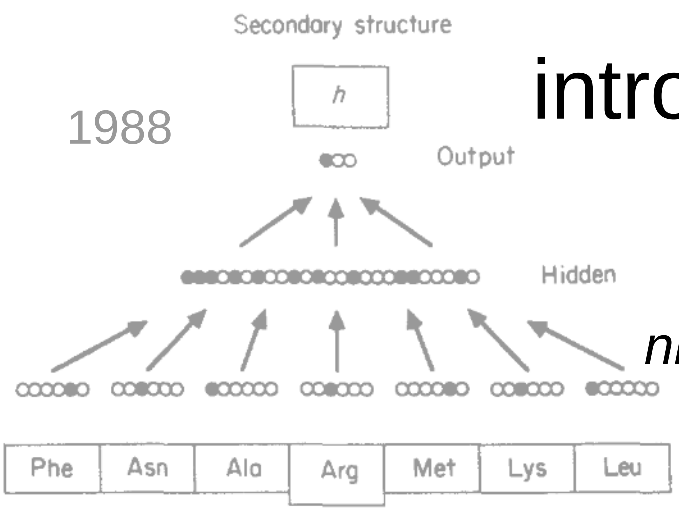
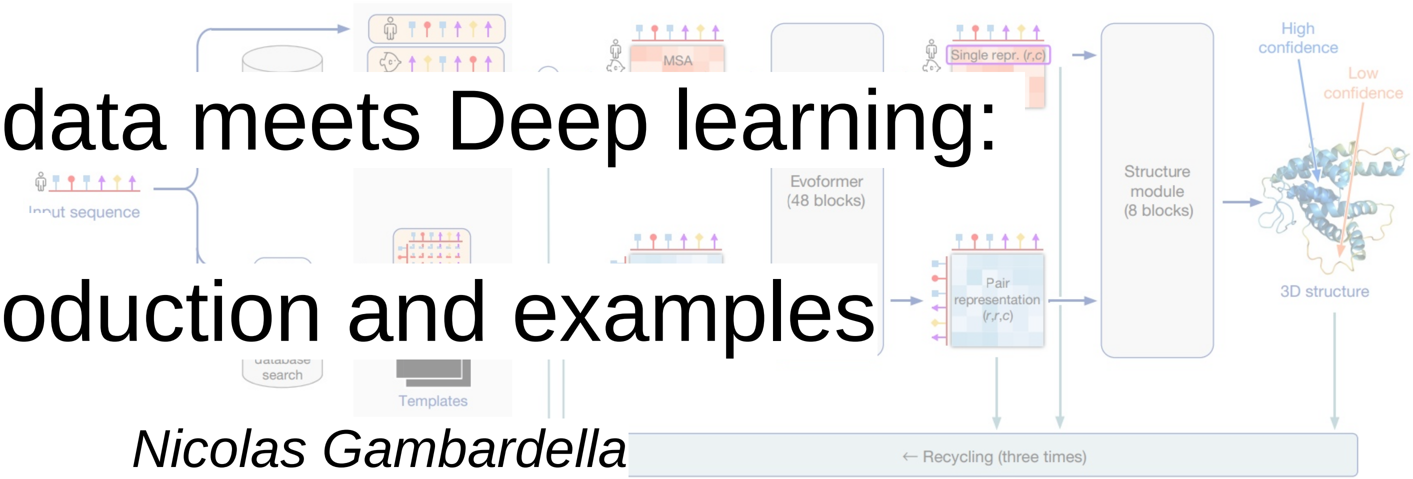
PreciDIAB
National Center for Precision Diabetic Medicine

Egid
European Genomic Institute for Diabetes

Omics data meets Deep learning: introduction and examples

1988





2021

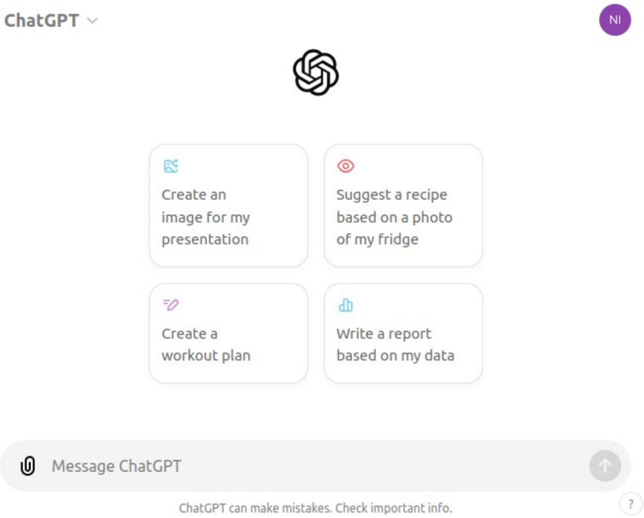
Nicolas Gambardella

nicolas.gambardella@univ-lille.fr

Région
Hauts-de-France

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


HD-03


ChatGPT
NI
Create an image for my presentation
Suggest a recipe based on a photo of my fridge
Create a workout plan
Write a report based on my data
Message ChatGPT
ChatGPT can make mistakes. Check important info.
?


AlphaGo
Lee Sedol


AlphaFold

What are we going to (attempt to) cover?

1) What is artificial intelligence?
2) Multi-Layer Perceptrons (MLP)
3) MLPs in action
4) Convolutional Neural Networks (CNN)
5) Embeddings and latent space
6) Encoder-Decoders,
   (variational) AutoEncoders (VAEs)
7) Attention and the Transformer
8) Graph Neural Networks (GNNs)
9) Alphafold2
10) Recurrent Neural Networks (RNNs)
    and Long Short-Term Memory (LSTM)
11) RNNs are back: Rise of the Mamba
12) Generative Adversarial Networks (GANs)

1

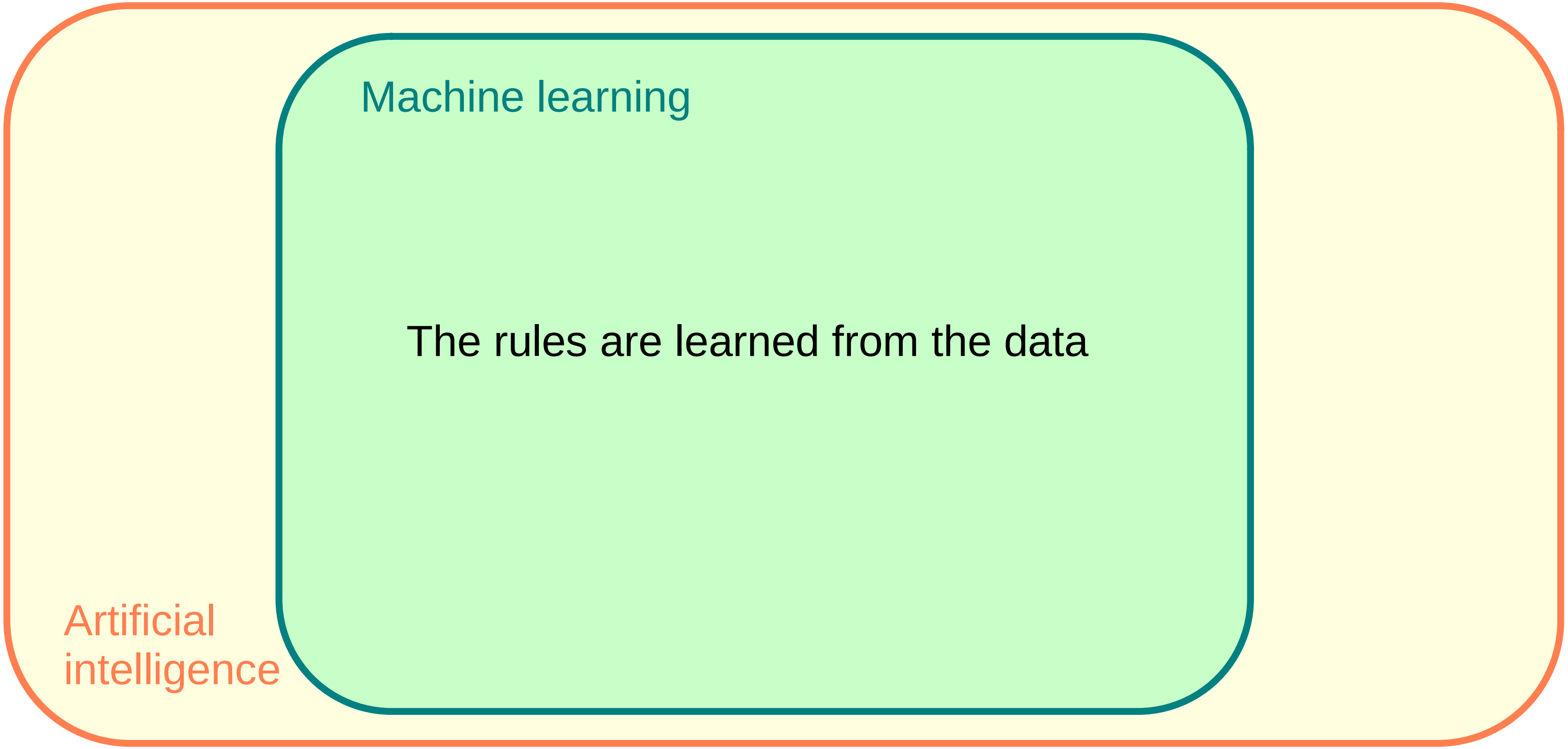
What is artificial intelligence?

Some terminology

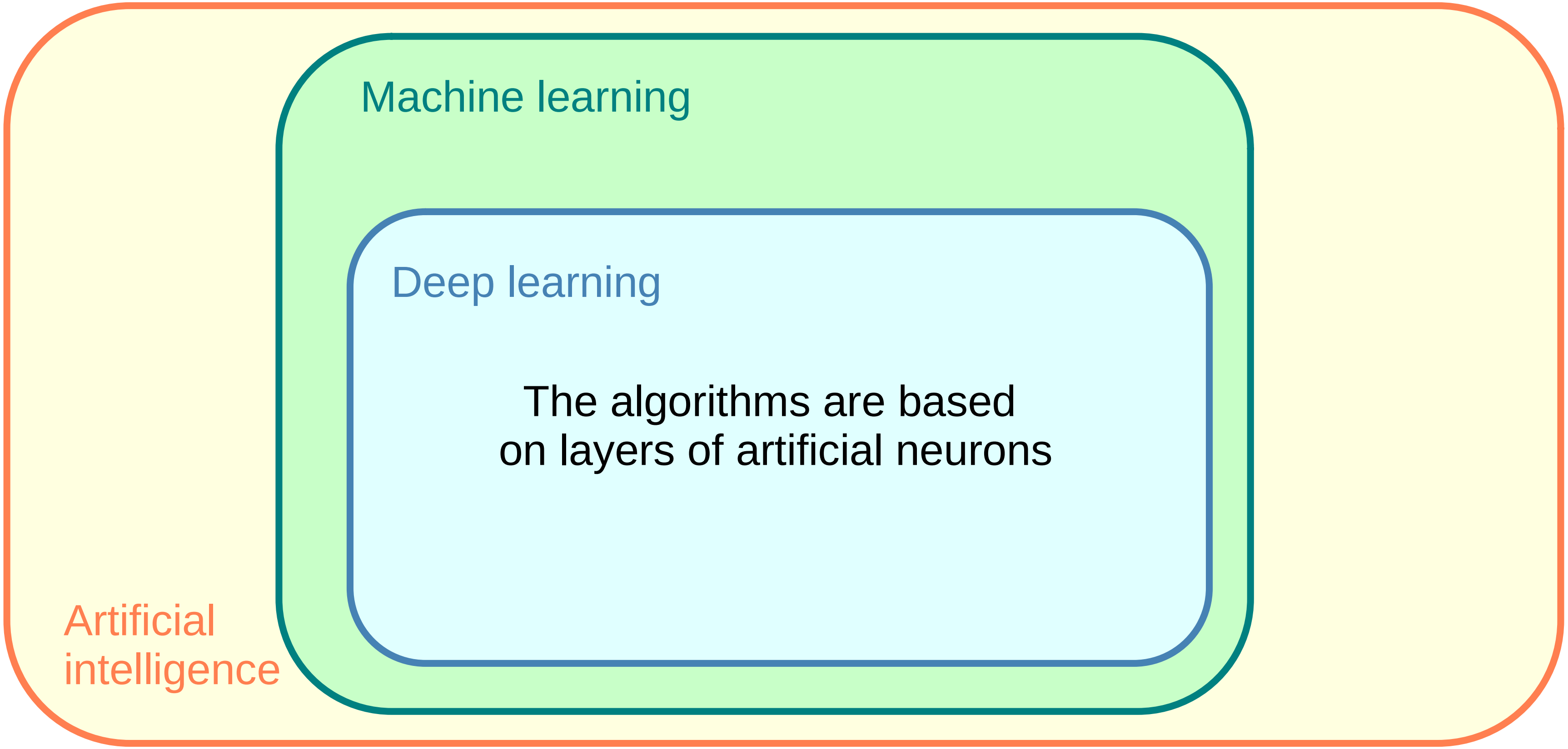
Assessments, evaluations, decisions, predictions made by software tools

Artificial intelligence

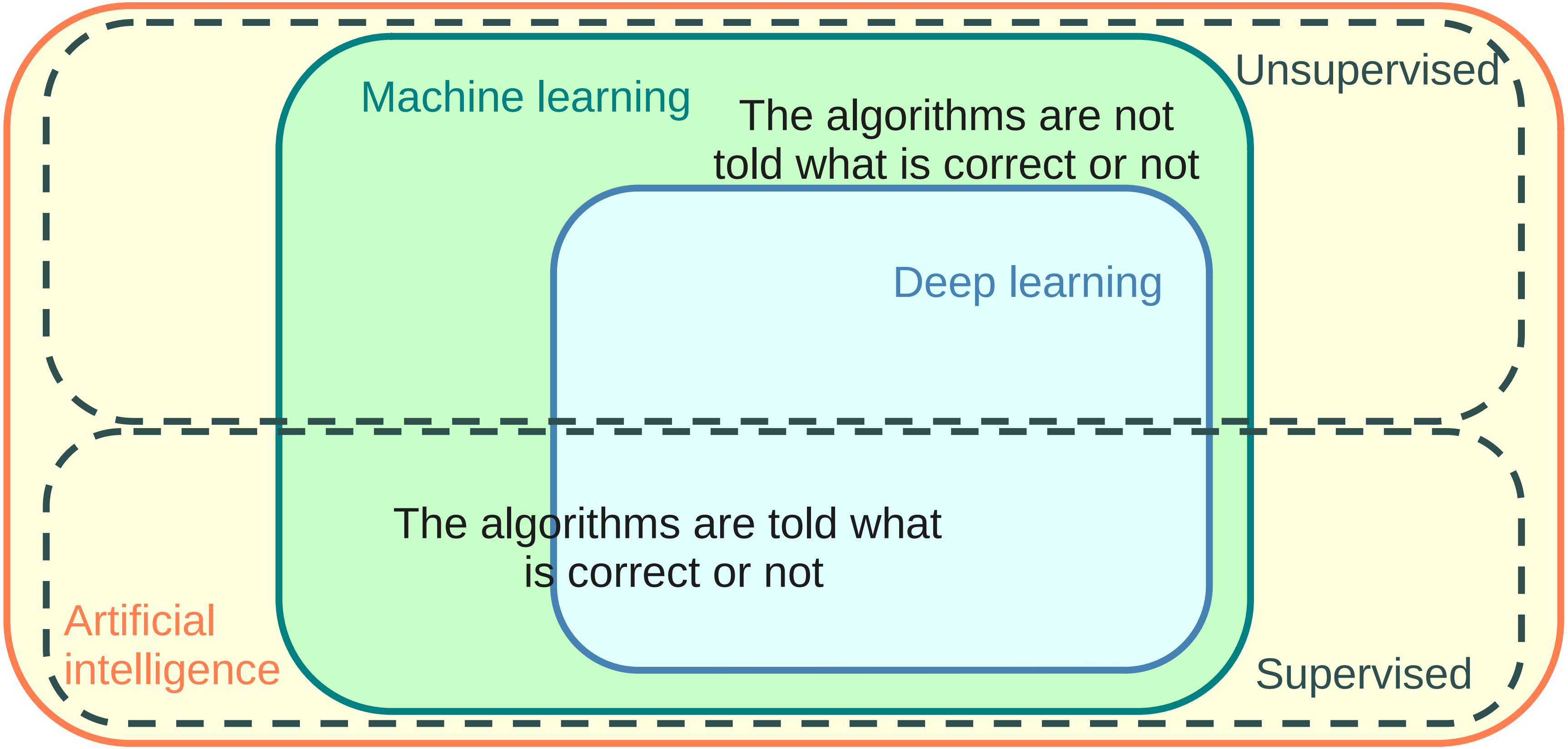
Some terminology



Some terminology

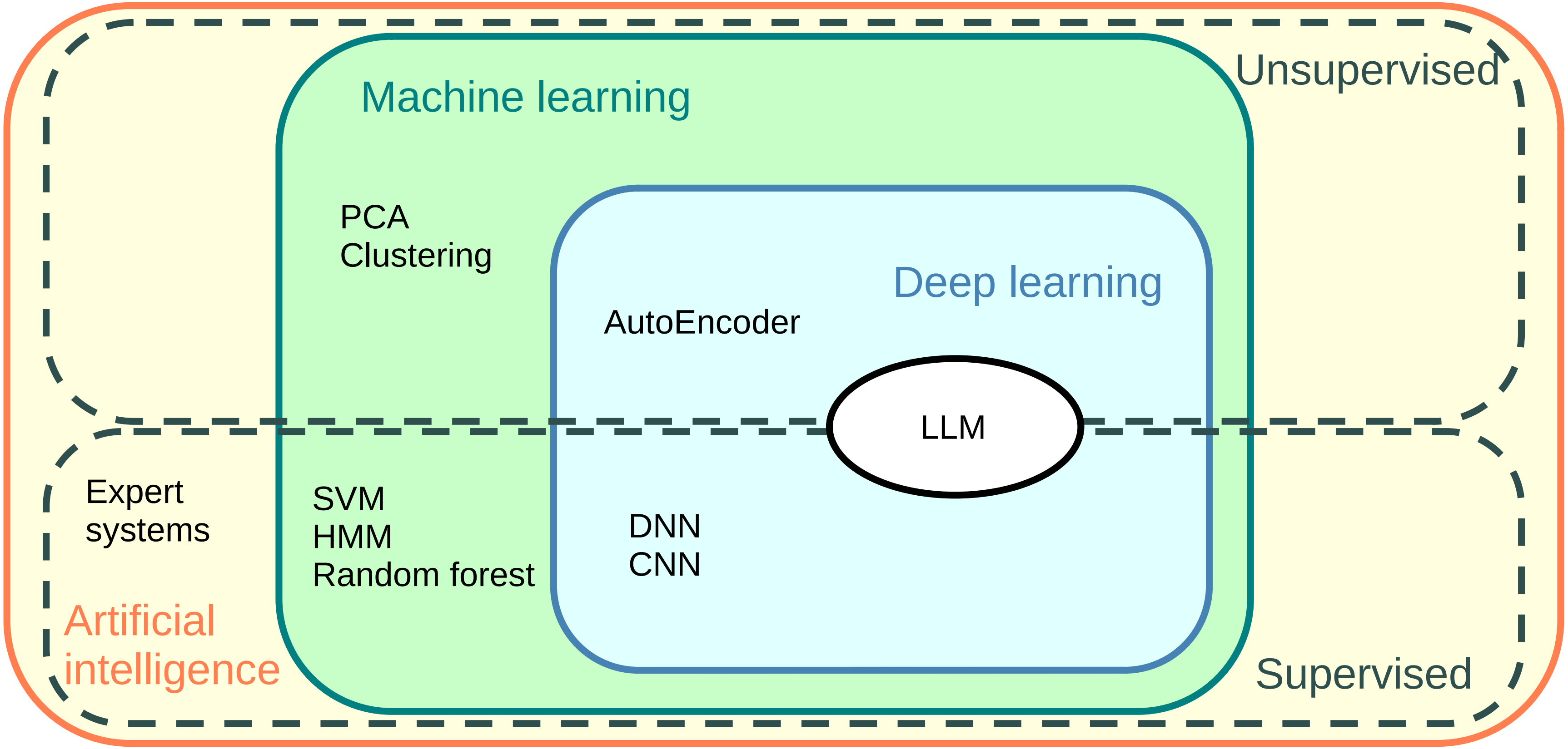


Some terminology

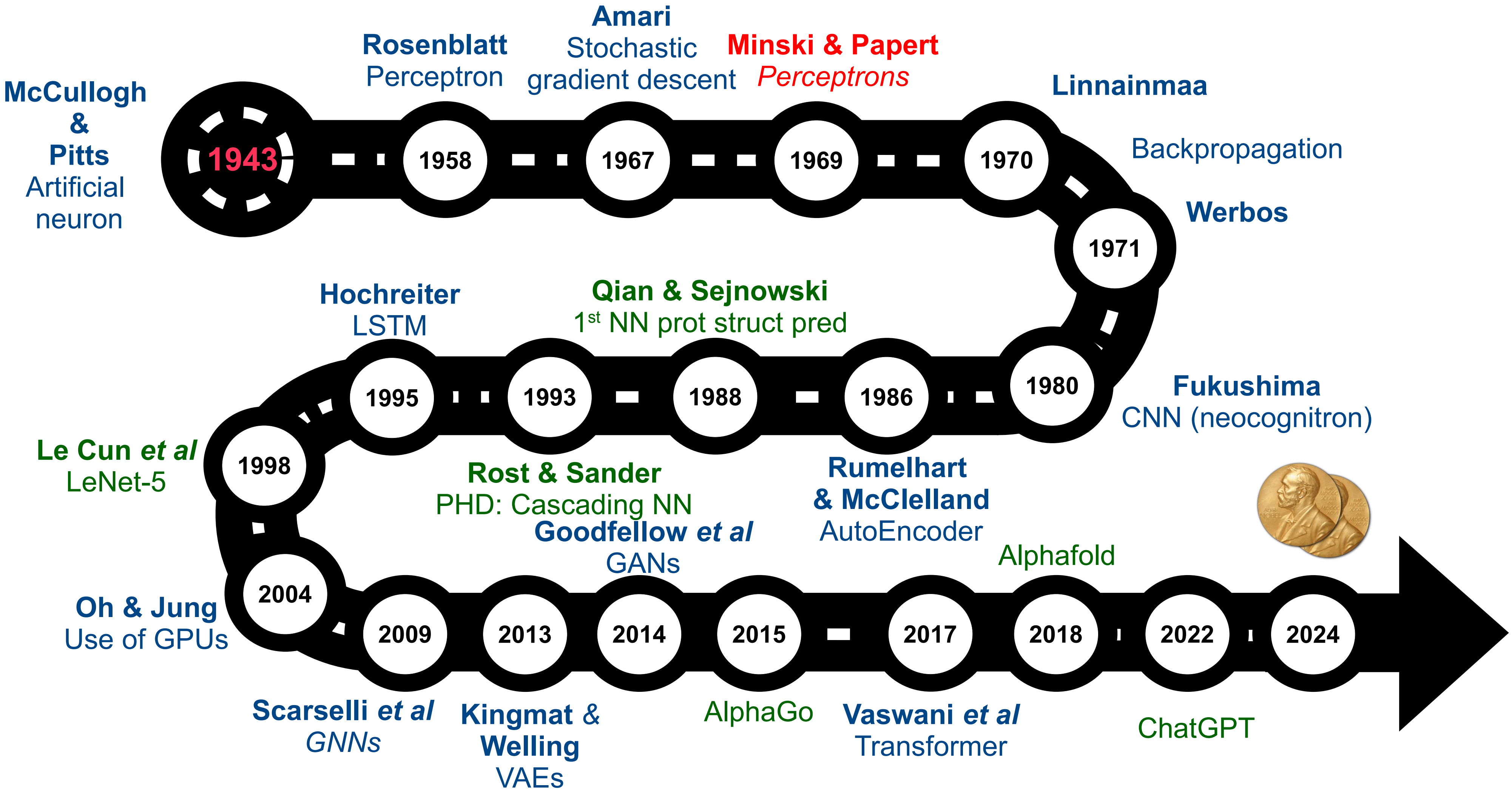


(NB: I consider reinforcement learning as part of supervised, but this is controversial)

Some terminology



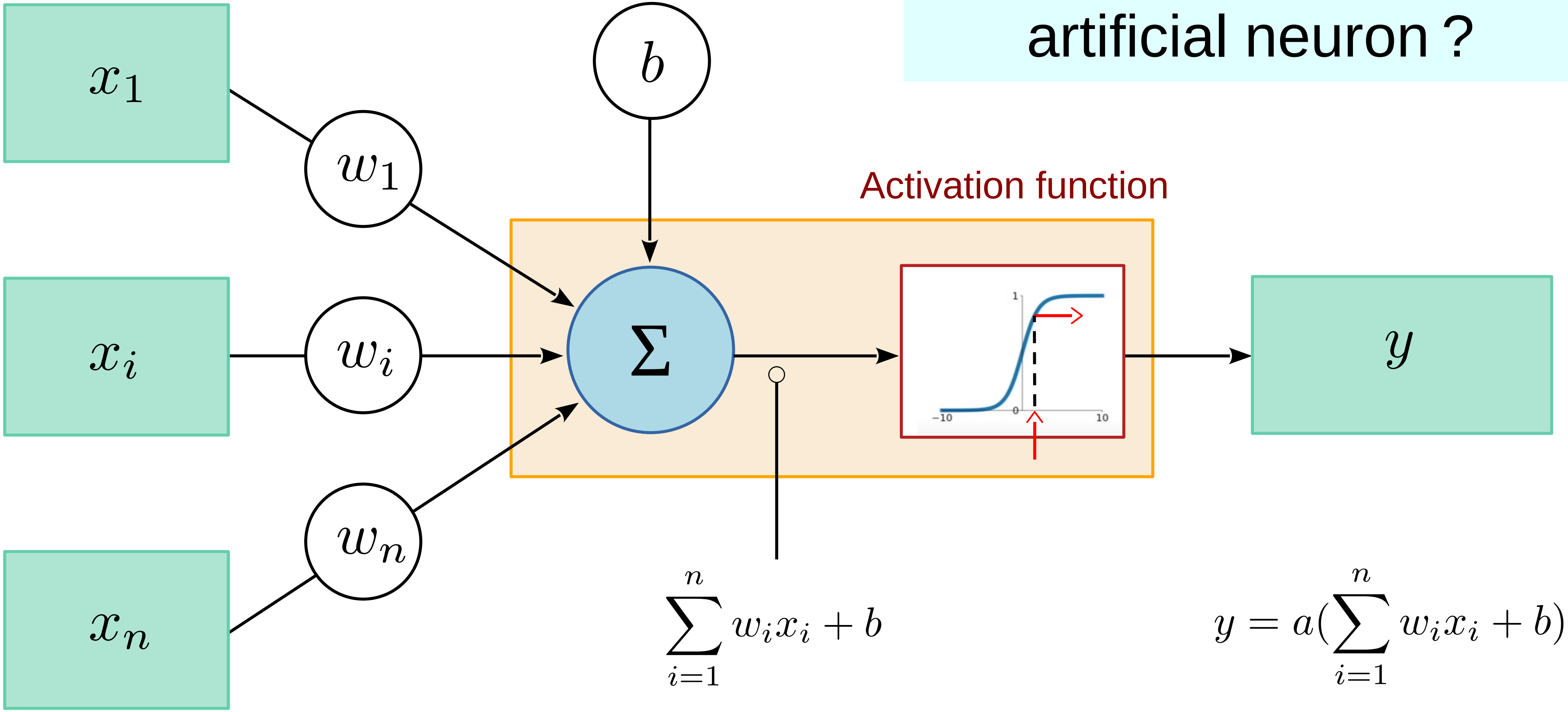
(NB: I consider reinforcement learning as part of supervised, but this is controversial)


McCullogh
&
Pitts
Artificial
neuron
1943
Rosenblatt
Perceptron
1958
Amari
Stochastic
gradient descent
1967
Minski & Papert
Perceptrons
1969
Linnainmaa
Backpropagation
1970
Werbos
1971
Fukushima
CNN (neocognitron)
1980
Rumelhart
& McClelland
AutoEncoder
1986
Qian & Sejnowski
1st NN prot struct pred
1988
Rost & Sander
PHD: Cascading NN
1993
Hochreiter
LSTM
1995
Le Cun *et al*
LeNet-5
1998
Oh & Jung
Use of GPUs
2004
Scarselli *et al*
GNNs
2009
Kingmat &
Welling
VAEs
2013
Goodfellow *et al*
GANs
2014
AlphaGo
2015
Vaswani *et al*
Transformer
2017
Alphafold
2018
ChatGPT
2022
2024

2

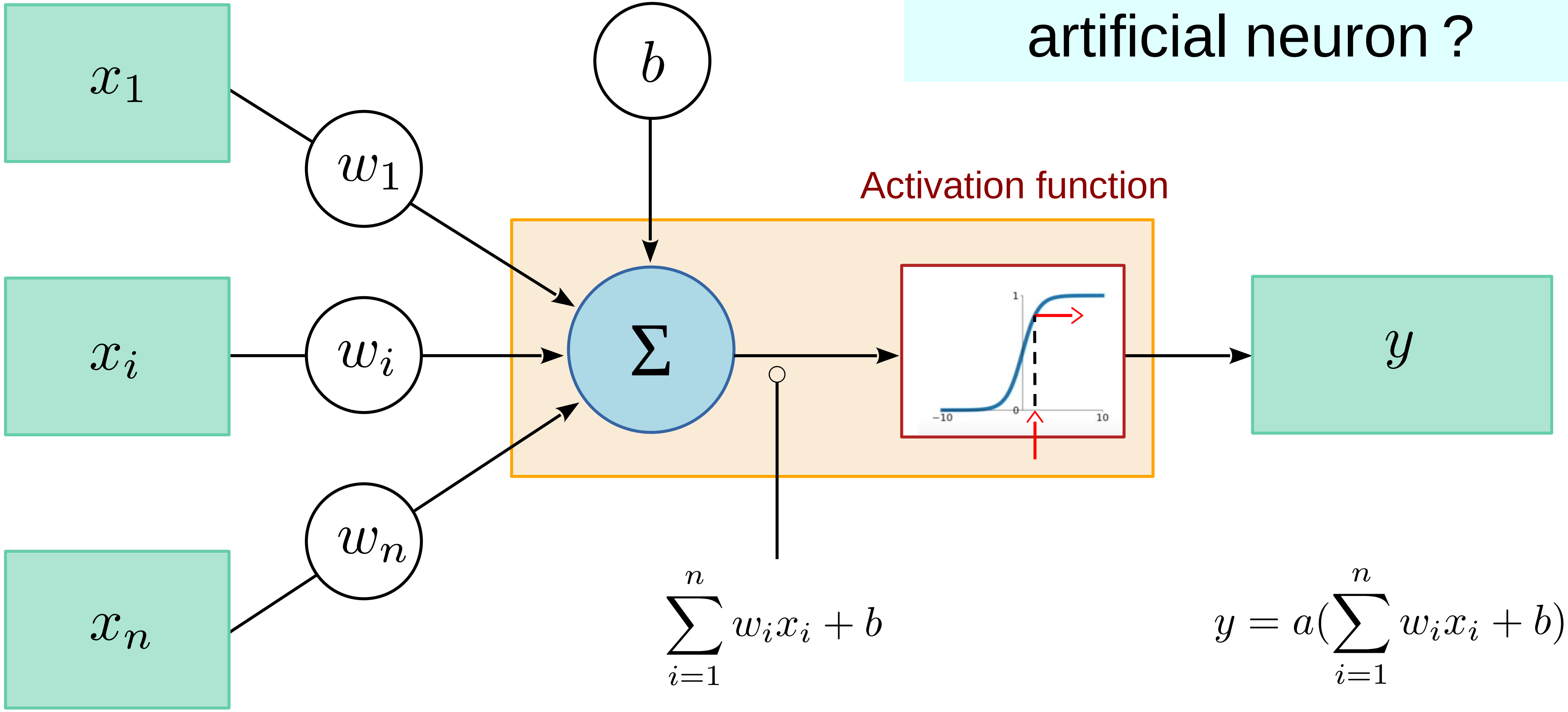
Multi-Layer Perceptrons (MLP)
a.k.a Fully Connected Networks (FCN)

What is an artificial neuron ?



McCulloch and Pitts (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115-133
Rosenblatt (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65(6):386-408
Widrow and Hoff (1960) Adaptive Switching circuits. *WESCON Convention record* part IV: 96-104

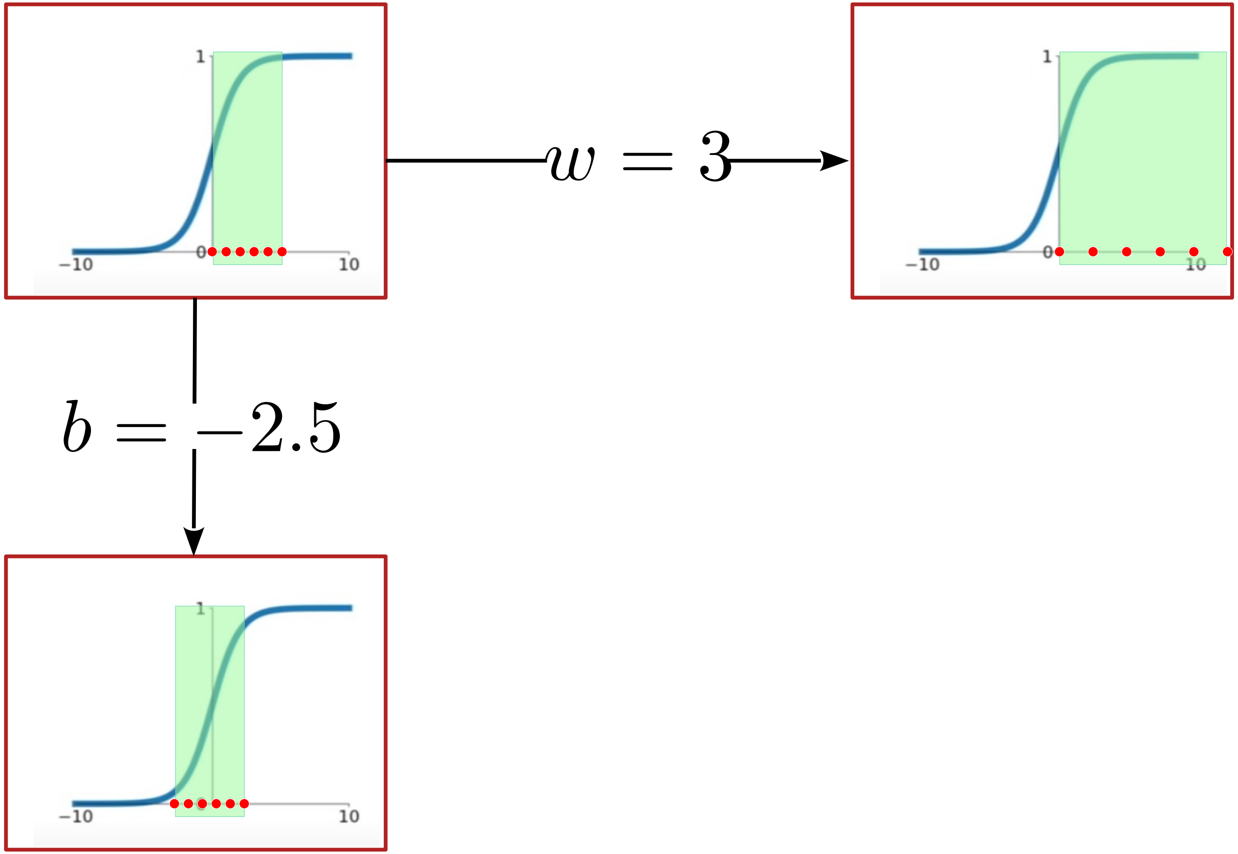
What is an artificial neuron ?



NB: when the activation function is logistic (sigmoid), this is actually a logistic regression...

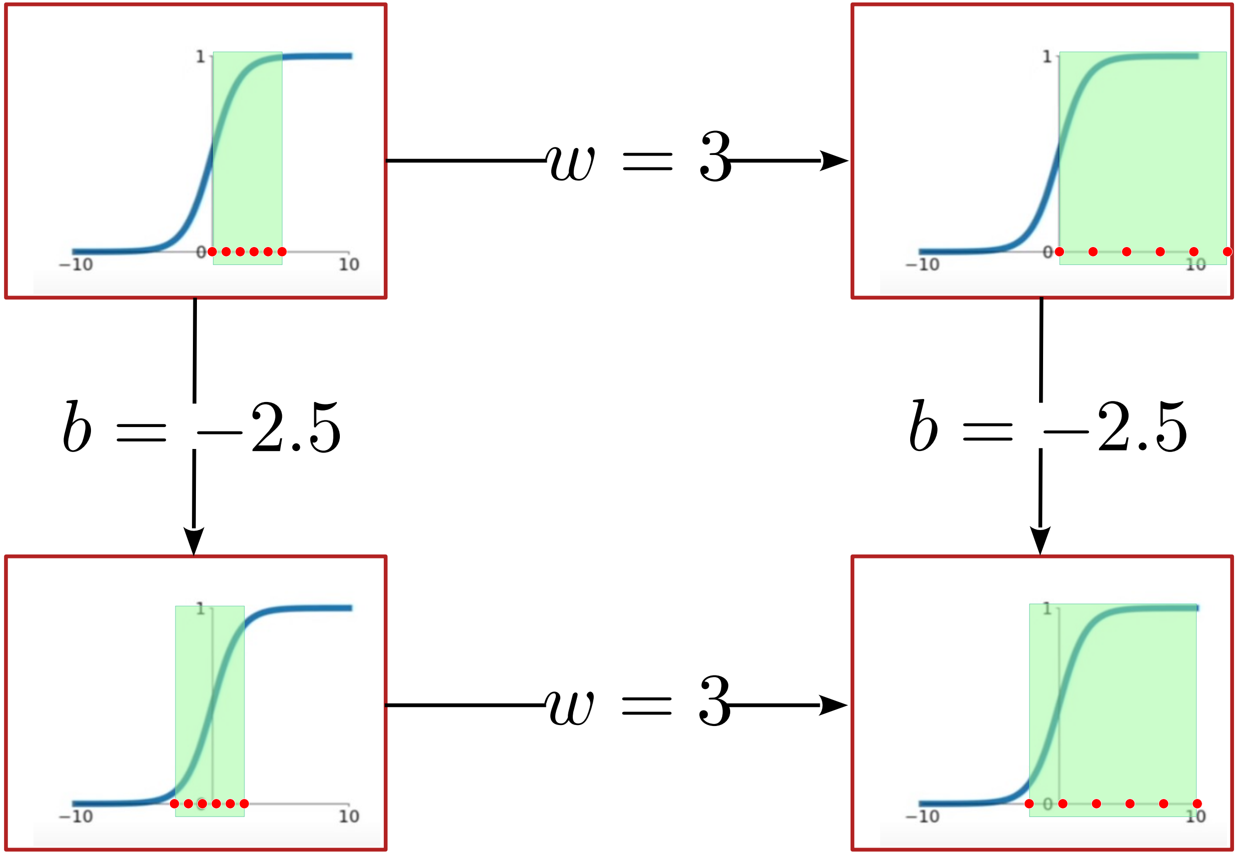
Impact of the weights and the bias

inputs = [0,1,2,3,4,5]

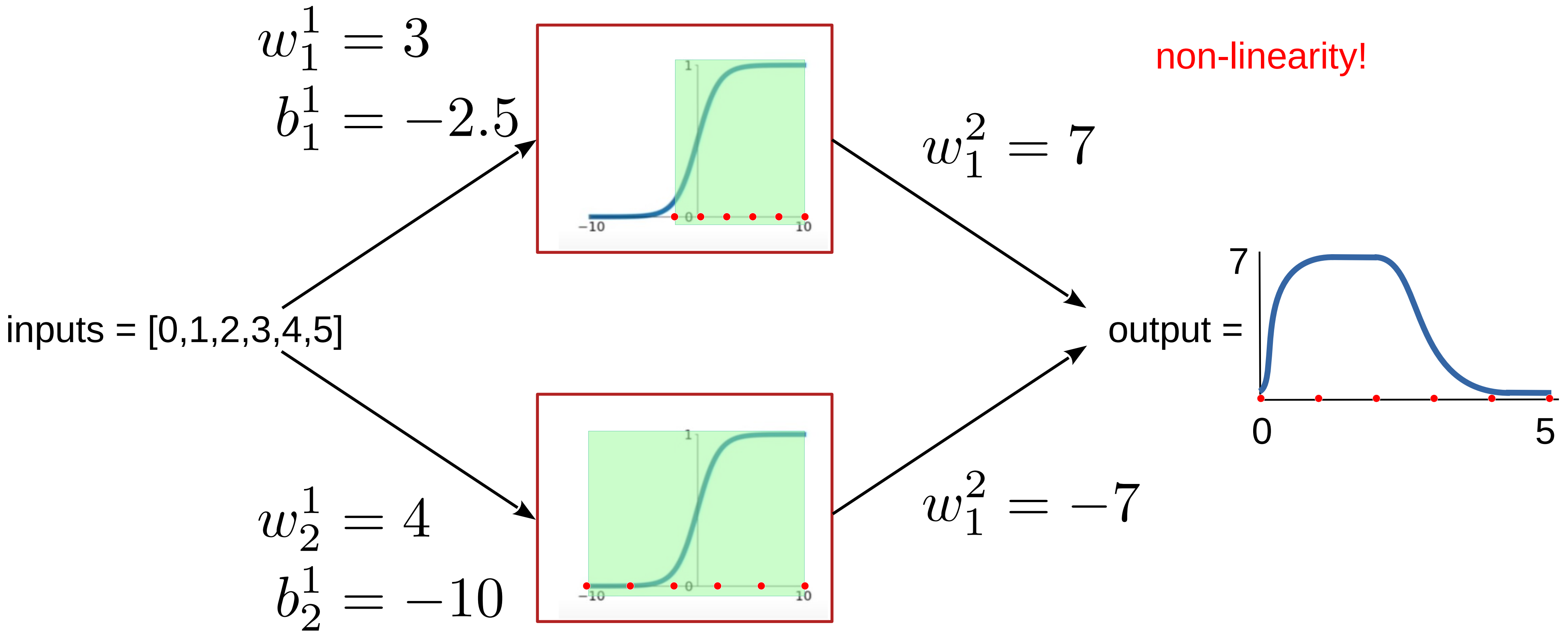


Impact of the weights and the bias

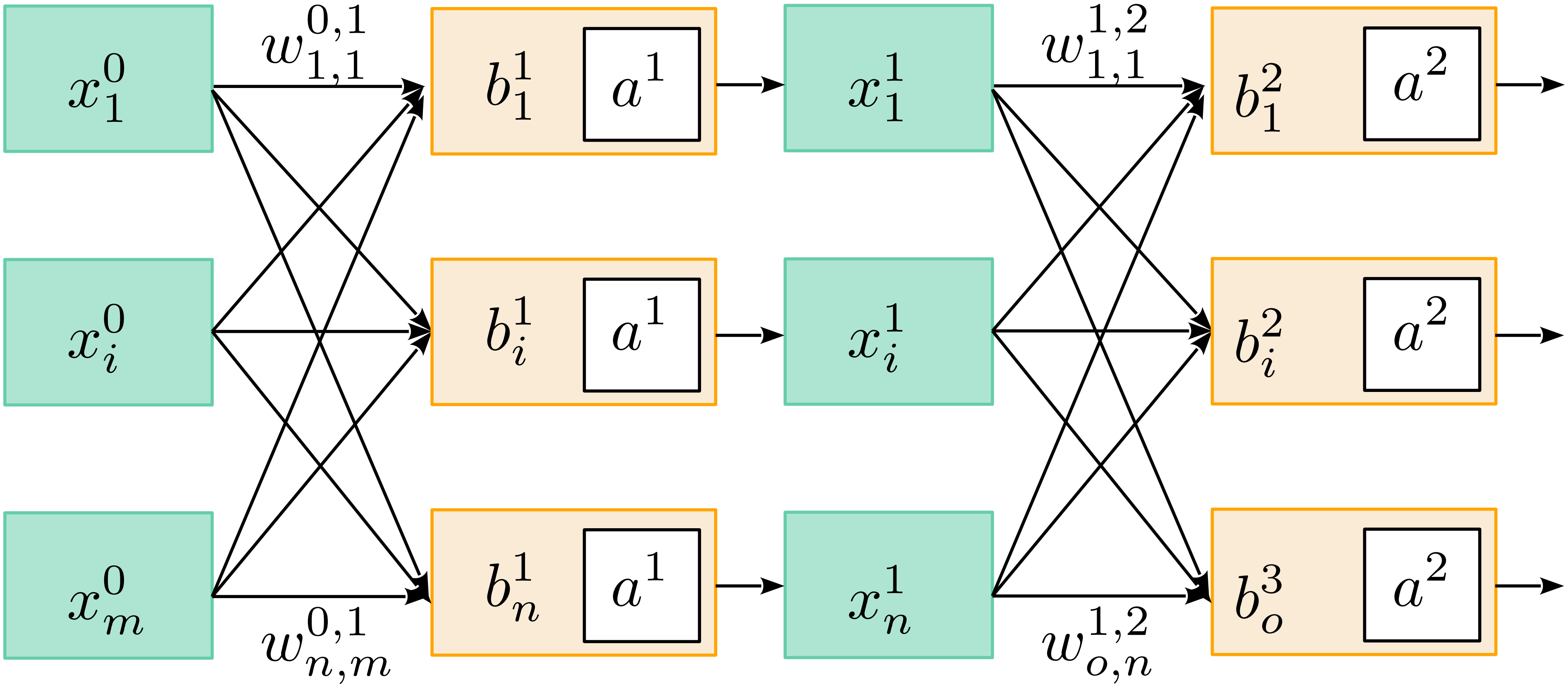
inputs = [0,1,2,3,4,5]



