Recurrent Neural networks

Machine intelligence

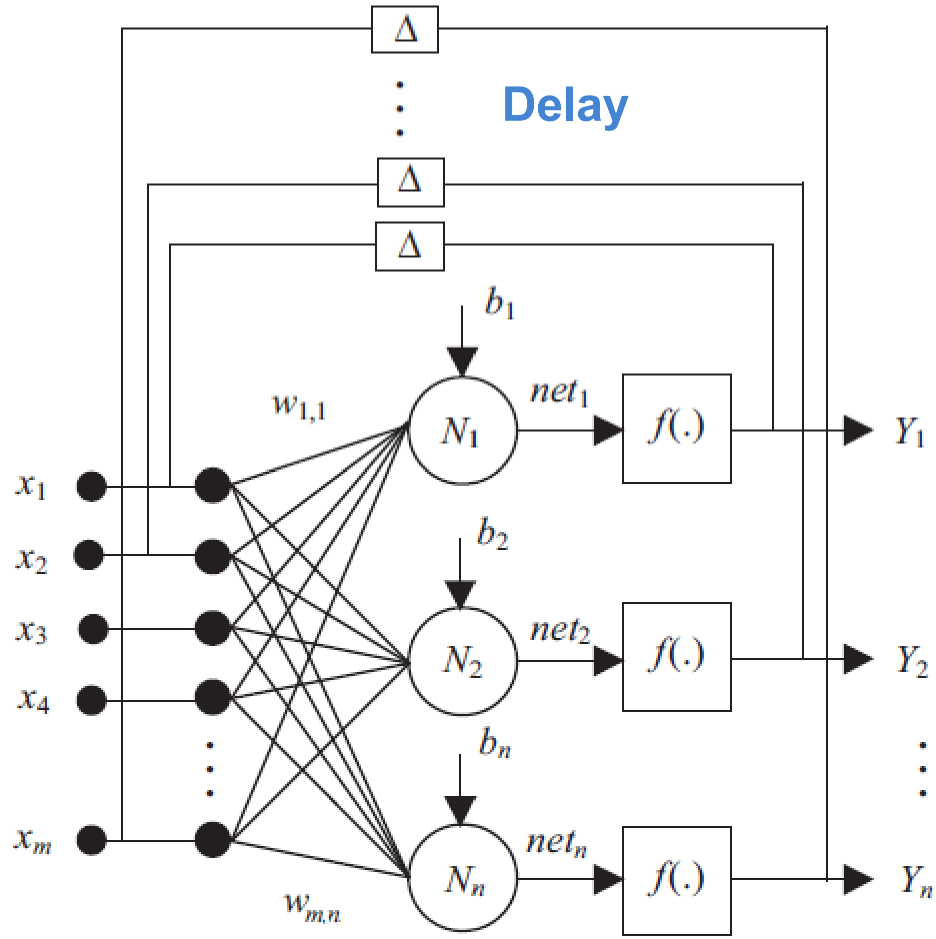
Dr. Ahmad Al-Mahasneh

Outline

- Recurrent networks
- Elman Networks
- Jordan Networks
- Example

Recurrent Neural Networks

- A **recurrent network** is obtained from the feedforward network by **adding a connecting the neuron's output to their inputs**.
- Recurrent networks are also called **feedback networks**



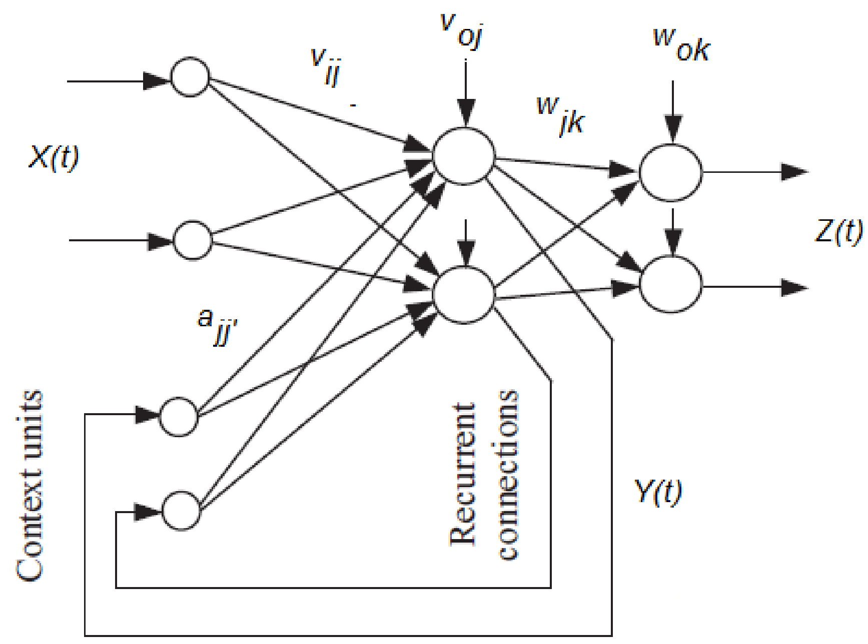
Elman Networks

- Elman networks are **three-layer backpropagation networks** with the addition of a **feedback** connection from the output of the **hidden** layer to its **input**.

