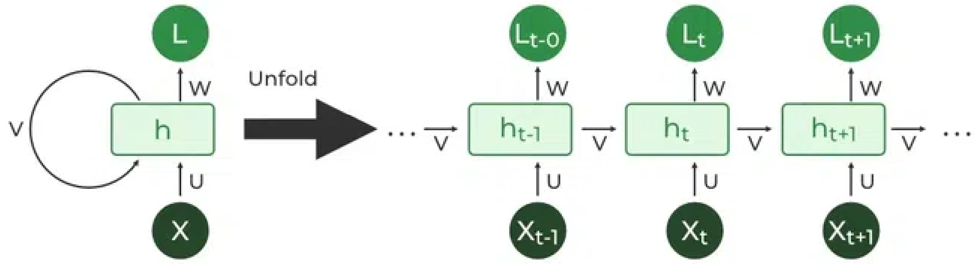

Fig. 5 Picture fuzzy-based Recurrent Neural Network

3.2.1 Functioning of PF-RNN

The current hidden state $h_t = [\mu_t^h, \eta_t^h, \nu_t^h]$ depends on the previous state $h_{t-1} = [\mu_{t-1}^h, \eta_{t-1}^h, \nu_{t-1}^h]$ and the current input $x_t = [\mu_t^x, \eta_t^x, \nu_t^x]$, and is calculated using the following relations:

State Update:
$h_t = f(h_{t-1}, x_i)$, where h_t, h_{t-1}, and x_i are the variables in the form of PFNs and represent the current state, previous state and the input of the current time step respectively.
Activation Function Application:

$$h_t = \begin{cases} V_{hh}.h_{t-1} + U_{hx}.x_t & SIM(x_t, x_j) \neq 0, j = 1, 2, ..., t-1 \\ h_{t-1} & SIM(x_t, x_j) = 0, j = 1, 2, ..., t-1 \end{cases}$$

Here, $V_{hh} = [v_1^{hh}, v_2^{hh}, v_3^{hh}]$ is the weight matrix for the recurrent neuron, and $U_{hx} = [u_1^{hx}, u_2^{hx}, u_3^{hx}]$ is the weight matrix for the input neuron. The $SIM(\cdot, \cdot)$ operation in the activation function measures the similarity between two inputs in the form of PFNs.
Output Calculation:
$y_t = W_{hy}.h_t$, where $y_t = [\mu_t^y, \eta_t^y, \nu_t^y]$ is the output and $W_{hy} = [w_1^{hy}, w_2^{hy}, w_3^{hy}]$ is the weight at the output layer.
These parameters are updated using backpropagation. However, since RNN works on sequential data, we use an updated backpropagation, known as backpropagation through time.
Generate the random 7^{th} dimensional one thousand data set and Implement the modern recurrent neural network known as PF-RNN using python.