Soft Computing (2020) 24:3727–3735
https://doi.org/10.1007/s00500-019-04141-w

METHODOLOGIES AND APPLICATION

An ensemble learning framework for convolutional neural network based on multiple classifiers

Yanyan Guo[1,2] · Xin Wang[1,2] · Pengcheng Xiao[3] · Xinzheng Xu[1,4]

Published online: 20 June 2019

Abstract

Traditional machine learning methods have certain limitations in constructing high-precision estimation models and improving generalization ability, but ensemble learning that combines multiple different single models into one model is significantly better than that obtained by a single machine learning model. When the types of data sets are diversified and the scale is increasing, the ensemble learning algorithm has the problem of incomplete representation of features. At this time, convolutional neural network (CNN) with excellent feature learning ability makes up for the shortcomings of ensemble learning. In this paper, an ensemble learning framework for convolutional neural network based on multiple classifiers is proposed. First, this method mainly classifies UCI data sets using the ensemble learning algorithms based on multiple classifiers. Then, feature extraction is performed on the image data set MNIST using a convolutional neural network, and the extracted features are applied as input to be classified using an ensemble learning framework. The experimental results show that the accuracy of ensemble learning is higher than the accuracy of a single classifier and the accuracy of CNN + ensemble learning framework is higher than the accuracy of ensemble learning framework.

Keywords Ensemble learning · Convolutional neural network · Bagging · Boosting · Random forest

1 Introduction

In recent decades, ensemble learning has been attracting attention in the field of machine learning because it can effectively solve practical application problems. Dasarathy and Sheela (1979) first proposed the idea of ensemble learning. Ensemble learning is a machine learning method that uses a series of base learners to learn and uses a certain rule to integrate individual learning results to achieve better outcomes than a single learner. Since most ensemble learning algorithms have no restrictions on the type of base learner and it has good applicability to many mature machine learning frameworks, ensemble learning is widely used in various fields. With the development of the times, more ensemble learning algorithms have been proposed, and major breakthroughs have been made in many fields. However, the existing ensemble learning algorithms also have some limitations. For example, the base learner needs to have high sensitivity and efficient learning ability; otherwise, it is easy to produce overfitting, increasing time cost and computational overhead.

With the increase in data scale, the improvement in computing power and the great innovation of algorithms, deep learning has begun to rise. Geoffrey Hinton, a professor at the University of Toronto in Canada and the master of machine learning, and his student Ruslan Salakhutdinov (2006) published an article in science that opened the wave of deep learning in academia and industry. Deep learning, especially convolutional neural networks, not only has excellent feature learning capabilities, but also overcomes training difficulties through layer-by-

Communicated by V. Loia.

✉ Xinzheng Xu
xuxinzh@163.com

[1] College of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

[2] Engineering Research Center of Mining Digital Ministry of Education, Xuzhou 221116, China

[3] Department of Mathematics, University of Evansville, Evansville, IN 47722, USA

[4] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou 363000, China

layer initialization. Therefore, convolutional neural network has been the most in-depth research, and it is widely used in various fields such as computer vision (Krizhevsky et al. 2012) and speech recognition (Hinton et al. 2012), and has achieved best performance in these fields.

In order to improve the classification efficiency, this paper proposes an ensemble learning framework for convolutional neural network based on multiple classifiers for improving the accuracy of classification. This paper mainly consists of two aspects of work: (1) Construct an ensemble learning framework based on different classifiers and compare it with the accuracy of a single classifier; and (2) extract features using CNN for MNIST data set, and then, classify extracted features using an ensemble learning framework.

This article is divided into five parts: Sect. 1 mainly introduces the relevant background, purpose, methods and the organizational structure; Sect. 2 mainly introduces the related work in recent years; Sect. 3 is the introduction of basic theory of convolutional neural network and ensemble learning; Sect. 4 is mainly to introduce the overall framework which describes the whole process; Sect. 5 is the experimental part, which introduces the data sets required, the experimental details, experimental results and analysis. Section 6 is a summary of this paper.

2 Related work

Ensemble learning is a new machine learning paradigm that uses multiple learners to solve the same problem. Because ensemble learning can significantly improve the generalization ability of the learning system, it is widely used in various fields, such as robot assistance (Adama et al. 2018), Web security detection (Zhou and Wang 2019) and online streaming data anomaly detection (Ding et al. 2017). With the deepening of research, many algorithms for optimizing ensemble learning have been proposed: Wang et al.(2016) proposed a multi-core ensemble learning algorithm that can make full use of historical data information to improve the ability of software to predict defects; Zhang et al.(2017) proposed a local hierarchical ensemble framework for hierarchical multi-label classification to solve the problem of ignoring the structural information between different classes in classification algorithm; a hybrid incremental ensemble learning (HIEL) approach proposed by Zhiwen et al. (2019) can simultaneously consider the feature space and the sample space to process the noise data set. Although ensemble learning shows better performance than a single classifier, as data present type diversity and rapid growth, ensemble learning shows some problems such as high computational complexity and low efficiency.

