Hindawi
Complexity
Volume 2022, Article ID 7330823, 8 pages
https://doi.org/10.1155/2022/7330823

Research Article

EW-CACTUs-MAML: A Robust Metalearning System for Rapid Classification on a Large Number of Tasks

Wen-Feng Wang [1,2] Jingjing Zhang,[1] and Peng An[2]

[1]*Shanghai Institute of Technology, Shanghai 201418, China*
[2]*Ningbo University of Technology, Ningbo 315211, China*

Correspondence should be addressed to Wen-Feng Wang; wangwenfeng@nimte.ac.cn

Received 27 September 2021; Accepted 16 December 2021; Published 3 February 2022

Academic Editor: Zhijie Wang

This study aims to develop a robust metalearning system for rapid classification on a large number of tasks. The model-agnostic metalearning (MAML) with the CACTUs method (clustering to automatically construct tasks for unsupervised metalearning) is improved as EW-CACTUs-MAML after integrated with the entropy weight (EW) method. Few-shot mechanisms are introduced in the deep network for efficient learning of a large number of tasks. The process of implementation is theoretically interpreted as "gene intelligence." Validation of EW-CACTUs-MAML on a typical dataset (Omniglot) indicates an accuracy of 97.42%, performing better than CACTUs-MAML (validation accuracy = 97.22%). At the end of this paper, the availability of our thoughts to improve another metalearning system (EW-CACTUs-ProtoNets) is also preliminarily discussed based on a cross-validation on another typical dataset (Miniimagenet).

1. Introduction

Generally, a learning algorithm f is defined as a procedure for processing the data D to make predictions $\widehat{y}^*$ from every input x^* [1]. That is, f is a particular function that maps x^* to $\widehat{y}^*$. In this sense, the goal of machine learning is to recover a function from data, including learning classifiers, regression, and policies [2]. Consequently, the learning algorithm f is said to be consistent if

$$\lim_{|D|\longrightarrow\infty} f\left(D, x_i^*\right) \longrightarrow y_i^*, \quad \forall\left(x_i^*, y_i^*\right). \tag{1}$$

Differing from traditional machine learning, metalearning is interpreted as "learn to learn," which can achieve (1), where the function from x_i^* to y_i^* can be actually presented as a universal metalearner [3]. The main research directions of metalearning include metalearning based on the metric space (e.g., prototypical networks), metalearning based on parameter optimization (e.g., model-agnostic metalearning), and model-based metalearning (e.g., reinforcement metalearning) [1–5]. The datasets for metalearning are very large, and hence, the automatic classification of learning tasks is always a great challenge [6]. Due to this challenge, few engineering applications of metalearning are reported [7, 8].

The objectives of this study are (1) to analyze the major reasons for the challenge, (2) to develop a method for tackling the challenge, and (3) to propose a scheme for engineering applications of metalearning. The organization of the whole paper is as follows. In Section 2, we formulate the problem as a challenge in a large-scale matrix operation, and in Section 3, we theoretically analyze how to further improve the accuracy and efficiency in classification. Experiments and discussion are presented in Section 4, where the room for improvement in parameter optimization is also highlighted.

2. Problem Formulation

2.1. Representation of the Model. We utilize the entropy weight method to improve the metalearning processes, where model-agnostic metalearning (MAML) is employed as the prototypical network [9–12].

Let θ be the vector of initial parameters for the model f and $\varnothing_J$ denote the updated parameters. Let α be the nonzero learning rate. For K-shot learning, we use 5-way-5-shot to build the prediction model [13–15].

$$\hat{y}^* = f_{MAML}\left(D_J, x^*; \theta\right) = f\left(x^*; \varnothing_J\right) = f\left(x^*; \theta - \alpha \nabla_\theta L\left(\theta, D_J\right)\right) = f\left(x^*; \theta - \alpha \nabla_\theta \frac{1}{K} \sum_{k=1}^{K} l\left(y_k, f\left(x_k; \theta\right)\right)\right). \quad (2)$$

According to the universal function approximation theorem [16–20], f_{MAML} can also be represented as an approximator for functions on x^*.

2.2. Interpreting the Learning Process. Let l be the lth task and $\varnothing\left(\cdot; w_j^l, \theta_{ft}^l, \theta_b^l\right)$ represent the input feature values which were evaluated by parameters θ_{ft}^l, bias θ_b^l, and transformation variable θ_b^l. Let $\prod_{i=1}^{N} W_i$ represent the weight matrices, which include a set of linear layers with nonnegative input and activations. Let $f_{\text{out}}\left(\cdot, \theta_{\text{out}}^l\right)$ be the output function. Let θ: $\left\{\theta_{ft}^l, \theta_b^l, \{W_i\}, \theta_{\text{out}}^l\right\}$ be the learned parameters.

