Fuzzy Logic Project 2-Forecasting of Time Series

YangYin
202122011320

April 17, 2022

Abstract

In this project, we design a singleton type-1 Fuzzy Logic System and a Non-singleton type-1 Fuzzy Logic System for data forecasting. In section 2, we create a Mackey-Glass Chaotic Time Series as our data for trainning and forecasting.In section 3,We design an One-Pass Singleton Fuzzification systerm to forecast the time series data and calculating it's RMSE(root-mean-square error).In section 4, we Apply back-propagation method to design a singleton FLS and use the time series data to train and test the FLS. Also, we calculate it's RMSE for six epochs. Finally, in section 5, we Apply the smae back-propagation method to design a Non-singleton FLS and use the same time series data to train and test it. And, we calculate it's RMSEs for six epochs.
keywords:FLS, One-Pass, data forecast.

1 requirement

1.1 Basic requirement

1. Generate 2 Mackey-Glass chaotic time series (2000 samples each) with tao=15 and tao=35, respectively. Plot their time series for samples 1000-1500 and their phase, like figure 4.2 in the text book.
2. Apply one-pass method to design a singleton FLS (Gaussian MFs, 4 antecedents, height defuzzifier, product t-norm) based on the Mackey-Glass chaotic time series in 1. with tao=35. Find RMSE for the one-pass method.
3. Apply back-propagation method to design a singleton FLS (Gaussian MFs, 4 antecedents,16 rules, height defuzzifier, product t-norm) based on the Mackey-Glass chaotic time series in 1. with tao=35. Samples 1001-1504 are for training, and samples 1505-2000 are for testing. Provide initial values and final values for MF parameters, like Tables 4.7-4.9. Find RMSE for the tested data for 6 epochs.
4. Plot the RMSE of 2. and the RMSE of 3. in the same figure

1.2 Advanced requirement

5. Regenerate a Mackey-Glass chaotic time series (2000 samples) with tao=35 by adding uniformly distributed noise with SNR=0dB. Finish 3. again and plot its RMSE.

6. Do 5.again with non-singleton fuzzifier. Plot the RMSE of 3. the RMSE of 5. and the RMSE of 6. in the same figure.

2 Mackey-Glass chaotic time series

Mackey and Glass [1] is an important paper in which the authors "associate the onset of disease with bifurcations in the dynamics of first-order differential- delay equations which model physiological systems". Equation of that paper, which has become known as the Mackey–Glass equation, is a nonlinear delay differential equation, one form of which is:

$$\frac{\partial s(t)}{\partial t} = \frac{0.2s(t-\tau)}{1+s^{10}(t-\tau)} - 0.1s(t) \tag{1}$$

We Generate 2 Mackey-Glass chaotic time series (2000 samples each) with tao=15 and tao=35, respectively. And we plot their time series for samples 1000-1500 and their phase in figure 1 as follows.

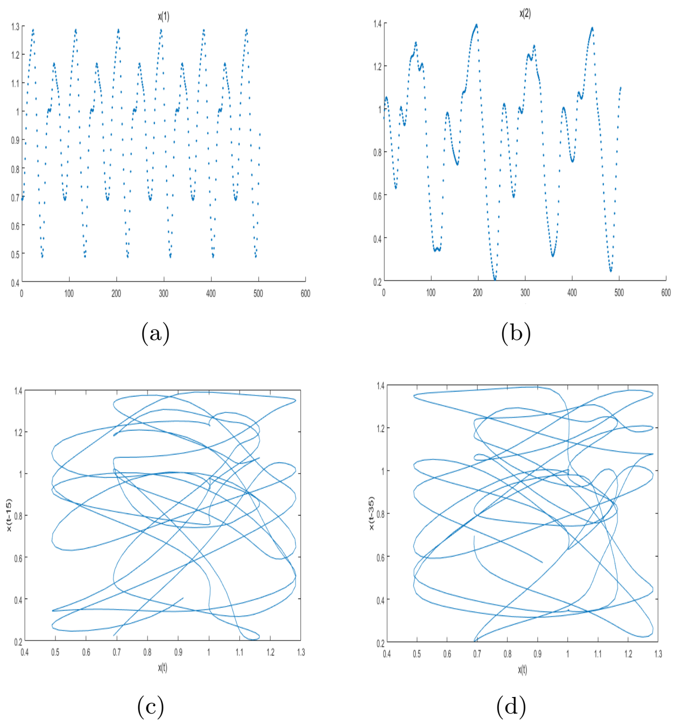


Figure 1: **a** and **b** are representative samples of the Mackey–Glass time series after letting transients relax. c and d are the corresponding phase plots of the time segments depicted in **(a)** and **(b)**. Note that "d" in $s(t-\tau)$ on the vertical axis denotes the delay(τ) used in the Mackey–Glass equation; it is 15 for **(a)** and **(c)** and 35 for **(b)** and **(d)**

