Day 2 Lecture 6

Recurrent Neural Networks

Xavier Giró-i-Nieto

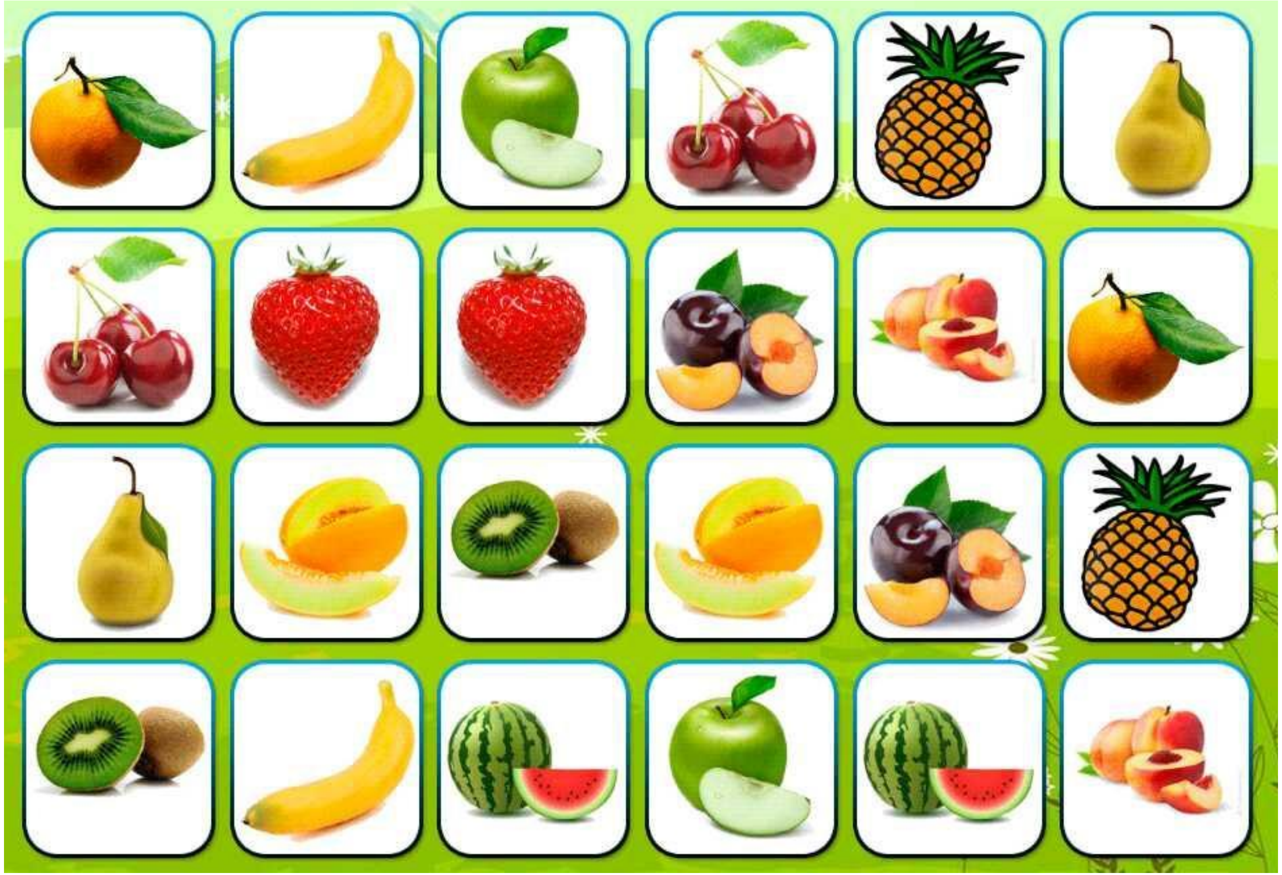
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Department of Signal Theory and Communications
Image Processing Group

Acknowledgments

Santi Pascual

TALP Research Center

General idea

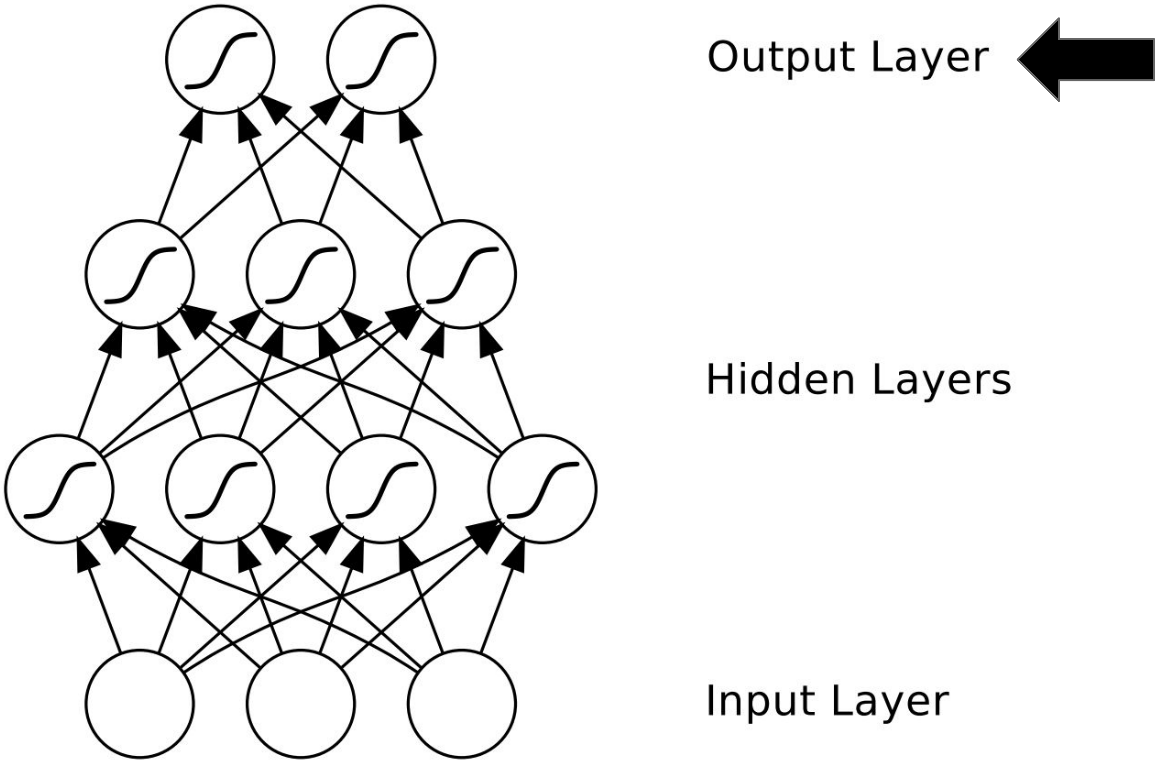


ConvNet
(or CNN)

General idea

ConvNet
(or CNN)

Multilayer Perceptron

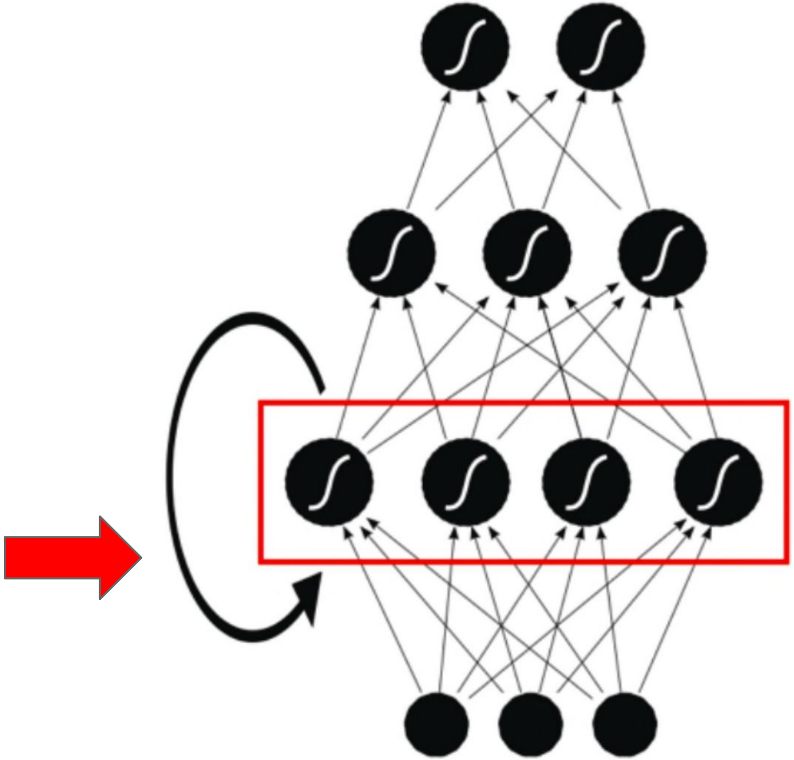


The output depends ONLY on the current input.

