A UniverApproCNN with Universal Approximation and Explicit Training Strategy

Yin Yang, Yifeng Wang(✉), and Senqiao Yang

Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen, Guangdong Province, China

Abstract. Approximation theory has achieved many results in the field of deep learning. However, these results and conclusions can rarely be directly applied to the solution of practical problems or directly guide the training and optimization of deep learning models in the actual context. To address this issue, we construct a CNN structure with universal approximation, which is called UniverApproCNN. It is ensured that the approximation error of such CNN is bounded by an explicit approximation upper bound that relies on the hyper parameters of this model. Moreover, a general case of multidimensional is considered by generalizing the conclusion of the universality property of CNN. A practical problem in the field of inertial guidance is used as a background to conduct experiments, so that the theory can give an explicit training strategy and break the barrier between the theory and its application. We use the curve similarity index defined by Fréchet distance to prove that the experimental results are highly consistent with the functional relationship given by the theory. On this basis, we define the 'approximation coefficient' of UniverApproCNN, which can give the stop time of model training and related training strategies. Specifically, taking the operation of normalization as a widely used technique into consideration, we then show that this operation does not take effect on the approximation performance of the CNN and UniverApproCNN.

Keywords: Approximation theory · Universality of CNN · Normalization · Inertial guidance · Fréchet distance

1 Introduction

In the 1980s, the neural network model received people's attention again because of the successful applications such as on signal processing and control. The dense problem naturally arises from these applications [1], that is, for such a set

$$\mathcal{M}(\sigma) = span\{\sigma(\mathbf{w} \cdot \mathbf{x} - \theta) : \theta \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^n\},$$

Supported by special fund for scientific and technological innovation strategy of Guangdong Province.

Y. Yang, Y. Wang and S. Yang—These authors contributed equally.

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 407, pp. 297–315, 2021.
https://doi.org/10.1007/978-3-030-92638-0_18

what properties does the activation function need to make it dense in the corresponding function space? This denseness is considered to be fundamental important in neural network theory as the t heoretical ability of approximation [2,3]. For the single hidden layer perceptron model

$$\mathcal{M}_r(\sigma) = \left\{ \sum_{i=1}^{r} c_i \sigma(\mathbf{w}^i \mathbf{x} - \theta_i) : c_i, \theta_i \in \mathbb{R}, \mathbf{w}^i \in \mathbb{R}^n \right\},$$

which is one of the more simpler neural networks models mathematically, Pinkus [2] has made a detailed summary, including the approximation bound with different activation functions and under different norms [4–8]. For networks with a certain depth and special connection structure with sparsity

$$h^{(j)}(x) = \left(\sigma\left(w_i^{(j)} h^{(j-1)}(x) - b_i^{(j)} \right) \right)_{i=1}^{d_j},$$

compared with their success in practical applications, there is relatively little research on their theoretical approximation capabilities [9], although there are some theoretical interpretations for superior behavior due to their depth and special connection structure [10–13]. Recently, Zhou Dingxuan proved that the deep convolutional neural network is universally approximated in the continuous function space, and given the upper bound of the approximation under certain conditions [14].

Following Zhou Dingxuan's work, we propose the construction of a generic convolutional neural network with explicit approximation bound, which is called the UniverApproCNN model. Based on the inertial guidance data, an experimental design was made to verify the approximation bound, as well as some necessary theoretical expansions. Since Zhou's proof method is constructive, we find it is easy to prove that the approximation performance of the network with layer normalization operation [15,16] is unchanged using his method.

In the second chapter, some preliminary knowledge is given. In the third chapter, we will conduct in-depth analysis and model design of the CNN with known approximation bound introduced in Zhou Dingxuan's article. Based on the data of inertial guidance, we need to prove the universality of approximation and model design of CNN suitable for two-dimensional input, these works will be presented in Sect. 4. At the same time, we construct UniverApproCNN with universal approximation, and design an object motion tracking problem based on inertial guidance theory to further explore and analyze the performance of UniverApproCNN in Sect. 5. In Sect. 6, we use the interpretability advantage of UniverApproCNN to guide the model design and training strategy by defining "approximation coefficient".

2 Preliminaries

Before introducing the main work of this article, we first give the symbols that will be used in the article and some results in the approximation theory.

Consider the activation function defined as $\sigma(u) = max\{u, 0\}, u \in \mathbb{R}$, called ReLU (rectified linear unit), and a sequence of convolutional filter masks$\{w^{(j)}\}_j$ with finite support, that is, $w_k^{(j)} \neq 0$ if and only if $0 \leq k \leq s$ Convolution of two sequences can be written as the product of a matrix and a vector. The convolution kernel represented by the mask slides on the input of length D($s \leq D$), that is, the process of convolution, which can induce the following Toeplitz type matrix T of size $(D + s) \times D$:

$$T = \begin{pmatrix} w_0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ w_1 & w_0 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \cdots & \cdots & \vdots \\ w_s & w_{s-1} & \cdots & w_0 & 0 & \cdots & 0 \\ 0 & w_s & w_{s-1} & \cdots & w_0 & 0 \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & w_s & w_{s-1} & \cdots & w_0 \\ 0 & \cdots & \cdots & 0 & w_s & w_{s-1} & w_1 \\ 0 & \cdots & \cdots & \cdots & 0 & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & w_s \end{pmatrix}$$

Definition 1. *Consider the input vector $x \in \mathbb{R}^d$, a convolutional neural network with depth J and width $d_j = d+js$ can be induced by a sequence $\{w^{(j)}\}_{j=1}^J$, whose hidden layer can be defined a vector-valued function column*

$$\{h^{(j)}(x)\}_{j=1}^J,$$

generated by iteration:

$$h^{(0)}(x) = x,$$

$$h^{(j)}(x) = \sigma(T^{w^{(j)}} h^{(j-1)}(x) - b^{(j)}), \quad j = 1, 2, \cdots, J.$$

where $T^{w^{(j)}} = (w_{i-k}^{(j)})_{d_j \times d_{j-1}}$ is the Toeplitz type convolutional matrix induced by $w^{(j)}$, σ acts on vectors by component, and $b^{(j)}$ is the bias vector.

