Deep Learning for NLP

- Oliver Guhr
  I833 SS2019

Natural Language Processing

NLP Tasks

- Easy
  - Spell Checking
  - Keyword Search

NLP Tasks

- Easy
  - Spell Checking
  - Keyword Search
- Medium
  - Parsing information from unstructured text

NLP Tasks

- Easy
  - Spell Checking
  - Keyword Search
- Medium
  - Parsing information from unstructured text
- Hard
  - Machine Translation
  - Semantic Analysis
  - Question Answering

DeepL Übersetzer Linguee DeepL Pro Blog Info

Übersetze **Englisch** (erkannt)

A very simple way to improve the performance of almost any machine learning algorithm is to train many different models on the same data and then to average their predictions. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow deployment to a large number of users, especially if the individual models are large neural nets. Caruana and his collaborators have shown that it is possible to compress the knowledge in an ensemble into a single model which is much easier to deploy and we develop this approach further using a different compression technique. We achieve some surprising results on MNIST and we show that we can significantly improve the acoustic model of a heavily used commercial system by distilling the knowledge in an ensemble of models into a single model. We also introduce a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full models confuse. Unlike a mixture of experts, these specialist models can be trained rapidly and in parallel.

Dokument übersetzen 1130/5000

Übersetze nach **Deutsch**

Eine sehr einfache Möglichkeit, die Leistung fast jedes maschinellen Lernalgorithmus zu verbessern, besteht darin, viele verschiedene Modelle auf den gleichen Daten zu trainieren und dann ihre Vorhersagen zu berechnen. Leider ist die Vorhersage mit einem ganzen Ensemble von Modellen umständlich und kann zu rechenintensiv sein, um die Bereitstellung für eine große Anzahl von Benutzern zu ermöglichen, insbesondere wenn die einzelnen Modelle große neuronale Netze sind. Caruana und seine Mitarbeiter haben gezeigt, dass es möglich ist, das Wissen in einem Ensemble zu einem einzigen, viel einfacher zu implementierenden Modell zu komprimieren, und wir entwickeln diesen Ansatz mit einer anderen Kompressionstechnik weiter. Wir erzielen einige überraschende Ergebnisse auf MNIST und zeigen, dass wir das akustische Modell eines stark genutzten kommerziellen Systems signifikant verbessern können, indem wir das Wissen in einem Ensemble von Modellen in ein einziges Modell überführen. Wir stellen auch eine neue Art von Ensemble vor, das aus einem oder mehreren Vollmodellen und vielen Spezialmodellen besteht, die lernen, feinkörnige Klassen zu unterscheiden, die die Vollmodelle verwirren. Im Gegensatz zu einer Mischung von Experten können diese Spezialmodelle schnell und parallel trainiert werden.

Experte noun, masculine

expert n (plural: experts)

Experten bewerteten die Qualität der Produkte.	Experts evaluated the quality of the products.
Die Maschinen werden von Experten kontrolliert.	The machines will be inspected by experts.
Ein Gremium aus Experten berät die Regierung.	A body of experts is advising the government.

specialist n (plural: specialists)

Artificial intelligence (AI)

New AI fake text generator may be too dangerous to release, say creators

The Elon Musk-backed nonprofit company OpenAI declines to release research publicly for fear of misuse

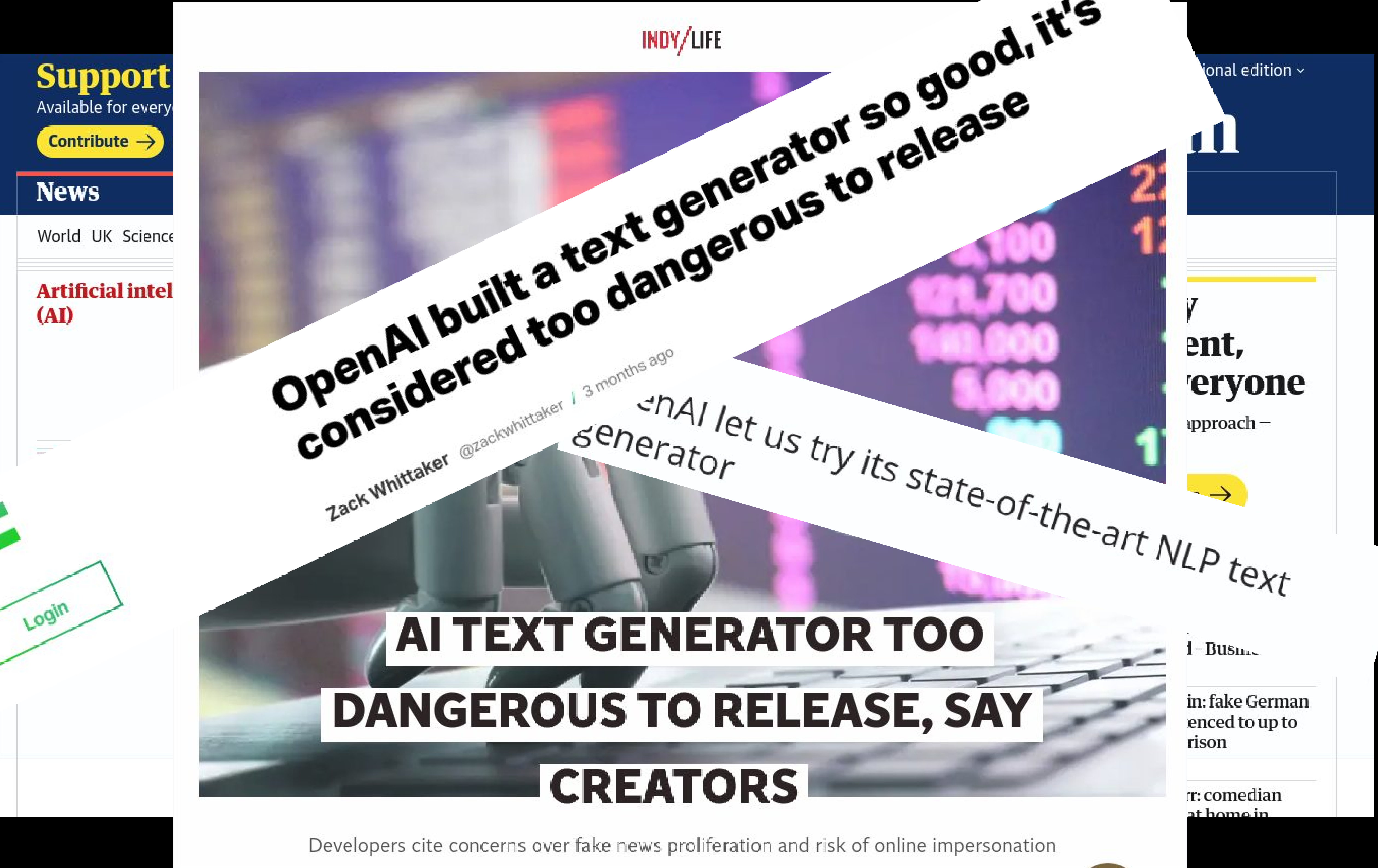
Alex Hern
@alexhern
Thu 14 Feb 2019 17.00 GMT

most viewed


Live US-China trade war: Beijing vows to retaliate as tariffs raised – Business live


Anna Sorokin: fake German heiress sentenced to up to 12 years in prison


Freddie Starr: comedian found dead at home in



INDY/LIFE
OpenAI built a text generator so good, it's considered too dangerous to release
Zack Whittaker @zackwhittaker / 3 months ago
enAI let us try its state-of-the-art NLP text generator
AI TEXT GENERATOR TOO DANGEROUS TO RELEASE, SAY CREATORS
Developers cite concerns over fake news proliferation and risk of online impersonation
Support
Available for every
Contribute
News
World UK Science
Artificial intel (AI)
Login
ional edition

The GTP-2 Model

- Model trained on 40GB text
- transformer-based language model
- Objective: predict the next word, given all of the previous words within some text
- Try (a smaller) version online:
  - https://talktotransformer.com
- Paper and Code:
  - https://openai.com/blog/better-language-models/

human-written input:

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

model output:

The scientist named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.
[…]

SQuAD2.0

The Stanford Question Answering Dataset

1973_oil_crisis

The Stanford Question Answering Dataset

The 1973 oil crisis began in October 1973 when the members of the Organization of Arab Petroleum Exporting Countries (OAPEC, consisting of the Arab members of OPEC plus Egypt and Syria) proclaimed an oil embargo. By the end of the embargo in March 1974, the price of oil had risen from US$3 per barrel to nearly $12 globally; US prices were significantly higher. The embargo caused an oil crisis, or "shock", with many short- and long-term effects on global politics and the global economy. It was later called the "first oil shock", followed by the 1979 oil crisis, termed the "second oil shock."

When did the 1973 oil crisis begin?
Ground Truth Answers: October 1973 | October 1973 | October 1973 | October | 1973

What was the price of oil in March of 1974?
Ground Truth Answers: nearly $12 | $12 | $12 | $12 | $12

When was the second oil crisis?
Ground Truth Answers: 1979 | 1979 | 1979 | 1979 | 1979

What was another term used for the oil crisis?
Ground Truth Answers: first oil shock | shock | shock | first oil shock | shock

Who proclaimed the oil embargo?
Ground Truth Answers: members of the Organization of Arab Petroleum Exporting Countries | members of the Organization of Arab Petroleum Exporting Countries | Organization of Arab Petroleum Exporting Countries | members of the Organization of Arab Petroleum Exporting Countries | OAPEC

https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/1973_oil_crisis.html

Leaderboard

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Mar 20, 2019	BERT + DAE + AoA (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	**87.147**	**89.474**
2 Mar 15, 2019	BERT + ConvLSTM + MTL + Verifier (ensemble) *Layer 6 AI*	86.730	89.286
3 Mar 05, 2019	BERT + N-Gram Masking + Synthetic Self-Training (ensemble) *Google AI Language* https://github.com/google-research/bert	86.673	89.147

Leaderboard

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Mar 20, 2019	BERT + DAE + AoA (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	**87.147**	**89.474**
2 Mar 15, 2019	BERT + ConvLSTM + MTL + Verifier (ensemble) *Layer 6 AI*	86.730	89.286
3 Mar 05, 2019	BERT + N-Gram Masking + Synthetic Self-Training (ensemble) *Google AI Language* https://github.com/google-research/bert	86.673	89.147
4 Apr 13, 2019	SemBERT(ensemble) *Shanghai Jiao Tong University*	86.166	88.886
5 Mar 16, 2019	BERT + DAE + AoA (single model) *Joint Laboratory of HIT and iFLYTEK Research*	85.884	88.621
6 Mar 05, 2019	BERT + N-Gram Masking + Synthetic Self-Training (single model) *Google AI Language* https://github.com/google-research/bert	85.150	87.715
7 Jan 15, 2019	BERT + MMFT + ADA (ensemble) *Microsoft Research Asia*	85.082	87.615
7 Mar 13, 2019	BERT + ConvLSTM + MTL + Verifier (single model) *Layer 6 AI*	84.924	88.204
8 Apr 16, 2019	Insight-baseline-BERT (single model) *PAII Insight Team*	84.834	87.644

Leaderboard

Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Mar 20, 2019	BERT + DAE + AoA (ensemble) Joint Laboratory of HIT and iFLYTEK Research	87.147	89.474
2 Mar 15, 2019	BERT + ConvLSTM + MTL + Verifier (ensemble) Layer 6 AI	86.730	89.286
3 Mar 05, 2019	BERT + N-Gram Masking + Synthetic Self-Training (ensemble) Google AI Language https://github.com/google-research/bert	86.673	89.147
4 Apr 13, 2019	SemBERT (ensemble) Shanghai Jiao Tong University	86.166	88.886
5 Mar 16, 2019	BERT + DAE + AoA (single model) Joint Laboratory of HIT and iFLYTEK Research	85.884	88.621
6 Mar 05, 2019	BERT + N-Gram Masking + Synthetic Self-Training (single model) Google AI Language https://github.com/google-research/bert	85.150	87.715
7 Jan 15, 2019	BERT + MMFT + ADA (ensemble) Microsoft Research Asia	85.082	87.615
7 Mar 13, 2019	BERT + ConvLSTM + MTL + Verifier (single model) Layer 6 AI	84.924	88.204
8 Apr 14, 2019	Insight-baseline-BERT (single model) PAII Insight Team	84.834	87.644
8 Apr 13, 2019	SemBERT (single model) Shanghai Jiao Tong University	84.800	87.864
9 Jan 10, 2019	BERT + Synthetic Self-Training (ensemble) Google AI Language https://github.com/google-research/bert	84.292	86.967
10 Dec 21, 2018	PAML+BERT (ensemble model) PINGAN GammaLab	83.457	86.122
10 Dec 13, 2018	BERT finetune baseline (ensemble) Anonymous	83.536	86.096
11 Dec 16, 2018	Lunet + Verifier + BERT (ensemble) Layer 6 AI NLP Team	83.469	86.043
11 Mar 20, 2019	BERT + ours (ensemble) None	83.404	86.036
12 Dec 15, 2018	Lunet + Verifier + BERT (single model) Layer 6 AI NLP Team	82.995	86.035
12 Feb 26, 2019	BERT with Something (ensemble) Anonymous	83.051	85.737
12 Mar 22, 2019	BERT + ours (ensemble) None	83.119	85.910
12 Feb 16, 2019	BERT + ours (ensemble) None	83.171	85.631
12 Jan 16, 2019	BERT + MMFT + ADA (single model) Microsoft Research Asia	83.040	85.892
13 Jan 10, 2019	BERT + Synthetic Self-Training (single model) Google AI Language https://github.com/google-research/bert	82.972	85.810
14 Feb 15, 2019	BERT + NeurQuRI (ensemble) 2YNH	82.803	85.703
15 Dec 16, 2018	PAML+BERT (single model) PINGAN GammaLab	82.577	85.603
15 Feb 27, 2019	BERT + NeurQuRI (ensemble) 2YNH	82.713	85.584
16 Mar 02, 2019	Unnamed submission by zcoff	82.431	85.178
16 Nov 16, 2018	AoA + DA + BERT (ensemble) Joint Laboratory of HIT and iFLYTEK Research	82.374	85.310
17 Dec 13, 2018	BERT finetune baseline (single model) Anonymous	82.126	84.820
17 Feb 28, 2019	BERT_s (single model) Anonymous	81.979	84.846
17 Dec 18, 2018	Candi-Net+BERT (ensemble) 42Maru NLP Team	82.126	84.624
18 Feb 28, 2019	BERT-large ABBT (single model) anonymous	81.573	84.535
19 Feb 20, 2019	BERT with Something (single model) Anonymous	81.330	84.386
19 Feb 15, 2019	BERT + NeurQuRI (single model) 2YNH	81.217	84.342
20 Nov 16, 2018	AoA + DA + BERT (single model) Joint Laboratory of HIT and iFLYTEK Research	81.178	84.251
21 Mar 20, 2019	BERT + ours (single) None	80.693	83.922
21 Mar 07, 2019	BERT + UnAnsQ (single model) Anonymous	80.749	83.851
22 Apr 07, 2019	BERT + AL (single model) Anonymous	80.731	83.827
23 Dec 09, 2018	Candi-Net+BERT (single model) 42Maru NLP Team	80.659	83.562
24 Jan 09, 2019	Unnamed submission by null	80.532	83.539
25 Mar 20, 2019	BERT + ours (single) None	80.431	83.457
25 Jan 22, 2019	BERT + NeurQuRI (single model) 2YNH	80.591	83.391
26 Apr 10, 2019	(BERT-Span) (single model) Anonymous	80.354	83.329
27 Feb 16, 2019	BERT + ours (single model) None	80.343	83.243
27 Jan 09, 2019	Unnamed submission by null	80.343	83.221
28 Feb 09, 2019	BERT + UDA (single model) Anonymous	80.005	83.208
28 Dec 03, 2018	PwP+BERT (single model) AITRICS	80.117	83.189
29 Apr 10, 2019	BERT (single model) stevde anonymous	79.971	83.184
29 Apr 04, 2019	RMAN-CC (single model) Seoul National University & Hyundai Motors	80.208	83.149
30 Nov 08, 2018	BERT (single model) Google AI Language	80.005	83.061
31 Feb 02, 2019	BERT + Sparse-Transformer single model	79.948	83.023
31 Dec 05, 2018	Candi-Net+BERT (single model) 42Maru NLP Team	80.388	82.908
31 Feb 28, 2019	ST_bl single model	80.140	82.962
32 Dec 06, 2018	NEXYS_BASE (single model) NEXYS, DGIST ICT	79.779	82.912
32 Mar 07, 2019	BERT uncased (single model) Anonymous	79.781	83.020
33 Feb 05, 2019	(BERT-finetuning) (single model) hust	79.632	82.852
34 Mar 14, 2019	(Anonymous) (single model) Anonymous	78.876	82.524
34 Nov 09, 2018	LifNet + BERT (single model) Layer 6 AI	79.181	82.259
35 Mar 14, 2019	RMAN (single model) Seoul National University & Hyundai Motors	78.481	81.531
35 Apr 26, 2019	BERT + MAAN (ensemble) Infinyou Limited	78.610	81.457
36 Jan 09, 2019	Unnamed submission by null	78.301	81.350
37 Dec 16, 2018	BERT uAC (single model) Hithink RoyalFlush	78.032	81.174
38 Nov 16, 2018	SLQA+BERT (single model) Alibaba DAMO NLP	77.003	80.209