Deep convolutional neural networks (DCNNs) have become one of the most studied models in deep neural networks along with the continuous development of deep learning algorithms. Convolutional neural networks have demonstrated powerful application capabilities in many fields: In computer vision, Mask R-CNN proposed by He et al. (2017) achieved object detection by adding a branch of the target mask in parallel with the existing branch; if the segmentation goal of Mask R-CNN was turned to one-hot, it can also be used for human pose estimation. In natural language processing, WaveNet (van den Oord et al. 2016) used a convolutional neural network to generate a model to output a conditional probability of speech and sample synthesized speech; Zhang et al. (2018) proposed a deep convolutional neural network (DCNN) model using linear support vector machine to achieve speech recognition. In addition, Ji et al. (2019) proposed a Siamese fully convolutional network that can be used to extract buildings from satellite remote sensing images.

As can be seen that the existing methods are a combination of convolutional neural network and a single classifier, algorithms that fuse ensemble learning are lacking. Therefore, this paper proposes an ensemble learning framework for convolutional neural networks to deal with multi-classification problem.

3 Basic theory

3.1 Convolutional neural network

Convolutional neural network (CNN) is a feedforward neural network that includes alternating convolutional layers and pooling layer. The structure of CNN is shown in Fig. 1.

The difference between a convolutional neural network and an ordinary neural network is that convolutional neural network contains a feature extractor composed of a convolutional layer and a subsampling layer. In the convolutional layer, one neuron is only connected to a portion of the adjacent layer neurons. After the convolution operation

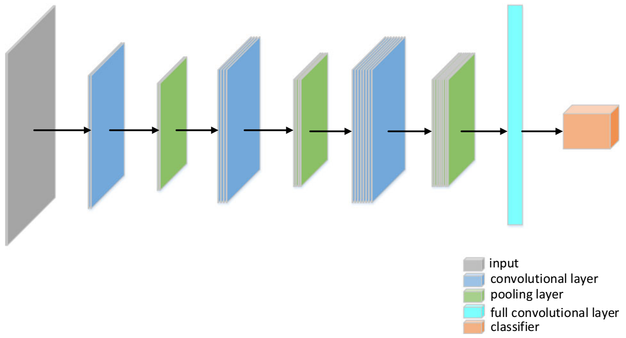


Fig. 1 Structure of CNN

is completed, several feature maps are generated and the neurons of the same feature map share the weight, which means the shared convolution kernel. The immediate benefit of shared weights is the reduction in connections between layers of the network while reducing the risk of overfitting. Subsampling is also called pooling. It usually has two forms: mean pooling and max pooling. Subsampling can be seen as a special convolution process. Convolution and subsampling greatly simplify the model complexity and reduce the parameters of the model.

CNN has great advantages in feature extraction: First, due to the nature of convolution and pooling, the extracted features are less likely to overfitting. Second, the features extracted by CNN are more scientific than the simple projection, direction and center of gravity. Third, the fit of the overall model can be controlled by the size of the different convolution, pooling and final output feature vectors. The dimension of the feature vector can be reduced when the model is overfitting, and the output dimension of the convolution layer can be increased when the model is underfitting.

3.2 Ensemble learning

Ensemble learning accomplishes learning tasks by building and combining multiple classifiers. Specifically, a plurality of *N* classifiers which have independent decision-making capabilities are combined according to a certain strategy to make a decision. Classifiers are also known as learners. The idea of ensemble learning is shown in Fig. 2.

At present, ensemble learning can be divided into two categories according to whether there are dependencies between individual learners: The first type is that there is no strong dependency between individual learners. A series of individual learners can be generated in parallel, and the representative algorithm is Bagging (Breiman 1996) series. The second category is that there is a strong dependency between individual learners. A series of individual learners basically need to be serially generated, and the representative algorithm is Boosting (Schapire 1989) series.