We improve the traditional gradient descent utilized in the prototypical network to update the weights of the learner f, which can be represented as

$$\hat{f}(\cdot; \theta) = f_{\text{out}}\left(\left(\left(\prod_{i=1}^{N} W_i\right) \varnothing\left(\cdot; w_j^l, \theta_{ft}^l, \theta_b^l\right); \theta_{\text{out}}^l\right), l = 1, 2, 3, \ldots, n. \quad (3)$$

Choose θ_{ft} and $\theta_h\{A_i; i > 1\}, \{B_i; i < N\}$ such that

$$\hat{f}\left(x^*; \varnothing\right) = h_{\text{post}}\left(-\alpha \sum_{i=1}^{N} A_i \bar{e}(y) k_i\left(x, x^*\right); \theta_h\right). \quad (4)$$

Let $\operatorname{disc}(\cdot)$ denote a function that produces a K-shot discretization of its inputs. Select θ_{ft} and B_{jl} such that

$$k_{jl}\left(x, x^*\right) \coloneqq \begin{cases} 1 & \text{if } \operatorname{discr}(x) = e_j \text{ and } \operatorname{discr}\left(x^*\right) = e_l, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

The loss in classification is calculated with a cross-entropy function

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y)\ln(a)]. \quad (6)$$

A simplified interpretation of metalearning processes is shown in Figure 1.

3. Theoretical Analyses

3.1. Construction of Tasks. Suppose there is an embedding learning algorithm $\mathcal{F}$ on D; then, we can obtain the mapping data $\{x_i\}$ from the embedding space μ. For the cluster C_c, the centroid of cluster c_i is calculated from

$$P\{c_i\} = \arg_{\{C_c\}} \min_{\{c_i\}} \sum_{i=1}^{k} \sum_{\mu \in C_c} \left\|\mu - c_i\right\|^2. \quad (7)$$

Given a source matrix

$$r = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{pmatrix}. \quad (8)$$

The weight matrix calculated from r with the entropy weight method is

$$w = \begin{pmatrix} w_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{mn} \end{pmatrix}. \quad (9)$$

The prototype of the kth class is generated from

$$z_k = r w^T. \quad (10)$$

Hence, the set of examples labeled with class k is

$$S_k = \left(z_k, y_k\right). \quad (11)$$

We utilize k-means clustering division to get P and a set of partitions [21–28]. Let N be a support set of one-shot labels and Q be a query set. Each task can be sampled from a permutation with the one-shot labels $\{y_i\}$ obtained from CACTUs. That is,

$$T_t = \left\{\left(x_{m,n}, y_i\right)\right\}, \quad y_i \in \{1, 2, \ldots, K\}. \quad (12)$$

3.2. Parameters' Optimization. Entropy weight method is utilized in computing relative weights w_k for every data of tasks D_J and adapting to new tasks D_{Ji} which also determine the parameters of the model through the calculations of gradient descents $\varnothing_i$ [29, 30]. Let α be the global learning rate (a fixed metalearning parameter). Then,

$$\begin{aligned} \hat{y}^* &= f_{MAML}\left(D_J, w_k, x^*; \theta\right) \\ &= f\left(x^*; \varnothing_J\right) \\ &= f\left(x^*; \theta - \alpha w_k L\left(\theta, D_J\right)\right) \\ &= f\left(x^*; \theta - \alpha w_k \nabla_\theta \frac{1}{K} \sum_{k=1}^{K} l\left(y_k, f\left(x_k; \theta\right)\right)\right), \end{aligned} \quad (13)$$

$$\varnothing_i = \theta - \alpha w_k \nabla_\theta L\left(\theta, D_{J_i}^{tr}\right).$$

Parameters are optimized by sampling tasks from $P(J)$—associated with $f_{\varnothing_i}$.

$$\min_\theta \sum_{J_i \sim p(J)} L\left(\varnothing_i, D_{J_i}^{\text{test}}\right) = \min_\theta \sum_{J_i \sim p(J)} L\left(\theta - \alpha w_k \nabla_\theta L\left(\theta, D_{J_i}^{tr}\right), D_{J_i}^{\text{test}}\right). \quad (14)$$

