# Recurrent Neural Network
## (CS.231)

# Sequence data

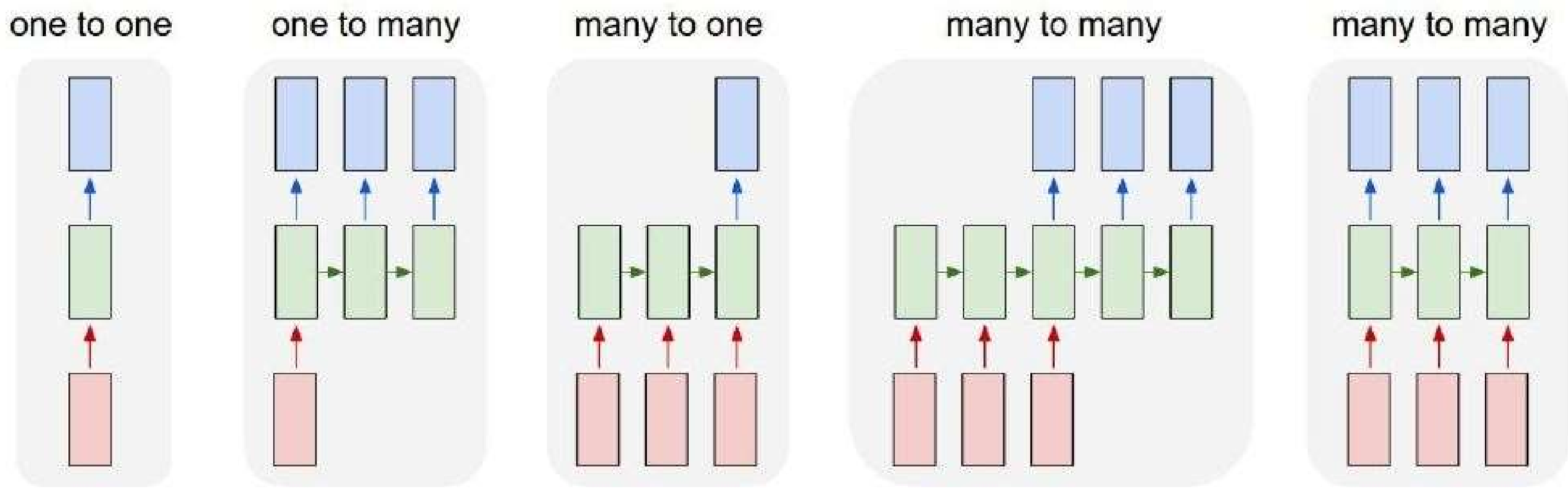
Sequential data includes text streams, audio clips, video clips, time-series data and etc.

# Recurrent Neural Network

RNN maintains internal memory
RNN is very efficient for machine learning problems that involve sequential data
RNNs are also used in time series predictions

https://towardsdatascience.com/sequence-models-and-recurrent-neural-networks-rnns-62cadeb4f1e1

# Recurrent Networks offer a lot of flexibility:




**Vanilla Neural Networks**

**Image Captioning**
(image → sequence of words)

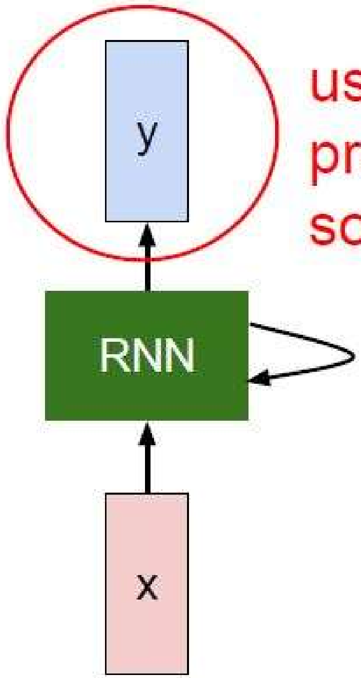
**Sentiment Classification**
(seq. of words → sentiment)

**Machine Translation**
(seq. of words → seq. of words)

**Video classification on frame level**

Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Recurrent Neural Network




usually want to predict a vector at some time steps

Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Recurrent Neural Network

We can process a sequence of vectors **x** by applying a recurrence formula at every time step:

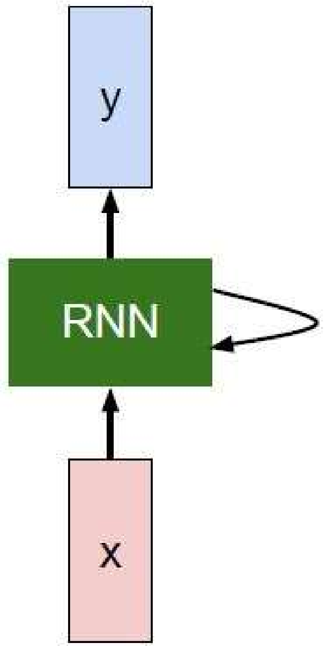
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

some function with parameters W

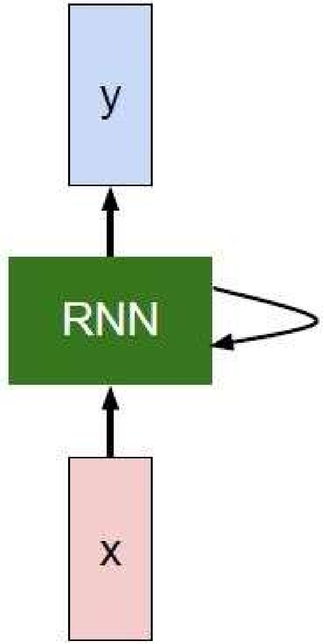



Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Recurrent Neural Network

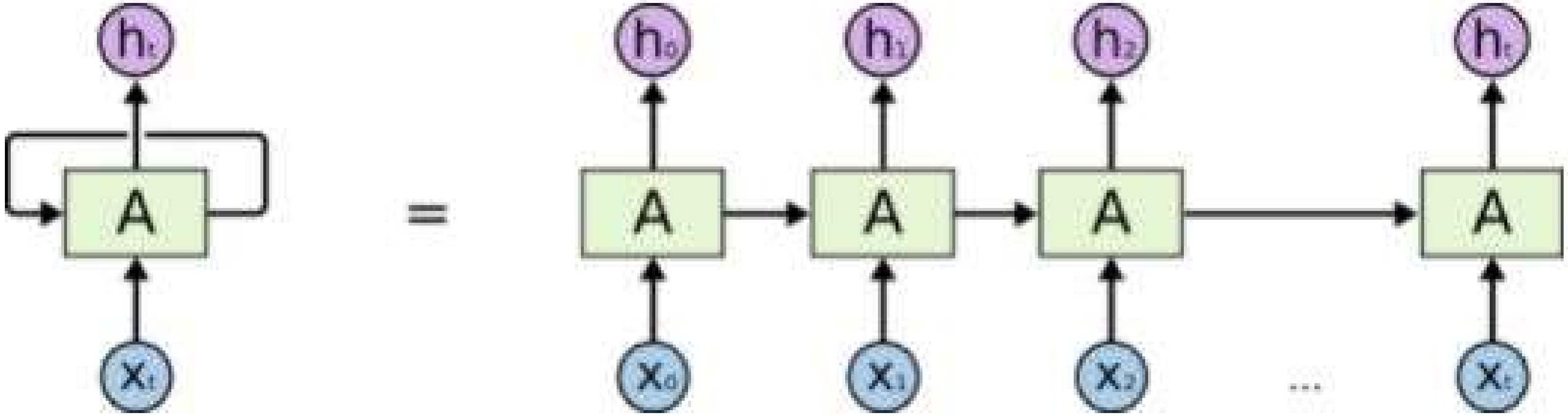
We can process a sequence of vectors **x** by applying a recurrence formula at every time step:

$$h_t = f_W(h_{t-1}, x_t)$$




Notice: the same function and the same set of parameters are used at every time step.

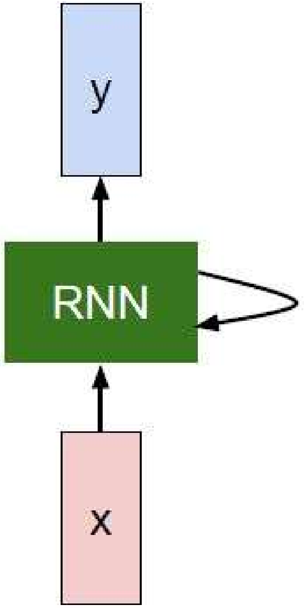
Fei-Fei Li & Andrej Karpathy & Justin Johnson




http://colah.github.io/posts/2015-08-Understanding-LSTMs/

# (Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:




$$h_t = f_W(h_{t-1}, x_t)$$

$$\downarrow$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

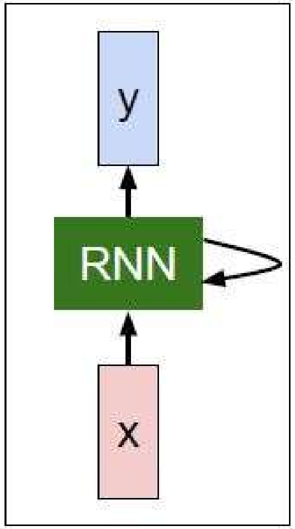
$$y_t = W_{hy}h_t$$

Fei-Fei Li & Andrej Karpathy & Justin Johnson

**Character-level language model example**

Vocabulary:
[h,e,l,o]

