



Modeling Sequences

RNNs - part I

Outline

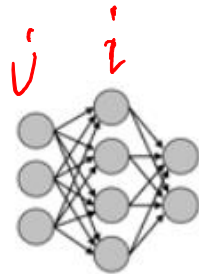
- Applications on modeling sequences of data
- Recurrent Neural Networks Models (RNNs)
- Vanishing (and exploding) gradients
- Long Short-Term Memory Models (LSTMs)

Sequences of Data

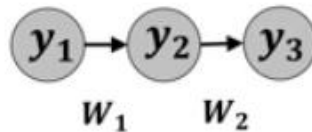
- Sequences in our world:
 - Audio
 - Text
 - Video
 - Weather
 - Stock market
- Sequential data is why we build RNN architectures.
- RNNs are tools for making predictions about sequences.

Recap: Feed-Fwd Networks

- Forward:



$$y_i = g\left(\sum_j W_{ij}x_j + b_i\right)$$



$$\mathbf{y}_k = g(W\mathbf{y}_{k-1} + \mathbf{b})$$

- Backward:

Alternative Notation

E: error

C: cost

L: loss

Error

$$\frac{\partial \boxed{E}}{\partial W} = \frac{\partial E}{\partial g} \frac{\partial g}{\partial a} \frac{\partial a}{\partial W}$$

Limitations of Feed-Fwd Networks

- Limitations of feed-forward networks

- **Fixed length**

- Inputs and outputs are of fixed lengths*

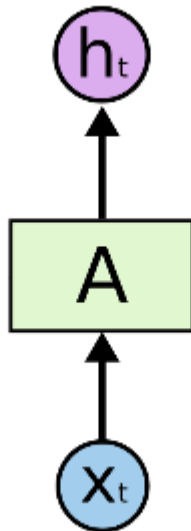
- **Independence**

- Data (example: images) are independent of one another*

Advantages of RNN Models

- What feed-forward networks cannot do
 - **Variable length**
“We would like to accommodate temporal sequences of various lengths.”
 - **Temporal dependence**
“To predict where a pedestrian is at the next point in time, this depends on where he/she were in the previous time step.”

Vanilla Neural Network (NN)



- NN

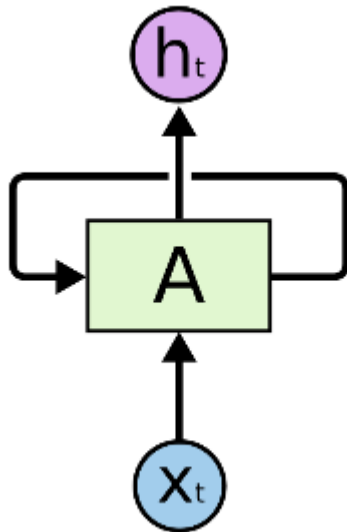
x_t : input/event

h_t : output/prediction

A : chunk of NN

Every input is treated independently.

Recurrent Neural Network (RNN)



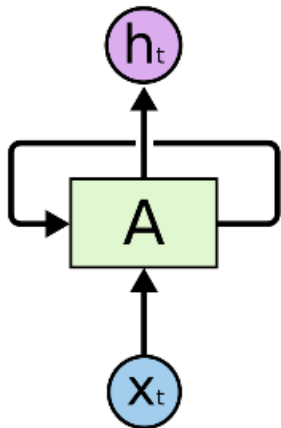
- RNN

The loop allows information to be passed from one time step to the next.

Now we are modeling the dynamics.

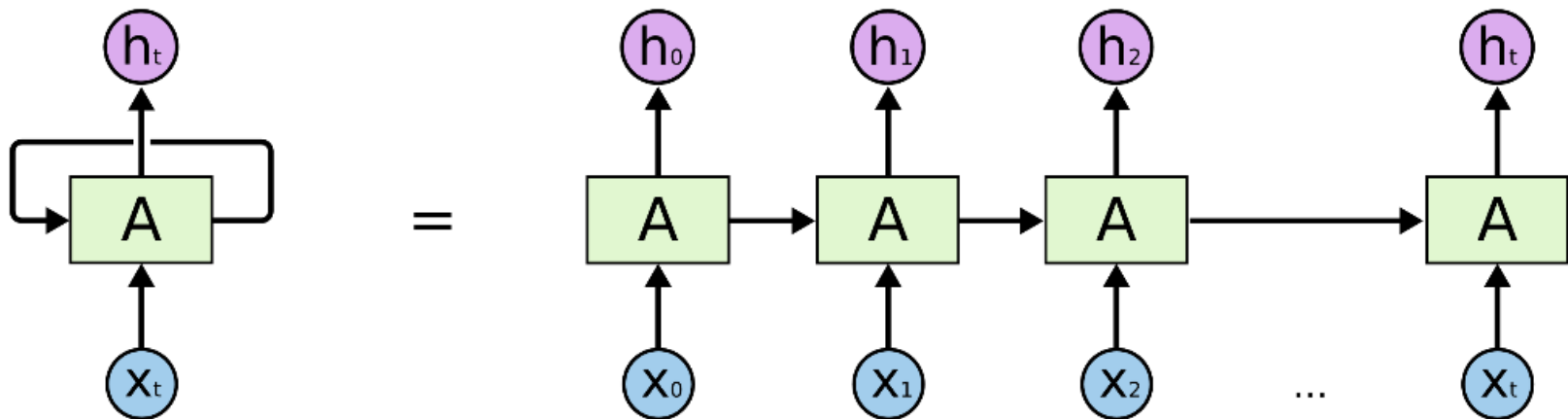
Recurrent Neural Network (RNN)

- A recurrent neural network can be thought of as multiple copies of the same network, each **passing a message to a successor**.

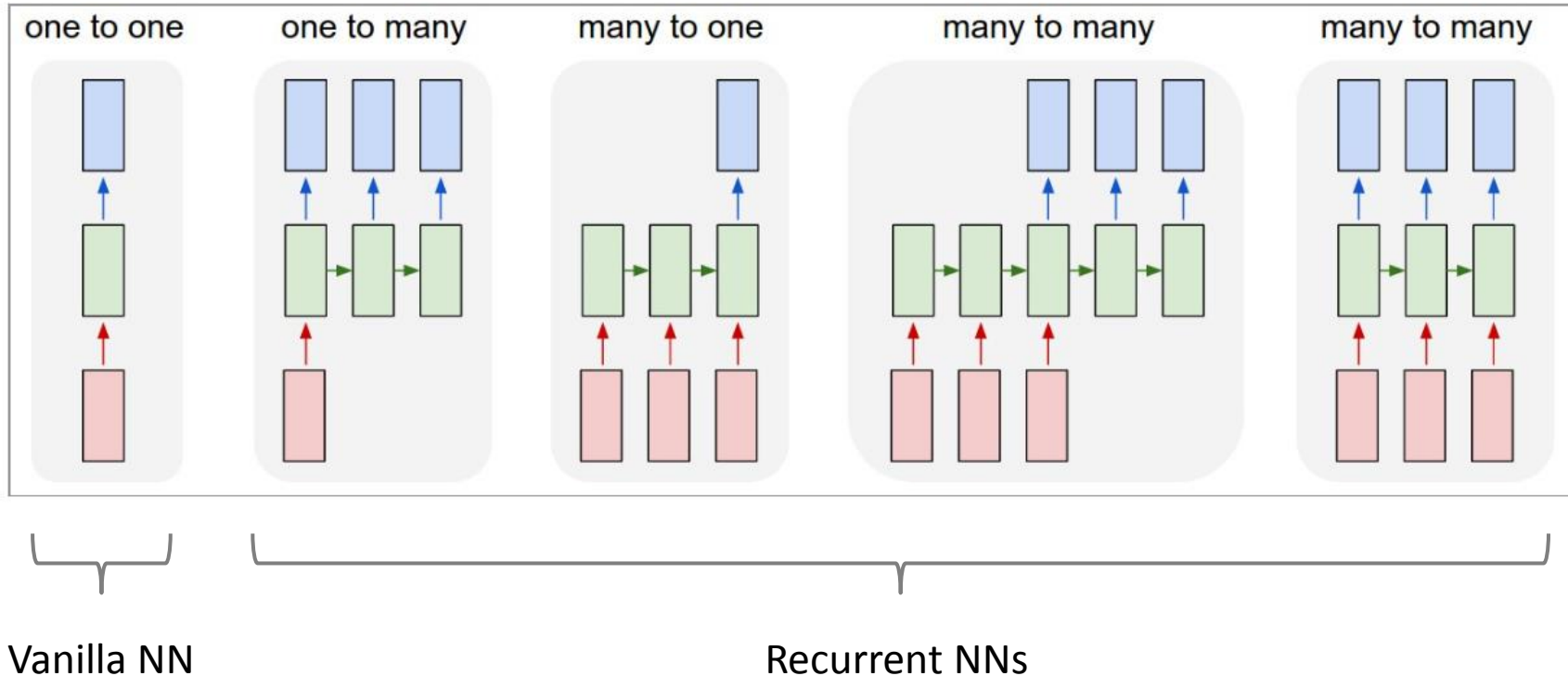


Recurrent Neural Network (RNN)

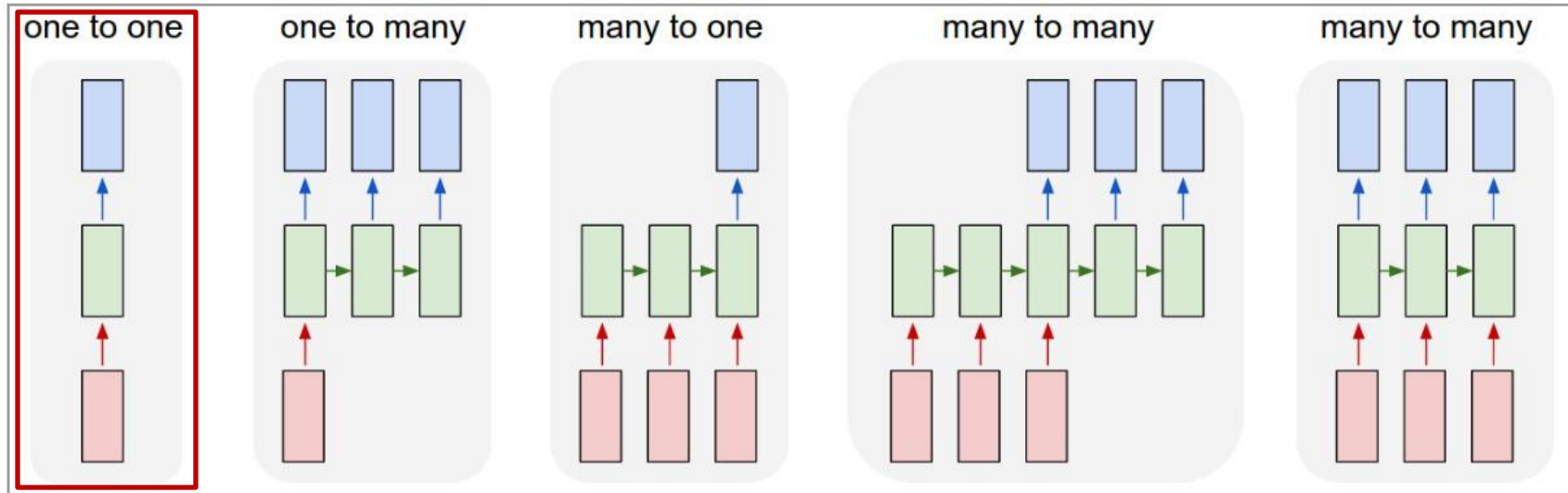
- A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.



