

Received August 21, 2019, accepted October 3, 2019, date of publication October 8, 2019, date of current version October 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946264

# MPCE: A Maximum Probability Based Cross Entropy Loss Function for Neural Network Classification

YANGFAN ZHOU<sup>1</sup>, XIN WANG<sup>2</sup>, MINGCHUAN ZHANG<sup>1,3,4</sup>, JUNLONG ZHU<sup>1,3,4</sup>,  
RUIJUAN ZHENG<sup>1,3</sup>, AND QINGTAO WU<sup>1</sup>

<sup>1</sup>College of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

<sup>2</sup>Laboratory of Applied Brain and Cognitive Sciences, Postdoctoral Research Station, School of Business and Management, Shanghai International Studies University, Shanghai 200083, China

<sup>3</sup>Henan Qunzhi Information Technology Company Ltd., Luoyang 471003, China

<sup>4</sup>Guangzhou Xiangxue Pharmaceutical Company Ltd., Guangzhou 510663, China

Corresponding authors: Xin Wang (wangxin@shisu.edu.cn) and Mingchuan Zhang (zhang\_mch@haust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant U1604155, Grant 61602155, and Grant 61871430, in part by the Scientific and Technological Innovation Team of Colleges and Universities in Henan Province under Grant 20IRTSTHN018, in part by the China Postdoctoral Science Foundation under Grant 2018M630461, in part by the Science Foundation of Ministry of Education of China under Grant 19YJC630174, in part by the basic research projects in the University of Henan Province under Grant 19zx010, and in part by the Science and Technology Development Programs of Henan Province under Grant 192102210284.


**ABSTRACT** In recent years, multi-classifier learning is of significant interest in industrial and economic fields. Moreover, neural network is a popular approach in multi-classifier learning. However, the accuracies of neural networks are often limited by their loss functions. For this reason, we design a novel cross entropy loss function, named MPCE, which based on the maximum probability in predictive results. In this paper, we first analyze the difference of gradients between MPCE and the cross entropy loss function. Then, we propose the gradient update algorithm based on MPCE. In the experimental part of this paper, we utilize four groups of experiments to verify the performance of the proposed algorithm on six public datasets. The first group of experimental results show that the proposed algorithm converge faster than the algorithms based on other loss functions. Moreover, the results of the second group show that the proposed algorithm obtains the highest training and test accuracy on the six datasets, and the proposed algorithm perform better than others when class number changing on the sensor dataset. Furthermore, we use the model of convolutional neural network to implement the compared methods on the mnist dataset in the fourth group of experiments. The results show that the proposed algorithm has the highest accuracy among all executed methods.

**INDEX TERMS** Cross entropy, loss function, maximum probability, neural network classification, softmax.

## I. INTRODUCTION

In the last few years, multi-classifier learning has received significant attention in many fields. For example, fuzzy system [1], wireless networks [2], power delivery system [3], medical imaging [4], etc. Meanwhile, many methods have been proposed to solve multi-classifier problems. For instance, Support Vector Machine (SVM) [5], Decision Tree (DT) [6], Bayesian method [7], K-means [8], neural networks [12], etc. However, despite SVM, DT, Bayesian and K-means were wildly researched in the past years,

their ability to deal with nonlinear multi-classifier problems is always poor. Many practical classification problems are actually nonlinear due to the complexity of the real environment [9], [10], [11]. Moreover, neural networks are famous for their strong ability to deal with nonlinear multi-classifier problems. Furthermore, neural networks can represent high-dimensional parameters better than other methods due to their complex hidden layers. Therefore, in order to address the multi-classifier problems that are nonlinear or with high-dimensional parameters, neural networks have successfully been used in many hot fields of artificial intelligence, such as image classification [12], embedded computation [13], biomedical engineering [14], etc.

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Jin .

Moreover, neural networks always have higher classifier accuracy than other traditional methods [15], [16], [17].

However, despite neural networks can obtain an outstanding accuracy in multi-class classification problems, their training process is always a difficulty for researchers. In order to alleviate this difficulty, many methods have been proposed recently from different aspects. For instance, the acceleration of convergence speed for optimization algorithms [18], the structure optimization of neural networks [19], the improvement of activation functions [20] and the improved version of loss functions [21]. Among of them, the research of loss function is always a hotspot in many subjects, such as statistics [22], decision theory [23], neural networks [24], etc. The loss function maps values of variables onto a real number that intuitively represents some loss associated with the variables. Moreover, when dealing with the multi-classifier problems by neural networks, the associated loss is the difference of the output value of activation function and the true value of samples. Therefore, the choice of loss functions is closely related to the activation function. At present, softmax function has been widely used for output layers in multi-classifier learning because of its ability to convert output values into probabilities [25], [26].

Many loss functions have been utilized in the neural networks based on softmax activation function, such as Mean Square Error (MSE) loss function [27], Cross Entropy (CE) loss function [28], etc. However, the gradient of MSE tends to disappear when softmax is used as output layers in neural networks. Moreover, due to the back-propagation error of CE less than that of MSE at each iteration, CE has a faster convergence rate in these cases [29], [30]. Therefore, CE has become a popular loss function in problems of multi-class classification. Furthermore, many variants of CE has been proposed in the past few years. For instance, a visualized variant [30], a symmetric variant [31], an improved variant which is used in multi-scale convolutional neural network [32].

CE uses entropy to measure the differences between predictive distribution and true distribution, and it performs better than other loss functions and has a great flourish of studies. However, CE let the value of real class to 1 and let the values of other non-real classes to 0. This method may result in information redundancy of back-propagation error. In order to further improve the loss measure of CE, we propose a novel cross entropy loss function based on the maximum probability of the predictive distribution, called MPCE. We first determine the true class based on the true distribution. Then we choose the maximum probability to calculate the cross entropy of object variables. Finally, we update the object variables along the gradient direction. The main contributions of this paper are as follows:

- The proposed method reduces redundant information for each back-propagation error, which brings better benefits to the training of the objective function.
- We give the gradient derivation of the proposed method and show the result that the proposed method has less back-propagation error than CE at each iteration.

- We present the algorithm based on the proposed method. Moreover, the simulated experiments on six public datasets show that the proposed method has a faster convergence rate than CE, Taming Cross Entropy (TCE) and Accelerated Cross Entropy (ACE).
- The proposed method is simple in structure and easy to apply as same as the standard CE.

The rest of the paper is organized as follows. We review the related works in Section II, and present the multi-class classification problem in Section III. We give the gradient derivation and the algorithm design of the proposed method in Section IV. The simulated experiments on six public datasets are presented in Section V. Finally, we conclude the paper in Section VI.

## II. RELATED WORK

The training process is an important work for machine learning. And the training process of many algorithms in machine learning requires massive of labeled data. However, the massive labeled data, especially large-scale and high-dimension data, will take lots of cost that be labeled by people. Therefore, many researchers focus on improving the training efficiency. The design of the loss function with better performance is one of the effective ways. For example, the Mean Square Error (MSE) function was used as loss function in neural network classification. However, the training process of this case is very slow. To this end, MSE is replaced by CE in neural network classification that accelerates the training process.

CE loss function originates from information theory, which measures the loss between the true distribution and the predicted distribution of the current model. CE, as a loss function, has been widely used in the training of model parameters for machine learning. For instance, convolutional neural network [32], random tree [33], clustering [34], etc.

The CE loss function is popular mainly because of its excellent performance in terms of multi-class classification accuracy. CE is a calibrated loss and requires well-behaved probability estimates. However, as the classification accuracy increases, the calibration of the classifier using CE deteriorates. The reason of this case is caused by outliers. For this reason, many robust variants of CE have been proposed. For example, Ghosh *et al.* studied some robust loss functions under label noise for deep neural networks [35], and Martinez *et al.* presented a robust derivative of the standard CE used in deep learning for classification tasks [38].

