Finding Significant Features for Few-Shot Learning using Dimensionality Reduction Techniques

Mauricio Mendez Ruiz[1], Ivan Garcia[2], Andres Mendez-Vazquez[2], Gilberto Ochoa-Ruiz[1]

[1]Tecnologico de Monterrey, School of Engineering and Sciences, Mexico
[2]CINVESTAV-IPN, Unidad Guadalajara, Mexico

a00812794@itesm.mx, gilberto.ochoa@tec.mx, andres@mendez@cinvestav.mx

Abstract

Few-shot learning is a fairly new technique that specialize in problems where we have little amount of data. The goal of this method is to classify categories that hasn't been seen before with just a handful of samples. Recent approaches, such as metric learning, adopt the meta-learning setting in which we have episodic tasks conformed by support (training) data and query (test) data. Metric learning methods has demonstrated that simple models can achieve good performance, by learning a similarity function to compare the support and the query data. However, the feature space learned by the metric learning may not exploit the information given by a specific few-shot task. In this work, we explore the use of dimension reduction techniques as a way to find task-significant features. We measure the performance of the reduced features by giving a score based on the intra-class and inter-class distance, and select the method in which instances of different classes are distant and instances of the same class are close. This module helps to improve the accuracy performance by allowing the similarity function, given by the metric learning method, to have more discriminative features for the classification.

1. Introduction

In recent years, we have witnessed the great progress of successful deep learning models and architectures [7], and the application in real-world problems has been increasing in the last decade achieving great performance in a broad spectrum of research fields such as vision, language, speech, games, medicine, etc. Despite the advances of deep learning models in important domains such as vision and language, the standard supervised learning does not offer a satisfactory solution for learning new concepts from little data, given that training current deep learning models with low amount of data highly increase the risk of overfitting and fail to produce a good generalization. There are many problem domains, like health and medical problems, where obtaining labeled data can be very difficult or the amount of work required to obtain the ground truth representations is very large. Facing the problem of scarcity of data is something that humans are capable of, possessing the ability to identify an object after seeing it just a few times. Few-shot learning methods has been proposed [10, 5, 17, 20, 18] to imitate this ability, by classifying unseen data from a few new categories. There are two main few-shot learning approaches: The first one is Meta-learning based methods [5, 1, 13, 4], which idea is to learn across tasks and adapt to new tasks. The second is Metric-learning based methods [17, 20, 18], which objective is to learn a pairwise similarity metric where a similar sample gets a high score and dissimilar samples gets a low score. These metric learning methods may also adopt the metric learning policy to learn across tasks. The main objective of these methods is to learn an effective embedding network in order to extract useful features of the task and discriminate on the classes which we are trying to predict. From this basic learning, there have been many extensions proposed to improve the performance of metric learning methods, some of those focus on pre-training the embedding network [2], task attention modules [3, 12, 21], optimization of embeddings [11] and use of different loss functions [21].

In this work, we focus on finding task-significant features, by applying different feature reduction techniques, and giving a score based on the inter and intra class separability. We believe that finding those relevant features for each task is important, as we can better discriminate between classes and obtain a better inference.

2. Material and Methods

2.1. Few-shot learning setting

The few-shot meta-learning setting consists of tasks, which can be seen as batches in traditional deep learning. A

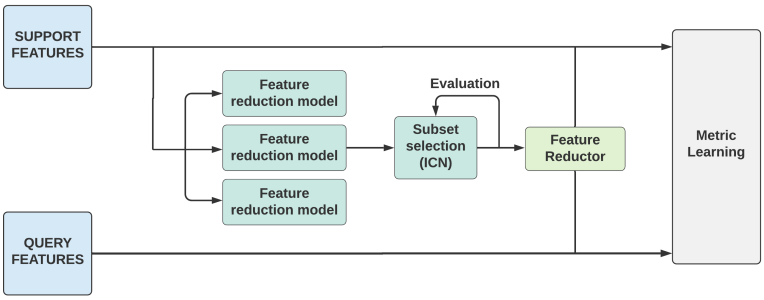

Figure 1. The components of the (ICNNnet). After obtaining the features from the feature extractor, a number of feature reduction models are applied to the support embeddings. The features obtained from these models are measured via the ICNN Score to select the best feature reductor. Finally, this reductor is applied to the support and query embeddings to continue with the metric learning inference.

task is made up of support data and query data. The support set contains k previously unseen classes and n instances for each class, and the objective is to classify q queries using the support data. This setting is also known as k-way n-shot (e.g. 5-way 1-shot or 5-way 5-shot). As described in [20, 15], the model is trained using an episodic mechanism, where each episode is loaded with a new random task taken from the training data.