Definition 2. *The hypothesis space of the convolutional neural network model defined above is the following function set:*

$$\mathcal{H}_J^{\mathbf{w},\mathbf{b}} = \left\{ \sum_{k=1}^{d_J} c_k h_k^{(J)}(x) : c \in \mathbb{R}^{d_J} \right\},$$

where $\mathbf{w} = \{w^{(j)}\}_{j=1}^J, \quad \mathbf{b} = \{b^{(j)}\}_{j=1}^J$.

Zhou proved the universal approximation of the network to the continuous function space $C(\Omega)$:

Theorem 1. *Let $2 \leq s \leq d$ and $\Omega \subseteq [-1,1]^d$. If $J \geq 2d/(s-1)$ and $f = F|_\Omega$, with $F \in H^r(\mathbb{R}^d)$ and an integer index $r > 2 + d/2$, then there exist $\mathbf{w}$, $\mathbf{b}$ and $f_J^{\mathbf{w},\mathbf{b}} \in \mathcal{H}_J^{\mathbf{w},\mathbf{b}}$ such that*

$$\|f - f_J^{\mathbf{w},\mathbf{b}}\|_{C(\Omega)} \leq c\|F\|\sqrt{log\, J}(1/J)^{\frac{1}{2}+\frac{1}{d}},$$

where c is an absolute constant and $\|F\|$ denotes the Soblev norm of $F \in H^r(\mathbb{R}^d)$.

The theorem above is based on a result on ridge approximation to $F|_{[-1,1]^d}$ shown as follows:

Theorem 2. *Let $D = [-1,1]^d$. Suppose f admits a Fourier representation $f(x) = \int_{\mathbb{R}^d} e^{ix\cdot\omega}\tilde{f}(\omega)\mathrm{d}\omega$ and*

$$v_{f,2} = \int_{\mathbb{R}^d} \|\omega\|_1^2 |\tilde{f}(\omega)|\mathrm{d}\omega < \infty.$$

There exists a linear combination of ramp ridge functions of the form

$$f_m(x) = \beta_0 + x\cdot\alpha_0 + \frac{v}{m}\sum_{k=1}^{m}\beta_k(x\cdot\alpha_k - t_k)_+$$

with $\beta \in [-1,1]$, $\|\alpha_k\|_1 = 1$, $0 \leq t_k \leq 1$, $\beta_0 = f(0)$, $\alpha_0 = \nabla f(0)$, and $v \leq 2v_{f,2}$ such that

$$\|f - f_m\|_{C(\Omega)} \leq cv_{f,2}max\{\sqrt{log\, m}, \sqrt{d}\}m^{-1/2-1/d}.$$

3 A CNN Structure with Universal Approximation: UniverApproCNN

3.1 Model Design

In the proof of Zhou's article, they tried to decompose the coefficients $W = [W_{(m+1)d-1}\cdots W_1\ W_0] = [\alpha_m^T\cdots\alpha_1^T\ \alpha_0^T]$ in the ramp ridge function into the weight parameters of each layer. With the help of the generating function of a sequence, they decompose W into the convolution of multiple sequences $W = w^{(J)} * \cdots * w^{(1)}$, that is, decompose $\widetilde{W}$ into $\widetilde{W}(z) = \widetilde{w}^{(J)}(z)\widetilde{w}^{(J-1)}(z)\cdots\widetilde{w}^{(1)}(z)$. Note that $\widetilde{W}(z)$ can be factored into

$$\widetilde{W}(z) = W_M\prod_{k=1}^{K}[z^2 - 2x_k z + (x_k^2 + y_k^2)]\prod_{k=2K+1}^{M}(x - z_k),$$

where $M = (m+1)d - 1$. Next, we decide the hyper parameters of the network according to the above decomposition formula.

Zhou's decomposition emphasizes the existence of $\{s_j\}_{j=1}^J$, where $s_j \leq s$, satisfying $deg(\widetilde{w}^{(j)}(z)) = s_j$ and $s_1 + s_2 + \cdots + s_J = M$. Since K is unknown, the size of s_j is ambiguous. In order to determine the number of nodes in each

hidden layer, we use an even-numbered mask length s_j. Further, in order to make an explicit relationship between depth J and M, we use a fixed $s_j = s$, then J, s and M satisfy the following relationship:

$$M = (J-1)s + t,$$

that is,

$$(m+1)d - 1 = (J-1)s + t, \quad J, t \in \mathbb{N}^*, 0 < t \leq s,$$

where

$$deg(\widetilde{w}^{(J)}(z)) = t, deg(\widetilde{w}^{(j)}(z)) = s, j = 1, \cdots, J-1,$$

the corresponding sequences are

$$w^{(J)} = \{w_0^{(J)}, \cdots, w_t^{(J)}\}, \quad w^{(j)} = \{w_0^{(j)}, \cdots, w_s^{(j)}\}$$

satisfying $W = w^{(J)} * \cdots * w^{(1)}$. Let

$$d_j = d + (j-1)s, \quad j = 1, \cdots, J-1,$$

$$d_J = d_{J-1} + t,$$

then $T^{w^{(j)}} \in \mathbb{R}^{d_j \times d_{j-1}}, j = 1, \cdots, J$, $T^{w^{(j)}}$ represents the Toeplitz type matrix generated by the sequence $w^{(j)}$.

Now we have $T^W = T^{w^{(J)}} \cdots T^{w^{(1)}}$. Let $w^{(J+1)} = \{w_0^{(J+1)}\} = \{0\}$, the Toeplitz matrix induced by $w^{(J+1)}$ is identity matrix, that is, $T^{w^{(J+1)}}_{d_{J+1} \times d_J} = I$, where $d_{J+1} = d_J$, the corresponding physical meaning is not to do zero padding to the output of the J-th hidden layer.

