Sapere utile

# IFOA
# Istituto Formazione Operatori Aziendali

# BIG DATA e Analisi dei Dati

Mauro Bellone,
Robotics and AI researcher

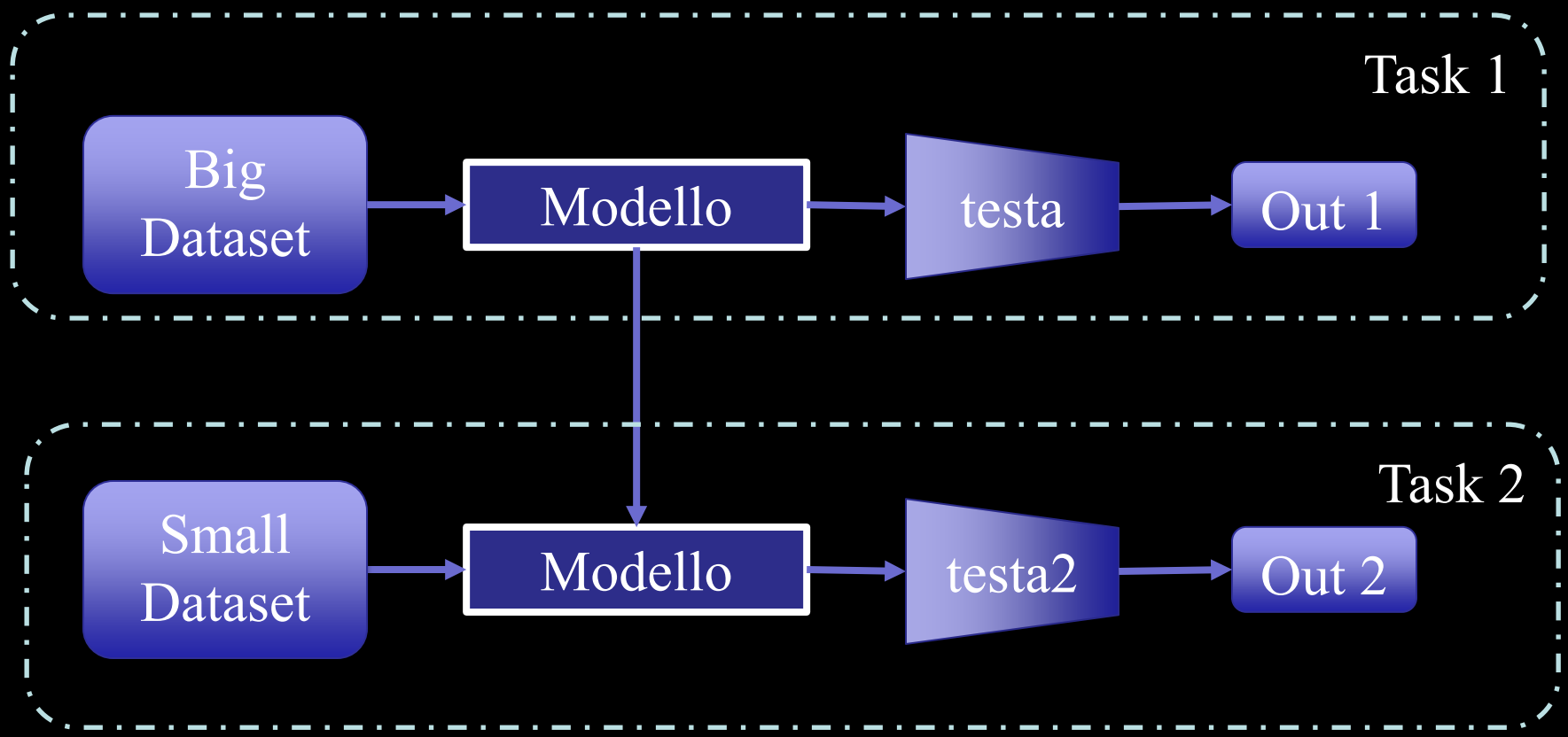
bellonemauro@gmail.com
www.maurobellone.com

## Obiettivo

- ✓ Tutorial transfer learning
- ✓ Tutorial segmentazione
- ✓ Reti ricorsive
- ✓ Reti neurali con memoria

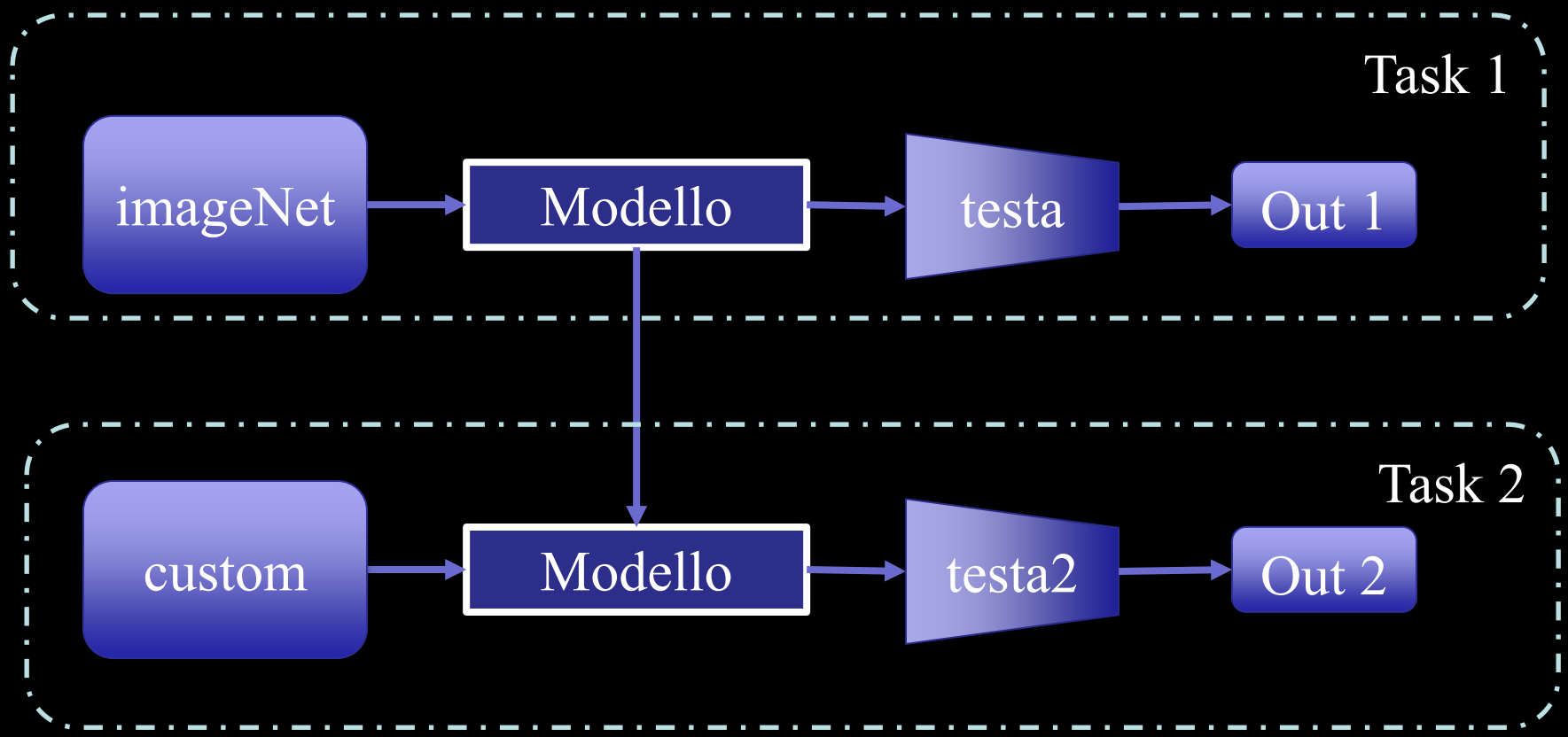
# Transfer learning

Le inizializzazioni dei pesi possono essere fatte in maniera random oppure usando un altro modello pre-addestrato su dati diversi



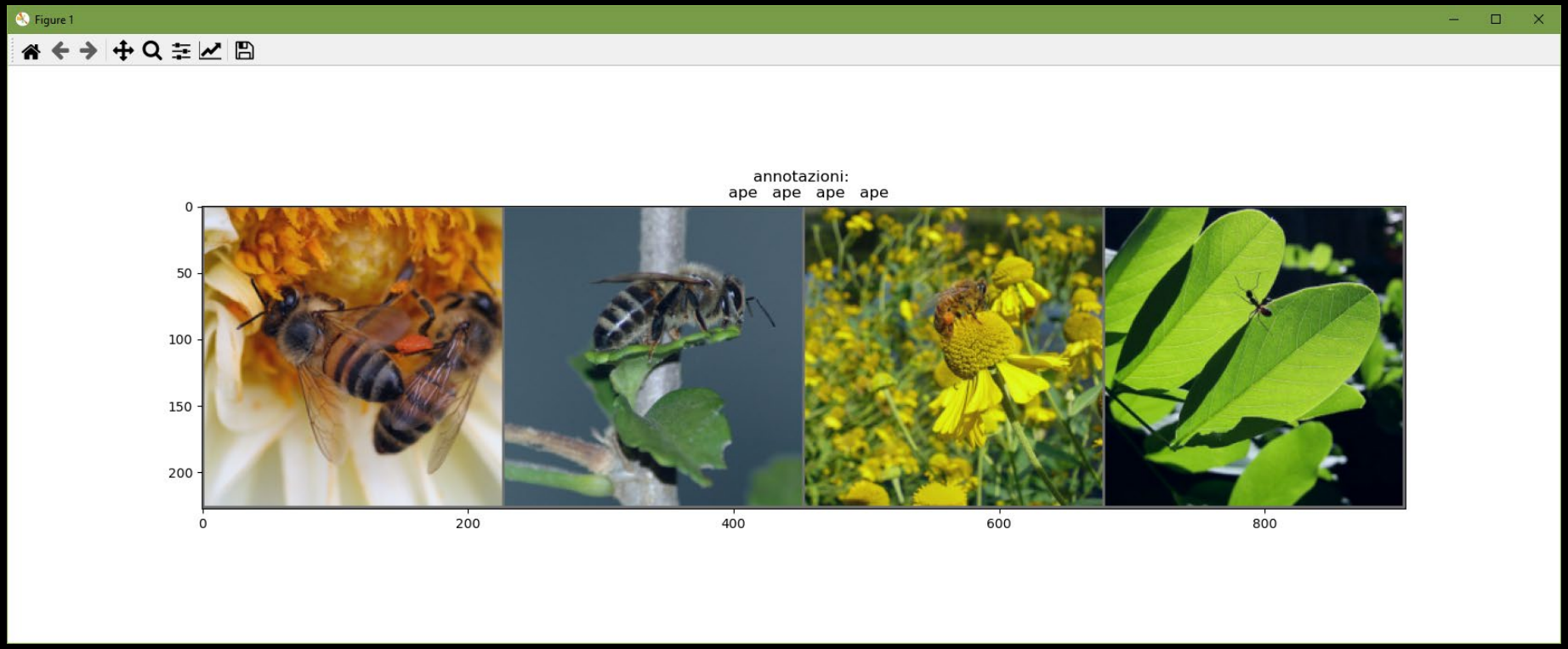
# Transfer learning

Le inizializzazioni dei pesi possono essere fatte in maniera random oppure usando un altro modello pre-addestrato su dati diversi

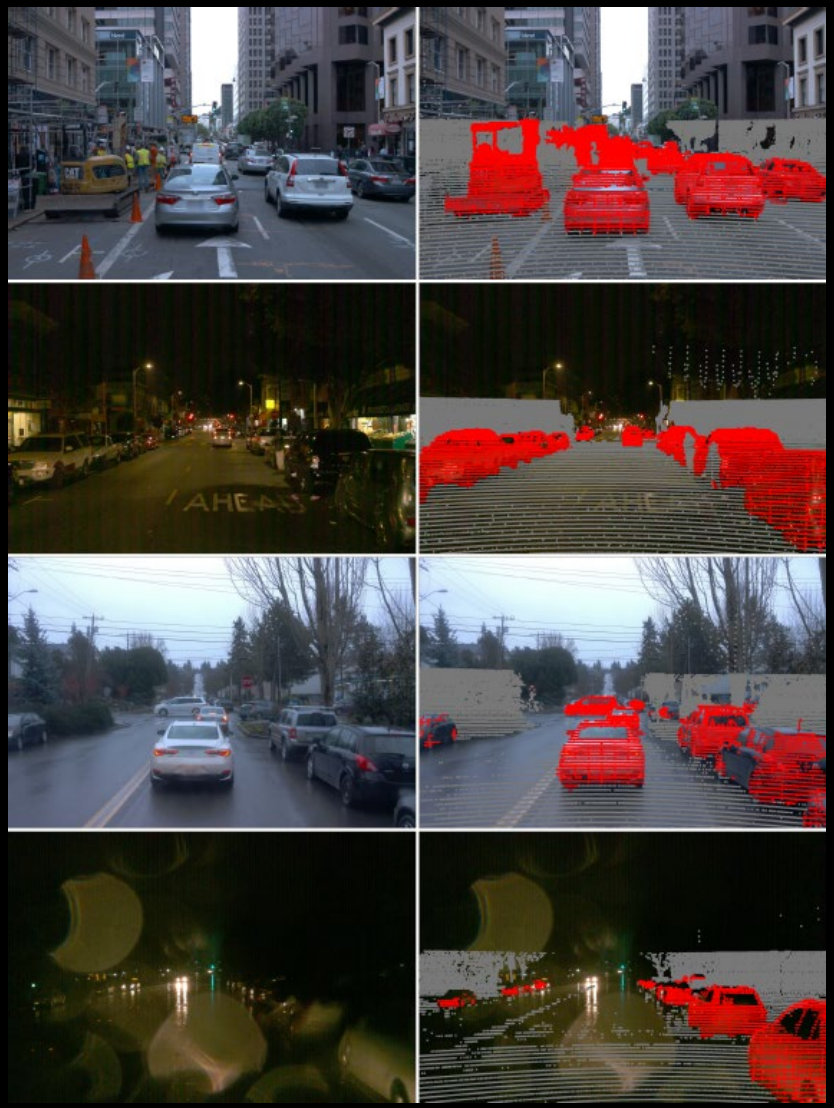


# Transfer learning – Tutorial

Usiamo resNet per classificare api e formiche in un piccolo insieme di immagini



# Segmentazione



source: L. Caltagirone, M. Bellone, L. Svensson, M. Wahde, and R. Sell - "Lidar–Camera Semi-Supervised Learning for Semantic Segmentation" Sensors, 2021 Vol. 21, issue no. 14:4813 https://doi.org/10.3390/s21144813

