CS 4803 / 7643: Deep Learning

Topics:

- Recurrent Neural Networks (RNNs)
- BackProp Through Time (BPTT)

Dhruv Batra

Georgia Tech

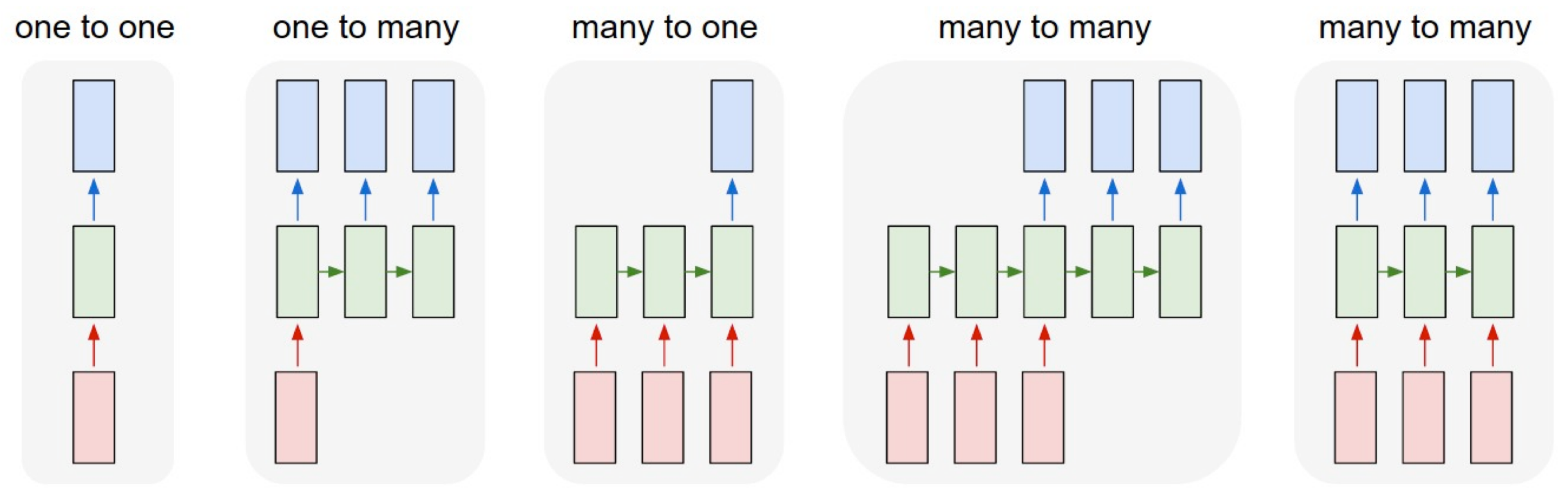
Administrativia

- HW3 Released
  - Due: 11/06, 11:55pm
  - Last HW
  - Focus on projects after this
  - https://www.cc.gatech.edu/classes/AY2019/cs7643_fall/assets/hw3.pdf

Plan for Today

- Model
  - Recurrent Neural Networks (RNNs)
- Learning
  - BackProp Through Time (BPTT)

New Topic: RNNs



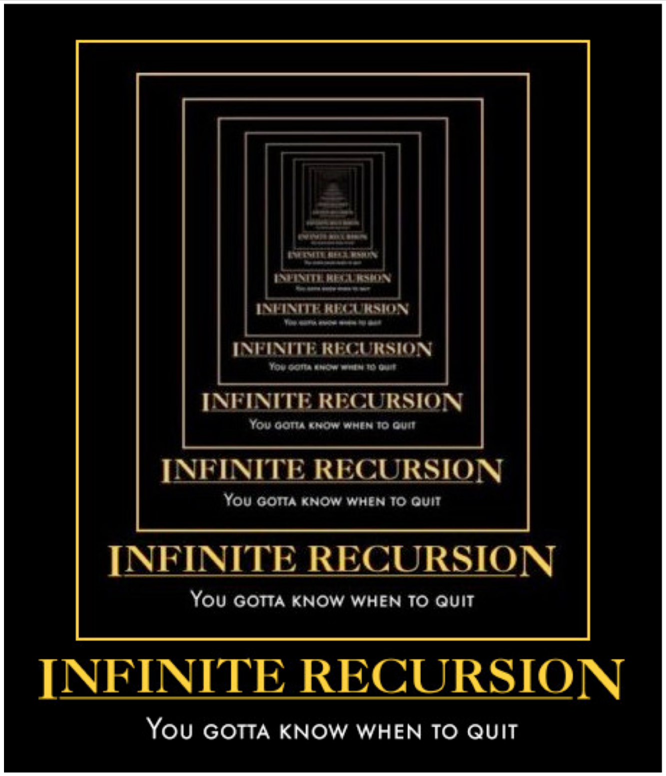


Image Credit: Andrej Karpathy

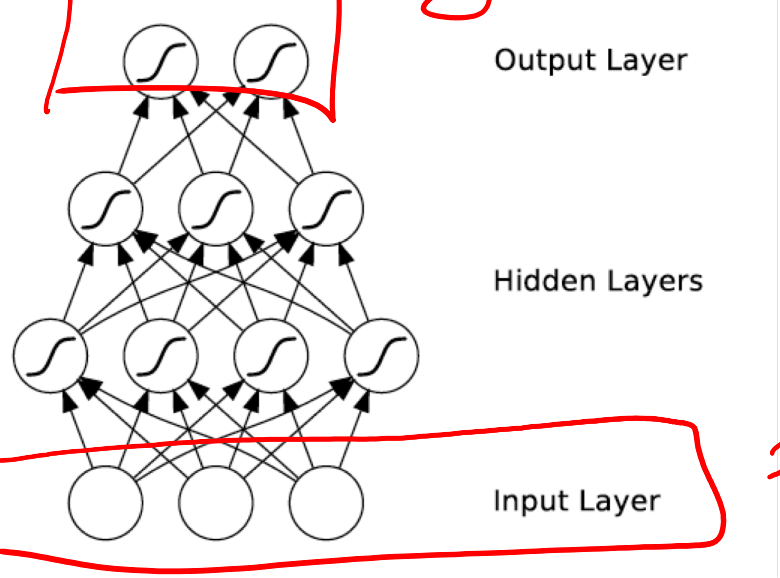
New Words

- Recurrent Neural Networks (RNNs)
- Recursive Neural Networks
  - General family; think graphs instead of chains
- Types:
  - "Vanilla" RNNs (Elman Networks)
  - Long Short Term Memory (LSTMs)
  - Gated Recurrent Units (GRUs)
  - …
- Algorithms
  - BackProp Through Time (BPTT)
  - BackProp Through Structure (BPTS)

What's wrong with MLPs?

- Problem 1: Can't model sequences
  - Fixed-sized Inputs & Outputs
  - No temporal structure

$y \in \mathbb{R}^k$

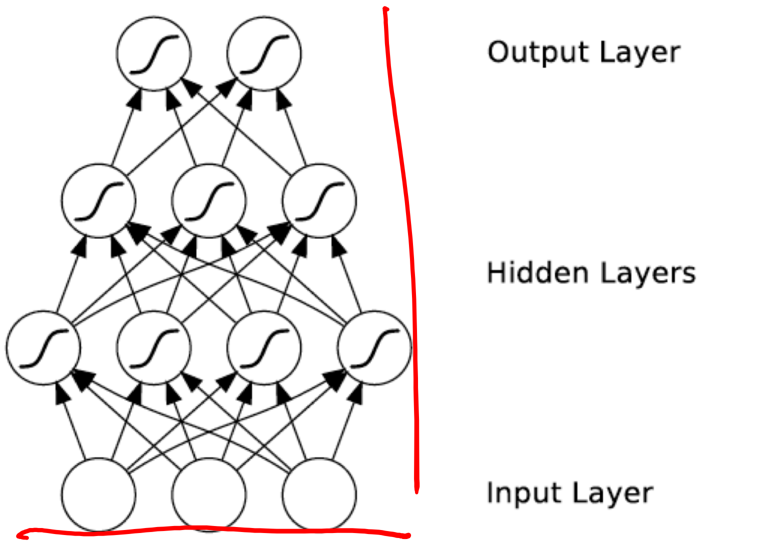


$x \in \mathbb{R}^d$

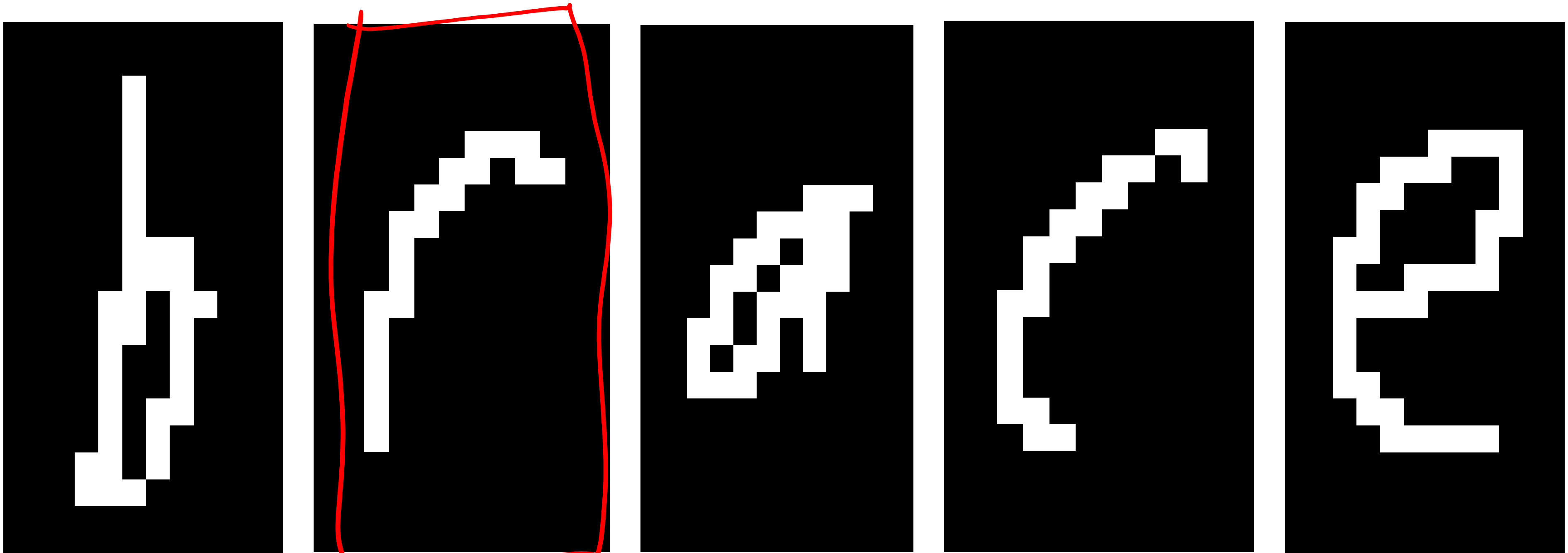
Image Credit: Alex Graves, book

What's wrong with MLPs?

- Problem 1: Can't model sequences
  - Fixed-sized Inputs & Outputs
  - No temporal structure
- Problem 2: Pure feed-forward processing
  - No "memory", no feedback



Why model sequences?



r c

Figure Credit: Carlos Guestrin

Why model sequences?

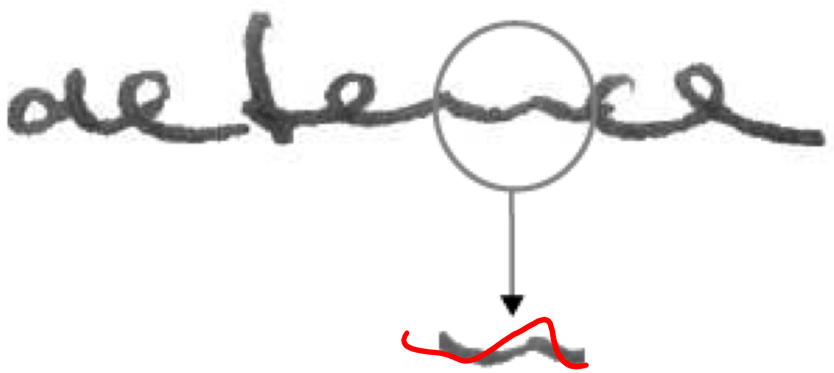
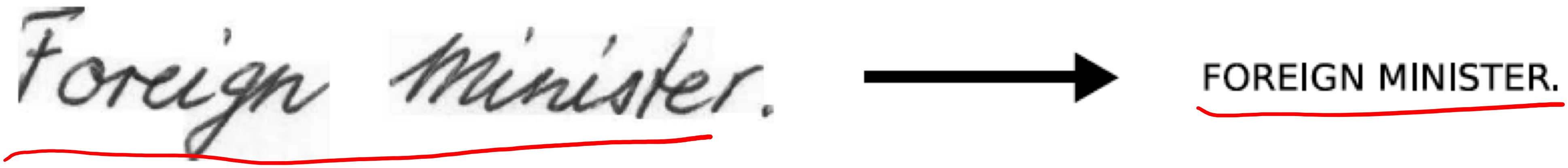


