中山大學 SUN YAT-SEN UNIVERSITY

Growing-Type Weights and Structure Determination of 2-Input Legendre Orthogonal Polynomial Neuronet

Yunong Zhang, Jinhao Chen, Dongsheng Guo, Yonghua Yin and Wenchao Lao

School of Information Science and Technology

Sun Yat-sen University

1. Introduction
2. Theoretical Basis and Analysis
3. Model and Algorithms
4. Numerical Study Results
5. Conclusion

1. Introduction

- Traditional BP neuronet
  - slow convergence
  - local-minima existence
- Most practical systems have multiple inputs
- Thus, a special multi-input neuronet equipped with weights-and-structure-determination algorithms is needed

2. Theoretical Basis and Analysis

Definition 1. For the variable $x \in [-1, 1]$, the $(j+2)$th Legendre orthogonal polynomial can be defined as

$$\varphi_{j+2}(x) = \frac{2j+1}{j+1} x \varphi_{j+1}(x) - \frac{j}{j+1} \varphi_j(x),$$
$$\text{with } \varphi_1(x) = 1 \text{ and } \varphi_2(x) = x,$$

where $\varphi_j(x)$ also denotes the Legendre orthogonal polynomial of degree j, with $j = 1, 2, 3, \cdots$.

Proposition 1. For two continuous independent variables x_1 and x_2, let $F(x_1, x_2)$ denote a given continuous function. Then, there exist polynomials $g_k(x_1)$ and $h_k(x_2)$ (with $k = 1, 2, 3, \cdots$) to formulate $F(x_1, x_2)$; i.e.,

$$F(x_1, x_2) = \sum_{k=1}^{\infty} g_k(x_1) h_k(x_2). \quad (1)$$

中山大學
SUN YAT-SEN UNIVERSITY

2. Theoretical Basis and Analysis

Definition 1. For the variable $x \in [-1, 1]$, the $(j+2)$th Legendre orthogonal polynomial can be defined as

$$\varphi_{j+2}(x) = \frac{2j+1}{j+1} x \varphi_{j+1}(x) - \frac{j}{j+1} \varphi_j(x),$$
$$\text{with } \varphi_1(x) = 1 \text{ and } \varphi_2(x) = x,$$

where $\varphi_j(x)$ also denotes the Legendre orthogonal polynomial of degree j, with $j = 1, 2, 3, \cdots$.

Proposition 1. For two continuous independent variables x_1 and x_2, let $F(x_1, x_2)$ denote a given continuous function. Then, there exist polynomials $g_k(x_1)$ and $h_k(x_2)$ (with $k = 1, 2, 3, \cdots$) to formulate $F(x_1, x_2)$; i.e.,

$$F(x_1, x_2) = \sum_{k=1}^{\infty} g_k(x_1) h_k(x_2). \tag{1}$$

(1) can be reformulated as

$$\begin{aligned} F(x_1, x_2) &= \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} a_m \varphi_m(x_1) b_n \varphi_n(x_2) \\ &= \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \omega_{m,n} \varphi_m(x_1) \varphi_n(x_2) \\ &\approx \sum_{m=1}^{M} \sum_{n=1}^{N_m} \omega_{m,n} \varphi_m(x_1) \varphi_n(x_2) \\ &= \sum_{m=1}^{M} \varphi_m(x_1) \left(\sum_{n=1}^{N_m} \omega_{m,n} \varphi_n(x_2) \right) \\ &= \sum_{k=1}^{K} w_k \phi_k(x_1, x_2) \end{aligned}$$

3. Model and Algorithms

With $k = \sum_{d=1}^{m} N_d - N_m + n$,
we detail $\{\phi_k(x_1, x_2)\}$ below:

$$\begin{cases} \phi_1(x_1,x_2) = \varphi_1(x_1)\varphi_1(x_2), \\ \phi_2(x_1,x_2) = \varphi_1(x_1)\varphi_2(x_2), \\ \qquad \vdots \\ \phi_{N_1}(x_1,x_2) = \varphi_1(x_1)\varphi_{N_1}(x_2), \\ \phi_{N_1+1}(x_1,x_2) = \varphi_2(x_1)\varphi_1(x_2), \\ \qquad \vdots \\ \phi_{N_1+N_2}(x_1,x_2) = \varphi_2(x_1)\varphi_{N_2}(x_2), \\ \qquad \vdots \\ \phi_k(x_1,x_2) = \varphi_m(x_1)\varphi_n(x_2), \\ \qquad \vdots \\ \phi_K(x_1,x_2) = \varphi_M(x_1)\varphi_{N_M}(x_2). \end{cases}$$