Example training sequence:
**“hello”**

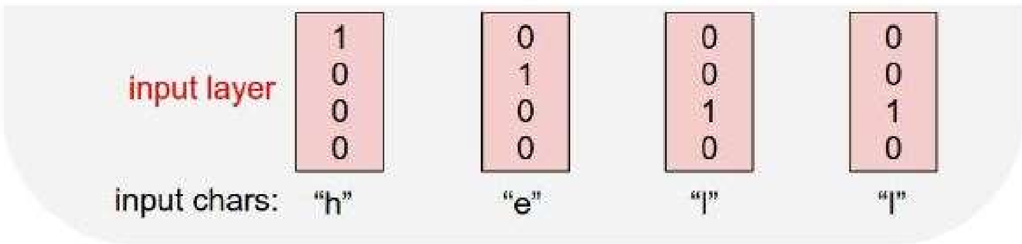



Fei-Fei Li & Andrej Karpathy & Justin Johnson

**Character-level language model example**

Vocabulary:
[h,e,l,o]

Example training sequence:
**"hello"**



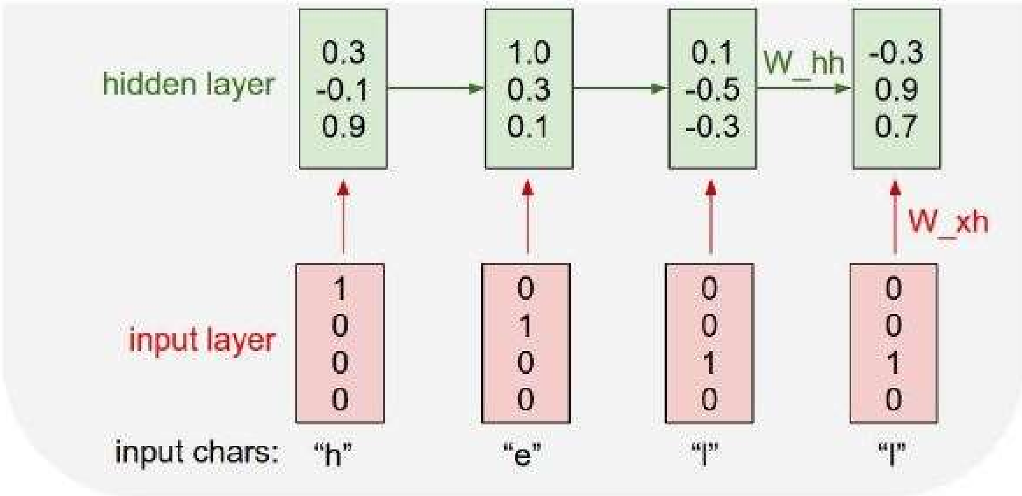

Fei-Fei Li & Andrej Karpathy & Justin Johnson

**Character-level language model example**

Vocabulary:
[h,e,l,o]

Example training sequence:
**“hello”**

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

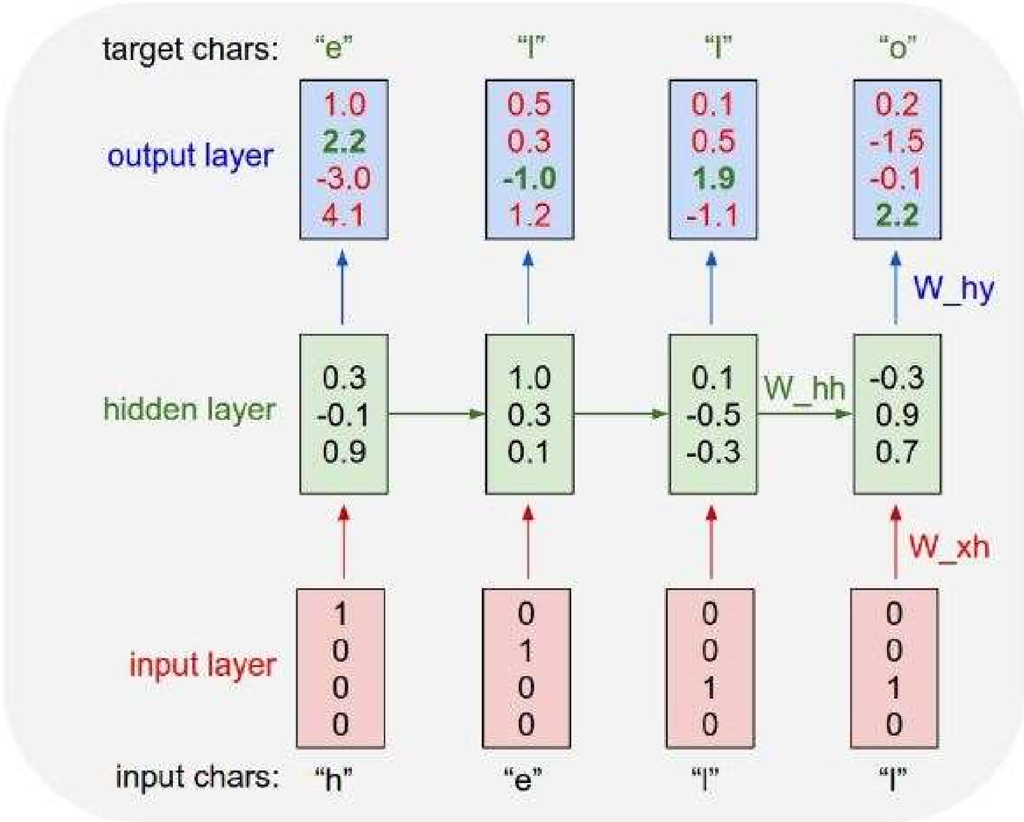



Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Character-level language model example

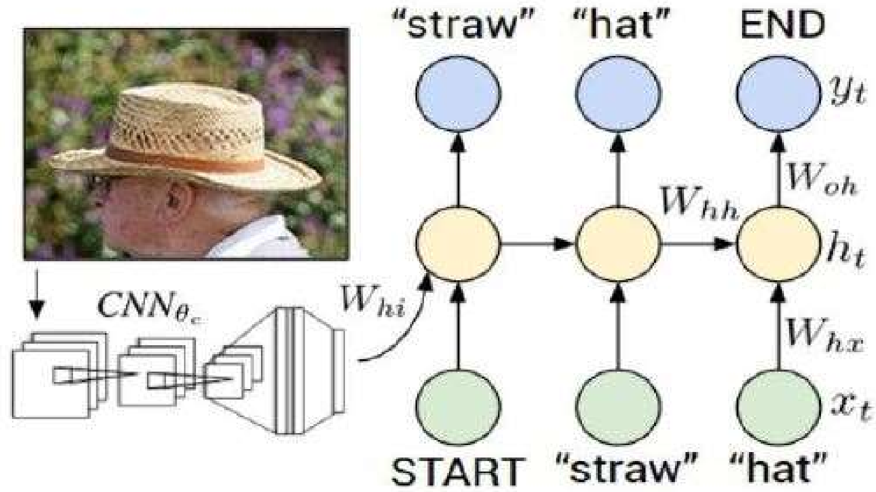
Vocabulary:
[h,e,l,o]

