



---

*Research article*

## ODE-RU: a dynamical system view on recurrent neural networks

Pinchao Meng\*, Xinyu Wang and Weishi Yin

Changchun University of Science and Technology, changchun, China

\* **Correspondence:** Email: mengpc@cust.edu.cn; Tel: +8613944890712.

**Abstract:** The core of the demonstration of this paper is to interpret the forward propagation process of machine learning as a parameter estimation problem of nonlinear dynamical systems. This process is to establish a connection between the Recurrent Neural Network and the discrete differential equation, so as to construct a new network structure: ODE-RU. At the same time, under the inspiration of the theory of ordinary differential equations, we propose a new forward propagation mode. In a large number of simulations and experiments, the forward propagation not only shows the trainability of the new architecture, but also achieves a low training error on the basis of main-taining the stability of the network. For the problem requiring long-term memory, we specifically study the obstacle shape reconstruction problem using the backscattering far-field features data set, and demonstrate the effectiveness of the proposed architecture using the data set. The results show that the network can effectively reduce the sensitivity to small changes in the input feature. And the error generated by the ordinary differential equation cyclic unit network in inverting the shape and position of obstacles is less than  $10^{-2}$ .

**Keywords:** ordinary differential equation; stability; obstacle inversion; recurrent neural network; gated cell structure

---

### 1. Introduction

Gated recurrent neural networks, which have proven to be very effective against machine learning tasks involving continuous data. This network architecture allows feature information to be transferred from any previous time node to the end node of the network more easily, and can solve the problem of partial gradient disappearance [1], such as long short-term memory networks (LSTM) [2] and gated recurrent units (GRU) [3]. In addition, other problems of machine learning arise from the potential instability in the forward propagation process, especially when the feature data is propagated through the deep neural network (DNN), the noise of the original feature data will affect the stability of the network output [4].

In 2019, the antisymmetric recurrent neural network (AntisymmetricRNN) proposed by Chang et

al. [5] and the semiimplicit method IMEXnet through adaptation of partial differential equations proposed by Haber et al. [6], are neural network structures with stability, so this kind of network can weaken the problem of gradient disappearance or even gradient explosion. The main idea of constructing this type of network comes from the machine learning method based on a dynamic system proposed by Weinan E in 2017 [7], this method regards the residual neural network as a high-dimensional nonlinear function and establishes a connection with the dynamic system. Subsequently, Lu et al. proved that many convolutional networks could be interpreted as different numerical discrete forms of differential equations, and combined the multi-step architecture (LM-architecture) with the residual neural network to form the LM-ResNet and LM-ResNeXt [8] (i.e., the networks obtained by applying the LM-architecture on ResNet and ResNeXt respectively).

In 2018, Chen et al. constructed a new neural network structure based on ordinary differential equations: ODE-NET [9], this shows that neural networks can not only be explained by differential equations but also can use differential equations to construct neural networks satisfying certain conditions.

An improved recurrent neural network with stability is proposed in this paper. Our network is based on simple but effective changes in the popular RNN architecture and is driven by the discrete form of ordinary differential equations. Within the time step, the input vector is transmitted to the network unit node by node, and the extracted feature data is finally output. The network driven by differential equations can make the network stable during operation, reduce the number of parameters and computational complexity, and have nonlinear mapping capabilities. The network is applied to typical ill-posed problems such as wave inverse scattering [10–17], and the shape reconstruction of a single obstacle in the acoustic wave field is discussed. Use the ODE-RU network to solve the obstacle shape inversion problem, use the nonlinear mapping ability of the neural network to fit the relationship between the data, and then reconstruct the obstacle shape.

The paper is structured as follows. In Section 2, we construct the recurrent unit network of ordinary differential equations (ODE-RU) based on the differential equations, and the stability of the network is proved in the sense of Lyapunov stability. In Section 3, we conduct numerical experiments on a synthetic dataset that is constructed to demonstrate the advantages and limitations of the method. We summarize the paper in Section 4.

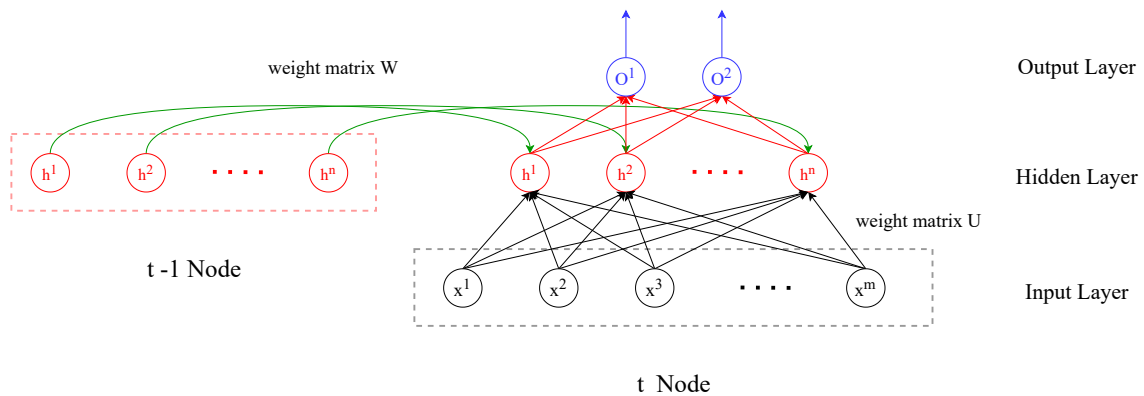
## 2. Ordinary differential equations recurrent unit network

We briefly review recurrent neural networks (RNN) and outline their contact in terms of differential equations and RNN. Based on the basic recurrent neural network structure, borrowing the ideas of gated neural network GRU and LSTM and the stable differential equation form, the recurrent neural network is improved and a new recurrent neural network is constructed. Finally, we analyze the improved stability of our network.

