Neural Processing Letters (2024) 56:65
https://doi.org/10.1007/s11063-024-11543-x

Stability Analysis of Deep Belief Network: Based SD-AR Model for Nonlinear Time Series

Wenquan Xu[1] · Hui Hu[2]

Accepted: 16 January 2024

Abstract
As for nonlinear time series prediction, many different kinds of varying-coefficient models have been proposed and analysised in recent years. A kind of varying functional-coefficient autoregressive model, called the deep belief network-based state-dependent autoregressive (DBN-AR) model is considered in this paper. The stability conditions and existing conditions of limit cycle of the DBN-AR model are also studied. An especial designed parameter estimation method is used to identify the DBN-AR model. The DBN-AR model is used to predict the famous Canadian lynx data and Henon chaotic series, the prediction capability of the DBN-AR model is compared with other prediction models, the experimental results show that the DBN-AR model obtains better prediction accuracy.

Keywords Deep belief network · Stability analysis · Nonlinear time series · Varying-coefficient

1 Introduction

The parameters of varying-coefficient (or functional-coefficient) models may vary with the value of some variables, and a flexible structure is given to model nonlinear time series [1–3]. Up to date, this kind of model has been studied in many literatures, and many applications have been achieved by researchers. For example, a varying coefficient partially functional linear regression (VCPFLM) model was proposed by Peng et al. [4], and the problem of this functional parameter estimation in a VCPFLM is also be studied. Kim et al. [5] proposed a nonparametric bivariate varying coefficient generalized linear model to predict time series. Vu et al. [6] proposed an autoregressive based time varying (ARTV) mode to predict electricity demand in a short-term period. Cai et al. [7] applied the local linear regression technique for estimation of functional-coefficient regression models for time series data. A linear varying coefficient spatial autoregressive (AR) model is proposed in Ref. [8], and a real data is used

✉ Wenquan Xu
zyheijunma@163.com

1 School of Electronic Engineering and Intelligent Manufacturing, Anqing Normal University, Anqing 246001, Anhui, China

2 School of Teacher Education, Anqing Normal University, Anqing 246001, Anhui, China

Published online: 23 February 2024

to illustrate the feasible of the proposed methodology. In Ref. [9], a functional coefficient autoregressive model (FCAR) fit via spline-backfitted local linear (SBLL) smoothing is proposed to predict time series. In Ref. [10], Bussas et al. proposed a varying-coefficient model that the coefficicents changed smoothly in space and time. Tan et al. [11] developed an eigenvector spatial filtering based spatially vary coefficient (ESF-SVC) model to identify ground PM2.5 concentration. Many practical applications have been obtained by the varying-coefficient models, however, some researchers still believe that this research is just at the beginning [12].

The above mentioned varying-functional model can be traced back to state-dependent model proposed by Priestley[13], and given as follows.

$$y(t) = \rho(\mathrm{T}(t-1)) + \sum_{i=1}^{k} \theta_i(\mathrm{T}(t-1))y(t-i) + \sum_{j=1}^{\lambda} \gamma_j(\mathrm{T}(t-1))\varepsilon(t-j) + \varepsilon(t) \quad (1)$$

where $\{y(t), t = 1, 2, \cdots, N\}$ denote the given time series, N represents the length of the time series. k and λ represent positive integers, respectively. $\mathrm{T}(t-1) = (y(t-1), y(t-2), \cdots y(t-k), \varepsilon(t-1), \varepsilon(t-2), \cdots, \varepsilon(t-\lambda))^{\mathrm{T}}$ represents "the state vector", $\varepsilon(t)$ is a random variable. $\rho(\bullet)$, $\{\theta_i(\bullet)\}$ and $\left\{\gamma_j(\bullet)\right\}$ are structure with measurable functions form. Many familiar time series models are state-dependent model (1), just a few examples are given as follows.

Tong [14] proposed a threshold autoregressive (TAR) model, and used to predict nonlinear time series. The TAR model can be considered as a structure of state-dependent model (1), the model is given as follows.

$$y(t) = \sum_{i=1}^{k} \left(\phi_1^i y(t-1) + \phi_2^i y(t-2) + \cdots + \phi_p^i y(t-p) + \varepsilon^i(t)\right) \text{ if } \mathrm{y}(t-d) \in \Omega_i \quad (2)$$

where $\{\Omega_i\}$ form a partition of the real line.

The following random coefficient regression model is considered by Granger [15].

$$y(t) = \boldsymbol{\beta}(t)^{\mathrm{T}}\mathrm{K}(t) + \chi(t) \quad (3)$$

where $\{\chi(t)\}$ is a sequece of random variable with $E(\chi(t)) = 0$ and $Var(\chi(t)) = \delta^2$. If $\mathrm{K}(t) = (y(t-1), \cdots, y(t-p))^{\mathrm{T}}$, then model (3) is called the random coefficient autoregressive model.

The coefficients of the aboved model are constant, however, if a set of neural networks are used to approximate the coefficients of the state-dependent model, then the coefficients of the obtained model are variable. Therefore, if a set of deep belief networks (DBNs) are used to approximate the functional coefficients of the state-dependent autoregressive (SD-AR) model. The derived model, call the DBN network-based autoregressive (DBN-AR) model [16], takes the form

$$\begin{cases} y(t) = \psi_0(\mathbf{\Theta}(t-1)) + \sum_{i=1}^{p} \psi_i(\mathbf{\Theta}(t-1)) y(t-i) + \varepsilon(t) \\ \psi_j(\mathbf{\Theta}(t-1)) = \varphi\left(u_{1,j}^{\left(\Upsilon_r^{(j)}\right)}(t)\right) = \varphi\left(\mathbf{w}_{1,j}^{\left(\Upsilon_r^{(j)}\right)} \mathbf{h}_j^{\left(\Upsilon_r^{(j)}-1\right)}(t) + b_{1,j}^{\left(\Upsilon_r^{(j)}\right)}\right), j \in \{0, 1, 2, \cdots, p\} \\ \mathbf{h}_j^{(\ell_j)}(t) = \left(h_{1,j}^{(\ell_j)}(t), h_{2,j}^{(\ell_j)}(t), \cdots, h_{Q_{\ell_j}^{(j)},j}^{(\ell_j)}(t)\right)^{\mathrm{T}}, \ell_j \in \left\{1, 2, \cdots, \Upsilon_r^{(j)} - 1\right\} \\ h_{n_{\ell_j}^{(j)},j}^{(\ell_j)}(t) = \varphi\left(u_{n_{\ell_j}^{(j)},j}^{(\ell_j)}(t)\right) = \varphi\left(\mathbf{w}_{n_{\ell_j}^{(j)},j}^{(\ell_j)} \mathbf{h}_j^{(\ell_j-1)}(t) + b_{n_{\ell_j}^{(j)},j}^{(\ell_j)}\right),\ n_{\ell_j}^{(j)} \in \left\{1, 2, \cdots, Q_{\ell_j}^{(j)}\right\} \\ \mathbf{w}_{n_{\ell_j}^{(j)},j}^{(\ell_j)} = \left(w_{n_{\ell_j}^{(j)},1,j}^{(\ell_j)}, w_{n_{\ell_j}^{(j)},2,j}^{(\ell_j)}, \cdots, w_{n_{\ell_j}^{(j)},Q_{\ell_j-1}^{(j)},j}^{(\ell_j)}\right),\ Q_0^{(j)} = p \\ \mathbf{h}_j^{(0)}(t) = \mathbf{\Theta}(t-1) = (y(t-1), y(t-2), \cdots, y(t-p))^{\mathrm{T}} \end{cases} \tag{4}$$

where p represents the input order of DBN-AR model, $\Upsilon_r^{(j)}$ represents the total number of layers in the j - th DBN module. $\mathbf{w}_{n_{\ell_j}^{(j)},j}^{(\ell_j)}$ denotes the weight matrix between layer ℓ_j and layer $\ell_j - 1$ in the j - th DBN module, $Q_{\ell_j}^{(j)}$ represents the number of nodes on layer ℓ_j in the j - th DBN module,$\mathbf{h}_j^{(\ell_j)}(t)$ represents the output value of ℓ_j - th hidden layer in the j - th DBN module, $\left(b_{1,j}^{(\ell_j)}, b_{2,j}^{(\ell_j)}, \cdots, b_{Q_{\ell_j}^{(j)},j}^{(\ell_j)}\right)$ represents the bias of ℓ_j - th hidden layer in the j - th DBN module, $\psi_j(\mathbf{\Theta}(t-1))$ are the state-dependent functional coefficients which are all composed of DBN networks, and also represents the output of the j - th DBN module, i.e. the j - th state-dependent coefficient of the DBN-AR model (4). The DBN-AR model (4) deals with the nonlinear process by decomposing the state space into a large number of small segments, and regards the nonlinear process in each segment as "local linearization". The functional coefficient $\psi_j(\mathbf{\Theta}(t-1))$ in DBN-AR model (4) are constantly changing due to different input signals. The DBN-AR model gives the flexibility in characterizing complex dynamics with the DBN because of it contains the DBN as a component. Xu et al. [17] extended the DBN-AR model to the case where there are several exogenous variables (DBN-ARX model) to the system. Compared with other time-varying linear models, the parameters of the DBN-AR(X) model are estimated off-line.