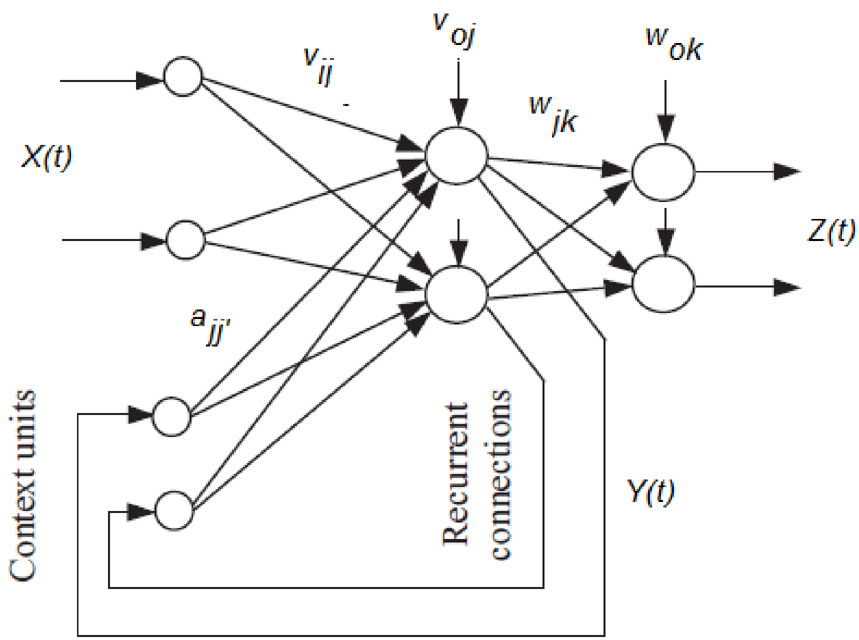
- The Elman architecture have an extra layers of neurons that copy the current activations in the hidden-layer neurons, and after **delaying** these values for **one time unit**, feed them back **as additional inputs** into the hidden layer neurons.

Elman Networks

- The Elman network will therefore have **three layers**:
  1. **Input layer** that consists of two different groups of neurons: **external inputs** and **internal inputs**
  2. **Hidden layer**
  3. **Output layer**



Elman Network Equations



$$Z = f(W^T Y + W_o)$$

$$Y = f(V^T X + V_o + A^T Y)$$

Y is a function of Y ??

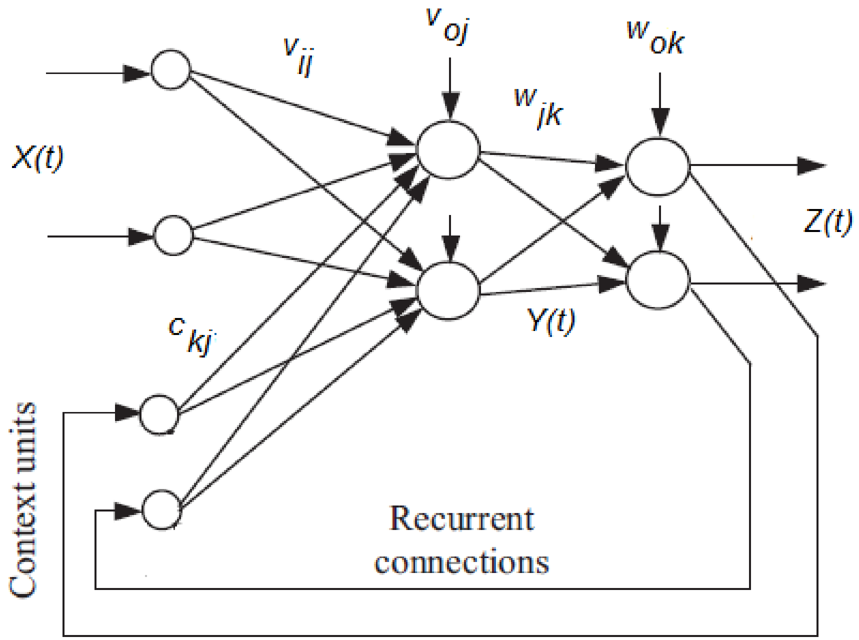
But there is a delay! And that is why we use '**t**' to represent the time sequence

