Memory Augmented Matching Networks for Few-Shot Learnings

Kien Tran, Hiroshi Sato, and Masao Kubo

Abstract—**Despite many successful efforts have been made in one-shot/few-shot learning tasks recently, learning from few data remains one of the toughest areas in machine learning. Regarding traditional machine learning methods, the more data gathered, the more accurate the intelligent system could perform. Hence for this kind of tasks, there must be different approaches to guarantee systems could learn even with only one sample per class but provide the most accurate results. Existing solutions are ranged from Bayesian approaches to meta-learning approaches. With the advances in Deep learning field, meta-learning methods with deep networks could be better such as recurrent networks with enhanced external memory (Neural Turing Machines - NTM), metric learning with Matching Network framework, etc. In our research, we propose a metric learning method for few-shot learning tasks by taking advantage of NTMs and Matching Network to improve few-shot learning task's learning accuracy on both Omniglot dataset and Mini-Imagenet dataset. In addition, a weighted prototype is also introduced to improve the overall performance of the proposed model, especially on the complicated benchmark datasets such as mini-ImageNet.**

Index Terms—**Few shot learning, matching network, memory augmented neural network, prototypical network.**

I. Introduction

Importance of Artificial Intelligent (AI) in human life has been proven. Thanks to the rapid advancement of AI, many sectors have started using AI technology to reduce human efforts. People gain benefit from them in many areas, from virtual assistants to the space industry. Deep learning, which is a subfield of machine learning, is one of the most powerful and fastest-growing applications of AI. For an image classification problem, deep learning methods overcome the traditional machine learning methods. It is gaining much popularity because of its supremacy in terms of accuracy of complicated problems, i.e., Deep Convolutional Neural Networks (ConvNets).

Deep learning occurs through the use of neural networks, which are layered to recognize patterns and complex relationships in images, and run through lots of iterations of stochastic gradient descent and gradually refine the weights of the network. However, the state-of-the-art deep learning models usually require large dataset with numerous samples such as ImageNet, CIFAR10, and mighty computational power to implement. This capability is one of the most significant limitations of using deep learning in image recognition applications. Thus, deep networks are often broken down when they have to learn a new concept on the fly or to deal with the problems which information/data is insufficient (few or even a single example). These situations are very common in the computer vision field, such as detecting a new object with only a few available captured images. For those purposes, specialists define the one-shot/few-shot learning algorithms that could solve a task with only one or few known samples (e.g., less than 20 examples per category).

One-shot learning/few-shot learning is a challenging domain for neural networks since gradient-based learning is often too slow to quickly cope with never-before-seen classes, and they are evaluated as one of the hardest challenges of AI. Recently, some researches have been proposed and bring very well results. For examples, the works based on the Bayesian approach, such as the works of L. Fei-Fei *et al*. [1], or Hierarchical Nonparametric Bayesian Model by R. Salakhutdinov *et al*. [2], or meta-learning approach with Memory Augmented Neural Network introduced by A. Santoro *et al*. [3], Model-Agnostic Meta-Learning (MAML) by C.Finn *et al*. [4], or Metric learning such as Siamese Neural Network by G.Koch *et al*. [5], Matching Network (MNet) for one-shot learning by O. Vinyals *et al*. [6], or Prototypical Network by Snell *et al*. [7].

Regarding meta-learning approaches, the Memory Augmented Neural Network approach uses the Neural Turing Machine (NTM) as recurrent models to achieve one-shot learning tasks. Matching Network and Siamese Neural Network are metric learning methods that consider a distance between training samples, or support samples, and a test sample.

In this paper, the Memory Augmented Matching Network, which is based on the Matching Network framework and Memory Augmented Neural Network, is introduced. This model is a combination of the Matching Network and the Memory Augmented Neural Network for one-shot learning problems and evaluated on the Mini-ImageNet and the Omniglot dataset. We also improve the overall performance of this model by taking advantage of a prototypical class, in which each class is represented by only one mean sample.