In order to get the desired term, we construct the bias as follows. Denote

$$\|w\|_1 = \sum_{k=-\infty}^{\infty} |w_k|, \quad B^{(0)} = max_{x \in \Omega} max_{k=1,\cdots,d} |x_k|,$$

$$B^{(j)} = \|w^{(j)}\|_1 \cdots \|w^{(1)}\|_1 B^{(0)}, j \geq 1.$$

If nonlinear activation is considered only in the J-th layer, let

$$b_l^{(J)} = \begin{cases} -B^{(J)}, & l = d \\ t_k, & l = (k+1)d, 1 \leq k \leq m \\ B^{(J)}, & \text{otherwise} \end{cases},$$

then

$$\begin{aligned} h_l^{(J)} &= \sigma\Big((T^W x)_l - b_l^{(J)}\Big) \\ &= \begin{cases} \alpha_0 \cdot x + B^{(J)}, & l = d \\ (\alpha_k \cdot x - t_k)_+, & l = (k+1)d, 1 \leq k \leq m \\ 0, & \text{otherwise} \end{cases}. \end{aligned}$$

Let

$$b_l^{(J+1)} = \begin{cases} -B^{(J)}, l = 1 \\ 0, \quad \text{otherwise} \end{cases},$$

then $h^{(J+1)}(x) = \sigma\Big(T^{w^{(J+1)}} h^{(J)}(x) - b^{(J+1)}\Big)$, expressed by component as

$$h_l^{(J+1)} = \begin{cases} \alpha_0 \cdot x + B^{(J)}, & l = d \\ (\alpha_k \cdot x - t_k)_+, & l = (k+1)d, 1 \le k \le m \\ B^{(J)}, & l = 1 \\ 0, & \text{otherwise} \end{cases}.$$

For the above $\mathbf{w}, \mathbf{b}$, we can take

$$f_{J+1}^{\mathbf{w},\mathbf{b}} = F_m|_\Omega \in span\{h_k^{J+1}(x)\}_{k=1}^{d_{J+1}},$$

satisfying

$$\begin{aligned} \|f - f_J^{\mathbf{w},\mathbf{b}}\|_{C(\Omega)} &\le \|F - F_m\|_{C([-1,1]^d)} \\ &\le c_0 v_{F,2} max\{\sqrt{\log m}, \sqrt{d}\} m^{-1/2-1/d}. \end{aligned}$$

Remark 1. When $t = 0$, we only get $J - 1$ sequences from the decomposition, and then let $w^{(J)} = \{W_0^{(J)}\} = \{1\}$ and assign the value of $b^{(J)}$ to $b^{(J-1)}$, $b^{(J+1)}$ to $b^{(J)}$.

It is worth noting that the UniverApproCNN model not only has strong interpretability, but its backpropagation process is compatible with almost all optimization algorithms. This advantage brings a broad application prospect for UniverApproCNN. The back propagation process is shown in the appendix.

3.2 UniverApproCNN for Multiple Outputs

The method in the Theorem 1 can be extended to multi-dimensional output, that is, the UniverApproCNN also has universal approximation to the vector-valued function space. For a l-dimensional continuous vector-valued function $f(x) = \big(f_1(x), \cdots, f_l(x)\big)$, there exist ramp ridge functions

$$F_m^i = \beta_0^i + \alpha_0^i x + \frac{v^i}{m} \sum_{k=1}^{m} \beta_k^i (\alpha_k^i x - t_k^i)_+, \quad i = 1, \cdots, l,$$

such that

$$\|f_i - F_m^i\|_{C(\Omega)} \le c_0^i v_{F^1,2} max\{\sqrt{\log m}, \sqrt{d}\}^{-\frac{1}{2}-\frac{1}{d}}.$$

We stack all the coefficients $\alpha_k^i, i = 1, \cdots, l, k = 0, \cdots, m$ into a row vector, let

$$W^i = \big(W_{(m+1)d-1}^i, \cdots, W_0^i\big) = \big((\alpha_m^i)^T, \cdots, (\alpha_0^i)^T\big)$$

$$W = \big(W_{l(m+1)d-1}, \cdots, W_0\big) = \big(W^l, \cdots, W^1\big),$$

it is easy to find by definition that

$$T^W_{(i-1)(m+1)d+(k+1)d;:} = (\alpha^i_k)^T,$$

which means the term $(\alpha^i_k)^T$ can be represented by the node with order $(i-1)(m+1)d+(k+1)d$ of the last convolutional layer. Using the decomposition strategy mentioned in the first section in this chapter, we can decompose W into the weight parameters of the CNN with depth $J+1$, satisfying

$$l(m+1)d-1 = (J-1)s+t, \quad J,t \in \mathbb{N}^*, 0 < t \leq s.$$

3.3 Normalized CNN and UniverApproCNN

In this section, we try to show that the approximation performance of CNN and UniverApproCNN with batch normalization operation will not be weakened at least. Because UniverApproCNN is a special form of CNN, the universal approximation condition of normalized CNN is more stringent, and its establishment can guarantee the universal approximation of normalized UniverApproCNN. Therefore, we verify the universal approximation of the normalized CNN.

Consider N input vectors of length d, and define $a^{k,j} = T^{w^{(j)}} \cdot h^{(j-1)}(x_k), j = 1, 2, \cdots, J$ as the summed input of the j-th hidden layer of the k-th input vector. The batch normalization method normalizes the summed input of each unit in the same layer, that is, scales the summed input according to the batch normalization statistics defined below:

$$\mu = \frac{1}{Nd_j}\sum_{k=1}^{N}\sum_{i=1}^{d_j} a_i^{k,j}, \quad \sigma_j = \sqrt{\frac{1}{Nd_j}\sum_{k=1}^{N}\sum_{i=1}^{d_j}(a_i^{k,j}-\mu_j)^2},$$

then, the input of the hidden layer after batch normalization can be defined as

$$\overline{a_i^{k,j}} = \frac{g_i^j}{\sigma_j}\cdot(a_i^{k,j}-\mu_j),$$

where $g_i^j, j = 1, 2, \cdots, J, i = 1, 2, \cdots, d_j$ are gain parameters, further, the output of the hidden layer can be defined as

$$\overline{h^j} = \sigma(\overline{a^{k,j}} - b^{(j)}) = \sigma\left(\frac{g^j}{\sigma_j} \odot (a^{k,j} - \mu_j \mathbf{1}_{d_j}) - b^{(j)}\right),$$

where $\odot$ means product by component between two vectors, $\mathbf{1}_{d_j} = (1, 1, \cdots, 1)^T$ with length d_j, $b^{(j)}$ is the bias parameter.