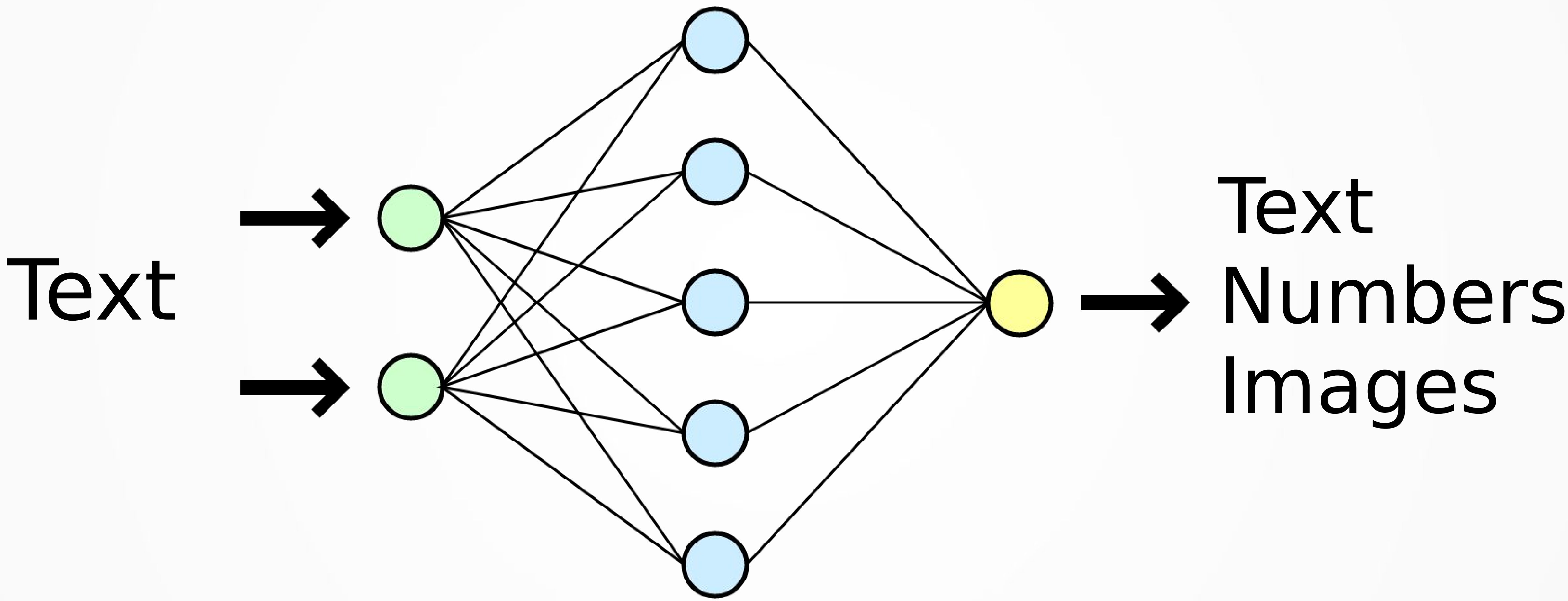
BERT will be part of the course

Literature and Sources

Sources

- Deep Learning by Andrew W. Trask
  - very practical approch
- Deep Learning by Goodfellow, Bengio, Curville
  - A lot of theoretical background, online available
- Stanford CS224: Natural Language Processing with Deep Learning
  - public available course, with videos from all lectures

Text and Neural Networks


Text
Text
Numbers
Images

How do we encode characters for a neural network?

One Hot Encoding

Lets encode the word „hello“

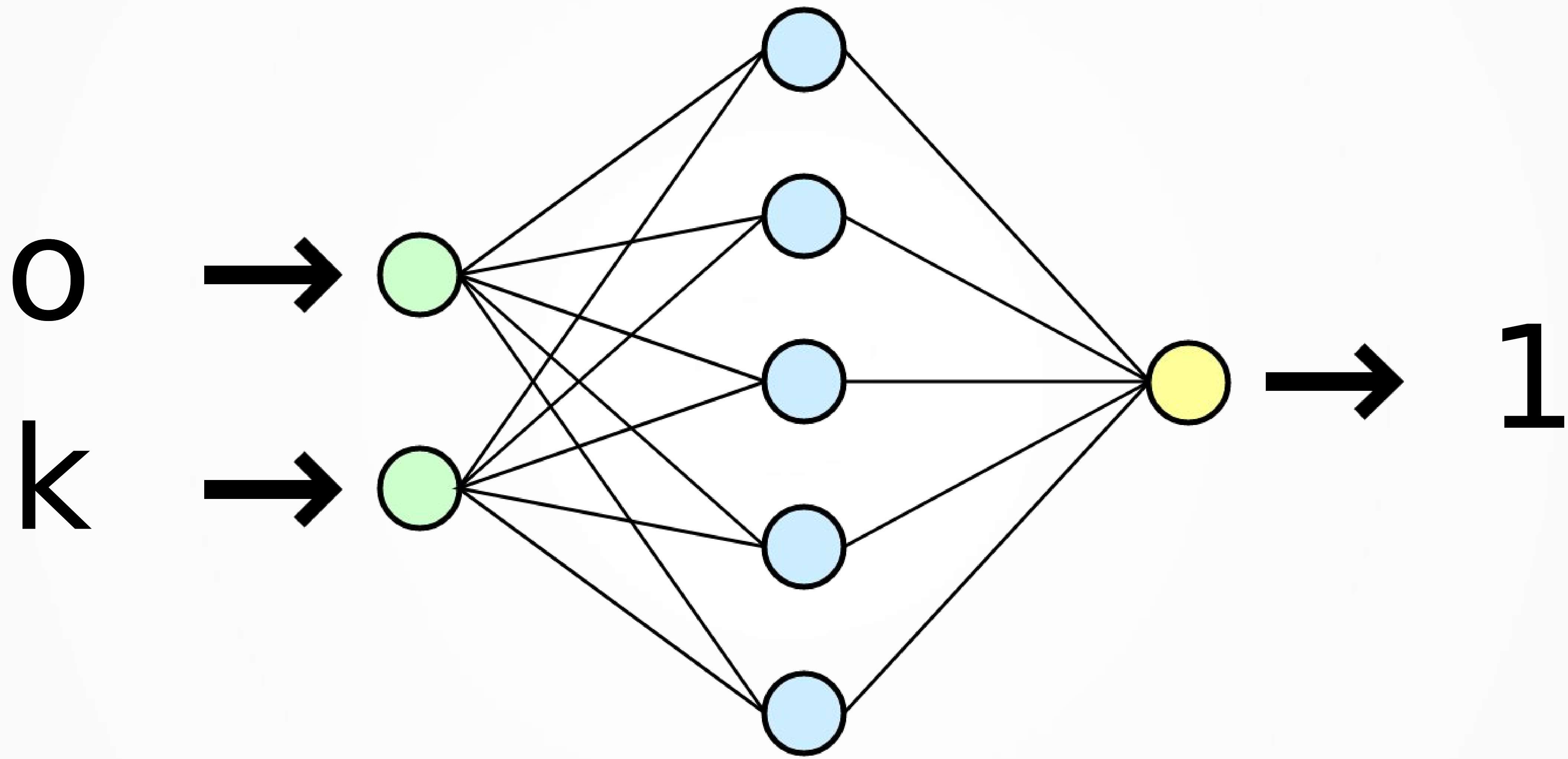
h	0	0	0	1
e	0	0	1	0
l	0	1	0	0
o	1	0	0	0

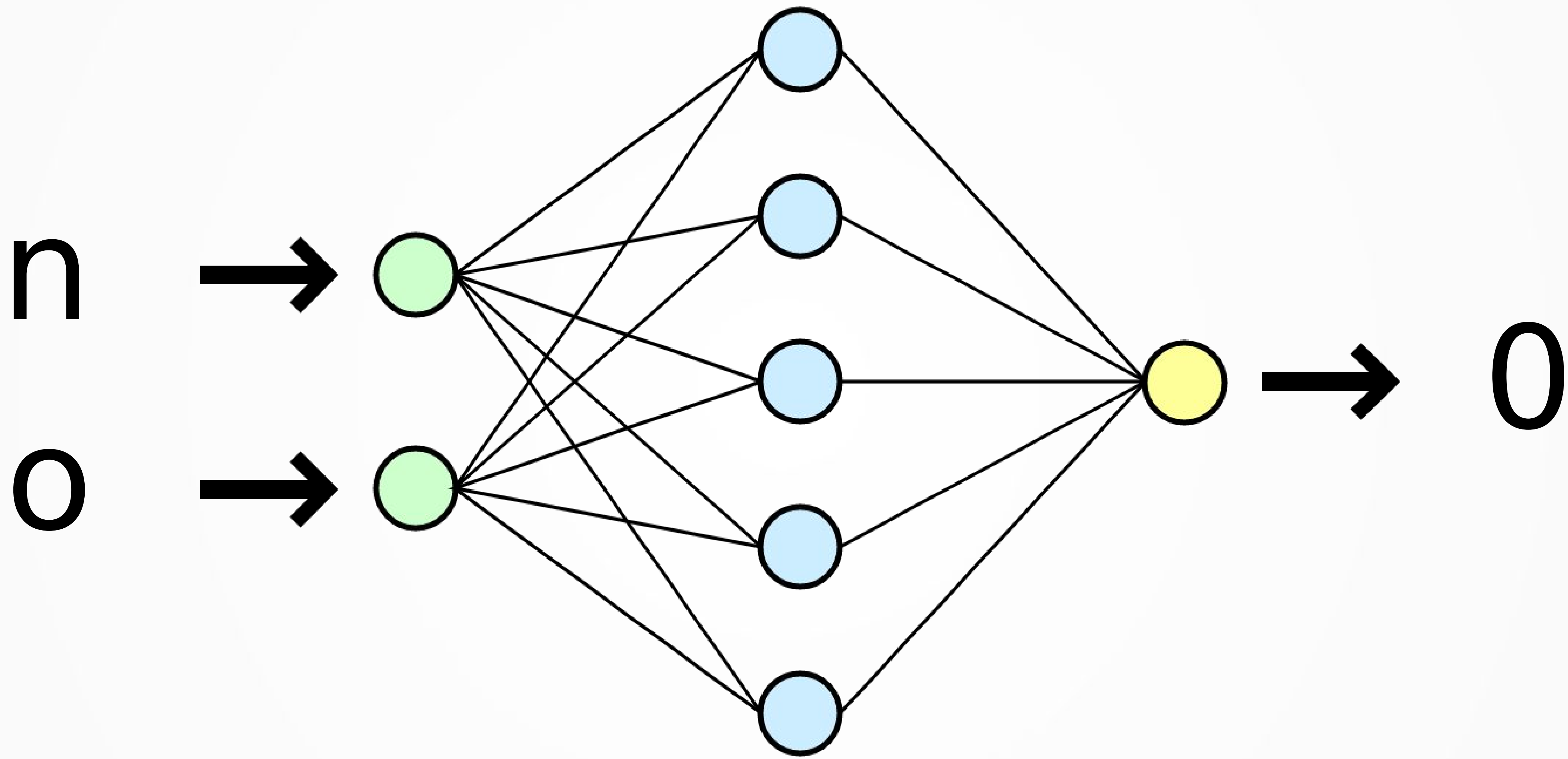
One Hot Encoding

Lets encode the word „hello“

h	0	0	0	1
e	0	0	1	0
l	0	1	0	0
o	1	0	0	0

$$v^h = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad v^e = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \dots \quad \longrightarrow \quad V^{hello} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$


o
k
1


n
o
0

What about:

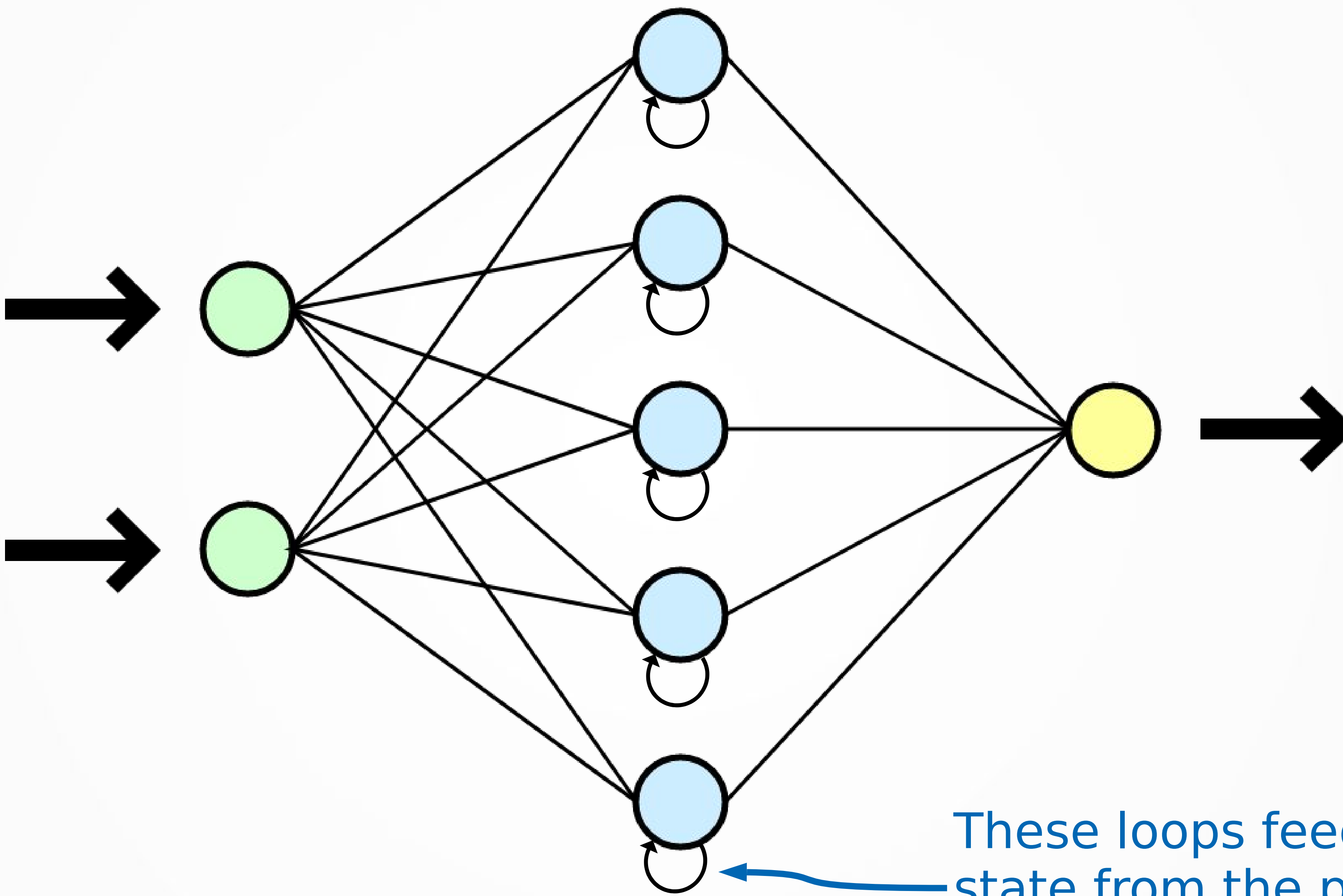
Not
Never
Nope
yes
...

or even sentences?

