Recurrent Neural Networks (RNNs)

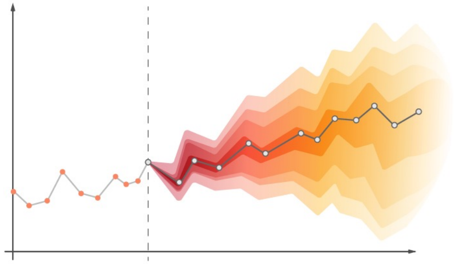
Michel RIVEILL
michel.riveill@univ-cotedazur.fr

- [illegible], [illegible]...

Order of words is important :

- E[illegible]:
  - [illegible]:
    - I [illegible] ≠ I [illegible]
  - [illegible]?

  - [illegible]?



Motivation

- Humans don’t start their thinking from scratch every second
  - Thoughts have persistence
- Traditional neural networks can’t characterize this phenomena
  - Ex: classify what is happening at every point in a movie
  - How a neural network can inform later events about the previous ones
- Recurrent neural networks address this issue
  - Some applications
    - [illegible]: [illegible].
      - □ [illegible]E[illegible]- A[illegible]CE[illegible] [illegible] C[illegible]
      - □ [illegible]- [illegible] ([illegible])
    - F[illegible]- Time-series Prediction
- How?
  - Add state to artificial neurons

What are RNNs?

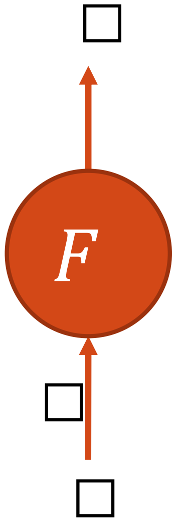
- Main idea is to make use of sequential information
- How RNN is different from neural network?
  - Vanilla neural networks (MLP) assume all inputs are independent of each other
    - Features independence
  - But for many tasks, that’s a very bad idea
- What RNN does?
  - Perform the same task for every element of a sequence
    - That’s what recurrent stands for
  - Output depends on the previous computations!
- Another way of interpretation – RNNs have a “memory”
  - To store previous computations

Some applications (not recent)

- [illegible] G [illegible] ED [illegible]
  - [illegible] :// [illegible] . [illegible] /- [illegible] H [illegible] ? = 31 [illegible]
- [illegible] G [illegible] E [illegible]
  - [illegible] :// [illegible] . [illegible] / [illegible] / [illegible] - [illegible] - [illegible]
- [illegible] G [illegible]
  - [illegible] :// [illegible] . [illegible] . [illegible] /2015/08/03/ [illegible] - [illegible] - [illegible] - [illegible] - [illegible] - [illegible] /

From vanilla NN to recurrent NN

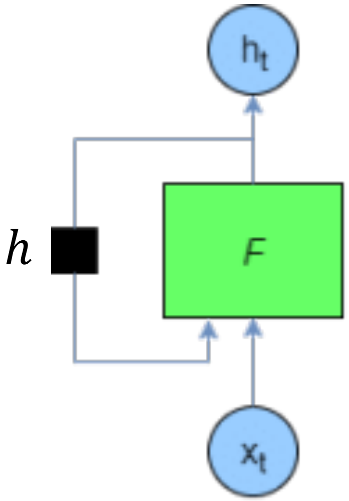
- □□□□□□□□□□□□□□□□□
  - $y = F(U.X)$



One i/p
1 weight

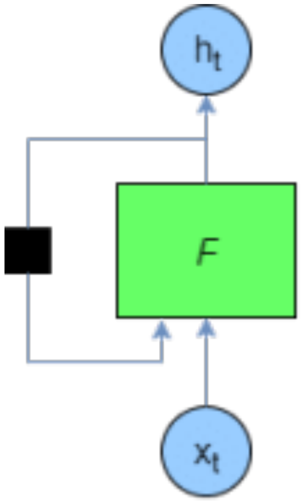
From vanilla NN to recurrent NN

- [illegible]
  - $y = F(U.X)$
- [illegible] → [illegible]2 [illegible]
  - A[illegible]; h
  - [illegible];
    - $h_t = F(W.h_{t-1} + U.X_t)$ ✓✓
    - C[illegible] $h_t = F(V.[h_{t-1}, X_t])$



From vanilla NN to recurrent NN

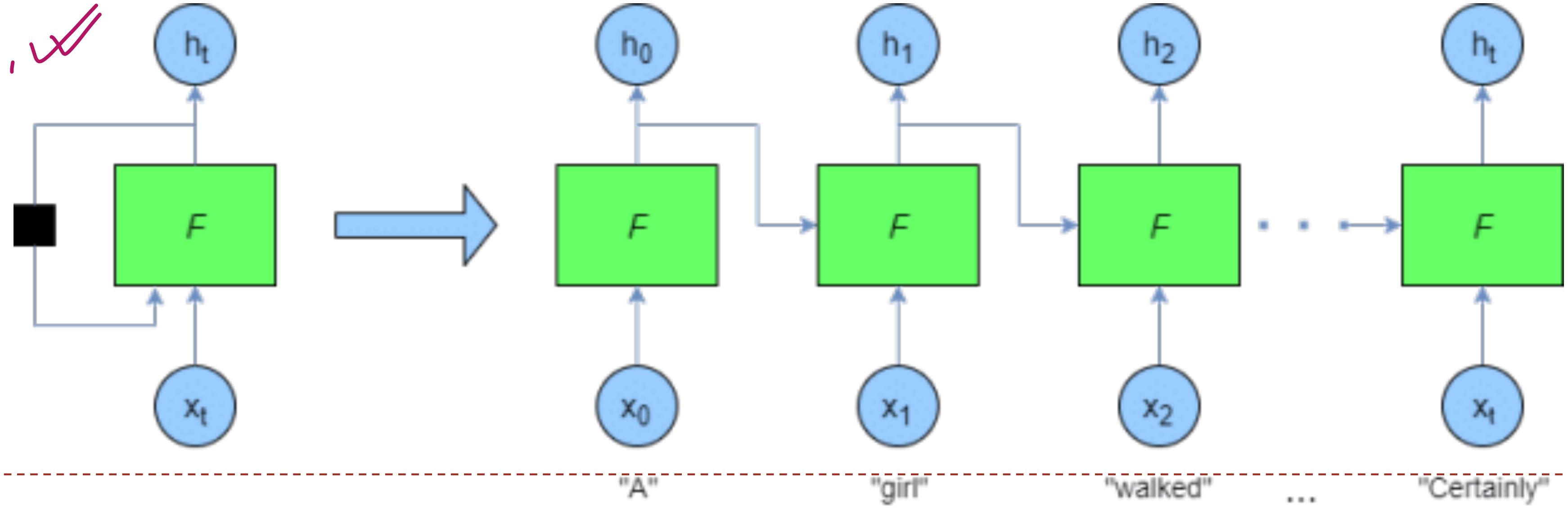
- [illegible]
  - $y = F(U.X)$
- [illegible] → [illegible] [illegible]
  - $h_t = F(W.[h_{t \_ 1}, Xt])$
- [illegible], [illegible]
  - [illegible]
    - A [illegible] [illegible]
    - A[illegible] [illegible](h_t) [illegible]:
      - □ [illegible] [illegible]X_t
      - □ [illegible]$h_{t \_ 1}$



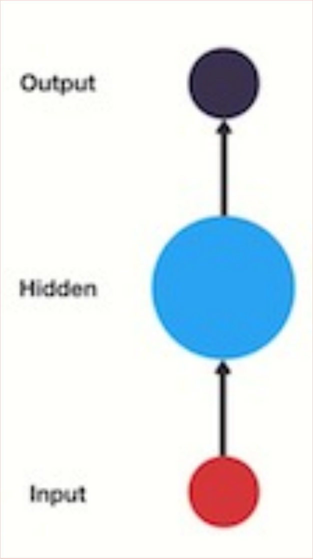
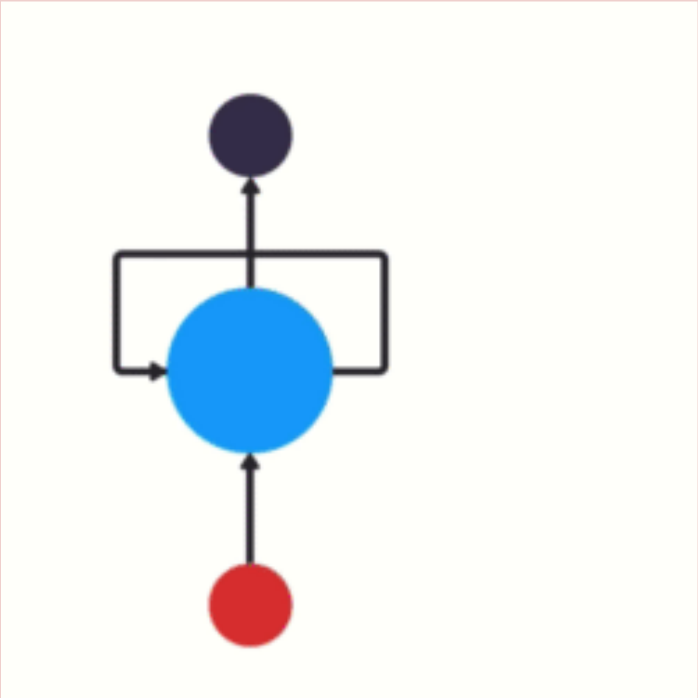
  - $h_1 = F(W.[h_0, X_1])$
  - $h_2 = F(W.h_1, X_2])$
  - $h_3 = F(W.h_2, X_3])$
  - …

From vanilla NN to recurrent NN

- [illegible]
  - $y = F(U.X)$
- [illegible]
  - $h_t \underline{= F(W.[h_{t\_1}, X_t])}$
- [illegible]
  - A [illegible]

