Introduction to Machine Learning

Lectures On Modern Scientific Programming

Wigner Scientific Computing Laboratory

14-15 11 2022

GÁBOR BÍRÓ
biro.gabor@wigner.hu

Outline

What is Machine Learning? – a brief historical overview

A modern motivation

The main ingredients

Popular architectures

Hands-on: build a MLP from scratch (and more)

History

1950: Alan Turing creates the “Turing Test”

1957: Frank Rosenblatt: the first neural network for computers (the **perceptron**), which simulate the thought processes of the human brain.

1959: Arthur Samuel, IBM: *Machine Learning*

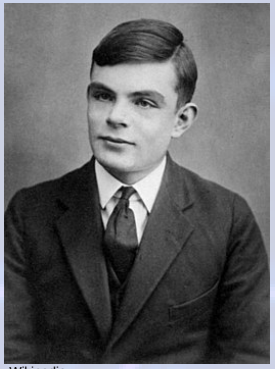
1967: The first general, working learning algorithm for supervised, deep, feedforward, multilayer perceptrons by A. G. Ivakhnenko and V. G. Lapa

1986: First mention of *Deep Learning* by Rina Dechter (*Learning While Searching in Constraint-Satisfaction Problems*)

1989: Yann LeCun et al: standard backpropagation algorithm for recognizing handwritten ZIP codes on mail

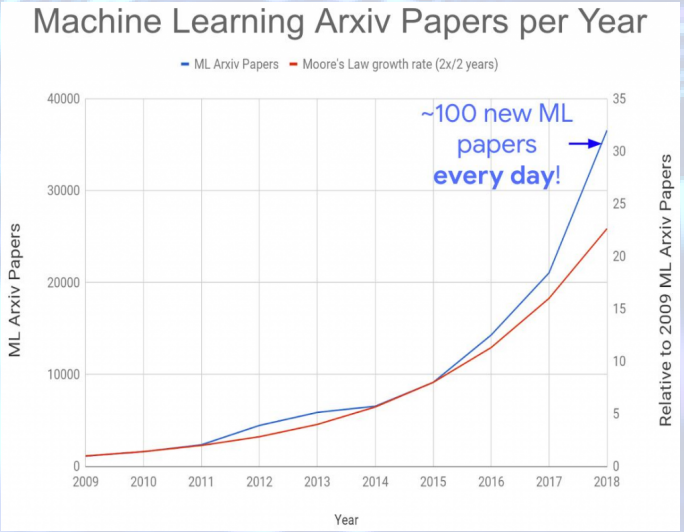
1997: “*A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P** if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.*” - Tom M. Mitchell: Machine Learning

1997: IBM’s Deep Blue beats Garri Kaszparov (the world champion at chess). Computing capacity: 11.38 GFLOPS, TOP500: 259th (comparison: Nvidia RTX 4090: 82.6 TFLOPS)



Wikipedia

https://researcher.watson.ibm.com/researcher/view_page.php?id=6814



arXiv:1911.05289

History

2009: ImageNet by prof. Fei-Fei Li a database of 14 million labeled images in 2009

2011: IBM's Watson: winner of game show Jeopardy!

2011: Google Brain: cats in Youtube videos

2012: AlexNet by Alex Krizhevsky: first CNN

2013: Word2vec algorithms: foundations for language models

2014: DeepFace by Facebook

2014: Generative adversarial networks (GAN) by Ian Goodfellow

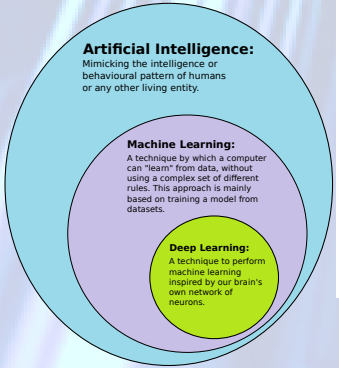
2016: AlphaGo by Deepmind

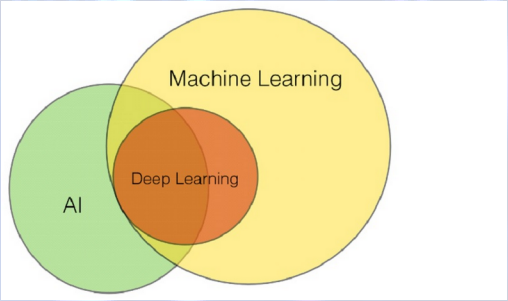
2016: Face2Face (baseline for 'DeepFake')

2017: Waymo: first self-driving car company to operate without human intervention

2018: AlphaFold by Deepmind

2020: GPT-3 by OpenAI to generate human-like text. Trainable parameters: **175 billion**





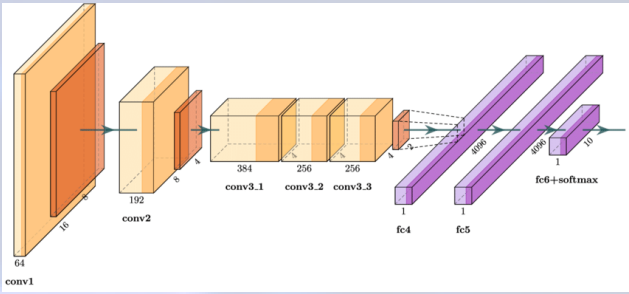
Wikipedia

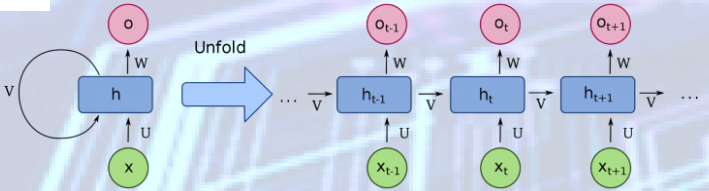
History

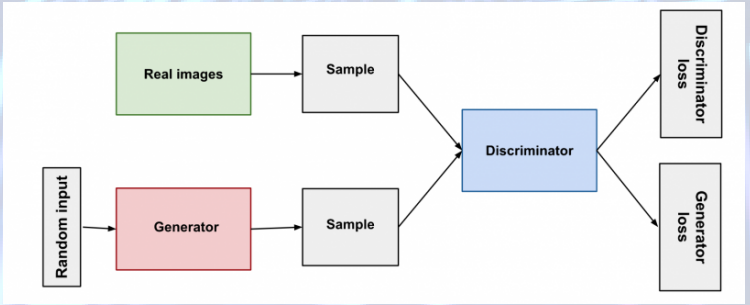
CNN (image classification, object detection, recommender systems)...

Recurrent/recursive neural networks (RNNs): Sequence modeling, next word prediction, translating sounds to words, human language translation...

