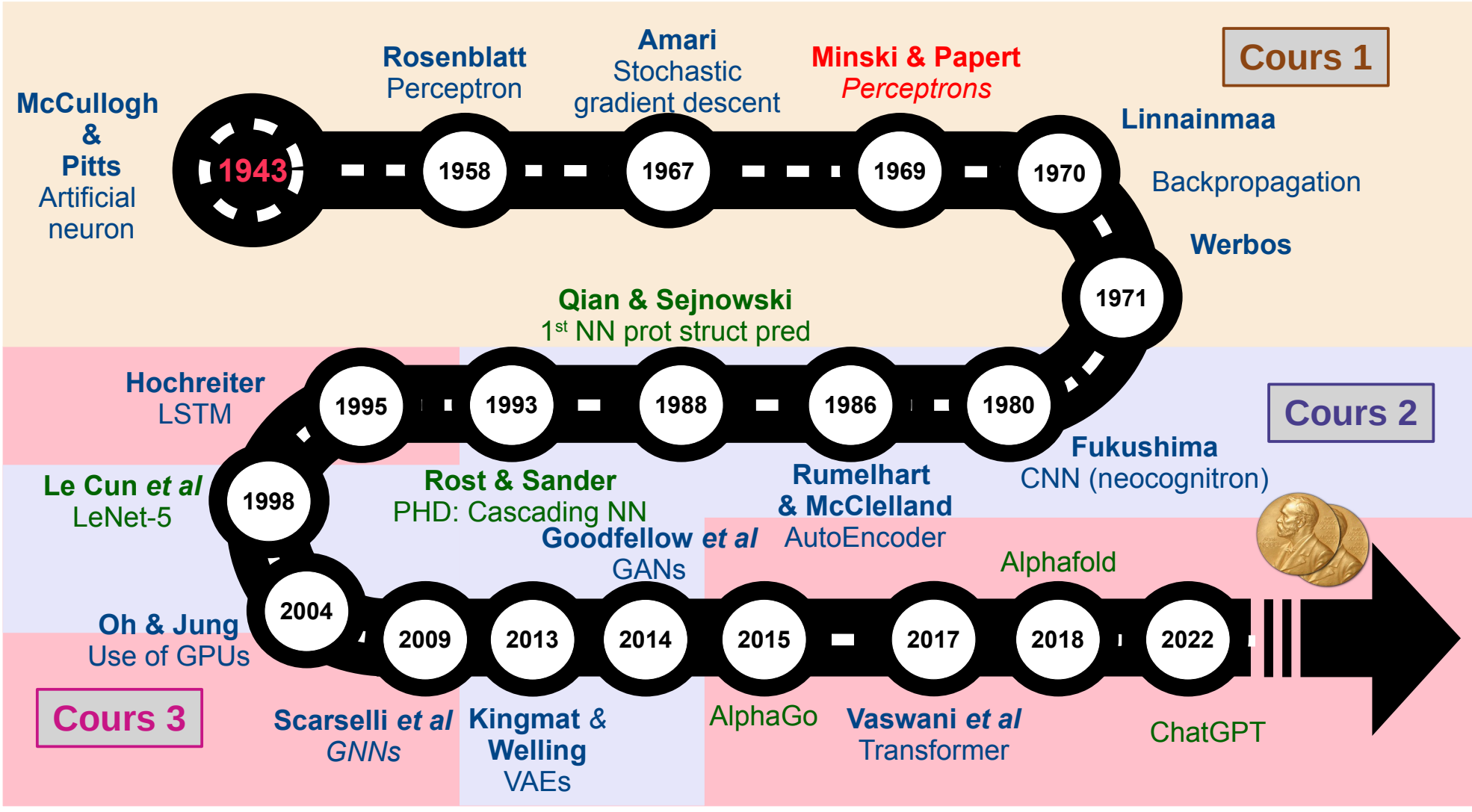
Beyond MLPs

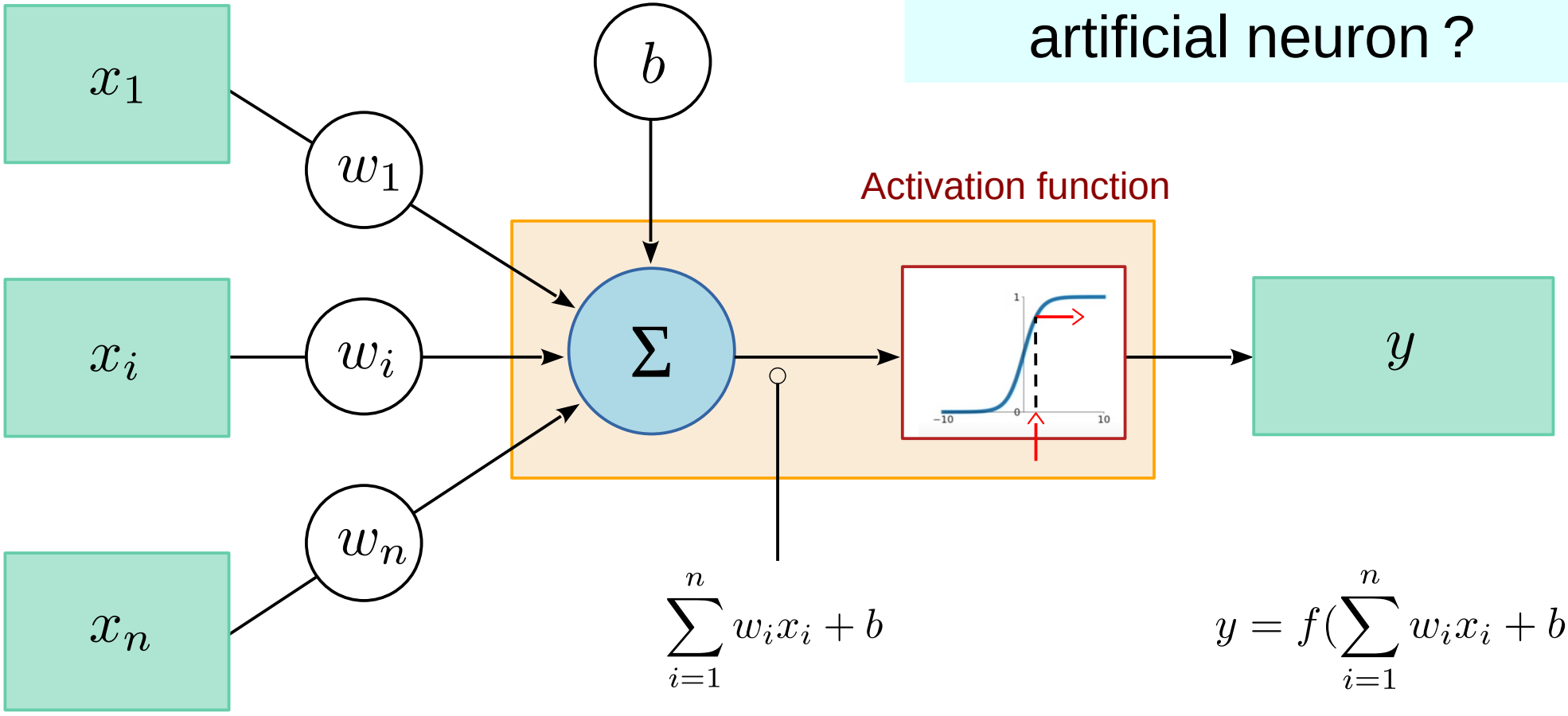
Part 2/2: RNNs, Attention and GNNs

Nicolas Gambardella

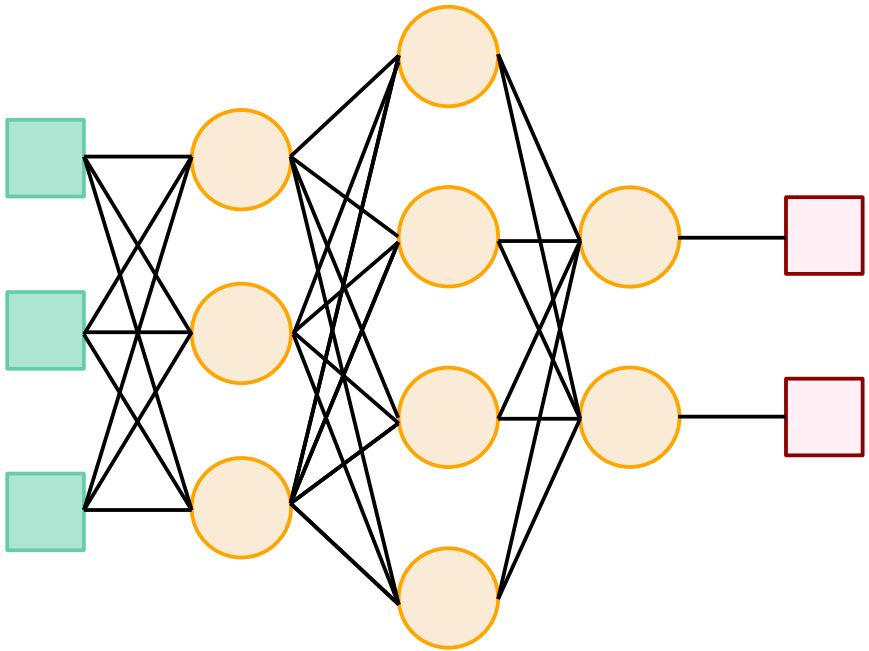
nicolas.gambardella@univ-lille.fr


Cours 1
McCullogh & Pitts
Artificial neuron
1943
Rosenblatt
Perceptron
1958
Amari
Stochastic gradient descent
1967
Minski & Papert
Perceptrons
1969
Linnainmaa
Backpropagation
1970
Werbos
1971
Qian & Sejnowski
1st NN prot struct pred
1980
Rumelhart & McClelland
AutoEncoder
1986
Fukushima
CNN (neocognitron)
Cours 2
Hochreiter
LSTM
1988
1993
Rost & Sander
PHD: Cascading NN
1995
Le Cun et al
LeNet-5
1998
Goodfellow et al
GANs
2004
Oh & Jung
Use of GPUs
2009
Scarselli et al
GNNs
2013
Kingmat & Welling
VAEs
2014
AlphaGo
2015
Vaswani et al
Transformer
2017
Alphafold
2018
ChatGPT
2022
Cours 3

What is an artificial neuron ?

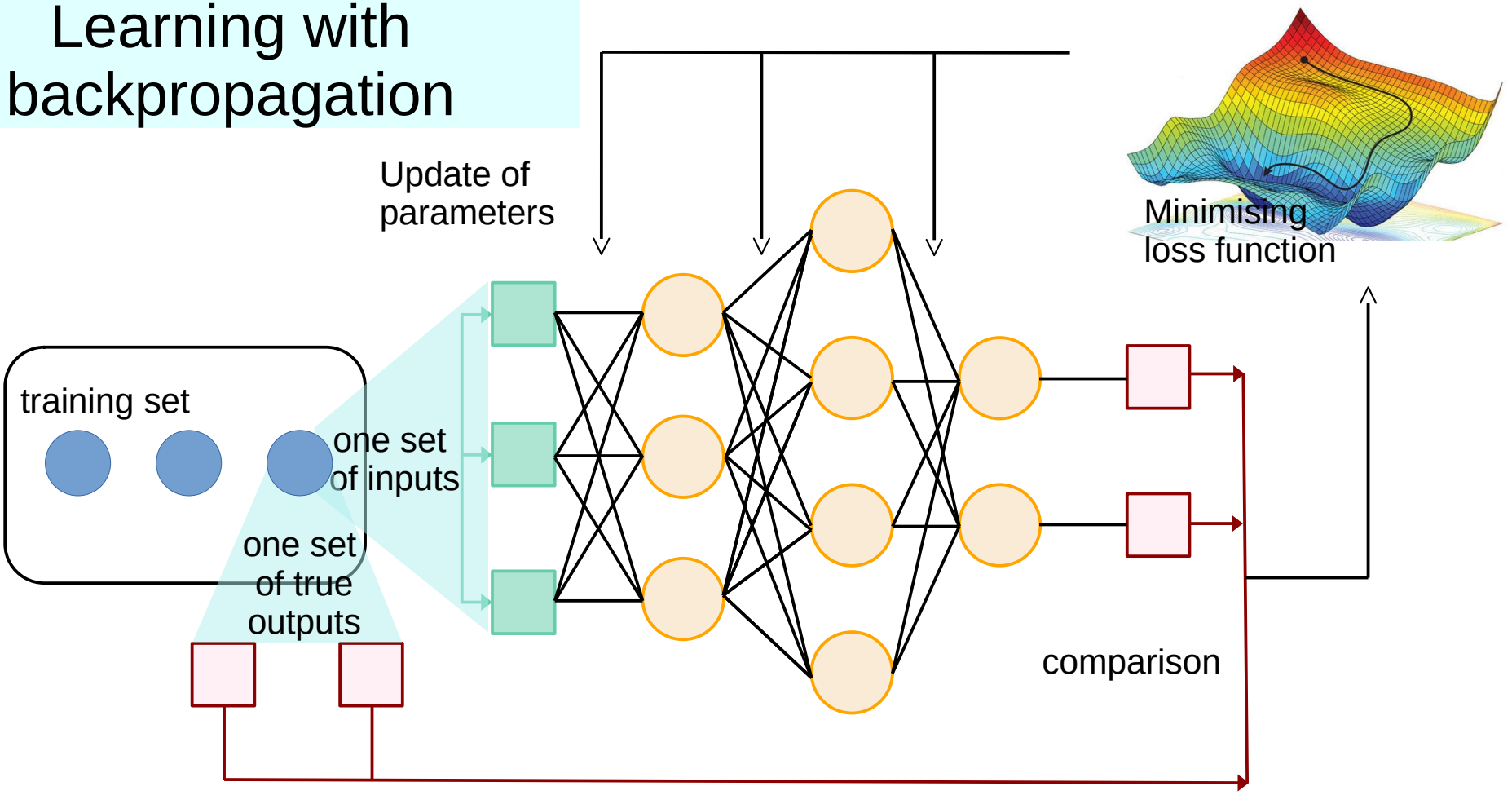


All inputs can be independent and everything connected to everything

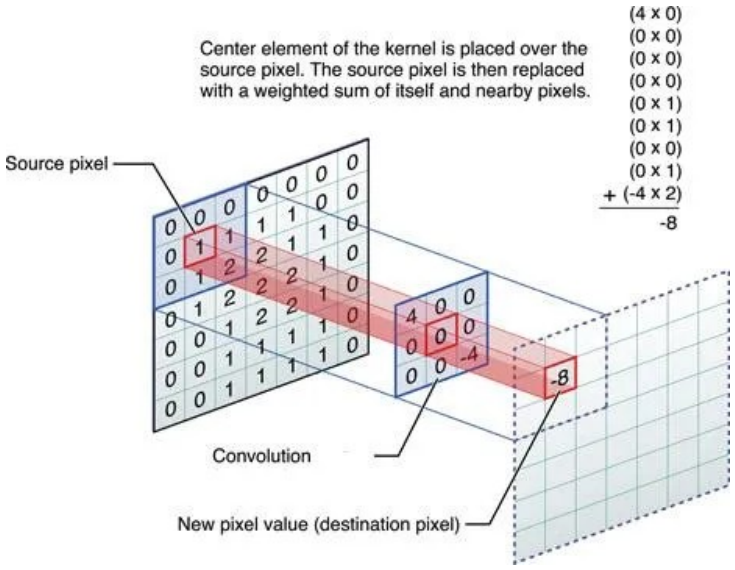


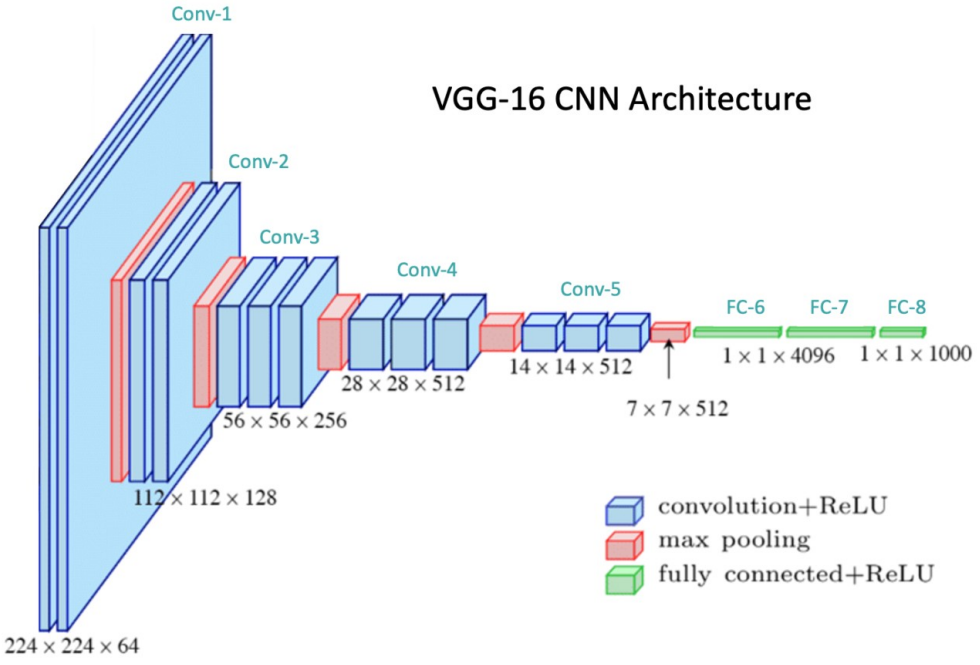
Multi-Layer Perceptrons (MLP)
or Dense neural networks (DNN)
made of Fully Connected layers (FC)

Learning with backpropagation



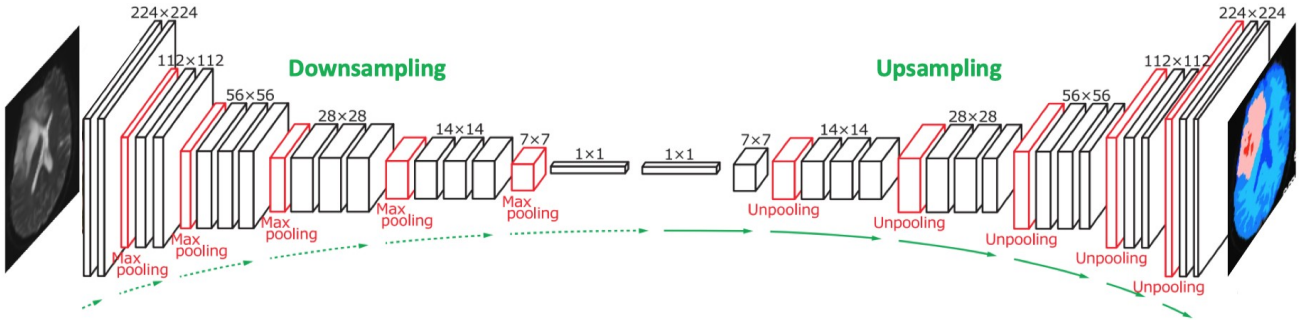
We can detect local features by linking neighbouring inputs

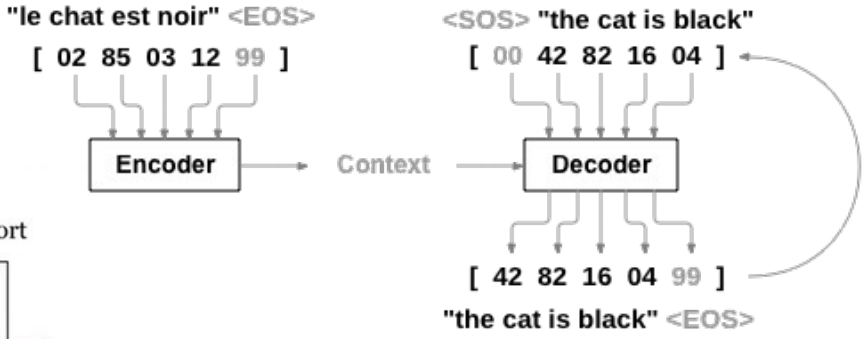


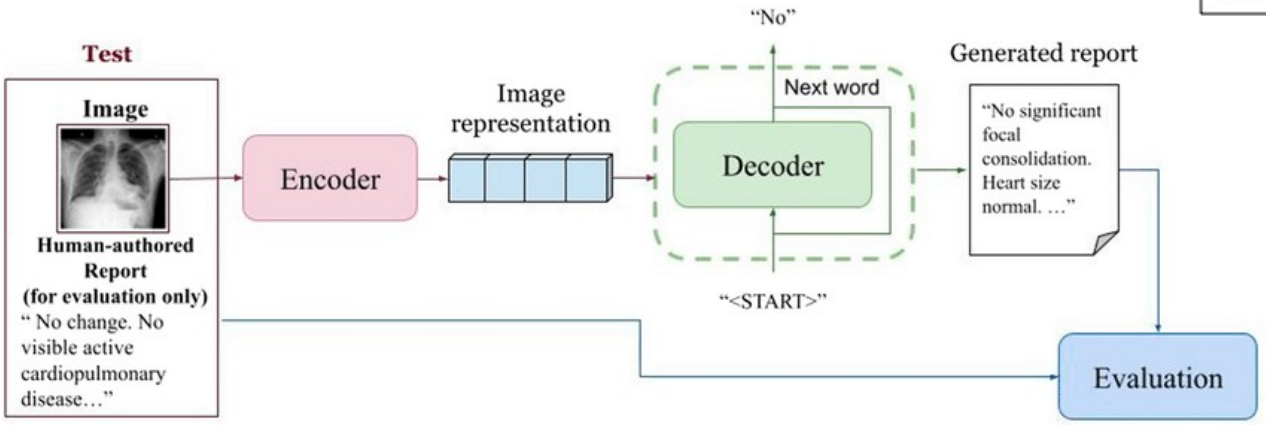


Deep Convolutional Neural Networks (CNN)

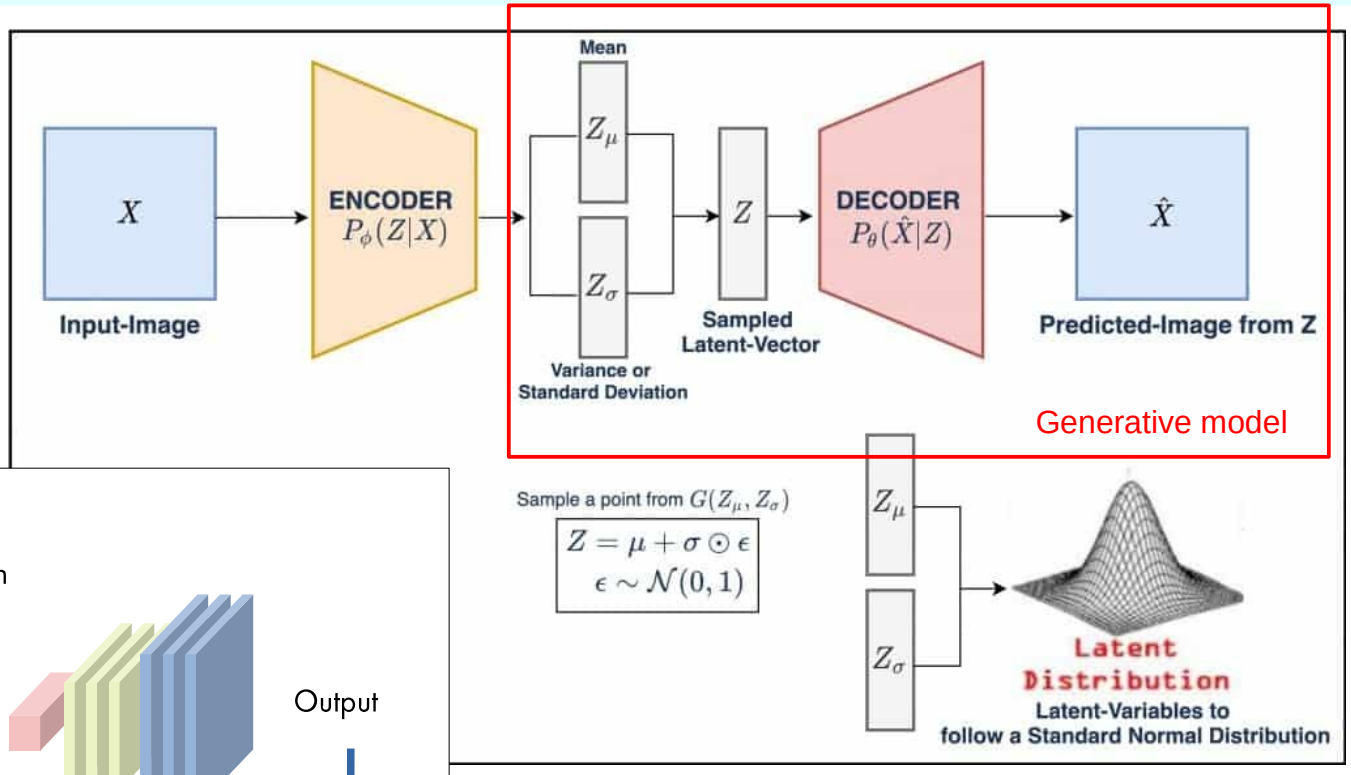
Encoder networks can embed information in a latent space
Decoder networks can reconstruct the information from it

