International Journal of Advanced Smart Convergence Vol.8 No.4 75-81 (2019)
http://dx.doi.org/10.7236/IJASC.2019.8.4.75

IJASC 19-4-10

Learning Deep Representation by Increasing ConvNets Depth for Few Shot Learning

Fabian H. S. Tan[1] and Dae-Ki Kang[2]

[1]*Master Student,* [2]*Professor, Department of Computer Engineering, Dongseo University, Korea*
fnever520@gmail.com, dkkang@dongseo.ac.kr

Abstract

Though recent advancement of deep learning methods have provided satisfactory results from large data domain, somehow yield poor performance on few-shot classification tasks. In order to train a model with strong performance, i.e. deep convolutional neural network, it depends heavily on huge dataset and the labeled classes of the dataset can be extremely humongous. The cost of human annotation and scarcity of the data among the classes have drastically limited the capability of current image classification model. On the contrary, humans are excellent in terms of learning or recognizing new unseen classes with merely small set of labeled examples. Few-shot learning aims to train a classification model with limited labeled samples to recognize new classes that have never seen during training process. In this paper, we increase the backbone depth of the embedding network in order to learn the variation between the intra-class. By increasing the network depth of the embedding module, we are able to achieve competitive performance due to the minimized intra-class variation.

Keywords: *Embedding, Few-shot Learning, Classification, Network Depth*

1. Introduction

Human exhibits strong capability to understand concept and make inference quickly in real life. We can perceive and recognize the objects' variation fairly easy based on prior knowledge or deductive reasoning that we have. For example, we can easily generalize the appearance of a “panda” by learning from a picture on the web or book. Throughout all these years, deep learning system have attained promising performance on image recognition. However, in order to train a model with strong performance, i.e. deep convolutional neural network [1], it depends heavily on huge dataset and the labeled classes of the dataset can be extremely humongous. The cost of human annotation and scarcity of the data among the classes have drastically limited the capability of current image classification model. On the contrary, humans are excellent in terms of learning or recognizing new unseen classes with merely small set of labeled examples. It has piqued the interest to increase model's performance to generalize new unseen classes with a few labeled examples for novel class.

Few-shot learning has recently been the focus within the community of machine learning [2]. Tremendous endeavors have been committed to overcome the efficiency issue. The aim of one-shot or few-shot learning is

Manuscript Received: October. 2, 2019 / Revised: October. 10, 2019 / Accepted: October. 16, 2019
Corresponding Author: dkkang@gmail.com
Tel: +82-10-7557-2944, Fax: +82-10-7557-2944
Professor, Department of Computer Engineering, Dongseo University, Korea

to train model with merely one training example or limited amount of training examples. They are learned by the means of good initial condition for the parameters [3], distance metric learning [4-6], or optimization [7]. Moreover, some researchers also introduced weight imprinting approach that achieved state-of-the-art performance [8]. It allows instant learning by computing the embedding activation and then extend the weight to form a new set of network weights on the classifier layer.

In this paper, we conducted experiments on the backbone depth to study the effects of network depth on the model's performance. We compared our approach with existing work, and it has showed promising outcome. The paper is organized as following: Section 2 talks about related work, Section 3 describes the general architecture of the network, and Section 4 shows the experiment results and followed by Section 5 concludes the paper.

2. Related Work

There are several approaches have been introduced for the case of few-shot learning. For instance, the **meta-learning based approaches** introduced by [3,7,9]. Ravi et al. [7] proposed the method to train the classifier by using a LSTM module to replace the stochastic gradient-based optimizer. Their line of work concentrates on learning the optimization algorithm. In addition to that, MAML method introduced by [3] can train the meta-learner model to provide a good initial condition for the parameters of the network, where these parameters can later be adapted easily with just few gradient updates steps and small number of labeled samples.

