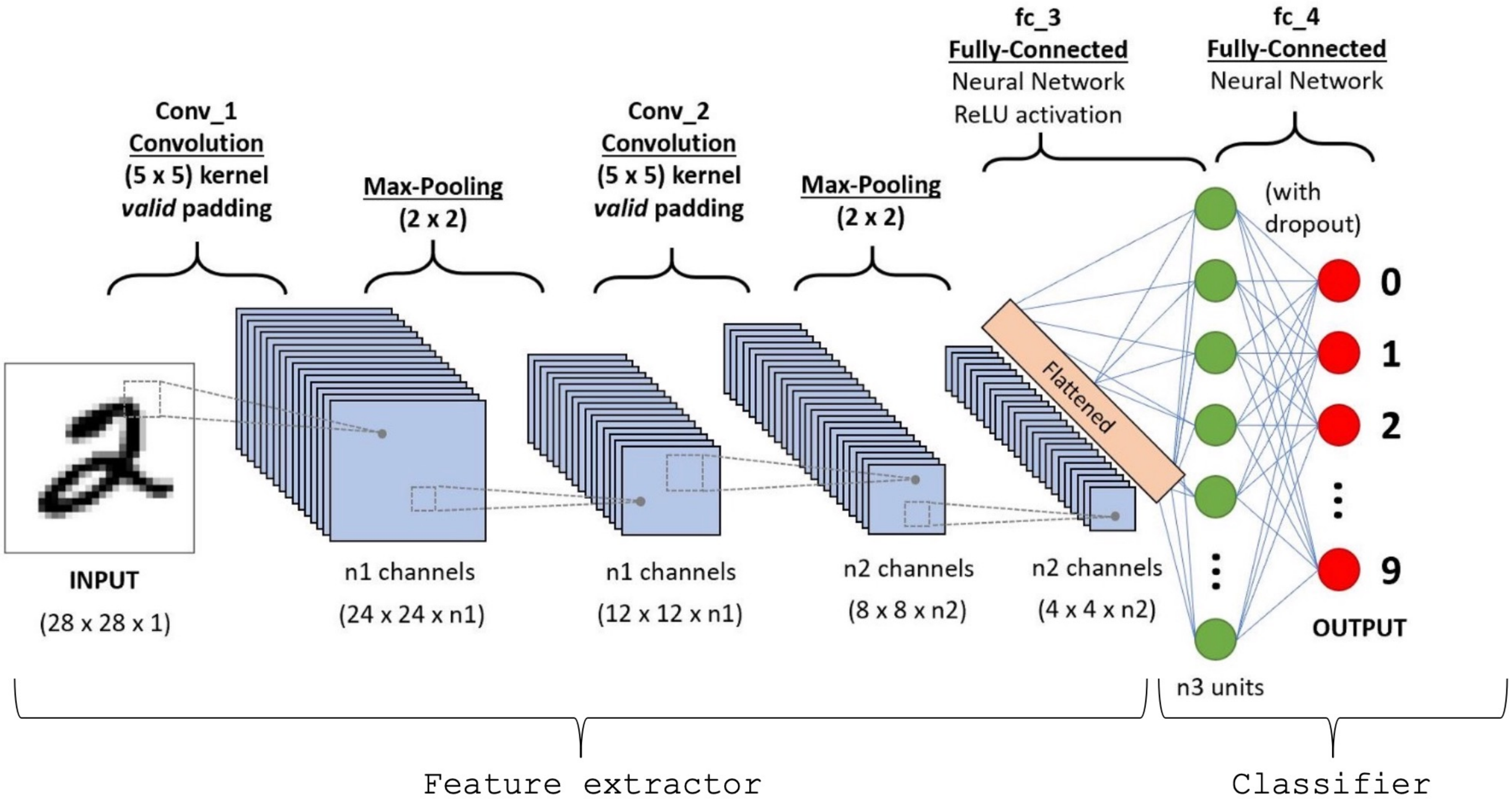
Deep Learning: Session 10

RNNs (LSTM’s)

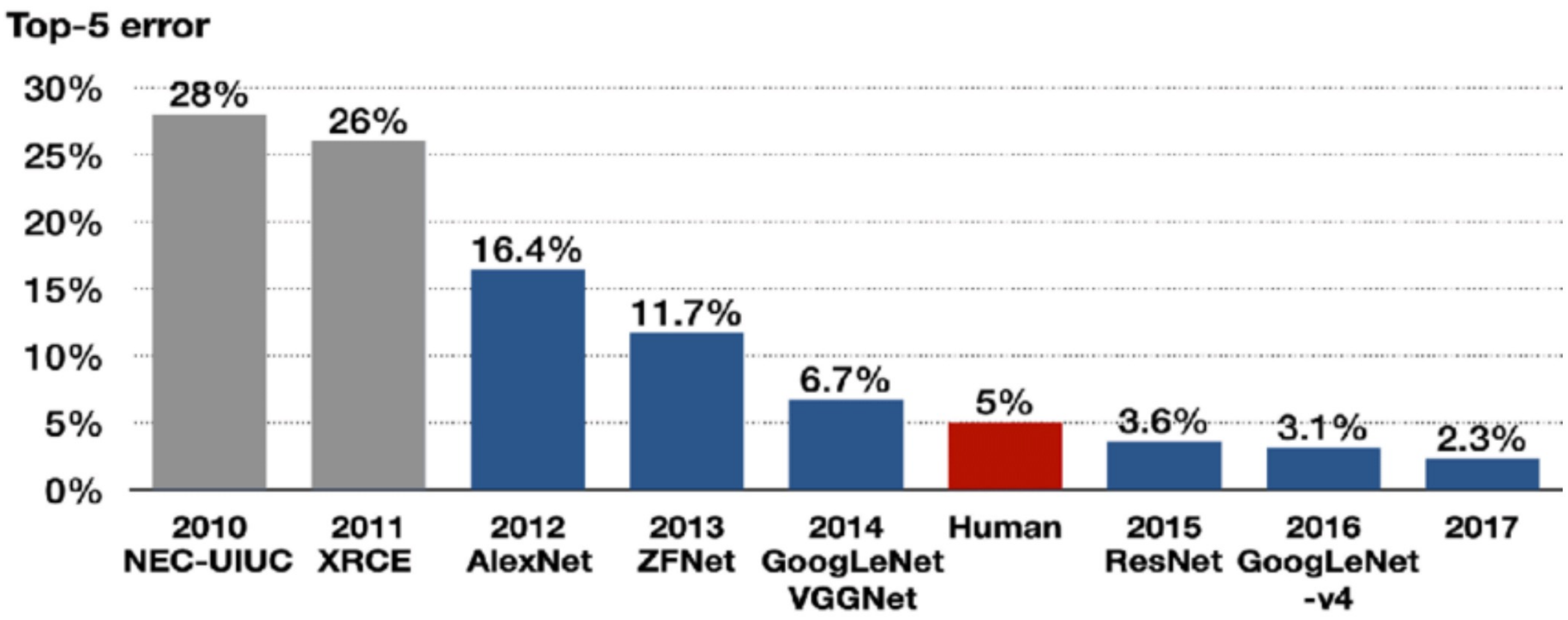
Outline

1. Recap
2. Sequence Data
3. RNNs
4. LSTMs
5. Transformers

The architecture: Classifier

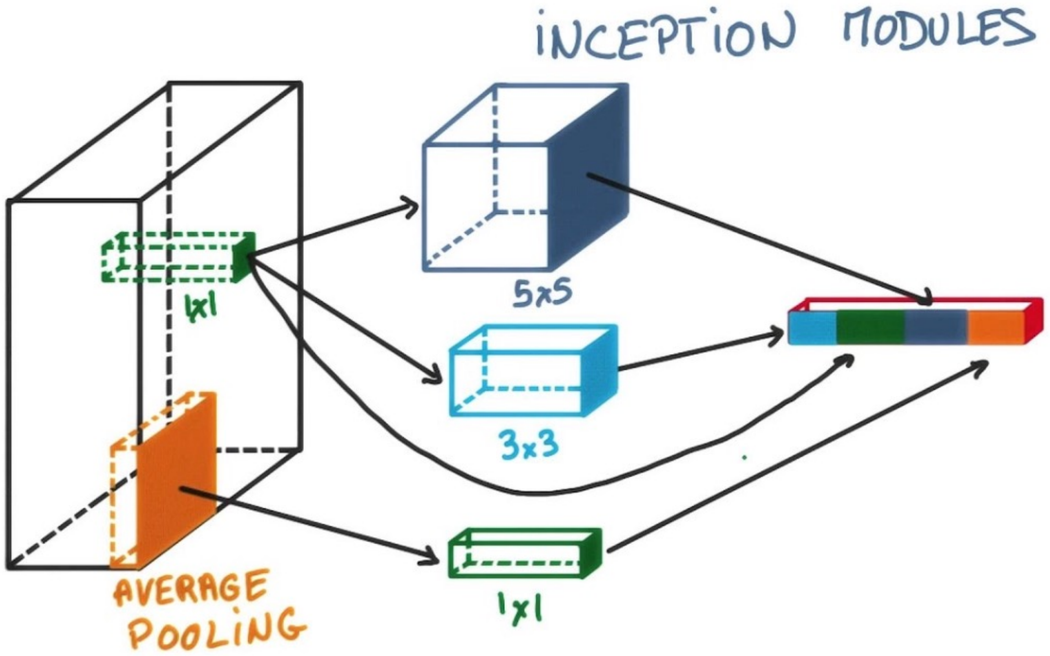


ILSVRC Error Evolution



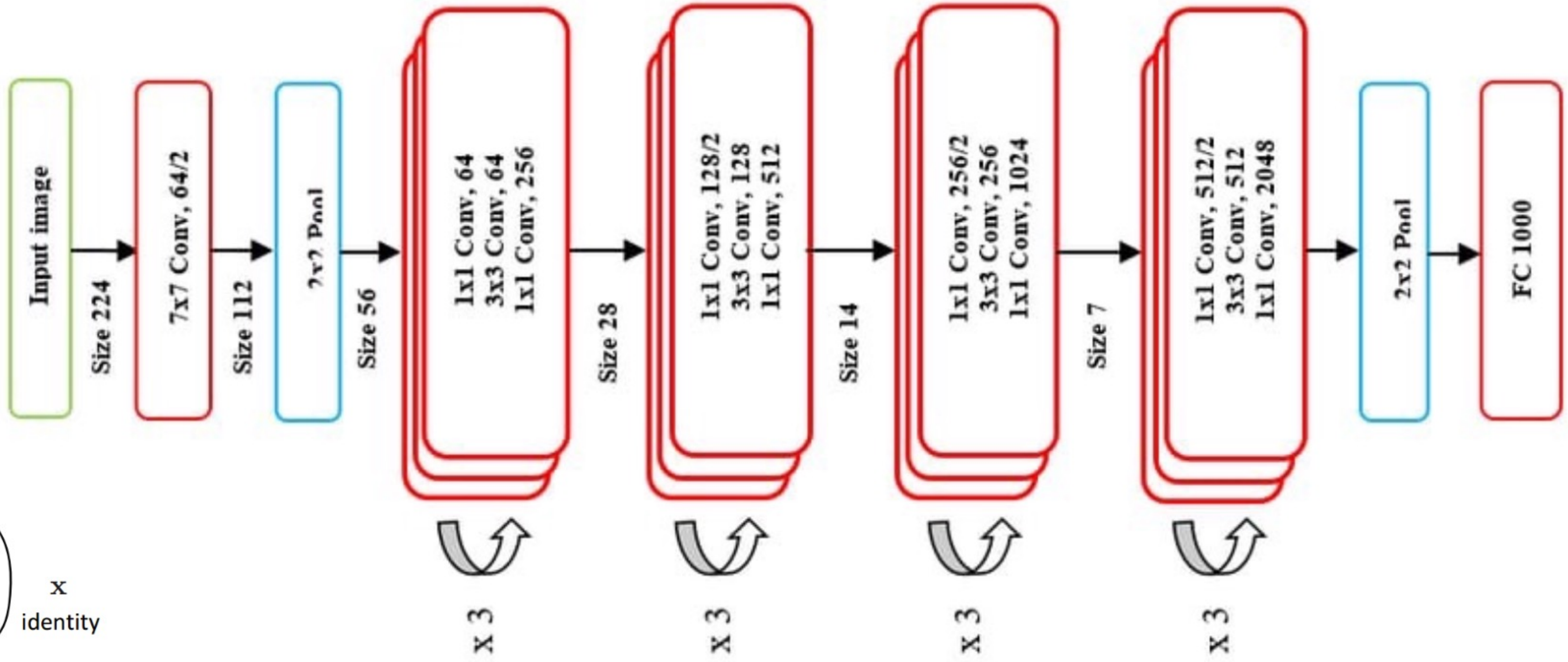
Famous CNNs

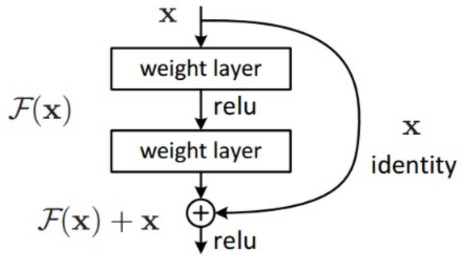
- **GoogleLenet – Inception (2014-)**
  - ~6.6M trainable params
  - 22 layers
  - Inception modules combine different filter sizes and pooling layers