In this paper, some probalistic properties, the identification method and application of the DBN-AR model are studied. Stability conditions and existing conditions of limit cycle of the DBN-AR model are studied in Sect. 2. The estimation method for the varying-coefficient DBN-AR model is given in Sect. 3. To compare the prediction capabilities of the DBN-AR model with some other time series prediction models, the DBN-AR model is applied to the famous Canadian lynx and the Henon chaotic series prediction in Sect. 4, the prediction results of the DBN-AR model is better than other models. Section 5 offers a conclusion.

2 Stability Analysis and Limit Cycle of the DBN-AR Model

The stability conditions and existing conditions of limit cycle of the DBN-AR model is discussed in this section. As for a given nonlinear time series model, we usually first described the time series as a vector-valued Markov chain. Then, we deduce the stability of the model by proving whether the corresponding Markov chain is ergodic. Then, a Markov Chain in the p-dimensional Euclidean is established for the DBN-AR model (4). Define.

$\mathbf{\Theta}(t-1) = (y(t-1), y(t-2), \cdots, y(t-p))^{\mathrm{T}}$, $\boldsymbol{\varepsilon}(t) = (\varepsilon(t), 0, \cdots, 0)^{\mathrm{T}}$.

Set $\boldsymbol{\alpha}(\mathbf{\Theta}(t-1)) = (\psi_0(\mathbf{\Theta}(t-1)), 0, \cdots, 0)^{\mathrm{T}}$

$$\mathbf{\Gamma}(\mathbf{\Theta}(t-1)) = \begin{pmatrix} \psi_1(\mathbf{\Theta}(t-1)) & \psi_1(\mathbf{\Theta}(t-1)) & \cdots & \psi_{p-1}(\mathbf{\Theta}(t-1)) & \psi_p(\mathbf{\Theta}(t-1)) \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

Then, the DBN-AR model (4) can be rewritten as follows

$$\mathbf{Y}(t) = \boldsymbol{\beta}(\mathbf{\Theta}(t-1)) + \mathbf{\Gamma}(\mathbf{\Theta}(t-1))\mathbf{Y}(t-1) + \boldsymbol{\varepsilon}(t) \tag{5}$$

For convenience of explanation, the DBN-AR model (4) is represented in a more general nonlinear autoregressive (AR) model.

$$y(t) = \nu(y(t-1), y(t-2), \cdots, y(t-p)) + \varepsilon(t) \tag{6}$$

As for $\mathbf{y} = (y(t-1), y(t-2), \cdots, y(t-p))^{\mathrm{T}}$, set $\|\mathbf{y}\| = \sqrt{y^2(t-1) + y^2(t-2) + \cdots + y^2(t-p)}$.

Lemma 2.1 [18]. $\nu(\bullet)$ in model (6) is supposed as a measurable function and bounded over bounded sets, and $E\varepsilon(t) = 0$. If the following condition is satisfied, the model (6) is geometrically ergodic.

$$\lim_{\|\mathbf{y}\|\to\infty} \frac{\left|\nu(\mathbf{\Theta}(t-1)) - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{\Theta}(t-1)\right|}{\|\mathbf{y}\|} = 0 \tag{7}$$

where $\boldsymbol{\beta} = \left(\beta_1, \beta_2, \cdots, \beta_p\right)^{\mathrm{T}}$ satisfies the following given condition

$$z^p - \beta_1 z^{p-1} - \cdots - \beta_p \neq 0 \text{ for all } |z| \geq 1 \tag{8}$$

Lemma 2.2 [19]. Assume that $\nu(\bullet)$ in model (6) is bounded over bounded sets, and $\varepsilon(t)$ is positive density everywhere. Then $\{\mathbf{\Theta}(t-1)\}$ satisfying (6) is aperiodic.

Lemma 2.3 [20]. $\{\mathbf{\Theta}(t-1)\}$ is considered as a μ-irreducible Markov Chain on a normed topological space. $\Pr(\mathbf{\Theta}(t-1), \bullet)$ represent the transition probability, and if $\Pr(\mathbf{\Theta}(t-1), \bullet)$ is strongly continuous, then a sufficient condition for the ergodicity of geometric equations is given [1], the existence of a compact set k and a constant $0 < \lambda < 1$ are also given.

$$E(\|\mathbf{\Theta}(t)\| | \mathbf{\Theta}(t-1) = \mathbf{y}) < \begin{cases} \infty, & \mathbf{y} \in \mathbf{K} \\ \lambda\|\mathbf{y}\|, & \mathbf{y} \notin \mathbf{K} \end{cases} \tag{9}$$

Lemma 2.4 [21]**.** $\{\mathbf{\Theta}(t-1)\}$ is considered as an aperiodic Markov chain. The original series $\{\mathbf{\Theta}(t-1)\}$ is geometric ergodic, and a fixed positive integer h is existed. Then the geometric ergodicity of the original sequence $\{\mathbf{\Theta}(t-1)\}$ can be deduced from the geometric ergodicity of the subsequence $\{\mathbf{\Theta}_h(t-1)\}$.

According to Lemmas 2.1–2.4, the following theorems can be obtained.

Theorem 2.1. Assume that the density function of $\varepsilon(t)$ in the DBN-AR model (4) is positive everywhere on the real line $\Re$. If all roots of the characteristic function defined as follows.

$$z^p - \omega_1 z^{p-1} - \cdots \omega_{p-1} z - \omega_p = 0 \tag{10}$$

If all roots of Eq. (10) are inside the unit circle, then the DBN-AR process in model (4) is geometrically ergodic.

Proof Set $\boldsymbol{\beta} = \left(\omega_1, \omega_2, \cdots, \omega_p\right)^{\mathrm{T}}$, then.

$$\lim_{\|\mathbf{y}\|\to\infty} \frac{\left|\nu(\boldsymbol{\Theta}(t-1)) - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Theta}(t-1)\right|}{\|\mathbf{y}\|} = \lim_{\|\mathbf{y}\|\to\infty} \frac{\left|\psi_0(\boldsymbol{\Theta}(t-1)) + \sum_{i=1}^{p}\psi_i(\boldsymbol{\Theta}(t-1))y(t-i) - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Theta}(t-1)\right|}{\|\mathbf{y}\|}$$

$$= \lim_{\|\mathbf{y}\|\to\infty} \frac{\left|\varphi\left(\mathbf{w}_{1,0}^{\left(\Upsilon_r^{(0)}\right)}\mathbf{h}_0^{\left(\Upsilon_r^{(0)}-1\right)}(t) + b_{1,0}^{\left(\Upsilon_r^{(0)}\right)}\right) + \sum_{i=0}^{p}\varphi\left(\mathbf{w}_{1,i}^{\left(\Upsilon_r^{(i)}\right)}\mathbf{h}_i^{\left(\Upsilon_r^{(i)}-1\right)}(t) + b_{1,i}^{\left(\Upsilon_r^{(i)}\right)}\right)y(t-i)\right|}{\|\mathbf{y}\|}$$

$$\lim_{\|\mathbf{y}\|\to\infty} \frac{\left|\varphi\left(\mathbf{w}_{1,0}^{\left(\Upsilon_r^{(0)}\right)}\mathbf{h}_0^{\left(\Upsilon_r^{(0)}-1\right)}(t) + b_{1,0}^{\left(\Upsilon_r^{(0)}\right)}\right)\right|}{\|\mathbf{y}\|} = 0$$

$$\lim_{\|\mathbf{y}\|\to\infty} \frac{\left|\sum_{i=0}^{p}\varphi\left(\mathbf{w}_{1,i}^{\left(\Upsilon_r^{(i)}\right)}\mathbf{h}_i^{\left(\Upsilon_r^{(i)}-1\right)}(t) + b_{1,i}^{\left(\Upsilon_r^{(i)}\right)}\right)y(t-i)\right|}{\|\mathbf{y}\|} = 0$$

Then, we have

$$\lim_{\|\mathbf{y}\|\to\infty} \frac{\left|\nu(\boldsymbol{\Theta}(t-1)) - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Theta}(t-1)\right|}{\|\mathbf{y}\|} = 0 \tag{11}$$

Therefore, according to Lemma 2.1, the DBN-AR model is geometric ergodic.