$$Y(t) = f(V^T X + V_o + A^T Y(t-1))$$

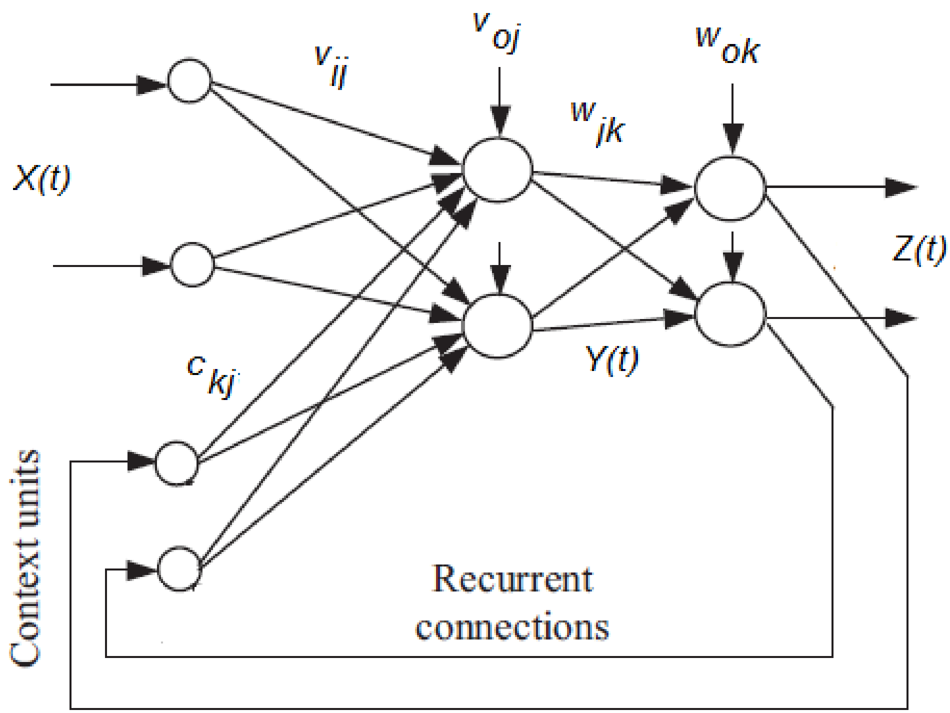
Jordan Networks

- **Jordan networks** are **three-layer networks**, with the main **feedback** connections taken **from the output layer to the input layer**.

- The Jordan Network can be trained using the standard BP algorithm



Jordan Network Equations



$$Z = f(W^T Y + W_o)$$

$$Y = f(V^T X + V_o + C^T Z)$$

Use BP to update the weights

$$E = \frac{1}{2}\sum_{k=1}^{n}(z_k - d_k)^2$$

- $Z = f(W^T Y + W_o)$

- **Step One: Output Weights (same as regular BP equations for feedforward)**

- $\delta_k = \quad z_k - d_k \; z'_k$

- $w_{jk} = w_{jk} - \alpha \delta_k y_j$

Use BP to update the weights

$$E = \frac{1}{2}(z_k - d_k)^2$$

$$Z = f(W^T Y + W_o)$$

$$Y = f(V^T X + V_o + C^T Z)$$

Step Two: Hidden Weights from inputs (same as regular BP feedforward)

$$\frac{\partial E}{\partial v_{ij}} = \left(\frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial y_j}\right)\frac{\partial y_j}{\partial v_{ij}}$$

$$\frac{\partial E}{\partial v_{ij}} = (\delta_k w_{jk})\, y'_j x_i$$

$$v_{ij} = v_{ij} - \alpha \delta_j x_i$$

where

$$\delta_j = y'_j\, (\delta_k w_{jk})$$

Step Three: Hidden Weights from output feedback

$$\frac{\partial E}{\partial c_{kj}} \quad \left(\frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial y_j}\right)\frac{\partial y_j}{\partial c_{kj}}$$

