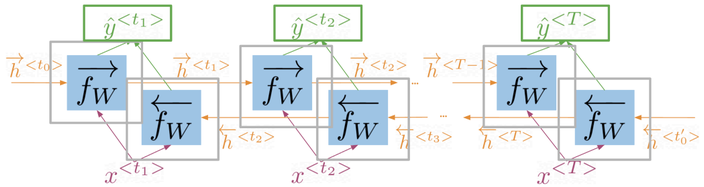
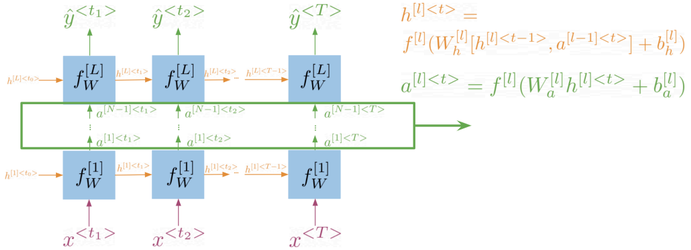
Deep and Bi-directional RNNs

Bi-directional RNNs are important, because knowing what is next in the sentence could give you more context about the sentence itself.



So you can see, in order to make a prediction $\hat{y}$, you will use the hidden states from both directions and combine them to make one hidden state, you can then proceed as you would with a simple vanilla RNN. When implementing Deep RNNs, you would compute the following.



$$h^{[l]<t>} = f^{[l]}(W_h^{[l]}[h^{[l]<t-1>}, a^{[l-1]<t>}] + b_h^{[l]})$$

$$a^{[l]<t>} = f^{[l]}(W_a^{[l]}h^{[l]<t>} + b_a^{[l]})$$

Note that at layer *l*, you are using the input from the bottom $a^{[l-1]}$ and the hidden state h^l. That allows you to get your new h, and then to get your new a, you will train another weight matrix Wa, which you will multiply by the corresponding h add the bias and then run it through an activation layer.

Mark as completed

Like Dislike Report an issue