PreciDIAB
National Center for
Precision Diabetic Medicine

Egid
European
Genomic
Institute
for Diabetes

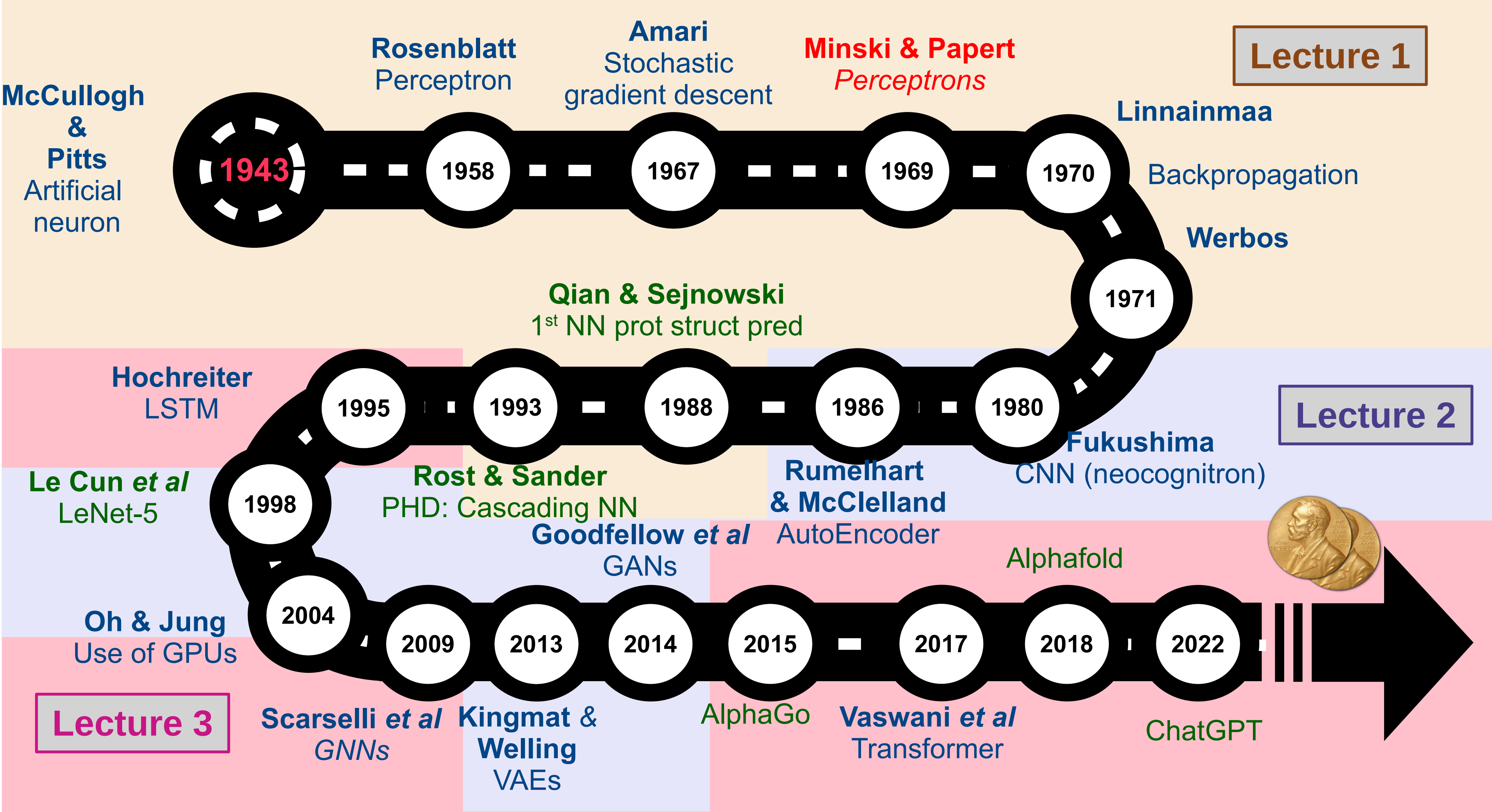
Beyond MLPs

Part 2/2: RNNs, Attention and GNNs

nicolas.gambardella@univ-lille.fr

Région
Hauts-de-France

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



Lecture 1

1943 McCullogh & Pitts Artificial neuron

1958 Rosenblatt Perceptron

1967 Amari Stochastic gradient descent

1969 Minski & Papert *Perceptrons*

1970 Linnainmaa Backpropagation

1971 Werbos

Lecture 2

1980 Fukushima CNN (neocognitron)

1986 Rumelhart & McClelland AutoEncoder

1988 Qian & Sejnowski 1st NN prot struct pred

1993 Rost & Sander PHD: Cascading NN

1995 Hochreiter LSTM

1998 Le Cun *et al* LeNet-5

Lecture 3

2004 Oh & Jung Use of GPUs

2009 Scarselli *et al* *GNNs*

2013 Kingmat & Welling VAEs

2014 Goodfellow *et al* GANs

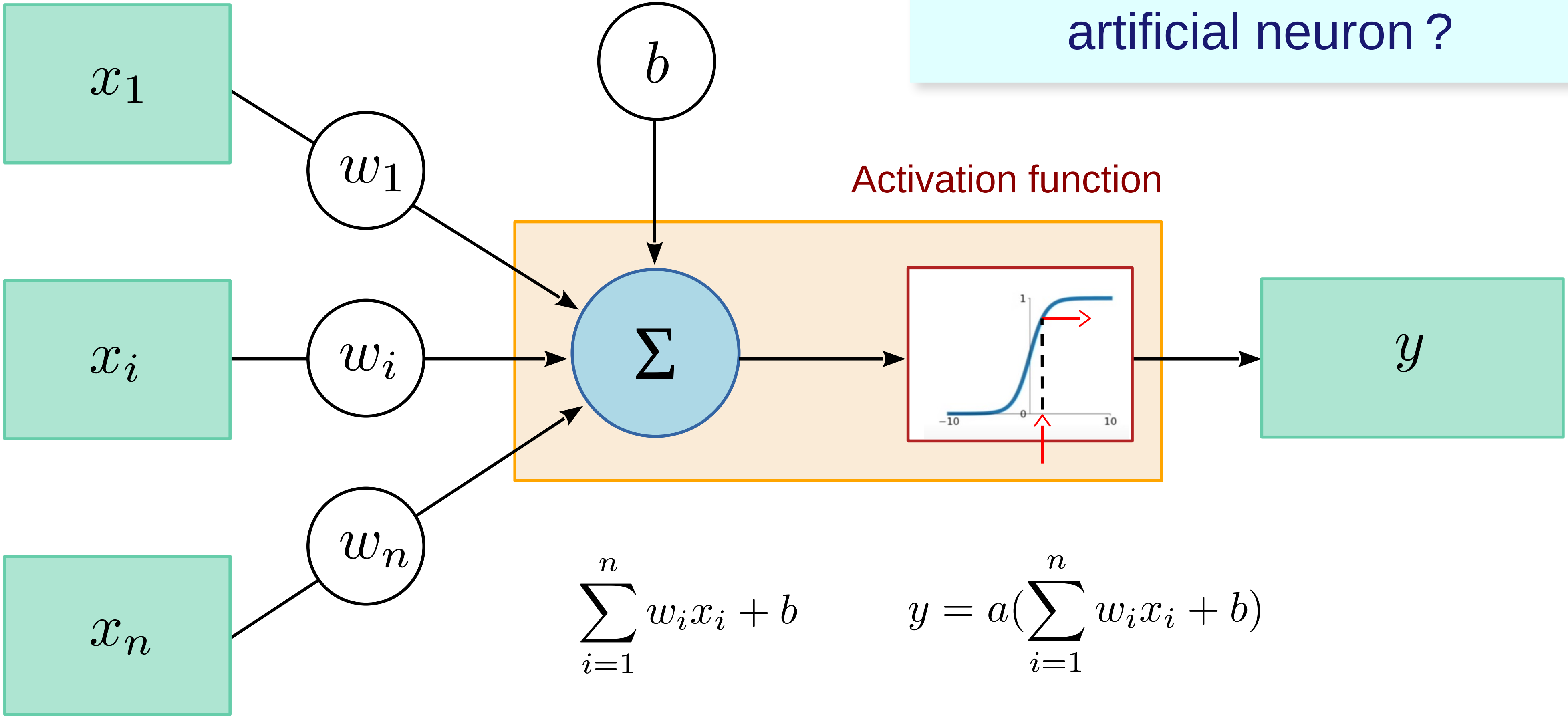
2015 AlphaGo

2017 Vaswani *et al* Transformer

2018 Alphafold

2022 ChatGPT

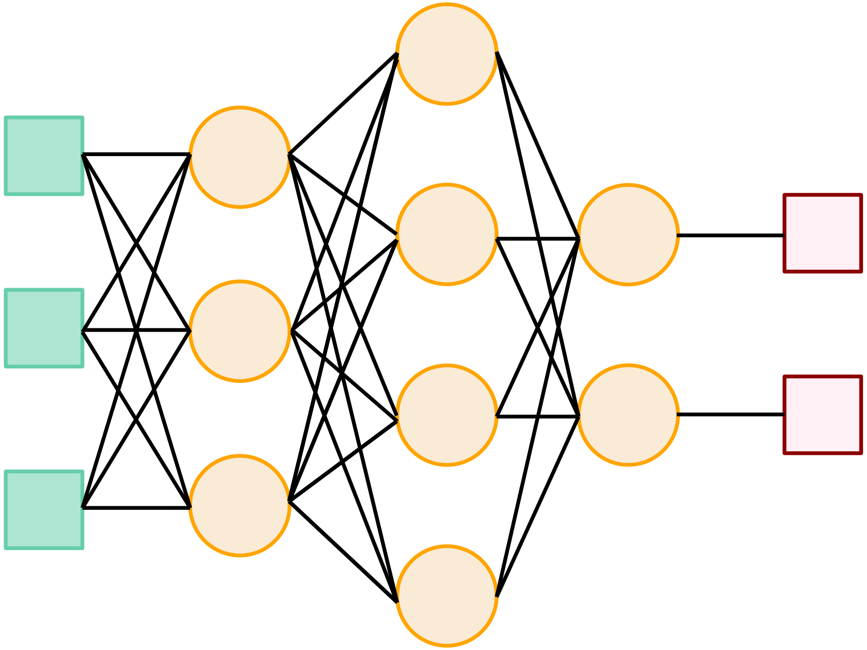
What is an artificial neuron ?



$$\sum_{i=1}^{n} w_i x_i + b$$

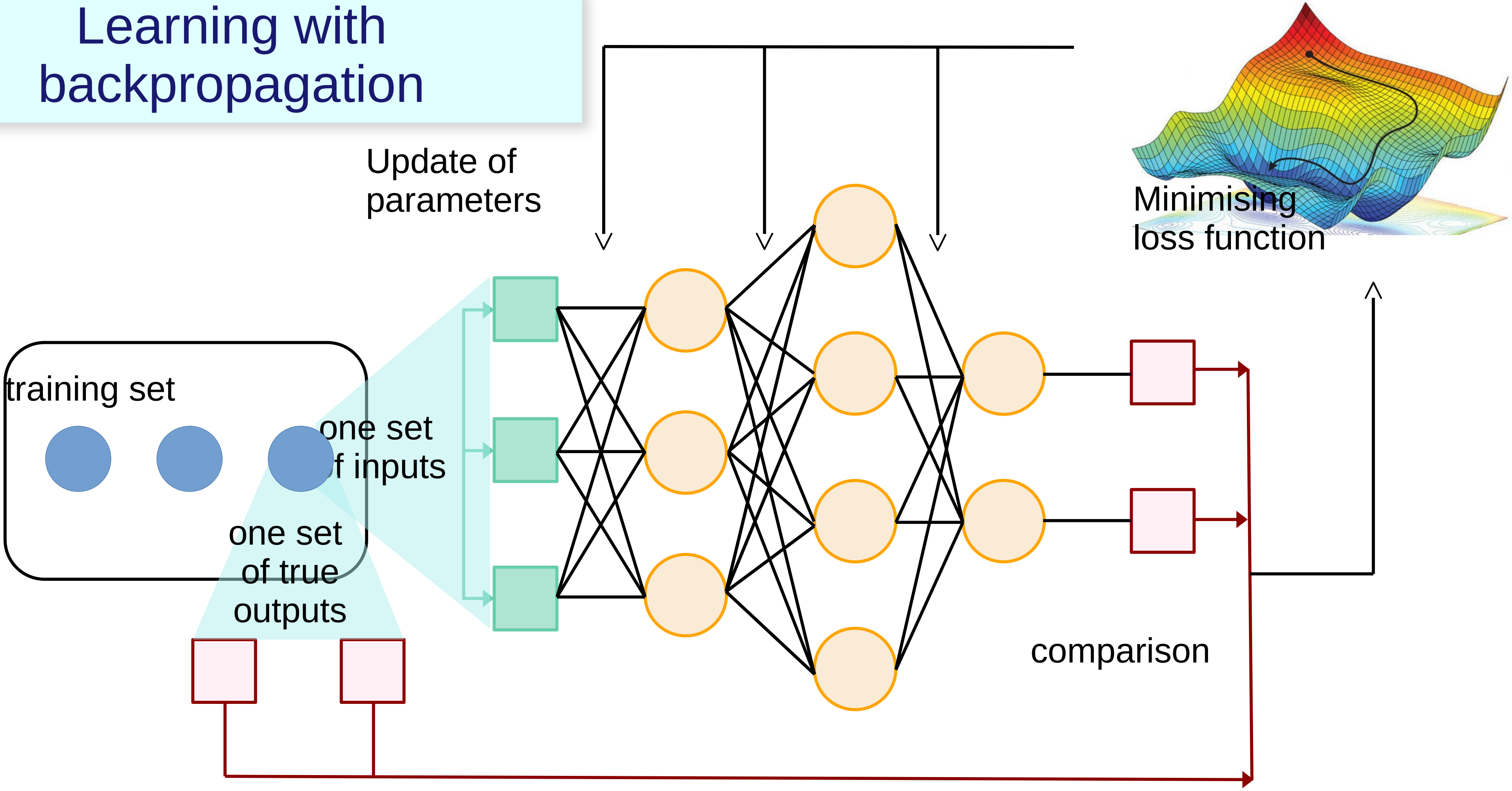
$$y = a(\sum_{i=1}^{n} w_i x_i + b)$$

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



Update of parameters

Minimising loss function

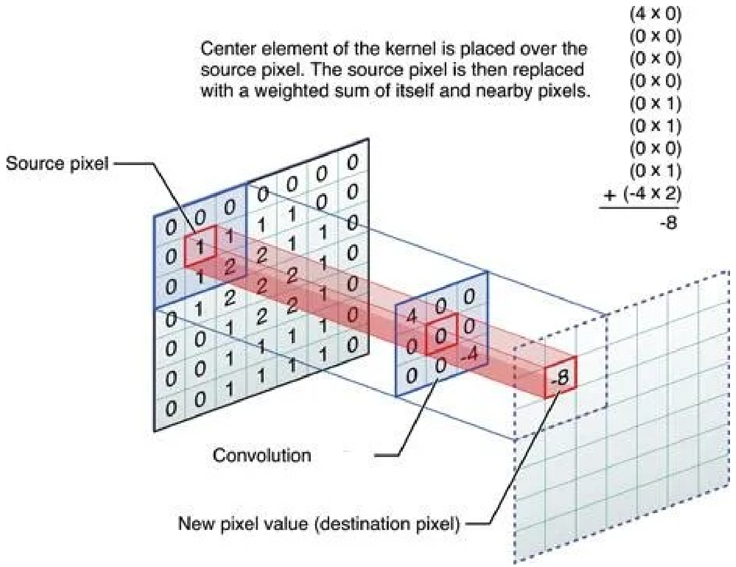
training set

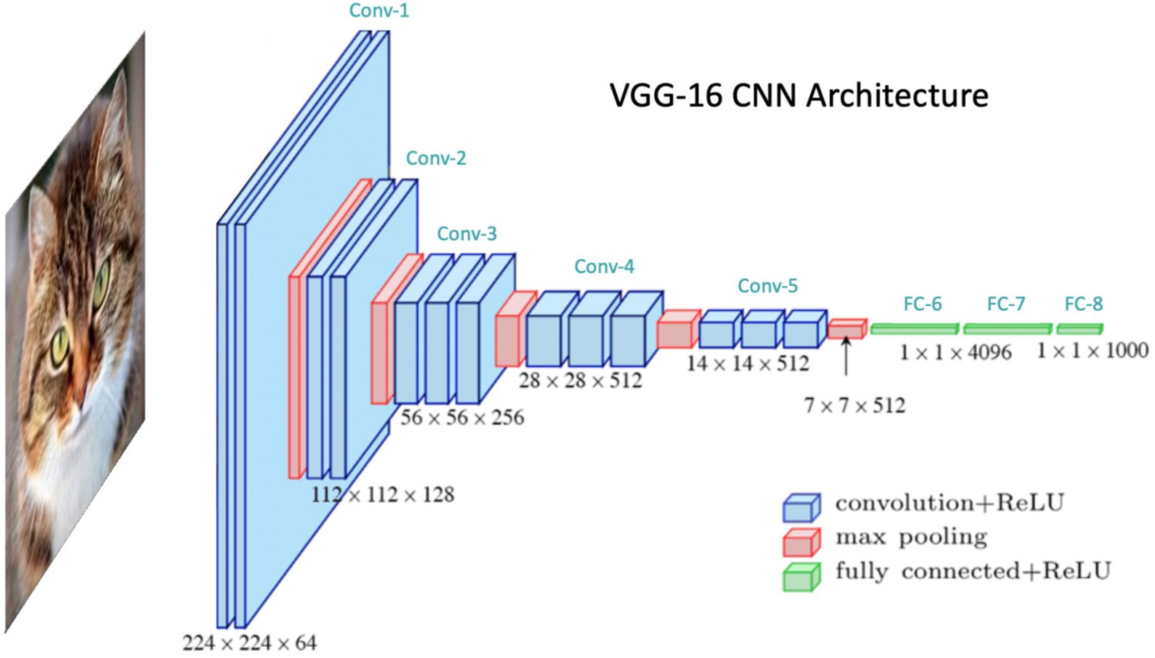
one set of inputs

one set of true outputs

comparison

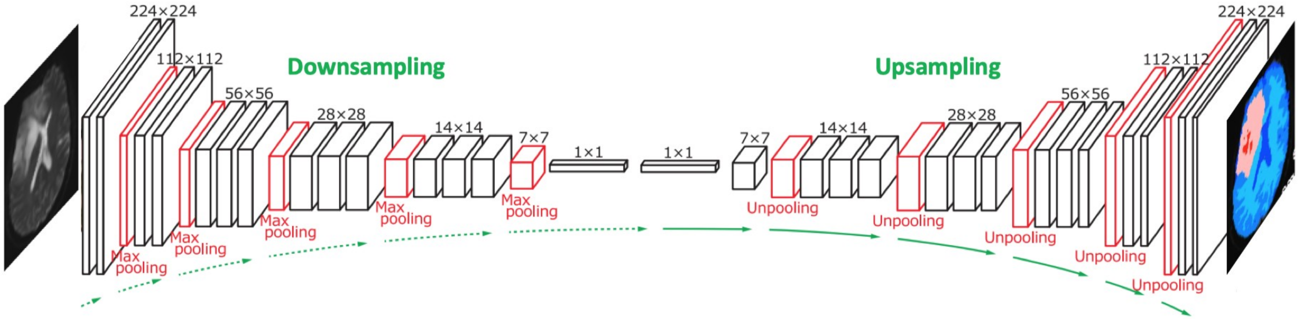
We can detect local features by linking neighbouring inputs

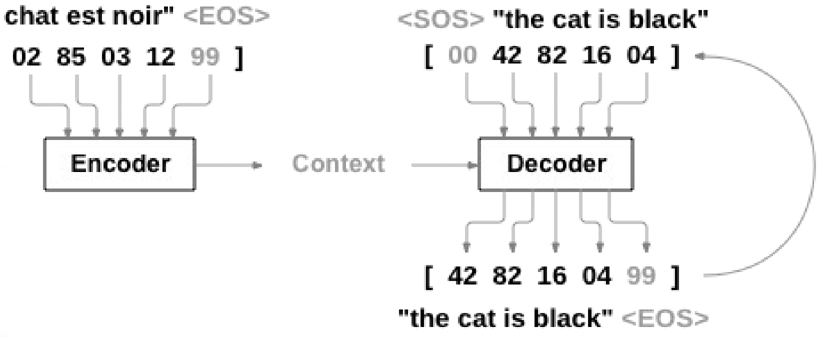

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8
Source pixel
Convolution
New pixel value (destination pixel)
4 0 0
0 0 0
0 0 -4
-8

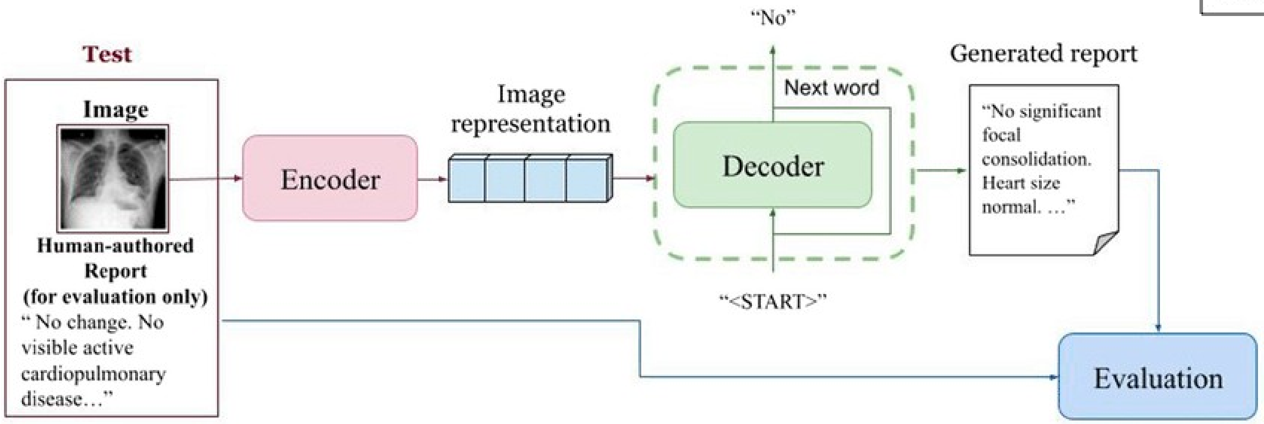


Deep Convolutional Neural Networks (CNN)

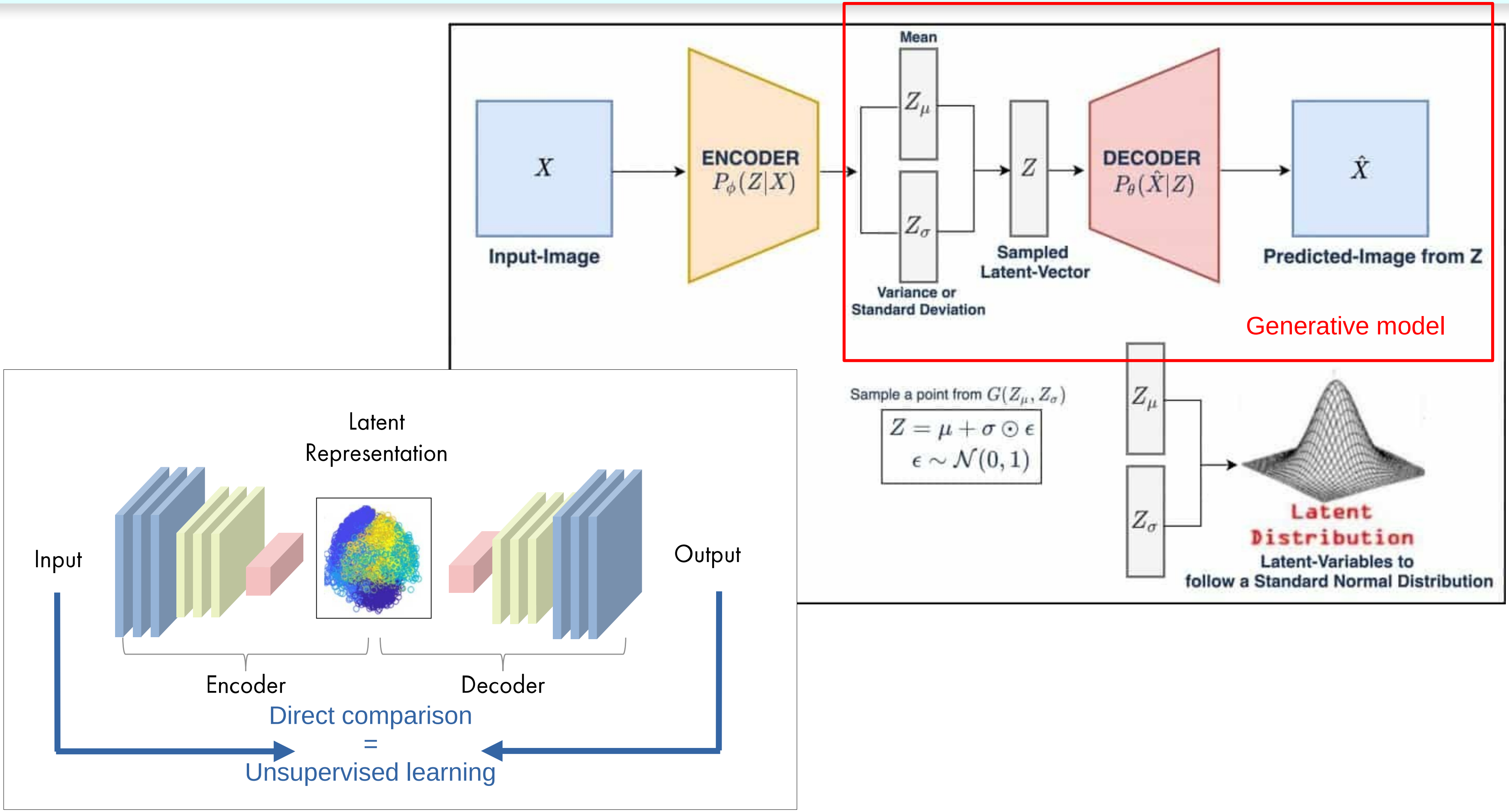
Encoder networks can embed information in a latent space
Decoder networks can reconstruct the information from it






Test
Image
Human-authored Report (for evaluation only)
" No change. No visible active cardiopulmonary disease…"
Encoder
Image representation
"No"
Next word
Decoder
"<START>"
Generated report
"No significant focal consolidation. Heart size normal. …"
Evaluation

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



New Results

Deep learning models reading clinical data and liver omics strongly distinguish NASH from steatosis and suggest new genes involved in liver disease severity

Nicolas Gambardella, Smaïn Fettem, Mathilde Boissel, Lijiao Ning, Violeta Raverdy, Marwa Afnouch, Souhila Amanzougarene, Mehdi Derhourhi, Bénédicte Toussaint, Emmanuel Vaillant, Amna Khamis, Philippe Lefebvre, Bart Staels, Francois Pattou, Philippe Froguel, Amelie Bonnefond

doi: https://doi.org/10.1101/2025.10.10.681581

This article is a preprint and has not been certified by peer review [what does this mean?].

Posted October 10, 2025.