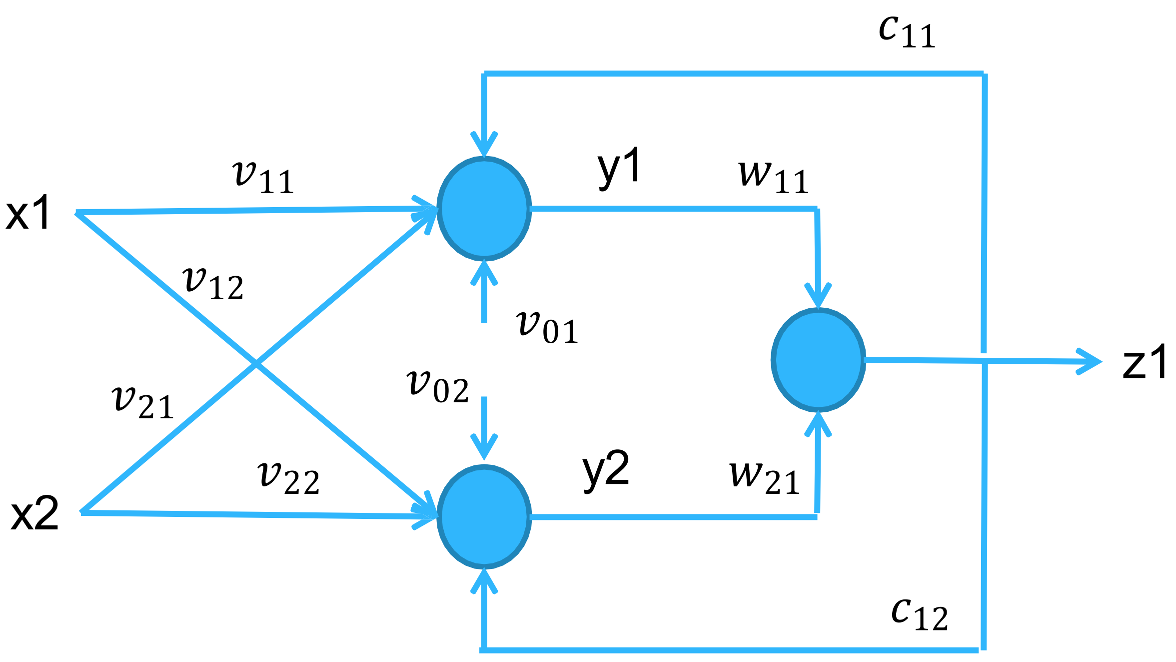
$$\frac{\partial E}{\partial c_{kj}} = (\delta_k w_{jk})\, y'_j z_k$$

$$c_{kj} = c_{kj} - \alpha \delta_j z_k$$

where

$$\delta_j = y'_j\, \delta_k w_{jk}$$

Example: Simple Jordan Network

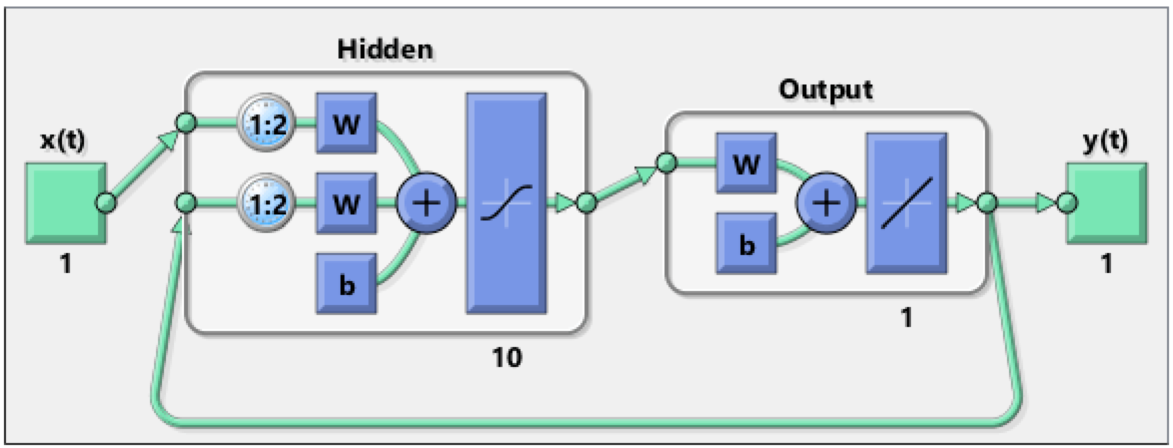


$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = f\left(\begin{bmatrix} v_{11} & v_{21} \\ v_{12} & v_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} v_{01} \\ v_{02} \end{bmatrix} + \begin{bmatrix} c_{11} \\ c_{12} \end{bmatrix} z_1\right)$$

$$z_1 = g\left(\begin{bmatrix} w_{11} & w_{21} \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right)$$

NARX

The nonlinear autoregressive network with exogenous inputs (NARX) is a recurrent dynamic network



References

- Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing (Chapter 4) by Siddique and Adeli. Wiley Publishing 2013

- Neural Networks and Learning Machine (Chapter 15) by Simon Haykin 3rd Edition. Pearson 2009

End

Thank you