Neural Computing and Applications (2021) 33:9703–9721
https://doi.org/10.1007/s00521-021-05736-x

ORIGINAL ARTICLE

Signals classification based on IA-optimal CNN

Yalun Zhang[1] · Wenjing Yu[1] · Lin He[1] · Lilin Cui[1] · Guo Cheng[1]

Received: 22 October 2020 / Accepted: 16 January 2021 / Published online: 27 May 2021

Abstract

The versatility of the existing A-optimal-based CNN for solving multiple types of signals classification problems has not been verified by different signals datasets. Moreover, the existing A-optimal-based CNN uses a simplified approximate function as the optimization objective function instead of precise analytical function, which affects the signals classification accuracy to a certain extent. In this paper, a classification method called IA-optimal CNN is proposed. To improve the stability of the classifier, the trace of the covariance matrix of the weights of the fully connected layer is used as the optimization objective function, and the parameter optimization model is established without any simplification of the optimization objective function. In addition, to avoid the difficulty of not being able to obtain the analytical expression formula of the partial derivative of the inverse matrix with regard to the networks parameters, a novel dual function is introduced to transform the optimization problem into an equivalent binary function optimization problem. Furthermore, based on the above analytical solution results, the parameters are updated using the alternate iterative optimization method and the accurate weight update formula is deduced in detail. Five signals datasets are used to test the universality of the IA-optimal CNN in signals classification fields. The performance of IA-optimal CNN is showed, and the experimental results are compared with the existing A-optimal-based classification algorithm. Lastly, the following conclusion is proved theoretically: For the A-optimal-based CNN, the trace of the covariance matrix will continue to decrease and approach a convergence value in the iterative process, but it is impossible for the networks to strictly reach the A-optimal state.

Keywords A-optimal · Convolutional neural networks · Signals classification · Alternate iterative optimization · Dual function

1 Introduction

The classification and recognition of signals data can mine the hidden information in the signals, thereby providing the application basis for various fields. For example, mechanical equipment will vibrate due to the operation of parts when working. The equipment vibration signals under negative conditions and the signals under normal operating conditions often have different characteristics. Therefore, vibration signals analysis for power equipment can distinguish different operating conditions of equipment, thereby helping people to diagnose faults and find the cause [1–14]. Different types of sound signals are closely related to the characteristics of different space or objects. The sound quality classification assessment of public places is an important technical background of noise control technology study. In the field of medical applications, in addition to diagnosing heart diseases with the help of electrocardiogram, the acoustic signals of different heart sounds also reflect the health of the heart. The sound signals of human speaking can be used for voice recognition, which is an important research topic in the field of human–computer interaction [15–21]. The human brain is composed of a large number of neurons, and EEG (Electroencephalogram)

✉ Guo Cheng
zcz@hust.edu.cn

Yalun Zhang
carterdickson@163.com

Wenjing Yu
ywjada@163.com

Lin He
1794975283@qq.com

Lilin Cui
cuililinwyt@163.com

[1] Institute of Noise & Vibration, Naval University of Engineering Hubei, Wuhan, China

is the electrical signals generated by the activities between these neurons. Relevant research and experiments have proved that there is a specific connection between the EEG potential and the type of brain thinking tasks or the external stimuli received. Moreover, the EEG characteristics caused by different brain thinking tasks and external stimuli are also different. Based on above research, brain-computer interface (BCI) technology can be used to identify the thinking intention of the subject by analyzing EEG. Then, this intention can be converted into instruction signals, and mechanical equipment can help people complete a series of tasks by identifying the instruction signals [22–33].

The purpose of signal classification is to identify the real category of signals with unknown labels. Traditional signals classification methods based on feature extraction can be divided into two steps: (1) Classifier training based on training dataset: The signals features of the training dataset will be extracted, and the extracted feature parameters will be input to the initial classifier. The classifier will continuously optimize the parameters according to the error between the classification results and the actual labels, so that the classifier will converge and reach a certain classification accuracy standard; (2) Signals classification based on testing dataset: The signals features of the testing dataset will be extracted, and the extracted feature parameters will be input to the trained classifier. The classifier will recognize the signals according to the input

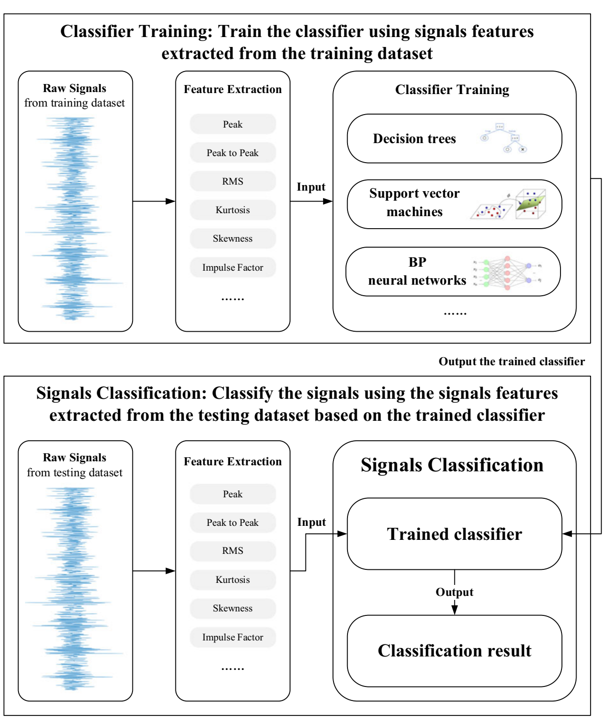

Fig. 1 Traditional feature extraction-based signals classification method process

characteristic parameters. The above classification process is shown in Fig. 1.

Both above two steps need to perform feature extraction from the training dataset and the testing dataset. Feature extraction is a preprocessing of the data. For the signals classification problem, the raw data are a vector composed of many vibration amplitudes or electric potential amplitudes. The raw signals without any processing have long data length and large data scale, so it cannot show clear signals characteristics. Therefore, the raw data are generally not directly input to the classifier. Researchers often first extract some feature parameters that can reflect the characteristics of the signals, and then input these feature parameters into the classifier. The classifier will recognize the type of signals based on the different feature parameters. For example, the peak-to-peak represents the difference between the maximum value and minimum value of the signals, which reflects the fluctuation range of the signals. The kurtosis of the signals measures the steepness of the signals waveform. The kurtosis of the waveform which is similar to the normal distribution function is approximately equal to 3, and the kurtosis of a smoother signals is generally less than 3. The RMS (Root Mean Square) of the signals is used to measure the amount of energy contained in the signals.

However, an inconvenience of the traditional method is that there are many features that can be extracted from the signals, such as peak, peak to peak, RMS, kurtosis, skewness, impulse factor, et al. [33–37]. In different application scenarios, the types and quantities of extracted signals features need to be selected based on historical experience. Therefore, for classification problems without prior knowledge, people do not know in advance which features are helpful for signals classification. At this time, it is necessary to manually complete a large number of feature extractions, and the different signals feature combinations will be input into the classifier to test the usability of this combination. Finally, the available feature combinations will be selected according to the discrimination accuracy of the classifier, which greatly increases labor costs and time costs.

Compared with traditional classification methods, CNN (Convolutional Neural Networks) not only has excellent classification and prediction capabilities, but also can automatically extract signals features by special operations such as convolution and pooling. In this way, it is no longer necessary to test the availability of different feature parameters one by one, which greatly reduces the cost of labor and time. CNN first completes the feature extraction of the input data by convolution and pooling operations. Then, it uses the fully connected layer to complete the data classification. When the output results deviate from the real situation, the networks parameters will be continuously updated to make the networks converges adaptively and finally approach the correct classification conclusion. Therefore, since the advent of CNN, people have begun to carry out CNN-based application research in many fields such as Internet behavior detection, image recognition, and signals classification [38–40]. Currently, the world is facing the risk of a Covid-19 pandemic. CNN has even been applied to the diagnosis of Covid-19, showing the strong vitality and adaptability of CNN in solving new problems in new fields [41].

A strategy of the CNN training is that the networks parameters are constantly updated according to the value of loss function. However, due to the different characteristics of different training datasets, networks parameters often be too sensitive to training samples. In particular, the weight of the fully connected layer is likely to have huge differences due to the changes of input dataset. The general existing CNN parameter training models are usually focus on the value of loss function. In this case, the classifier parameters of the fully connected layer will be highly sensitive to samples. However, an excellent classifier should be stable when different training datasets are input to the algorithm [51]. For a classifier optimization model, in the most ideal case, the stability of the networks under various training sample conditions can be maintained when the loss function value is as small as possible.

A-optimal method is a parameter optimization method to improve the stability of the classifier [42–46]. The optimization goal of the A-optimal method is the trace of the covariance matrix of the classification model parameters. The diagonal elements of the covariance matrix of the classifier weights are the variances of the weights. So, the trace of the covariance matrix of the weights is the sum of the weight variances. The variance of the weight measures the dispersion of the weight. The smaller the sum of the weight variances, the smaller the overall volatility of the weights, and the higher the stability of the classifier parameters. He et al. proposed an image classification algorithm based on A-optimal subspace learning [47], which optimizes the parameters by minimizing the trace of the covariance matrix of the classifier parameters, thereby improving the stability of the classifier. Liu et al. proposed the A-optimal NMF model to train the classifier for image recognition. This method introduces nonnegative matrix factorization and uses the A-optimal method to update the parameters of the linear classifier [48]; Li et al. [49] and Yang et al. [50], respectively, introduce the neighborhood regularization and hessian regularization to improve the classification recognition algorithm based on A-optimal. In order to solve the problem that the existing A-optimal classification method can only deal with single-instance and linearly separable problems, Yin introduced the A-optimal method to CNN for the first time to solve the multi-instance classification question. Moreover, compared with existing A-optimal classification methods, the A-optimal-based CNN method is stable for multi-instance learning and has the higher classification accuracy [51].

In the method proposed by Yin, the trace of the covariance matrix of the weight of the fully connected layer is used as the optimization objective function, and the gradient calculation of objective function with regard to the classifier parameters is based on the above objective function [51]. In this part, Yin simplified the formula of the trace of the covariance matrix and replaced the original optimization objective function with the simplified formula. Therefore, the calculation of the gradient is not based on the precise optimization objective function but on an approximate function, which may affect the final accuracy of the model. In addition, Yin's literature uses only one signals dataset to test the performance of different algorithm, but the conclusions based on a single type of sample dataset may not be comprehensive and accurate. Finally, Yin introduced the A-optimal method to CNN and gave the implementation steps of the A-optimal CNN method, but did not discuss whether the A-optimal-based CNN can finally reach the A-optimal state when the algorithm training ends.

This paper proposes an Improved A-optimal CNN method. Hereinafter, the "Improved A-optimal CNN" proposed in this paper will be referred to as "IA-optimal CNN" for short. In the following, the IA-optimal CNN architecture including the convolutional layer, the pooling layer, the activation layer, and the fully connected layer will be introduced, and its data processing to complete the multi-instance classification will be described. Thereafter, to maintain the stability of networks under the condition that the loss function value is as small as possible, the trace of the covariance matrix of the weight of the fully connected layer is used as the optimization goal and the goal of networks parameter update is to minimize the value of the trace. In this part, the calculation expression formula of the trace of the covariance matrix will not be simplified in any way, and the precise optimization objective function is used to derive the weight update formula in detail to eliminate the possible classification error caused by the simplified objective function. To avoid the problem that it is difficult to give an analytical solution to the partial derivative of the inverse matrix with regard to convolution kernel, a novel binary dual function is introduced and the optimization of the trace is transformed into an optimization problem. Then, the alternate iterative optimization method is adopted as the weight update strategy. In addition, different types of activation functions are test their ability to express nonlinear characteristics of signals. To test the algorithm performance and the universality in more signals classification fields, five signals datasets are used as samples datasets. The algorithm result is compared with existing A-optimal classification algorithms. Finally, the following question is discussed: In the iterative process of the A-optimal-based CNN algorithm, whether the networks can finally reach the optimal A-optimal state.