On the other hand, other approaches like **metric learning** are employed to make the classification on the unseen class by comparing the distance between the input embeddings. The mapping is learned from the input images to embedding space in which the images embedding from the same class are close to each other, and those of different classes are mapped far apart. The idea is that if a model was able to compute the similarities between two images, it can make the prediction on unseen novel class. Every feature of the query sample is compared to each local feature of the support samples. For example, some researchers train the classifier model to recognize image labels by introducing distance metric learning approach [10]. Matching network proposed by Vinyals et al. uses cosine similarity on the model to classify samples [5]. Whereas, some researchers introduced a classifier network called Prototypical network which employs the network to compute single mean prototype for every class label and classify the query sample based on the nearest neighbor prototype [6]. Relation network possesses the similar idea, the difference is it replaces the distance with learnable relation module [11].

Different research areas make use of **additional memory block** to retain crucial and prior information and later access and retrieve them on newly encountered episodes [10,12]. The idea of their network was inspired by Neural Turing Machine, whereby the models are extended with additional memory module [13]. Their Memory-Augmented Neural Network (MANN) can store the information in the external memory blocks for an extended period. Throughout the iterative training process, it enables the model to learn general representation of the input and quick memory access can fetch these general representations to new data.

3. Proposed Method

In this section, we explore the network depth described in [11]. We continue to make our modification on the network architecture, which extends the layer of the embedding module to 6 layers from 4 layers. The similarity module does not differ apart from its embedding module.

3.1 Problem Definition

Here we define support set $\xi_{spt} = \{(x_i, y_i)\}_{i=1}^{n}$ and query set $\xi_{qry} = \{(x_j, y_j)\}_{j=1}^{m}$, where x_{ij} denotes the input's feature, and y_{ij} denotes its corresponding label. Consider the number of classes is N and the number of samples per class is K, and hence the task represents N-way K-shot learning. Instances from both the query set and support set will be fed into the network model.

The model is made up of two segments - embedding module $\mathcal{F}_\theta$, and similarity module S_ϕ, as illustrated in Fig 1. The embedding module, $\mathcal{F}_\theta$ extracts respective features of the query samples and support samples, where θ indicates its parameters. The underlying idea is to learn the embedding function of the inputs and project them into embedding space. Then compute the similarity between the query sample and support samples on their embeddings using similarity module, S_ϕ. Few-shot recognition task is performed by learning to compare the feature pairs between the query set and the support set.

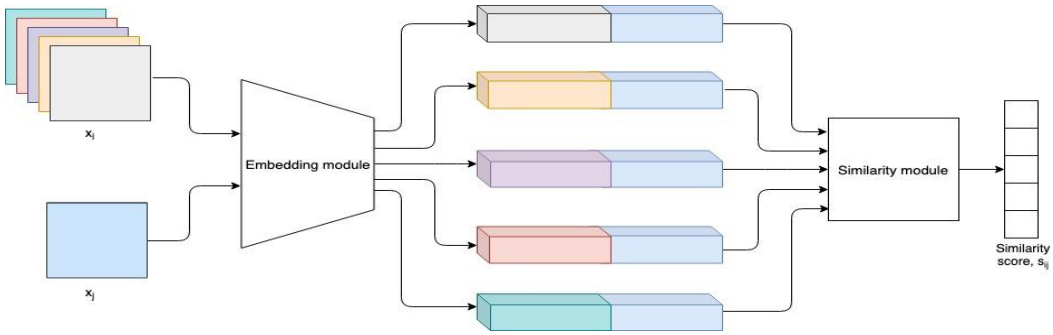

Figure 1. Diagram of the network architecture. Here it shows 5-way 1-shot classification task, where x_i represents the instance from the support set and x_j represents the instance from the query set.

3.2 Network Architecture

The detailed network architecture of our model is illustrated in Fig 2.