Theorem 2.2. Suppose that the value of $\varepsilon(t)$ in the DBN-AR model (4) is positive everywhere on the real line $\Re$. If all the roots of the following characteristic function are inside the unit circle, then the DBN-AR process in (4) is geometrically ergodic.

$$z^p - \kappa_1 z^{p-1} - \cdots - \kappa_p = 0 \tag{12}$$

Proof As for model (5), according to Lemma (2.2), $\{\boldsymbol{\Theta}(t)\}$ is a $\mu-$ irreducible and aperiodic Markov chain. Let.

$$\mathbf{A} = \begin{pmatrix} \kappa_1 & \kappa_2 & \cdots & \kappa_{p-1} & \kappa_p \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

and $\mathbf{I}_p$ represents the identity matrix of order p. And then, the eigenpolynomial of matrix $\mathbf{A}$ is described as follows.

$$|z\mathbf{I} - \mathbf{A}| = z^p - \kappa_1 z^{p-1} - \cdots - \kappa_p \tag{13}$$

Therefore, all the roots of the characteristic function (12) are eigenvalues of matrix $\mathbf{A}$. Let $z_{\max}$ represent the maximum eigenvalue of $\mathbf{A}$.

$$\begin{aligned}&\mathbf{E}(\|\mathbf{Y}(t+h)\||\mathbf{Y}(t)=\boldsymbol{\Theta}(t-1))\\&=\mathbf{E}\left(\left\|\prod_{i=0}^{h-1}\boldsymbol{\Gamma}(\mathbf{Y}(t+h))\mathbf{Y}(t)+\sum_{i=1}^{h}\left[\prod_{j=i}^{h-1}\boldsymbol{\Gamma}(\mathbf{Y}(t+i))\right]\beta(\mathbf{Y}(t+i-1))+\sum_{i=1}^{h}\left[\prod_{j=i}^{h-1}\boldsymbol{\Gamma}(\mathbf{Y}(t+i))\right]\varepsilon(t+i)\right\||\mathbf{Y}(t)=\boldsymbol{\Theta}(t-1)\right)\end{aligned} \tag{14}$$

As for any given vector $\boldsymbol{\tau}=\left(\tau_1,\tau_2,\cdots,\tau_p\right)^{\mathrm{T}}$, then, we have

$$\|\boldsymbol{\Gamma}(\boldsymbol{\Theta}(t-1))\boldsymbol{\tau}\|\leq\|\mathbf{A}|\boldsymbol{\tau}|\| \tag{15}$$

where $|\boldsymbol{\tau}|=\left(|\tau_1|,|\tau_2|,\cdots,\left|\tau_p\right|\right)^{\mathrm{T}}$.

we can obtain the following equation when repeate apply Eq. (15) to Eq. (13, 14).

$$\begin{aligned}\mathbf{E}(\|\mathbf{Y}(t+h)\||\mathbf{Y}(t)=\boldsymbol{\Theta}(t-1))\leq&\left\|\mathbf{A}^h|\boldsymbol{\Theta}(t-1)|\right\|+E\left\|\sum_{i=1}^{h}\mathbf{A}^{h-i}|\boldsymbol{\beta}(\mathbf{Y}(t+i-1))|\right\|\mathbf{Y}(t)\\&=\|\boldsymbol{\Theta}(t-1)\|+E\left\|\sum_{i=1}^{h}\mathbf{A}^{h-i}|\boldsymbol{\varepsilon}(t+i)|\right\|\leq\left\|\mathbf{A}^h\right\|\|\boldsymbol{\Theta}(t-1)\|+E\left\|\sum_{i=1}^{h}\mathbf{A}^{h-i}|\boldsymbol{\beta}(\mathbf{Y}(t+i-1))|\right\|\mathbf{Y}(t)\\&=\|\boldsymbol{\Theta}(t-1)\|+E\left\|\sum_{i=1}^{h}\mathbf{A}^{h-i}|\boldsymbol{\varepsilon}(t+i)|\right\|\leq\delta\|\boldsymbol{\Theta}(t-1)\|+E\left\|\sum_{i=1}^{h}\mathbf{A}^{h-i}|\boldsymbol{\beta}(\mathbf{Y}(t+i-1))|\right\|\mathbf{Y}(t)\\&=\|\boldsymbol{\Theta}(t-1)\|+E\left\|\sum_{i=1}^{h}\mathbf{A}^{h-i}|\boldsymbol{\varepsilon}(t+i)|\right\|\end{aligned} \tag{16}$$

where $0<\delta<1$ is a positive constant.

According to the definition of DBN-AR model (4) and suppose the modeling error of DBN-AR model is bounded. We find a sufficiently large $\mathbb{C}>0$ such that when $\|\boldsymbol{\Theta}(t-1)\|>\mathbb{C}$.

$$\begin{aligned}\delta\|\boldsymbol{\Theta}(t-1)\|+E&\left\|\sum_{i=1}^{h}\mathbf{A}^{h-i}|\boldsymbol{\beta}(\mathbf{Y}(t+i-1))||\mathbf{Y}(t)=\boldsymbol{\Theta}(t-1)\right\|\\&+E\left\|\sum_{i=1}^{h}\mathbf{A}^{h-i}|\varepsilon(t+i)|\right\|<\rho\|\boldsymbol{\Theta}(t-1)\|\end{aligned} \tag{17}$$

where $\rho\in(\delta,1)$. Therefore, the compact set $\mathbf{K}=\{\boldsymbol{\Theta}(t-1):\|\boldsymbol{\Theta}(t-1)\|\leq\mathbf{M}\}$ satisfies that when $\boldsymbol{\Theta}(t-1)\notin\mathbf{K}$,$E(\|y(t+h)\||y(t)=\boldsymbol{\Theta}(t-1)<\rho\|\boldsymbol{\Theta}(t-1)\|)$. According to Lemma 2.3 and 2.4, $\{y(t)\}$ is geometric ergodic.

3 Identification of the DBN-AR Model

In order to identify the parameters of DBN-AR model, the indentification process can be divided into the pre-training learning and the local weight adjustment. Before the pre-training learning stage, the input order of DBN-AR model is select by the Akaike Information Criterion (AIC). The value of AIC can be calculated by Eq. (18):

$$\mathrm{AIC}=N\log\vartheta_e^2+2(\varpi+1),N>>p \tag{18}$$

where ϑ_e represents the value of modeling residual, p represents the input order of the DBN-AR model, ϖ represents the total number of the parameters to be estimated in the DBN-AR model, and N represents the length of given time series data.

At first, the original data series are normalized to [0,1]. In the pre-training learning stage of the DBN-AR model, the target values of functional coefficients $\psi_j(\mathbf{\Theta}(t-1))$, $(j = 0, 1, \cdots, p)$ are calculated by a pseudo inverse matrix, and the reference values $\{\psi_0, \psi_1, \cdots, \psi_p\}$ for the each DBN module are obtained at sample instant, and the calculate equation is given as follows.

$$y(p+i) = \mathrm{E}_{p+i-1}\boldsymbol{\psi}_{p+i-1},\ i \in \{1, 2, \cdots, N-p\} \tag{19}$$

where

$$\begin{cases} \mathrm{E}_{p+i-1} = (1, y(p+i-1), y(p+i-2), \cdots, y(i)) \\ \boldsymbol{\psi}_{p+i-1} = \left(\psi_0(\mathbf{\Theta}(p+i-1)), \psi_1(\mathbf{\Theta}(p+i-1)), \psi_2(\mathbf{\Theta}(p+i-1)), \cdots, \psi_p(\mathbf{\Theta}(p+i-1))\right)^{\mathrm{T}} \end{cases}$$

$\boldsymbol{\psi}_{p+i-1}$ represents the reference outputs of the DBN module at time $p+i-1$, and E_{p+i-1} denotes the correlation coefficients in the DBN-AR model at time $p+i-1$. The target values of the each DBN module can be calculated by the following equation

$$\boldsymbol{\psi}_{p+i-1} = \mathrm{E}^{+}_{p+i-1}y(p+i) \tag{20}$$

where, E^{+}_{p+i-1} represents the pseudo inverse matrix of E_{p+i-1}. Then unsupervised pre-training proposed by Hinton [22] is used to train each DBN module. The prediction values of DBN-AR model in pre-training stage are obtained, and given as follows.

$$\hat{y}(t) = \hat{\psi}_0(\mathbf{\Theta}(t-1)) + \sum_{i=1}^{p}\hat{\psi}_i(\mathbf{\Theta}(t-1))y(t-i), \quad t = p+1, p+2, \cdots, N \tag{21}$$

where $\hat{y}(t)$ represents the prediction values of the DBN-AR model, and $\hat{\psi}_j(\mathbf{\Theta}(t-1))$ $(j = 0, 1, \cdots, p)$ are the output values of the DBN modules after pre-training of the DBNs. The obtained values of $\hat{\psi}_j(\mathbf{\Theta}(t-1))$ $(j = 0, 1, \cdots, p)$ are varies with the input signal $\mathbf{\Theta}(t-1)$ obviously, because DBN-AR model is a variable functional-coefficient model.