The main improvements of the model proposed in this paper and the corresponding work focus are shown in Fig. 2.

2 IA-optimal CNN method

The proposed IA-optimal CNN method will be introduced in detail in this chapter, which includes three parts: (1) Architecture of IA-optimal CNN: In this section, the architecture of IA-optimal CNN will be showed. IA-optimal CNN includes the input layer, the convolutional layer, the activation layer, the pooling layer, the fully connected layer, and the output layer. The data processing and calculation rules of different layer will be introduced, respectively; (2) Optimization goal: In this section, the trace of the covariance matrix of the weight of the fully connected layer will be used as the optimization goal to train network parameters. To ensure the mathematical rigor of the algorithm, the subsequent formulas will be derived based on the precise optimization objective function without simplification. In addition, a novel dual function is introduced as the optimization strategy to avoid calculation difficulties during the formula derivation, which provide a theoretical basis for network parameter training; (3) Training strategy: In this section, based on the conclusions of the previous section, the original optimization problem is transformed into a binary matrix function optimization problem. The alternating iteration method is used to train the network parameters, and the algorithm steps for parameter updating are explained.

2.1 Architecture of IA-optimal CNN

The IA-optimal CNN includes the input layer, the convolutional layer, the activation layer, the pooling layer, the fully connected layer, and the output layer. The networks is oriented to multi-instance learning and recognition. The input of the networks is a dataset including a number of signals (bag), and the output of the networks is a sequence of classification results of the input signals dataset. In a bag, if the classification result of any instance (a sub-sequence signals) is positive, the classification result of this bag is positive; if the classification result of all the instances is negative, the classification result of this bag is negative. The architecture of IA-optimal CNN is shown in Fig. 2.

Input Layer: The input layer is used to accept input signals. Suppose that the dataset input to the networks

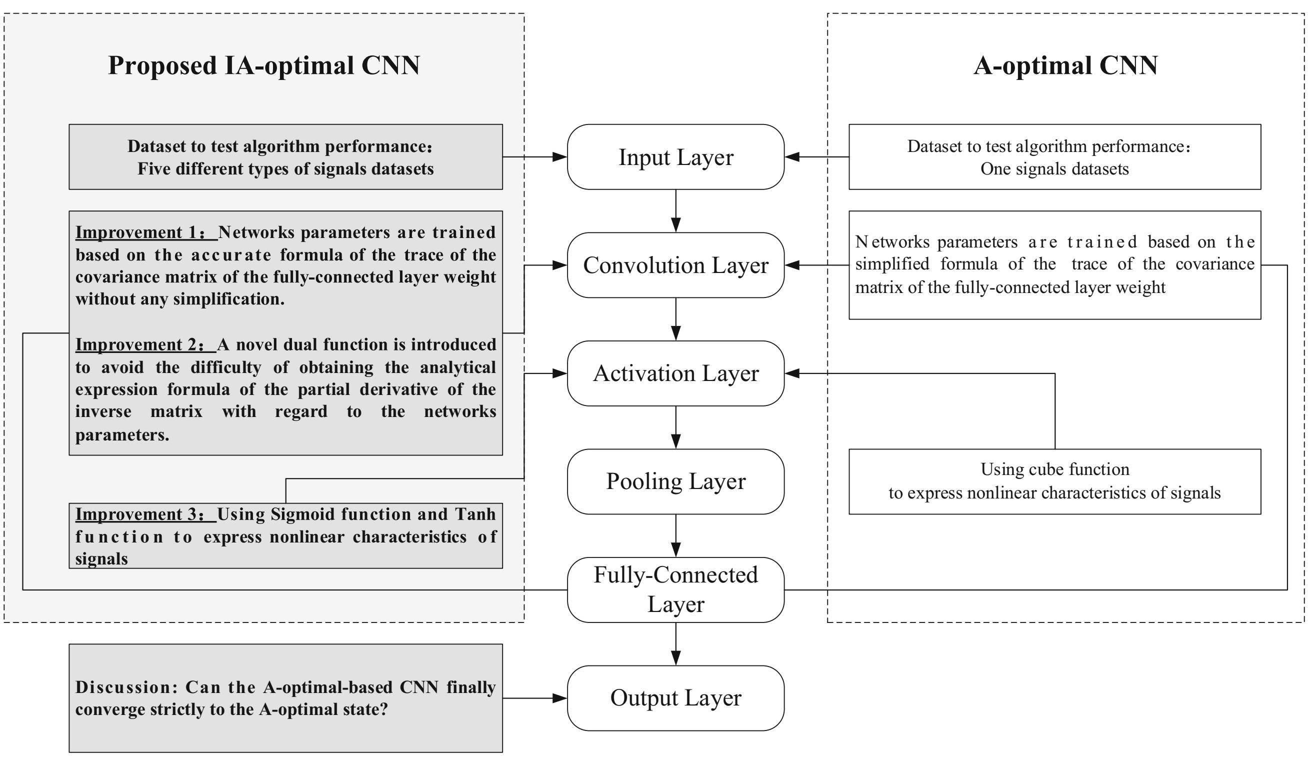

Fig. 2 The main improvements of the model proposed in this paper and the corresponding work focus

contains n signals whose sampling points are all D, and each signals S_i is regarded as a bag. Each signals will be divided into l equal parts by a sliding window, each subsequence signals is an instance, and the number of sampling points in each instance is d. Then, the j-th subsequence signals of the i-th sample signals can be denoted as x_{ij}. For the convenience of subsequent calculations, the l sub-sequences are combined by column into a matrix, then each matrix X_i can represent a bag, denoted as $X_i = [x_{i1}\, x_{i2} \cdots x_{il}] \in R^{d\times l}$, $i = 1, 2, \cdots n$.

Convolution Layer: The convolution layer performs convolution operations on the input data. The output of convolution layer is a feature map. Suppose that the convolutional layer contains m convolution kernels, and each convolution kernel w_k is a column vector, denoted as $w_k \in R^{d\times 1}$, $k = 1, 2, \cdots m$. For the convenience of subsequent calculations, the m convolution kernels are combined by column into a matrix, then the kernels of the convolution layer can be expressed as a convolution kernel matrix, denoted as $W = [w_1\, w_2 \cdots w_m] \in R^{d\times m}$. After the convolution kernels of m channels are, respectively, convolved with X_i, the m feature row vectors $c_i \in R^{1\times l}$ $(i = 1, 2, \cdots n)$ will be obtained. The above row vectors are combined by row into a matrix, then the output of X_i after the convolution operation can be expressed as an matrix C_i, denoted as:

$$C_i = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} w_1^T x_{i1} & w_1^T x_{i2} & \cdots & w_1^T x_{il} \\ w_2^T x_{i1} & w_2^T x_{i2} & \cdots & w_2^T x_{il} \\ \cdots & \cdots & \cdots & \cdots \\ w_m^T x_{i1} & w_m^T x_{i2} & \cdots & w_m^T x_{il} \end{bmatrix} = W^T X_i \tag{1}$$

Activation Layer: An activation function $\sigma(\cdot)$ is used to process the matrix C_i so that the networks can express nonlinear characteristics of the signals. After using the activation function $\sigma(\cdot)$ for each element in the matrix C_i, the output of the activation layer is obtained, denoted as $D_i = \sigma(C_i) \in R^{m\times l}$.

Pooling Layer: The pooling layer performs down-sampling operations on the feature maps output by the convolutional layer to obtain feature vectors that can be used for classification and discrimination. The pooling size is $1 \times l$, the row stride is 1, and the maximum pooling is selected as the pooling method. A column feature vector z_i of size $m \times 1$ can be obtained:

$$z_i = \begin{bmatrix} \max_j\{\sigma(c_1)\} \\ \max_j\{\sigma(c_2)\} \\ \vdots \\ \max_j\{\sigma(c_m)\} \end{bmatrix} = \begin{bmatrix} \max_j\{\sigma(w_1 x_{ij})\} \\ \max_j\{\sigma(w_2 x_{ij})\} \\ \vdots \\ \max_j\{\sigma(w_m x_{ij})\} \end{bmatrix} \tag{2}$$

Input n sample signals (n bags) in a batch, and then n column feature vectors z_i can be obtained. The above vectors are combined by column into a matrix, the output of n sample signals through the pooling layer can be expressed as an output matrix $Z = [z_1, z_2, \cdots z_n] \in R^{m\times n}$.

Fully Connected Layer: The fully connected layer reduces the dimensionality of the output vector of the pooling layer and initially completes the signals classification. The classification of each input sample S_i is a binary discrimination, and the output is a Boolean scalar b_i. The calculation method of the output b_i of the fully connected layer is $b_i = \omega^T z_i$, ω is the fully connected layer weight. '$b_i = 1$' means 'the sample S_i is judged as positive', and '$b_i = 0$' means 'the sample S_i is judged as negative.' In different questions, the meaning of positive or negative is different.

Output Layer: The real category label of sample S_i is y_i. '$y_i = 1$' means 'sample S_i is a positive signals' and '$y_i = 0$' means 'sample S_i is a negative signals.' Then, the loss function for evaluating the classification accuracy of sample S_i is:

$$\mathrm{error}(S_i) = (y_i - \omega^T z_i)^2 \tag{3}$$

If n samples (n bags) are input in a batch, the classification label of each sample can be obtained. The n Boolean scalars are combined into a row vector b, the total loss function is:

$$\mathrm{error}(S) = \sum_{i=1}^{n} (y_i - \omega^T z_i)^2 = \sum_{i=1}^{n} (y_i - b_i)^2 \tag{4}$$

In order to avoid the over-fitting, an L2 regularization term is added to the total loss function to limit the weight. Finally, the loss function used in the IA-optimal CNN method is:

$$\begin{aligned} \mathrm{loss}(S) &= \sum_{i=1}^{n} (y_i - \omega^T z_i)^2 + \alpha\|\omega\|_2^2 \\ &= \left\|y - \omega^T Z\right\|_2^2 + \alpha\|\omega\|_2^2 \end{aligned} \tag{5}$$

where, $\|\cdot\|_2$ is the 2-norm, α is the L2 regularization coefficient, and $y = (y_1, y_2, \cdots y_n)$ is the label row vector of n samples.

IA-optimal CNN needs to be continuously trained to minimize the total loss function value of n samples input in a batch, so the algorithm parameters need to be continuously updated. Ideally, we hope to directly obtain the weight ω_0 that minimizes loss*(S)*:

$$\omega_0 = \arg\min_{\omega}\left\{loss(S) = \left\|y - \omega^T Z\right\|_2^2 + \alpha\|\omega\|_2^2\right\} \tag{6}$$

It should be noted that although the architecture of the networks, the form of the loss function, and the selection of the regularization method can be changed, these changes may make it difficult to obtain the analytical expression of the partial derivative of the optimization objective function with regard to the networks' parameter in the subsequent derivation process. The analytical expression formula of the partial derivative is necessary to achieve A-optimal state, so the networks design has its unique characteristics.

2.2 Optimization goal

In order to continuously update the weight ω_0 of the fully connected layer and the convolution kernel matrix W, it is necessary to derive the mathematical relationship between loss*(S)*, W and ω_0. In this section, the weight ω_0 will be derived when loss*(S)* takes the minimum value. According to the A-optimal criterion, the optimization goal of the convolution kernel matrix W is to take the minimum value of the trace of the weight covariance matrix $Tr(cov(\omega_0))$. In addition, a novel dual function is introduced to transform this optimization problem into a problem of finding the minimum value of a binary function. This approach will provide important support for the gradient calculation in Sect. 2.3.