We need to handle inputs of arbitrary length.

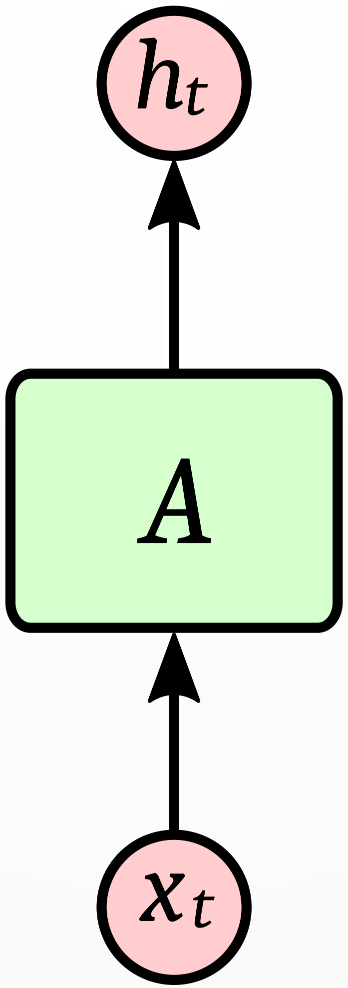
Recurrent Neural Networks

RNNs are networks with loops, allowing information to persist. [Rummelhart et al. 1986]

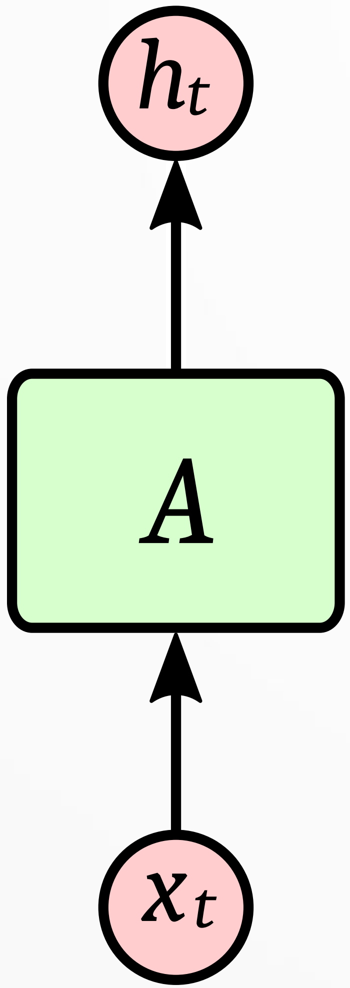


These loops feed in the cell state from the previous timestep.

Simple Neural Network



Simple Neural Network



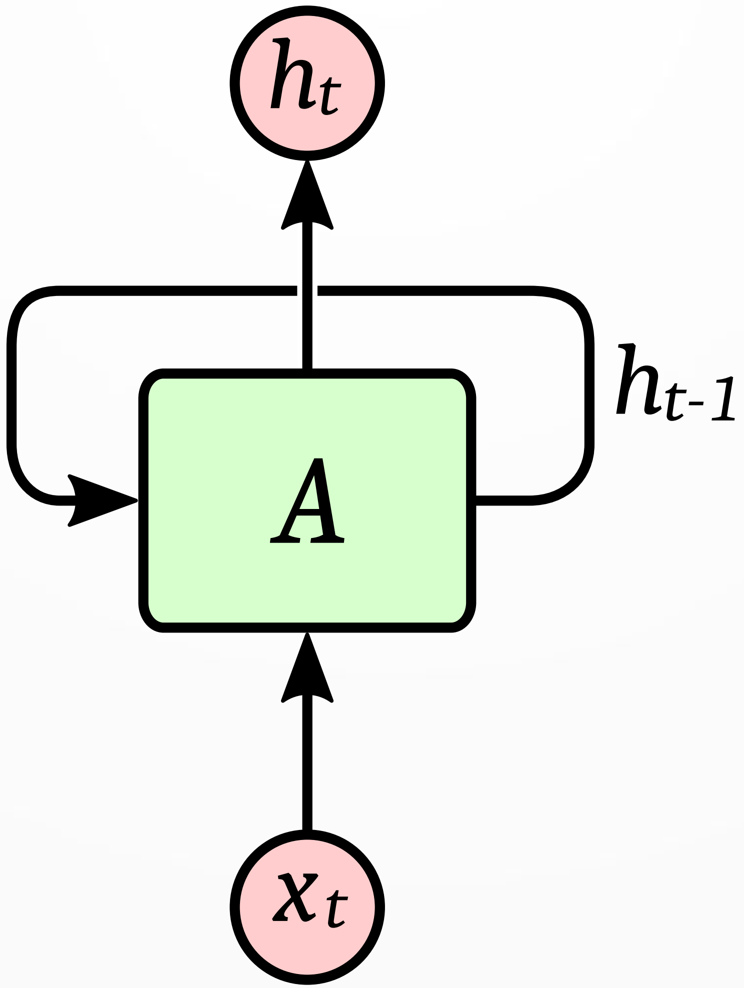
$$h_t = A(x_t)$$

new state

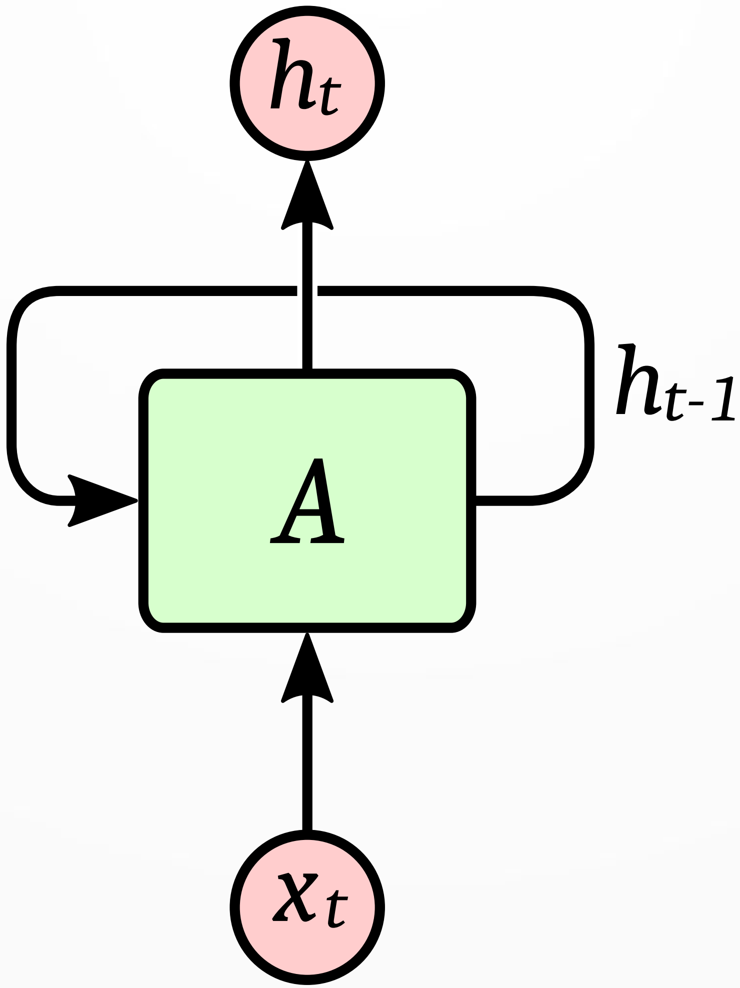
network function

input vector

Recurrent Neural Network

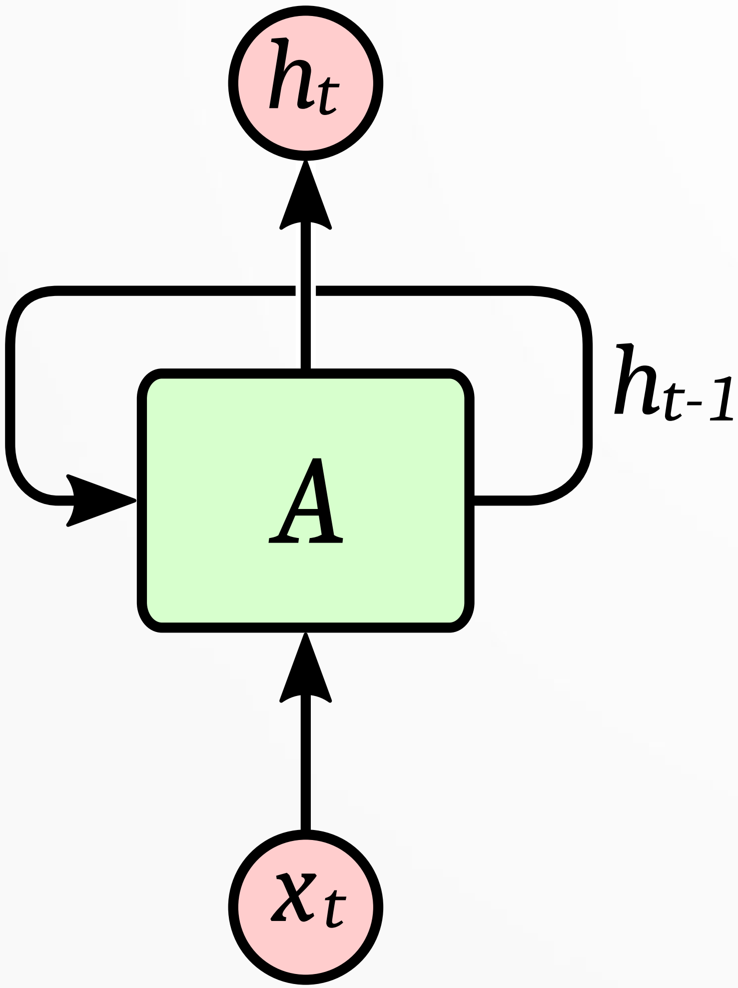


Recurrent Neural Network



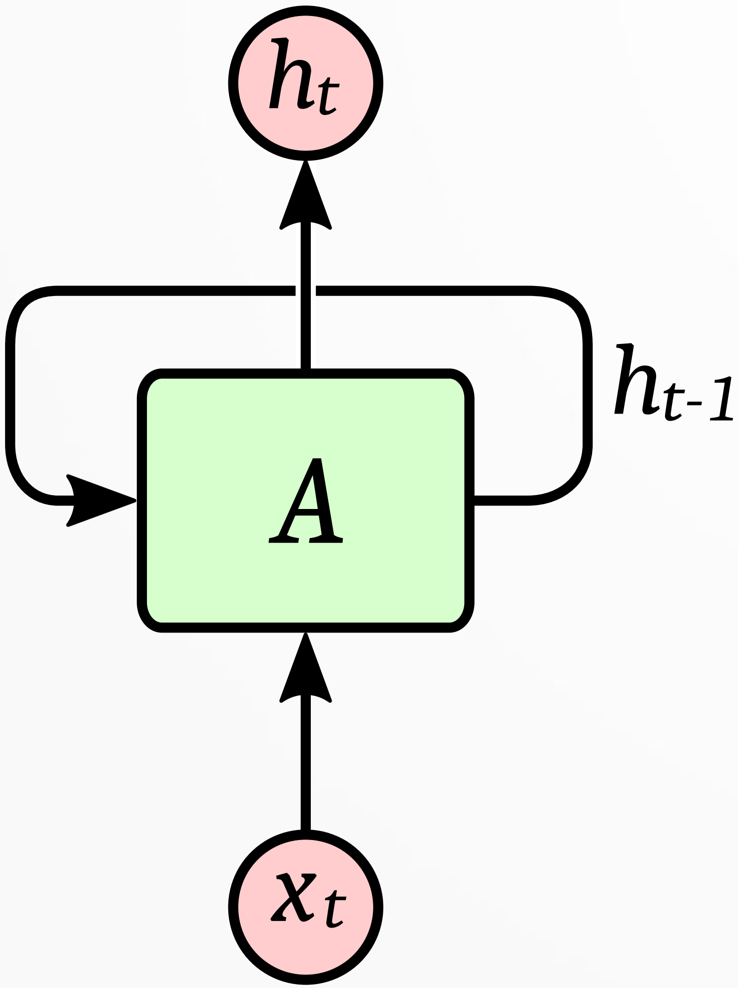
By passing the previous hidden state, the network can keep an „internal state“as the input sequence is processed.