RNN Architectures



One-to-one



Vanilla mode of processing without RNN

Example: Image classification

Example: One-to-one

Vanilla mode of processing without RNN

Example: Image classification



Prediction: "Dog"

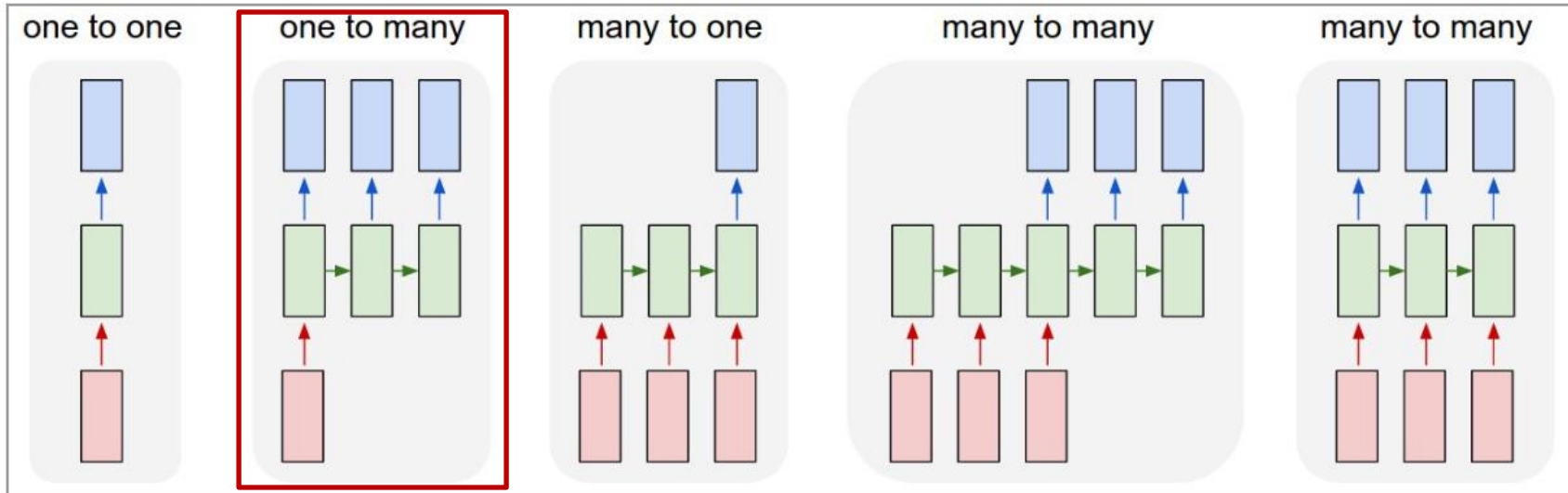


Vanilla NN



Image

One-to-many



Sequence output

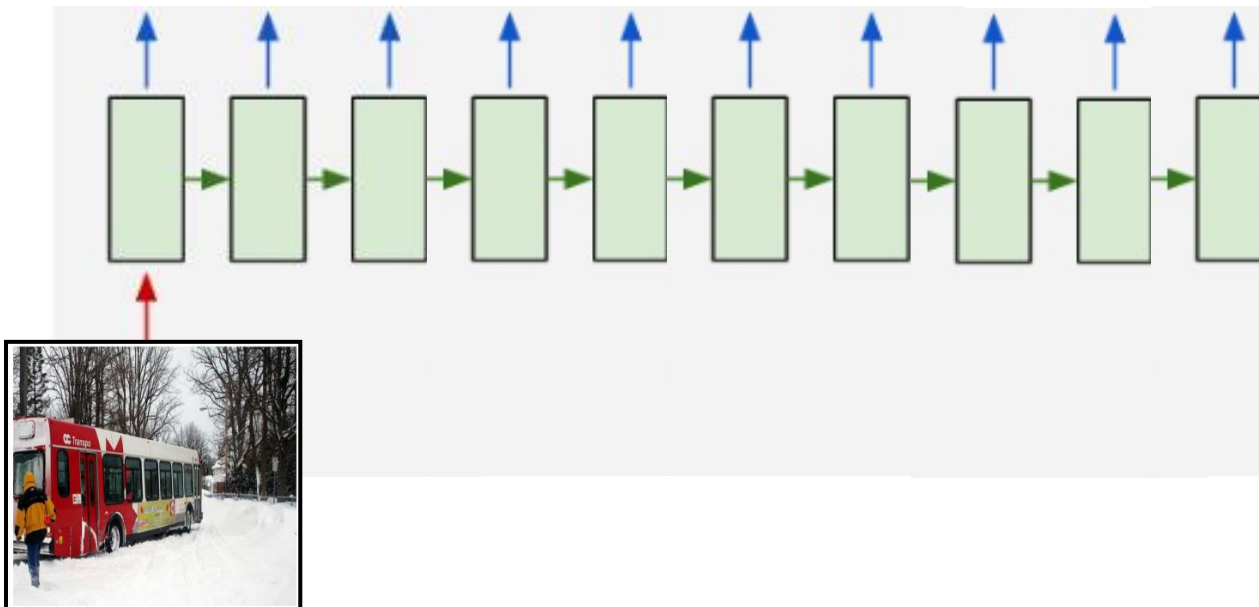
Example: Image captioning

Example: One-to-many

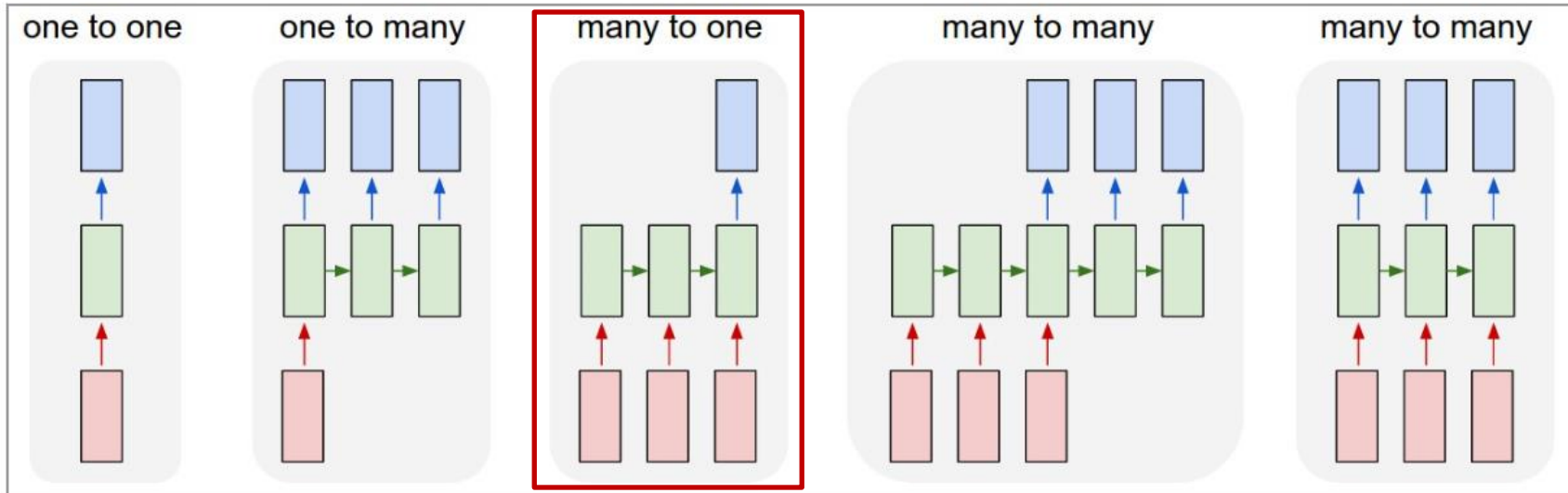
Sequence output

Example: Image Captioning

Bus driving down a snowy road next to trees <EOS>



Many-to-one



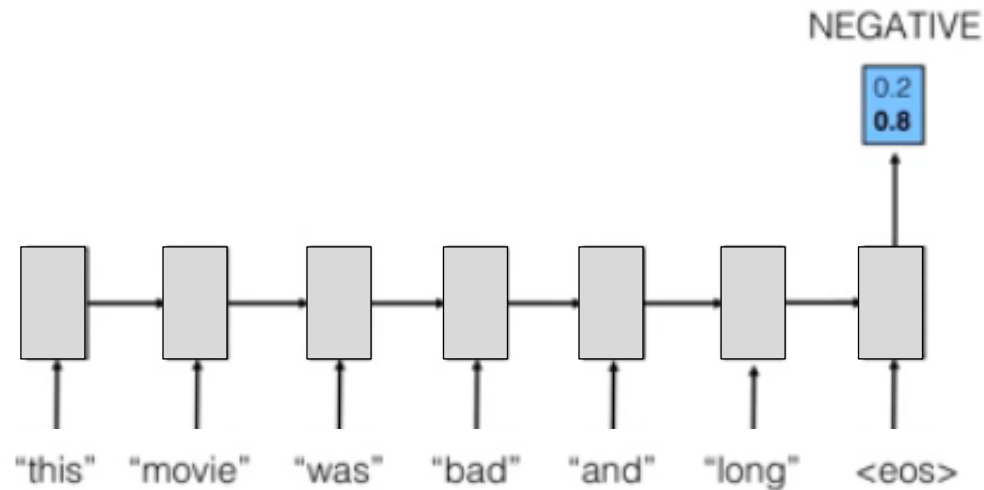
Sequence input

Examples: *Sentiment analysis*
Action recognition

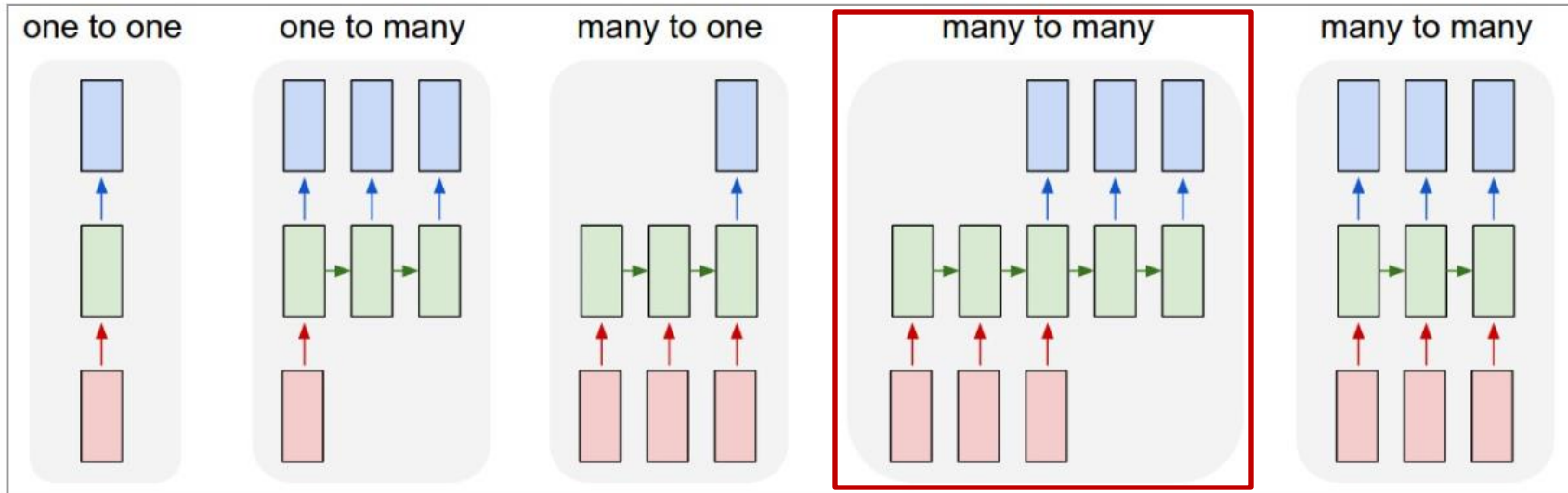
Example: Many-to-one

Sequence input

Example: Sentiment analysis



Many-to-many



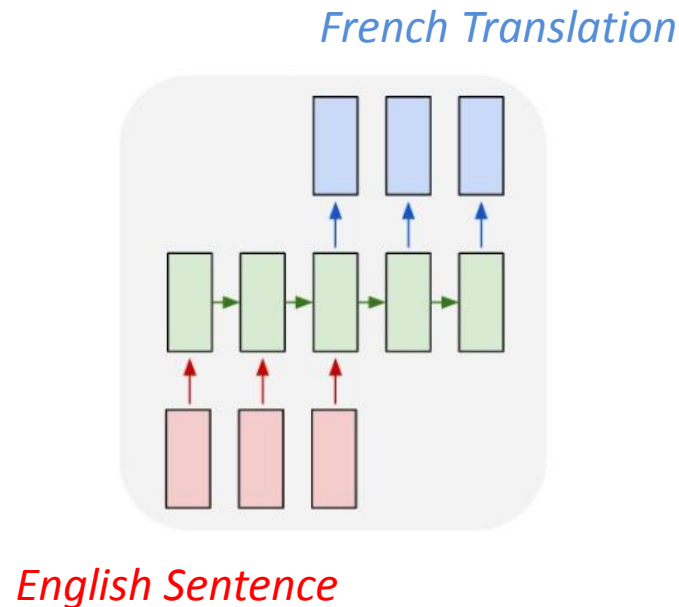
Sequence input and sequence output

Example: **Machine translation**

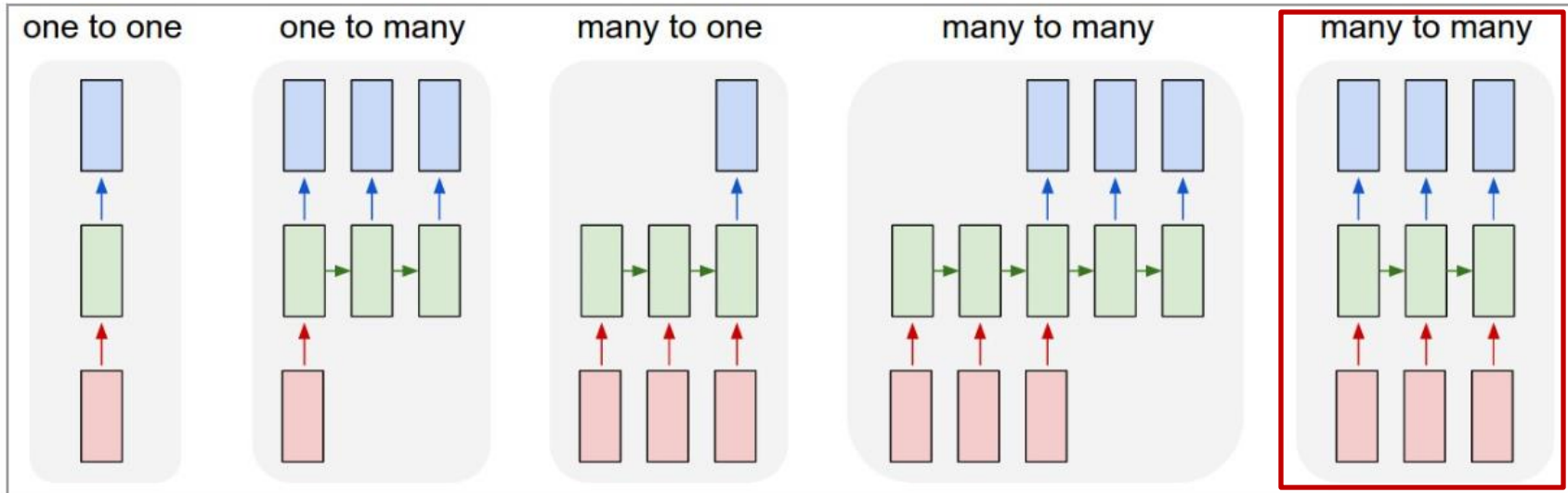
Example: Many-to-many

Sequence input and sequence output

Example: Machine translation



Synced Many-to-many

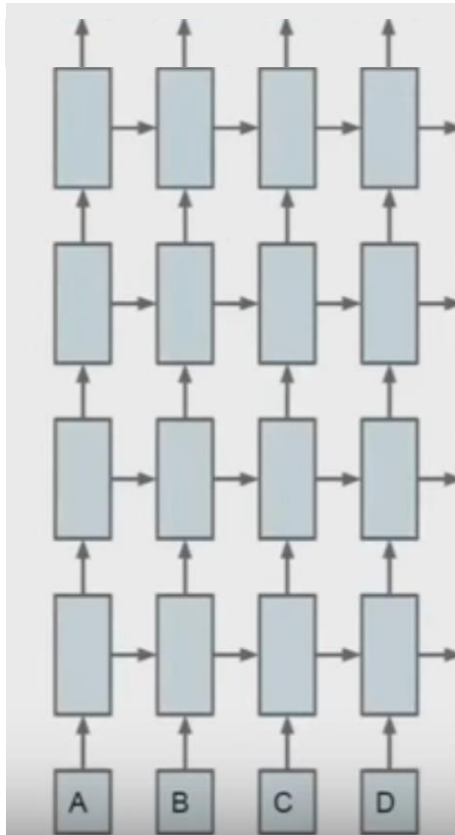