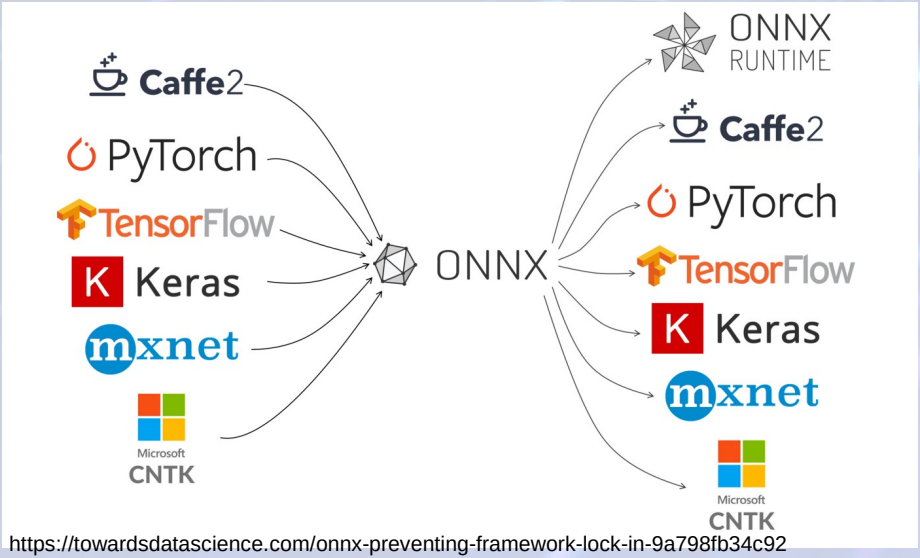
Generative models: anomaly detection, pattern recognition, reinforced learning







Various frameworks for training and inference:



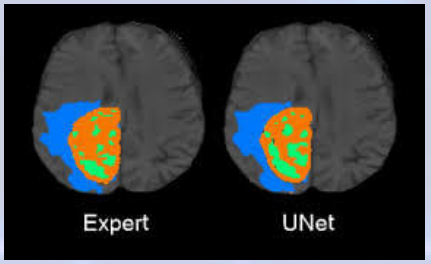
https://towardsdatascience.com/onnx-preventing-framework-lock-in-9a798fb34c92

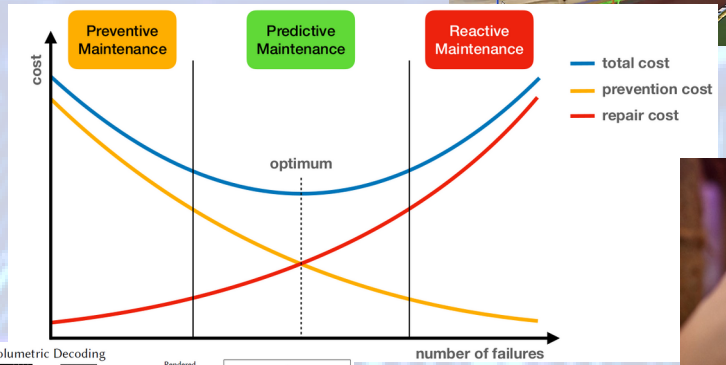
Motivaton - data, data, more data

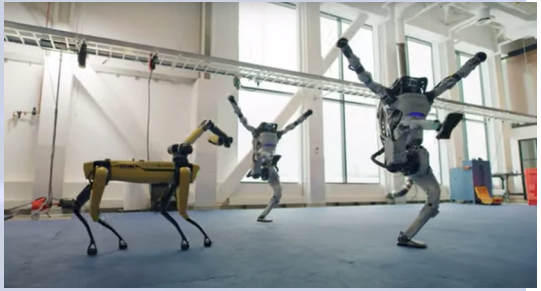
Autonomous driving
Medical imaging
Predictive maintenance
Anomaly detection, fake news detection
Search of BSM physics
Stock price prediction
Natural Language Processing
Virtual Assistants
Virtual reality
Colorization of Black and White Images
Content generation, examples:

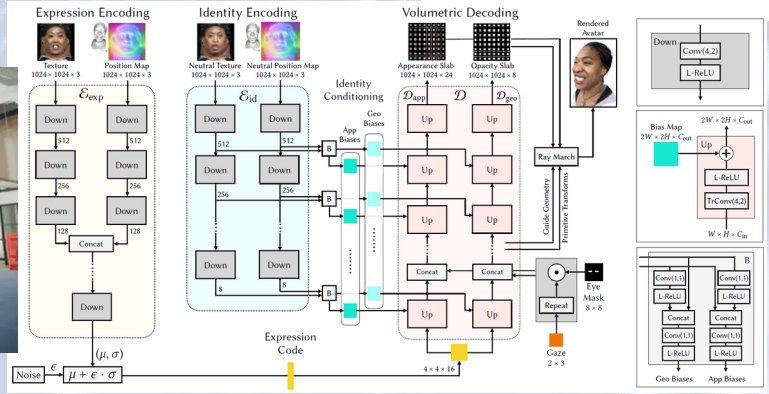
https://infiniteconversation.com/
https://huggingface.co/spaces/stabilityai/stable-diffusion

Robotics
...

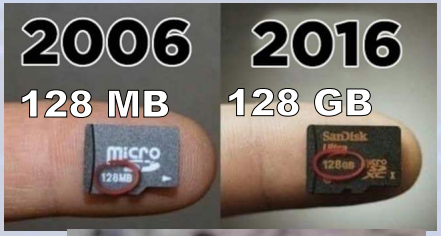








Motivaton - data, data, more data

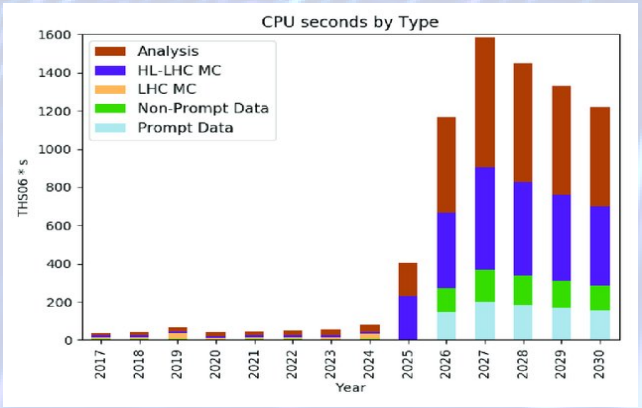


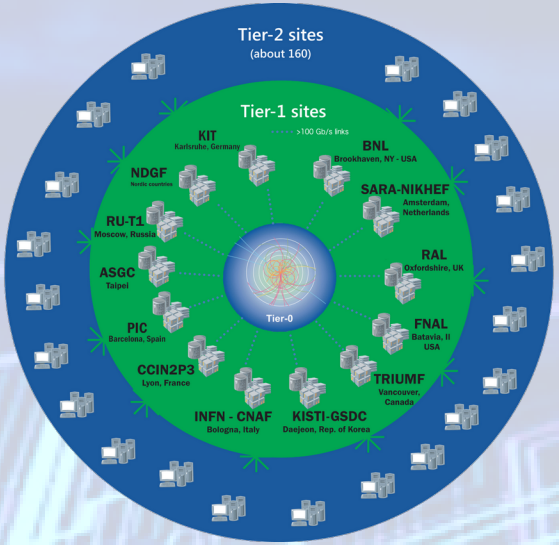
LHC in numbers: **2013** and **now:**

Data:	15 PB/év	vs	200+ PB/year
Tape:	180 PB	vs	740+ PB
Disk:	200 PB	vs	570+ PB
HS06:	2M	vs	100+ B

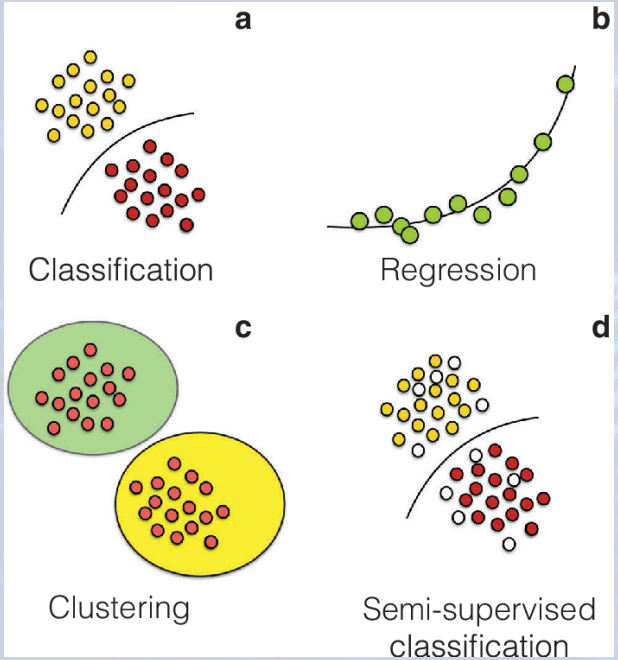
Storing and distributing the data is only one side of the challange

