Deep Learning

Chapter 5 Recurrent Neural Network

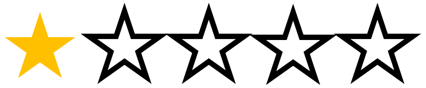
Dr. Van-Toi NGUYEN

EEE, Phenikaa University

Chapter 5: Recurrent Neural Network

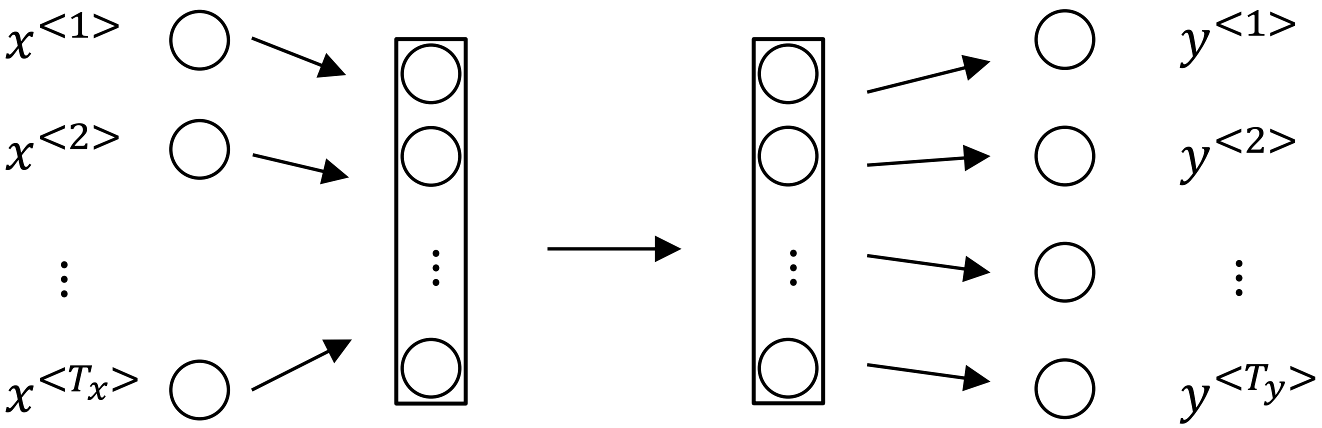
1. Unfolding Computational Graphs
2. Building a Recurrent Neural Networks
3. Long Short-Term Memory
4. Vision with Language Processing
5. Application of RNN
6. Practice

5.1 Unfolding Computational Graphs

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.

5.1 Unfolding Computational Graphs

Neural Network / CNN



Problems:

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

5.1 Unfolding Computational Graphs

Why existing convnets are insufficient?

Variable sequence length inputs and outputs!

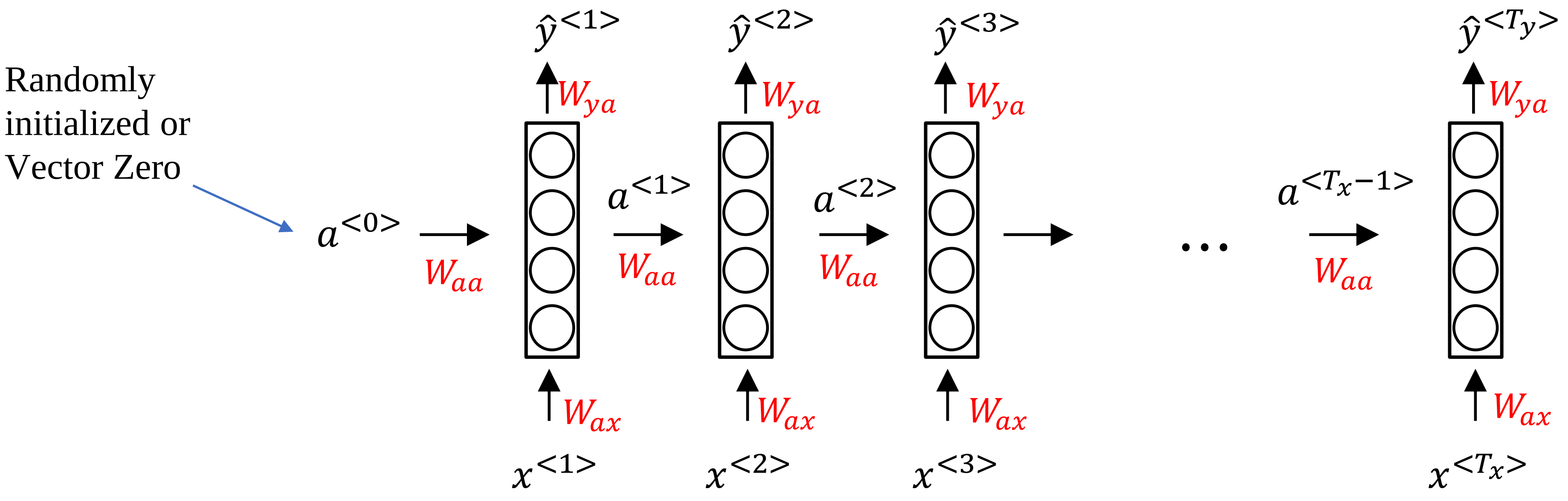
Example task: video captioning

Input video can have variable number of frames

Output captions can be variable length.

5.1 Unfolding Computational Graphs

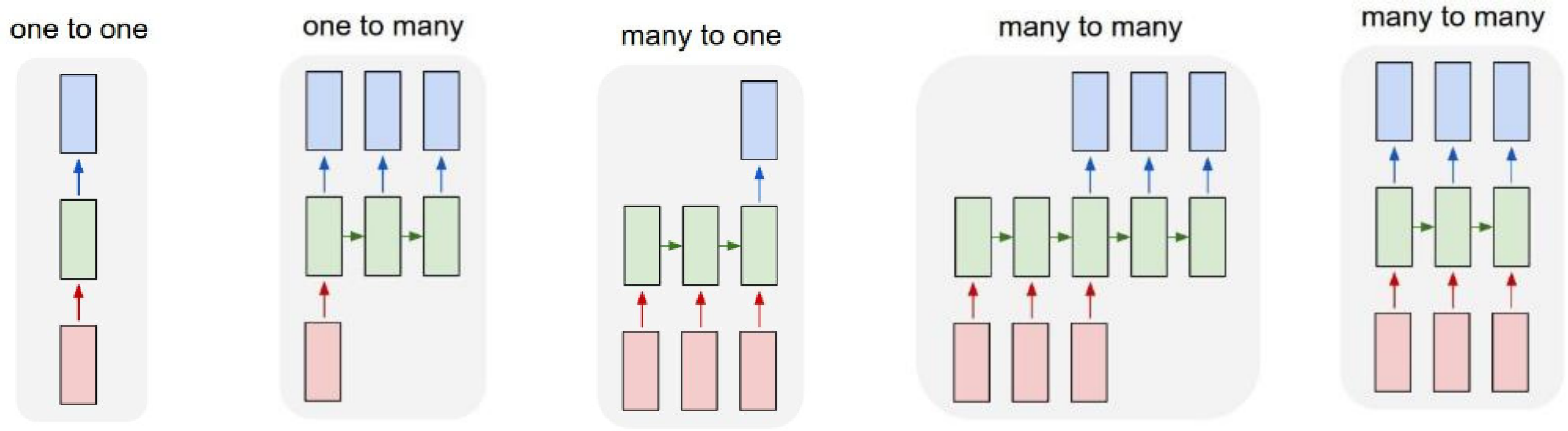
- **RNN uses information from the previous inputs**



He said, "Teddy Roosevelt was a great President."

He said, "Teddy bears are on sale!"