Only consider batch normalization operations on the first layer, when $g_i^1 : \Omega \to \mathbb{R}, i = 1, 2, \cdots, d_1$ are continuous, consider the influence of the activation function σ, since there exists a similar bound for the normalized summed input, that is, for any $x \in \Omega$, we have

$$\left|\frac{g_i^1}{\sqrt{\frac{1}{Nd_j}\sum_{k=1}^{N}\sum_{i=1}^{d_j}(a_i^{k,1}-\mu_j)^2}}\cdot(a_i^{k,1}-\mu_1)\right|^2 \leq MNd_1,$$

Therefore, by reconstructing the bias parameters $\beta^{(1)}$ of the first hidden layer, the weight parameter $T^{(2)}$ and bias parameter $\beta^{(2)}$ of the second hidden layer, the regularized network output $f_J^{\mathbf{w}',\mathbf{b}'}$ can be restored to the output $f_J^{\mathbf{w},\mathbf{b}}$ of the original network, where $\mathbf{w}' = \{w^{(1)}, T^{(2)}, w^{(3)}, \cdots, w^{(J)}\}, \mathbf{b}' = \{\beta^{(1)}, \beta^{(2)}, b^{(3)}, \cdots b^{(J)}\}$, which means the approximation performance of the normalized CNN does not changed compared with the original network.

4 UniverApproCNN for Two-Dimensional Input

4.1 Proof of the Approximation of the CNN Suitable for Two-dimensional Input

Consider a two-dimensional input I of size $d \times n$, the size of the first layer of convolution kernel is $d \times (s+1)$, with s column zero padding on both sides of the input, the first layer convolution is regarded as a matrix form, which is expressed as follows:

First, expand the input matrix by rows, namely

$$x\big((i-1)n+k\big) = I(i,k), 1 \le i \le d, 1 \le k \le n,$$

Denote the output of the first layer as

$$h^{(1)}(x) = \sigma\big(T^{(1)}h^{(0)}(x) - b^{(1)}\big),$$

where $h^{(0)} = x$, $T^{(1)} = \big(T^{w^{(1),1}}, \cdots, T^{w^{(1),d}}\big)$ is a matrix of size $n_1 \times dn$, every block of which is a Toeplitz matrix generated by the corresponding row vector of the first convolution kernel.

Next we can define a CNN with depth J generated by a sequence of finitely supported convolutional filter masks $\{w^{(j)}\}_{j=1}^{J}$ as a sequence of vector-valued functions $\{h^{(j)}\}_{j=1}^{J}$, and $h^{(j)}(x) = \sigma\big(T^{(j)}h^{(j-1)}(x) - b^{(j)}\big), j = 1, 2, \cdots, J$, where $T^{w^{(j)}} = (w_{i-k}^{(j)})$ is a Toeplitz type matrix of size $n_j \times n_{j-1}$. The hypothesis space of such network can be expressed as

$$\mathcal{H}_J^{\mathbf{w},\mathbf{b}} = \left\{ \sum_{k=1}^{d_J} c_k h_k^{(J)}(x) : c \in \mathbb{R}^{d_J} \right\}.$$

The following theorem about universality of the CNN above is the extension of Zhou's work. We give the proof in the appendix.

Theorem 3. *For any compact subset Ω of $[-1,1]^{nd}$ and any $f \in C(\Omega)$, there exist $\mathbf{w}, \mathbf{b}$ and $f_{J+1}^{\mathbf{w},\mathbf{b}}$ such that*

$$\|f - f_{J+1}^{\mathbf{w},\mathbf{b}}\|_{C(\Omega)} \le c_0 v_{F,2} max\{\sqrt{log\, m}, \sqrt{nd}\} m^{-\frac{1}{2}-\frac{1}{nd}}.$$

4.2 Model Design

Since the objective function is unknown, we cannot find non-trivial common factors. We only verify the model with the following settings:

- For a two-dimensional input vector $I \in \Omega, \Omega \subseteq [-1, 1]^{d \times n}$, consider two layers of convolution.
- The size of the convolutional kernel of the first layer is $d \times (m+1)n$, perform $(m+n)n-1$ columns of zero padding on both sides of the input, set the stride equal to 1.
- In the first hidden layer, first reduce the bias and then activate by the non-linear function, where

$$b_l^{(1)} = \begin{cases} -B^{(1)}, & l = n \\ t_k, & l = (k+1)n, 1 \le k \le m \\ B^{(1)}, & \text{otherwise} \end{cases},$$

  and $B^{(1)} = \|w^{(1),1}\|_1 + \cdots + \|w^{(1),d}\|_1$.
- The convolutional filter mask of the second is $w^{(2)} = \{w_0^{(2)}\} = \{1\}$, then reduce the bias and activate, where

$$b_l^{(2)} = \begin{cases} -B^{(1)}, & l = 1 \\ 0, & \text{otherwise} \end{cases}.$$

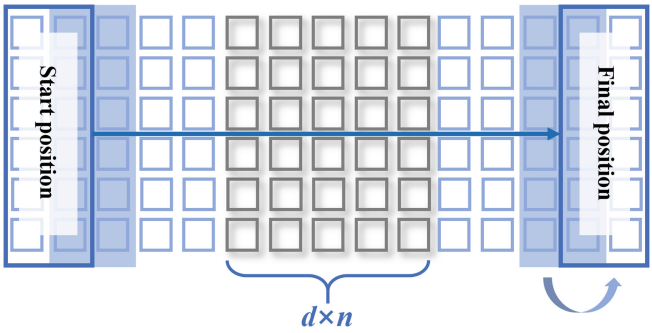

Fig. 1. The structure of UniverApproCNN model.

Figure 1 shows the structure of UniverApproCNN model and the sliding mode of convolution kernel. Since the structure of the model completely conforms to the universality theorem, under the condition of sufficient training, theoretically, when the hyper parameters of the model change, in addition to the structure of the model itself, its corresponding approximation upper bound also changes accordingly, the corresponding relationship between these two changes can guide us to get the ideal training strategy in the experiment.

5 Performance Experiment of UniverApproCNN for Inertial Guidance

In order to verify the relationship between the approximation bound of UniverApproCNN and related parameters, we designed an experiment oriented to inertial guidance theory. We will use the UniverApproCNN model to approximate a complex model in this theory. The model is composed of many highly nonlinear functions, involving coordinate system transformation, normalization, multiple integration, cosine matrix calculation, quaternion calculation, and other operations. The complex coupling relationship of these operations is very suitable for testing the approximation effect of the UniverApproCNN model. Moreover, since our verification is based on actual problems, the experimental process can be directly guided by theory, and the experimental results can directly echo the theory. Therefore, our experiment actually breaks through the barrier between theory and application, which is of great help to the development and application of the theory.