# Segmentazione



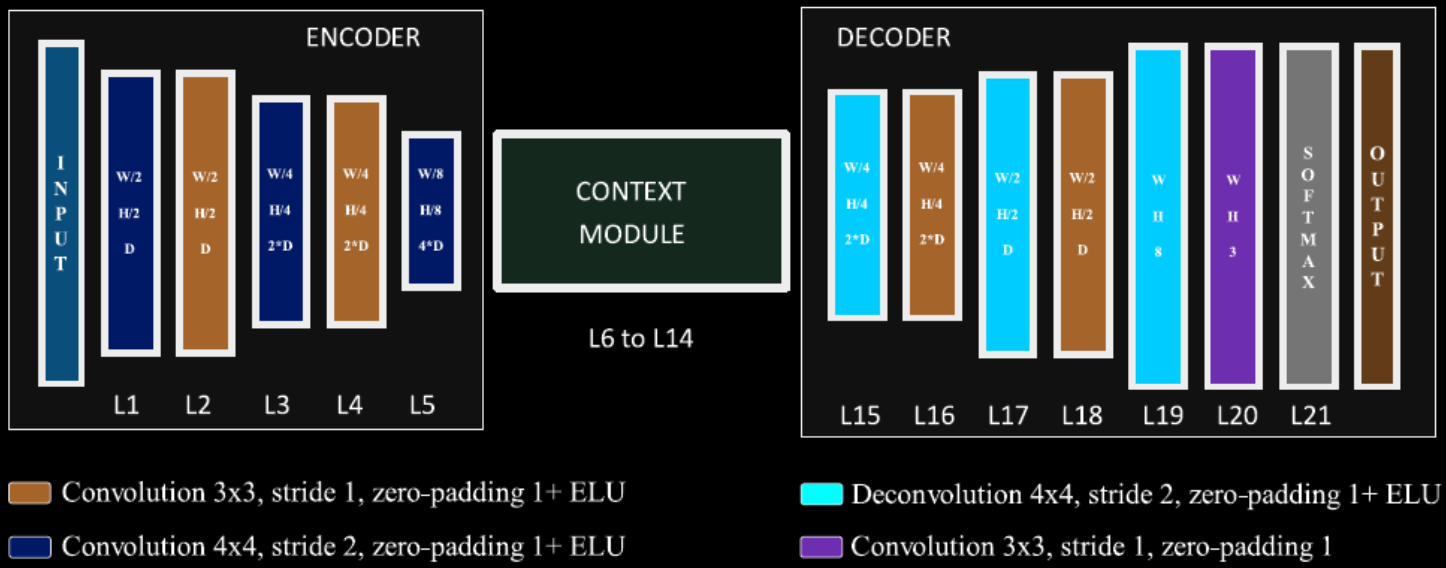
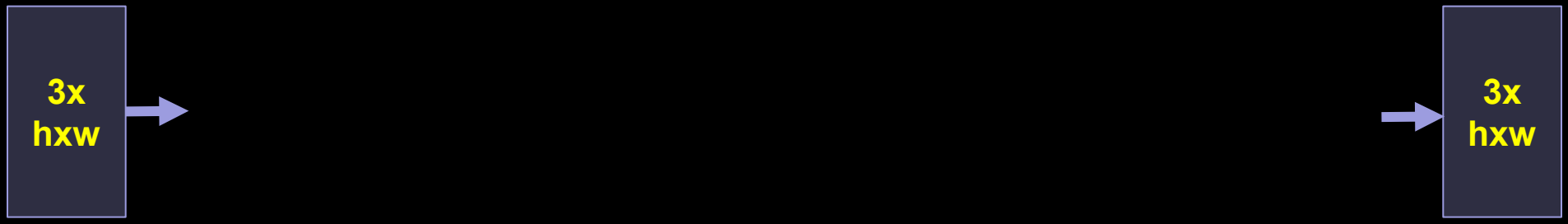



**Fig. 1.** A schematic illustration of the proposed base FCN architecture which consists of 21 layers. W represents the width, H denotes the height, and D is the number of feature maps in the first layer which was set to 32. The FCN uses the exponential linear unit (ELU) activation function after each convolutional layer. See Table 1 for details about the context module architecture.

source: Caltagirone, L., Bellone, M., Svensson, L. and Wahde, M., 2019. LIDAR–camera fusion for road detection using fully convolutional neural networks. Robotics and Autonomous Systems, 111, pp.125-131.

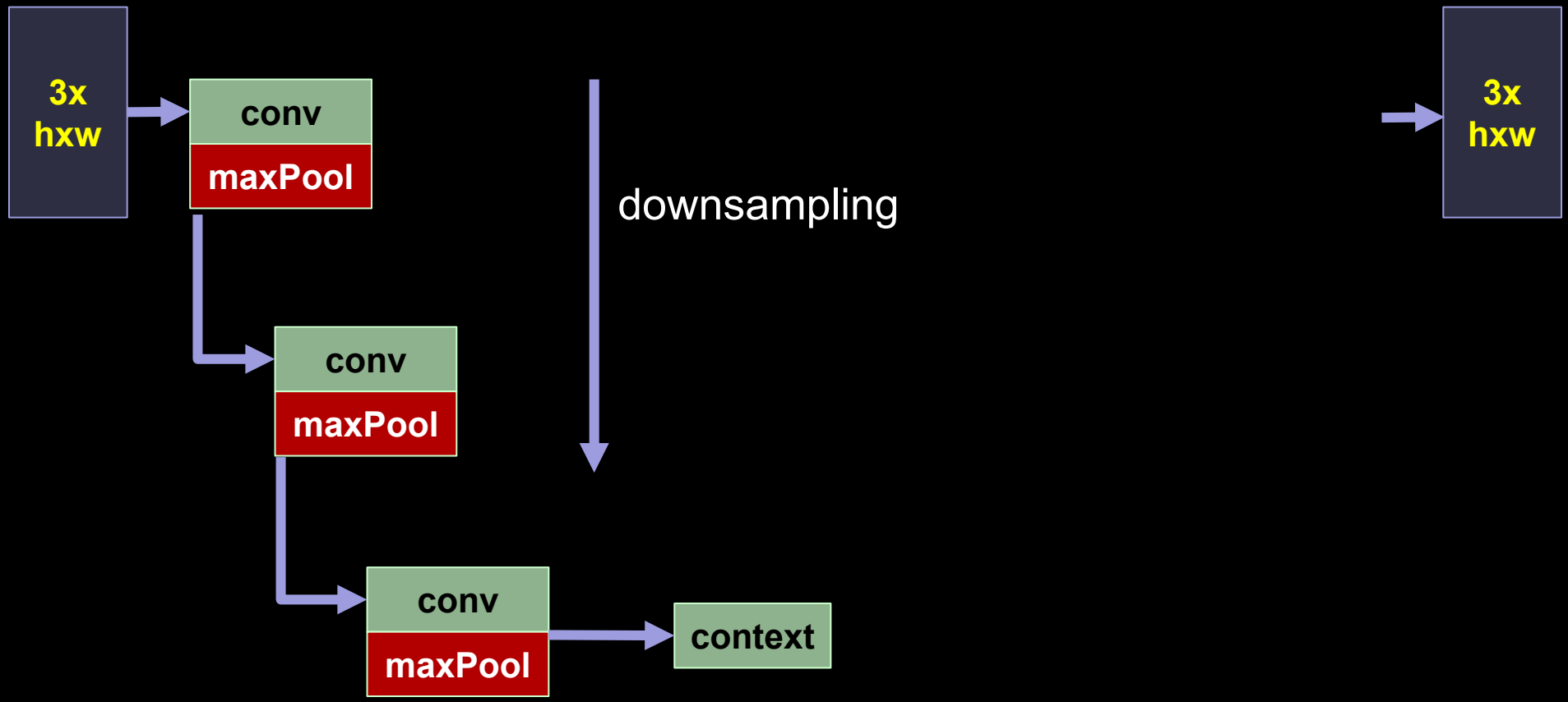
# Rete U-NET

Struttura di rete pensata con l'obiettivo di produrre in uscita una immagine della stessa risoluzione dell'immagine di partenza con l'aggiunta di alcuni elementi



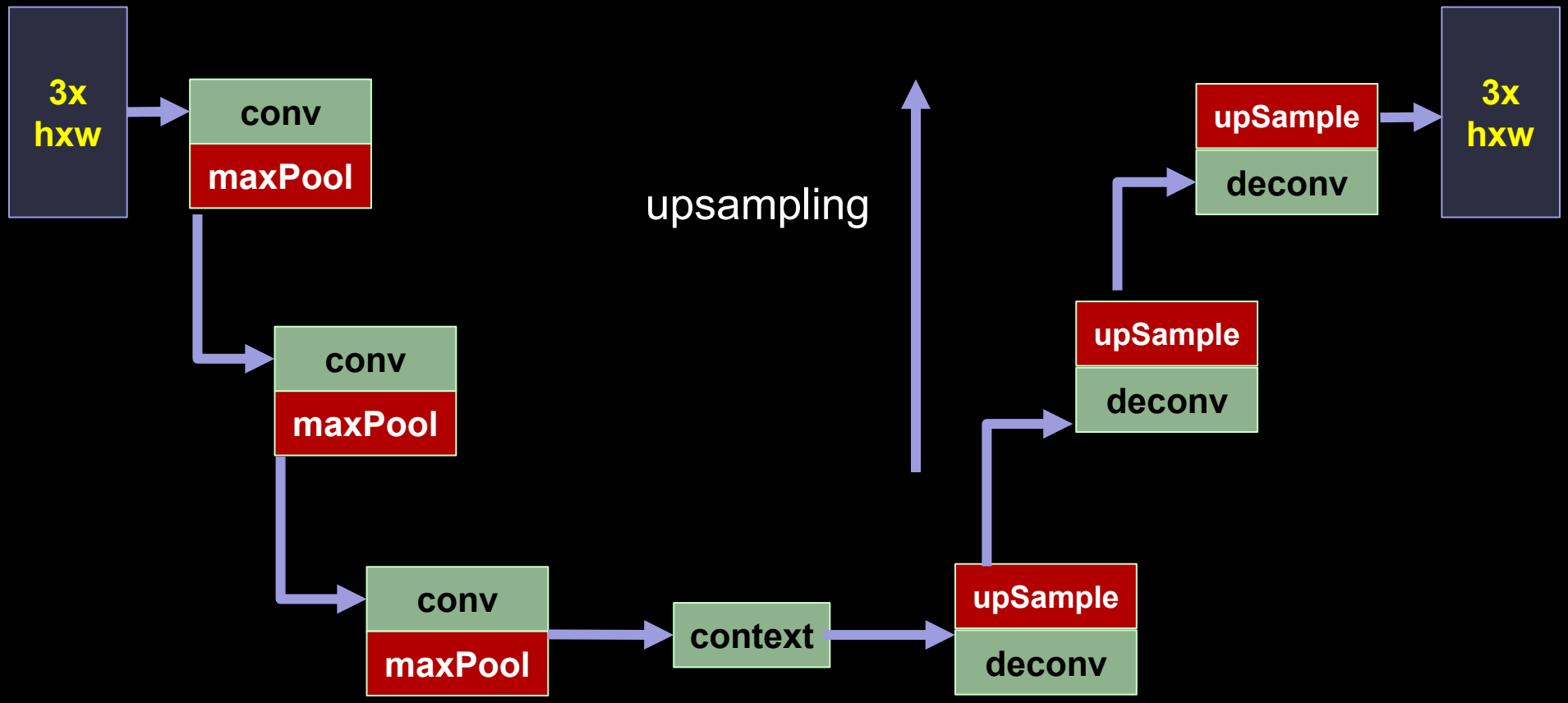
# Rete U-NET

Struttura di rete pensata con l'obiettivo di produrre in uscita una immagine della stessa risoluzione dell'immagine di partenza con l'aggiunta di alcuni elementi



