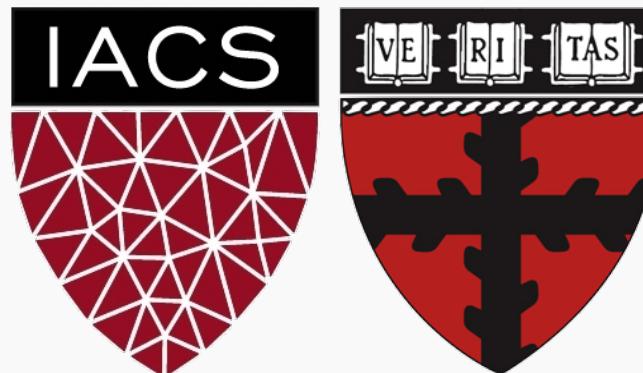


Lecture 14: Recurrent Neural Networks

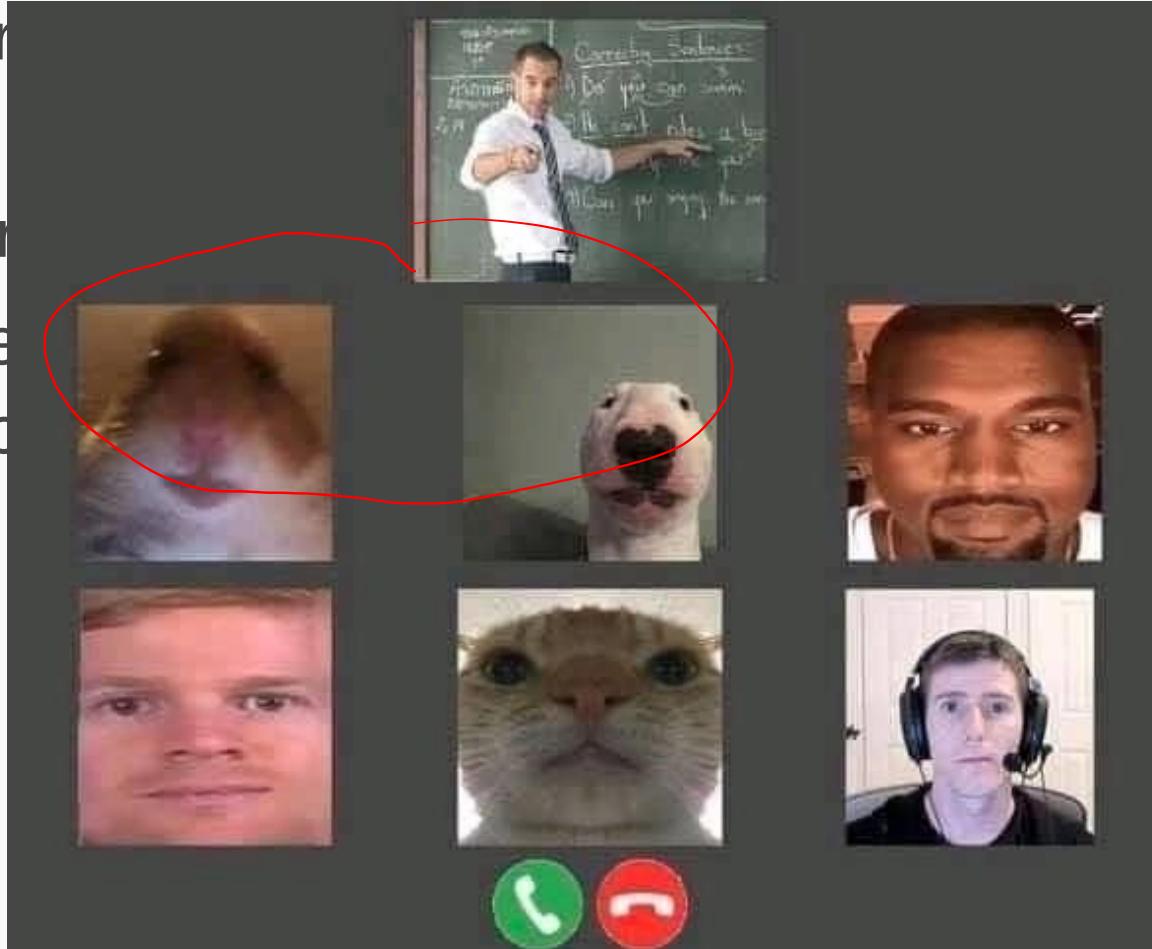
CS109B Data Science 2

Pavlos Protopapas, Mark Glickman, and Chris Tanner



Online lectures guidelines

- We would prefer you have your video on, but it is OK if you have it off.
- We would prefer you have your video on, but it is OK if you have it off.
- All lectures, including the first one, will be recorded and made available for review.
- We will have a Q&A session at the end of each lecture. You can also ask questions during the lecture or enter your own questions in the Q&A section.
- Quizzed will be based on the material covered in the lectures.



Outline

Why Recurrent Neural Networks (RNNs)

Main Concept of RNNs

More Details of RNNs

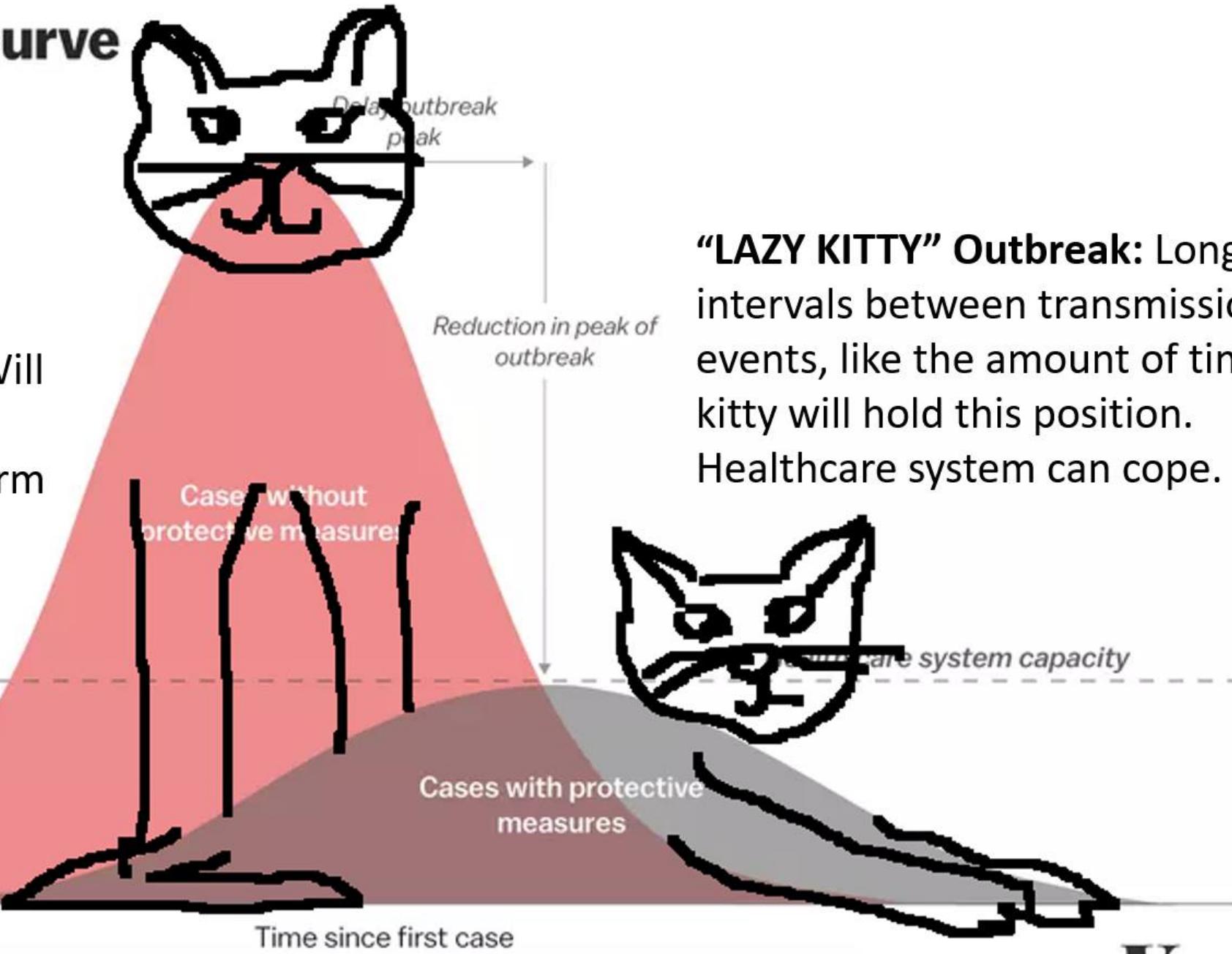
RNN training

Gated RNN

~~Flattening the curve~~ CAT

Daily
number of
cases

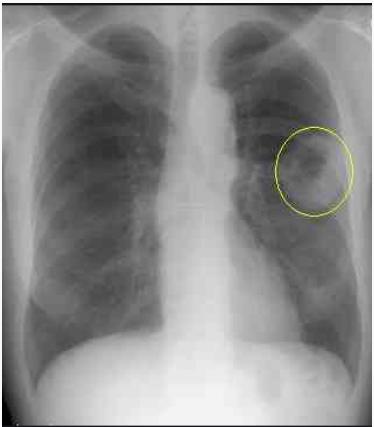
"ALERT KITTY" Outbreak: Will pounce and shred the healthcare system like the arm of your couch.



"LAZY KITTY" Outbreak: Long intervals between transmission events, like the amount of time kitty will hold this position. Healthcare system can cope.

Background

Many classification and regression tasks involve data that is assumed to be **independent and identically distributed (i.i.d.)**.
For example:



Detecting lung cancer



Face recognition



Risk of heart attack

Background

Much of our data is inherently **sequential**

scale

examples

WORLD

Natural disasters (e.g., earthquakes)

Climate change

HUMANITY

Stock market

Virus outbreaks

INDIVIDUAL PEOPLE

Speech recognition

Machine Translation (e.g., English -> French)

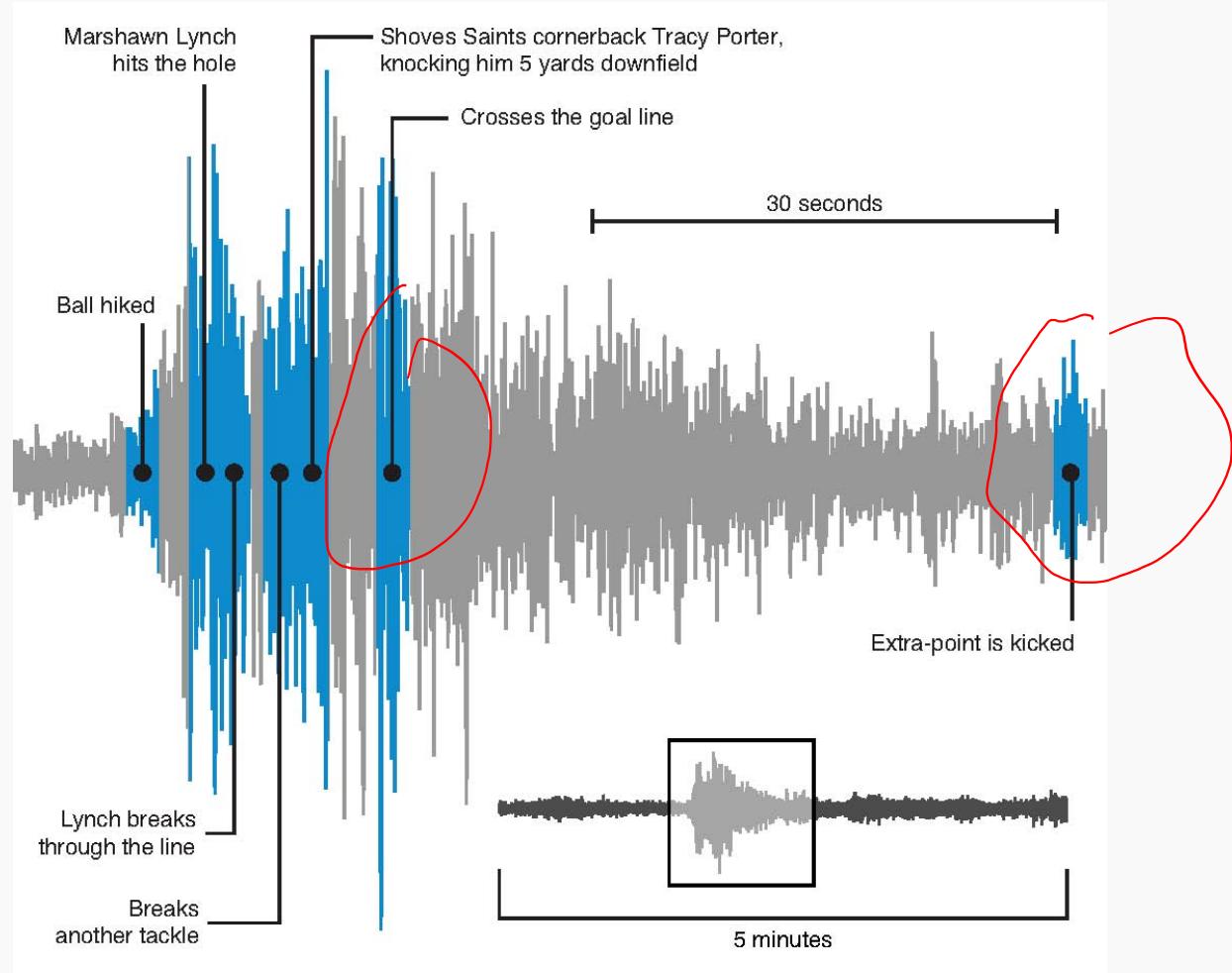
Cancer treatment



Background

Much of our data is inherently **sequential**

PREDICTING EARTHQUAKES



Background

Much of our data is inherently **sequential**

STOCK MARKET PREDICTIONS



Background

Much of our data is inherently **sequential**

SPEECH RECOGNITION

“What is the weather today?”

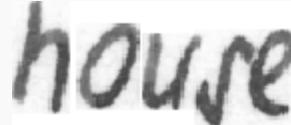
“What is the weather two day?”

“What is the whether too day?”

“What is, the Wrether to Dae?”



Sequence Modeling: Handwritten Text

 → "house"

Winter is here. Go to
the store and buy some
snow shovels.

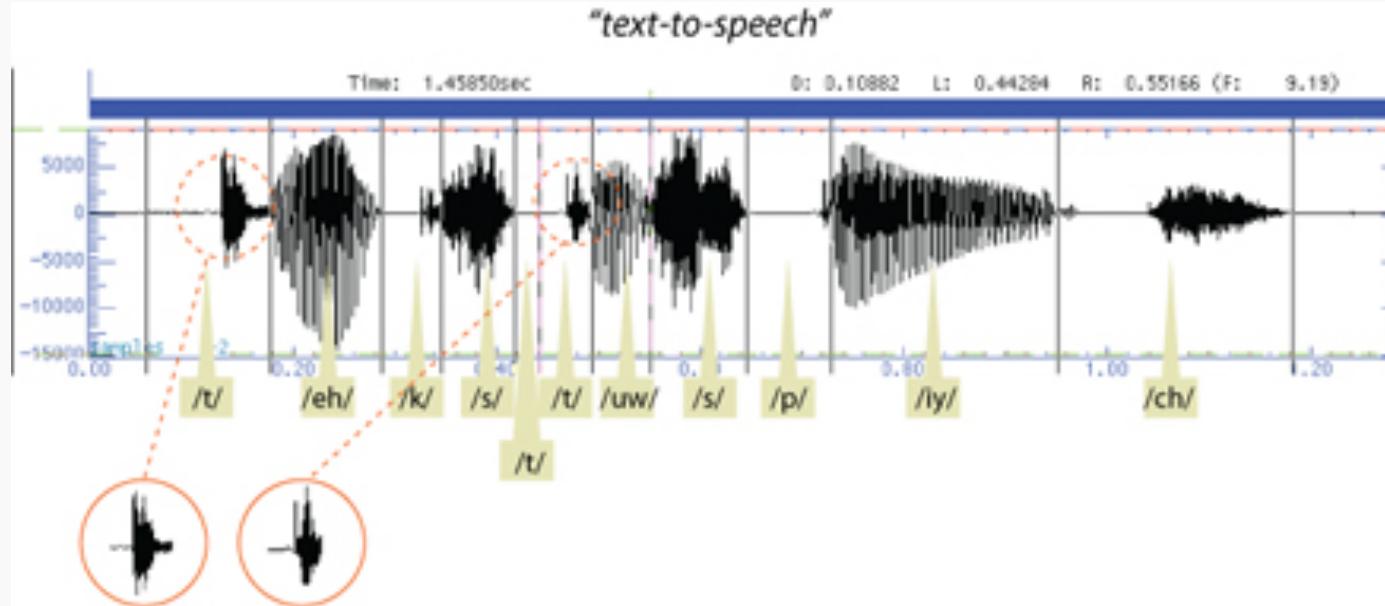
Winter is here. Go to the store and buy
some snow shovels.

