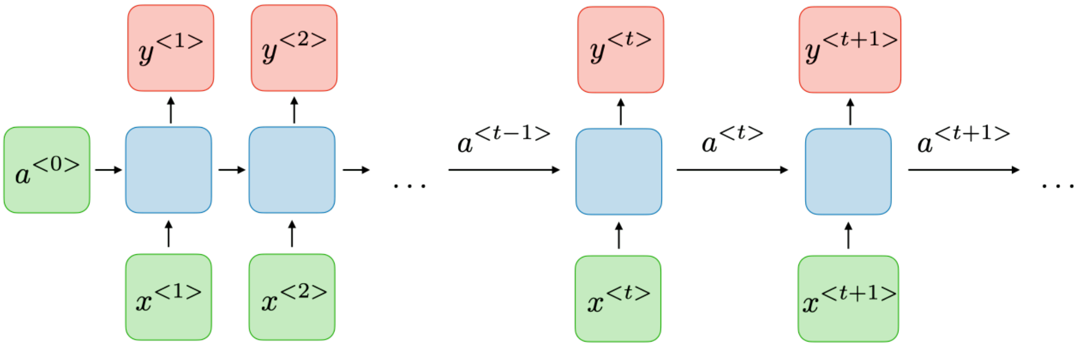
Recurrent Neural Network(RNN)

A Recurrent Neural Network is a Neural network that is specialised for processing sequence of data.

x(t) = x(1), x(2), x(2), ……., x(n) where, $t \in [1, n]$



$a^{<t>} = g1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$ $\qquad$ $y^{<t>} = g2(W_{ya}a^{<t>} + b_y)$

Where:

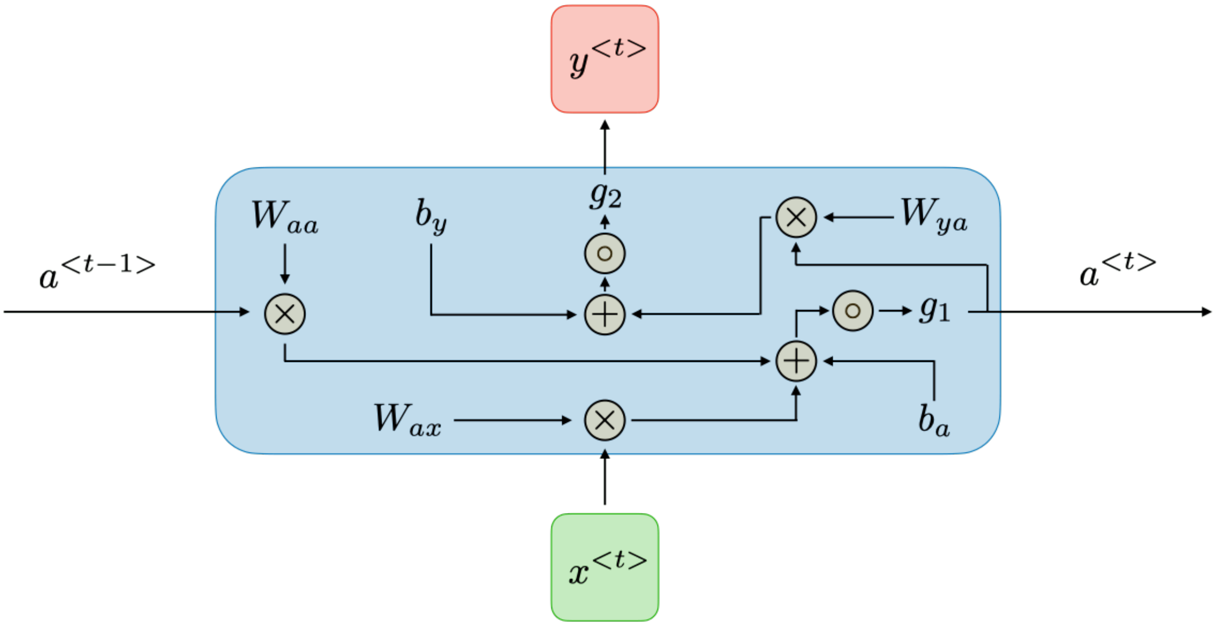
t : timestamp

$a^{<t>}$: activation

$y^{<t>}$: output

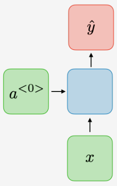
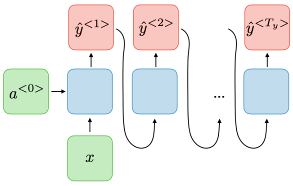
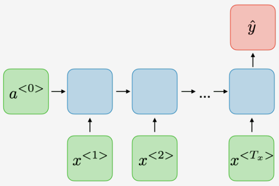
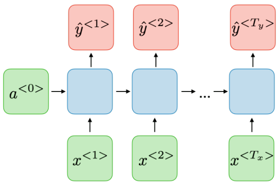
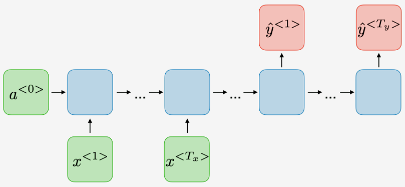
g1, g2 : activation function

$W_{aa}, W_{ax}, W_{ya}, b_a, b_y$: are the coefficients



Advantages	Drawbacks
Possibility of processing input of any length	Computation being slow
Model size not increase with size of input	Difficult of accessing information from a long time ago
Computation takes into account historical information	Cannot consider any future input for the current state
Weights are shared across time	

Applications of RNNs

Type of RNN	Illustration	Example
One-to-One $T_x(input) = T_y(output) = 1$		Traditional Neural network
One-to-Many $T_x = 1, T_x > 1$	 $\hat{y}^{<1>}$ $\hat{y}^{<2>}$ $\hat{y}^{<T_y>}$ $a^{<0>}$ ... x	Music generation
Many-to-One $T_x > 1, T_y = 1$	 $\hat{y}$ $a^{<0>}$ ... $x^{<1>}$ $x^{<2>}$ $x^{<T_x>}$	Sentiment Classification
Many-to-Many $T_x = T_x$	 $\hat{y}^{<1>}$ $\hat{y}^{<2>}$ $\hat{y}^{<T_y>}$ $a^{<0>}$ ... $x^{<1>}$ $x^{<2>}$ $x^{<T_x>}$	Name entity recognition
Many-to-Many $T_x \neq T_y$	 $\hat{y}^{<1>}$ $\hat{y}^{<T_y>}$ $a^{<0>}$ ... ... $x^{<1>}$ $x^{<T_x>}$	Machine Translation

Loss Function

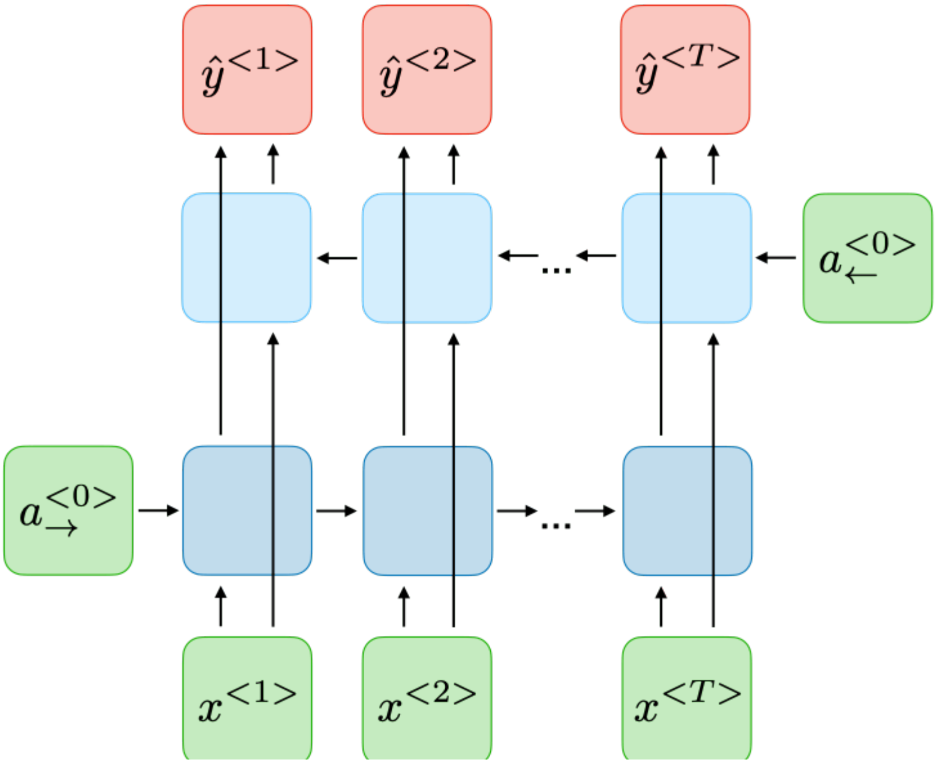
$$E^{(t)} = \Sigma(actual^{<t>} - predicted^{<t>})^2$$