The magic happens with several neurons

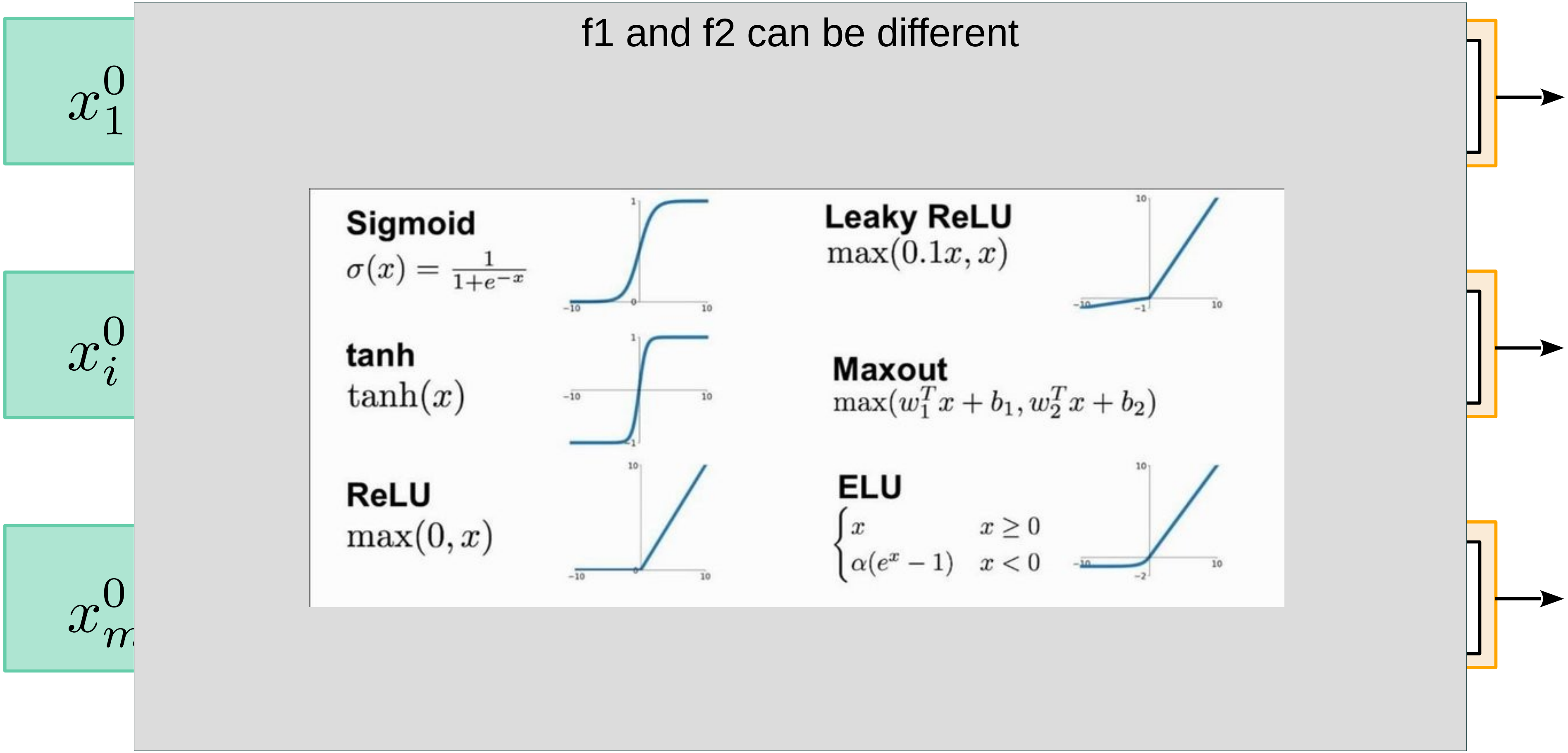


Then we add layers (the “Deep”)

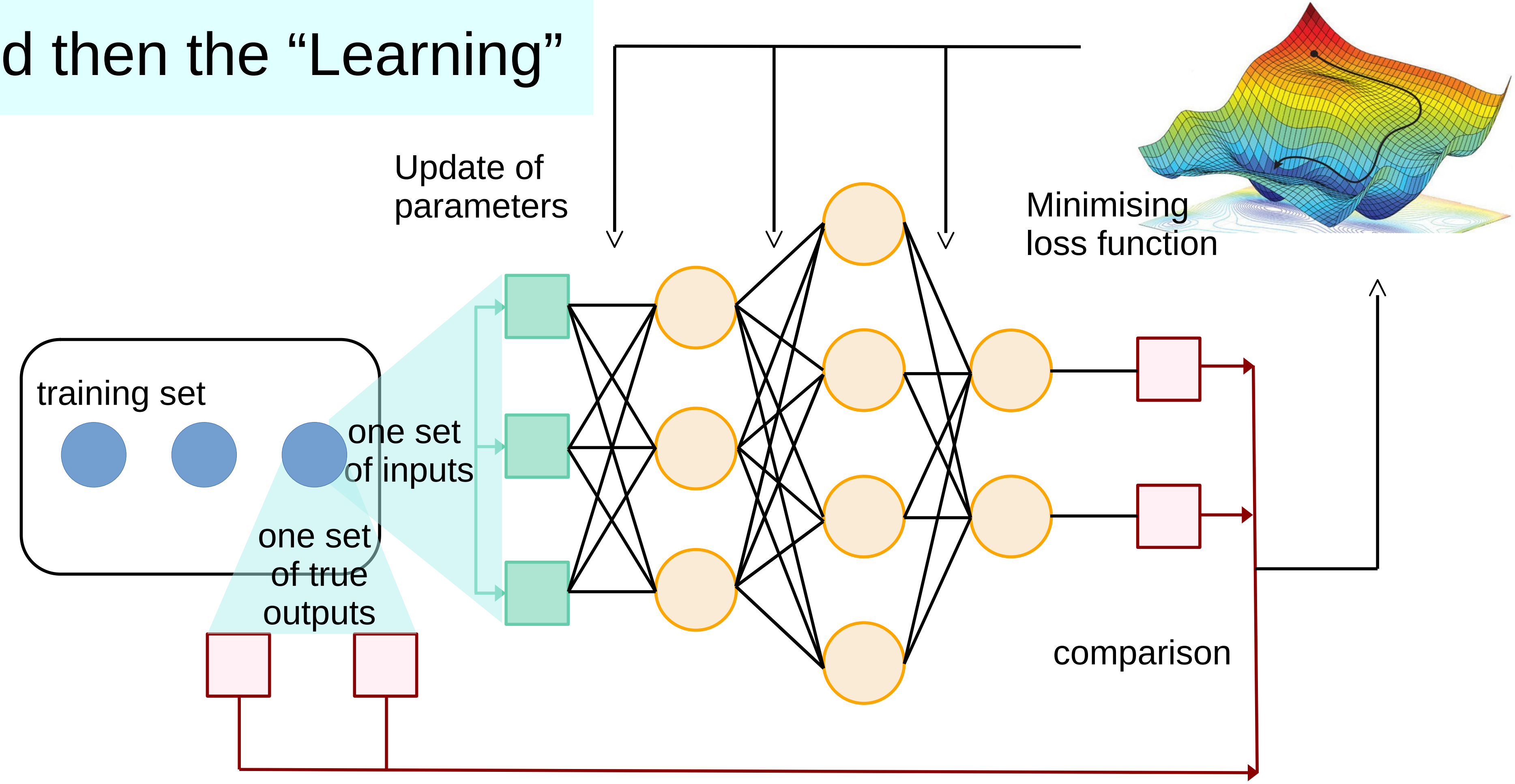


Rosenblatt (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65(6):386-408

And then we add layers (the "Deep")



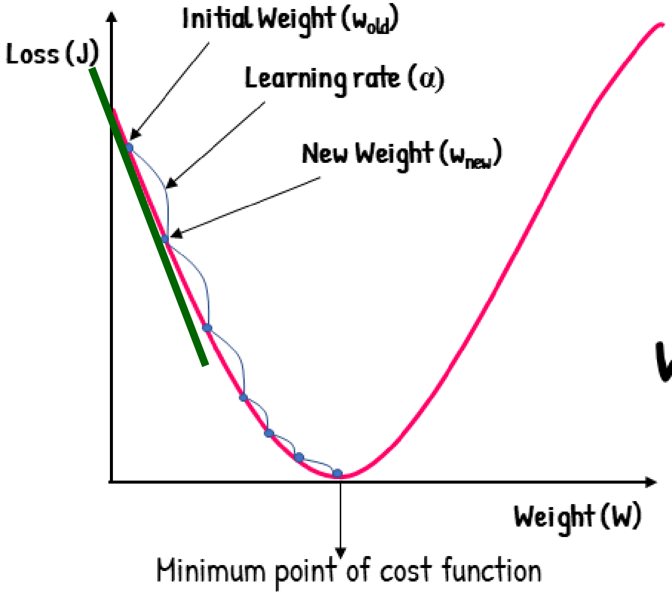
And then the "Learning"



Widrow and Hoff (1960) Adaptive Switching circuits. *WESCON Convention record* part IV: 96-104
S Amari (1967). A theory of adaptive pattern classifier. *IEEE Transactions*. EC (16): 279–307
S Linnainmaa (1970-1976). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors (Masters). University of Helsinki. p. 6–7.
P Werbos (1971-1982) Applications of advances in non-linear sensitivity analysis. *LNCIS* 38: 762-770

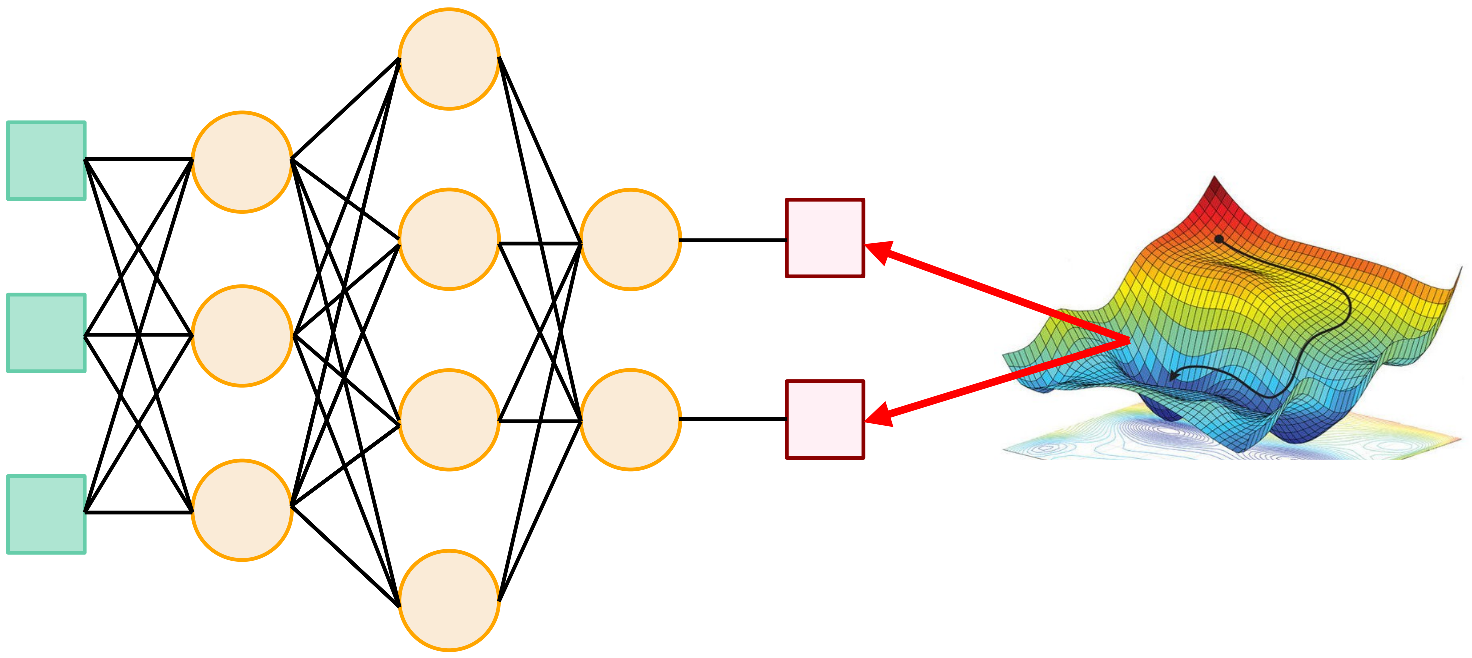
Optimization, e.g., gradient descent

To minimise the loss function (called L, C, or most often J), we will calculate the “gradient” – the derivative – of the loss function with a set of parameters and calculate a new set using this gradient and the *learning rate*.

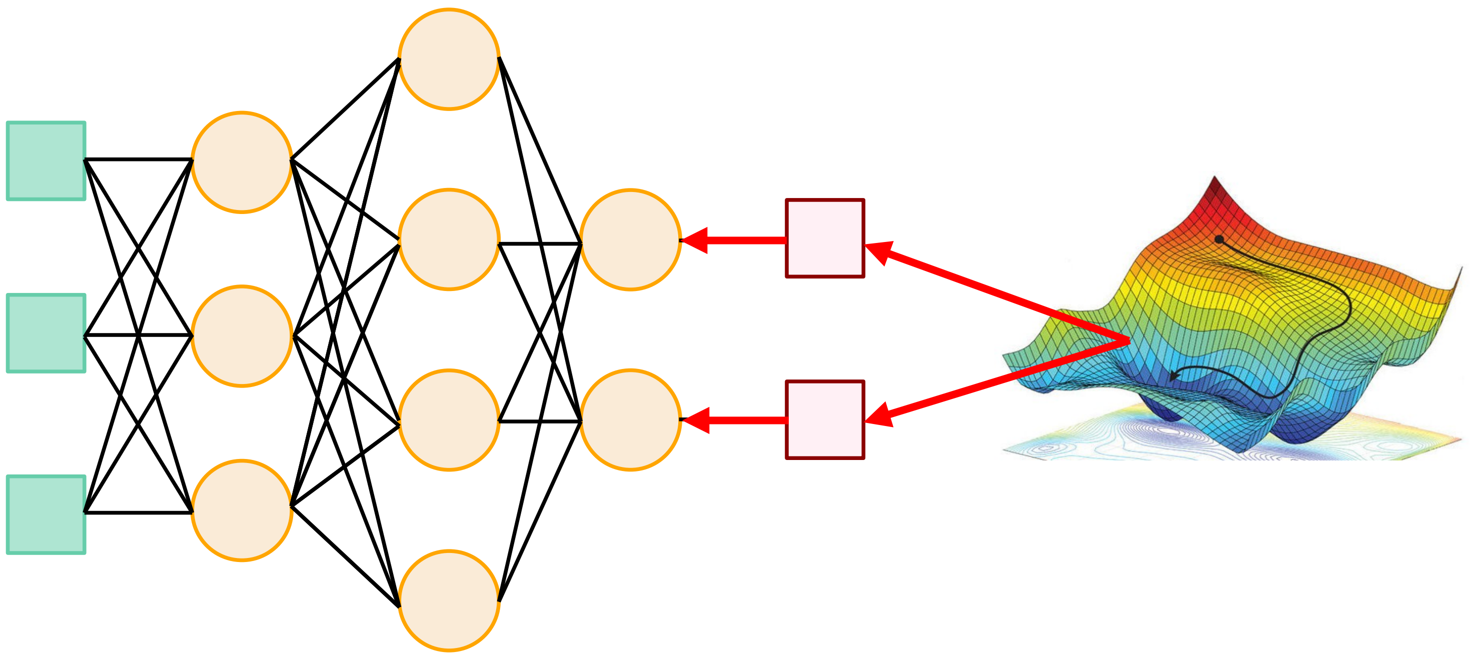


$$w_{new} = w_{old} - \alpha \frac{\delta J}{\delta w}$$

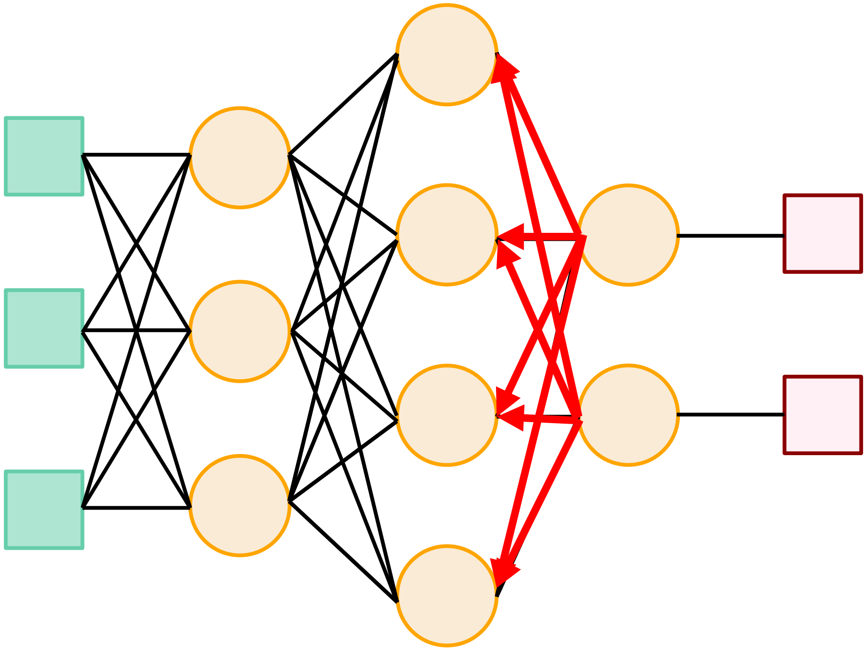
Error backpropagation one layer at a time

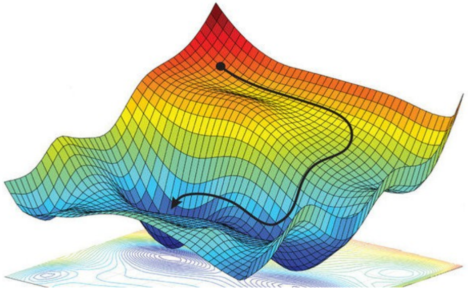


Error backpropagation one layer at a time

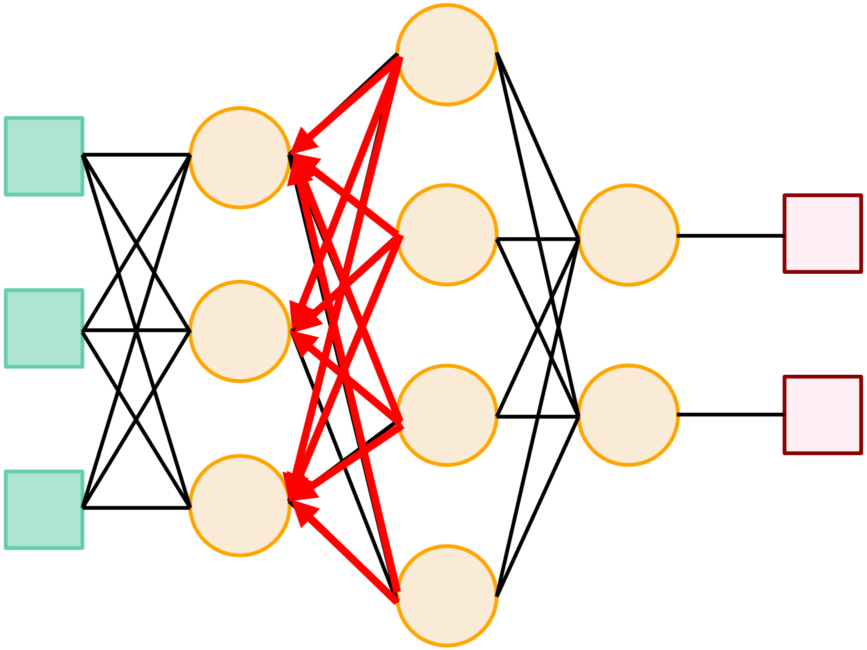


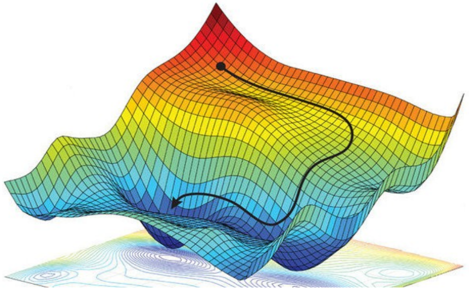
Error backpropagation one layer at a time



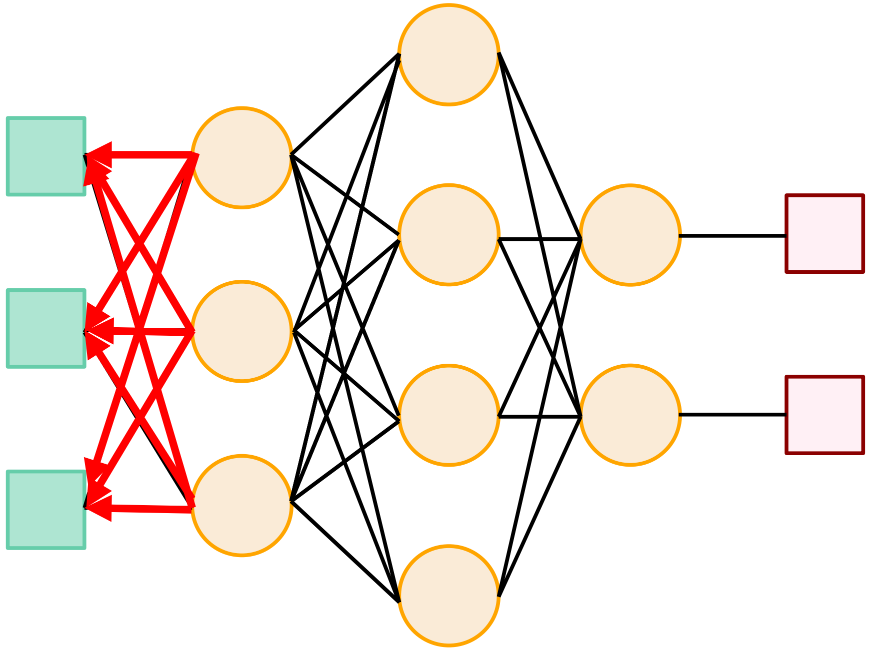


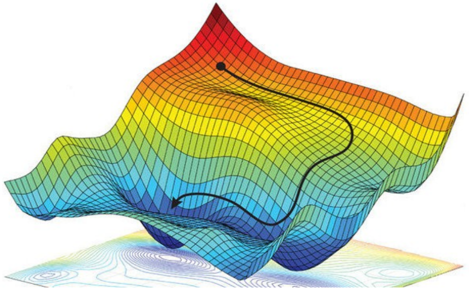
Error backpropagation one layer at a time



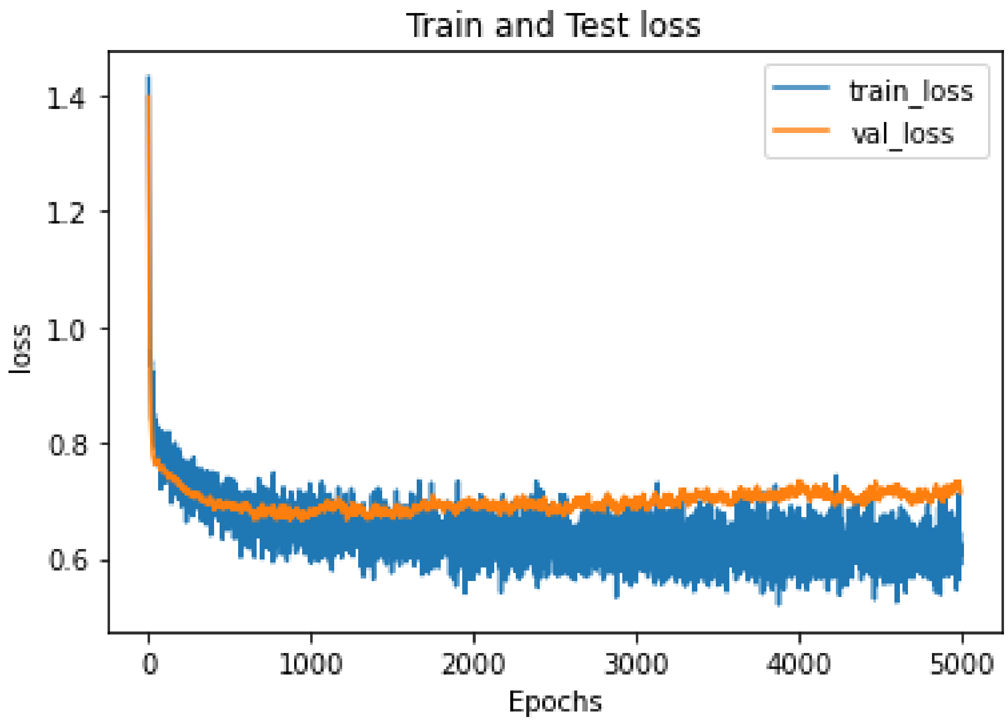


Error backpropagation one layer at a time





Learning



Train and Test loss
train_loss
val_loss
loss
1.4
1.2
1.0
0.8
0.6
0
1000
2000
3000
4000
5000
Epochs

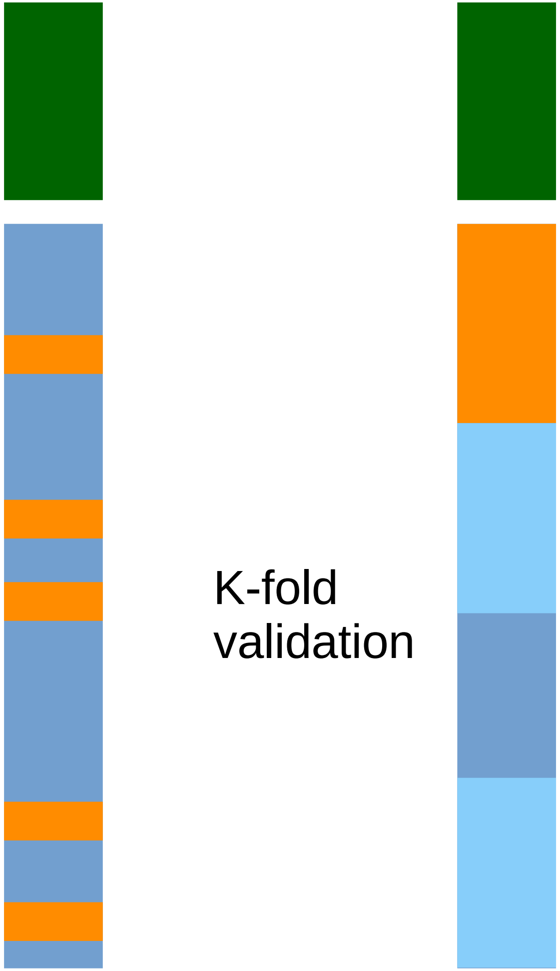
Training, testing, and validation sets

“validation” (never seen)
Same for all model instances
Used to assess the model at the end

Training set: used to learn

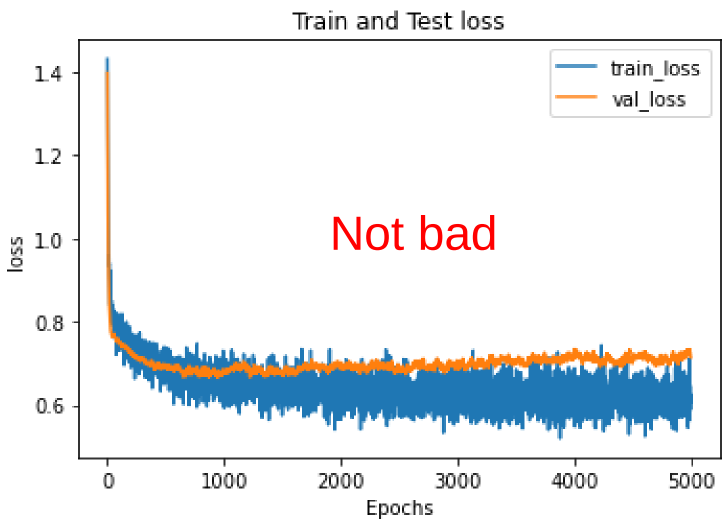
(training + test set = learning set)

“test” set: used to assess the model
during the learning phase
Different for each model instance

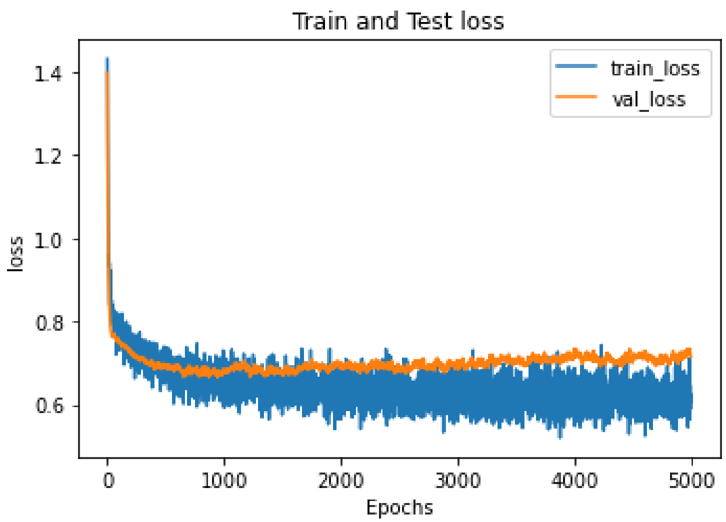


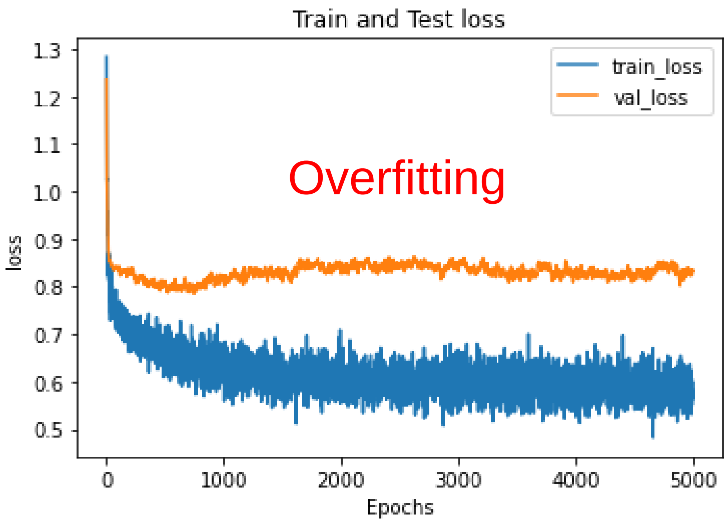
Beware: “validation” and “test” are used the other way around a lot in deep learning, at the opposite of all other fields of machine learning, or even life science in general

Learning and overfitting

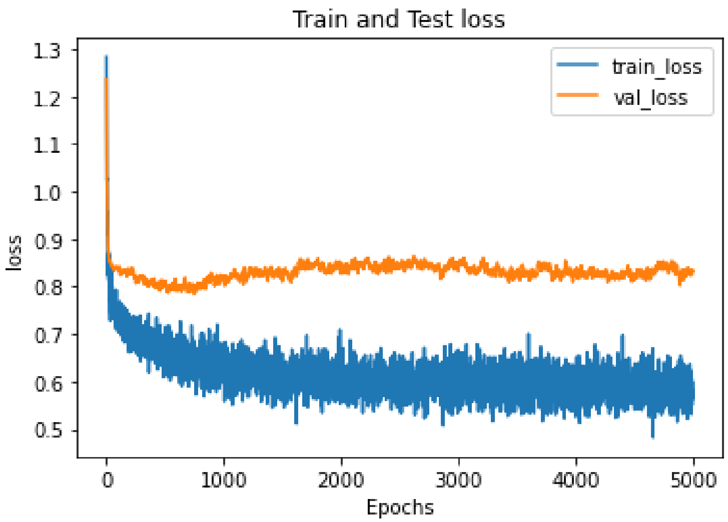


Learning and overfitting

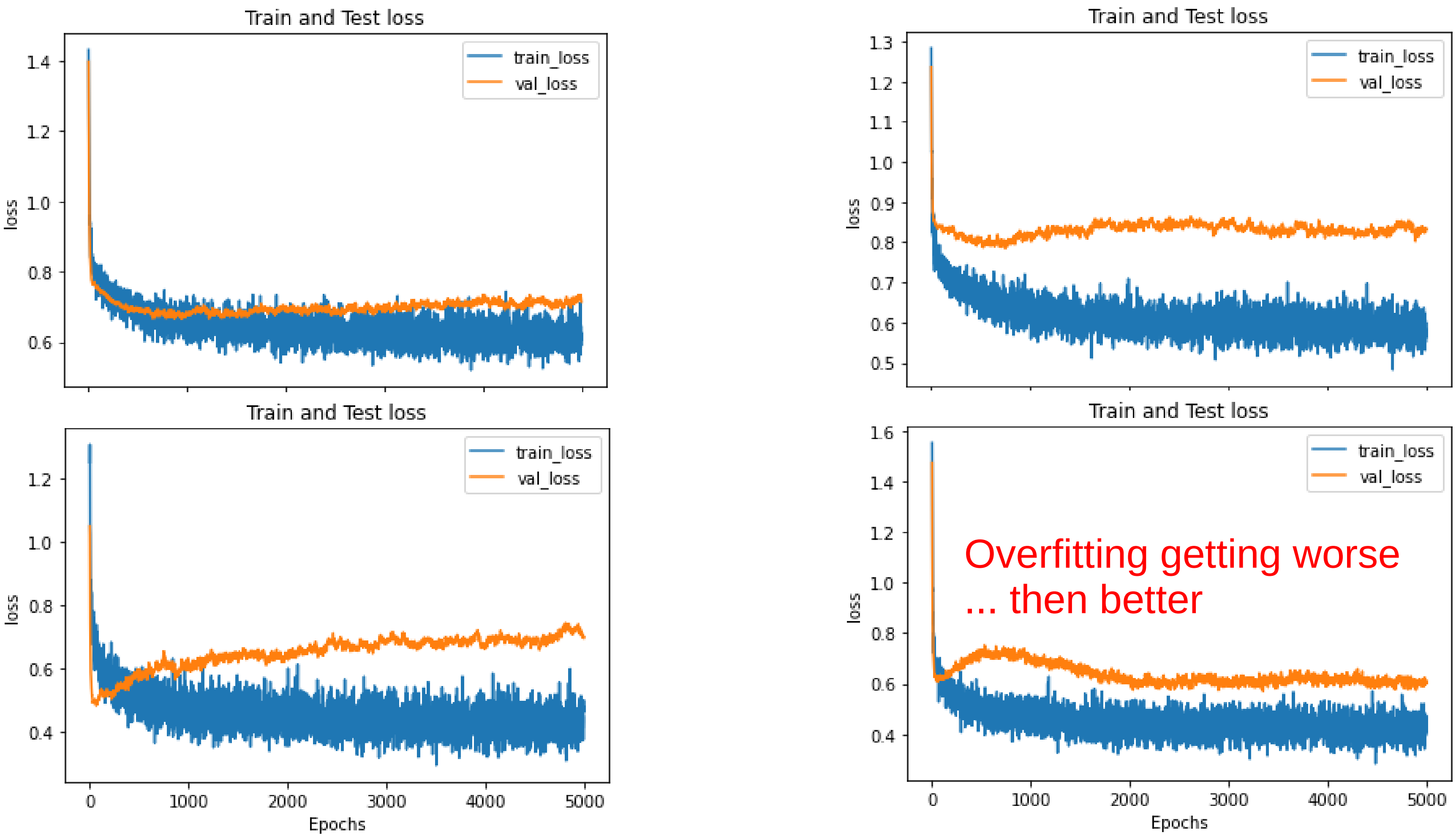




Learning and overfitting



Learning and overfitting



Why a test AND a validation set?

Underperforming on the test set means the model overfitted
the training set, the parameters are too specific of the training samples.
This overfitting is learned.

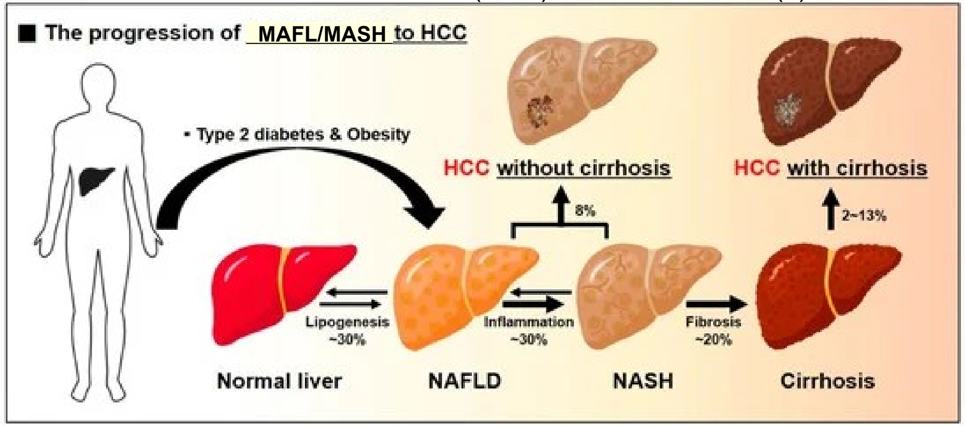
Underperforming on the validation set means the hyperparameters are
too specific of the test set as well! When you modifies the structure
of the model to avoid overfitting, you actually made the model overfit
the entire learning set. YOU biased the model.
This overfitting is built-in.

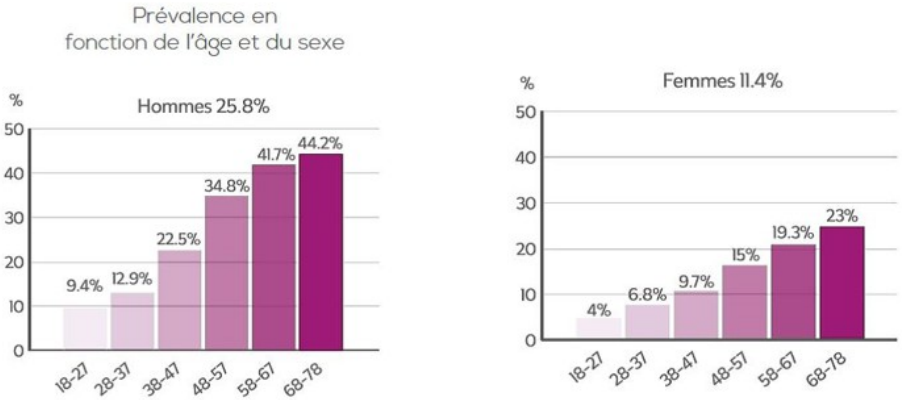
3

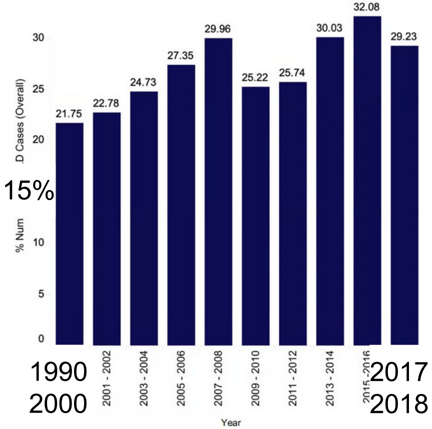
MLPs in action
Multi-omics real-world example

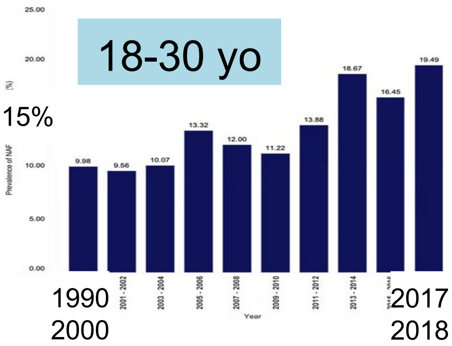
Let’s try to recognise a disease severity

Kim *et al* (2021) *Int. J. Mol. Sci.*, 22(9): 4495



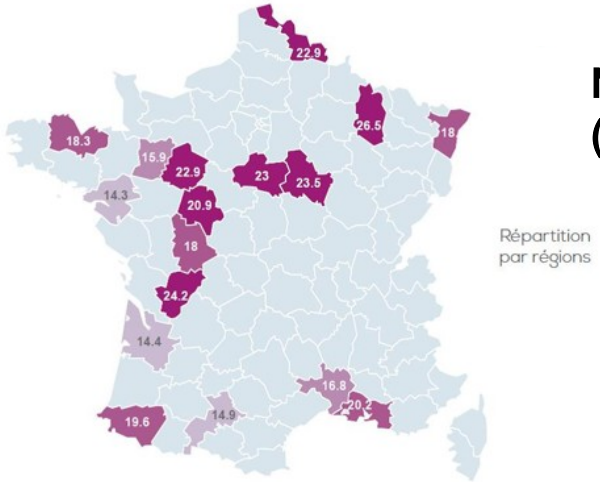






Kim *et al* (2022) *Met. Target Organ Damage*, 2: 19

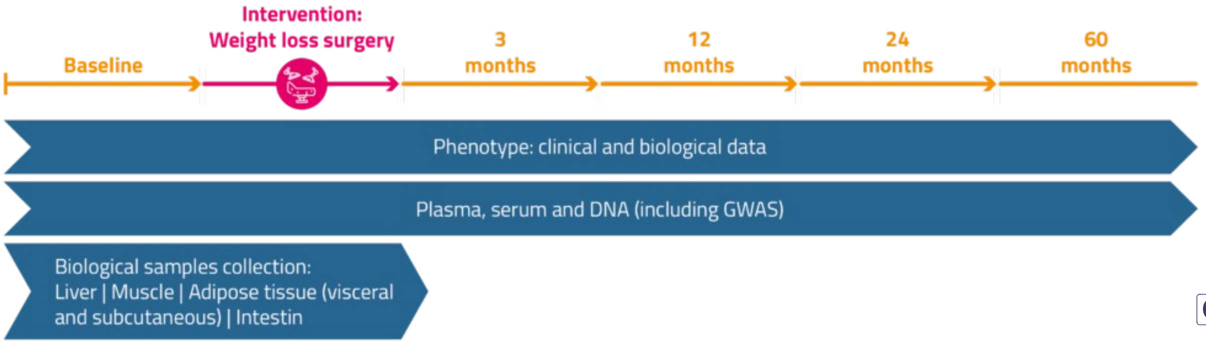
NALFD (now MASLD)



NASH (now MASH)

Paris MASH Meeting (11-12 juillet 2019)

Cohort

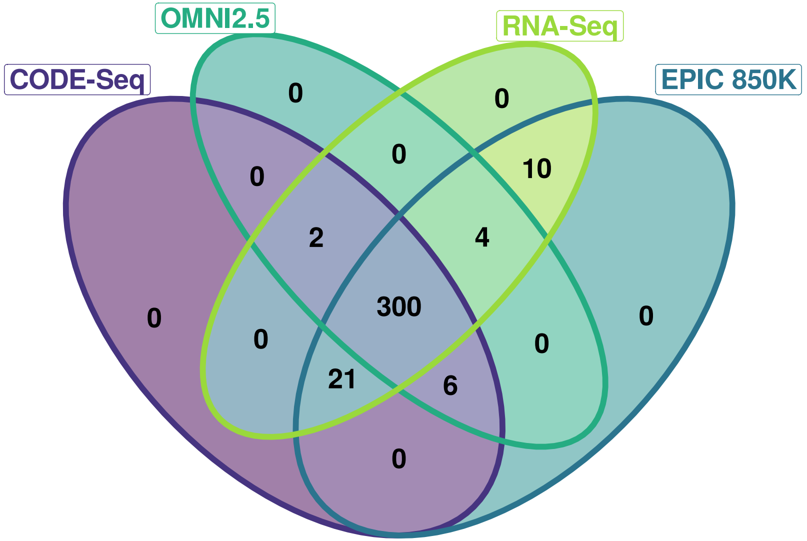


ABOS (Biological Atlas of Severe Obesity)

All subjects had bariatric surgery

PreciNASH project

ABOS subset: Only European ancestry and unrelated individuals

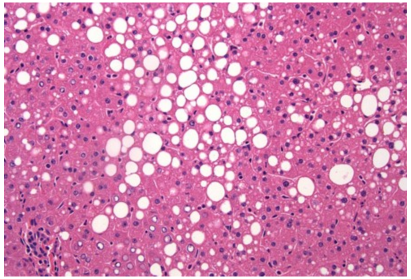


(+66 clinical and personal data
+1076 identified metabolites
in blood and liver)

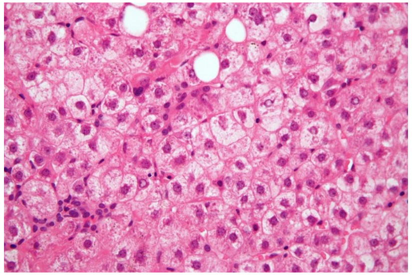
Subject grouping

Scoring on liver biopsy with the method from Kleiner and Brunt 2005

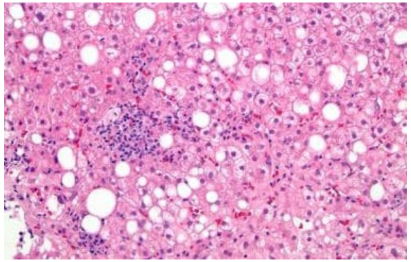
Steatosis
Categorical [0-3] from
quantitative measurement



Ballooning
Categorical [0-2]
= {none, some, much}



Inflammation
Categorical [0-3] from
number of foci



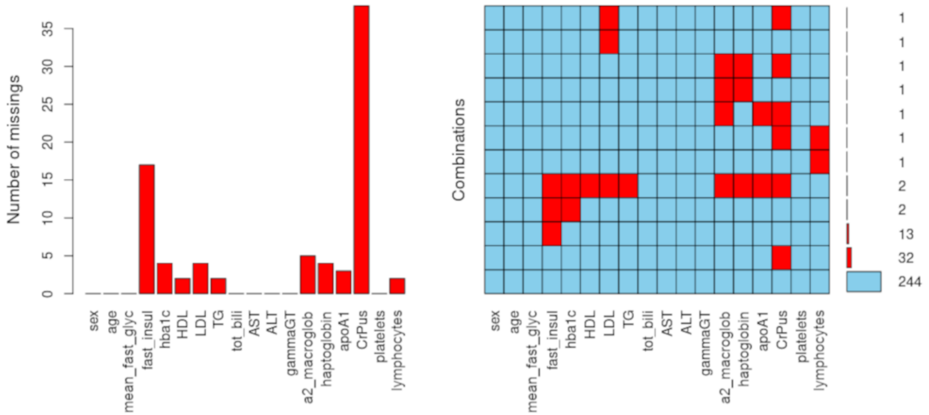
Final score:

Healthy: $S = 0, B = 0, I = 0$ $n = 80$

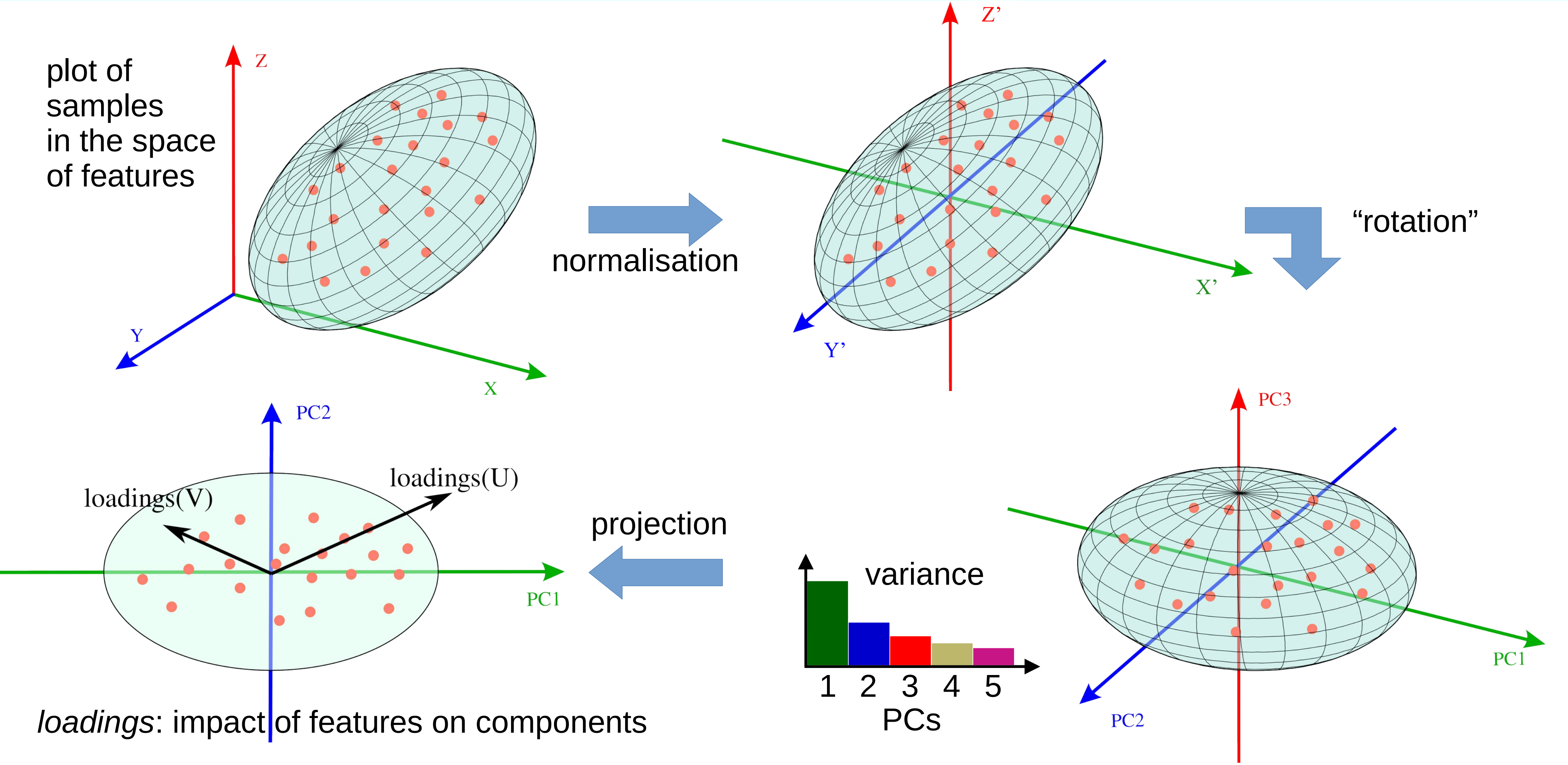
NAFL: $S > 1, B = 0, I \geq 1$ $n = 137$
$S > 1, B > 1, I = 0$