Alex Graves, [“Supervised Sequence Labelling with Recurrent Neural Networks”]

Recurrent Neural Network (RNN)

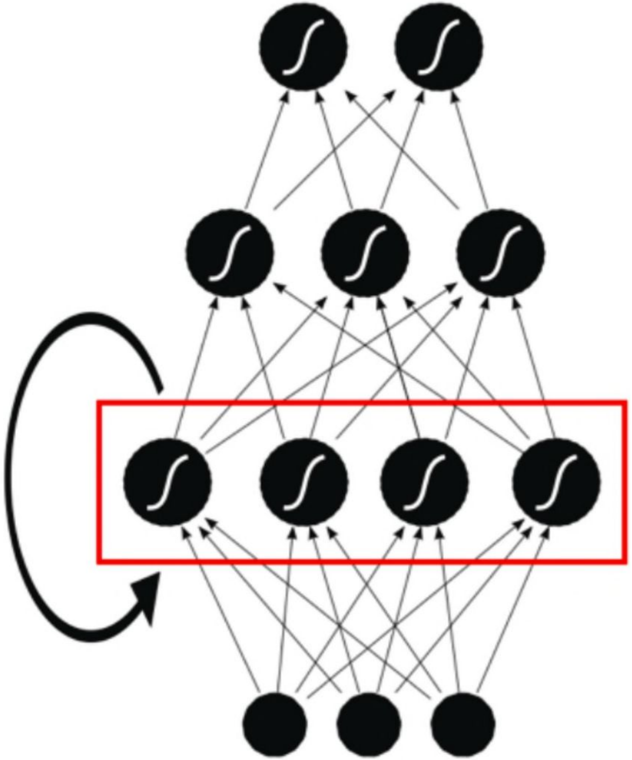
The hidden layers and the
output depend from previous
states of the hidden layers



Alex Graves, [“Supervised Sequence Labelling with Recurrent Neural Networks”]

Recurrent Neural Network (RNN)

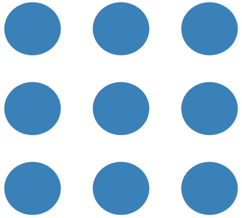
The hidden layers and the output depend from previous states of the hidden layers



Alex Graves, [“Supervised Sequence Labelling with Recurrent Neural Networks”]

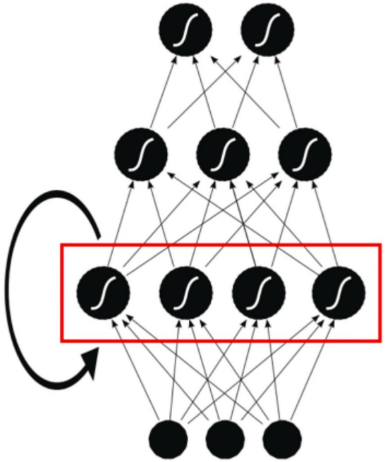
Recurrent Neural Network (RNN)

Front View

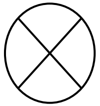


Rotation 90°

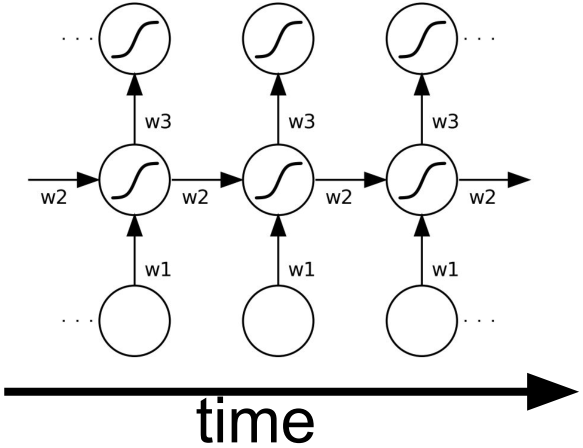
Side View



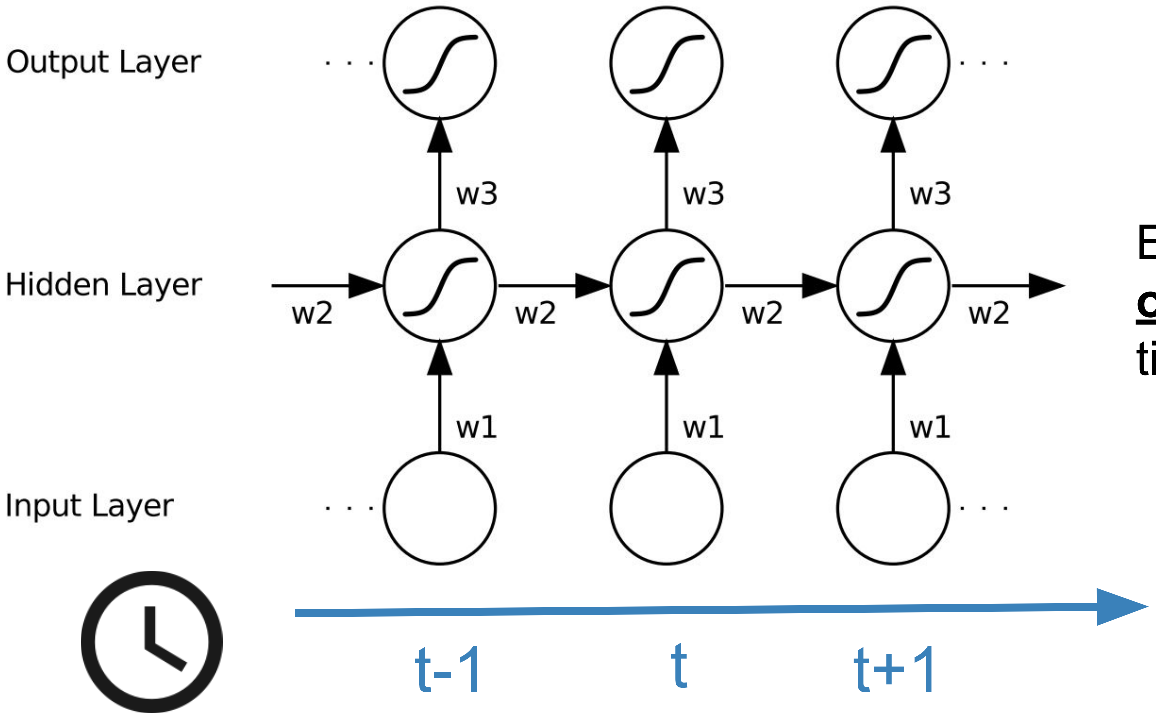
time



Rotation 90°



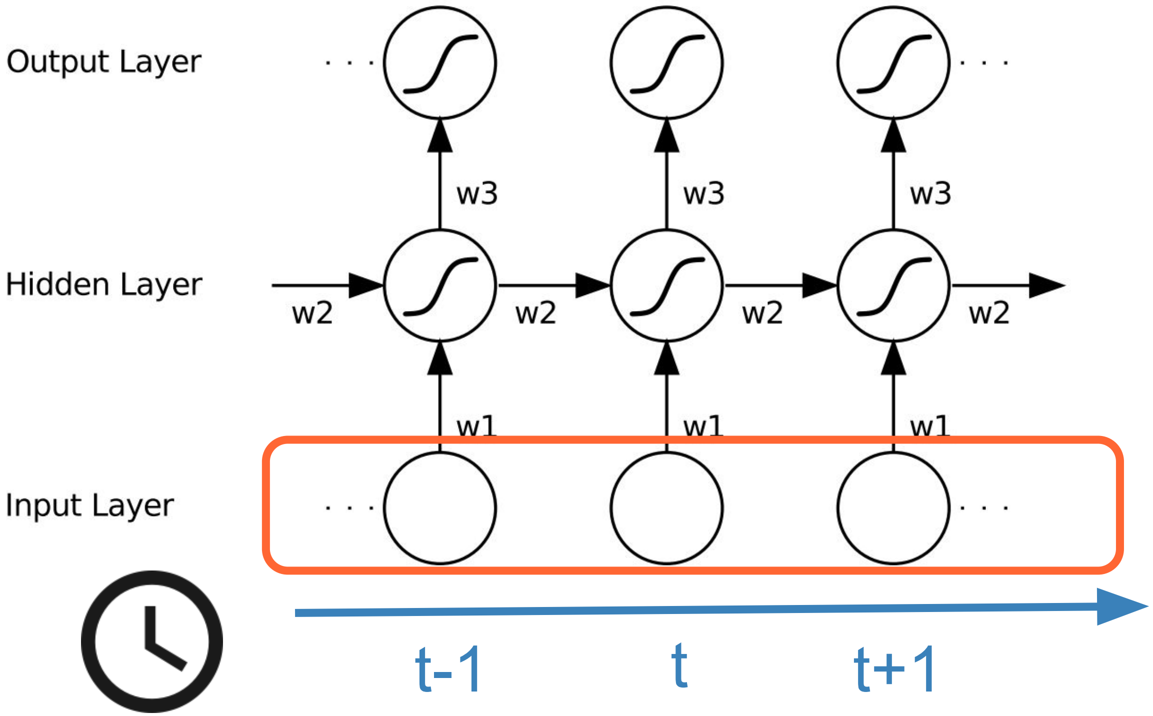
Recurrent Neural Networks (RNN)



Each node represents **a layer of neurons** at a single timestep.