Back propagation

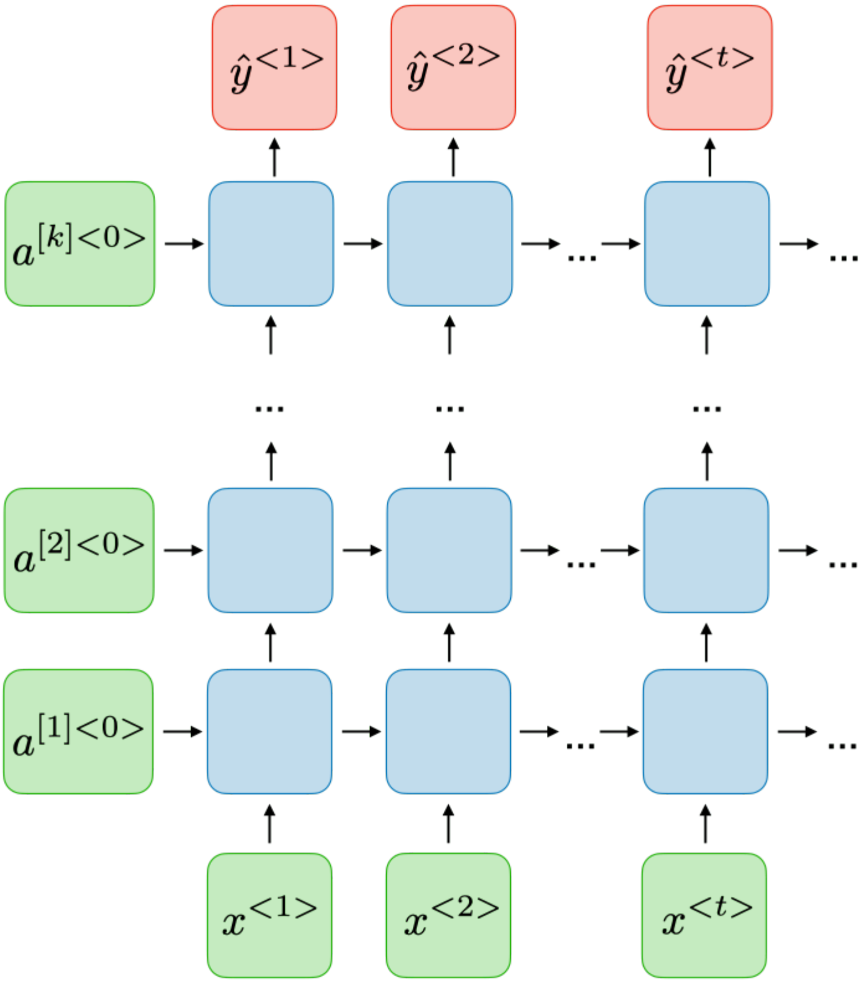
$$E^{'(t)} = \sum \frac{dE^{(t)}}{dw}|_t$$

Variants of RNN

1. Bidirectional



2. Deep RNN



$\hat{y}^{<1>}$
$\hat{y}^{<2>}$
$\hat{y}^{<t>}$
$a^{[k]<0>}$
...
...
...
...
...
$a^{[2]<0>}$
...
...
$a^{[1]<0>}$
...
...
$x^{<1>}$
$x^{<2>}$
$x^{<t>}$