Create a reservoir network to solve a given task. Make slight changes to it to improve its performance on the same tests.

This task need to create a reservoir netowrk.

I do some research, and i know about Reservoir computing, is a framework for computation derived from recurrent neural network theory that maps input signals into higher dimensional computational spaces through the dynamics of a fixed, non-linear system called a reservoir.

And then i know that, Echo state network is a type of reservoir computing.

First. I have a look at [4], this paper have a summary about the Echo state networks. And then i konw that there is a type of combation with deep learning and Echo state networks, called DeepESN, the paper is [3].

And then i found that, the [6] have a simple example, use the Echo state networkd to do a prediction.

So in this task, I decide to use the Echo state network(ESN) in the MackeyGlass to do a predictive.

Echo state network

Echo state networks (ESN) provide an architecture and supervised learning principle for recurrent neural networks (RNNs). The main idea is (i) to drive a random, large, fixed recurrent neural network with the input signal, thereby inducing in each neuron within this "reservoir" network a nonlinear response signal, and (ii) combine a desired output signal by a trainable linear combination of all of these response signals.

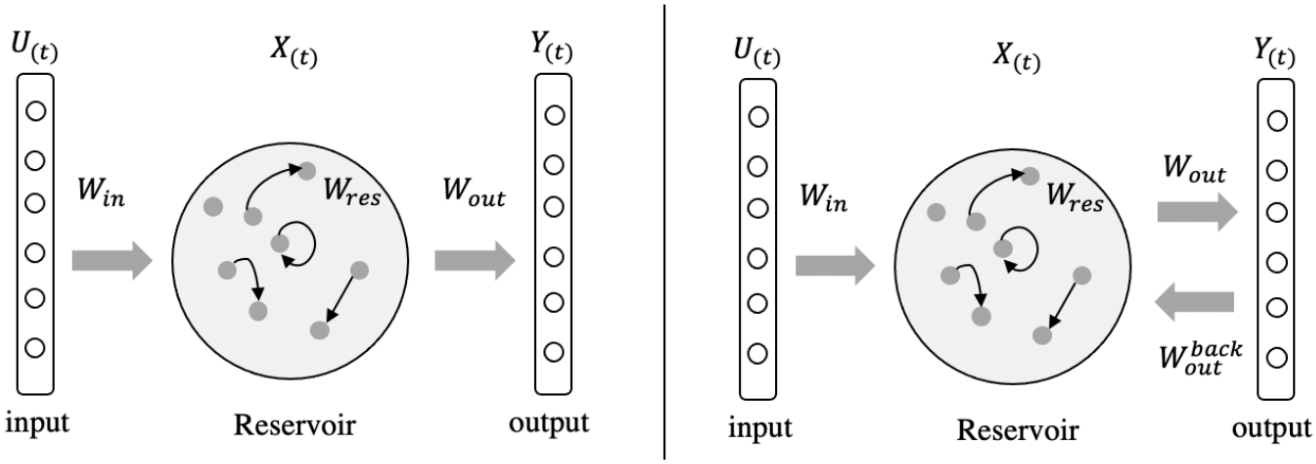


Figure 1: Basic echo state networks

It is also composed of , input layer, hidden layer, and output layer.
It designs the hidden layer as a sparse network composed of many neurons, and achieves the function of memorizing data by adjusting the characteristics of the internal weight of the network.
Its internal dynamic reserve pool (DR) contains a large number of sparsely connected neurons, contains the operating state of the system, and has a short-term training memory function.
The process of ESN training is the process of training the connection weight (Wout) from the hidden layer to the output layer.

The connection from the input layer to the reserve pool is $Win(N * K)$, and the connection from the reserve pool to the next time the reserve pool state is $W(N * N)$. The connection from the pool to the output layer is $Wout(L * (K + N + L))$.
In addition, there is a connection $Wback(N * L)$ from the output layer at the previous moment to the reserve pool at the next moment. This connection is not necessary.

Every time u(t) is input, the state of the reserve pool must be updated, and its state update equation is:

$$x(t+1) = f(W_{in} * u(t+1) + W_{back}x(t))$$

Both Win and $Wback$ are randomly initialized when the network is initially established, and are fixed.
$u(t+1)$ is the input at this moment, $x(t+1)$ is the state of the reserve pool at this moment, and $x(t)$ is the state of the reserve pool at the last moment can be initialized with 0 at t=0.
f is the internal neuron activation function of DR, and the hyperbolic tangent function (tanh) is usually used.

x is a vector of length N.

The output state equation of ESN is:

$$y(t+1) = f_{out} * (W_{out} * (u(t+1), x(t+1)))$$

$fout$ is the neuron activation function of the output layer.