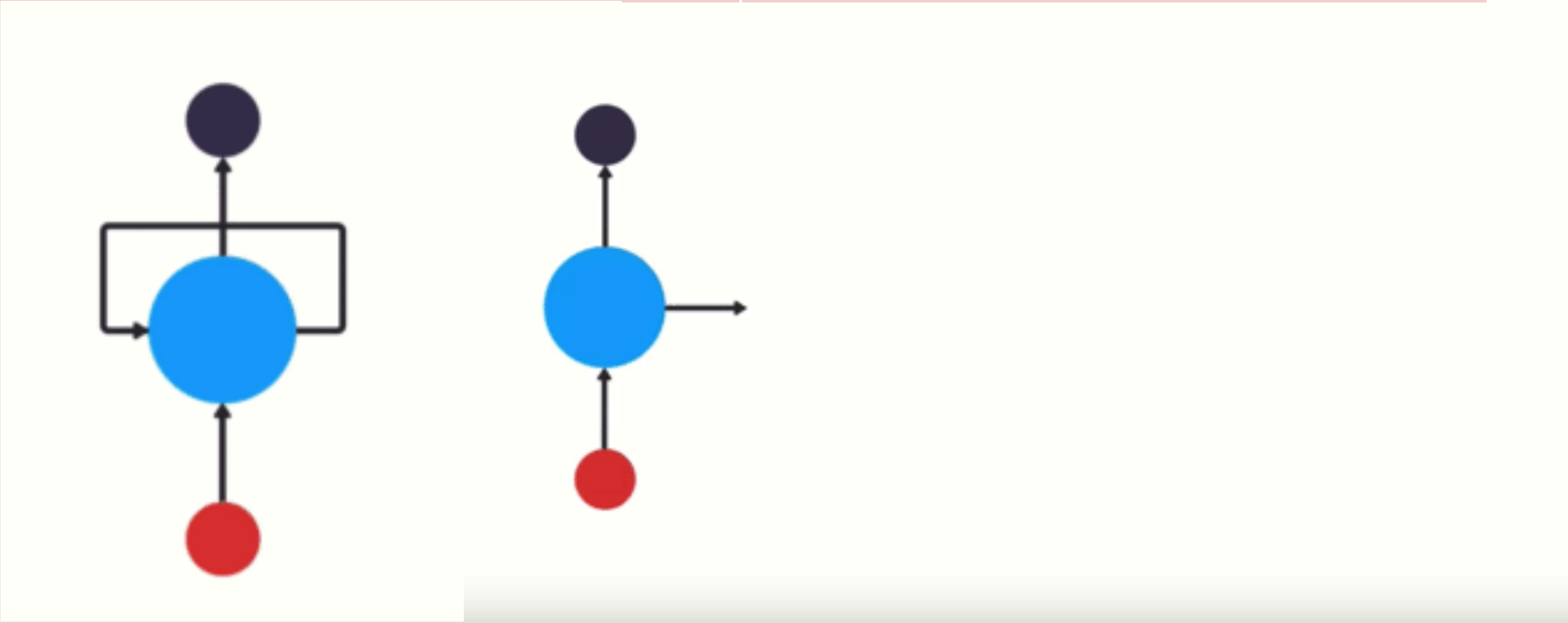


Remember

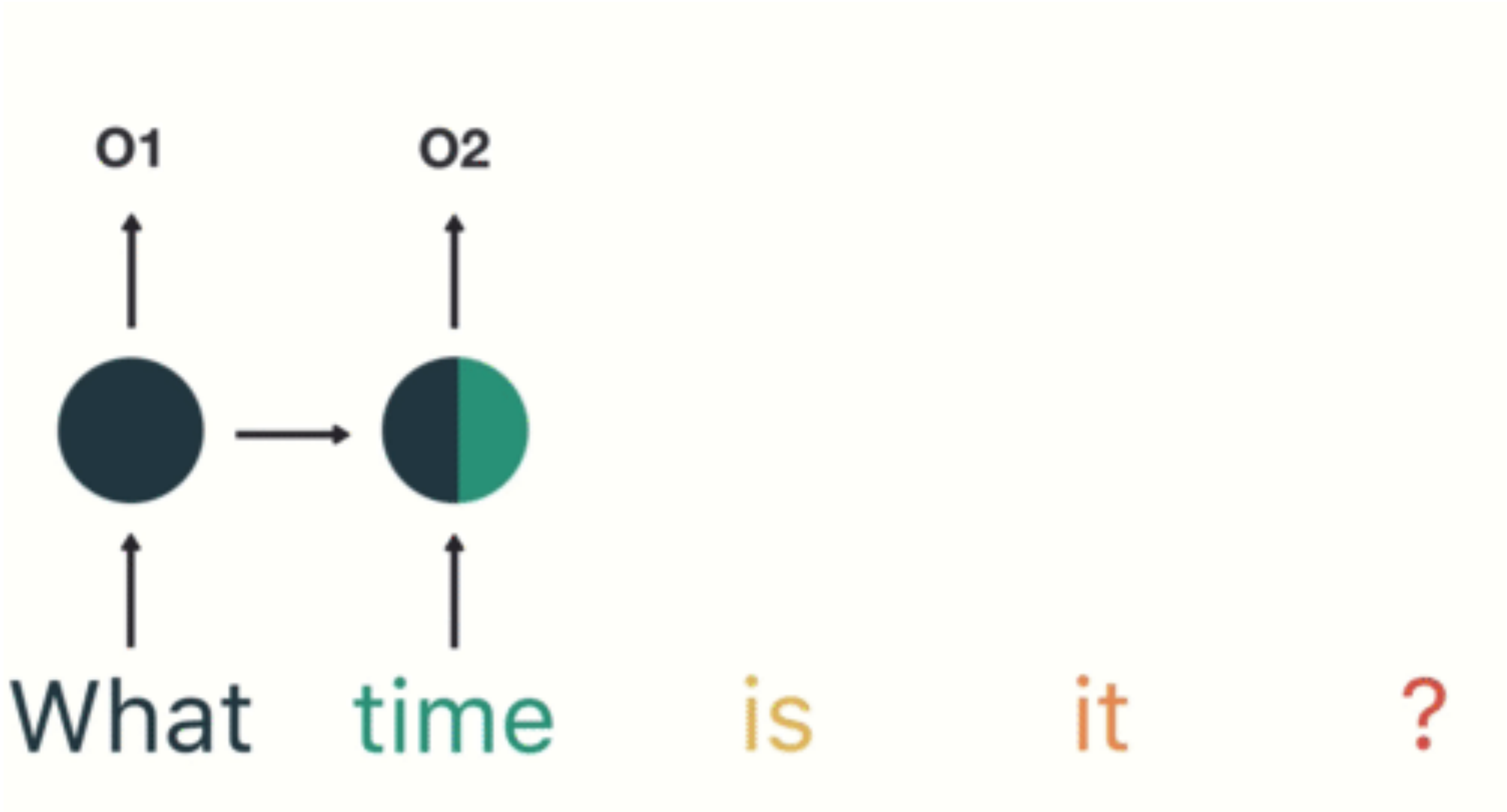
F□□□□	□□
	

Remember

F□□□□□	□□□

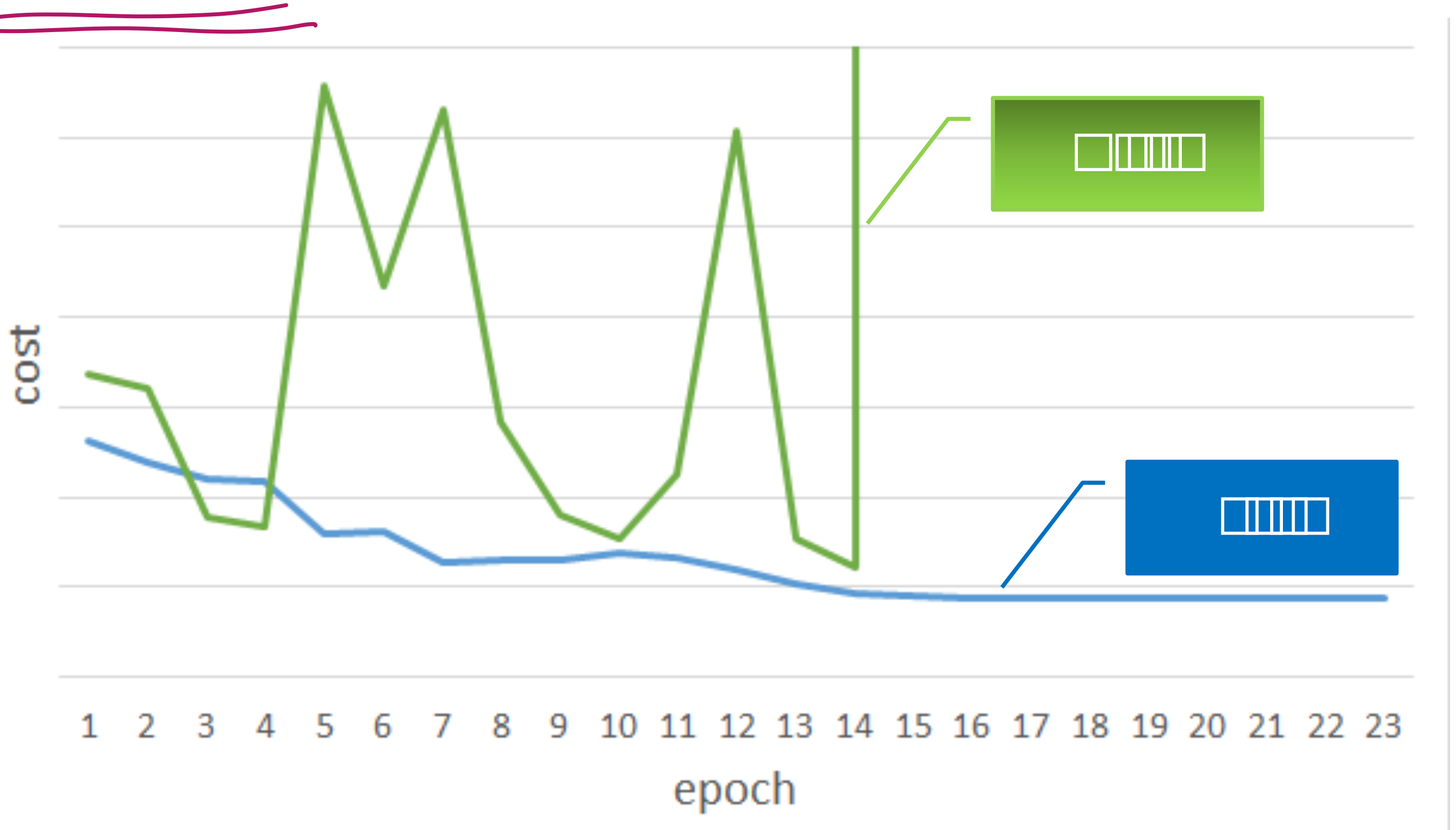


RNN in action



Problems with naive RNN

- [illegible]
- [illegible] [illegible]!



Learning process

Vanishing gradient problem

https://www.youtube.com/watch?v=8z3DFk4VxRo

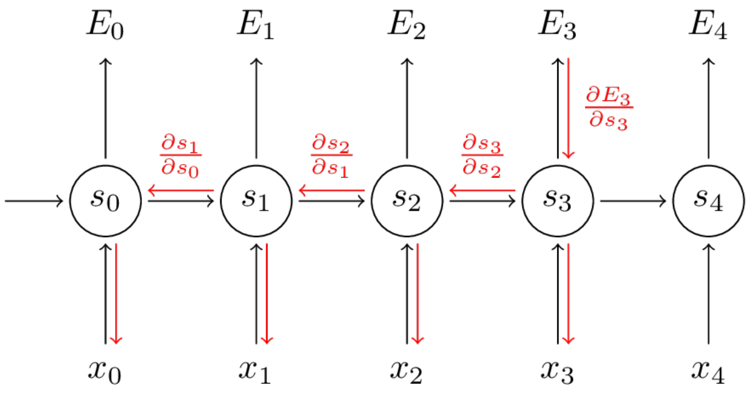
For rigorous proofs and derivations, please refer to

On the difficulty of training recurrent neural networks, Pascanu *et al.*, 2013

Long Short-Term Memory, Hochreiter *et al.*, 1997

Main learning problem

- [illegible]
  - [illegible]
    - $W_{new} = W_{old} - \lambda\, gradient$
  - [illegible], [illegible] ([illegible])



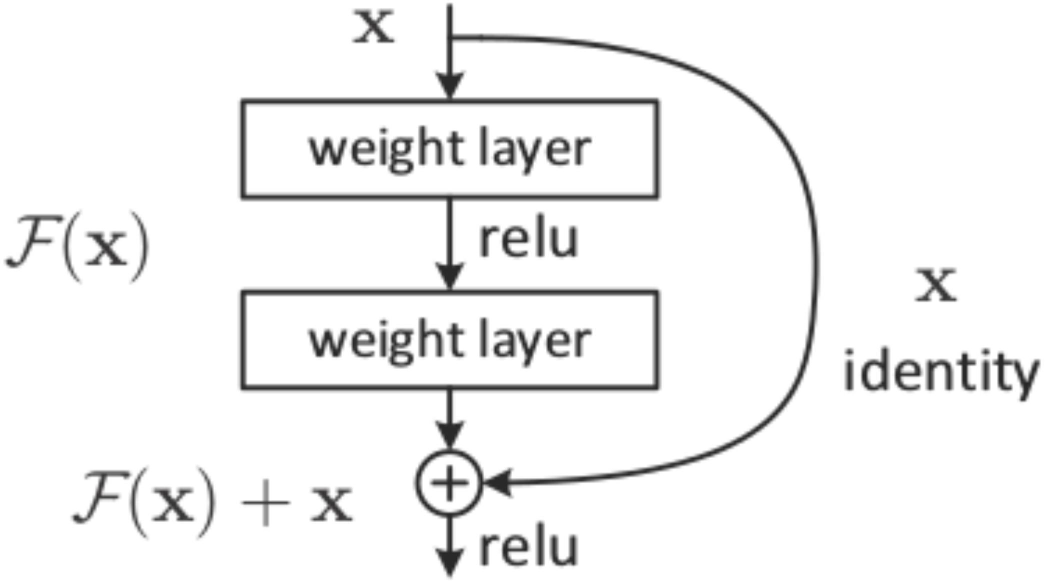
E_0
E_1
E_2
E_3
E_4
$\frac{\partial E_3}{\partial s_3}$
$\frac{\partial s_1}{\partial s_0}$
$\frac{\partial s_2}{\partial s_1}$
$\frac{\partial s_3}{\partial s_2}$
s_0
s_1
s_2
s_3
s_4
x_0
x_1
x_2
x_3
x_4

- $\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \widehat{y_3}} \frac{\partial \widehat{y_3}}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W}$
- $\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial \widehat{y_3}} \frac{\partial \widehat{y_3}}{\partial s_3} \frac{\partial s_3}{\partial W} + \frac{\partial E_3}{\partial \widehat{y_3}} \frac{\partial \widehat{y_3}}{\partial s_3} \frac{\partial s_3}{\partial s_2} \frac{\partial s_2}{\partial W} + \square + \frac{\partial E_3}{\partial \widehat{y_3}} \frac{\partial \widehat{y_3}}{\partial s_3} \frac{\partial s_3}{\partial s_2} \frac{\partial s_2}{\partial s_1} \frac{\partial s_1}{\partial s_0} \frac{\partial s_0}{\partial W}$

What is the value of the derivative 'chain’? $\frac{\partial s_3}{\partial s_2}\frac{\partial s_2}{\partial s_1}\frac{\partial s_1}{\partial s_0}$

- F[illegible]:
  - [illegible] = 0.25
  - $0.25^2 = 0.0625$
  - $0.25^4 = 0.00391$
  - $0.25^8 = 0.0000152$
  - $0.25^{16} = 0.000000000233$
- **[illegible], [illegible] 50**
  - **G[illegible]: $7 * 10^{-31}$**
- **[illegible], [illegible]?**
  - ✓ F[illegible]: [illegible] → [illegible], [illegible]
  - ✓ **F[illegible]: [illegible]** ✓

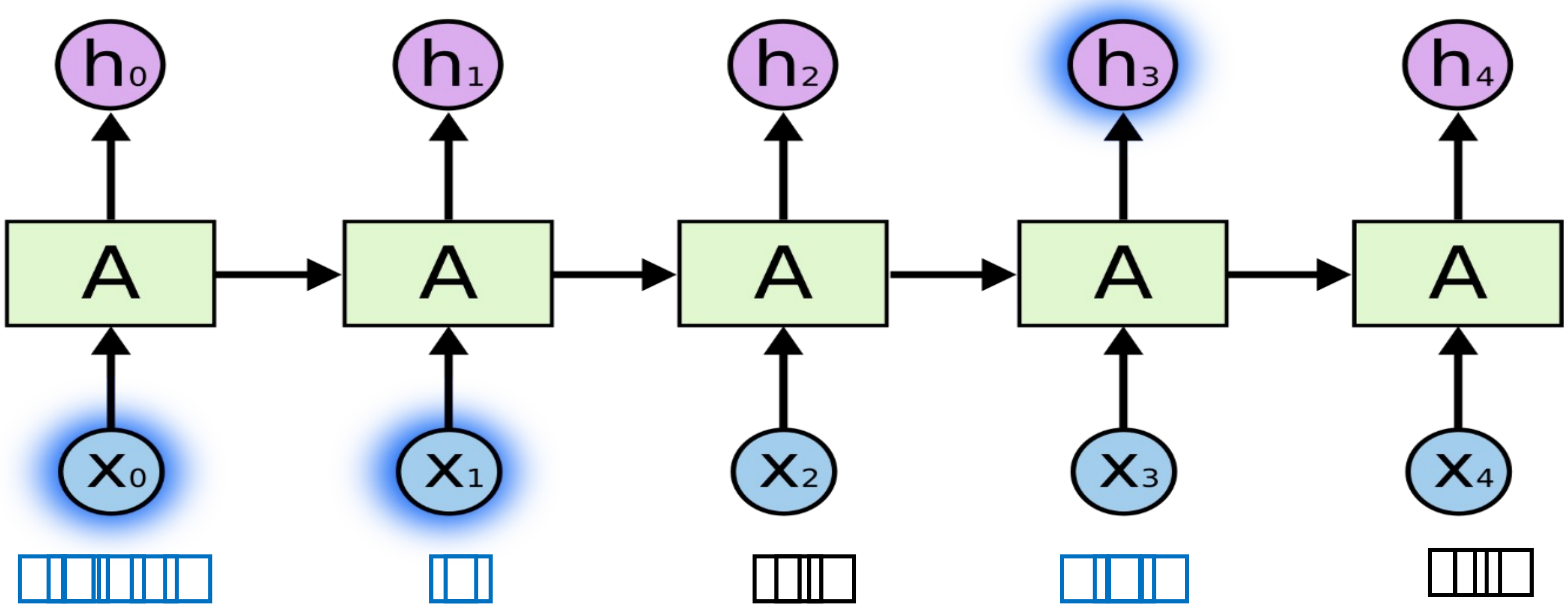
Residual ?


$\mathbf{x}$
weight layer
$\mathcal{F}(\mathbf{x})$
relu
$\mathbf{x}$
weight layer
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

How to introduce residual in RNN

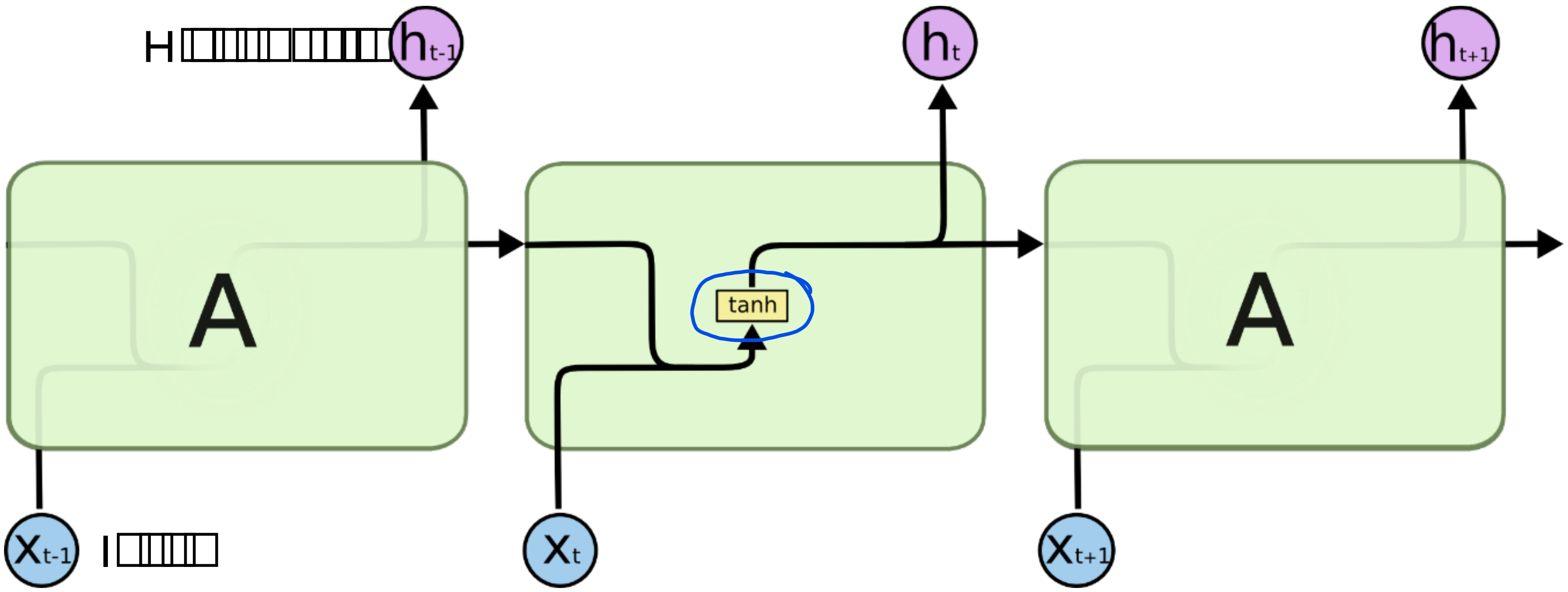
From vanilla Sort Term Memory…

- [illegible]
  - F[illegible]
  - [illegible]
    - F[illegible]:
      - [illegible]