After the DBN-AR model (4) is pre-trained, the modeling error in the pre-training stage is obtained, and given as follows.

$$\xi(t) = y(t) - \hat{y}(t), \ \ t = p+1, p+2, \cdots, N \tag{22}$$

Then, $\xi(t)$ is used to fine tune the parameters of DBN-AR model (4) by the especially designed BP algorithm that is given in Appendix A until the mean square deviation $\frac{1}{N-p}\sum\limits_{t=p+1}^{N}\xi^2(t)$ reaches minimum [16]. In the fine-tuning stage, the objective function is given as follows

$$\begin{aligned} \mathrm{H}(t) &= \frac{1}{2}\xi^2(t) = \frac{1}{2}\left(y(t) - \hat{y}(t)\right)^2 \\ &= \frac{1}{2}\left(y(t) - \hat{\psi}_0(\mathbf{\Theta}(t-1)) - \sum_{i=1}^{p}\hat{\psi}_i(\mathbf{\Theta}(t-1))y(t-i)\right)^2, \ t = p+1, p+2, \cdots, N \end{aligned} \tag{23}$$

where $y(t)$ represent the actual output of DBN-AR model (4), and $\hat{y}(t)$ represent the predicted output of DBN-AR model (4).

As for all of the training data sets, the parameters of DBN-AR model (4) are fine tuned by Eq. (A.20) (see Appendix A), and then the final obtained updated model parameters are used to recaculate the output of the DBN-AR model, the final predicted value is calculated by the following equation.

$$\tilde{y}(t) = \tilde{\psi}_0(\mathbf{\Theta}(t-1)) + \sum_{i=1}^{p} \tilde{\psi}_i(\mathbf{\Theta}(t-1))y(t-i) \tag{24}$$

If the value of mean square error, $\mathrm{MSE} = \frac{1}{N-p} \sum_{t=p+1}^{N} (y(t) - \tilde{y}(t))^2$ is small enough or smaller than a given value, the parameter's fine-tuning process is stop, otherwise, Eq. (A. 20) is continued to fine tune the parameters of DBN-AR model. Finally, the final prediction value of the output of DBN-AR model $\tilde{y}(t)$ is obtained after denormalized the obtained data series.

4 Case Study

4.1 Canadian Lynx

The data of Canadian lynx is a famous data that have attracted a lot of attention of many researchers. The data of lynx contains the number of lynx trapped per year in the Mackenzie River District of North–West Canada from 1821 to 1934[1]. The data of lynx are plotted in Fig. 1.

To compare the prediction performance of different models or methods in the other literatures, the first 100 data points are used to estimate the parameters of DBN-AR model, and remaining 14 data points are used to test the prediction performance of DBN-AR model. The same as other literatures, the logarithms (to the base 10) of the data of lynx are also used in this paper. The identification procedure is described in Sect. 3 to optimize the parameters of DBN-AR model. Take the same input order as the other reference, the input order is choosen as 2, that is $p = 2$. The structure parameters of each DBN module in model (4) is given in Table 1. In the pre-training stage, the number of iteration in the training of DBN module 1, DBN module 2 and DBN module 3 are 5600, 1100 and 50,000, respectively. The number of iteration in the fine tuning stage is choosen as 50. The following DBN-AR model is used to

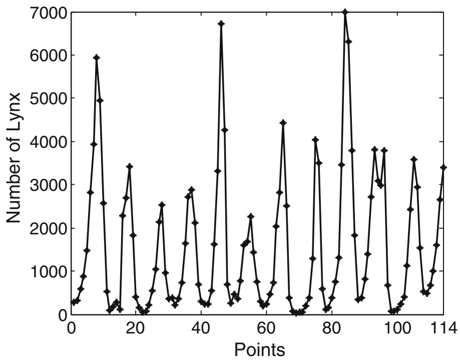


Fig. 1 The data of Canadian lynx

Table 1 Structure parameters of each DBN module in model (25)

j	Parameters of the j-th DBN module
0	$N_r^{(0)}=3;\ Q_0^{(0)}=2;\ Q_1^{(0)}=25;\ Q_2^{(0)}=55; Q_3^{(0)}=1$
1	$N_r^{(1)}=2;\ Q_0^{(1)}=2;\ Q_1^{(1)}=55; Q_2^{(1)}=1$
2	$N_r^{(2)}=3;\ Q_0^{(2)}=2;\ Q_1^{(2)}=25;\ Q_2^{(2)}=55; Q_3^{(2)}=1$

predict the one-step-ahead prediction output of this lynx data series.

$$\begin{cases} y(t)=\psi_0(\mathbf{\Theta}(t-1))+\sum_{i=1}^{2}\psi_i(\mathbf{\Theta}(t-1))y(t-i)+\varepsilon(t) \\ \psi_j(\mathbf{\Theta}(t-1))=\varphi\left(u_{1,j}^{\left(\Upsilon_r^{(j)}\right)}(t)\right)=\varphi\left(\mathbf{w}_{1,j}^{\left(\Upsilon_r^{(j)}\right)}\mathbf{h}_j^{\left(\Upsilon_r^{(j)}-1\right)}(t)+b_{1,j}^{\left(\Upsilon_r^{(j)}\right)}\right), j\in\{0,1,2\} \\ \mathbf{h}_j^{(\ell_j)}(t)=\left(h_{1,j}^{(\ell_j)}(t),h_{2,j}^{(\ell_j)}(t),\cdots,h_{Q_{\ell_j}^{(j)},j}^{(\ell_j)}(t)\right)^{\mathrm{T}}, \ell_j\in\left\{1,2,\cdots,\Upsilon_r^{(j)}-1\right\} \\ h_{n_{\ell_j}^{(j)},j}^{(\ell_j)}(t)=\varphi\left(u_{n_{\ell_j}^{(j)},j}^{(\ell_j)}(t)\right)=\varphi\left(\mathbf{w}_{n_{\ell_j}^{(j)},j}^{(\ell_j)}\mathbf{h}_j^{(\ell_j-1)}(t)+b_{n_{\ell_j}^{(j)},j}^{(\ell_j)}\right),\ n_{\ell_j}^{(j)}\in\left\{1,2,\cdots,Q_{\ell_j}^{(j)}\right\} \\ \mathbf{w}_{n_{\ell_j}^{(j)},j}^{(\ell_j)}=\left(w_{n_{\ell_j}^{(j)},1,j}^{(\ell_j)},w_{n_{\ell_j}^{(j)},2,j}^{(\ell_j)},\cdots,w_{n_{\ell_j}^{(j)},Q_{\ell_j-1}^{(j)},j}^{(\ell_j)}\right),\ Q_0^{(j)}=2 \\ \mathbf{h}_j^{(0)}(t)=\mathbf{\Theta}(t-1)=(y(t-1),y(t-2))^{\mathrm{T}} \end{cases} \tag{25}$$

After the DBN-AR model is fine tuned, the target values and predicted values of the each DBN module for the testing data are given in Fig. 2, it can be seen from Fig. 2 that the predicted values can well fit the given target values after the DBN-AR model is fine tuned by the optimization method.

Table 2 gives the mean square error (MSE) values of the DBN-AR model and some other prediction models for the testing data series. It can be seen from Table 2 that the DBN-AR model obtains smaller prediction error compared with other models. Figure 3 shows comparison results of the ideal output and predicted value of the DBN-AR model for the testing data. It also can be seen from the Fig. 3 that the DBN-AR model obtains good prediction accuracy for the testing data in this paper.