Alex Graves, [“Supervised Sequence Labelling with Recurrent Neural Networks”]

Recurrent Neural Networks (RNN)

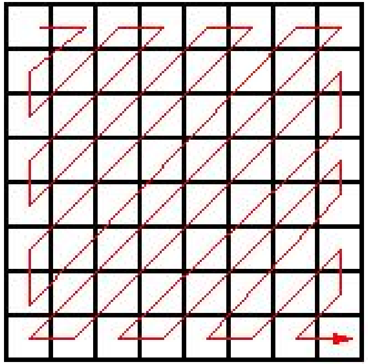


The input is a **SEQUENCE x(t)** of any length.

Alex Graves, [“Supervised Sequence Labelling with Recurrent Neural Networks”]

Recurrent Neural Networks (RNN)

Common visual sequences:

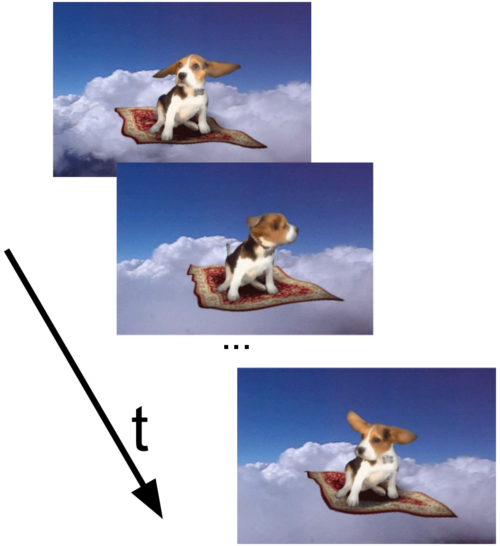


The input is a **SEQUENCE x(t)** of any length.

Still image + Spatial scan
(zigzag, snake)

Recurrent Neural Networks (RNN)

Common visual sequences:

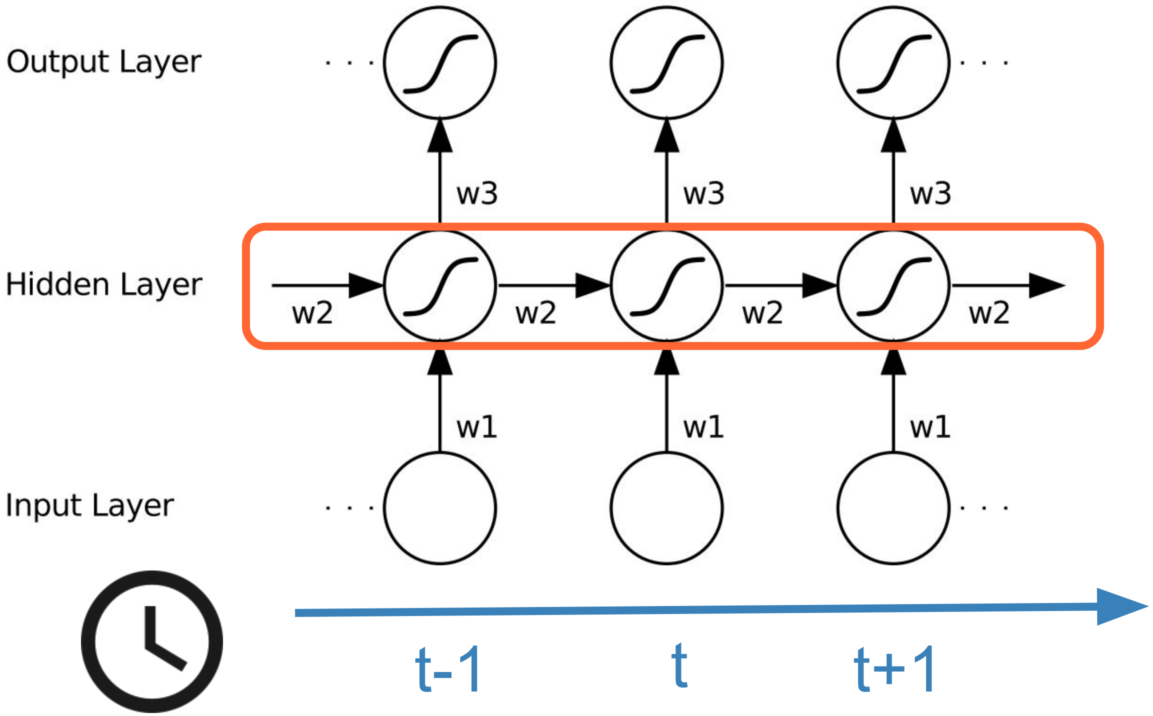


The input is a **<u>SEQUENCE</u> x(t)** of any length.

Video

Temporal sampling

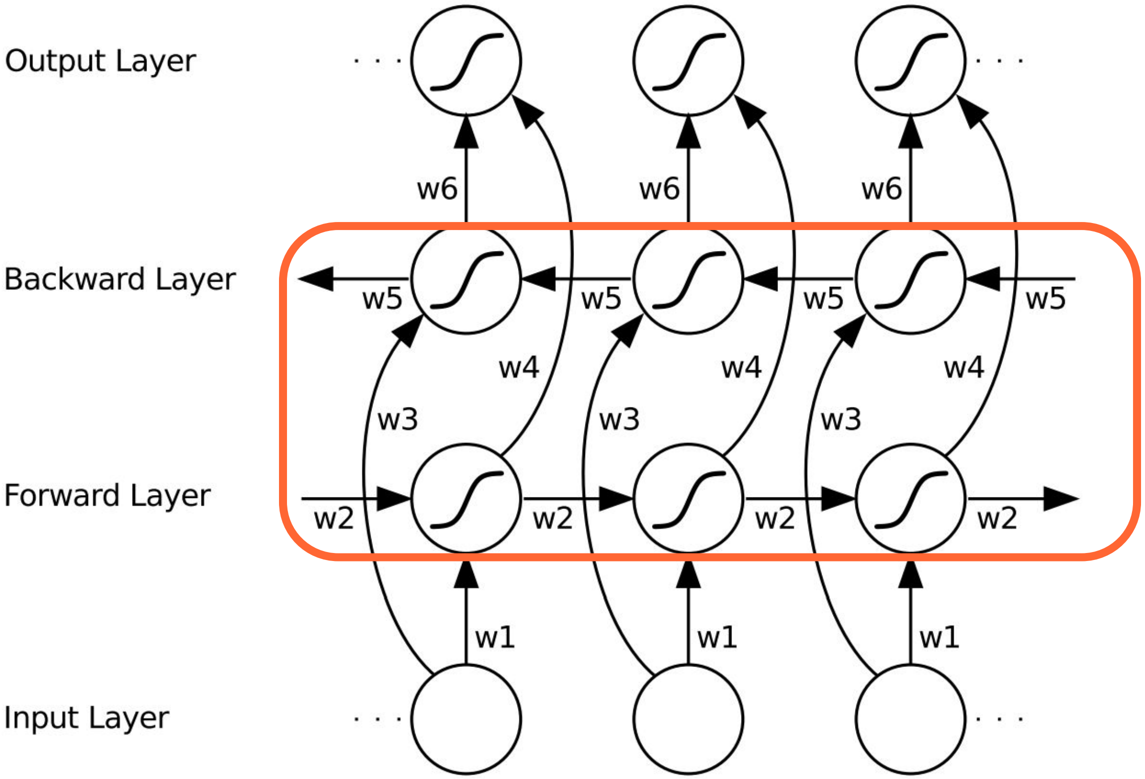
Recurrent Neural Networks (RNN)



Must learn temporally shared weights w2; in addition to w1 & w3.

