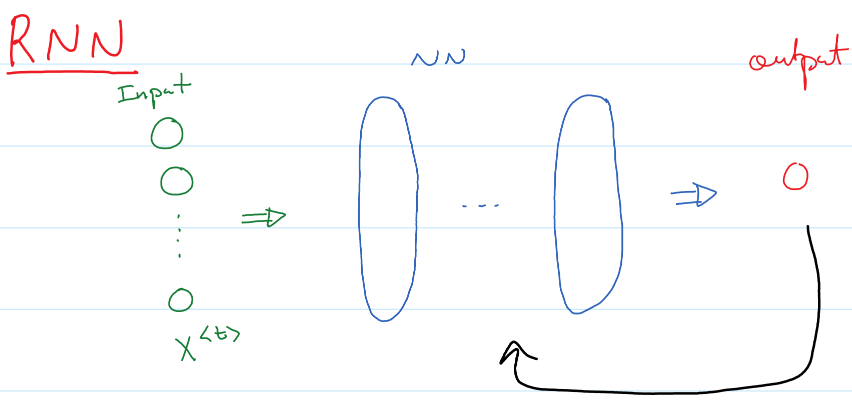
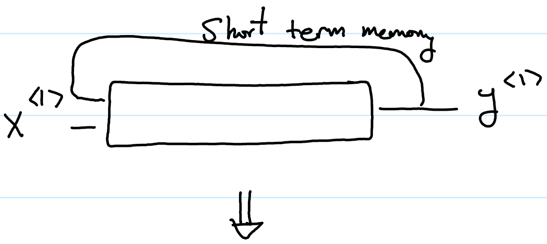
Recurrent Neural Networks

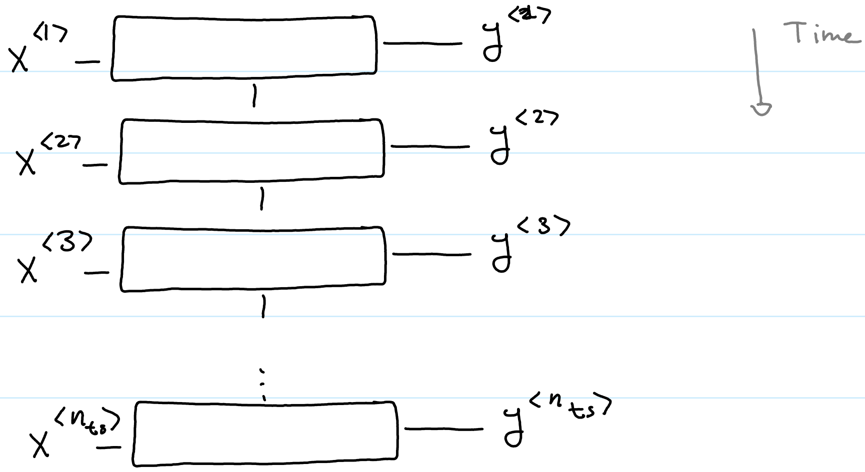
May 26, 2019 12:06 PM

RNN





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



Notation :

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

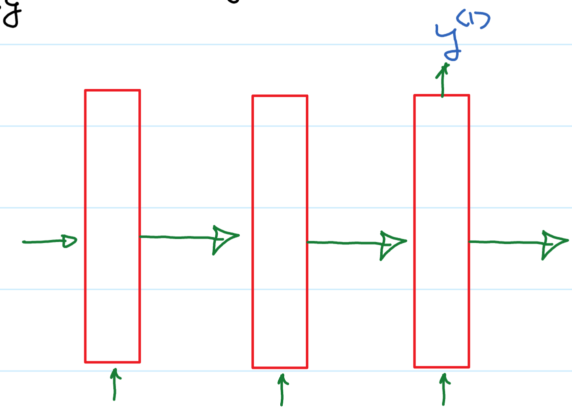
n_s: # Samples
n_{ts}: # time-series in each Sample
n_f: # features.

