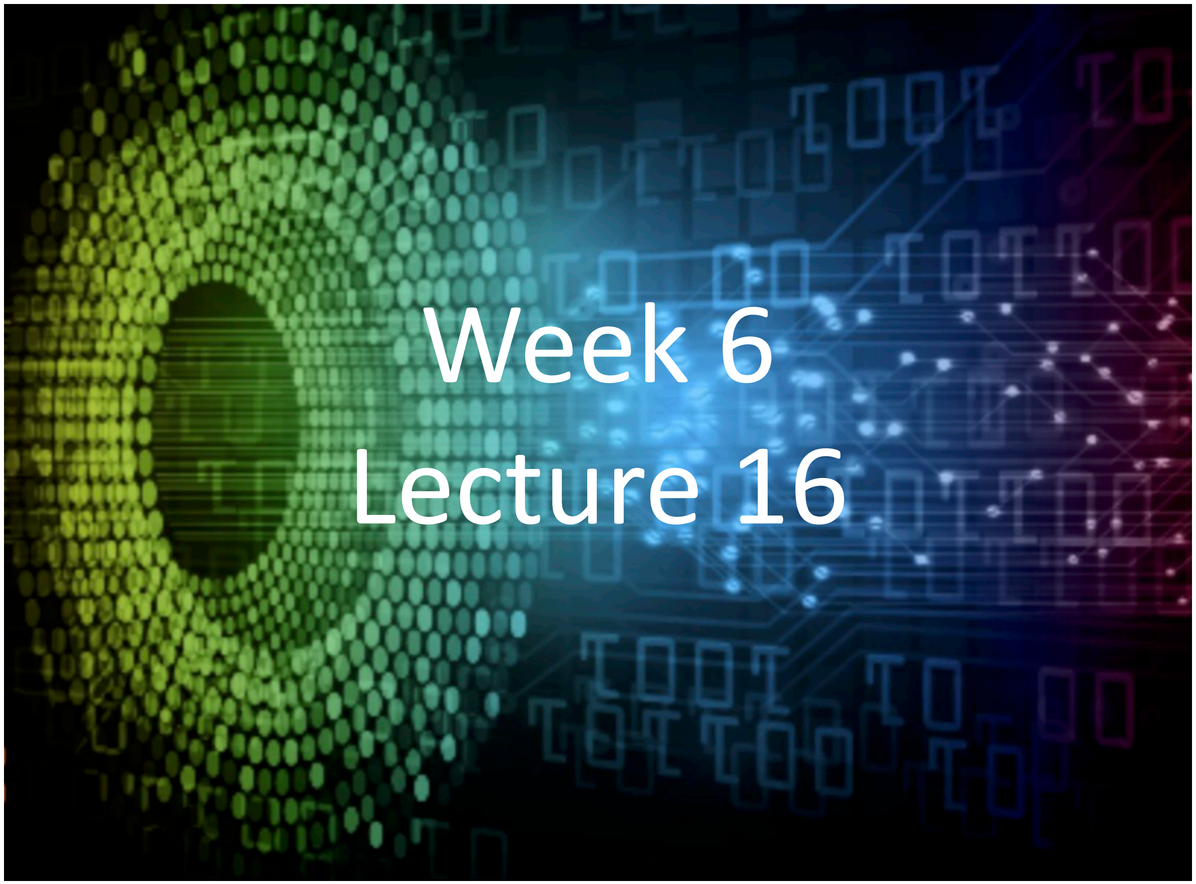
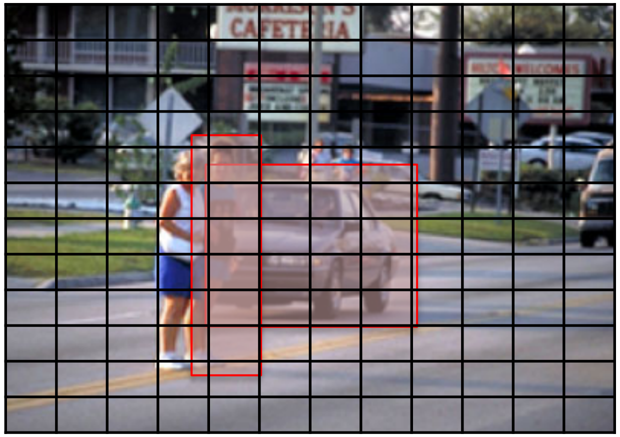
Introduction to Deep Learning Applications and Theory



ECE 596 / AMATH 563

Previous Lecture (Weeks 4-5): CNNs

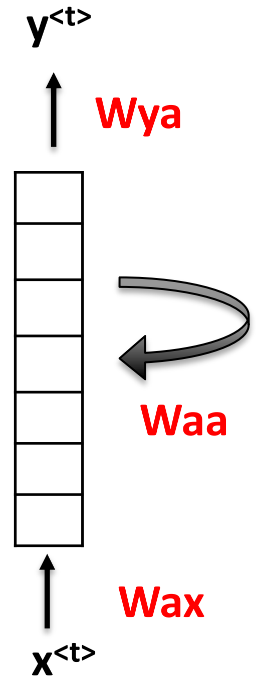
- Definition
- Generalization
- Applications
- Visualization
- Interpretation
- Generation



