation score was predicted as a regression problem in the Relation network.

![Figure 1. The framework of Metric Based Methods.](image)

However, the performance of metric-based techniques may suffer when tasks at the test stage become more distant from the tasks sampled at the training stage, as these methods are unable to capture the information of new tasks by the network.

2.2. Optimization Based Methods

Optimization-based models provide a different perspective from the metric-based methods. These methods learn a good initialization of the model parameters, which could be fast adapted to new tasks, by converting meta-learning into a bi-level optimization problem. The inner-level learns task-specific information fastly, while the outer-level that aims at optimizing the cross tasks performance updates
slowly. The well-known Model agnostic meta-learning (MAML) \cite{12} is a framework that aims at finding more transferable representations with sensitive parameters. Some methods also attempted to find a good set of initialization parameters $\theta$ by exploring the potential of meta-learning algorithms based on the first-order gradient information. Latent Embedding Optimization (LEO) in \cite{13} learned a lower-dimensional latent embedding space using an encoder and relation net. And the decoder network could generate task-specific mean and variance from latent code for each class to sample initialization weights $\theta$.

3. Mutual Information Maximization
An important measurement of similarity between random variables is Mutual Information (MI), which captures general and statistical dependencies between the random variables. Mutual Information Neural Estimator (MINE) \cite{14} used neural networks to estimate tight variational lower bounds of MI, which assumed a sufficient supply of samples to prevent the networks from overfitting. Recently, Deep InfoMax (DIM) \cite{15} provided new opinions for learning unsupervised representations, which trained efficient representations using unsupervised learning by maximizing the mutual information between the global and local features in a deep neural network.

![Figure 2. Mutual Information Maximization Module](image)

4. Proposed Method
In this section, the details of proposed Transductive Mutual Information Encoder Networks (TMIN) for few-shot learning problems will be introduced.

4.1. Problem Setting
In this work, we use the episodic formulation of \cite{9} which efficiently trains a meta-learner for few-shot classification tasks. The whole process is separated into two stages, i.e. meta-training stage and the meta-testing stage. In the meta training stage, the dataset $C_{train}$ contains a relatively large labeled dataset with a set of classes, the objective of this stage is to train a classifier model, which could be well adapted to other tasks, while each task only contains a few labeled examples of novel classes from testing dataset $C_{test}$ in the testing stage. The datasets $C_{train}$ and $C_{test}$ have completely disjoint classes, as the purpose is to evaluate whether the model can obtain accurate prediction results with samples of novel classes.

Specifically, for the N-way K-shot FSL problem, each task is composed of a support set $S$ and a query set $Q$. The label space of $S$ and $Q$ is the same, but they have disjoint data samples. The number of classes in $S$ is $N$, and each class contains $K$ samples denoted as

$$S = \{x_i, y_i | i = 1, 2, ... N \times K\}$$ (1)
where $x_i$ is an input sample and $y_i$ is the corresponding label, and as we mainly focus on image classification tasks $x_i$ will represent an image in the rest of this paper. The query set $Q$ contains other $T$ samples with the same class labels in $S$. The model is trained or fine-tuned on $S$ to minimize the loss of its predictions on $Q$, i.e., samples will be classified into the class that has the highest probability between the similarity scores, which could also be done by a linear classifier such as MLP or MPCVM[16].

4.2. Network Architecture
The framework of our proposed network mainly comprises two modules, the encoder network, and the DIM module, and they will be introduced separately as follows.

4.2.1. Encoder network. The encoder network $E$ aims at extracting the latent features of each sample. For image classification, $E$ comprises several convolutional blocks and maps each image $x_i$ to a high dimensional vector $v$. The output of one intermediate layer is called feature map $M$, which is used in the DIM module. And the final output $g$ of the encoder is denoted as the global feature vector, which is used both in the DIM module and classifier module. Softmax cross-entropy is calculated as the loss function for separating samples from different classes,

$$L_E = - \sum \log \left( \frac{e^{-d(v_q c_k)}}{e^{-d(v_q c_k)} + e^{-d(v_q \hat{c}_k)}} \right)$$  \hspace{1cm} (2)

Where $d(.)$ is a distance metric like Euclidean distance, $v_q$ is a testing sample from class $k$, $c_k$ and $\hat{c}_k$ represent the center of class $c$ and the center of rest class except $c$, respectively.