NASH: $S > 0, B > 0, I > 0$ $n = 83$

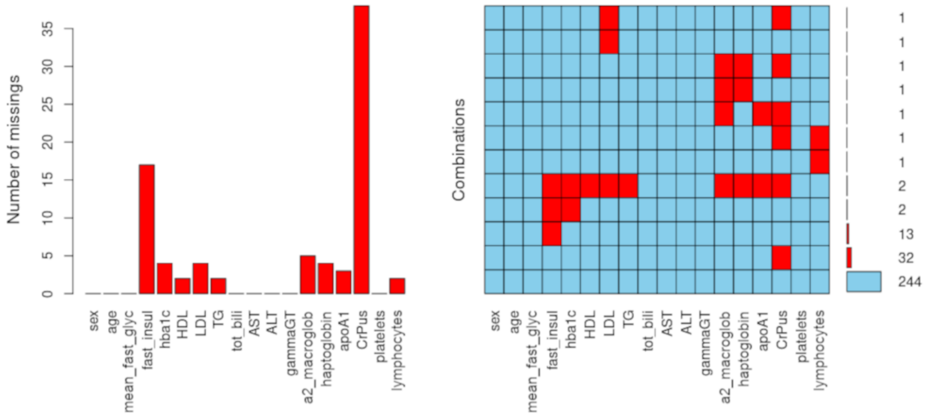
Clinical data

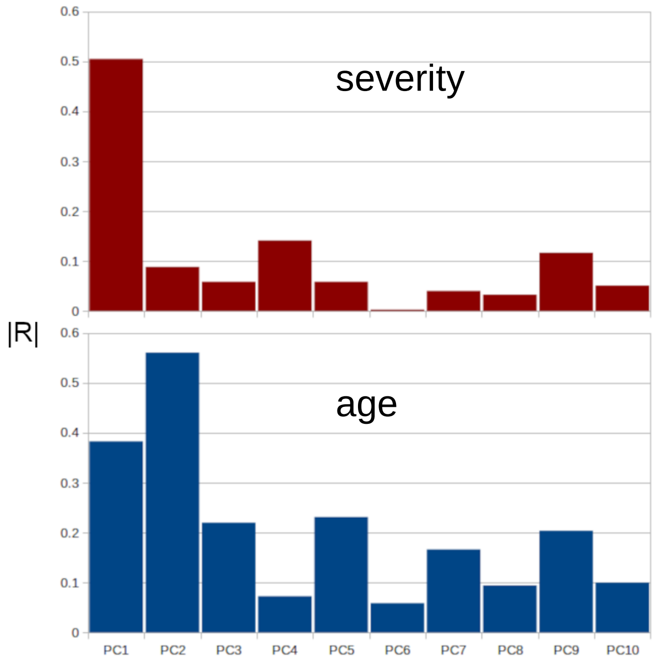


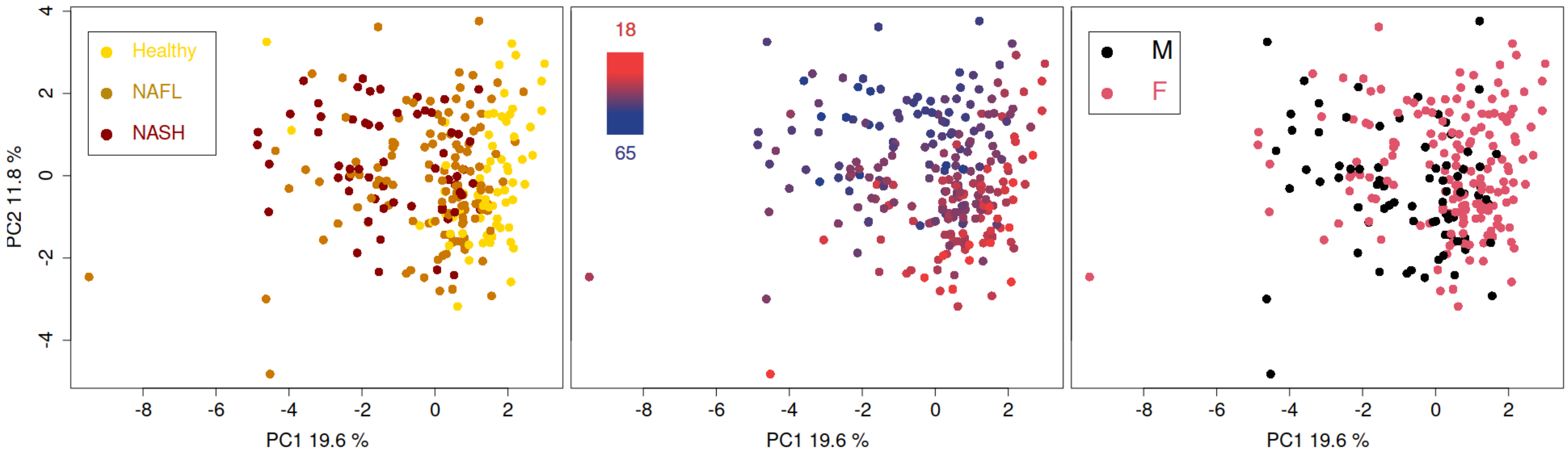
Principal component analysis (PCA)



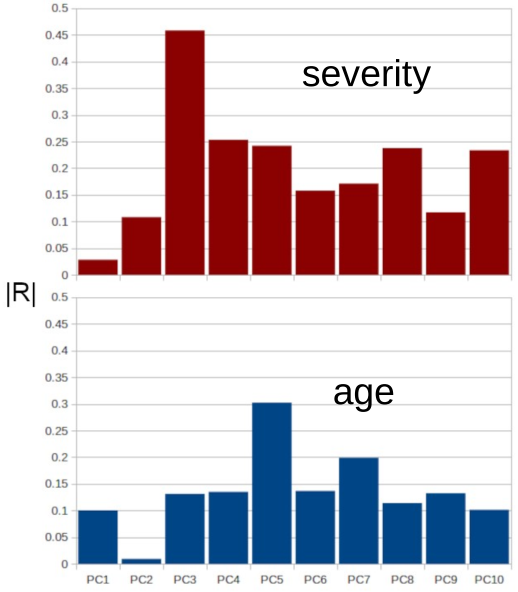
Clinical data

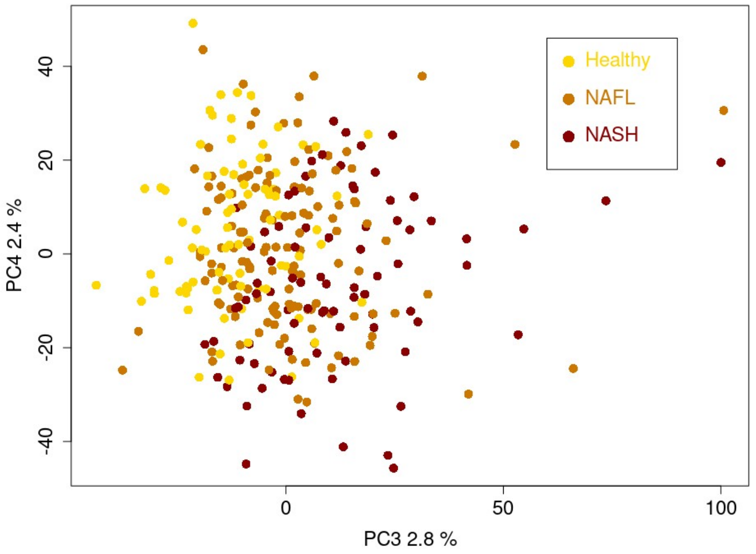




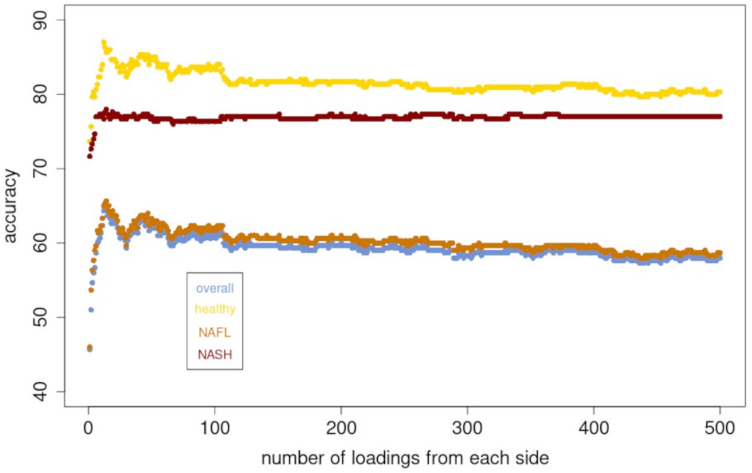


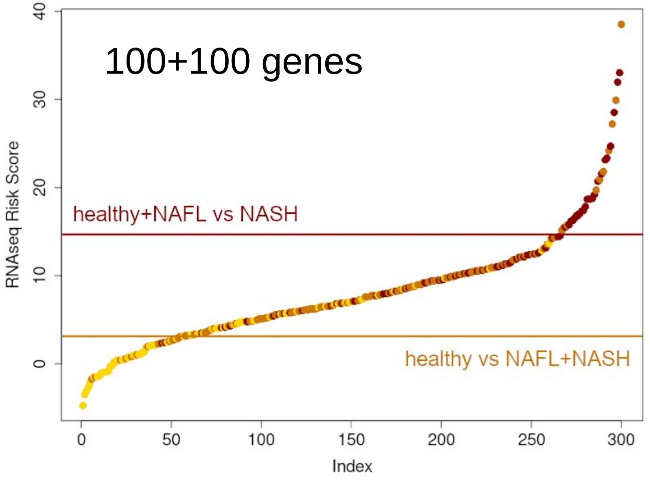
RNAseq





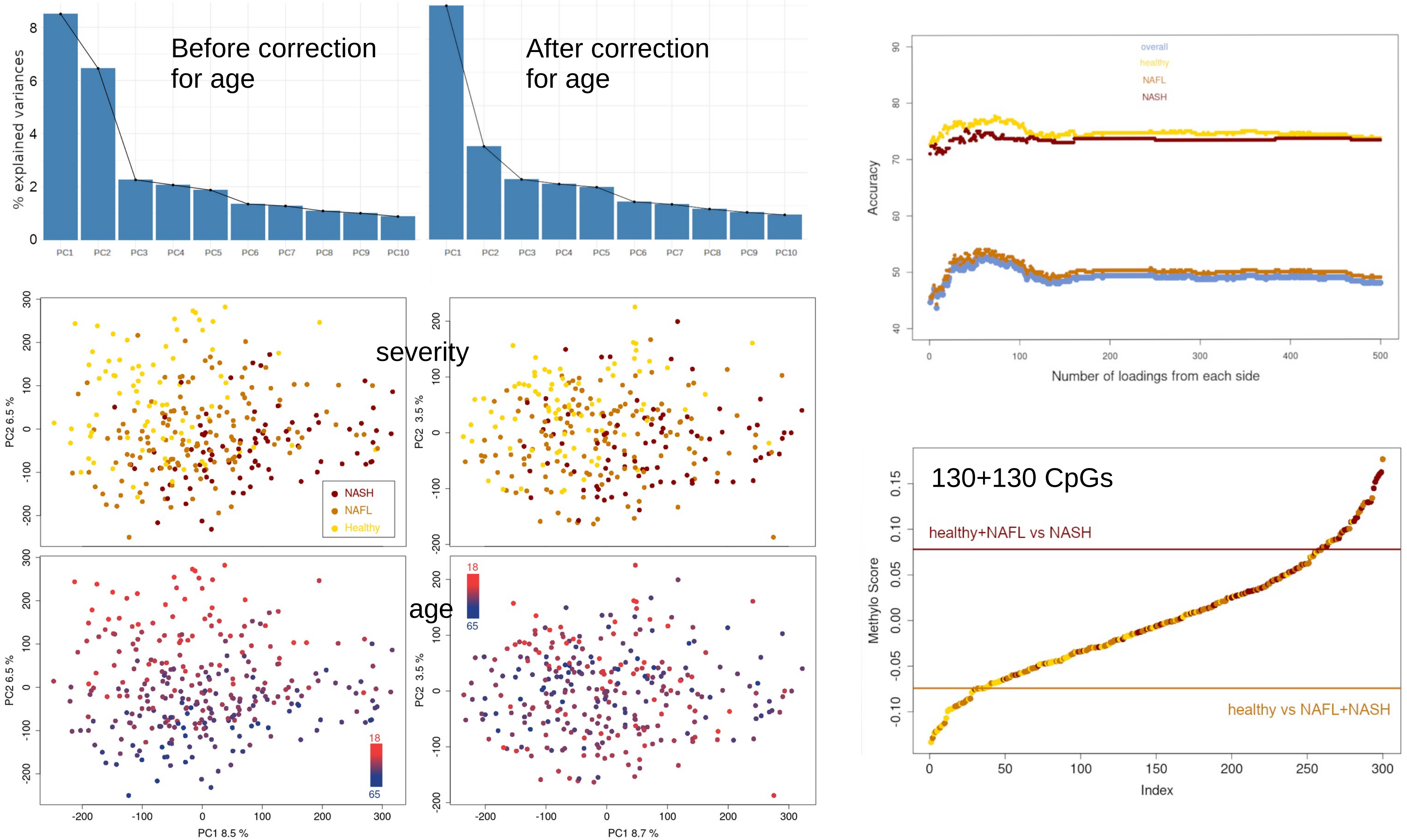
Score based on gene expression and gene “loadings” (impact of a gene on a given principal component)

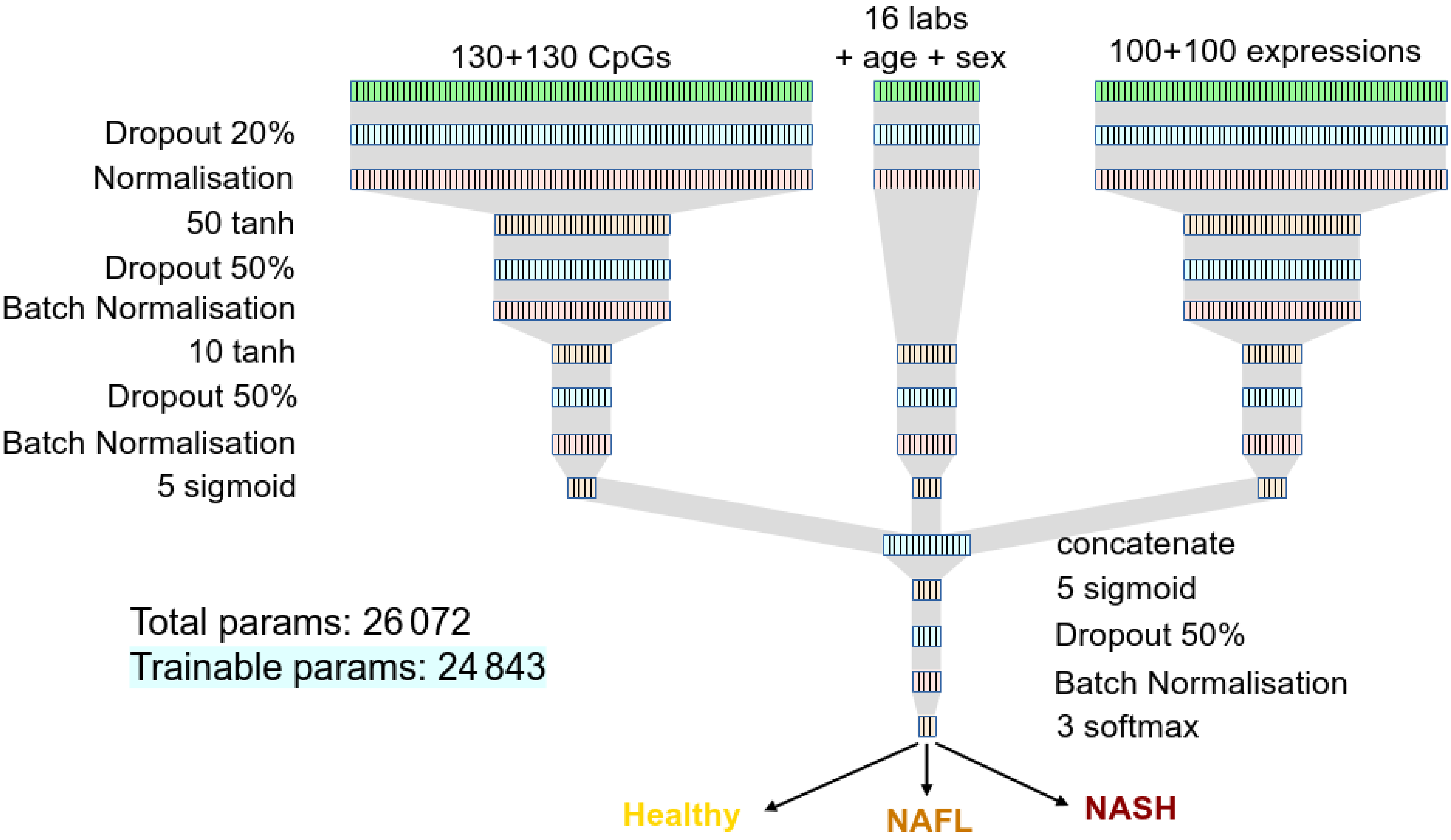




Logistic regression to find the thresholds best separating the severity groups

DNA Methylation

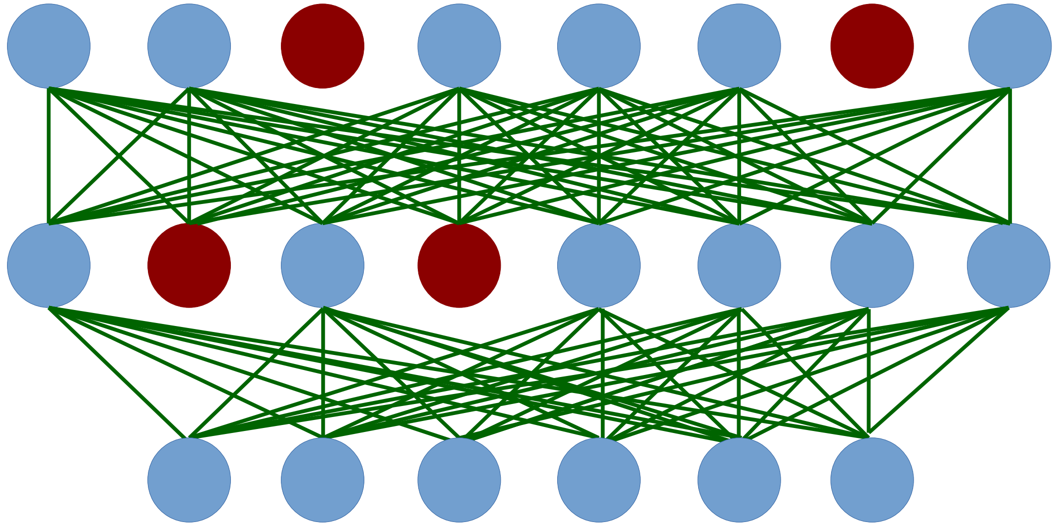



130+130 CpGs
16 labs
+ age + sex
100+100 expressions
Dropout 20%
Normalisation
50 tanh
Dropout 50%
Batch Normalisation
10 tanh
Dropout 50%
Batch Normalisation
5 sigmoid
concatenate
5 sigmoid
Dropout 50%
Batch Normalisation
3 softmax
Total params: 26 072
Trainable params: 24 843
Healthy
NAFL
NASH

Dropout

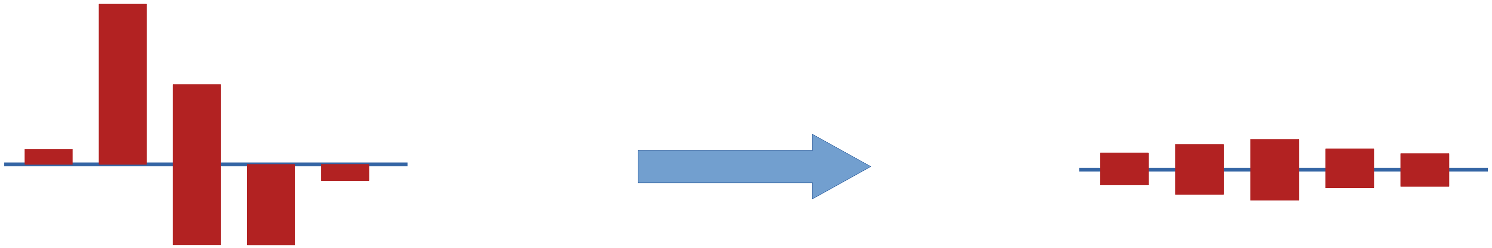
1) Purpose: avoiding overfitting

2) Disable some connections at random (set the weights at 0)



Srivastava et al (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. JMLR 15(56):1929−1958

Normalisation



1) Normalisation during data processing

2) Normalisation before training: on the whole dataset

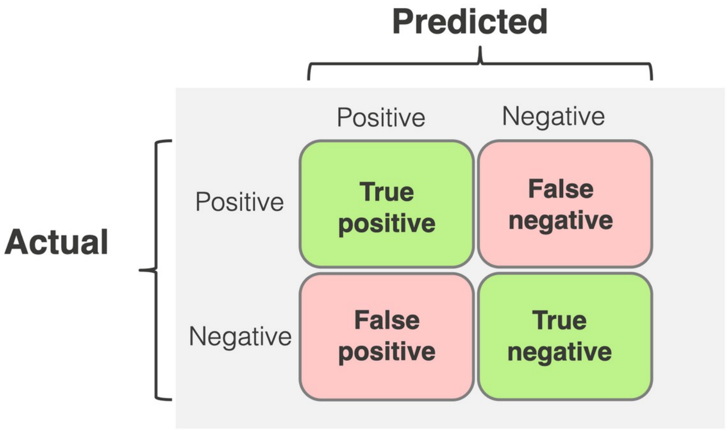
3) Normalisation during training: after dropout

4) Batch normalisation: normalisation on the current batch
(not on the entire dataset)

5) Layer normalisation: normalisation of the input of a layer

Ba, Kiros, Hinton (2016) Layer Normalization. arXiv:1607.06450v1
Ioffe and Czegedy (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proc 32nd Intl Conf Machine Learning, Lille, France volume 37

Evaluating a model's performance



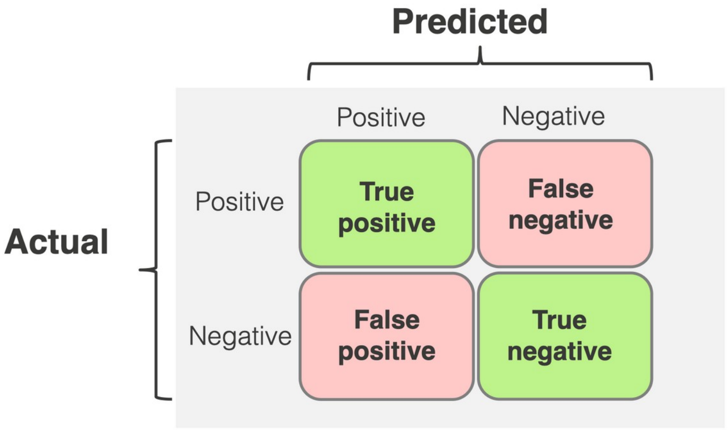
Accuracy = (TP+TN)/(TP+FN+TN+FP)

Precision = TP/(TP+FP)

Sensitivity (true positive rate) = TP/(TP+FN)

Specificity (true negative rate) = TN/(TN+FP)

Evaluating a model's performance



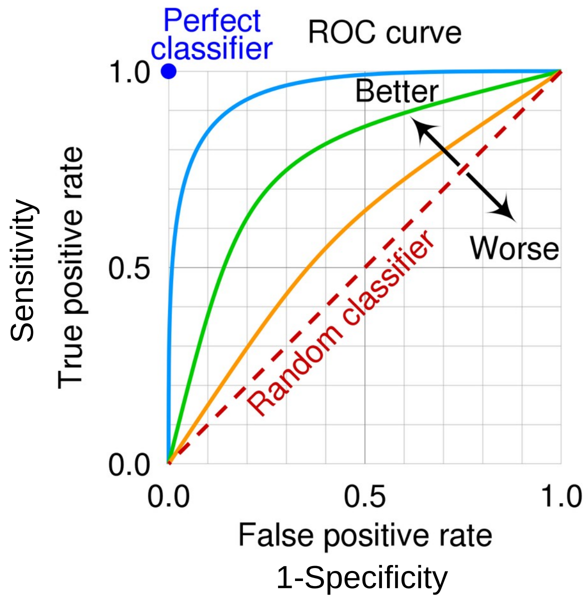
Accuracy = (TP+TN)/(TP+FN+TN+FP)

Precision = TP/(TP+FP)

Sensitivity (true positive rate) = TP/(TP+FN)

Specificity (true negative rate) = TN/(TN+FP)

Receiver operating characteristic (ROC) curve

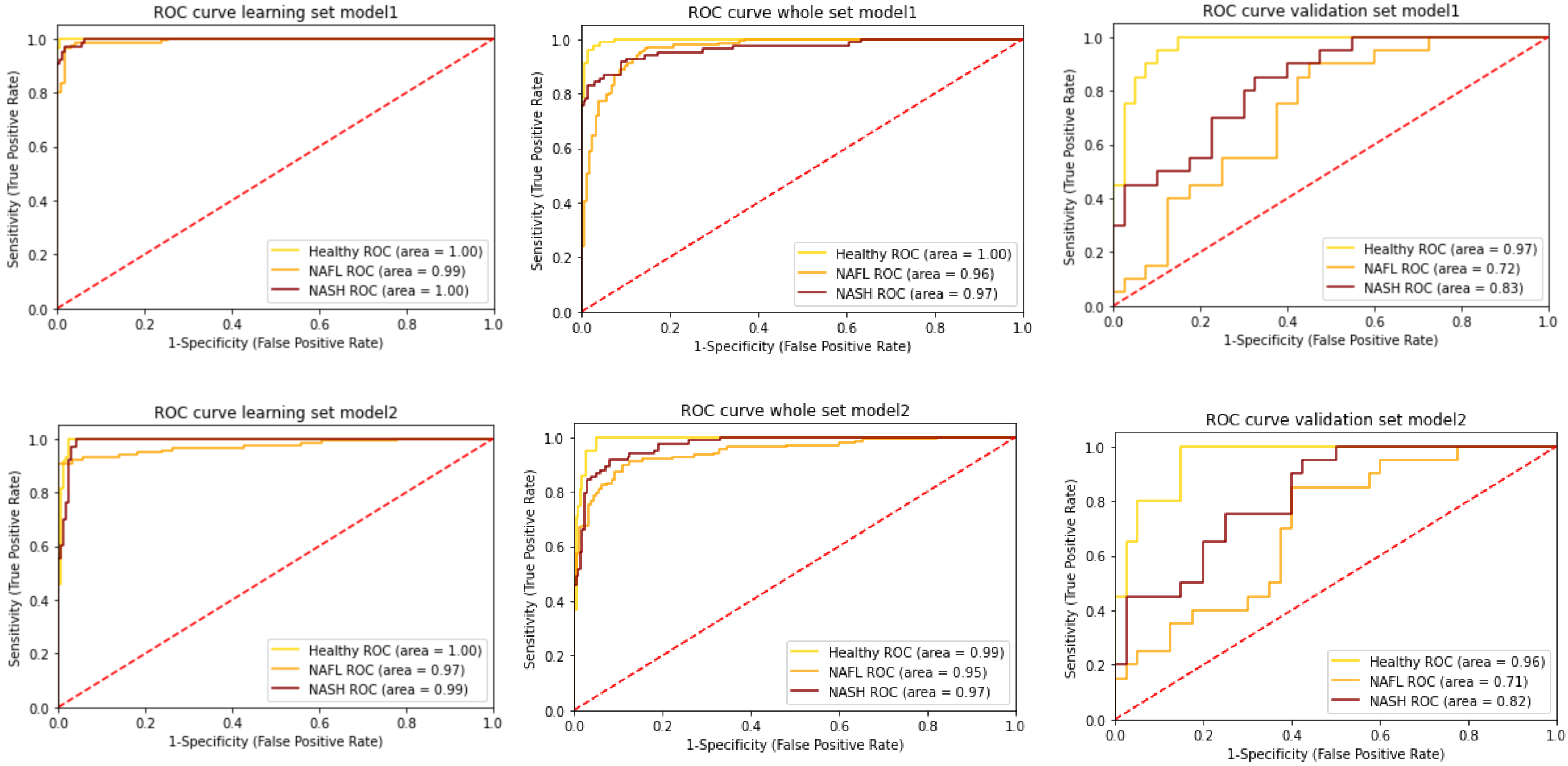


Different accuracies on different datasets

Learning set | Whole set | Validation set

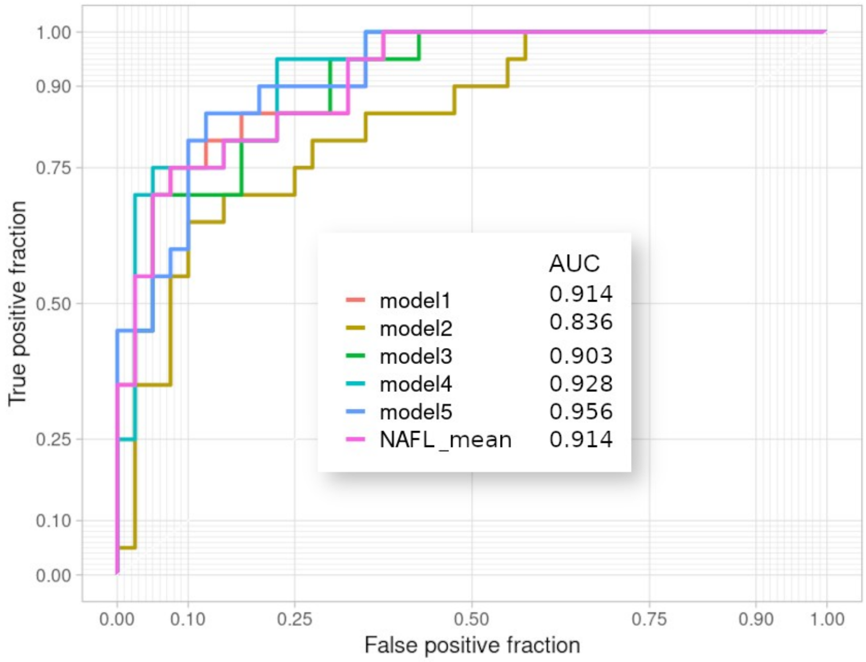
Model 1

Model 2

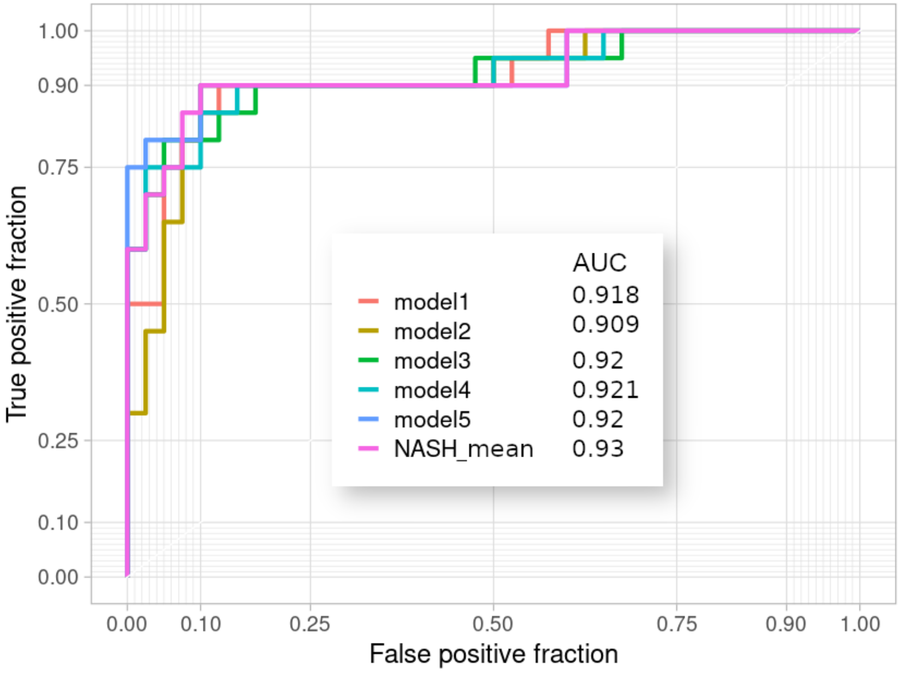


How good is the model to distinguish NAFL and NASH?

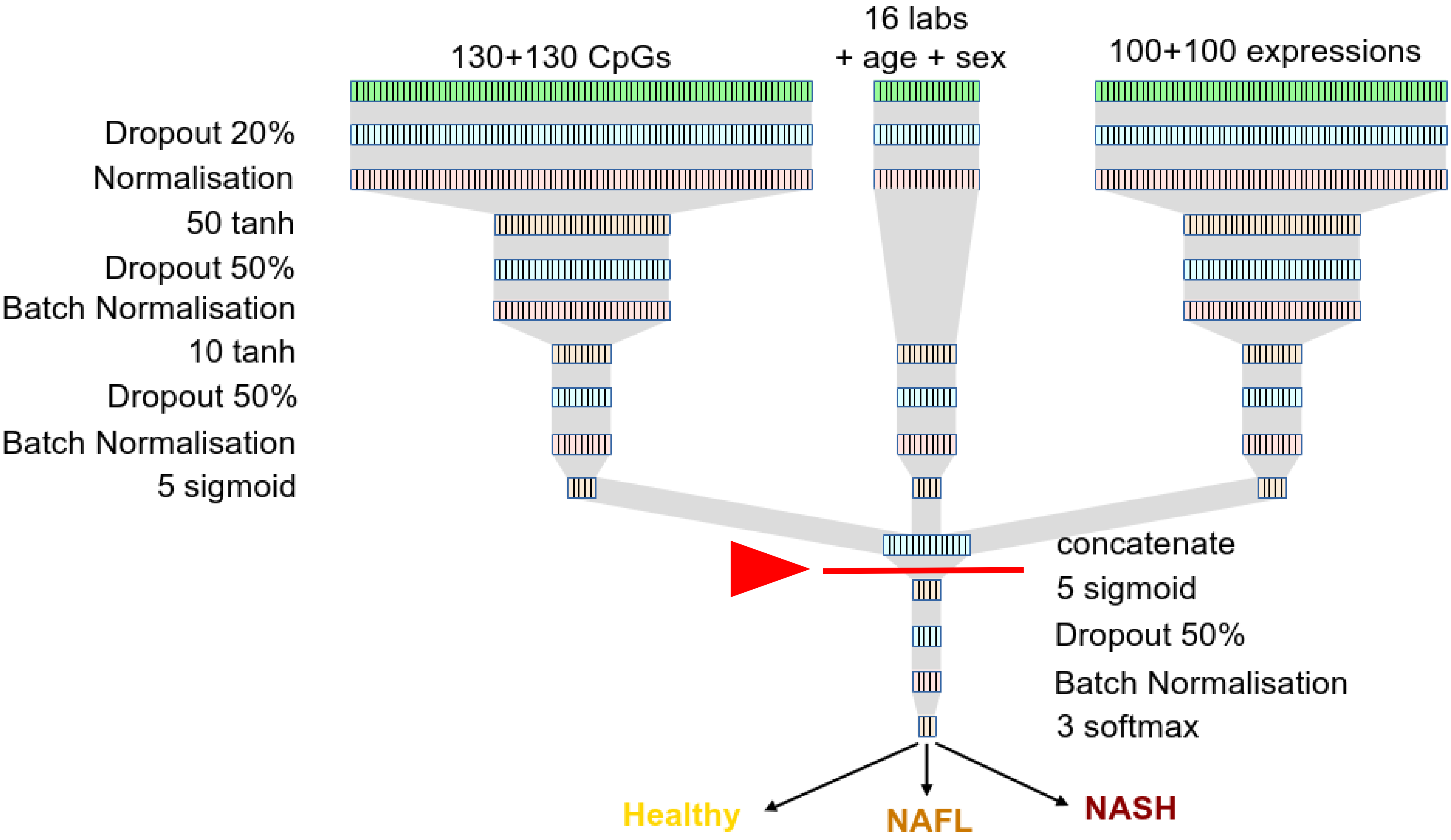
sensitivity



1 - specificity

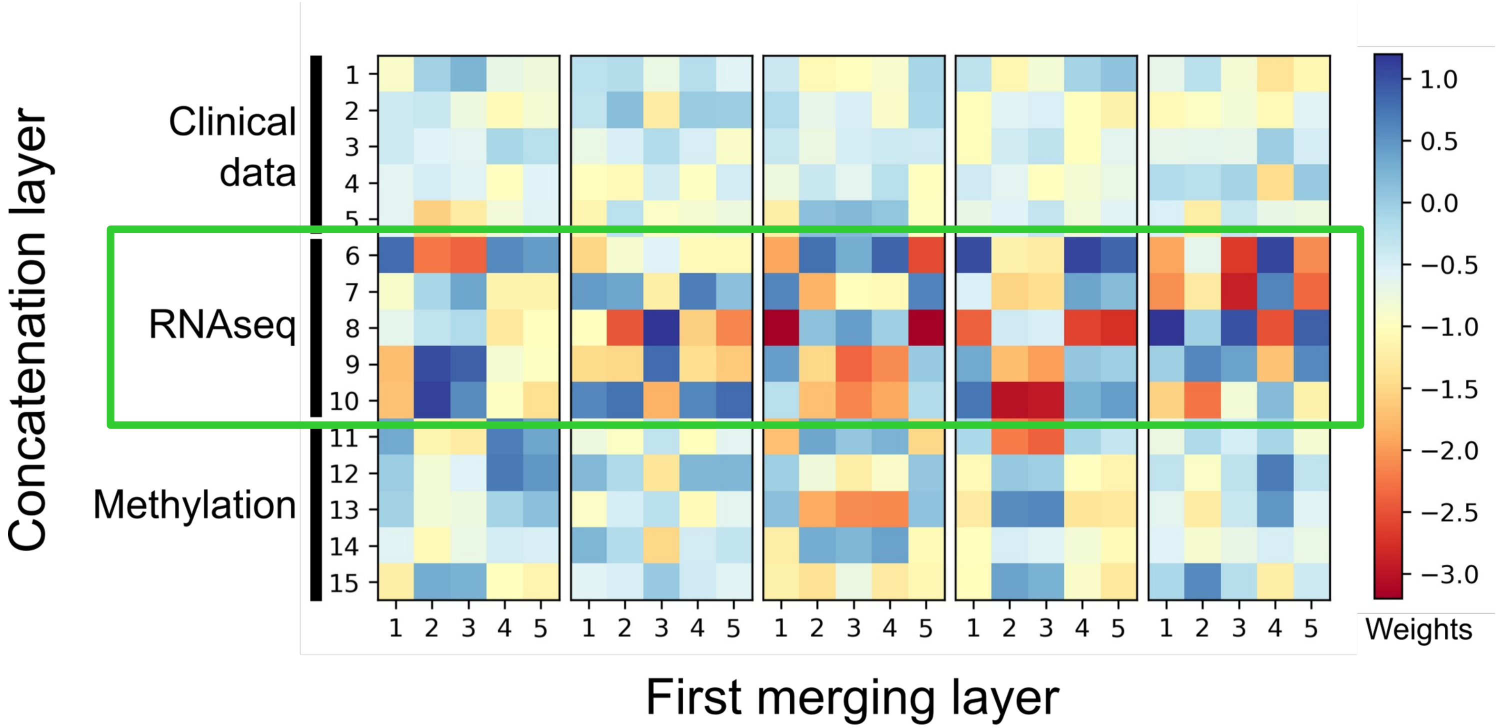


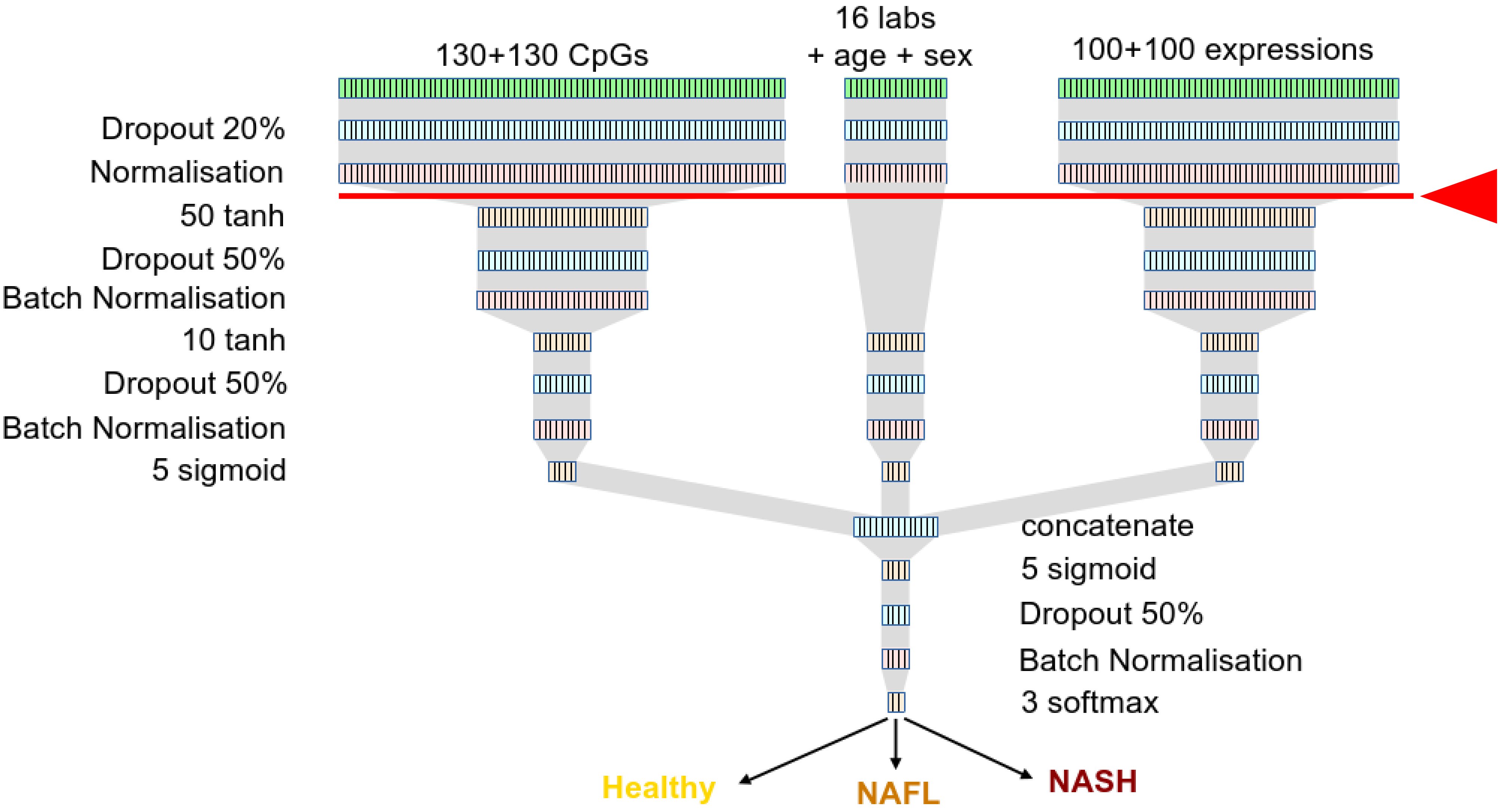
1 - specificity


130+130 CpGs
16 labs
+ age + sex
100+100 expressions
Dropout 20%
Normalisation
50 tanh
Dropout 50%
Batch Normalisation
10 tanh
Dropout 50%
Batch Normalisation
5 sigmoid
concatenate
5 sigmoid
Dropout 50%
Batch Normalisation
3 softmax
Healthy
NAFL
NASH

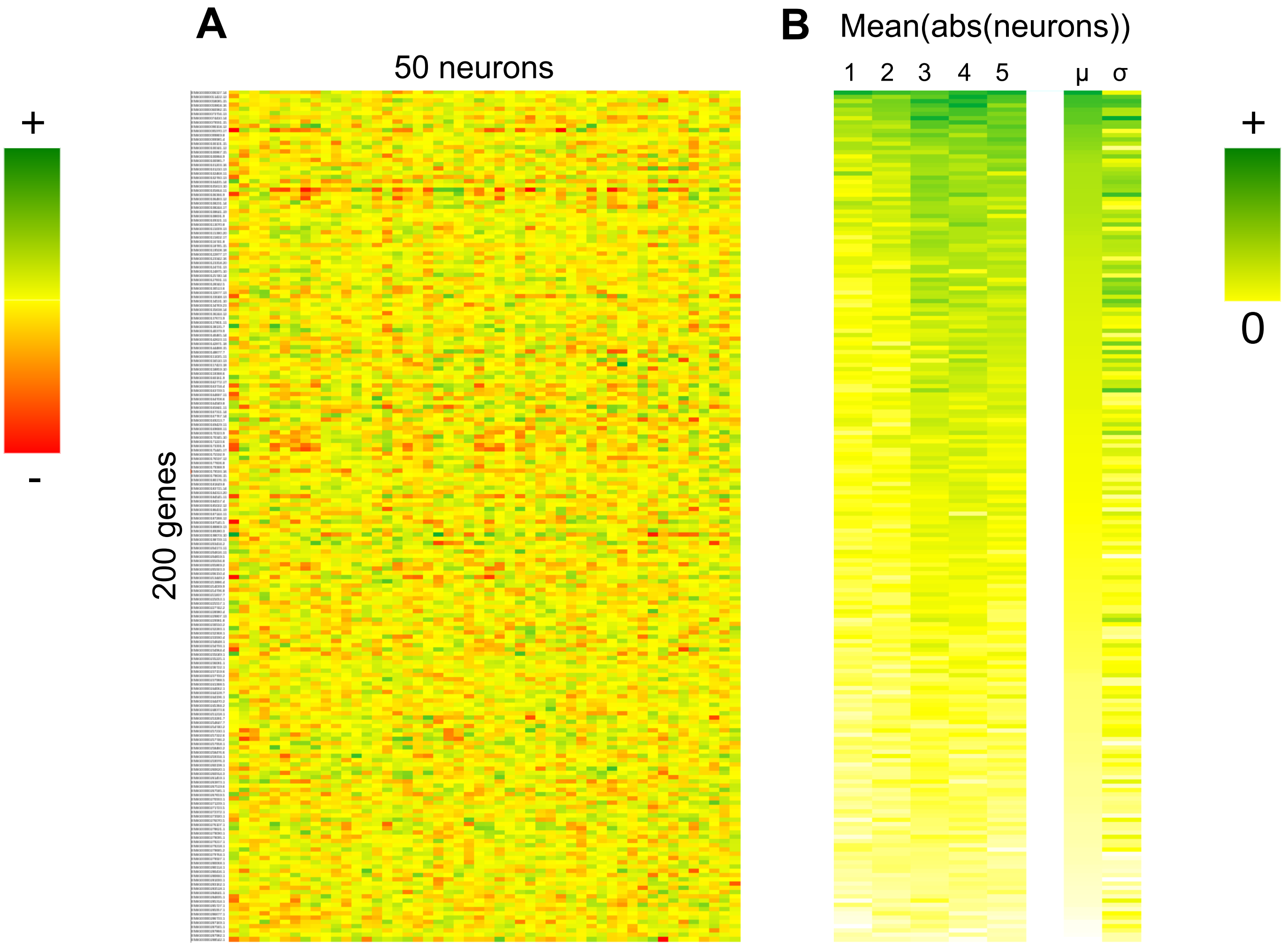
Peeping into the black box

The RNAseq module has the most impact on output




130+130 CpGs
16 labs
+ age + sex
100+100 expressions
Dropout 20%
Normalisation
50 tanh
Dropout 50%
Batch Normalisation
10 tanh
Dropout 50%
Batch Normalisation
5 sigmoid
concatenate
5 sigmoid
Dropout 50%
Batch Normalisation
3 softmax
Healthy
NAFL
NASH