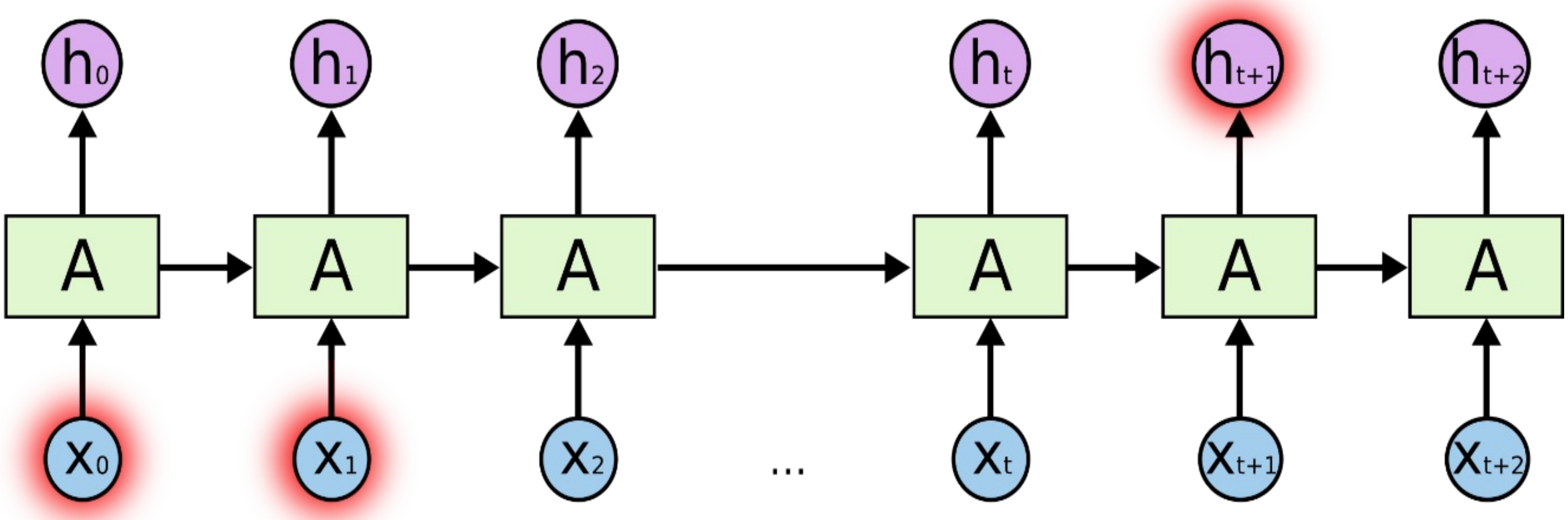
From vanilla Sort Term Memory…

- [illegible]
  - $\square_t = tan\square(W\square\square_{t-1}, xt\ \square)$
  - I[illegible]
  - [illegible]
  - C[illegible]



… to LSTM (Long Short Term Memory)

- B[illegible]
  - [illegible]!
  - [illegible]1: [illegible]
  - [illegible]2: [illegible]
    [illegible]: [illegible], [illegible], [illegible]



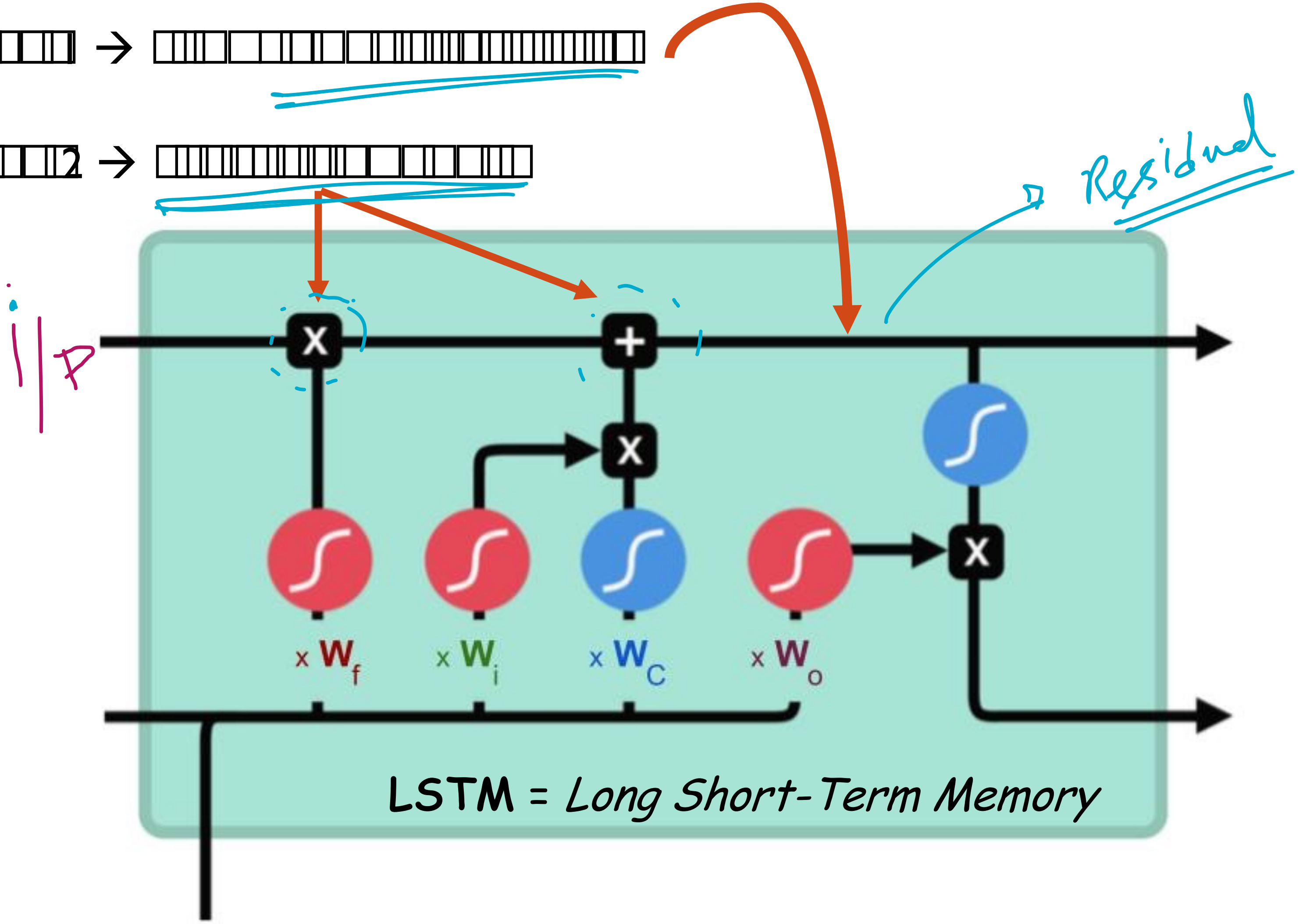
I [illegible] F [illegible]…

I [illegible]F[illegible]…

Dealing with the vanishing gradient problem → LSTM cell

[illegible] → [illegible]

[illegible]2 → [illegible]

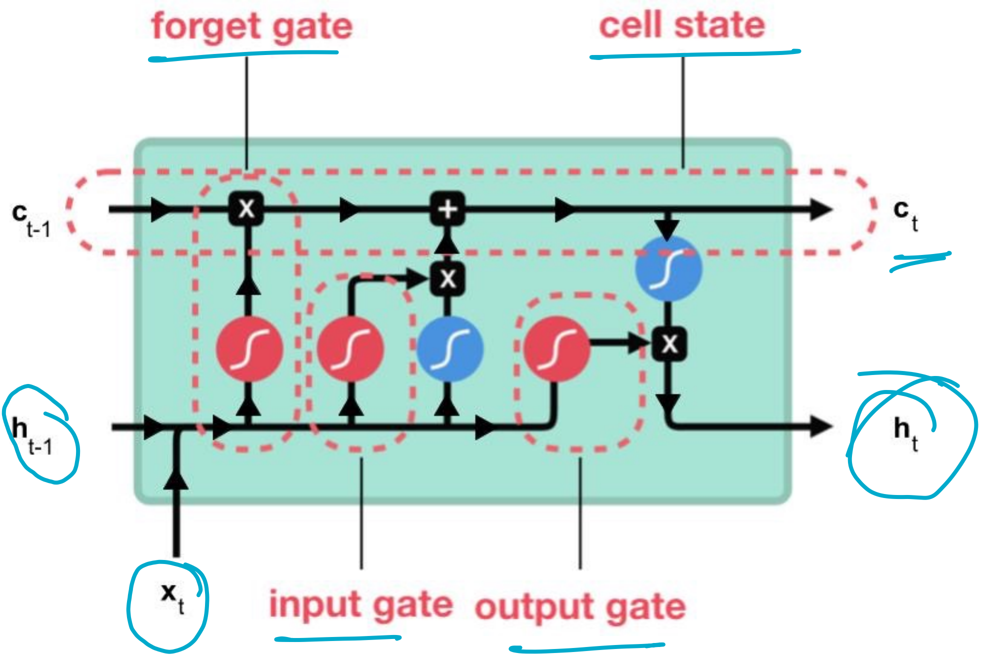


LSTM = *Long Short-Term Memory*

(crédit : image modifiée de Michaël Nguyen)

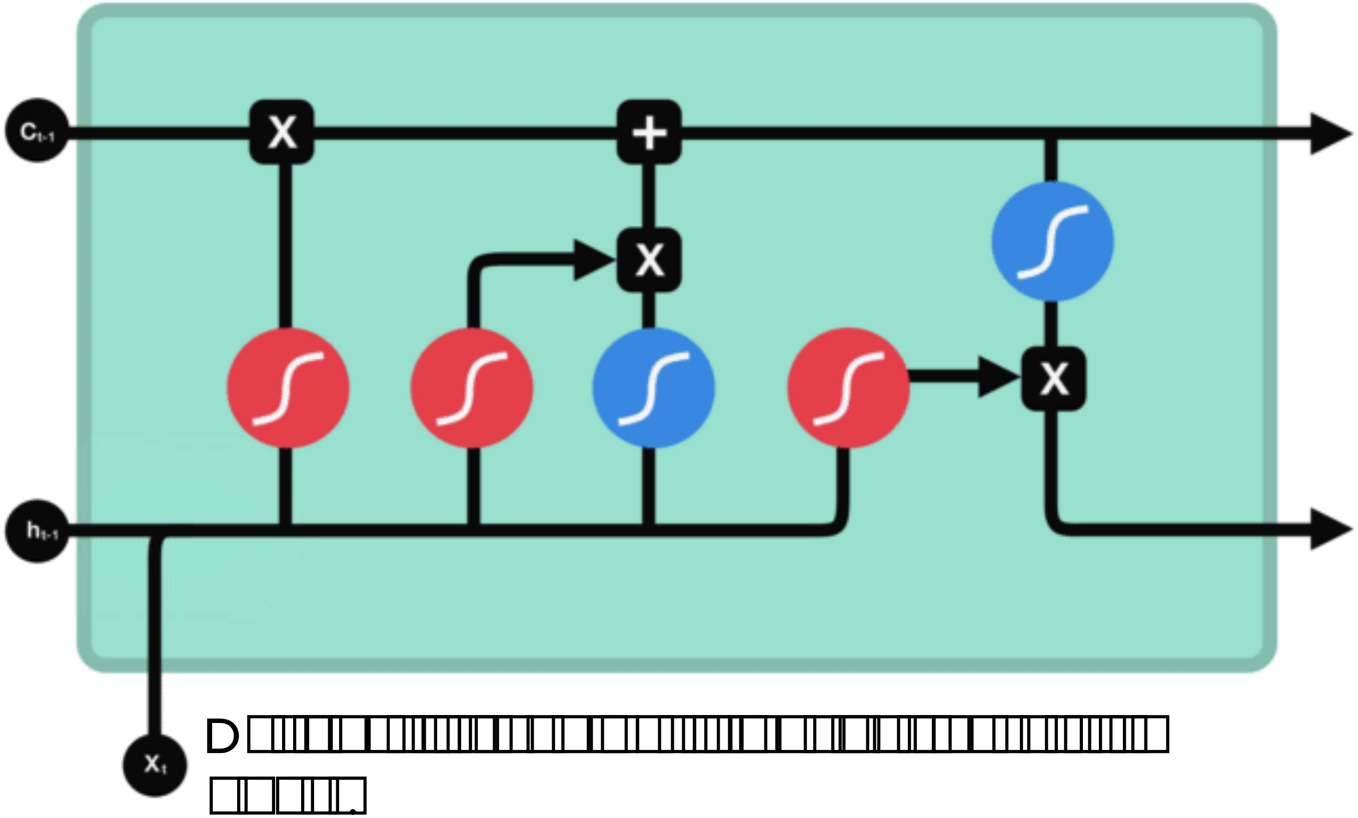
LSTM cell

- C[illegible]'[illegible]":
  - C[illegible] [illegible]([illegible] [illegible])
  - F[illegible]([illegible])
  - I[illegible]([illegible])
  - [illegible]([illegible])
- H[illegible]([illegible])
  - [illegible]
- C[illegible]([illegible]
  - [illegible]



(crédit : image modifiée de Michaël Nguyen)

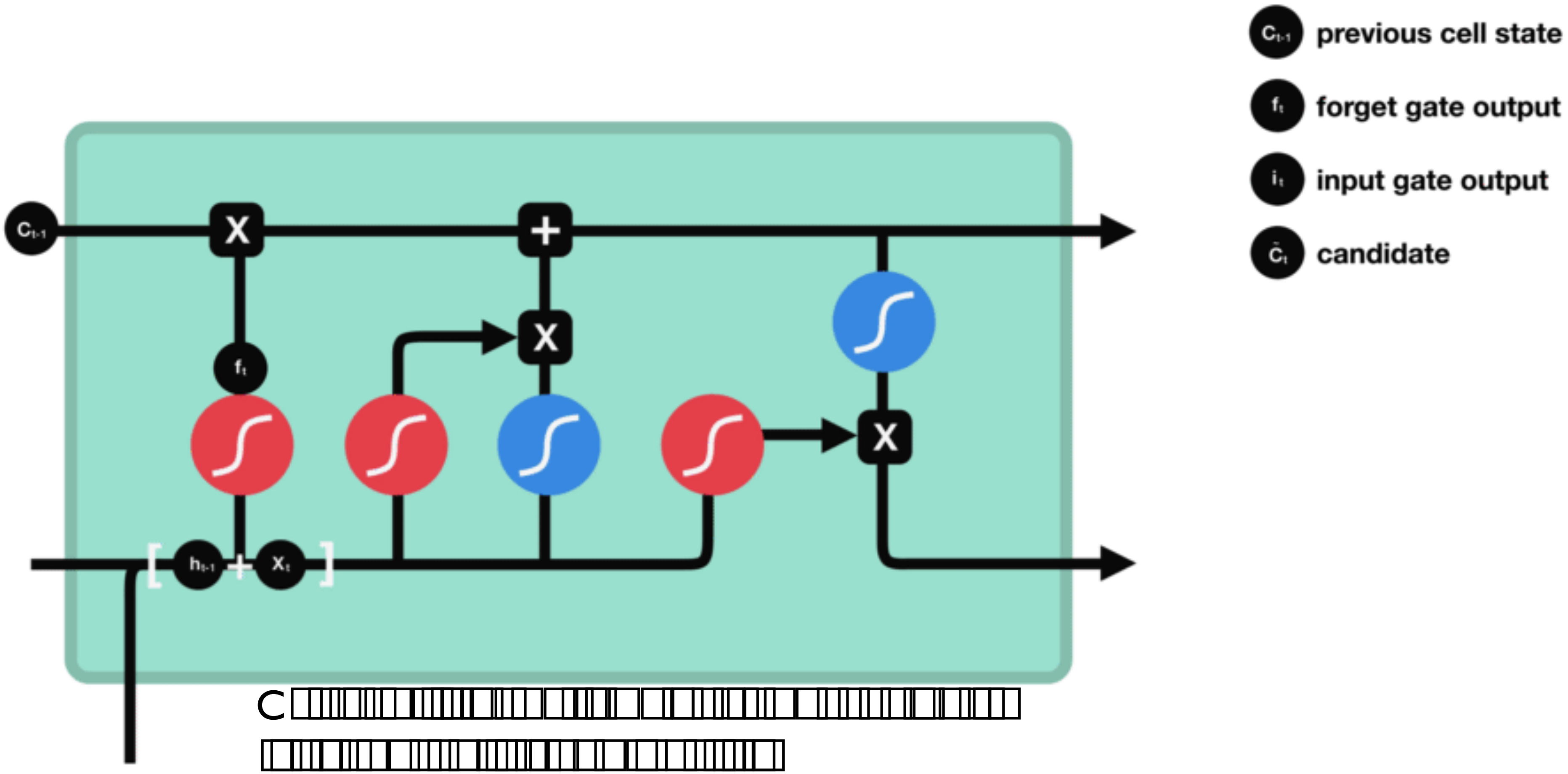
LSTM cell (porte oubli / forget get)