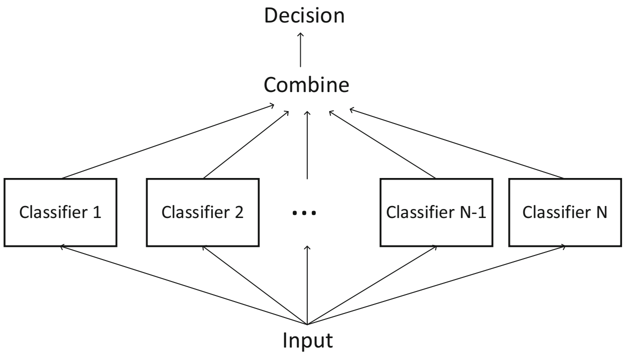


Fig. 2 General idea of ensemble learning

3.3 Bagging

Bagging (Breiman 1996) improves the accuracy of learning algorithms by combining randomly generated training sets, constructing a series of predictive functions and combining them into a predictive function by some strategy. The classification method required by Bagging is unstable. Unstable means that if the selected data set changes very little, it will make a huge change in the final classification result.

Random forest (Breiman 2001) is a specialized and advanced version of Bagging. The so-called specialized is because the base learners of random forests are decision trees. The so-called advanced is random forest which is based on the random sampling of Bagging, plus the random selection of features. The basic idea is not out of the Bagging category yet.

3.4 Boosting

Boosting (Schapire 1989) is an algorithm that combines base learning algorithms into strong learning algorithms. It improves the accuracy of any given learning algorithm by constructing a number of prediction functions and combining them according to matching strategies to generate prediction functions. But Boosting has a major drawback the model needs to know in advance the lower accuracy limit of the base classifier classification. In response to these problems and defects, Adaboost (Freund and Schapire 1995) was proposed.

XGBoost (Chen and Guestrin 2016) is the abbreviation of extreme gradient boosting. It is a machine learning function library focusing on gradient boosting algorithm. This library has gained wide attention due to its excellent learning effect and efficient training speed. XGBoost implements a generic tree boosting algorithm, and the tree model used is usually the CART. The idea of XGBoost is to constantly add trees and continuously perform feature segmentation to generate trees.

4 Framework description

4.1 Overview

This paper mainly contains the following two aspects: (1) ensemble learning framework: Multiple classifiers are used as base classifier of Bagging and Boosting for classification; and (2) CNN + ensemble learning framework: Firstly, the feature is extracted using convolutional neural network, and then, the ensemble learning algorithm in (1) is used for classification prediction.

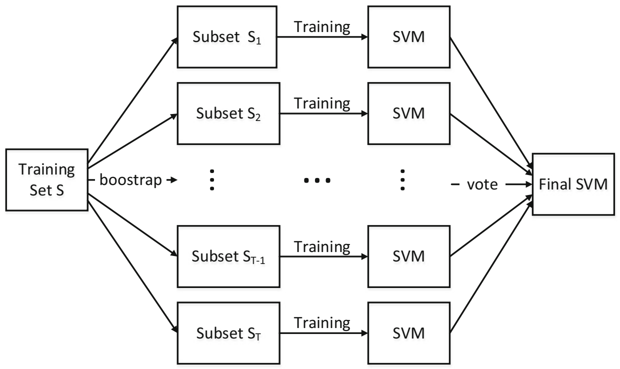


Fig. 3 Frame of the Bagging based on SVM

4.2 Ensemble learning framework

There are two types of ensemble learning algorithms framework: One is based on Bagging and another is based on Boosting.

4.3 Framework based on Bagging

The main steps of the Bagging-based framework are as follows:

(1) T training is performed on the training set $S = \{(x_1, y_1), (x_2, y_2),\ldots, (x_m, y_m)\}$: In each training, the bootstrap method is used to put back the random sampling of the training set S; that is, only a certain subset S_t of S is used as the current training set. After T training, T different classifiers C_t can be obtained;

2) Combine the base learning classifiers obtained from each training into final classifier: When classifying a test sample, the T classifiers are, respectively, called to obtain and count T classification results, and the class with the most occurrences is used as the last label.

The frame of the Bagging based on SVM (Cortes and Vapnik 1995) as the base learning classifier C is shown in Fig. 3.

Among them, the base classifier SVM can be replaced by decision tree (Breiman et al. 1984), MLP (Longstaff and Cross 1987), Naive Bayesian (Lewis 1998), KNN (Cover and Hart 1967) and Perceptron (Rosenblatt 1958). The algorithm of Bagging is shown in Algorithm 1.

