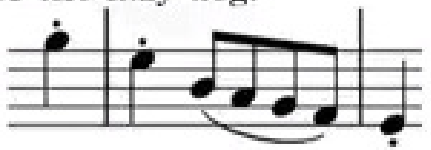
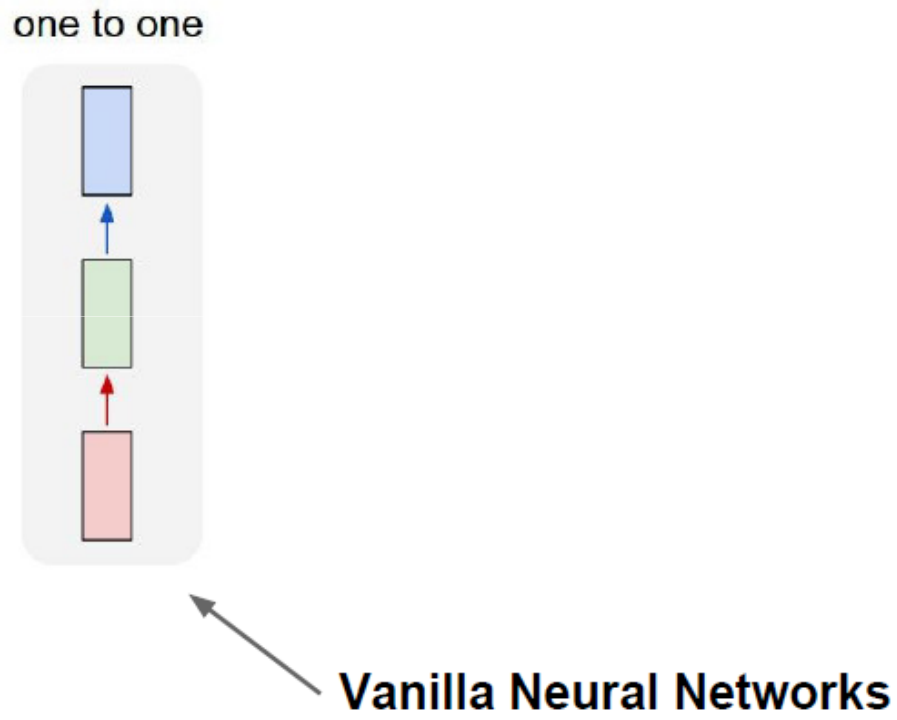
RNN

Sequence Learning

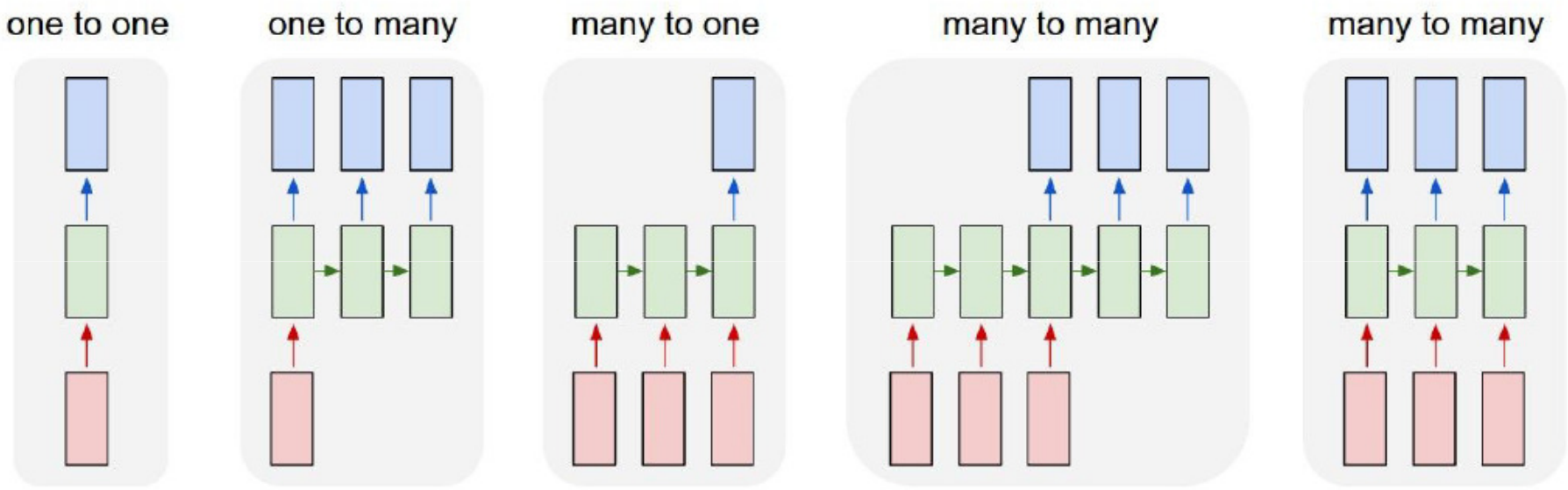
Speech recognition		→	"The quick brown fox jumped over the lazy dog."
Music generation	∅	→	
Sentiment classification	"There is nothing to like in this movie."	→	★☆☆☆☆
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	→	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	→	Do you want to sing with me?
Video activity recognition		→	Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	→	Yesterday, Harry Potter met Hermione Granger.

Vanilla Neural Network



Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

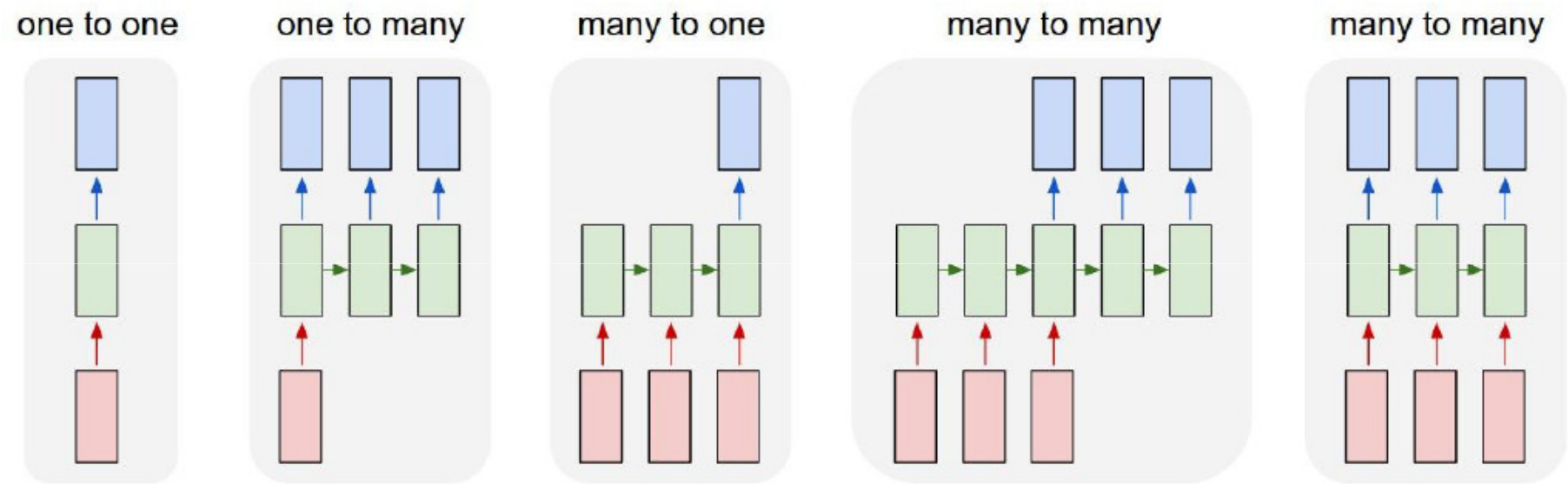
Recurrent Neural Networks: Process Sequences



e.g. **Image Captioning**
image -> sequence of words

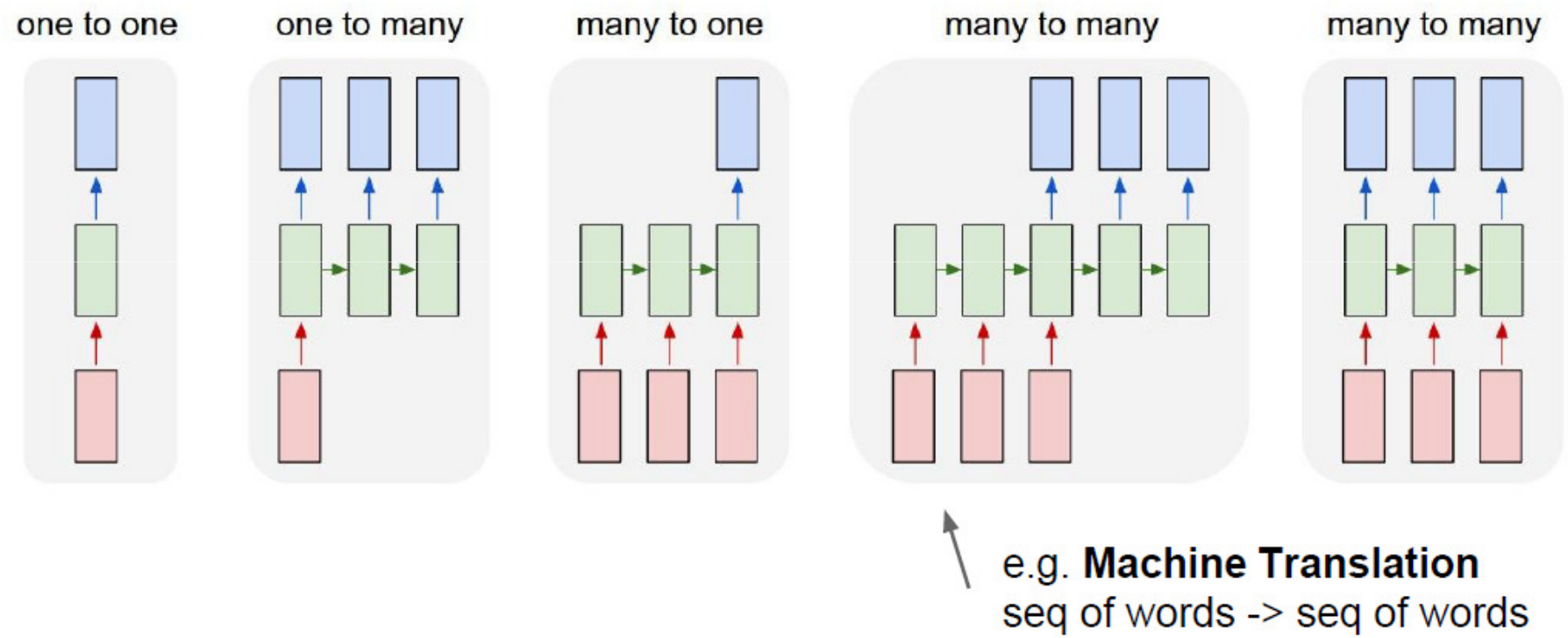
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Recurrent Neural Networks: Process Sequences