Besides, the basis functions used to approximate the target function $F(x_1, x_2)$ can be generated in a given order. For better understanding, two typical orders are presented, which can be expressed as the following limitations via a given positive integer i larger than 1 (i.e., $i > 1$):

Limitation I. $M = i$, and $N_m = i$ with $m = 1, 2, \cdots, M$;

Limitation II. $M = i - 1$, and $N_m = i - m$ with $m = 1, 2, 3, \cdots, M$.

So, the total number of basis functions K can be determined by $K = i^2$ corresponding to Limitation I or $K = i(i-1)/2$ corresponding to Limitation II. It is worth noting that these two limitations are further investigated in the ensuing sections.

3. Model and Algorithms

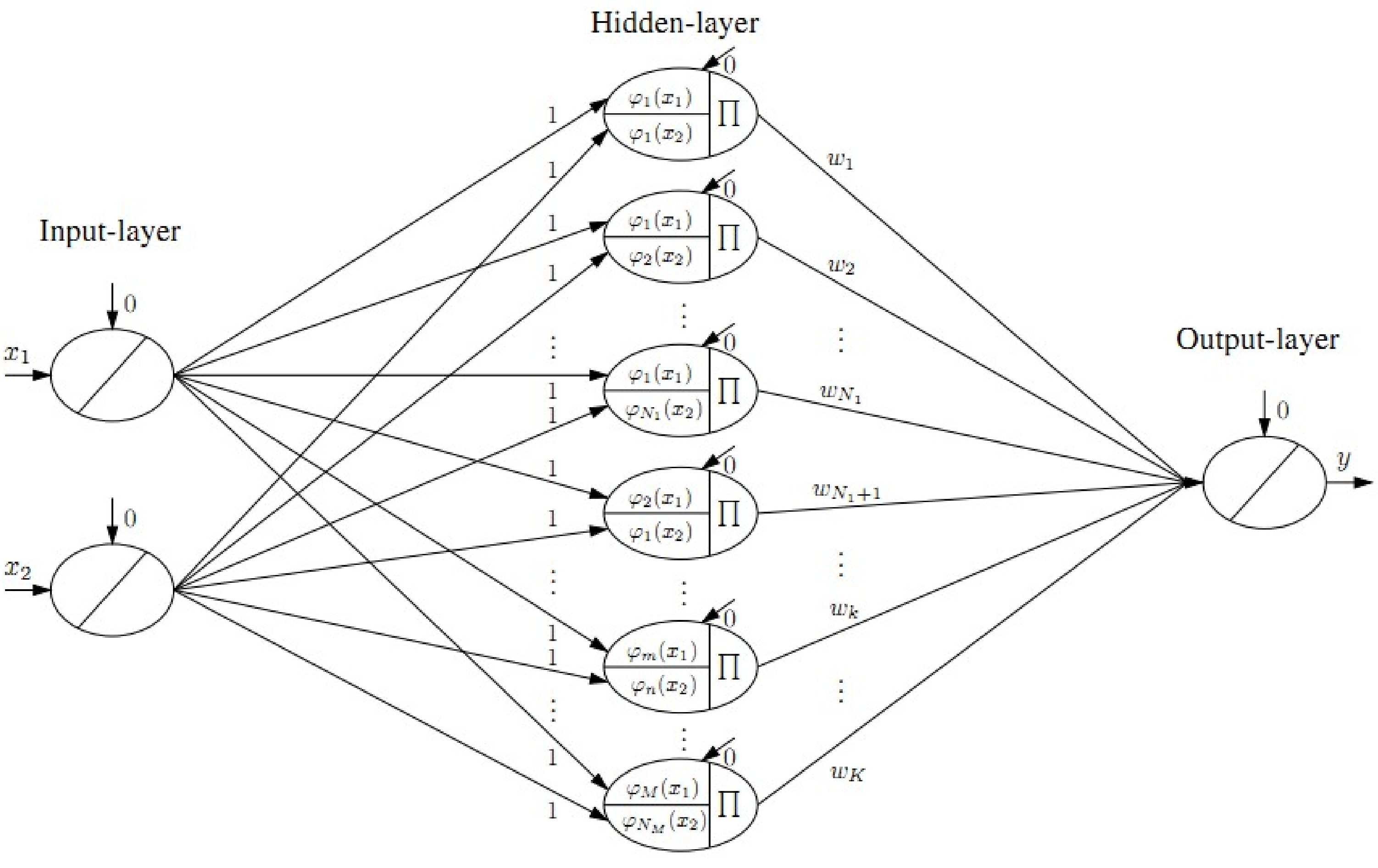


Fig. 1. Model structure of the 2-input Legendre orthogonal polynomial neuronet (2ILOPN).

3. Model and Algorithms

With the mean square error (MSE) defined as

$$E = \sum_{q=1}^{Q} \left(\gamma_q - \sum_{k=1}^{K} w_k \phi_k(\boldsymbol{\chi}_q) \right)^2 / Q. \tag{3}$$