3 One-Pass Design: Singleton Fuzzification

3.1 Data Assignment Method

In this method, the data pairs establish the centers of the fuzzy sets that appear in the antecedents and consequents of the rules. Here, for example, are N rules that can be extracted from the N data pairs,$(x^{(1)}, y^{(1)}), (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(N)}, y^{(N)})$[2]

1.R1: IF x_1 is F_1^1 and x_2 is F_2^1 and ... and x_p is F_p^1 ; THEN y is G^1—In this rule, which is obtained from ,$(x^{(1)}, y^{(1)})$, F_1^1 is a fuzzy set whose MF is centered at $x_1^{(1)}$, F_2^1 is a fuzzy set whose MF is centered at $x_2^{(1)}$,...,F_p^1 is a fuzzy set whose MF is centered at $x_p^{(1)}$, and G^1 is a fuzzy set whose MF is centered at $y^{(1)}$.

2.R2: IF x_1 is F_1^2 and x_2 is F_2^2 and ... and x_p is F_p^2 ; THEN y is G^2—In this rule, which is obtained from ,$(x^{(2)}, y^{(2)})$, F_1^2 is a fuzzy set whose MF is centered at $x_1^{(2)}$, F_2^2 is a fuzzy set whose MF is centered at $x_2^{(2)}$,...,F_p^2 is a fuzzy set whose MF is centered at $x_p^{(2)}$, and G^2 is a fuzzy set whose MF is centered at $y^{(2)}$.

...

N.RN: IF x_1 is F_1^N and x_2 is F_2^N and ... and x_p is F_p^N ; THEN y is G^N—In this rule, which is obtained from ,$(x^{(N)}, y^{(N)})$, F_1^N is a fuzzy set whose MF is centered at $x_1^{(N)}$, F_2^N is a fuzzy set whose MF is centered at $x_2^{(N)}$,...,F_p^N is a fuzzy set whose MF is centered at $x_p^{(N)}$, and G^N is a fuzzy set whose MF is centered at $y^{(N)}$.

3.2 One-Pass Singleton FLS

In this method, we only need to traverse the data series once. To begin, the first 504 data,$s(1001), s(1002), ..., s(1504)$,were used to design a one-pass fuzzy system forecaster and remaining 496 data,$s(1505), s(1506), ..., s(2000)$,were used for testing the design. In this case, we use Gaussian MFS product t-norm and height defuzzifier, and only the four antecedents $s(k-3), s(k-2), s(k-1)$and$s(k)$ were used to predict $s(k+1)$.The performance of this design was evaluated using the following root mean-squared error (RMSE):

$$RMSE_S = \sqrt{\frac{1}{496} \sum_{k=1504}^{1999} [s(k+1) - y_s(\mathbf{s}^{(k)})]^2} \tag{2}$$

where $\mathbf{s}^{(k)} = [s(k-3), s(k-2), s(k-1), s(k)]^T$ is found by substituting the top line of (3.3) into (3.2),and the subscript "s" on RMSE reminds us that this is a singleton design

$$y(\mathbf{x}) = \sum_{l=1}^{M} \lambda^l \phi_l(\mathbf{x}) \tag{3}$$

$$\varphi_l(\mathbf{x}) = \begin{cases} \frac{\prod_{i=1}^{p} \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{F_i^l}(x_i)} & sigleton\ fuzzification \\ \frac{\prod_{k=1}^{p} \mu_{Q_l^k}(x_{i,max}^l|x_i)}{\sum_{l=1}^{M} \prod_{k=1}^{p} \mu_{Q_l^k}(x_{i,max}^l|x_i)} & non-sigleton\ fuzzification \end{cases} \tag{4}$$

Besides,we use the method which is mentioned in section 3.1 to construct 500 Zadeh rules from $s(1001), s(1002), ..., s(1504)$. The antecedent and consequent MFs were centered at the noise-free measurements in each one of the 500 rules, and the standard deviation of the Gaussians was set equal to 0.1. This established the Mamdani singleton fuzzy system forecaster $y_s(\mathbf{s}^{(k)})$ .Then, $s(1505), s(1506), ..., s(2000)$, were used to compute the one-pass RMSE in (3.1) as $RMSE_S = 0.0368$