Algorithm 1 Procedures of Bagging

```
Input:Training set S;
        Base learning classifier C;
        Number of learning rounds T.
Process:
    For t=1,...,T:
        $S_t$=boostrap sample from S;
        train classifier $C_t$ based on $S_t$;
    End For
Output:Majority vote of {$C_1$,$C_2$,...,$C_t$}
```

4.4 Framework based on Boosting

The Boosting framework is based on the Adaboost algorithm. The goal of the algorithm is to update the weight D. If a sample has been accurately classified, its weight is reduced when constructing the next training set; conversely, if a sample is not accurately classified, its weight is increased. At the same time, the right to speak of the base learning classifier is obtained. Then, the sample set after updating the weight is used to train the next classifier. The main steps of Boosting-based framework are as follows:

1. Initialize the weight distribution D_1 of the training data $S = \{(x_1, y_1), (x_2, y_2),\ldots, (x_m, y_m)\}$: If there are m samples, each sample point is given the same weight $1/m$ at the beginning

$$D_1 = \left(w_{1,1}, w_{1,2}, \ldots, w_{1,n}\right), \quad w_{1,i} = \frac{1}{m}, \quad i = 1, 2, \ldots, m \tag{1}$$

where $w_{1,i}$ represents the weight of the ith sample of the first iteration.

2. T-round training for the weighted training set S: In each iterative, the base learning classifier C_t is trained on the training set S given the weight D_t. According to the classification result, the error function value e_t and the current classifier's utterance right a_t are calculated:

$$e_t = \sum_{i=1}^{m} w_{t,i} I(C_t(x_i) \neq y_i) \tag{2}$$

$$a_t = \frac{1}{2} \log \frac{1 - e_t}{e_t} \tag{3}$$

where $C_t(x_i)$ represents the category predicted by using C_t for the ith sample point in the tth iteration; $I(\cdot)$ represents the error between the real category and the prediction category; the utterance right a_t indicates the importance of $C_t(x)$ in the final classifier.

The weight distribution D_{i+1} used for the next iteration is updated according to e_t and a_t:

$$D_{i+1} = \left(w_{t+1,1}, w_{t+1,2}, \ldots, w_{t+1,m}\right), \quad i = 1, 2, \ldots, m \tag{4}$$

$$w_{t+1,i} = \frac{w_{t,i}}{Z_t} \exp(-a_t y_i C_t(x_i)) \tag{5}$$

$$Z_t = \sum_{i=1}^{N} w_{t,i} \exp(-a_t y_i C_t(x_i)) \tag{6}$$

where $w_{t,\,i}$ is the weight of the ith sample at the tth iteration; y_i is the real class of the ith sample; Z_t is the normalization factor such that the sum of the weights corresponding to all samples is 1.

3. Combine the base learning classifiers obtained from each training into a strong final classifier. After the training, the base learning classifier with small classification error rate has a large utterance right, which plays a greater role in the final classification function, and base learning classifier with large classification error rate has a small utterance right, which plays a minor role in the final classification function. In other words, a base learning classifier with a low error rate accounts for a larger proportion in the final classifier, and vice versa. As shown in Eq. (7):

$$C(x) = sign(\sum_{t=1}^{T} a_t \, C_t(x)) \tag{7}$$

where sign(·) is a symbolic function and x is a sample to be prediction.

The frame of the Boosting based on SVM as the base learning classifier C is shown in Fig. 4.

Among them, the base classifier SVM can be replaced by decision tree, SVM, Naive Bayesian and Perceptron because the base classifier of the Boosting algorithm needs to support the sample weight. The algorithm of Bagging is shown in Algorithm 2.

Algorithm 2 Procedures of Boosting

```
Input: Training set S;
        Base learning classifier C;
        Number of learning rounds T.
Process:
        $D_1$=1/m;
        For t=1,...,T:
            Train classifier Ct from S using
            distribution;
            Calculate $e_t$ according to formula 2;
            Calculate $a_t$ according to formula 3;
            Update $D_{t+1}$ according to formula 4~6;
        end
Output: C(x) according to formula 7.
```

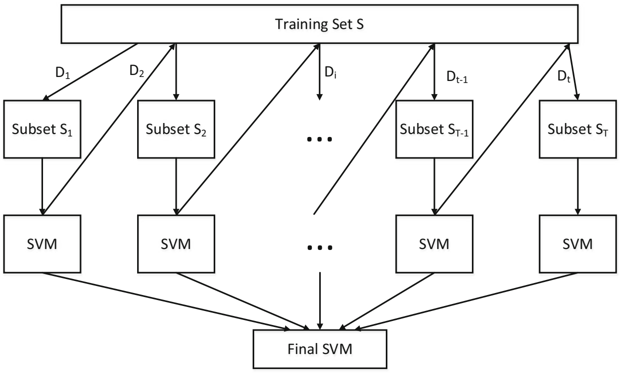


Fig. 4 Frame of the Boosting based on SVM

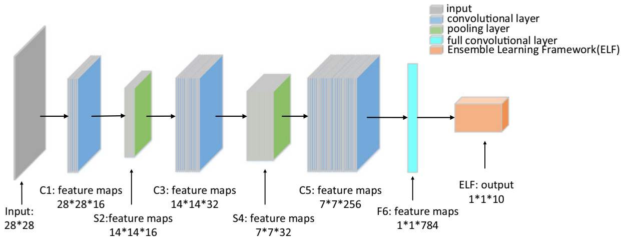


Fig. 5 Frame of the CNN + ensemble learning

4.5 CNN + ensemble learning framework

CNN has excellent advantages in feature extraction, so in this framework, CNN is first used to extract features from the image and then classified using the ensemble learning framework. The framework is shown in Fig. 5.

