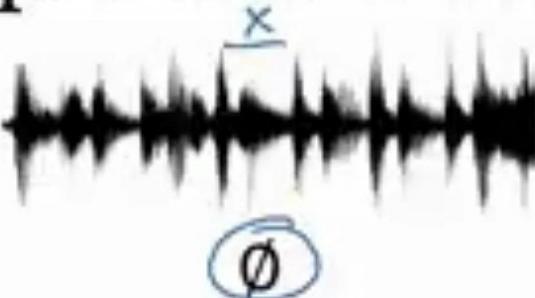


# Sequence Models – RNN/LSTM

Aniket Chhabra

# Examples of sequence data

Speech recognition



"The quick brown fox jumped  
over the lazy dog."

Music generation



Sentiment classification

"There is nothing to like  
in this movie."



DNA sequence analysis

AGCCCCTGTGAGGAAC TAG



AGCCCCTGTGAGGAAC **TAG**

Machine translation

Voulez-vous chanter avec  
moi?



Do you want to sing with  
me?

Video activity recognition



Running

Name entity recognition

Yesterday, Harry Potter  
met Hermione Granger.



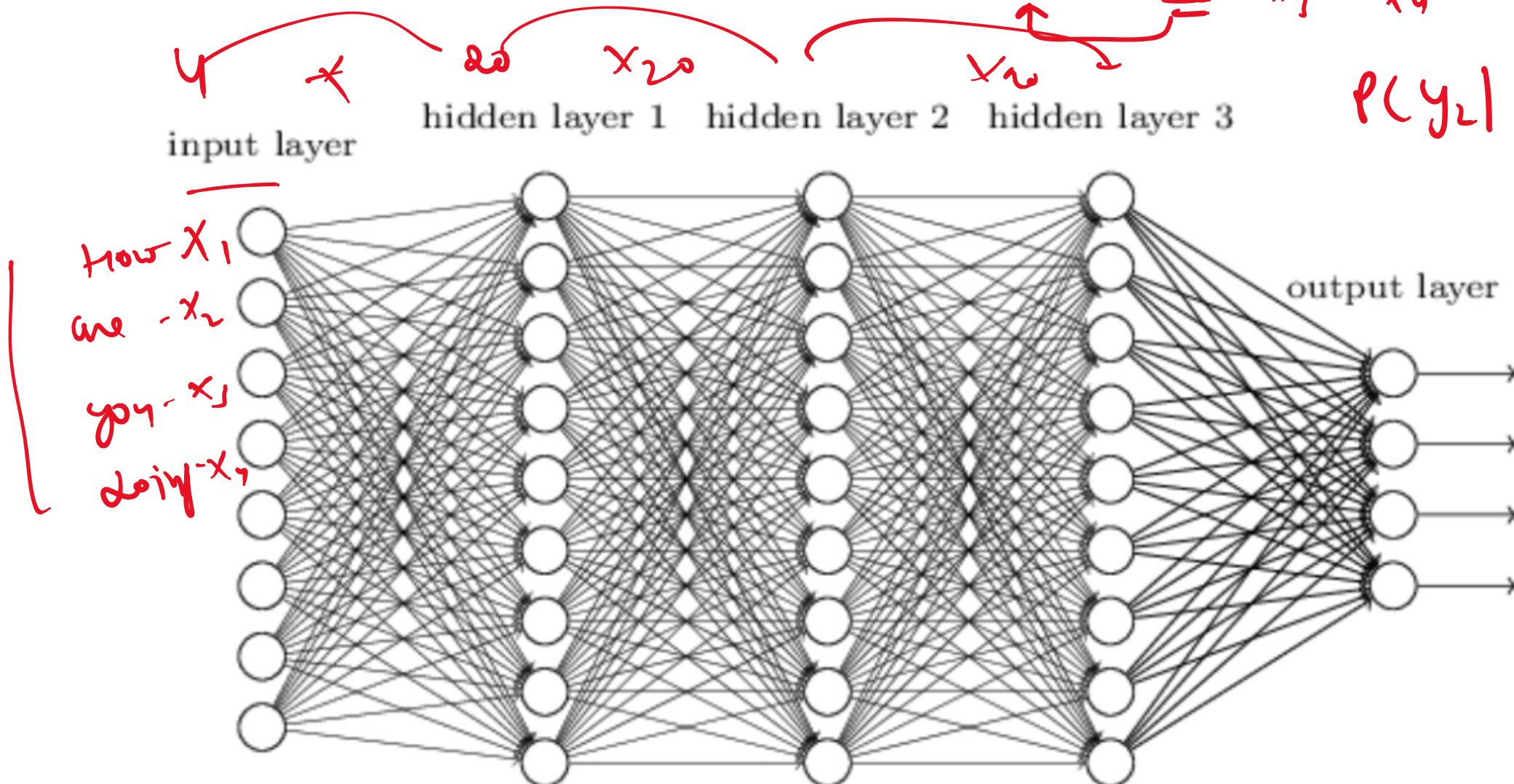
Yesterday, **Harry** Potter  
met **Hermione Granger**.

Andrew Ng

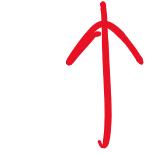
# Neural Networks limitation

How are you doing - tfsof  
 $x_1 \quad x_2$   
 $x_3 \quad x_4$

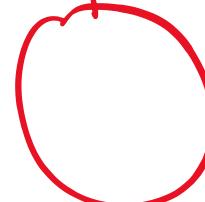
$$p(y_2 | x_1, x_2)$$



$x$   
 $y_1 - \text{ans}$



$w, z = h_i$



$x_1 - \text{how}$

$\frac{y_{04}}{P(y_2 | K_1, x_2)} =$   
↑  
---  
|  
|

$x_2 - \text{ans}$

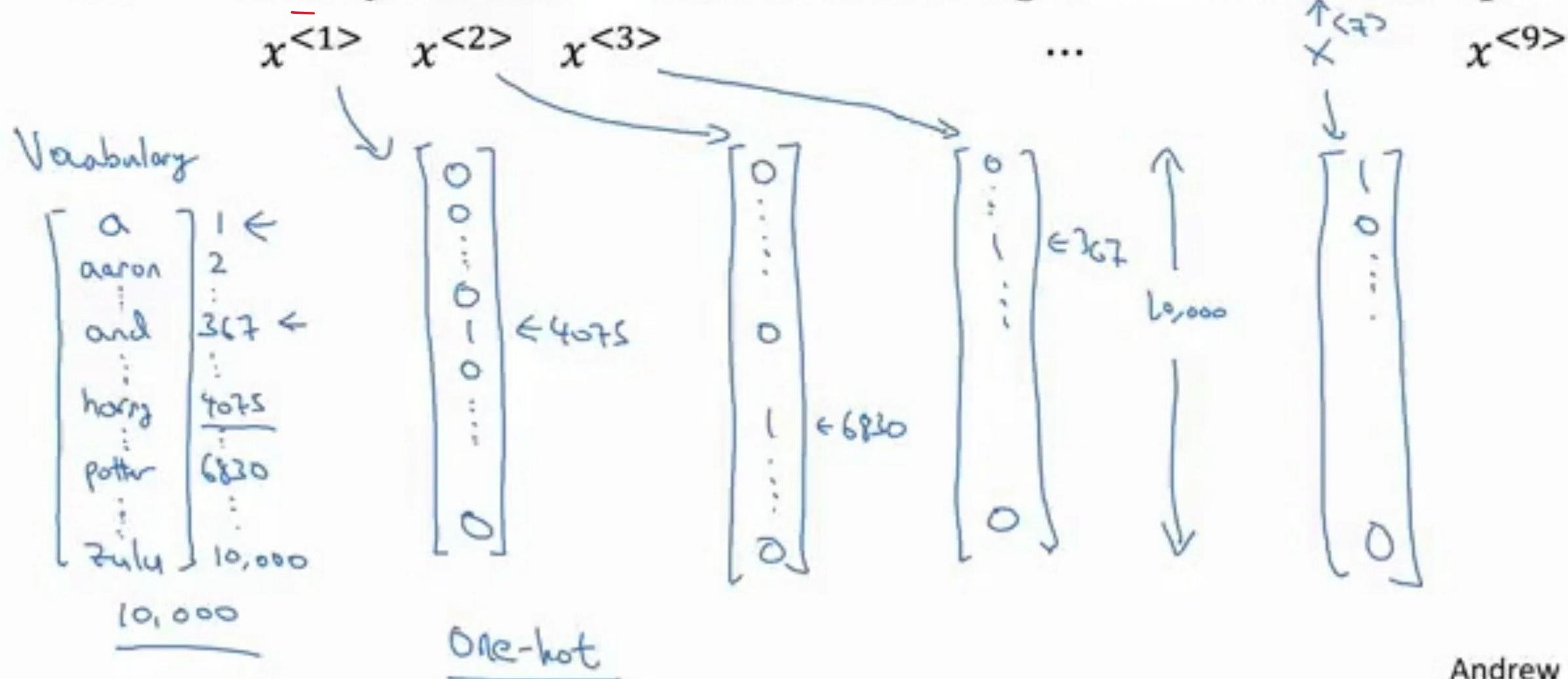
End=??  
[ ]

$x_3 - y_{04} - x_n - \text{ans}$

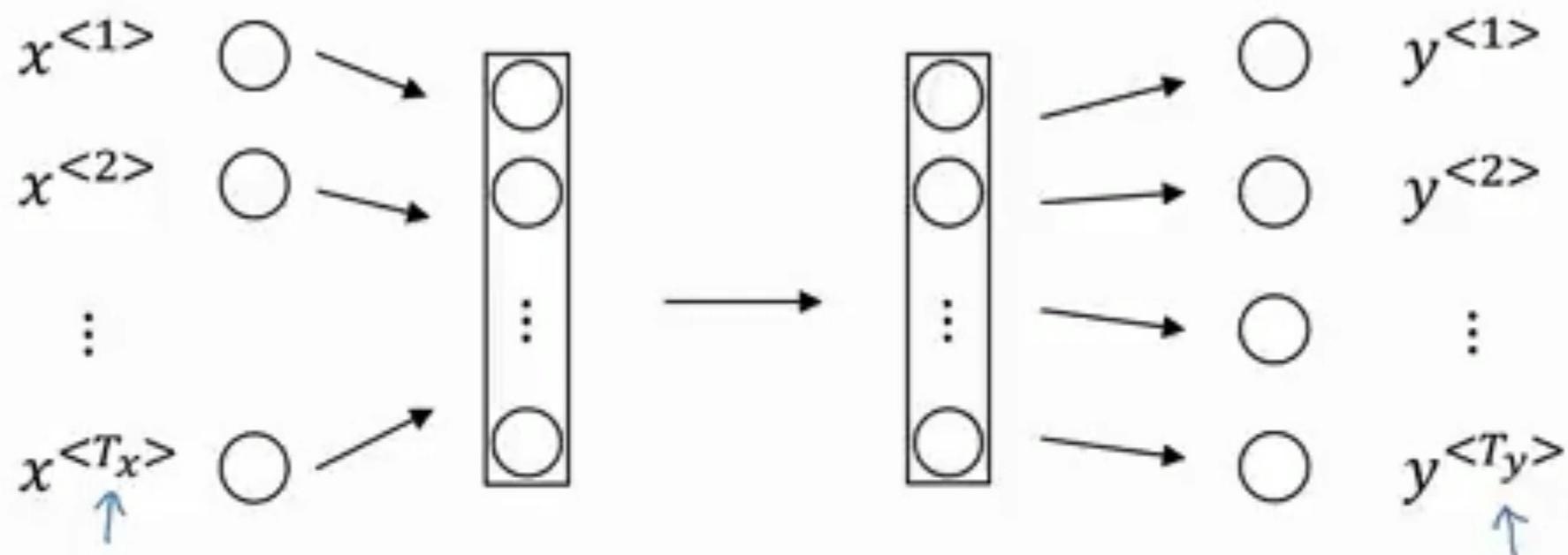
# Representing words

$\times^{<\leftrightarrow>}$

$x:$  Harry Potter and Hermione Granger invented a new spell.



# Why not a standard network?

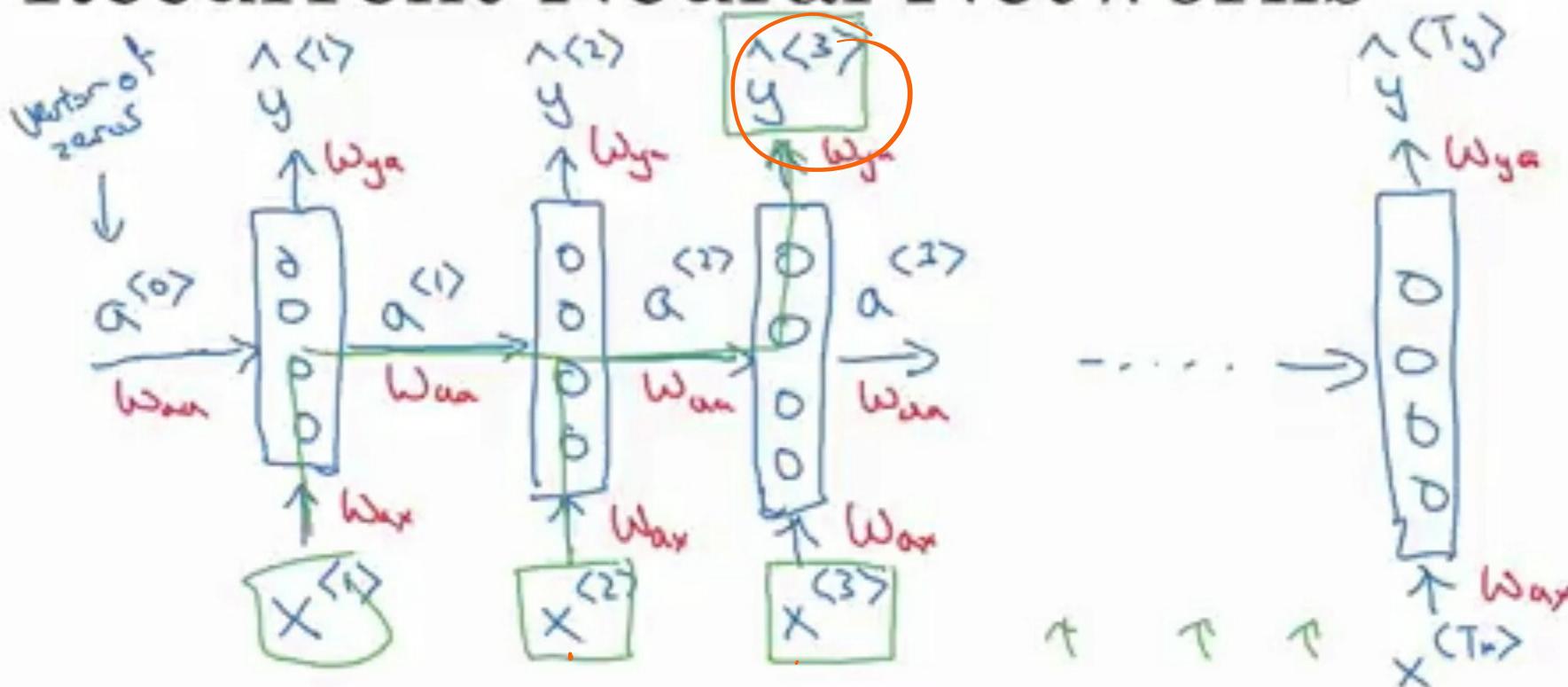


Problems:

- Inputs, outputs can be different lengths in different examples.
- Doesn't share features learned across different positions of text.