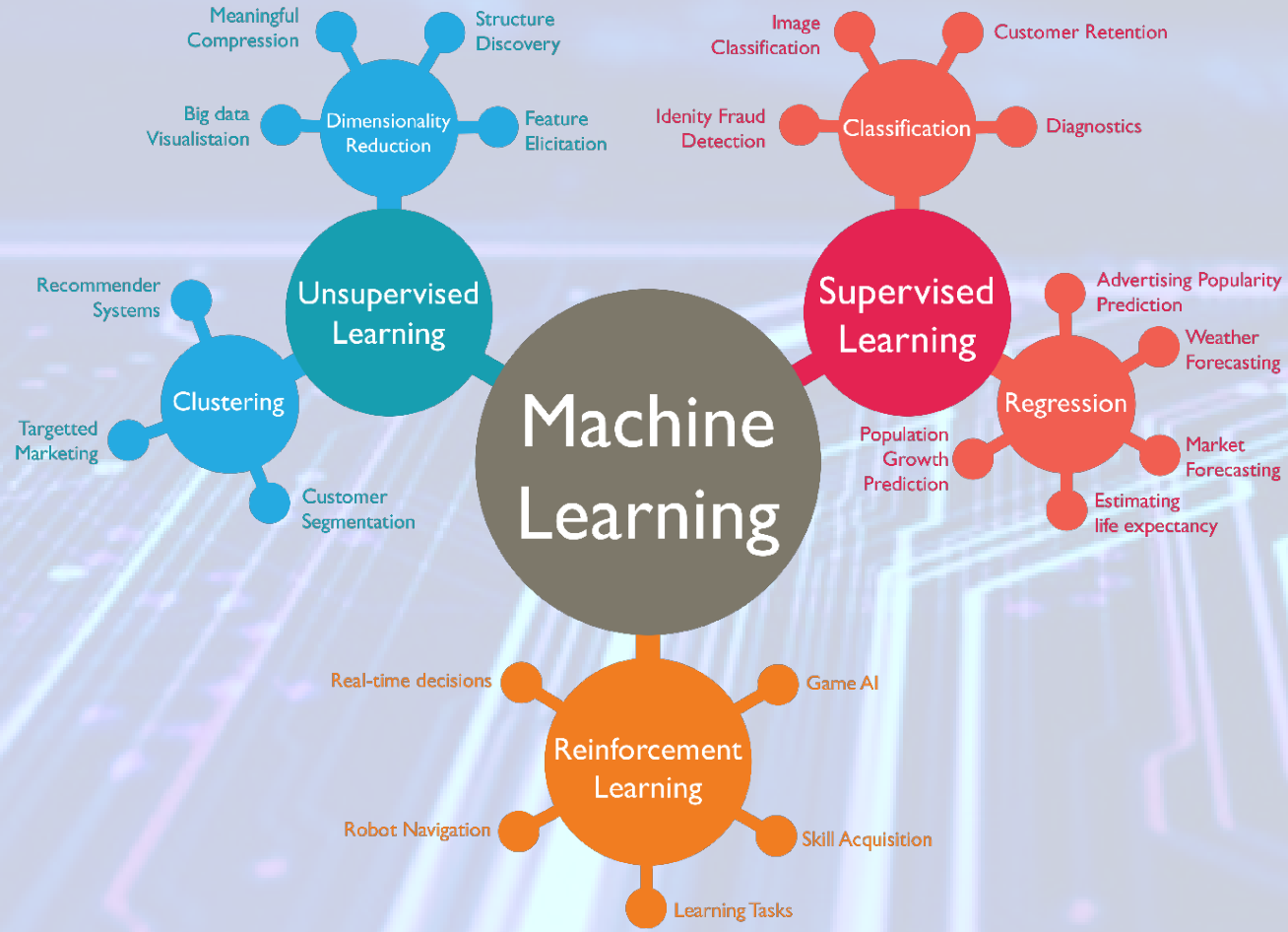
→ analysis, simulations





Approaches

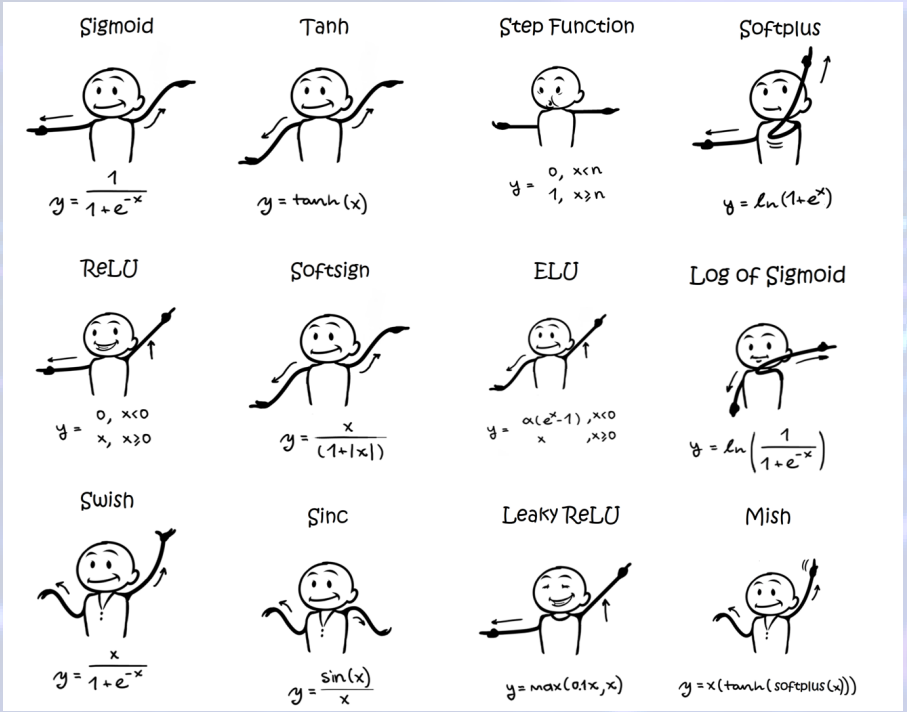

a
b
Classification
Regression
c
d
Clustering
Semi-supervised
classification

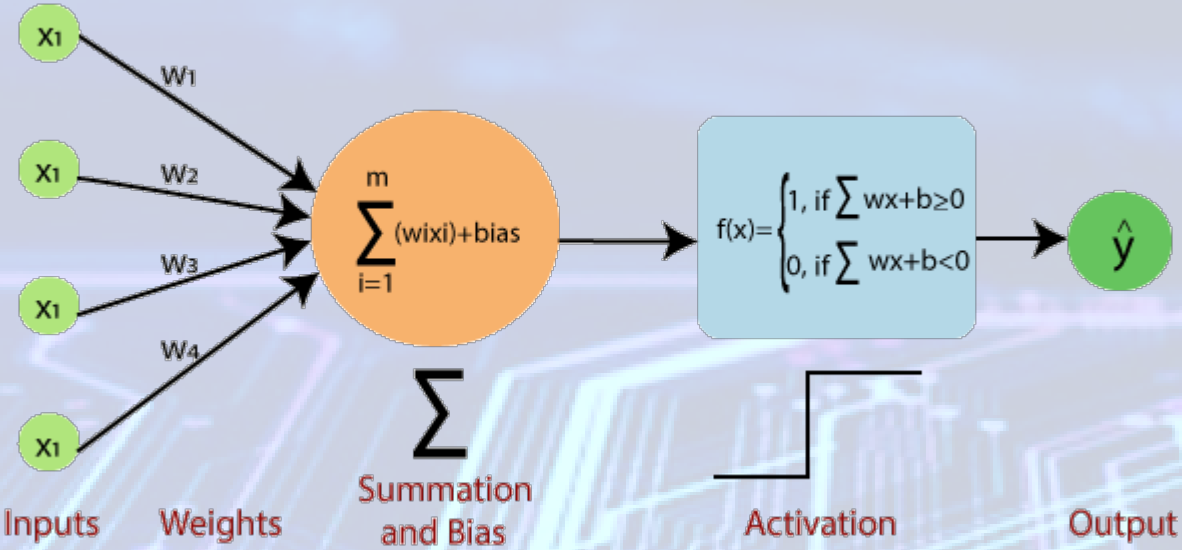

Machine Learning
Unsupervised Learning
Meaningful Compression
Structure Discovery
Dimensionality Reduction
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Image Classification
Customer Retention
Classification
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Market Forecasting
Estimating life expectancy
Population Growth Prediction
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

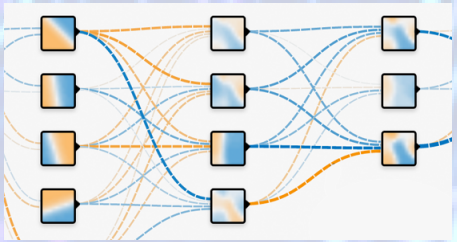
Main ingredients

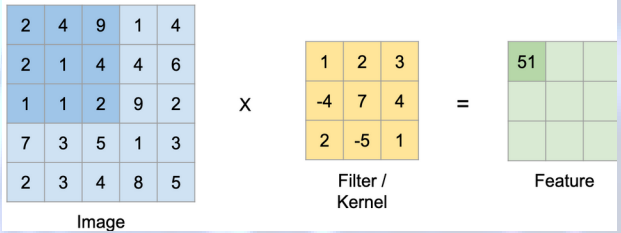
Perceptrons:

- Input value(s)
- Weight: the connection between the units
- Bias: the intercept added in a linear equation
- Activation Function