- Input : Image
- Output: Text

<https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5>

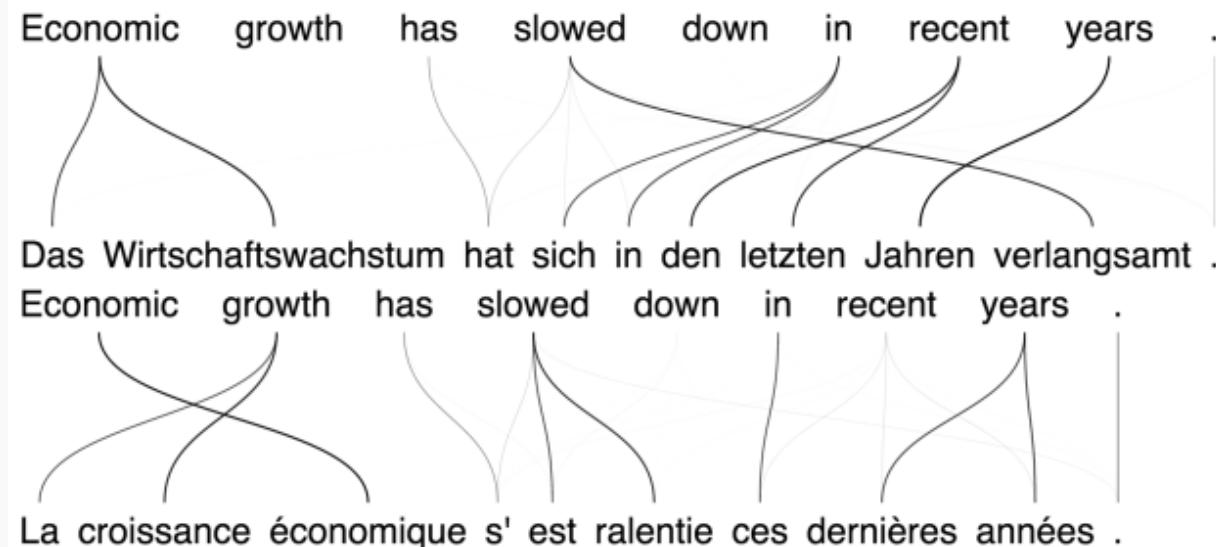


Sequence Modeling: Text-to-Speech



- Input : Text
- Output: Audio

Sequence Modeling: Machine Translation



- Input : Text
- Output: Translated Text

Outline

Why RNNs

Main Concept of RNNs (part 1)

More Details of RNNs

RNN training

Gated RNN

WHERE'S WALDO?

SOCIAL DISTANCING EDITION



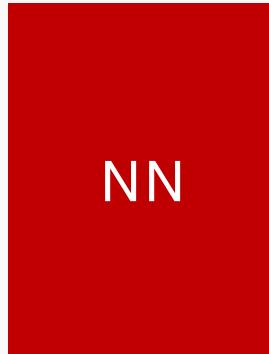
What can my NN do?

Training: Present to the NN examples and learn from them.

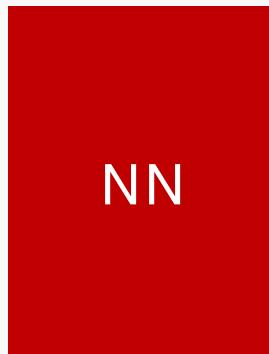


What can my NN do?

Prediction: Given an example



George

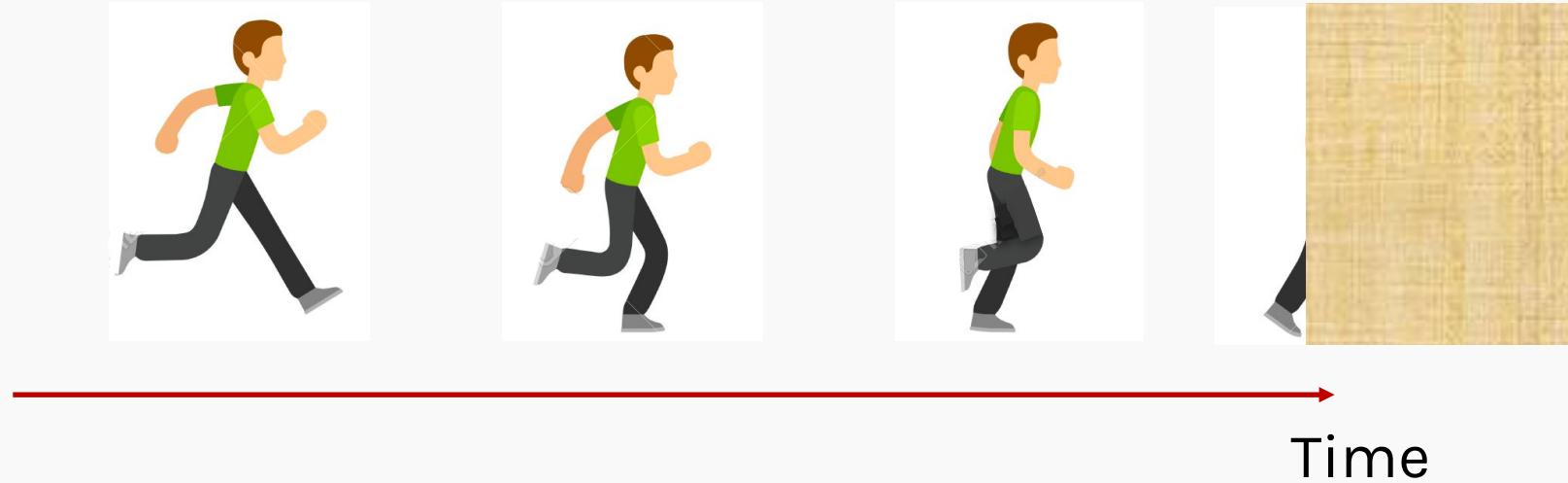


Mary

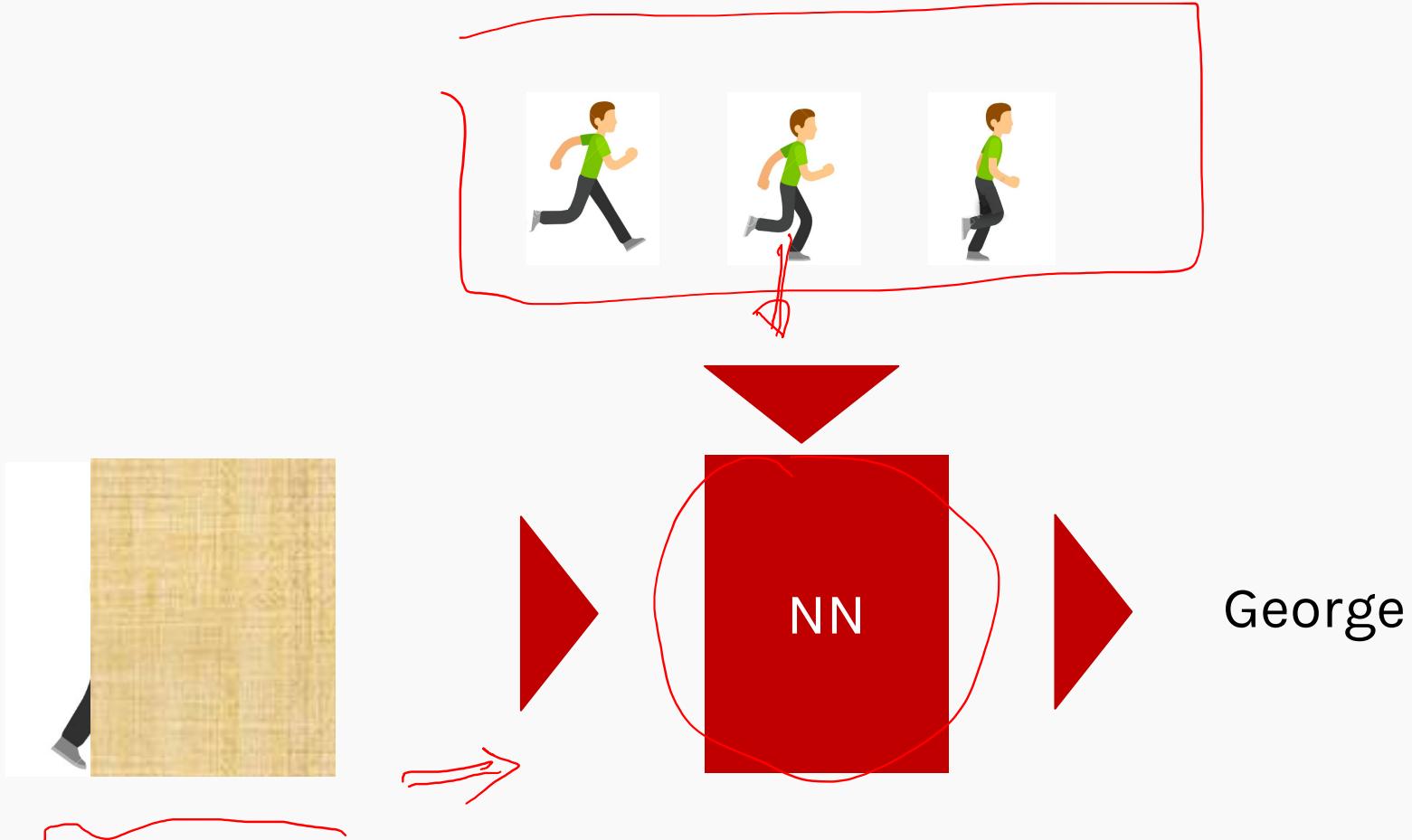
What my NN can NOT do?



Learn from previous examples

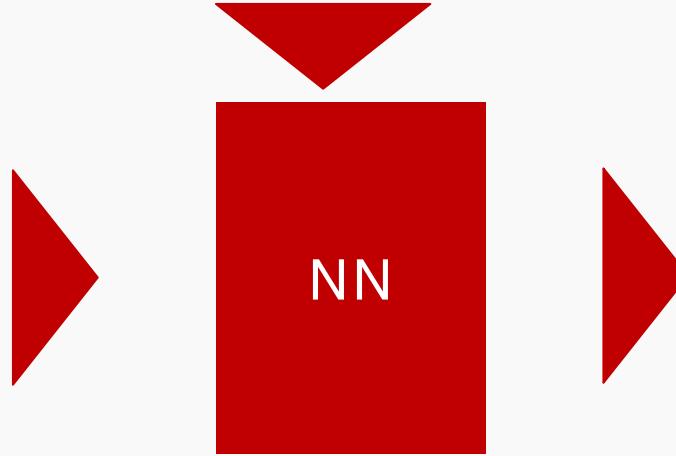


Recurrent Neural Network (RNN)



Recurrent Neural Network (RNN)

I have seen George
moving in this way
before.



RNNs recognize the data's sequential characteristics and use patterns to predict the next likely scenario.

Recurrent Neural Network (RNN)

He **told** me I could have it



I do not know. I need to know who said that and what he said before. Can you tell me more?

} Our model requires context - or contextual information - to understand the subject (he) and the direct object (it) in the sentence.

RNN – Another Example with Text

- Hellen: Nice sweater Joe.
- Joe: Thanks, Hellen. It used
to belong to my brother and ~~he~~ **he**
told me I could have it.



I see what you mean now!
The noun “he” stands for
Joe’s brother while “it” for
the sweater.

After providing sequential information, the model recognize the subject (Joe’s brother) and the object (sweater) in the sentence.

Introducing the new

Batch_size = 2048



shopping cart for all your paranoid needs...

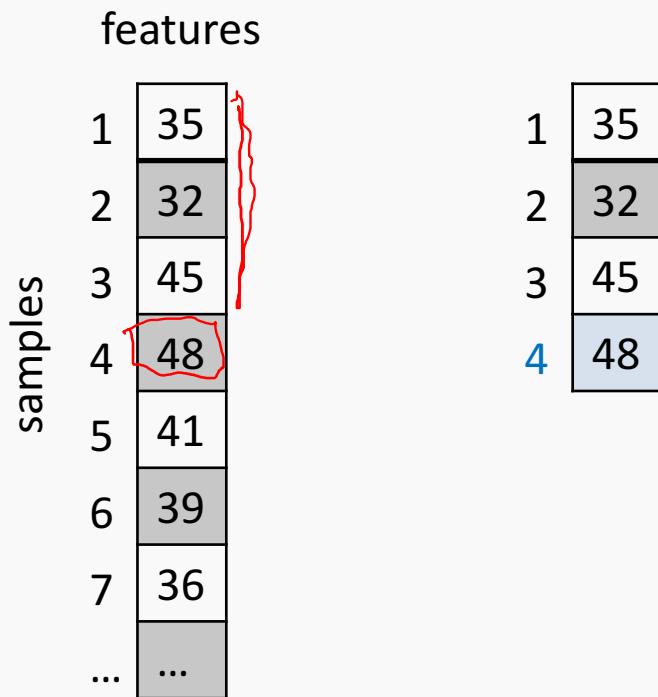
Sequences

- We want a machine learning model to understand sequences, not isolated samples.
- Can MLP do this?
- Assume we have a sequence of temperature measurements and we want to take 3 sequential measurements and predict the next one

	features
samples	
1	35
2	32
3	45
4	48
5	41
6	39
7	36
...	...

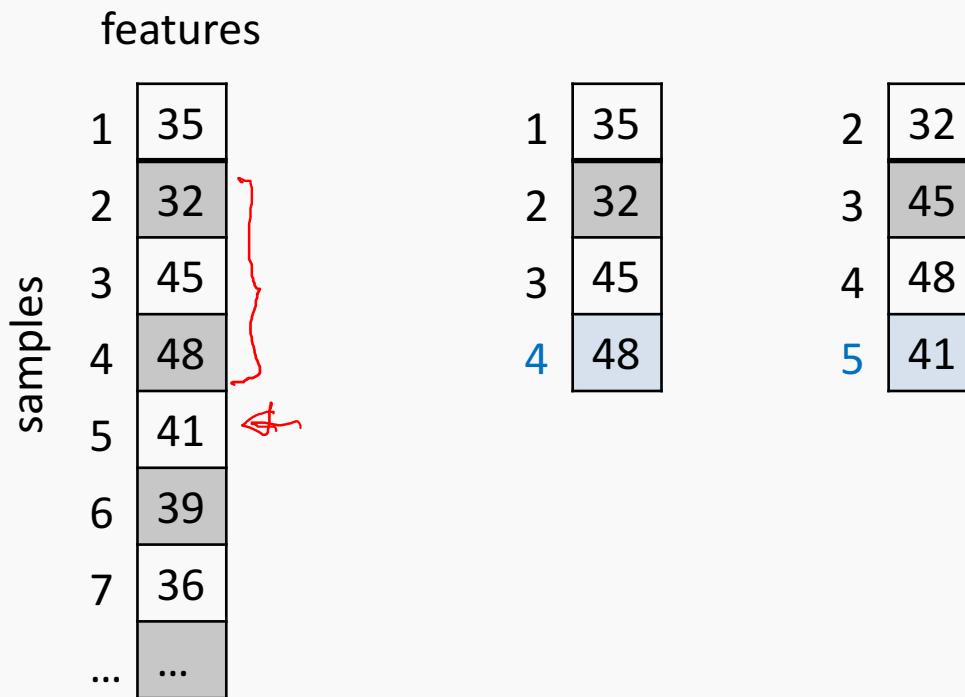
Sequences

- We want a machine learning model to understand sequences, not isolated samples.
- Can MLP do this?
- Assume we have a sequence of temperature measurements and we want to take 3 sequential measurements and predict the next one