Alex Graves, [“Supervised Sequence Labelling with Recurrent Neural Networks”]

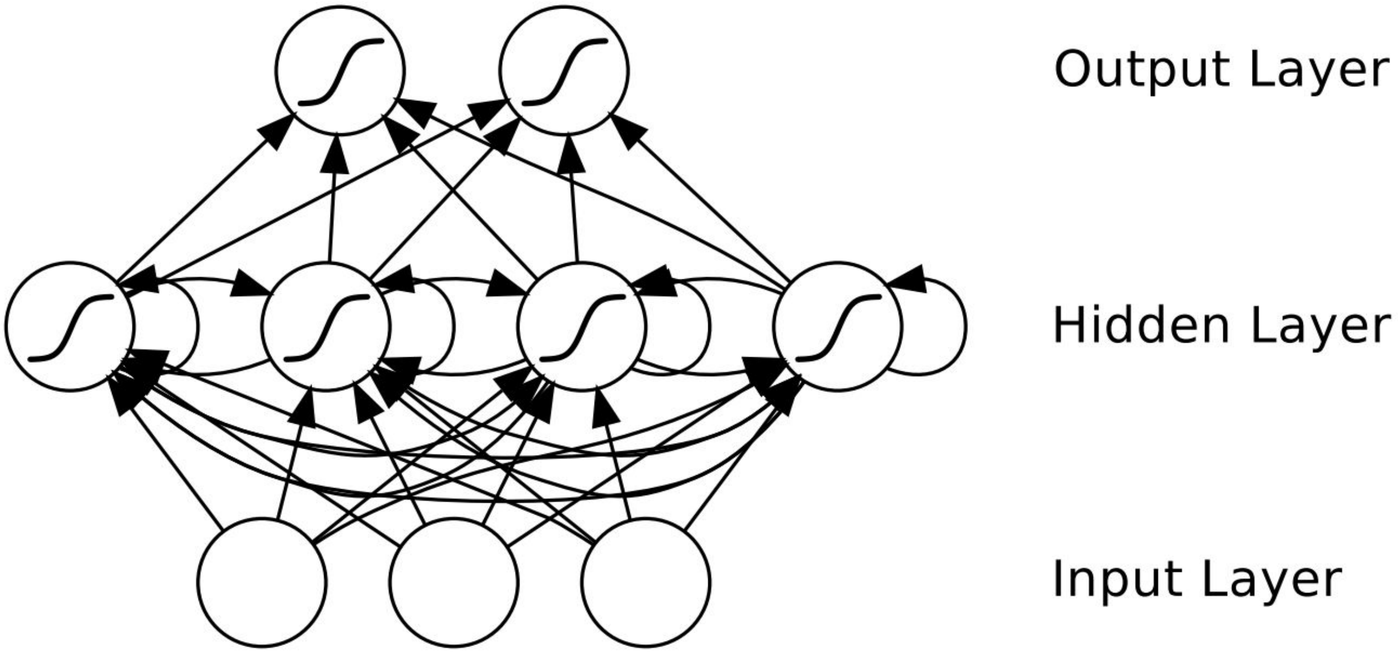
Bidirectional RNN (BRNN)



Must learn weights w2, w3, w4 & w5; in addition to w1 & w6.

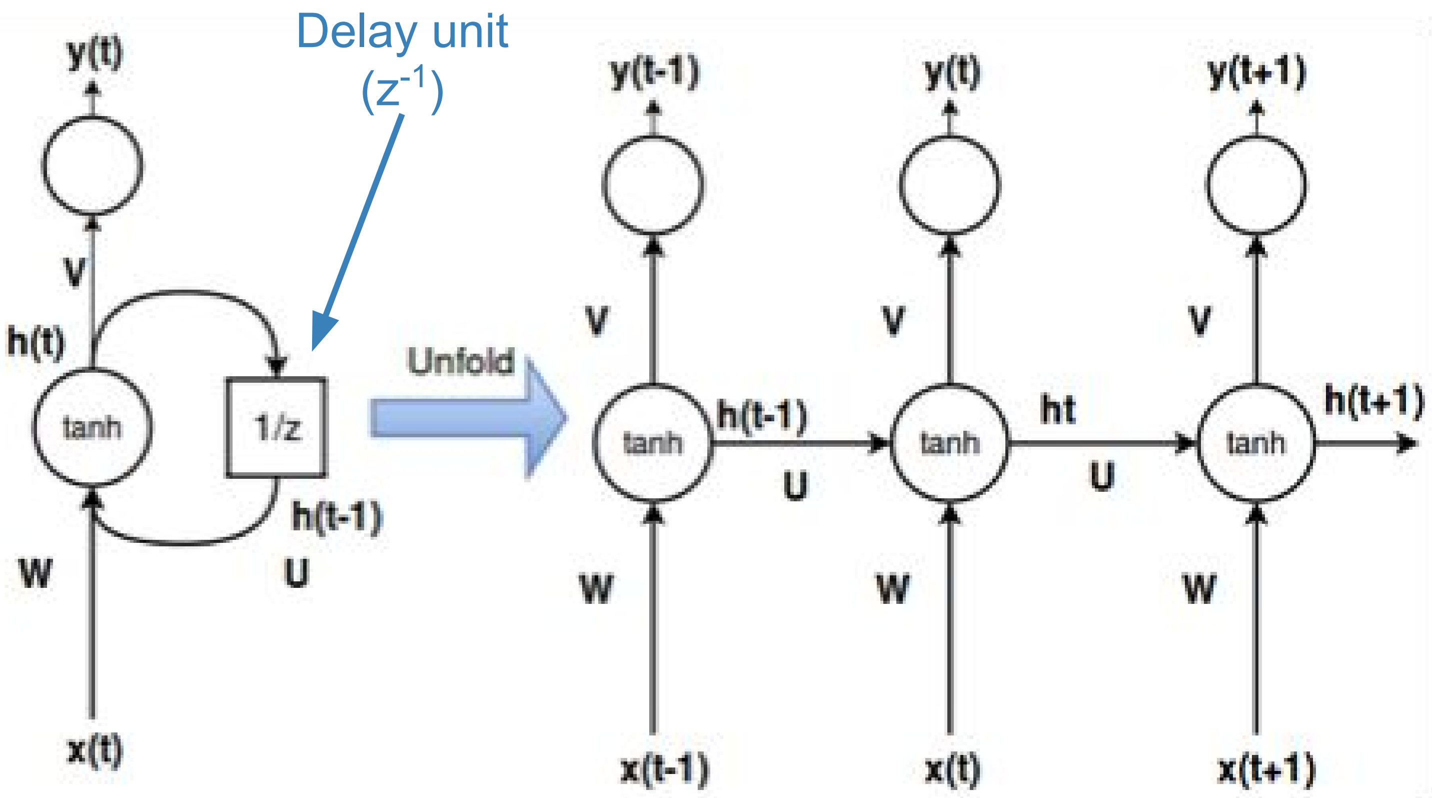
Alex Graves, [“Supervised Sequence Labelling with Recurrent Neural Networks”]

Bidirectional RNN (BRNN)



Alex Graves, [“Supervised Sequence Labelling with Recurrent Neural Networks”]

Formulation: One hidden layer

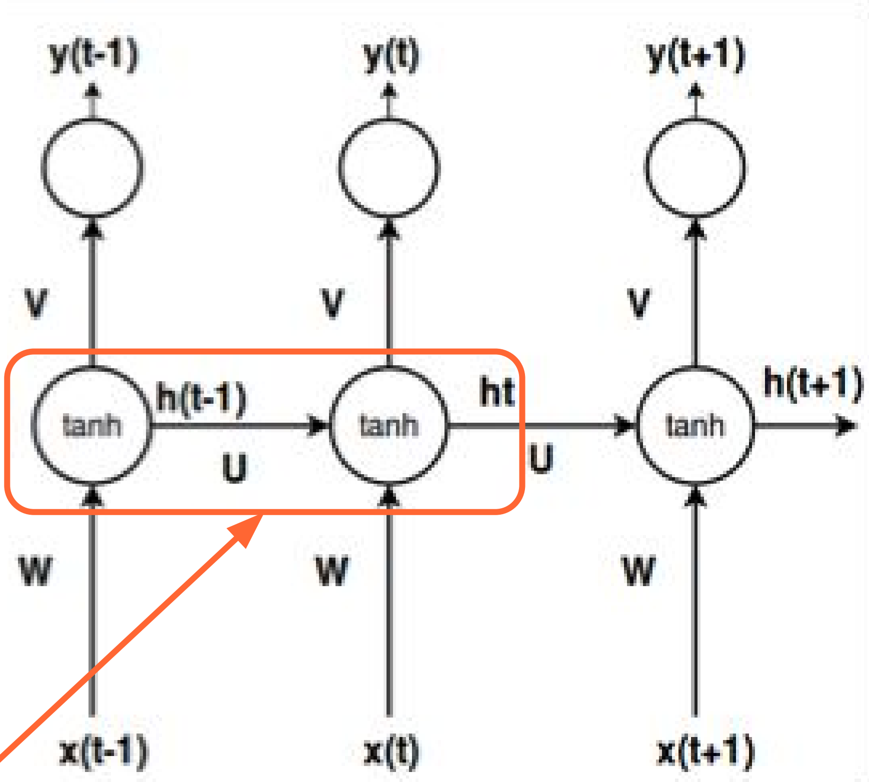


Formulation: Single recurrence

$$\mathbf{y}_t = f(\mathbf{V} \cdot \mathbf{h}_t + \mathbf{b}_v)$$

$$\mathbf{h}_t = g(\mathbf{W} \cdot \mathbf{x}_t + \boxed{\mathbf{U} \cdot \mathbf{h}_{t-1}} + \mathbf{b}_h)$$