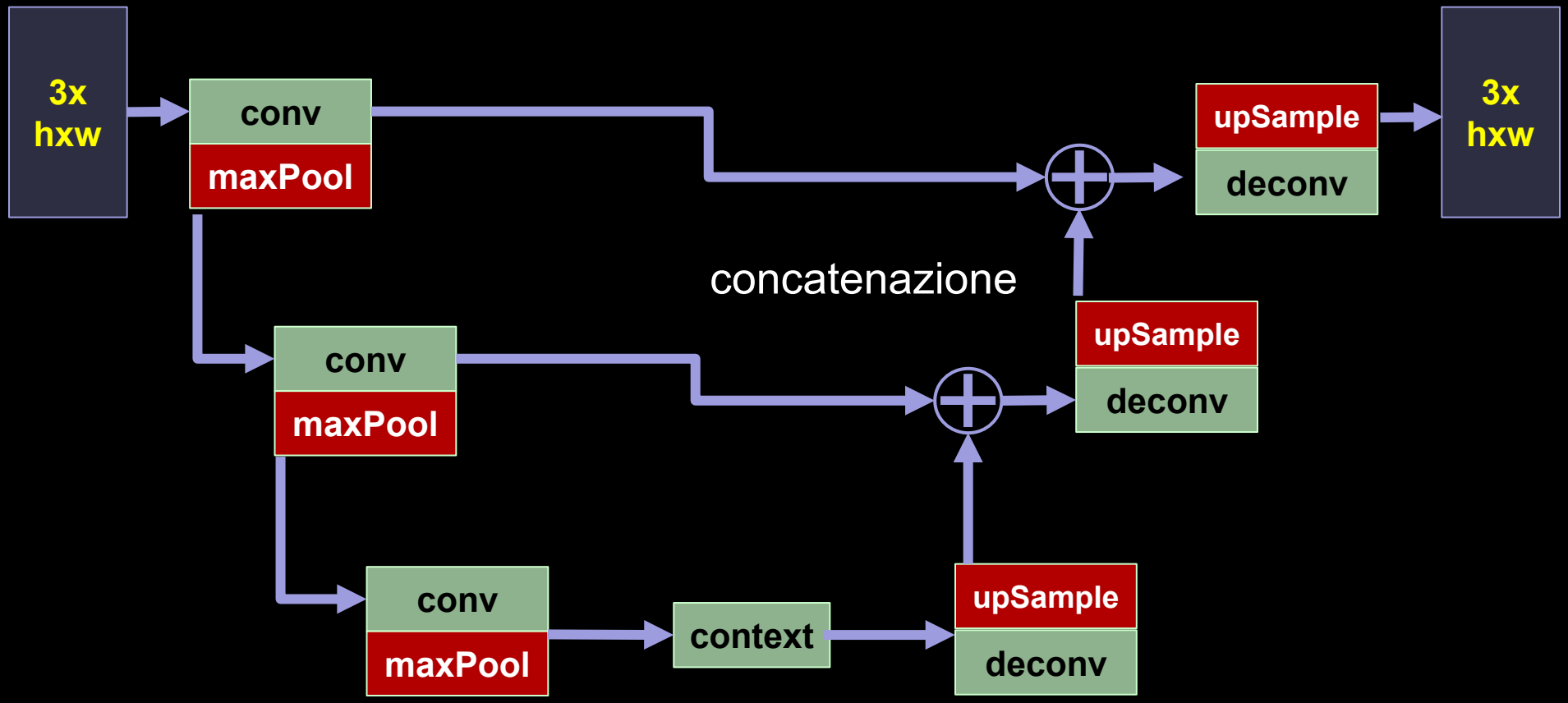
# Rete U-NET

Struttura di rete pensata con l'obiettivo di produrre in uscita una immagine della stessa risoluzione dell'immagine di partenza con l'aggiunta di alcuni elementi

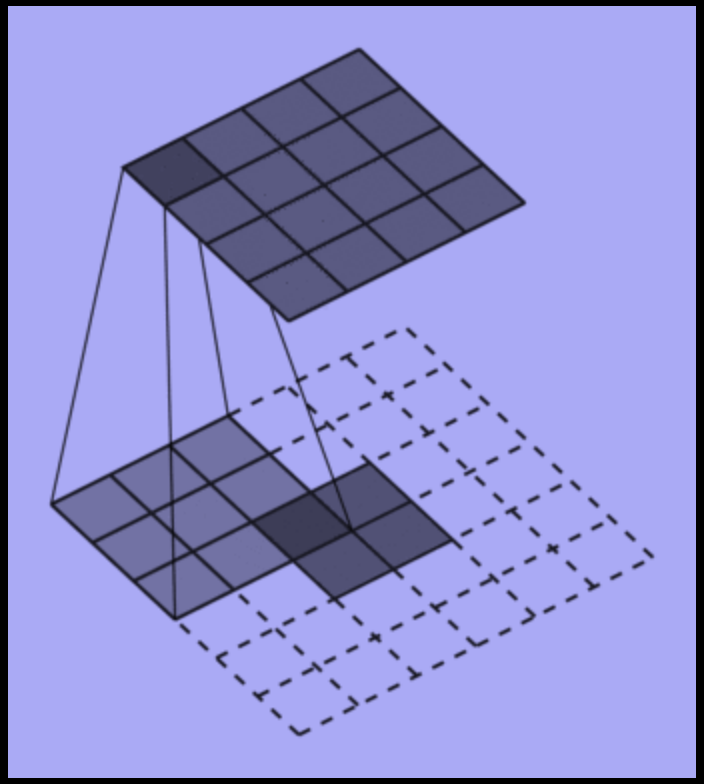


# Rete U-NET

Struttura di rete pensata con l'obiettivo di produrre in uscita una immagine della stessa risoluzione dell'immagine di partenza con l'aggiunta di alcuni elementi



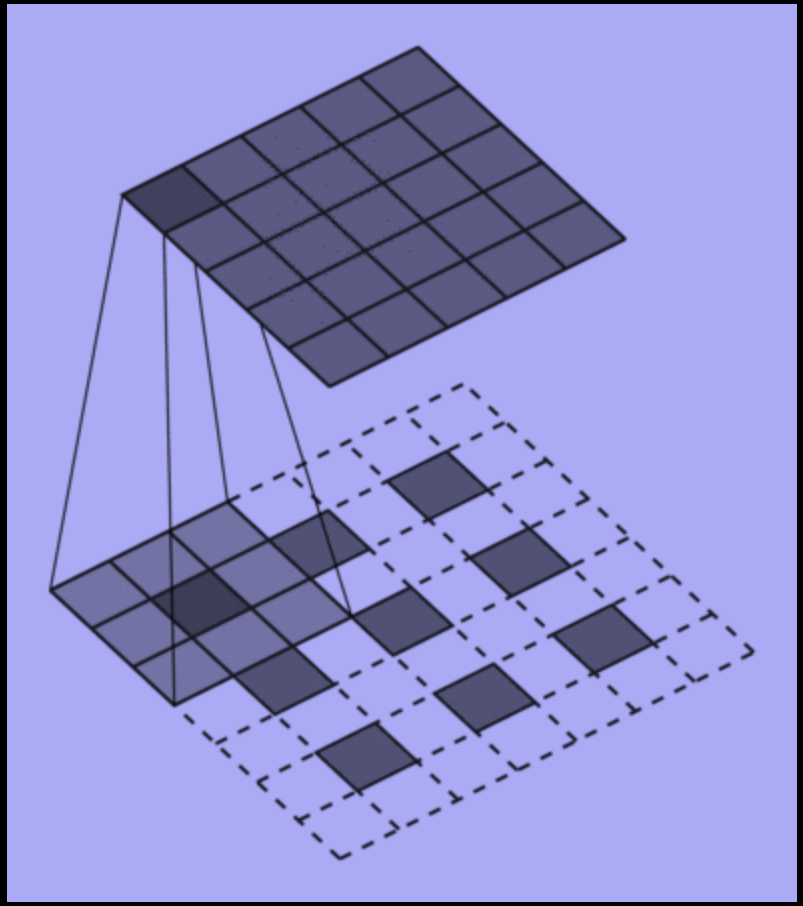
# Convoluzione trasposta



https://github.com/vdumoulin/conv_arithmetic

source: Dumoulin, V. and Visin, F., 2016. A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285.

# Convoluzione trasposta



https://github.com/vdumoulin/conv_arithmetic

source: Dumoulin, V. and Visin, F., 2016. A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285.