5.1 Unfolding Computational Graphs



Vanilla Neural Networks

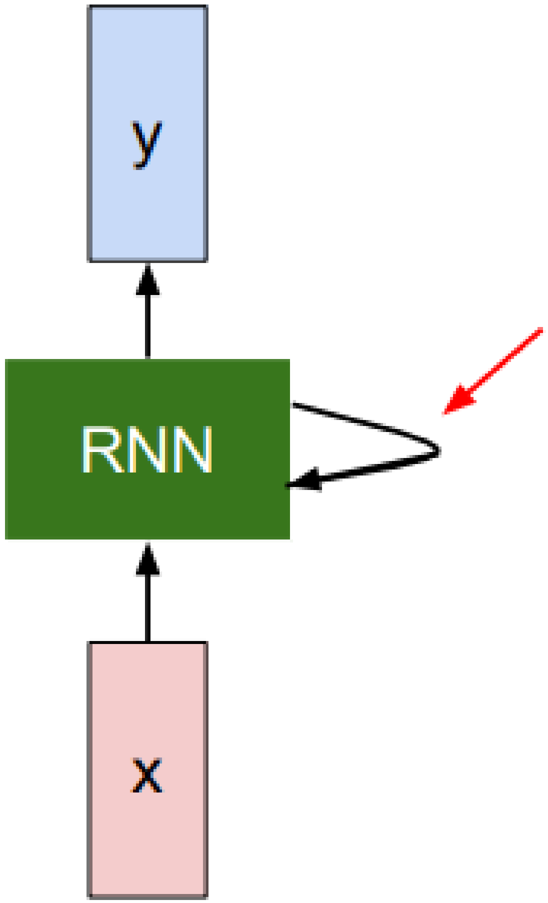
e.g. **Image Captioning**
image -> sequence of words

e.g. **action prediction**
sequence of video frames -> action class

E.g. **Video Captioning**
Sequence of video frames -> caption

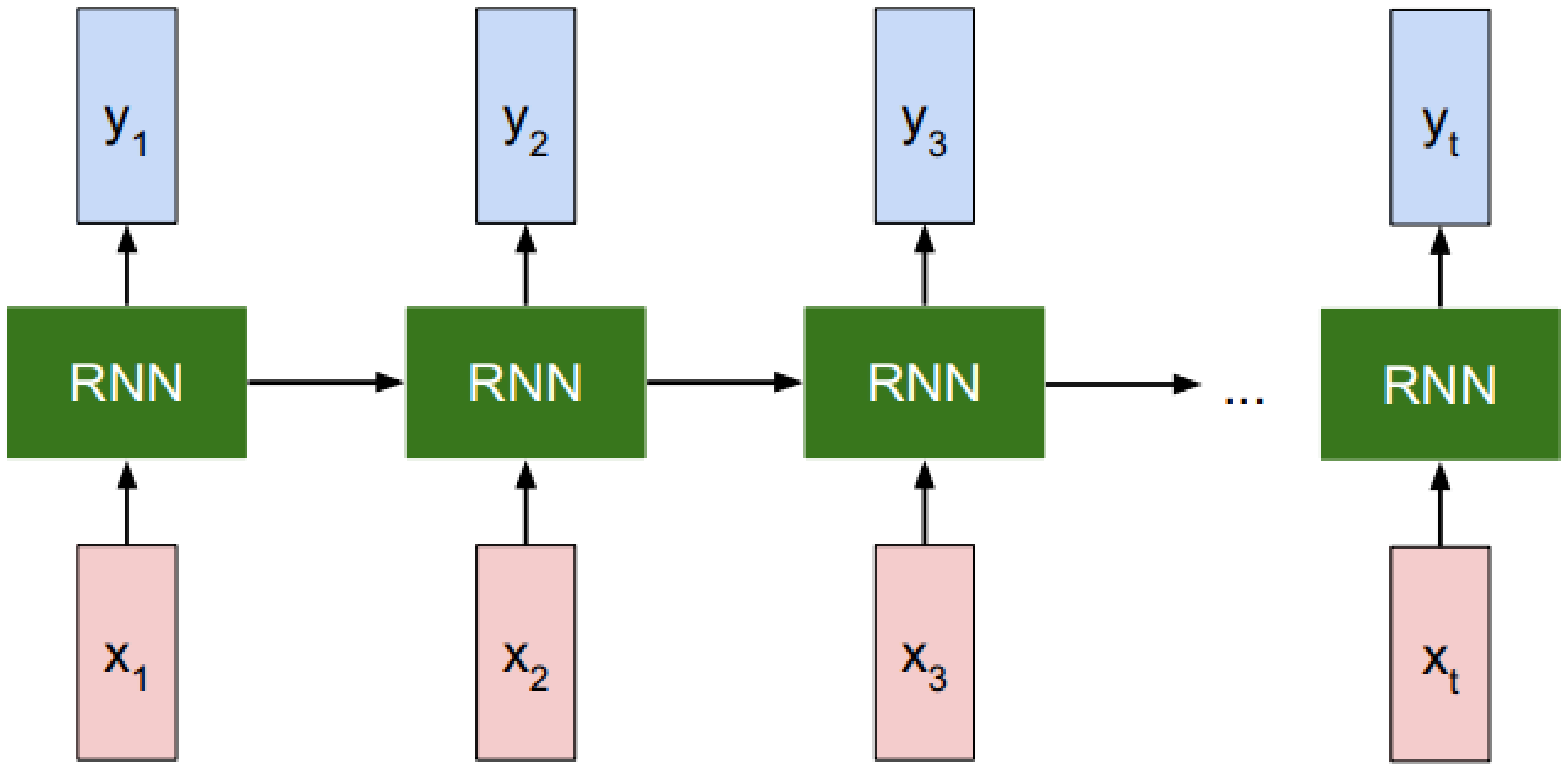
e.g. **Video classification on frame level**

5.1 Unfolding Computational Graphs

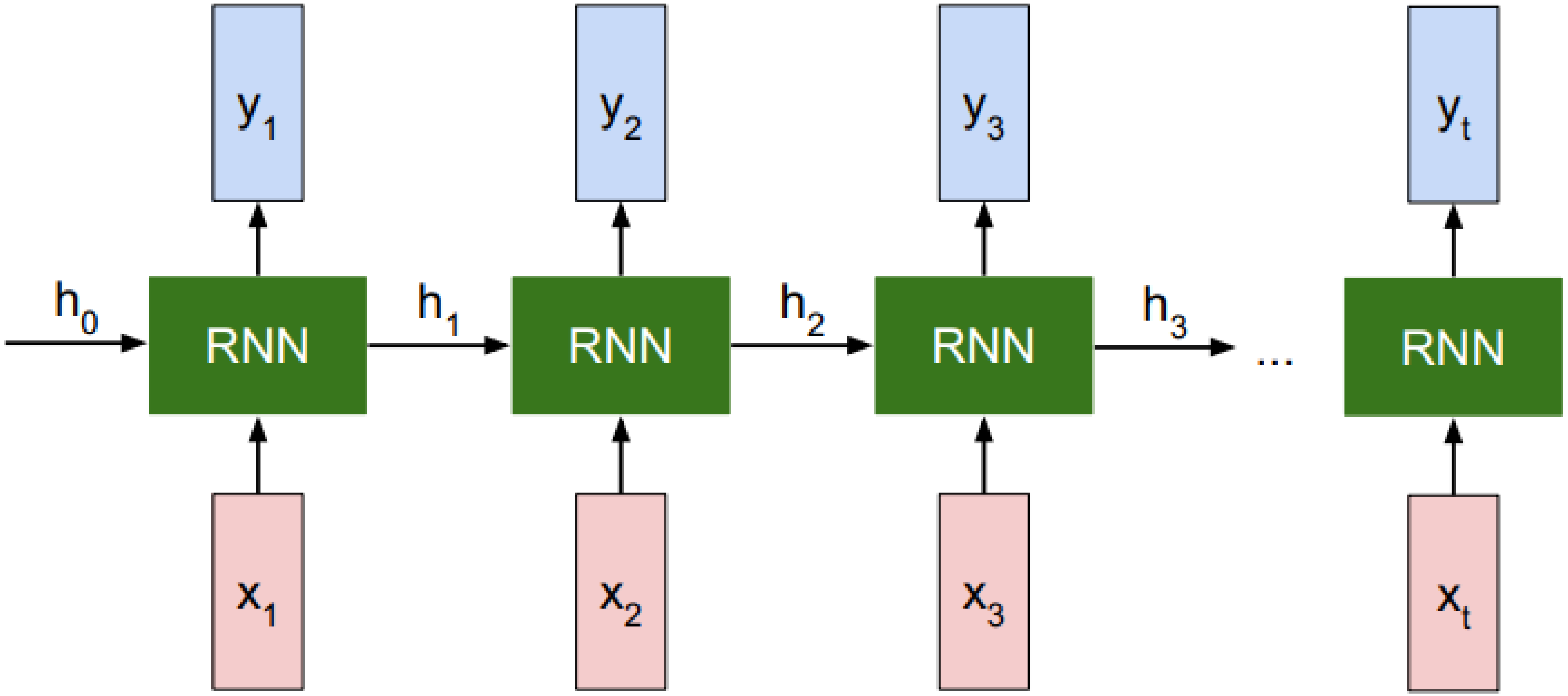


Key idea: RNNs have an "internal state" that is updated as a sequence is processed

