INF721

2024/2

Deep Learning

L13: Recurrent Neural Networks

Logistics

Announcements

- PA3 is due this Wednesday, 11:59pm

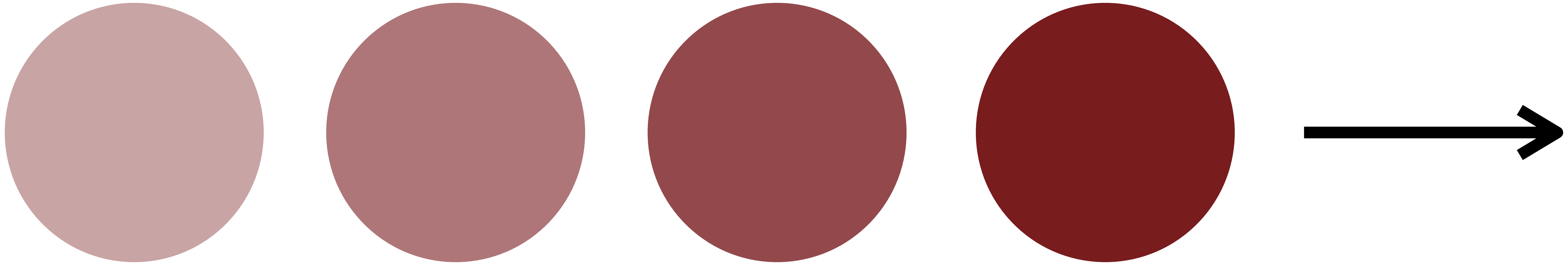
Last Lecture

- Input normalization
- Batch normalization
- Layer normalization

Lecture Outline

- Sequential Problems
- Recurrent Neural Networks (RNNs)
- Type of RNNs
- Backpropagation Through Time
- Language Models
- Exploding/Vanishing Gradients

What is the next position of this ball?



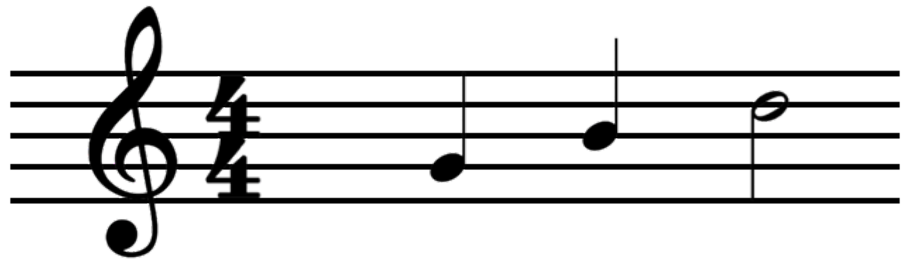
Recurrent Neural Networks are used for classification, regression or generation of sequential data!

What letter comes after T in the alphabet?

R S T U

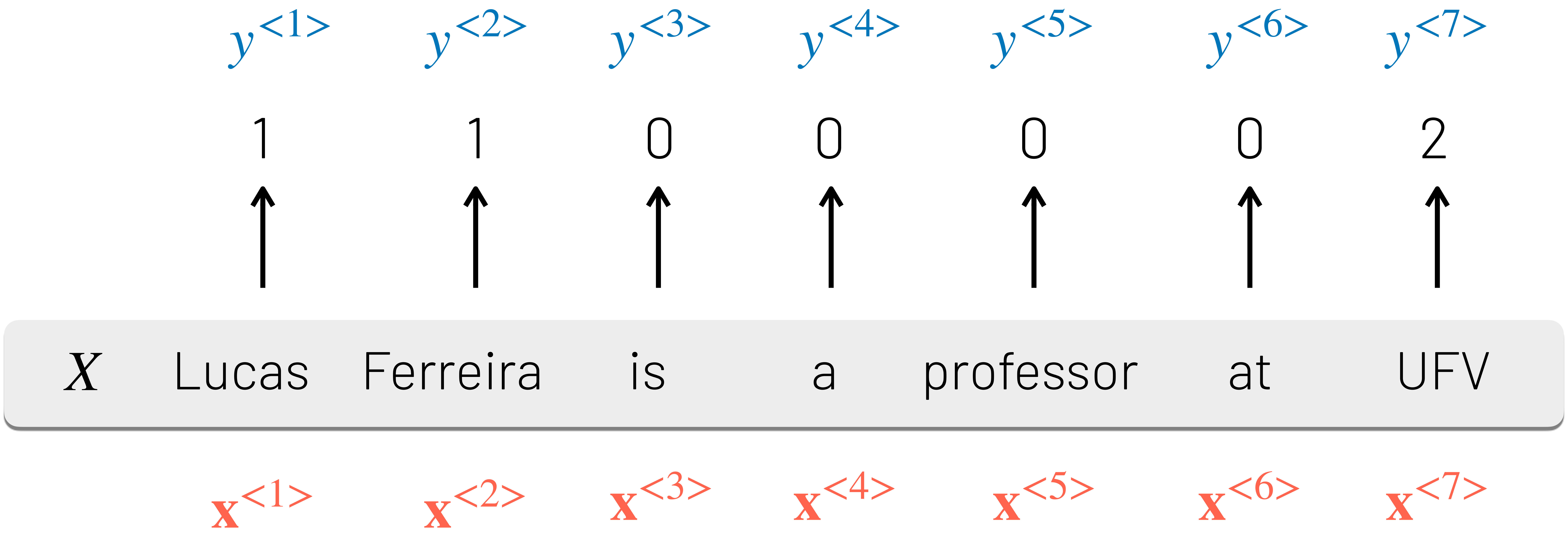
Recurrent Neural Networks are used for classification, regression or generation of sequential data!

Sequential Problems in Artificial Intelligence

	Input	Output
Speech Recognition		“Alexa, play The Beatles on Spotify”
Sentiment Analysis	“This is a terrible product.”	★☆☆☆☆
Machine Translation	“The book is on the table.”	“O livro está em cima da mesa.”
Image Captioning		“A cat lying by the window.“
Music Generation	None	
Named Entity Recognition	“Lucas Ferreira is a professor at UFV”	“Lucas Ferreira is a professor at UFV”

Example: Named Entity Recognition

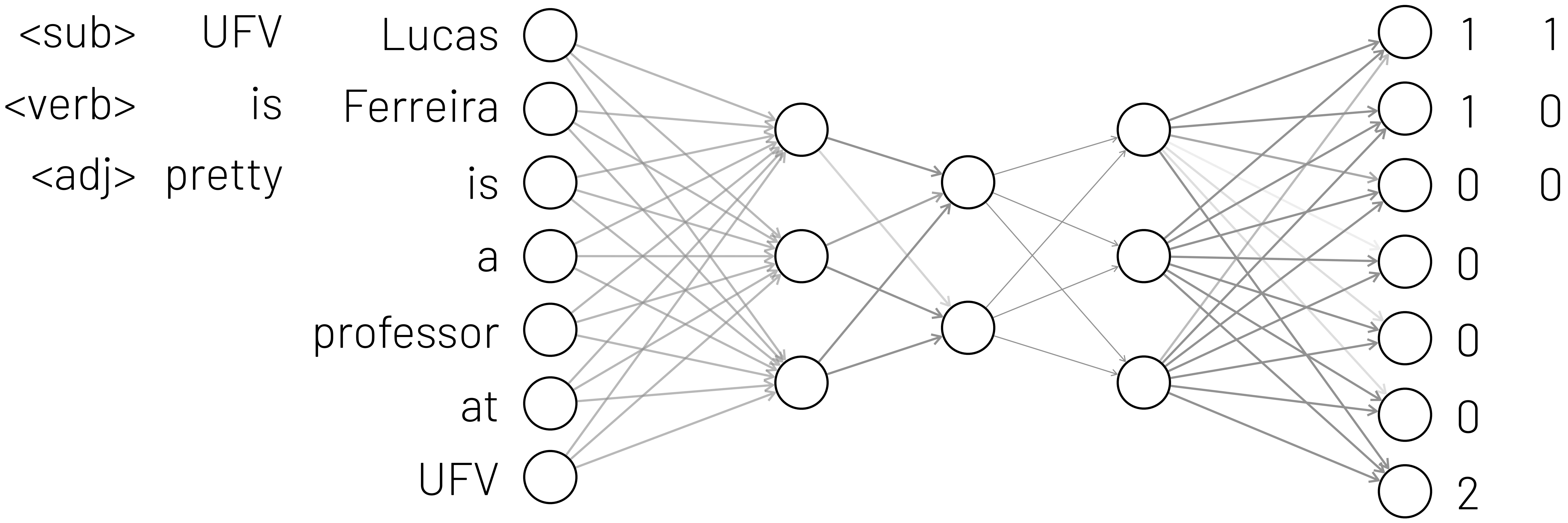
Locate and classify named entities mentioned in unstructured text:



	$y^{<1>}$	$y^{<2>}$	$y^{<3>}$	$y^{<4>}$	$y^{<5>}$	$y^{<6>}$	$y^{<7>}$
	1	1	0	0	0	0	2
X	Lucas	Ferreira	is	a	professor	at	UFV
	$\mathbf{x}^{<1>}$	$\mathbf{x}^{<2>}$	$\mathbf{x}^{<3>}$	$\mathbf{x}^{<4>}$	$\mathbf{x}^{<5>}$	$\mathbf{x}^{<6>}$	$\mathbf{x}^{<7>}$

In sequential problems, each input $\mathbf{x}^{<t>}$ can have an associated output $y^{<t>}$

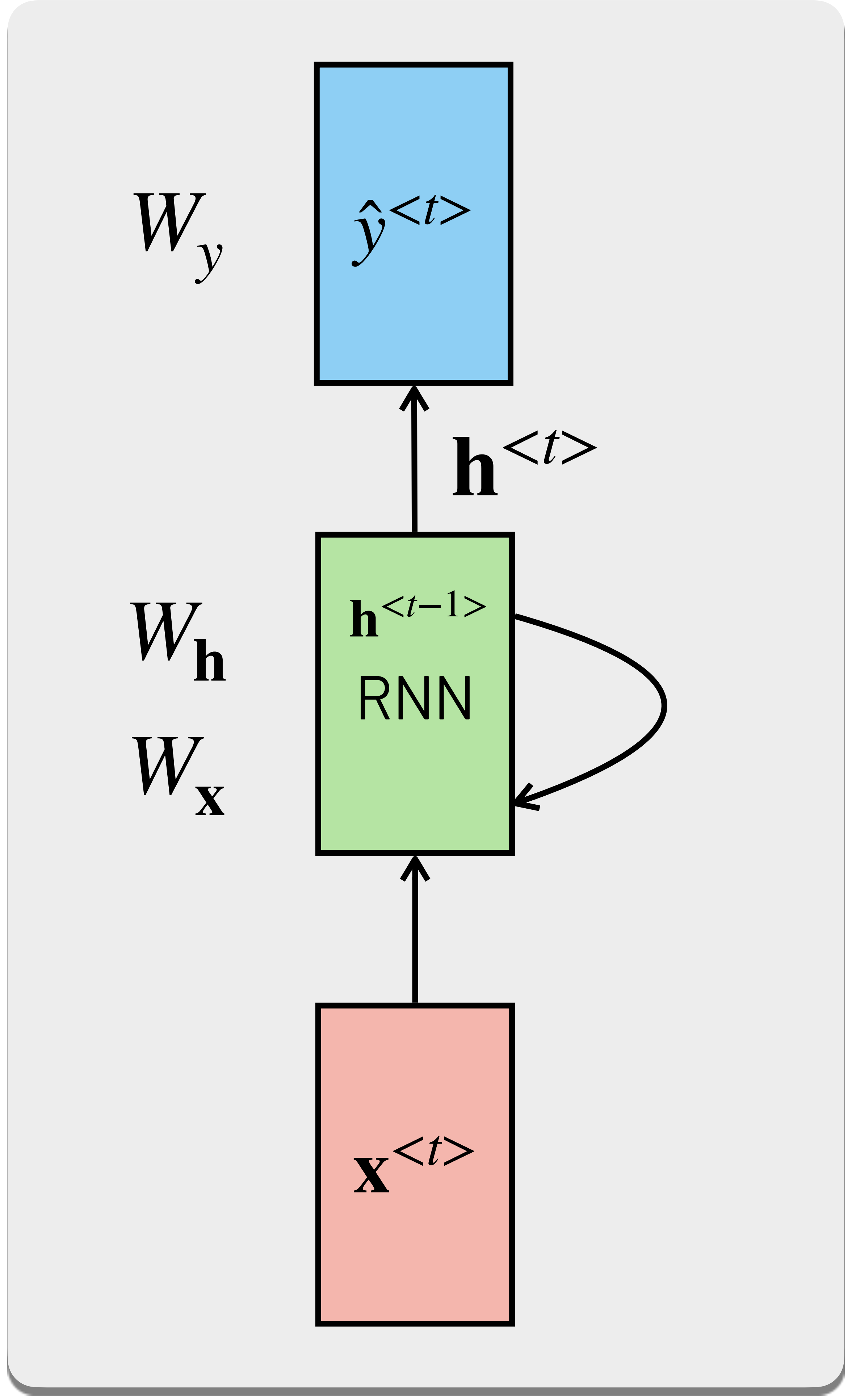
Why not MLPs for sequential problems?



Problem 1: Inputs and outputs may have different sizes in different examples.

Problem 2: MLPs do not capture temporal dependencies between elements of a sequence.

Recurrent Neural Networks (RNNs)



RNNs process each input element $\mathbf{x}^{<t>}$ at a time, keeping a state (vector) $\mathbf{h}^{<t>}$ that is updated at each time step t to produce the output $y^{<t>}$

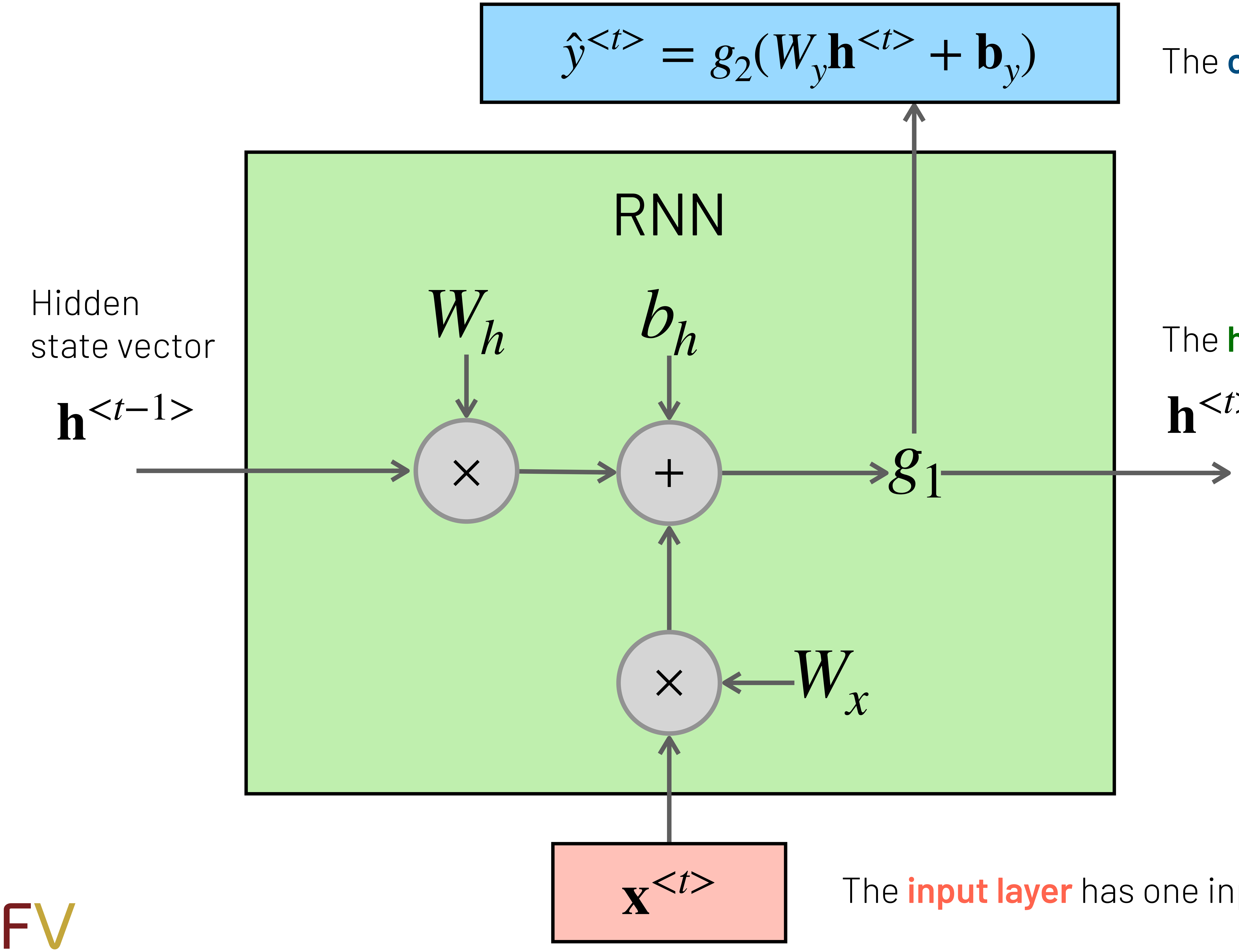
$$\mathbf{h}^{<t>} = g_1(W_h \mathbf{h}^{<t-1>} + W_\mathbf{x} \mathbf{x}^{<t>} + \mathbf{b}_h)$$

