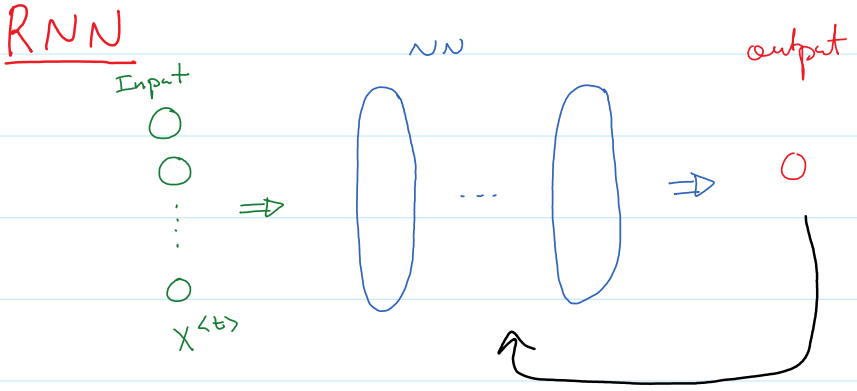
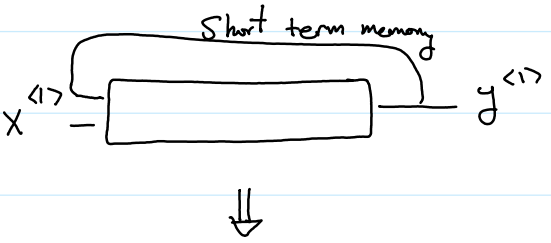
Recurrent Neural Networks

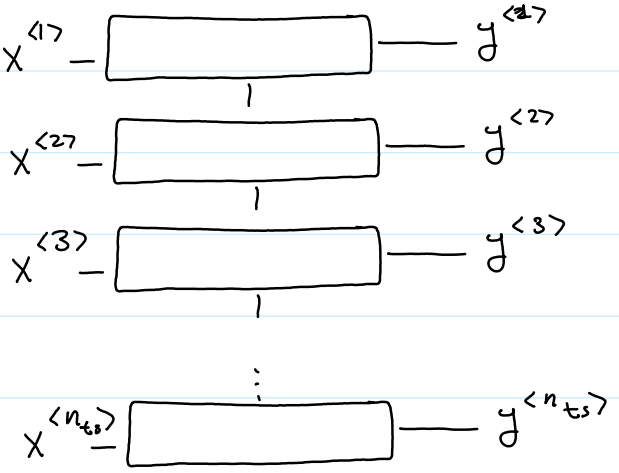
May 26, 2019 12:06 PM

RNN





$X^{\langle t \rangle}$ represents the t^{th} element in the time series.



Time

Notation :

X : A collection of time-series : shape = (n_s, n_{ts}, n_f)

- n_s : # Samples
- n_{ts} : # time-series in each Sample
- n_f : # features.

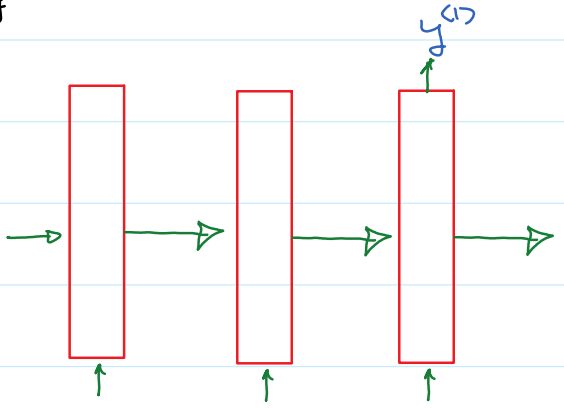
Y : A collection of labels : shape = (n_s, n_{ts_y})

- n_{ts_y} : length of y outputs

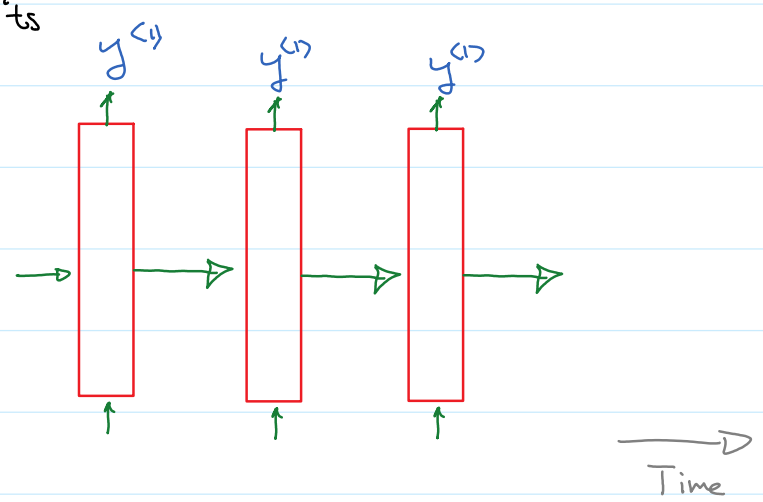
$n_{ts_y} = 1$ - Single output

n_{tsy} : 1 → Single outcome

length of y outputs



$n_{tsy} = n_{ts}$



$X^{(i)\langle t \rangle}$: t^{th} instance in $X^{(i)}$.