5.1 Unfolding Computational Graphs



5.1 Unfolding Computational Graphs



Hidden state

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

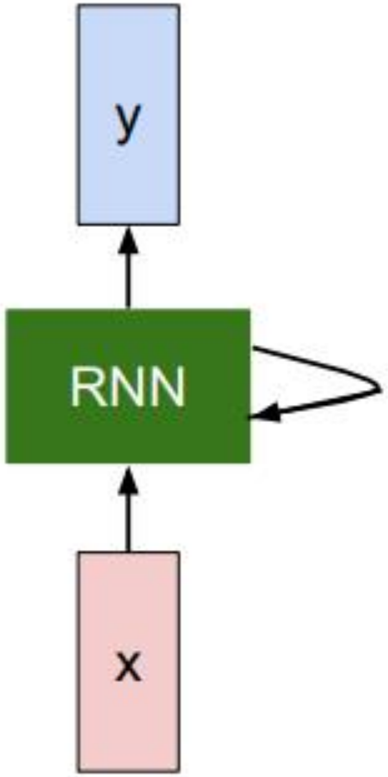
$$h_t = f_W(h_{t-1}, x_t)$$

h_t: new state

f_W: some function with parameters W

h_{t-1}: old state

x_t: input vector at some time step



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

Output

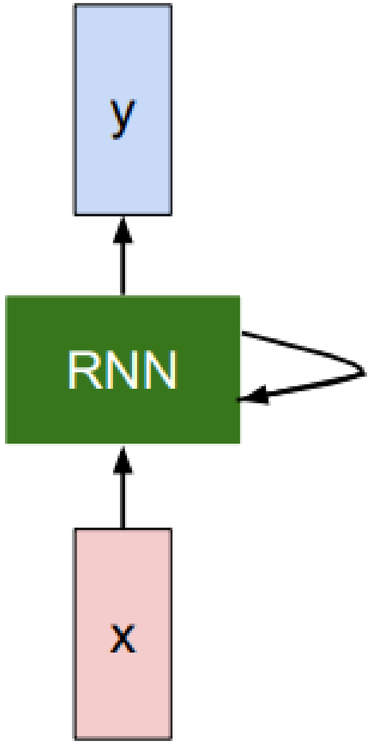
We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

$$y_t = f_{W_{hy}}(h_t)$$

output

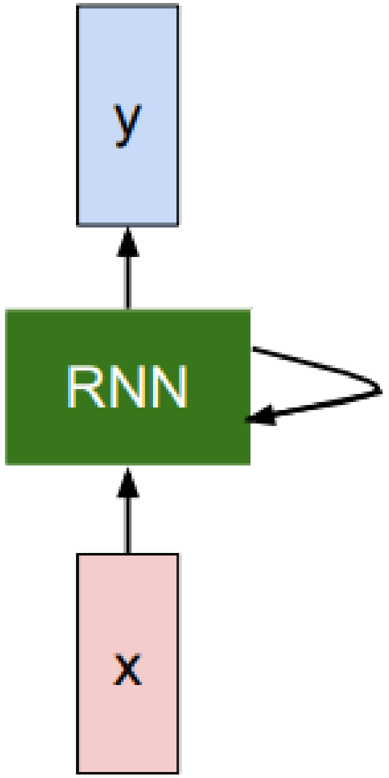
new state

another function with parameters W_o



(Simple) Recurrent Neural Network

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



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

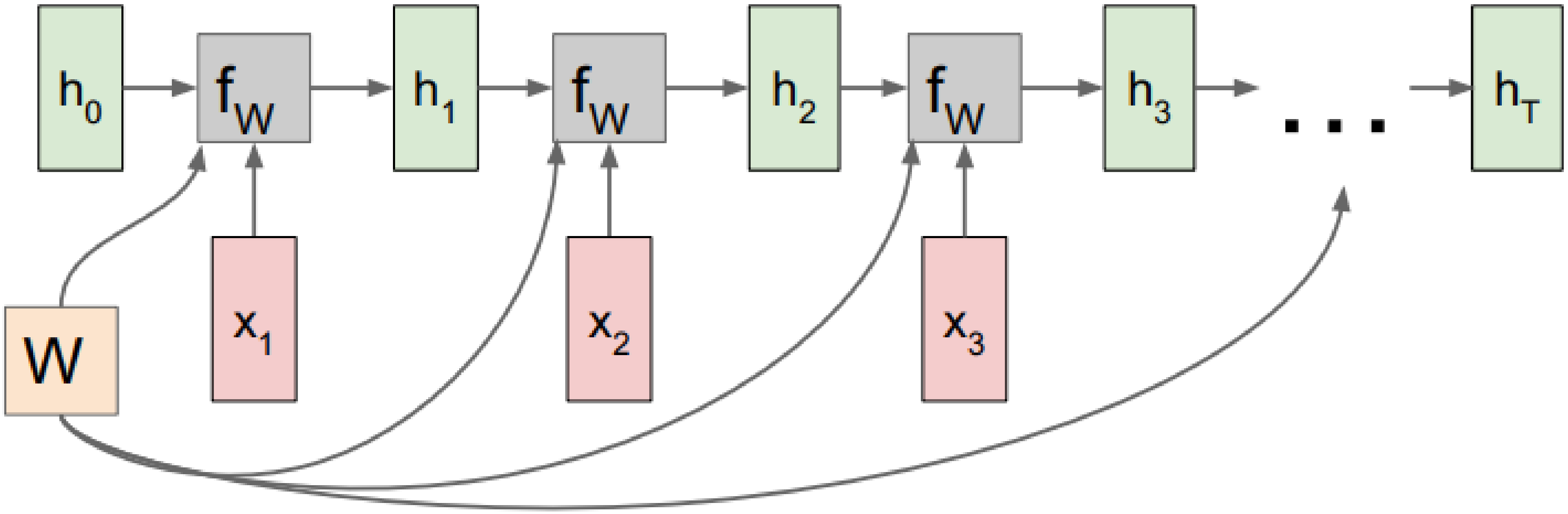
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