Synced sequence input and output

Examples: *Tracking*

Early action detection

Deep RNNs



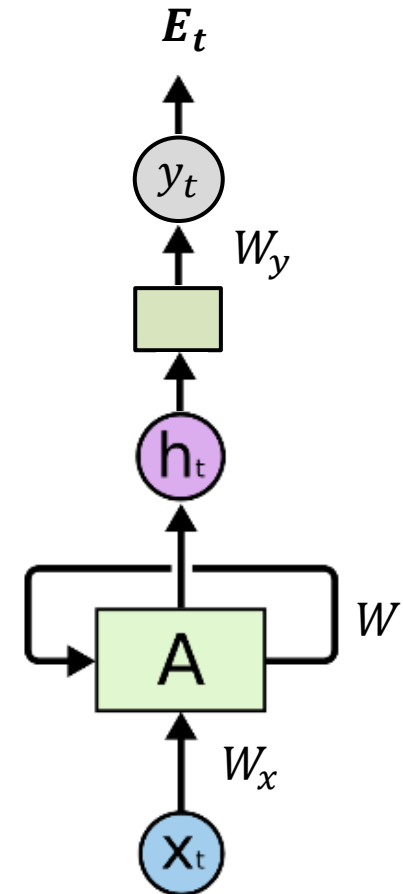
- Stacking RNNs
- More ways of propagating information!
- Requires a lot of data!

Fwd RNN TT

- Forward pass through time

$$h_t = W\phi(h_{t-1}) + W_x x_t$$

$$y_t = W_y \phi(h_t)$$

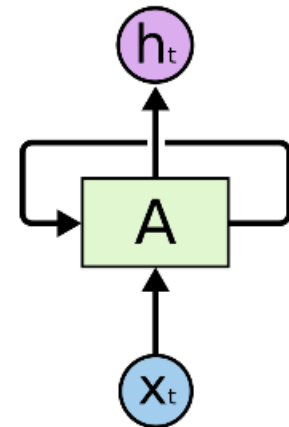


Recurrent Neural Network (RNN)

Aside: Forward pass

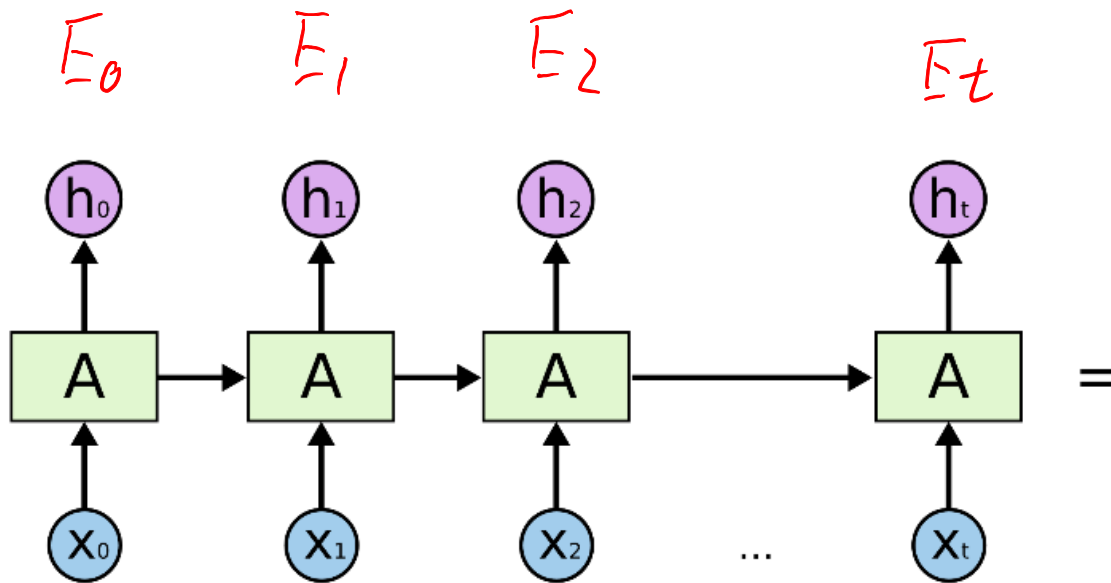
$$h_t = W\phi(h_{t-1}) + W_x x_t$$

$$y_t = W_y \phi(h_t)$$



Recurrent Neural Network (RNN)

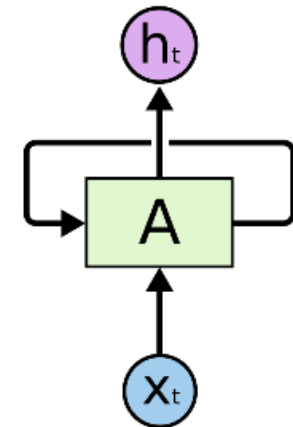
- Error or cost is computed for each prediction.



Aside: Forward pass

$$h_t = W\phi(h_{t-1}) + W_x x_t$$

$$y_t = W_y \phi(h_t)$$



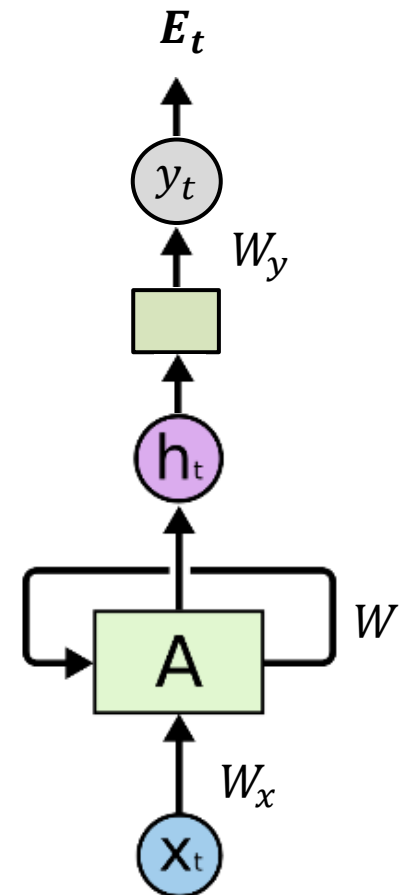
BP TT

- Backpropagation through time

$$\frac{\partial E}{\partial W} = \sum_{t=1}^T \frac{\partial E_t}{\partial W}$$

*W is same for
 t_1, t_2, \dots, T*

Aside: Forward pass



BP TT

- Backpropagation through time

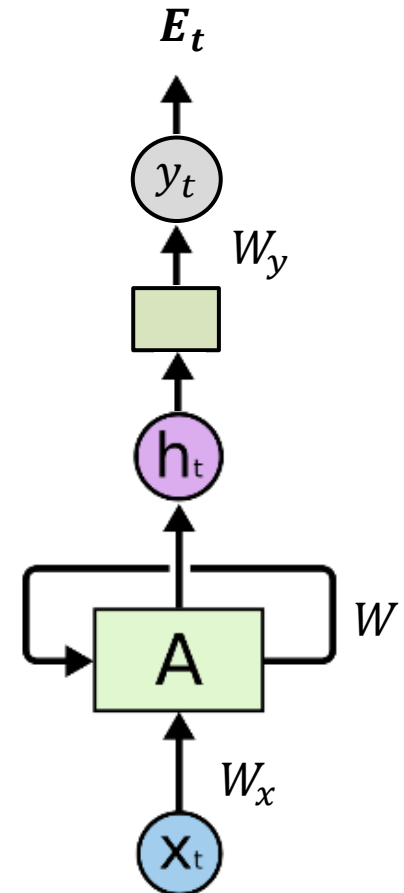
$$\frac{\partial E}{\partial W} = \sum_{t=1}^T \boxed{\frac{\partial E_t}{\partial W}}$$