Image Credit: Alex Graves

Sequences are everywhere…



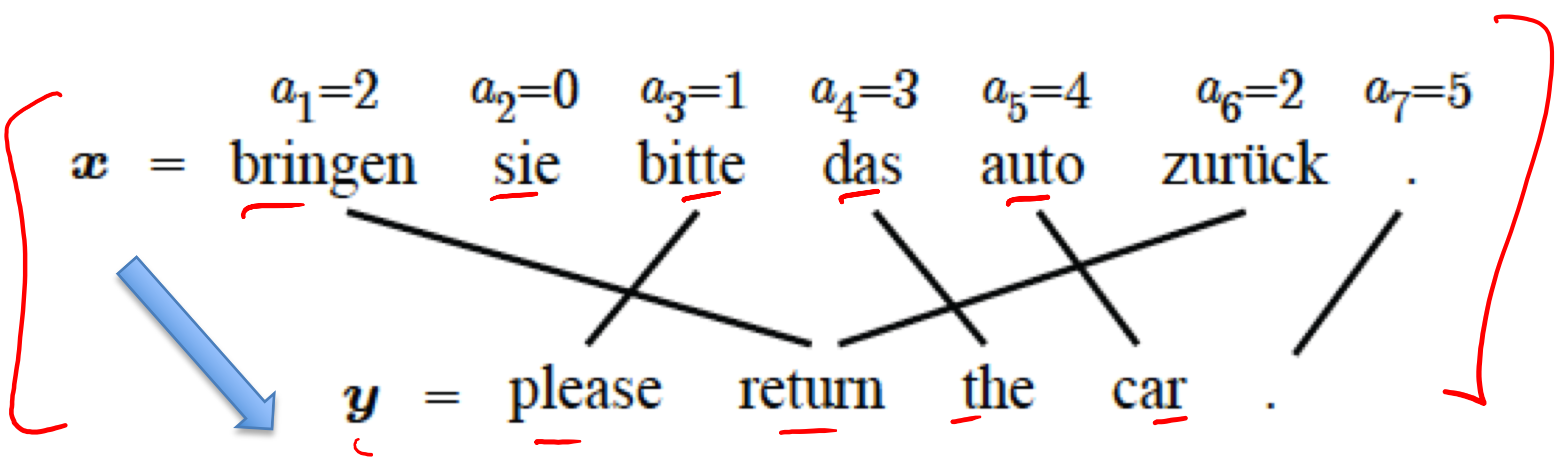
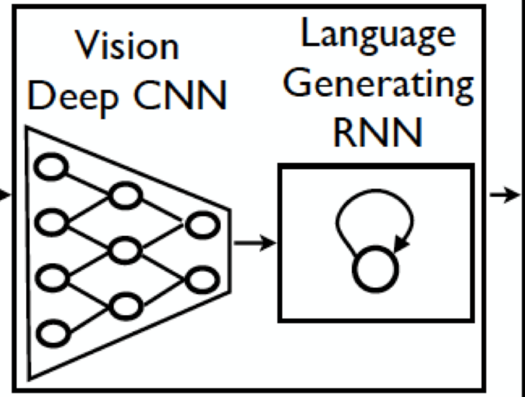


Image Credit: Alex Graves and Kevin Gimpel

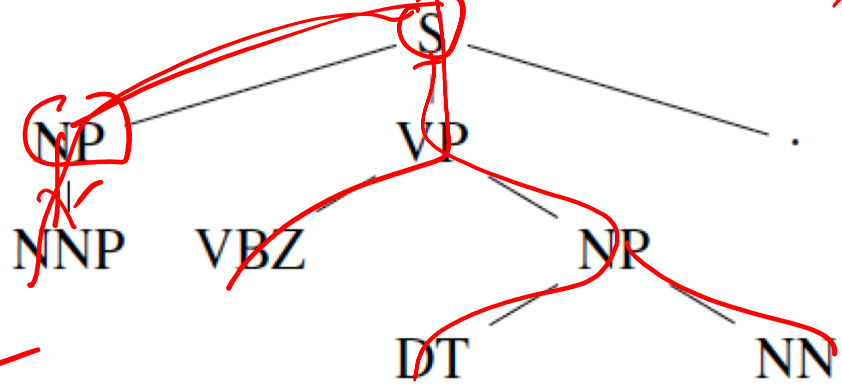
Even where you might not expect a sequence…



A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

John has a dog . →



John has a dog . → (S (NP NNP)$_{NP}$ (VP VBZ (NP DT NN)$_{NP}$)$_{VP}$.)$_S$

Image Credit: Vinyals et al.

Even where you might not expect a sequence…

Classify images by taking a series of "glimpses"

Ba, Mnih, and Kavukcuoglu, "Multiple Object Recognition with Visual Attention", ICLR 2015.
Gregor et al, "DRAW: A Recurrent Neural Network For Image Generation", ICML 2015

Even where you might not expect a sequence…

- Output ordering = sequence

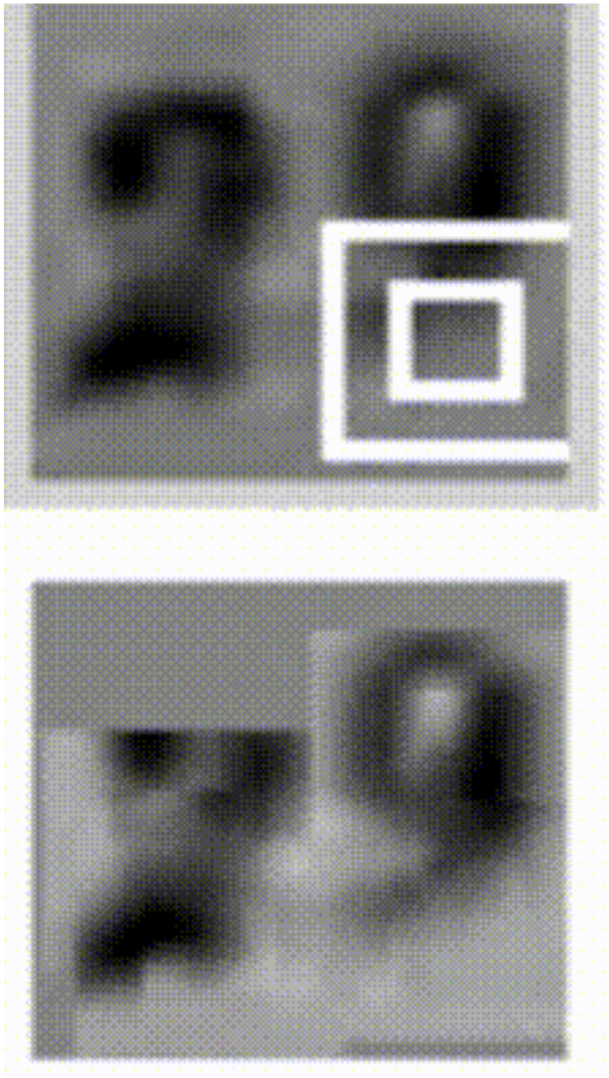
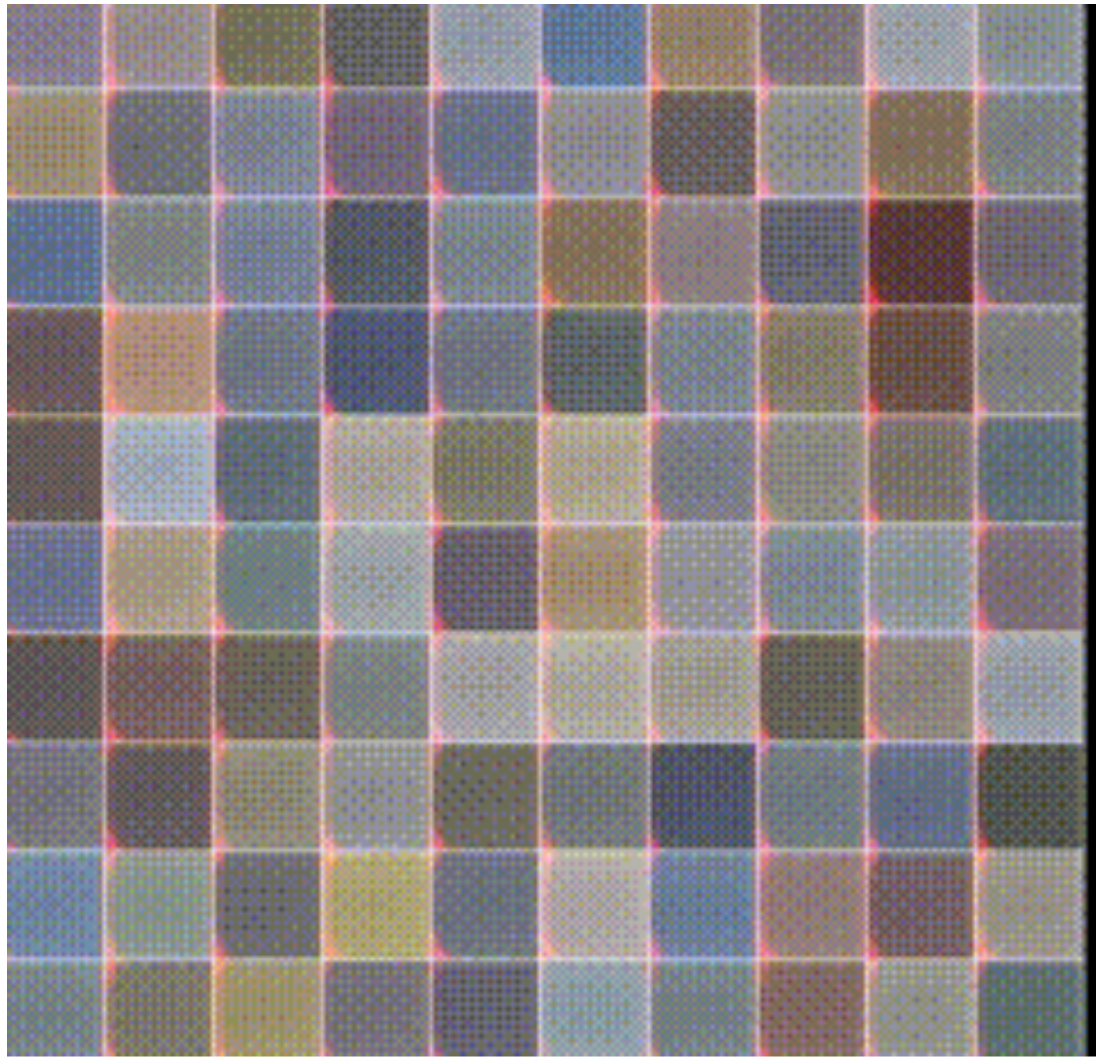
 

Image Credit: Ba et al.; Gregor et al

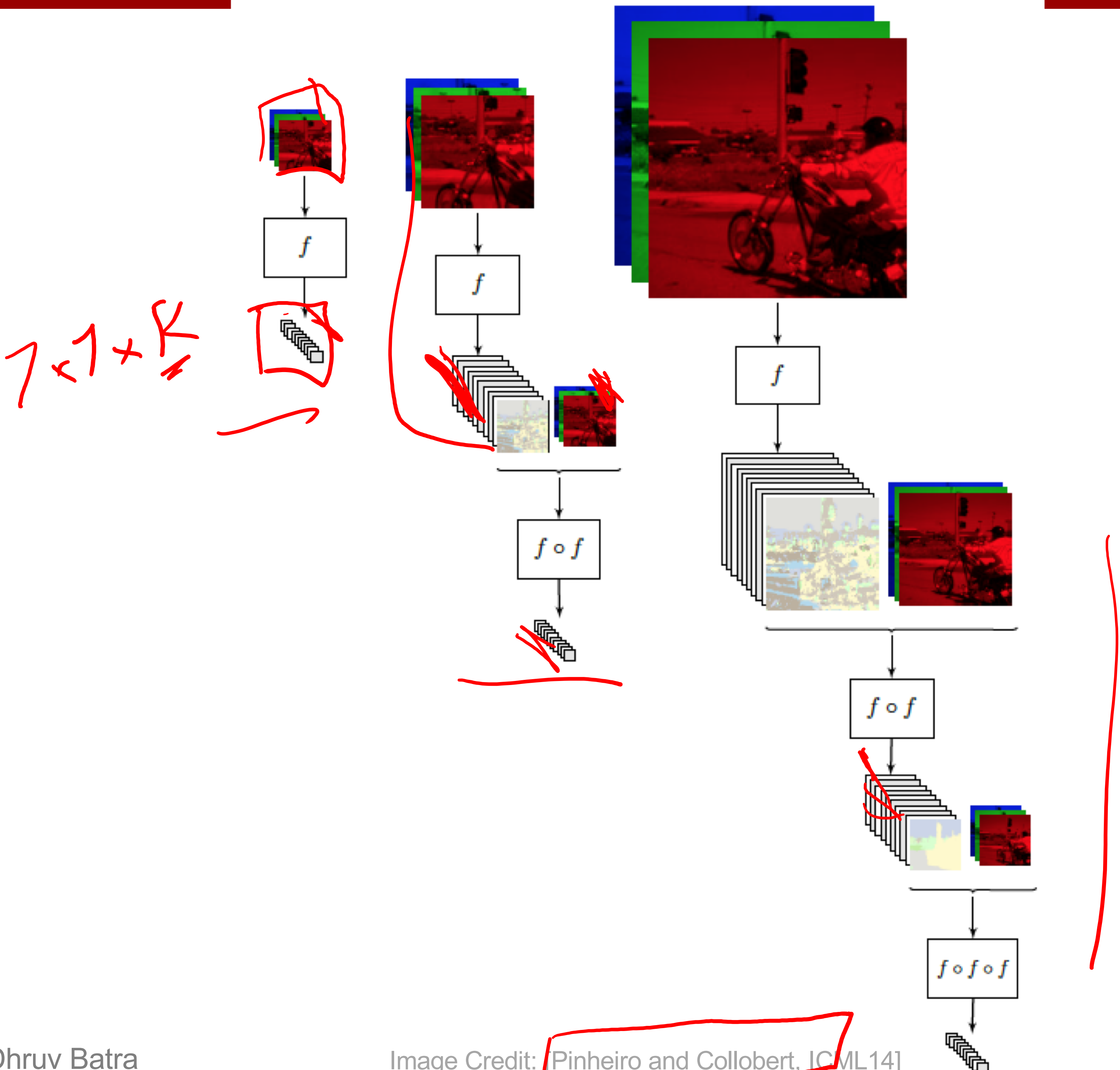
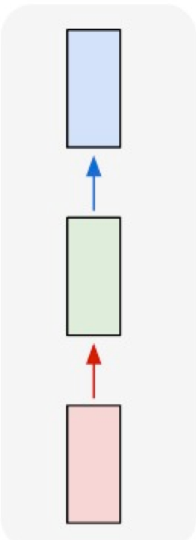


Image Credit: [Pinheiro and Collobert, ICML14]

Sequences in Input or Output?