$$\mathbf{x}_t \in \mathbb{R}^n$$



One-time
Recurrence

Formulation: Multiple recurrences

One time-step recurrence

$$\mathbf{h}_t = g(\mathbf{W} \cdot \mathbf{x}_t + \boxed{\mathbf{U} \cdot \mathbf{h}_{t-1}} + \mathbf{b}_h)$$

Recurrence

T time steps recurrences

$$\mathbf{h}_t = g(\mathbf{W} \cdot \mathbf{x}_t + \boxed{\mathbf{U} \cdot g(\cdots g(\mathbf{W} \cdot \mathbf{x}_{t-T} + \mathbf{U} \cdot \mathbf{h}_{t-T} + \mathbf{b}_h) \cdots)} + \mathbf{b}_h)$$

RNN problems

Long term memory vanishes because of the T nested multiplications by U.

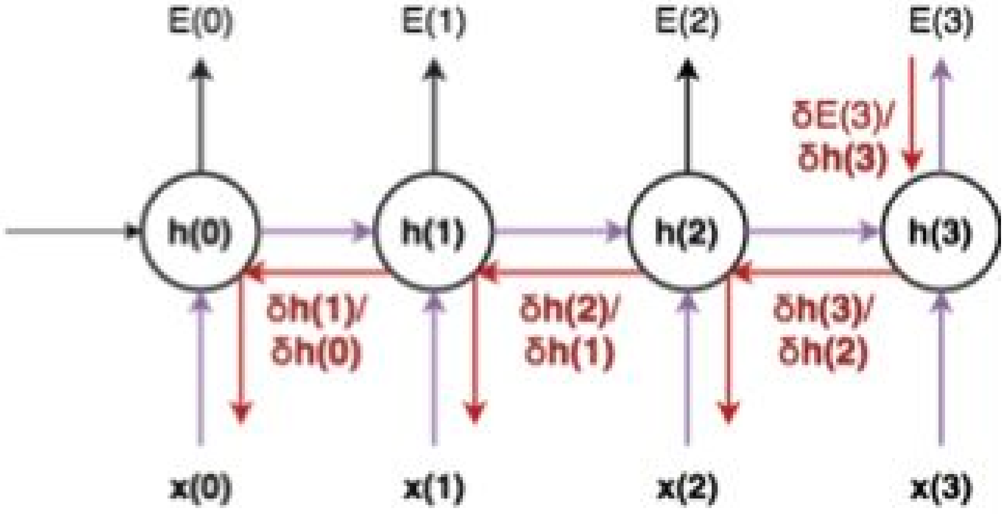
$$\mathbf{h}_t = g(\mathbf{W} \cdot \mathbf{x}_t + \boxed{\mathbf{U}} \cdot g(\boxed{\cdots} g(\mathbf{W} \cdot \mathbf{x}_{t-T} + \boxed{\mathbf{U}} \cdot \mathbf{h}_{t-T} + \mathbf{b}_h) \cdots) + \mathbf{b}_h)$$

...

RNN problems

During training, gradients may explode or vanish because of temporal depth.

Example: Back-propagation in time with 3 steps.



Long Short-Term Memory (LSTM)

Stanford NLP Group
@stanfordnlp

Seguint

LSTMs are really mainstream now … just referenced in the @Apple #WWDC2016 keynote for iOS QuickType auto-completion

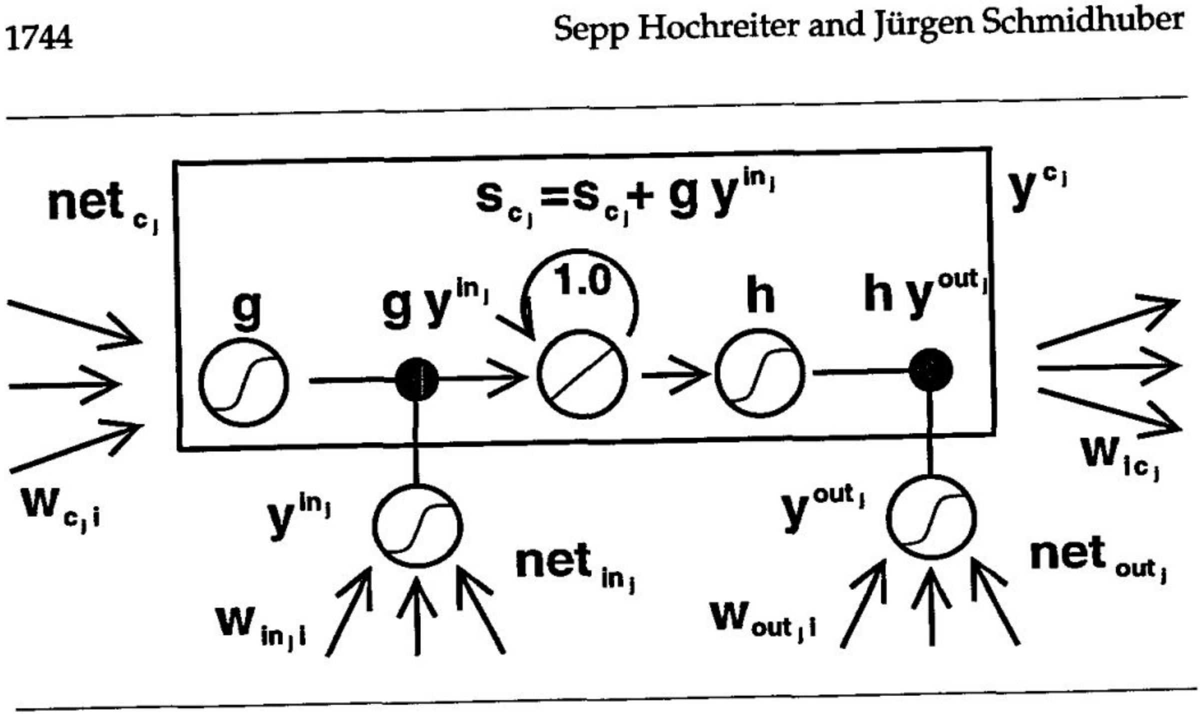
Mostra-ho traduït

RETUITS	AGRADA A
49	60

20:03 - 13 juny 2016

49 60

Long Short-Term Memory (LSTM)



Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9, no. 8 (1997): 1735-1780.

Long Short-Term Memory (LSTM)

Based on a standard RNN whose neuron activates with *tanh*...