Recurrent Neural Network



$$\underbrace{h_t}_{\text{new state}} = \underbrace{A}_{\text{network function}} \left(\underbrace{h_{t-1}}_{\text{previous state}}, \underbrace{x_t}_{\text{input vector}} \right)$$

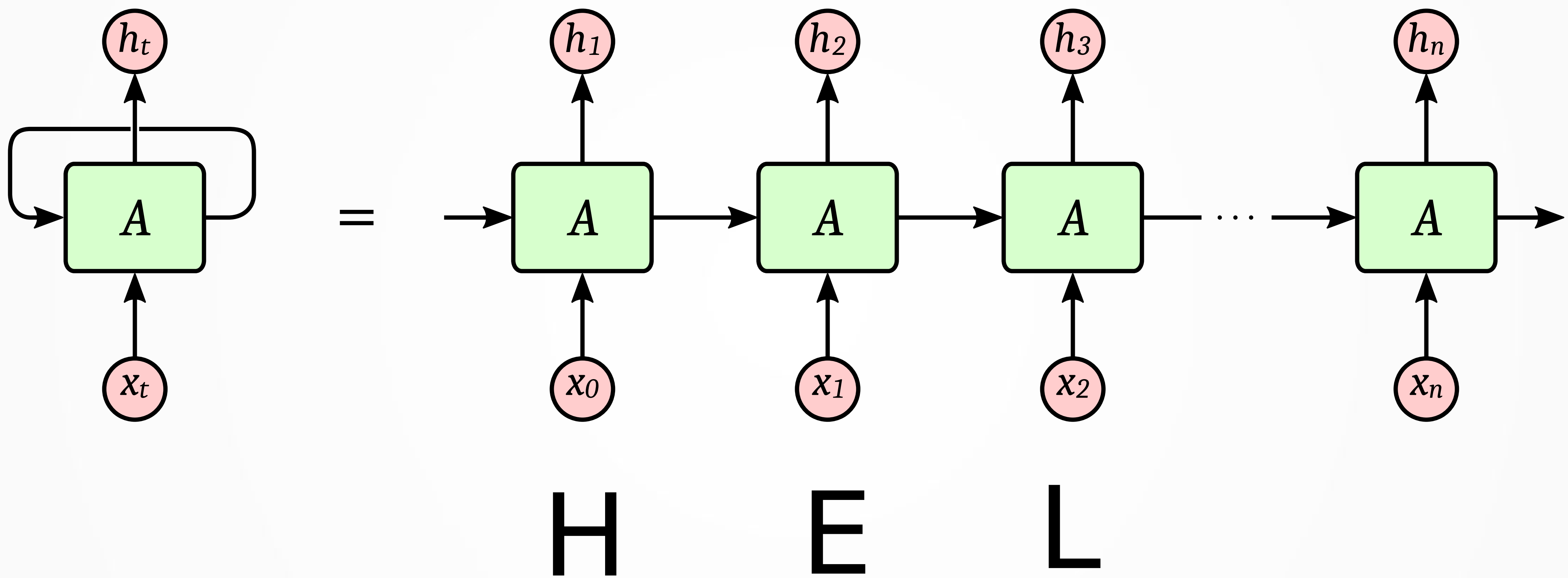
Recurrent Neural Network



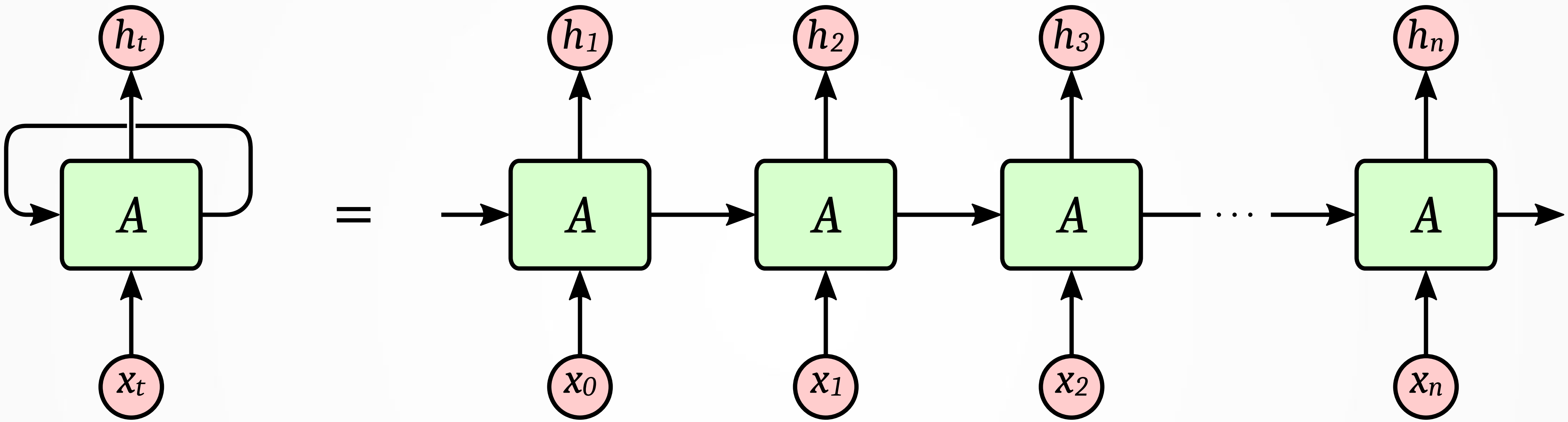
$$\underbrace{h_t}_{\text{new state}} = \underbrace{A}_{\text{network function}} \left(\underbrace{h_{t-1}}_{\text{previous state}}, \underbrace{x_t}_{\text{input vector}} \right)$$

Now we can process a sequence by recursively applying this formula.

Unfolding in time



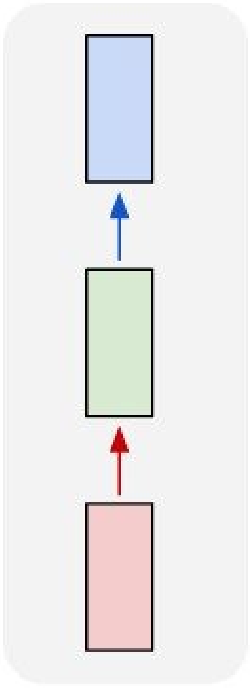
h_t
A
x_t
=
h_1
h_2
h_3
h_n
A
A
A
A
x_0
x_1
x_2
x_n
H
E
L



H E L ....

Network Architectures

one to one



Simple Neural Network

Source: A. Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/

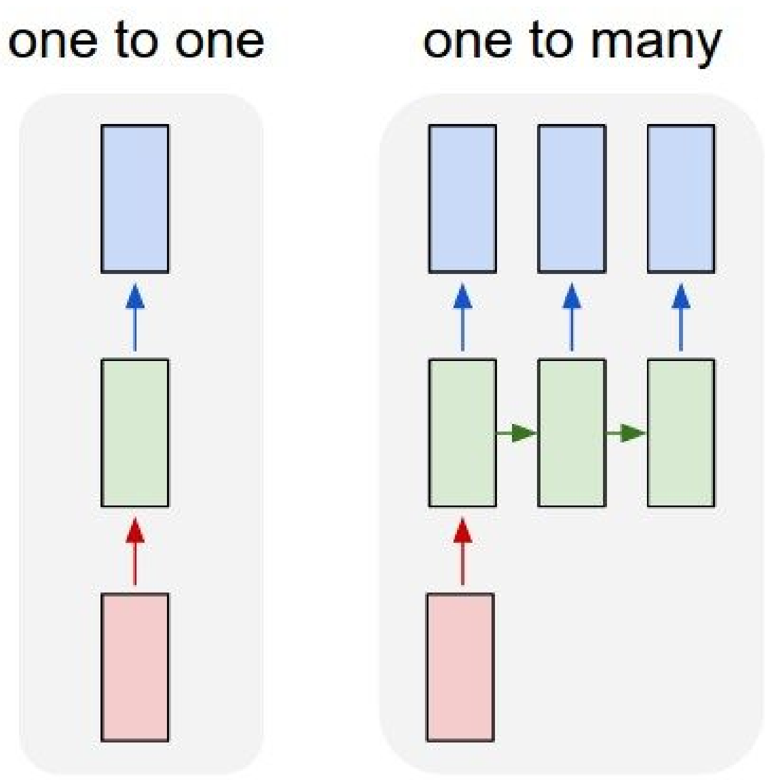
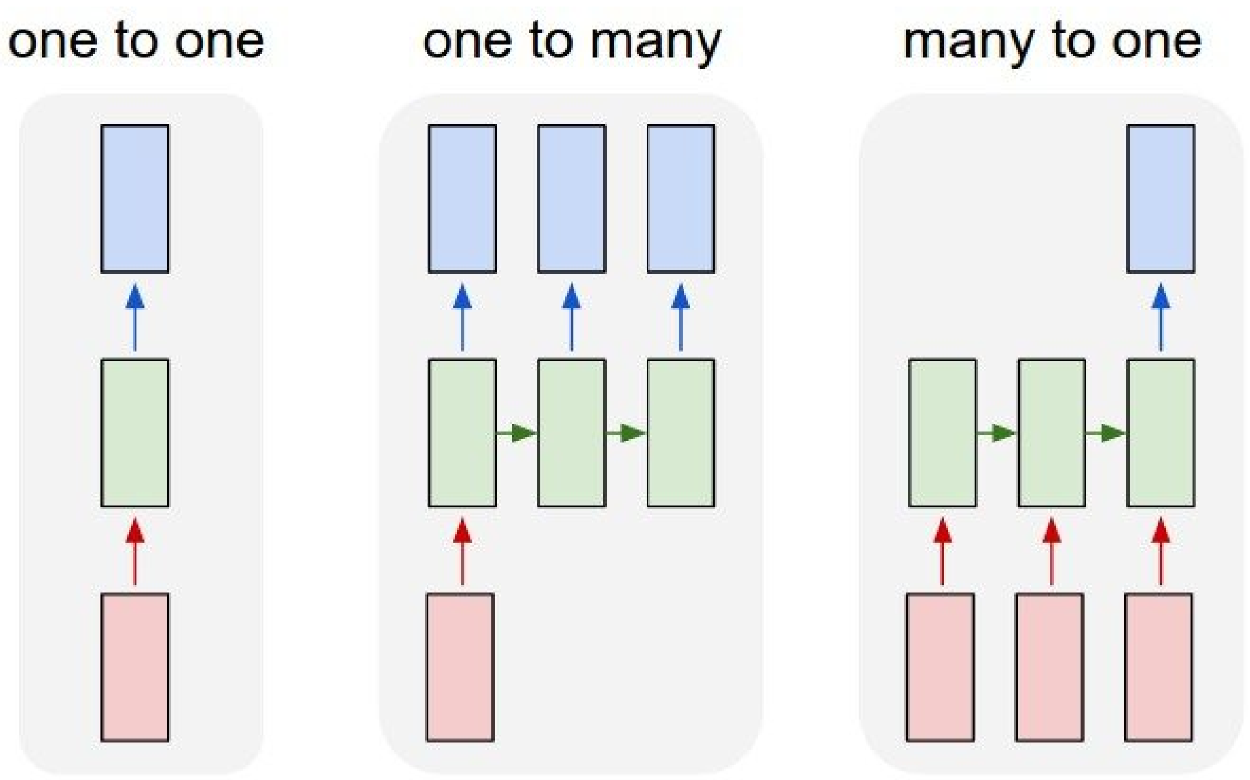


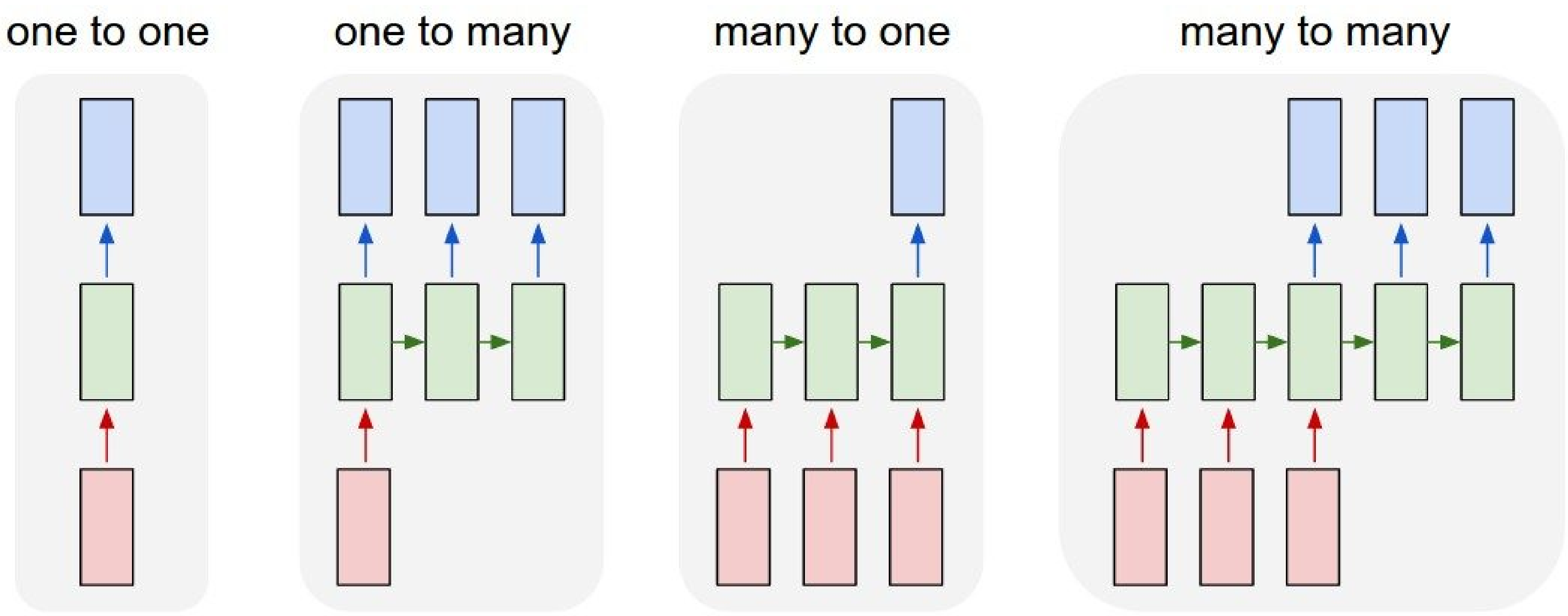
image captioning
Image to a sequence of words

Source: A. Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/



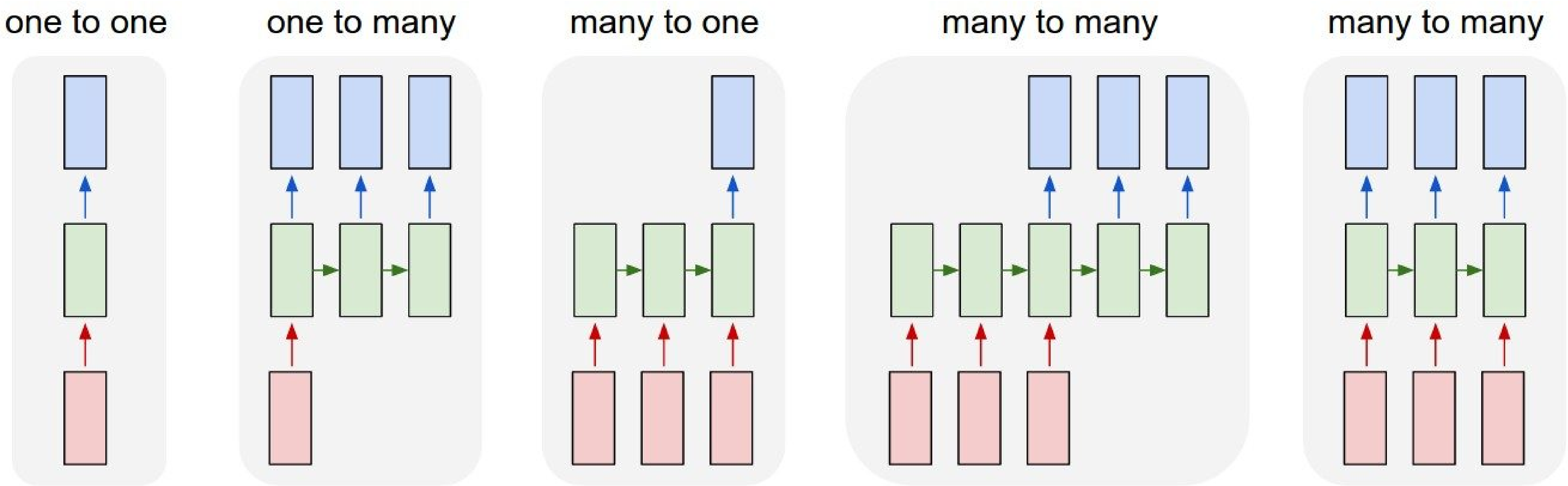
classification
sequence of words to a class

