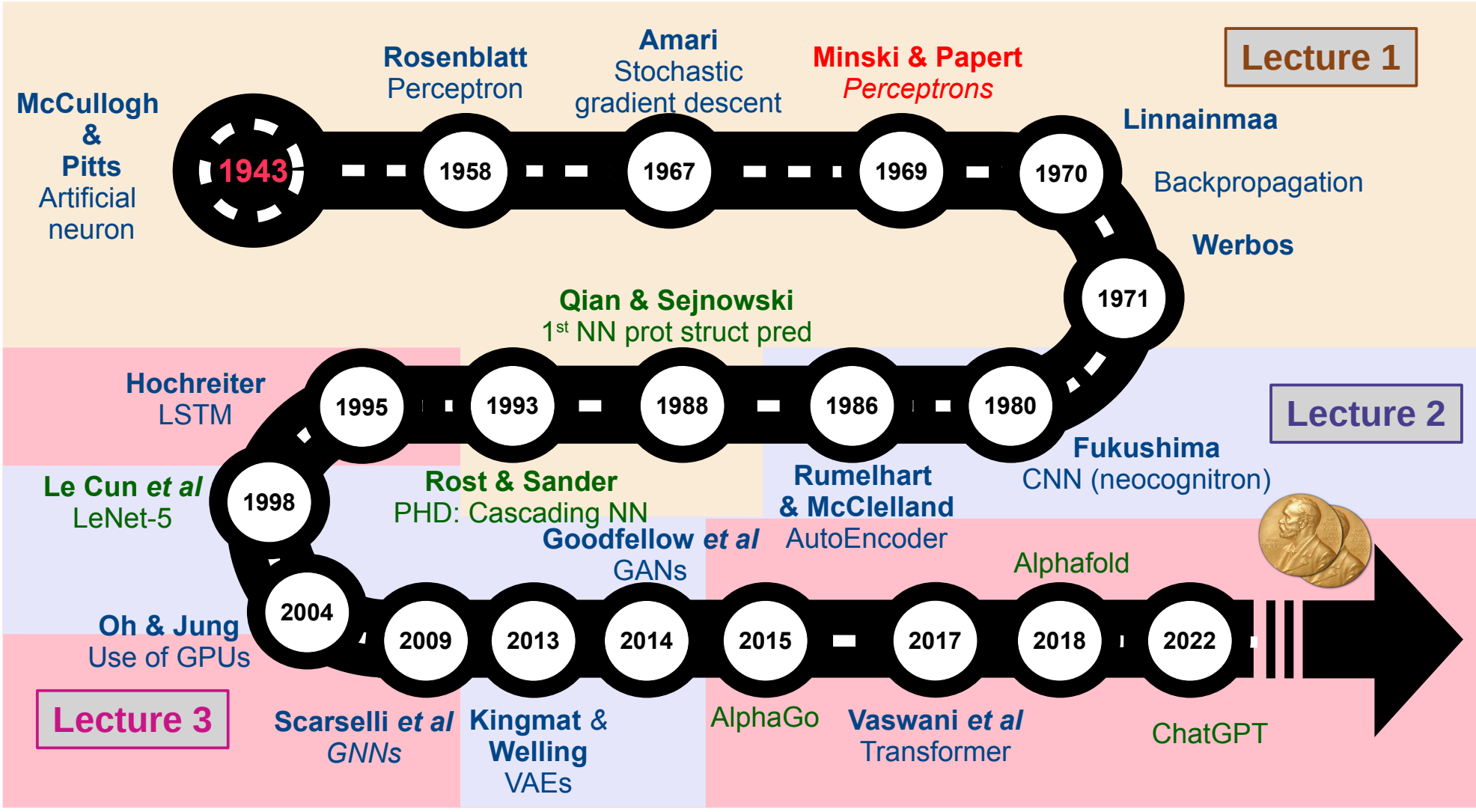
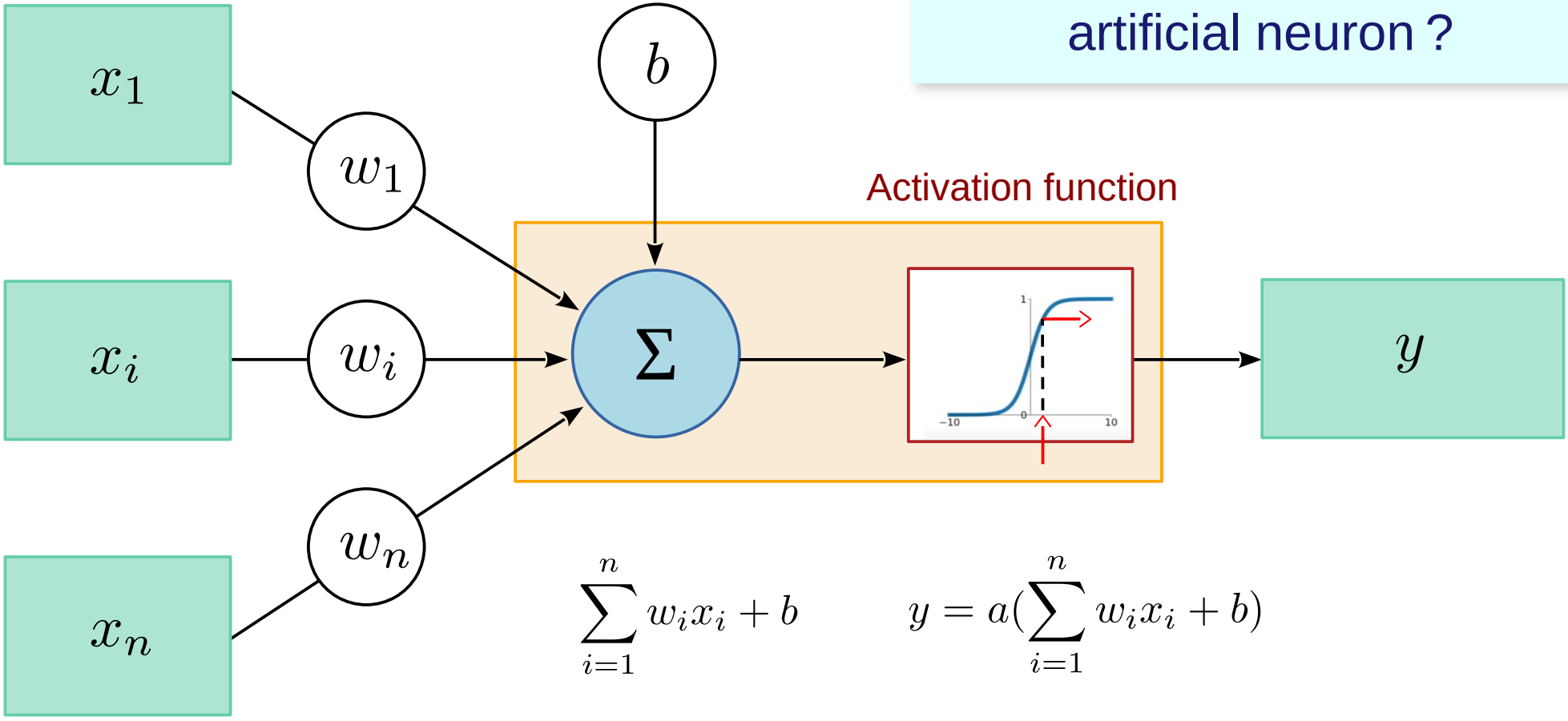
Beyond MLPs

Part 1/2: CNN and AutoEncoders

nicolas.gambardella@univ-lille.fr

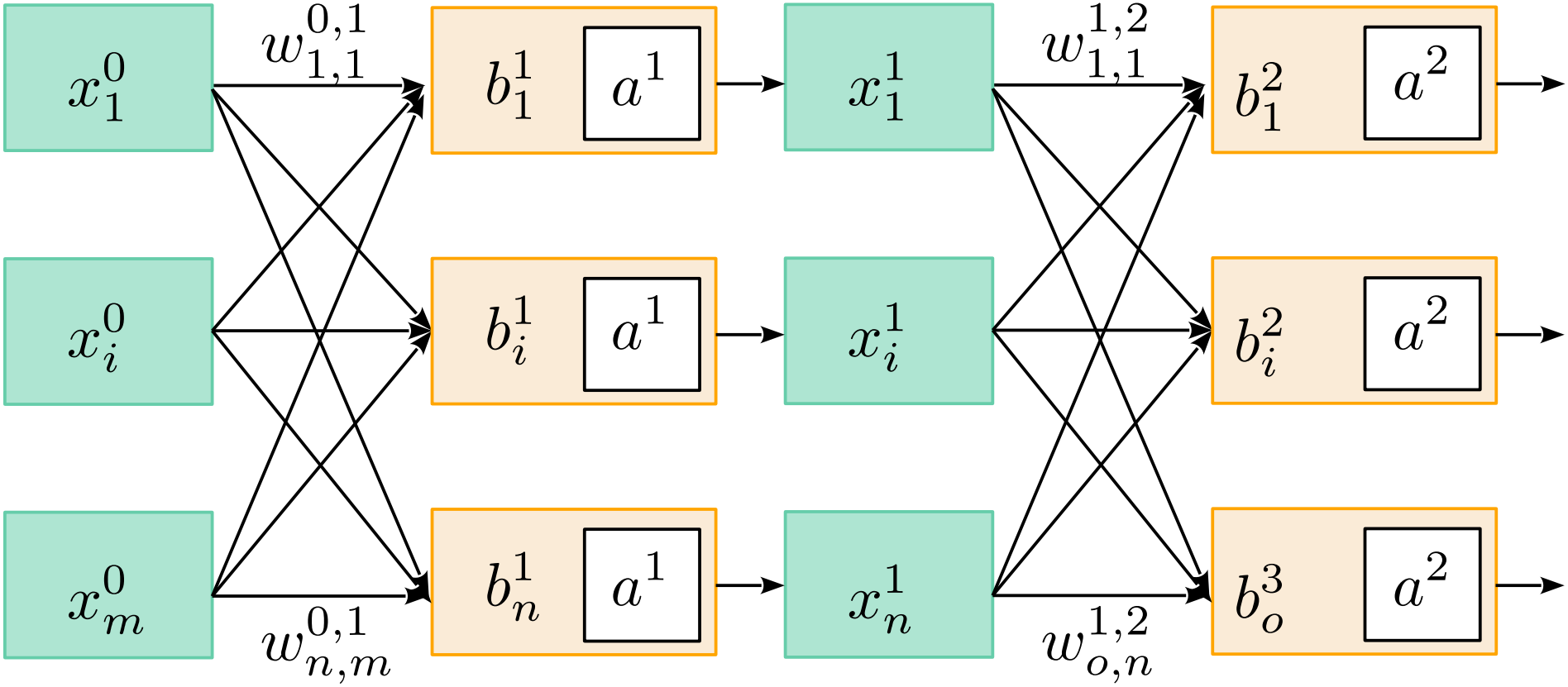

Lecture 1
McCullogh & Pitts
Artificial neuron
1943
Rosenblatt
Perceptron
1958
Amari
Stochastic gradient descent
1967
Minski & Papert
Perceptrons
1969
Linnainmaa
Backpropagation
1970
Werbos
1971
Qian & Sejnowski
1st NN prot struct pred
1980
1986
Rumelhart & McClelland
AutoEncoder
Fukushima
CNN (neocognitron)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

What is an artificial neuron ?



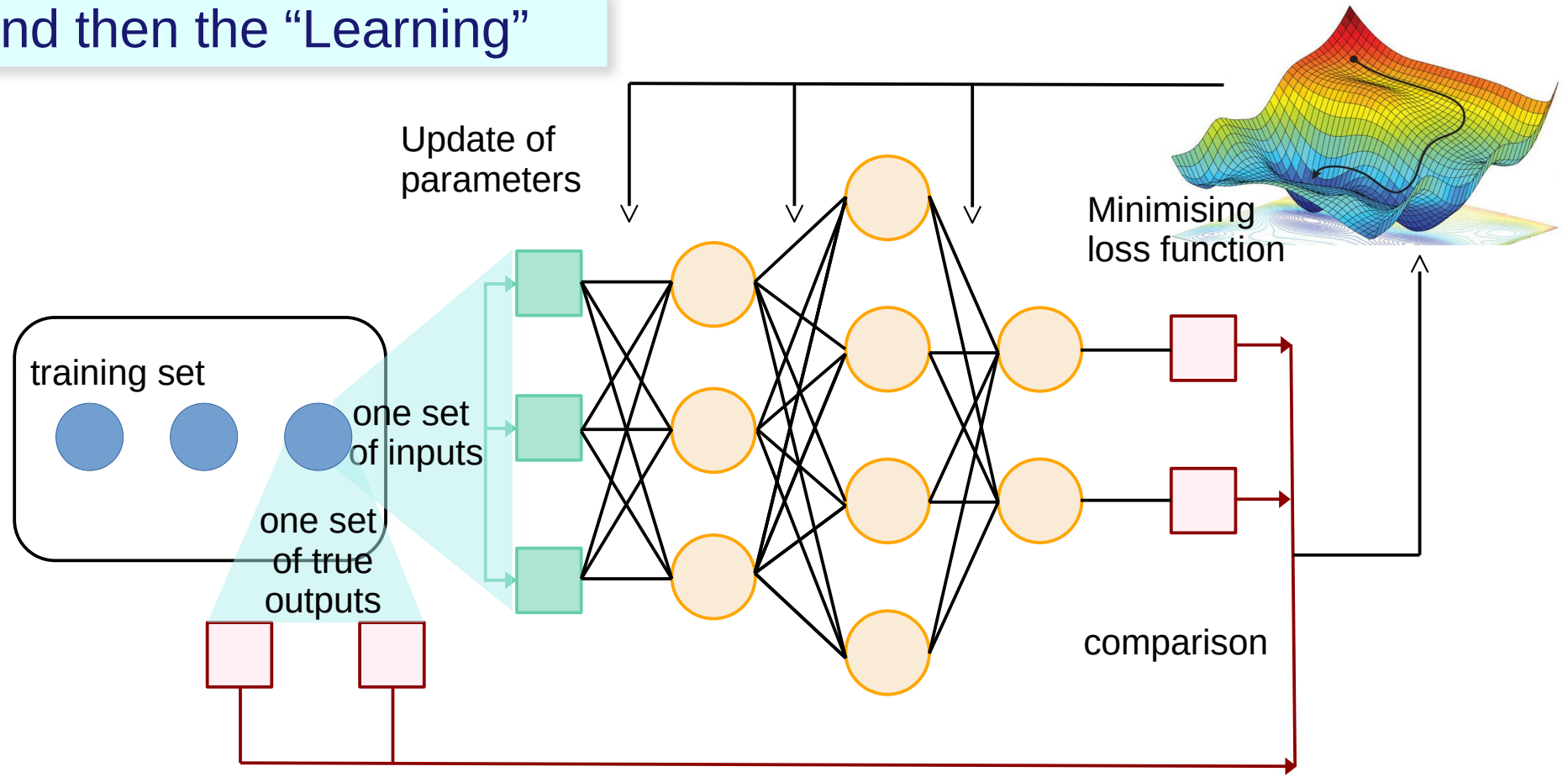
McCulloch and Pitts (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115-133
Rosenblatt (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65(6):386-408
Widrow and Hoff (1960) Adaptive Switching circuits.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

And then we add layers (the “Deep”)



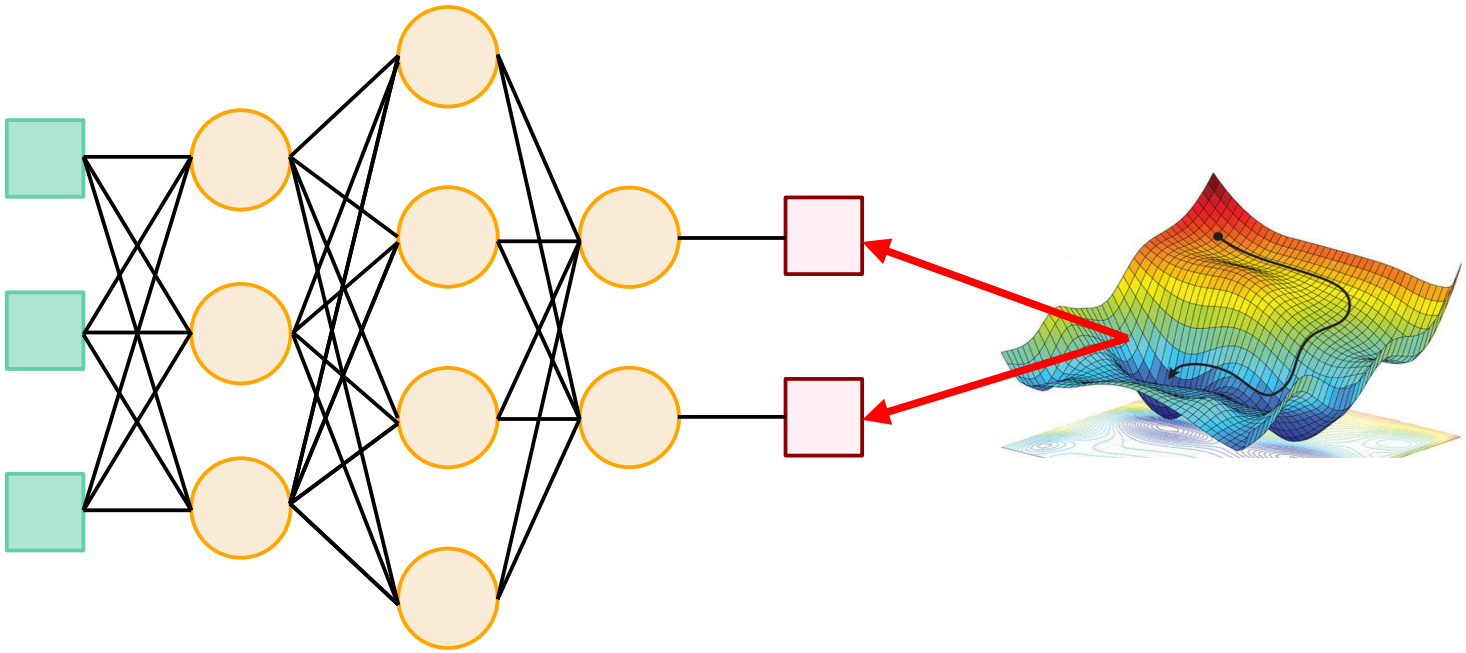
Rosenblatt (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65(6):386-408

And then the “Learning”

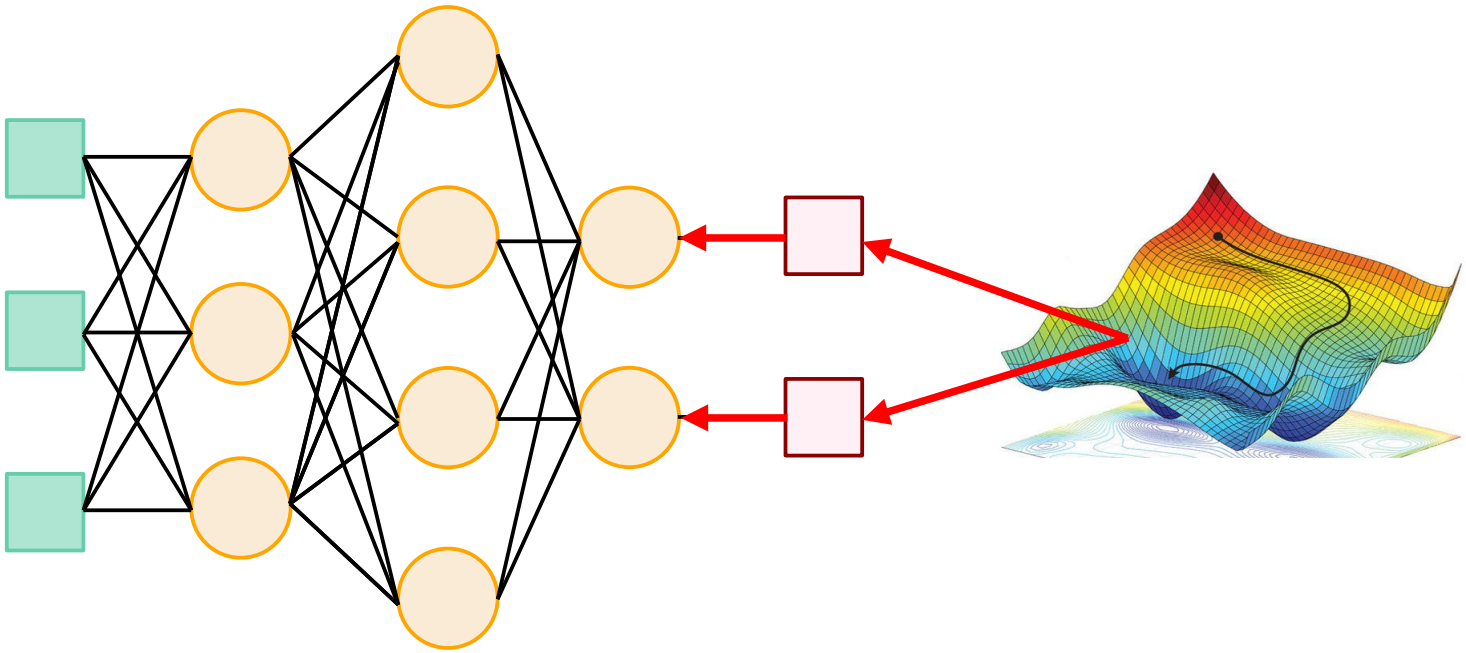


Widrow and Hoff (1960) Adaptive Switching circuits. *WESCON Convention record* part IV: 96-104; S Amari (1967). A theory of adaptive pattern classifier. *IEEE Transactions*. EC (16): 279–307; S Linnainmaa (1970-1976). The representation of the cumulative rounding error of an algorithm as a Taylor expansion 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
LeCun Y (1985) Une procédure d'apprentissage pour réseau à seuil asymétrique. *Proc Cognitiva* 85, 599-604.
Rumelhart, Hinton, and Williams (1986) Learning representations by back-propagating errors." *Nature* 323(6088): 533-536.