- It's a spectrum…

one to one



Input: No sequence

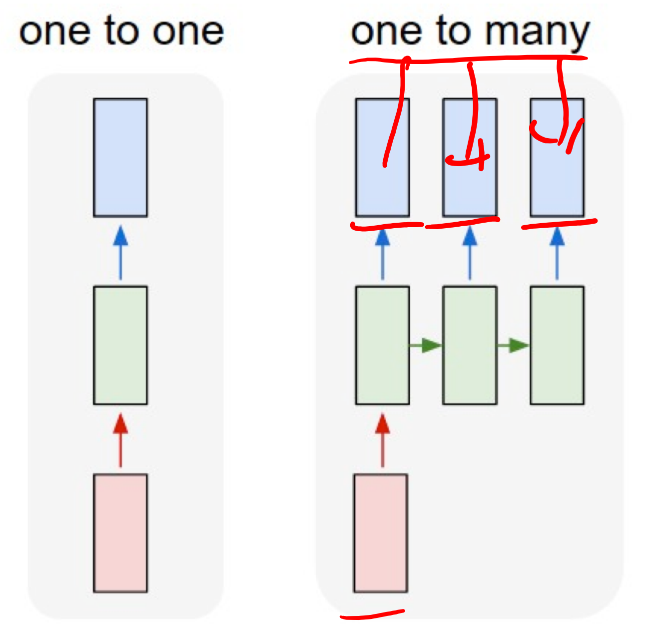
Output: No sequence

Example: "standard" classification / regression problems

Image Credit: Andrej Karpathy

Sequences in Input or Output?

- It's a spectrum…



Input: No sequence

Output: No sequence

Example: "standard" classification / regression problems

Input: No sequence

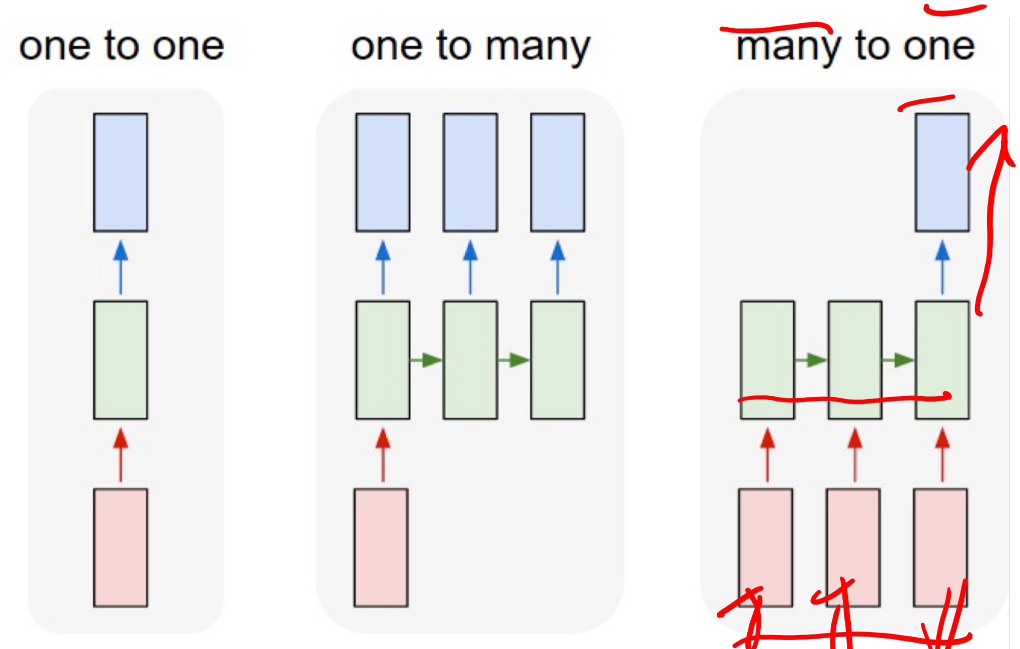
Output: Sequence

Example: Im2Caption

Image Credit: Andrej Karpathy

Sequences in Input or Output?

- It's a spectrum…

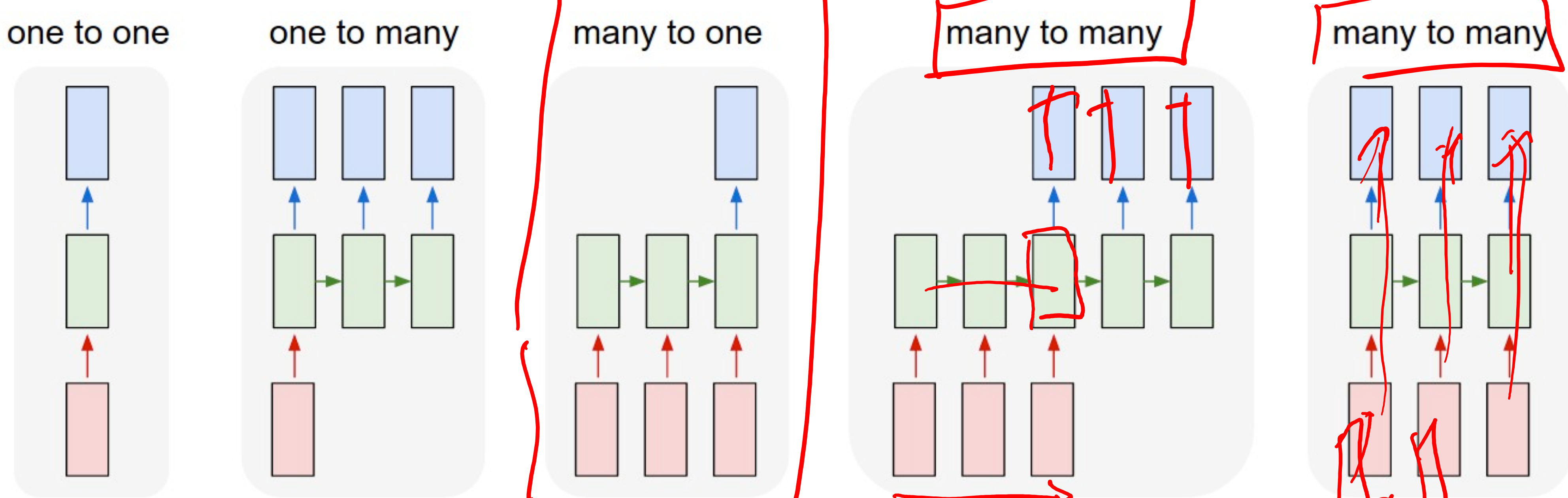


one to one	one to many	many to one
Input: No sequence	Input: No sequence	Input: Sequence
Output: No sequence	Output: Sequence	Output: No sequence
Example: "standard" classification / regression problems	Example: Im2Caption	Example: sentence classification, multiple-choice question answering

Image Credit: Andrej Karpathy

Sequences in Input or Output?

- It's a spectrum…



Input: No sequence

Output: No sequence

Example: "standard" classification / regression problems

Input: No sequence

Output: Sequence

Example: Im2Caption

Input: Sequence

Output: No sequence

Example: sentence classification, multiple-choice question answering

Input: Sequence

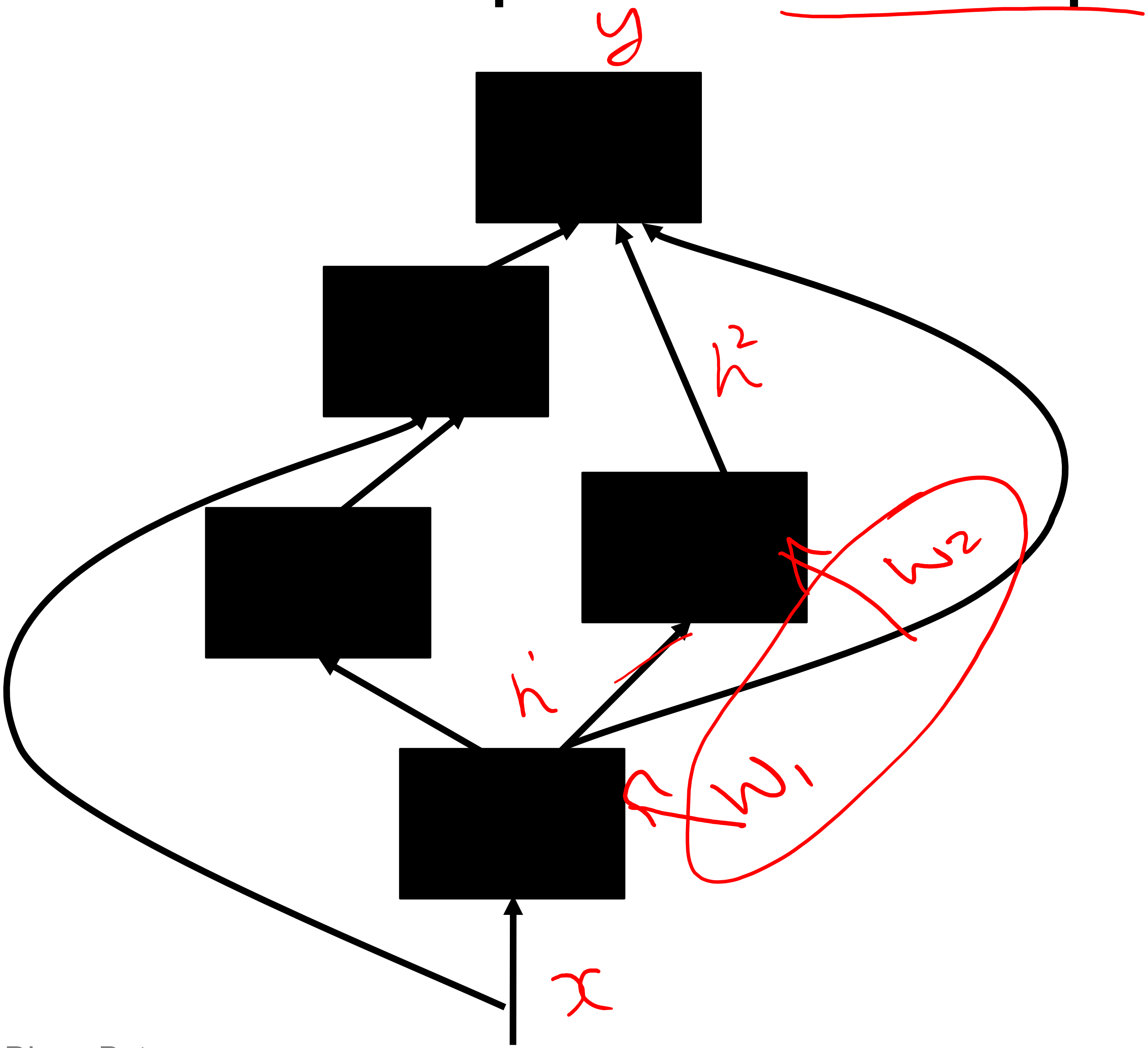
Output: Sequence

Example: machine translation, video classification, video captioning, open-ended question answering

2 Key Ideas

- Parameter Sharing
  - in computation graphs = adding gradients

Computational Graph



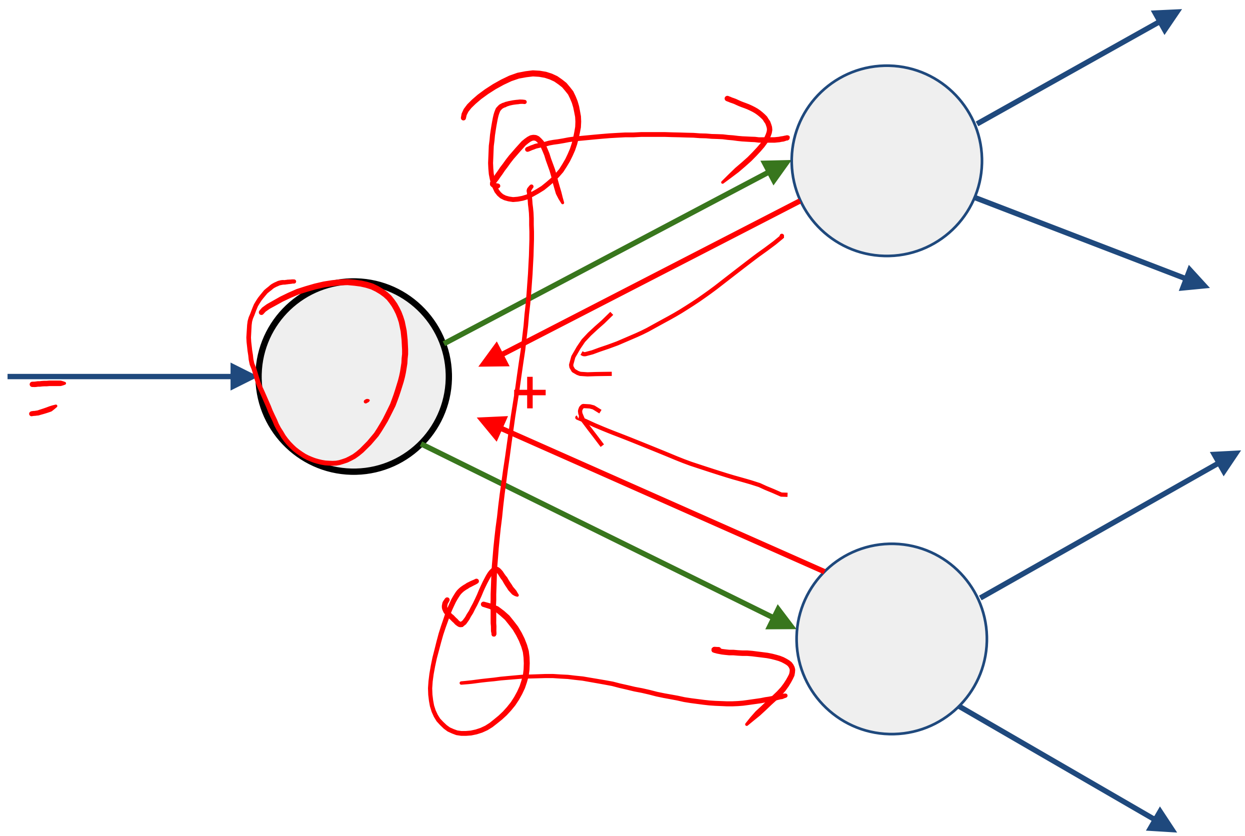
$$f(w_1, w_2) \qquad w_1(t) \quad w_2(t)$$