### 2.1. RNN and ordinary differential equations

Recurrent Neural Network (RNN) was first applied in the field of natural language processing and can be modeled according to the characteristics of a language. The state value  $h_t$  of RNN hidden layer depends not only on the current node input  $x_t$ , but also on the output value of the previous hidden layer  $h_{t-1}$ , and the value of the weight matrix  $W, U$  on each hidden layer is shared with each other. Its

network structure is shown in Figure 1.



**Figure 1.** recurrent neural network structure diagram.

On the  $t$ -th node, propagates the features  $h_t$  as follows:

$$h_t = f(Ux_t + Wh_{t-1} + b) \quad (2.1)$$

here,  $f(x)$  is the activation function. In general, the hyperbolic tangent function is selected.

It can be seen from Eq (2.1) that the output  $h_t$  of each hidden layer node is a function of the output  $h_{t-1}$  at the time of last  $t - 1$ , namely:

$$h_t = f(h_{t-1}, \theta) = f(Ux_t + Wh_{t-1} + b) \quad (2.2)$$

Equation (2.2) can also be regarded as the expression of a nonlinear discrete dynamical system of RNN, where  $f$  is a nonlinear vector function,  $\theta = [W, U, x]$  is the parameter vector.

Consider a first-order ordinary differential equation containing only one time variable  $t$ :

$$h'(t) = F(h(t - 1)), \quad t \geq 0 \quad (2.3)$$

where  $h(t) \in \mathbb{R}^n$  and  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ . For most ordinary differential equations, the approximate solution of the equation can be calculated using discretized numerical methods. Here the difference quotient  $\frac{h_t - h_{t-1}}{\varepsilon}$  is used to replace the derivative of the left end of the Eq (2.3). The corresponding difference equation can be written as:

$$\frac{h_t - h_{t-1}}{\varepsilon} = F(h_{t-1}).$$

In order to be approximately valid, by the definition of the derivative,  $\varepsilon > 0$  and  $\varepsilon$  are small enough. Assuming that  $F(h)$  satisfies Lipschitz continuity on  $h$  and the eigenvalue of the corresponding Jacobi matrix has a negative real part, then the result of such a discrete equation is stable.

Given a forward Euler discrete equation with initial values:

$$\begin{aligned} h_t &= h_{t-1} + \varepsilon F(h_{t-1}), \\ h_{t_0} &= h(0) \end{aligned} \quad (2.4)$$

Geometrically, each iteration step of the forward Euler advances from  $h_{t-1}$  along the tangent equation by a step size  $\varepsilon$ . The iterative formula (2.3) can be replaced by the finite difference form and written into a discrete form such as the difference Eq (2.4). There is an approximate equivalence relationship between the two.

The above equation can also be summarized into the following discrete dynamic system form, and  $f$  is a vector function,  $\theta$  is a parameter vector.

$$h_t = f(h_{t-1}, \theta), h_{t_0} = h(0) \quad (2.5)$$

According to Eq (2.2), the forward propagation process of RNN is equivalent to that of the dynamic system (2.5).

In this paper, ordinary differential equations with ideal properties is designed by using the theory of dynamical systems, so that the generated recurrent neural network can inherit these properties. Stability is one of the important properties to be considered in network construction, and we will discuss the form of recurrent neural networks with stability in the next section.

## 2.2. Ordinary differential equations recurrent unit network

Based on the basic recurrent neural network Eq (2.2) formula, we can get

$$h'(t) = H(h(t-1), \eta) - h(t-1) \quad (2.6)$$

where,  $H(h(t-1), \eta)$  represents the functional relationship between the variable  $h(t-1)$  and the parameter vector  $\eta$  under the action of nonlinear function. The forward Euler method is used to express the Eq (2.6) in an approximate iterative form:  $h_t = h_{t-1} + \varepsilon \cdot (H(h_{t-1}, \eta) - h_{t-1})$ .

The function  $H(h_{t-1}, \eta) = \tanh(\tanh(Qx_{t-1} + Wh_{t-1}) \odot h_{t-1} + Mx_{t-1} + b_2)$ , the expression of the improved network can be obtained:

$$h_t = h_{t-1} + \varepsilon \cdot (\tanh(\tanh(Qx_{t-1} + Wh_{t-1}) \odot h_{t-1} + Mx_{t-1} + b_2) - h_{t-1}). \quad (2.7)$$

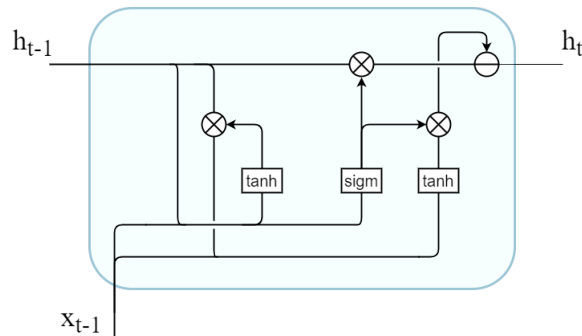
RNNs usually use a gated mechanism, Each gate is often modeled as a single layer network taking the previous hidden state  $h_{t-1}$  and data  $x_t$  as inputs, followed by a sigmoid activation. As an example, LSTM cells make use of three gates, a forget gate, an input gate, and an output gate, GRU cells is simplified to update gate and reset gate, these gates provide the main support for good network performance.

