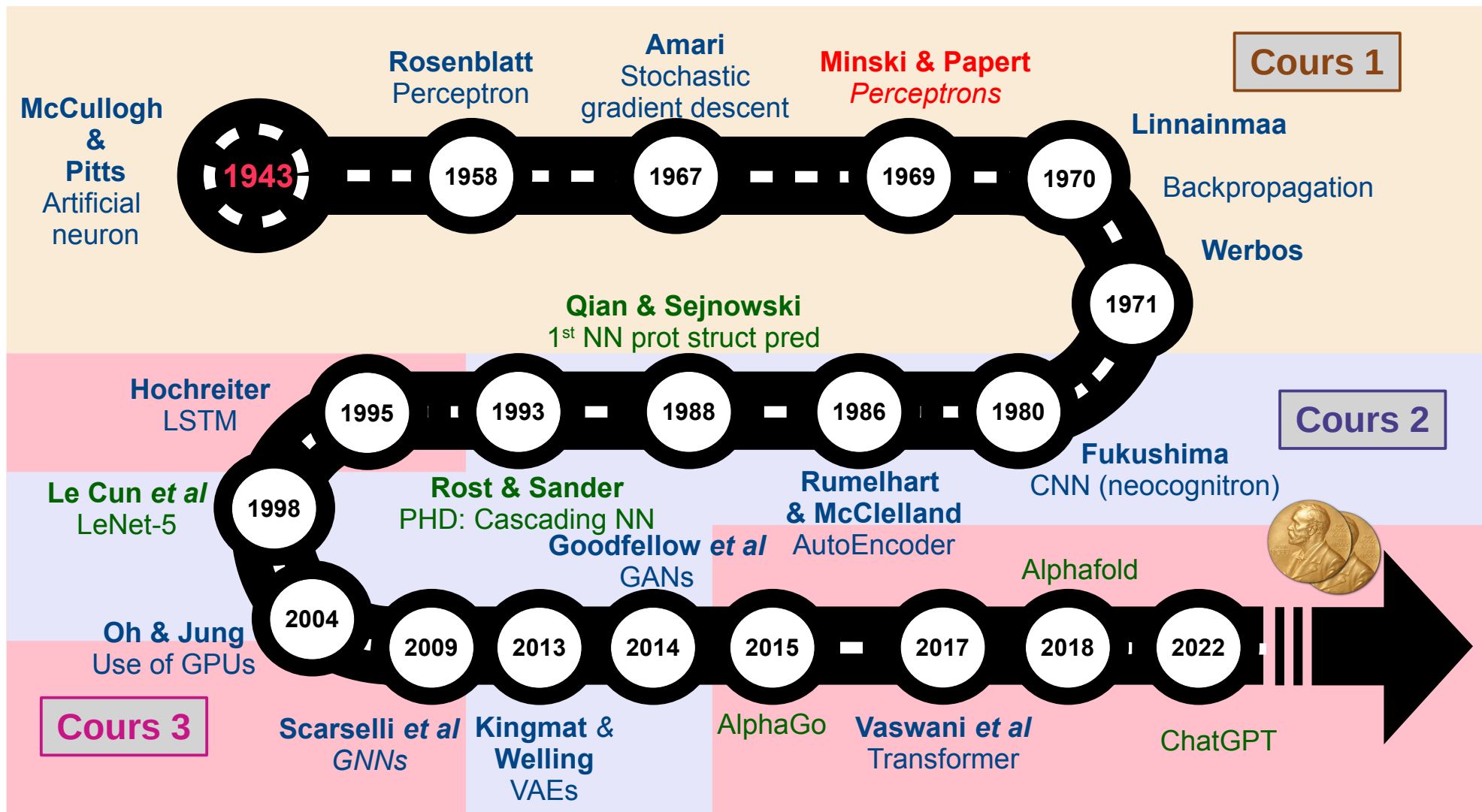


Beyond MLPs

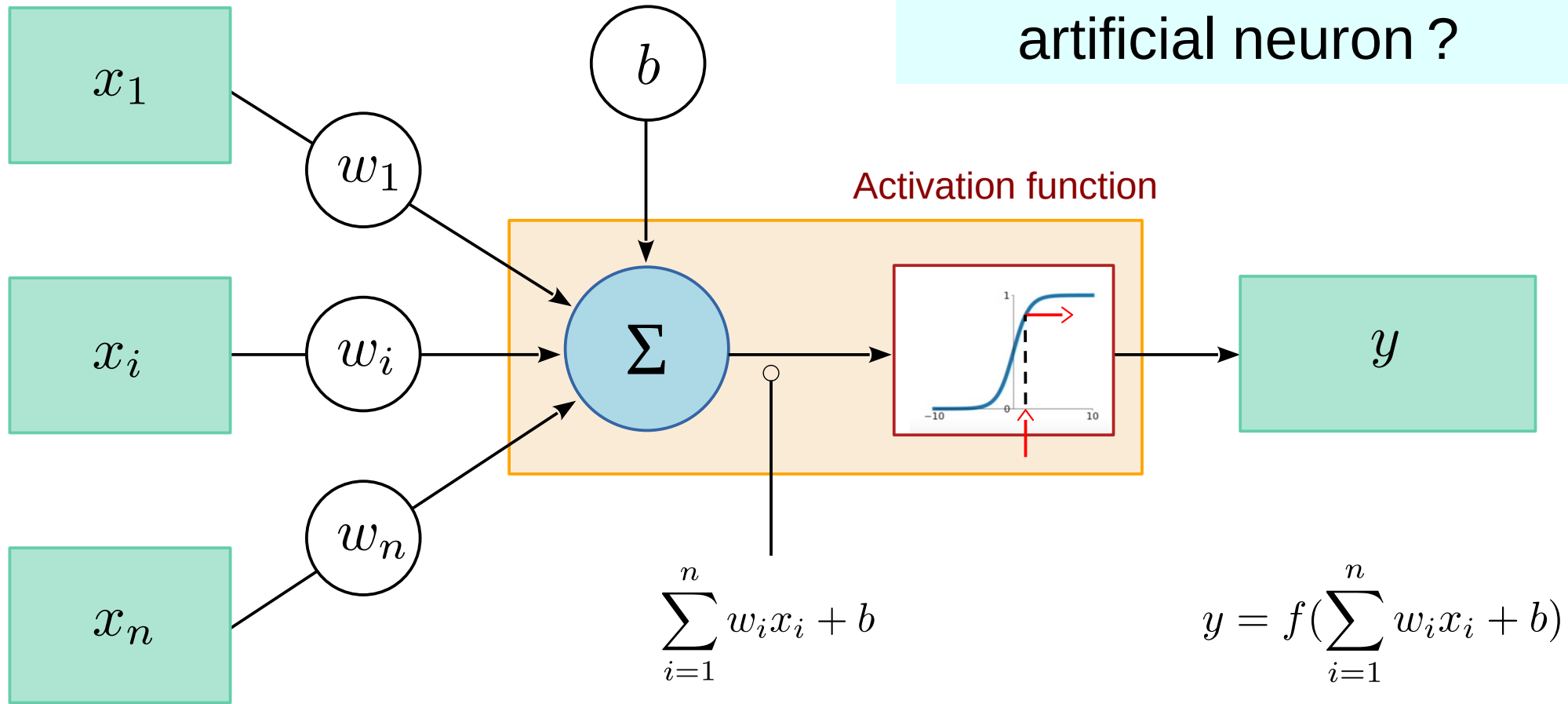
Part 1/2: CNN and AutoEncoders

Nicolas Gambardella

nicolas.gambardella@univ-lille.fr



What is an artificial neuron ?

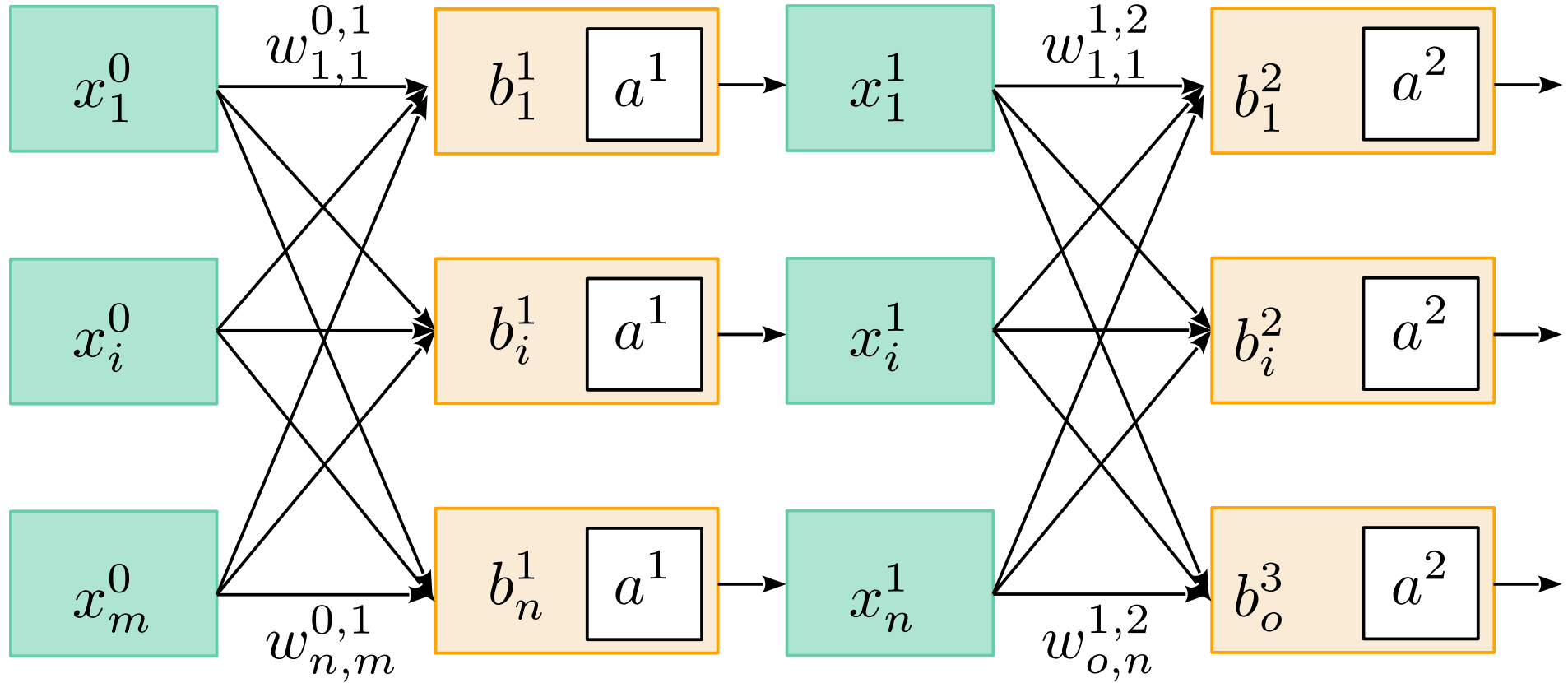


McCulloch and Pitts (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115-133

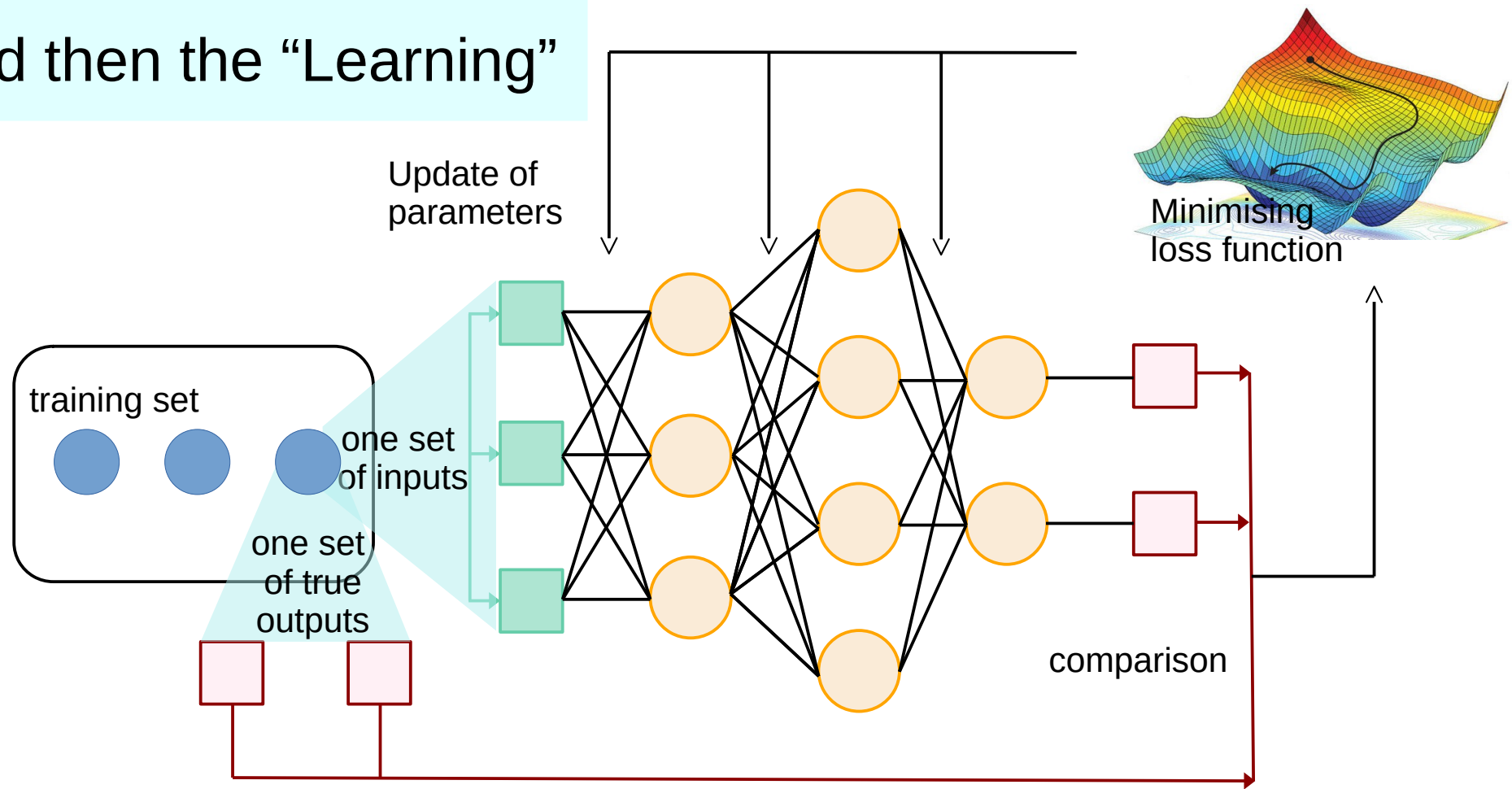
Rosenblatt (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65(6):386-408

Widrow and Hoff (1960) Adaptive Switching circuits. *WESCON Convention record* part IV: 96-104

And then we add layers (the “Deep”)



And then the “Learning”



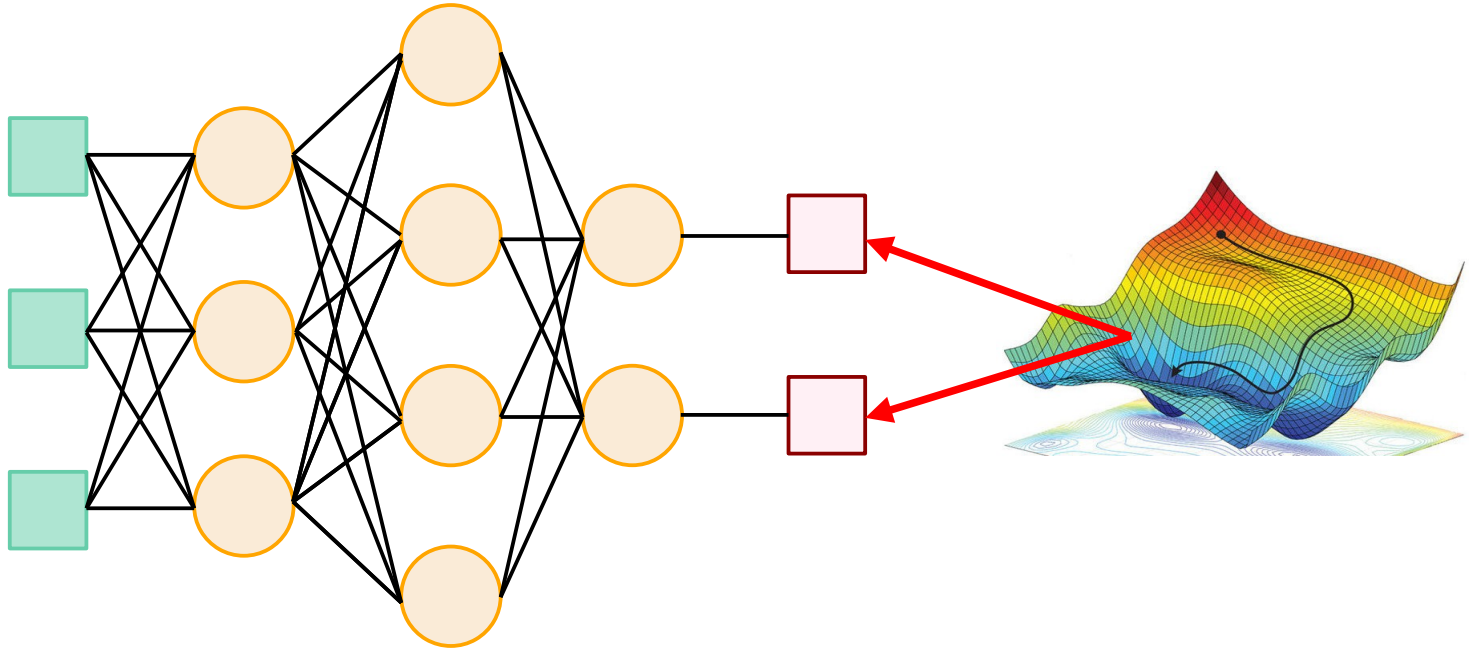
Widrow and Hoff (1960) Adaptive Switching circuits. *WESCON Convention record* part IV: 96-104

S Amari (1967). A theory of adaptive pattern classifier. *IEEE Transactions*. EC (16): 279–307

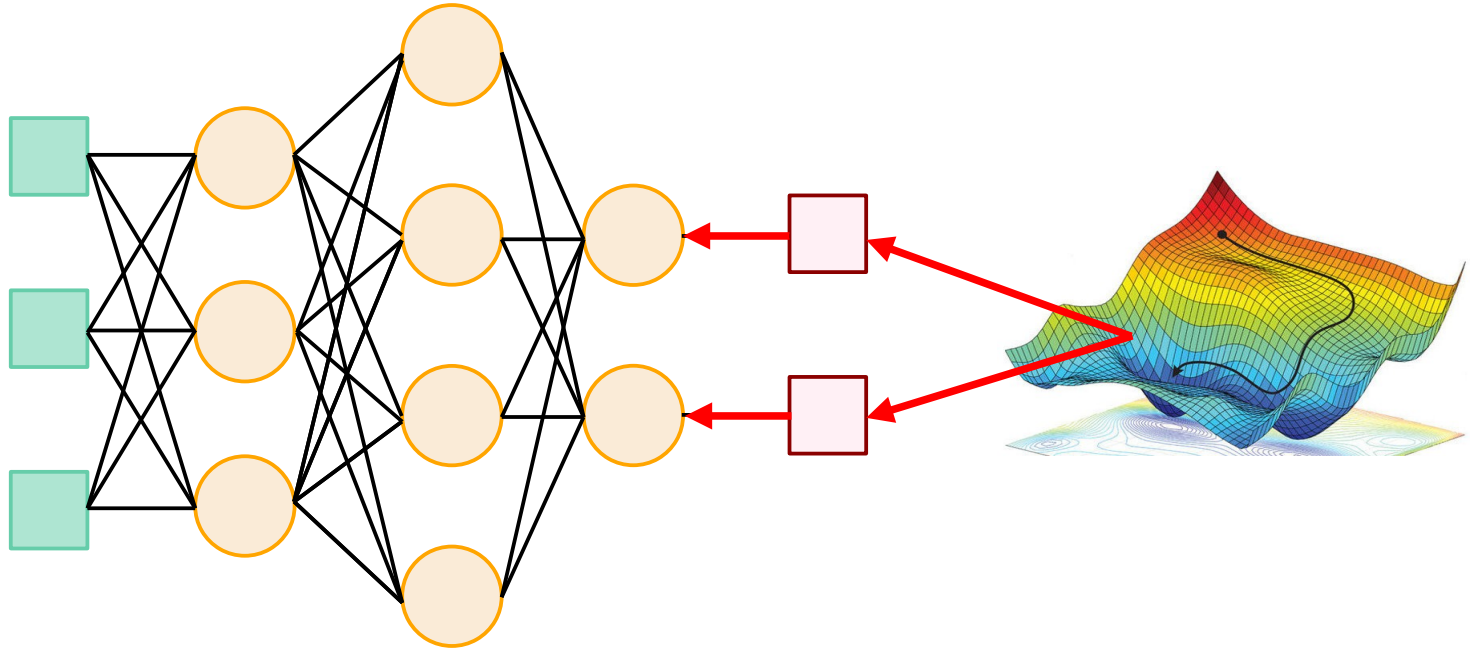
S Linnainmaa (1970-1976). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors (Masters). University of Helsinki. p. 6–7.