we can have the WDD method.

$$\boldsymbol{w} = (\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\gamma}, \tag{4}$$

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix} \in R^K, \quad \boldsymbol{\gamma} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_Q \end{bmatrix} \in R^Q,$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(\boldsymbol{\chi}_1) & \phi_2(\boldsymbol{\chi}_1) & \cdots & \phi_K(\boldsymbol{\chi}_1) \\ \phi_1(\boldsymbol{\chi}_2) & \phi_2(\boldsymbol{\chi}_2) & \cdots & \phi_K(\boldsymbol{\chi}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{\chi}_Q) & \phi_2(\boldsymbol{\chi}_Q) & \cdots & \phi_K(\boldsymbol{\chi}_Q) \end{bmatrix} \in R^{Q \times K}.$$

3. Model and Algorithms

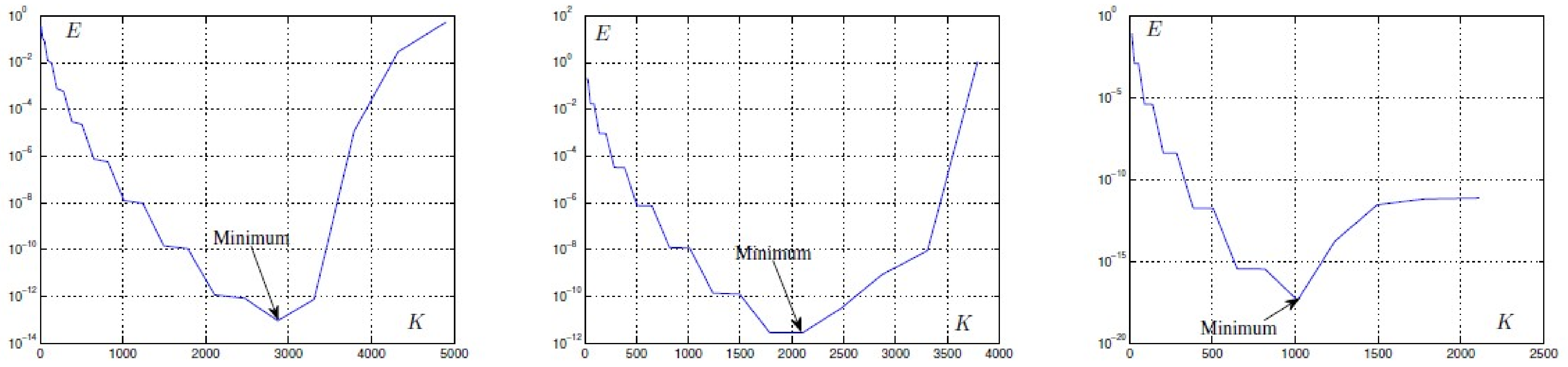
Before presenting such weights-and-structure-determination (WASD) algorithms, let us test the following three target functions to investigate the relationship between the MSE (3) and the number of hidden-layer neurons.