D[illegible]

D[illegible]

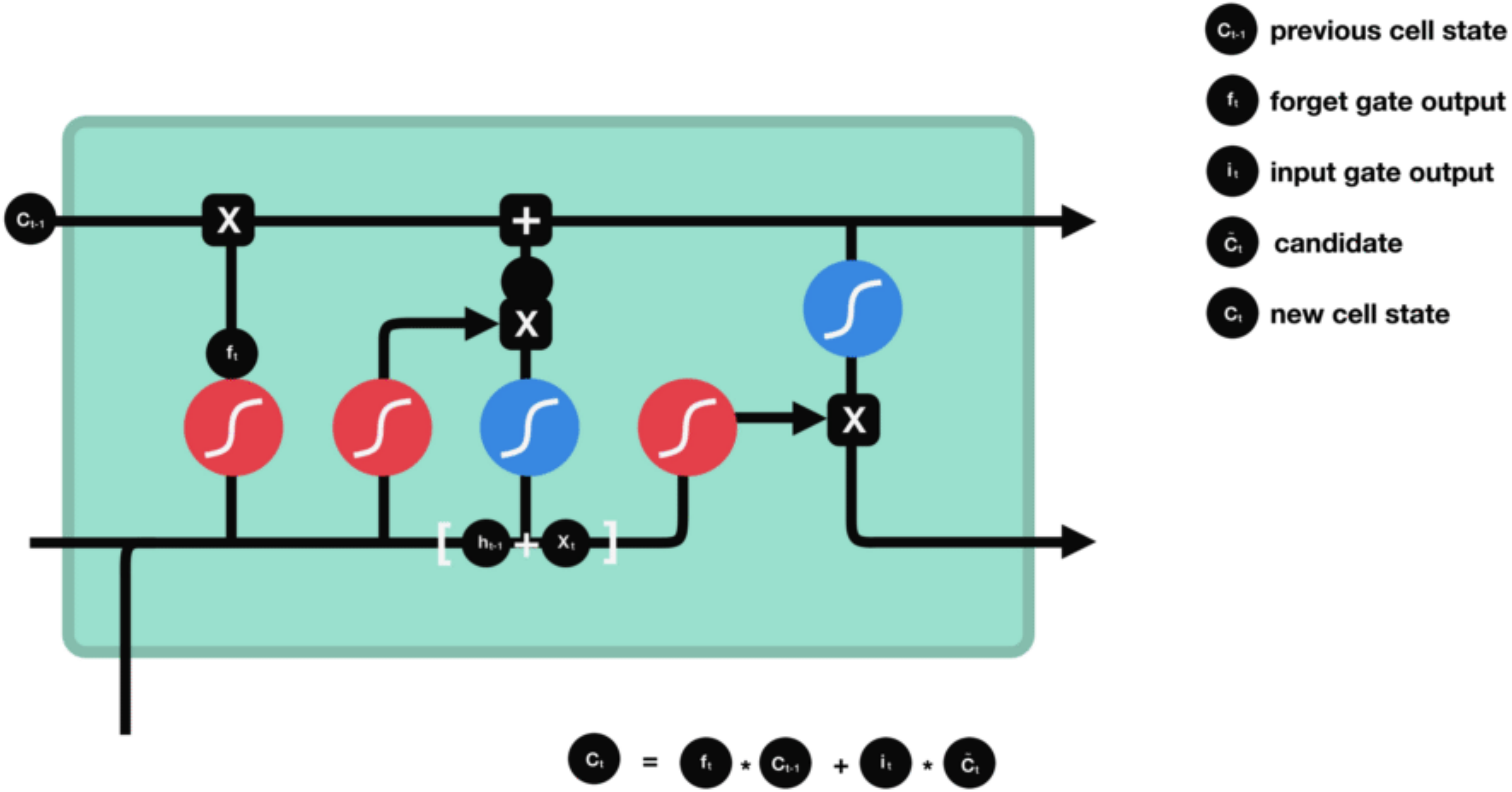
LSTM cell (porte entrée / input get)



C[illegible]

C[illegible]

LSTM cell (état de la cellule / cell state)

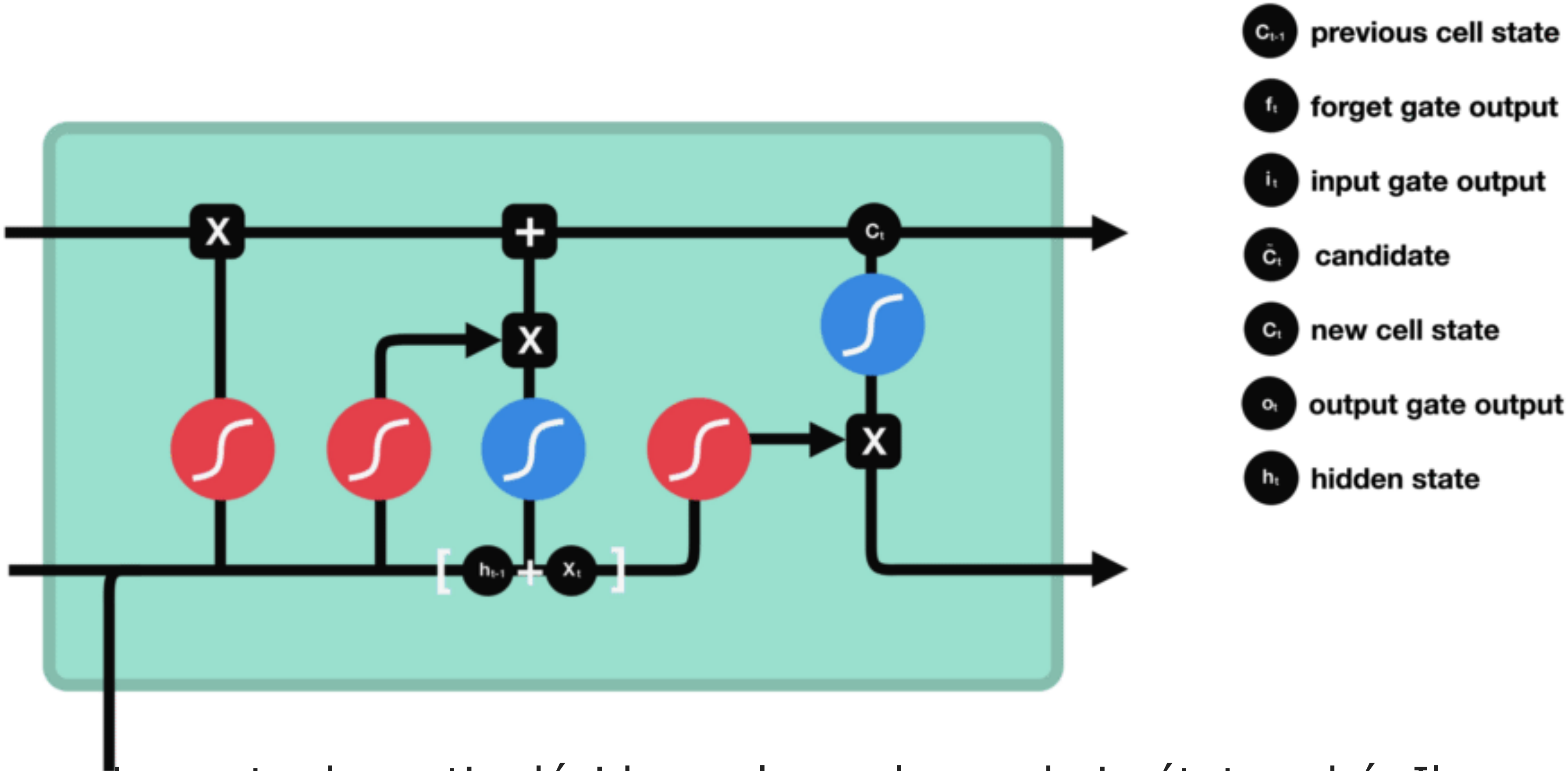


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

[illegible]

[illegible]

LSTM cell (porte de sortie / output gate)

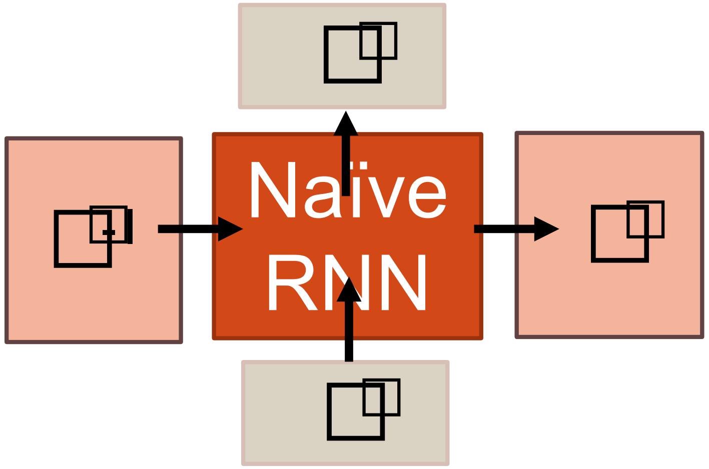


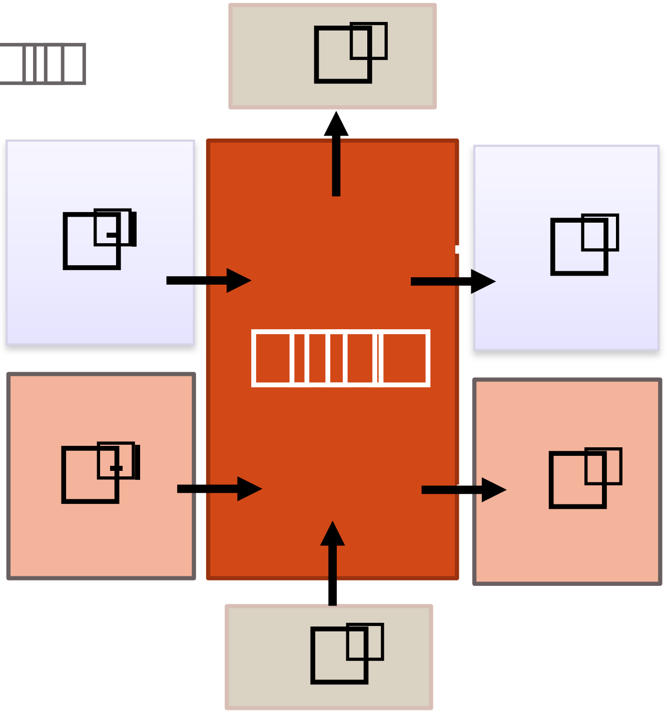
La porte de sortie décide quel sera le prochain état caché. Il contient des informations sur les entrées précédentes du réseau et sert aux prédictions.

The output gate decides what the next hidden state will be. It contains information about previous inputs to the network and is used for predictions.

Naïve RNN vs LSTM

- [illegible]
  - [illegible]
- [illegible]
  - A[illegible]B [illegible] [illegible],C[illegible] [illegible]



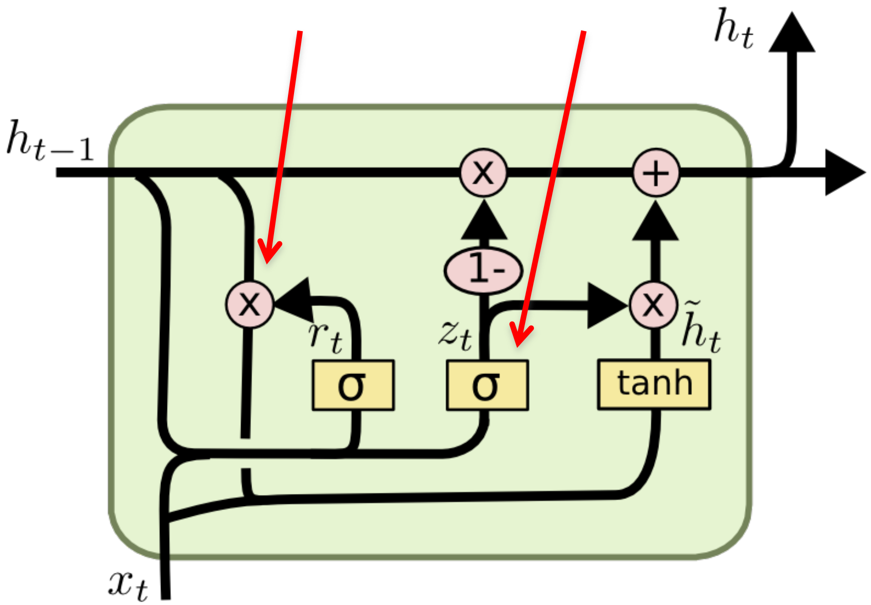


c changes slowly → c^t is c^{t-1} added by something

h changes faster → h^t and h^{t-1} can be very different

GRU – gated recurrent unit

- G□□= □□□□□□□□□□□□ C□□□



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

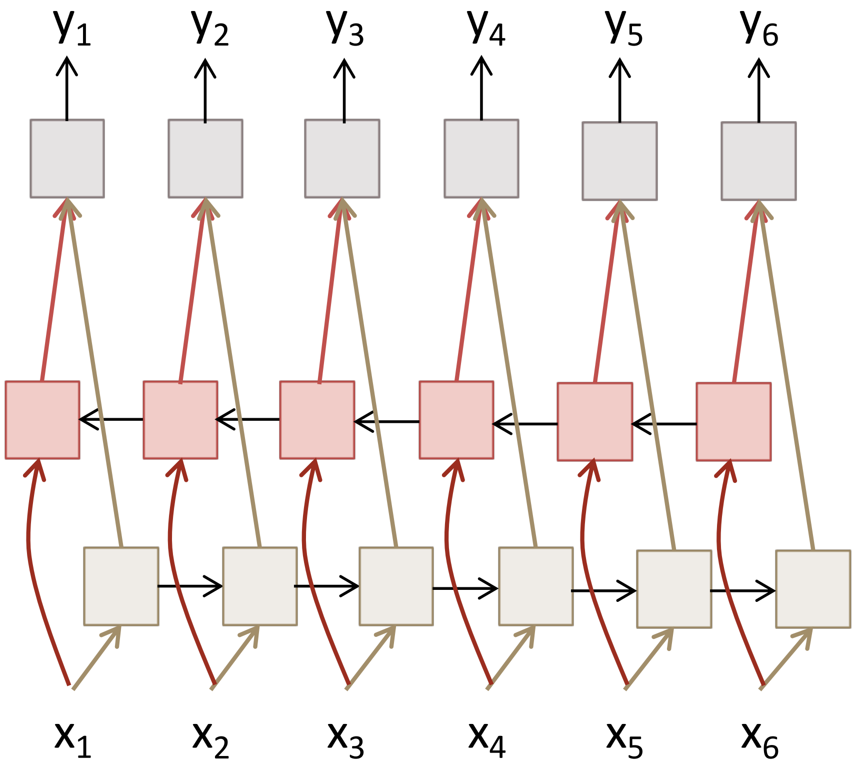
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

- It combines the forget and input into a single update gate.
- It also merges the cell state and hidden state.