In order to increase the operational performance of the network, a gating mechanism is added to the recurrent unit network of ordinary differential equations. On the basis of Eq (2.7), input gate  $z_t \in \mathbb{R}^{m \times 1}$  is added to regulate the feature selection process from the input layer to hidden layer, then ODE-RU with gated structure can be written as:

$$\begin{aligned} z_t &= \sigma(Vx_t + Uh_t), \\ h_t &= h_{t-1} + \varepsilon \cdot z_{t-1} \odot (\tanh(\tanh(Qx_{t-1} + Wh_{t-1}) \odot h_{t-1} + Mx_{t-1} + b_2) - h_{t-1}) \end{aligned} \quad (2.8)$$

where  $x_t \in \mathbb{R}^n$  and  $h_t \in \mathbb{R}^m$  represent the input vector and the state vector of the hidden layer at the  $t$  node respectively,  $V, Q, M \in \mathbb{R}^{m \times n}$  and  $U, W \in \mathbb{R}^{m \times m}$  are weight matrices,  $\sigma$  denotes the sigmoid function and  $\odot$  denotes the Hadamard product,  $\varepsilon > 0$  is a hyperparameter that represents the step size.

The structure of the operation unit is as follows:



**Figure 2.** ODE-RU unit.

### 2.3. Stability of the methods

In the cyclic neural network, the state of  $t$  node changes as the state of  $t - 1$  node changes. If there exists a state  $h_t^*$  such that  $h_t^* = f(h_t^*, \theta)$ , this state is called fixed point (zero solution)  $h_*$ . If the parameter vector  $\theta$  in the function  $f$  changes, the state of the end node still converges to  $h_*$ , and this property is called stability [18, 19]. If the real parts of the eigenvalues of the Jacobi matrix at the fixed-point  $h_*$  are all negative, then the fixed points are stable. If there is an eigenvalue in the Jacobi matrix whose real part is zero, then the eigenvalue cannot be used to determine whether it is stable or not.

The network structure proposed in this paper is improved on the basis of differential equations. Therefore, to obtain the stability of ODE-RU network, it is necessary to study the stability of the differential equations first. The stability theorem of differential equations can be found in reference [20]. Among them, Lemma 1 gives the stability theorem for the solution of linear differential equations, and Lemma 2 gives the stability relationship between linear equations and their approximate nonlinear equations. And the discrete Eq (2.4) has a corresponding relationship with the network structure in this paper, then the network studied in this paper can inherit the stability of the discrete equation.

**Proposition 2.1.** *An ordinary differential equation is stable when the eigenvalues of the equation's Jacobi matrix satisfy*

$$\max_{i=1,2,\dots,n} \operatorname{Re}(\lambda_i(J(t))) < 0, \quad \forall t \geq 0$$

where  $\operatorname{Re}$  is the real part of the complex number.

**Lemma 1.** *If the zero solution of the linear stationary system (the singularity) will satisfy the following properties according to the properties of the characteristic equation roots: If the characteristic equation has a repeated root  $\lambda$ , then the singularity is usually a degenerate node. When  $\lambda < 0$ , both types of nodes are stable, while the zero solution is asymptotically stable. At  $\lambda > 0$ , both the singularity and the corresponding zero solution are unstable.*

**Lemma 2.** *If the characteristic equation does not have zero roots or zero real part roots, the stability state of the zero solution of the corresponding nonlinear equation is consistent with the stability state of the zero solution of the linear approximation equation.*

In other words, when the roots of the characteristic equation all have negative real parts, the zero solution of the nonlinear equation is asymptotically stable, and when the roots of the characteristic equation have positive real parts, the zero solution is unstable.

Relying on stability alone is not enough for them to capture long term dependencies. As Haber and Ruthotto point out in the literature.  $Re(\lambda_i(J(t))) \ll 0$  results in a lossy system, where the energy or signal in the initial state is dissipated over time. Using such an ODE as the underlying dynamical system of a recurrent network will lead to catastrophic forgetting of the past inputs during the forward propagation [21]. To this end, Chang et al. proposed that under standard critical conditions,

$$\max_{i=1,2,\dots,n} Re(\lambda_i(J(t))) \approx 0 \quad \forall i = 1, 2, \dots, n,$$

the system preserves the long-term dependencies of the inputs while being stable [22].

To discuss the influence of the choice of weight matrix on the stability of the network, it is necessary to reduce the complexity of network computation and study the properties of the network without input data. Therefore, the forward propagation formula (2.8) of ODE-RU network is written as follows, and study its stability.

$$h_t = h_{t-1} + \varepsilon \cdot \sigma(Uh_{t-1}) \odot (\tanh(\tanh(Wh_{t-1}) \odot h_{t-1} + b_2) - h_{t-1}) \quad (2.9)$$

This can also be viewed as an ordinary differential equation. Then Eq (2.10) can be regarded as the forward Euler discrete form of the following ordinary differential equation:

$$h' = \sigma(Uh) \odot (\tanh(\tanh(Wh) \odot h) - h). \quad (2.10)$$

The right end of Eq (2.9) is the state vector studied in the same node. For this reason, Eq (2.10) can be abbreviated to the form without node number.

This network can inherit the stability of the solution of the differential equation. Therefore, for ODE-RU, the following theorem holds:

**Theorem 2.1.** *For any weighted matrix  $W, U$ , the ordinary differential equations (2.9) is asymptotically stable at the singularity  $h = 0$ .*

**proof of Theorem 2.1.** First, consider the two-dimensional first-order stationary differential equations:

$$\begin{cases} \frac{dh'}{dh_1} = P(h_1, h_2) = \sigma(U_1 h) \odot (\tanh(\tanh(Wh) \odot h_1) - h_1), \\ \frac{dh'}{dh_2} = Q(h_1, h_2) = \sigma(U_2 h) \odot (\tanh(\tanh(Wh) \odot h_2) - h_2) \end{cases} \quad (2.11)$$

The point  $(x^*, y^*)$  which simultaneously satisfies  $P(x, y) = 0, Q(x, y) = 0$  is the singularity of the system of differential equations (2.11),  $x = x^*, y = y^*$  is the solution of the equation. Obviously,  $(0, 0)$  is the only singularity of this equation.  $\lambda_1, \lambda_2$  represents the root of the characteristic equation of the corresponding square matrix. The Jacobi matrix corresponding to Eq (2.10) is

$$J = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}.$$

At this time,  $\lambda_1 = \lambda_2 = -\frac{1}{2}$ . According to the matritized Jordan canonical form theory, when the Jacobi matrix  $J = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$ , the eigenroot is double root, if  $\lambda_1 < 0$ , when  $t \rightarrow +\infty$ , there is a solution  $(\bar{x}, \bar{y}) = (0, 0)$  of the equation, then such singularities are called stable critical nodes.