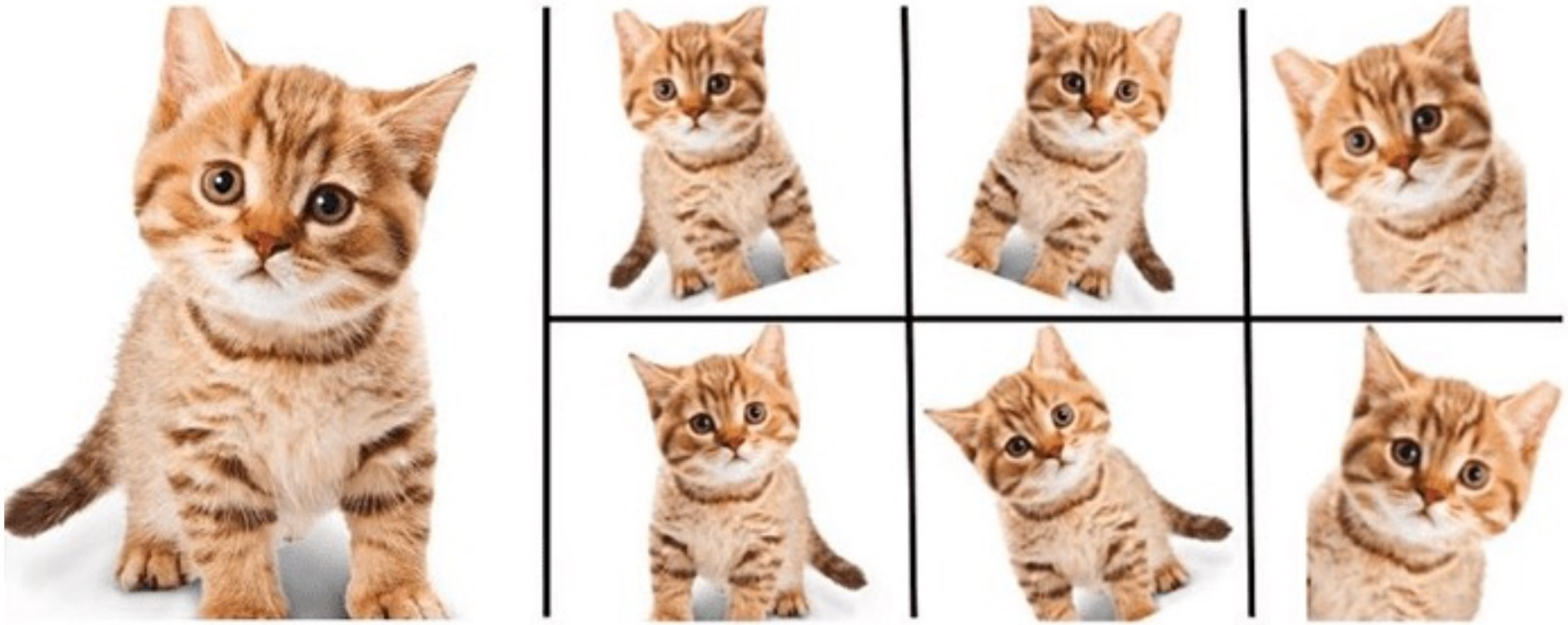
Famous CNNs

- **ResNet - 50 (2015)**
  - ~25.6M trainable params
  - 50 layers!
  - Use skip connections to deal with the vanishing gradient problem



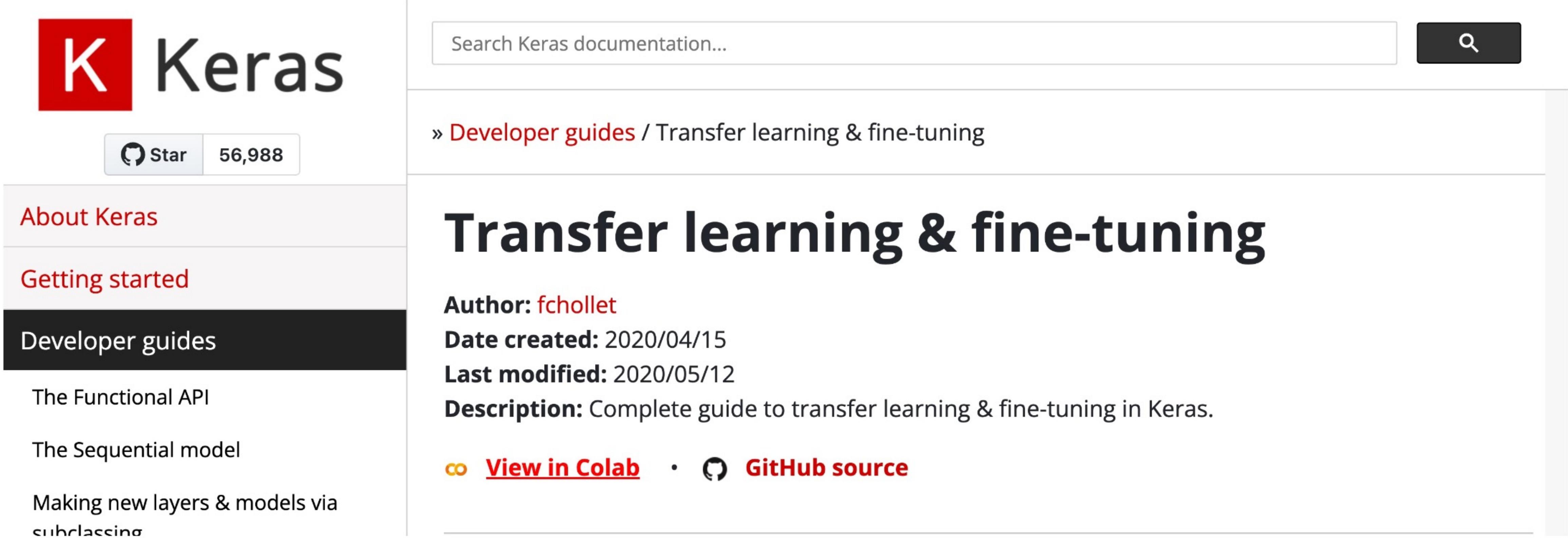


Data Augmentation



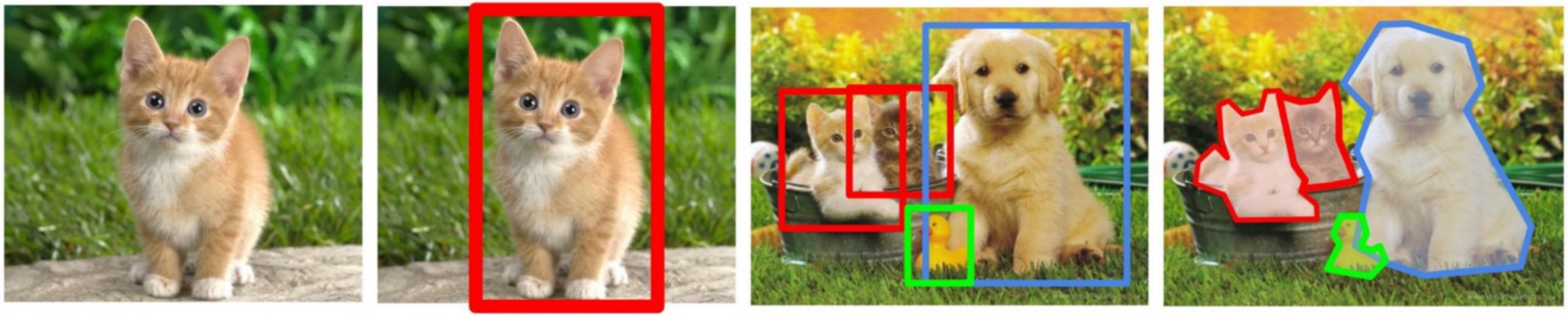
Transfer Learning

https://keras.io/guides/transfer_learning/



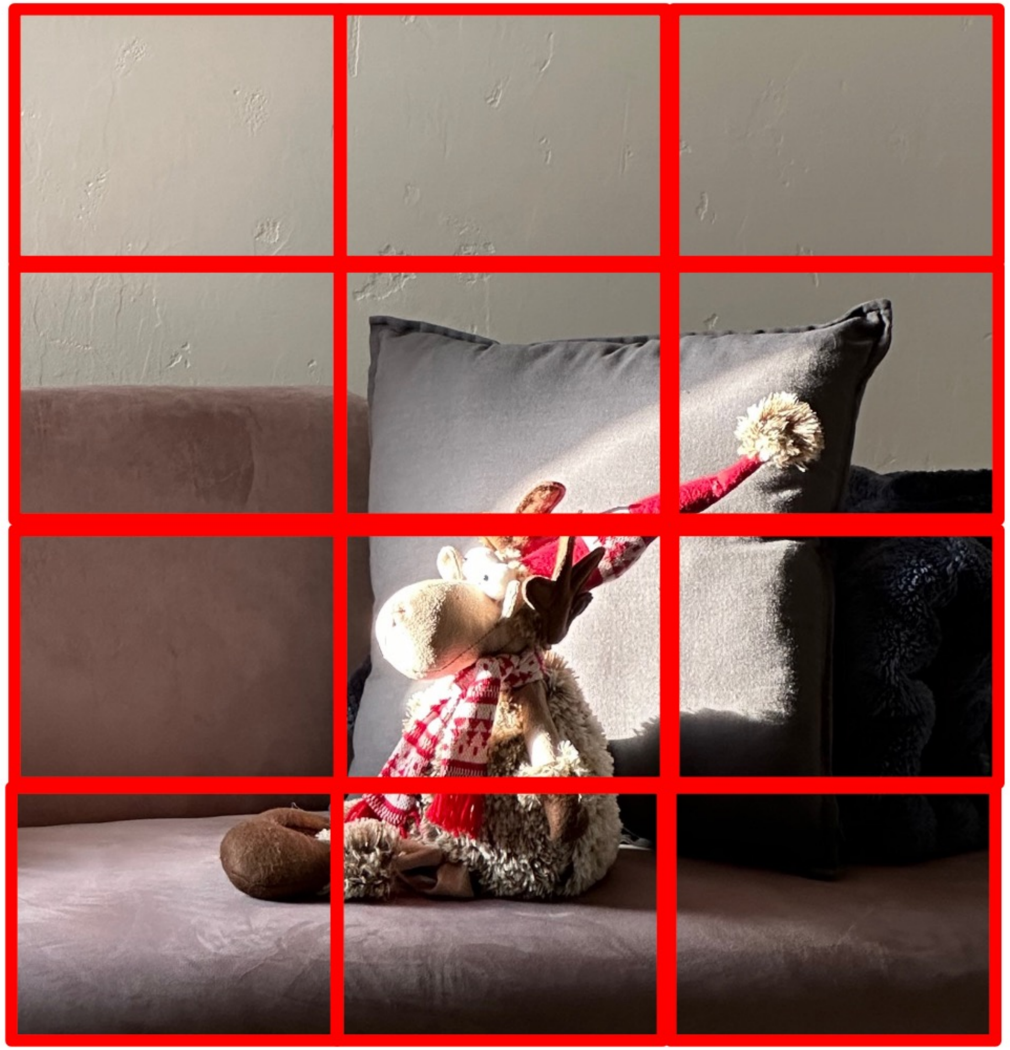
Applications

Classification | **Classification & Localization** | **Object Detection** | **Instance Segmentation**



CAT | CAT | CAT, DOG, DUCK | CAT, DOG, DUCK

OD: YOLO



- Allows to get more precise bounding boxes with different sizes.