Anchor boxes

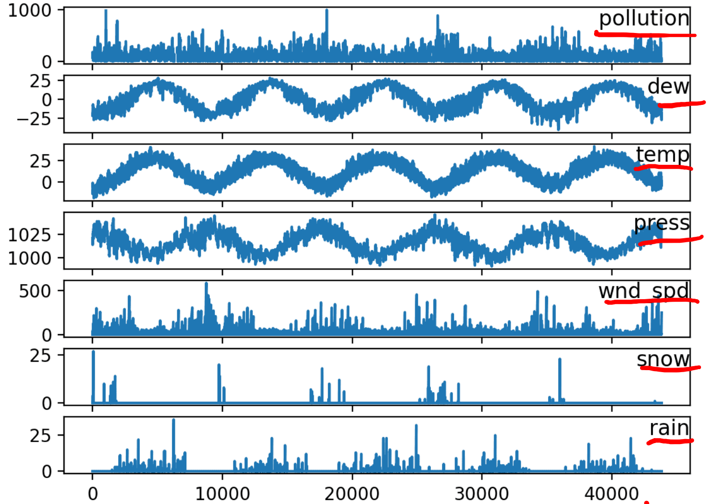
This Week and Lecture: Recurrent Neural Networks (RNNs)

- Input-Output
- Definition
- Notation
- Backward Propagation
- Diminishing/ Exploding Gradients



Input - Output

space

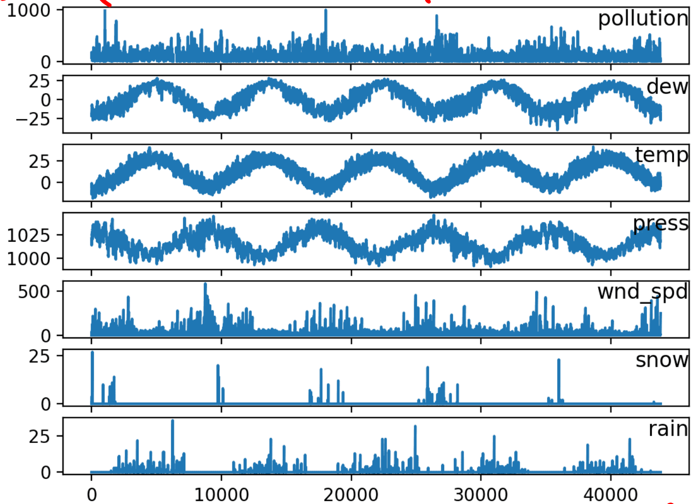


time

Input

prediction
filtering
⋮

space



time

Output

Time Series

Sequences

Recurrent Neural Networks (RNNs)

RNNs: Sequence models

Applications:

	input	**output**

Recurrent Neural Networks (RNNs)

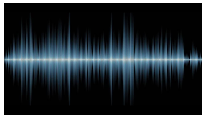
RNNs: Sequence models

Applications:	**input**	**output**
Speech recognition		

Recurrent Neural Networks (RNNs)

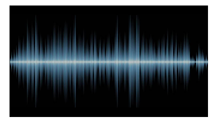
RNNs: Sequence models

Applications:

	input	**output**
Speech recognition	audio 	

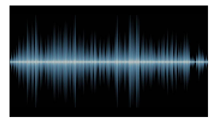
Recurrent Neural Networks (RNNs)

RNNs: Sequence models

Applications:	**input**	**output**
Speech recognition		"Hey Siri, where can I buy a Tesla?"

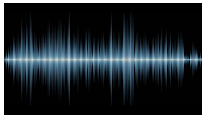
Recurrent Neural Networks (RNNs)

RNNs: Sequence models

Applications:	**input**	**output**
Speech recognition		"Hey Siri, where can I buy a Tesla?"
Music Generation	**Genre**	

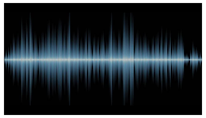
Recurrent Neural Networks (RNNs)

RNNs: Sequence models

Applications:	**input**	**output**
Speech recognition		“Hey Siri, where can I buy a Tesla?”
Music Generation	**Genre**	
DNA Sequence Analysis	AGCCCCTGTGAG	AG**CCCCTGTG**AG

Recurrent Neural Networks (RNNs)

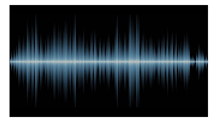
RNNs: Sequence models

Applications:	**<u>input</u>**	**<u>output</u>**
Speech recognition		"Hey Siri, where can I buy a Tesla?"
Music Generation	**Genre**	
DNA Sequence Analysis	AGCCCCTGTGAG	AG**CCCCTGTG**AG
Sentiment Classification	"There is nothing to like in this movie."	★☆☆☆☆

Recurrent Neural Networks (RNNs)

RNNs: Sequence models

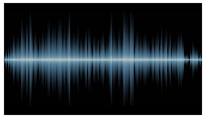
Applications:

	input	**output**
Speech recognition		"Hey Siri, where can I buy a Tesla?"
Music Generation	**Genre**	
DNA Sequence Analysis	AGCCCCTGTGAG	AG**CCCCTGTG**AG
Sentiment Classification	"There is nothing to like in this movie."	★☆☆☆☆
Machine Translation	«Hey Siri, où puis-je acheter une Tesla?»	"Hey Siri, where can I buy a Tesla?"

Recurrent Neural Networks (RNNs)

RNNs: Sequence models

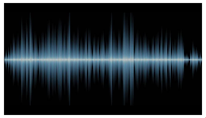
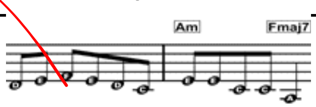
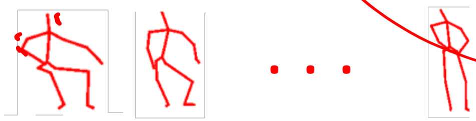
Applications:

	input	**output**
Speech recognition		“Hey Siri, where can I buy a Tesla?”
Music Generation	**Genre**	
DNA Sequence Analysis	AGCCCCTGTGAG	AG**CCCCTGTG**AG
Sentiment Classification	“There is nothing to like in this movie.”	★☆☆☆☆
Machine Translation	«Hey Siri, où puis-je acheter une Tesla?»	“Hey Siri, where can I buy a Tesla?”
Name/Entity Recognition	“SpaceX is managed by Elon Musk”	“SpaceX is managed by Elon Musk”