Figure 4 gives the poles of the DBN-AR model changing with the input signal $\mathbf{\Theta}(t-1)$. The different color in Fig. 4 represents the different pole. It can be seen from Fig. 4 that the dynamics of the nonlinear system vary obviously with the input signal $\mathbf{\Theta}(t-1)$. It also can be seen from Fig. 4 that the DBN-AR model is stable, because all the characteristic roots are inside the unit circle. The result of experiment is consistent with the theory which described in Sect. 3.

Table 3 gives the one-step ahead absolute prediction errors of the last 12 data points from different varying coefficient models. From Table 3, we can see that the DBN-AR model obtains smaller AAPE than TAR model and RBF-AR model, and the DBN-AR model obtains very similar AAPE with that of the FAR model. Figure 4 gives the poles of the DBN-AR model varying with the variation of the input signal $\mathbf{\Theta}(t-1)$. The different color in Fig. 4

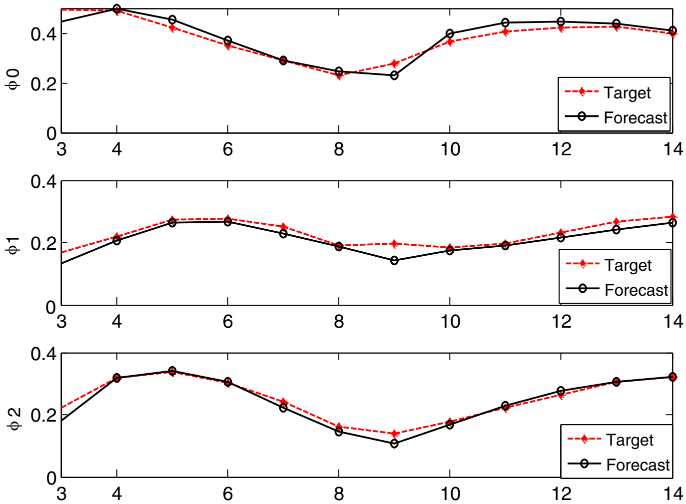
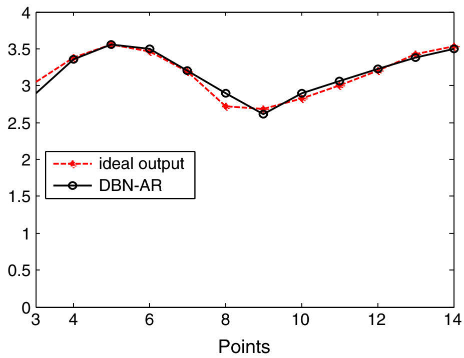


Fig. 2 Comparison of target values and forecast values of the each DBN module for the testing data

Table 2 Comparison results among different prediction models for the testing data

	Model	MSE
Zhang[23]	ANN	0.0205
Zhang[23]	ARIMA–ANN	0.0172
Kajtani et al.[24]	FFNN	0.0130
Aladag et al. [25]	SARIMA–ERNN	0.009
RBF-AR[1]	RBF-AR	0.0073
This paper	DBN-AR	0.0061

Fig. 3 Comparison results of ideal output and DBN-AR model for the testing data



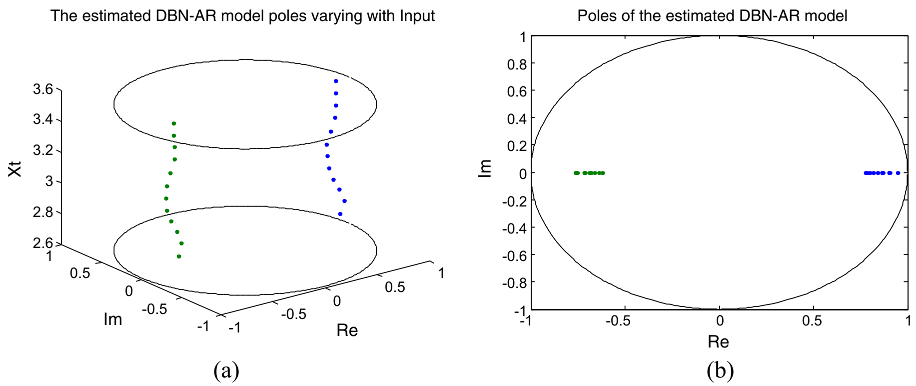


Fig. 4 Model's poles changing with the input signal

Table 3 The average absolute prediction error of different functional-coefficient models for the Canadian lynx data

Year	Vaule	DBN-AR model	RBF-AR(2,1,2) model [1]	FAR model [7]	TAR(2) model[7]
1923	3.054	0.156	0.201	0.157	0.187
1924	3.386	0.032	0.017	0.012	0.035
1925	3.553	0.010	0.017	0.021	0.014
1926	3.468	0.035	0.023	0.008	0.022
1927	3.187	0.020	0.118	0.085	0.059
1928	2.723	0.170	0.043	0.055	0.075
1929	2.686	0.072	0.174	0.135	0.273
1930	2.821	0.071	0.025	0.016	0.026
1931	3.000	0.065	0.040	0.017	0.030
1932	3.201	0.030	0.020	0.007	0.060
1933	3.424	0.043	0.056	0.089	0.076
1934	3.531	0.028	0.060	0.053	0.072
AAPE		0.0608	0.066	0.055	0.073

denotes the different pole. It can be seen from Fig. 4 that the DBN-AR model's dynamic behavior varies with the input signal $\boldsymbol{\Theta}(t-1)$ obviously.

4.2 Henon Chaotic Series Prediction

In this section, the DBN-AR model is used to predict the Henon chaotic system with one delay and two sensitive parameters generated by the following equation. Many researchers have analyzed this data series, and achieved a certain prediction accuracy [26–30].

$$y_p(t+1) = -P \bullet y_p^2(t) + Q \bullet y_p(t-1) + 1 \tag{26}$$

Table 4 Structure parameters of each DBN module in model (25)

j	Parameters of the j-th DBN module
0	$N_r^{(0)} = 3;\ Q_0^{(0)} = 2;\ Q_1^{(0)} = 9;\ Q_2^{(0)} = 25; Q_3^{(0)} = 1$
1	$N_r^{(1)} = 2;\ Q_0^{(1)} = 2;\ Q_1^{(1)} = 9;\ Q_2^{(1)} = 15; Q_3^{(1)} = 1$
2	$N_r^{(2)} = 3;\ Q_0^{(2)} = 2;\ Q_1^{(2)} = 25;\ Q_2^{(2)} = 55; Q_3^{(2)} = 1$

where, $P = 1.4$ and $Q = 0.3$, produces a chaotic attractor. The initial states $\left[y_p(1), y_p(0)\right] = [0.4, 0.4]$ generate 2000 patterns, and the first 1000 patterns are used as the training data sets, the remaining 1000 patterns are used as the testing data sets.

For fair comparison, the same as other references [26–30], the order of input is choosen as 2. The parameters of each DBN module in model (4) is given in Table 4. In the pre-training stage, the number of iteration in the training of first DBN module and second DBN module are 5000, and the number of iteration of the third DBN module is 10,500. The number of iteration in the fine tuning stage is choosen as 130,000. Then the obtained DBN-AR model is used to predict the one-step-ahead prediction output of this chaotic series. Figure 5 gives the phase plot of the ideal and DBN-AR predicted results for the test patterns.

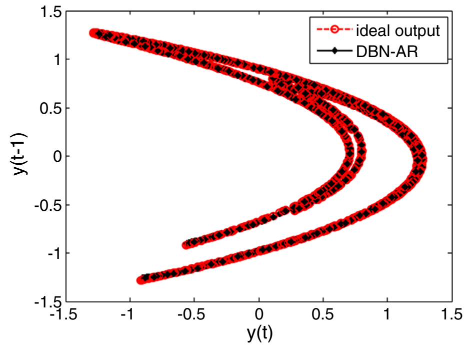


Fig. 5 Results of the phase plot for the chaotic system (red) and DBN-AR model (black)

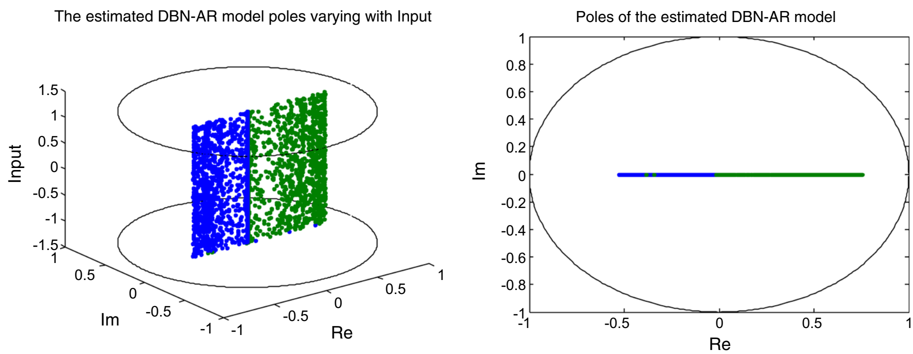


Fig. 6 Poles changing of DBN-AR model; different colour denotes different pole

Table 5 Performance of DBN-AR model and other prediction models for chaotic sequence prediction

Algorithm	Training RMSE	Testing RMSE
RFNN[28]	0.463	0.191
WRFNN[29]	0.469	0.188
TRFN-S[27]	0.028	0.027
RSEFNN-LF[30]	0.032	0.023
IRSFNN(TSK)[31]	0.017	0.015
IRSFNN(Ful)[31]	0.016	0.014
IOA-RRBFNN[26]	0.0041	0.0083
DBN-AR	0.0039	0.0040

Figure 6 represents the moving tracks of the poles that change with the input variable in the DBN-AR model, which are the poles of the local linear model derived from the estimated DBN-AR model. It can be seen from Fig. 6 that the system's dynamic behavior varies with the input signal obviously. That is, the nonlinear characteristics of the Henon chaotic system are relatively strong.