From the two-dimensional conclusion to the higher dimensional case, we know that the singularity of the equation in the  $m$  dimension is also  $(\bar{x}, \bar{y}) = (0, 0)$ . Let  $y(h) = \tanh(\tanh(Wh) \odot h) - h$ , then  $h' = \sigma(Uh) \odot y(h)$ , and  $y(h) \in \mathbb{R}^m$ . First, calculate the partial derivative of  $y(h)$  with respect to  $h$ :

$$y_i = \tanh\left(\left(\sum_{j=1}^m F_{ij} \cdot \tanh\left(\sum_{j=1}^m W_{ij}h_j\right)\right) \times h_i\right) - h_i, \quad (i = 1, 2, \dots, m),$$

$$\frac{\partial y_i}{\partial h_j} = \begin{cases} (1 - \tanh^2(F_i \cdot \tanh(Wh) \odot h)) \times \left[\left(\sum_{j=1}^m W_{ij}\right)h_i + \sum_{j=1}^m \tanh(W_j h)\right], & i = j, \\ (1 - \tanh^2(F_i \cdot \tanh(Wh) \odot h)) \times \left[\left(\sum_{j=1}^m W_{ij}\right)h_i\right], & i \neq j, \end{cases}$$

At the stable critical point  $h = (0, 0, \dots, 0) \in \mathbb{R}^m$ , we have  $\frac{\partial y}{\partial h} = \begin{cases} -1, & i = j \\ 0, & i \neq j \end{cases}$ . Then, the Jacobi matrix corresponding to Eq (2.9) can be written as  $J = [j_{ij}] = \left[\frac{\partial h'_i}{\partial h_j}\right]$ , where

$$j_{ij} = \frac{\partial h'_i}{\partial h_j} = \sigma(U_i h)(1 - \sigma(U_i h)) \cdot U_{ij} \cdot y_i + \sigma(U_i h) \cdot \frac{\partial y_i}{\partial h_j}.$$

At the fixed point, the Jacobi matrix of the equation is  $J = -\frac{1}{2}E$ , where  $E$  is the identity matrix. The characteristic root of the equation is  $\lambda_i = -\frac{1}{2}(i = 1, 2, \dots, m)$ . From Lemmas 1 and 2, it can be known that the equation is asymptotically stable at the zero solution  $h = (0, 0, \dots, 0) \in \mathbb{R}^m$ .

Therefore, the ordinary differential Eq (2.10) is asymptotically stable at the zero solution(singularity).

Given a stable ordinary differential equation, the stability of its forward Euler discretization is still uncertain. By studying the stability conditions of the forward Euler method, the following propositions are obtained [23]:

**Proposition 2.2.** *The forward propagation of the forward Euler stability is stable, when*

$$\max_{i=1,2,\dots,n} |1 + \varepsilon \lambda_i(J_i)| \leq 1 \quad (2.12)$$

where  $|\cdot|$  represents the absolute value or modulus of the complex number, and  $J_i$  is the Jacobi matrix of ordinary differential equations.

**Theorem 2.2.** *If the ordinary differential Eq (2.9) satisfies the condition of Proposition 2, then the forward Euler discrete form of Eq (2.10) is stable, that is, the discrete Eq (2.10) is stable at the singularity  $h = 0$ .*

Theorem 2.2 shows that the ODE-RU network established in this paper is stable. The following is a proof of Theorem 2.2.

**proof of Theorem 2.2.** According to Proposition 2.2, the stability of the forward Euler method needs to satisfy Eq (2.12). From the proof process of Theorem 2.1, we know that the eigenvalues  $\lambda_i(J_i)$  of the Jacobian matrix  $J = -\frac{1}{2}E$  are all negative numbers. so  $|1 + \varepsilon\lambda_i(J_i)| < 1$  is true. The ODE-RU network structure is the same as the discrete form of the ordinary differential Eq (2.9). Therefore, the discrete equation is stable at the singularity, that is, the ODE-RU network is stable at the singularity  $h = 0$ .

When the ODE-RU network is slightly disturbed, it can return to the initial equilibrium state when the time tends to infinity, and it has the ability to fit nonlinear systems. However, the inverse problem has nonlinear ill-posedness. Therefore, ODE-RU is used to solve the sound wave obstacle shape inversion problem.

### 3. Numerical experiments

The ODE-RU proposed in this paper can be applied to obstacle inversion problems ([24–31]). Suppose there exists an impenetrable obstacle  $D$  with a soundsoft boundary  $\partial D$  in two dimensions space. The acoustic scattering problem can be illustrated as:

$$\begin{cases} \Delta u + k^2 u = 0, & \text{in } R^2 \setminus \bar{D} \\ u = u^s + u^i = 0, & \text{on } \partial D \\ \lim_{r \rightarrow \infty} r^{1/2} \left( \frac{\partial u^s}{\partial r} - iku^s \right) = 0, & r = |x| \end{cases} \quad (3.1)$$

where,  $u$  is the total field,  $u^i$  is the incident field,  $u^s$  is the scatter field,  $k$  is the wave number,  $u = 0$  represents the Dirichlet boundary condition, which depends on the physical features of the obstacle boundary. The asymptotic relationship between the scatter field  $u^s$  and the far field  $u^\infty(\hat{x})$  can be described as:

$$u^s(x) = \frac{e^{ik|x|}}{\sqrt{|x|}} \left( u^\infty(\hat{x}) + O\left(\frac{1}{|x|}\right) \right), \quad |x| \rightarrow \infty \quad (3.2)$$

Consistently holds for all observation directions  $\hat{x} = \frac{x}{|x|}$ .