The goal of the optimization process is to use the updated parameters to calculate the outer layer updates. Let β be the learning rate in the inner layer. The parameters' heredity during the optimization process is

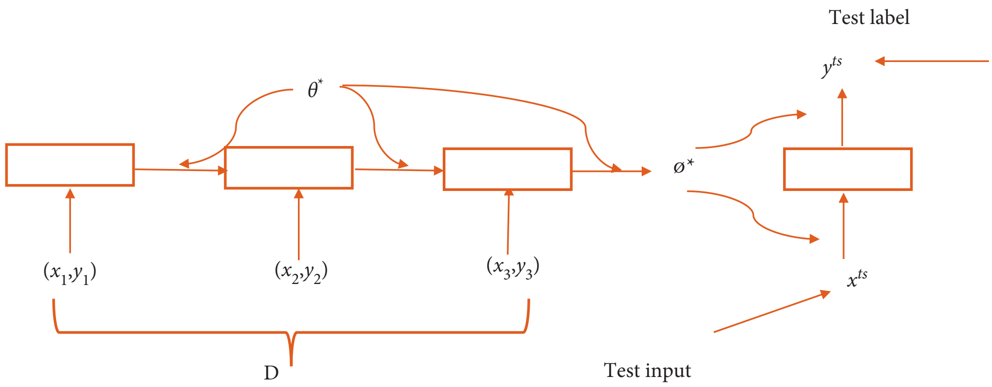

Figure 1: A simplified interpretation of metalearning processes.

$$\begin{aligned}
&\theta \longleftarrow \theta - \beta w_k \nabla_\theta \sum_{J_i \sim p(J)} L\left(\varnothing_i, D_{J_i}^{\text{test}}\right),\\
&\varnothing \longleftarrow \varnothing - \eta w_k \nabla_\varnothing L(\varnothing),\\
&L(\varnothing) = \sum_{n=1}^{N} l^n\left(\widehat{\theta}^n\right),\\
&\widehat{\theta} = \varnothing - \varepsilon w_k \nabla_\varnothing l(\varnothing).
\end{aligned} \tag{15}$$

The relationship between the total loss and the task loss during the parameters' optimization process is shown in Figure 2.

3.3. Theoretical Implementation. The implementation of EW-CACTUs-MAML includes two steps, which can be theoretically interpreted as "gene intelligence" (to highlight parameters' heredity).

First, in order to implement multistep gradients' updates, define an initial gene (that is, the initialization parameter). The multistep gradients' updates can be implemented through the calculation of the input training tasks' update genes. Second, continue to join the training data for each task and update genes. The optimal genes will be obtained in multiple gradient descents. Certainly, the parameters for a certain task may need to be updated several times to get the optimal result, as shown in Figure 3.

In order to simplify the genetic process, a future expectation for the best situation is that one update is enough for finding a gene, and during the whole process, only limited data with small samples are necessary for learning, as shown in Figure 4.

4. Experiments and Discussion

4.1. Performance of the Model. Two typical datasets, the Omniglot dataset and the Miniimagenet dataset, will be employed in this section. The Miniimagenet dataset has been widely used in the fields of metalearning and few-shot learning [31–37]. The famous original reference of the dataset is [37], where the matching networks for one-shot learning were presented to tackle a key challenge in machine learning—learning from a few examples. Up to now, Miniimagenet has become a benchmark dataset in the field of metalearning and few-shot learning [38–40]. The dataset contains 60000 colorful pictures with size 84×84 in 100 categories, including 600 samples in each category [41]. The Omniglot dataset contains 1623 handwritten characters from 50 different letters, which were drawn online by 20 different people with Amazon's Mechanical Turk [42]. Each image is paired with stroke data, and for the coordinate sequence $[x, y, t]$ of each stroke data, the time t is in milliseconds [43]. Omniglot is a benchmark dataset in the field of one-shot and few-shot learning [40, 44–49]. We utilize 60% of the Omniglot dataset as the training set and 40% of this dataset as the validation set, as shown in Figure 5.

According to the 300 iterations of the training and testing datasets in the deep cluster of the Omniglot dataset, the average value of the validation accuracy is 97.42%, which indicates that EW-CACTUs-MAML is robust on the Omniglot dataset.

4.2. Competitiveness and Practicability. The performance of CACTUs-MAML on the Omniglot dataset is shown in Figure 6, including details for the training process and validation process. According to the 300 iterations of the training and testing datasets in the deep cluster of the Omniglot dataset, the average value of the validation accuracy is 97.22%.