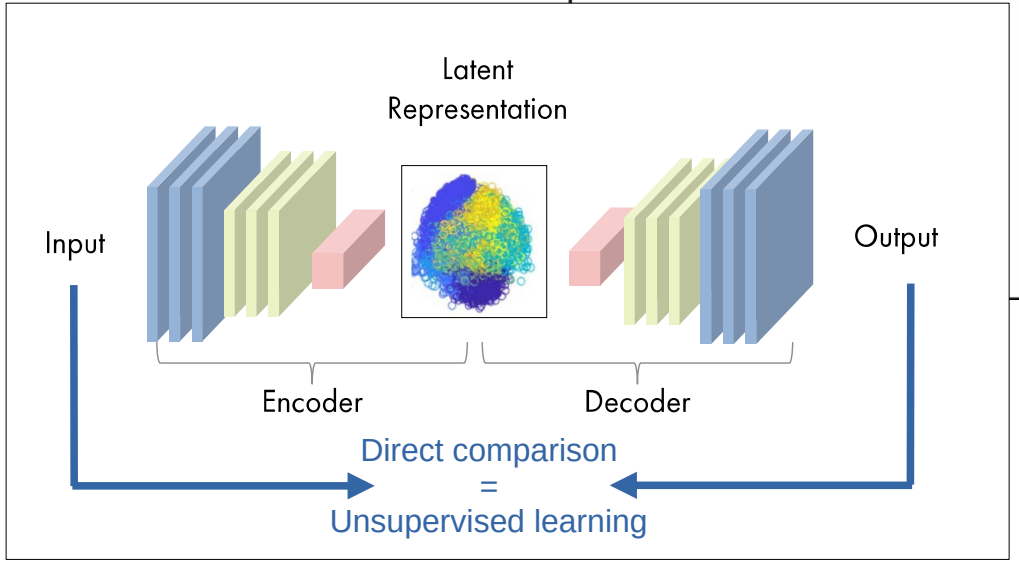




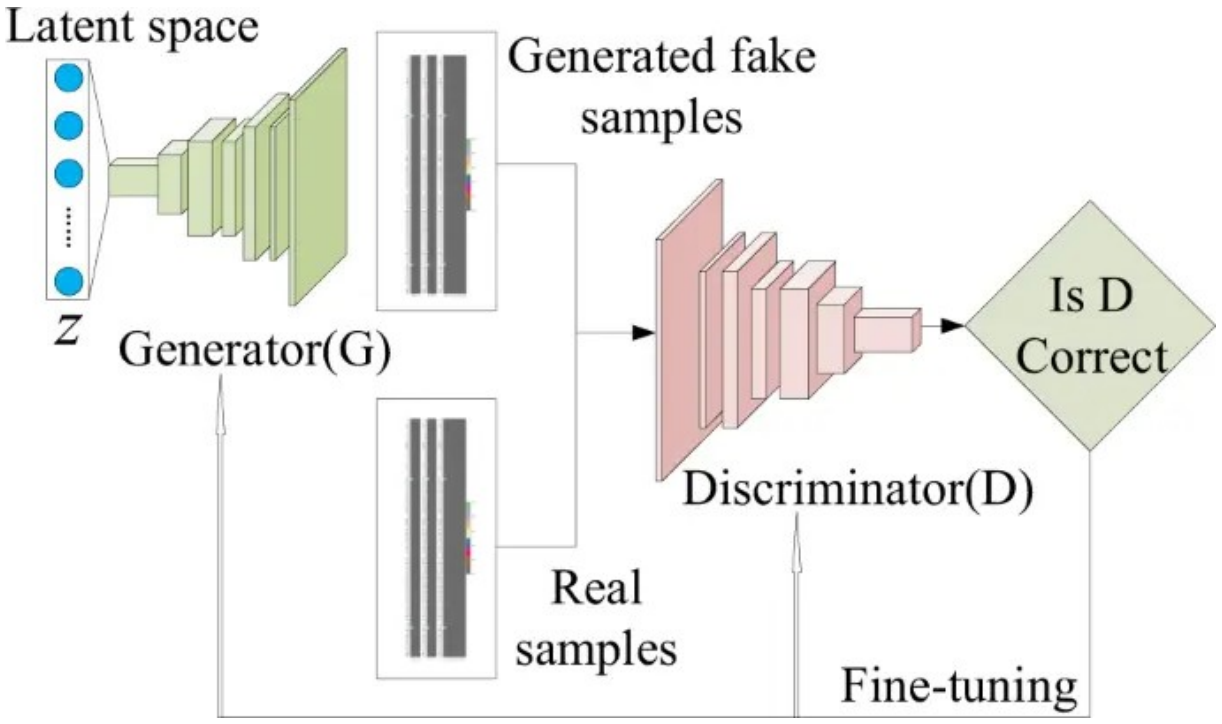


AutoEncoders can train themselves unsupervised
Variational AutoEncoders learn mean and std distributions



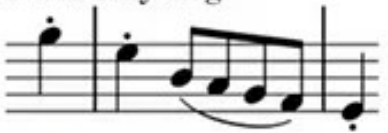


Generative Adversarial Networks learn by trying to deceive themselves



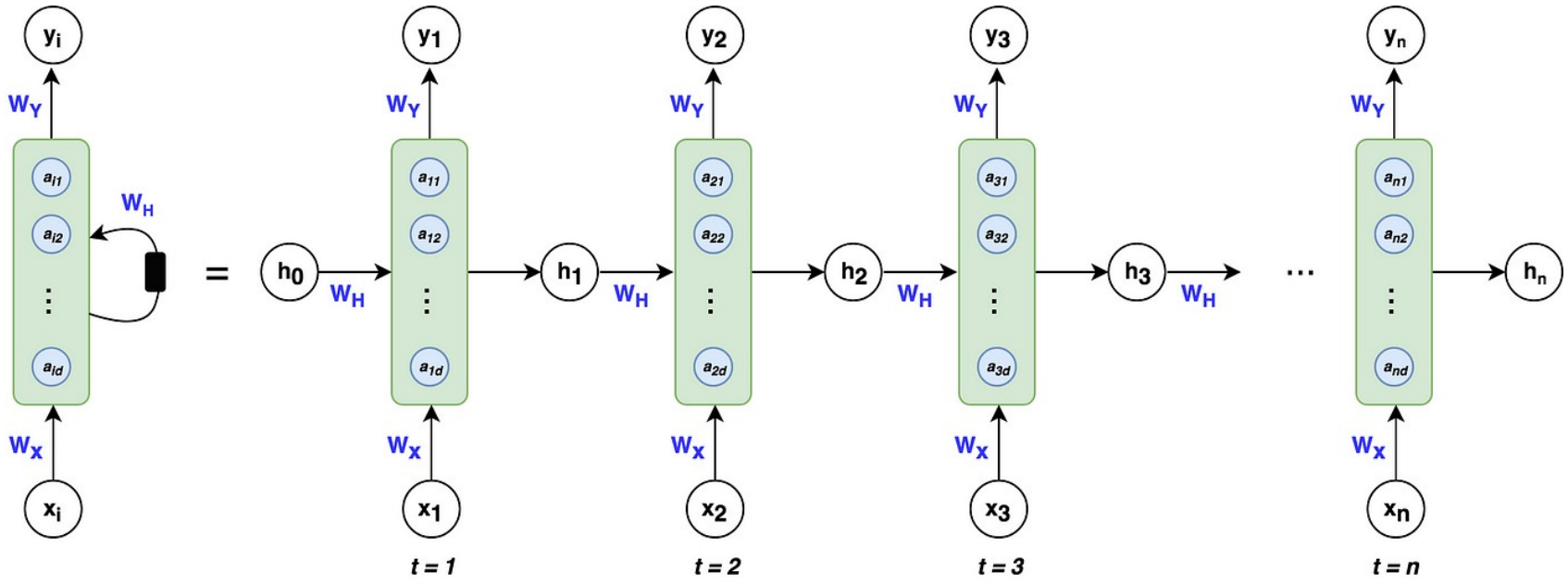
Time series and sequences of variable lengths

Time series and sequences of variable lengths

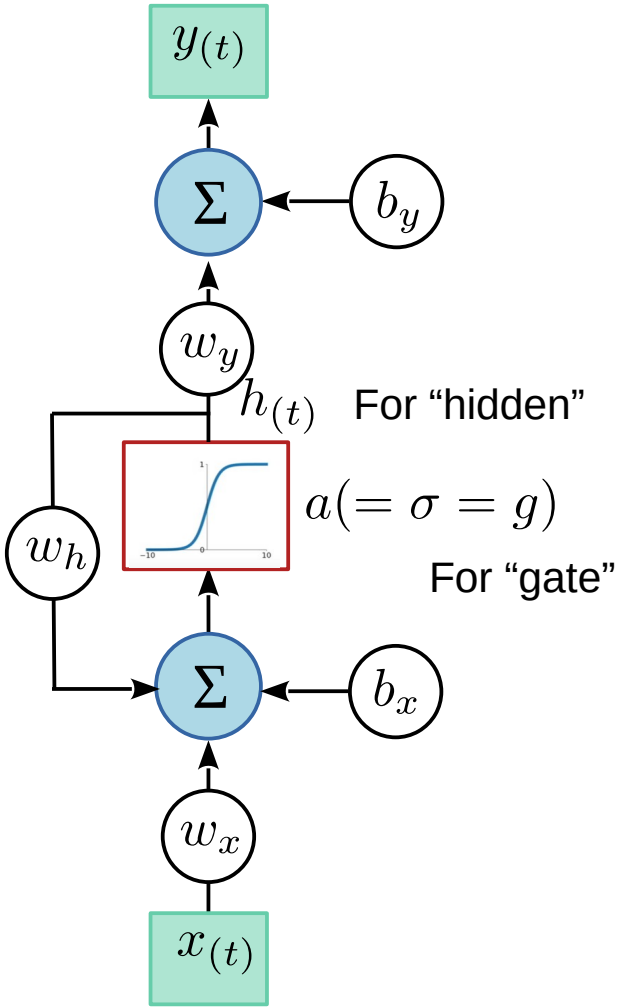
Speech recognition		→	"The quick brown fox jumped over the lazy dog."
Music generation		→	
Sentiment classification	"There is nothing to like in this movie."	→	
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	→	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	→	Do you want to sing with me?
Video activity recognition		→	Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	→	Yesterday, Harry Potter met Hermione Granger.

Andrew Ng

Recurrent Neural Networks: successive inputs are not independent



RNN: 1 cell (here, 1 neuron)



NB: implicit “identity” activation function

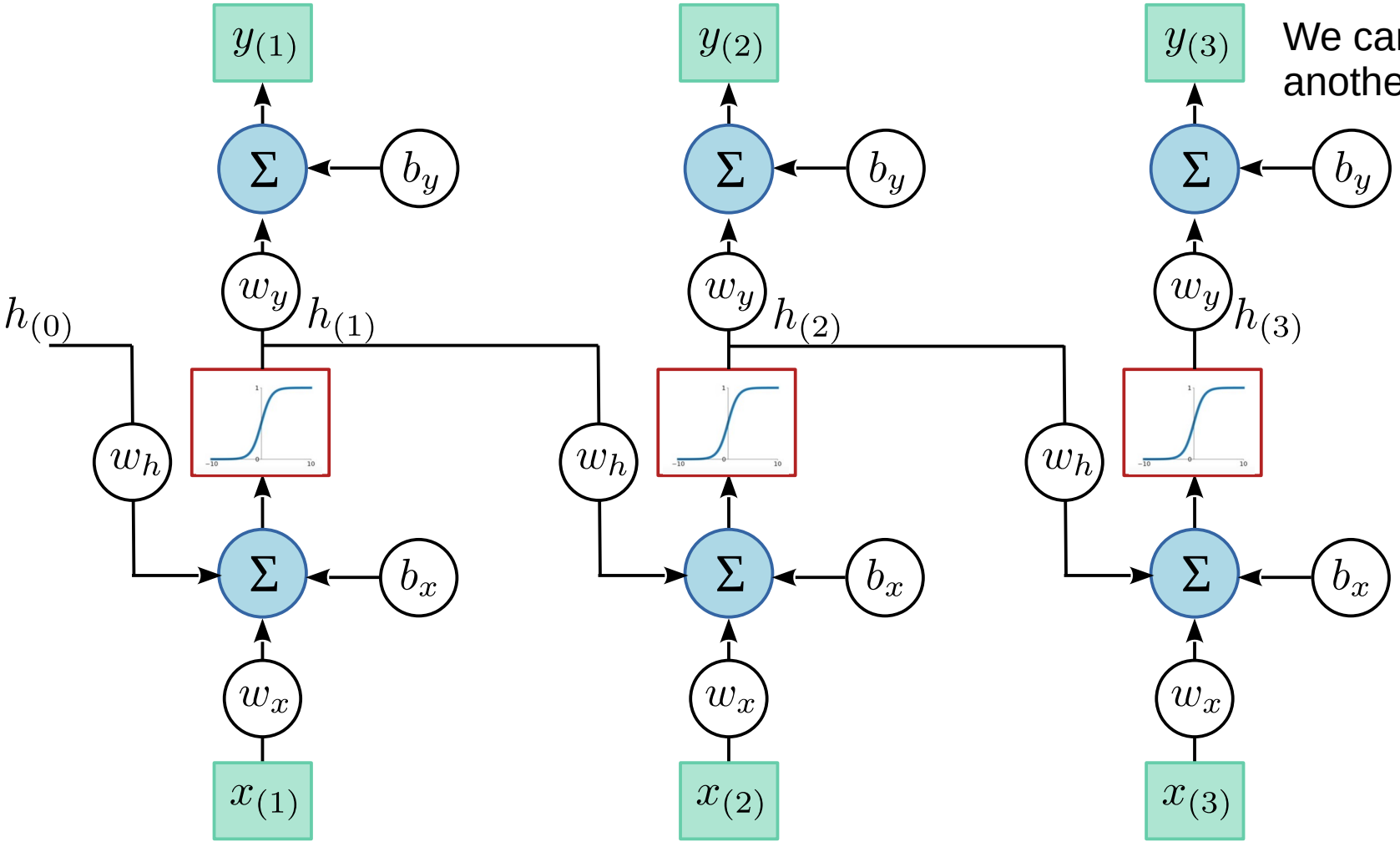
$$y_{(t)} = w_y \times h_{(t)} + b_y$$

not the same timepoint

$$h_{(t)} = a(w_x \times x_{(t)} + b_x + w_h \times h_{(t-1)})$$

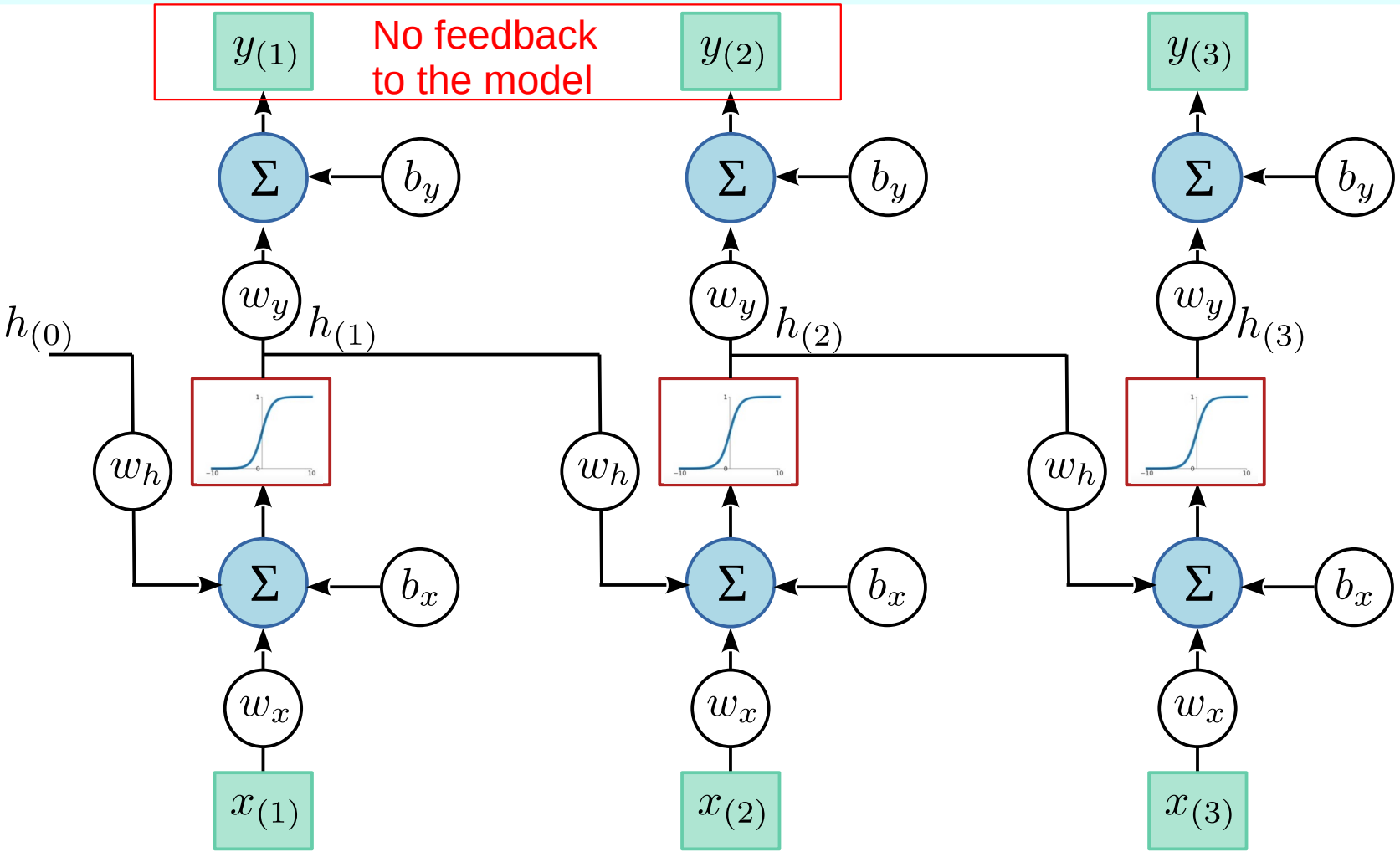
“memory”

RNN: 1 cell - unfolding



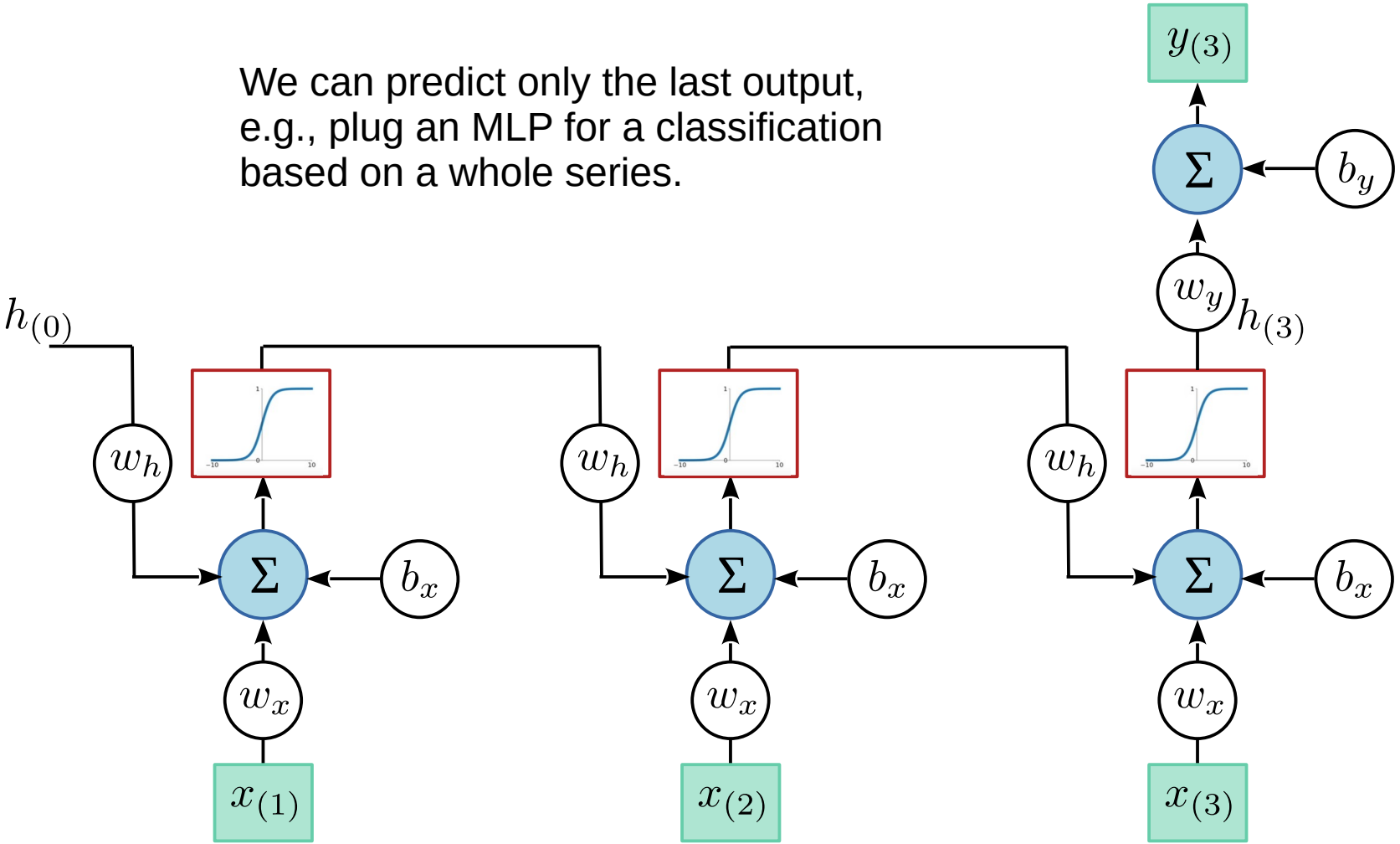
We can output another series, e.g, transcription, translation, conversion.