# Recurrent Neural Networks

$$T_x = T_y$$



Left to Right

Bidirectional

RNN

(BRNN)

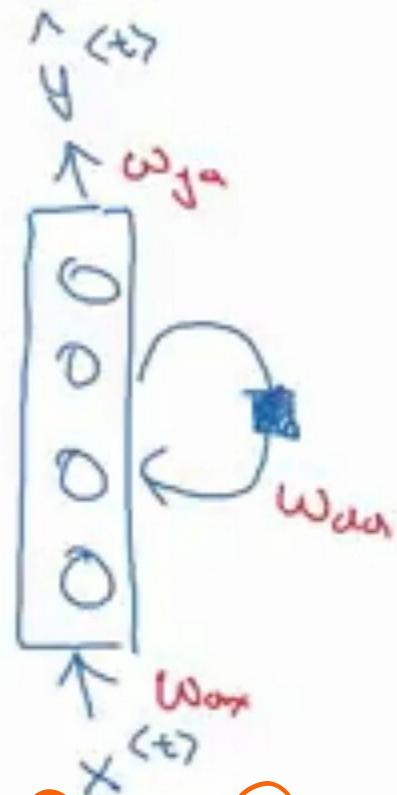
uni

[ He said, "Teddy Roosevelt was a great President."  
He said, "Teddy bears are on sale!" ]

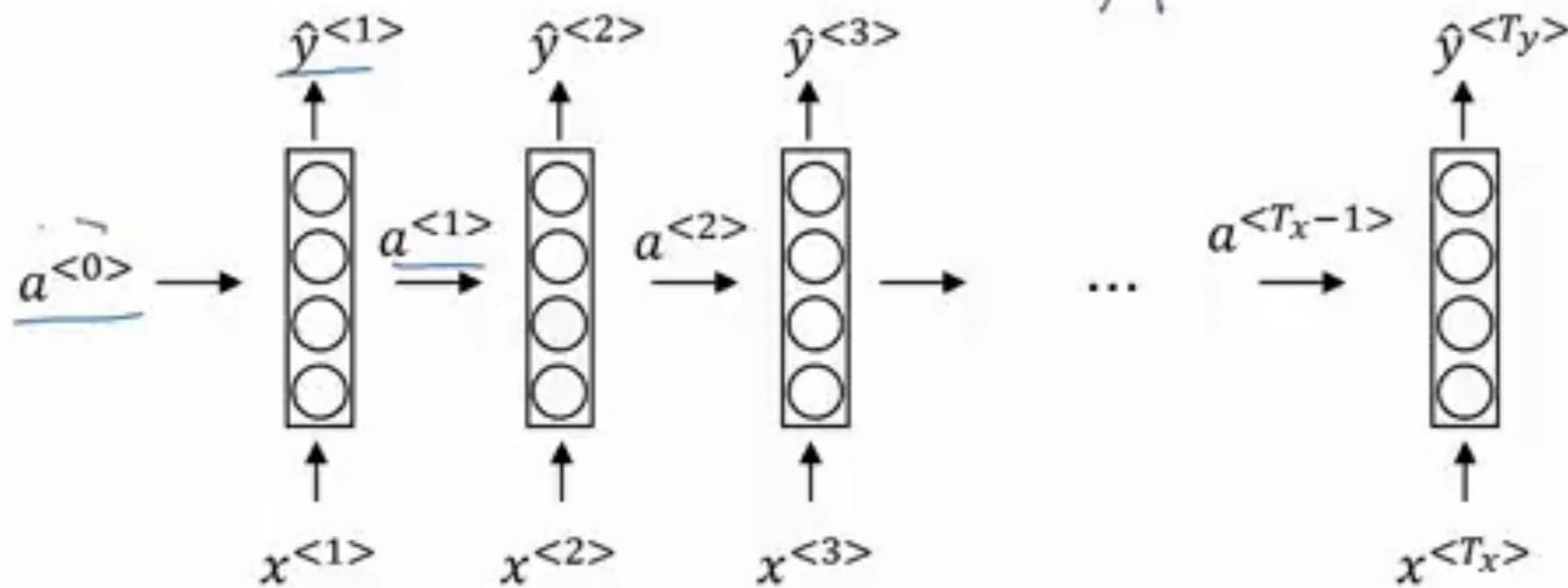
Bi-directional  
Right to left

Bi-directional

Andrew Ng



# Forward Propagation



$$a^{<0>} = \vec{0}.$$

$$\underline{a}^{<1>} = g_1(W_{aa} \underline{a}^{<0>} + \underline{W_{ax}} x^{<1>} + b_a) \leftarrow \tanh \text{ or ReLU}$$

$$\hat{y}^{<1>} = g_2(W_{ya} \underline{a}^{<1>} + b_y) \leftarrow \text{Sigmoid}$$

$$\underline{a}^{<t>} = g(W_{aa} \underline{a}^{<t-1>} + W_{ax} x^{<t>} + b_a)$$

$$\hat{y}^{<t>} = g(W_{ya} \underline{a}^{<t>} + b_y)$$

# Simplified RNN notation

$$a^{(t)} = g(W_{aa}a^{(t-1)} + W_{ax}x^{(t)} + b_a)$$

Dimensions:  
 $W_{aa}$ :  $(100, 100)$   
 $x^{(t)}$ :  $(100, 10,000)$   
 $a^{(t)}$ :  $(100, 100)$

$$\hat{y}^{(t)} = g(W_{ya}a^{(t)} + b_y)$$

$$\hat{y}^{(t)} = g(W_y a^{(t)} + b_y)$$

$\uparrow$                $\uparrow$                $\uparrow$

$$a^{(t)} = g(W_a [a^{(t-1)}, x^{(t)}] + b_a)$$

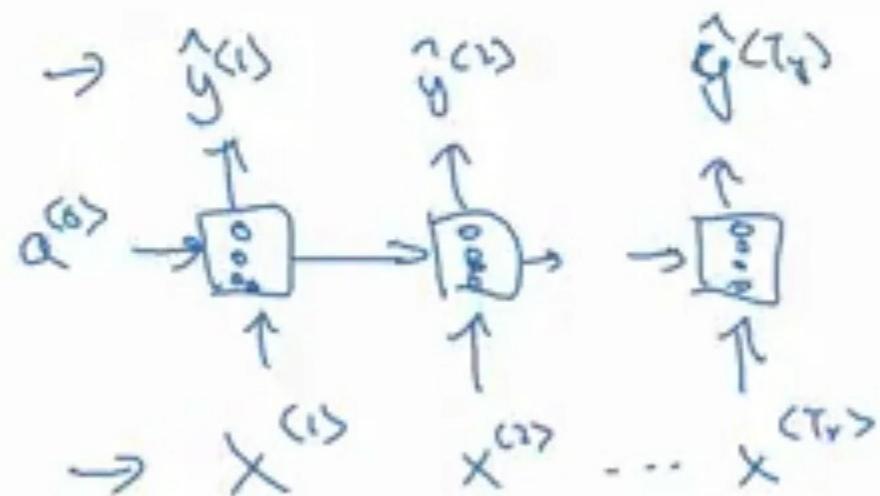
$$\begin{bmatrix} 100 \\ \hline W_{aa} & W_{ax} \\ \hline 100 & 10,000 \end{bmatrix} = W_a \quad (100, 10,000)$$

$$[a^{(t-1)}, x^{(t)}] = \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix} \quad \begin{array}{c} \uparrow 100 \\ \uparrow 10,000 \\ \hline 10,100 \end{array}$$

$$[W_{aa}; W_{ax}] \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix} = W_{aa}a^{(t-1)} + W_{ax}x^{(t)}$$

# Examples of RNN architectures

$$T_x = T_y$$



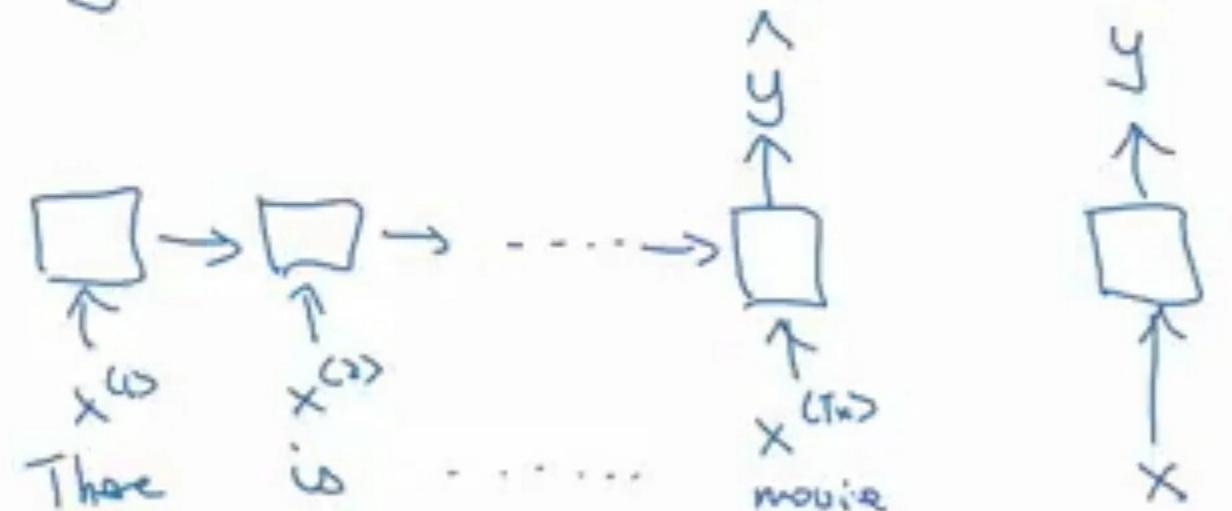
Many-to-many

[Speech Recognition]

Sentiment classification

$x = \text{text}$

$y = 0/1 \quad 1\cdots 5$

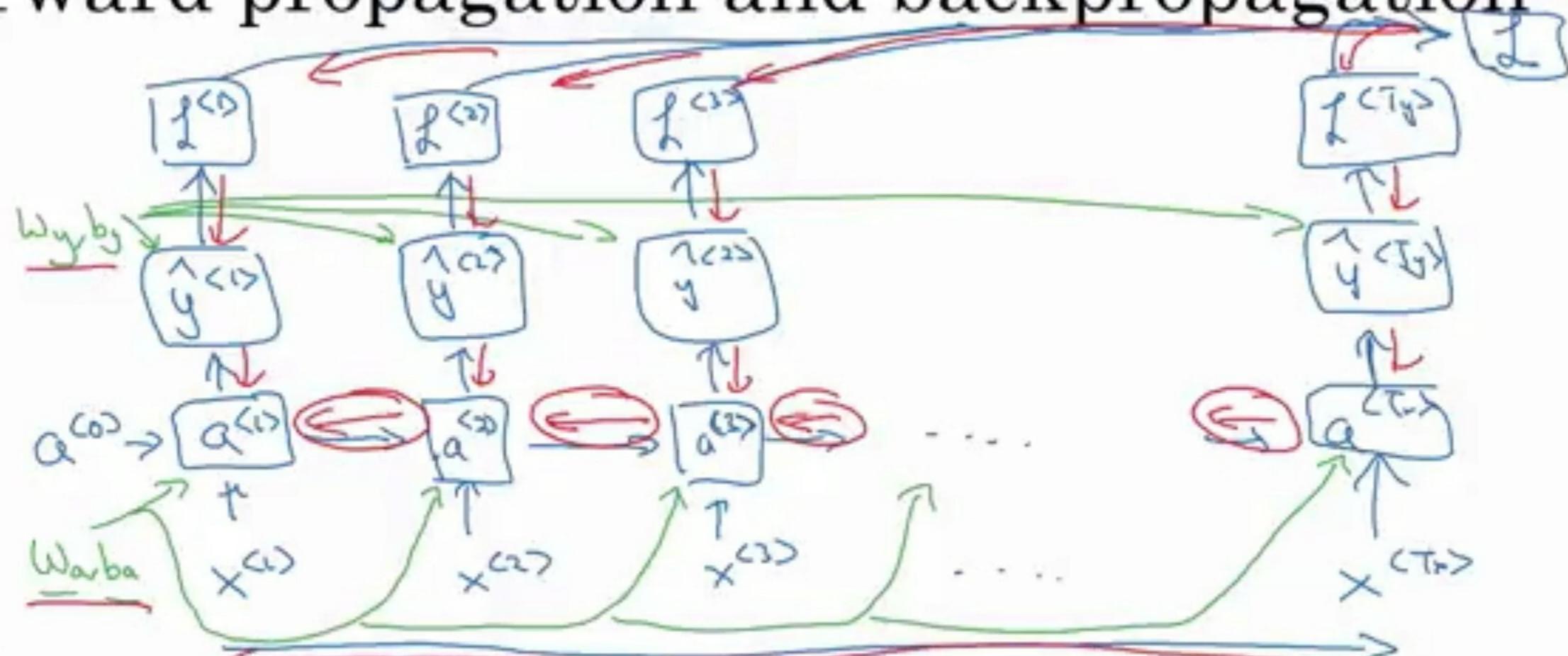


Many-to-one

One-to-one

Andrew Ng

# Forward propagation and backpropagation

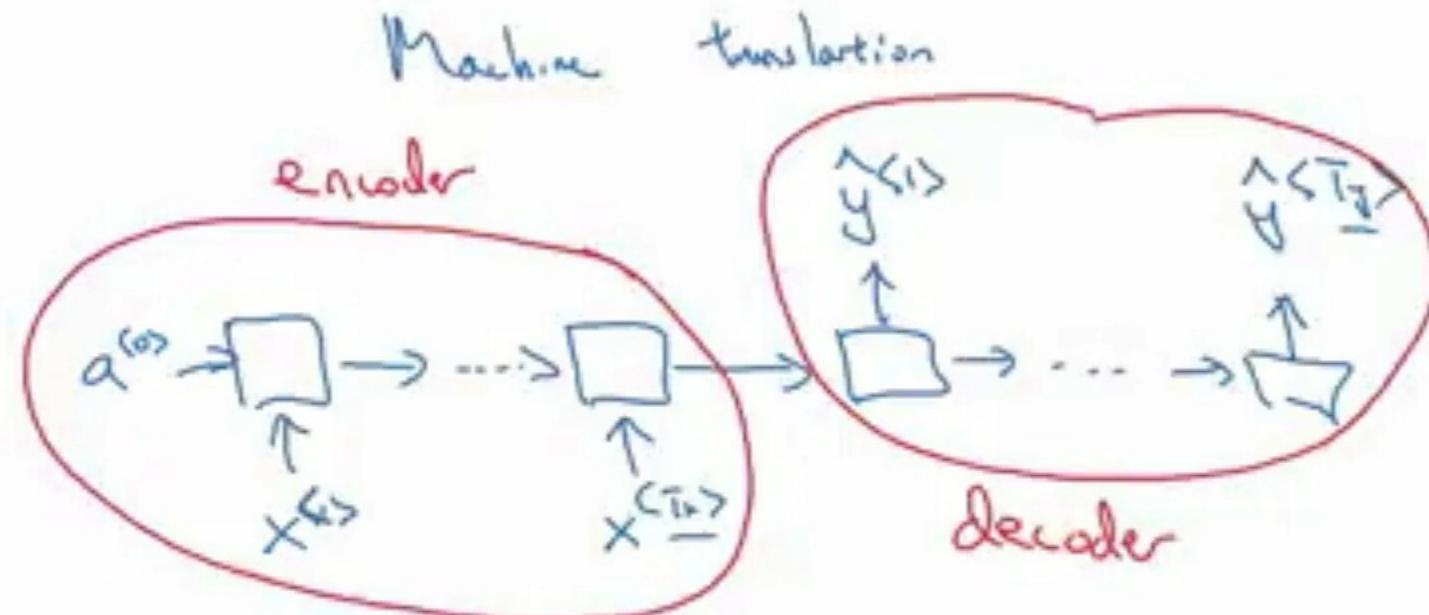
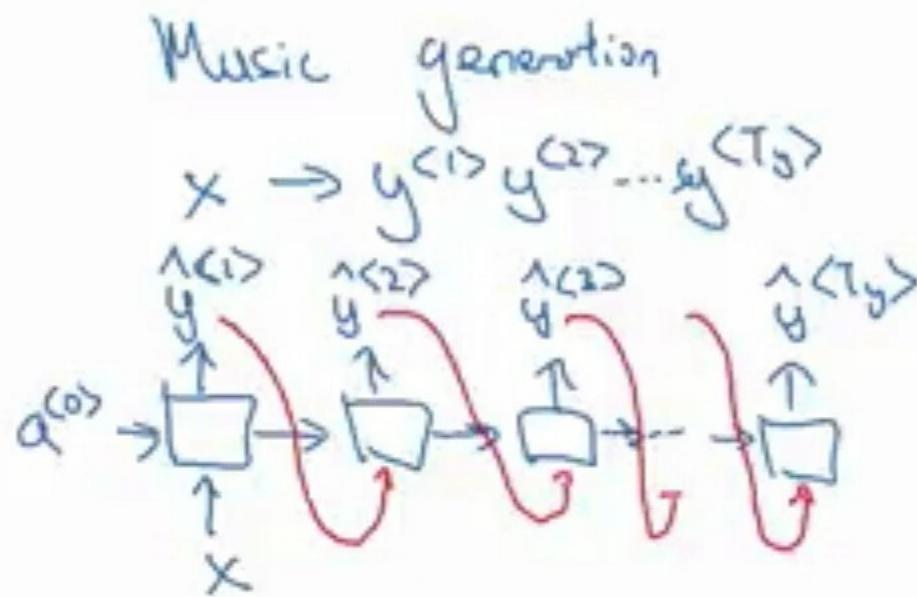


$$\mathcal{L}^{(t)}(\hat{y}^{(t)}, y^{(t)}) = -y^{(t)} \log \hat{y}^{(t)} - (1-y^{(t)}) \log (1-\hat{y}^{(t)})$$

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^T \mathcal{L}^{(t)}(\hat{y}^{(t)}, y^{(t)})$$