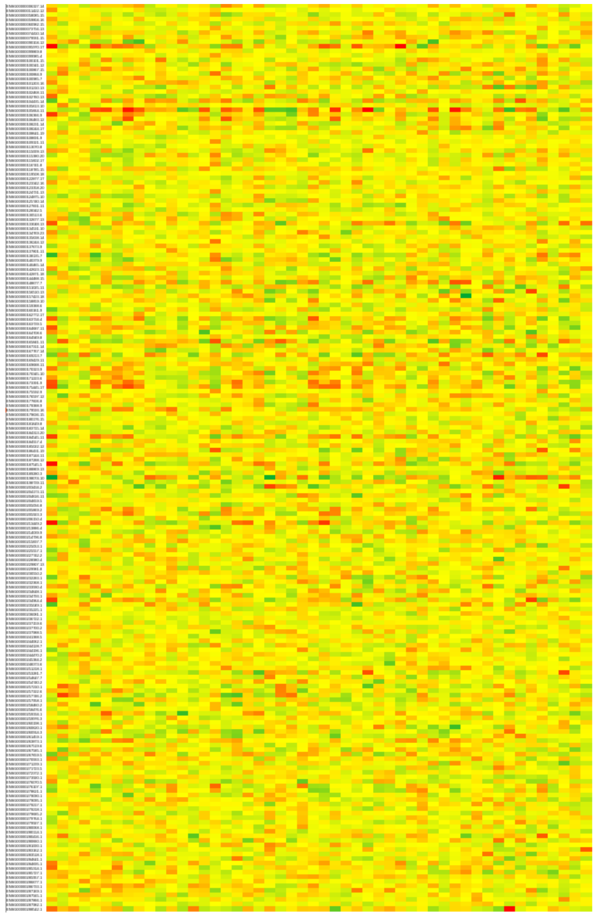
Independently trained models learn from the same genes



Known and new genes in MASLD severity

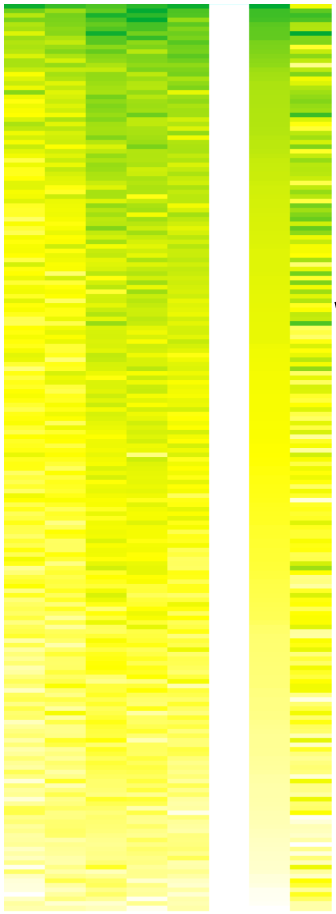
50 neurons

200 genes



Mean(abs(neurons))

1 2 3 4 5 μ σ



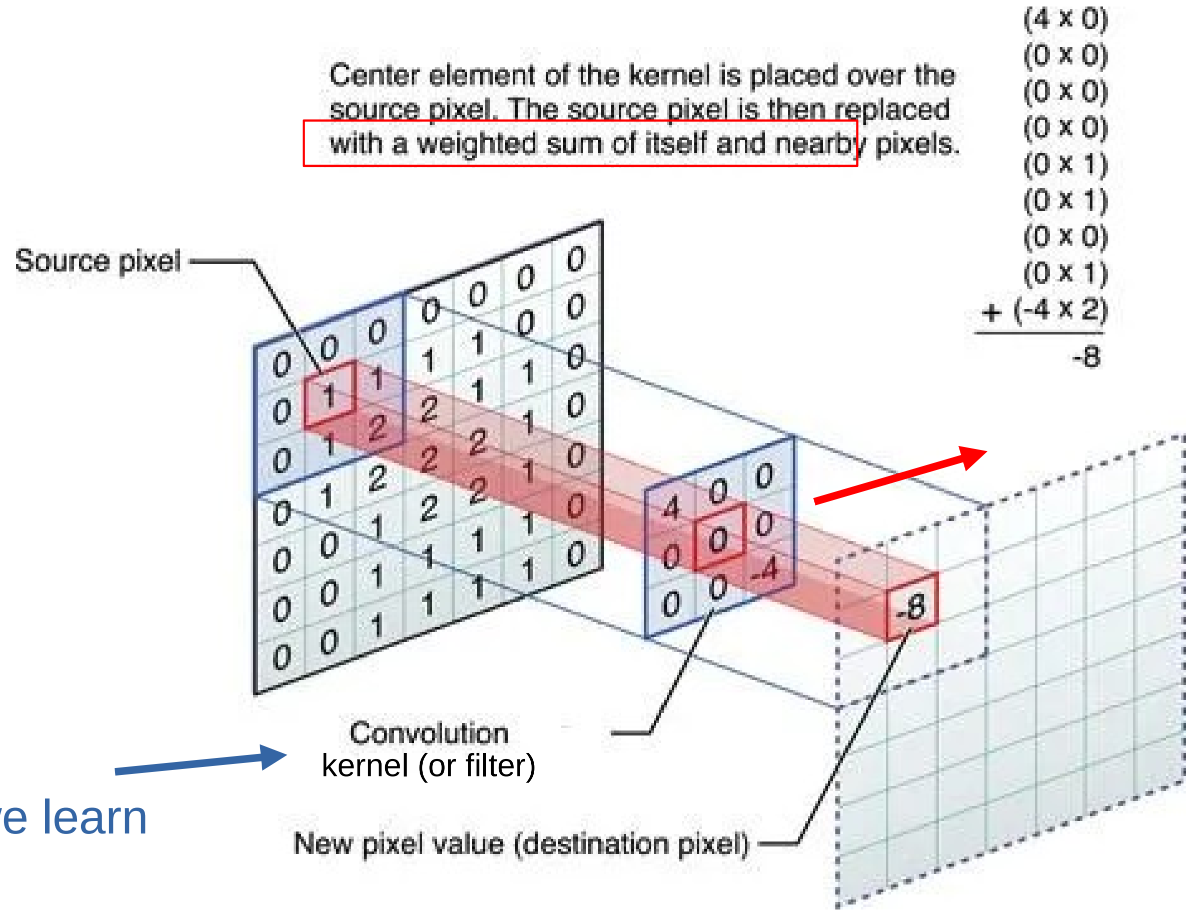
COMP
ANKRD1
PRAMEF10
SFRP4
CHI3L1
unknown
RAB3B
FABP5P7
KRTAP5-1
PGAM2
THBS1-AS1
FABP4
PADI1
CXCL3
THY1-AS1
RGS1
unknown
PRAMEF33
GDF15
PNPLA5
GPR158
FCAR
LINC02348
MMP7
ESPNL
CYP1A1
MT1B
KRTAP5-AS1
SDHAP2

TREM2
AKR1B10
LPL
STMN2
DUSP8
unknown
GAPDHP28
CA12
unknown
CYP2C19
SPP1
ART5
EEF1A2
unknown
CH25H
OLR1
THBS1-IT1
CEMP1
DHRS2
ALOX15B
CCL20
HKDC1
MMP9
unknown
KRT80
TRIM31
BCL2A1
LINC00940
FOS

4

Convolutional Neural Networks (CNN)

We can detect local features by linking neighbouring inputs



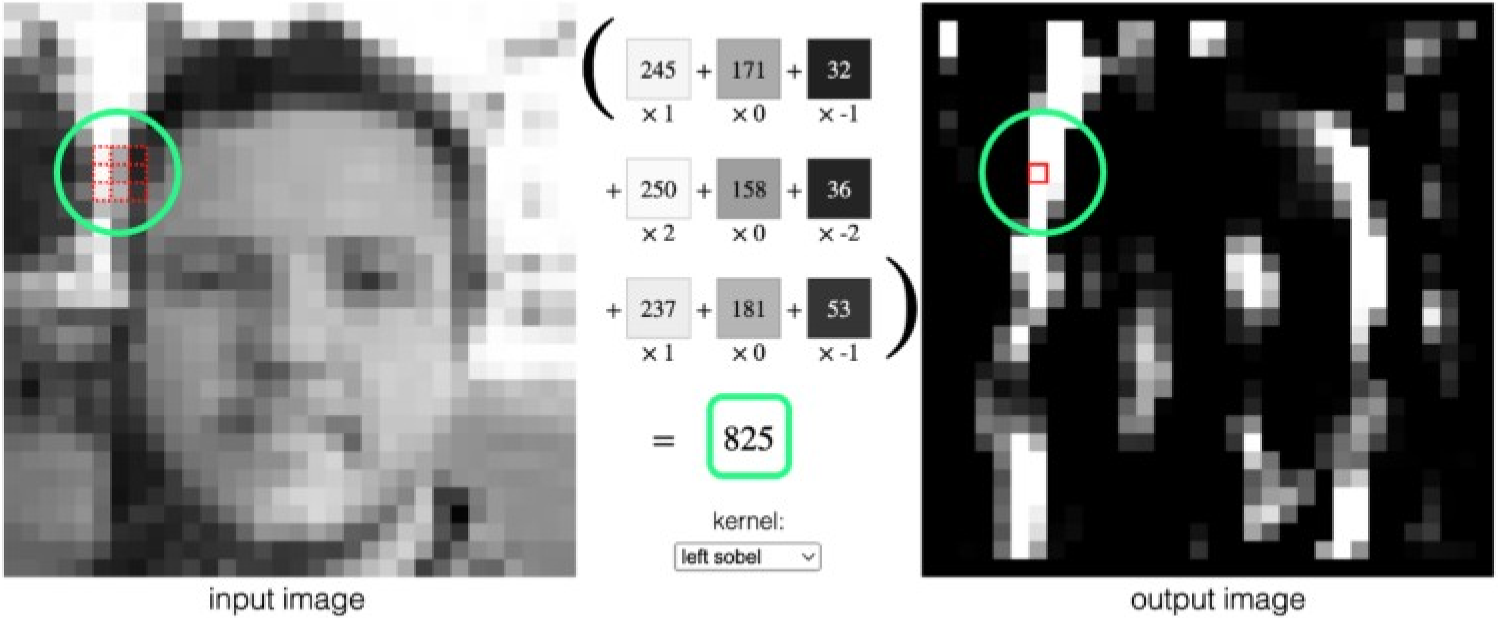
Detection of local features, such as edges, orientation, colour gradients, etc

This is what we learn

Example of feature detection: vertical edges

Sobel Kernel
(3 x 3)

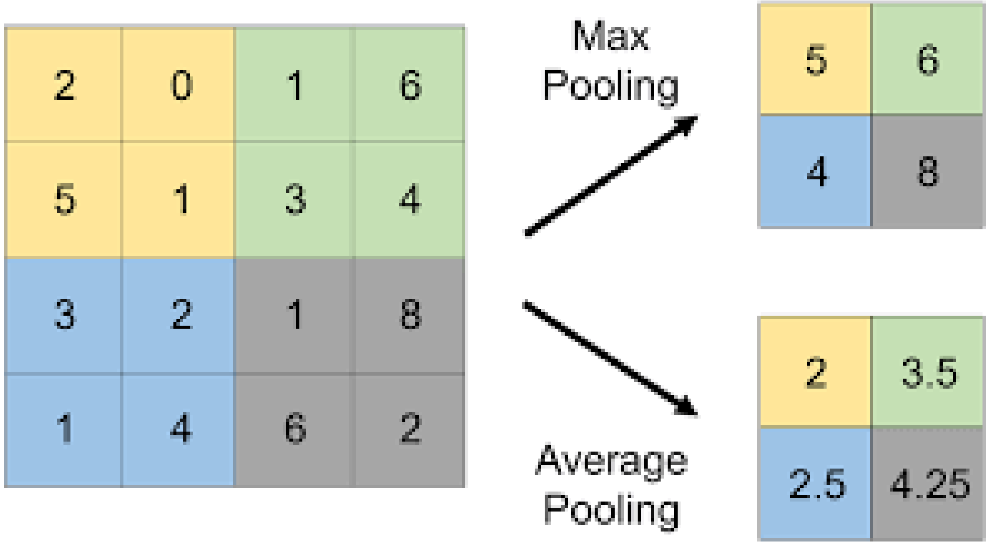
1	0	-1
2	0	-2
1	0	-1



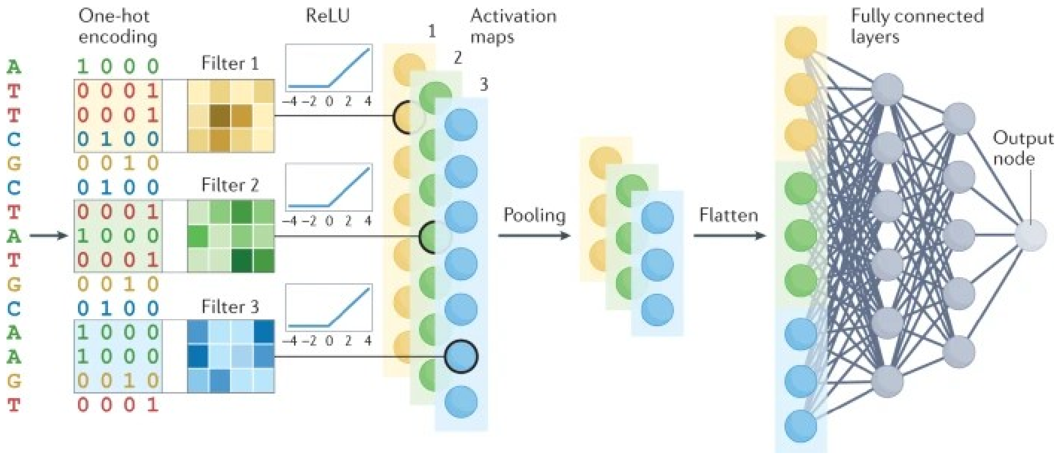
https://setosa.io/ev/image-kernels/

NB: here, we provide the kernel.
In CNNs, the kernel is learned

Downsampling: Max/average pooling



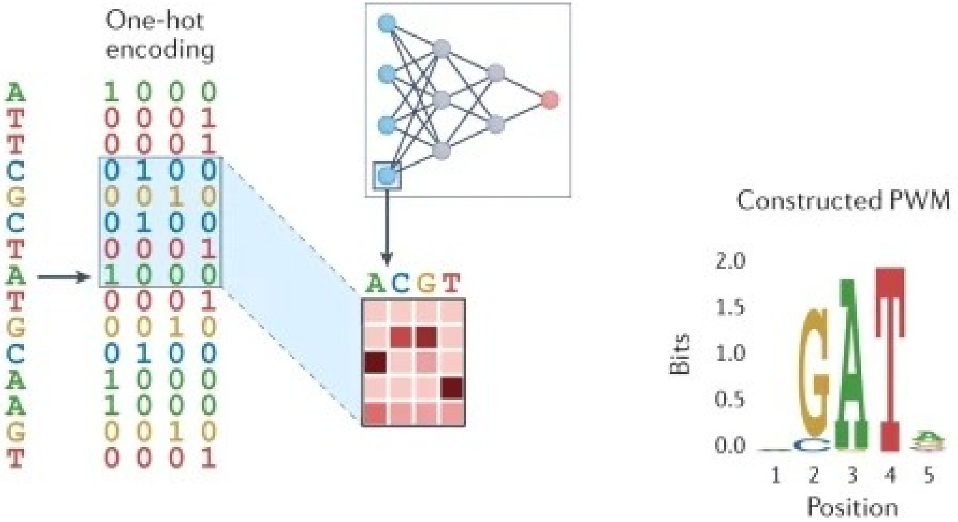
Everything is an image: DNA sequences to images



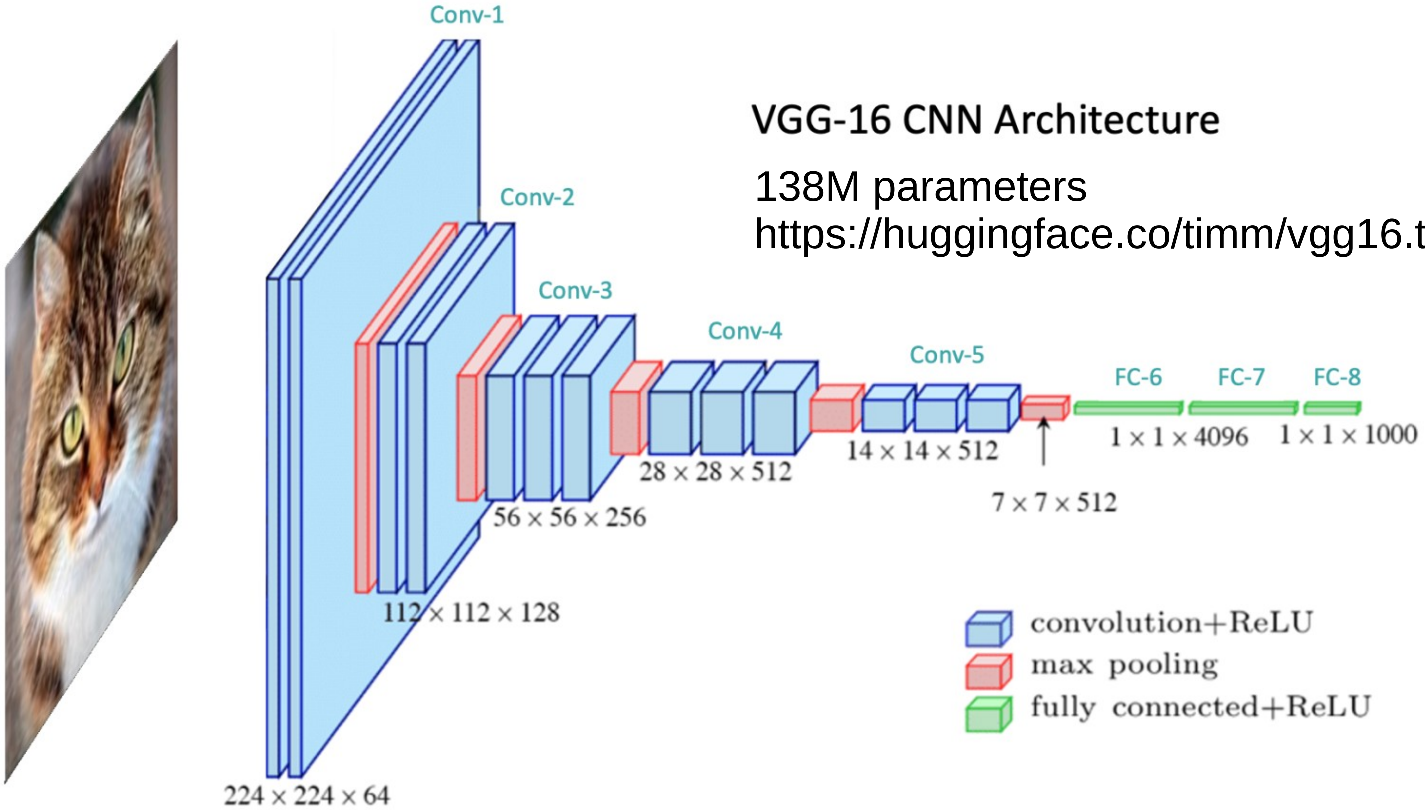
Obtaining genetics insights from deep learning via explainable artificial intelligence

Gherman Novakovsky, Nick Dexter, Maxwell W. Libbrecht, Wyeth W. Wasserman & Sara Mostafavi

Nature Reviews Genetics **24**, 125–137 (2023) | Cite this article



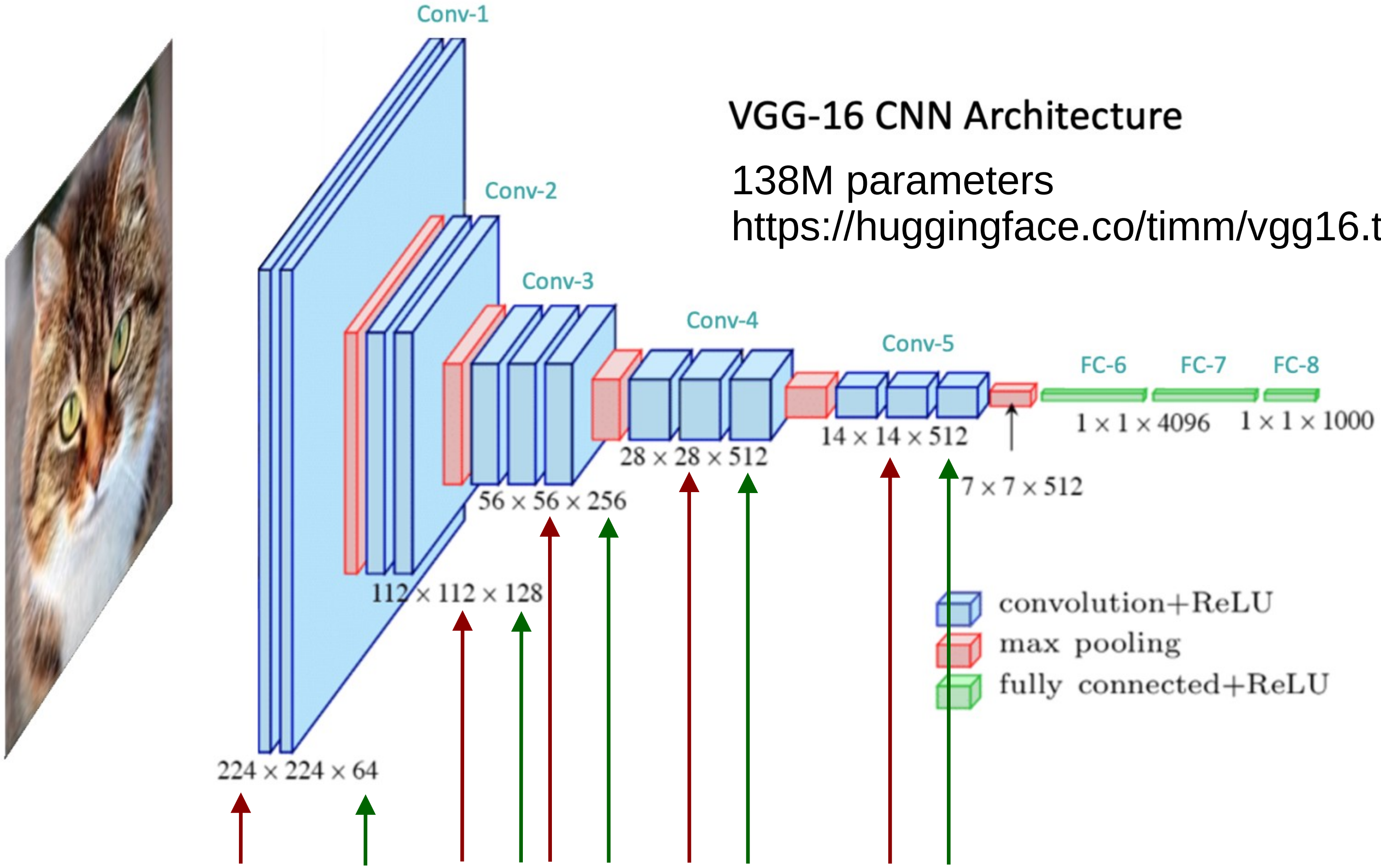
Real example: VGG



138M parameters
https://huggingface.co/timm/vgg16.tv_in1k

Simonyan K, Zisserman A (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015
https://arxiv.org/pdf/1409.1556

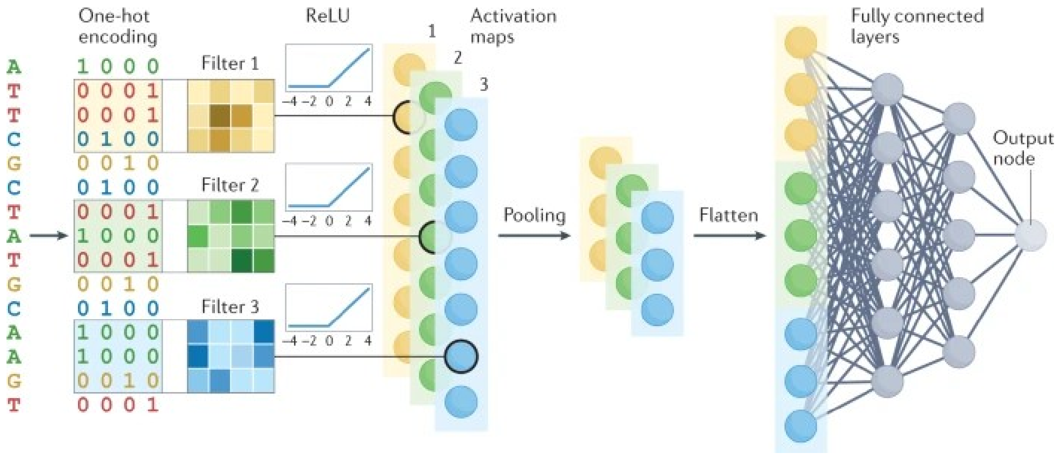
Decreasing size, increasing feature number



138M parameters
https://huggingface.co/timm/vgg16.tv_in1k

Simonyan K, Zisserman A (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015
https://arxiv.org/pdf/1409.1556

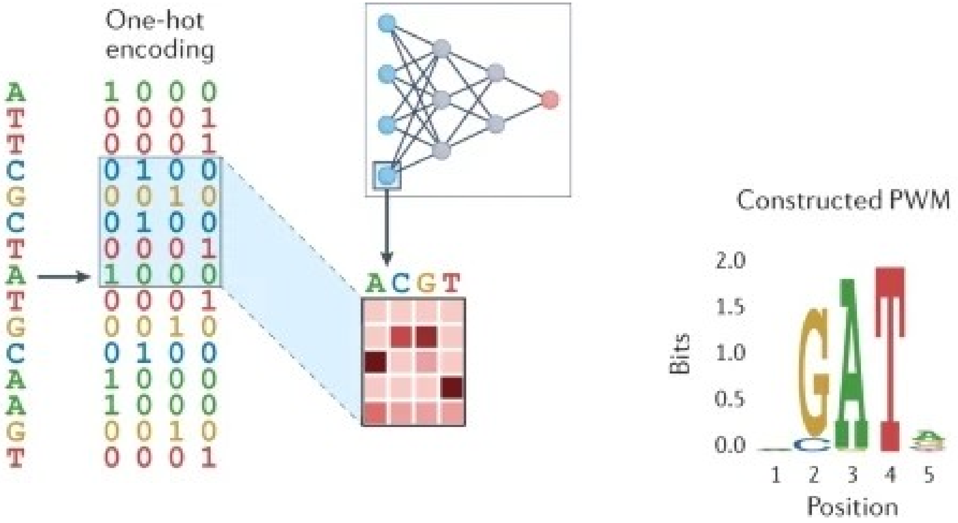
Everything is an image: DNA sequences to images



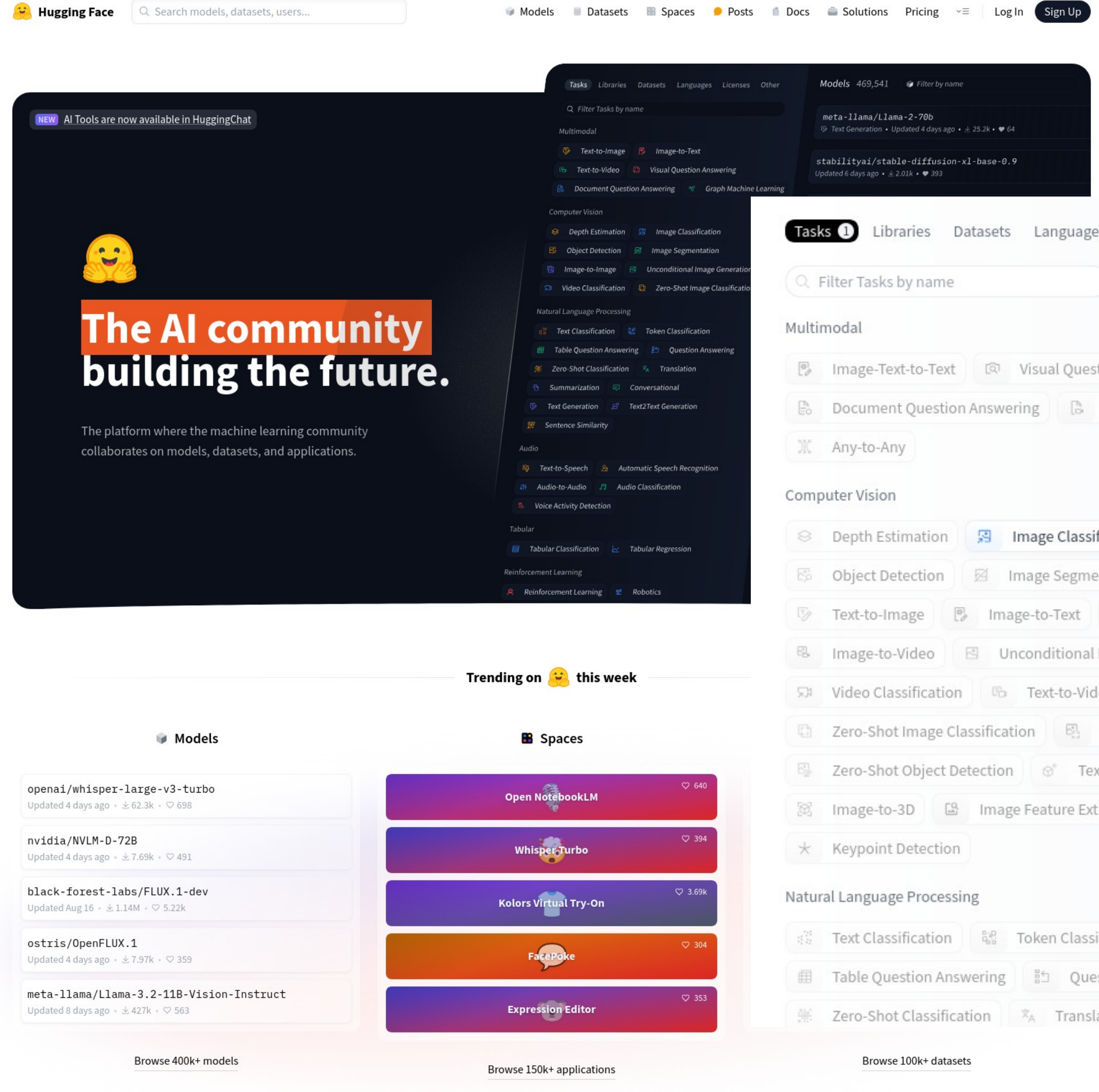
Obtaining genetics insights from deep learning via explainable artificial intelligence

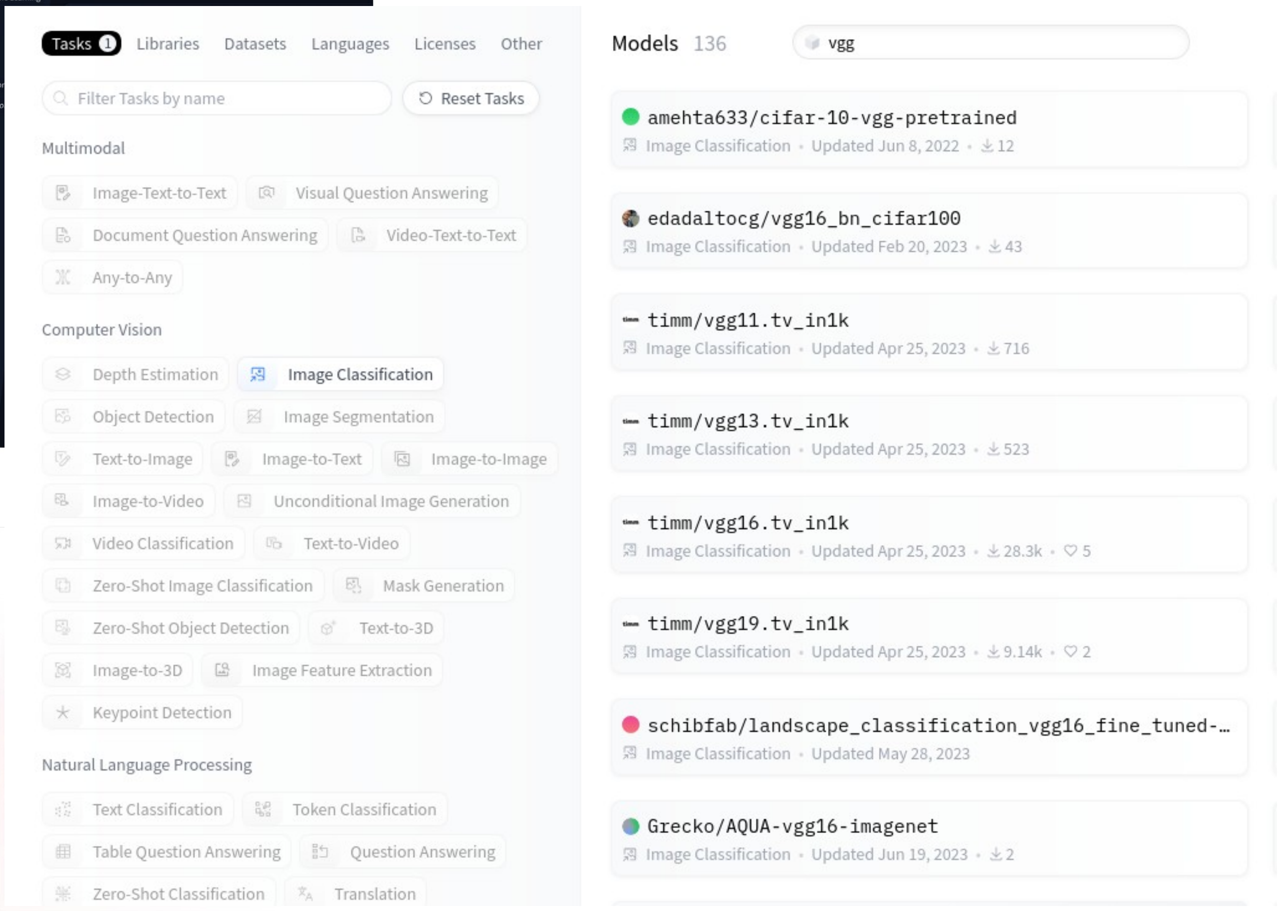
Gherman Novakovsky, Nick Dexter, Maxwell W. Libbrecht, Wyeth W. Wasserman & Sara Mostafavi

Nature Reviews Genetics **24**, 125–137 (2023) | Cite this article



Where to find models: Hugging face





Reusing models without full retraining

Skin Cancer Classification using VGG-16 and Googlenet CNN Models

January 2023 · International Journal of Computer Applications 184(42):5-9

Anju, T.E., Vimala, S. (2023). Finetuned-VGG16 CNN Model for Tissue Classification of Colorectal Cancer. In: Raj, J.S., Perikos, I., Balas, V.E. (eds) Intelligent Sustainable Systems. ICoISS 2023. Lecture Notes in Networks and Systems, vol 665. Springer, Singapore.

VGG 16 Pre-Trained Model for Early Detection of Retinal Diseases

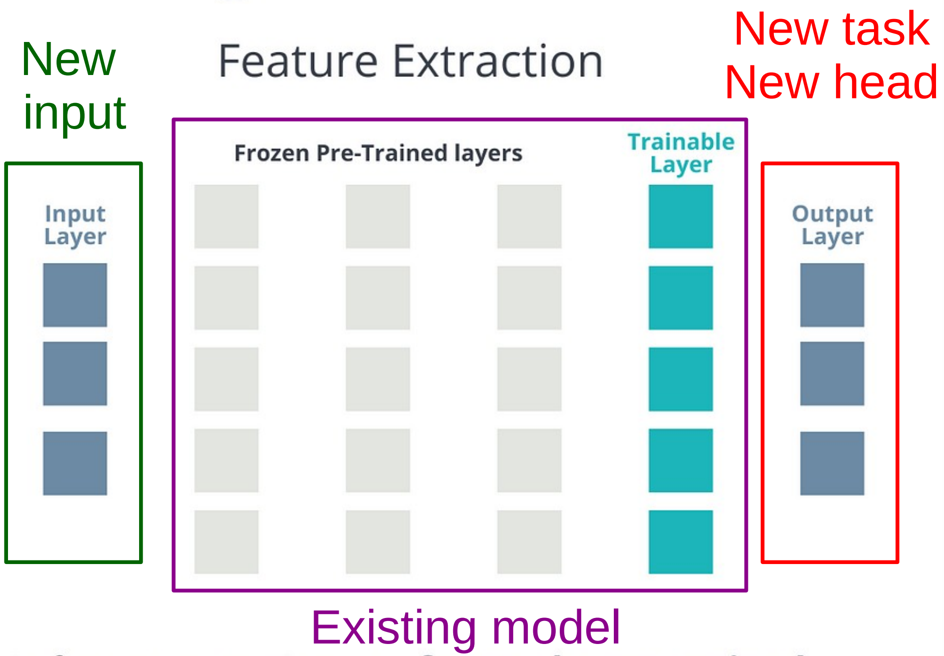
December 2023

DOI:10.1109/SMARTGENCON60755.2023.10442010

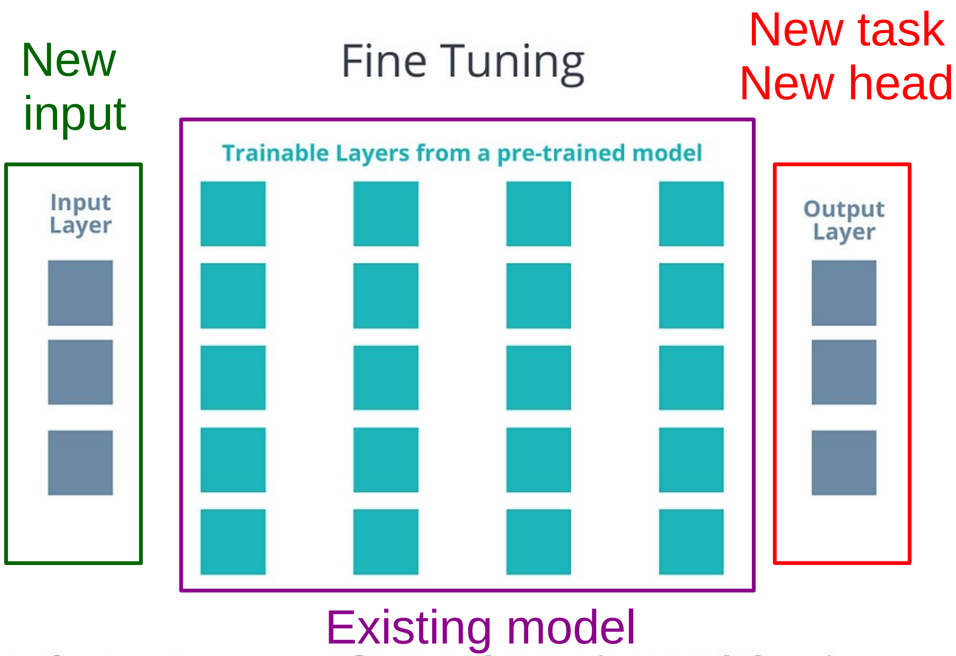
Conference: 2023 3rd International Conference on Smart Generation Computing,

Raghuvanshi S, Dhariwal S. The VGG16 Method Is a Powerful Tool for Detecting Brain Tumors Using Deep Learning Techniques. *Engineering Proceedings*. 2023; 59(1):46. https://doi.org/10.3390/engproc2023059046

Transfer Learning: How Feature Extraction & Fine-Tuning work?



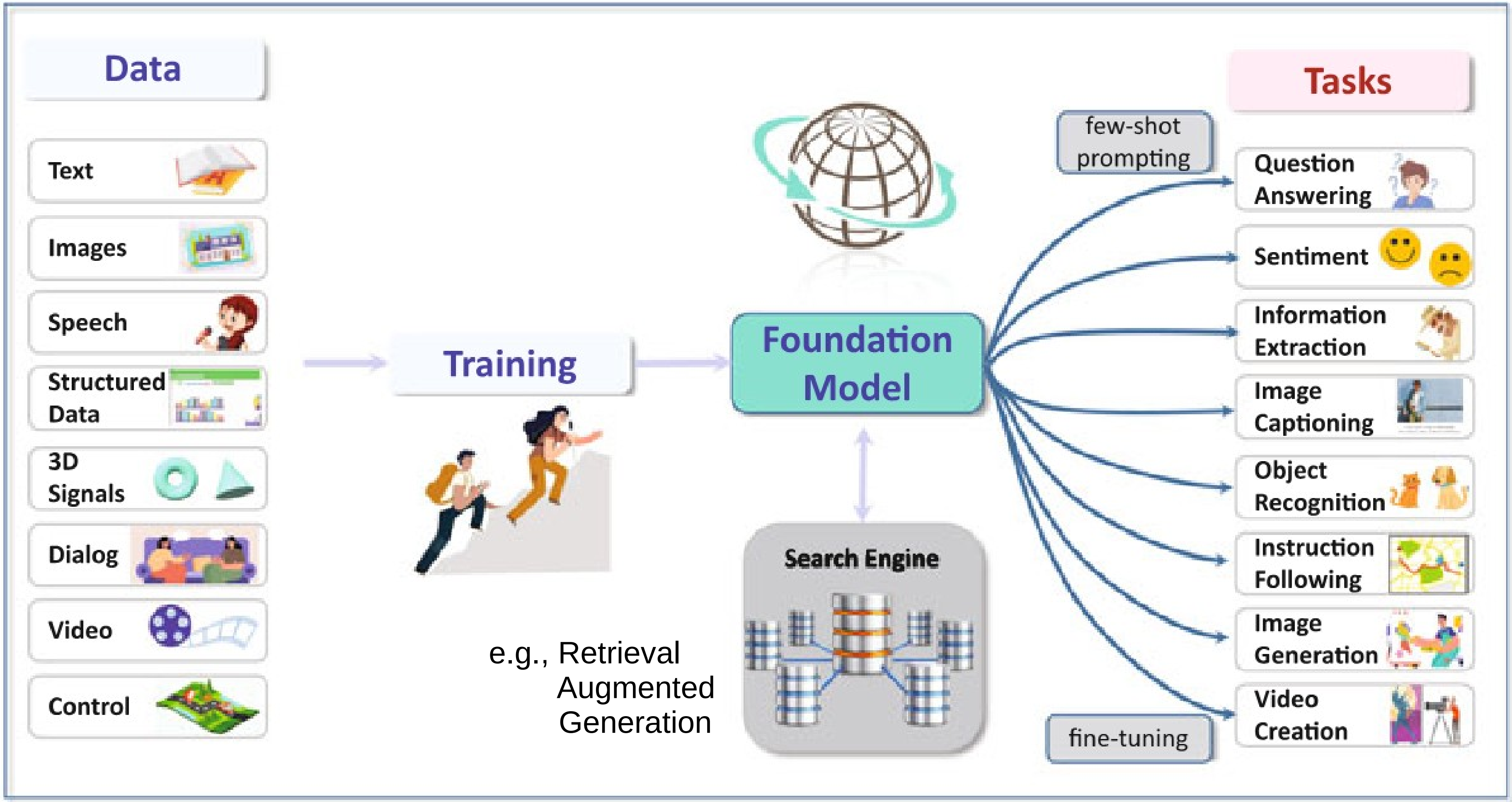
In feature extraction, you **freeze the pre-trained model** layers to preserve existing learning and **add new layers** to learn additional information.



In fine-tuning, you **unfreeze the entire model** and **train it with a lower learning rate** to adapt to new challenges.