The above figure shows the overall framework of CNN + ensemble learning framework. After extracting features using CNN, an ensemble learning algorithm framework was added behind the F6 layer for classification. In CNN, all convolution kernel sizes are 5 × 5, stride is 1, the pooling layer uses the max pooling size of 2 × 2. The details are shown in Table 1.

5 Experimental results and analysis

5.1 Data set

This section describes the UCI data sets Iris, Wine quality-red and handwritten numerical recognition of MNIST. Among them, Iris contains 150 samples, a total of 3 categories, corresponding to each row of data in the data set; the classes of Wine quality-red are ordered and not balanced; MNIST is composed of 60,000 training pictures and 10,000 test pictures, a total of 10 categories. Each picture is 28 × 28 in size, and both are black and white. It is a 0–1 floating point number. The darker the black color is, the closer the value is to 1. Table 2 shows the detailed description of the three data sets.

5.2 Experimental parameters

Since the experiments are performed on three data sets, in order to achieve the comparability of the experimental results, the base classifiers with the same parameters are used for the same data set. For the three data sets, the parameters of these base classifiers are the same: decision tree uses entropy to select features; MLP has 2 layers, 100 neurons in the first layer and 50 neurons in the second layer, the loss function is stochastic gradient descent, and the activation function is ReLU; the K-Nearest Neighbor has a k value of 5; random forest uses bootstrap samples

Table 1 Structure and parameters of the CNN

Number	Layers	Input size	Output size	Kernel size
1	C1	28 × 28 × 1	28 × 28 × 16	5 × 5 × 16
2	S2	28 × 28 × 16	14 × 14 × 16	2 × 2
3	C3	14 × 14 × 16	14 × 14 × 32	5 × 5 × 32
4	S4	14 × 14 × 32	7 × 7 × 32	2 × 2
5	C5	7 × 7 × 32	7 × 7 × 256	5 × 5 × 256
6	F6	7 × 7 × 256	1 × 1 × 784	
7	EL	1 × 1 × 784	1 × 1 × 10	

Table 2 Details of the three data sets

Number	Detail	Iris	Wine quality-red	MNIST
1	Instances	150	1599	70,000
2	Attributes	4	11	784
3	Categories	3	6	10
4	Training set	105	1120	60,000
5	Test set	45	479	10,000

and selects features based on Gini coefficient, while using out-of-bag samples to estimate generalization accuracy; XGBoost uses general balanced trees as the base classifier. The parameters of these base classifiers are different: For Iris, the kernel function of SVM is the radial basis function; the kernel function of Naive Bayes is Polynomial. For MNIST, the kernel function of SVM is Polynomial; the kernel function of Naive Bayes is Gaussian. For Wine quality-red, the kernel function of SVM is Sigmoid; the kernel function of Naive Bayes is Polynomial.

5.3 Evaluation

The evaluation of the model uses a confusion matrix. The confusion matrix consists of four important indicators: TP (true positives), which means that it is a positive example and is also recognized as a positive proportion; FP (false positives), which means itself is a negative example, but is identified as a positive proportion; FN (false negatives), which means that it is a positive example, but it is recognized as a negative proportion; and TN (true negatives), which means that it is a negative example, and it is also recognized as a negative proportion. According to these parameters, the four evaluation indicators of accuracy, precision, recall and F1 score are used in this experiment. The specific calculation is as shown in Eqs. (8)–(11).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{8}$$

Table 3 Accuracy of single classifier

Number	Classifier	Iris	Wine quality-red	MNIST
1	Decision tree	0.956	0.592	0.886
2	SVM	0.955	0.431	0.971
3	MLP	0.955	0.494	0.929
4	NB	0.955	0.542	0.837
5	KNN	0.967	0.504	0.969
6	Perceptron	0.91	0.463	0.952

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$F1\ \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

5.4 An ensemble learning framework based on different base classifiers

Ensemble learning is an algorithm that combines multiple base classifiers to get a more comprehensive and powerful classifier; thus, this section explains the following experiments on the Iris, Wine quality-red and MNIST data sets. First, a single classifier is used for classification. Then classification is performed using Bagging and Boosting based on different base classifiers. The accuracy (%) is shown in Tables 3, 4 and 6. Finally, the accuracy (%) for the random forest and XGBoost is given in Tables 5 and 7. And Fig. 6 shows a graph of the relationship between the accuracy and the number of base classifiers.