Subject Area

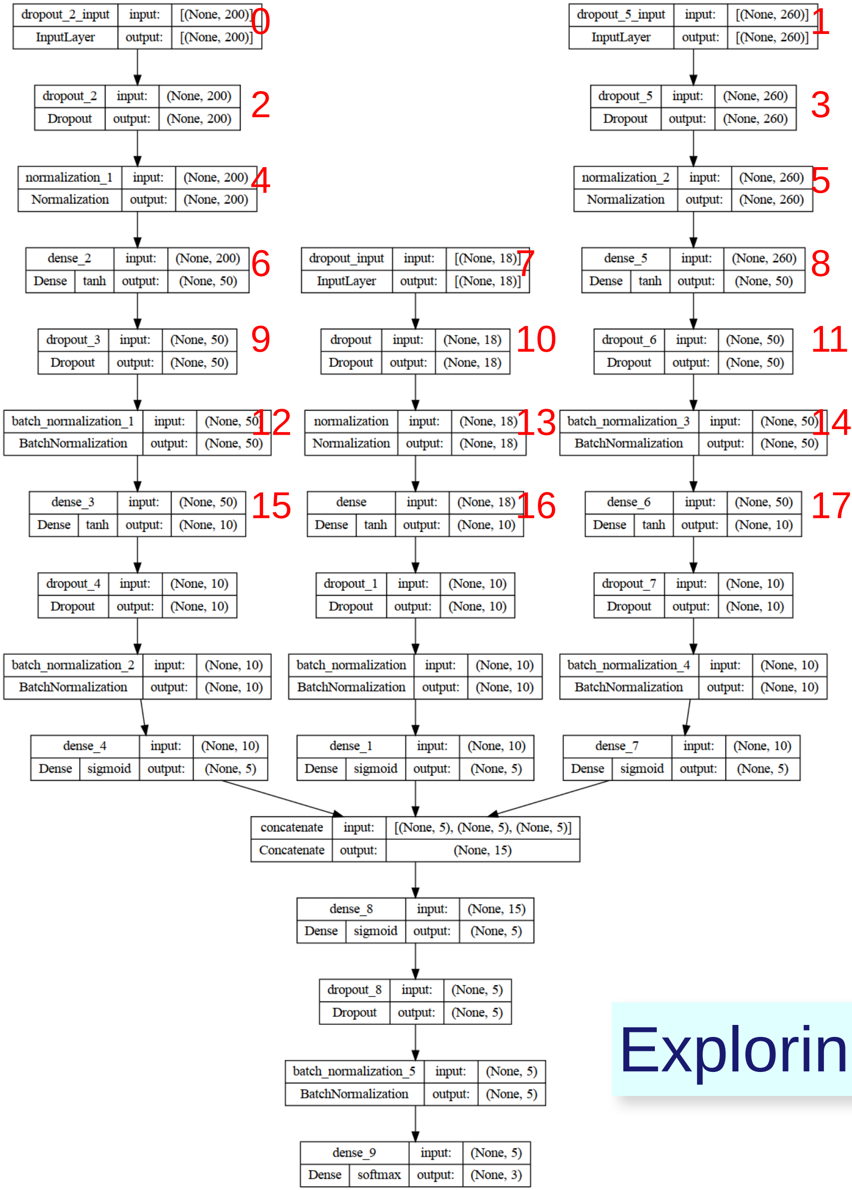
Molecular Biology

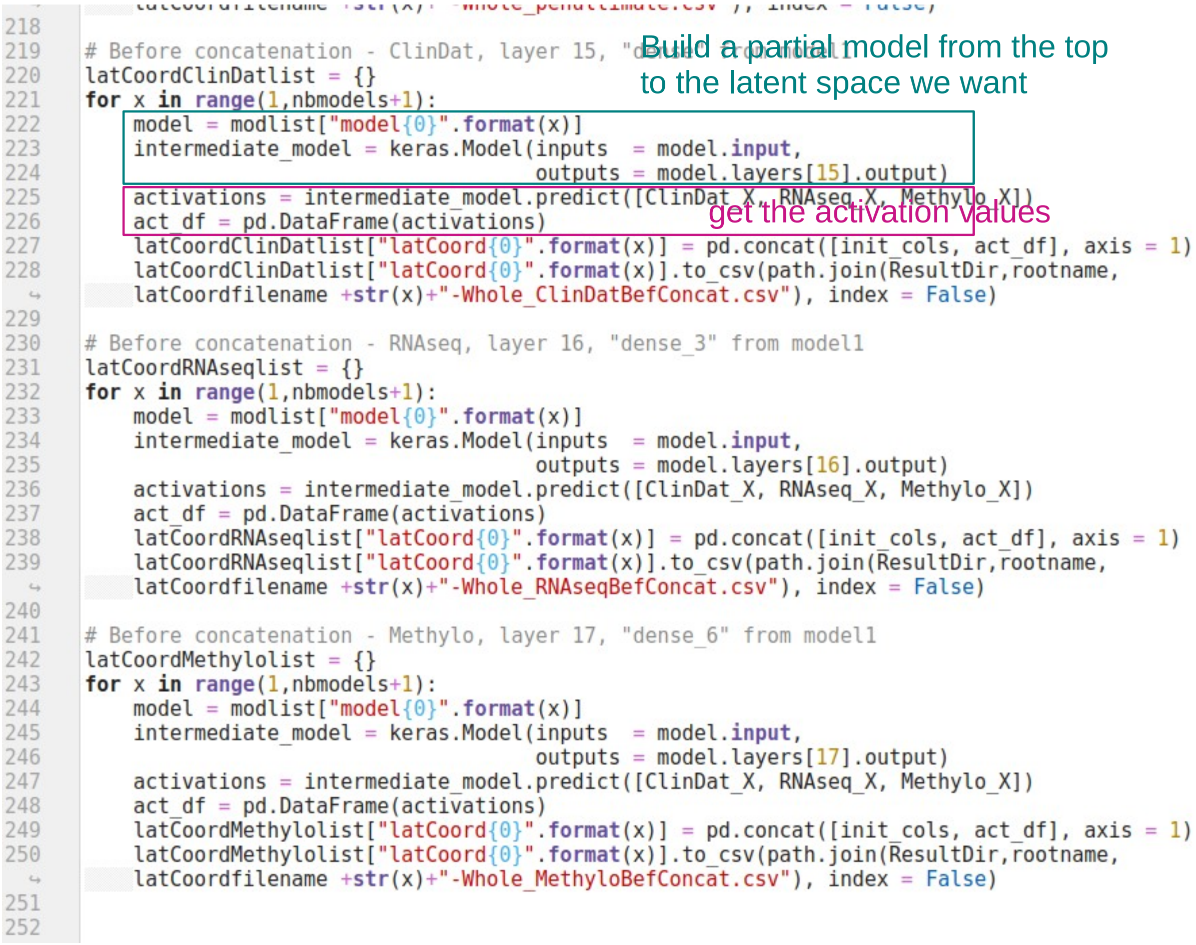
Abstract

Background & Aims Metabolic dysfunction-associated steatotic liver disease (MASLD) is a frequent co-morbidity of obesity and diabetes, with prevalence increasing worldwide. Recognising liver disease stages and elucidating the molecular underpinning of their progression are thus medically important. Methods Using data gathered from 300 patients with obesity of the ABOS cohort, we selected non-redundant clinical variables, gene expressions and CpGs methylation levels most associated with severity using unsupervised approaches to train a multi-module, multi-layer perceptron predicting patients liver status. Results The combination of five models trained on the three modalities reached an AUC of 0.945 on a

Reviews and Context

0	Comment
0	TRIP Peer Reviews
0	Community Reviews
1	Automated Services
0	Blogs/Media
0	Author Videos


dropout_2_input | InputLayer | input: [(None, 200)] | output: [(None, 200)]
0
dropout_5_input | InputLayer | input: [(None, 260)] | output: [(None, 260)]
1
dropout_2 | Dropout | input: (None, 200) | output: (None, 200)
2
dropout_5 | Dropout | input: (None, 260) | output: (None, 260)
3
normalization_1 | Normalization | input: (None, 200) | output: (None, 200)
4
normalization_2 | Normalization | input: (None, 260) | output: (None, 260)
5
dense_2 | Dense | tanh | input: (None, 200) | output: (None, 50)
6
dropout_input | InputLayer | input: [(None, 18)] | output: [(None, 18)]
7
dense_5 | Dense | tanh | input: (None, 260) | output: (None, 50)
8
dropout_3 | Dropout | input: (None, 50) | output: (None, 50)
9
dropout | Dropout | input: (None, 18) | output: (None, 18)
10
dropout_6 | Dropout | input: (None, 50) | output: (None, 50)
11
batch_normalization_1 | BatchNormalization | input: (None, 50) | output: (None, 50)
12
normalization | Normalization | input: (None, 18) | output: (None, 18)
13
batch_normalization_3 | BatchNormalization | input: (None, 50) | output: (None, 50)
14
dense_3 | Dense | tanh | input: (None, 50) | output: (None, 10)
15
dense | Dense | tanh | input: (None, 18) | output: (None, 10)
16
dense_6 | Dense | tanh | input: (None, 50) | output: (None, 10)
17
dropout_4 | Dropout | input: (None, 10) | output: (None, 10)
dropout_1 | Dropout | input: (None, 10) | output: (None, 10)
dropout_7 | Dropout | input: (None, 10) | output: (None, 10)
batch_normalization_2 | BatchNormalization | input: (None, 10) | output: (None, 10)
batch_normalization | BatchNormalization | input: (None, 10) | output: (None, 10)
batch_normalization_4 | BatchNormalization | input: (None, 10) | output: (None, 10)
dense_4 | Dense | sigmoid | input: (None, 10) | output: (None, 5)
dense_1 | Dense | sigmoid | input: (None, 10) | output: (None, 5)
dense_7 | Dense | sigmoid | input: (None, 10) | output: (None, 5)
concatenate | Concatenate | input: [(None, 5), (None, 5), (None, 5)] | output: (None, 15)
dense_8 | Dense | sigmoid | input: (None, 15) | output: (None, 5)
dropout_8 | Dropout | input: (None, 5) | output: (None, 5)
batch_normalization_5 | BatchNormalization | input: (None, 5) | output: (None, 5)
dense_9 | Dense | softmax | input: (None, 5) | output: (None, 3)



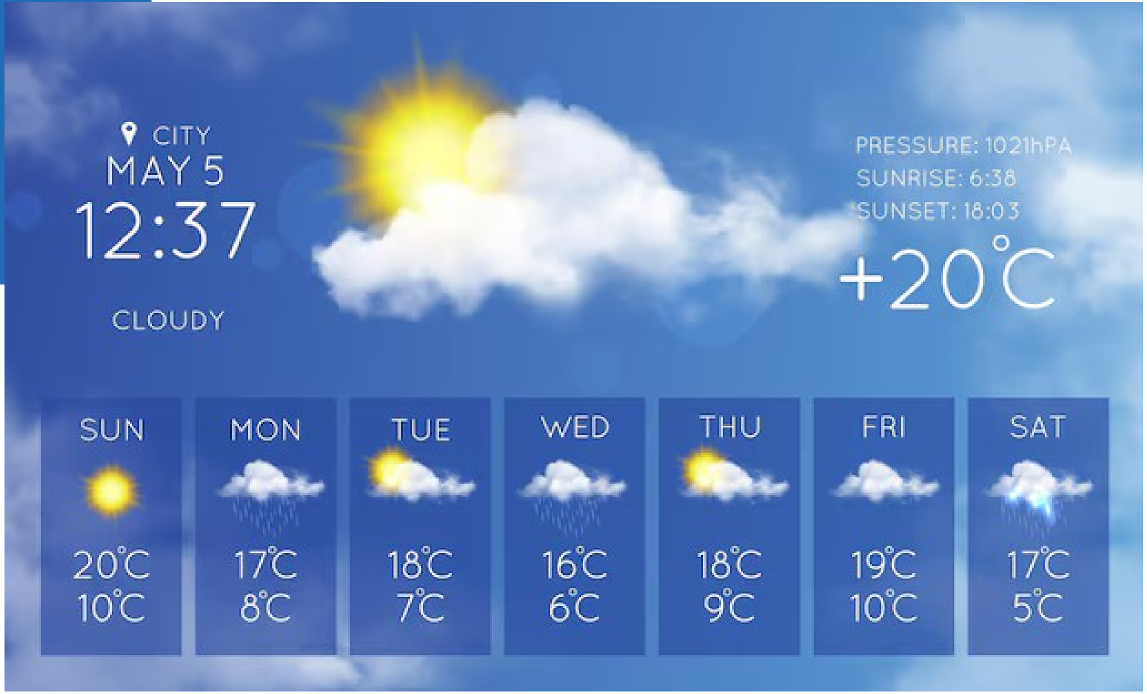
```python
    latCoordfilename +str(x)+"-Whole_penultimate.csv"), index = False)

# Before concatenation - ClinDat, layer 15, "dense" from model1
latCoordClinDatlist = {}
for x in range(1,nbmodels+1):
    model = modlist["model{0}".format(x)]
    intermediate_model = keras.Model(inputs  = model.input,
                                     outputs = model.layers[15].output)
    activations = intermediate_model.predict([ClinDat_X, RNAseq_X, Methylo_X])
    act_df = pd.DataFrame(activations)
    latCoordClinDatlist["latCoord{0}".format(x)] = pd.concat([init_cols, act_df], axis = 1)
    latCoordClinDatlist["latCoord{0}".format(x)].to_csv(path.join(ResultDir,rootname,
    latCoordfilename +str(x)+"-Whole_ClinDatBefConcat.csv"), index = False)

# Before concatenation - RNAseq, layer 16, "dense_3" from model1
latCoordRNAseqlist = {}
for x in range(1,nbmodels+1):
    model = modlist["model{0}".format(x)]
    intermediate_model = keras.Model(inputs  = model.input,
                                     outputs = model.layers[16].output)
    activations = intermediate_model.predict([ClinDat_X, RNAseq_X, Methylo_X])
    act_df = pd.DataFrame(activations)
    latCoordRNAseqlist["latCoord{0}".format(x)] = pd.concat([init_cols, act_df], axis = 1)
    latCoordRNAseqlist["latCoord{0}".format(x)].to_csv(path.join(ResultDir,rootname,
    latCoordfilename +str(x)+"-Whole_RNAseqBefConcat.csv"), index = False)

# Before concatenation - Methylo, layer 17, "dense_6" from model1
latCoordMethylolist = {}
for x in range(1,nbmodels+1):
    model = modlist["model{0}".format(x)]
    intermediate_model = keras.Model(inputs  = model.input,
                                     outputs = model.layers[17].output)
    activations = intermediate_model.predict([ClinDat_X, RNAseq_X, Methylo_X])
    act_df = pd.DataFrame(activations)
    latCoordMethylolist["latCoord{0}".format(x)] = pd.concat([init_cols, act_df], axis = 1)
    latCoordMethylolist["latCoord{0}".format(x)].to_csv(path.join(ResultDir,rootname,
    latCoordfilename +str(x)+"-Whole_MethyloBefConcat.csv"), index = False)
```

Build a partial model from the top to the latent space we want

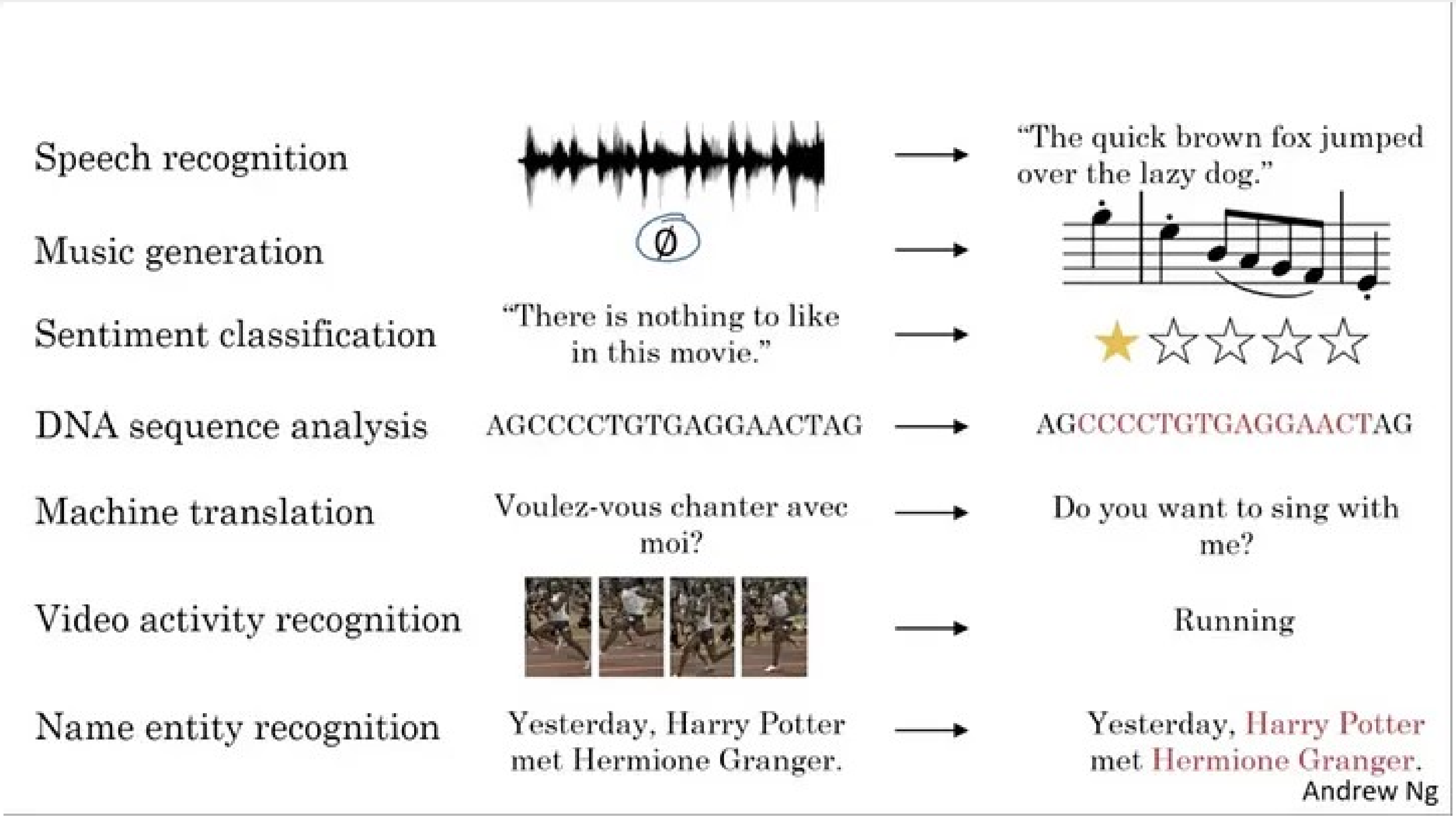
get the activation values

Exploring latent spaces

Time series and sequences of variable lengths



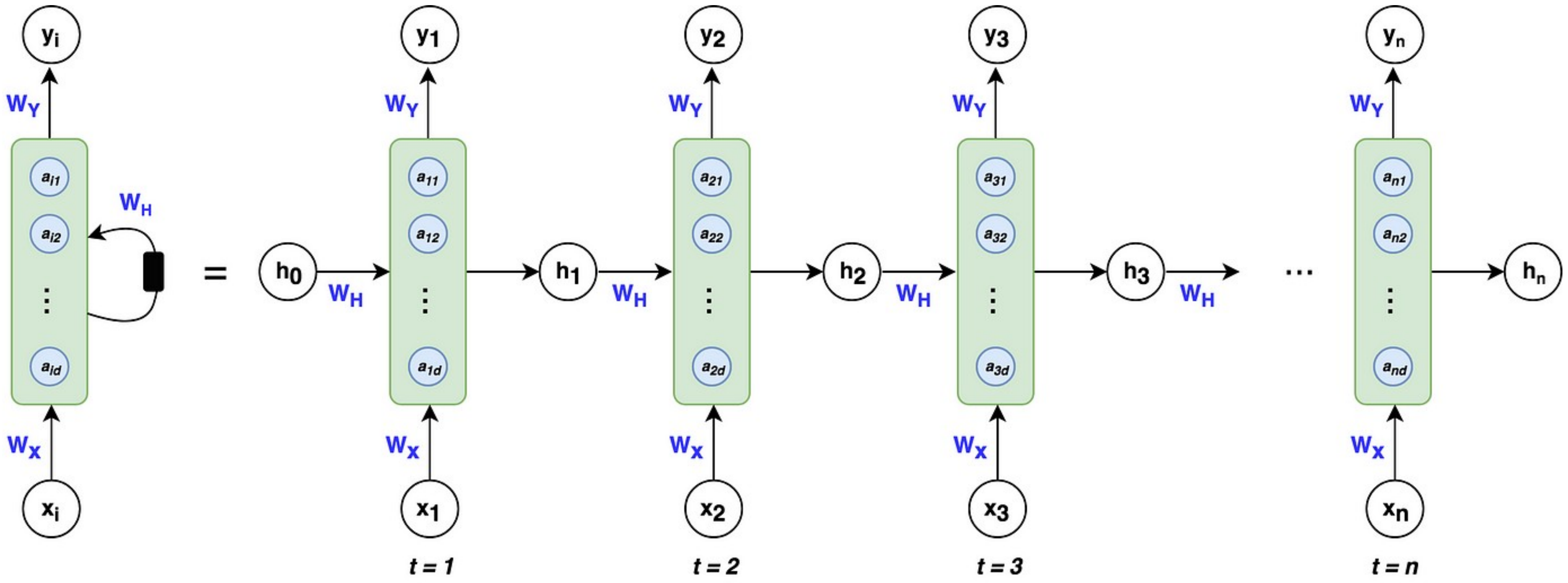
Time series and sequences of variable lengths



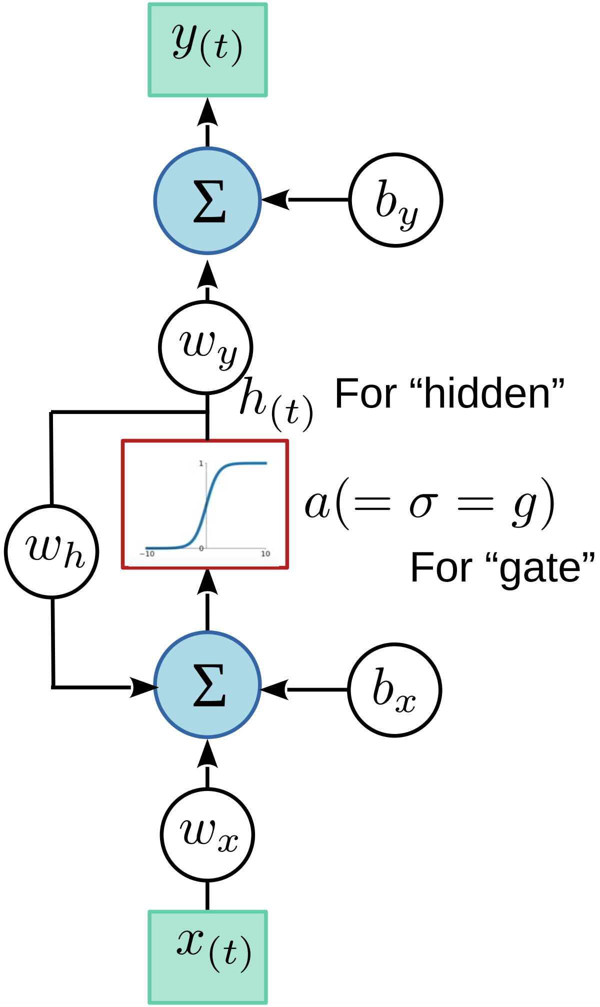
Speech recognition		→	"The quick brown fox jumped over the lazy dog."
Music generation	∅	→	
Sentiment classification	"There is nothing to like in this movie."	→	★☆☆☆☆
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	→	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	→	Do you want to sing 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

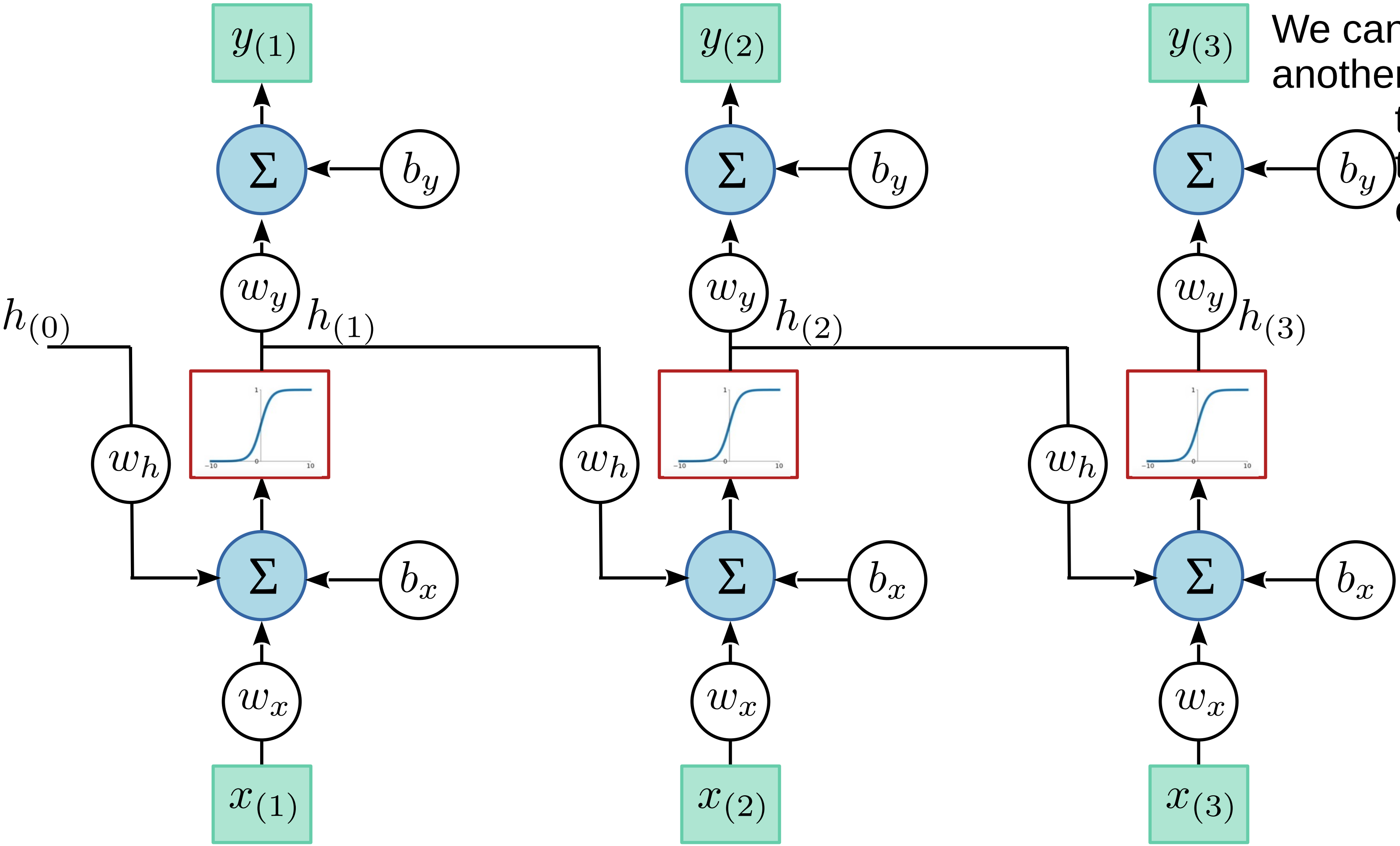
$$y_{(t)} = w_y \times h_{(t)} + b_y$$

not the same timepoint

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

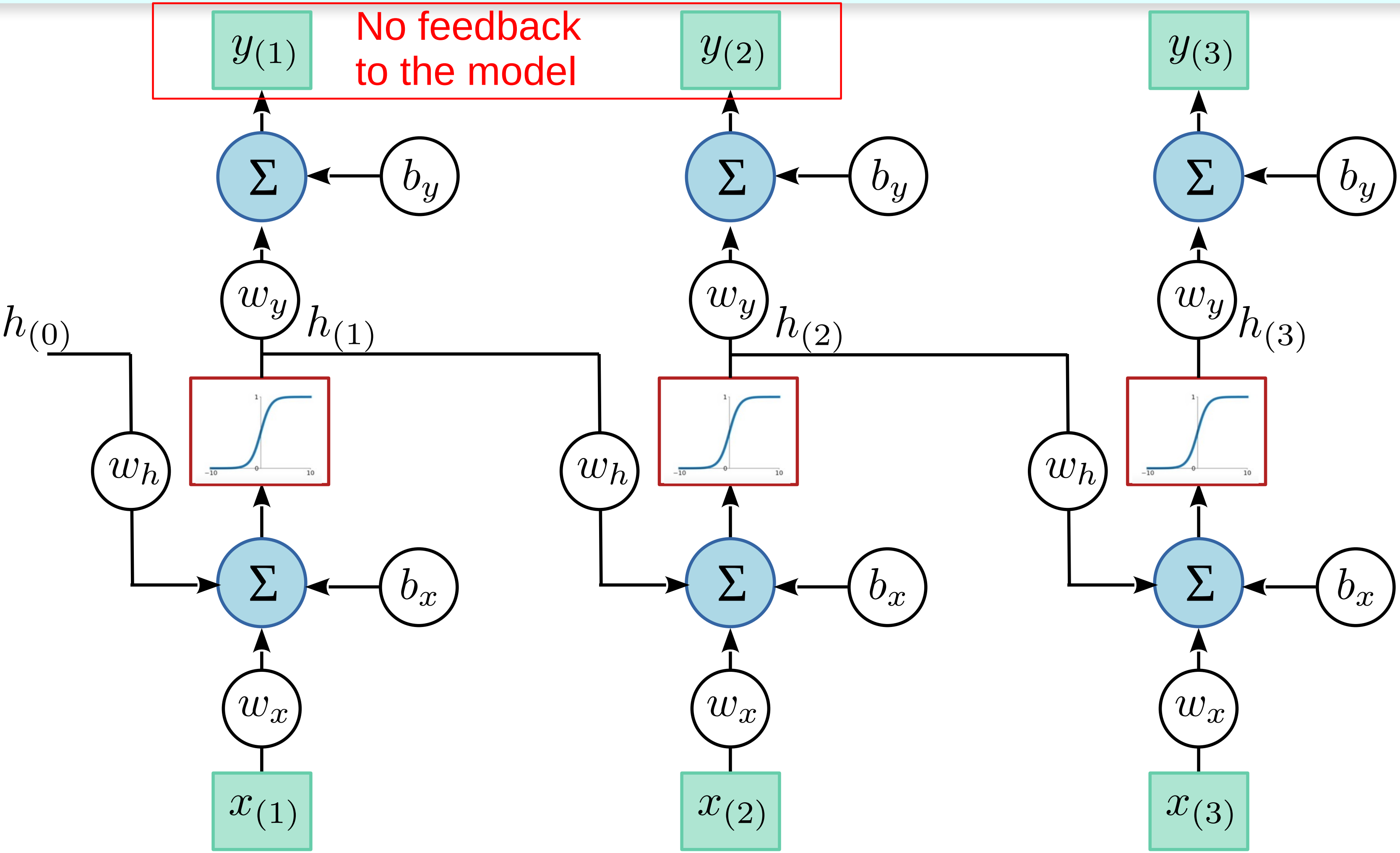
“memory”

RNN: 1 cell - unfolding



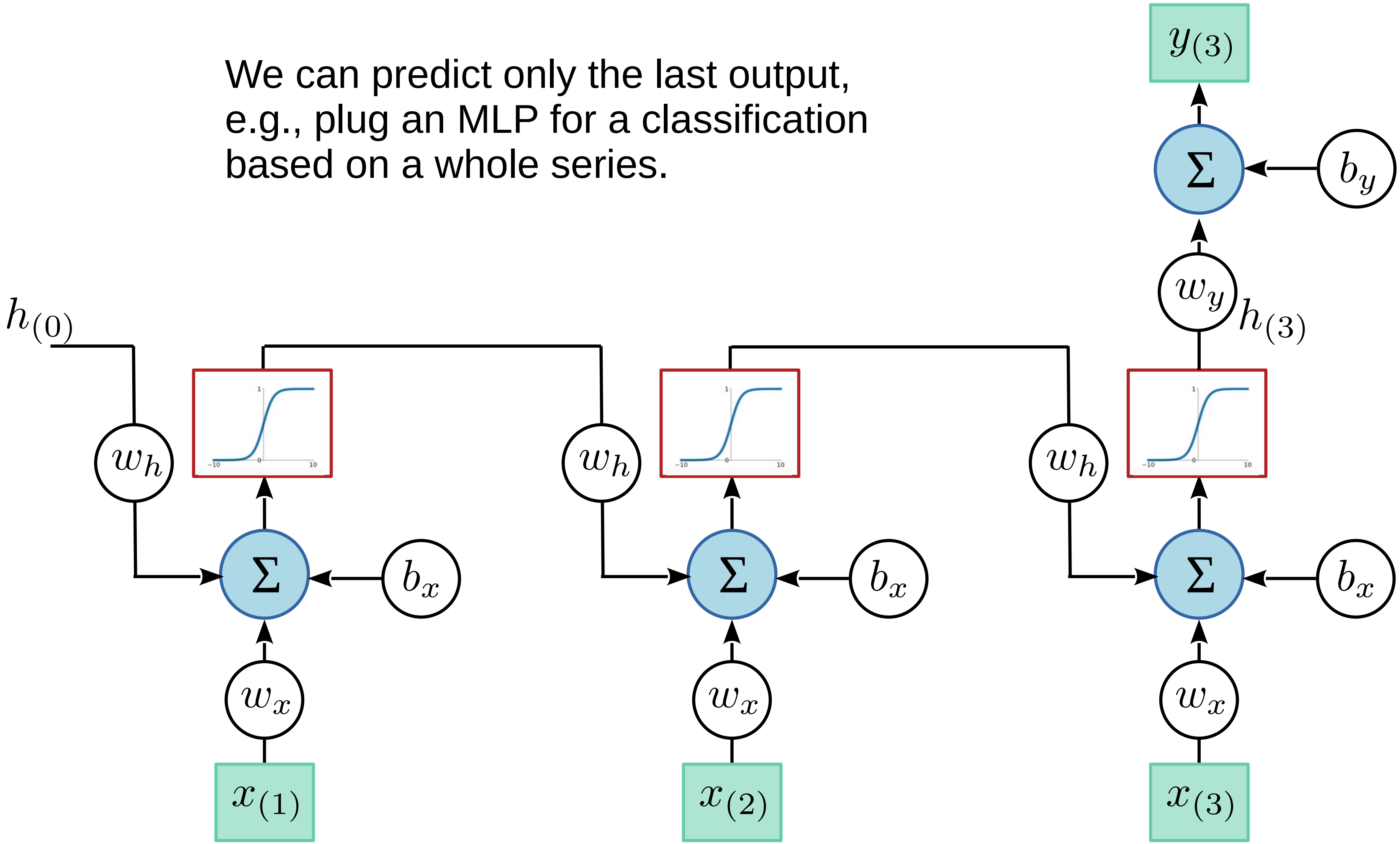
We can output another series, e.g, transcription, translation, conversion.