The main contribution of this work is to successfully integrate the NTM units into the old Matching Network model, consequently improves the overall results of the model in one-shot/few-shot learning tasks. Since this approach is a combination of meta-learning method (MANN) and metric learning (MNet), in some cases, fine-tuning is necessary to improve the performance of the model. Further, our approach also introduces a new definition of a representative sample based on the distances of all samples in the same class. This approach improves better performance compared to the original method of the prototypical class.

Manuscript received July 19, 2019; revised October 7, 2019.

The authors are with the Department of Computer Science at National Defense Academy of Japan (e-mail: t2kien@ gmail.com, hsato@nda.ac.jp, masaok@nda.ac.jp).

doi: 10.18178/ijmlc.2019.9.6.867

II. Related Works

A. *Neural Turing Machine*

Introduced by Graves *et al*. [10], Neural Turing Machine (or NTM) is the most cutting edge research at that moment that resembles an external working memory. This model is constructed according to the Interface–Controller abstraction, which consists of two main intimately connected components: a controller and an external memory bank. A neural network acts as a controller that provides an internal representation of the inputs controls the read and write operations of the augmented memory component via read/write heads. The whole architecture of the NTM is differentiable to compute the gradient of output loss regarding the hyper-parameters of the model. So it could be viewed as a differentiable version of a Turing machine. Regarding memory access mechanism, the NTM uses Attention based Memory Access allowing each read/write head to generate a normalized softmax attention vector to examine all the location in the memory at the same time. Then, it could select the most appropriate content depending on the memory addressing mechanisms. According to the authors, two addressing mechanisms are introduced, content-based addressing and location-based mechanism. Thanks to these mechanisms as well as the external memory component, it could preserve kinds of sub-sequence pattern explicitly, generalizing over longer sequences than the training sequences, and more effective to solve algorithmic tasks than other recurrent models like LSTMs. Although this architecture could do some simple tasks in practice such as copying, sorting, as well as associative recall from input and output, and still suffers from some issues, it is described by experts as a promising direction for future research.

B. *Meta Learning with Memory Augmented Neural Network for One-Shot Learning Task*

Recent works have suggested the Memory Augmented Neural Network (MANN) for one-shot learning tasks via meta-learning approach. Taking advantage of the Neural Turing Machine (NTM), which provides a promising approach for meta-learning in deep learning networks, the MANN with an ability to rapidly assimilate new data in its memory, demonstrates the capability of solving one-shot learning tasks in term of short and long term memory demands. In models such as the MANN, the input sequences and their corresponding labels from the previous step are bound together in the same memory locations. These data representations will be used to achieve perfect accuracy thereafter via a new addressing mechanism called Least Recently Used Access (LRUA). This access module replaces the two previous memory access types of the NTM for the tasks of emphasizing a conjunctive coding of information independent of sequence. By writing information into either the least used memory location (rarely-used locations) or the most recently used memory location (last used location - update memory with newer information) of the external memory, it makes the model "*the ability to slowly learn an abstract method for obtaining useful representations of raw data, via gradient descent, and the ability to rapidly bind never-before-seen information after a single presentation, via an external memory module.*" [3].

C. *Matching Network for One-Shot Learning Tasks*

In addition to the MANN, our works also follow the architecture of the Matching Network proposed by O. Vinyals *et al*. [6]. This model is inspired by the work of some neural networks with extended memory such as Memory networks, Pointer networks, etc. Differ from the MANN, which is a meta-learning model using recurrent networks, this framework is a nonparametric approach that based on metric learning to focus on extracting features and computing a distance between these features vectors via an attention kernel. Depend on the tasks, the feature extractor functions could be varied, such as Deep Convolution Networks, a simple form of word embedding function, etc.

Basically, this model computes output $\hat{y}$ as follow:

$$\hat{y} = \sum_i^k a(\hat{x}, x_i) y_i \tag{1}$$

The attention function $a(\hat{x}, x_i)$ uses the softmax over the cosine distance with two embedding functions f and *g*. Moreover, to archive maximum accuracy through the classification function described in (1), the full context embedding functions are used to modify the way of mapping input samples into the memory via attention mechanism. The memory caches the common pattern of representation and corresponding label of training samples. Then, the model predicts label by matching input samples with memory caches and generating a weighted sum label (with matching distribution) as a final output.