$$\frac{\partial f}{\partial w_1} = \lim_{\delta \to 0} \frac{f(w_1+\delta, w_2) - f(w_1, w_2)}{\delta}$$

$$\boxed{\frac{df}{dt}} = \boxed{\frac{\partial f}{\partial w_1}} \frac{\partial w_1}{\partial t} + \boxed{\frac{\partial f}{\partial w_2}} \frac{\partial w_2}{\partial t}$$

$$w_1 = t = w_2$$

Gradients add at branches



Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

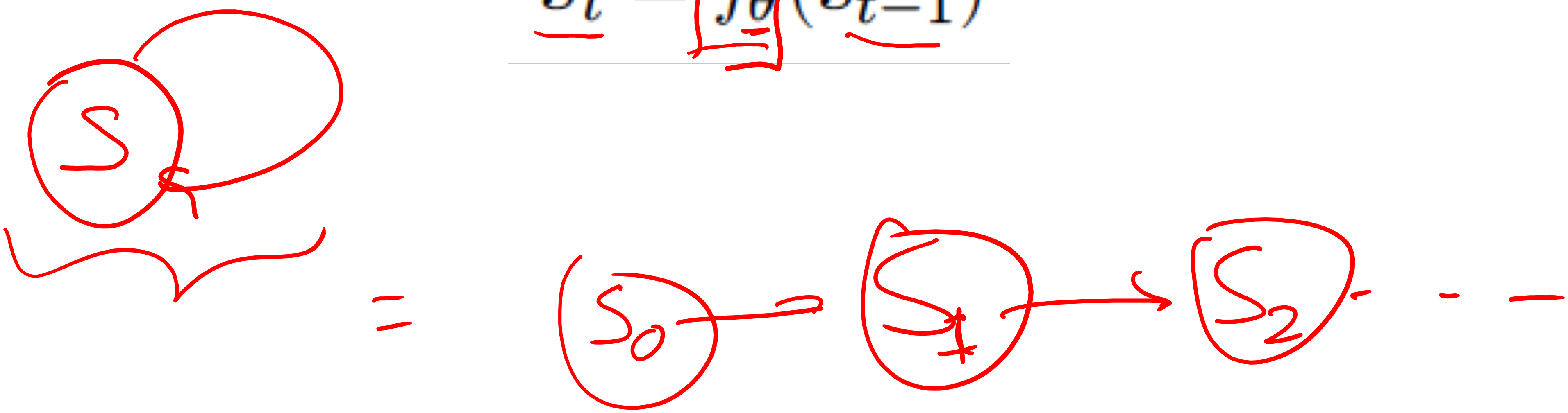
2 Key Ideas

- Parameter Sharing
  - in computation graphs = adding gradients
- “Unrolling”
  - in computation graphs with parameter sharing

How do we model sequences?

- No input

$$s_t = \boxed{f_\theta}(s_{t-1})$$



How do we model sequences?

- No input

$$s_t = f_\theta(s_{t-1})$$

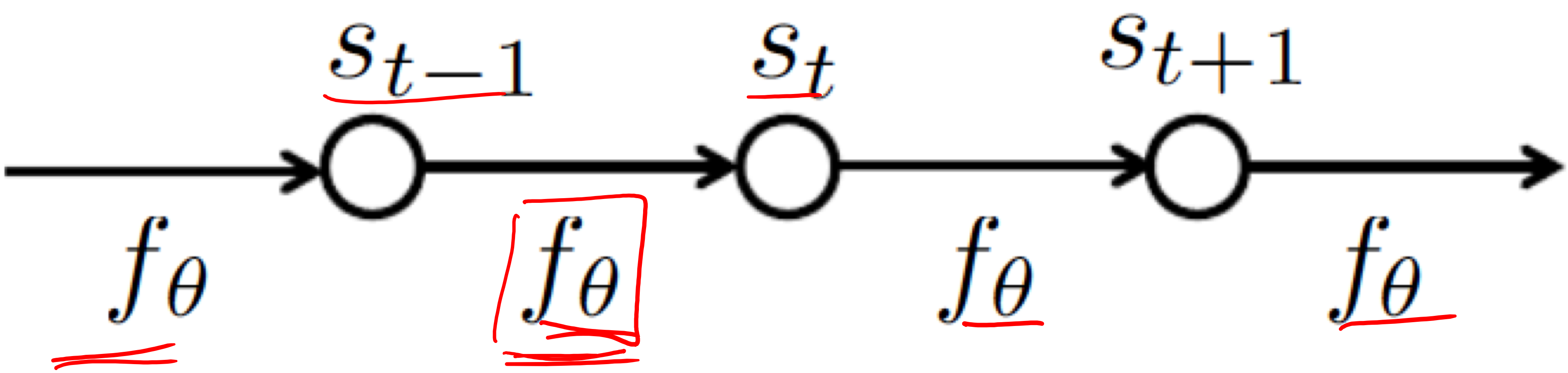


Image Credit: Bengio, Goodfellow, Courville

How do we model sequences?

- With inputs

$$\boxed{s_t} = \boxed{f_\theta}(s_{t-1}, x_t)$$

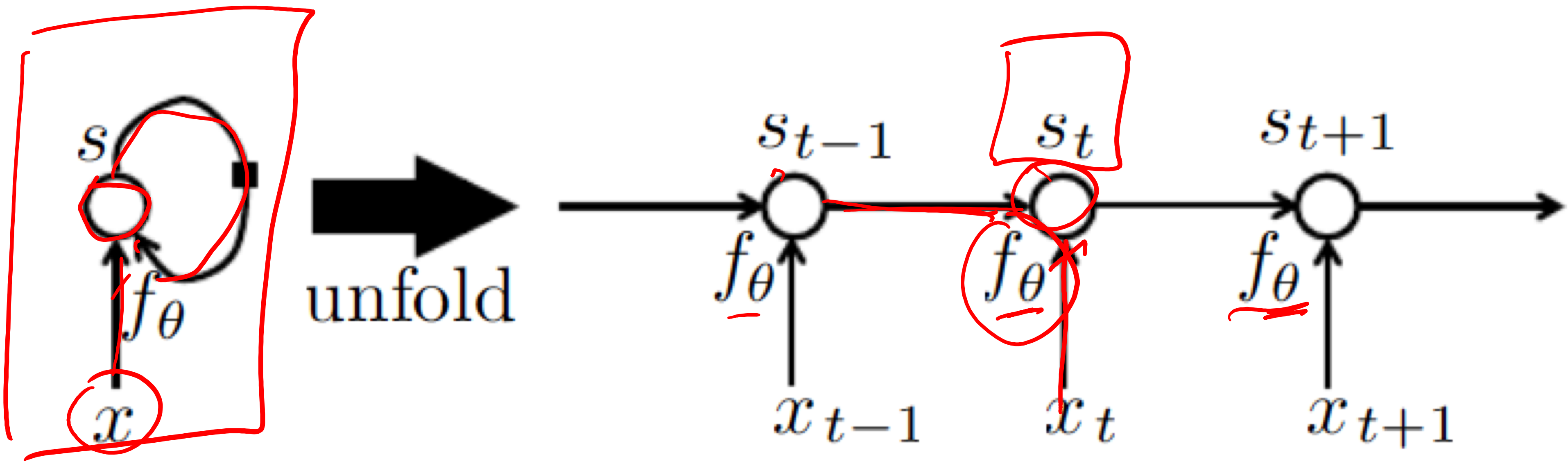
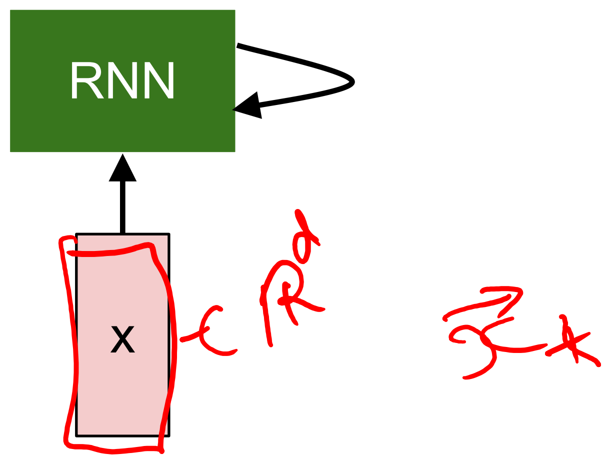


Image Credit: Bengio, Goodfellow, Courville

2 Key Ideas

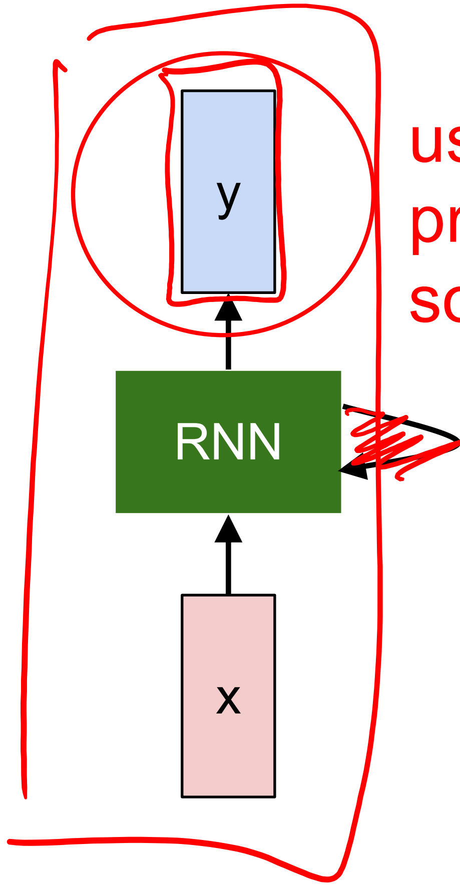
- Parameter Sharing
  – in computation graphs = adding gradients
- "Unrolling"
  – in computation graphs with parameter sharing
- Parameter sharing + Unrolling
  – Allows modeling arbitrary sequence lengths!
  – Keeps numbers of parameters in check

Recurrent Neural Network



Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Recurrent Neural Network



usually want to
predict a vector at
some time steps

Recurrent Neural Network

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

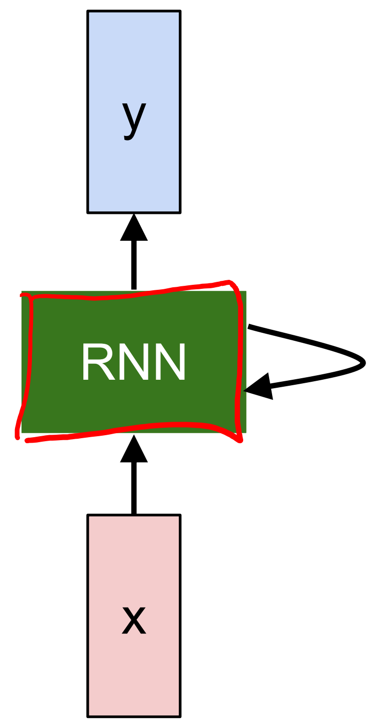
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

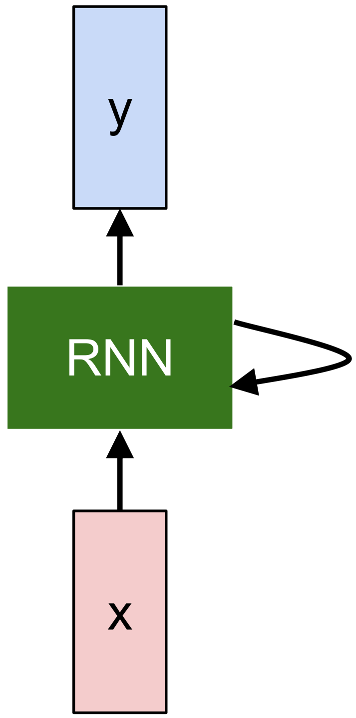
some function with parameters W



Recurrent Neural Network

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

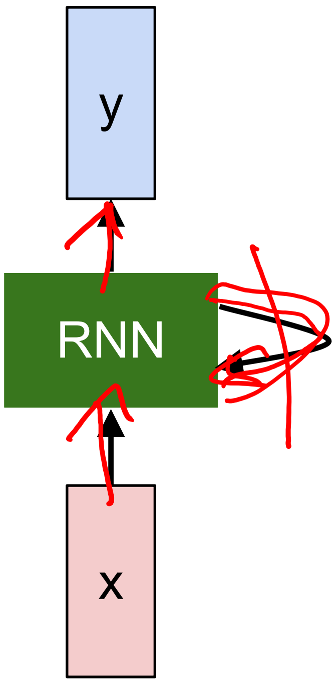
$$h_t = f_W(h_{t-1}, x_t)$$



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

(Vanilla) Recurrent Neural Network

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



$$\boxed{y_t} = \boxed{W_{hy}} h_t + b_y$$

cell

$$\boxed{\boxed{h_t} = \boxed{f_W}(\vec{h}_{t-1}, \vec{x}_t)}$$

$W \; d_2 \times (d_1 + d_2) + d_2$

$[W_{hh} \;\; W_{xh}]$

$$\vec{h}_t = \tanh(\boxed{W_{hh} h_{t-1}} + W_{xh} \vec{x}_t + \vec{b}_h)$$

$\vec{x}_t \in \mathbb{R}^{d_1}$

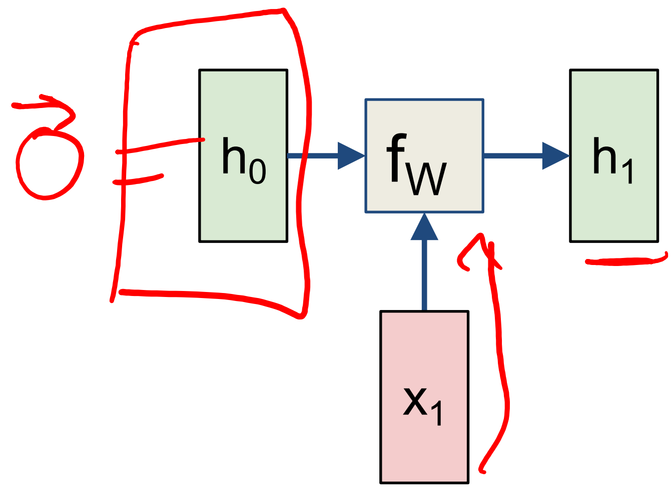
$\vec{h}_t \in \mathbb{R}^{d_2}$

$$\vec{h}_t = \tanh\left(W \begin{bmatrix} \vec{h}_{t-1} \\ \vec{x}_t \end{bmatrix} + \vec{b} \right)$$