2.2.1 Optimization goal: trace of the covariance matrix of the weight

The loss function can be written as follows:

$$\begin{aligned}\text{loss}(S) &= \left\|y - \omega^T Z\right\|_2^2 + \alpha\|\omega\|_2^2\\ &= (y - \omega^T Z)(y - \omega^T Z)^T + \alpha\omega^T\omega\\ &= yy^T - 2\omega^T Zy^T + \omega^T ZZ^T\omega + \alpha\omega^T\omega\end{aligned} \tag{7}$$

In order to obtain the weight ω_0 when loss*(S)* takes the minimum value, the derivative of loss*(S)* with regard to ω is set to 0.

$$\frac{\partial(\text{loss}(S))}{\partial\omega} = -2Zy^T + 2ZZ^T\omega + 2\alpha\omega = 0 \tag{8}$$

The analytical expression of ω_0 is

$$\omega_0 = (ZZ^T + \alpha I)^{-1} Zy^T \tag{9}$$

For a certain sample set $\{S\}$, y is an invariant, so only Z in formula (9) is a variable. Z is a function of the convolution kernel matrix W, so ω_0 is also a function of the convolution kernel matrix W.

The trace of the covariance matrix of the weight ω_0 can be written as follows:

$$\begin{aligned}\text{cov}(\omega_0) &= \text{cov}\left((ZZ^T + \alpha I)^{-1} Zy^T\right)\\ &= (ZZ^T + \alpha I)^{-1} Z\text{cov}(y^T) Z^T (ZZ^T + \alpha I)^{-1}\\ &= \sigma^2 (ZZ^T + \alpha I)^{-1} ZZ^T (ZZ^T + \alpha I)^{-1}\\ &= \sigma^2 (ZZ^T + \alpha I)^{-1} (ZZ^T + \alpha I - \alpha I)(ZZ^T + \alpha I)^{-1}\\ &= \sigma^2\left[(ZZ^T + \alpha I)^{-1} - \alpha(ZZ^T + \alpha I)^{-2}\right]\end{aligned} \tag{10}$$

where $\sigma^2 I = \text{cov}(y^T)$.

In the literature [51] published by Yin, in order to simplify the calculation, the above formula (10) is simplified, and the following conclusions are used:

$$\text{cov}(\omega_0) = \sigma^2 (ZZ^T + \alpha I)^{-1} \tag{11}$$

Moreover, Yin took function (11) as the optimization objective function to complete the subsequent formula derivation. This approach reduces the complexity of formula calculation to a certain extent, but may have a certain impact on the rigor of the conclusion.

In order to ensure the accuracy, this article does not simplify the formula, but complete the subsequent formula derivation based on the expression formula (10). Therefore, the trace of the covariance matrix of the weight ω_0 is:

$$Tr(\text{cov}(\omega_0)) = \sigma^2 Tr\left((ZZ^T + \alpha I)^{-1} - \alpha(ZZ^T + \alpha I)^{-2}\right) \tag{12}$$

In the process of networks training, we hope to find the convolution kernel matrix W that minimizes $Tr(\text{cov}(\omega_0))$:

$$W_0 = \arg\min_{W}\{Tr(\text{cov}(\omega_0))\} \tag{13}$$

In the following section, the detailed steps to find W_0 and the formula derivation process will be given.

2.2.2 Optimization strategy: dual function

Since σ^2 is a constant value, so the following conclusion is established:

$$W_0 = \arg\min_{W}\left\{Q(Z) = Tr\left((ZZ^T + \alpha I)^{-1} - \alpha(ZZ^T + \alpha I)^{-2}\right)\right\} \tag{14}$$

In order to find the convolution kernel matrix W_0 that minimizes $Tr(\text{cov}(\omega_0))$, a novel dual function $H(\varphi, Z)$ is introduced, and the expression is as follows:

$$H(\varphi, Z) = Tr\left(-\varphi(ZZ^T + \alpha I)^2\varphi^T + 2Z^T\varphi^T\right) \tag{15}$$

Theorem *$Q(Z)$ is the minimum value of the dual function $H(\varphi, Z)$ with regard to the matrix $\varphi \in R^{n\times m}$*

$$Q(Z) = \min_{\varphi \in R^{n\times m}} \left\{ H(\varphi, Z) = Tr\left(-\varphi(ZZ^T + \alpha I)^2\varphi^T + 2Z^T\varphi^T\right)\right\} \tag{16}$$

Proof In order to obtain the weight matrix $\varphi_0 \in R^{n\times m}$ when $H(\varphi, Z)$ takes the minimum value, the derivative of $H(\varphi, Z)$ with regard to φ is set to 0.

$$\frac{\partial H(\varphi, Z)}{\partial \varphi} = -2\varphi(ZZ^T + \alpha I)^2 + 2Z^T = 0 \tag{17}$$

The analytical expression of φ_0 is:

$$\varphi_0 = Z^T(ZZ^T + \alpha I)^{-2} \tag{18}$$

Substitute φ_0 into $H(\varphi, Z)$, the minimum value of $H(\varphi, Z)$ can be calculated as follows:

$$\begin{aligned} H(\varphi_0, Z) &= Tr\left(-\varphi_0(ZZ^T + \alpha I)^2\varphi_0^T + 2Z^T\varphi_0^T\right) \\ &= Tr\left(-Z^T(ZZ^T + \alpha I)^{-2}Z + 2Z^T(ZZ^T + \alpha I)^{-2}Z\right) \\ &= Tr\left((ZZ^T + \alpha I)^{-2}ZZ^T\right) \\ &= Tr\left((ZZ^T + \alpha I)^{-2}(ZZ^T + \alpha I - \alpha I)\right) \\ &= Tr\left((ZZ^T + \alpha I)^{-1} - \alpha(ZZ^T + \alpha I)^{-2}\right) \\ &= Q(Z) \end{aligned} \tag{19}$$

Thus, $Q(Z)$ is the minimum value of the dual function $H(\varphi, Z)$ with regard to the matrix $\varphi \in R^{n\times m}$. The optimization problem can be equivalently converted to the following problem:

$$\begin{aligned} W_0 &= \arg\min_{W \in R^{d\times m}} \left\{ \min_{\varphi \in R^{n\times m}} H(\varphi, Z) \right\} \\ &= \arg\min_{W \in R^{d\times m}} \left\{ \min_{\varphi \in R^{n\times m}} Tr\left(-\varphi(ZZ^T + \alpha I)^2\varphi^T + 2Z^T\varphi^T\right)\right\} \end{aligned} \tag{20}$$

The optimization problem can be regarded as the optimization of a binary matrix function.

$$\begin{aligned} W_0, \varphi_0 &= \arg\min_{W \in R^{d\times m}, \varphi \in R^{n\times m}} \{H(\varphi, Z)\} \\ &= \arg\min_{W \in R^{d\times m}, \varphi \in R^{n\times m}} \left\{Tr\left(-\varphi(ZZ^T + \alpha I)^2\varphi^T + 2Z^T\varphi^T\right)\right\} \end{aligned} \tag{21}$$

The significance of this step is that the inverse matrix $(ZZ^T + \alpha I)^{-1}$ is included in the analytical solution of *Tr*(cov(ω_0)). In order to calculate $\partial Tr(\mathrm{cov}(\omega_0))/\partial w_k$ to obtain the update strategy, the derivative of $(ZZ^T + \alpha I)^{-1}$ with regard to w_k must be calculated. Due to the existence of the inverse matrix, it is very difficult to directly obtain the analytical solution of $\partial Tr(\mathrm{cov}(\omega_0))/\partial w_k$. Therefore, the problem of finding the minimum value of *Tr(cov(ω_0))* needs to be transformed into the problem of finding the minimum value of its dual function $H(\varphi, Z)$. By doing so, the inconvenience of obtaining the derivative of the inverse matrix can be avoided.

Based on the conclusions of this section, the training process of the IA-optimal CNN method will be given in the next section, including the update strategy of the networks parameter and the detailed derivation process.

3 Training strategy

In this section, based on the dual function and its properties introduced in the previous section, the analytical formula of the gradient of the $H(\varphi, Z)$ with regard to the convolution kernel w_k will be derived, the minimum value of $H(\varphi, Z)$ will be found by the alternate iteration method, and the detailed steps of parameter updating based on the gradient descent method will be given.

3.1 Alternate iterative optimization

CNN can continuously adjust the networks parameters according to the loss function value during the training process, so the loss function value of CNN will continuously decrease and gradually converge. The commonly used gradient descent method is to make the networks parameters change in the direction of the gradient. Therefore, the IA-optimal CNN method also uses the gradient descent method to complete the parameter updating.

According to Sect. 2.2.1, the analytical expression of the weight ω_0 when loss(*S*) takes the minimum value is $\omega_0 = (ZZ^T + \alpha I)^{-1}Zy^T$. In the iterative process, the weight of the fully connected layer can be updated according to this formula. The weight ω_0 is a function of the convolution kernel matrix *W*, so the next round of convolution kernel matrix *W* can be adjusted according to the ω_0. The optimization goal of *W* is to make *Tr*(cov(ω_0)) as small as possible.

According to Sect. 2.2.2, the problem of finding the minimum value of *Tr*(cov(ω_0)) has been transformed into the problem of finding the minimum value of $H(\varphi, Z)$. $H(\varphi, Z)$ is a binary matrix function, so the alternate iteration method is used for optimization in parameter update. The main steps are as follows:

1. After the end of the t-th iteration, calculate *loss(S)* and update the weight ω_{t+1} of the fully connected layer according to formula (9);
2. Keep the convolution kernel matrix W_t unchanged, and update φ_{t+1} according to formula (18);

3. Keep φ_{t+1} unchanged and update w_k according to the gradient descent method:

$$w_k(t+1) = w_k(t) - \beta \frac{\partial H(\varphi_{t+1}, Z)}{\partial w_k(t)} \tag{22}$$

where, β is the learning rate coefficient, and W_{t+1} can be updated by combining all the $w_k(t+1)$. In the next section, the calculation process of the partial derivative of the dual function with regard to the convolution kernel w_k will be showed.

3.2 Derivative of the dual function with regard to the convolution kernel

The $H(\varphi, Z)$ can be written as follows:

$$\begin{aligned} H(\varphi, Z) &= Tr\left(-\varphi(ZZ^T + \alpha I)^2\varphi^T + 2Z^T\varphi^T\right) \\ &= Tr\left(-\varphi ZZ^T ZZ^T\varphi^T - 2\alpha\varphi ZZ^T\varphi^T - \alpha^2\varphi\varphi^T + 2Z^T\varphi^T\right) \\ &= Tr\left(-ZZ^T ZZ^T\varphi^T\varphi - 2\alpha ZZ^T\varphi^T\varphi - \alpha^2\varphi\varphi^T + 2Z^T\varphi^T\right) \\ &= -Tr(ZZ^T ZZ^T\varphi^T\varphi) - 2\alpha Tr(ZZ^T\varphi^T\varphi) \\ &\quad - \alpha^2 Tr(\varphi\varphi^T) + 2Tr(Z^T\varphi^T) \end{aligned} \tag{23}$$

In order to facilitate writing, the following provisions are made:

$$\begin{cases} Tr(1) \triangleq Tr(ZZ^T ZZ^T\varphi^T\varphi) \\ Tr(2) \triangleq Tr(ZZ^T\varphi^T\varphi) \\ Tr(3) \triangleq Tr(Z^T\varphi^T) \end{cases} \tag{24}$$

Then, formula (23) can be rewritten as:

$$H(\varphi, Z) = -Tr(1) - 2\alpha Tr(2) + 2Tr(3) - \alpha^2 Tr(\varphi\varphi^T) \tag{25}$$

In the following, $\partial Tr(1)/\partial w_k$, $\partial Tr(2)/\partial w_k$ and $\partial Tr(3)/\partial w_k$ will be calculated, respectively.