Example training sequence:
**“hello”**



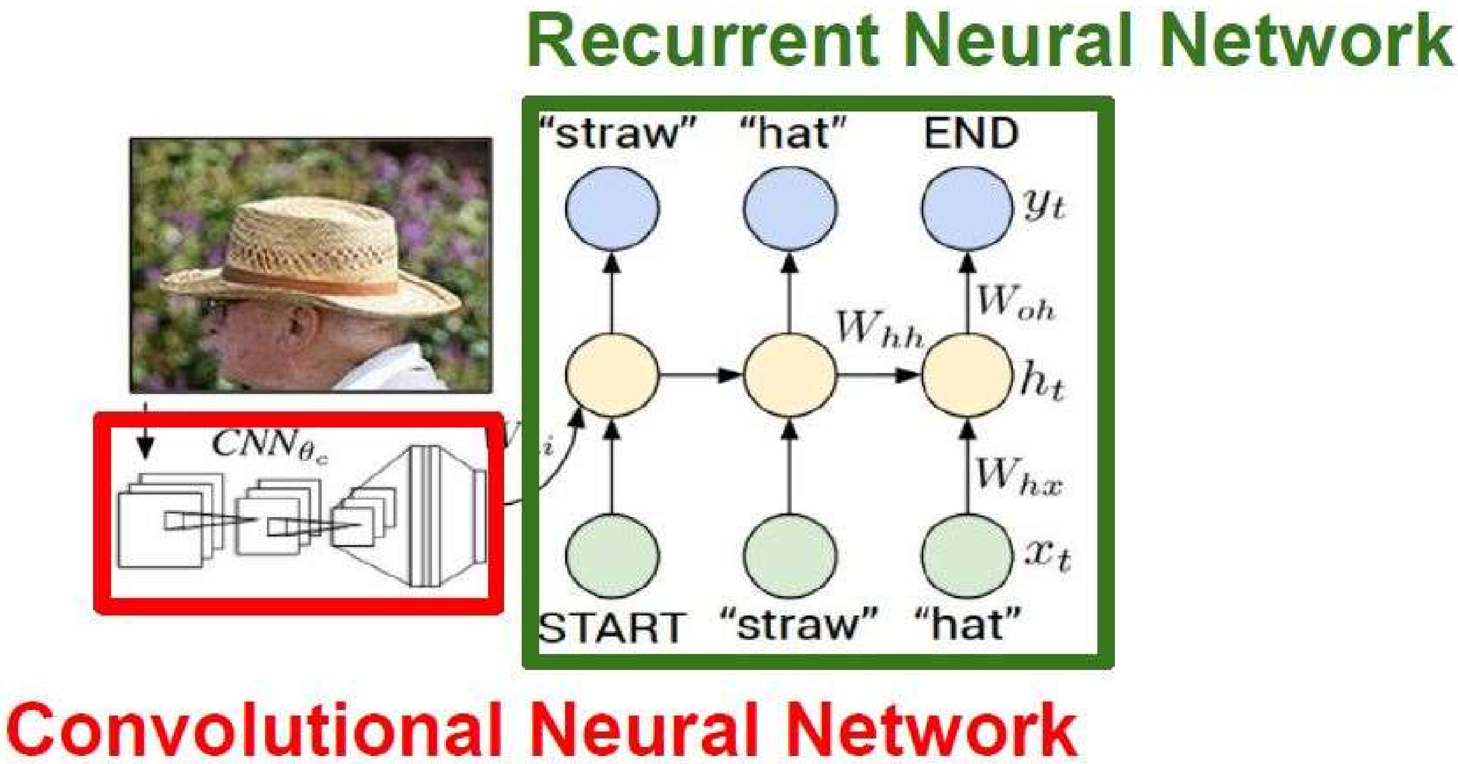

Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Image Captioning

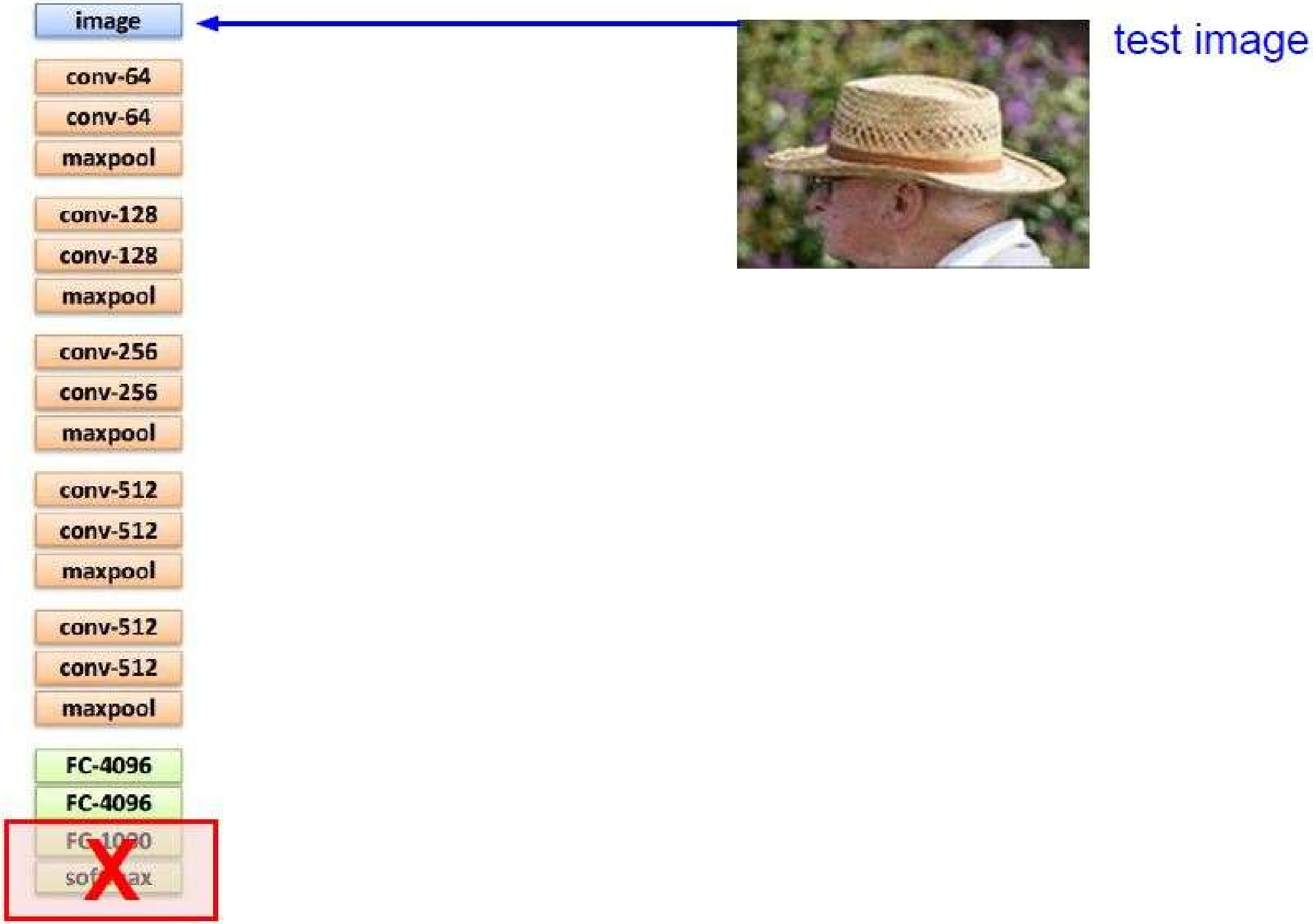



Explain Images with Multimodal Recurrent Neural Networks, Mao et al.
Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
Show and Tell: A Neural Image Caption Generator, Vinyals et al.
Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

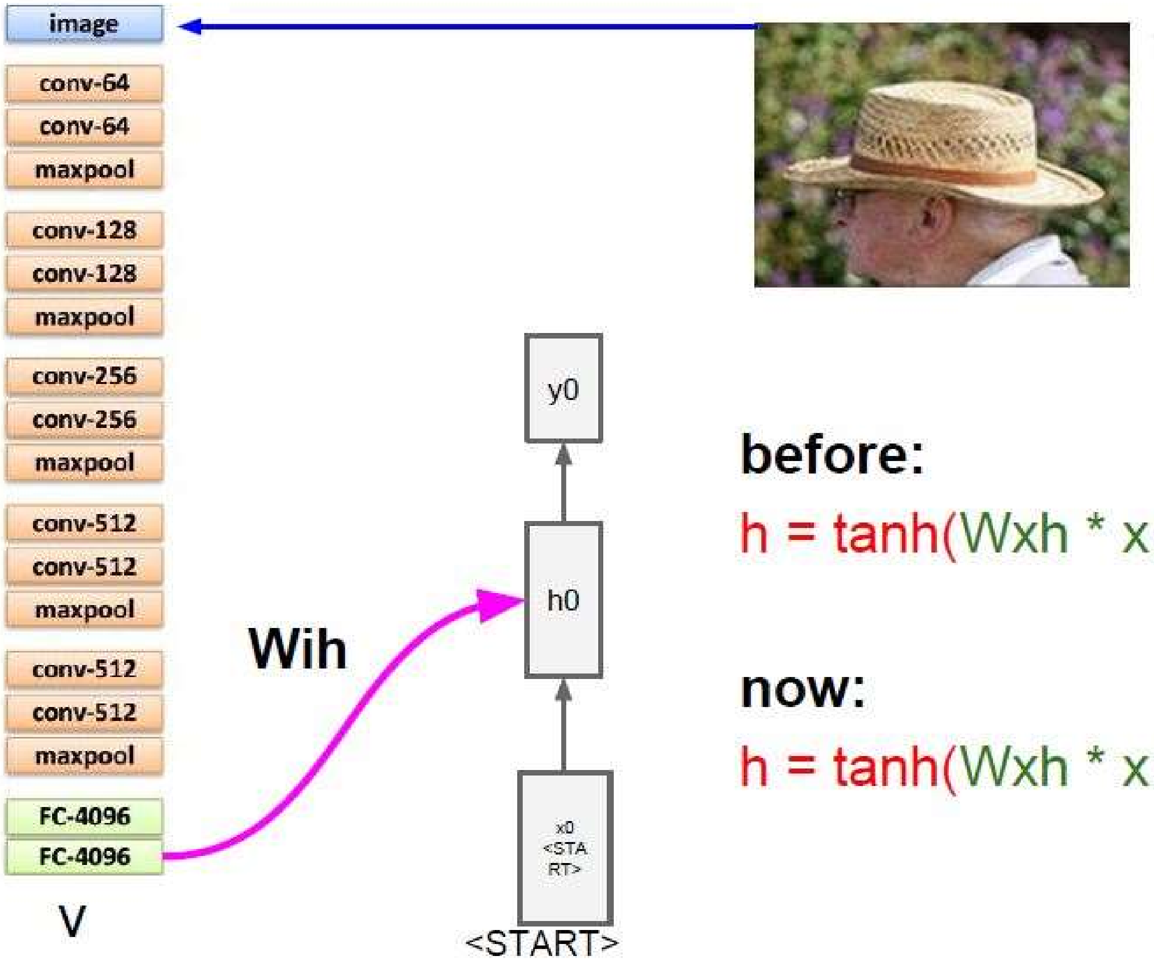
Fei-Fei Li & Andrej Karpathy & Justin Johnson