D. *Prototypical Networks*

Snell *et al*. [7] evolved Siamese networks by aggregating information within each support sets. The author takes advantage of the use of class mean as prototypes to counter the issue of overfitting due to the limitation of the data in few-shot learning tasks. Compare to recent approaches for few-shot learning, this network shows benefit in limited-data situations and achieved excellent results.

III. Proposed Methods – Memory Augmented Matching Network

A. *Matching Network with External Memory*

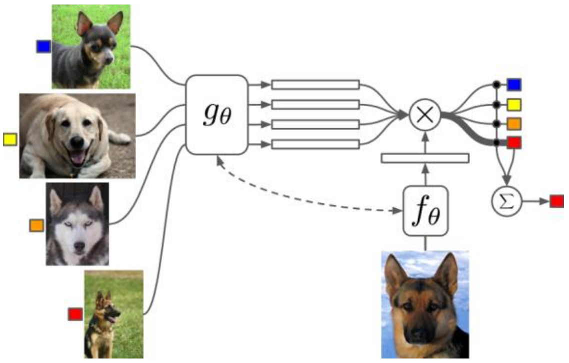

Fig. 1. Illustration of embedding learning methods for few-shot learning classification tasks. The figure is an excerpt from [6].

To address a challenge of K-shot N-way classification tasks, the proposed model apply embedding learning methods that embed $x \in X \subseteq R^d$ to a smaller embedding space $z \in Z \subseteq R^m$. Using these new spaces, it is easy to identify similar and dissimilar pairs of support samples and test sample. Currently, these methods have three main functions: function f(.) embeds sample $x^{\text{test}} \in D^{\text{test}}$ to *Z*, function *g*(.) embeds $x^{\text{support}} \in D^{\text{support}}$ to *Z* and a similarity measure *s*(.,.) calculates the similarities between the output of *f*(.) and each output of *g*(.) in the new space *Z*. An overview of this architect is illustrated in Fig. 1.

This proposed model is drew inspiration from architecture of the Matching Network model as well as the MANN model for one-shot learning tasks. Thus, its strategy is to enhance an embedding space with memory components, help accordingly recognizing unseen objects based on the content located in these memory matrices.

The whole model is a classifier $C(\hat{x})$ that correctly classify a test example $\hat{x}$ into a corresponding class according to the given support samples' labels. With a support set $S=\{(x_i,y_i)\}_{i=1..k}$, which contains k samples randomly sampled from *N* unique classes, the predicted output $\hat{y}$ of an unseen sample $\hat{x}$ with the prior knowledge gained from the support set S could be defined as:

$$\hat{y} = argmax_{i=1..k} P(y_i|\hat{x}, S) \tag{2}$$

where, $P(y_i\ /x, \hat{S})$ is the probability of classifying $\hat{x}$ with the class y_i conditioned on the support set *S*. The probability function *P* is parameterized by a neural network, and gradually improved through the back-propagation training of the neural networks.

For those purposes above, in this work, the model of MAMN consists of 4 modules: a feature embedding module, the two full context embedding modules with memory augmentation, and an attention kernel for similarity measure function, which determine the output $\hat{y}$ according to the distance of support set and test sample in the new embedding space.

The feature embedding module is used to extract only the necessary information from input samples for the feature space. This parameterized deep network is shared for both support set x_i and the test sample $\hat{x}$, and guarantees two input feature spaces uniform and treated equally for similarity measurement functions. In this work, a simple yet powerful CNN is used as the embedding function. An output is denoted as *e(.)*, used in the following functions.

The full context embedding module for the support set g(.)

This embedding function utilizes the Neural Turing Machine to map embedding features of support samples into its external memory. This function follows the implementation of MANN [3] for one-shot learning task using NTM units with an attentional mechanism called Least Recently Used Access mechanism (LRUA).