$$\hat{y}^{<t>} = g_2(W_y \mathbf{h}^{<t>} + \mathbf{b}_y)$$

- g_1: hidden layer activation function (tanh/relu)
- g_2: output layer activation function (sigmoid/softmax)

UFV

Recurrent Neural Networks (RNNs)



The **output layer** produces the output $y^{<t>}$

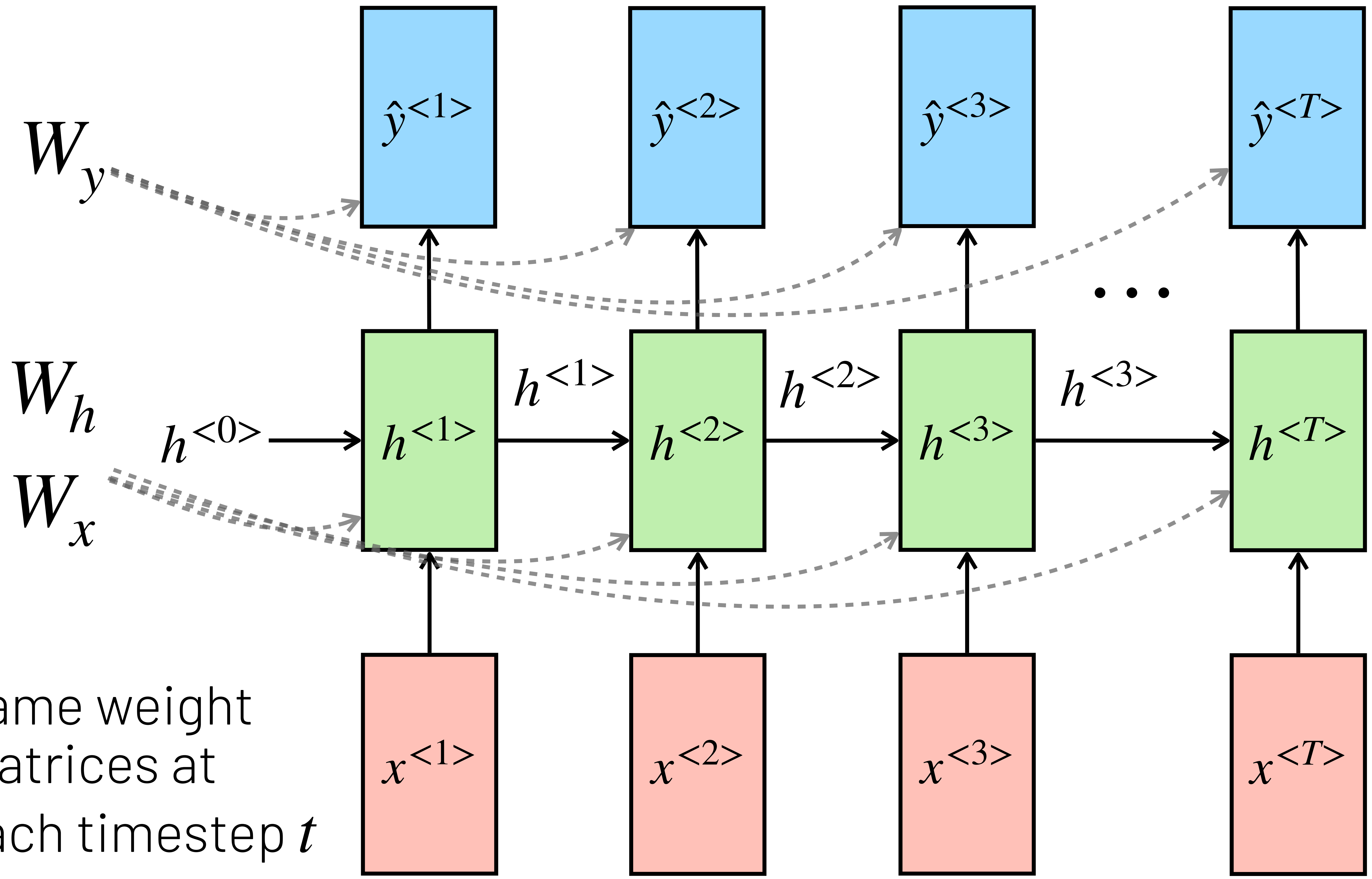
The **hidden layer** updates the hidden state $\mathbf{h}^{<t>}$

$$\mathbf{h}^{<t>} = g_1(W_h\mathbf{h}^{<t-1>} + W_x\mathbf{x}^{<t>} + \mathbf{b}_h)$$

The **input layer** has one input vector $\mathbf{x}^{<t>}$ for each timestep t

Recurrent Neural Networks (RNNs)

RNNs can be seen unrolled over a fixed number of timesteps T



Same weight matrices at each timestep t

$$h^{<1>} = g_1(W_h h^{<0>} + W_x x^{<1>} + b_h)$$
$$\hat{y}^{<1>} = g_2(W_y h^{<1>} + b_y)$$

$$h^{<2>} = g_1(W_h h^{<1>} + W_x x^{<2>} + b_h)$$
$$\hat{y}^{<2>} = g_2(W_y h^{<2>} + b_y)$$