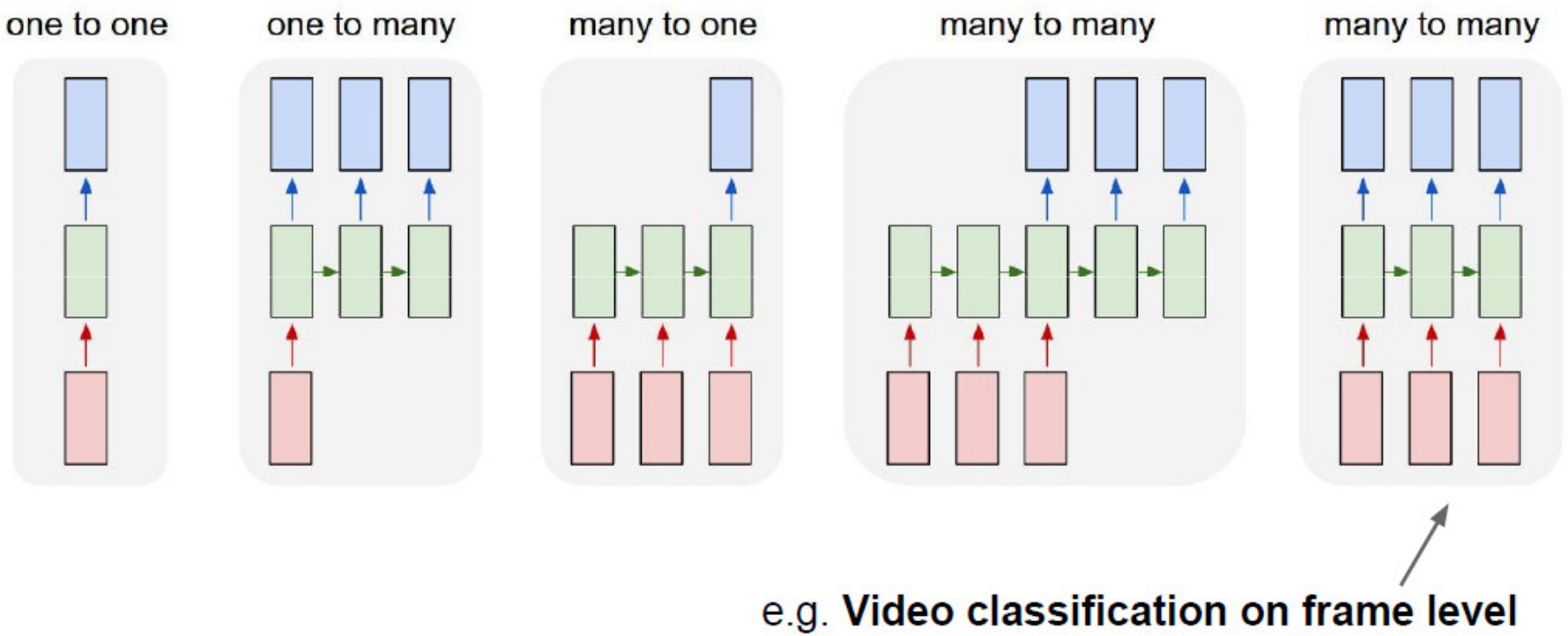
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Recurrent Neural Networks: Process Sequences



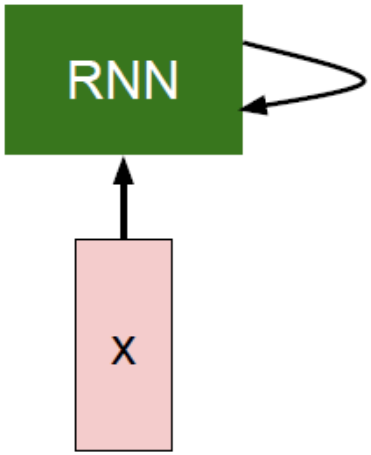
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Recurrent Neural Networks: Process Sequences



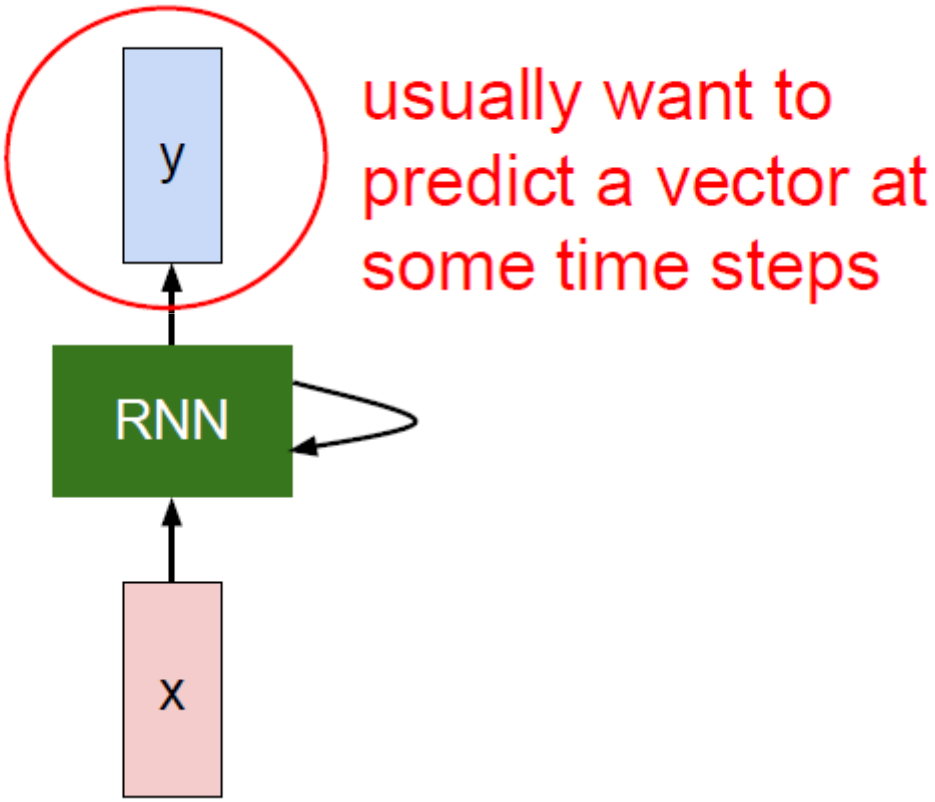
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Recurrent Neural Network



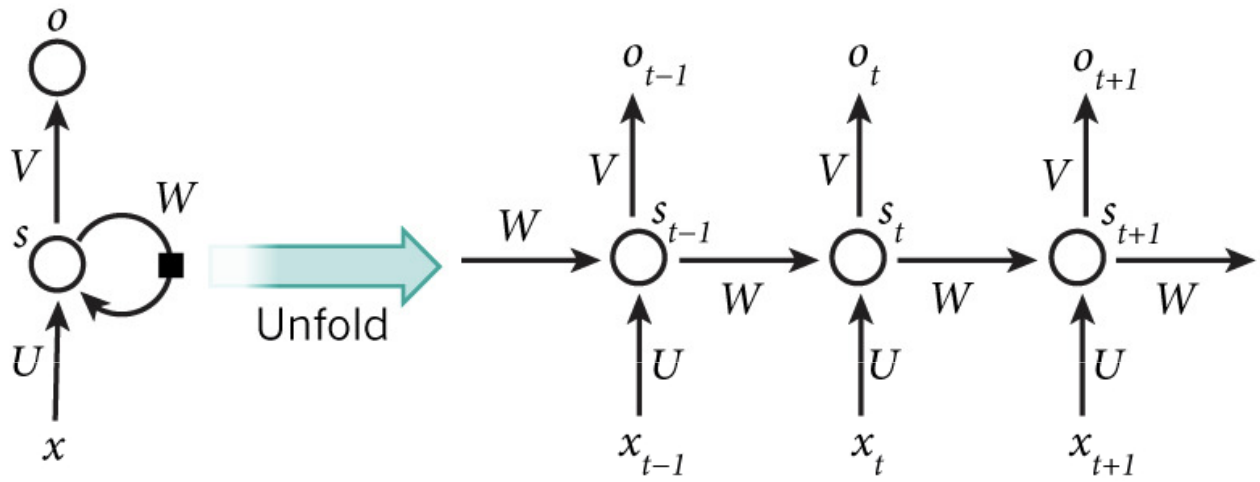
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

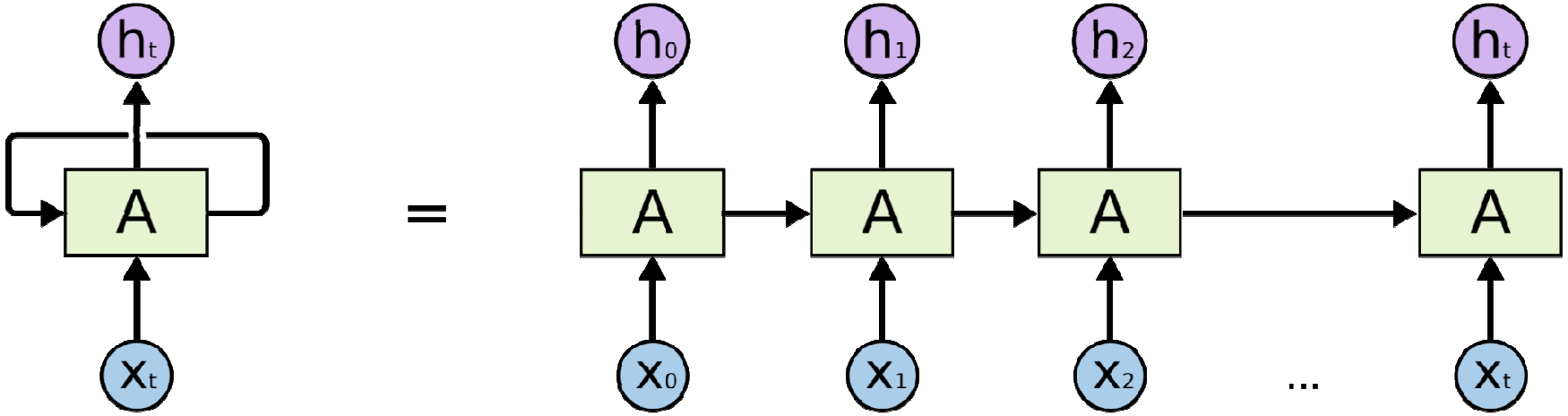
Recurrent Neural Network



Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Recurrent Neural Network





Recurrent Neural Network

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

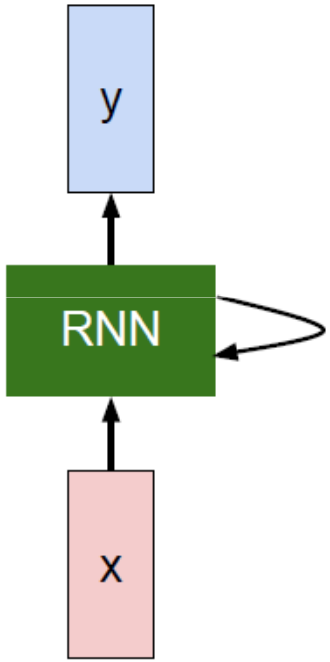
$$h_t = f_W(h_{t-1}, x_t)$$

new state

old state

input vector at some time step

some function with parameters W

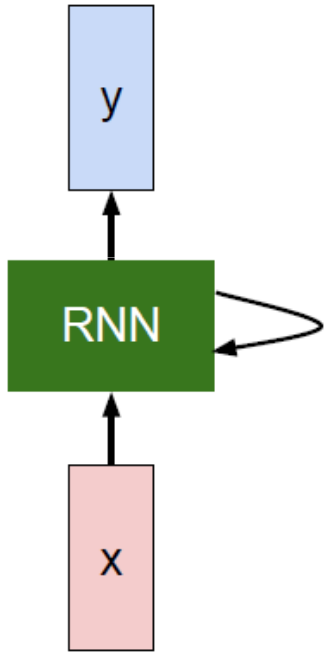


Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Recurrent Neural Network

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

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

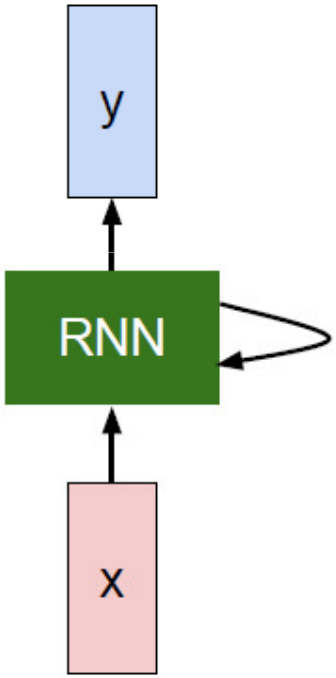


Notice: the same function and the same set of parameters are used at every time step.

Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Recurrent Neural Network

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



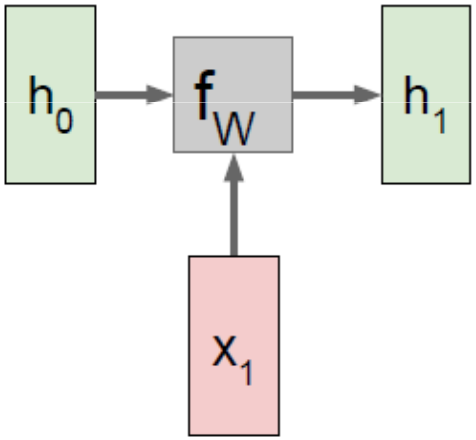
$$h_t = f_W(h_{t-1}, x_t)$$

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

$$y_t = W_{hy}h_t$$

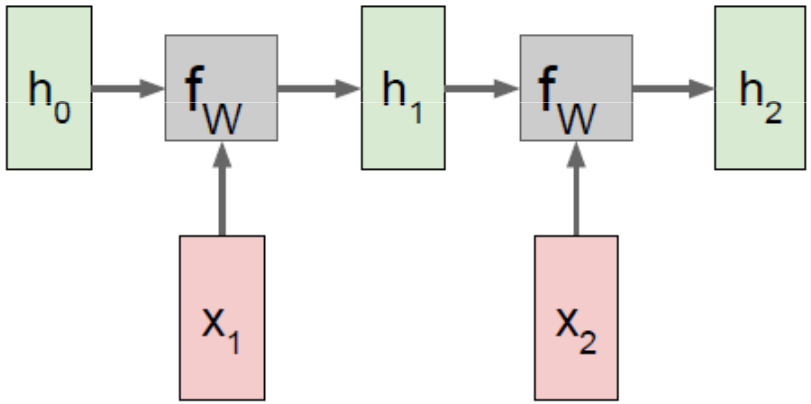
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

RNN: Computational Graph



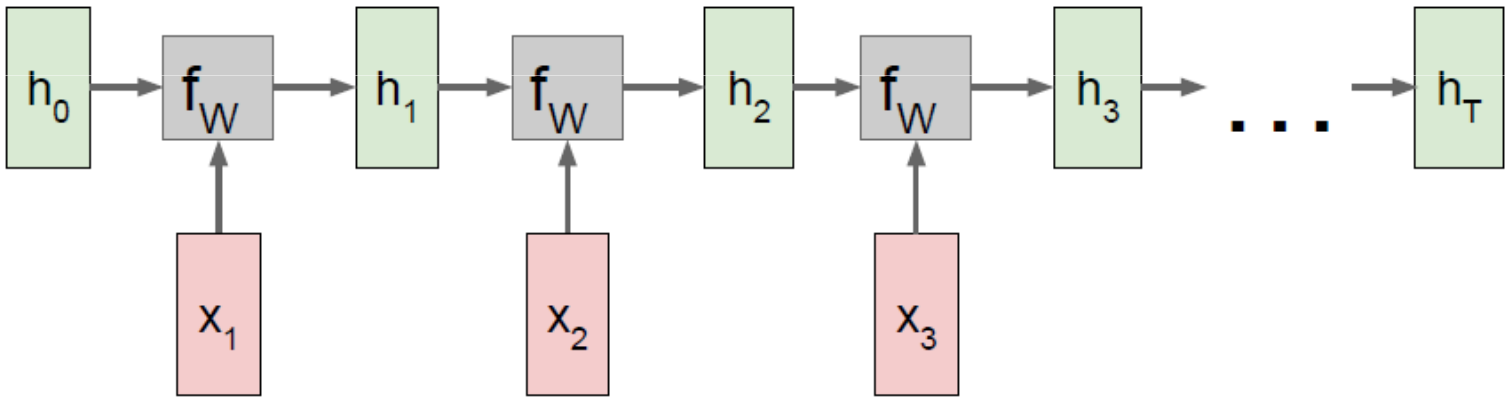
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