Furthermore, training speed is another research point in machine learning based on CE loss. The common method to speed up training is stochastic chooses a subset of samples at each iteration. However, this method requires a large number of samples to mitigate errors. For this reason, Bengio and Audry [39] proposed used adaptive importance sampling to accelerate training. Moreover, Blanc and Rendle [40] proposed an adaptive sampling method based on kernel for classification problems using CE loss. In addition, to speed up the convergence rate,

George and Shalabh [41] used cross entropy method in the reinforcement online learning.

Despite the robust variants of CE could obviously eliminate the influence of outliers, and the adaptive sampling variants of CE could speed up the training, however, they often at the expense of classification accuracy. The accuracy is always the most important point to the classification problems. For this reason, we propose a novel cross entropy loss function to improve the classification accuracy. The maximum probability in predictive distribution is used to improve the standard CE, therefore, a more accurately loss of variables is obtained at each iteration. The proposed method thereby obtains the optimal solution earlier.

### III. MULTI-CLASS CLASSIFICATION

In problems of multi-class classification, let  $\mathbb{S}$  denote a sample space and  $\mathcal{L}$  denote a limited set of labels, where  $\mathcal{L} = \{l_1, l_2, \dots, l_m\}$ ,  $m > 2$ . Therefore, the mapping relationship of a sample  $\mathbf{x} \in \mathbb{S}^N$  to label set  $\mathcal{L}$  is many-to-one, i.e., a sample can not have more than one label, but different samples can have the same label. A multi-class classification task is to predict the class of a sample from multi classes. Next, we introduce how neural networks accomplish multi-class classification tasks.

As shown in FIGURE 1. A general neural network based on softmax activation function has three layers, i.e. input layer, hidden layer and output layer. A softmax function transform the numerical results of neural network to the output of probability. Moreover, the predicted class is decided by the max probability in the output. Then, at each training iteration, cross entropy function is used to measure the difference between the predicted class and the true label. Whereafter, the difference (i.e., error) is propagated back to the hidden layer to adjust the weights. Let  $w_{ij}$  denote the connection weight of the  $i$ -th neuron to the  $j$ -th neuron. Moreover,  $W$  denotes the weight matrix which is consisted of  $w_{ij}$ , and  $w_i$  denotes the  $i$ -th row vector of matrix  $W$ . Therefore, the output  $\mathbf{z} = W\mathbf{x}$ . However, the output  $\mathbf{z}$  is not normalized. In order to solve this problem, softmax function has received significant attention in multi-class classification, which is used be the output layer. The output layer of neural networks contains  $m$  cells, and each cell corresponds to a label. Moreover,  $y_i$  denotes the  $i$ -th coordinate value of vector  $\mathbf{y}$ , and  $\tilde{y}_i$  denotes the  $i$ -th coordinate value of vector  $\tilde{\mathbf{y}}$ . Softmax maps output results to the interval  $(0, 1)$ , in which sum of all outputs is 1. Therefore, it converts the classification problem into the form of probability, which provides intuitive choices for the final decision. The softmax is shown as follows,

$$y_i = \frac{\exp(z_i)}{\sum_{j=1}^m \exp(z_j)}, \quad (1)$$

where  $i \in \{1, 2, \dots, m\}$ , and the neural output  $z_i = \sum_j w_{ij}x_{ij}$ . Therefore, we have

$$\sum_{i=1}^m y_i = 1. \quad (2)$$

According to the mentioned above, the main target of neural network is training weight matrix  $W$ . For this reason, many gradient descent based optimization methods have been used in these cases. The value of weights is adjusted by back-propagation of each iteration error. Moreover, the error is generated by a loss function (or cost function). Let  $f^t(W)$  denote the loss function at iteration  $t$ . Then, we have an optimization problem,

$$\min \sum_{t=1}^T f^t(W), \quad (3)$$

where  $T$  is the total number of training iterations. In order to obtain a optimal solution, optimization algorithms search along the gradient direction. Moreover, when considering the  $i$ -th output cell, the gradient of its loss function at  $t$ -th iteration can be expressed in the following mathematical form,

$$\nabla f^t(w_i) = \frac{\partial f^t(w_i)}{\partial w_i}. \quad (4)$$

Therefore, we have update strategy of the weight vector  $w_i$  in gradient descent based algorithms,

$$w_i^t = w_i^{t-1} - \eta \nabla f^t(w_i^{t-1}), \quad (5)$$

where  $\eta$  denotes the learning rate.

#### A. CROSS ENTROPY

In the next step, we will introduce the formula form of CE loss function and its gradients when using in multi-classifier learning. First, the CE function is as follows,

$$\begin{aligned} f^t(W) &= - \sum_{i=1}^m \tilde{y}_i \log(\text{softmax}(w_i x)) \\ &= - \sum_{i=1}^m \tilde{y}_i \log(y_i), \end{aligned} \quad (6)$$

where  $m$  denotes the total number of classes,  $y_i$  denotes the  $i$ -th prediction class of MPCE, and  $\tilde{y}_i$  denotes the  $i$ -th true class of training samples. Moreover, the softmax function is defined in Eq. (1).

Next, we give the gradient derivation process of the  $i$ -th output cell. From Eqs. (4) and (6), and according to the chain rule, we have

$$\nabla f^t(w_i) = - \frac{\partial f^t(w_i)}{\partial y_j} \frac{\partial y_j}{\partial z_i} \frac{\partial z_i}{\partial w_i}, \quad (7)$$

where

$$\begin{aligned} \frac{\partial f^t(w_i)}{\partial y_j} &= \sum_{j=1}^k \frac{\partial(-\tilde{y}_j \log y_j)}{\partial y_j} \\ &= - \sum_{j=1}^k \tilde{y}_j \frac{1}{y_j}, \end{aligned} \quad (8)$$

and  $\mathbf{x} \in \mathbb{S}^N$  is normalized before training process, therefore, we have

$$\frac{\partial z_i}{\partial w_i} = \sum_{j=1}^k x_{ij} = 1, \quad (9)$$

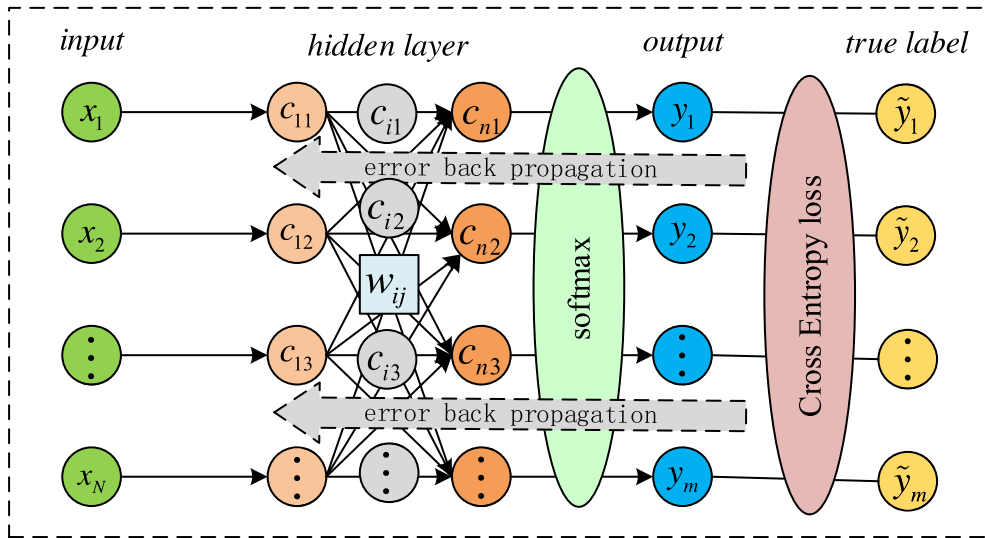


FIGURE 1. The structure of neural network in which softmax is used as activation function and CE is loss function.