4.2.2. DIM module. The DIM module is illustrated in Figure 2. It maximizes the mutual information between the high-level feature vector and the local feature map of each image to generate a better representation for classification, it works as follows: firstly the encoder $E$ encodes the input sample $x_i$ to a feature map $M$ with parameter $\theta$, for image samples, the shape of feature map $M$ is $h \times w \times c$, where $h$ and $w$ represent the height and width of the spatial structure of the feature map separately. $c$ is the number of channels of output. And each cell, $M_{ij} \in \mathbb{R}^c$ is a high dimensional feature vector. Next, the encoder continues to encode this local feature map into a high-level global feature vector $g$. Then the discriminator $D_o$ maximizing the average estimated MI on global/local pairs with parameter $\omega$. During training, images of batch size $B$ are fed into the network and get the batch global feature vectors $\hat{G}$, and the corresponding feature maps $\hat{M}$. Then the order of $\hat{M}$ is shuffled to generate negative global/local pairs, i.e., the global feature vector $G_1$ and the feature map $M_1$ from the same sample $x_1$ are treated as "Real" by $D_o$, while the feature vector $G_1$ of sample $x_1$ and the feature map $M_2$ from sample $x_2$ are treated as "Fake". The loss function for this module is

$$L_d = - \sum_{i \neq j} \log D_o(g_i, M_i) + \log \left( 1 - D_o(g_i, M_j) \right)$$ \hspace{1cm} (3)

These two modules are trained simultaneously using loss functions in Equation 2 and Equation 3.

5. Experiment Results
In this section, we experiment with two widely used datasets, and then the results are illustrated and analyzed.

5.1. Datasets

5.1.1. MiniImagenet. MiniImagenet is a mini variant of the large-scale ImageNet dataset for image classification. The MiniImagenet dataset contains a total of 60000 colored images which are resized to $84 \times 84$. There are 100 classes in this dataset, with 600 examples per class. This dataset is split into three parts according to class labels: 64 classes for training, 16 classes for validation, and the rest 20 classes for testing.
5.1.2. TieredImagenet. TieredImagenet divides classes into more categories corresponding to higher-level nodes in the ImageNet with a hierarchical structure. There is a total of 34 categories, each category contains 10 to 30 classes. 20 categories are split for training, 6 categories for validation, and the rest 8 for testing. All images in this dataset are also resized to 84 × 84.

5.2. Results and Analysis
These methods are evaluated on two scenarios on both datasets, 5-way 1-shot, and 5-way 5-shot, as they are the most commonly used scenarios in FSL research. Results in tables are the average results of 5 experiments with different random seeds. Results on MiniImageNet and TieredImageNet are illustrated in Table 1 and Table 2 respectively.

| Models     | 5way 1shot | 5way 5shot |
|------------|------------|------------|
| MatchingNet | 43.56      | 55.31      |
| ProtoNet    | 48.70      | 63.10      |
| RelationNet | 50.44      | 65.32      |
| PFA         | 54.53      | 67.87      |
| TADAM       | 58.50      | 76.70      |
| LEO         | 60.06      | 75.72      |
| TMIN(ours)  | 55.13      | 67.17      |

Table 1. Results On MiniImageNet (accuracy)

| backbone     | 5way 1shot | 5way 5shot |
|--------------|------------|------------|
| MatchingNets | ResNet12   | 68.50      |
| ProtoNet     | Convnet    | 48.58      |
| RelationNet  | Convnet    | 54.48      |
| LEO          | WRN-28-10  | 66.33      |
| MetaOptNet   | ResNet12   | 65.99      |
| TMIN(ours)   | ResNet12   | 67.17      |

Table 2. Results On TieredImageNet (accuracy)

We follow these previous works and conduct experiments with the same network backbones on both datasets. Convnet comprises four convolutional blocks while ResNet consists of 4 residual blocks. Each block contains a convolutional layer with kernel size 3×3, a batch normalization layer, and a nonlinearity layer. Images are resized to 80×80 as the input of both networks.

