Pattern Recognition 131 (2022) 108873

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

HCFNN: High-order coverage function neural network for image classification

Xin Ning[a,b,c], Weijuan Tian[c], Zaiyang Yu[a,b,c], Weijun Li[a,b,*], Xiao Bai[d,*], Yuebao Wang[c]

[a] Institute of Semiconductors, Chinese Academy of Sciences, Beijing, 100083, China
[b] Center of Materials Science and Optoelectronics Engineering & School of Microelectronics, University of Chinese Academy of Sciences, Beijing, 100049, China
[c] Cognitive Computing Technology Joint Laboratory, Wave Group, Beijing, 100083, China
[d] School of Computer Science and Engineering, Beihang University, Beijing, China

ARTICLE INFO

Article history:
Received 22 December 2021
Revised 19 June 2022
Accepted 22 June 2022
Available online 24 June 2022

Keywords:
DNNs
Neuron modeling
Heuristic algorithm
Back propagation
Computer vision

ABSTRACT

Recent advances in deep neural networks (DNNs) have mainly focused on innovations in network architecture and loss function. In this paper, we introduce a flexible high-order coverage function (HCF) neuron model to replace the fully-connected (FC) layers. The approximation theorem and proof for the HCF are also presented to demonstrate its fitting ability. Unlike the FC layers, which cannot handle high-dimensional data well, the HCF utilizes weight coefficients and hyper-parameters to mine underlying geometries with arbitrary shapes in an n-dimensional space. To explore the power and potential of our HCF neuron model, a high-order coverage function neural network (HCFNN) is proposed, which incorporates the HCF neuron as the building block. Moreover, a novel adaptive optimization method for weights and hyper-parameters is designed to achieve effective network learning. Comprehensive experiments on nine datasets in several domains validate the effectiveness and generalizability of the HCF and HCFNN. The proposed method provides a new perspective for further developments in DNNs and ensures wide application in the field of image classification. The source code is available at https://github.com/Tough2011/HCFNet.git

1. Introduction

In recent years, considerable advances have been made in the field of computer vision by using scalable end-to-end neural networks, especially deep neural networks (DNNs). DNNs have achieved state-of-the-art performance in various computer vision tasks, such as detection, tracking, segmentation, and classification. Most recent advances in DNNs focus on network architecture and loss function optimization. The performance of DNNs can be improved by deeply expanding the network and modifying multiple network layers of the network structure. In recent years, many classical neural network architectures were designed [1,2]. Another major innovation in DNNs is the loss function optimization. The loss function is used to acquire precise error measurements so that more information can be retrieved from the characteristics of DNNs [3,4]. Although the performance of DNNs [5,6] has been significantly improved in terms of network structure optimization and loss function optimization, DNNs still suffer from many problems, such as poor interpretability, dependence on large samples, difficulty in scaling up and training [7].

Research on neurobiology and cognitive neuroscience shows that the learning ability of neurons is an important basis for the biological nervous system to complete the tasks of learning and memory [8]. These problems motivate us to further improve the performance of DNNs in terms of neuron design and optimization. The current classical McCulloch and Pitts [9] (M-P) neuron and similar neurons are the most widely used neuron models. M-P neuron abstracts the information processing function of nerve cells as a simple function transformation between input and output and lacks the simulation of the learning mechanism of biological nerve cells [10]. In addition, these single neuron models based on M-P neurons are limited in dealing with complex problems because of lacking their nonlinear computing power [11], particularly the nonlinear separation problem [12]. The role of neuron models in DNNs is similar to that of cells in the human body. To simulate the partial response attributes exclusive to biological neurons, a radial basis function (RBF) was introduced to neural

* Corresponding author.
E-mail addresses: wjli@semi.ac.cn (W. Li), baixiao@buaa.edu.cn (X. Bai), yuebao17@163.com (Y. Wang).

https://doi.org/10.1016/j.patcog.2022.108873
0031-3203/

networks [13] by Broomhead and Lowe to project low-dimensional input data to RBF-based hidden space. RBF networks have been investigated in several studies from various perspectives. However, the uncertainty of RBF centers often has a significant impact on the performance, and the process and basis of reasoning in RBFs cannot be explained well. Flexible Transmitter (FT) model [14] is a novel bio-plausible neuron model with flexible synaptic plasticity. It employs a pair of parameters to model the transmitters between neurons and puts up a neuron-exclusive variable to record the regulated neurotrophin density, which leads to the formulation of the FT model as a two-variable two-value function, taking the commonly used MP neuron model as its special case. This modeling manner makes the FT model not only biologically more realistic, but also capable of handling complicated data, even time series data. Operational neural networks (ONNs) [15], a new generation of network models that address the major drawbacks of the conventional convolutional neural networks (CNNs): the homogenous network configuration with the "linear" neuron model that can 50 only perform linear transformations over the previous layer outputscan perform any (non-linear) transformation with a proper combination of "nodal" and "pool" operators. However, their restriction is the sole use of a single nodal operator for all (synaptic) connections of each neuron.

To solve the "black box" problem, which is difficult to explain in the field of neural network, many researchers have analyzed and studied neural networks in terms of geometry [16,17]. Zhang et al. [18] developed a geometric representation of the M-P neurons and proposed a neuronal construction algorithm called the covering learning method. The M-P neuron functions geometrically as a spatial discriminator of an n-dimensional space partitioned by a hyperplane. More importantly, they introduced the concept of "sphere neighborhoods," which are suitable for neuron improvement studies. Thus, we can consider that DNNs map multiple features of a sample (e.g., x_1, x_2, ... x_i, etc.) into n-dimensional space. In the n-dimensional space, each feature (x_i) can be represented as a "domain". The neural network is trained to obtain different neurons to cover these "domains". The combined output of multiple neurons enables the classification of the samples. Based on this interpretation, we find the performance of M-P neurons relatively limited[19]. Hence, the design of neurons can start from the construction of a hypersurface with flexible deformation ability in an n-dimensional space. The formed hypersurface has the following three properties: i) specification of the spatial size of the hypersurface, i.e., the degree of freedom, ii) specification of the spatial position information of the hypersurface, and iii) clarification of the stretching and deformation direction of the hypersurface. Wang et al. [20] proposed a general computational model to implement neural network hardware, that has the potential to implement many different hypersurfaces in n-dimensional space. Inspired by this concept, we further extend the neuron construction approach and learning paradigm by introducing a flexible high-order coverage function (HCF) neuron model for DNNs, which gives a theoretical justification for this neuron construction. HCF is characterized by adjusting the weights to iteratively cover the region of the sample subspace with a hypersurface, and it can be easily scaled up with the size of the parameter space. Furthermore, a high-order coverage function neural network (HCFNN) based on HCF is introduced. HCFNN can use various types of classical network structures (e.g., Res50, VGG16, etc.) as backbone networks, and the full-connected layer is replaced by the HCF neuron model. After features are extracted through the backbone network, the HCF neurons can accomplish image classification more flexibly and effectively. Our main contributions and innovations are as follows:

1) More flexible HCF neuron model for DNNs that constructs geometries in an n-dimensional space by changing weights and hyper- parameters and thus, possesses higher variability and plasticity. Furthermore, the approximation theorem and proof for arbitrary continuous in finite functions are presented, and the fitting ability of the HCF neuron model is demonstrated.
2) HCFNN architecture based on the HCF neuron is proposed; it is used to mine specific feature representations and achieve adaptive parameter learning. Next, a novel adaptive optimization method for weights and hyper-parameters is proposed to achieve effective network learning that has better expression and learning ability with fewer neurons.
3) We conduct experiments on nine datasets in several domains, including the two-spirals problem, natural object recognition, face recognition, and person re-ID. The results show that the proposed method has better learning performance and generalizability than the commonly used M-P and RBF neural networks. Our method can improve the performance of various image recognition tasks and acquire good generalization.

The structure of the remainder parts of the paper is as follows. Section 2 briefly introduces related works about covering learning. Section 3 describes the introduced HCF neuron model. Section 4 introduces HCFNN. Section 5 presents the related experiments and analyses. Finally, Finally, we conclude the paper and point out the future research direction in Section 6.

2. Related works

In this section, we give a brief review on the previous studies covering learning methods.