Recurrent Neural Networks (RNNs)

RNNs: Sequence models

Applications:

	input	**output**
Speech recognition		"Hey Siri, where can I buy a Tesla?"
Music Generation	**Genre**	
DNA Sequence Analysis	AGCCCCTGTGAG	AG**CCCCTGTG**AG
Sentiment Classification	"There is nothing to like in this movie."	★☆☆☆☆
Machine Translation	«Hey Siri, où puis-je acheter une Tesla?»	"Hey Siri, where can I buy a Tesla?"
Name/Entity Recognition	"SpaceX is managed by Elon Musk"	"SpaceX is managed by Elon Musk"
Activity Recognition		"Standing up to buy a Tesla"

Notations

Example: Name Entity Recognition (NLP)

X:

Harry	Potter	and	Hermione	Granger	invented	a	new	spell.
$x^{<1>}$	$x^{<2>}$							$x^{<9>}$

Y:

$X^{(i)<t>}$

Sequence length = T_x =9

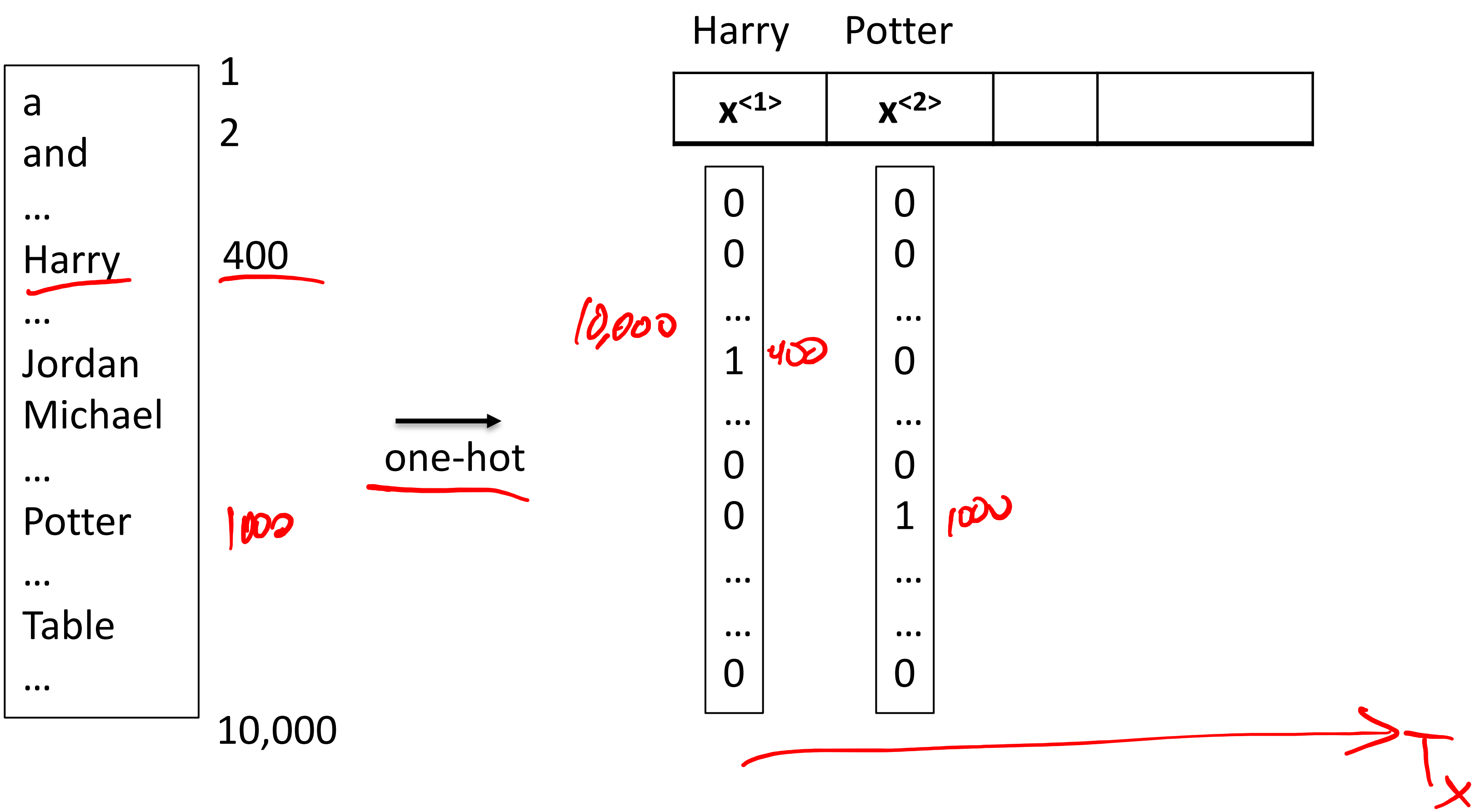
1	1	0	1	1	0	0	0	0
$y^{<1>}$	$y^{<2>}$							$y^{<9>}$