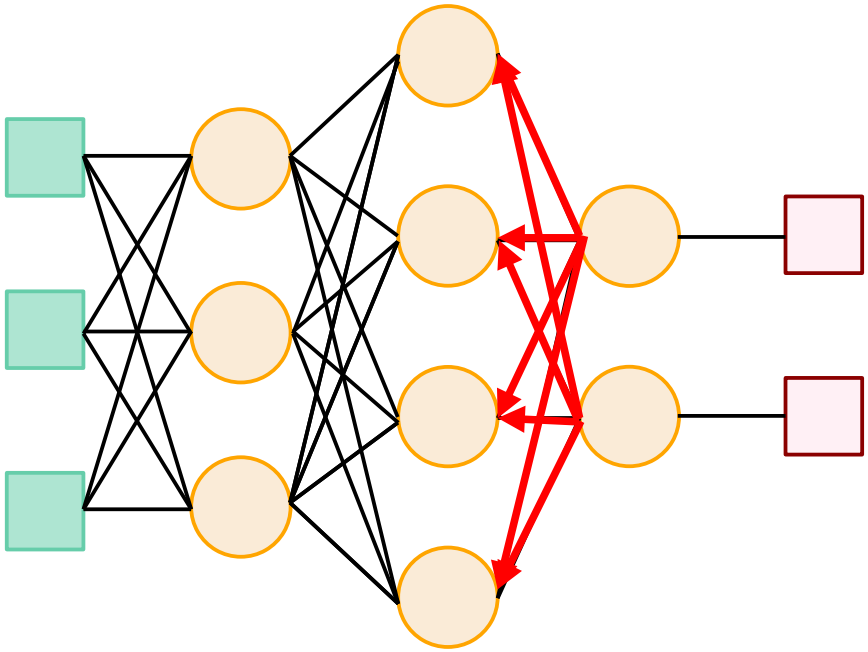
Error backpropagation one layer at a time

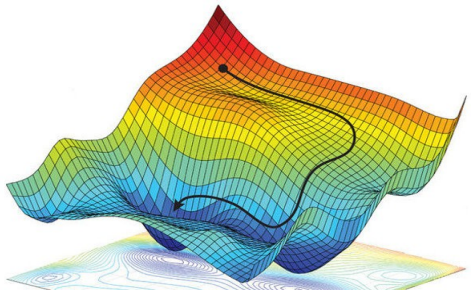


Error backpropagation one layer at a time

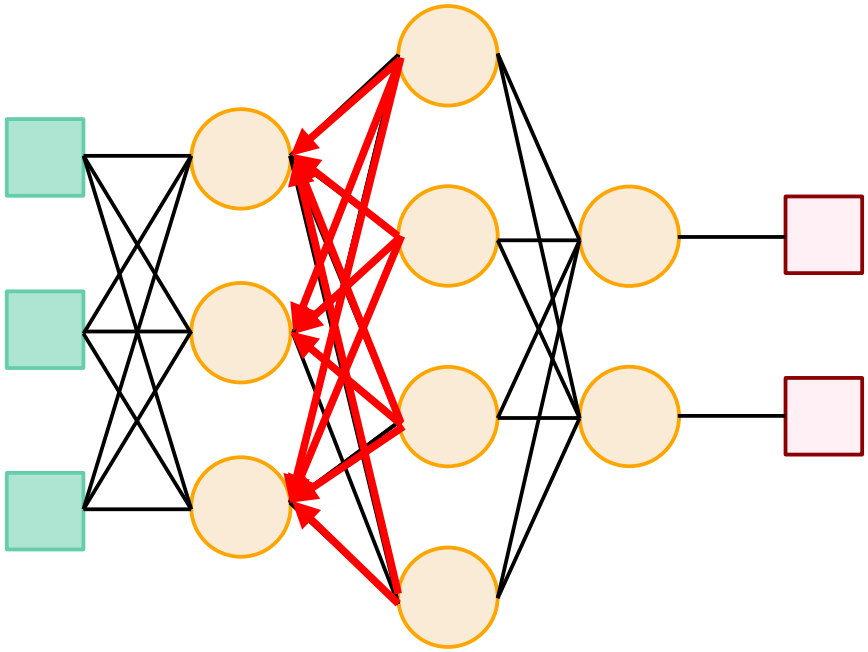


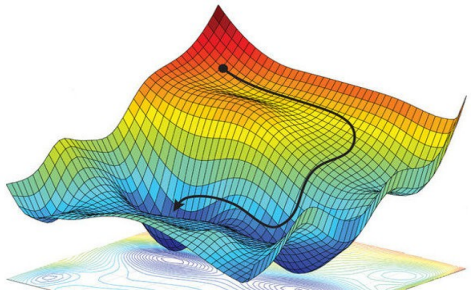
Error backpropagation one layer at a time



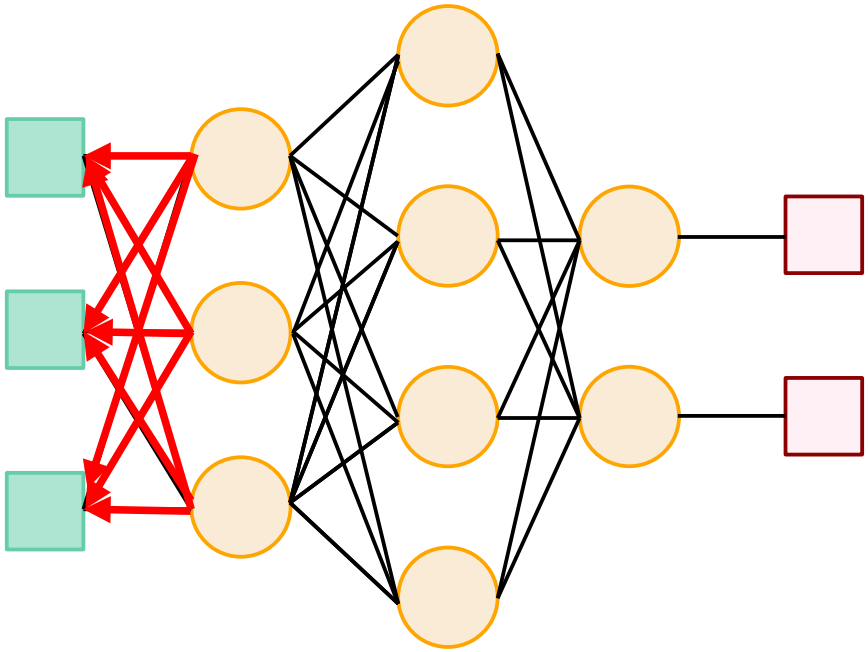


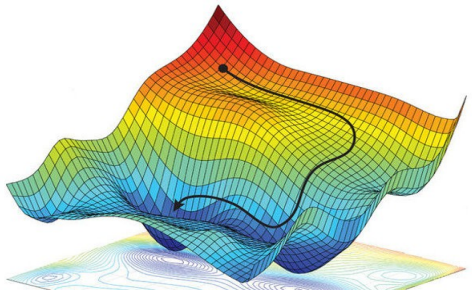
Error backpropagation one layer at a time





Error backpropagation one layer at a time





Output nodes (excl. EncoderDecoders)

(Linear) regression: Identity. The ouput values are linear combinations of the penultimate layer (plus bias)

$$\hat{y}_i = \sum_j^N w_{ij} \cdot x_j + b_i$$

Two classes-classification: logistic function.
Only 1 neuron! ($\hat{y_2} = 1 - \hat{y_1}$)

$$\hat{y}_i = \frac{1}{1 + e^{-z_i}}$$

Three or more classes-classification: Softmax
→ Pseudo-probabilities, i.e., N neurons, not N-1
think about it when one-hot encoding.

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$$

z_i	e^{z_i}	$\sum_{j=i}^{N} e^{Z_j}$	$\hat{y}_i$	$\sum_{j=i}^{N} \hat{y}_i$
2	7.3905	156.85	0.047410	1
5	148.41	156.85	0.952269	1
-3	0.0498	156.85	0.000319	1

Softmax does NOT keep proportionality.
It sharpens the decision.

Categorical values

DNA sequence: `AGGTCCGATTAGACTA`

Solution 1: associate a number with each nucleotide:

`0113221033010230`

Problem: Values are ranked. 1 thymidine is worth 3 guanosines, while 1 cytosine is worth 2 guanosines. What does than mean? And this is different is we choose 1234 instead of 0123.

Another example: maternal income classes of the finishNFBC86 cohort

1, Receives maternity benefit / parental benefit
2, Is at home, receives support of home care
3, Is at wage work / entrepreneur
4, Is unemployed, receives benefit for unemployment
5, Is in early / sickness / unemployment / retirement pension
6, Is student, receives financial aid to students
7, Is full-time mother without incomes

No relationship between class code and actual income, but the ANN does not know

$$z_j = \sum_i^N w_j i \cdot x_i + b_j$$

One-hot encoding for categorical values

	A	G	G	T	C	C	G	A	T	T	A	G	A	C	T	C
A	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
G	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0
C	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1
T	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0

Each input is now a vector,
each value being associated to its own weight

One-hot encoding for categorical values

	A	G	G	T	C	C	G	A	T	T	A	G	A	C	T	C
A	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
G	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0
C	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1
T	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0

Or even
(full rank matrix)

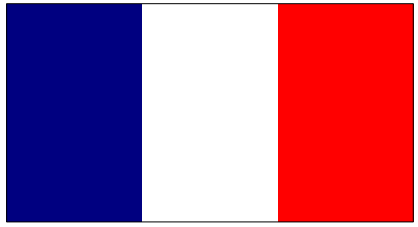
	A	G	G	T	C	C	G	A	T	T	A	G	A	C	T	C
A	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
G	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0
C	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1

(if not A, not G, not C, then it is a T...)

For input, not outputs! (remember, softmax are pseudo-probabilities. Sum is 1)

How about analysing images?

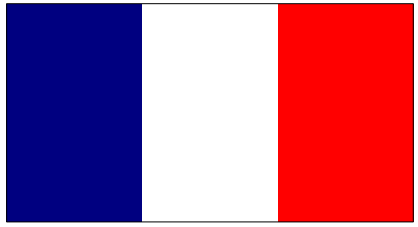
FR



DE

How about analysing images?

FR



DE

R

0	255	255
0	255	255
0	255	255

0	0	0
255	255	255
255	255	255

G

0	255	0
0	255	0
0	255	0

0	0	0
0	0	0
255	255	255

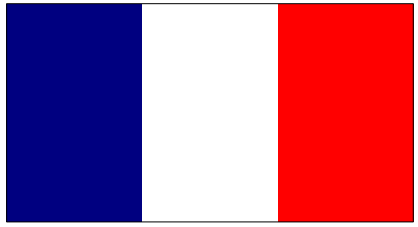
B

128	255	0
128	255	0
128	255	0

0	0	0
0	0	0
0	0	0

How about analysing images?

FR



DE

R

0	255	255
0	255	255
0	255	255

0	0	0
255	255	255
255	255	255

G

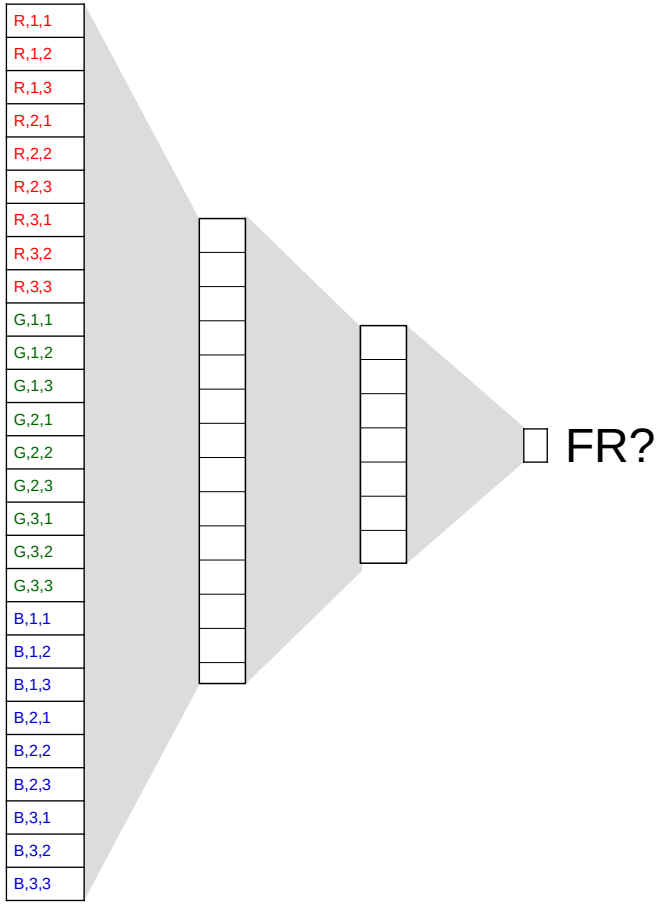
0	255	0
0	255	0
0	255	0

0	0	0
0	0	0
255	255	255

B

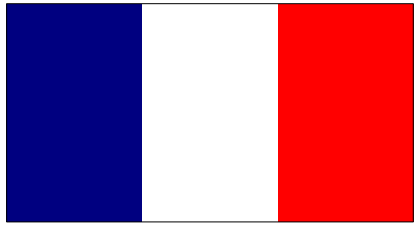
128	255	0
128	255	0
128	255	0

0	0	0
0	0	0
0	0	0



Problem: parameter explosion

FR



DE

R	0	255	255
	0	255	255
	0	255	255

R	0	0	0
	255	255	255
	255	255	255

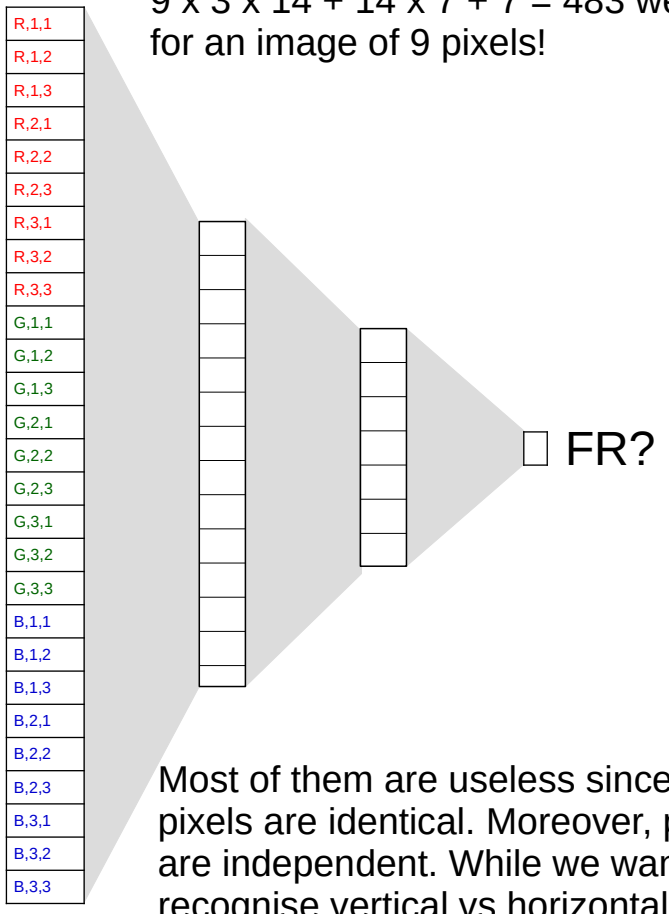
G	0	255	0
	0	255	0
	0	255	0

G	0	0	0
	0	0	0
	255	255	255

B	128	255	0
	128	255	0
	128	255	0

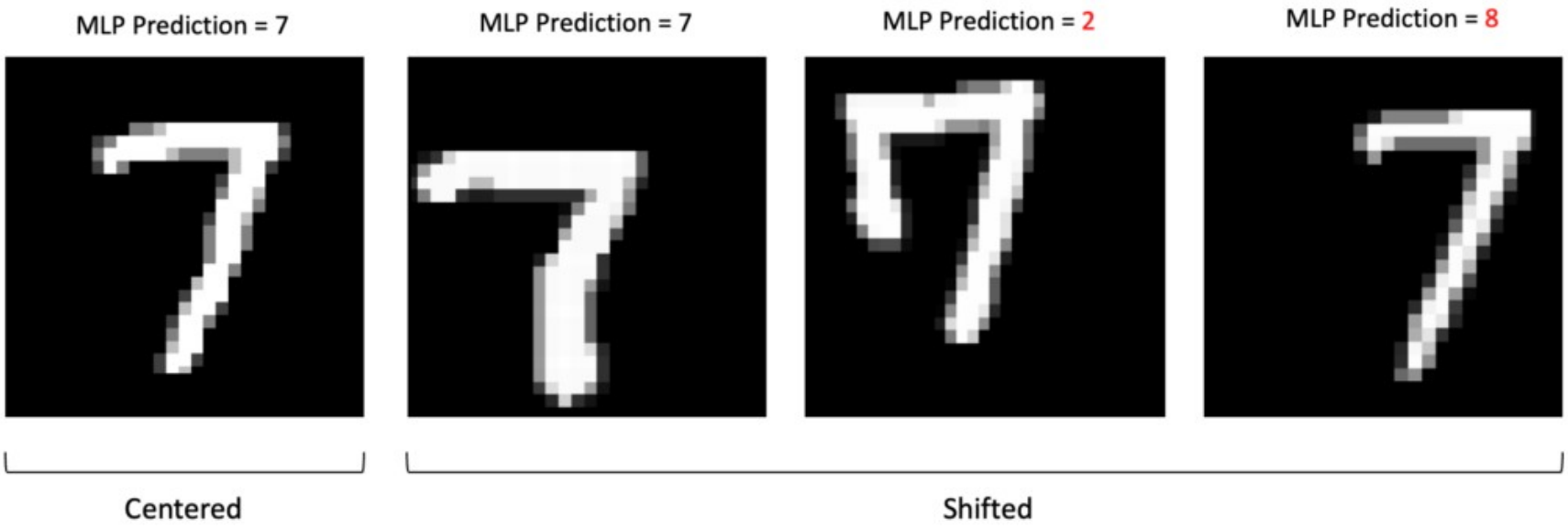
B	0	0	0
	0	0	0
	0	0	0

9 x 3 x 14 + 14 x 7 + 7 = 483 weights for an image of 9 pixels!

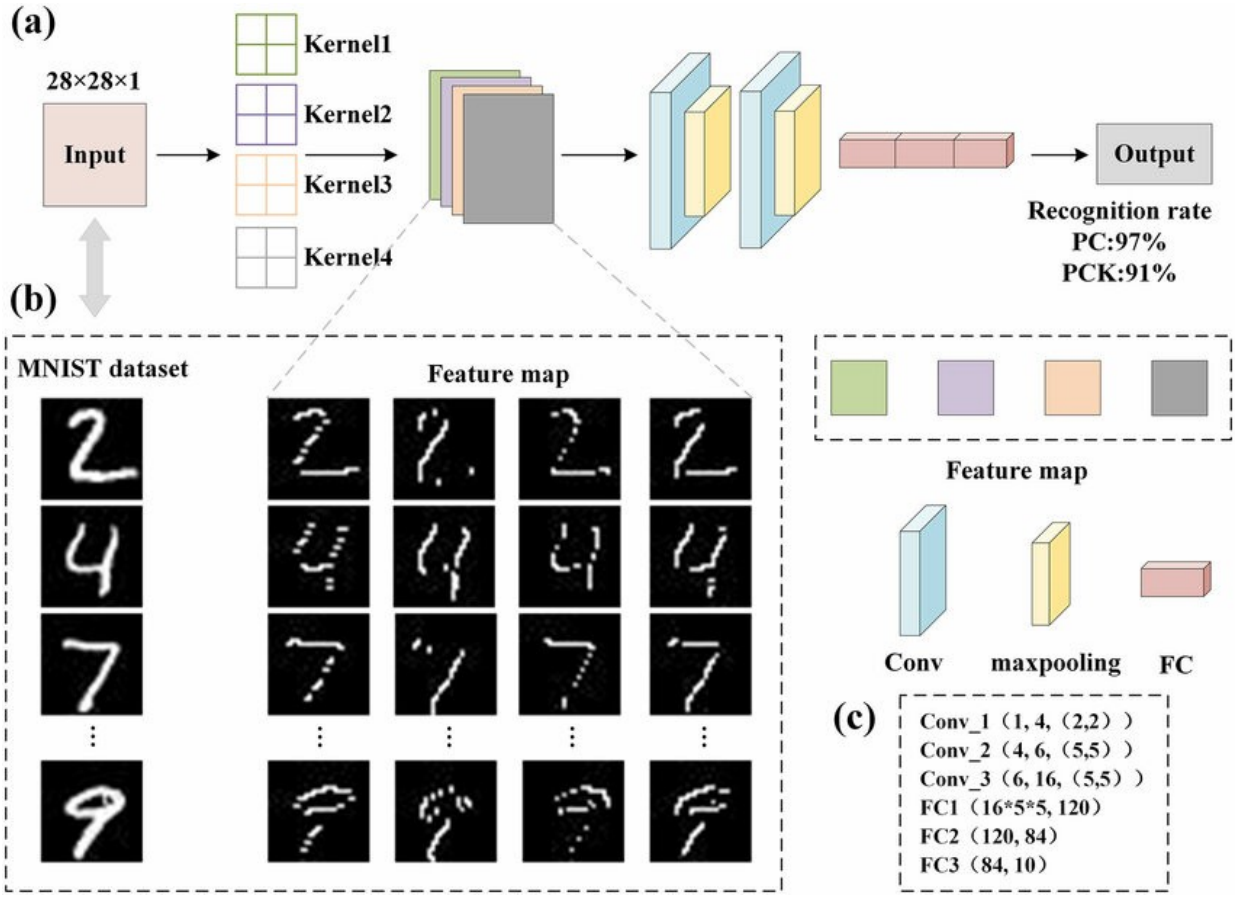


Most of them are useless since most pixels are identical. Moreover, pixels are independent. While we want to recognise vertical vs horizontal boundaries, and 3 colours

Problem: lack of translation invariance

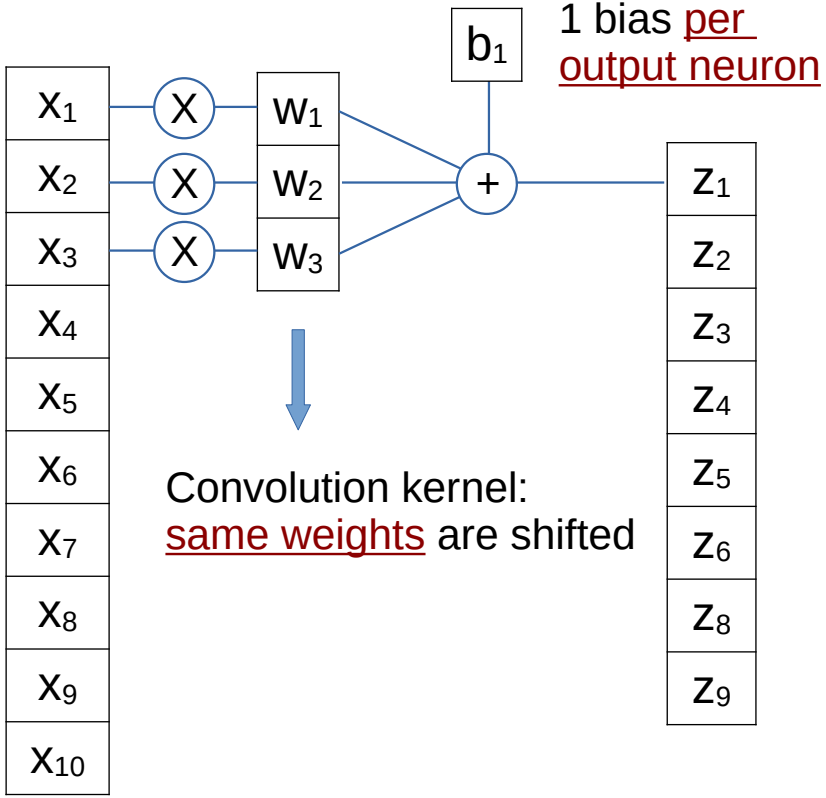


Convolutional Neural Networks (CNN)



Find features, no matter their size, no matter where in the image, and use these features to recognize, classify, predict

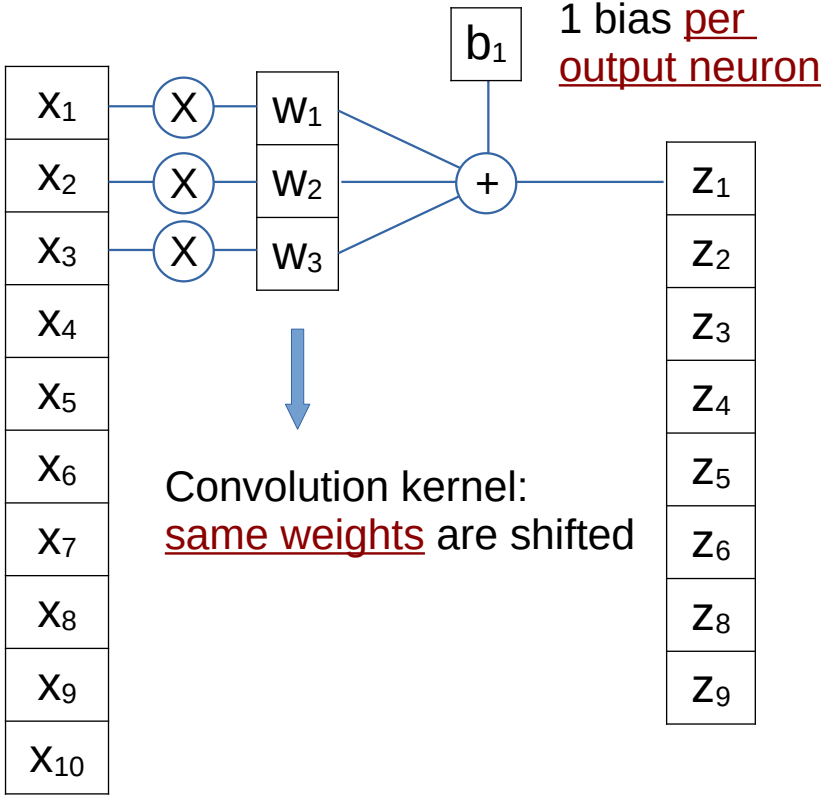
Convolutional neural networks: local correlations between inputs