P Werbos (1971-1982) Applications of advances in non-linear sensitivity analysis. *LNCIS* 38: 762-770

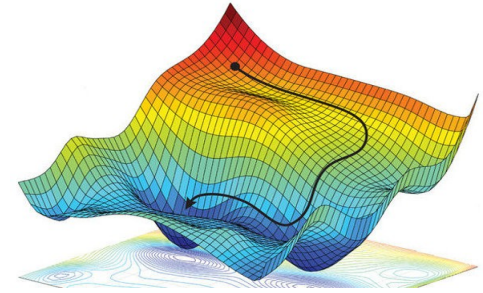
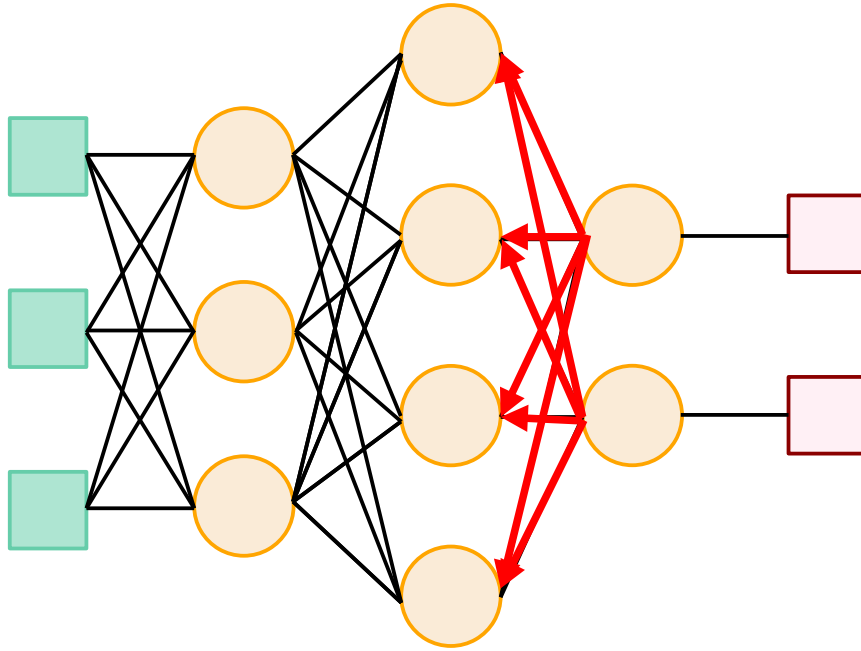
Error backpropagation one layer at a time



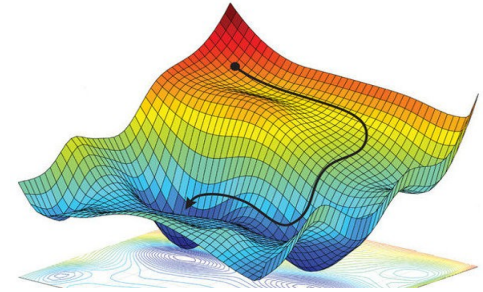
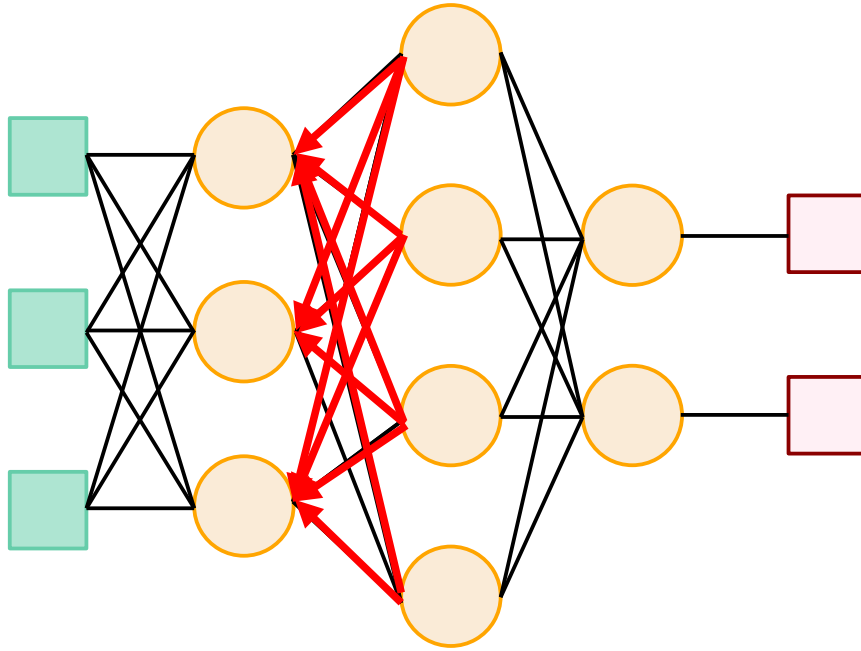
Error backpropagation one layer at a time



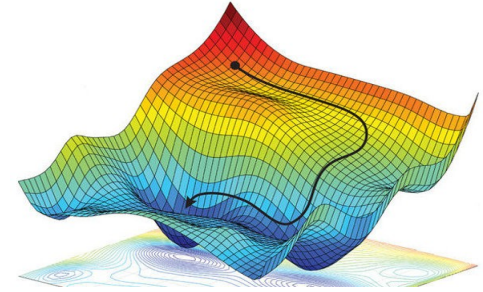
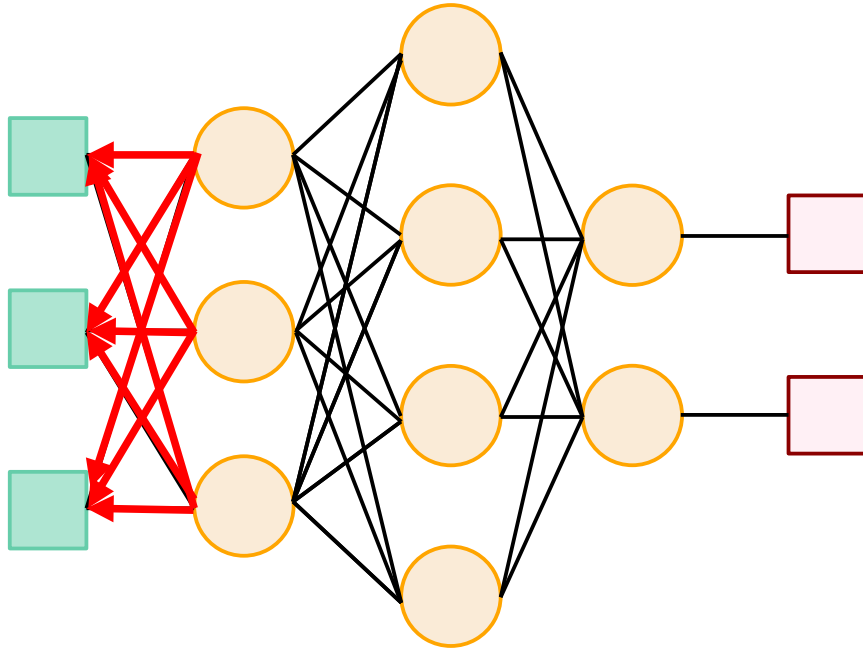
Error backpropagation one layer at a time



Error backpropagation one layer at a time



Error backpropagation one layer at a time



Output nodes (excl. EncoderDecoders)

(Linear) regression: Identity. The output 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}	$\sum_{j=1}^N e^{z_j}$	$\hat{y}_i$	$\sum_{j=1}^N \hat{y}_j$
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 that mean? And this is different if we choose 1234 instead of 0123.

Another example: maternal income classes of the Finnish NFBC86 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

$$z_j = \sum_i^N w_{ji} \cdot x_i + b_j$$

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

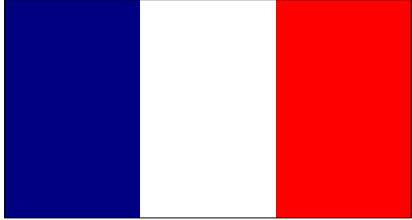
	A	G	G	T	C	C	G	A	T	T	A	G	A	C	T	C
A	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
G	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0
C	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1

(if not A, not G, not C...)

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

How about analysing images?

FR

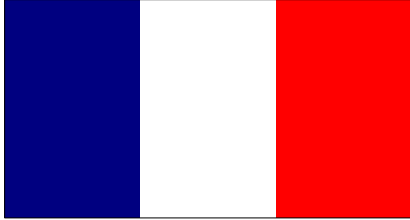


DE



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

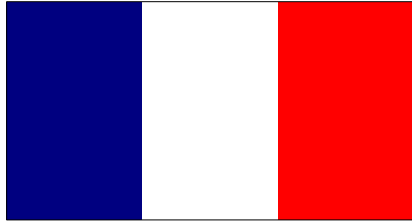
B

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

G

0	255	0
0	255	0
0	255	0

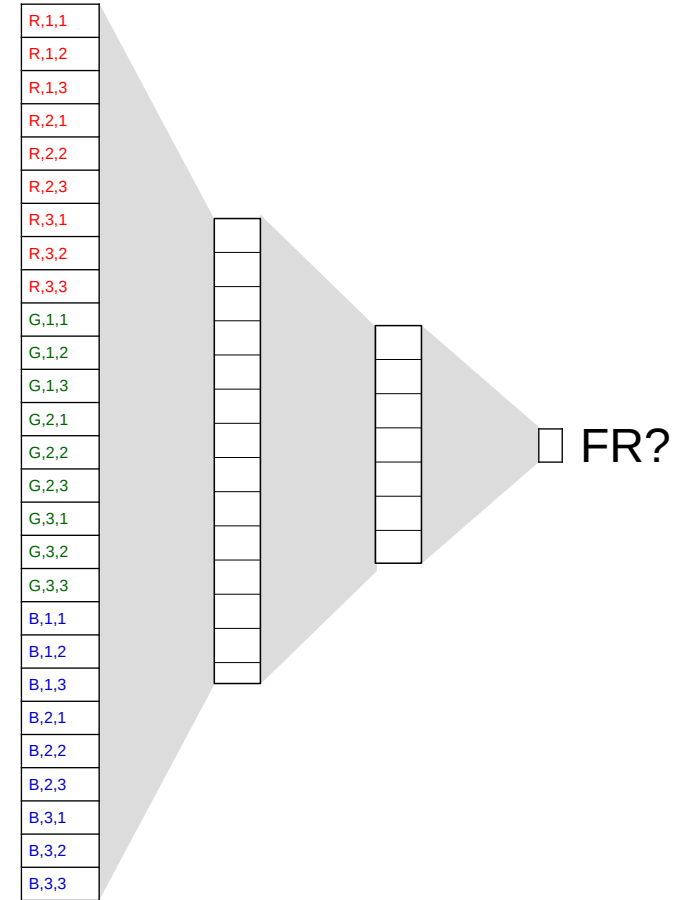
B

128	255	0
128	255	0
128	255	0

0	0	0
255	255	255
255	255	255

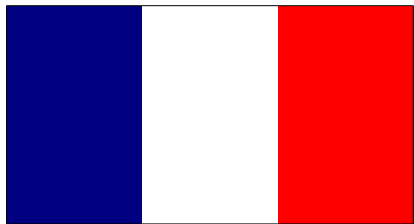
0	0	0
0	0	0
255	255	255

0	0	0
0	0	0
0	0	0



Problem: parameter explosion

FR



DE



R

0	255	255
0	255	255
0	255	255

G

0	255	0
0	255	0
0	255	0

B

128	255	0
128	255	0
128	255	0

0	0	0
255	255	255
255	255	255

0	0	0
0	0	0
255	255	255

0	0	0
0	0	0
0	0	0

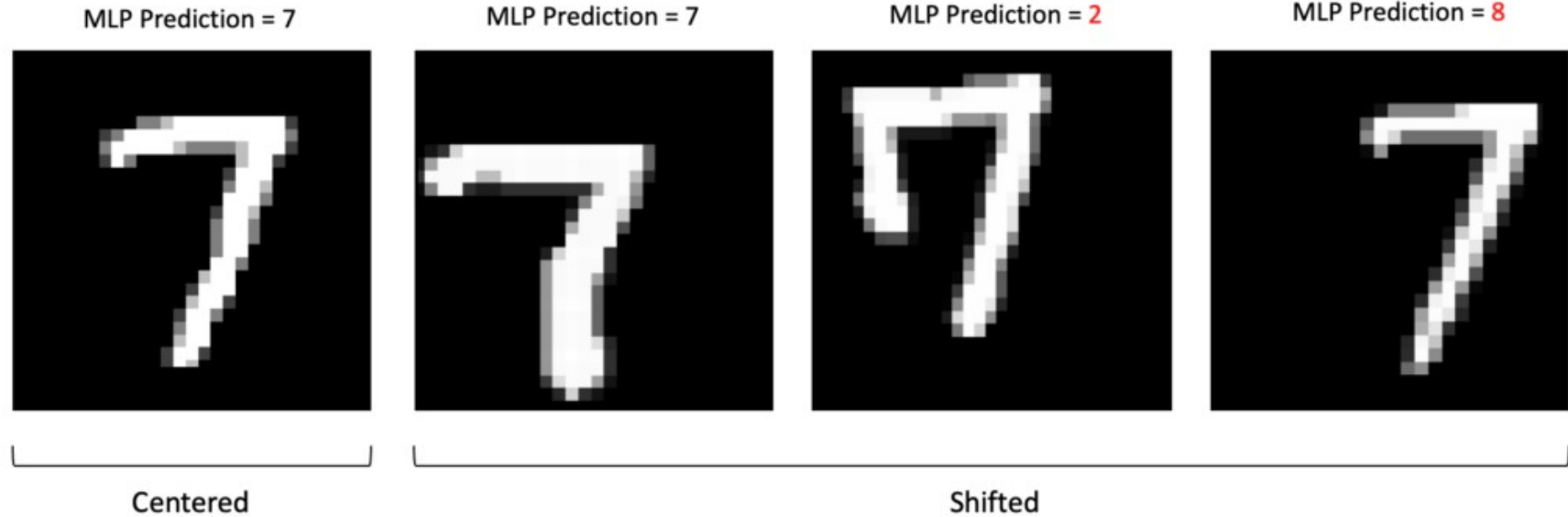
R,1,1
R,1,2
R,1,3
R,2,1
R,2,2
R,2,3
R,3,1
R,3,2
R,3,3
G,1,1
G,1,2
G,1,3
G,2,1
G,2,2
G,2,3
G,3,1
G,3,2
G,3,3
B,1,1
B,1,2
B,1,3
B,2,1
B,2,2
B,2,3
B,3,1
B,3,2
B,3,3

$9 \times 3 \times 14 + 14 \times 7 + 7 = 483$ weights for an image of 9 pixels!

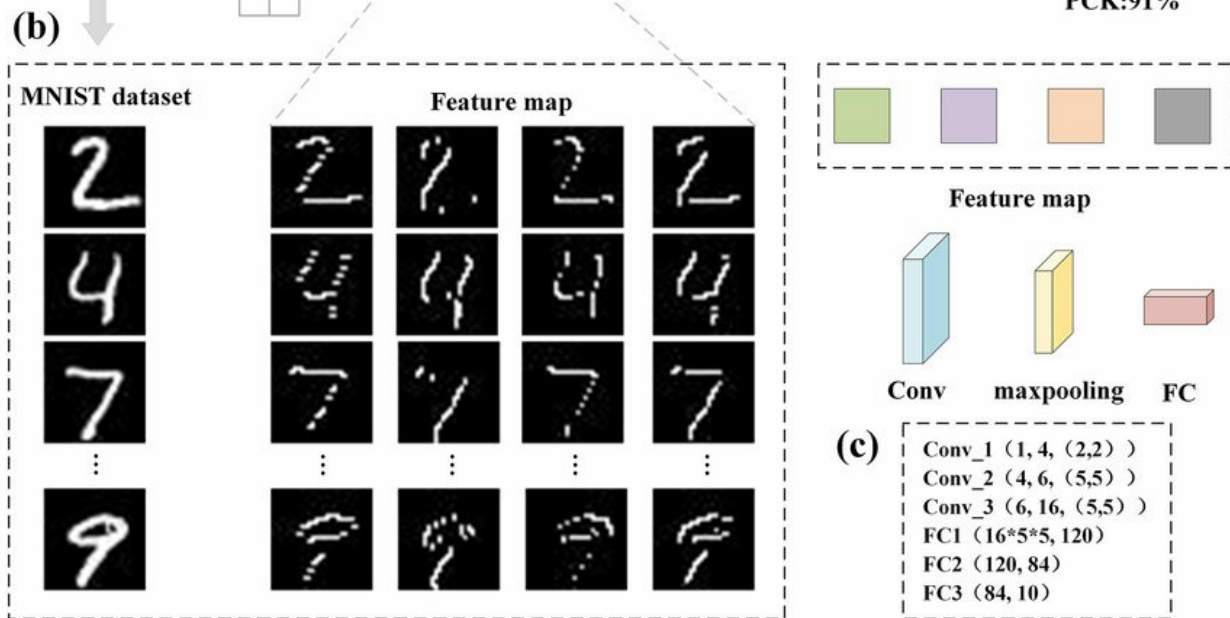
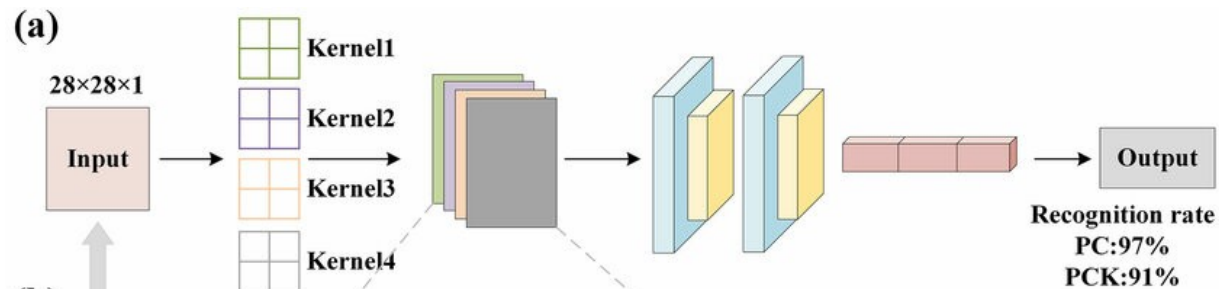
Most of them are useless since most pixels are identical. We want to recognise vertical vs horizontal boundaries, and 3 colours

FR?

Problem: lack of translation invariance

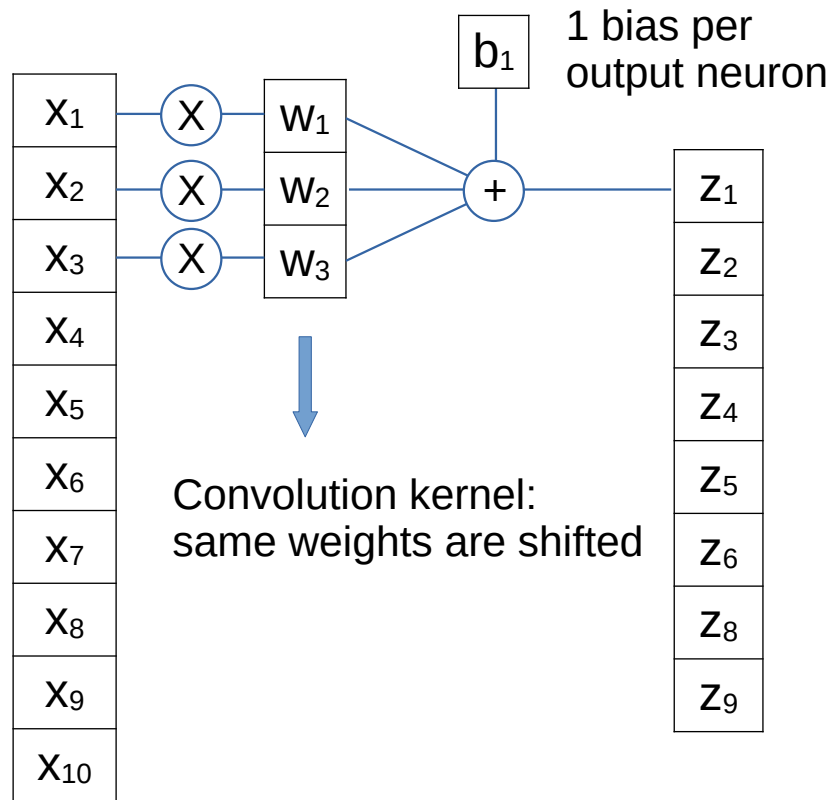


Convolutional Neural Networks (CNN)



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

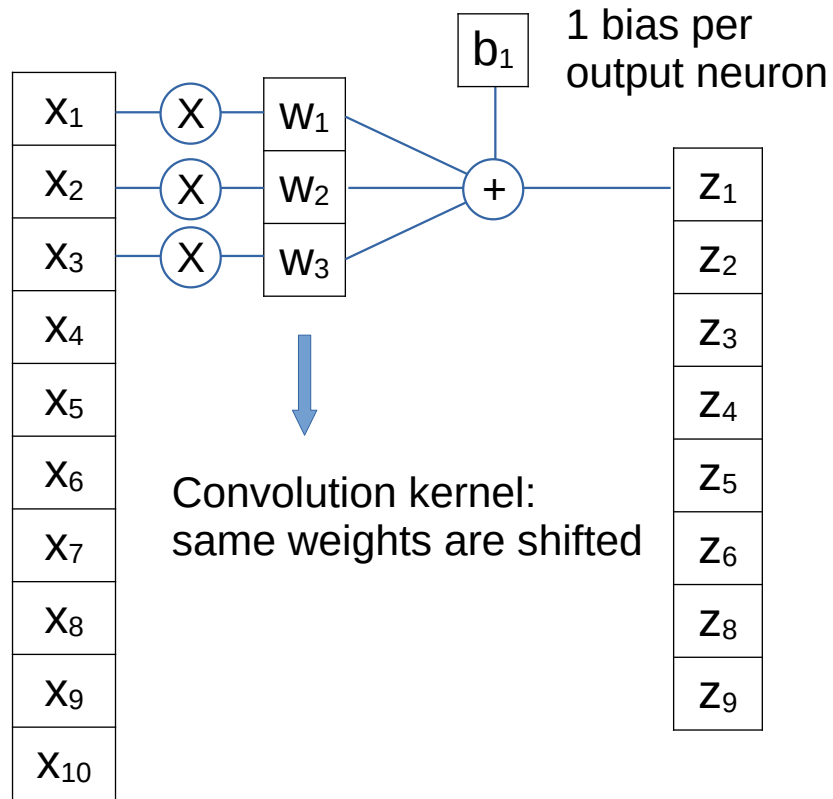
Convolutional neural networks: local correlations between inputs