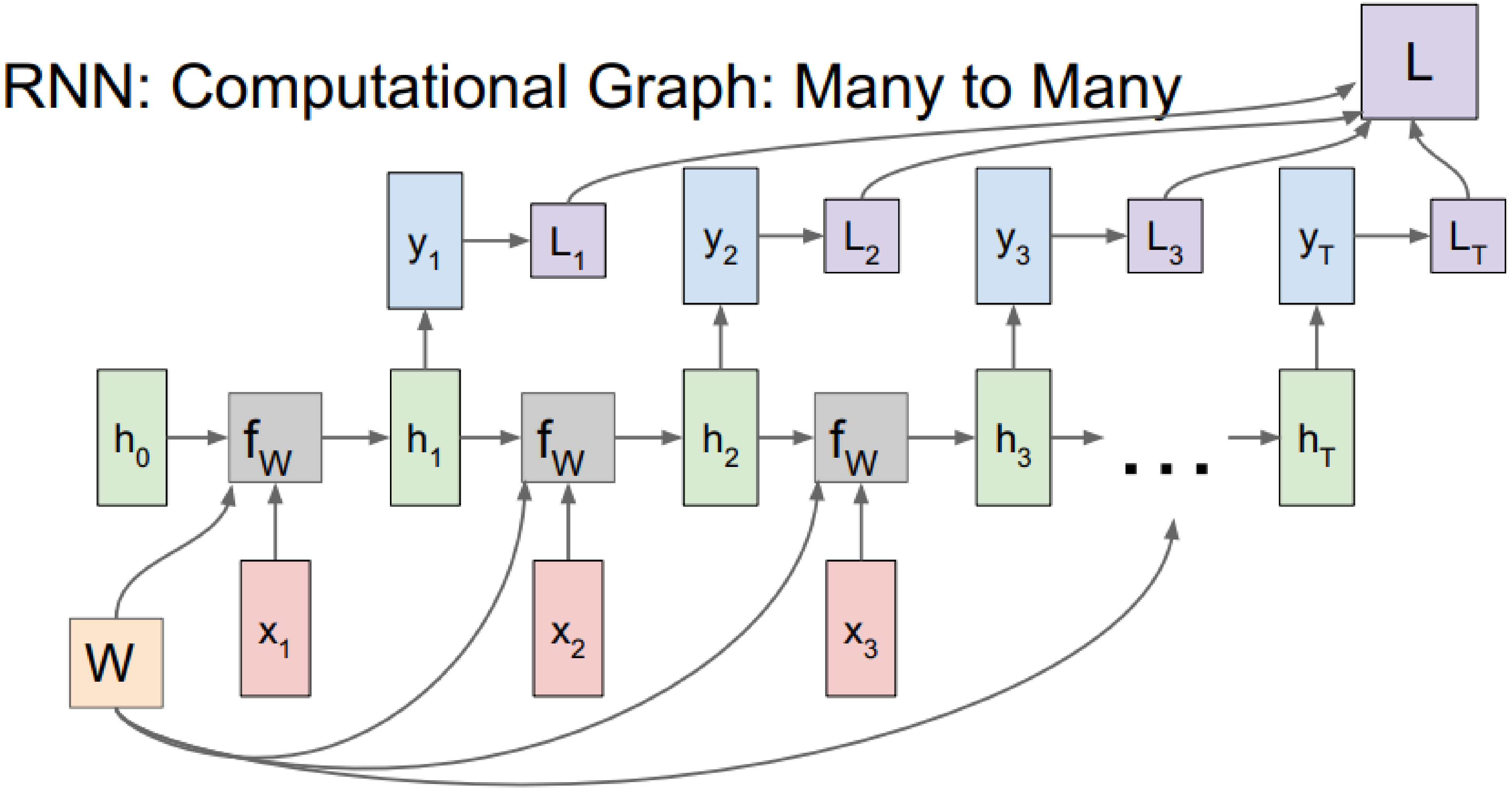
Sometimes called a "Vanilla RNN" or an "Elman RNN" after Prof. Jeffrey Elman

RNN: Computational Graph

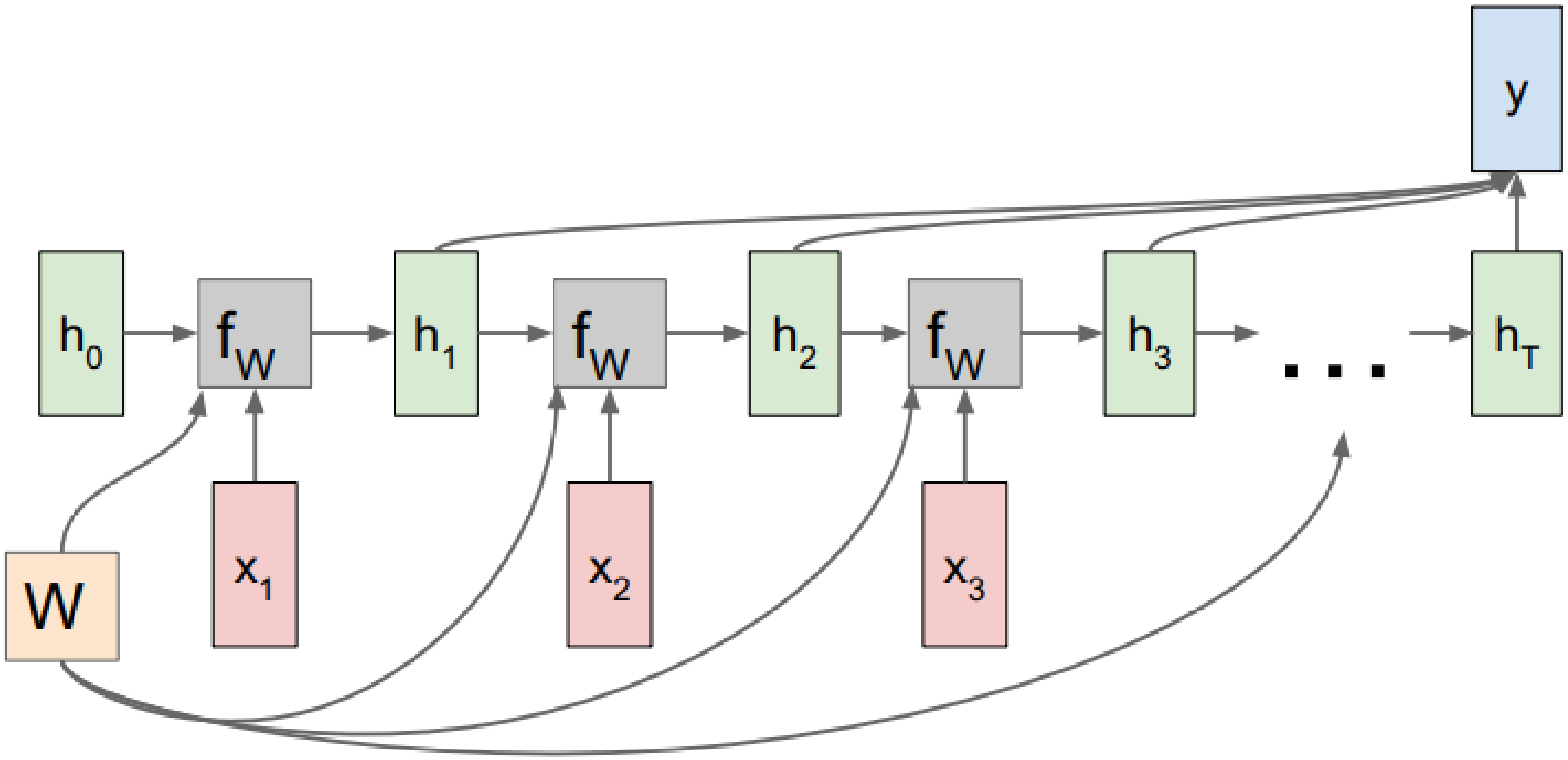
Re-use the same weight matrix at every time-step