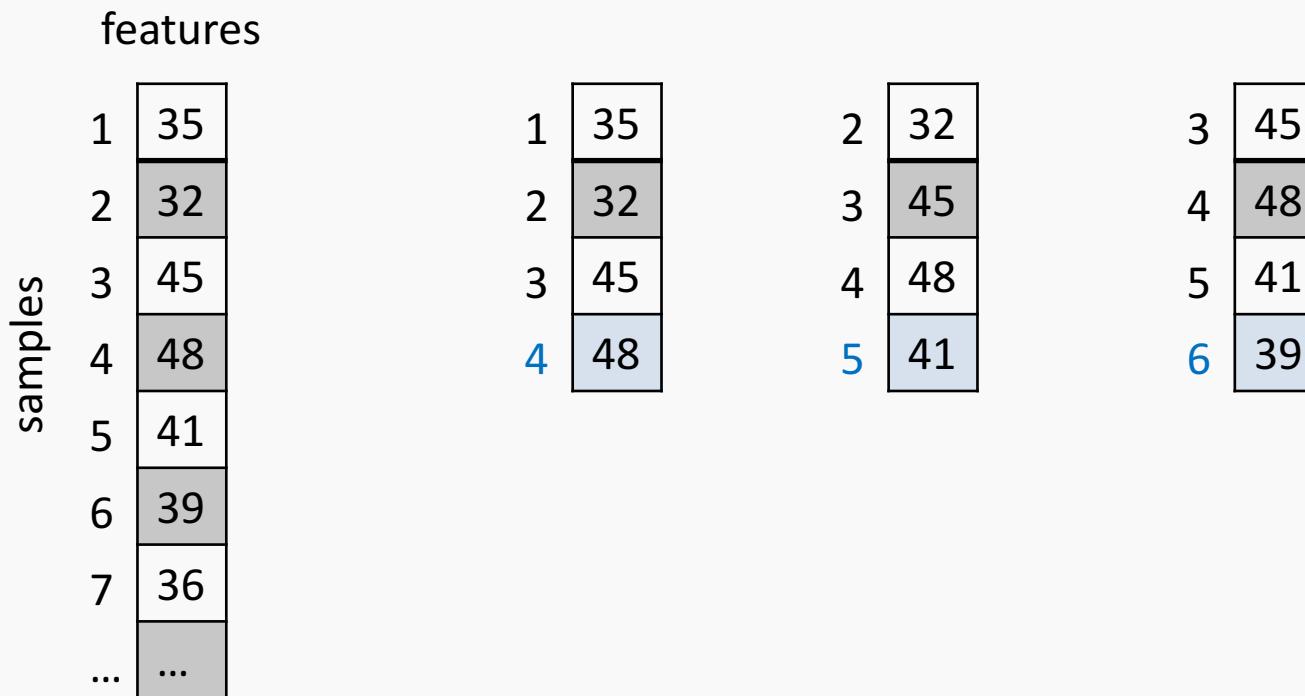
Sequences

- We want a machine learning model to understand sequences, not isolated samples.
- Can MLP do this?
- Assume we have a sequence of temperature measurements and we want to take 3 sequential measurements and predict the next one



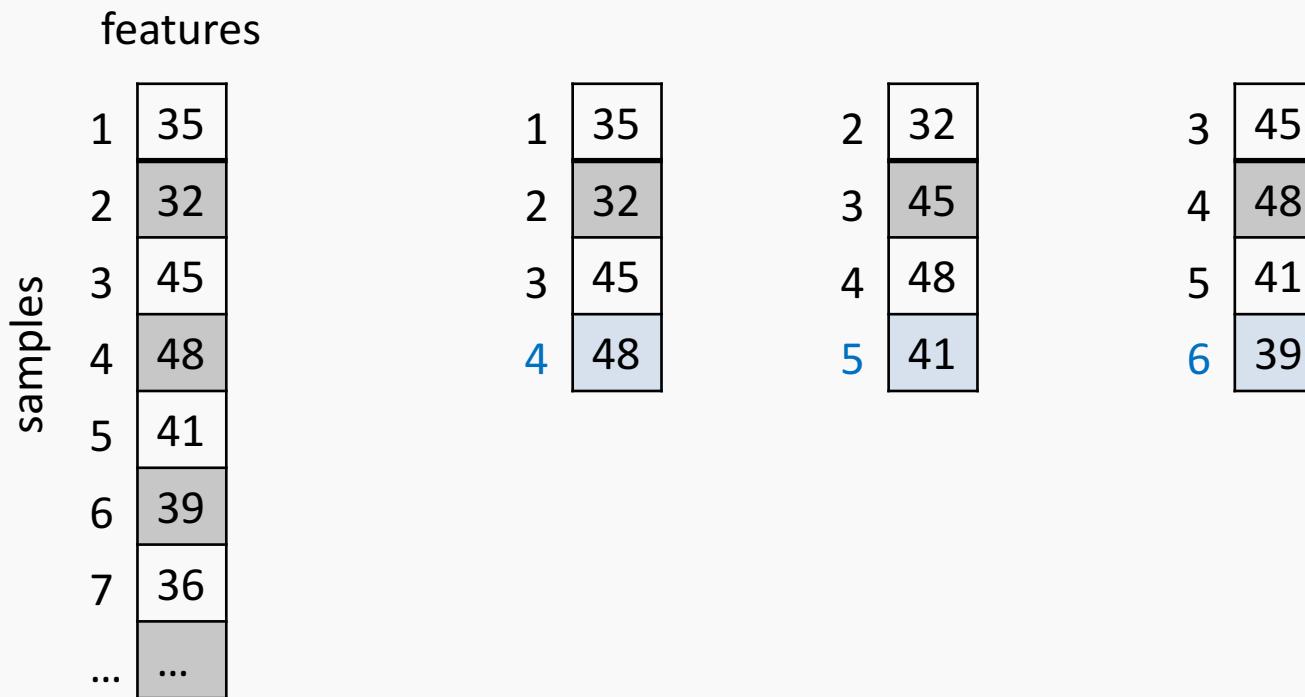
Sequences

- We want a machine learning model to understand sequences, not isolated samples.
- Can MLP do this?
- Assume we have a sequence of temperature measurements and we want to take 3 sequential measurements and predict the next one



Sequences

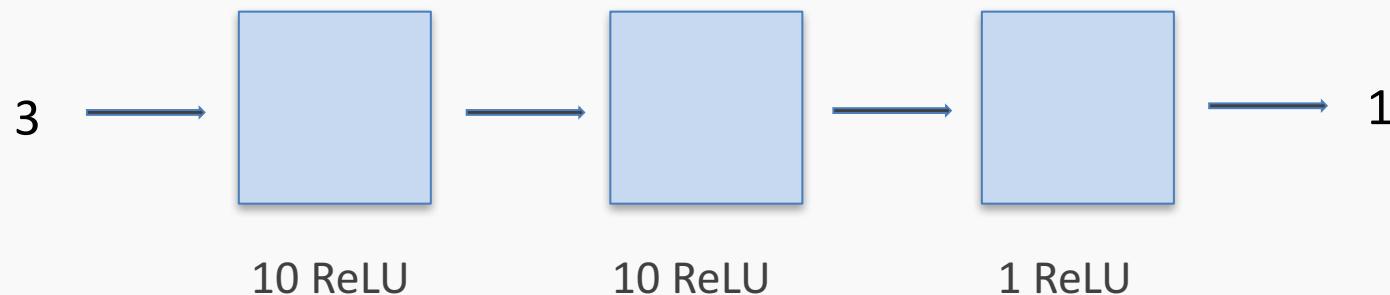
- We want a machine learning model to understand sequences, not isolated samples.
- Can MLP do this?
- Assume we have a sequence of temperature measurements and we want to take 3 sequential measurements and predict the next one



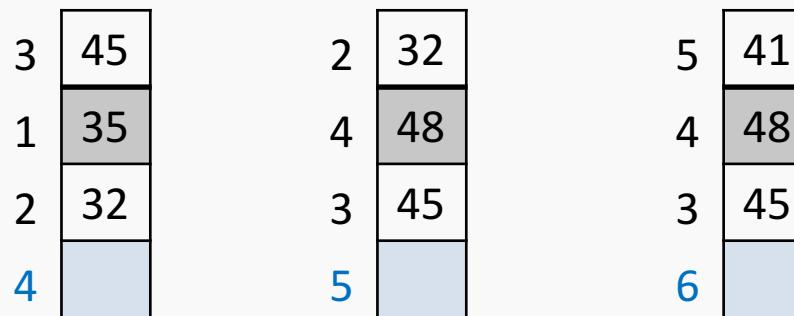
Windowed dataset

This is called **overlapping windowed** dataset, since we're windowing observations to create new.

We can easily do using a MLS:



But re-arranging the order of the inputs like:



will produce the same results



Why not CNNs or MLPs?

1. MLPs/CNNs require fixed input and output size
2. MLPs/CNNs can't classify inputs in multiple places

Windowed dataset

What follows after: `I got in the car and' ?

`drove away'

What follows after: `In car the and I got' ?

Not obvious that it should be `drove away'

The order of words matters. This is true for most sequential data.

A fully connected network will not distinguish the order and therefore missing some information.



A couple of weeks of isolation with the
family. What can go wrong?

Outline

Why RNNs

Main Concept of RNNs

More Details of RNNs

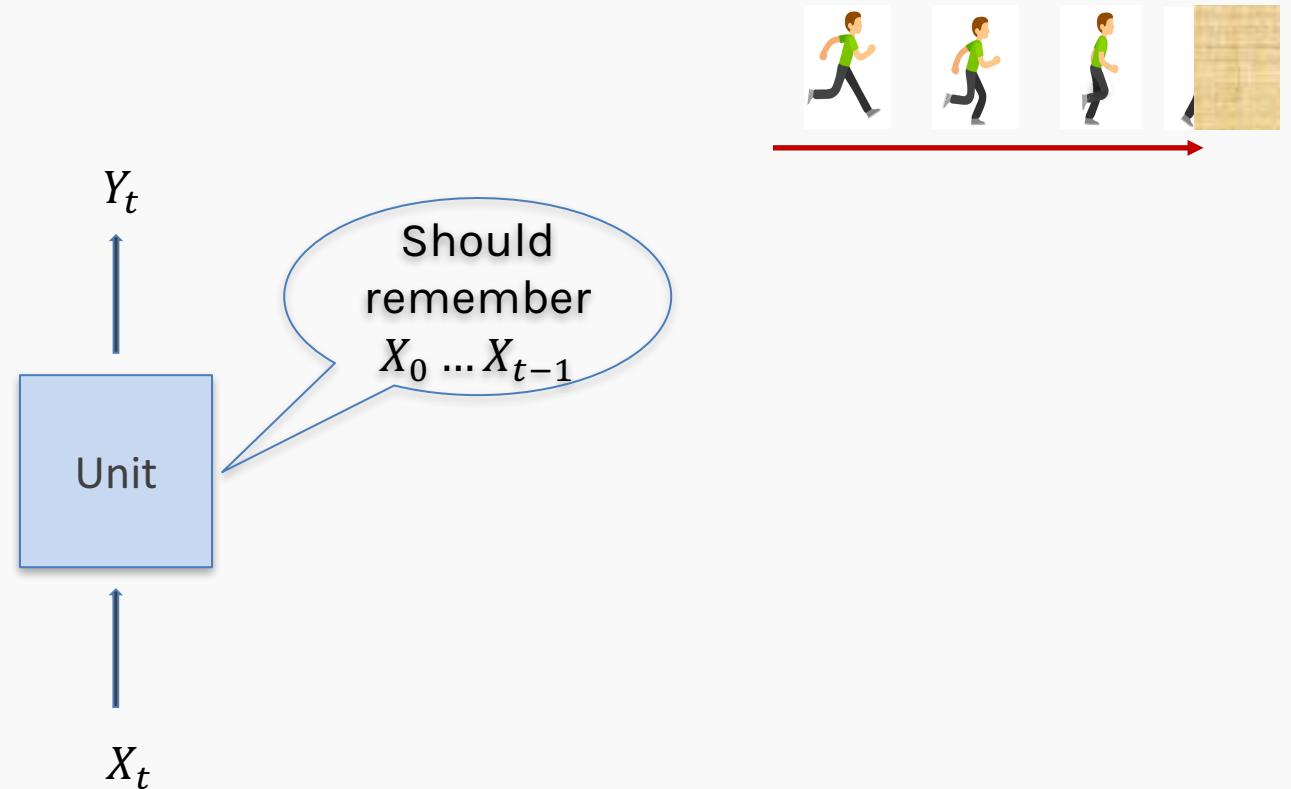
RNN training

Gated RNN



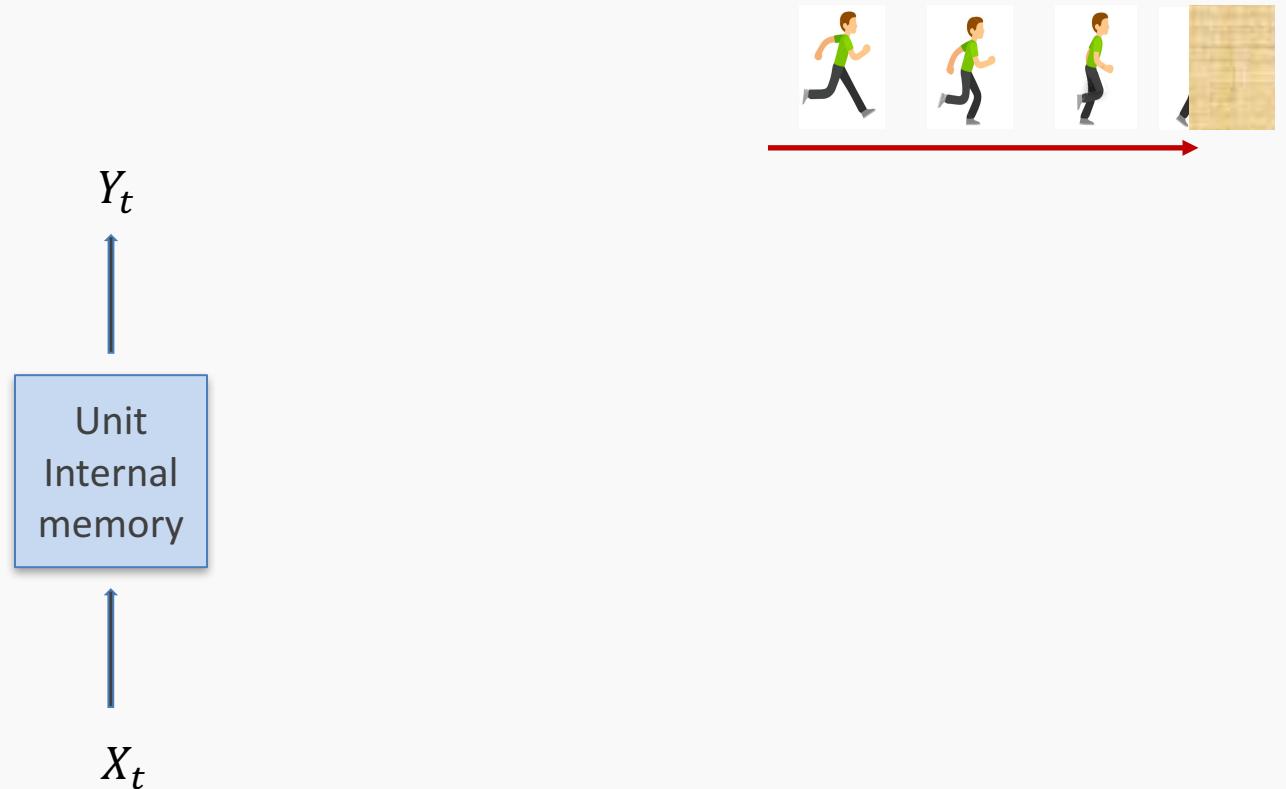
Memory

Somehow the computational unit should remember what it has seen before.



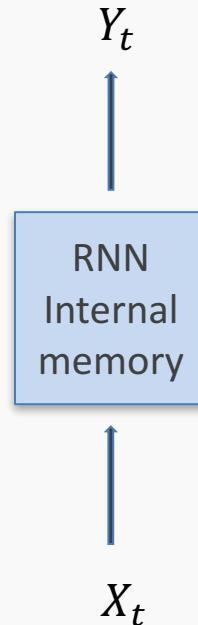
Memory

Somehow the computational unit should remember what it has seen before.



Memory

Somehow the computational unit should remember what it has seen before.
We'll call the information the unit's **state**.



Memory

In neural networks, once training is over, the weights do not change. This means that the network is done learning and done changing.

Then, we feed in values, and it simply applies the operations that make up the network, using the values it has learned.

But the RNN units can remember new information after training has completed.

That is, they're able to keep changing after training is over.

Memory

Question: How can we do this? How can build a unit that remembers the past?

The memory or **state** can be written to a file but in RNNs, we keep it inside the recurrent unit.

In an array or in a vector!

Work with an example:

Anna Sofia said her shoes are too ugly. Her here means Anna Sofia.