Given the input $e(x_i)$, the read vector *r* from the external memory and the hidden states (*h,c*) from the previous timestep, a LSTM unit, which act as the controller of NTM, produces key vectors (hidden states), then writes them into either the least used location or the most recently used location of the memory via write head using LRUA mechanism. It calculates the write-weight vector w_t^was (2)

$$w_t^w \leftarrow \sigma(\alpha) w_{t-1}^r + (1-\sigma(\alpha)) w_{t-1}^{lu} \tag{3}$$

where, $\sigma(\alpha)$ is a sigmoid function of a scalar parameter α, w_{t-1}^r is a read-weight vector of a previous step, and w_{t-1}^{lu} is the least used weight vector, generated from the usage weight vector w_{t-1}^u that update every step with a decay parameter *γ*.

$$w_t^u \leftarrow \gamma w_{t-1}^u + w_t^u + w_t^w \tag{4}$$

Then, the least-used weight w_t^{lu} is defined accordingly:

$$w_t^{lu} = \begin{cases} 0, & w_t^u(i) > m(w_t^u, n) \\ 1, & w_t^u(i) \le m(w_t^u, n) \end{cases} \tag{5}$$

where, the notation $m(w_t^u, n)$ represents the *n*th smallest element of w_t^u . The memory will be written in accordance with this write-weight vector

$$M_t\ (i) \leftarrow M_{t-1}(\text{i}) + w_t^w(i) k_t \tag{6}$$

The read heads compute the read vectors for each sample. These read vectors are the convex combination of the memory matrix and weighting vector, and are defined as:

$$\vec{r_t} \leftarrow \sum_i\ w_t^r(i) M_t(i) \tag{7}$$

where, *Mt(i)* is a cell ith of the memory used with sample tth in the support set. *wrt(i)* is a read-weight vector of cell i, and is produced according to a softmax of the similarity measure K between key vector *kt* and all the memory cell *Mt(i)*:

$$w_t^r(i) \leftarrow \frac{\exp\left(K(k_t, M_t(i))\right)}{\sum_j \exp\left(K(k_t, M_t(j))\right)} \tag{8}$$

The NTM receives a cascade of support samples and produces read vectors accordingly, results in a list of embedding vectors in the new space.

$$\vec{r_t} = NTM\left(e(x_i), \overrightarrow{r_{t-1}}, \overrightarrow{h_{t-1}}, \overrightarrow{c_{t-1}}\right) \tag{9}$$

$$g(x_i) = \vec{r_t} + \ e(x_i) \tag{10}$$

$$g(S) = \{g(x_i)\}_{i \in S} \tag{11}$$

The full context embedding module for the test sample f(.)

Similar to the function *g*(.), another NTM is used to embed test sample $\hat{x}$ into the same space for similarity measure. Beside the LRUA attentional mechanism, this function is also take advantage of a sequence to sequence attention mechanism by O. Vynial *et al.* [8]. This mechanism is also successfully used in the Matching Network for one-shot learning tasks.

$$\vec{r_t}, \vec{h_t}, \vec{c_t} = NTM(e(\hat{x}), \overrightarrow{r_{t-1}}, \overrightarrow{h_{t-1}}, \overrightarrow{c_{t-1}}) \tag{12}$$

The output of the function $f(\hat{x})$ is a combination of $\vec{r_t}$ and $e(\hat{x})$.

$$f(\hat{x}) = \vec{r_t} + \ e(\hat{x}) \tag{13}$$

However, the attention mechanism is continued by comparing $f(\hat{x})$ against each row in the output list of *g(S)*. The function $a(f(\hat{x}), g(x_i))$ is referred to as content based attention. In (14) below, the T represents a transpose of $f(\hat{x})$.

$$a(f(\hat{x}), g(x_i)) = softmax(T(f(\hat{x})), g(x_i)) \tag{14}$$

$$f\ '\ (\hat{x}) = f(\hat{x}) + \sum_i^{|S|} a\left(f(\hat{x}), g(x_i)\right) g(x_i) \tag{15}$$

To fully fit the size of NTM units' state, the $f'(\hat{x})$ is feed through a fully connected neural network before combining with the previous state of the NTM as in (16).

$$\overrightarrow{h_t} = \overrightarrow{h_t} + Dense\left(f'(\hat{x})\right) \tag{16}$$

The final output $f(\hat{x})$ is given after K times of implementing of this attention process.