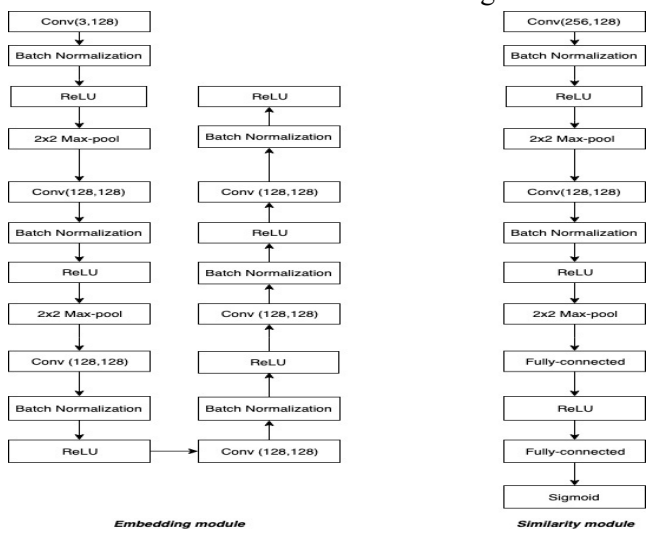

Figure 2. The detailed network architecture of the model

We begin with embedding module, $\mathcal{F}_\theta$. Conventional convolutional neural network (CNN) is used as the feature extractor to extract the features from the input sample. Instead of using Conv-4 in [5,11], we increase the backbone depth of the convolutional blocks to form a 6-layer CNN module, whereby it has two auxiliary convolutional blocks without the pooling layer. Conv-6 network is constructed as following sequence: a 3x3 kernel-size window with 128 filters, batch normalization layer [11], ReLU activation function and followed by 2x2 max-pooling layer. Besides the first two layers, subsequent layers in the embedding module employ padding to maintain the shape of the output feature maps.

Images from query set and support set are fed into Conv-6 and convolved with 3x3 kernel-size window to generate feature maps. Embedding module parses the pixels from the input images into a set of objects. Rich visual representations are learned during the training on the base classes. Every feature map from the query set and support set will be concatenated and formed a set of pairs following the rule $\mathcal{C}$ in Eq. 1.

$$\mathcal{C}(\hat{x_i}, \hat{x_j}) = concat\left(\mathcal{F}_\theta(x_i), \mathcal{F}_\theta(x_j)\right) \tag{1}$$

Similarity module, S_ϕ will operate on these set of feature pairs and classify them via sigmoid function. Sigmoid function is preferred choice because the output classes of the images are independent to each other. S_ϕ makes the inference based on the embedding between the features and compute the similarity score via Eq 2.

$$s_{ij} = S_\phi\left(\mathcal{C}(\hat{x_i}, \hat{x_j})\right) \tag{2}$$

Mean square error function is used as the training objective (Eq. 3) to minimize N-way prediction loss. By optimizing this training objective, the learned parameters can be updated, and the predicted similarity will be close to ground truth value.

$$\theta \leftarrow argmin \sum_{i=1}^{m} \sum_{j=1}^{n} (s_{ij} - y)^2 \tag{3}$$

4. Experiment Result

We evaluate our model on two benchmark dataset, namely Omniglot and mini-Imagenet. With these two benchmark datasets, we carried out our approach revolving around N-way K-shot settings. They will be explained in the following subsections.

4.1 Evaluation on Omniglot Dataset

The Omniglot dataset comprises of 1,623 handwritten characters from 50 different alphabets, with each drawn by different individuals. It was aggregated by Brenden [15]. It has been considered as the transpose of MNIST dataset for its wide range of classes and small set of instances per every class.

In this experiment, we downsized the size of the images to 28x28 and augmented them with 4 rotated degrees, namely 0°, 90°, 180° and 270° respectively. This generates more data samples from the existing dataset. We then partitioned the data into two subsets - the first 1200 classes plus their corresponding augmentation are used as training set whereas the remaining 423 classes plus corresponding augmentation are used as the testing set.

Table 1. Averaged accuracies of the few-shot classification task with 95% confidence intervals on Omniglot dataset. Best performance is highlighted in bold. The field filled with '-' indicates not reported.