RNN: 1 cell - unfolding

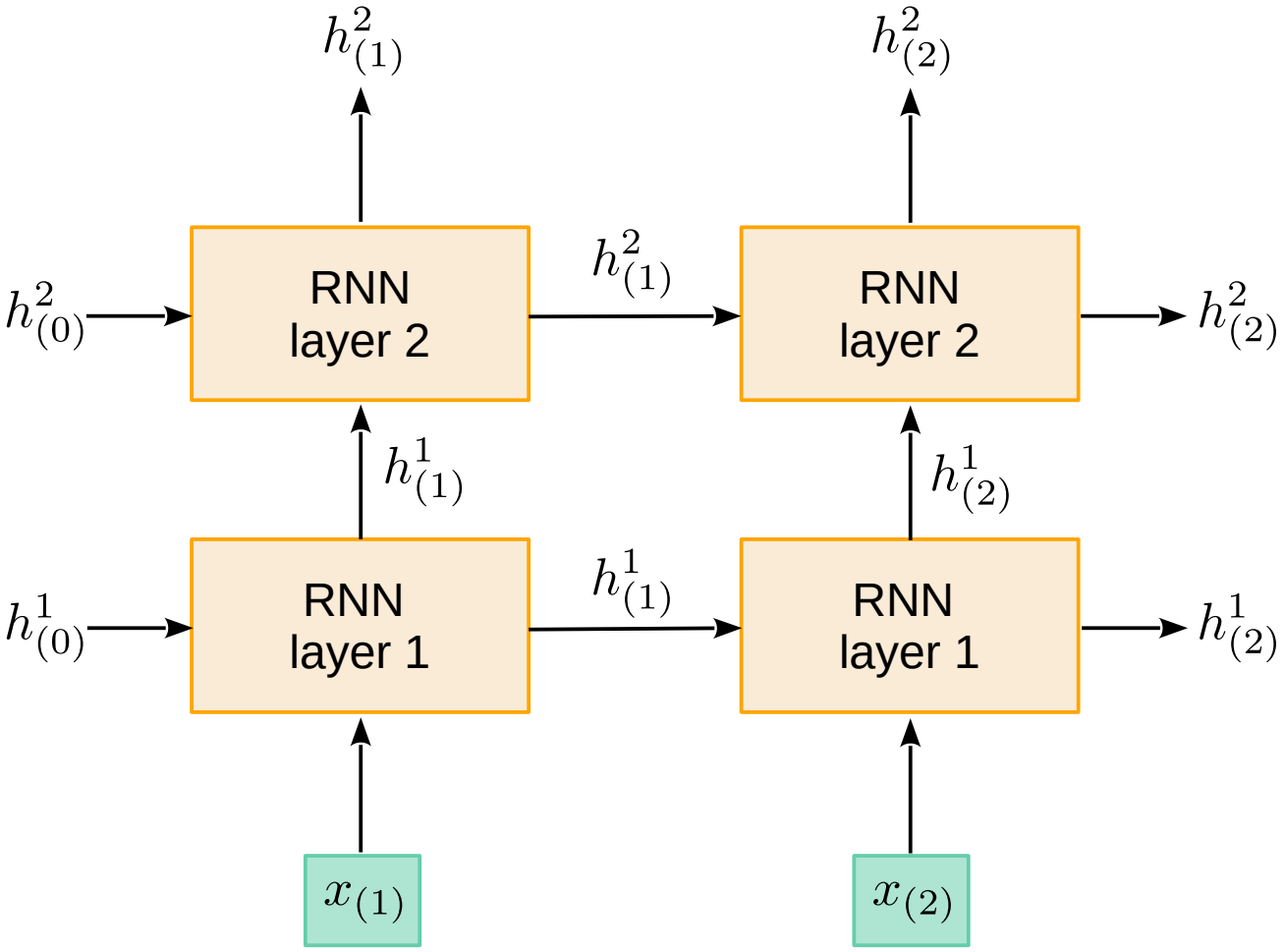


RNN: 1 cell - unfolding

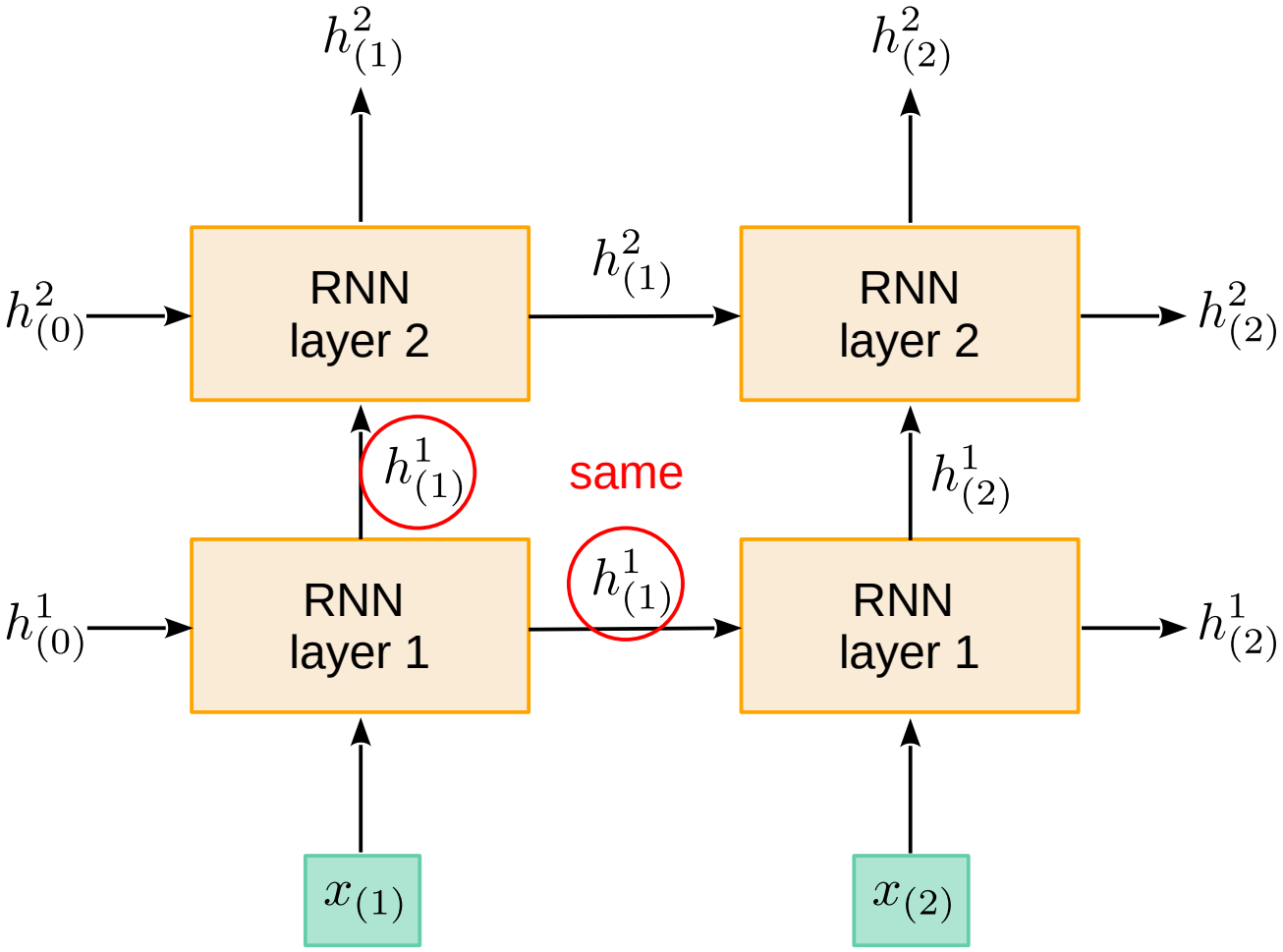
We can predict only the last output, e.g., plug an MLP for a classification based on a whole series.



Stacked RNNs

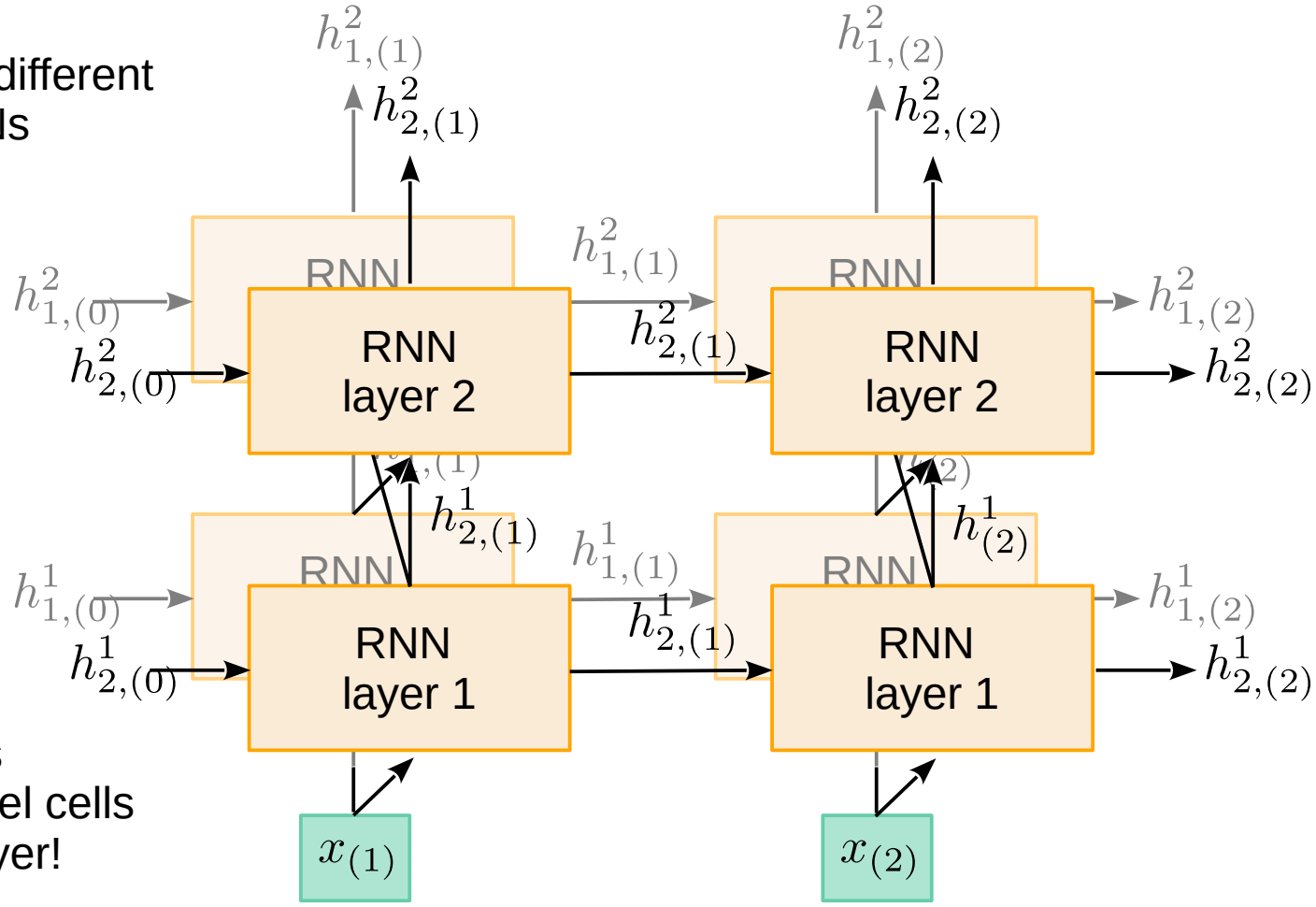


Stacked RNN


$h^2_{(1)}$
$h^2_{(2)}$
$h^2_{(0)}$
RNN layer 2
$h^2_{(1)}$
RNN layer 2
$h^2_{(2)}$
$h^1_{(1)}$
same
$h^1_{(2)}$
$h^1_{(1)}$
$h^1_{(0)}$
RNN layer 1
RNN layer 1
$h^1_{(2)}$
$x_{(1)}$
$x_{(2)}$

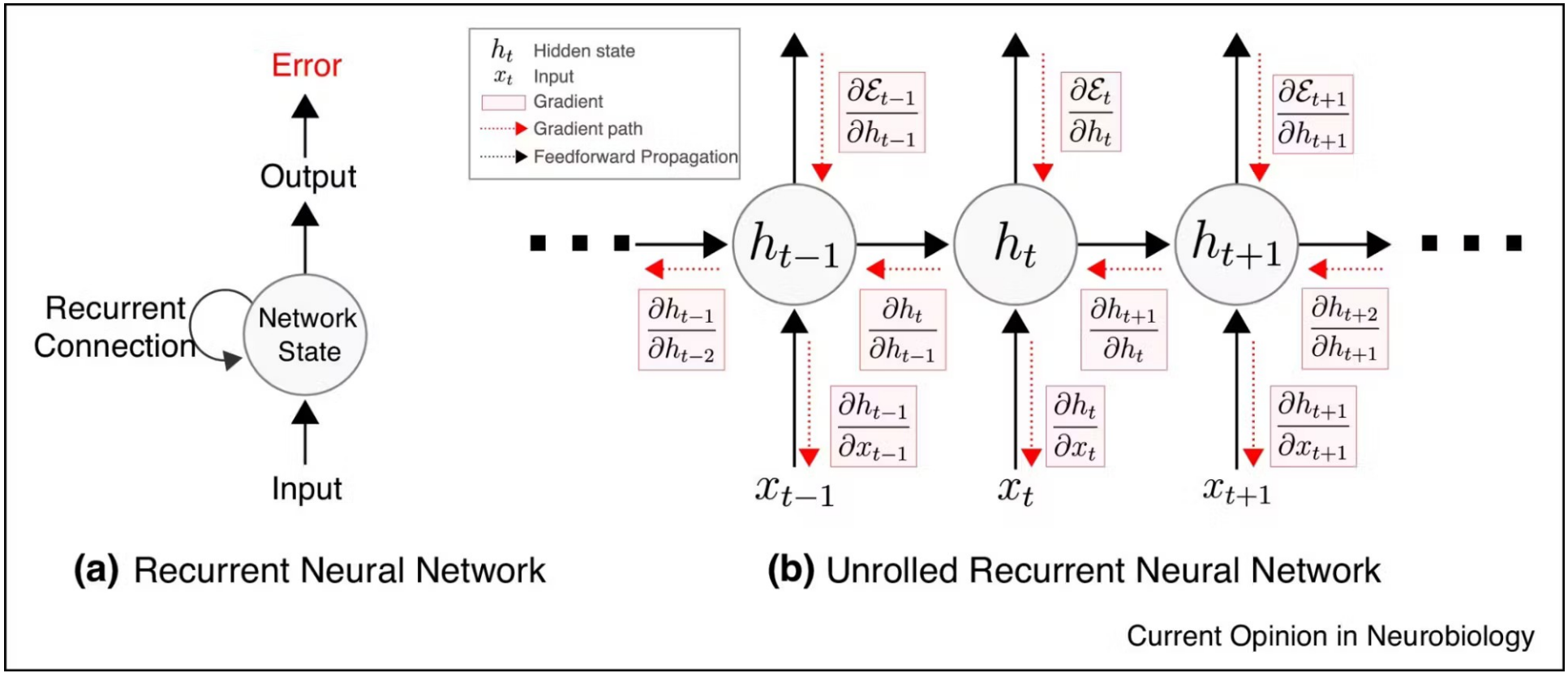
Several RNNs may learn different patterns in parallel

Same idea as different kernels in CNNs



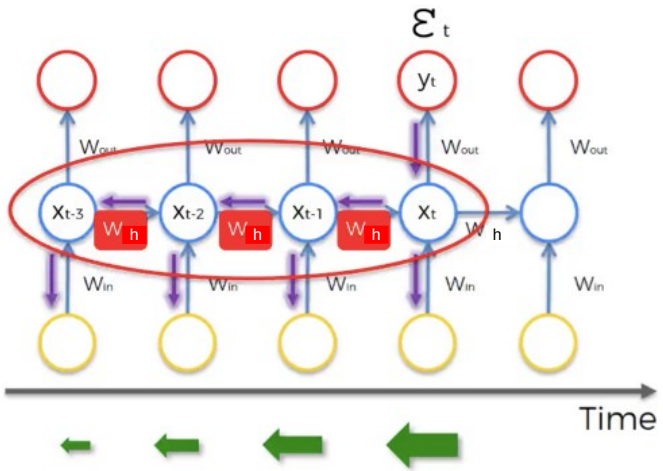
No connexions between parallel cells of the same layer!

Learning by backpropagation in time



Lillicrap and Santaro (2019) Backpropagation through time and the brain. *Curr Op Neurobiol*, 55:81-89

Exploding and vanishing gradients



$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{1 \le t \le T} \frac{\partial \mathcal{E}_t}{\partial \theta}$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{1 \le k \le t} \left(\frac{\partial \mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} \frac{\partial^+ \mathbf{x}_k}{\partial \theta} \right)$$

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \ge i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \prod_{t \ge i > k} \mathbf{W}_h^T \cdot diag(\sigma'(\mathbf{x}_{i-1}))$$

W_h ~ small ⇨ Vanishing

W_h ~ large ⇨ Exploding

E.g.: activation function = ReLU

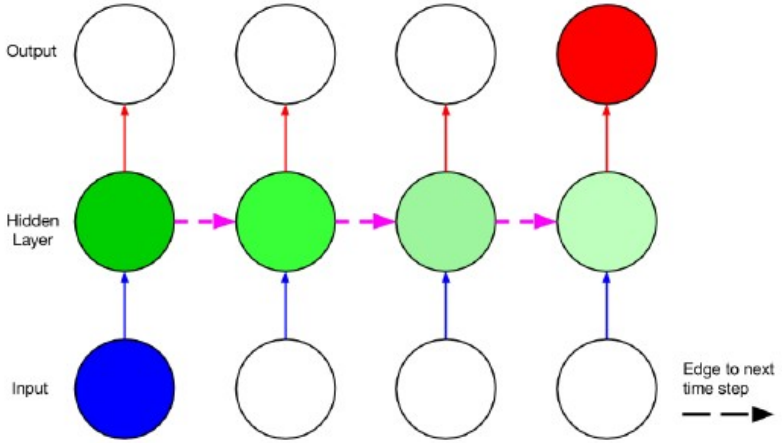
$$\frac{\partial x_i}{\partial x_{i-1}} = w_h$$

$$w_h = 0.1; \frac{\partial x_{10}}{\partial x_1} = w_h^{10} = 0.0000000001$$

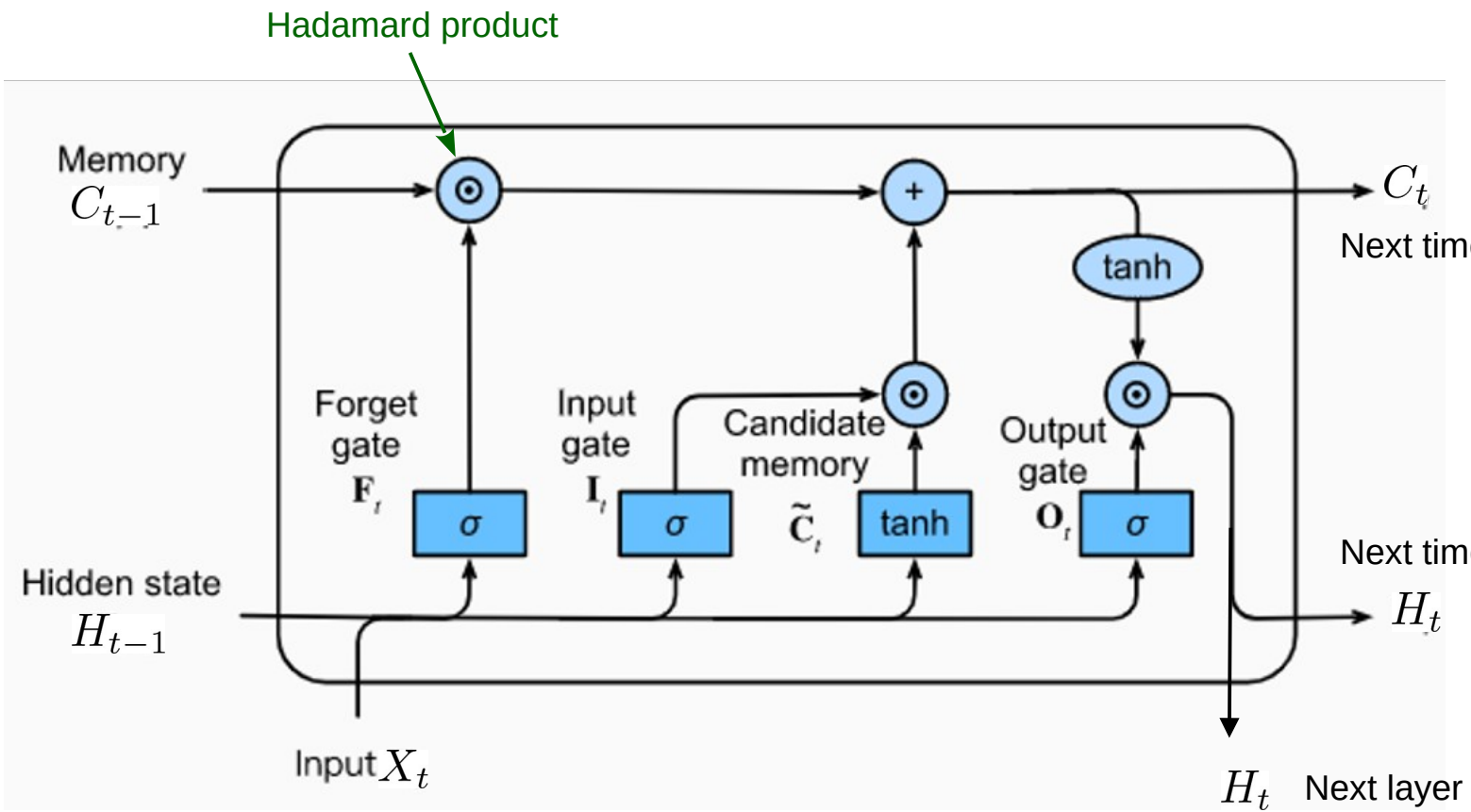
$$w_h = 10; \frac{\partial x_{10}}{\partial x_1} = w_h^{10} = 10000000000$$

Source:
SuperDataScience

Green = sensitivity of output on input



Solution: Long Short-Term Memory (LSTM)

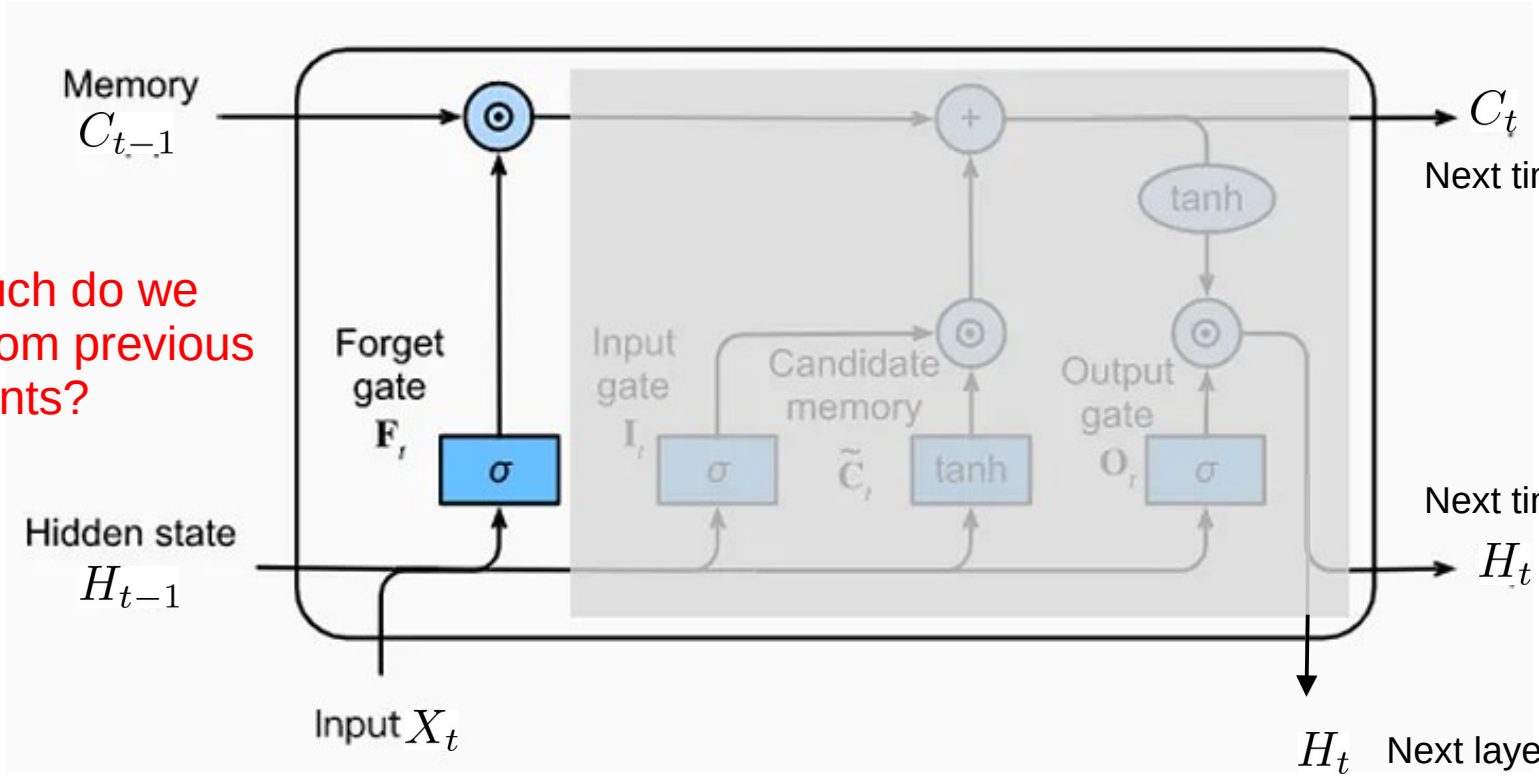


Hochreiter and Schmidhuber (1997) Long short-term memory. *Neur Comput*, 9(8):1735-1780