Backpropagation through time

# Examples of RNN architectures



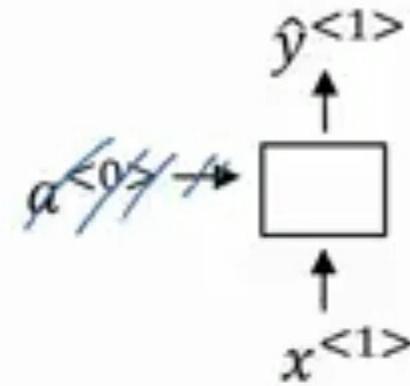
One-to-many

$$x = \phi$$

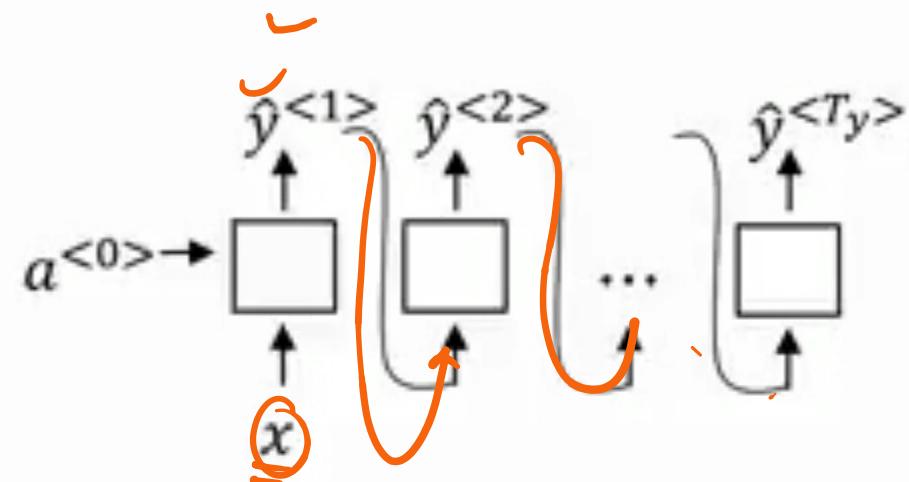
→ Attention

Many-to-many

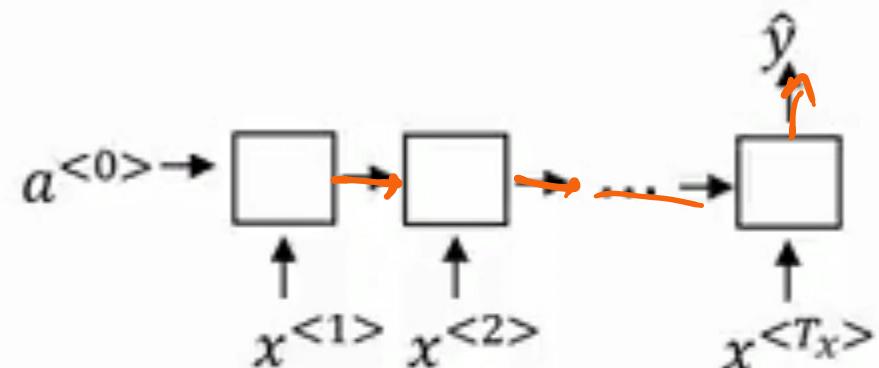
# Summary of RNN types



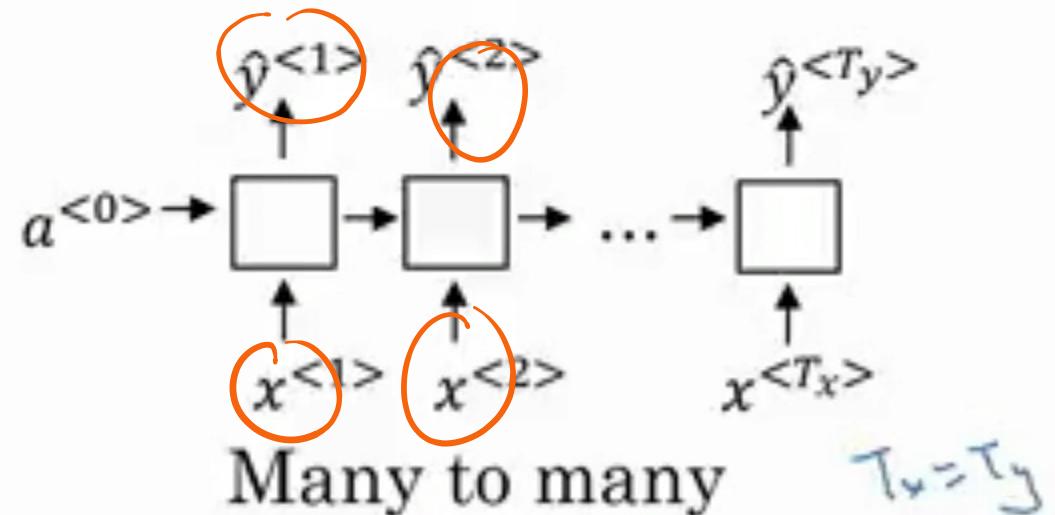
One to one



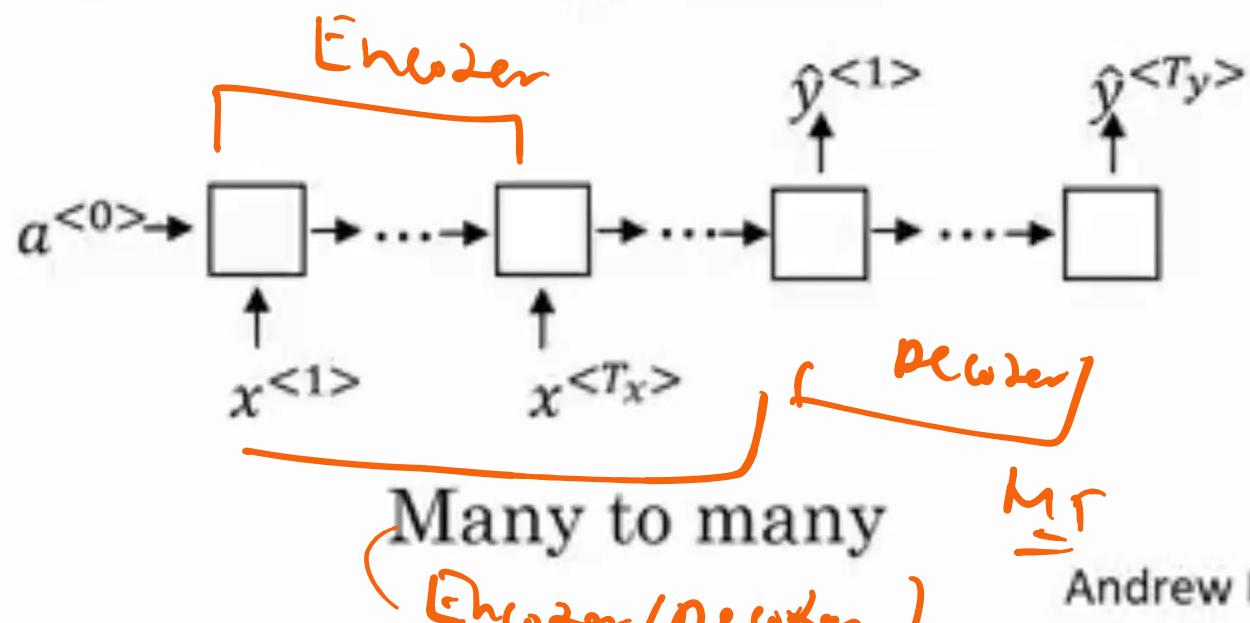
One to many



Many to one



Many to many

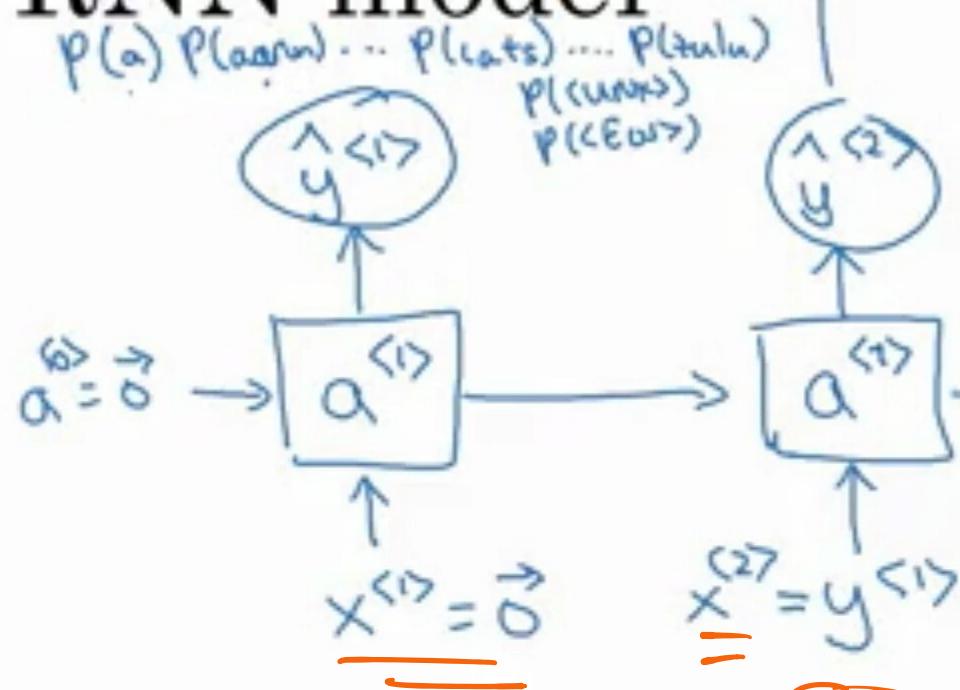


Many to many

Encoder/Decoder

Andrew Ng

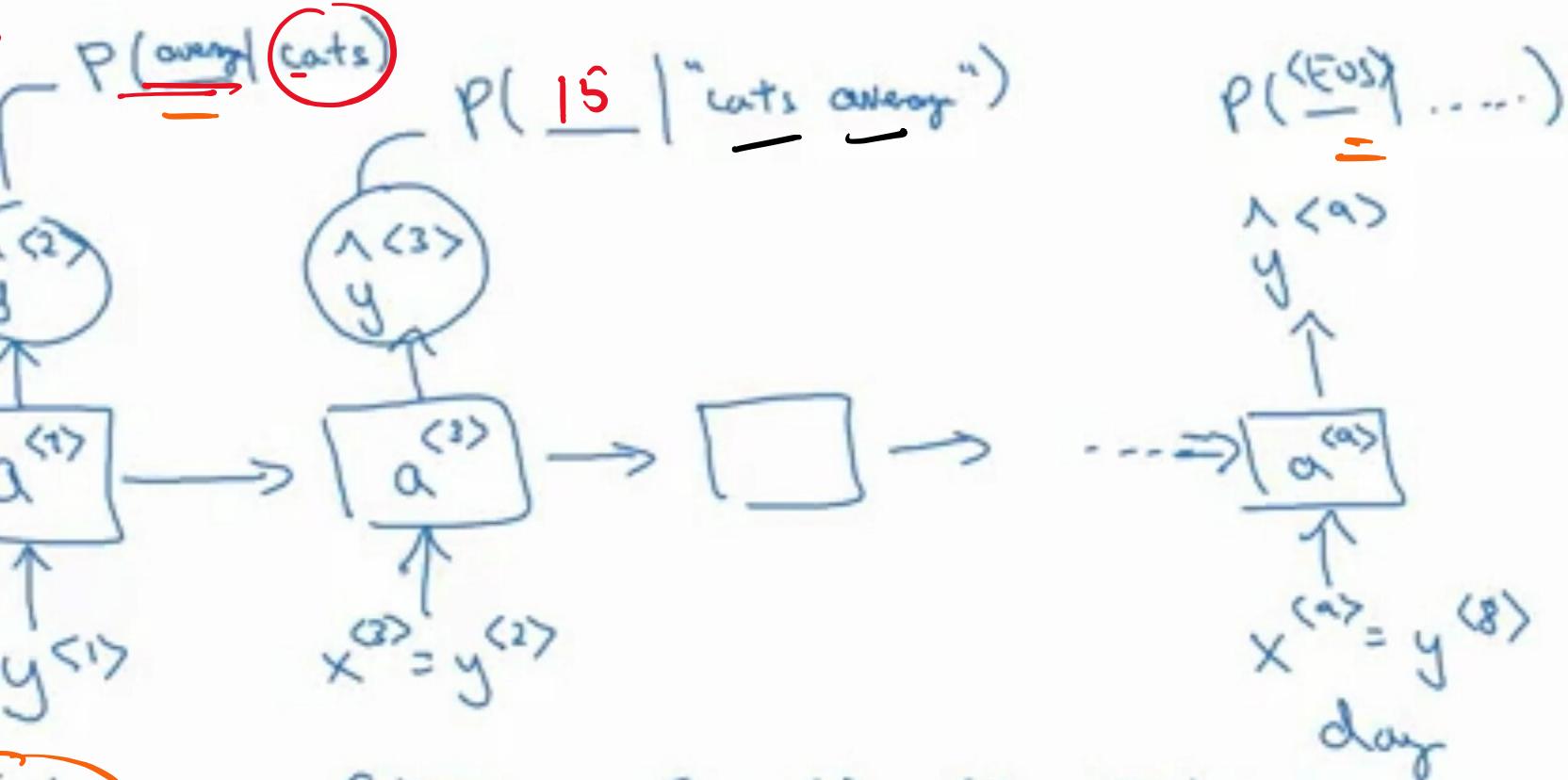
# RNN model



→ Cats average 15 hours of sleep a day. <EOS>

$$\mathcal{L}(\hat{y}^{(t)}, y^{(t)}) = - \sum_i y_i^{(t)} \log \hat{y}_i^{(t)}$$

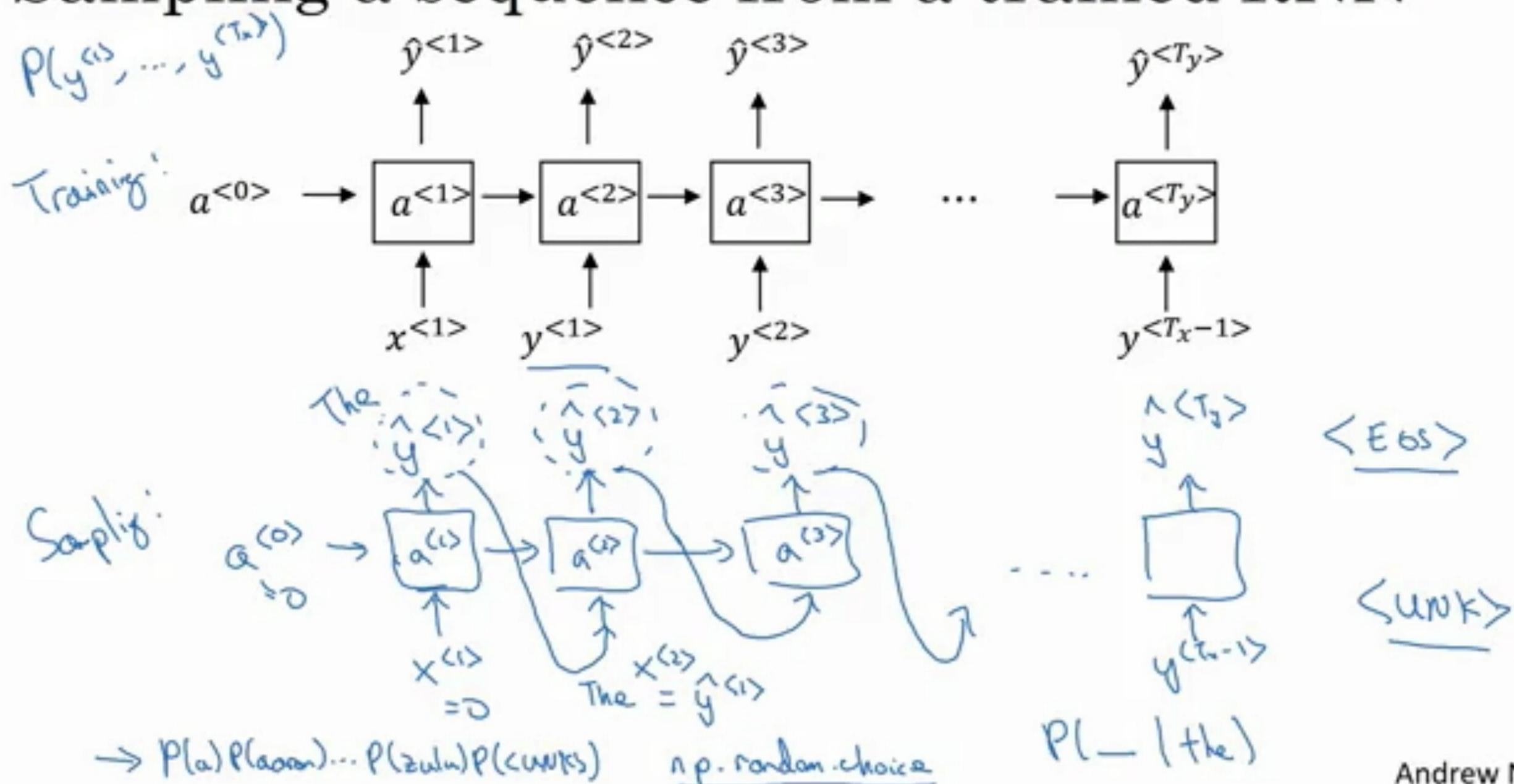
$$\mathcal{L} = \sum_t \mathcal{L}^{(t)}(\hat{y}^{(t)}, y^{(t)})$$



