Research Article

A Joint Optimization Framework of the Embedding Model and Classifier for Meta-Learning

Zhongyu Liu ¹, Xu Chu,¹ Yan Lu,^{1,2} Wanli Yu,³ Shuguang Miao,⁴ and Enjie Ding ¹

¹National and Local Joint Engineering Laboratory of Internet Application Technology on Mine, China University of Mining and Technology, Xuzhou 221000, Jiangsu, China

²College of Information Engineering, Hainan Vocational University of Science and Technology, Haikou 570100, Hainan, China

³Institute of Electrodynamics and Microelectronics, University of Bremen, Bremen 28359, Germany

⁴School of Physics and Electronic Information, Huaibei Normal University, Huaibei 235000, Anhui, China

Correspondence should be addressed to Enjie Ding; enjied@cumt.edu.cn

Received 29 July 2021; Accepted 8 September 2021; Published 1 October 2021

Academic Editor: Punit Gupta

Copyright © 2021 Zhongyu Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The aim of meta-learning is to train the machine to learn quickly and accurately. Improving the performance of the meta-learning model is important in solving the problem of small samples and in achieving general artificial intelligence. A meta-learning method based on feature embedding that exhibits good performance on the few-shot problem was previously proposed. In this method, the pretrained deep convolution neural network was used as the embedding model of sample features, and the output of one layer was used as the feature representation of samples. The main limitation of the method is the inability to fuse low-level texture features and high-level semantic features of the embedding model and joint optimization of the embedding model and classifier. Therefore, a multilayer adaptive joint training and optimization method of the embedding model was proposed in the current study. The main characteristics of the current method include using multilayer adaptive hierarchical loss to train the embedding model and using the quantum genetic algorithm to jointly optimize the embedding model and classifier. Validation was performed based on multiple public datasets for meta-learning model testing. The proposed method shows higher accuracy compared with multiple baseline methods.

1. Introduction

The computing capacity of computers has recently developed and the amount of data has significantly increased. Deep learning has shown significant application in several tasks such as computer vision and natural language processing [1]. However, data collection and annotation of sample requires a lot of manual work, mainly data obtained from the industrial field, which is challenging to analyze [2]. Currently, the overfitting phenomenon appears easily with deep learning models based on supervised learning trained with small sample sizes [3]. Therefore, machine learning model design based on small samples has high application value. Small samples learning is also known as few-shot learning in the machine learning field. Notably, a machine learning model can achieve strong artificial intelligence [2].

Meta-learning is mainly used to model for few-shot learning problems. Meta-learning can learn new tasks quickly with the purpose of training the model to learn to learn. The traditional meta-learning methods can be divided into four categories including: methods based on data augmentation, methods based on metric learning, methods based on strong generalization of initialization, and methods based on parameter optimization. Their characteristics are explored in detail in the following section.

The method based on data augmentation is mainly used to expand the data volume and increase the number of samples. This process prevents the model from premature fitting in the training process and improves the generalization performance [4]. However, the number of samples for the deep neural network is often too large, and this method cannot meet the need of model training alone [5].

Therefore, the method is conjoined with other methods [6]. The metric-based method identifies small samples by calculating the feature vectors between samples and determining the feature distance between different samples [7]. This method mainly focuses on the designed similarity measurement [8]. Notably, the performance of the model is low if the samples have a close feature distance, mainly in the classification of fine-grained samples. The method based on external memory introduces the external memory module and achieves fast encoding of new information through the long-term and short-term memory functions. The main limitation of this model is that the efficiency of information filtering and classification storage is low and the number of external memory network parameters is too large, thus leading to an increase in training difficulty. The method based on initialization is mainly used to optimize the loss function which refers to the learning effect obtained by the expected gradient updating of a few steps. The training process comprises two parts: the inner cycle and the outer cycle [9]. Therefore, the model can quickly fit the distribution of multitask data and not the distribution of single task features. The gradient descent of the inner cycle in this method mainly depends on the setting of the learning rate and requires significant manual intervention.

In addition to the meta-learning methods described above, a new method was proposed and verified by [10]. This method comprises two stages. Firstly, a good feature embedding space is sought to map small samples to feature space. Second, a classifier is trained using these embedding features to quickly determine the feature distribution of samples and then achieve the inference of small samples.

The new method makes use of supervised learning to train a deep neural network to represent the features of the samples. In addition, it explores the performance of small samples embedded in different layers of the network on the classifier in the second stage of training. The characteristics of different layers of the deep convolutional neural network comprise different information. The bottom layer comprises high amounts of low-level visual information, whereas the top layer comprises semantic information [11, 12]. Low-level visual information and semantic information are important in visual recognition tasks. Therefore, the fusion of the features output by different layers of the CNNs (convolutional neural networks) is recommended for the effective embedding of the samples into the feature space. An adaptive hierarchical weighted loss training method of deep CNNs can optimize the feature representation of samples as presented in the current study.

The main contributions of the current study are presented as follows:

A joint optimization strategy of embedding model and meta-learner classifier is proposed. The embedding model is trained by adaptive multilevel loss, and the embedding model is optimized by combining results of the meta-learning classifier.