RNN: Computational Graph



Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

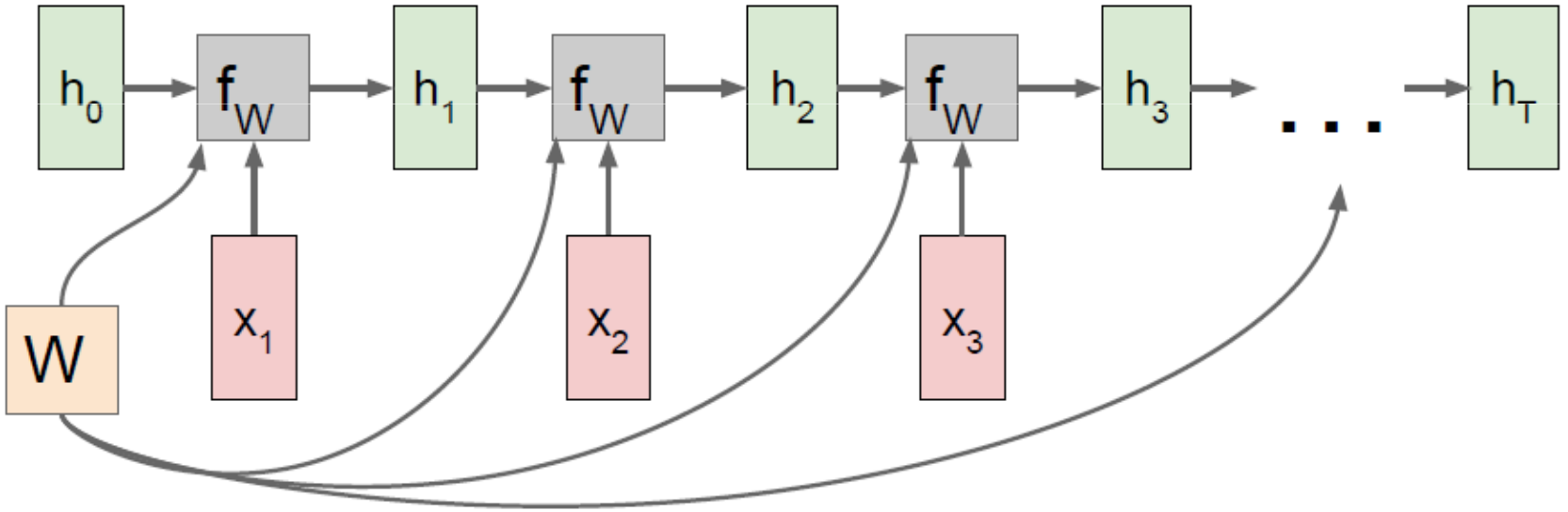
RNN: Computational Graph



Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

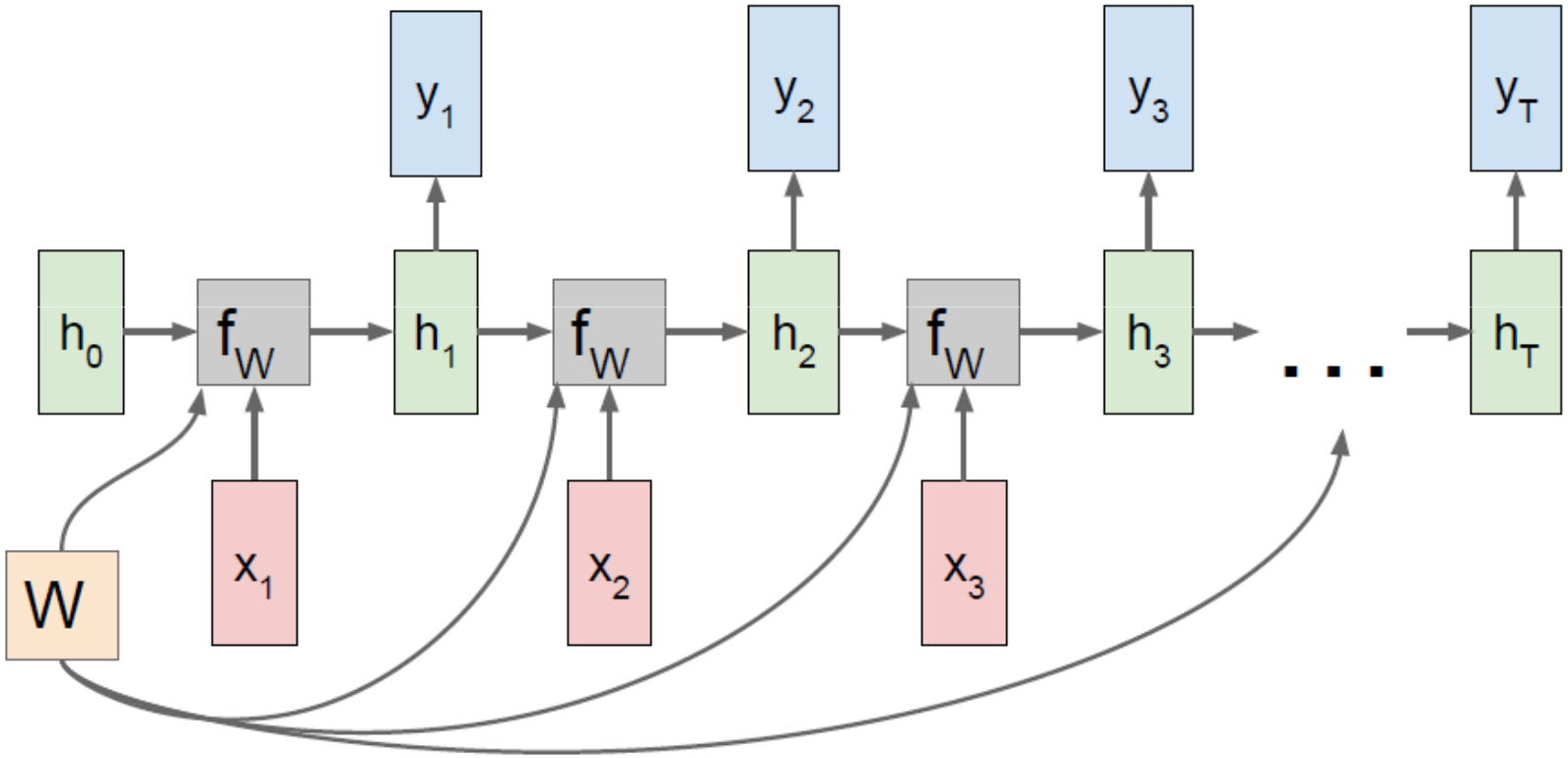
RNN: Computational Graph

Re-use the same weight matrix at every time-step



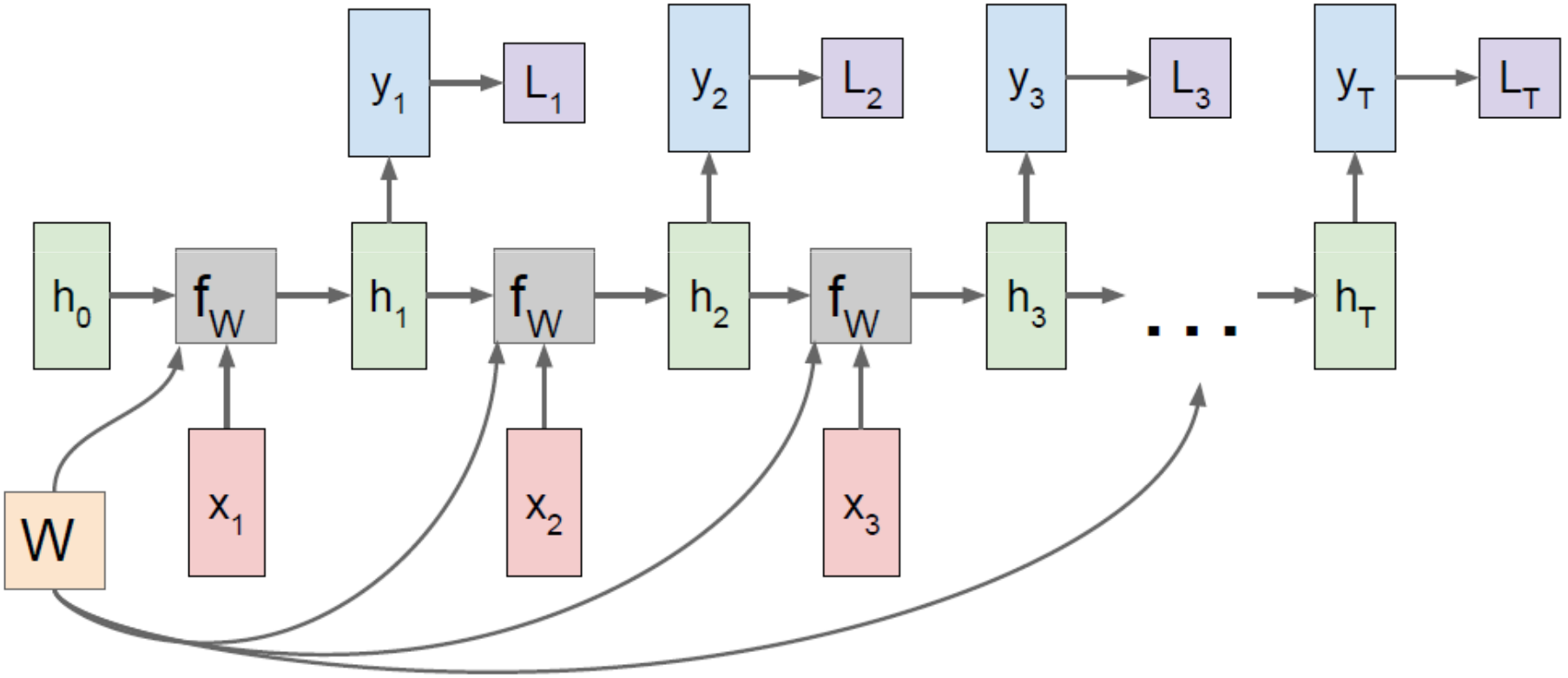
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

RNN: Computational Graph: Many to Many



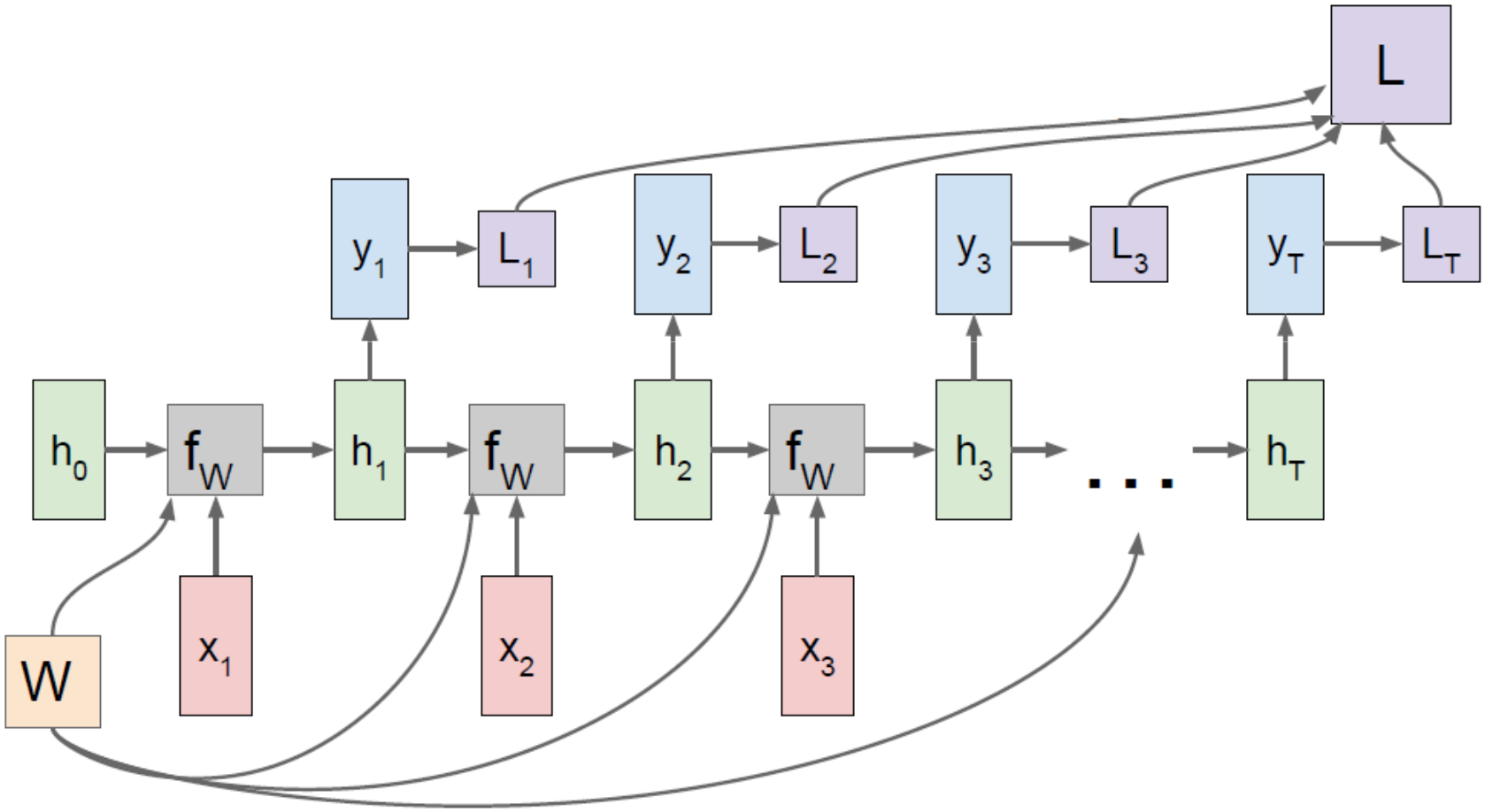
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

RNN: Computational Graph: Many to Many



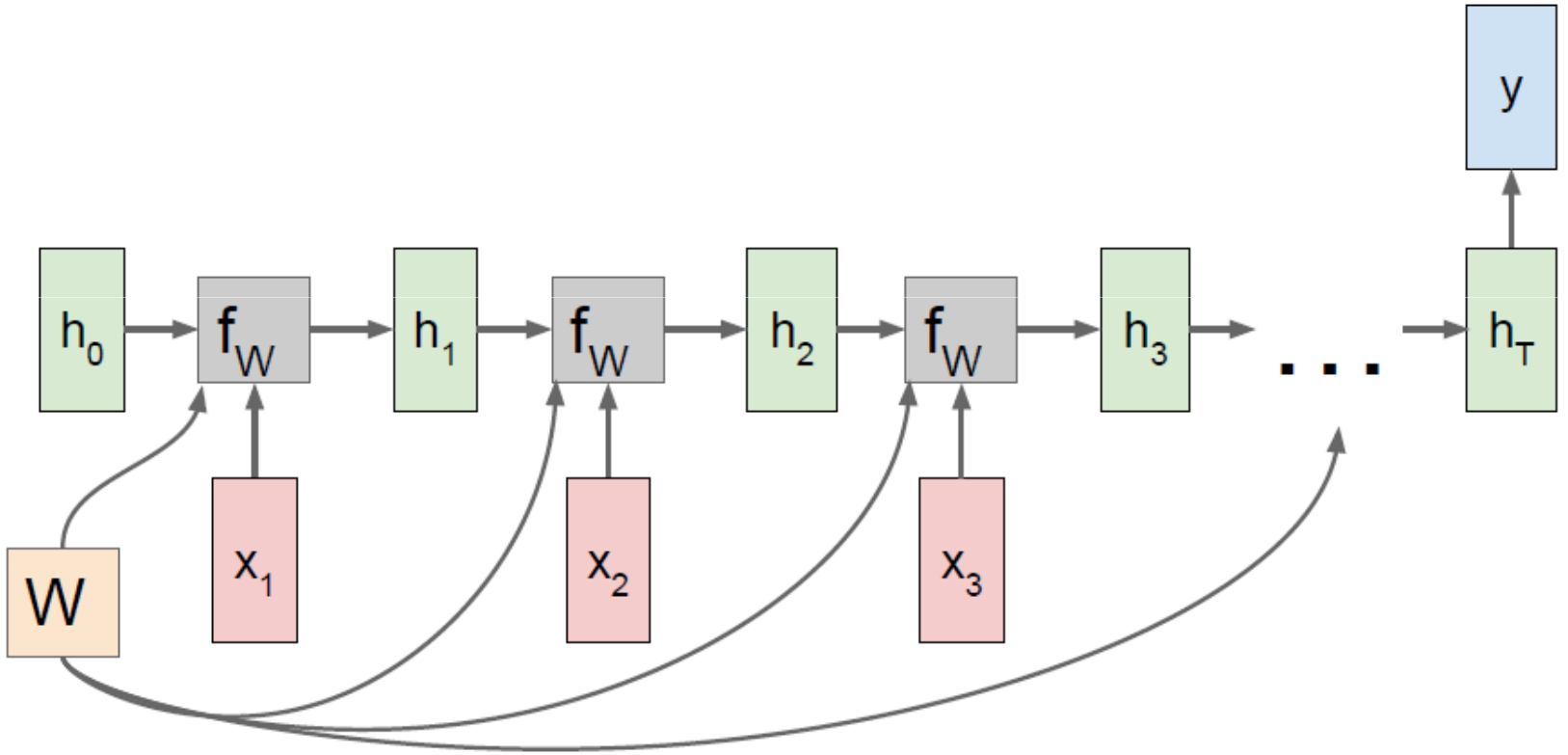
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