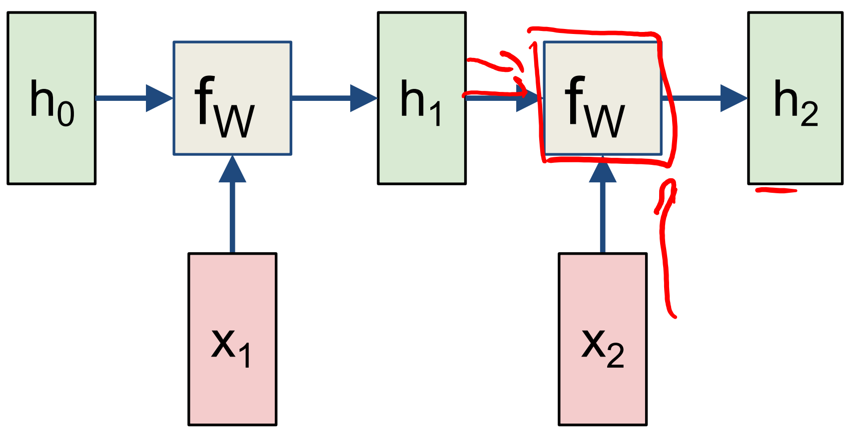
Sometimes called a "Vanilla RNN" or an "Elman RNN" after Prof. Jeffrey Elman

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

RNN: Computational Graph

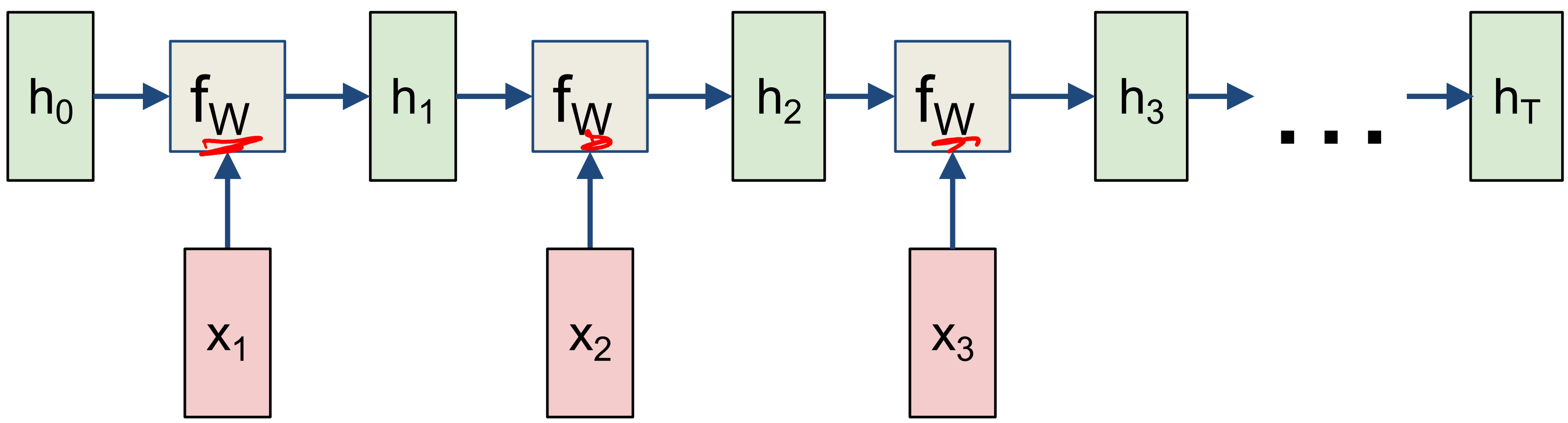


RNN: Computational Graph



Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

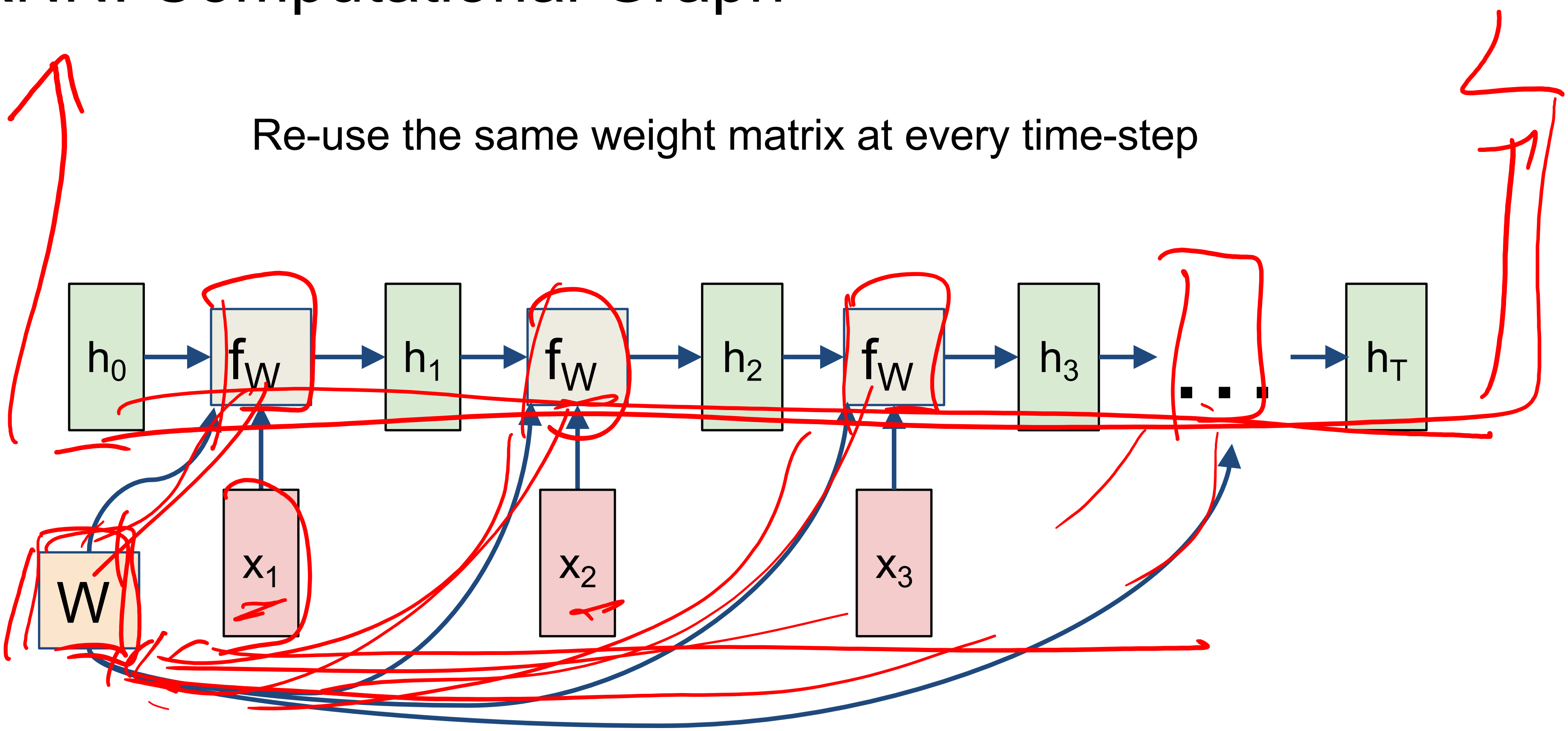
RNN: Computational Graph



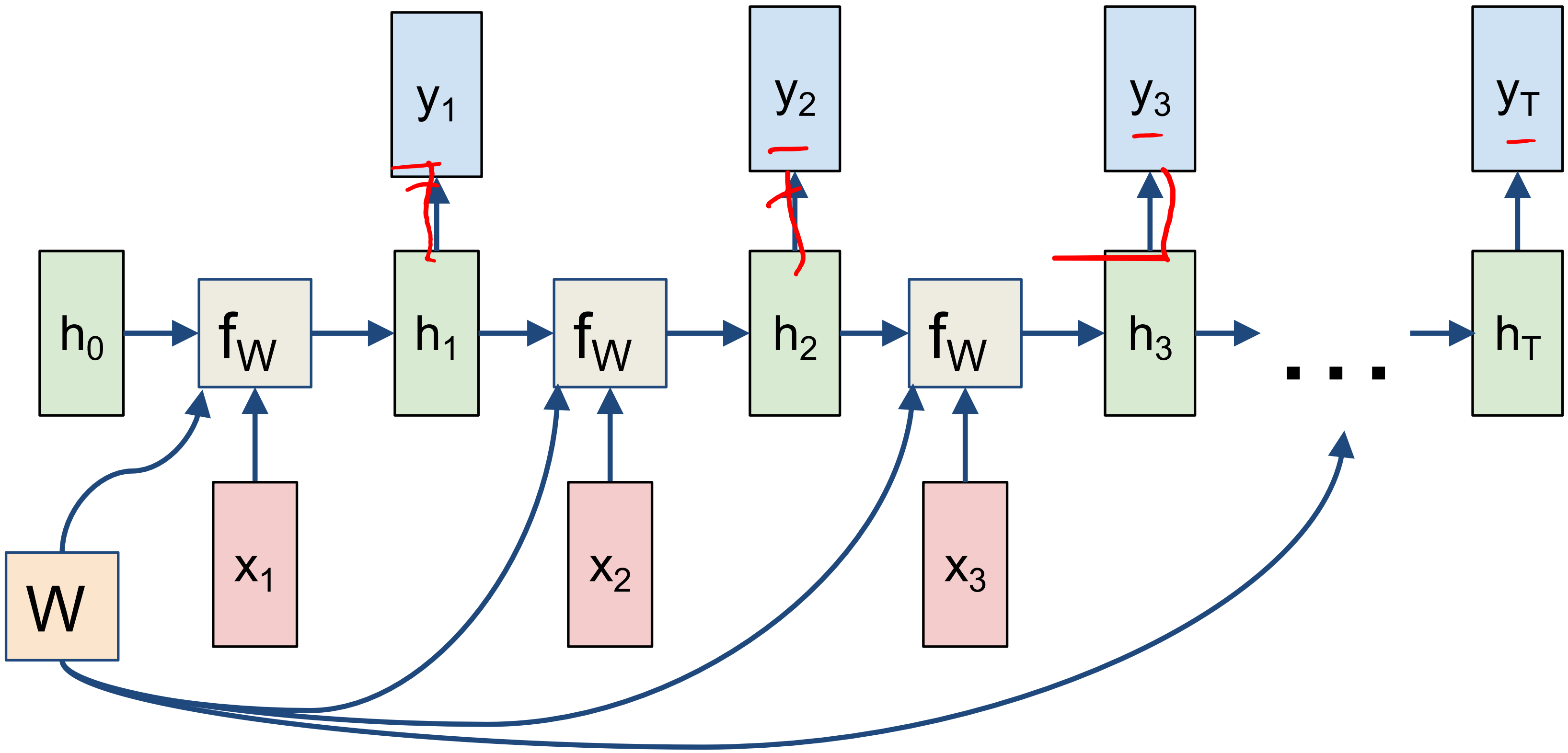
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