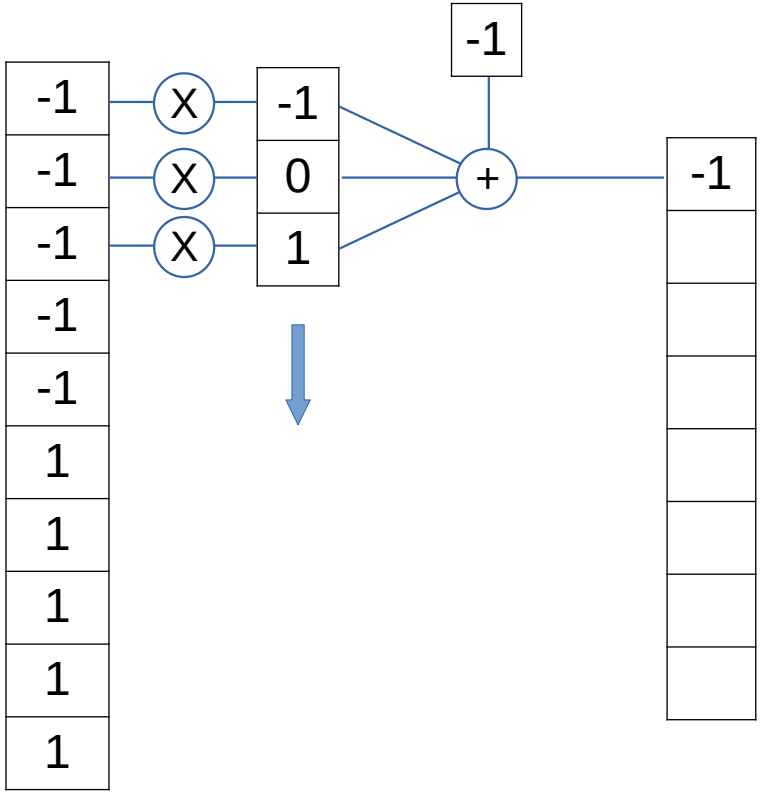


$$z_1 = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 + b$$

Fukushima K (1979) Neural network model for a mechanism of pattern recognition unaffected by shift in position —Neocognitron. *Trans. IECE*, J62-A(10): 658-665

Convolutional neural networks: local correlations between inputs

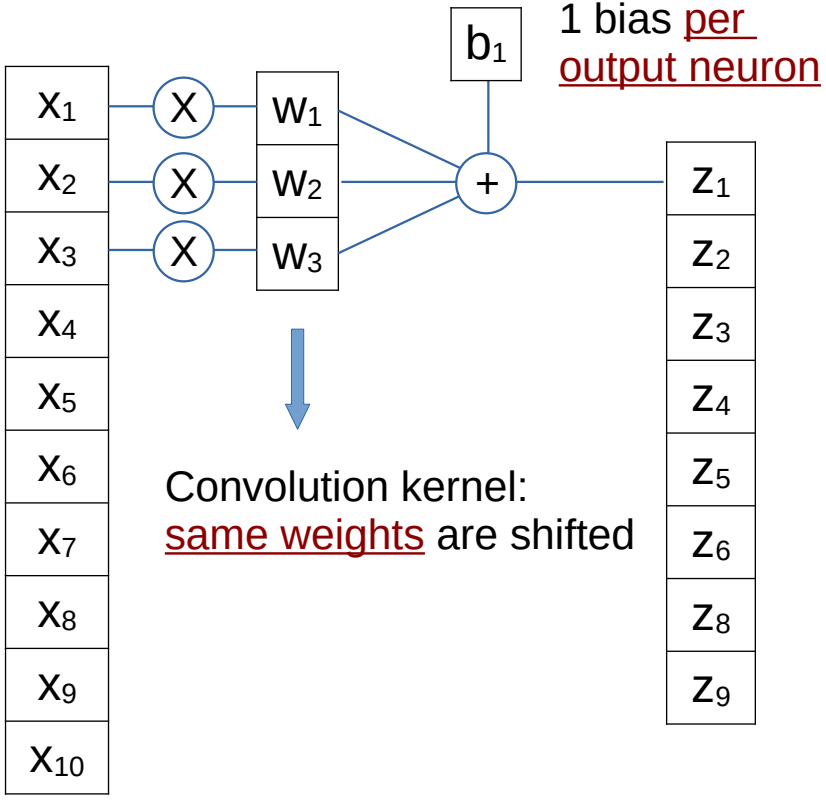


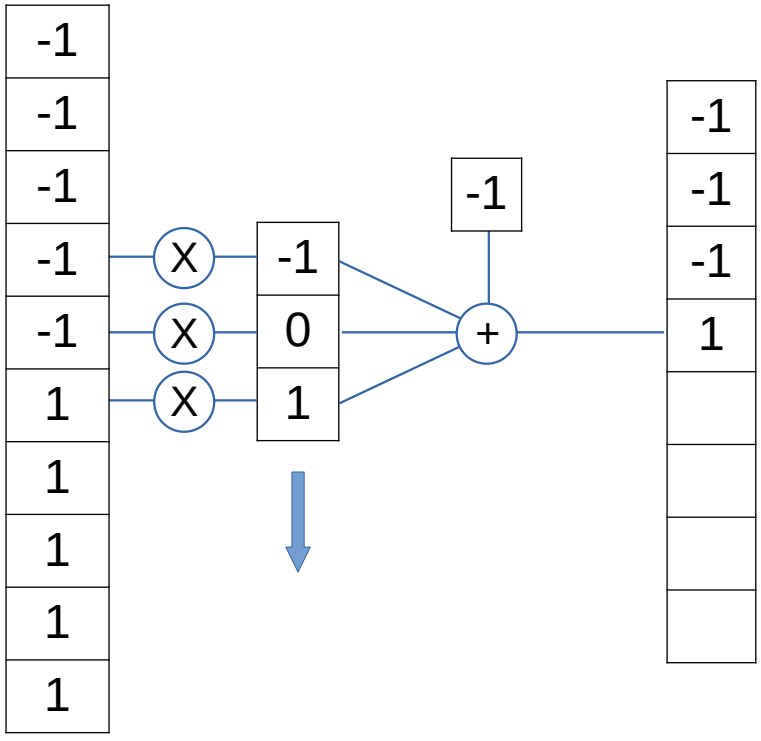


$$z_1 = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 + b$$

$$z_1 = -1 \times (-1) + 0 \times (-1) + 1 \times (-1) + (-1)$$

Convolutional neural networks: local correlations between inputs

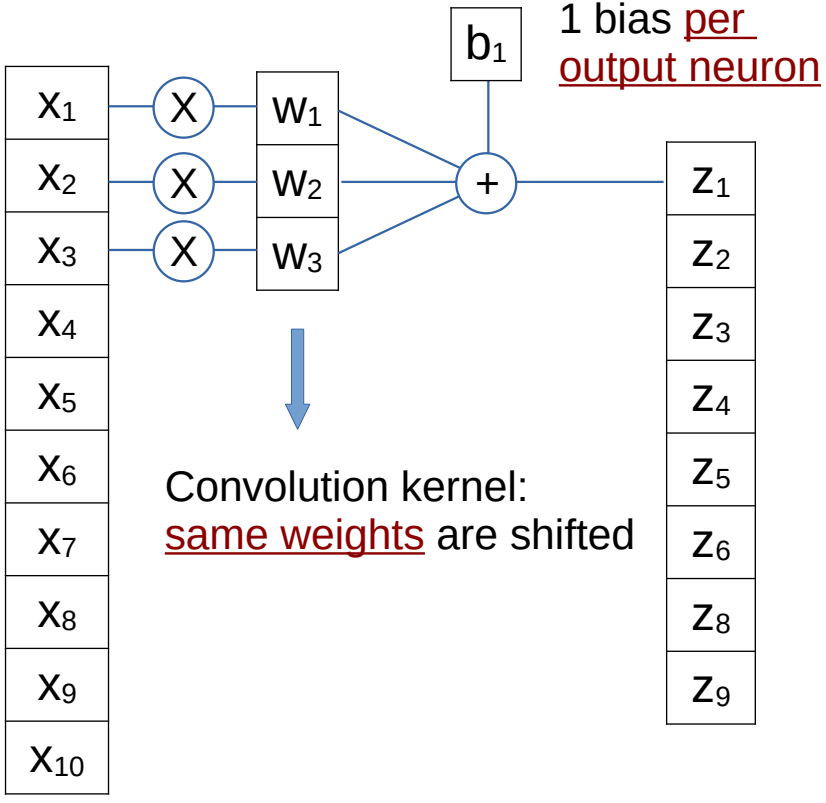


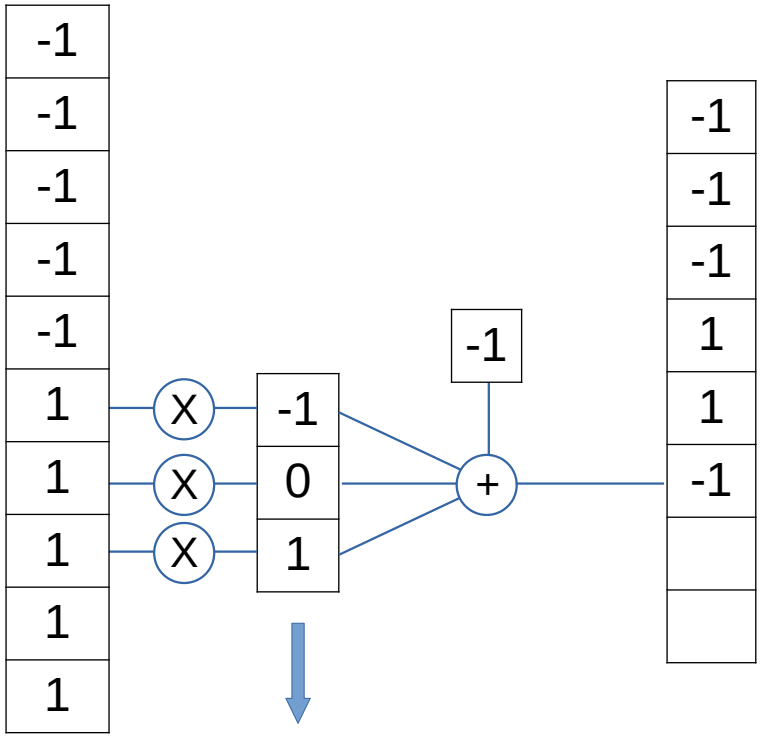


$$z_1 = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 + b$$

$$z_4 = -1 \times (-1) + 0 \times (-1) + 1 \times 1 + (-1)$$

Convolutional neural networks: local correlations between inputs

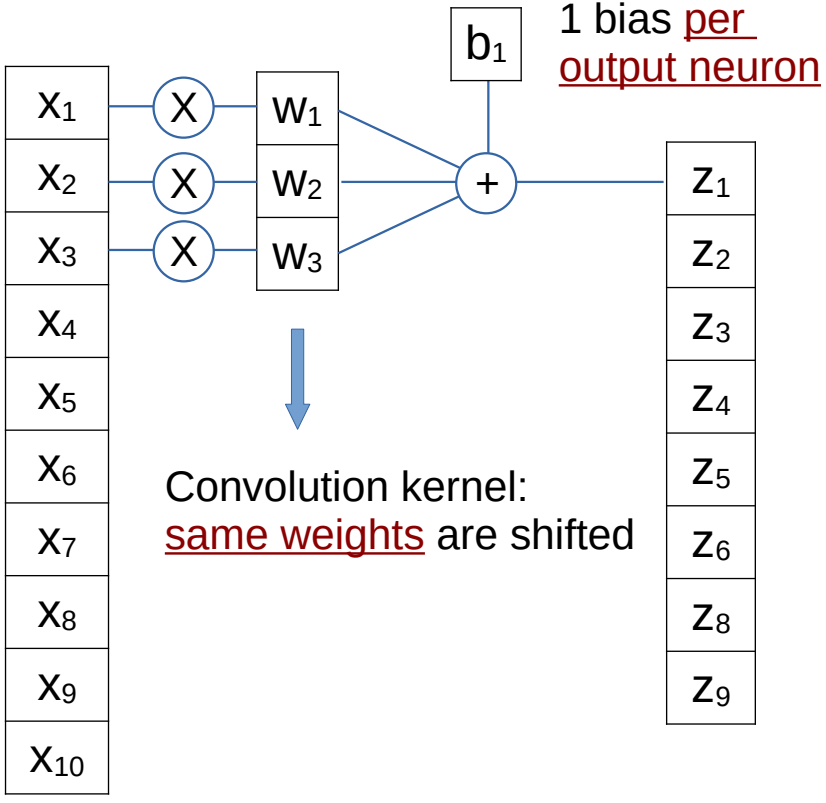




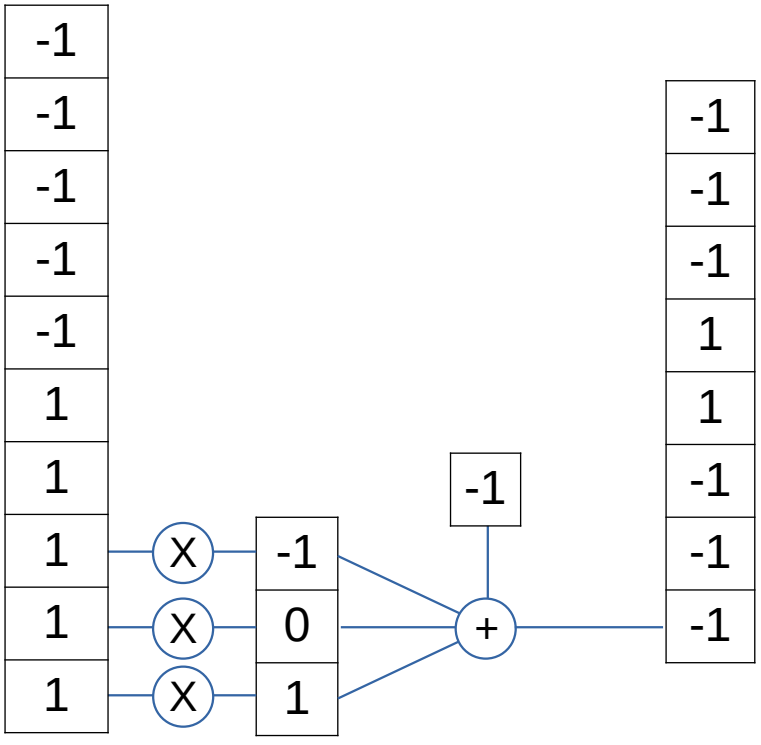
$$z_1 = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 + b$$

$$z_6 = -1 \times 1 + 0 \times 1 + 1 \times 1 + (-1)$$

Convolutional neural networks: local correlations between inputs

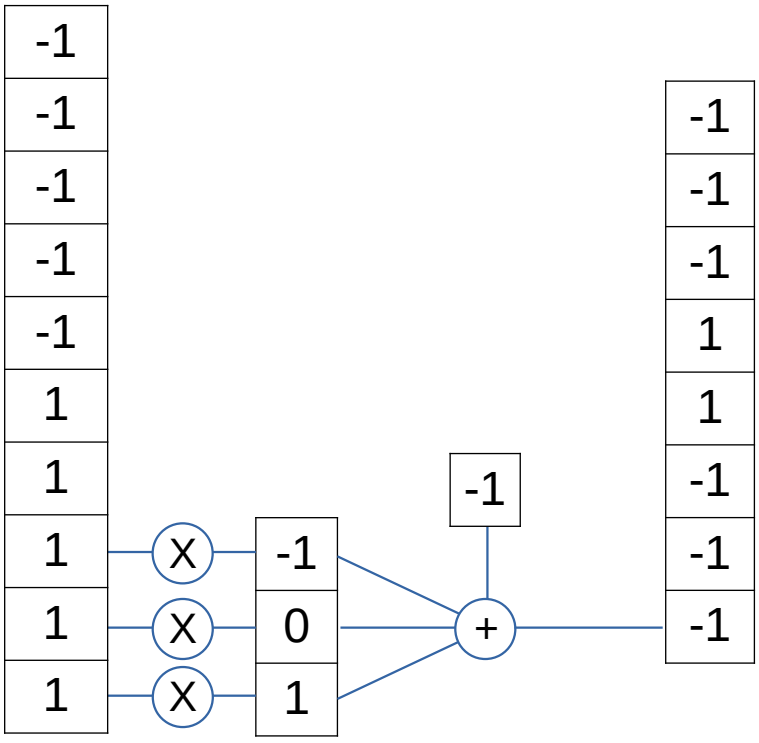


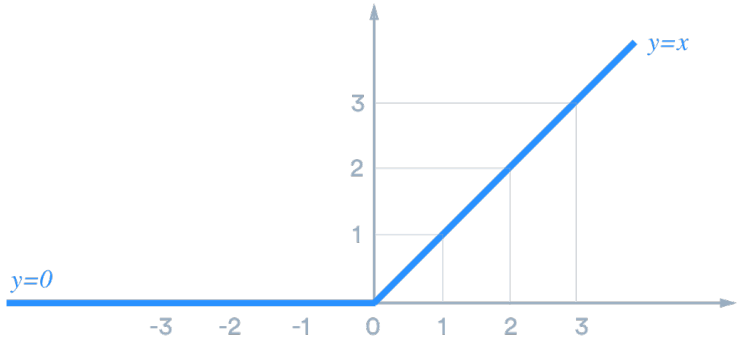
$$z_1 = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 + b$$



$$z_8 = -1 \times 1 + 0 \times 1 + 1 \times 1 + (-1)$$

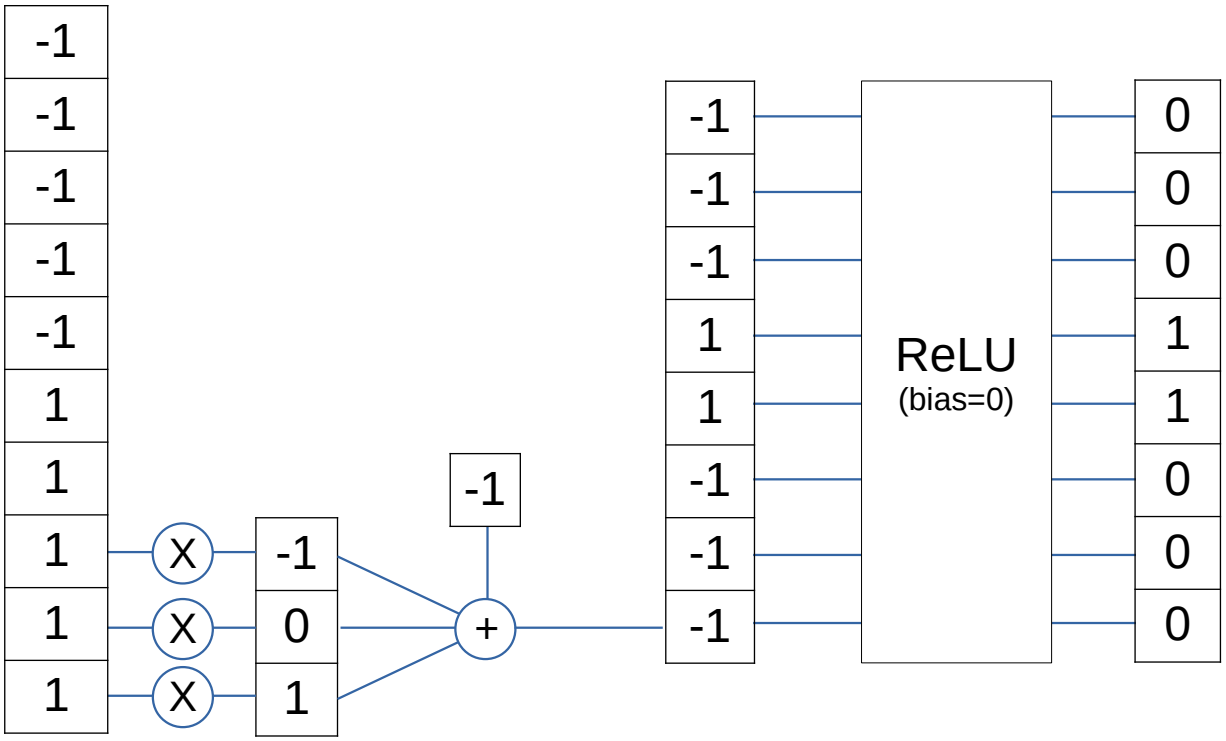
Adding ReLU (Rectified Linear Unit)



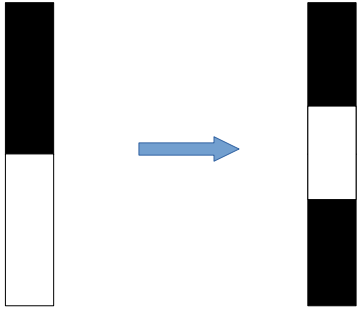


$$f(x) = max(0, x) \qquad \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

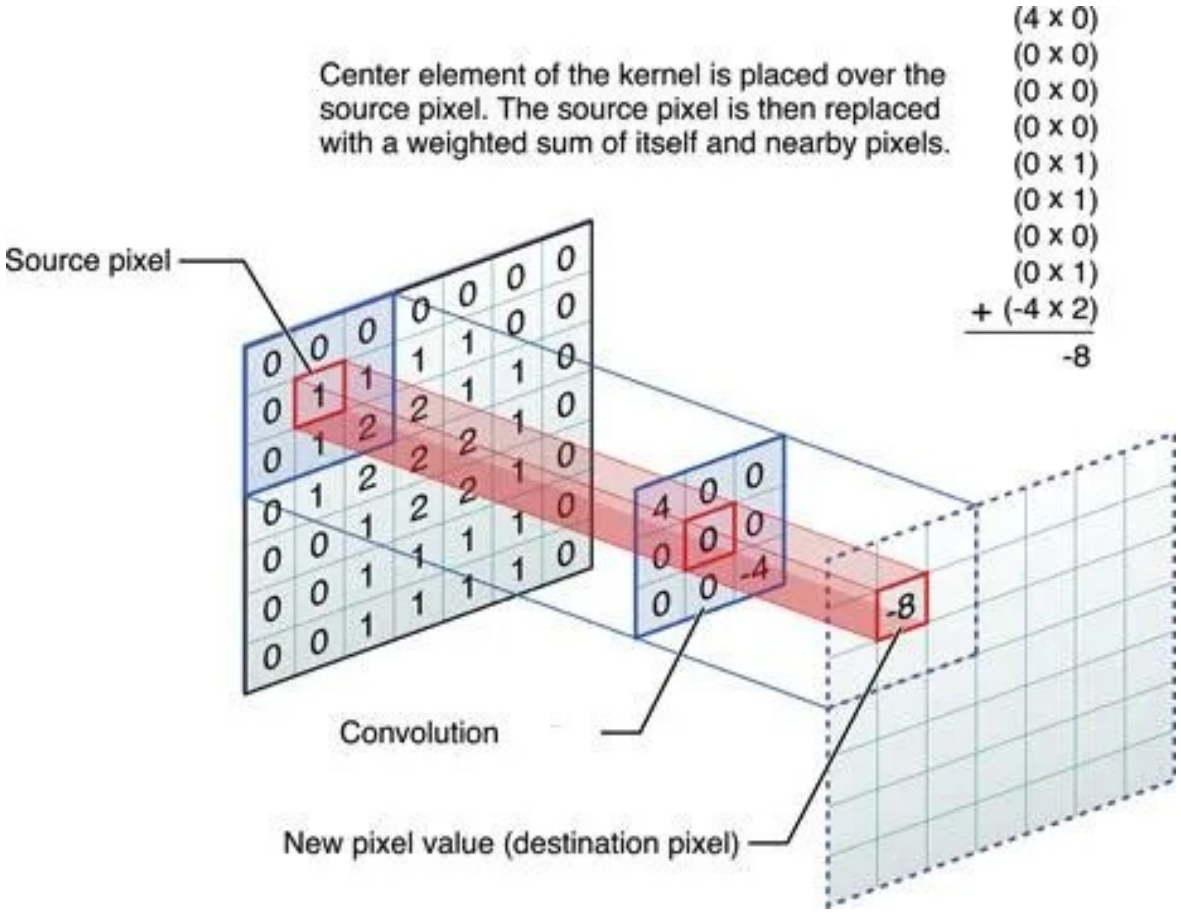
Adding ReLU (Rectified Linear Unit)