where  $k$  is the total number of the connection units of cell  $i$ . Then, we calculate the  $\frac{\partial y_j}{\partial z_i}$  in Eq. (7). First, we should consider two cases of connection units, i.e. one is that connection unit of cell is itself, the other is that connection unit of cell is other cells. If  $i = j$ , then

$$\begin{aligned} \frac{\partial y_i}{\partial z_i} &= \frac{\partial \left( \frac{\exp(z_i)}{\sum_{k=1}^m \exp(z_k)} \right)}{\partial z_i} \\ &= \frac{\sum_{k=1}^m \exp(z_k) \exp(z_i) - \exp^2(z_i)}{\sum_{k=1}^m \exp^2(z_k)} \\ &= \frac{\exp(z_i)}{\sum_{k=1}^m \exp(z_k)} \left( 1 - \frac{\exp(z_i)}{\sum_{k=1}^m \exp(z_k)} \right) \\ &= y_i(1 - y_i). \end{aligned} \tag{10}$$

Furthermore, if  $i \neq j$ , then

$$\begin{aligned} \frac{\partial y_i}{\partial z_i} &= \frac{\partial \left( \frac{\exp(z_{ij})}{\sum_{a=1}^m \exp(z_{ia})} \right)}{\partial z_i} \\ &= -\exp(z_{ij}) \left( \frac{1}{\sum_{a=1}^m \exp(z_{ia})} \right) \exp(z_i) \\ &= -y_i y_j. \end{aligned} \tag{11}$$

Applying Eqs. (7) - (11), we have

$$\begin{aligned} \nabla f^t(w_i) &= \left( -\sum_{j=1}^k \tilde{y}_j \frac{1}{y_j} \right) \frac{\partial y_j}{\partial z_i} \\ &= \sum_{j \neq i} \frac{\tilde{y}_j}{y_j} y_i y_j - \frac{\tilde{y}_i}{y_i} y_i (1 - y_i) \\ &= \sum_{j \neq i} \tilde{y}_j y_i + \tilde{y}_i y_i - \tilde{y}_i \\ &= y_i \sum_j \tilde{y}_j - \tilde{y}_i. \end{aligned} \tag{12}$$

Furthermore, according to Eq. (2) and Eq. (12), we have

$$\nabla f^t(w_i) = y_i - \tilde{y}_i. \tag{13}$$

#### IV. GRADIENT DERIVATION AND ALGORITHM DESIGN

In order to improve training effect of optimization algorithms in classification problems, we utilize maximum probability of predictive value to reduce the error of cross entropy function at each iteration, and the novel loss function is called MPCE. In this section, we give the mathematical expression and algorithm design of MPCE.

Let  $y_{\max}$  denote the maximum in  $\{y_1, y_2, \dots, y_m\}$ , where  $m$  is the number of class and the  $u$ -th class is the true class. Moreover, the  $u$ -th coordinate of  $\tilde{y}$  is 1. Let  $\mathbf{y}' := (y_{\max} - y_u)\tilde{\mathbf{y}}$  with  $\tilde{\mathbf{y}}$  is the vector of real classes. Due to the values of untrue classes are both zero, we have

$$\sum_{i=1}^m (y_{\max} - y_i)\tilde{y}_i = y_{\max} - y_u. \tag{14}$$

Therefore, we have a maximized cross entropy loss function,

$$\begin{aligned} f^t(W) &= -\sum_{i=1}^m y'_i \log(y_i) \\ &= -\sum_{i=1}^m (y_{\max} - y_u)\tilde{y}_i \log(y_i), \end{aligned} \tag{15}$$

where  $y'_i$  denotes the  $i$ -th coordinate value of vector  $\mathbf{y}'$ .

In order to analyze performance of MPCE, we give the gradient derivation process of Eq. (15). According to Eqs. (4) and (15), we have

$$\begin{aligned} \frac{\partial f^t(w_i)}{\partial y_j} &= \frac{\partial \left( -\sum_j (y_{\max} - y_j)\tilde{y}_i \log y_j \right)}{\partial y_j} \\ &= -\sum_j (y_{\max} - y_j) \frac{\tilde{y}_i}{y_j}. \end{aligned} \tag{16}$$

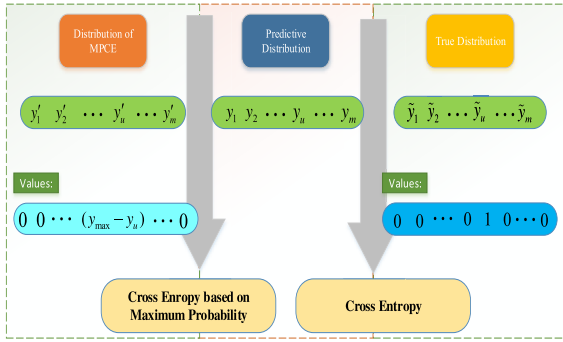


FIGURE 2. Differences between MPCE distribution and CE distribution.

From Eqs. (7), (11), (12), (14) and (16), we have

$$\begin{aligned}
 \nabla f^t(w_i) &= \left( -\sum_j (y_{\max} - y_j) \frac{\tilde{y}_j}{y_j} \right) \frac{\partial y_j}{\partial z_i} \\
 &= \sum_{j \neq i} \frac{(y_{\max} - y_j)}{y_j} y_i y_j \tilde{y}_i - \frac{(y_{\max} - y_i)}{y_i} y_i \tilde{y}_i (1 - y_i) \\
 &= \sum_{j \neq i} (y_{\max} - y_j) y_i \tilde{y}_i + (y_{\max} - y_i) y_i \tilde{y}_i - (y_{\max} - y_i) \tilde{y}_i \\
 &= y_i \sum_j (y_{\max} - y_j) \tilde{y}_i - (y_{\max} - y_i) \tilde{y}_i \\
 &= y_i (y_{\max} - y_u) - (y_{\max} - y_i) \tilde{y}_i. \tag{17}
 \end{aligned}$$

Due to  $y_i, y_{\max}, y_u \in [0, 1]$ , and  $\tilde{y}_i \in \{0, 1\}$ , we have

$$\begin{aligned}
 \sum_{i=1}^m \|\nabla f^t(w_i)\| &= \sum_{i=1}^m \|y_i (y_{\max} - y_u) - (y_{\max} - y_i) \tilde{y}_i\| \\
 &\leq \sum_{i=1}^m \|(y_{\max} - y_u) - (y_{\max} - y_i)\| \\
 &= \sum_{i=1}^m \|y_i - y_u\|. \tag{18}
 \end{aligned}$$

According to the aforementioned,  $y_u$  is the predictive probability of true class, therefore,  $0 \leq y_u < 1$ . Moreover, the real classes distribution  $\tilde{\mathbf{y}} = (0, \dots, 0, 1, 0, \dots, 0)$ . Therefore, we obtain

$$\sum_{i=1}^m \|\nabla f^t(w_i)\| \leq \sum_{i=1}^m \|y_i - \tilde{y}_i\|. \tag{19}$$

Therefore, the back propagation error of MPCE is less than that of CE for each iteration. In other words, CE may have a faster convergence rate at the start of iterations than MPCE. However, due to the bigger back propagation error, CE is more difficult to get the optimization point. On the contrary, MPCE backs a more accurate loss at each iteration when approaching the optimal point. Moreover, the difference of structures between CE and MPCE is shown as FIGURE 2. CE sets the probability of true class to 1, and others to 0. However, the probability 1 may too large for true class because of it contains some inaccuracy and redundant information. For this reason, the setting of CE makes the gradient direction too big at each iteration. To reduce the redundant information,

MPCE sets the probability of true class to  $y_{\max} - y_u$ . Since  $y_{\max} - y_u < 1$ , MPCE eliminates some redundant information of CE, and obtains a more accurate entropy for weight update.