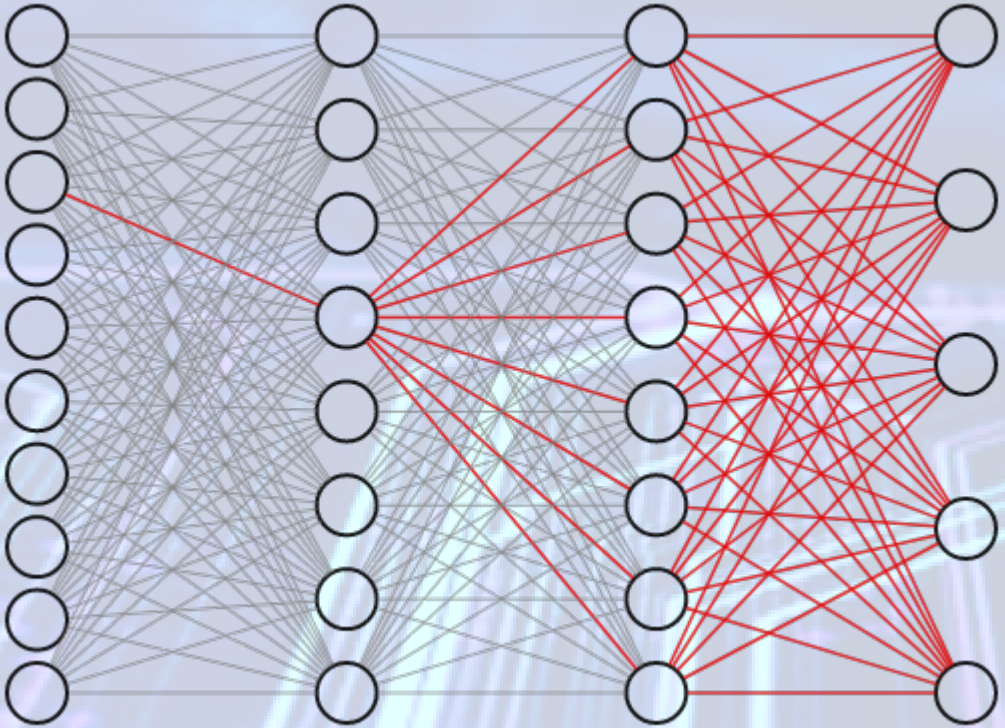




Other important components: pooling layers, regularization and normalization, recurrent layers...

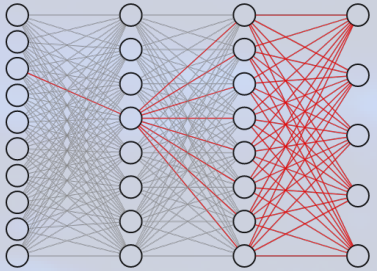
https://sefiks.com/2020/02/02/dance-moves-of-deep-learning-activation-functions/

Main ingredients



Main ingredients

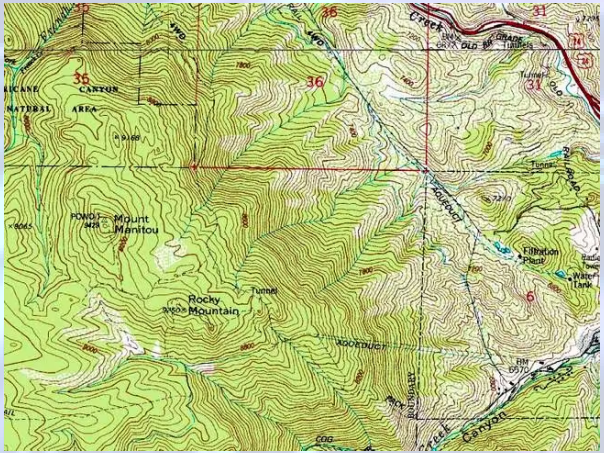
The *learning* part: optimizing "somehow" the weights (Curse of dimensionality)



Loss function: $\mathcal{L} = \frac{1}{n}\sum_i (y_i - f(x_i))^2 := \frac{1}{n}\sum_i (y_i - (mx_i + b))^2$

The gradient descent method:

1) Start with random weights
2) Evaluate the loss
3) Figure out which direction the loss function steeps downward the most (with respect to changing the parameters)
4) Repeat from 2)



Gradient of the loss function with respect to all of the parameters

$$\frac{\partial \mathcal{L}}{\partial m} = \frac{2}{n}\sum_i x_i \cdot (y_i - (mx_i + b))$$