The attention kernel An attention function $a(.,.)$ is the softmax over the cosine distance c between the two full context embedding functions f and g, which map extracted features from the output of the feature embedding module into the same feature space. It is defined as:

$$a\left(f(\hat{x}), g(x_i)\right) = \frac{\exp\left(c(g(x_i), f(\hat{x}))\right)}{\sum_j \exp\left(c(g(x_j), f(\hat{x}))\right)} \tag{17}$$

B. Weighted Prototypical Class

For the N-way K-shot classification tasks, Snell *et al.* [7] proposed a prototype that computes a representation c_k of class k (k=1..N) based on an average calculation of instances of that class.

$$c_k = \frac{1}{K}\sum_{i=1}^{K} g\left(x_{k,i}\right) \tag{18}$$

where, $g(x_{k,i})$ is an embedding function of sample x_i belonged to class k.

In some cases, the class distribution is skewed. That is, some samples could locate outside the range of major samples in the class. Appling prototype from (18) in such situation could lead to a biased mean sample of the class. One way to overcome this is to treat those samples unequally based on their weights. These weights, which are used to determine the relative importance of each data point, are considered as the distance of a point to other points in the same class. So, the contributions of the points to the representative point of their class are proportional to the distance between them and the others. We find a new representative sample for a class as:

$$c_k^w = \frac{1}{\sum_{i=1}^{K} w_{k,i}} \sum_{i=1}^{K} w_{k,i}\, g\left(x_{k,i}\right) \tag{19}$$

where, $w_{k.i}$ is the weight of sample $x_{k,i}$, and is the inverse of the total distance $d_{k,i}$ from sample i to other samples of class k as:

$$w_{k,j} = \frac{1}{\sum_{j=1, j\neq i}^{K} d_{k,i,j}} \tag{20}$$

Generally, with an increase of the number of the samples per class as well as the unbiased class distribution, the difference between the prototype sample and weighted prototype sample is not considerable. However, in few-shot learning problems, in which the number of labelled sample is small, if the class distribution is biased, the weighted prototype method could give better results.

In this paper, we use both prototype and weighted prototype vectors of the supporting samples to feed the function $g(.)$. In the case of one-shot learning tasks, there is no difference between the model using the prototype, weighted prototype, and the pure model since there is only one support point per class $x_k = c_k$.

IV. Experiments

For the few-shot learning experiments we used two benchmark datasets: the Omniglot dataset presented by Lake *et al.* [9] and Mini-ImageNet dataset introduced by Vinyals *et al.* [6] which is a small version of ILSVRC-12 Krizhevsky *et al.* All experiments are based on the N-way k-shot setting, and run under the used the same setting for both training and testing. For each run, a set of k labelled examples from each of N classes are randomly selected and fed into the model. Class labels are randomly chosen for each class from epoch-to-epoch. For example, labels are of size N where N is the number of unique classes in N-way classification tasks, and the inputs are labelled accordingly. The results are also compared to the original model as well as other baselines models. Each benchmark dataset is split into two disjointed sets, one is for training, and another is for testing. Either 5 or 20 unique classes per epoch task (i.e. 5-way 1- shot, 5-way 5-shot, 20-way 1 shot respectively) is trained and validated within 200 epochs. Subsequently, the best model is evaluated with the disjointed set under the same training condition. Moreover, the simple CNN is used as the embedding function to extract features from the input images. The architecture of CNN is varied according to the dataset used for experiments.