Table 3 shows that for a small data set Iris, the accuracy of all single classifiers is more than 90% and KNN has the highest accuracy 96.7%; for the unbalanced data set Wine quality-red, the accuracy is relatively low, between 40 and 50%, decision tree has the highest accuracy 59.2%; for the largest picture data set MNIST, the correct rate is more than 80% and SVM has the highest accuracy 97.1%.

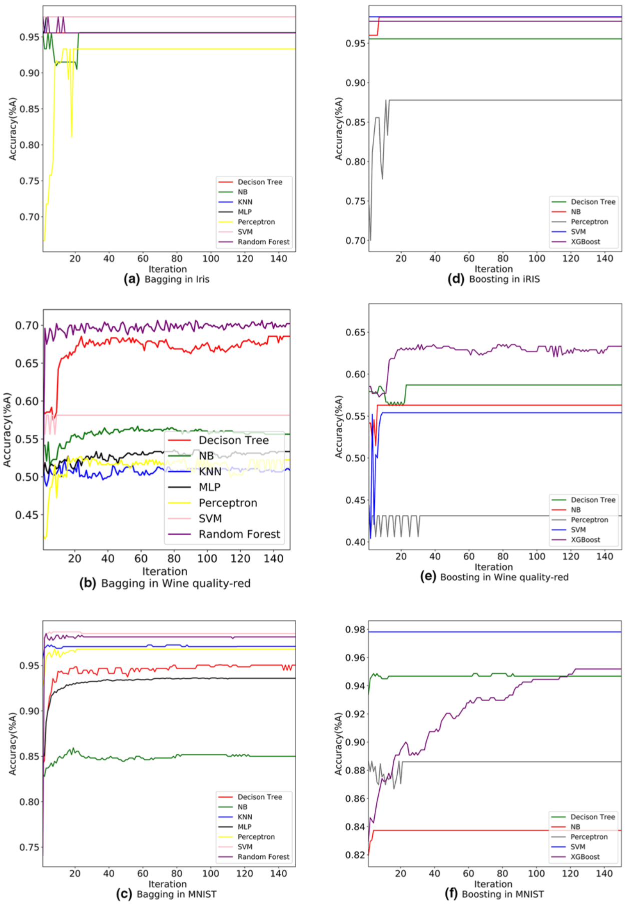
Combined with Tables 4 and 5, it can be seen that for Iris and MNIST, the accuracy of most algorithm is more than 90%, which is a few percentage points higher than the accuracy in Table 3. KNN and SVM also have the highest accuracy 97% and 98.5, respectively; for unbalanced Wine quality-red, the accuracy is 50–60%, which is almost 10% higher than single classifier. Although decision tree is lower than the highest accuracy of random forest, it also reached 68.7%. In addition, random forest has the highest accuracy rate of 70% for the unbalanced data set Wine quality-red and the performance of Iris and MNIST is also great. Therefore, random forest has better robustness.

Table 4 Accuracy of Bagging

Number	Base classifier	Iris	Wine quality-red	MNIST
1	Decision tree	0.956	0.687	0.95
2	SVM	0.977	0.586	0.985
3	MLP	0.977	0.533	0.935
4	NB	0.956	0.56	0.857
5	KNN	0.977	0.508	0.971
6	Perceptron	0.933	0.511	0.97

Table 5 Accuracy of random forest

Number	Algorithm	Iris	Wine quality-red	MNIST
1	Random forest	0.977	0.7	0.981



(a) Bagging in Iris

(d) Boosting in iRIS

(b) Bagging in Wine quality-red

(e) Boosting in Wine quality-red

(c) Bagging in MNIST

(f) Boosting in MNIST

Fig. 6 Relationship between accuracy and the number of base classifier

Table 6 Accuracy of Boosting

Number	Base classifier	Iris	Wine quality-red	MNIST
1	Decision tree	0.95	0.587	0.946
2	SVM	0.983	0.554	0.978
3	NB	0.983	0.563	0.837
4	Perceptron	0.877	0.438	0.886

Table 7 Accuracy of XGBoost

Number	Algorithm	Iris	Wine quality-red	MNIST
1	XGBoost	0.977	0.633	0.954

Combined with Tables 6 and 7, it can be concluded that for Iris, the accuracy of most Boosting is more than 90%, which is a few percentage points higher than the accuracy in Table 3, except Boosting based on Perceptron; for Wine quality-red, the accuracy of most algorithm is more than 50%, and Boosting based on Perceptron has a lower accuracy 43.8%; for MNIST, the accuracy is more than 80% and SVM has the highest accuracy 97.8%. XGBoost achieved the best results on the Wine quality-red data set compared to other Boosting methods. It is as robust as random forest.