RNN: Computational Graph

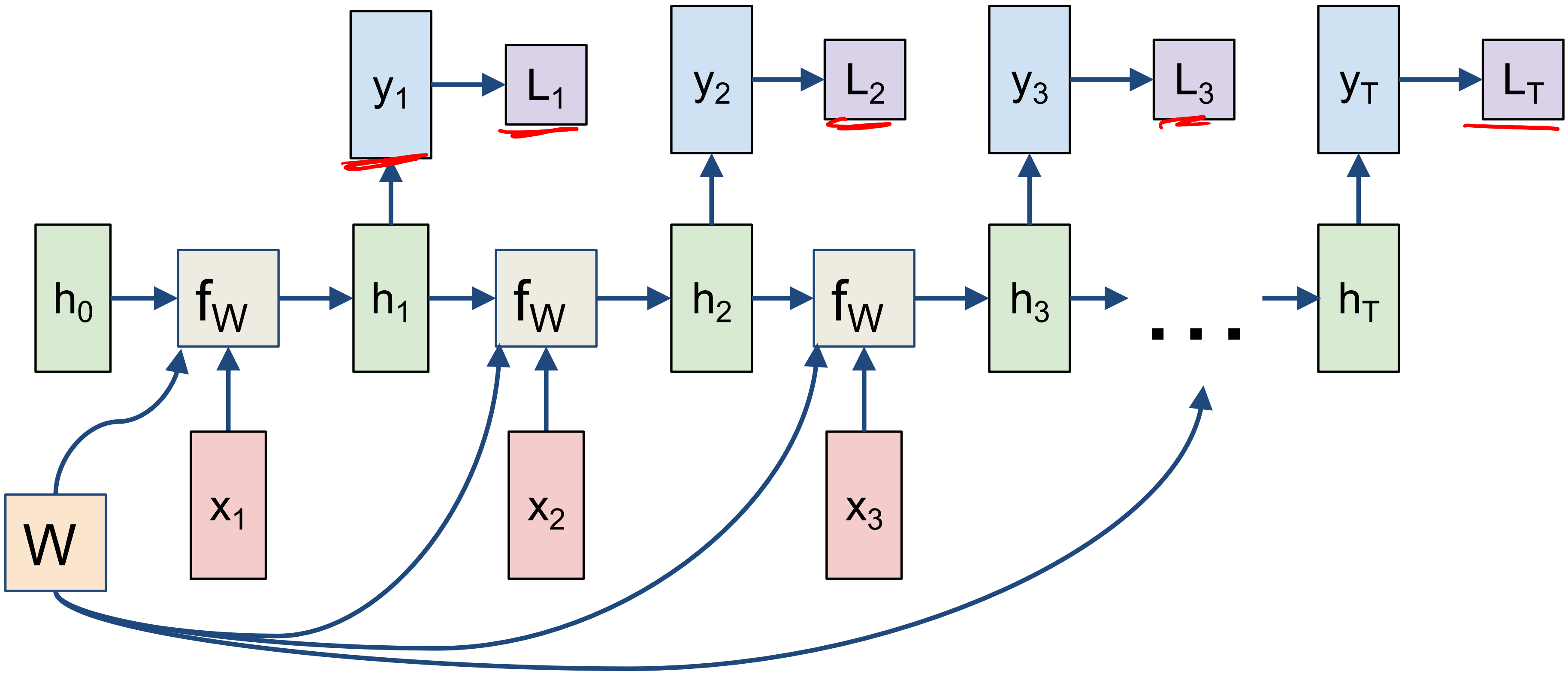
Re-use the same weight matrix at every time-step



RNN: Computational Graph: Many to Many

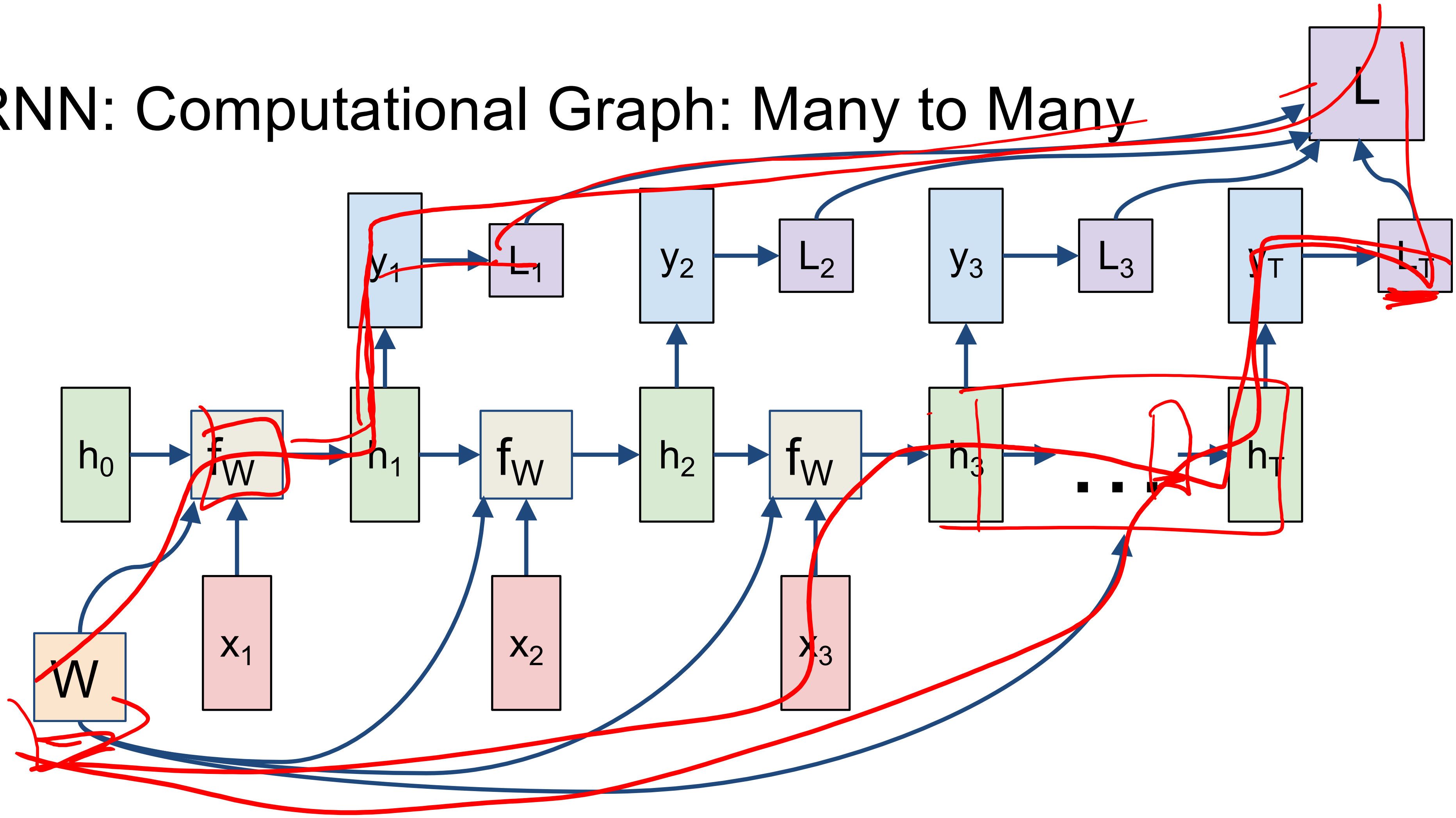


RNN: Computational Graph: Many to Many



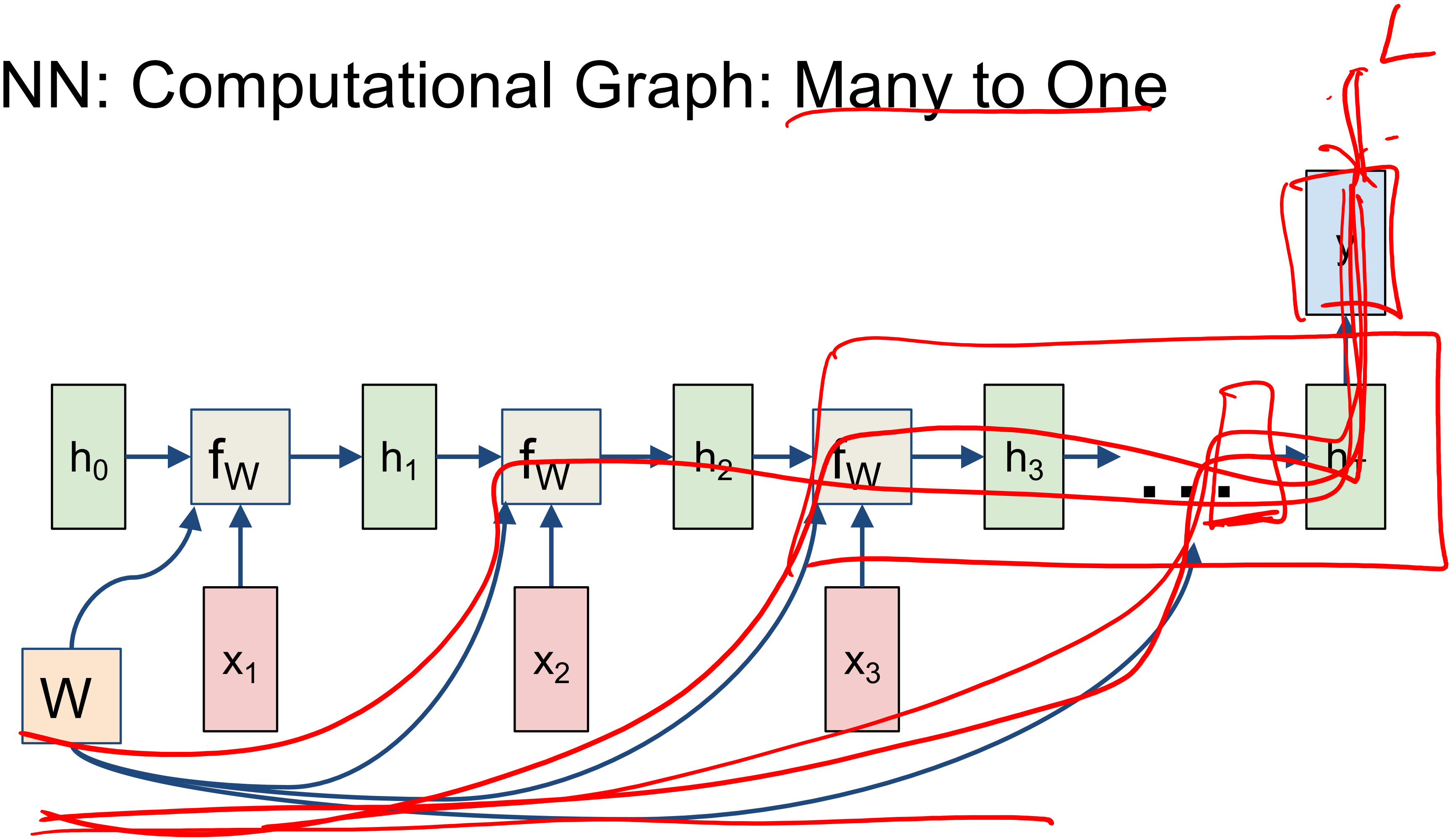
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

RNN: Computational Graph: Many to Many



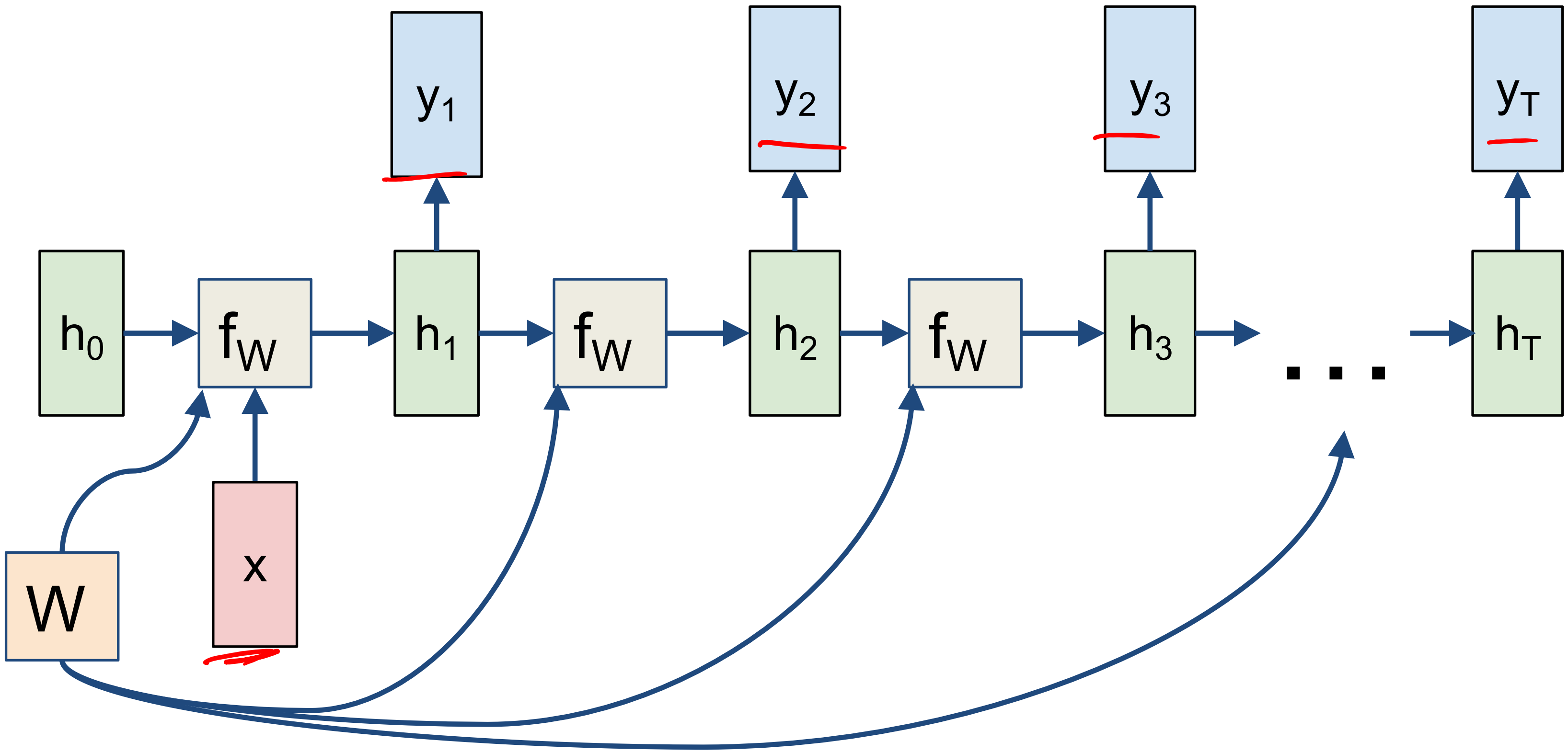
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