**Algorithm 1** The Gradient Update Algorithm Based on MPCE

```

Input:  $\mathbf{x}$  (input samples)
 $\eta$  (learning rate)
 $u$  (the coordinate of true class in  $\{1, 2, \dots, m\}$ )  $m$  (the number of multi-classes)

Output:  $W$ 
1: for  $t = 1, 2, \dots, T$  do
2:    $t = t + 1$ 
3:   for  $i = 1, 2, \dots, m$  do
4:      $\mathbf{z} = \mathbf{w}_i^{t-1} \mathbf{x}$ 
5:      $\mathbf{y} = \text{softmax}(\mathbf{z})$ 
6:      $y_{\max} = \max\{y_1, y_2, \dots, y_m\}$ 
7:     obtaining  $y_u$  based on  $\tilde{\mathbf{y}}$  for  $\tilde{y}_u = 1$ 
8:      $\nabla f(\mathbf{w}_i^{t-1}) = (y_{\max} - y_u) \mathbf{y} - (y_{\max} - y_i) \tilde{\mathbf{y}}$ 
9:      $\mathbf{w}_i^t = \mathbf{w}_i^{t-1} - \eta \nabla f(\mathbf{w}_i^{t-1})$ 
10:    updating  $W$  by  $\mathbf{w}_i^t$ 
11:   end for
12: end for
13: Return  $W$ 
    
```

The gradient update algorithm based on MPCE is shown in Algorithm 1. The algorithm shows the update rule of the  $i$ -th cell weight. The input value is the training samples  $\mathbf{x}$ , and  $\mathbf{z}$  is the output of neural networks, learning rate  $\eta$  and true class  $\tilde{y}_u$ . Learning rate  $\eta$  is a constant that set before performing experiments. For the  $t$  iteration,  $\mathbf{z}$  first be transformed in probability by softmax function. Then, MPCE searches the maximum value  $y_{\max}$  in  $\{y_1, y_2, \dots, y_m\}$ . According to  $y_{\max}$  and  $y_u$ , MPCE computes gradient  $\nabla f(\mathbf{w}_i)$ . Finally, MPCE updates weight  $\mathbf{w}_i$  along the gradient direction.

**V. SIMULATED EXPERIMENTS**

In order to evaluate the performance of the proposed algorithm, we conduct three groups of simulated experiments for multi-class classification in neural networks. We focus on some experiment indicators: the cross entropy loss respectively on the same number of epoches and the same running time, the training and the test accuracy on the same number of samples, and the relationship between the changing of class number and the performance of loss functions.

**A. FULLY CONNECTED NEURAL NETWORK**

We firstly use fully connected neural network to finish multi-class classification problem on six public datasets.

1) EXPERIMENTAL SETUP

*a: DATASETS*

We use six public datasets<sup>1</sup>, news20, aloi, mnist, connect-4, sensorless and sector, in our simulated experiments.

<sup>1</sup><https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

The news20 is a famous dataset for text classification, which contains twenty topics of news. The aloi is produced by University of Amsterdam which collects color images of one-thousand small objects. The mnist is a widely used handwritten digital dataset, which from Arabic numeral 0 to 9. The connect-4 contains position information of a game. Moreover, the sensorless collects lots of features that are extracted from electric current drive signals. Besides, the sector is used for text classification. Furthermore, the six datasets are summarized in Table 1.

TABLE 1. Summary of the multi-class datasets.

Dataset	Features	Classes	Instances
news20	62,061	20	15,935
aloi	128	1,000	108,000
mnist	784	10	60,000
connect-4	126	3	67,557
sensorless	48	11	58,509
sector	55,197	105	64,120

b: COMPARED METHODS

In order to evaluate the performance of the proposed method, we compare it with three methods: Bayesian [36], SVM [37], CE, TCE [38] and ACE [40]. Next, we will introduce the following formulas of TCE and ACE.

TCE: The formula of TCE is as follows,

$$f_{\alpha}(y, \tilde{y}) = \frac{1}{1-\alpha} \sum_{i=1}^m y_i \left( (1 - \log \tilde{y}_i)^{1-\alpha} - \frac{1}{1-\alpha} \right),$$

where  $\alpha \in (0, 1)$  is the parameter.

ACE: First, ACE chooses a sample  $s$  from the full classes.

$$z'_i = \begin{cases} z_{s_i} - \log(mq_{s_i}) & \text{if } y_{s_i} = 0 \\ z_{s_i} & \text{else,} \end{cases}$$

where  $m$  denotes the number of classes,  $q$  is the sampled probability of each negative class, and  $z$  denotes the output of neural networks. ACE then obtains its new softmax function as follows,

$$y'_i = \frac{\exp(z'_i)}{\sum_{j=1}^{m+1} \exp z'_j}.$$

Finally, the loss function of ACE is given by,

$$\begin{aligned} f(y, y') &= - \sum_{i=1}^m y_i \log y' \\ &= \log \sum_{i=1}^m \exp z_i - \sum_{i=1}^m y_i z_i. \end{aligned}$$

c: PARAMETER SETTINGS

We set the learning rate  $\eta = \frac{1}{\sqrt{t}}$  for all methods on the six datasets. Moreover, the initialization of weight  $w_{ij}^0$  is zero for all methods. Furthermore, the parameters of neural networks on the six datasets are summarized in Table 2. The neural

TABLE 2. Summary of parameters in neural networks.

Dataset	Input Layer	Hidden Layer	Output Layer	Iterations	Testing Samples
news20	1 × 62,061	62,061 × 20	1 × 20	15,935	10,000
aloi	1 × 128	128 × 1000	1 × 1,000	108,000	10,000
mnist	1 × 784	784 × 10	1 × 10	60,000	10,000
connect-4	1 × 126	126 × 3	1 × 3	67,557	10,000
sensorless	1 × 48	48 × 11	1 × 11	58,509	10,000
sector	55,197	105	105	64,120	10,000

number of input layer is related to the dimensions of input data. The neural number of hidden layers is decided by the input layer and output layer. Moreover, the number of output layer is always same as the number of classes.

In addition, the hardware environment of our simulated experiments is shown in TABLE 3. The version of GPU is the GTX 1080 Ti which is produced by Nvidia. Moreover, CPU is the i7-6850k from Inter. Besides, the internal storage has 32GB. In addition, we use a solid state disk (SSD) with 250GB to speed up data reading, and a 2TB hard disk drive (HDD) to ensure sufficient data storage space.

TABLE 3. The Hardware Environment of the Experiments.

GPU	CPU	Internal Storage	SSD	HDD
GTX 1080 Ti	i7-6850k	32GB	250GB	2TB

2) EXPERIMENTAL RESULTS AND ANALYSIS

First of all, we give the experimental results of cross entropy loss on epoches. We run every dataset for 10 times (i.e. 10 epoches in experiments) and calculated the average for the 10 times. In our experiments, the ratio of each dataset is set to the same for all comparison methods. FIGURE 3 shows that the results of four methods on six public datasets. The experimental results show that the cross entropy loss of MPCE is the least of the four methods on six datasets. Specifically, the experiments on news20 is shown in FIGURE 3 (a). From first epoch to tenth epoch, the cross entropy loss of MPCE decreases to 0.1 from 1. However, the loss of CE, TCE and ACE all decrease to about 3 from 4. FIGURE 3 (b) shows the results on aloi. The losses of all the four method have obvious decrease. Meanwhile, MPCE reduces to the lowest point. In FIGURE 3 (c), despite the start loss of MPCE higher than that of ACE on mnist, however, the final loss of MPCE is the least. FIGURE 3 (d) shows that the start losses of TCE and ACE less than MPCE and CE on connect-4, nevertheless, MPCE decreases the fastest and the greatest. The downward trend of cross entropy loss on sensorless is shown in FIGURE 3 (e). In this case, the start loss of MPCE is the highest, but the final loss of MPCE is the least.