*Single
time step*

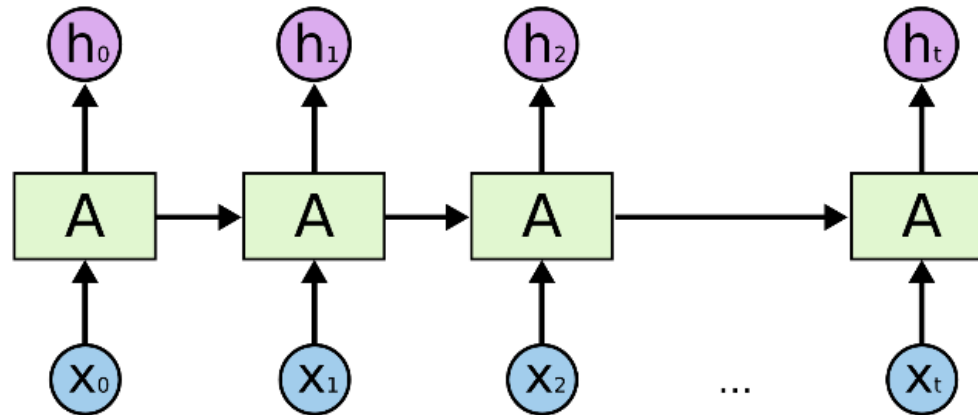
$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

*k is previous time steps
k can be t-1, t-2 ... 1*

Aside: Forward pass

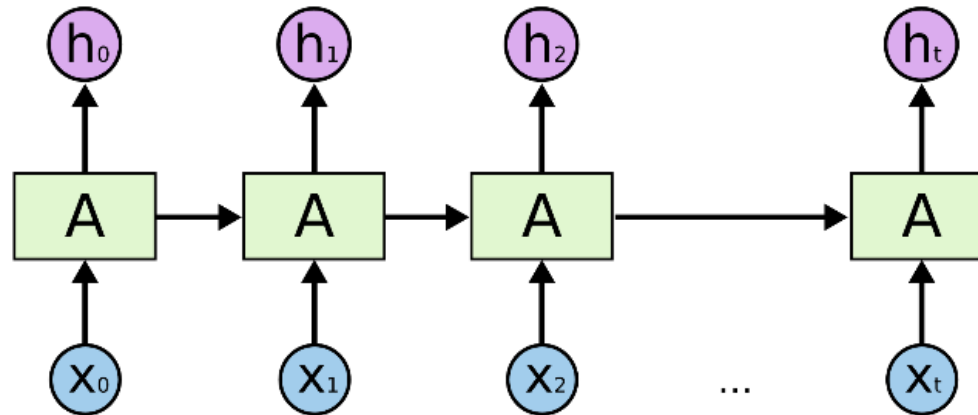


BP TT



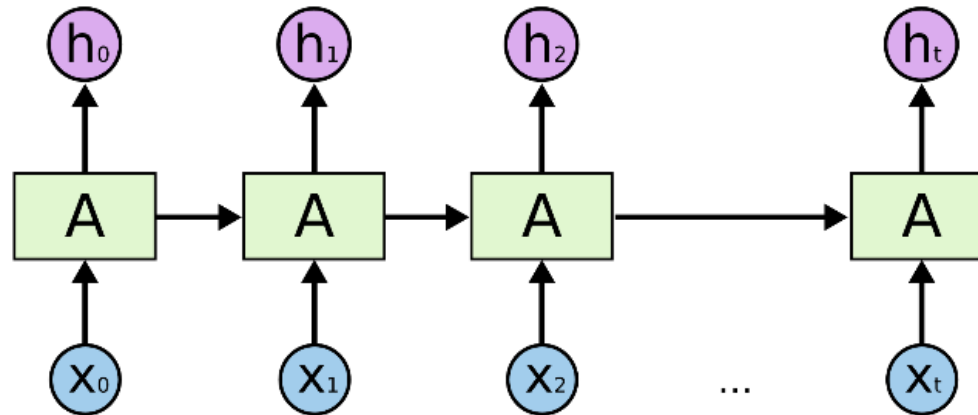
$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

BP TT




$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \boxed{\frac{\partial h_t}{\partial h_k}} \frac{\partial h_k}{\partial W}$$

BP TT

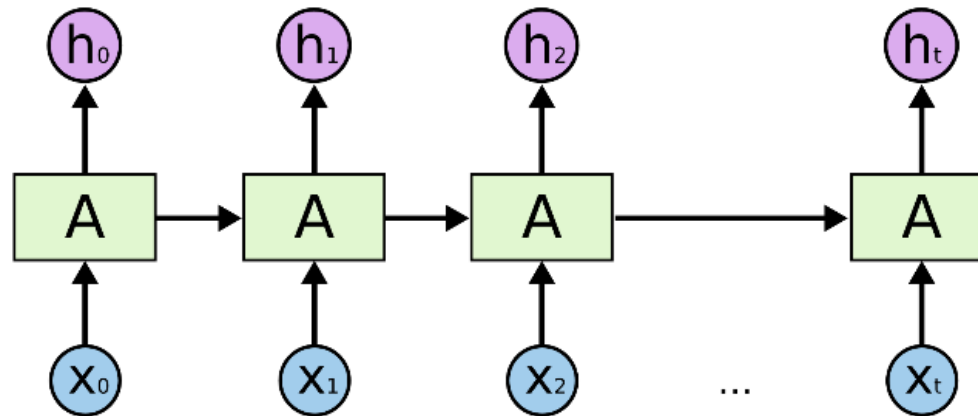


$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \boxed{\frac{\partial h_t}{\partial h_k}} \frac{\partial h_k}{\partial W}$$



$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}}$$

BP TT



$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \boxed{\frac{\partial h_t}{\partial h_k}} \frac{\partial h_k}{\partial W}$$

t-k steps

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}}$$

For example @ $t = 2$,

$$\frac{\partial h_2}{\partial h_0} = \prod_{i=1}^2 \frac{\partial h_i}{\partial h_{i-1}} = \frac{\partial h_1}{\partial h_0} \frac{\partial h_2}{\partial h_1}$$

Vanishing (and Exploding) Gradients

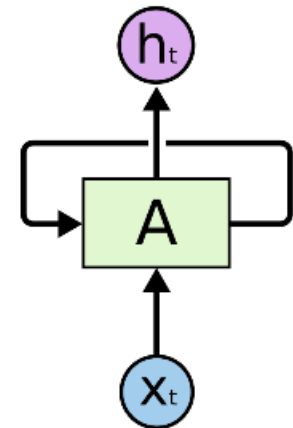
$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}}$$

Aside: Forward pass

$$h_t = W\phi(h_{t-1}) + W_x x_t$$

$$y_t = W_y \phi(h_t)$$



Vanishing (and Exploding) Gradients

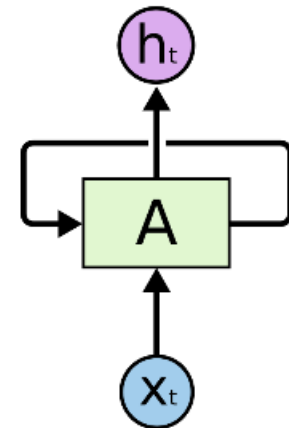
$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i=k+1}^t W^T \text{diag}[\phi'(h_{i-1})]$$

Aside: Forward pass

$$h_t = W\phi(h_{t-1}) + W_x x_t$$

$$y_t = W_y \phi(h_t)$$



Vanishing (and Exploding) Gradients

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

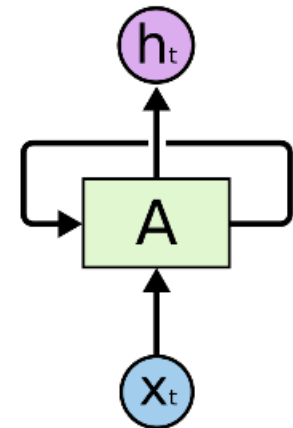
$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i=k+1}^t W^T \text{diag}[\phi'(h_{i-1})]$$

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|$$

Aside: Forward pass

$$h_t = W\phi(h_{t-1}) + W_x x_t$$

$$y_t = W_y \phi(h_t)$$



Vanishing (and Exploding) Gradients

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

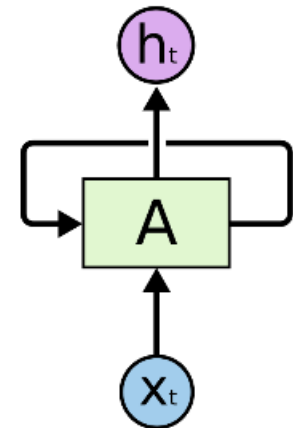
$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i=k+1}^t W^T \text{diag}[\phi'(h_{i-1})]$$

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\| \leq \|W^T\| \|\text{diag}[\phi'(h_{i-1})]\|$$

Aside: Forward pass

$$h_t = W\phi(h_{t-1}) + W_x x_t$$

$$y_t = W_y \phi(h_t)$$



Vanishing (and Exploding) Gradients

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i=k+1}^t W^T \text{diag}[\phi'(h_{i-1})]$$

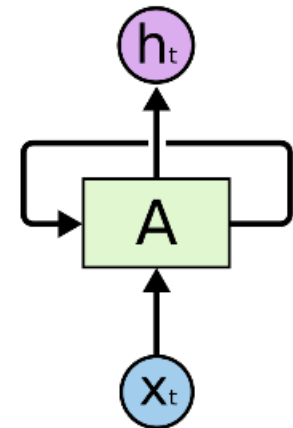
$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\| \leq \|W^T\| \|\text{diag}[\phi'(h_{i-1})]\| \leq \gamma_W \gamma_\phi$$

*largest
eigen value of W*

Aside: Forward pass

$$h_t = W\phi(h_{t-1}) + W_x x_t$$

$$y_t = W_y \phi(h_t)$$



Vanishing (and Exploding) Gradients

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i=k+1}^t W^T \text{diag}[\phi'(h_{i-1})]$$

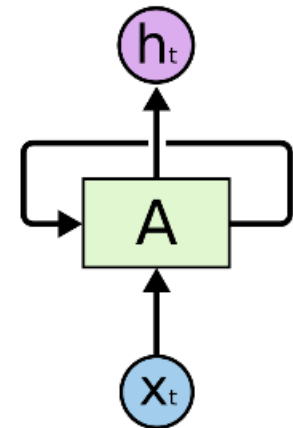
$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\| \leq \|W^T\| \|\text{diag}[\phi'(h_{i-1})]\| \leq \gamma_W \gamma_\phi$$

$$\prod_{i=k+1}^t \left\| \frac{\partial h_i}{\partial h_{i-1}} \right\| \leq (\gamma_W \gamma_\phi)^{t-k}$$

Aside: Forward pass

$$h_t = W\phi(h_{t-1}) + W_x x_t$$

$$y_t = W_y \phi(h_t)$$



Vanishing (and Exploding) Gradients

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i=k+1}^t W^T \text{diag}[\phi'(h_{i-1})]$$

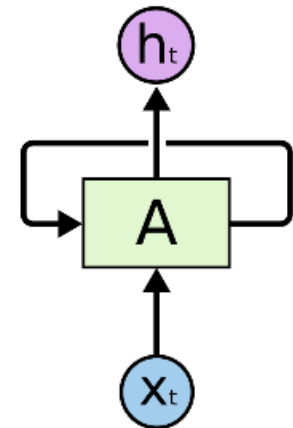
$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\| \leq \|W^T\| \|\text{diag}[\phi'(h_{i-1})]\| \leq \gamma_W \gamma_\phi$$

$$\prod_{i=k+1}^t \left\| \frac{\partial h_i}{\partial h_{i-1}} \right\| \leq \underbrace{(\gamma_W \gamma_\phi)^{t-k}}_{\substack{<1 \text{ vanishing} \\ >1 \text{ exploding}}}$$