$Y^{(i)<t>}$

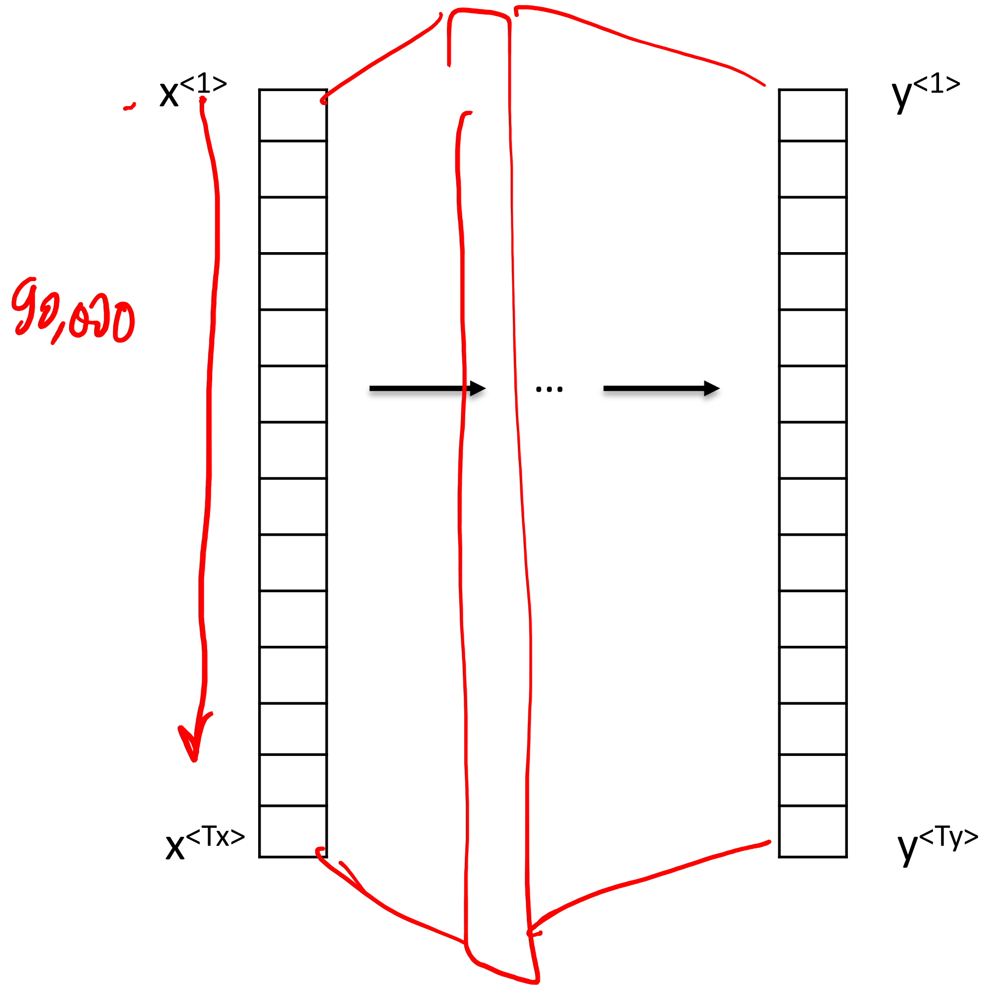
Sequence length = T_y =9

Words Representation

Vocabulary/Dictionary



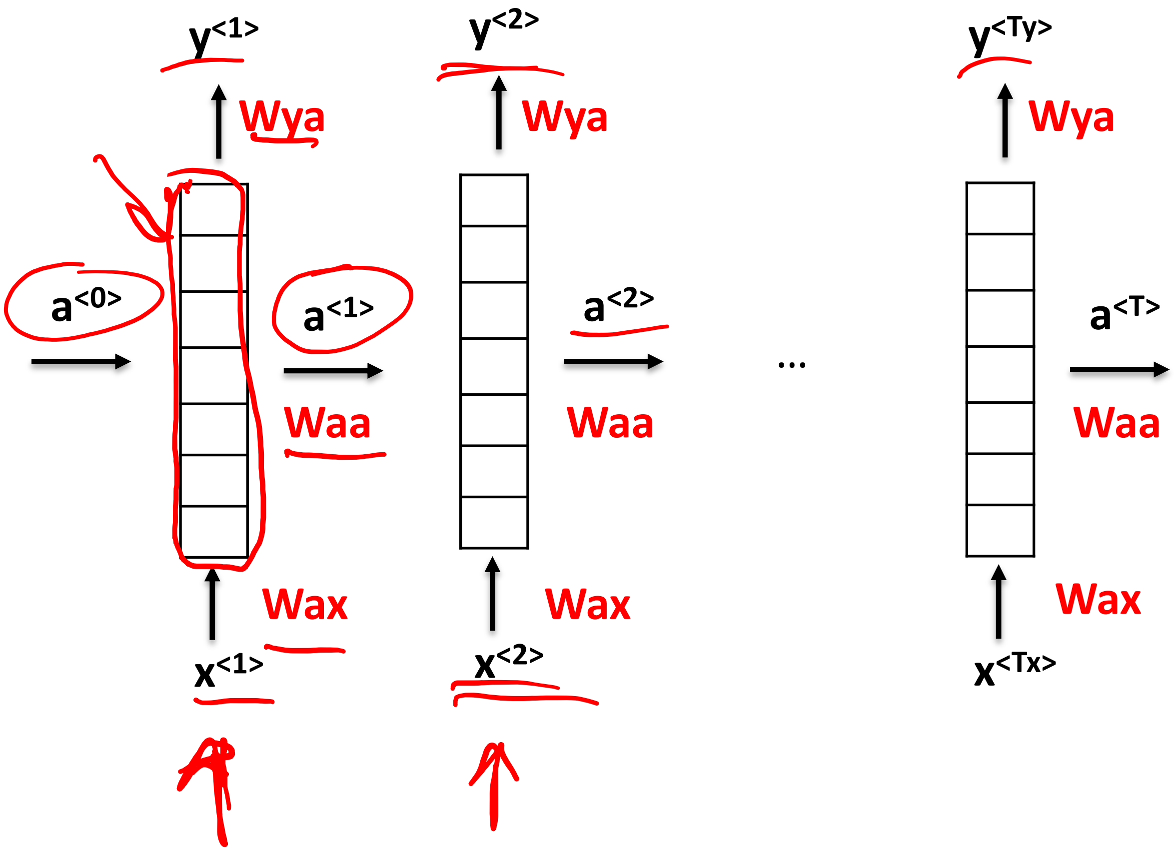
Direct Approach



Direct Approach Cons

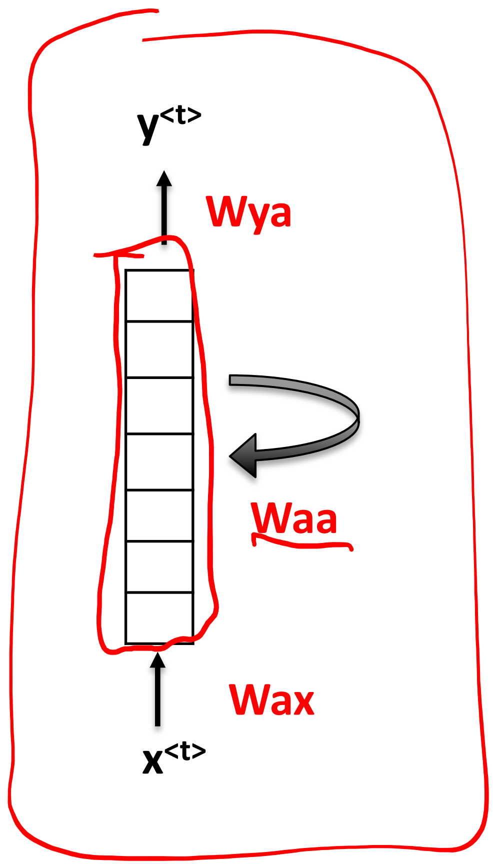
- Input & output dimensions change for each input
- Doesn’t share features across sequence stepping
- Large dimensions