Table 5 shows the training and testing RMSEs of the DBN-AR model. It can be seen from Table 5 that the DBN-AR model achieves greater learning performance. For a meaningful comparison, the compared models include a recurrent FNN, a wavelet recurrent FNN, a TSK-TRFN, a recurrent self-evolving FNN with local feedback and an interactively recurrent self-evolving fuzzy neural network (IRSFNN), and so on. Table 5 shows that the DBN-AR model obtains the best performance.

5 Conclusion

The DBN-AR model belongs to a class of varying coefficient models. The model combines the advantage of the state-dependent AR model in the description of nonlinear dynamics and the merit of the DBN in function approximation. The stability conditions are discussed in this paper. The optimization method is proposed to estimate the parameters of the DBN-AR model. The prediction performance of DBN-AR model was compared with other models by applying the model to the famous Canadian lynx data and Henon chaotic series. The result of the experimental shows that the DBN-AR modeling approach exhibits better results than other time series datas.

Acknowledgements This research was supported by the key projects of natural science research in colleges and universities of Anhui Province (Grant No. KJ2020A0508). Humanities and social sciences research project of Anhui Province (Grant No. SK2021A0374). Key research and development plan of Anhui Province (Grant No. 202104a05020050).

Author contributions WX wrote the main manuscript text and HH did some experiments.

Declarations

Conflict of 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.

Appendix A. Fine Tuning Procedure of DBN-AR Model [16]

The following Equation is used as the objective function in the fine-tuning stage according to formula (23).

$$\begin{aligned}\mathrm{H}(t) &= \frac{1}{2}\xi^2(t) = \frac{1}{2}\big(y(t) - \hat{y}(t)\big)^2 \\ &= \frac{1}{2}\left(y(t) - \hat{\psi}_0(\boldsymbol{\Theta}(t-1)) - \sum_{i=1}^{p}\hat{\psi}_i(\boldsymbol{\Theta}(t-1))y(t-i)\right)^2, \ t = p+1, p+2, \cdots, N\end{aligned} \tag{A.1}$$

where $y(t)$ and $\hat{y}(t)$ represent the real output and prediction output of DBN-AR model, respectively.

All the parameters of DBN-AR model are fine tuned by the especially designed gradient descent method. As for the neuron in the $N_r^{(j)}$-th layer, the parameters are updated by the gradient, and calculated by Eq. (24), the calculated process is given as follows.

$$\begin{aligned}\frac{\partial \mathrm{H}(t)}{\partial w^{\left(\Upsilon_r^{(j)}\right)}_{1,n^{(j)}_{\Upsilon_r^{(j)}-1},j}} &= \frac{\partial \mathrm{H}(t)}{\partial \xi(t)}\frac{\partial \xi(t)}{\partial \hat{y}(t)}\frac{\partial \hat{y}(t)}{\partial \hat{\psi}_j(t)}\frac{\partial \hat{\psi}_j(t)}{\partial u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}(t)}\frac{\partial u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}(t)}{\partial w^{\left(\Upsilon_r^{(j)}\right)}_{1,n^{(j)}_{\Upsilon_r^{(j)}-1},j}} \\ &= -\,\xi(t)a(t-j)\varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}(t)\right)h^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}(t) \\ &= \varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}(t)\right)c(t-j)h^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)\end{aligned} \tag{A.2}$$

where

$$a(t-j) = y(t-j), \ j = 1, 2, \cdots, p; \ a(t) = 1$$

$c(t-j) = -\xi(t)a(t-j), \ j = 0, 1, 2, \cdots, p$

and $\varphi'(u)$ denote the derivative of $\varphi(u)$ with respect to u. Let

$$\delta^{\left(\Upsilon_r^{(j)}\right)}_{1,j}(t) = \varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}\right)c(t-j) \tag{A.3}$$

Eq. (A. 3) denote the local gradient with respect to neuron of the last layer in the j - th DBN module. Hence, Eq. (A.2) can be rewritten, and expressed as follows

$$\frac{\partial \mathrm{H}(t)}{\partial w_{1,n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r\right)}} = \delta_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t) h_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t) \tag{A.4}$$

Similarly, Eq. (A. 5) can be obtained by the same method

$$\begin{aligned}\frac{\partial \mathrm{H}(t)}{\partial b_{1,j}^{\left(\Upsilon^{(j)}_r\right)}} &= \frac{\partial \mathrm{H}(t)}{\partial \xi(t)} \frac{\partial \xi(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial \hat{\psi}_j(t)} \frac{\partial \hat{\psi}_j(t)}{\partial u_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t)} \frac{\partial u_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t)}{\partial b_{1,j}^{\left(\Upsilon^{(j)}_r\right)}} \\ &= -\xi(t)a(t-j)\varphi'\left(u_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t)\right) \\ &= \varphi'\left(u_{1,j}^{\left(\Upsilon^{(j)}_r\right)}\right)c(t-j) \\ &= \delta_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t)\end{aligned} \tag{A.5}$$

As for neuron $n^{(j)}_{\Upsilon^{(j)}_r-1} \in \left\{1, 2, \cdots, Q^{(j)}_{\Upsilon^{(j)}_r-1}\right\}$, the gradient for parameter in the $(\Upsilon^{(j)}_r-1)$-th layer is update by the following Eq. (A. 6).

$$\begin{aligned}&\frac{\partial \mathrm{H}(t)}{\partial w_{n^{(j)}_{\Upsilon^{(j)}_r-1},n^{(j)}_{\Upsilon^{(j)}_r-2},j}^{\left(\Upsilon^{(j)}_r-1\right)}} = \frac{\partial \mathrm{H}(t)}{\partial \xi(t)} \frac{\partial \xi(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial \hat{\psi}_j(t)} \frac{\partial \hat{\psi}_j(t)}{\partial u_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t)} \frac{\partial u_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t)}{\partial h_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t)} \frac{\partial h_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t)}{\partial u_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t)} \frac{\partial u_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t)}{\partial w_{n^{(j)}_{\Upsilon^{(j)}_r-1},n^{(j)}_{\Upsilon^{(j)}_r-2},j}^{\left(\Upsilon^{(j)}_r-1\right)}} \\ &= -\xi(t)a(t-j)\varphi'\left(u_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t)\right) w_{1,n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r\right)} \varphi'\left(u_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t)\right) h_{n^{(j)}_{\Upsilon^{(j)}_r-2},j}^{\left(\Upsilon^{(j)}_r-2\right)}(t) \\ &= \varphi'\left(u_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t)\right) w_{1,n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r\right)} \varphi'\left(u_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t)\right) c(t-j) h_{n^{(j)}_{\Upsilon^{(j)}_r-2},j}^{\left(\Upsilon^{(j)}_r-2\right)}(t) \\ &= \varphi'\left(u_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t)\right) w_{1,n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r\right)} \delta_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t) h_{n^{(j)}_{\Upsilon^{(j)}_r-2},j}^{\left(\Upsilon^{(j)}_r-2\right)}(t)\end{aligned} \tag{A.6}$$