→ This is simpler/faster than LSTM.

Bi-directional RNNs

- RNNs can process the input sequence in forward and in the reverse direction

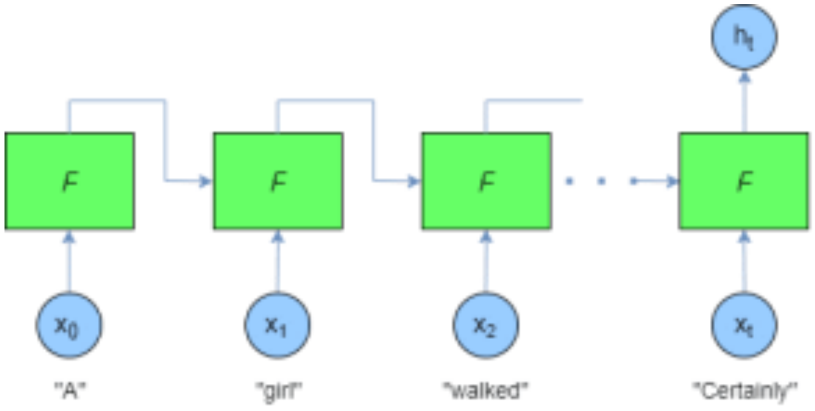


- Popular in speech recognition, could be used also with text

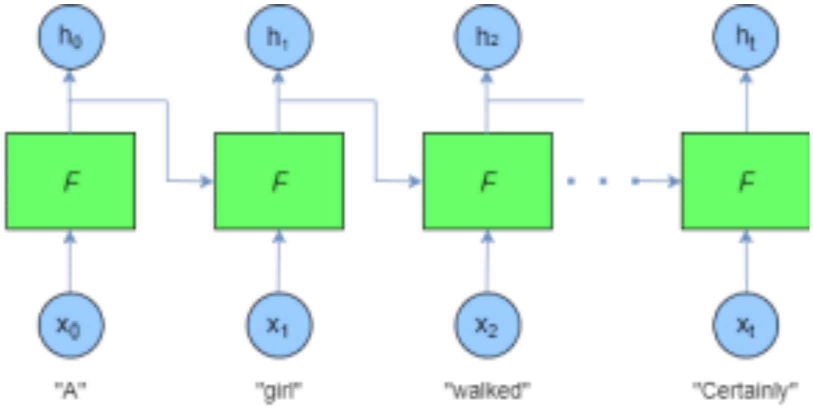
RNN cell in Keras

Keras Long Short-Term Memory Cell

- [illegible]
- [illegible]
  - ✓ [illegible]: [illegible]
  - ✓ [illegible]: [illegible] ✱✱
    - [illegible]
    - [illegible]
      - □ [illegible] = [illegible] ([illegible])
  - ✓ [illegible]: [illegible] ✱✱
    - [illegible] 3 [illegible]
      - □ [illegible] ([illegible])
      - □ [illegible]
      - □ [illegible]
    - [illegible]



return_sequences = False



return_sequences = True

If there's a Dense after LSTM. How many dense cells are used?

- □= □□□□(500, □□□□□□□□□□□□□□□□= F□□□□□□□□)
- □□□□□□= D□□□□(□)(□)

No of Neurons

With □□□□□□□□□□□□□□□□□= F□□□□,

D□□□□layer is applied only once at the last cell

With □□□□□□□□□□□□□□□□□= □□□□

D□□□□layer is applied to every timestep

LSTM → 3D data (, ,)

LSTM → 3D data (, ,) ←

A basic example: forcasting

[illegible] = I[illegible]([illegible]=([illegible],1)) # [illegible]

- [illegible]=[illegible], [illegible], 1[illegible] → One Features.

[illegible] = [illegible](16, [illegible]=F[illegible])([illegible])

- [illegible]=[illegible], 16[illegible]

[illegible] = D[illegible](1, [illegible]=[illegible])([illegible])

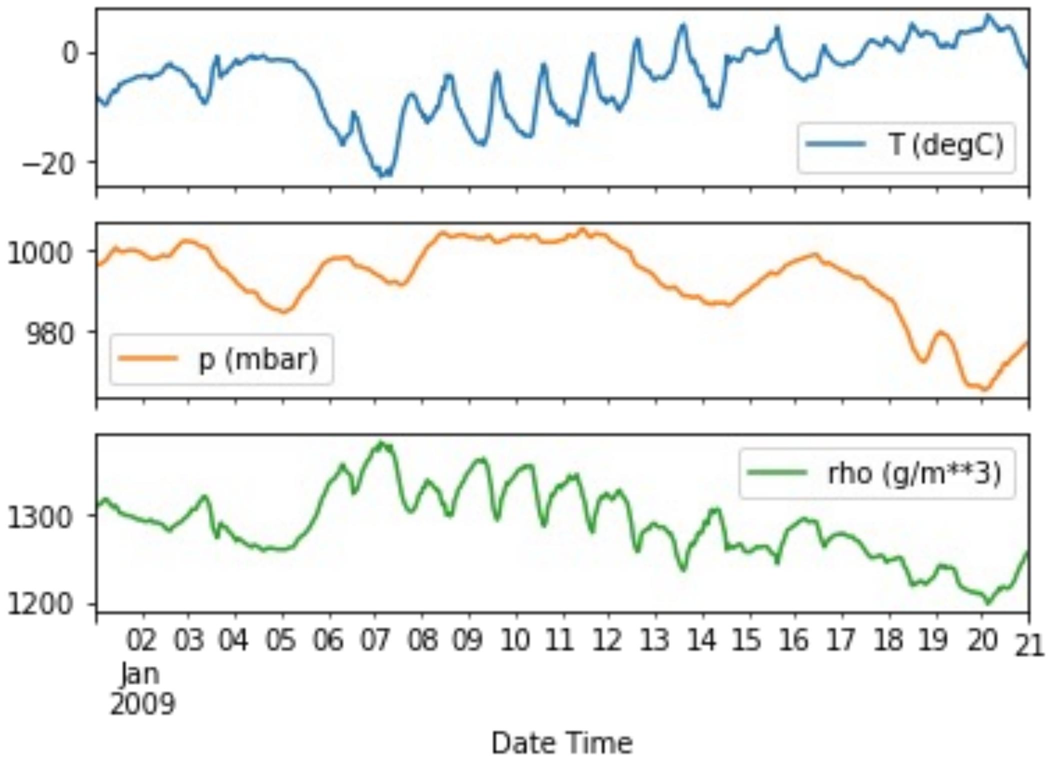
- [illegible]=[illegible], 1[illegible] → Regression.

- F[illegible]
  - [illegible]([illegible], [illegible]). ← [illegible] [illegible]

- F[illegible]
  - F[illegible] [illegible]([illegible]([illegible])): ← [illegible]
    - [illegible]([illegible],[illegible],[illegible])

RNN for forcasting

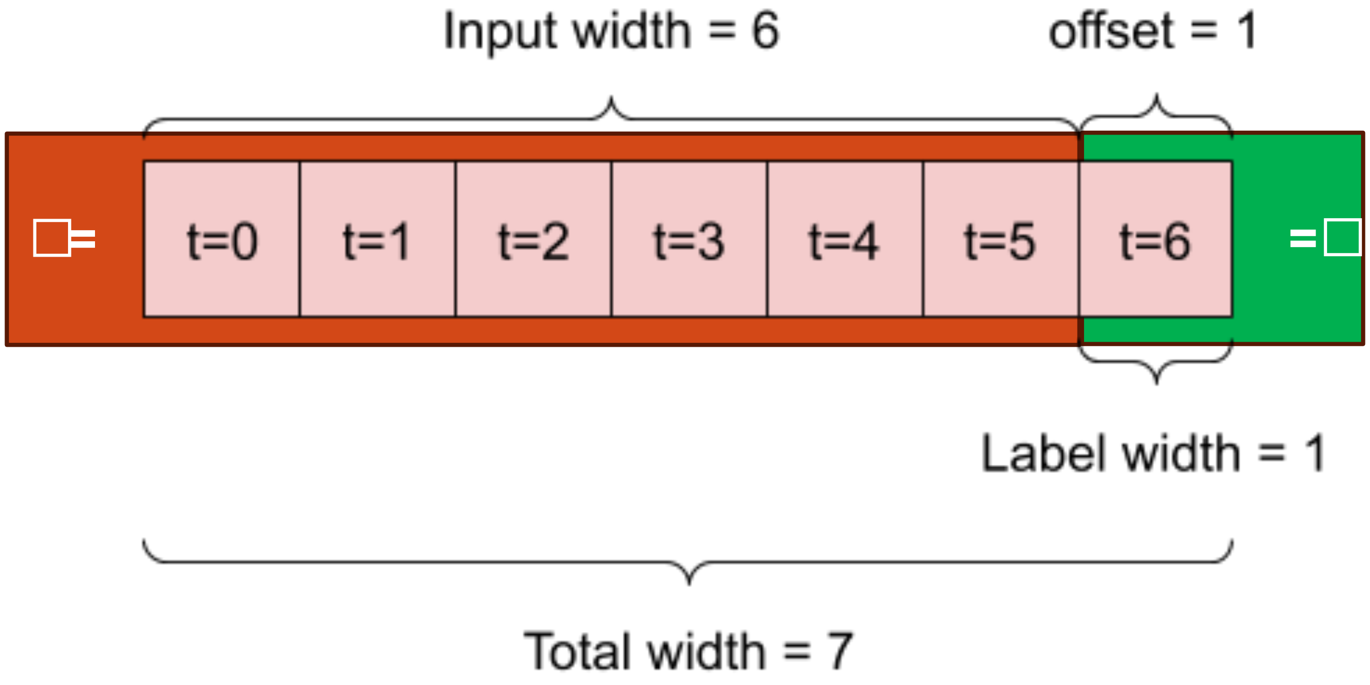
RNN for forecasting



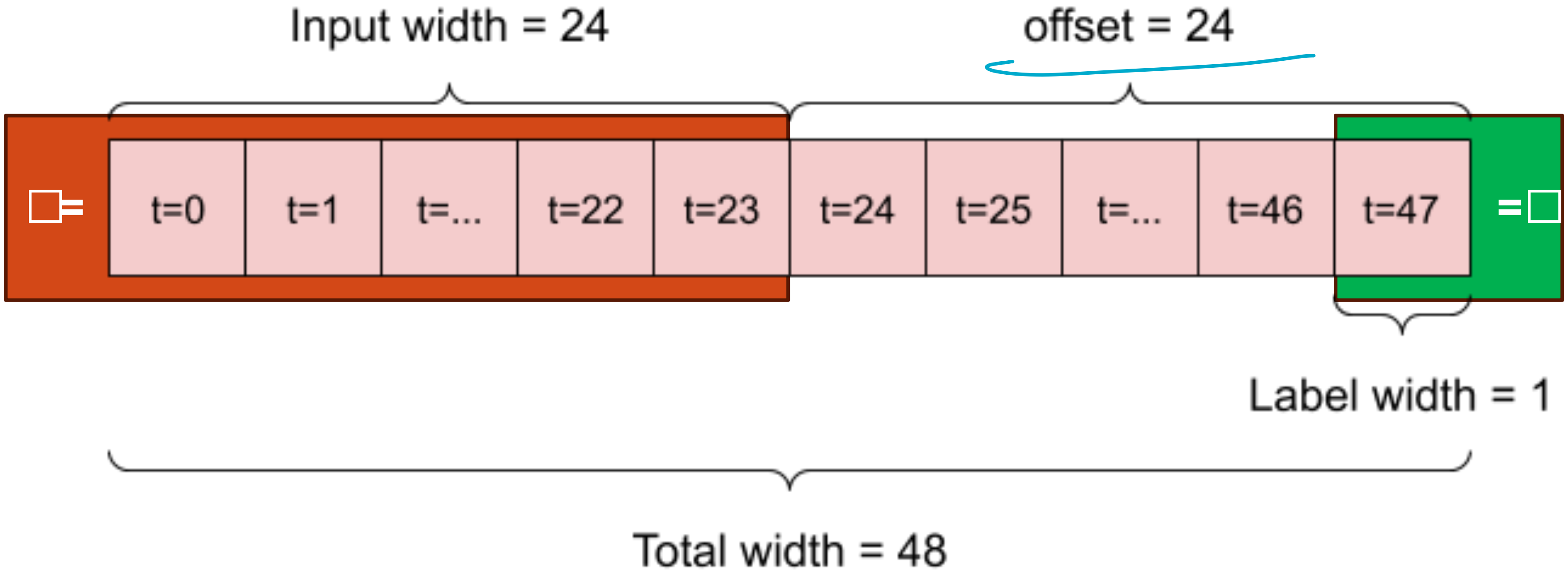
[illegible]

RNN for forecasting

- A [illegible]
  - [illegible]
  - [illegible]

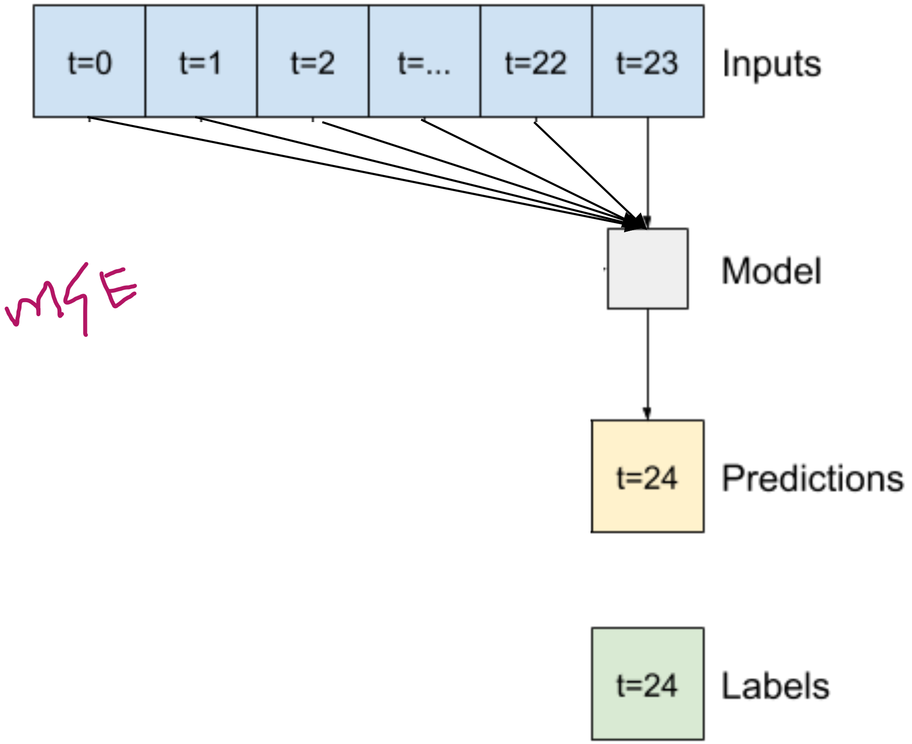


- [illegible]
  - 24 [illegible],
  - [illegible]24 [illegible]