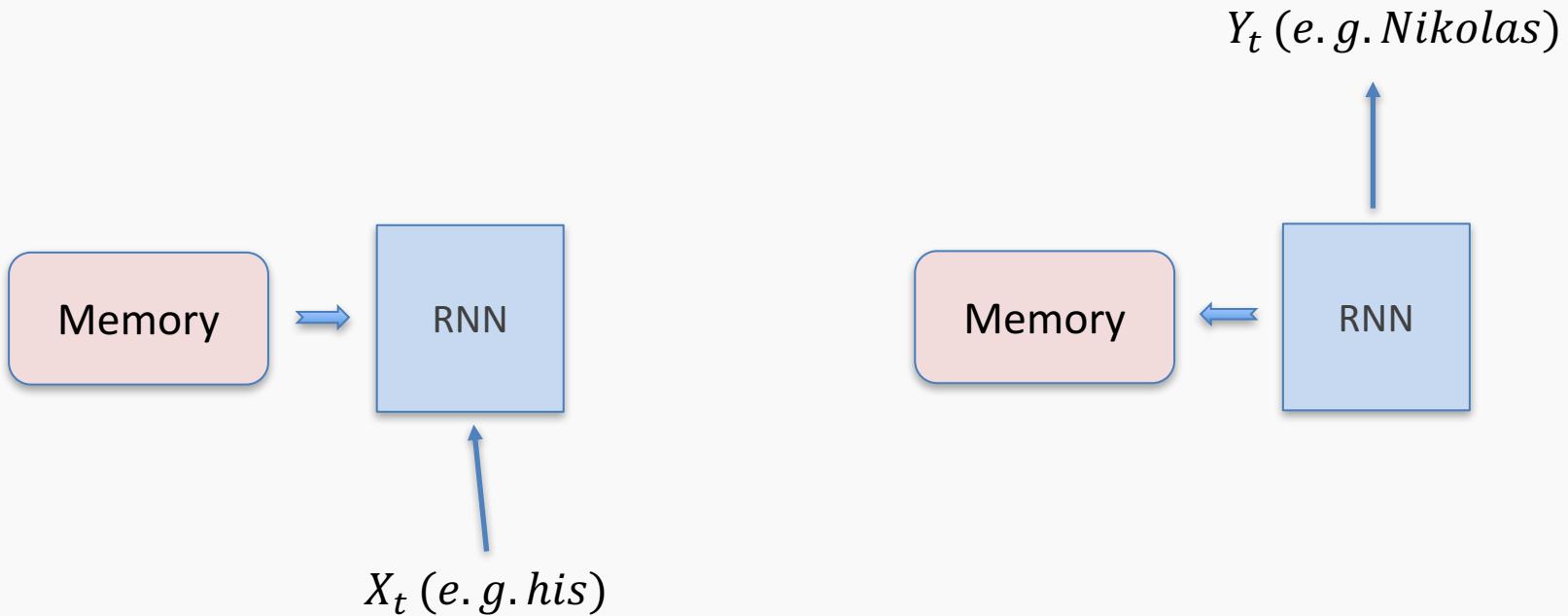
Nikolas put his keys on the table. His here means Nikolas

Memory

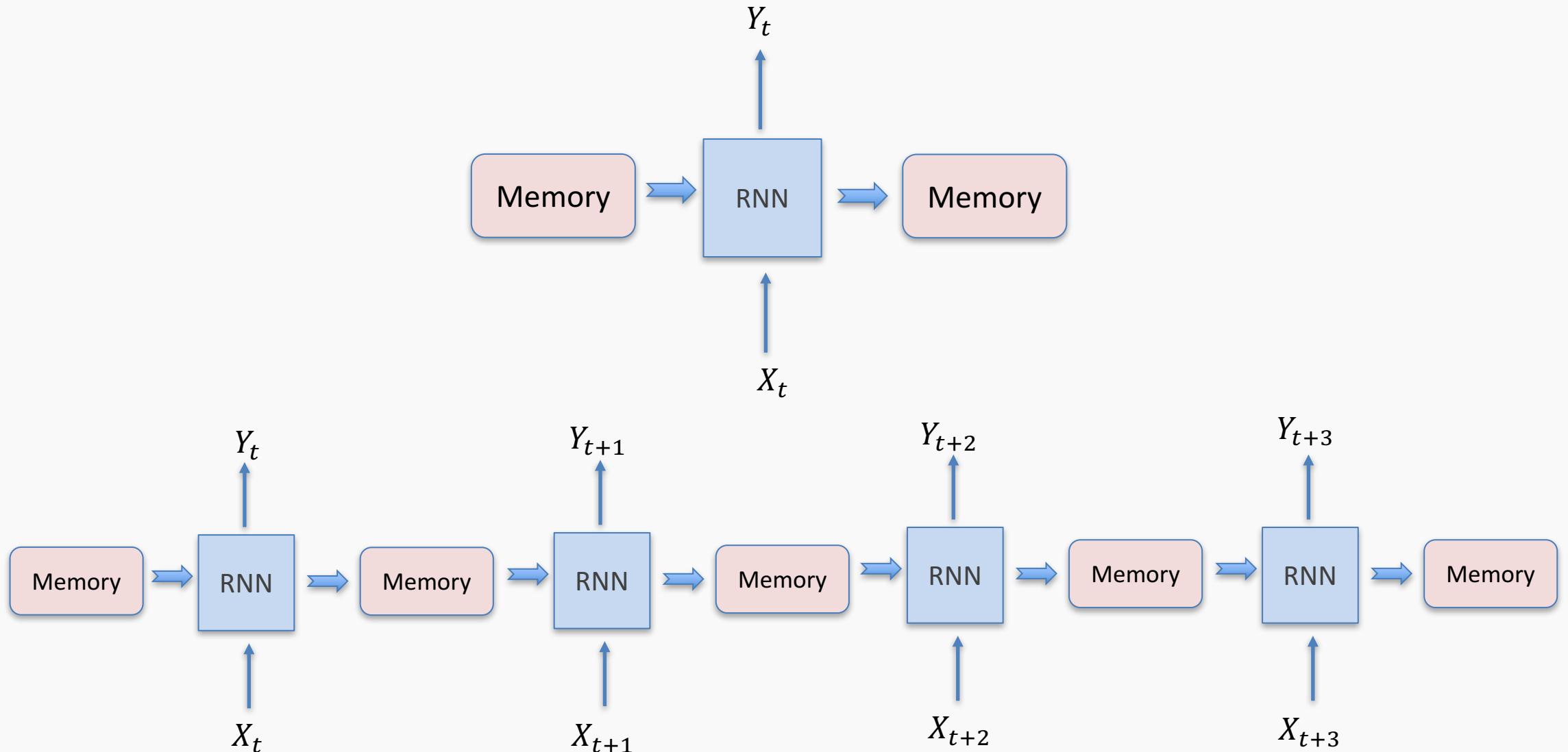
Question: How can we do this? How can we build a unit that remembers the past?

The memory or **state** can be written to a file but in RNNs, we keep it inside the recurrent unit.

In an array or in a vector!



Building an RNN



Outline

Why RNNs

Main Concept of RNNs

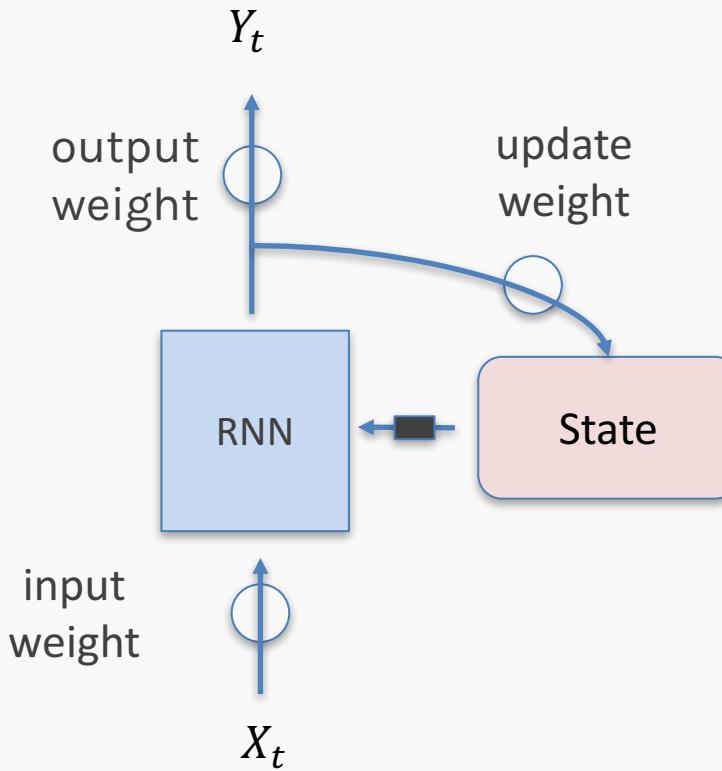
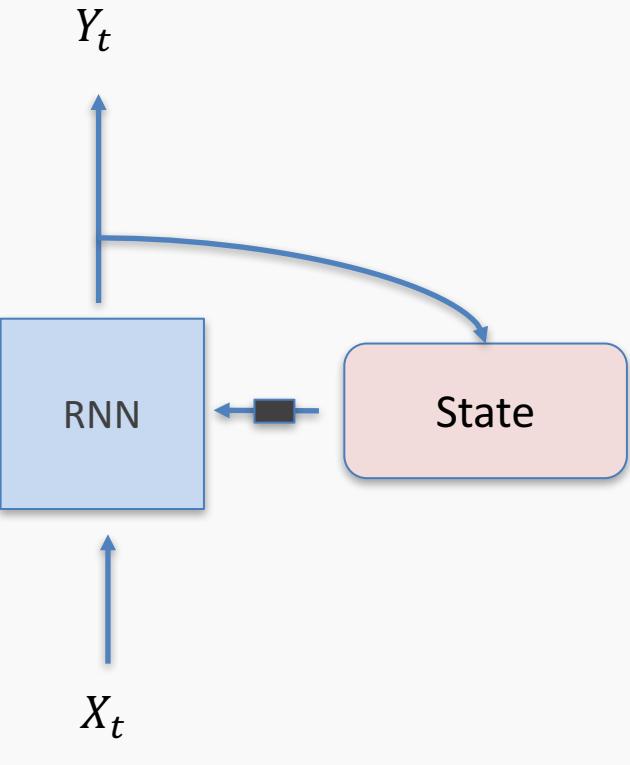
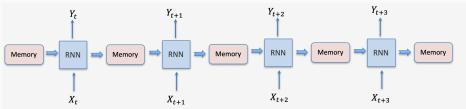
More Details of RNNs

RNN training

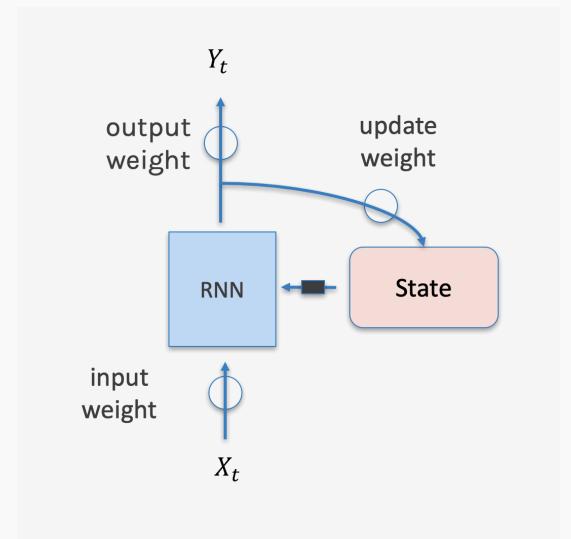
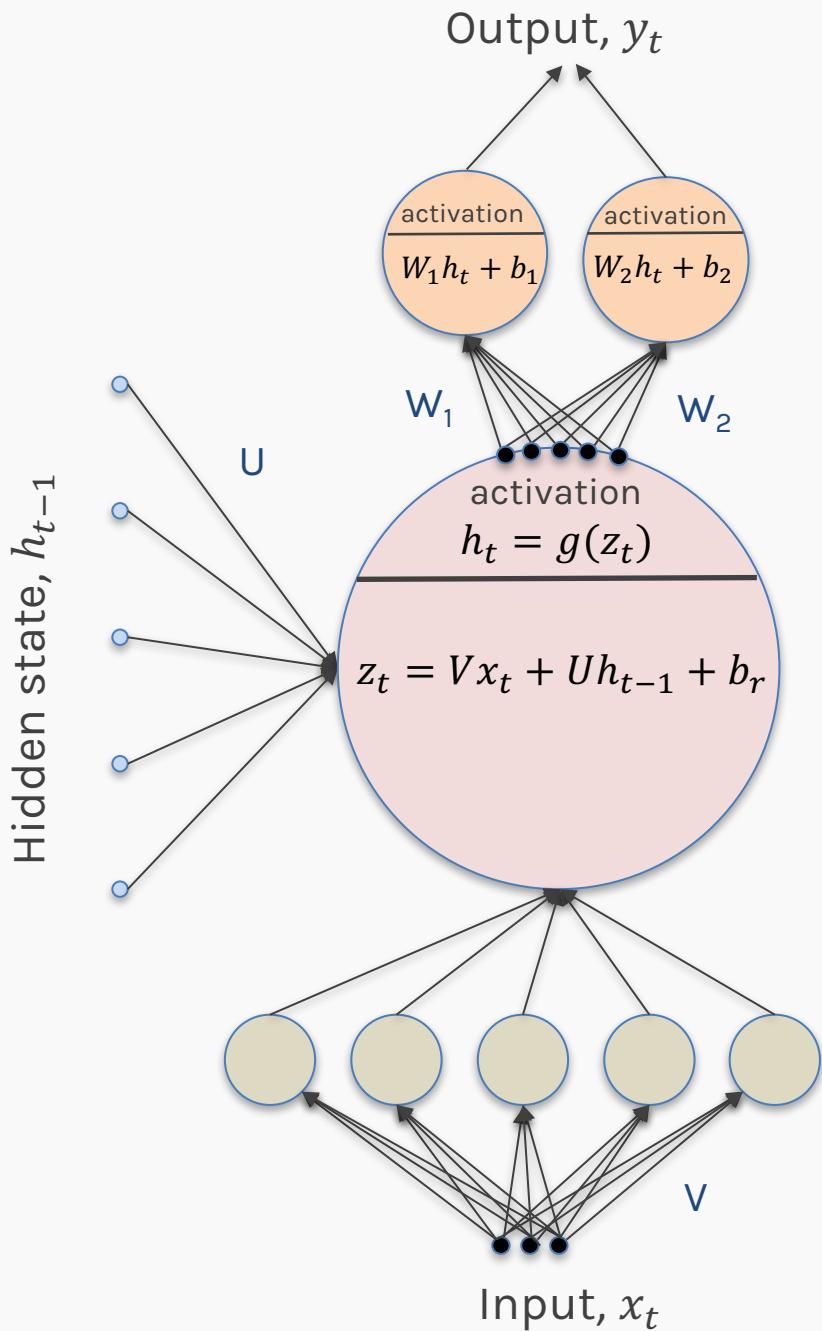
Gated RNN



Structure of an RNN cell



Anatomy of an RNN unit



Outline

Why RNNs

Main Concept of RNNs

More Details of RNNs

RNN training

Gated RNN

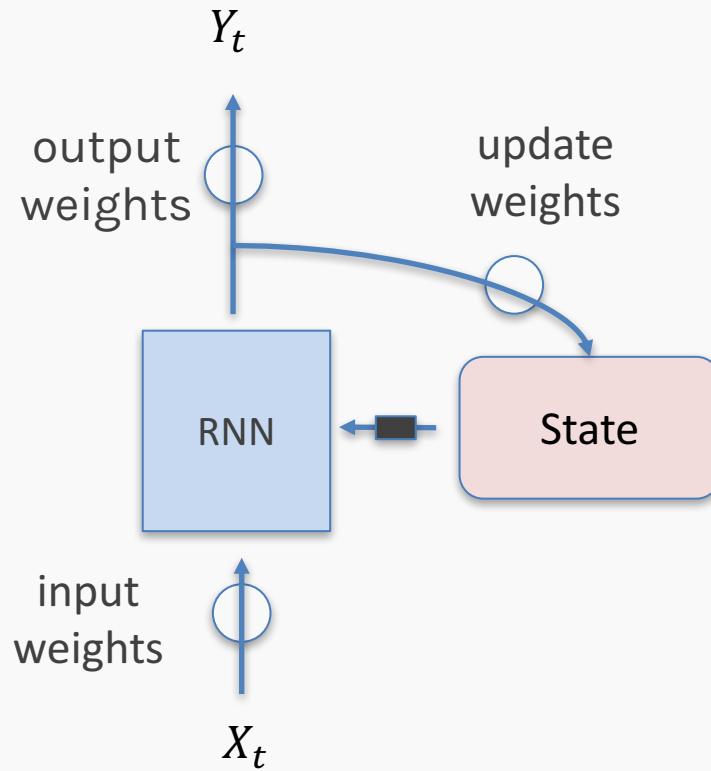


Backprop Through Time

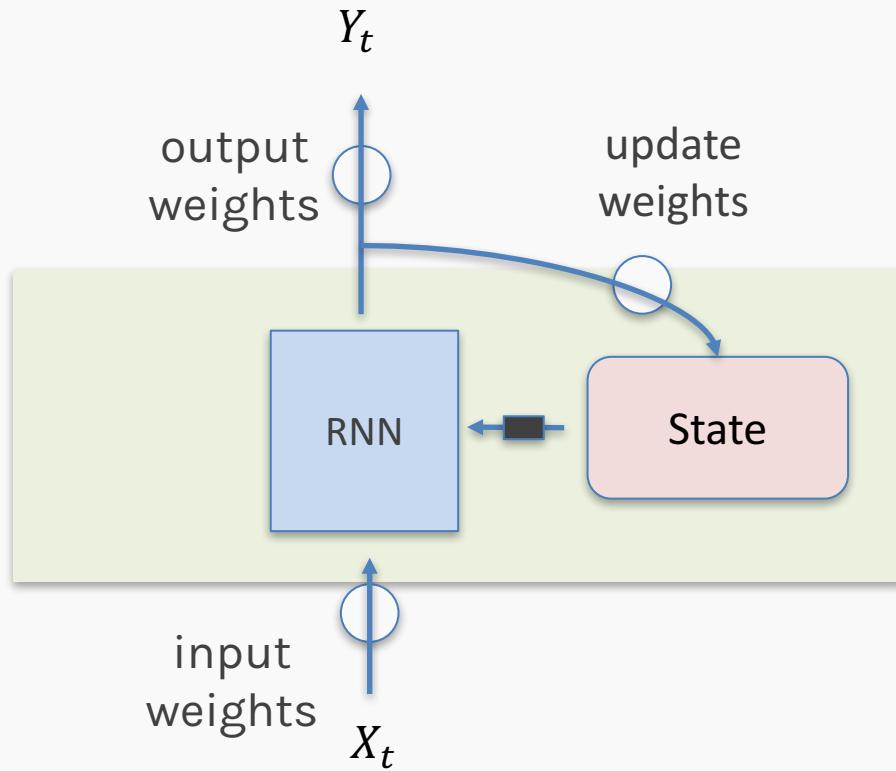
- For each input, **unfold network for the sequence length T**
- Back-propagation: apply forward and backward pass on unfolded network
- **Memory cost: $O(T)$**



