

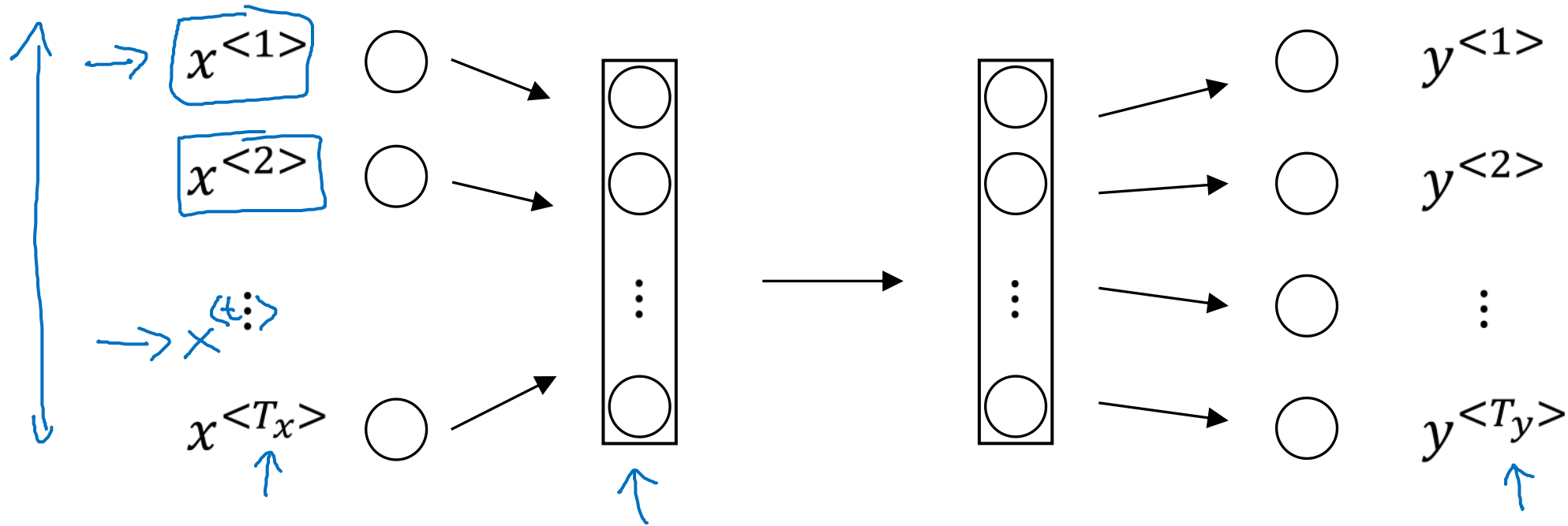


deeplearning.ai

Recurrent Neural Networks

Recurrent Neural Network Model

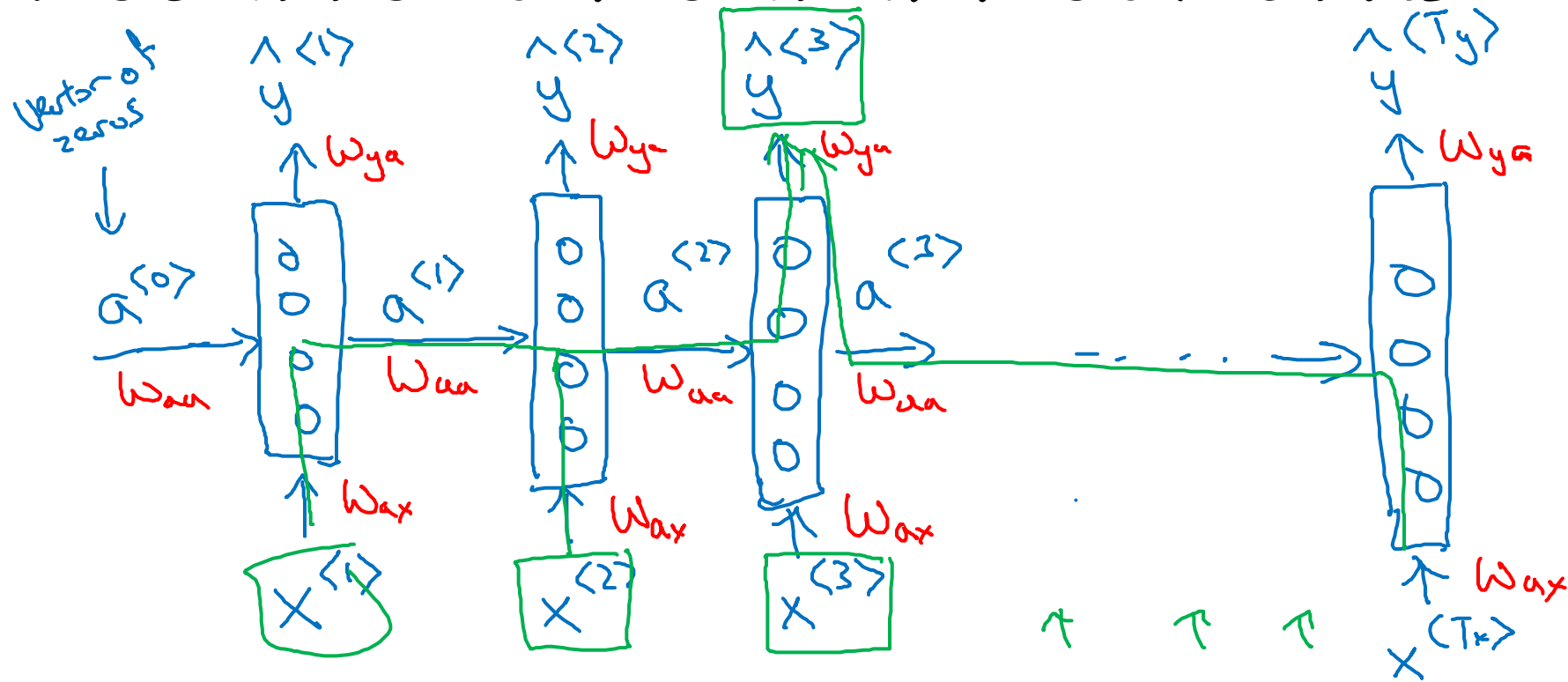
Why not a standard network?