Hence, this paper adopts the method of integral equation to derive data in the scatter field from the obstacle boundary  $\partial D$  and the incident field  $u^i$ . Let the incident wave number  $k = 1.5$ , and solves the far-field data by formula (3.2). The inverse problem of our concern is the recovery of obstacle  $\partial D$  by information from  $u^\infty(\hat{x})$ .

**Assumption 3.1.** For the incident field  $u^i(x) = e^{ikx \cdot d}$  with incident directions  $d = (\cos(\alpha), \sin(\alpha))$ , the incident angle  $\alpha$  is uniformly distributed within  $[0, 2\pi)$ :

$$\alpha = \left\{ 0, \frac{2\pi}{n}, \dots, \frac{2(n-1)\pi}{n} \right\}$$

where  $n$  is the number of incident directions.

**Assumption 3.2.** The far-field data can be expressed as  $(x_{11}, x_{12}, \dots, x_{1N}) \in C^{N'}$  for the incident angle  $\alpha = 0$ ,  $(x_{n1}, x_{n2}, \dots, x_{nN}) \in C^{N'}$  for the incident angle  $\alpha = \frac{2(n-1)\pi}{N}$ .

For  $n$  incident directions and  $N$  observation directions, the far-field data can be obtained as:

$$X = (x_{11}, x_{12}, \dots, x_{21}, \dots, x_{2N}, \dots, x_{n1}, \dots, x_{nN}) \in C^{nN'}$$



Let  $M = n \times N$ . Then, the far-field data can be rewritten as:  $X = (x_1, x_2, \dots, x_M) \in C^M$ . Each far-field data  $x_t = a_t + i \cdot b_t$  was rewritten as  $x_t = (a_t, b_t)^T, t = 1, 2, \dots, M$ . The set of all far-field data can be expressed as  $X = (x_1, x_2, \dots, x_M) \in \mathbb{R}^{2 \times M}$ . The far-field data were taken as the scattering information to reconstruct the shape of the obstacle.

**Assumption 3.3.** We assume that the boundary curve of the obstacle  $D$  has the following parameterized representation:

$$\partial D : Z(t) = (Z_1(t), Z_2(t)), 0 \leq t \leq 2\pi,$$

where  $Z_1(t)$  and  $Z_2(t)$  admit the following (truncated) Fourier representations:

$$\begin{cases} Z_1(t) = \frac{a_0}{2} + \sum_{q=1}^Q a_q \cos(qt) + \sum_{q=1}^Q b_q \sin(qt), \\ Z_2(t) = \frac{b_0}{2} + \sum_{q=1}^Q c_q \cos(qt) + \sum_{q=1}^Q d_q \sin(qt), \end{cases}$$

where  $Q \in \mathbb{N}$ . Let  $Y = (y_1, y_2, \dots, y_m)$  denote the ordered set of the Fourier coefficients  $a_0, b_0, a_q, b_q, c_q, d_q, q = 1, 2, \dots, Q$ . Where  $m = 4Q + 2$ .

Considering the spatial relationship between far-field data and the interaction between shape parameters, the ODE-RU was established based on stability of differential equations and recurrent neural network, and used for inversion of shape parameters  $Y' = (y'_1, y'_2, \dots, y'_m) \in \mathbb{R}^m$ .

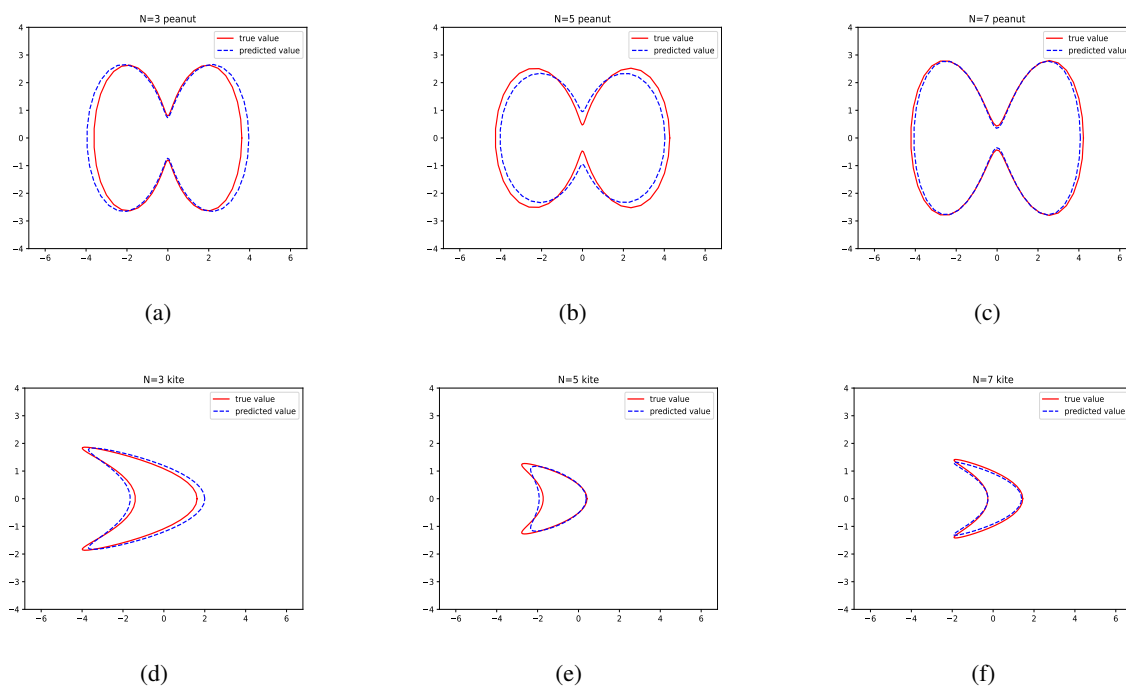
The real parameter information and far-field data of obstacles are used to construct the training data set, and the dataset is divided into a training set and a test set at the ratio of 9:1.