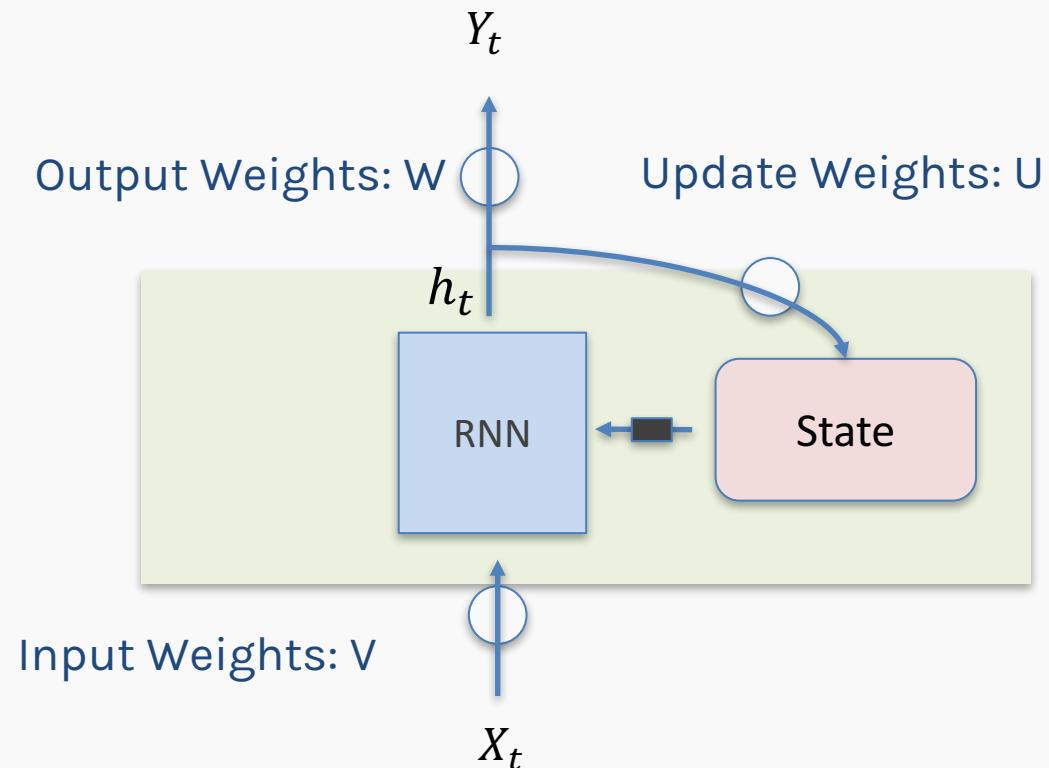
Backprop Through Time



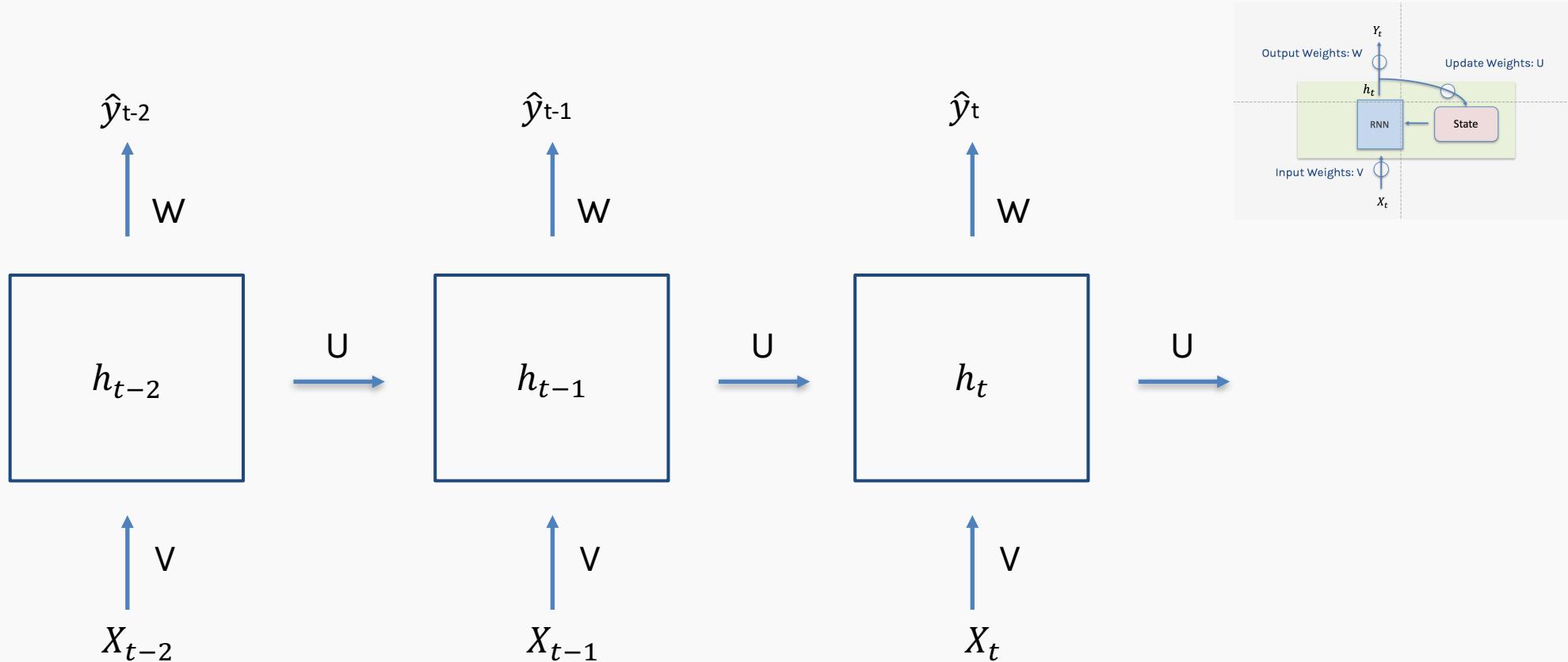
Backprop Through Time



Backprop Through Time

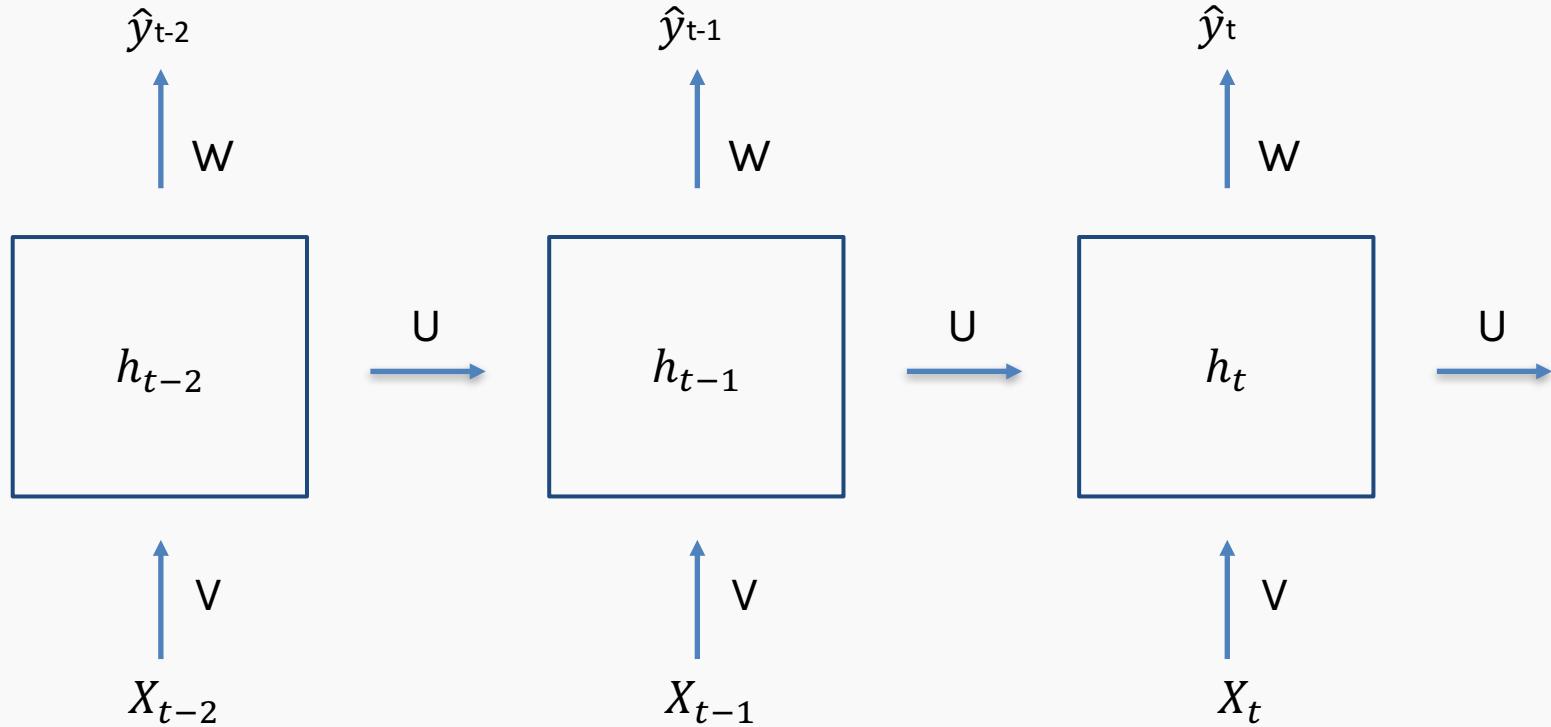


Backprop Through Time



You have two activation functions g_h which serves as the activation for the hidden state and g_y which is the activation of the output. In the example shown before g_y was the identity.

Backprop Through Time



Backprop Through Time

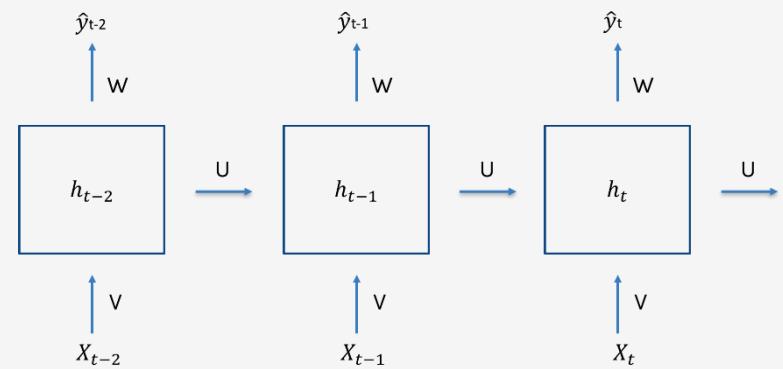
$$\hat{y}_t = g_y(Wh_t + b)$$

$$L = \sum_t L_t$$

$$L_t = L_t(\hat{y}_t)$$

$$\frac{dL}{dW} = \sum_t \frac{dL_t}{dW} = \sum_t \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial W}$$

$$\frac{\partial \hat{y}_t}{\partial W} = g_y' h_t$$



Backprop Through Time

$$\hat{y}_t = g_y(Wh_t + b)$$

$$h_t = g_h(Vx_t + Uh_{t-1} + b')$$

$$\hat{y}_t = g_y(Wg_h(Vx_t + Uh_{t-1} + b') + b)$$

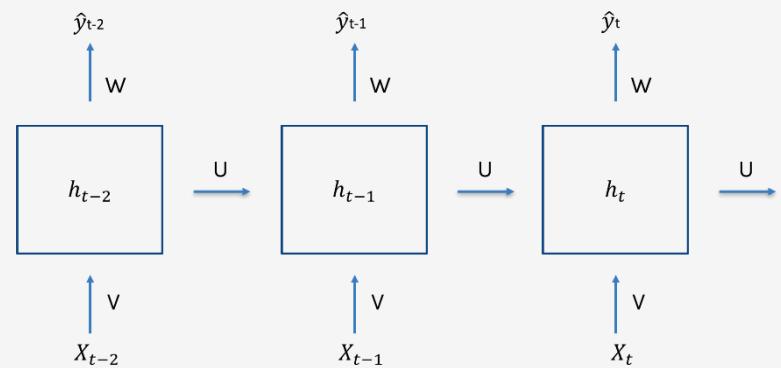
$$L = \sum_t L_t \quad L_t = L_t(\hat{y}_t)$$

$$\frac{dL}{dU} = \sum_t \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial U}$$

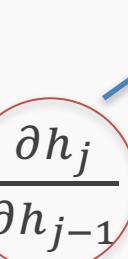
$$\frac{\partial h_t}{\partial U} = \sum_{k=1}^t \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial U}$$

$$\frac{\partial h_t}{\partial h_k} = \frac{\partial h_t}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial h_{t-2}} \dots \frac{\partial h_{k+1}}{\partial h_k} = \prod_{j=k+1}^t \frac{\partial h_j}{\partial h_{j-1}}$$

$$\frac{\partial L_t}{\partial U} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \left(\frac{dh_t}{dU} + \frac{dh_t}{dh_{t-1}} \frac{dh_{t-1}}{dU} + \frac{dh_t}{dh_{t-1}} \frac{dh_{t-1}}{dh_{t-2}} \frac{dh_{t-2}}{dU} + \dots \right)$$



$$\frac{\partial h_j}{\partial h_{j-1}} = g'_h U$$



“Quarantine day 6”



Gradient Clipping

Prevents exploding gradients

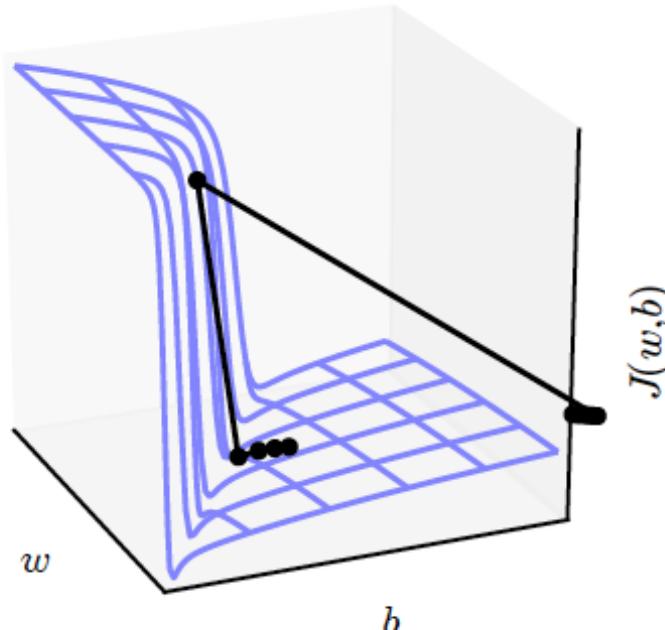
Clip the norm of gradient before update.