Source: Ottavio Calzone (2002) An Intuitive Explanation of LSTM. https://medium.com/@ottaviocalzone

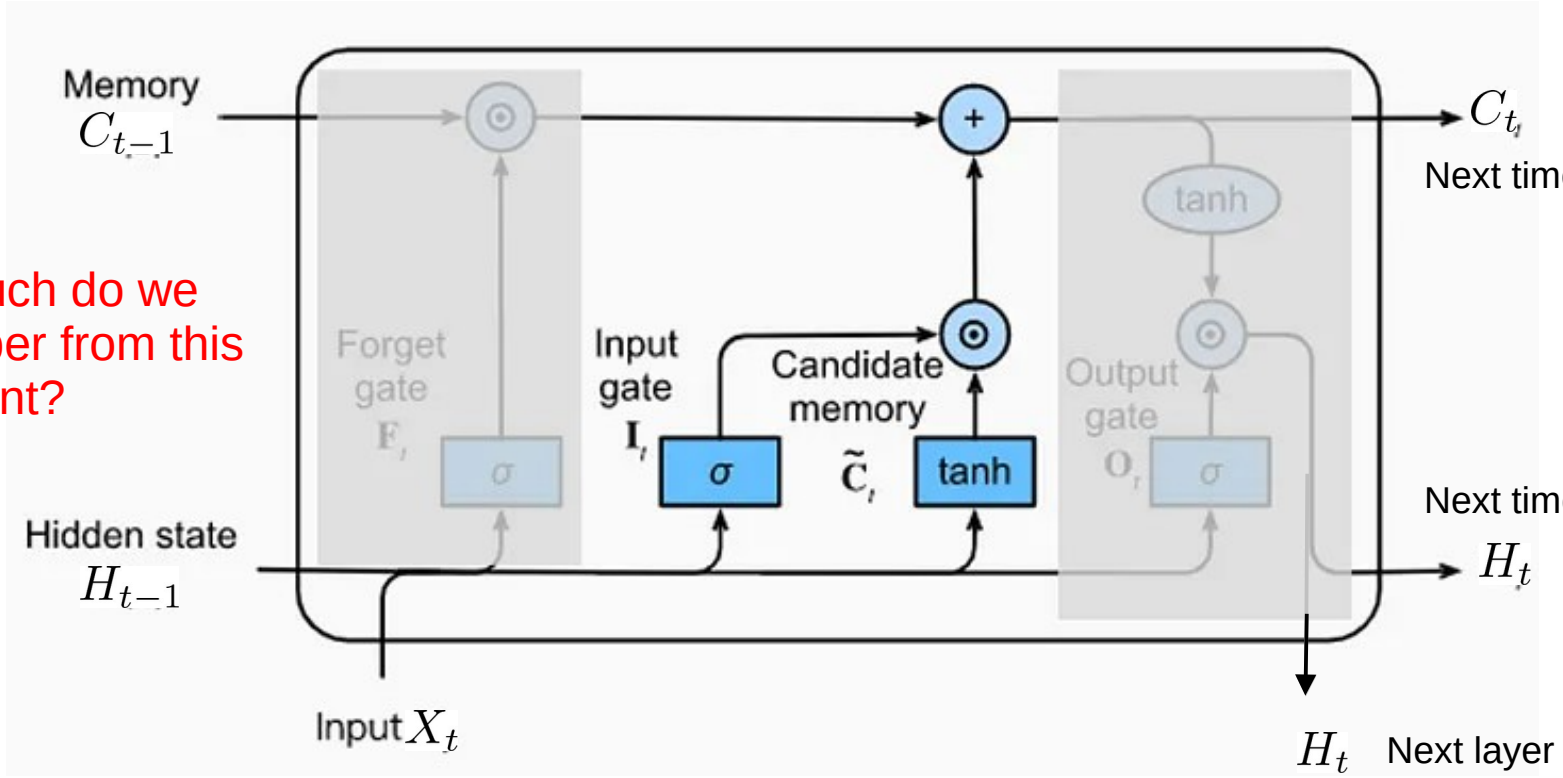
Solution: Long Short-Term Memory (LSTM)

1-How much do we forget from previous time-points?



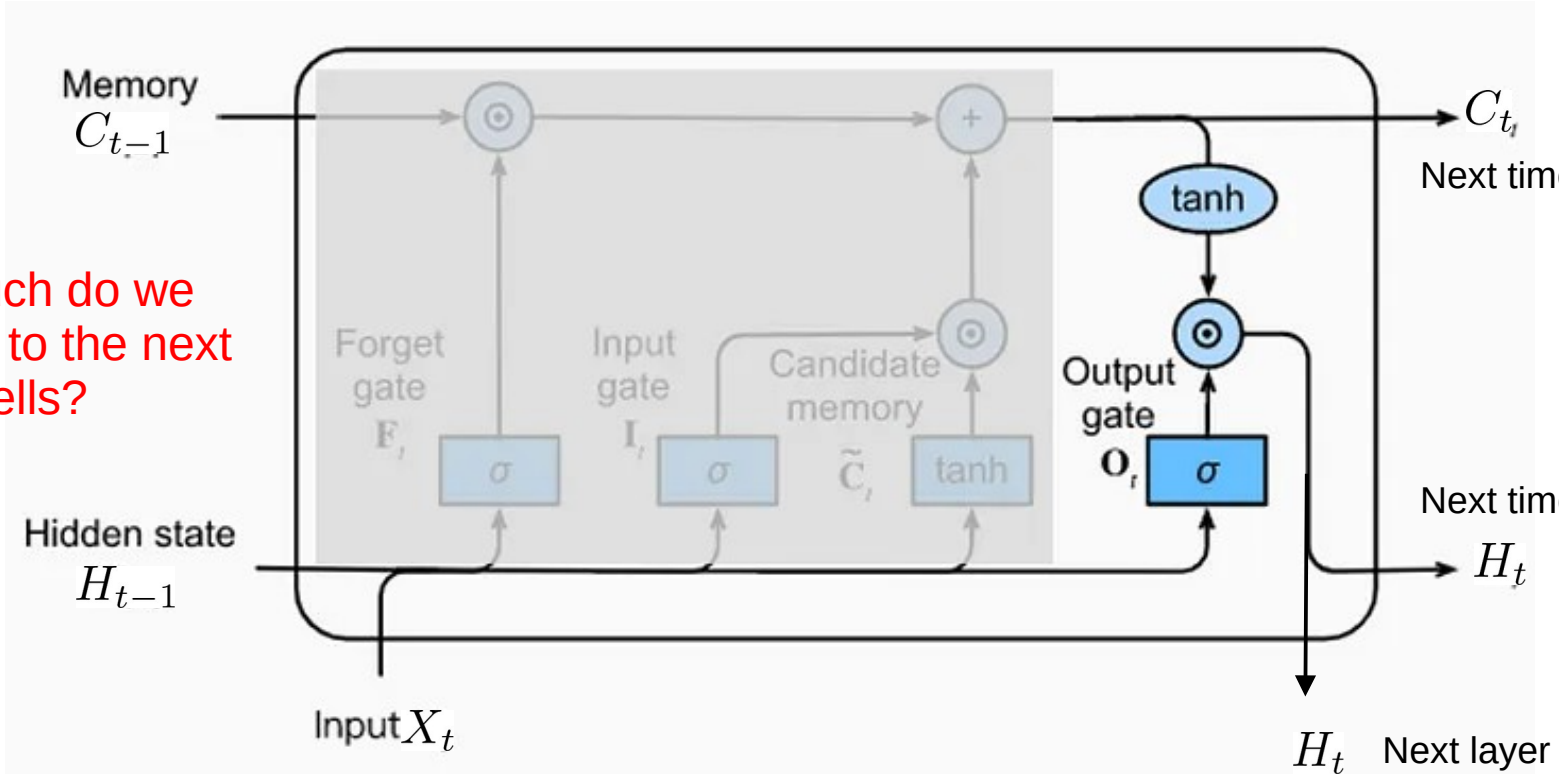
Solution: Long Short-Term Memory (LSTM)

2-How much do we remember from this time-point?



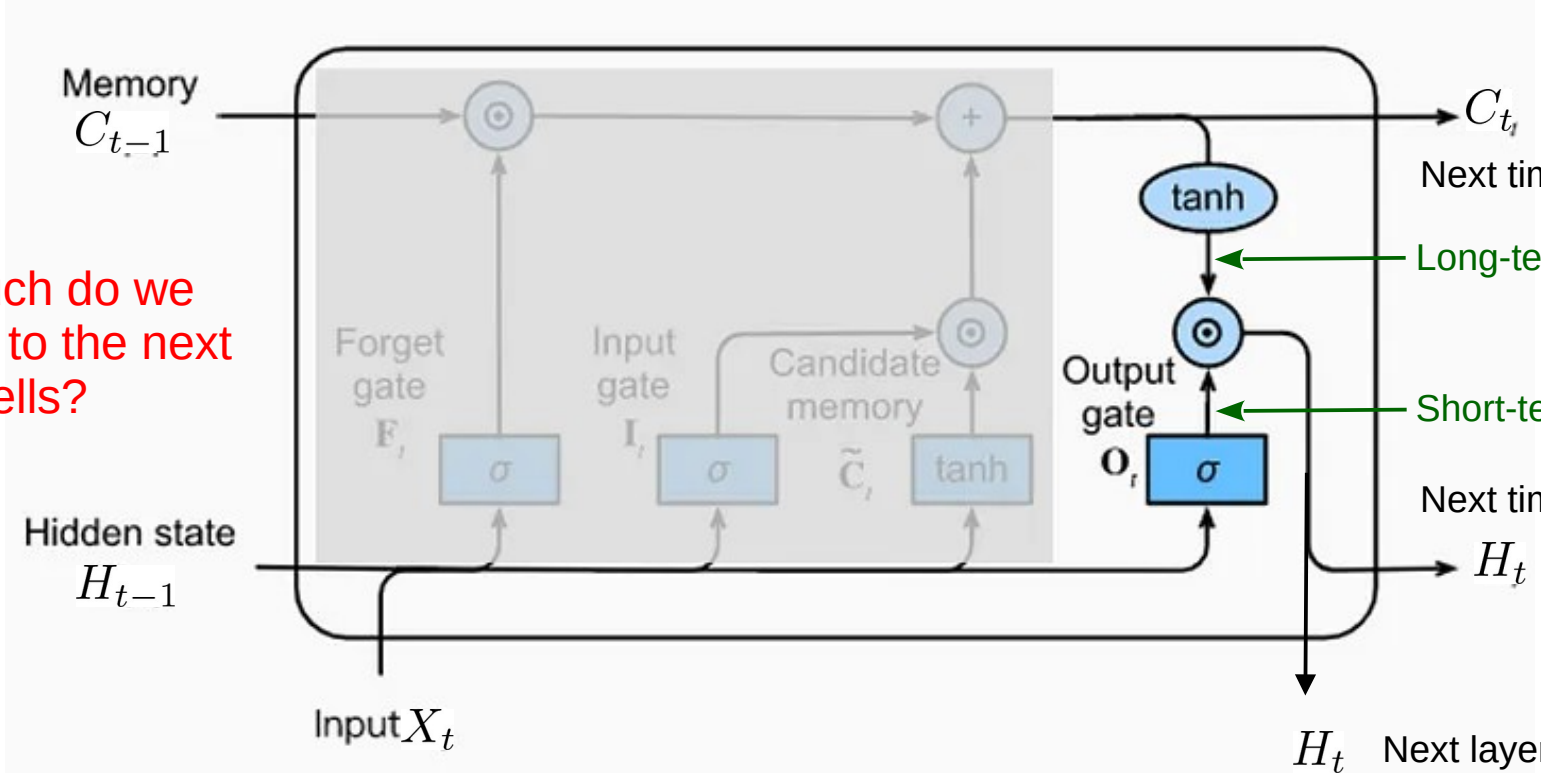
Solution: Long Short-Term Memory (LSTM)

3-How much do we transmit to the next LSTM cells?

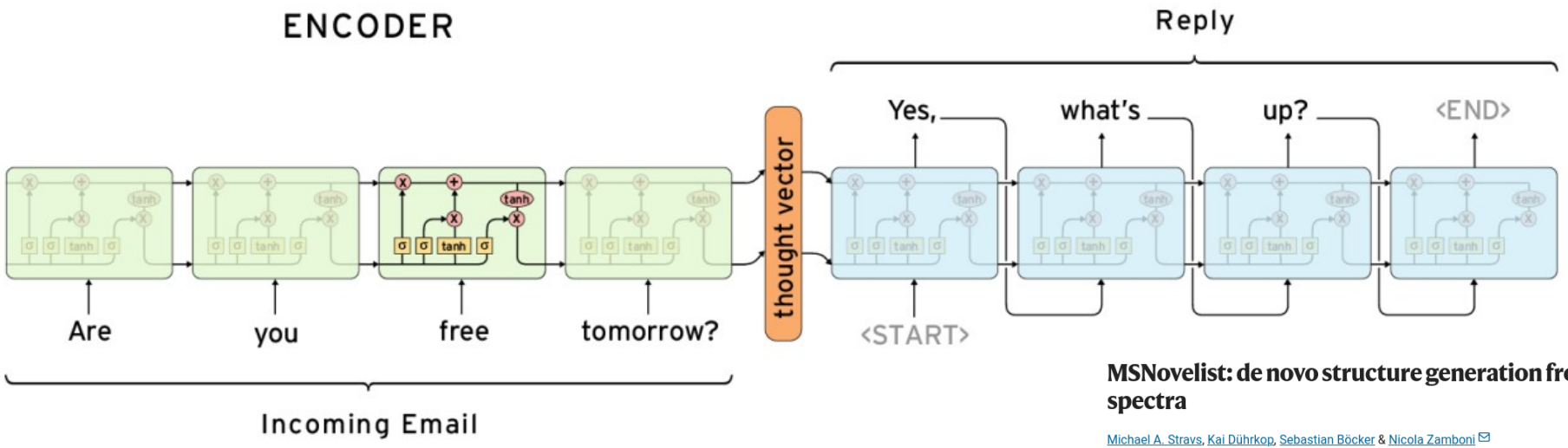


Solution: Long Short-Term Memory (LSTM)

3-How much do we transmit to the next LSTM cells?



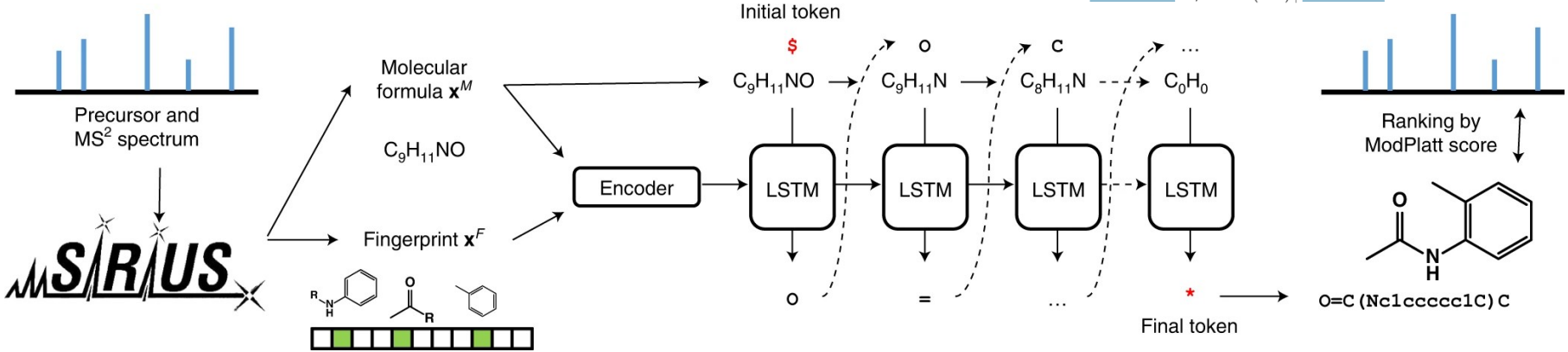
LSTMs for Encoder-Decoder



MSNovelist: de novo structure generation from mass spectra

Michael A. Stravs, Kai Dührkop, Sebastian Böcker & Nicola Zamboni

Nature Methods **19**, 865–870 (2022) | Cite this article



Example in bioinformatics

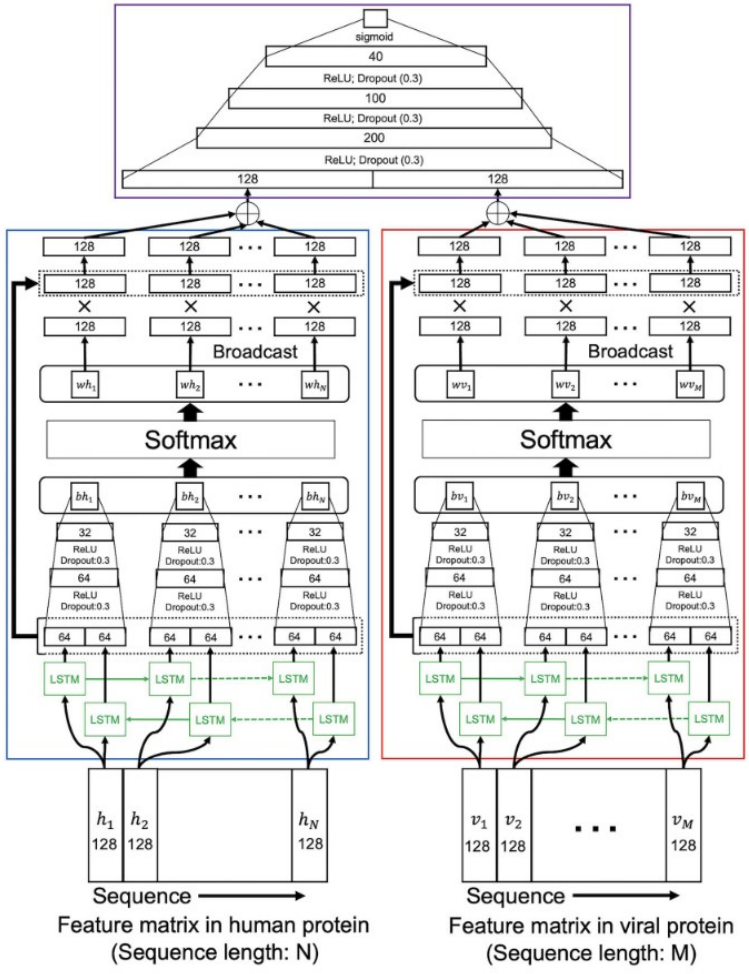
OXFORD

Briefings in Bioinformatics, 22(6), 2021, 1–9

https://doi.org/10.1093/bib/bbab228
Problem Solving Protocol

LSTM-PHV: prediction of human-virus protein–protein interactions by LSTM with word2vec

Sho Tsukiyama, Md Mehedi Hasan, Satoshi Fujii and Hiroyuki Kurata



Example in clinical setting

Contents lists available at ScienceDirect

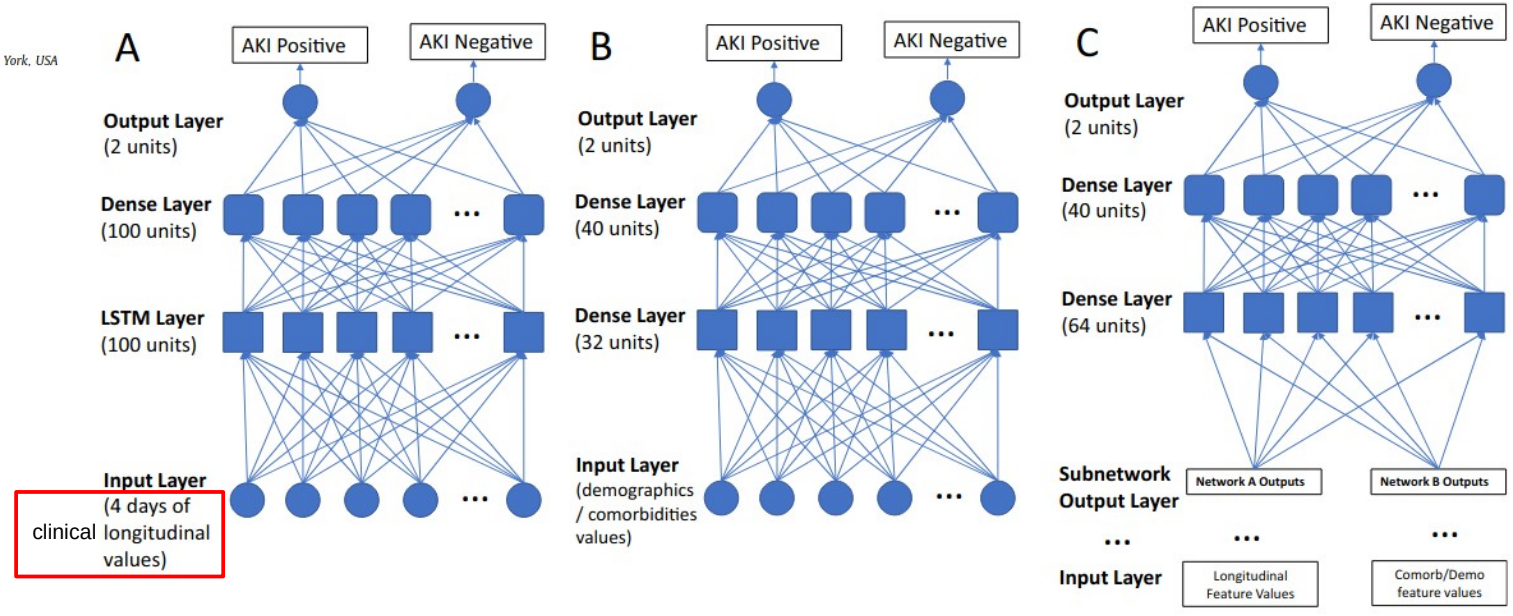
International Journal of Infectious Diseases

journal homepage: www.elsevier.com/locate/ijid

Long-short-term memory machine learning of longitudinal clinical data accurately predicts acute kidney injury onset in COVID-19: a two-center study

Justin Y. Lu, Joanna Zhu, Jocelyn Zhu, Tim Q Duong*

Department of Radiology, Montefiore Medical Center, Albert Einstein College of Medicine, New York, USA



The paper that changed everything: the Transfomer

Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*] [†]
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

Illia Polosukhin[*] [‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

[*]Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

[†]Work performed while at Google Brain.

[‡]Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

The paper that changed everything: the Transfomer

Attention Is All You Need

Cool title

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

All authors equal

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and

*Equal contribution. Listing order is random.

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

Never published in a journal

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Cited... 138640 times as of 27 October 2024!

The paper that changed everything: the Transfomer

Attention Is All You Need

Ashish Vaswani[*] Google Brain avaswani@google.com

Noam Shazeer[*] Google Brain noam@google.com

Niki Parmar[*] Google Research nikip@google.com

Jakob Uszkoreit[*] Google Research usz@google.com

Llion Jones[*] Google Research llion@google.com

Aidan N. Gomez[*] [†] University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser[*] Google Brain lukaszkaiser@google.com

Illia Polosukhin[*] [‡] illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

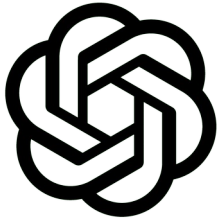
Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

[*]Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

[†]Work performed while at Google Brain.