Aside: Forward pass

$$h_t = W\phi(h_{t-1}) + W_x x_t$$

$$y_t = W_y \phi(h_t)$$

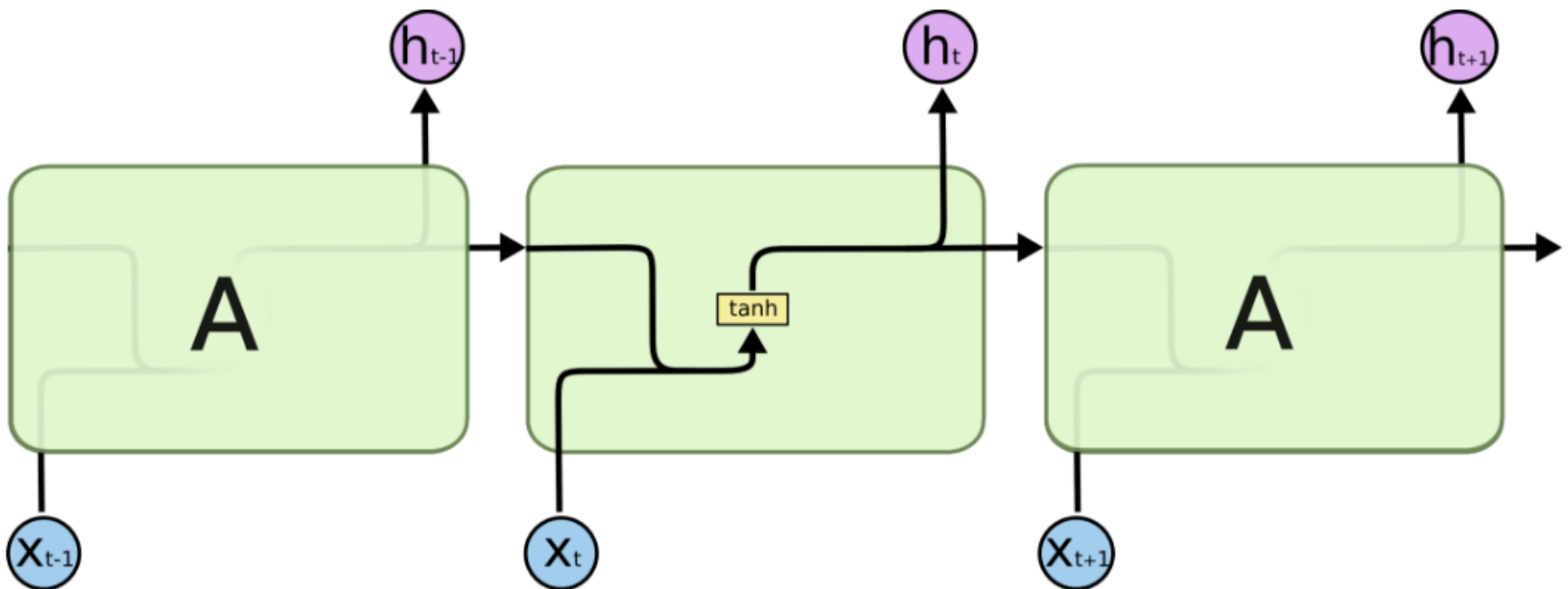


Vanishing (and Exploding) Gradients

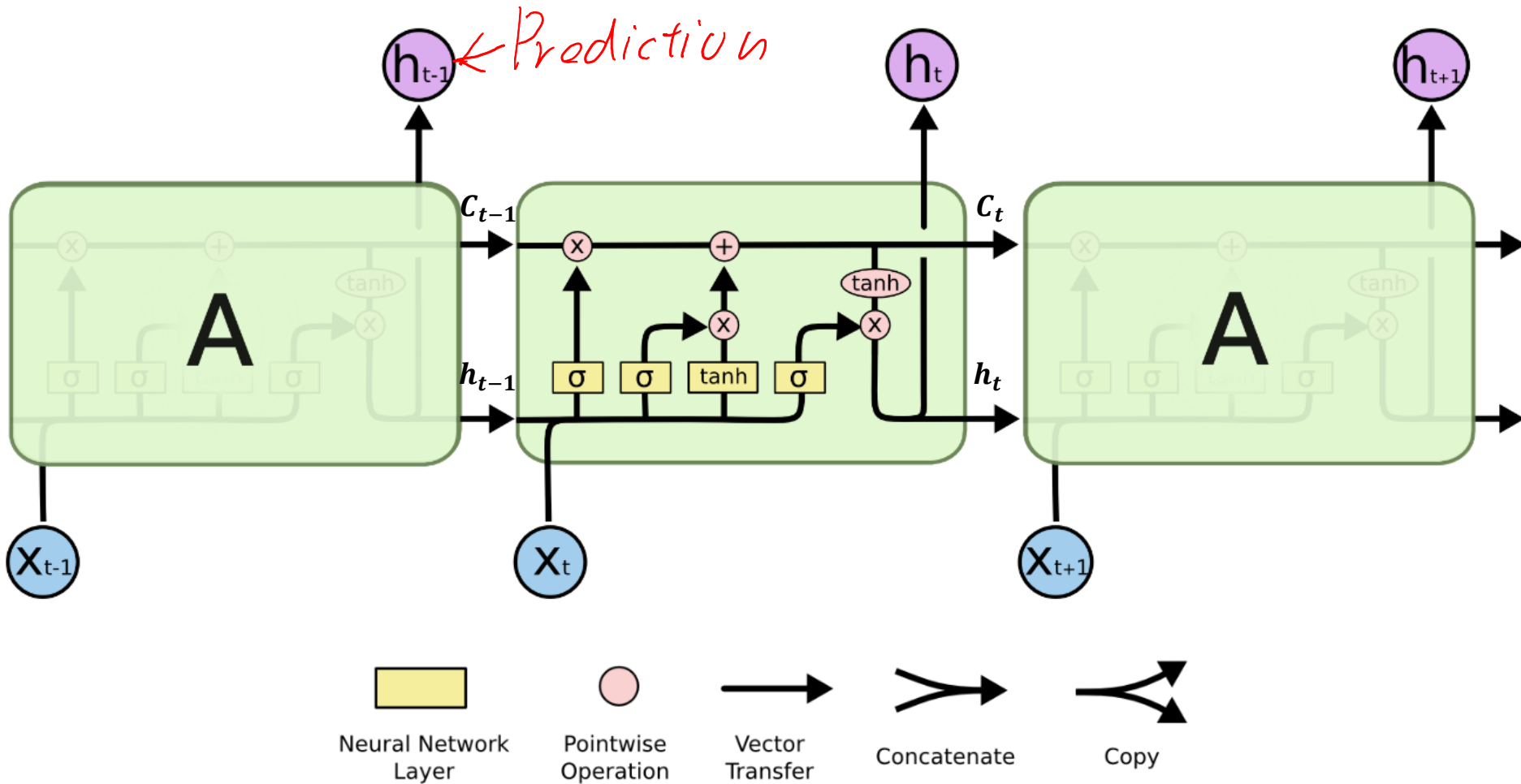
- Exploding Gradients
 - Easy to detect
 - Clip the gradient at a threshold
- Vanishing Gradients
 - More difficult to detect
 - Architectures designed to combat the problem of vanishing gradients. Example: LSTMs by *Schmidhuber et al.*

RNNs

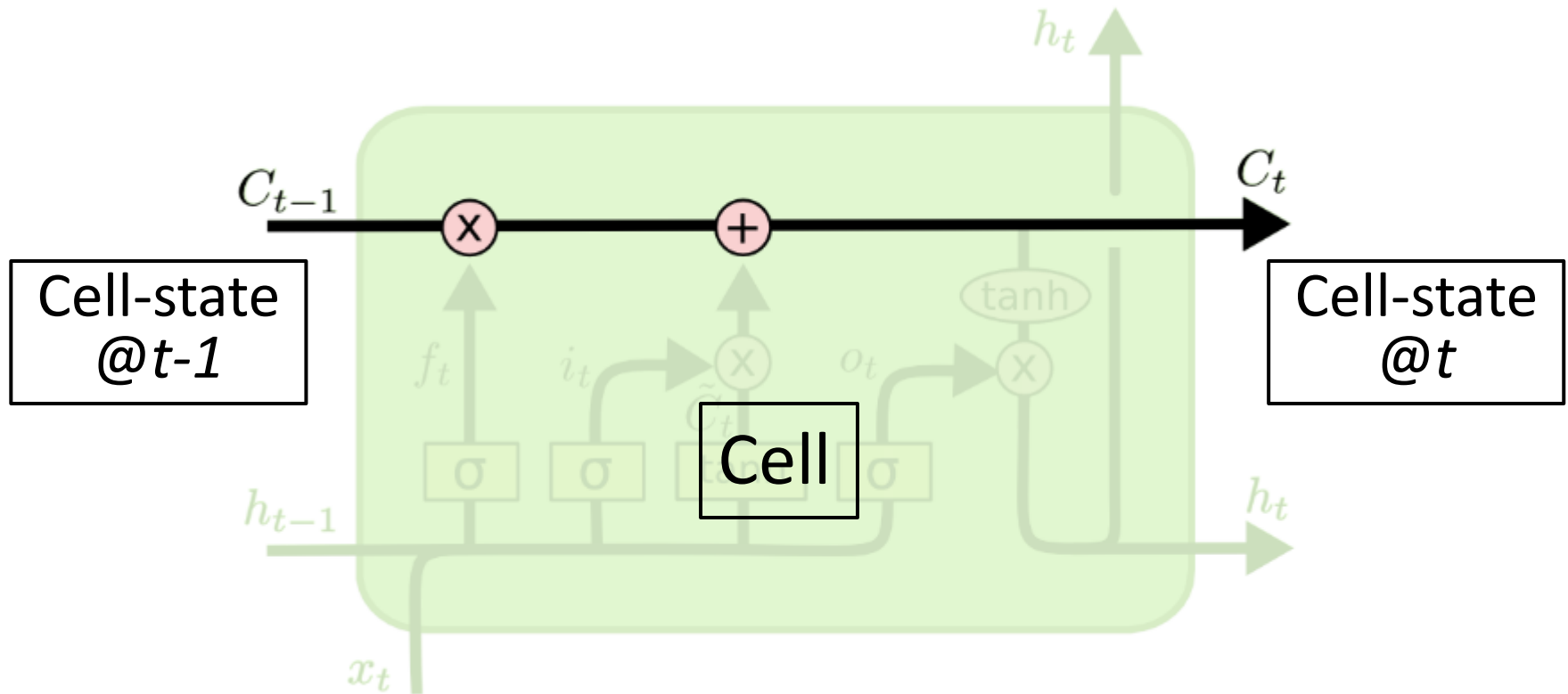
- In a standard RNN the repeating module has a simple structure. Example:



LSTMs

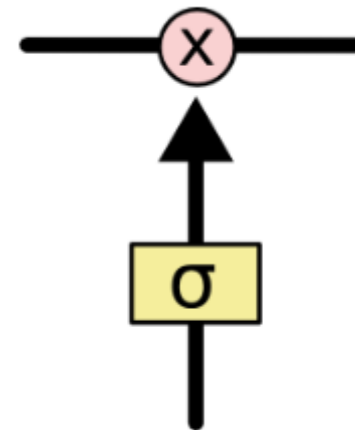


LSTM Memory / Cell State



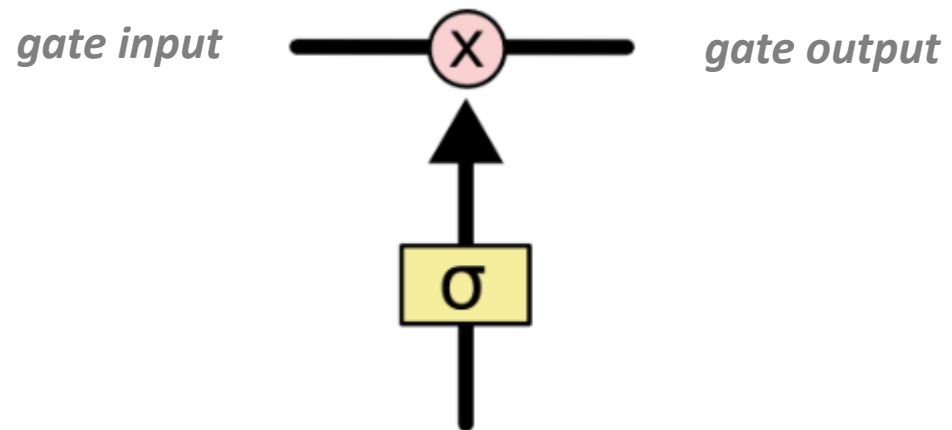
Gate

- Composed of a sigmoid neural net layer and a pointwise multiplication operation.