Source: A. Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/



machine translation
sequence of words to sequence of words

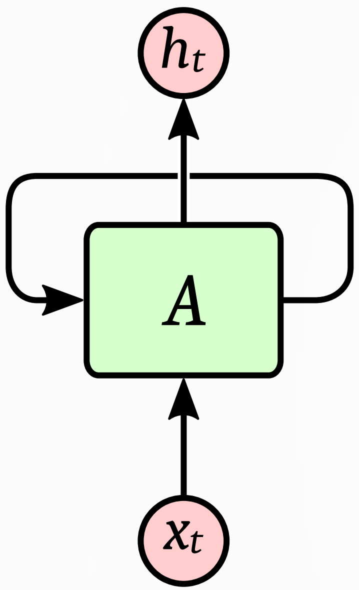
Source: A. Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/



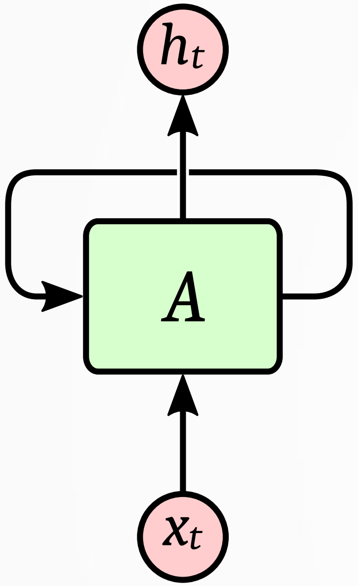
Video classification
A list of frames to a list of classes

Source: A. Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/

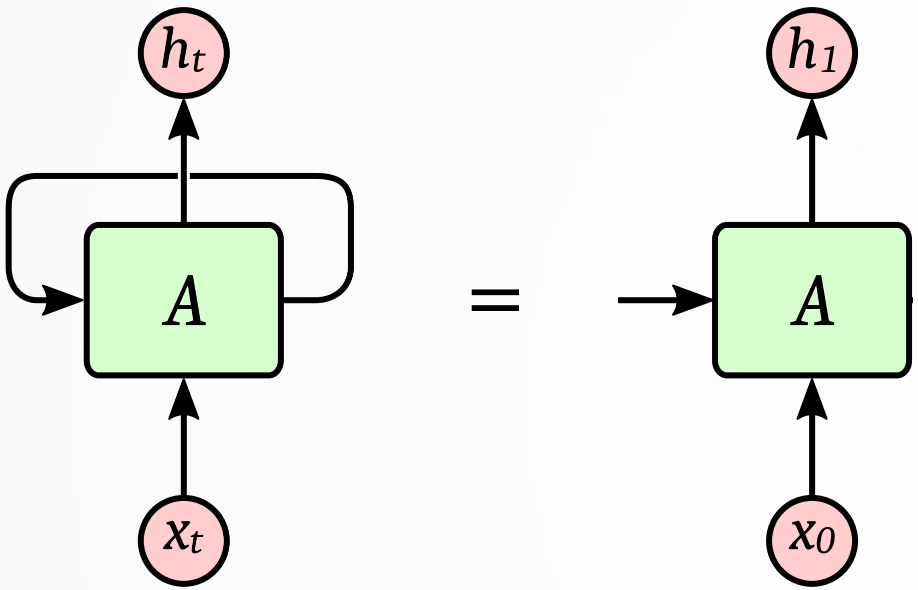
Backpropagation

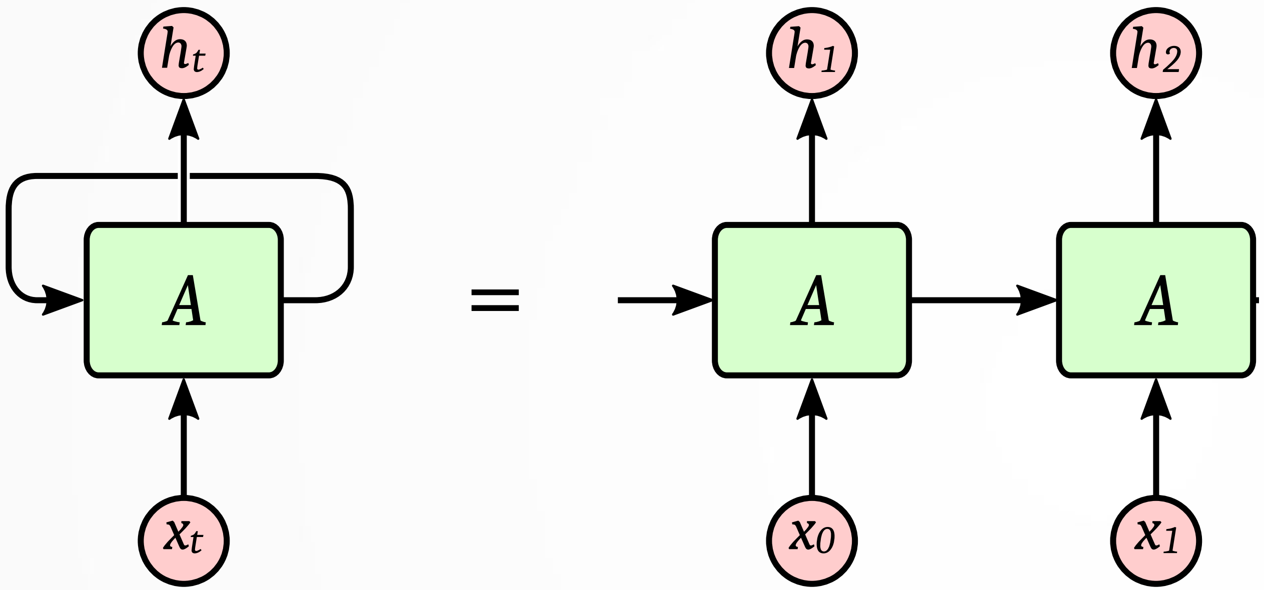


How can we apply backpropagation to nets with loops?



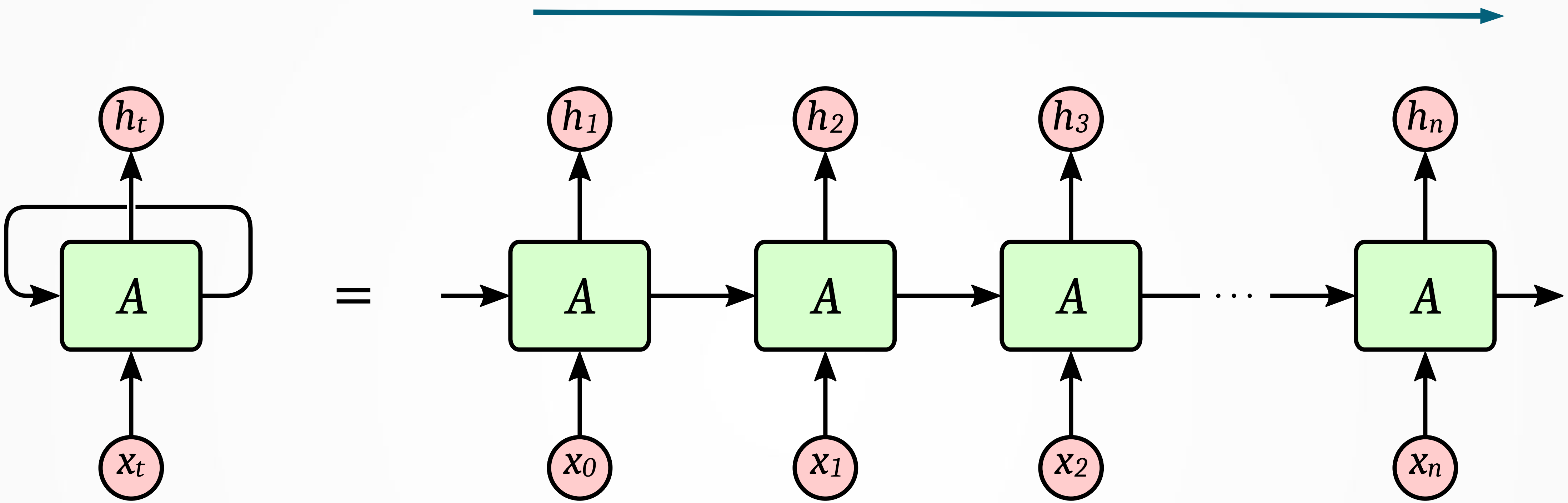
with backpropagation through time (BPTT)


h_t
A
x_t
=
h_1
A
x_0

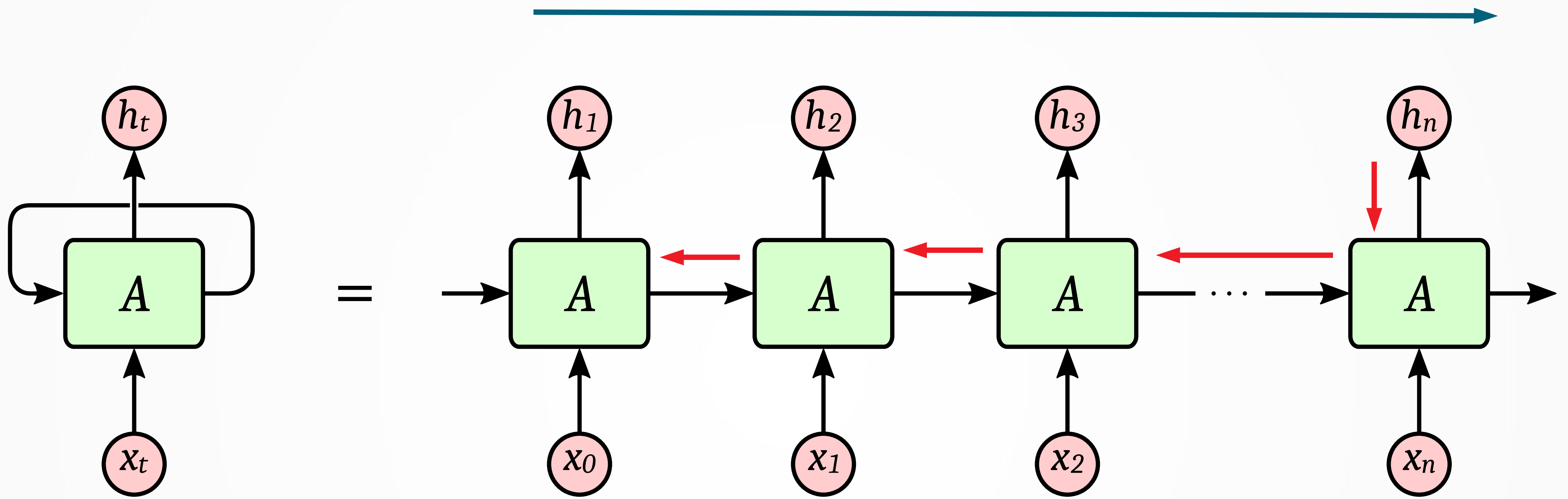


h_t
A
x_t
=
h_1
A
x_0
h_2
A
x_1

1. forward-propagate the inputs over the unfolded network

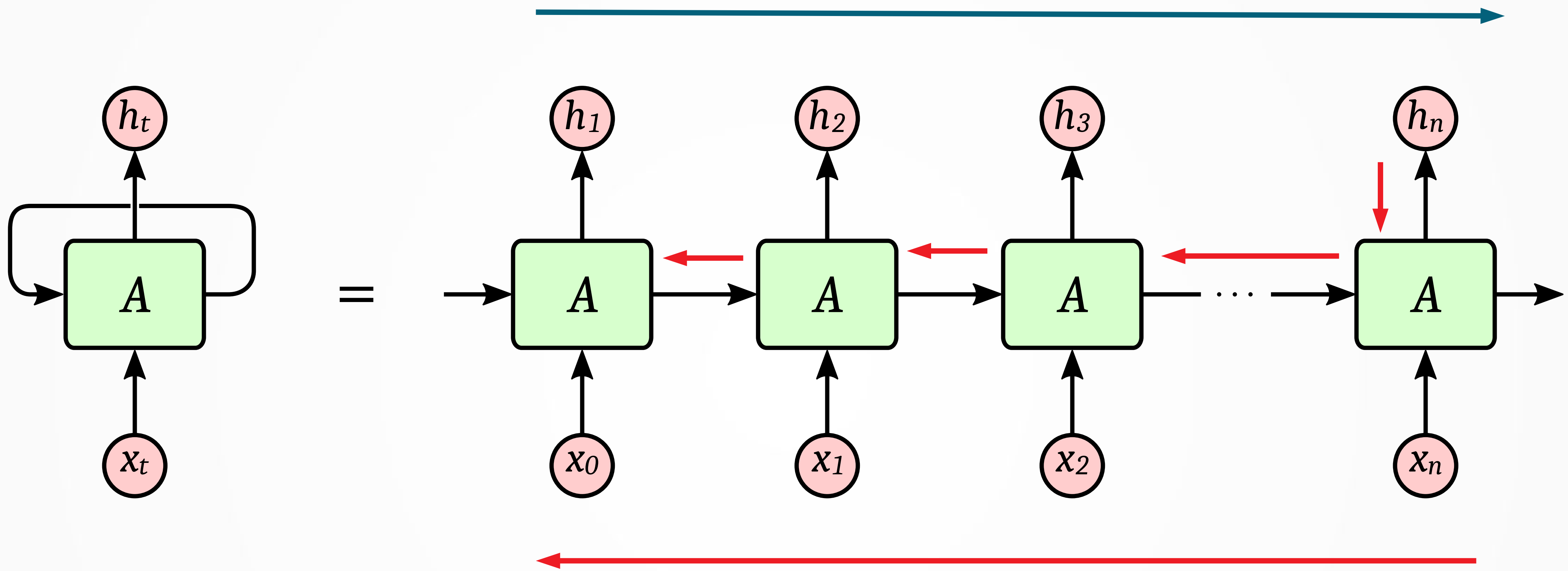


1. forward-propagate the inputs over the unfolded network



2. back-propagate the error, back across the unfolded network

1. forward-propagate the inputs over the unfolded network



2. back-propagate the error, back across the unfolded network

3. sum the weight changes and update all weights

Further Information

- A well explained implementation of BPTT an be found here

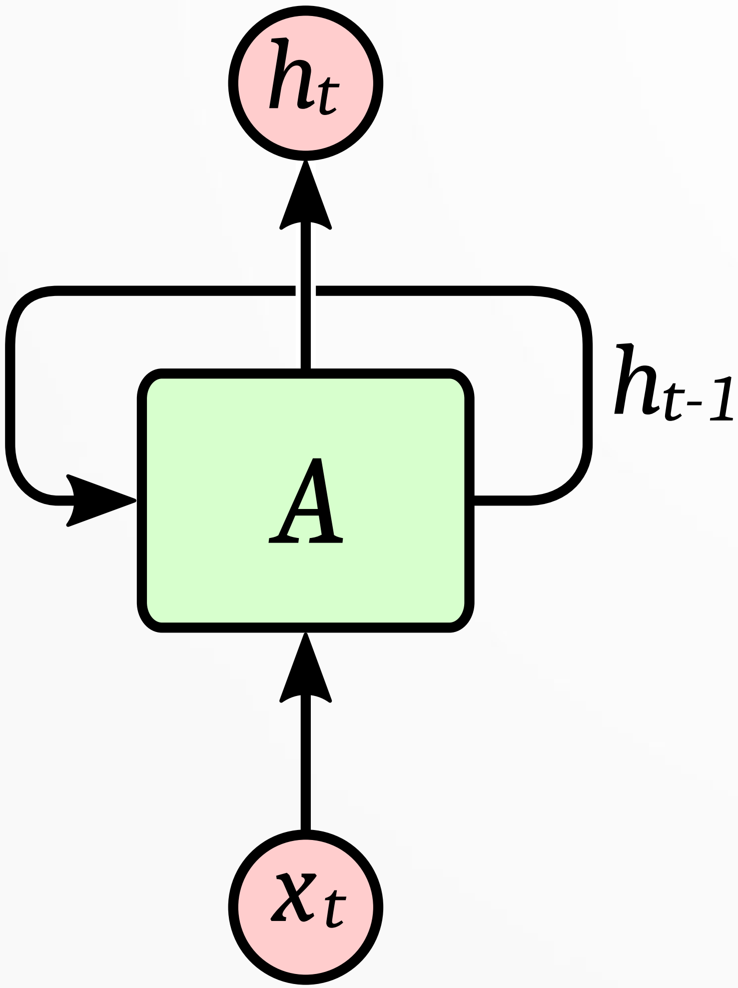
- Andrew Ng explaining BPTT

The Vanishing Gradient Problem

Learning long-term dependencies with gradient descent is difficult

Y. Bengio, P. Simard and P. Frasconi in IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, March 1994.

Recurrent Neural Network



$$\underbrace{h_t}_{\text{new state}} = \underbrace{A}_{\text{network function}}\left(\underbrace{h_{t-1}}_{\text{previous state}}, \underbrace{x_t}_{\text{input vector}}\right)$$