First of all, the analytic formulas of the elements of the matrices Z, ZZ^T and ZZ^TZZ^T need to calculate. The elements of matrix Z can be written directly:

$$\begin{cases} Z(k,i) = \max_{j=1}^{n}\{\sigma(w_k^T x_{ij})\} \triangleq \sigma(w_k^T x_{i\delta_i^k}) \\ \delta_i^k = \arg\max_j\{\sigma(w_k^T x_{ij})\} \end{cases} \tag{26}$$

where, *Z(k,i)* represents the element in the k-th row and i-th column of the matrix Z, and other matrix elements are expressed in the same way in the following.

According to (26), the expressions of the row vector of the k-th row of the matrix Z and the column vector of the k'-th column of the matrix Z^T can be obtained:

$$\begin{cases} Z(k,:) = \left[\sigma(w_k^T x_{1\delta_1^k}) \quad \sigma(w_k^T x_{2\delta_2^k}) \quad \cdots \quad \sigma(w_k^T x_{n\delta_n^k})\right] \\ Z^T(:,k') = \left[\sigma(w_{k'}^T x_{1\delta_1^{k'}}) \quad \sigma(w_{k'}^T x_{2\delta_2^{k'}}) \quad \cdots \quad \sigma(w_{k'}^T x_{n\delta_n^{k'}})\right]^T \end{cases} \tag{27}$$

The analytic formula for the elements in the k-th row and k'-th column of matrix ZZ^T can be given:

$$ZZ^T(k,k') = Z(k,:)\cdot Z^T(:,k') = \sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_{k'}^T x_{i\delta_i^{k'}}) \tag{28}$$

the expressions of the row vector of the k-th row and the column vector of the k'-th column of the matrix ZZ^T can be obtained:

$$\begin{cases} ZZ^T(k,:) = \left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_1^T x_{i\delta_i^1}) \quad \cdots \quad \sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_m^T x_{i\delta_i^m})\right] \\ ZZ^T(:,k') = \left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_1^T x_{i\delta_i^1}) \quad \cdots \quad \sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_m^T x_{i\delta_i^m})\right]^T \end{cases} \tag{29}$$

Based on (29), the analytic formula for the elements in the k-th row and k'-th column of matrix ZZ^TZZ^T can be given:

$$ZZ^TZZ^T(k,k') = \sum_{j=1}^{m}\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_j^T x_{i\delta_i^j})\right]\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_j^T x_{i\delta_i^j})\right] \tag{30}$$

So, the analytical expressions of *Tr(1)*, *Tr(2)*, and *Tr(3)* can be written as following:

$$\begin{aligned} Tr(1) &= Tr(ZZ^TZZ^T\varphi^T\varphi) \\ &= Tr\left[(ZZ^TZZ^T)(\varphi^T\varphi)\right] \\ &= \sum_{k=1}^{m}\sum_{k'=1}^{m} ZZ^TZZ^T(k,k')\cdot\varphi^T\varphi(k',k) \\ &= \sum_{k=1}^{m}\sum_{k'=1}^{m}\left\{\sum_{j=1}^{m}\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_j^T x_{i\delta_i^j})\right]\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_j^T x_{i\delta_i^j})\right]\right\} \\ &\quad \cdot\varphi^T\varphi(k',k) \end{aligned} \tag{31}$$

$$\begin{aligned} Tr(2) &= Tr(ZZ^T\varphi^T\varphi) \\ &= Tr\left[(ZZ^T)(\varphi^T\varphi)\right] \\ &= \sum_{k=1}^{m}\sum_{k'=1}^{m} ZZ^T(k,k')\cdot\varphi^T\varphi(k',k) \\ &= \sum_{k=1}^{m}\sum_{k'=1}^{m}\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_{k'}^T x_{i\delta_i^{k'}})\right]\cdot\varphi^T\varphi(k',k) \end{aligned} \tag{32}$$

$$\begin{aligned} Tr(3) &= Tr(Z^T \varphi^T) \\ &= Tr(Z\varphi) \\ &= \sum_{k=1}^{m}\sum_{i=1}^{n} Z(k,i)\cdot\varphi(i,k) \\ &= \sum_{k=1}^{m}\sum_{i=1}^{n} \sigma(w_k^T x_{i\delta_i^k})\cdot\varphi(i,k) \end{aligned} \tag{33}$$

In order to calculate the derivative of (31)–(33) with regard to w_k, write the terms containing w_k and the terms not containing w_k separately:

$$\begin{aligned} Tr(1) = &\sum_{k'\neq k}\sum_{k''\neq k}\left\{\sum_{j\neq k}\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_j^T x_{i\delta_i^j})\right]\left[\sum_{i=1}^{n}\sigma(w_{k''}^T x_{i\delta_i^{k''}})\sigma(w_j^T x_{i\delta_i^j})\right]\right\} \\ &\cdot\varphi^T\varphi(k'',k') \\ &+\sum_{k'\neq k}\sum_{k''\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_k^T x_{i\delta_i^k})\right]\left[\sum_{i=1}^{n}\sigma(w_{k''}^T x_{i\delta_i^{k''}})\sigma(w_k^T x_{i\delta_i^k})\right]\right\}\cdot\varphi^T\varphi(k'',k') \\ &+\sum_{k'\neq k}\sum_{j\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_j^T x_{i\delta_i^j})\right]\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_j^T x_{i\delta_i^j})\right]\right\}\cdot\varphi^T\varphi(k,k') \\ &+\sum_{k''\neq k}\sum_{j\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_j^T x_{i\delta_i^j})\right]\left[\sum_{i=1}^{n}\sigma(w_{k''}^T x_{i\delta_i^{k''}})\sigma(w_j^T x_{i\delta_i^j})\right]\right\}\cdot\varphi^T\varphi(k'',k) \\ &+\sum_{k'\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_k^T x_{i\delta_i^k})\right]\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_k^T x_{i\delta_i^k})\right]\right\}\cdot\varphi^T\varphi(k,k') \\ &+\sum_{k''\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_k^T x_{i\delta_i^k})\right]\left[\sum_{i=1}^{n}\sigma(w_{k''}^T x_{i\delta_i^{k''}})\sigma(w_k^T x_{i\delta_i^k})\right]\right\}\cdot\varphi^T\varphi(k'',k) \\ &+\sum_{j\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_j^T x_{i\delta_i^j})\right]\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_j^T x_{i\delta_i^j})\right]\right\}\cdot\varphi^T\varphi(k,k) \\ &+\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_k^T x_{i\delta_i^k})\right]^2\cdot\varphi^T\varphi(k,k) \end{aligned} \tag{34}$$

$$\begin{aligned} Tr(2) = &\sum_{k'\neq k}\sum_{k''\neq k}\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_{k''}^T x_{i\delta_i^{k''}})\right]\cdot\varphi^T\varphi(k'',k') \\ &+\sum_{k'\neq k}\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_k^T x_{i\delta_i^k})\right]\cdot\varphi^T\varphi(k,k') \\ &+\sum_{k''\neq k}\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_{k''}^T x_{i\delta_i^{k''}})\right]\cdot\varphi^T\varphi(k'',k) \\ &+\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})^2\right]\cdot\varphi^T\varphi(k,k) \end{aligned} \tag{35}$$

$$Tr(3) = \sum_{k'\neq k}\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\cdot\varphi(i,k') + \sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\cdot\varphi(i,k) \tag{36}$$

According to the above conclusion, the partial derivative of the dual function $H(\varphi, Z)$ with regard to w_k can be obtained. For the convenience of writing, the analytical formulas of $\partial Tr(1)/\partial w_k$, $\partial Tr(2)/\partial w_k$, and $\partial Tr(3)/\partial w_k$ will be derived, respectively:

$$\begin{aligned} \frac{\partial Tr(1)}{\partial w_k} = &\sum_{k'\neq k}\sum_{k''\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\right]\left[\sum_{i=1}^{n}\sigma(w_{k''}^T x_{i\delta_i^{k''}})\sigma(w_k^T x_{i\delta_i^k})\right]\right\} \\ &\cdot\varphi^T\varphi(k'',k') \\ &+\sum_{k'\neq k}\sum_{k''\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_k^T x_{i\delta_i^k})\right]\left[\sum_{i=1}^{n}\sigma(w_{k''}^T x_{i\delta_i^{k''}})\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\right]\right\} \\ &\cdot\varphi^T\varphi(k'',k') \\ &+\sum_{k'\neq k}\sum_{j\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_j^T x_{i\delta_i^j})\right]\left[\sum_{i=1}^{n}\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\sigma(w_j^T x_{i\delta_i^j})\right]\right\} \\ &\cdot\varphi^T\varphi(k,k') \\ &+\sum_{k''\neq k}\sum_{j\neq k}\left\{\left[\sum_{i=1}^{n}\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\sigma(w_j^T x_{i\delta_i^j})\right]\left[\sum_{i=1}^{n}\sigma(w_{k''}^T x_{i\delta_i^{k''}})\sigma(w_j^T x_{i\delta_i^j})\right]\right\} \\ &\cdot\varphi^T\varphi(k'',k) \\ &+\sum_{k'\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\right]\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_k^T x_{i\delta_i^k})\right]\right\}\cdot\varphi^T\varphi(k,k') \\ &+\sum_{k'\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\sigma(w_k^T x_{i\delta_i^k})\right]\left[2\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\right]\right\}\cdot\varphi^T\varphi(k,k') \\ &+\sum_{k''\neq k}\left\{\left[2\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\right]\left[\sum_{i=1}^{n}\sigma(w_{k''}^T x_{i\delta_i^{k''}})\sigma(w_k^T x_{i\delta_i^k})\right]\right\}\cdot\varphi^T\varphi(k'',k) \\ &+\sum_{k''\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_k^T x_{i\delta_i^k})\right]\left[\sum_{i=1}^{n}\sigma(w_{k''}^T x_{i\delta_i^{k''}})\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\right]\right\}\cdot\varphi^T\varphi(k'',k) \\ &+\sum_{j\neq k}\left\{\left[\sum_{i=1}^{n}\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\sigma(w_j^T x_{i\delta_i^j})\right]\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_j^T x_{i\delta_i^j})\right]\right\}\cdot\varphi^T\varphi(k,k) \\ &+\sum_{j\neq k}\left\{\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_j^T x_{i\delta_i^j})\right]\left[\sum_{i=1}^{n}\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\sigma(w_j^T x_{i\delta_i^j})\right]\right\}\cdot\varphi^T\varphi(k,k) \\ &+4\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\sigma(w_k^T x_{i\delta_i^k})\right]\cdot\left[\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\right]\cdot\varphi^T\varphi(k,k) \end{aligned} \tag{37}$$

$$\begin{aligned} \frac{\partial Tr(2)}{\partial w_k} = &\sum_{k'\neq k}\left[\sum_{i=1}^{n}\sigma(w_{k'}^T x_{i\delta_i^{k'}})\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\right]\cdot\varphi^T\varphi(k,k') \\ &+\sum_{k''\neq k}\left[\sum_{i=1}^{n}\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\sigma(w_{k''}^T x_{i\delta_i^{k''}})\right]\cdot\varphi^T\varphi(k'',k) \\ &+\left[2\sum_{i=1}^{n}\sigma(w_k^T x_{i\delta_i^k})\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\right]\cdot\varphi^T\varphi(k,k) \end{aligned} \tag{38}$$

$$\frac{\partial Tr(3)}{\partial w_k} = \sum_{i=1}^{n}\frac{\partial\sigma(w_k^T x_{i\delta_i^k})}{\partial w_k}\cdot\varphi(i,k) \tag{39}$$

The partial derivative of the dual function $H(\varphi, Z)$ with regard to w_k can be written as:

$$\frac{\partial H(\varphi,Z)}{\partial w_k} = -\frac{\partial Tr(1)}{\partial w_k} - 2\alpha\frac{\partial Tr(2)}{\partial w_k} + 2\frac{\partial Tr(3)}{\partial w_k} \tag{40}$$

Substitute (37) $\sim$ (39) into (40), the analytical expression of $\partial H(\varphi,Z)/\partial w_k$ can be obtained. Substitute $\partial H(\varphi,Z)/\partial w_k$ into (22), the update formula of the convolution kernel can be obtained.