2.2. miniImageNet dataset

For the experimental results we used MiniImageNet [20], which is a subset of ImageNet version of ILSVRC-2012 [16] and is used as a benchmark for the evaluation of few-shot learning methods. This subset is comprised of 100 classes, each one containing 600 images, making up a total of 60,000 images. We follow the split proposed by Ravi and Larochelle [15], dividing the dataset into 64 classes for training, 16 classes for validation and 20 classes for testing.

2.3. Metrics and baselines

Following most of the metric learning methods [17, 20, 18], we report our results on the mean accuracy (%) over 600 test episodes with 95% of confidence intervals.

As our work is an extension to the metric learning models, we are comparing against the three main methods: (1) Prototypical Networks [17], (2) Matching Networks [20] and (3) Relation Networks [18].

3. Proposed Model

For our proposed model (see Figure 1), we adopt a feature selection strategy based on the inter-intra class nearest neighbors distance. After obtaining the embeddings from the feature extractor, we are left with a number of feature vectors, each one representing a sample. From here, we want to obtain the features relevant for the given task. We apply different feature reduction methods, and obtain an intra-class and inter-class score for each one. These scores are used to select the method which helps us to obtain the best dimensions for the current task. These features obtained are then used by a metric learner to produce a classification.

3.1. Feature reduction techniques

We selected the following feature reduction strategies to apply them with the feature vectors obtained from the few-shot learning task:

- Principal Component Analysis (PCA) [6]: A dimensionality reduction method which tries to preserve as much information of the data as possible.
- Uniform Manifold Approximation and Projection (UMAP) [14]: A manifold learning technique for non-linear dimension reduction that tries to preserve the global structure of the data.
- Isometric mapping (Isomap) [19]: A non-linear dimensionality reduction which seeks to preserve the geodesic distances between the data points.

3.2. Inter and Intra Class Nearest Neighbors Score (ICNN Score)

There are two main concepts for the ICNN feature selection technique: Inter-class distance and Intra-class distance. Inter-class distance refers to the distance between points of different classes, and Intra-class distance refers to the distance between points of the same class. The idea for a successful feature selection is to choose the one which increase the inter-class distance and reduce the intra-class distance, in order to allow the task to be differentiated.

The ICNN Score is a measure that combines the distance and variance of the inter-intra k-nearest neighbors of each instance in the data:

$$ICNN(X) = \frac{1}{|X|} \sum_{x_i \in X} \lambda(X_i)^{\frac{1}{p}} \omega(X_i)^{\frac{1}{q}} \gamma(X_i)^{\frac{1}{r}} \quad (1)$$

where p, q and r are control constants.

λ is a function that penalizes the neighbors of X_i with the same class based on how distant they are, and the neighbors of different classes based on how close they are:

$$\lambda(X_i) = \frac{\sum_{p \in K_{\tilde{x}_i}} \frac{d(X_i,p)-\theta(X_i)}{\alpha(X_i)-\theta(X_i)} + \sum_{q \in K_{x_i}} 1 - \frac{d(X_i,q)-\theta(X_i)}{\alpha(X_i)-\theta(X_i)}}{|K_{x_i}| + |K_{\tilde{x}_i}|} \quad (2)$$

where $K_{x_i} = KNN(x_i) \in y_i$ are the set of k-nearest neighbors of x_i that have the same class,

$K_{\tilde{x}_i} = KNN(x_i) \in y_j \neq y_i$ are the set of k-nearest neighbors of x_i that has different class,

$d(a, b)$ is a distance function, which in this case is the euclidean distance,

$\alpha(X_i)$ and $\theta(X_i)$ are the maximum distance and the minimum distance respectively of the x_i neighbors.

In the ideal scenario, the neighbor's distance of the same class are close to 0 and the distance with different classes are close to 1.

ω is a function that penalizes the distance variance of neighbors:

$$\omega(X_i) = 1 - (Var(\sum_{p \in K_{\tilde{x}_i}} \frac{d(X_i, p) - \theta(X_i)}{\alpha(X_i) - \theta(X_i)}) + Var(\sum_{q \in K_{x_i}} 1 - \frac{d(X_i, q) - \theta(X_i)}{\alpha(X_i) - \theta(X_i)})) \quad (3)$$

where a high variance is penalized because it increase the possibility of overlapping classes.

The γ function describes the ratio of the neighbor's classes:

$$\gamma(x_i) = \frac{|K_{x_i}|}{|K_{x_i}| + |K_{\tilde{x}_i}|} \quad (4)$$

where is penalized based on the neighbors in the same class of x_i. Each of the three functions (λ, ω and γ) have an output between 0 and 1.

Using this metric, we can evaluate each feature reduction technique, as well as the original feature vector, to choose the best selection of features that are relevant for the current task.