Fei-Fei Li & Andrej Karpathy & Justin Johnson




Fei-Fei Li & Andrej Karpathy & Justin Johnson



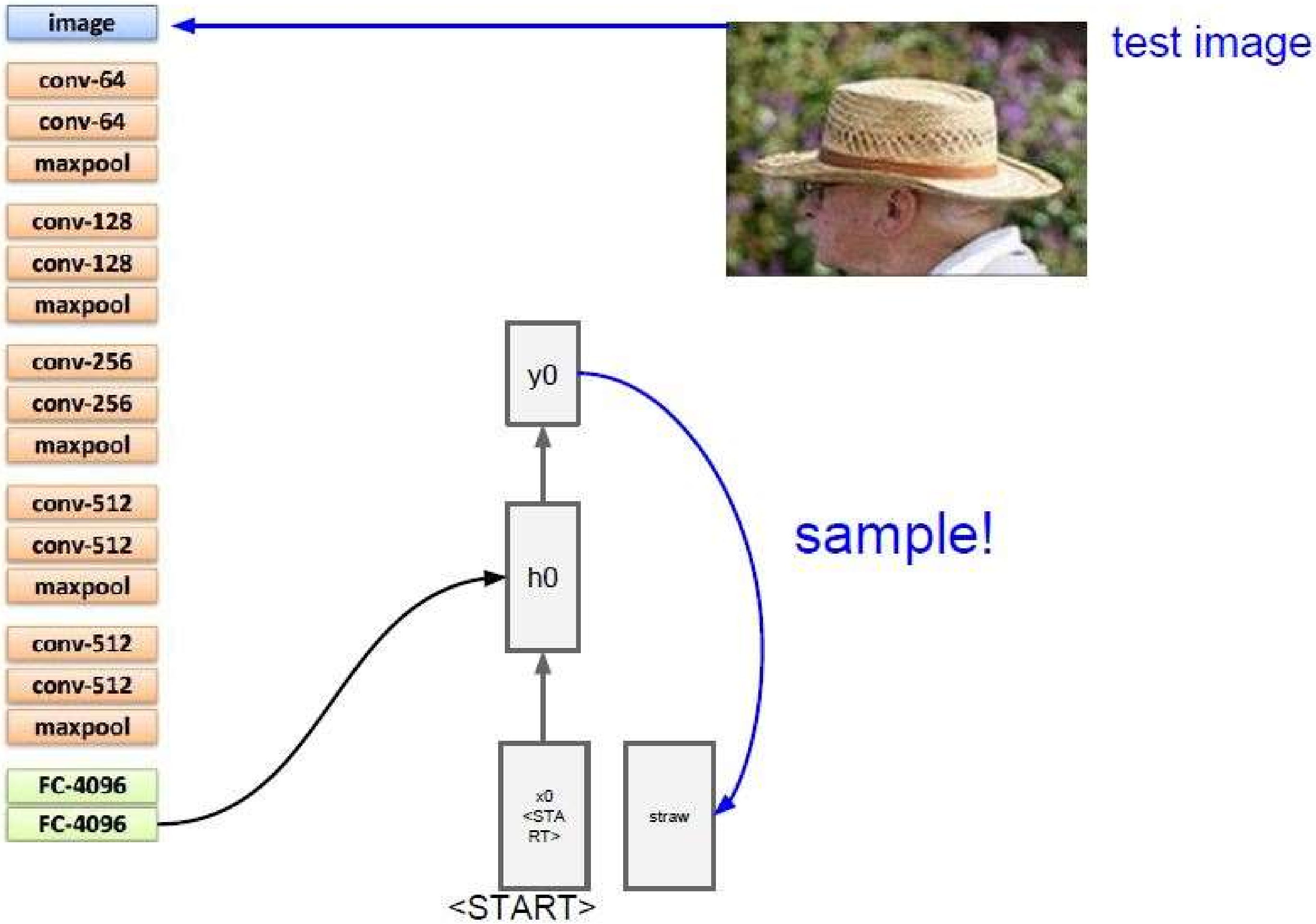

**before:**

h = tanh(Wxh * x + Whh * h)

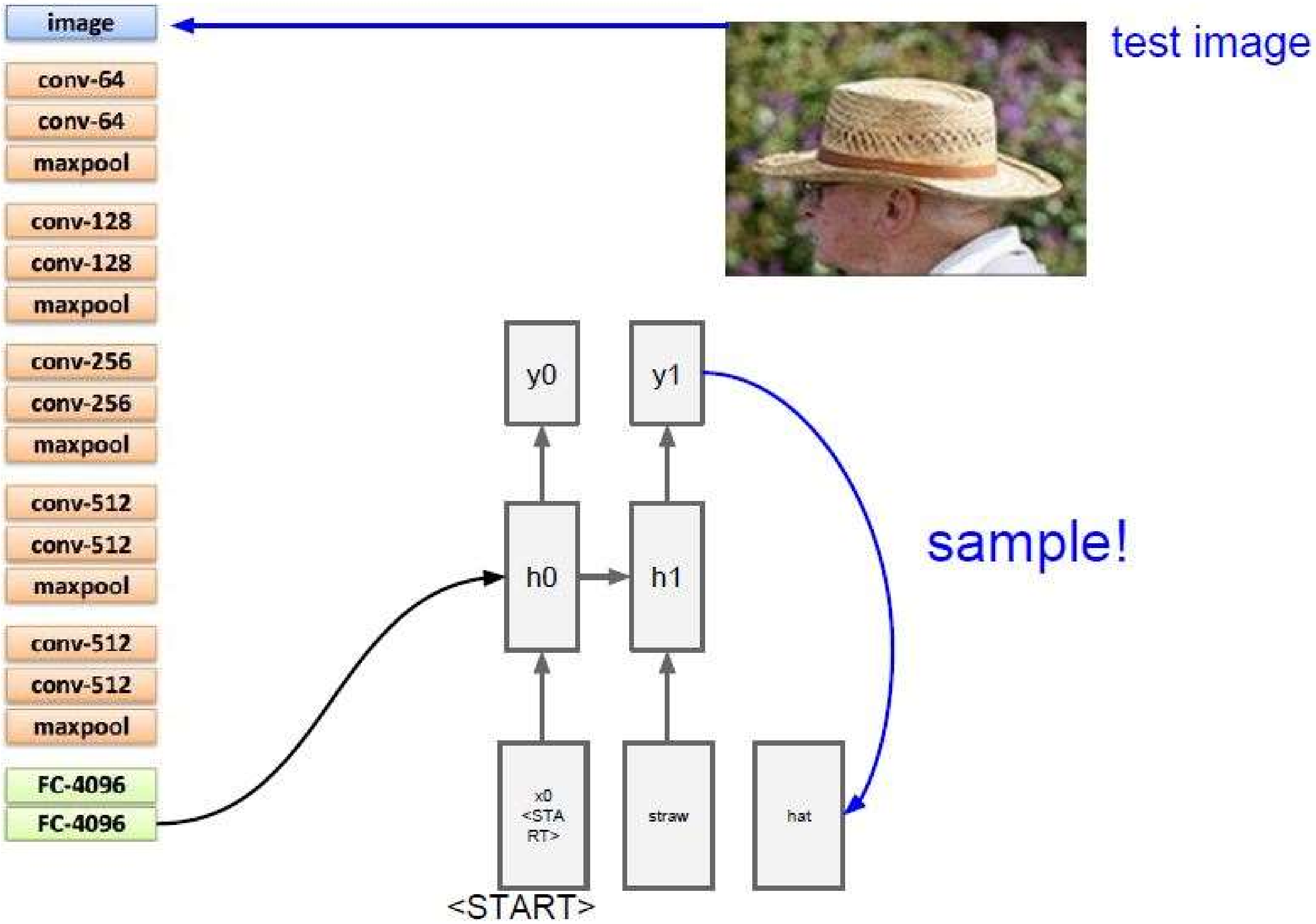
**now:**

h = tanh(Wxh * x + Whh * h + Wih * v)