5.2 Building Recurrent Neural Networks

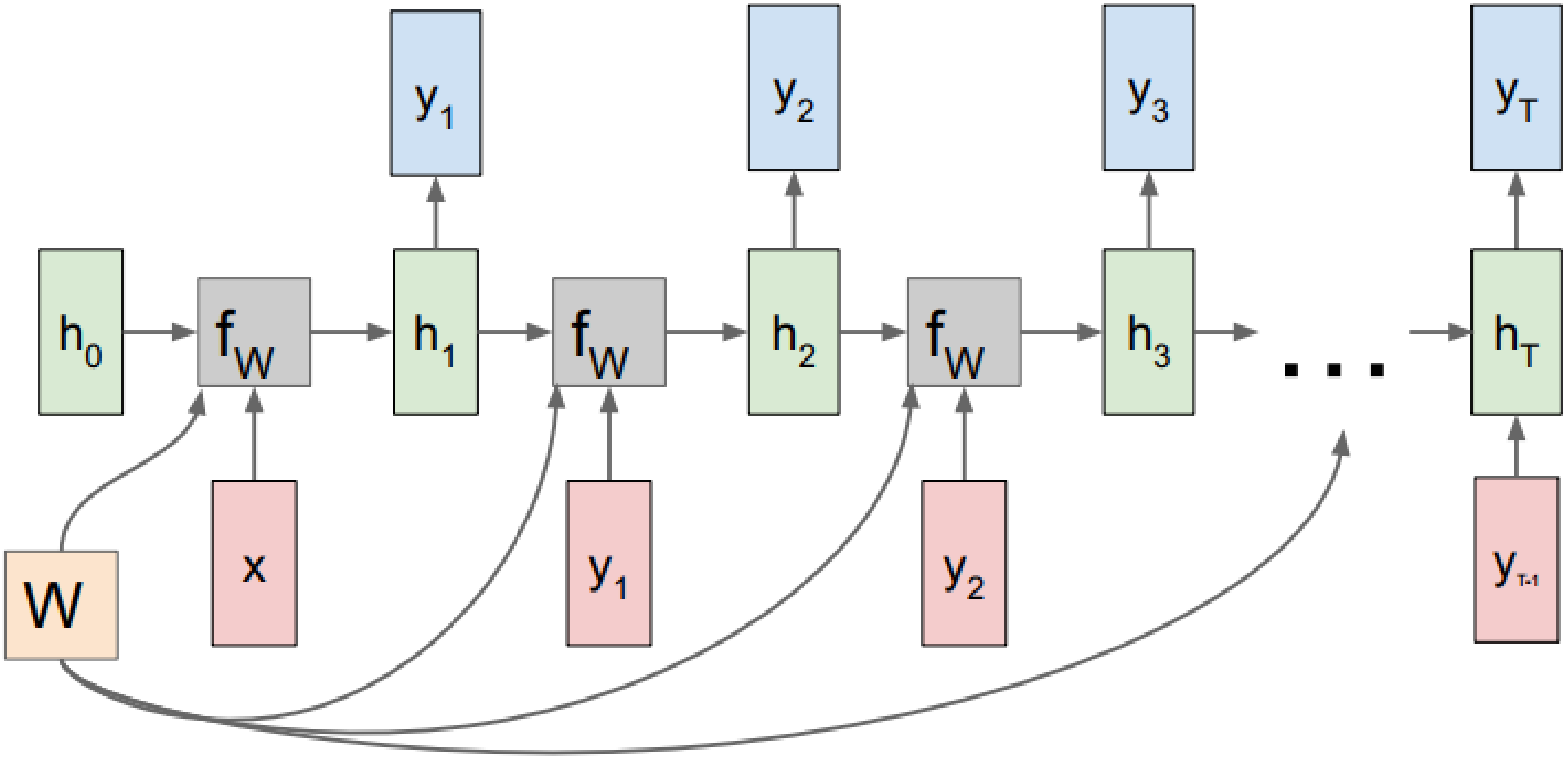


RNN: Computational Graph: Many to One



5.2 Building Recurrent Neural Networks

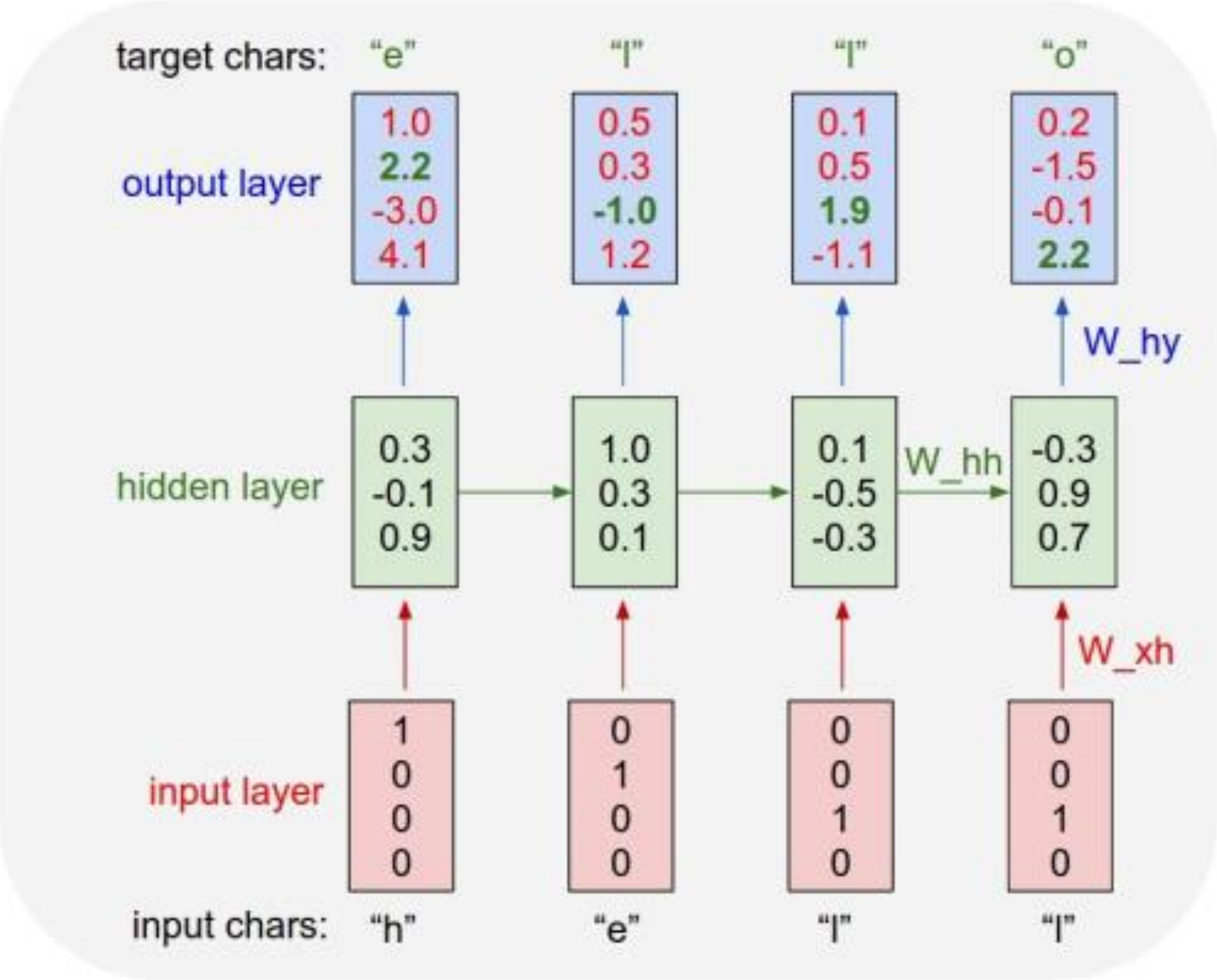
RNN: Computational Graph: One to Many



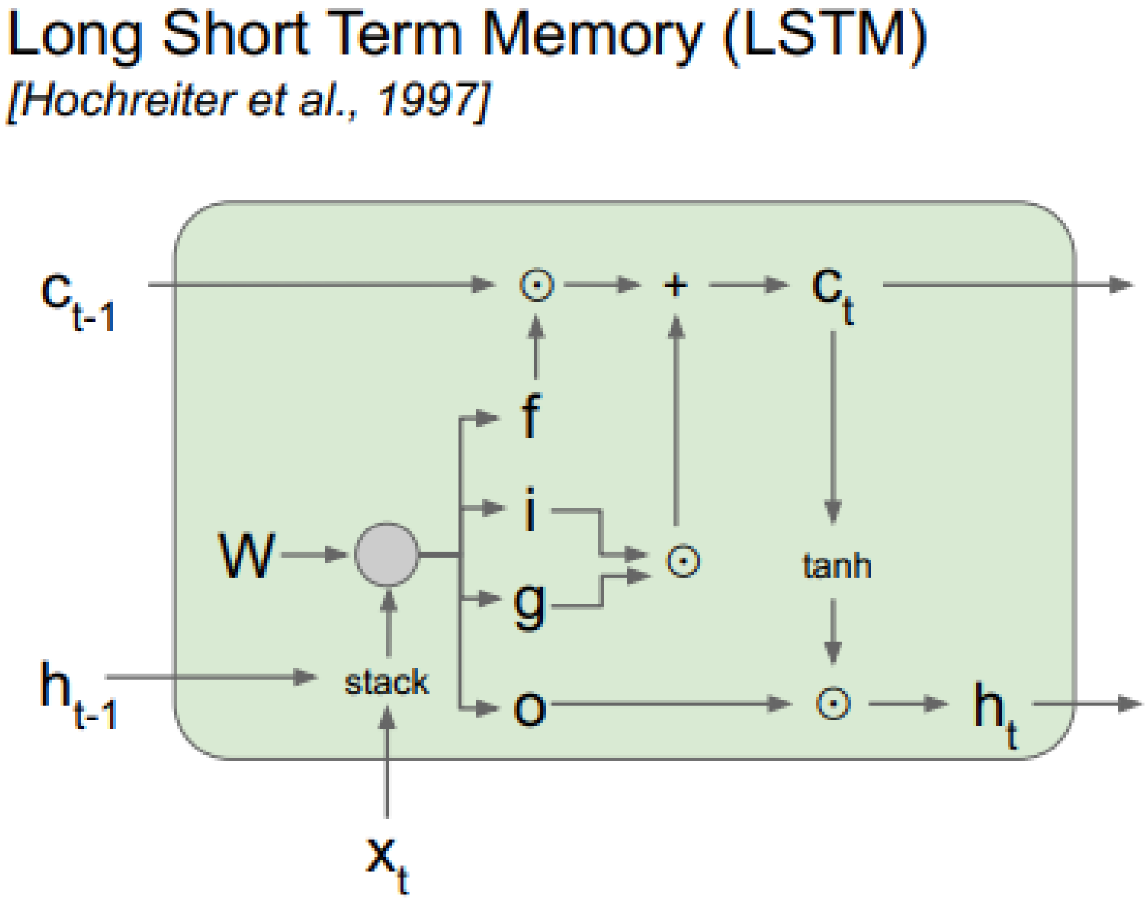
Example: Character-level Language Model

Vocabulary: [h,e,l,o]

Example training sequence: **“hello”**



5.3 Long Short-Term Memory (LSTM)



LSTM

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

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

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

Vanilla RNN

$$h_t = \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

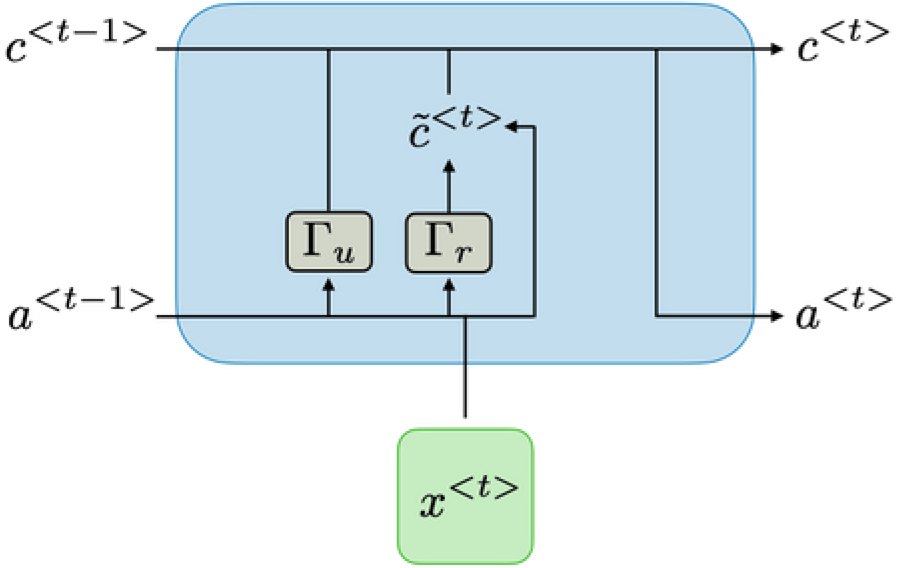
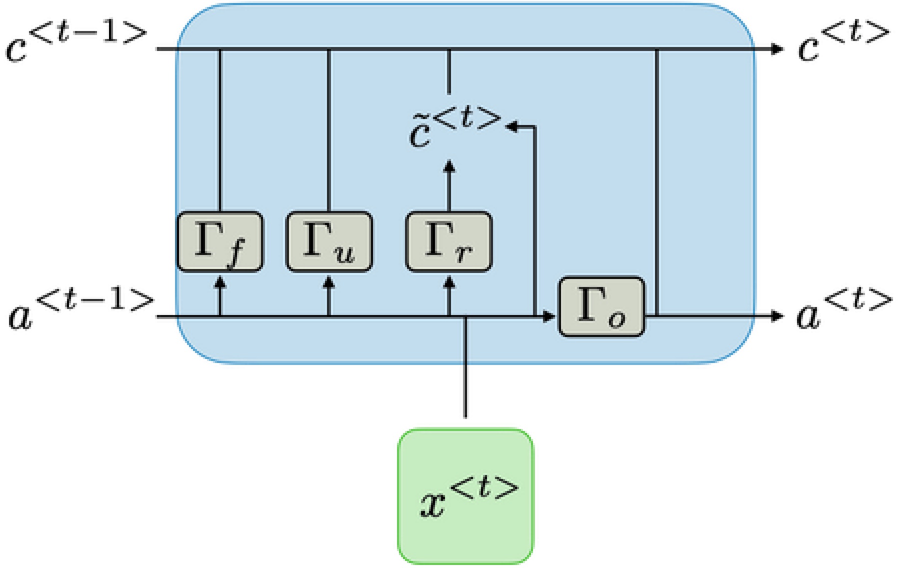
Other RNN Variants

GRU [*Learning phrase representations using rnn encoder-decoder for statistical machine translation*, Cho et al. 2014]

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$
$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$
$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h)$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

[*LSTM: A Search Space Odyssey*, Greff et al., 2015]

5.3 Long Short-Term Memory (LSTM)

Characterization	**Gated Recurrent Unit** (GRU)	**Long Short-Term Memory** (LSTM)
$\tilde{c}^{<t>}$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$
$c^{<t>}$	$\Gamma_u \star \tilde{c}^{<t>} + (1 - \Gamma_u) \star c^{<t-1>}$	$\Gamma_u \star \tilde{c}^{<t>} + \Gamma_f \star c^{<t-1>}$
$a^{<t>}$	$c^{<t>}$	$\Gamma_o \star c^{<t>}$
Dependencies		