3.3. Implementation details

To compare against the baselines, our experiments are made under the 5-way 1-shot and 5-way 5-shot setting with 15 query images for each class in the task. All the input images are resized to 84 × 84. On the training phase, we randomly construct 100 tasks over 200 epochs and apply validation over 500 tasks after every epoch. We train the network using Adam optimizer [9] with cross-entropy loss. The initial learning rate is set to 0.001 and is reduced by half every 20 epochs. For the testing phase, we randomly construct 1,000 tasks and measure the mean accuracy with 95% confidence intervals. For the feature extractor, we use a ConvNet with 4 layers, each one with a 3 × 3 convolution, followed by a Batch Normalization and ReLU layer.

4. Results and Discussion

There are several experiments and design choices that we test and discuss below. (1) We found that using UMAP in our feature reduction models, the training phase execution time greatly increased. For this reason, we decided to remove UMAP from the methods used in training, and use it only on the testing. (2) The UMAP and Isomap methods can perform the dimension reduction in a supervised and unsupervised way, we found that using a supervised reduction doesn't have an advantage over the unsupervised reduction. (3) In order to fit the feature reduction model, we can use only the support data or the data from the support and the query set. We found that using the support and query data, allowed the feature reduction methods to better interpret the structure of the data, thus obtaining a better ICNN score.

Method	5-way 1-shot	5-way 5-shot
Matching Net [20]	43.56	55.31
Prototypical Net [17]	49.52	**68.20**
Relation Net [18]	50.44	65.32
Prototypical Net + ICNN	**50.96**	67.72

Table 1. Test accuracy results for 5-way setting

Table 1, compares the three main metric learning methods and our proposed model. We obtained an improvement of around 1.5% for the 5-way 1-shot setting on the test set using Prototypical Networks. We also achieved a better performance than Matching Nets and Relation Nets on 5-way 5-shot setting, but obtained around 0.5% less accuracy than Prototypical Nets.

5. Future Work

There are still some design choices we need to test, in order to decide whether our techique can help the accuracy of the baseline models. (1) Up until now, we have only tested the model by reducing the features to 6 components. We need to test if reducing to different number of components would give a better ICNN score and find the best number of components for each method. (2) From the current results, the training phase is yielding a slightly worse accuracy than training with the baseline model. Another approach would be to use the feature reduction with ICNN score only on the testing phase, which would allow us to reduce dimensions using UMAP. If this experiment results in a better accuracy, we could propose our method for using it only to improve the results of inference. (3) Recently, models proposed for few-shot learning use different embedding networks, typically variations of ResNet [8], to greatly improve the accuracy of the models. The pre-training of the embedding network has also show great improvement on the performance of metric-learning methods [2]. Adding these components to our proposed method, we should be able to obtain a much better performance.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. *CoRR*, abs/1606.04474, 2016. 1

[2] Da Chen, Yuefeng Chen, Yuhong Li, Feng Mao, Yuan He, and Hui Xue. Self-supervised learning for few-shot image classification. *CoRR*, abs/1911.06045, 2019. 1, 3

[3] Haoxing Chen, Huaxiong Li, Yaohui Li, and Chunlin Chen. Multi-scale adaptive task attention network for few-shot learning, 2020. 1

[4] Yutian Chen, Matthew W Hoffman, Sergio Gómez Colmenarejo, Misha Denil, Timothy P Lillicrap, Matt Botvinick, and Nando Freitas. Learning to learn without gradient descent by gradient descent. In *International Conference on Machine Learning*, pages 748–756. PMLR, 2017. 1

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017. 1

[6] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 2

[7] Nilay Ganatra and Atul Patel. A comprehensive study of deep learning architectures, applications and tools. *International Journal of Computer Sciences and Engineering*, 6:701–705, 12 2018. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 3

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[10] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015. 1

[11] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. *CoRR*, abs/1904.03758, 2019. 1

[12] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. *CoRR*, abs/1905.11116, 2019. 1

[13] Ke Li and Jitendra Malik. Learning to optimize. *CoRR*, abs/1606.01885, 2016. 1

[14] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 2

[15] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. 2

[17] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4077–4087. Curran Associates, Inc., 2017. 1, 2, 3

[18] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. *CoRR*, abs/1711.06025, 2017. 1, 2, 3

[19] Michael W Trosset and Gokcen Buyukbas. Rehabilitating isomap: Euclidean representation of geodesic structure. *arXiv preprint arXiv:2006.10858*, 2020. 2

[20] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016. 1, 2, 3

[21] Yan Zheng, Ronggui Wang, Juan Yang, Lixia Xue, and Min Hu. Principal characteristic networks for few-shot learning. *Journal of Visual Communication and Image Representation*, 59:563 – 573, 2019. 1