# Segmentazione tramite reti convoluzionali - Tutorial

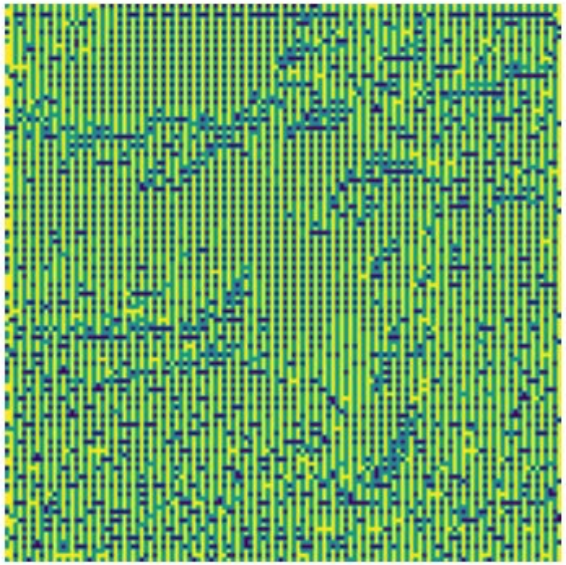

Segmentazione su dataset Oxford  https://www.robots.ox.ac.uk/~vgg/data/pets/

# Segmentazione – ep0

Immagine in input

Maschera - Ground truth

Maschera - Predizione

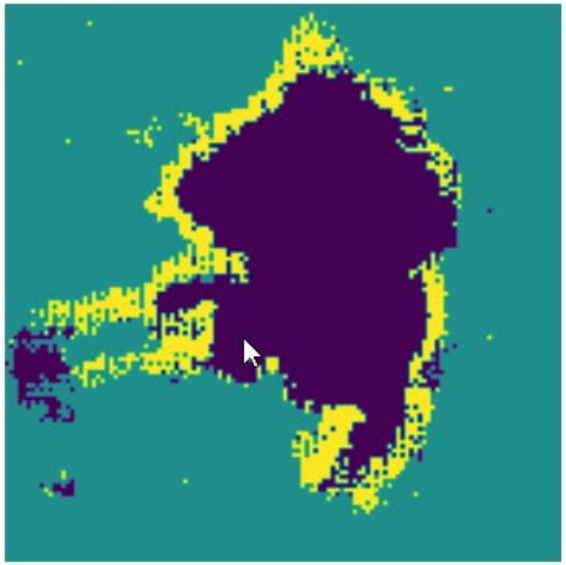


# Segmentazione – ep1

Immagine in input

Maschera - Ground truth

Maschera - Predizione

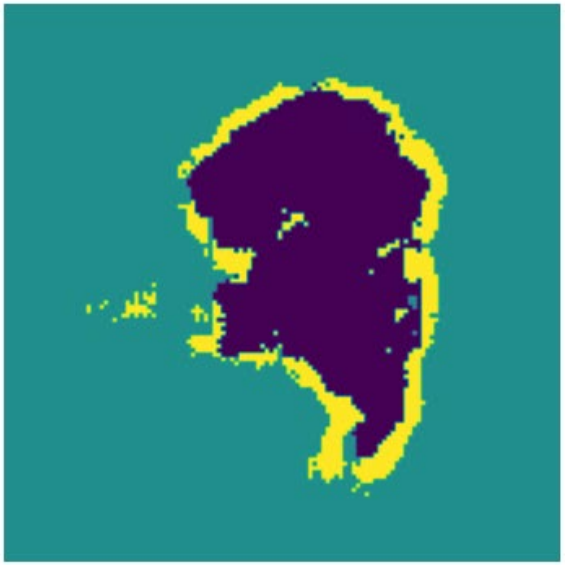


# Segmentazione – ep4

Immagine in input

Maschera - Ground truth

Maschera - Predizione

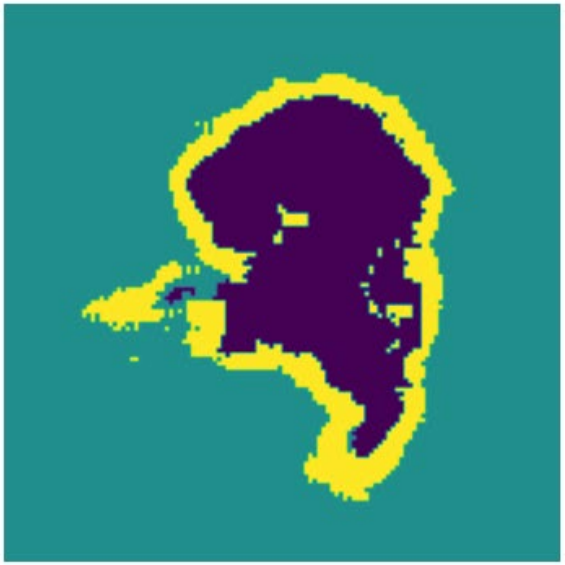


# Segmentazione – ep8

Immagine in input

Maschera - Ground truth

Maschera - Predizione

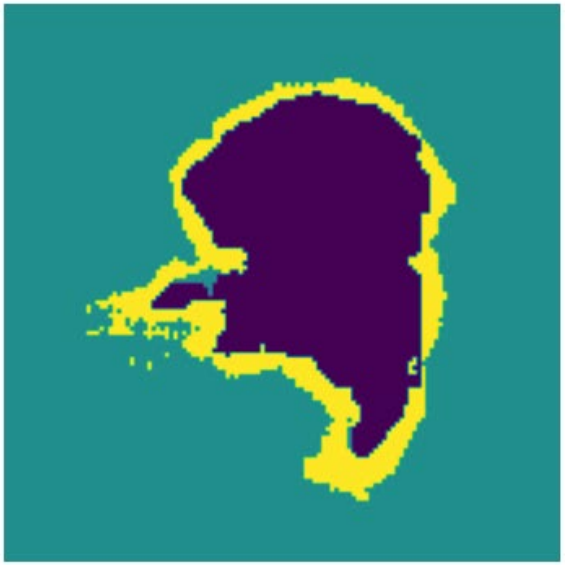


# Segmentazione – ep12

Immagine in input

Maschera - Ground truth

Maschera - Predizione

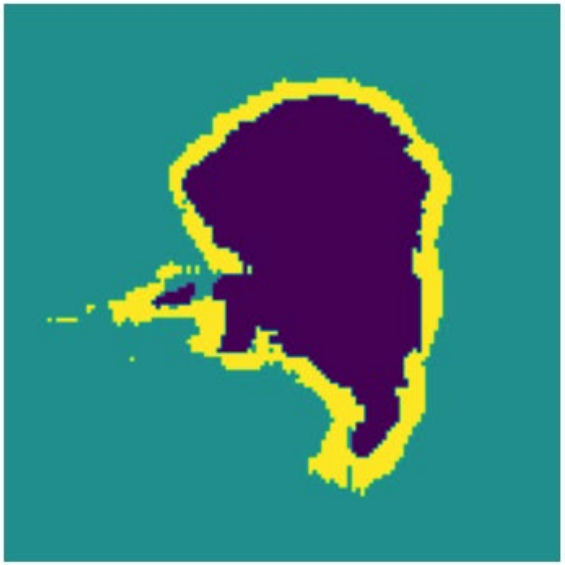


# Segmentazione – ep16

Immagine in input

Maschera - Ground truth

Maschera - Predizione

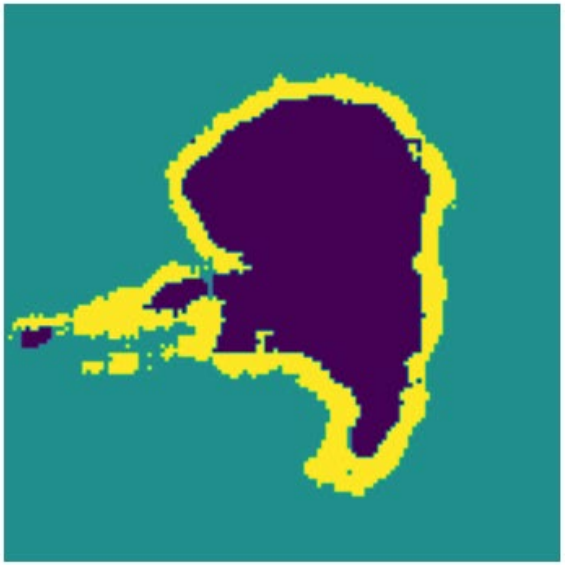


# Segmentazione – ep20

Immagine in input

Maschera - Ground truth

Maschera - Predizione

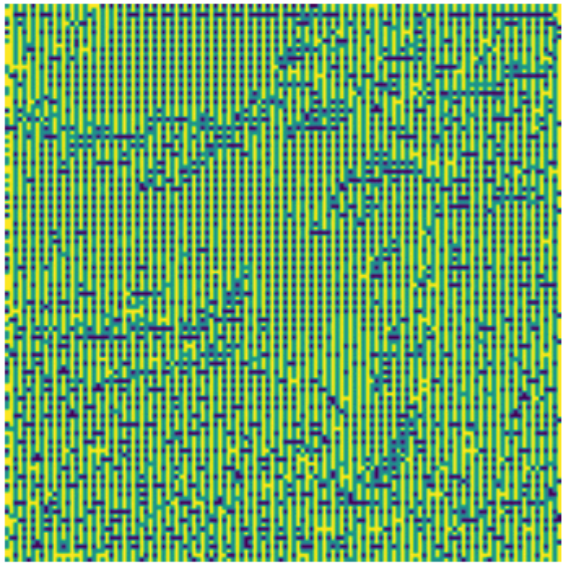


# Segmentazione

Immagine in input

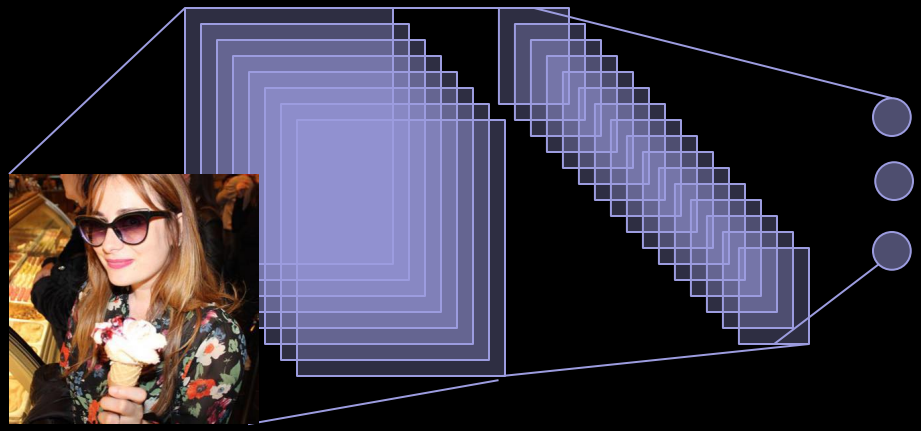
Maschera - Ground truth

Maschera - Predizione



# Autoencoders

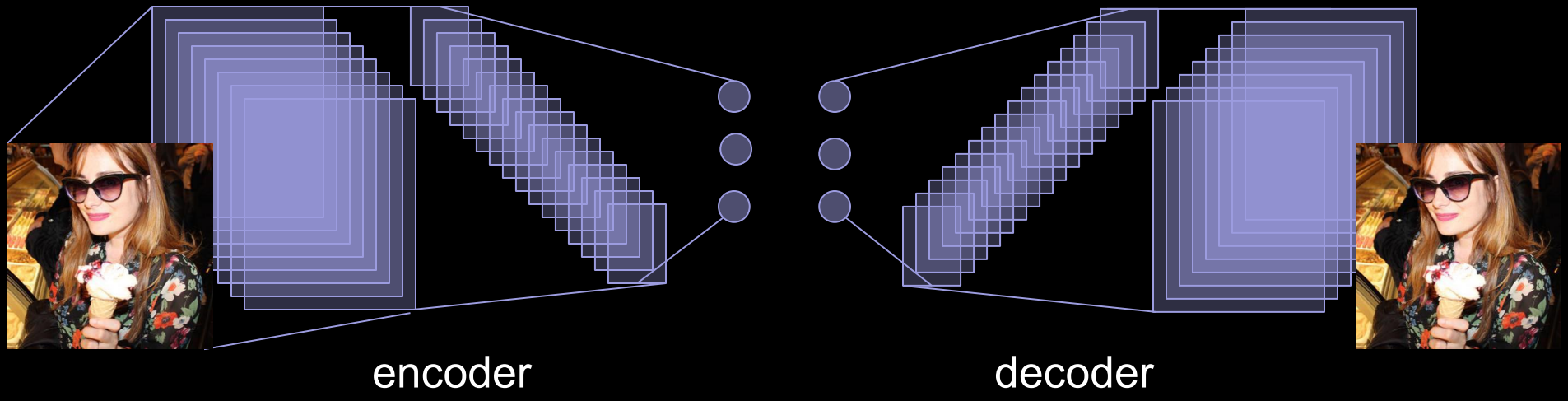
Un autoencoder è una rete neurale che prova a fare delle copie del suo input sul suo output



encoder

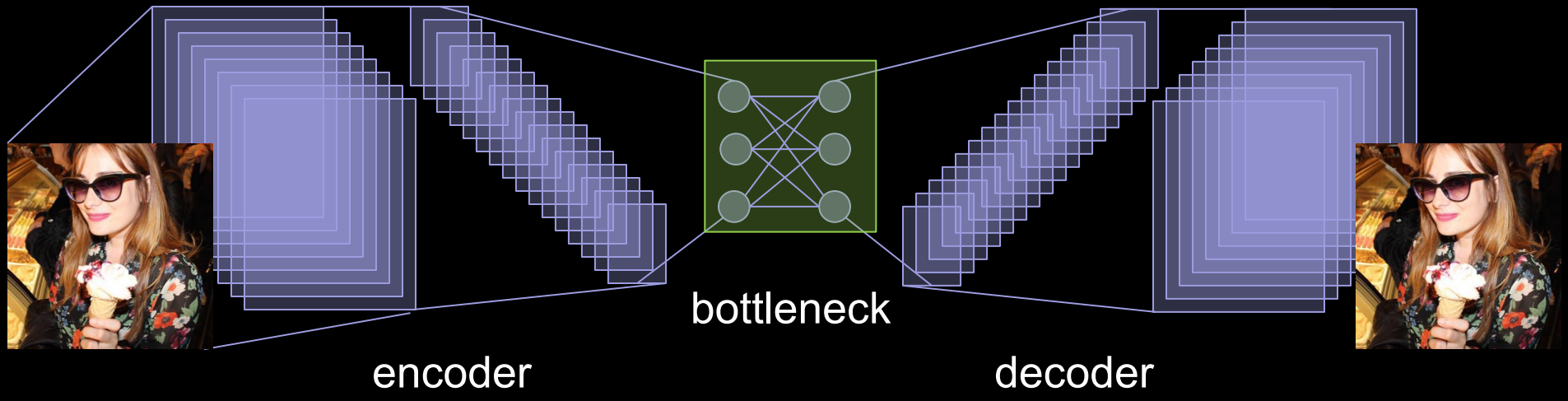
# Autoencoders

Un autoencoder è una rete neurale che prova a fare delle copie del suo input sul suo output



# Autoencoders

Un autoencoder è una rete neurale che prova a fare delle copie del suo input sul suo output



# GANs Generative adversarial networks

Si usano dei modelli generativi per creare delle immagini false e un discriminatore per individuarle

generatore

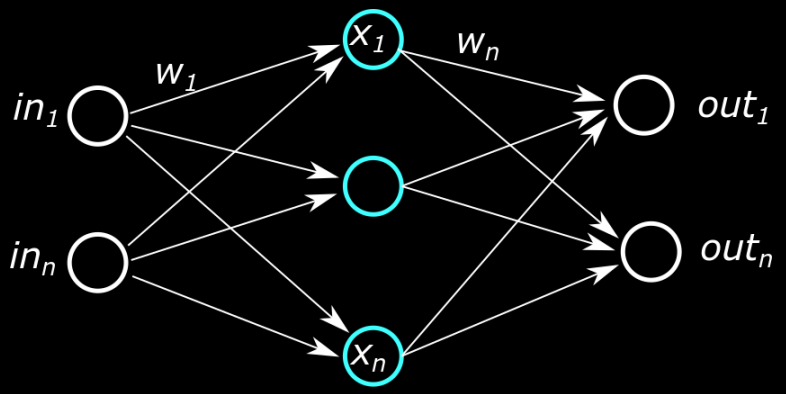
discriminatore

fake?

# Reti ricorsive

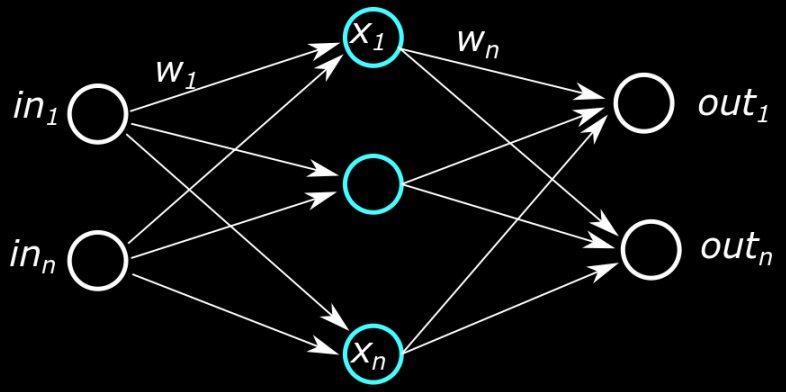
# Recap – reti neurali feed forward

Tutte le reti che abbiamo visto fino a questo momento sono delle reti in avanti o feed-forward. Non sono ammessi cicli o anelli di alcun genere.



# Recap – reti neurali feed forward

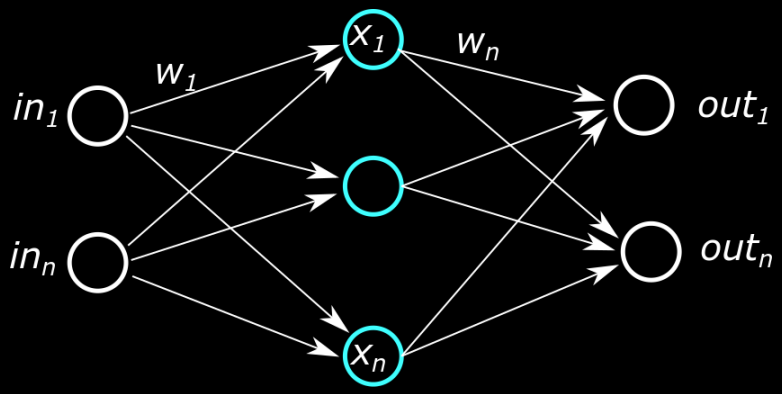
Tutte le reti che abbiamo visto fino a questo momento sono delle reti in avanti o feed-forward. Non sono ammessi cicli o anelli di alcun genere.

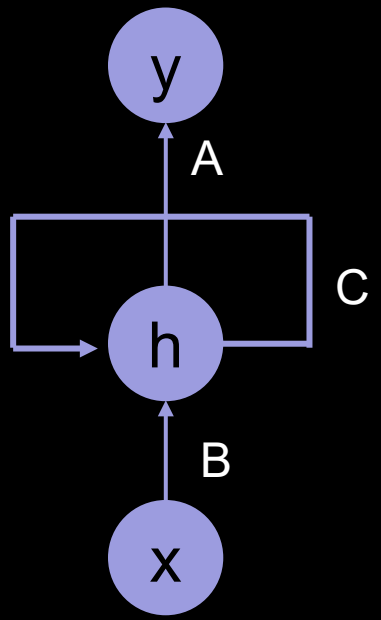



- ✓ Tutte le decisioni sono basate sull'input corrente
- ✓ No serie temporali
- ✓ Non c'è memoria del passato
- ✓ Non c'è scopo futuro

# Recap – reti neurali feed forward

Tutte le reti che abbiamo visto fino a questo momento sono delle reti in avanti o feed-forward. Non sono ammessi cicli o anelli di alcun genere.