4 Back-Propagation:Singleton Fuzzification

4.1 Back-Propagation Methods

Two of the most popular and widely used derivative-based optimization algorithms are steepest descent and Marquardt—Levenberg. When they are used, none of the antecedent or consequent parameters are fixed ahead of time. Both algorithms need the first derivative of a mathematical objective function with respect to each MF parameter. In this section, the focus is on a steepest descent algorithm for a sin- gleton Mamdani fuzzy system and product t-norm, for which (3.3), that uses the first line of (3.2), can be expressed as:

$$y(\mathbf{x}^{(t)}) = \sum_{l=1}^{M} \lambda^l \phi_l(\mathbf{x}^{(t)}) = \frac{\sum_{l=1}^{M} \lambda^l \prod_{i=1}^{p} exp\left\{-\frac{1}{2}[(x_i^{(t)} - m_{F_i^l})/\sigma_{F_i^l}]^2\right\}}{\sum_{l=1}^{M} \prod_{i=1}^{p} exp\left\{-\frac{1}{2}[(x_i^{(t)} - m_{F_i^l})/\sigma_{F_i^l}]^2\right\}} \tag{5}$$

Given the data pairs $\{\mathbf{x}^{(t)} : \mathbf{y}^{(t)}\}_{t=1}^{N}$, the goal here is to design this fuzzy system such that the following error function is minimized $(t = 1, ..., N)$:

$$\mathbf{J}(\theta) = e^{(t)} = \frac{1}{2}[y(\mathbf{x}^{(t)} - y^{(t)})]^2 \tag{6}$$

For clarity, instead of putting all of the unknown parameters into one vector of all such parameters, the parameters are considered below one at a time,i.e$\theta = \lambda^l or m_{F_i^l} or \sigma_{F_i^l}$.

A steepest descent optimization algorithm that only updates the parameter θ one time for each $e^{(t)}$ has the following generic structure:

$$\theta(t+1) = \theta(t) - \beta_\theta \frac{\partial \mathbf{J}(\theta)}{\partial \theta} \tag{7}$$

where, regardless of what θ is:

$$\frac{\partial \mathbf{J}(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}\left\{\frac{1}{2}[y(\mathbf{x}^{(t)} - y^{(t)})]^2\right\} = [y(\mathbf{x}^{(t)}) - y^{(t)}]\frac{\partial}{\partial \theta} y(\mathbf{x}^{(t)}) \tag{8}$$

(a)$\theta = \lambda^l$: In this case, it is very easy to compute$\partial(\mathbf{x}^{(t)})/\partial\lambda^l$ from the first term on the right-hand side of (4.1), as $\partial(\mathbf{x}^{(t)})/\partial\lambda^l = \phi_l(\mathbf{x}^{(t)})$; hence, a steepest descent algorithm for λ^l is $(t = 1, 2, ...)$:

$$\lambda^l(t+1) = \lambda^l(t) - \beta_\lambda[y(\mathbf{x}^{(t)} - y^{(t)})]\phi_l(\mathbf{x}^{(t)}) \tag{9}$$

(b)$\theta = m_{F_i^l}$: Computing $\partial(\mathbf{x}^{(t)})/\partial_{F_i^l}$ is much more challenging than computing $\partial(\mathbf{x}^{(t)})/\partial\lambda^l$, because $m_{F_i^l}$ appears very non-linearly in (5) and is in both its numerator and denominator. To begin, let y in (5) be expressed as $y \equiv \frac{h}{g}$, where $h = \sum_{l=1}^{M} \lambda^l w^l$,$g = \sum_{l=1}^{M} w^l$ and $w^l = \prod_{i=1}^{p} exp\left\{-\frac{1}{2}[(x_i^{(t)} - m_{F_i^l})/\sigma_{F_i^l}]^2\right\}$ so that $\phi_l = w^l/g$. Using the chain rule, one finds:

$$\frac{\partial y}{\partial m_{F_i^l}} = \frac{\partial y}{\partial w^l}\frac{\partial w^l}{\partial m_{F_i^l}} \tag{10}$$

$$\frac{\partial y}{\partial w^l} = \frac{g\frac{\partial h}{\partial w^l} - h\frac{\partial g}{\partial w^l}}{g^2} = \frac{g\lambda^l - \frac{h}{g}}{g} = \frac{g\lambda^l - y}{g} \tag{11}$$