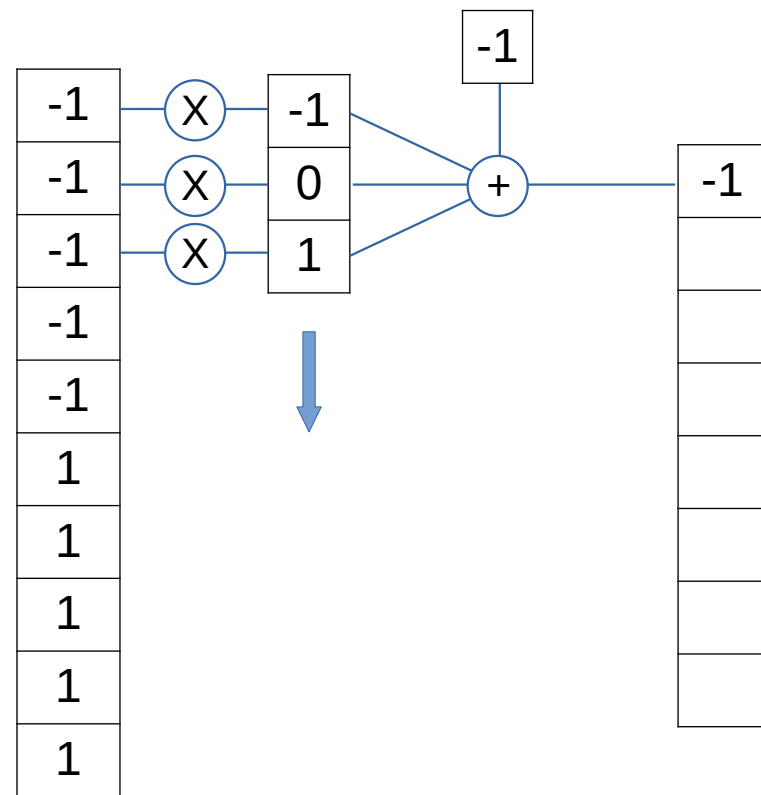
$$z_1 = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 + b$$

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

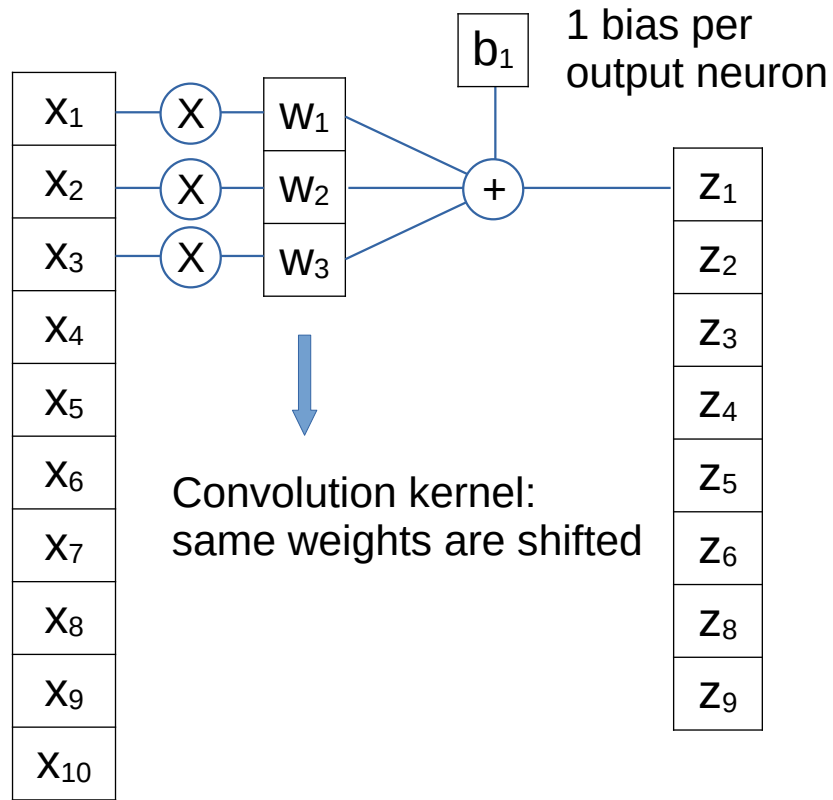


$$z_1 = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 + b$$

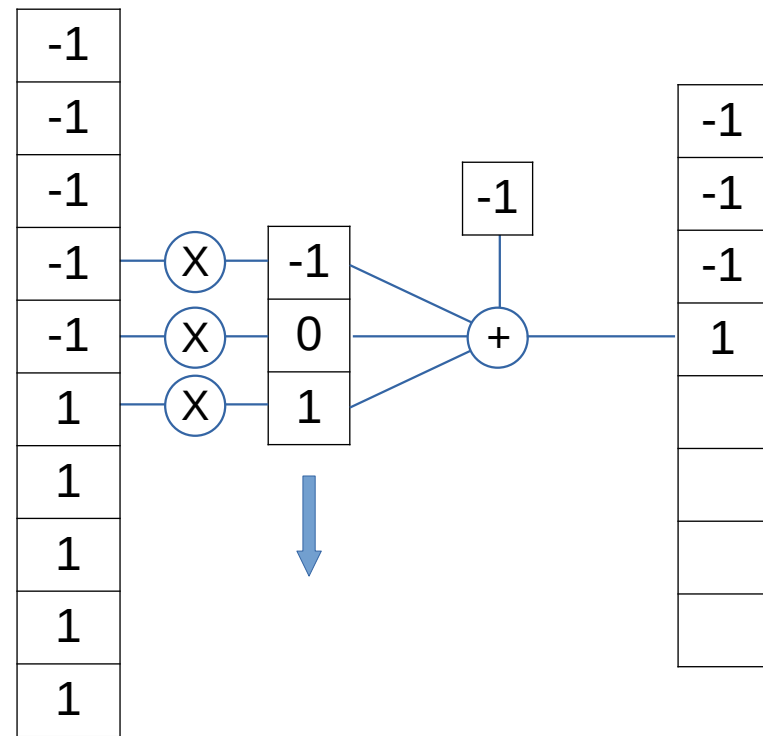


$$z_1 = -1 \times (-1) + 0 \times (-1) + 1 \times (-1) + (-1)$$

Convolutional neural networks: local correlations between inputs

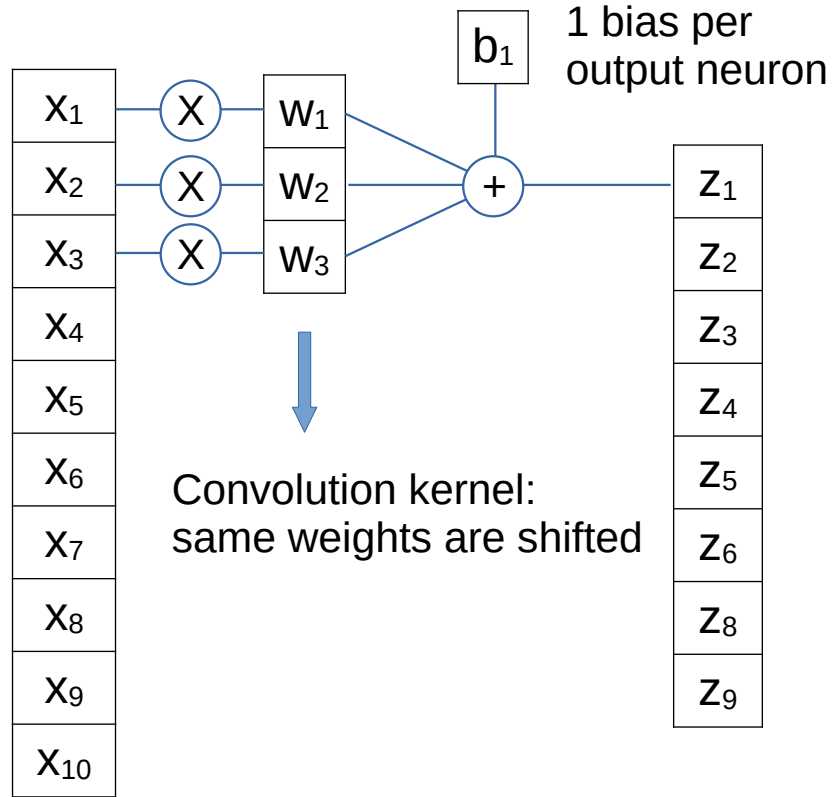


$$z_1 = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 + b$$

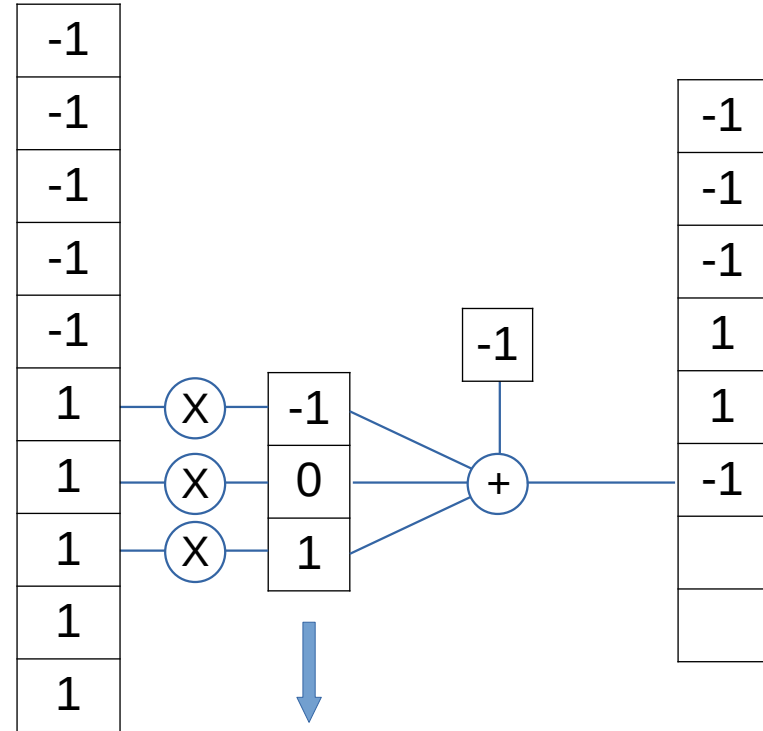


$$z_4 = -1 \times (-1) + 0 \times (-1) + 1 \times 1 + (-1)$$

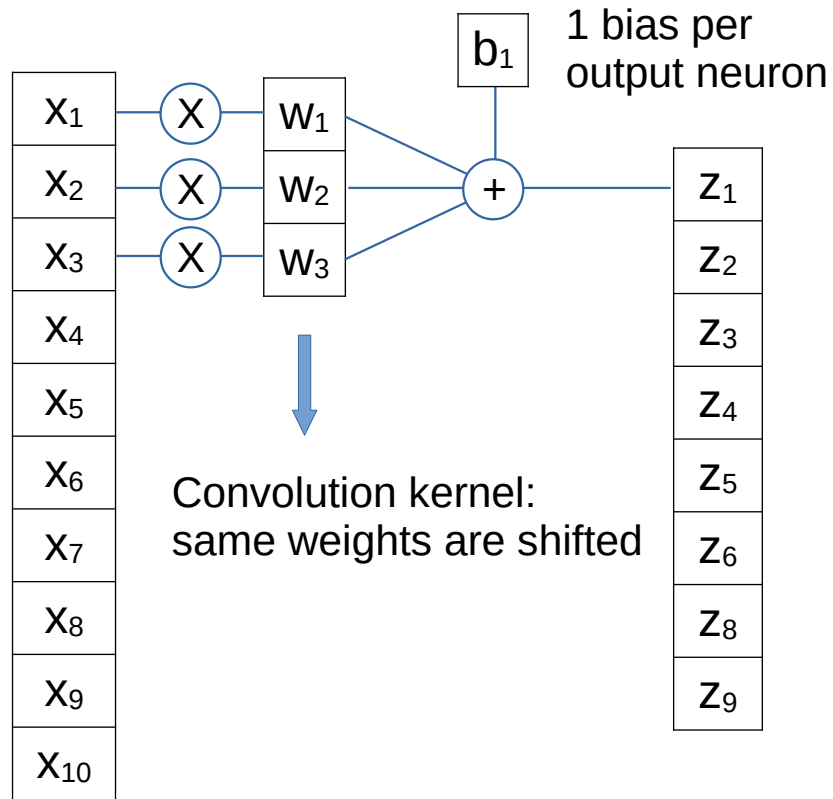
Convolutional neural networks: local correlations between inputs



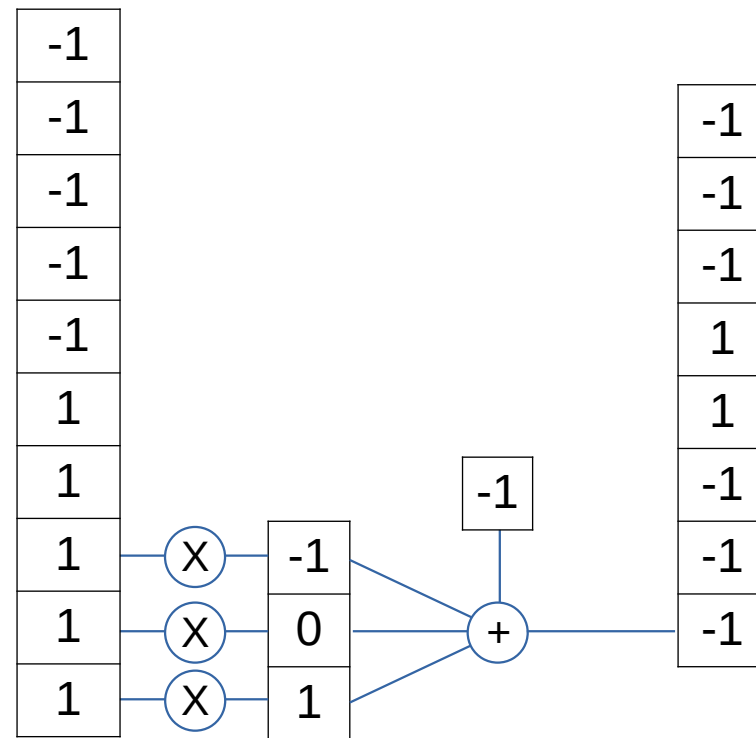
$$z_1 = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 + b$$



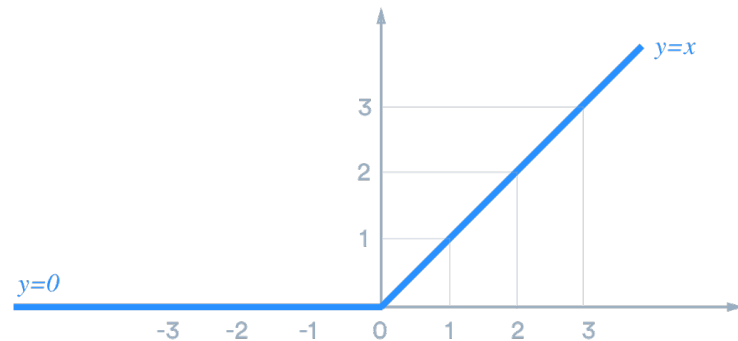
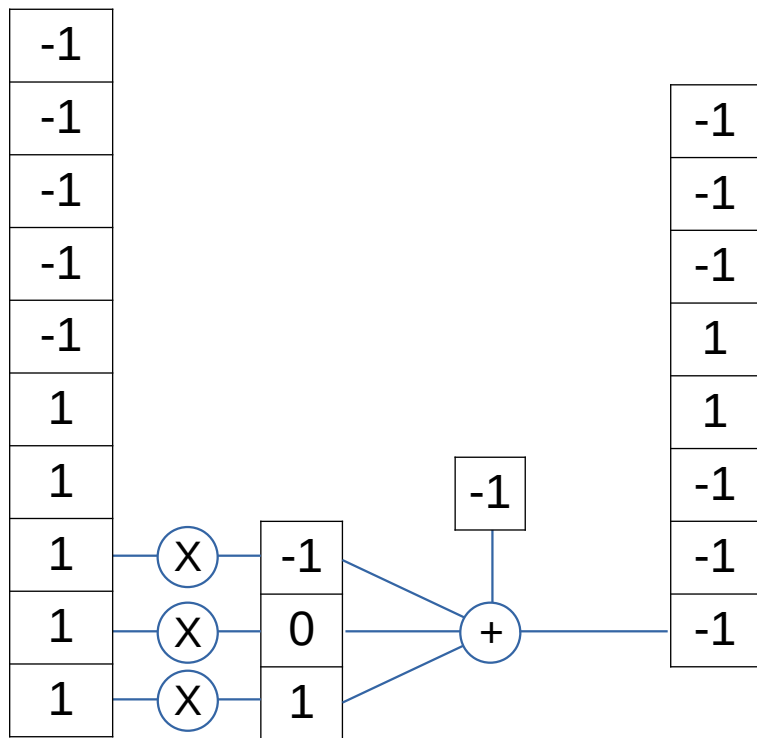
Convolutional neural networks: local correlations between inputs



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

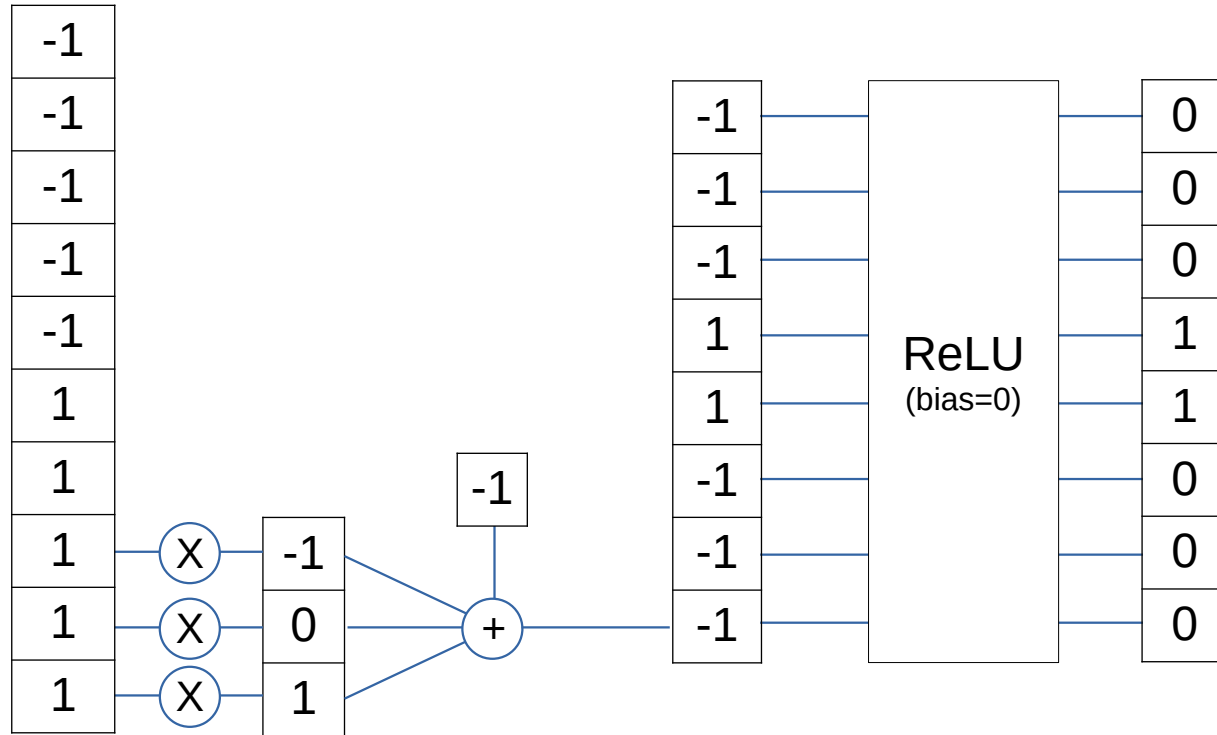


Adding ReLU (Rectified Linear Unit)