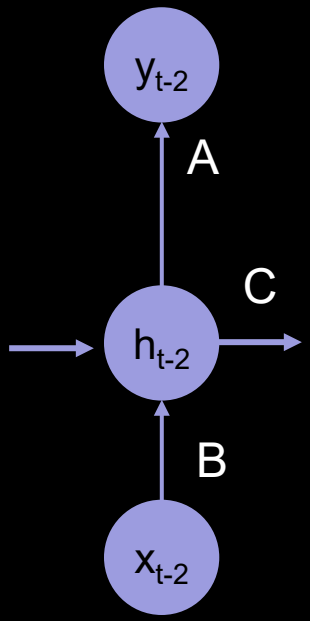


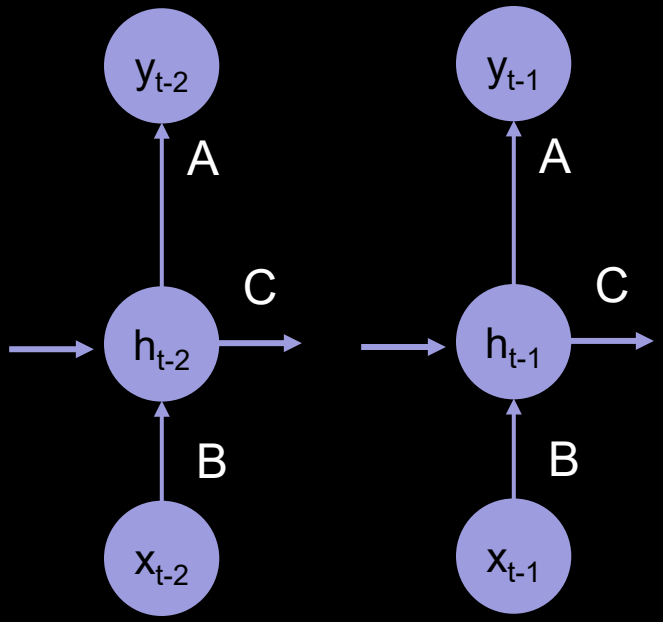
## Task con reti ricorsive

- ✓ Funzioni di autocompletamento
- ✓ Funzioni di traduzione
- ✓ Riconoscimento vocale
- ✓ Natural language processing
- ✓ Image captioning
- ✓ Predizione delle serie temporali (azioni)

# Reti ricorsive



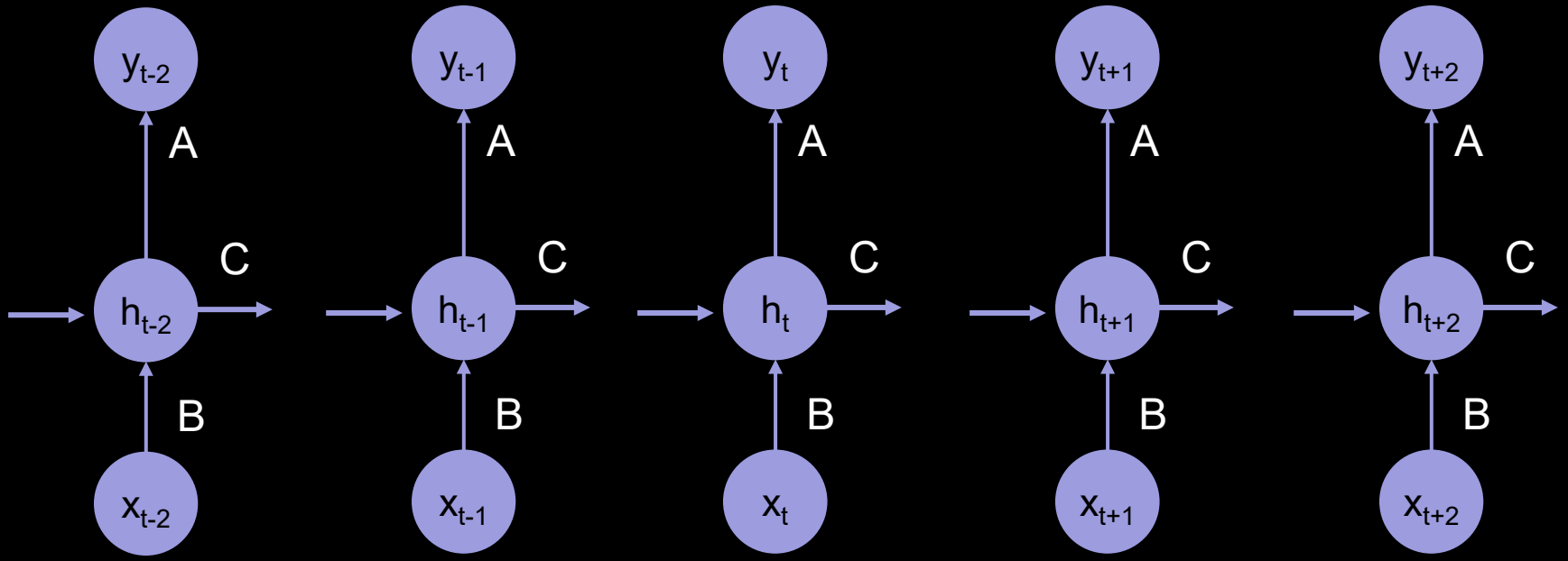
# Reti ricorsive



# Reti ricorsive

$$h_t = f_c(h_{t-1}, x_t)$$

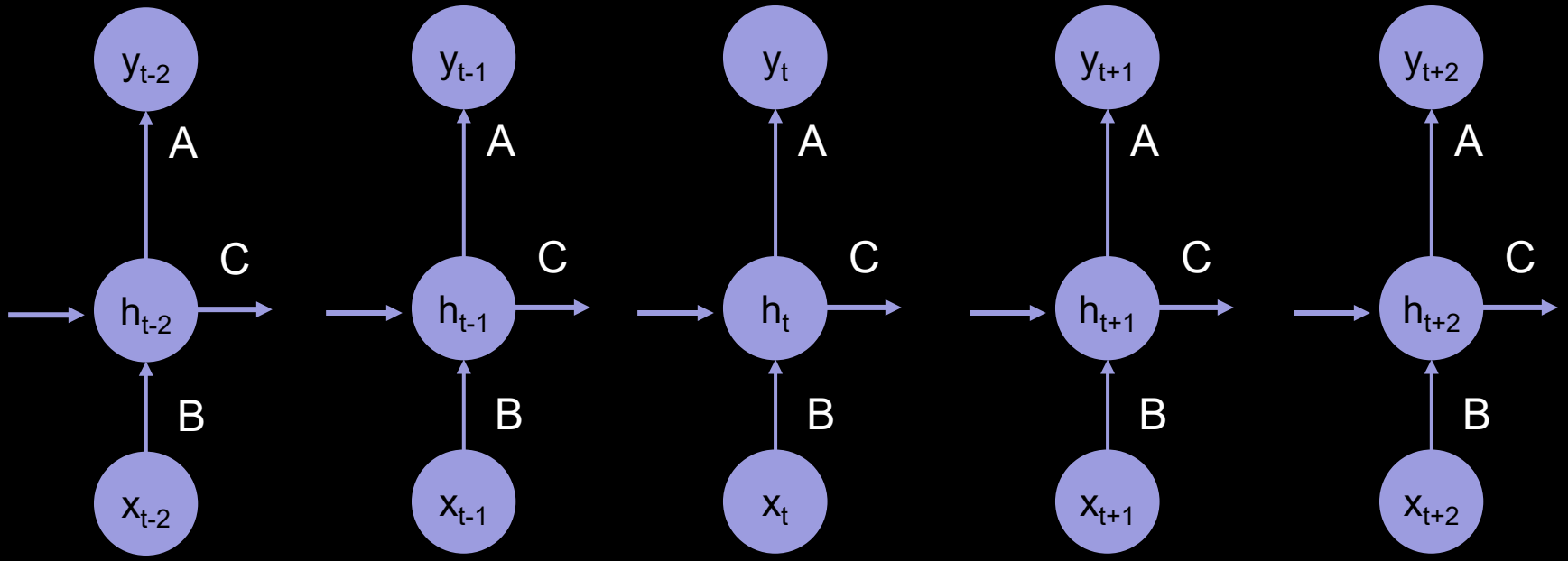
$h_t$ = stato successivo
$f_c$ = funzione di stato
$h_{t-1}$ = stato precedente
$x_t$ = input corrente



# Reti ricorsive

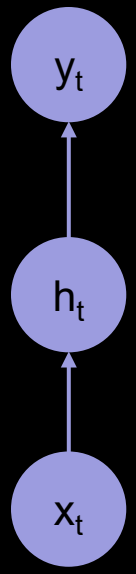
$$h_t = f_c(h_{t-1}, x_t)$$

$h_t$ = stato successivo
$f_c$ = funzione di stato
$h_{t-1}$ = stato precedente
$x_t$ = input corrente



# Tipologie di reti ricorsive

- ✓ **singolo ingresso – singola uscita**



# Tipologie di reti ricorsive

- ✓ singolo ingresso – singola uscita
- ✓ **singolo ingresso – uscita multipla**

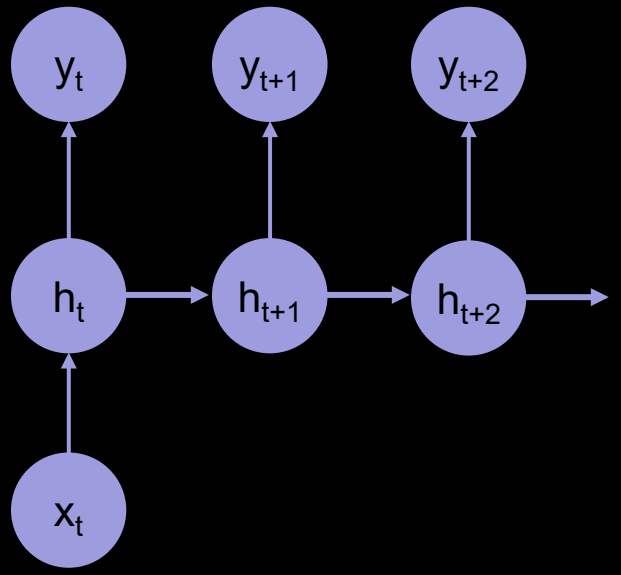
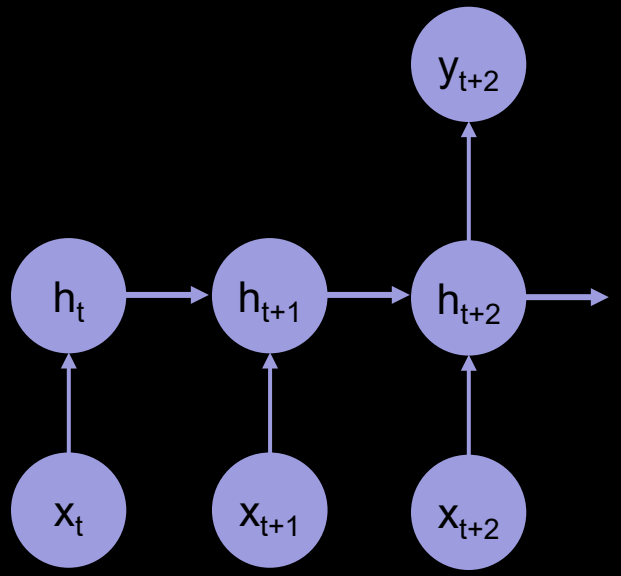



Image capturing

# Tipologie di reti ricorsive

- ✓ singolo ingresso – singola uscita
- ✓ singolo ingresso – uscita multipla
- ✓ **multiplo ingresso – singola uscita**

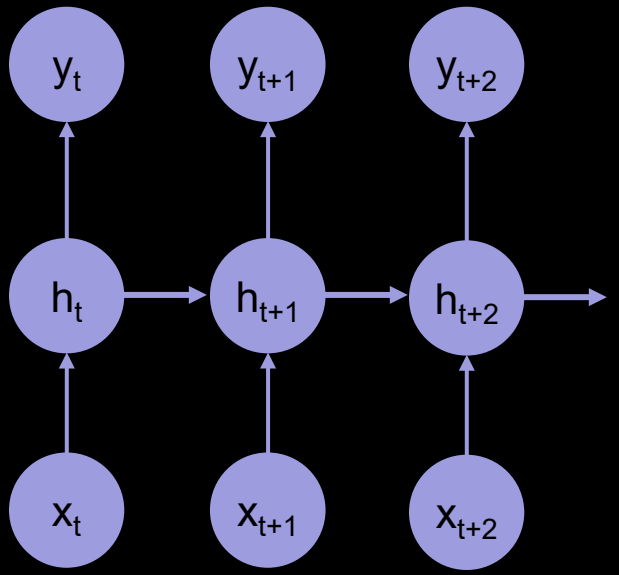



sentiment analysis – una data frase può essere classificata come espressione di un sentiment positive o negativo

espressione dei sentimenti come positivo o negative

# Tipologie di reti ricorsive

- ✓ singolo ingresso – singola uscita
- ✓ singolo ingresso – uscita multipla
- ✓ multiplo ingresso – singola uscita
- ✓ **multipli ingressi – multiple uscite**




Traduzione testo

## Vanishing and exploding gradient

L’addestramento di una RNN può generare dei gradienti che possono essere molto grandi o molto piccoli, ciò rende il training molto complesso.