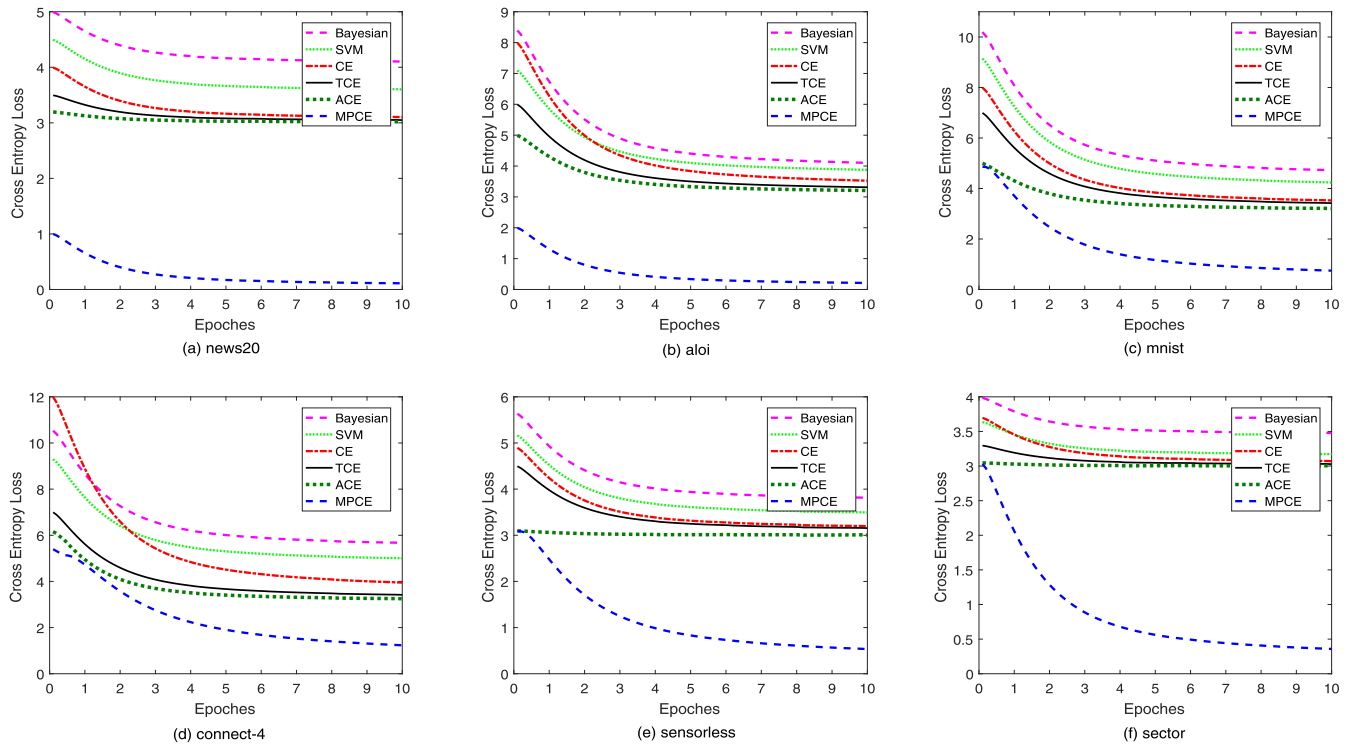


FIGURE 3. Convergence of Bayesian, SVM, CE, TCE, ACE and MPCE, i.e. the decrease trend of cross entropy loss from one epoch to ten epoch.

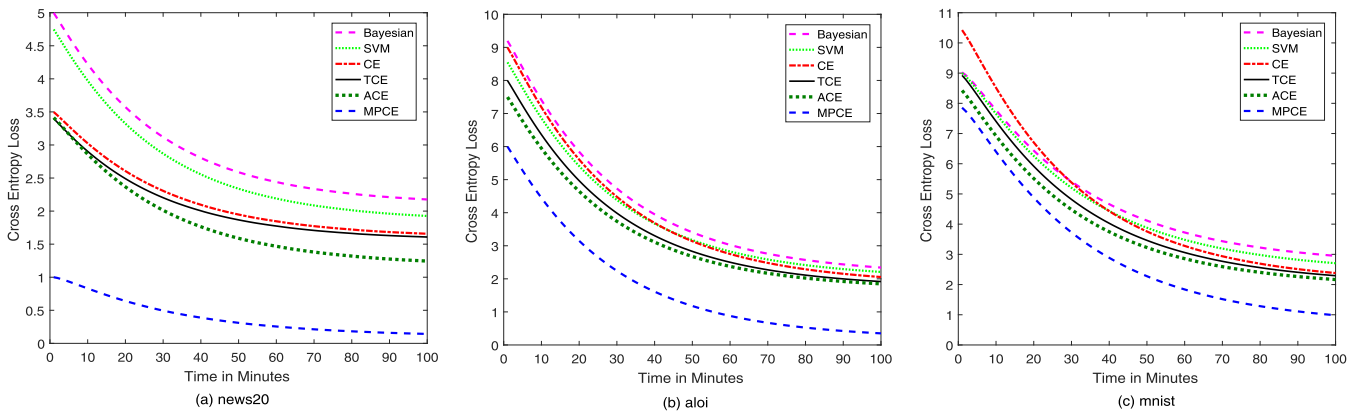


FIGURE 4. The relationship of time and loss of Bayesian, SVM, CE, TCE, ACE and MPCE on news20, aloi and mnist datasets.

In FIGURE 3 (f), CE, TCE and ACE have poor performance on sector, they reduce few losses during ten epoches. On the contrary, MPCE still performs well.

Moreover, we respectively run the compared algorithms on news20, aloi and mnist datasets to find the relationship of time and cross entropy loss. And the results are shown in FIGURE 4 that demonstrate the cross entropy loss of MPCE reaches the lower value than other algorithms at the same fixed time on the three datasets.

The results in FIGURE 3 and FIGURE 4 demonstrate the correctness of our aforementioned theory analysis. MPCE eliminates some redundant information by using  $y_{max} - y_u$  for true class, which brings a faster convergence rate. We utilize

a smaller error to adjust the weights at each iteration, which guarantees the optimization algorithm converge precisely to the optimal point. In effect, because of the oversize error adjustment, the standard cross entropy loss function makes the optimization algorithm oscillate around the optimal point, which leads to a long time to fail to converge.

Secondly, we observe the training accuracy of four methods on same number of samples. For the fairness of the experiment, we respectively take 20,000 samples from training samples on these datasets for training accuracy calculating. In FIGURES 5 (a) (c) and (d) (i.e. news20, mnist and connect-4), the accuracy of MPCE is the highest from 1,000 samples to 20,000 samples. FIGURE 5 (b) shows that