The far-field data is passed as input data to the network through the nodes in turn, and the extracted obstacle parameter information is output through the network operation. We use the crossentropy calculate the loss and SGD with momentum as optimizers. According to multiple experiments, the hyper parameters for the inversion model were set and listed in Table 1.

**Table 1.** Hyper parameters of the ODE-RU.

Hyper parameters	batchsize	batch number	Number of iterations
Value	512	1000	20

**Example 1.** When the incident point and the observation point are the same as  $N$ , the shape of the obstacle is inverted, and the incident angle  $\alpha = \{0, \frac{2\pi}{N}, \dots, \frac{2(N-1)\pi}{N}\}$ . If the number of incident points and observation points are both  $N = 7$ , then the far-field data is the input matrix of  $49 \times 2$ , and the 49 input vectors are assigned to each node of the network according to the order of observation points. In other words, the input dimension  $n = 2$ , the number of nodes  $t = 49$ , and the shape inversion task is to output the extracted feature vectors after the network gets all 49 input vectors, so as to reconstruct the shape and position of obstacles. Figure 3 is the inverse effect of the shape and position of the peanut and the kite at  $N = 3, 5, 7$ .



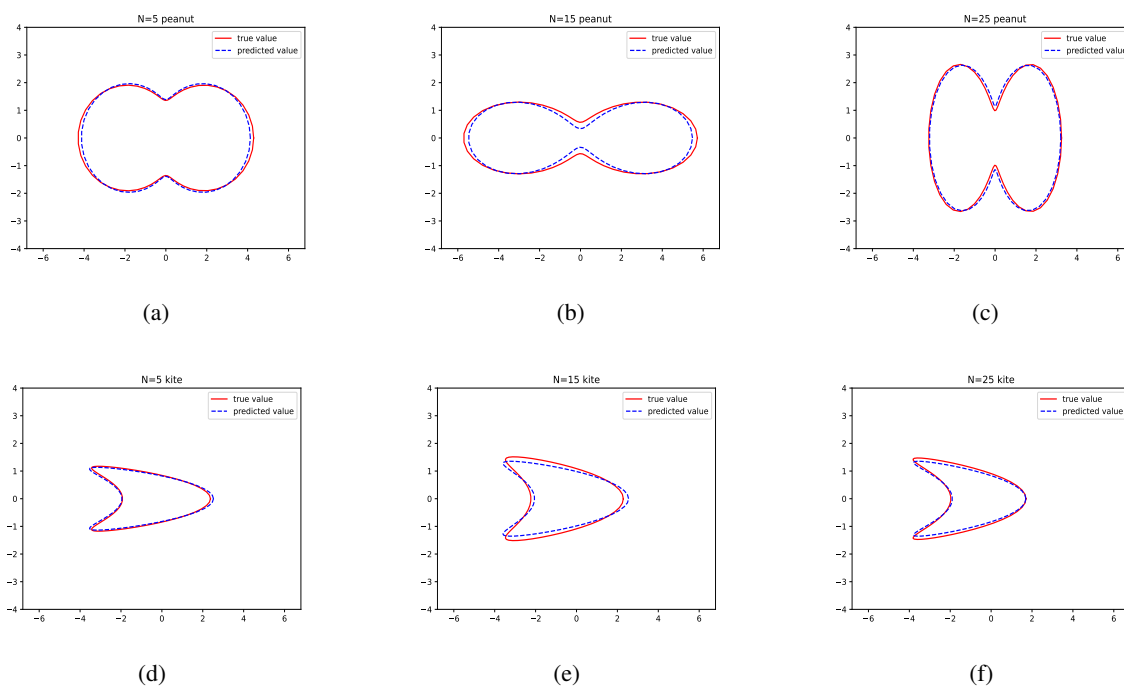
**Figure 3.** Inversion of peanut and kite shapes when the number of observation points and incidence points are both  $N = 3, 5, 7$ .

Table 2 summarizes the performance of the existing methods and the network studied in this paper, where the error is the error of the inversion of different shapes. For the current inversion problems, the ODE-RU network proposed in this paper is superior to other gated recurrent network models in inversion time, and has the same inversion effect.

**Table 2.** Error evaluation of inversion effect under different networks.

Network	N=3	N=5	N=7
GRU (peanut)	3.64E-02	2.14E-02	2.17E-02
LSTM (peanut)	3.47E-02	2.21E-02	2.09E-02
ODE-RU (peanut)	3.55E-02	2.20E-02	2.18E-02
GRU (kite)	3.59E-02	2.26E-02	2.16E-02
LSTM (kite)	3.63E-02	2.19E-02	2.17E-02
ODE-RU (kite)	3.61E-02	2.25E-02	2.20E-02

**Example 2.** Verify the network inversion effect under a small amount of information. Take the number of incident points as one, then the fewer the number of observation points, the less the obstacle shape information contained in the far-field data. Similar to the previous experiment, an input vector is transferred to the network at each node, with input dimension  $n = 2$  and the number of nodes  $N = 5, 15, 25$ , uses a single planar wave with the incident direction  $d = (1, 0)$ . The inversion results are shown in Figure 4.