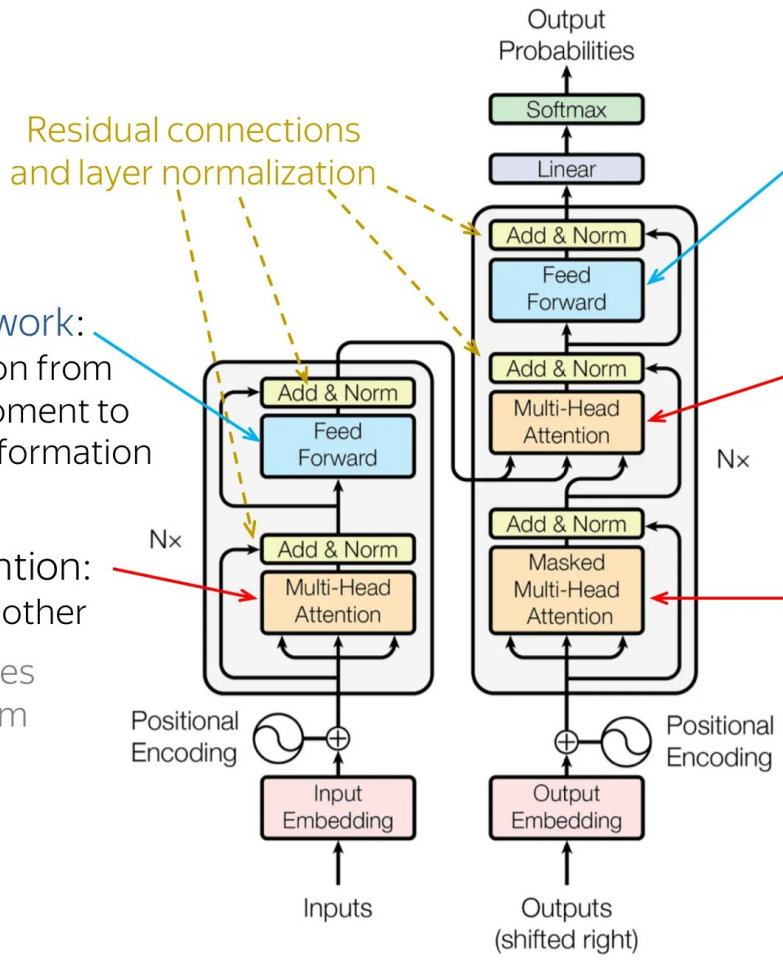
[‡]Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.



Claude

The Transformer: Memory + context = attention



Residual connections and layer normalization

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

Encoder self-attention: tokens look at each other

queries, keys, values are computed from encoder states

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

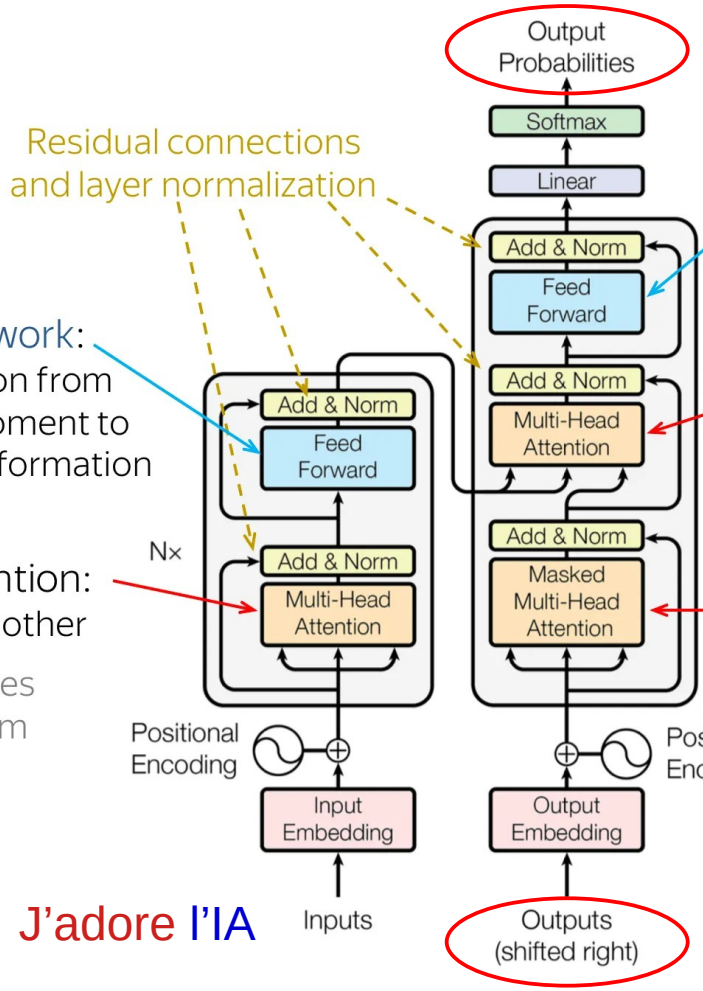
Decoder-encoder attention: target token looks at the source

queries – from decoder states; keys and values from encoder states

Decoder self-attention (masked): tokens look at the previous tokens

queries, keys, values are computed from decoder states

The Transformer: Memory + context = attention



I love AI

J'adore l'IA

<start> I love

Residual connections and layer normalization

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

Encoder self-attention: tokens look at each other

queries, keys, values are computed from encoder states

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

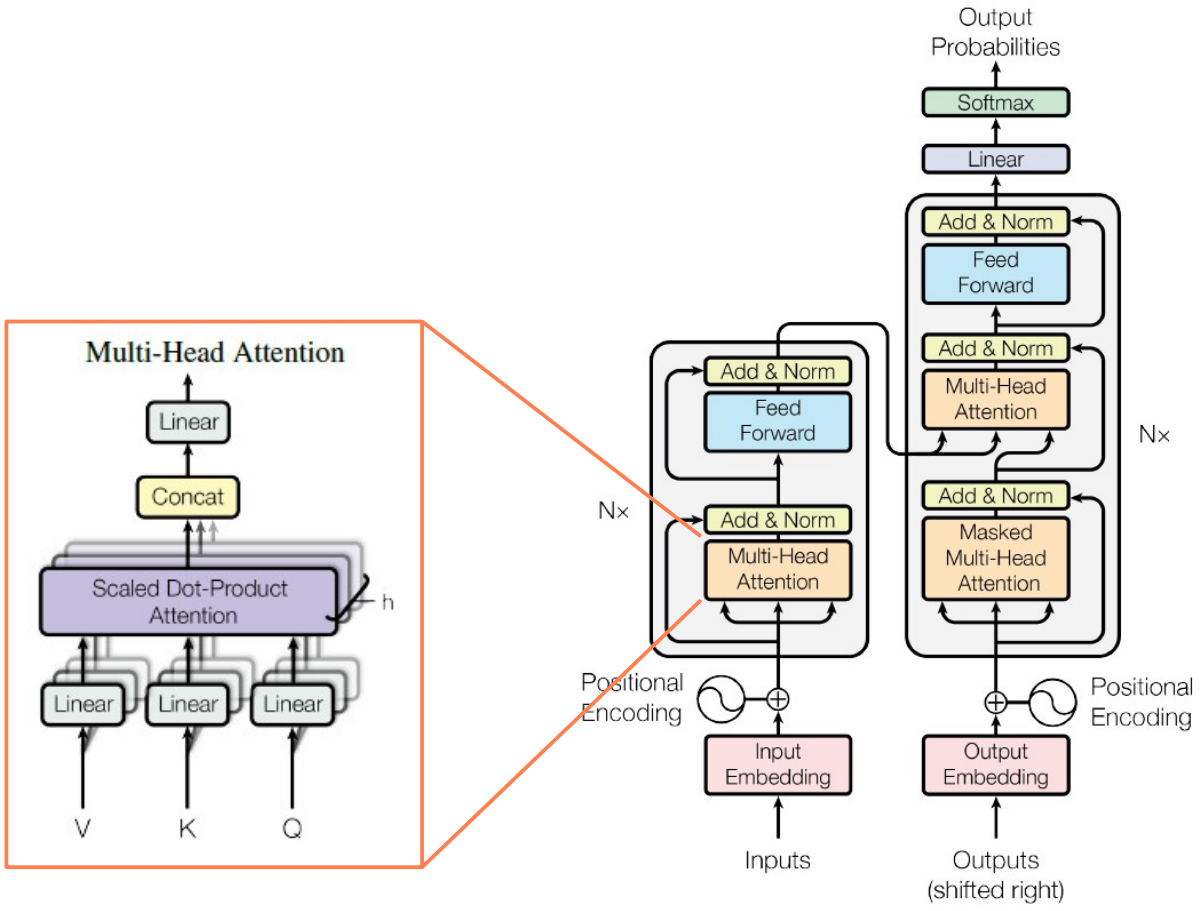
Decoder-encoder attention: target token looks at the source

queries – from decoder states; keys and values from encoder states

Decoder self-attention (masked): tokens look at the previous tokens

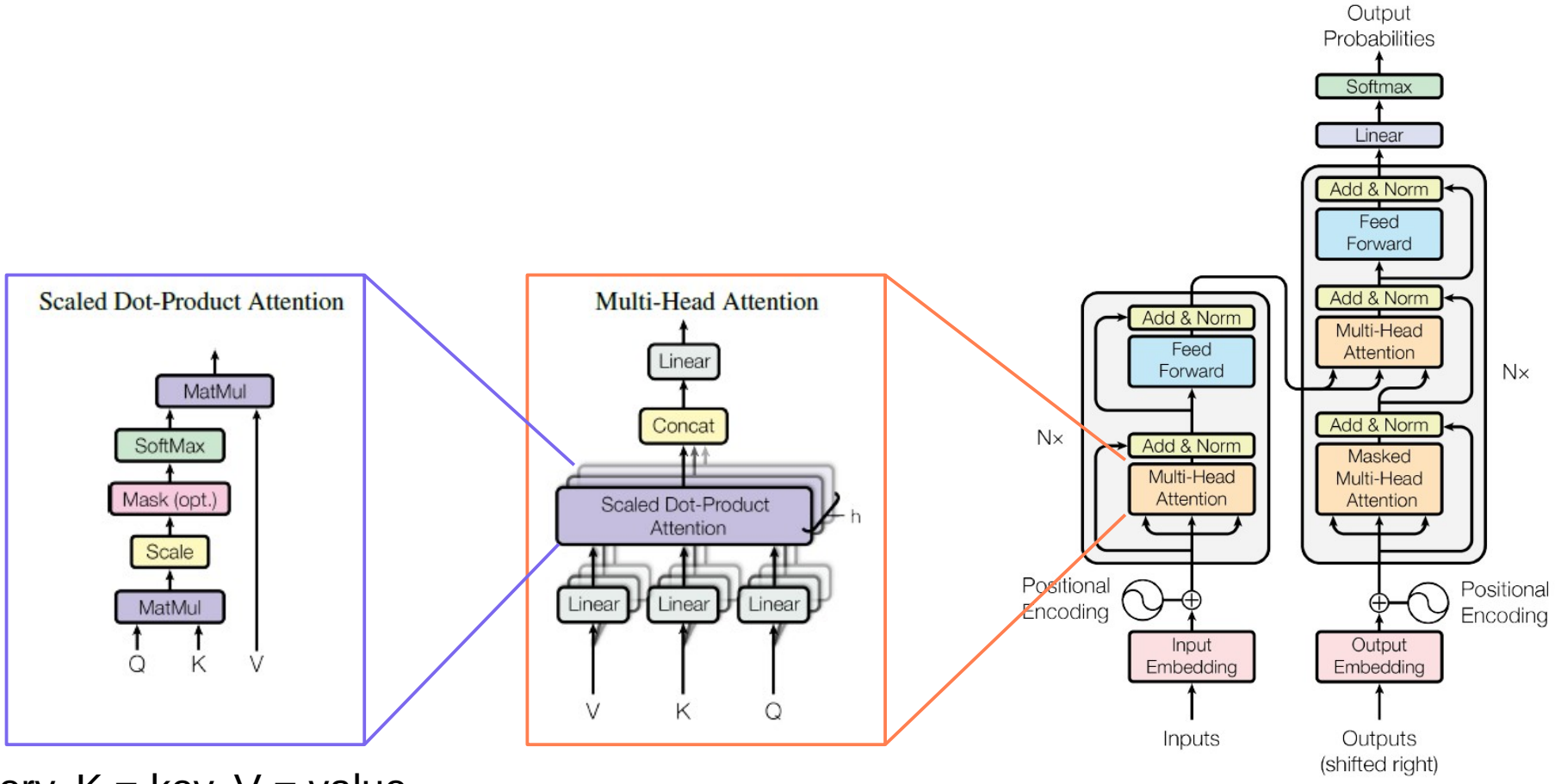
queries, keys, values are computed from decoder states

Attention in the Transformer



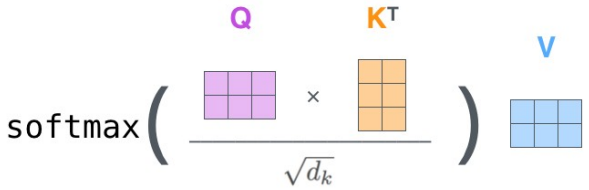
Q = query, K = key, V = value

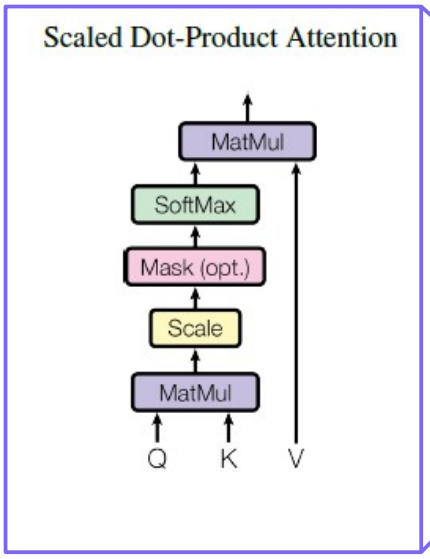
Attention in the Transformer

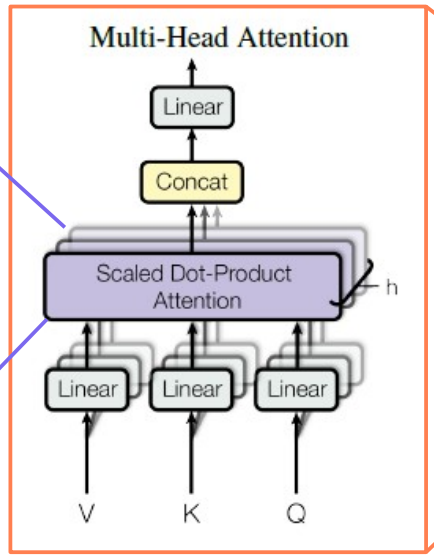


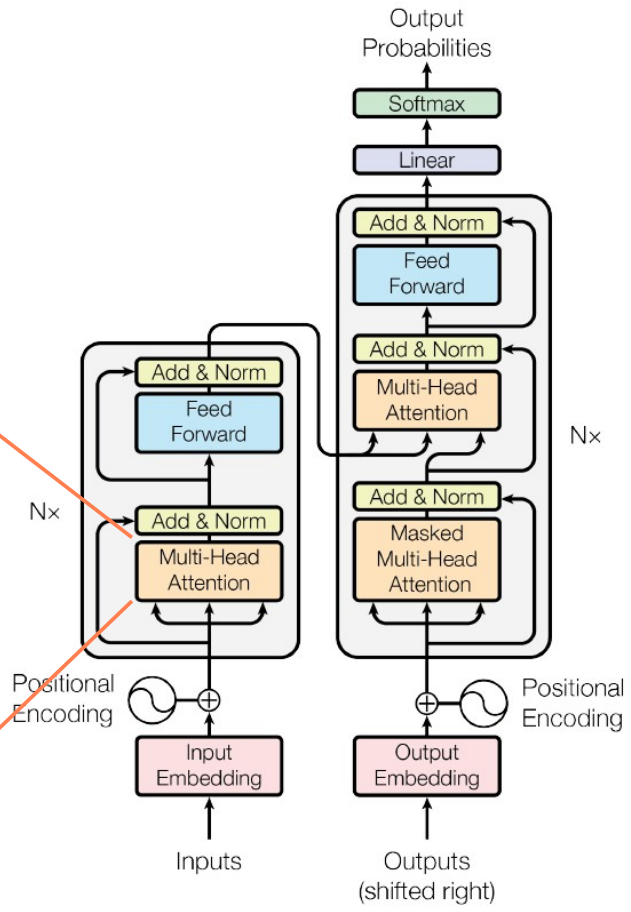
Q = query, K = key, V = value

Attention in the Transformer







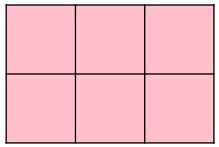


Q = query, K = key, V = value

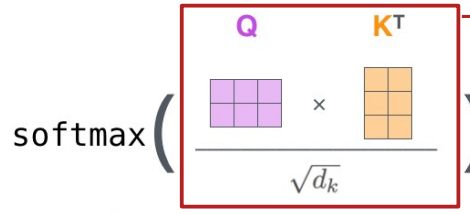
Attention in the Transformer

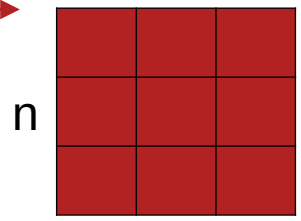
n = #tokens

d = dimension of embeddings

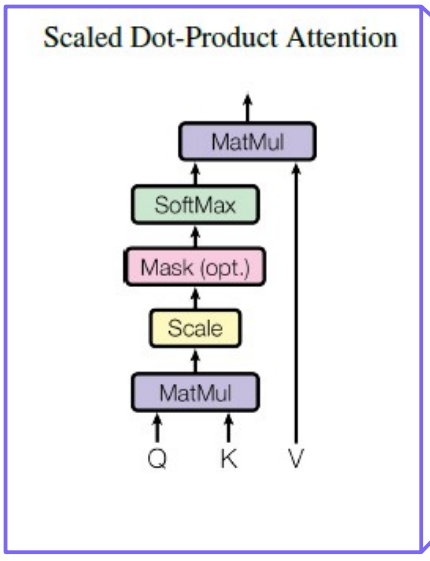


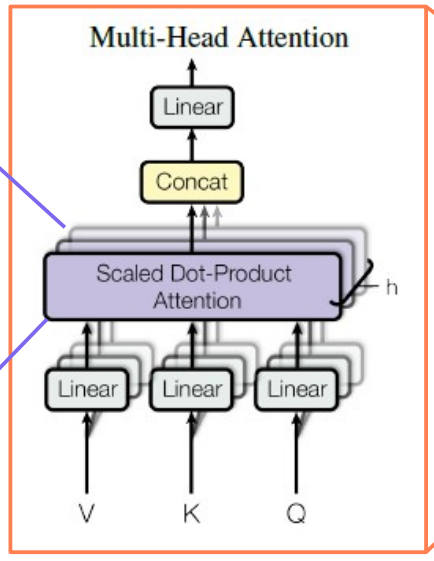
Attention: aggregated attention per embedding





Attention matrix:
Impact of each embedding of each token on all the others

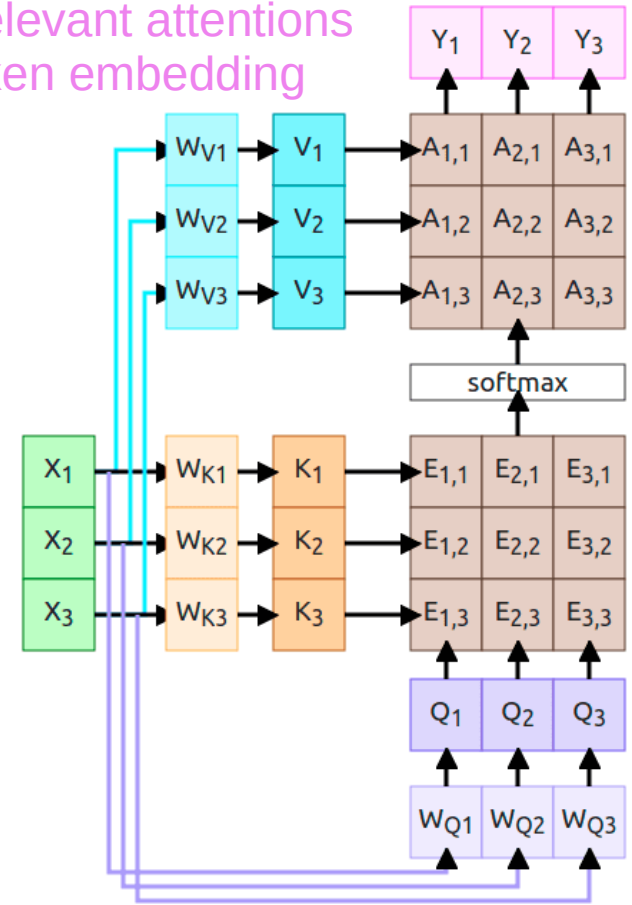




Q = query, K = key, V = value

Attention in the Transformer

Actual relevant attentions
1 per token embedding



Self-Attention Layer

Input vectors: X (shape $N_X \times D_X$)
Key matrix: W_K (Shape $D_X \times D_Q$)
Value matrix: W_V (Shape $D_X \times D_V$)
Query matrix: W_Q (Shape $D_X \times D_Q$)

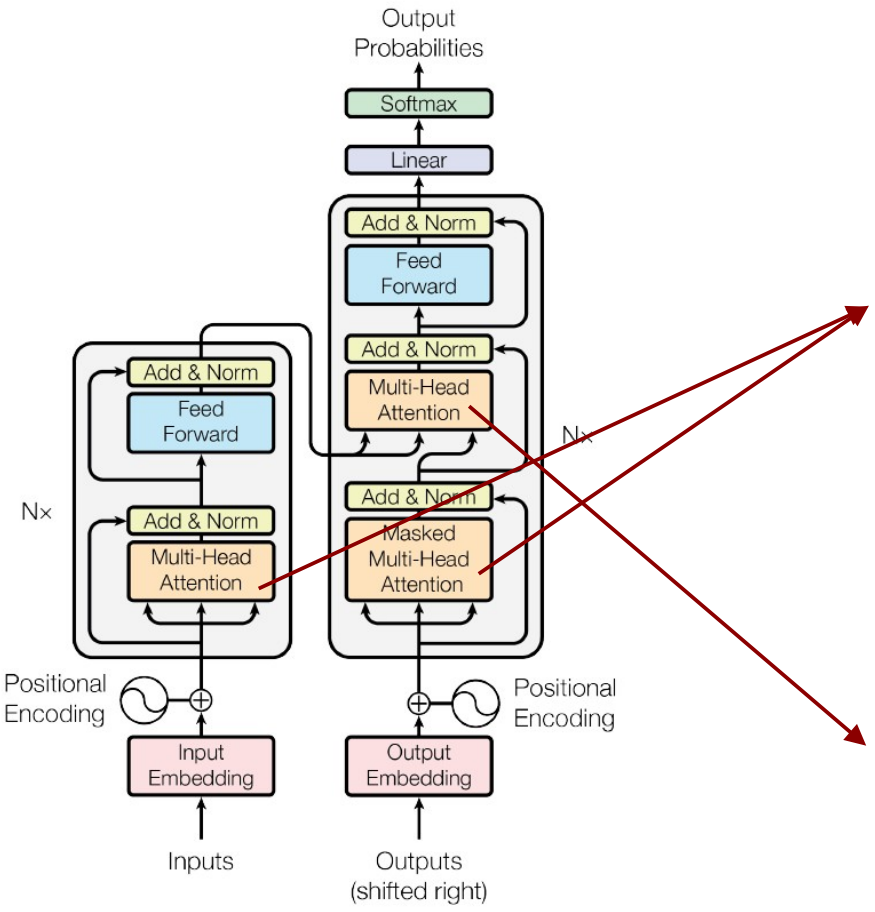
Query vectors: $Q = XW_Q$ (Shape $N_X \times D_Q$)
Key vectors: $K = XW_K$ (Shape $N_X \times D_Q$)
Value vectors: $V = XW_V$ (Shape $N_X \times D_V$)
Similarity function: *scaled dot product*
Similarities: $E = QK^T$ (shape $N_X \times N_X$), $E_{i,j} = Q_i \cdot K_j / \sqrt{D_Q}$
Attention weights: $A = \text{softmax}(E, dim = 1)$ (shape $N_X \times N_X$)
Output: $Y = AV$ (shape $N_X \times D_V$) where $Y_i = \sum_j (A_{i,j}, V_j)$