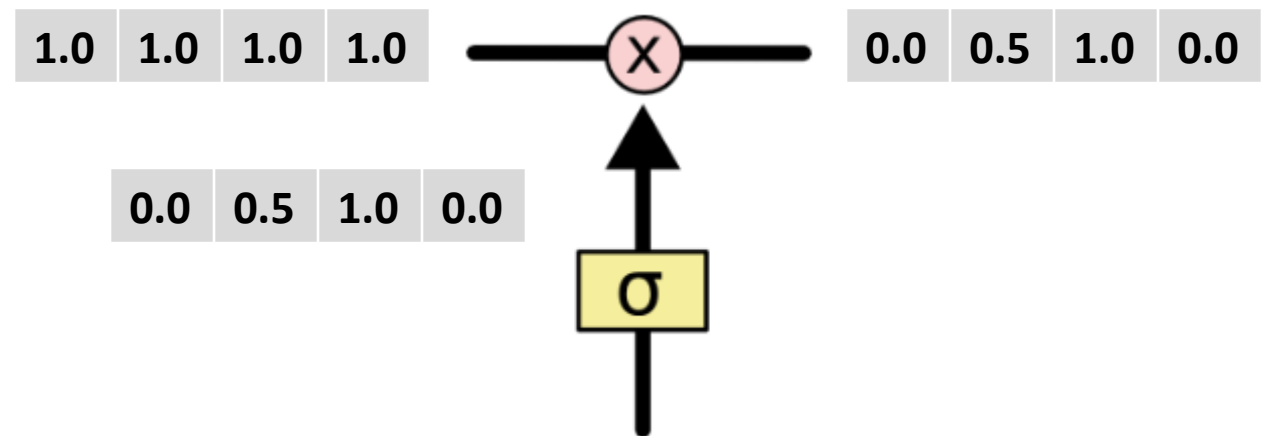
Gate

- sigmoid: outputs numbers between:
 - zero “let nothing through,” and
 - one, “let everything through!”
- Example:



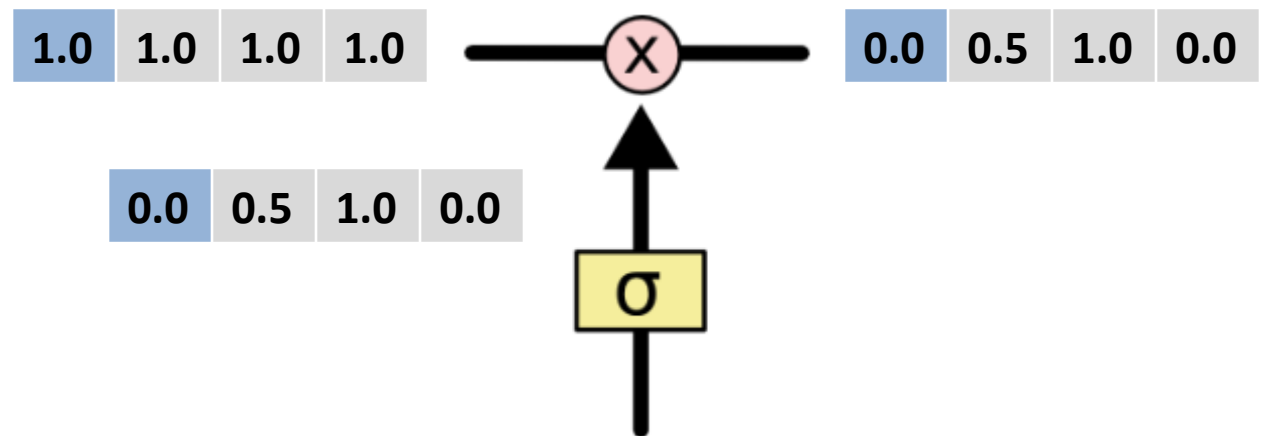
Gate

- sigmoid: outputs numbers between:
 - zero “let nothing through,” and
 - one, “let everything through!”
- Example:



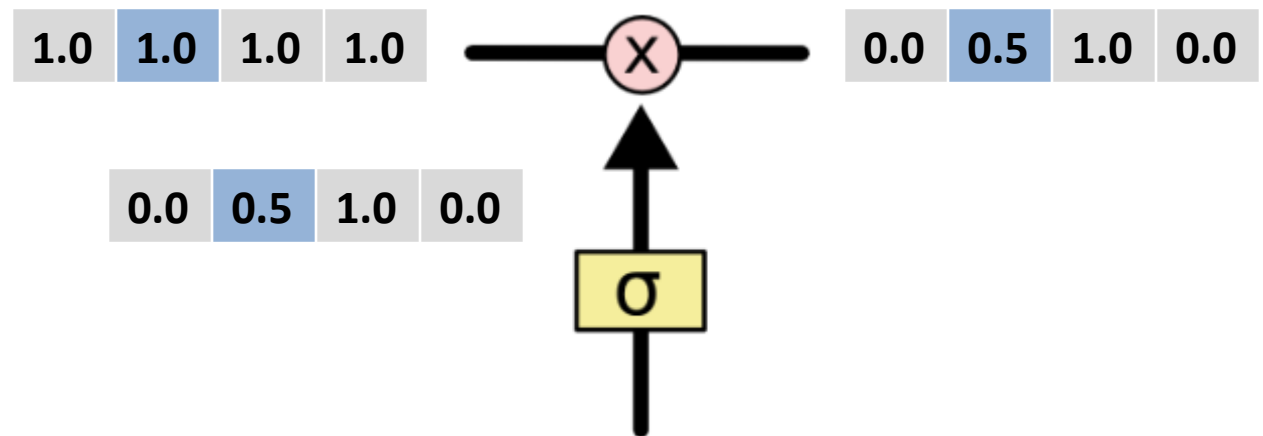
Gate

- sigmoid: outputs numbers between:
 - zero “let nothing through,” and
 - one, “let everything through!”
- Example:



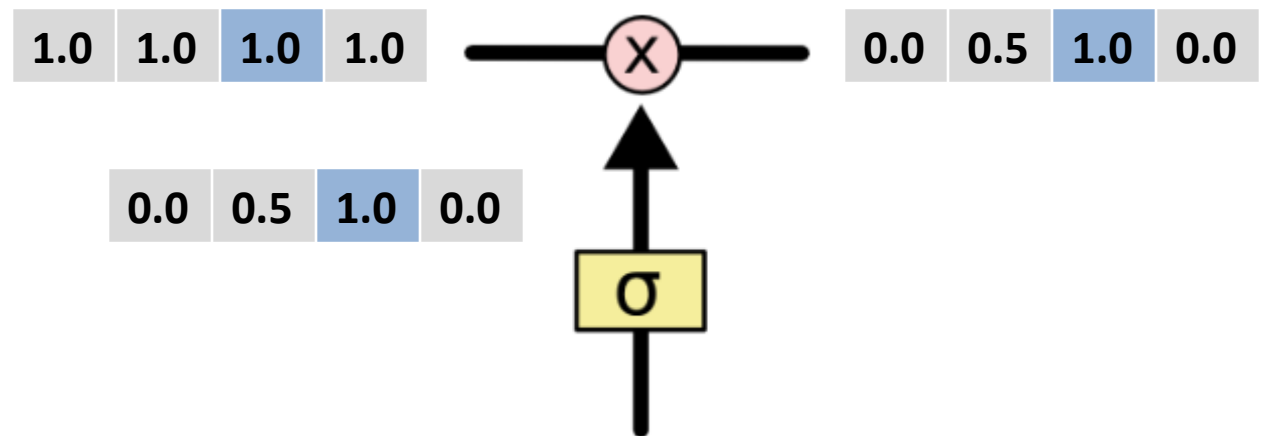
Gate

- sigmoid: outputs numbers between:
 - zero “let nothing through,” and
 - one, “let everything through!”
- Example:



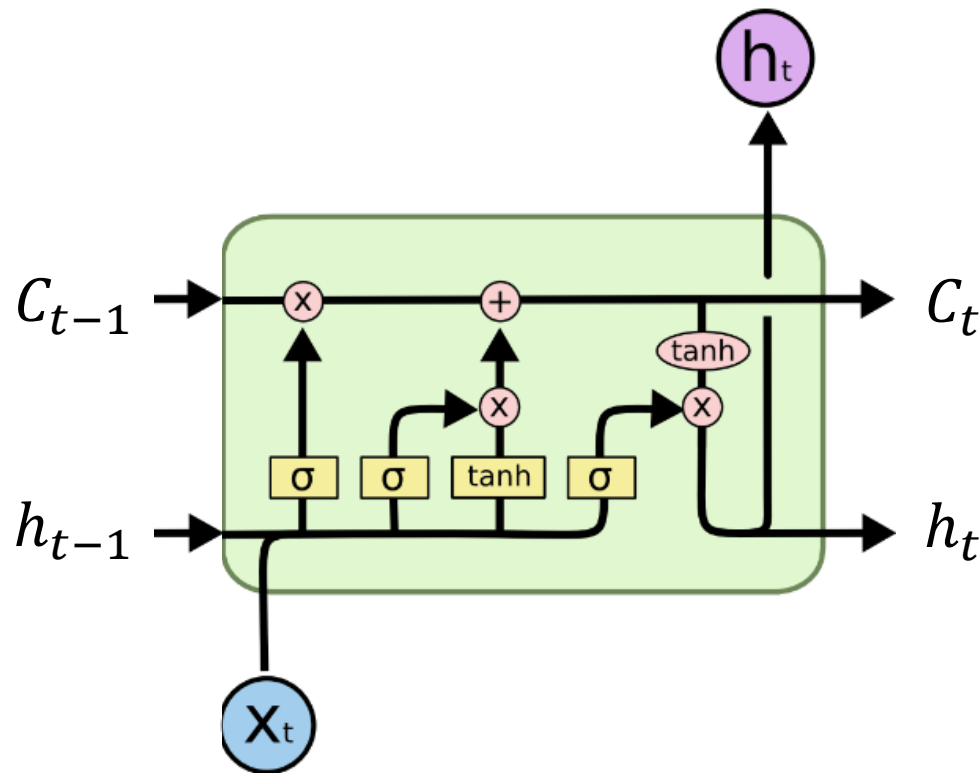
Gate

- sigmoid: outputs numbers between:
 - zero “let nothing through,” and
 - one, “let everything through!”
- Example:



LSTM Gates

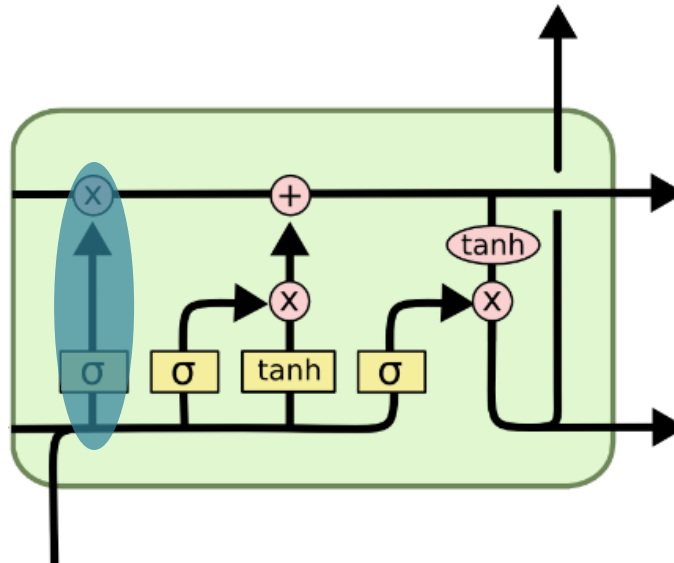
- An LSTM has three of these gates.