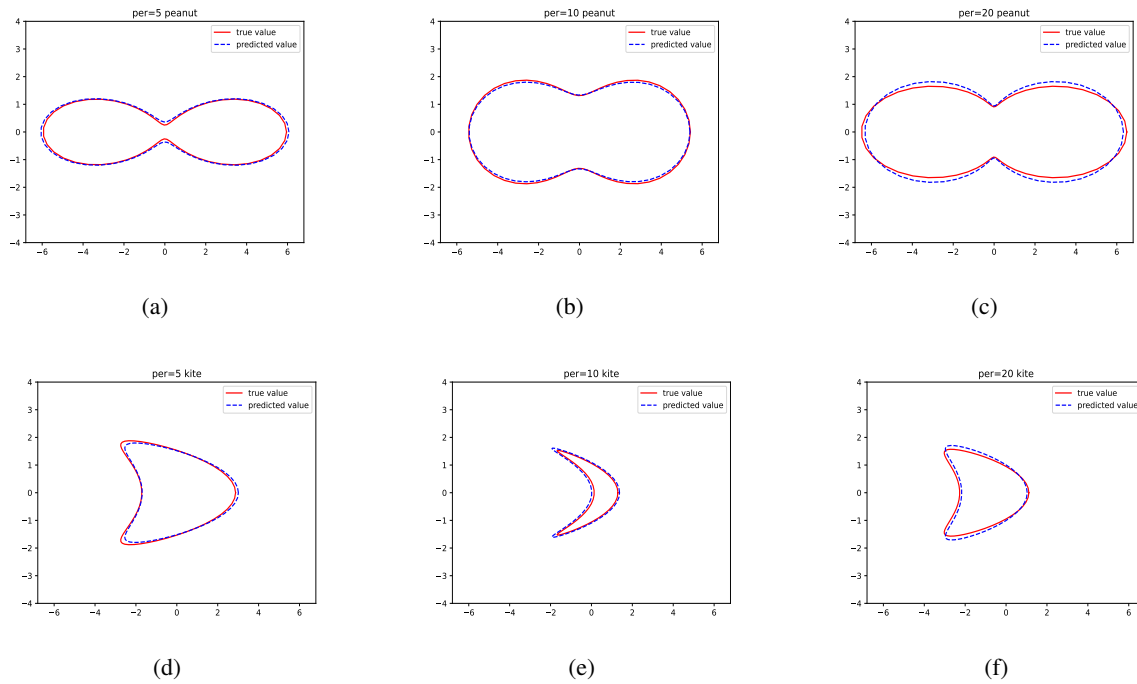
**Figure 4.** Inversion of peanut and kite shapes when the number of observation points is  $N = 5, 15, 25$ .

It can be seen from Figure 4. that with the addition of far-field information, the recovery effect is obviously better. The data results are shown in Table 3. The performance of ODE-RU network is comparable to that of the other two networks, and the error of the results is basically similar to that of LSTM.

**Table 3.** Error evaluation of inversion effect under different networks.

Network	N=5	N=15	N=25
GRU (peanut)	2.94E-02	2.15E-02	2.02E-02
LSTM (peanut)	3.07E-02	2.24E-02	2.36E-02
ODE-RU (peanut)	3.09E-02	2.16E-02	2.19E-02
GRU (kite)	3.05E-02	2.06E-02	2.15E-02
LSTM (kite)	2.89E-02	2.13E-02	2.09E-02
ODE-RU (kite)	3.09E-02	2.15E-02	2.12E-02

**Example 3.** In order to better show the inversion effect of the network studied in this paper, we add noise on the basis of the original data, and consider the inversion effect of different networks when the far-field characteristic data is not accurate enough. In the case of a single incident point and seven observation points, the proportion of noise is added to the original characteristic data. At this point, the input data dimension of each node is still  $n = 2$  and the number of nodes is  $t = 7$ . Figure 5 shows represents the inversion effect with the introduction of 5, 10 and 20% noise.



**Figure 5.** The peanut and kite shapes obtained by inversion when the noise ratio is  $Per = 5, 10, 20$ .

**Table 4.** Error of inversion effect under different networks.

Network	per=5	per=10	per=20
GRU (peanut)	3.04E-02	2.57E-02	3.64E-03
LSTM (peanut)	2.96E-02	2.62E-02	0.37E-02
ODE-RU (peanut)	2.95E-02	2.56E-02	3.55E-03
GRU (kite)	2.99E-02	2.49E-02	2.59E-03
LSTM (kite)	2.87E-02	2.52E-02	0.31E-02
ODE-RU (kite)	3.01E-02	2.51E-02	2.63E-03

As shown in Figure 5 and Table 4, when the far-field data contained a low level of noises, our model could accurately inverse the shape parameters and reconstruct the shape of the obstacle. It can be seen that the network is very robust against noises.

#### 4. Conclusions

From the point of view of a dynamical system, this paper presents a viewpoint of interpreting recurrent neural network as an ordinary differential equation. By connecting neural network with the theory of ordinary differential equation, a new recurrent network structure is designed through the discrete form of an ordinary differential equation, and the network structure designed by this method can inherit the properties of the differential equation. By using the obstacle shape inversion experiment,

it is verified that the proposed network can be successfully trained and predicted, and it is found that the method can maintain the performance of the network.

## Acknowledgments

The work of P Meng, X Wang, W Yin was supported by the Jilin Sci-Tech fund under JJKH20210797KJ.