An M-P neuron [9] is an element constructed with multiple inputs and one output. The general form of the M-P neuron function is represented as $y = \mathrm{sgn}(\mathbf{w}^T\mathbf{x} - \varphi)$, where $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ is the input vector, $\mathbf{w} = (w_1, w_2, \cdots, w_n)^T$ denotes the weight vector, and φ is the threshold; thus,

$$\mathrm{sgn}(v) = \begin{cases} 1, & v > 0 \\ -1, & v \le 0 \end{cases}. \tag{1}$$

Note that $\mathbf{w}^T\mathbf{x} - \varphi = 0$ could be interpreted as a hyperplane P in an n-dimensional space. When $\mathbf{w}^T\mathbf{x} - \varphi > 0$, input vector $\mathbf{x}$ falls into the positive half-space of hyperplane P. Moreover, $y = \mathrm{sgn}(\mathbf{w}^T\mathbf{x} - \varphi) = 1$. When $\mathbf{w}^T\mathbf{x} - \varphi < 0$, input vector $\mathbf{x}$ falls into the negative half-space of hyperplane P, and then $y = -1$. In conclusion, the function of an M-P neuron can be geometrically considered as a spatial discriminator of an n-dimensional space partitioned by hyperplane P.

To improve the explanation and performance of the neuron models, Zhang et al. [18] proposed a new representation as follows. First, each input vector, $\mathbf{x}$, is assumed to be of equal length (norm). Thus, all input vectors are restricted to an n-dimensional sphere, S^n. Then, $\mathbf{w}^T\mathbf{x} - \varphi > 0$ denotes the positive half-space divided by hyperplane P and the intersection part between the positive half-space, called the "sphere neighborhood," as shown in Fig. 1. When input $\mathbf{x}$ falls into this region, then output $y = 1$; otherwise, $y = -1$.

If the weight vector, $\mathbf{w}$, has the same length as input $\mathbf{x}$, then $\mathbf{w}$ becomes the center of the neighborhood of the sphere, and it is a monotonically decreasing function of $r(\varphi)$, which becomes its radius. If the node function is a characteristic function, $\sigma(v)$, then

$$\sigma(v) = \begin{cases} 1, & v > 0 \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

The function $\sigma(\mathbf{w}^T\mathbf{x} - \varphi)$ of a neuron is the characteristic function of the sphere neighborhood on the sphere; that is, an M-P neuron corresponds to a sphere neighborhood in an n-dimensional sphere. A clear and intuitive picture of an M-P neuron is very helpful in the analysis of neural networks. We assume that the domain

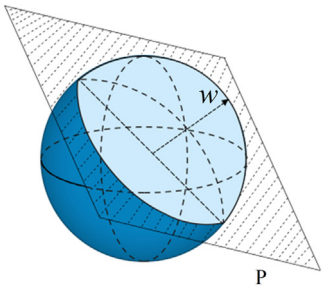

Fig. 1. Schematic of "sphere neighborhood." The function of an M-P neuron can be geometrically viewed as a spatial discriminator of an n-dimensional space partitioned by hyperplane *P*. When input $\mathbf{x}$ falls into positive region, then output $y = 1$; otherwise, $y = -1$.

of the input vectors is a bounded set in an n-dimensional space. S^n is an n-dimensional sphere in an (n+1)-dimensional space.

Assume that the domain of input vectors is a bounded set D of an n-dimensional space. We define a transformation, $T : D \rightarrow S^n, \mathbf{x} \in D$, such that

$$T(\mathbf{x}) = \left(\mathbf{x}; \sqrt{d^2 - |\mathbf{x}|^2}\right), \tag{3}$$

where $d \geq \max\{|\ \mathbf{x}\| \mathbf{x} \in D\}$. Thus all points of D are projected through transformation T. Evidently, neuron $(\mathbf{w}, \varphi)$ corresponds to a feature function on a "sphere neighborhood" on S^n with $\mathbf{w}$ as its center and $r(\varphi)$ as its radius. $T(\mathbf{x})$ denotes a transformation function, while the parentheses represent the vertical concatenate of $\mathbf{x}$ and $\sqrt{d^2 - |\mathbf{x}|^2}$, thereby this equation obtain $n+1$ dimensions of coordinates.

A hyperplane, i.e., the same linear form, $\mathbf{w}^T\mathbf{x} - \varphi$, in the transformed curved (quadratic) space, S^n, is equivalent to a quadratic surface in the original Euclidean space. Thus, partitioning by planes in a spherical space is equivalent to partitioning by quadratic surfaces in the original Euclidean space. For any given point on S^n, $n - dimensional$ sphere neighborhoods can always be utilized to partition them so that each point belongs to only one neighborhood. Thus, the "sphere neighborhood" is considered a geometrical coverage. The utilization of multiple "sphere neighborhoods" corresponds to multiple M-P neuron models to cover all the training samples. The exploration of geometric neighborhoods is a method of covering learning. The advantage of the spherical domain covering learning is its intuitive nature, based on which, the neural network training problem is converted into a point set coverage problem. This helps construct the artificial neural network.

3. HCF neuron model

3.1. HCF model definition

In this paper, a flexible HCF neuron model for DNNs is introduced, and is designed to explore the geometries to cover the feature subspace regions [19,20]. The introduced HCF neuron model possesses strong universality and complete functionality, and the model is formulated as follows:

$$y_i = \phi\left\{\left(\frac{\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})}{\left|\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})\right|}\right)^s \left|\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})\right|^p - \theta\right\}, \tag{4}$$

where $y_i \in [0, 1]$ is the output of the $i-th$ neuron; $\mathbf{x}$ is the input vector; $\mathbf{w}_{id}$ and $\mathbf{w}_{ic}$ are referred as the direction and center weights,and control the stretching direction and the center location of the geometry respectively. p denotes an exponential parameter, which is also the origin of our naming with high-order neurons, used to determine the freedom of the geometry. s is used to determine the sign of single entry, being either 0 or 1. If $s = 0$, the sign of single entry is always positive, which is usually used to construct the closed geometry. If $s = 1$, the sign of single entry is the same as the sign of $\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})$, which is usually used to construct the open hyperfaces. θ represents the activation threshold of neuron. ϕ represents activation function, which adopts Gaussian kernel activation. The utilization of the Gaussian kernel makes the output value meaningful in some specific intervals to compute the gradient in some specific intervals.

The introduced HCF neuron model possesses the functions of M-P and RBF neuron models. Taking the three-dimensional space as an example, if $\mathbf{w}_{ic} = [0, 0, 0]$, $s = 1$, and $p = 1$, the model degenerates to the perceptron M-P neuron, and in the case of $\mathbf{w}_{id} = [1, 1, 1]^T$, $s = 0$, and $p = 2$, the model degenerates to the RBF neuron model. While for other values of these parameters, the possibilities to implement much more different hypersurfaces is assured. Furthermore, the HCF neuron model can infinitely approximate continuous functions.

3.2. HCF model visualization

For the HCF neuron model, different parameters are utilized to decide different shapes of the coverage geometry. Fig. 2 displays the scope of geometries formed by different parameters in the three-dimensional space. The former two columns suggest the closed hypersurface, while the third column indicates a non-closed hypersurface. As shown in Fig. 2, dfferent parameter values can easily implement various geometries. As shown in the first two columns, when the components $\mathbf{w}_{id}$ and $\mathbf{w}_{ic}$ are set as constant, different values of p can be used to change the shape of geometries, and the edge of the geometry becomes more apparent with increasing p until $p = 1$, and then the geometry becomes closer to the hyperellipsoid with continued increase until $p = 2$, and after that point approaches a cuboid with increasing p. While for the last column, the hypersurfaces become more complicated with increasing p. Different values of $\mathbf{w}_{id}$ and $\mathbf{w}_{ic}$ change the stretch direction and the location of the geometries. Practically, the neuron has far more than three dimensions.