- We run (using convolution) a classification and localization convnet over a grid

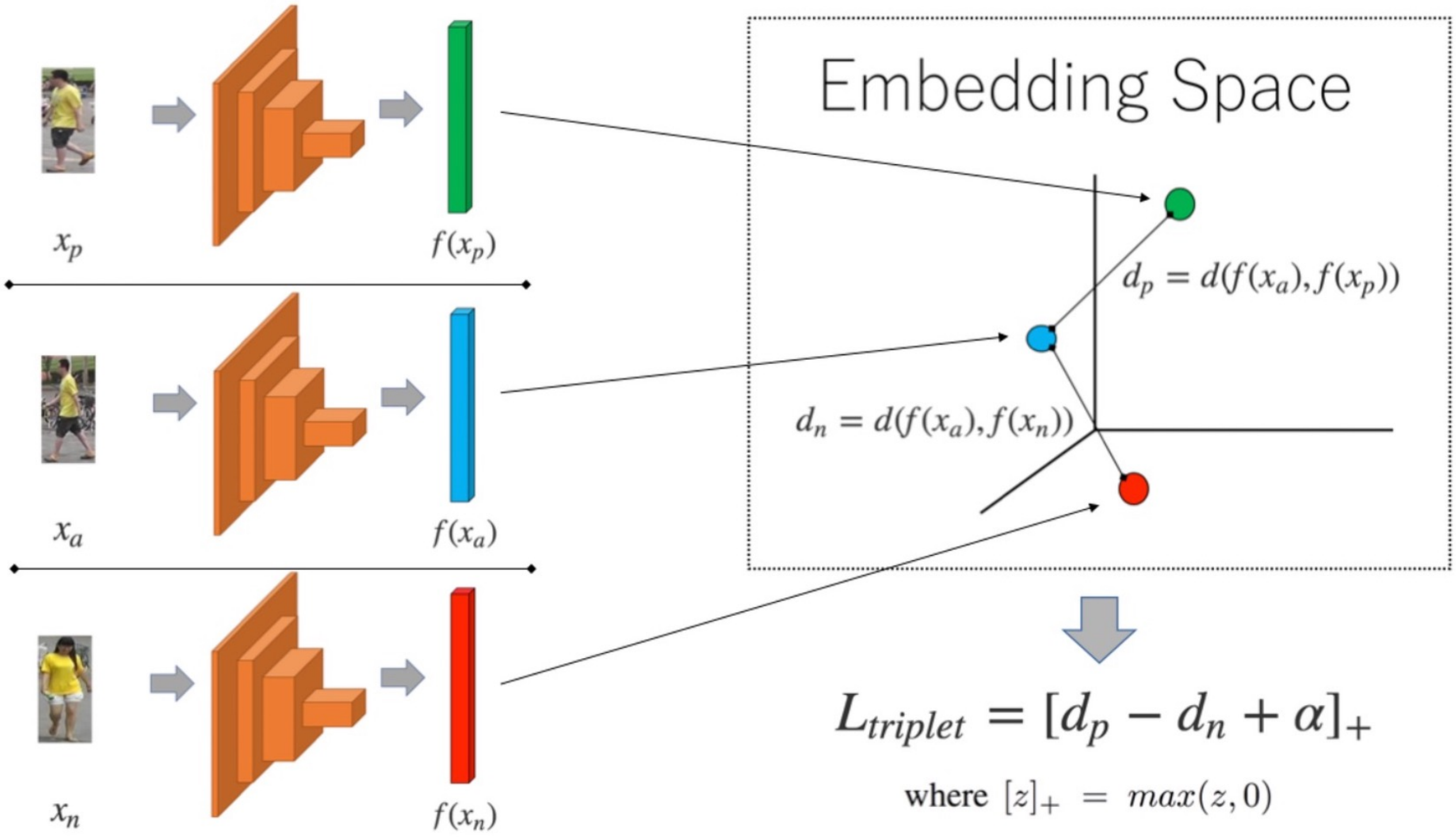
- Labels include the actual bounding boxes

- Engineering:

  – Non-Max Suppression

  – Anchor boxes

Face Verification



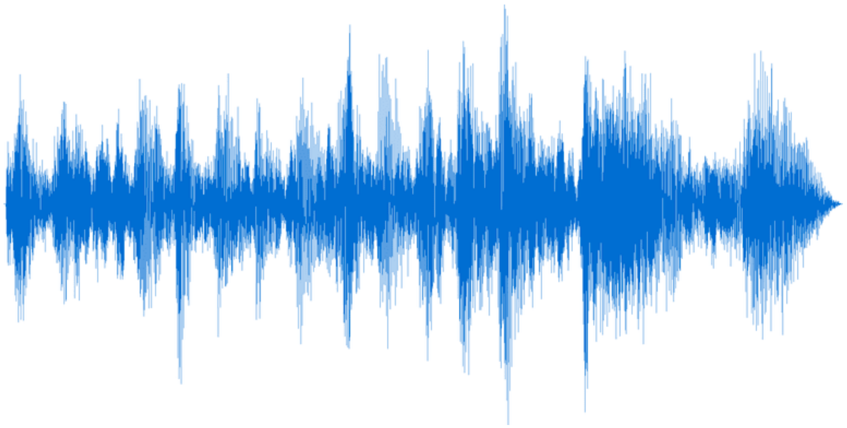
$$L_{triplet} = [d_p - d_n + \alpha]_+$$

where $[z]_+ = max(z, 0)$

Outline

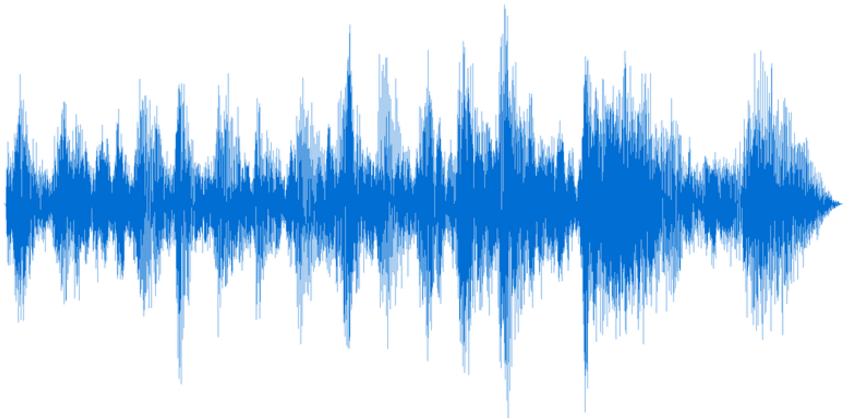
1. Recap
2. Sequence Data
3. RNNs
4. LSTMs
5. Transformers

Sequence Data



"Hello, Hello, Hello, is anybody in there?"

Sequence Data

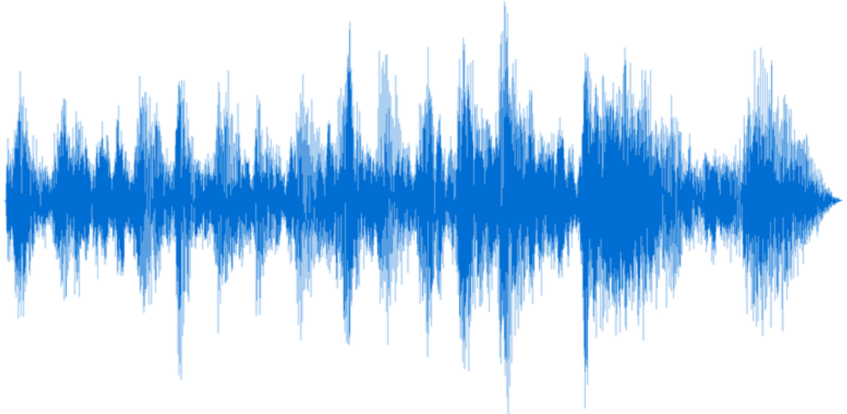


“Hello, Hello, Hello, is anybody in there?”

“Hello, Hello, Hello, is anybody in there?”

“Hola, Hola, Hola, ¿hay alguien ahí?”

Sequence Data



“Hello, Hello, Hello, is anybody in there?”

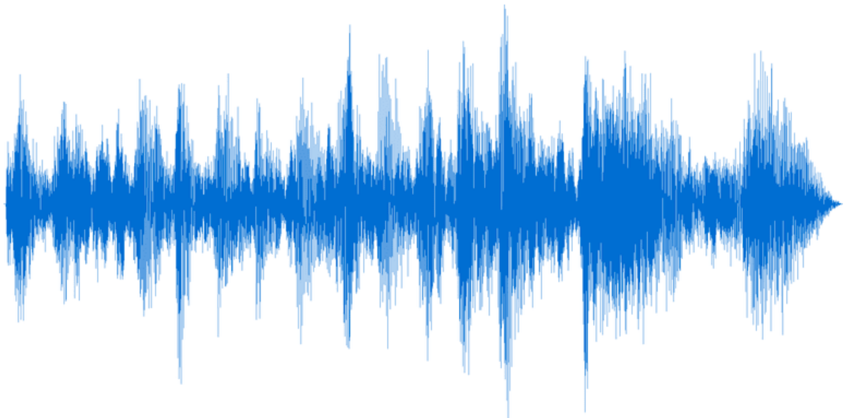
“Hello, Hello, Hello, is anybody in there?”

“Hola, Hola, Hola, ¿hay alguien ahí?”

INPUT IS A SEQUENCE OF N

OUTPUT IS A SEQUENCE OF M

Sequence Data



“Hello, Hello, Hello, is anybody in there?”

“Hello, Hello, Hello, is anybody in there?” → “Hola, Hola, Hola, ¿hay alguien ahí?”

INPUT IS A SEQUENCE OF N

OUTPUT IS A SEQUENCE OF M

TIME

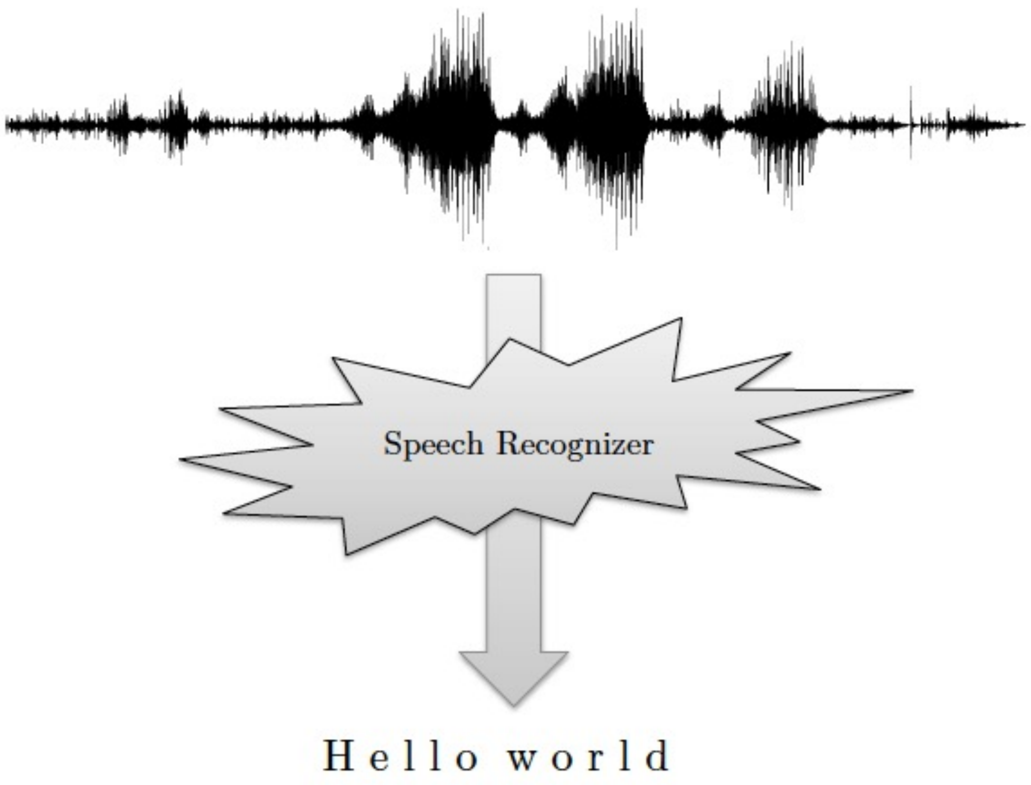
Sequence Data: Representing Words

"Hello, Hello, Hello, is anybody in there?" → "Hola, Hola, Hola, ¿hay alguien ahí?"

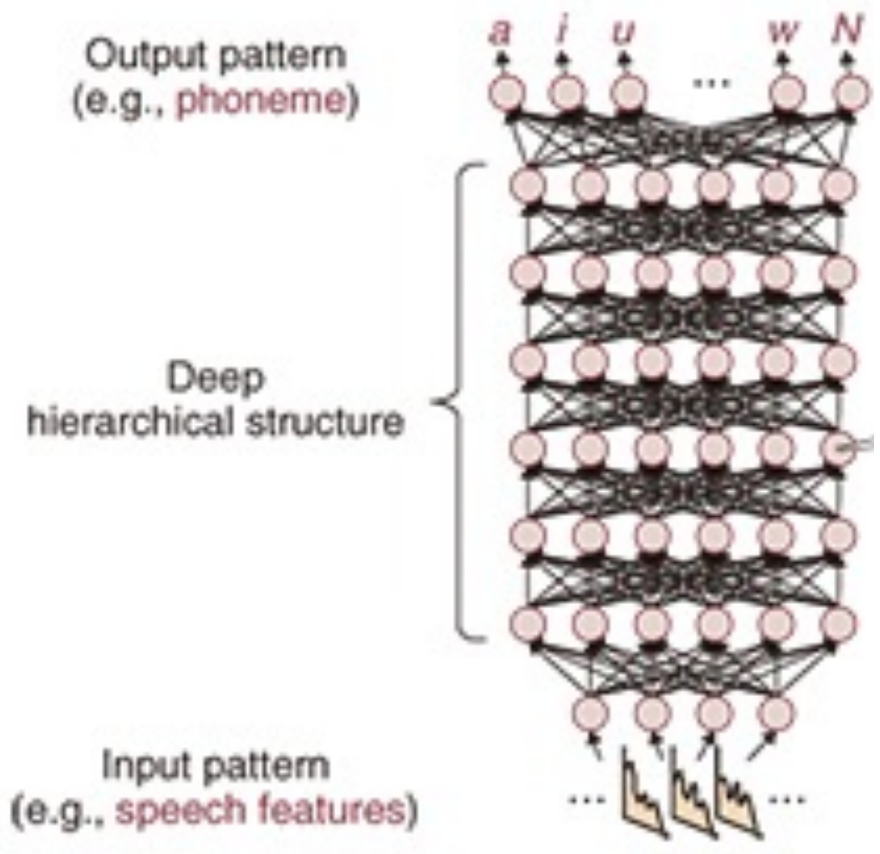
DICTIONARY	ONE HOT VECTOR
A	0
…	…
And	0
…	…
Hello	1
…	…
Zoo	0

DICTIONARY	ONE HOT VECTOR
A	0
…	…
Animal	0
…	…
Hola	1
…	…
Zoo	0

What if we use a multilayer perceptron?