Comparing the dynamic curves of the train loss, train accuracy, validation loss, and validation accuracy of CACTUs-MAML with those of EW-CACTUs-MAML in Figure 5, we conclude that the proposed model is competitive with CACTUs-MAML. The comparisons of EW-CACTUs-MAML and CACTUs-MAML in the performance on the Omniglot dataset are shown in Table 1.

The loss of EW-CACTUs-MAML in validation is 0.20578947, which is less than the loss of CACTUs-MAML in validation. The accuracy of EW-CACTUs-MAML in validation is 97.42%, which is higher than the accuracy of CACTUs-MAML in validation. It must be noted that CACTUs-MAML can represent one most competitive model on the Omniglot dataset [50]. Consequently, these

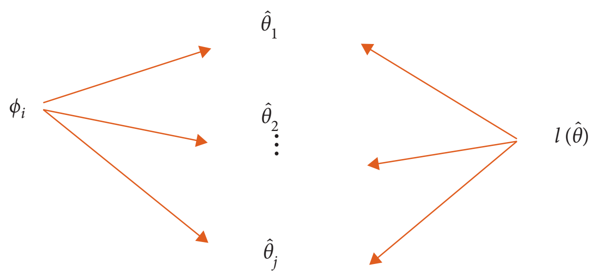

FIGURE 2: The relationship between $\varnothing_i$ and $\mathbf{l}(\hat{\theta})$.

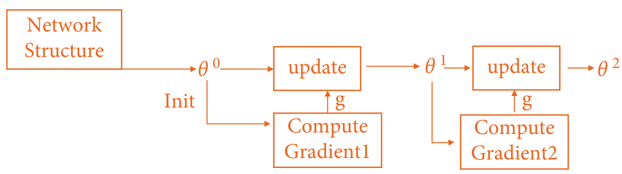

FIGURE 3: Gene updates in optimization processes.

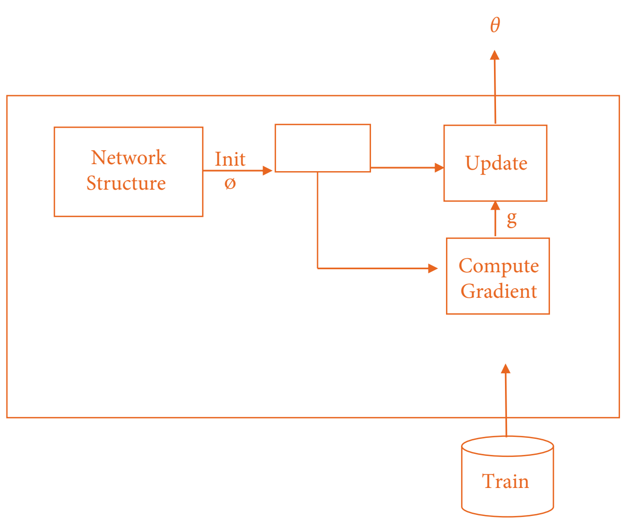

FIGURE 4: Simplified heredity: one update is enough for finding a gene.

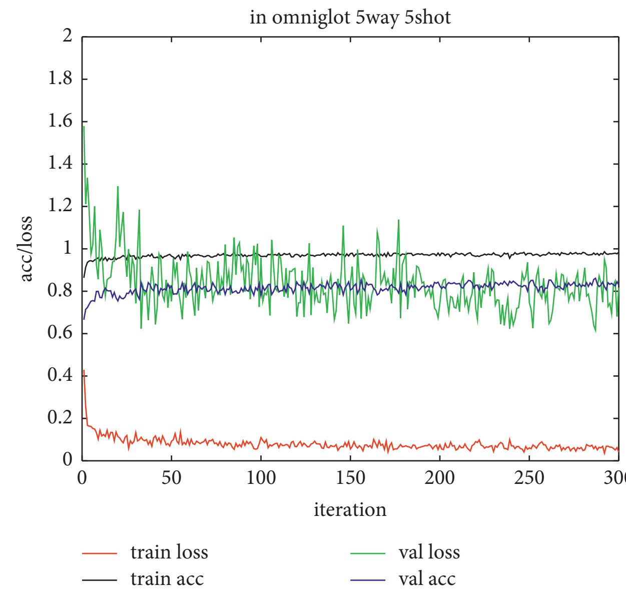

FIGURE 5: Performance of EW-CACTUs-MAML on Omniglot.