UniDirectional Recurrent Neural Network (RNN)



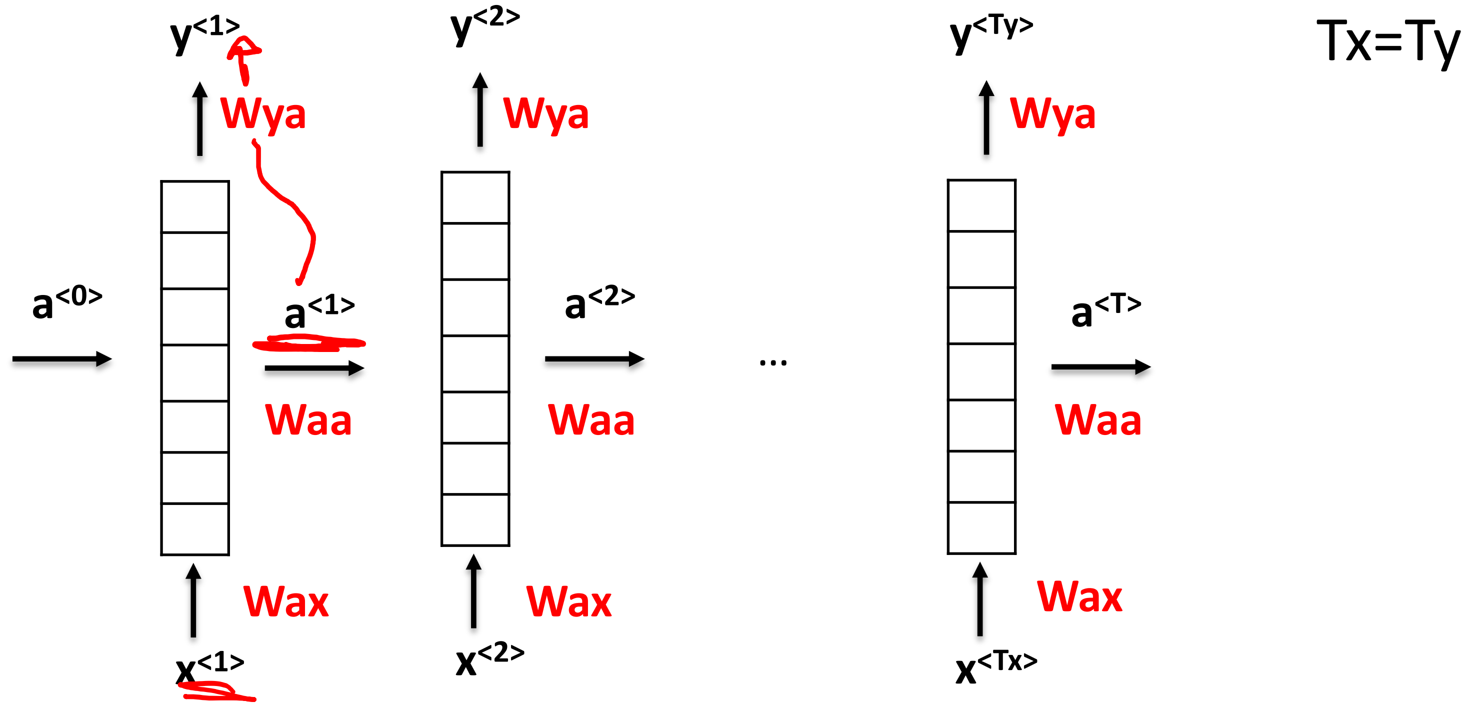
Tx=Ty

Recurrent Neural Network (RNN)



Tx=Ty

Forward Propagation



$a^{<1>} = g_1(W_{aa}\, a^{<0>} + W_{ax}\, x^{<1>} + b_a)$ $\quad y^{<1>} = g_2(W_{ya}\, a^{<1>} + b_y)$

$\boxed{a^{<t>} = g_1(W_{aa}\, a^{<t-1>} + W_{ax}\, x^{<t>} + b_a)}$ $\quad y^{<t>} = g_2(W_{ya}\, a^{<t>} + b_y)$