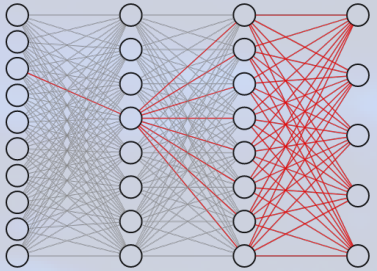
$$\frac{\partial \mathcal{L}}{\partial b} = \frac{2}{n}\sum_i (y_i - (mx_i + b))$$

$$\Rightarrow \quad m := m - \alpha \cdot \frac{\partial \mathcal{L}}{\partial m}$$

$$b := b - \alpha \cdot \frac{\partial \mathcal{L}}{\partial b}$$

Main ingredients

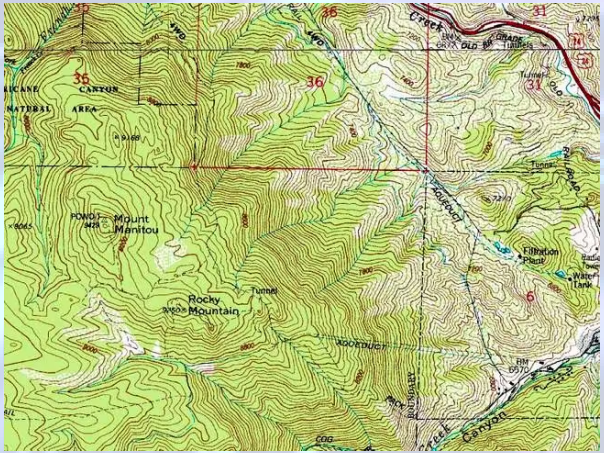
The *learning* part: optimizing "somehow" the weights (Curse of dimensionality)



Loss function: $\mathcal{L} = \frac{1}{n}\sum_i (y_i - f(x_i))^2 := \frac{1}{n}\sum_i (y_i - (mx_i + b))^2$

The gradient descent method:

1) Start with random weights
2) Evaluate the loss
3) Figure out which direction the loss function steeps downward the most (with respect to changing the parameters)
4) Repeat from 2)



Gradient of the loss function with respect to all of the parameters

$$\frac{\partial \mathcal{L}}{\partial m} = \frac{2}{n}\sum_i x_i \cdot (y_i - (mx_i + b))$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{2}{n}\sum_i (y_i - (mx_i + b))$$

$$m := m - \alpha \cdot \frac{\partial \mathcal{L}}{\partial m}$$

$$b := b - \alpha \cdot \frac{\partial \mathcal{L}}{\partial b}$$

MORE ON THIS DURING THE HANDS-ON SESSION!

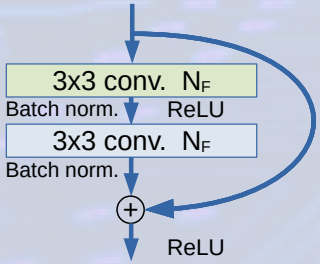
Popular architectures

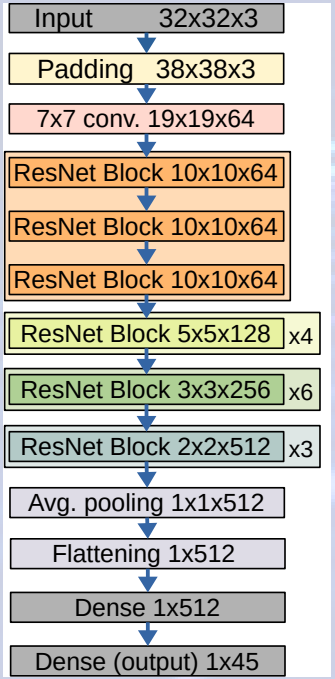
Stacking more layers: solve complex problems more efficiently, get highly accurate results

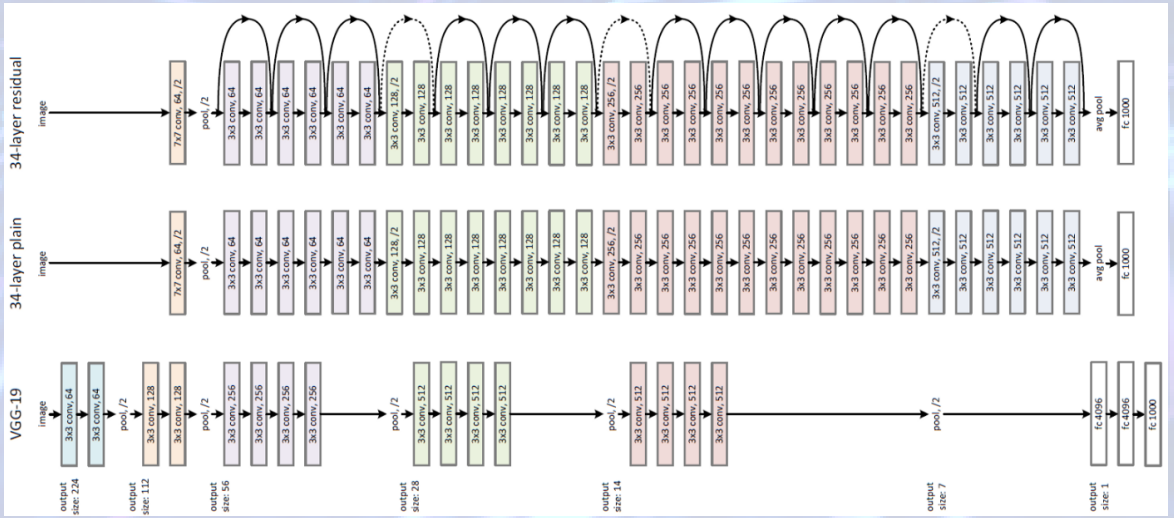
BUT:

Vanishing/exploding gradients

ResNet: **Residual blocks with "skip connections"** (SOTA image classifier of 2015)