A quantum genetic algorithm is adopted to optimize the algorithm. The algorithm optimizes the weight of the hierarchical loss function of the embedding model

and optimizes the joint feature expression for the classifier.

Experimental results based on multiple meta-learning datasets indicate that the proposed method is effective and reliable.

The study explores related studies and summarizes related methods of meta-learning. In addition, the proposed method is illustrated in detail. The method is then verified through experiments and the findings are presented.

2. Related Studies

2.1. Meta-Learning Based on Measurement. The goal of measurement of learning [13] is to obtain a paired similarity measurement function, in which similar sample pairs have a high similarity score, while dissimilar sample pairs have a low similarity score. Meta-learning methods adopt measurement learning strategies. Koch et al. [7] proposed a twin network that calculates the similarity between the query sample and all single-annotated samples. Hoffer and Ailon [14] designed a ternary network, which adopted positive and negative sample rules and predicted the category of the sample by calculating the distance between the query sample and the positive and negative samples. Vinyals et al. [15] designed a matching network based on memory and attention mechanisms, which can quickly learn the distribution of features of training samples. Snell et al. [16] used a prototype network to predict the category of the sample by learning the prototype points and then calculating the distance between the new sample and the prototype points. In addition, Sung et al. [17] proposed a relational network in 2018, in which the similarity between samples was calculated by embedding modules and relationship modules to complete the prediction of categories of new tasks.

2.2. Meta-Learning Based on Optimization. A previous study reported a meta-learning model based on optimization LSTM (Long Short-Term Memory) with hidden units in the method set as learners' parameters [18]. Meta-learners can adjust their learning rate based on the responses of different spatiotemporal models so as to train the spatiotemporal models faster. Deep learning usually adopts back-propagation and gradient descending methods to optimize massive parameters in the network. The feature expression performance of deep neural networks depends on a large amount of data. A meta-learning method based on the strong generalization of initialization parameters was previously reported [9]. The goal of the optimization of the network parameters of MAML (model-agnostic meta-learning) is to separate the gradient descent and back-propagation from the used data, thus obtaining better generalization performance. Several methods have been used to improve MAML, such as combining the hidden space method [19] and Bayesian prior [20] and optimizing the gradient descent process [21]. Moreover, MAML can be used for a series of models based on gradient descent,

including classification, regression, and reinforcement learning owing to its openness and flexibility [22].

2.3. Meta-Learning Based on Data Expansion. A simple and general data enhancement framework called MetaGan was proposed previously [4]. This method mainly defined a tighter decision boundary for the model by distinguishing real data from virtual data with the aim of improving the feature extraction ability of the model. A previous study proposed a feature “analogy” method that divides the model into two parts: representation learning and small sample learning [23]. In the presentation learning stage, the learner extracts accurate feature representations on the base classes containing a large amount of data, whereas, in the small sample learning stage, the learner learns the classifier on the joint space of the new type containing a small amount of data and the previous base types. A data generation structure has been previously reported that mainly uses CycleGan as the generator of the new category of data and adds noise to the new data to make its distribution more diversified [24].

3. Methods

The current study reviews the meta-learning method based on feature embedding and then illustrates the algorithm framework proposed in the current work. The algorithm framework mainly comprises an adaptive hierarchical loss function optimization stage, an adaptive feature fusion stage, and the original classifier training evaluation and testing stage. The method is described in detail in the following section.

The datasets of meta-learning tasks can be divided into meta-training sets, meta-evaluation sets, and meta-testing sets. The traditional meta-learning method constructs the learning task based on the training set, which can be represented by a binary set $\{D^{\text{train},s}, D^{\text{train},q}\}$. The binary set then forms the support set and the query set. The support set and the query set normally have a few small samples, and this form is called N-way-K-shot. A basic learner is defined as follows: $\mathcal{A}: y_* = f_\theta(x_*)$, in which $*$ represents S and Q. $\mathcal{A}$ learns based on the training set $\{D^{\text{train},s}, D^{\text{train},q}\}$ to gain strong learning ability. A relative support set and query set should be constructed on the evaluation set or test set $\{D^{\text{val},s}, D^{\text{val},q}\} \{D^{\text{test},s}, D^{\text{test},q}\}$ to test the learning performance on the new task of $\mathcal{A}$. A new meta-learning method is proposed by a previous study [10], which defines an embedding mapping to map the features of samples into the embedding space. The training tasks of basic learners are then transformed into

$$\begin{aligned} \theta &= A(D_j^{\text{train}}, \phi) \\ &= \arg \min_{\theta} L^{\text{base}}(D_j^{\text{train}}; \theta, \phi) + R(\theta), \end{aligned} \quad (1)$$

where L represents the loss function and R represents the regular term. A good embedding model in the algorithm is aimed at high accuracy. The performance of the embedding model and meta-learners can be determined by finding the error between the two using the test datasets.