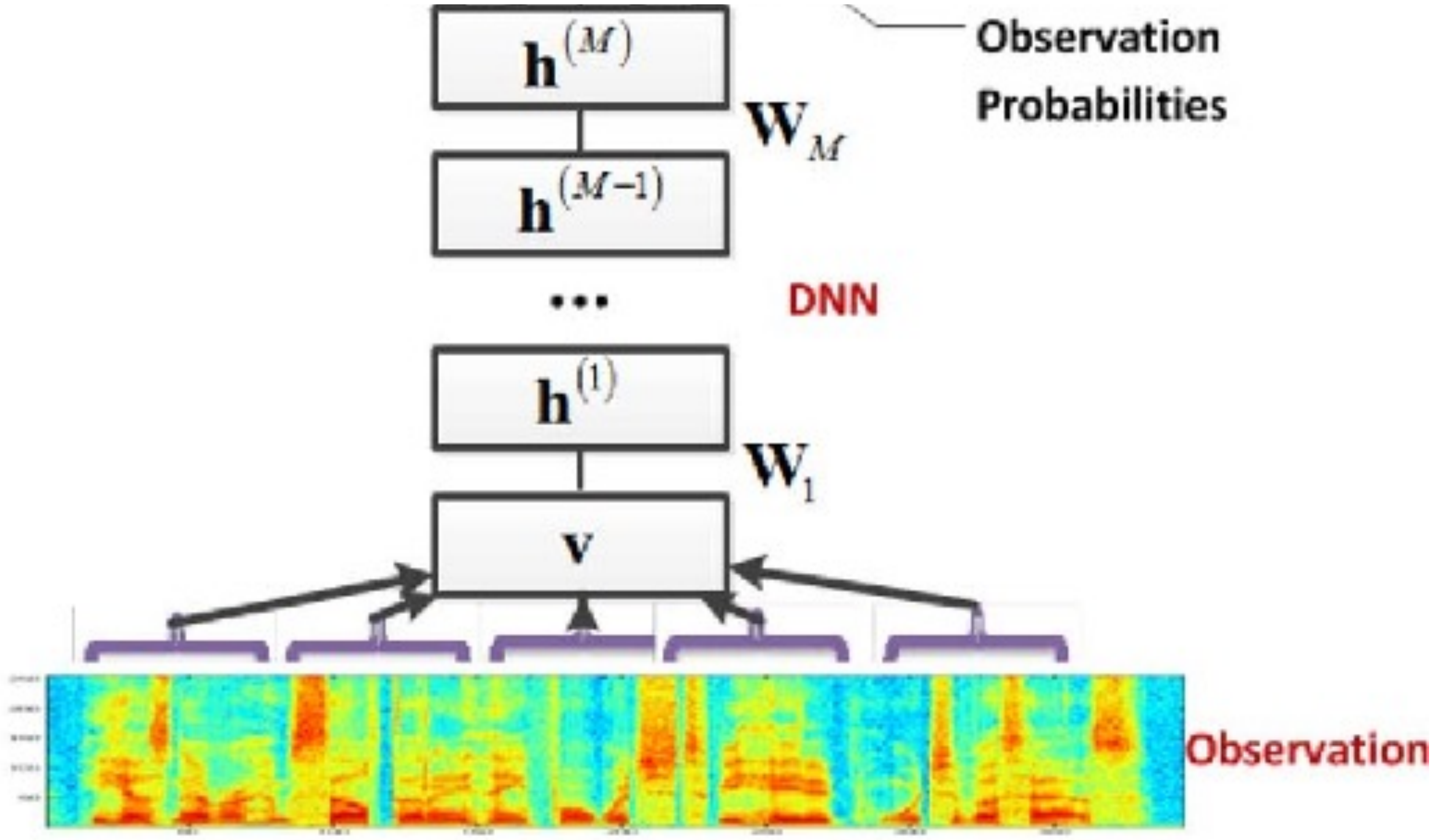
What if we use a multilayer perceptron?



What if we use a multilayer perceptron?

- But we know voice is a sequence… can we do better?

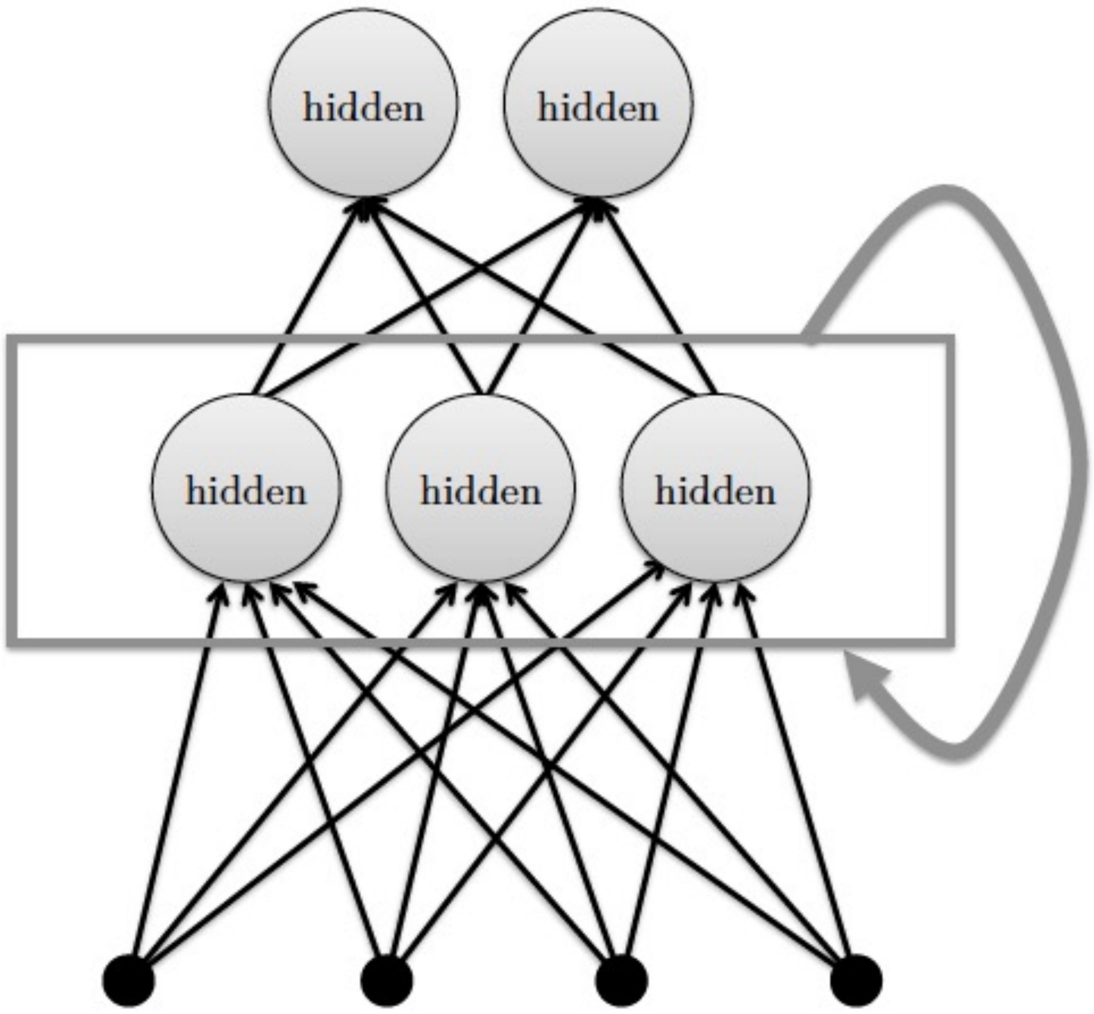
What if we use a multilayer perceptron?



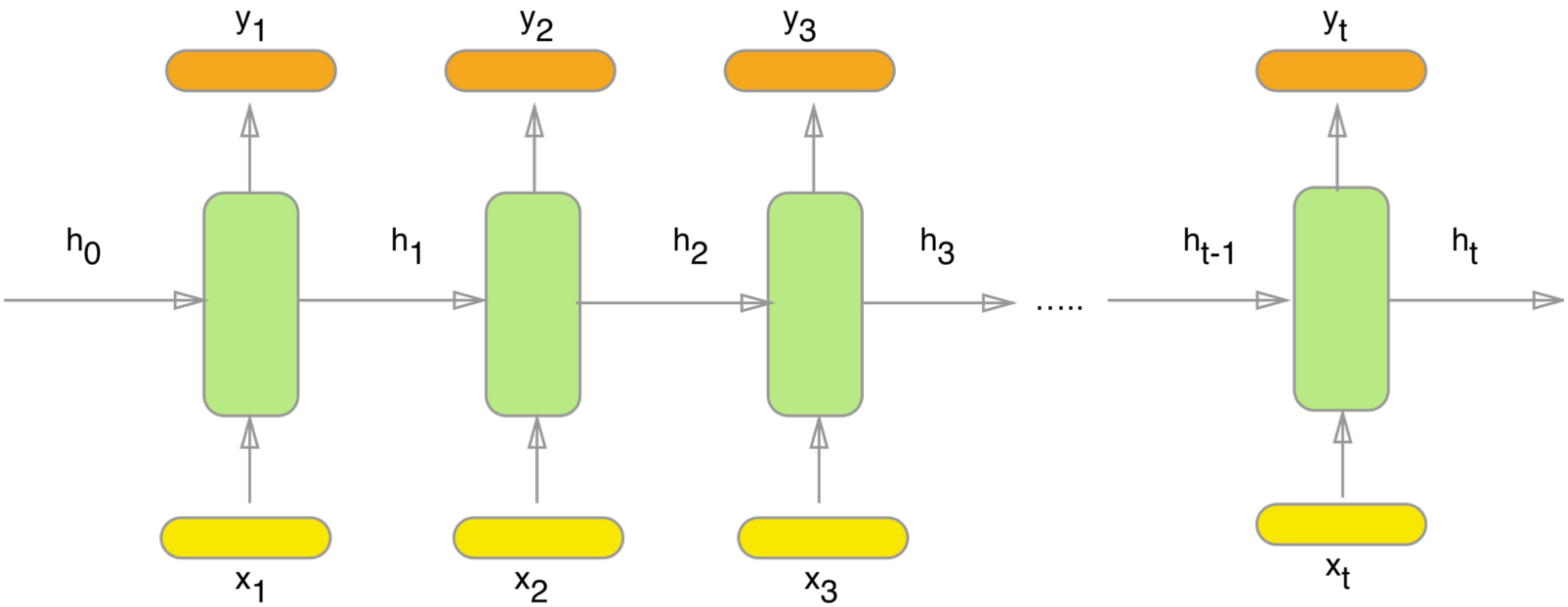
What if we use a multilayer perceptron?

- Infeasibility. The size of the input and therefore parameters in first layer can be enormous.
- The length of the input/output sequence can be different for every training example.
- As the multilayer perceptron for images, there is no way to take advantage of the sentence structure.