3.3. HCF model classification

HCF neuron model is established based on covering learning. Thus, the classification strategies are transformed to determine if the samples are covered in the geometries. Fig. 3 visualizes the covering geometries of three universal neuron models and their classification strategies for the artificial data. For more intuitive representation, each neuron model uses only one neuron. We take $s = 1$ as an example to interpret the covering strategy for HCF neuron model. The basis function is simplified as: $\left|\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})\right|^p - \theta$, and the classification surface that corresponds to the equation is a closed geometry. Then, Fig. 3 can be interpreted as follows:

1) The M-P neuron model. The corresponding classification surface is hyperplane, the covering ability is proved detail in [18], which gives the geometric representation of M-P neuron. The working mechanism is interpreted as exploring the sphere neighborhoods, shown in Fig. 3, marked as M-P neuron.
2) The RBF neuron model. The corresponding classification surface is hypersphere, the covering ability is realized by using radius constraint, which restricts the positive samples inside of the hyperspheres, and then the function of RBF neuron is interpreted as exploring the hyperspheres, shown in Fig. 3, marked as RBF neuron.
3) The HCF neuron model. The corresponding classification surface is hypersurface with various shapes. The covering ability is realized by using threshold to determine if the feature points

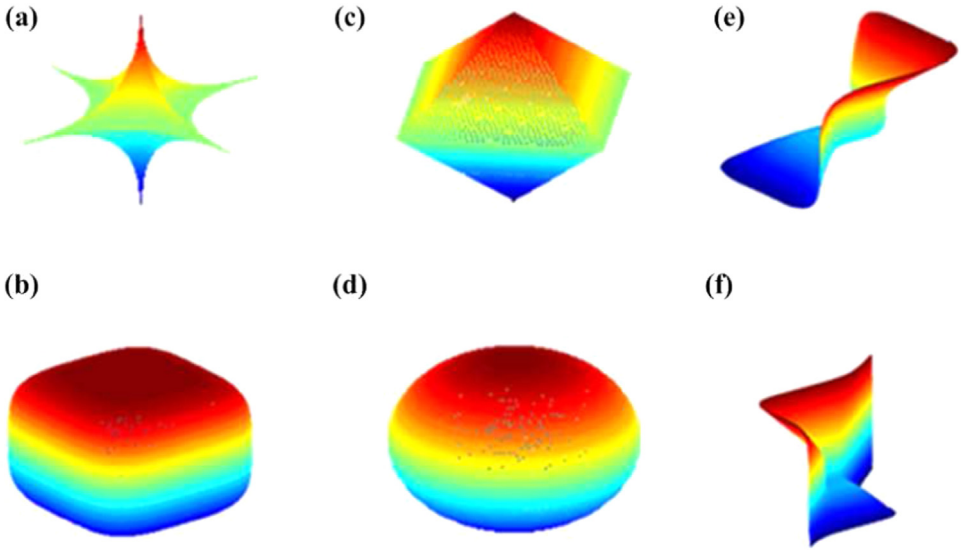

Fig. 2. The hypersurfaces of HCF neuron model with various parameters in Eq. (4), where p determines the fatness of the geometry, $\mathbf{w}_{id}$ is used for the stretch and deformation of the geometry in different directions, and $\mathbf{w}_{ic}$ determines the location of the geometry in the space. (a) $\mathbf{w}_{id} = [0.5, 0.5, 0.5]^T$, $\mathbf{w}_{ic} = [0, 0, 0]^T$, $s = 0$, $p = 1/2$. (b) $\mathbf{w}_{id} = [0.5, 0.5, 0.5]^T$, $\mathbf{w}_{ic} = [0, 0, 0]^T$, $s = 0$, $p = 4$. (c) $\mathbf{w}_{id} = [0.5, 0.5, 0.5]^T$, $\mathbf{w}_{ic} = [0, 0, 0]^T$, $s = 0$, $p = 1$. (d) $\mathbf{w}_{id} = [0.5, 0.5, 0.5]^T$, $\mathbf{w}_{ic} = [0, 0, 0]^T$, $s = 0$, $p = 2$. (e) $\mathbf{w}_{id} = [0.5, -0.5, -0.5]^T$, $\mathbf{w}_{ic} = [0, 0, 0]^T$, $s = 1$, $p = 2$. (f) $\mathbf{w}_{id} = [0.5, -0.5, -0.5]^T$, $\mathbf{w}_{ic} = [0, 0, 0]^T$, $s = 1$, $p = 5$.

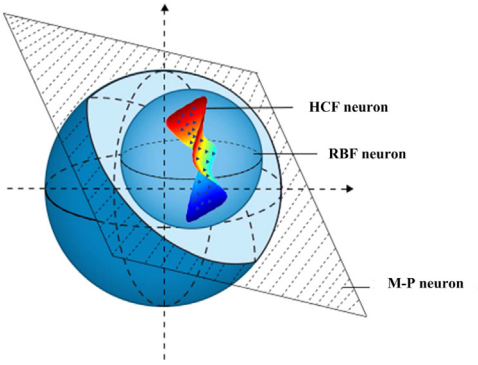

Fig. 3. Fitting regions of different neurons on unit sphere space for positive samples. The geometric representations of M-P, RBF, and HCF in the figure are planar, spherical, and special surfaces, respectively. The positive samples are represented as blue triangles, which are distributed only in a small region in the sphere's feature space. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

are inside of the covering geometries, and then the function of HCF neuron is interpreted as exploring the geometries, shown in Fig. 3, marked as HCF neuron.

As mentioned above, aiming to cover the feature subspace regions, the introduced HCF neuron model possesses great possibility to explore abundant geometries. HCF modifies double weights and the hyper-parameters according to the data distribution, and the acquired geometry is more compact in covering areas than others, thus ensuring stronger fitting ability.

3.4. Model approximation study

The introduced HCF neuron model uses double weights and exponentials to project the sample data into n-dimensional space, which benefits the construction of flexible geometries in the n-dimensional space. The introduced HCF neuron model possesses uniform approximation, which alleviates the problem of sample inseparability in low-dimensional space. To demonstrate the fitting ability of the HCF neuron model, the proof that a linear combination of HCF neurons can approximate any continuous bounded function is given below. For the proof of **Theorem 1**, we refer to the approximation proof method [21].

Theorem 1 For any $f \in C(I_n)$ and $\varepsilon > 0$, there is a sum of the form $G(\mathbf{x})$, for which, $|G(\mathbf{x}) - f(\mathbf{x})| < \varepsilon$, for all $\mathbf{x} \in I_n$,

$$G(x) = \sum_{i=1}^{m} \phi\left(\left(\frac{\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})}{\left|\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})\right|}\right)^s \left|\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})\right|^p - \theta\right). \tag{5}$$

where I_n denotes the n-dimensional unit cube $[0, 1]^n$, $C(I_n)$ is the space of all continuous bounded functions defined in I_n. Borel measures on I_n is denoted by $M(I_n)$. $\phi(t) = \frac{1}{1+e^{-t}}$ is sigmoidal function.

Proof. Let V be the set of all functions of the form $G(\mathbf{x})$. It is easy to infer that V is a linear subspace of $C(I_n)$. Let the closure of V be $\bar{V}$, where $\bar{V} \neq C(I_n)$. $\bar{V}$ is a subspace of $C(I_n)$. From the Hahn-Banach theorem there is a bounded linear function H on $C(I_n)$, and H satisfy $H(C(I_n)) \neq 0$ but $H(\bar{V}) = H(V) = 0$. That is, H is the extension of some function defined in $\bar{V}$ on $C(I_n)$. By the Riesz Representation Theorem, bounded linear functional H has the following form:

$$H(h) = \int_{I_n} h(\mathbf{x}) d\mu(\mathbf{x}). \tag{6}$$

for some $\mu \in M(I_n)$ and for $h \in C(I_n)$. We know that $\forall \mathbf{w}_{ic}, \mathbf{w}_{id}^T$ and θ,

$$\phi\left(\left(\frac{\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})}{\left|\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})\right|}\right)^s \left|\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})\right|^p - \theta\right) \in \bar{V}. \tag{7}$$

According to Eq. (6), $\forall \mathbf{w}_{ic}, \mathbf{w}_{id}^T$ and θ satisfy the following equation:

$$\int_{I_n} \phi\left(\left(\frac{\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})}{\left|\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})\right|}\right)^s \left|\mathbf{w}_{id}^T(\mathbf{x} - \mathbf{w}_{ic})\right|^p - \theta\right) d\mu(\mathbf{x}) = 0. \tag{8}$$

According to **Lemma 1**, this condition implies that $\mu = 0$ contradicting our assumption. That means that $\bar{V}$ and $C(I_n)$ are equal. Hence, the subspace V is dense in $C(I_n)$. that is, for any $f \in C(I_n)$ and $\varepsilon > 0$, there is a $G(\mathbf{x})$, $\forall \mathbf{x} \in I_n$ have $|G(\mathbf{x}) - f(\mathbf{x})| < \varepsilon$. □

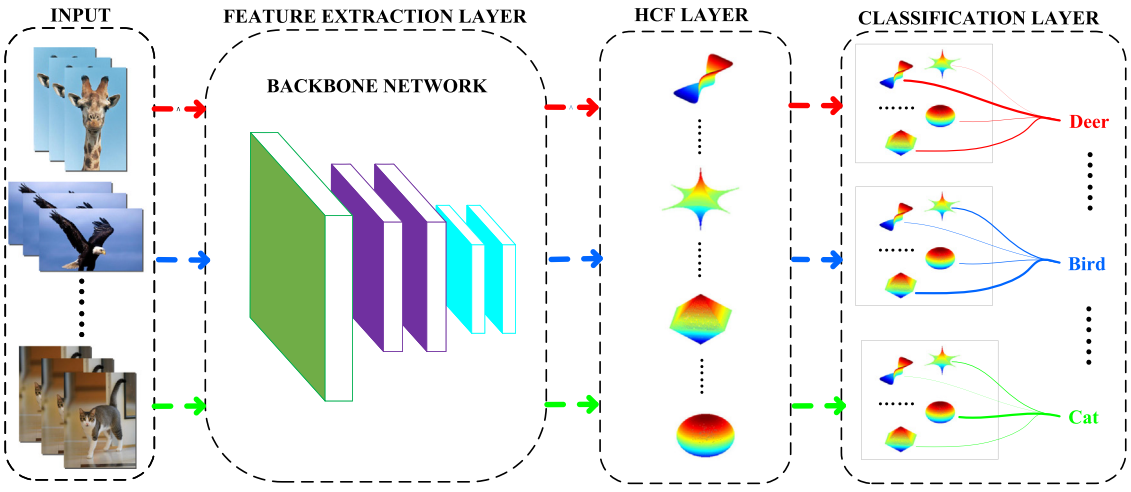

Fig. 4. Overview of the proposed high-order coverage function neural network (HCFNN) framework.The thickness of the line indicates the weight size of each neuron; the greater the weight, the thicker the line.