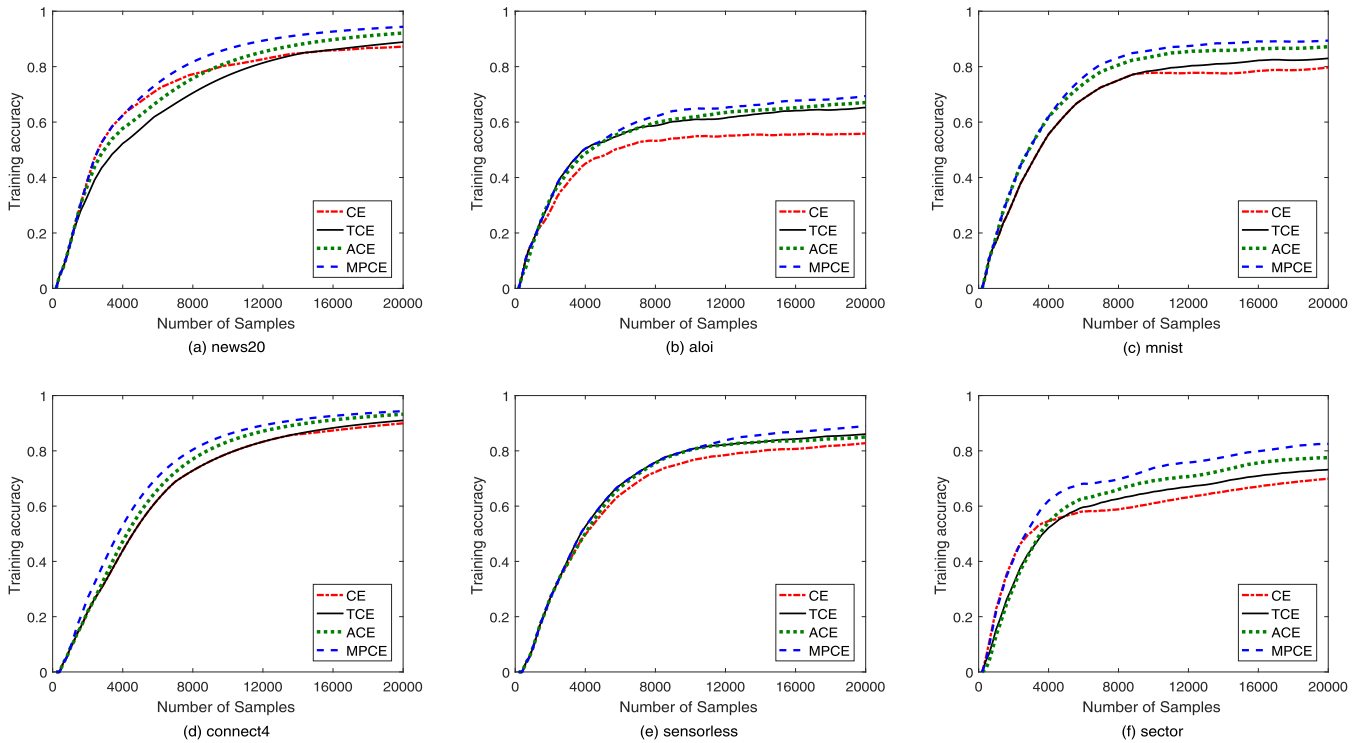


FIGURE 5. The training accuracy of CE, TCE, ACE and MPCE. Each method takes 20, 000 samples from training datasets to calculate the accuracy.

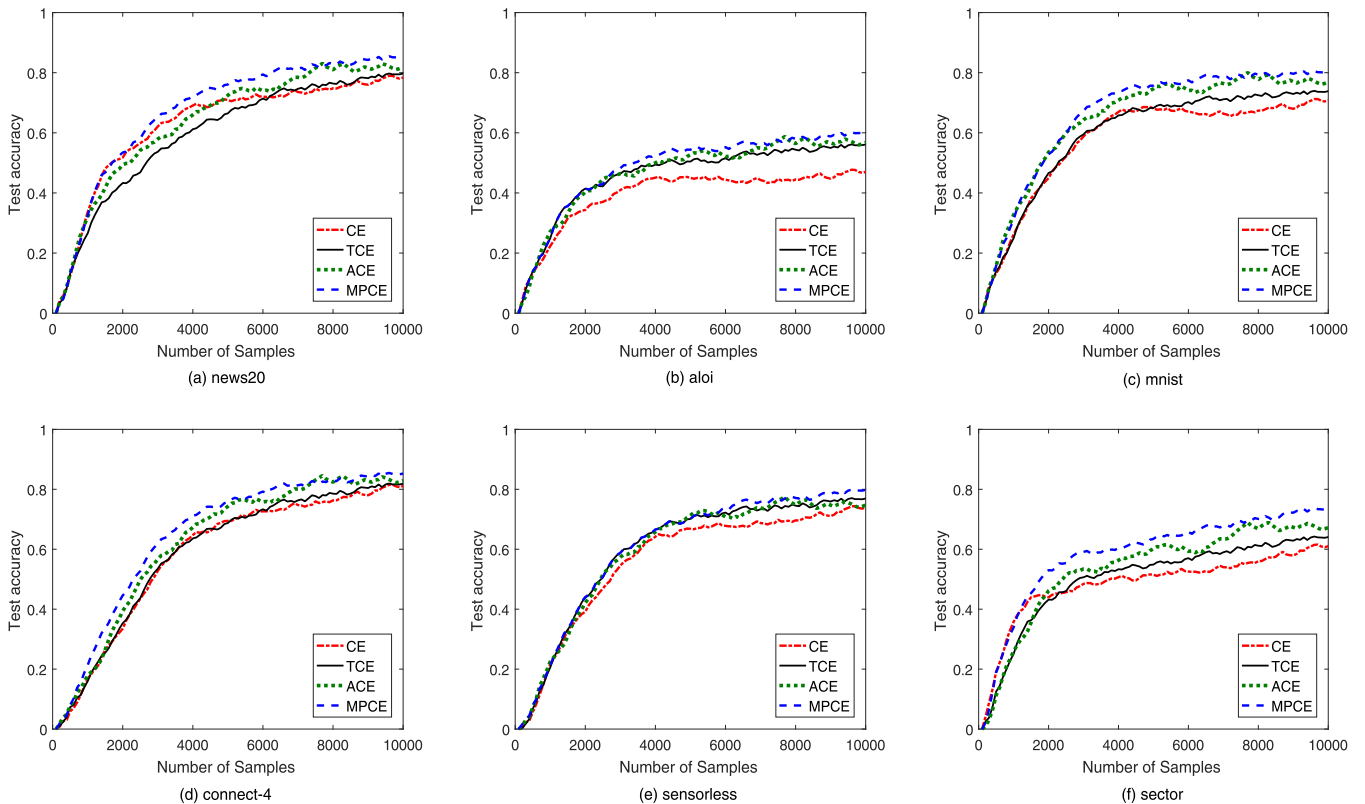


FIGURE 6. The test accuracy of CE, TCE, ACE and MPCE. Each method takes 10, 000 samples from test datasets to calculate the accuracy.

the accuracy of TCE is similar as that of MPCE when samples less than 4, 000 on aloi. However, MPCE keeps the highest accuracy when samples bigger than 4, 000. When samples

less than 10, 000, TCE, ACE and MPCE have almost the same accuracy on sensorless (i.e. FIGURE 5 (e)), however, MPCE reach the highest accuracy after 10, 000 samples.



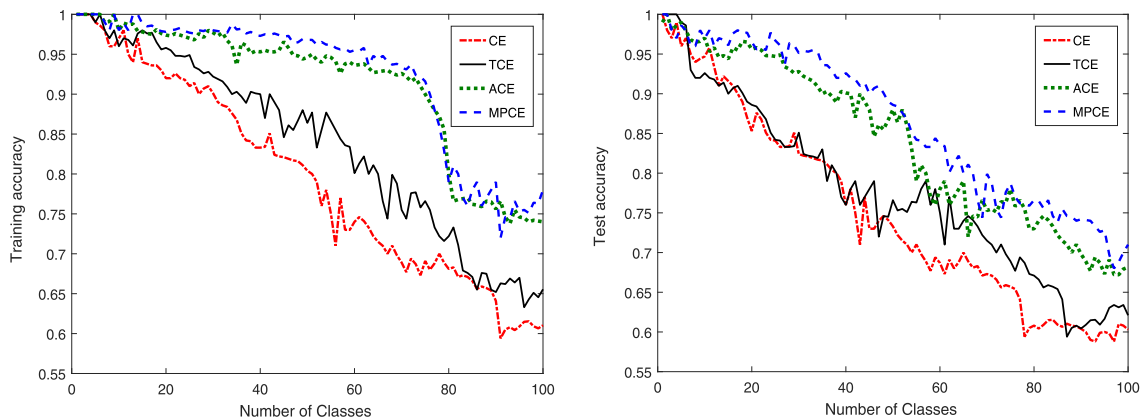


FIGURE 7. The relationship between classes and accuracy with compared methods on sector dataset.