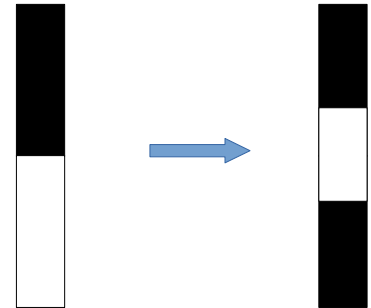


$$f(x) = \max(0, x) \quad \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

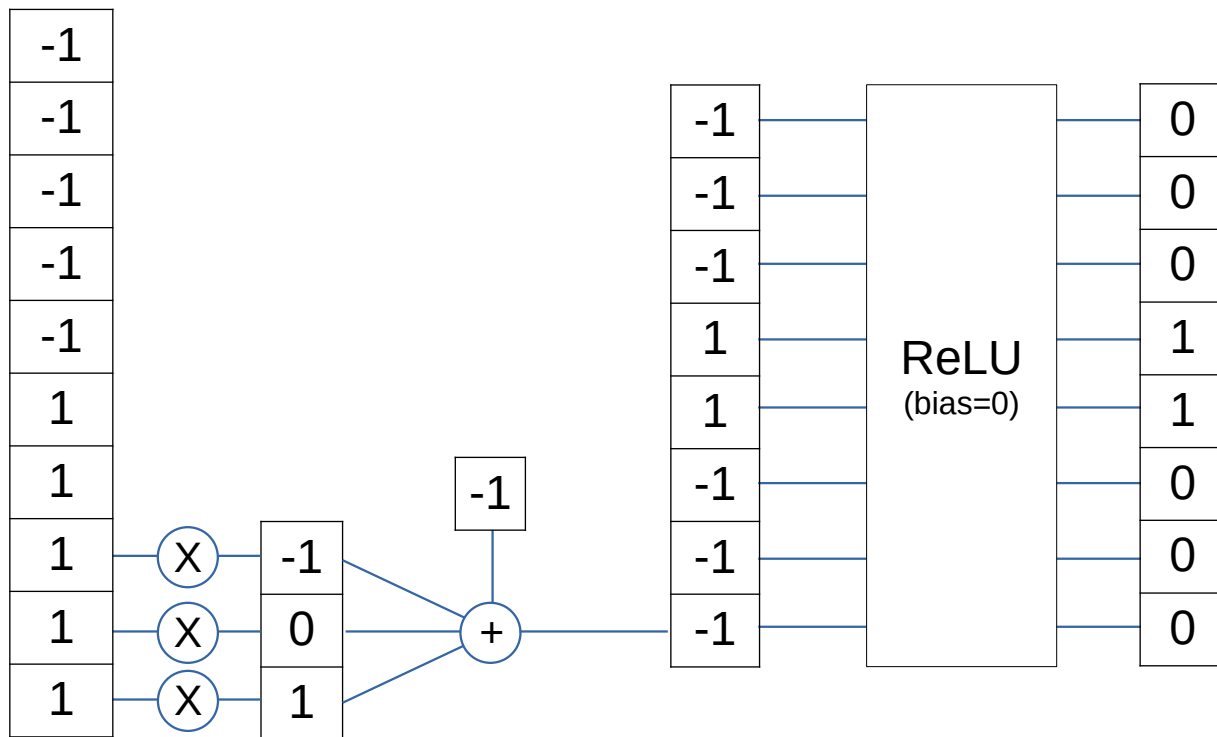
Adding ReLU (Rectified Linear Unit)



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



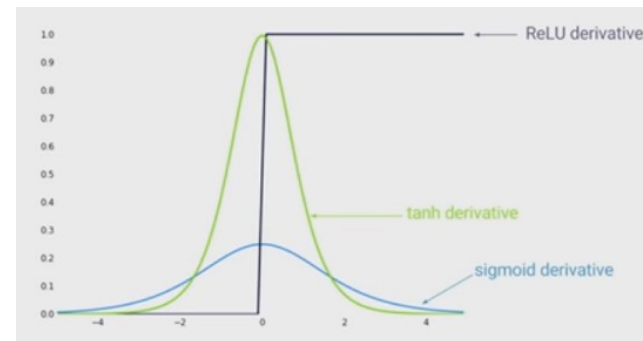
Adding ReLU (Rectified Linear Unit)



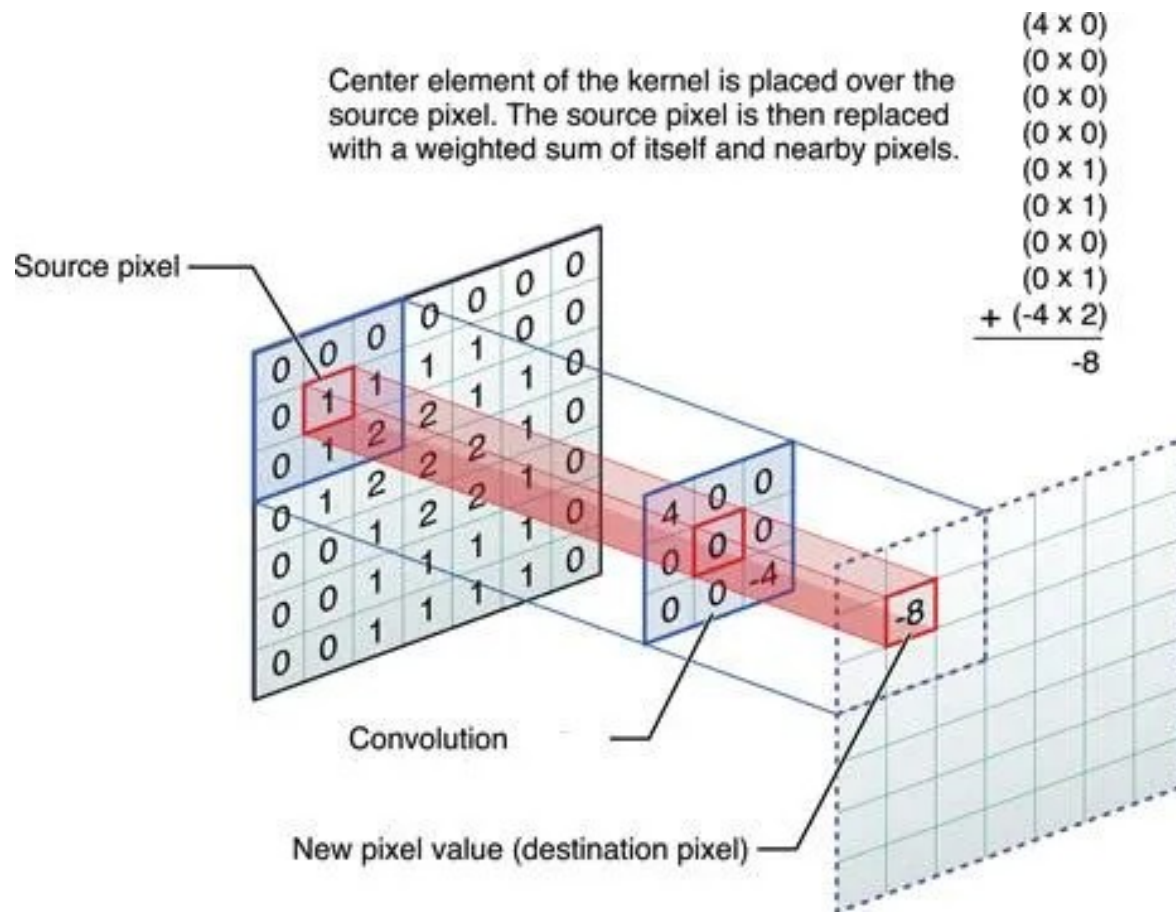
Remember backpropagation?

$$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

- Derivatives easy to compute
- Less subject to the “vanishing gradient problem”



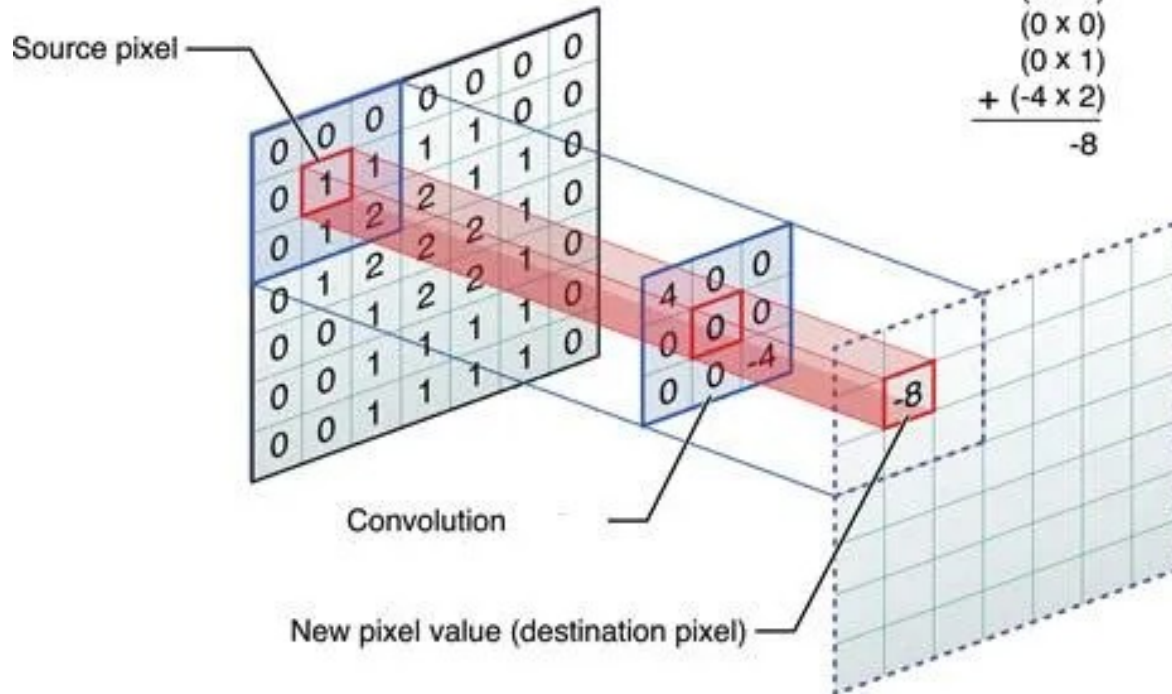
2D convolution



2D convolution

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

$$\begin{array}{r} (4 \times 0) \\ (0 \times 0) \\ (0 \times 0) \\ (0 \times 0) \\ (0 \times 1) \\ (0 \times 1) \\ (0 \times 0) \\ (0 \times 1) \\ + (-4 \times 2) \\ \hline -8 \end{array}$$



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

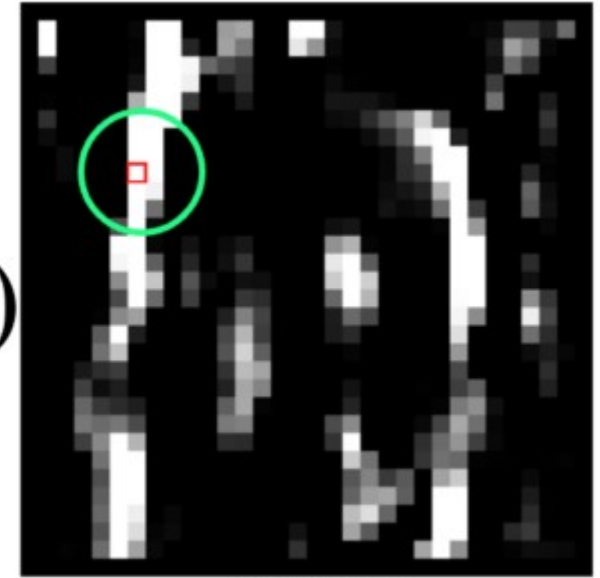


input image

$$\begin{pmatrix} 245 \times 1 + 171 \times 0 + 32 \times -1 \\ + 250 \times 2 + 158 \times 0 + 36 \times -2 \\ + 237 \times 1 + 181 \times 0 + 53 \times -1 \end{pmatrix} = 825$$

kernel:

left sobel

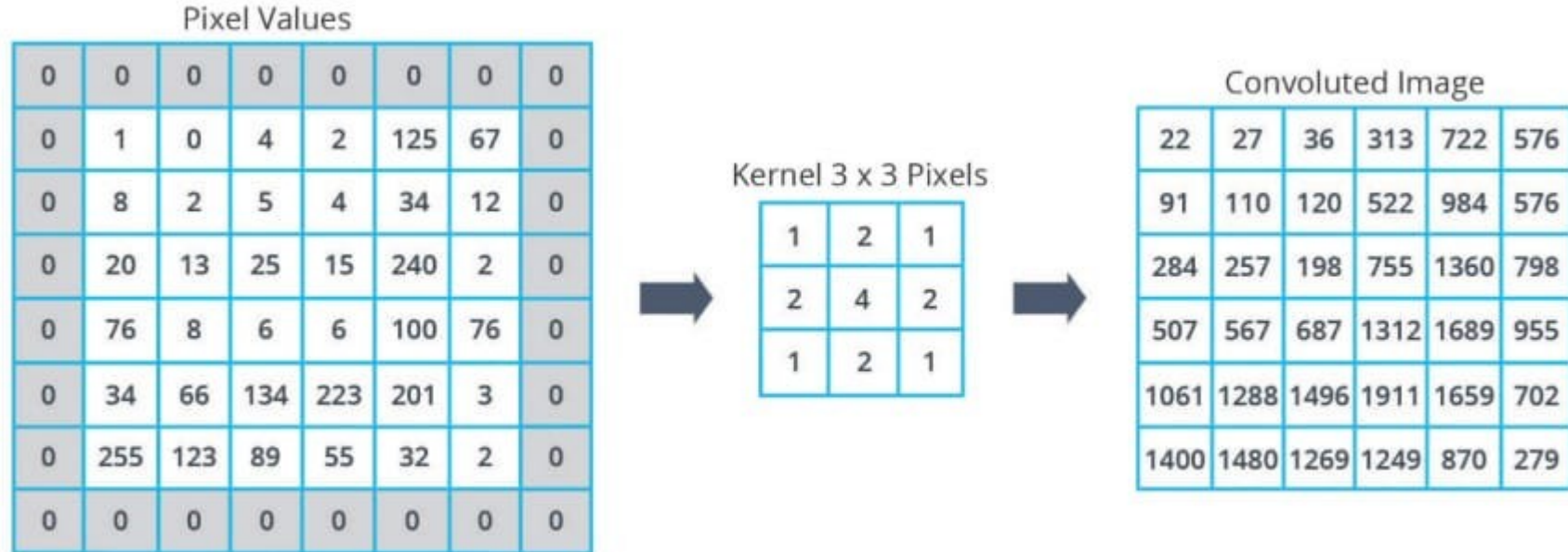


output image

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

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

Convolution “erodes” the image



Padding to avoid erosion

K 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



55	52
57	50

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

*

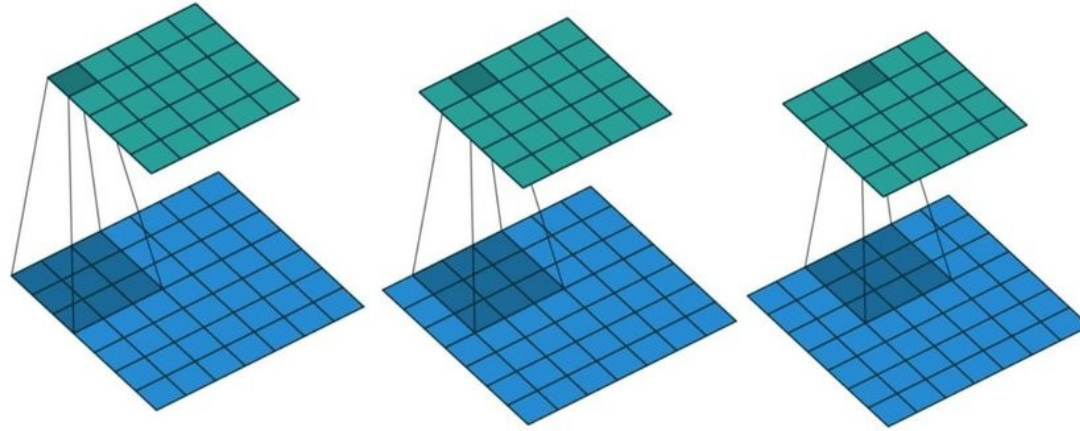
1	2	1
2	1	2
1	1	2



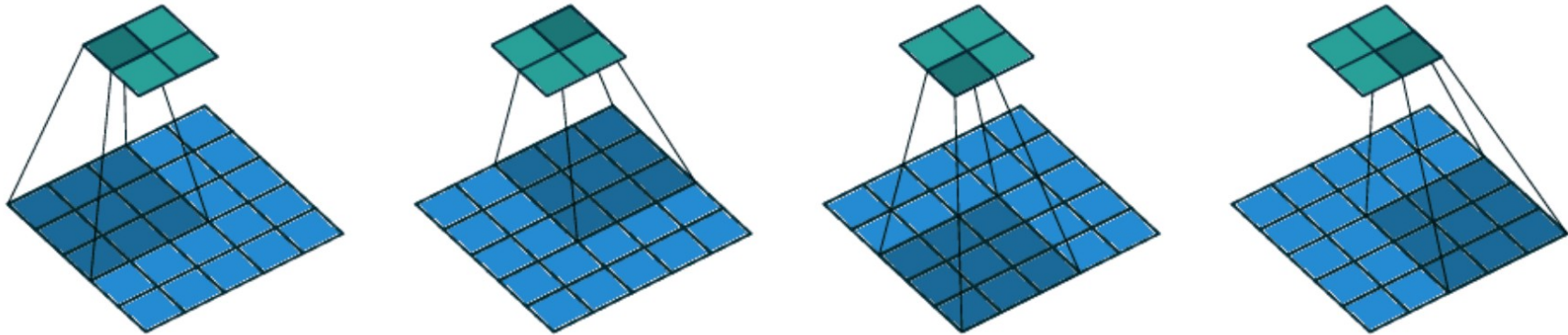
19	26	46	22
29	55	52	40
42	57	50	43
36	46	44	19

Stride to speed up processing (and erosion)