Input width = 24
offset = 24
□=
t=0
t=1
t=...
t=22
t=23
t=24
t=25
t=...
t=46
t=47
=□
Label width = 1
Total width = 48

Dense model

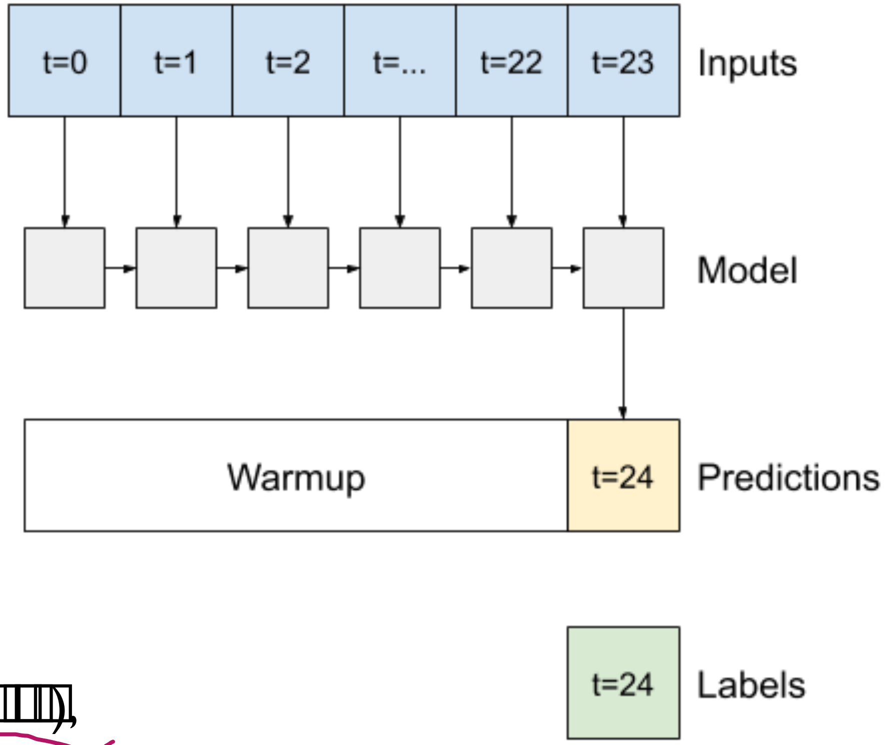


MSE

```
[illegible]= [illegible]([illegible]
    [illegible]F[illegible](),
    # [illegible]:([illegible],[illegible])=> ([illegible]*[illegible])

    [illegible]D[illegible]([illegible]=32, [illegible]='[illegible]'),
    [illegible]D[illegible]([illegible]=1),
[illegible])
```

Recurrent model: return_state=False

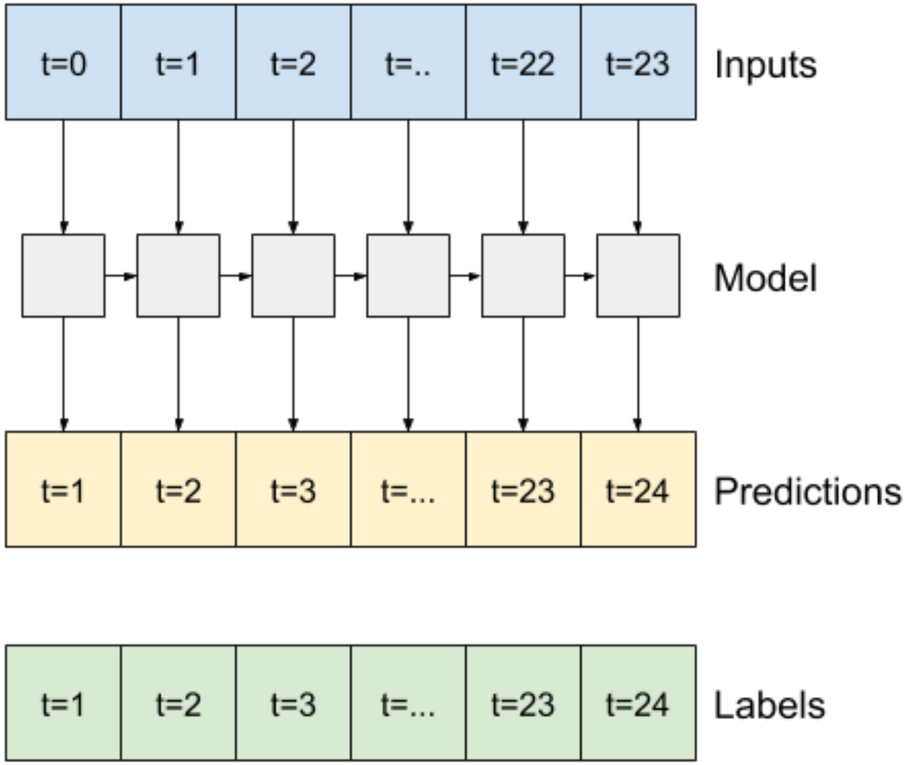


```
[illegible]= [illegible]([illegible]

    [illegible](32, [illegible]=F[illegible]),
   # [illegible], [illegible], [illegible]=> [illegible], [illegible]

    [illegible]D[illegible]([illegible]=1)
[illegible])
```

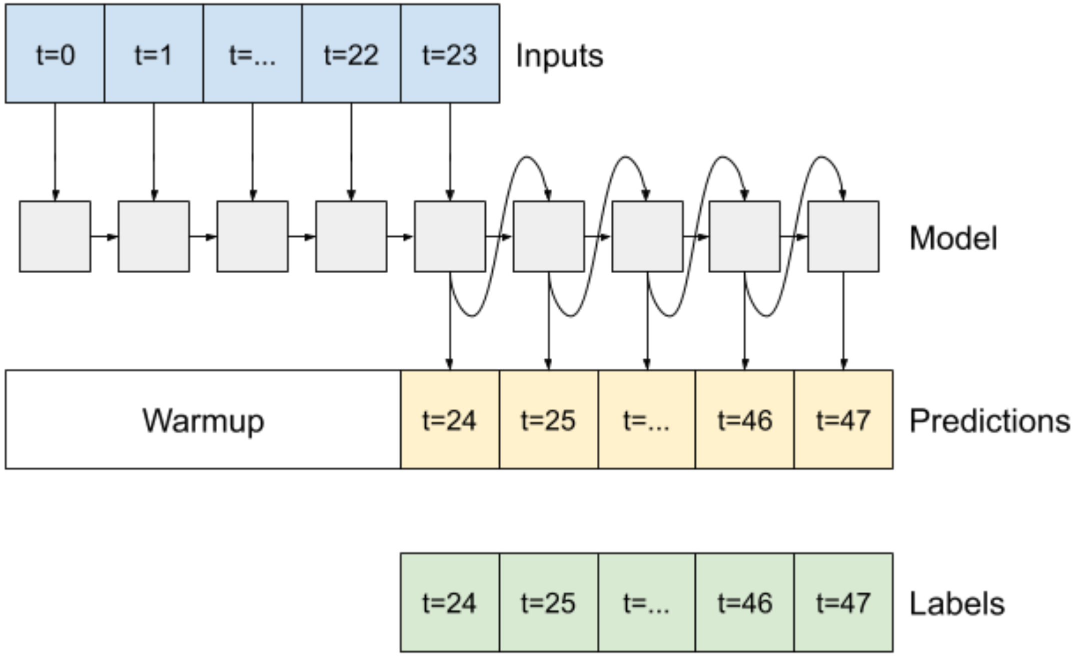
Recurrent model: return_state=True



```
[illegible] = [illegible]([illegible]
    [illegible](32, [illegible]=[illegible]),
   # [illegible], [illegible], [illegible] => [illegible], [illegible], [illegible]

    [illegible]D[illegible]([illegible]=1)
    # [illegible]D[illegible]
)
```

How to predict futur multiple values

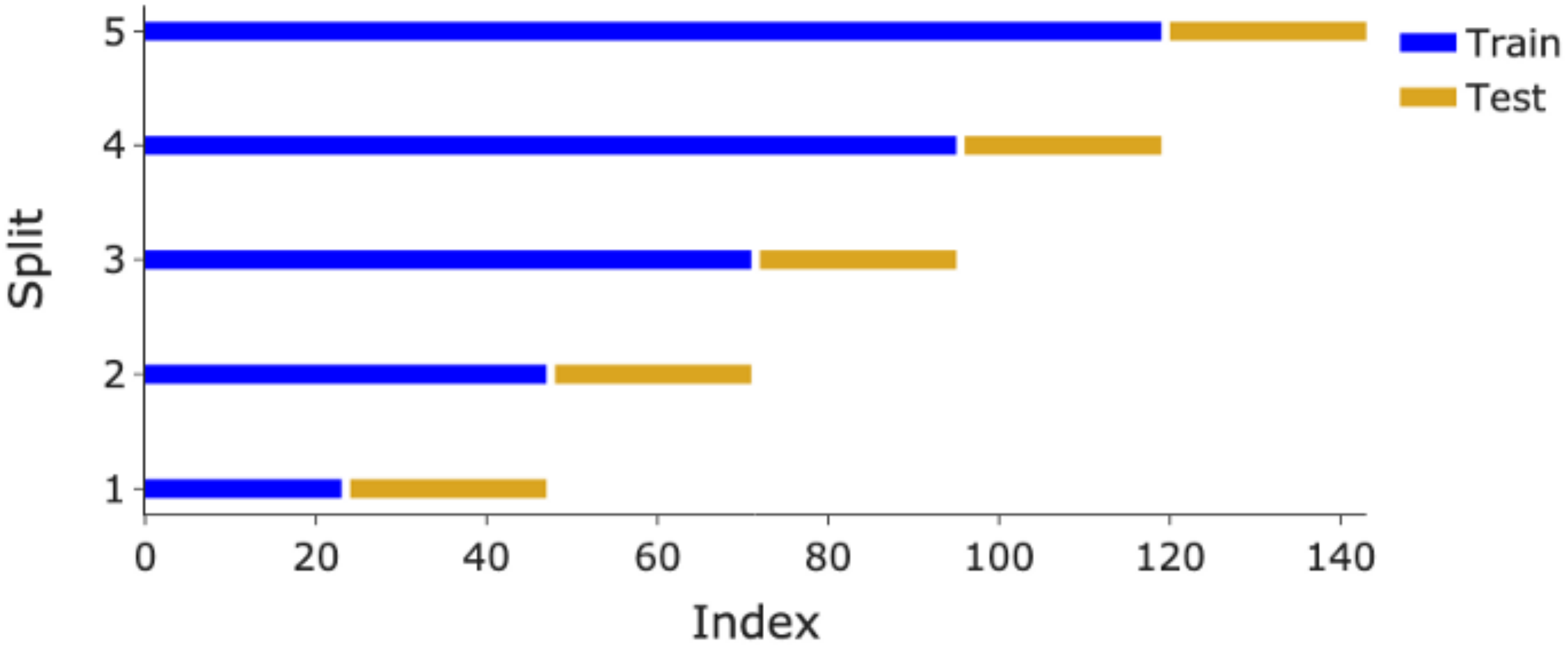


[illegible]

[illegible] 2 [illegible]

Time series cross-validation

```
from sklearn.model_selection import TimeSeriesSplit
```



RNN in NLP task

Some use of RNN
→ Text Classification / Sentiment analysis

A [illegible]

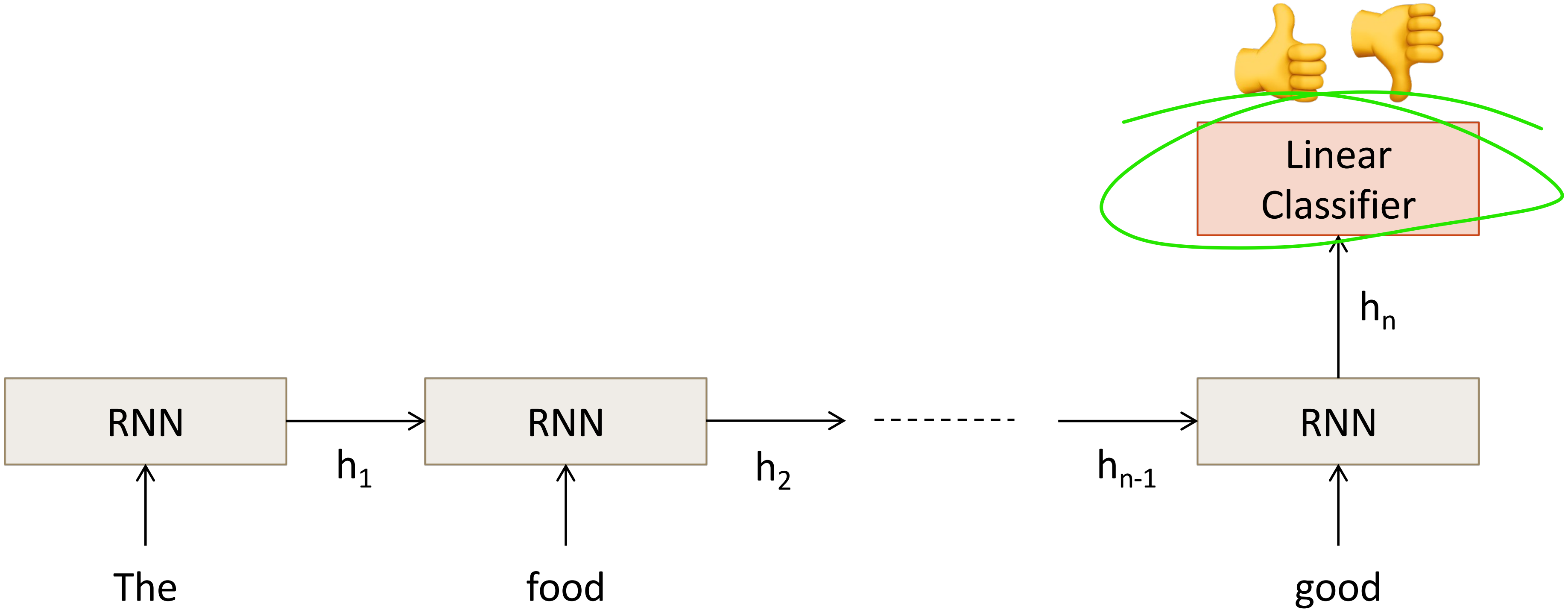
- C[illegible]
  - [illegible]!
  - [illegible] I[illegible]DB

    [illegible]

    [illegible]
- I[illegible]:
  - [illegible], [illegible]
- [illegible]:
  - [illegible]/ [illegible]
- [illegible]
- [illegible]

