# Stacked LSTM

Hugh Nguyen

8/14/2020

## Layer 1

This layers takes into the input sequence from 4 features: stock price, volume, price moving average and volume moving average. It tries to output the prediction then passes these predictions into the next layer to learn even more complex and abstracting representation of input sequence.

$$
\begin{aligned}
i_{(t)}^{(1)} &= \sigma(W_{xi}^T.x_{(t)} + W_{hi}^T.h_{(t-1)} + b_i) \\
f_{(t)}^{(1)} &= \sigma(W_{xf}^T.x_{(t)} + W_{hf}^T.h_{(t-1)} + b_f) \\
o_{(t)}^{(1)} &= \sigma(W_{xo}^T.x_{(t)} + W_{ho}^T.h_{(t-1)} + b_o) \\
g_{(t)}^{(1)} &= tanh(W_{xg}^T.x_{(t)} + W_{hg}^T.h_{(t-1)} + b_g) \\
c_{(t)}^{(1)} &= f_{(t)}^{(1)} \otimes c_{(t-1)}^{(1)} + i_{(t)}^{(1)} \otimes g_{(t)}^{(1)} \\
\hat{y}_{(t)}^{(1)} &= h_{(t)}^{(1)} = o_{(t)}^{(1)} \otimes tanh(c_{(t)}^{(1)})
\end{aligned}
$$

$x_{(t)}$ : input vector at time t

$\mathbf{b}$ : bias terms

$\mathbf{W}$ : weight matrices

$i_{(t)}^{(1)}$ : input gate computation

$f_{(t)}^{(1)}$ : forget gate computation

$o_{(t)}^{(1)}$ : output gate computation

$g_{(t)}^{(1)}$ : main layer output computation, it learns new information

$c_{(t)}^{(1)}$ : long term memory computation

$\hat{y}_{(t)}^{(1)}$ $h_{(t)}^{(1)}$ : estimated prediction computation

## Layer 2:

Layer 2 basically consumes the hidden features that approximates the short term relationship of the raw input from Layer 1 $h_{(1)}, ..., h_{(t)}$ then applies the exact same computation as above. (i.e. 2nd layer takes in a 1 by 10 vector output from the previous layer). Its final output will finalize the predicted values for a given input sequence by making use of both insights extracted from 1 layer to another.

$$
\begin{aligned}
\hat{y}_{(t)}^{(2)} &= h_{(t)}^{(2)} \\
&= \phi^{(2)}(h_t^{(1)}, h_{(t-1)}^{(2)})
\end{aligned}
$$