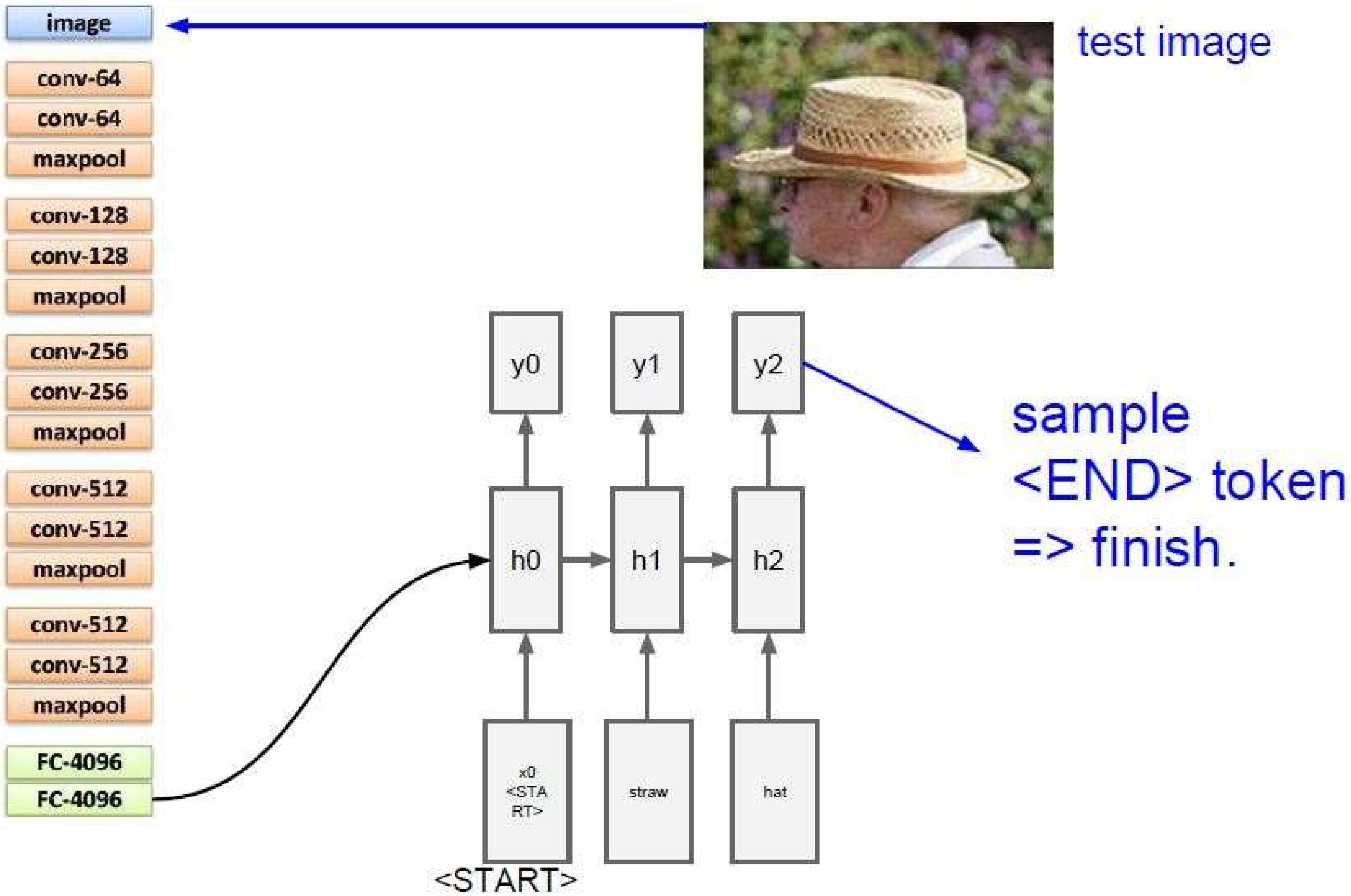
Fei-Fei Li & Andrej Karpathy & Justin Johnson




Fei-Fei Li & Andrej Karpathy & Justin Johnson




Fei-Fei Li & Andrej Karpathy & Justin Johnson

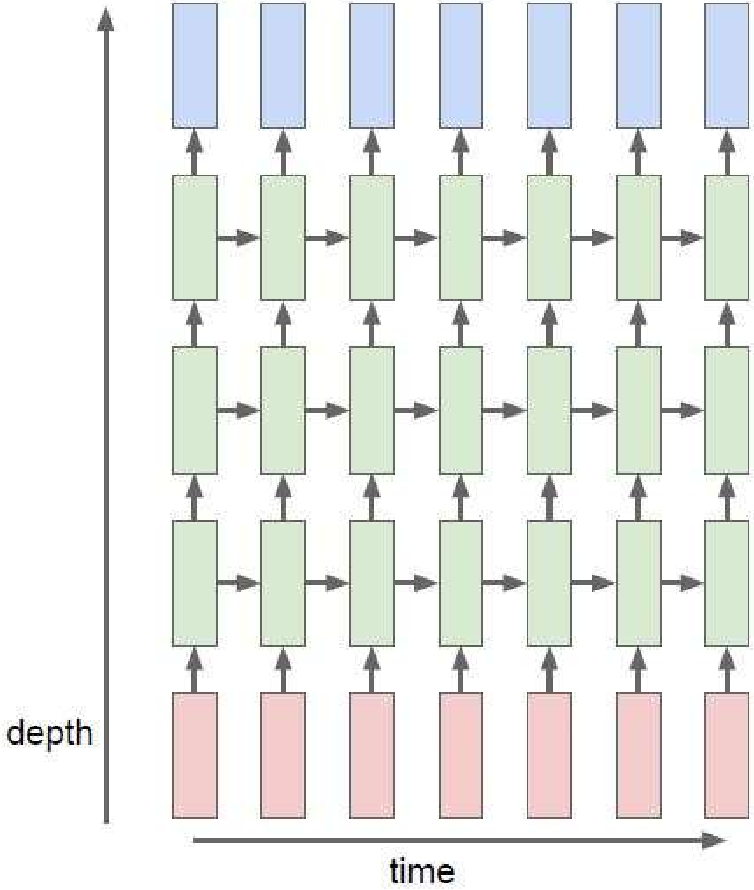



Fei-Fei Li & Andrej Karpathy & Justin Johnson

# RNN:

$$\boxed{h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}}$$

$h \in \mathbb{R}^n$. $\qquad W^l \; [n \times 2n]$




Fei-Fei Li & Andrej Karpathy & Justin Johnson

## RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$h \in \mathbb{R}^n$. $\quad W^l \ [n \times 2n]$

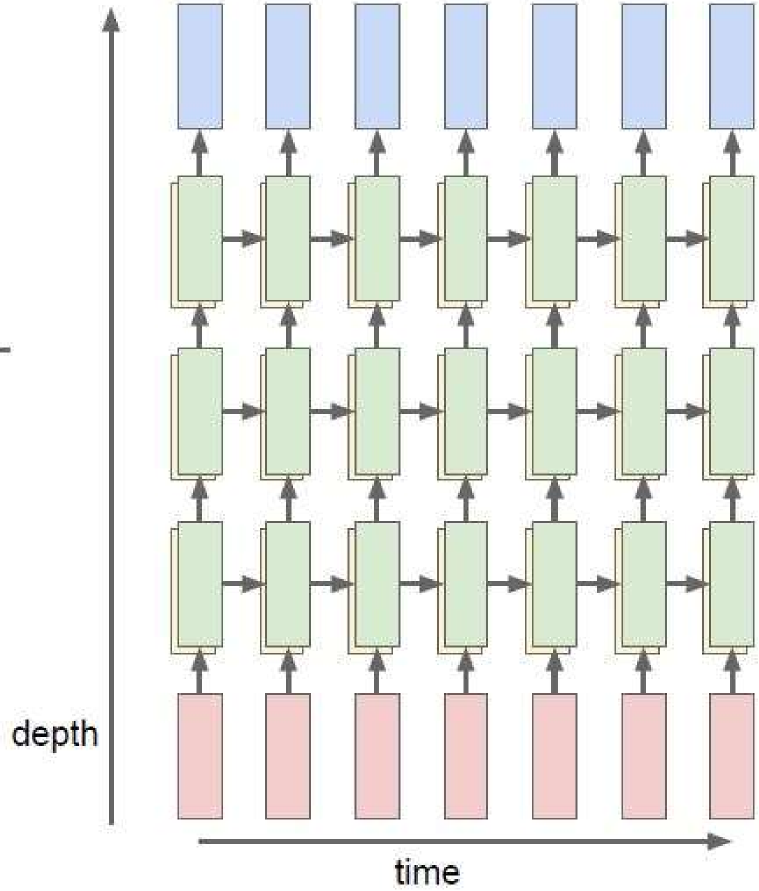
## LSTM:

$W^l \ [4n \times 2n]$

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

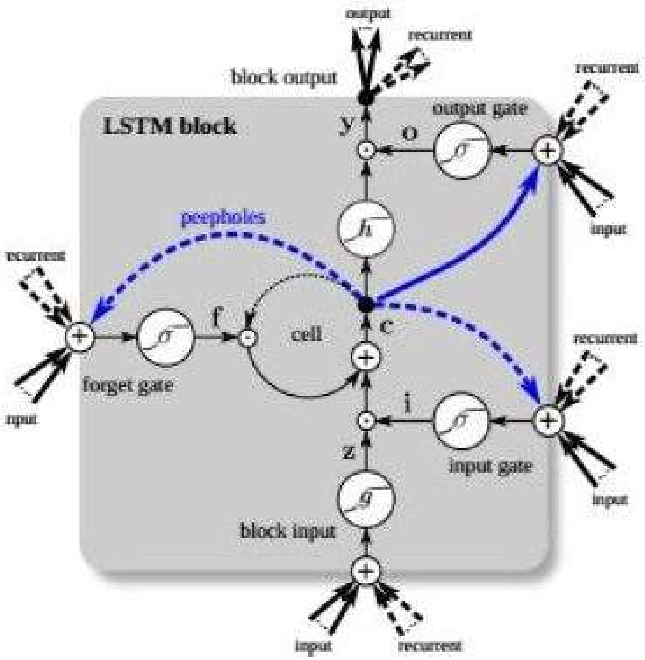
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