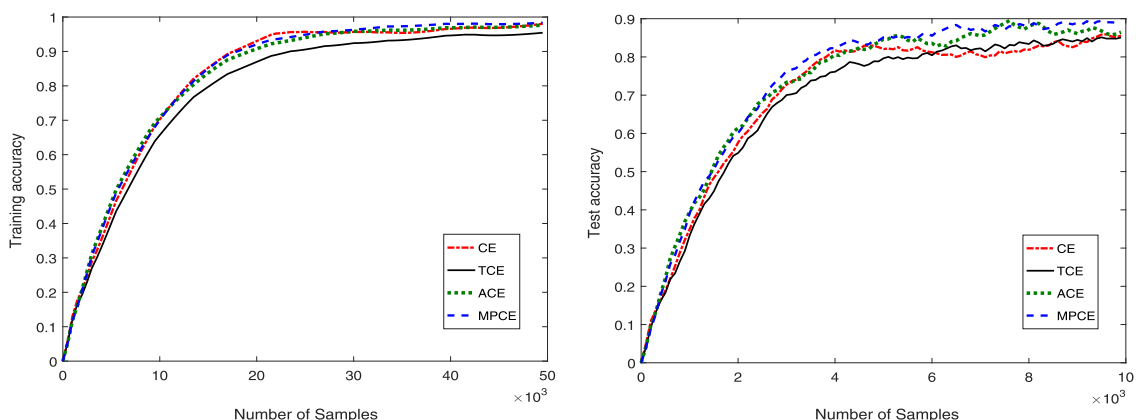


FIGURE 8. The training and test accuracy of compared methods in CNN on mnist.

In FIGURE 5 (f), MPCE keeps the highest accuracy after 3,000 samples on sector.

Theoretically, the algorithms based on CE and MPCE could both have a perfect performance if a massive of training samples exist. However, in fact, it is very difficult for us to obtain such a large amount of samples, furthermore, these samples should be labeled by people. Therefore, our propose MPCE to make the optimization algorithm reach a higher accuracy under the limited samples, which demonstrated in FIGURE 5.

Then, we demonstrate the performance of MPCE through a set of experiments on test accuracy. We respectively take 10,000 samples from test sets of six datasets. MPCE has the highest test accuracy on all the six datasets when samples is 10,000. As shown in FIGURE 6, the test accuracy of MPCE on news20, mnist, sensorless and connect-4 (i.e. FIGURE 6 (a), (c), (e) and (d)) higher than 80%. However, MPCE has a test accuracy of less than 80% on aloi and sector (i.e. FIGURE 6 (b) and (f)). The reason of this case is that the class number of news20, mnist, sensorless and connect-4 is far smaller than that of aloi and sector. The class number of aloi is 10,000, it thus has the lowest test accuracy. However, the influence of class number is limited in a span. When the difference of class number of two datasets is very small, the test accuracy will not be affected very

much. Therefore, although the class number of sensorless slightly smaller than that of news20, the test accuracy of sensorless worse than that of news20.

The most direct indicator to evaluate a loss function is the predictive accuracy. When choosing 10,000 samples for testing, MPCE retains a slightly better testing accuracy than others compared methods on six public datasets. Finally, we focus on the different influences of CE, TCE, ACE and MPCE on the samples. We choose the sensor dataset to run the group of experiments. The number of classes is set to a fixed value at each running time. The class number increases from 1 to 100 as the experiments go on. The results of the experiments is shown in FIGURE 7. The left of FIGURE 7 shows the relationship between class number and training accuracy, and MPCE performs better than others. The right of FIGURE 7 presents the connection of class number and test accuracy. From this figure, we can see that the performance of MPCE is better than other methods.

**B. CONVOLUTIONAL NEURAL NETWORK**

Next, we consider the case that MPCE is used in the Convolutional Neural Network (CNN). CNN is a successful method for image classification, therefore, we will run the methods on the image dataset mnist.

## 1) EXPERIMENTAL SETUP

Mnist dataset is composed by images of size  $28 \times 28$  and has 10 classes. We set training samples to 50,000 (i.e. 5,000 samples per class), and set test samples to 10,000 (i.e. 1,000 samples per class). Moreover, we use the modern version of LeNet-5 [42], [43] as the model for CNN, and use SGD as the optimization algorithm in the model. The architecture of LeNet-5 for CNN is shown in following Table 4.

**TABLE 4. Architecture of LeNet-5 for CNN.**

Layer	Width	Height	Depth	Kernel size	Stride
Input	28	28	1	-	-
Convolution	24	24	20	5	1
Max Pooling	12	12	20	2	2
Convolution	8	8	50	5	1
Max Pooling	4	4	20	2	2
Fully Connected	1	1	500	-	-
ReLU	1	1	500	-	-
Fully Connected	1	1	10	-	-
Softmax	1	1	10	-	-

## 2) EXPERIMENTAL RESULTS

The results of CNN on the mnist dataset is shown in FIGURE 8. In the left of FIGURE 8, we use 50,000 samples for the training process and respectively use four compared loss function in CNN. In this part of experiments, the training accuracy of MPCE is the highest, which is about 98%. Then, we use 10,000 samples to test the accuracy of the CNN model. From the right of FIGURE 8, we can see that MPCE also obtains the highest test accuracy in this part of experiments. Therefore, in CNN model on mnist dataset, the accuracy of the proposed method is higher than other three methods.

## VI. CONCLUSION

In this paper, our work shows the importance of loss function for multi-class classifier learning in neural networks. We have proposed a new cross entropy loss function, named MPCE, that utilizes the maximum probability of predictive value to reduce the cross entropy loss of each iteration. We give the mathematical expression and gradient derivation process of MPCE. Moreover, the gradient derivation process shows that MPCE has less loss than MSE and CE on theory. In order to demonstrate the performance of MPCE on experiments, we aim at three aspects on six public classification datasets. The first set of experiments show that MPCE has a faster convergence rate and a smaller cross entropy loss than other compared methods. Moreover, the second set of experiments show that MPCE has the highest training accuracy when sample number is 20,000, and the training accuracy of MPCE respectively is 95.12%, 71.14%, 90.61%, 95.33%, 90.43% and 83.23%. Furthermore, the last set of experiments show that MPCE has the highest test accuracy when test sample number is 10,000, and the test accuracy respectively

is 85.25%, 60.27%, 80.27%, 85.36%, 79.83% and 73.42%. We also use CNN to evaluate the performance of MPCE on mnist dataset. MPCE has the highest training and test accuracy on image classification in our experiments.

## REFERENCES

- [1] M. Elkano, M. Galar, J. A. Sanz, A. Fernandez, E. Barrenechea, F. Herrera, and H. Bustince, "Enhancing multiclass classification in FARC-HD fuzzy classifier: On the synergy between  $n$ -dimensional overlap functions and decomposition strategies," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1562–1580, 2015.
- [2] M. P. Arthur and K. Kannan, "Cross-layer based multiclass intrusion detection system for secure multicast communication of MANET in military networks," *Wireless Netw.*, vol. 22, no. 3, pp. 1035–1059, 2016.
- [3] B. Ravikumar, D. Thukaram, and H. P. Khincha, "Comparison of multiclass SVM classification methods to use in a supportive system for distance relay coordination," *IEEE Trans. Power Del.*, vol. 25, no. 3, pp. 1296–1305, Jul. 2010.
- [4] A. Sáez, J. Sánchez-Monedero, P. A. Gutierrez, and C. Hervás-Martínez, "Machine learning methods for binary and multiclass classification of melanoma thickness from dermoscopic images," *IEEE Trans. Med. Imag.*, vol. 35, no. 4, pp. 1036–1045, Apr. 2016.
- [5] J. Xu, Y. Tang, B. Zou, Z. Xu, L. Li, Y. Lu, and B. Zhang, "The generalization ability of SVM classification based on Markov sampling," *IEEE Trans. Cybern.*, vol. 45, no. 6, pp. 1169–1179, Jun. 2015.
- [6] A. Jamehbozorg and S. M. Shahrtash, "A decision-tree-based method for fault classification in single-circuit transmission lines," *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2190–2196, Oct. 2010.
- [7] H. Fehri, A. Gooya, Y. Lu, E. Meijering, S. A. Johnston, and A. F. Frangi, "Bayesian polytrees with learned deep features for multi-class cell segmentation," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3246–3260, Jul. 2019.
- [8] J. Xu, J. Han, F. Nie, and X. Li, "Re-weighted discriminatively embedded  $K$ -means for multi-view clustering," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 3016–3027, Jun. 2017.
- [9] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
- [10] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [11] K. Xiong, B. Wang, and K. J. R. Liu, "Rate-energy region of SWIPT for MIMO broadcasting under nonlinear energy harvesting model," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5147–5161, Aug. 2017.
- [12] P. Tang, X. Wang, B. Feng, and W. Liu, "Learning multi-instance deep discriminative patterns for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3385–3396, Jul. 2017.
- [13] Y. Chi, E. J. Griffith, J. Y. Goulermas, and J. F. Ralph, "Binary data embedding framework for multiclass classification," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 4, pp. 453–464, Aug. 2015.
- [14] M. N. I. Qureshi, B. Min, H. Park, D. Cho, W. Choi, and B. Lee, "Multi-class classification of word imagination speech with hybrid connectivity features," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 10, pp. 2168–2177, Oct. 2018.
- [15] V. Kůrková and M. Sanguineti, "Classification by sparse neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2746–2754, Sep. 2019.
- [16] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017.
- [17] M. Huang, Q. Qian, and X. Zhu, "Encoding syntactic knowledge in neural networks for sentiment classification," *ACM Trans. Inf. Syst.*, vol. 35, no. 3, pp. 1–27, 2017.
- [18] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *Proc. ICLR*, Vancouver, BC, Canada, May 2018, pp. 1–23.
- [19] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.
- [20] A. Rakhlin, A. Davydow, and S. Nikolenko, "Land cover classification from satellite imagery with U-net and Lovasz-softmax loss," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 1–5.