Vanishing gradient → perdita di informazione nel tempo

Exploding gradient → saturazione di tutti i neuroni

# Vanishing and exploding gradient

L'addestramento di una RNN può generare dei gradienti che possono essere molto grandi o molto piccoli, ciò rende il training molto complesso.

Vanishing gradient ➔ perdita di informazione nel tempo

Exploding gradient ➔ saturazione di tutti i neuroni

Soluzioni:

- ✓ Inizializzazione dei pesi accurate
- ✓ Troncaggio della backpropagation
- ✓ Scelta delle funzioni di attivazione
- ✓ LSTM – long short-term memory

## Dipendenze di lungo periodo

Uno dei maggiori problemi è rappresentato da stati che hanno una forte dipendenza dallo stato in *k* passi.

Il caso tipico è proprio la comprensione del contesto nel discorso che può dipendere fortemente da quanto detto precedentemente, delle volte persino minuti o ore prima

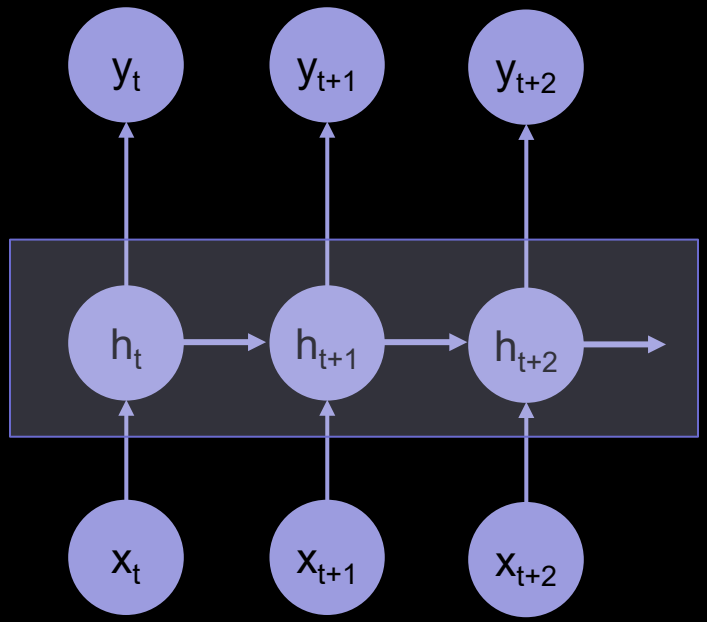
# Long short-term memory networks

Le LSTM sono dei tipi speciali di reti neurali n grado di catturare dipendenze tipicamente espresse in molti passi

# Long short-term memory networks

Le LSTM sono dei tipi speciali di reti neurali n grado di catturare dipendenze tipicamente espresse in molti passi

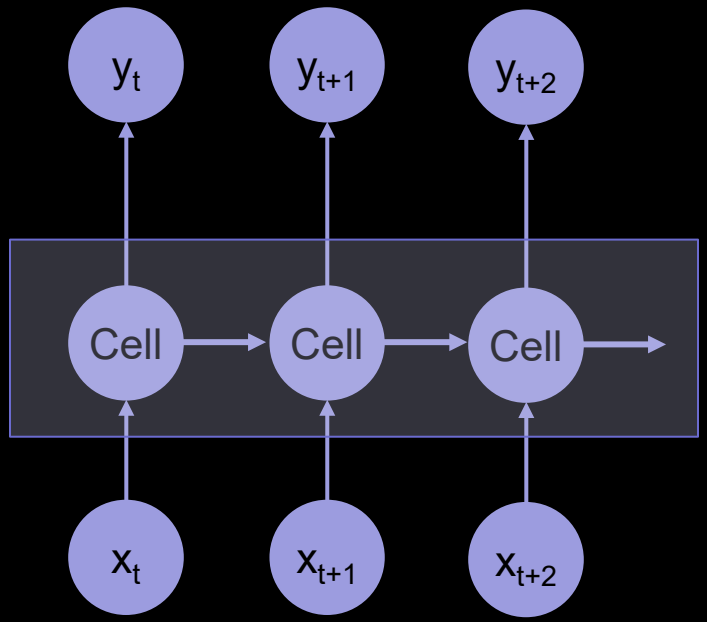
gli strati nascosti sono dei normali percettroni con una funzione di attivazione, es. tanh



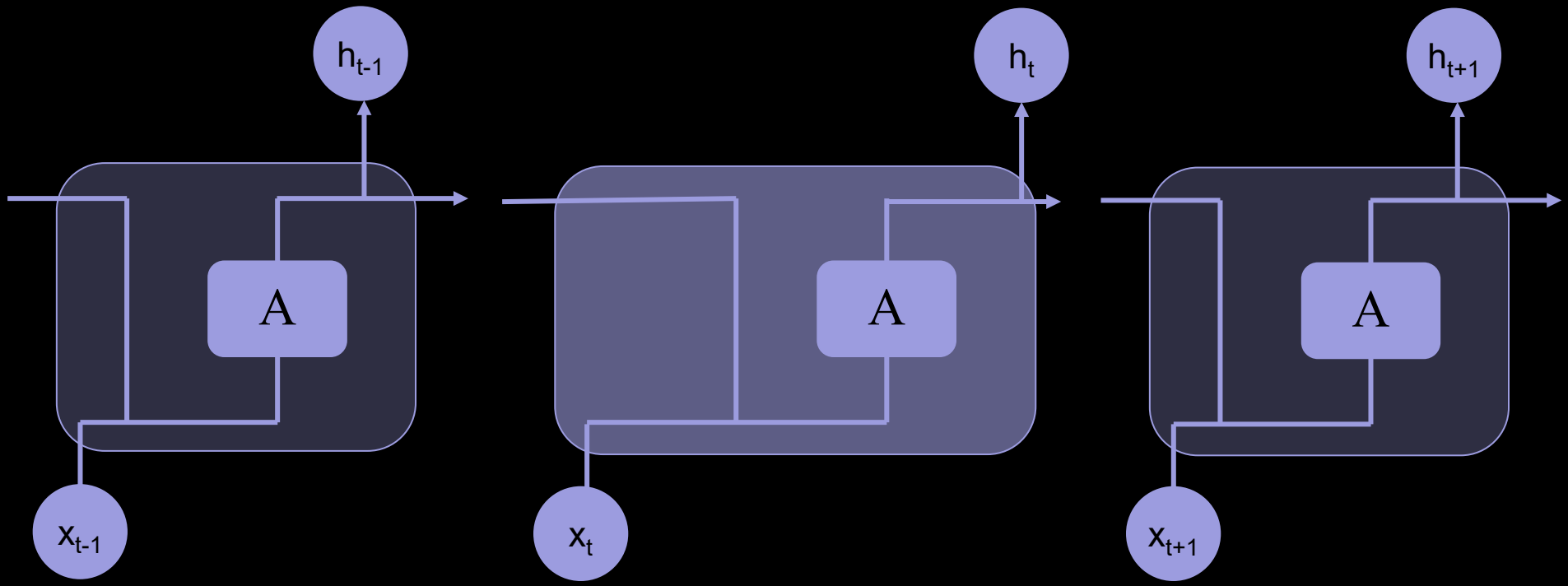
# Long short-term memory networks

Le LSTM sono dei tipi speciali di reti neurali n grado di catturare dipendenze tipicamente espresse in molti passi

viene sostituita da un particolare insieme di neuroni chiamato “cella di memoria”

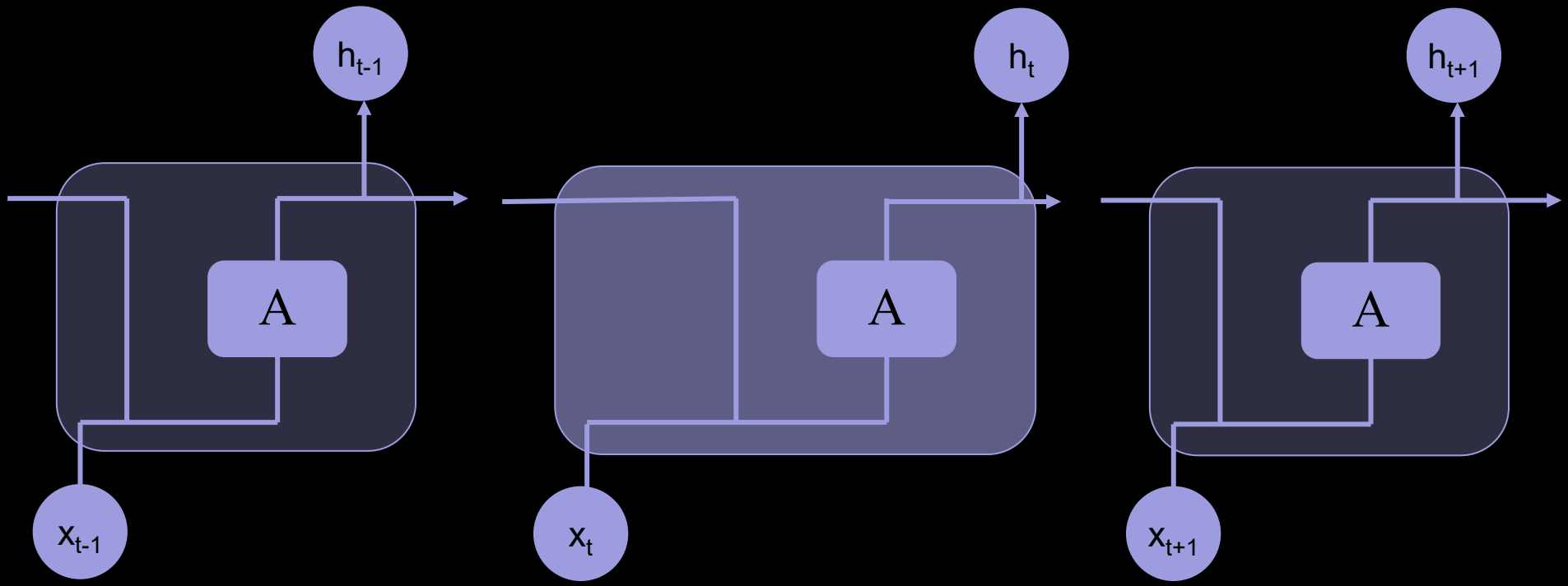


# Long short-term memory networks



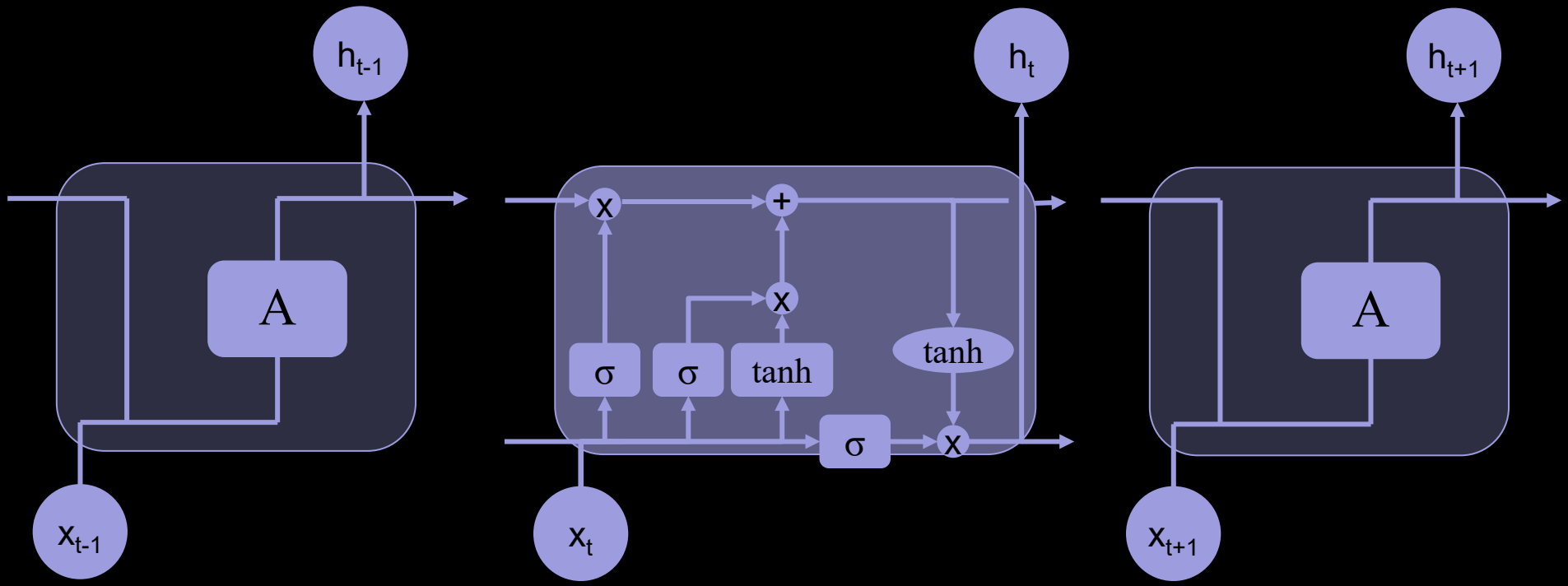
# Long short-term memory networks

- Dimenticare informazioni irrilevanti
- Creare aggiornamenti selettivi
- Ignorare alcuni elementi