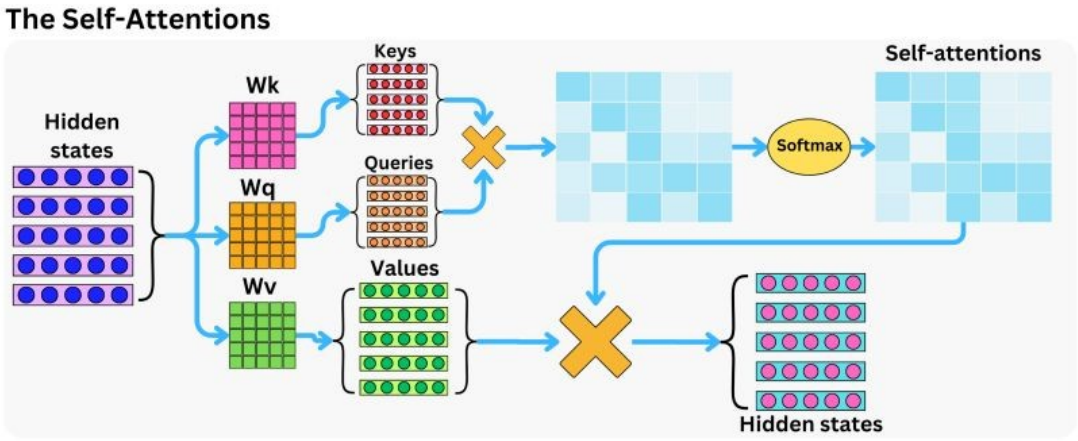
The attention of all token embeddings on all token embeddings

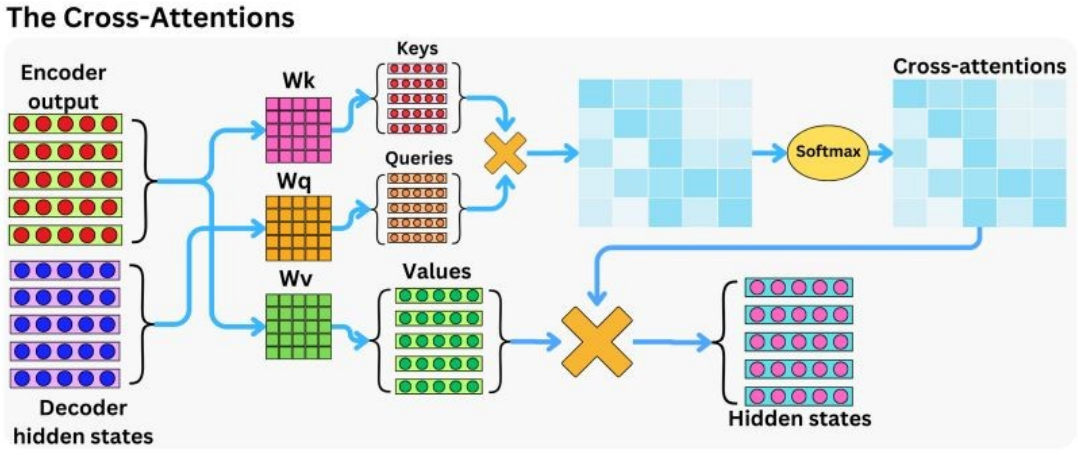
X is entered
W_Q, W_K, and W_V are learned
Everything else is computed

Source: https://erdem.pl/2021/05/introduction-to-attention-mechanism

Self versus cross-attention

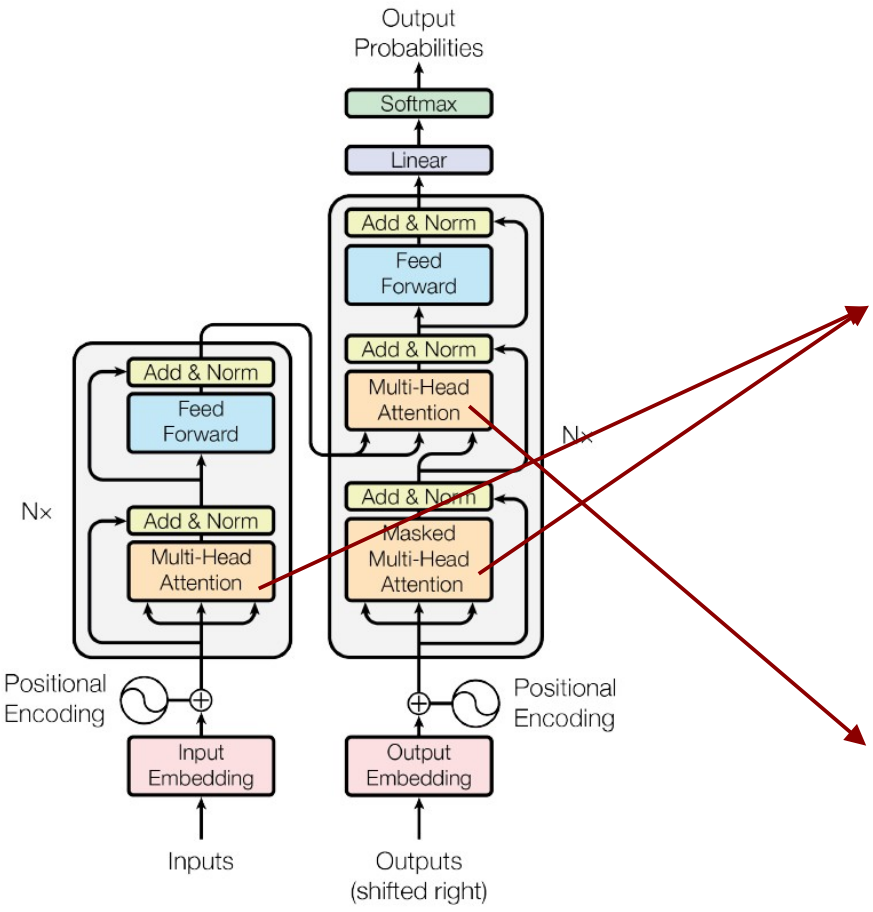


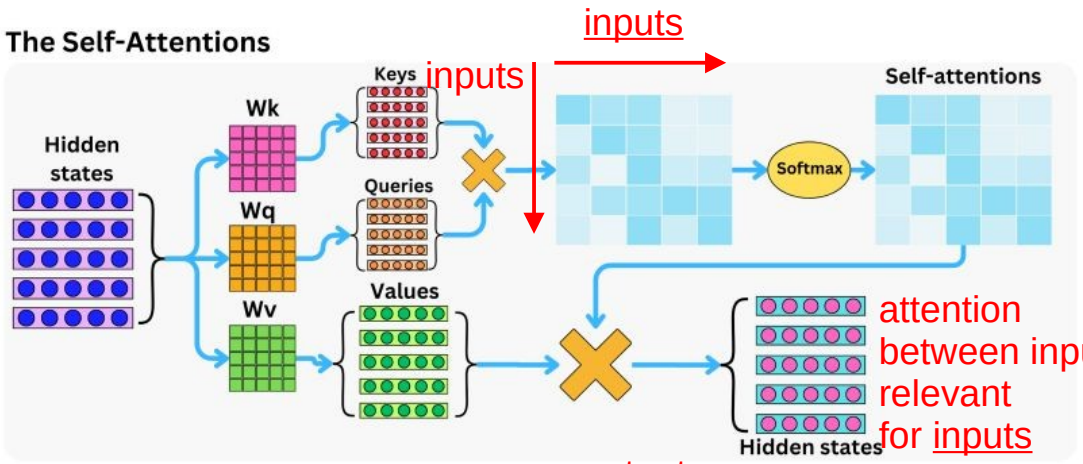


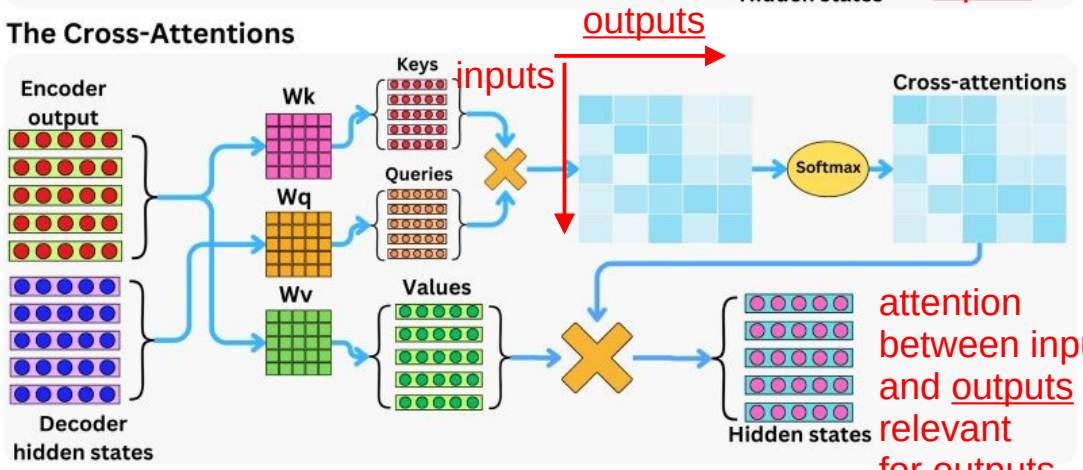


Source: https://theaiedge.io

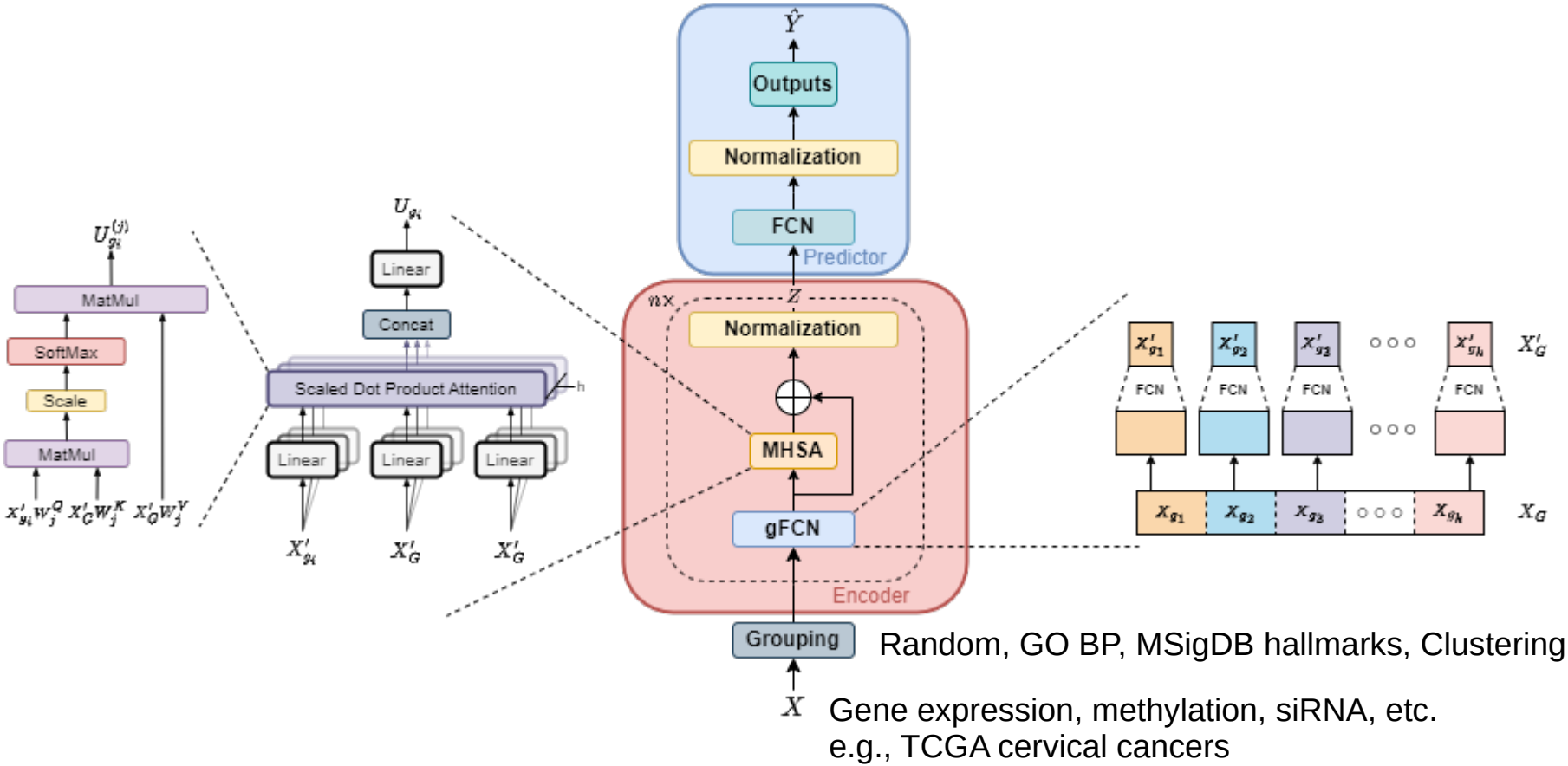
Self versus cross-attention







AttOmics

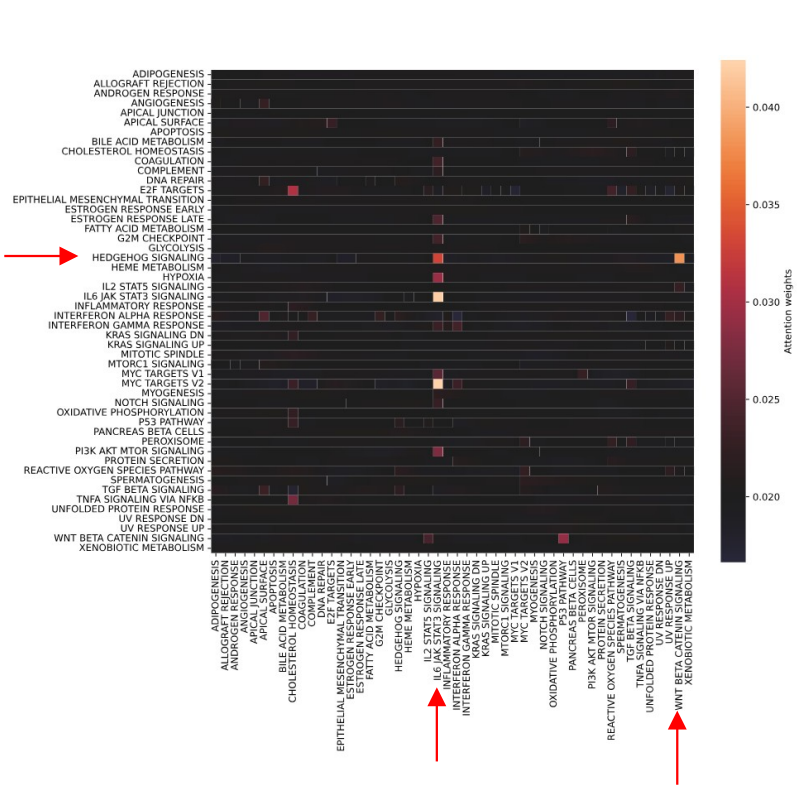


Random, GO BP, MSigDB hallmarks, Clustering

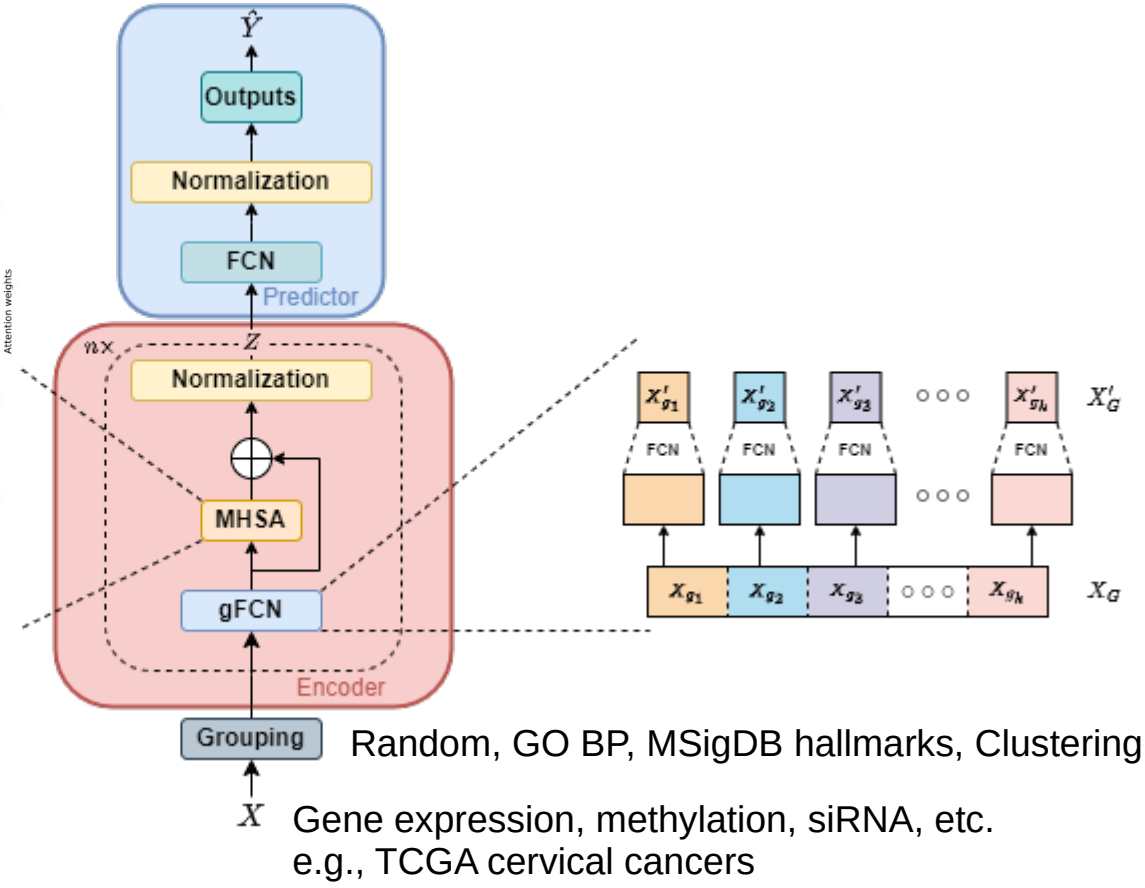
Gene expression, methylation, siRNA, etc.
e.g., TCGA cervical cancers

Beaude, A., Rafiee Vahid, M., Augé, F., Zehraoui, F., & Hanczar, B. (2023). AttOmics: attention-based architecture for diagnosis and prognosis from omics data. *Bioinformatics*, 39(Supplement_1), i94-i102.

AttOmics



Pathways affected in cervical cancer



Random, GO BP, MSigDB hallmarks, Clustering

Gene expression, methylation, siRNA, etc.
e.g., TCGA cervical cancers

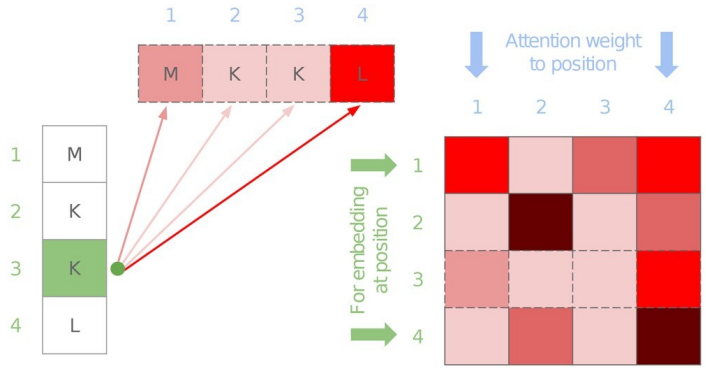
Beaude, A., Rafiee Vahid, M., Augé, F., Zehraoui, F., & Hanczar, B. (2023). AttOmics: attention-based architecture for diagnosis and prognosis from omics data. *Bioinformatics*, 39(Supplement_1), i94-i102.

EnzBERT

Predicting enzymatic function of protein sequences with attention

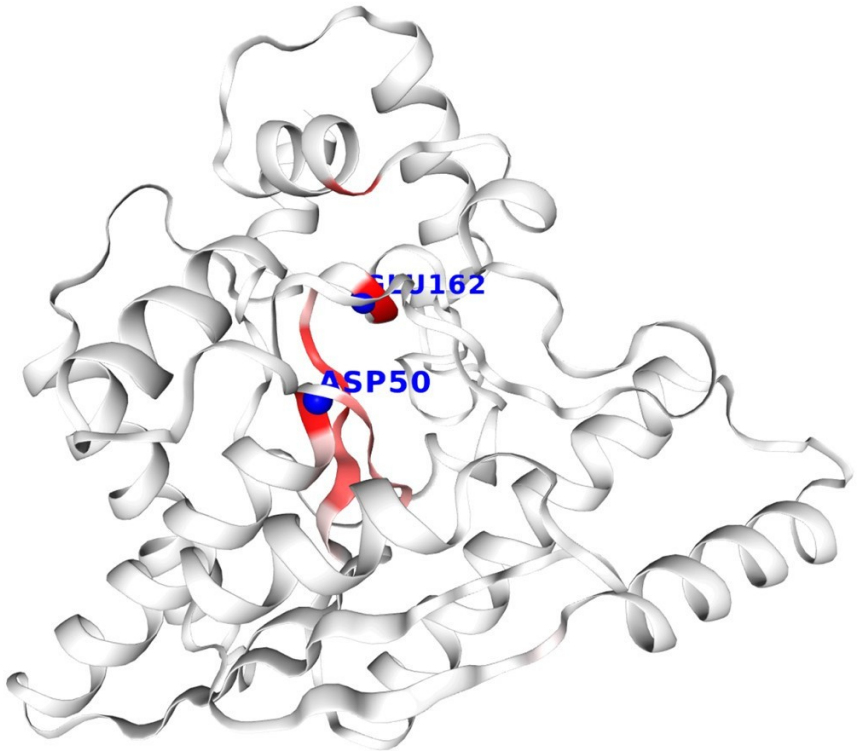
Nicolas Buton ✉, François Coste, Yann Le Cunff

Bioinformatics, Volume 39, Issue 10, October 2023, btad620, https://doi.org/10.1093/bioinformatics/btad620

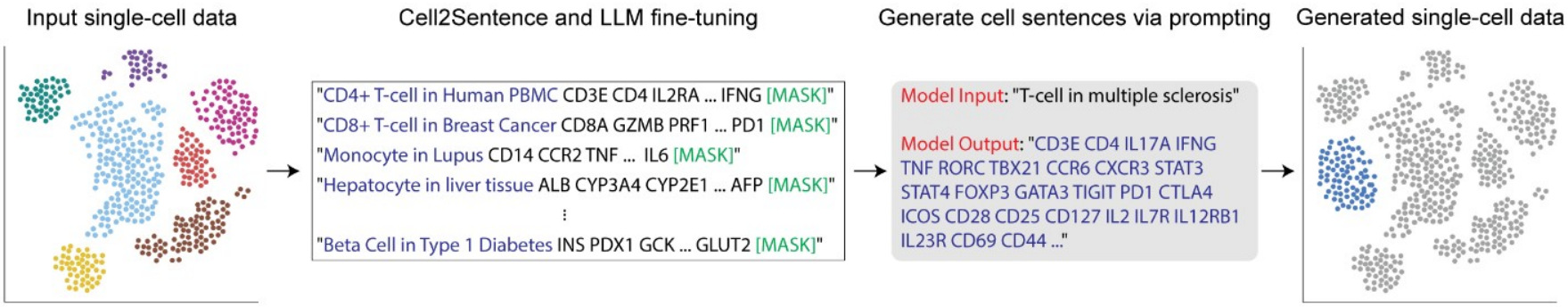


Aggregated attention
for each token (amino acid)