Our experimental object is a trajectory restoration model based on inertial sensors. The input is six-axis inertial sensor data: three-axis acceleration, three-axis angular velocity, and the output is the spatial position of the object's movement. Regarding the model as a mapping from input to output, the mapping relationship f is the object we want to approximate with UniverApproCNN. The form of f is closely related to the way of movement of the object. We define the form of movement as follows:

Suppose an object with a mass of m makes horizontal projectile motion in three-dimensional space at an initial velocity v_0, and its direction is the positive direction of the x-axis; the acceleration of gravity is $g = 9.8\,\mathrm{m/s^2}$, whose direction is the negative direction of the z-axis; The air resistance is $F = kv^2$, where v is the speed of the object, and k is the coefficient of air resistance; at the same time, the object is in the state of rotation during the motion, and the rotation angular velocity around the x, y, and z axes at the initial time of the horizontal throwing is ω_x, ω_y, ω_z, its attenuation law is $\omega_{t_1} = \omega_{t_0} e^{-Kt}$, where K is the angular velocity attenuation coefficient, and t is the interval time between t_0 and t_1.

After the movement mode is determined, the mapping relationship f between the input data and the output data is determined accordingly. In this horizontal throwing movement, since the object is always in a weightless state, inertial sensors can only measure the acceleration of an object due to air resistance from all directions. The acceleration and angular velocity measured by the inertial sensor are all based on its own coordinate system (Object Coordinate System), but the calculation of its trajectory is based on the spatial absolute coordinate system (World Coordinate System). At the same time, the spatial posture of the object changes all the time, so the transformation relationship between the two coordinate systems is constantly changing. In addition, the effect of the decomposition of the gravitational acceleration in the object coordinate system must also be considered. The entire complex system is included in the mapping relationship f, so it is difficult to train UniverApproCNN to approximate it.

At this time, we need to generate a data set based on a large number of variable parameters during the movement for training, verification and testing of the UniverApproCNN model. In the motion process we define, the variable parameters include: mass m, initial horizontal throwing velocity v_0, air resistance coefficient k, initial angular velocity of the object rotation ω_x, ω_y, ω_z, and rotation attenuation coefficient K. We use a grid method to set 2 specific values for the parameter $\theta : \pi/3, \pi/6$, and 10 values for each of the remaining 7 parameters. A total of 2×10^7 sets of parameter combinations are generated for 8 parameters. Combining each set of parameters with f can determine the specific motion details, such as the acceleration, angular velocity at each moment, and the spatial coordinates of the object at each moment. For each set of parameters, we start from the initial moment of motion, generate 100 frames of object motion acceleration and angular velocity information as inertial data, that is, the input data of the UniverApproCNN model, and generate the position information of the 101st frame of object motion as the data to be predicted, which is the output data of UniverApproCNN. In the end, a total of 2×10^7 training samples were generated. We randomly divide 2×10^7 training samples into training set, validation set, and test set according to the ratio of $98 : 1 : 1$. In order to enhance the credibility of the experimental results, we independently predict the motion trajectories of objects in different directions, that is, use the parallel experimental method to predict the motion trajectories of x, y, and z axes. The final experimental result of x axis is shown as Fig. 2, and the results of y, z axes are shown in the appendix. Figure 4 and Fig. 5:

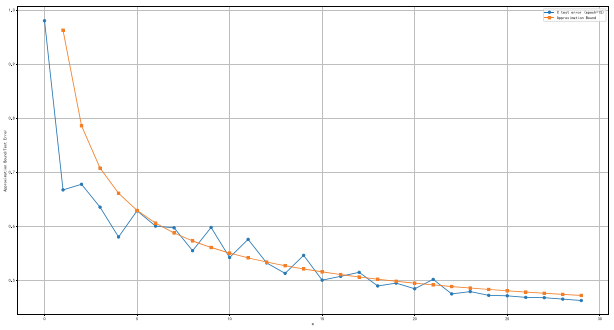

Fig. 2. The experiment on x-axis.

It can be seen that the experimental results are highly consistent with the functional relationship given by the theory. In order to measure the degree of agreement, we use Fréchet distance to define the similarity index between the experimental result curve and the theoretical curve: assuming that the experimental result curve and the theoretical curve can be respectively expressed as a sequence: $P = [u_1, u_2, u_3, \cdots, u_p]$ and $Q = [v_1, v_2, v_3, \cdots, v_q]$, we can obtain the following sequence pairs L:

$$(u_{a_1}, v_{b_1}), (u_{a_2}, v_{b_2}), \cdots, (u_{a_m}, v_{b_m})$$

where $a_1 = 1, b_1 = 1$ and $a_m = p, b_m = q$ for any $i = 1, 2, \cdots, q$, let $a_{i+1} = a_i$ or $a_{i+1} = a_i + 1$ and $b_{i+1} = b_i$ or $b_{i+1} = b_i + 1$. For any point $m = (x_m, y_m, z_m)$ and $n = (x_n, y_n, z_n)$, the Euclidean distance between the two points can be defined as

$$D(m, n) = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + (z_m - z_n)^2}.$$

The length $\|L\|$ between two sequences P and Q is defined as the maximum Euclidean distance in each sequence pair. The expression is as follows:

$$\|L\| = \max_{i=1,2,\cdots,m} D(u_{a_i}, v_{a_i}).$$

Then the Fréchet distance between sequence P and Q is defined as

$$Fd(P, Q) = min\|L\|.$$

Calculate the module of P and Q as $R_P = \sum_{i=1}^{p-1} D(u_i, u_{i+1})$ and $R_Q = \sum_{i=1}^{q-1} D(v_i, v_{i+1})$ respectively. Then we define the similarity between the P and Q curves as

$$Accuracy = max\left\{0, 100 - \frac{100 \times Fd(P, Q)}{\sqrt{2}(R_P + R_Q)}\right\}.$$

After checking the similarity of the result curves of the three experiments, the agreement between the experimental results and the theoretical values reached 96.45%, 95.76%, and 98.31% respectively, which completed the theoretical verification.

Only when the amount of data is sufficient, the data types are rich, and the training strategy is appropriate, the trained model can meet the upper bound of the error given above. This puts high demands on our training process. In addition, as the model parameter m (this m is not the mass, pay attention to distinguish) changes, the scale and performance of the whole model are changing accordingly, so the applicable training strategy will also continue to change, as shown in Fig. 3.