# Long short-term memory networks

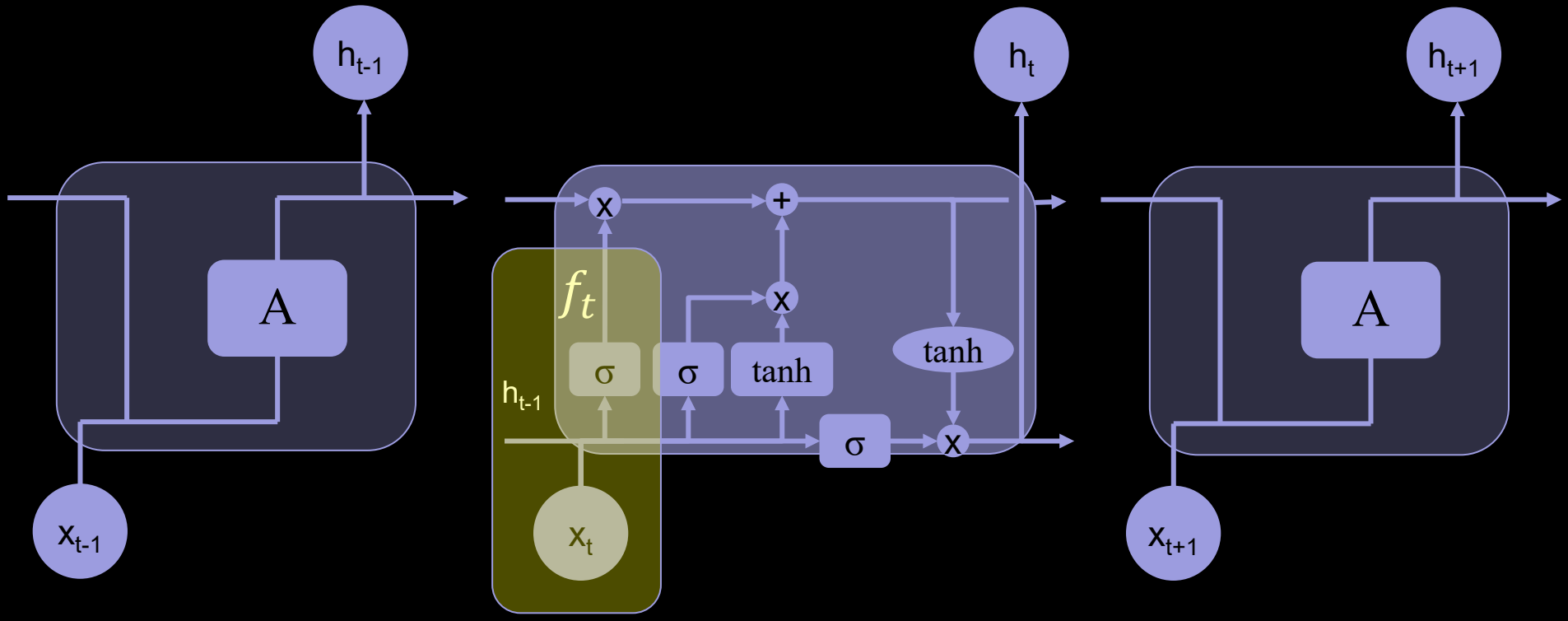
- Dimenticare informazioni irrilevanti
- Creare aggiornamenti selettivi
- Ignorare alcuni elementi



# Step 1: passi da ricordare

Il primo passo in una cella di memoria è scegliere quale informazione dovrebbe essere conservata e quale dovrebbe essere omessa in ogni passo.

E' scelto tramite una sigmoide e bisogna guardare al passo precedente

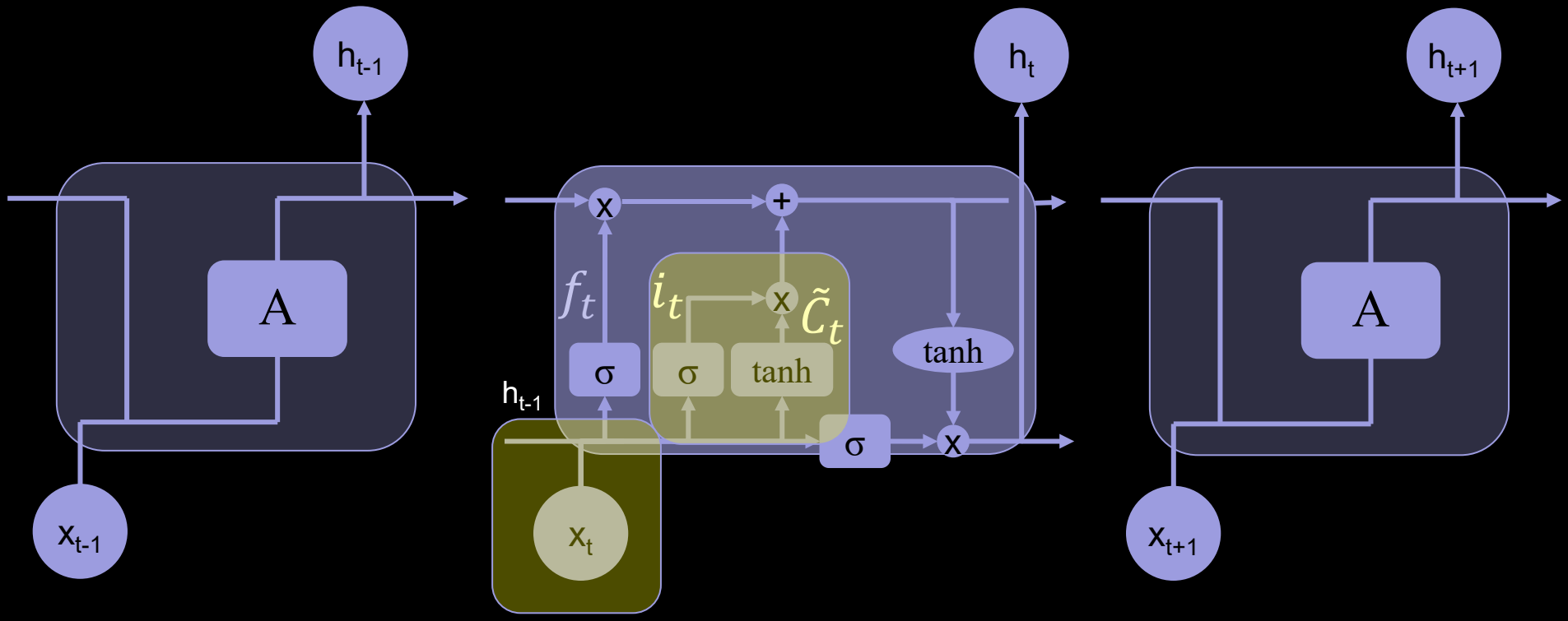



Forget gate: $f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f)$

# Step 2: decidere il peso sullo stato corrente

Include 2 componenti:

1. scegliere quale input lasciar entrare (la sigmoide fornirà solo 0 o 1)
2. pesare il valore dell'input corrente tramite la tanh scegliendo il livello di rilevanza (-1,1)



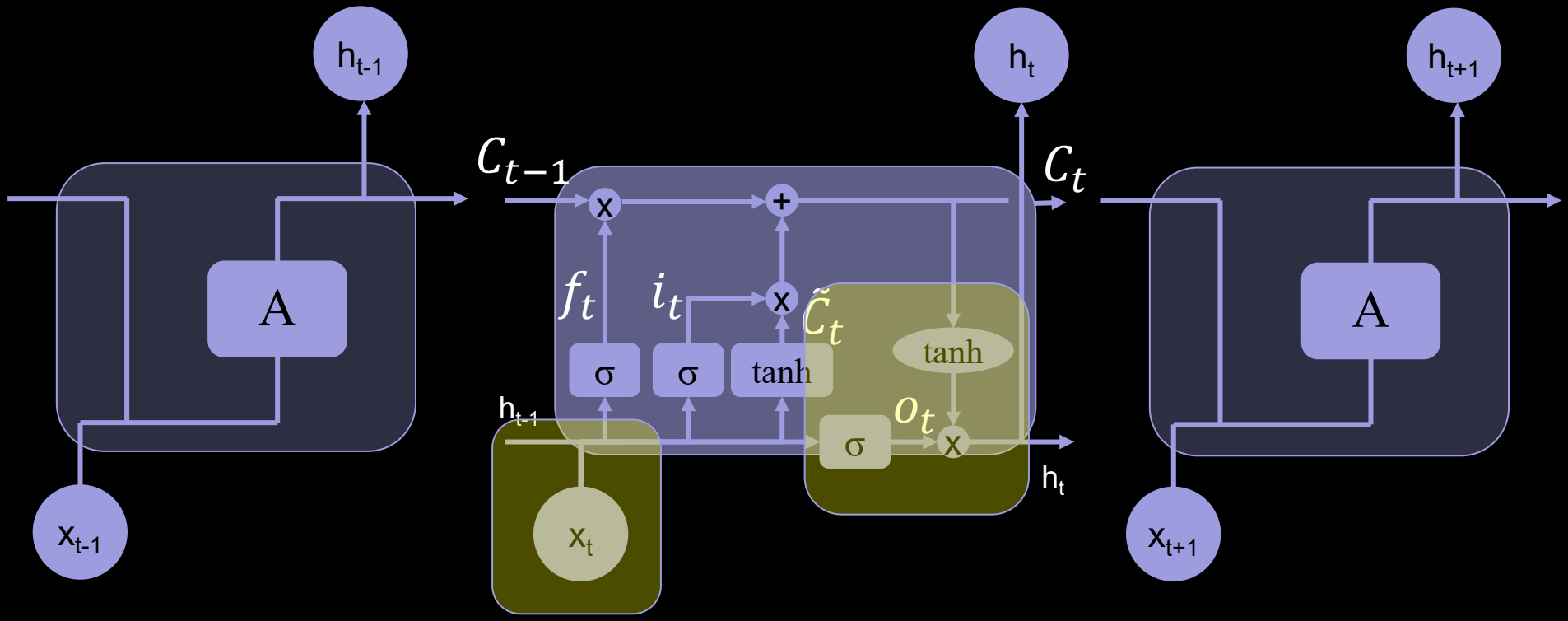

input gate: $i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$

rilevanza: $\tilde{C}_t = tanh(w_C \cdot [h_{t-1}, x_t] + b_C)$

# Step 3: decidere quale parte della rete contribuisce all'output

La sigmoide decide quale parte della cella arriva all'output $o_t$

L'output viene passato in un livello con tanh




Output gate: $o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$

layer: $h_t = o_t * \tanh(C_t)$

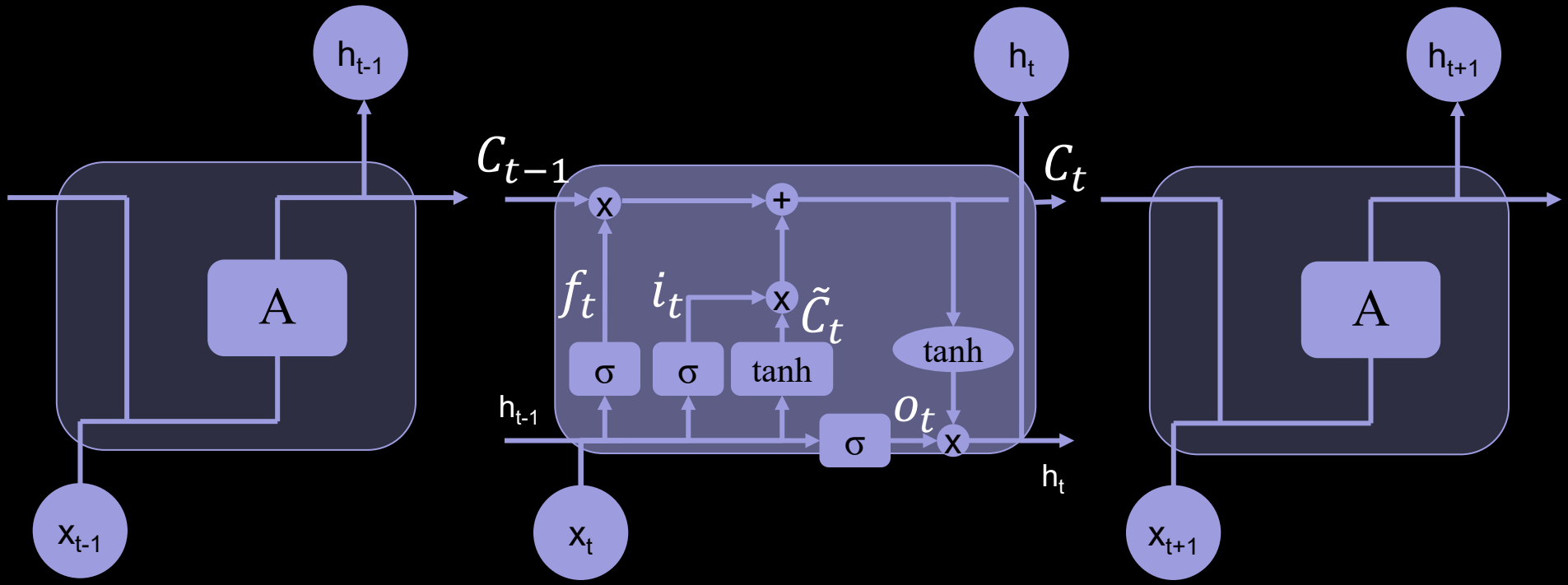
Forget gate: $f_t = \sigma\left(w_f \cdot [h_{t-1}, x_t] + b_f\right)$

Input gate: $i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$

Rilevanza: $\tilde{C}_t = tanh(w_C \cdot [h_{t-1}, x_t] + b_C)$

Output gate: $o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$

$h_t = o_t * \tanh(C_t)$



## Cell gate

Osservazioni:

- Tutte le equazioni dei gate sono in realtà relazioni matriciali con tensori
- L'output è dato tramite una serie di decisori su diversi passi della rete