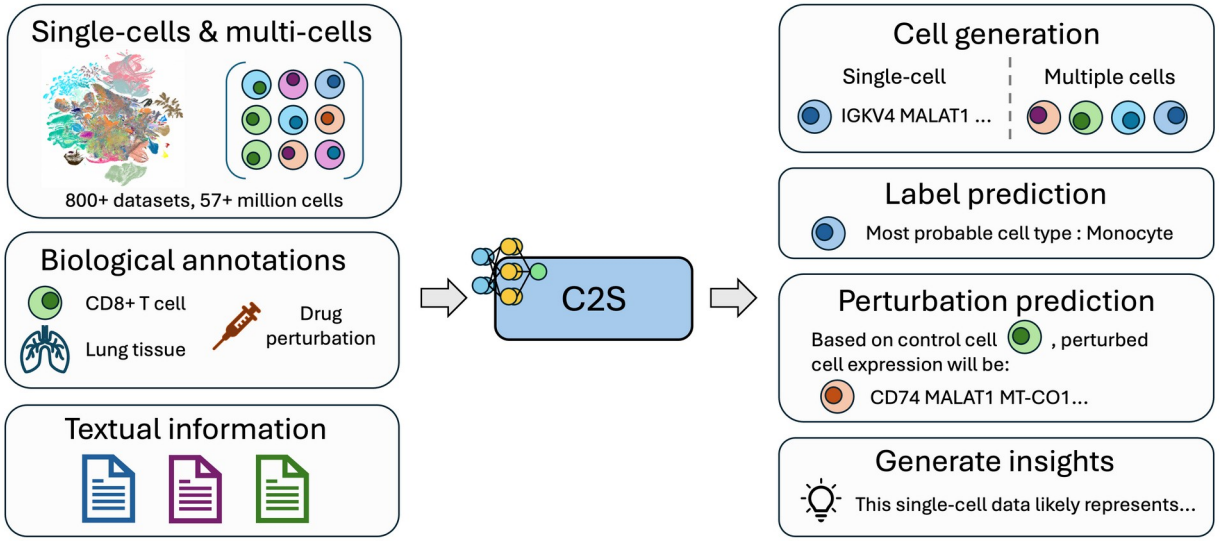
Nh(3)-dependent nad(+) synthetase



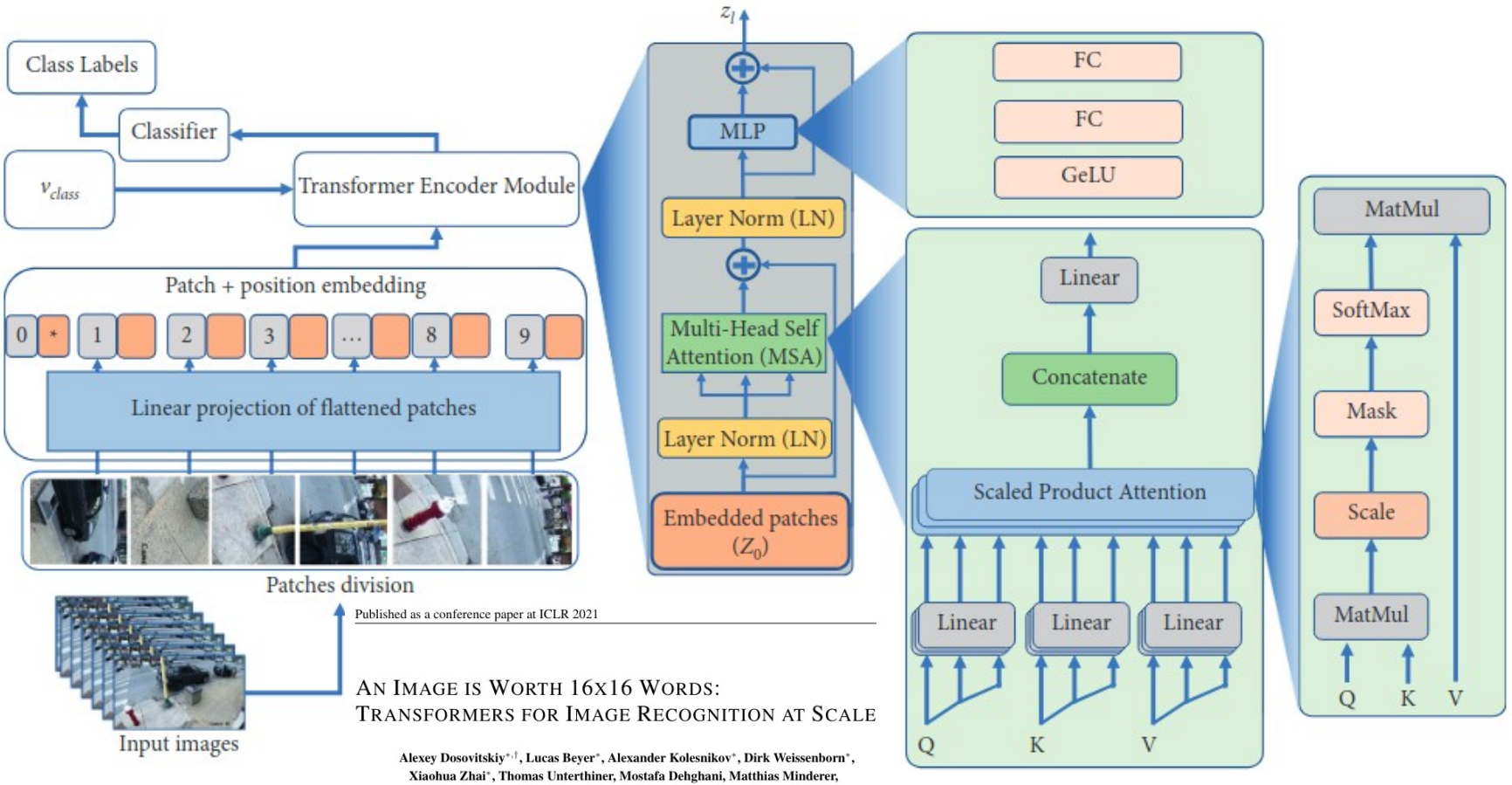
Cell2Sentence



Levine *et al* (2024). Cell2Sentence: Teaching Large Language Models the Language of Biology. *BioRxiv* https://doi.org/10.1101/2023.09.11.557287



Vision Transformer



Published as a conference paper at ICLR 2021

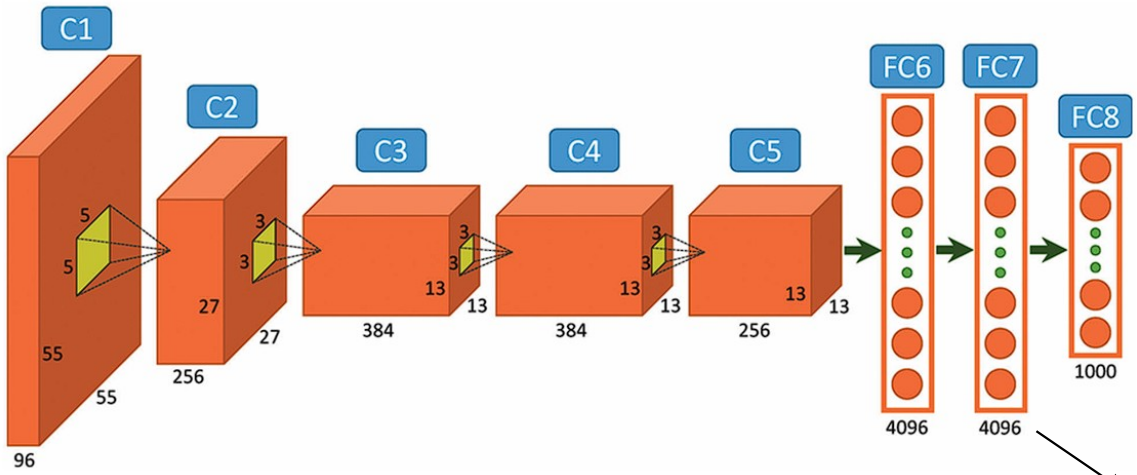
AN IMAGE IS WORTH 16X16 WORDS:
TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy[*,†], **Lucas Beyer**[*], **Alexander Kolesnikov**[*], **Dirk Weissenborn**[*],
Xiaohua Zhai[*], **Thomas Unterthiner**, **Mostafa Dehghani**, **Matthias Minderer**,
Georg Heigold, **Sylvain Gelly**, **Jakob Uszkoreit**, **Neil Houlsby**[*,†]
[*]equal technical contribution, [†]equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com

CNN = local features
ViT = relations between distant features

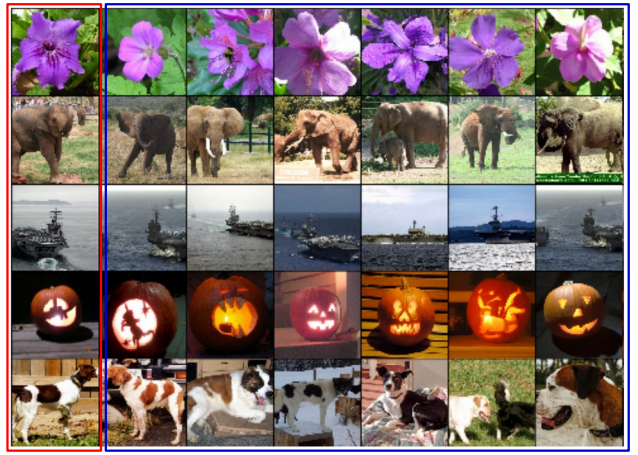
source: https://doi.org/10.1155/2022/3454167

Patches are embedded by CNNs



6 nearest neighbours in the 4096 dimension space

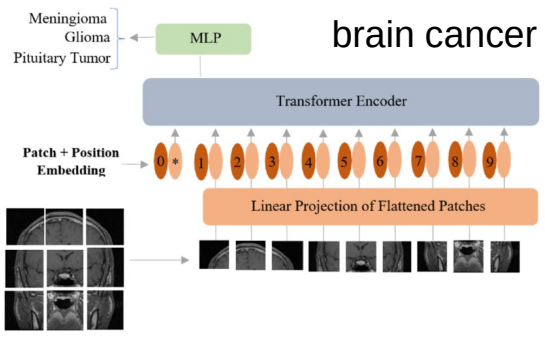
input image



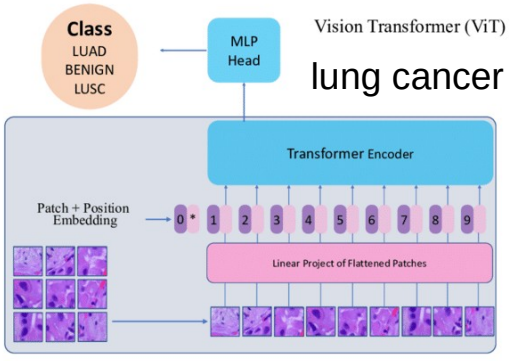
Krizhevsky A, Sutskever I, Hinton GE (2012)
ImageNet Classification with Deep
Convolutional Neural Networks
https://proceedings.neurips.cc/paper/2012/file/
c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

(presenting AlexNet, the first Deep Convolutional Network)

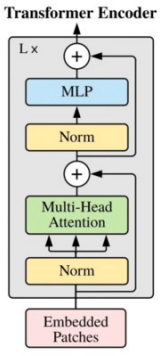
ViTs are replacing vanilla CNNs

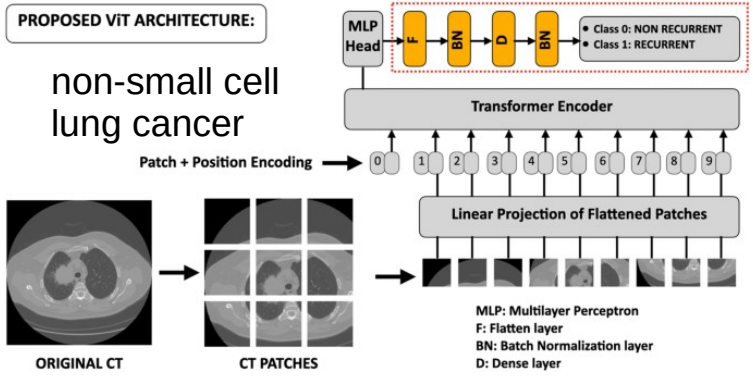


brain cancer

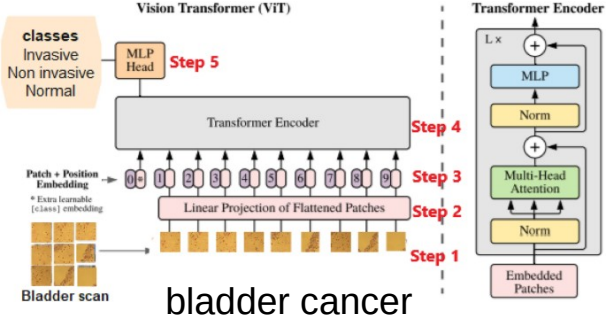


lung cancer

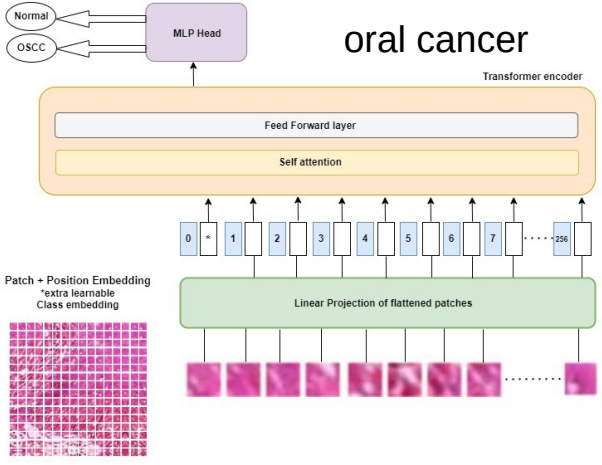




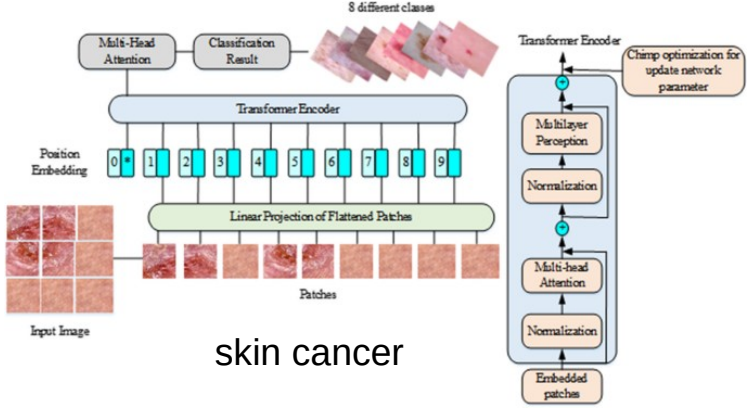
non-small cell lung cancer



bladder cancer



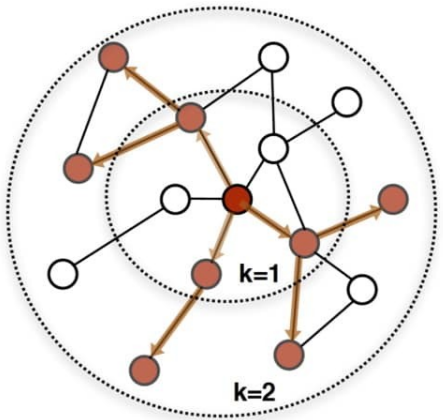
oral cancer



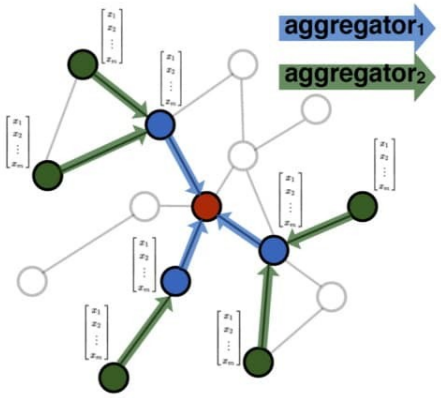
skin cancer

Graph Neural Networks (GNNs)

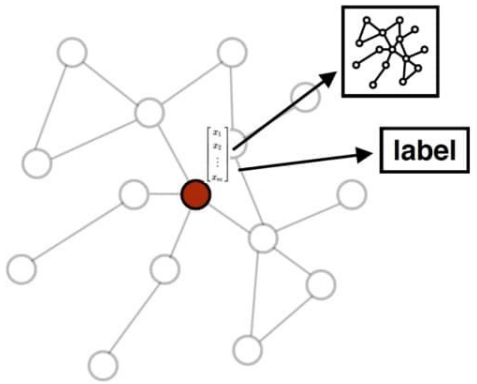
source: https://blogs.nvidia.com/blog/what-are-graph-neural-networks/



1. Sample neighborhood



2. Aggregate feature information from neighbors



3. Predict graph context and label using aggregated information

IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 20, NO. 1, JANUARY 2009 61

The Graph Neural Network Model

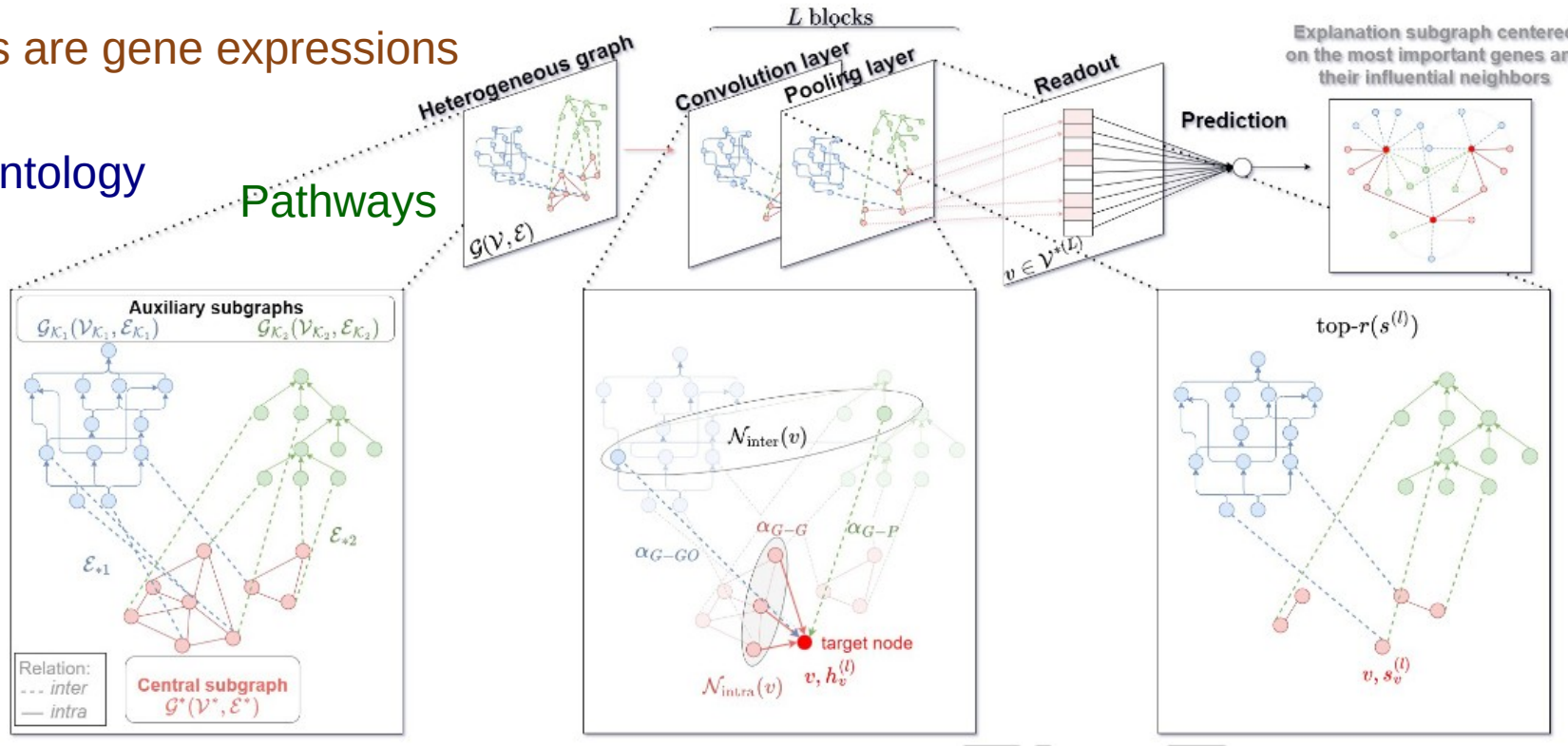
Franco Scarselli, Marco Gori, *Fellow, IEEE*, Ah Chung Tsoi, Markus Hagenbuchner, *Member, IEEE*, and Gabriele Monfardini

GNN can be heterogeneous

Inputs are gene expressions

Gene ontology

Pathways



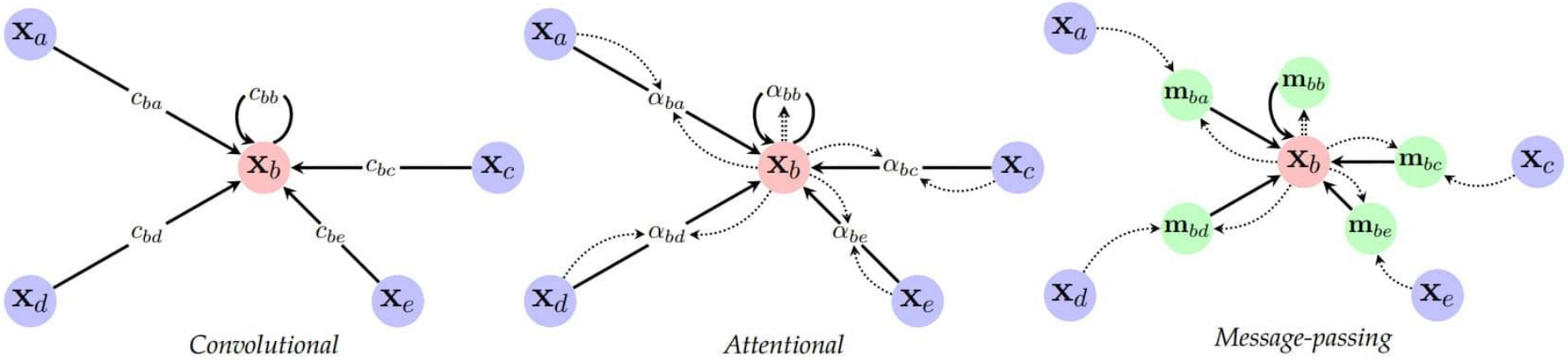
BioHAN: a Knowledge-based Heterogeneous Graph Neural Network for precision medicine on transcriptomic data

Victoria Bourgeais, Farida Zehraoui, Blaise Hanczar

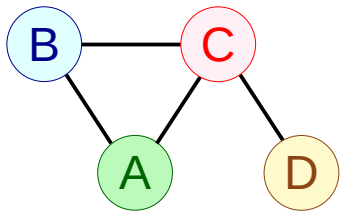
https://hal.science/hal-04092210/

Many different ways to update GNNs

source: https://blogs.nvidia.com/blog/what-are-graph-neural-networks/



Example of GNN convolution



$$D = \begin{array}{c} \begin{matrix} A & B & C & D \end{matrix} \\ \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{matrix} A \\ B \\ C \\ D \end{matrix}$$

Degree matrix

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Adjacency matrix

$$L = D - A$$

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Laplacian matrix

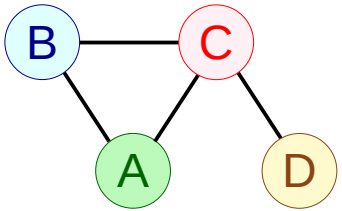
NB: undirected graph → all matrices are symmetric
This could be different for a directed graph

Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering

Michaël Defferrard **Xavier Bresson** **Pierre Vandergheynst**
EPFL, Lausanne, Switzerland
{michael.defferrard,xavier.bresson,pierre.vandergheynst}@epfl.ch