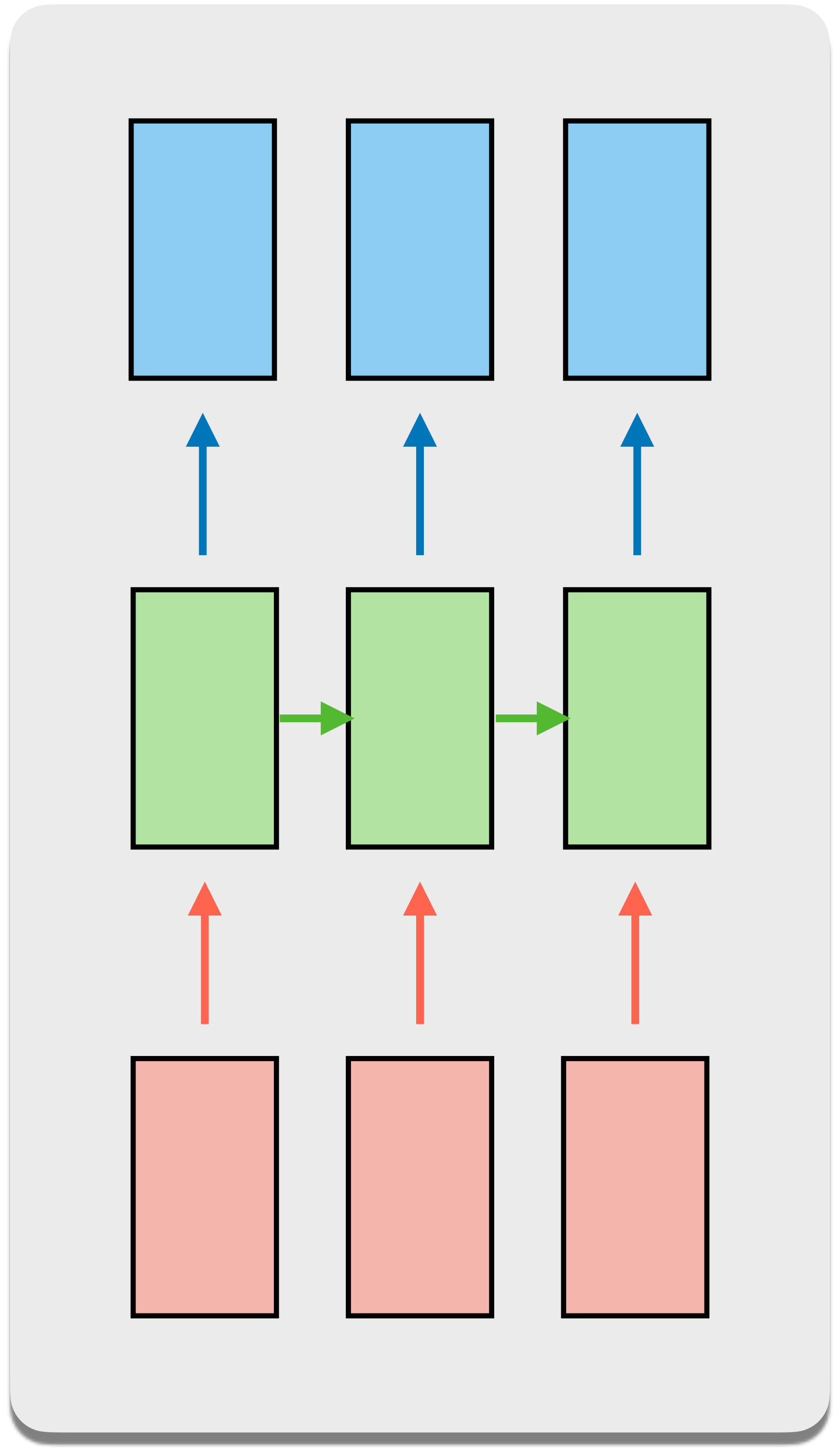
$$h^{<3>} = g_1(W_h h^{<2>} + W_x x^{<3>} + b_h)$$
$$\hat{y}^{<3>} = g_2(W_y h^{<3>} + b_y)$$

$$\vdots$$

$$h^{<T>} = g_1(W_h h^{<T-1>} + W_x x^{<T>} + b_h)$$
$$\hat{y}^{<T>} = g_2(W_y h^{<T>} + b_y)$$