LSTM Gates

- An LSTM has three of these gates.

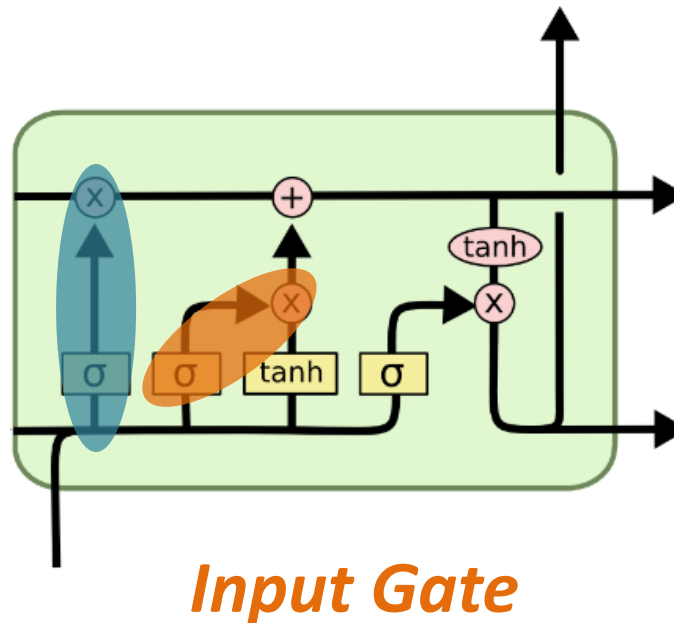
Forget gate



LSTM Gates

- An LSTM has three of these gates.

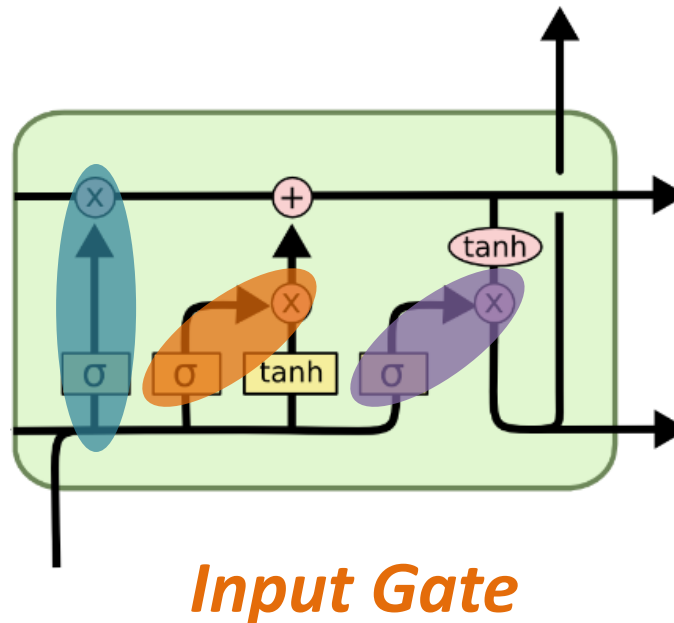
Forget gate



LSTM Gates

- An LSTM has three of these gates.

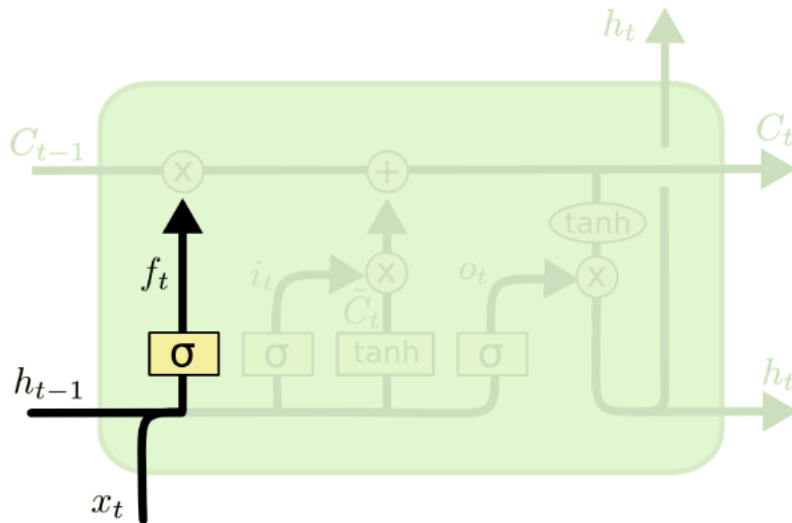
Forget gate



Output Gate

Step 1: Forget Gate

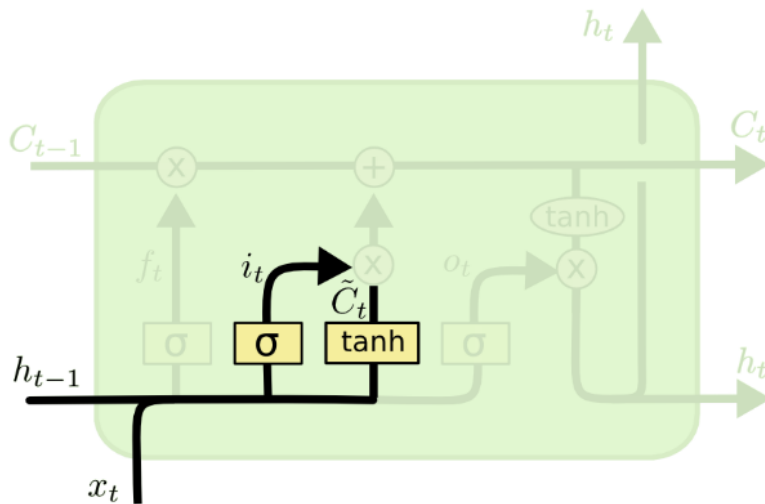
- Decide what to forget/ignore.



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

Step 2: Input Gate

- Propose new candidate memory \tilde{C}_t .
- Modulate the proposal.

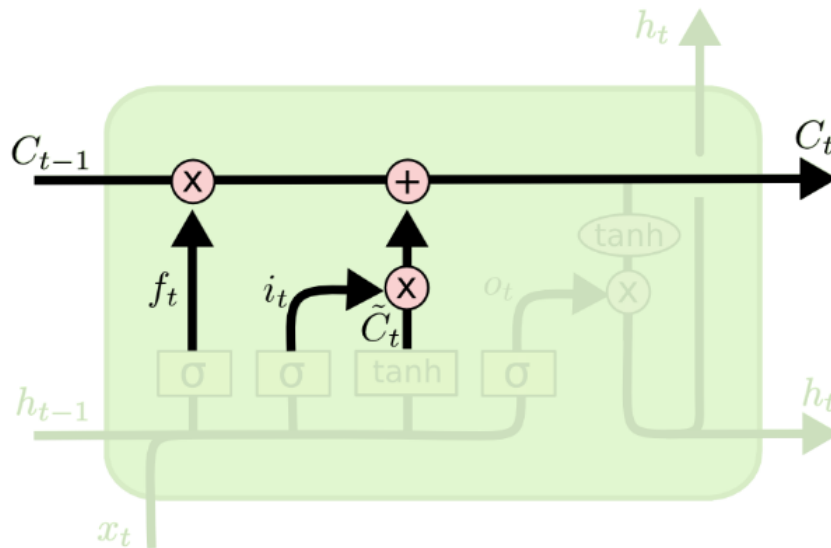


$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

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

Step 3: Memory Update

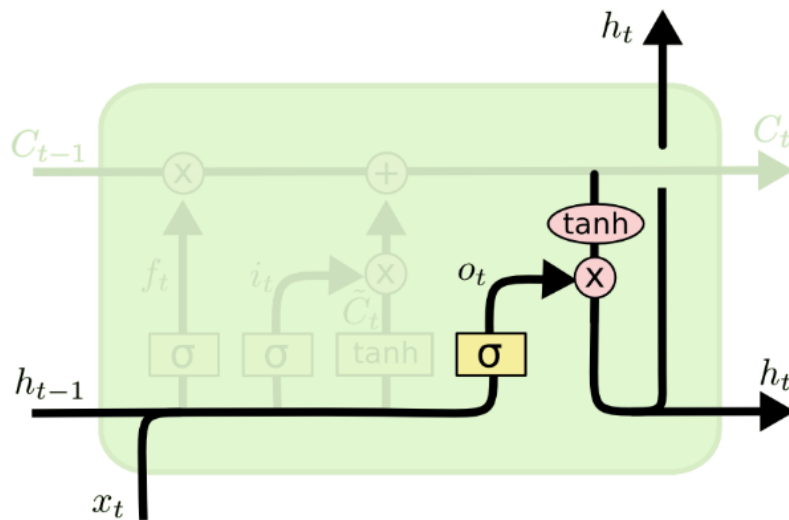
- Perform the memory update.



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

Step 4: Output Gate

- Decide what to output based on the memory.



$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

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

LSTM BP TT

Forward Pass: $h^t = o^t \odot \tanh(c^t)$

Given $\delta h^t = \frac{\partial E}{\partial h^t}$, find $\delta o^t, \delta c^t$

$$\begin{aligned}\frac{\partial E}{\partial o_i^t} &= \frac{\partial E}{\partial h_i^t} \cdot \frac{\partial h_i^t}{\partial o_i^t} \\ &= \delta h_i^t \cdot \tanh(c_i^t) \\ \therefore \delta o^t &= \delta h^t \odot \tanh(c^t)\end{aligned}$$

$$\begin{aligned}\frac{\partial E}{\partial c_i^t} &= \frac{\partial E}{\partial h_i^t} \cdot \frac{\partial h_i^t}{\partial c_i^t} \\ &= \delta h_i^t \cdot o_i^t \cdot (1 - \tanh^2(c_i^t)) \\ \therefore \delta c^t &= \delta h^t \odot o^t \odot (1 - \tanh^2(c^t))\end{aligned}$$

Forward Pass: $z^t = W \times I^t$

Given δz^t , find $\delta W^t, \delta h^{t-1}$

$$\begin{aligned}\delta I^t &= W^T \times \delta z^t \\ \text{As } I^t &= \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}, \\ \delta h^{t-1} &\text{ can be retrieved from } \delta I^t \\ \delta W^t &= \delta z^t \times (I^t)^T\end{aligned}$$

Forward Pass: $c^t = i^t \odot a^t + f^t \odot c^{t-1}$

Given $\delta c^t = \frac{\partial E}{\partial c^t}$, find $\delta i^t, \delta a^t, \delta f^t, \delta c^{t-1}$

$$\begin{aligned}\frac{\partial E}{\partial i_i^t} &= \frac{\partial E}{\partial c_i^t} \cdot \frac{\partial c_i^t}{\partial i_i^t} \\ &= \delta c_i^t \cdot a_i^t \\ \therefore \delta i^t &= \delta c^t \odot a^t\end{aligned}$$

$$\begin{aligned}\frac{\partial E}{\partial a_i^t} &= \frac{\partial E}{\partial c_i^t} \cdot \frac{\partial c_i^t}{\partial a_i^t} \\ &= \delta c_i^t \cdot i_i^t \\ \therefore \delta a^t &= \delta c^t \odot i^t\end{aligned}$$

$$\begin{aligned}\frac{\partial E}{\partial f_i^t} &= \frac{\partial E}{\partial c_i^t} \cdot \frac{\partial c_i^t}{\partial f_i^t} \\ &= \delta c_i^t \cdot c_i^{t-1} \\ \therefore \delta f^t &= \delta c^t \odot c^{t-1}\end{aligned}$$

$$\begin{aligned}\frac{\partial E}{\partial c_i^{t-1}} &= \frac{\partial E}{\partial c_i^t} \cdot \frac{\partial c_i^t}{\partial c_i^{t-1}} \\ &= \delta c_i^t \cdot f_i^t \\ \therefore \delta c^{t-1} &= \delta c^t \odot f^t\end{aligned}$$

Forward Pass: $z^t = \begin{bmatrix} \hat{a}^t \\ \hat{i}^t \\ \hat{f}^t \\ \hat{o}^t \end{bmatrix} = W \times I^t$