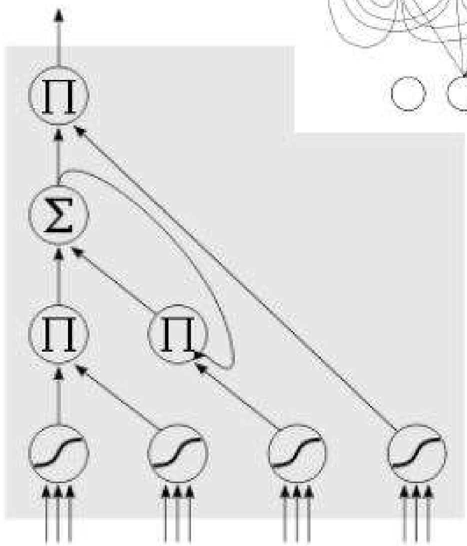
$$h_t^l = o \odot \tanh(c_t^l)$$

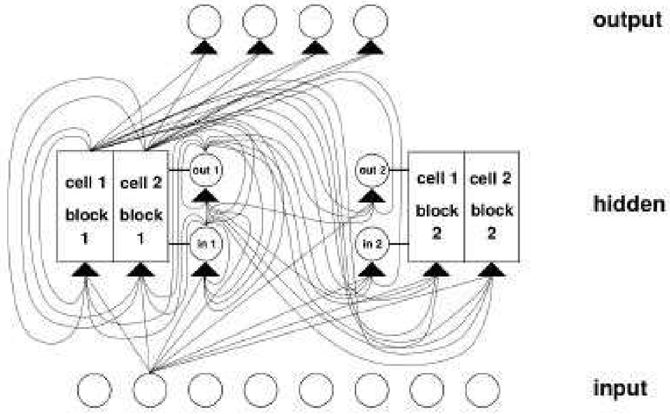


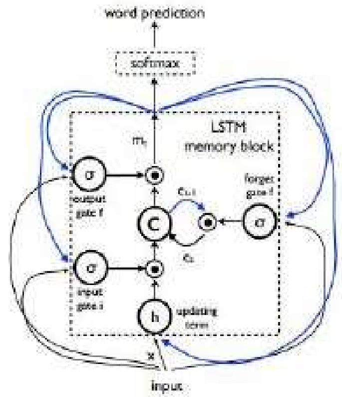

Fei-Fei Li & Andrej Karpathy & Justin Johnson

# LSTM



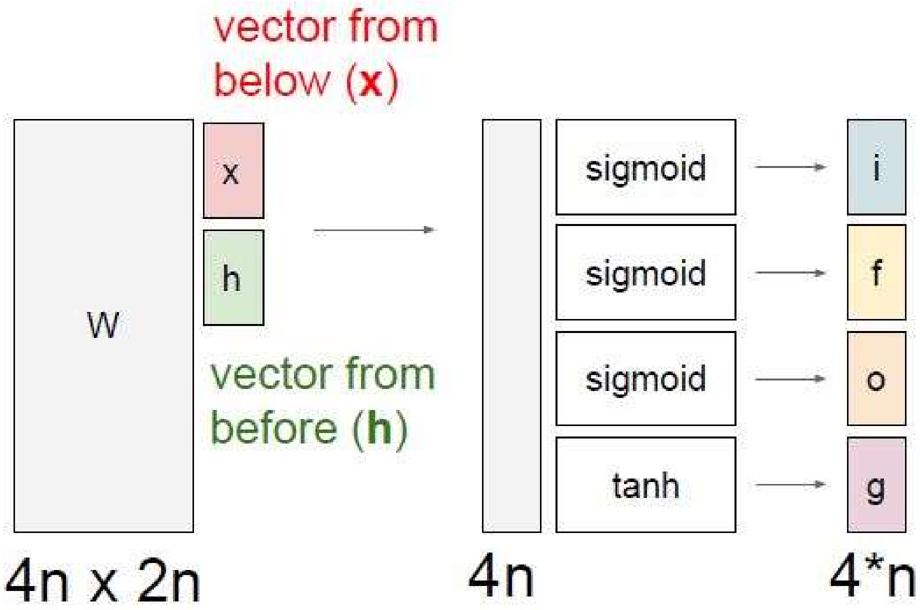



Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Long Short Term Memory (LSTM)

*[Hochreiter et al., 1997]*




$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

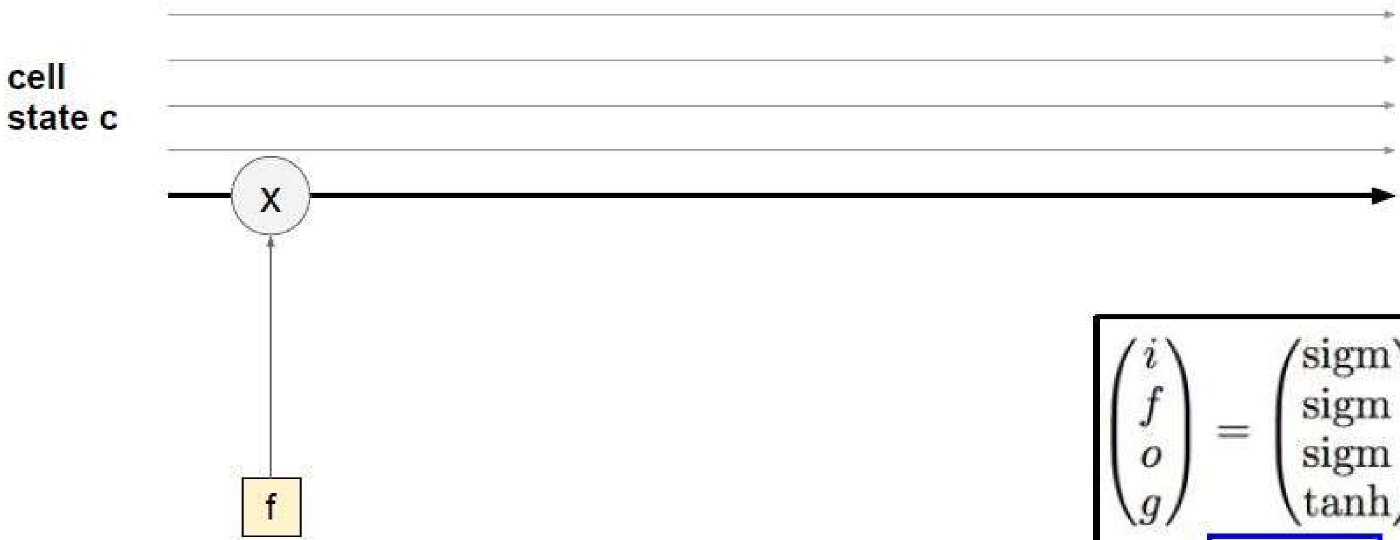
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Long Short Term Memory (LSTM)

*[Hochreiter et al., 1997]*




$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

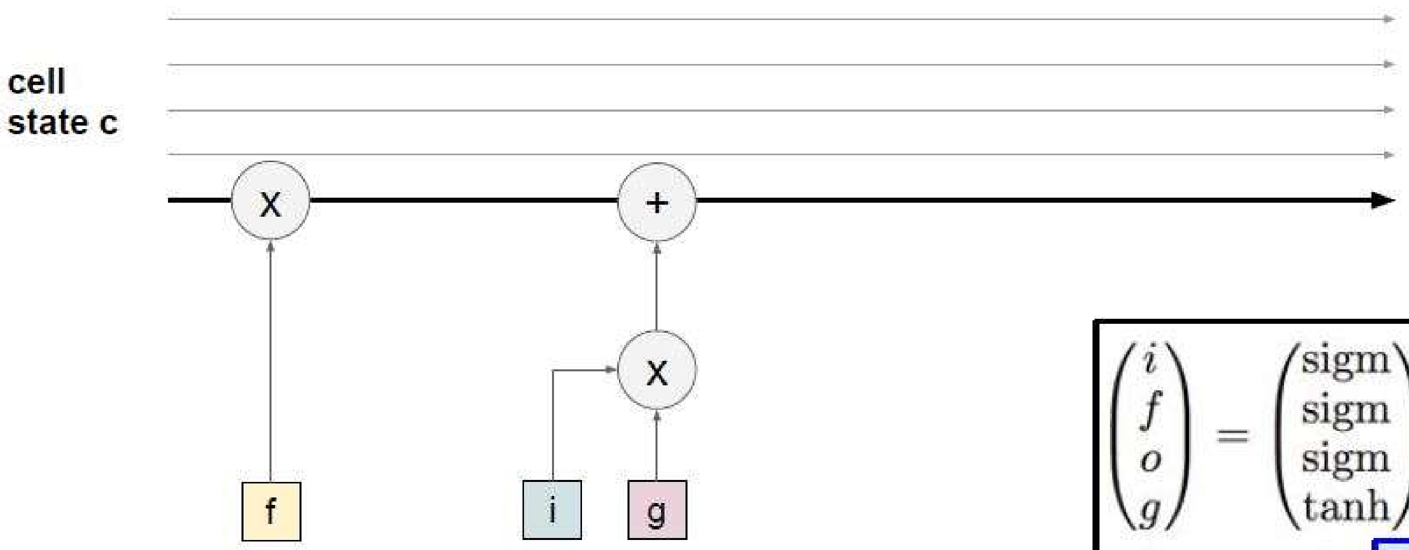
$$c_t^l = \boxed{f \odot c_{t-1}^l} + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Long Short Term Memory (LSTM)