		5-Way		20-Way	
Model	Fine Tune	1-shot	5-shot	1-shot	5-shot
Memory-Augmented NN [9]	N	82.8%	94.9%	-	-
Convolutional SiameseNets [12]	N	96.7%	98.4%	88.0%	96.5%
Convolutional SiameseNets [12]	Y	97.3%	98.4%	88.1%	97.0%
MatchingNets [13]	N	98.1%	98.9%	93.8%	98.5%
MatchingNets [13]	Y	97.9%	98.7%	93.5%	98.7%
PrototypicalNets [14]	N	98.8%	99.7%	96.0%	98.9%
MAML [15]	Y	98.7±0.4%	99.9±0.1%	95.8±0.3%	98.9±0.2%
RelationNet [16]	N	99.6±0.2%	99.8±0.1%	97.6±0.2%	99.1±0.1%
Ours	N	**99.7±0.2%**	99.8±0.1%	**98.2±0.4%**	**99.1±0.2%**

We configured our model by setting up the 6-layer deep embedding module and similarity module, as illustrated in Fig. 2. To begin with, the value of the input channel is set as 1. As the input images are fed into model, $\mathcal{F}_\theta$ produces 128 feature maps for the output layer, with each comes with the size of 5x5. Subsequently, S_ϕ takes the feature pairs outputted from the Conv-6 embedding module and computes the similarity score. The learned parameters (θ,ϕ) are updated with Adam optimizer [16] with learning rate of 0.001.

We evaluate our model on four classification tasks, i.e. 5-way 1-shot, 5-way 5-shot, 20-way 1-shot and 20-way 5-shot respectively. The results of the existing approaches [4-6,10-11] and ours are compared and shown in Table 1. Our result is reported in form of averaged accuracy that sampled from 1,000 episodes, along with 95% confidence interval.

From the table, it can be seen that our approach outperformed prior existing model in 5-way 1-shot, 20-way 1-shot and 20-way 5-shot. Although the result of the existing models has been state-of-the-art, particularly in 5-way tasks, a small breakthrough on the accuracy can be considered non-trivial. Our approach achieved state-of-the-art results especially for 1-shot classification tasks, which are 99.7% in 5-way and 98.2% in 20-way respectively. Therefore, it indicates that the model can classify the data in an extremely high accuracy with merely one input sample. We then challenge our model by performing the experiment on miniImagenet, which described in the next subsection.

4.2 Evaluation on mini-Imagenet Dataset

The mini-Imagenet dataset contains 60,000 images from 100 different classes, in total. It was collected by [7]. All these images are in the form of 3-channels compared to Omniglot. In this experiment, these samples are downsized to 84x84. It is more challenging to perform good classification task in miniImagenet owing to the number of channels and the great variation among the images.

The configuration of the embedding module and similarity module are kept the same as Omniglot, which described above. The value of input channel is configured as 3, instead of 1 in Omniglot. $\mathcal{F}_\theta$ produces 128 feature maps, and each with the size of 19x19. S_ϕ is used to generate the similarity score of the corresponding feature pairs of each query sample and support samples. We also update learned parameters (θ,ϕ) with Adam optimizer with the learning rate of 0.001.

5-way, K-shot classification (K = 1, 5) tasks are performed and recorded. Table 2 shows the K-shot classification accuracies on the mini-Imagenet. We compared our method to MatchingNet, Meta-LSTM, MAML, PrototypicalNet and RelationNet. Our results are computed in the form of averaged accuracy that sampled from 1,000 episodes from the test set. They are also presented with 95% confidence intervals.

Table 2. Averaged accuracies of the few-shot classification task with 95% confidence intervals on miniImageNet dataset. Best performance is highlighted in bold.