RNN: 1 cell - unfolding

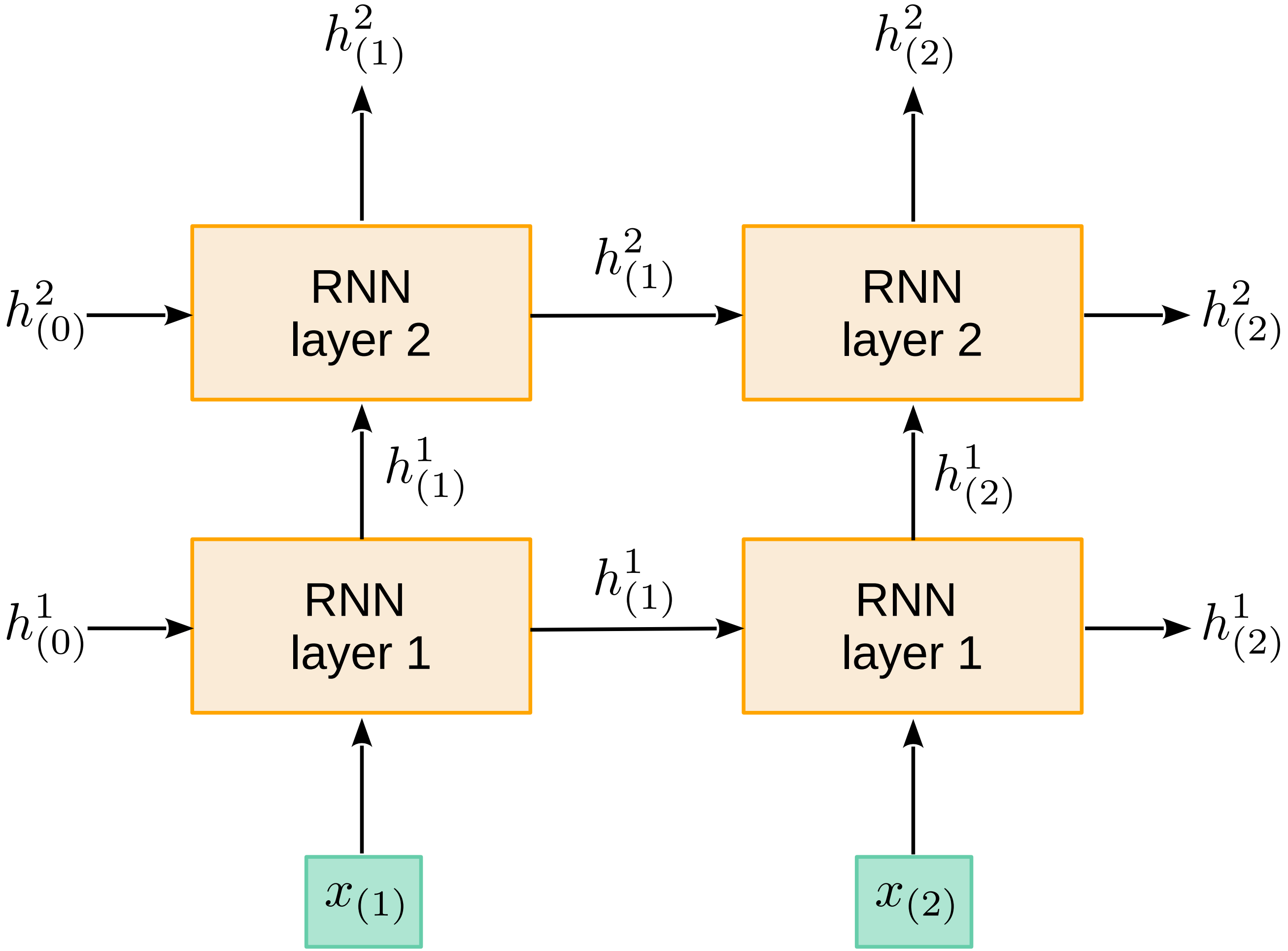


RNN: 1 cell - unfolding

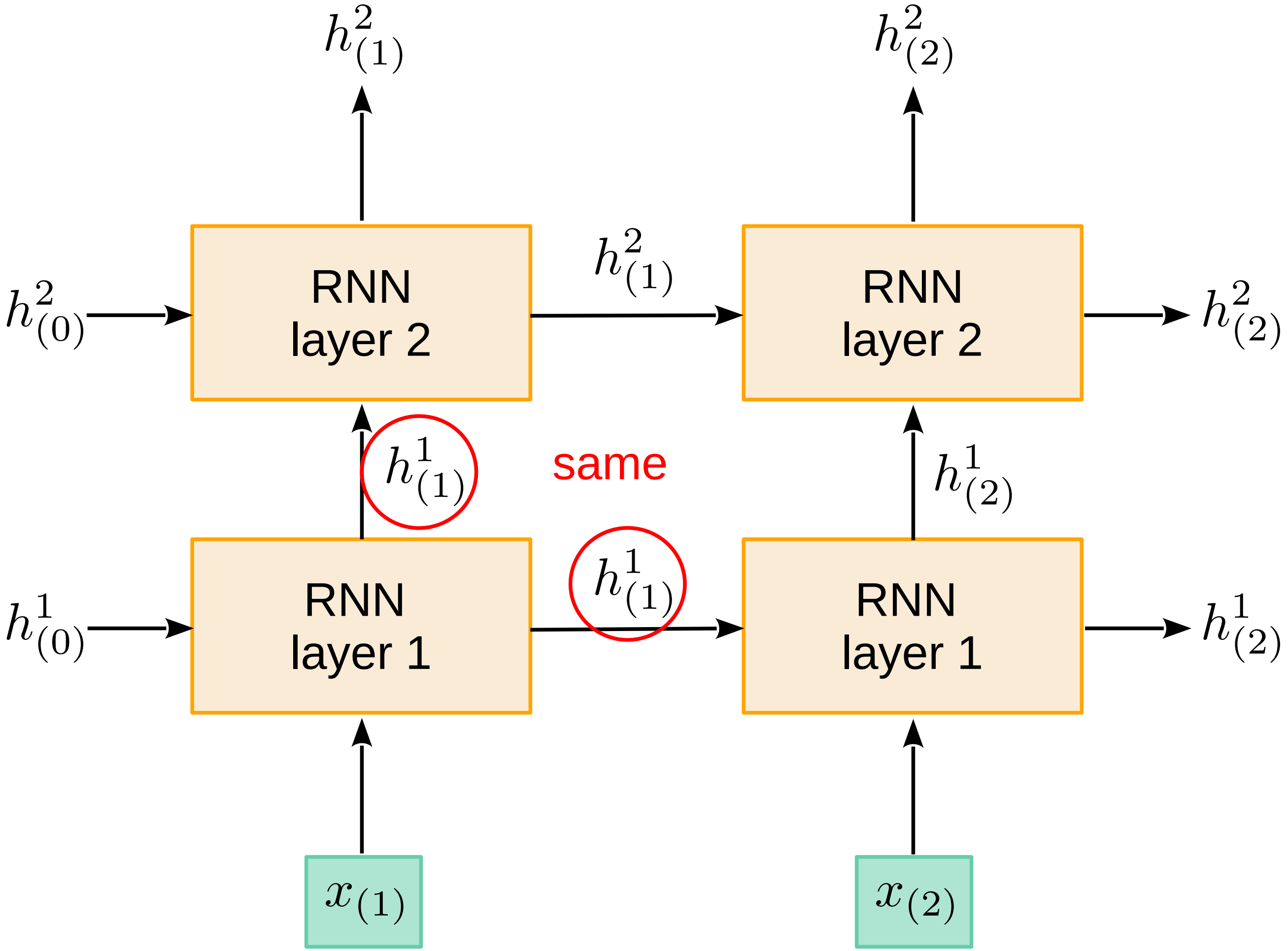
We can predict only the last output, e.g., plug an MLP for a classification based on a whole series.



Stacked RNNs

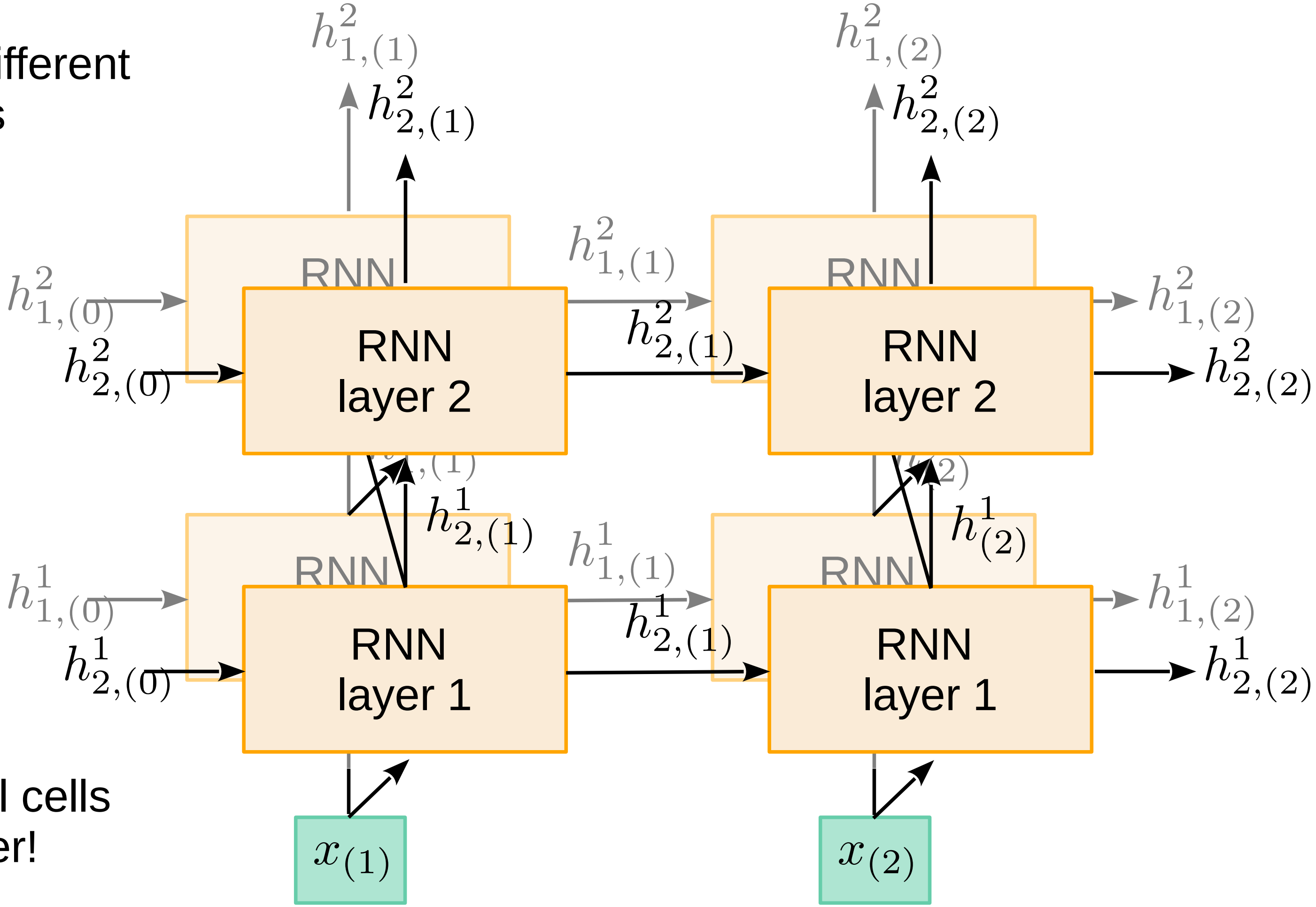


Stacked RNN


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

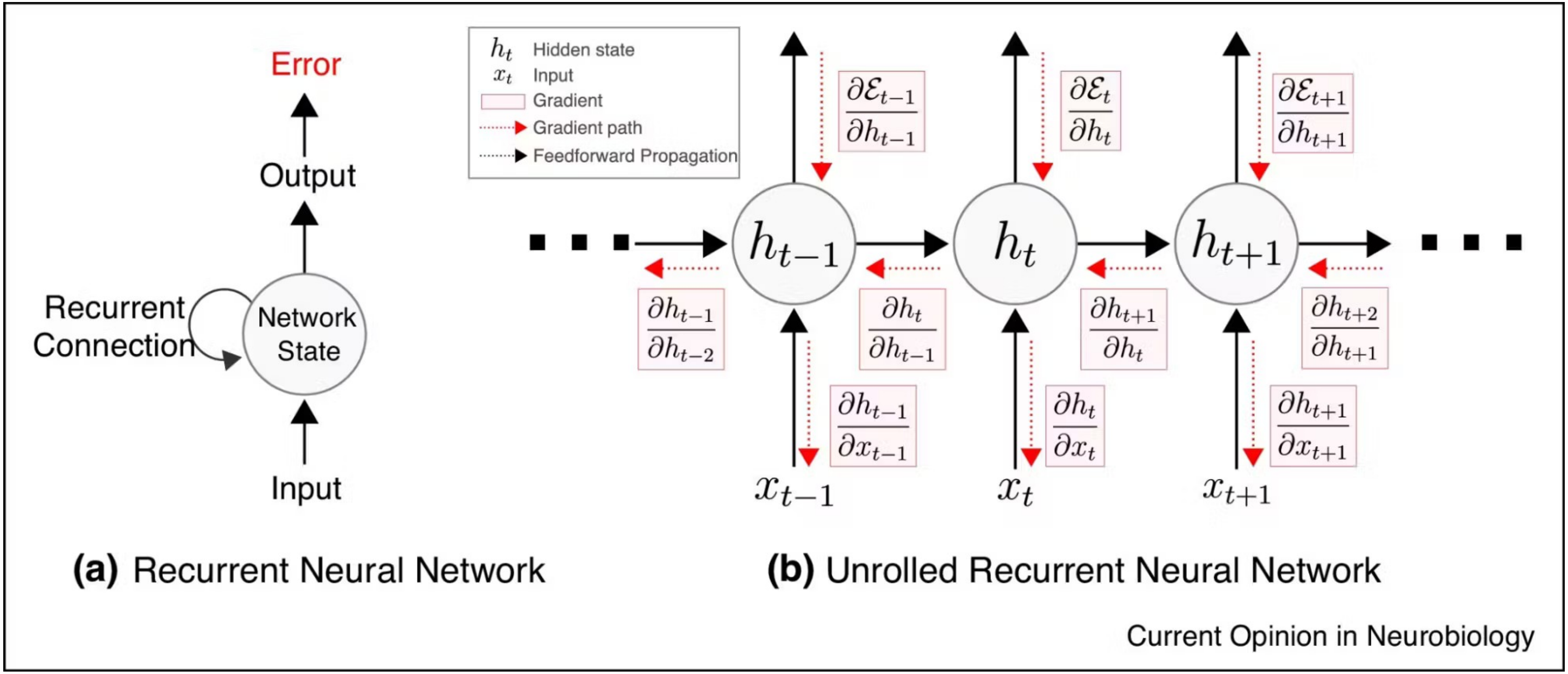
Several RNNs may learn different patterns in parallel

Same idea as different kernels in CNNs



No connexions between parallel cells of the same layer!

Training RNNs by backpropagation in time


Error
Output
Recurrent Connection
Network State
Input
h_t Hidden state
x_t Input
Gradient
Gradient path
Feedforward Propagation
h_{t-1}
h_t
h_{t+1}
$\frac{\partial \mathcal{E}_{t-1}}{\partial h_{t-1}}$
$\frac{\partial \mathcal{E}_t}{\partial h_t}$
$\frac{\partial \mathcal{E}_{t+1}}{\partial h_{t+1}}$
$\frac{\partial h_{t-1}}{\partial h_{t-2}}$
$\frac{\partial h_t}{\partial h_{t-1}}$
$\frac{\partial h_{t+1}}{\partial h_t}$
$\frac{\partial h_{t+2}}{\partial h_{t+1}}$
$\frac{\partial h_{t-1}}{\partial x_{t-1}}$
$\frac{\partial h_t}{\partial x_t}$
$\frac{\partial h_{t+1}}{\partial x_{t+1}}$
x_{t-1}
x_t
x_{t+1}

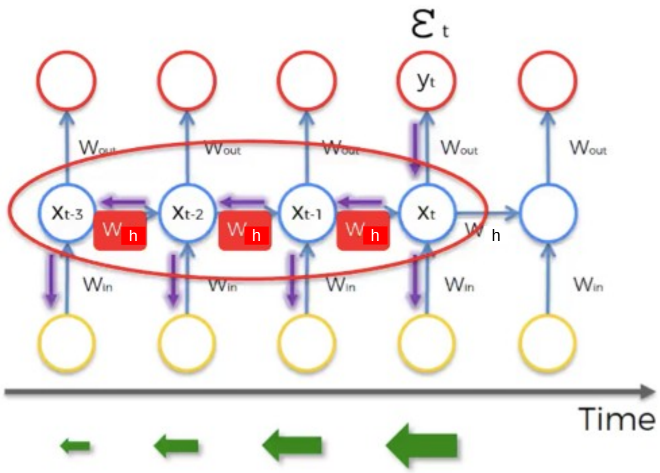
(a) Recurrent Neural Network

(b) Unrolled Recurrent Neural Network

Current Opinion in Neurobiology

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

Exploding and vanishing gradients



Source: SuperDataScience

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

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

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

W_h ~ small ⇨ Vanishing

W_h ~ large ⇨ Exploding

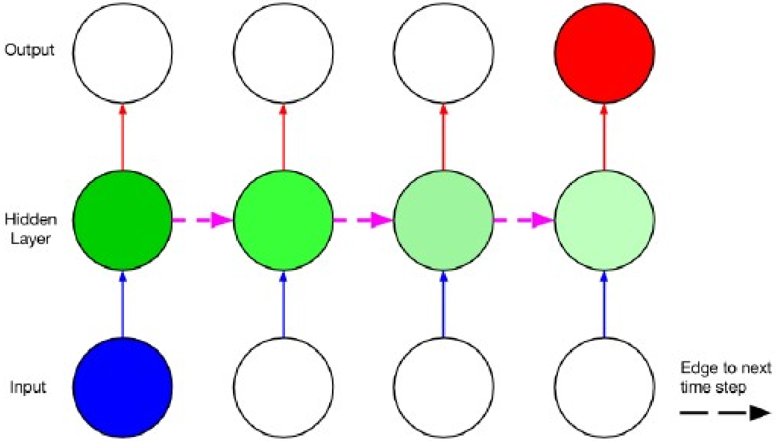
E.g.: activation function = ReLU

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

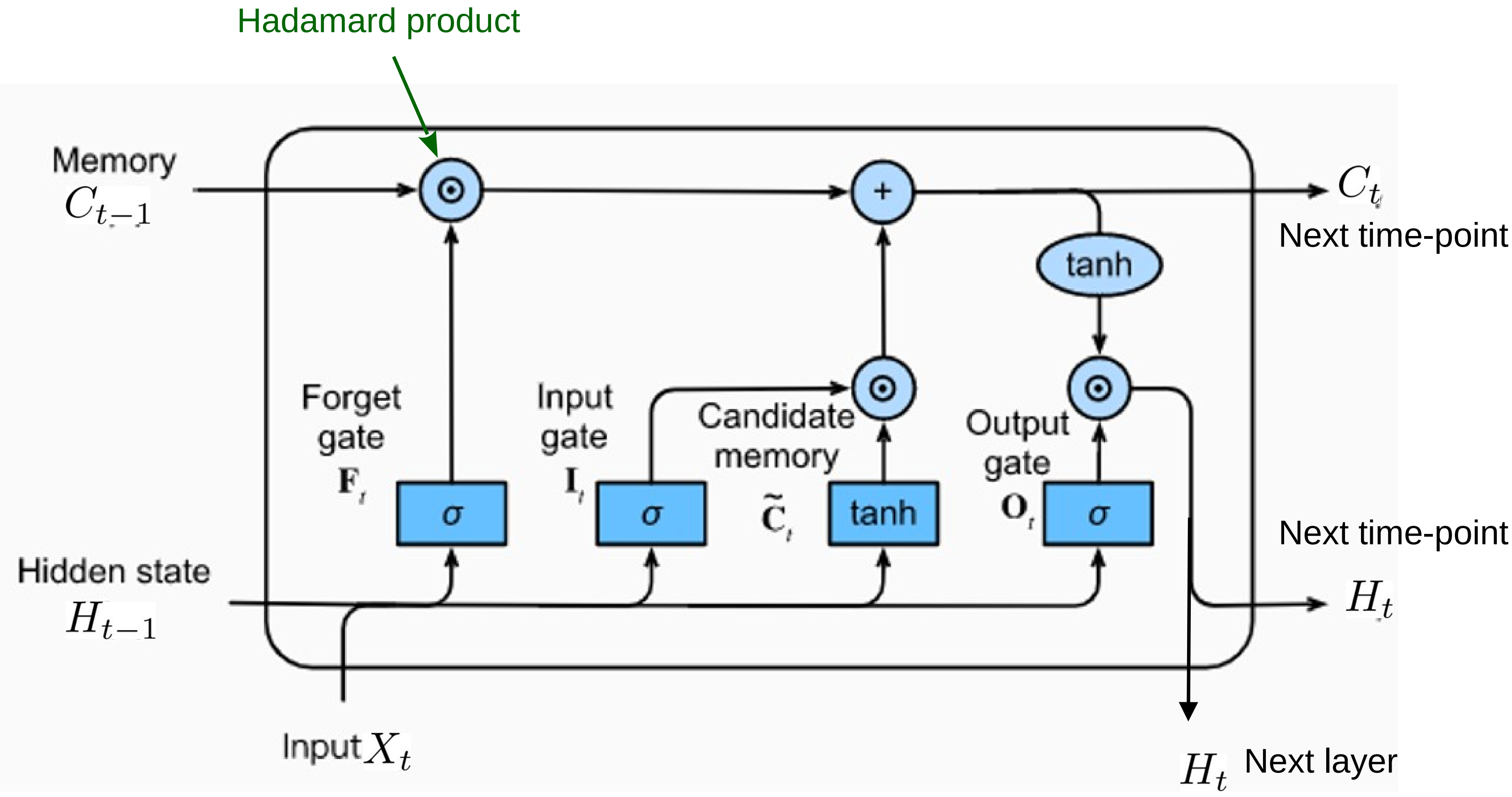
$$w_h = 0.1; \frac{\partial x_{10}}{\partial x_1} = w_h^{10} = 0.0000000001$$

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

Green = sensitivity of output on input



Solution: Long Short-Term Memory (LSTM)

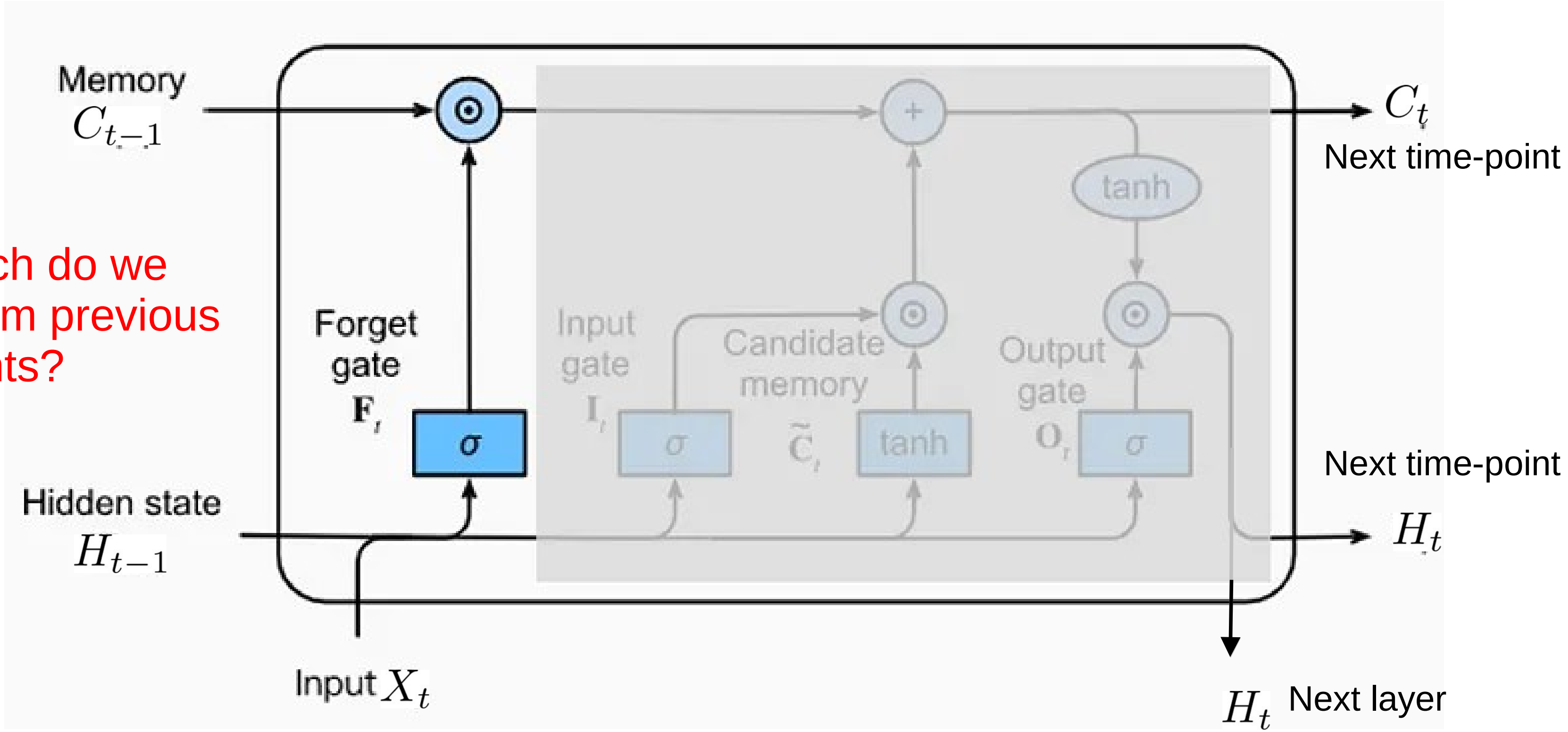


Hochreiter and Schmidhuber (1997) Long short-term memory. *Neur Comput*, 9(8):1735-1780

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

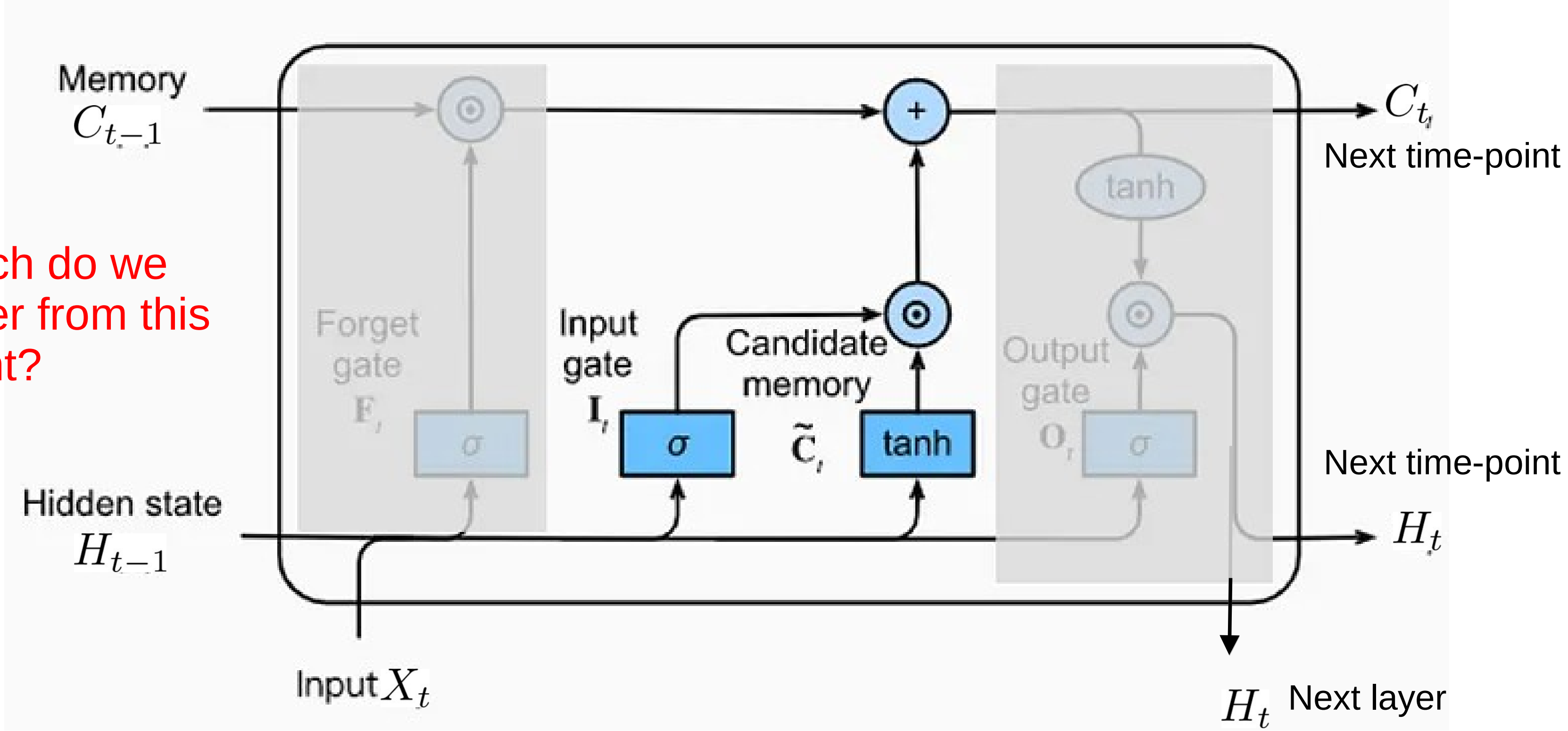
Solution: Long Short-Term Memory (LSTM)

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



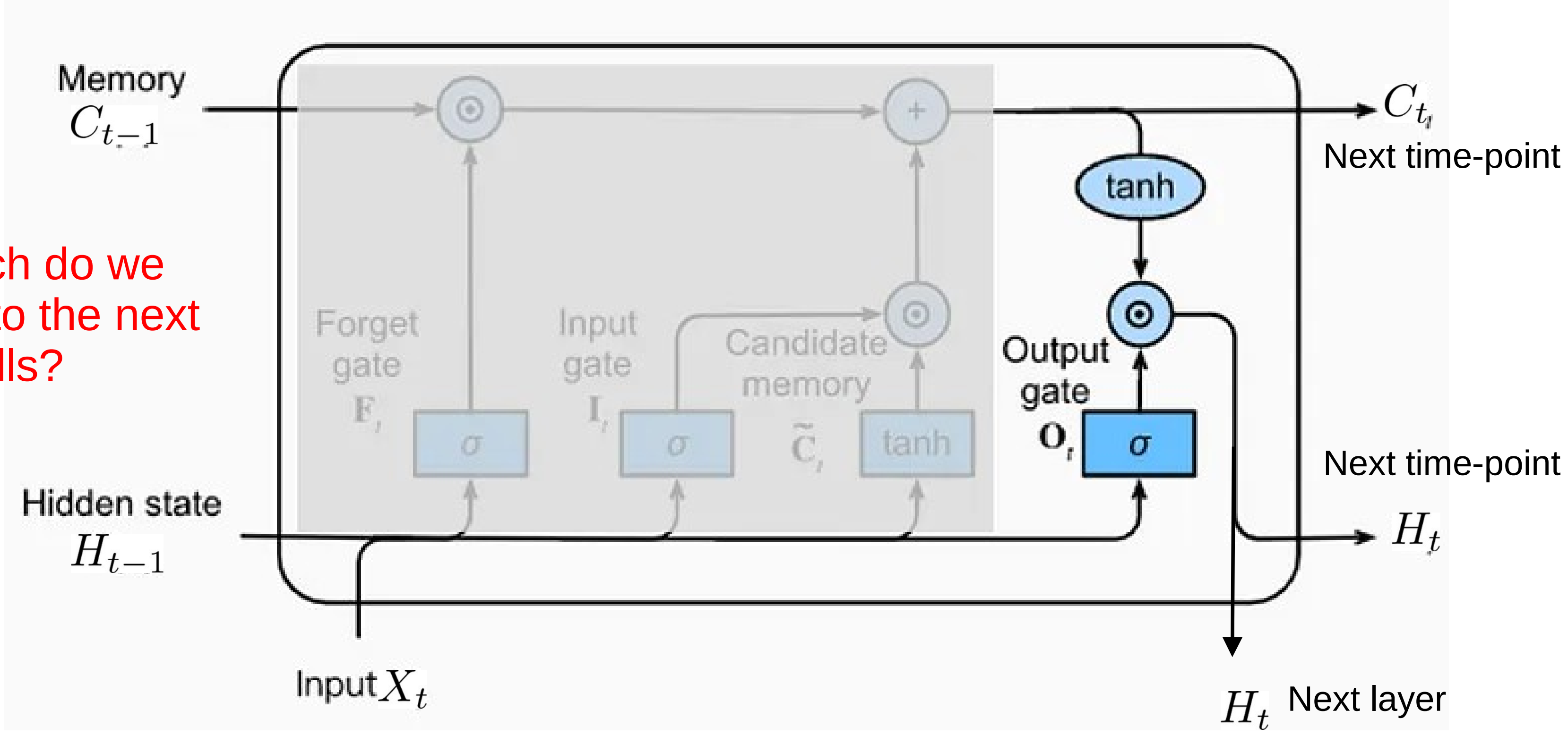
Solution: Long Short-Term Memory (LSTM)