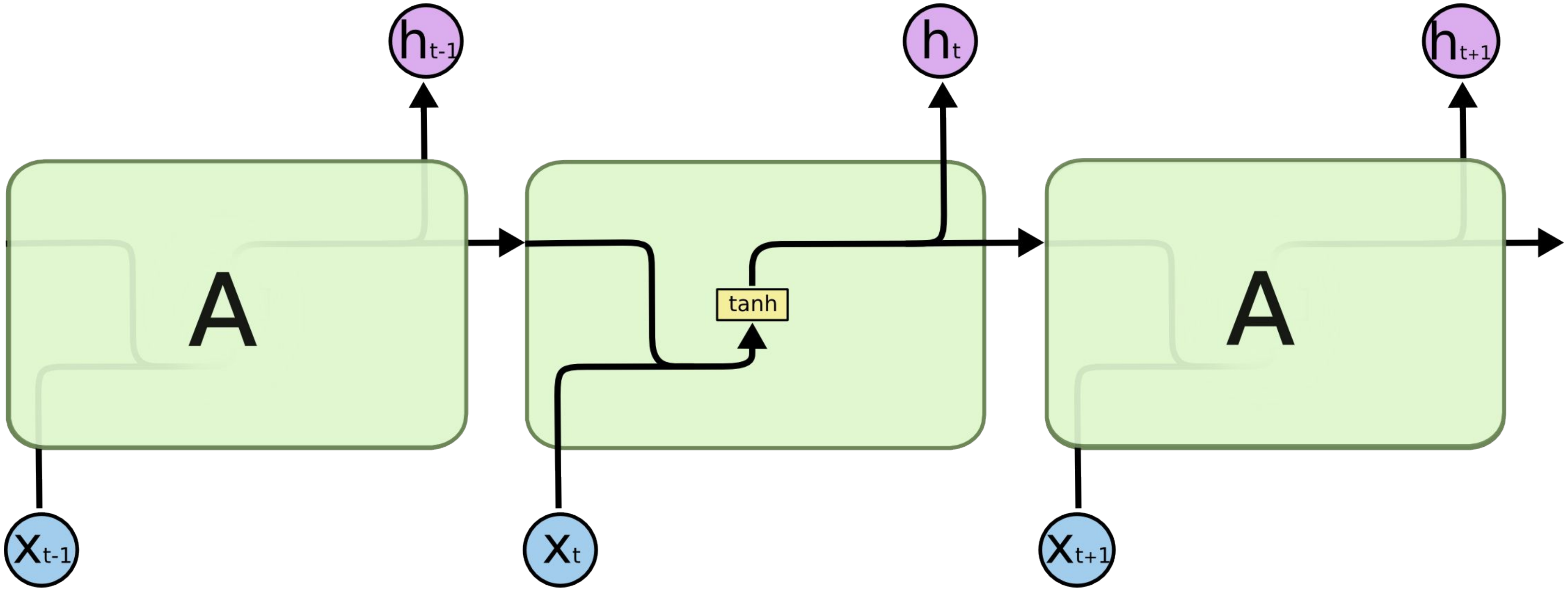


Figure: Cristopher Olah, "Understanding LSTM Networks" (2015)

Long Short-Term Memory (LSTM)

C_t is the cell state, which flows through the entire chain...

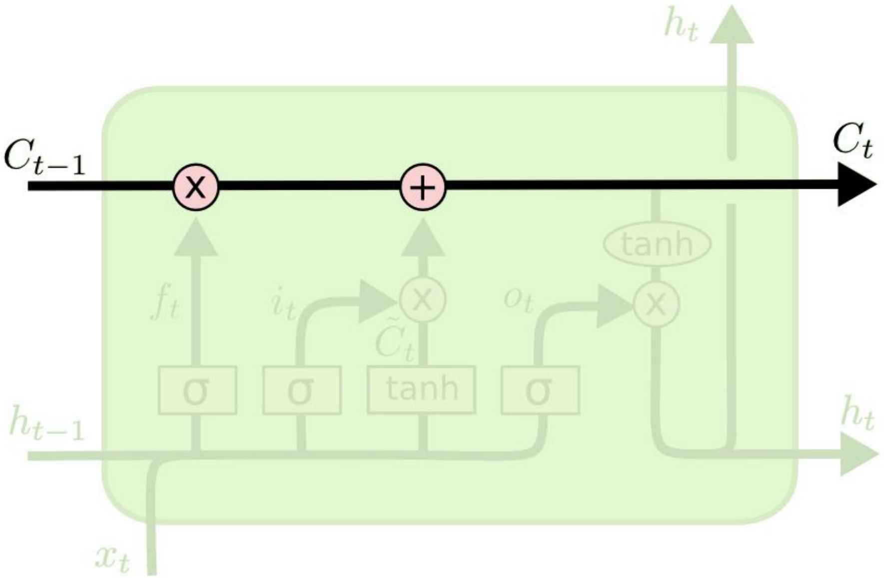


Figure: Cristopher Olah, [“Understanding LSTM Networks”]() (2015)

Long Short-Term Memory (LSTM)

...and is updated with a **sum** instead of a product. This avoid memory vanishing and exploding/vanishing backprop gradients.

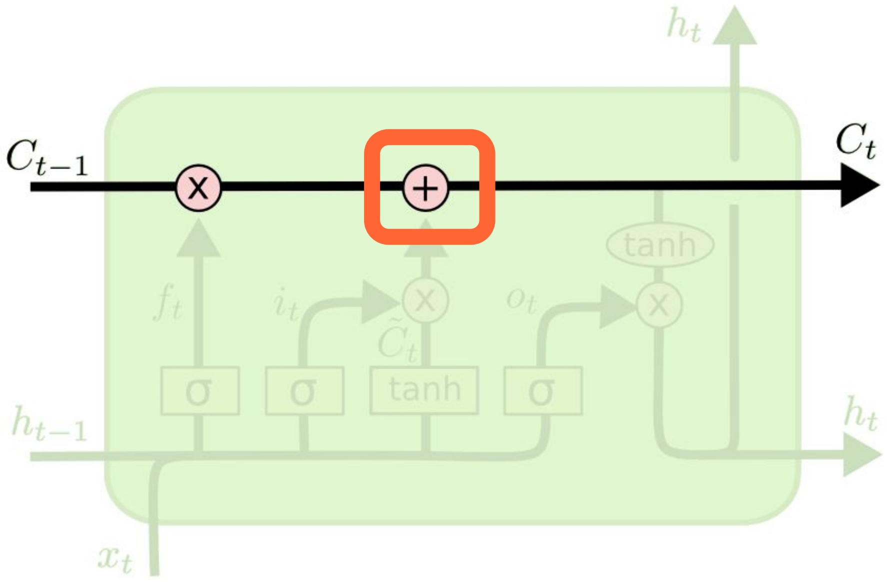
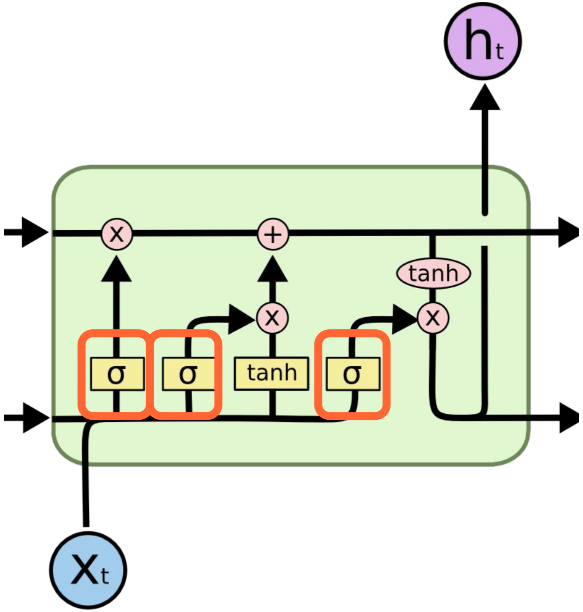
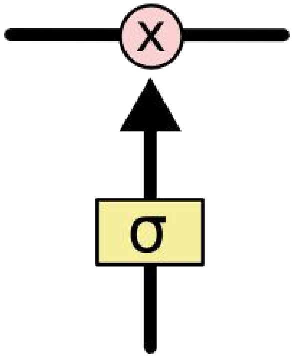


Figure: Cristopher Olah, "Understanding LSTM Networks" (2015)

Long Short-Term Memory (LSTM)

Three **gates** are governed by *sigmoid* units (btw [0,1]) define the control of in & out information..