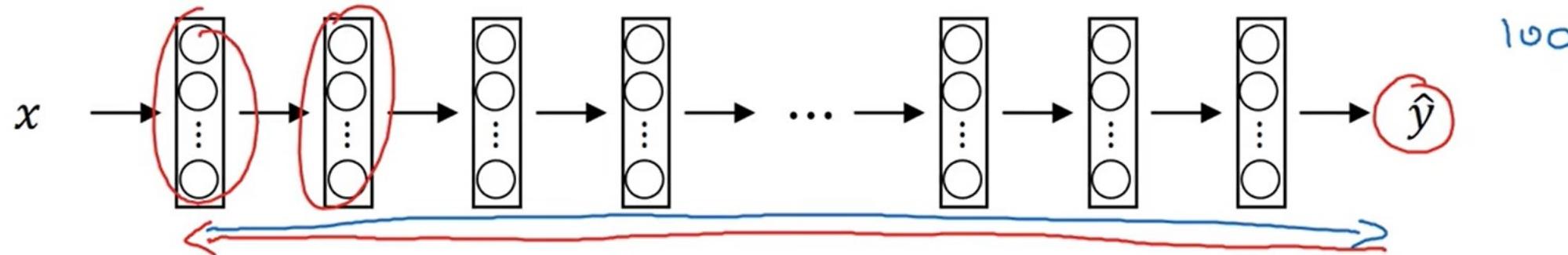
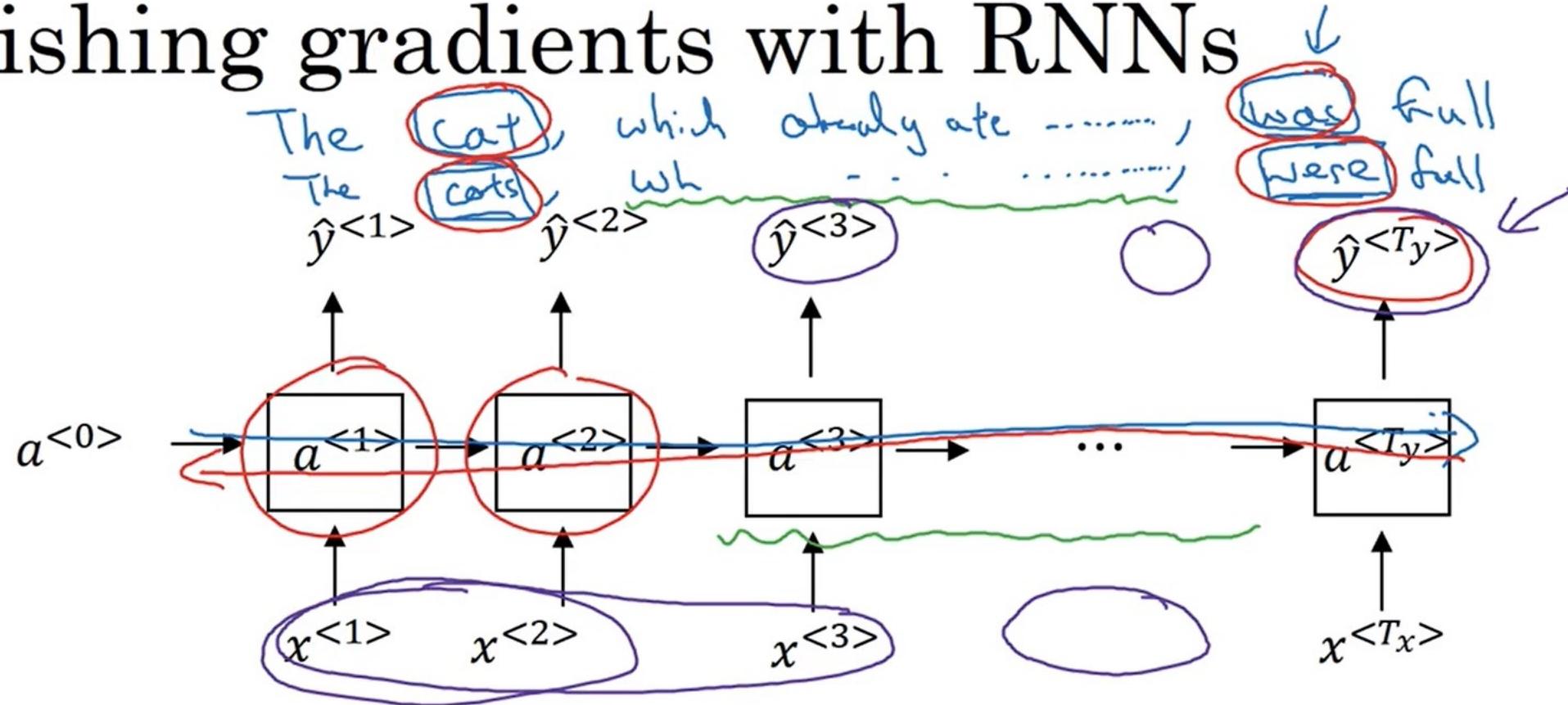
$$P(y^{(1)}, y^{(2)}, y^{(3)}) \leftarrow$$

$$= \frac{p(y^{(1)})}{p(y^{(2)} | y^{(1)})} \frac{p(y^{(2)} | y^{(1)}, y^{(2)})}{p(y^{(3)} | y^{(2)}, y^{(3)})}$$

# Sampling a sequence from a trained RNN



# Vanishing gradients with RNNs



Exploding gradients.

Nan

Gradient clipping - exponentially large

Andrew Ng

# Why LSTM

drawback

Drawback of RNN ? ↗

RNN

(1) The sky is blue.

long-term sequence

"RNN"

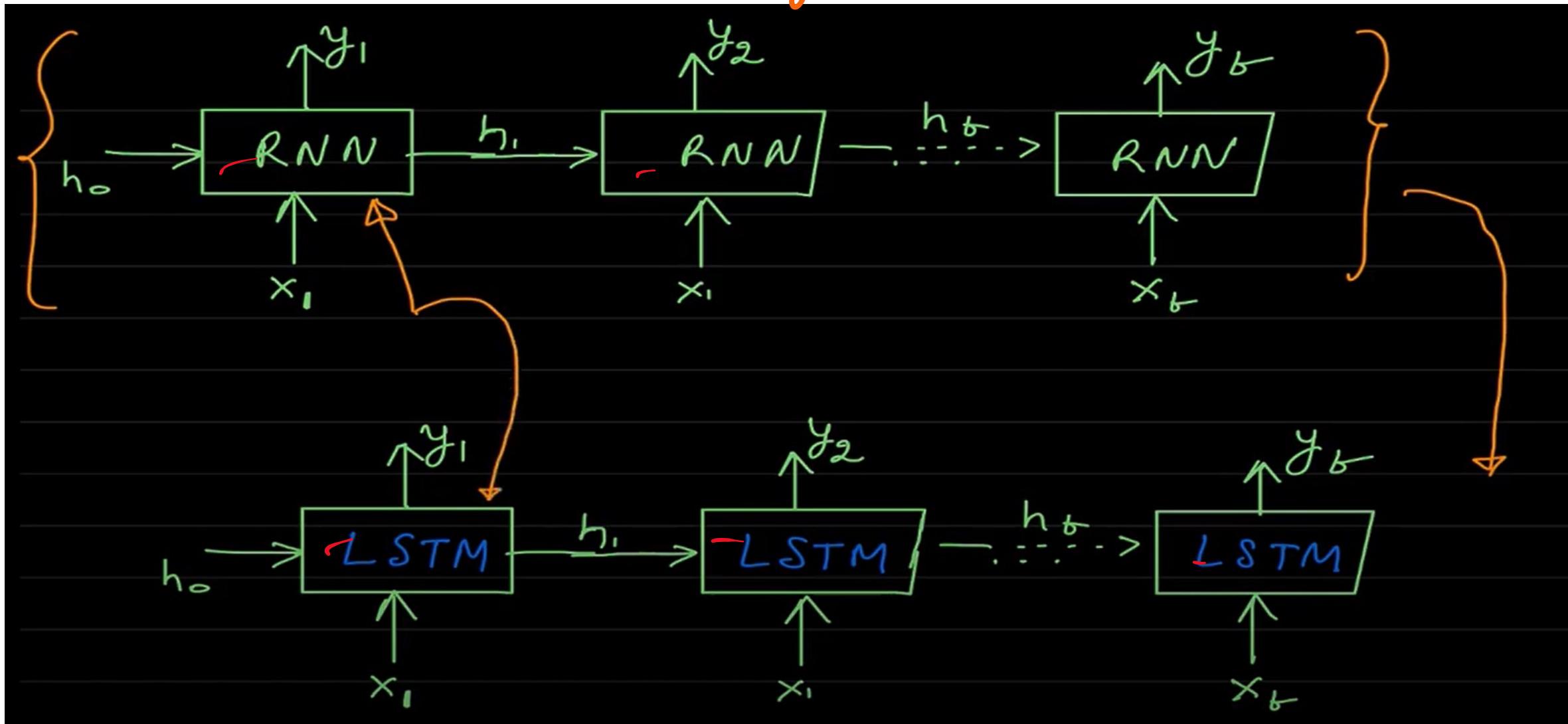
(2) Ganesh lived in {India} for 13 years. He loves watching movies. He is a fan of drama. He is fluent in {Hindi}?

```
graph LR; H1[H1] --> H2[H2]; H2 --> H1; H2 --> H3[H3]; H3 --> H2; H3 --> H4[H4]; H4 --> H3; H4 --> H5[H5]; H5 --> H4; H5 --> H6[H6]; H6 --> H5; H6 --> H7[H7]; H7 --> H6; H7 --> H8[H8]; H8 --> H7; H8 --> H9[H9]; H9 --> H8; H9 --> H10[H10]; H10 --> H9;
```

# LSTM v/s RNN Architecture

Long short term memory

Short Memory

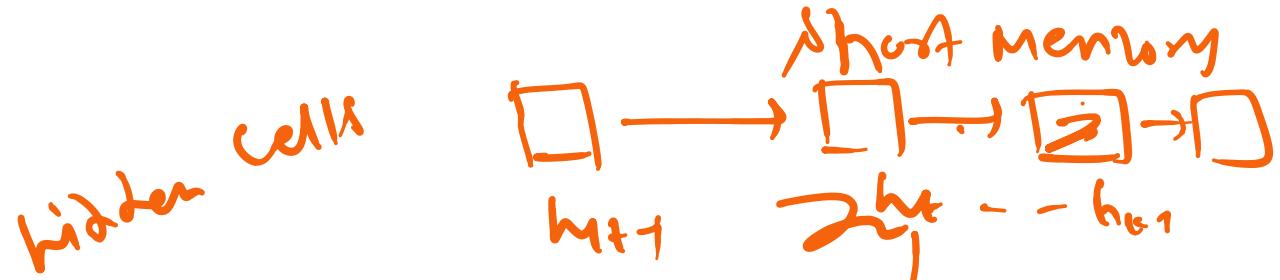


# Questions we try to answer in LSTM

- What makes LSTM cell special ?
- How do LSTM cell achieve long term dependency ?
- How does it know what information to keep &
- what information to discard from the memory.?

\* All these question can be answered using Gates

# RNN Cell v/s LSTM Cell

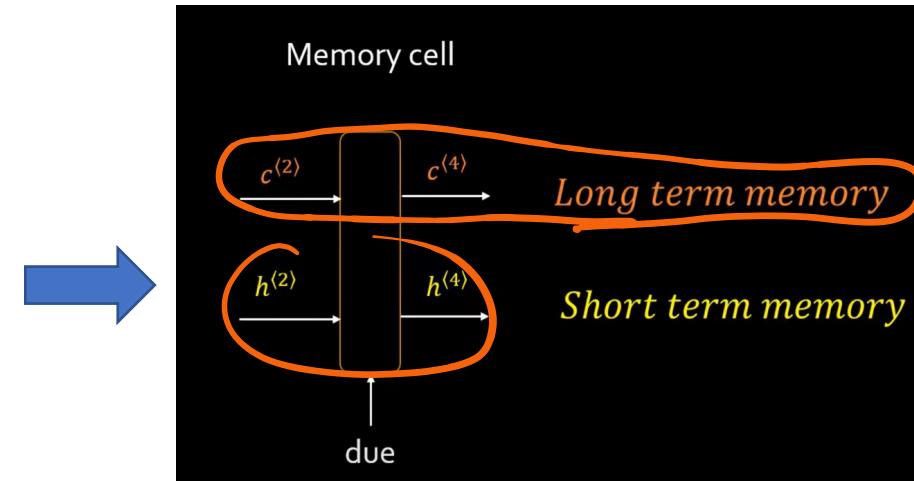


LSTM: Here  $\xrightarrow{\text{hidden state}}$  is broken into two states

(1) Cell state: Called internal memory where all info: will be stored

(2) Hidden state: Used for computing the output.

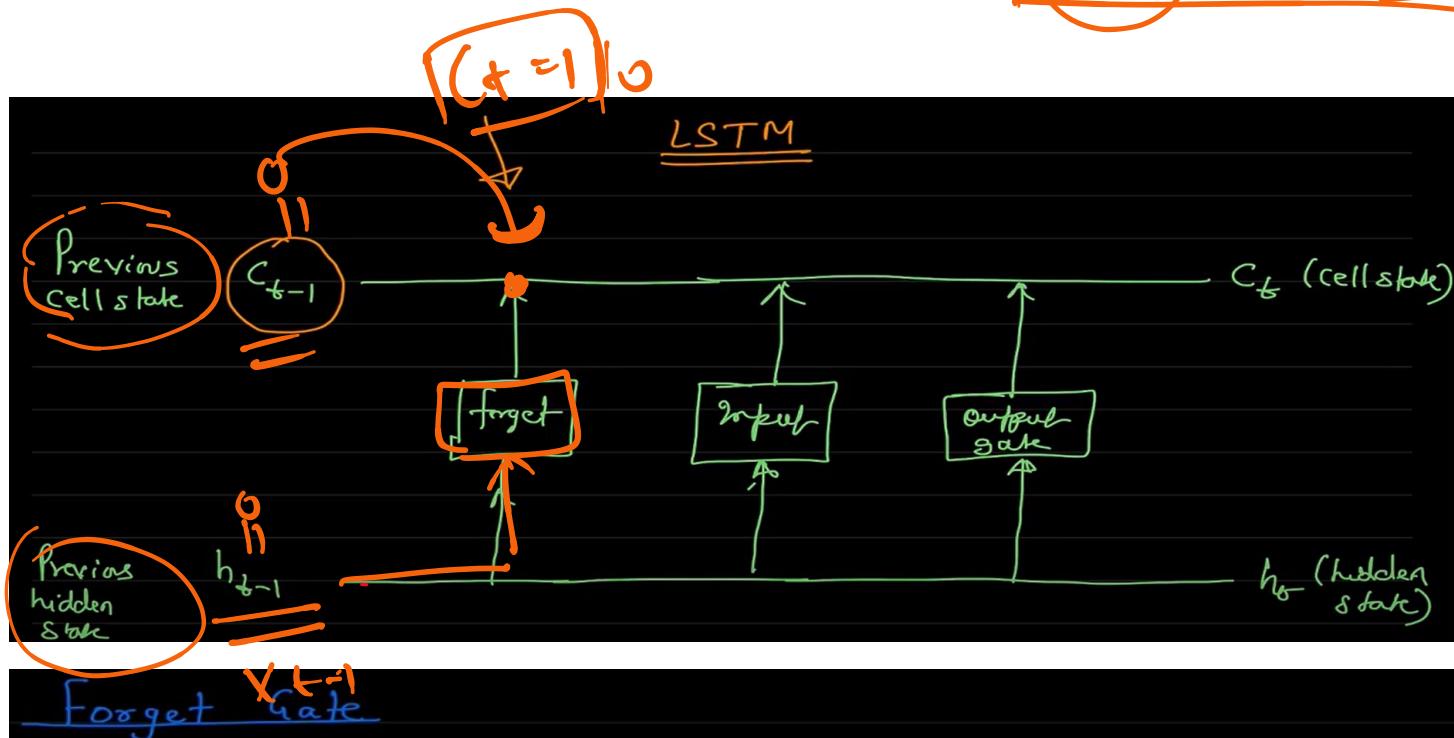
Note: Cell state & hidden state shared across every time



# LSTM - Forget <sup>zz</sup> gate

$$h_t = \underline{w_t x_t + b_t} +$$

Widder State



It is responsible for deciding what information should be removed from the cell state.