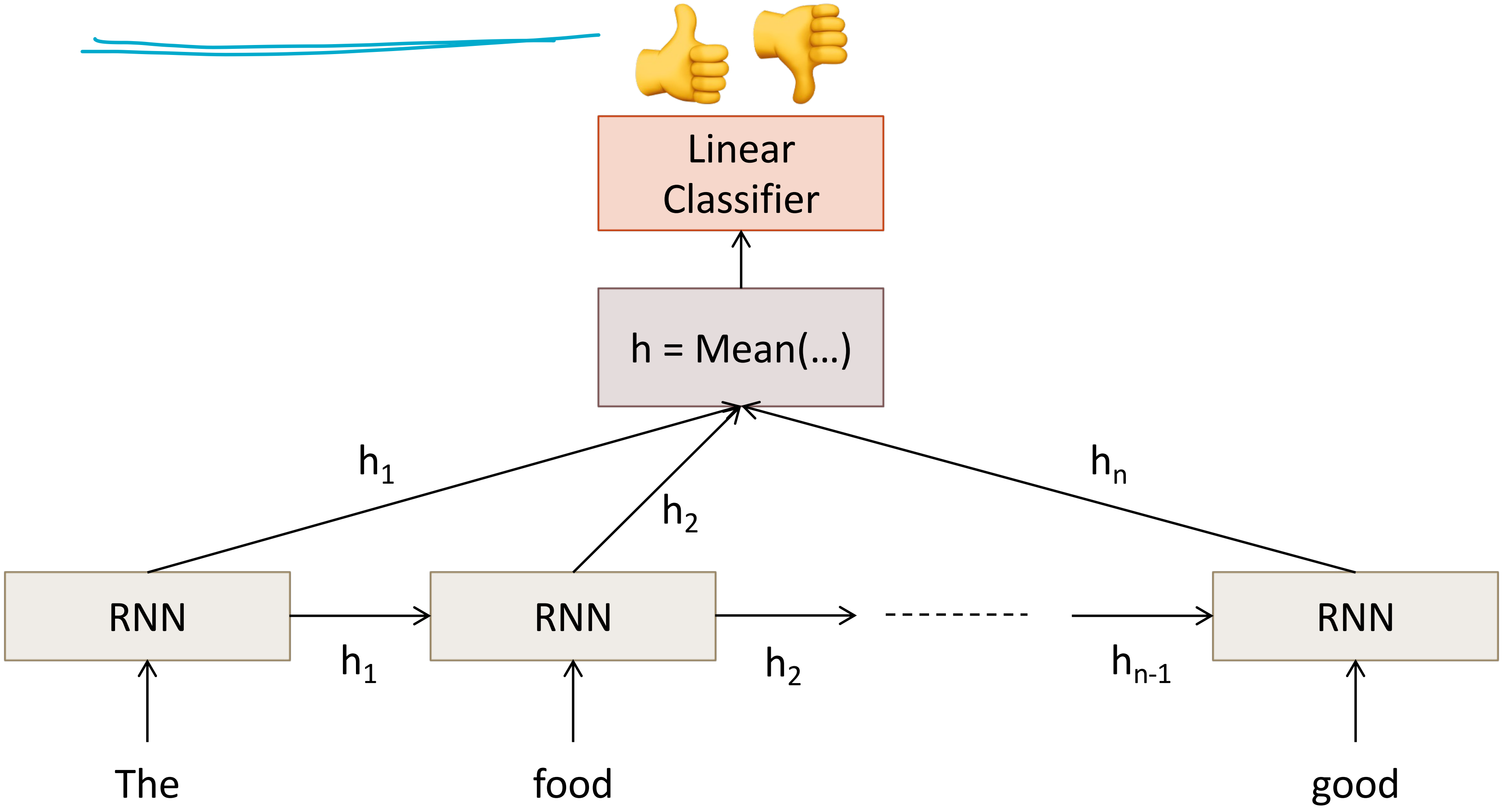
Sentiment analysis - solution 1

- [illegible]



Sentiment analysis – solution2

- [illegible]
  - A [illegible]

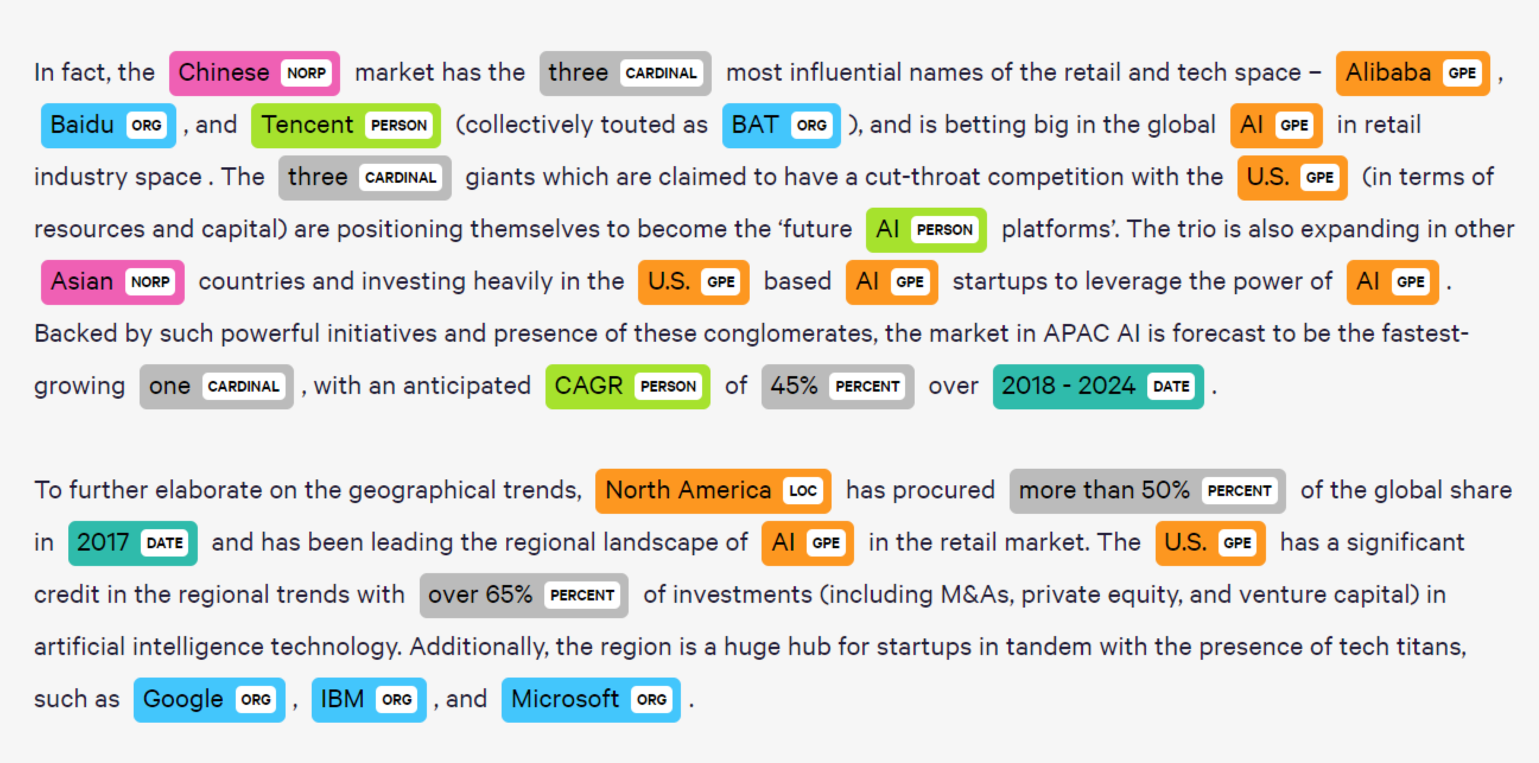


http://deeplearning.net/tutorial/lstm.html

Some use of RNN

→ Named Entity Recognition / Part of Speech Tagging

- A [illegible]
  - [illegible]
    - C[illegible]: [illegible], [illegible], [illegible], [illegible] (E[illegible])
    - C[illegible]: [illegible], [illegible], [illegible], [illegible] ([illegible])



In fact, the Chinese NORP market has the three CARDINAL most influential names of the retail and tech space – Alibaba GPE , Baidu ORG , and Tencent PERSON (collectively touted as BAT ORG), and is betting big in the global AI GPE in retail industry space . The three CARDINAL giants which are claimed to have a cut-throat competition with the U.S. GPE (in terms of resources and capital) are positioning themselves to become the 'future AI PERSON platforms'. The trio is also expanding in other Asian NORP countries and investing heavily in the U.S. GPE based AI GPE startups to leverage the power of AI GPE . Backed by such powerful initiatives and presence of these conglomerates, the market in APAC AI is forecast to be the fastest-growing one CARDINAL , with an anticipated CAGR PERSON of 45% PERCENT over 2018 - 2024 DATE .

To further elaborate on the geographical trends, North America LOC has procured more than 50% PERCENT of the global share in 2017 DATE and has been leading the regional landscape of AI GPE in the retail market. The U.S. GPE has a significant credit in the regional trends with over 65% PERCENT of investments (including M&As, private equity, and venture capital) in artificial intelligence technology. Additionally, the region is a huge hub for startups in tandem with the presence of tech titans, such as Google ORG , IBM ORG , and Microsoft ORG .

Extract information from text

Vente Villa 4 pièces Nice (06000)
Réf. 12390: Sur les Hauteurs de Nice. Superbe villa moderne (190m2), 2 chambres et 1 suite parentale, 3 salles de bain. Très grand salon/salle à manger, cuisine américaine équipée. Prestations de haut standing. Vue panoramique sur la mer. Cette villa a été construite en 2005. 1 270 000 euros. Si vous êtes intéressés, contactez vite Mimi LASOURIS 06.43.43.43. 43

REAL ESTATE TEMPLATE
Reference: 12390
Prize: 1 270 000
Surface: 190 m2
Year Built: 2005
Rooms: 4
Owner: Mimi LASOURIS
Telephone: 06.43.43.43. 43

Machine Learning approach

- F[illegible]
  - [illegible]
    - C[illegible] ([illegible])
    - [illegible]/[illegible] ([illegible])
    - [illegible]-[illegible]-[illegible]
  - [illegible]
    - [illegible] ([illegible]) [illegible]

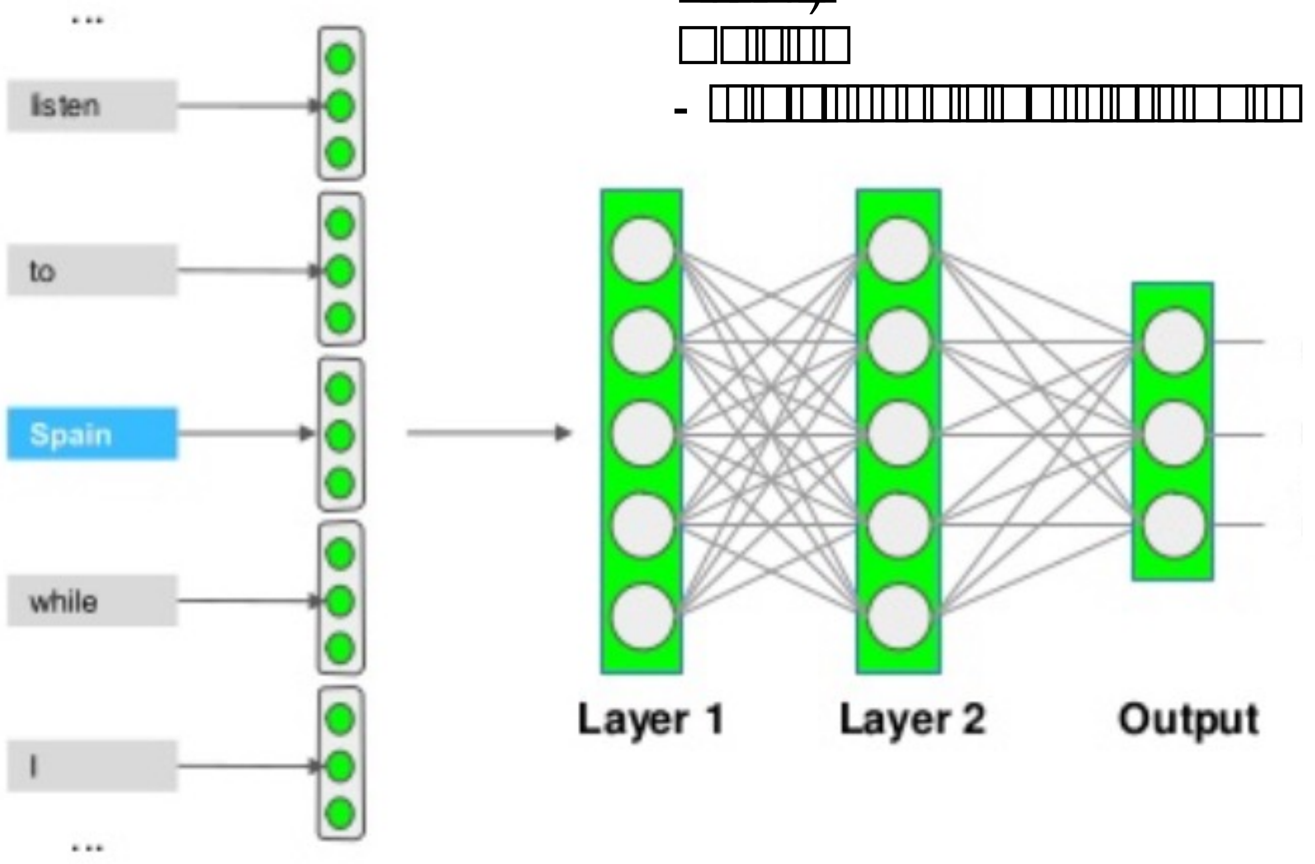
MLP for NER/POS

I[illegible]:

- [illegible] ([illegible] [illegible])

[illegible]

- [illegible]



[illegible]E[illegible]

[illegible]C

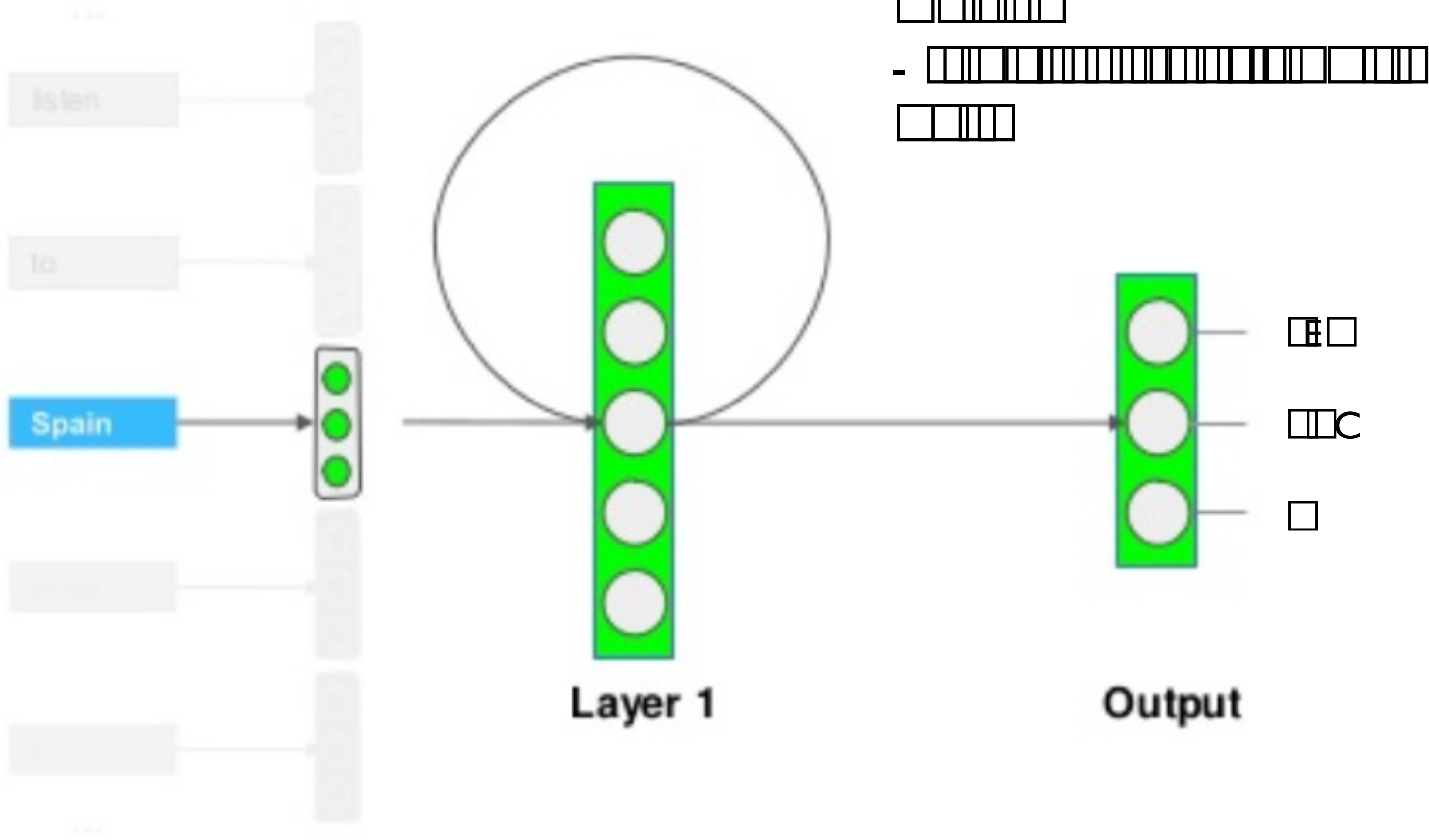
[illegible]