$$\begin{aligned}
\frac{\partial w^l}{\partial m_{F_i^l}} &= \frac{\partial}{\partial m_{F_i^l}} \prod_{j=1}^{p} exp\left\{-\frac{1}{2}[(x_j^{(t)} - m_{F_j^l})/\sigma_{F_j^l}]^2\right\} \\
&= \frac{\partial}{\partial m_{F_i^l}} exp\left\{-\frac{1}{2}[(x_i^{(t)} - m_{F_i^l})/\sigma_{F_i^l}]^2\right\} \times \prod_{j=1\ j\neq i}^{p} exp\left\{-\frac{1}{2}[(x_j^{(t)} - m_{F_j^l})/\sigma_{F_j^l}]^2\right\} \\
&= \prod_{j=1}^{p} exp\left\{-\frac{1}{2}[(x_j^{(t)} - m_{F_j^l})/\sigma_{F_j^l}]^2\right\} \times \frac{x_i^{(i)} - m_{F_i^l}}{\sigma_{F_i^l}^2} \\
&= \frac{x_i^{(i)} - m_{F_i^l}}{\sigma_{F_i^l}^2} \times w^l
\end{aligned} \tag{12}$$

Substituting (12) and (11) into (10), one finds that:

$$\frac{\partial y}{\partial m_{F_i^l}} = \frac{g\lambda^l - y}{g} \times \frac{(x_i^{(i)} - m_{F_i^l})}{\sigma_{F_i^l}^2} \times w^l = (\lambda^l - y_s) \times \frac{(x_i^{(i)} - m_{F_i^l})}{\sigma_{F_i^l}^2} \times \phi_l \tag{13}$$

One therefore obtains the following steepest descent algorithm for updating $m_{F_i^l}$:

$$\begin{aligned}
m_{F_i^l}(t+1) &= m_{F_i^l}(t) - \beta_m[y(\mathbf{x}^t) - y^{(t)}]\frac{\partial y}{\partial m_{F_i^l}}|_t \\
&= m_{F_i^l}(t) - \beta_m[y(\mathbf{x}^t) - y^{(t)}] \times [\lambda^l(t) - y(\mathbf{x}^{(t)})] \times \frac{(x_i^{(i)} - m_{F_i^l})}{\sigma_{F_i^l}^2} \times \phi_l(\mathbf{x}^t)
\end{aligned} \tag{14}$$

(c)$\theta = \sigma_{F_i^l}$:Instead of updating $\sigma^2_{F_i^l}$, $\sigma_{F_i^l}$ is updated, because $\sigma^2_{F_i^l}$ must be positive whereas $\sigma_{F_i^l}$ can be positive or negative:

$$\frac{\partial y}{\partial \sigma_{F_i^l}} = \frac{\partial y}{\partial w^l}\frac{\partial w^l}{\partial \sigma_{F_i^l}} \tag{15}$$

$$\begin{aligned}
\frac{\partial w^l}{\partial \sigma_{F_i^l}} &= \frac{\partial}{\partial \sigma_{F_i^l}} \prod_{j=1}^{p} exp\left\{-\frac{1}{2}[(x_j^{(t)} - m_{F_j^l})/\sigma_{F_j^l}]^2\right\} \\
&= \frac{\partial}{\partial \sigma_{F_i^l}} exp\left\{-\frac{1}{2}[(x_i^{(t)} - \sigma_{F_i^l})/\sigma_{F_i^l}]^2\right\} \times \prod_{j=1\ j\neq i}^{p} exp\left\{-\frac{1}{2}[(x_j^{(t)} - m_{F_j^l})/\sigma_{F_j^l}]^2\right\} \\
&= \prod_{j=1}^{p} exp\left\{-\frac{1}{2}[(x_j^{(t)} - m_{F_j^l})/\sigma_{F_j^l}]^2\right\} \times \frac{(x_i^{(i)} - m_{F_i^l})^2}{\sigma^3_{F_i^l}} \\
&= \frac{(x_i^{(i)} - m_{F_i^l})^2}{\sigma^3_{F_i^l}} \times w^l
\end{aligned} \tag{16}$$

$$\frac{\partial y}{\partial \sigma_{F_i^l}} = \frac{g\lambda^l - y}{g} \times \frac{(x_i^{(i)} - m_{F_i^l})^2}{\sigma^3_{F_i^l}} \times w^l = (\lambda^l - y_s) \times \frac{(x_i^{(i)} - m_{F_i^l})^2}{\sigma^3_{F_i^l}} \times \phi_l \tag{17}$$