Simplified Notation

$$y^{<t>} = g(W_{ya}\, a^{<t>} + b_y)$$

$$\boxed{a^{<t>} = g(W_{aa}\, a^{<t-1>} + W_{ax}\, x^{<t>} + b_a)}$$

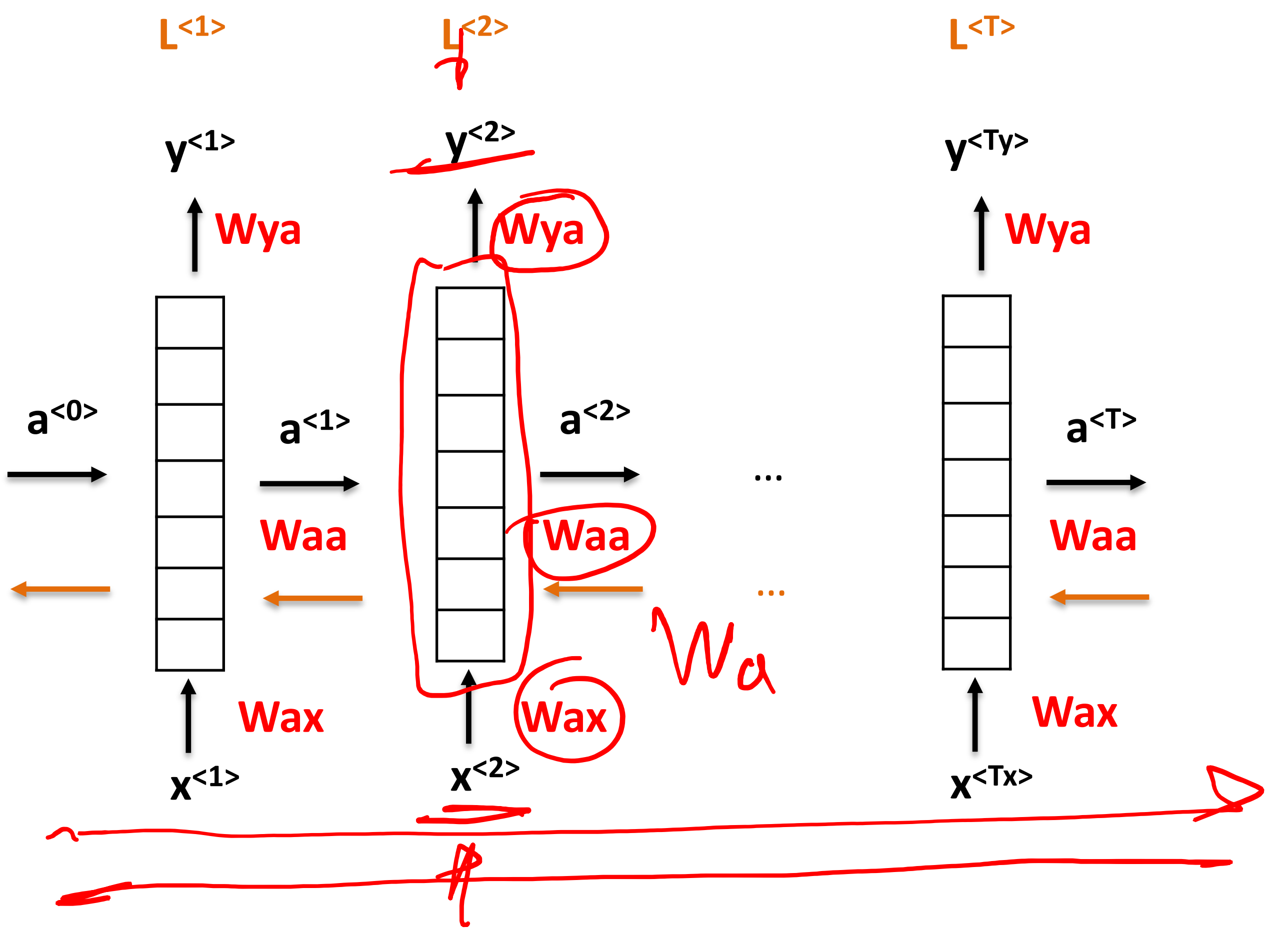
$$a^{<t>} = g(W_a[a^{<t-1>}; x^{<t>}] + b_a)$$