3.1. Algorithm Framework. The output features of different layers of CNNs have unique characteristics. The output in the top layer of CNNs comprises abundant semantic information, whereas the convolutional layer near the input comprises abundant low-level visual information. An embedding model and basic learner joint optimization algorithm are used to effectively fuse and utilize the information of the different layers. The algorithm framework is presented in Figure 1. The framework mainly comprises two parts: the optimization module of the adaptive hierarchical embedding model and the adaptive hierarchical feature fusion module. The optimization goal of the embedding model is the same as that of the feature fusion model, and both aim at the accuracy of meta test. Therefore, the optimization of the two models uses the same fitness function, which makes the modules of different stages evolve in the same direction. In addition, the optimization process of each embedding model is a supervised training process of a convolutional neural network. The training of the neural network usually takes a lot of time, so parallel programming is used to set the optimization process of each embedding model as a process. According to the graphics card equipped with the computer and its performance, n processes can be run at the same time, that is, n neural network models can be trained at the same time, which will greatly improve the efficiency of the algorithm. Compared with serial programming, it will effectively save a lot of time. The framework below describes the algorithm flow of the two modules in detail.

3.2. Quantum Genetic Algorithm. Encoding of chromosomes in the quantum genetic algorithm (QGA) uses the probability range of qubits in quantum physics, such that each chromosome represents multiple states. Quantum gates and revolving gates in quantum physics are used to update chromosomes to achieve the purpose of optimizing the population. An advantage of quantum genetic algorithm is its quantum parallelism, which improves the search ability of population size and increases the probability of obtaining the optimal solution compared with the traditional genetic algorithm [25].

Qubits are the basic storage units in quantum computing and are different from the bits referred to in existing computers; however, they are a two-state system. The two states in this case refer to two independent states in quantum computing: the 0 state and the 1 state. Borrowing the Dirac symbol “ $|\rangle$ ”, $|0\rangle$ and $|1\rangle$ are used to represent the spin-down state (0 state) and the spin-up state (1 state), respectively. A quantum state is the superposition of these two states, which can be expressed as a linear combination of the two states as follows:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (2)$$

where α and β represent the probability amplitudes of $|0\rangle$ and $|1\rangle$, respectively, and they must meet the conditions:

$$|\alpha|^2 + |\beta|^2 = 1, \quad (3)$$

where $|\alpha_i|^2$ indicates the probability of $|0\rangle$ state and $|\beta_i|^2$ indicates the probability of $|1\rangle$ state. This implies that when

FIGURE 1: Joint optimization framework of the embedding model and classifier based on quantum genetic algorithm.

the probability of state 0 is 1, the probability of state 1 must be zero. Conversely, when the probability of state 1 is 1, the probability of state 0 is 0. The amplitude probability $[\alpha, \beta]^T$ represents the chromosome of the qubit. The set of all chromosomes is expressed as $P = \{p_1, p_2, \dots, p_n\}$, where n represents the size of the total chromosomes. A quantum chromosome consisting of qubits can be described as follows:

$$p_i = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \beta_3 & \dots & \beta_m \end{bmatrix}, \quad (4)$$

where m represents the gene number of individual chromosomes.

To produce excellent individuals with a high probability in the iterative process of the population, the quantum

genetic algorithm introduces a quantum revolving gate to realize the change of the population and abandons the selection, crossover, and mutation operations used in the genetic algorithm. The quantum revolving gate can increase the diversity of the population gene pool by adjusting the quantum state probability, such that the solution appears on the individuals with the highest adaptability. Therefore, the quantum revolving gate is very important in the quantum genetic algorithm. The expression of the quantum revolving gate is as follows:

$$R(\theta) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix}. \quad (5)$$

The iterative process of the quantum revolving gate is expressed as follows:

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = R(\theta) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}. \quad (6)$$

Here, $[\alpha_i, \beta_i]^T$ represents the i -th qubit of the current chromosome, $[\alpha'_i, \beta'_i]$ indicates the state of the qubit after rotation, θ_i represents the rotation angle, and the direction and value of θ_i is set by adjustment strategy. The adjustment strategy is presented in Table 1.

$\theta_i = s(\alpha_i, \beta_i) \Delta \theta_i$ represents the angle of rotation, whereas $s(\alpha_i, \beta_i)$ represents the direction of rotation and $\Delta \theta_i$ represents the increment of rotation angle, to obtain convergence direction and convergence speed of the algorithm. In addition, in the adjustment strategy of the revolving gate, x_i represents the i -th gene of the current chromosome and $f(x_i)$ indicates the fitness of the individual. b_i represents the i -th gene in the current best chromosome, and $f(b_i)$ represents the fitness of the best individual. This method uses the fitness $f(b_i)$ of the best individual in the current population as a target value and compares the fitness $f(x_i)$ of the individuals in the population with it. If $f(x_i) > f(b_i)$, then the corresponding qubits in the individual are adjusted such that the probability amplitude (α_i, β_i) approaches x_i . Otherwise, if $f(x_i) < f(b_i)$, the corresponding quantum of the individual in the population bit is adjusted, such that the probability amplitude (α_i, β_i) approaches b_i .