Consider the following sequence

Mark is a good singer. He lives in California.

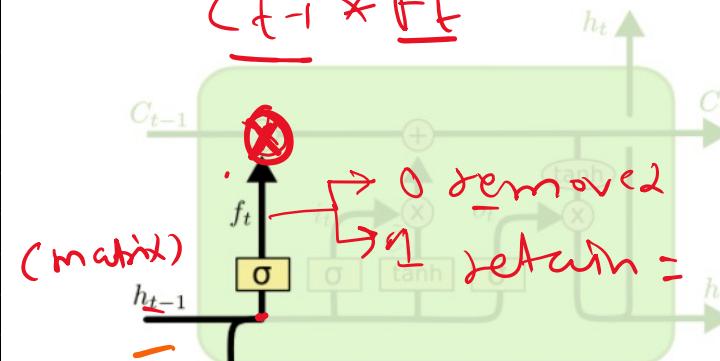
Jacob is also a good singer.

Mark is forgotten by forget gate

$$f_t = \sigma\left(\frac{W_f \cdot h_{t-1}}{N} + W_f \cdot x_t + b_f\right)$$

$\xrightarrow{\text{if } f_t > 0 \text{ - keep}}$

$$c_{t-1} * \underbrace{f_t}_{\begin{matrix} (1,0) \\ h_t \uparrow \end{matrix}} = 0$$



0 - remove  
1 - re-task

$$f_t = \sigma(w_f \cdot h_{t-1} + w_x \cdot x_t + b)$$

# LSTM – Input gate and Candidate gate

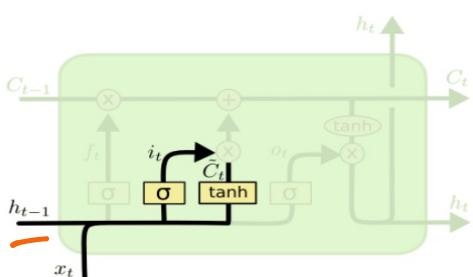
(-1, +1)

{ Input Gate: }

Input gate is responsible for deciding what information should be stored in the cell state.

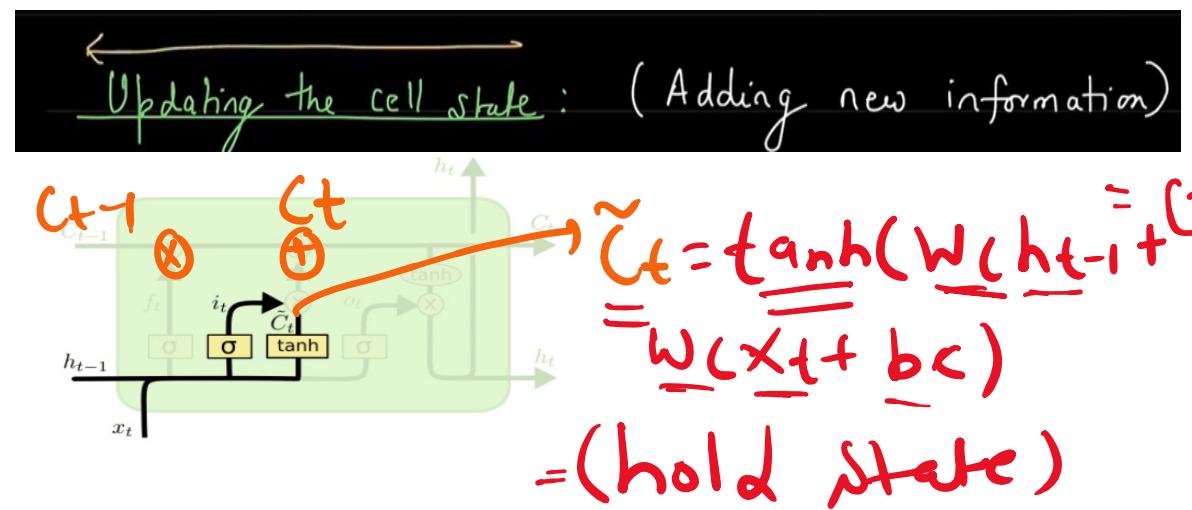
{ Mark is a good singer. He lives in California. }  
Jacob is also a good singer.

→ Add this into  $C_{t-1}$



0 - info will not be stored and vice versa

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i)$$



This hold state is controlled by input stage

if  $i_t * \tilde{C}_t = 1$  - add new info to cell

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

# LSTM – Output gate

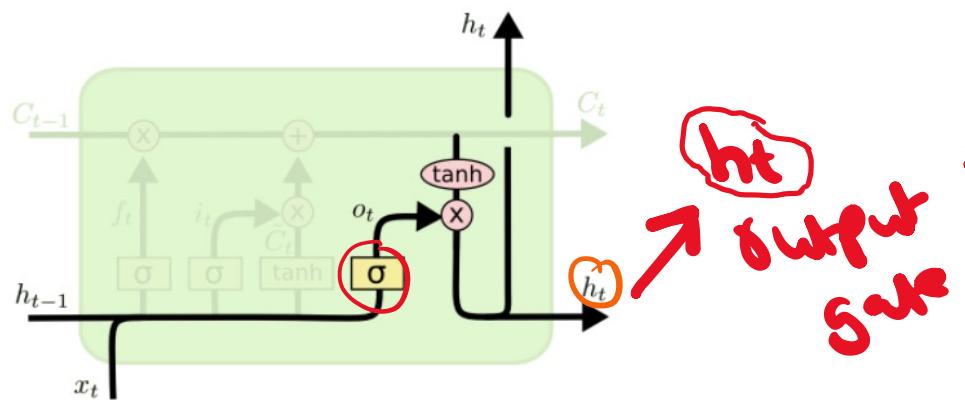


{ Output Gate : }

A lot of information is in cell state (memory). The output gate is responsible for deciding what information should be taken from the cell to give as an output.

Consider the following sentence:

Jacob debut album was a huge success. Congratz



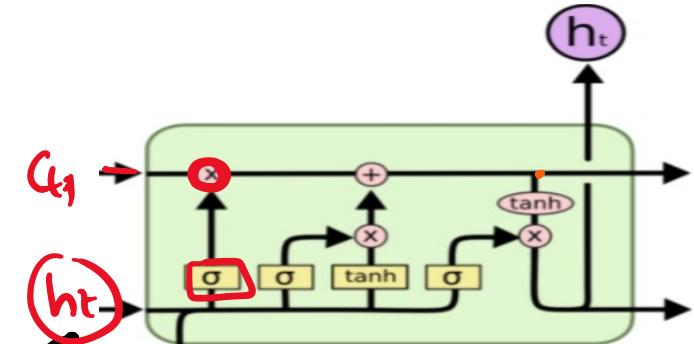
$$h_t = o_t \times \tanh(C_t)$$

$$y_t = \text{Softmax}(h_t)$$

final  
o/p

$$o_t = \sigma(W_o h_{t-1} + W_o x_t + b_o)$$

$\hookrightarrow$  0 information will not be passed to o/p



# LSTM – Forward Propagation

## Forward Propagation

$$\left\{ \begin{array}{l} \text{Input gate: } i_t = \sigma(U_i \tilde{x}_t + W_i \tilde{h}_{t-1} + b_i) \\ \text{Forget gate: } f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \\ \text{Output gate: } o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) \\ \text{Candidate state: } g_t = \tanh(U_g x_t + W_g h_{t-1} + b_g) \\ \text{Cell state: } c_t = f_t c_{t-1} + i_t g_t \\ \text{Hidden state: } h_t = o_t \tanh(c_t) \\ \text{Output: } \tilde{y}_t = \text{softmax}(V h_t) \end{array} \right. \quad \checkmark$$

# LSTM – Backward Propagation

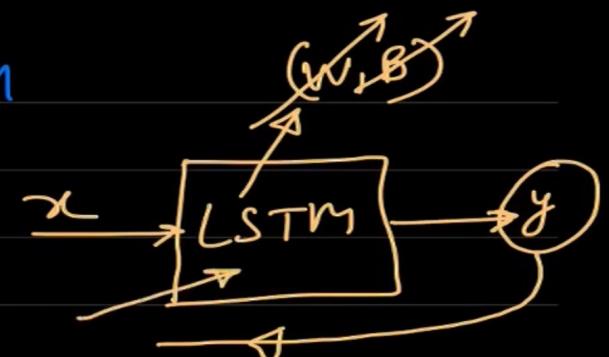
Backpropagation in LSTM

Loss:

$$L_t = -y_t \log \hat{y}_t$$

$$L = \sum_{j=0}^T L_j$$

$$\left[ \frac{\partial L}{\partial u_i}, \frac{\partial L}{\partial u_f}, \frac{\partial L}{\partial u_o}, \frac{\partial L}{\partial u_g} \right], \left[ \frac{\partial L}{\partial w_i}, \frac{\partial L}{\partial w_f}, \frac{\partial L}{\partial w_o}, \frac{\partial L}{\partial w_g} \right], \frac{\partial L}{\partial v}$$



→ Calculating gradients w.r.t weight involve calculation  
of gradients w.r.t gates & states

# LSTM – Backward Propagation Derivatives wrt to states

Derivative of loss function wrt  $h_t$

$$\textcircled{\times} \quad \sigma'(x) = \sigma(x)(1 - \sigma(x))$$

$$\left\{ \hat{y}_t, g_t, h_t \right\}$$

$$\frac{\partial L}{\partial g_t}, \frac{\partial L}{\partial c_t}, \frac{\partial L}{\partial h_t}$$

$$\textcircled{\times} \quad \tanh'(x) = 1 - \tanh^2(x)$$

# Calculation of  $\frac{\partial L}{\partial h_t}$  (Gradient w.r.t states)

$$\text{We know } \hat{y}_t = \text{softmax}(Vh_t) \quad \frac{\partial L}{\partial g_t} -$$

$$\text{Let } z_t = Vh_t, \quad \hat{y}_t = \text{softmax}(z_t)$$

$$\& \quad L = -y_t \log \hat{y}_t$$

$$\left( \frac{\partial L}{\partial h_t} \right) = \left\{ \frac{\partial L}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial z_t} \cdot \frac{\partial z_t}{\partial h_t} \right\} = \left( dh_t \right)$$

$$\frac{\partial L}{\partial h_t} = (\hat{y}_t - y_t) V^T = dh_t$$

$B \times P \leftrightarrow P \times H \leftrightarrow B \times H$

Derivative of loss function wrt  $C_t$

$$\# \quad \left( \frac{\partial L}{\partial C_t} \right) = ? = dC_t \quad \begin{cases} L = -y_t \log \hat{y}_t, \quad \hat{y}_t = \text{softmax}(Vh_t) \\ h_t = o_t \tanh(C_t) \end{cases}$$

Hadamard matrix

$$dC_t = \frac{\partial L}{\partial C_t} = \frac{\partial L}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial C_t} = (\hat{y}_t - y_t) V^T o_t (1 - \tanh^2(C_t))$$

$B \times P \leftrightarrow P \times H \leftrightarrow B \times H$

Hadamard Matrix

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \cdot \begin{bmatrix} x & y & z \\ m & n & o \end{bmatrix} = \begin{bmatrix} ax & by & cz \\ dm & en & fo \end{bmatrix}$$

Derivative of loss function wrt  $G_t$

$$\text{Similarly } \left( \frac{\partial L}{\partial g_t} \right) \text{ (candidate state)} = \left\{ \frac{\partial L}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial C_t} \cdot \frac{\partial C_t}{\partial g_t} \right\}$$

$$= (\hat{y}_t - y_t) V^T o_t (1 - \tanh^2(C_t)) \cdot i_t$$

$B \times P \leftrightarrow P \times H \leftrightarrow B \times H \leftrightarrow B \times H$

# LSTM – Backward Propagation Derivatives wrt to Gates

Derivative of loss function wrt  $O_t$

$$\textcircled{X} \quad \begin{array}{l} \text{Gradients wrt Gates} \leftrightarrow (\text{Output gate}) \\ \left\{ \frac{\partial L}{\partial O_t} = ? \right\} = dO_t \end{array}$$

$L = -y_t \log \hat{y}_t$

$\frac{\partial L}{\partial O_t}, \frac{\partial L}{\partial f_t}, \frac{\partial L}{\partial i_t}$

$\hat{y}_t = \text{softmax}(Vh_t)$

$h_t = O_t \tanh(C_t)$

$dO_t = \frac{\partial L}{\partial O_t} = \frac{\partial L}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial O_t} = (\hat{y}_t - y_t) V^T \tanh(C_t)$

Hadamard matrix

$B \times P \quad P \times N \quad B \times N$

Derivative of loss function wrt  $I_t$

$$\textcircled{X} \quad \left\{ \frac{\partial L}{\partial i_t} \right\} = ? \quad (\text{Input gate}) = di_t$$

$\frac{\partial L}{\partial i_t} = \frac{\partial L}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial C_t} \cdot \left( \frac{\partial C_t}{\partial i_t} \right) = di_t$

Hadamard matrix

$L = -y_t \log \hat{y}_t$

$\hat{y}_t = \text{softmax}(Vh_t)$

$h_t = O_t \tanh(C_t)$

$C_t = f_t C_{t-1} + i_t \hat{g}_t$

$\frac{\partial L}{\partial i_t} = (\hat{y}_t - y_t) V^T \bullet O_t (1 - \tanh^2 C_t) \bullet \hat{g}_t$

$B \times P \quad P \times N \quad B \times N \quad B \times N$

Derivative of loss function wrt  $f_t$

$$\textcircled{X} \quad \frac{\partial L}{\partial f_t} = ? \quad (\text{forget gate}) \quad df_t = \frac{\partial L}{\partial f_t}$$

$\frac{\partial L}{\partial f_t} = \left\{ \frac{\partial L}{\partial \hat{g}_t} \cdot \frac{\partial \hat{g}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial C_t} \cdot \frac{\partial C_t}{\partial f_t} \right\}$

$\frac{\partial L}{\partial f_t} = (\hat{y}_t - y_t) V^T \bullet O_t (1 - \tanh^2 C_t) \bullet C_{t-1} \quad \left( \frac{\partial L}{\partial f_t} \right) = df_t$

$B \times P \quad P \times N \quad B \times N \quad B \times N$

# LSTM – Backward Propagation Derivatives wrt to weights

Gradients with respect to weights

$$\frac{\partial L}{\partial V} = ?$$

$$\frac{\partial L}{\partial V} = \left\{ \sum_{i=1}^T \left\{ \frac{\partial L_i}{\partial V} \right\} \right\}$$

$$\left( \frac{\partial L}{\partial V} \right) = \sum_{i=1}^T \frac{\partial L_i}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z_i} \cdot \frac{\partial z_i}{\partial v} = \sum_{i=1}^T h_i^T (\hat{y}_i - y_i)$$

$\longleftrightarrow$   $H \times P$

$$\begin{array}{c} \hat{y}_i \\ \downarrow \text{FC} \\ h_i \\ \downarrow \text{FC} \\ \hat{y}_i \end{array} \quad \left\{ \frac{\partial L}{\partial w_i}, \frac{\partial L}{\partial w_o}, \frac{\partial L}{\partial w_g}, \frac{\partial L}{\partial w_f} \right\} \quad \frac{\partial L}{\partial v} \quad \left\{ h_i \xrightarrow{\text{FC}} \hat{y}_i \right\}$$

$$\left\{ \frac{\partial L}{\partial w_i} \right\} = ? \quad dW_i$$

$\frac{\partial L}{\partial w_i} = \sum_{t=0}^{T-1} \frac{\partial L_t}{\partial i_t} \cdot \frac{\partial i_t}{\partial w_i}$

$\frac{\partial L}{\partial w_i} = \sum_{t=0}^{T-1} (\hat{y}_t - \hat{y}_t) o_t (1 - \tanh^2 o_t) g_t \left\{ \frac{\partial i_t}{\partial w_i} \right\}$

$\frac{\partial i_t}{\partial w_i} = \left\{ \sigma(U_i x_t + W_i h_{t-1} + b_i) [1 - \sigma(U_i x_t + W_i h_{t-1} + b_i)] \right\}$

$$\text{We Know, } \{i_t\} = \sigma(U_i x_t + W_i h_{t-1} + b_i)$$

$$\text{So } \frac{\partial i_t}{\partial w_i} = i_t (1 - i_t) \otimes h_{t-1}$$

$$\text{So } \left\{ \frac{\partial L}{\partial w_i} \right\} = \sum_{t=0}^{T-1} \left[ (\hat{y}_t - \hat{y}_t) V \cdot o_t (1 - \tanh^2 o_t) \cdot g_t \cdot i_t (1 - i_t) \right] h_{t-1}$$