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



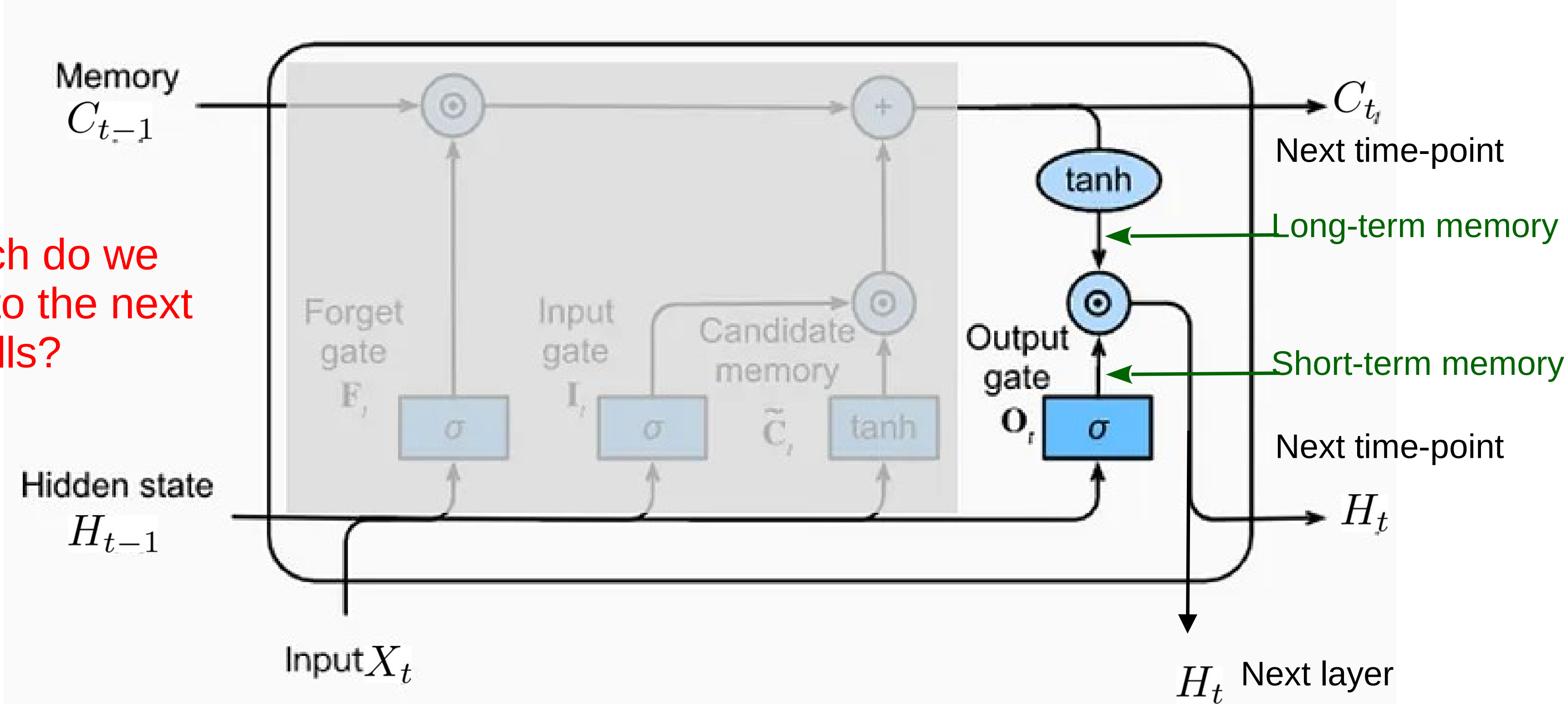
Solution: Long Short-Term Memory (LSTM)

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

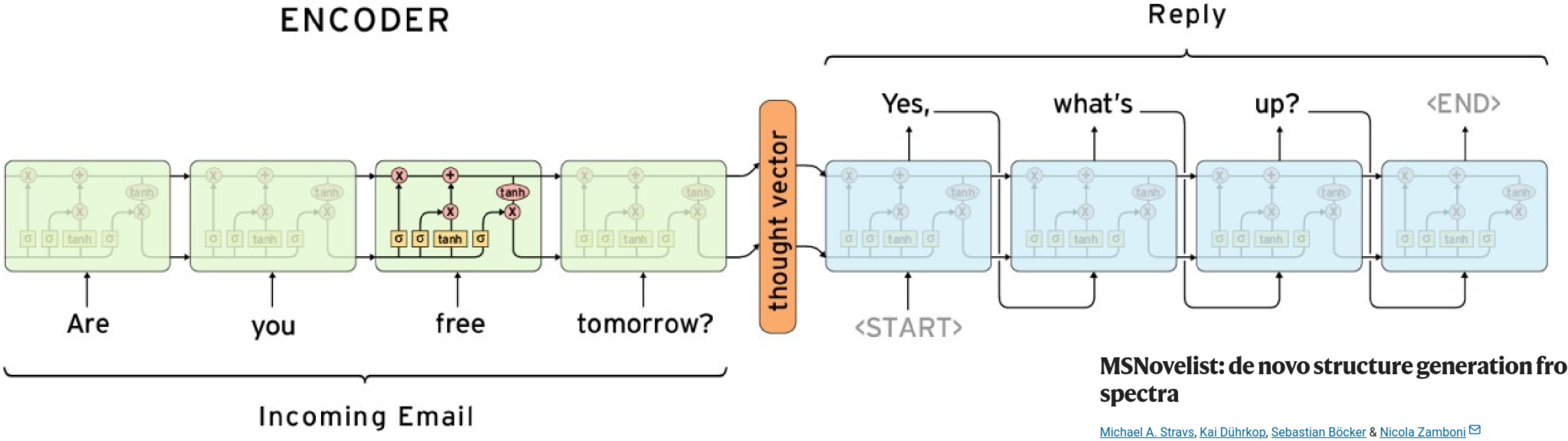


Solution: Long Short-Term Memory (LSTM)

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



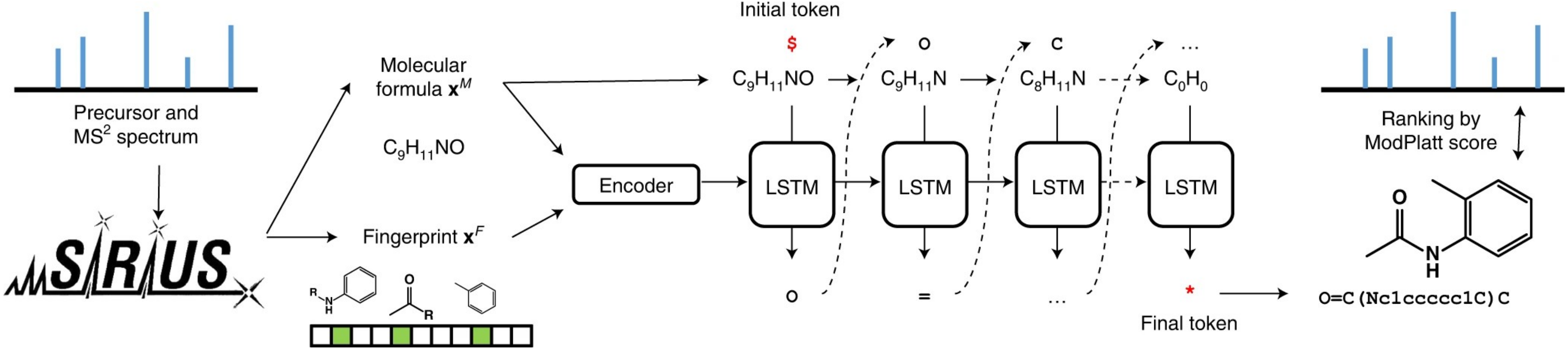
LSTMs for Encoder-Decoder



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



Examples in the biomedical domain

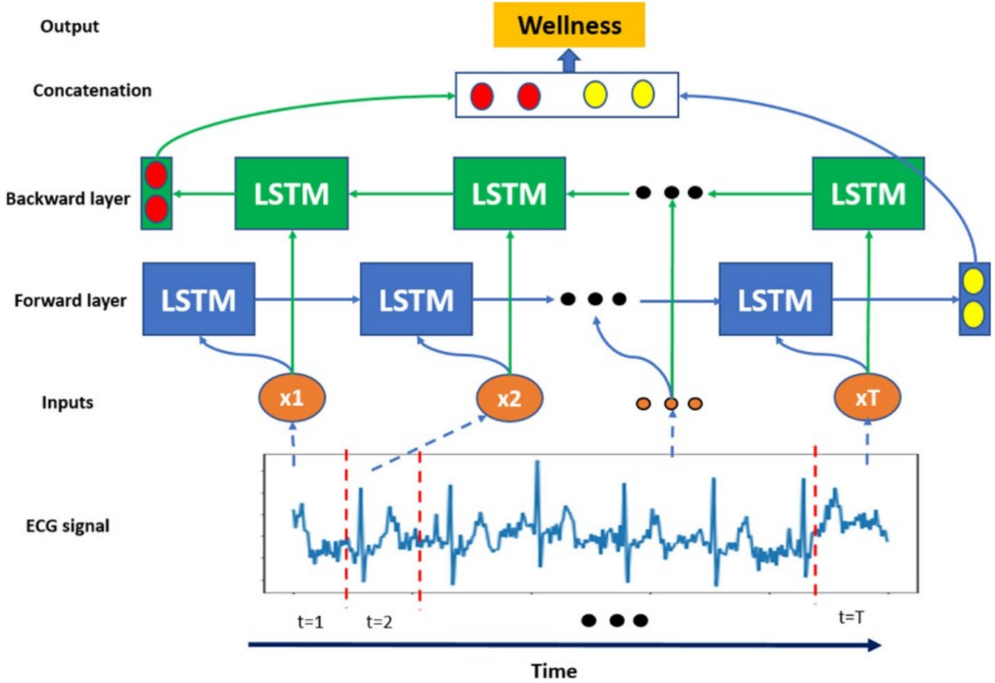
Fan *et al. BMC Medical Informatics and Decision Making* (2019) 19:285
https://doi.org/10.1186/s12911-019-1012-8

BMC Medical Informatics and Decision Making

RESEARCH ARTICLE **Open Access**

Forecasting one-day-forward wellness conditions for community-dwelling elderly with single lead short electrocardiogram signals

Xiaomao Fan[1], Yang Zhao[2*], Hailiang Wang[2] and Kwok Leung Tsui[1,2]



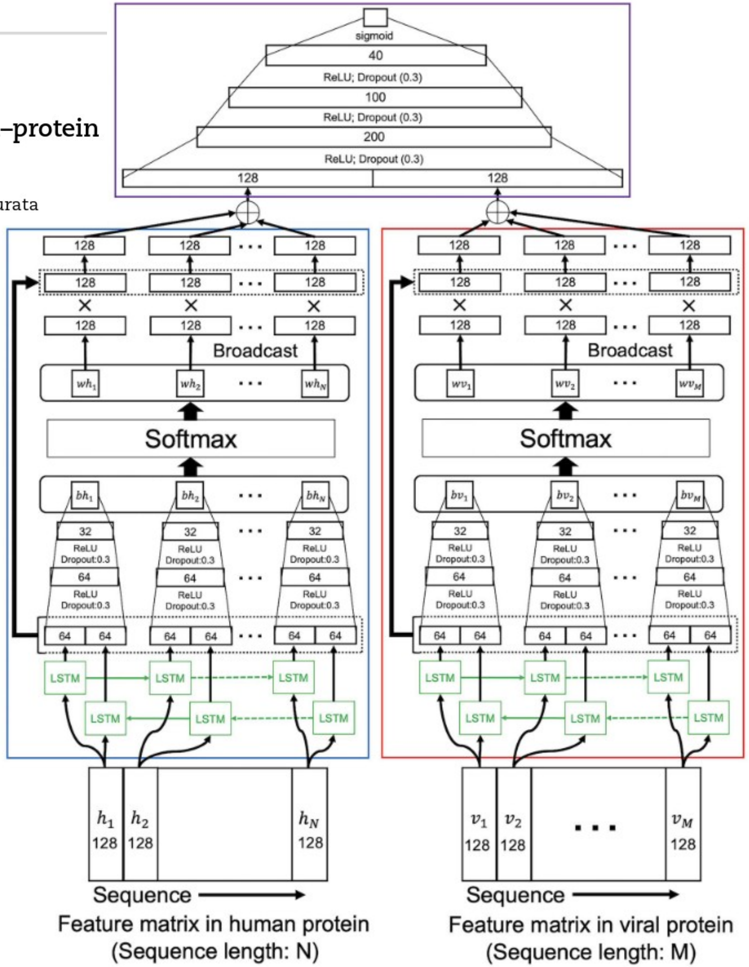
OXFORD

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



Examples in the biomedical domain

Contents lists available at ScienceDirect

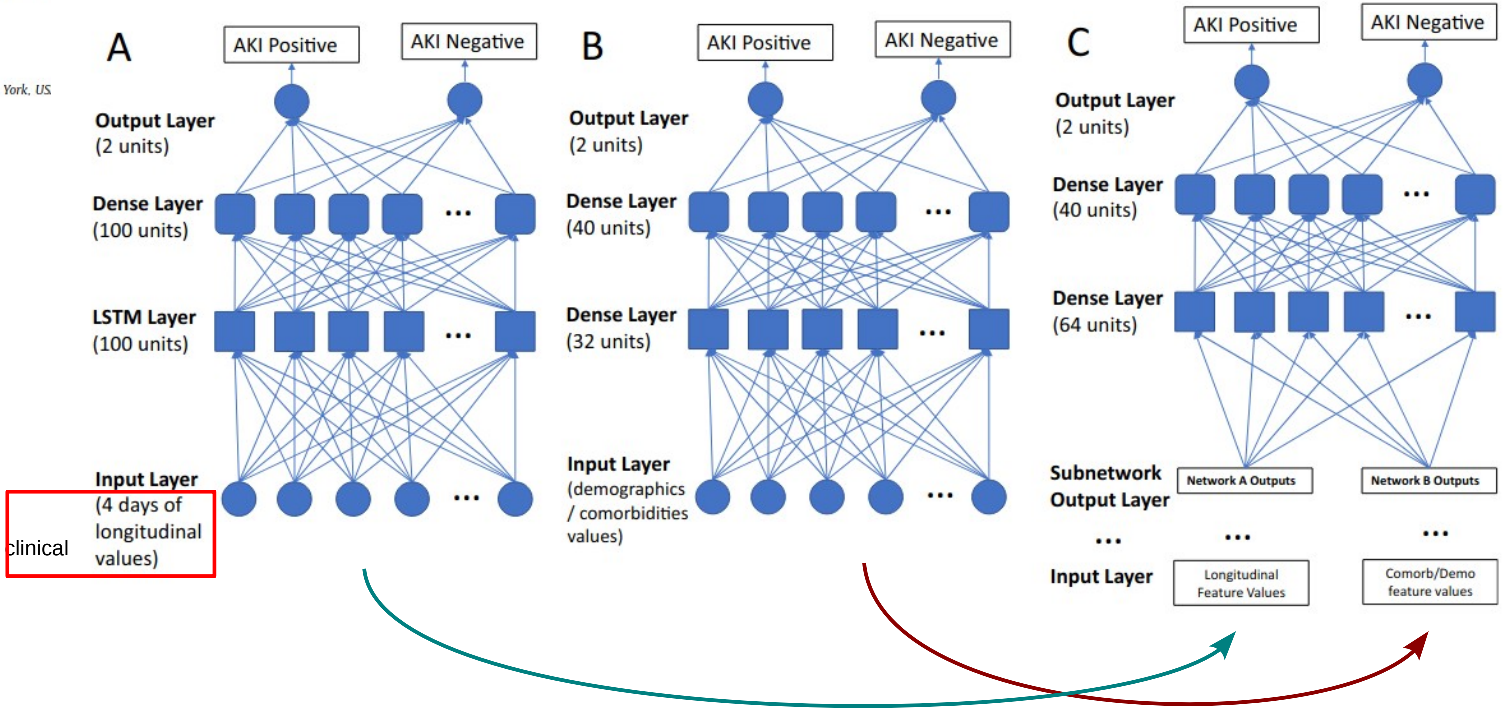
International Journal of Infectious Diseases

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

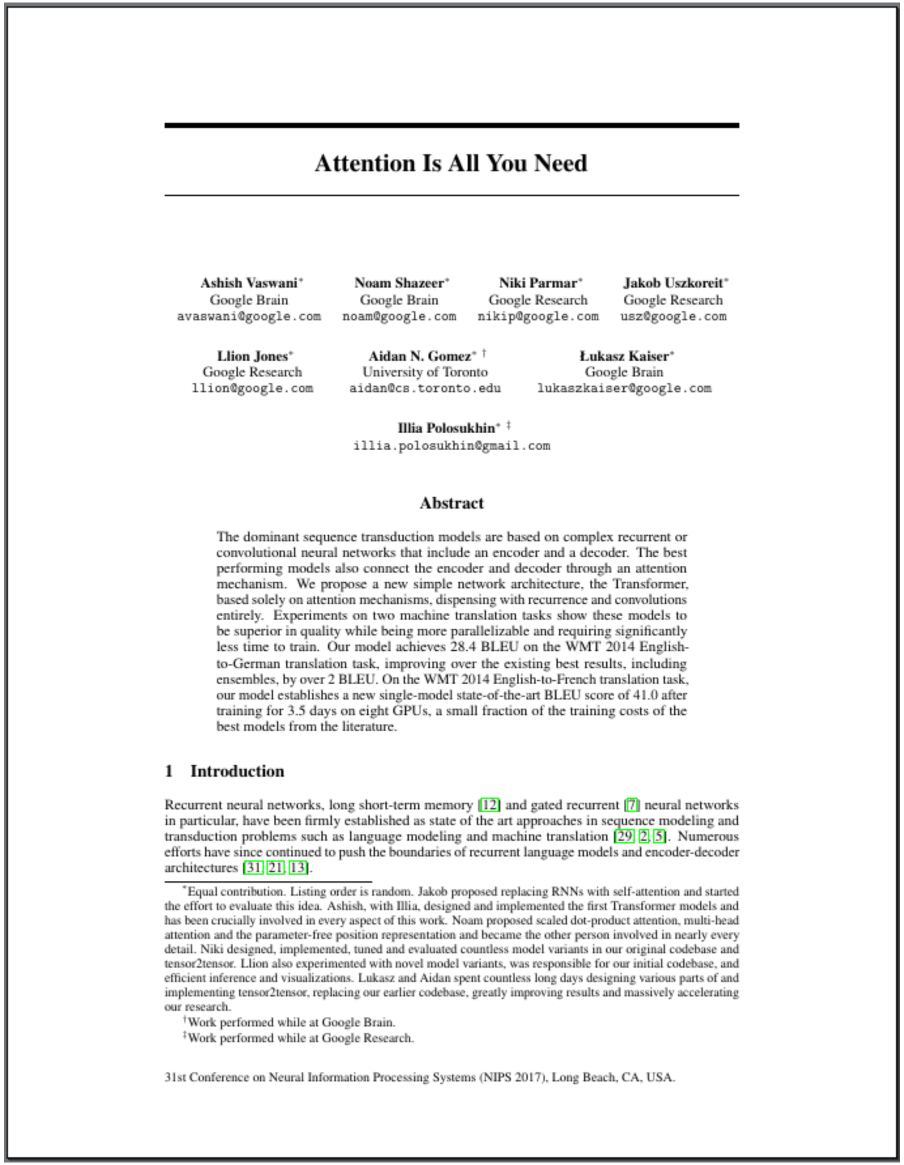
Long-short-term memory machine learning of longitudinal clinical data accurately predicts acute kidney injury onset in COVID-19: a two-center study

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

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



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].

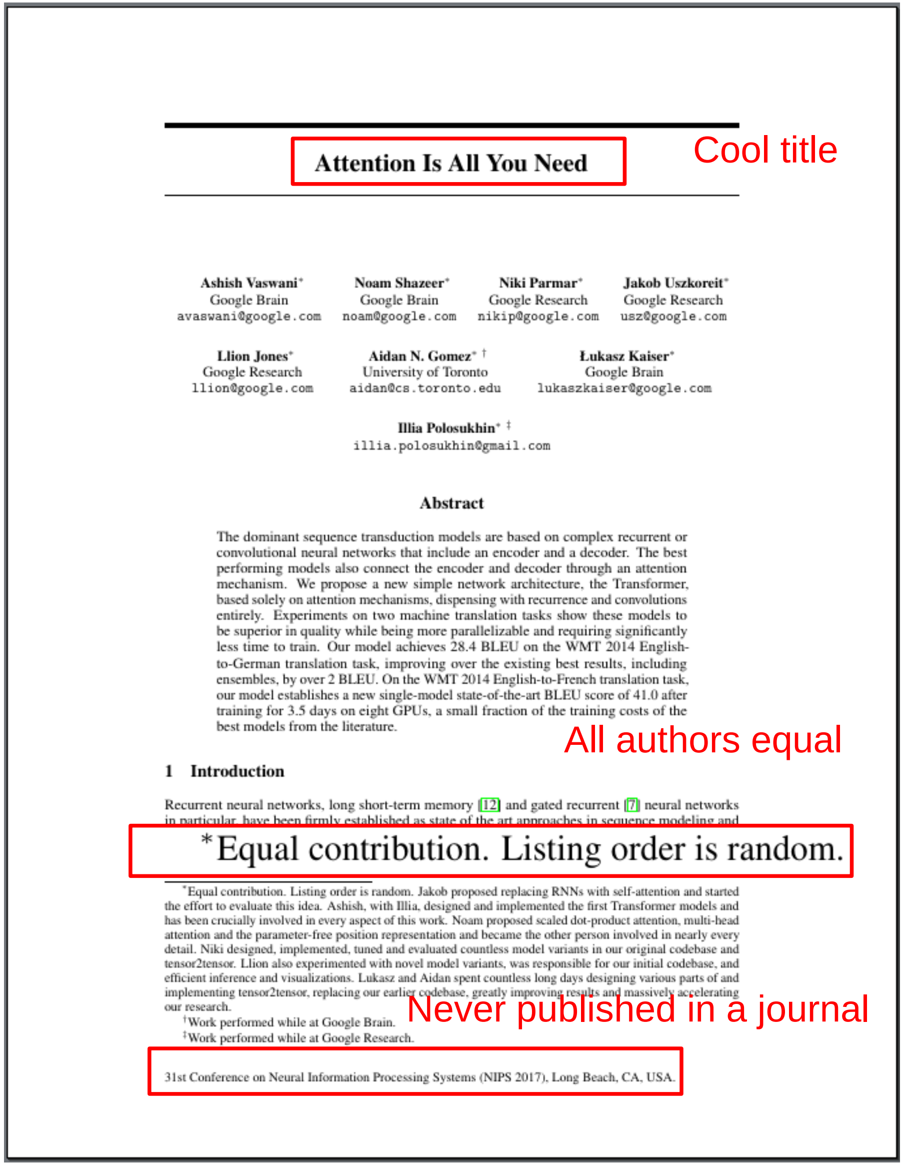
[*] Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled 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

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

[*] 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.

Cool title

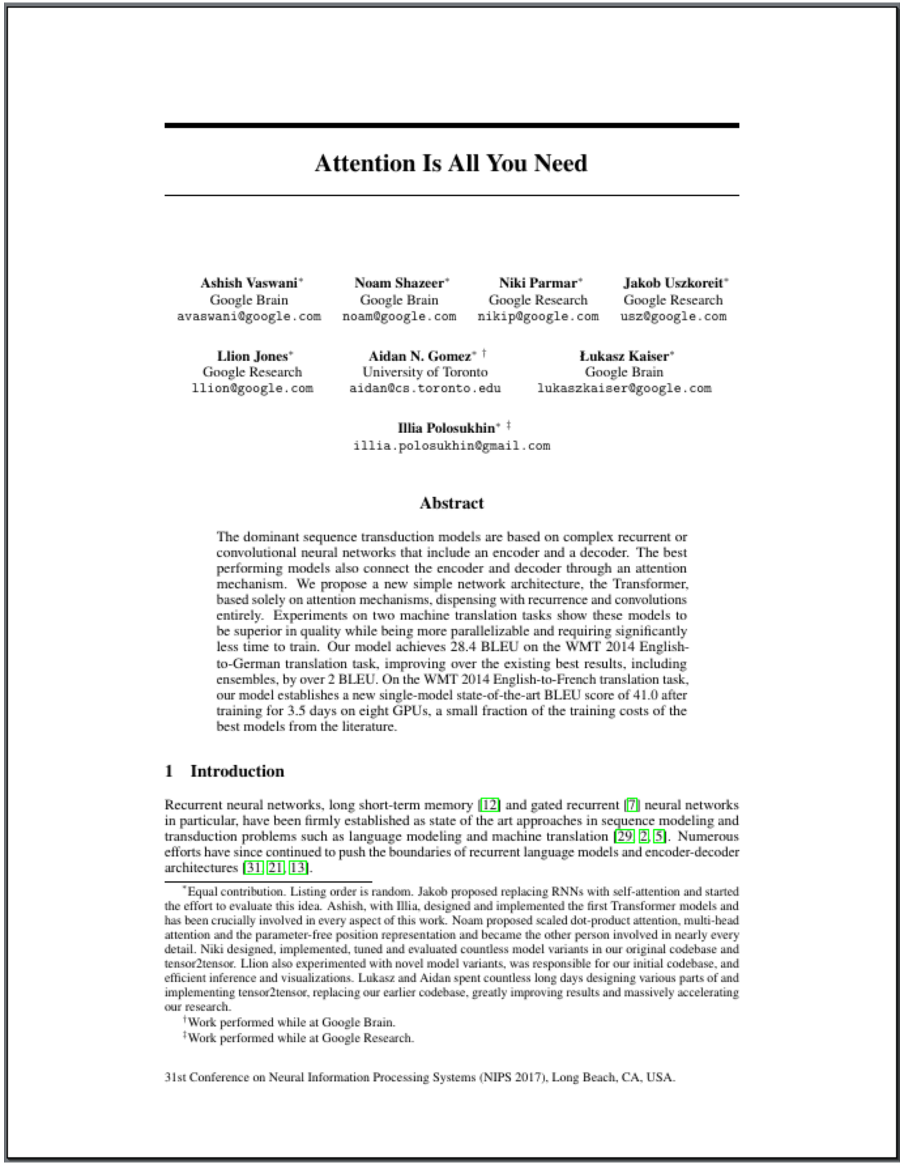
All authors equal

[*] Equal contribution. Listing order is random.

Never published in a journal

Cited... 198836 times as of 14 October 2025!

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.

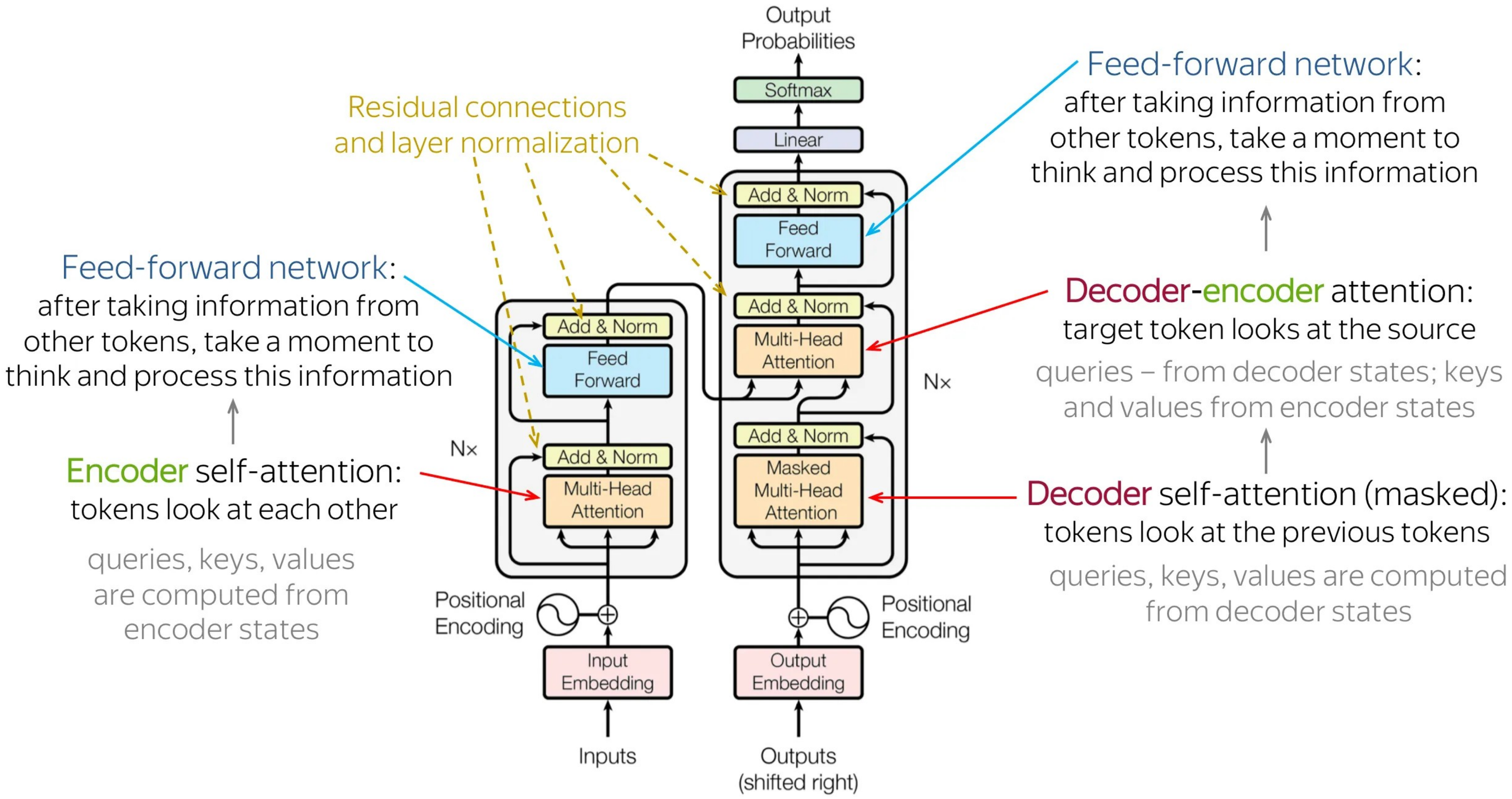
[‡] Work performed while at Google Research.