Mathematical representation of RNN:

$$a^{\langle t \rangle} = g_a\left(W_a a^{\langle t-1 \rangle} + U_a X^{\langle t \rangle} + b_a\right) \qquad g_a: \tanh/\text{Relu}$$

$$y^{\langle t \rangle} = g_y\left(W_y a^{\langle t-1 \rangle} + U_y X^{\langle t \rangle} + b_y\right) \qquad g_y: \text{Sigmoid} \ldots$$

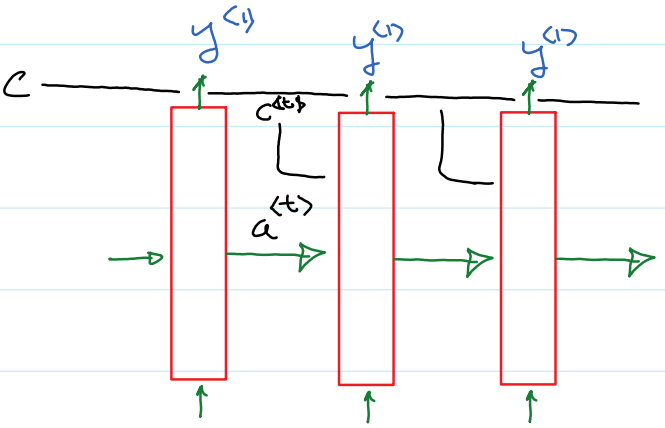
GRU & LSTM

§

LSTM

Issue with RNN: Vanishing / Exploding Gradient

Solution: Keep a memory



Memory:

* Store some info → Γ_s
* Forget " " → Γ_f
* Retrieve " " → Γ_r → output

LSTM

$$\tilde{c}^{\langle t \rangle} = \tanh\left(W_m a^{\langle t-1 \rangle} + U_m x^{\langle t \rangle} + b_m\right)$$

$$\Gamma_s = \text{sig}\left(W_s a^{\langle t-1 \rangle} + U_s x^{\langle t \rangle} + b_s\right)$$

$$\Gamma_f = \text{sig}\left(W_f a^{\langle t-1 \rangle} + U_f x^{\langle t \rangle} + b_f\right)$$

$$c^{\langle t \rangle} = \Gamma_s \tilde{c}^{\langle t \rangle} + \Gamma_f c^{\langle t-1 \rangle}$$

$$\Gamma_r = \text{sig}\left(W_r a^{\langle t-1 \rangle} + U_r x^{\langle t \rangle} + b_o\right)$$

$$a^{\langle t \rangle} = \Gamma_r c^{\langle t \rangle}$$

For GRU

$$\begin{cases} \Gamma_f = 1 - \Gamma_s \\ \Gamma_r = 1 \end{cases}$$

There's some additional Γ_I to indicate the impact of the memory.

$$\tilde{c}^{\langle t \rangle} = \tanh\left(W_m \Gamma_I a^{\langle t-1 \rangle} + U_m x^{\langle t \rangle} + b_m\right)$$

Intuition

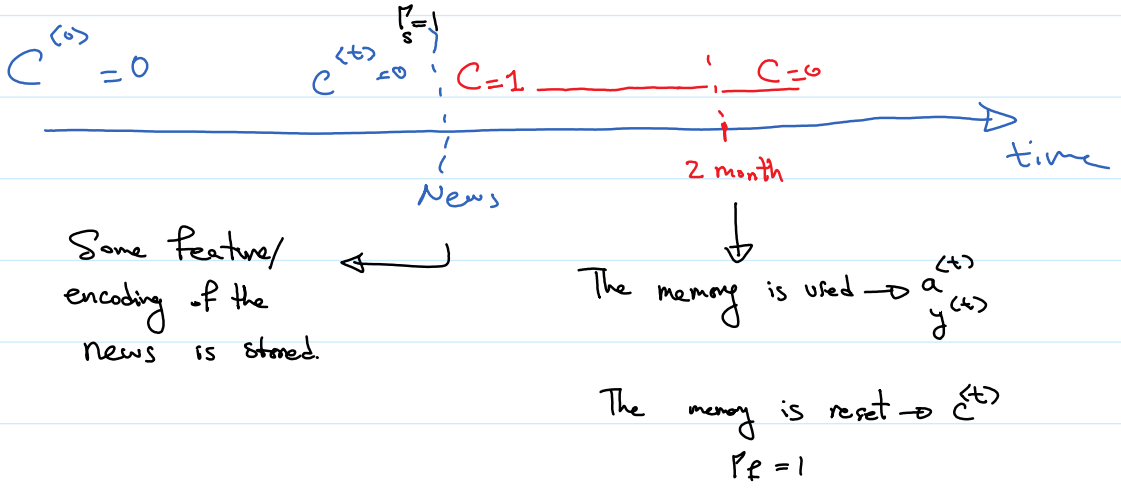
$a^{\langle t \rangle}$ only provides a short memory.

$c^{\langle t \rangle}$ is to extend this SM over a longer time.

Consider the price of some goods (some metal, etc.)
There's the typical input, daily price that helps predicting the price of tomorrow.

Then there's a news on tariff change in 2 month.

With a normal RNN, does not fall in the memory window and is neglected.
But with LSTM, here's what could happen (Naively):



This way the memory $c^{\langle t \rangle}$ can provide a tool for keeping significant info over a longer range of time.