One therefore obtains the following steepest descent algorithm for updating $\sigma_{F_i^l}$:

$$\begin{aligned}
\sigma_{F_i^l}(t+1) &= \sigma_{F_i^l}(t) - \beta_\sigma[y(\mathbf{x}^t) - y^{(t)}]\frac{\partial y}{\partial \sigma_{F_i^l}}|_t \\
&= \sigma_{F_i^l}(t) - \beta_\sigma[y(\mathbf{x}^t) - y^{(t)}] \times [\lambda^l(t) - y(\mathbf{x}^{(t)})] \times \frac{(x_i^{(i)} - m_{F_i^l})^2}{\sigma^3_{F_i^l}} \times \phi_l(\mathbf{x}^t)
\end{aligned} \tag{18}$$

4.2 Back-Propagation Design: Singleton Fuzzification

As it demonstrated in [3], only two fuzzy sets were used for each of the four antecedents. hence, the number of rules is very small and equals $2^4 = 16$,hence, Each rule is characterized by eight antecedent MF parameters and one consequent parameter,$\bar{y}$. The initial location of each Gaussian antecedent MF was based on the mean, m_s and the standard deviation σs of the data in the 504 training samples. More specifically, the means of each and every antecedent's two Gaussian MFs were initially chosen as $m_s - 2\sigma_s = 0.3363$ or $m_s + 2\sigma_s = 1.4418$,respectively, and their standard deviations were initially chosen as $2\sigma_s = 0.5528$. The center of each consequent'sMF,$\bar{y}$,was initially chosen to be a random number from the interval $[0, 1]$. After training using a steepest descent algorithm, as described in Sect.4.1, in which the learning parameter $\beta_\theta = 0.2$ (this same value was used for all of theparameters), the fuzzy system forecaster was fixed. Its performance was then evaluated using the RMSE in (19)

Table 4.1: Initial values for the centers if the Gaussian antecendent MFs and the center if the consequent set

$RuleNo:$	m_1	m_2	m_3	m_4	σ_1	σ_2	σ_3	σ_4	$\hat{y}$
1	0.3363	0.3363	0.3363	0.3363	0.5528	0.5528	0.5528	0.5528	0.8451
2	0.3363	0.3363	0.3363	1.4418	0.5528	0.5528	0.5528	0.5528	0.3651
3	0.3363	0.3363	1.4418	0.3363	0.5528	0.5528	0.5528	0.5528	0.7951
4	0.3363	0.3363	1.4418	1.4418	0.5528	0.5528	0.5528	0.5528	0.2415
5	0.3363	1.4418	0.3363	0.3363	0.5528	0.5528	0.5528	0.5528	0.0215
6	0.3363	1.4418	0.3363	1.4418	0.5528	0.5528	0.5528	0.5528	0.6324
7	0.3363	1.4418	1.4418	0.3363	0.5528	0.5528	0.5528	0.5528	0.3548
8	0.3363	1.4418	1.4418	1.4418	0.5528	0.5528	0.5528	0.5528	0.7456
9	1.4418	0.3363	0.3363	0.3363	0.5528	0.5528	0.5528	0.5528	0.6541
10	1.4418	0.3363	0.3363	1.4418	0.5528	0.5528	0.5528	0.5528	0.0652
11	1.4418	0.3363	1.4418	0.3363	0.5528	0.5528	0.5528	0.5528	0.3887
12	1.4418	0.3363	1.4418	1.4418	0.5528	0.5528	0.5528	0.5528	0.9987
13	1.4418	1.4418	0.3363	0.3363	0.5528	0.5528	0.5528	0.5528	0.7451
14	1.4418	1.4418	0.3363	1.4418	0.5528	0.5528	0.5528	0.5528	0.2654
15	1.4418	1.4418	1.4418	0.3363	0.5528	0.5528	0.5528	0.5528	0.1452
16	1.4418	1.4418	1.4418	1.4418	0.5528	0.5528	0.5528	0.5528	0.4784

Table 4.2: Final values for the centers if the Gaussian antecendent MFs and the centorid of the consequent set after six epochs of tuning