Adapted from thuwarakesh.medium.com

Extreme transfer learning: Foundation models



5

Embeddings and latent spaces

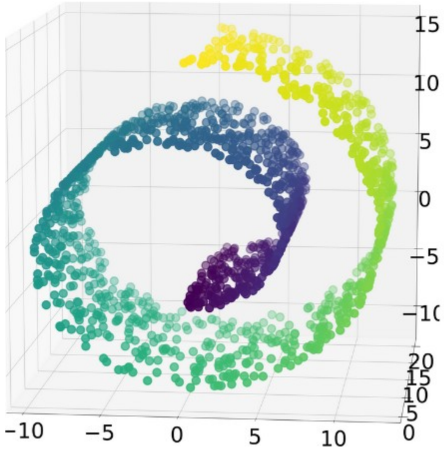
Embeddings (“plongement”)

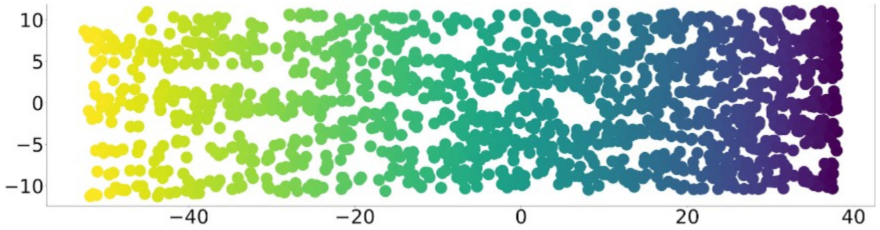
Values in reference frame A
m vectors of size **n**
from a dictionary of **o**
(i.e. **o** coordinates)

→

Values in reference frame B
m vectors of size **n’**
from a dictionary of **o’**
(i.e. **o’** coordinates)

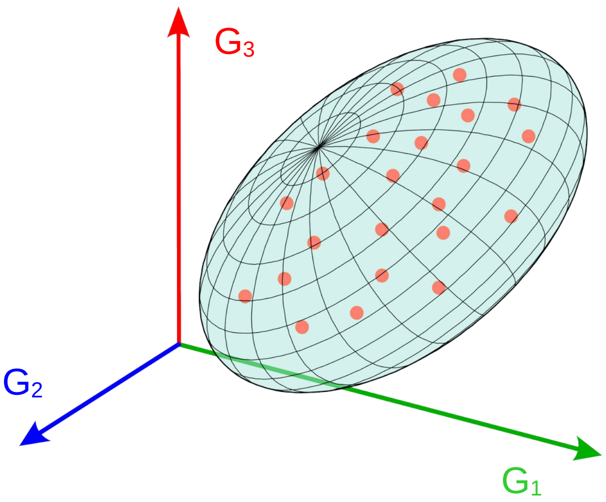
Embedding from a space with **n** dimensions
into a space of **o** dimensions





A PCA is an embedding

Axes = N genes
Coordinate = expressions

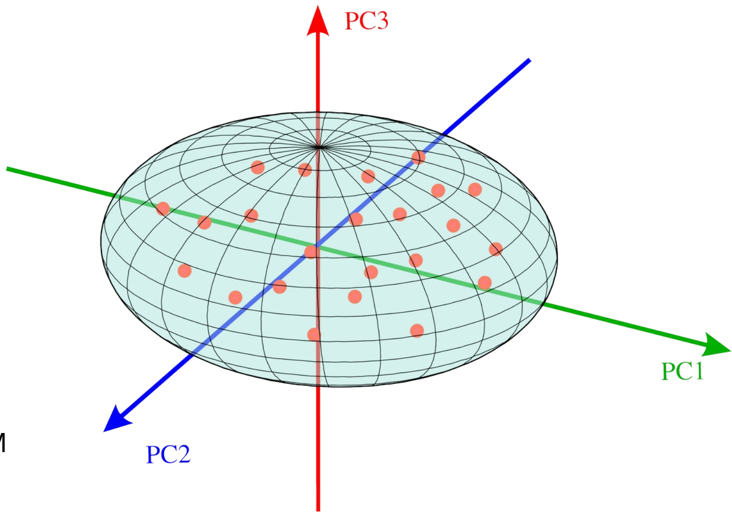


G_1
G_2
G_3
G_4
…
G_i
…
G_N

→

Axes = M principal components
Coordinates = Rotations x expressions

PC_1
PC_2
PC_3
PC_4
…
PC_j
…
PC_M



“Similar” objects are neighbours in the latent space

Dictionary (initial space)

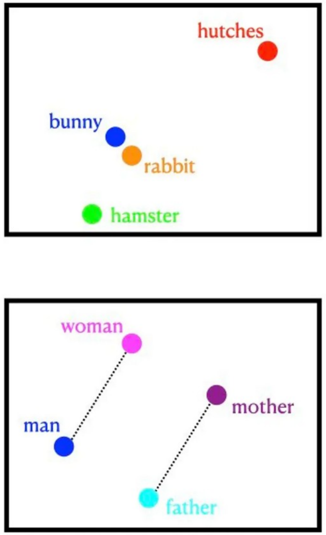
bunny	father	hamster	hutches	man	mother	rabbit	tractor	woman
1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0

bunny	father	hamster	hutches	man	mother	rabbit	tractor	woman
0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0	0

Semantics (Embedding space)

Word	alive	mammal	rodent	bipedel	gender	plural
bunny	0.9	0.7	-0.1	-0.1	0.2	-0.3
rabbit	0.9	0.8	-0.1	-0.1	0.4	-0.1
hamster	0.9	0.6	0.7	-0.3	-0.4	-0.4
hutches	-0.9	-0.3	-0.1	-0.9	0.0	0.9
mother	0.9	0.1	0.1	0.2	0.8	-0.7
father	0.9	0.2	0.1	0.2	-0.8	-0.7
woman	0.9	0.8	-0.9	0.9	0.8	-0.8
man	0.9	0.8	-0.7	0.9	-0.8	-0.8

Dimensionality Reduction



Word

Vector Embedding

Dimensionality

2D Visualization

Arithmetic operation in latent space = semantic statement

Dictionary (initial space)

bunny	father	hamster	hutches	man	mother	rabbit	tractor	woman
1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0

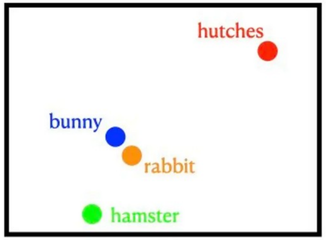
bunny	father	hamster	hutches	man	mother	rabbit	tractor	woman
0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0	0

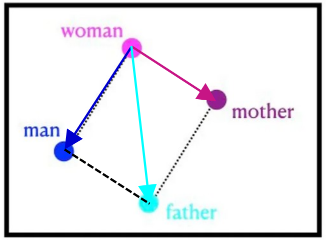
Semantics (Embedding space)

Word	alive	mammal	rodent	bipedel	gender	plural
bunny	0.9	0.7	-0.1	-0.1	0.2	-0.3
rabbit	0.9	0.8	-0.1	-0.1	0.4	-0.1
hamster	0.9	0.6	0.7	-0.3	-0.4	-0.4
hutches	-0.9	-0.3	-0.1	-0.9	0.0	0.9

Word	alive	mammal	rodent	bipedel	gender	plural
mother	0.9	0.1	0.1	0.2	0.8	-0.7
father	0.9	0.2	0.1	0.2	-0.8	-0.7
woman	0.9	0.8	-0.9	0.9	0.8	-0.8
man	0.9	0.8	-0.7	0.9	-0.8	-0.8

Dimensionality Reduction





Word

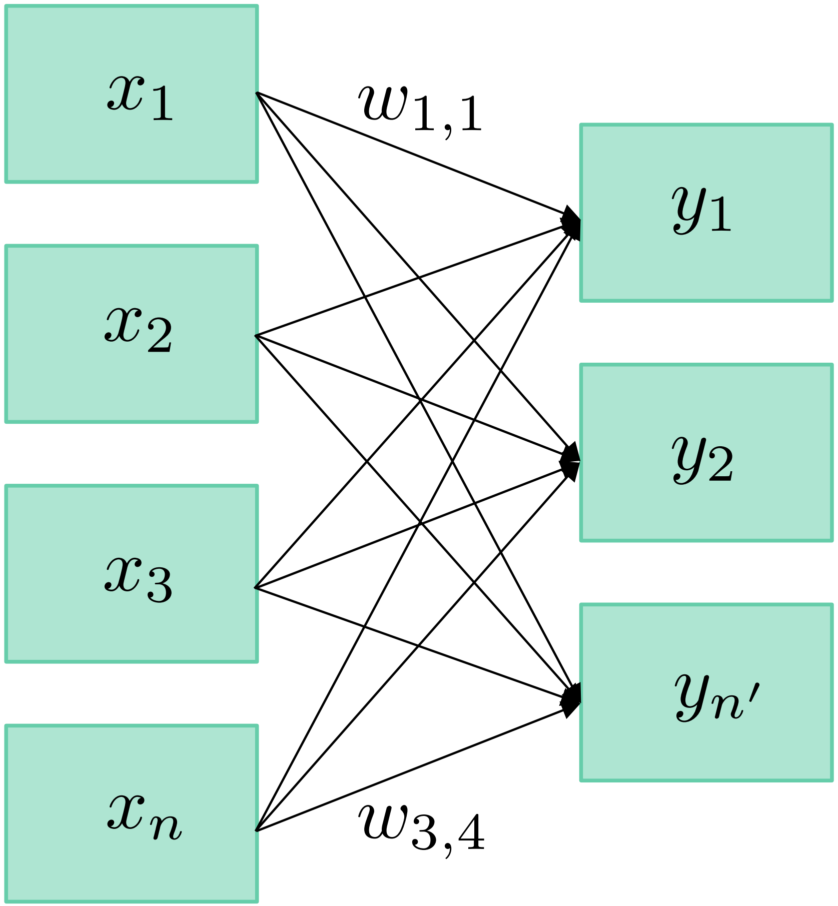
Vector Embedding

Dimensionality

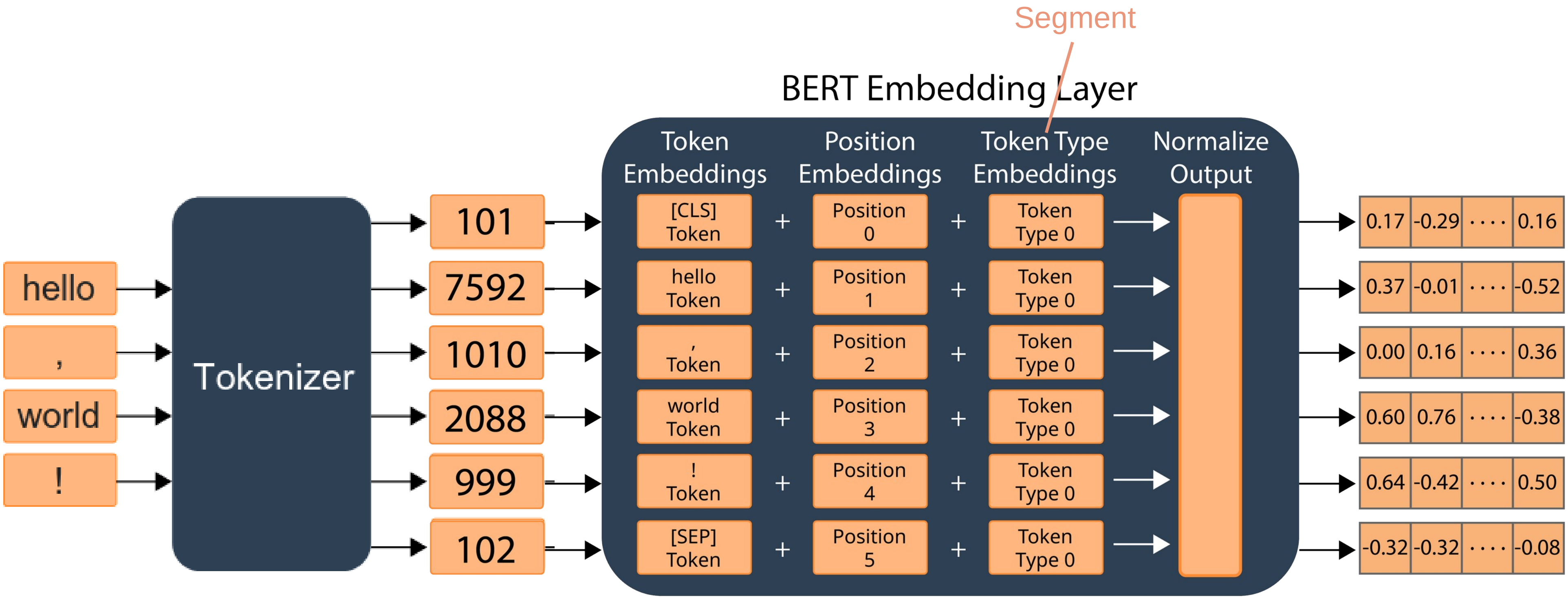
2D Visualization

In the latent space, the vector going from woman to father is equal
to the vector going from woman to man <u>plus</u> the vector going from woman to mother

DL: Fully connected layers to learn the embedding

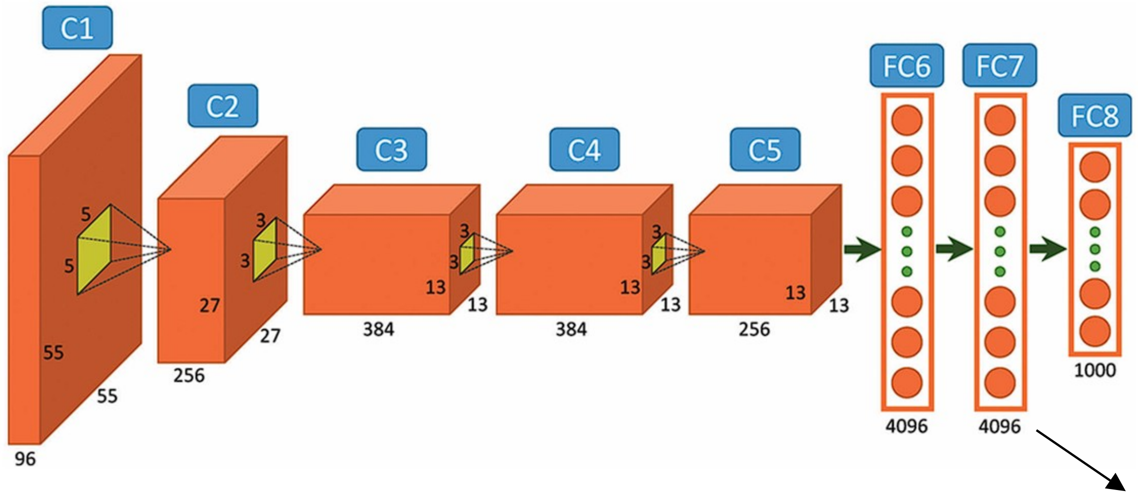


There can be several embeddings



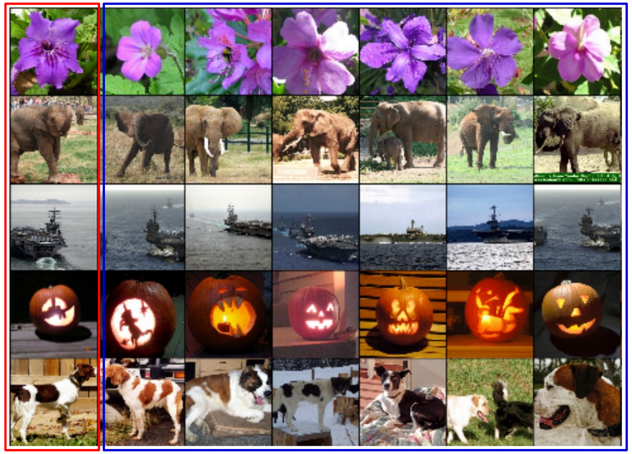
Source: https://tinkerd.net/blog/machine-learning/bert-embeddings/

Deep CNNs are “embedding” the images in a “latent space”



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)

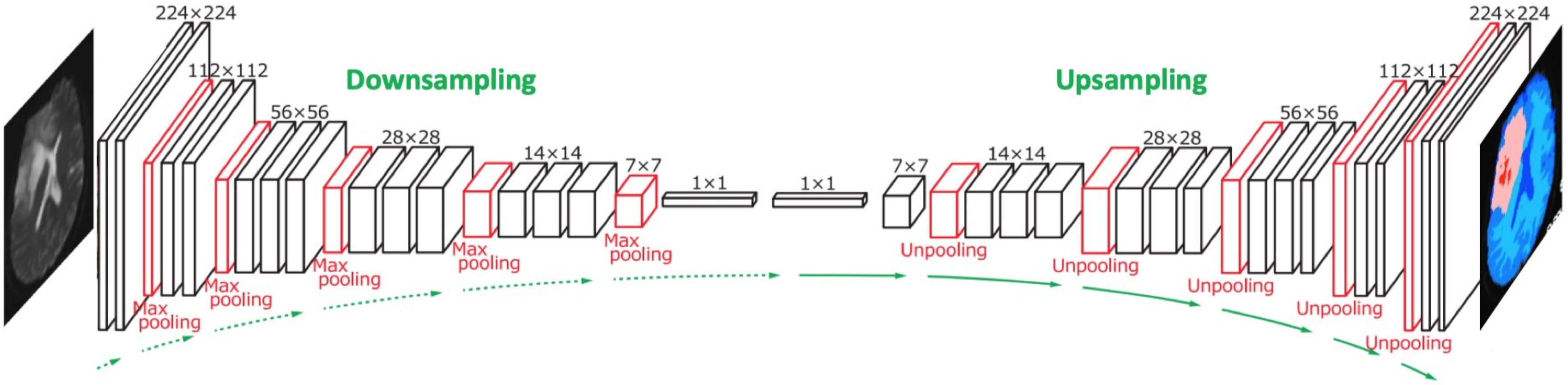
https://huggingface.co/debashd/AlexNet

6

Encoder-Decoders

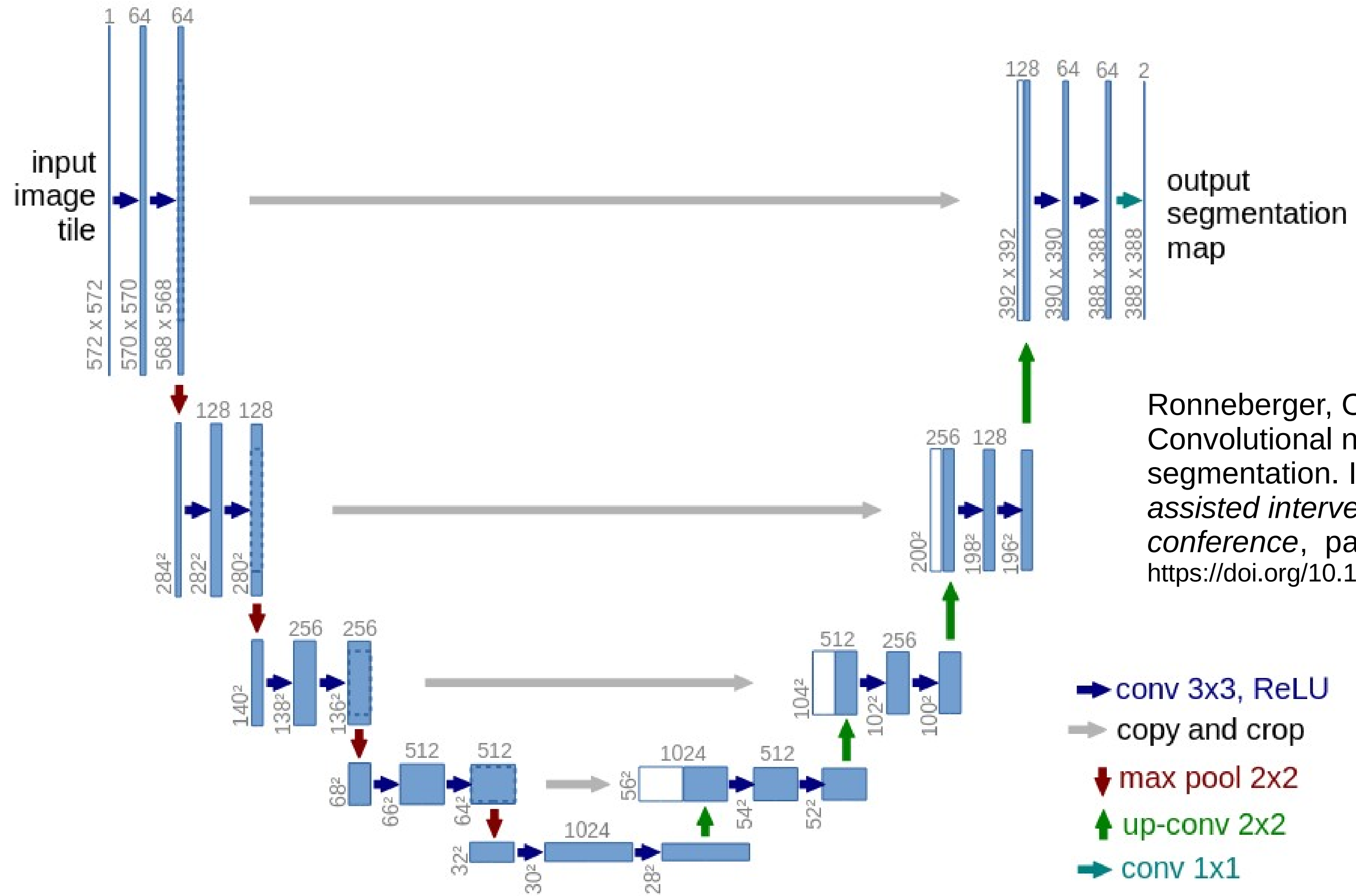
(variational) AutoEncoders (VAEs)

Encoder-decoder



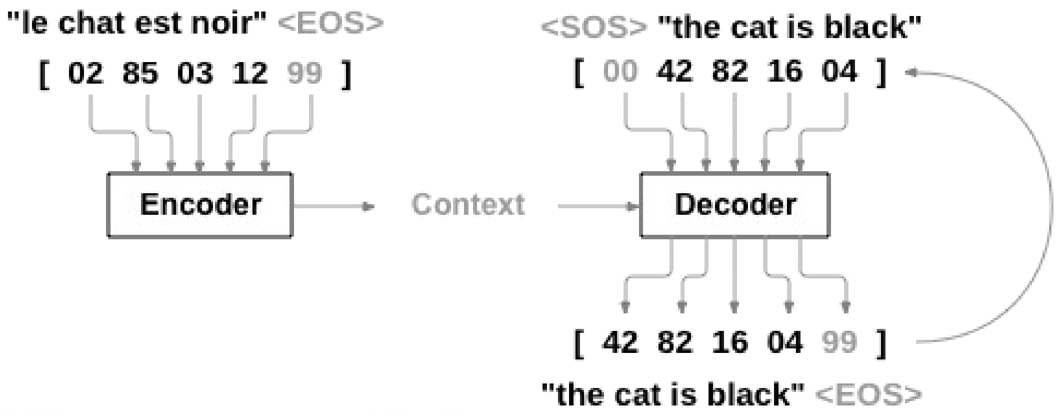
Example of segmentation to identify brain tumours

Disclaimer: it is more complicated

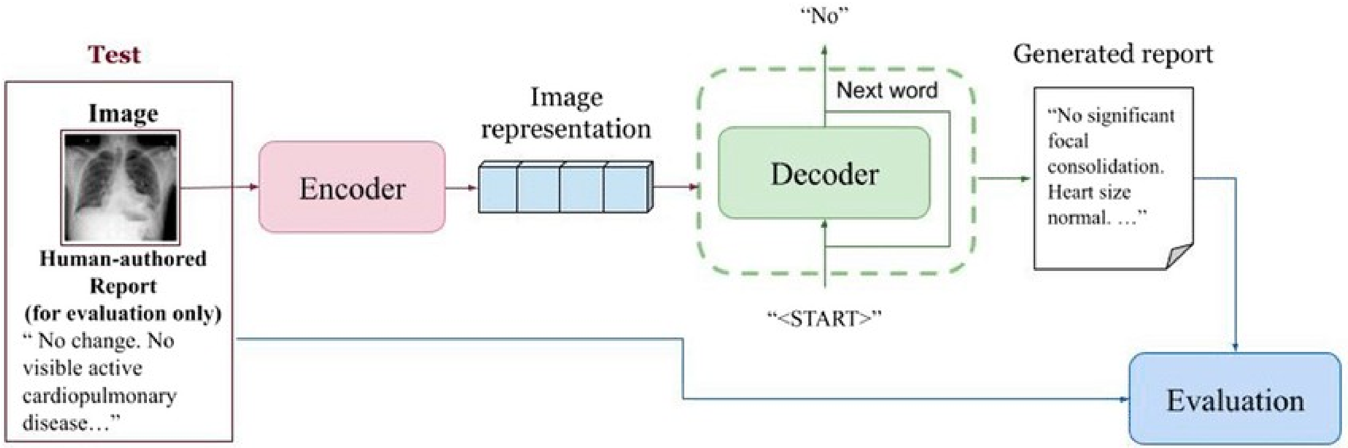


Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference*, part III 18 (pp. 234-241)
https://doi.org/10.1007/978-3-319-24574-4_28

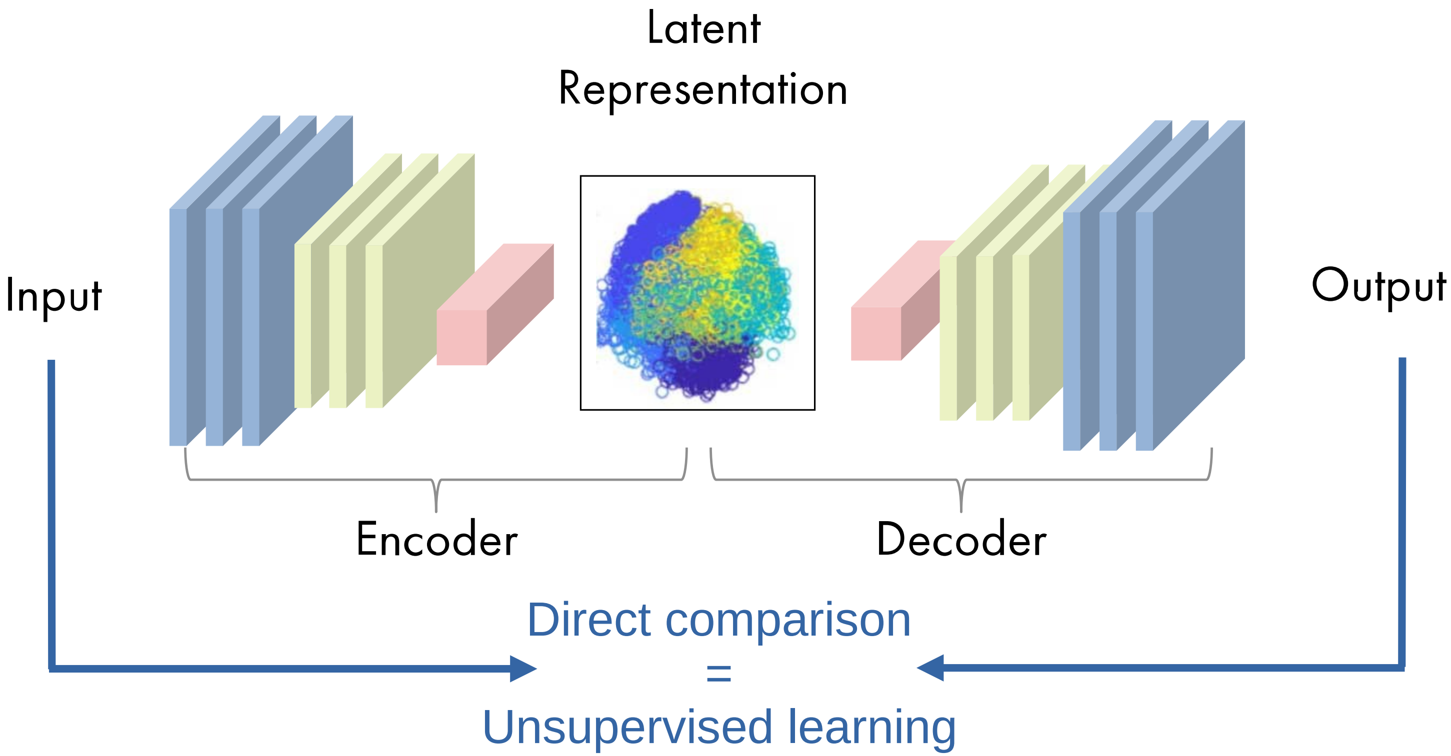
Encoders and decoders can be anything



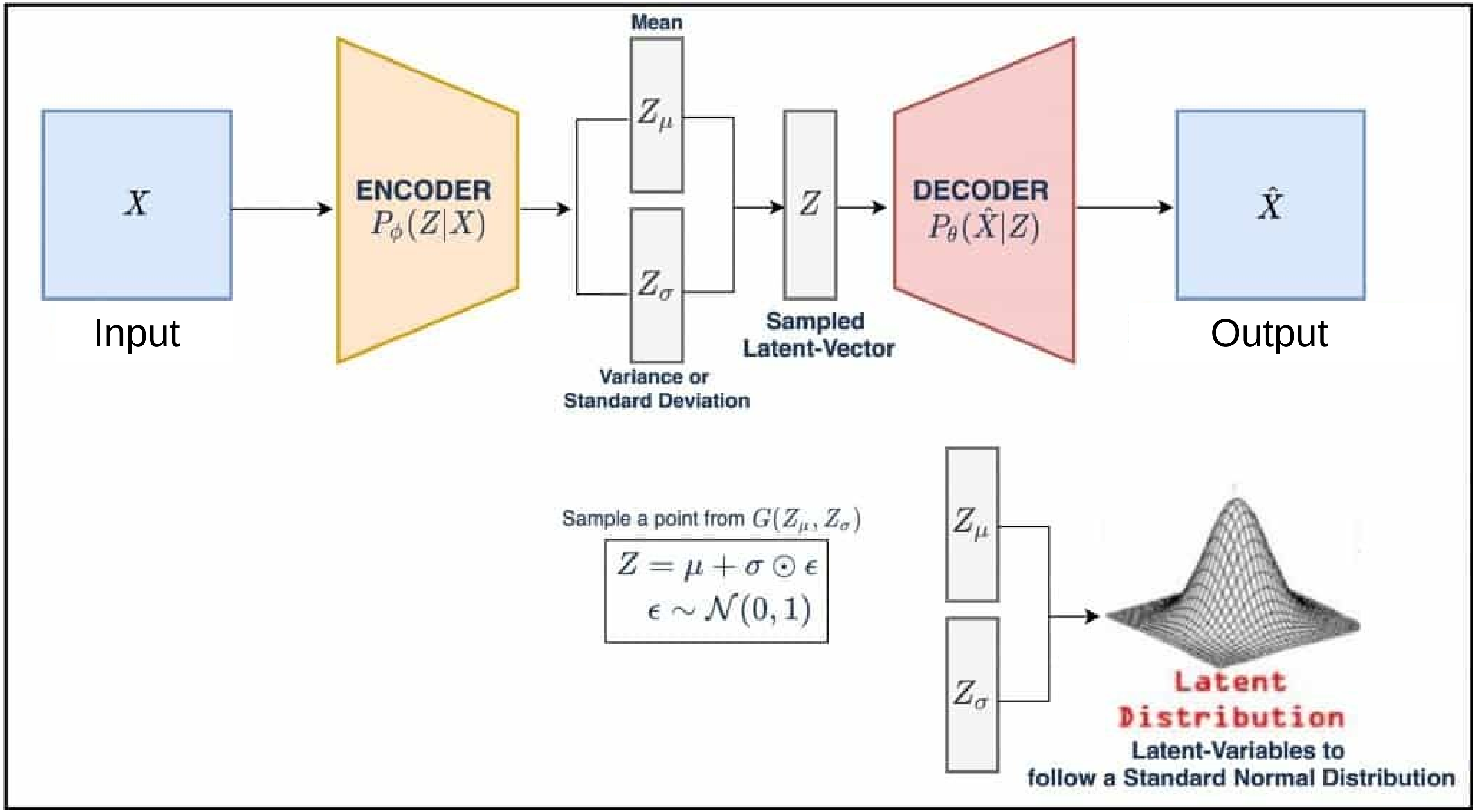
Pavlopoulos *et al* (2022)
doi:10.1007/s10115-022-01684-7



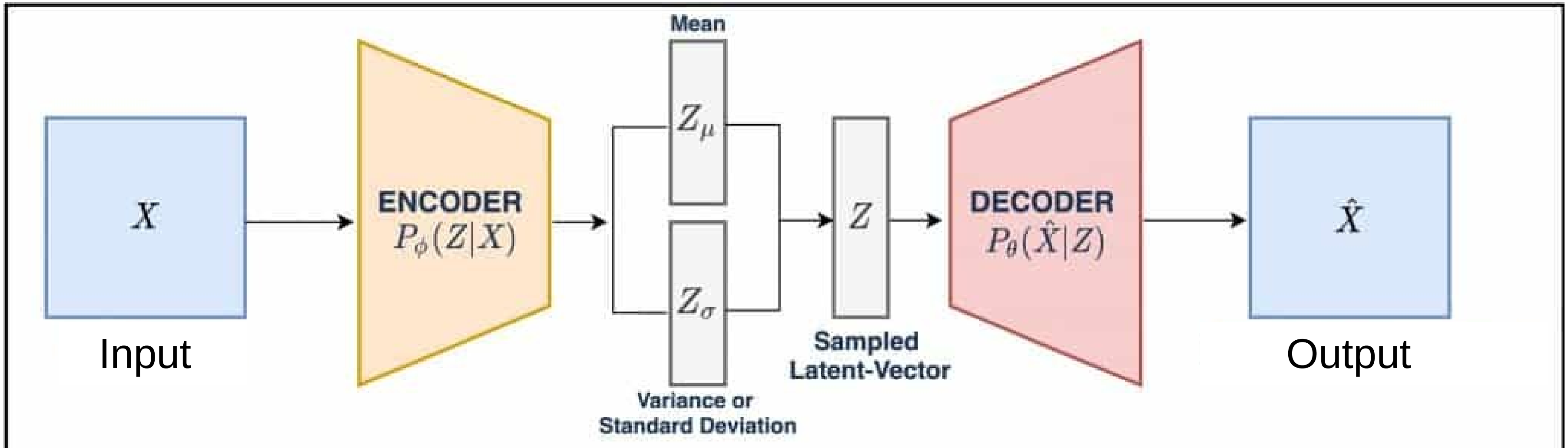
Learn by itself: AutoEncoder



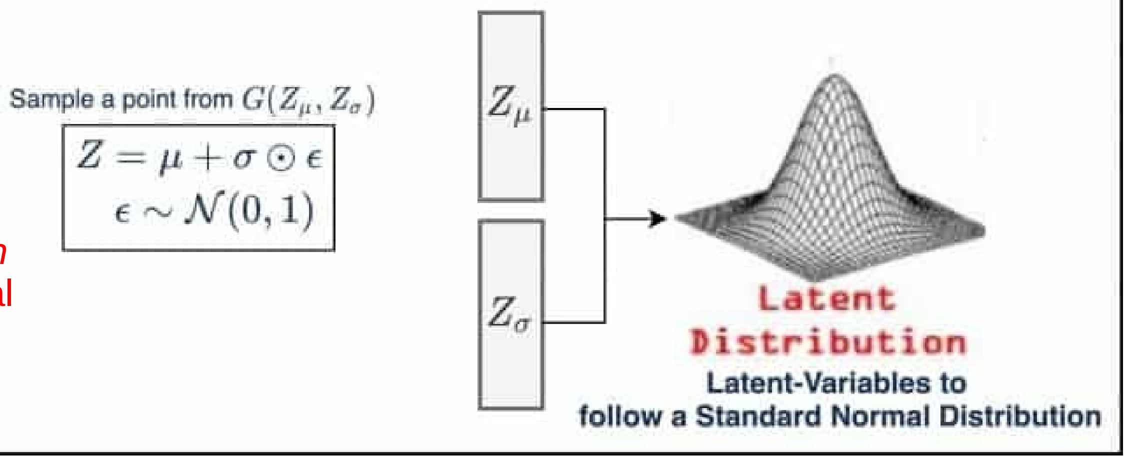
Variational Auto Encoder (VAE)



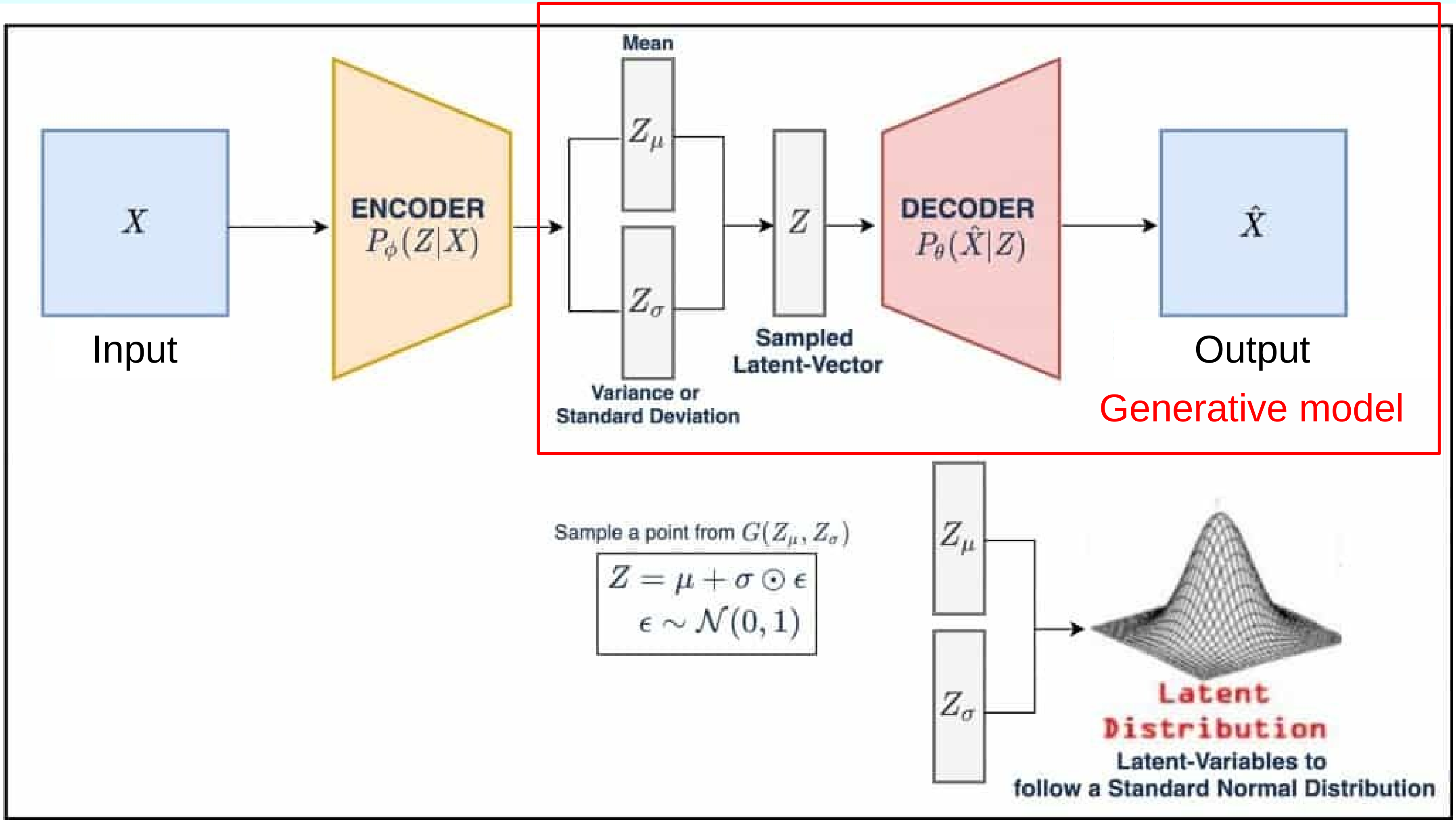
Variational Auto Encoder (VAE)



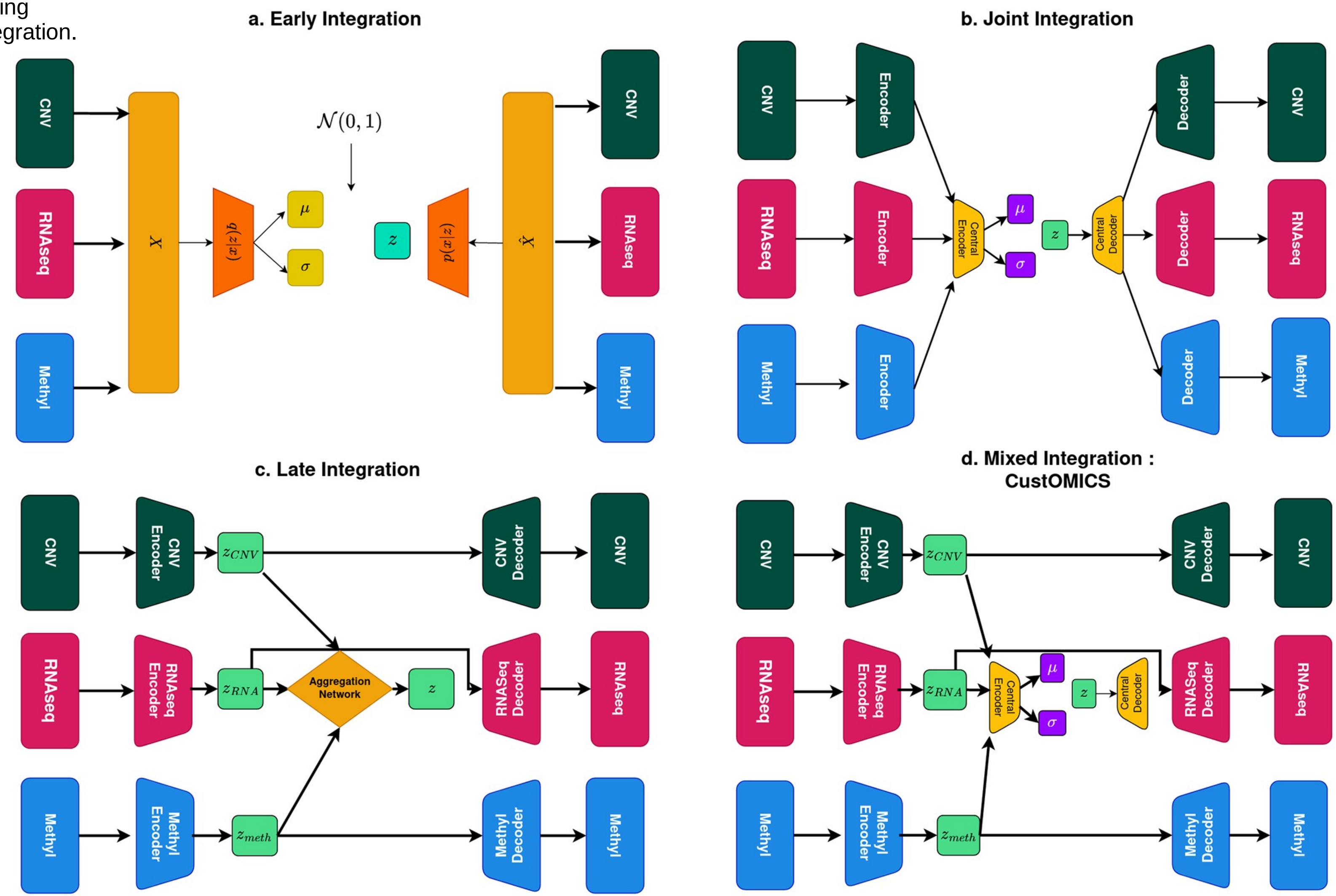
“Reparametrisation trick”
Upon learning, μ and σ
are guaranteed to be
mean and *standard deviation*
of a multi-dimensional normal
distribution



Variational Auto Encoder (VAE)



Benkirane, H., Pradat, Y.,
Michiels, S., Cournède, P. H. (2023).
CustOmics: A versatile deep-learning
based strategy for multi-omics integration.
PLoS Computational Biology,
19(3), e1010921.

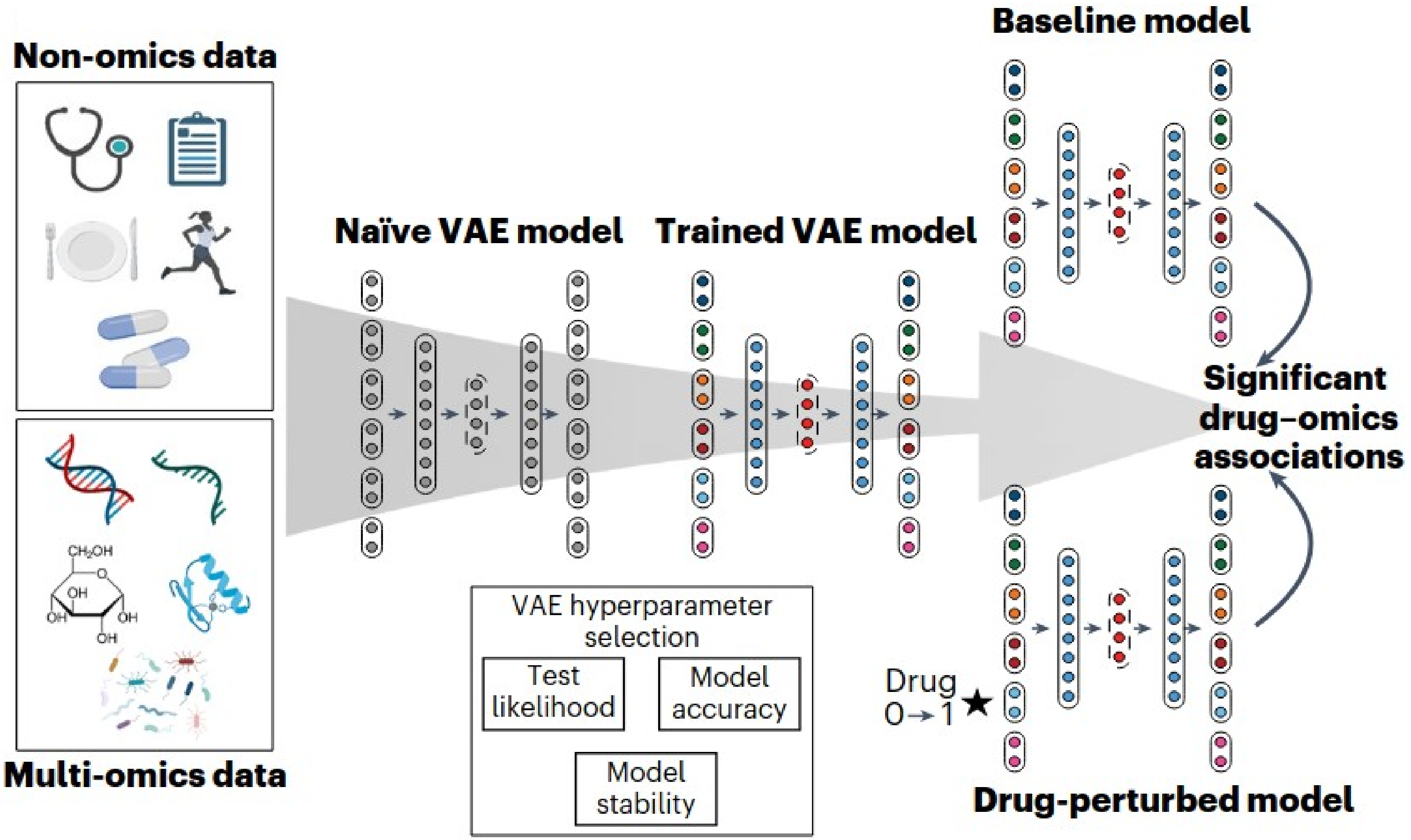


nature biotechnology

Article

https://doi.org/10.1038/s41587-022-01520-x

Discovery of drug–omics associations in type 2 diabetes with generative deep-learning models



7

Attention and the Transformer

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].

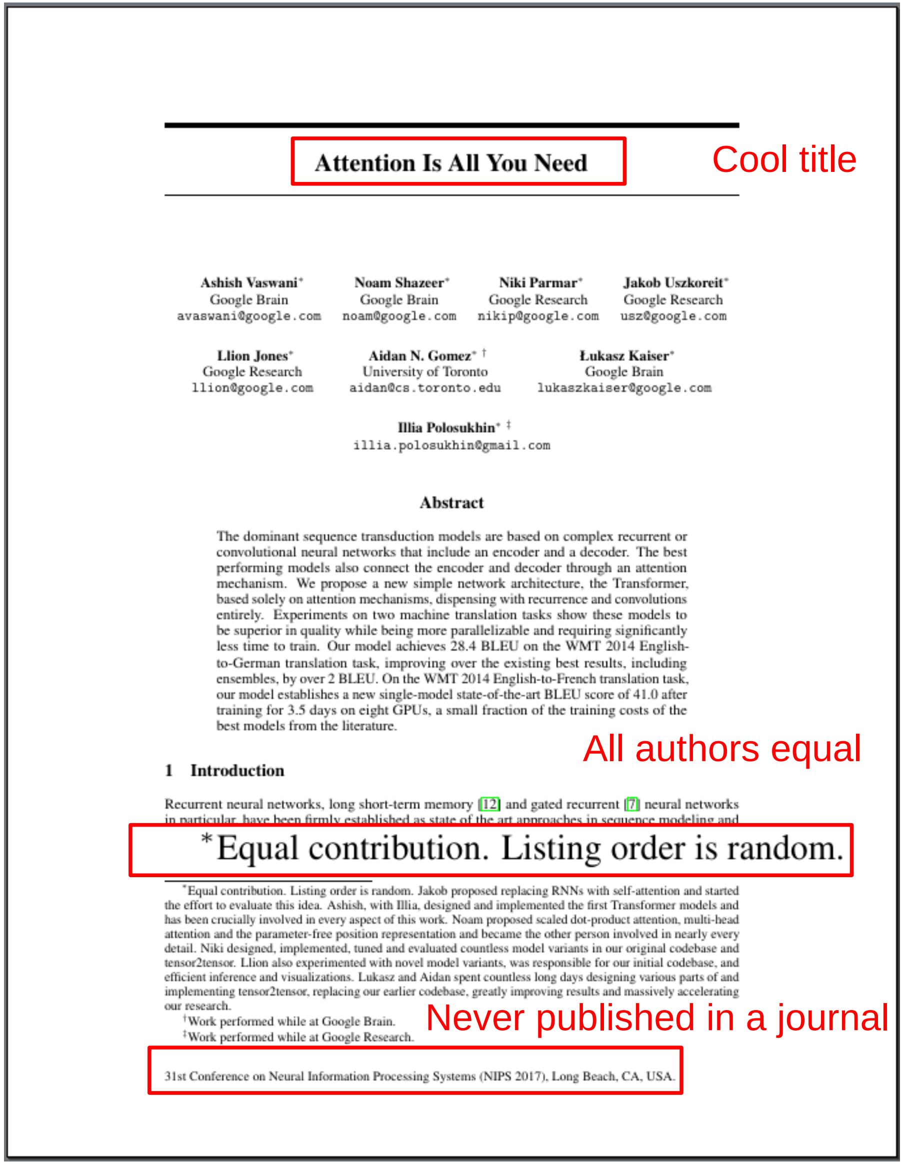
*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product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

Cited... 140833 times as of 13 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.