$$F(x_1, x_2) = 10\sin(x_1)e^{-(2x_1)^2-(2x_2)^2}, \quad (5)$$

$$F(x_1, x_2) = 4e^{-x_1^2-(2x_2)^2} + 10, \quad (6)$$

$$F(x_1, x_2) = \sin(\pi x_1 x_2) + 20. \quad (7)$$

3. Model and Algorithms



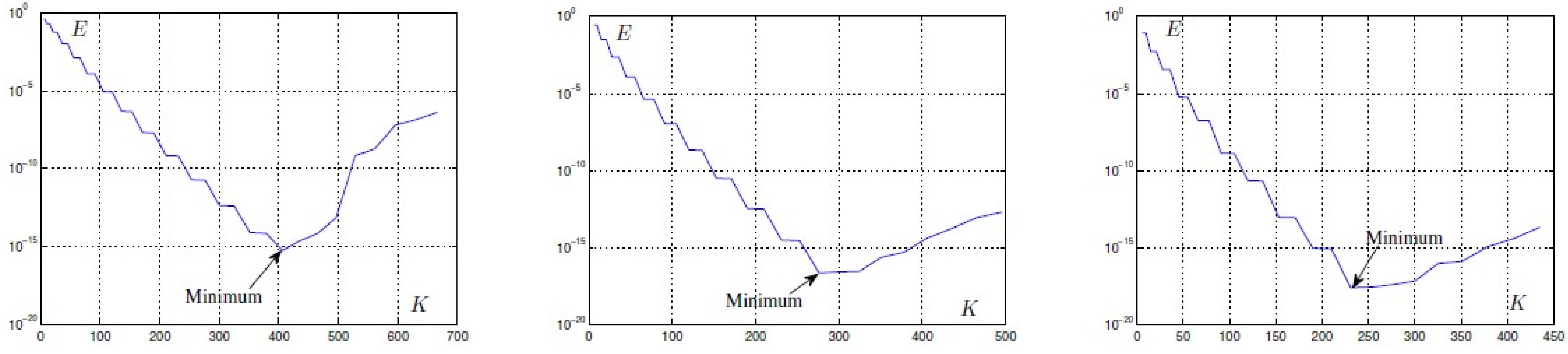
(a) For target function (5)

(b) For target function (6)

(c) For target function (7)

Fig. 2. Relationship between the MSE (i.e., E with $Q = 1156$) and the number of hidden-layer neurons (i.e., K) with Limitation I.

3. Model and Algorithms



(a) For target function (5) (b) For target function (6) (c) For target function (7)

Fig. 3. Relationship between the MSE (i.e., E with $Q = 1156$) and the number of hidden-layer neurons (i.e., K) with Limitation II.