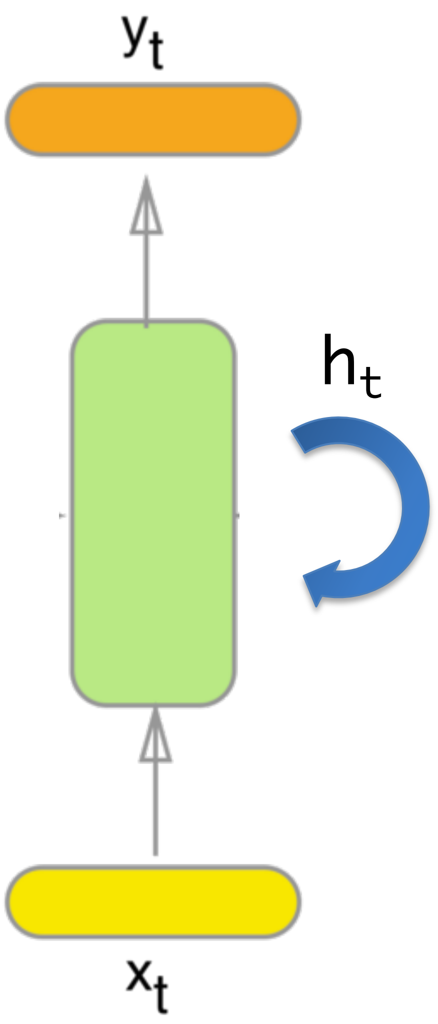
Recurrent Neural Network



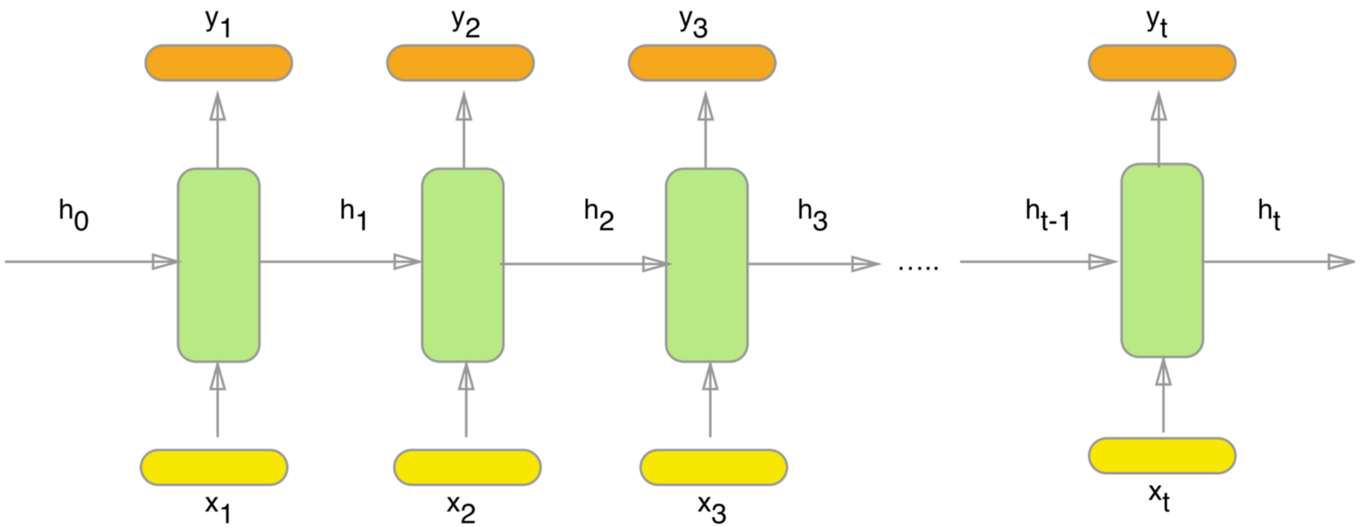
Recurrent Neural Network



Recurrent Neural Network



Recurrent Neural Network

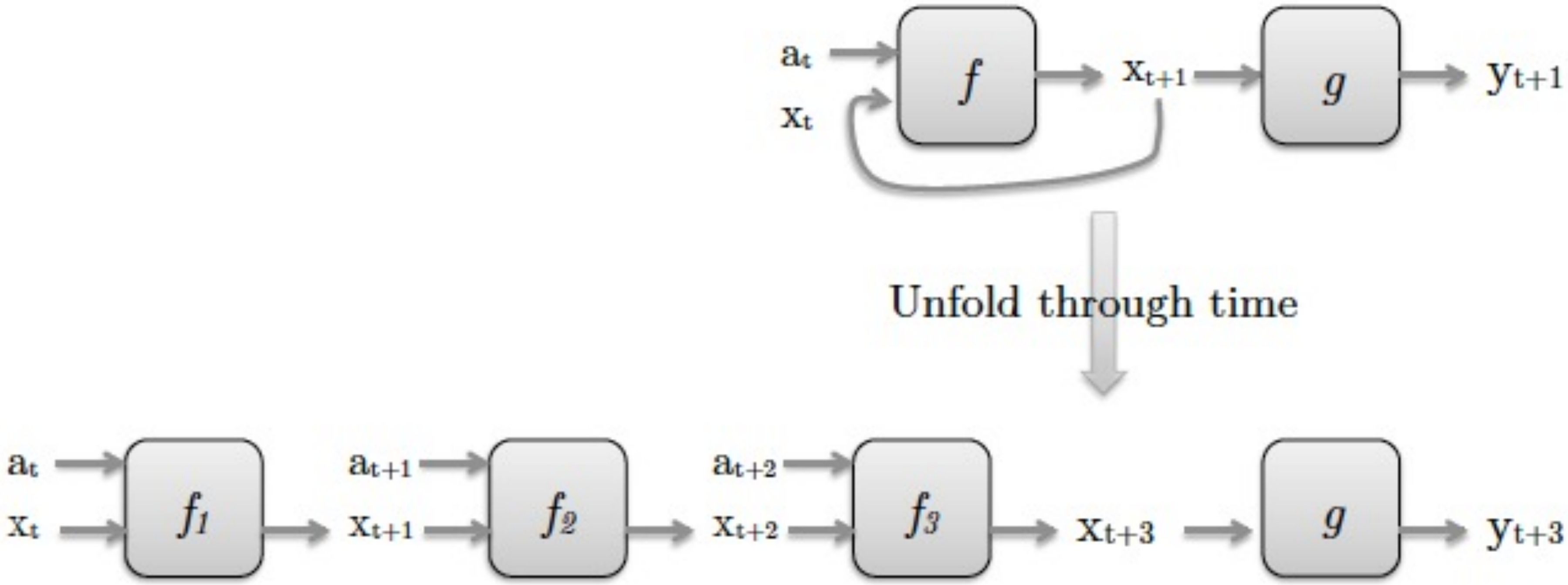


$$h_t = g(W_h(x_t, h_{t-1}) + b_h)$$

$$y_t = g(W_y h_t + b_y)$$

How we train this?: BPTT

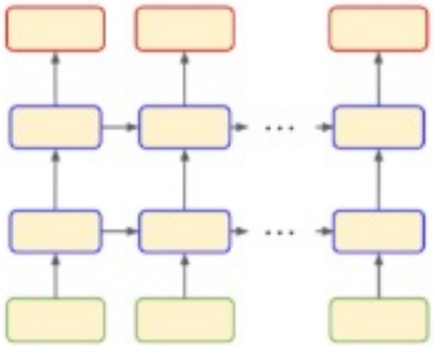
We have to modify the backpropagation algorithm: backpropagation through time



Recurrent Neural Network

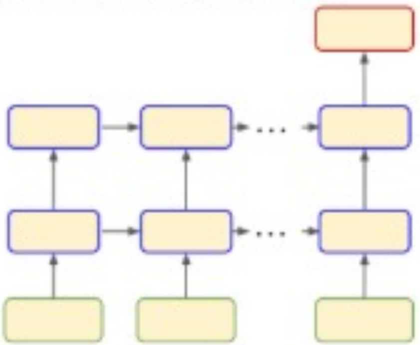
- RNNs architectures

MANY TO MANY:



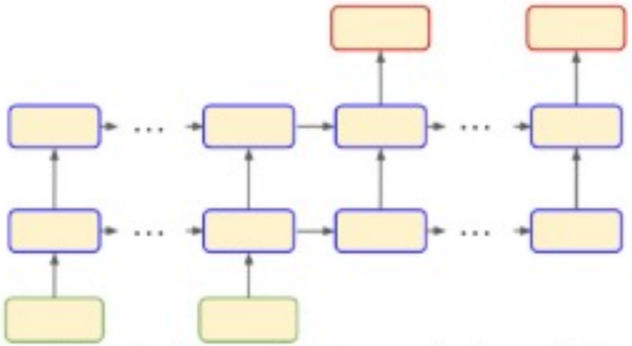
e.g. Frequency samples in, phonemes out.

MANY TO ONE:



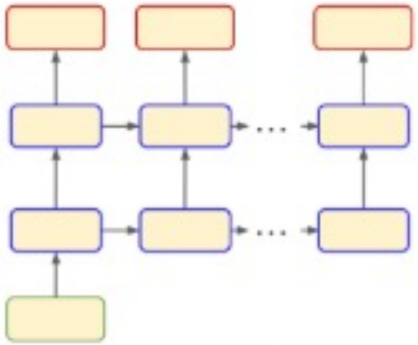
e.g. Proposed model for SV and LID.

MANY TO MANY:



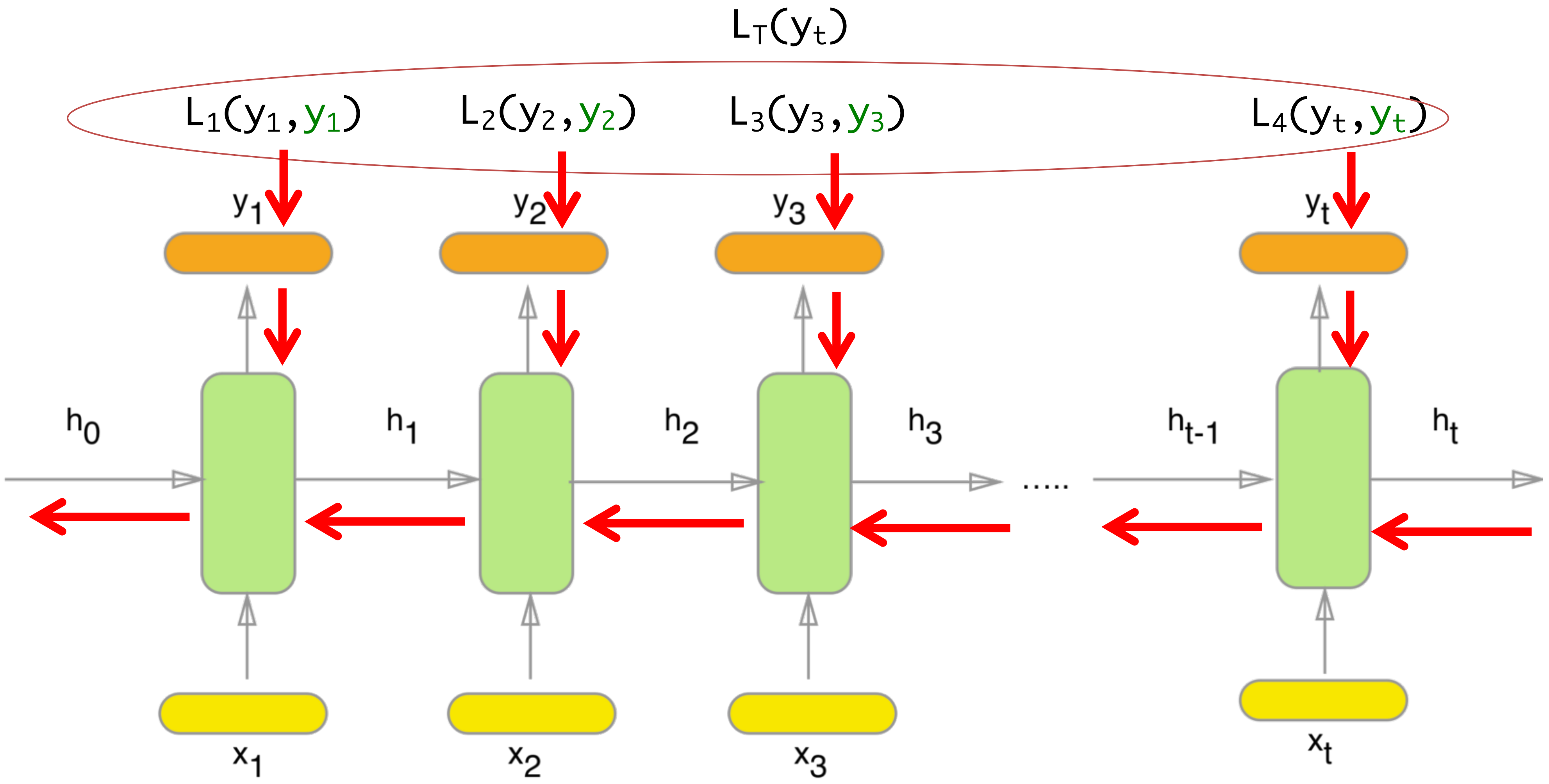
e.g. Words in spanish in, words in english out.