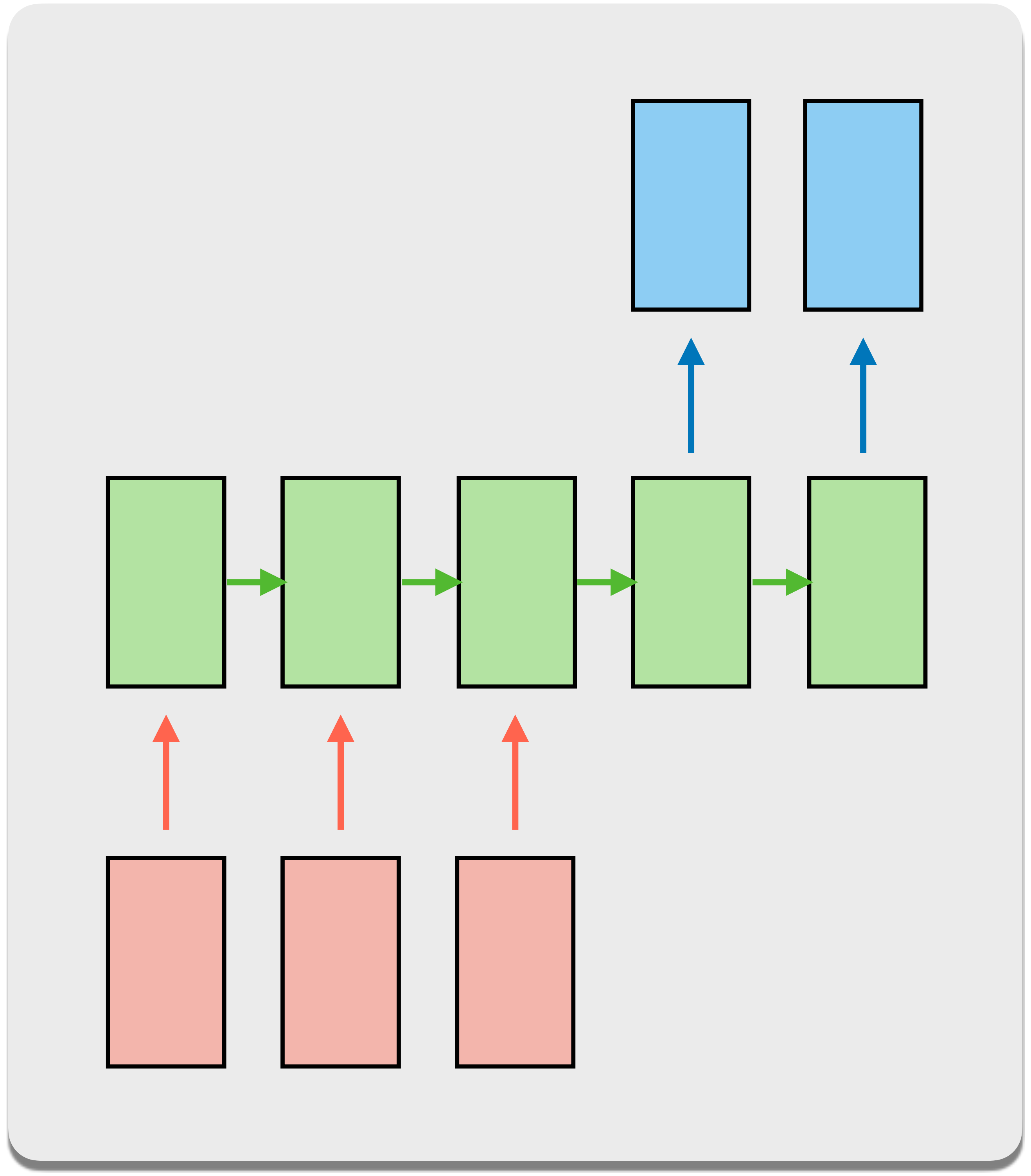
Types of RNNs

Many to **Many**



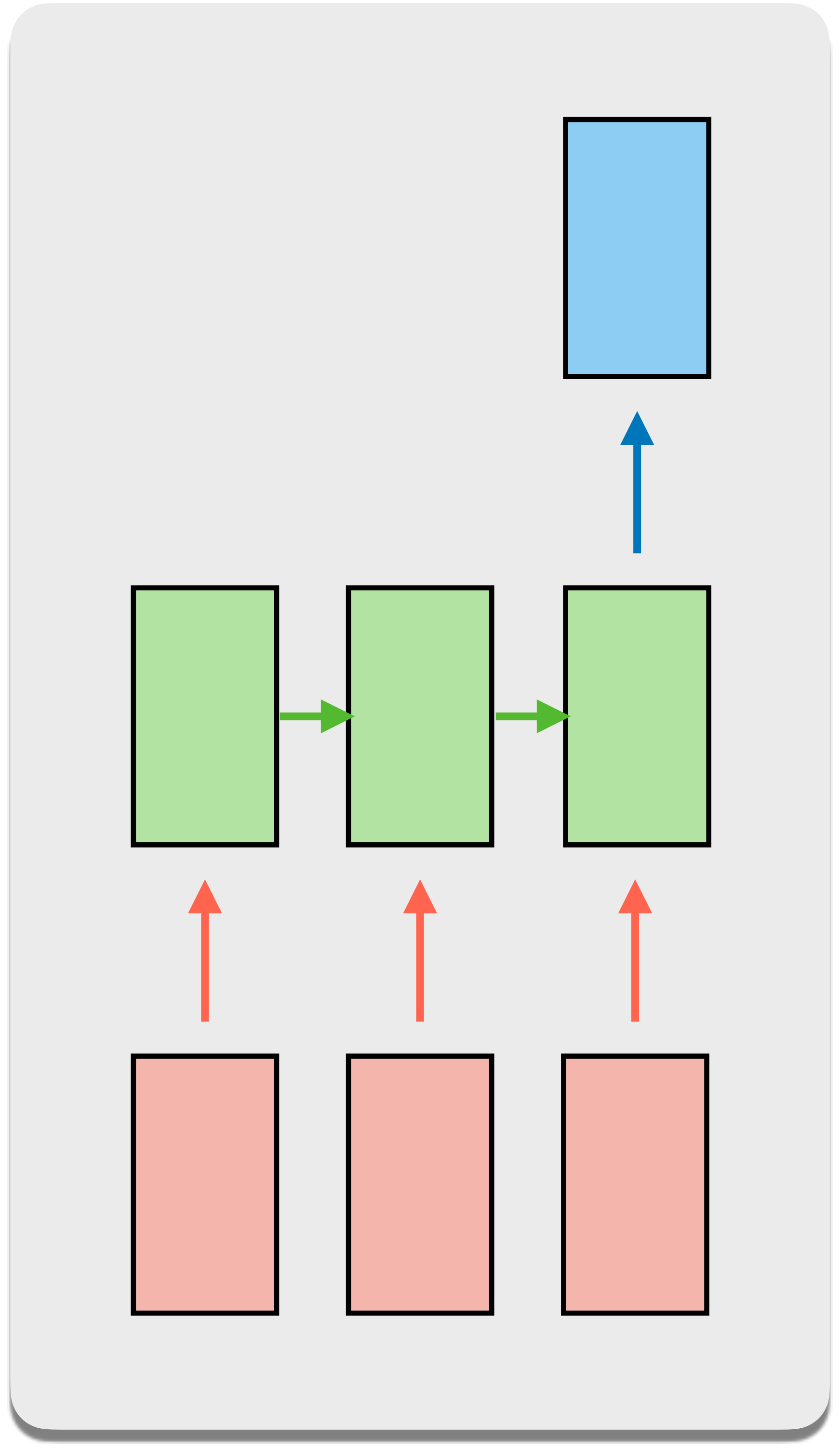
Example
Named Entity Recognition

Many to **Many (Seq2Seq)**



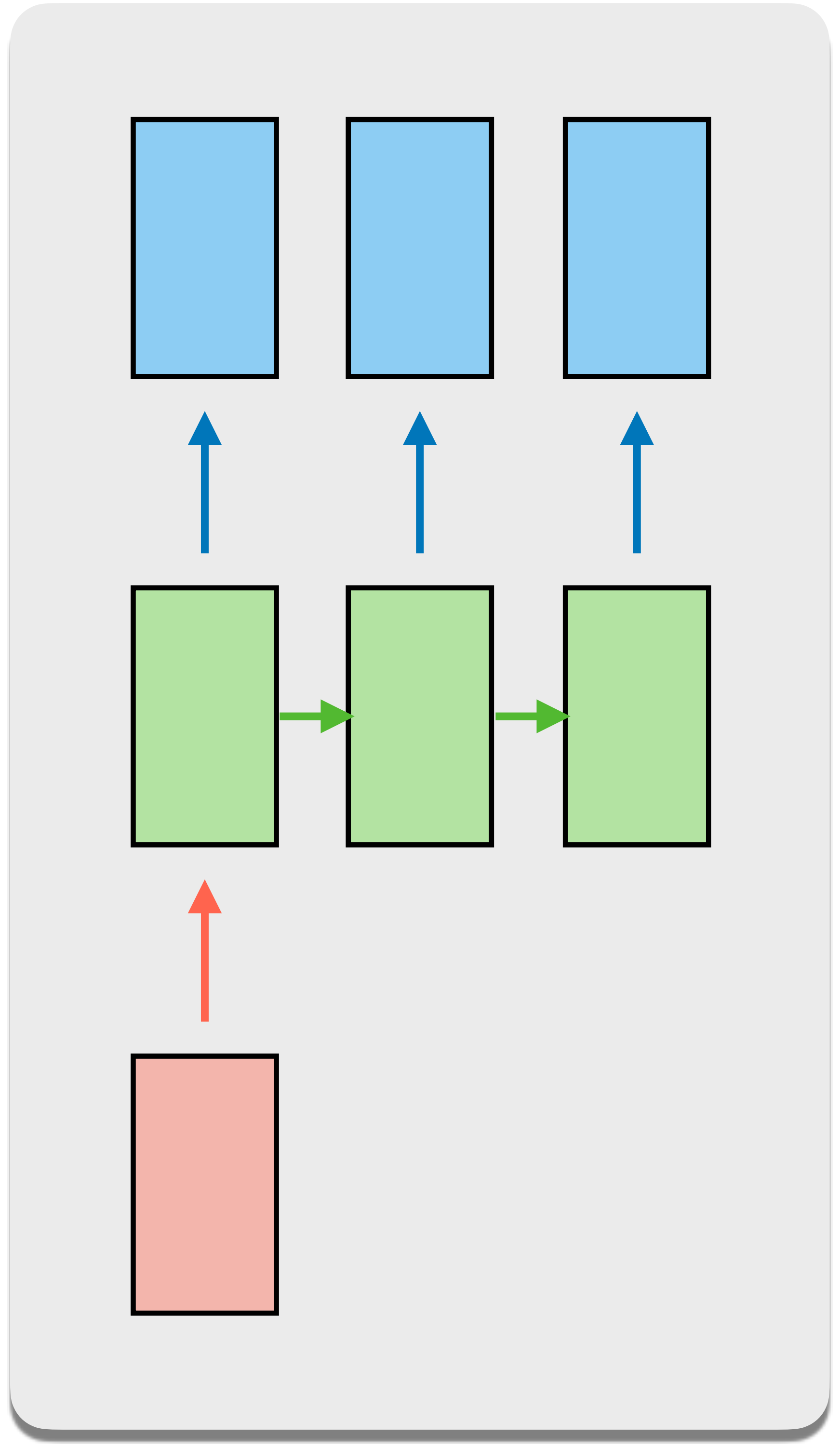
Example
Machine Translation

Many to **one**



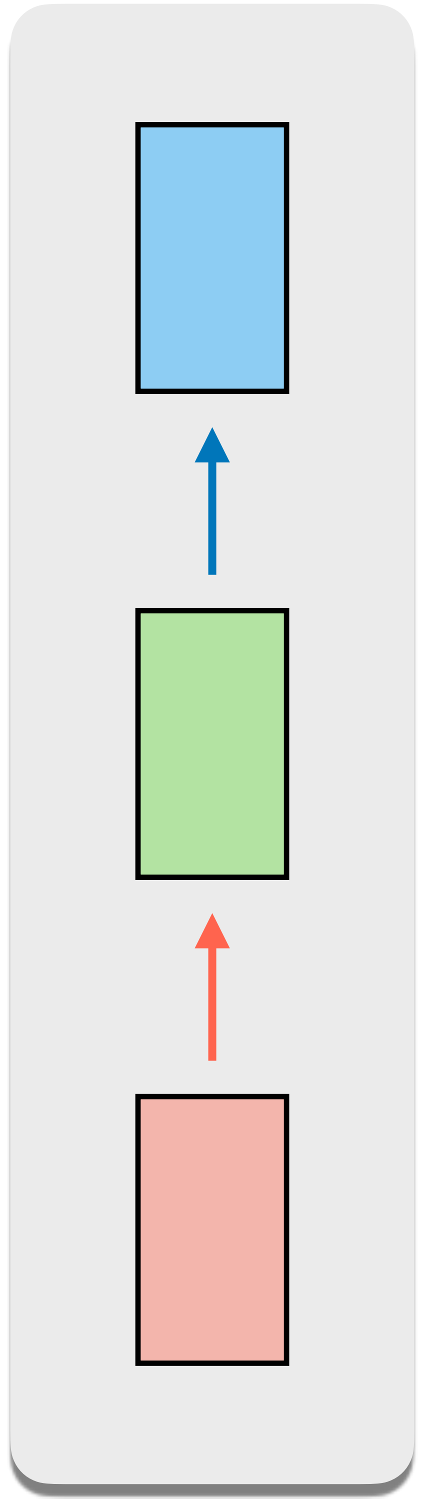
Example