3.3. Adaptive Hierarchical Loss Training Stage of the Embedding Model. The main purpose of the embedding model is to obtain the feature vector of the sample by mapping the original sample to the feature space. The meta-learner then learns the feature distribution of the sample based on the feature vector. Therefore, the performance of the embedding model is very important. The adaptive multilevel loss training method is adopted to train the embedding model to improve the feature representation performance of different layers of CNNs. The trained model is fused by a multiscale adaptive feature fusion algorithm. The framework of the adaptive multilayer loss training algorithm is shown in Figure 2.

The formal expression of the adaptive multilayer loss training algorithm is as follows: let F be any lightweight network, which normally has L stages. The output feature map from any intermediate stage is represented by $F^l \in \mathbb{R}^{H_l \times W_l \times C_l}$, where H_l, W_l, C_l represent the height, width, and number of channels of the feature map at l -th stage, and $l = \{1, 2, \dots, L\}$. The aim is to impose classification losses on the feature map extracted at different intermediate stages. The feature map of each stage performs the average pooling operation to obtain the feature vector f^l of the stage:

$$f^l = \text{Pool}_{\text{avg}}(F^l). \quad (7)$$

The Dot product similarity of the class prototype vector at this stage is then calculated to obtain the logits of the class, and finally, the loss function at this stage is obtained as shown in Figure 2:

$$\begin{aligned} \text{loss}^l &= \ell^l_{\text{Softmax}}(f^l) \\ &= -\log \frac{e^{f^l, y}}{\sum_{k=1}^C e^{f^l, k}}. \end{aligned} \quad (8)$$

Loss functions of all stages are weighted and summed to obtain the loss function of the network as follows:

$$\text{loss} = \sum_{l=1}^L \omega^l \cdot \text{loss}^l. \quad (9)$$

3.4. Meta-Evaluation and Meta-Test. The trained embedding model is used to extract the features of the samples in the meta-evaluation set and meta-test set. To make full use of the features of different layers of the embedding model, the method of feature fusion is used to represent the features of the samples. Several fusion methods have been explored. In the current study, the method of feature vector Mosaic was used as follows:

$$F^{\text{fuse}} = \text{concat}(\gamma^1 f^1, \gamma^2 f^2, \dots, \gamma^n f^n). \quad (10)$$

The fused input classifier is trained or tested, and the calculation formula is expressed as follows:

$$z^{\text{predict}} = \text{classifier}(F^{\text{fuse}}). \quad (11)$$

The fitness function of the quantum genetic algorithm is expressed as follows:

$$\text{fitness}(m_i) = g(x),$$

$$g(x) = \frac{\# \text{ of correctly predicted samples}}{\text{total number of samples}} \times 100\%. \quad (12)$$

All models are tested using the meta-test set when the optimization of the embedding model and feature fusion method is completed. The process of the meta-testing stage is the same as that of meta-evaluation. Formulas (10) and (11) are used for meta-testing. The algorithm flow is shown in Figure 3. The next section describes the details of meta-testing.

3.5. Meta-Classifer. Meta-learners are an important part of meta-learning. Meta-learners are referred to as meta-classifiers in the current study. Several models are available for classification tasks, such as nearest neighbor algorithm, linear regression, logistic regression, and support vector machine. The current study mainly uses the widely used logistic regression model for meta-classification. Logistic regression and linear regression are similar to the generalized linear model. Logistic regression assumes that the dependent variable y follows a Bernoulli distribution, whereas linear regression assumes that the dependent variable y follows a Gaussian distribution. Therefore, there are several similarities between logistic regression and linear regression. However, logistic regression introduces

TABLE 1: Quantum rotation gate adjustment strategy [26].

x_i	b_i	$f(x_i) \geq f(b_i)$	$\Delta\theta_i$	$s(\alpha_i, \beta_i)$			
				$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	False	0	—	—	—	—
0	0	True	0	—	—	—	—
0	0	False	δ	+1	-1	0	± 1
0	0	True	δ	-1	+1	± 1	0
1	0	False	δ	-1	+1	± 1	0
1	0	True	δ	+1	-1	0	± 1
1	1	False	0	—	—	—	—
1	1	True	0	—	—	—	—

FIGURE 2: Adaptive hierarchical multistage loss training framework for the embedding model.

FIGURE 3: Adaptive hierarchical feature fusion method for meta-evaluation and meta-test.

nonlinear factors through the Sigmoid function; thus, it can easily handle the 0/1 classification problem. If the Sigmoid mapping function is removed, the logistic regression

algorithm is similar to linear regression. This implies that logistic regression is theoretically supported by linear regression.

4. Experiments

4.1. Datasets. MiniImageNet: this is a dataset extracted from ImageNet by Vinyals et al. of the Google DeepMind team. This dataset contains a total of 100 categories and a total of 60,000 color images. Each category has 600 samples, and the size of each image is 84×84 pixels. The DeepMind team used the MiniImageNet dataset for small sample learning for the first time, and since then MiniImageNet has become the benchmark dataset for meta-learning and small sample research.

CIFAR-FS: the CIFAR-FS dataset refers to the CIFAR100 FEW-SHOTS dataset and is derived from the CIFAR100 dataset. It comprises 100 categories, 600 images per category, and a total of 60,000 images. For application purposes, it is divided into a training set (64 kinds), verification set (16 kinds), and test set (20 kinds) and the image size is unified as 32×32 pixels.