Forget gate: $f_t = \sigma\left(w_f \cdot [h_{t-1}, x_t] + b_f\right)$

Input gate: $i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$

Rilevanza: $\tilde{C}_t = tanh(w_C \cdot [h_{t-1}, x_t] + b_C)$

Output gate: $o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$

$$h_t = o_t * \tanh(C_t)$$

## Cell gate

Osservazioni:

- Tutte le equazioni dei gate sono in realtà relazioni matriciali con tensori
- L'output è dato tramite una serie di decisori su diversi passi della rete

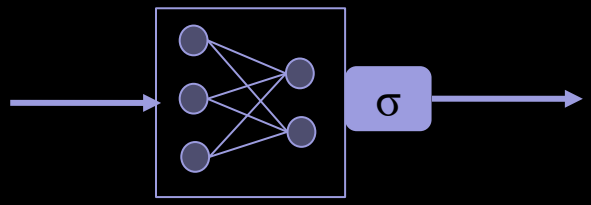
Forget gate: $f_t = \sigma\left(w_f \cdot [h_{t-1}, x_t] + b_f\right)$

Input gate: $i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$

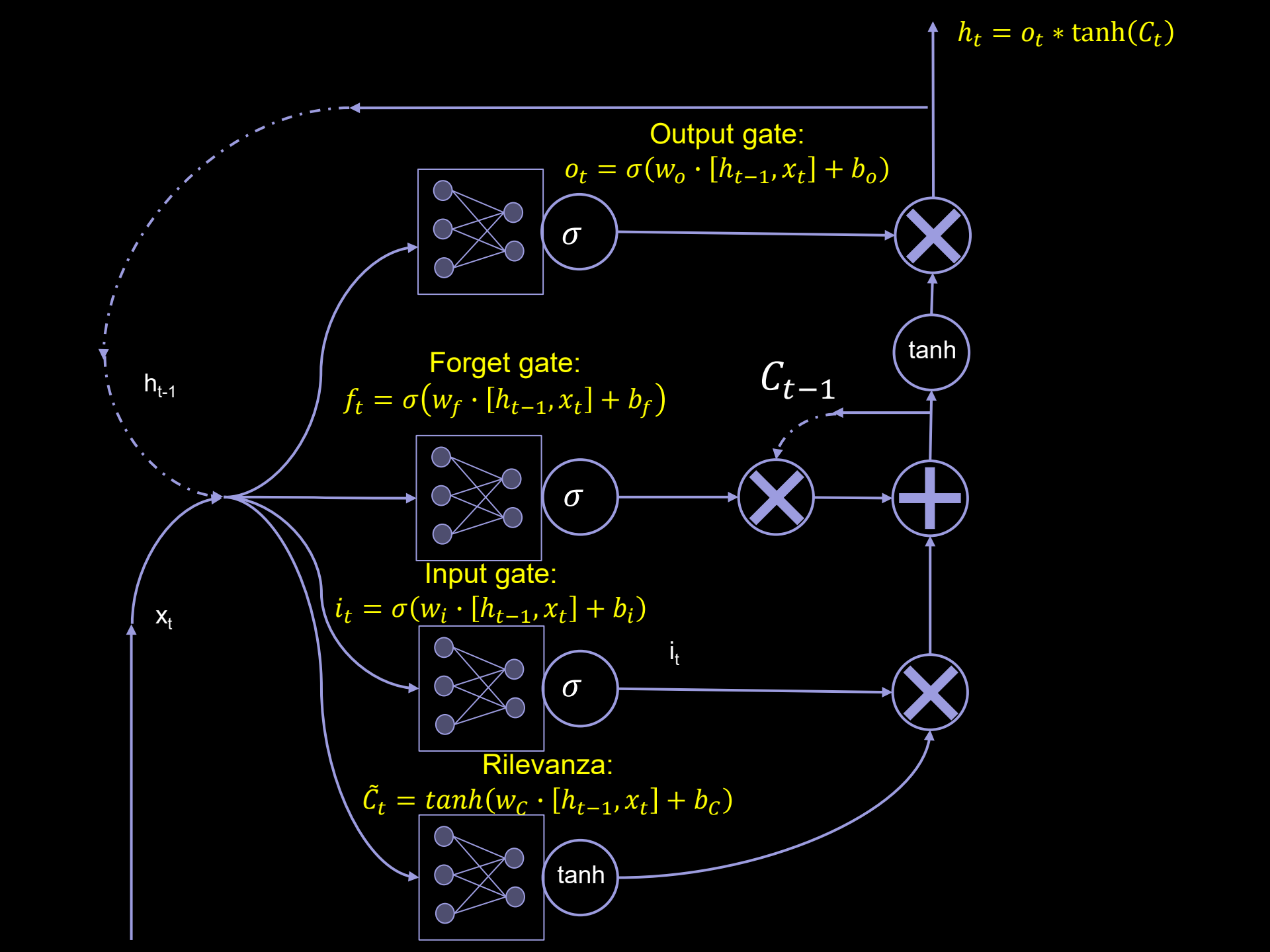
Rilevanza: $\tilde{C}_t = tanh(w_C \cdot [h_{t-1}, x_t] + b_C)$

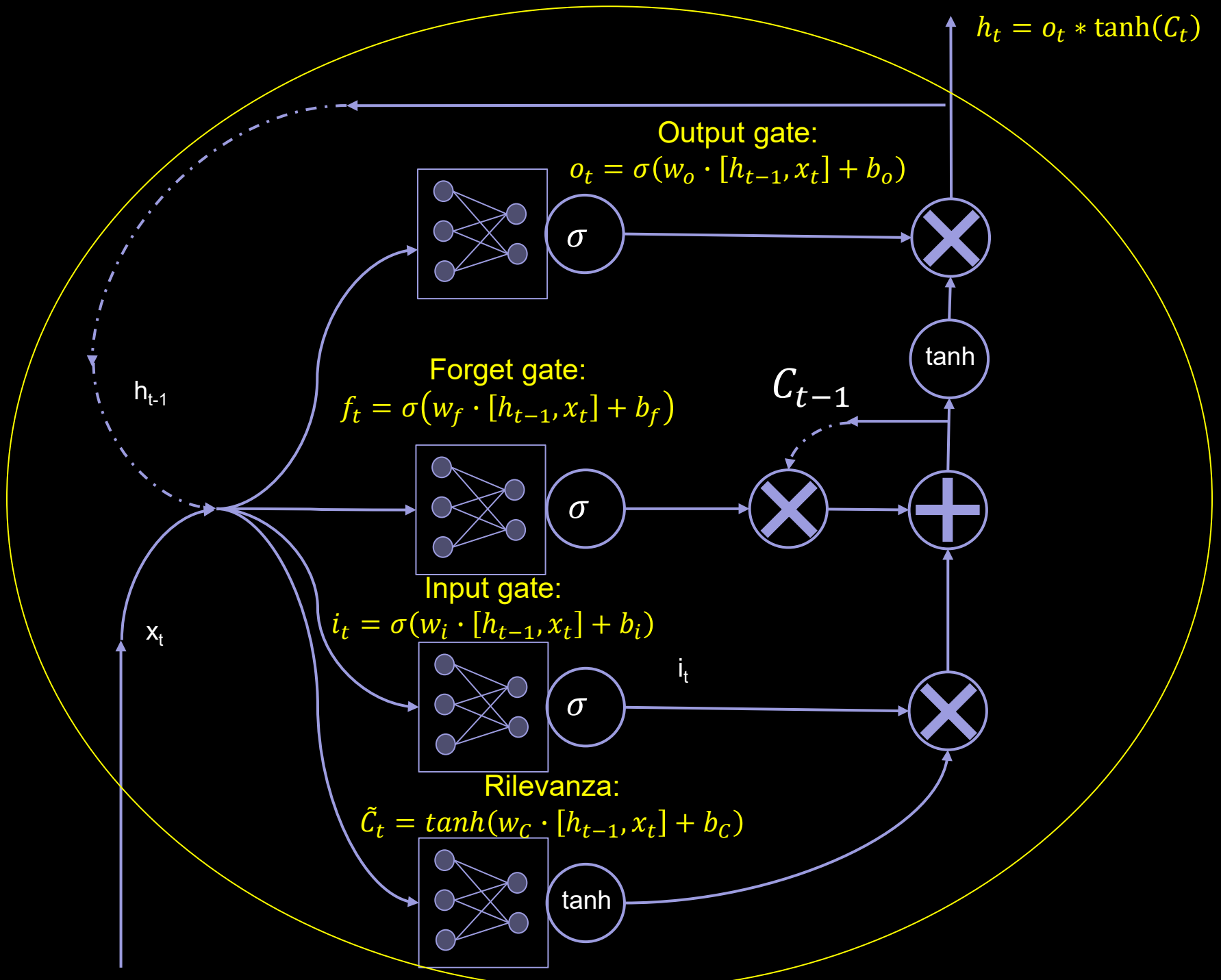
Output gate: $o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$

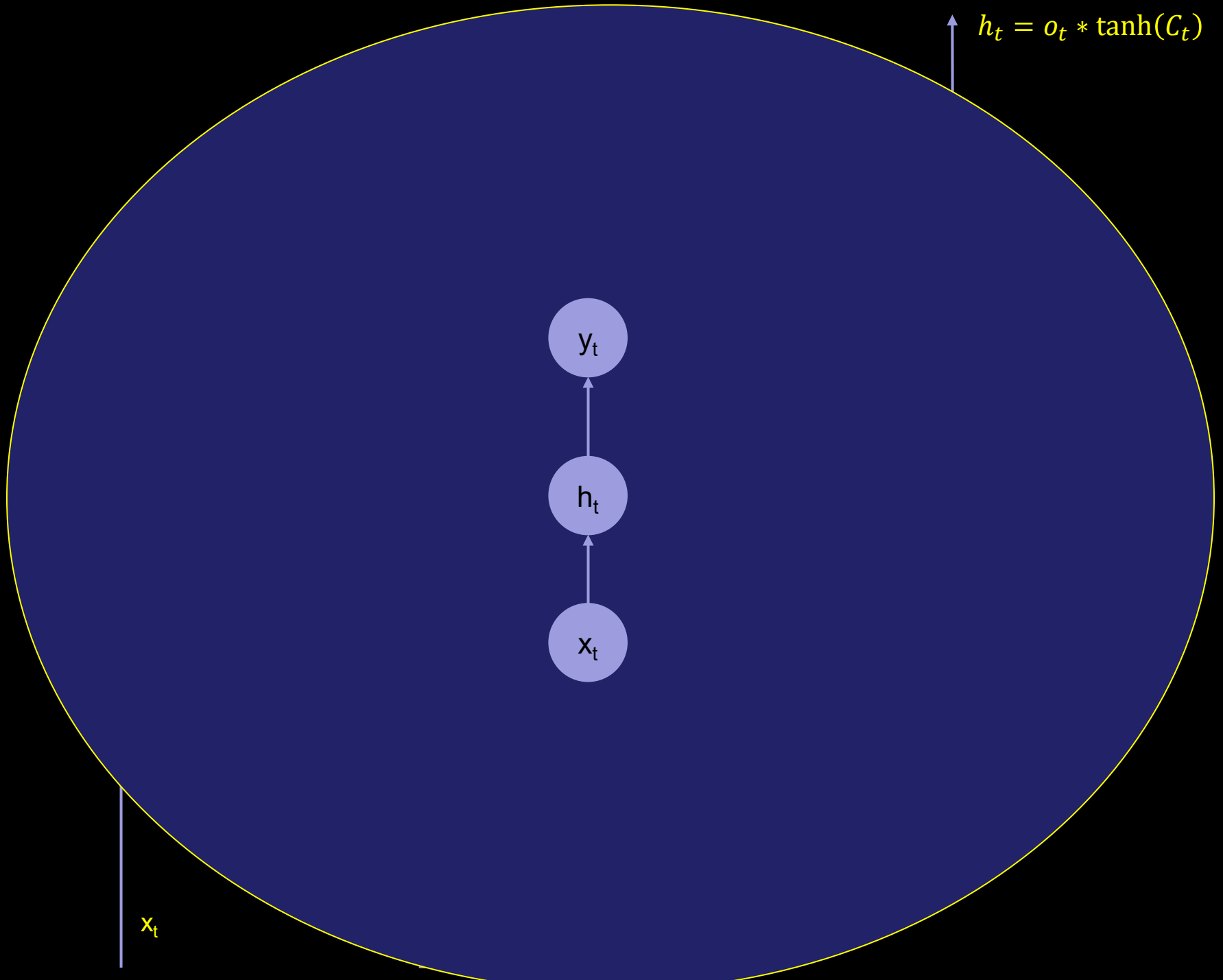
$$h_t = o_t * \tanh(C_t)$$




Il reale output è calcolato tramite una piccola rete neurale lineare sul tensore di ingresso


$h_t = o_t * \tanh(C_t)$
Output gate:
$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$
$\sigma$
tanh
$h_{t-1}$
Forget gate:
$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f)$
$C_{t-1}$
$\sigma$
Input gate:
$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$
$x_t$
$i_t$
$\sigma$
Rilevanza:
$\tilde{C}_t = tanh(w_C \cdot [h_{t-1}, x_t] + b_C)$
tanh


$h_t = o_t * \tanh(C_t)$
Output gate:
$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$
$\sigma$
tanh
$h_{t-1}$
Forget gate:
$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f)$
$C_{t-1}$
$\sigma$
Input gate:
$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$
$x_t$
$i_t$
$\sigma$
Rilevanza:
$\tilde{C}_t = tanh(w_C \cdot [h_{t-1}, x_t] + b_C)$
tanh


$h_t = o_t * \tanh(C_t)$
$y_t$
$h_t$
$x_t$
$x_t$