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

Claude

AlphaFold

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

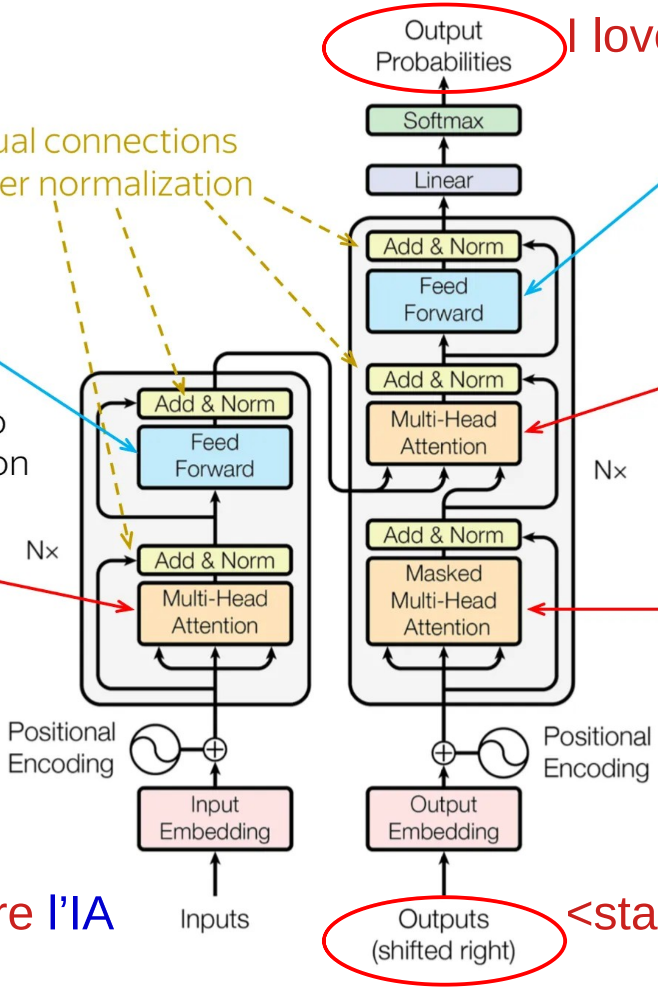
Residual connections and layer normalization

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

J'adore l'IA



I love AI

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

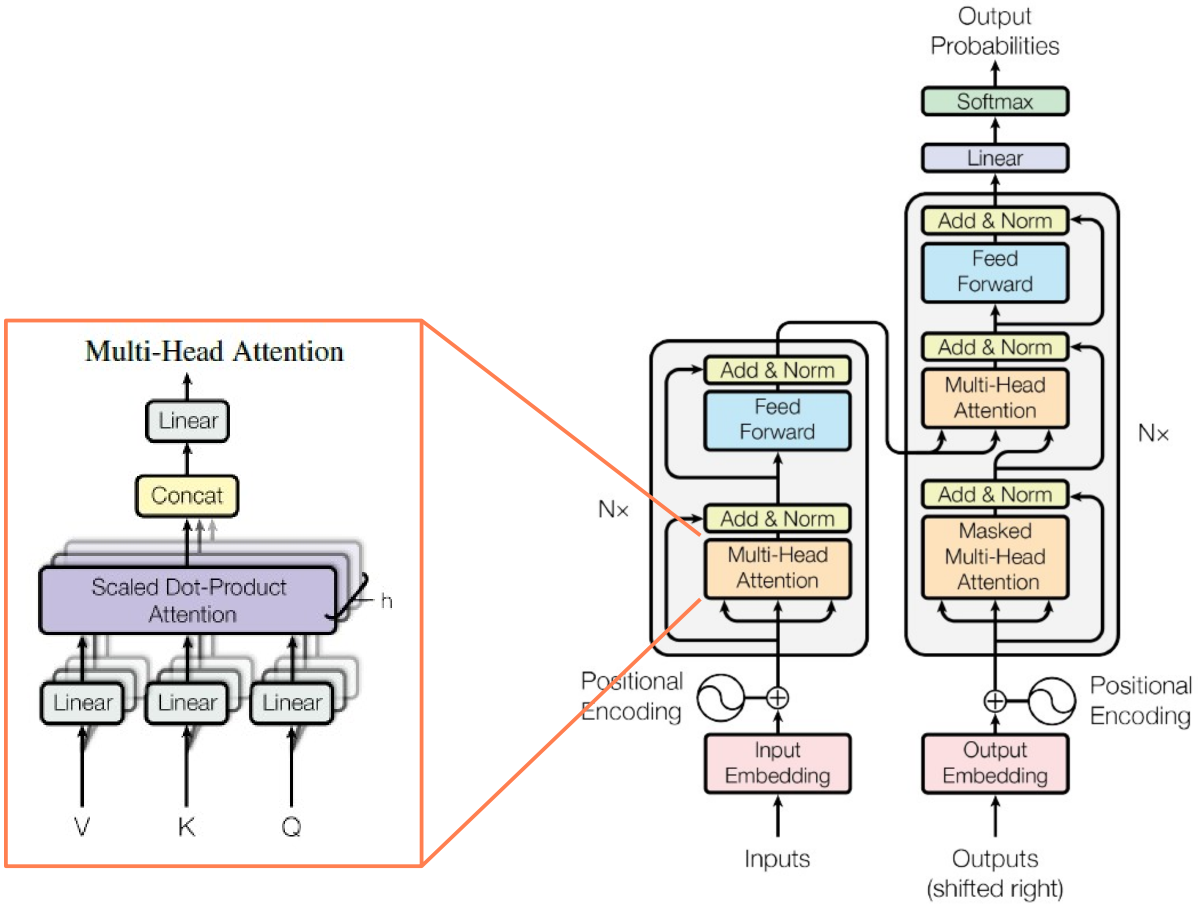
queries – from decoder states; keys and values from encoder states

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

queries, keys, values are computed from decoder states

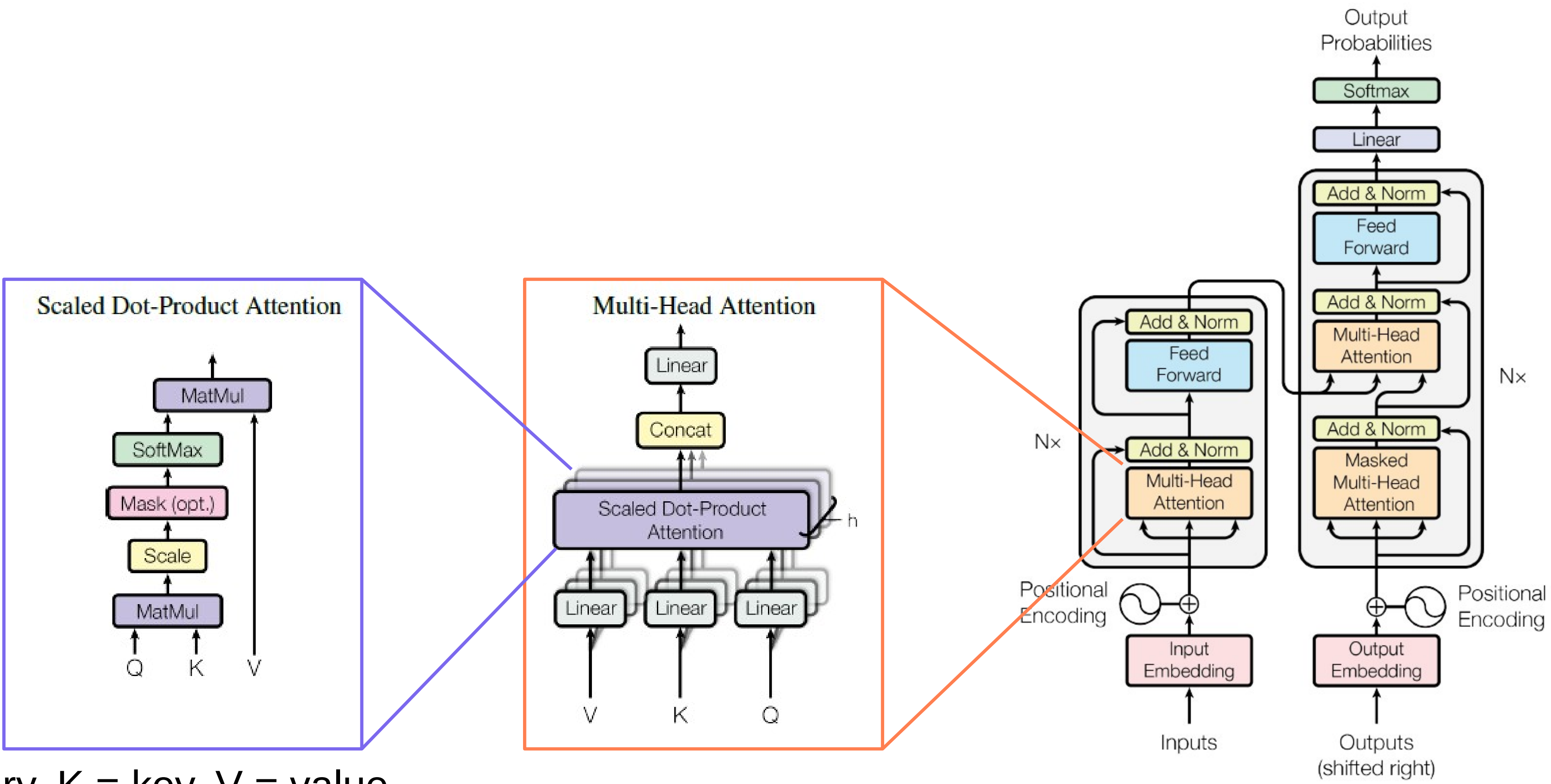
<start> I love

Attention in the Transformer



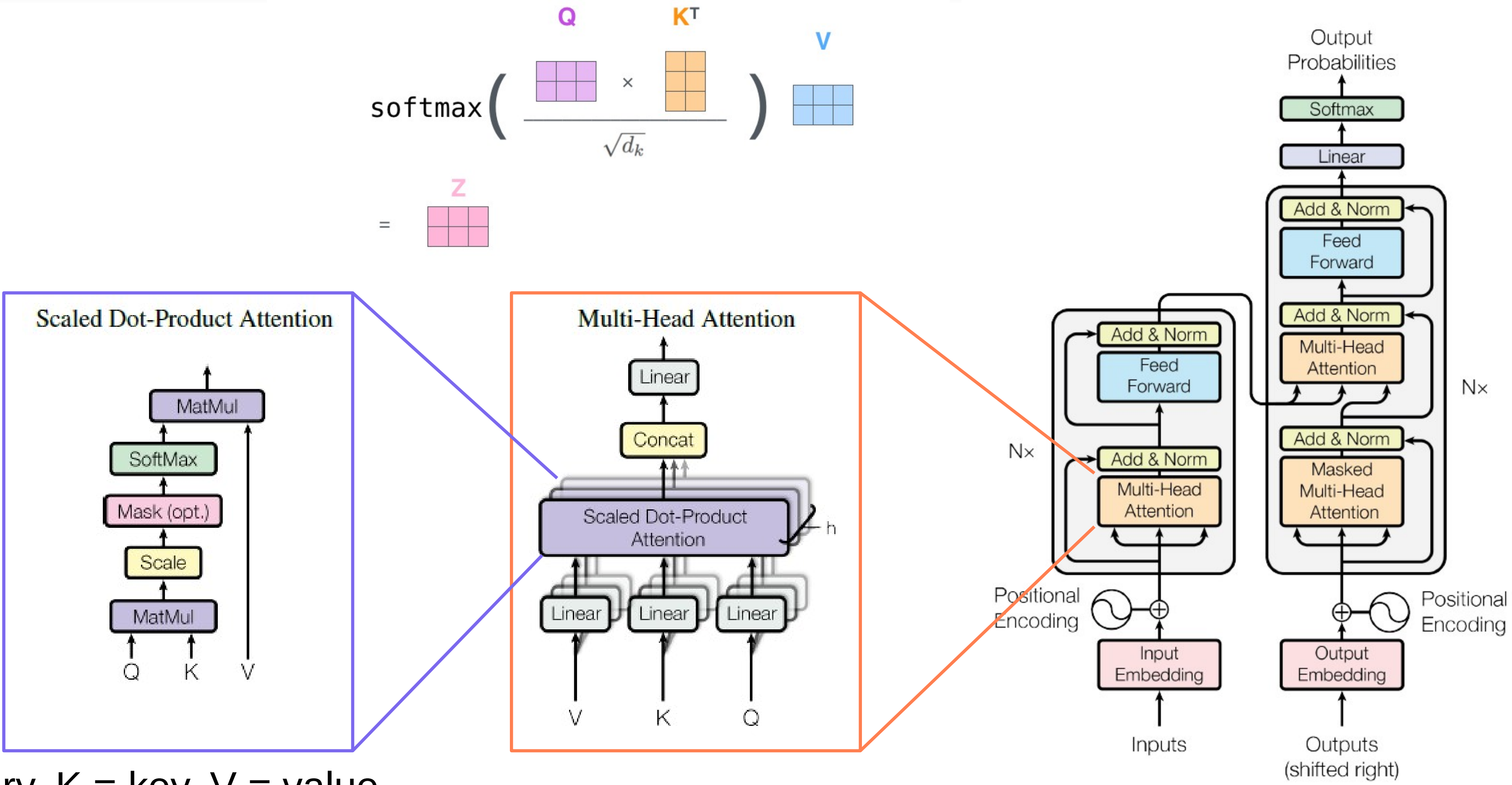
Q = query, K = key, V = value

Attention in the Transformer



Q = query, K = key, V = value

Attention in the Transformer

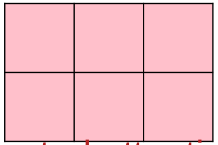


Q = query, K = key, V = value

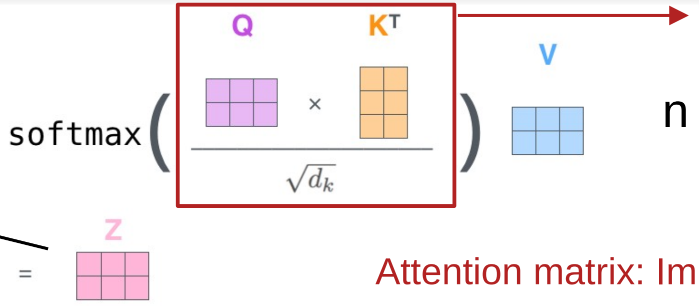
Attention in the Transformer

n = #tokens

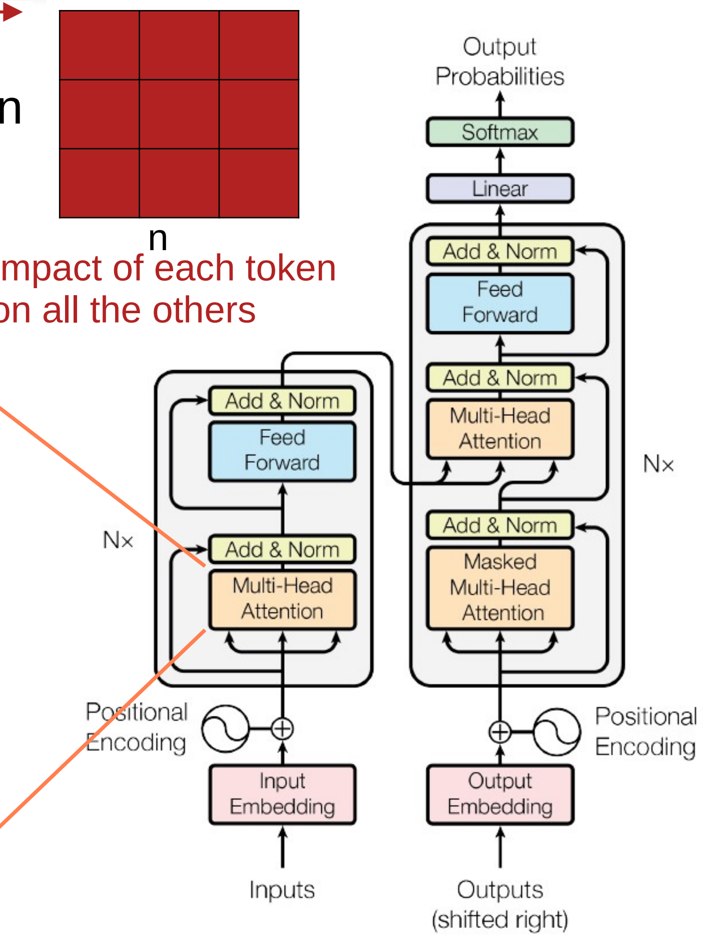
d = embedding dimension



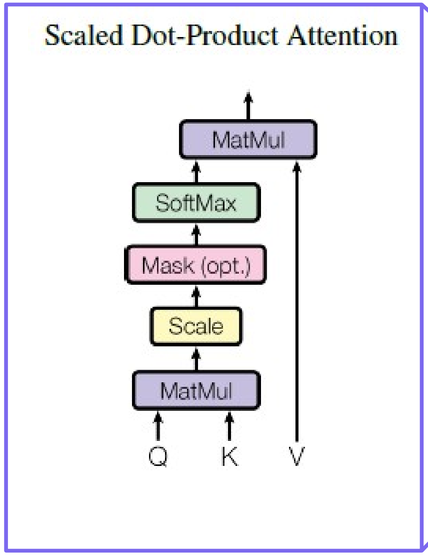
Attention: aggregated attention

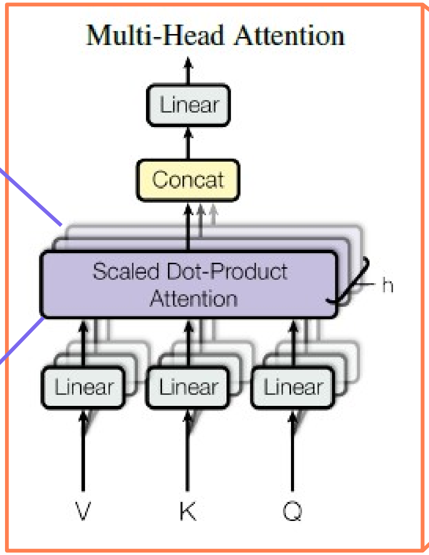


$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$



Attention matrix: Impact of each token on all the others

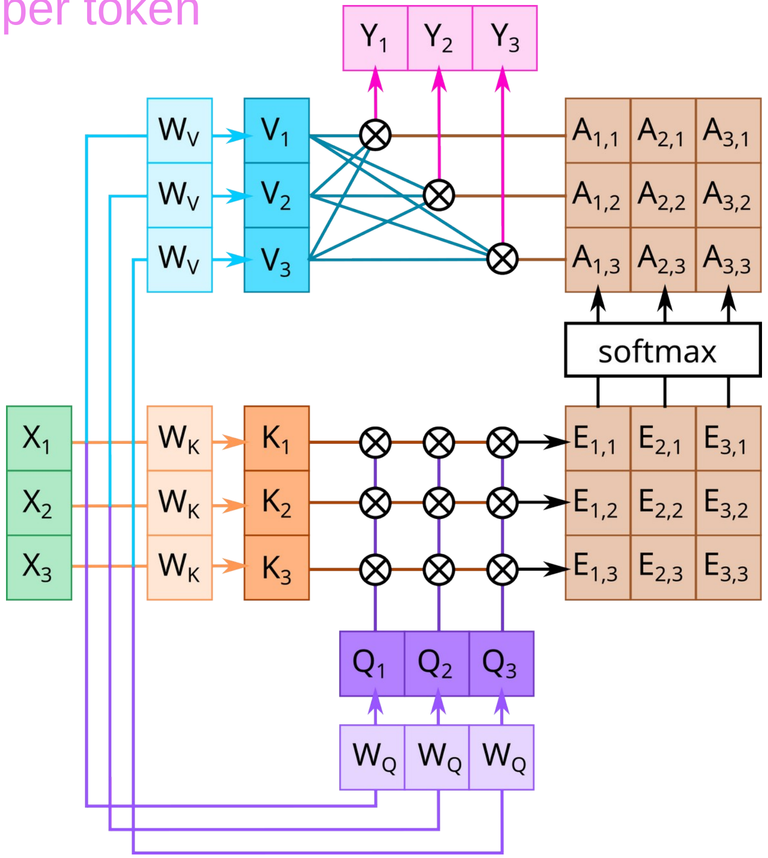




Q = query, K = key, V = value

Attention in the Transformer

Actual relevant attentions
1 per token

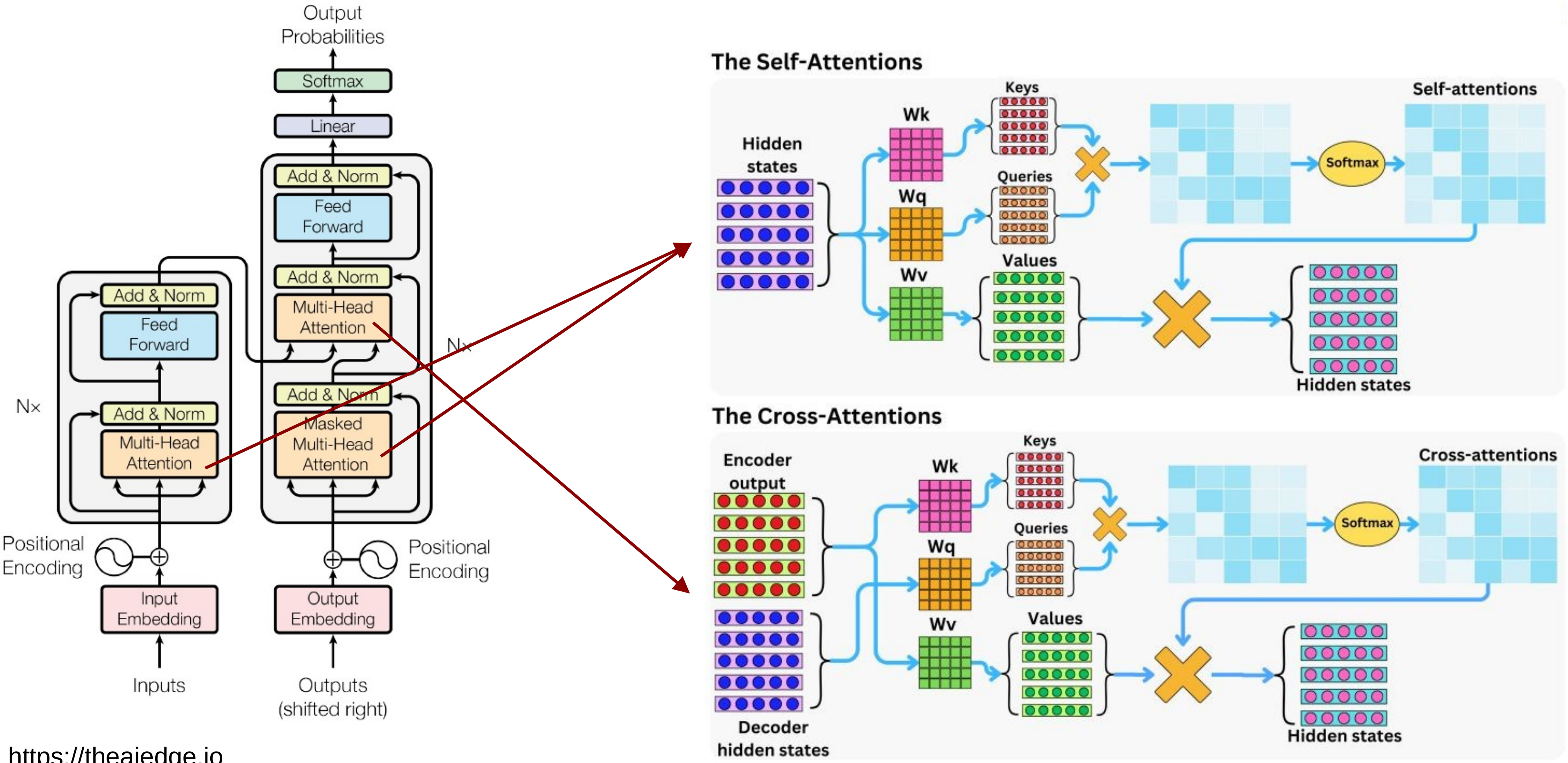


Attentions of all tokens on all tokens

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

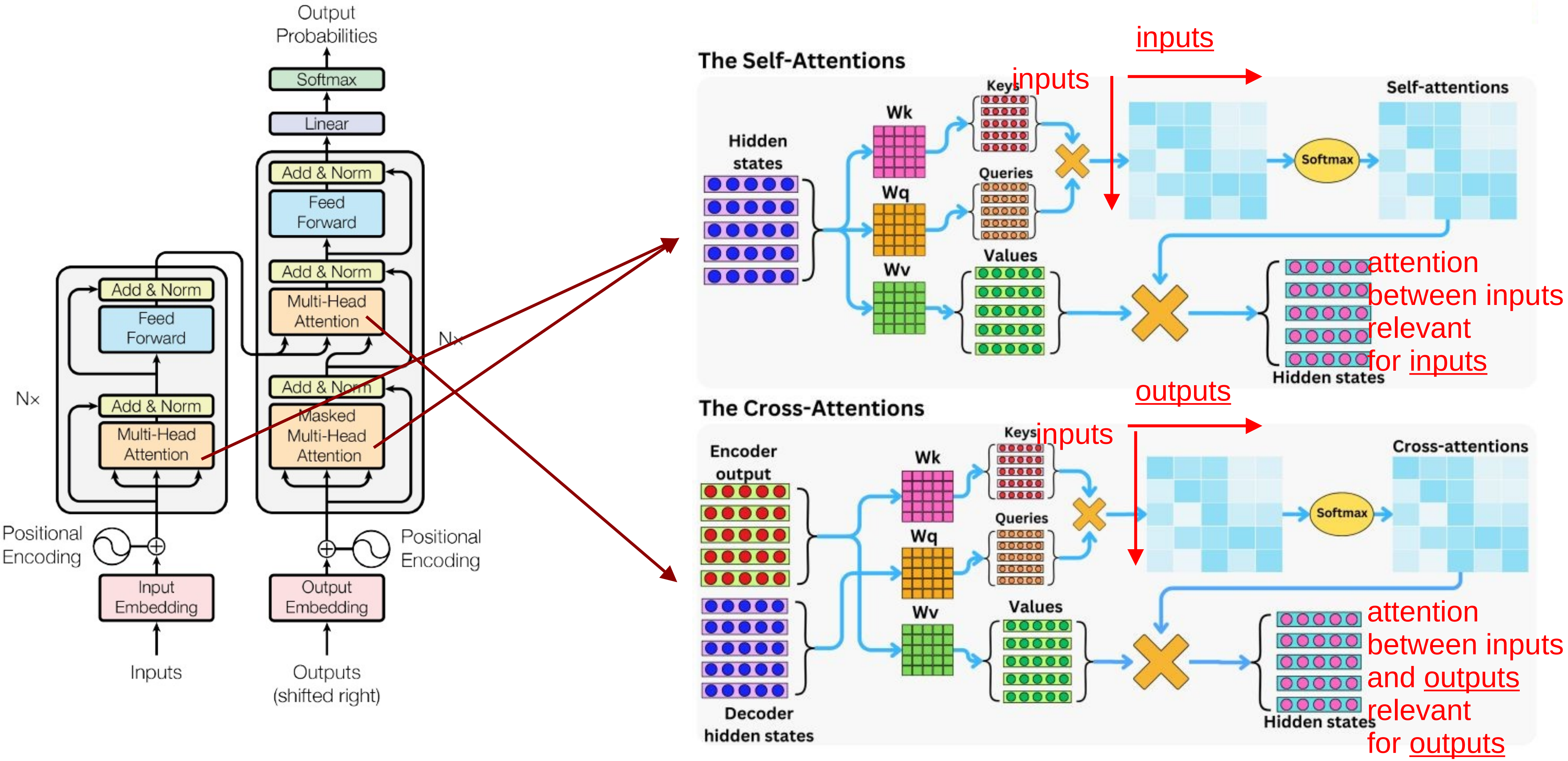
The dot product between Q and K^T compute how aligned are the vectors encoding two tokens (~cosine similarity)

Self versus cross-attention



ource: https://theaiedge.io

Self versus cross-attention



inputs

inputs

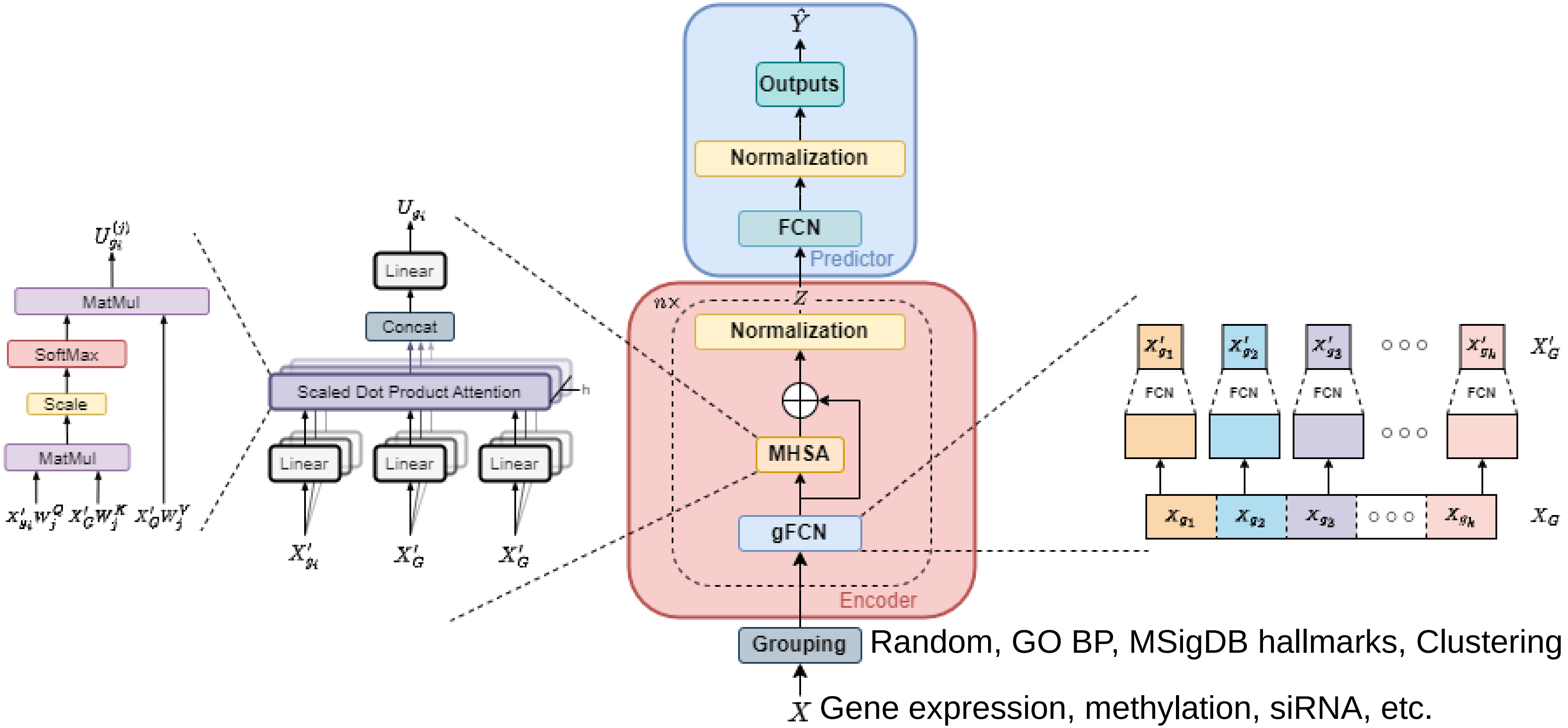
attention between inputs relevant for inputs

outputs

inputs

attention between inputs and outputs relevant for outputs

AttOmics: Omics values as tokens

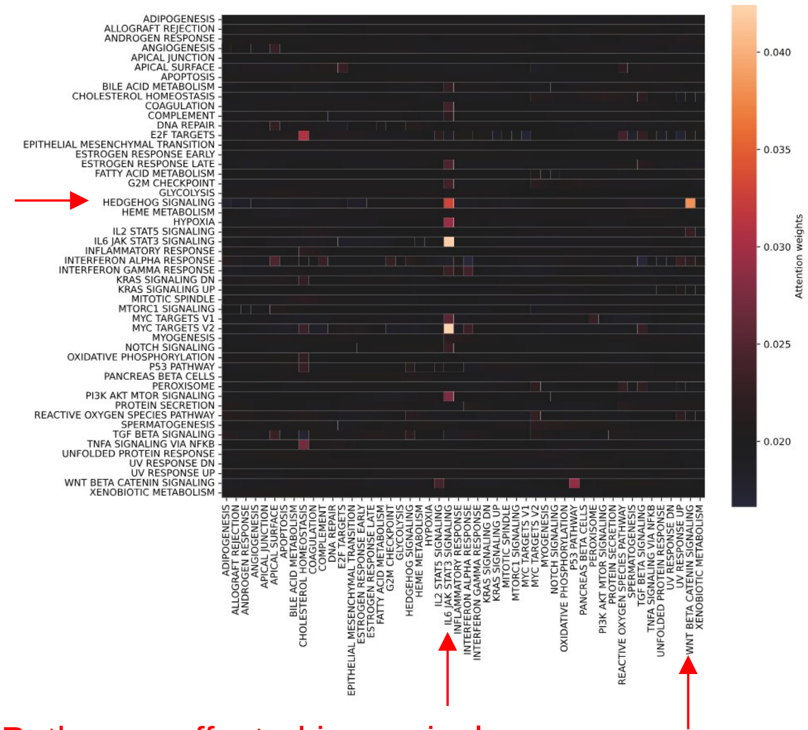


Random, GO BP, MSigDB hallmarks, Clustering

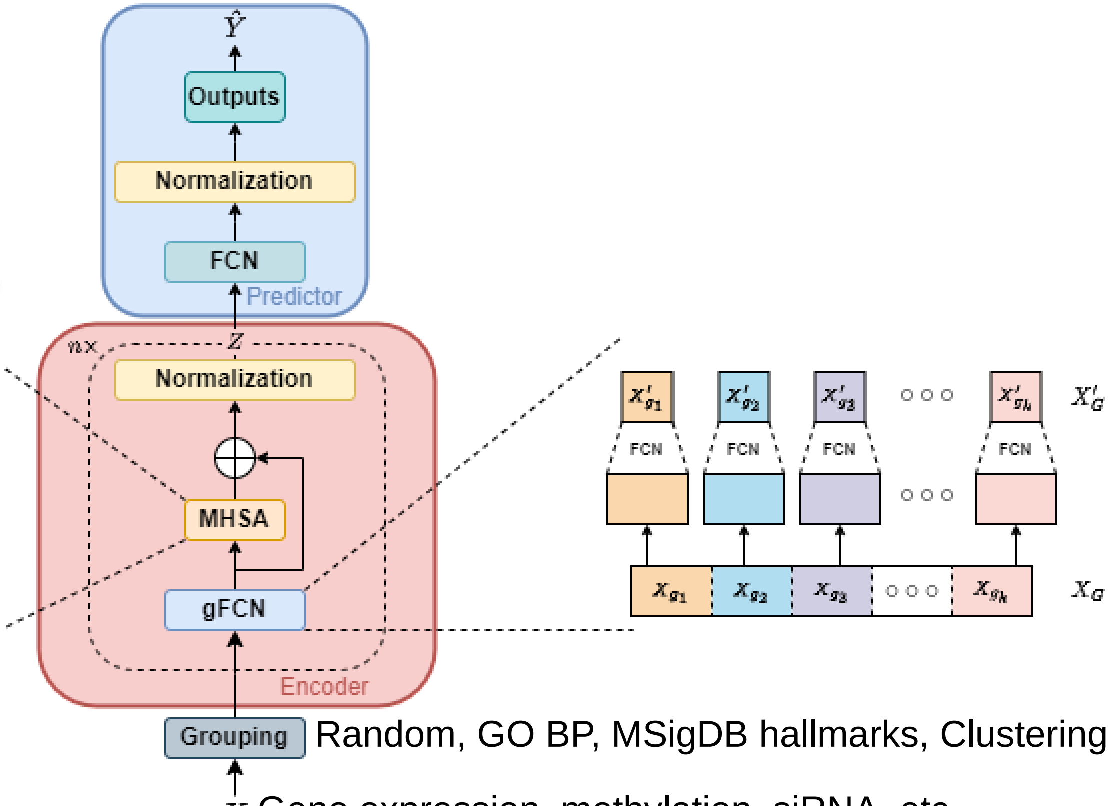
Gene expression, methylation, siRNA, etc.
e.g., TCGA cervical cancers