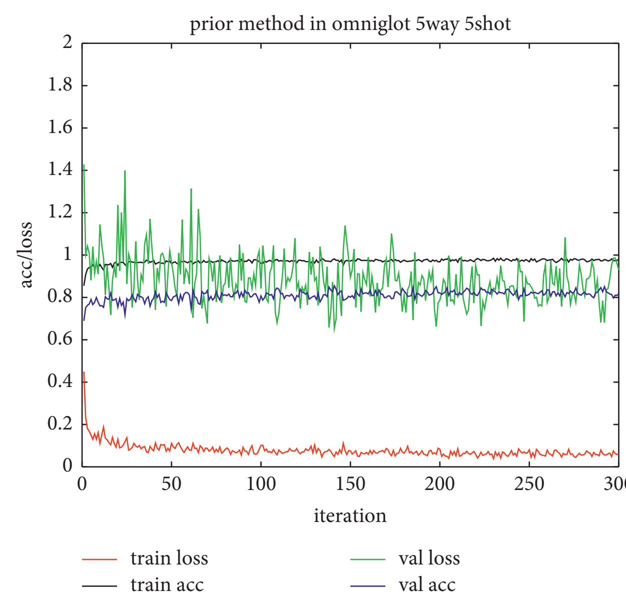

FIGURE 6: Performance of CACTUs-MAML on Omniglot.

results already demonstrate that the proposed model is competitive and practicable.

4.3. Uncertainty Analysis and Discussion. We tried to validate the model EW-CACTUs-MAML on another typical dataset Miniimagenet, but the size of this dataset is too big so that the computer sources were spent out before completing the performance of EW-CACTUs-MAML. Since we also want to validate the availability of the EW method in improving other metalearning systems, we then tried to improve another competitive metalearning system CACTUs-ProtoNets [50] as EW-CACTUs-ProtoNets. Fortunately, the computer sources are enough for performing the alternative model on both the Miniimagenet dataset. Details for the training and validation processes of CACTUs-ProtoNets and EW-CACTUs-ProtoNets on the Miniimagenet dataset are shown in Figure 7.

It must be pointed out that we utilized 80% of the Miniimagenet dataset as the training set and 20% of this dataset as the validation set for training/testing EW-CACTUs-ProtoNets and CACTUs-ProtoNets, which is similar with our strategy for training/testing EW-CACTUs-MAML and CACTUs-MAML. We explicitly compared the performance of the models EW-CACTUs-ProtoNets and CACTUs-ProtoNets on the Miniimagenet dataset, as shown in Table 2.

The Miniimagenet dataset is really challenging. The CACTUs-ProtoNets model is already most competitive on the Miniimagenet dataset, but the validation accuracy is still

Table 1: Results of experiments with $k = 500$ for each partition.

Dataset	Algorithm	Val loss	Val acc (%)
Omniglot (5-way/5-shot)	EW-CACTUs-MAML	0.20578947	97.42
	CACTUs-MAML	0.20888889	97.22

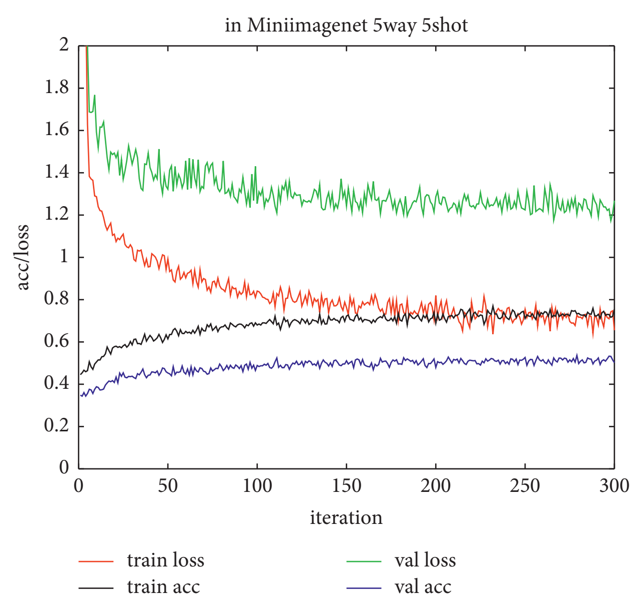

Figure 7: Performance of EW-CACTUs-ProtoNets and CACTUs-ProtoNets.

Table 2: Comparisons of the performance of EW-CACTUs-ProtoNets and CACTUs-ProtoNets on Miniimagenet with $k = 100$ for each partition.