Lemma 1 $\forall \mathbf{w}_{id}, \mathbf{w}_{ic} \in R^n$ and $\theta \in R$, for a measure $\mu \in M(I_n)$, if

$$\int_{I_n} \phi\left(\left(\frac{\mathbf{w}_{id}^T(\mathbf{x}-\mathbf{w}_{ic})}{\left|\mathbf{w}_{id}^T(x-\mathbf{w}_{ic})\right|}\right)^s \left|\mathbf{w}_{id}^T(\mathbf{x}-\mathbf{w}_{ic})\right|^p - \theta\right) d\mu(\mathbf{x}) = 0, \tag{9}$$

then $\mu = 0$.

Proof. when $s = 0$, Eq. (9) is simplified to:

$$\int_{I_n} \phi\left(\left|\mathbf{w}_{id}^T(\mathbf{x}-\mathbf{w}_{ic})\right|^p - \theta\right) d\mu(\mathbf{x}). \tag{10}$$

The integral region I_n is divided into k equal parts by Riemann partition, then $I_n = \bigcup_{j=1}^{k} I_n^j$. when $k \to \infty$, we write the integral of Eq. (10) as a series:

$$\begin{aligned} &\textstyle\int_{I_n} \phi\left(\left|\mathbf{w}_{id}^T(\mathbf{x}-\mathbf{w}_{ic})\right|^p - \theta\right) d\mu(x) = \\ &\textstyle\sum_{j=1}^{k} \phi\left(\left|\mathbf{w}_{id}^T\left(\mathbf{x}^j-\mathbf{w}_{ic}\right)\right|^p - \theta\right)_{\mathbf{x}^j \in I_n^j} \cdot \mu\left(I_n^j\right). \end{aligned} \tag{11}$$

If $\forall \mathbf{w}_{id}, \mathbf{w}_{ic} \in R^n$ and $\theta \in R$,

$$\int_{I_n} \phi\left(\left|\mathbf{w}_{id}^T(\mathbf{x}-\mathbf{w}_{ic})\right|^p - \theta\right) d\mu(\mathbf{x}) = 0, \tag{12}$$

namely,

$$\sum_{j=1}^{k} \phi\left(\left|\mathbf{w}_{id}^T\left(\mathbf{x}^j-\mathbf{w}_{ic}\right)\right|^p - \theta\right)_{\mathbf{x}^j \in I_n^j} \cdot \mu\left(I_n^j\right) = 0. \tag{13}$$

Because $\phi(t) = \frac{1}{1+e^{-t}}$, $\forall t \in R$ where $\phi(t) > 0$. Therefore, $\mu(I_n{}^j) = 0$ can be deduced from Eq. (13).

Similarly, when $s = 1$, the lemma also holds. □

4. HCFNN Architecture

4.1. Network structure

In this section, an HCFNN framework based on the HCF neuron model, which assures the learning adaptability and great performance. The network structure is illustrated in Fig. 4, consisting of three components: 1) Image features are extracted by the backbone network. The backbone is different in specific tasks; 2) The introduced flexible HCF neuron model is able to construct various geometries for feature points covering; 3)The classification layer consists of a weighted summation combination of HCF neurons. The thicknesses of lines represent the weights. There exists a normal assumption that each HCF neuron model is used for a specific attribute classification. For the purpose of extracting more abundant information from feature, the numbers of neurons in the HCF generally exceeds the number of classification labels.

4.2. Network learning formulation

The proposed HCFNN, the back propagation (BP) [22] and genetic algorithm (GA) [23] alternately updating parameters. Specifically, BP is used to update weights, including the direction and the center weights, $\mathbf{w}_{id}$ and $\mathbf{w}_{ic}$. GA is used to update the power parameter p and symbol parameter s. The opposite of the loss function, namely the -1 multiplying the loss value, is used as the fitness function. The learning of the model imitates the training criterion of the EM model. Firstly, the initial values of p_m and p_c are given to train the model weight parameters and hyperparameters until the model converges. The training steps are repeated with the value of p_m is increased, and p_c is decreased to obtain the optimal hyperparameters. Three operations, selection, crossover, and mutation, are used for parameter updates. The selection operation is beneficial obtain the optimum solution. The crossover operation is used for optimal value generation, while the mutation operation enables the approximation of the global optimal coverage. The chromosomes are encoded in IEEE standard floating point code. The new parameters are generated using the random single-point crossover method and the real-valued mutation method. Details of the HCFNN training algorithm are shown in Algorithm 1, where W_{bp} is the trainable parameters of HCFNN with BP, H_{ge} represents the hyper-parameters of HCF, T_{iter}, T_{bp} and T_{ge} respectively denote the total steps, BP iterations, and GA generations in one step. W_{bp} contains the backbone learnable parameters as well as the HCF neuron's weight parameters W_{id} and W_{ic}.

The GA algorithm is used to update of hyper-parameter H_{ge}, which include p and s. The detailed implementation is shown in Algorithm 2, where crossover probability p_c and mutation probability p_m are hyperparameters. They were determined empirically and their value were fixed during implementation.

Based on the above learning algorithm, our proposed HCFNN can automatically adjust weight coefficients and hyper-parameters using BP and GA algorithm. Thus different hypersurfaces can be acquired to cover the feature distribution of the samples. Besides, HCF introduces more multiplication and exponential operations, thus increasing the model's access to memory and conclusively

Algorithm 1: HCFNN Training.

Input: backbone network weights: W_{bp}, hyper-parameters H_{ge}, total iterations T_{iter}, BP iterations T_{bp}, generations T_{ge}, and the candidate individuals for p and s, and HCF neuron weights w_{ic} and w_{id}.

Conduct the step 1–2 in GA **Algorithm 2** to select the optimal hyper-parameters H_{ge} in the candidate individual;
for *iter in* T_{iter} **do**
  Conduct BP algorithm;
  Update weight W_{bp}, w_{ic} and w_{id} for T_{bp} times;
  Conduct the step 3–11 in GA **Algorithm 2** to obtain the current optimal p and s;
  Update hyper-parameters H_{ge} for T_{ge} times;
end

Output: Optimal parameters W_{bp}, w_{ic}, w_{id} and H_{ge}.

Algorithm 2: Genetic Algorithm Learning.

Input: generations T_{ge} and chromosome num N, cross probability p_c and mutation probability p_m, where $p_m + p_c = 0.98$.

Encode the value of p and s with float-point encoding to construct N chromosomes, used as initial population m_1;
Select the optimal hyper-parameters H_{ge} in the candidate individual m_1;**for** *gen in* T_{ge} **do**
  Conduct forward propagation to obtain fitness value, and sorted in descending order;
  Select the top $0.15 \times N$ individuals as the next generation m_2, the individual number is n_1, and the remaining are m_3;
  while *the new individuals* $< N - n_1$ **do**
    Select a pair of chromosomes with high fitness value in m_3;
    Conduct random single point crossover to generate new individuals with p_c;
    Conduct real value mutation to generate new individuals with p_m;
  end
end

Output: The optimal p and s.

reducing computational cost. The number of Flops increased by 3.3% and the inference run time reduced by approximately 1%.

5. Experimental results and analyses

5.1. Datasets and experimental settings

To validate the performance of our proposed method, two-spirals problem (TSP) [27] and three different types of image classification datasets were utilized: natural object recognition, face recognition, and person re-ID. Examples of images of the three types of datasets are shown in Fig. 5.