As shown in Table 1, it is obvious that more complex architecture and more support data both improve the model’s performance. The results on 5 way 5 shot setting are typically more accurate than results on 5 way 1 shot, since the former one takes more samples in training tasks, and more data will be closer to the realistic data distribution and generate more robust embedding. On the other hand, the models using Resnet are better than models using Convnet, as ResNet takes more parameters and can capture more latent information. The ResNet improves 5 points of accuracy on 5 way 1 shot setting, while it improves about 10 points in 5 way 5 shot setting. And note that the same backbone both improve more than 10 points switching from 1 shot to 5 shot. And our proposed model TMIN is better than other models in many scenarios except 5 way 5 shot using convnet. The reason for its inefficiency may be that the parameter of Convnet in this setting is small to compete with the DIM module, as DIM tries to generate unique representations for each sample while the encoder tries to classifier them.
On tieredImagenet, these models also perform better in 5 shot settings than 1 shot settings. Our model can achieve competitive results with other models. And outperform others in 5 way 5 shot setting. Though TMIN is not as good as MatchingNet in 5 way 1 shot setting, it can achieve better performance when giving more samples on support set and query set, as MatchingNet may overfitting when support set is small and it’s hard to transfer to query set. As TMIN is trained in an unsupervised manner, its performance is proportional to the number of samples.

6. Conclusion
In this paper, we propose the Transductive Mutual Information Encoder Network for few-shot learning, which is an efficient and simple framework of meta-learning. For FSL tasks, it learns the representations for the samples in both the support set and query set in an unsupervised manner. Then the similarity scores are measured by a distance metric. The unlabelled samples are tagged into one category by the labeled samples with the highest similarity score. And experiments indicate that our method could achieve competitive results compared with other methods. However, the current methods are almost limited to the N-way K-shot problem, there are few works to learn dynamic tasks, i.e. the number of classes or the number of samples per class are different in each task. And we leave this for future work to transfer meta-learning to different types of tasks.

7. References
[1] Lyu S and Liu J 2020 Hybrid Framework of Convolution and Recurrent Neural Networks for Text Classification Proc. of IEEE Int. Conf. on Knowledge Graph pp 313-320.
[2] Lyu S, Cheng J, Wu X, Cui L, Chen H and Miao C 2020 Auxiliary Learning for Relation Extraction IEEE Transactions on Emerging Topics in Computational Intelligence.
[3] Zhou X, Chen H and Li J 2018 An Automatic GPR B-scan Image Interpreting Model IEEE Transactions on Geoscience and Remote Sensing.
[4] Yao Y, Liu Y, Liu Z and Chen H 2018 Human Activity Recognition with Posture Tendency Descriptors on Action Snippets IEEE Transactions on Big Data.
[5] Chen H, Tino P, Rodan A and Yao X 2014 Learning in the Model Space for Cognitive Fault Diagnosis IEEE Transactions on Neural Networks and Learning Systems vol 25 pp 24-136.
[6] Chen H, Tino P and Yao X 2014 Cognitive Fault Diagnosis in Tennessee Eastman Process using Learning in the Model Space Computers & Chemical Engineering.
[7] Chen H, Tang F, Tino P, Cohn A and Yao X 2015 Model Metric Co-learning for Time Series Classification Proc. of the 25th Int. Joint Conf. on Artificial Intelligence.
[8] Chen H, Tang F, Tino P and Yao X 2013 Model-based kernel for efficient time series analysis Proc. of the 19th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining.
[9] Oriol V, Charles B, Timothy L, Koray K and Daan W 2016 Matching networks for one shot learning Advances in Neural Information Processing Systems pp 3630–3638.
[10] Snell J, Swersky K and Zemel R 2017 Prototypical networks for few-shot learning Advances in Neural Information Processing Systems.
[11] Sung F, Yang Y, Zhang L, Xiang T and Hospedales T 2018 Learning to compare: Relation network for few-shot learning IEEE/CVF Conf. on Computer Vision and Pattern Recognition.
[12] Finn C, Abbeel P and Levine S 2017 Model-agnostic meta-learning for fast adaptation of deep networks Proc. of the 34th Int. Conf. on Machine Learning vol 70 pp 1126–1135.
[13] Rusu A, Rao D, Szygnowski J, Vinyals O, Pascanu R, Osindero S and Hadsell R 2019 Meta-learning with latent embedding optimization 7th Int. Conf. on Learning Representations.
[14] Belghazi I, Rajeswar S, Baratin A, Hjelm R and Courville A 2018 Mine: Mutual information neural estimation Int. Conf. on Machine Learning pp 530–539.
[15] Hjelm D, Fedorov A and Lavoie-Marchildon S et.al. 2019 Learning deep representations by mutual information estimation and maximization Int. Conf. on Learning Representations.
[16] Lyu S, Tian X, Li Y, Jiang B and Chen H 2019 Multi-class Probabilistic Classification Vector Machines IEEE Transactions on Neural Networks and Learning Systems vol 31 pp 3906-3919.