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Example of GNN convolution



$$D = \begin{array}{c} \begin{matrix} A & B & C & D \end{matrix} \\ \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{matrix} A \\ B \\ C \\ D \end{matrix}$$

Degree matrix

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Adjacency matrix

$$L = D - A$$

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Laplacian matrix

identity matrix
no influence from neighbours

influence from
immediate neighbours

influence from neighbours and
neighbours of neighbours

$$p_w(L) = w_0 I + w_1 L + w_2 L^2 + ... + w_k L^k$$

$$x' = p_w(L)x$$

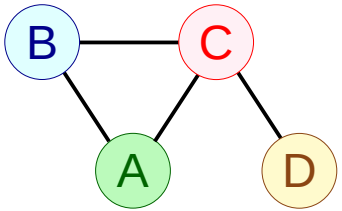
$$w = [w_0, w_1, w_2, ..., w_k]$$ kernel de convolution

Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering

Michaël Defferrard Xavier Bresson Pierre Vandergheynst
EPFL, Lausanne, Switzerland
{michael.defferrard,xavier.bresson,pierre.vandergheynst}@epfl.ch

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Example of GNN convolution



D only influences C

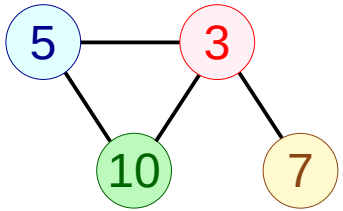
$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

D influences A, B, C

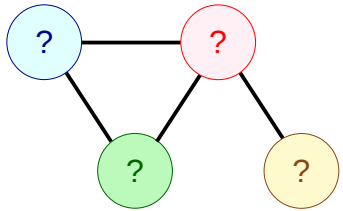
$$L^2 = \begin{bmatrix} 6 & -3 & -4 & 1 \\ -3 & 6 & -4 & 1 \\ -4 & -4 & 12 & -4 \\ 1 & 1 & -4 & 2 \end{bmatrix}$$

$w = [1, 0.1, 0.01]$

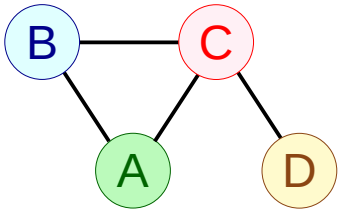
layer 1



layer 2



Example of GNN convolution



D only influences C

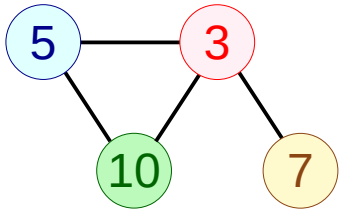
D influences A, B, C

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \qquad L^2 = \begin{bmatrix} 6 & -3 & -4 & 1 \\ -3 & 6 & -4 & 1 \\ -4 & -4 & 12 & -4 \\ 1 & 1 & -4 & 2 \end{bmatrix}$$

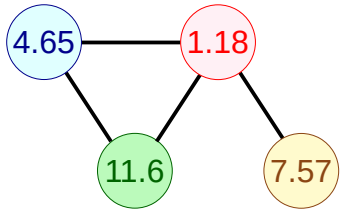
$$w = [1, 0.1, 0.01]$$

$$1 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 5 \\ 3 \\ 7 \end{bmatrix} + 0.1 \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 5 \\ 3 \\ 7 \end{bmatrix} + 0.01 \begin{bmatrix} 6 & -3 & -4 & 1 \\ -3 & 6 & -4 & 1 \\ -4 & -4 & 12 & -4 \\ 1 & 1 & -4 & 2 \end{bmatrix} \begin{bmatrix} 10 \\ 5 \\ 3 \\ 7 \end{bmatrix}$$

layer 1

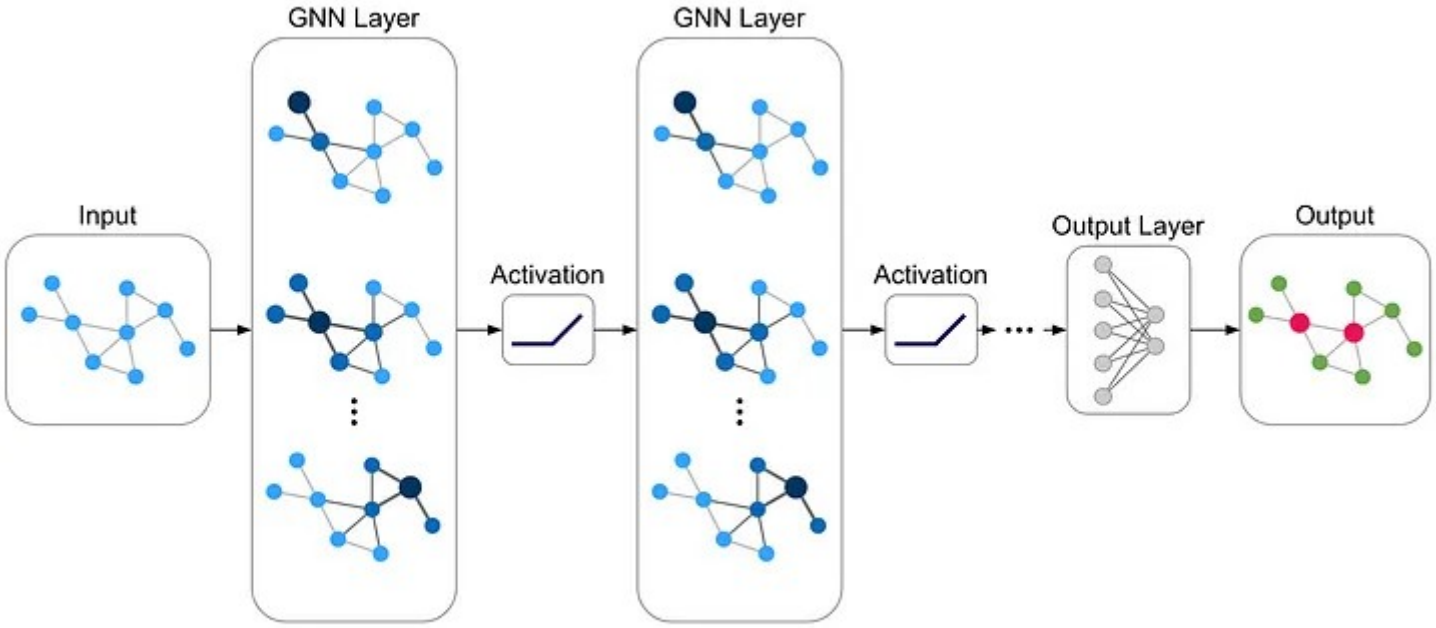


layer 2



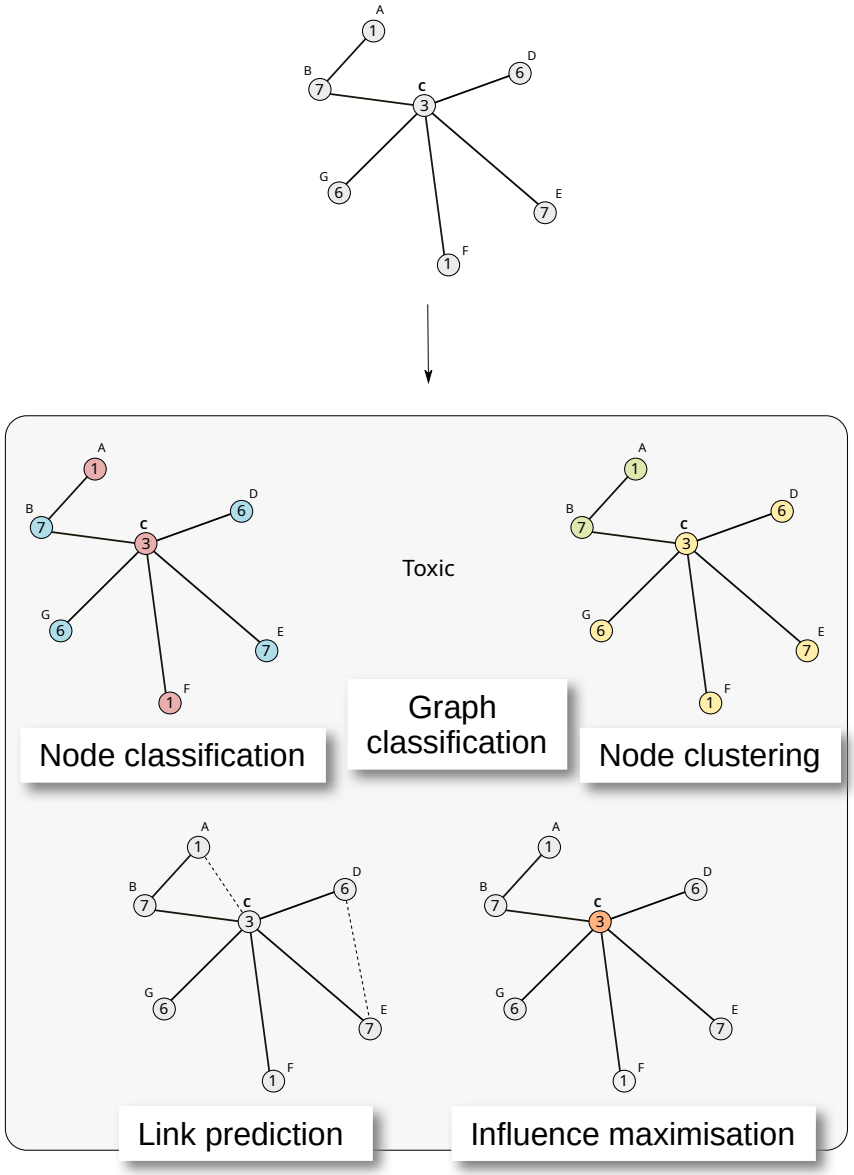
$$\sum \text{layer1} = \sum \text{layer2}$$

Graph Neural Networks (GNNs)



NB: GNNs generally comprise 3 embeddings that are updated at each iteration,
i.e nodes (vertices), edges, and graph

GNNs: What can we do?

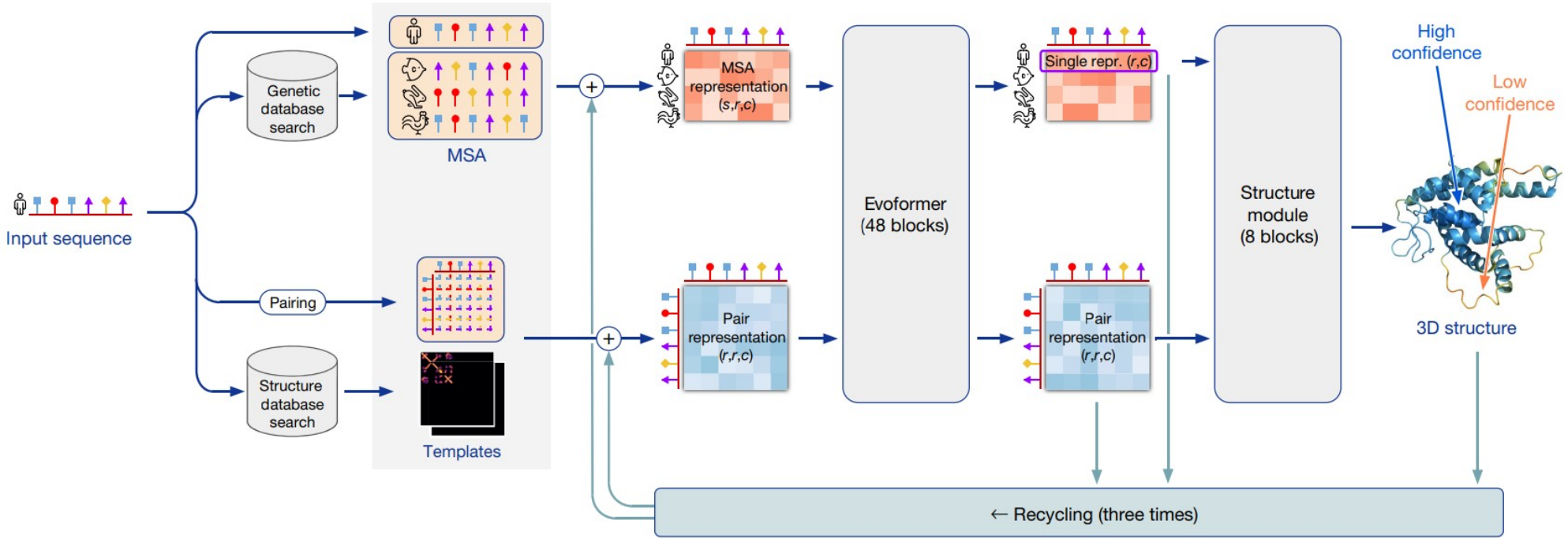


Source: Understanding Convolutions on Graphs
https://distill.pub/2021/understanding-gnns/

See also: A Gentle Introduction to Graph Neural Networks
https://distill.pub/2021/gnn-intro/

Both by Google Research teams

AlphaFold2



Highly accurate protein structure prediction with AlphaFold

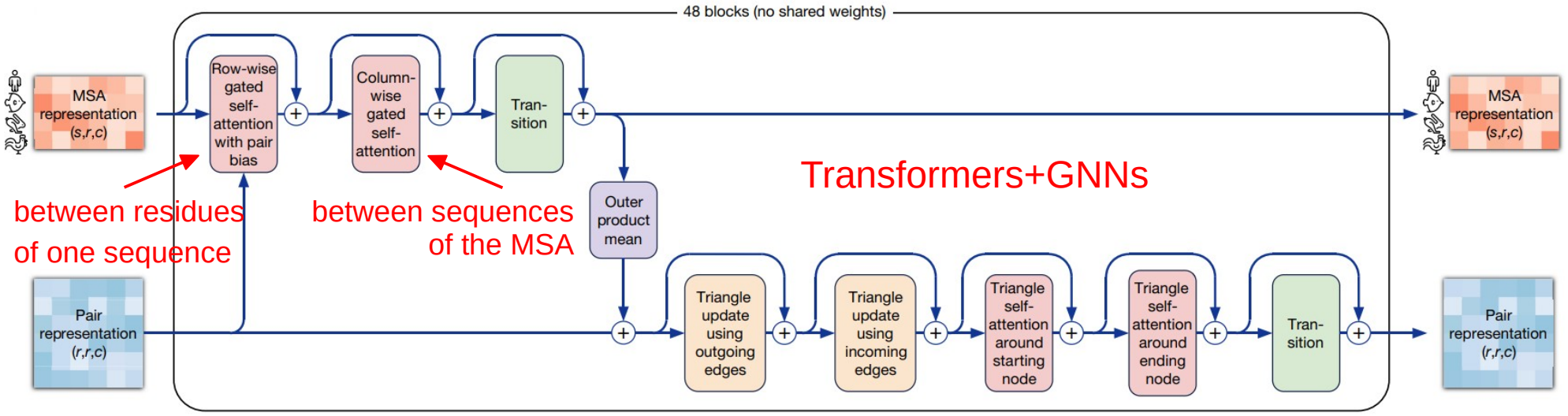
https://doi.org/10.1038/s41586-021-03819-2
Received: 11 May 2021
Accepted: 12 July 2021
Published online: 15 July 2021
Open access
Check for updates

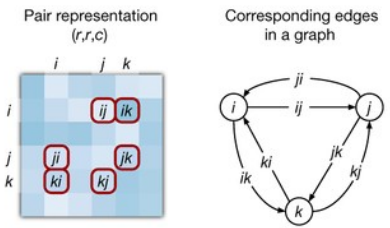
John Jumper[1,4]✉, Richard Evans[1,4], Alexander Pritzel[1,4], Tim Green[1,4], Michael Figurnov[1,4], Olaf Ronneberger[1,4], Kathryn Tunyasuvunakool[1,4], Russ Bates[1,4], Augustin Žídek[1,4], Anna Potapenko[1,4], Alex Bridgland[1,4], Clemens Meyer[1,4], Simon A. A. Kohl[1,4], Andrew J. Ballard[1,4], Andrew Cowie[1,4], Bernardino Romera-Paredes[1,4], Stanislav Nikolov[1,4], Rishub Jain[1,4], Jonas Adler[1], Trevor Back[1], Stig Petersen[1], David Reiman[1], Ellen Clancy[1], Michal Zielinski[1], Martin Steinegger[2,3], Michalina Pacholska[1], Tamas Berghammer[1], Sebastian Bodenstein[1], David Silver[1], Oriol Vinyals[1], Andrew W. Senior[1], Koray Kavukcuoglu[1], Pushmeet Kohli[1] & Demis Hassabis[1,4]✉

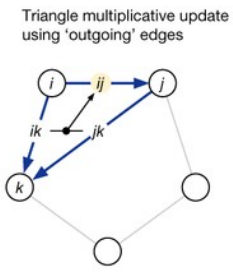
~93 million parameters (weights+biases)

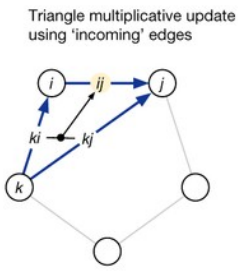
https://github.com/google-deepmind/alphafold

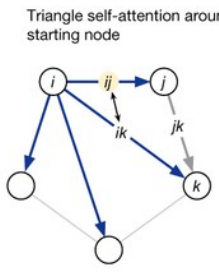
AlphaFold2: evoformer

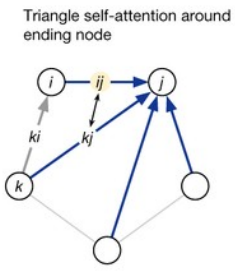












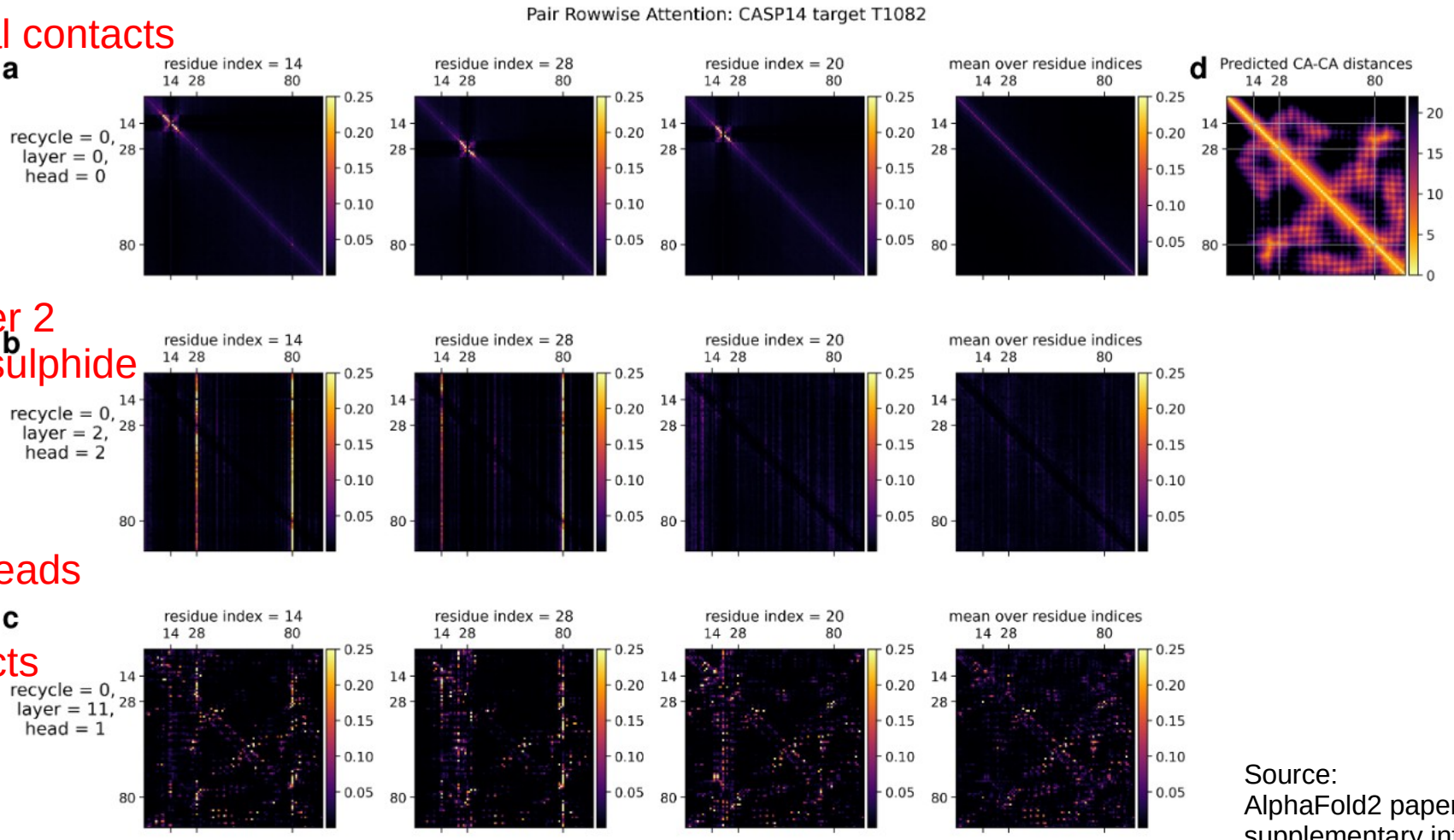
see also: https://www.blopig.com/blog/2021/07/alphafold-2-is-here-whats-behind-the-structure-prediction-miracle/

Row-wise attention: between residues of a sequence

Layer 0 = local contacts

Head 2 of layer 2 recognises disulphide bonds

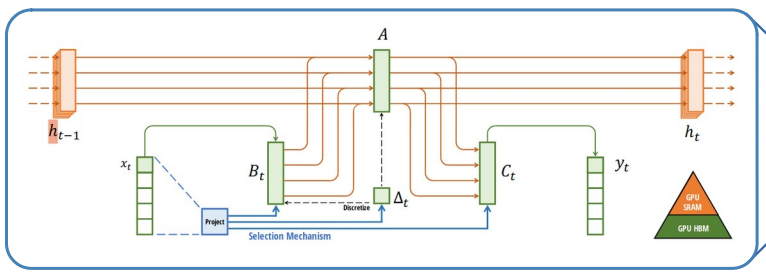
At layer 11, heads recognise distant contacts



Source: AlphaFold2 paper supplementary info

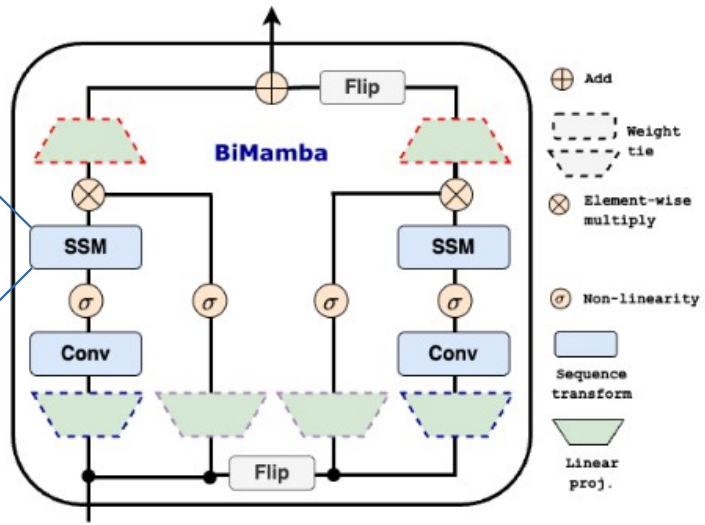
RNNs are back. Rise of the Mamba

"Attention" = embedding size x input length
→ linear growth with input length
(not quadratic like transformers)



SSM = State Space Model

Prediction/classification



...AGGCTAGGATATCGATAAGCTGACTGAT...

The model learns about variants' Interactions

→ Reusable foundation model

Gu and Dao (2023) *arXiv*:2312.00752*:*
Schiff *et al* (2024) a*rXiv*:2403.03234

PreciDIAB
National Center for
Precision Diabetic Medicine

E.g.i.d.
European
Genomic
Institute
for Diabetes

Région
Hauts-de-France

UMR 1283
8199
Functional (Epi)genomics and Mechanisms of Type 2 Diabetes
and Related Diseases