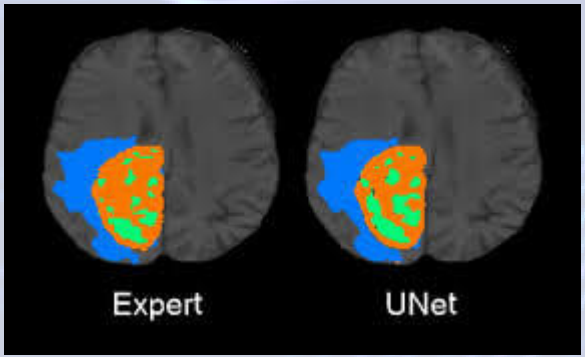


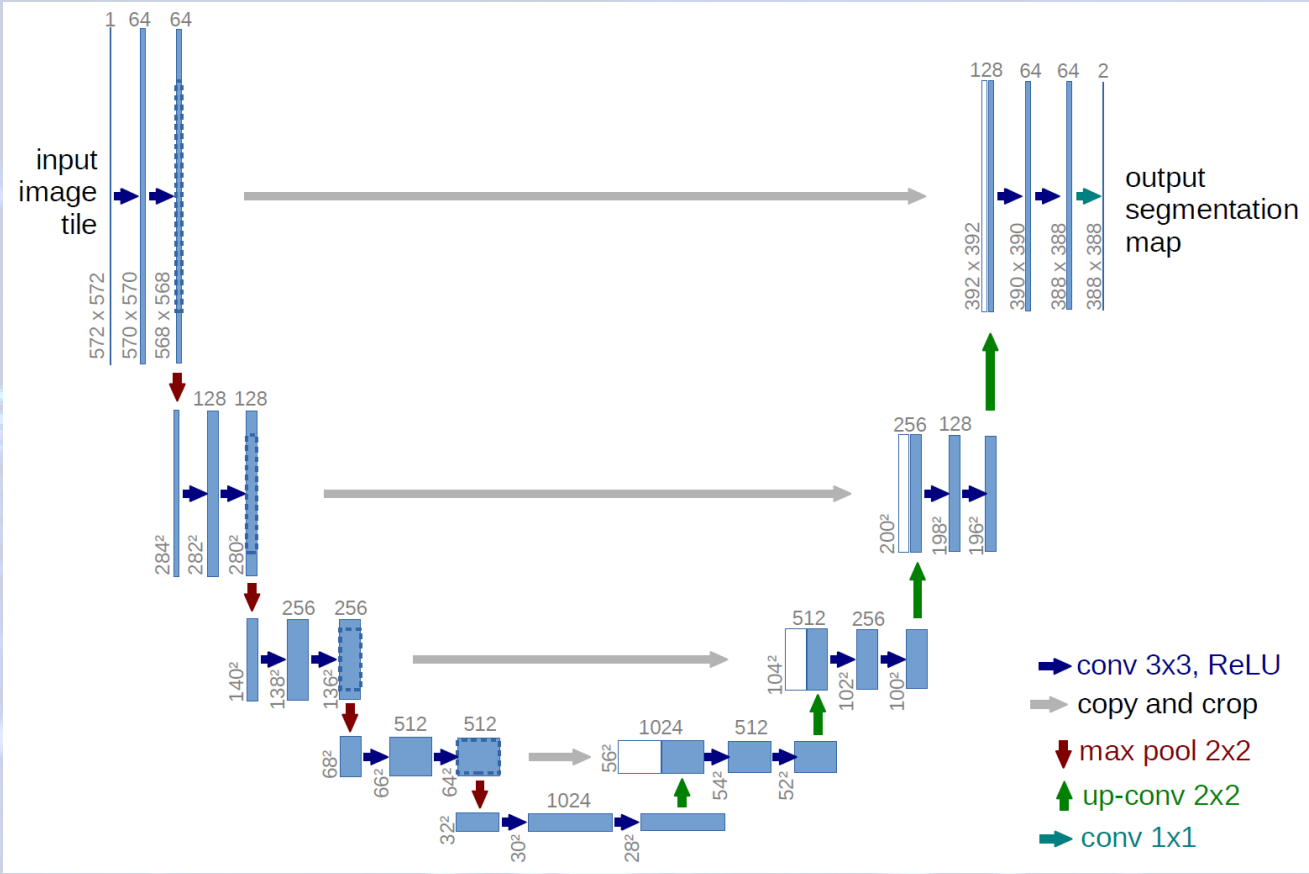




Popular architectures

U-Net: biomedical image segmentation





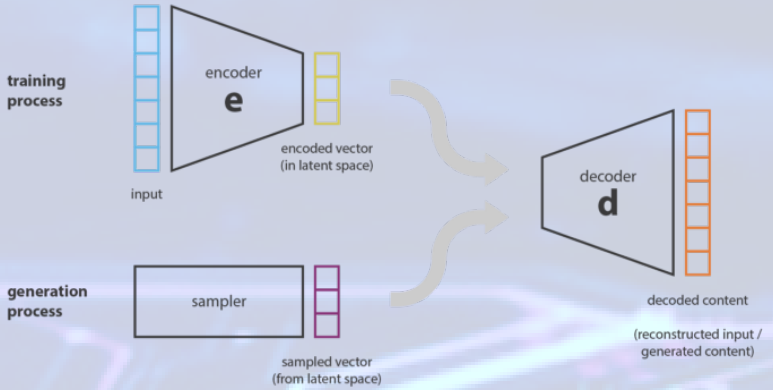
Popular architectures

(Conditional (Variational)) autoencoders

Dimension reduction

Denoising data

Latent space conditioning



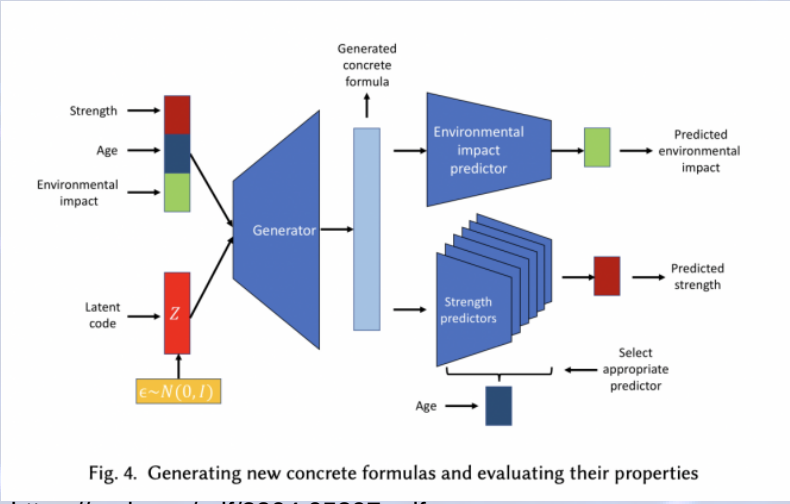
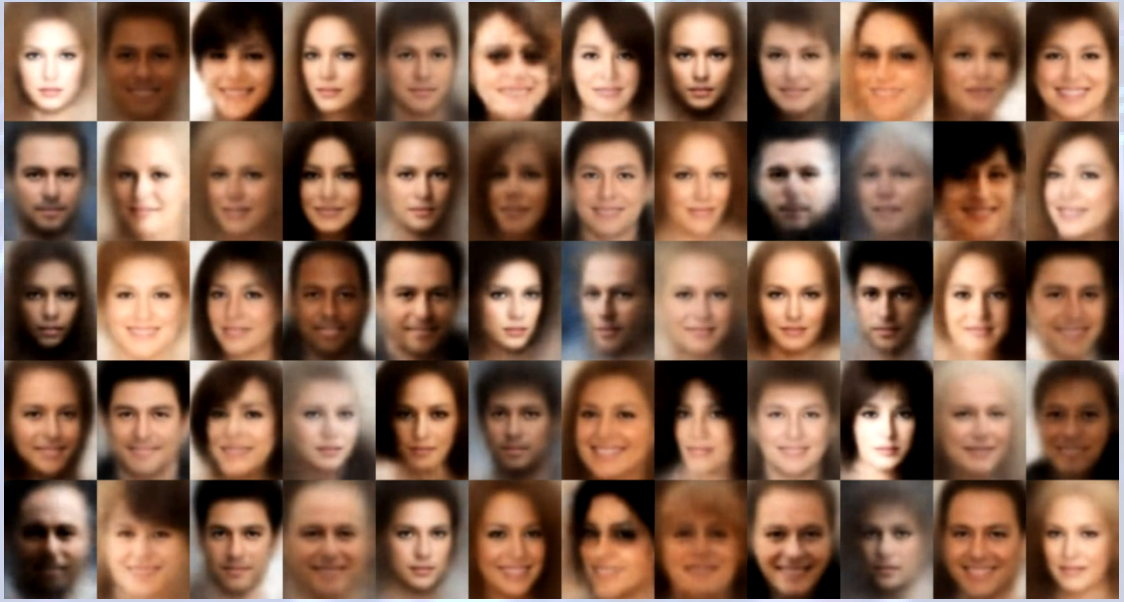