5.4 Long Short-Term Memory (LSTM)

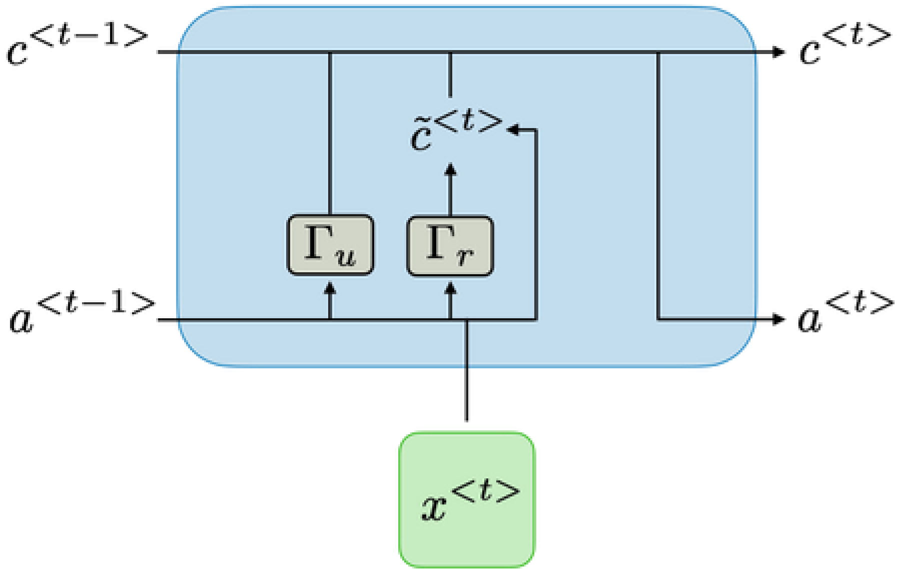
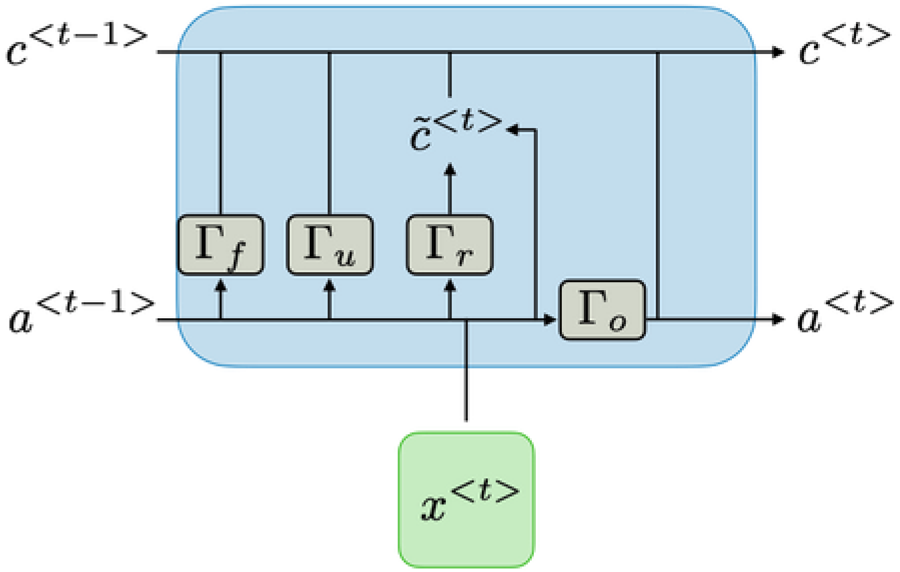
Characterization	**Gated Recurrent Unit** (GRU)	**Long Short-Term Memory** (LSTM)
$\tilde{c}^{<t>}$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$
$c^{<t>}$	$\Gamma_u \star \tilde{c}^{<t>} + (1 - \Gamma_u) \star c^{<t-1>}$	$\Gamma_u \star \tilde{c}^{<t>} + \Gamma_f \star c^{<t-1>}$
$a^{<t>}$	$c^{<t>}$	$\Gamma_o \star c^{<t>}$
Dependencies		

Image Captioning

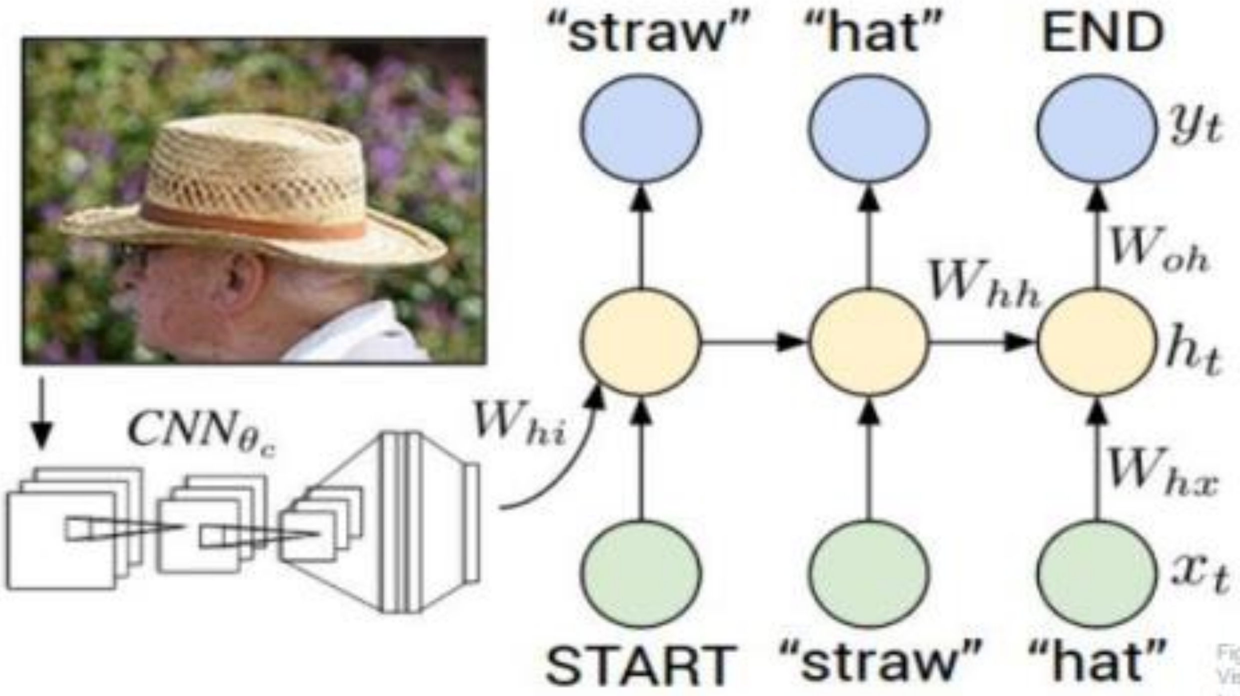


Figure from Karpathy et a, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015; figure copyright IEEE, 2015. Reproduced for educational purposes.