$$\boxed{W_a = [W_{aa} \,|\, W_{ax}]}$$

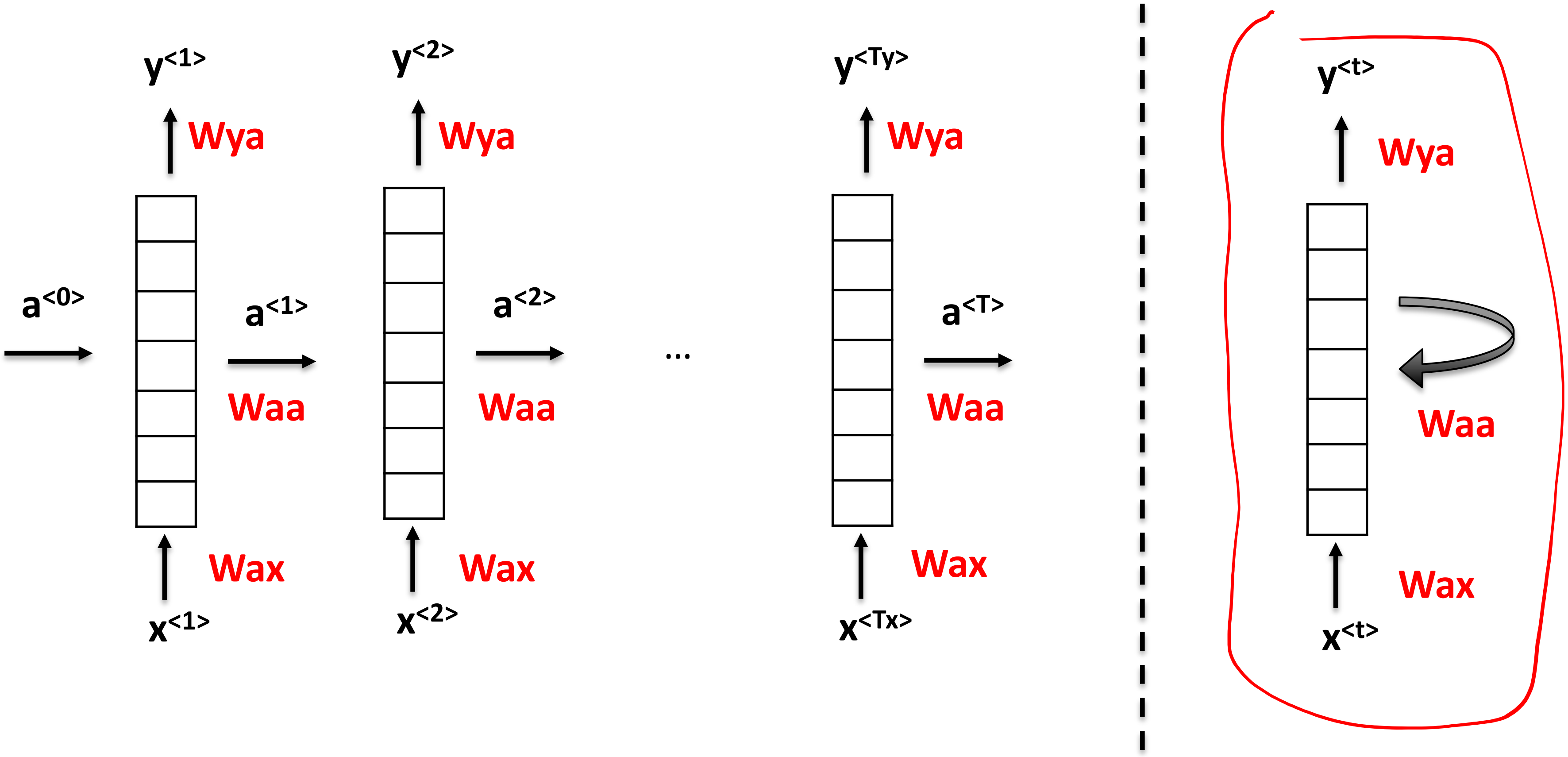
$$[a^{<t-1>}; x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix}$$

Backward Propagation

through time

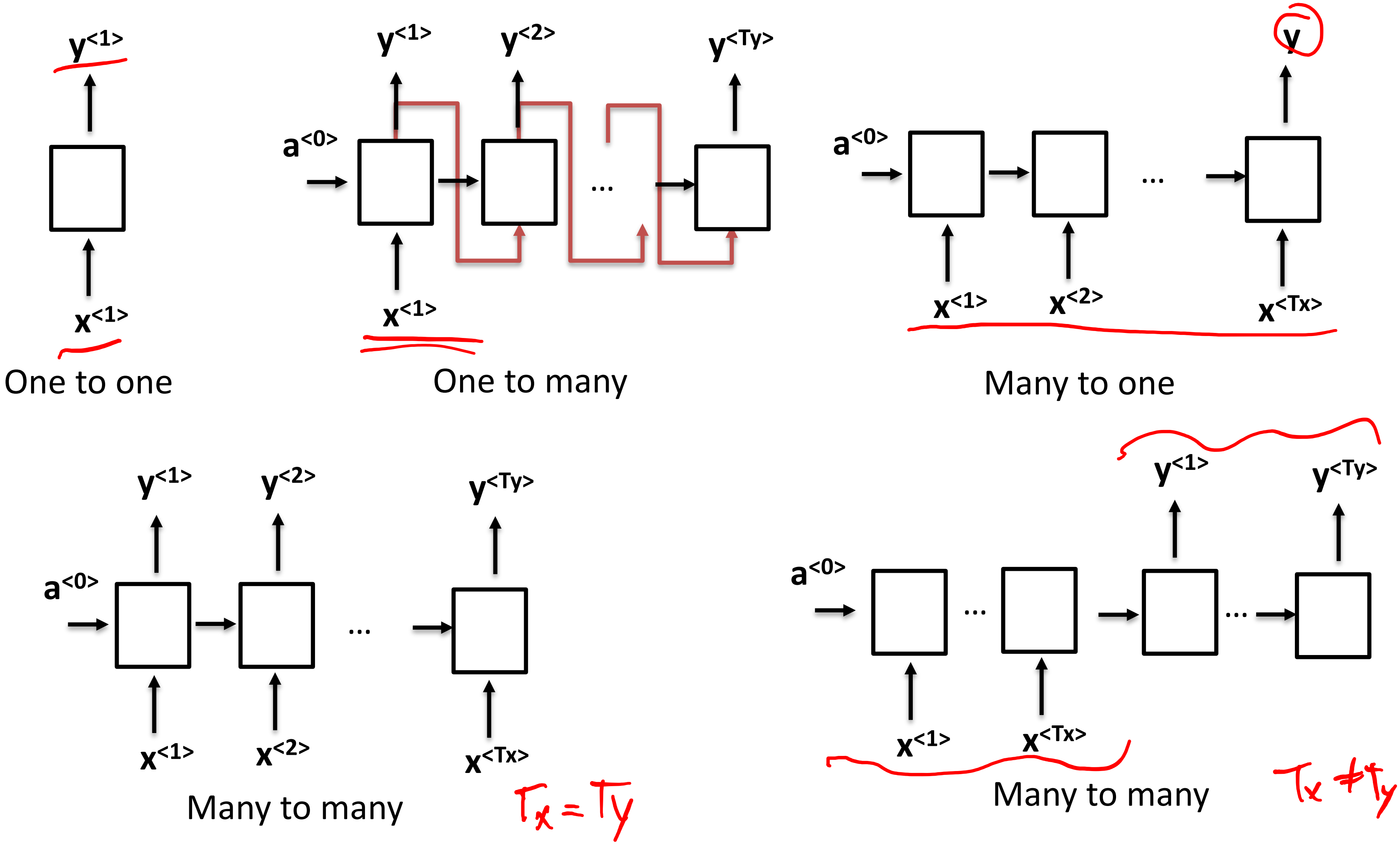


UniDirectional Recurrent Neural Network (RNN)



Tx=Ty

Types of RNNs



Language Model

Language model: probability distribution over sequences of words. Given such a sequence, say of length m, it assigns a probability to the whole sequence.