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: Omics values as tokens



Pathways affected in cervical cancer



Random, GO BP, MSigDB hallmarks, Clustering

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

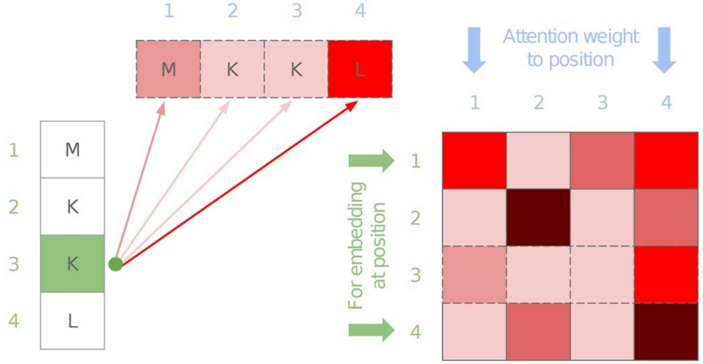
Beaude, A., Rafiee Vahid, M., Augé, F., Zehraoui, F., & Hanczar, B. (2023). AttOmics: attention-based architecture for diagnosis and prognosis from omics data. *Bioinformatics*, 39(Supplement_1), i94-i102.

EnzBERT: amino-acids as tokens

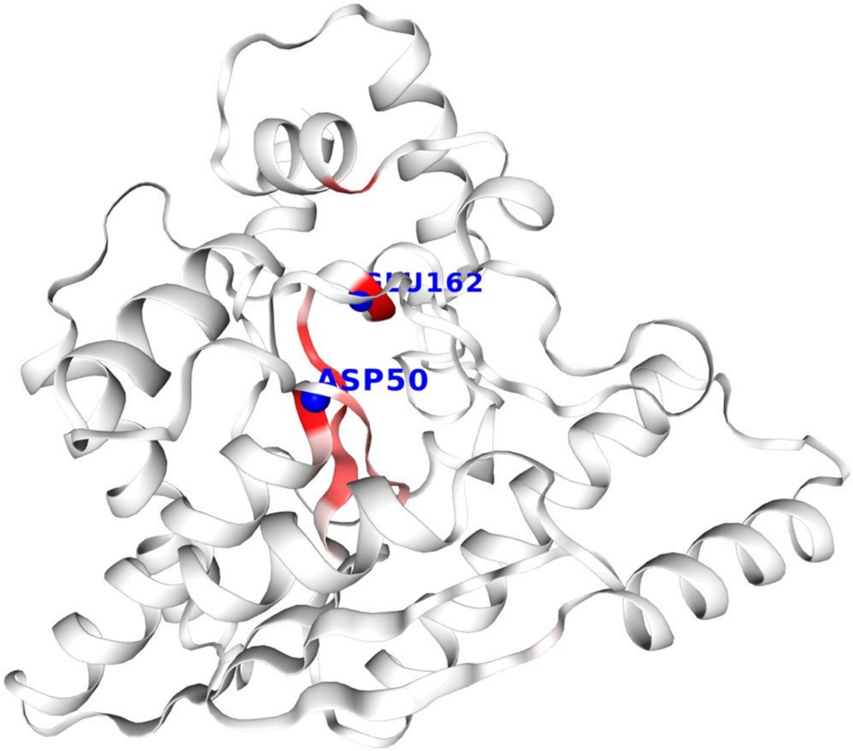
Predicting enzymatic function of protein sequences with attention

Nicolas Buton, François Coste, Yann Le Cunff

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



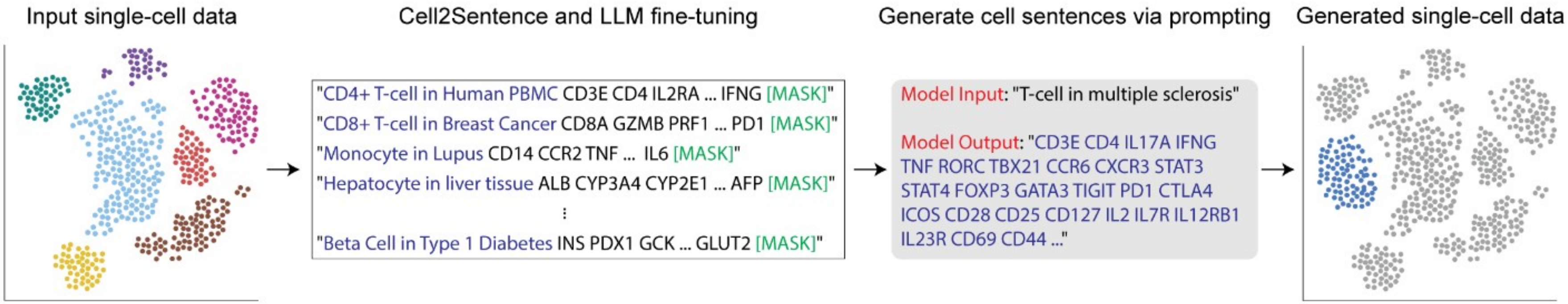
Nh(3)-dependent nad(+) synthetase



```
0 MSMQEKIMRE LHVKPSIDPK QEIEDRVNFL KQYVKKTGAK GFVLGISGGQ DSTLAGRLAQ LAVESIREEG GDAQFIAVRL PHGTQQDEDD AQLALKFIKP
1 DKSWKFDIKS TVSAFSDQYQ QETGDQLTDF NKGNVKARTR MIAQYAIGGQ EGLLVLGTDH AAEAVTGFFT KYGDGGADLL PLTGLTKRQG RTLLKELGAP
2 ERLYLKEPTA DLLDEKPQQS DETELGISYD EIDDYLEGKE VSAKVSEALE KRYSMTEHKR QVPASMFDDW WK
```

Aggregated attention
for each token (amino acid)

Cell2Sentence: gene names as token

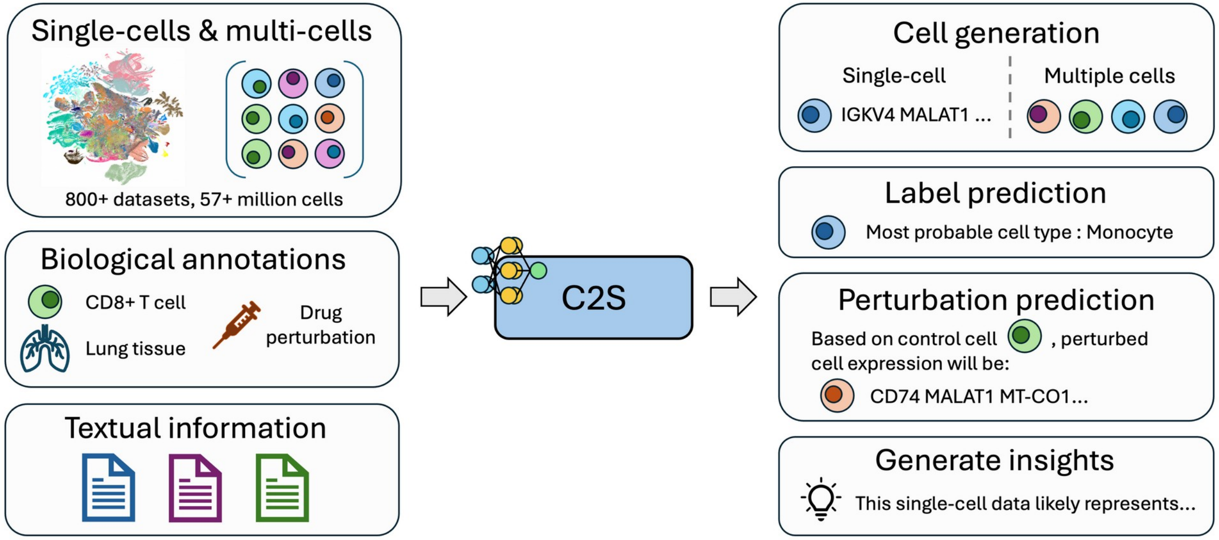

Input single-cell data
Cell2Sentence and LLM fine-tuning
Generate cell sentences via prompting
Generated single-cell data

"CD4+ T-cell in Human PBMC CD3E CD4 IL2RA ... IFNG [MASK]"
"CD8+ T-cell in Breast Cancer CD8A GZMB PRF1 ... PD1 [MASK]"
"Monocyte in Lupus CD14 CCR2 TNF ... IL6 [MASK]"
"Hepatocyte in liver tissue ALB CYP3A4 CYP2E1 ... AFP [MASK]"
⋮
"Beta Cell in Type 1 Diabetes INS PDX1 GCK ... GLUT2 [MASK]"

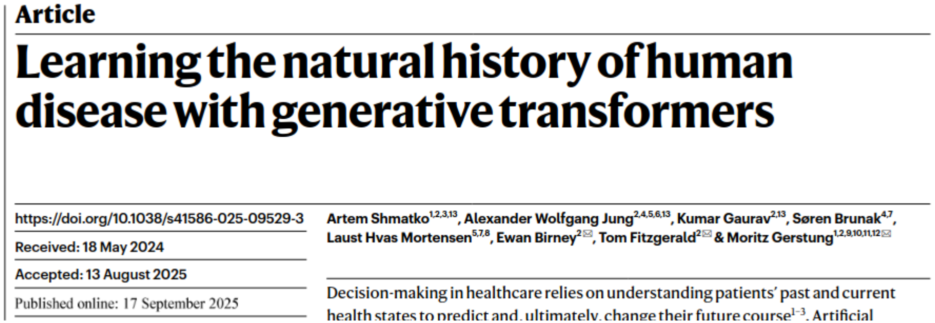
Model Input: "T-cell in multiple sclerosis"

Model Output: "CD3E CD4 IL17A IFNG TNF RORC TBX21 CCR6 CXCR3 STAT3 STAT4 FOXP3 GATA3 TIGIT PD1 CTLA4 ICOS CD28 CD25 CD127 IL2 IL7R IL12RB1 IL23R CD69 CD44 ..."

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



Delphi-2M: Life events as tokens



Article

Learning the natural history of human disease with generative transformers

https://doi.org/10.1038/s41586-025-09529-3

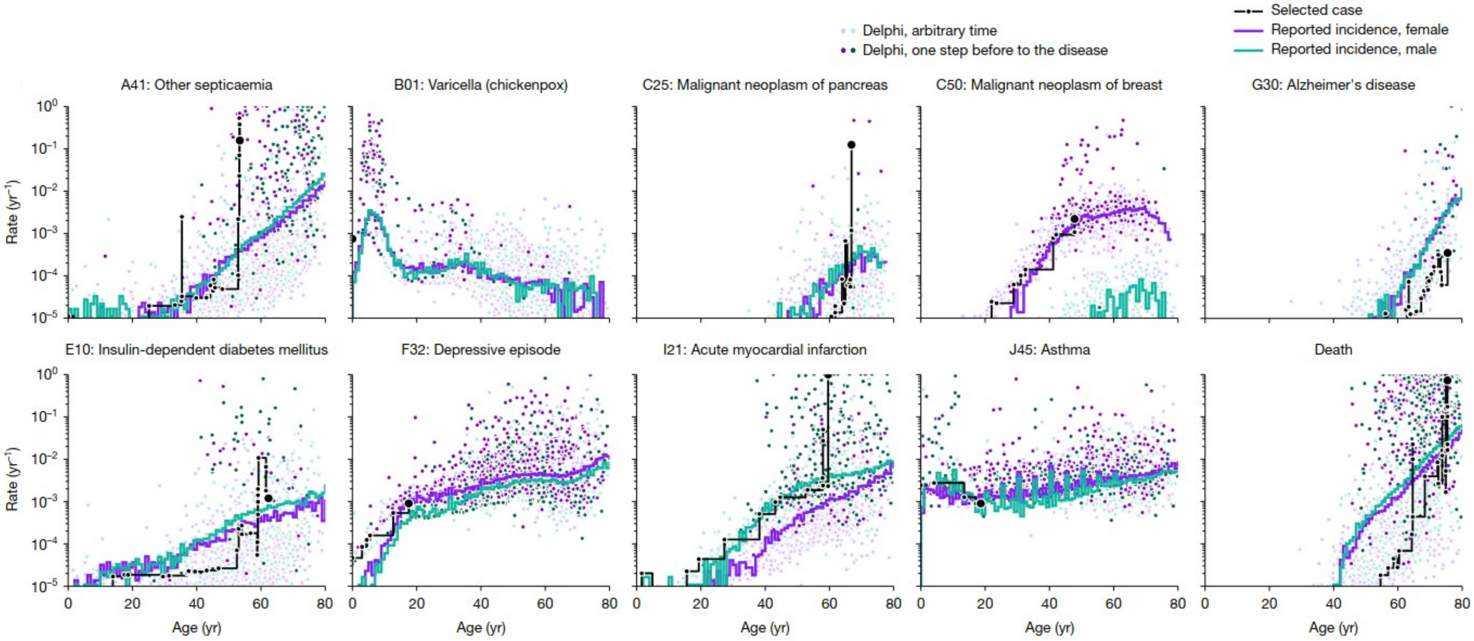
Received: 18 May 2024

Accepted: 13 August 2025

Published online: 17 September 2025

Artem Shmatko[1,2,3,13], Alexander Wolfgang Jung[2,4,5,6,13], Kumar Gaurav[2,13], Søren Brunak[4,7], Laust Hvas Mortensen[5,7,8], Ewan Birney[2]✉, Tom Fitzgerald[2]✉ & Moritz Gerstung[1,2,9,10,11,12]✉

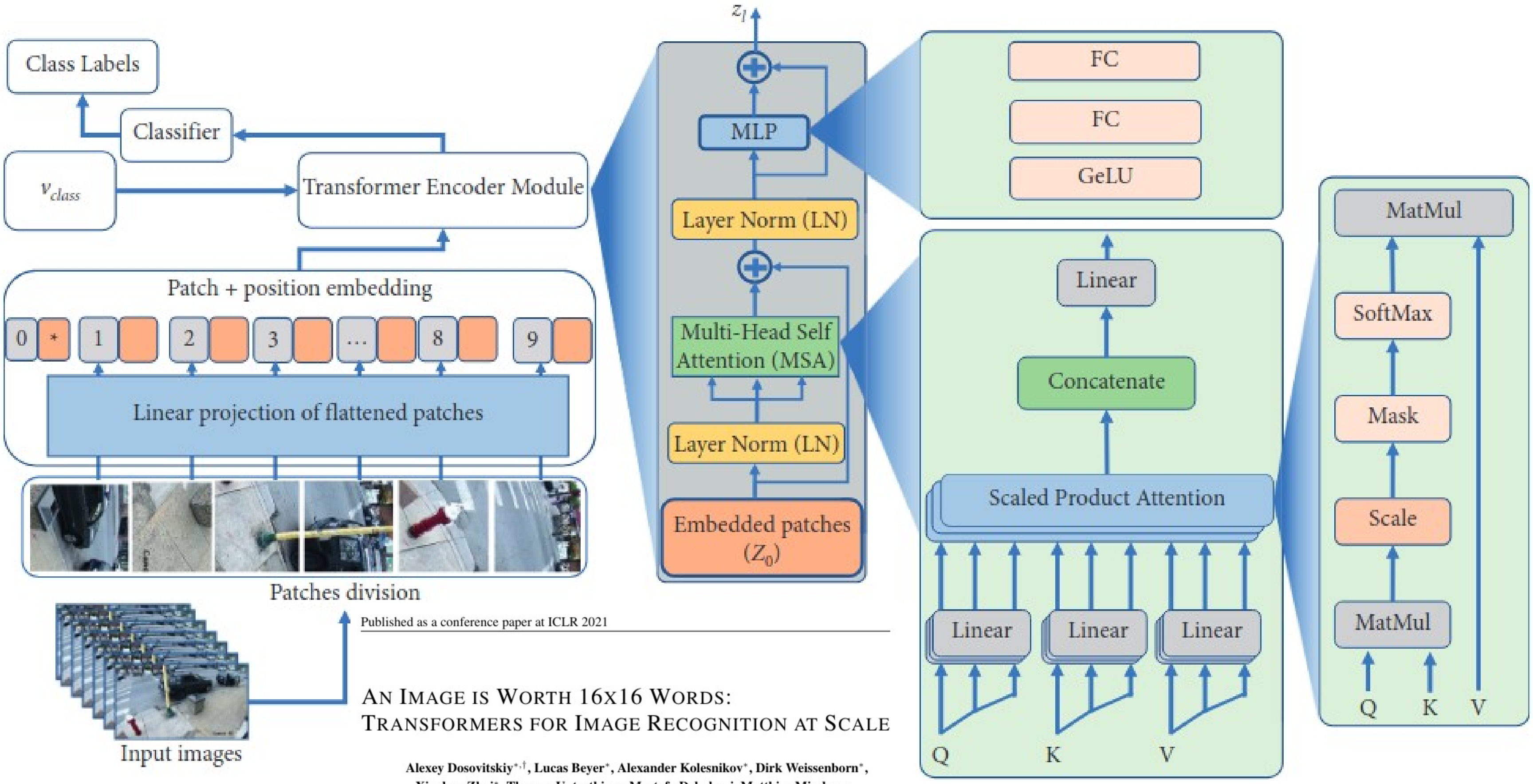
Decision-making in healthcare relies on understanding patients' past and current health states to predict and, ultimately, change their future course[1–3]. Artificial



Input:
Age: Token
0.0: Male
2.0: B01 varicella (chickenpox)
3.0: L20 atopic dermatitis
5.0: No event
10.0: No event
15.0: No event
20.0: No event
20.0: G43 migraine
21.0: E73 lactose intolerance
22.0: B27 infectious mononucleosis
25.0: No event
28.0: J11 influenza, virus not identified
30.0: No event
35.0: No event
40.0: No event
41.0: Smoking low
41.0: BMI mid
41.0: Alcohol low
42.0: No event

Output:
43.2: No event
43.5: M54 dorsalgia
44.6: I86 varicose veins of other sites
50.4: K52 other non-infective gastroenteritis and colitis
52.2: H83 other diseases of inner ear
53.9: J22 unspecified acute lower respiratory infection
54.5: L30 other dermatitis
55.3: No event
57.5: L50 urticaria
59.4: K62 other diseases of anus and rectum
...
69.8: J90 pleural effusion, not elsewhere classified
70.0: K21 gastro-oesophageal reflux disease
70.1: K76 other diseases of liver
70.3: I10 essential primary hypertension
70.4: M85 other disorders of bone density and structure
70.7: M81 osteoporosis without pathological fracture
71.2: J98 other respiratory disorders
72.1: J80 adult respiratory distress syndrome
72.2: No event
72.7: Death

Vision Transformer



Published as a conference paper at ICLR 2021

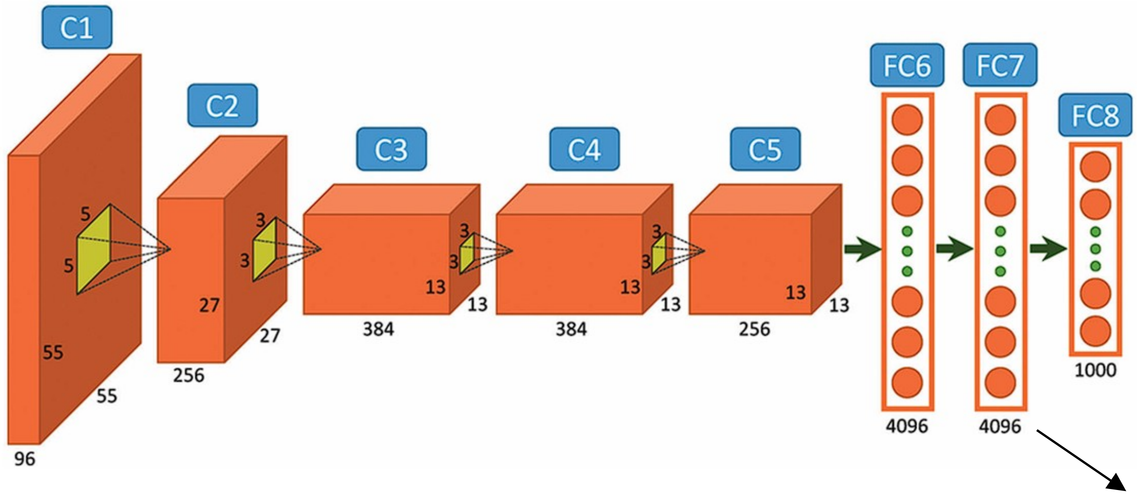
AN IMAGE IS WORTH 16X16 WORDS:
TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

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

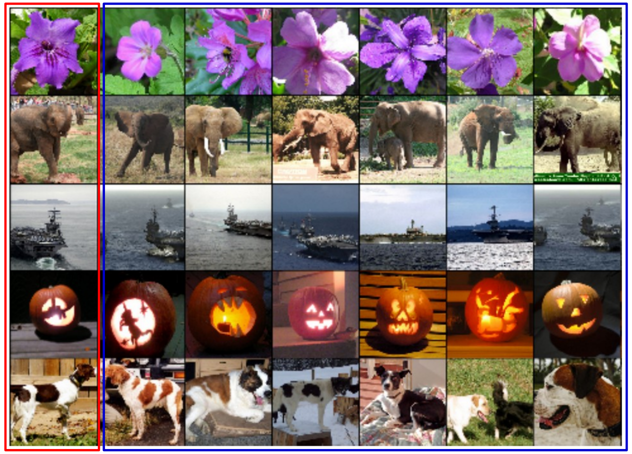
source: https://doi.org/10.1155/2022/3454167

Patches are embedded by CNNs



6 nearest neighbours in the 4096 dimension space

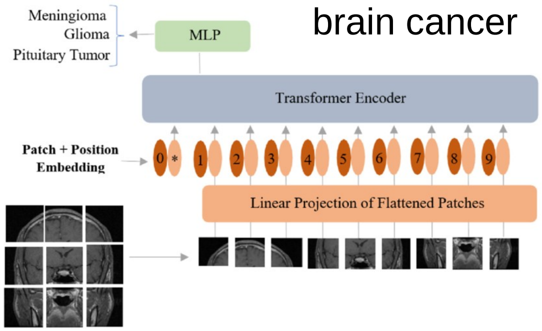
input image



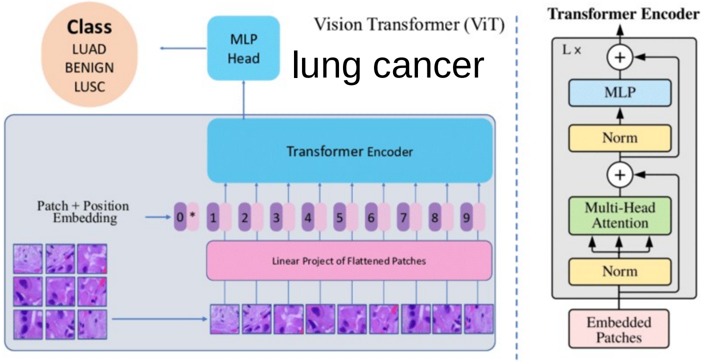
Krizhevsky A, Sutskever I, Hinton GE (2012)
ImageNet Classification with Deep
Convolutional Neural Networks
https://proceedings.neurips.cc/paper/2012/file/
c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

(presenting AlexNet, the first Deep Convolutional Network)

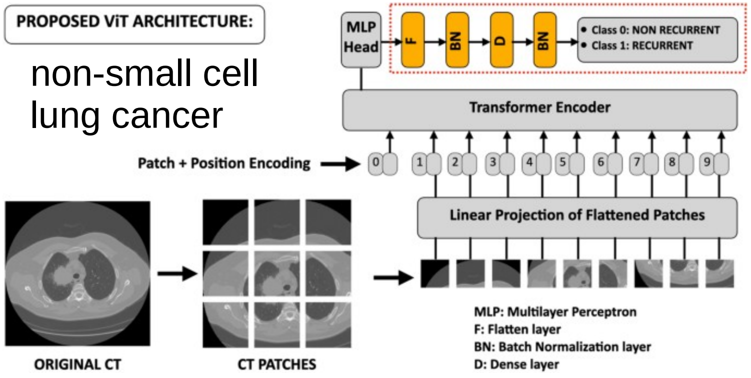
ViTs are replacing vanilla CNNs



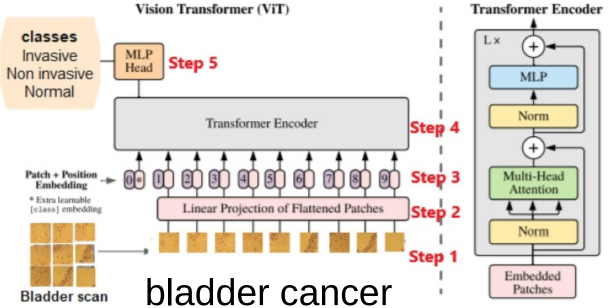
brain cancer



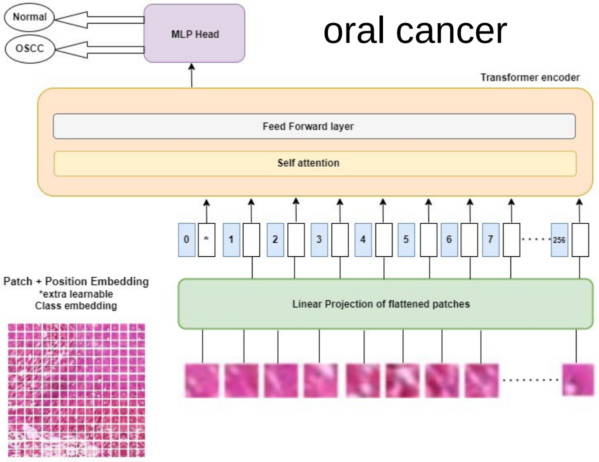
lung cancer


PROPOSED ViT ARCHITECTURE:
MLP Head
F
BN
D
BN
Class 0: NON RECURRENT
Class 1: RECURRENT
Transformer Encoder
Patch + Position Encoding
0 1 2 3 4 5 6 7 8 9
Linear Projection of Flattened Patches
ORIGINAL CT
CT PATCHES
MLP: Multilayer Perceptron
F: Flatten layer
BN: Batch Normalization layer
D: Dense layer