$\longleftrightarrow$   $H \times H$

$\frac{\partial L}{\partial w_i} \longleftrightarrow (H \times P) \cdot \dots \cdot n \times n$

# LSTM – Backward Propagation Derivatives wrt to weights contd...

$$\left\{ \frac{\partial L}{\partial w_f} \right\} = ?$$

$$\frac{\partial L}{\partial w_f} = \sum_{t=0}^{T-1} \left( \frac{\partial L_t}{\partial f_t} \right) \cdot \frac{\partial f_t}{\partial w_f}$$

$$= \sum_{t=0}^{T-1} (\hat{y}_t - \hat{y}_t) V \cdot o_t (1 - \tanh^2 c_t) \cdot c_{t-1} \frac{\partial f_t}{\partial w_f}$$

$$\frac{\partial f_t}{\partial w_f} = f_t (1 - f_t) \otimes h_{t-1}$$

$$\Rightarrow \left\{ \frac{\partial L}{\partial w_f} \right\} = \sum_{t=0}^{T-1} \left\{ (\hat{y}_t - \hat{y}_t) V \cdot o_t (1 - \tanh^2 c_t) \cdot c_{t-1} \right\} \cdot f_t (1 - f_t) \otimes h_{t-1}$$

$\longleftrightarrow$   
 $H \times B$

$$f_t = \sigma (V_f x_t + W_f h_{t-1} + b_f)$$

Next  $\left( \frac{\partial L}{\partial w_o} \right) = ? \quad (d w_o)$  Given  $o_t = \sigma (V_o x_t + W_o h_{t-1} + b_o)$

$$\frac{\partial L}{\partial w_o} = \sum_{t=0}^{T-1} \frac{\partial L_t}{\partial o_t} \cdot \frac{\partial o_t}{\partial w_o} = \sum_{t=0}^{T-1} \frac{\partial L_t}{\partial o_t} \cdot o_t (1 - o_t) \bullet h_{t-1}$$

$$\frac{\partial L}{\partial w_o} = \sum_{t=0}^{T-1} \left\{ \left[ (\hat{y}_t - \hat{y}_t) V^T \tanh(c_t) \right]^T \cdot o_t (1 - o_t) \right\} \bullet h_{t-1}$$

$\longleftrightarrow$   
 $B \times P$        $P \times H$        $B \times H$        $B \times H$        $H \times H$

$$\left( \frac{\partial L}{\partial w_g} \right) = ? \quad (d w_g)$$

We know  $g_t = \tanh (V_g x_t + W_g h_{t-1} + b_g)$

$$\frac{\partial L}{\partial w_g} = \sum_{t=0}^{T-1} \frac{\partial L_t}{\partial g_t} \left( \frac{\partial g_t}{\partial w_g} \right) = \sum_{t=0}^{T-1} \left( \frac{\partial L_t}{\partial g_t} \right) \cdot (1 - g_t^2) \bullet h_{t-1}$$

$$\left( \frac{\partial L}{\partial w_g} \right) = \sum_{t=0}^{T-1} \left\{ (\hat{y}_t - \hat{y}_t) V^T o_t (1 - \tanh^2 c_t) \cdot c_t \right\} \cdot (1 - g_t^2) \bullet h_{t-1}$$

$\longleftrightarrow$   
 $H \times B$        $B \times H$        $B \times H$        $H \times B$

# LSTM – Backward Propagation Derivatives wrt to weights contd...

Gradients w.r.t  $U$

$$\frac{\partial L}{\partial U_i} = \sum_{t=0}^{T-1} \frac{\partial L_t}{\partial i_t} \frac{\partial i_t}{\partial U_i} \quad \text{with } i_t = \sigma(U_i x_b + W_f h_{t-1} + b_i)$$

$$\frac{\partial U_i}{\partial XH} = \left[ \sum_{t=0}^{T-1} x_t^T \left( \frac{\partial L}{\partial i_t} \cdot i_t (1 - i_t) \right) \right]$$

*→ Already evaluated earlier*

$\frac{\partial L}{\partial i_t}$   $\frac{\partial i_t}{\partial U_i}$   
 $B \times H$   $B \times H$

$$\# \left( \frac{\partial L}{\partial U_f} \right) = \sum_{t=0}^{T-1} \frac{\partial L_t}{\partial f_t} \cdot \frac{\partial f_t}{\partial U_f}$$

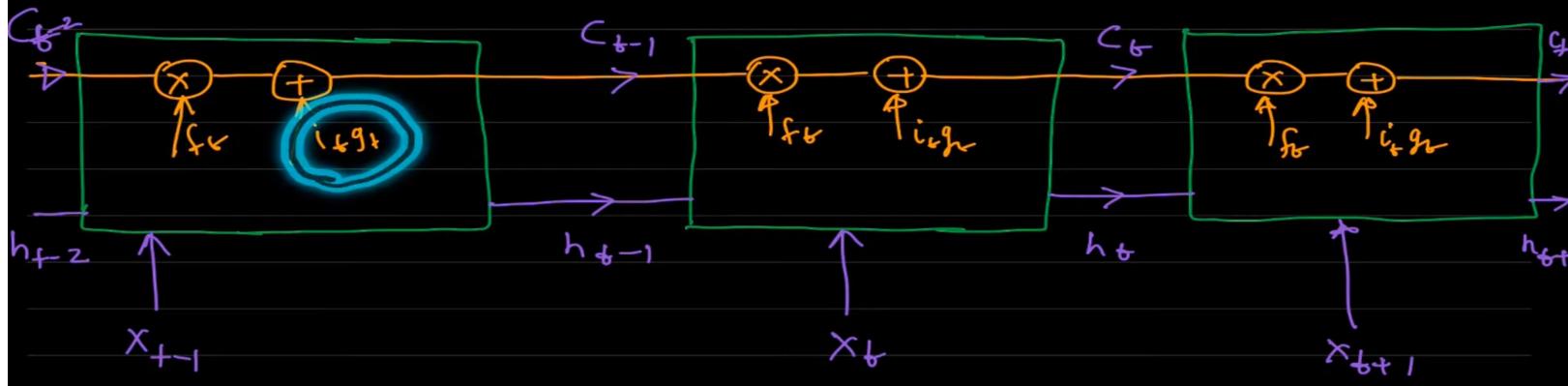
$$= \sum_{t=0}^{T-1} x_t^T \frac{\partial L_t}{\partial f_t} \cdot f_t (1 - f_t)$$

$$\# \left( \frac{\partial L}{\partial U_o} \right) = \sum_{t=0}^{T-1} \frac{\partial L_t}{\partial o_t} \frac{\partial o_t}{\partial U_o} = \sum_{t=0}^{T-1} x_t^T \frac{\partial L_t}{\partial o_t} \cdot o_t (1 - o_t)$$

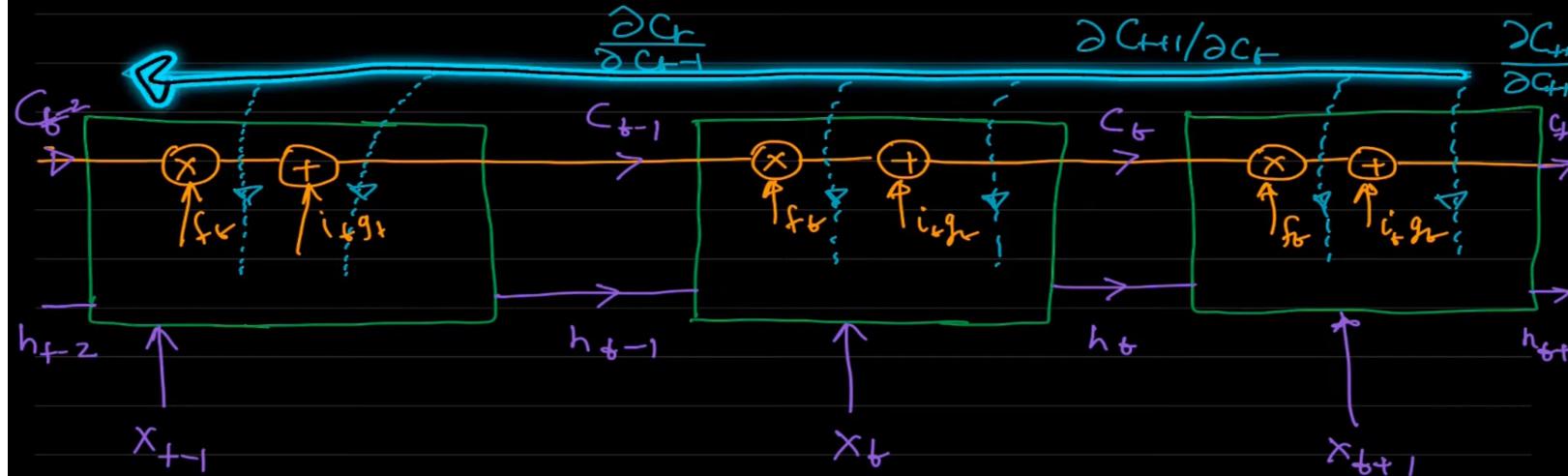
$$\# \left( \frac{\partial L}{\partial U_g} \right) = \sum_{t=0}^{T-1} \frac{\partial L_t}{\partial g_t} \cdot \frac{\partial g_t}{\partial U_g} = \sum_{t=0}^{T-1} x_t^T \frac{\partial L_t}{\partial g_t} \cdot g_t (1 - g_t)$$

# Vanishing Gradient Problem in LSTM

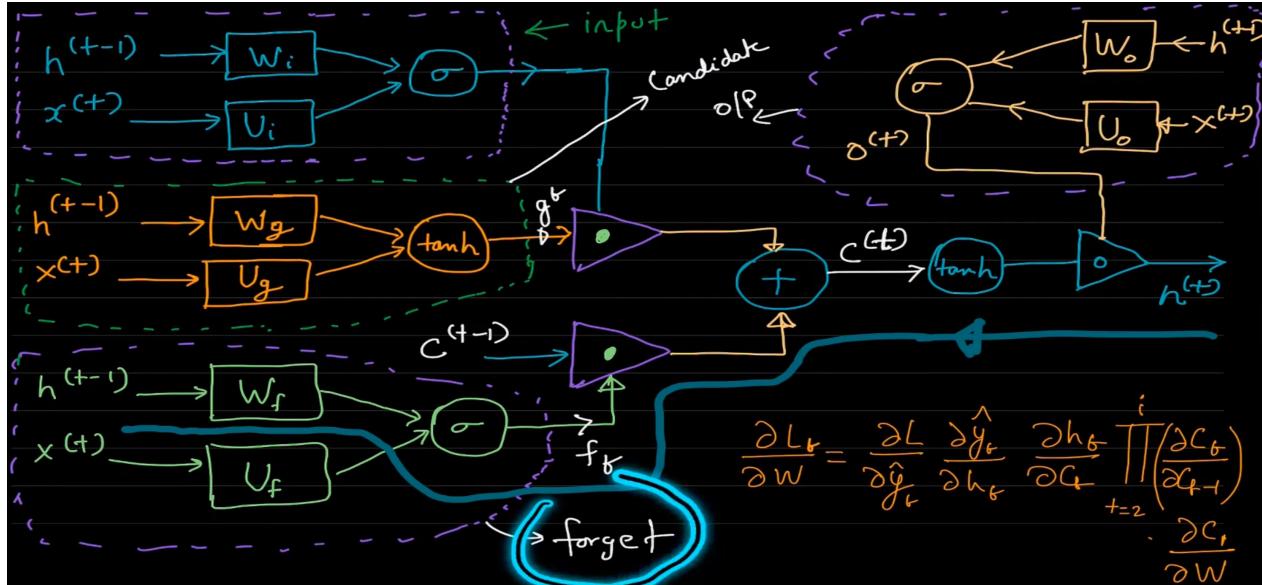
How LSTM solve the problem of vanishing gradient  
(Forward)



Back propagation



# LSTM - Gradients Flow through all Gates



$$\frac{\partial L_t}{\partial w} = \frac{\partial L}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial c_t} \left[ \prod_{t=2}^i \frac{\partial c_t}{\partial c_{t-1}} \right] \frac{\partial c_t}{\partial w}$$

(Motivation) (from RNN)

$$\frac{\partial L}{\partial w} = \sum_{i=1}^T \sum_{k=1}^i \frac{\partial L_i}{\partial y_i} \frac{\partial y_i}{\partial h_i} \left( \prod_{m=k+1}^i \frac{\partial h_m}{\partial h_{m-1}} \right) \frac{\partial h_i}{\partial w}$$

exploding  
vanishing

Solution: if we make ratio of  $\frac{\partial h_m}{\partial h_{m-1}} = 1$ , then no problem because  $(1)^{t-k} = 1$ . This is achieved

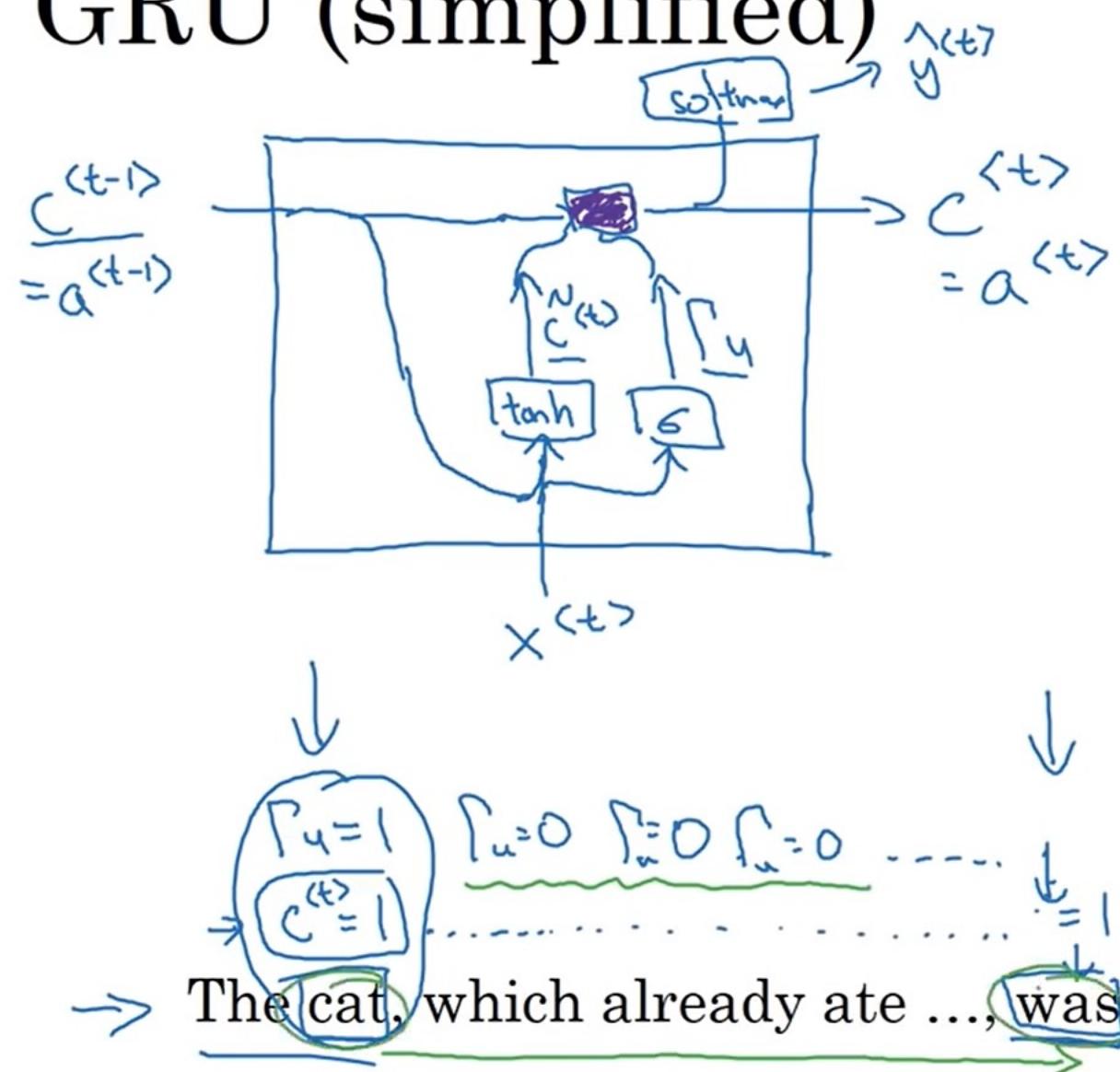
in case of LSTM where the architecture ensures

$$\frac{\partial c_t}{\partial c_{t-1}} = 1$$

Because this gradient is exactly one, it ensures that

information stored in memory cell not suffer from vanishing gradient.