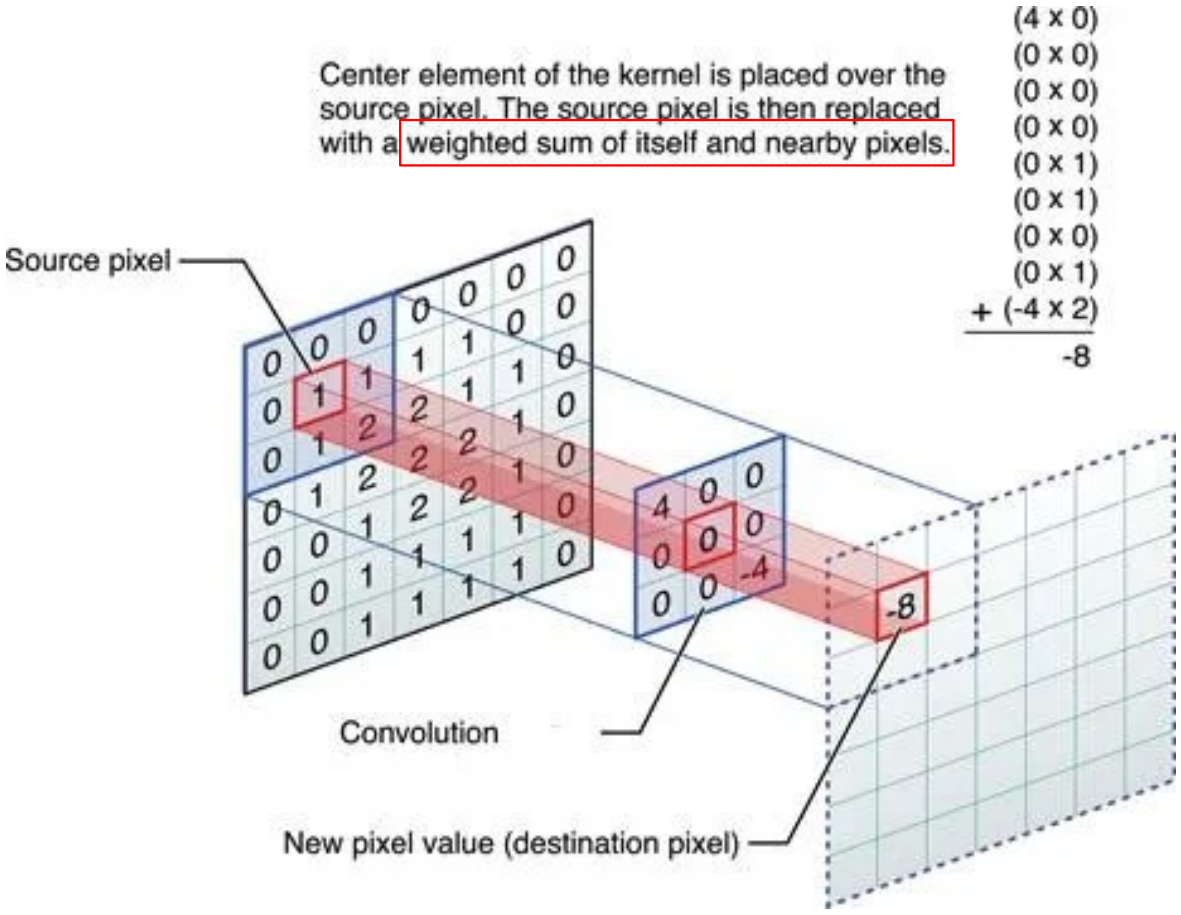
Returns a binary representation (presence/absence) of the domain boundaries



2D convolution



2D convolution

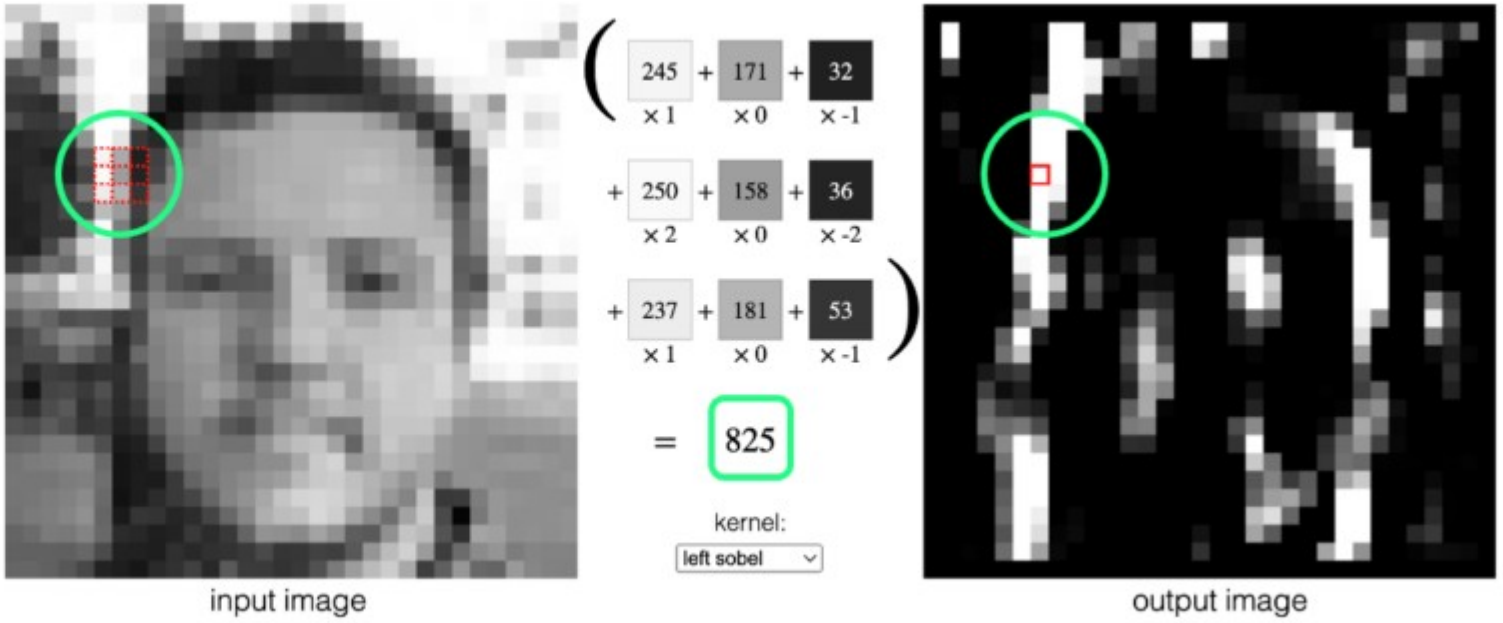


Detection of local features, such as edges, orientation, colour gradients, etc

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

Convolution “erodes” the image

Pixel Values

0	0	0	0	0	0	0	0
0	1	0	4	2	125	67	0
0	8	2	5	4	34	12	0
0	20	13	25	15	240	2	0
0	76	8	6	6	100	76	0
0	34	66	134	223	201	3	0
0	255	123	89	55	32	2	0
0	0	0	0	0	0	0	0

Kernel 3 x 3 Pixels

1	2	1
2	4	2
1	2	1

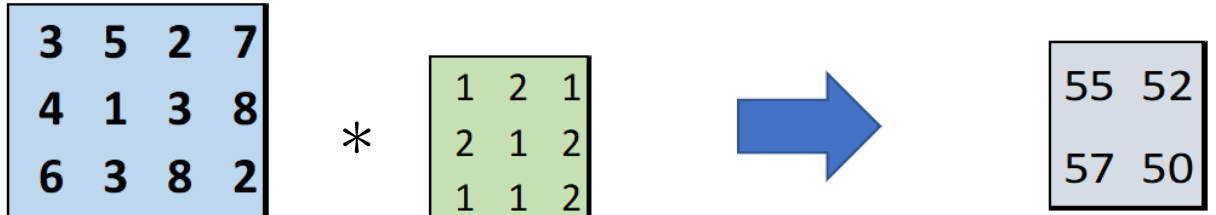
Convoluted Image

22	27	36	313	722	576
91	110	120	522	984	576
284	257	198	755	1360	798
507	567	687	1312	1689	955
1061	1288	1496	1911	1659	702
1400	1480	1269	1249	870	279

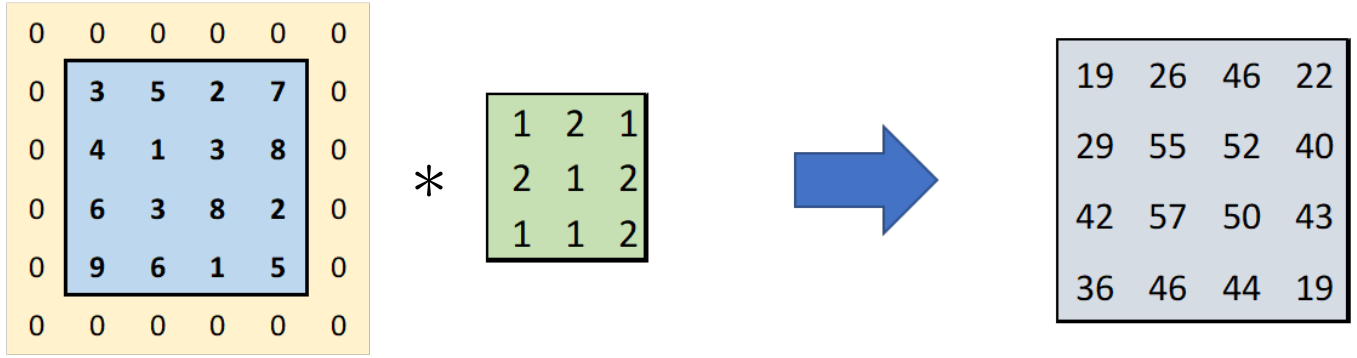
Padding to avoid erosion

Keras

Padding: “valid”



Padding: “same”



Padding to avoid erosion

Keras

Padding: “valid”

3	5	2	7
4	1	3	8
6	3	8	2
9	6	1	5

$*$

1	2	1
2	1	2
1	1	2

NB1: To make the output independent of kernel size, we can divide the output by the number of weights (here, 9).

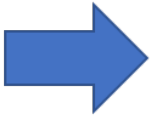
NB2: For each feature, there are generally as many kernels as layers in the image, e.g. 3 for RGB images. The outputs are averaged.

Padding: “same”

0	0	0	0	0	0
0	3	5	2	7	0
0	4	1	3	8	0
0	6	3	8	2	0
0	9	6	1	5	0
0	0	0	0	0	0

$*$

1	2	1
2	1	2
1	1	2

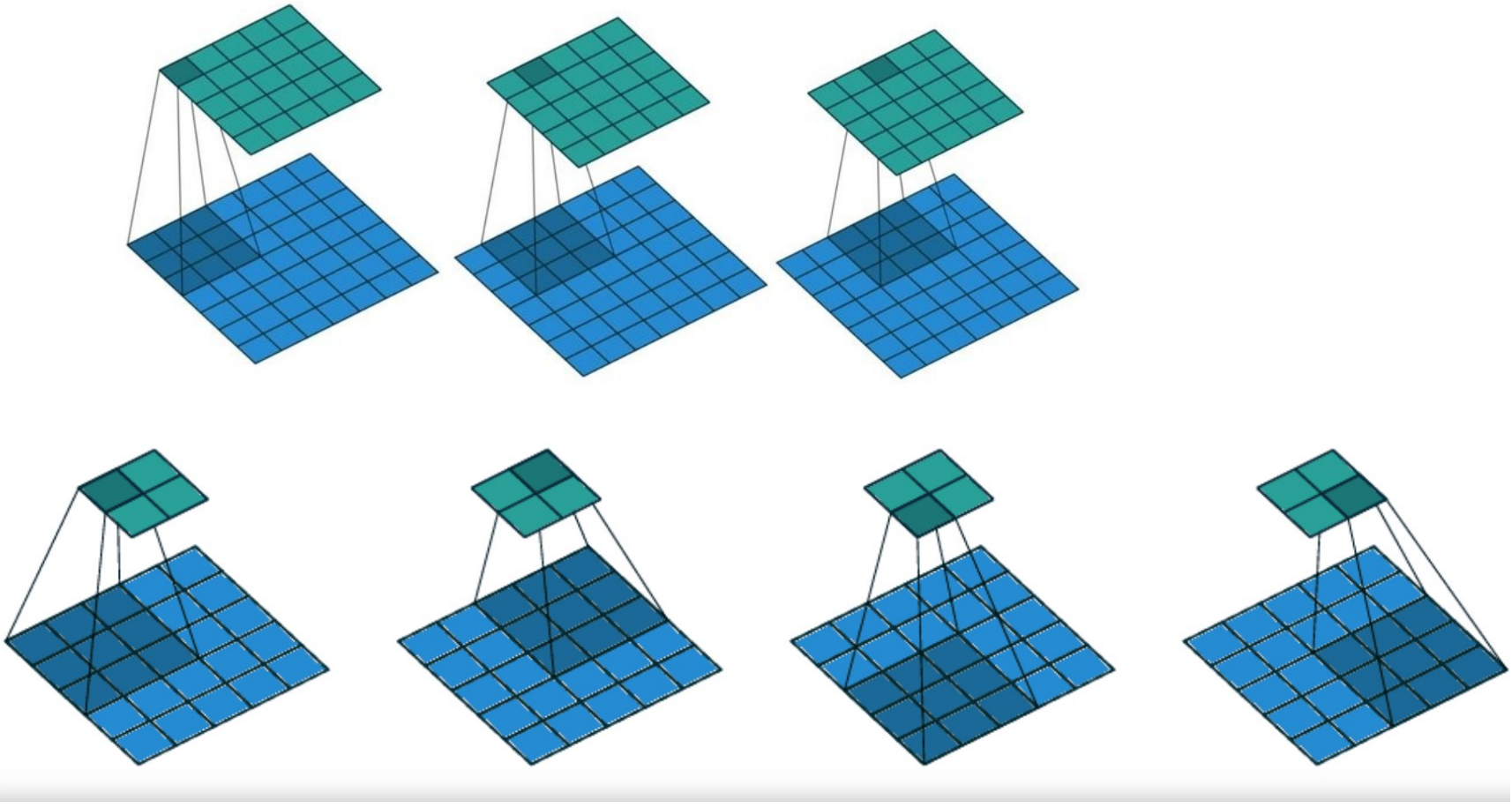


19	26	46	22
29	55	52	40
42	57	50	43
36	46	44	19

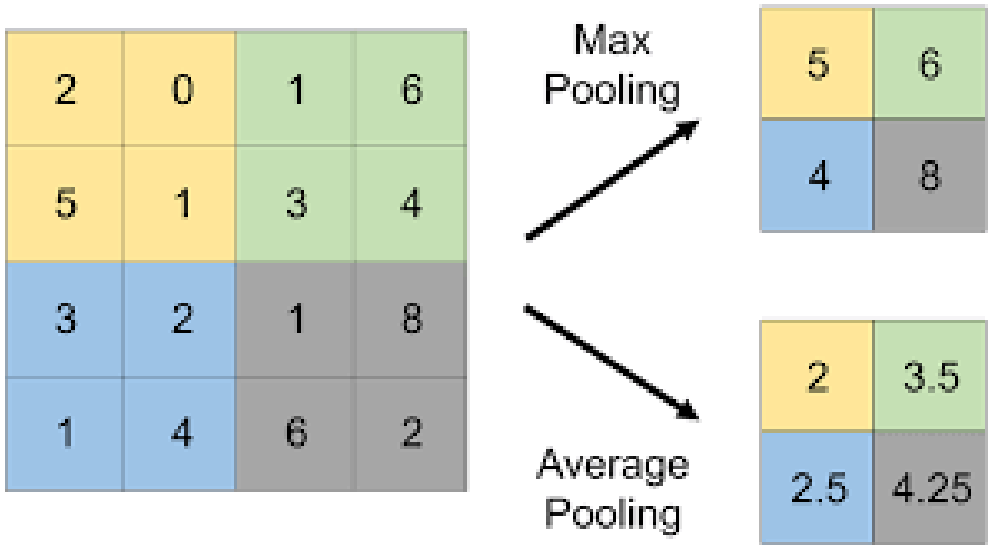
Stride to speed up processing (and erosion)

Stride = 1

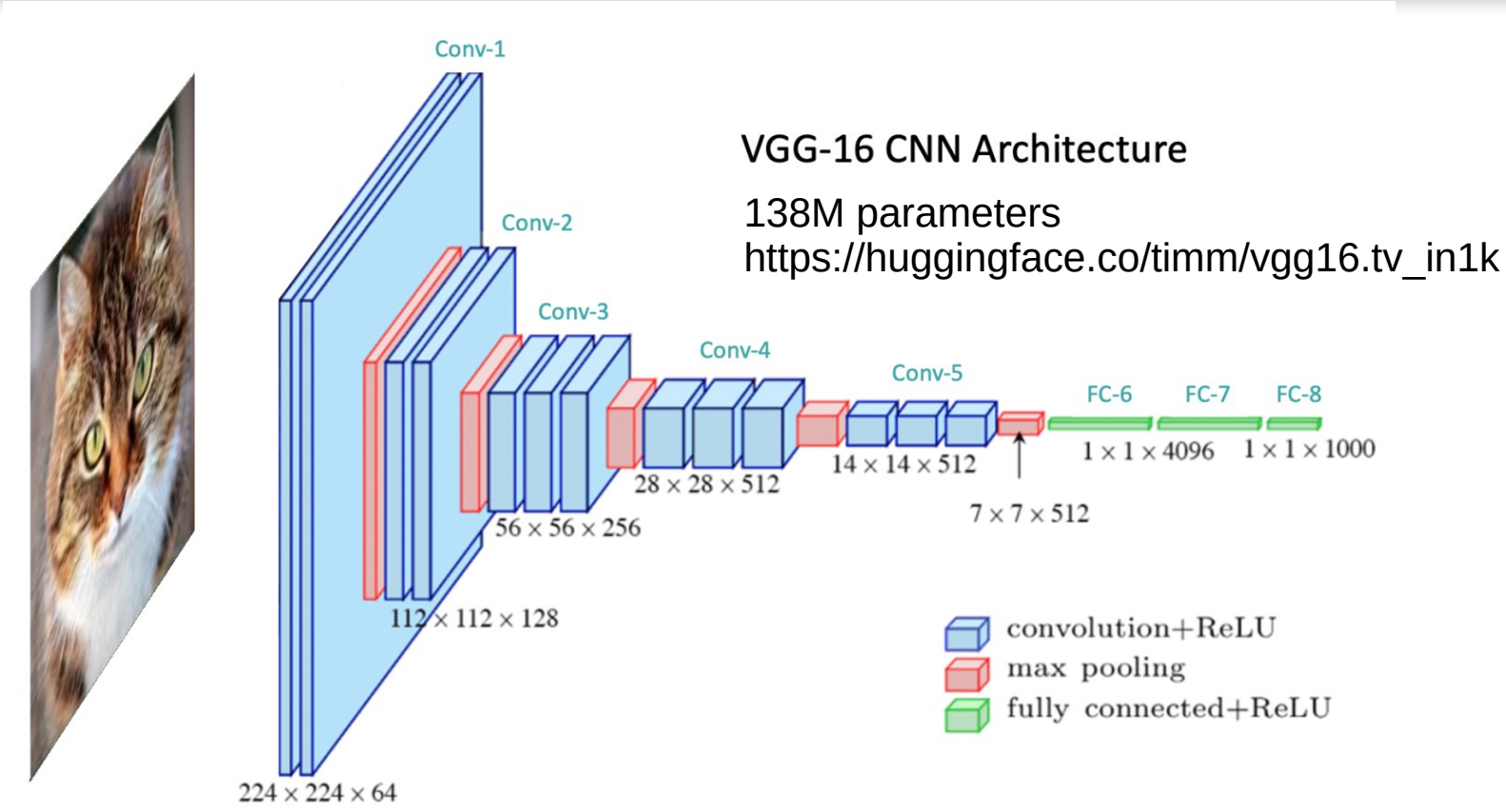
Stride = 2



Downsampling: Max/average pooling

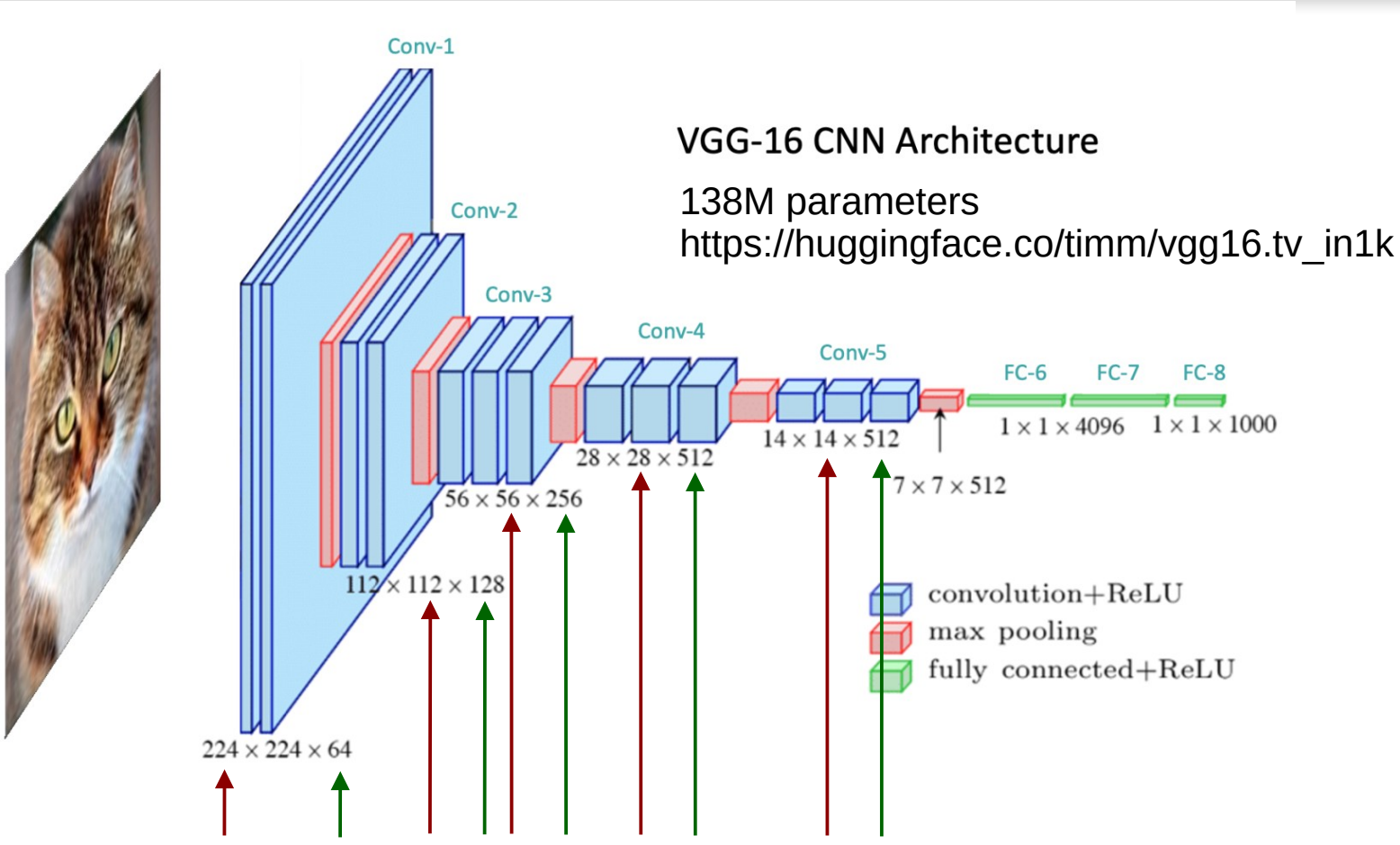


Real example: VGG



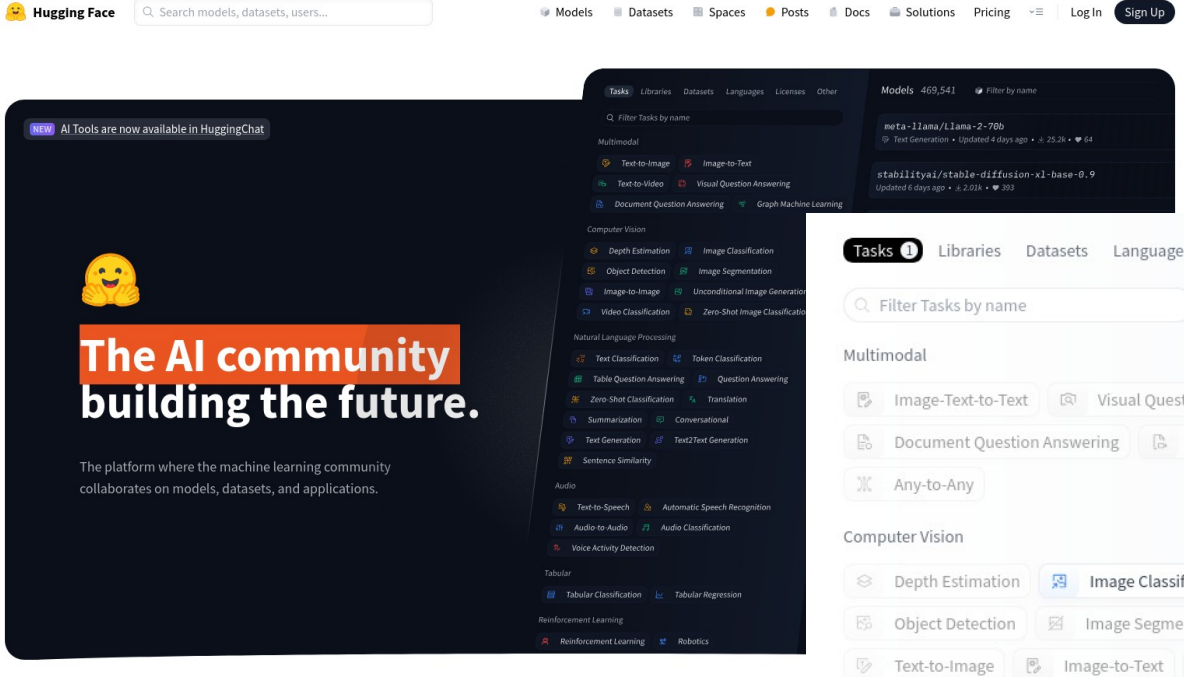
Simonyan K, Zisserman A (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015
https://arxiv.org/pdf/1409.1556