3.3 Algorithm steps for updating

In Sect. 2.3.1 and 2.3.2, the optimization goal of networks parameter and calculation process of parameter updating is studied. Summarize the training process of IA-optimal CNN as follows: Initialize the networks structure and networks parameters. Input the training dataset and output the classification result. If the number of iterations p is less than the upper limit P, the convolution kernel matrix and weight will be, respectively, updated according to the parameter update strategy proposed in Sect. 2.3.2; otherwise, the algorithm ends and the final convolution kernel matrix and weight are output.

Based on this, the training process of IA-optimal CNN can be written as follows:

- Algorithm: Training of IA-optimal CNN

-START-

- Input: Signals samples dataset- $\{S\}$
  Label vector- y
  L2 regularization coefficient- α
  Learning rate- β
  Maximum number of iterations- P
- Step1: Initialization
  Convolution kernel matrix- W
  Weight - ω
  Iteration counter- $p=1$
- Step2: Input data and obtain the classification result;
- Step3: if $p<P$
  {update ω base on formula (9);
  update φ base on formula (18);
  update W base on formula (22);
  to Step 2 }
  else
  {to Step 4 }
- Step4: Output: W and ω

-END-

The output convolution kernel matrix W and weight ω can be used as the final parameters of IA-optimal CNN, and the trained networks can be used to classify the signals from the testing dataset.

4 Experiments

In this chapter, experiments are carried out with different types of signals data to test the classification ability of the IA-optimal CNN. The classification results are showed, and the algorithm performance are compared with the existing A-optimal-based classification algorithm.

4.1 Experimental data

In order to verify the ability of IA-optimal CNN on classifying different types of signals data, this article uses five datasets: P300 EEG signals dataset, bearing fault signals dataset, English spoken digit signals dataset, arrhythmia signals dataset, gearbox signals dataset. The above five datasets cover a wide range of engineering application fields, and the signals generation mechanisms are diverse, which can be used to test the classification universality of the different method for different signals datasets.

P300 EEG signals is an evoked EEG signals, which is generated when the human brain is stimulated by a small probability event. The EEG data are collected by the character matrix flashing experiment. During the experiment, a character matrix is displayed on the computer screen, and a "target character" is displayed at the same time to let the subjects pay attention to this character. Then, each row or column of the character matrix flashes in a random order. When the row or column contains the target character is flashed, the P300 potential will appear in the EEG signals. The experimental dataset contains 20 channels of collected data from the subjects. The IA-optimal CNN method will recognize the input EEG signals sample and determine whether it contains P300 [22]. The bearing failure signals of mechanical equipment are collected by the sensors installed on the bearing failure test stand, and the sampling frequency is 12 kHz. In order to distinguish the fault signals from the normal signals, the dataset also contains the normal bearing signals waveform at the same sampling frequency. The selected fault type is that outer ring fault magnitude is 0.021 mil, and the fault location is 6 o'clock. The IA-optimal CNN method will be used to recognize the input signals samples and determine whether the signals are a fault signals and then determine whether the bearing has this type of fault [5]. The English spoken digit dataset consists of the audio of 6 speakers, each of whom repeats different digits 50 times, and the speakers all use English pronunciation when speaking. In this experiment, digital 1 audio data are used, and all audio will be converted into acoustic signals, and then input into the networks for signals classification. The IA-optimal CNN method will recognize the input acoustic signals samples and determine whether the signals is the audio signals of the target digit [52]. The arrhythmia signals dataset contains 48 dual-channel Holter signals records, and each record contains the ECG waveform signals with a duration of more than 30 min and expert diagnosis results. The records came from 47 subjects in the arrhythmia laboratory. The IA-optimal CNN method will use the dataset to distinguish between normal beat signals and arrhythmia signals [53]. The gearbox signals dataset collects signals data under different working conditions based on drivetrain dynamic simulator. The gearbox is driven by a motor, and the experiment is carried out on a simulated failure test bench. The rotating speed-system load is set to be 20 Hz–0 V. The IA-optimal CNN method will distinguish the signals under normal operating conditions and the fault signals under the damage conditions that the gear miss one of feet [54].

The computer brand used for calculation is DELL, and the processor model is Intel(R)Core(TM)i5-8250U CPU@ 1.60 GHz 1.80 GHz. The experiment uses cross-validation. Each rotation takes 80% of the sample set as training samples dataset and the remaining 20% as testing samples dataset. The final metrics is the average of all test results. L2 regularization coefficient $\alpha = 0.02$, learning rate $\beta = 10^{-4}$, maximum number of iterations $P = 1000$.

4.2 Performance comparison

The performance comparison section is divided into two parts: In the first part, the performance difference of the IA-optimal CNN method using different activation functions will be compared, and the method with the best performance will be selected as the object for subsequent comparison with other algorithms; In the second part, the IA-optimal CNN will be compared with the existing A-optimal-based classification algorithm to show the performance difference.

4.2.1 Performance comparison of IA-optimal CNN using different activation functions

Based on the same dataset, the accuracy of the IA-optimal CNN using six different activation functions will be compared with each other. The six activation functions are Sigmoid function, Tanh function, ReLu function, Leaky-ReLu function, ELU function, and Cube function used by Yin [51]. The activation functions $\sigma(\cdot)$ and the corresponding partial derivative $\partial\sigma(w_k^T x_{i\delta_i^k})/\partial w_k$ can be calculated by the following formula [55, 56]:

$$\begin{cases} \sigma_1(x) = \mathrm{Sigmoid}(x) = \dfrac{1}{1+\exp(-x)} \\ \dfrac{\partial\sigma_1(w_k^T x_{i\delta_i^k})}{\partial w_k} = \left(1-\sigma_1(w_k^T x_{i\delta_i^k})\right)\cdot\sigma_1(w_k^T x_{i\delta_i^k})\cdot x_{i\delta_i^k} \end{cases} \tag{41}$$

$$\begin{cases} \sigma_2(x) = \tanh(x) = \dfrac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)} \\ \dfrac{\partial\sigma_2(w_k^T x_{i\delta_i^k})}{\partial w_k} = \left(1-\sigma_2^2(w_k^T x_{i\delta_i^k})\right)\cdot x_{i\delta_i^k} \end{cases} \tag{42}$$

$$\begin{cases} \sigma_3(x) = ReLu(x) = \begin{cases} 0(x<0) \\ x(x\geq 0) \end{cases} \\ \dfrac{\partial\sigma_3(w_k^T x_{i\delta_i^k})}{\partial w_k} = \begin{cases} 0(w_k^T x_{i\delta_i^k}<0) \\ x_{i\delta_i^k}(w_k^T x_{i\delta_i^k}\geq 0) \end{cases} \end{cases} \tag{43}$$

$$\begin{cases} \sigma_4(x) = Leaky\,ReLu(x) = \begin{cases} \gamma x(x<0) \\ x(x\geq 0) \end{cases} \\ \dfrac{\partial\sigma_4(w_k^T x_{i\delta_i^k})}{\partial w_k} = \begin{cases} \gamma x_{i\delta_i^k}(w_k^T x_{i\delta_i^k}<0) \\ x_{i\delta_i^k}(w_k^T x_{i\delta_i^k}\geq 0) \end{cases} \end{cases} \tag{44}$$

$$\begin{cases} \sigma_5(x) = ELU(x) = \begin{cases} \varepsilon\cdot(e^{-x}-1)(x<0) \\ x(x\geq 0) \end{cases} \\ \dfrac{\partial\sigma_5(w_k^T x_{i\delta_i^k})}{\partial w_k} = \begin{cases} -\varepsilon\cdot\exp(-w_k^T x_{i\delta_i^k})\cdot x_{i\delta_i^k}(w_k^T x_{i\delta_i^k}<0) \\ x_{i\delta_i^k}(w_k^T x_{i\delta_i^k}\geq 0) \end{cases} \end{cases} \tag{45}$$

$$\begin{cases} \sigma_6(x) = \mathrm{Cube}(x) = x^3 \\ \dfrac{\partial\sigma_6(w_k^T x_{i\delta_i^k})}{\partial w_k} = 3(w_k^T x_{i\delta_i^k})^2\cdot x_{i\delta_i^k} \end{cases} \tag{46}$$

The accuracy comparison results are shown in the following table:

According to the results in the table, the following conclusions can be obtained: (1) When using IA-optimal CNN for signals recognition, arrange the recognition accuracy in descending order, the rank is as follows: arrhythmia signals, English spoken digit signals, bearing failure signals, gearbox signals, P300 signals; (2)When classifying P300 signals, arrhythmia signals and bearing failure signals, the networks using sigmoid activation function have the highest accuracy. When classifying English spoken digit signals and gearbox signals, the networks using tanh activation function have the highest accuracy; (3) When using ReLu or Leaky-ReLu activation functions for classification and recognition, the accuracy rate is generally low. The possible reason is that the ability to extract nonlinear features of the signals is insufficient when ReLu or Leaky-ReLu is used as activation functions based on the IA-optimal CNN structure.

4.2.2 Performance comparison of different A-optimal-based classification algorithms

In this section, the algorithm performance of IA-optimal CNN will be compared with five existing A-optimal-based classification method. Algorithms used for comparison are A-optimal subspace learning method proposed by He [36], the nonnegative matrix factorization method proposed by Liu [37], the method based on neighborhood regularization proposed by Li [38], the method based on Hessian energy Regularization proposed by Yang [50], and the A-optimal CNN method proposed by Yin [51]. The five metrics to evaluate networks performance are accuracy, precision, recall, G-score, and F1-score. The above metrics can be calculated using following equations:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (47)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (48)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (49)$$

$$G - \text{score} = \sqrt{\text{Precision} \cdot \text{Recall}} \quad (50)$$

$$F1 - \text{score} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (51)$$

where, '*TP*'' means '*Ture-Positive*', '*TN*'' means '*Ture-Negative*', '*FP*'' means '*False-Positive*', '*FN*'' means '*False-Negative*'. The experimental results are shown in the following table:

In order to more intuitively show the performance difference between the algorithms, the above data are drawn into the following picture:

According to the results in Tables 2–6 and Figs. 3–7, the following conclusions can be obtained: (1) The five metrics of IA-optimal CNN are superior to other A-optimal classification algorithms. The five metrics measure the performance of different classification methods from different aspects to ensure that the evaluation metric system is as comprehensive as possible. It can be seen that IA-optimal CNN has advantages in all metrics and shows stronger recognition capabilities; (2) IA-optimal CNN has the best classification performance on all signals datasets, which proves that IA-optimal CNN algorithm has good versatility for a variety of signals classification problems. The five different signals datasets used in this paper involve different application fields, and the generation mechanism of signals are also different. Among the existing classification algorithms based on A-optimal, using five types of datasets, IA-optimal CNN ranks first in all the classification metrics, which can show the universality of IA-optimal CNN; (3) From Table 2–6, it can be found that if the datasets are arranged according to the classification accuracy of IA-optimal CNN from largest to smallest, the order is: arrhythmia signals dataset, English spoken digit signals dataset, bearing failure signals dataset, gearbox signals dataset, P300 EEG signals dataset. The accuracy rankings of other A-optimal-based classification algorithms are roughly the same. This sort can show the difficulty degree of distinguishing different types of signals in different datasets. The quality of the data in the arrhythmia signals dataset and the English spoken digit signals dataset are relatively better. For most classification algorithms, different types of signals in these two datasets are easier to distinguish. The signals in the bearing failure signals dataset and gearbox signals dataset are more complex and are easily interfered by noise in the engineering environment, so the classification accuracy is relatively low. Compared with the signals in other datasets, the signals characteristics of P300 EEG are the least obvious, and the classification accuracy is generally the lowest; (4) The performances of the two A-optimal-based CNN methods are better than the other four methods. Both IA-optimal CNN and A-optimal CNN introduce the A-optimal criterion into the CNN framework. The convolutional layer and the pooling layer further enhance the nonlinear expression ability of the network, which can better extract the nonlinear features of the signals; 5) The performance of the IA-optimal CNN proposed in this paper is better than that of A-optimal CNN. Compared with A-optimal CNN, the IA-optimal CNN uses an optimized objective function that has not been simplified to improve the accuracy. In addition, A-optimal CNN uses the cube function as the activation function, while IA-optimal CNN uses the Sigmoid function and Tanh function as the activation functions, so that the network can approximate higher-order nonlinear features; 6) In other A-optimal classification algorithms, the performance of algorithms that introduced regularization improvement methods has generally been improved to a certain extent.