Sentiment Analysis

One to **many**



Example
Image Description

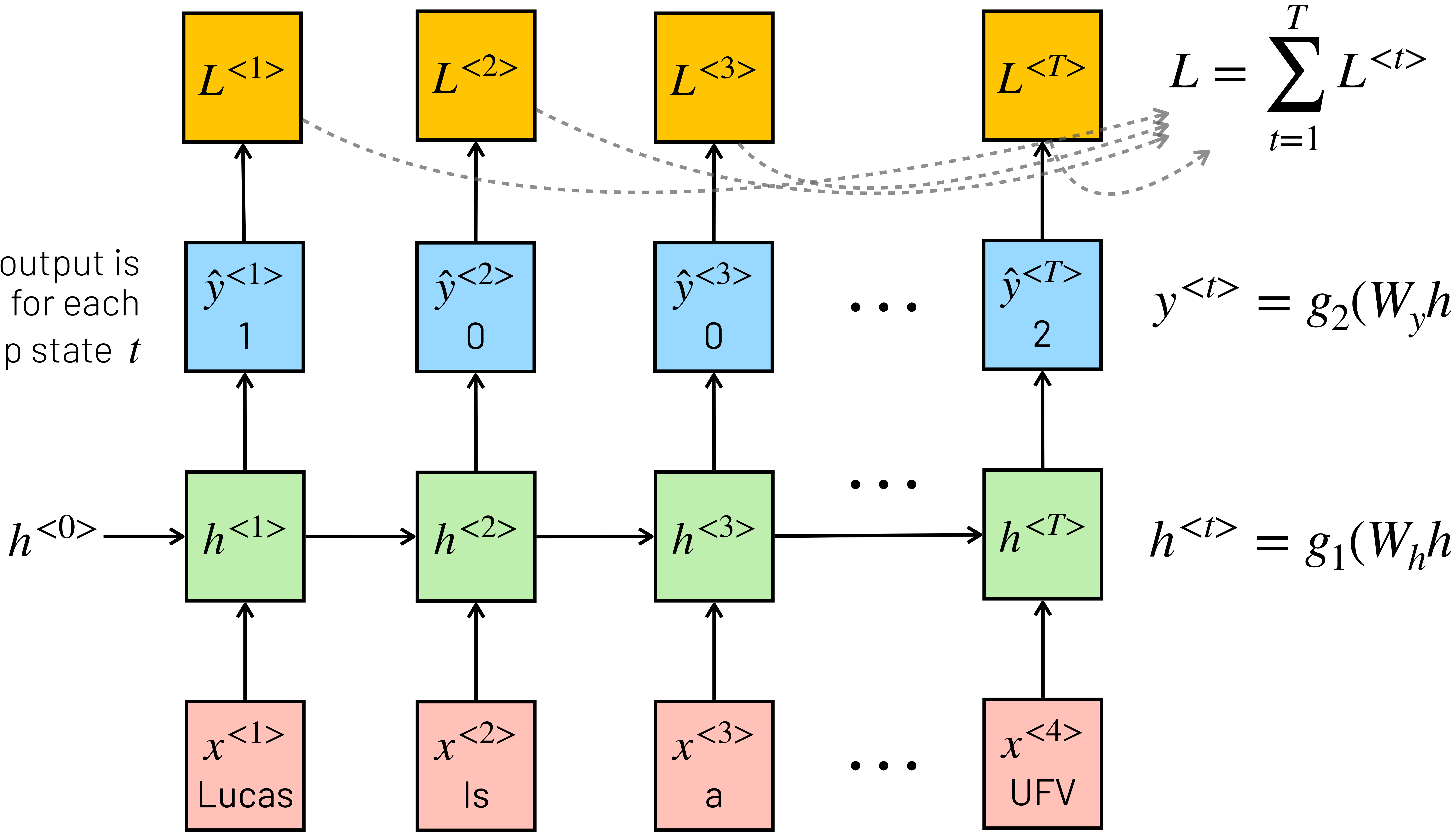
one to **one**



MLP

Many to Many

Named Entity Recognition	"Lucas is a professor at UFV"	1 0 0 0 0 2



$$L = \sum_{t=1}^{T} L^{<t>}$$

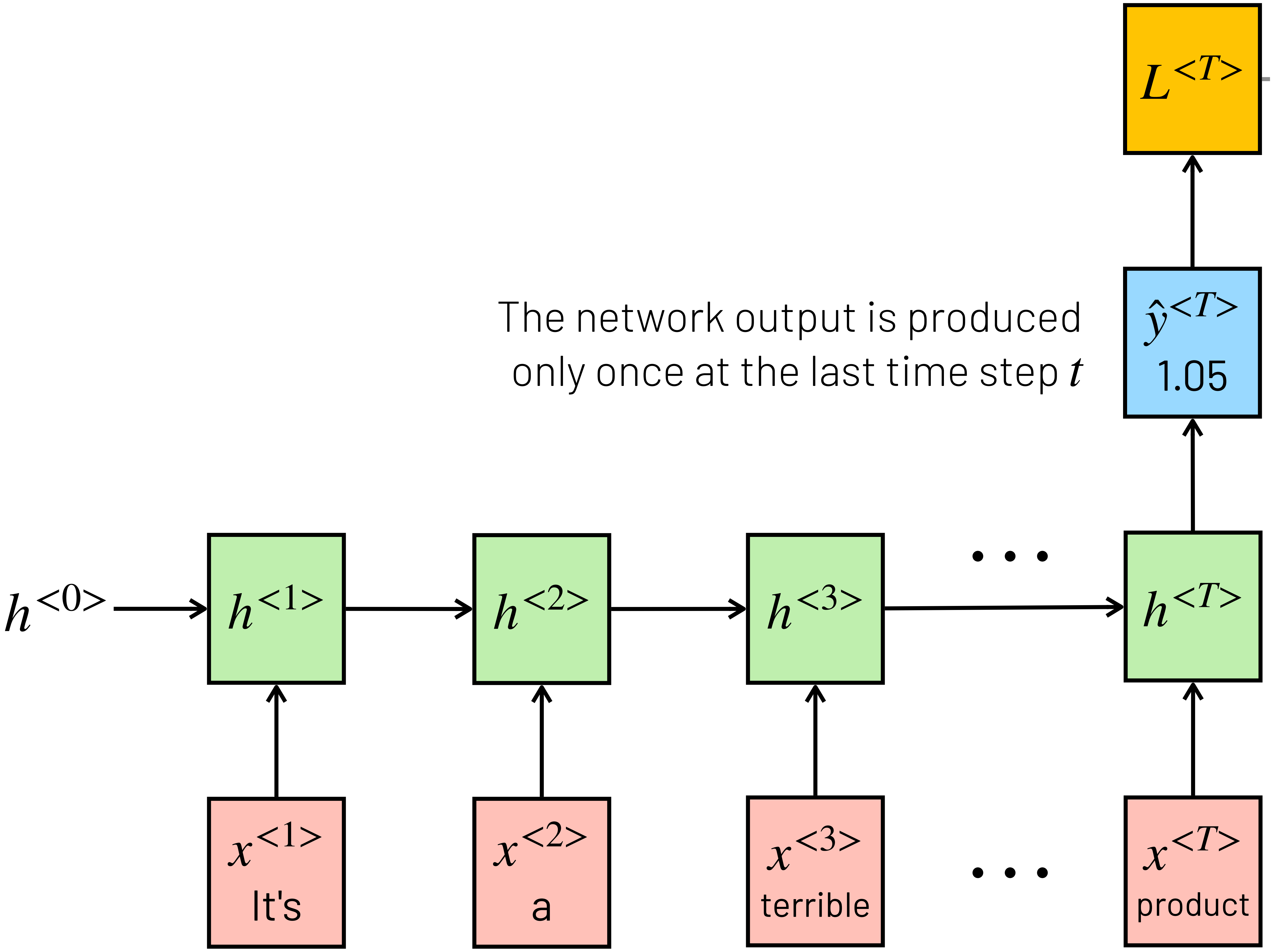
An output is produced for each timestep state t