Fig. 4. Generating new concrete formulas and evaluating their properties

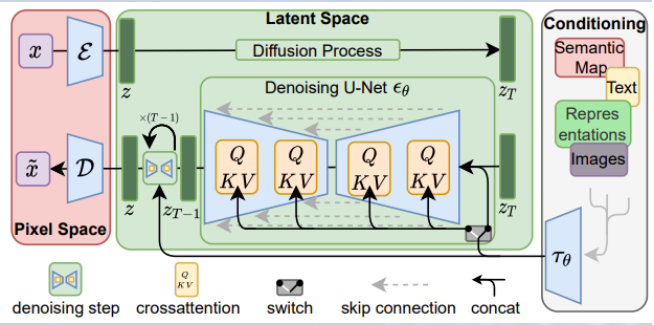
https://arxiv.org/pdf/2204.05397.pdf

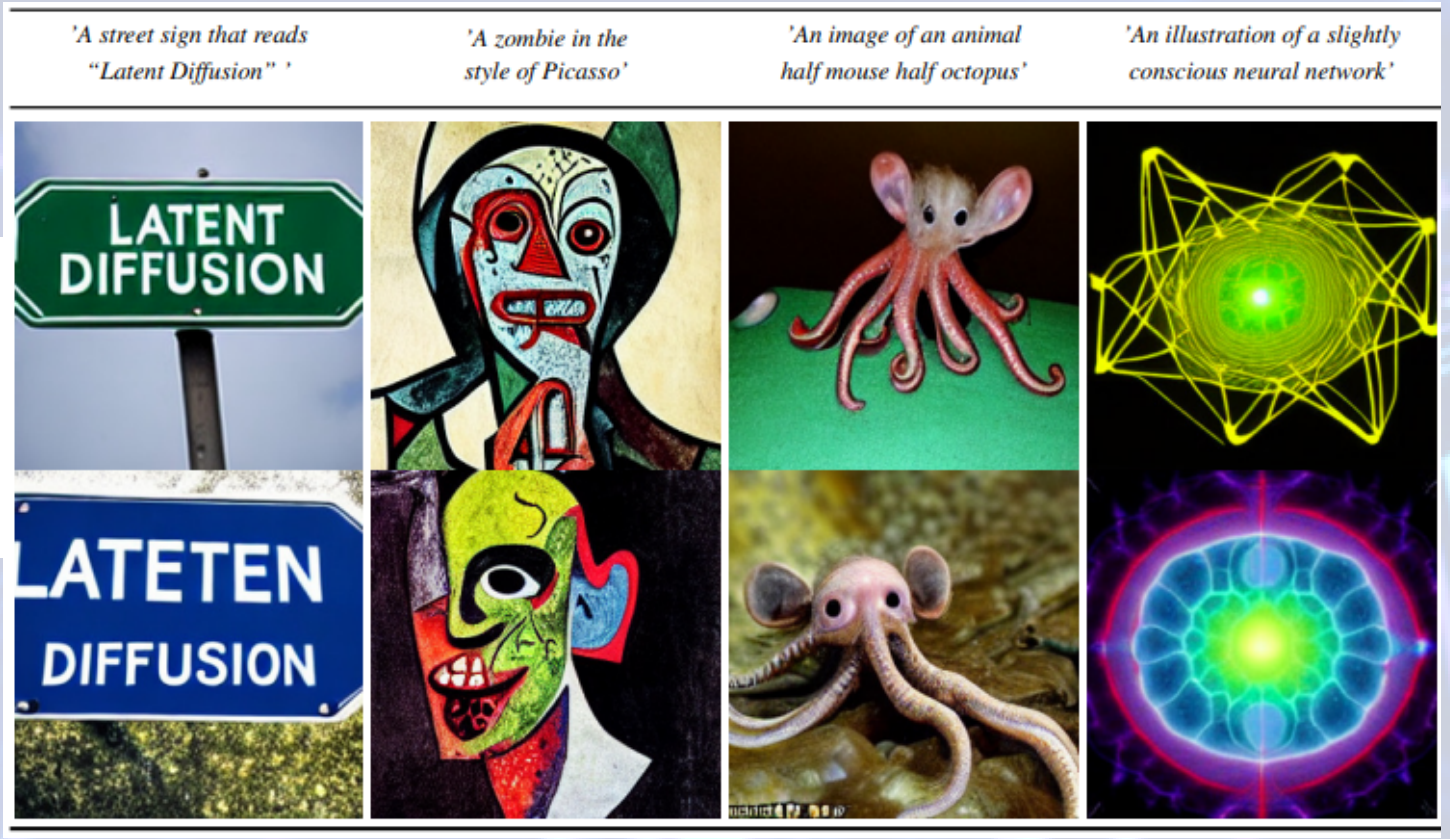


Popular architectures

Diffusion models: https://huggingface.co/spaces/stabilityai/stable-diffusion

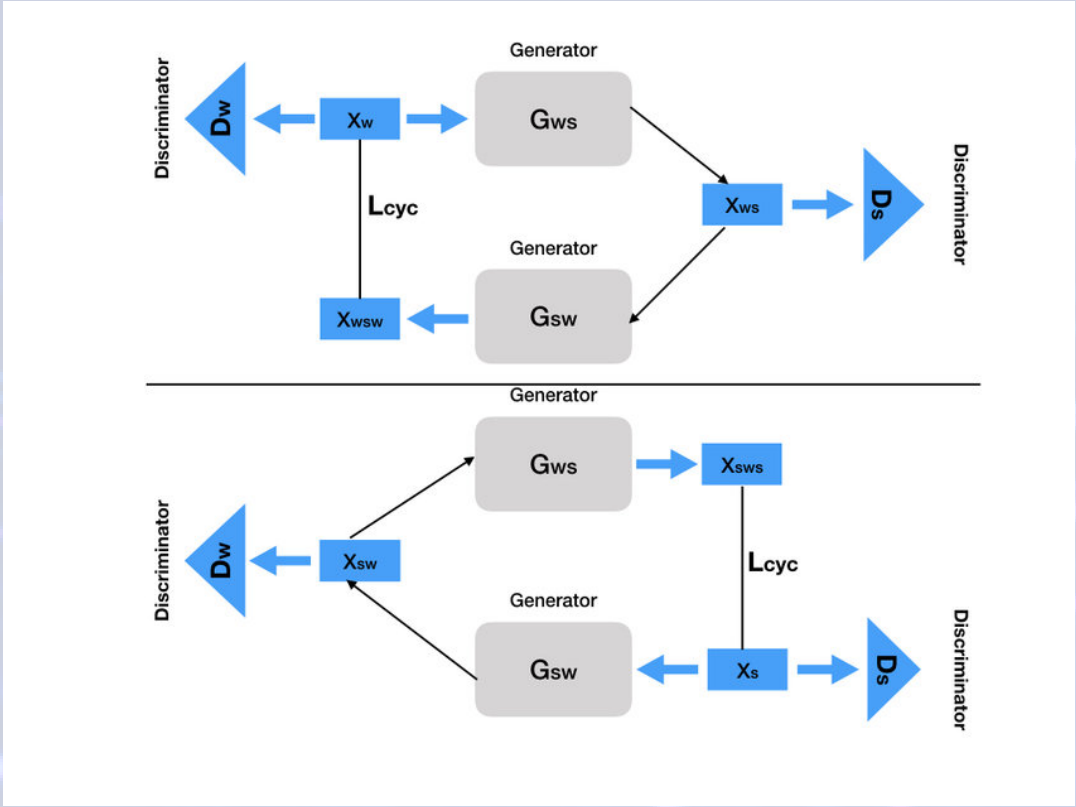
Gradually perturbate he input data over several steps by adding Gaussian noise