In Fig. 6, the first column is a graph of the accuracy and the number of base classifiers based on Bagging framework for three data sets; the second column is a graph of the accuracy and the number of base classifiers based on Boosting framework for three data sets. Among them, the base classifier iterates once for every 50 increments. As can be seen that for the small-scale data set Iris, only a few classifiers can achieve the highest classification accuracy; for the unbalanced data set Wine quality-red, Boosting framework converges faster than Bagging framework. Besides, random forest and XGBoost have better robustness; for MNIST, Boosting framework is also faster than Bagging framework convergence, and only a few classifiers can achieve better accuracy.

In summary, the accuracy of Bagging framework and Boosting framework based on different base classifiers is improved relative to a single classifier. The accuracy of Boosting framework will be a little lower than Bagging framework on the whole, but it has a faster convergence. Random forest and XGBoost have great performance of all data sets, especially on the unbalanced data set Wine quality-red. This shows that random forest and XGBoost have better robustness.

5.5 An ensemble learning framework for CNN based on different base classifiers

If the extracted features are not comprehensive, how to adjust the parameters of the classification does not reach an ideal state. On the contrary, if the extracted features are great, it is easy to get a higher accuracy. Therefore, in this experiment, first use CNN with superior feature extraction ability to extract the features from MNIST image data set and then use the ensemble learning algorithm of Sect. 5.3 to classify and obtain the accuracy and precision. The experimental results are shown in Tables 8, 9, 10, 11. And Fig. 7 shows a graph of the relationship between the accuracy and the number of base classifiers.

Combined with Tables 8 and 9, it can be found that the accuracy of CNN + Bagging framework is above 98%, which is significantly improved compared with the single Bagging framework; in particular, NB is the most obvious. According to F1 score, the classification performance of SVM-based Bagging framework is the best.

Combined with Tables 10 and 11, it can be discovered that the accuracy of CNN + Boosting framework is above 97%, which also has a conspicuous improvement over Boosting framework. According to F1 score, SVM-based Bagging framework has the best performance and XGBoost is close behind.

In Fig. 7, record a value every 50 iterations. It can be seen that the accuracy of using CNN + ensemble learning framework is higher than single ensemble learning framework, and after using CNN to extract features, the gap between each ensemble learning is not large as the number of iterations increases. However, the accuracy of Perceptron-based CNN + Boosting is still lower than other algorithms, consistent with the results of the single Boosting framework.

In summary, the accuracy of the features extracted from convolutional neural network using the ensemble learning framework is higher than the accuracy using only ensemble learning framework. In the case of using convolutional neural network at the same time, the results between each ensemble learning framework are not much different.

Table 8 Accuracy of CNN + Bagging

Number	Base classifier	Accuracy	Precision	Recall	F1 score
1	Decision tree	0.981	0.97	0.829	0.894
2	SVM	0.992	0.981	0.939	0.96
3	MLP	0.99	0.979	0.914	0.945
4	NB	0.98	0.948	0.817	0.885
5	KNN	0.992	0.971	0.911	0.94
6	Perceptron	0.979	0.974	0.907	0.939

Table 9 Accuracy of CNN + random forest

Number	Base classifier	Accuracy	Precision	Recall	F1 score
1	Random forest	0.991	0.952	0.948	0.95

Table 10 Accuracy of CNN + Boosting

Number	Base classifier	Accuracy	Precision	Recall	F1 score
1	Decision tree	0.97	0.969	0.816	0.886
2	SVM	0.979	0.958	0.923	0.94
3	NB	0.956	0.869	0.896	0.882
4	Perceptron	0.876	0.832	0.87	0.851

Table 11 Accuracy of CNN + XGBoost

Number	Base classifier	Accuracy	Precision	Recall	F1 score
1	XGBoost	0.973	0.943	0.921	0.932