$RuleNo:$	m_1	m_2	m_3	m_4	σ_1	σ_2	σ_3	σ_4	$\hat{y}$
1	0.3726	0.3136	0.2144	-0.0061	0.5062	0.5561	0.4935	0.3573	0.2291
2	0.2551	0.2310	0.1972	1.5675	0.4698	0.4407	0.3991	0.3697	0.1856
3	0.3102	0.3001	1.4924	0.2655	0.5252	0.5092	0.4675	0.4543	0.3719
4	0.2574	0.3211	1.2345	1.2612	0.5517	0.5572	0.8835	0.6053	1.2965
5	0.3083	1.5353	0.3381	0.3535	0.5257	0.3685	0.5380	0.5421	0.8391
6	0.2351	1.5178	0.2105	1.4964	0.4293	0.4585	0.3875	0.5132	0.0636
7	0.3139	1.4678	1.4627	0.2950	0.5453	0.5275	0.5201	0.4678	0.7157
8	0.3239	1.3598	1.3794	1.4117	0.5413	0.6548	0.6105	0.5551	1.2980
9	1.3329	0.3893	0.3489	0.2709	0.5266	0.5565	0.5286	0.4612	0.0149
10	1.4941	0.2550	0.2371	1.4626	0.4707	0.4262	0.3874	0.5810	0.1073
11	1.4524	0.3286	1.4278	0.2816	0.5302	0.5529	0.5906	0.4337	0.2544
12	1.5010	0.1897	1.4941	1.4897	0.4507	0.2600	0.4790	0.5012	0.1769
13	1.4651	1.4463	0.3751	0.3461	0.5292	0.5448	0.6109	0.5168	0.4097
14	1.5119	1.4930	0.2309	1.4493	0.4881	0.5084	0.4096	0.5640	0.5200
15	1.5146	1.4700	1.4481	0.4205	0.4700	0.5131	0.5697	0.5361	0.3431
16	1.1408	1.2045	1.3101	1.5095	0.7032	0.6800	0.6180	0.4853	0.3431

The initial values of the centers of the Gaussian antecedent MFs as well as of all $\bar{y}$ are given in Table 4.1. The final values of all these parameters after six epochs of tuning are given in Table 4.2.

The steepest descent (SD) $RMSE_S$, $RMSE_S(SD)$, for each of the six epochs oftun-ing are:

$$RMSE_s(SD) = \{0.0516, 0.0471, 0.0441, 0.0332, 0.0272, 0.0258\}$$

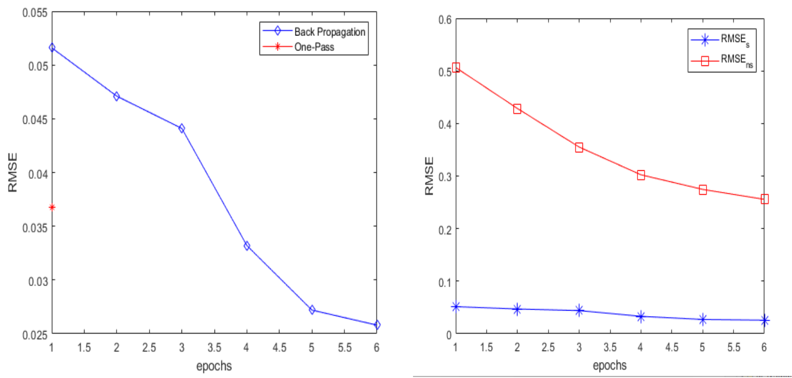
Next,we add uniformly disteributed noise with SNR=0dB to the Mackey-Glass time series,and then we retrain our module and test it again.The $RMSE_{ns}$ we finally obtain by using (19) are:

$$RMSE_{ns}(SD) = \{0.1812, 0.1785, 0.1751, 0.1724, 0.1646, 0.1621\}$$

$$RMSE_{ns} = \sqrt{\frac{1}{496}\sum_{k=1504}^{1999}[s(k+1) - y_{ns}(\mathbf{s}^{(k)})]^2} \tag{19}$$

in which $y_{ns}(\mathbf{s}^{(k)})$ means the consequense we obtain from the model training with noisy data.

The polts of the RMSEs of One-Pass method and Back-Propagation and the RMSEs of data with and without noise are in Figure 2.



(a) (b)

Figure 2: **(a)**The RMSEs of One-Pass method and Back-Propagation **(b)**The RMSEs of data with and without noise

5 Back-Propagation:Non-singleton Fuzzification

5.1 Back-Propagation method for Non-singleton Fuzzification

When it comes to Non-sigleton Fuzification,the $y(\mathbf{x}^{(t)})$ we can obtain from the FLS that uses product t-norm, Gaussian MFs and height defuzzifier is as follows [4]:

$$\begin{aligned} y(\mathbf{x^{(t)}}) =& \sum_{l=1}^{M} \lambda^l \phi_l(\mathbf{x^{(t)}}) = \frac{\sum_{\mathbf{l=1}}^{\mathbf{M}} \lambda^{\mathbf{l}} \prod_{\mathbf{i=1}}^{\mathbf{p}} \mu_{\mathbf{Q_i^l}}(\mathbf{x_{i,max}^l}|\mathbf{x_i'})}{\sum_{\mathbf{l=1}}^{\mathbf{M}} \prod_{\mathbf{i=1}}^{\mathbf{p}} \mu_{\mathbf{Q_i^l}}(\mathbf{x_{i,max}^l}|\mathbf{x_i'})} \\ =& \frac{\sum_{l=1}^{M} \lambda^l \prod_{i=1}^{p} \left\{ \exp[-(x_i' - m_{F_i^l})^2/2(\sigma_{X_i}^2 + \sigma_{F_i^l}^2)] \right\}}{\sum_{l=1}^{M} \prod_{i=1}^{p} \left\{ \exp[-(x_i' - m_{F_i^l})^2/2(\sigma_{X_i}^2 + \sigma_{F_i^l}^2)] \right\}} \end{aligned} \tag{20}$$

The same as we obtian the recursive formula of $\theta = \lambda^l, m_{F_i^l}$ *and* $\sigma_{F_i^l}$ of singleton fuzzification in Sect.4.1,We calculate the recursive formula of $\theta = \sigma_{F_i^l}, \sigma_x$ of non-singleton fuzzification as follows:

$$\begin{aligned} \frac{\partial y}{\partial \sigma_{F_i^l}} =& \frac{\partial y}{\partial \omega^l} \frac{\partial \omega^l}{\partial \sigma_{F_i^l}} \\ =& \frac{g\frac{\partial y}{\partial \omega^l} - h\frac{\partial g}{\partial \omega^l}}{g^2} \frac{\partial}{\partial \sigma_{F_i^l}} \prod_{i=1}^{p} \exp\left\{ -\frac{(x_i^{(t)} - m_{F_i^l})^2}{2(\sigma_{X_i}^2 + \sigma_{F_i^l}^2)} \right\} \\ =& \frac{\lambda^l - y}{g} \prod_{i=1}^{p} \exp\left\{ -\frac{(x_i^{(t)} - m_{F_i^l})^2}{2(\sigma_{X_i}^2 + \sigma_{F_i^l}^2)} \right\} \frac{\partial}{\partial \sigma_{F_i^l}} \left\{ -\frac{(x_i^{(t)} - m_{F_i^l})^2}{2(\sigma_{X_i}^2 + \sigma_{F_i^l}^2)} \right\} \\ =& \frac{\lambda^l - y}{g} \times \omega^l \times \frac{\sigma_{F_i^l}(x_i^{(t)} - m_{F_i^l})^2}{(\sigma_{X_i}^2 + \sigma_{F_i^l}^2)^2} \end{aligned} \tag{21}$$

$$\begin{aligned} \sigma_{F_i^l}(t+1) =& \sigma_{F_i^l}(t) - \beta_\sigma[y(\mathbf{x}^t) - y^{(t)}] \frac{\partial y}{\partial \sigma_{F_i^l}}|_t \\ =& \sigma_{F_i^l}(t) - \beta_\sigma[y(\mathbf{x}^t) - y^{(t)}] \times [\lambda^l(t) - y(\mathbf{x}^{(t)})] \times \frac{\sigma_{F_i^l}(x_i^{(i)} - m_{F_i^l})^2}{(\sigma_{F_i^l}^2 + \sigma_{X_i})^2} \times \phi_l(\mathbf{x}^t) \end{aligned} \tag{22}$$

$$\begin{aligned} \frac{\partial y}{\partial \sigma_X} =& \frac{g\frac{\partial h}{\partial \sigma_X} - h\frac{\partial g}{\partial \sigma_X}}{g^2} \\ =& \frac{g\sum_{l=1}^{M} \lambda^l \frac{\partial \omega_l}{\partial \sigma_X} - h\sum_{l=1}^{M} \frac{\partial \omega_l}{\partial \sigma_X}}{g^2} \end{aligned} \tag{23}$$

$$\frac{\partial \omega_l}{\partial \sigma_X} = \frac{\partial}{\partial \sigma_X} \prod_{i=1}^{p} \exp\left\{ -\frac{(x_i^{(t)} - m_{F_i^l})^2}{2(\sigma_X^2 + \sigma_{F_i^l}^2)} \right\} = \omega_l \sigma_X \sum_{i=1}^{p} \frac{(x_i^{(t)} - m_{F_i^l})^2}{(\sigma_X^2 + \sigma_{F_i^l}^2)^2} \tag{24}$$

$$\frac{\partial y}{\partial \sigma_X} = \frac{1}{g}\sum_{l=1}^{M}(\lambda^l - y)\sigma_X\omega_l \sum_{i=1}^{p}\frac{(x_i^{(t)} - m_{F_i^l})^2}{(\sigma_X^2 + \sigma_{F_i^l}^2)^2} \tag{25}$$