		5-Way	
Model	Fine Tune	1-shot	5-shot
MatchingNet [13]	N	43.56±0.84%	55.31±0.73%
Meta LSTM [4]	N	43.44±0.77%	60.60±0.71%
MAML [15]	Y	48.70±1.84%	63.11±0.92%
PrototypicalNets [14]	N	49.42±0.78%	**68.20±0.66%**
RelationNet [16]	N	50.44±0.82%	65.32±0.70%
Ours	N	**50.77±0.91%**	65.00±0.73%

As shown in Table 2, our method surpassed the performance of the existing methods and [11] on the 5-way 1-shot task except for the 5-shot scenario. On the 1-shot classification task, it achieved 50.77% accuracy and has higher performance compared to [11]. We observed that increased number of filters and 6 layers of embedding module possess trivial influence to the performance. Hence, Conv-6 in the embedding module improves the classification accuracy because it learns deeper embedding and minimizes the intra-class variation. Note that network model with much denser convolutional layers (>6) gives poor result as it starts to overfit.

5. Conclusion

In summary, we proposed a simple Conv-6 embedding module to the network for few-shot learning. Few-shot learning is one of the challenging tasks due to the intricacy of the training algorithms and the scarcity of the data. Our method gives competitive result compared to existing state-of-the-art approaches. We show promising improvements on both Omniglot dataset and mini-Imagenet dataset. It learns deeper embedding and helps to minimize the intra-class variation among the images and hence improves the accuracy performance of the model. The method is intuitive and has great generalization performance. Our future research include the application of our proposed algorithm to the broader range of tasks [17,18].

Acknowledgement

This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2018R1D1A1A02050166) and Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00245, Development of prevention technology against AI dysfunction induced by deception attack).

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in Proc. NIPS, pp. 1106-1114, 2012.

[2] H. Edwards and A. Storkey, "Towards a Neural Statistician," in Proc. International Conference on Learning Representation (ICLR), Poster, 2017.

[3] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," in Proc. International Conference on Learning Representation (ICML), 2017.

[4] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-Shot Image Recognition," in Proc. ICML, 2015.

[5] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching Networks for One Shot Learning,"

in Proc. NIPS, 2016.
[6] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical Networks for Few-Shot Learning," in Proc. NIPS, 2017.
[7] S. Ravi and H. Larochelle, "Optimization as a Model for Few-Shot Learning," in Proc. ICLR, 2017.
[8] H. Qi, M. Brown, and D. G. Lowe, "Low-Shot Learning with Imprinted Weights," in In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[9] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to Learn Quickly for Few-Shot Learning," in Proc. CoRR, 2017.
[10] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-Learning with Memory Augmented Neural Network," in ICML, 2016.
[11] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H.S. Torr, and T. M. Hospedales, "Learning to Compare: Relation Network for Few-Shot Learning," in Proc. CVPR, 2018.
[12] L. Kaiser, O. Nachum, A. Roy, and S. Bengio, "Learning to Remember Rare Events," in Proc. ICLR, 2017.
[13] A. Graves, G. Wayne, I. Danichelka, "Neural Turing Machines," arXiv preprint arXiv:1410.5401, 2014.
[14] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv preprint arXiv:1502.03167, 2015.
[15] B. Lake, R. M. Salakhutdinov, J. Gross and J. B. Tenenbaum, "One Shot Learning of Simple Visual Concept," in Proc. 33rd Annual Conference of the Cognitive Science Society, Vol. 172, 2011.
[16] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in Proc. ICLR, 2015.
[17] K. Li and D.-K. Kang, "FAST-ADAM in Semi-Supervised Generative Adversarial Networks," International Journal of Internet, Broadcasting and Communication (IJIBC), Vol. 11, No. 4, pp. 31-36, Nov. 2019.
DOI: http://dx.doi.org/10.7236/IJIBC.2019.11.4.31.
[18] Z.-Y. Wang and D.-K. Kang, "Experimental Analysis of Equilibrization in Binary Classification for Non-Image Imbalanced Data Using Wasserstein GAN," International Journal of Internet, Broadcasting and Communication (IJIBC), Vol. 11, No. 4, pp. 37-42, Nov. 2019.
DOI: http://dx.doi.org/10.7236/IJIBC.2019.11.4.37.