Y : A collection of labels : Shape $= (n_s, n_{tsy})$

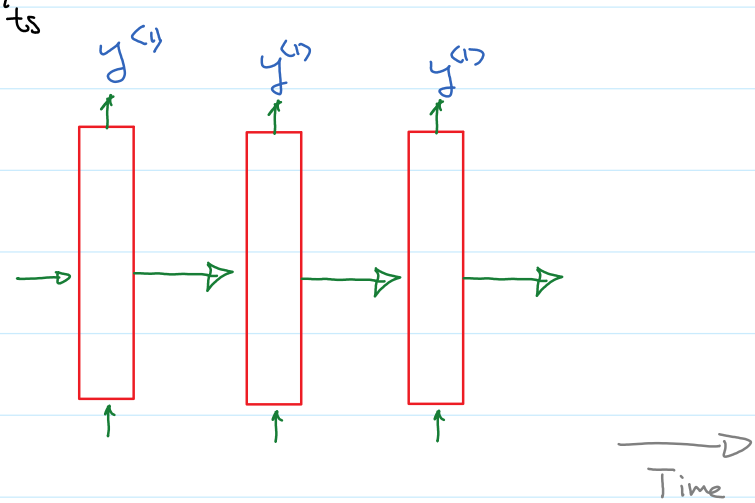
n_{tsy}: length of y outputs

n_{tsy} : 1 → Single output

length of y outputs



$n_{tsy} = n_{ts}$



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

Mathematical representation of RNN:

$$a^{\langle t\rangle} = g_a\left(W_a a^{\langle t-1\rangle} + U_a x^{\langle t\rangle} + b_a\right)$$

g_a: tanh/Relu

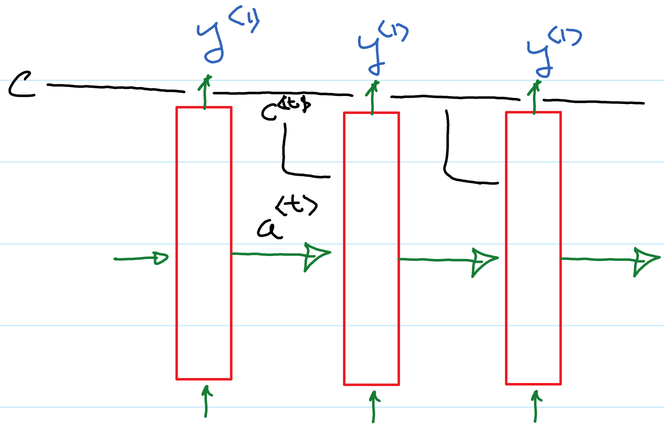
$$y^{\langle t\rangle} = g_y\left(W_y a^{\langle t\rangle} + b_y\right)$$

g_y: Sigmoid ...

GRU & LSTM

Issue with RNN: Vanishing / Exploding Gradient

Solution: Keep a memory



Memory:
- Store some info $\longrightarrow \Gamma_s$
- Forget " " $\longrightarrow \Gamma_f$
- Retrieve " " $\longrightarrow \Gamma_r \longrightarrow$ output

LSTM

$$\begin{cases} C^{\langle t \rangle} = tgh(W_m a^{\langle t-1 \rangle} + U_m X^{\langle t \rangle} + b_m) \\ \Gamma_s = Sig(W_s a^{\langle t-1 \rangle} + U_s X^{\langle t \rangle} + b_s) \\ \Gamma_f = Sig(W_f a^{\langle t-1 \rangle} + U_f X^{\langle t \rangle} + b_f) \\ C^{\langle t \rangle} = \Gamma_s C^{\langle t \rangle} + \Gamma_f C^{\langle t-1 \rangle} \\ \Gamma_r = Sig(W_r a^{\langle t-1 \rangle} + U_r X^{\langle t \rangle} + b_o) \\ a^{\langle t \rangle} = \Gamma_r c^{\langle t \rangle} \end{cases}$$

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.

$$C^{\langle t \rangle} = tgh(W_m \Gamma_I a^{\langle t-1 \rangle} + U_m X^{\langle t \rangle} + b_m)$$

Intuition

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

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

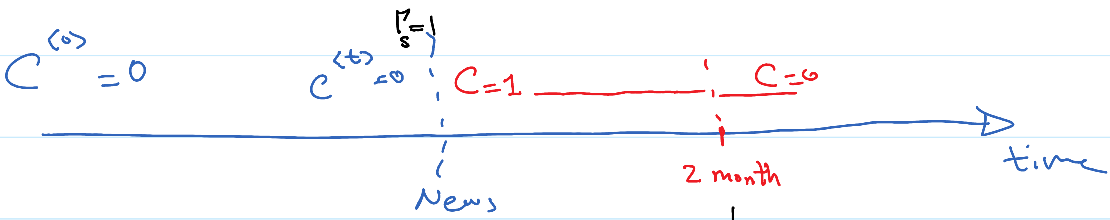
Consider the price of some goods (some metal, etc.)

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):



Some feature/encoding of the news is stored.

The memory is used $\rightarrow a^{\langle t \rangle}$, $y^{\langle t \rangle}$

The memory is reset $\rightarrow c^{\langle t \rangle}$

$\Gamma_f = 1$

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