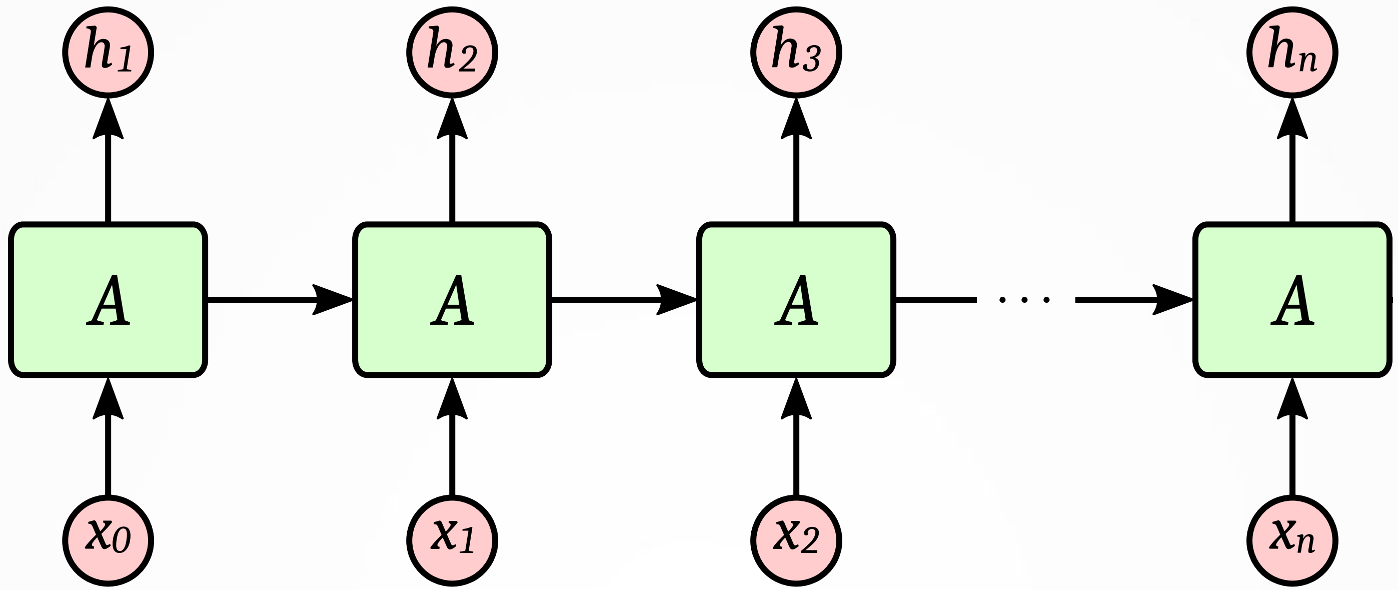
Recurrent Neural Network

$$\underbrace{h_t}_{\text{new state}} = \underbrace{A}_{\text{network function}}\left(\underbrace{h_{t-1}}_{\text{previous state}}, \underbrace{x_t}_{\text{input vector}}\right)$$

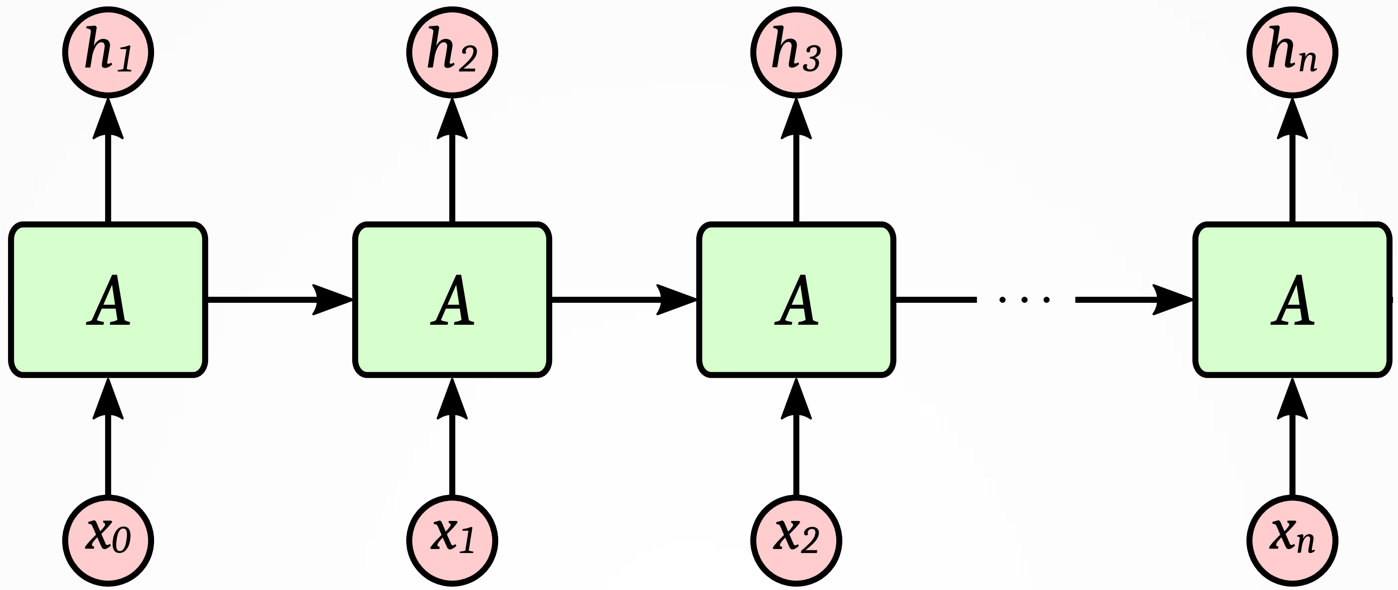
A simple example of a Elman-network

$$h_t = \tanh\left(W_{hh} h_{t-1} + W_{xh} x_t\right)$$

$$y_t = W_{yh} h_t$$

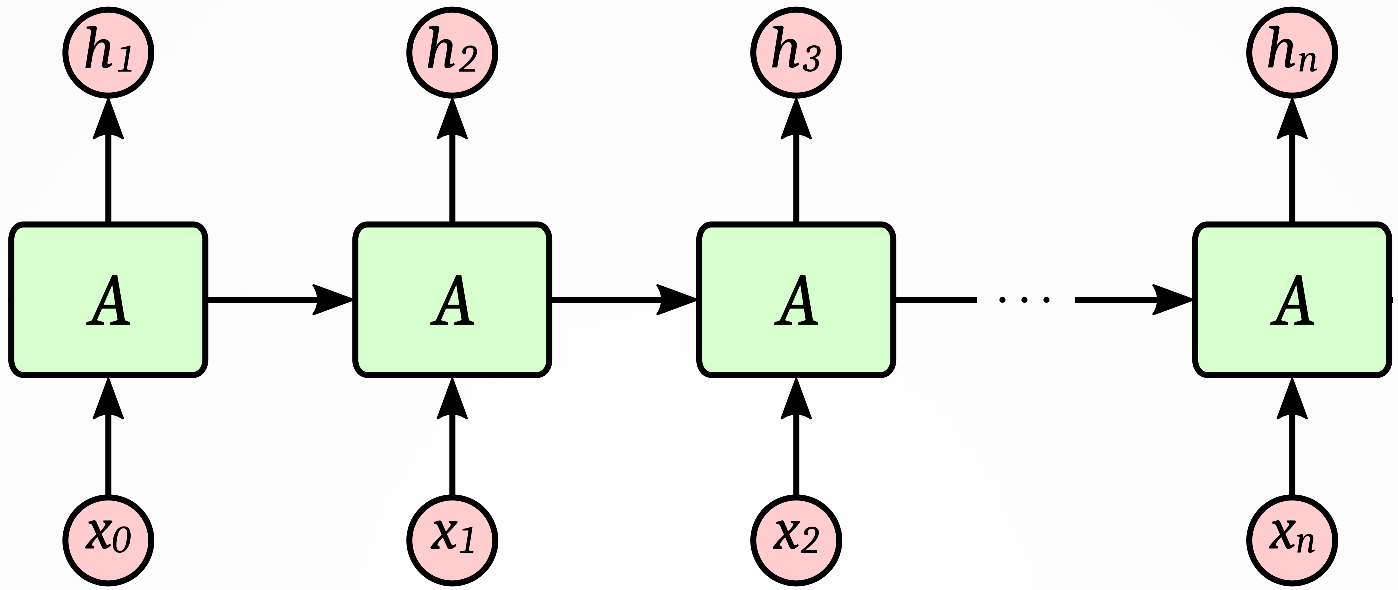


$$h_1 = \tanh\left(W_{hh} h_0 + W_{xh} x_1\right)$$



$$h_1 = \tanh\left(W_{hh} h_0 + W_{xh} x_1\right)$$

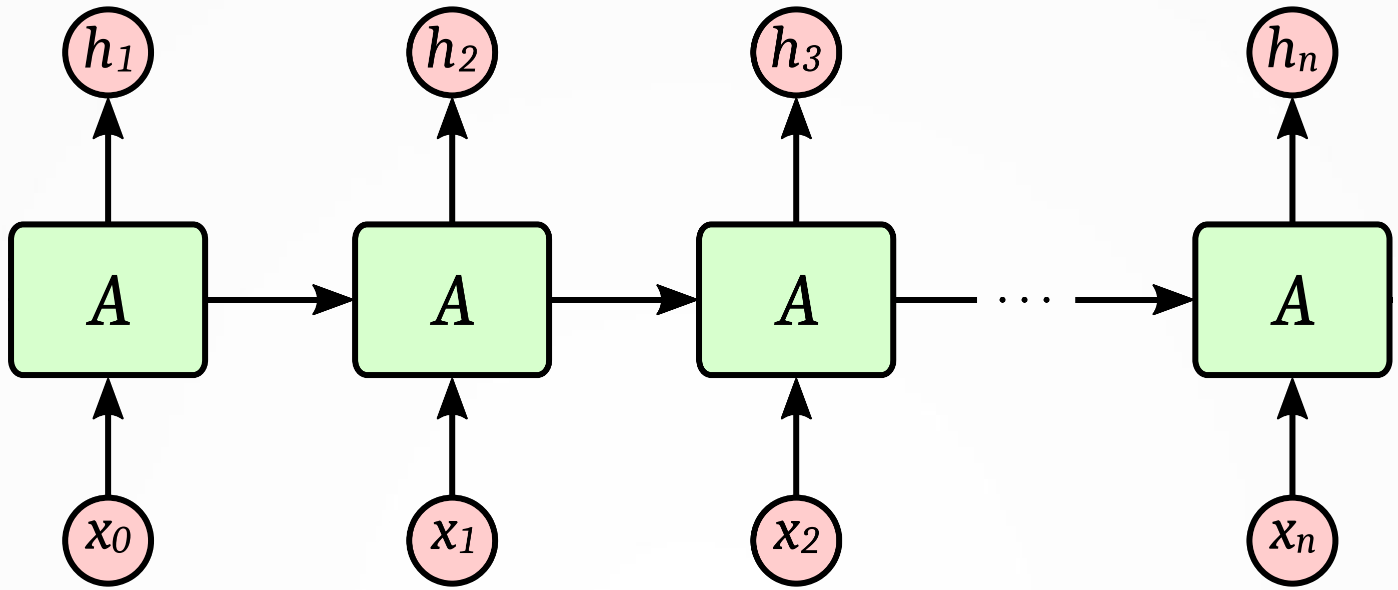
$$h_2 = \tanh\left(W_{hh}\left(\tanh\left(W_{hh} h_0 + W_{xh} x_1\right)\right) + W_{xh} x_2\right)$$



$$h_1 = \tanh\left(W_{hh} h_0 + W_{xh} x_1\right)$$

$$h_2 = \tanh\left(W_{hh}\left(\tanh\left(W_{hh} h_0 + W_{xh} x_1\right)\right) + W_{xh} x_2\right)$$

$$h_3 = \tanh\left(W_{hh}\left(\tanh\left(W_{hh}\left(\tanh\left(W_{hh} h_0 + W_{xh} x_1\right)\right) + W_{xh} x_2\right)\right) + W_{xh} x_3\right)$$

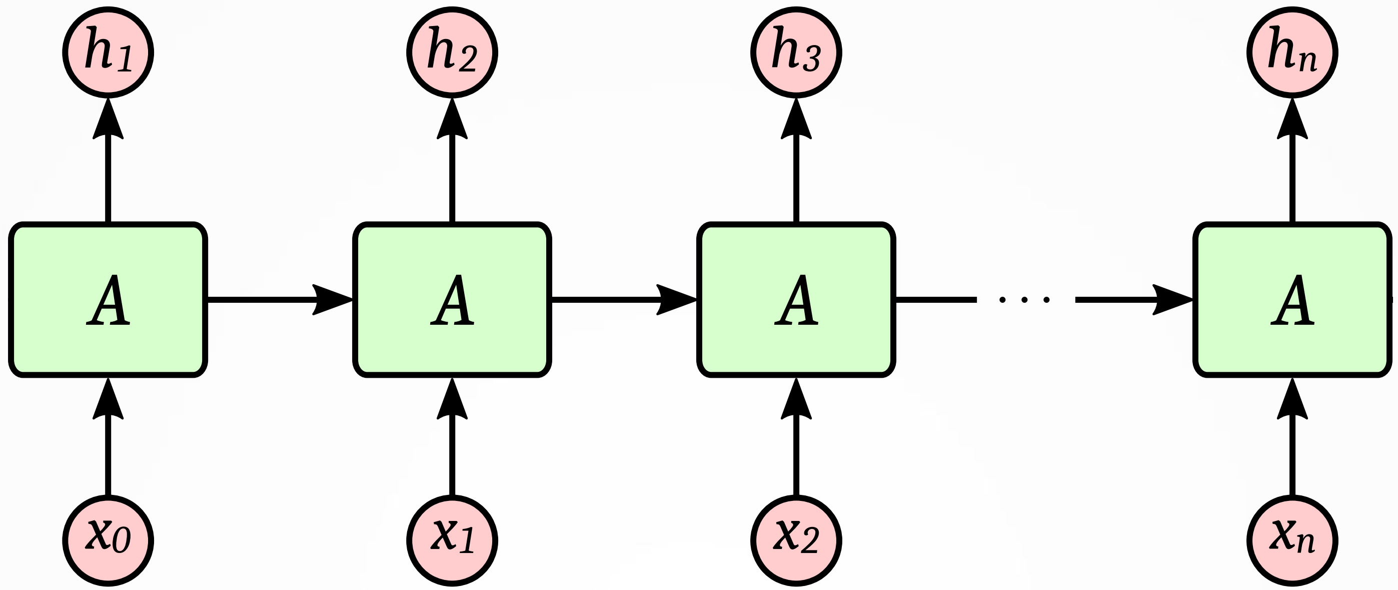


$$h_1 = \tanh(W_{hh} h_0 + W_{xh} x_1)$$

$$h_2 = \tanh(W_{hh}(\tanh(W_{hh} h_0 + W_{xh} x_1)) + W_{xh} x_2)$$

$$h_3 = \tanh(W_{hh}(\tanh(W_{hh}(\tanh(W_{hh} h_0 + W_{xh} x_1)) + W_{xh} x_2)) + W_{xh} x_3)$$

$$h_4 = \underline{\tanh(W_{hh}(\tanh(W_{hh}(\tanh(W_{hh}(\tanh(W_{hh}} h_0 + W_{xh} x_1)) + W_{xh} x_2)) + W_{xh} x$$



$$h_1 = \tanh\left(W_{hh} h_0 + W_{xh} x_1\right)$$

$$h_2 = \tanh\left(W_{hh}\left(\tanh\left(W_{hh} h_0 + W_{xh} x_1\right)\right) + W_{xh} x_2\right)$$

$$h_3 = \tanh\left(W_{hh}\left(\tanh\left(W_{hh}\left(\tanh\left(W_{hh} h_0 + W_{xh} x_1\right)\right) + W_{xh} x_2\right)\right) + W_{xh} x_3\right)$$

$$h_4 = \underline{\tanh\left(W_{hh}\left(\tanh\left(W_{hh}\left(\tanh\left(W_{hh}\left(\tanh\left(W_{hh}\right.\right.\right.\right.\right.\right.\right.} h_0 + W_{xh} x_1\right)\right) + W_{xh} x_2\right)\right) + W_{xh} x$$

Backpropagating this recursive function leads to exploding or vanishing gradients.