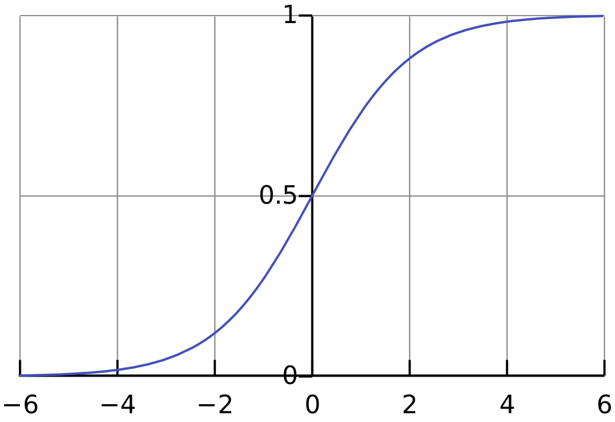
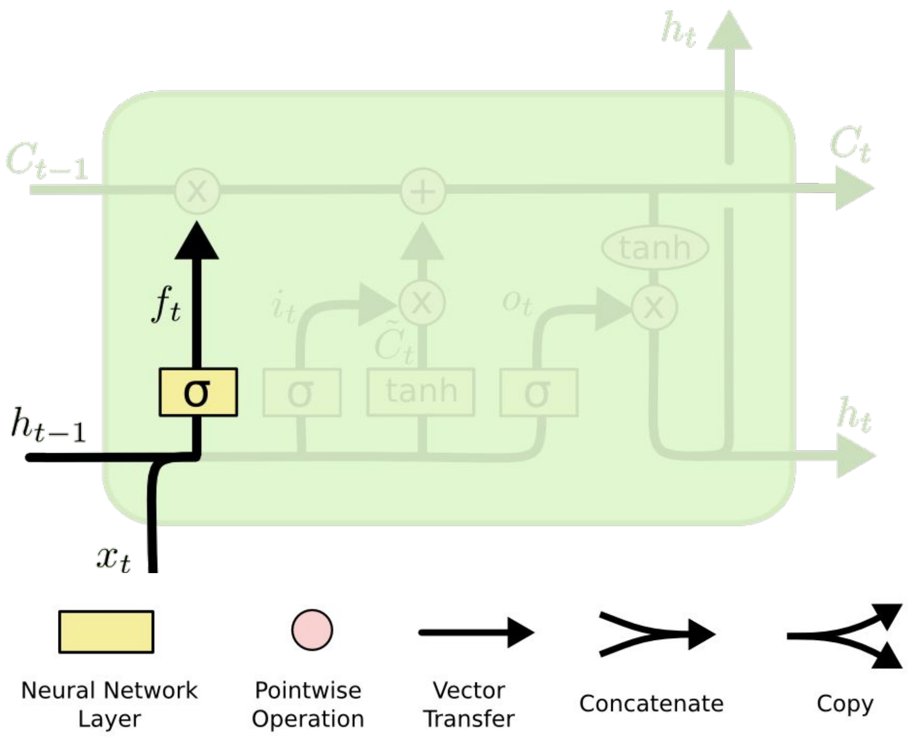


Figure: Cristopher Olah, "Understanding LSTM Networks" (2015)

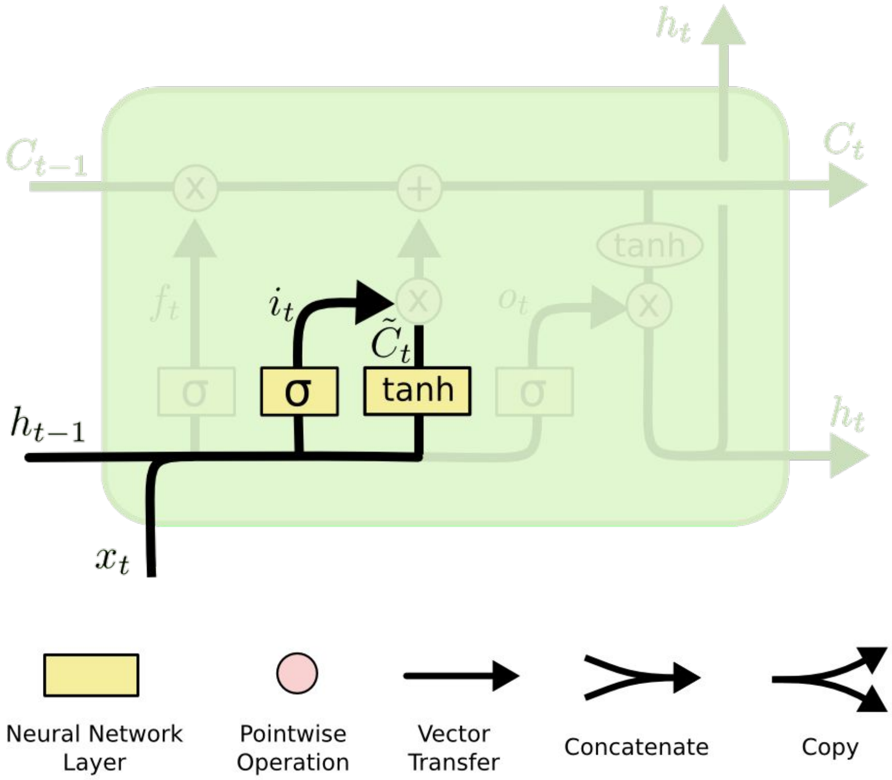
Long Short-Term Memory (LSTM)



Forget Gate:

$$f_t = \sigma\left(W_f \cdot \underbrace{[h_{t-1}, x_t]}_{\text{Concatenate}} + b_f\right)$$

Long Short-Term Memory (LSTM)



Input Gate Layer

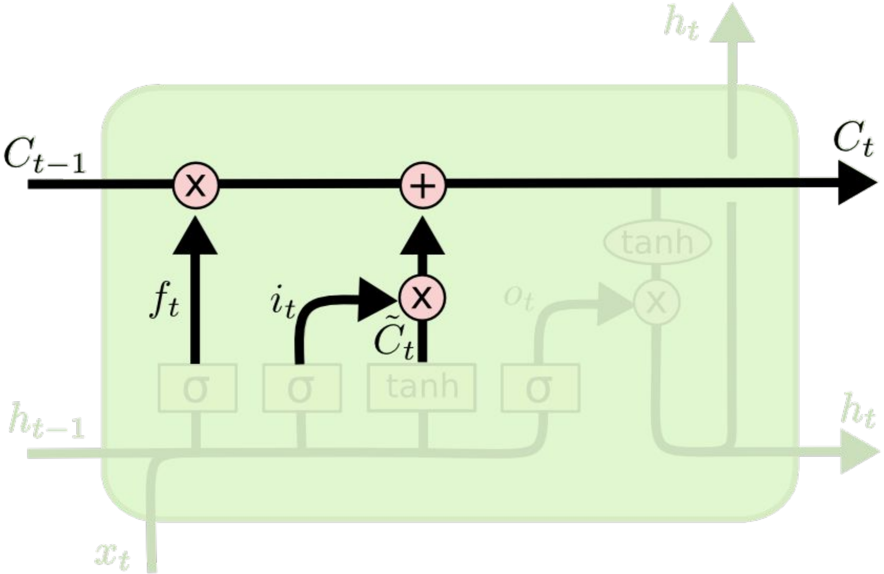
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

New contribution to cell state

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

Figure: Cristopher Olah, "Understanding LSTM Networks" (2015) / Slide: Alberto Montes

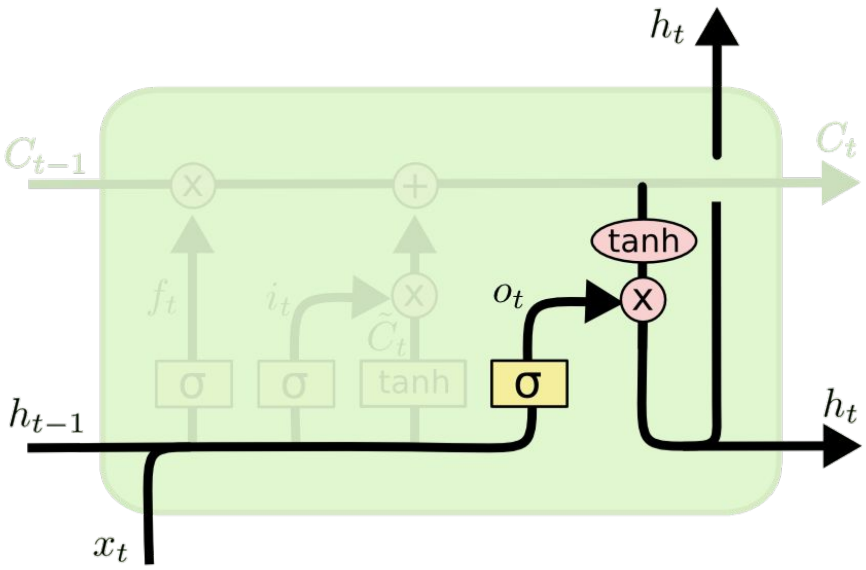
Long Short-Term Memory (LSTM)



Update Cell State (memory):

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

Long Short-Term Memory (LSTM)



Output Gate Layer

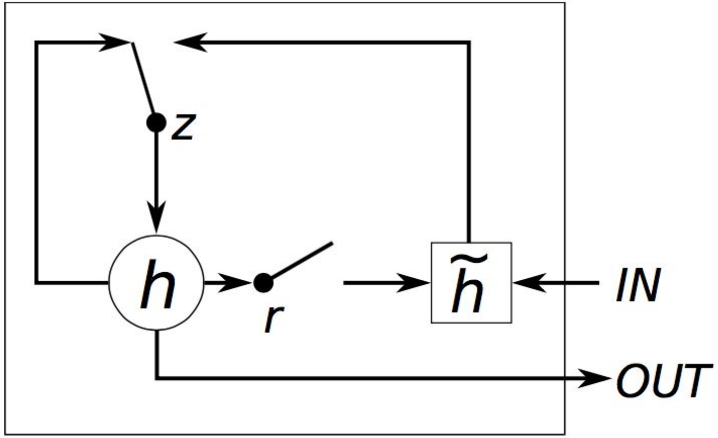
$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