FC100: a few-shot CIFAR100 dataset. This dataset is similar to the CIFAR-FS dataset and is derived from the CIFAR100 dataset. It comprises 100 categories with 600 images for each category and a total of 60,000 images. The difference between CIFAR-FS and FC100 is that FC100 is not classified by category, but by Superclass. FC100 comprises 20 superclasses in total, including 12 superclasses in the training set, 4 superclasses in the verification set, and 4 superclasses in the testing set.

4.2. Setup. The PyTorch deep learning architecture was used to verify the performance of the proposed algorithms on meta-learning tasks [27]. The experiment was performed on an UltraLAB graphics workstation equipped with 192 GB memory and 8 NVIDIA GTX-2080 graphics processors, and each graphics card had 8 GB memory. The workstation was operated on Windows server operating system.

A ResNet-12 was chosen as the backbone. It included 4 residual blocks, and each block comprised 3 convolutional layers with 3×3 kernels. A 2×2 maximum pool layer was applied after the first 3 blocks. A global averaging pool layer is applied at the top of the fourth block to generate feature embedding. Drop-block was applied as a regularizer and the number of filters was varied from (64, 128, 256, 512) to (64, 160, 320, 640) as reported previously [28]. Consequently, the ResNet-12 presented in the current study is similar to that reported previously [28, 29]. The feature vectors of different layers were fused, and the feature fusion method and fusion weight were calculated based on formula (10) to obtain a good sample feature representation. Besides ResNet-12, SEResNet-12 is also used as the backbone network in the ablation experiments. The differences in residual block structure between the two backbone networks are shown in Figure 4 where (a) is the residual block of ResNet-12 and (b) is for the residual block of SEResNet-12. SEResNet-12 adds Squeeze and Excitation attention mechanism to ResNet-12, which is made up of two layers of full connection and one layer of pooling.

Optimization setup: SGD optimizer with a momentum of 0.9 and a weight decay of $5e-4$ was used in the current study. Each batch comprised 64 samples. The learning rate

was initialized as 0.05 and decayed with a factor of 0.1 by three times for all datasets, except for miniImageNet where it only decayed twice as the third decay had no effect. A total of 100 epochs were trained for miniImageNet and 90 epochs for both CIFAR-FS and FC100.

Random clipping, color dithering, and random horizontal flipping were used to extend the data for training the embedding model on the converted meta-training dataset as reported previously [28]. In the current study, N-way logistic regression basis classifier was trained in the meta-test phase. The implementation was applied in Scikit-Learn.

4.3. Results and Discussion. Experiments were conducted using the miniImageNet dataset. ResNet-12 was used as the backbone network. The random seed number of the experiment was fixed to ensure repeatability of the experiment. The weight of the loss function in formula (9) and the fusion weight of formula (10) are optimized using the accuracy of the meta-evaluation set as the fitness function of the quantum genetic algorithm (Table 2). In the 1-shot strategy, the method proposed in the current study achieved similar results to MetaOptNet [29]. The proposed method in the current study achieved better performance in the 5-shot strategy compared with MetaOptNet (Table 2). These findings indicate that the proposed method is effective and reliable.

The proposed method was further verified on FC100 and CIFAR-FS. The fixed random seed number strategy was used to conduct the experiment to ensure the reproducibility of the experiment. The experimental results showed that the proposed method in the current study performed better compared with existing methods, especially when using the FC100 dataset (Table 3). The current method showed higher performance compared with existing methods. This indicated that the joint optimization of the embedded model and the classifier and use of feature fusion strategies improves the classification accuracy of the new task.

4.4. Ablation Study

4.4.1. Comparison of Different Classifiers. A variety of different classifiers were used to verify the two meta-learning datasets, FC100 and CIFAR-FC, to explore the effect of classifiers on the proposed method. Several kinds of algorithms are available for classification tasks. The current study mainly adopted three methods: the nearest neighbor, logistic regression, and support vector machine (Table 4). Logistic regression classifier showed the best performance in classification. The accuracy of the logistic regression classifier using the meta-test set of FC100 data and the current method was best. This accuracy was higher compared with that of the nearest-neighbor classifier which had the worst performance. Therefore, the logistic regression classifier was selected for subsequent analysis.

The current study proposes an adaptive hierarchical loss embedding model training method. An adaptive hierarchical loss training method was adopted to verify the effectiveness of the method developed in the current study; the model was

FIGURE 4: Comparison of residual block structure between ResNet-12 and SEResNet-12. (a) Residual block of ResNet-12. (b) Residual block of SEResNet-12.

TABLE 2: Experimental results using the miniImageNet dataset and comparison of the method developed in the current study with the existing classical methods.