3. Model and Algorithms

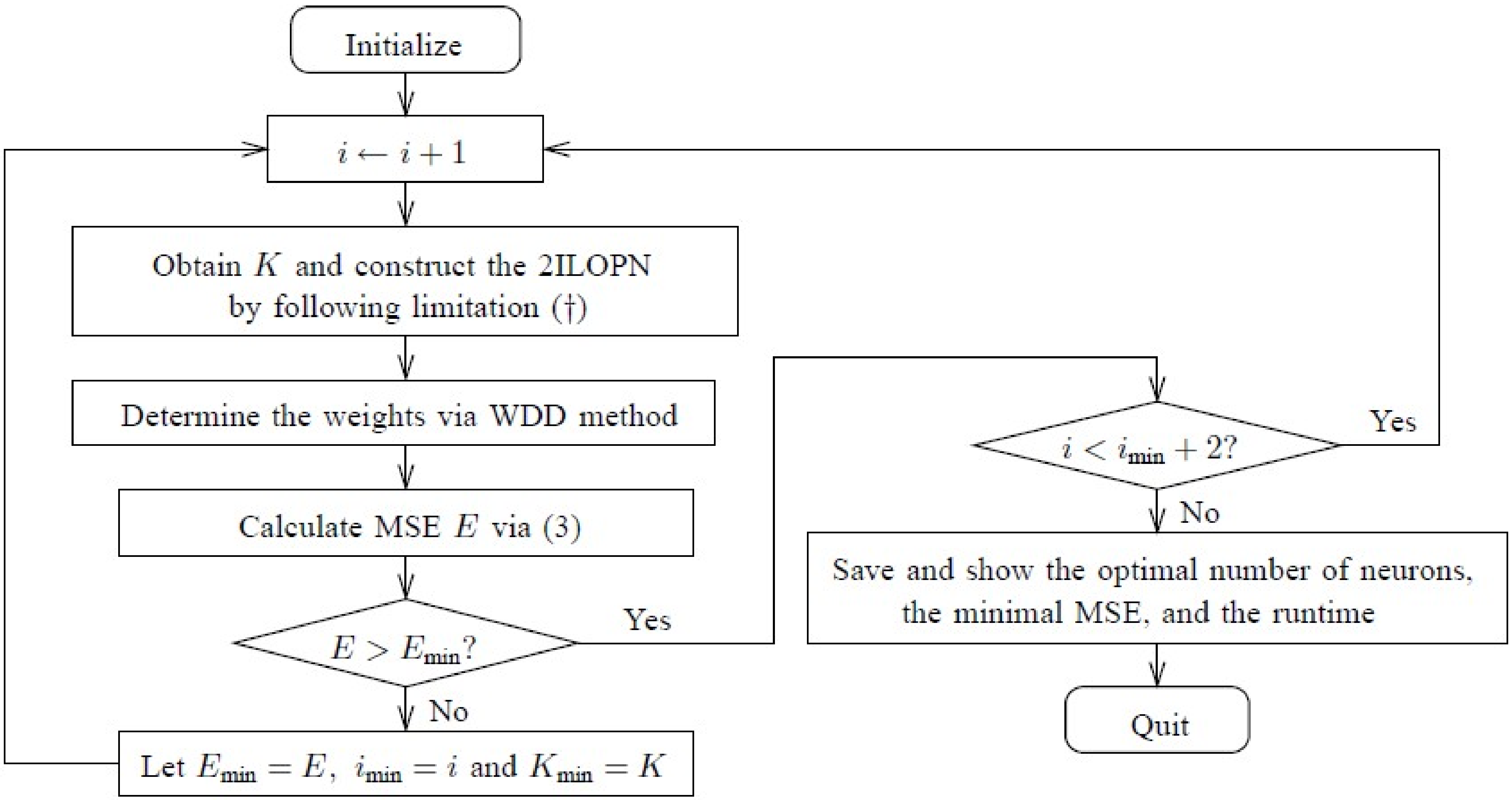


Fig. 4. Flowchart of the WASD algorithms for the 2ILOPN.