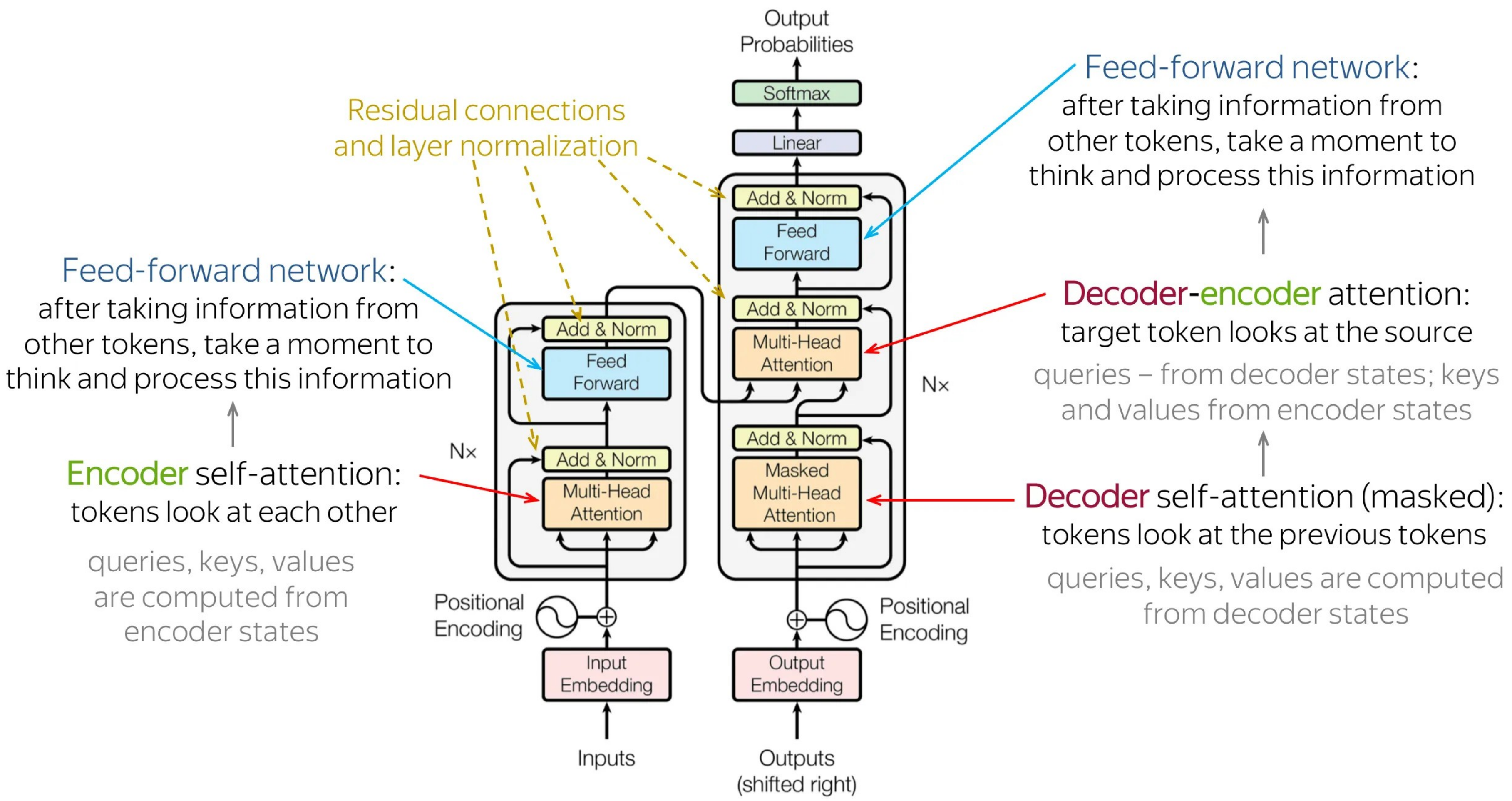
‡Work performed while at Google Research.

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

Claude

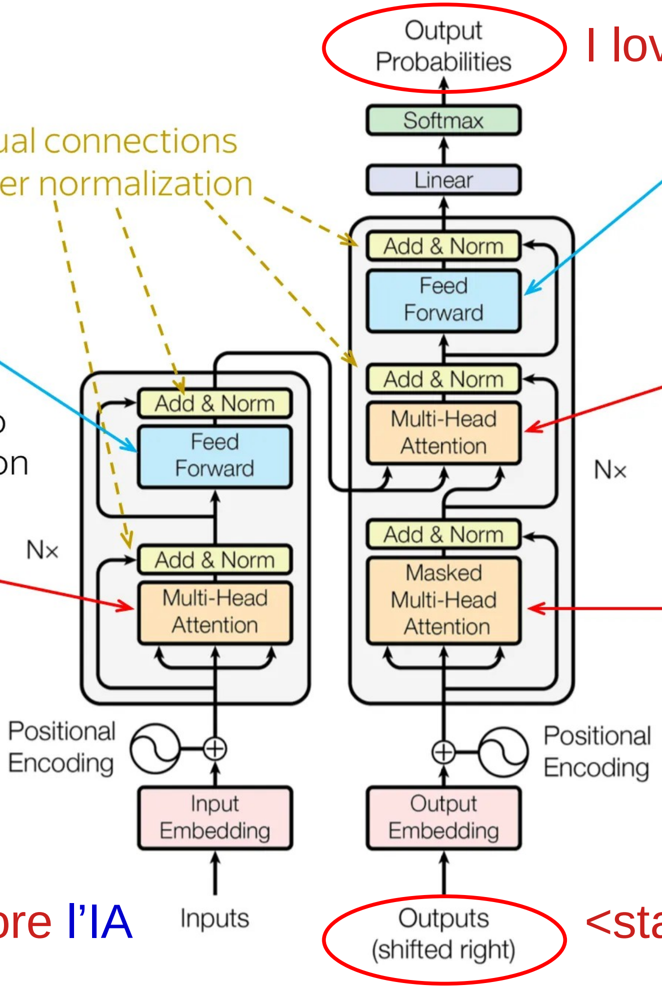
AlphaFold

The Transformer: Memory + context = attention



The Transformer: Memory + context = attention

Residual connections and layer normalization



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

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

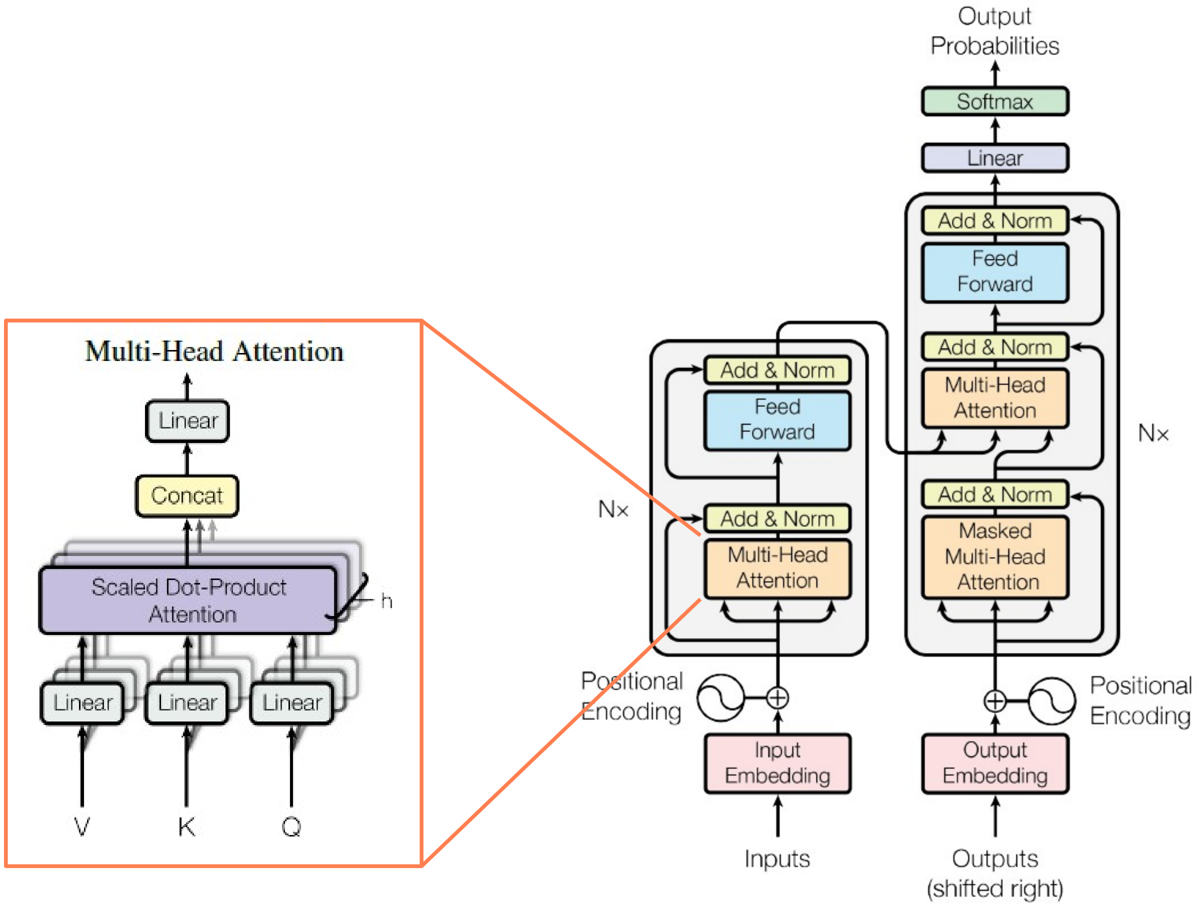
Encoder self-attention: tokens look at each other

queries, keys, values are computed from encoder states

J'adore l'IA

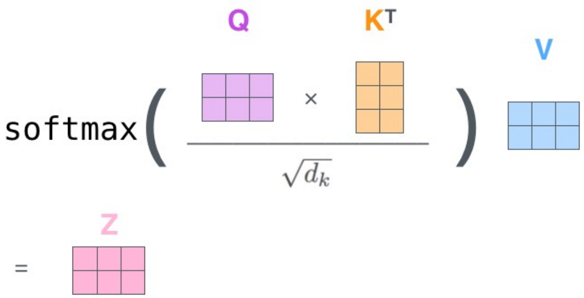
<start> I love

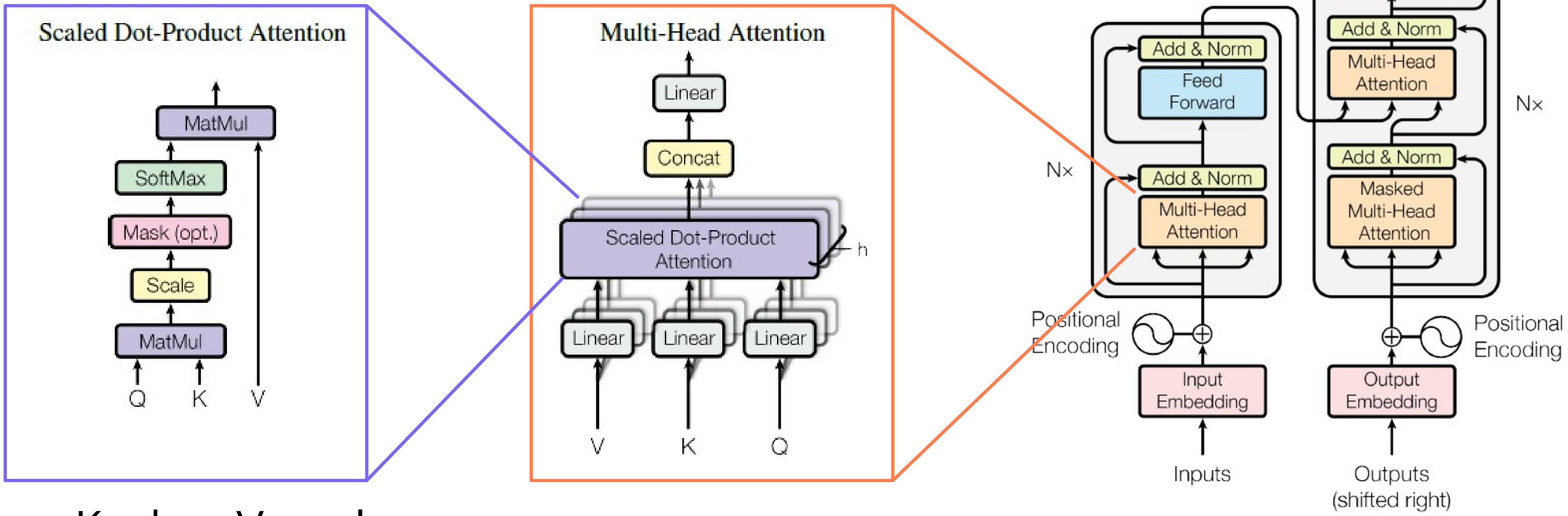
Attention in the Transformer



Q = query, K = key, V = value

Attention in the Transformer



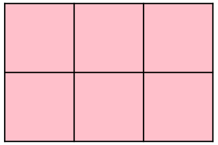


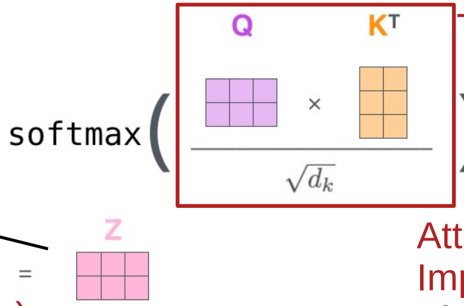
Q = query, K = key, V = value

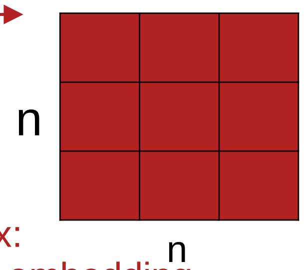
Attention in the Transformer

n = #tokens

d = dimension
of embeddings

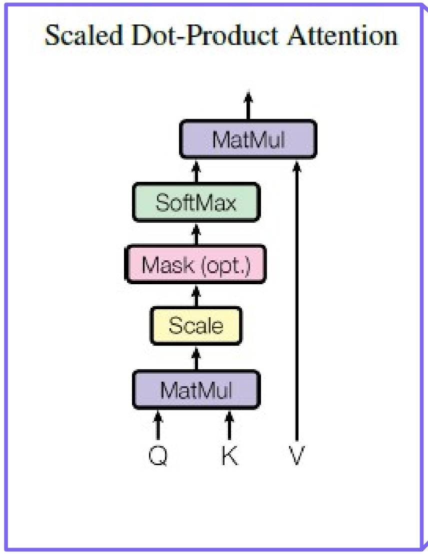


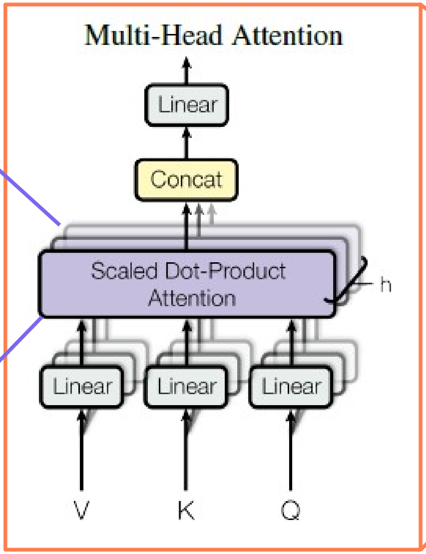


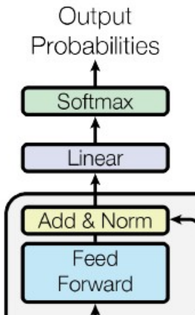


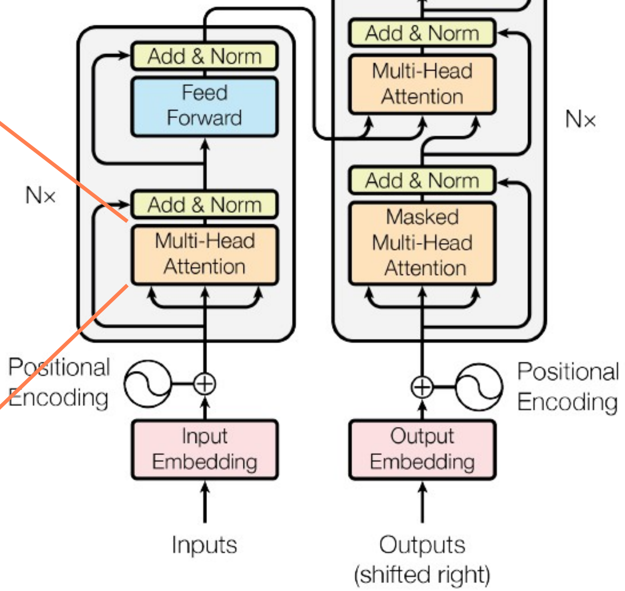
Attention: aggregated attention
per embedding (~per token)

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





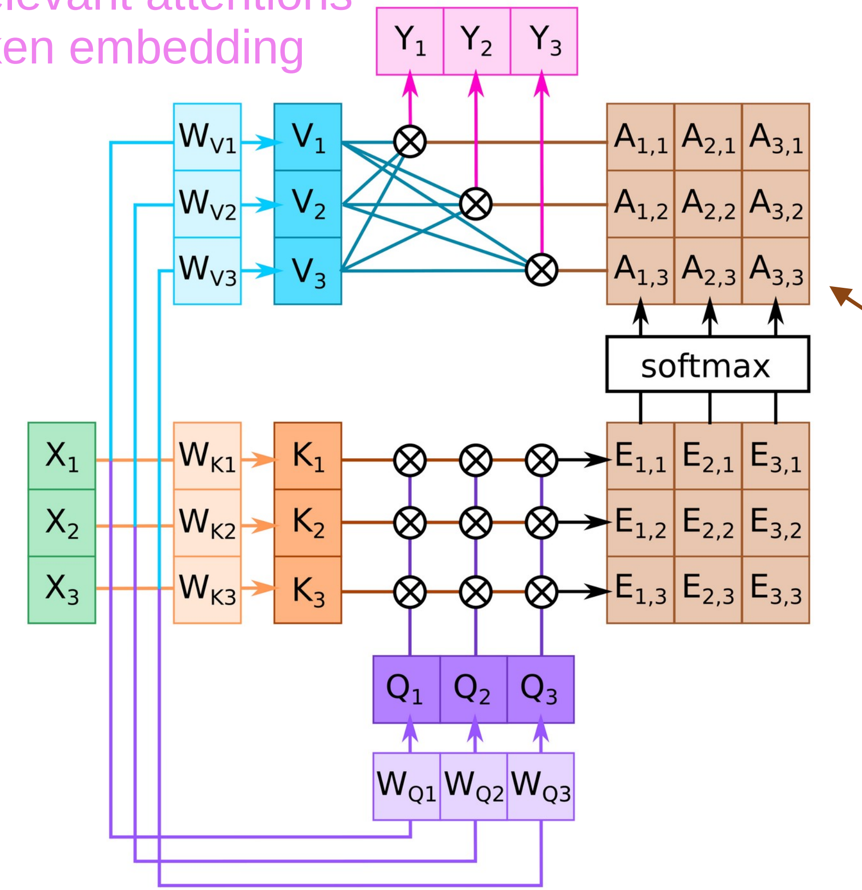




Q = query, K = key, V = value

Attention in the Transformer

Actual relevant attentions
1 per token embedding



Query: What are the things I am looking for?

Key: What are the thing I have?

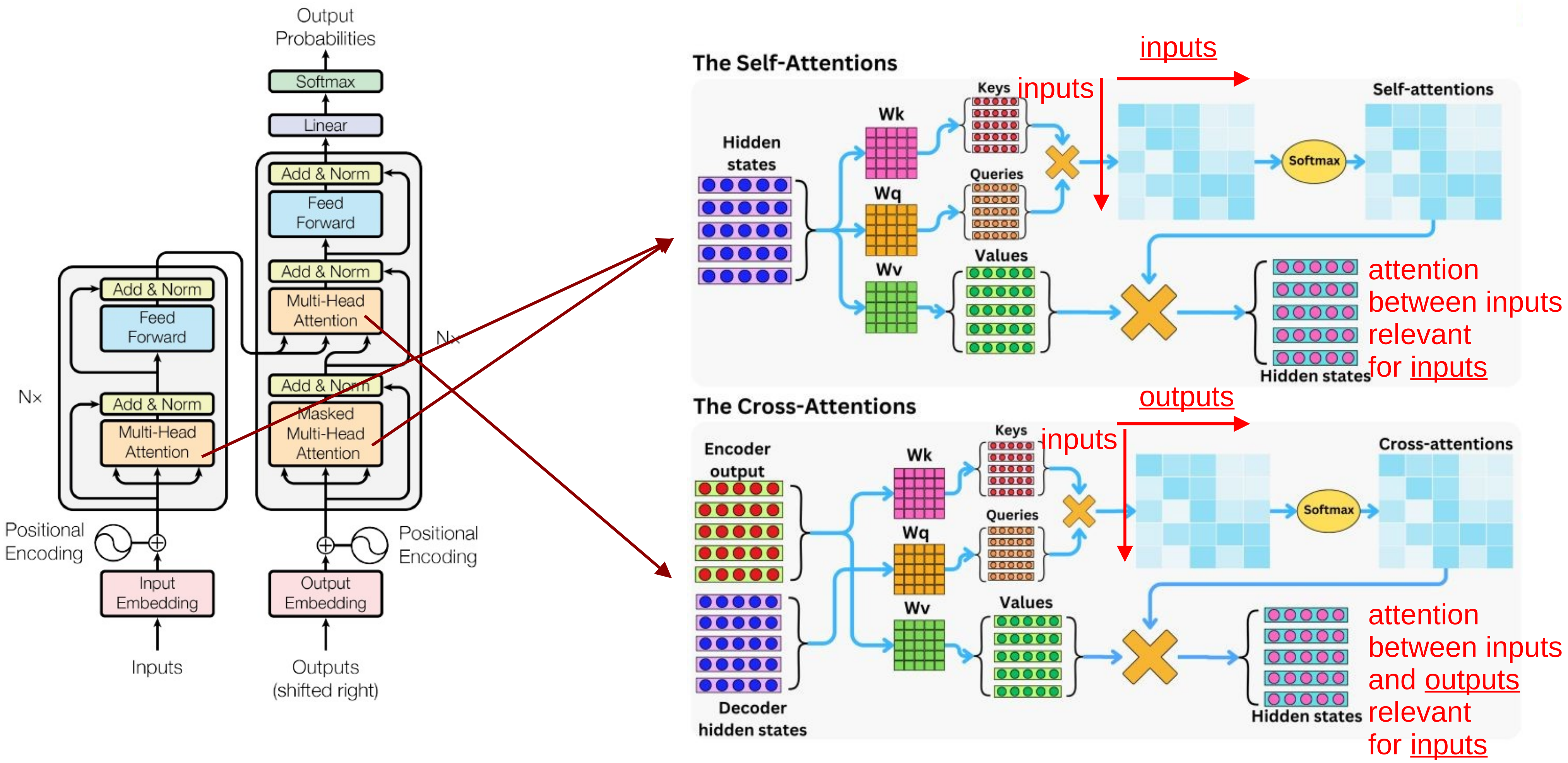
Value: What are the things that I will communicate?

The attention of all token embeddings
on all token embeddings

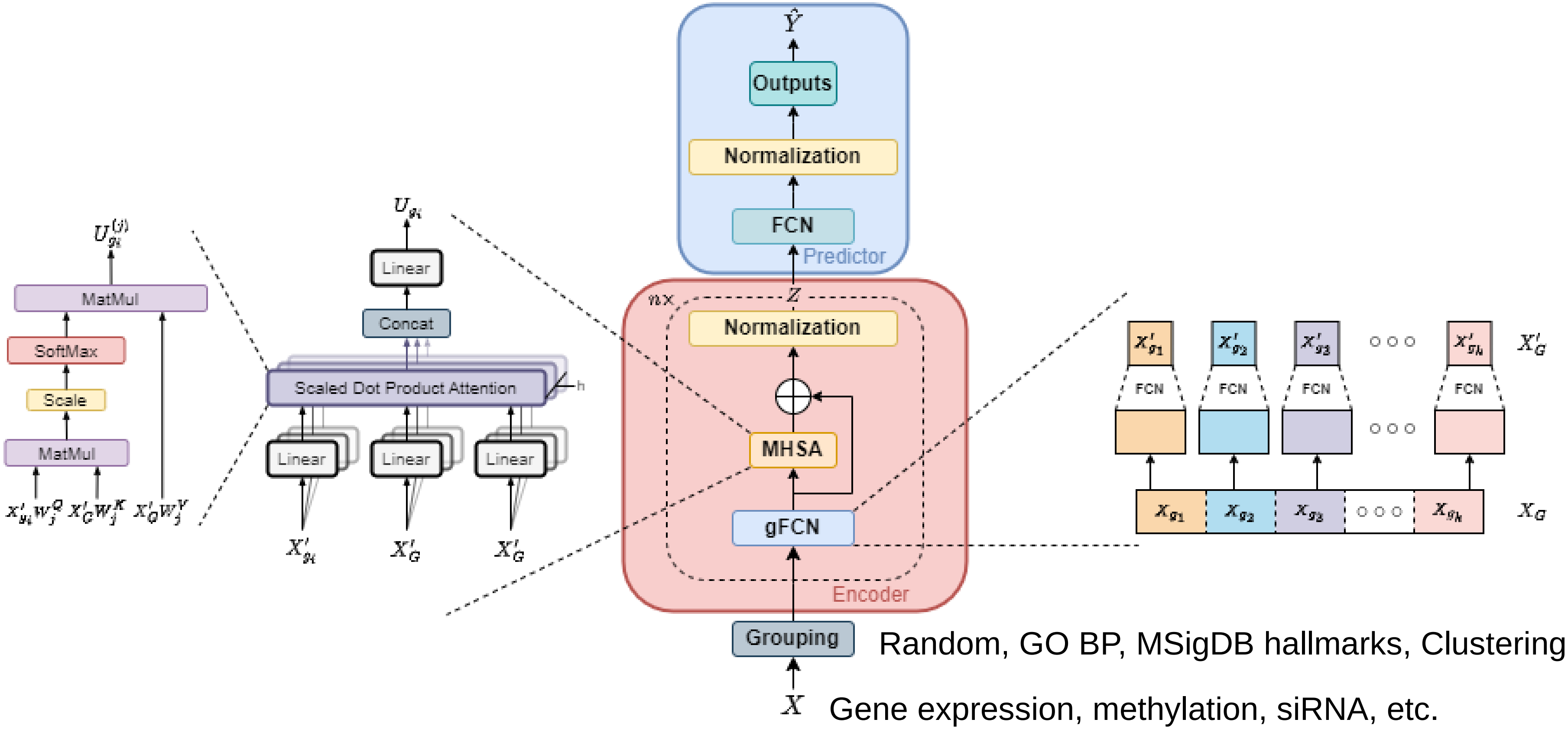
X is entered
W_Q, W_K, and W_V are learned
Everything else is computed

Only one sequence (X) as input,
transformed into different sequences
of embeddings when multiplied
by the learned weights W_Q, W_K, and W_V

Self versus Cross-attention



AttOmics

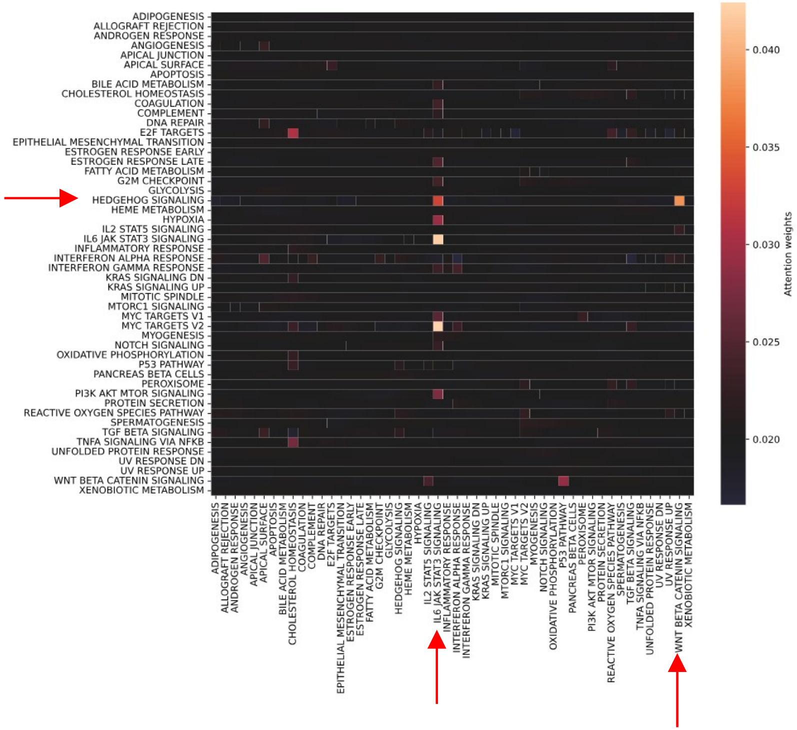


Random, GO BP, MSigDB hallmarks, Clustering

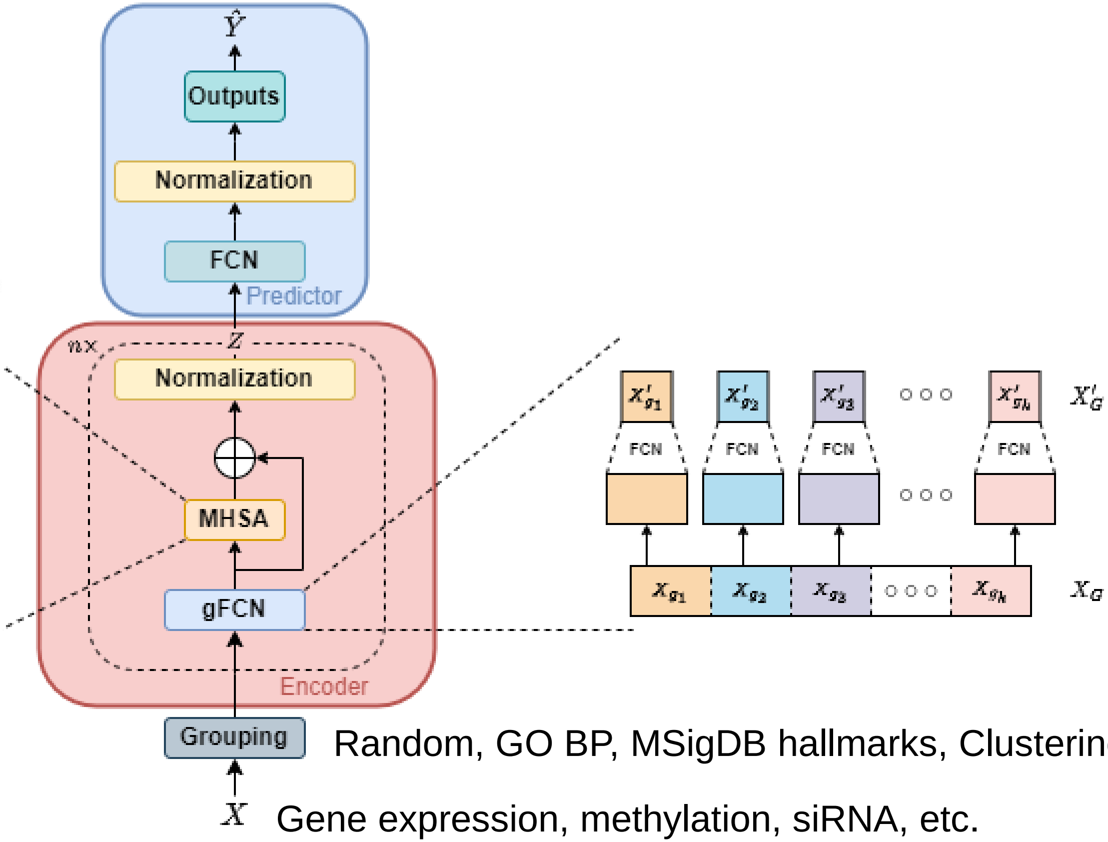
Gene expression, methylation, siRNA, etc.
e.g., TCGA cervical cancers

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

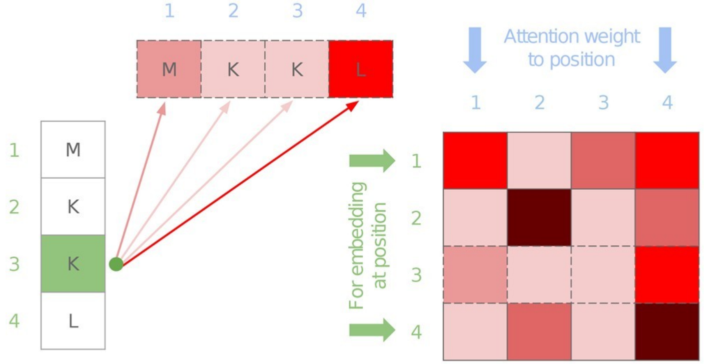
Beaude, A., Rafiee Vahid, M., Augé, F., Zehraoui, F., & Hanczar, B. (2023). AttOmics: attention-based architecture for diagnosis and prognosis from omics data. *Bioinformatics*, 39(Supplement_1), i94-i102.

EnzBERT

Predicting enzymatic function of protein sequences with attention

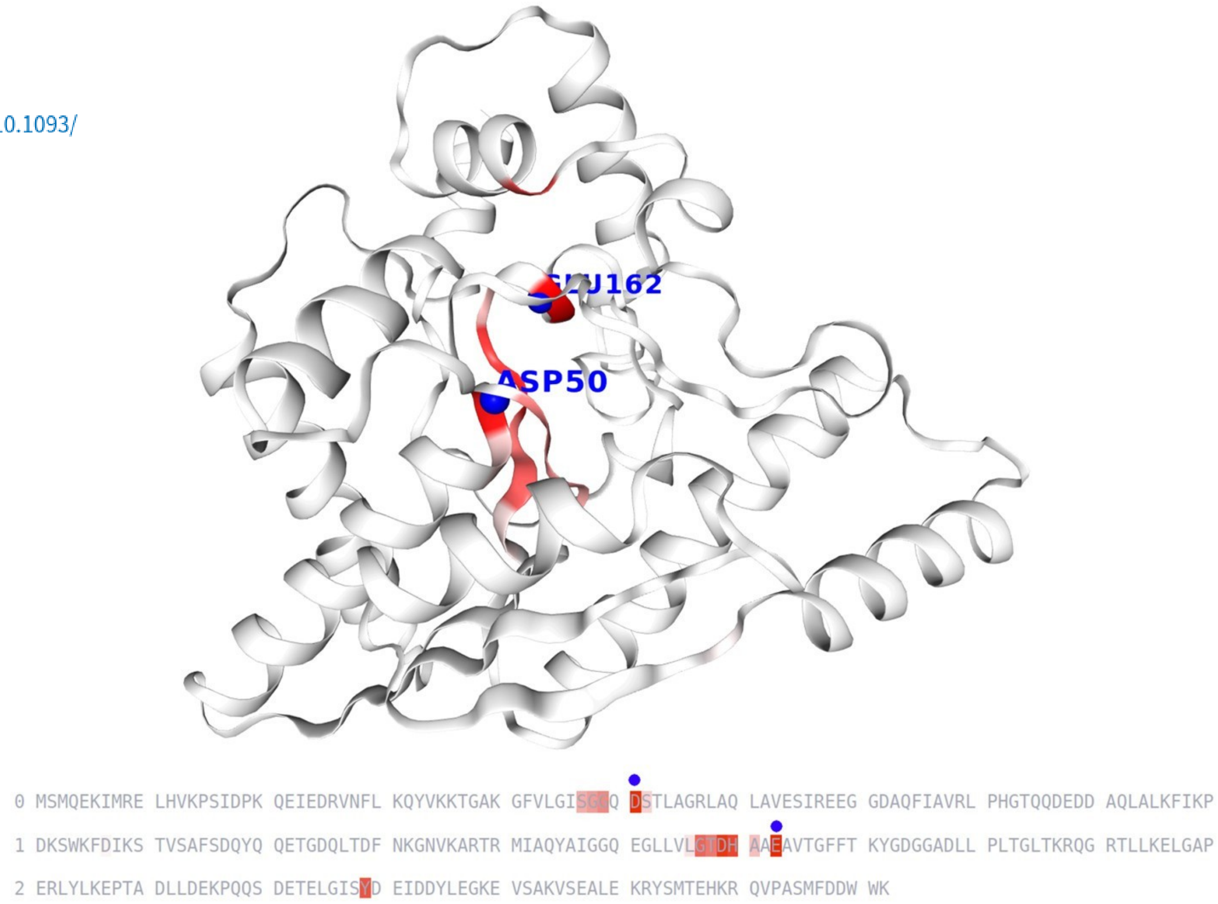
Nicolas Buton, François Coste, Yann Le Cunff

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

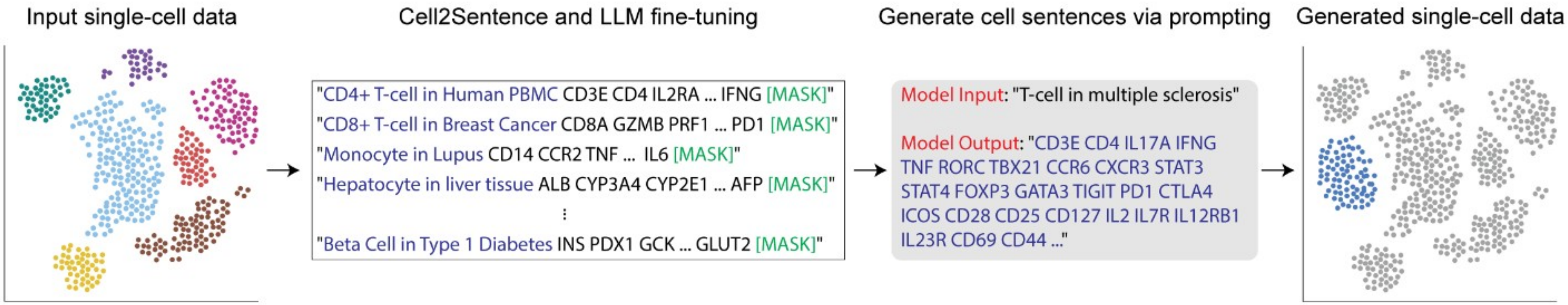


Aggregated attention
for each token (amino acid)

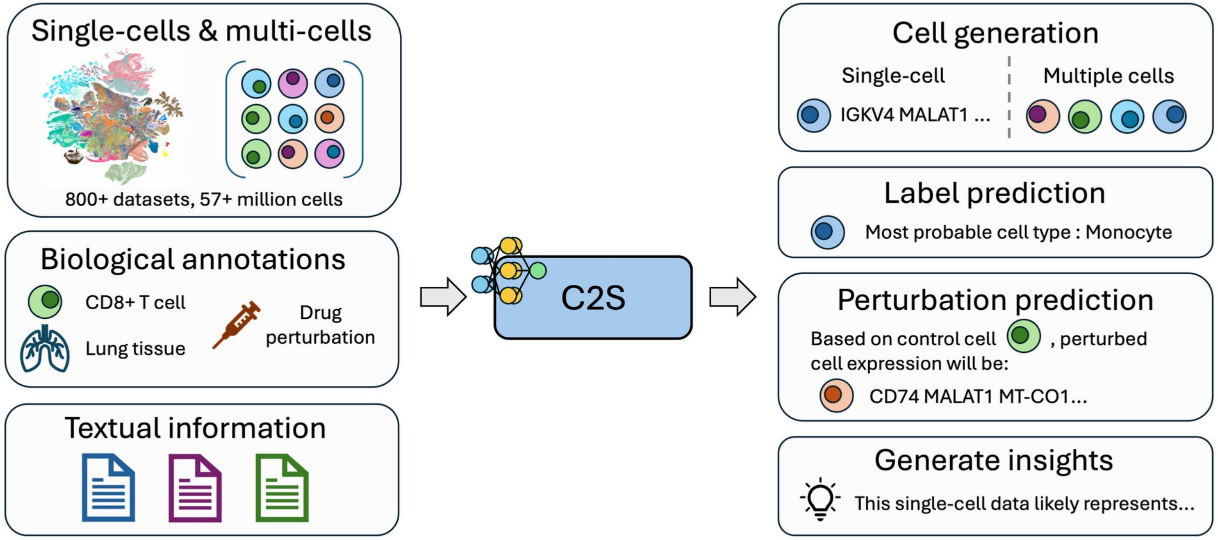
Nh(3)-dependent nad(+) synthetase



Cell2Sentence

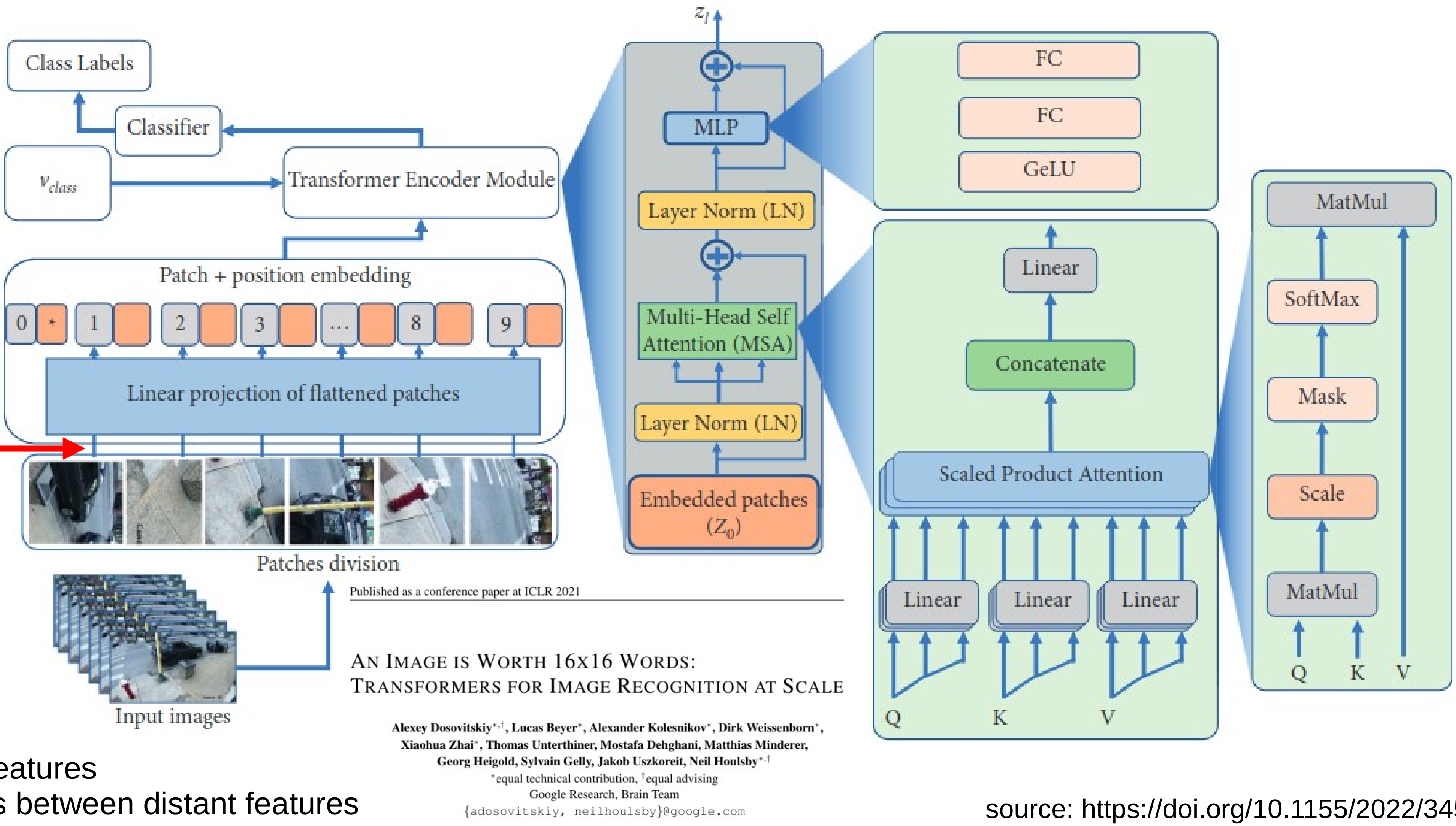


Levine *et al* (2024). Cell2Sentence: Teaching Large Language Models the Language of Biology. *BioRxiv* https://doi.org/10.1101/2023.09.11.557287



Vision Transformer

Patch
embedding
by a CNN

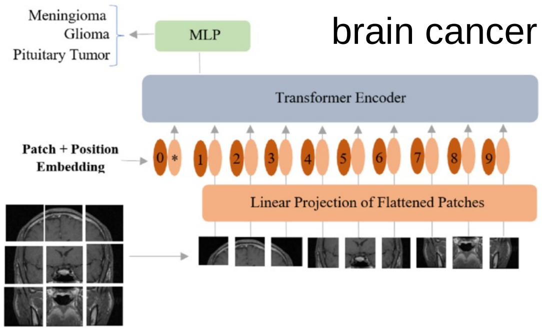


CNN = local features
ViT = relations between distant features

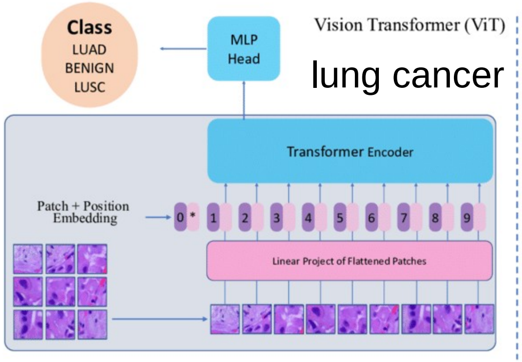
source: https://doi.org/10.1155/2022/3454167