Papers

On the difficulty of training recurrent neural networks

Pascanu, Mikolov and Bengio, 2013

http://proceedings.mlr.press/v28/pascanu13.pdf

Learning long-term dependencies with gradient descent is difficult

Bengio, Simard and Frasconi, 1994

https://ieeexplore.ieee.org/document/279181

Untersuchungen zu dynamischen neuronalen Netzen

Hochreiter, 1991

http://people.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf

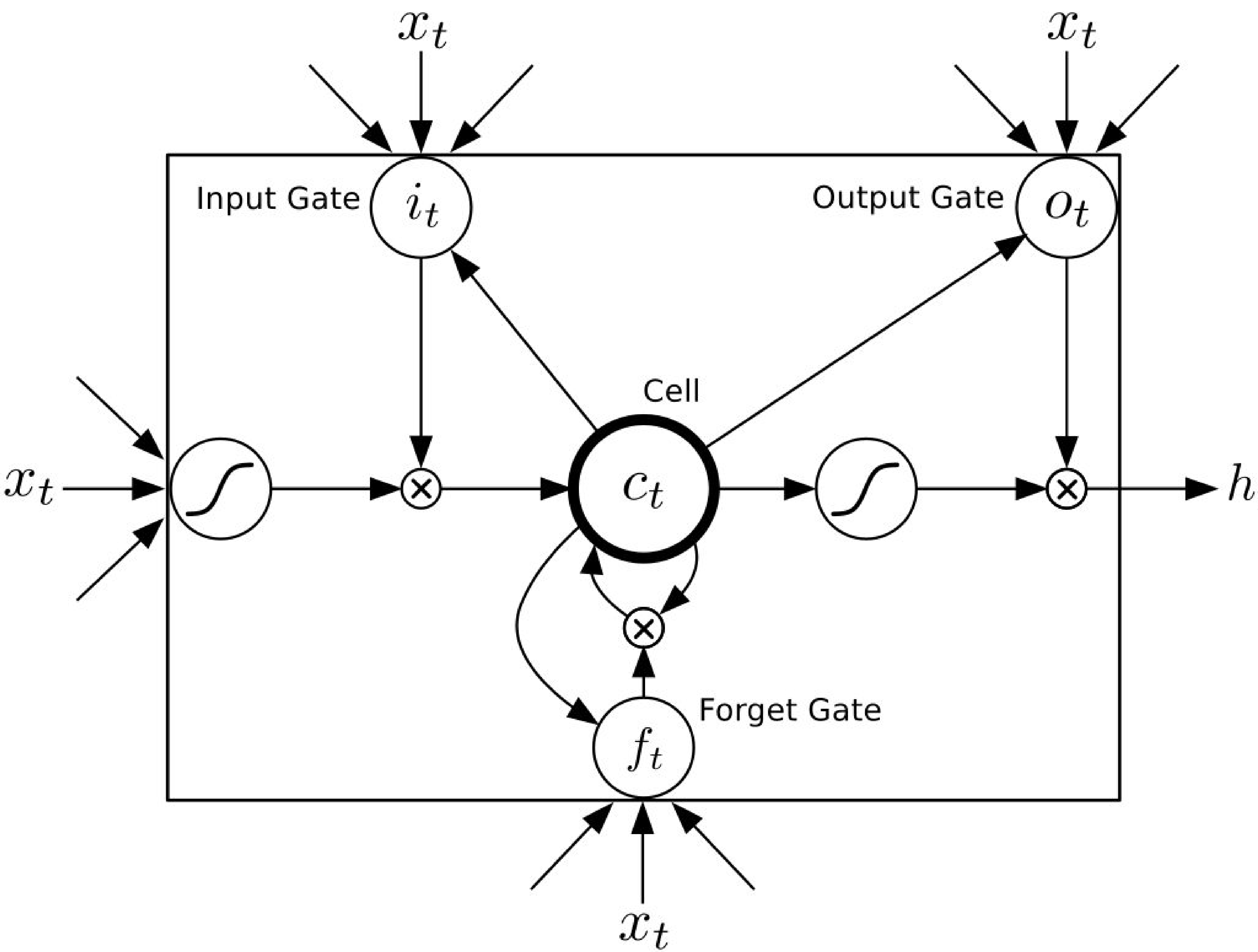
Solutions for this problem:

- Limiting the number of past time steps (Hochreiter, 1991)
- Exploding gradient can be fixed with gradient clipping
- Vanishing gradients can be controlled different architectures (LSTM)
- New: Not using recursions :)

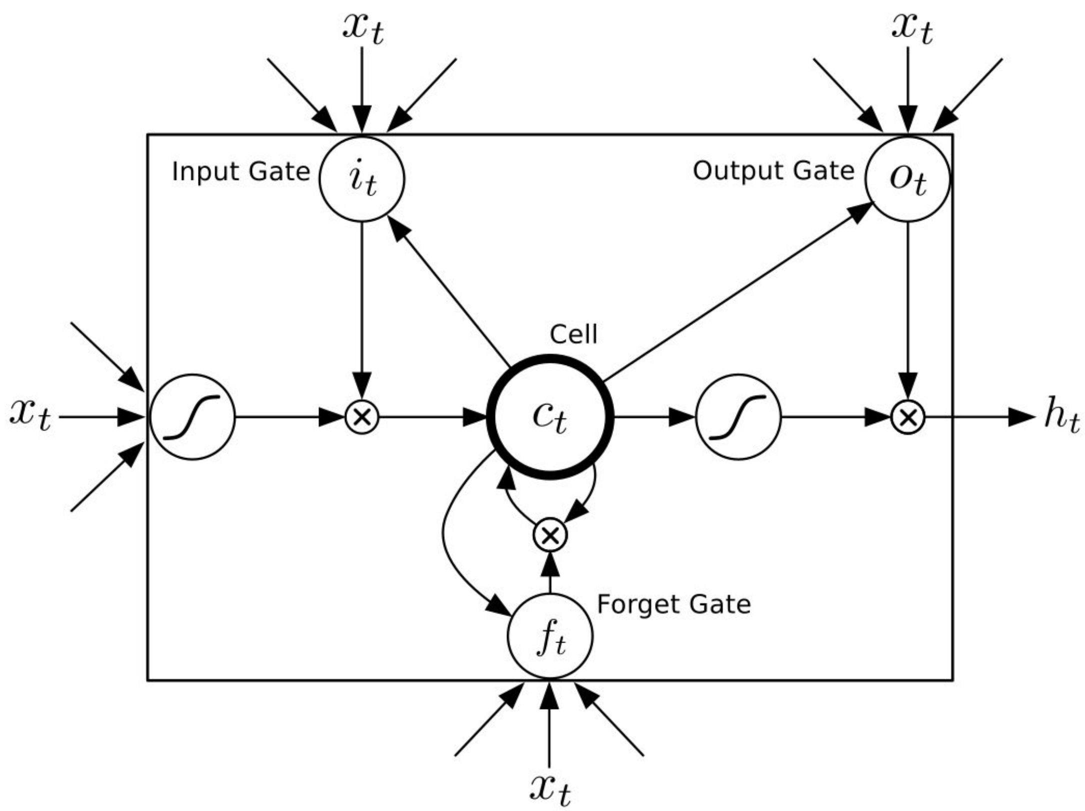
LSTM

Long short-term memory

[Hochreiter et al., 1997]

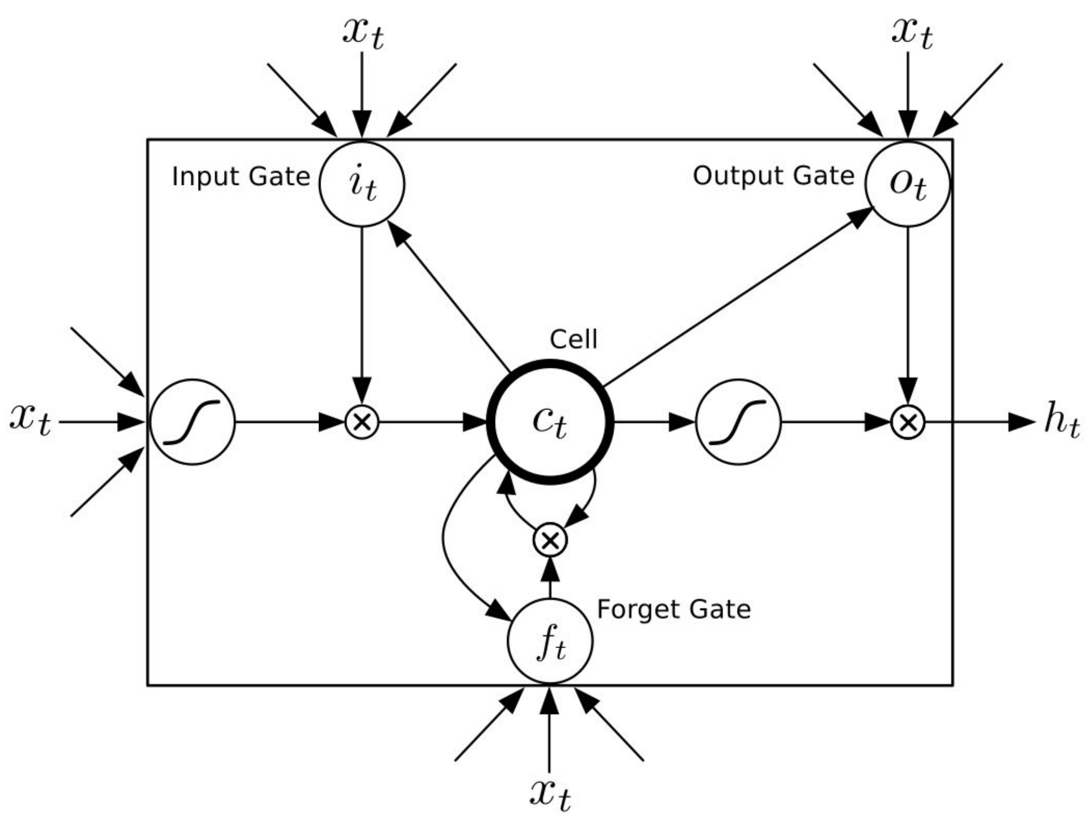


Graves et al. 2013, Speech Recognition with Deep Recurrent Neural Networks



$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$
$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right)$$
$$h_t = o_t \tanh(c_t)$$

Graves et al. 2013, Speech Recognition with Deep Recurrent Neural Networks

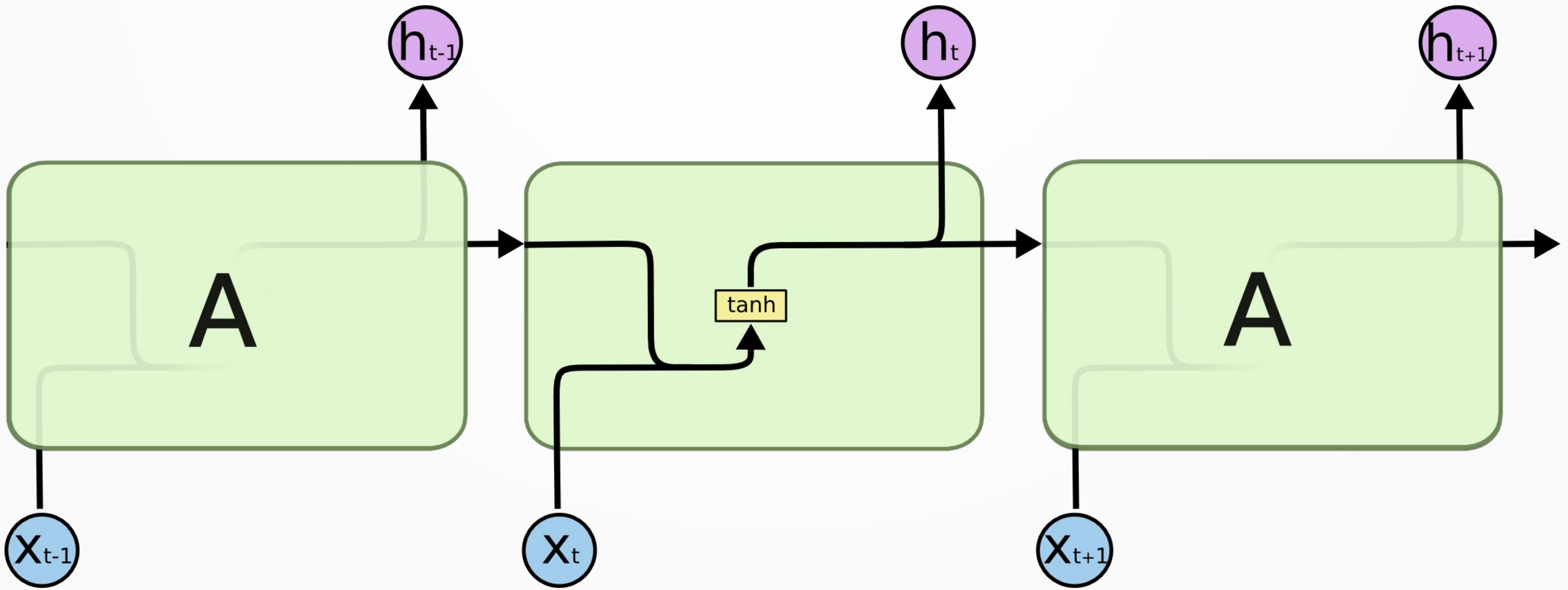


The sigmoid function outputs a number between 0 and 1

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$
$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right)$$
$$h_t = o_t \tanh(c_t)$$