ONE TO MANY:



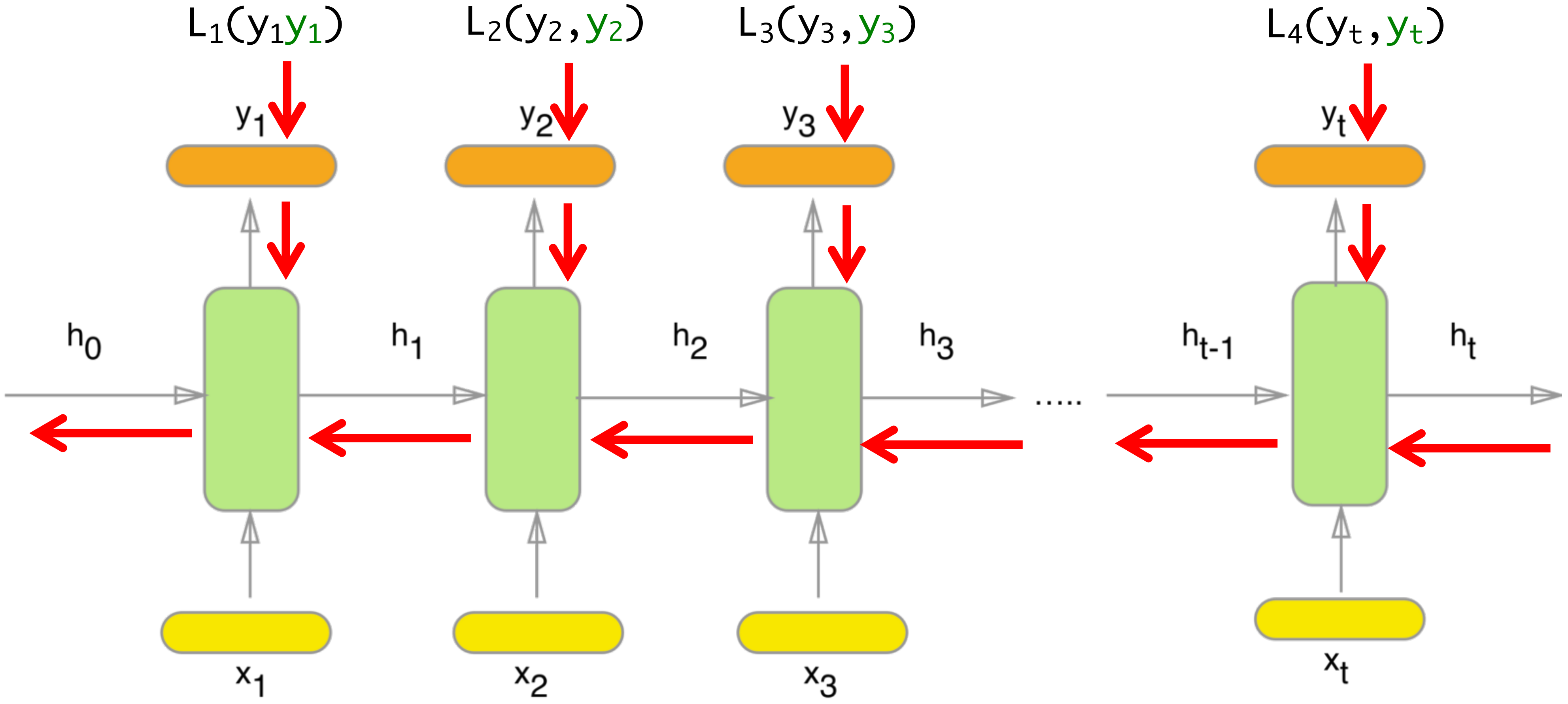
e.g. Image to word sequence.

The problem: Vanishing gradients



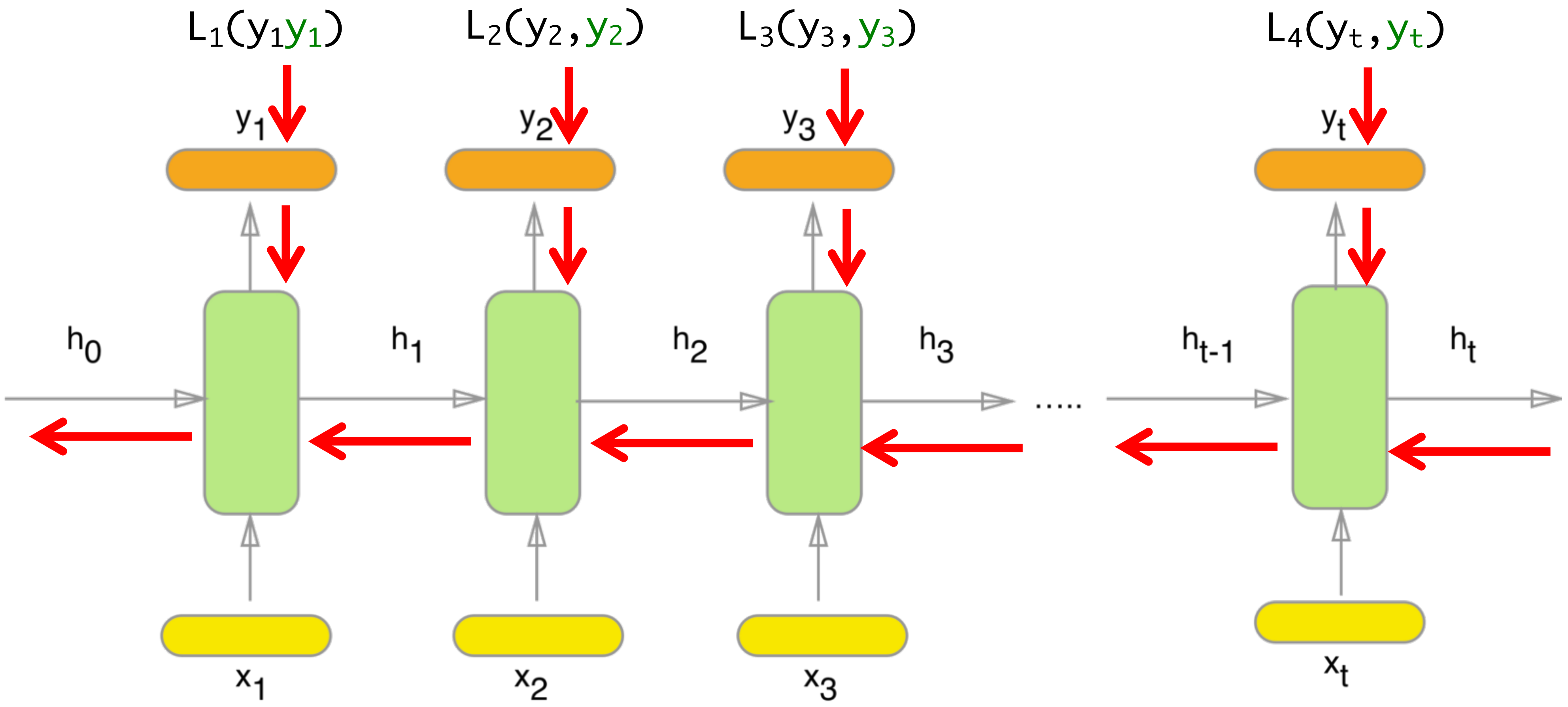
The problem: Vanishing gradients

She, who stayed in the train station waiting for him…, was wondering if
We, who stayed in the train station waiting for him…, were wondering if

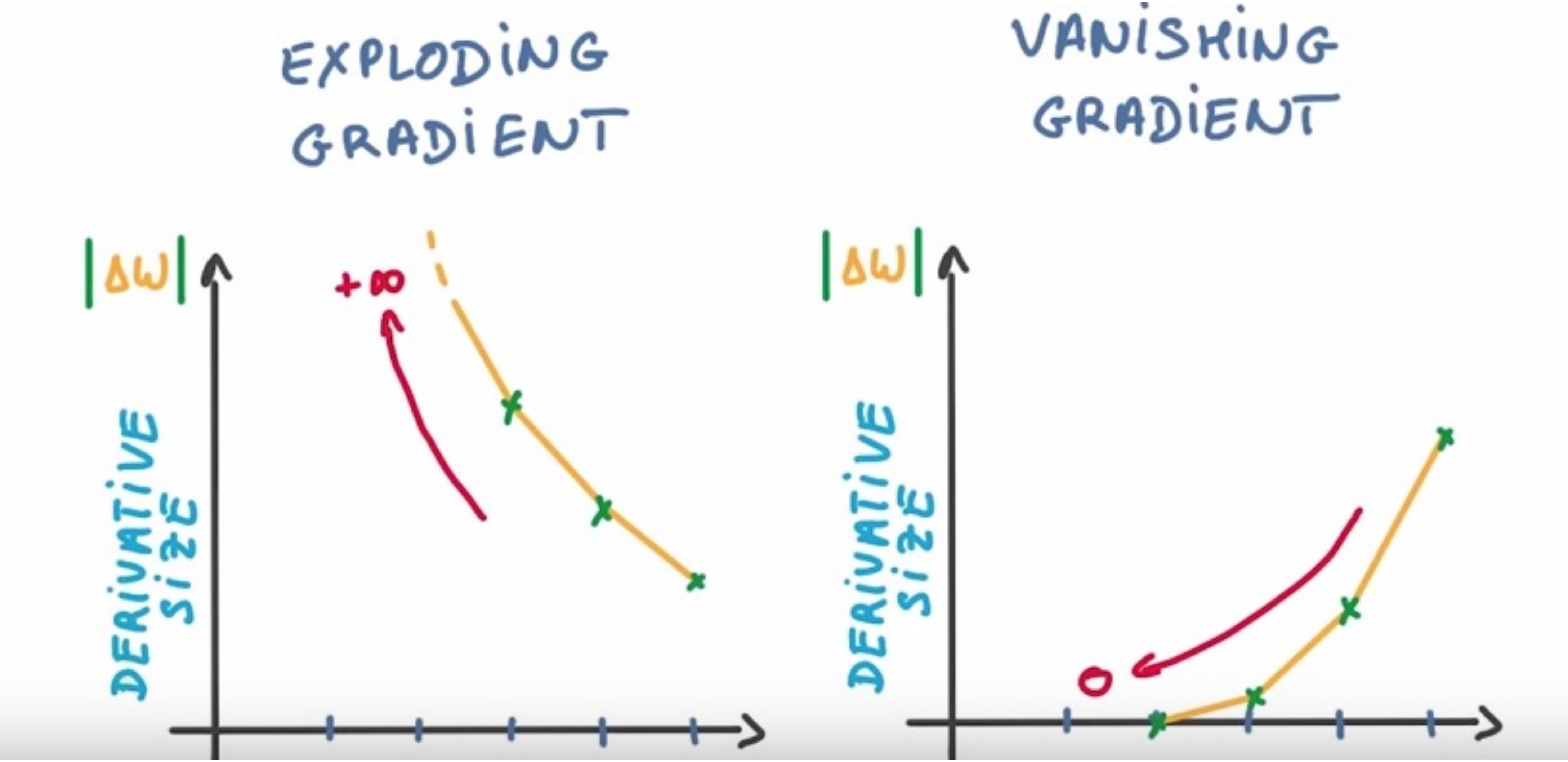


The problem: Vanishing gradients

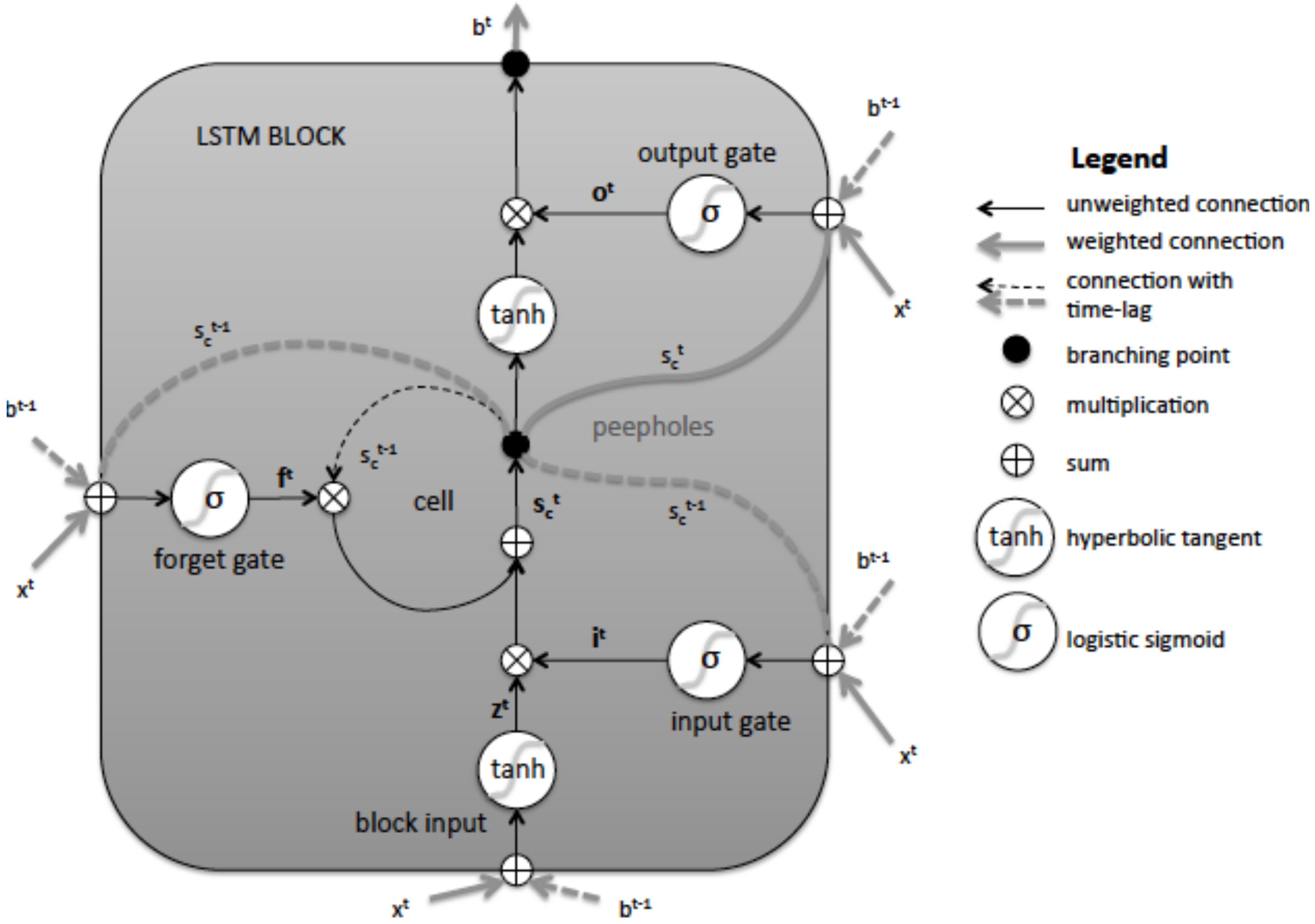
To capture this we need a long term dependency… and if in every step the gradient decay the effect is that probably we do not have influence in earlier layers.