ViTs are replacing vanilla CNNs

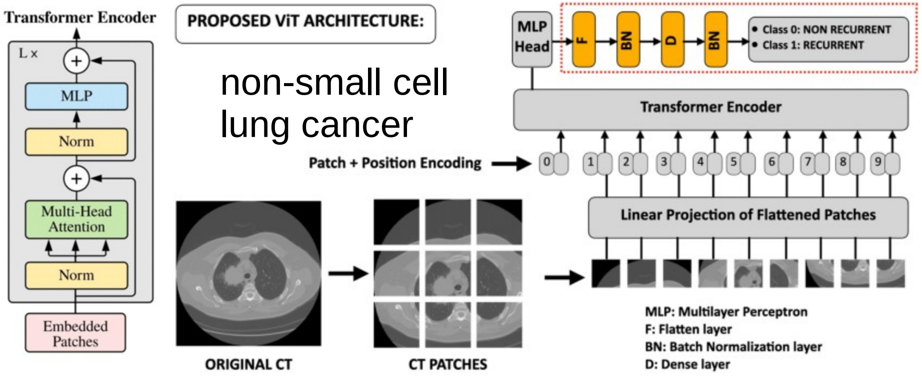
brain cancer

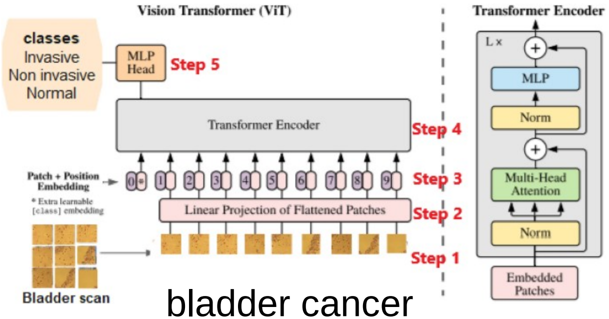


lung cancer



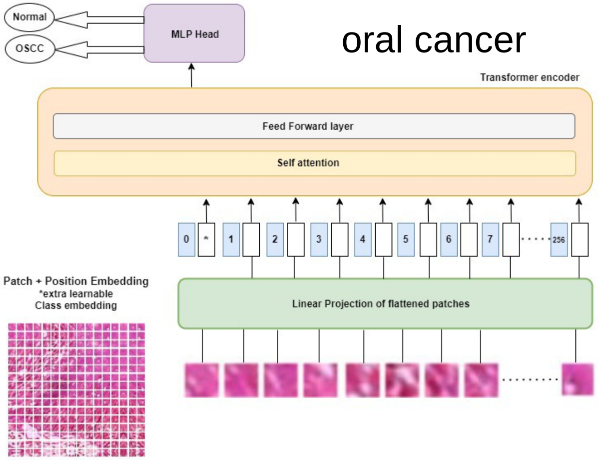
non-small cell lung cancer

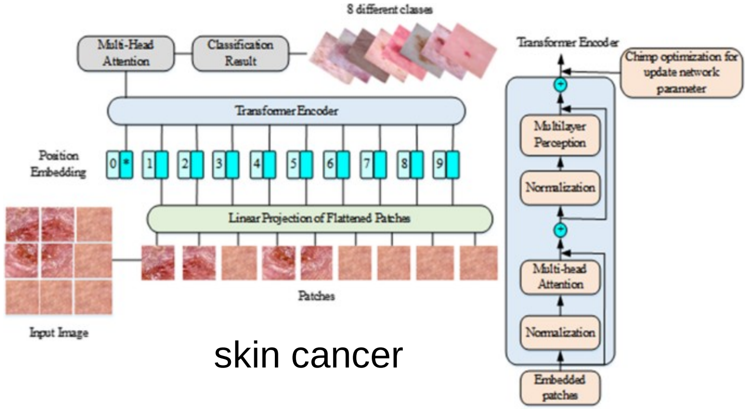




bladder cancer

oral cancer





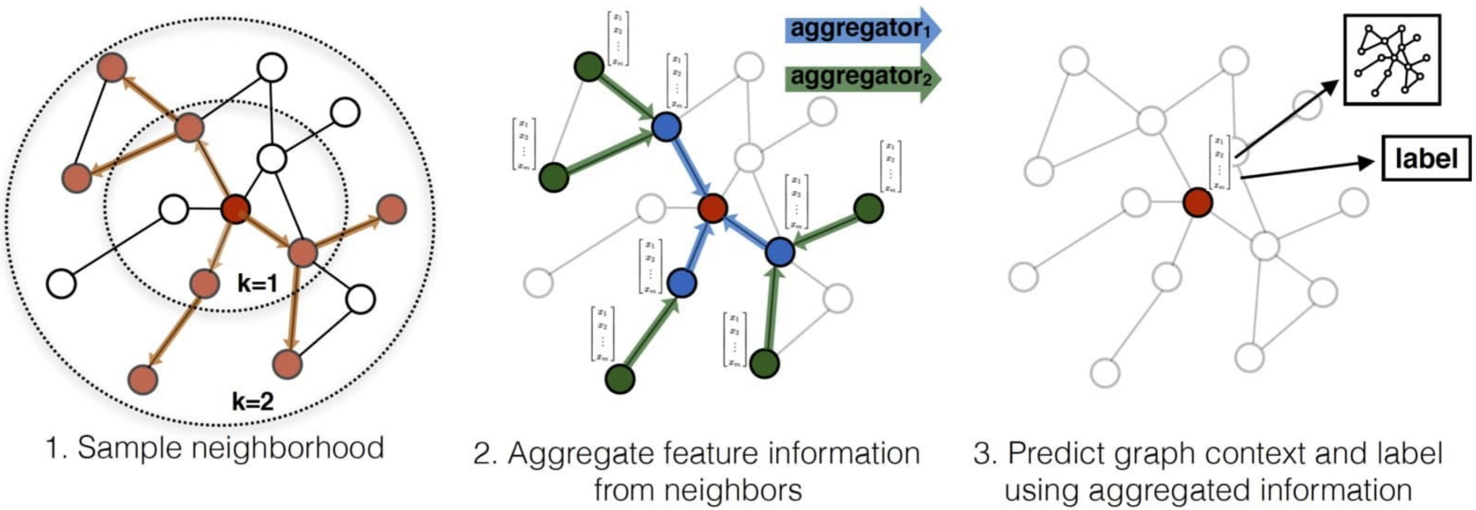
skin cancer

8

Graph Neural Networks (GNNs)

Graph Neural Networks (GNNs)

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



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

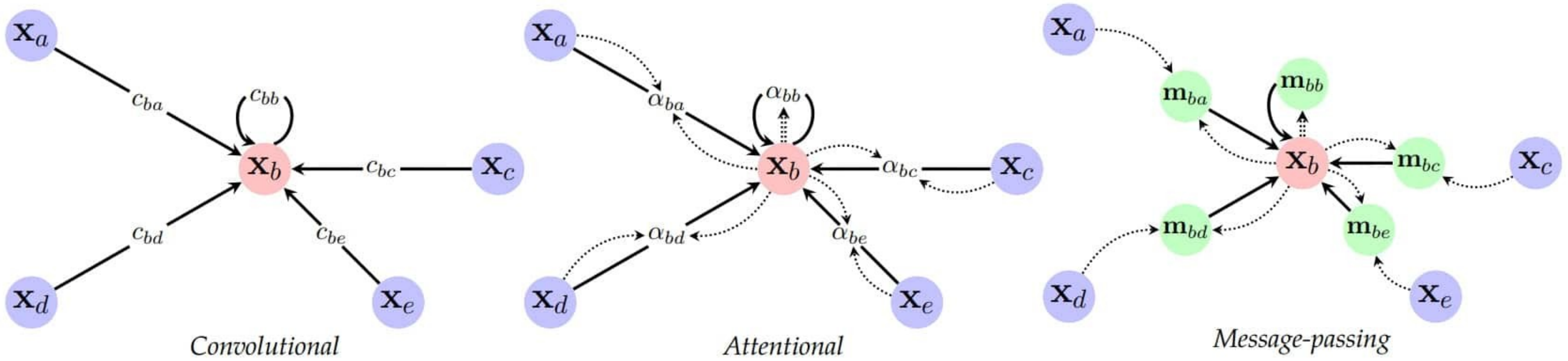
Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering

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

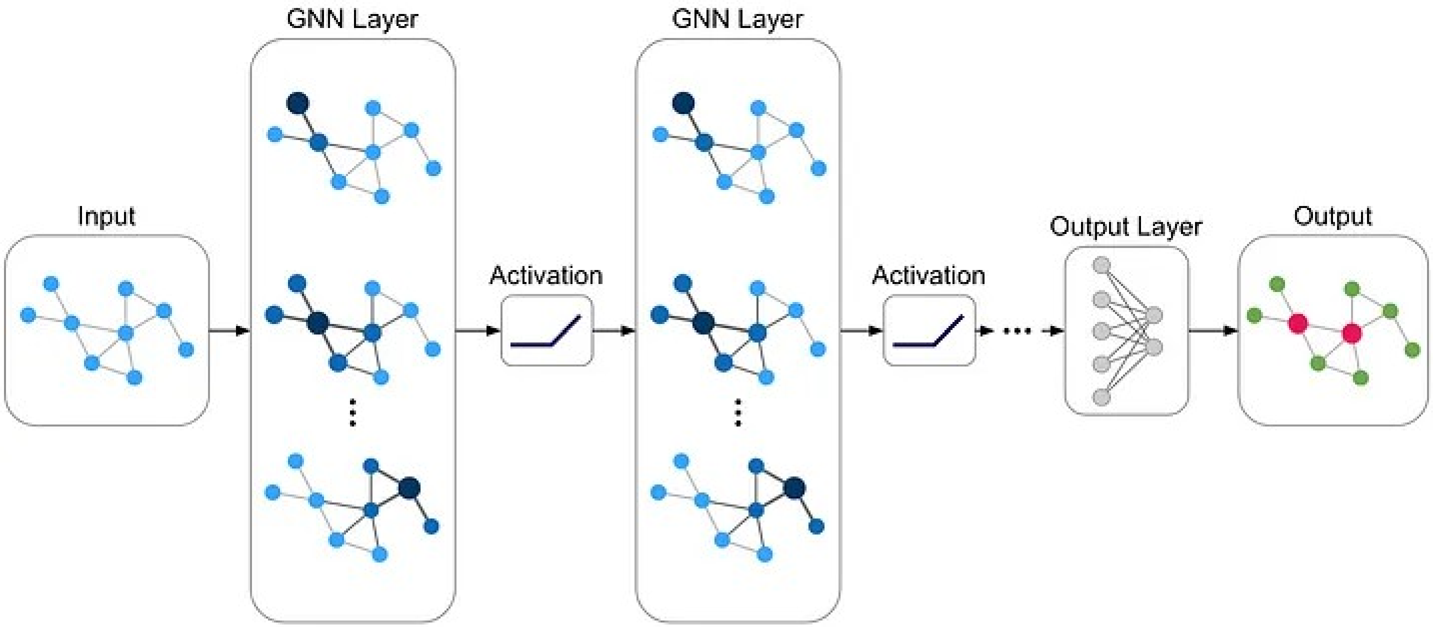
30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Many different ways to update GNNs

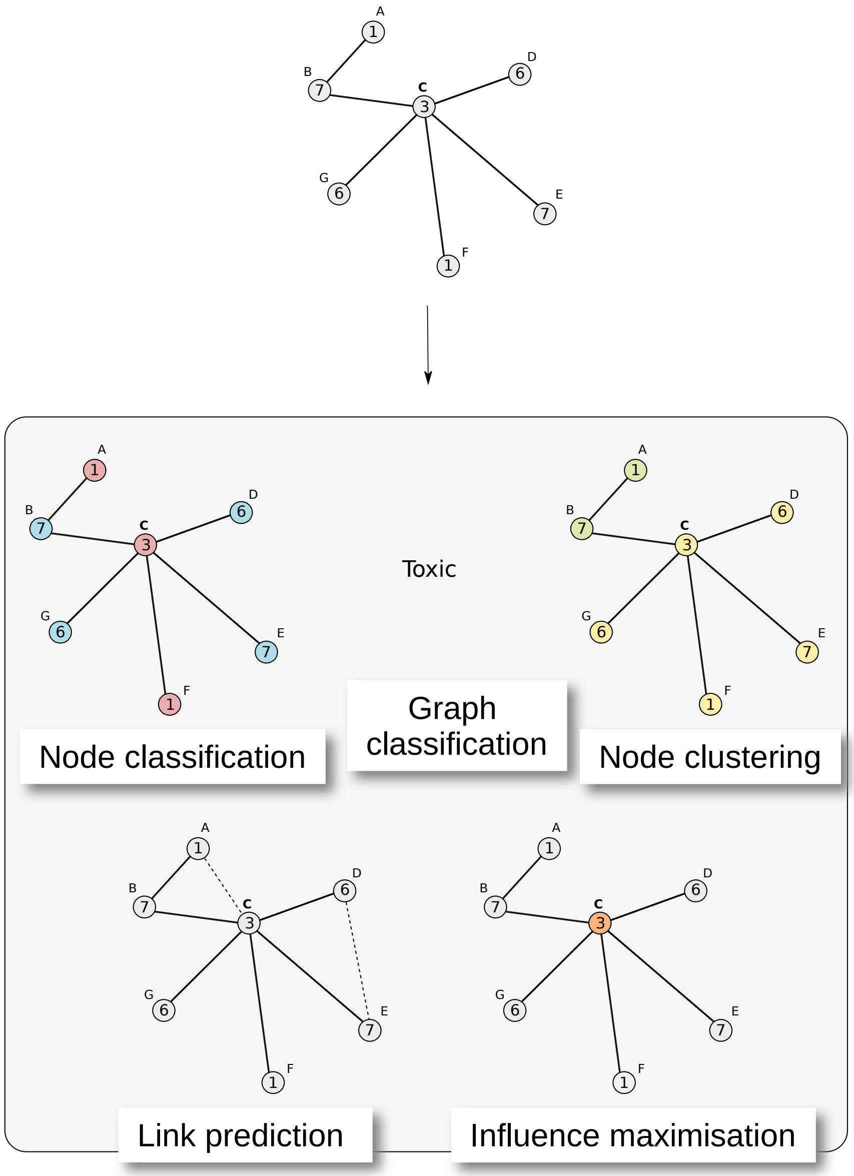
source: https://blogs.nvidia.com/blog/what-are-graph-neural-networks/



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/

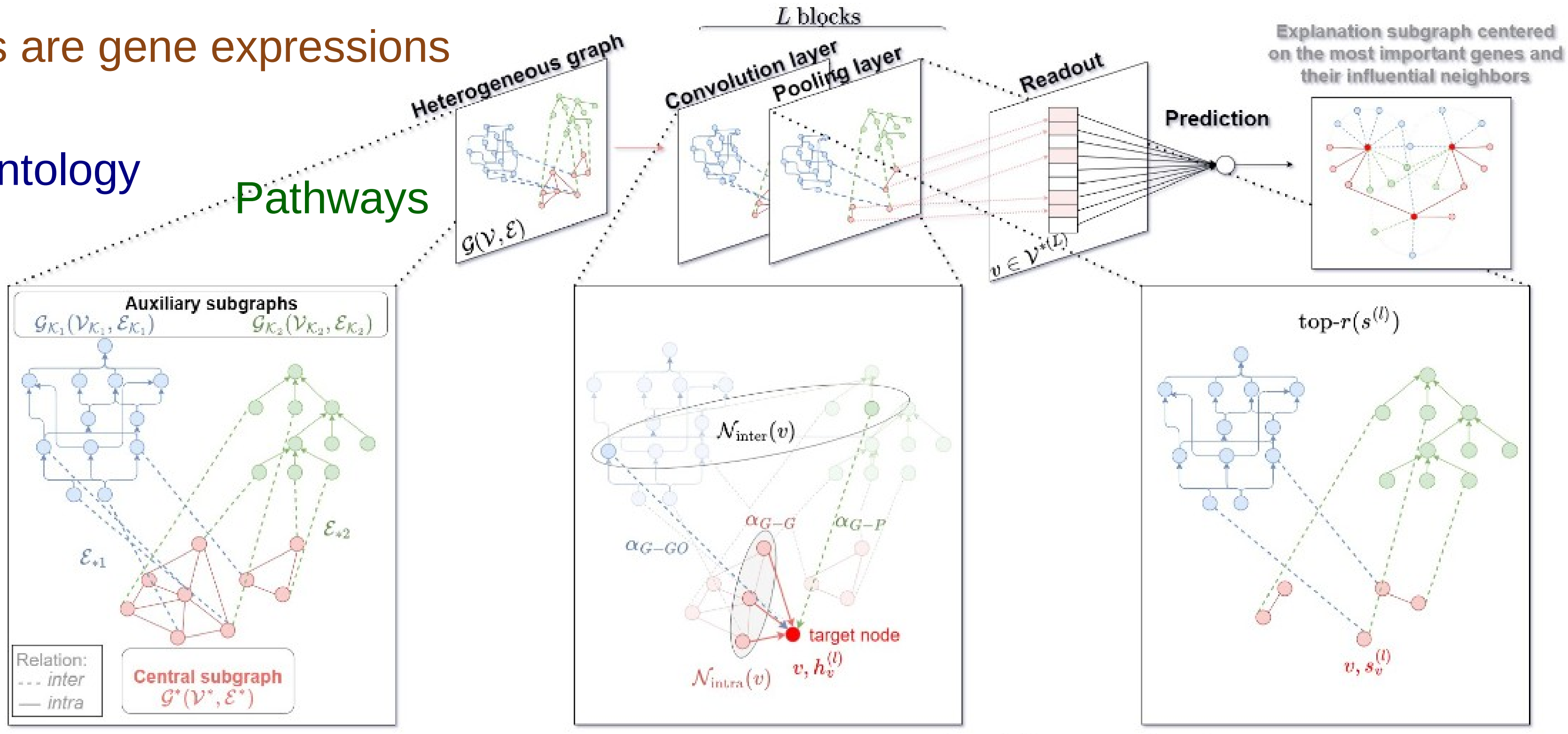
Both by Google Research teams

GNN can be heterogeneous

Inputs are gene expressions

Gene ontology

Pathways



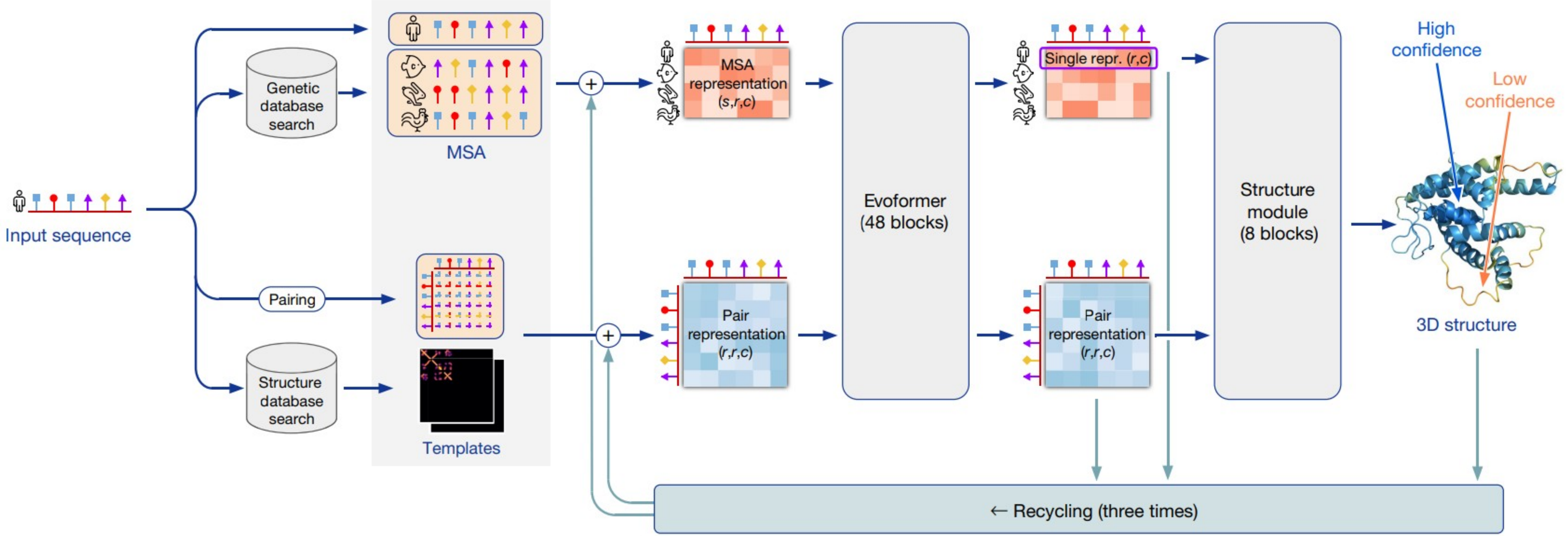
BioHAN: a Knowledge-based Heterogeneous Graph Neural Network for precision medicine on transcriptomic data

Victoria Bourgeais, Farida Zehraoui, Blaise Hanczar https://hal.science/hal-04092210/

9

Alphafold2

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

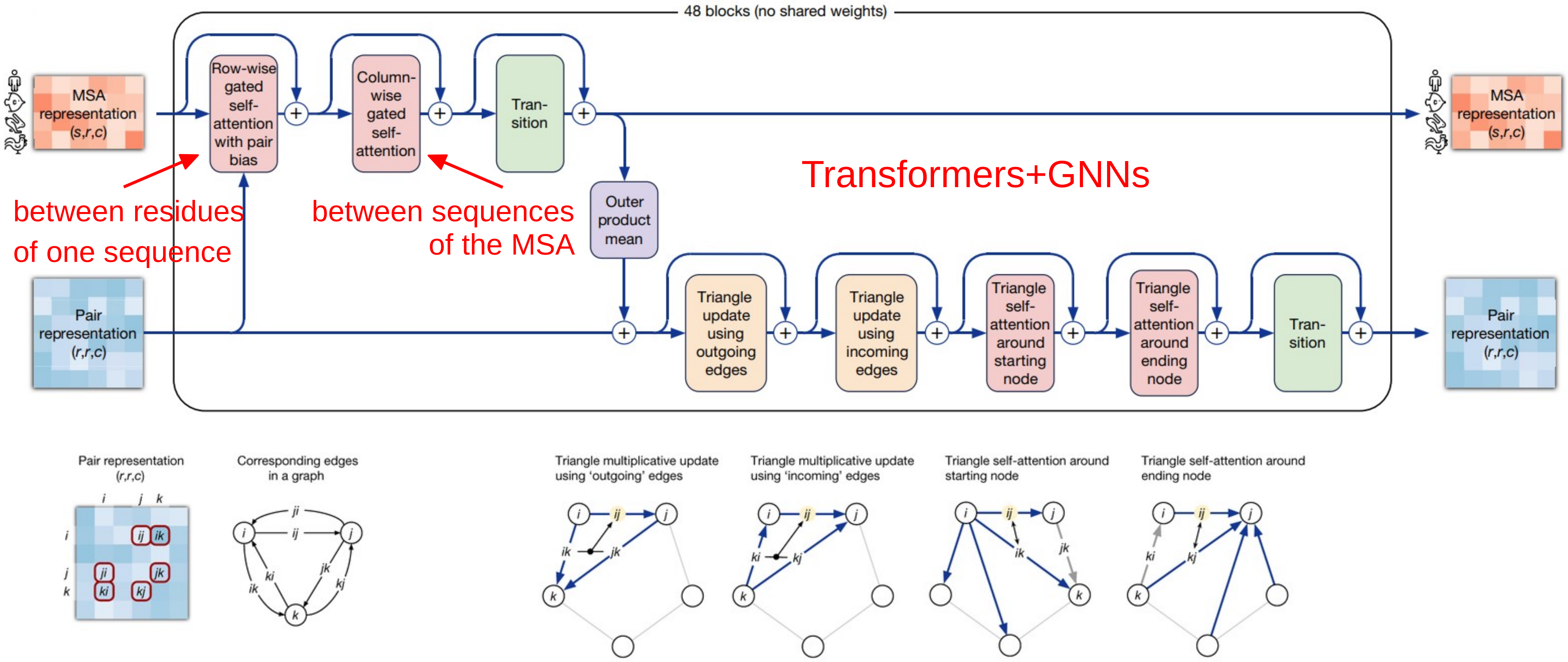
Check for updates

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



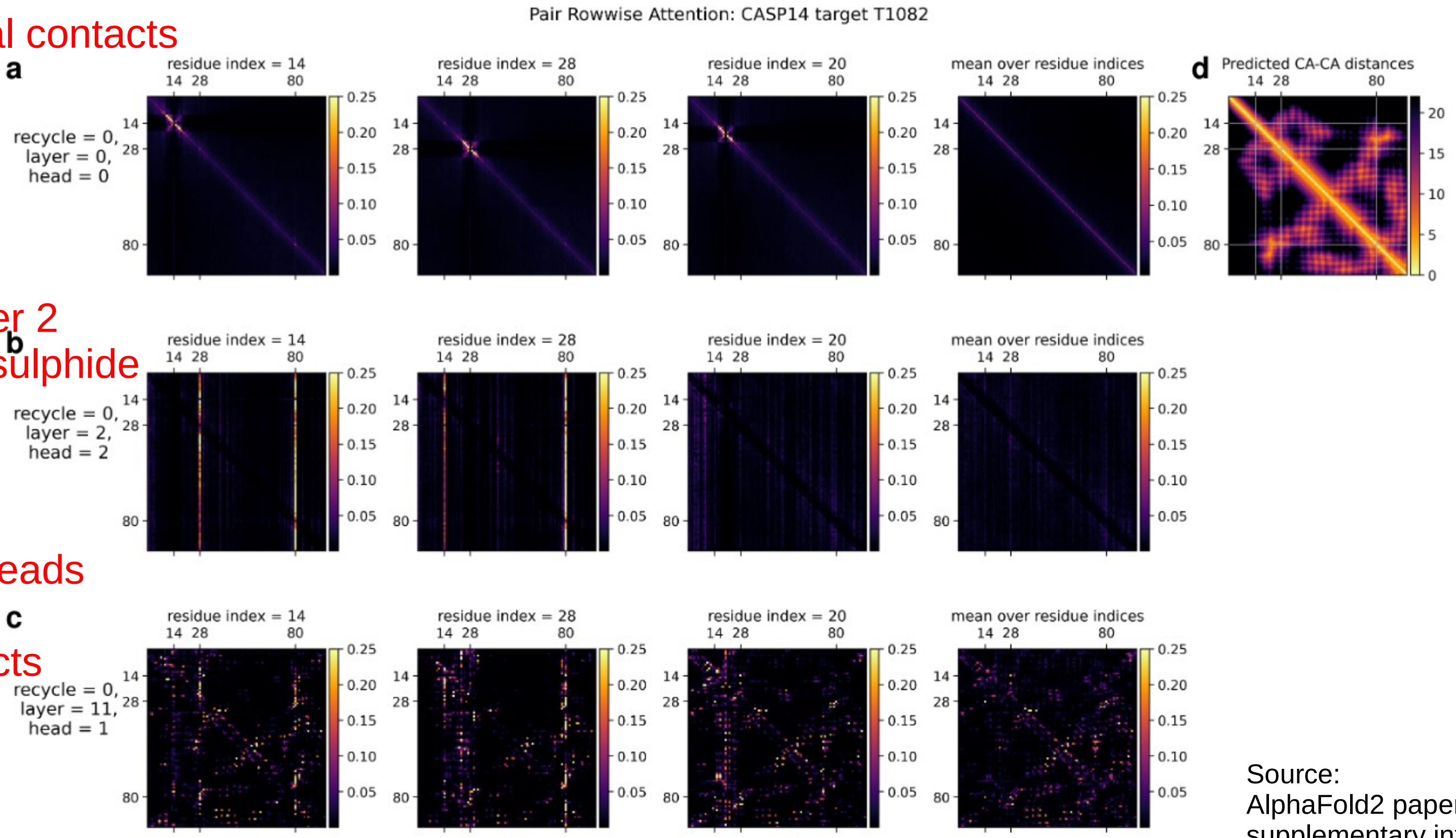
see also: https://www.blopig.com/blog/2021/07/alphafold-2-is-here-whats-behind-the-structure-prediction-miracle/

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

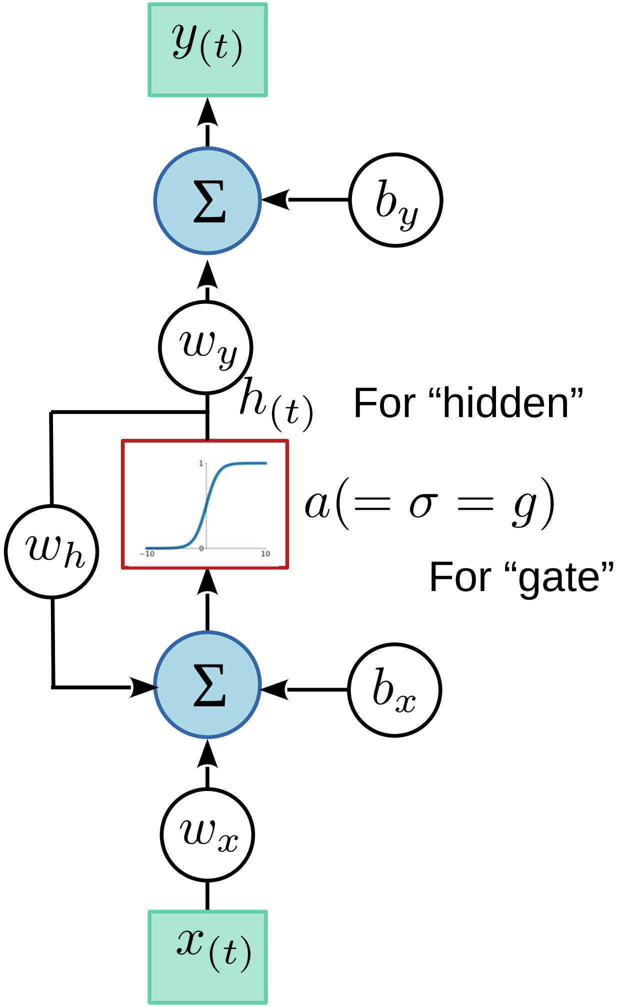
10

Analysing and predicting series:

Recurrent Neural Networks (RNNs)

Long Short-Term Memory (LSTM)

RNN: 1 cell (here, 1 neuron)



NB: implicit "identity" activation function

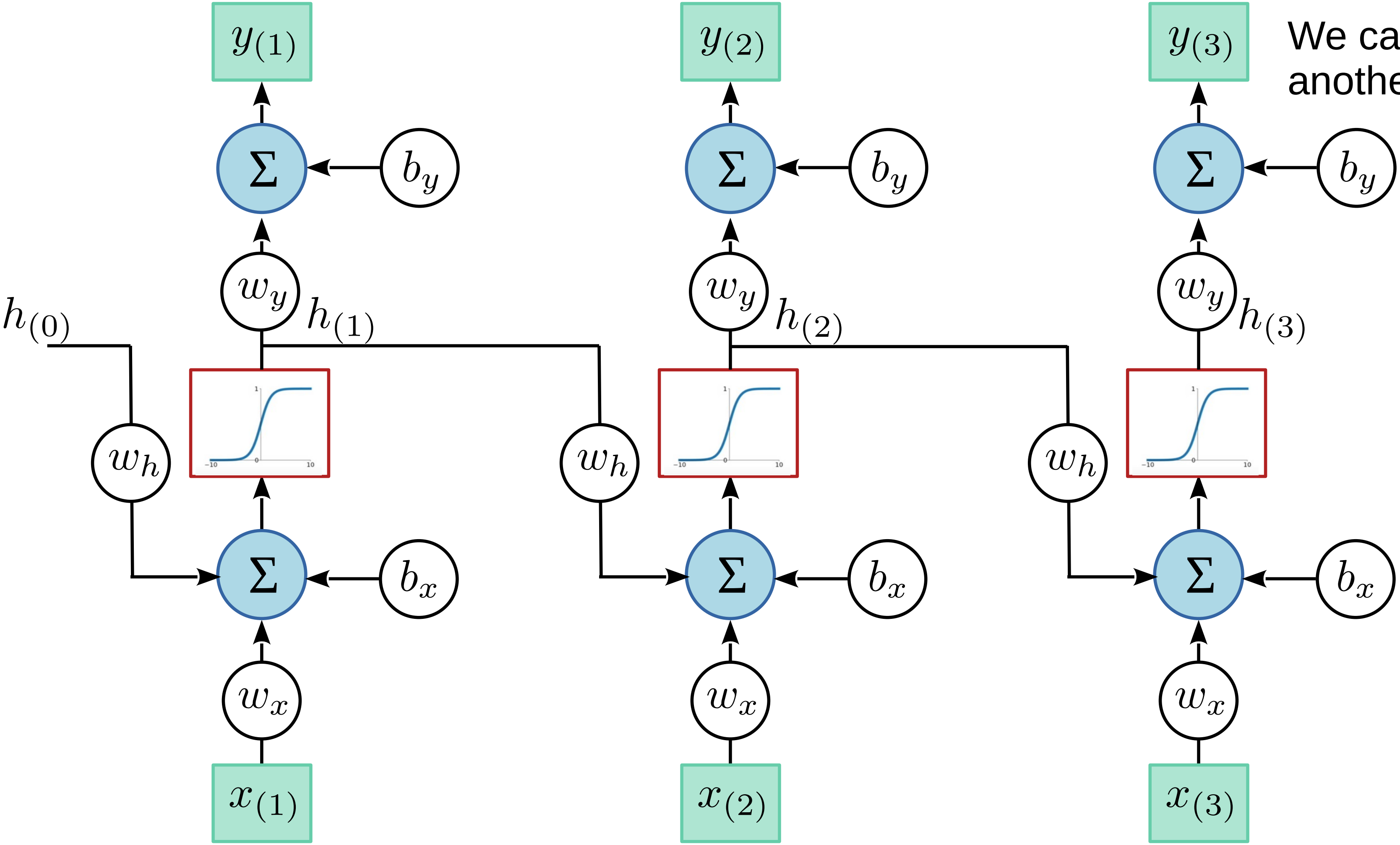
$$y_{(t)} = w_y \times h_{(t)} + b_y$$

not the same timepoint

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

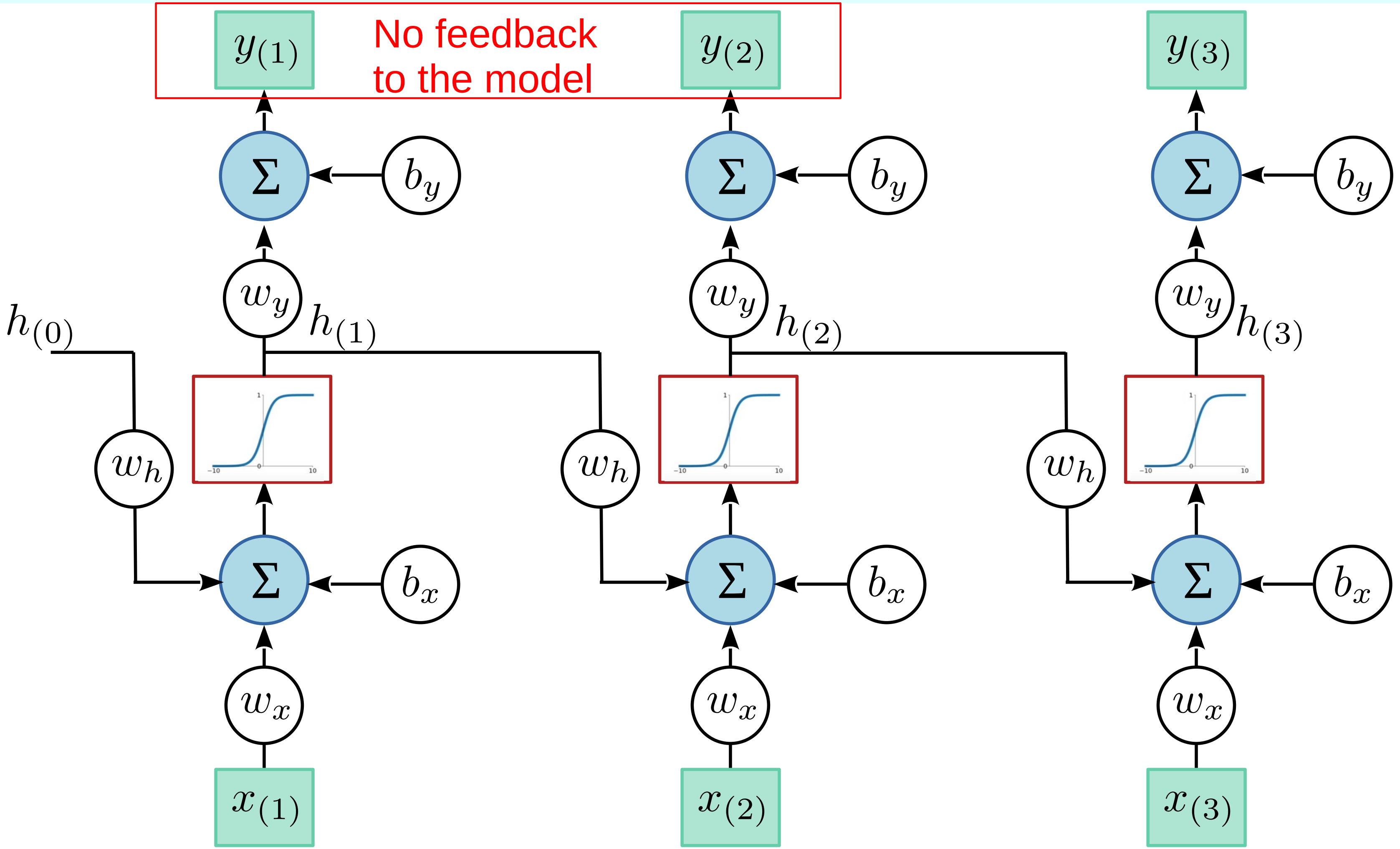
"memory"

RNN: 1 cell - unfolding



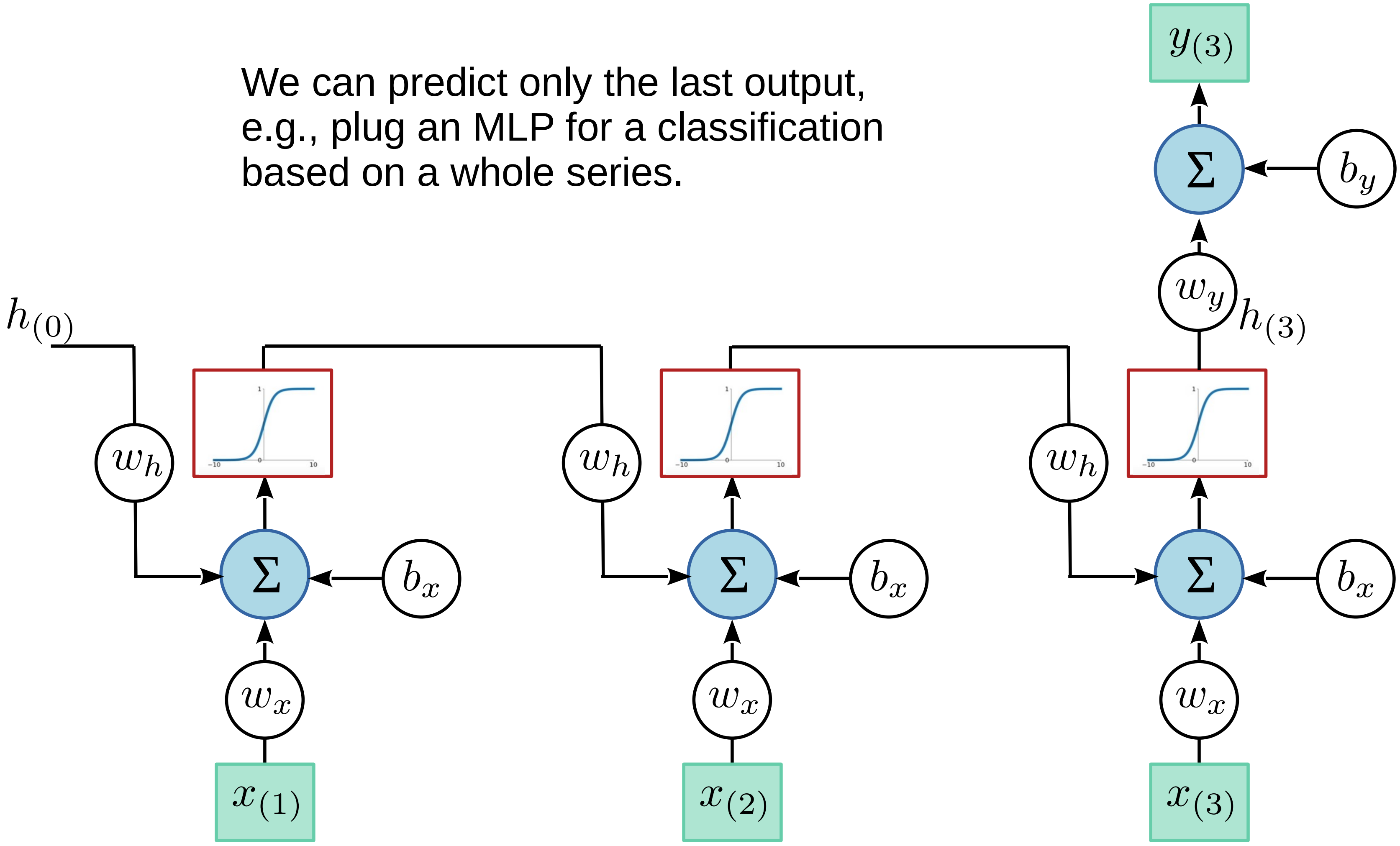
We can output another series, e.g, transcription, translation, conversion.

RNN: 1 cell - unfolding

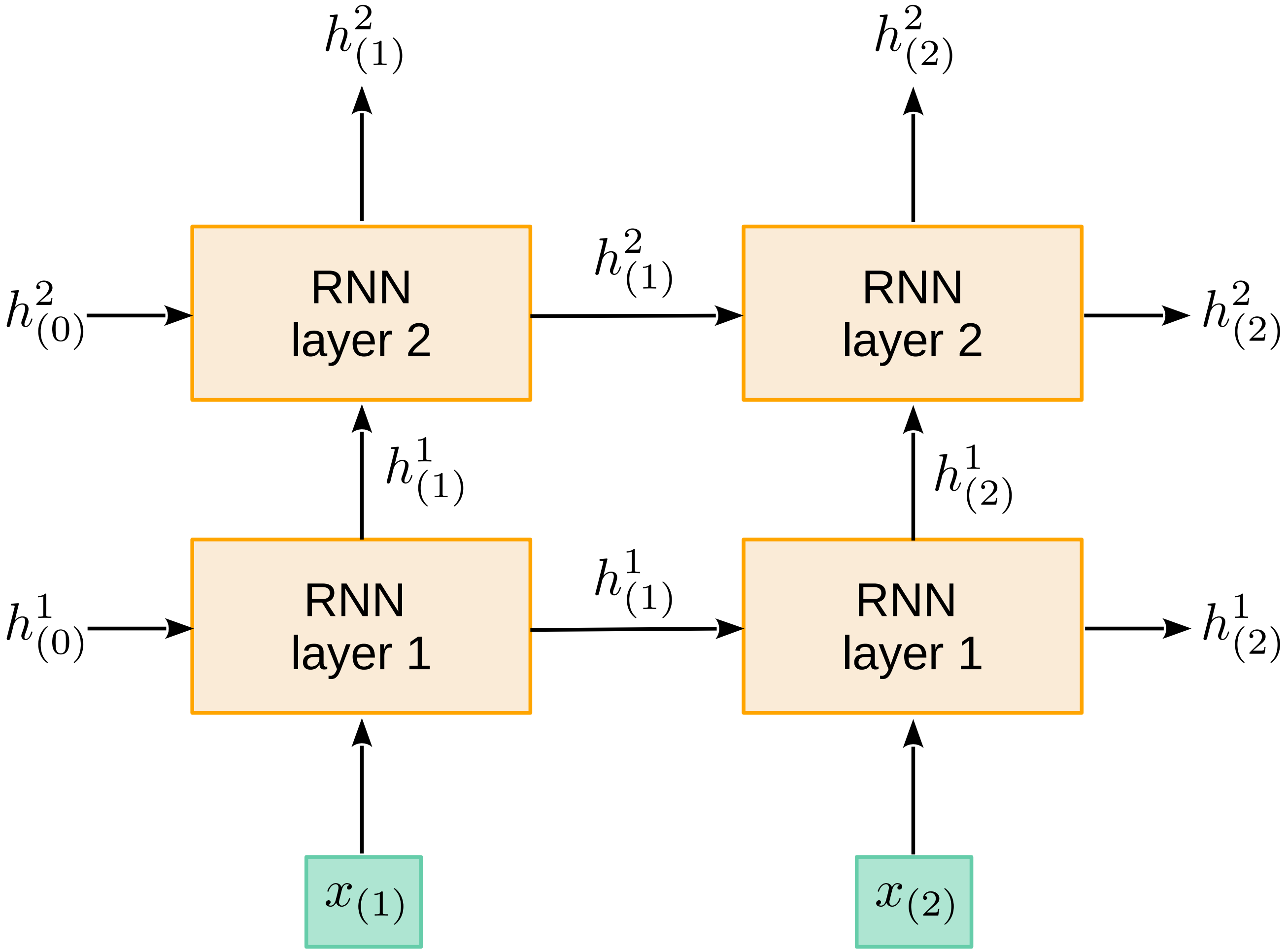


RNN: 1 cell - unfolding

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

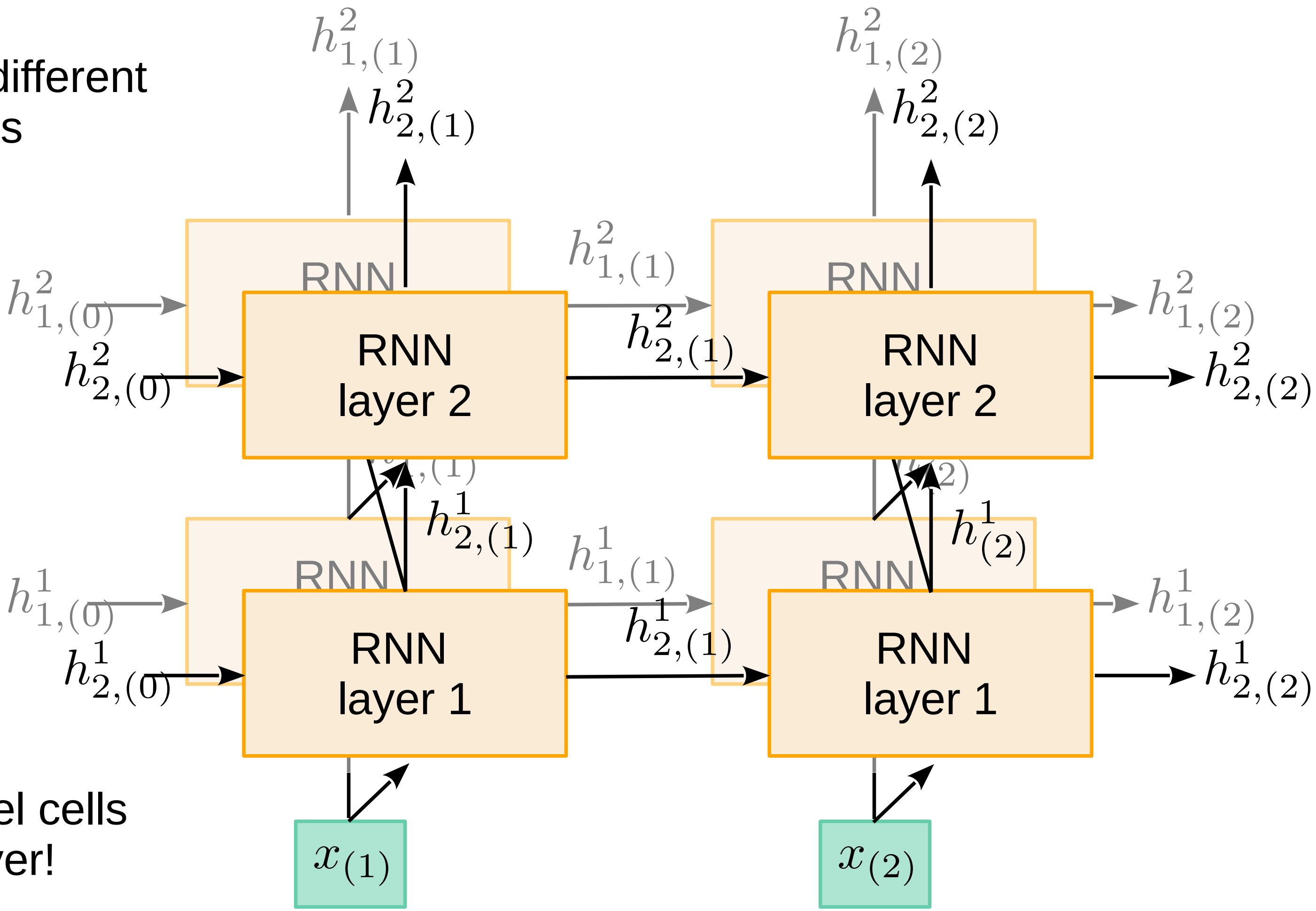


Stacked RNNs



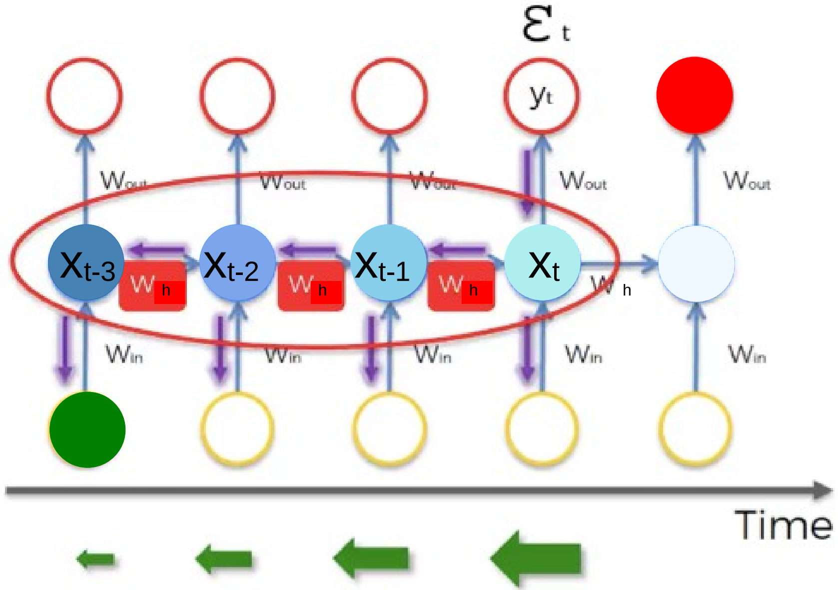
Several RNNs may learn different patterns in parallel

Same idea as different kernels in CNNs



No connexions between parallel cells of the same layer!