RNN: Computational Graph: Many to One



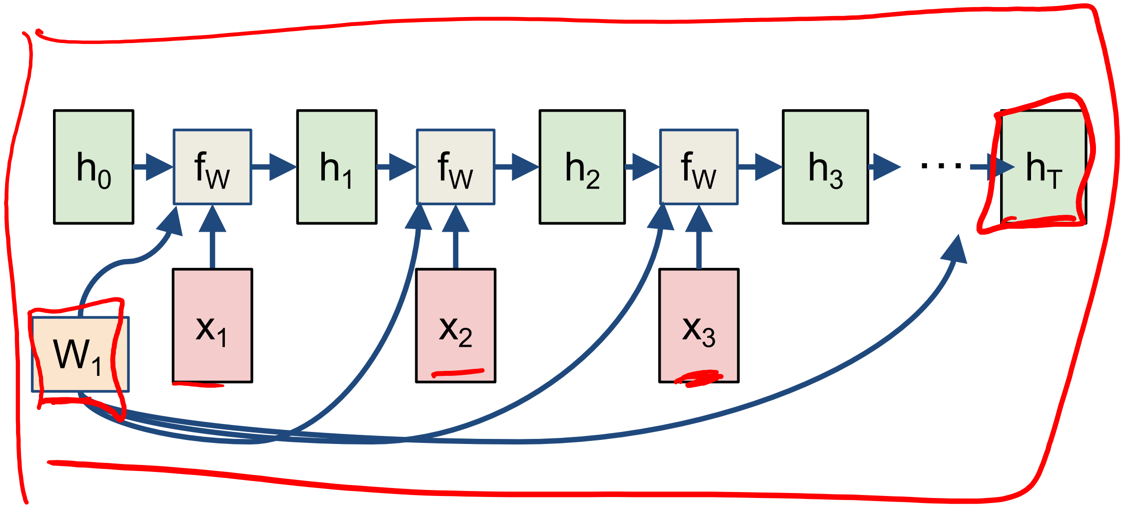
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

RNN: Computational Graph: One to Many



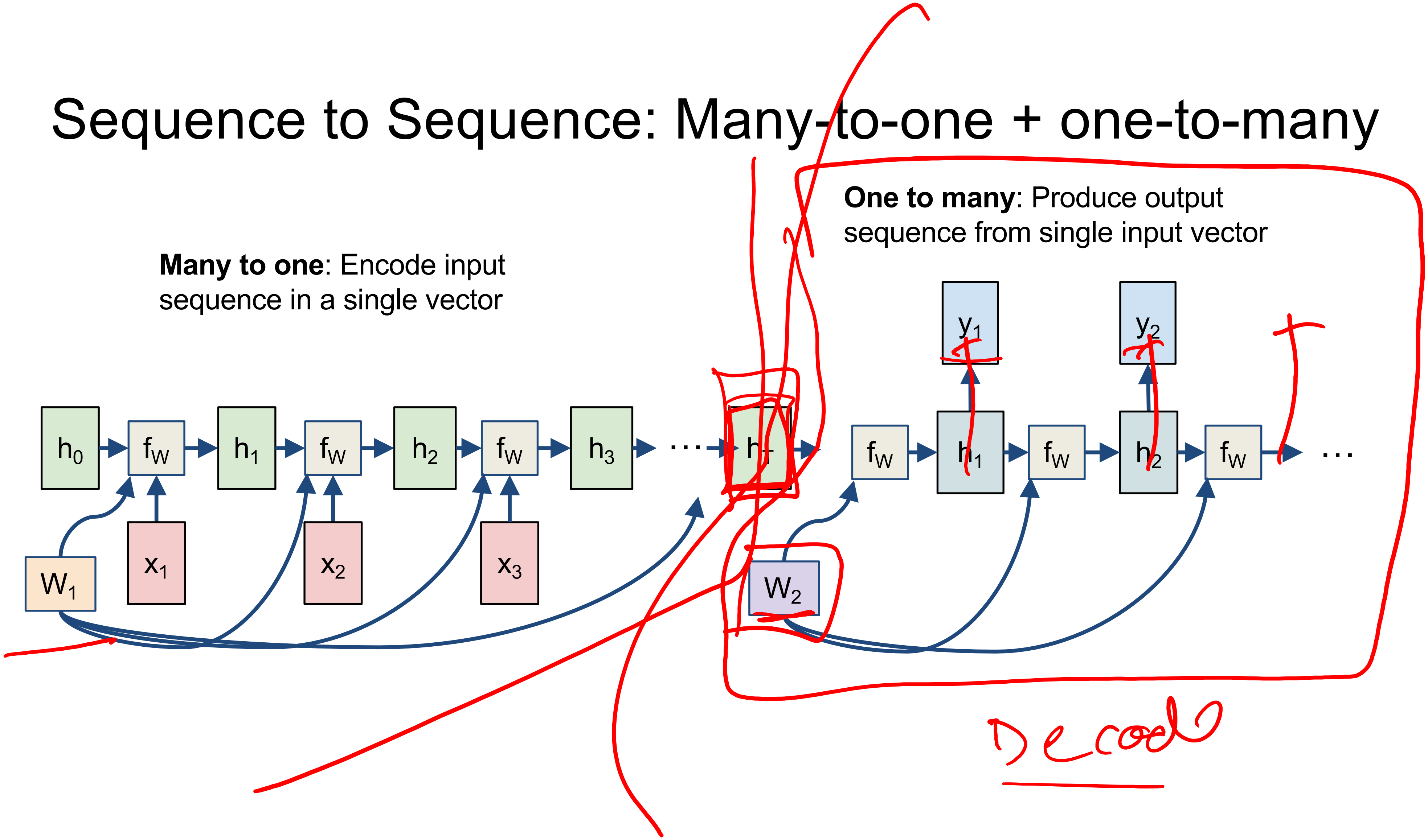
Sequence to Sequence: Many-to-one + one-to-many

Many to one: Encode input sequence in a single vector



Encoder

Sequence to Sequence: Many-to-one + one-to-many



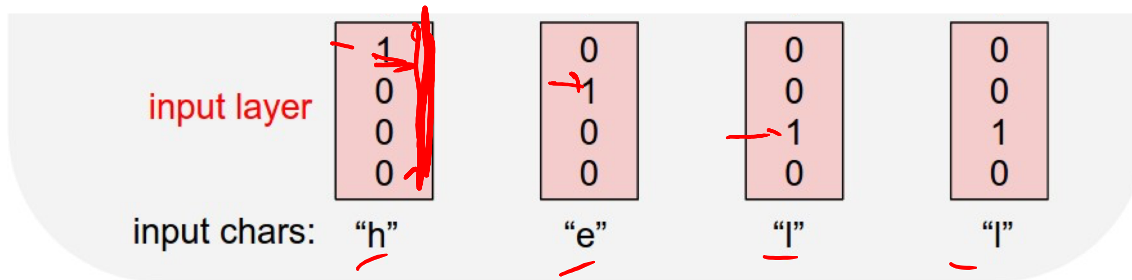
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

$$\boxed{P(x_1 \ldots x_T)} = P(x_1)P(x_2|x_1) \ldots \ldots P(x_t|x_1 \ldots x_{t-1}) \ldots$$

Example:
Character-level Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”



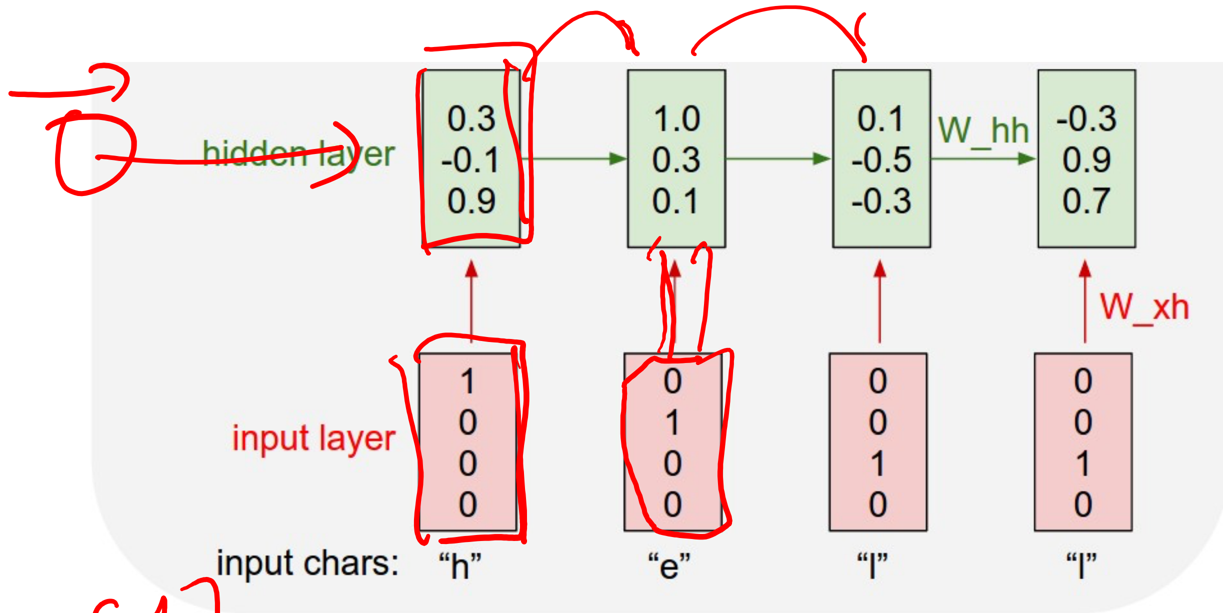
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Example: Character-level Language Model

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

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”



$$\begin{bmatrix} | & | & | & | \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

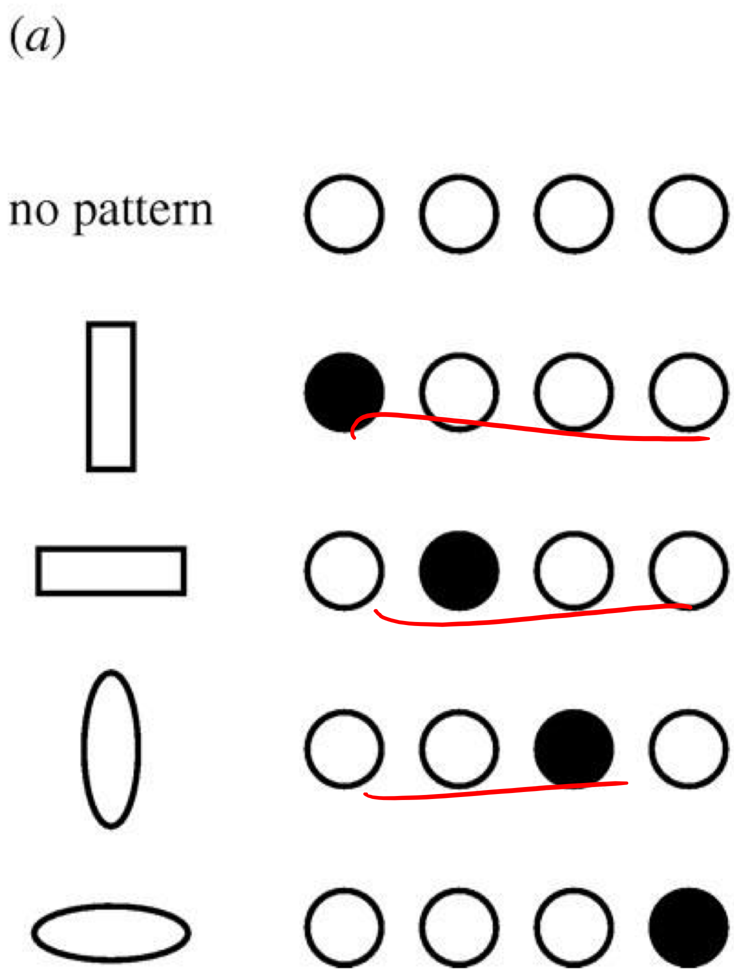
$|H| \times |X|$

3×4

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

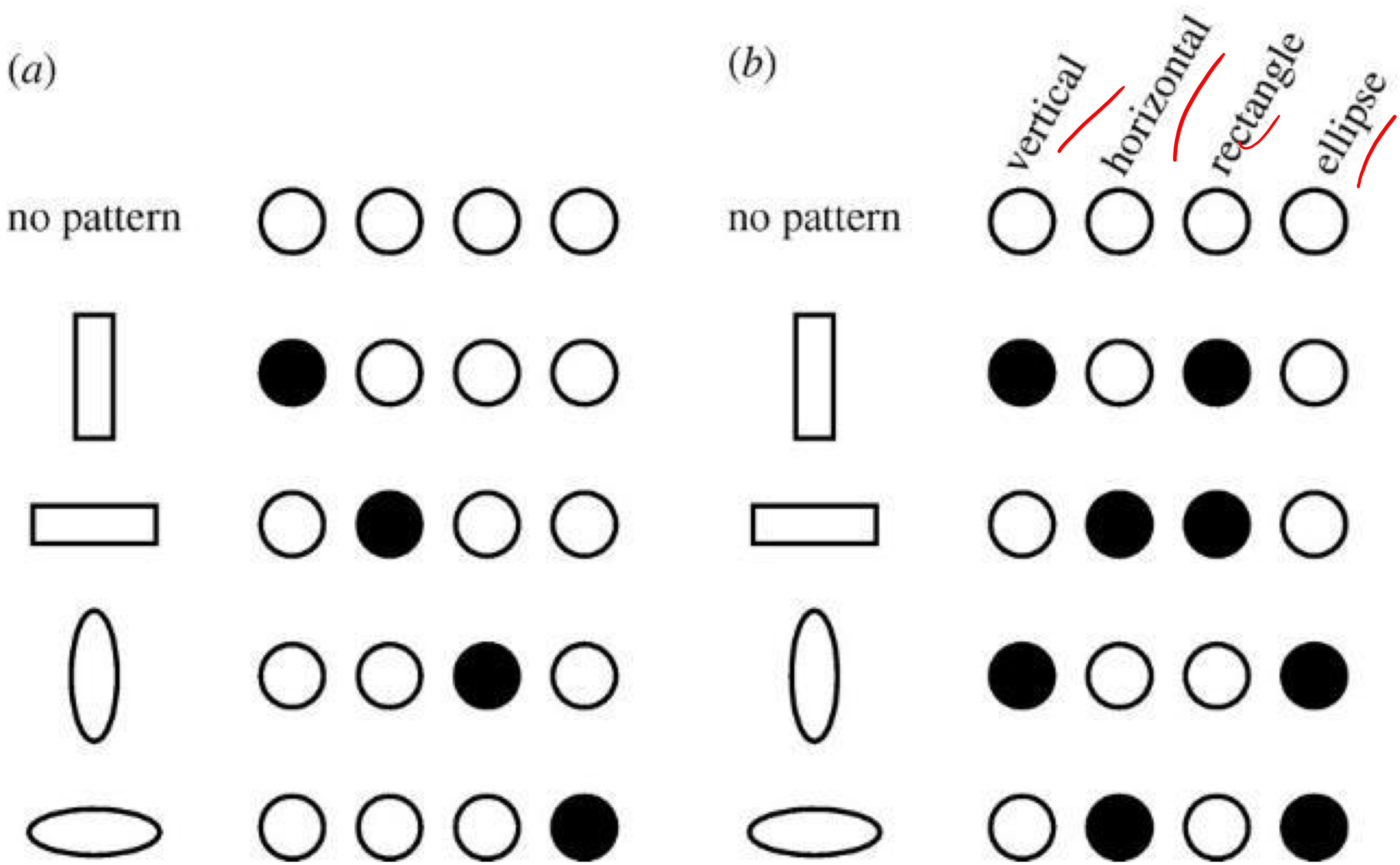
Distributed Representations Toy Example

- Local vs Distributed



Distributed Representations Toy Example

- Can we interpret each dimension?