It can be seen that when m is small, the model converges slowly. At this time, we need more iterations. However, as m increases, the model converges faster and faster. At this time, if the number of iterations is not appropriately reduced, there will be overfitting.

The change of m causes the number of training iterations (epoch) to also change instantly. Although this brings inconvenience to the experiment, this feature also reveals the relationship between m and many hyper parameters in the training strategy such as epoch, batchsize, and learning rate. If we can analyze the relationship through some methods, then we can formulate a rigorous and effective training strategy for the hyper parameter adjustment of the model, so we designed follow-up related experiments.

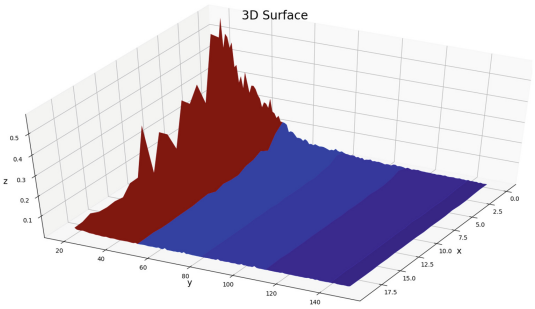

Fig. 3. Convergence rate of model cahanging with parameter m during training.

6 Approximation Coefficients of UniverApproCNN

Although deep learning model has achieved very good results in many fields, it lacks relevant explanatory power, which is also one of its most criticized defects. As a function, the essence of deep learning model is to approximate the function behind the real problem. The approximation theory ensures its approximation effect, and demonstrates the approximation potential of specific deep learning model from the perspective of function space.

However, although the approximation theory gives the approximation potential of deep learning model, the research on the approximation potential can only play a certain effect in theory. Although the research results in this aspect can give a theoretical upper bound of the error of deep learning model, there are two major problems in the application of the upper bound:

1. The upper bound of the error is only theoretical, that is, the default deep learning model has been trained to the optimal state. However, in fact, due to the data quantity, data richness, data error [17], optimization theory and other related limitations, deep learning model is almost impossible to achieve its theoretical maximum approximation potential, especially for some complex problems, as well as the corresponding more complex deep learning model. The complex network structure, complex combination of hyper parameters, the limitation of computing resources and the possible over fitting phenomenon all make it difficult for deep learning model to reach the limit of its approximation potential.
2. The theory can only give the upper bound of the approximation error of the model. In fact, the upper bound is much larger than the error of the model in the training set, verification set and test set, and they are not even in the same order of magnitude. Therefore, the theory can not play a guiding role in our modeling and training process.

These two problems make the gap between approximation theory and deep learning model application appear. In order to fill the gap between theory and

application, and make the relevant conclusions obtained in the study of approximation theory can be used to guide the design and training process of deep learning model, we propose the concept of "approximation coefficient" based on the experimental results.

The UniverApproCNN constructed by us can give the upper bound of the approximation error of the model under specific structure and specific problem, that is, the approximation rate. As like as two peas, we found that the error patterns of different structural models on test set are very similar to the approximation rate, and the trend between the two is almost the same. Therefore, we multiply the error upper bound given by the UniverApproCNN model by a coefficient, which is named "approximation coefficient". When the coefficient is in a certain range, the two curves almost coincide. In order to measure the similarity between the error upper bound curve and the test error curve, we use the curve similarity measurement index defined by Fréchet distance i.e. the similarity between P and Q curves

$$Accuracy = max\left\{0, 100 - \frac{100 \times Fd(P,Q)}{\sqrt{2}(R_P + R_Q)}\right\}.$$

This index can help us to find the value of approximation coefficient under different problem background and different model structure.

As a matter of fact, for a fixed problem and a fixed data set, we will choose to train several times on the model with a relatively simple structure, and explore a relatively reasonable training strategy for the simple structure. Under the condition of sufficient training, the trained model is tested on the test set, and some test errors are obtained. Through these simple structure test errors, combined with the curve similarity measure we defined, we can determine the value range of the approximation coefficient, and then predict the possible errors of the complex structure deep learning model in the test set.

At this time, we train the deep learning model with complex structure, and constantly test it on the test set during the training process, but the error calculation results on the test set do not participate in the process of back propagation and parameter adjustment. This operation can be regarded as a "monitor" in the process of model training. Since we have obtained the approximate error of the model in the test set in advance, if the test error of the model is always significantly larger than the error range predicted by us during training, it means that the training is not sufficient, and we need to increase the number of iterations, on the contrary, it shows that the training effect has reached a good level, and the training should be stopped at this time, otherwise the situation of over fitting may appear.

To sum up, we build a bridge between approximation theory and model design, model implementation, model training by defining "approximation coefficient", so that the relevant conclusions of approximation theory can be applied to specific experiments. In the future, it can also be extended to more research on models and approximation theory, so it has high application value.

7 Conclusion

This paper proposed a CNN model supported by related approximation theory: UniverApproCNN. We successfully applied it to the object motion tracking problem in inertial guidance theory, and successfully verified the relevant conclusions in the approximation theory. In the process of verifying the relationship between the approximation bound and the network structure parameters, we discovered the relationship between the training strategy and the network structure, and we successfully used this relationship to guide the training of the model. As a deep learning network, although it has sacrificed performance, it has made significant progress in interpretability, and this interpretability can guide the formulation of model training strategies very well. In the future, we can use this as a basis to optimize performance to enhance its interpretability while ensuring performance, or use this solution as a basis to find a balance between the performance and interpretability of more deep learning models.

A Appendix

A.1 Proof of the Theorem 3

Proof. Because of the density of $H^r(\Omega)$ in $C(\Omega)$ when $r \geq \frac{nd}{2} + 2$, without loss of generality, we assume $f \in H^r(\Omega)$. When the activation function $\sigma(u) = max\{u, 0\}, u \in \mathbb{R}$ is not considered, the input-related coefficients in the network are all included in the convolution matrix

$$\left(T^{w^{(J)}} \cdots T^{w^{(2)}} T^{w^{(1),1}}, \cdots\cdots, T^{w^{(J)}} \cdots T^{w^{(2)}} T^{w^{(1),d}}\right),$$

given the following notation:

$$\left(T^{w^{(J)}} \cdots T^{w^{(2)}} T^{w^{(1),1}}, \cdots\cdots, T^{w^{(J)}} \cdots T^{w^{(2)}} T^{w^{(1),d}}\right)$$

$$\triangleq \left(T^{W^1}, \cdots\cdots, T^{W^d}\right) \triangleq T$$

where $W^i = w^{(J)} * \cdots * w^{(2)} * w^{(1),i}$ and T^{W^i} is the Toeplitz type matrix induced by W^i.