$$y^{<t>} = g_2(W_y h^{<t>})$$

$$h^{<t>} = g_1(W_h h^{<t-1>} + W_x x^{<t>} + b_h)$$

Many to One

Sentiment Analysis	“It’s a terrible product.”	★☆☆☆☆



The network output is produced only once at the last time step t

$$L = L^{<t>}$$

$$y^{<T>} = g_2(W_y h^{<T>})$$

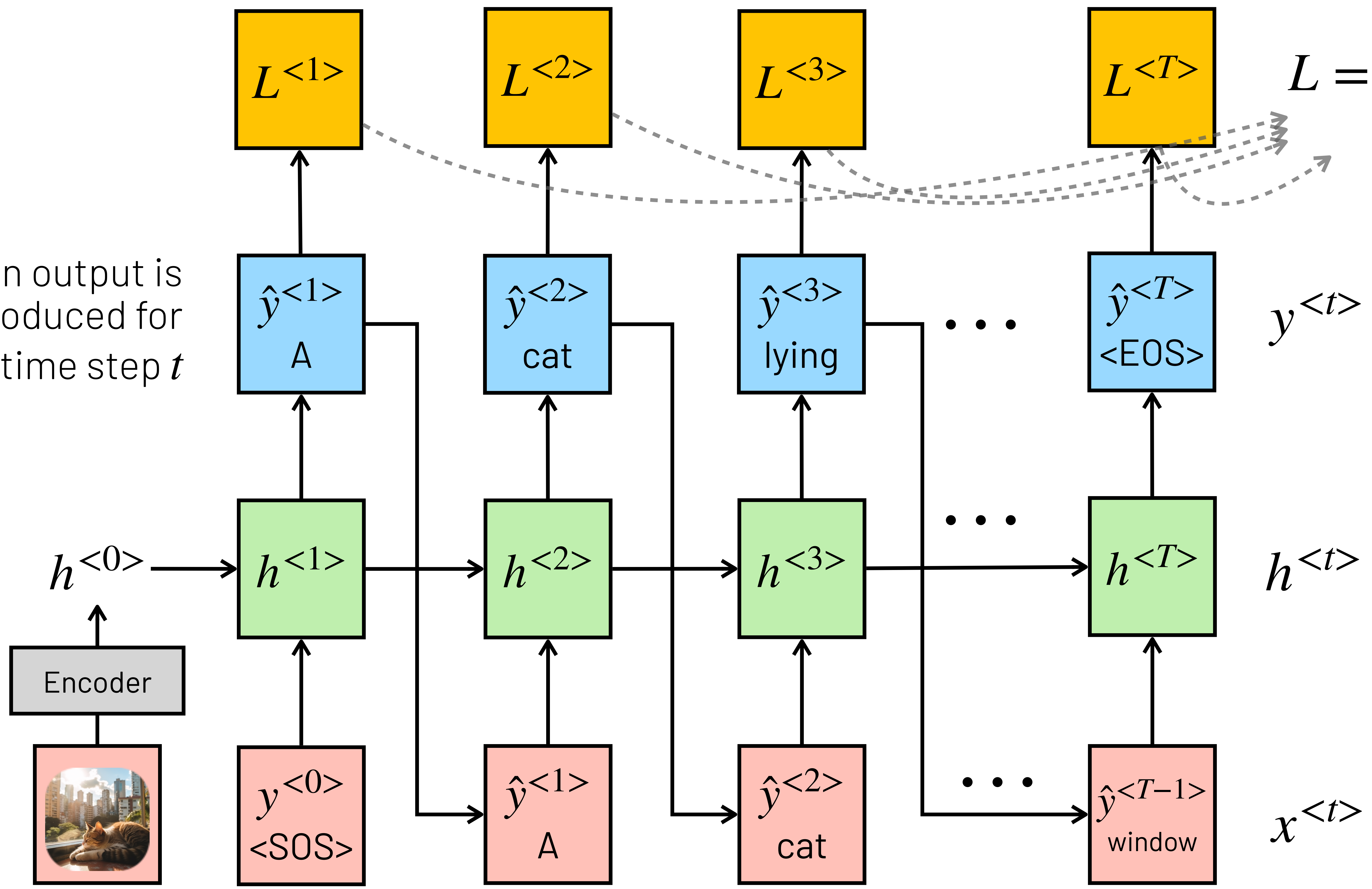
$$h^{<t>} = g_1(W_h h^{<t-1>} + W_x x^{<t>} + b_h)$$

One to Many

Image Captioning

“A cat lying by the window.”