RNN: Computational Graph: Many to Many



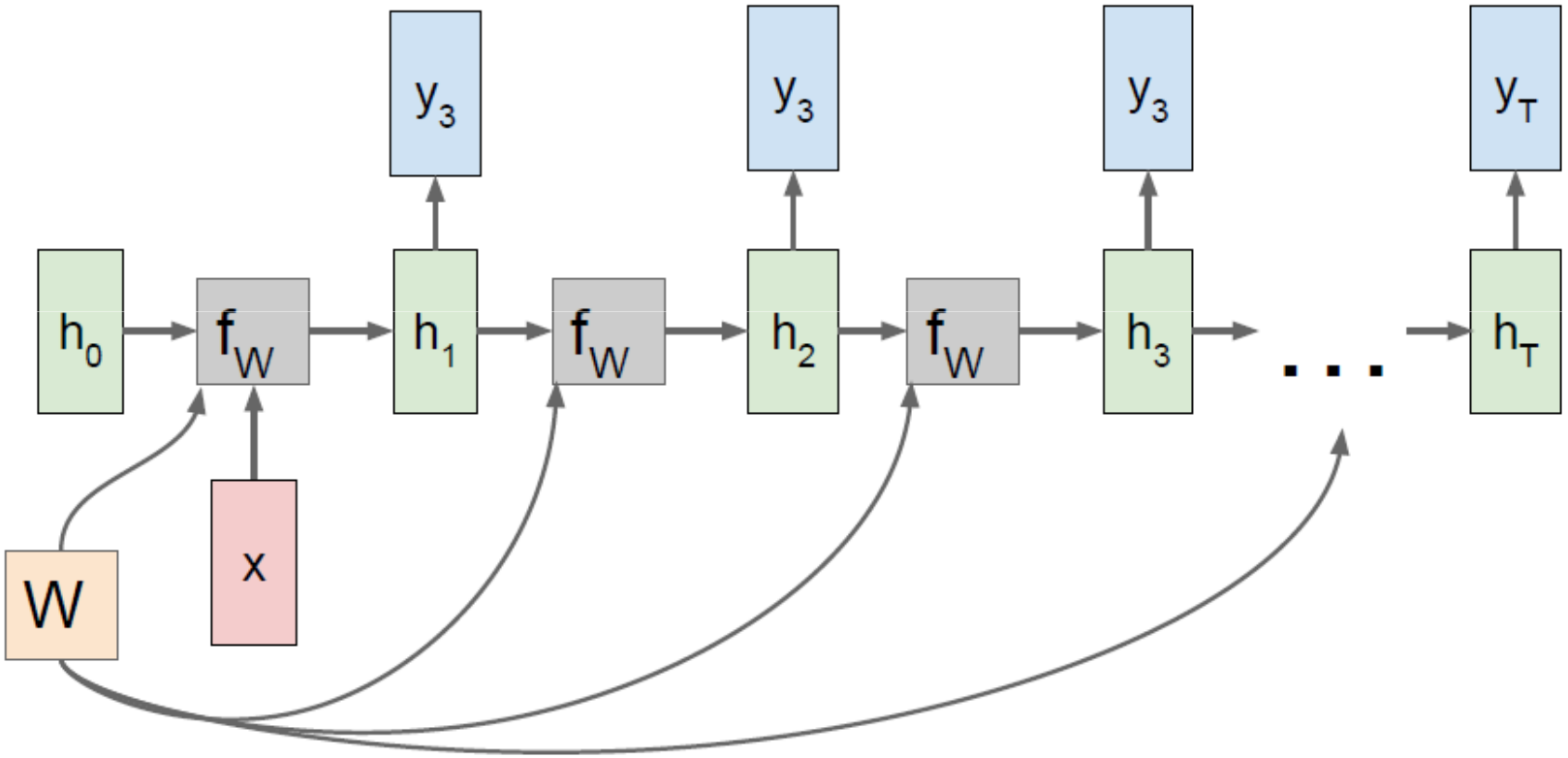
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

RNN: Computational Graph: Many to One



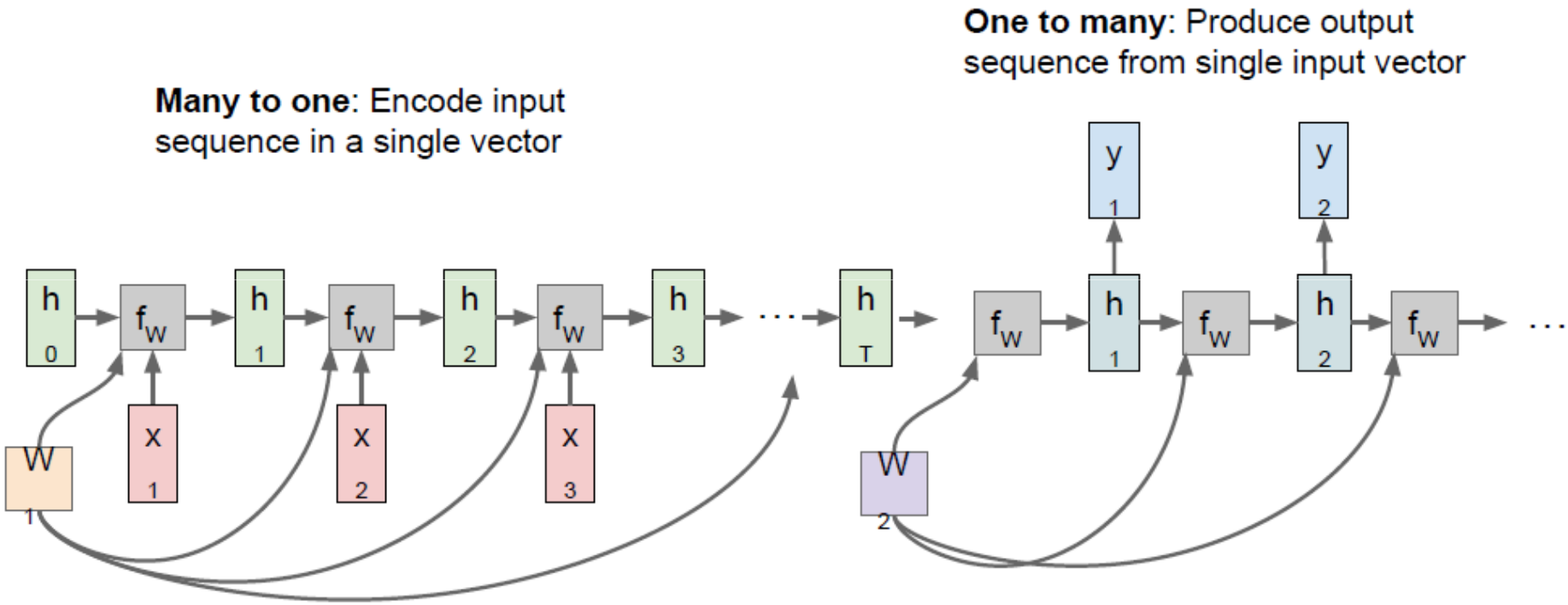
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

RNN: Computational Graph: One to Many



Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

RNN: Computational Graph: Many-to-one + one-to-many

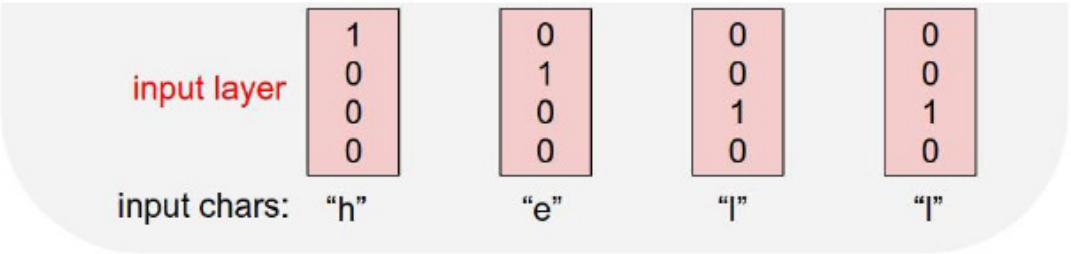


Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”



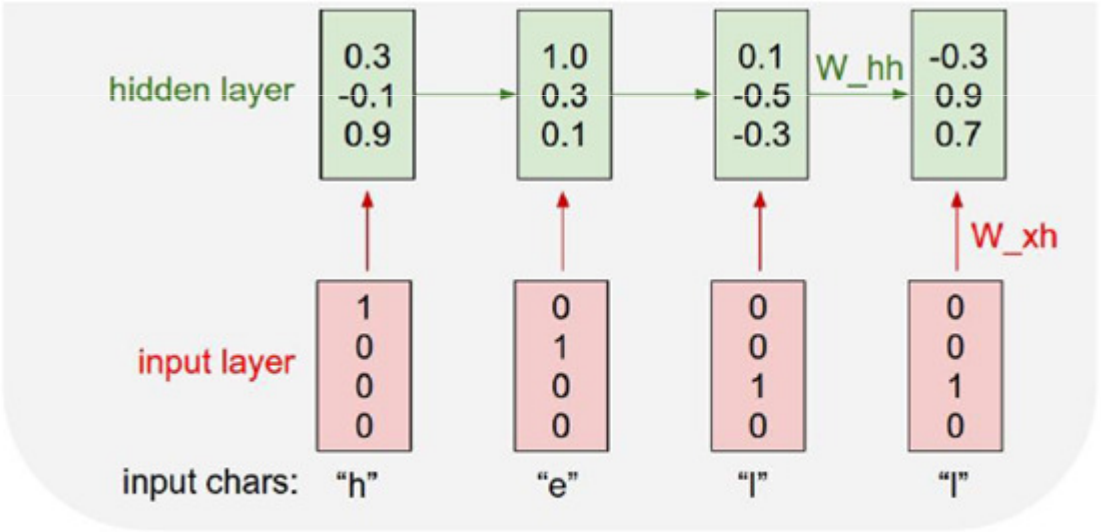
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Example: Character-level Language Model