# GRU (simplified)



$C$  = memory cell

$$\rightarrow C^{(t)} = \alpha^{(t)}$$

$$\rightarrow \tilde{C}^{(t)} = \tanh(W_c [C^{(t-1)}, x^{(t)}] + b_c)$$

$$\rightarrow \Gamma_u = \sigma(W_u [C^{(t-1)}, x^{(t)}] + b_u)$$

↑ "update"

$$\left\{ \begin{array}{l} C^{(t)} = \Gamma_u \times \tilde{C}^{(t)} + (1 - \Gamma_u) \times C^{(t-1)} \\ \Gamma_u = 0.000001 \end{array} \right.$$

Gate

[Cho et al., 2014. On the properties of neural machine translation: Encoder-decoder approaches]

[Chung et al., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling]



# Full GRU

$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * \underline{c}^{<t-1>}, \underline{x}^{<t>}] + b_c)$$

$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

LSTM

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

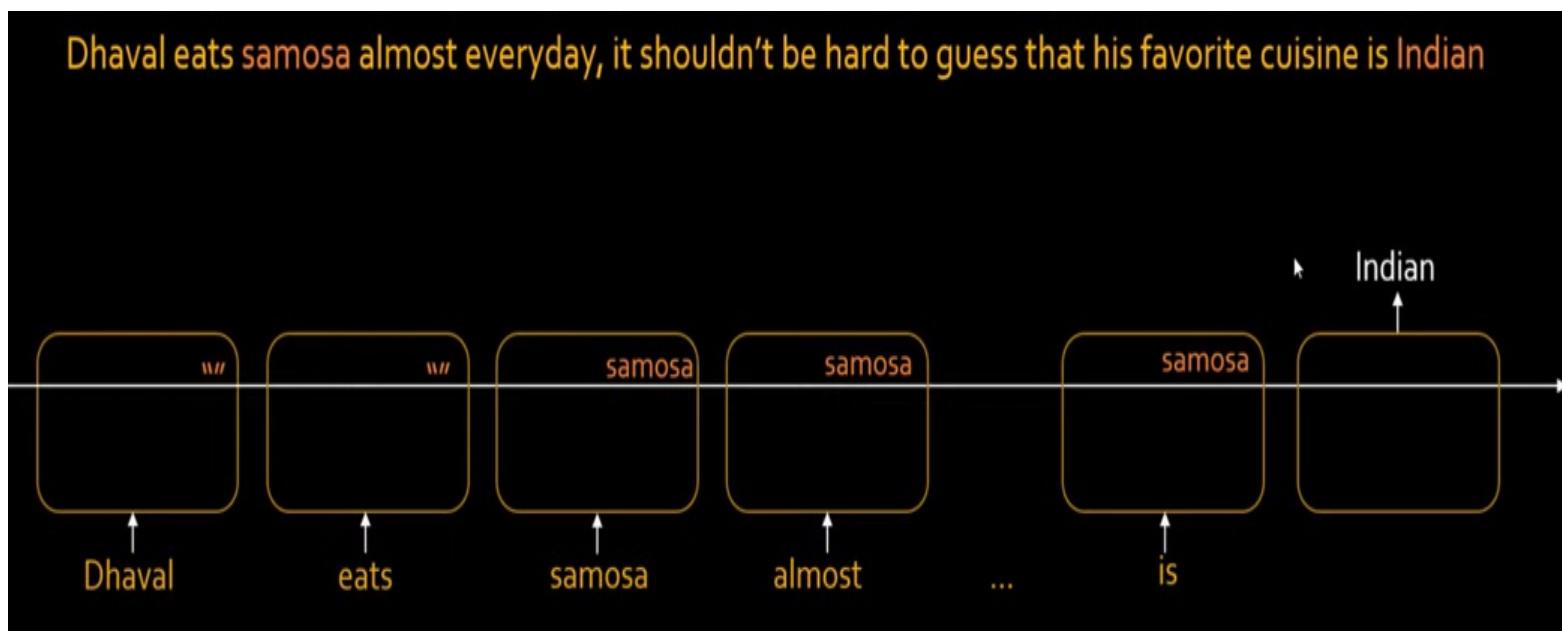
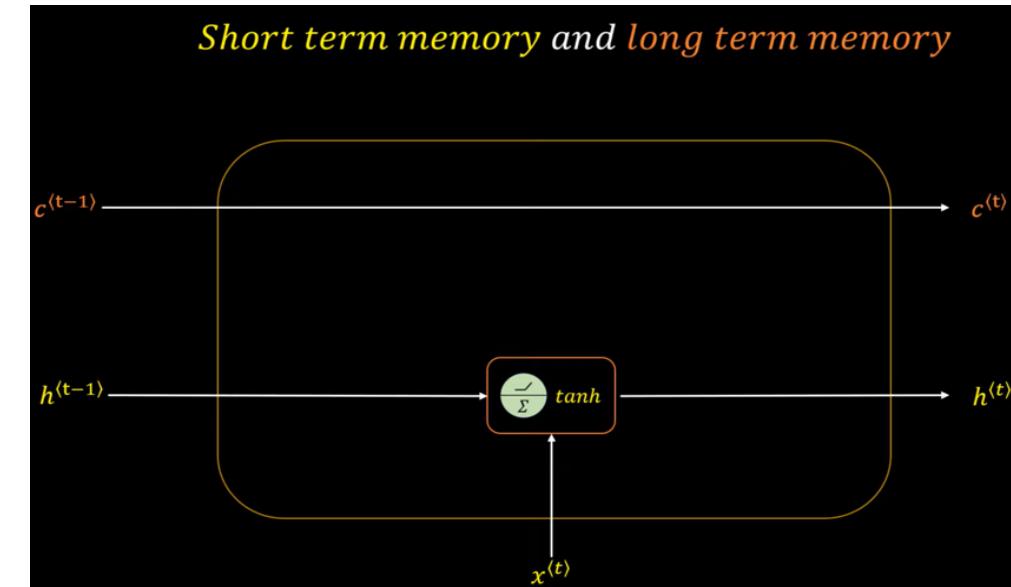
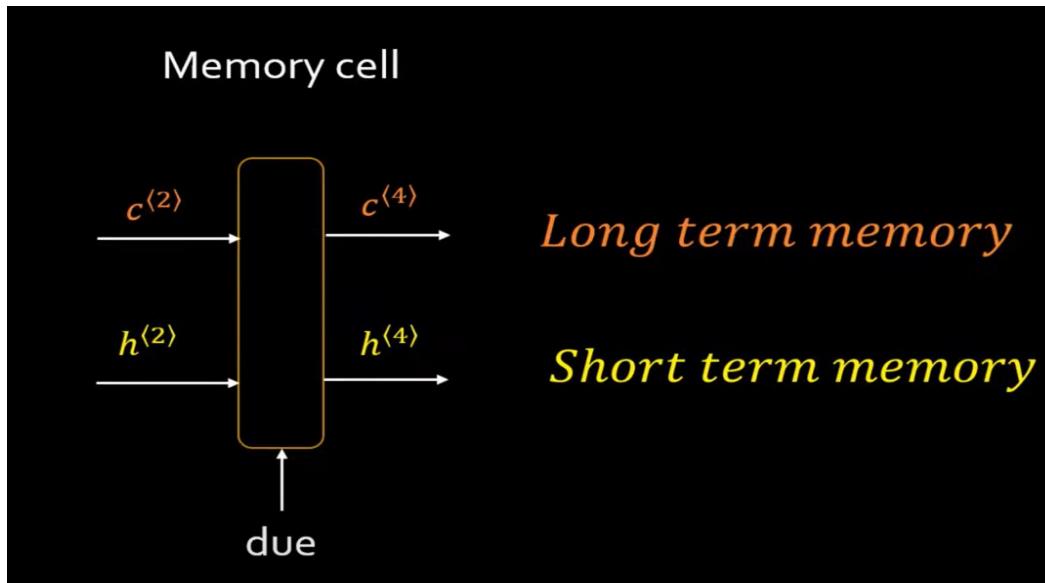
The cat, which ate already, was full.



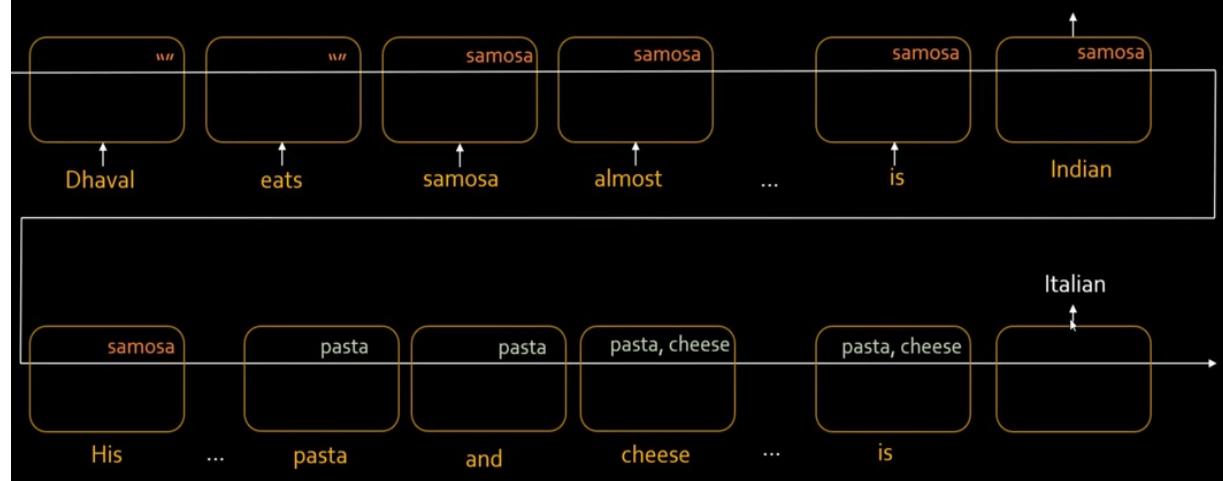
# Why tanh is used in LSTM

- <https://stackoverflow.com/questions/40761185/what-is-the-intuition-of-using-tanh-in-lstm>
- [https://www.reddit.com/r/MachineLearning/comments/9elxs8/why do you use tanh in a rnn/](https://www.reddit.com/r/MachineLearning/comments/9elxs8/why_do_you_use_tanh_in_a_rnn/)

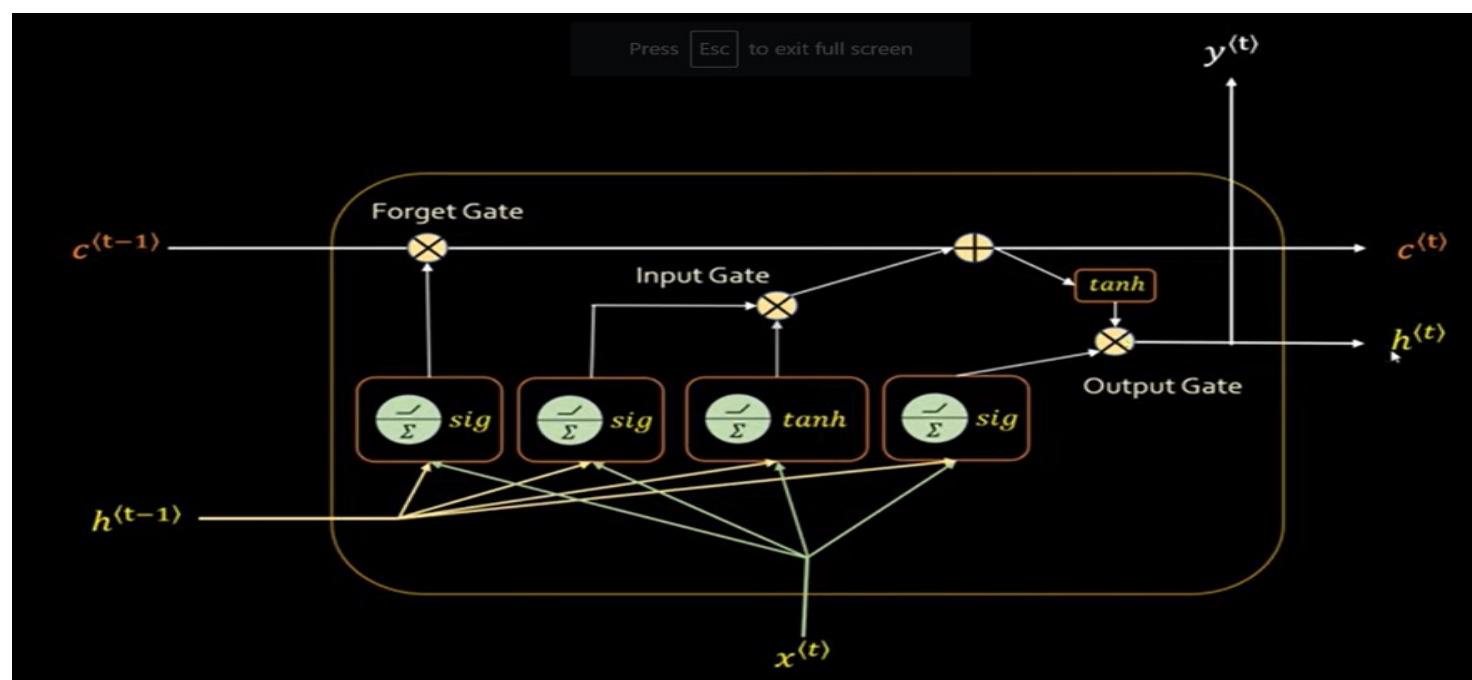
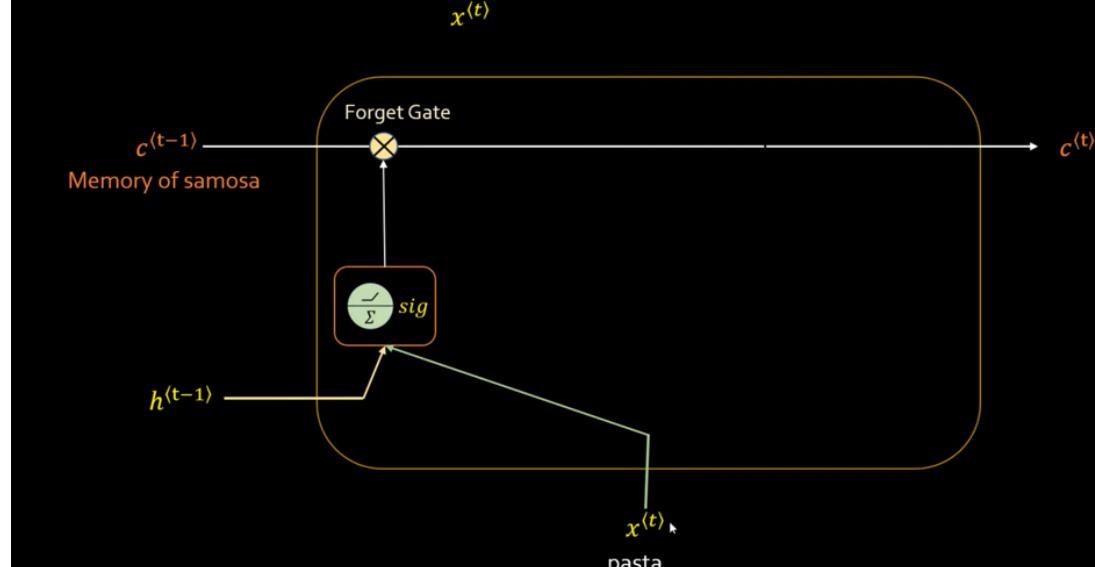
# LSTM Example:



Dhaval eats samosa almost everyday, it shouldn't be hard to guess that his favorite cuisine is Indian. His brother Bhavin however is a lover of pasta and cheese that means Bhavin's favorite cuisine is Italian



Dhaval eats samosa almost everyday, it shouldn't be hard to guess that his favorite cuisine is Indian. His brother Bhavin however is a lover of pasta and cheese that means Bhavin's favorite cuisine is Italian