3. Model and Algorithms

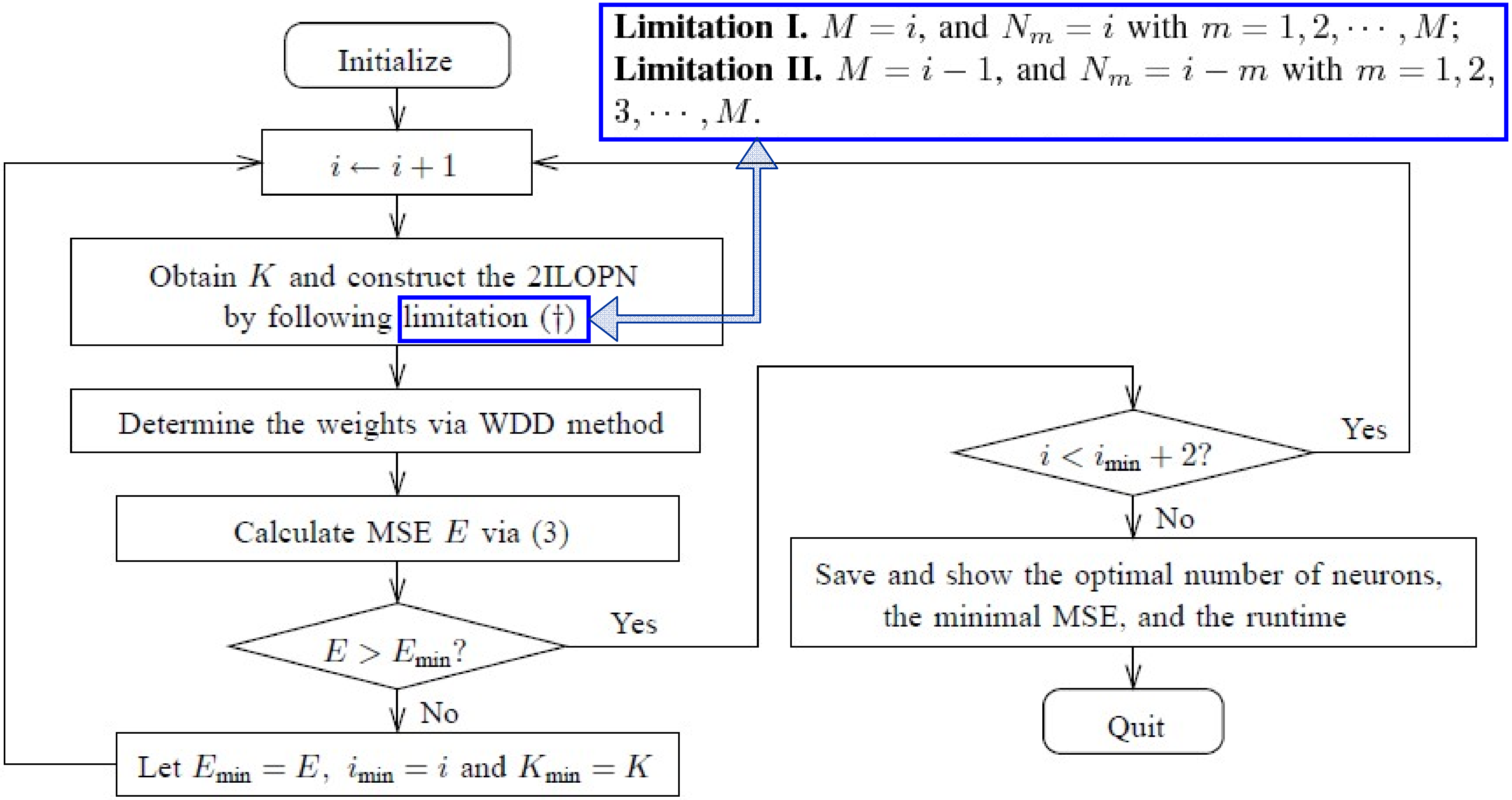


Fig. 4. Flowchart of the WASD algorithms for the 2ILOPN.