Dataset	Algorithm	Train loss	Train acc (%)	Val loss	Val acc (%)
Miniimagenet (5-way/1-shot)	EW-CACTUs-ProtoNets	0.8724325	68.35	1.344944	48.52
	CACTUs-ProtoNets	0.8353898	68.86	1.3138521	48.78

less than 50% [50]. The low validation accuracy has not been improved after integrated with the EW method. As a cross-validation, performance of the model EW-CACTUs-ProtoNets on the Miniimagenet dataset revealed a challenge in the practical applications to complicated datasets [51–54]. The EW method can improve CACTUs-MAML, but it cannot improve CACTUs-ProtoNets.

One possible explanation for this is that CACTUs-MAML is a parameter-based model, while CACTUs-ProtoNets is a metric-based model. An unresolved issue is how to improve the performance of CACTUs-ProtoNets on the Miniimagenet dataset and other complicated datasets. Although the performance of our method on the Omniglot dataset implies the feasibility of the practical applications in optical character recognition (OCR), further validation on other engineering datasets is still necessary [55–59]. These should be next research priorities.

5. Conclusion

We apply few-shot mechanisms in completing task construction and propose a new method to optimize the previous algorithm, which is a competitive metalearning system. Entropy weight method is utilized to improve the prototypical network. The traditional gradient descent is in turn improved and utilized in the prototypical network to update the weights of the basic learner. The implementation of the proposed method is interpreted as "gene intelligence" to highlight parameters' heredity. The performance of EW-CACTUs-MAML indicates a robust prediction, which is competitive in the comparisons with CACTUs-MAML. Next research priorities are to further improve the performance of CACTUs-ProtoNets on the Miniimagenet dataset and to further validate the model on more complicated engineering datasets.

Data Availability

All the data utilized to support the theory and models of the present study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (41571299) and the High-Level Base-Building Project for Industrial Technology Innovation (1021GN204005-A06).

References

[1] P. B. Brazdil, C. Soares, and J. P. da Costa, "Ranking learning algorithms: using ibl and meta-learning on accuracy and time results," *Machine Learning*, vol. 50, no. 3, pp. 251–277, 2003.

[2] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[3] C. Giraud-Carrier, R. Vilalta, and P. Brazdil, "Introduction to the special issue on meta-learning," *Machine Learning*, vol. 54, no. 3, pp. 187–193, 2004.

[4] R. B. C. Prudêncio and T. B. Ludermir, "Meta-learning approaches to selecting time series models," *Neurocomputing*, vol. 61, no. 1, pp. 121–137, 2004.

[5] K. Zhang, Y. Han, J. Chen, Z. Zhang, and S. Wang, "Semantic segmentation for remote sensing based on rgb images and lidar data using model-agnostic meta-learning and partical swarm optimization," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 397–402, 2020.

[6] P. Kordík, J. Koutník, J. Drchal, O. Kovarik, M. Cepek, and M. Snorek, "Meta-learning approach to neural network optimization," *Neural Networks*, vol. 23, no. 4, pp. 568–582, 2010.

[7] A. Delgado and I. Romero, "Environmental conflict analysis using an integrated grey clustering and entropy-weight method: a case study of a mining project in Peru," *Environmental Modelling & Software*, vol. 77, no. 3, pp. 108–121, 2016.

[8] H. Abe and T. Yamaguchi, "Constructive meta-learning with machine learning method repositories," in *Innovations in Applied Artificial Intelligence*, B. Orchard, C. Yang, and M. Ali, Eds., Springer, Berlin, Germany, 2004.

[9] K. Zou, Z. Wang, and H. Ming, "A new initialization method for fuzzy c-means algorithm," *Fuzzy Optimization and Decision Making*, vol. 7, no. 4, pp. 409–416, 2008.

[10] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Long Beach, CA, USA, June 2019.

[11] N. Jiang and J. Wang, *Information Theory and Coding Theory*, Tsinghua University Press, Bejjing, China, 2010.

[12] S. Lin, *Topology of Metric Spaces and Functional Spaces*, Science Press, Bejjing, China, 2004.

[13] D. Petz, "Bregman divergence as relative operator entropy," *Acta Mathematica Hungarica*, vol. 116, no. 1-2, pp. 127–131, 2007.

[14] R. Singh, V. Bharti, V. Purohit, A. Kumar, A. K. Singh, and S. K. Singh, "Metamed: few-shot medical image classification using gradient-based meta-learning," *Pattern Recognition*, vol. 120, no. 1, pp. 108–111, 2021.