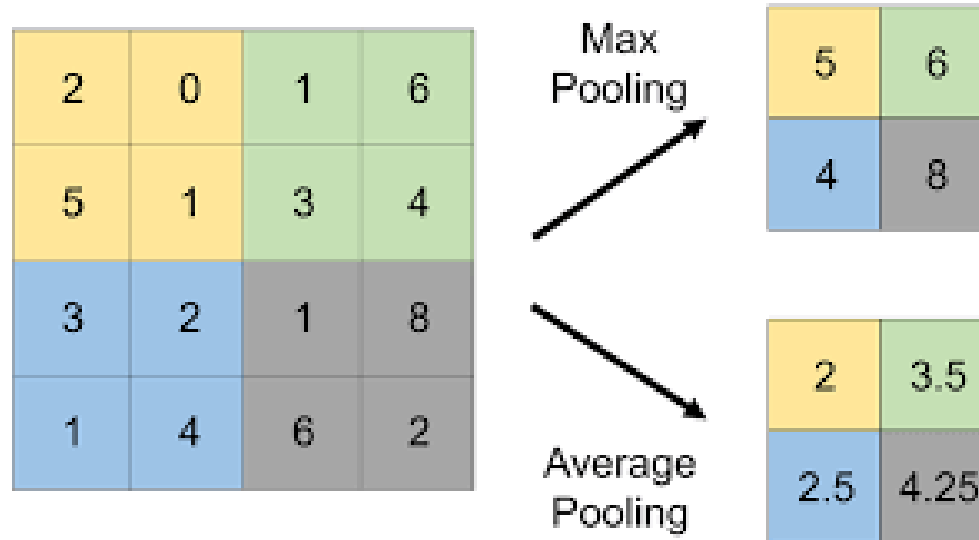
Stride = 1



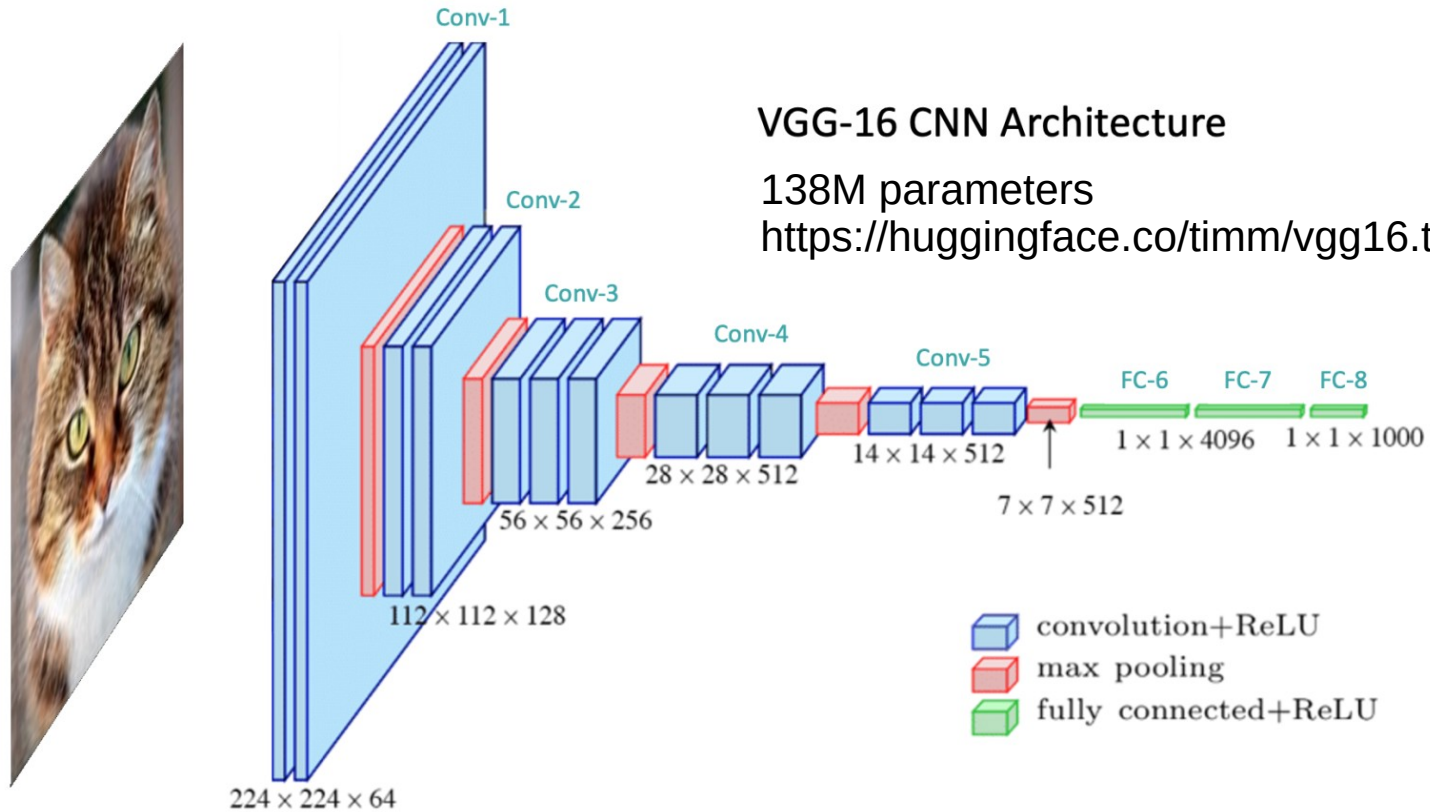
Stride = 2



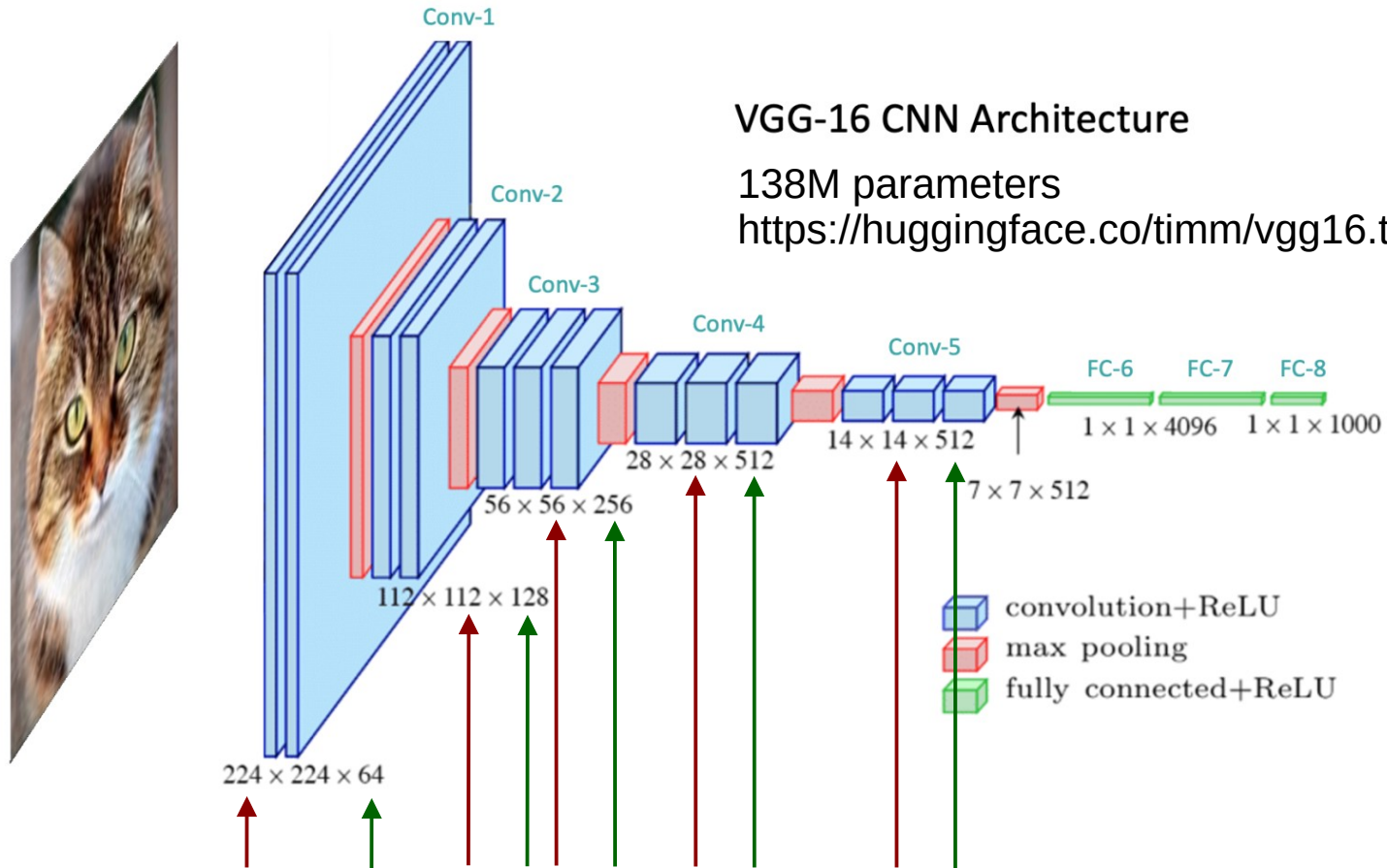
Downsampling: Max/average pooling



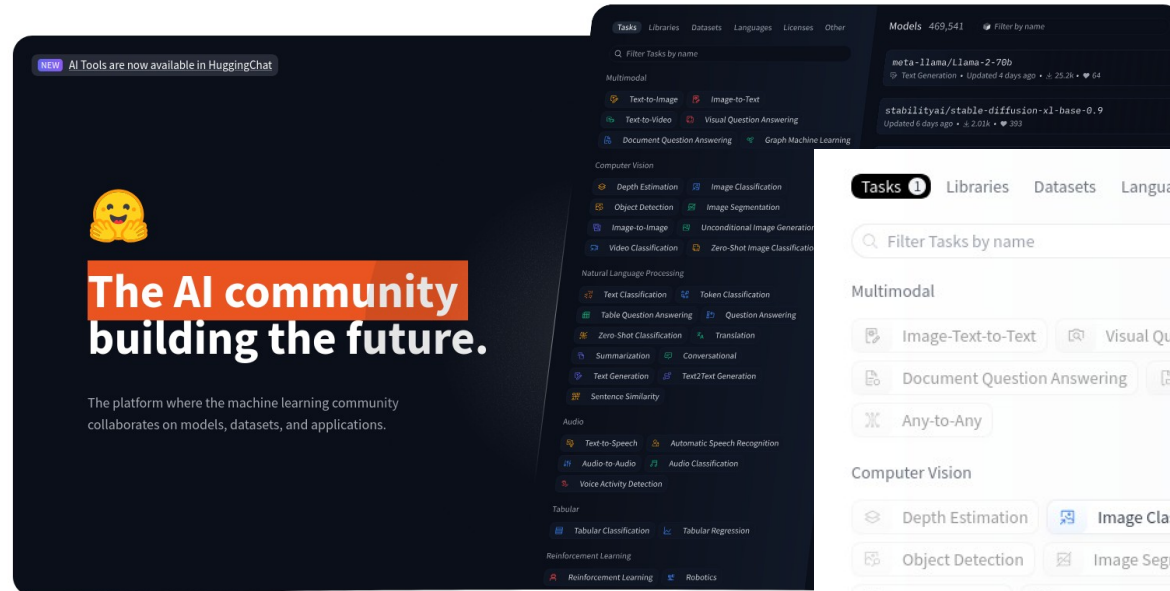
Real example: VGG



Decreasing size, increasing feature number



Hugging face



Trending on 🤖 this week

Models

openai/whisper-large-v3-turbo

Updated 4 days ago • 62.3k • 698

nvidia/NVLM-D-72B

Updated 4 days ago • 7.69k • 491

black-forest-labs/FLUX.1-dev

Updated Aug 16 • 1.14M • 5,22k

ostris/OpenFLUX.1

Updated 4 days ago • 7.97k • 359

meta-llama/Llama-3.2-11B-Vision-Instruct

Updated 8 days ago • 427k • 563

Spaces

Open NotebookLM

640

Whisper Turbo

394

Kolors Virtual Try-On

3,69k

FacePoke

304

Expression Editor

353

Browse 400k+ models

Browse 150k+ applications

Browse 100k+ datasets

Tasks Libraries Datasets Languages Licenses Other

Filter Tasks by name

Reset Tasks

Multimodal

Image-Text-to-Text

Visual Question Answering

Document Question Answering

Video-Text-to-Text

Any-to-Any

Computer Vision

Depth Estimation

Image Classification

Object Detection

Image Segmentation

Text-to-Image

Image-to-Text

Image-to-Image

Image-to-Video

Unconditional Image Generation

Video Classification

Text-to-Video

Zero-Shot Image Classification

Mask Generation

Zero-Shot Object Detection

Text-to-3D

Image-to-3D

Image Feature Extraction

Keypoint Detection

Natural Language Processing

Text Classification

Token Classification

Table Question Answering

Question Answering

Zero-Shot Classification

Translation

Models 136

vgg

amehta633/cifar-10-vgg-pretrained

Image Classification • Updated Jun 8, 2022 • 12

edadaltocg/vgg16_bn_cifar100

Image Classification • Updated Feb 20, 2023 • 43

timm/vgg11.tv_in1k

Image Classification • Updated Apr 25, 2023 • 716

timm/vgg13.tv_in1k

Image Classification • Updated Apr 25, 2023 • 523

timm/vgg16.tv_in1k

Image Classification • Updated Apr 25, 2023 • 28.3k • 5

timm/vgg19.tv_in1k

Image Classification • Updated Apr 25, 2023 • 9.14k • 2

schibfab/landscape_classification_vgg16_fine_tuned-...

Image Classification • Updated May 28, 2023

Grecko/AQUA-vgg16-imagenet

Image Classification • Updated Jun 19, 2023 • 2

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

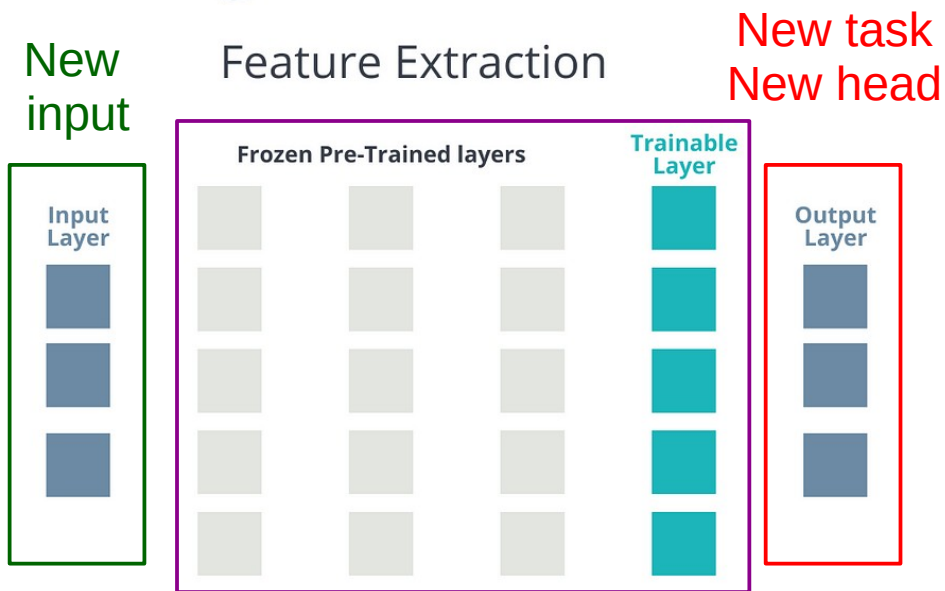
December 2023

DOI: [10.1109/SMARTGENCON60755.2023.10442010](#)

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

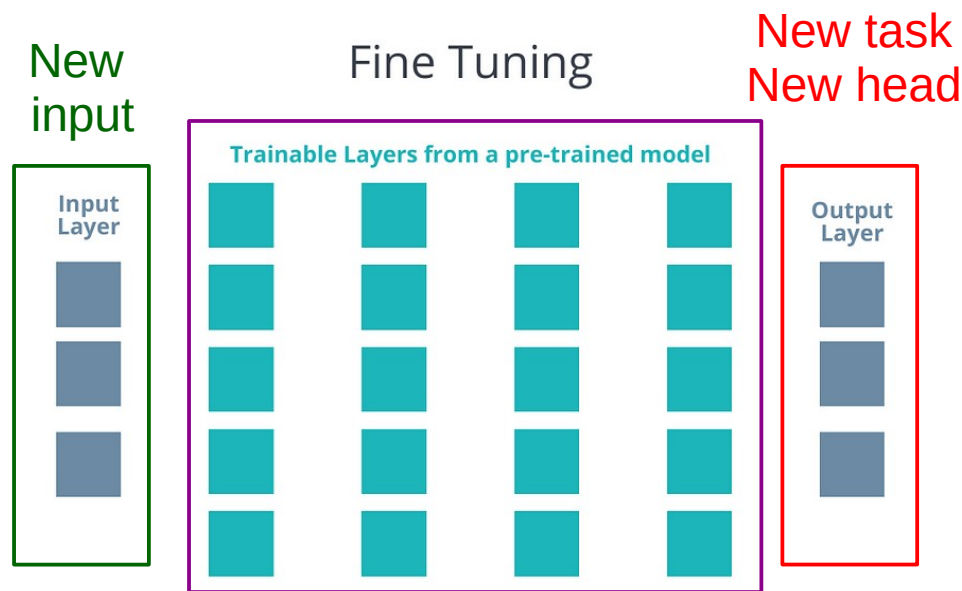
Raghuvanshi S, Dhariwal S. The VGG16 Method Is a Powerful Tool for Detecting Brain Tumors Using Deep Learning Techniques. *Engineering Proceedings*. 2023; 59(1):46. <https://doi.org/10.3390/engproc2023059046>

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



Existing model

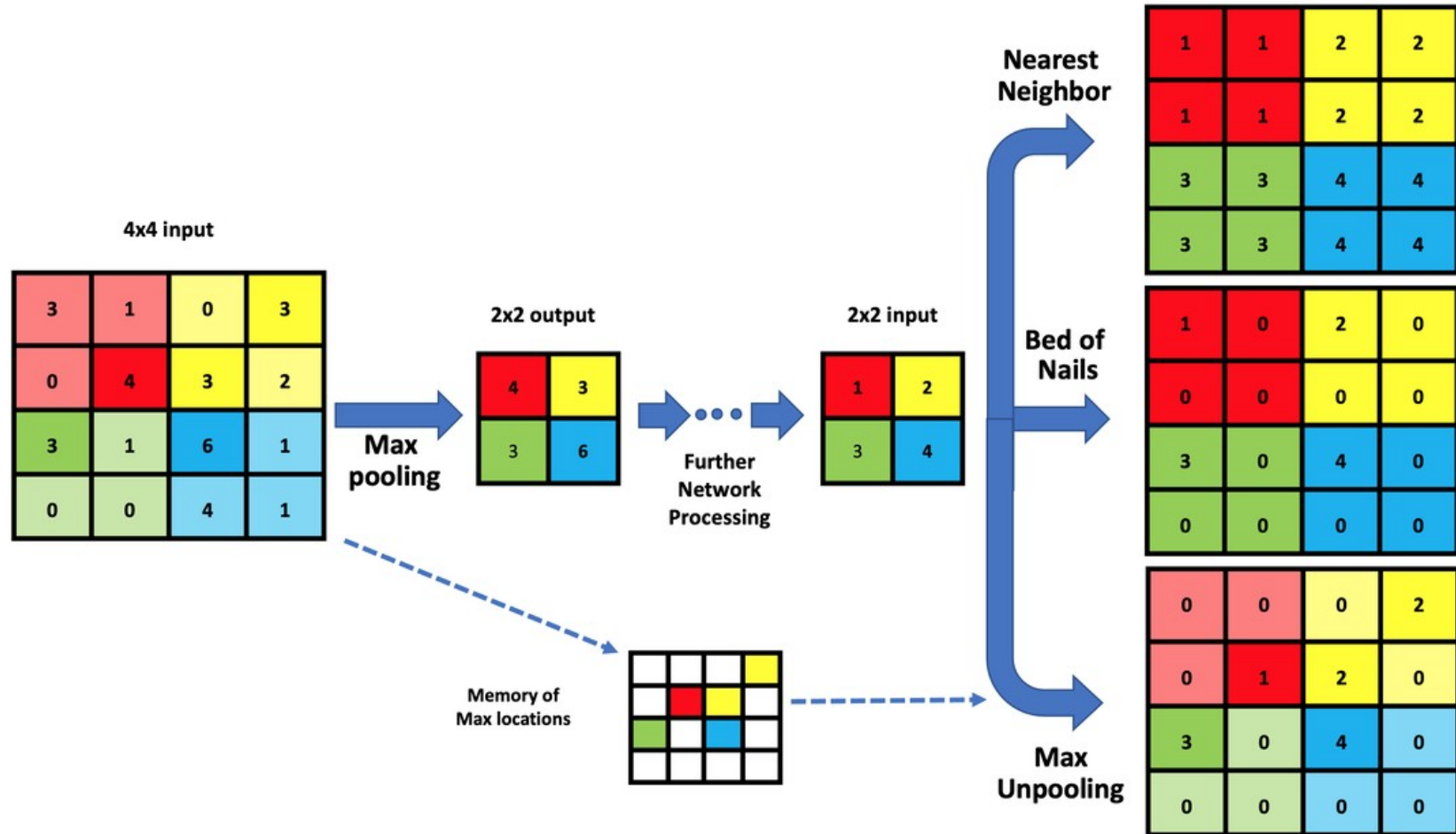
In feature extraction, you **freeze the pre-trained model** layers to preserve existing learning and **add new layers** to learn additional information.



Existing model

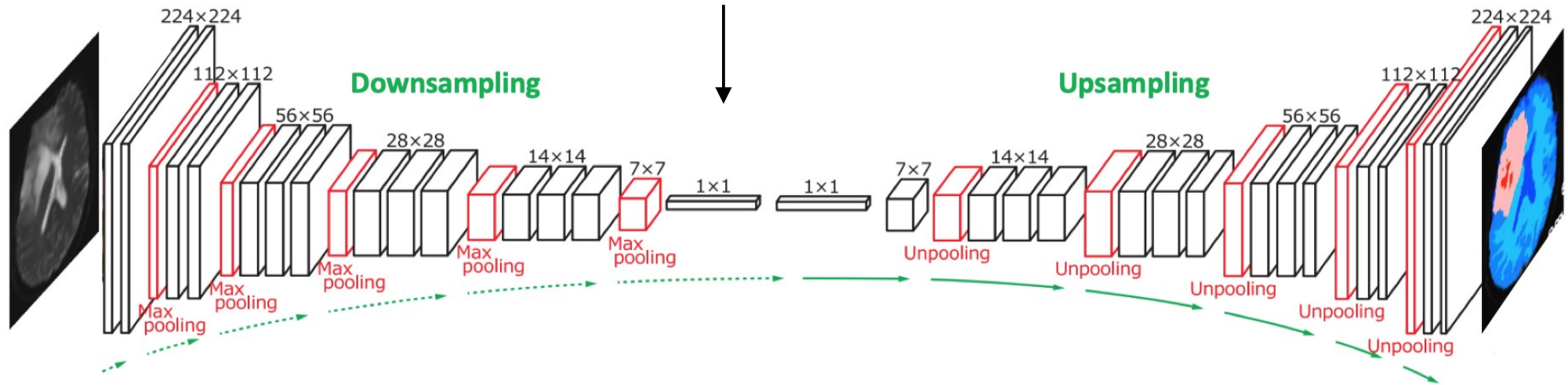
In fine-tuning, you **unfreeze the entire model** and **train it with a lower learning rate** to adapt to new challenges.

Upsampling

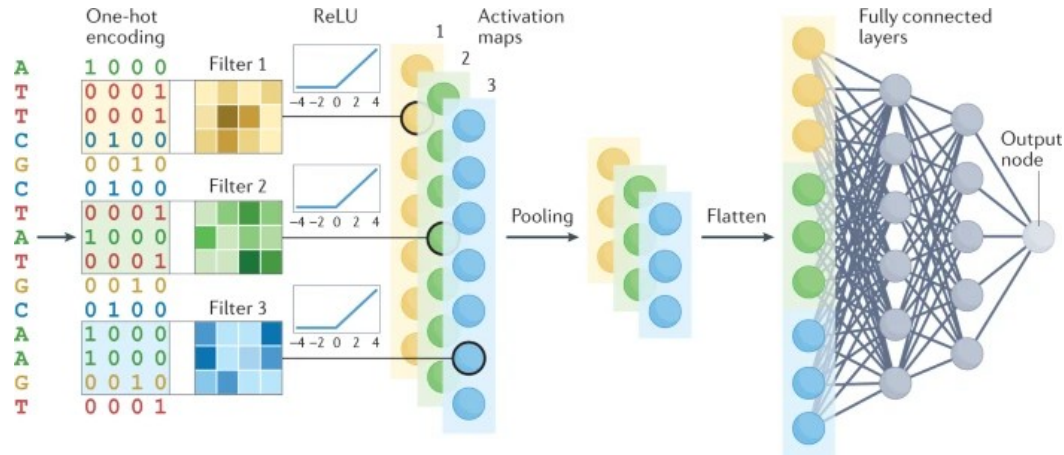


Examples: tumour detection

Latent space; here, a vector
that contains all the information
we want and only that



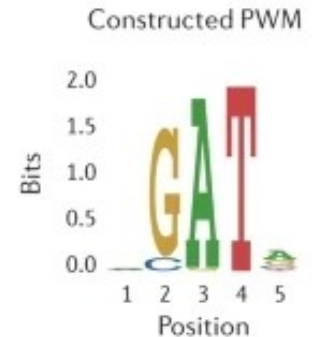
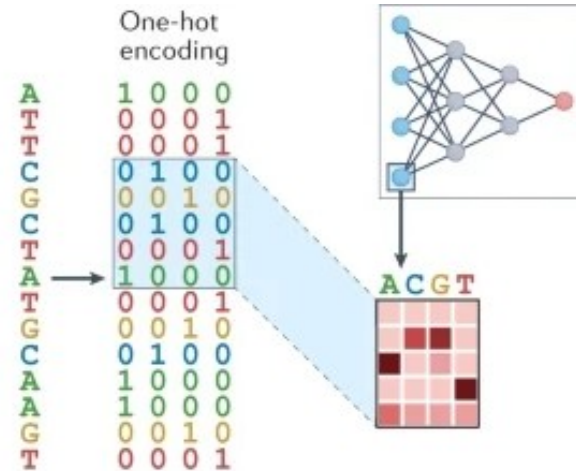
Everything is an image: DNA sequences to images