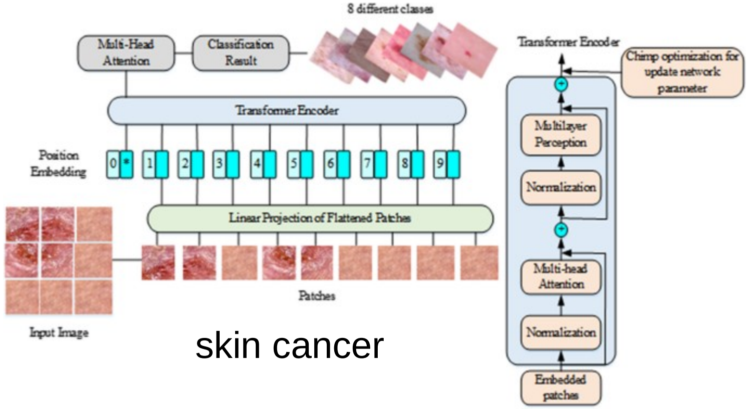
non-small cell lung cancer



bladder cancer

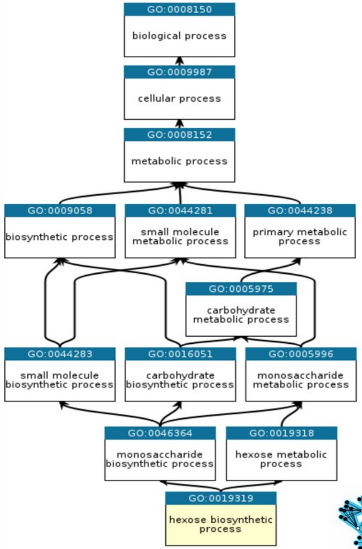


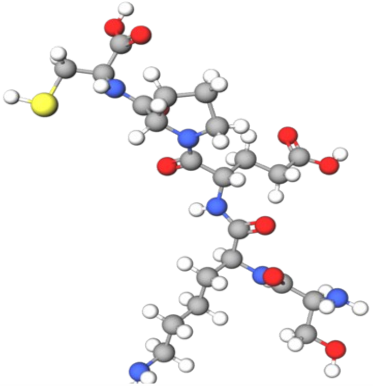
oral cancer

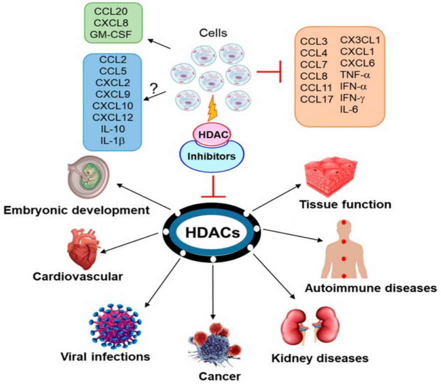


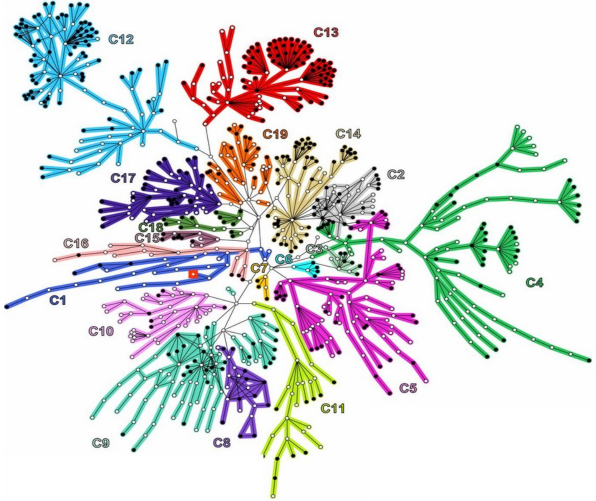
skin cancer

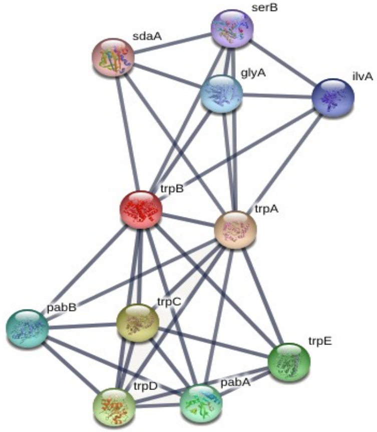
Most biological knowledge comes as graphs

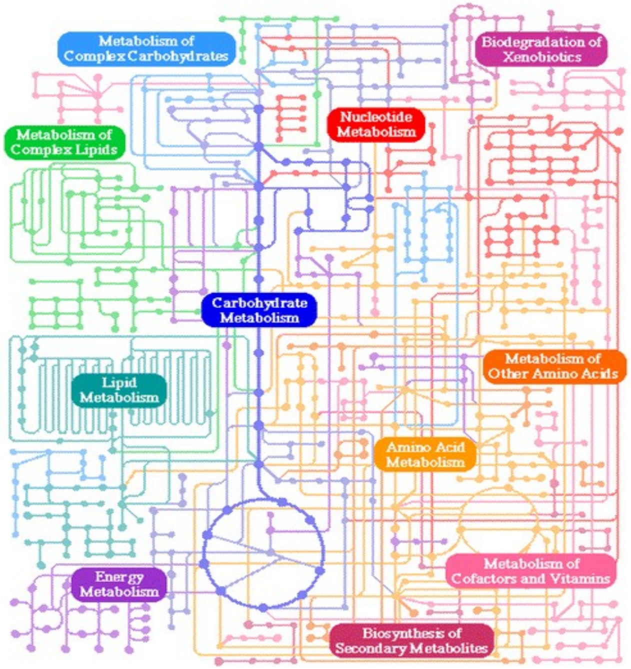

GO:0008150
biological process
GO:0009987
cellular process
GO:0008152
metabolic process
GO:0009058
biosynthetic process
GO:0044281
small molecule metabolic process
GO:0044238
primary metabolic process
GO:0005975
carbohydrate metabolic process
GO:0044283
small molecule biosynthetic process
GO:0016051
carbohydrate biosynthetic process
GO:0005996
monosaccharide metabolic process
GO:0046364
monosaccharide biosynthetic process
GO:0019318
hexose metabolic process
GO:0019319
hexose biosynthetic process





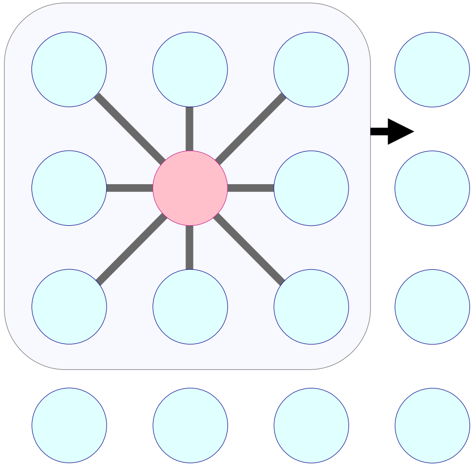




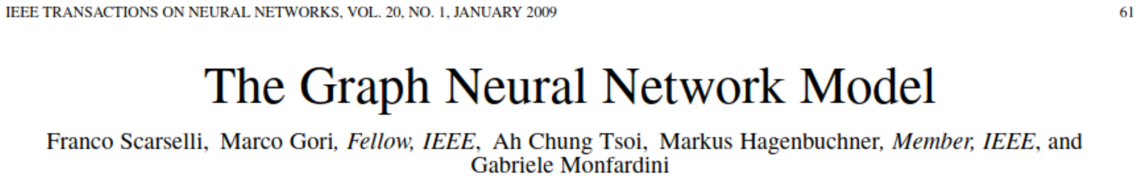

Metabolism of Complex Carbohydrates
Biodegradation of Xenobiotics
Nucleotide Metabolism
Metabolism of Complex Lipids
Carbohydrate Metabolism
Metabolism of Other Amino Acids
Lipid Metabolism
Amino Acid Metabolism
Energy Metabolism
Metabolism of Cofactors and Vitamins
Biosynthesis of Secondary Metabolites

Graph Neural Networks (GNNs)

CNNs



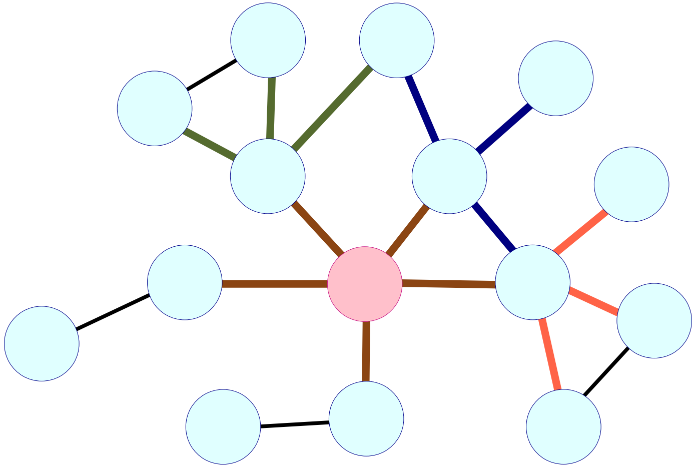
Regular grid (same
number of neighbours)
Homogeneous kernels



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

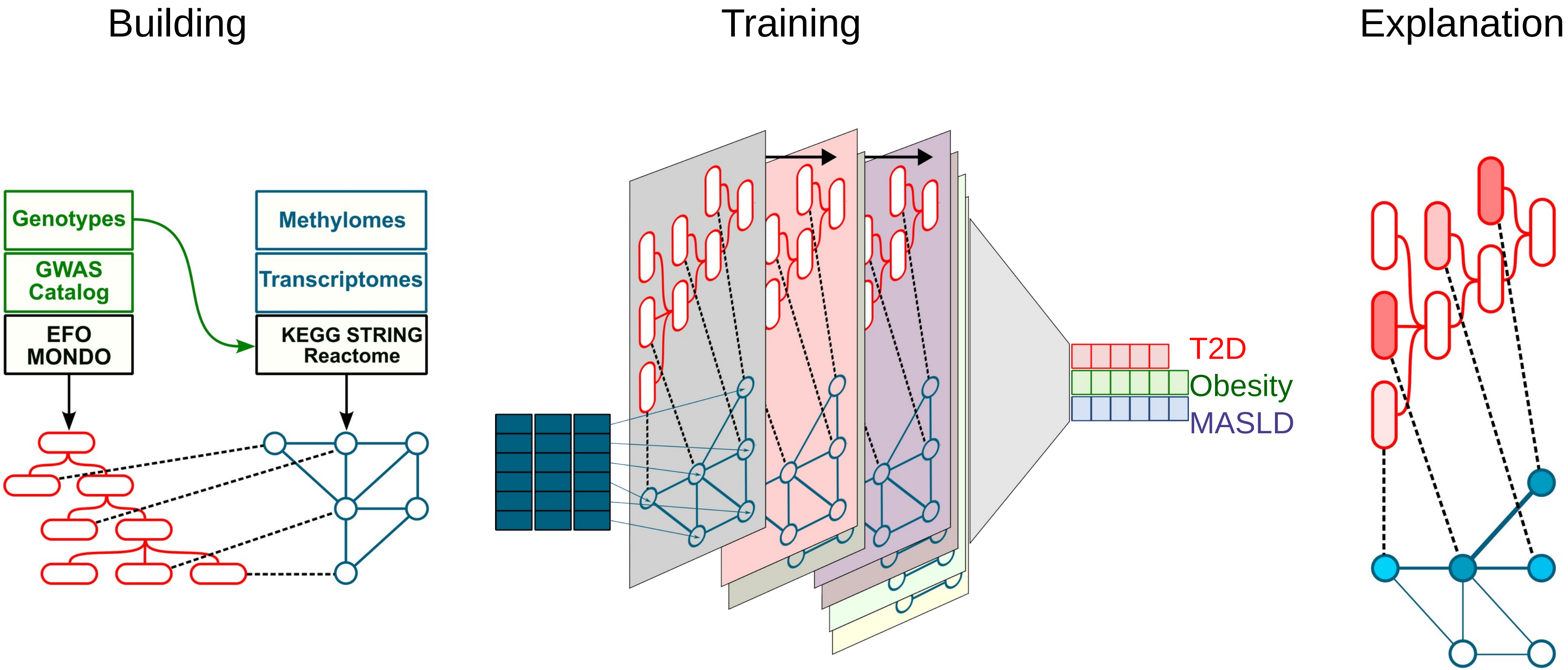
The Graph Neural Network Model

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



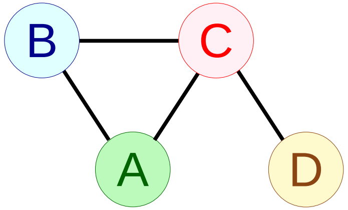
Any number of neighbours
Information passed from
neighbours depends on contexts
and positions.

GNN can be heterogeneous

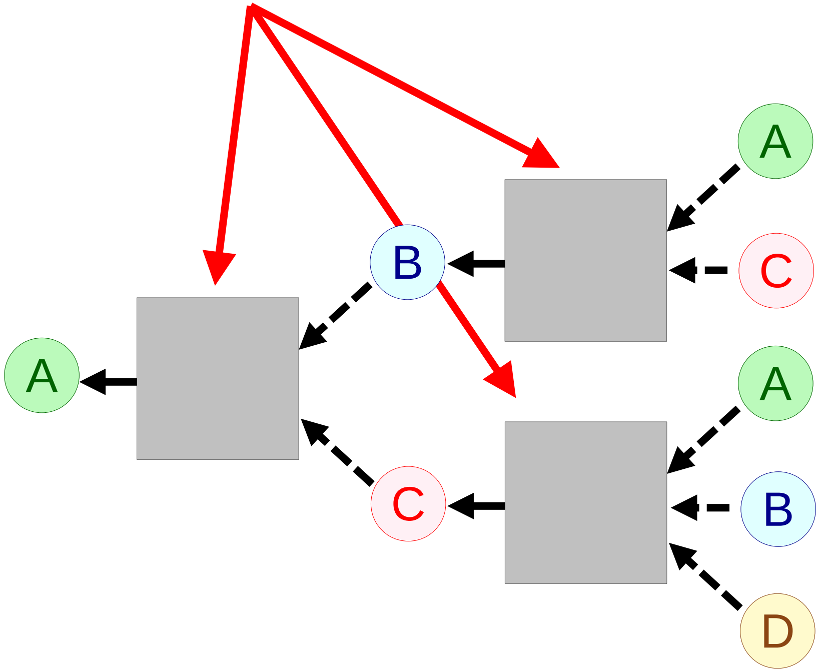


NB: GNNs generally comprise 3 embeddings that are updated at each iteration,
i.e nodes (vertices), edges, and graph

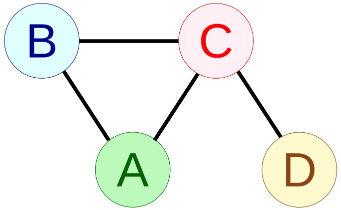
Many different ways to update GNNs



Can be message passing (MLP), convolutions, attention-based, or KANs



Example of GNN convolution



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

Degree matrix

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

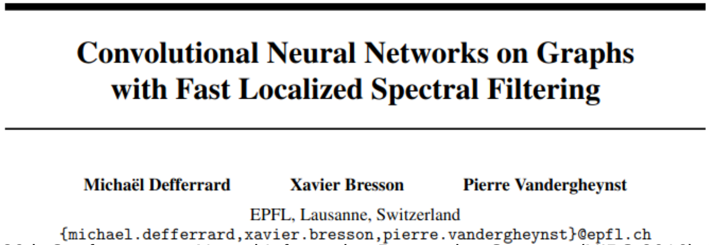
Adjacency matrix

$$L=D-A$$

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

Laplacian matrix

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

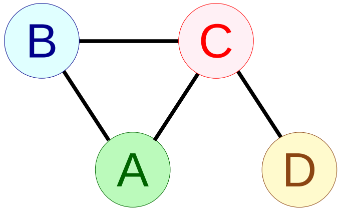


Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering

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



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

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

$$L = D - A$$

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

Laplacian matrix

identity matrix
no influence from neighbours

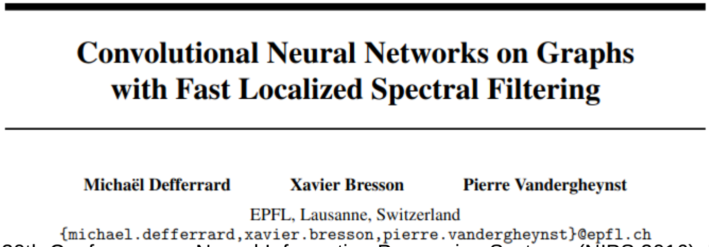
influence from
immediate neighbours

influence from neighbours and
neighbours of neighbours

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

$$y = p_w(L)x$$

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

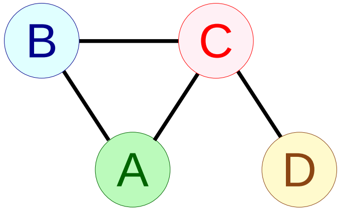


Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering

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

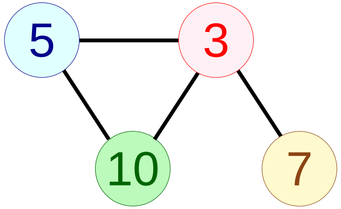
$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

D influences A, B, C

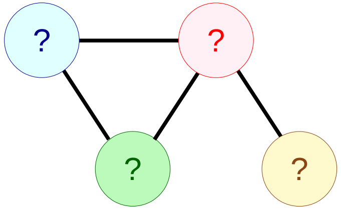
$$L^2 = \begin{bmatrix} 6 & -3 & -4 & 1 \\ -3 & 6 & -4 & 1 \\ -4 & -4 & 12 & -4 \\ 1 & 1 & -4 & 2 \end{bmatrix}$$

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

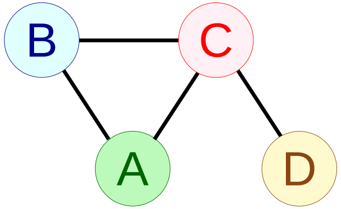
layer 1



layer 2



Example of GNN convolution



D only influences C

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

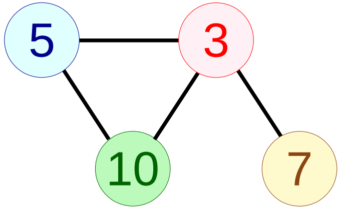
D influences A, B, C

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

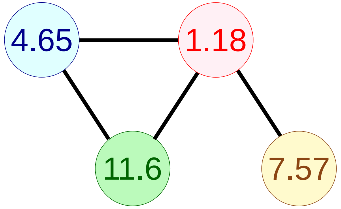
$$w = [1, 0.1, 0.01]$$

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

layer 1

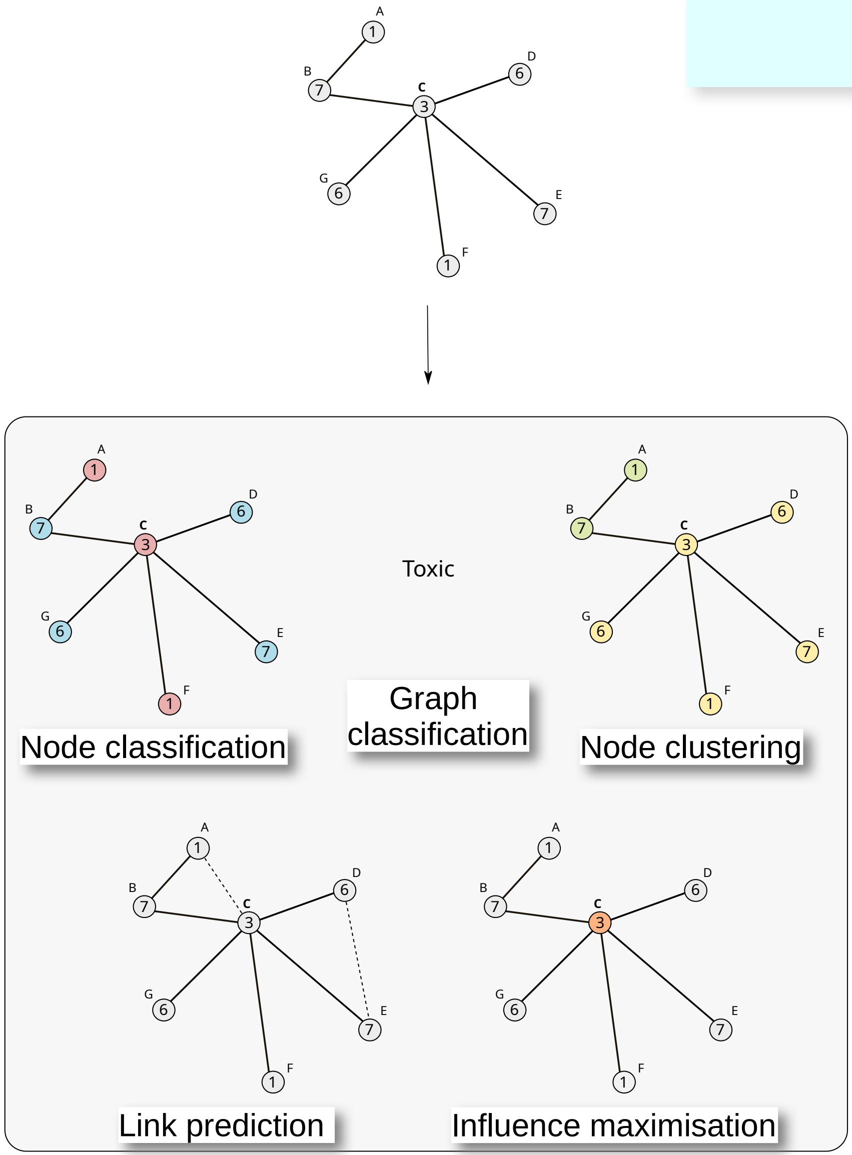


layer 2



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

What can we do with GNN?



Node classification

Graph classification

Node clustering

Link prediction

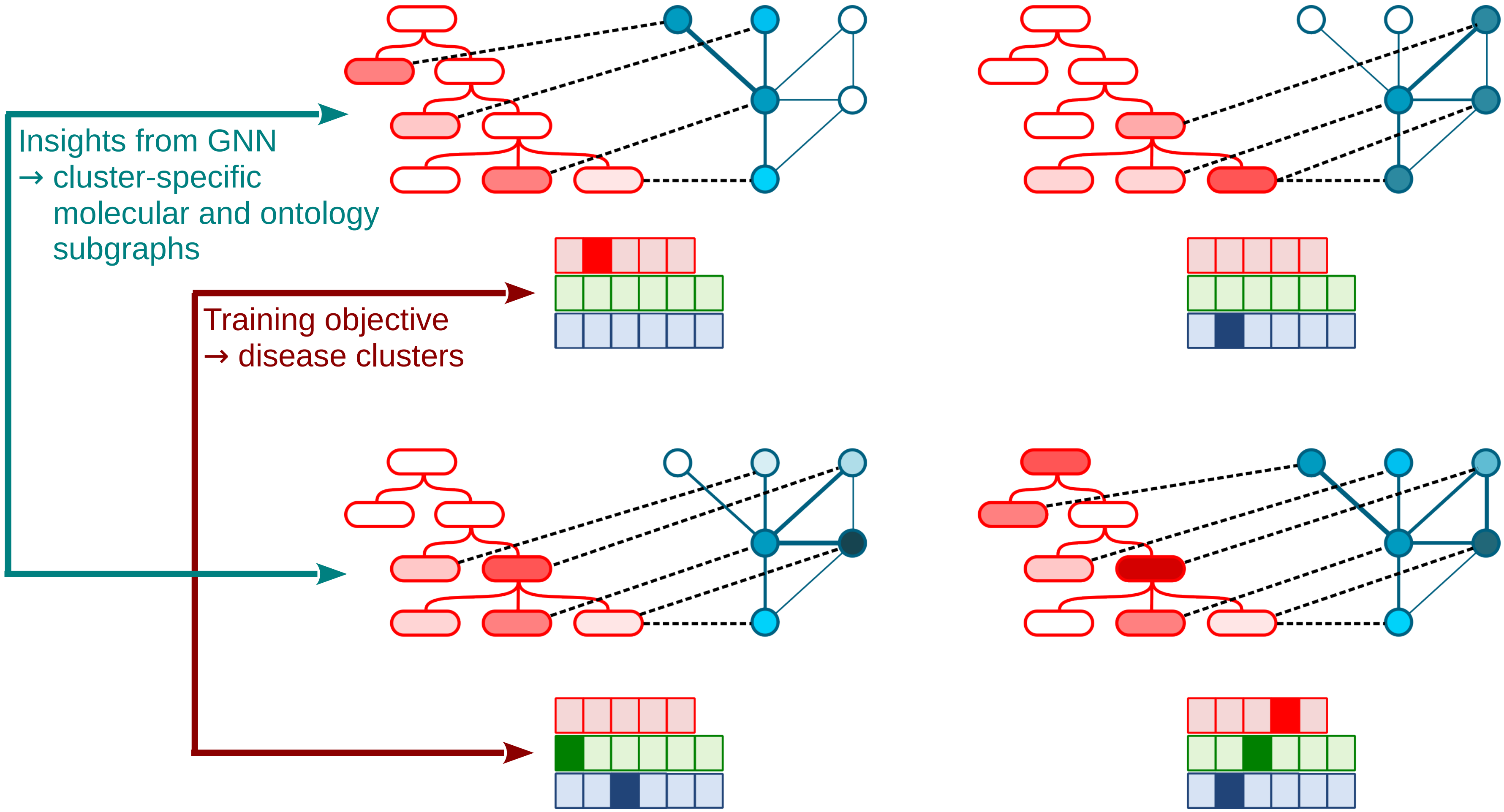
Influence maximisation

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

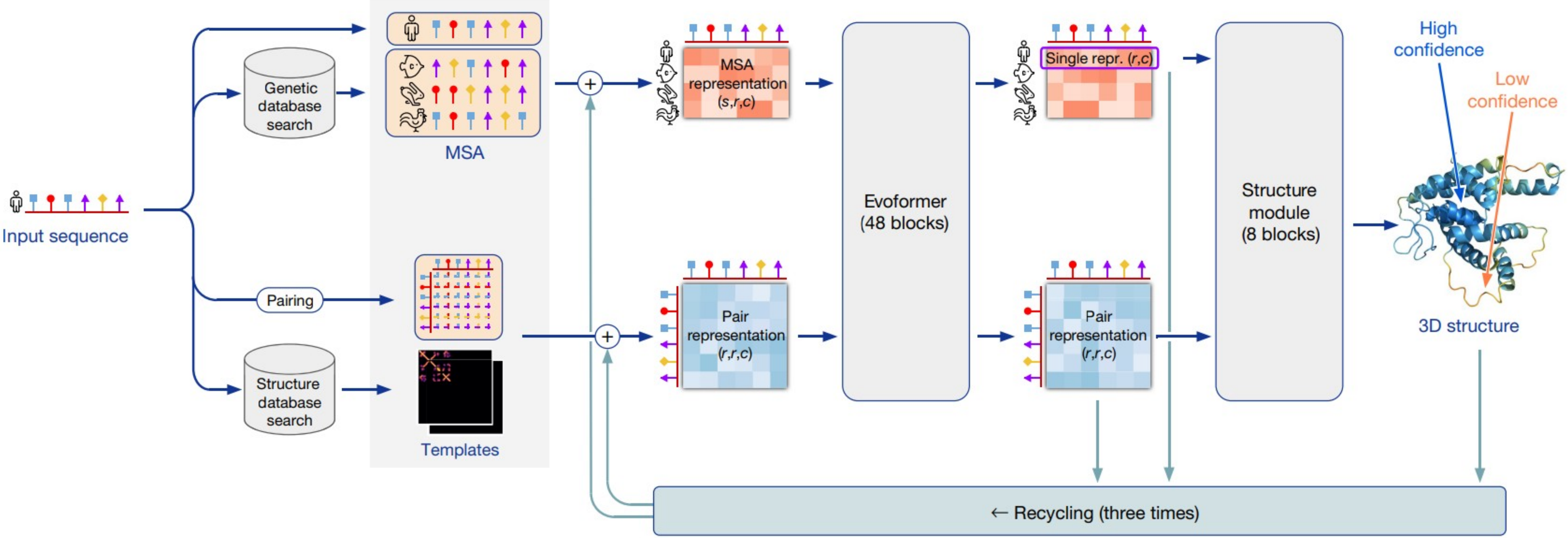
GNN insights can be subgraphs

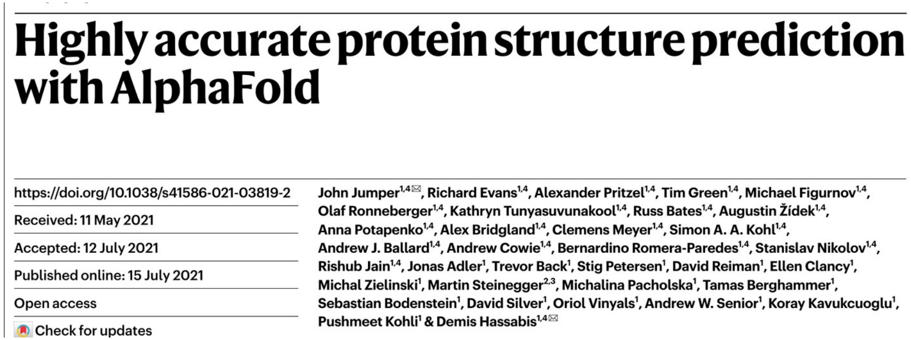


Insights from GNN
→ cluster-specific
molecular and ontology
subgraphs

Training objective
→ disease clusters

AlphaFold2


Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)



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

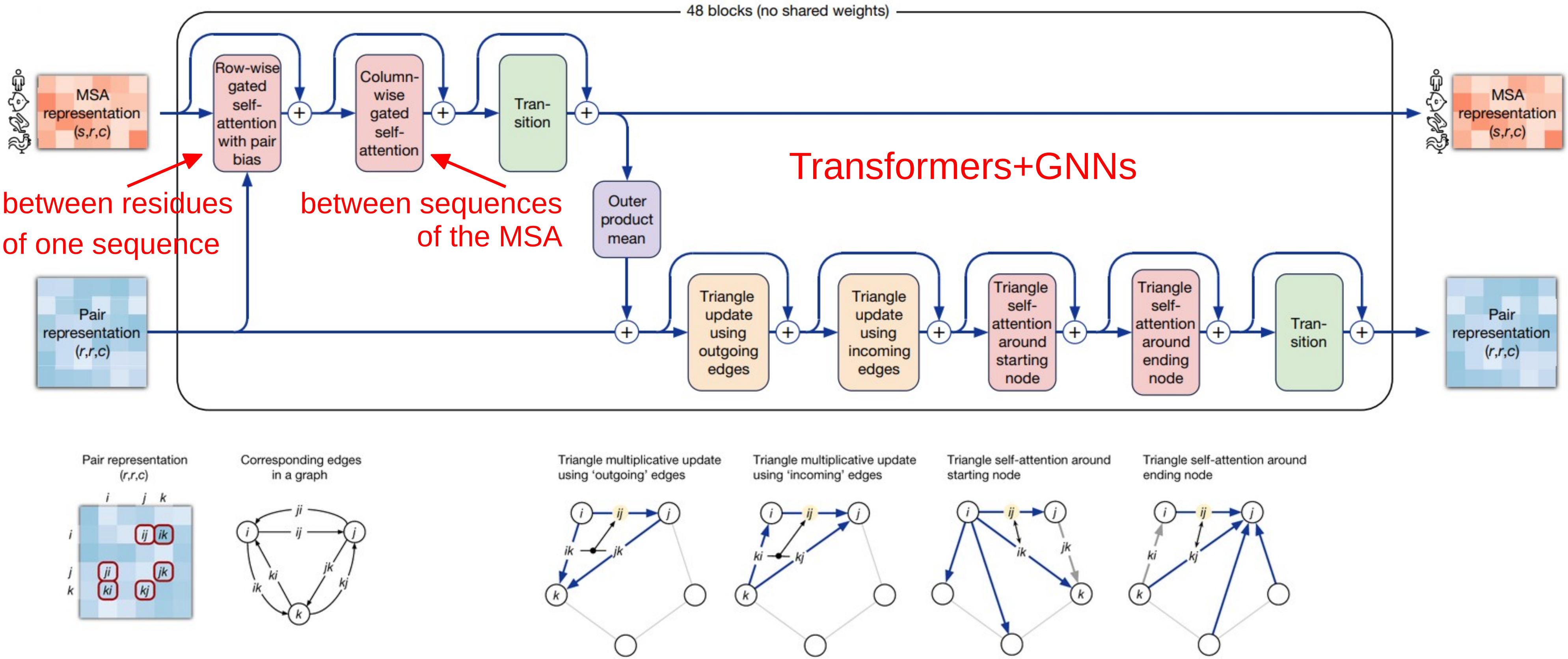
Check for updates

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


48 blocks (no shared weights)
MSA representation (s,r,c)
Row-wise gated self-attention with pair bias
Column-wise gated self-attention
Tran-sition
Outer product mean
Pair representation (r,r,c)
Triangle update using outgoing edges
Triangle update using incoming edges
Triangle self-attention around starting node
Triangle self-attention around ending node
Tran-sition
MSA representation (s,r,c)
Pair representation (r,r,c)
Pair representation (r,r,c)
i j k
ij ik
ji jk
ki kj
Corresponding edges in a graph
ji ij ki ik jk kj
i j k
Triangle multiplicative update using 'outgoing' edges
Triangle multiplicative update using 'incoming' edges
Triangle self-attention around starting node
Triangle self-attention around ending node
i ij j ik jk k
i ij j ki kj k
i ij j ik jk k
i ij j ki kj k

between residues of one sequence

between sequences of the MSA

Transformers+GNNs

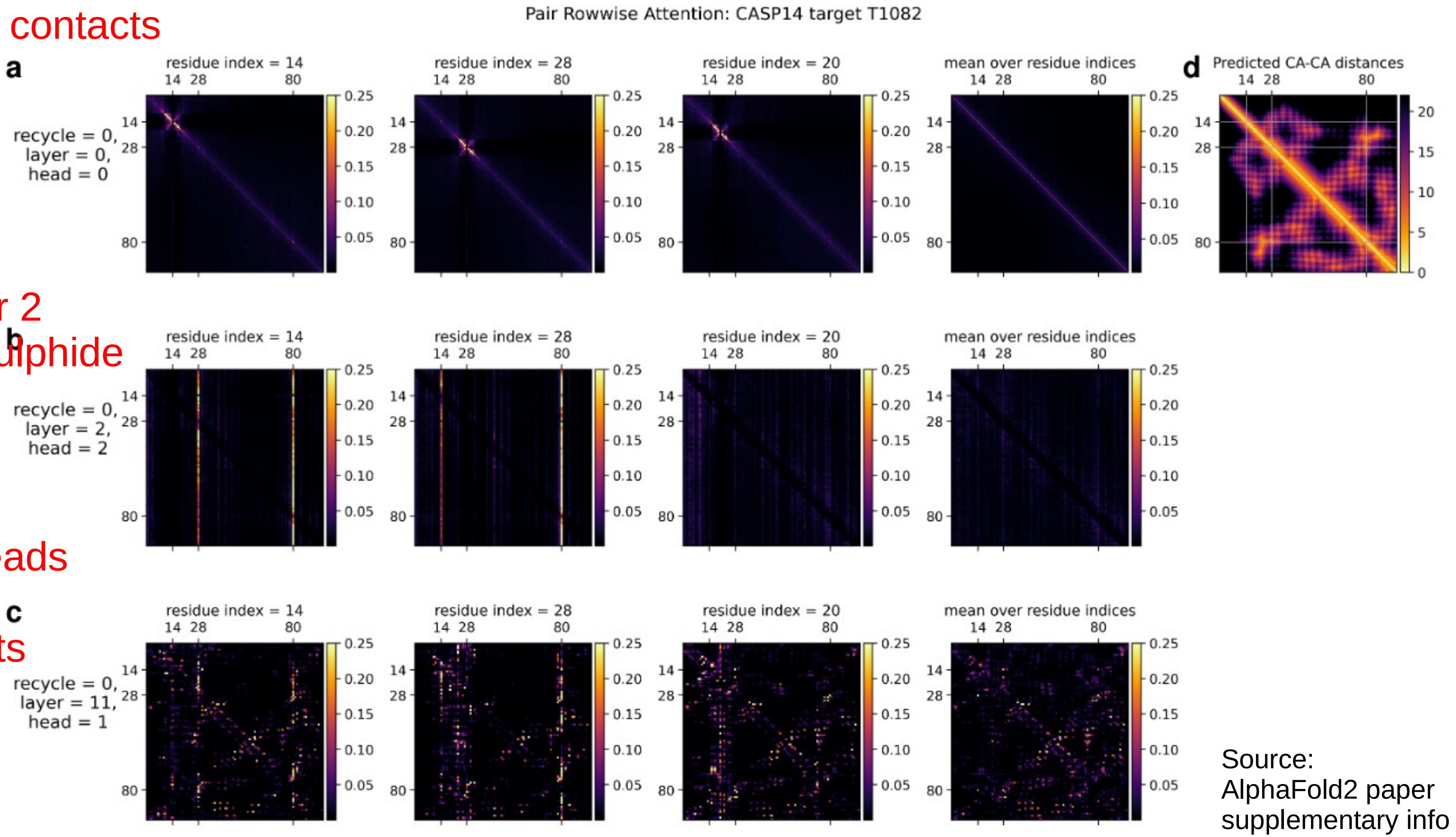
see also: https://www.blopig.com/blog/2021/07/alphafold-2-is-here-whats-behind-the-structure-prediction-miracle/