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

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

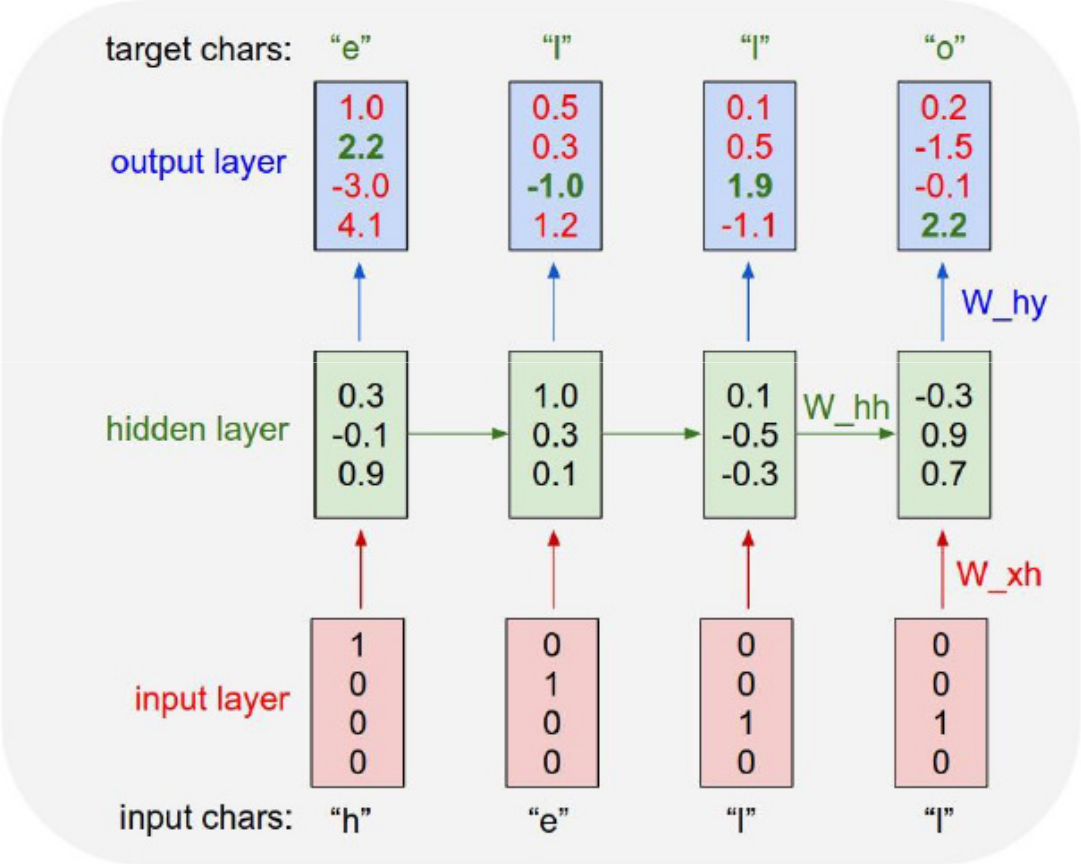


Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

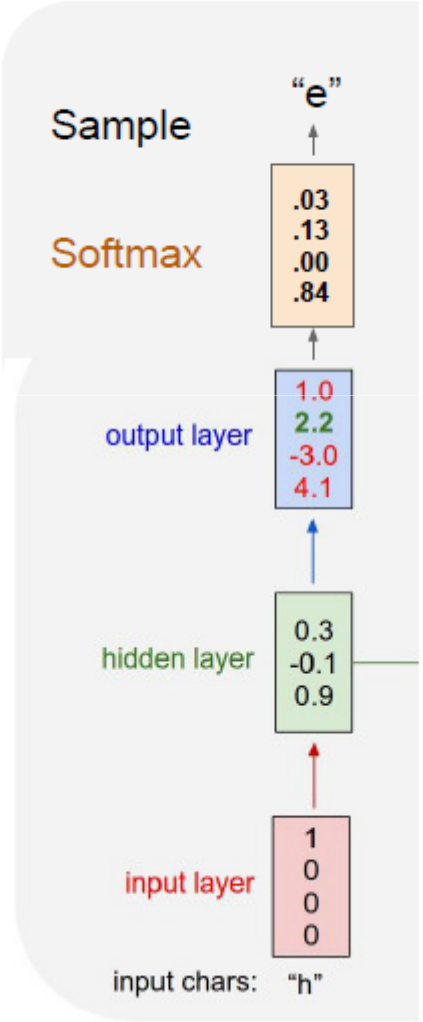


Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model

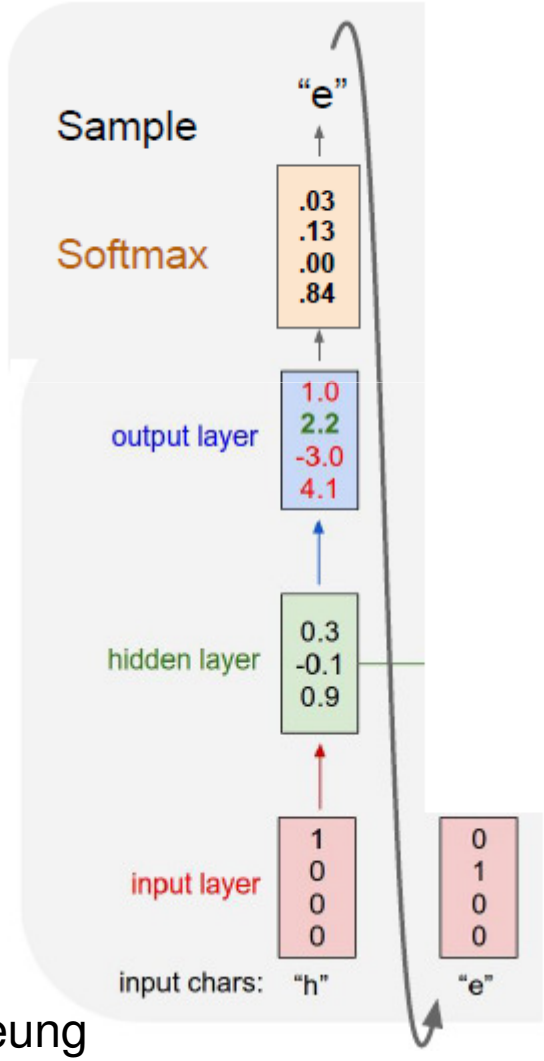


Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model

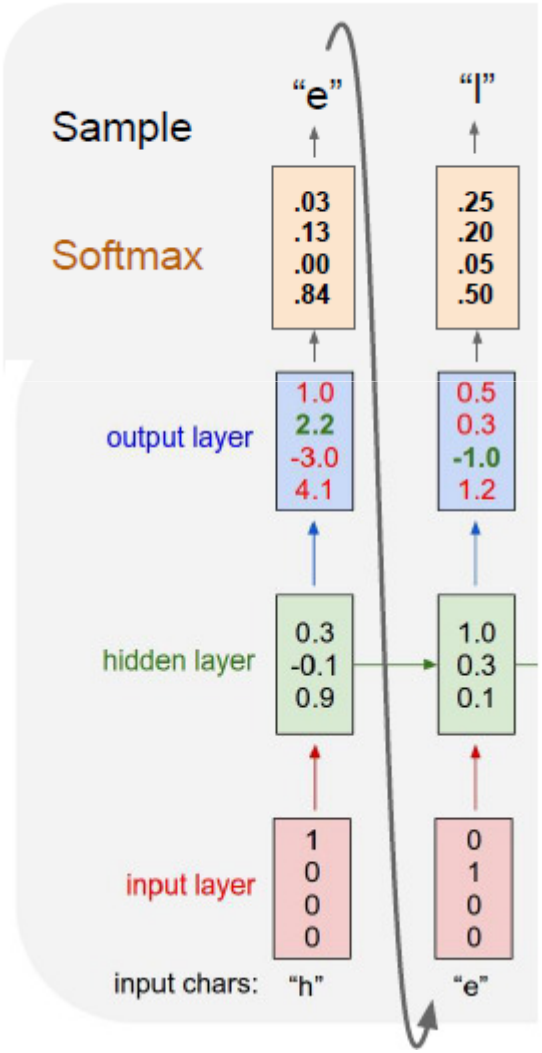


Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model

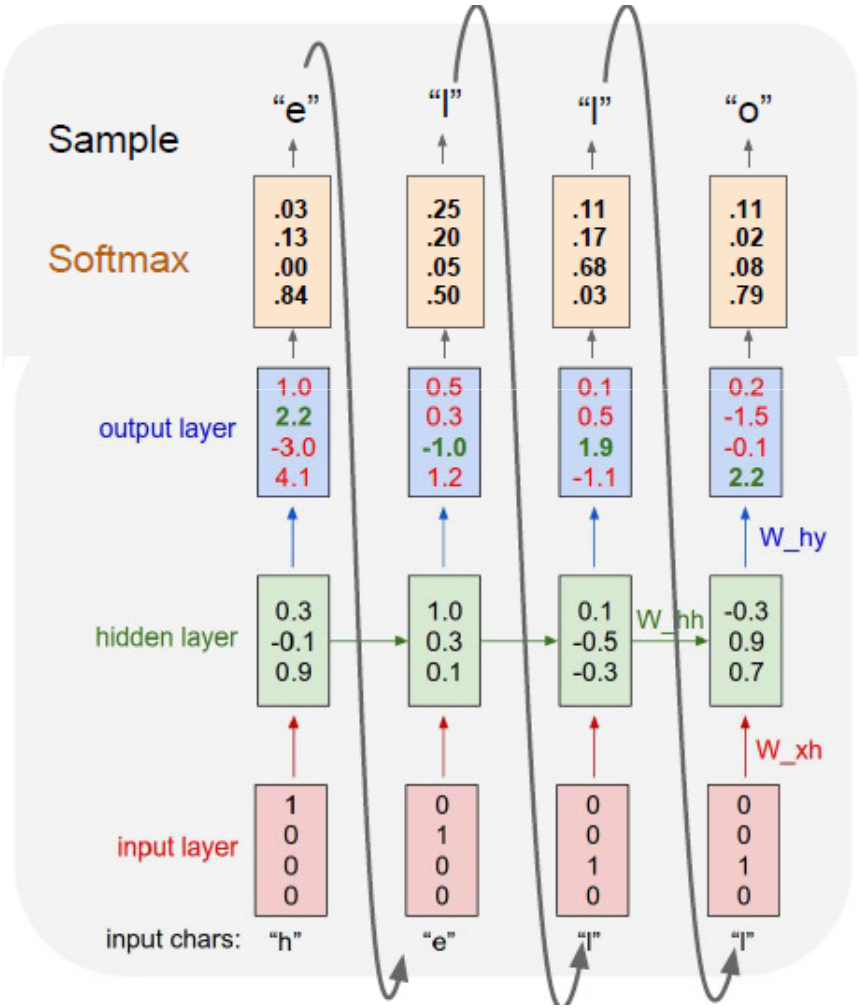


Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Example: Character-level Language Model

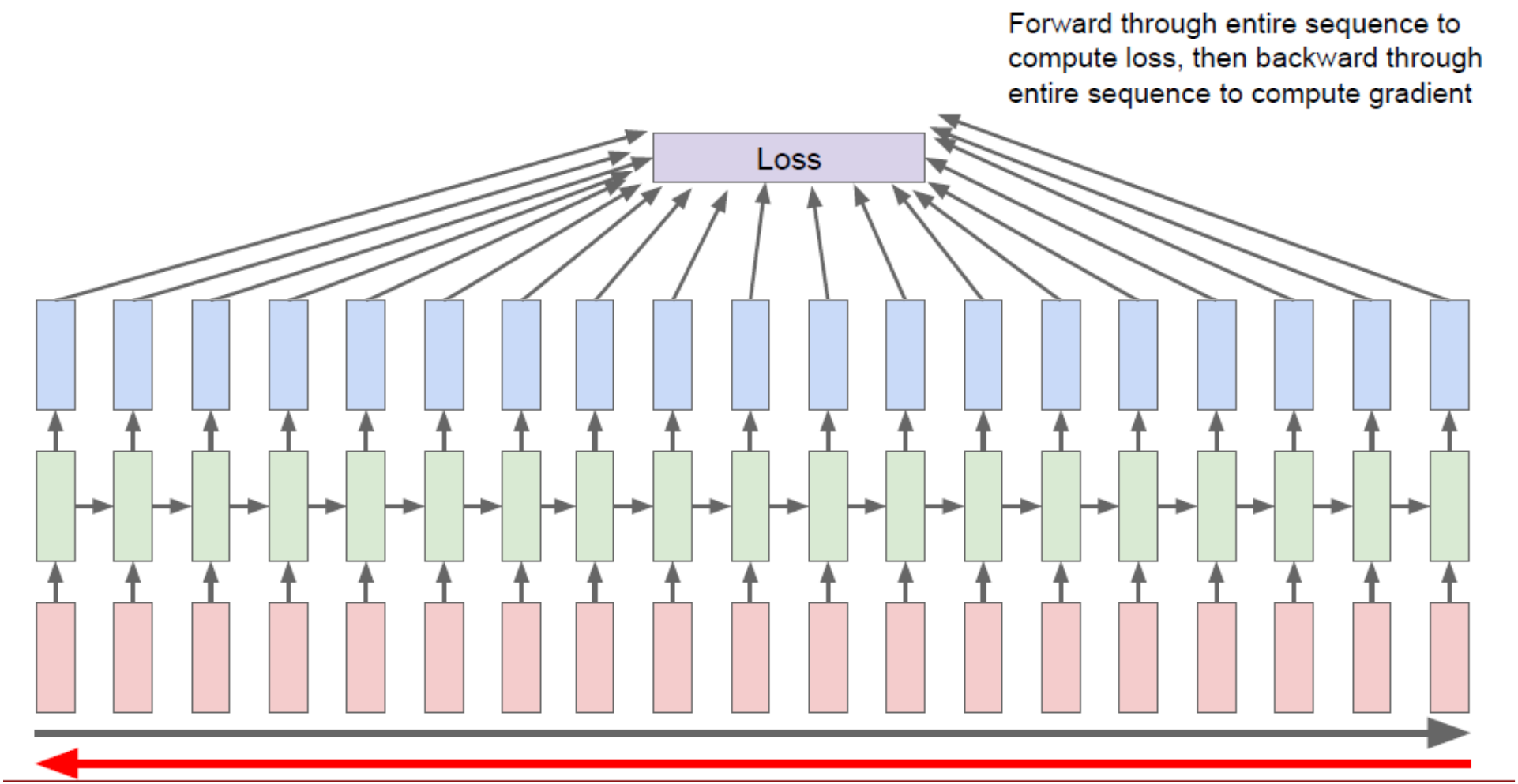
Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model