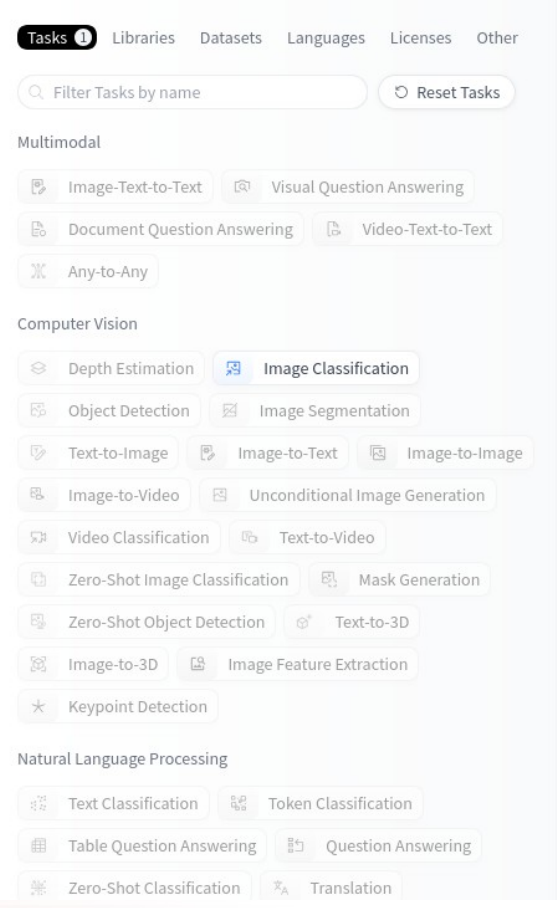
Decreasing size, increasing feature number

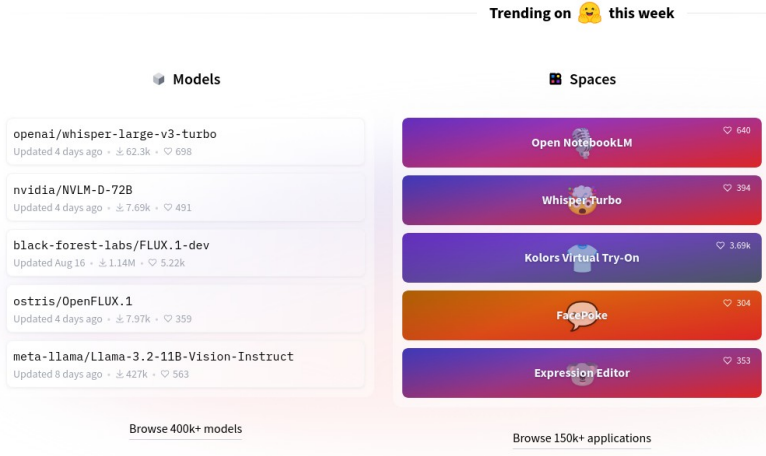


Simonyan K, Zisserman A (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015
https://arxiv.org/pdf/1409.1556

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

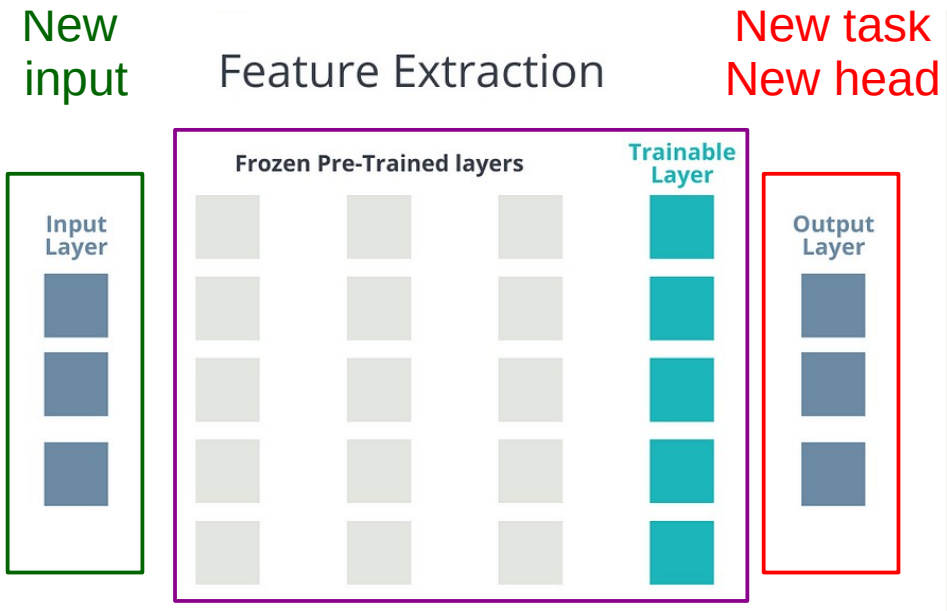
December 2023

DOI:10.1109/SMARTGENCON60755.2023.10442010

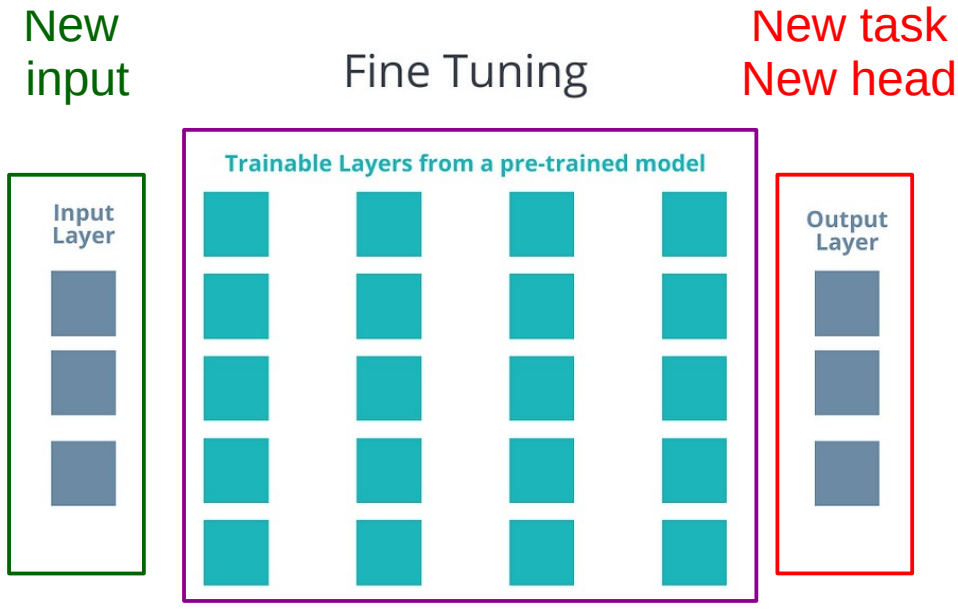
Conference: 2023 3rd International Conference on Smart Generation Computing,

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



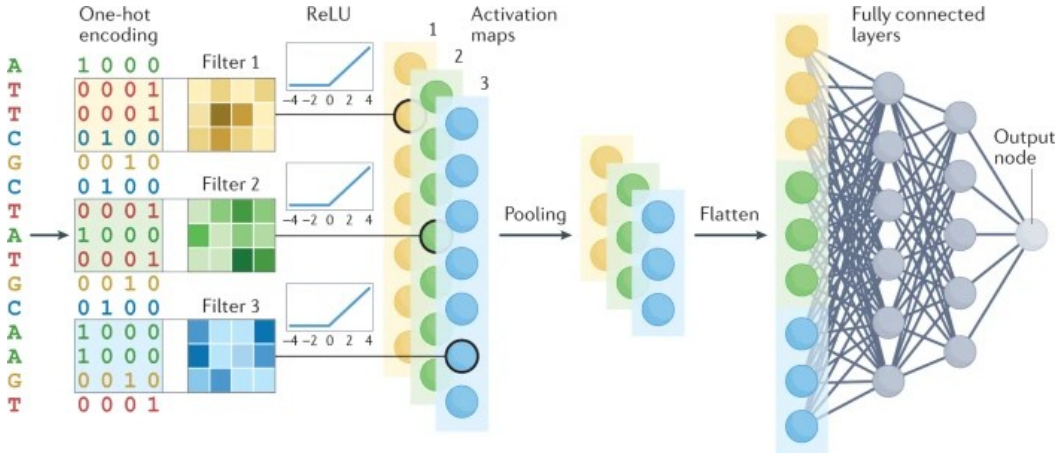
In feature extraction, you **freeze the pre-trained model** layers to preserve existing learning and **add new layers** to learn additional information.



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

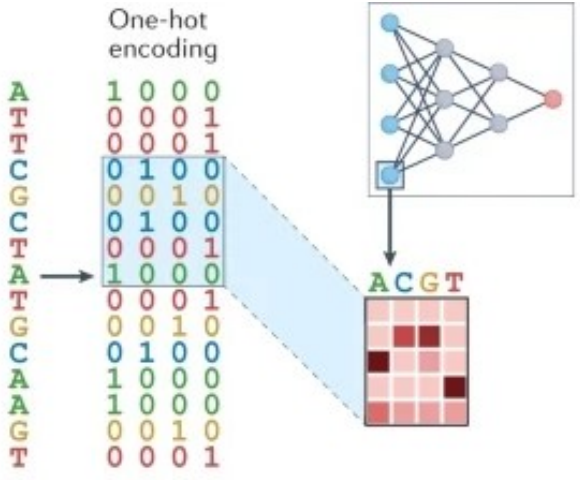
Everything is an image: DNA sequences to images

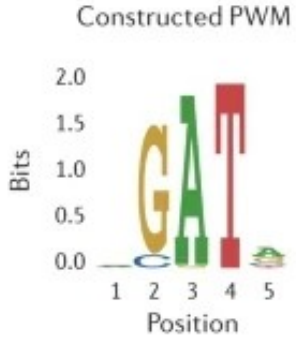


Obtaining genetics insights from deep learning via explainable artificial intelligence

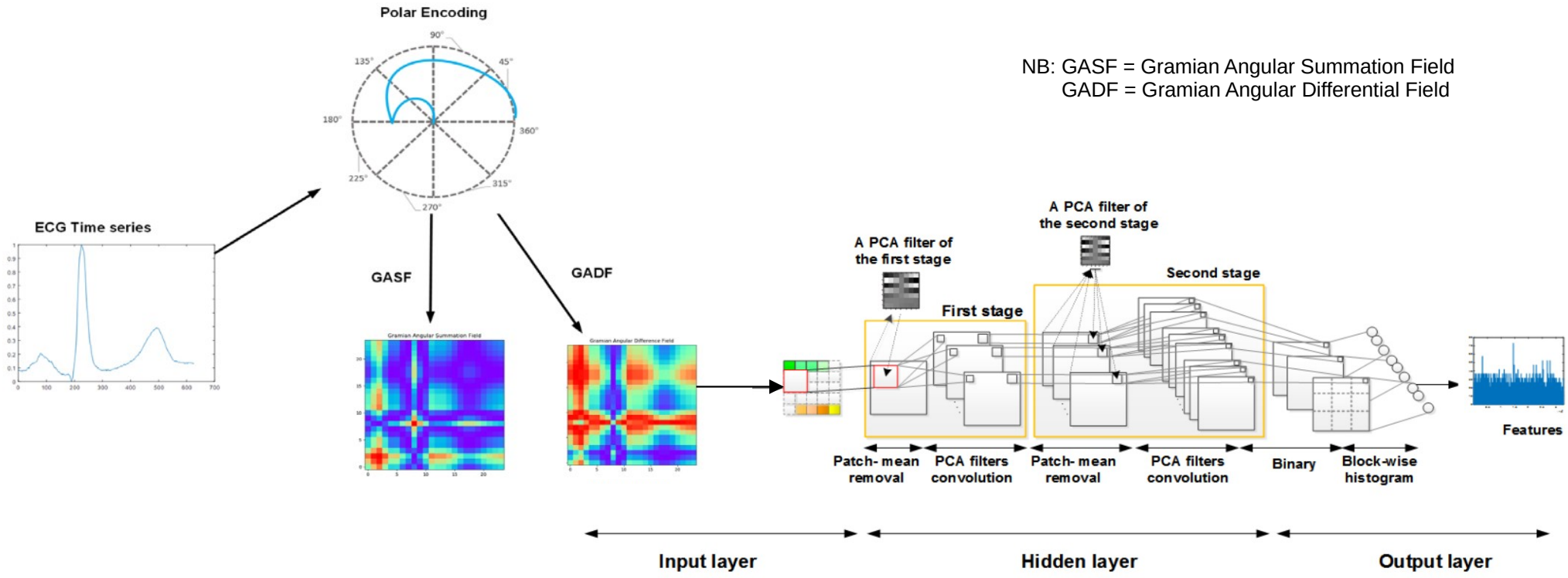
Gherman Novakovsky, Nick Dexter, Maxwell W. Libbrecht ✉, Wyeth W. Wasserman ✉ & Sara Mostafavi ✉

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





Everything is an image: Time series to images

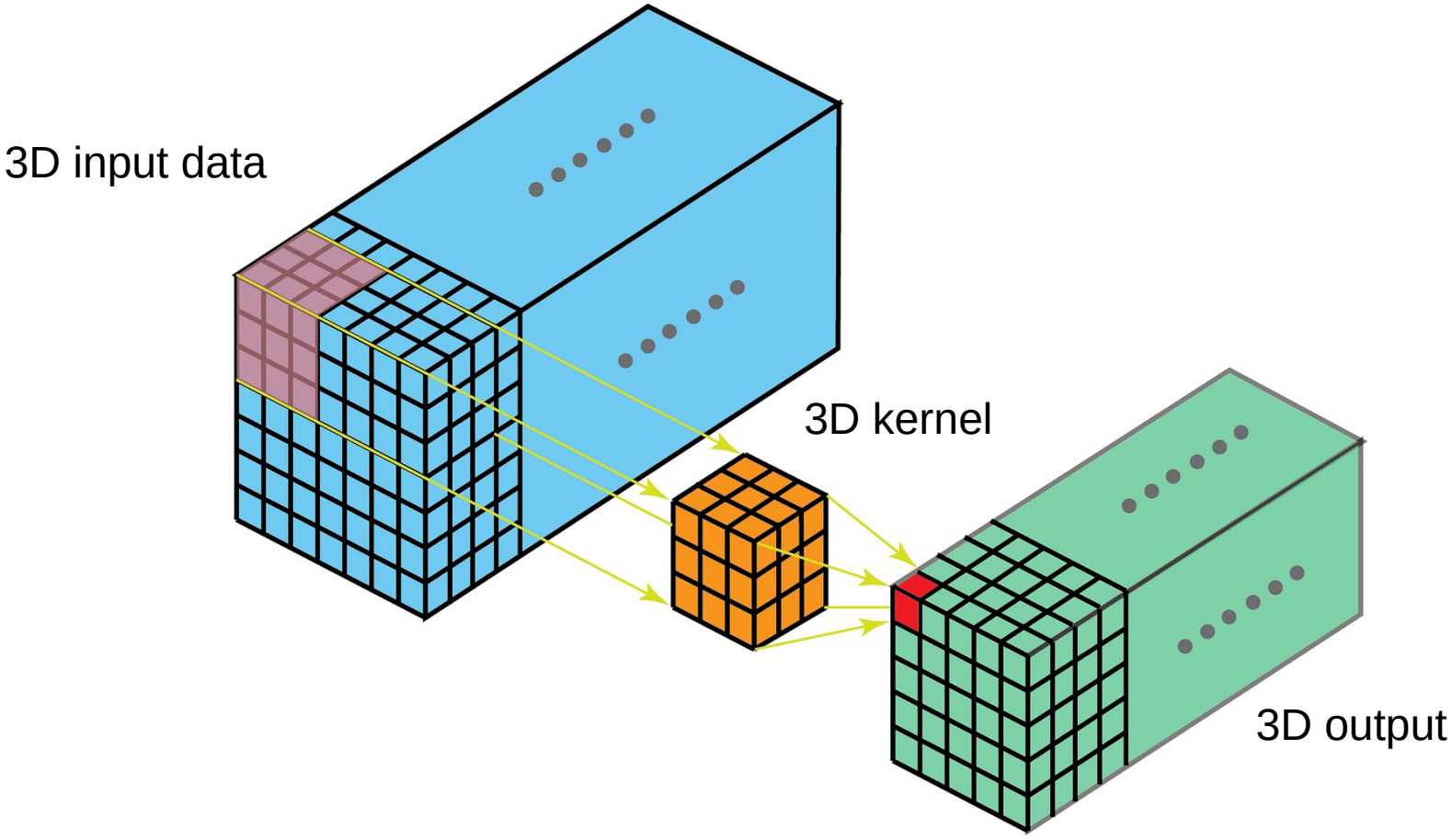


NB: GASF = Gramian Angular Summation Field
GADF = Gramian Angular Differential Field

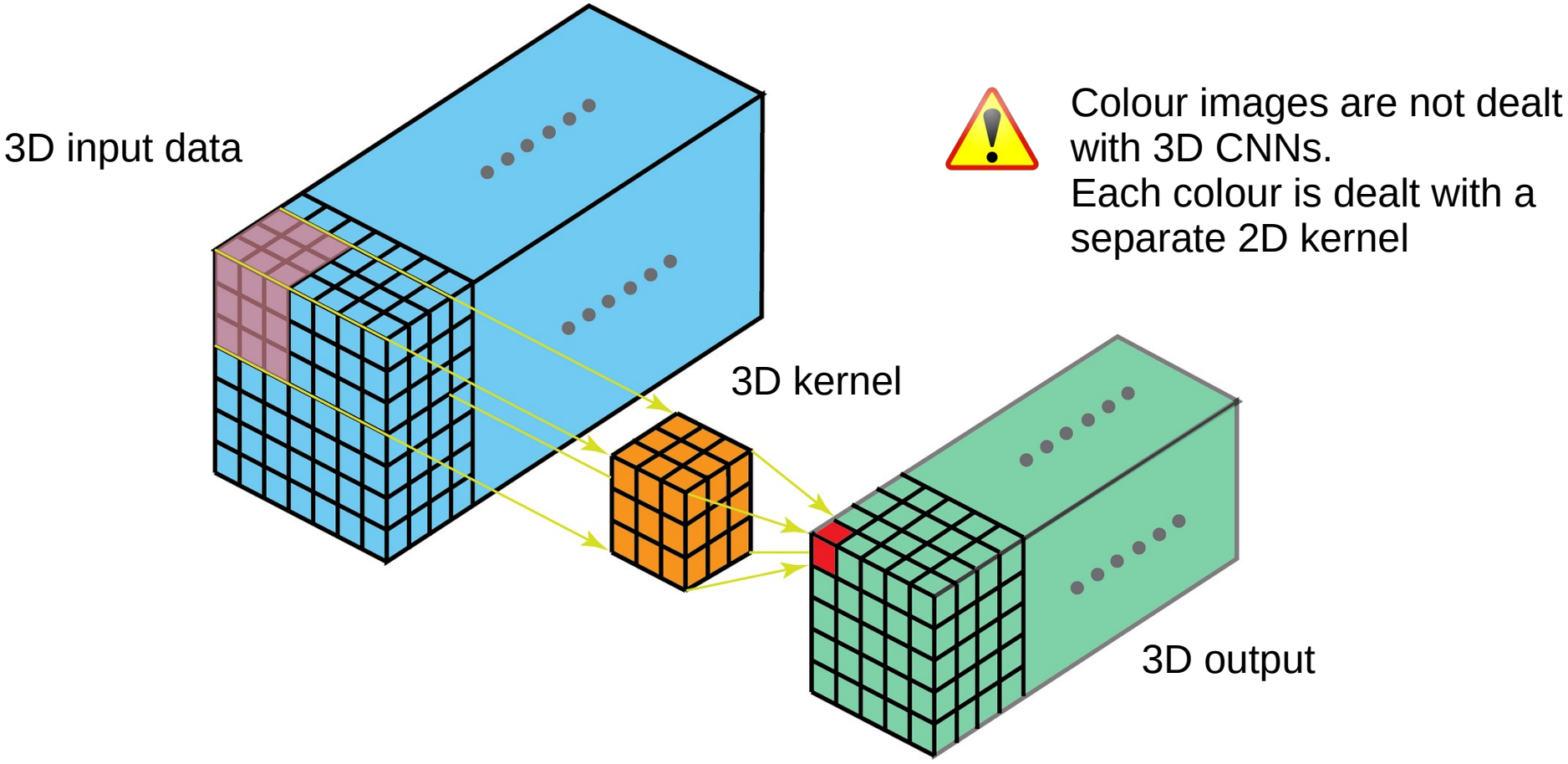
Wang and Oates (2015) Spatially Encoding Temporal Correlations to Classify Temporal Data Using Convolutional Neural Networks. arXiv:1509.07481v1

Zhang *et al* (2019) Automated Detection of Myocardial Infarction Using a Gramian Angular Field and Principal Component Analysis Network. *IEEE Access*, 7:171570-171583. doi:10.1109/ACCESS.2019.2955555

CNNs do not stop at 2D



CNNs do not stop at 2D



Colour images are not dealt with 3D CNNs.
Each colour is dealt with a separate 2D kernel

CNNs overfit too

Explain the Prediction

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Husky
True: Husky

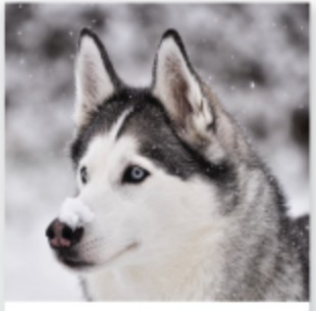
Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Wolf

Predicted: Wolf
True: Wolf



Predicted: Wolf
True: Husky

Predicted: Husky
True: Husky

CNNs overfit too

Predicted: Wolf
True: Husky

Explanation of the decision: The network recognises the snow...

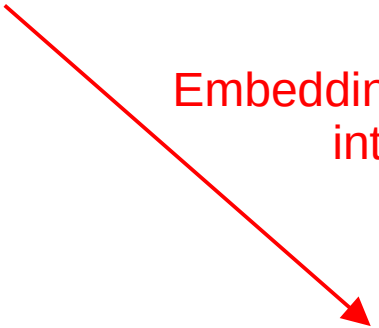
Tulio Ribeiro M, Singh S, Guestrin C (2016) "Why Should I Trust You?": Explaining the Predictions of Any Classifier. KDD '16: Proc 22nd ACM SIGKDD Intl Conf Knowledge Discov Data Mining, 1135 – 1144 https://arxiv.org/pdf/1602.04938

NB1: This paper introduced the Lime algorithm.

NB2: They ***hand-picked*** the dataset so the model overfitted (the point of the experiment was to evaluate the trust put in the model by observers, before and after they were given the explanation)

Embeddings (“plongement”)

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



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

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

Embeddings (“plongement”)

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

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

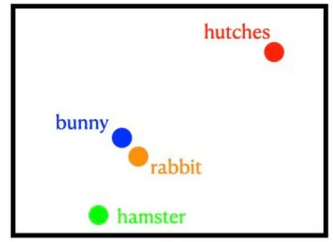
Semantics (Embedding space)

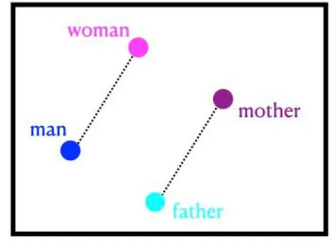
Word	alive	mammal	rodent	bipedel	gender	plural
bunny	0.9	0.7	-0.1	-0.1	0.2	-0.3
rabbit	0.9	0.8	-0.1	-0.1	0.4	-0.1
hamster	0.9	0.6	0.7	-0.3	-0.4	-0.4
hutches	-0.9	-0.3	-0.1	-0.9	0.0	0.9
mother	0.9	0.1	0.1	0.2	0.8	-0.7
father	0.9	0.2	0.1	0.2	-0.8	-0.7
woman	0.9	0.8	-0.9	0.9	0.8	-0.8
man	0.9	0.8	-0.7	0.9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

Vector Embedding

Dimensionality Reduction

Dimensionality





2D Visualization

Mikolov *et al* (2015) Linguistic Regularities in Continuous Space Word Representations In *Proceedings of NAACL-HLT,* pp 746–751, Atlanta, Georgia.

