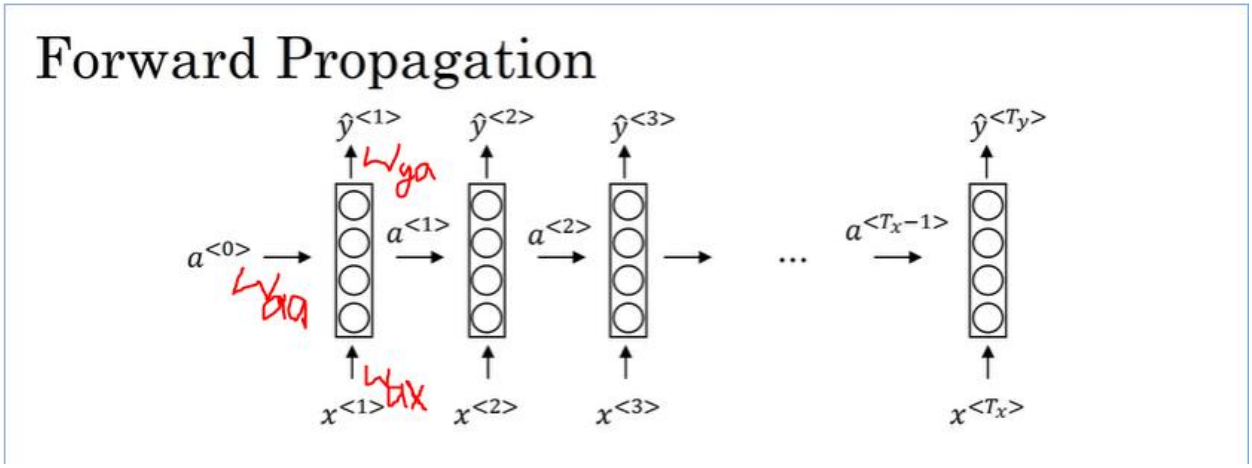
Why not a standard network ?

1. Inputs and outputs can be of different lengths in different example, like music generation, sentiment classification for a given input sequence.
2. Doesn't share features learned previously,
   i.e. for name entity recognition if a word harry is learned as person then in the feature instance if harry appears then we can use previous learning to reduce cost.

RNN model :

Recurrent Neural Network Model



- Every time RNN passes the previous activation solving the problem 2 mentioned above.
- For every time stamp we pass the same weights W_{aa} , W_{ax} for activation and W_{ya} to calculate the output $y^{<1>}$ (predicted value).

$a^{<0>}$ is a zero vector.

$a^{<1>} = g_1(W_{ax}X^{<1>} + W_{aa}\, a^{<0>} + b_a)$, $y^{<1>}_{pred} = g_2(W_{ya}\, a^{<1>} + b_y)$

generalizing for t timestamp we get :

$$a^{<t>} = g_1(W_{ax}X^{<t>} + W_{aa}\, a^{<t-1>} + b_a)$$

$y^{<t>}_{pred} = g_2(W_{ya}\, a^{<t>} + b_y)$, Where g_1 and g_2 are different activation functions

We can also reduce the computation cost by combining the weights and (input, activation), i.e.

$$W_a = (W_{aa} \mid W_{ax})$$

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

$$then,\ a^{<t>} = g(W_a[X^{<t>}, a^{<t-1>}] + b_a) \text{ and}$$

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

Drawback with RNN is that it considers information only from previous steps but not from future,

Ex: In name entity recognition, suppose we have some training examples as below,

1. He said, ”Teddy Rosevelt was a great President” .
2. He said, “Teddy bears are on sale”.

Then if our model didn’t consider feature instance it will incorrectly classify teddy in second case as name which is not .