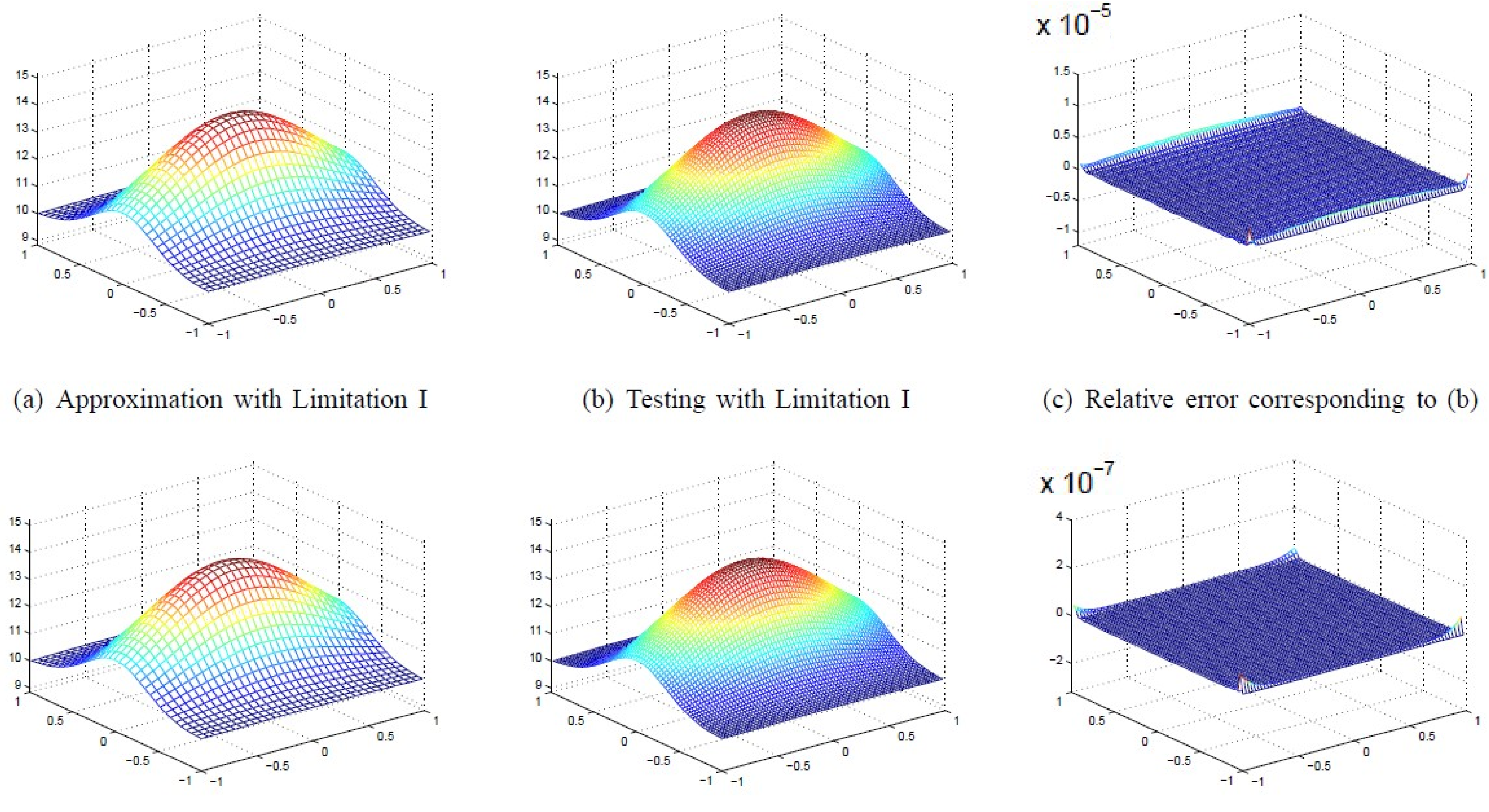
4. Numerical Study Results

TABLE I
APPROXIMATION AND TESTING RESULTS OF THE 2ILOPN ABOUT THREE TARGET FUNCTIONS (5)-(7)

	Target function (5)		Target function (6)		Target function (7)	
	Limitation I	Limitation II	Limitation I	Limitation II	Limitation I	Limitation II
$K_{\min}$	2869	406	2108	276	1014	231
E_{approx}	9.240×10^{-14}	5.937×10^{-16}	2.934×10^{-12}	2.454×10^{-17}	4.194×10^{-18}	2.645×10^{-18}
E_{test}	6.606×10^{-13}	9.928×10^{-16}	6.726×10^{-12}	8.715×10^{-16}	2.403×10^{-18}	2.926×10^{-18}
WASD runtime (s)	326.935	13.731	211.585	5.309	73.346	3.570