Now that we have the state of the reserve pool, and with the ESN output method, W_{out} can be determined according to the target output $y(target)$, so that the loss between $y(t+1)$ and $y(target)$ is as small as possible. This is a simple linear regression problem.

Reservior

The reserve pool is the core structure of the network.
The reserve pool is a recursive structure of randomly generated, large-scale, sparse connections (SD usually maintains 1%-5% connections).

The final performance of the ESN is determined by the various parameters of the reserve pool, including: the internal connection weight spectrum radius SR of the reserve pool, the reserve pool size N, the reserve pool input unit scale IS, and the reserve pool sparseness SD.

1. The internal connection weight spectrum radius SR of the reserve pool.
   It is the eigenvalue with the largest absolute value of the connection weight matrix W, denoted as λmax, $\lambda max < 1$ is a necessary condition to ensure the stability of the network
   $\lambda_{max} = max(abs(W))$
2. The size of the reserve pool N.
   It is the number of neurons in the reserve pool. The size of the reserve pool is related to the number of samples, which has a great impact on network performance. The larger the reserve pool size, the more accurate the ESN description of a given dynamic system will be Overfitting problem.
3. The storage tank input unit scale IS.
   It is a scale factor that needs to be multiplied before the input signal of the reserve pool is connected to the neurons in the reserve pool, that is, the input signal is scaled to a certain extent. Generally, the stronger the nonlinearity of the object to be processed, the larger the IS.
4. The sparseness of the reserve pool SD.
   It represents the connection between neurons in the reserve pool, and not all neurons in the reserve pool have connections. SD represents the percentage of the total number of interconnected neurons in the reserve pool to the total neuron N. The larger the value, the stronger the nonlinear approximation ability.

ESN network training

The training process of ESN is the process of determining the coefficient and outputting the connection weight matrix $Wout$ according to the given training sample.
The training is divided into two stages: sampling stage and weight calculation stage.

For the sake of simplicity, it is assumed that $Wback$ is 0, and the input-to-output and output-to-output connection weights are also assumed to be 0.

Sampling stage

In the sampling phase, the initial state of the network is selected arbitrarily, but usually the initial state of the network is selected as 0, that is, $x(0) = 0$.

1. The training samples $(u(t), t = 1, 2, ..., P)$ are added to the reserve pool through the input connection weight matrix Win.
2. According to the aforementioned two state equations, complete the calculation and collection of the system state and output $y(t)$ in sequence.

In order to calculate the output connection weight matrix, the internal state variables need to be collected (sampled) from a certain time m, and the vector $(x_1(i), x_2(i), ..., x_N(i))(i = m, m + 1, ..., P)$ is used as the row to form the matrix $B(P - m + 1, N)$, and the corresponding sample data $y(t)$ is also are collected and form a column vector $T(P - m + 1, 1)$.

Weight calculation stage

The weight calculation is to calculate the output connection weight $Wout$ based on the system state matrix and sample data collected in the sampling phase. Because there is a linear relationship between the state variable $x(t)$ and the predicted output $\widehat{y}(t)$, the goal that needs to be achieved is to use the predicted output to approximate the expected output $y(t)$:

$$y(k) \approx \widehat{y}(k) = \sum_{i=1}^{L} W_i^{out} x_i(k)$$

$$\min \frac{1}{P - m + 1} \sum_{k=m}^{P} (y(k) - \sum_{i=1}^{L} W_i^{out} x_i(k))^2$$