5.1.1. Two-spirals problem

TSP [27] is a famous benchmark of the artificial neural network test. We conduct TSP task to observe the ability of the model to solve linearly inseparable problems.

5.1.2. Natural object recognition datasets

CIFAR-10 [28] is composed of 60,000 color images with size 32×32 from 10 categories. ImageNet [24], as shown in Fig. 5 (a), composed of 1.2 million labeled images and 1000 classes in the training set, and 50,000 and 100,000 tagged images in the validation and testing sets, respectively.

5.1.3. Face recognition datasets

YTF [29] collects 3425 YouTube videos from 1595 subjects (a subset of the celebrities in LFW [29]). These videos are divided into 5000 video pairs and 10 splits, used to implement video-level face verification. LFW [29] includes 13,233 face images collected from the web. This dataset consists of 5749 identities with two or more images of 1680 people. MegaFace [25], as shown in Fig. 5 (b), disturbed by one million level interference is extremely challenging beyond expectation. MegaFace [25] dataset consists of a gallery set and a search test set, in which the former covers over 1 million images gathered from 690,000 individuals.

5.1.4. Person re-ID datasets

DukeMTMC-reID [26] in Fig. 5 (c) is a subset of DukeMTMC, which contains 36,411 photos of 1812 people collected by 8 cameras, of which 1404 people appeared under two cameras, and 408 people appeared under only one camera. Market-1501 [30] was collected from the Tsinghua University campus and contains 1501 people captured by 6 cameras. CUHK03 [31] collected 14,097 photos of 1467 people.

The experimental parameter settings were as follows: 1) $p \varepsilon (0, 5]$ in Eq. (4). Repeated trials demonstrate that the coverage geometry shape changed obviously until $p = 5$, and then remained unchanged with the increase of p. 2) The number of BP iterations T_{bk} was set as $T_{bk} = 5$ in one iteration, and the generations $T_{ge} = 10$. 3) All of the experiments were conducted on an NVIDIA GeForce GTX-1080Ti GPU. 4) Details of the individual experiments are described in the corresponding sections.

5.2. Parameters selection

In this section, we investigate the selection of two genetic parameters, namely cross probability p_c and mutation probability p_m of GA in Algorithm 2 during HCF neuron model learning, which is used to indicate the importance of p_c and p_m for classification. Taking CIFAR-10 [28] dataset as an example, the experiment aimed to investigate the influence of varying p_c and p_m on classification performance. We need more abundant parameter p, and thus p_m and p_c were initialized as 0.01 and 0.97, both of which were ascertained by the greedy search method, that is, p_c decreased to 0.17, and meanwhile, p_m was increased to 0.81, and the intervals were set as 0.05, where $p_m + p_c = 0.98$ for convenience. Fig. 6 shows the curves of natural object recognition accuracies on CIFAR-10 [28] dataset versus p_c and p_m, respectively.

The result proves that the classification performance increases with the decrease of p_c and increase of p_m until $p_c = 0.57$ and $p_m = 0.41$, and then deteriorates with the continuing decrease of p_c and continuing increase of p_m. The two curves in Fig. 6 are completely symmetrical, it holds mathematically. The result in the Fig. 6 was obtained under the condition that the crossover probability and the mutation probability were both available and changing simultaneously. In general, the crossover probability is set to be larger, and the mutation probability is set to be smaller. Empirically, the crossover probability gradually reduces from 0.97 (the upper limit) at an interval of 0.1. At the same time, the mutation probability increases from 0.01(the lower limit) at an interval of 0.1. Through further analysis, it was found that smaller cross probability p_c and larger mutation probability p_m imply that most new generated values of p had higher relevance with the parents, meanwhile the newly generated values with higher randomness were relatively few, which greatly differed from the parents, and helped reach the global optimization. Conversely, a larger value of cross probability p_c and smaller mutation probability p_m indicate

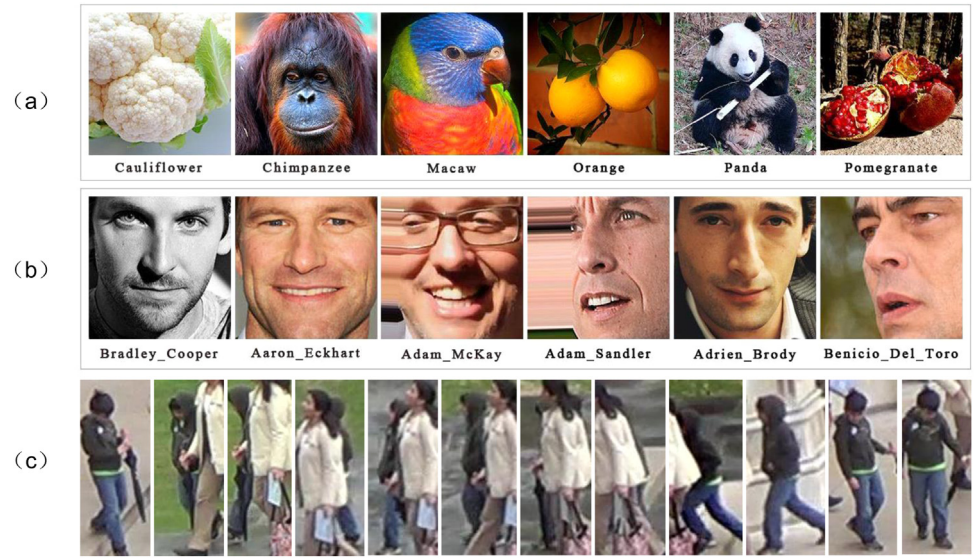

Fig. 5. Examples of the three types of datasets. From top to bottom: a) Natural object recognition task (ImageNet [24]); b) Face recognition task (MegaFace [25]); c) Person re-ID task (DukeMTMC-reID [26]).

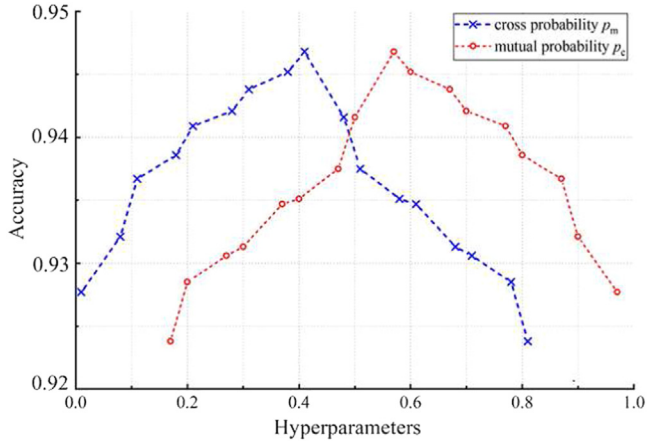

Fig. 6. Classification accuracy vs. threshold cross probability p_c and mutation probability p_m on CIFAR-10 dataset.

that most newly generated values of p without good inheritance from the parents with great performance were retained for natural object recognition, and this assuredly resulted in a low accuracy. Consequently, the optimum cross and mutation probability for CIFAR-10 [28] dataset were eventually determined as 0.57 and 0.41, respectively. Similarly, $p_c = 0.58$ and $p_m = 0.40$ for MegaFace [25] dataset, $p_c = 0.51$ and $p_m = 0.47$ for ImageNet [24] dataset.

5.3. Evaluation on two-spirals problem

In this experiment, we compared the HCF neuron with the M-P and RBF neurons in TSP for nonlinear recognition problems. We set two different spiral points, where each point on the spiral corresponded to a category. There were 120 points per class. It was randomly and uniformly divided into a training set and a test set, the test set contains exactly 60 randomly selected points per class. To ensure the fairness of the experiment, we used the same network structure, that is a three-layer neural network structure (one input layer, one hidden layer, and one output layer). The neurons used for comparison were set in the hidden layer. The only experimental variable was the number of neurons in the hidden layer.

Table 1 illustrates the favorable performance of the HCF neuron model over the classical neural models in terms of classification accuracy and neuron number. When the number of neurons is small (e.g., when there are 7, 8), the accuracy of the HCF neuron model is nearly 10% higher than that of M-P model. Also, the HCF model got the highest accuracy rate of 99.84%. The M-P model and the RBF model achieve the best accuracy when the number of neurons was 11.However, the HCF model only required 10 neurons to achieve the best accuracy. The experimental results show that the HCF neuron model has completely better performance on TSP compared to RBF and M-P neuron models. The advantage of HCF is more noticeable when the number of neurons is small. The reason is that the HCF neuron model can generate a variety of topological structures and approximate any continuous function, so it has better nonlinear fitting ability.