Model	Backbone	MiniImageNet 5-way	
		1-shot	5-shot
MAML[9]	32-32-32-32	48.70 ± 1.84	63.11 ± 0.92
Matching Networks [15]	64-64-64-64	43.56 ± 0.84	55.31 ± 0.73
IMP [30]	64-64-64-64	49.2 ± 0.7	64.7 ± 0.7
Prototypical Networks [16]	64-64-64-64	49.42 ± 0.78	68.20 ± 0.66
TAML [31]	64-64-64-64	51.77 ± 1.86	66.05 ± 0.85
SAML [32]	64-64-64-64	$52.22 \pm \text{n/a}$	$66.49 \pm \text{n/a}$
GCR [33]	64-64-64-64	53.21 ± 0.80	72.34 ± 0.64
KTN [34]	64-64-64-64	54.61 ± 0.80	71.21 ± 0.66
PARN [35]	64-64-64-64	55.22 ± 0.84	71.55 ± 0.66
Dynamic Few-Shot [36]	64-64-128-128	56.20 ± 0.86	73.00 ± 0.64
Relation Networks [17]	64-64-128-128	50.44 ± 0.82	65.32 ± 0.70
AdaResNet [37]	ResNet-12	56.88 ± 0.62	71.94 ± 0.57
TADAM [38]	ResNet-12	58.50 ± 0.30	76.70 ± 0.30
Shot-Free [29]	ResNet-12	$59.04 \pm \text{n/a}$	$77.64 \pm \text{n/a}$
TEWAM [39]	ResNet-12	$60.07 \pm \text{n/a}$	$75.90 \pm \text{n/a}$
MTL [40]	ResNet-12	61.20 ± 1.80	75.50 ± 0.80
Variational FSL [41]	ResNet-12	61.23 ± 0.26	77.69 ± 0.17
MetaOptNet [28]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46
Diversity w/Cooperation [42]	ResNet-18	59.48 ± 0.65	75.62 ± 0.48
LEO-trainval [19]	WRN-28-10	61.76 ± 0.08	77.59 ± 0.12
GEFS [10]	ResNet-12	62.02 ± 0.63	79.64 ± 0.44
Ours	ResNet-12	63.21 ± 0.78	81.55 ± 0.29

Average few-shot classification accuracies (%) with 95% confidence intervals on miniImageNet meta-test splits. a-b-c-d represents the size of the convolutional layer of the backbone network.

TABLE 3: Experimental results using CIFAR-FS and FC100 datasets and comparison of the current method with existing classic methods.

Model	Backbone	CIFAR-FC 5-way		FC100 5-way	
		1-shot	5-shot	1-shot	5-shot
MAML [9]	32-32-32-32	58.9 ± 1.9	71.5 ± 1.0	—	—
Prototypical Networks [16]	64-64-64-64	55.5 ± 0.7	72.0 ± 0.6	35.3 ± 0.6	48.6 ± 0.6
Relation Networks [17]	64-64-128-128	55.0 ± 1.0	69.3 ± 0.8	—	—
TADAM [38]	ResNet-12	—	—	40.1 ± 0.4	56.1 ± 0.4
Shot-Free [29]	ResNet-12	$69.2 \pm \text{n/a}$	$84.7 \pm \text{n/a}$	—	—
TEWAM [39]	ResNet-12	$70.4 \pm \text{n/a}$	$81.3 \pm \text{n/a}$	—	—
Prototypical Networks [16]	ResNet-12	72.2 ± 0.7	83.5 ± 0.5	37.5 ± 0.6	52.5 ± 0.6
MetaOptNet [28]	ResNet-12	72.6 ± 0.7	84.3 ± 0.5	41.1 ± 0.6	55.5 ± 0.6
GEFS [10]	ResNet-12	71.5 ± 0.8	86.0 ± 0.5	42.6 ± 0.7	59.1 ± 0.6
Ours	ResNet-12	72.62 ± 0.5	87.34 ± 0.6	45.18 ± 0.7	63.47 ± 0.6

Average few-shot classification accuracies (%) with 95% confidence intervals on CIFAR-FS and FC100. a-b-c-d represents the size of the convolutional layer of the backbone network.

TABLE 4: Ablation study on three benchmarks with ResNet-12 as the backbone network.

Model	CIFAR-FC 5-way		FC100 5-way	
	1-shot	5-shot	1-shot	5-shot
NN	69.58 ± 0.6	82.43 ± 0.6	43.3 ± 0.6	59.3 ± 0.6
LR	72.62 ± 0.5	87.34 ± 0.6	45.18 ± 0.7	63.47 ± 0.6
SVM	71.94 ± 0.6	84.57 ± 0.6	41.8 ± 0.7	59.6 ± 0.6

“NN” and “LR” represent the nearest neighbor classifier and logistic regression. “SVM” denotes support vector machine.

TABLE 5: Comparisons of different training strategies for the embedding model on CIFAR-FS and FC100.

Model	CIFAR-FC 5-way		FC100 5-way		MiniImageNet 5-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Ours simple	72.38 ± 0.6	86.67 ± 0.85	44.1 ± 0.73	60.8 ± 0.6	62.49 ± 0.55	79.91 ± 0.40
Ours	72.62 ± 0.5	87.34 ± 0.6	45.18 ± 0.7	63.47 ± 0.6	63.21 ± 0.78	81.55 ± 0.29

FIGURE 5: Activation map of selected results on the miniImageNet dataset using ResNet-12 as the base model. From left to right: Column 1 is original images; Column 2 is Grad-CAM visualizations for the model trained with standard cross entropy; Column 3 indicates Grad-CAM visualizations for the model trained with the currently proposed method.