Given $\delta a^t, \delta i^t, \delta f^t, \delta o^t$, find δz^t

$$\begin{aligned}\delta \hat{a}^t &= \delta a^t \odot (1 - \tanh^2(\hat{a}^t)) \\ \delta \hat{i}^t &= \delta i^t \odot i^t \odot (1 - i^t) \\ \delta \hat{f}^t &= \delta f^t \odot f^t \odot (1 - f^t) \\ \delta \hat{o}^t &= \delta o^t \odot o^t \odot (1 - o^t) \\ \delta z^t &= [\delta \hat{a}^t, \delta \hat{i}^t, \delta \hat{f}^t, \delta \hat{o}^t]^T\end{aligned}$$

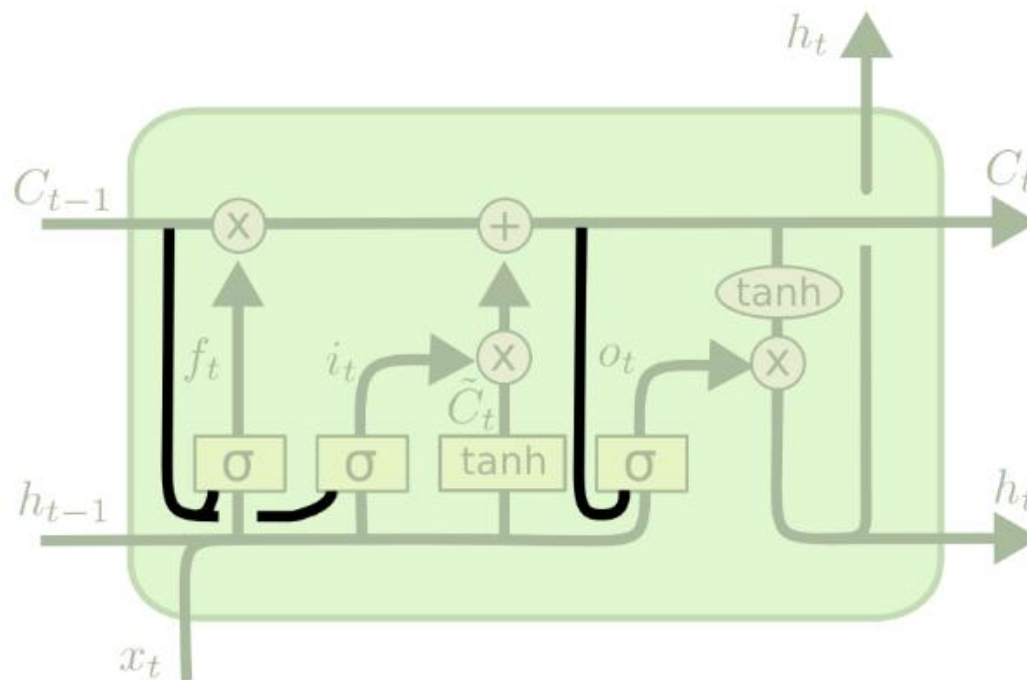
If input x has T time-steps, i.e. $x = [x^1, x^2, \dots, x^T]$, then

$$\delta W = \sum_{t=1}^T \delta W^t$$

W is then updated using an appropriate Stochastic Gradient Descent solver.

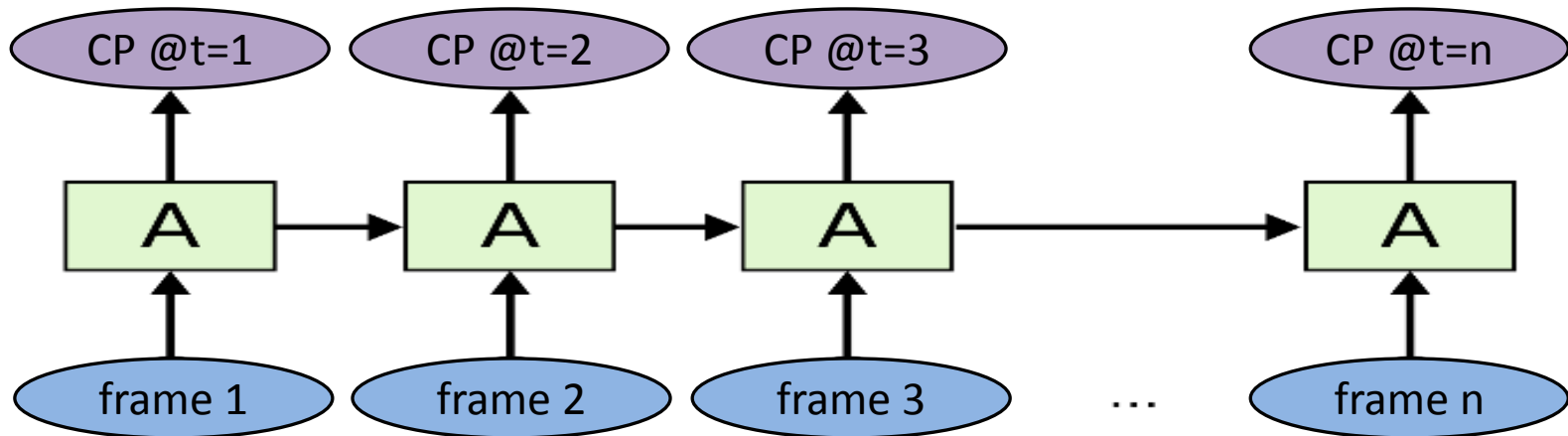
LSTM Variants

- Example: PeepHole Model

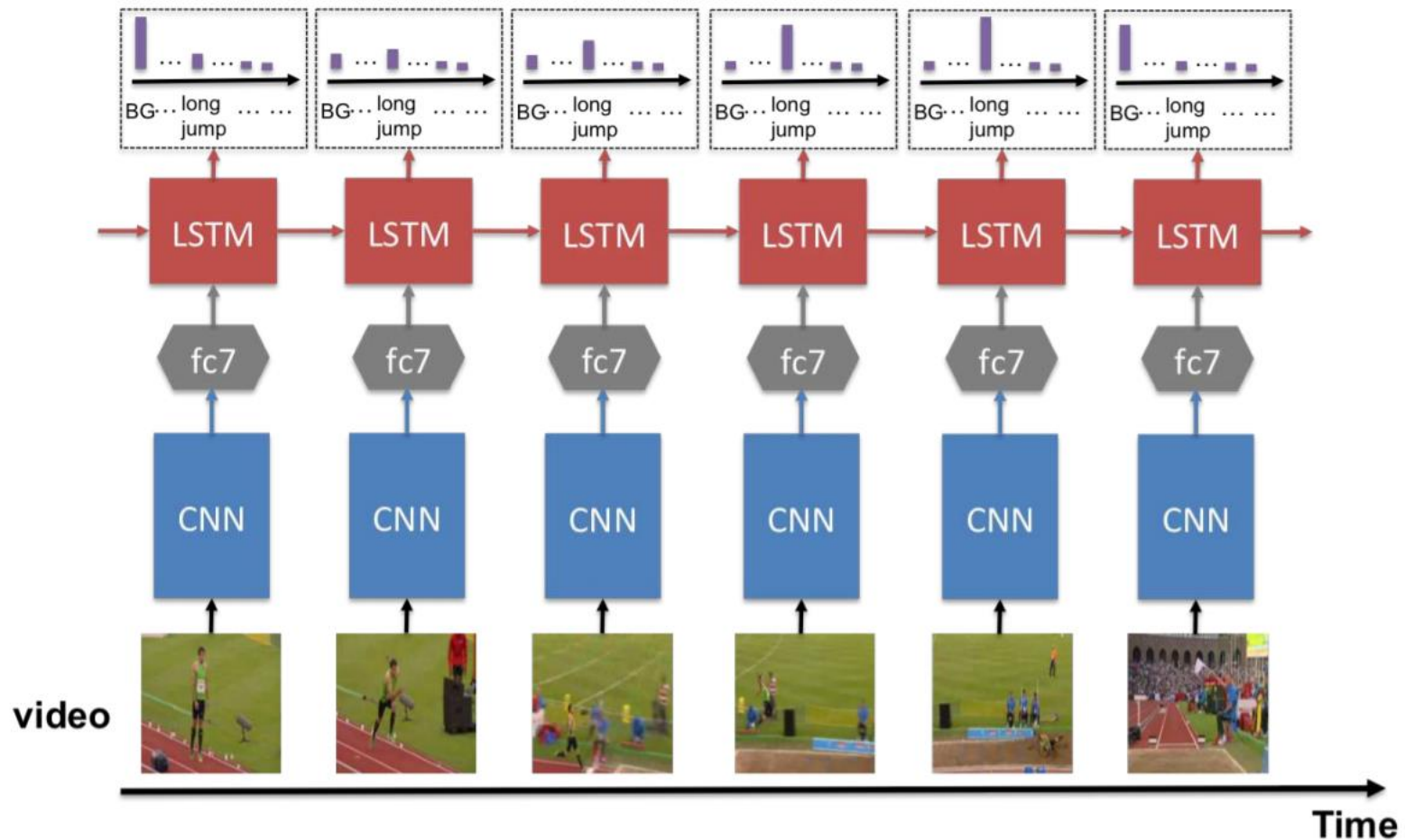


Application 1: Video Classification

- CP: conditional class probability
- $\text{frame } i$ could be a feature describing frame i , example: CNN feature

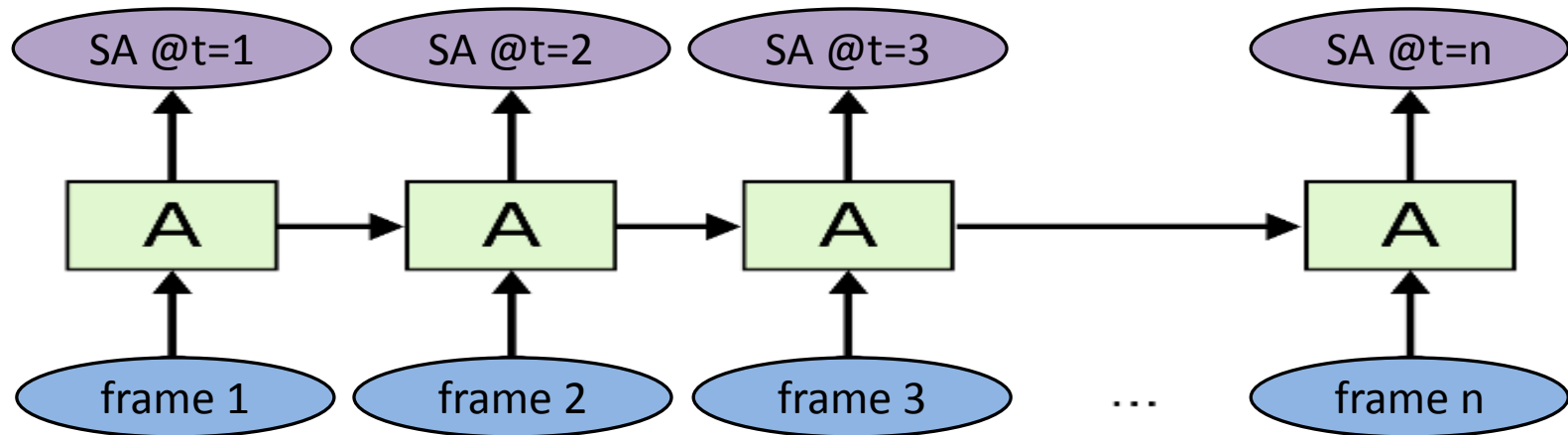


Application 1: Video Classification



Application 2: Self-Driving Cars

- SA: steering angle
- *frame i* could be a feature describing frame i , example: 3D-CNN feature



Application 2: Self-Driving Cars

DeepTesla



Application 2: Self-Driving Cars

- Udacity winning team: Team Komanda
 - x_t : 3D convolution of image sequence
 - h_t : steering angle, speed, torque



DeepTraffic

- Check this from MIT:

[DeepTraffic](http://selfdrivingcars.mit.edu/deeptraffic/)