*[Hochreiter et al., 1997]*




$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

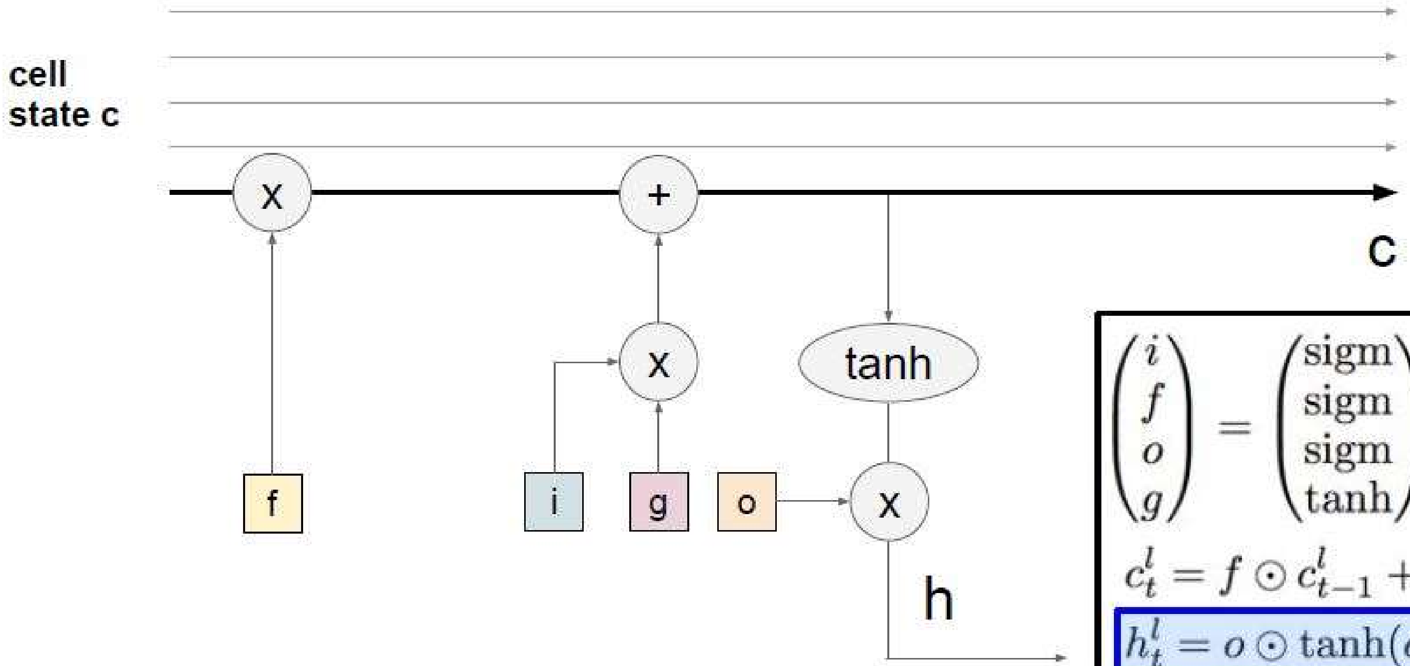
$$c_t^l = f \odot c_{t-1}^l \boxed{+ i \odot g}$$

$$h_t^l = o \odot \tanh(c_t^l)$$

Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Long Short Term Memory (LSTM)

*[Hochreiter et al., 1997]*




$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

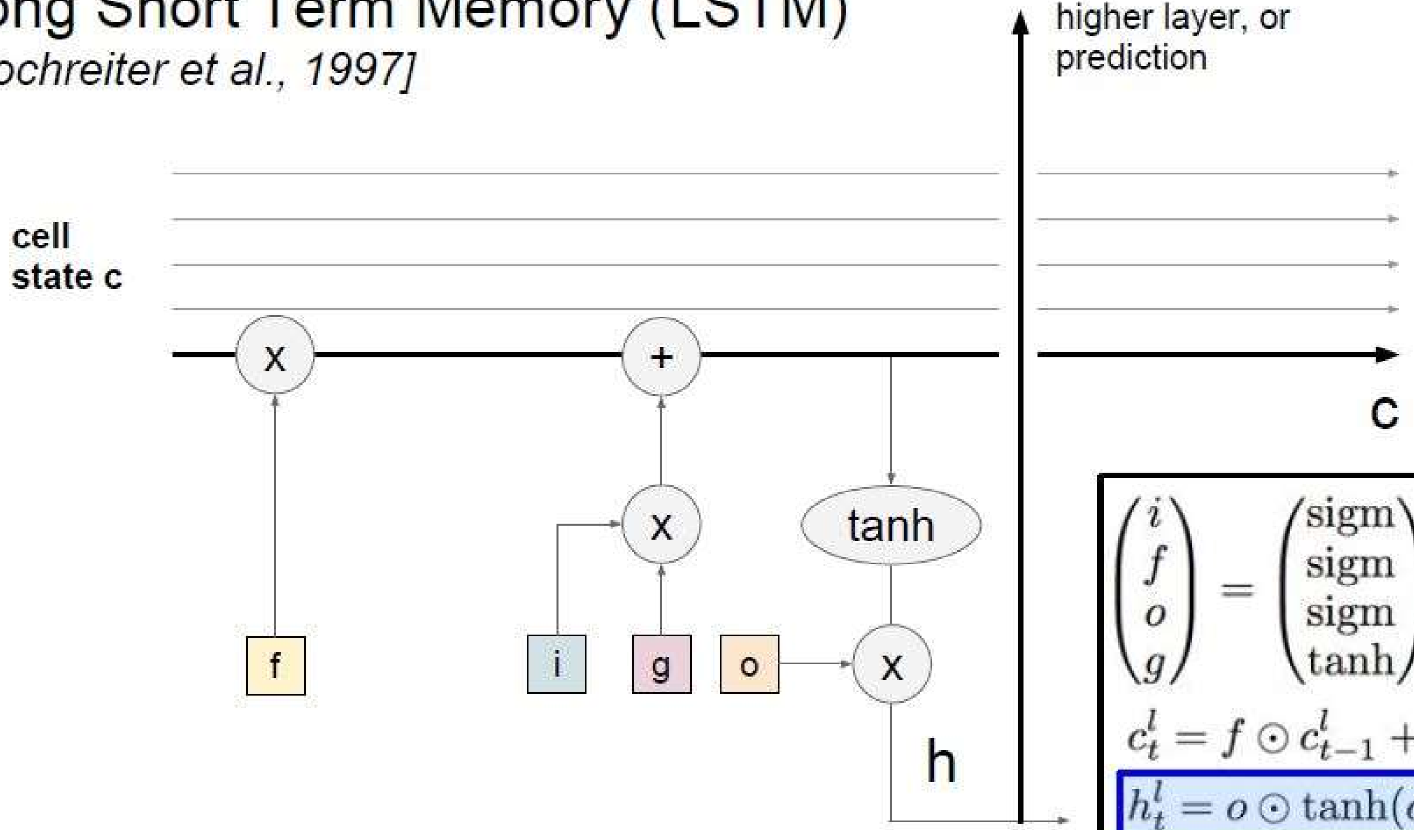
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$\boxed{h_t^l = o \odot \tanh(c_t^l)}$$

Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Long Short Term Memory (LSTM)