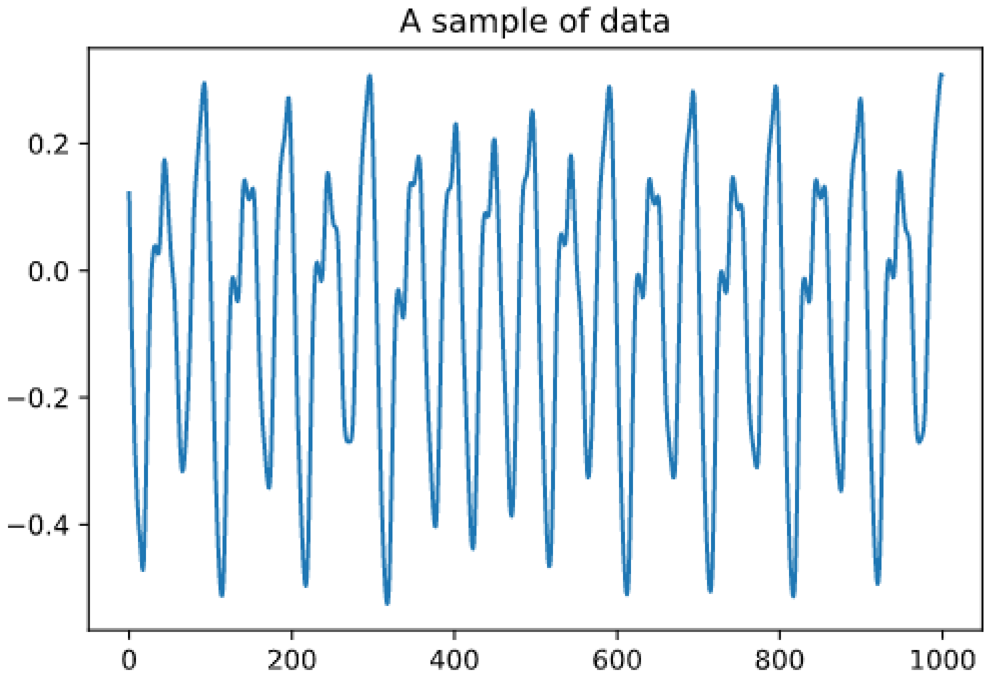
$$W_{out} = (M^{-1} \times T)^T$$

> M is the input of $x_1(k), x_2(k), ..., x_n(k), k = m,$
> $m + 1, ..., P$ forms a matrix of $(P - m + 1) \times N,$
> T is the output of the $y(k)$ forms a matrix of $(P - m + 1) \times 1$

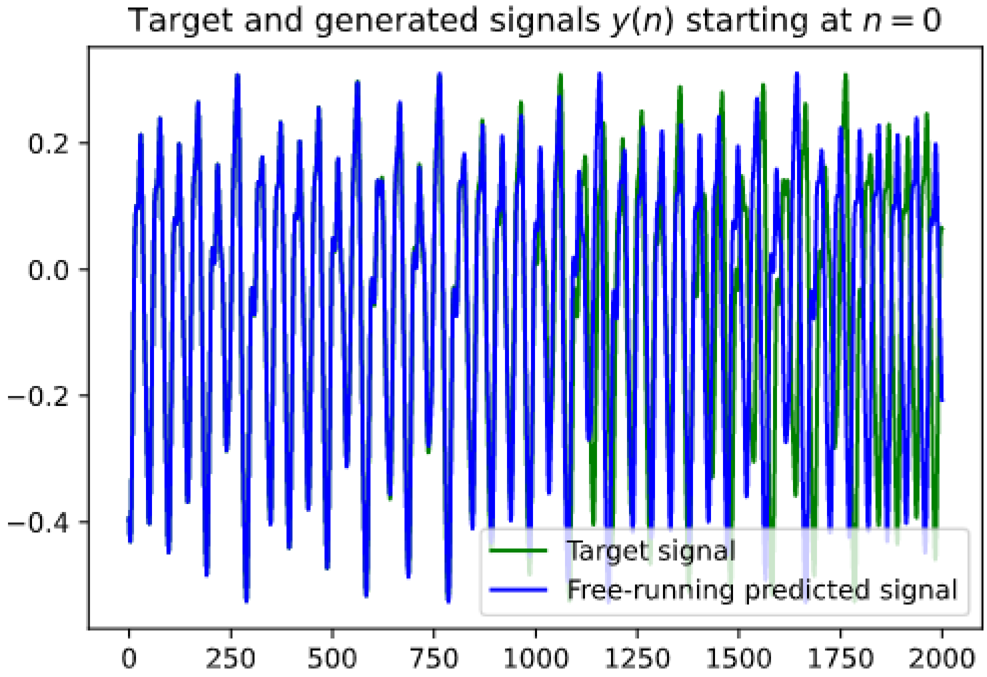
At this point, the ESN network training has been completed, and the trained network can be used for specific problems in time series modeling.