The accuracy box plots of different algorithms using different datasets are as follows:

According to the results in the above figures, the following conclusions can be obtained: The accuracy of the IA-optimal CNN recognition method is generally high. In addition, the accuracy distribution of IA-optimal CNN is relatively more stable and the degree of dispersion is low.

5 Discussion: can the A-optimal-based CNN finally converge strictly to the A-optimal state?

In the above, the IA-optimal CNN framework is given, the update strategy of the networks is studied in detail, and the performance of algorithms is compared with different methods based on different types of datasets.

In the IA-optimal CNN method, the alternate iterative optimization method is used to update the convolution kernel matrix W. The convolution kernel matrix is continuously updated along the gradient direction, so that $Tr(\text{cov}(\omega))$ continues to decrease until it converges. In this case, the new problem will arise: Can this method make the trained networks reach a strict A-optimal state? Can $Tr(\text{cov}(\omega))$ converge to its theoretical minimum in the end?

According to Sect. 2.2.1, when $\partial(\text{loss}(S))/\partial\omega = 0$, the expression of ω_0 is $\omega_0 = (ZZ^T + \alpha I)^{-1} Zy^T$. In this formula, only Z is a variable. Set $B = (ZZ^T + \alpha I)^{-1}$, then formula (12) can be written as:

$$Tr(\text{cov}(\omega)) = \sigma^2\left(Tr(B) - \alpha Tr(B^2)\right) \quad (52)$$

Table 1 Accuracy comparison—six activation functions

$\sigma(x)$	P300 EEG signals	Bearing failure signals	English spoken digit signals	Arrhythmia signals	Gearbox signals
Sigmoid	0.778	0.851	0.869	0.899	0.836
Tanh	0.765	0.823	0.889	0.879	0.837
ReLu	0.661	0.725	0.811	0.814	0.717
Leaky-ReLu	0.643	0.722	0.784	0.813	0.702
ELU	0.738	0.787	0.830	0.851	0.759
Cube [51]	0.741	0.827	0.844	0.856	0.788

Table 2 Algorithm performance comparison/P300 EEG signals

Method	Accuracy	Precision	Recall	G- score	F1-score
A-optimal subspace learning [49:He]	0.661	0.515	0.634	0.571	0.568
nonnegative matrix factorization [50:Liu]	0.685	0.535	0.628	0.580	0.578
neighborhood regularization [51:Li]	0.703	0.549	0.649	0.597	0.595
Hessian energy Regularization [52:Yang]	0.673	0.524	0.618	0.569	0.567
A-optimal CNN [53:Yin]	0.741	0.563	0.692	0.624	0.621
IA-optimal CNN	0.778	0.587	0.720	0.650	0.647

Table 3 Algorithm performance comparison / bearing failure signals

Method	Accuracy	Precision	Recall	G-score	F1-score
A-optimal subspace learning [49:He]	0.701	0.489	0.665	0.570	0.564
nonnegative matrix factorization [50:Liu]	0.796	0.627	0.748	0.685	0.682
neighborhood regularization [51:Li]	0.753	0.525	0.750	0.628	0.618
Hessian energy Regularization [52:Yang]	0.724	0.506	0.769	0.624	0.610
A-optimal CNN [53:Yin]	0.827	0.703	0.846	0.771	0.768
IA-optimal CNN	0.851	0.789	0.871	0.829	0.828

In order to obtain the matrix B when $Tr(\mathrm{cov}(\omega))$ takes the minimum value, the derivative of $Tr(\mathrm{cov}(\omega))$ with regard to B is set to 0:

$$\frac{\partial Tr(\mathrm{cov}(\omega))}{\partial B} = \sigma^2(I - 2\alpha B^T) = 0 \tag{53}$$

The analytical expression of $B_0 \in R^{m\times m}$ is:

$$B_0 = \frac{1}{2\alpha} I \tag{54}$$

The matrix $Z_0 Z_0^T \in R^{m\times m}$ when $Tr(\mathrm{cov}(\omega))$ takes the minimum value can be calculated:

$$Z_0 Z_0^T = \begin{bmatrix} \alpha & 0 & \cdots & 0 \\ 0 & \alpha & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \alpha \end{bmatrix} \tag{55}$$

Table 4 Algorithm performance comparison/English spoken digit signals

Method	Accuracy	Precision	Recall	G-score	F1-score
A-optimal subspace learning [49:He]	0.711	0.571	0.703	0.634	0.630
nonnegative matrix factorization [50:Liu]	0.778	0.594	0.725	0.656	0.653
neighborhood regularization [51:Li]	0.803	0.652	0.784	0.715	0.712
Hessian energy Regularization [52:Yang]	0.731	0.568	0.691	0.627	0.624
A-optimal CNN [53:Yin]	0.844	0.747	0.850	0.797	0.795
IA-optimal CNN	0.889	0.833	0.900	0.866	0.865

Table 5 Algorithm performance comparison/arrhythmia signals

Method	Accuracy	Precision	Recall	G-score	F1-score
A-optimal subspace learning [49:He]	0.704	0.685	0.698	0.692	0.691
nonnegative matrix factorization [50:Liu]	0.786	0.726	0.754	0.740	0.740
neighborhood regularization [51:Li]	0.793	0.743	0.806	0.774	0.773
Hessian energy Regularization [52:Yang]	0.774	0.700	0.780	0.739	0.738
A-optimal CNN [53:Yin]	0.855	0.813	0.871	0.842	0.841
IA-optimal CNN	0.899	0.880	0.884	0.882	0.882

Table 6 Algorithm performance comparison/gearbox signals

Method	Accuracy	Precision	Recall	G-score	F1-score
A-optimal subspace learning [49:He]	0.695	0.524	0.681	0.597	0.592
nonnegative matrix factorization [50:Liu]	0.733	0.665	0.716	0.690	0.690
neighborhood regularization [51:Li]	0.708	0.596	0.740	0.664	0.660
Hessian energy Regularization [52:Yang]	0.717	0.632	0.702	0.666	0.665
A-optimal CNN [53:Yin]	0.807	0.726	0.822	0.773	0.771
IA-optimal CNN	0.837	0.787	0.847	0.816	0.816

To satisfy this condition, the matrix Z must have the following form:

$$Z_0(k,i) = \begin{cases} \sqrt{\alpha}(k=i) \\ 0(k \neq i) \end{cases} \tag{56}$$

According to Sect. 2.3.2, the elements in Z_0 are obtained by maximum pooling, namely:

$$Z_0(k,i) = \max_{j=1}^{n}\{\sigma(w_k^T x_{ij})\} \tag{57}$$

Obviously, for any input sample X_i, it is almost impossible to find a convolution kernel matrix W that meets the formula (57).

Therefore, the answer to the question in this chapter is: For all the classification algorithms based on A-optimal including IA-optimal CNN and A-optimal CNN proposed by Yin [51], $Tr(\mathrm{cov}(\omega))$ can continuously reduce in the process of algorithm iterations, but the theoretical minimum cannot be achieved, and the networks cannot strictly achieve the best A-optimal state. Nevertheless, the classification algorithm based on A-optimal still makes the value of $Tr(cov(\omega))$ as small as possible when conditions permit, which can ensure the stability of the optimization model to a greater extent.

6 Conclusions

This paper proposes an novel IA-optimal CNN method and test it universality to solve different signals classification problems. The main contributions of our works include five aspects: (1) The existing A-optimal-based CNN simplifies the trace of the covariance matrix of the fully connected layer weights and uses the simplified calculation formula as parameter optimization objective function, which affect the classification accuracy to a certain extent. IA-optimal CNN did not perform any simplification on the trace of the covariance matrix and directly used the precise analytical expression of optimization objective function to complete the subsequent formula derivation. This approach makes the results more precise; (2) The A-optimal-based CNN framework uses gradient descent method for network training, so it is necessary to calculate the analytical expression of the partial derivative of the optimization objective function with regard to the network parameters. The precise expression of the trace of the covariance matrix contains the inverse matrix, so it is difficult to directly write the analytical solution of the partial derivative with regard to the network parameter. For this reason, a novel dual function is proposed and the following conclusions are proved mathematically: the minimum value of this dual function is exactly equal to the optimization objective function. According to this conclusion, the original