To further clarify the details and classification performance of the three neuron models, Fig. 7 reveals the visualization of the two-spirals classification strategy. The M-P neuron model can only construct hyperplanes, and its representation ability is very limited. The RBF and HCF neuron models based on covering learning adopt multiple covering units to achieve sample coverage in the complex feature space. However, the RBF neuron model obtains various hypersurfaces by adaptively modifying the radius only. In contrast, the HCF neuron model constructs various hypersurfaces by adaptively modifying parameters and hyper-parameters according to the distribution of data, ensuring stronger fitting ability. Meanwhile, hypersurfaces constructed by the introduced HCF neuron model can precisely cover the support of the data distribution, which means that it can mine better feature representations and hypersurfaces, which benefit classification.

5.4. Evaluation on natural object recognition

For the natural object recognition task, we used CIFAR-10 [28] dataset and the large dataset ImageNet [24]. In the CIFAR10 dataset experiment, we used the stochastic gradient descent(SGD) with batch size of 128. A total of 100 epochs were set for the experiment. We use a weight decay of 0.0001 and a momentum of 0.9 and a learning rate of 0.01. In ImageNet dataset experiment, the image is resized with its shorter side randomly sampled in [256, 480] for scale augmentation. Randomly samples the short

Table 1
Classification results(%) of different neural models on TSP benchmark.

Neuron types	Number of neurons					
	7	8	9	10	11	12
M-P	85.36 ± 0.21	90.62 ± 0.11	93.56 ± 0.25	94.42 ± 0.32	96.06 ± 0.18	94.21 ± 0.23
RBF	90.21 ± 0.37	94.39 ± 0.54	98.75 ± 0.79	99.31 ± 0.19	99.42 ± 1.02	98.53 ± 0.38
HCF	95.03 ± 0.02	96.53 ± 0.20	98.65 ± 0.53	99.84 ± 0.16	99.20 ± 0.28	99.01 ± 0.96

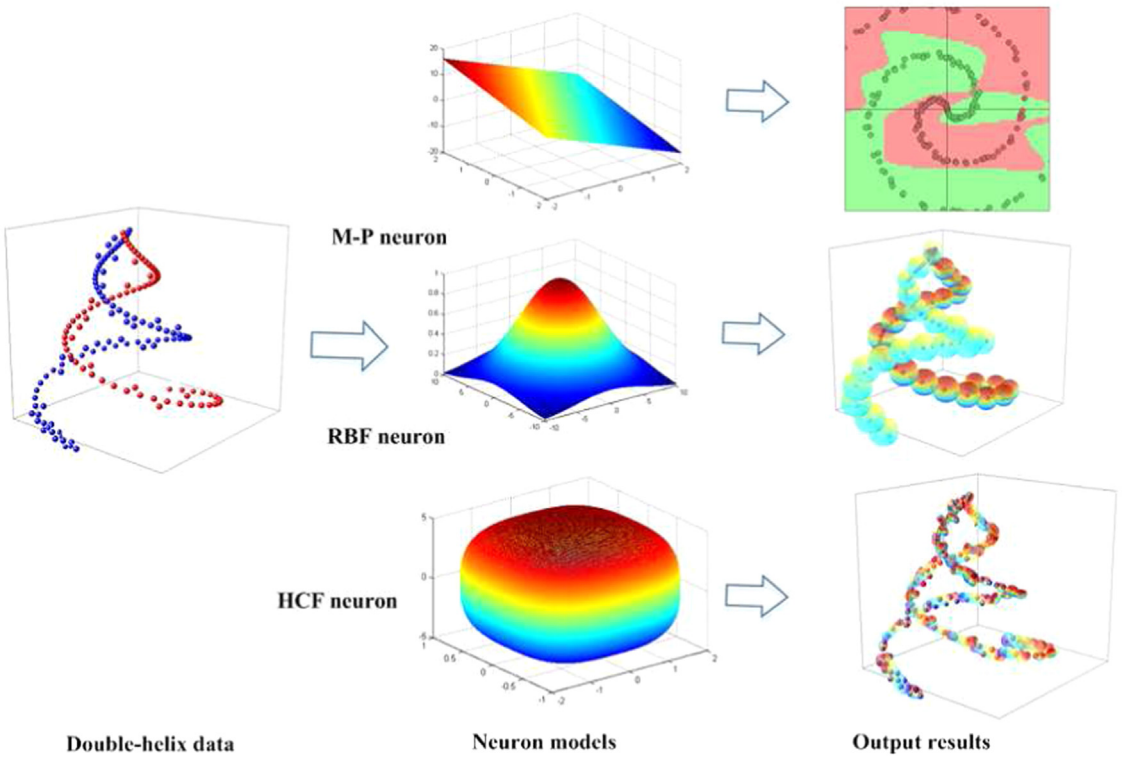

Fig. 7. Classification result visualization. From left to right:1) the synthetic double-helix data; 2) the geometry representation in three-dimensional space, that is the hyperplane, hypersphere and hypersurface respectively using M-P, RBF and HCF neuron models; 3) classification representation.

side of the image in [256, 480] for scale up augmentation, and crop 224×224 by randomly sampling from an image or its horizontal flip and subtracting the average of each pixel. We used SGD with a mini-batch size of 256, and the learning rate started at 0.1 and was divided by 10 as the error plateaued. We used a weight decay of 0.0001 and a momentum of 0.9. For the test, we used the standard 10-crop test for the comparison study. We used ResNet50 [32], VGG16 [33], and GoogleLeNet [34] as the backbone networks for comparison, respectively. In the classification layer, we selected M-P, RBF and HCF neuron models as classifiers for comparison, respectively. To make the presentation uniform, we call the neural networks using HCF neurons collectively as HCFNN, those using different backbone are called HCFNN(VGG), HCFNN(GoogLeNet) and HCFNN(ResNet50).

As shown in Table 2, the HCF neuron model showed a significant advantage on ImageNet and CIFAR-10 datasets. The classification error rate and running time per batch obtained the best performance for all ackbone networks useing HCF. Therefore, HCF can improve the performance of many classical networks by replacing the neuron model. The reason is that the adaptive learning algorithm in HCFNN can better combine HCF with any backbone network, which also makes the proposed method more universal. From the experiment of HCF combined with different backbone networks, using ResNet50 as the backbone network achieved the best result. On ImageNet dataset, the top-5 error rate reached $3.38 \pm 1.06\%$ and the running time reached 36.30 ms. On CIFAR-10 dataset, the accuracy rate reached 96.17% and the running time reached 5.23 ms.

5.5. Evaluation on face recognition

This task is suitable for evaluating the ability of neurons in classifiers. The experiments are conducted on LFW [29], YTF [35], and MegaFace [25] datasets. The methods used are DeepFace, FaceNet, Center Face, SphereFace, ArcFace and CosFace. To ensure the fairness of the experiments, all methods used the same protocol [29], which comes from LFW dataset. The fine-tuned networks on our face datasets used SGD and standard back-propagation with a mini-batch size of 60. We used an initial learning rate of 1e-3, and a momentum of 0.9. This was applied to the entire networks.

From Table 3, the recognition accuracy of FaceNet with RBF neuron model outperforms FaceNet with M-P neuron model on LFW, YTF and MegaFace datasets. After switching to HCFNN(FaceNet), the accuracy on the three datasets improved to 99.85%, 97.94% and 81.25% respectively. After combining HCFNN with other methods, the combination of HCFNN(ArcFace) could achieved the best results. This model achieved 99.87%, 98.32% and 82.36% results on LFW, YTF, and MegaFace datasets, respectively.

To further obtain fair experimental results, we retrained HCFNN(ArcFace) as ArcFace employs MS1MV2 dataset. After that, The cumulative match characteristic (CMC) and receiver operating

Table 2
Classification error and inference running time on imagenet dataset and CIFAR10 dataset .