Output to next layer

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

Gated Recurrent Unit (GRU)

Similar performance as LSTM with less computation.



$$u_i = \sigma\left(W^{(u)}x_i + U^{(u)}h_{i-1} + b^{(u)}\right) \quad (1)$$

$$r_i = \sigma\left(W^{(r)}x_i + U^{(r)}h_{i-1} + b^{(r)}\right) \quad (2)$$

$$\tilde{h}_i = \tanh\left(Wx_i + r_i \circ Uh_{i-1} + b^{(h)}\right) \quad (3)$$

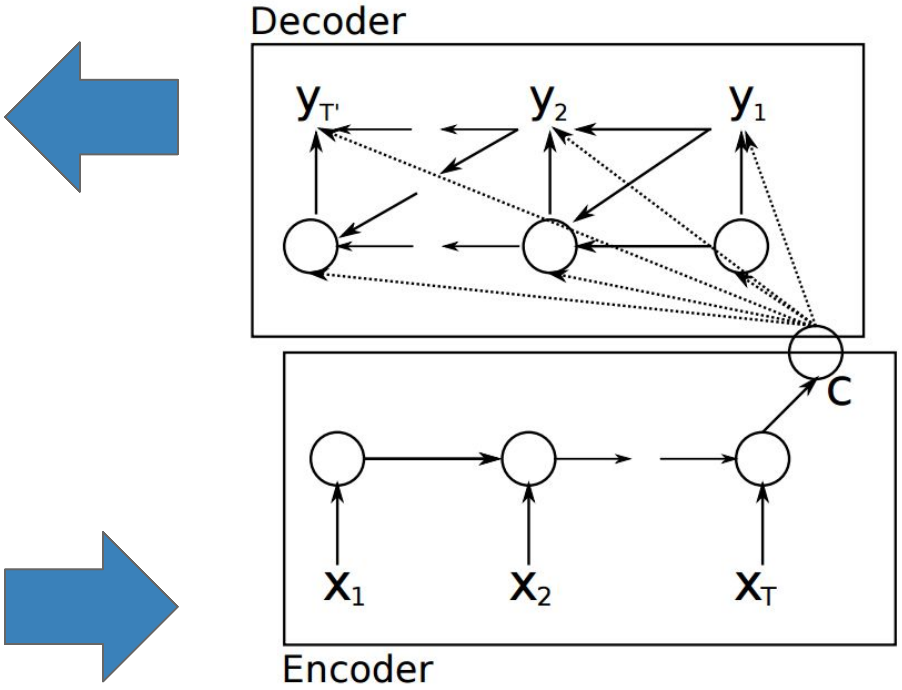
$$h_i = u_i \circ \tilde{h}_i + (1 - u_i) \circ h_{i-1} \quad (4)$$

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

Applications: Machine Translation

Language OUT

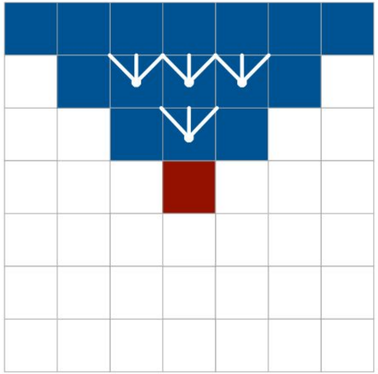
Language IN



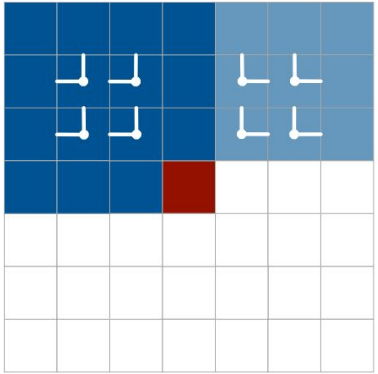
Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. ["Learning phrase representations using RNN encoder-decoder for statistical machine translation."]() arXiv preprint arXiv:1406.1078 (2014).

Applications: Image Classification

RowLSTM



Diagonal BiLSTM

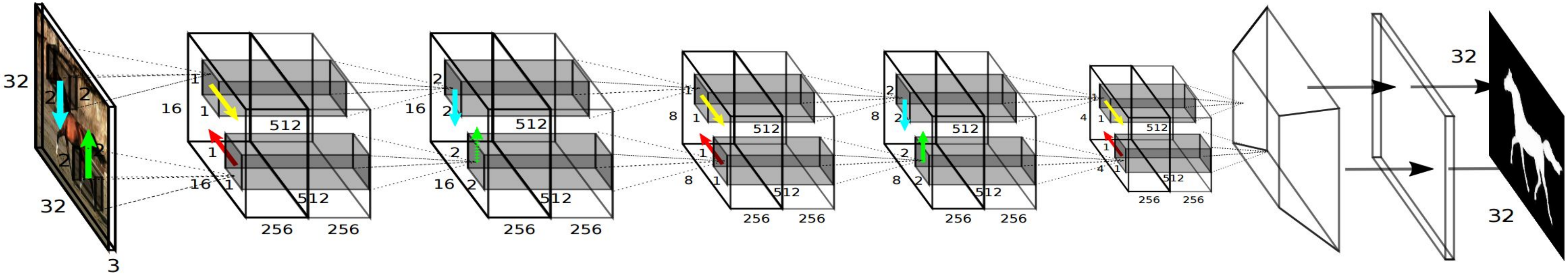


Classification MNIST

Model	NLL Test
DBM 2hl [1]:	≈ 84.62
DBN 2hl [2]:	≈ 84.55
NADE [3]:	88.33
EoNADE 2hl (128 orderings) [3]:	85.10
EoNADE-5 2hl (128 orderings) [4]:	84.68
DLGM [5]:	≈ 86.60
DLGM 8 leapfrog steps [6]:	≈ 85.51
DARN 1hl [7]:	≈ 84.13
MADE 2hl (32 masks) [8]:	86.64
DRAW [9]:	≤ 80.97
Diagonal BiLSTM (1 layer, $h = 32$):	**80.75**
Diagonal BiLSTM (7 layers, $h = 16$):	**79.20**

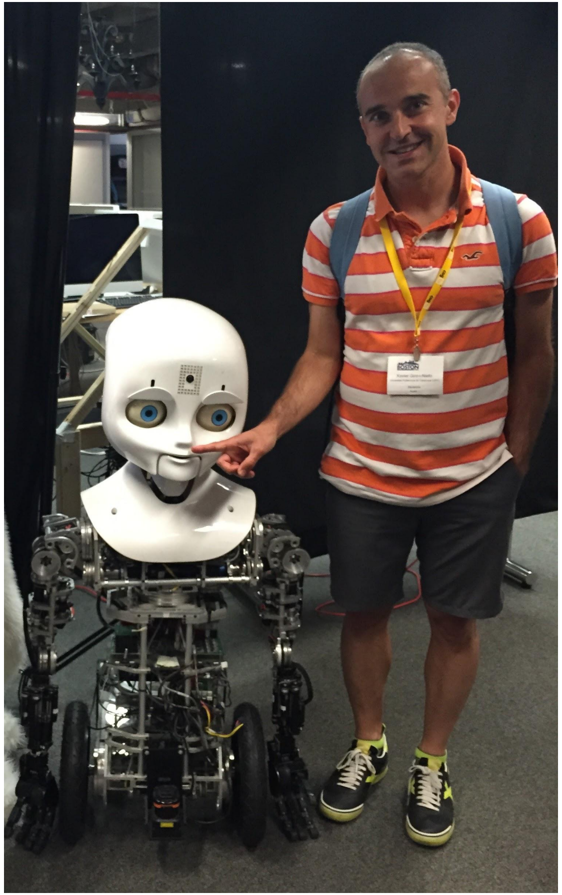
van den Oord, Aaron, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel Recurrent Neural Networks." arXiv preprint arXiv:1601.06759 (2016).

Applications: Segmentation



Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, Aaron Courville, [“ReSeg: A Recurrent Neural Network-Based Model for Semantic Segmentation”](). DeepVision CVPRW 2016.

Thanks ! Q&A ?



Follow me at

/ProfessorXavi

@DocXavi

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Department of Signal Theory
and Communications

Image Processing Group

https://imatge.upc.edu/web/people/xavier-giro