[15] S. Fusi, "Hebbian spike-driven synaptic plasticity for learning patterns of mean firing rates," *Biological Cybernetics*, vol. 87, no. 5-6, pp. 459–470, 2002.

[16] M. R. Baker and R. B. Patil, "Universal approximation theorem for interval neural networks," *Reliable Computing*, vol. 4, no. 3, pp. 235–239, 1998.

[17] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, no. 3, pp. 218–229, 2021.

[18] R. R. Yager and V. Kreinovich, "Universal approximation theorem for uninorm-based fuzzy systems modeling," *Fuzzy Sets and Systems*, vol. 140, no. 2, pp. 331–339, 2003.

[19] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 807–814, 1992.

[20] S. Tuna and B. Tunga, “A novel piecewise multivariate function approximation method via universal matrix representation,” *Journal of Mathematical Chemistry*, vol. 51, no. 7, pp. 1784–1801, 2013.

[21] D. Steinley, “K-means clustering: a half-century synthesis,” *British Journal of Mathematical and Statistical Psychology*, vol. 59, no. 1, pp. 1–34, 2011.

[22] K. V. Tilburg, “Identifying boosted objects with N-subjettiness and linear k-means clustering,” *Journal of High Energy Physics*, vol. 2011, no. 3, pp. 1–28, 2011.

[23] K. J. Kim and H. Ahn, “A recommender system using GA K-means clustering in an online shopping market,” *Expert Systems with Applications*, vol. 34, no. 2, pp. 1200–1209, 2008.

[24] M. Laszlo and S. Mukherjee, “A genetic algorithm that exchanges neighboring centers for k-means clustering,” *Pattern Recognition Letters*, vol. 28, no. 16, pp. 2359–2366, 2007.

[25] M. J. Brusco and J. D. Cradit, “A variable-selection heuristic for K-means clustering,” *Psychometrika*, vol. 66, no. 2, pp. 249–270, 2001.

[26] Z. Wei, G. Altun, R. Harrison, P. C. Tai, and Y. Pan, “Improved K-means clustering algorithm for exploring local protein sequence motifs representing common structural property,” *IEEE Transactions on NanoBioscience*, vol. 4, no. 3, pp. 255–265, 2005.

[27] M. Mahdavi, M. H. Chehreghani, H. Abolhassani, and R. Forsati, “Novel meta-heuristic algorithms for clustering web documents,” *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 441–451, 2008.

[28] A. M. Newman and J. B. Cooper, “AutoSOME: a clustering method for identifying gene expression modules without prior knowledge of cluster number,” *BMC Bioinformatics*, vol. 11, no. 1, p. 117, 2010.

[29] S. Hochreiter, A. S. Younger, and P. R. Conwell, “Learning to learn using gradient descent,” in *Proceedings of the International Conference on Artificial Neural Networks*, Springer, Vienna, Austria, August 2001.

[30] A. S. Younger, S. Hochreiter, and P. R. Conwell, “Meta-learning with backpropagation,” in *Proceedings of the International Joint Conference on Neural Networks*, IEEE, Washington, DC, USA, July 2001.

[31] C. Tarn and W. F. Zeng, “pDeep3: toward more accurate spectrum prediction with fast few-shot learning,” *Analytical Chemistry*, vol. 93, no. 14, pp. 5815–5822, 2021.

[32] F. Sung, Y. Yang, L. Zhang, and T. Xiang, “Learning to compare: relation network for few-shot learning,” 2017, https://arxiv.org/abs/1711.06025.

[33] Z. Li, F. Zhou, C. Fei, and L. Hang, “Meta-SGD: learning to learn quickly for few-shot learning,” 2017, https://arxiv.org/abs/1707.09835.

[34] S. Gidaris and N. Komodakis, “Dynamic few-shot visual learning without forgetting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT, USA, June 2018.

[35] S. Blaes and T. Burwick, “Few-shot learning in deep networks through global prototyping,” *Neural Networks: The Official Journal of the International Neural Network Society*, vol. 94, no. 10, pp. 159–172, 2017.

[36] Q. Cai, Y. Pan, T. Yao, C. Yan, and T. Mei, “Memory matching networks for one-shot image recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT, USA, June 2018.

[37] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” 2017, https://arxiv.org/abs/1606.04080.

[38] C. Liu, C. Xu, Y. Wang, L. Zhang, and Y. Fu, “An embarrassingly simple baseline to one-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, Seattle, WA, USA, June 2020.