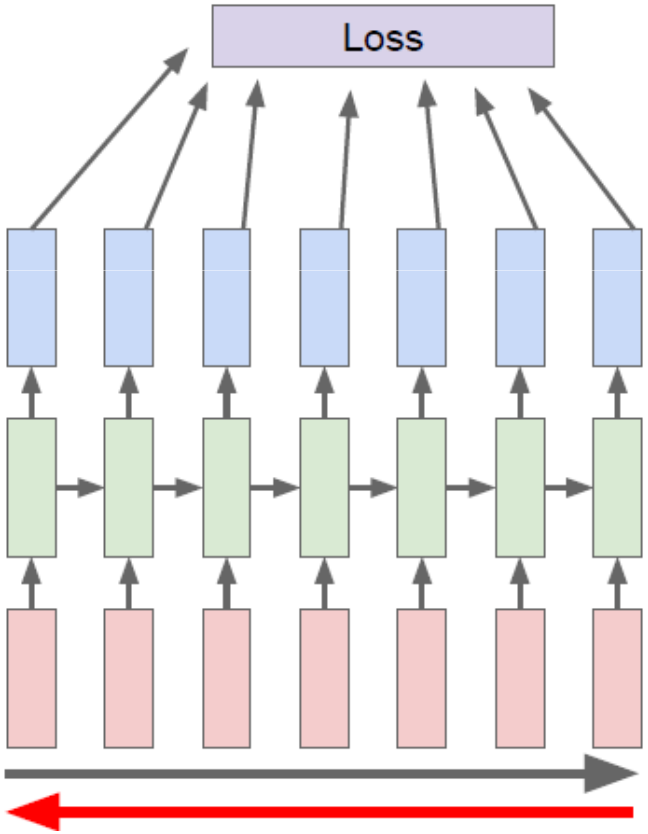
Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Backpropagation through time



Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

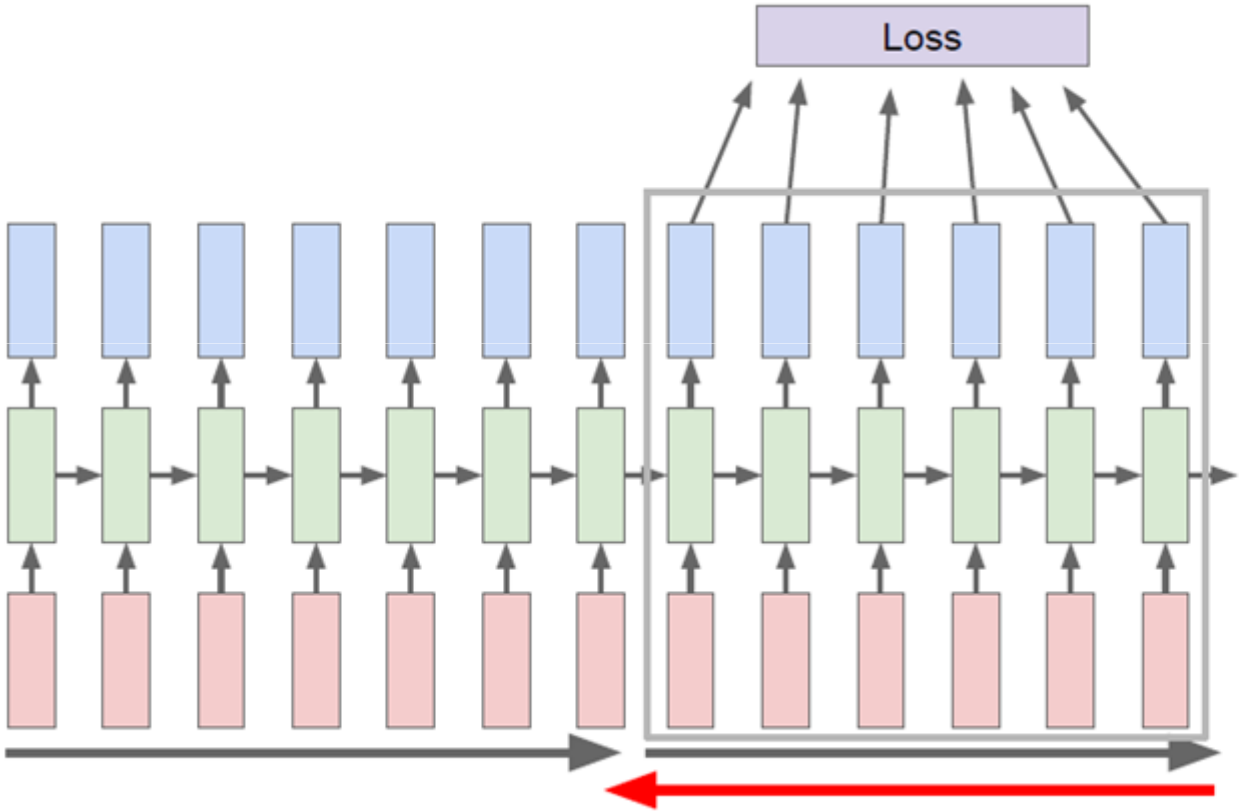
Truncated Backpropagation through time



Run forward and backward through chunks of the sequence instead of whole sequence

Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

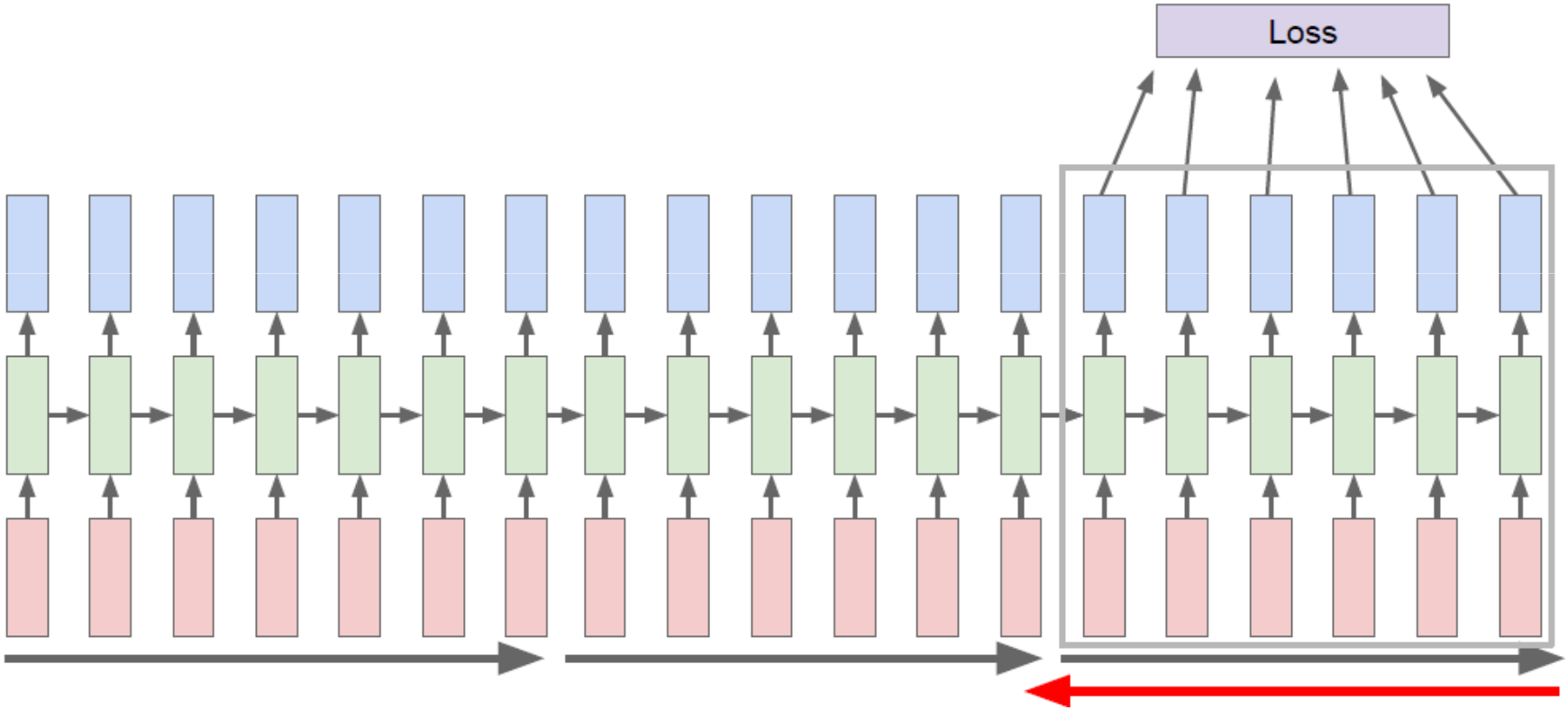
Truncated Backpropagation through time



Carry hidden states forward in time forever, but only backpropagate for some smaller number of steps

Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Truncated Backpropagation through time

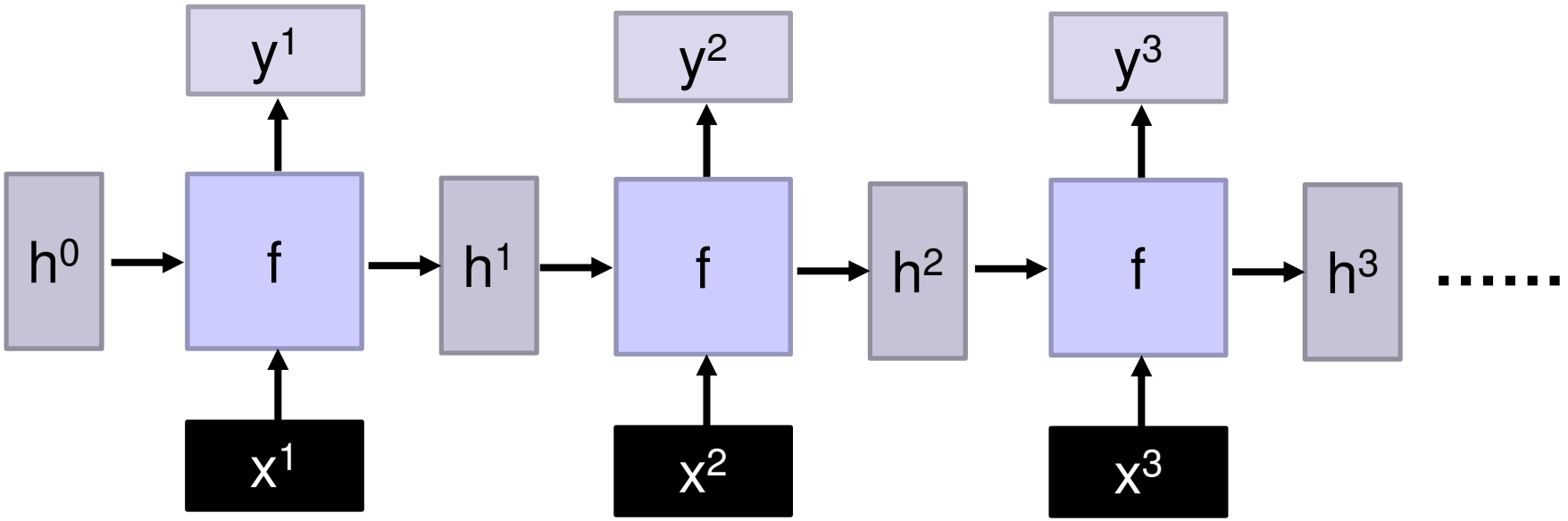


Slide Credit: Fei-Fei Li & Justin Johnson & Serena Yeung

How does RNN reduce complexity?