Embeddings (“plongement”)

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

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

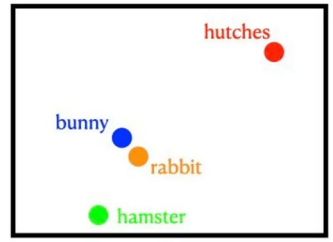
Semantics (Embedding space)

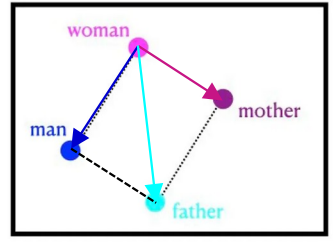
Word	alive	mammal	rodent	bipedel	gender	plural
bunny	0.9	0.7	-0.1	-0.1	0.2	-0.3
rabbit	0.9	0.8	-0.1	-0.1	0.4	-0.1
hamster	0.9	0.6	0.7	-0.3	-0.4	-0.4
hutches	-0.9	-0.3	-0.1	-0.9	0.0	0.9
mother	0.9	0.1	0.1	0.2	0.8	-0.7
father	0.9	0.2	0.1	0.2	-0.8	-0.7
woman	0.9	0.8	-0.9	0.9	0.8	-0.8
man	0.9	0.8	-0.7	0.9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

Vector Embedding

Dimensionality Reduction

Dimensionality





2D Visualization

In the embedding space, the vector going from woman to father is equal to the vector going from woman to man plus the vector going from woman to mother

A PCA is an embedding

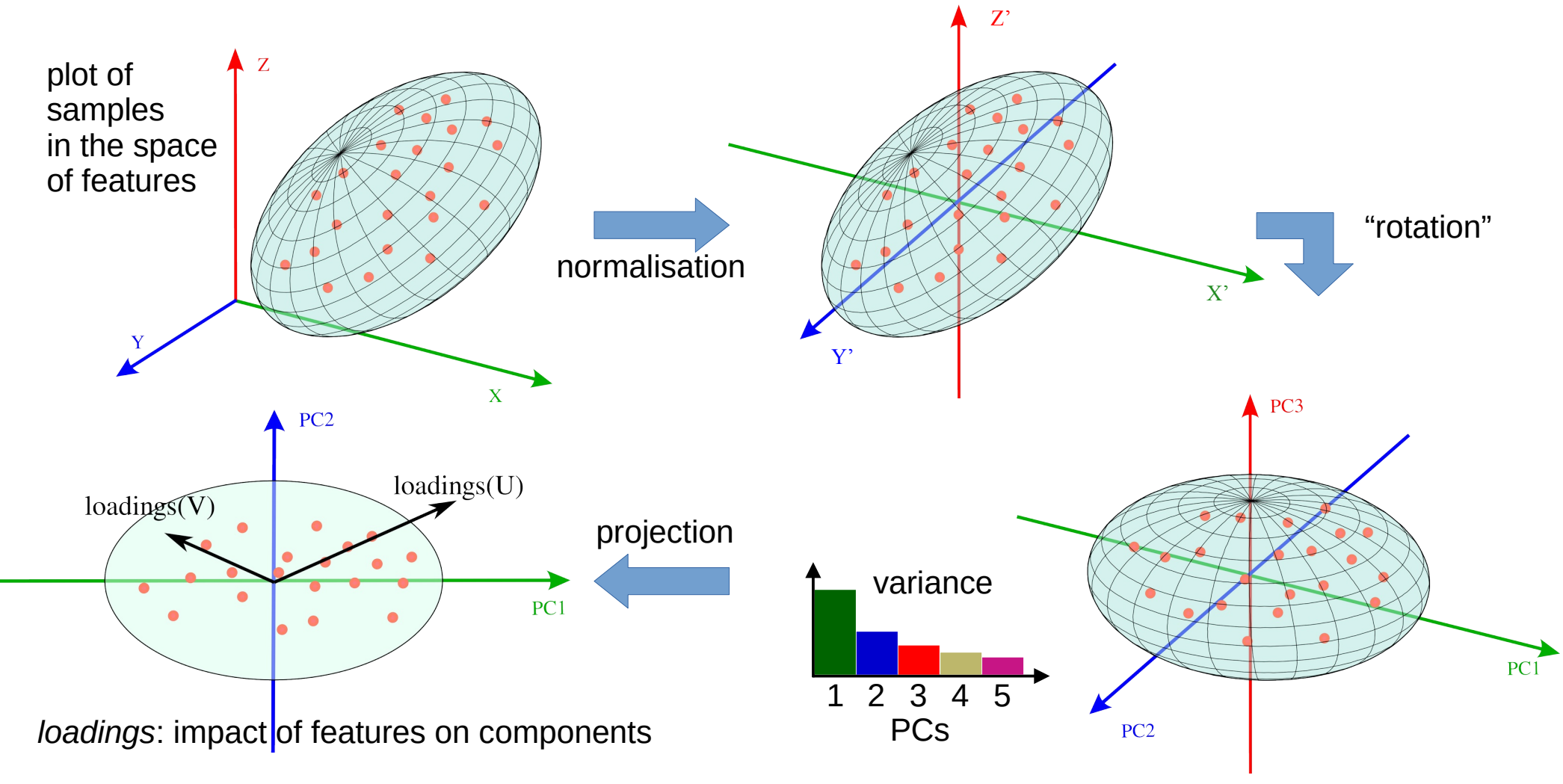
Coordinates = N genes

G_1
G_2
G_3
G_4
...
G_i
...
G_N

Coordinates = M principal components

PC_1
PC_2
PC_3
PC_4
...
PC_j
...
PC_M

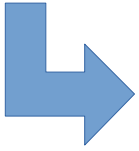
What is the principal component analysis (PCA)?



loadings: impact of features on components

What is the principal component analysis (PCA)?

$$
\begin{aligned}
&S_1 : (G_{1,1}, G_{1,2}, \ldots, G_{1,g}) \\
&S_2 : (G_{2,1}, G_{2,2}, \ldots, G_{2,g}) \\
&\ldots : \ldots \\
&S_s : (G_{s,1}, G_{s,2}, \ldots, G_{s,g})
\end{aligned}
$$



$$
\begin{aligned}
S_1 : (&\alpha_{\mathrm{PC1,G_1}} \times G_{1,1}, \alpha_{\mathrm{PC1,G_2}} \times G_{1,2}, \ldots, \alpha_{\mathrm{PC1,G_g}} \times G_{1,g} \\
&\alpha_{\mathrm{PC2,G_1}} \times G_{1,1}, \alpha_{\mathrm{PC2,G_2}} \times G_{1,2}, \ldots, \alpha_{\mathrm{PC2,G_g}} \times G_{1,g} \\
&\ldots \\
&\alpha_{\mathrm{PCp,G_1}} \times G_{1,1}, \alpha_{\mathrm{PCp,G_2}} \times G_{1,2}, \ldots, \alpha_{\mathrm{PCp,G_g}} \times G_{1,g}
\end{aligned}
$$

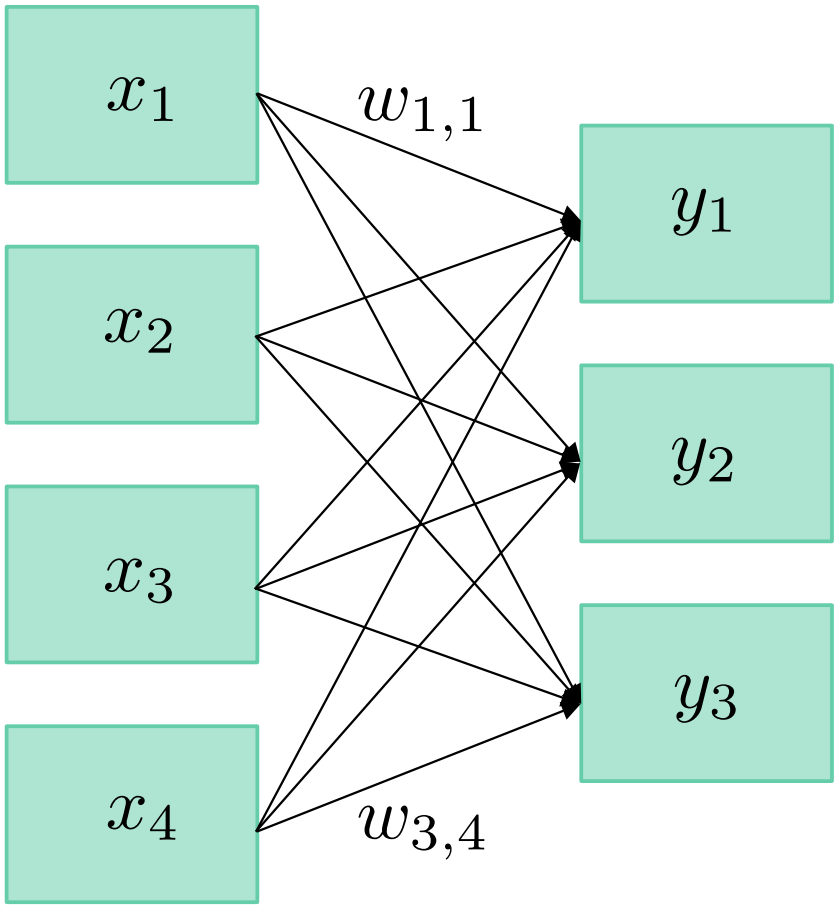
$$
\begin{aligned}
S_2 : (&\alpha_{\mathrm{PC1,G_1}} \times G_{2,1}, \alpha_{\mathrm{PC1,G_2}} \times G_{2,2}, \ldots, \alpha_{\mathrm{PC1,G_g}} \times G_{2,g}, \\
&\alpha_{\mathrm{PC2,G_1}} \times G_{2,1}, \alpha_{\mathrm{PC2,G_2}} \times G_{2,2}, \ldots, \alpha_{\mathrm{PC2,G_g}} \times G_{2,g}, \\
&\ldots, \\
&\alpha_{\mathrm{PCp,G_1}} \times G_{2,1}, \alpha_{\mathrm{PCp,G_2}} \times G_{2,2}, \ldots, \alpha_{\mathrm{PCp,G_g}} \times G_{2,g}
\end{aligned}
$$

$$
\ldots : \ldots
$$

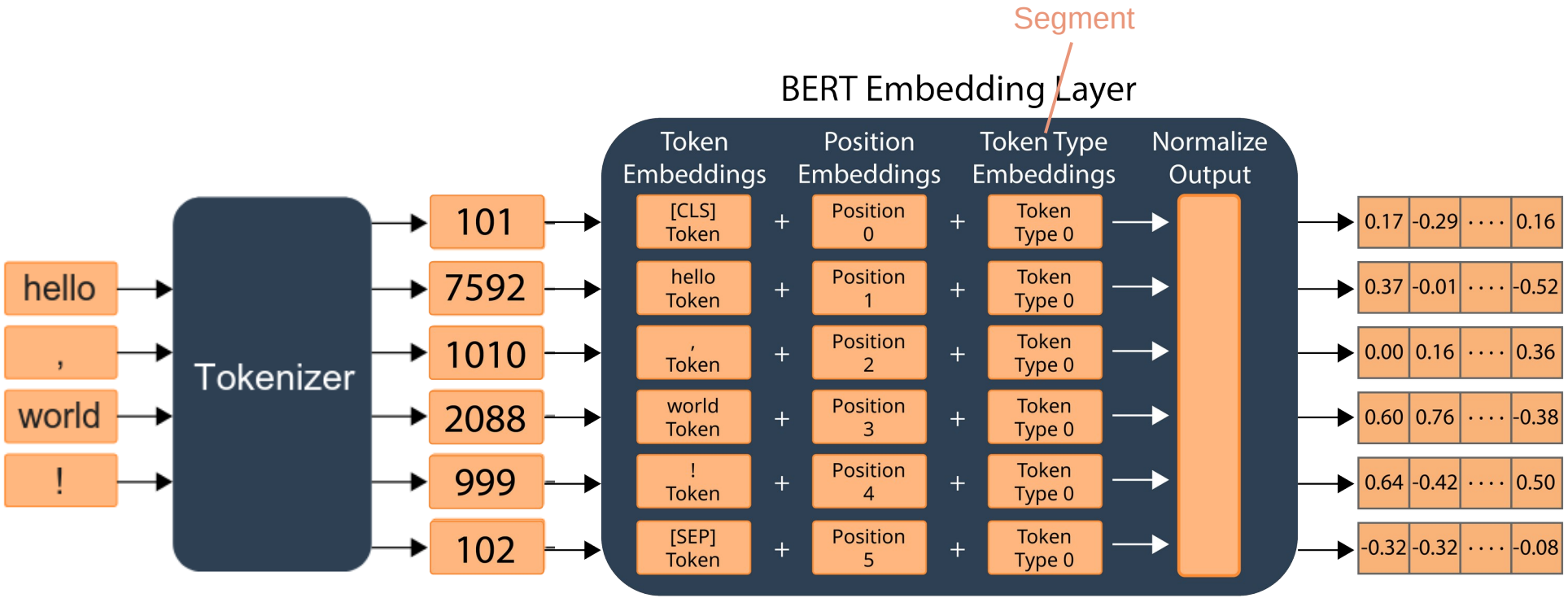
$$
\begin{aligned}
S_s : (&\alpha_{\mathrm{PC1,G_1}} \times G_{s,1}, \alpha_{\mathrm{PC1,G_2}} \times G_{s,2}, \ldots, \alpha_{\mathrm{PC1,G_g}} \times G_{s,g}, \\
&\alpha_{\mathrm{PC2,G_1}} \times G_{s,1}, \alpha_{\mathrm{PC2,G_2}} \times G_{s,2}, \ldots, \alpha_{\mathrm{PC2,G_g}} \times G_{s,g}, \\
&\ldots, \\
&\alpha_{\mathrm{PCp,G_1}} \times G_{s,1}, \alpha_{\mathrm{PCp,G_2}} \times G_{s,2}, \ldots, \alpha_{\mathrm{PCp,G_g}} \times G_{s,g}
\end{aligned}
$$

S samples
G genes
P principal components

DL: layer to learn the embedding

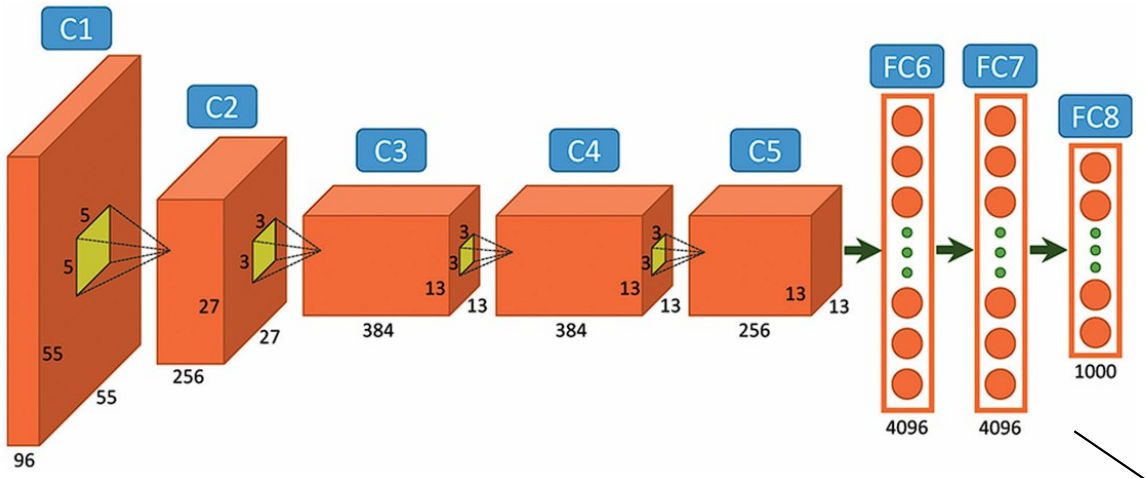


There can be several embeddings



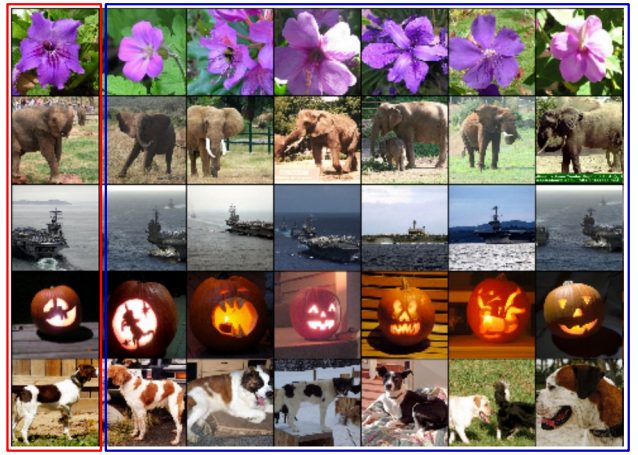
Source: https://tinkerd.net/blog/machine-learning/bert-embeddings/

Deep CNNs are embedding the images



6 nearest neighbours in the 4096 dimension space

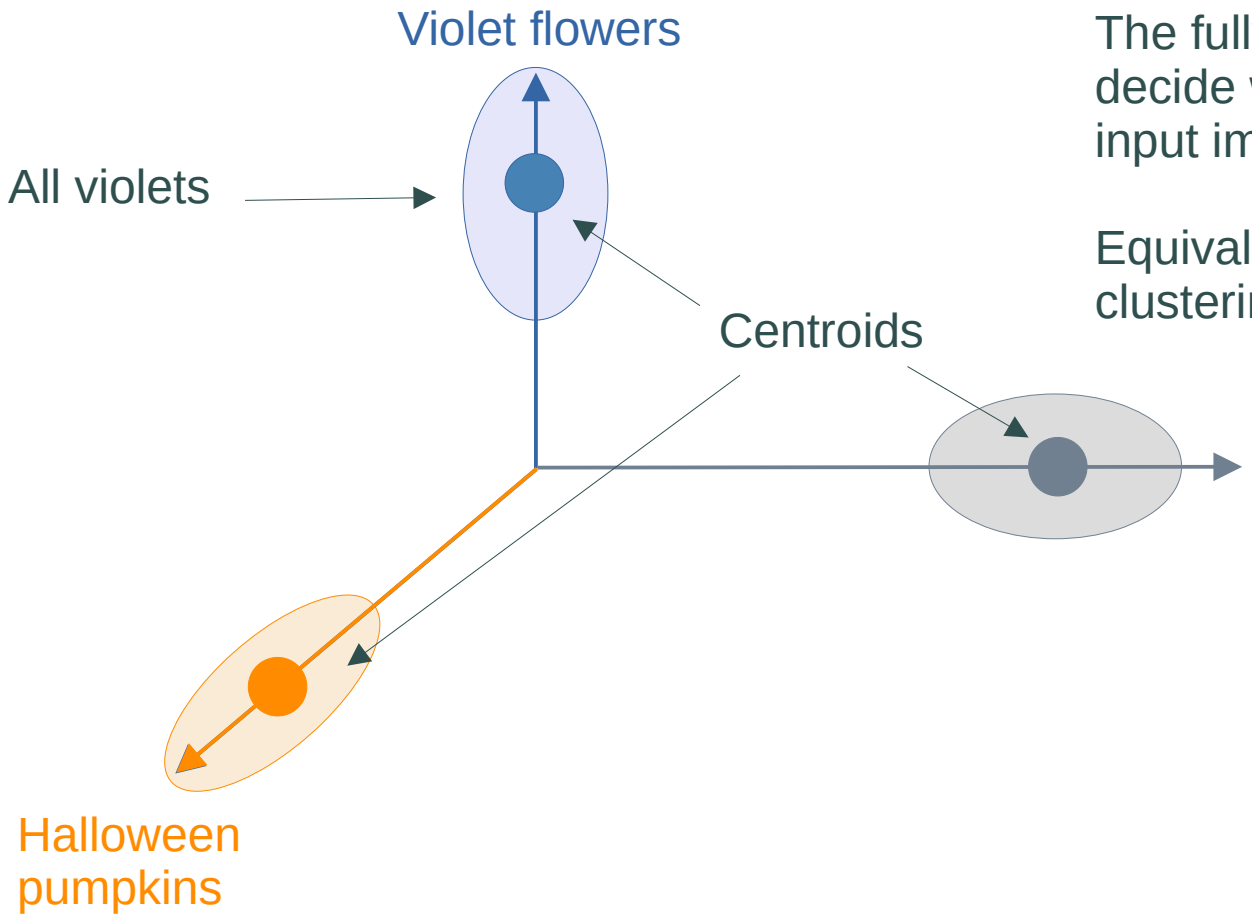
input image



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)

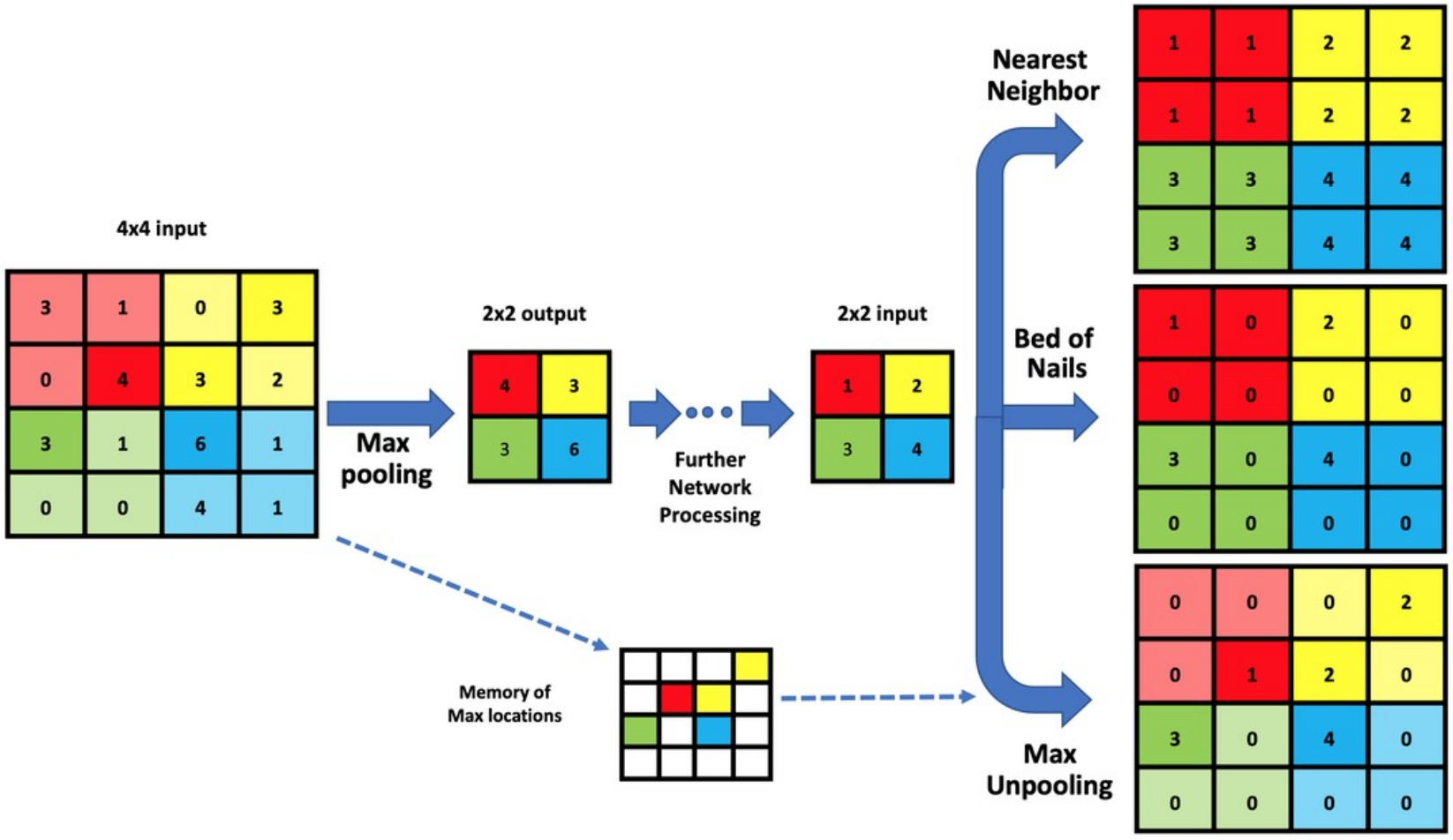
Each feature neuron points towards a cluster



The fully connected layers decide which centroids the input image is closer too.