- [21] M. R. Bonyadi and D. C. Reutens, "Optimal-margin evolutionary classifier," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 885–898, Oct. 2019.
- [22] F. Zhang, Y. Shi, H. K. T. Ng, and R. Wang, "Information geometry of generalized Bayesian prediction using  $\alpha$ -divergences as loss functions," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1812–1824, Mar. 2018.
- [23] D. Liang, Z. Xu, and D. Liu, "A new aggregation method-based error analysis for decision-theoretic rough sets and its application in hesitant fuzzy information systems," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1685–1697, Dec. 2017.
- [24] Y. Chang and C. Jung, "Single image reflection removal using convolutional neural networks," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1954–1966, Apr. 2019.
- [25] Y. Du, Y. Fu, and L. Wang, "Representation learning of temporal dynamics for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3010–3022, Jul. 2016.
- [26] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2247–2256, Aug. 2015.
- [27] S. Chang and L. Carin, "A modified SPIHT algorithm for image coding with a joint MSE and classification distortion measure," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 713–725, Mar. 2006.
- [28] P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, and A. Perallos, "A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross entropy," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 557–569, Feb. 2016.
- [29] A. Sangari and W. Sethares, "Convergence analysis of two loss functions in soft-max regression," *IEEE Trans. Signal Process.*, vol. 64, no. 5, pp. 1280–1288, Mar. 2016.
- [30] A. S. Bosman, A. Engelbrecht, and M. Helbig, "Visualising basins of attraction for the cross-entropy and the squared error neural network loss functions," Jan. 2019, *arxiv:1901.02302*. [Online]. Available: <https://arxiv.org/abs/1901.02302>
- [31] B. Han and Y. Wu, "A novel active contour model based on modified symmetric cross entropy for remote sensing river image segmentation," *Pattern Recognit.*, vol. 67, pp. 396–409, Jul. 2017.
- [32] K. Hu, Z. Zhang, X. Niu, Y. Zhang, C. Cao, F. Xiao, and X. Gao, "Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function," *Neurocomputing*, vol. 309, pp. 179–191, Oct. 2018.
- [33] J. Suh, J. Gong, and S. Oh, "Fast sampling-based cost-aware path planning with nonmyopic extensions using cross entropy," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1313–1326, Aug. 2017.
- [34] P. Qian, Y. Jiang, Z. Deng, L. Hu, S. Sun, S. Wang, and R. F. Muzic, "Cluster prototypes and fuzzy memberships jointly leveraged cross-domain maximum entropy clustering," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 181–193, Jan. 2016.
- [35] A. Ghosh, H. Kumar, and P. S. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proc. AAAI*, San Francisco, CA, USA, Feb. 2017, pp. 1–7.
- [36] S. Z. Dadaneh, E. R. Dougherty, and X. Qian, "Optimal Bayesian classification with missing values," *IEEE Trans. Signal Process.*, vol. 66, no. 16, pp. 4182–4192, Aug. 2018.
- [37] C. Zhao, W. Liu, Y. Xu, and J. Wen, "A spectral-spatial SVM-based multi-layer learning algorithm for hyperspectral image classification," *Remote Sens. Lett.*, vol. 9, no. 3, pp. 218–227, 2018.
- [38] M. Martinez and R. Stiefelwagen, "Taming the cross entropy loss," Oct. 2018, *arXiv:1810.05075*. [Online]. Available: <https://arxiv.org/abs/1810.05075>
- [39] Y. Bengio and J.-S. Senecal, "Adaptive importance sampling to accelerate training of a neural probabilistic language model," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 713–722, Apr. 2008.
- [40] G. Blanc and S. Rendle, "Adaptive sampled softmax with kernel based sampling," in *Proc. ICML*, Stockholm, Sweden, Jul. 2018, pp. 1–13.
- [41] A. G. Joseph and S. Bhatnagar, "An online prediction algorithm for reinforcement learning with linear function approximation using cross entropy method," *Mach. Learn.*, vol. 107, no. 8, pp. 1385–1429, Jun. 2018.
- [42] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*, Cavtat-Dubrovnik, Croatia, 2003, pp. 107–119.
- [43] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, Oct. 2017.



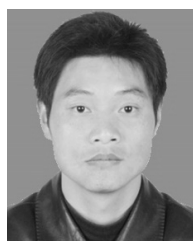
**YANGFAN ZHOU** is currently pursuing the master's degree in computer science and technology with the Henan University of Science and Technology. His current research interests include theoretical and algorithmic issues related to large-scale optimization, stochastic optimization, distributed multiagent optimization, and their applications to deep learning, meta-reinforcement learning, and networking.



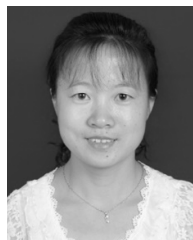
**XIN WANG** received the Ph.D. degree in management science and engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2017. She is currently an Assistant Professor with Shanghai International Studies University. Her research interests include neuroscience, artificial intelligence, information systems, and networks.



**MINGCHUAN ZHANG** was born in Henan, China, in May 1977. He received the Doctor of Engineering degree in communication and information system from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014. He was an Associate Professor with the Henan University of Science and Technology. He is also the CIO of Henan Qunzhi Information Technology Company Ltd., and Guangzhou Xiangxue Pharmaceutical Company Ltd. His research interests include bio-inspired networks, the future Internet, and optimization.



**JUNLONG ZHU** received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2018. In 2018, he joined the Henan University of Science and Technology, Luoyang, China, where he is currently a Lecturer with the Information Engineering College. He is also the CTO of Henan Qunzhi Information Technology Company Ltd., and an Engineer of Guangzhou Xiangxue Pharmaceutical Company Ltd. His research interests include large-scale optimization, distributed multiagent optimization, stochastic optimization, and their applications to machine learning, signal processing, communications, and networking.



**RUIJUAN ZHENG** was born in Henan, China, in March 1980. She received the Doctor of Engineering degree in computer application from Harbin Engineering University, Harbin, China, in 2008. Since March 2017, she has been a Professor with the Henan University of Science and Technology. She is also the vice CTO of Henan Qunzhi Information Technology Company Ltd. Her research interests include computer security and network optimization.



**QINGTAO WU** was born in Jiangsu, China, in March 1975. He received the Doctor of Engineering degree in computer application from the East China University of Science and Technology, Shanghai, China, from March 2003 to March 2006. He is currently a Professor with the Henan University of Science and Technology. His research interests include computer security, the future Internet security, machine learning, and cloud computing.

• • •