For some derivative g , and some threshold u

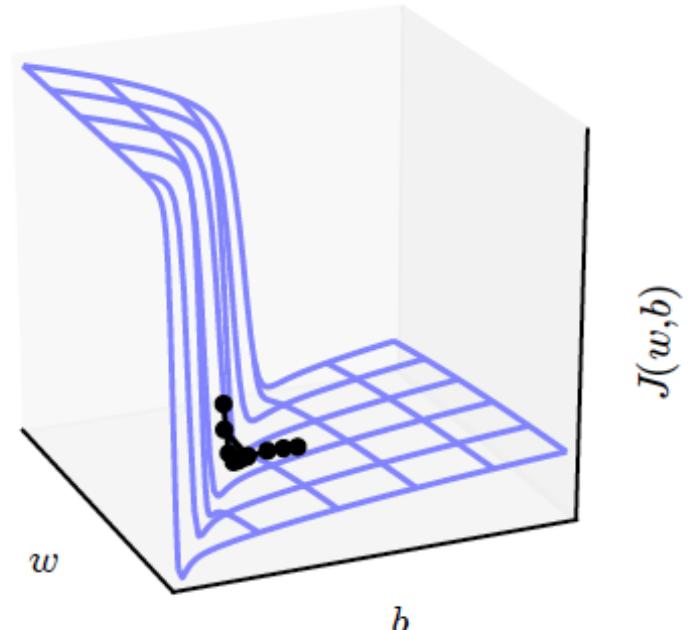
if $\|g\| > u$

$$g \leftarrow \frac{gu}{\|g\|}$$

Without clipping



With clipping



Outline

Why RNNs

Main Concept of RNNs

More Details of RNNs

RNN training

Gated RNN



Long-term Dependencies

Unfolded networks can be very deep

Long-term interactions are given exponentially smaller weights than small-term interactions

Gradients tend to either *vanish* or *explode*

Long Short-Term Memory

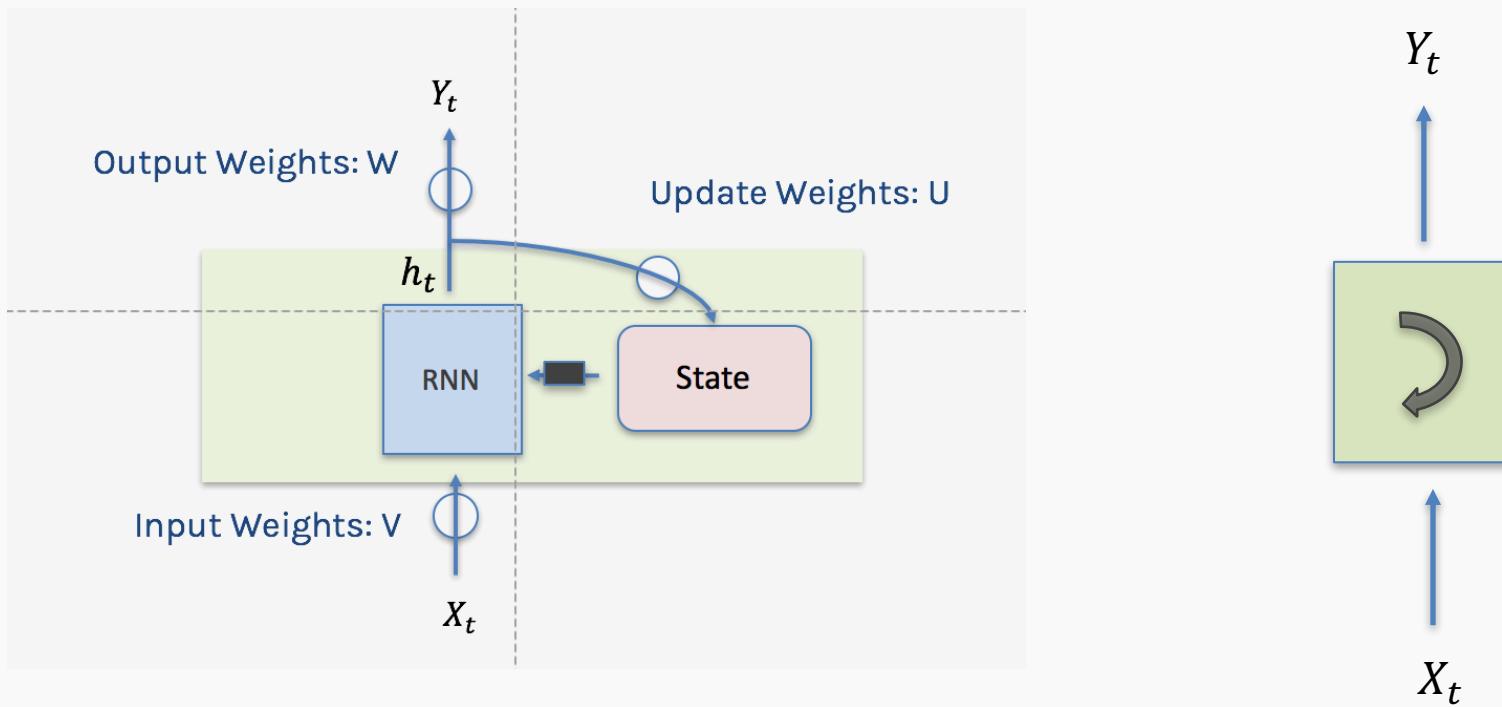
Handles long-term dependencies

Leaky units where weight on self-loop α is context-dependent

Allow network to decide whether to accumulate or forget past info

Notation

Using conventional and convenient notation



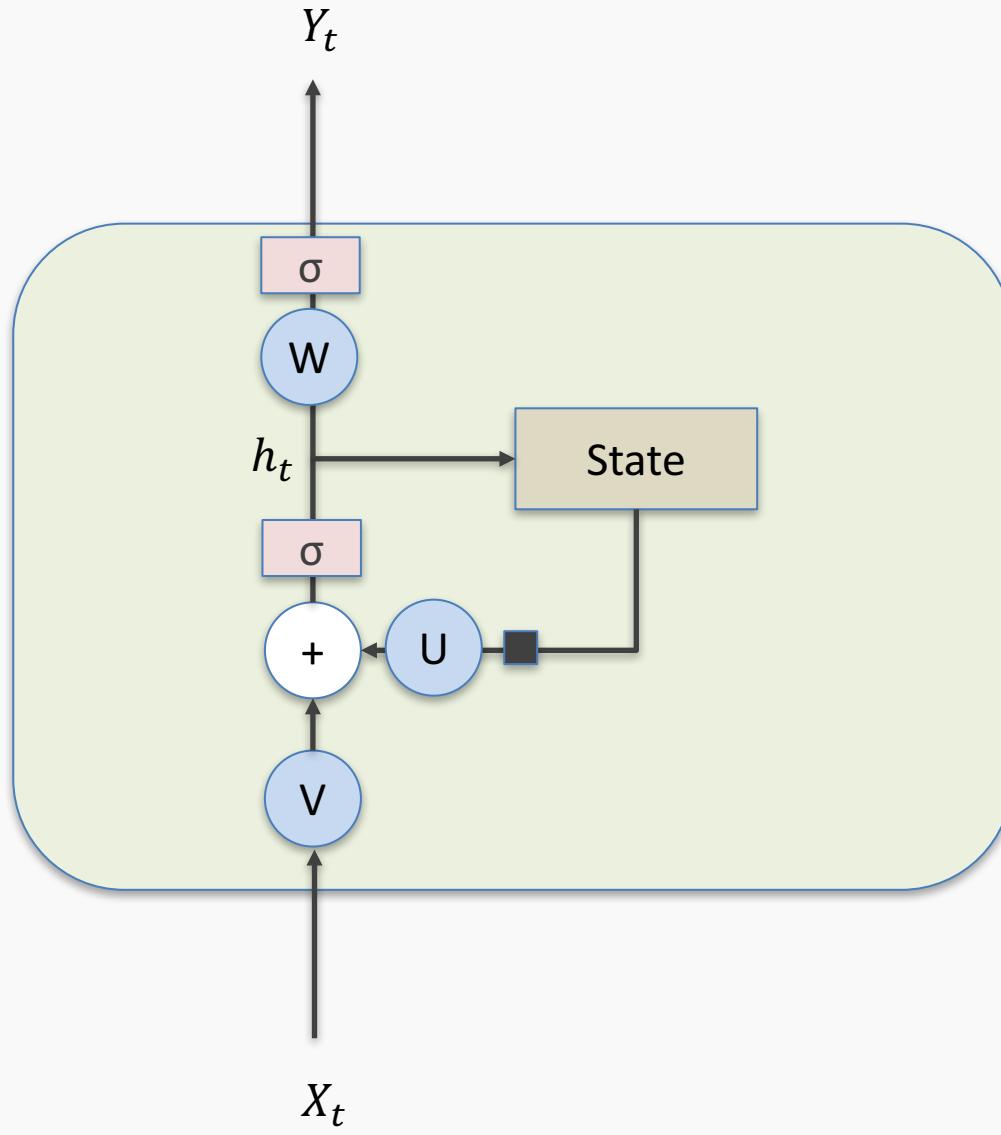
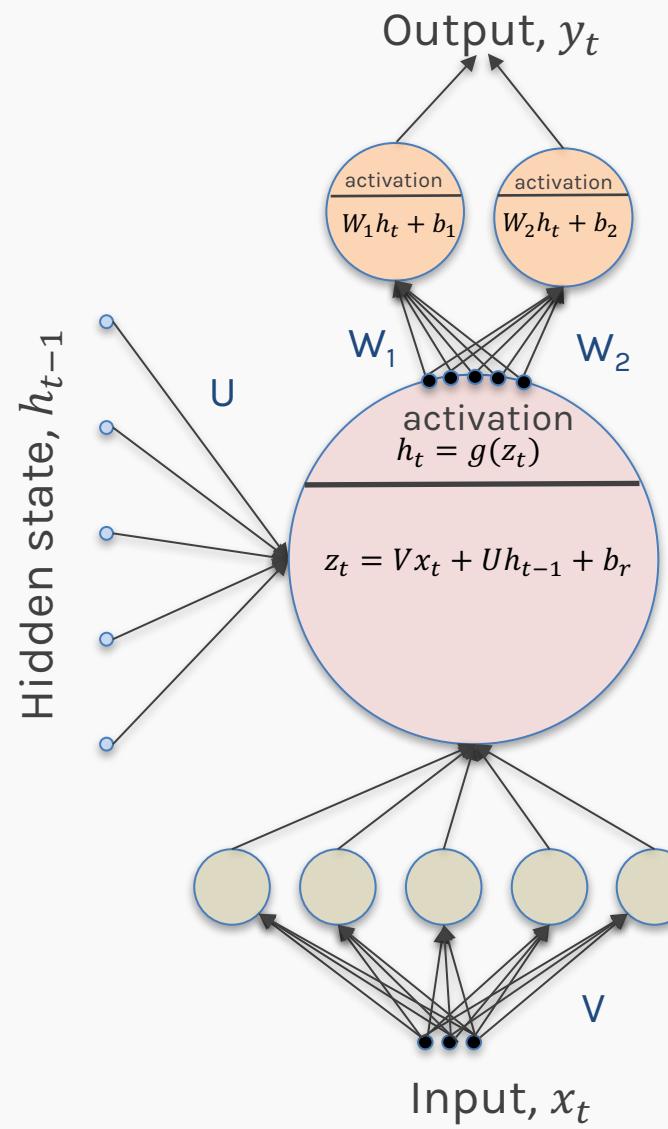
**when your basic RNN isn't
capable of catching long-term
dependencies**



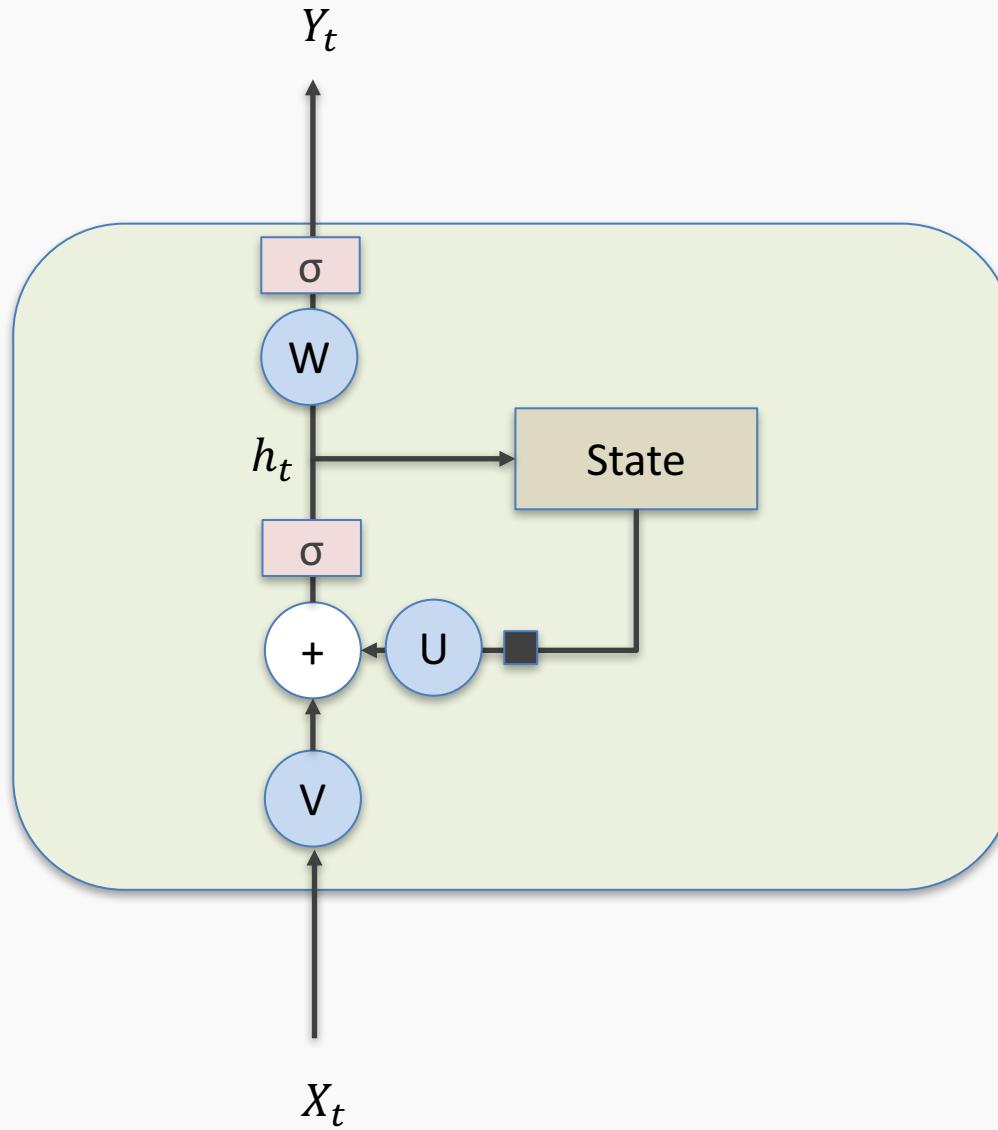
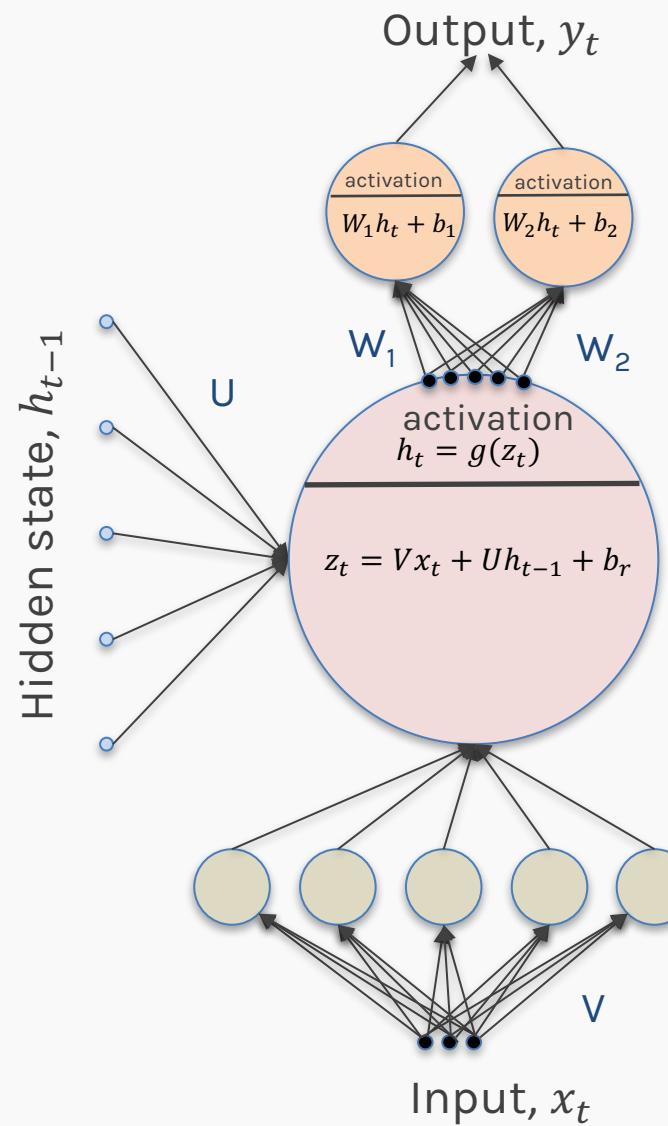
Things are about to get GRUesome