An output is produced for each time step t



$$L = \sum_{t=1}^{T} L^{<t>}$$

$$y^{<t>} = g_2(W_{yh}h^{<t>})$$

$$h^{<t>} = g_1(W_{hh}h^{<t-1>} + W_{hx}x^{<t>} + b_h)$$

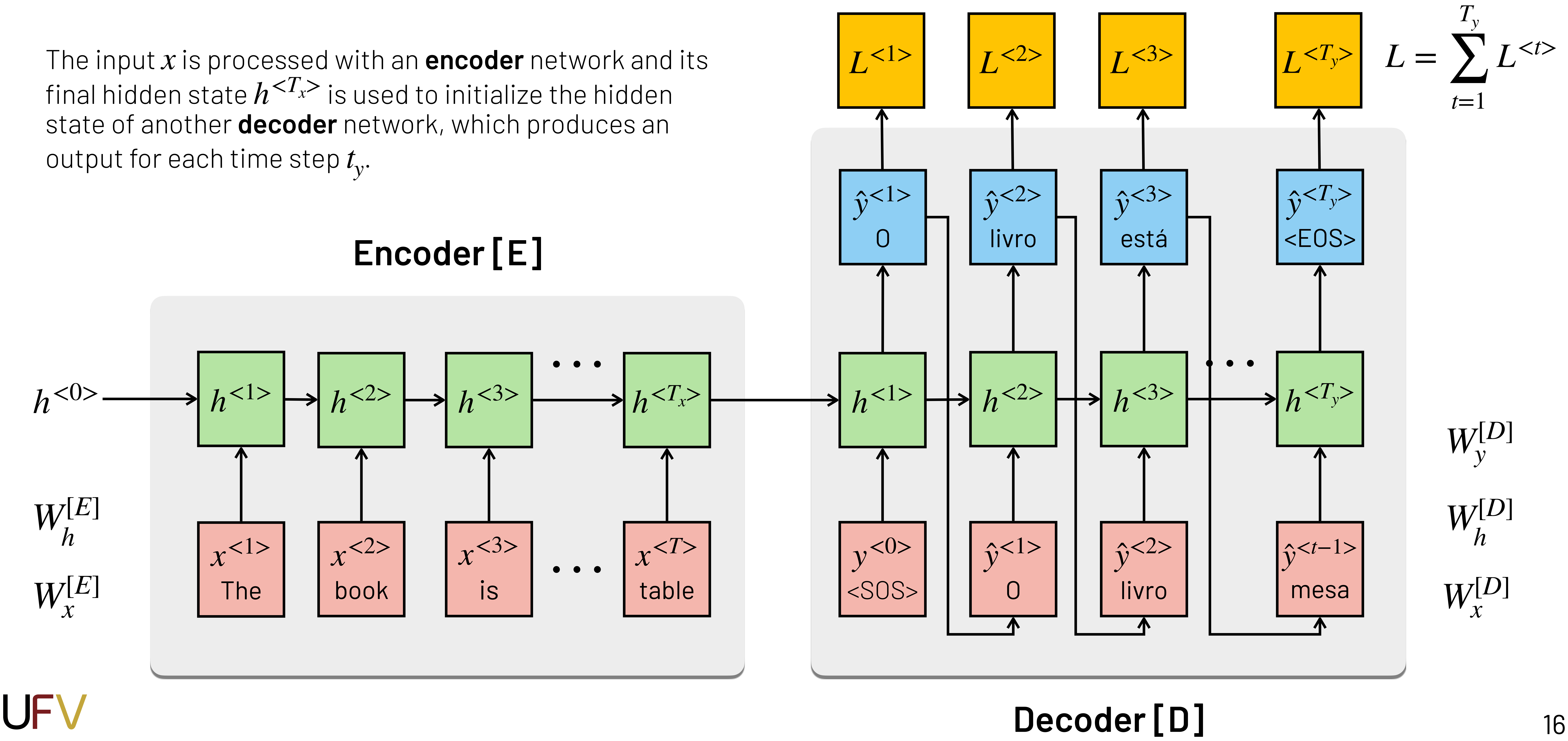
$$x^{<t>} = \hat{y}^{<t-1>}$$

Use $\mathbf{x}$ to initialize $h^{<0>}$

Seq2Seq

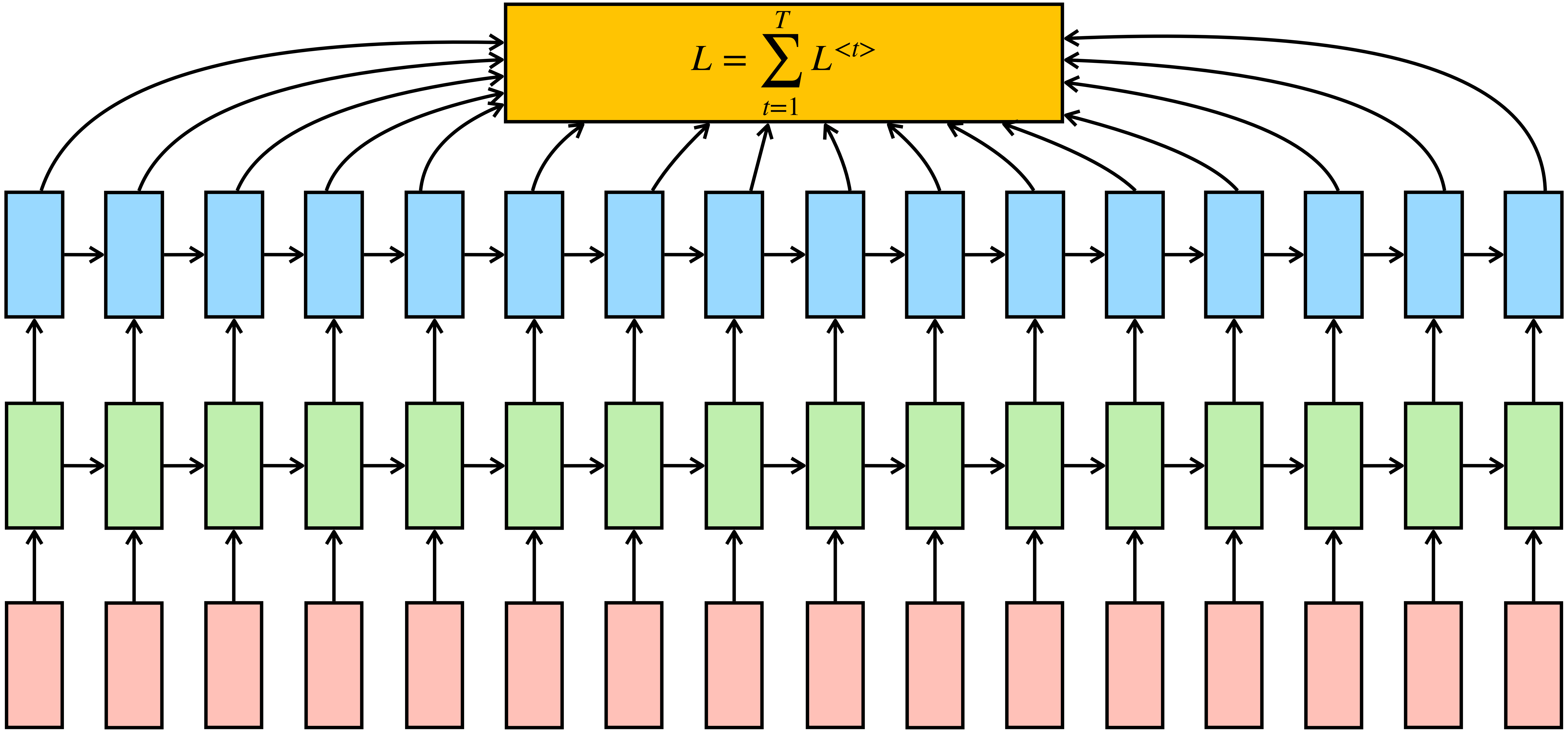
Machine Translation "The book is on the table." "O livro está em cima da mesa."

The input x is processed with an **encoder** network and its final hidden state $h^{<T_x>}$ is used to initialize the hidden state of another **decoder** network, which produces an output for each time step t_y.



UFV

Backpropagation Through Time

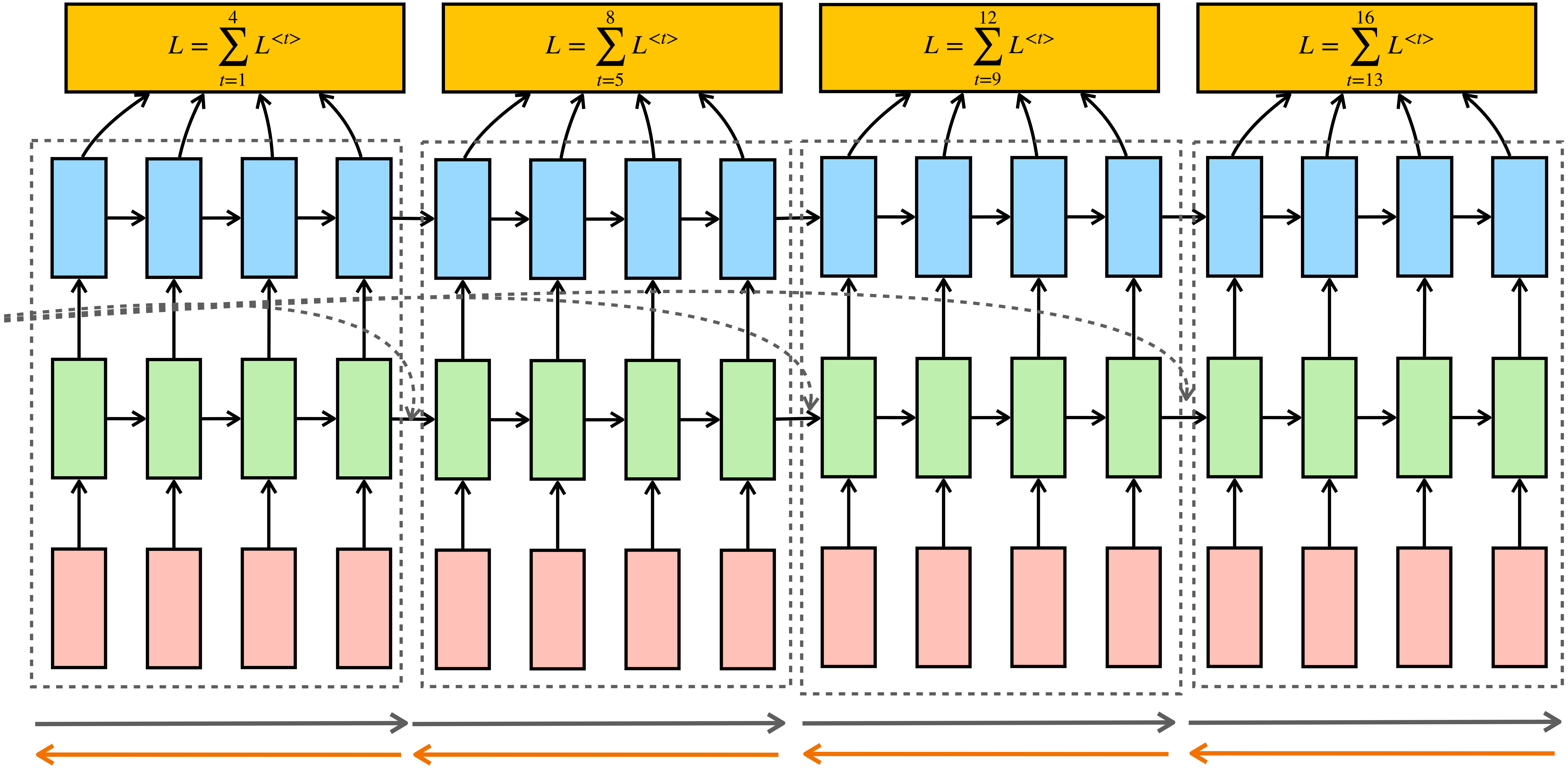


Process all T elements of the sequence to calculate the loss

Backpropagate through the entire sequence to calculate the gradient

Truncated Backpropagation Through Time

If the size of the sequence to be processed is very large or infinite (e.g., time series), perform propagation and backpropagation in windows of size j (e.g., 4)


$L = \sum_{t=1}^{4} L^{<t>}$
$L = \sum_{t=5}^{8} L^{<t>}$
$L = \sum_{t=9}^{12} L^{<t>}$
$L = \sum_{t=13}^{16} L^{<t>}$

The hidden state can be carried between windows.

Next Lecture

L14: Recurrent Neural Networks (Part II)

GRUs and LSTMs for processing with very long sequences.