Note that in the ramp ridge function, the coefficients associated with the first sub-block $x(1:n)$ of length n of input x are

$$\{\alpha_m^1, \cdots, \alpha_m^n, \cdots\cdots, \alpha_1^1, \cdots, \alpha_1^n, \alpha_0^1, \cdots, \alpha_0^n\},$$

the coefficients related to $x(1:n)$ in the network are included in T^{W^1}.

Let $\left(W^1_{(m+1)n-1}, \cdots, W_1^1, W_0^1\right)$

$$= (\alpha_m^1, \cdots, \alpha_m^n, \cdots\cdots, \alpha_1^1, \cdots, \alpha_1^n, \alpha_0^1, \cdots, \alpha_0^n),$$

We have

$$T^{W^1}_{(k+1)n;:} = \big(\alpha(1:n)\big)^T.$$

Similarly, let $\left(W^i_{(m+1)n-1}, \cdots, W^i_1, W^i_0\right)$

$$= \left(\alpha_m^{(i-1)n+1}, \cdots, \alpha_m^{in}, \cdots\cdots, \alpha_0^{(i-1)n+1}, \cdots, \alpha_0^{in}\right),$$

we can get

$$T^{W^i}_{(k+1)n;:} = \left(\alpha_k\big((i-1)n+1 : in\big)\right)^T$$

and

$$T_{(k+1)n;:} = \left(T^{W^1}_{(k+1)n;:}, \cdots, T^{W^d}_{(k+1)n;:}\right) = (\alpha_k)^T,$$
$$k = 0, 1, 2, \cdots, m.$$

Since $W^i = w^{(J)} * \cdots w^{(2)} * w^{(1),i}$, using the generating function of a sequence and the basic theorem of algebra, we can decompose $W^{(i)}$ into a finitely supported sequence $\{w^{(j)}\}_{j=1}^J$. So far, we have decomposed the coefficients $\{\alpha_0, \alpha_1, \cdots, \alpha_m\}$ in the ramp ridge function $F_m(x)$ into the weight parameters of each layer of this particular network.

Note that when we decompose

$$W^i = w^{(J)} * \cdots w^{(2)} * w^{(1),i}, i = 1, 2, \cdots, d$$

a common factor of $\widetilde{w^{(J)}}(z) \cdots \widetilde{w^{(2)}}(z)$ is required. When a untrivial common factor exists, we can decompose it into a sequence $\{w^{(j)}\}_{j=2}^J$ with finite support.

If nonlinear activation is considered only in the J-th layer, let

$$b_l^{(J)} = \begin{cases} -B^{(J)}, & l = n \\ t_k, & l = (k+1)n, 1 \le k \le m \\ B^{(J)}, & \text{otherwise} \end{cases},$$

what is different from the case in one-dimensional input is

$$B^{(0)} = \max_{x \in \Omega} \max_{k=1,2,\cdots,dn} |x_k|, \Omega \in [-1, 1]^{d \times n},$$

$$B^{(1)} = \left(\|w^{(1),1}\|_1, + \cdots + \|w^{(1),d}\|_1\right) B^{(0)}.$$

We have

$$\begin{aligned} h_l^{(J)} &= \sigma(T \cdot x - b^{(J)})_l \\ &= \begin{cases} \alpha_0 x + B^{(J)}, & l = n \\ (\alpha_k - t_k)_+, & l = (k+1)n, 1 \le k \le m \\ 0, & \text{otherwise} \end{cases}, \end{aligned}$$

let $w^{(J+1)} = \{w_0^{J+1}\} = \{1\}$ and

$$b_l^{(J+1)} = \begin{cases} -B^{(J)}, & l = 1 \\ 0, & \text{otherwise} \end{cases},$$

then $T^{w^{(J+1)}} = I_{d_J}$ and

$$h_l^{(J+1)} = \sigma(I \cdot h^{(J)} - b^{(J+1)})_l = \begin{cases} B^{(1)}, & l = 1 \\ \alpha_0 x + B^{(J)}, & l = n \\ (\alpha_k - t_k)_+, & l = (k+1)n, 1 \le k \le m \\ 0, & \text{otherwise} \end{cases}.$$

For a set of $\mathbf{w}$, $\mathbf{b}$ constructed above, there exists $f_{J+1}^{\mathbf{w},\mathbf{b}} \in \mathcal{H}_{J+1}^{\mathbf{w},\mathbf{b}}$, such that

$$\| f - f_{J+1}^{\mathbf{w},\mathbf{b}} \|_{C(\Omega)} \le c_0 v_{f,2} max\{\sqrt{log\, m}, \sqrt{nd}\} m^{-\frac{1}{2}-\frac{1}{nd}}.$$

If there is no untrivial common factor, we consider single convolutional layer, at this time, $w^{(1),i} = W^i$ is a sequence of length $(m+1)d$, then $s = (m+1)n-1$. In other words, this is a special case when $J = 1$, that is, there exists $f_2^{\mathbf{w},\mathbf{b}} \in \mathcal{H}_2^{\mathbf{w},\mathbf{b}}$, such that

$$\| f - f_2^{\mathbf{w},\mathbf{b}} \|_{C(\Omega)} \le c_0 v_{f,2} max\{\sqrt{log\, m}, \sqrt{nd}\} m^{-\frac{1}{2}-\frac{1}{nd}}.$$

A.2 Model Training Results in the Trajectory Prediction Experiments

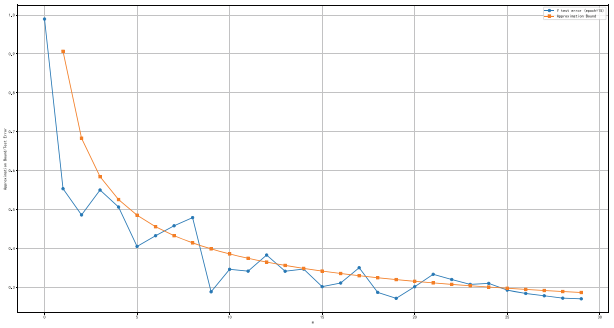

Fig. 4. The experiment on y-axis.