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

5.4 Vision with Language Processing

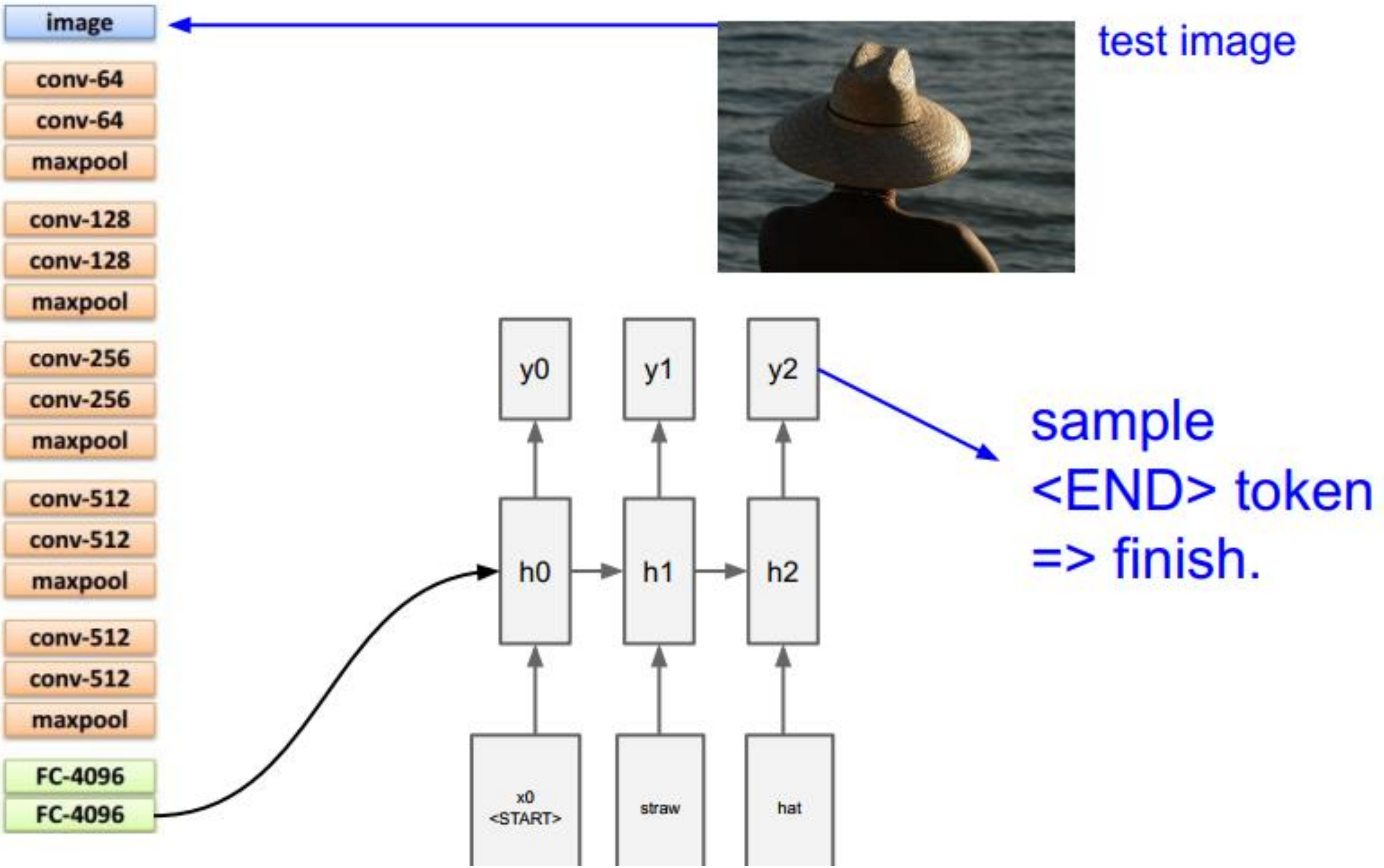


Image Captioning: Example Results

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

Two people walking on the beach with surfboards

A tennis player in action on the court

Two giraffes standing in a grassy field

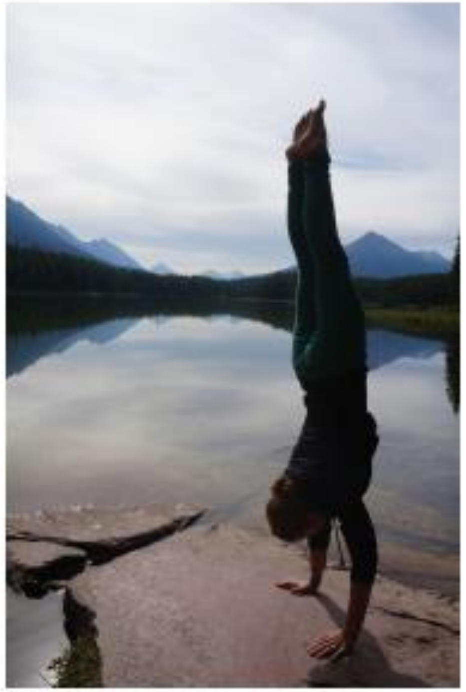
A man riding a dirt bike on a dirt track

Image Captioning: Failure Cases

Captions generated using neuraltalk2
All images are CC0 Public domain: fur coat, handstand, spider web, baseball

A woman is holding a cat in her hand

A person holding a computer mouse on a desk



A woman standing on a beach holding a surfboard

A bird is perched on a tree branch

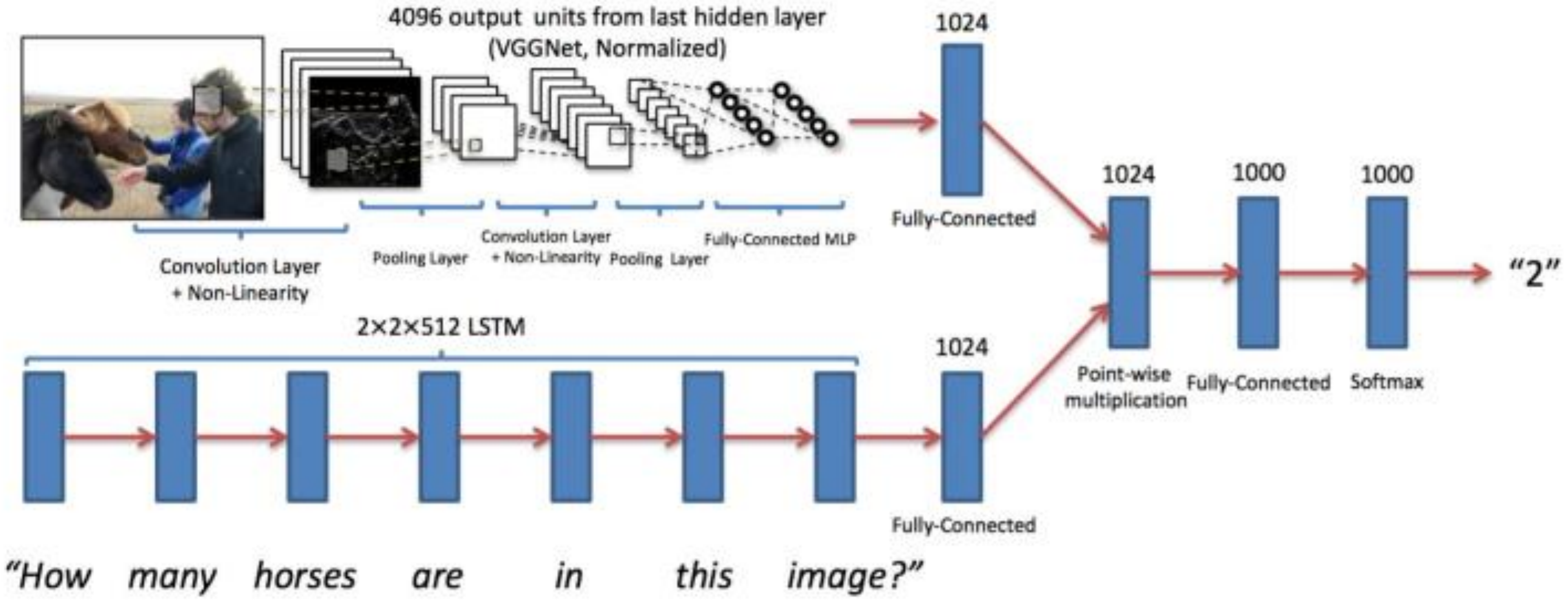
A man in a baseball uniform throwing a ball

Visual Question Answering (VQA)

Q: What endangered animal is featured on the truck?	Q: Where will the driver go if turning right?	Q: When was the picture taken?	Q: Who is under the umbrella?
A: **A bald eagle.**	A: **Onto 24 ¾ Rd.**	A: **During a wedding.**	A: **Two women.**
A: A sparrow.	A: Onto 25 ¾ Rd.	A: During a bar mitzvah.	A: A child.
A: A humming bird.	A: Onto 23 ¾ Rd.	A: During a funeral.	A: An old man.
A: A raven.	A: Onto Main Street.	A: During a Sunday church service	A: A husband and a wife.

Agrawal et al, "VQA: Visual Question Answering", ICCV 2015
Zhu et al, "Visual 7W: Grounded Question Answering in Images", CVPR 2016

Visual Question Answering: RNNs with Attention



Agrawal et al, "Visual 7W: Grounded Question Answering in Images", CVPR 2015

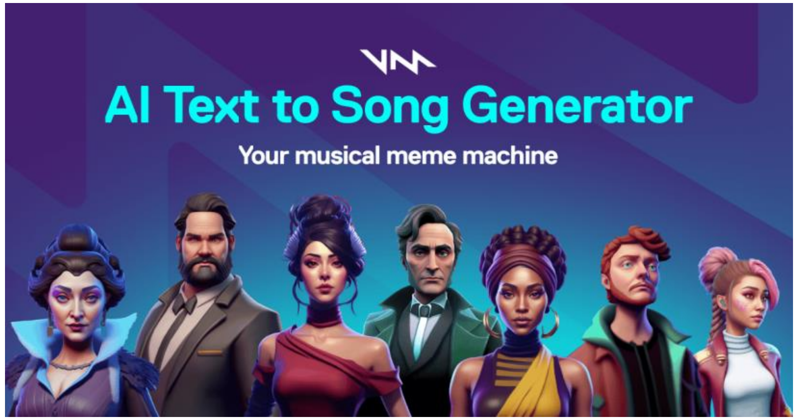
5.5 Application of RNN

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.

5.5 Application of RNN

ChatGPT

A dragon fruit wearing karate belt in the snow" and "*a photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach hat*"



Text2Image Generator

5.6 Practice

Stanford CS230 summary:

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

TensorFlow – RNN simple

https://www.tensorflow.org/guide/keras/rnn

Tensorflow – Time series forecasting

https://www.tensorflow.org/tutorials/structured_data/time_series