Power of distributed representations!

Local ● ● ○ ● = VR + HR + HE = ?

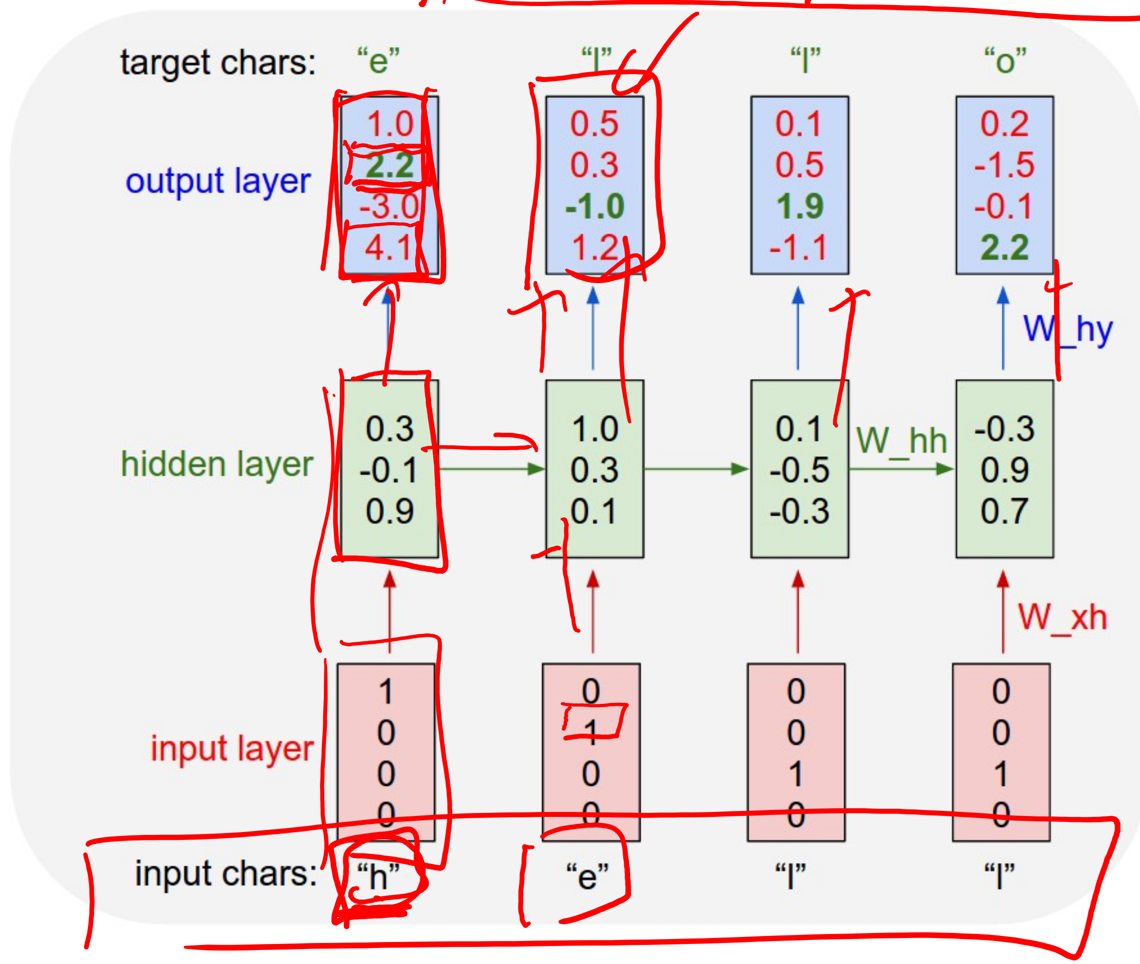
Distributed ● ● ○ ● = V + H + E ≈ ○

$$\max_{w} \log P(x_1 \ldots x_S \mid w) = \sum_t \log P(x_t \mid x_1 \ldots x_{t-1}, w)$$

Example: Character-level Language Model

Vocabulary: [h,e,l,o]

Example training sequence: **"hello"**

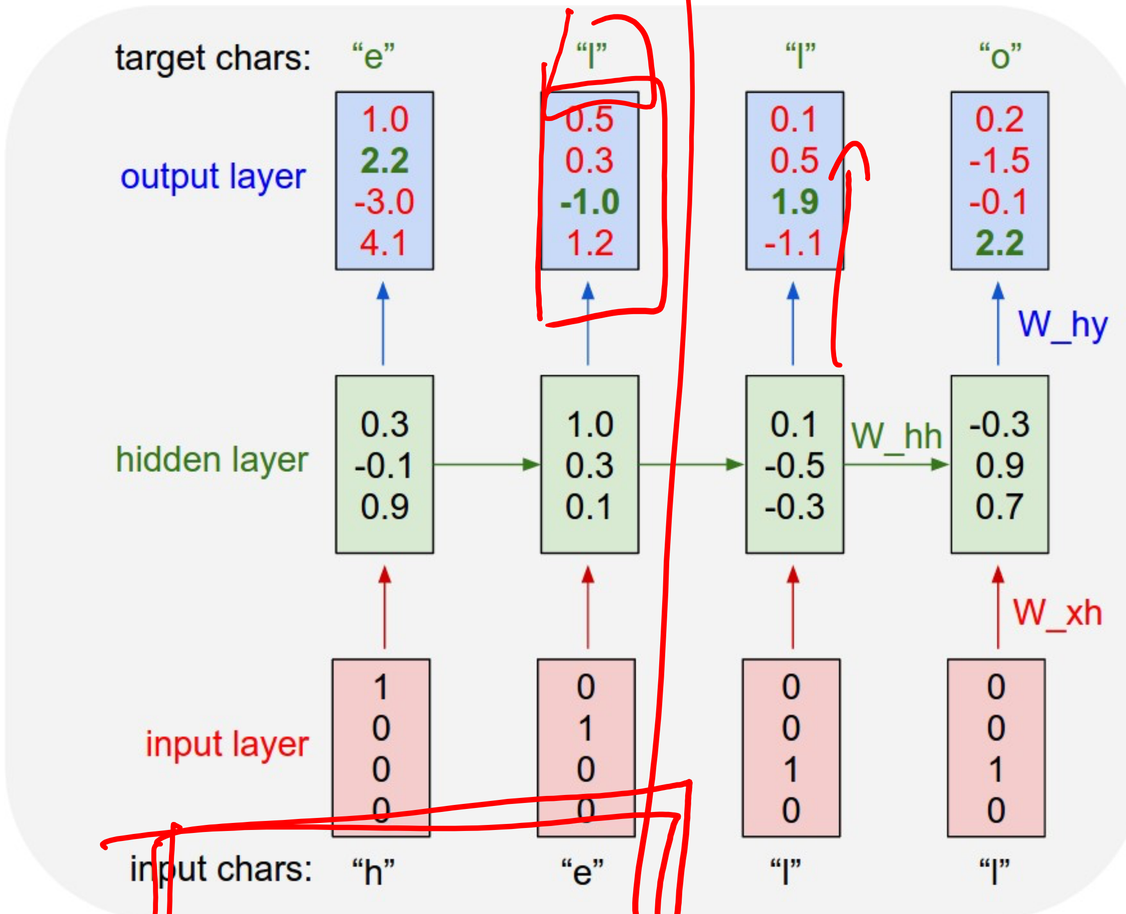


Training Time: MLE / "Teacher Forcing"

Example: Character-level Language Model

Vocabulary: [h,e,l,o]

Example training sequence: **"hello"**



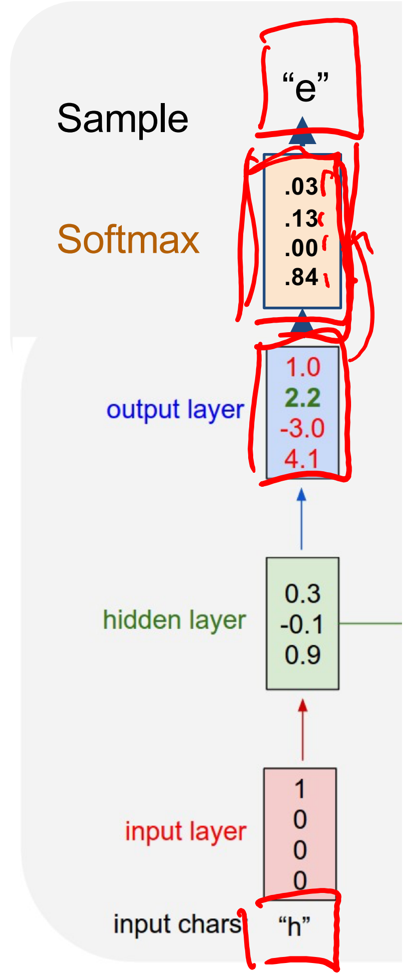
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Test Time: Sample / Argmax / Beam Search

Example: Character-level Language Model Sampling

Vocabulary: [h,e,l,o]

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

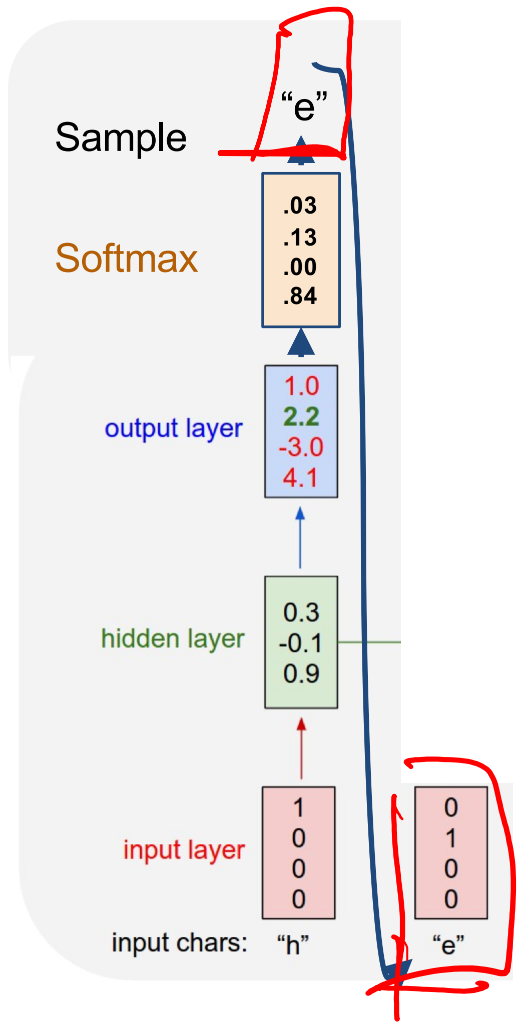


Test Time: Sample / Argmax / Beam Search

Example:
Character-level
Language Model
Sampling

Vocabulary:
[h,e,l,o]

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

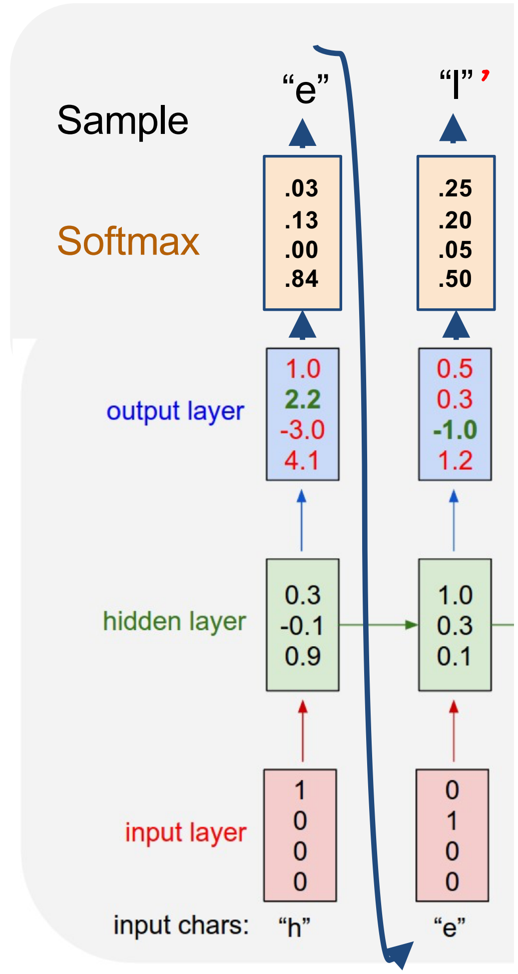


Test Time: Sample / Argmax / Beam Search

Example: Character-level Language Model Sampling

Vocabulary: [h,e,l,o]

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



Test Time: Sample / Argmax / Beam Search

Example: Character-level Language Model Sampling

Vocabulary: [h,e,l,o]

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