Row-wise attention: between residues of a sequence

Layer 0 = local contacts

Head 2 of layer 2 recognises disulphide bonds

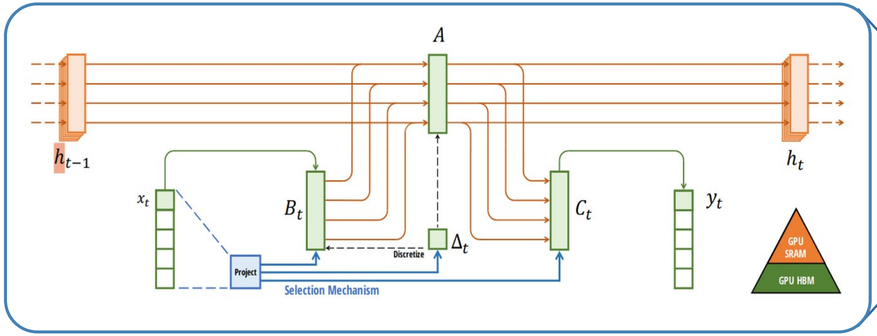
At layer 11, heads recognise distant contacts



Source: AlphaFold2 paper supplementary info

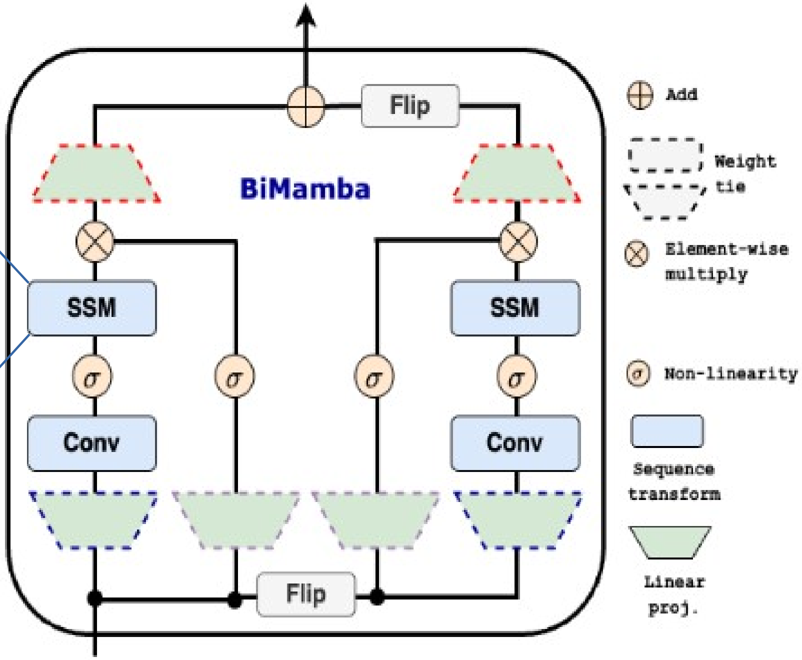
RNNs are back. Rise of the Mamba

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



SSM = State Space Model

Prediction/classification



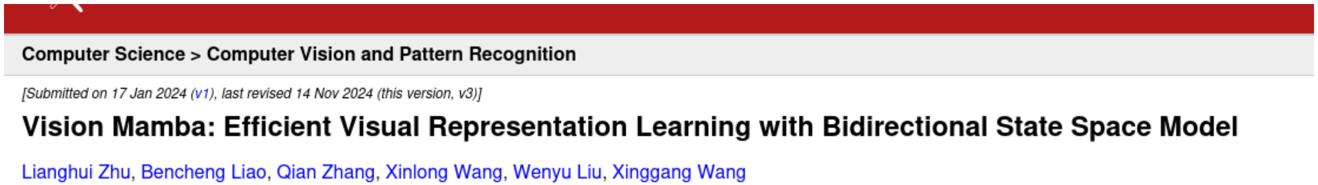
...AGGCTAGGATATCGATAAGCTGACTGAT...

The model learns about variants’ Interactions

→ Reusable foundation model

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

Mamba everywhere

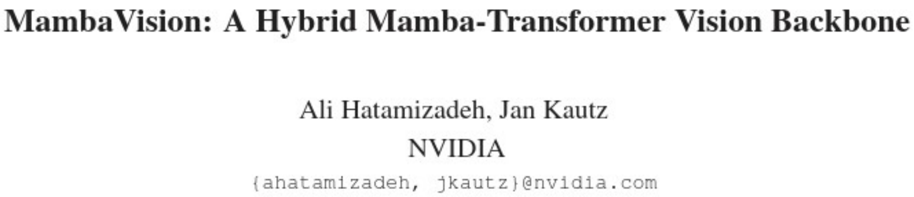


Computer Science > Computer Vision and Pattern Recognition

[Submitted on 17 Jan 2024 (v1), last revised 14 Nov 2024 (this version, v3)]

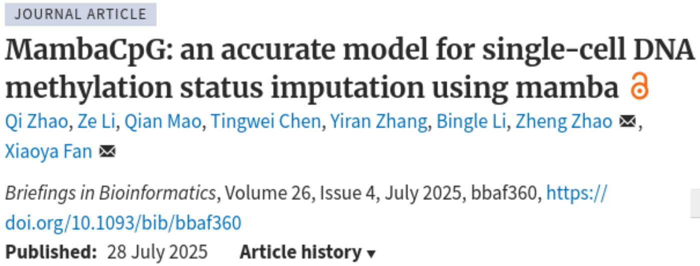
Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, Xinggang Wang



MambaVision: A Hybrid Mamba-Transformer Vision Backbone

Ali Hatamizadeh, Jan Kautz
NVIDIA
{ahatamizadeh, jkautz}@nvidia.com



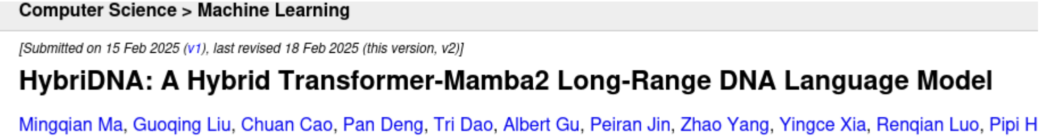
JOURNAL ARTICLE

MambaCpG: an accurate model for single-cell DNA methylation status imputation using mamba

Qi Zhao, Ze Li, Qian Mao, Tingwei Chen, Yiran Zhang, Bingle Li, Zheng Zhao ✉, Xiaoya Fan ✉

Briefings in Bioinformatics, Volume 26, Issue 4, July 2025, bbaf360, https://doi.org/10.1093/bib/bbaf360

Published: 28 July 2025 **Article history** ▾

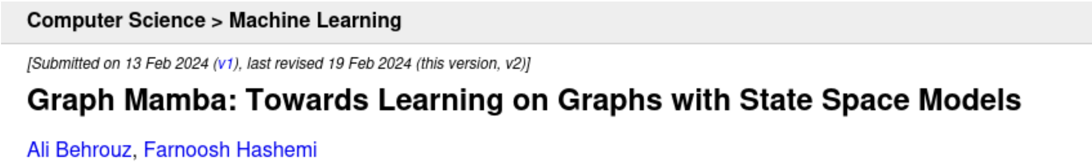


Computer Science > Machine Learning

[Submitted on 15 Feb 2025 (v1), last revised 18 Feb 2025 (this version, v2)]

HybriDNA: A Hybrid Transformer-Mamba2 Long-Range DNA Language Model

Mingqian Ma, Guoqing Liu, Chuan Cao, Pan Deng, Tri Dao, Albert Gu, Peiran Jin, Zhao Yang, Yingce Xia, Renqian Luo, Pipi H

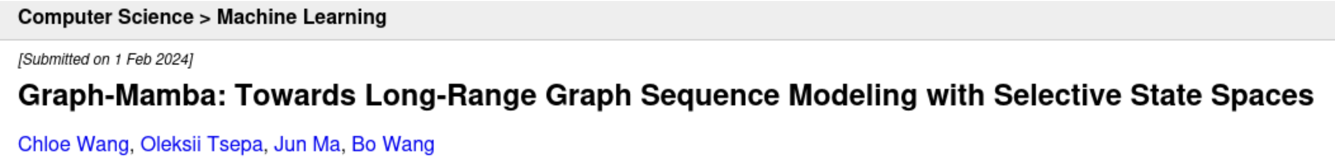


Computer Science > Machine Learning

[Submitted on 13 Feb 2024 (v1), last revised 19 Feb 2024 (this version, v2)]

Graph Mamba: Towards Learning on Graphs with State Space Models

Ali Behrouz, Farnoosh Hashemi



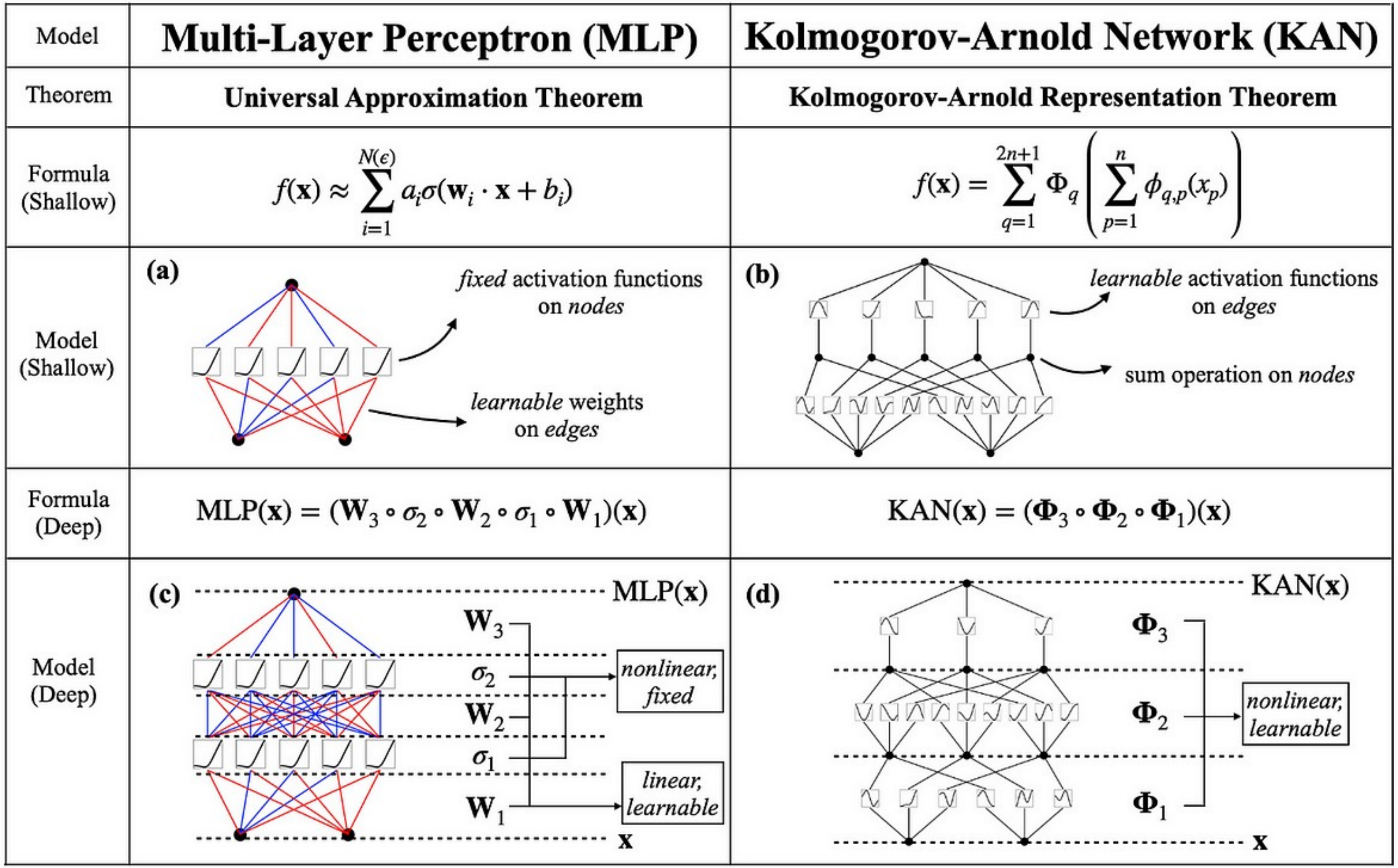
Computer Science > Machine Learning

[Submitted on 1 Feb 2024]

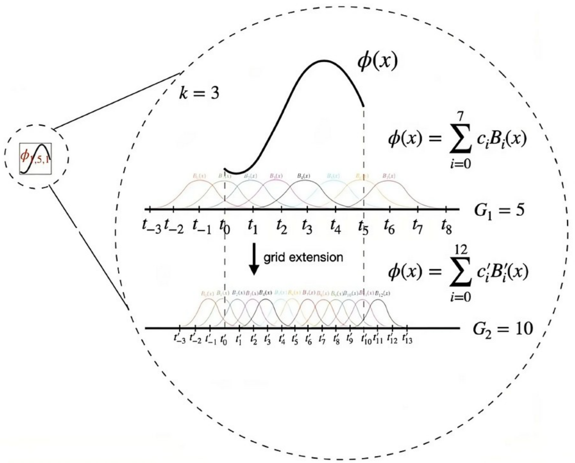
Graph-Mamba: Towards Long-Range Graph Sequence Modeling with Selective State Spaces

Chloe Wang, Oleksii Tsepa, Jun Ma, Bo Wang

Kolmogorov Arnold Networks



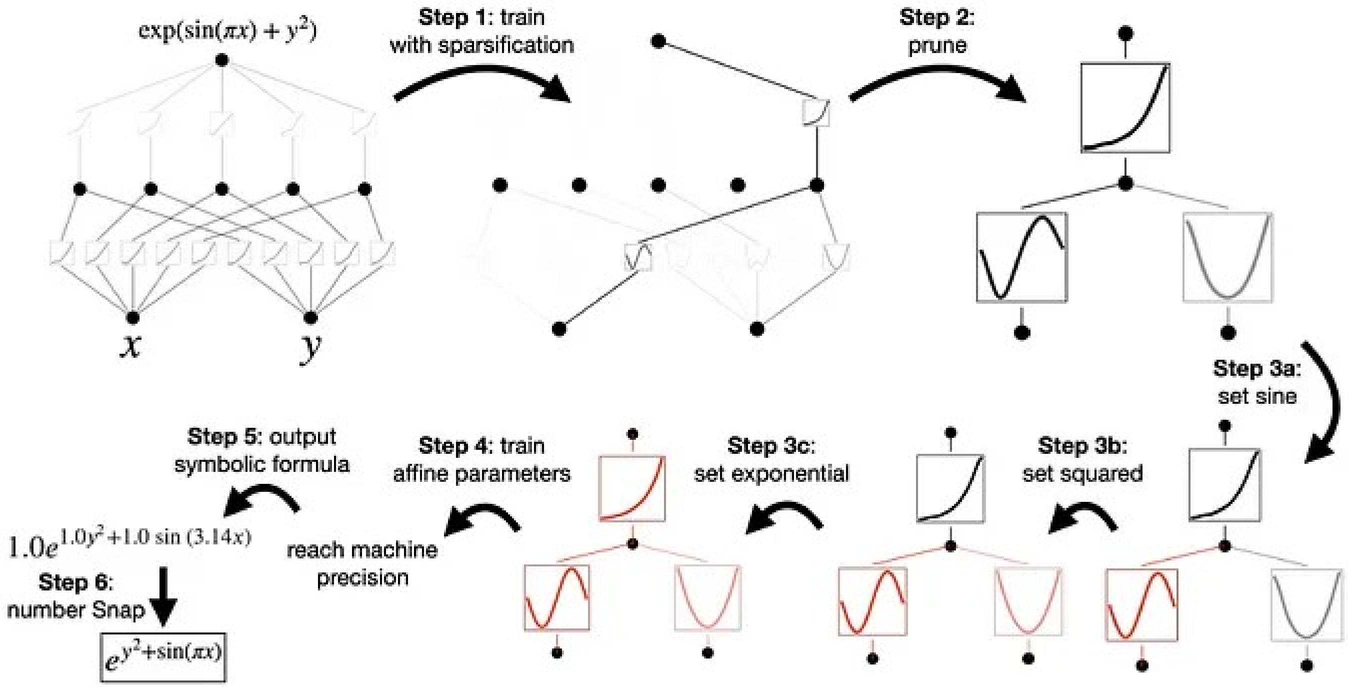
Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
Model (Shallow)	(a) *fixed* activation functions on *nodes* *learnable* weights on *edges*	(b) *learnable* activation functions on *edges* sum operation on *nodes*
Formula (Deep)	$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)	(c) MLP(**x**) $\mathbf{W}_3$ σ_2 $\mathbf{W}_2$ σ_1 $\mathbf{W}_1$ **x** *nonlinear, fixed* *linear, learnable*	(d) KAN(**x**) $\mathbf{\Phi}_3$ $\mathbf{\Phi}_2$ $\mathbf{\Phi}_1$ **x** *nonlinear, learnable*



$\phi_{l,5,1}$
$k = 3$
$\phi(x)$
$\phi(x) = \sum_{i=0}^{7} c_i B_i(x)$
t_{-3} t_{-2} t_{-1} t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8
$G_1 = 5$
grid extension
$\phi(x) = \sum_{i=0}^{12} c_i' B_i'(x)$
$G_2 = 10$

Liu *et al.*(2024) KAN: Kolmogorov-Arnold Networks. https://doi.org/10.48550/arXiv.2404.19756

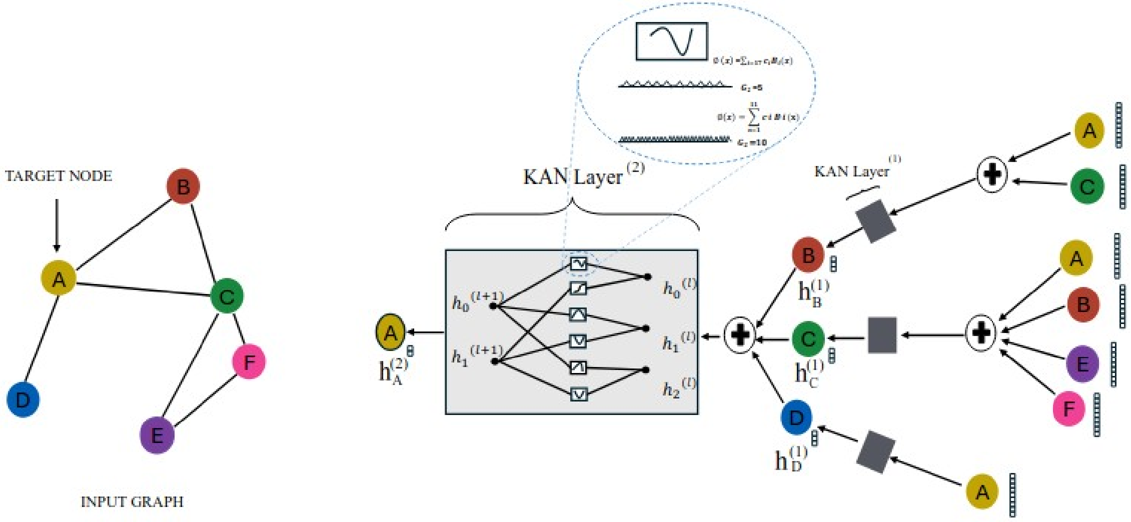
KANs to extract symbolic formulas



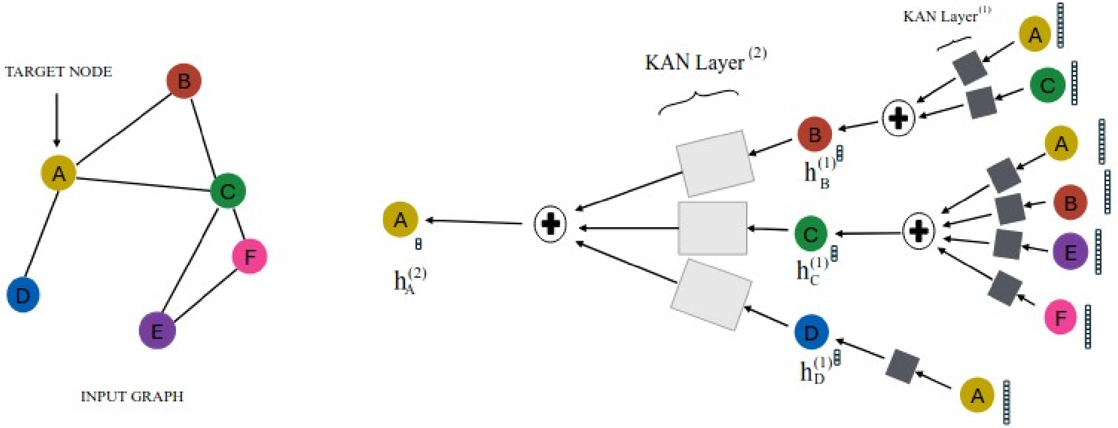
$\exp(\sin(\pi x) + y^2)$
x
y
Step 1: train with sparsification
Step 2: prune
Step 3a: set sine
Step 3b: set squared
Step 3c: set exponential
Step 4: train affine parameters
reach machine precision
Step 5: output symbolic formula
$1.0e^{1.0y^2+1.0\sin(3.14x)}$
Step 6: number Snap
$e^{y^2+\sin(\pi x)}$

Liu *et al.*(2024) KAN: Kolmogorov-Arnold Networks. https://doi.org/10.48550/arXiv.2404.19756

KANs in GNNs

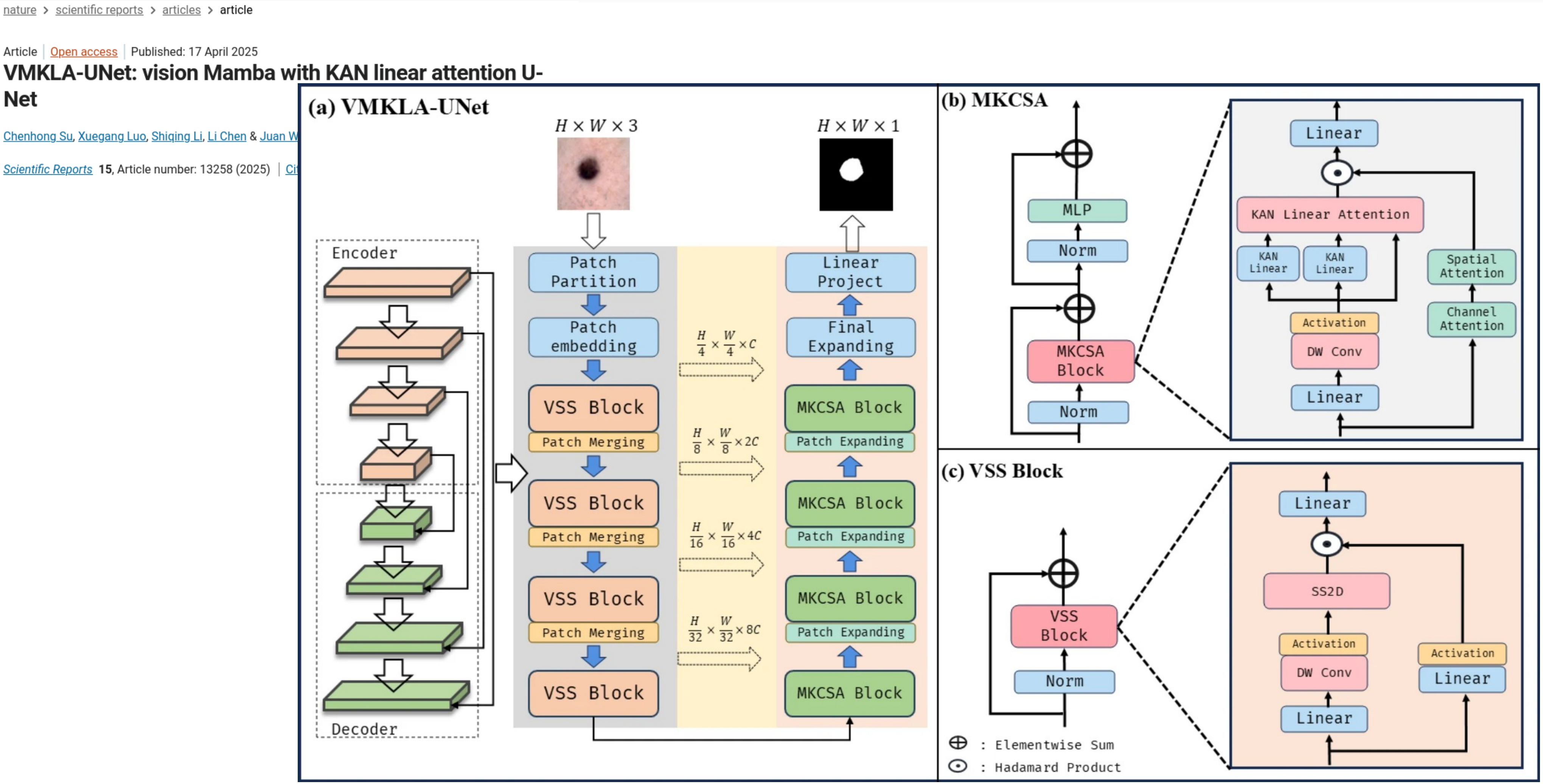


(b) Overview of a two-layer GKAN Architecture 1.



Kiamari *et al.*(2024)
GKAN: Graph Kolmogorov-Arnold Networks.
https://doi.org/10.48550/arXiv.2406.06470

How many architectures can you cram into one model?



nature > scientific reports > articles > article

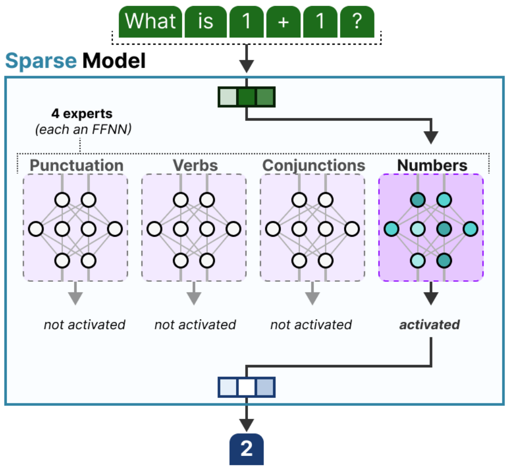
Article | Open access | Published: 17 April 2025

VMKLA-UNet: vision Mamba with KAN linear attention U-Net

Chenhong Su, Xuegang Luo, Shiqing Li, Li Chen & Juan W

Scientific Reports **15**, Article number: 13258 (2025) | Ci

Topics to explore



Mixture Of Experts

https://huggingface.co/blog/moe

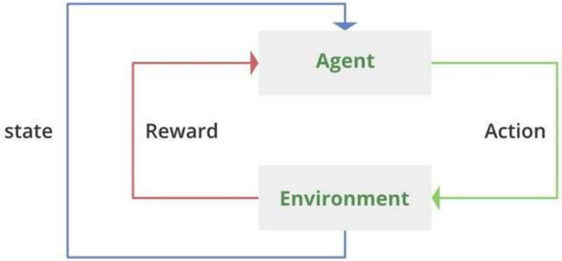
https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

Reinforcement learning

https://fr.mathworks.com/content/dam/mathworks/ebook/gated/reinforcement-learning-ebook-all-chapters.pdf

https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

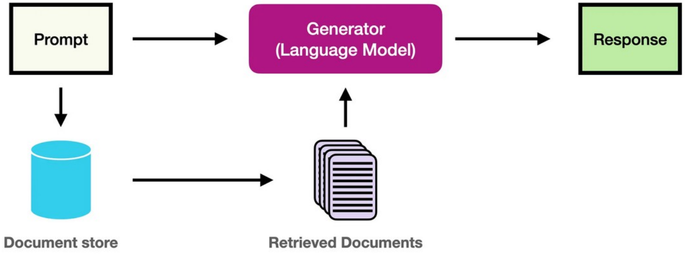
https://arxiv.org/pdf/2412.05265



RAG

https://en.wikipedia.org/wiki/Retrieval-augmented_generation

https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/



PreciDIAB
National Center for
Precision Diabetic Medicine

Egid
European
Genomic
Institute
for Diabetes

Région
Hauts-de-France

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