P(W=SpaceX will **take** me to Mars) = p1

P(W=SpaceX will **bake** me to Mars) = p2

p2 << p1

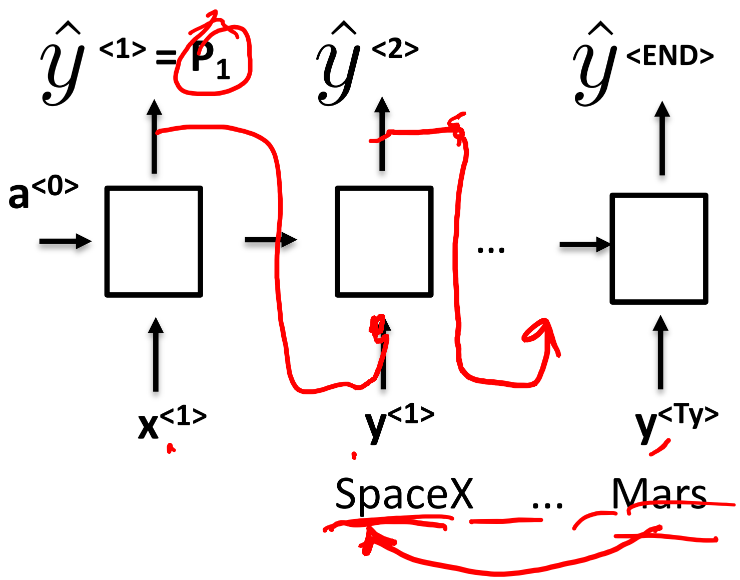
Chain rule is used to estimate probability:

$$P(w_1 w_2 \dots w_n) = \prod_i P(w_i | w_1 w_2 \dots w_{i-1})$$

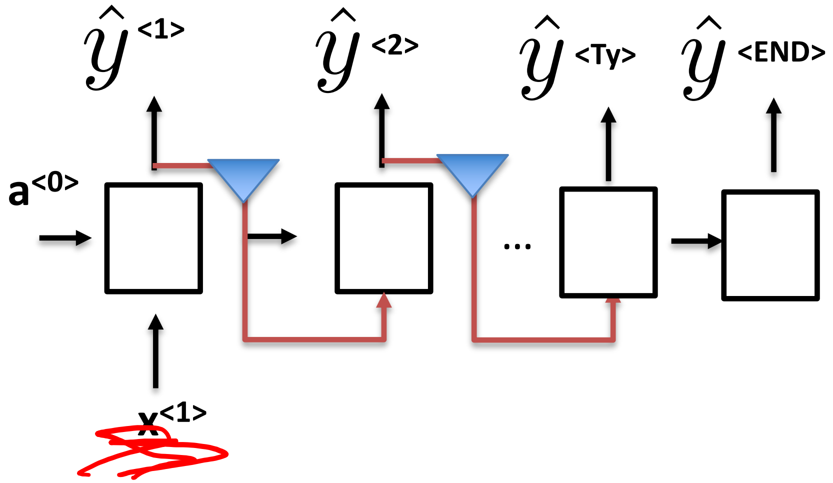
P(W) = P(**SpaceX**) P(**will**|SpaceX) P(**take**|SpaceX will) P(**me**|SpaceX will take) P(**to**|SpaceX will take me) P(**Mars**|SpaceX will take me to)

RNN Language Model

Training:

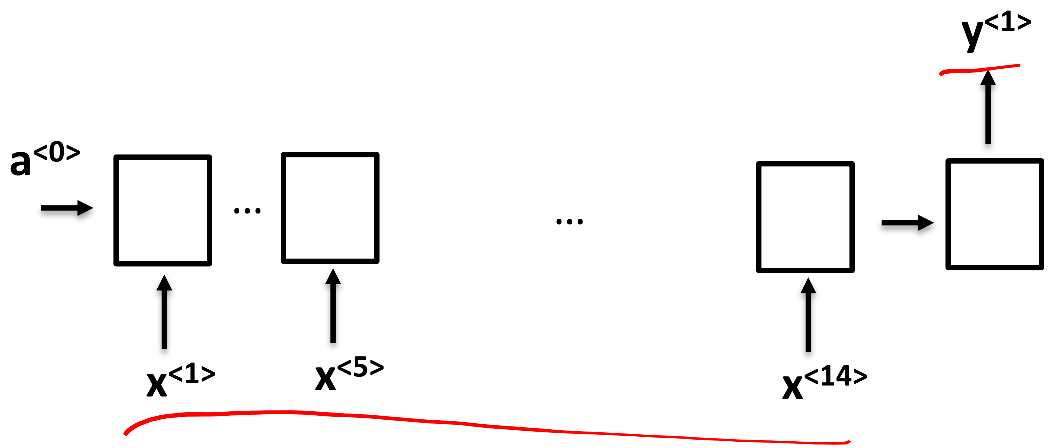


Sampling:



Vanishing/Exploding Gradients

"I grew up in France and moved to United States, therefore I speak _____ Franch"



Vanishing gradients: Gated Structure

Exploding gradients: Gradient Clipping