TABLE 6: Comparisons of different backbones on CIFAR-FS and FC100.

Model	CIFAR-FC 5-way		FC100 5-way	
	1-shot	5-shot	1-shot	5-shot
ResNet-12	72.62 ± 0.5	87.34 ± 0.6	45.18 ± 0.7	63.47 ± 0.6
SEResNet-12	72.91 ± 0.85	87.59 ± 0.7	45.47 ± 0.63	63.65 ± 0.6

then divided into loss training method and traditional training method for comparison. The dividing loss method was used to fix the loss weight of different layers, which is set as 1 in the current study. The traditional neural network training method was used to add the loss function in the last layer of the network. Cross entropy was used as the calculation method for the loss function of all layers. Experiments were conducted on three different datasets, and the results showed that the embedding model of adaptive hierarchical loss proposed in the current study was optimal by comparing with other methods (Table 5). These findings indicate that the hierarchical loss function can improve the shallow layer feature expression performance of the convolutional neural network.

In addition, the feature map of the 4th layer of ResNet-12 was visually visualized for adaptive hierarchical loss training. The visualization method used was CAM, which is used to visualize the feature map output by the convolutional layer in the neural network in the form of the heat map. The red area represents the response area of the neural network neurons in this layer. The findings showed that the embedding model of adaptive hierarchical loss function training proposed in the current study had a larger response area compared with that of the traditional method (Figure 5). This finding indicates the effectiveness of the current method.

4.4.2. Comparison of Different Network Backbones. In the above experiment, ResNet-12 was used as the backbone network to make fair comparisons with other methods. To verify the influence of the backbone network on the proposed algorithm, SEResNet-12 was also applied in the experiments. The experimental results are shown in Table 6. Notably, SEResNet-12 had a better representation of the sample characteristics.

5. Conclusion

This study presents a joint optimization method in which the embedding model and classifier are used for meta-learning. Specifically, we applied a quantum genetic algorithm to optimize the hierarchical multi-loss weight of the embedding model and the weight of feature fusion. The classification accuracy of the meta-learning evaluation set was used as the fitness function, which effectively combines the embedded model and classifier. The performance of the proposed method was tested on three well-known public meta-learning datasets. It was found that the performance of the proposed method was superior to that of most existing baseline meta-learning models. In the future, we plan to carry out studies on the joint optimization method of semisupervised or unsupervised embedding model and classifier. In addition, we shall explore more efficient evolution strategies to improve the efficiency of the joint optimization method.

Data Availability

The data used to support the findings of this study have been deposited in the Dropbox repository (<https://www.dropbox.com/sh/6yd1ygtyc3yd981/AABVeEqzC08YQv4UZk7lNHvya?dl=0>). The research uses public datasets, so readers can find resources online.

com/sh/6yd1ygtyc3yd981/AABVeEqzC08YQv4UZk7lNHvya?dl=0). The research uses public datasets, so readers can find resources online.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by Research on Online Identification of Coal Gangue Based on Terahertz Detection Technology (Grant no. 52074273) and by the State Key Research Development Program of China (Grant nos. 2017YFC0804400 and 2017YFC0804401). This work was also funded by the Key Natural Science Research Project for Colleges and Universities of Anhui Province (Grant no. KJ2020A0027).

References

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] J. Vanschoren, "Meta-learning," in *Automated Machine Learning* Springer, Berlin, Germany, 2019.
- [3] A. Saxe, S. Nelli, and C. Summerfield, "If deep learning is the answer, what is the question?" *Nature Reviews Neuroscience*, vol. 22, no. 1, pp. 55–67, 2021.
- [4] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Sona, "MetaGAN: an adversarial approach to few-shot learning," in *Proceedings of the Neural Information Processing Systems Conference*, p. 8, Montréal, Canada, December 2018.
- [5] Y. Liu, Y. Zhou, X. Liu, F. Dong, C. Wang, and Z. Wang, "Wasserstein GAN-based small-sample augmentation for new-generation artificial intelligence: a case study of cancer-staging data in biology," *Engineering*, vol. 5, no. 1, pp. 156–163, 2019.
- [6] 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.
- [7] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proceedings of the ICML Deep Learning Workshop*, Lille, France, July 2015.
- [8] X. Li, L. Yu, C.-W. Fu, M. Fang, and P.-A. Heng, "Revisiting metric learning for few-shot image classification," *Neurocomputing*, vol. 406, pp. 49–58, 2020.
- [9] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the International Conference on Machine Learning*, pp. 1126–1135, PMLR, Sydney, Australia, August 2017.
- [10] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?" in *Proceedings of the Computer Vision–ECCV 2020: 16th European Conference*, Glasgow, UK, August 2020.
- [11] W. Guo, W. Li, and W. Gong, "Extended feature pyramid network with adaptive scale training strategy and anchors for object detection in aerial images," *Remote Sensing*, vol. 12, no. 5, 2020.
- [12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongi, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