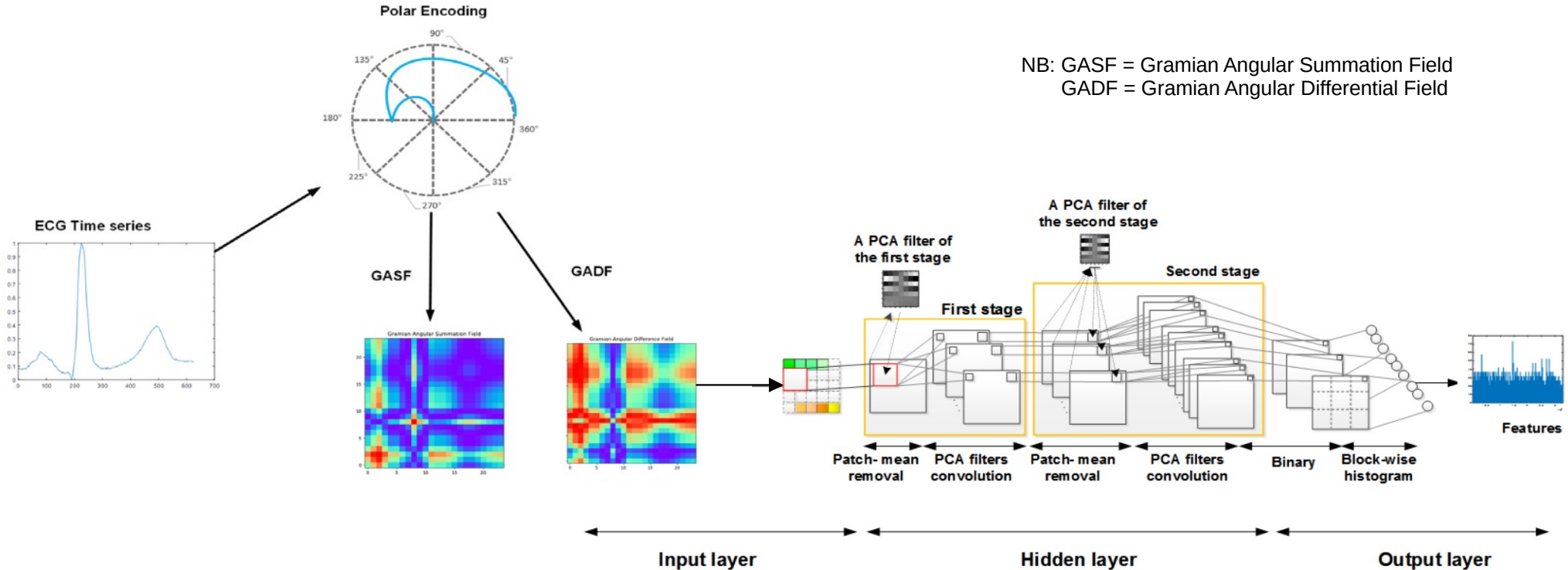
Obtaining genetics insights from deep learning via explainable artificial intelligence

[Gherman Novakovsky](#), [Nick Dexter](#), [Maxwell W. Libbrecht](#) , [Wyeth W. Wasserman](#)  & [Sara Mostafavi](#) 

Nature Reviews Genetics 24, 125–137 (2023) | [Cite this article](#)



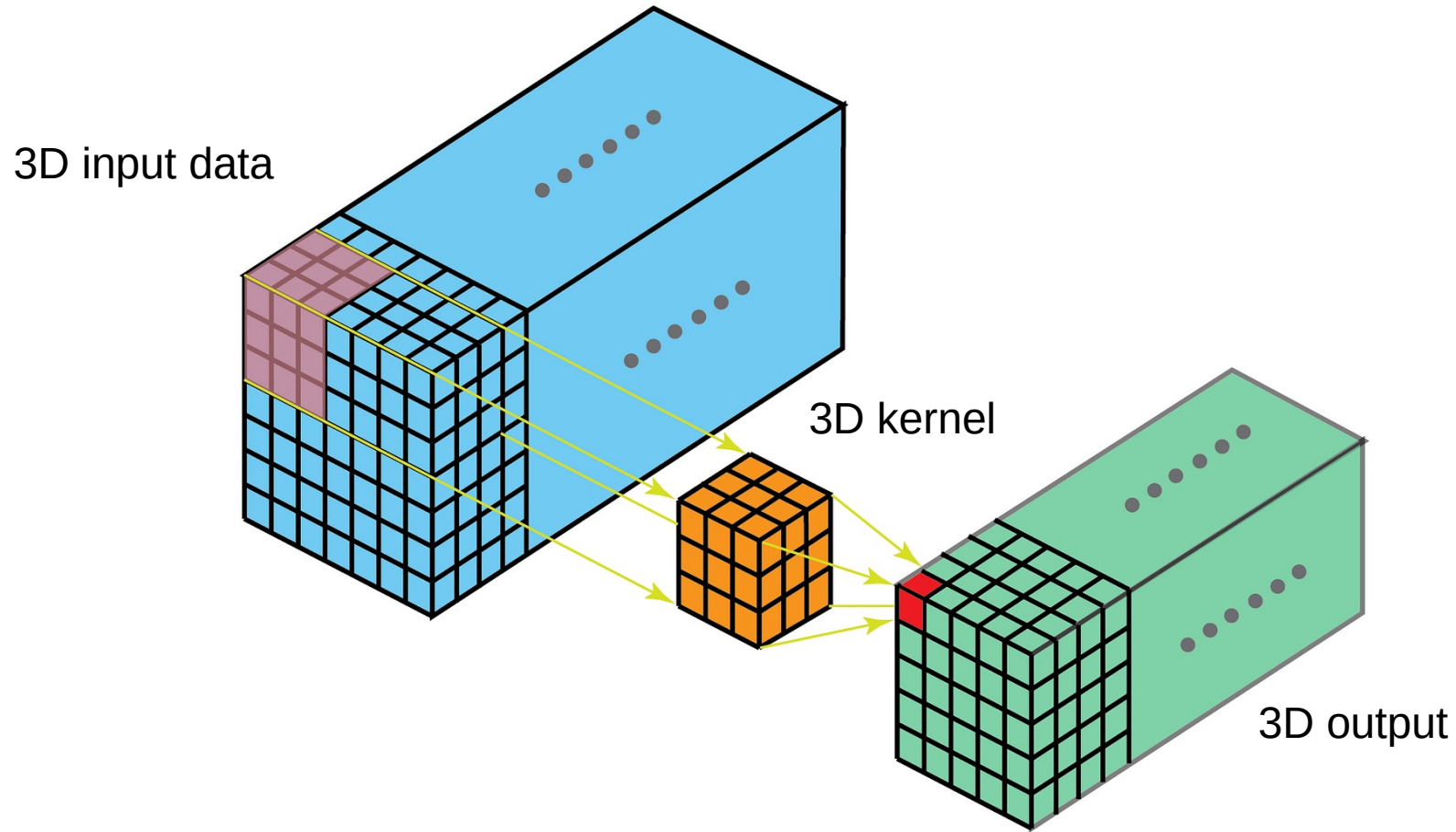
Everything is an image: Time series to images



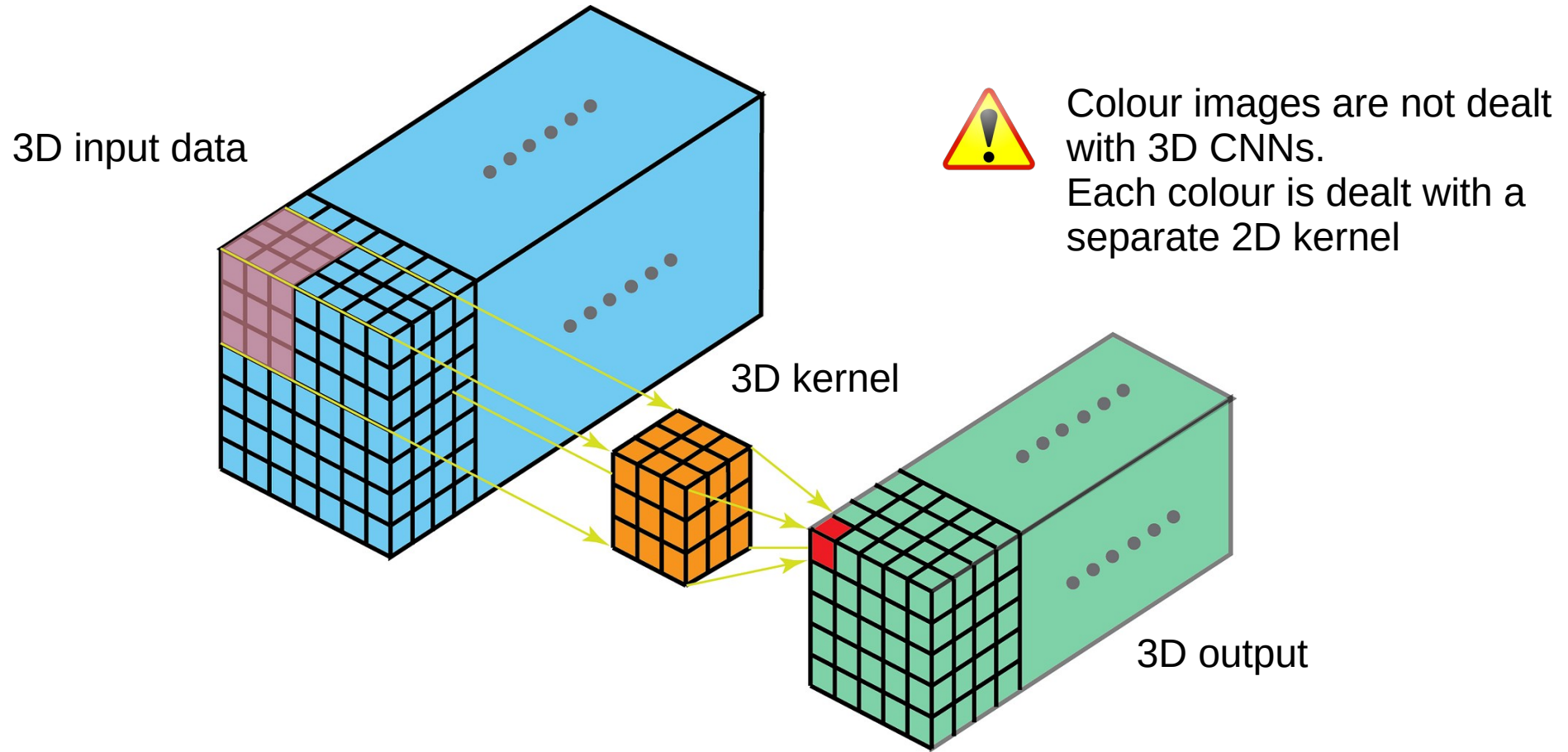
Wang and Oates (2015) Spatially Encoding Temporal Correlations to Classify Temporal Data Using Convolutional Neural Networks. arXiv:1509.07481v1

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



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**



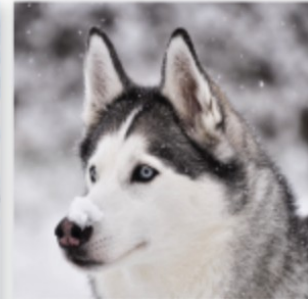
Predicted: **Wolf**
True: **Wolf**



Predicted: **Husky**
True: **Wolf**



Predicted: **Wolf**
True: **Wolf**



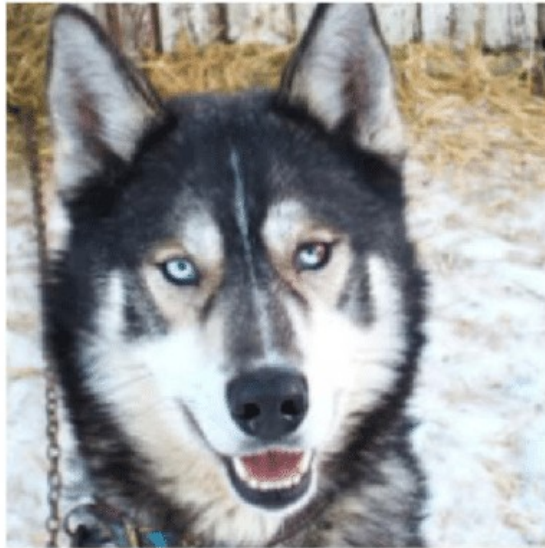
Predicted: **Wolf**
True: **Husky**



Predicted: **Husky**
True: **Husky**

CNNs overfit too

Predicted: **Wolf**
True: **Husky**



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

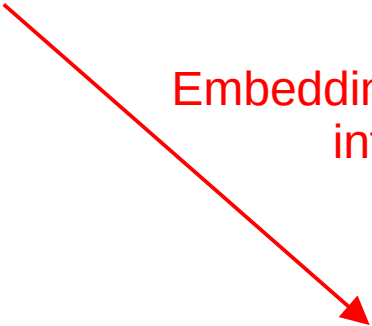
Tulio Ribeiro M, Singh S, Guestrin C (2016) "Why Should I Trust You?": Explaining the Predictions of Any Classifier. KDD '16: Proc 22nd ACM SIGKDD Intl Conf Knowledge Discov Data Mining, 1135 – 1144
<https://arxiv.org/pdf/1602.04938>

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
bunny	1	0	0	0	0	0	0	0	0
rabbit	0	0	0	0	0	0	1	0	0
hamster	0	0	1	0	0	0	0	0	0
hutches	0	0	0	1	0	0	0	0	0

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

Semantics (Embedding space)

	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

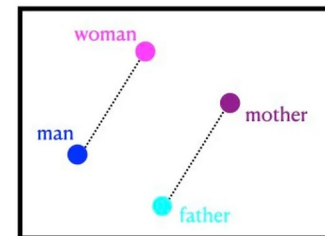
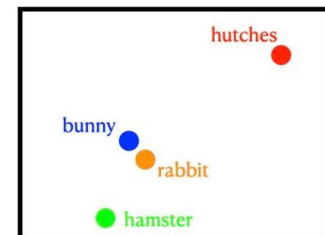
	alive	mammal	rodent	bipedel	gender	plural
mother	0.9	0.1	0.1	0.2	0.8	-0.7
father	0.9	0.2	0.1	0.2	-0.8	-0.7
woman	0.9	0.8	-0.9	0.9	0.8	-0.8
man	0.9	0.8	-0.7	0.9	-0.8	-0.8

Word

Vector Embedding

Dimensionality
Reduction

Dimensionality



2D Visualization

Embeddings (“plongement”)

Dictionary (initial space)

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

mother	0	0	0	0	0	1	0	0	0
father	0	1	0	0	0	0	0	0	0
woman	0	0	0	0	0	0	0	0	1
man	0	0	0	0	1	0	0	0	0

Semantics (Embedding space)

	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

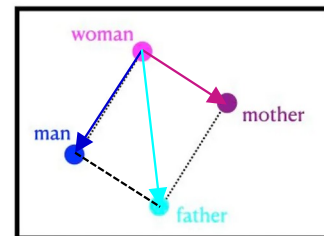
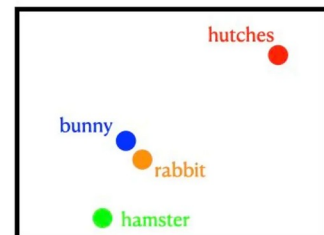
mother	0.9	0.1	0.1	0.2	0.8	-0.7
father	0.9	0.2	0.1	0.2	-0.8	-0.7
woman	0.9	0.8	-0.9	0.9	0.8	-0.8
man	0.9	0.8	-0.7	0.9	-0.8	-0.8

Word

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

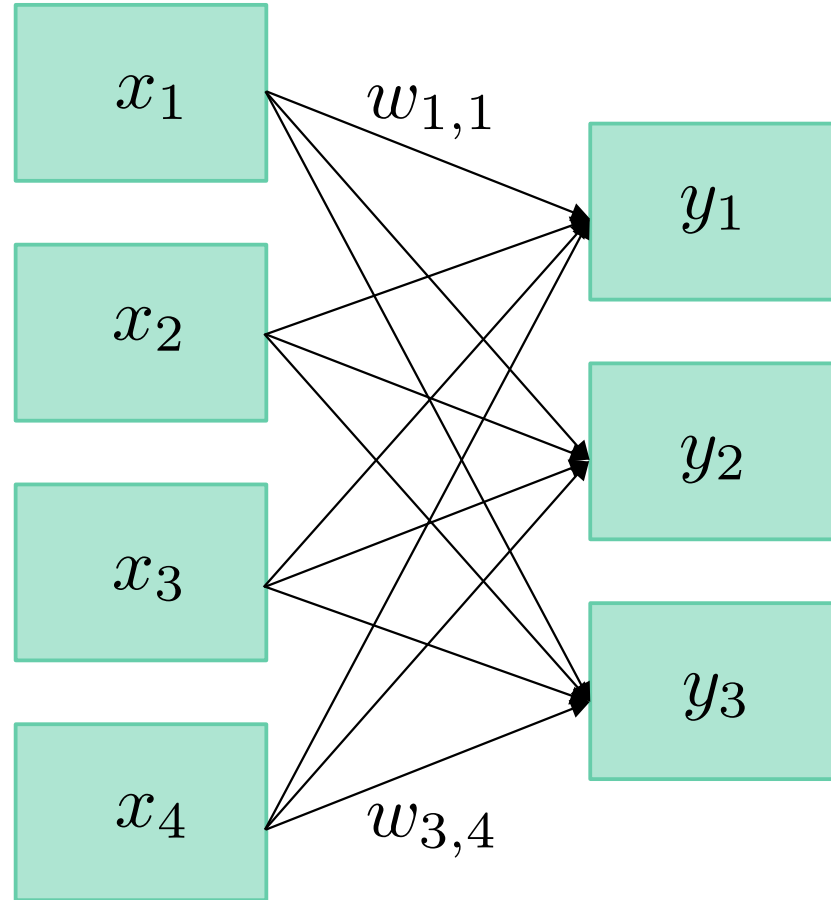
G_1
 G_2
 G_3
 G_4
...
 G_i
...
 G_N

Coordinates = M principal components

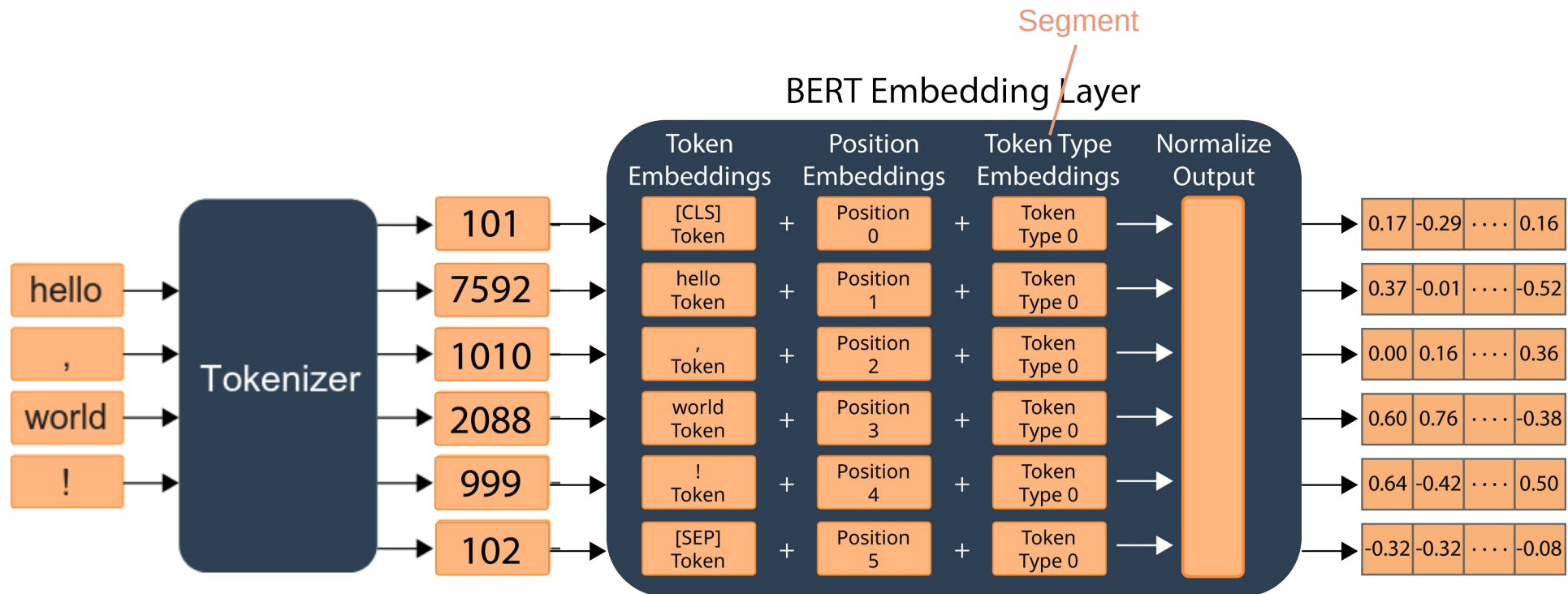
PC_1
 PC_2
 PC_3
 PC_4
...
 PC_j
...
 PC_M