- A[illegible] [illegible]:[illegible]

Recurrent neural network for NER

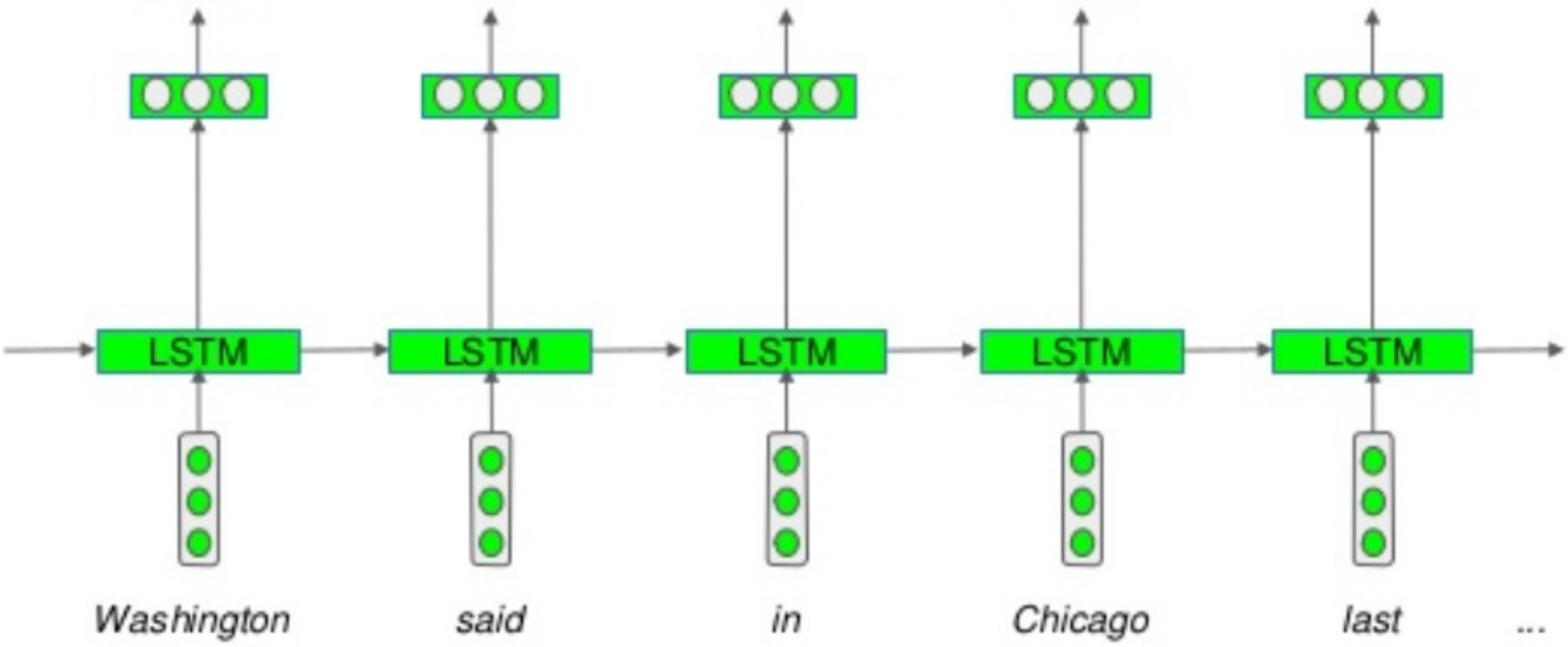
I[illegible]:

- [illegible] (.[illegible]. [illegible]) [illegible]
- [illegible]



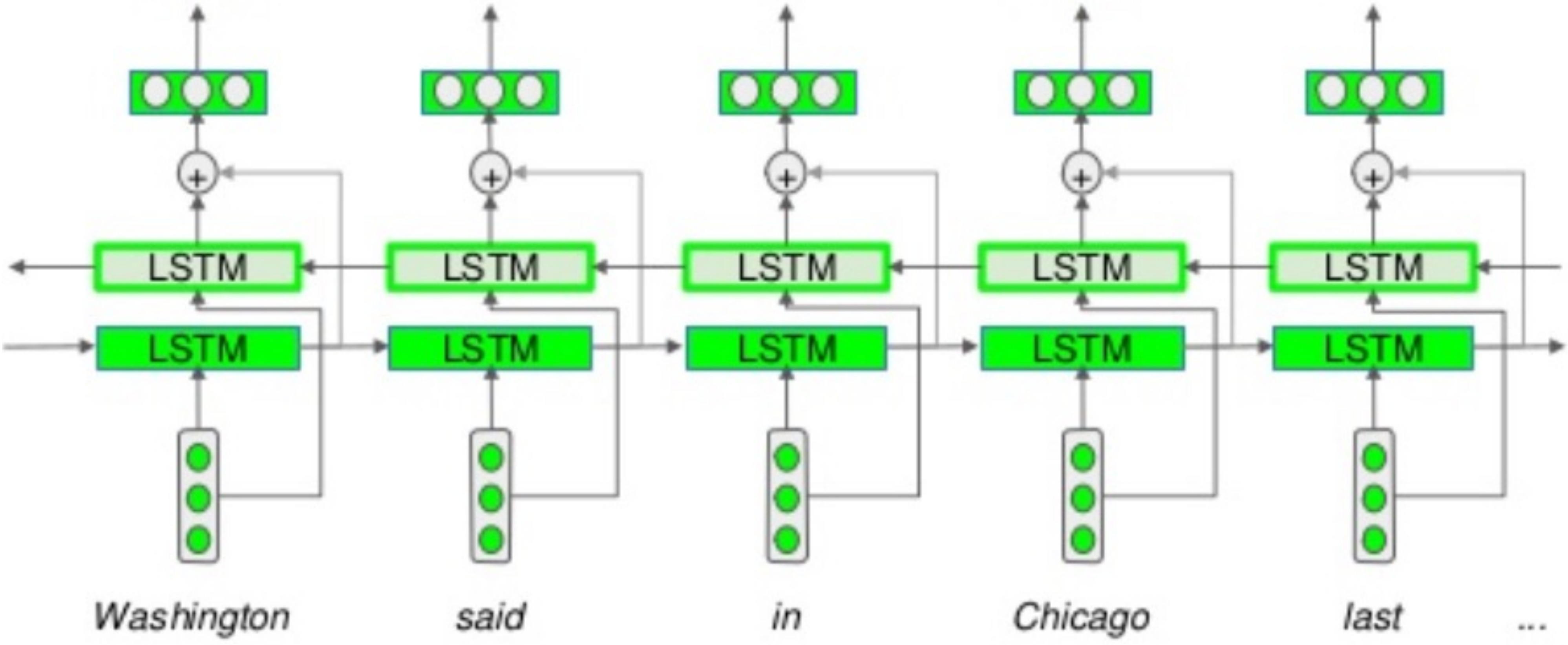
- A[illegible] [illegible]:[illegible]

Recurrent neural network for NER (same network but unfolded)



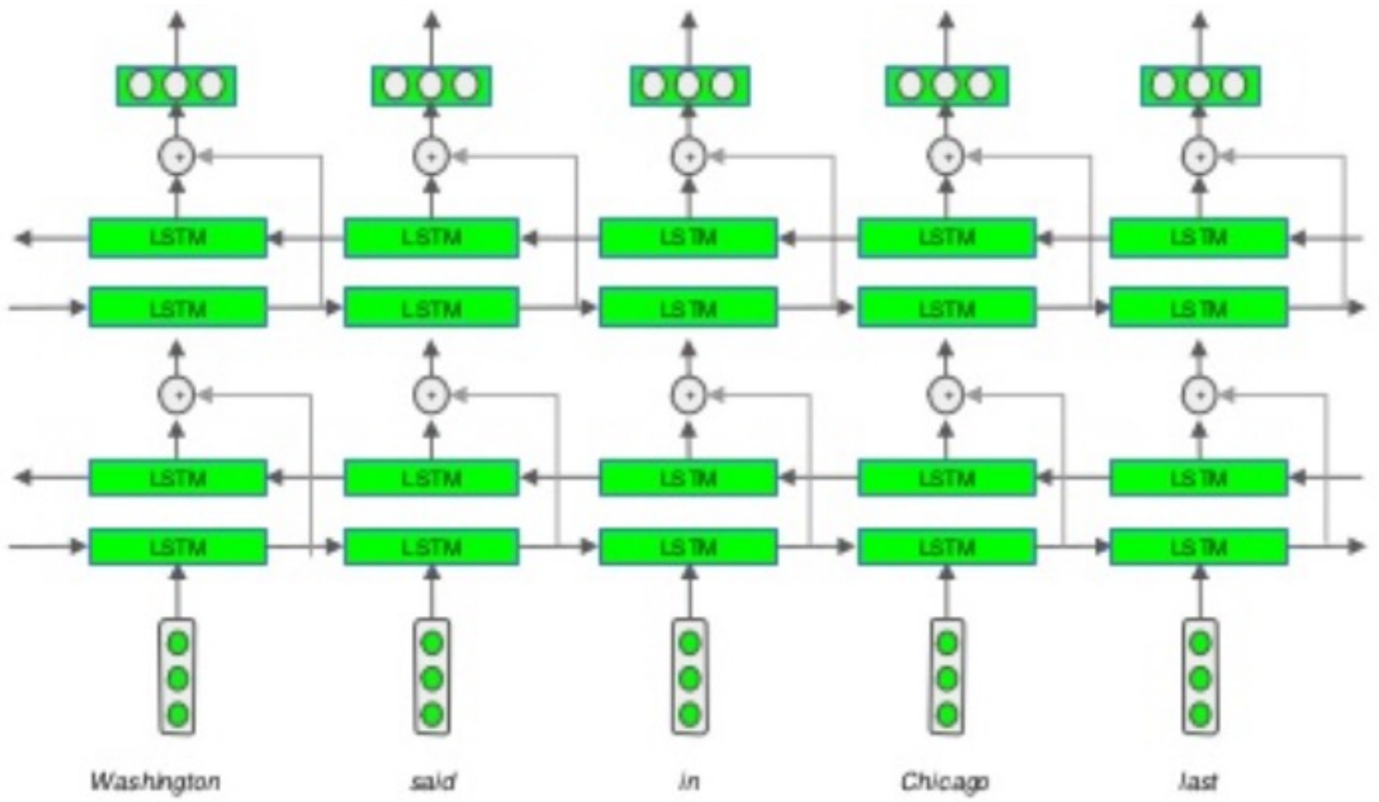
- A[illegible]
  [illegible]

Bi directional recurrent neural network for NER



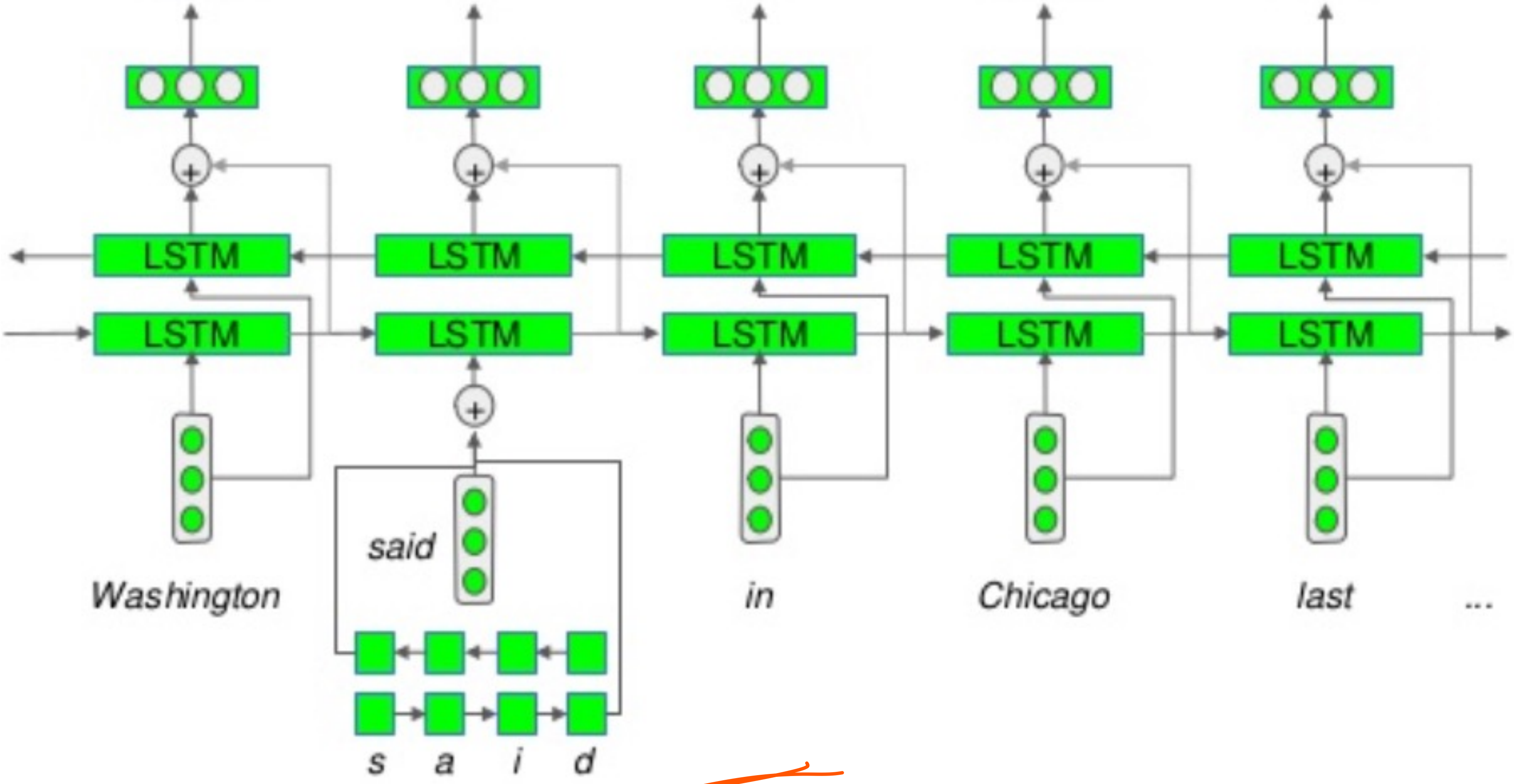
- A[illegible]
  [illegible]:[illegible]

Stacked Bi-RNN



- A [illegible]

Multi-level encoding
char encoding + word encoding



Today Lab

F[illegible]

- B[illegible]
  - [illegible]
  - [illegible] ([illegible])
  - [illegible] ([illegible])
  - [illegible]: ([illegible], [illegible], [illegible])
  - [illegible]: ([illegible], 1, [illegible]) [illegible] ([illegible], [illegible]) B[illegible] 2 [illegible]
- F[illegible]
  - A[illegible] D[illegible]
    - [illegible] = "[illegible]://[illegible].3[illegible]/[illegible]/[illegible]/[illegible].[illegible]"
    - [illegible] = [illegible].[illegible]([illegible], [illegible]="[illegible]", [illegible]="[illegible]", [illegible]="[illegible]")
  - [illegible] 24 [illegible]
  - [illegible] 5 [illegible] ([illegible].[illegible].[illegible])
  - [illegible] E [illegible]