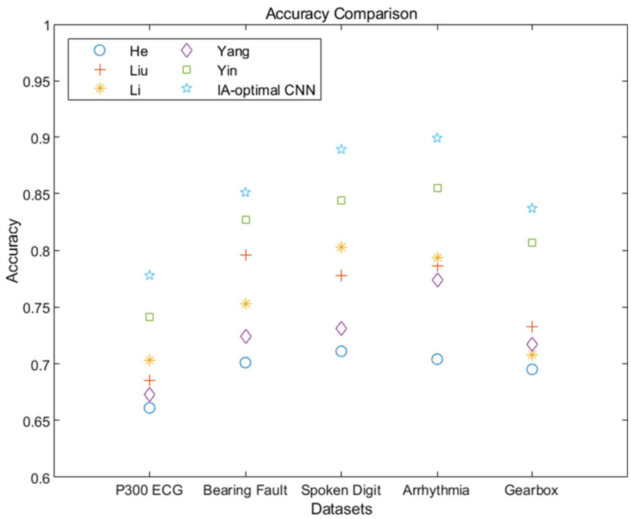

Fig. 3 Algorithm performance comparison/accuracy

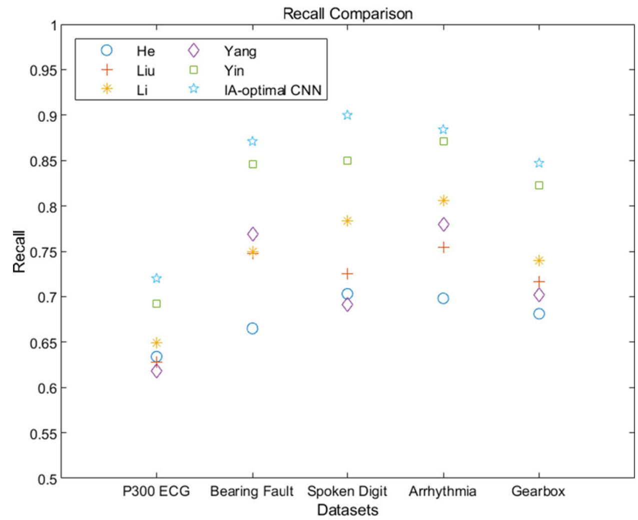

Fig. 5 Algorithm performance comparison/recall

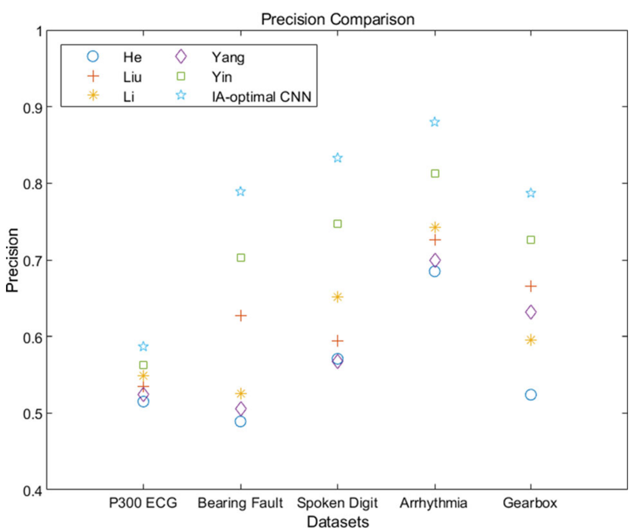

Fig. 4 Algorithm performance comparison/precision

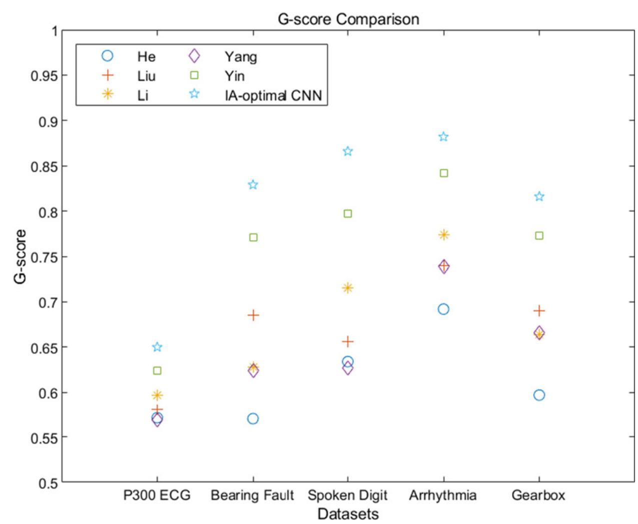

Fig. 6 Algorithm performance comparison/G-score

optimization problem is transformed into an equivalent binary matrix function optimization problem, avoiding the difficulty of not being able to obtain the analytical solution of the derivative of the inverse matrix with regard to network parameter; (3) To improve the nonlinear expression ability of the network, the Tanh function and Sigmoid function are used as the activation function instead of Cube function which is used in A-optimal CNN; (4) To further verify the versatility of the A-optimal-based CNN algorithm on signals classification problems, five different signals datasets are used as samples to test the algorithm performance; (5) The following conclusion is proved mathematically: In the iterative process of A-optimal-based CNN, the trace of the covariance matrix will continuously shrink and approach a convergence value, but the theoretical minimum cannot be achieved.

The results of algorithm performance test show that compared with the existing A-optimal classification algorithm, the IA-optimal CNN has the best performance in classifying five datasets. In addition, the accuracy distribution of IA-optimal CNN is relatively more stable and the degree of dispersion is low, which verifies the accuracy and stability of the IA-optimal CNN.

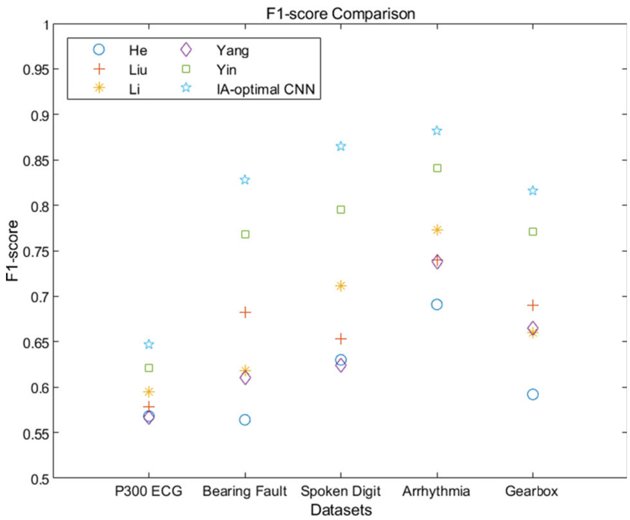

Fig. 7 Algorithm performance comparison/F1-score

7 Future works

At present, the research on the A-optimal-based CNN network architecture and mathematical principles is still in its infancy, and there are few related theories and application results. The A-optimal-based CNN has specific operating rules, so the further expansion of the network depth is restricted. According to the principle of A-optimal CNN, the dimensionality of the data will reduce once after each round of convolution and pooling. The signals are a one-dimensional vector. If the depth of the convolutional layer and the pooling layer increase, the structure of the input signals must to be changed continuously according to the network depth. We have not yet found a data input structure with clear mathematical meaning to ensure the feasibility of this approach. So, further research is necessary for the issues such as "how to expand the network structure to a deeper network," "how to adjust the data input format or network architecture to adapt to the deeper network," and "how the training time, algorithm complexity and classification accuracy will change as the network depth increases." Therefore, the depth expansion of the IA-optimal CNN network, the time-consuming of deep network iteration, the computational complexity, and the accuracy of algorithm will be the focus of the future research.

In addition, based on the specific application background, the structure of IA-optimal CNN can be improved to make it more targeted to solve different types of signals classification problems. For example, when using vibration signals for fault diagnosis, based on full research for the

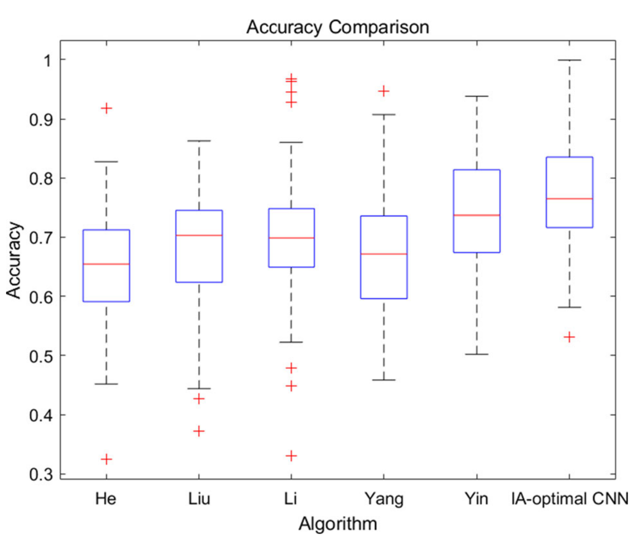

Fig. 8 Accuracy box plots/P300 EEG signals

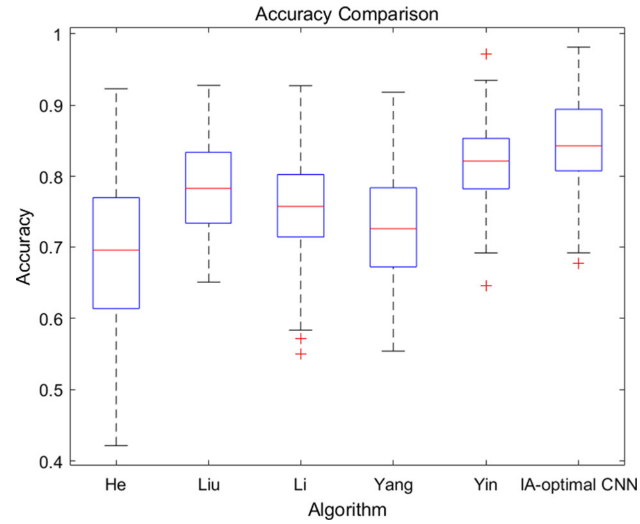

Fig. 9 Accuracy box plots/bearing failure signals

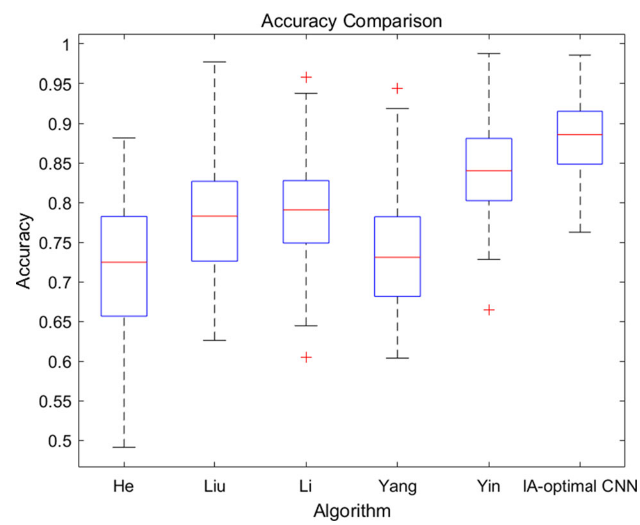

Fig. 10 Accuracy box plots/English spoken digit signals

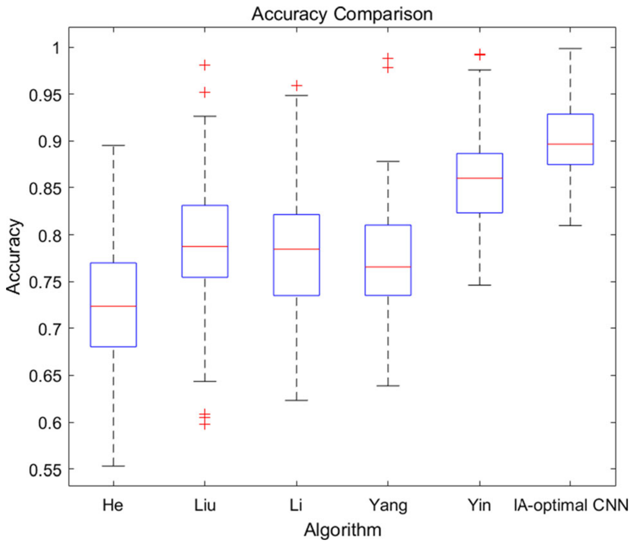

Fig. 11 Accuracy box plots/arrhythmia signals

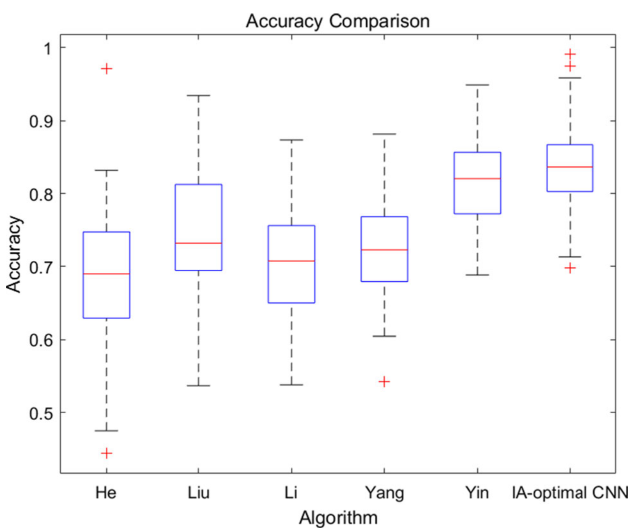

Fig. 12 Accuracy box plots/gearbox signals

structural mechanism and mechanical characteristics of the equipment, the IA-optimal-CNN-based fault signals diagnosis networks for different types of power equipment can be designed [57, 58]. When classifying signals based on multiple sensor channels, the signals data from different sensors can be used as the input of different channels of CNN to complete the signals classification based on data fusion[41, 59]. Taking into account the noise interference and data loss that may occur in the process of sensor signals transmission, the model for loss data recovery can also be considered when designing the algorithm. In this case, multiple structures such as SNN, RNN, GRNN, LSTM and SGTM can be combined with A-optimal-based CNN [60–66].

Acknowledgements This work was supported by The National Natural Science Foundation of China 11904407.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval All authors have known that the manuscript is submitted to your journal. And all of us confirm that the content of the manuscript has not been published elsewhere.