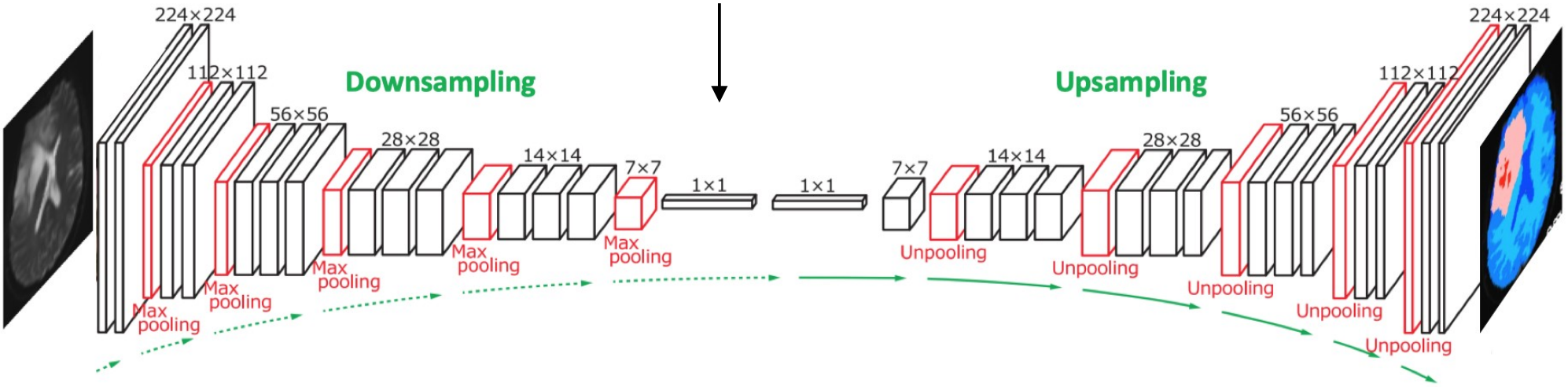
Equivalent to a PCA and a clustering (k-means, k-medoids...)

Upsampling

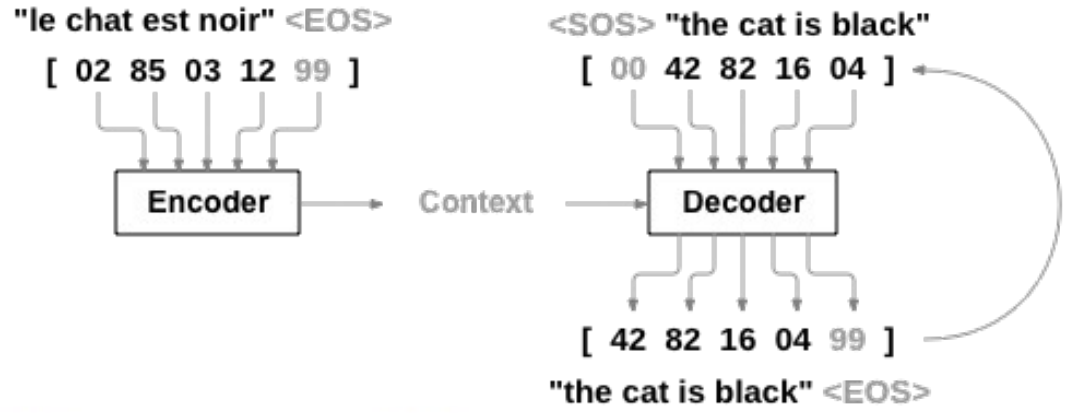

4x4 input
3 1 0 3
0 4 3 2
3 1 6 1
0 0 4 1
Max pooling
2x2 output
4 3
3 6
Further Network Processing
2x2 input
1 2
3 4
Memory of Max locations
Nearest Neighbor
1 1 2 2
1 1 2 2
3 3 4 4
3 3 4 4
Bed of Nails
1 0 2 0
0 0 0 0
3 0 4 0
0 0 0 0
Max Unpooling
0 0 0 2
0 1 2 0
3 0 4 0
0 0 0 0

Encoder-Decoder (example: tumour detection)

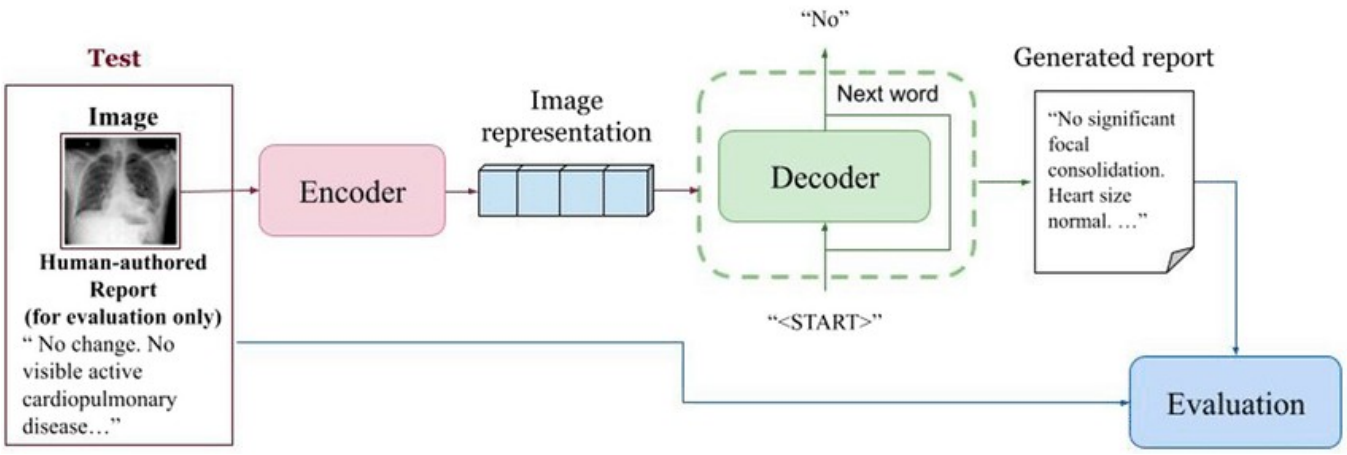
This vector (embedding in the latent space) contains all the information we need



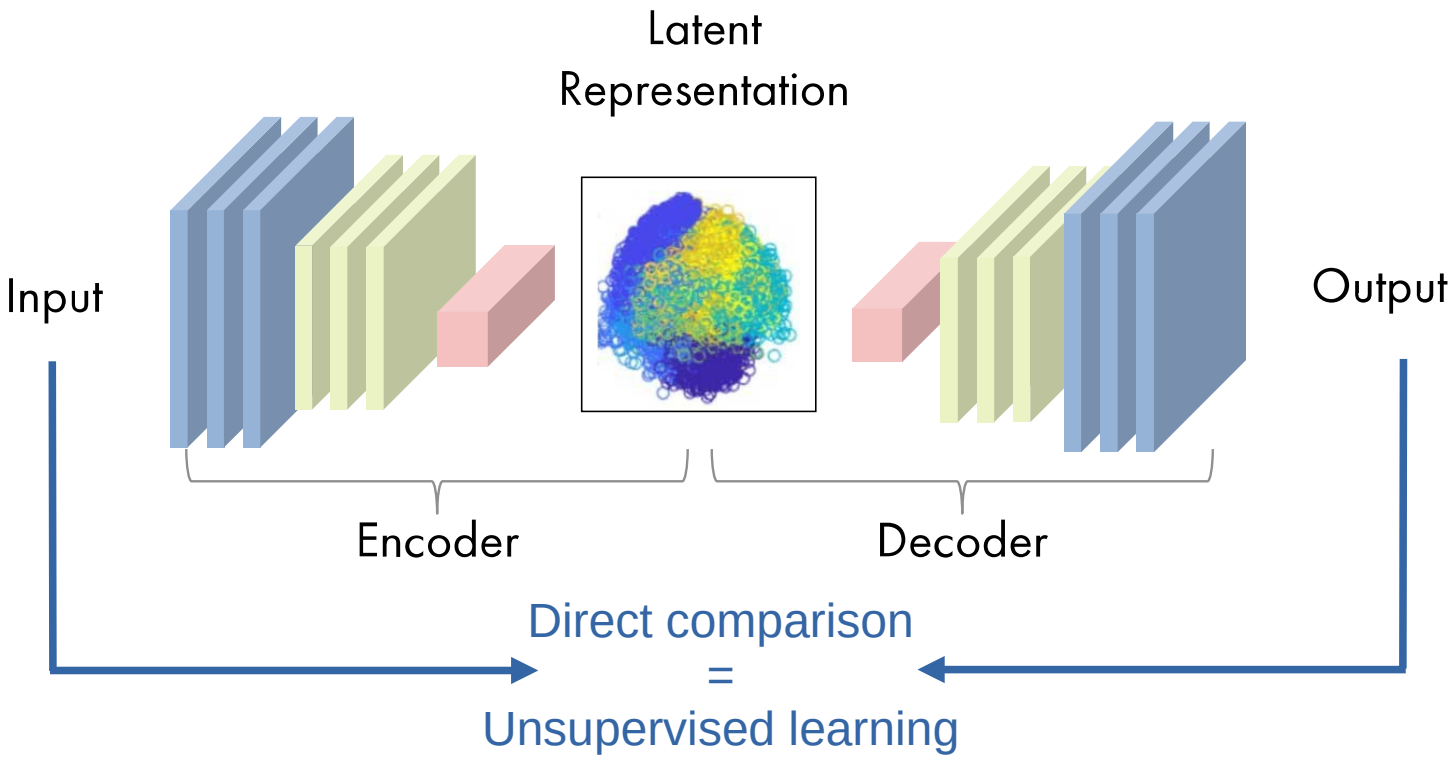
Encoders and decoders can be anything



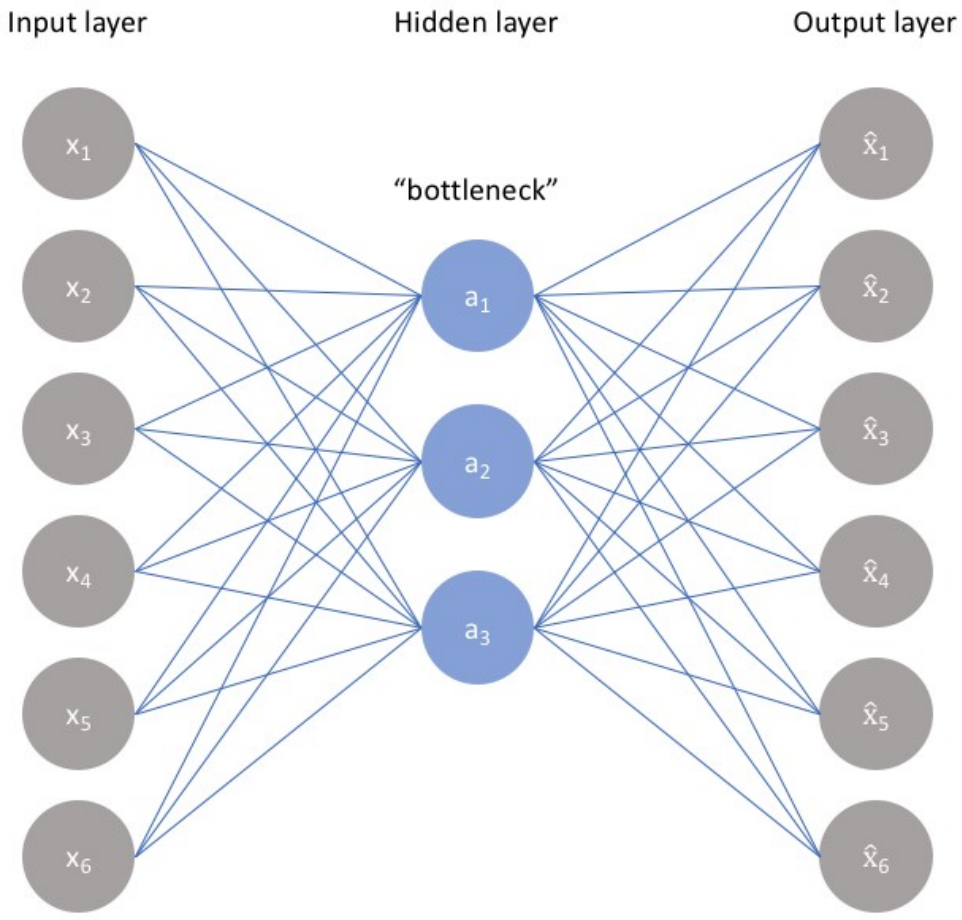
Pavlopoulos *et al* (2022)
doi:10.1007/s10115-022-01684-7



Learn by itself: AutoEncoder

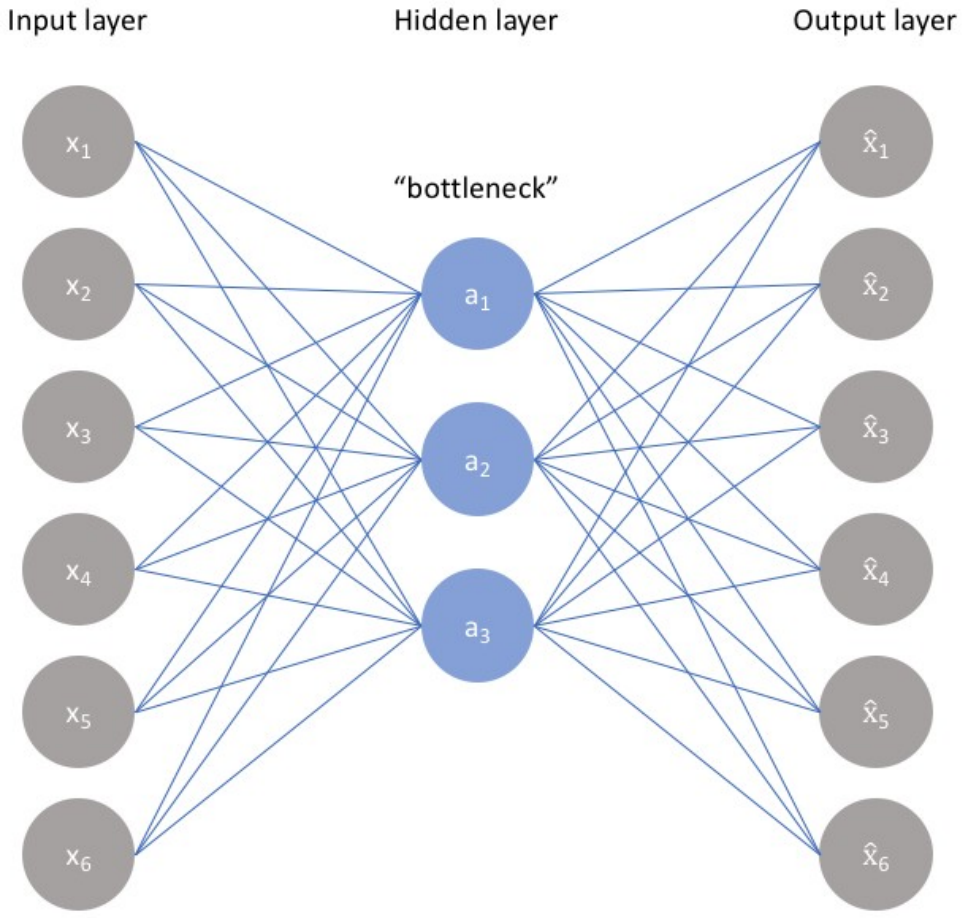


AutoEncoder ~ dimension reduction

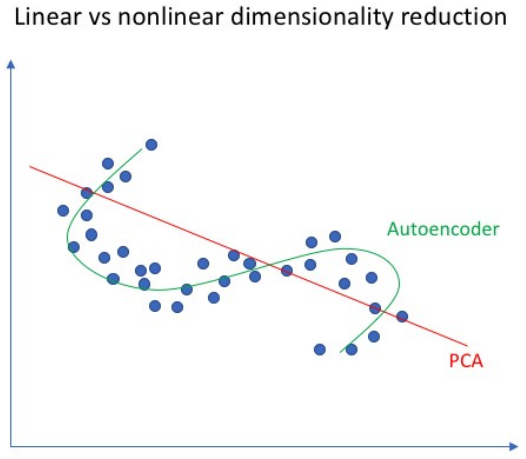


If you know the vector A,
you can reconstruct all the x
with the weights on the right.
Same weights for all Xs!
All the meaningful variability
is now contained in the A

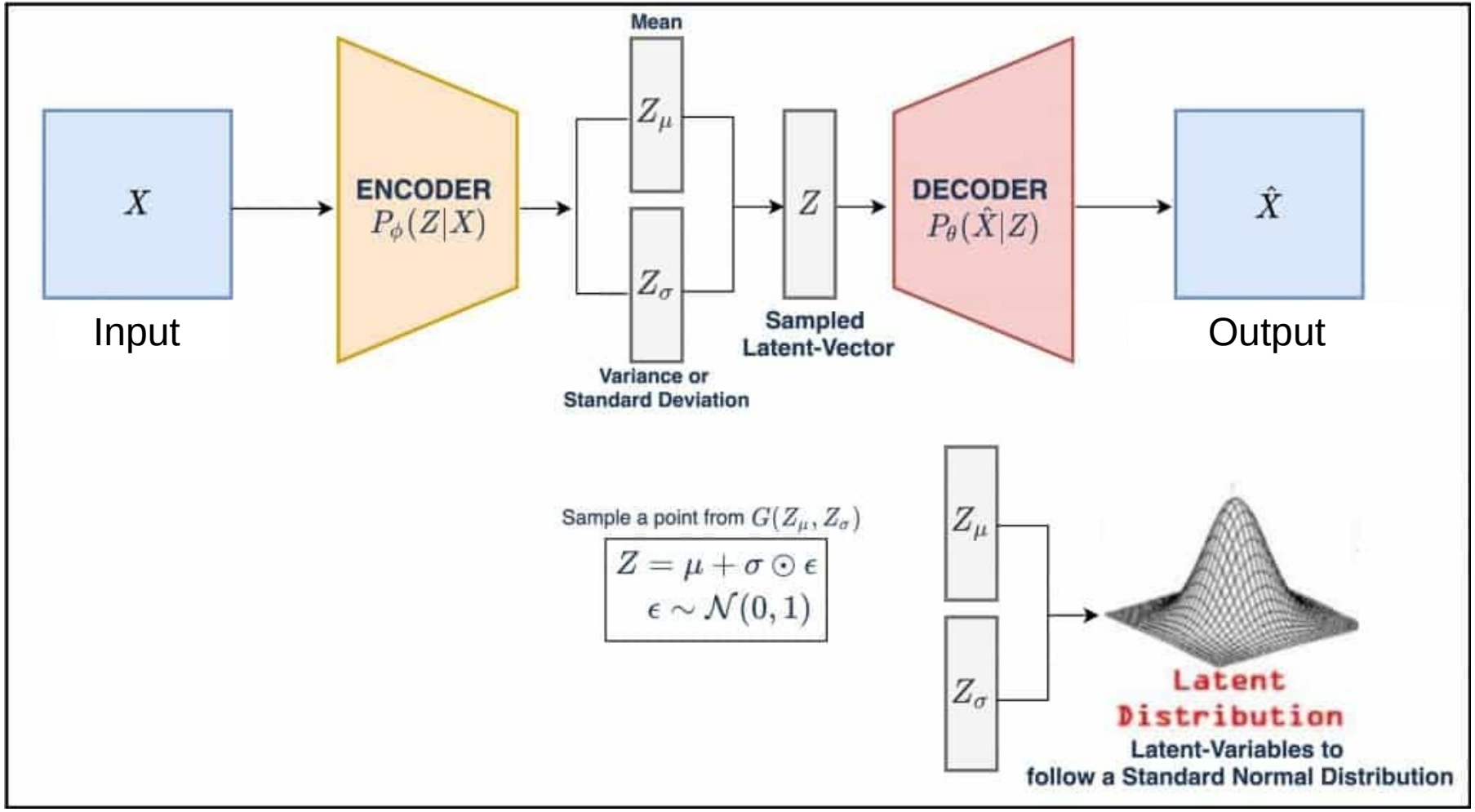
AutoEncoder ~ dimension reduction



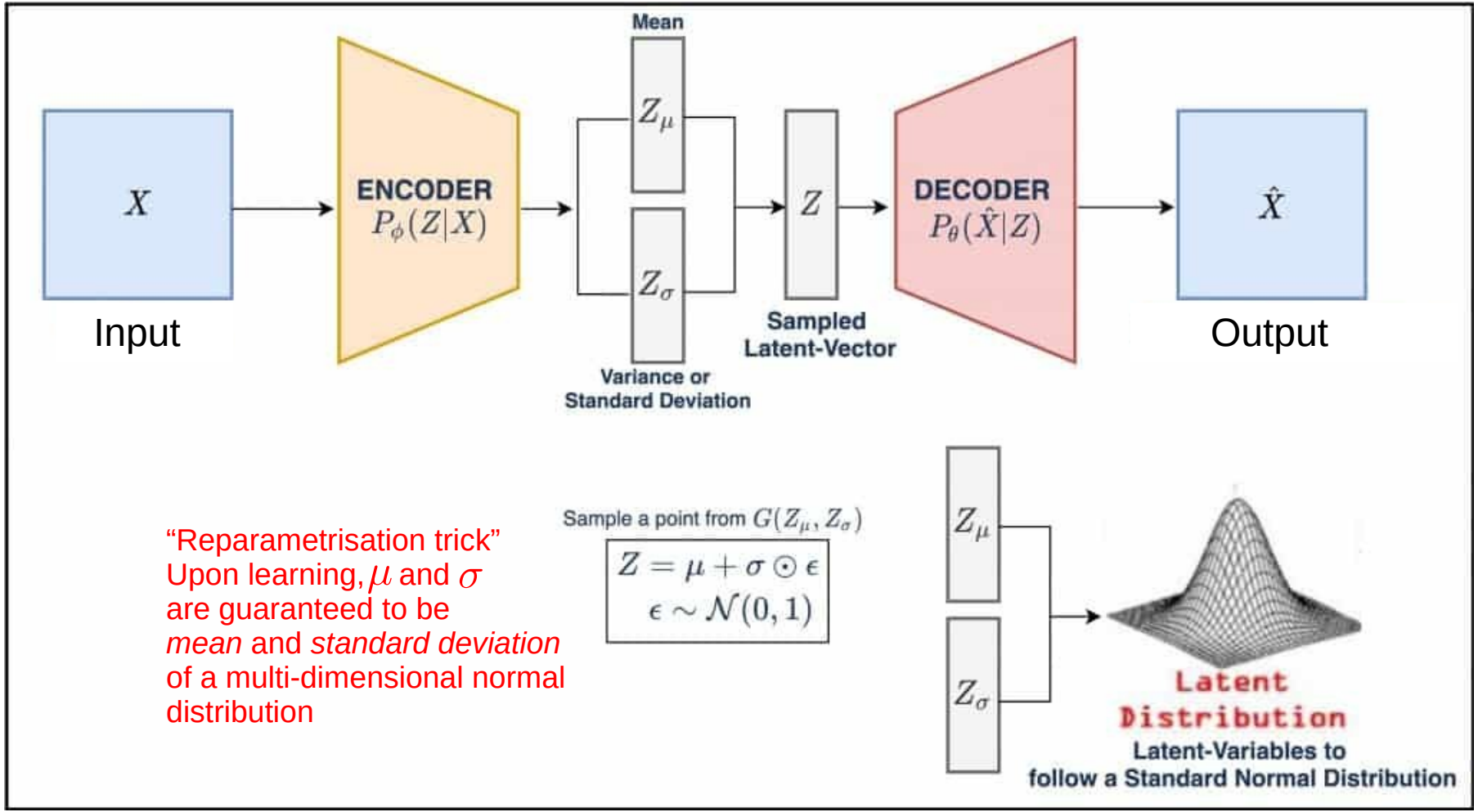
If you know the values A,
you can reconstruct all the x
with the weights on the right.
Same weights for all Xs!
All the variability is now in the As