NTM units are set up with LRUA mechanism. The parameters are chosen as the followings: 128 memory slots of size 40, a controller (LSTM) with 200 hidden units, 4 memory read-out heads, and usage decay of write weights of 0.99.

A. Baselines

Some one-shot learning models are used as baselines in our experiments listed as follow:

- **MANN** is one of the most favorite models using Neural Turing Machine for one-shot learning tasks. This model also plays an important role in our work. According to the original paper, this model is conducted with only 5-way k-shot learning tasks with the Omniglot dataset.
- **Convolutional Siamese Networks** (Siamese Nets) introduced by G.Koch *et al.* [5]. This approach is supervised metric learning with Siamese Neural Networks that consists of twin Convolutional Neural Networks. This ensures the input images are recognized based on the similarity measurement.
- **Matching Networks** (MNets) is the inspiration model of our work, and its results are the highest compared to the two models above.
- **The Model-Agnostic Meta-Learning** (MAML) for fast adaptation of Deep Networks introduced by C. Finn *et al.* [4]. This is one of the state-of-the-art approaches in one-shot/few-shot learning. It trains a meta-learner to provide a good initialization for the parameters of the classifier.
- **Prototypical Networks** (ProtoNets) introduced by Snell *et al.* [7]. This model is another metric learning approach for one-shot/few-shot learning tasks. This can be seen as a variation of Matching Networks, and use the idea of class prototypes, that is the mean embedded vector of the supporting samples within a class.
- **Graph Neural Network** for few-shot learning: Utilize the Graph Neural Network, V. Garcia [12] constructed a framework that bases on the node-labeling framework,

which implicitly models the intra-cluster similarity and the inter-cluster dissimilarity.

- **RelationNet** Sung *et al.* [13] proposed a Relation Network for few-shot learning tasks. This network leans to learn a deep distance metric between query samples and support samples by computing relation scores.
- **Meta-SGD**: is an optimization-based approach to tackle one-shot learning problems introduced by Z.Li *et al.* [14]. This approach is an extension to MAML that solve the problem of optimization via modifying stochastic gradient descent.

B. Case 1: Omniglot Dataset

To evaluate the improvement of this model over the original, in this experiment, we test our model with Omniglot dataset conducted by Lake *et al.* [9]. This dataset consists of 50 different alphabets divided into 1623 different classes. Each class represents a character drawn online by 20 different people to create a set of 20 different black & white handwriting styles. Since these images are very small and simple, this collection is an ideal dataset for evaluating one-shot learning classification tasks. Figure 2 illustrates some Japanese Hiragana writing system's characters extracted from this dataset.

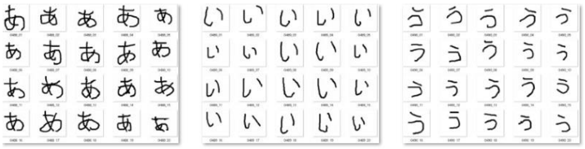

Fig. 2. Three examples of character classes of Omniglot dataset: letter a, i and u in Japanese Hiragana writing system. Each letter is written in 20 different handwriting styles.

TABLE I: Classification Accuracies for MAMN and Other Baselines on the Omniglot Dataset

Model	5-way Acc.		20-way Acc.	
	1-shot	5-shot	1-shot	5-shot
MANN	82.8 %	94.9%	-	-
Siamese Net	97.3%	98.4%	88.2%	97.0%
MNets	98.1%	98.9%	93.8%	98.5%
MAML	98.7 %	**99.9%**	95.8%	**98.9%**
ProtoNets	98.8 %	99.7%	96.0%	**98.9%**
MAMN	98.8%	99.5%	96.1%	98.6%
MAMN + ProtoClass	**98.9%**	99.7%	**96.3%**	**98.9%**
MAMN + Weighted ProtoClass	98.8%	99.6%	96.3%	98.8%

The Omniglot dataset is split into 3 subsets: 1200 characters are used for training, and the rest, which contains 422 characters, is split in half, one is for validation and the other is for evaluation. The inputs are also resized to 28×28, and performed data augmentation to overcome overfitting problem by randomly rotating by multiples of 90 degrees. We also perform fine-tune adjustments for the model using the support set sampled from the test dataset.