*[Hochreiter et al., 1997]*

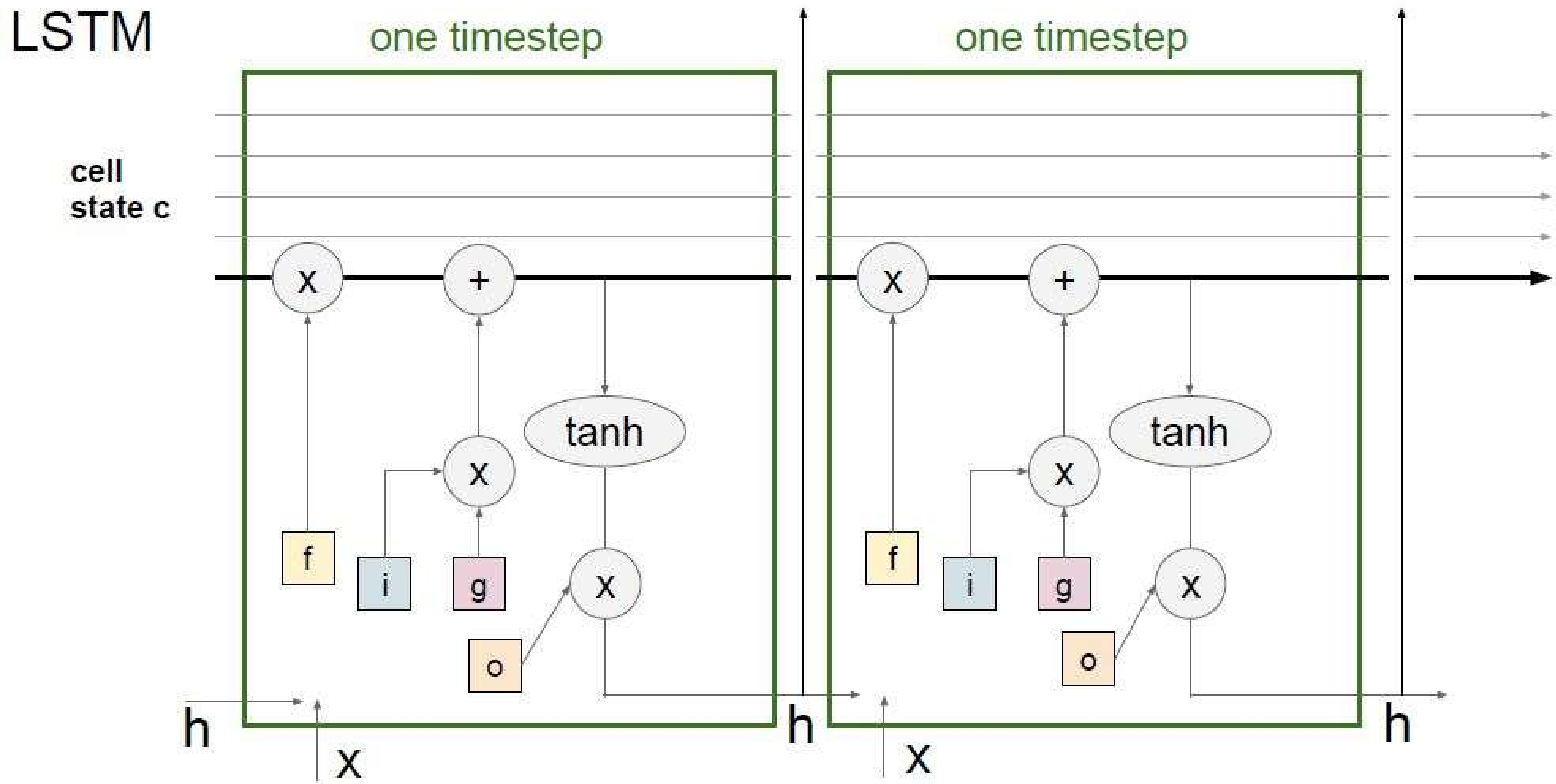



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

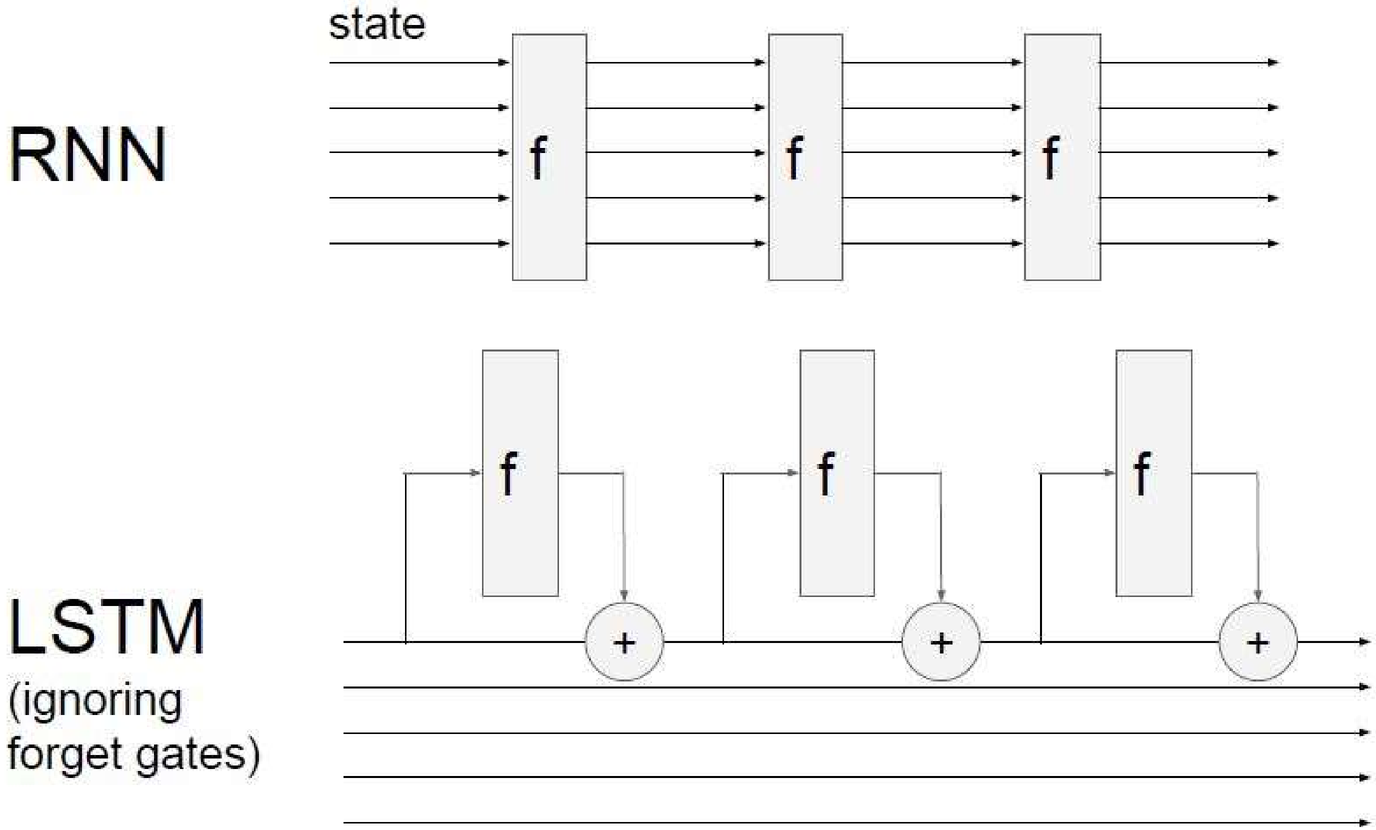
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$\boxed{h_t^l = o \odot \tanh(c_t^l)}$$

Fei-Fei Li & Andrej Karpathy & Justin Johnson




Fei-Fei Li & Andrej Karpathy & Justin Johnson

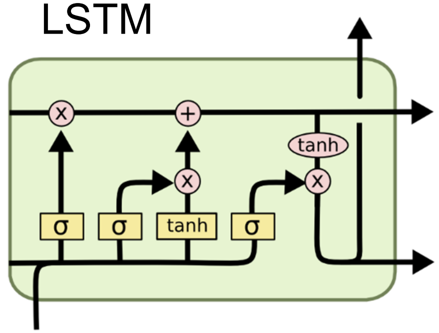


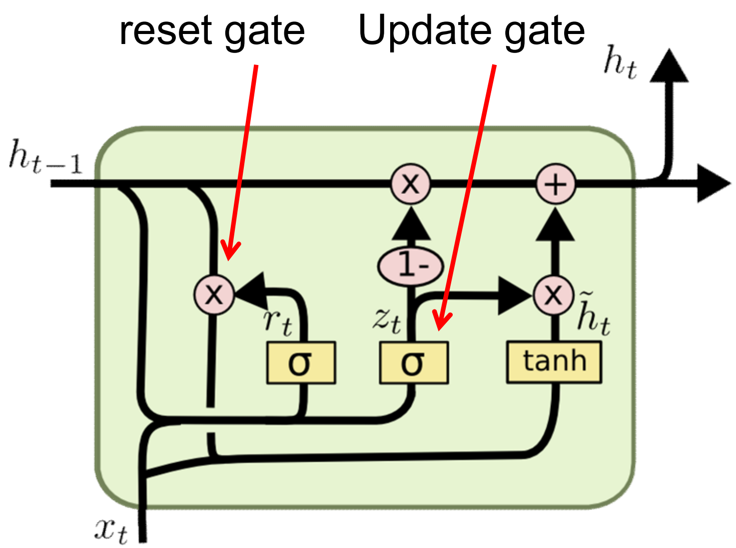

Fei-Fei Li & Andrej Karpathy & Justin Johnson

# GRU – gated recurrent uni

(more compression)

LSTM




$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

It combines the forget and input into a single update gate.
It also merges the cell state and hidden state. This is simpler than LSTM. There are many other variants too.

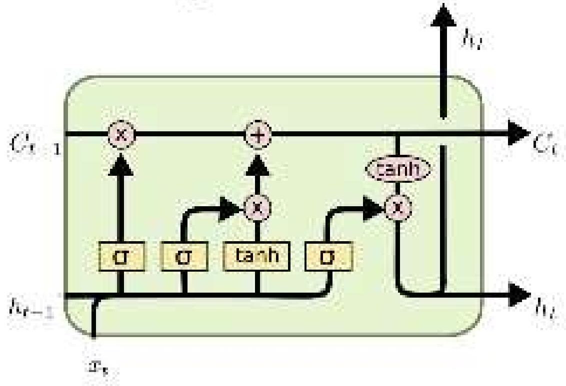
https://cs.uwaterloo.ca/~mli



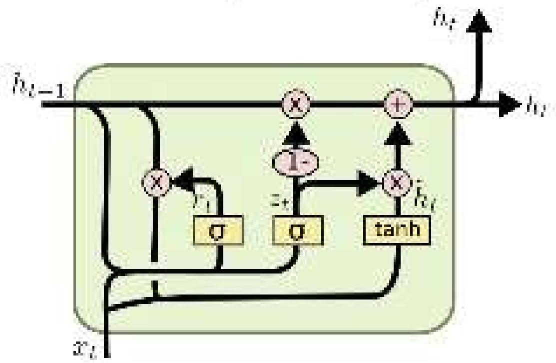
X *: element-wise multiply

# LSTM and GRU

- LSTM [Hochreiter&Schmidhuber97]



- GRU [Cho+14]



GRUs also takes $x_t$ and $h_{t-1}$ as inputs. They perform some calculations and then pass along $h_t$. What makes them different from LSTMs is that GRUs don't need the cell layer to pass values along. The calculations within each iteration insure that the $h_t$ values being passed along either retain a high amount of old information or are jump-started with a high amount of new information.

https://cs.uwaterloo.ca/~mli

# Summary

- RNNs allow a lot of flexibility in architecture design
- Vanilla RNNs are simple but don't work very well
- Common to use LSTM or GRU: their additive interactions improve gradient flow
- Backward flow of gradients in RNN can explode or vanish. Exploding is controlled with gradient clipping. Vanishing is controlled with additive interactions (LSTM)
- Better/simpler architectures are a hot topic of current research
- Better understanding (both theoretical and empirical) is needed.

Fei-Fei Li & Andrej Karpathy & Justin Johnson

# END