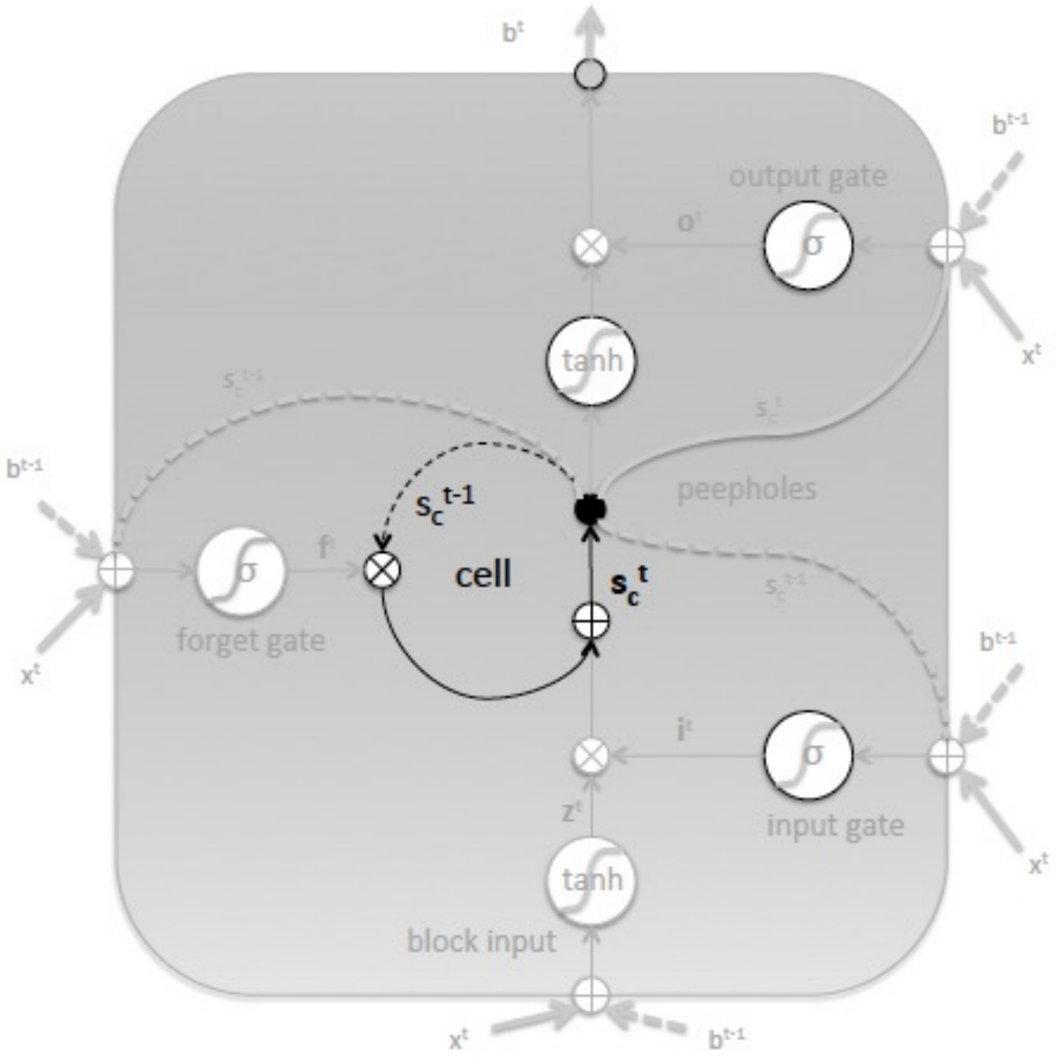
The problem: Vanishing gradients



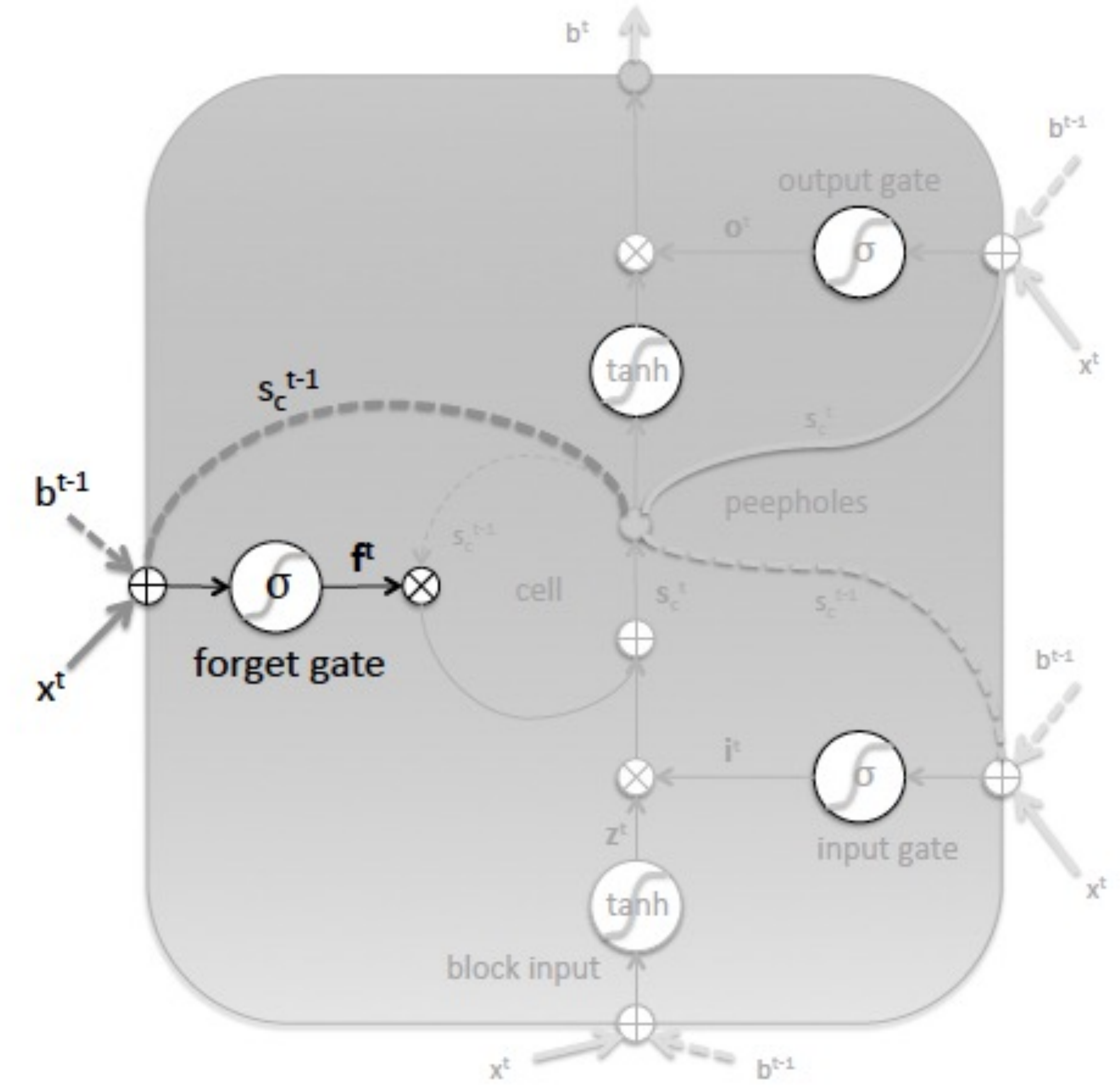
LSTM's: The unit



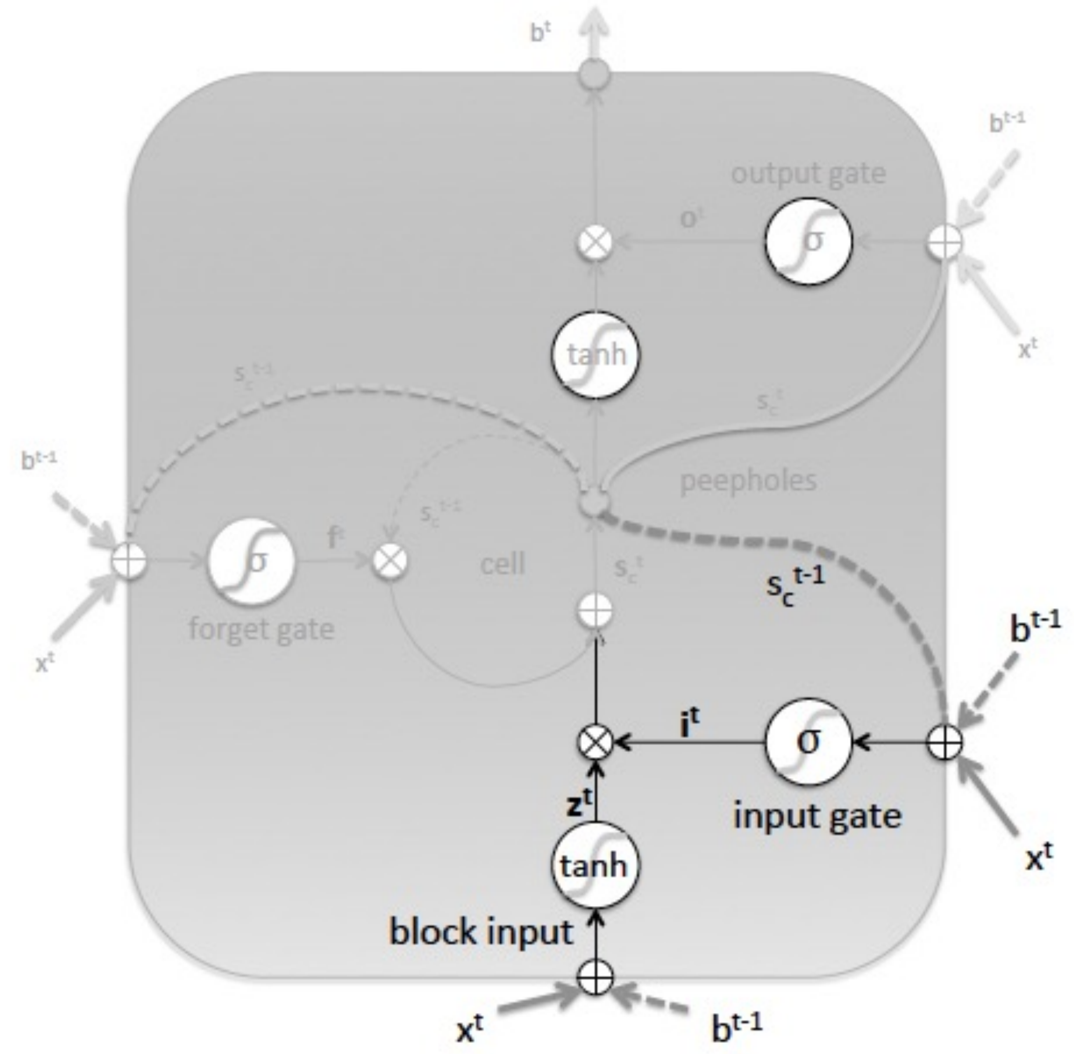
LSTM’s. The main core: Memory unit



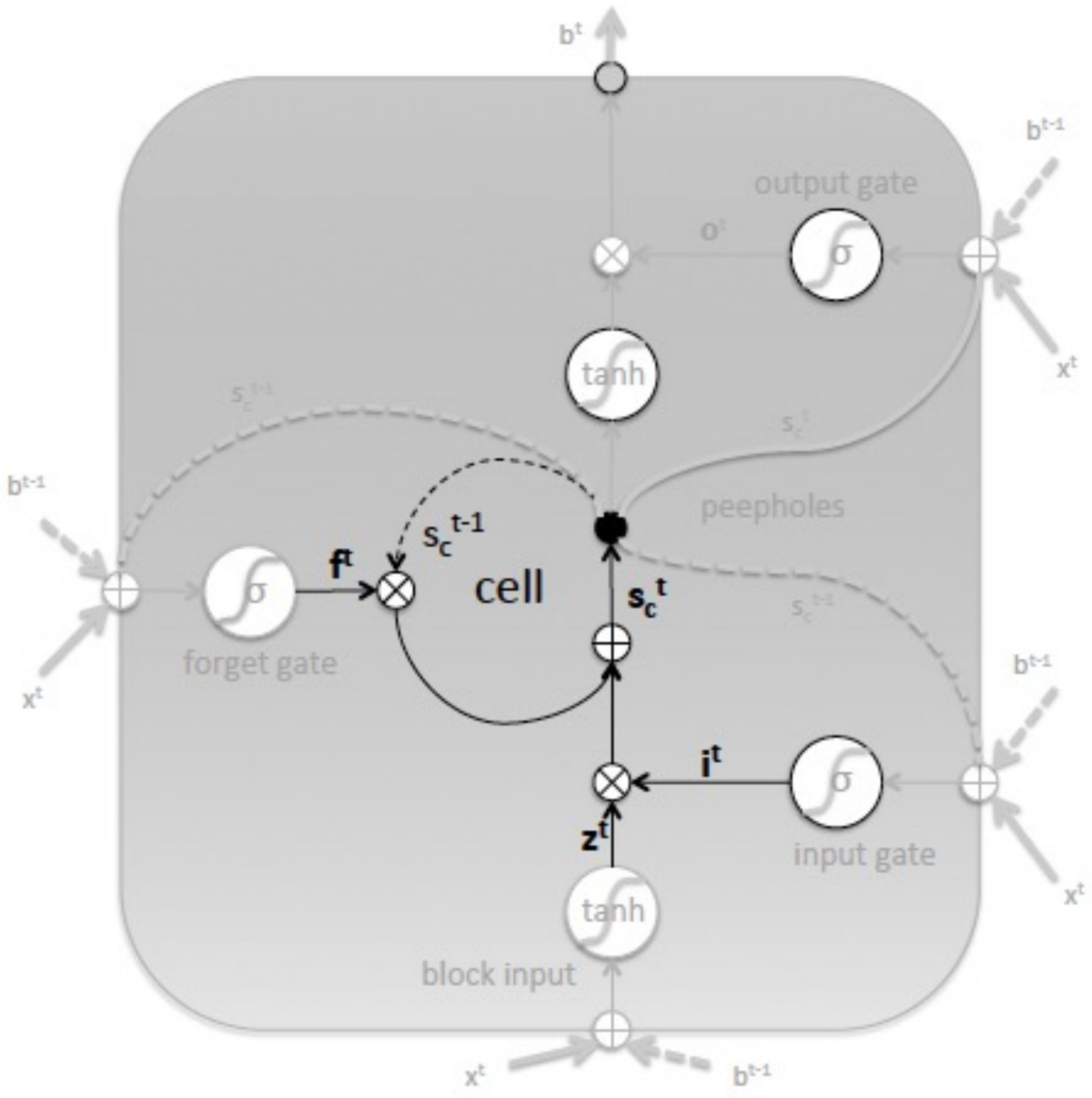
Step 1: What do we have to forget ?



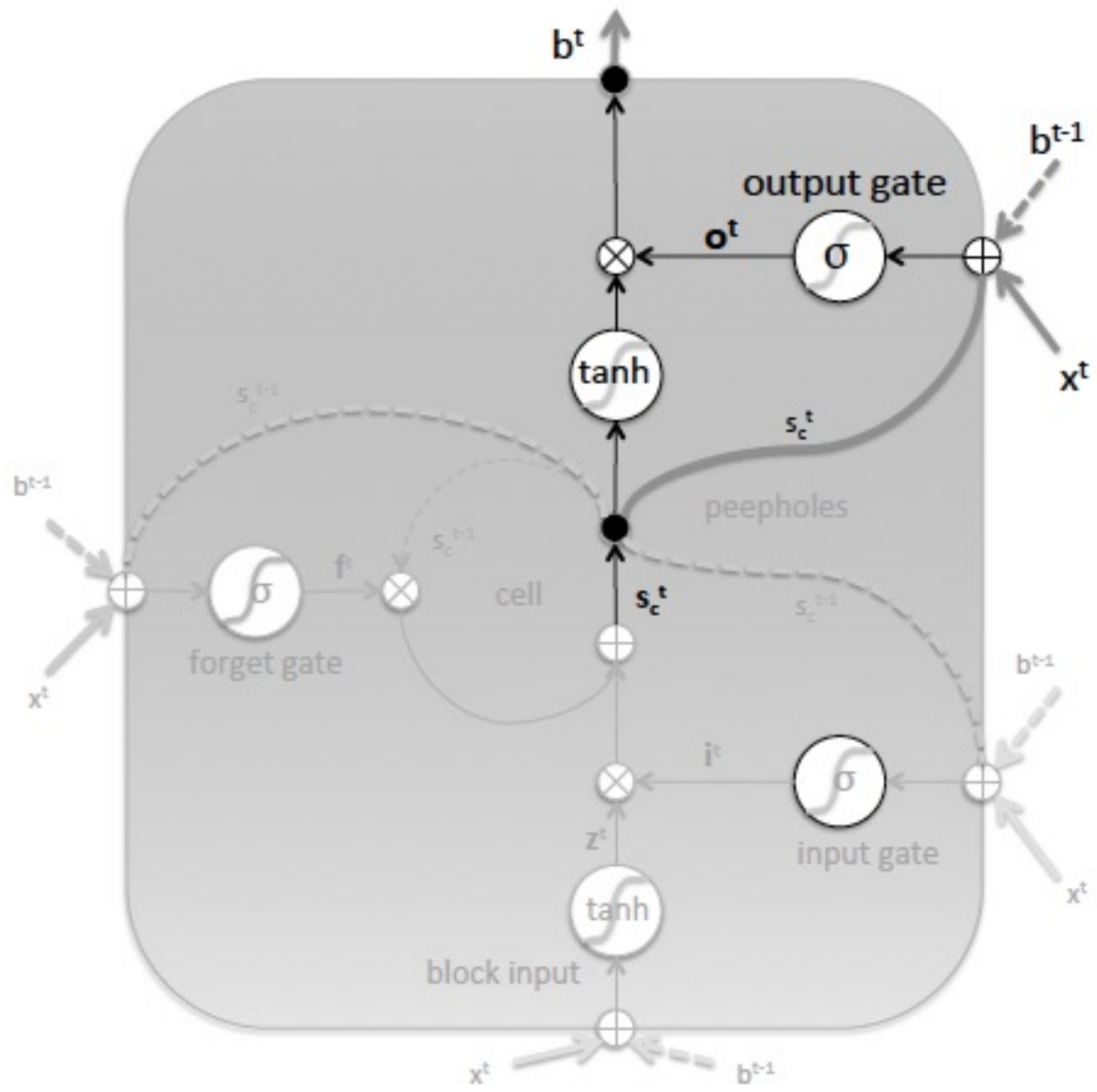