Graves et al. 2013, Speech Recognition with Deep Recurrent Neural Networks

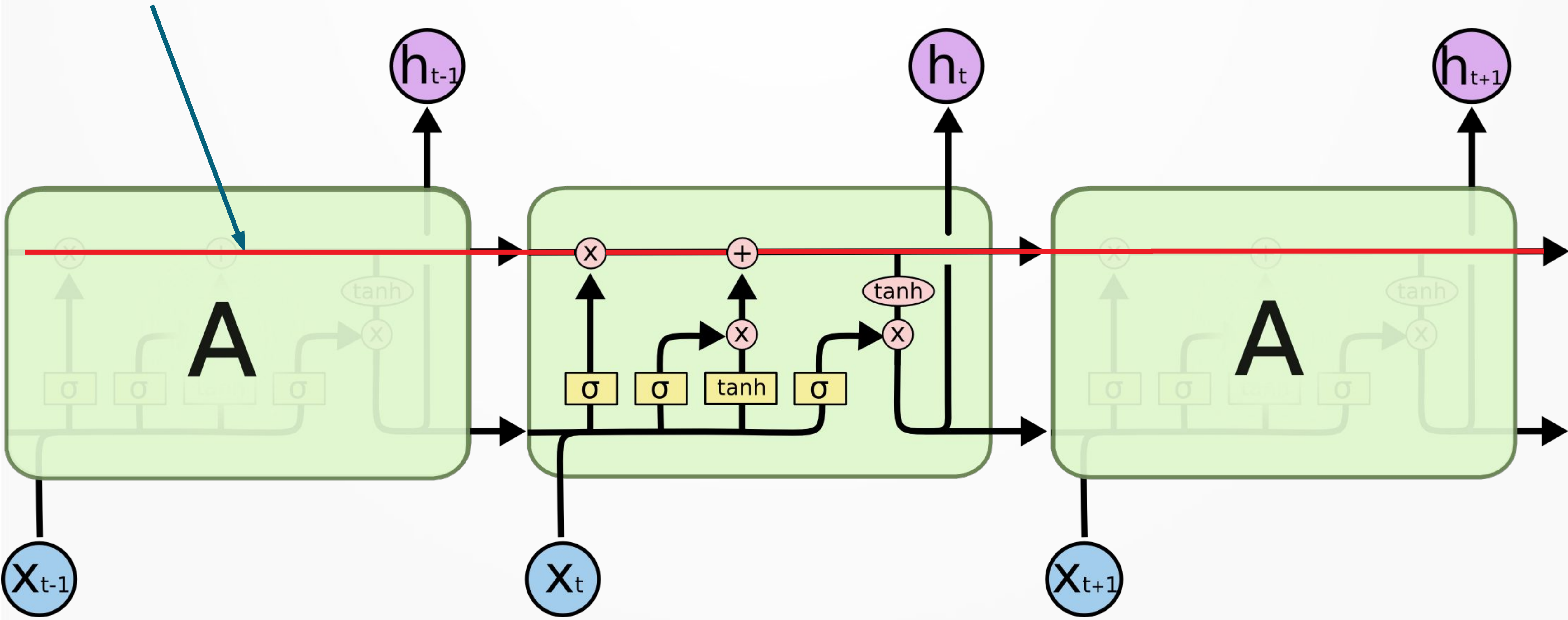
Standard RNN



Source Chris Olah: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM

A LSTM cell can keep its internal state unchanged over many timesteps.
This way a network can learn long term dependencies.



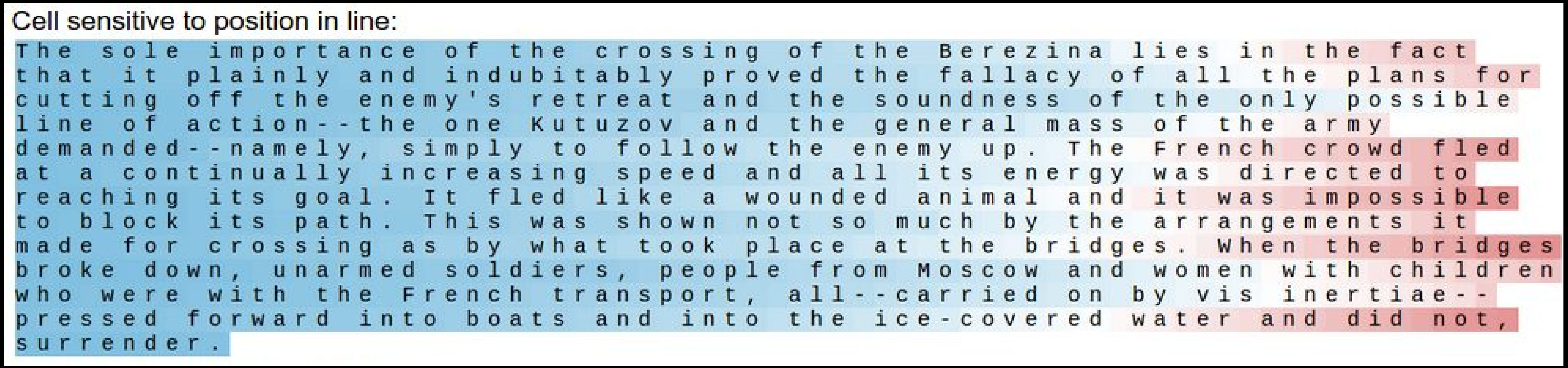
Source Chris Olah: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Further Information

- Chris Olah: Understanding LSTM Networks
- Jürgen Schmidhuber: Tutorial on LSTM Recurrent Networks

- **LSTM: A Search Space Odyssey**
  Klaus Greff, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, Jürgen Schmidhuber 2015
- **Speech Recognition with Deep Recurrent Neural Networks**
  Graves et al. 2013
- **Long Short-term Memory**
  Sepp Hochreiter, Jürgen Schmidhuber

What does the network learn?



Andrej Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Cell that turns on inside quotes:

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

Andrej Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Cell that robustly activates inside if statements:

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
    siginfo_t *info)
{
  int sig = next_signal(pending, mask);
  if (sig) {
    if (current->notifier) {
      if (sigismember(current->notifier_mask, sig)) {
        if (!(current->notifier)(current->notifier_data)) {
          clear_thread_flag(TIF_SIGPENDING);
          return 0;
        }
      }
    }
    collect_signal(sig, pending, info);
  }
  return sig;
}
```

Andrej Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/

A large portion of cells are not easily interpretable. Here is a typical example:

```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
  char *str;
  if (!*bufp || (len == 0) || (len > *remain))
    return ERR_PTR(-EINVAL);
  /* Of the currently implemented string fields, PATH_MAX
   * defines the longest valid length.
   */
```

Andrej Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/

What does the network learn?

- Visualizing the predictions and the “neuron” firings in the RNN
- Set the background color based on the neurons activation
- Literature tip:
  - http://karpathy.github.io/2015/05/21/rnn-effectiveness
- Paper:
  - Karpathy, Johnson, and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016

Other applications for RNNs

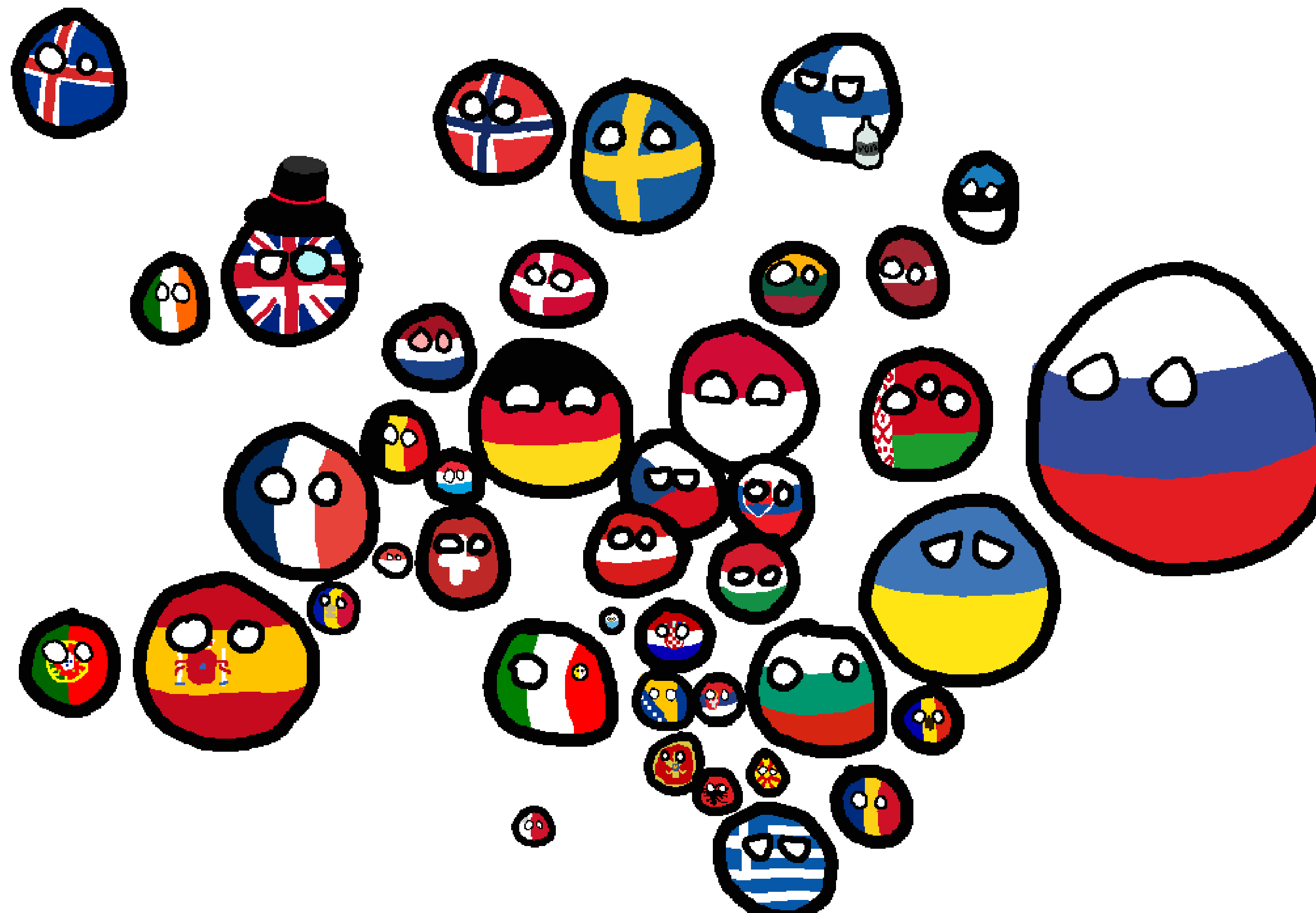
- Time Series Prediction
- Speech Recognition
  - Speech Recognition with Deep Recurrent Neural Networks Graves et al. 2013
- Drawing (Pictures, handwriting)
  - Generating Sequences With Recurrent Neural Networks Graves 2013
- Music Generation
  - Song From PI: A Musically Plausible Network for Pop Music Generation

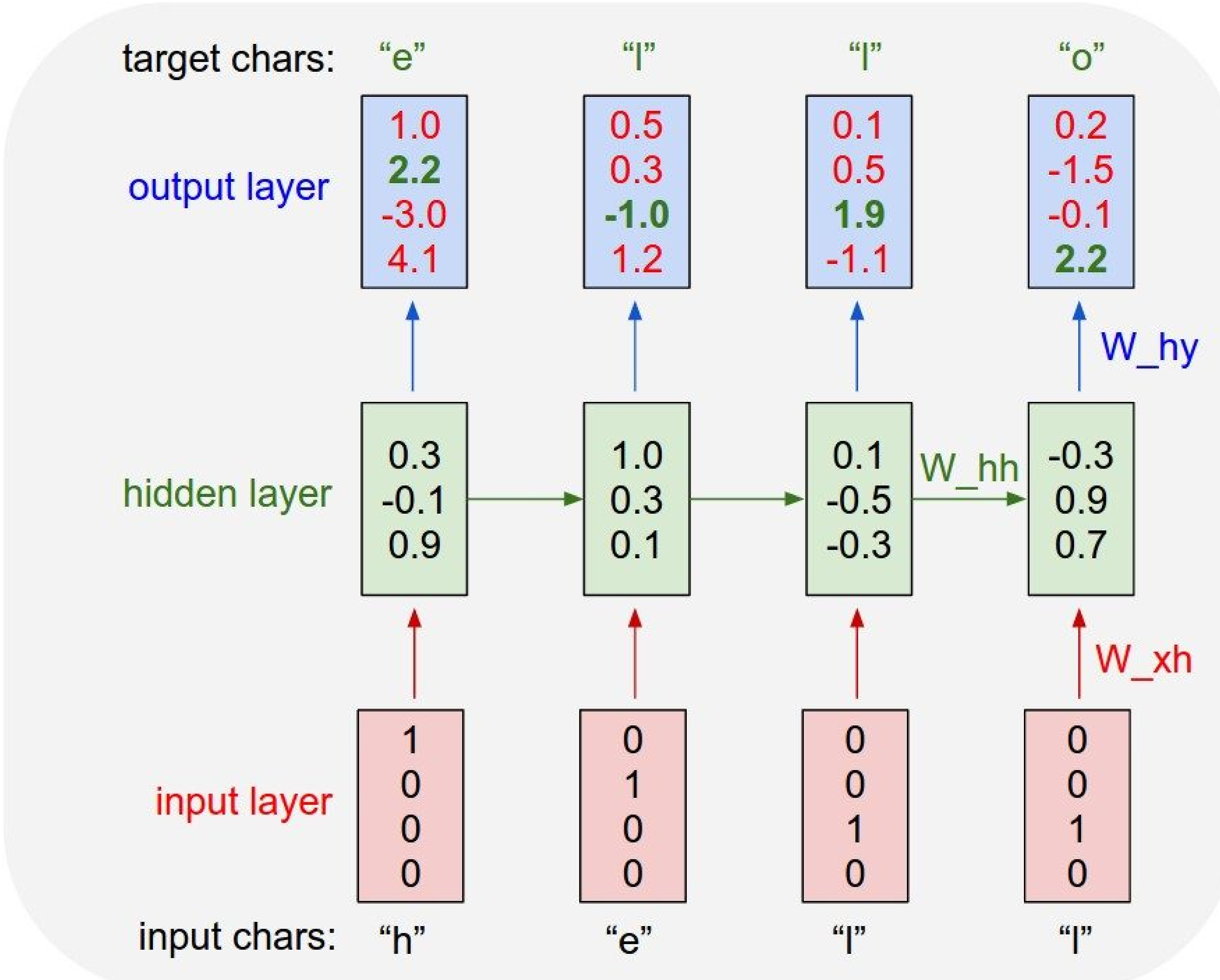
Summary

- RNNs are used to deal with sequential data
- They have applications in many other domains:
  - Speech Recognition
  - Time Series Prediction
  - Drawing (Pictures, Handwriteting)
- During Backpropagation gradients can explode or vanish
- Deep NLU is a hot topic of research (Bert, GTP-2)

Tutorial

Guess where is my name from?





Andrej Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Tutorial

- You will classify names using a character-level RNN
- Take a look at the notebook for your tasks
- What you will learn:
  - One Hot Encoding
  - Character level RNNs
  - How to use LSTM and GRUs