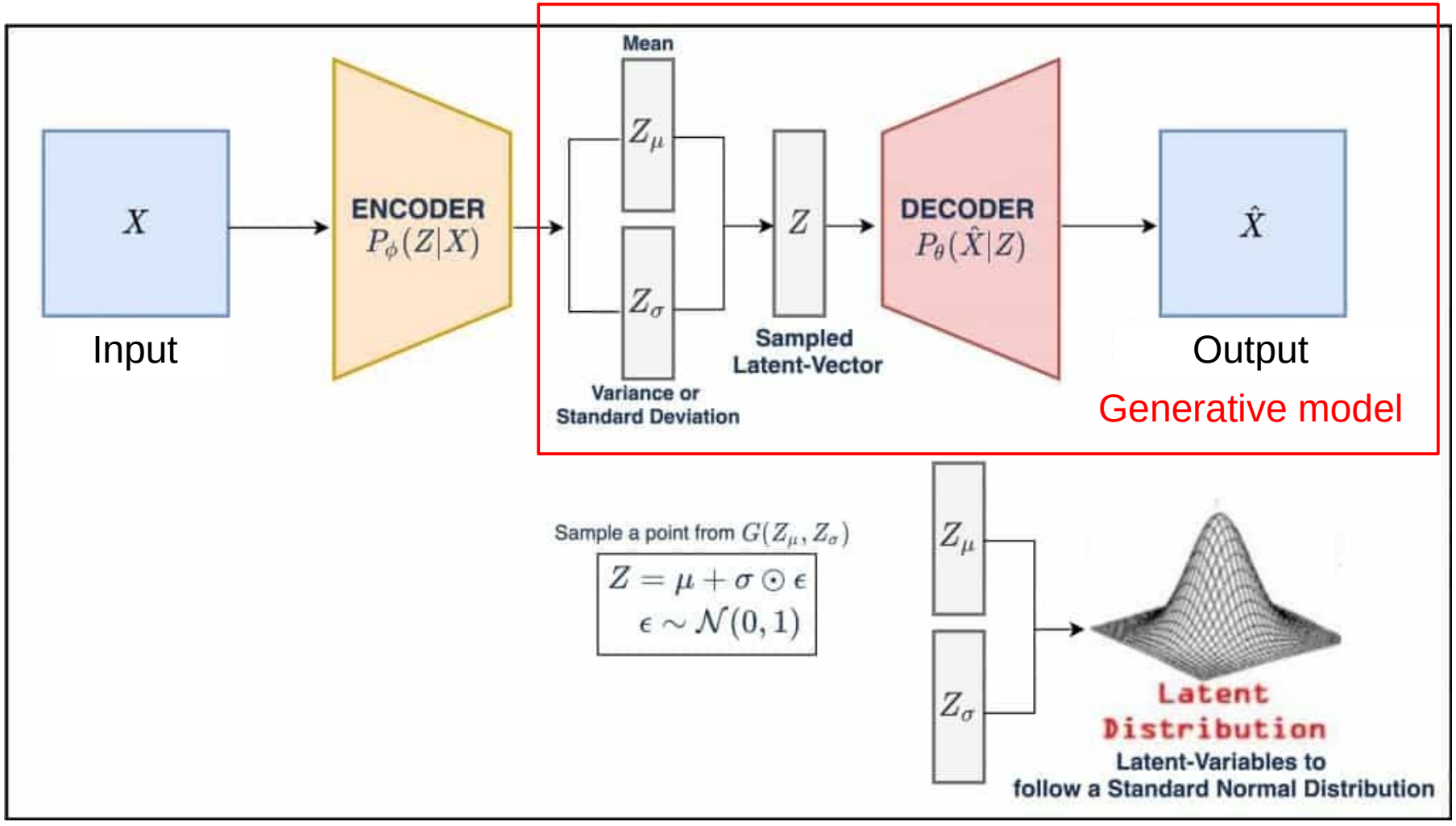
Variational Auto Encoder (VAE)



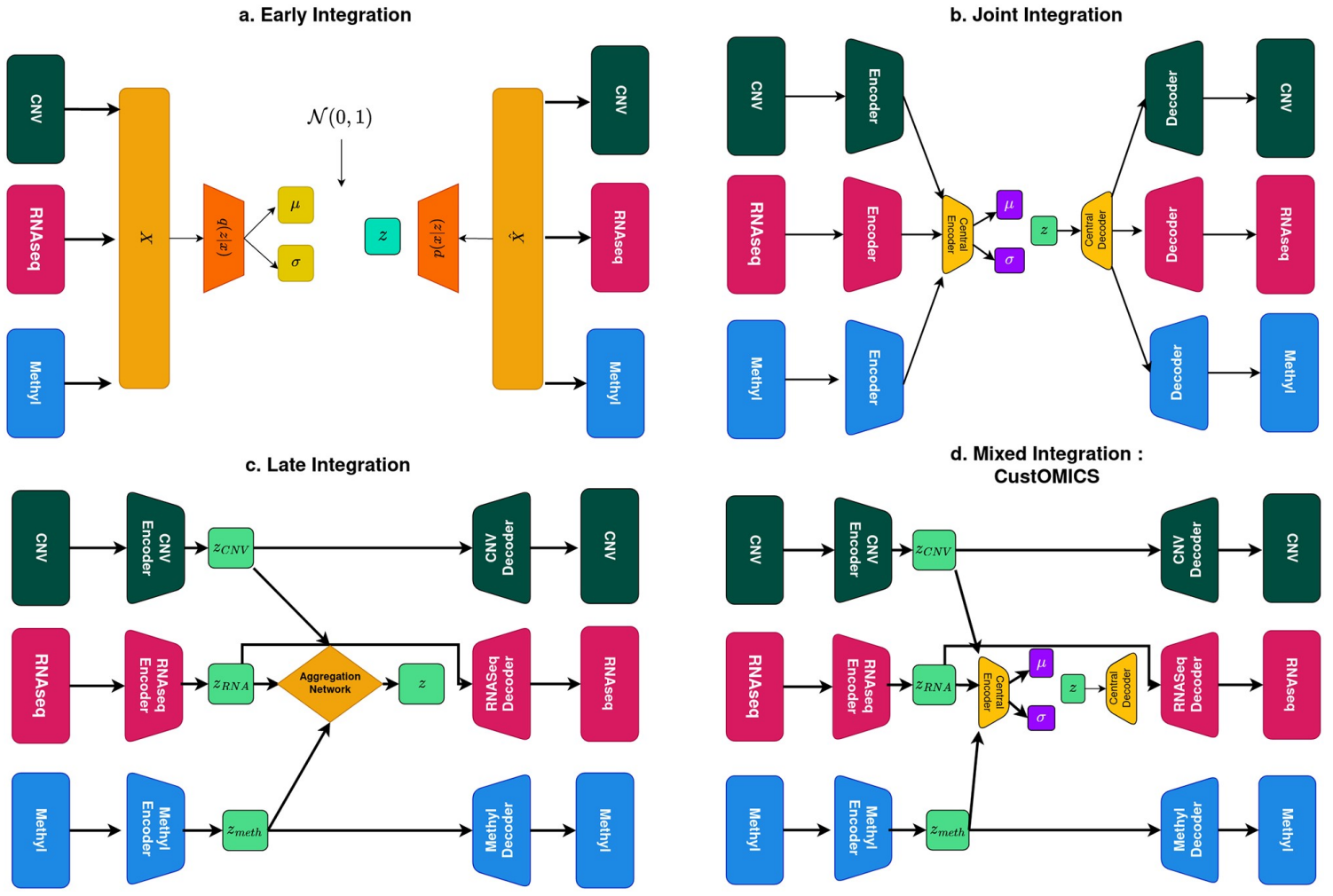
Variational Auto Encoder (VAE)



Variational Auto Encoder (VAE)



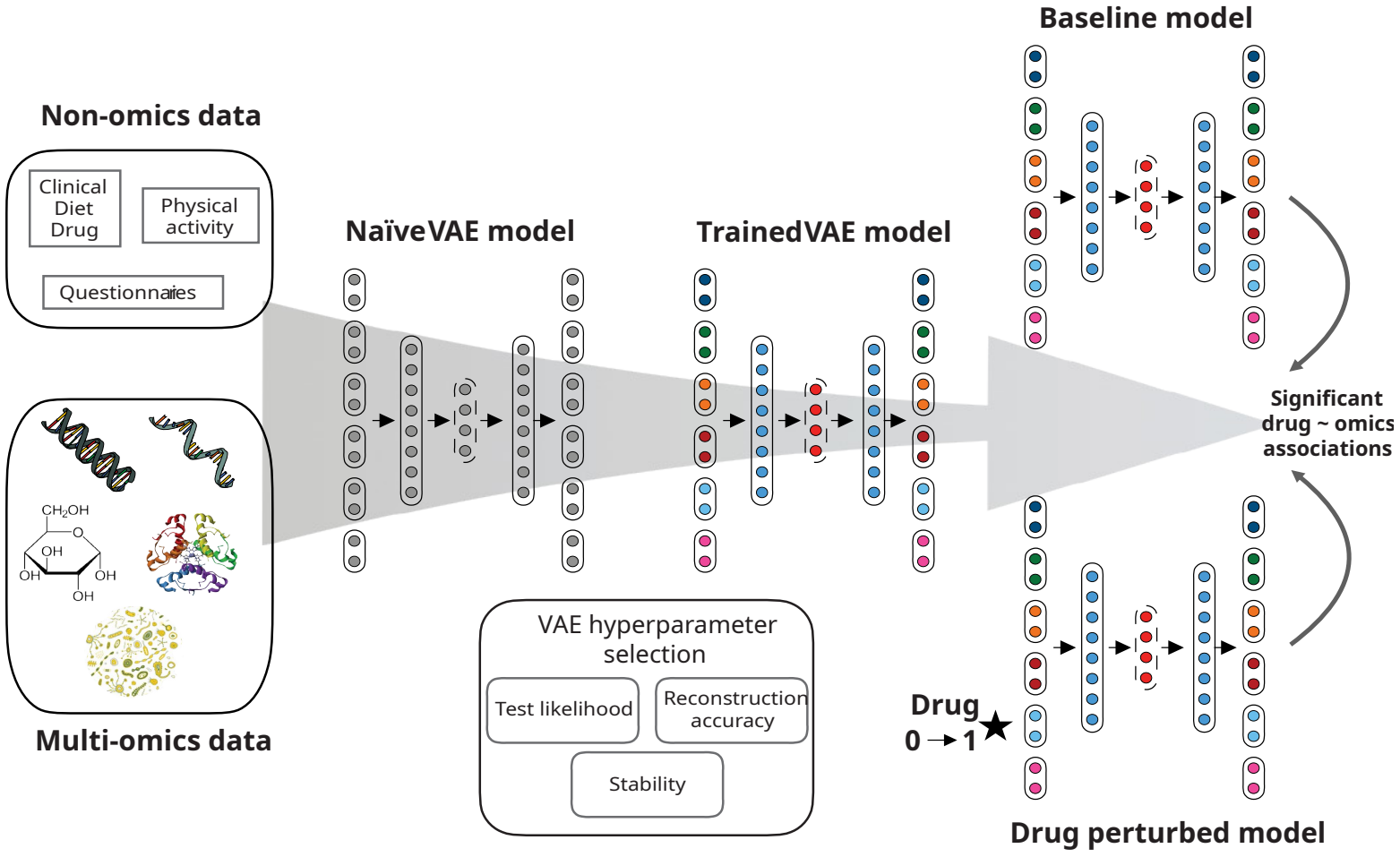
CustOMICS



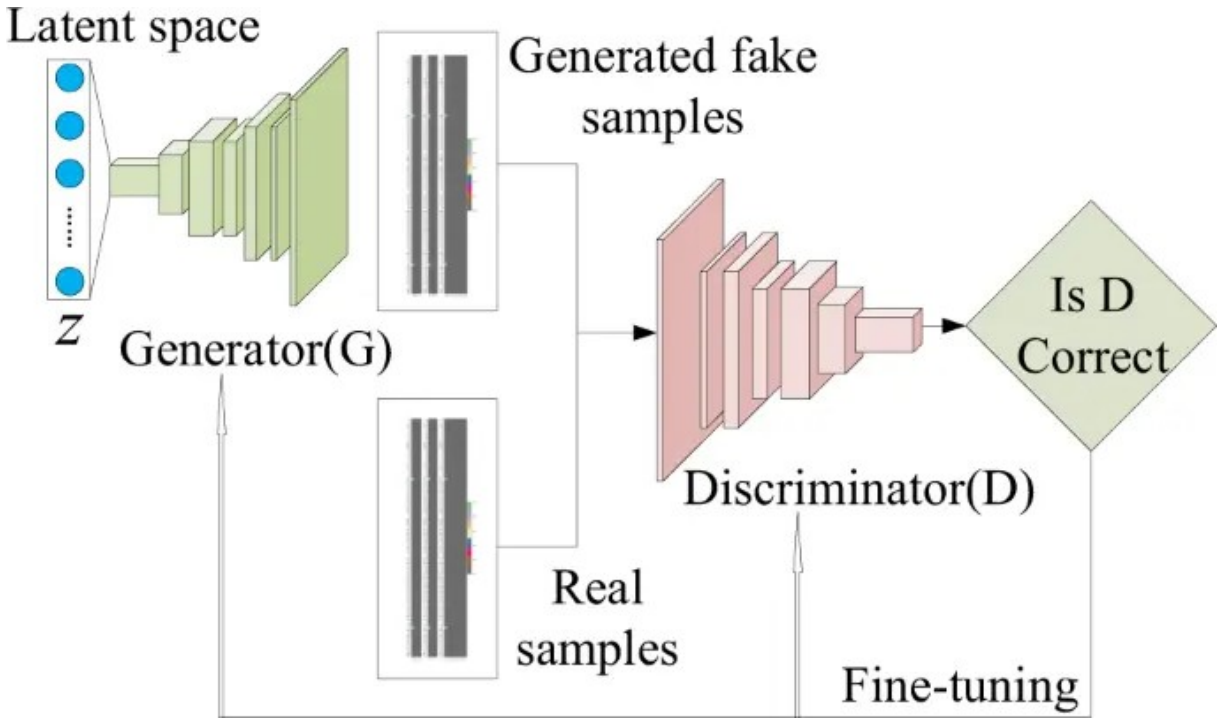
Benkirane, H., Pradat, Y., Michiels, S., Cournède, P. H. (2023).
CustOmics: A versatile deep-learning-based strategy for multi-omics integration.
PLoS Computational Biology, 19(3), e1010921.

Diabete treatments and omics

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.



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

ELSEVIER

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]

ELSEVIER

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

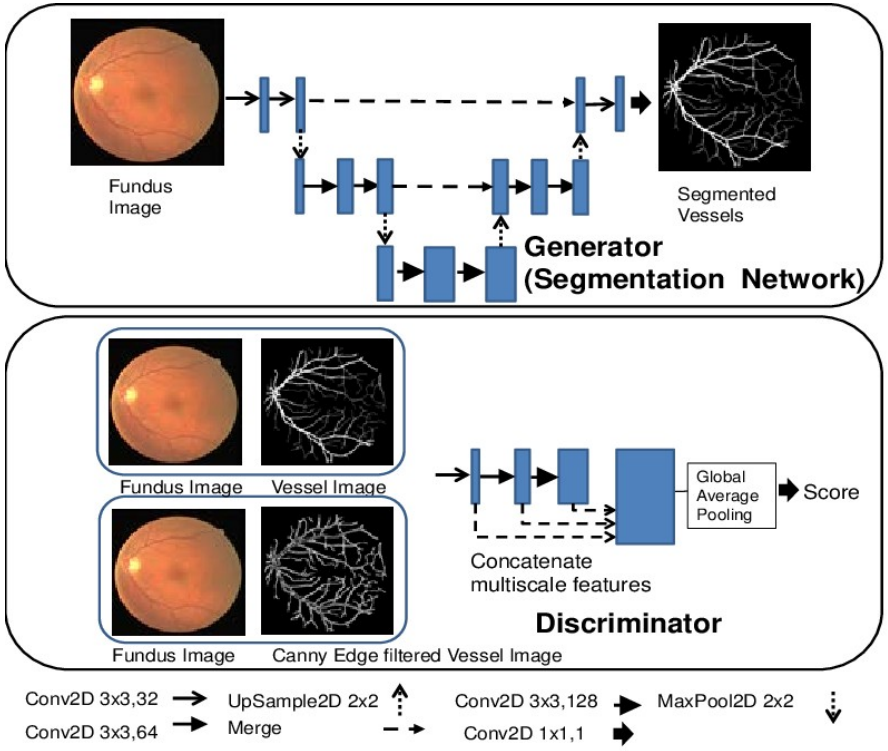
MR Image Synthesis Using Generative Adversarial Networks for Parkinson's Disease Classification

Conference paper | First Online: 02 July 2020

pp 317–327 | Cite this conference paper

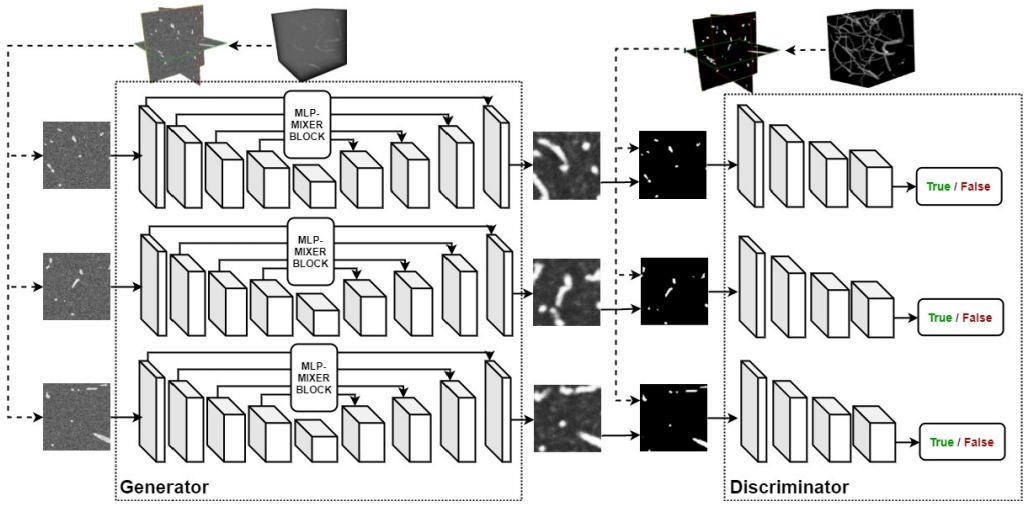
Sukhpal Kaur, Himanshu Aggarwal & Rinkle Rani

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

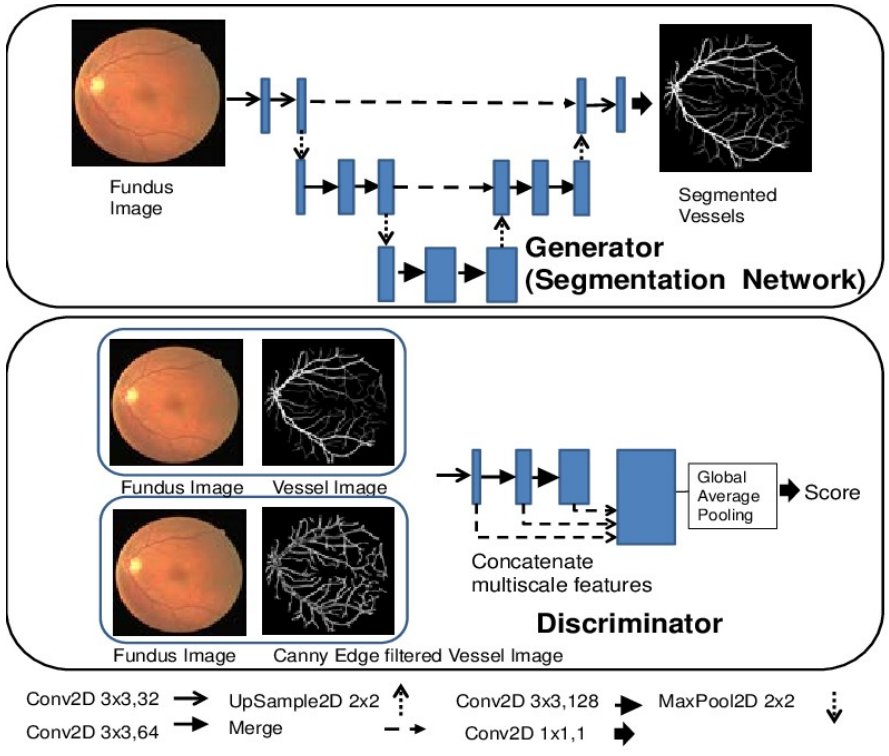
GAN can be built out of any DL architecture



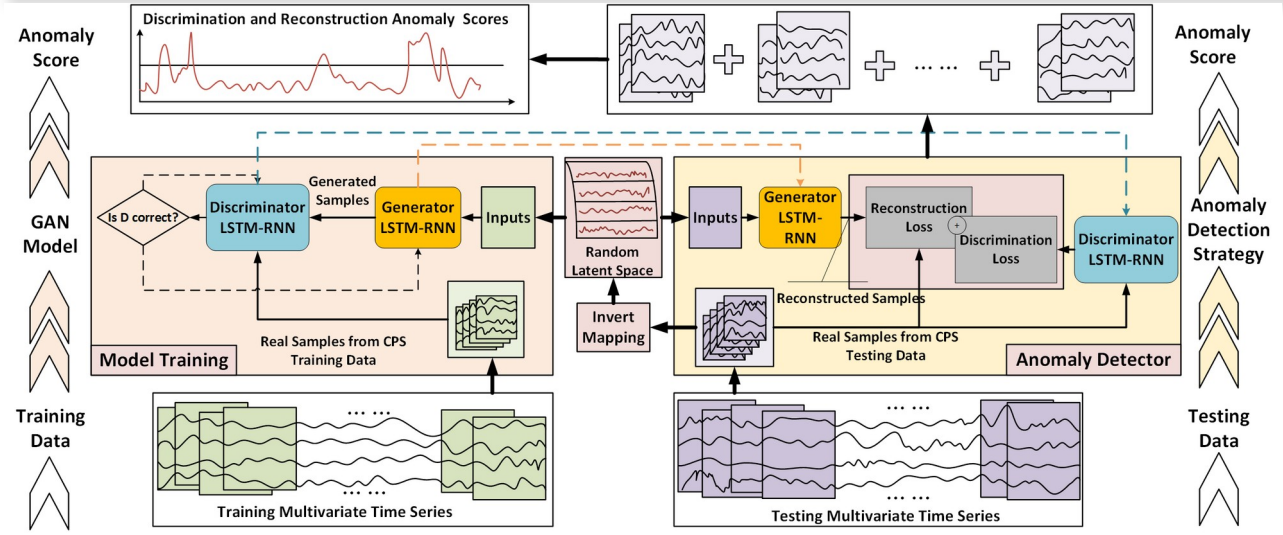
DOI: 10.1109/ICASSP49357.2023.10096997 • Corpus ID: 250626500

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

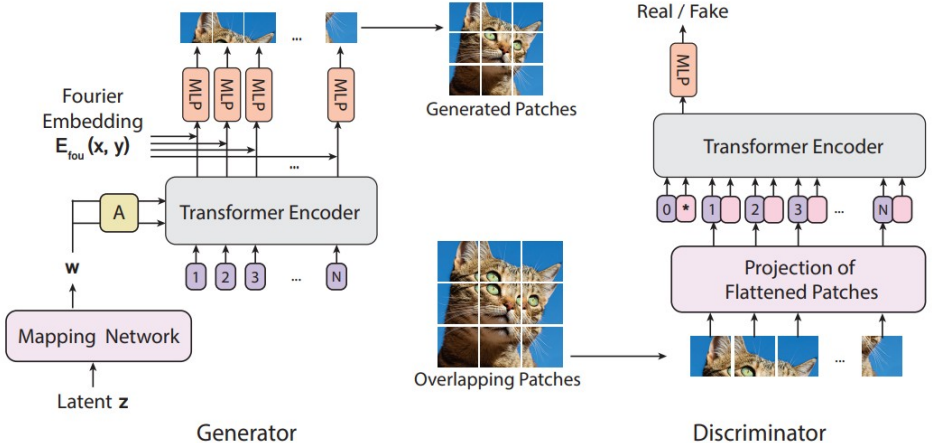


MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks

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



PreciDIAB
National Center for
Precision Diabetic Medicine

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

Région
Hauts-de-France

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