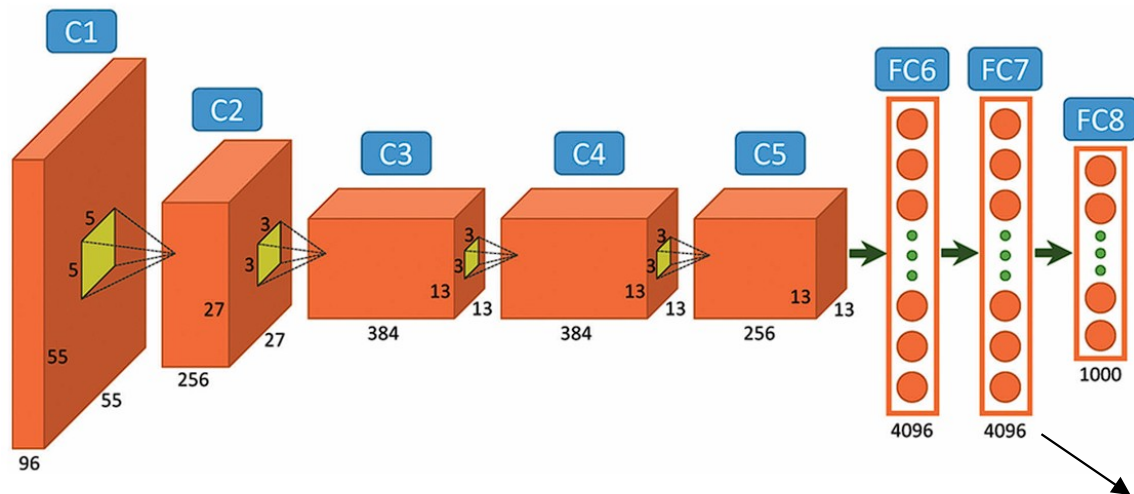
DL: layer to learn the embedding



There can be several embeddings



Deep CNNs are embedding the images



6 nearest neighbours in the 4096 dimension space

input image

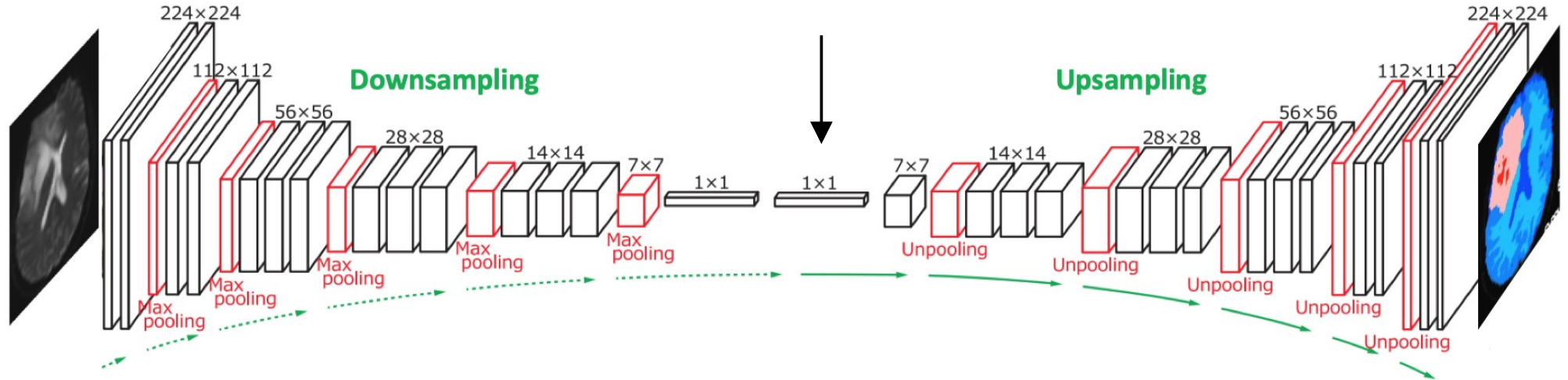


Krizhevsky A, Sutskever I, Hinton GE (2012)
ImageNet Classification with Deep
Convolutional Neural Networks
[https://proceedings.neurips.cc/paper/2012/file/
c399862d3b9d6b76c8436e924a68c45b-Paper.pdf](https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf)

(presenting AlexNet, the first Deep Convolutional Network)

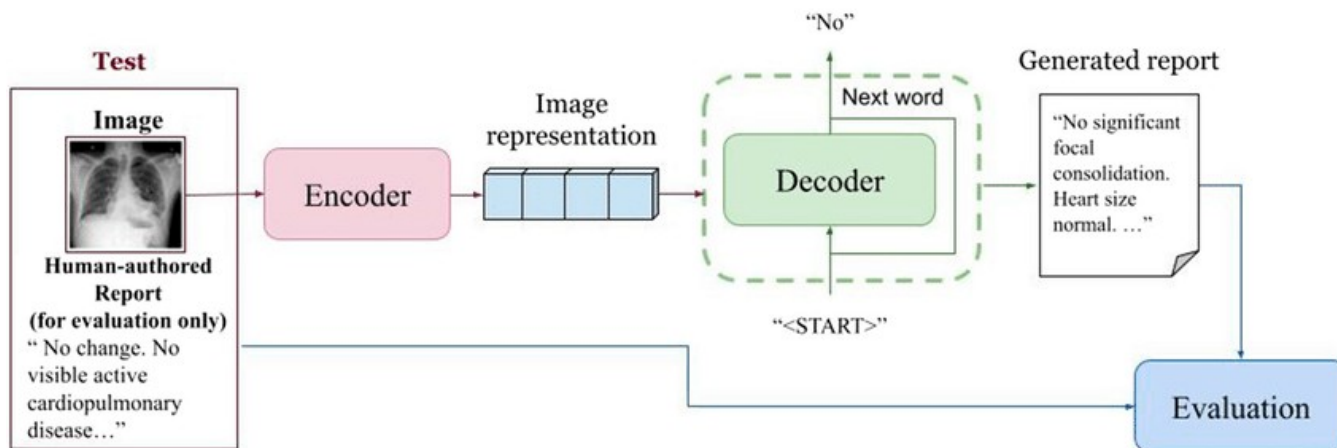
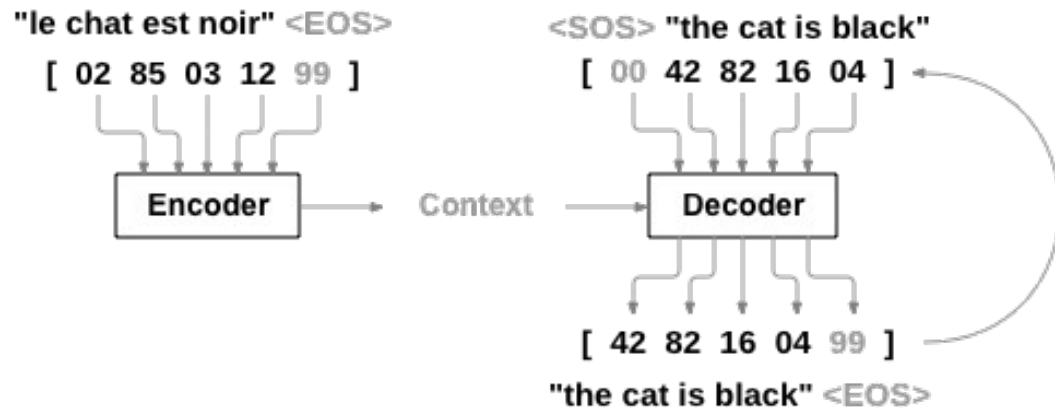
Encoder-decoder

Embedding in the feature space
Similar images will be close

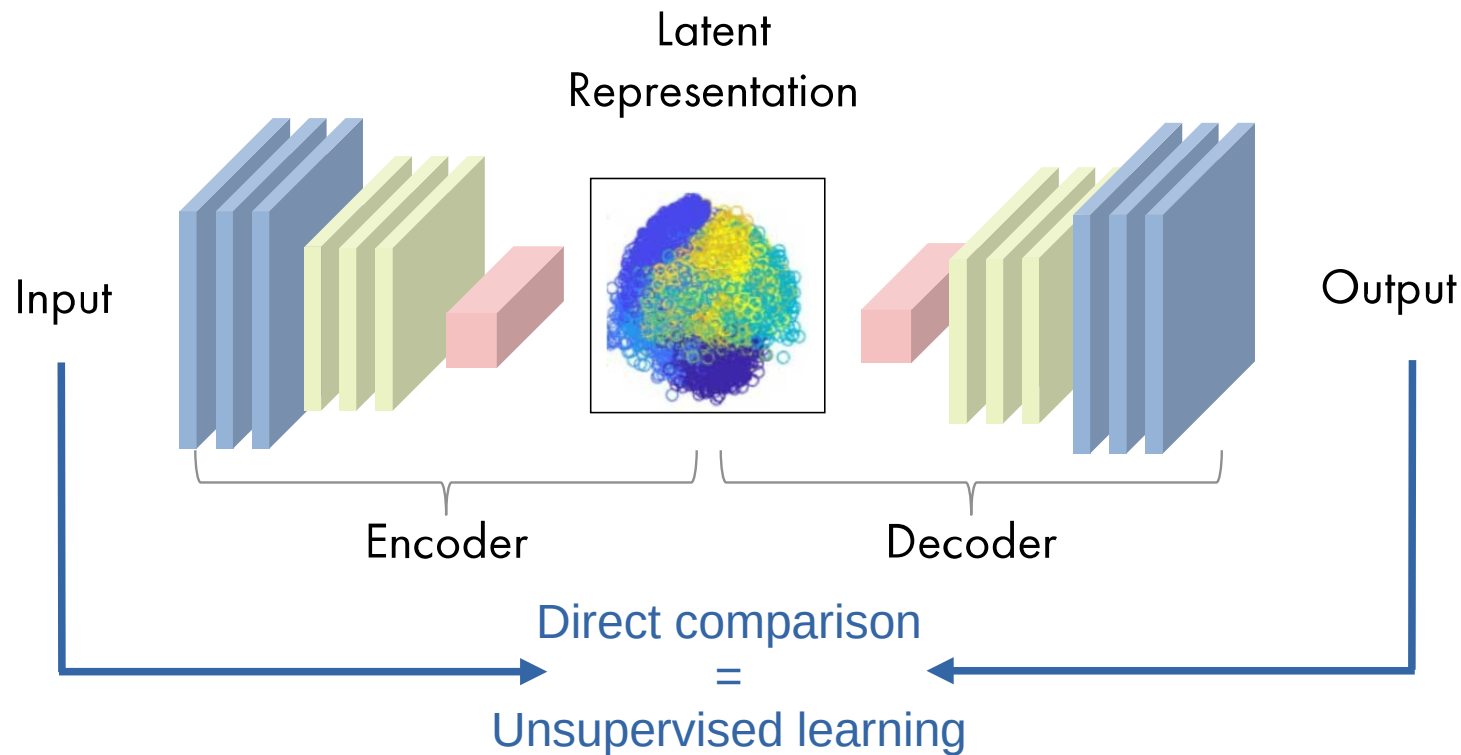


Encoders and decoders can be anything

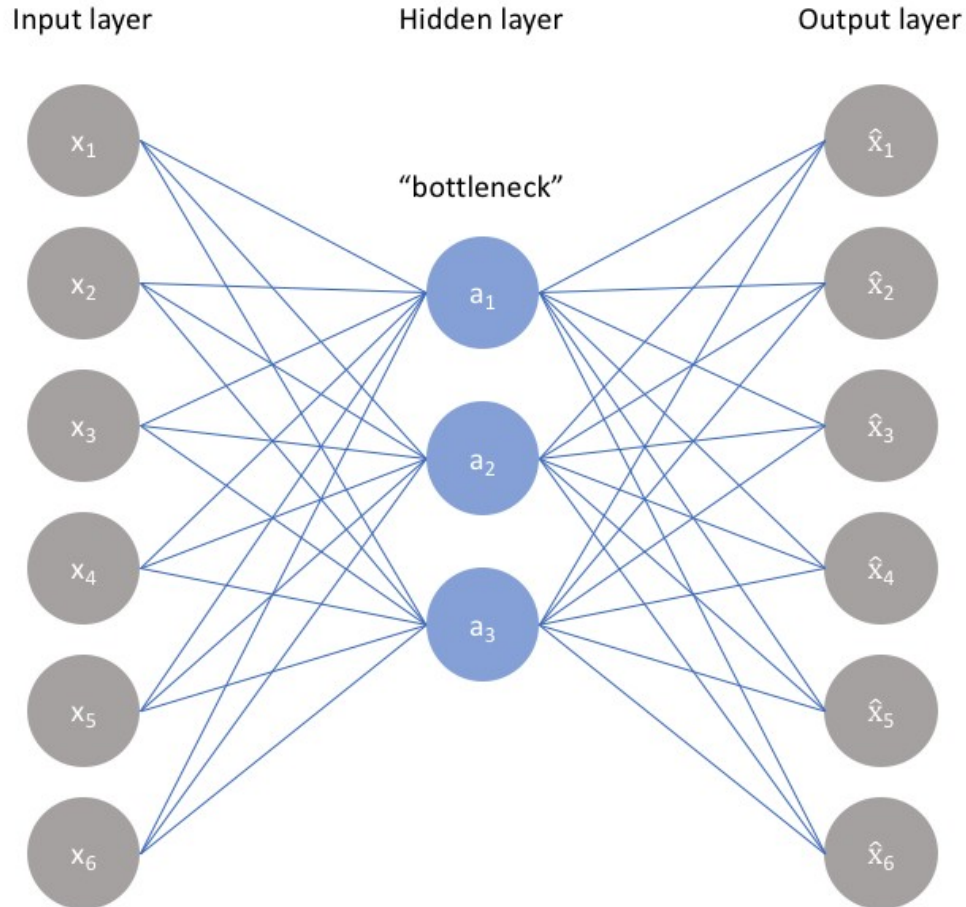
Pavlopoulos *et al* (2022)
doi:10.1007/s10115-022-01684-7



Learn by itself: AutoEncoder

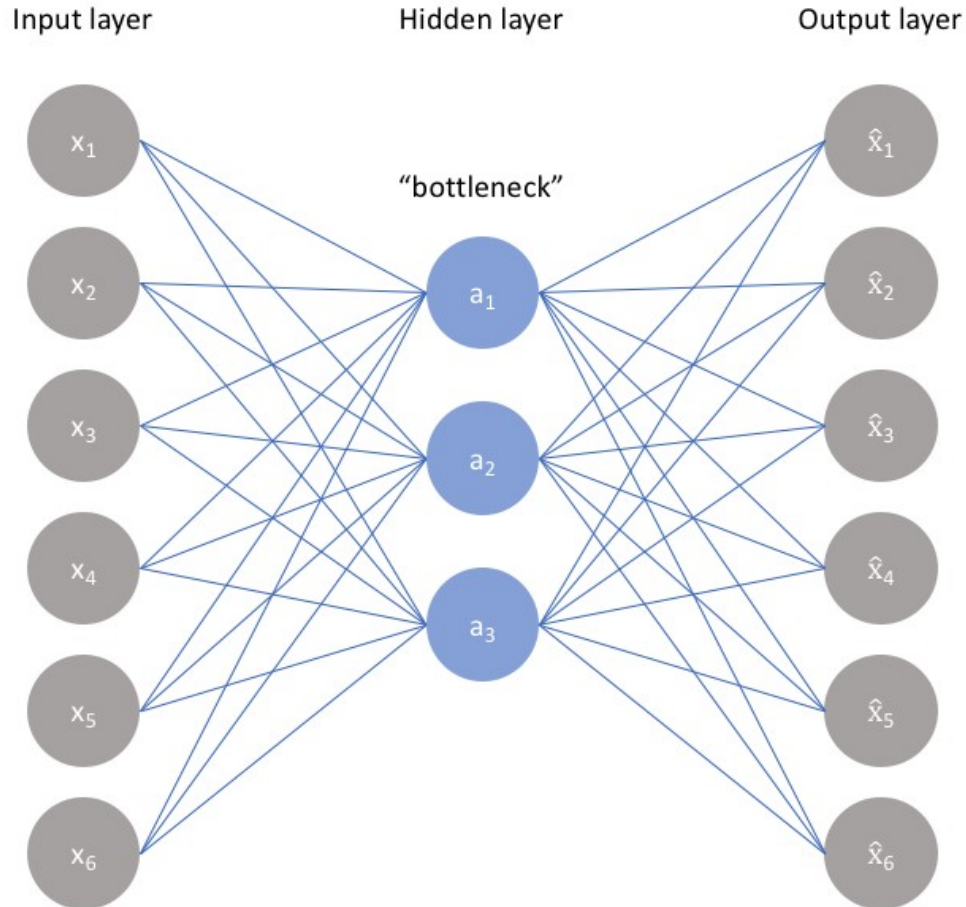


AutoEncoder ~ dimension reduction



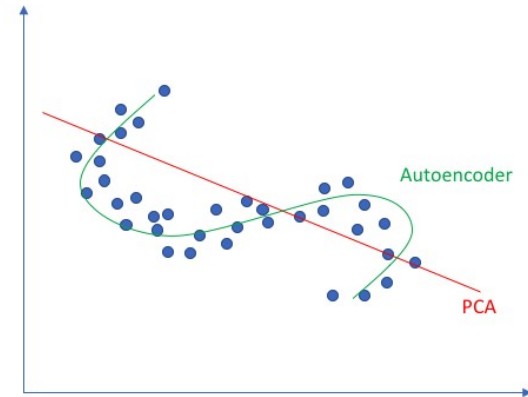
If you know the vector A , you can reconstruct all the x with the weights on the right. Same weights for all X s! All the meaningful variability is now contained in the A

AutoEncoder ~ dimension reduction

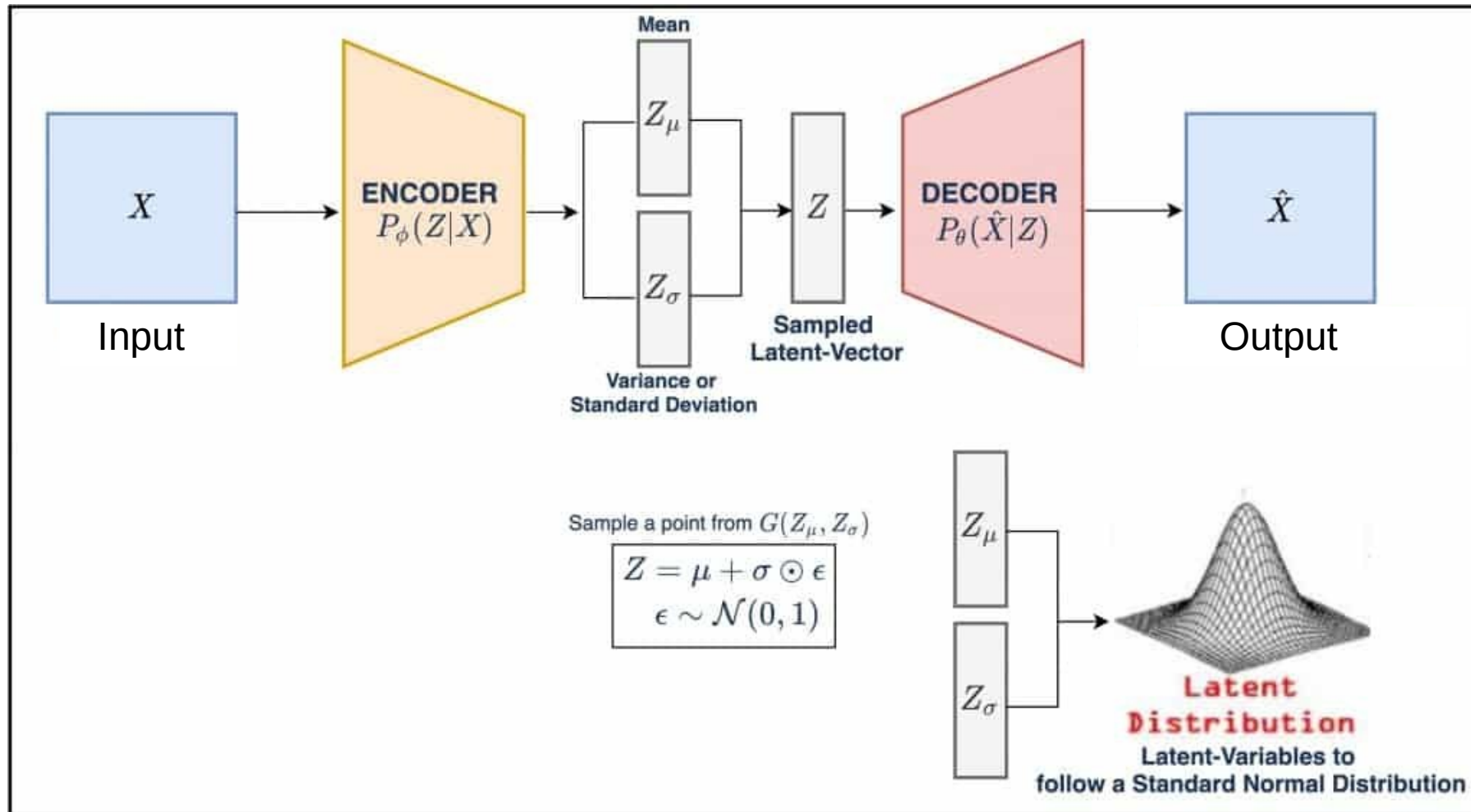


If you know the values A , you can reconstruct all the x with the weights on the right. Same weights for all X s! All the variability is now in the A s