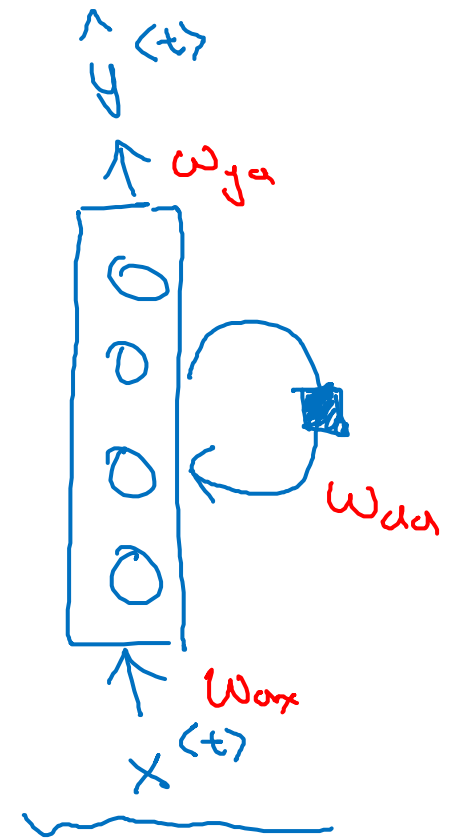
Problems:

- - Inputs, outputs can be different lengths in different examples
- - Doesn't share features learned across different positions of time