$$\begin{aligned}\sigma_X(t+1) &= \sigma_{F_i^l}(t) - \beta_\sigma[y(\mathbf{x}^t) - y^{(t)}]\frac{\partial y}{\partial \sigma_{F_i^l}}|_t \\ &= \sigma_{F_i^l}(t) - \beta_\sigma[y(\mathbf{x}^t) - y^{(t)}] \times \frac{1}{g}\sum_{l=1}^{M}(\lambda^l - y)\sigma_X w_l \sum_{i=1}^{p}\frac{(x_i^{(t)} - m_{F_i^l})^2}{(\sigma_X^2 + \sigma_{F_i^l}^2)^2}\end{aligned} \tag{26}$$

5.2 Back-Propagation: Non-singleton Fuzzification

The same as we design Back-Propagation: Singleton Fuzzification in sect.4.2, only two fuzzy sets were used for each of the four antecedents. hence, the number of rules is very small and equals $2^4 = 16$,hence, Each rule is characterized by eight antecedent MF parameters and one consequent parameter,$\bar{y}$. The initial location of each Gaussian antecedent MF was based on the mean, m_x and the standard deviation σ_x of the data in the 504 training samples. More specifically, the means of each and every antecedent's two Gaussian MFs were initially chosen as $m_x - 2\sigma_x = 0.3363$ or $m_x - 2\sigma_x = 1.4418$,respectively, and their standard deviations were initially chosen as $2\sigma_s = 0.5528$. The center of each consequent'sMF,$\bar{y}^i$,was initially chosen to be a random number from the interval $[0, 1]$.

Each fuzzy system was tuned using a steepest descent algorithm in which all of the learning parameters were set equal to the same $\beta_\theta = 0.1$. Training and testing were carried out for six epochs,after each epoch the testing data were used to evaluate how each fuzzy system performed, by computing $RMSE_{ns}$ using (19),the results are as follows:

$$RMSE_{ns} = \{0.0516, 0.0471, 0.0441, 0.0332, 0.0272, 0.0258\}$$

Next, we plot the $RMSE_{ns}$ we obtained by using singleton fuzzy logic system(SFLS) and Non-sigleton fuzzy logic system(NSFLS) from the same Mackey-Glass chaotic time series in Figure 3. From the figure, we can easily see that both FLS's $RMSE_{ns}$ converge to roughly the same value, but the NSFLS's $RMSE_{ns}$ is much smaller than the SFLS's after the first epoch, and it converge faster, which means NSFLS has better performance in Forecasting the time serise with noise.

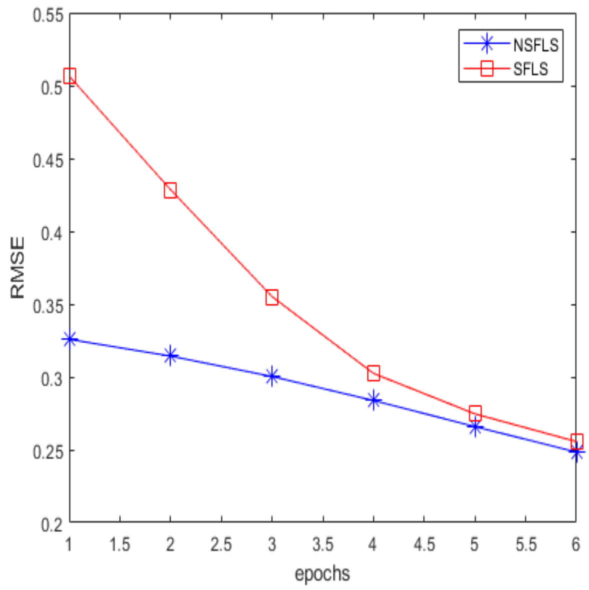


Figure 3: The $RMSE_{ns}$ of the same Mackey-Glass time series using SFLS and NSFLS

References

[1] Mackey, M.C., and L. Glass. 1977. Oscillation and chaos in physiological control systems. Science 197: 287–289.

[2] Mendel, J.M. 1995a. Fuzzy logic systems for engineering: A tutorial. Proceedings of the IEEE 83: 345–377.

[3] Jang, J.-S.R. 1993. ANFIS: Adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23: 665–684.

[4] George C. Mouzouris,Jerry M. Mendel. Nonsingleton fuzzy logic systems: theory and application.[J]. IEEE Trans. Fuzzy Systems,1997,5(1).