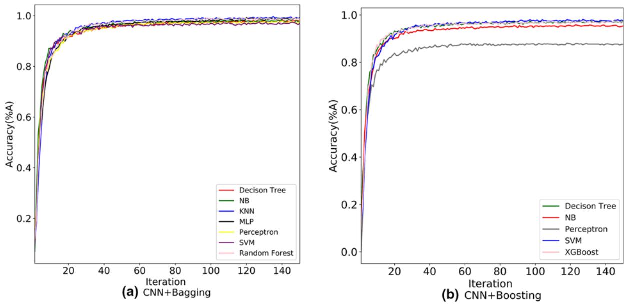


Fig. 7 Relationship between accuracy and the iteration

6 Conclusion

Because the ensemble learning algorithm can achieve better accuracy than a single traditional classifier and CNN can extract more comprehensive and deeper features, this paper proposes an ensemble learning framework for convolutional neural networks based on multiple classifiers. First, based on the Bagging and Boosting algorithms, the three data sets Iris, Wine quality-red and MNIST are classified using an ensemble learning algorithm that is integrated using different single classifiers. The results show that compared with the single classifier, this method improves the accuracy. Although the accuracy of the Bagging framework is generally higher than that of the Boosting framework, the Boosting framework can converge faster. Then using the CNN to extract the features from MNIST data set, the above-mentioned ensemble learning framework is used to classify the extracted features. The results show that the classification effect obtained is better than the single ensemble learning algorithm framework. At the same time, the model converges faster when using CNN to extract features. In addition, it can be found that random forest and XGBoost have better robustness. In the future, the work of this paper will focus on verifying larger data sets.

Acknowledgements This work is supported by the National Natural Science Foundation (Nos. 61672522, 41704115), the Opening Project of Key Laboratory of Data Science and Intelligence Application [No. D1804], and Jiangsu Graduate Research and Innovation Project [No. SJKY19-1889].

Compliance with ethical standards

Conflict of interest Yanyan Guo, Xin Wang, Pengcheng Xiao and Xinzheng Xu declare that they have no conflict of interest.

Informed consent Informed consent was not required as no human or animals were involved.

Human and animal rights This article does not contain any studies with human or animal subjects performed by the any of the authors.

References

Adama DA, Lotfi A, Langensiepen CS, Lee K, Trindade P (2018) Human activity learning for assistive robotics using a classifier ensemble. Soft Comput 22(21):7027–7039

Breiman L (1996) Bagging predictors. Int J Mach Learn 24(2):123–140

Breiman L (2001) Random Forests. Int J Alg 45(1):5–32

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth. ISBN 0-534-98053-8

Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 785–794

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27

Dasarathy BV, Sheela BV (1979) A composite classifier system design: concepts and methodology. Proc IEEE 67(5):708–713

Ding Z, Fei M, Dajun D, Yang F (2017) Streaming data anomaly detection method based on hyper-grid structure and online ensemble learning. Soft Comput 21(20):5905–5917

Freund Y, Schapire RE (1995) A decision-theoretic generalization of on-line learning and an application to boosting. EuroCOLT 1995:23–37

He K, Gkioxari G, Dollár P, Girshick RB (2017) Mask R-CNN. In: ICCV 2017, pp 2980–2988

Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Int J Neural Comput 18(7):1527–1554

Hinton G, Deng L, Yu D, Mohamed A-R, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath T, Dahl G, Kingsbury B (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag 29(6):82–97

Ji S, Wei S, Meng L (2019) Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. IEEE Trans Geosci Remote Sens 57(1):574–586

Krizhevsky A, Sutskever I, Hinton G (2012) Imagenet classification with deep convolutional neural networks. NIPS 25:1106–1114

Lewis DD (1998) Naive (Bayes) at forty: the independence assumption in information retrieval. In: The 10th Euro-pean conference on machine learning, New York, Springer, pp 4–15

Longstaff ID, Cross JF (1987) A pattern recognition approach to understanding the multi-layer perception. Pattern Recogn Lett 5(5):315–319

Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386–408

Schapire RE (1989) The strength of weak learnability (Extended Abstract). FOCS 1989:28–33

van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior AW, Kavukcuoglu K (2016) WaveNet: a generative model for raw audio. CoRR abs/1609.03499

Wang T, Zhang Z, Jing X, Zhang L (2016) Multiple kernel ensemble learning for software defect prediction. Autom Softw Eng 23(4):569–590

Zhang L, Shah SK, Kakadiaris IA (2017) Hierarchical multi-label classification using fully associative ensemble learning. Pattern Recogn 70:89–103

Zhang S, Zhang S, Huang T, Gao W (2018) Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching. IEEE Trans Multimed 20(6):1576–1590

Zhiwen Yu, Wang D, Zhuoxiong Zhao CL, Chen P, You J, Wong H-S, Zhang J (2019) Hybrid incremental ensemble learning for noisy real-world data classification. IEEE Trans Cybern 49(2):403–416

Zhou Y, Wang P (2019) An ensemble learning approach for XSS attack detection with domain knowledge and threat intelligence. Comput Secur 82:261–269