Recurrent Neural Networks $T_x = T_y$



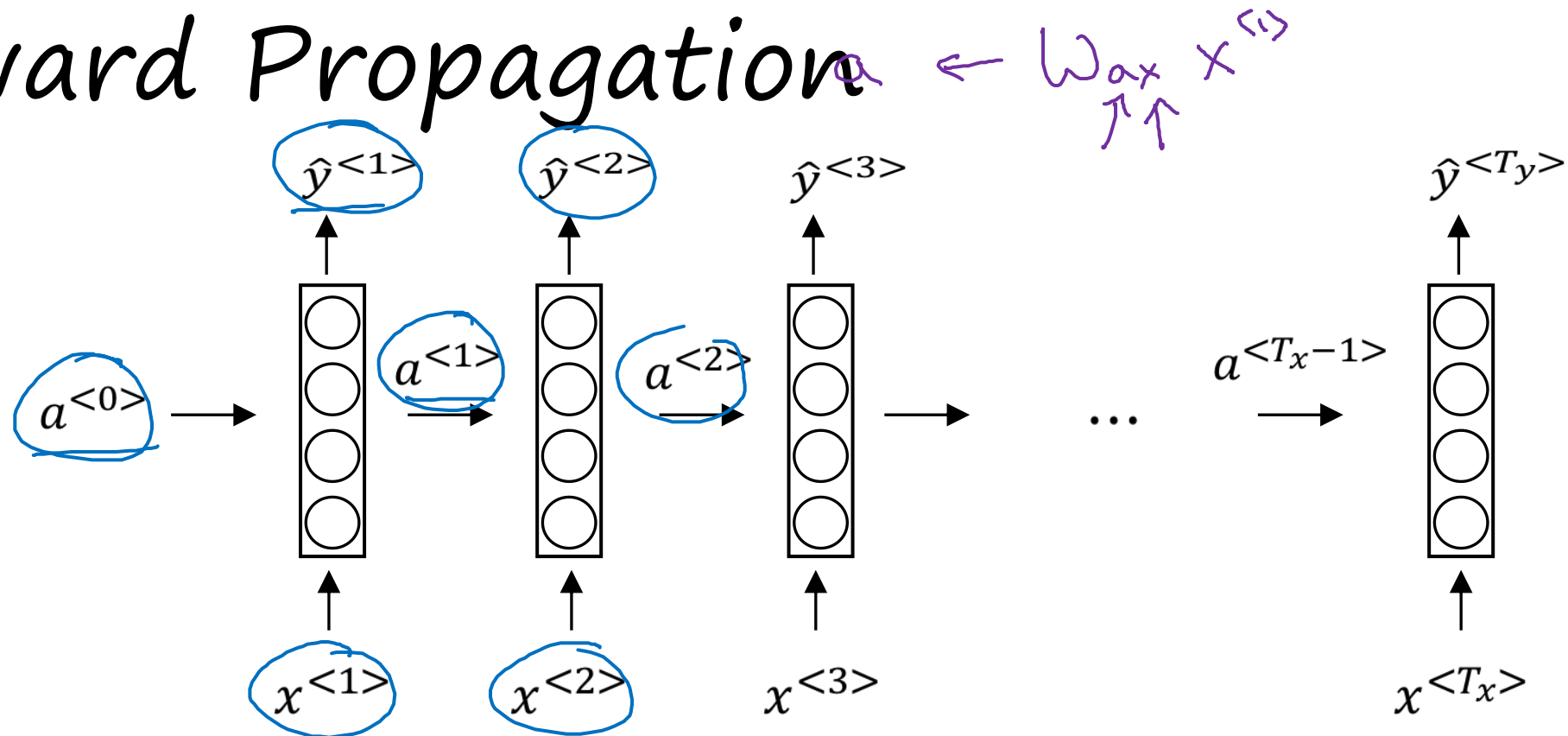
Bidirectional RNN (BRNN)



He said, "Teddy Roosevelt was a great President."

He said, "Teddy bears are on sale!"

Forward Propagation



$$a^{<0>} = \vec{0}$$

$$\underline{a}^{<1>} = g_1(W_{aa}a^{<0>} + \underline{W_{ax}}x^{<1>} + b_a) \leftarrow \underline{\tanh / \text{Relu}}$$

$$\underline{\hat{y}}^{<1>} = g_2(\underline{W_{ya}}a^{<1>} + b_y) \leftarrow \text{Sigmoid}$$

$$\begin{aligned} a^{<t>} &= g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \\ \hat{y}^{<t>} &= g(W_{ya}a^{<t>} + b_y) \end{aligned}$$

Simplified RNN notation

$$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

Diagram annotations: W_{aa} is a 100×100 matrix, and W_{ax} is a $100 \times 10,000$ matrix. Arrows indicate the flow of data from $a^{<t-1>}$ and $x^{<t>}$ into the equation.

$$\hat{y}^{<t>} = g(W_{ya}a^{<t>} + b_y)$$

$$y^{<t>} = g(W_y a^{<t>} + b_y)$$

Diagram annotations: Arrows indicate the flow of data from $a^{<t>}$ and b_y into the equation.

$$a^{<t>} = g(W_a [a^{<t-1>}, x^{<t>}] + b_a)$$

Diagram annotations: W_a is a 100×10100 matrix. The input vector $[a^{<t-1>}, x^{<t>}]$ is shown as a concatenation of $a^{<t-1>}$ (100) and $x^{<t>}$ (10,000).

$$\begin{bmatrix} W_{aa} & W_{ax} \end{bmatrix} = W_a$$

Diagram annotations: W_{aa} is 100×100 and W_{ax} is $100 \times 10,000$. The combined matrix W_a is 100×10100 .

$$[a^{<t-1>}, x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix}$$

Diagram annotations: $a^{<t-1>}$ is 100, $x^{<t>}$ is 10,000, and the concatenated vector is 10100.

$$\begin{bmatrix} W_{aa} & W_{ax} \end{bmatrix} \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix} = W_{aa}a^{<t-1>} + W_{ax}x^{<t>}$$