Following the embedding architecture from O. Vinyals *et al.*, a CNN consists of four stacked blocks of {3×3-convolutional layer with 64 filters, batch-normalization, 2×2 max-pooling, leaky-relu}. The output is passed through a fully connected layer resulting in a 64-dimensional embedding output. The results of the experiment on the Omniglot dataset are reported in Table I.

We extract the baseline results from the original paper [6] and other researches for comparison. Generally, the results of all the one-shot learning tasks (1- shot, 5-shot learning on 5 and 20 categories respectively) consistently point out that our developed model almost achieves remarkable performances against the competitive models. Particularly, the accuracies of 5-way 1-shot, 5-way 5-shot, and 20-way 1-shot and 5-shot tasks could reach 98.8%, 99.5% and 96.1% and 98.6% respectively making a significant improvement over its predecessor, the Matching Network. Compare to the two state-of-the-art approaches (MAML and Prototypical Networks), except the case of 5-way 5-shot that MAML model reached the maximum 99.9%, the MAMN model with prototypical class is superior although the gaps between it and others are not considerable. In contrast, the weighted prototypical class does not seem to help much in improving accuracy than the normal prototype on this dataset.

C. Case 2: Mini-Imagenet

The Mini-ImageNet dataset that O. Vinyals *et al.* [6] used consists of 60,000 color images of size 84×84 with 100 random classes from the ImageNet dataset [11]. The first 80 classes are used for training and the rest 20 classes for evaluating a model. Compare to the Omniglot dataset, this benchmark dataset is more difficult than the Omniglot dataset because of greater variations among the images within each class.

In this experiment, we construct CNN by using 5 blocks of {3×3-convolutional layer with 64 filters, batch-normalization, 2×2 max-pooling, leaky-relu} to generate the 64-dimensional outputs.

TABLE II: Classification Accuracies for MAMN and Other Baselines in 5-Way k-Shot Task on Mini-imagenet

Model	5-way Acc.	
	1-shot	5-shot
Matching Net	44.2%	57.0%
MAML	48.7%	63.1%
Meta SGD	**50.4%**	64.0%
ProtoNets	49.4%	68.2%
Graph Neural Network	50.3%	66.4%
RelationNet	**50.4%**	65.3%
MAMN	49.9%	63.2%
MAMN + ProtoClass	49.8%	66.5%
MAMN + Weighted ProtoClass	49.9%	**68.5%**

In Table II, we report the classification accuracies of our models and the baselines. The results yield higher performances of our models than the competitors. In 1-shot tasks, although our rivals such as meta-SGD or RelationNet could achieve higher accuracy with around 50.4%, the difference with our model (49.9%) is small. When more support images are provided in 5-shot tasks, the proposed model in combination with the weighted prototypical class reaches up to 68.5% accuracy, the highest result among all baselines.

D. Discussion

The proposed model could recognize specific features of support samples better and longer, lead to produce excellent accuracy. By observing experiments on the two datasets, our model definitely outperforms its originals. While on the Omniglot dataset, the improvement is not clear due to very high accuracy achieved on both models; on the mini-ImageNet dataset, the gap between models and their predecessors could be easily identified. For example, in the one-shot task, the MatchingNet produced only 44.2% while the improved model could give us 49.9% accuracy.

Moreover, the weighted prototype class has strongly shown its efficiency on such complex dataset as mini-ImageNet. Unlike the result of the previous experiment, since the features of the images in mini-ImageNet dataset are more complex than one of the Omniglot dataset, we could evaluate the performance of the model combining with the weighted representative samples. As indicated in Table 2, the weighted prototypical class considerably improves the overall performance of the MAMN model, typical around 5% accuracy, and 2% over the model using mean prototypical class. It is also noticed that in the one-shot learning task, as only one sample is used as support sample, there is ineffective to apply prototypical class as well as weighted prototypical class. In such situations, they become equivalent.