# LSTM in pictures

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$

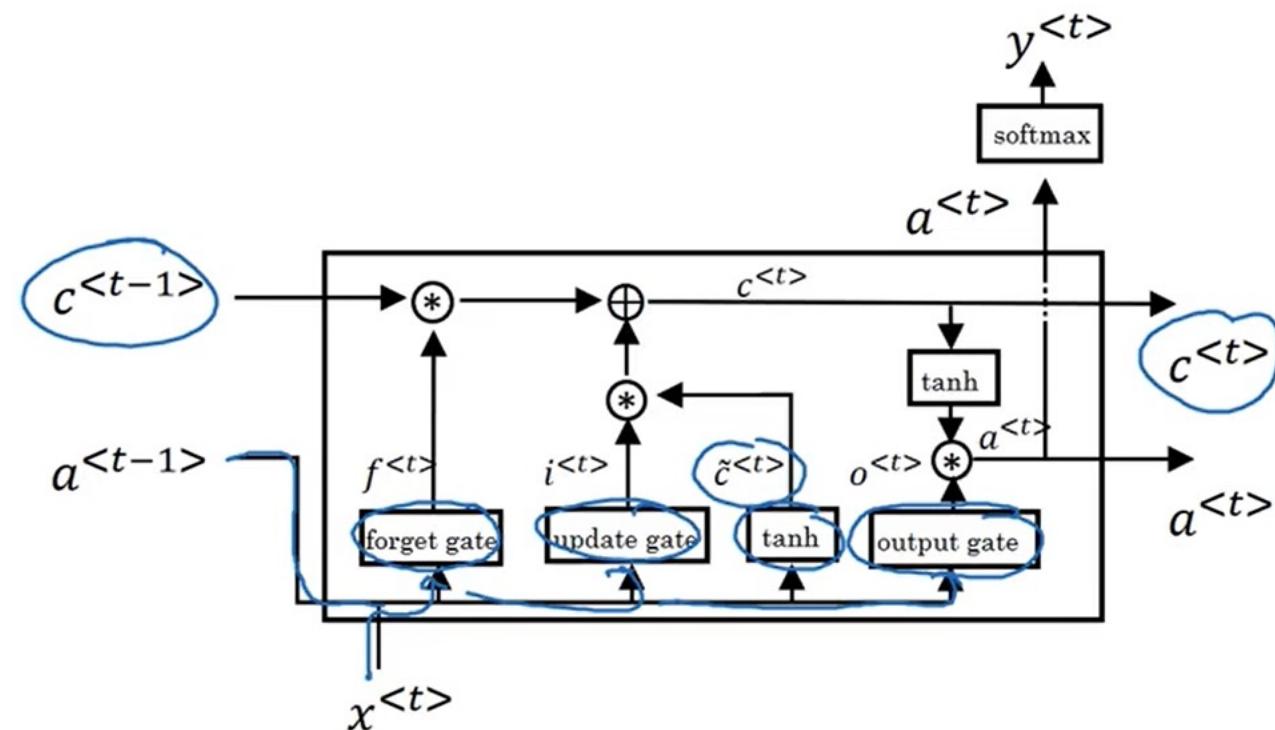
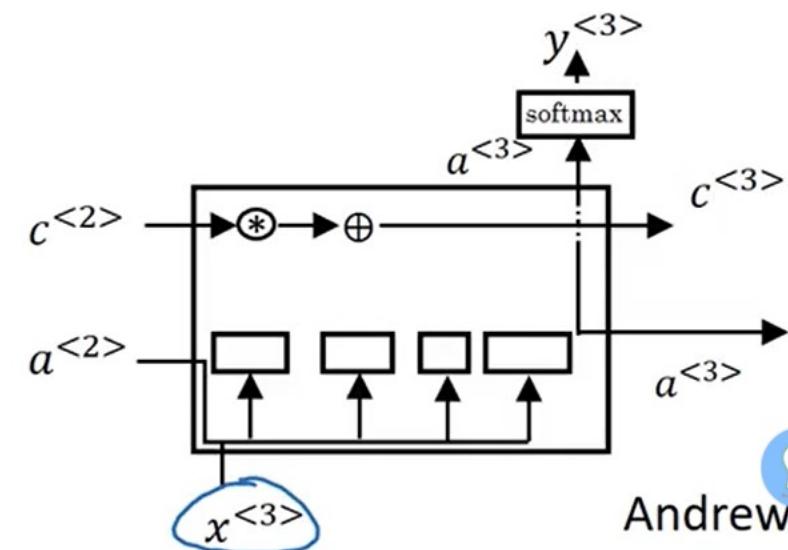
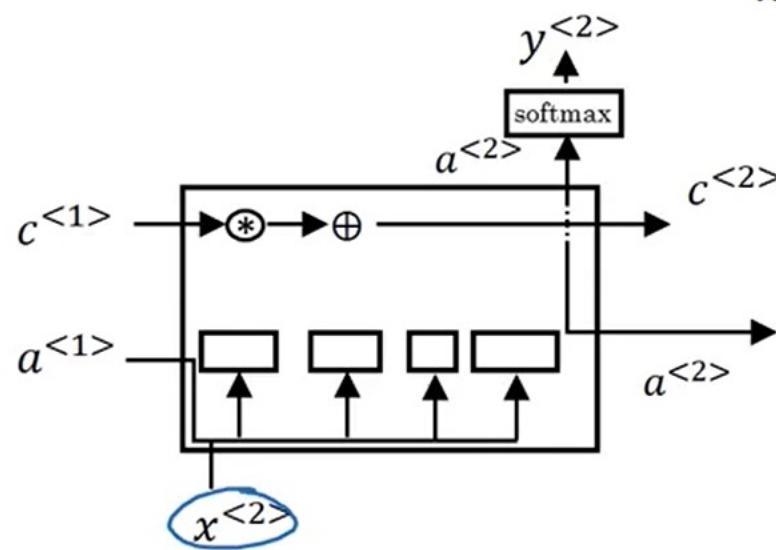
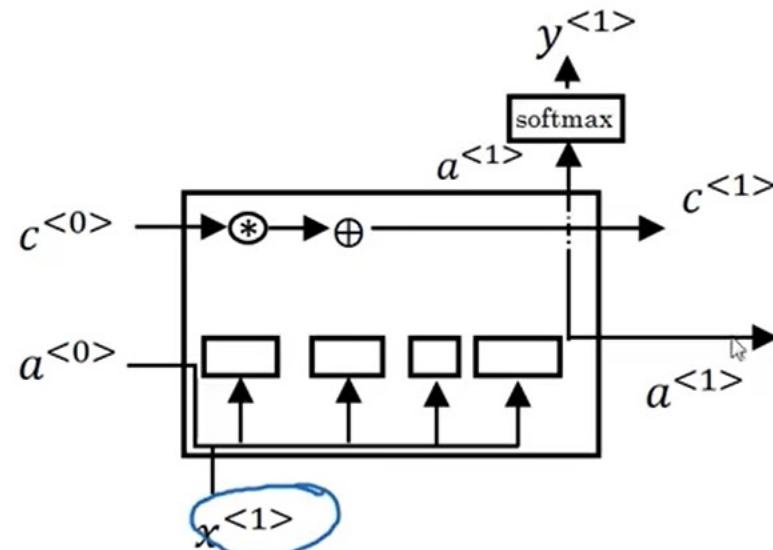
$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

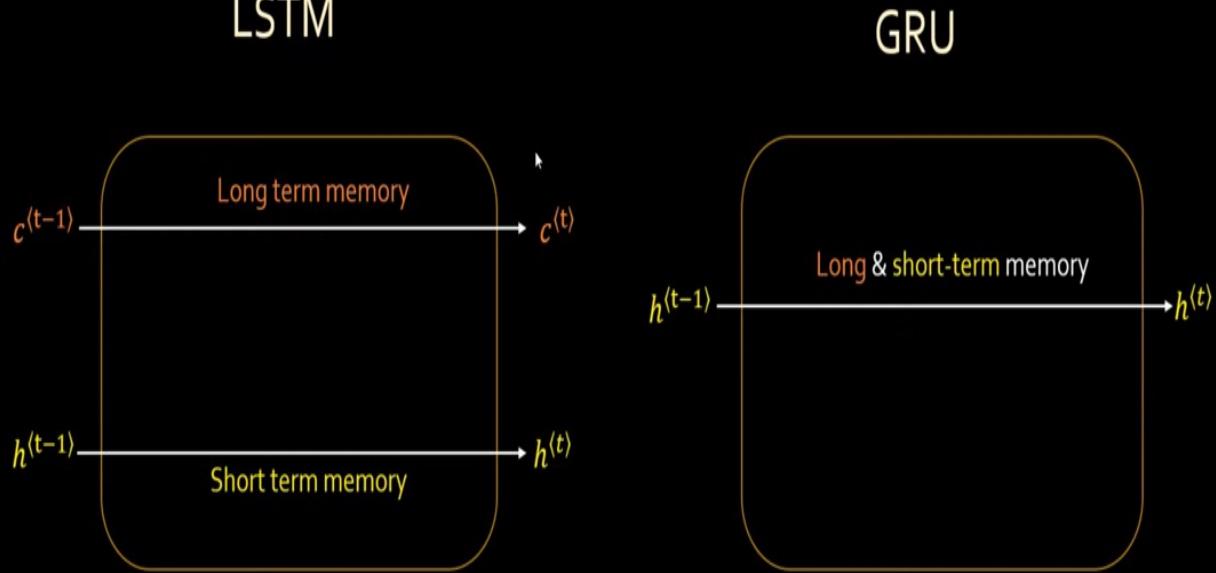
$$a^{<t>} = \Gamma_o * \tanh c^{<t>}$$



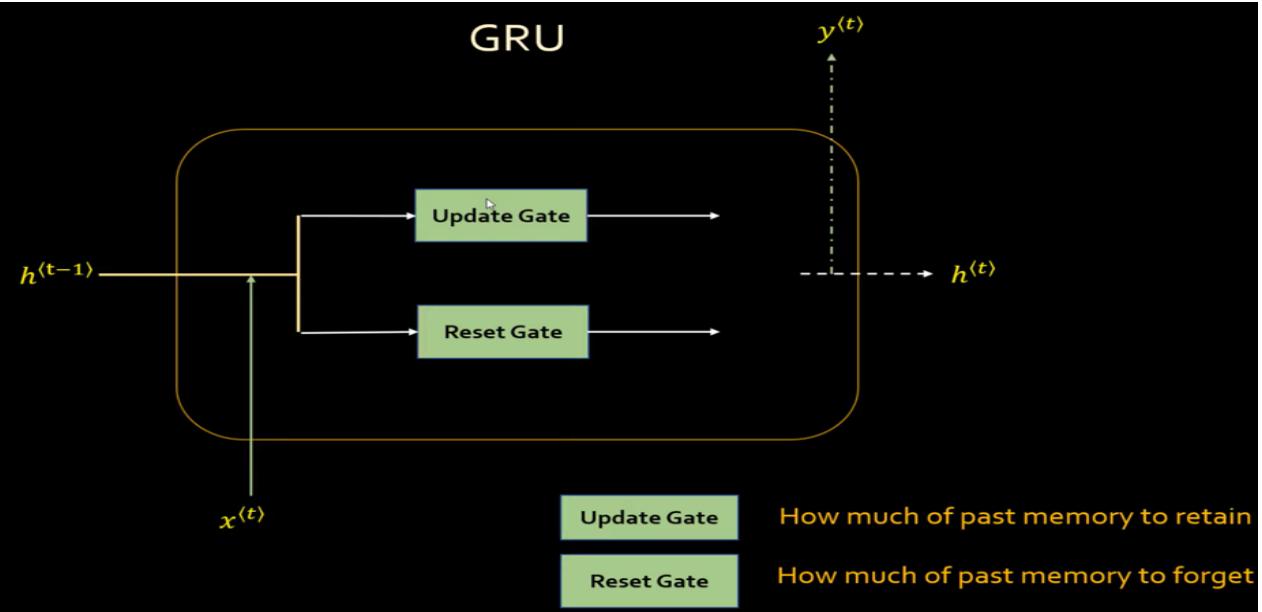
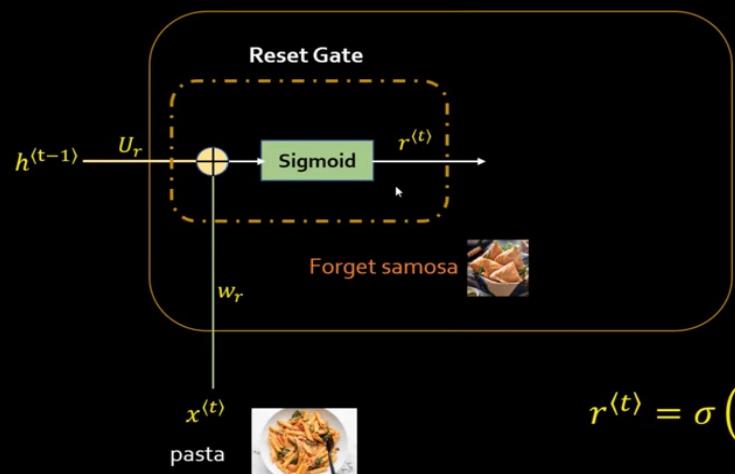
Andrew Ng

# RNN Example

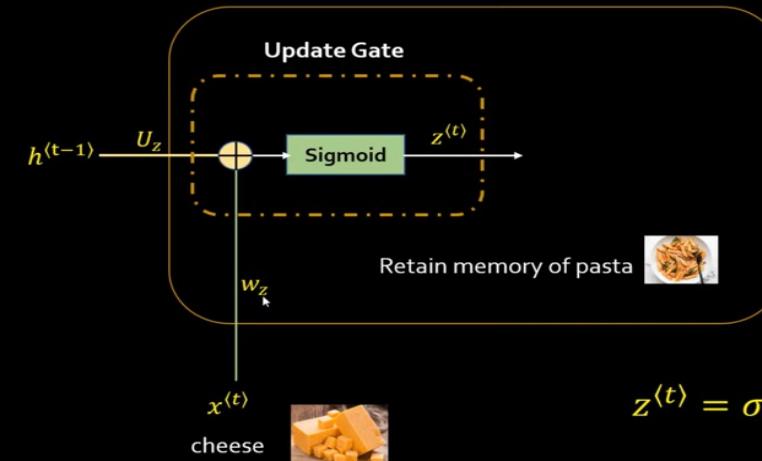
LSTM



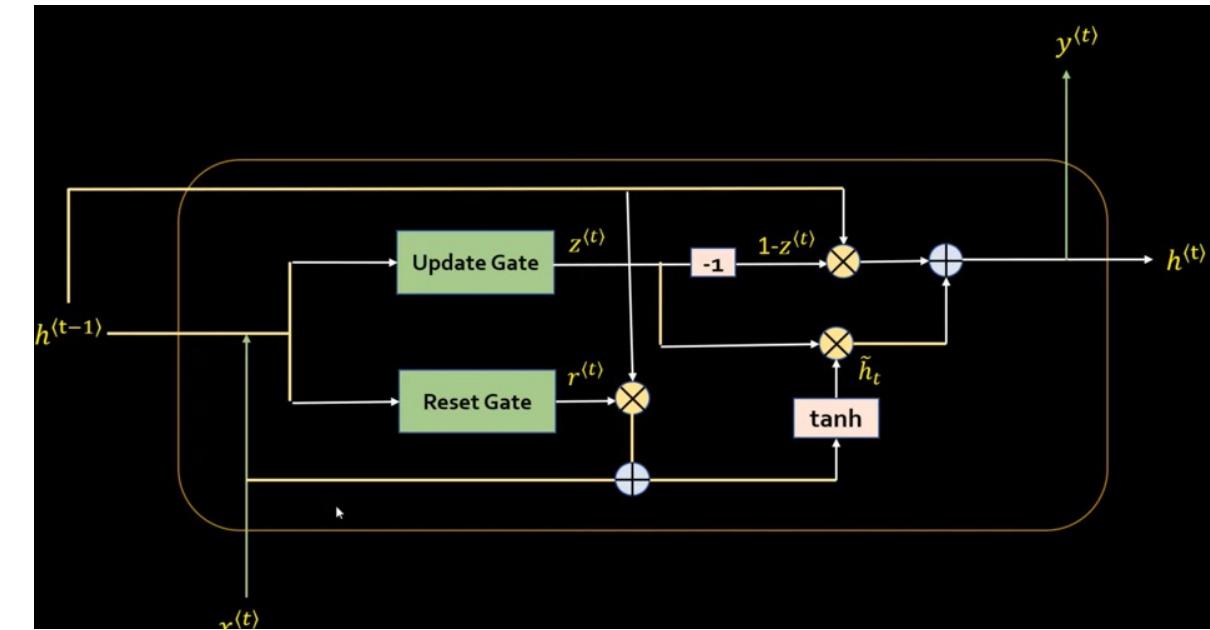
Dhaval eats samosa almost everyday, it shouldn't be hard to guess that his favorite cuisine is Indian. His brother Bhavin however is a lover of pasta and cheese that means Bhavin's favorite cuisine is Italian



Dhaval eats samosa almost everyday, it shouldn't be hard to guess that his favorite cuisine is Indian. His brother Bhavin however is a lover of pasta and cheese that means Bhavin's favorite cuisine is Italian



# RNN Example Contd...



LSTM	GRU
3 Gates: Input, output, forget	2 Gates: reset, update
More accurate on longer sequence, less efficient	More efficient computation wise. Getting more popular
Invented: 1995 - 1997	Invented: 2014