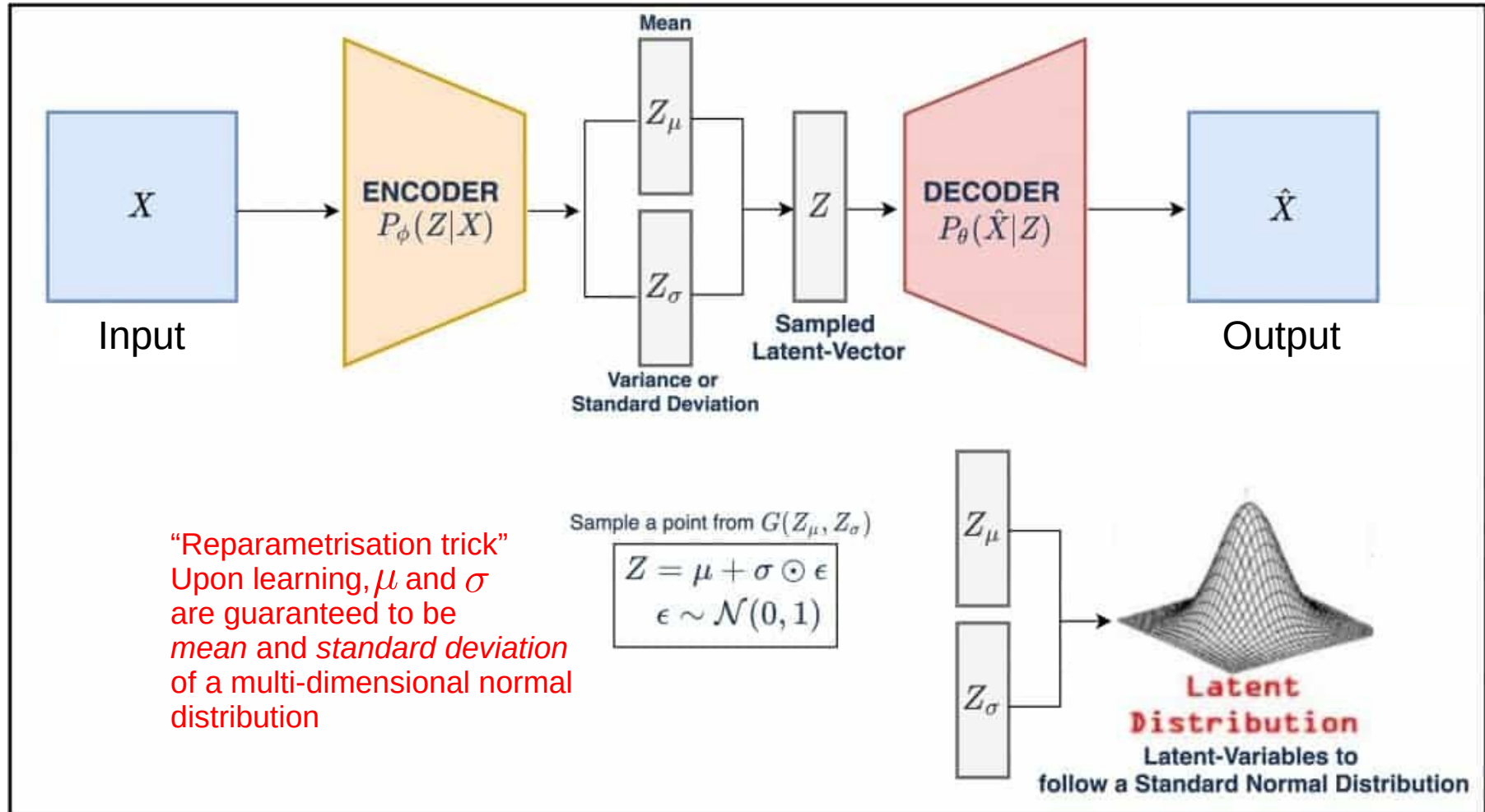
Linear vs nonlinear dimensionality reduction



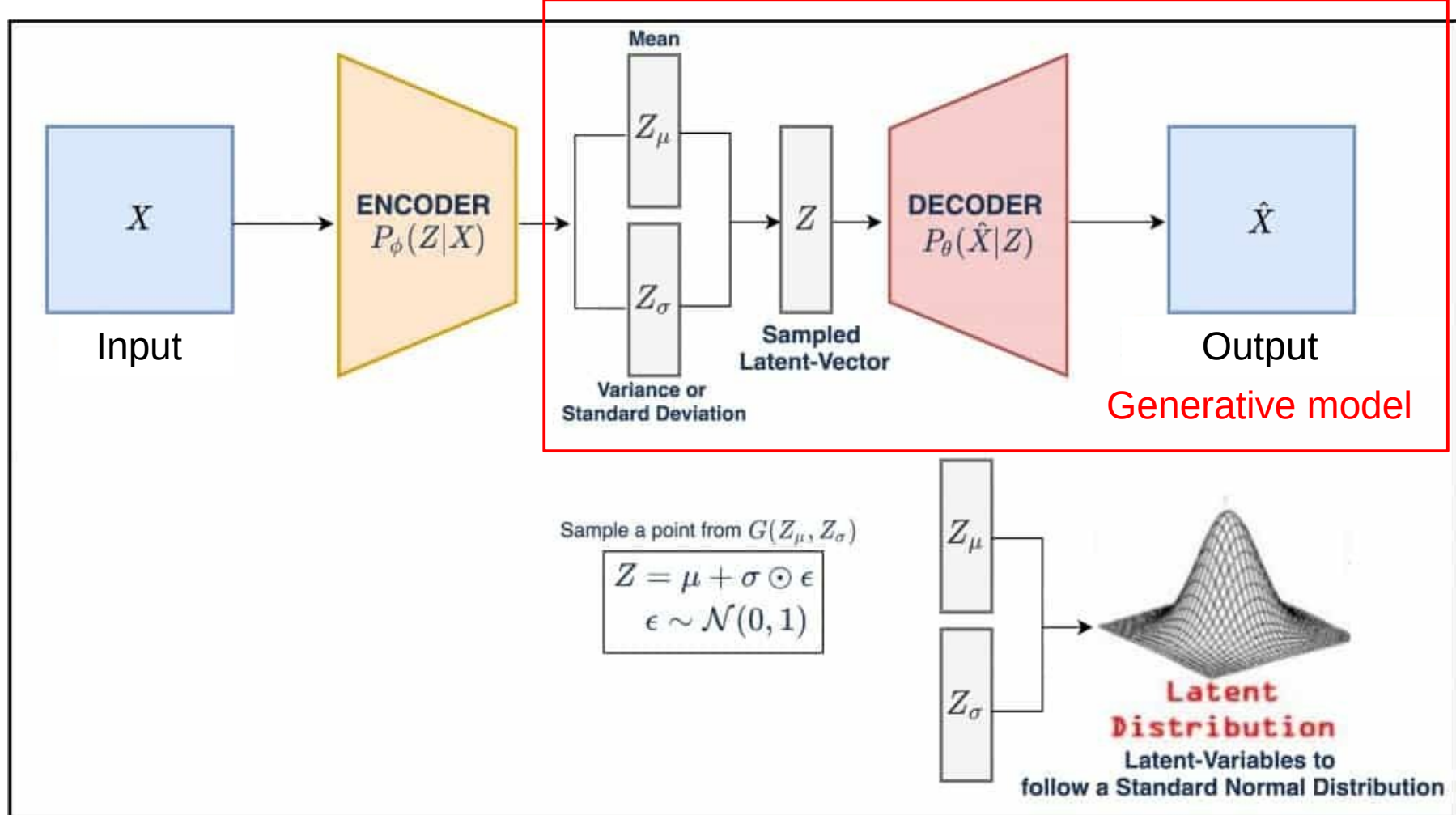
Variational Auto Encoder (VAE)



Variational Auto Encoder (VAE)

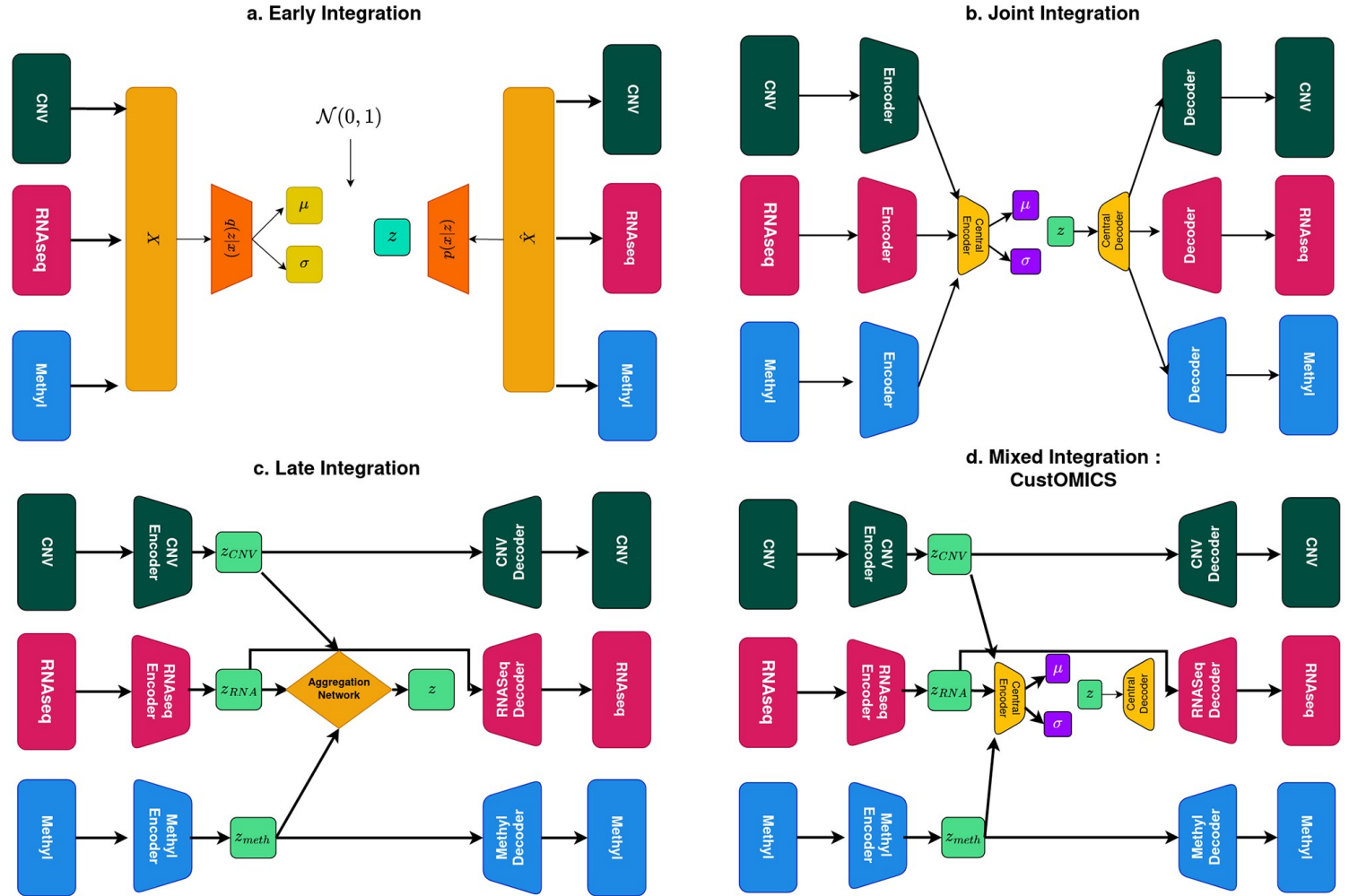


Variational Auto Encoder (VAE)

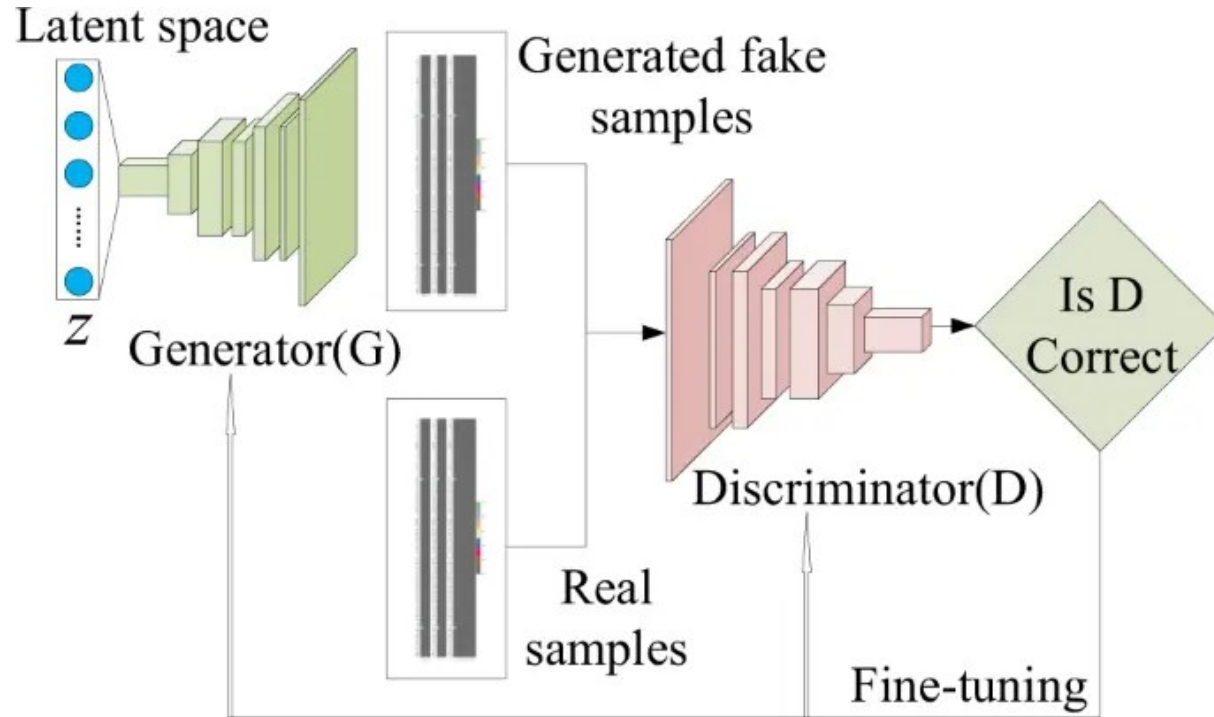


CustOMICS

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.



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  

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

Debaditya Mukherjee, Pritam Saha, Dmitry Kaplun , Aleksandr Sinitca & Ram Sarkar

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



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

[Home](#) > [Proceedings of International Conference on Artificial Intelligence and Applications](#) >
Conference paper

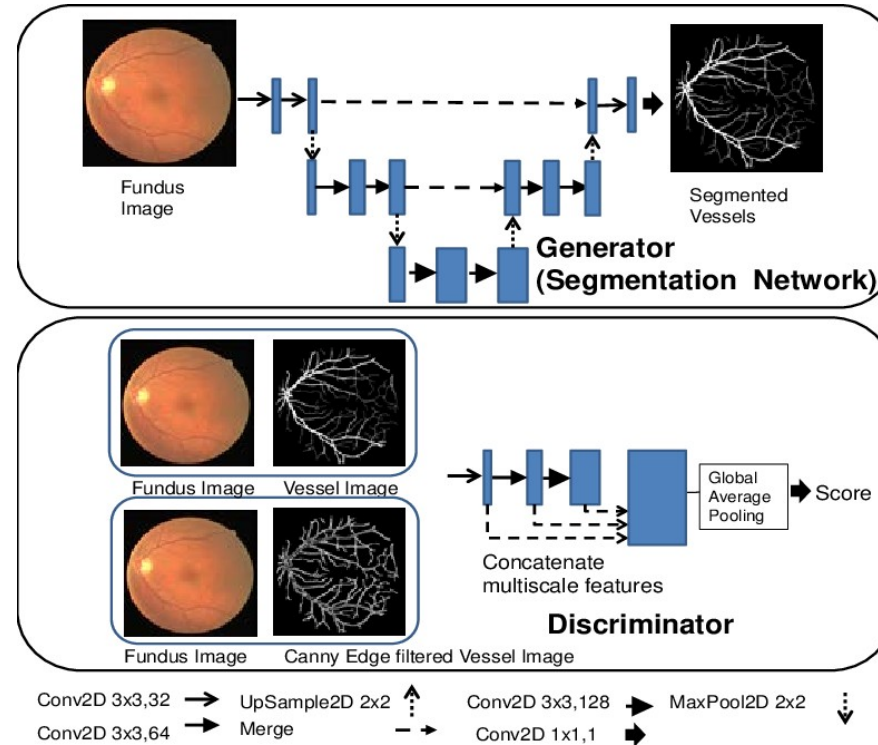
MR Image Synthesis Using Generative Adversarial Networks for Parkinson's Disease Classification

Conference paper | First Online: 02 July 2020

pp 317–327 | [Cite this conference paper](#)

Sukhpal Kaur , Himanshu Aggarwal & Rinkle Rani

GAN for segmentation



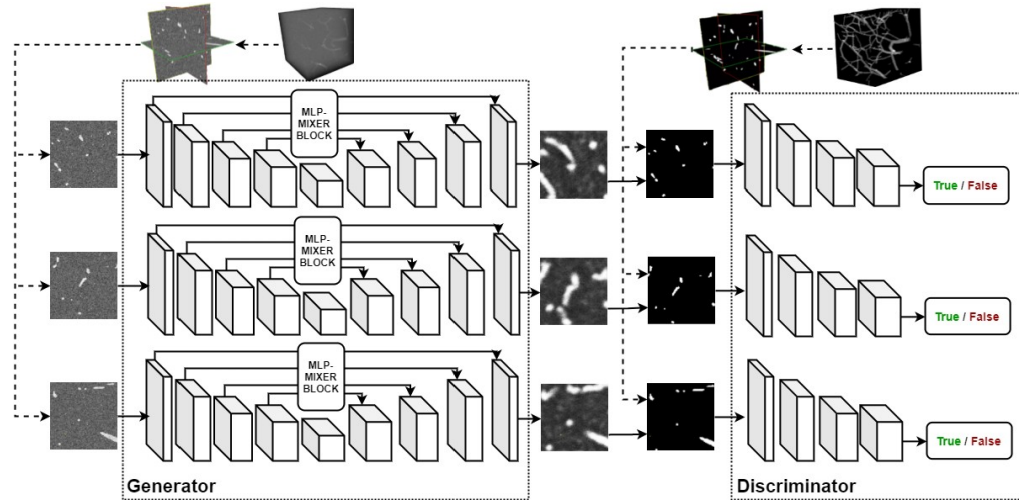
Tjio, G., Li, S., Xu, X., Ting, D.S.W., Liu, Y., Goh, R.S.M. (2019).

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

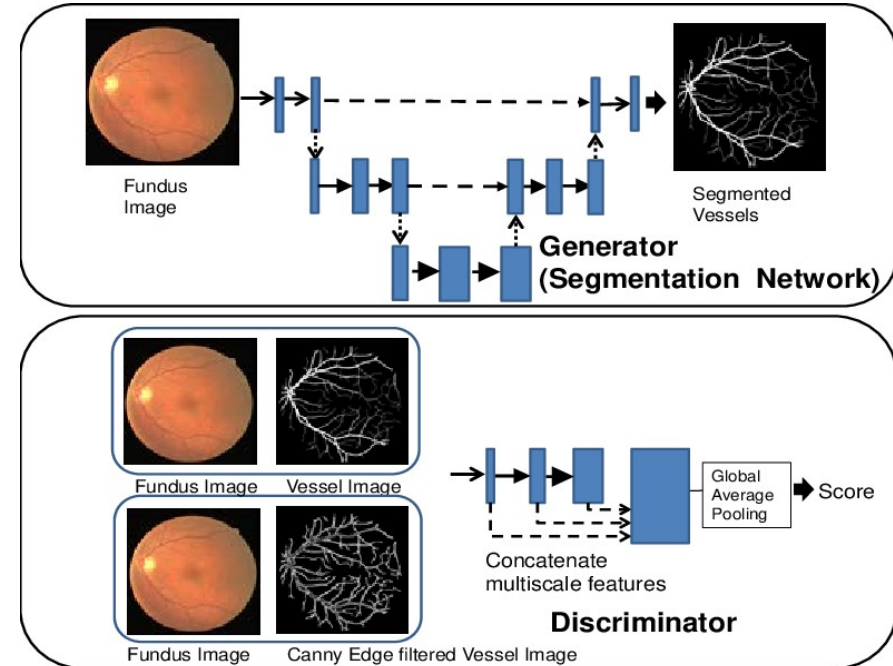
GAN can be built out of any DL architecture



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

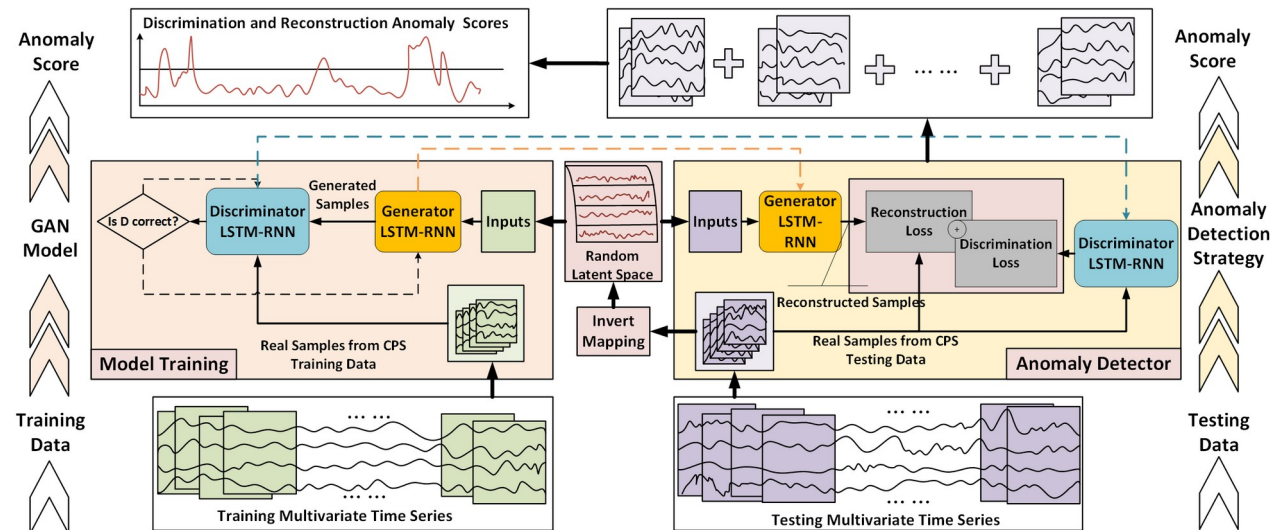
MLP-GAN for Brain Vessel Image Segmentation

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



Conv2D 3x3,32 → UpSample2D 2x2
Conv2D 3x3,64 → Merge
Conv2D 3x3,128 → MaxPool2D 2x2
Conv2D 1x1,1 →

GAN can be built out of any DL architecture



ViTGAN: Training GANs with Vision Transformers

Kwonjoon Lee¹ Huiwen Chang² Lu Jiang² Han Zhang²

Zhuowen Tu¹ Ce Liu²
¹UC San Diego ²Google Research
 {kw1042,ztu}@ucsd.edu {huiwenchang,lujiang,zhanghan,celiu}@google.com

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

Dan Li¹, Dacheng Chen¹, Lei Shi¹, Baihong Jin², Jonathan Goh³, and See-Kiong Ng¹