Further, since this approach is a combination of meta-learning method (MANN) and metric learning (MNet), in some cases, the overall performance could be improved by fine-tuning the features. This is proved by the fine-tuning results conducted by O. Vynials *et al*. with his MatchingNet.

However, as indicated in the original paper of Matching Network [6], one of the drawbacks of this model is that when the number of samples in the support set increases, the cost for training is also increased.

V. Conclusion

In this paper, we introduced a metric learning method in combination with meta-learning for one-shot learning task by way of enhancing the memory capacity for the embedding networks. It takes advantage of both Matching Network, Memory Augmented Neural Network with Least Recently Used Access mechanism, and Prototypical Networks for one-shot/few-shot learning tasks.

Moreover, a new prototype is also proposed to overcome the problem of biased distribution of class. The weighted mean is proportional to the distance between all vectors in the same cluster. It helps to improve the overall performance in some complicated dataset such as mini-ImageNet. However, to fully evaluate the out-performance of this research relative to its ancestors, there are still some experiments needed to conduct such as with one-shot language tasks or with longer input sequences i.e. 50-way k-shot tasks. In these tasks, the MANN units could show their advantages in one-shot/few-shot learning tasks. Finally, our work also introduces another practical application of Memory Augmented Neural Networks in one-shot learning challenge, suggests a replacement of the traditional RNN models with NTM units in one-shot learning models to achieve superior performance.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] F.-F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 594–611, vol. 28, no. 4, 2006.

[2] R. Salakhutdinov, J. Tenenbaum, and A. Torralba. "One-shot learning with a hierarchical nonparametric bayesian model," in *Proc. ICML Workshop on Unsupervised and Transfer Learning*, 2012, vol. 27, pp. 195–206.

[3] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. "Meta-learning with memory augmented neural networks," in *Proc. the 33rd International Conference on Machine Learning*, 2016, vol. 48, pp. 1842–1850.

[4] C. Finn, P. Abbeel, and S. Levine. "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. the 34th International Conference on Machine Learning*, 2017, vol. 70, pp. 1126-1135.

[5] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," presented at ICML 2015 Deep Learning Workshop, 2015.

[6] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3630-3638, 2016.

[7] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, pp. 4077–4087, 2017.

[8] O. Vinyals, S. Bengio, M. Kudlur, *Order Matters: Sequence to Sequence for Sets*, arXiv preprint arXiv:1511.06391, 2015.

[9] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, pp. 1332–1338, 2015.

[10] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *CoRR*, abs/1410.5401, 2014.

[11] O. Russakovsky, J. Deng, H. Su *et al*., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[12] V. G. Satorras and J. B. Estrach, "Few-shot learning with graph neural networks," presented at International Conference on Learning Representations, 2018.

[13] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[14] Z. G. Li, F. W. Zhou, F. Chen, and H. Li, *Meta-SGD: Learning to Learn Quickly for Few Shot Learning*, ArXiv, abs/1707.09835, 2017.

Kien Tran received the M.S. degrees in computer science from National Defense Academy of Japan in 2017. He currently is a doctoral student in the Department of Computer Science at National Defense Academy of Japan. His research is related to different applications of few-shot learning methods.

His area of interest includes machine learning, deep neural network, malware analysis, cyber security.

Hiroshi Sato is an associate professor in the Department of Computer Science at National Defense Academy in Japan. He received the B.E. degree in physics from Keio University in Japan, M.E and D.E in computer science from Tokyo Institute of Technology in Japan. His research interests include agent-based simulation, evolutionary computation, and artificial intelligence. He is a member of Japanese Society for Artificial Intelligence, Society of Instrument and Control Engineers, The Institute of Electronics, Information and Communication Engineers and so on.

Masao Kubo is an associate professor in the Department of Computer Science at National Defense Academy of Japan. He holds the B.E. degree in precision engineering from Hokkaido University, Japan in 1991, a Ph.D degree in computer science from Hokkaido University, Japan in 1996. His research interest is multi-agent systems.