- Given function f: h',y=f(h,x)

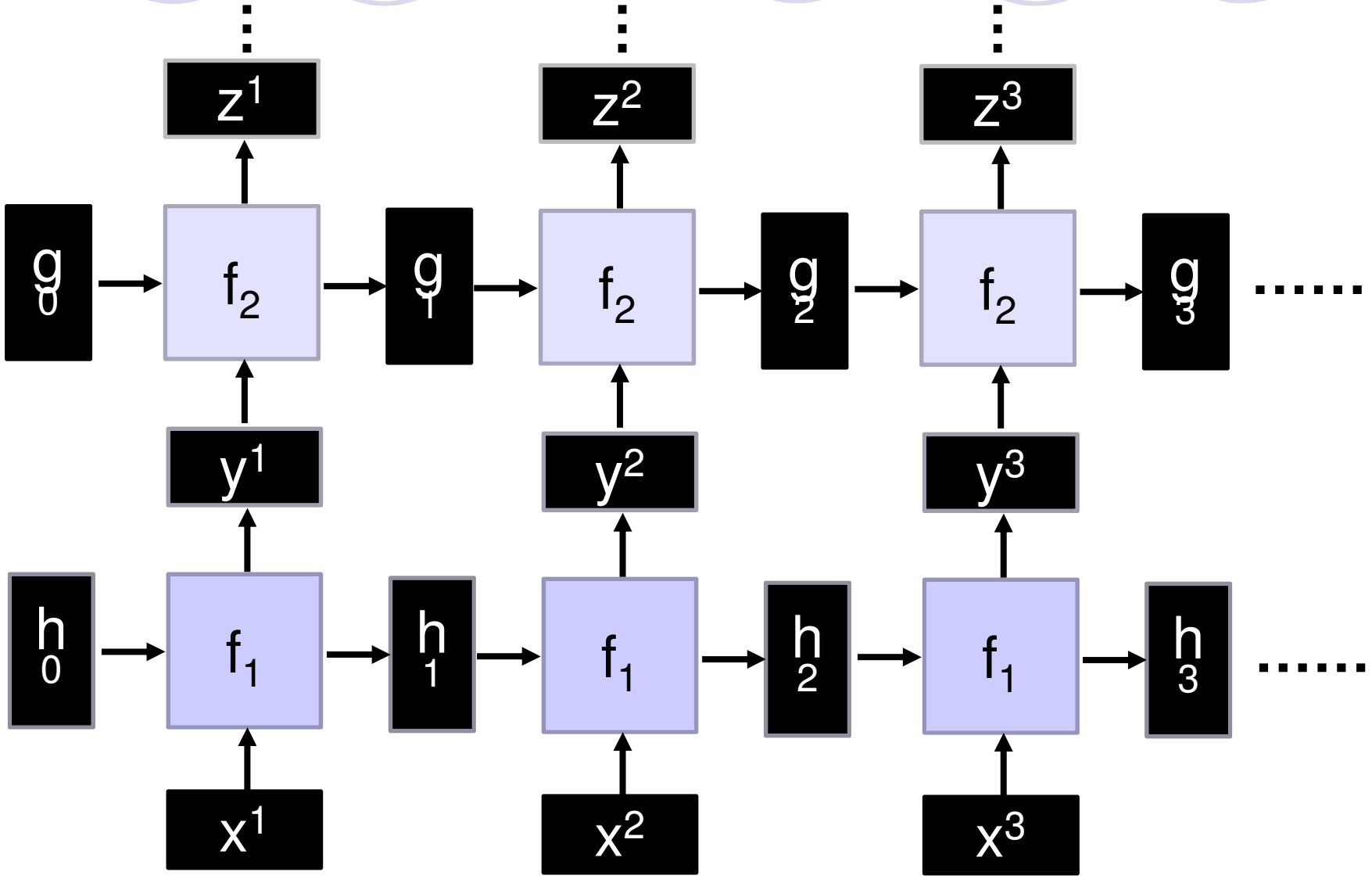
h and h' are vectors with the same dimension



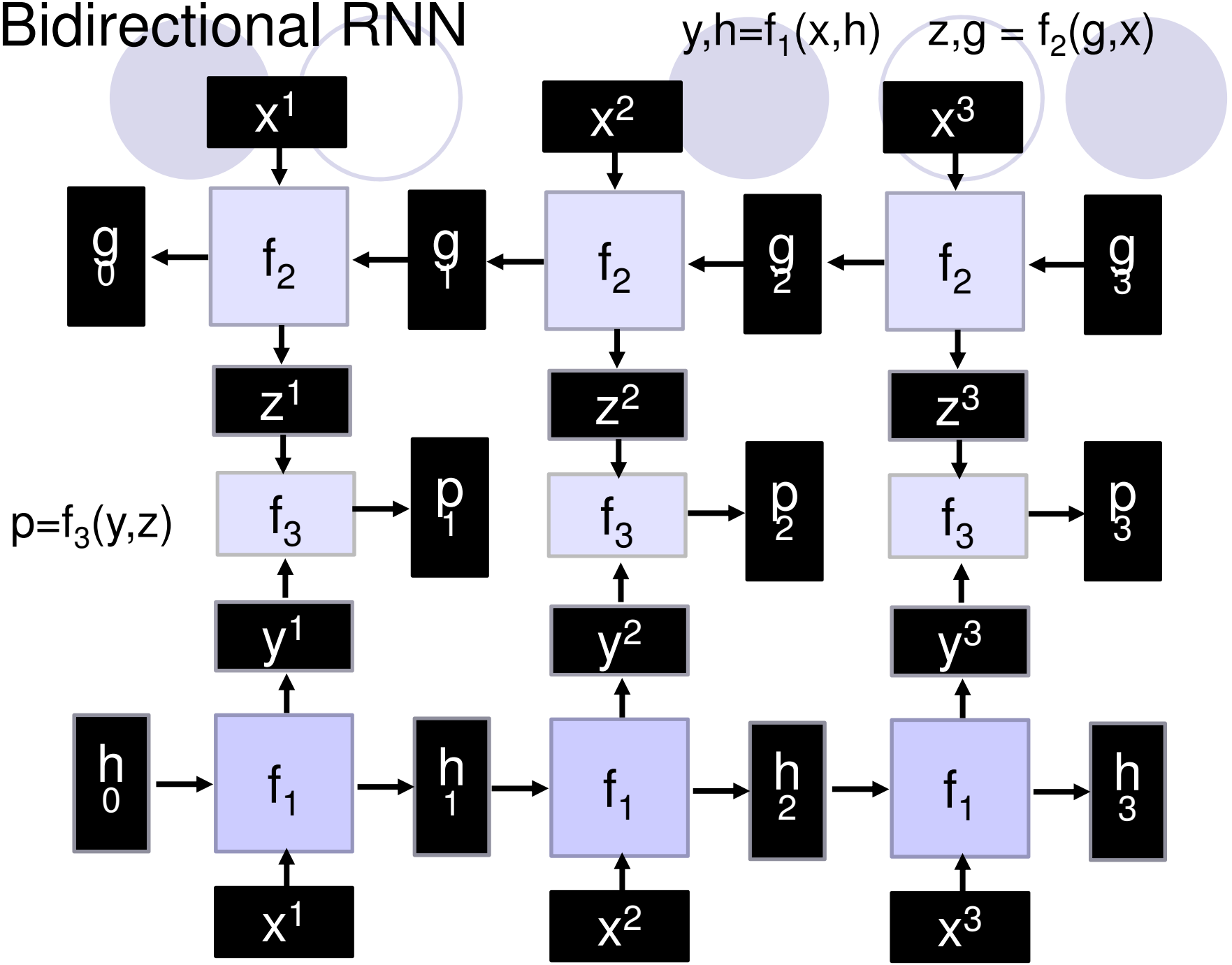
No matter how long the input/output sequence is, we only need one function f. If f's are different, then it becomes a feedforward NN. This may be treated as another compression from fully connected network.

Deep RNN

$h', y = f_1(h, x)$, $g', z = f_2(g, y)$ ...



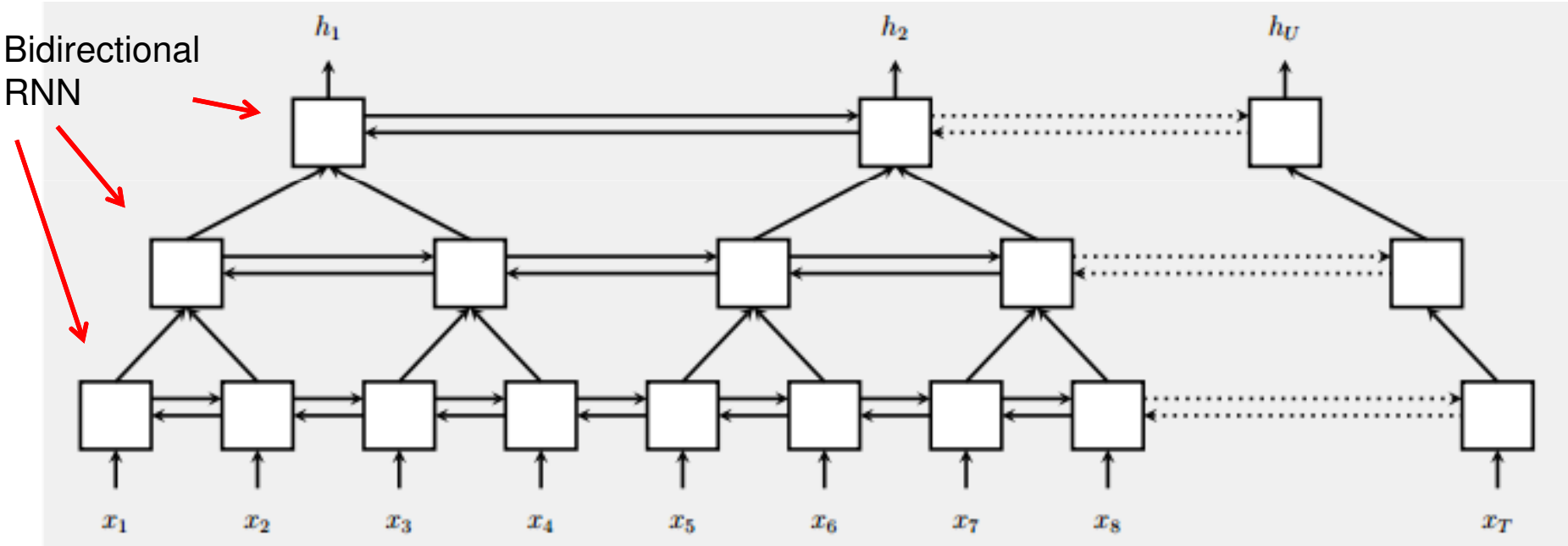
Bidirectional RNN



Pyramid RNN

Significantly speed up training

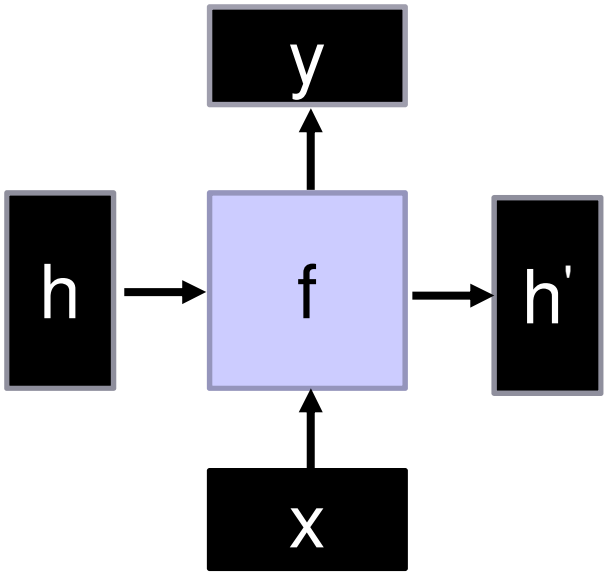
- Reducing the number of time steps



W. Chan, N. Jaitly, Q. Le and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," ICASSP, 2016

Naïve RNN

- Given function f: h',y=f(h,x)



$$h' = \sigma(W^h h + W^i x)$$

$$y = \sigma(W^o h')$$

softmax

Note, y is computed from h'

We have ignored the bias

Problems with naive RNN

- When dealing with a time series, it tends to forget old information. When there is a distant relationship of unknown length, we wish to have a “memory” to it.
- Vanishing gradient problem.

To continue…