Vanishing and exploding gradients: the network forgets



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

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

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

W_h ~ small	⇨	Vanishing
W_h ~ large	⇨	Exploding

Adapted from:
SuperDataScience

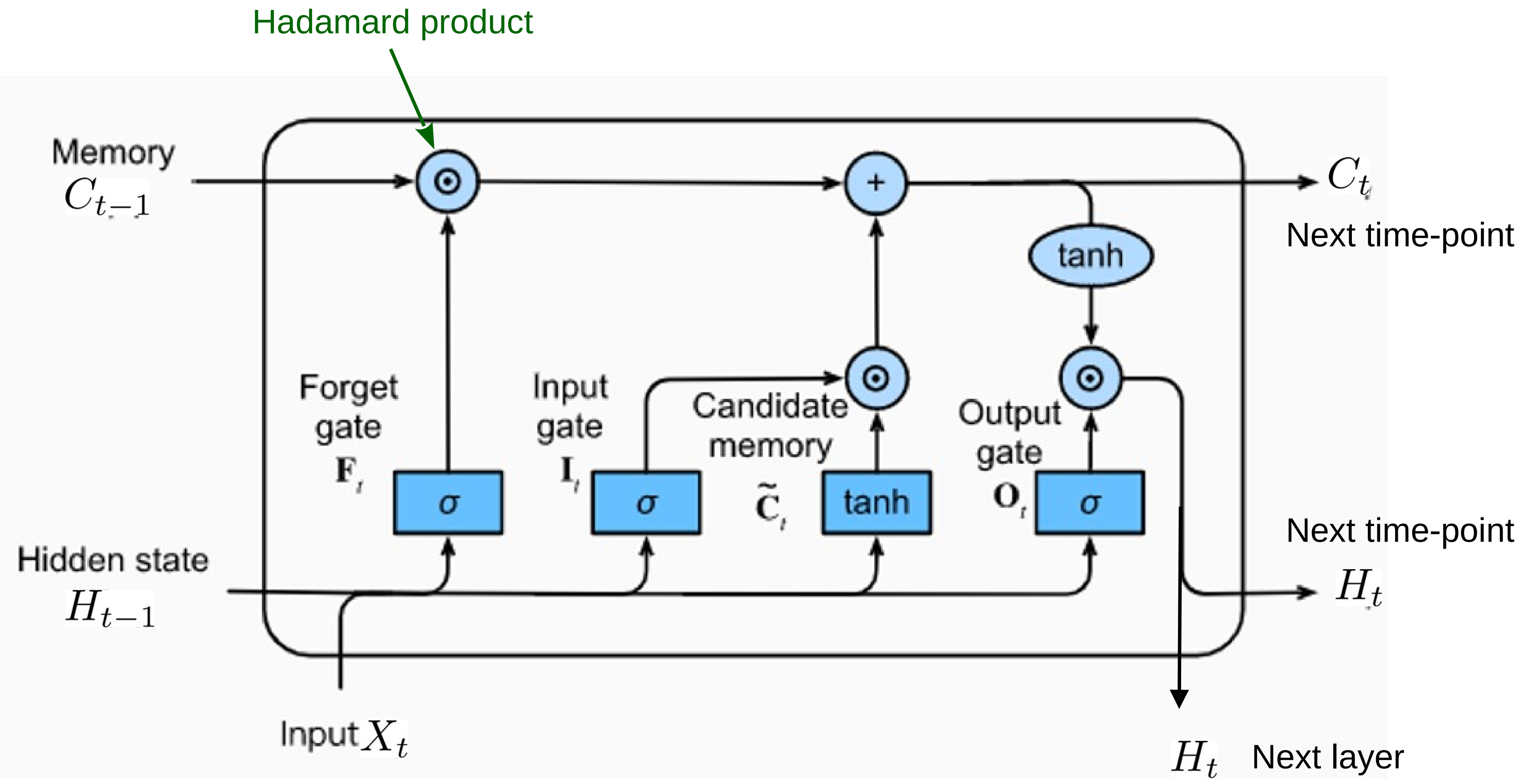
E.g.: activation function = ReLU

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

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

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

Solution: Long Short-Term Memory (LSTM)

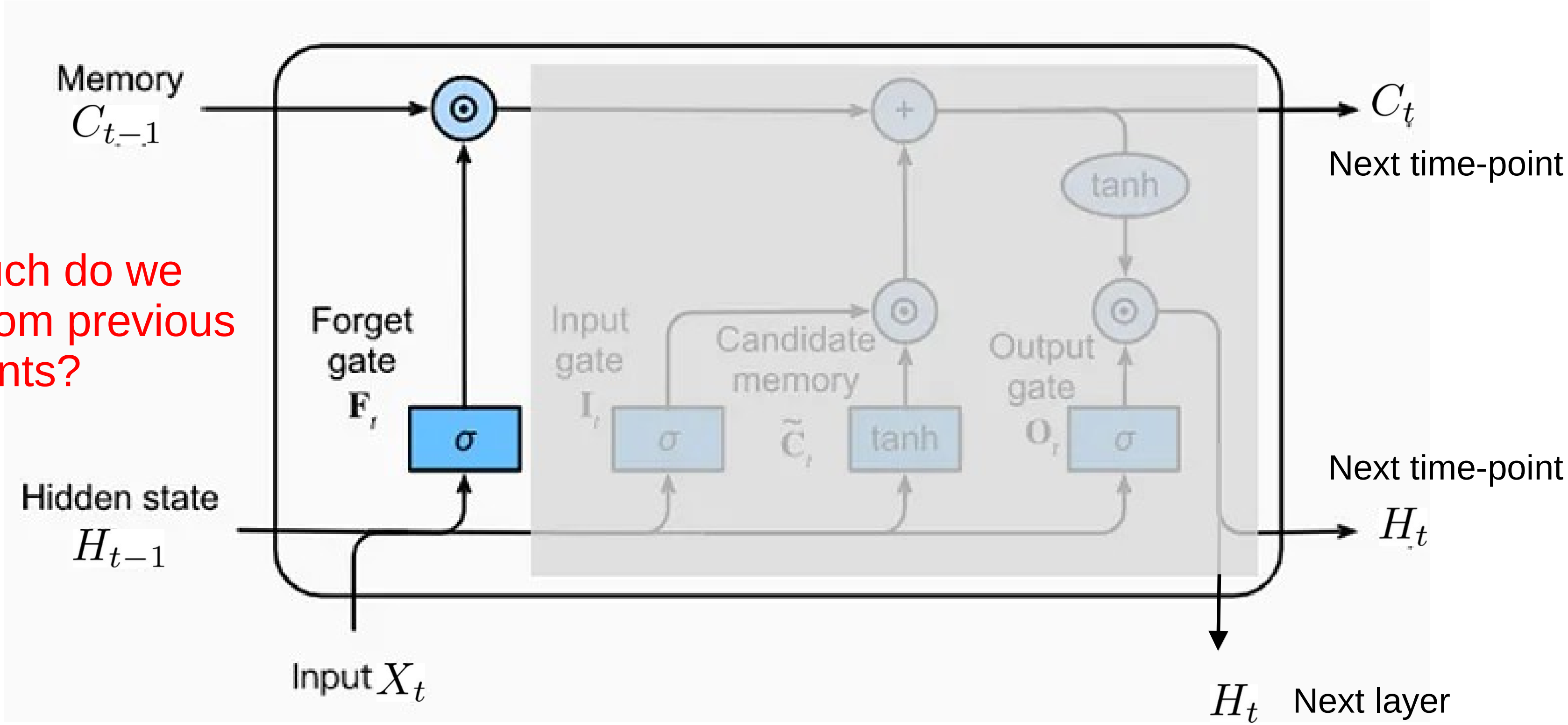


Hochreiter and Schmidhuber (1997) Long short-term memory. *Neur Comput*, 9(8):1735-1780

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

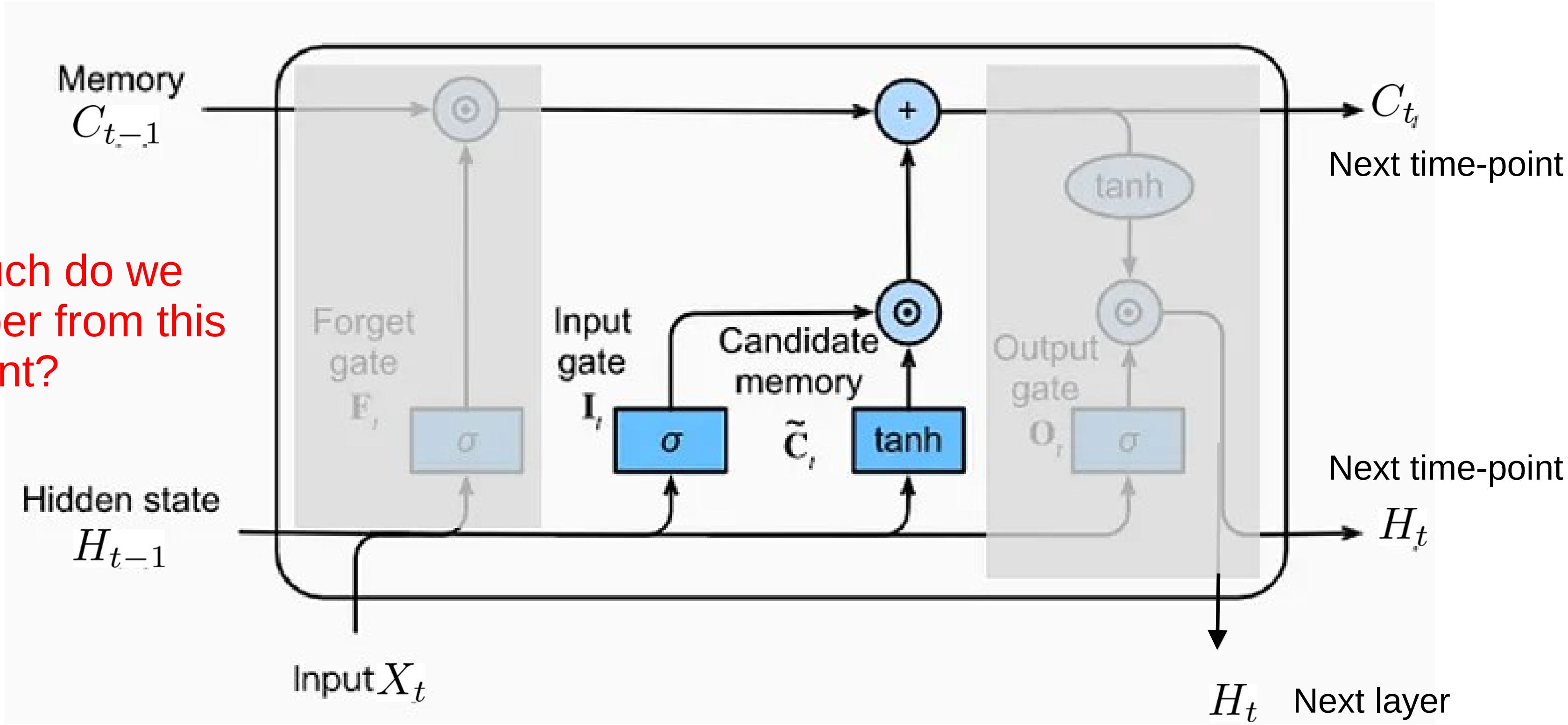
Solution: Long Short-Term Memory (LSTM)

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



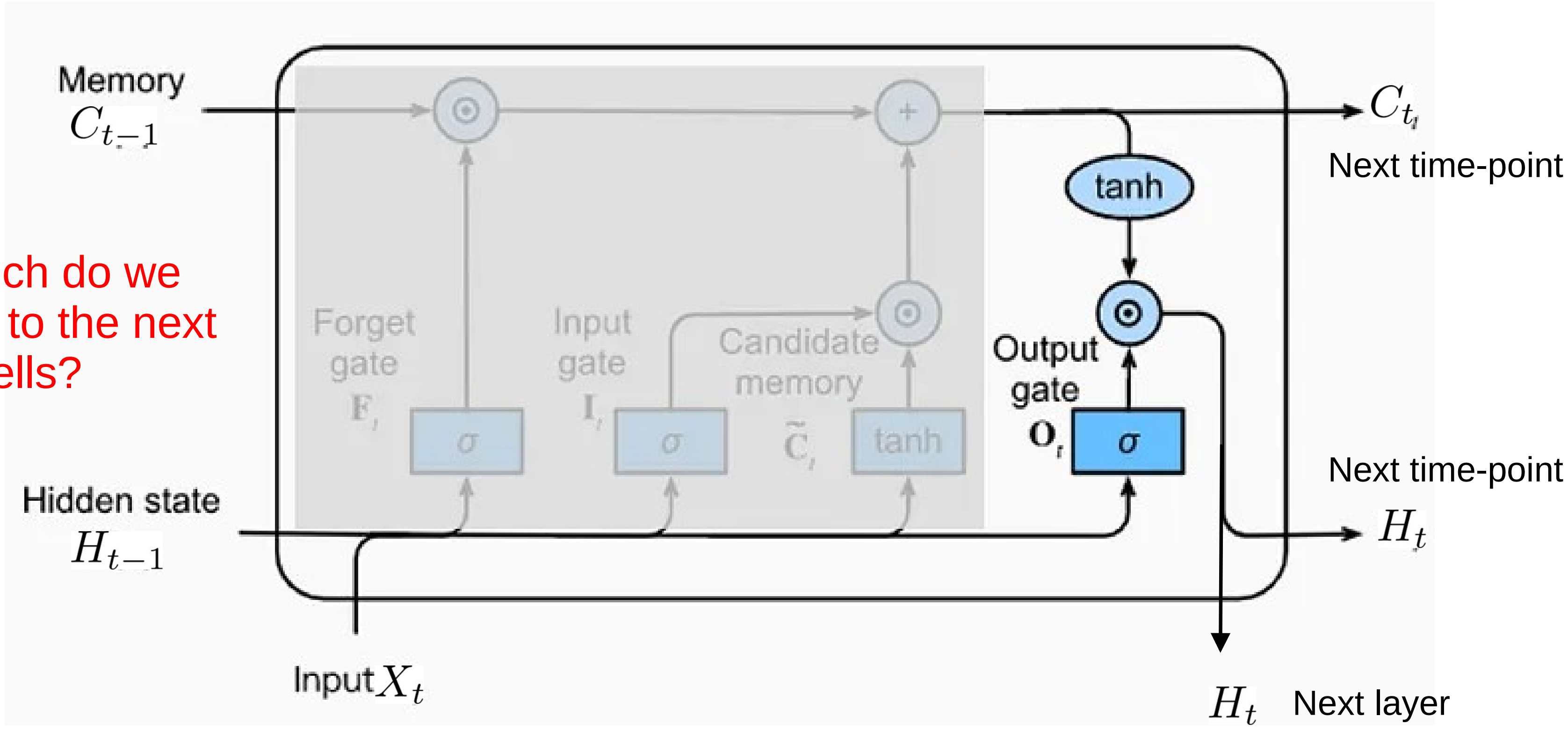
Solution: Long Short-Term Memory (LSTM)

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



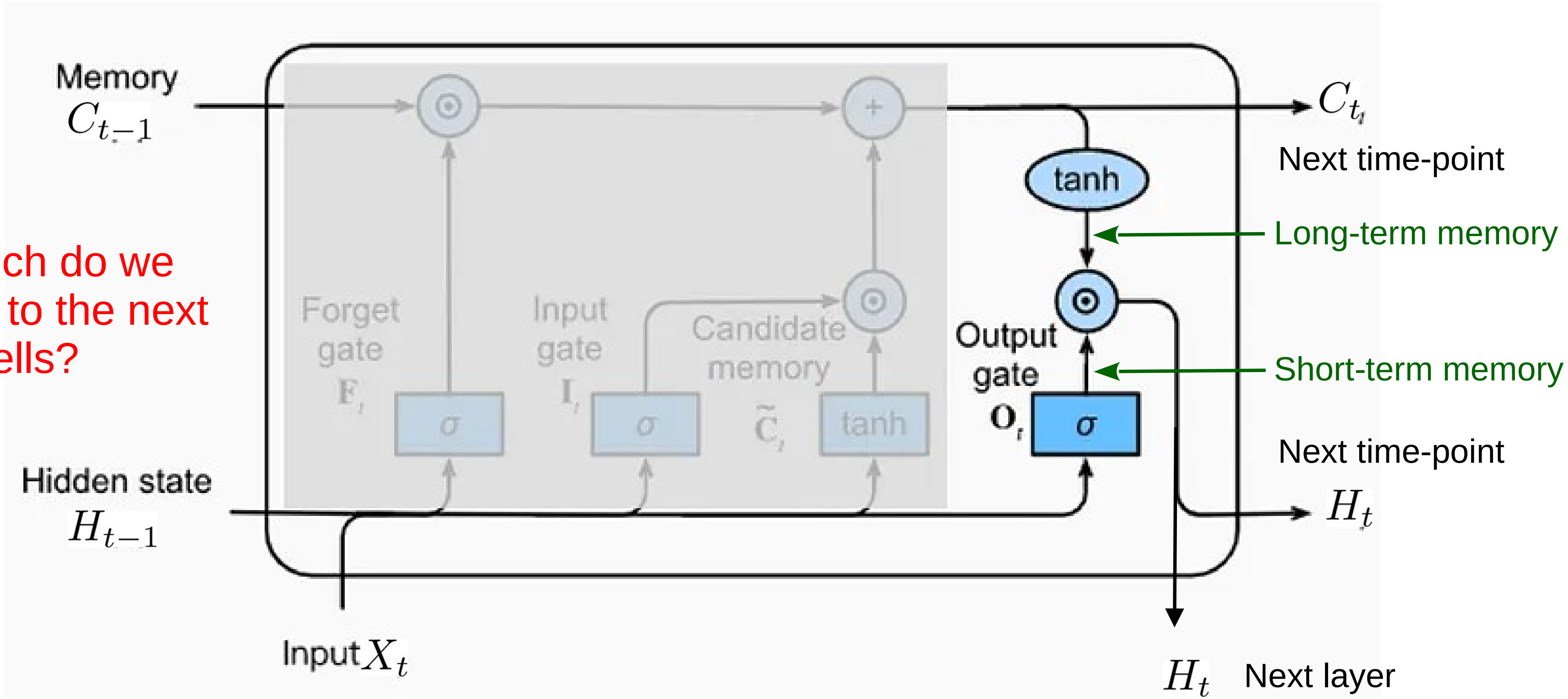
Solution: Long Short-Term Memory (LSTM)

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

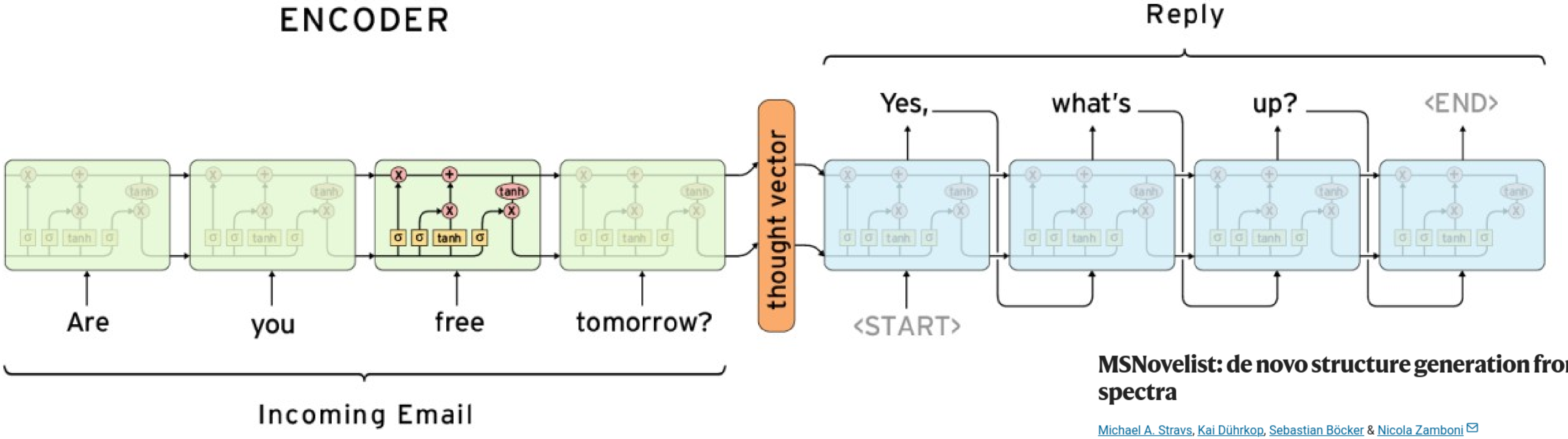


Solution: Long Short-Term Memory (LSTM)

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



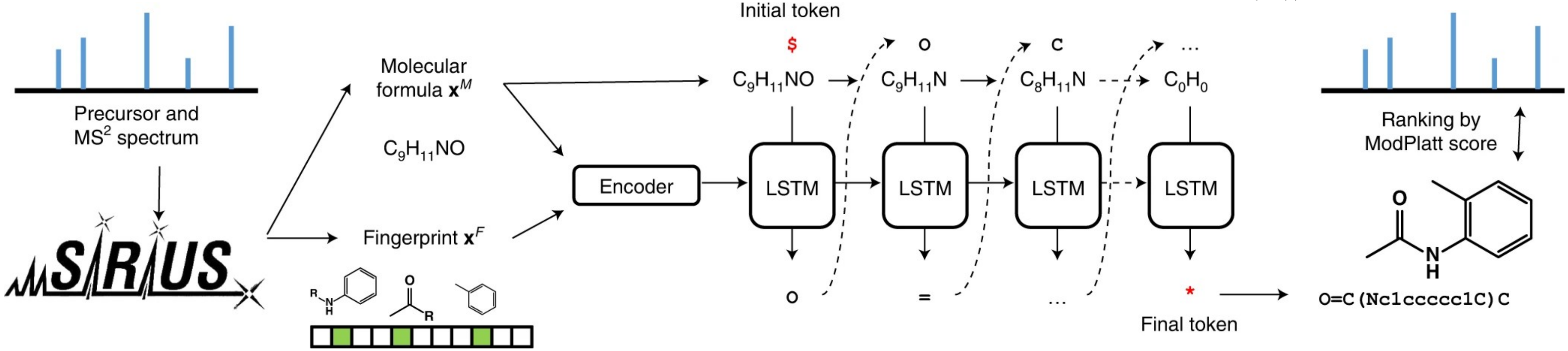
LSTMs for Encoder-Decoder



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



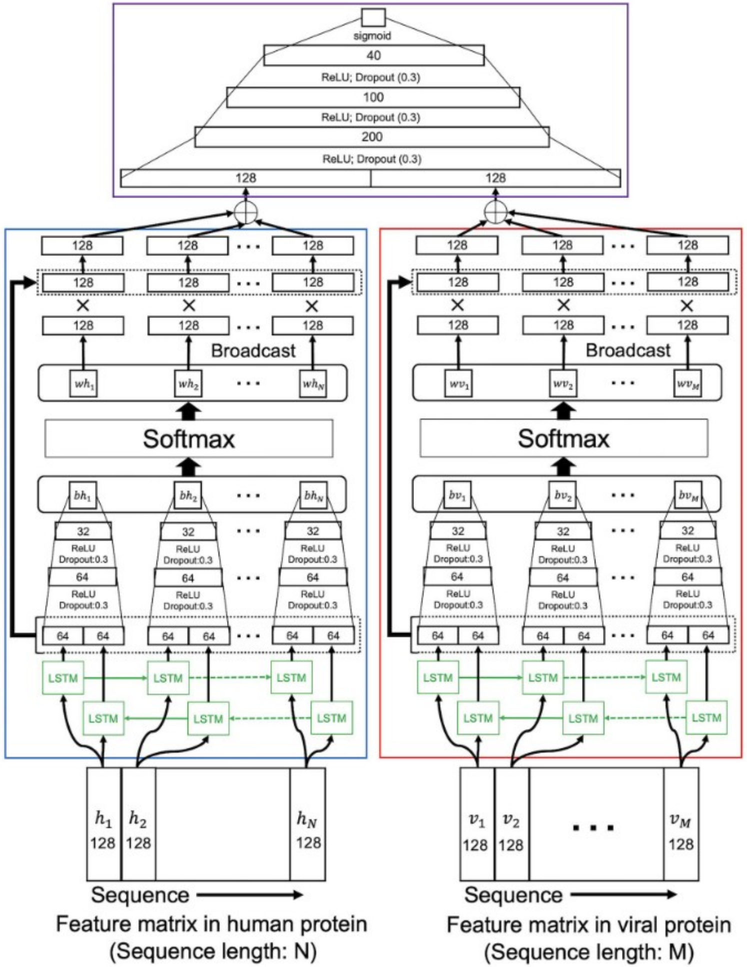
Example in bioinformatics

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

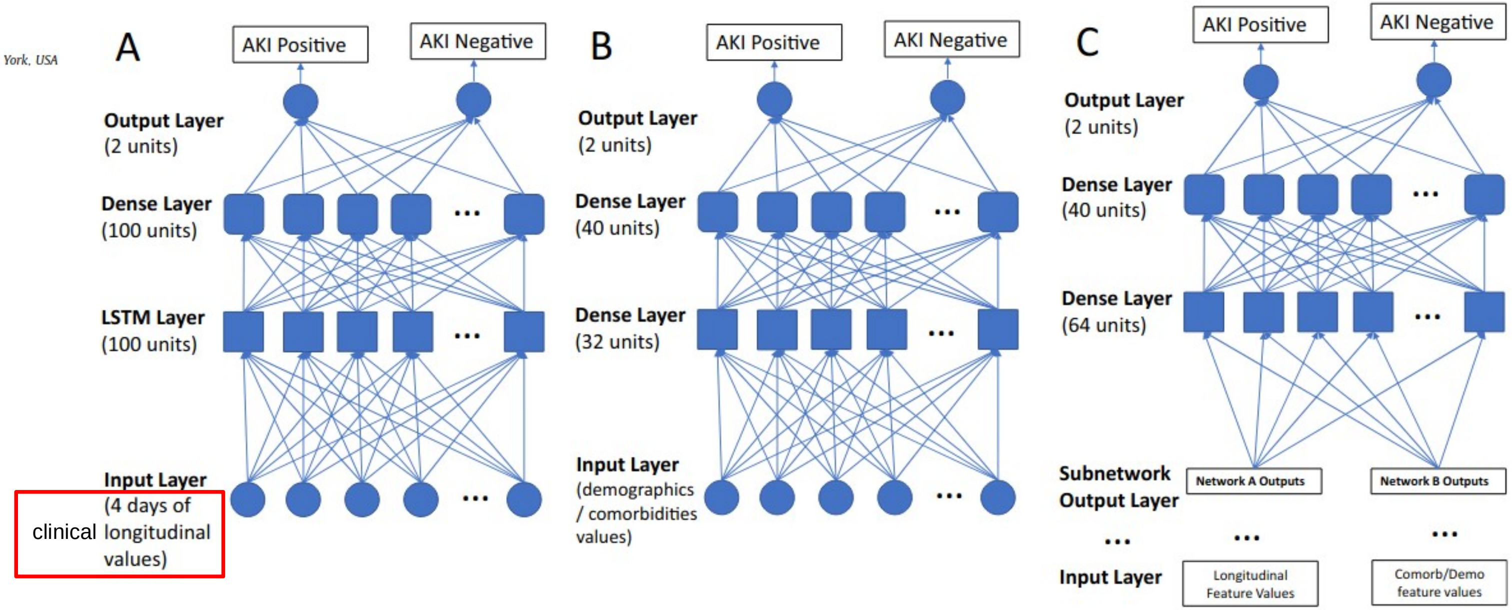
ELSEVIER

INTERNATIONAL SOCIETY FOR INFECTIOUS DISEASES

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



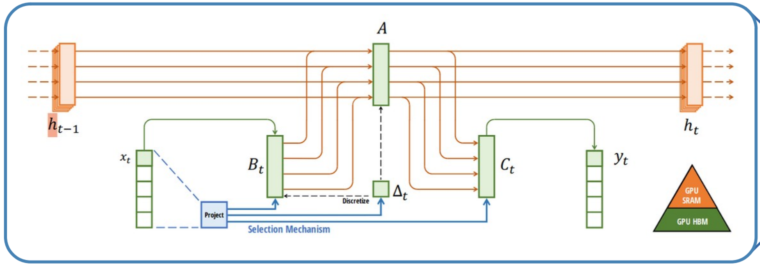
11

RNNs are back

Rise of the Mamba

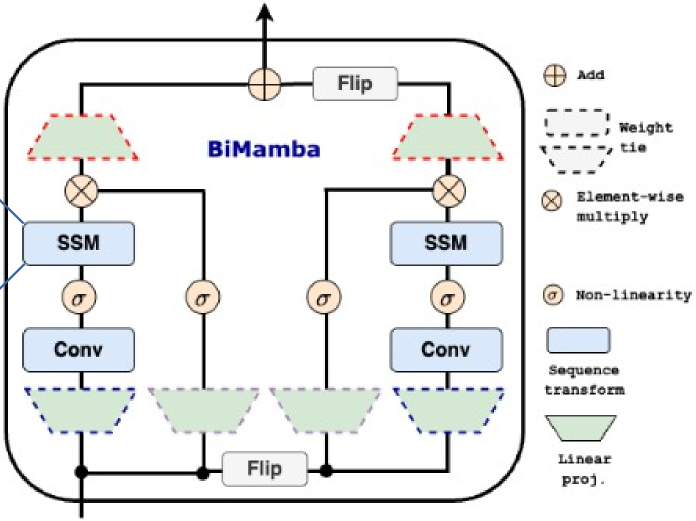
Selective State Space Model

“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

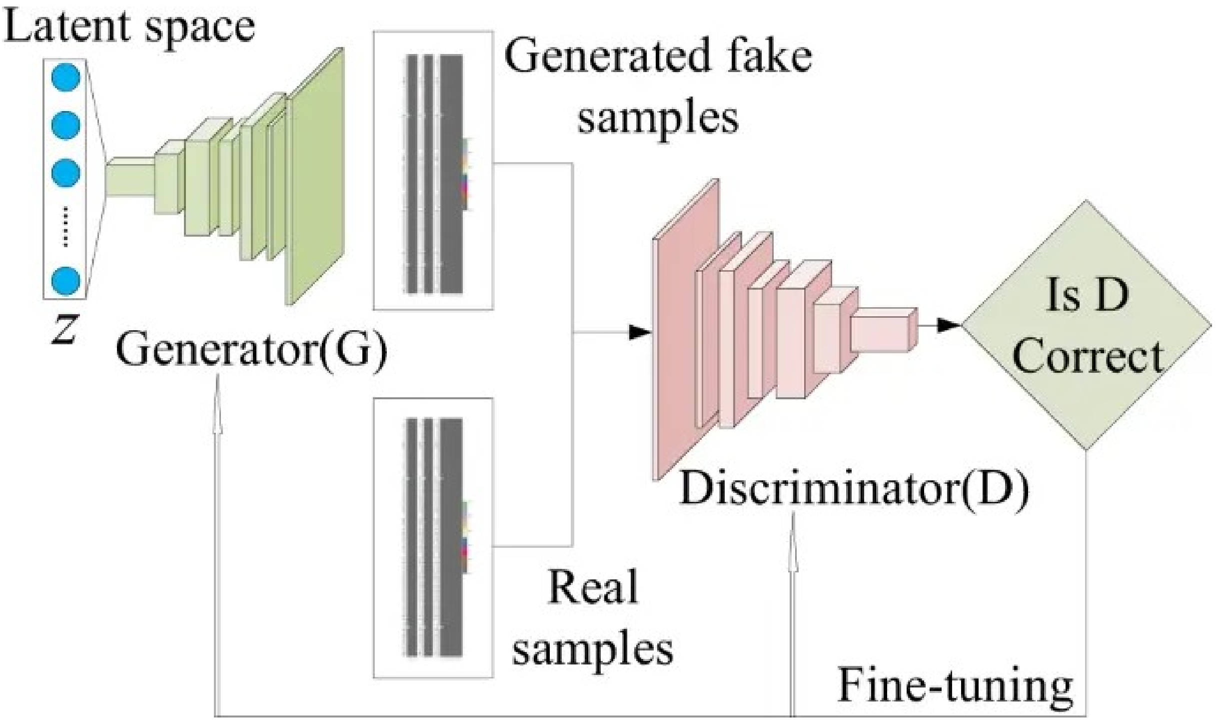
Gu and Dao (2023) *arXiv*:2312.00752
Schiff *et al* (2024) *arXiv*:2403.03234

https://github.com/state-spaces/mamba
https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mamba-and-state

12

Generative Adversarial Networks (GANs)

Learning by trying to trick itself: Generative Adversarial Network (GAN)



The Generator gets ever better at generating good fakes
The Discriminator gets ever better at detecting them

GAN for synthetic data

Neurocomputing
Volume 321, 10 December 2018, Pages 321-331

GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification

Maayan Frid-Adar [a], Idit Diamant [a], Eyal Klang [b], Michal Amitai [b], Jacob Goldberger [c], Hayit Greenspan [a]

Computer Methods and Programs in Biomedicine
Volume 195, October 2020, 105568

A GAN-based image synthesis method for skin lesion classification

Zhiwei Qin [a], Zhao Liu [b], Ping Zhu [a], Yongbo Xue [a]

Brain tumor image generation using an aggregation of GAN models with style transfer

Debadyuti Mukherkjee, Pritam Saha, Dmitry Kaplun, Aleksandr Sinitca & Ram Sarkar

Scientific Reports **12**, Article number: 9141 (2022) | Cite this article

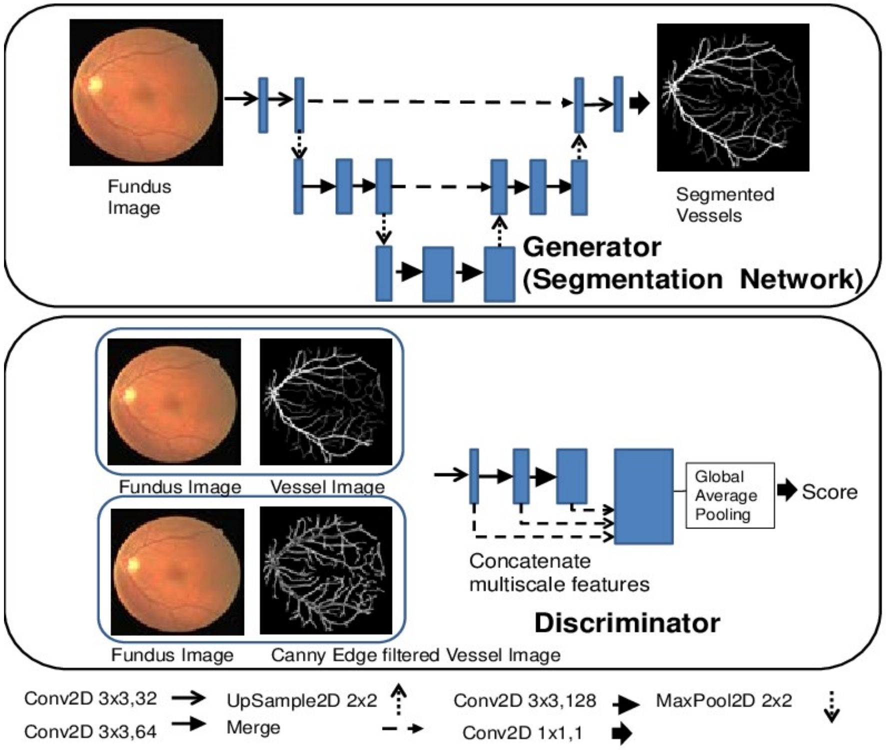
Home > Proceedings of International Conference on Artificial Intelligence and Applications >
Conference paper

MR Image Synthesis Using Generative Adversarial Networks for Parkinson's Disease Classification

Conference paper | First Online: 02 July 2020
pp 317–327 | Cite this conference paper

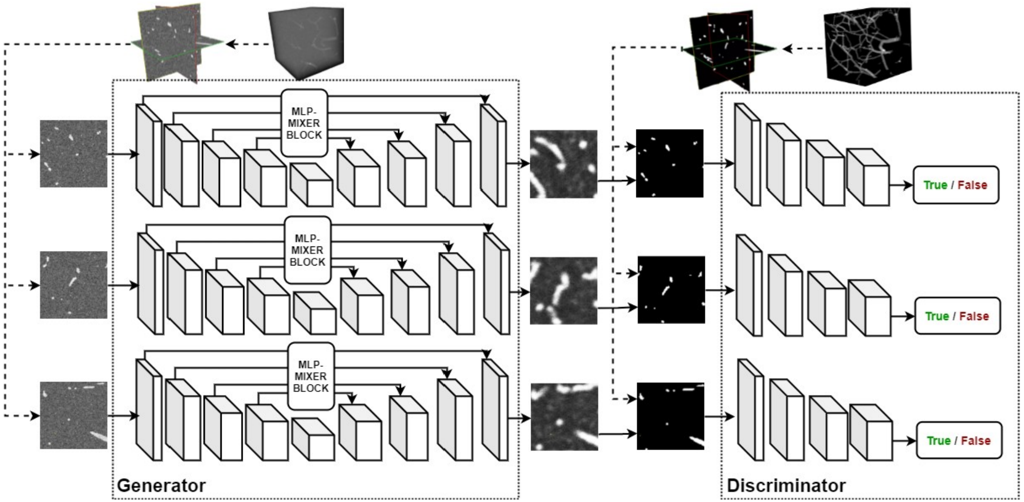
Sukhpal Kaur, Himanshu Aggarwal & Rinkle Rani

GAN for segmentation



Tjio, G., Li, S., Xu, X., Ting, D.S.W., Liu, Y., Goh, R.S.M. (2019).
Multi-discriminator Generative Adversarial Networks for Improved Thin Retinal Vessel Segmentation.
In: Fu, H., Garvin, M., MacGillivray, T., Xu, Y., Zheng, Y. (eds) Ophthalmic Medical Image Analysis. OMIA 2019.
Lecture Notes in Computer Science(), vol 11855. Springer, Cham. https://doi.org/10.1007/978-3-030-32956-3_18

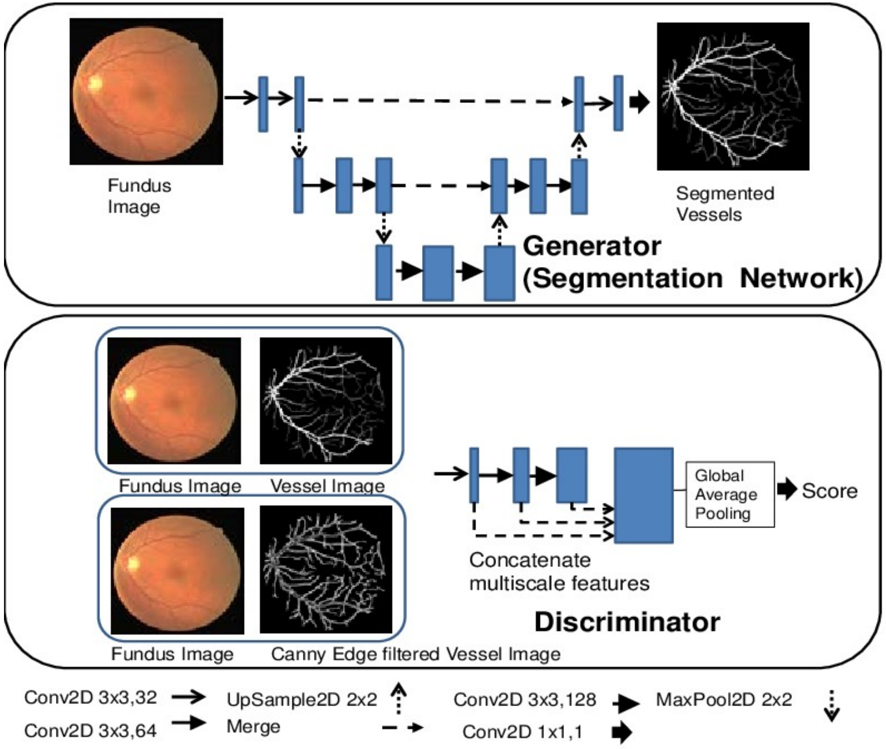
GAN can be built out of any DL architecture



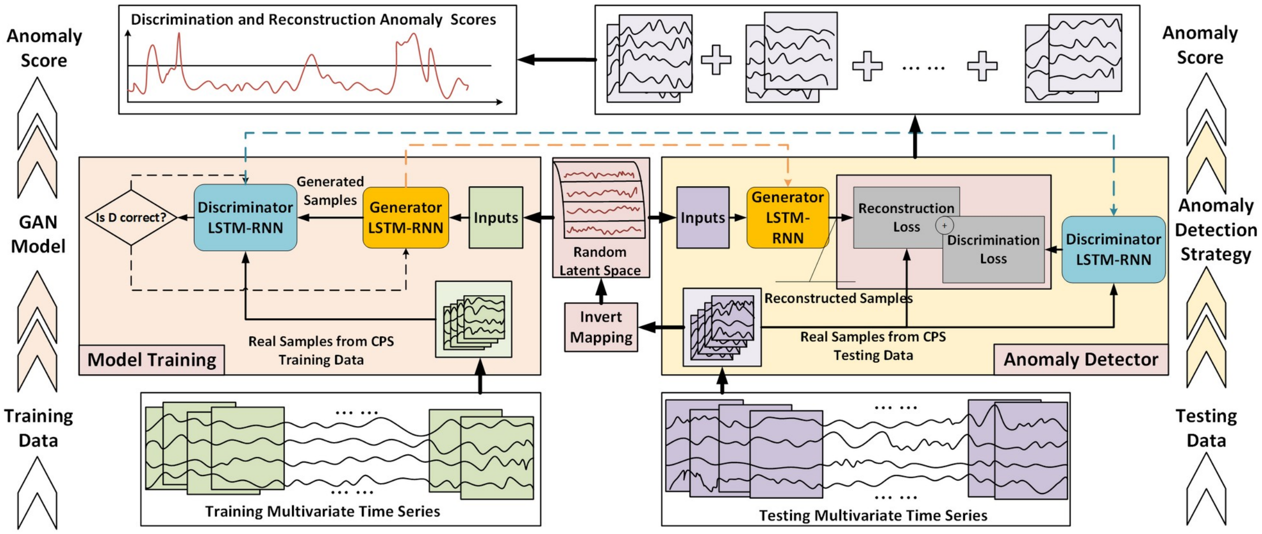
DOI: 10.1109/ICASSP49357.2023.10096997 • Corpus ID: 250626500

MLP-GAN for Brain Vessel Image Segmentation

B. Xie, Hao Tang, +2 authors Yan Yan • Published in IEEE International Conference... 17 July 2022 • Computer Science, Medicine



GAN can be built out of any DL architecture

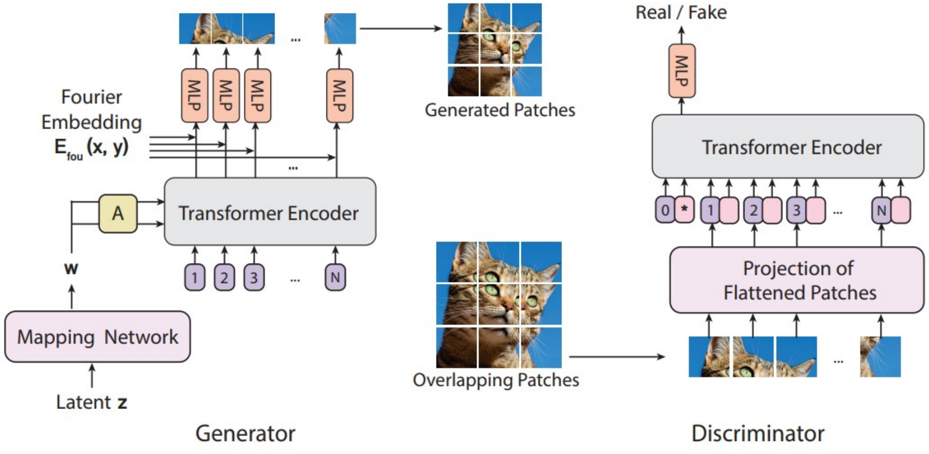


MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks

Dan Li[1], Dacheng Chen[1], Lei Shi[1], Baihong Jin[2], Jonathan Goh[3], and See-Kiong Ng[1]

ViTGAN: Training GANs with Vision Transformers

Kwonjoon Lee[1] Huiwen Chang[2] Lu Jiang[2] Han Zhang[2]
Zhuowen Tu[1] Ce Liu[2]
[1]UC San Diego [2]Google Research
{kwl042,ztu}@ucsd.edu {huiwenchang,lujiang,zhanghan,celiu}@google.com



Acknowledgements

Philippe Froguel
Amélie Bonnefond

Smaïn Fettem

Mathilde Boissel
Lijiao Ning
Emma Henriques
ShuangShuang Geng

Anne-Sophie Ledoux
Vincent Massy

Amna Khamis
Stefan Gaget
Mehdi Derhourhi

Hélène de Gavre
Mélanie Hocquet

François Pattou

R package developers
(*VIM*, *MICE*, *DESeq2*, *ChAMP*)

Python package developers
(*Tensorflow/Keras*, *Numpy*, *Pandas*, *Scikit-learn*)

PreciNASH recherche hospitalo-universitaire

Région Hauts-de-France

PreciDIAB National Center for Precision Diabetic Medicine

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

cnrs