CS109B, PROTOPAPAS, GLICKMAN, TANNER

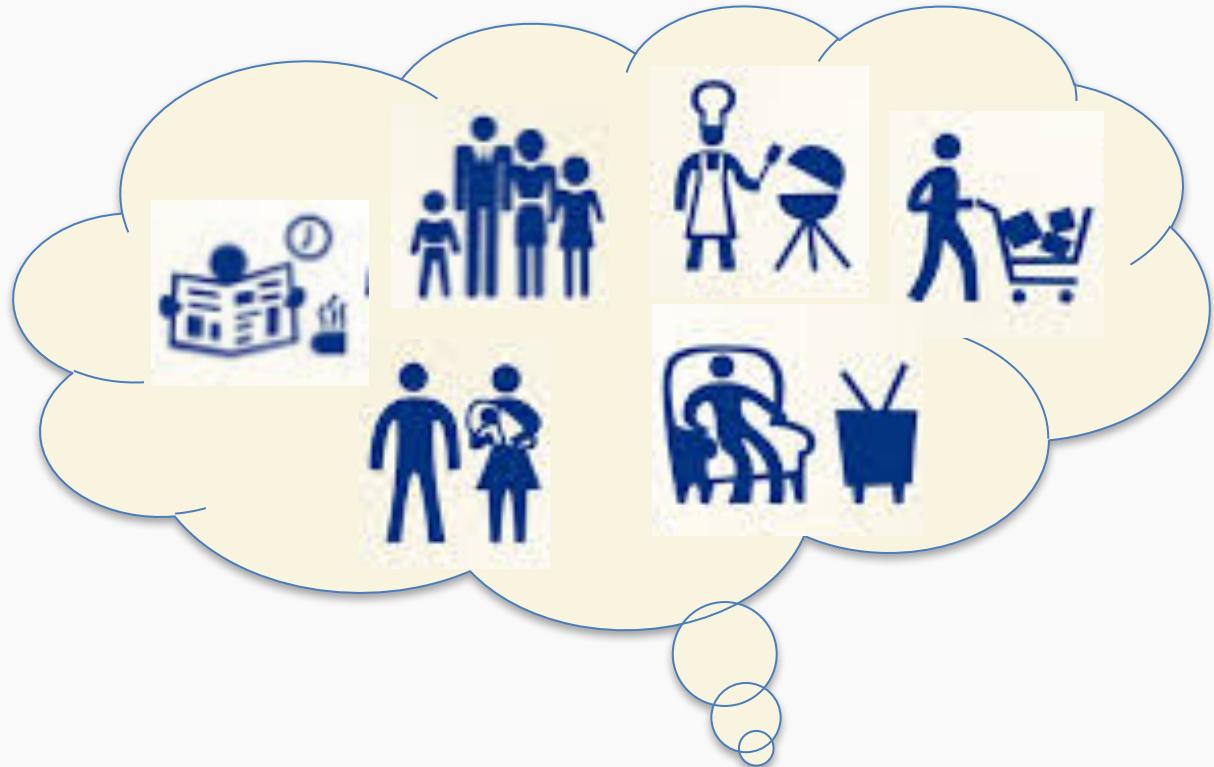
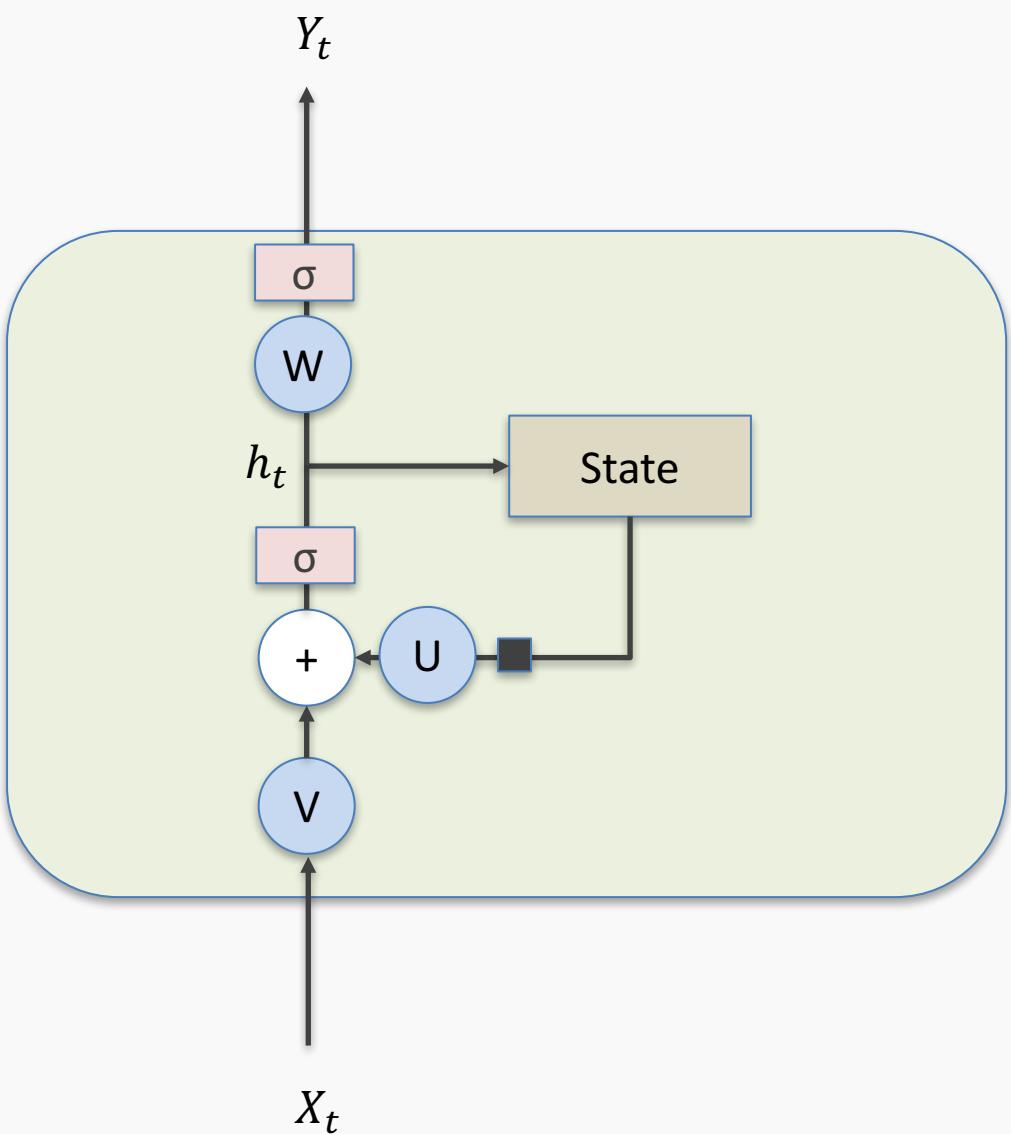
Simple RNN again



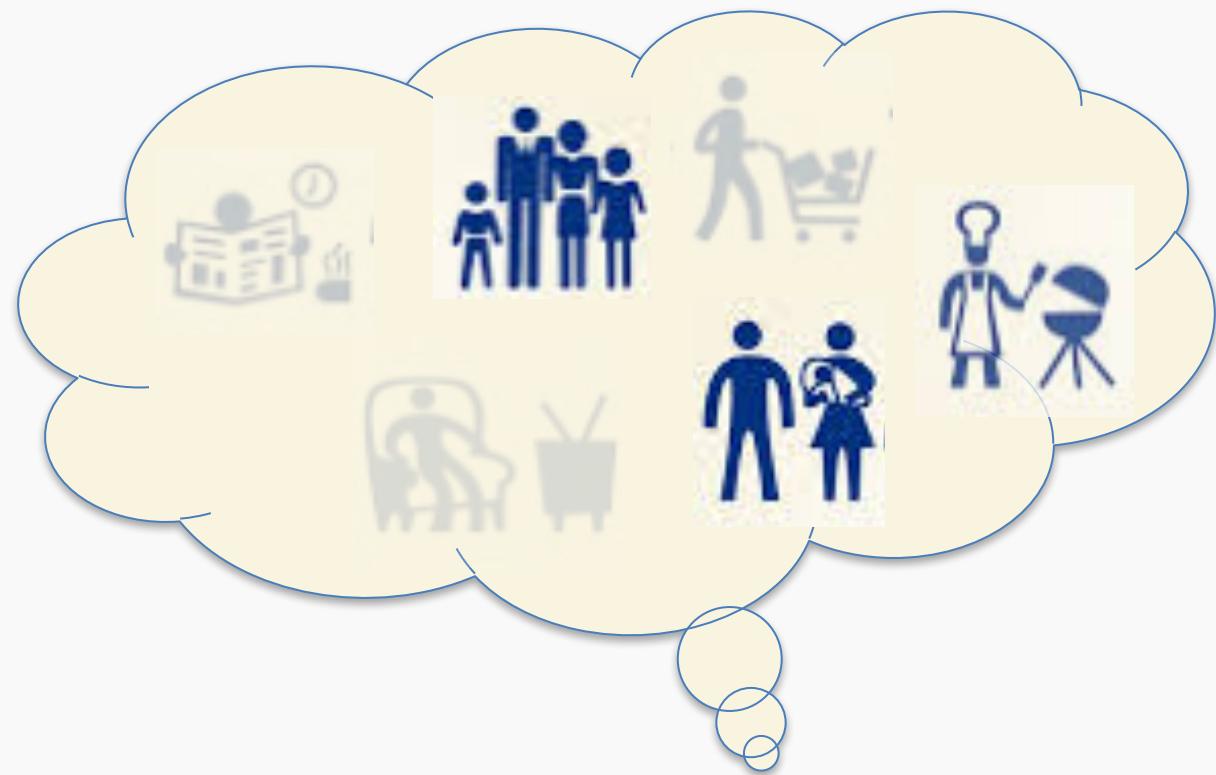
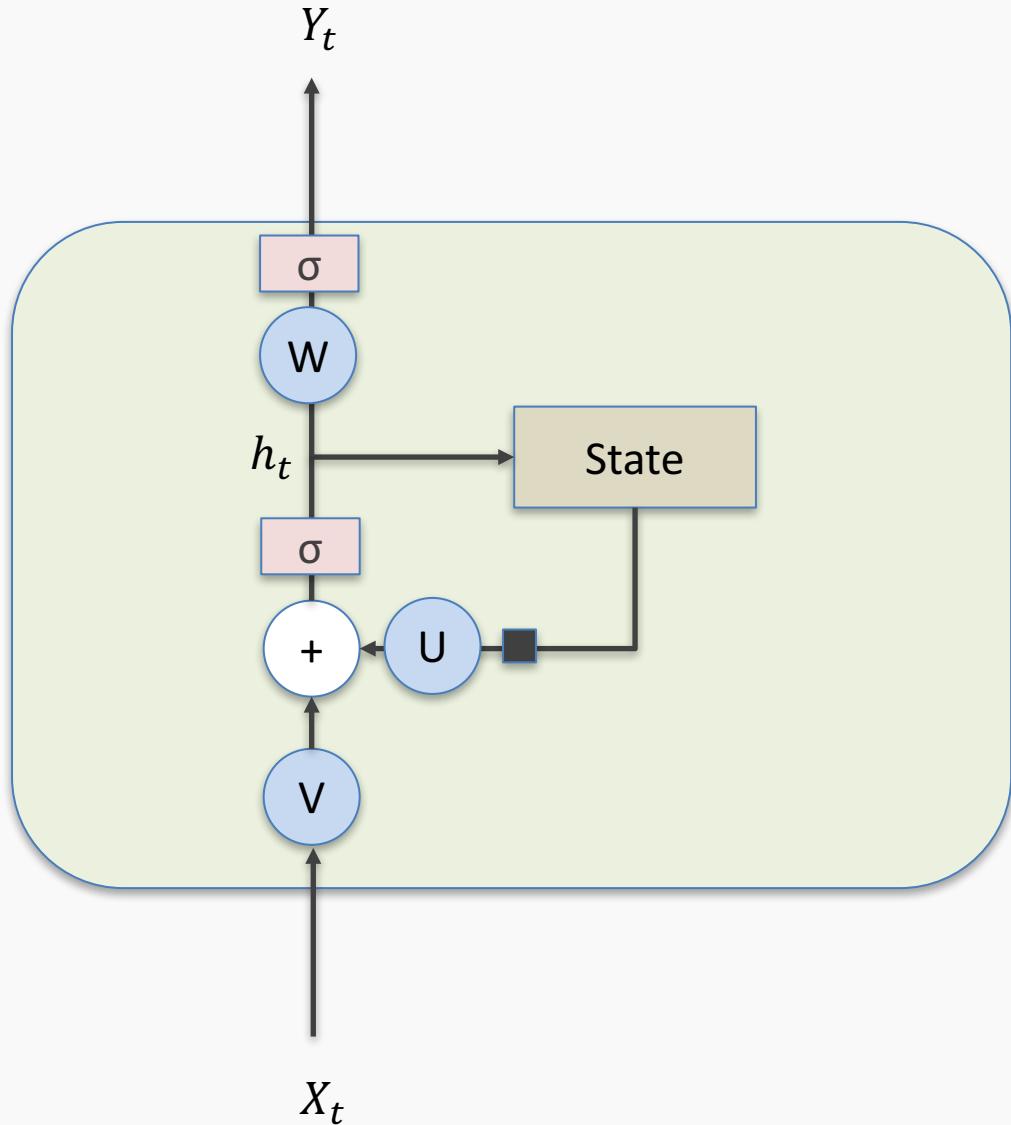
Simple RNN again



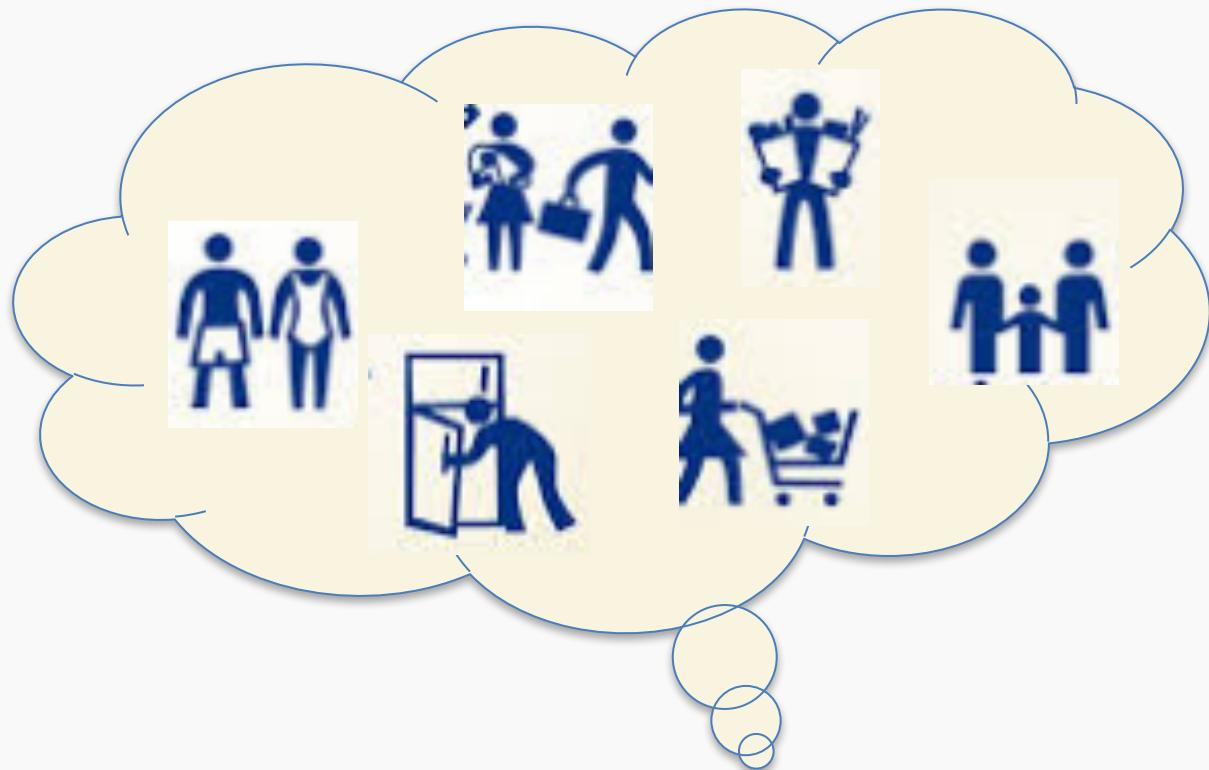
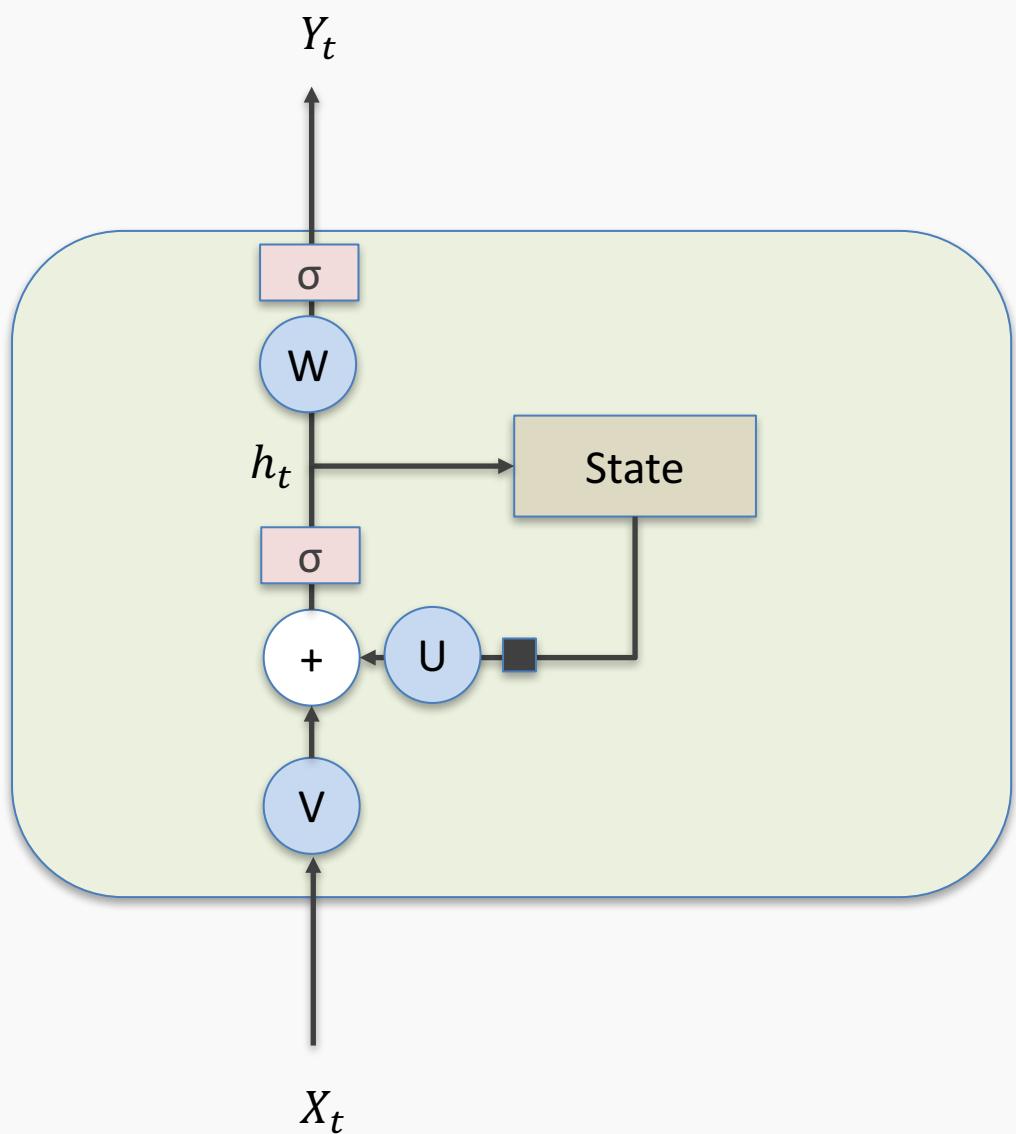
Simple RNN again: Memories