- [13] Q. Yang, P. Yan, Y. Zhang et al., "Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1348–1357, 2018.
- [14] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proceedings of the International Workshop on Similarity-Based Pattern Recognition*, pp. 84–92, Springer, Copenhagen, Denmark, October 2015.
- [15] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 29, pp. 3630–3638, Barcelona, Spain, 2016.
- [16] J. Snell, K. Swersky, and R. Zemel, "Prototypical Networks for Few-Shot Learning," 2017, <https://arxiv.org/abs/1703.05175>.
- [17] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. Torr, and T. M. Hospedales, "Learning to compare: relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, Salt Lake City, UT, USA, June 2018.
- [18] S. Ravi and H. Larochelle, "Optimization as a Model for Few-Shot Learning," in *Proceedings of the ICLR 2016*, San Juan, Puerto Rico, May 2016.
- [19] A. A. Rusu, D. Rao, J. Sygnowski et al., "Meta-learning with latent embedding optimization," 2018, <https://arxiv.org/abs/1807.05960>.
- [20] J. Yoon, T. Kim, and O. Dia, "Bayesian model-agnostic meta-learning," in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS)*, Toronto, Canada, June 2018.
- [21] A. Nichol, J. Achiam, and J. J. a. p. a. Schulman, "On first-order meta-learning algorithms," 2018, <https://arxiv.org/abs/1803.02999>.
- [22] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel, "Continuous adaptation via meta-learning in nonstationary and competitive environments," in *Proceedings of the ICLR 2017 Conference*, Toulon, France, April 2017.
- [23] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3018–3027, Venice, Italy, October 2017.
- [24] H. Gao, Z. Shou, A. Zareian, H. Zhang, and S. F. Chang, "Low-shot learning via covariance-preserving adversarial augmentation networks," in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS)*, Montréal Canada, December 2018.
- [25] G. Zhang and H. Rong, "Real-observation quantum-inspired evolutionary algorithm for a class of numerical optimization problems," in *Proceedings of the 7th International Conference on Computational Science (ICCS 2007)*, pp. 989–996, Beijing, China, May 2007.
- [26] K. H. Han and J. H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," in *Proceedings of the 2000 Congress on Evolutionary Computation (CEC2000)*, pp. 1354–1360, La Jolla, CA, USA, July 2000.
- [27] N. Ketkar, "Introduction to pytorch," in *Deep Learning with Python*, Springer, Berlin, Germany, 2017.
- [28] 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*, pp. 10657–10665, Long Beach, CA, USA, June 2019.
- [29] A. Ravichandran, R. Bhotika, and S. Soatto, "Few-shot learning with embedded class models and shot-free meta training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 331–339, Seoul, Korea, November 2019.
- [30] K. Allen, E. Shelhamer, H. Shin, and B. J. Tenenbaum, "Infinite mixture prototypes for few-shot learning," in *Proceedings of the International Conference on Machine Learning*, pp. 232–241, PMLR, Long Beach, CA, USA, June 2019.
- [31] M. A. Jamal and G.-J. Qi, "Task agnostic meta-learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11719–11727, Seoul, Korea, November 2019.
- [32] F. Hao, F. He, J. Cheng, L. Wang, L. Cao, and D. Tao, "Collect and select: semantic alignment metric learning for few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8460–8469, Seoul, Korea, November 2019.
- [33] A. Li, T. Luo, T. Xiang, W. Huang, and L. Wang, "Few-shot learning with global class representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9715–9724, Seoul, Korea, November 2019.
- [34] Z. Peng, Z. Li, J. Zhang, Y. Li, G. J. Qi, and J. Tang, "Few-shot image recognition with knowledge transfer," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, pp. 441–449, Seoul, Korea, November 2019.
- [35] Z. Wu, Y. Li, L. Guo, and K. Jia, "PARN: position-aware relation networks for few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6659–6667, Seoul, Korea, November 2019.
- [36] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, Salt Lake City, UT, USA, June 2019.
- [37] T. Munkhdalai, X. Yuan, S. Mehri, and A. Trischler, "Rapid adaptation with conditionally shifted neurons," in *Proceedings of the International Conference on Machine Learning*, pp. 3664–3673, PMLR, Stockholm Sweden, July 2018.
- [38] B. N. Oreshkin, P. Rodriguez, and A. Lacoste, "TADAM: task dependent adaptive metric for improved few-shot learning," in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS)*, pp. 124–137, Montréal Canada, December 2018.
- [39] L. Qiao, Y. Shi, J. Li, Y. Tian, T. Huang, and Y. Wang, "Transductive episodic-wise adaptive metric for few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3603–3612, Seoul, Korea, November 2019.
- [40] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 403–412, Long Beach, CA, USA, June 2019.
- [41] J. Zhang, C. Zhao, B. Ni, M. Xu, and X. Yang, "Variational few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1685–1694, Seoul, Korea, October 2019.
- [42] N. Dvornik, C. Schmid, and J. Mairal, "Diversity with co-operation: ensemble methods for few-shot classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3723–3731, Seoul, Korea, October 2019.