Step 2: What are we learning ?



Step 3: Updating the memory



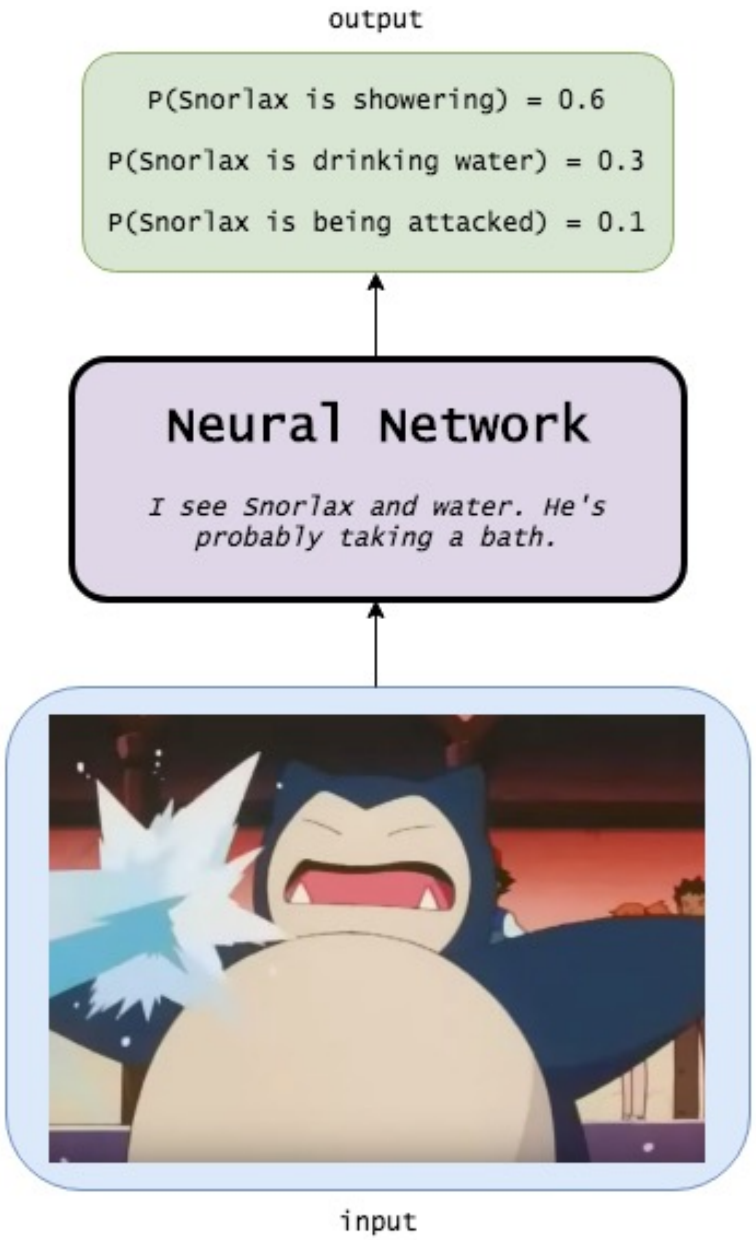
Step 4: What is our output?



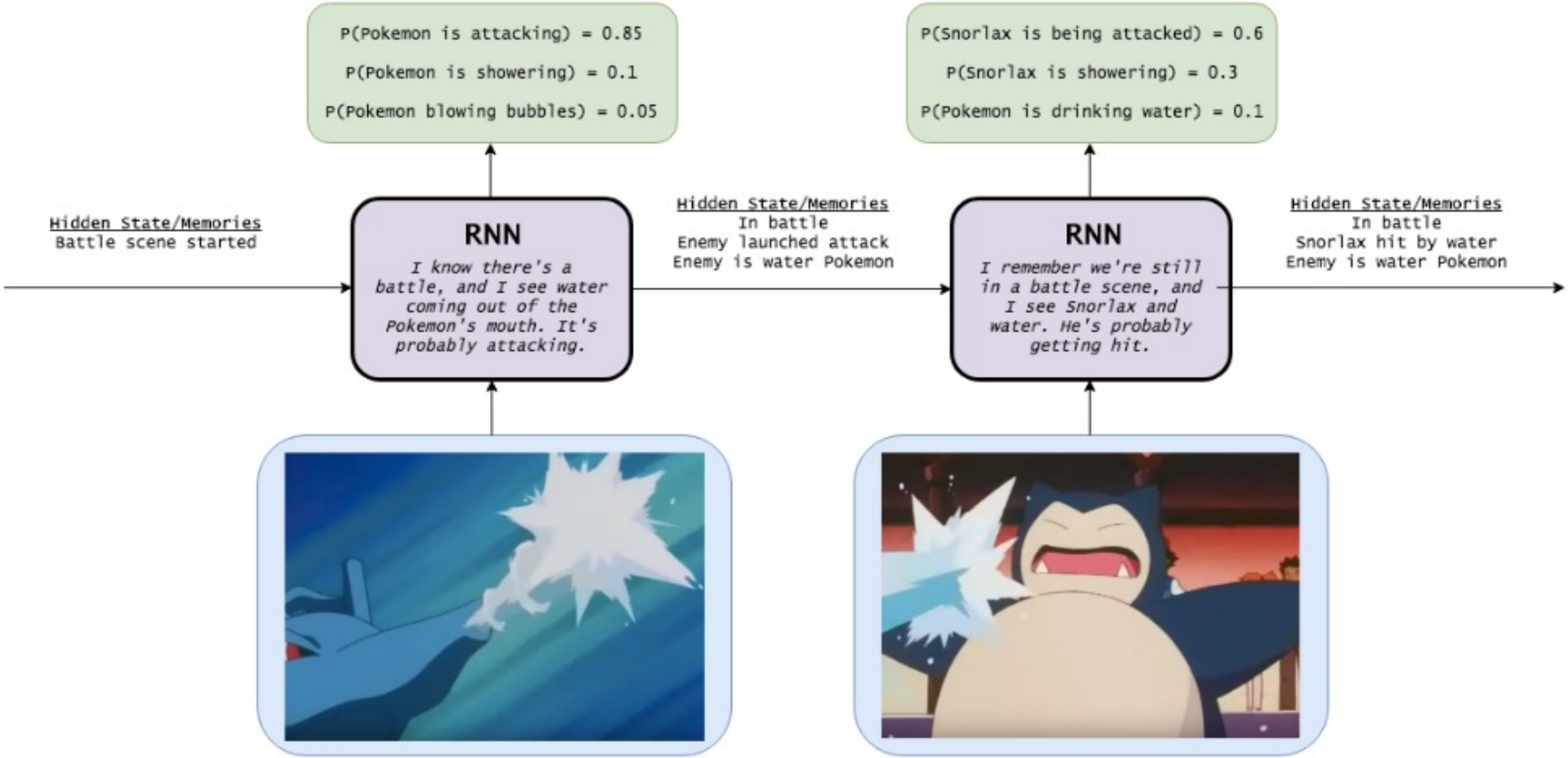
LSTM’s

https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

LSTM's: An intuitive vision



LSTM’s: An intuitive vision



LSTM's: An intuitive vision