Networks	Neurons	Flops(billion)	ImageNet			CIFAR-10	
			Top-5 err(%)	Top-1 err(%)	Running time(ms)	Top-1 err(%)	Running time(ms)
VGG16	M-	15.3	9.332.81	25.382.06	117.79	5.730.78	11.14
GoogLeNet	P	1.5	9.151.36	25.011.27	59.42	4.870.21	5.62
ResNet50		3.8	6.711.21	22.851.07	36.66	4.780.59	3.43
VGG16	RBF	15.7	9.052.67	24.972.93	117.53	5.640.58	11.04
GoogLeNet		1.9	9.011.47	24.841.51	59.02	4.780.55	5.57
ResNet50		4.2	6.341.15	22.261.23	36.54	4.690.73	3.39
VGG16	HCF	15.8	7.292.53	23.472.76	116.63	4.790.49	10.93
GoogLeNet		2	6.391.34	22.781.26	58.84	3.920.46	5.51
ResNet50		4.3	**5.371.01**	21.691.13	**36.3**	**3.830.62**	**3.36**

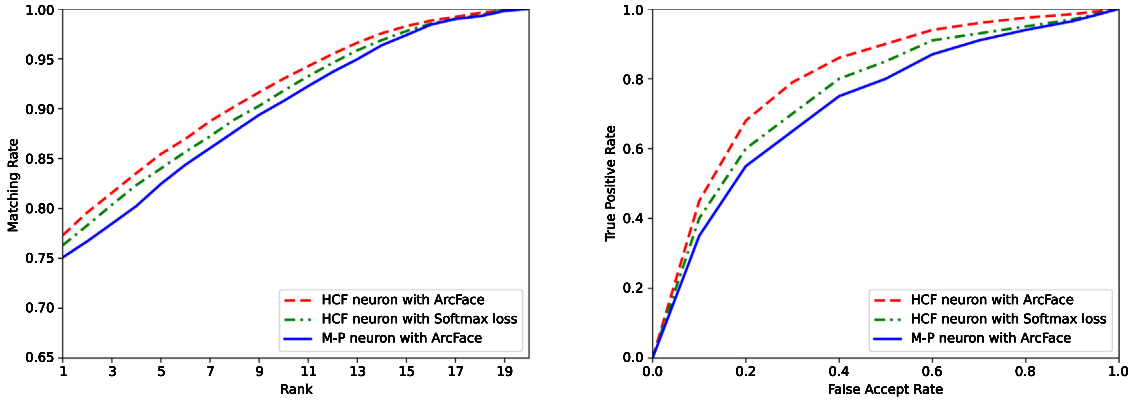

Fig. 8. ROC and CMC curves of HCFNN with different loss functions and neuron models for face recognition on MegaFace dataset.

Table 3
Recognition accuracy(%) on LFW, YTF and MegaFace datasets compared to state-of-the-art methods .

Method	Recognition accuracy/%			
	LFW[29]	YTF[35]	MegaFace[25]	
			Rank1	Veri
DeepFace[36]	97.35	91.40	–	–
FaceNet [37]	99.63	95.10	65.50	77.10
CenterFace[38]	99.28	94.90	65.23	76.52
SphereFace[39]	99.42	95.00	75.77	89.14
ArcFace[40]	99.83	98.02	81.03	96.98
CosFace[41]	99.73	97.60	77.11	89.88
SCF[42]	99.82	–	81.40	97.15
URFace[43]	99.78	97.92	78.60	95.04
HCFNN(FaceNet)	99.85	97.94	81.25	97.12
HCFNN(ArcFace)	**99.87**	**98.32**	**82.36**	**97.71**

Table 4
Comparison of our method with the state-of-the-art methods on market-1501 and DukeMTMC-reID datasets .

Method	Market-1501[30]		DukeMTMC-reID[26]	
	mAP(%)	Rank-1(%)	mAP(%)	Rank-1(%)
DuATM[45]	76.20	91.30	64.40	81.80
PCB+RPP[46]	81.60	93.80	69.20	83.30
HA-CNN[47]	75.70	91.20	63.80	80.50
AACN[48]	85.90	66.90	59.25	76.84
BDB[49]	86.70	95.30	76.00	89.00
BagOfTricks[50]	85.90	94.50	76.40	86.40
CDNet[51]	86.00	95.10	76.80	88.60
SONA[52]	88.67	95.68	78.05	89.25
SpCL[53]	76.70	90.30	68.80	82.90
Auto-ReID[54]	85.10	94.50	75.10	88.50
HCFNN	**88.90**	**95.70**	**79.30**	**89.40**

Table 5
Comparison of our method with the state-of-the-art methods on CUHK03 dataset .

Method	Labeled		Detected	
	mAP(%)	Rank-1(%)	mAP(%)	Rank-1(%)
PCB+RPP[46]	–	–	57.50	63.70
HA-CNN[47]	41.00	44.40	38.60	41.70
BDB[49]	76.70	79.40	73.50	76.40
SONA[52]	79.23	81.85	76.35	79.10
Auto-ReID[54]	73.00	77.90	69.30	73.30
HCFNN	**79.80**	**82.10**	**77.50**	**79.30**

characteristic (ROC) were tested on MegaFace dataset with the new proposed protocol [44].

From Fig. 8, HCFNN(ArcFace) which is the red curve performs best. The face recognition performance is improved by replacing the M-P neuron with our HCF neuron model. This result shows that HCFNN obtains better results in face recognition task and has good robustness to different loss functions.

5.6. Evaluation on person re-ID

The Re-ID task is characterized by the fact that the pictures of people have more similar appearance features and the difference between the pictures is relatively small. The experiments were conducted on three datasets, Market-1501 [30], DukeMTMC-reID [26] and CUHK03 [31]. Part-based convolutional baseline (PCB) was chosen for our basic method. Pre-trained ResNet-50 was chosen to build the backbone network of HCFNN. The downsampling in Resnet Block 4 was removed, and the batch size was set to 32. The picture size was uniformly adjusted to 384128, and the data were enhanced by random flipping and erasure of images. The training epochs were 200 and the initial learning rate was 0.00035, which was decayed by half every 50 epochs.

As shown in Tables 4 and 5, the experimental and analyses results prove that our method yields the best performance in comparison with the most advanced methods. For the Market-1501

[30] dataset, we attained mAP/Rank-1= 88.90%/95.70%; on DukeMTMC-reID [26] dataset, we achieve mAP/rank-1= 79.30%/89.40%. Notably, our method outperformed the currently best SOTA [52] model on CUHK03 [31] dataset as well. The better results show that HCFNN possesses strong generalizability. It is also explicit that the introduced HCF neuron model and HCFNN can deal with more complex and various features and exploit useful features of a person.

6. Conclusion

In this paper, a flexible HCF neuron model for DNNs is first introduced, which uses weights and hyper-parameters to construct various geometries in n-dimensional space. The constructed HCF adjusts the weights to iteratively cover the region of the sample subspace with a hypersurface, and it can be easily scaled up with the size of the parameter space. Besides, this paper provides a mathematical proof that HCF can approximate any continuous function with any precision. Thus, the rationality and potential of the HCF neuron in DNNs are explored, and the solid foundation for the wide application of HCF neuron is lied. Based on HCF neural model, an HCFNN architecture is introduced, which combines the feature extraction network and HCF model for classification. With the proposed functional neuron model, more specific and informative feature representations could be mined. Moreover, a new optimization method is proposed to adjust the weights and parameters adaptively using BP and GA algorithm. Thus, different hypersurfaces can be acquired to cover the feature distribution of the samples. The proposed method shows excellent performance on several tasks, including the two-spirals problem, natural object recognition, face recognition, and person re-ID. Experimental results show that the proposed method has better learning performance and generalization ability. In addition, our method can improve the performance of various image recognition tasks and acquire good generalization. More comprehensively, there is no relevant discussion about the how to determine the optimal number of neurons, that is, how many neurons are needed to complete the optimal coverage of one category. We set the number of HCF neuron models used in our HCFNN network empirically for different tasks, which is well worth studying and requires a large amount of work. This research will be beneficial for the adaptive dynamic network architecture design, and is our future core research.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The data that has been used is confidential.