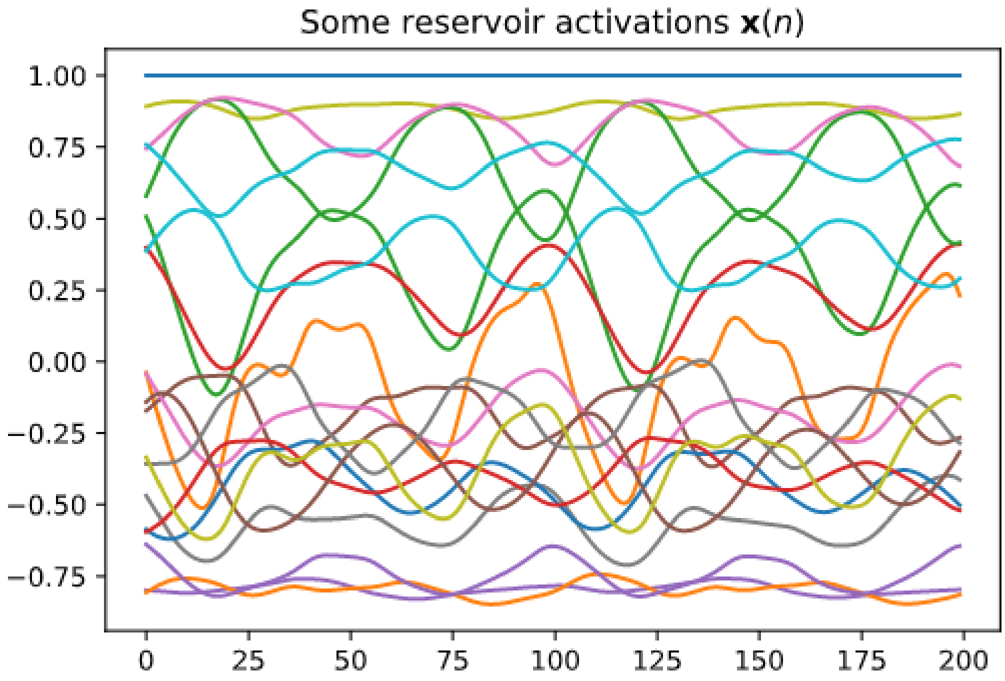
Result

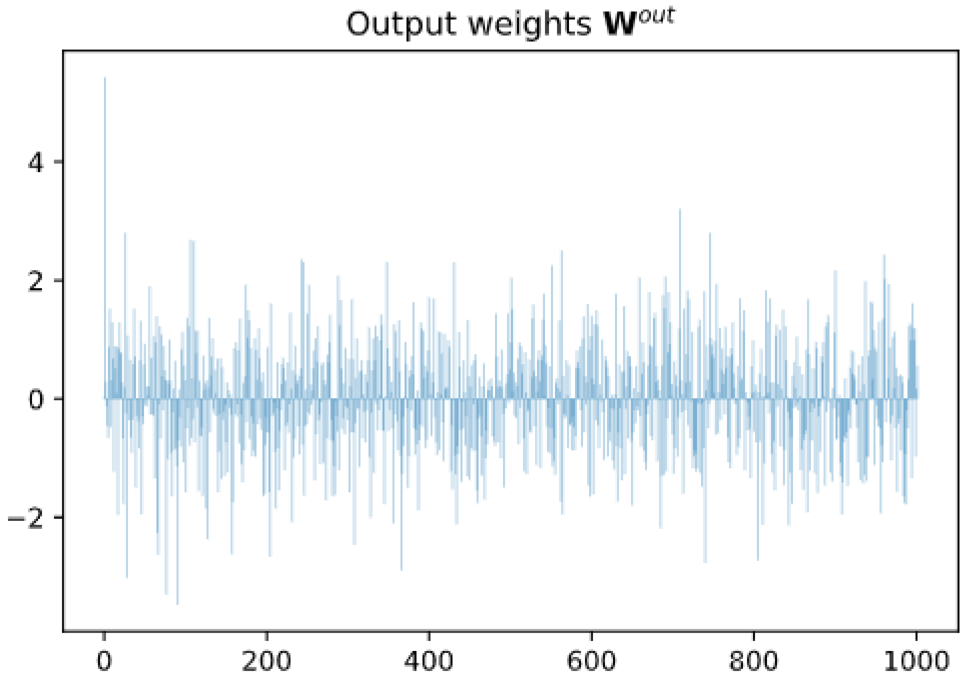
So in this time, I ues the Mackey-Glass time series to do this implement.
First to draw some data from the Mackey-Glass time series.



And then, after the ESN calculation, output some result and splot some signals.







Reference

1. Reservoir computing(wikipad)
2. Schaetti, N., Salomon, M., & Couturier, R. (2016). Echo State Networks-Based Reservoir Computing for MNIST Handwritten Digits Recognition. 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), 484-491.

3. Gallicchio C, Micheli A. Deep echo state network (deepesn): A brief survey[J]. arXiv preprint arXiv:1712.04323, 2017.
4. Sun C, Song M, Hong S, et al. A Review of Designs and Applications of Echo State Networks[J]. arXiv preprint arXiv:2012.02974, 2020.
5. Echo state network
6. 回声状态网络(ESN)原理详解(附源码实现)