[39] S. Doveh, E. Schwartz, C. Xue et al., “MetAdapt: meta-learned task-adaptive architecture for few-shot classification,” *Pattern Recognition Letters*, vol. 94, no. 10, pp. 130–136, 2021.

[40] H. Zhang, T. Zhan, and I. Davidson, “A self-supervised deep learning framework for unsupervised few-shot learning and clustering,” *Pattern Recognition Letters*, vol. 148, pp. 75–81, 2021.

[41] F. Wang, C. Li, Z. Zeng, and X. Ke, “Cornerstone network with feature extractor: a metric-based few-shot model for Chinese natural sign language,” *Applied Intelligence*, vol. 51, no. 5, pp. 7139–7150, 2021.

[42] S. R. Jiang, Y. R. Chen, J. C. Yang, C. Zhang, and T. Zhao, “Mixture variational autoencoders,” *Pattern Recognition Letters*, vol. 128, no. 12, pp. 263–269, 2019.

[43] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[44] Y. Zhang, M. Fang, and N. Wang, “Channel-spatial attention network for fewshot classification,” *PLoS One*, vol. 14, no. 12, Article ID e0225426, 2019.

[45] G. Karunaratne, M. Schmuck, M. L. Gallo, and G. Cherubini, “Robust high-dimensional memory-augmented neural networks,” *Nature Communications*, vol. 12, no. 1, 2021.

[46] Y. Cui, Q. Liao, D. Hu, W. An, and L. Liu, “Coarse-to-Fine pseudo supervision guided meta-task optimization for few-shot object classification,” *Pattern Recognition*, vol. 122, no. 6, Article ID 108296, 2021.

[47] Y. Xie, H. Wang, B. Yu, and Z. Chen, “Secure collaborative few-shot learning,” *Knowledge-Based Systems*, vol. 203, no. 7553, Article ID 106157, 2020.

[48] H. Xu, J. Wang, H. Li, D. Ouyang, and J. Shao, “Unsupervised meta-learning for few-shot learning,” *Pattern Recognition*, vol. 116, no. 6, Article ID 107951, 2021.

[49] J. Y. Lim, K. M. Lim, S. Y. Ooi, and C. P. Lee, “Efficient-PrototypicalNet with self knowledge distillation for few-shot learning,” *Neurocomputing*, vol. 459, no. 12, pp. 327–337, 2021.

[50] K. Hsu, S. Levine, and C. Finn, “Unsupervised learning via meta-learning,” 2018, https://arxiv.org/abs/1810.02334.

[51] A. Vailaya, A. Jain, and H. J. Zhang, “On image classification: city images vs. Landscapes,” *Pattern Recognition*, vol. 31, no. 12, pp. 1921–1935, 1998.

[52] M. Ohi, Y. Li, Y. Cheng, and T. Walz, “Negative staining and image classification — powerful tools in modern electron microscopy,” *Biological Procedures Online*, vol. 6, no. 1, pp. 23–34, 2004.

[53] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia, “A level set model for image classification,” *International Journal of Computer Vision*, vol. 40, no. 3, pp. 187–197, 1999.

[54] N. Orlov, L. Shamir, T. Macura, J. Johnston, D. M. Eckley, and L. G. Goldberg, “WND-CHARM: multi-purpose image classification using compound image transforms,” *Pattern Recognition Letters*, vol. 29, no. 11, pp. 1684–1693, 2008.

[55] A. Cheung, M. Bennamoun, and N. W. Bergmann, “An Arabic optical character recognition system using recognition-based segmentation,” *Pattern Recognition*, vol. 34, no. 2, pp. 215–233, 2001.

[56] S. Naz, K. Hayat, M. I. Razzak, M. W. Anwar, S. A. Madani, and S. U. Khan, “The optical character recognition of Urdu-

like cursive scripts," *Pattern Recognition*, vol. 47, no. 3, pp. 1229–1248, 2014.

[57] B. Braunecker, R. Hauck, and A. W. Lohmann, "Optical character recognition based on nonredundant correlation measurements," *Applied Optics*, vol. 18, no. 16, pp. 2746–2753, 1979.

[58] L. D. Jackel, D. Sharman, C. E. Stenard, B. I. Strom, and D. Zuckert, "Optical character recognition for self-service banking," *Bell Labs Technical Journal*, vol. 74, no. 4, pp. 16–24, 2013.

[59] F. Bensaali, R. Sotudeh, and X. Zhai, "Real-time optical character recognition on field programmable gate array for automatic number plate recognition system," *IET Circuits, Devices and Systems*, vol. 7, no. 6, pp. 337–344, 2013.