References

[1] S. He, L. Schomaker, GR-RNN: global-context residual recurrent neural networks for writer identification, Pattern Recognit 117 (2021) 107975.
[2] Z. Lei, S. Yang, A.U. Haq, D. Zhang, A.O. Francis, An entity-weights-based convolutional neural network for large-sale complex knowledge embedding, Pattern Recognit (2022) 108841.
[3] C. Yan, G. Pang, X. Bai, C. Liu, X. Ning, L. Gu, J. Zhou, Beyond triplet loss: person re-identification with fine-grained difference-aware pairwise loss, IEEE Trans Multimedia 24 (2021) 1665–1677.
[4] H. Shi, L. Wang, N. Zheng, G. Hua, W. Tang, Loss functions for pose guided person image generation, Pattern Recognit 122 (2022) 108351.
[5] C. Wang, X. Ning, L. Sun, L. Zhang, W. Li, X. Bai, Learning discriminative features by covering local geometric space for point cloud analysis, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–15.
[6] C. Wang, X. Wang, J. Zhang, L. Zhang, X. Bai, X. Ning, J. Zhou, E. Hancock, Uncertainty estimation for stereo matching based on evidential deep learning, Pattern Recognit 124 (2022) 108498.
[7] G. Wang, W. Li, L. Zhang, L. Sun, P. Chen, L. Yu, X. Ning, Encoder-x: solving unknown coefficients automatically in polynomial fitting by using an autoencoder, IEEE Trans Neural Netw Learn Syst (2021).
[8] Geldmacher, S. David, The neurobiology of learning and memory, in: Journal of Neuro-Ophthalmology, volume 29, 2010, p. 257.
[9] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull Math Biophys 5 (4) (1943) 115–133.
[10] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, J. Wang, Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction, in: IEEE Transactions on Neural Networks and Learning Systems, volume 30, 2019, pp. 601–614.
[11] C. Koch, I. Segev, The role of single neurons in information processing, in: Nature Neuroscience, volume 3, 2000, pp. 1171–1177.
[12] Costa, Rui, One cell to rule them all, and in the dendrites bind them, in: Frontiers in Synaptic Neuroscience, volume 3, 2011, p. 5.
[13] D.S. Broomhead, D. Lowe, Radial basis functions, multi-variable functional interpolation and adaptive networks, Technical Report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.
[14] S.-Q. Zhang, Z.-H. Zhou, Flexible transmitter network, Neural Comput 33 (11) (2021) 2951–2970.
[15] S. Kiranyaz, J. Malik, H.B. Abdallah, T. Ince, A. Iosifidis, M. Gabbouj, Self-organized operational neural networks with generative neurons, Neural Networks 140 (2021) 294–308.
[16] J. Xie, Z. Ma, D. Chang, G. Zhang, J. Guo, Gpca: a probabilistic framework for gaussian process embedded channel attention, IEEE Trans Pattern Anal Mach Intell (2021).
[17] R. Du, J. Xie, Z. Ma, D. Chang, Y.-Z. Song, J. Guo, Progressive learning of category-consistent multi-granularity features for fine-grained visual classification, IEEE Trans Pattern Anal Mach Intell (2021).
[18] L. Zhang, B. Zhang, A geometrical representation of mcculloch-pitts neural model and its applications, IEEE Trans. Neural Networks 10 (4) (1999) 925–929.
[19] W. Shoujue, L. Jiangliang, Geometrical learning, descriptive geometry, and biomimetic pattern recognition, Neurocomputing 67 (2005) 9–28, doi:10.1016/j.neucom.2004.11.034. Geometrical Methods in Neural Networks and Learning
[20] S.-j. Wang, Z.-z. Li, Discussion on the basic mathematical models of neurons in general purpose neurocomputer, Acta Electronica Sinica 29 (5) (2001) 577–580.
[21] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of control, signals and systems 2 (4) (1989) 303–314.
[22] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, Technical Report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
[23] M. Mitchell, An introduction to genetic algorithms, MIT press, 1998.
[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int J Comput Vis 115 (3) (2015) 211–252.
[25] I. Kemelmacher-Shlizerman, S.M. Seitz, D. Miller, E. Brossard, The megaface benchmark: 1 million faces for recognition at scale, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4873–4882.
[26] E. Ristani, F. Solera, R. Zou, R. Cucchiara, C. Tomasi, Performance measures and a data set for multi-target, multi-camera tracking, in: European Conference on Computer Vision, Springer, 2016, pp. 17–35.
[27] J.A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural processing letters 9 (3) (1999) 293–300.
[28] A. Krizhevsky, Learning multiple layers of features from tiny images, Technical Report, 2009.
[29] G.B. Huang, M. Mattar, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database forstudying face recognition in unconstrained environments, Workshop on faces in'Real-Life'Images: detection, alignment, and recognition, 2008.
[30] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable person re-identification: A benchmark, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1116–1124.
[31] W. Li, R. Zhao, T. Xiao, X. Wang, Deepreid: Deep filter pairing neural network for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 152–159.
[32] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[33] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Computer Science (2014).
[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[35] M. Wang, W. Deng, Deep face recognition: a survey, Neurocomputing 429 (2021) 215–244.
[36] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.
[37] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[38] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 499–515.
[39] W. Liu, Y. Wen, B. Raj, R. Singh, A. Weller, Sphereface revived: unifying hyperspherical face recognition, IEEE Trans Pattern Anal Mach Intell (2022).

[40] J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4690–4699.
[41] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, W. Liu, Cosface: Large margin cosine loss for deep face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5265–5274.
[42] S. Li, J. Xu, X. Xu, P. Shen, S. Li, B. Hooi, Spherical confidence learning for face recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15629–15637.
[43] Y. Shi, X. Yu, K. Sohn, M. Chandraker, A.K. Jain, Towards universal representation learning for deep face recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6817–6826.
[44] S. Liao, Z. Lei, D. Yi, S.Z. Li, A benchmark study of large-scale unconstrained face recognition, in: IEEE International Joint Conference on Biometrics, IEEE, 2014, pp. 1–8.
[45] J. Si, H. Zhang, C.-G. Li, J. Kuen, X. Kong, A.C. Kot, G. Wang, Dual attention matching network for context-aware feature sequence based person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5363–5372.
[46] Y. Sun, L. Zheng, Y. Yang, Q. Tian, S. Wang, Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline), in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 480–496.
[47] W. Li, X. Zhu, S. Gong, Harmonious attention network for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2285–2294.
[48] C.-P. Tay, S. Roy, K.-H. Yap, Aanet: Attribute attention network for person re-identifications, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7134–7143.
[49] Z. Dai, M. Chen, X. Gu, S. Zhu, P. Tan, Batch dropblock network for person re-identification and beyond, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3691–3701.
[50] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, J. Gu, A strong baseline and batch normalization neck for deep person re-identification, IEEE Trans Multimedia 22 (10) (2019) 2597–2609.
[51] W.-D. Jin, J. Xu, Q. Han, Y. Zhang, M.-M. Cheng, Cdnet: complementary depth network for rgb-d salient object detection, IEEE Trans. Image Process. 30 (2021) 3376–3390.
[52] B.N. Xia, Y. Gong, Y. Zhang, C. Poellabauer, Second-order non-local attention networks for person re-identification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3760–3769.
[53] Y. Ge, F. Zhu, D. Chen, R. Zhao, et al., Self-paced contrastive learning with hybrid memory for domain adaptive object re-id, Adv Neural Inf Process Syst 33 (2020) 11309–11321.
[54] R. Quan, X. Dong, Y. Wu, L. Zhu, Y. Yang, Auto-reid: Searching for a part-aware convnet for person re-identification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3750–3759.

Xin Ning received the B.S. degree in software engineering in 2012, and the Ph.D. degree in electronic circuit and system from university of Chinese Academy of Sciences, in 2017. He is currently an Associate Professor with the Laboratory of Artificial Neural Networks and High Speed Circuits, Institute of Semiconductors, Chinese Academy of Sciences. He has published by first or corresponding author more than 40 papers in journals and refereed conferences. His current research interests include pattern recognition, computer vision, and image processing.

Weijuan Tian received her M.S. degrees from School of Electronic Engineering in Xidian University in 2017. Currently, she worked as an algorithm research fellow in the Cognitive Computing Technology Joint Laboratory of Wave Group. Her research interests include machine learning, DNNs, neuron modeling, video and image processing.

Weijun Li received his Ph.D. in 2004 from Institute of Semiconductors, Chinese Academy of Sciences. He is currently a Professor of Artificial Intelligence at Institute of Semiconductors Chinese Academy of Sciences (ISCAS) and the University of Chinese Academy of Sciences. He is in charge of the Artificial intelligence research Center of ISCAS, also the Director of the Lab of Highspeed Circuits Neural Networks of ISCAS. His research interests include deep modeling, active 3D imaging systems, 3D object recognition, pattern recognition, artificial neural networks and intelligent system. He is a senior member of IEEE.

Xiao Bai received the B.Eng. degree in computer science from Beihang University of China, Beijing, China, in 2001, and the Ph.D. degree in computer science from the University of York, York, U.K., in 2006. He was a Research Officer (Fellow, Scientist) with the Computer Science Department, University of Bath, until 2008. He is currently a Full Professor with the School of Computer Science and Engineering, Beihang University. He has authored or co-authored more than 100 papers in journals and refereed conferences. His current research interests include pattern recognition, image processing, and remote sensing image analysis. He is the Associate Editor for journal of Pattern Recognition and Signal Processing.

Yuebao Wang received a B.S Degree in mathematics and Applied Mathematics from North University of China in 2013, and received M.S degree in operations research and cybernetics Currently in 2018. He worked as an algorithm research fellow in the Cognitive Computing Technology Joint Laboratory of Wave Group. His current research interests include machine learning, function approximation theory and TDA.