Let

$$\delta_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t) = \varphi'\left(u_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t)\right) w_{1,n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r\right)} \delta_{1,j}^{\left(\Upsilon^{(j)}_r\right)}(t) \tag{A.7}$$

then Eq. (A.6) is rewritten as follows

$$\frac{\partial \mathrm{H}(t)}{\partial w_{n^{(j)}_{\Upsilon^{(j)}_r-1},n^{(j)}_{\Upsilon^{(j)}_r-2},j}^{\left(\Upsilon^{(j)}_r-1\right)}} = \delta_{n^{(j)}_{\Upsilon^{(j)}_r-1},j}^{\left(\Upsilon^{(j)}_r-1\right)}(t) h_{n^{(j)}_{\Upsilon^{(j)}_r-2},j}^{\left(\Upsilon^{(j)}_r-2\right)}(t) \tag{A.8}$$

According to the same theory, the following equation can be calculated

$$
\begin{aligned}
\frac{\partial \mathrm{H}(t)}{\partial b^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}} &= \frac{\partial \mathrm{H}(t)}{\partial \xi(t)} \frac{\partial \xi(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial \hat{\psi}_j(t)} \frac{\partial \hat{\psi}_j(t)}{\partial u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}(t)} \frac{\partial u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}(t)}{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)}{\partial u^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)} \frac{\partial u^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)}{\partial b^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}} \\
&= -\xi(t)a(t-j)\varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}(t)\right) w^{\left(\Upsilon_r^{(j)}\right)}_{1,n^{(j)}_{\Upsilon_r^{(j)}-1},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)\right) \\
&= \varphi'\left(u^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)\right) w^{\left(\Upsilon_r^{(j)}\right)}_{1,n^{(j)}_{\Upsilon_r^{(j)}-1},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}(t)\right) c(t-j) \\
&= \delta^{\left(\Upsilon_r^{(j)}-1\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)
\end{aligned}
\tag{A.9}
$$

For neuron $n^{(j)}_{\Upsilon_r^{(j)}-2} \in \left\{1, 2, \cdots, Q^{(j)}_{\Upsilon_r^{(j)}-2}\right\}$, the gradient for parameter in the $(N_r^{(j)}-2)$-th layer is updated by the following Eq. (A. 10).

$$
\begin{aligned}
\frac{\partial \mathrm{H}(t)}{\partial w^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},n^{(j)}_{\Upsilon_r^{(j)}-3},j}} &= \frac{\partial \mathrm{H}(t)}{\partial \xi(t)} \frac{\partial \xi(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial \hat{\psi}_j(t)}
\left(\begin{array}{l}
\frac{\partial \hat{\psi}_j(t)}{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{1,j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{1,j}(t)}{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial w^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},n^{(j)}_{\Upsilon_r^{(j)}-3},j}} \\
+\frac{\partial \hat{\psi}_j(t)}{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{2,j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{2,j}(t)}{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial u^{\Upsilon_r^{j}-2}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial w^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},n^{(j)}_{\Upsilon_r^{(j)}-3},j}} \\
+\cdots \\
+\frac{\partial \hat{\psi}_j(t)}{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{Q^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{Q^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)}{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial w^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},n^{(j)}_{\Upsilon_r^{(j)}-3},j}}
\end{array}\right) \\
&= -\xi(t)a(t-j)
\left(\begin{array}{l}
\varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}\right) w^{\left(\Upsilon_r^{(j)}\right)}_{1,1,j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-1\right)}_{1,j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{1,n^{(j)}_{\Upsilon_r^{(j)}-2},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right) h^{\left(\Upsilon_r^{(j)}-3\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-3},j}(t) \\
+\varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}\right) w^{\left(\Upsilon_r^{(j)}\right)}_{1,2,j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-1\right)}_{2,j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{2,n^{(j)}_{\Upsilon_r^{(j)}-2},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right) h^{\left(\Upsilon_r^{(j)}-3\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-3},j}(t) \\
+\cdots \\
+\varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}\right) w^{\left(\Upsilon_r^{(j)}\right)}_{1,Q^{(j)}_{\Upsilon_r^{(j)}-1},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-1\right)}_{Q^{(j)}_{\Upsilon_r^{(j)}-1},j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{Q^{(j)}_{\Upsilon_r^{(j)}-1},n^{(j)}_{\Upsilon_r^{(j)}-2},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right) h^{\left(\Upsilon_r^{(j)}-3\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-3},j}(t)
\end{array}\right) \\
&= \sum_{v=1}^{Q^{(j)}_{\Upsilon_r^{(j)}-1}} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{v,n^{(j)}_{\Upsilon_r^{(j)}-2},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-1\right)}_{v,j}\right) w^{\left(\Upsilon_r^{(j)}\right)}_{1,v,j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}\right) c(t-j) h^{\left(\Upsilon_r^{(j)}-3\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-3},j}(t)
\end{aligned}
$$

$$= \sum_{v=1}^{Q^{(j)}_{\Upsilon_r^{(j)}-1}} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{v,n^{(j)}_{\Upsilon_r^{(j)}-2},j} \delta^{\left(\Upsilon_r^{(j)}-1\right)}_{v,j}(t) h^{\left(\Upsilon_r^{(j)}-3\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-3},j}(t) \tag{A.10}$$

Let

$$\delta^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t) = \sum_{v=1}^{Q^{(j)}_{\Upsilon_r^{(j)}-1}} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{v,n^{(j)}_{\Upsilon_r^{(j)}-2},j} \delta^{\left(\Upsilon_r^{(j)}-1\right)}_{v,j}(t) \tag{A.11}$$

then Eq. (A. 10) can be rewriten as follows

$$\frac{\partial \mathrm{H}(t)}{\partial w^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},n^{(j)}_{\Upsilon_r^{(j)}-3},j}} = \delta^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t) h^{\left(\Upsilon_r^{(j)}-3\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-3},j}(t) \tag{A.12}$$

According to the same theory, Eq. (A. 13) can be obtained as follows.

$$\begin{aligned}
\frac{\partial \mathrm{H}(t)}{\partial b^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}} &= \frac{\partial \mathrm{H}(t)}{\partial \xi(t)} \frac{\partial \xi(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial \hat{\psi}_j(t)} \left(\begin{array}{l}
\frac{\partial \hat{\psi}_j(t)}{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{1,j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{1,j}(t)}{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial b^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}} \\
+ \frac{\partial \hat{\psi}_j(t)}{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{2,j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{2,j}(t)}{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial u^{\Upsilon_r^{j}-2}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial b^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}} \\
+ \cdots \\
+ \frac{\partial \hat{\psi}_j(t)}{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{Q^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-1\right)}_{Q^{(j)}_{\Upsilon_r^{(j)}-1},j}(t)}{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial h^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)} \frac{\partial u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t)}{\partial b^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}}
\end{array} \right) \\
&= -\xi(t)a(t-j) \left(\begin{array}{l}
\varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}\right) w^{\left(\Upsilon_r^{(j)}\right)}_{1,1,j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-1\right)}_{1,j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{1,n^{(j)}_{\Upsilon_r^{(j)}-2},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right) \\
+\varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}\right) w^{\left(\Upsilon_r^{(j)}\right)}_{1,2,j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-1\right)}_{2,j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{2,n^{(j)}_{\Upsilon_r^{(j)}-2},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right) \\
+ \cdots \\
+\varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}\right) w^{\left(\Upsilon_r^{(j)}\right)}_{1,Q^{(j)}_{\Upsilon_r^{(j)}-1},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-1\right)}_{Q^{(j)}_{\Upsilon_r^{(j)}-1},j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{Q^{(j)}_{\Upsilon_r^{(j)}-1},n^{(j)}_{\Upsilon_r^{(j)}-2},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right)
\end{array} \right)
\end{aligned}$$

$$= \sum_{v=1}^{Q^{(j)}_{\Upsilon_r^{(j)}-1}} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{v,n^{(j)}_{\Upsilon_r^{(j)}-2},j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-1\right)}_{v,j}\right) w^{\left(\Upsilon_r^{(j)}\right)}_{1,v,j} \varphi'\left(u^{\left(\Upsilon_r^{(j)}\right)}_{1,j}\right) c(t-j)$$

$$= \sum_{v=1}^{Q^{(j)}_{\Upsilon_r^{(j)}-1}} \varphi'\left(u^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}\right) w^{\left(\Upsilon_r^{(j)}-1\right)}_{v,n^{(j)}_{\Upsilon_r^{(j)}-2},j} \delta^{\left(\Upsilon_r^{(j)}-1\right)}_{v,j}(t)$$

$$= \delta^{\left(\Upsilon_r^{(j)}-2\right)}_{n^{(j)}_{\Upsilon_r^{(j)}-2},j}(t) \tag{A.13}$$

Hence, the above discussed derivation process is used to calculate the local gradient of each neuron of layer ℓ_j in the j - th DBN module, and the following equation is used to calculate derivation.