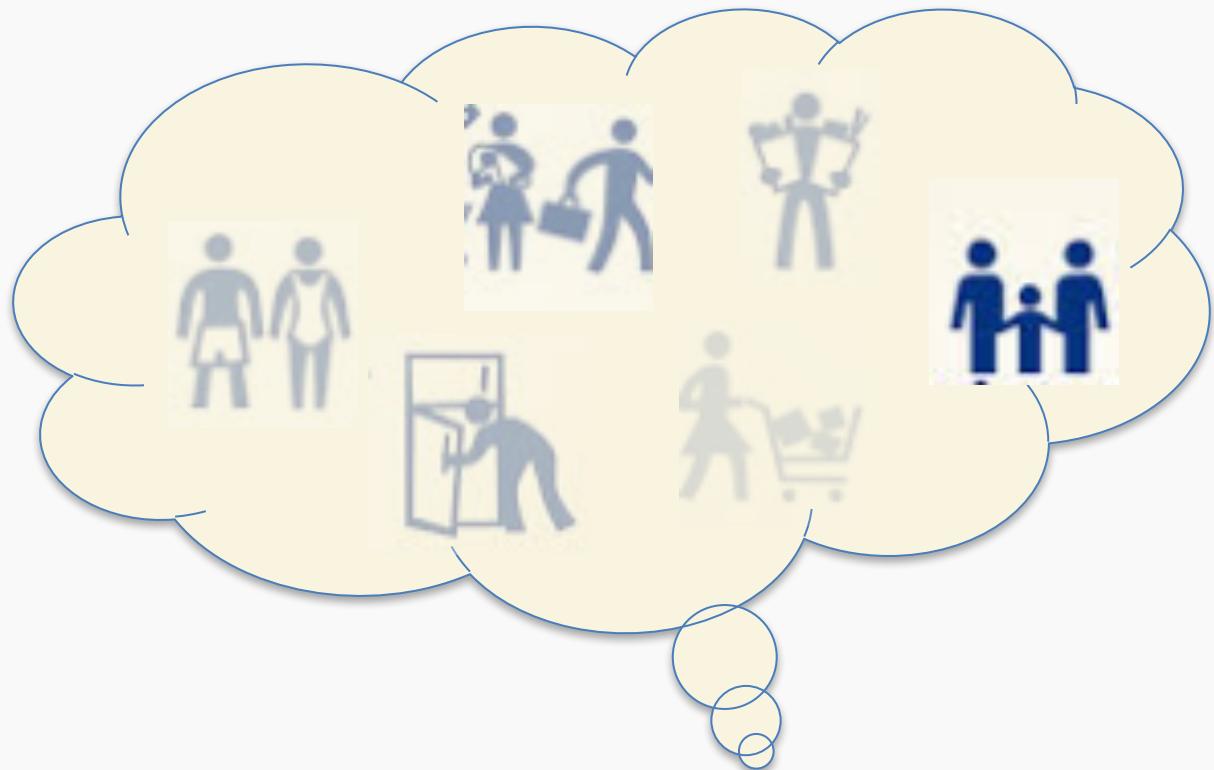
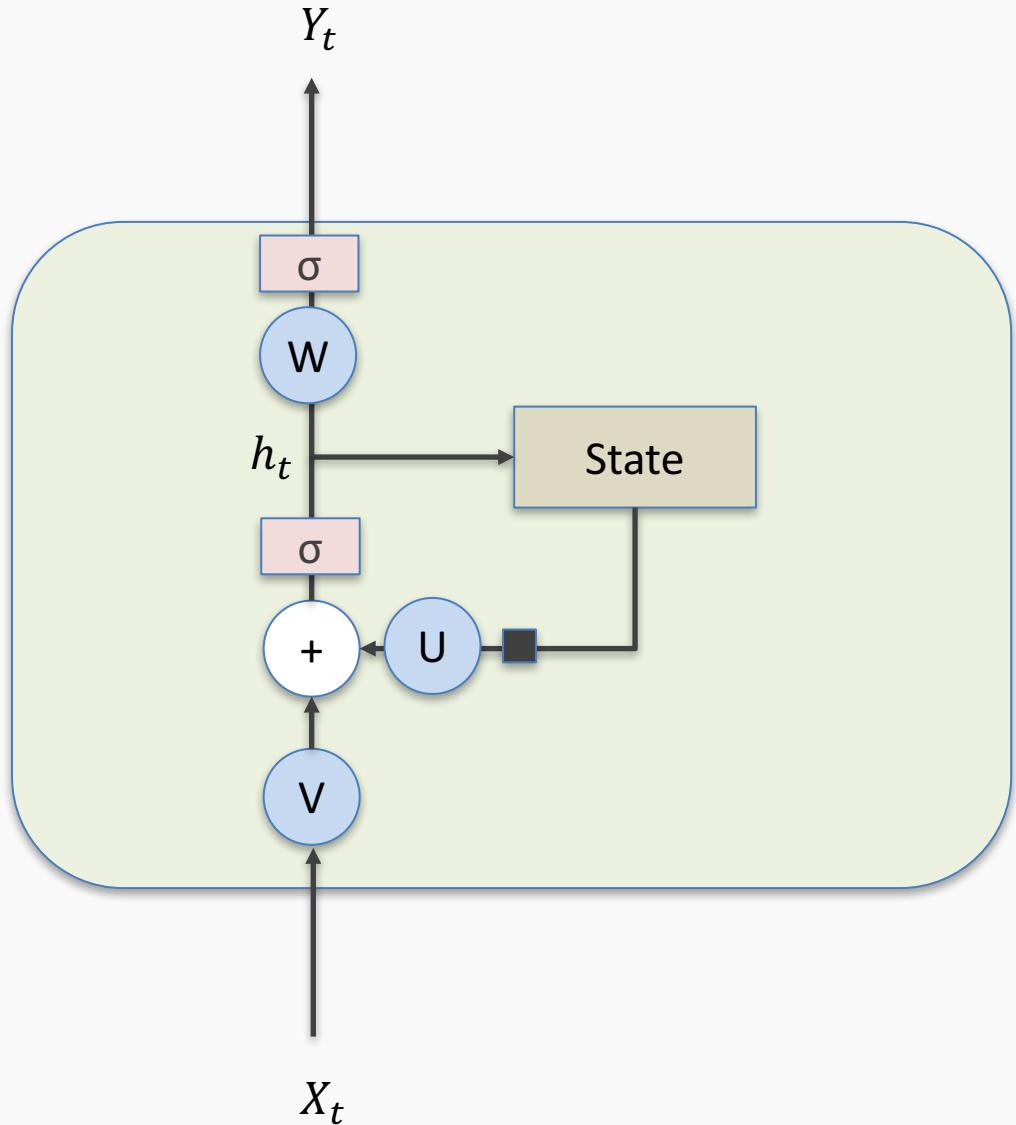
Simple RNN again: Memories - Forgetting



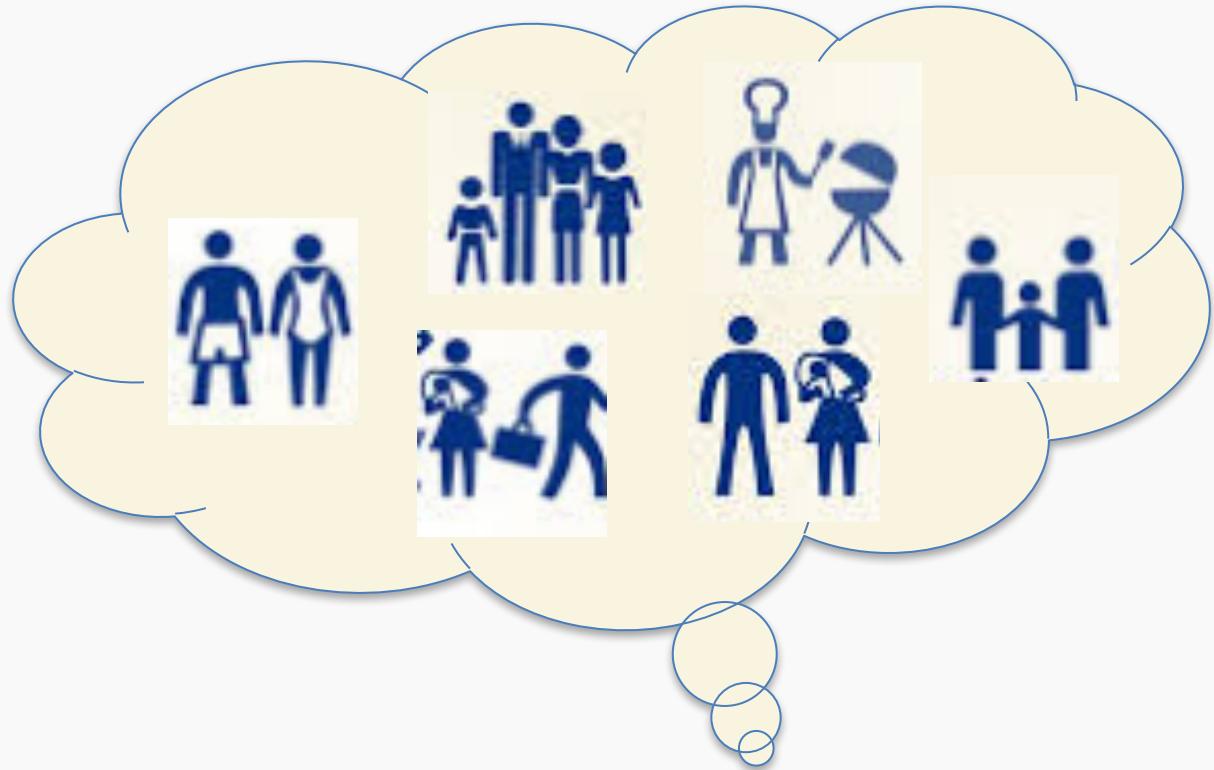
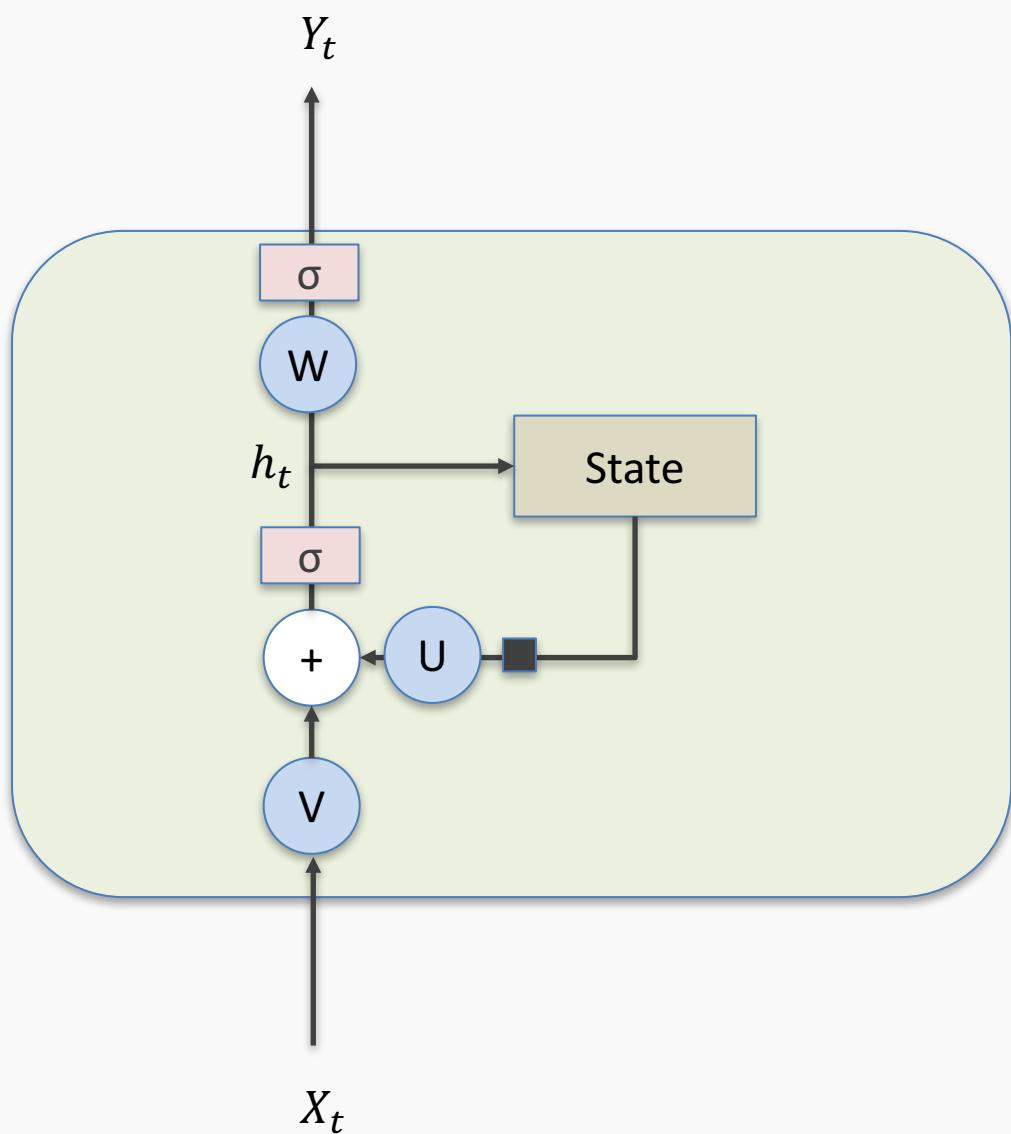
Simple RNN again: New Events



Simple RNN again: New Events Weighted



Simple RNN again: Updated memories



Ref

- [Chen17b] Qiming Chen, Ren Wu, “CNN Is All You Need”, arXiv 1712.09662, 2017.
<https://arxiv.org/abs/1712.09662>
- [Chu17] Hang Chu, Raquel Urtasun, Sanja Fidler, “Song From PI: A Musically Plausible Network for Pop Music Generation”, arXiv preprint, 2017.
<https://arxiv.org/abs/1611.03477>
- [Johnson17] Daniel Johnson, “Composing Music with Recurrent Neural Networks”, Heahedria, 2017. <http://www.hexahedria.com/2015/08/03/composing-music-with-recurrent-neural-networks/>
- [Deutsch16b] Max Deutsch, “Silicon Valley: A New Episode Written by AI”, Deep Writing blog post, 2017. <https://medium.com/deep-writing/silicon-valley-a-new-episode-written-by-ai-a8f832645bc2>
- [Fan16] Bo Fan, Lijuan Wang, Frank K. Soong, Lei Xie “Photo-Real Talking Head with Deep Bidirectional LSTM”, Multimedia Tools and Applications, 75(9), 2016.
https://www.microsoft.com/en-us/research/wp-content/uploads/2015/04/icassp2015_fanbo_1009.pdf