References

1. Jia F, Lei Y, Na Lu, Xing S (2018) Deep normalized convolutional neural networks for imbalanced fault classification of machinery and its understanding via visualization. Mech Syst Sign Process 110:349–367
2. Plakias S, Boutalis YS (2020) Fault detection and identification of rolling element bearings with Attentive Dense CNN. Neurocomput 405:208–217
3. Han T, Tian ZhiXin, Yin Z, Tan ACC (2020) Bearing fault identifcation based on convolutional neural networks by diferent input modes. J Brazilian Soc Mech Sci Eng 42:474–484
4. Yang Y, Zheng H, Li Y, Minqiang Xu, Chen Y (2019) A fault diagnosis scheme for rotating machinery using hierarchical symbolic analysis and convolutional neural networks. ISA Trans 91:235–252
5. Cheng C, Zhou B, Ma G, Dongrui Wu, Yuan Ye (2020) Wasserstein distance based deep adversarial transfer learning for intelligent fault diagnosis with unlabeled or insufficient labeled data. Neurocomput 409:35–45
6. Zifei Xu, Li C, Yang Y (2020) Fault diagnosis of rolling bearing of wind turbines based on the variational mode decomposition and deep convolutional neural networks. Appl Soft Comput J 95:106515
7. Huang W, Junsheng Cheng Yu, Yang GG (2019) An improved deep convolutional neural networks with multi-scale information for bearing fault diagnosis. Neurocomput 359:77–92
8. Hao S, Ge F-X, Li Y, Jiang J (2020) Multisensor bearing fault diagnosis based on one-dimensional convolutional long short-term memory networks. Measure 159:107802
9. Chen Z, Mauricio A, Li W, Gryllias K (2020) A deep learning method for bearing fault diagnosis based on cyclic spectral coherence and convolutional neural networks. Mech Syst Sign Process 140:106683
10. Ma Y, Jia X, Bai H, Liu G, Wang G, Guo C, Wang S (2019) A new fault diagnosis method based on convolutional neural networks and compressive sensing. J Mech Sci Technol 33:5177–5188
11. Zhong S-S, Song Fu, Lin L (2019) A novel gas turbine fault diagnosis method based on transfer learning with CNN. Measure 137:435–453
12. Wen L, Gao L, Li X, Wang L, Zhu J (2018) A jointed signals analysis and convolutional neural networks method for fault diagnosis. Procedia CIRP 72:1084–1087
13. Wang S, Xiang J, Zhong Y, Zhou Y (2017) Convolutional neural networks-based hidden Markov models for rolling element bearing fault identification. Knowl-Based Syst pp 1–12
14. Jing L, Zhao M, Li P, Xu X (2017) A convolutional neural networks based feature learning and fault diagnosis method for the condition monitoring of gearbox. Measure. https://doi.org/10.1016/j.measurement.2017.07.017

15. Chen Lu, Wang Z, Zhou Bo (2017) Intelligent fault diagnosis of rolling bearing using hierarchical convolutional networks based health state classification. Adv Eng Inform 32:139–151
16. Demir F, Turkoglu M, Aslan M, Sengur A (2020) A new pyramidal concatenated CNN approach for environmental sound classification. Appl Acoust 170:107520
17. Abdoli S, Cardinal P, Koerich AL (2019) End-to-end environmental sound classification using a 1D convolutional neural networks. Expert Syst Appl 136:252–263
18. Chen Y, Guo Q, Liang X, Wang J, Qian Y (2019) Environmental sound classification with dilated convolutions. Appl Acoust 148:123–132
19. Francesco B, Giacomo C, Grazia LS et al (2020) A novel training method to preserve generalization of RBPNN classifiers applied to ECG signals diagnosis. Neural Netw 108:331–338
20. Deng M, Meng T, Cao J, Wang S, Zhang J, Fan H (2020) Heart sound classification based on improved MFCC features and convolutional recurrent neural networks. Neural Netw 130:22–32
21. Demir F, Şengür A, Bajaj V, Polat K (2019) Towards the classification of heart sounds based on convolutional deep neural networks. Health Inf Sci Syst 7:16–24
22. Huang X, Huang H, Wu J, Yang M, Ding W (2020) Sound quality prediction and improving of vehicle interior noise based on deep convolutional neural networks. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113657
23. Blankertz B, Muller KR, Krusienski DJ (2006) The BCI competition III: Validating alternative approaches to actual BCI problems. IEEE Trans Neural Syst Rehabil Eng 14:153–159
24. Bayliss JD (2003) Use of the evoked potential P3 component for control in a virtual apartment. IEEE Trans Neural Syst Rehabil Eng 11:113–116
25. Farwell LA, Donchin E (1988) Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalogr Clin Neurophysiol 70:510–523
26. Pfurtscheller G, Neuper C, Guger C et al (2000) Current TRENds in Graz brain-computer interface (BCI) research. Rehabil Eng 8:216–219
27. Andrew C, Pfurtscheller G (1996) Event-related coherence as a tool for studying dynamic interaction of brain regions. Electroencephalogr Clin Neurophysiol 98:144–148
28. Mingai Li, Meng Z, YanJun S (2016) A novel motor imagery EEG recognition method based on deep learning. Int Forum Manage Edu Inf Tech Appl 1:728–733
29. Kong NCL, Kaneshiro B, Yamins DLK, Norcia AM (2020) Time-resolved correspondences between deep neural networks layers and EEG measurements in object processing. Vision Res 172:27–45
30. Jiao Z, Gao X, Wang Y, Li J, Haojun Xu (2017) Deep convolutional neural networks for mental load classification based on EEG data. Pattern Recogn. https://doi.org/10.1016/j.patcog.2017.12.002
31. Vareka L (2020) Evaluation of convolutional neural networks using a large multi-subject P300 dataset. Biomed Sig Process Control 58:101837
32. Tang Z, Li C, Sun S (2017) Single-trial EEG classification of motor imagery using deep convolutional neural networks. Optik 130:11–18
33. Sors A, Bonnet S, Mirek S, Vercueil L, Jean-Franc, ois Payen, (2018) A convolutional neural networks for sleep stage scoring from raw single-channel EEG. Biomed Signals Process Control 42:107–114
34. Acharya UR, Oh SL, Hagiwara Y, Tan JH, Adeli H (2017) Deep convolutional neural networks for the automated detection and diagnosis of seizure using EEG signals. Comput Biol Med. https://doi.org/10.1016/j.compbiomed.2017.09.017
35. Ding Ao, Zhang Y, Zhu L, Yanping Du, Ma L (2020) Recognition method research on rough handling of express parcels based on acceleration features and CNN. Measurement 163:107942
36. Hong L, Liu X, Zuo H (2019) Compound faults diagnosis based on customized balanced multiwavelets and adaptive maximum correlated kurtosis deconvolution. Measure 146:87–100
37. Yonggang Xu, Zhang K, Ma C, Cui L, Tian W (2019) Adaptive Kurtogram and its applications in rolling bearing fault diagnosis. Mech Syst Signals Process 130:87–107
38. Li J, Yao X, Wang H, Zhang J (2019) Periodic impulses extraction based on improved adaptive VMD and sparse code shrinkage denoising and its application in rotating machinery fault diagnosis. Mech Syst Sig Process 126:568–589
39. Chen F, Muhammad K, Wang S-H (2020) Three-dimensional reconstruction of CT image features based on multi-threaded deep learning calculation. Pattern Recogn Lett 136:309–315
40. Wei W, Ke Q, Nowak J et al (2020) Accurate and fast URL phishing detector: a convolutional neural network approach. Comput Netw. https://doi.org/10.1016/j.comnet.2020.107275
41. Jiang X, Bo Hu, Satapathy SC et al (2020) Fingerspelling identification for Chinese sign language via alexnet-based transfer learning and adam optimizer. Sci Progr. https://doi.org/10.1155/2020/3291426
42. Wang S-H, Govindaraj VV, Gorriz JMJM et al (2020) Covid-19 Classification by fgcnet with deep feature fusion from graph convolutional network and convolutional neural network. Inf Fus. https://doi.org/10.1016/j.inffus.2020.10.004
43. Shirakura T, Tong WP (1996) Weighted a-optimal for fractional 2 m factorial designs of resolution V. J Stat Plan Inference 56:243–256
44. Das A, Gupta S, Kageyama S (2006) A-optimal diallel crosses for test versus control comparisons. J Appl Stat 33(6):601–608
45. Limmun W, Borkowski JJ, Chomtee B (2018) Weighted a-optimal criterion for generating robust mixture designs. Comput Ind Eng 125:348–356
46. Hajiyev Ch (2010) Determination of optimum measurement points via A-optimal criterion for the calibration of measurement apparatus. Measure 43:563–569
47. López-Fidalgo J, Rivas-López MJ, Fernández-Garzón B (2007) A-optimal standardized through the coefficient of variation. Commun Stat Theor Methods 36(4):781–792
48. He X, Zhang C, Zhang L, Li X (2016) A-optimal projection for image representation. IEEE Trans Pattern Anal Mach Intell 38:1009–1015
49. Liu X, Wang J, Yin M, Edwards B, Xu P (2015) Supervised learning of sparse context reconstruction coefficients for data representation and classification. Neural Comput Appl 1:1–9
50. Li P, Bu J, Chen C, Wang C, Cai D (2013) Subspace learning via locally constrained a-optimal nonnegative projection. Neurocomput 115:49–62
51. Yang Z, Liu H, Cai D, Wu Z (2016) A-optimal non-negative projection with hessian regularization. Neurocomput 174:838–849
52. Yin Z, Kong D, Shao G, Ning X, Jin W, Wang J-Y (2018) A-optimal convolutional neural networks. Neural Comput Appl 30:2295–2304
53. Jackson Z, Souza C, Flaks J (2018). Jakobovski/free-spoken-digit-dataset: v1.0.8 (Version v1.0.8). Zenodo. https://doi.org/10.5281/zenodo.1342401
54. Wang H, Shi H, Lin Ke et al (2018) A high-precision arrhythmia classification method based on dual fully connected neural network. Biomed Signals Process Control. https://doi.org/10.1016/j.bspc.2020.101874
55. Shao S, McAleer S, Yan R et al (2018) Highly-accurate machine fault diagnosis using deep transfer learning. IEEE Trans Ind Inform 15(4):2446–2455

56. Wang S-H, Muhammad K, Hong J et al (2020) Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization. Neural Comput Appl 32:665–680
57. Wieczorek M, Siłka J, Wozniak M et al (2020) Neural network powered COVID-19 spread forecasting model. Chaos Solitons Fractals. https://doi.org/10.1016/j.chaos.2020.110203
58. Wozniak M, Połap D (2017) Hybrid neuro-heuristic methodology for simulation and control of dynamic systems over time interval. Neural Netw. https://doi.org/10.1016/j.neunet.2017.04.013
59. Honglin L, Lin Bo, Chang P et al (2020) Fault diagnosis for high-speed train axle-box bearing using simplified shallow information fusion convolutional neural network. Sens. https://doi.org/10.3390/s20174930
60. Zhang Y-D, Dong Z, Wang S-H et al (2020) Advances in multimodal data fusion in neuroimaging: overview, challenges, and novel orientation. Inf Fus 64:149–187
61. Marcin W, Jakub S, Michał W (2020) Recurrent neural network model for IoT and networking malware threads detection. IEEE Trans Industr Inf. https://doi.org/10.1109/TII.2020.3021689
62. Izonin I, Kryvinska N, Tkachenko R et al (2019) An extended-input GRNN and its application. Proced Comput Sci 160:578–583
63. Tkachenko R, Mishchuk O, Izonin I et al (2019) A non-iterative neural-like framework for missing data imputation. Proced Comput Sci 155:319–326
64. Izonin I, Kryvinska N, Tkachenko R et al (2019) An approach towards missing data recovery within IoT smart system. Procedia Computer Science 155:11–18
65. Tkachenko R, Izonin I, Kryvinska N et al (2020) An Approach towards Increasing Prediction Accuracy for the Recovery of Missing IoT Data Based on the GRNN-SGTM Ensemble. Sens 20(9):2625
66. Izonin I, Tkachenko R, Kryvinska N et al (2019) Recovery of incomplete IoT sensed data using high-performance extended-input neural-like structure. Proced Comput Sci 160:521–526