$$\delta^{(\ell_j)}_{n^{(j)}_{\ell_j},j}(t) = \sum_{v=1}^{Q^{(j)}_{\ell_j+1}} \varphi'\left(u^{(\ell_j)}_{n^{(j)}_{\ell_j},j}\right) w^{(\ell_j+1)}_{v,n^{(j)}_{\ell_j},j} \delta^{(\ell_j+1)}_{v,j}(t),\ \ell_j \in \left\{1, 2, \cdots, \Upsilon_r^{(j)} - 2\right\} \tag{A.14}$$

then, the Eq. (A.15) and Eq. (A.16) are used to update the gradients to the connection weight and bias used for parameters.

$$\frac{\partial \mathrm{H}(t)}{\partial w^{(\ell_j)}_{n^{(j)}_{\ell_j},n^{(j)}_{\ell_j-1},j}} = \delta^{(\ell_j)}_{n^{(j)}_{\ell_j},j}(t) h^{(\ell_j-1)}_{n^{(j)}_{\ell_j-1},j}(t) \tag{A.15}$$

$$\frac{\partial \mathrm{H}(t)}{\partial b^{(\ell_j)}_{n^{(j)}_{\ell_j},j}} = \delta^{(\ell_j)}_{n^{(j)}_{\ell_j},j}(t) \tag{A.16}$$

(A.3-A.5), (A.7-A.9) and (A.11-A.16) are used to calculate all the gradients, then the following Eq. (A.17) and Eq. (A.18) are used to update the parameters.

$$\Delta w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j} = -\eta \frac{\partial \mathrm{H}(t)}{\partial w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j}} = -\eta \delta^{(L)}_{n^{(j)}_L,j} h^{(L-1)}_{n^{(j)}_{L-1},j}(t) \tag{A.17}$$

$$\Delta b^{(L)}_{n^{(j)}_L,j} = -\eta \frac{\partial \mathrm{H}(t)}{\partial b^{(L)}_{n^{(j)}_L,j}} = -\eta \delta^{(L)}_{n^{(j)}_L,j} \tag{A.18}$$

where $L \in \left\{1, 2, \cdots, \Upsilon_r^{(j)} - 1, \Upsilon_r^{(j)}\right\}$ and $\eta > 0$ are the value of pre-determined learning rate, and Eq. (A.19) is used to update the parameters

$$\begin{cases} w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j} \Leftarrow w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j} + \Delta w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j} \\ b^{(L)}_{n^{(j)}_L,j} \Leftarrow b^{(L)}_{n^{(j)}_L,j} + \Delta b^{(L)}_{n^{(j)}_L,j} \end{cases} \tag{A.19}$$

where, the initial value of $w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j}$ and $b^{(L)}_{n^{(j)}_L,j}$ can be obtained in the pre-training stage of the DBN-AR model. In order to avoid the parameter oscillation in the fine-tuning process and slow down the convergence rate, the momentum term is used to add in Eq. (21) as the final parameters updating rule.

$$\begin{cases} w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j}(k) \Leftarrow w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j}(k-1) + \Delta w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j}(k) + \alpha\left(w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j}(k-1) - w^{(L)}_{n^{(j)}_L,n^{(j)}_{L-1},j}(k-2)\right) \\ b^{(L)}_{n^{(j)}_L,j}(k) \Leftarrow b^{(L)}_{n^{(j)}_L,j}(k-1) + \Delta b^{(L)}_{n^{(j)}_L,j}(k) + \alpha\left(b^{(L)}_{n^{(j)}_L,j}(k-1) - b^{(L)}_{n^{(j)}_L,j}(k-2)\right) \end{cases} \tag{A.20}$$

where k denotes the number of parameters updating iteration, $\alpha \in [0, 1)$ denotes the predetermined momentum factor.

References

1. Min G, Hui P (2012) Stability analysis of RBF network-based state-dependent autoregressive model for nonlinear time series. Appl Soft Comput 12(1):174–181
2. Wang S, Wang H, Zhao Y, Cao G, Li Y (2021) Empirical likelihood ratio tests for varying coefficient geo models. Stat Sin. https://doi.org/10.5705/ss.202021.0233
3. Keller JP, Zhou W, Zhou T, Kaplan A, Anderson GB (2022) Tracking the transmission dynamics of covid-19 with a time-varying coefficient state-space model. Stat Med 41(15):2745–2767
4. Peng QY, Zhou JJ, Tang NS (2016) Varying coefficient partially functional linear regression models. Stat Pap 57(3):827–841
5. Kim S, Cho HR, Kim MO (2021) Predictive generalized varying-coefficient longitudinal model. Stat Med 40(28):6243–6259
6. Vu DH, Muttaqi KM, Agalgaonkar AP, Bouzerdoum A (2017) Short-term electricity demand forecasting using autoregressive based time varying model incorporating representative data adjustment. Appl Energy 205:790–801
7. Cai Z, Yao FQ (2000) Functional-coefficient regression models for nonlinear time series. J Am Stat Assoc 95(451):941–956
8. Tian R, Xu D, Du J, Zhang J (2022) Bayesian estimation for partially linear varying coefficient spatial autoregressive models. Stat Interface 15(1):105–113
9. Patrick J, Harvill J, Sims J (2015) Spline-backfitted kernel forecasting for functional-coefficient autoregressive models. Statistics 35(6):2474–2503
10. Bussas M, Sawade C, Kuhn N, Scheffer T, Landwehr N (2017) Varying-coefficient models for geospatial transfer learning. Mach Learn 106(9–10):1–22
11. Tan H, Chen Y, Wilson JP, Zhang J, Cao J, Chu T (2020) An eigenvector spatial filtering based spatially varying coefficient model for PM (2.5) concentration estimation: a case study in Yangtze River Delta region of China. Atmos Environ 223:117205.1-117205.9
12. Zhang R (2009) Proportional functional coefficient time series models. J Stat Plan Inference 139(3):749–763
13. Priestley MB (1980) State-dependent models: a general approach to non-linear time series analysis. J Time Ser Anal 1(1):57–71
14. Tong H (1983) Threshold models in nonlinear time series analysis. Springer, New York
15. Granger C (1993) Modelling nonlinear economic relationships. Oxford University Press, Oxford
16. Xu W, Peng H, Zeng X, Zhou F, Peng X (2019) Deep belief network-based AR model for nonlinear time series forecasting. Appl Soft Comput 77(10):605–621
17. Xu W, Peng H, Tian X, Peng X (2020) DBN based SD-ARX model for nonlinear time series prediction and analysis. Appl Intell 50(2):4586–4601
18. An HZ, Huang FC (1996) The geometrical ergodicity of nonlinear autoregressive models. Stat Sin 6(4):943–956
19. Chan KS, Tong H (1985) On the use of the deterministic Lyapunov function for the ergodicity of stochastic difference equations. Adv Appl Probab 17:666–678
20. Tweedie RL (1975) Criteria for classifying general Markov chains. Adv Appl Probab 8:737–771
21. Tjtheim D (1990) Non-linear time series and Markov chains. Adv Appl Probab 22:587–611
22. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
23. Zhang GP (2003) Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50:159–175
24. Katijani Y, Hipel WK, Mcleod AL (2005) Forecasting nonlinear time series with feedforward neural networks: a case study of Canadian lynx data. J Forecast 24:105–117
25. Aladag CH, Egrioglu E, Kadilar C (2009) Forecasting nonlinear time series with a hybrid methodology. Appl Math Lett 22:1467–1470
26. Han HG, Guo YN, Qiao JF (2018) Nonlinear system modeling using a self-organizing recurrent radial basis function neural network. Appl Soft Comput 71:1105–1116
27. Juang CF (2002) A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithm. IEEE Trans Fuzzy Syst 10(2):155–170

28. Lee CH, Teng CC (2002) Identification and control of dynamic systems using recurrent fuzzy neural networks. IEEE Trans Fuzzy Syst 8(4):349–366
29. Lin CJ, Chin CC (2004) Prediction and identification using waveletbased recurrent fuzzy neural networks. IEEE Trans Syst Man Cybern Soc 34(5):2144–2154
30. Juang CF, Lin YY, Tu CC (2010) A recurrent self-evolving fuzzy neural network with local feedbacks and its application to dynamic system processing. Fuzzy Sets Syst 161(19):2552–2568
31. Lin YY, Chang JY, Lin CT (2013) Identification and prediction of dynamic systems using an interactively recurrent self-evolving fuzzy neural network. IEEE Trans Neural Netw Learn Syst 24(2):310–321