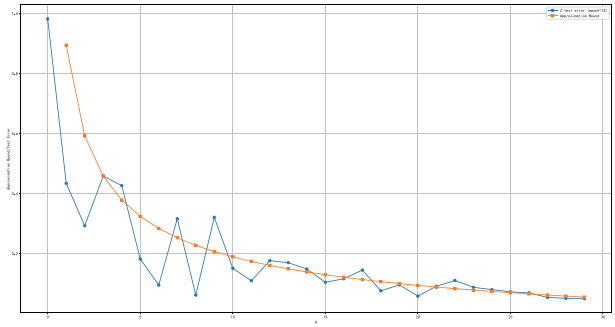

Fig. 5. The experiment on z-axis.

A.3 Back Propagation Process of UniverApproCNN

Take the single-output CNN with two convolutional layers and a full connected layer considered above as an example, find the derivative of the loss function L with respect to $w_k^{(1)}$, where $L = (Y - y)^2$, and $y = \sum_{l=1}^{d_2} c_l h_l^{(2)}$. The derivative of loss function $L = (Y - y)^2$ with respect to $w_k^{(1)}$ is

$$\frac{\partial L}{\partial w_k^{(1)}} = \frac{\partial L}{\partial y} \cdot \sum_{l=1}^{d_2} \frac{\partial y}{\partial h_l^{(2)}} \cdot \frac{\partial h_l^{(2)}}{\partial w_k^{(1)}},$$

For fixed $0 \leq k \leq s$,

Define $a^{(j)} = T^{w^{(j)}} \cdots T^{w^{(1)}} x$, then essentially we find $\frac{\partial a_l^{(2)}}{w_k^{(1)}}$ and $\frac{\partial B^{(2)}}{\partial w_k^{(1)}}$. For fixed $0 \leq k \leq s$, observe that when $k+1 \leq i \leq k+d$, $a_i^{(1)}$ contains items related to $w_k^{(1)}$, and $\frac{\partial a_i^{(1)}}{\partial w_k^{(1)}} = x_{i-k}$.

Further, when $k+1 \leq l \leq k+d+s$, $a_l^{(2)}$ contains items related to $w_k^{(1)}$, and we only need to consider $\sum_{i=k+1}^{k+d} w_{l-i}^{(2)} a_i^{(1)}$ in $a_l^{(2)}$. Since $w^{(2)}$ is finite supported, we only consider $\sum_{i=max\{k+1,l-s\}}^{min\{k+d,l\}} w_{l-i}^{(2)} a_i^{(1)}$, then $\frac{\partial a_l^{(2)}}{\partial a_i^{(1)}} = w_{l-i}^{(2)}$ and

$$\begin{aligned} \frac{\partial a_l^{(2)}}{w_k^{(1)}} &= \sum_{i=max\{k+1,l-s\}}^{min\{k+d,l\}} \frac{\partial a_l^{(2)}}{\partial a_i^{(1)}} \cdot \frac{\partial a_i^{(1)}}{\partial w_k^{(1)}} \\ &= \sum_{i=max\{k+1,l-s\}}^{min\{k+d,l\}} w_{l-i}^{(2)} \cdot x_{i-k}. \end{aligned}$$

It is easy to find that

$$\frac{\partial B^{(2)}}{\partial w_k^{(1)}} = \sum_{l=1}^{d_2} c_l \cdot \|w^{(2)}\|_1 \cdot (|w_k^{(1)}|)' \cdot B^{(0)}$$

Similar conclusions can be obtained for the CNN with depth $J+1$ and single output.

References

1. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Math. Control Signals Systems **2**(4), 303–314 (1989)
2. Pinkus, A.: Approximation theory of the MLP model in neural networks. Acta Numer. **8**, 143–195 (1999)
3. Zhou, D.X., Jetter, K.: Approximation with polynomial kernels and SVM classifiers. Adv. Comput. Math. **25**(1–3), 323–344 (2006)
4. Breiman, L.: Hinging hyperplanes for regression, classification, and function approximation. IEEE Trans. Inf. Theory **39**(3), 999–1013 (1993)
5. Klusowski, J.M., Barron, A.R.: Approximation by combinations of ReLU and squared ReLU ridge functions with l^1 and l^0 controls (2016)
6. Barron, A.R.: Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans. Inf. Theory **39**(3), 930–945 (1993)
7. Klusowski, J.M., Barron, A.R.: Uniform approximation by neural networks activated by first and second order ridge splines. IEEE Trans. Inf. Theory (2016)
8. Hornik, K.M., Stinchcomb, M., White, H.: Multilayer feedforward networks are universal approximator. IEEE Trans. Neural Networks **2**, 01 (1989)
9. Zhou, D.-X.: Deep distributed convolutional neural networks: Universality. Anal. Appl. **16**(92), 895–919 (2018)
10. Mhaskar, H.N., Poggio, T.: Deep vs. shallow networks: an approximation theory perspective. Anal. Appl. **14**(06), 829–848 (2016)
11. Mallat, S.: Understanding deep convolutional networks. Phil. Trans. R. Soc. A **374**(2065), 20150203–20150203 (2016)
12. Steinwart, I., Thomann, P., Schmid, N.: Learning with hierarchical gaussian kernels. arXiv Machine Learning (2016)
13. Bruna, J., Mallat, S.: Invariant scattering convolution networks. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1872–1886 (2013)
14. Zhou, D.: Universality of deep convolutional neural networks. Appl. Comput. Harmon. Anal. **48**(2), 787–794 (2020)
15. Santurkar, S., Tsipras, D., Ilyas, A., Mądry, A.: How does batch normalization help optimization? In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 2488–2498 (2018)
16. Wang, Y., Wang, Y., Hu, G., Liu, Y., Zhao, Y.: Adaptive skewness kurtosis neural network: enabling communication between neural nodes within a layer, pp. 498–507 (2020)
17. Burger, M., Engl, H.W.: Training neural networks with noisy data as an Ill-posed problem. Adv. Comput. Math. **13**(4), 335–354 (2000)