Training data : $[-1,1]^2$, 0.06 Testing data : $[-1\ 1]^2$, 0.029

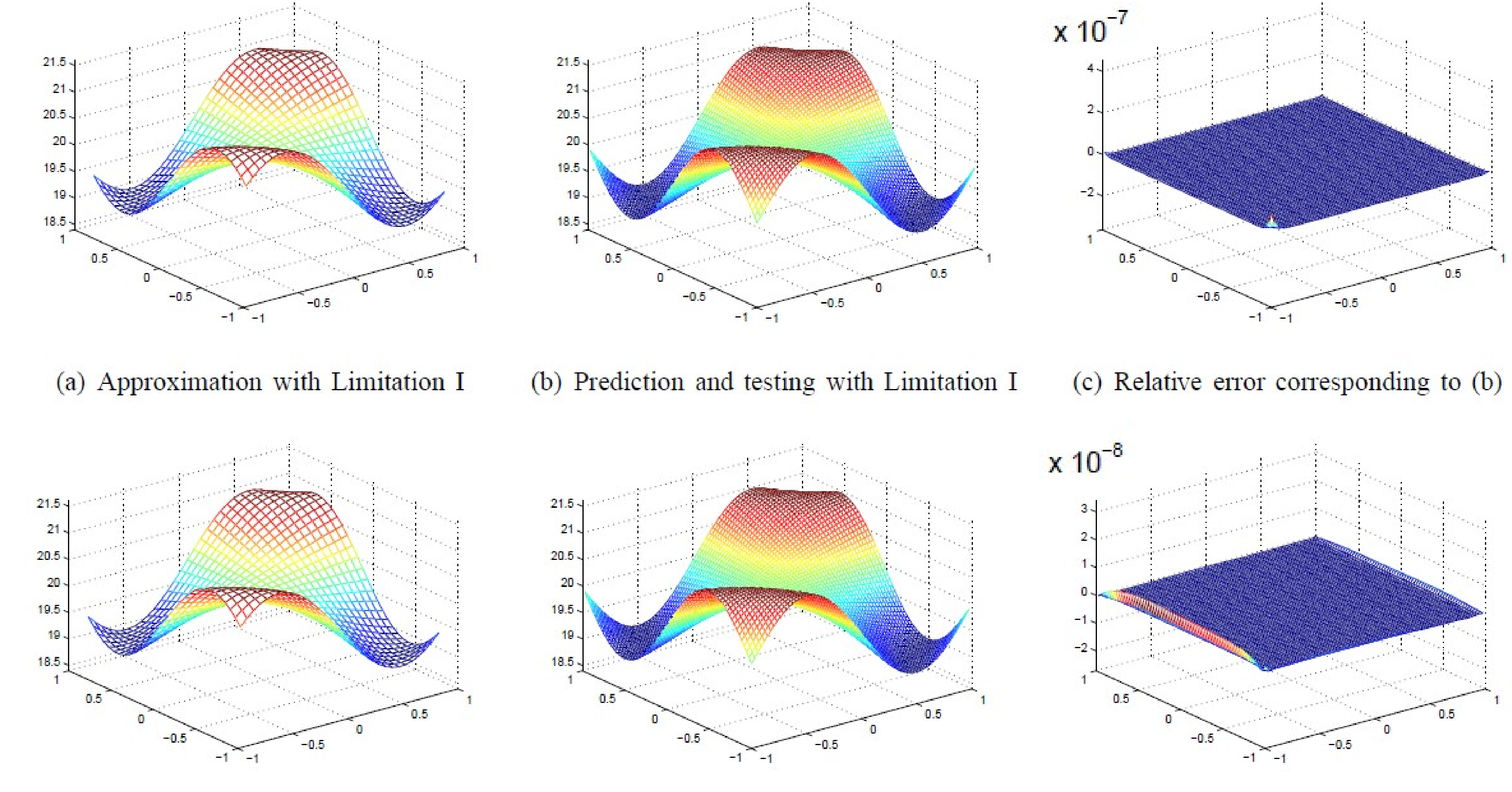
4. Numerical Study Results



(a) Approximation with Limitation I
(b) Testing with Limitation I
(c) Relative error corresponding to (b)
(d) Approximation with Limitation II
(e) Testing with Limitation II
(f) Relative error corresponding to (e)

Fig. 5. Approximation and testing results of the 2ILOPN with either Limitation I or Limitation II about target function (6).

4. Numerical Study Results



(a) Approximation with Limitation I (b) Prediction and testing with Limitation I (c) Relative error corresponding to (b)

(d) Approximation with Limitation II (e) Prediction and testing with Limitation II (f) Relative error corresponding to (e)

Fig. 6. Approximation, prediction and testing results of the 2ILOPN with either Limitation I or Limitation II about target function (7).

5. Conclusion

- In this paper, the 2-input Legendre orthogonal polynomial neuronet (2ILOPN) has been proposed and investigated, which has solidly laid a basis for further research on multi-input neuronets. In addition, two weights-and-structure-determination (WASD) algorithms of growing type have been developed to determine the optimal number of hidden-layer neurons and simultaneously obtain the weights between the hidden-layer and output-layer neurons directly. Numerical-study results have further demonstrated the efficacy of the 2ILOPN equipped with the two WASD algorithms on approximation, generalization and prediction.