## Conflict of interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled,

## References

1. J. Collins, J. Sohl-Dickstein, D. Sussillo, Capacity and trainability in recurrent neural networks, Paper presented at 5th International Conference on Learning Representations, 2017.
2. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
3. K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et al., Learning phrase representations using RNN encoder-decoder for statistical machine translation, *Comput. Sci.*, (2014), 1723–1734. <https://doi.org/10.3115/v1/D14-1179>
4. L. Bottou, F. E. Curtis, J. Nocedal, Optimization methods for large-scale machine learning, *SIAM*, **60** (2018), 223–231. <https://doi.org/10.1137/16M1080173>
5. B. Chang, M. Chen, E. Haber, E. H. Chi, AntisymmetricRNN: a dynamical system view on recurrent neural networks, Paper presented at 7th International Conference on Learning Representations, 2019.
6. E. Haber, K. Lensink, T. Eran, L. Ruthotto, IMEXnet: A forward stable deep neural network, Paper presented at: Proceedings of the 36th International Conference on Machine Learning, 2019.
7. W. E, A proposal on machine learning via dynamical systems, *Commun. Math. Stat.*, **1** (2017), 1–11. <http://doi.org/10.1007/s40304-017-0103-z>
8. Y. Lu, A. Zhong, Q. Li, B. Dong, Beyond finite layer neural networks: bridging deep architectures and numerical differential equations, Paper presented at: Proceedings of Machine Learning Research Proceedings of the 35th International Conference on Machine Learning, 2018.
9. R. T. Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, Neural ordinary differential equations, Paper present at :Annual Conference on Neural Information Processing Systems, 2018.
10. P. Meng, L. Su, W. Yin, S. Zhang, Solving a kind of inverse scattering problem of acoustic waves based on linear sampling method and neural network, *Alexandria Eng. J.*, **59** (2020), 1451–1462. <http://doi.org/10.1016/j.aej.2020.03.047>

11. W. Yin, W. Yang, H. Liu, A neural network scheme for recovering scattering obstacles with limited phaseless far-field data, *J. Comput. Phys.*, **417** (2020). <http://doi.org/10.1016/j.jcp.2020.109594>
12. Y. Guo, D. Hoemberg, G. Hu, J. Li, H. Liu, A time domain sampling method for inverse acoustic scattering problems, *J. Comput. Phys.*, **314** (2016), 647–660. <http://doi.org/10.1016/j.jcp.2016.03.046>
13. D. Zhang, F. Sun, L. Lu, Y. Guo, A harmonic polynomial method with a regularization strategy for the boundary value problems of Laplace's equation, *Appl. Math. Lett.*, **79** (2018), 100–104. <http://doi.org/10.1016/j.aml.2017.12.003>
14. H. Liu, M. Petrini, L. Rondi, J. Xiao, Stable determination of sound-hard polyhedral scatterers by a minimal number of scattering measurements, *J. Differ. Equations*, **262** (2018), 1631–1670. <http://doi.org/10.1016/j.jde.2016.10.021>
15. H. Liu, X. Liu, Recovery of an embedded obstacle and its surrounding medium from formally determined scattering data, *Inverse Probl.*, **33** (2017), 1–20. <http://doi.org/10.1088/1361-6420/aa6770>
16. H. Liu, X. Liu, X. Wang, Y. Wang, On a novel inverse scattering scheme using resonant modes with enhanced imaging resolution, *Inverse Probl.*, **35** (2019). <http://doi.org/10.1088/1361-6420/ab2932>
17. W. Yin, J. Ge, P. Meng, F. Qu, A neural network method for the inverse scattering problem of impenetrable cavities, *Electron. Res. Arch.*, **28** (2020), 1123–1142. <http://doi.org/10.3934/era.2020062>
18. J. Xie, Y. chen, A numerical analysis method of fixed points and their stability in a complex dynamical systems, *CCAMMS*, (2019), 59–62.
19. W. Stephen, D. S. Mazel, Introduction to applied nonlinear dynamical systems and chaos, *Name J.*, **33** (1991), 81. <http://doi.org/10.1063/1.4822950>
20. G. Wang, Ordinary differential equations, *Higher Education Press*, (2013), 250–260.
21. E. Haber, L. Ruthotto, Stable architectures for deep neural networks, *Inverse Probl.*, **34** (2018). <http://doi.org/10.1088/1361-6420/aa9a90>
22. C. Bo, M. Chen, E. Haber, E. Chi, Antisymmetricrnn: a dynamical system view on recurrent neural networks, Paper present at: 7th International Conference on Learning Representations, 2019.
23. U. Ascher, L. Petzold, Computer methods for ordinary differential equations and differential-algebraic equations, in *Science press*, 2009.
24. D. Colton, Inverse acoustic and electromagnetic scattering theory, *Inverse Probl.*, **47** (2003), 67–110. <http://doi.org/10.1007/978-3-662-03537-5>
25. D. Colton, J. Coyle, P. Monk, Recent developments in inverse acoustic scattering theory, *SIAM*, **42** (2000), 369–414. <http://doi.org/10.1137/S0036144500367337>
26. A. Kirsch, New characterizations of solutions in inverse scattering theory, *Appl. Anal.*, **76** (2000), 319–350. <http://doi.org/10.1080/00036810008840888>
27. A. Kirsch, N. Grinberg, The factorization method for inverse problems, *Oxford University Press*, 2008.

28. F. Zeng, P. Suarez, J. Sun, A decomposition method for an interior inverse scattering problem, *Inverse Probl. Imaging*, **7** (2013), 291–303. <http://doi.org/10.3934/ipi.2013.7.291>
29. J. Li, H. Liu, W. Tsui, X. Wang, An inverse scattering approach for geometric body generation: a machine learning perspective, *Math. Eng.*, **1** (2019), 800–823. <http://doi.org/10.3934/mine.2019.4.800>
30. J. Li, H. Liu, J. Zou, Locating multiple multiscale acoustic scatterers, *Multiscale Model. Simul.*, **12** (2014), 927–952. <http://doi.org/10.1137/13093409X>
31. H. Liu, J. Zou, Uniqueness in an inverse acoustic obstacle scattering problem for both sound-hard and sound-soft polyhedral scatterers, *Inverse Probl.*, **22** (2016), 515–524. <http://doi.org/10.1088/0266-5611/22/2/008>
32. Q. V. Le, N. Jaitly, G. E. Hinton, A simple way to initialize recurrent networks of rectified linear units, *Comput. Sci.*, preprint, [arXiv:1504.00941](https://arxiv.org/abs/1504.00941).



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)