Popular architectures

GAN: data generation via competing generator-discriminator



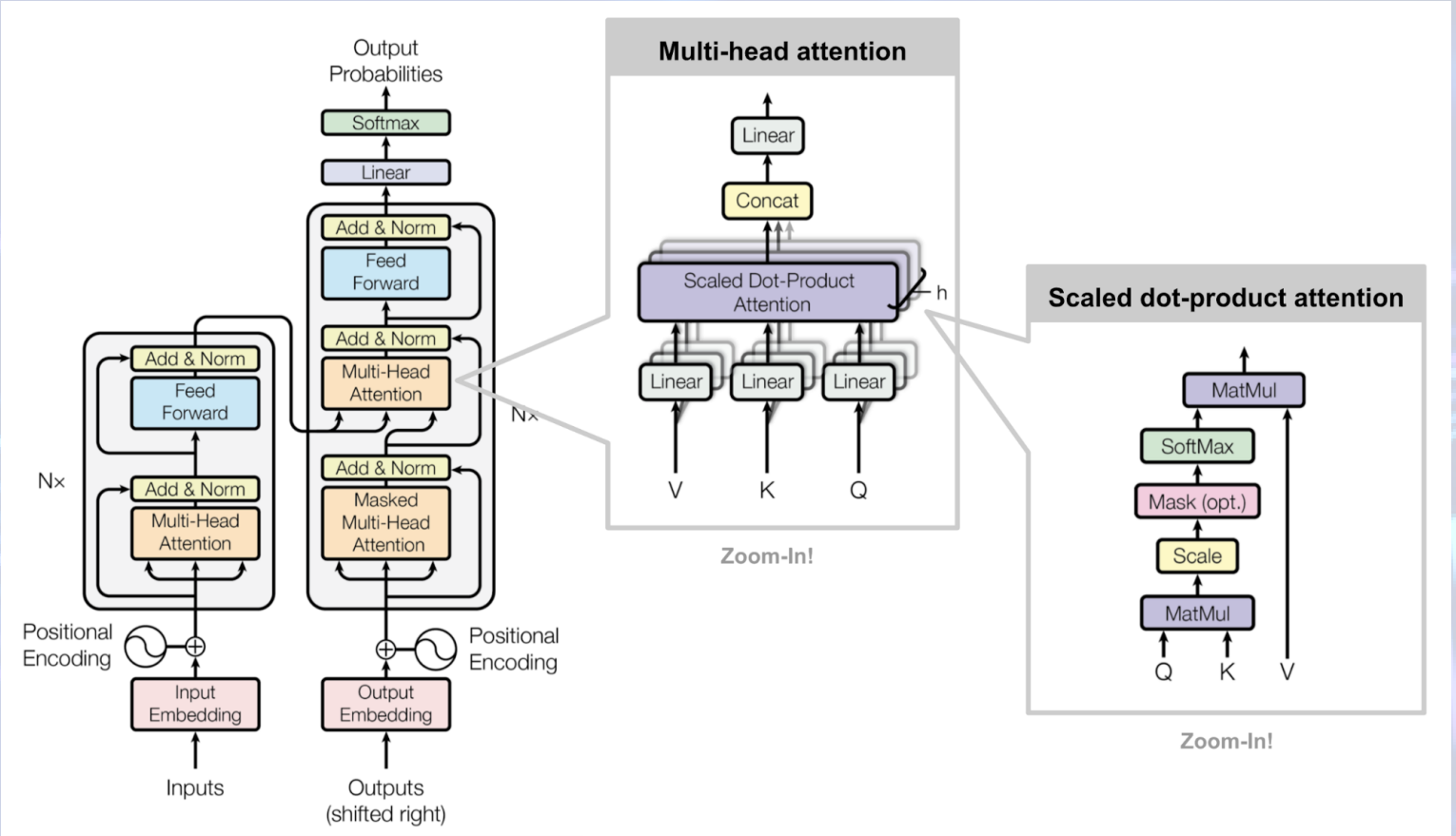
Popular architectures

GAN: data generation via competing generator-discriminator

Popular architectures

Attention and Transformers :

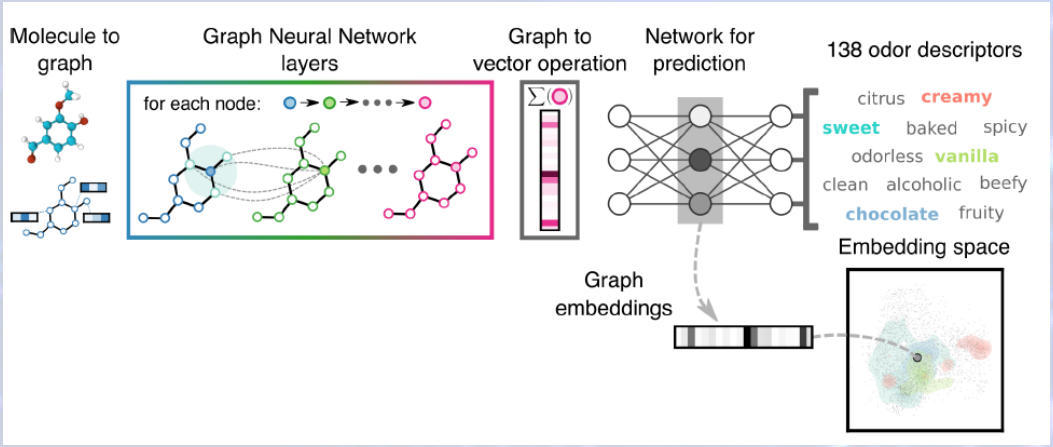
A revolution in natural language processing

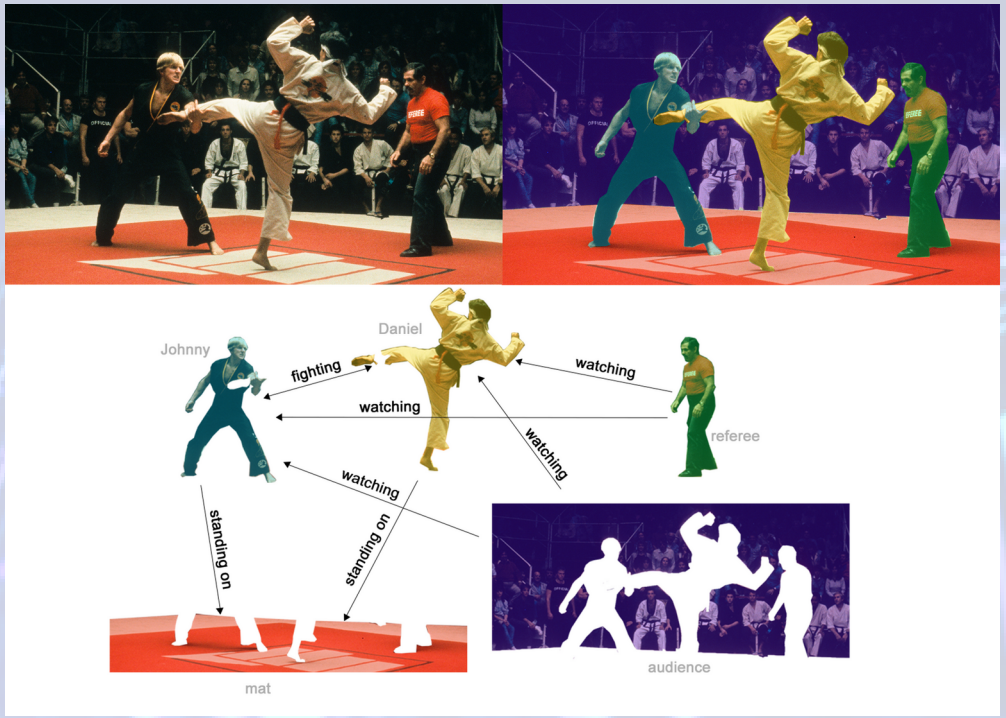


https://arxiv.org/abs/1706.03762

Popular architectures

Graph Neural Networks





Hands-on

Get the Jupyter notebook download links from the indico page:

https://indico.wigner.hu/event/1409/contributions/3284/

Or: https://gitlab.wigner.hu/biro.gabor/modernlectures22

Acknowledgement: OTKA K135515, NKFIH 2019-2.1.6-NEMZKI-2019-00011, 2020-2.1.1-ED-2021-00179, the Wigner Scientific Computing Laboratory and MILAB RRF-2.3.1-21-2022-00004.

Hands-on

Get the Jupyter notebook download links from the indico page:

https://indico.wigner.hu/event/1409/contributions/3284/

Or: https://gitlab.wigner.hu/biro.gabor/modernlectures22

https://www.youtube.com/watch?v=WhvMvbqWNYo

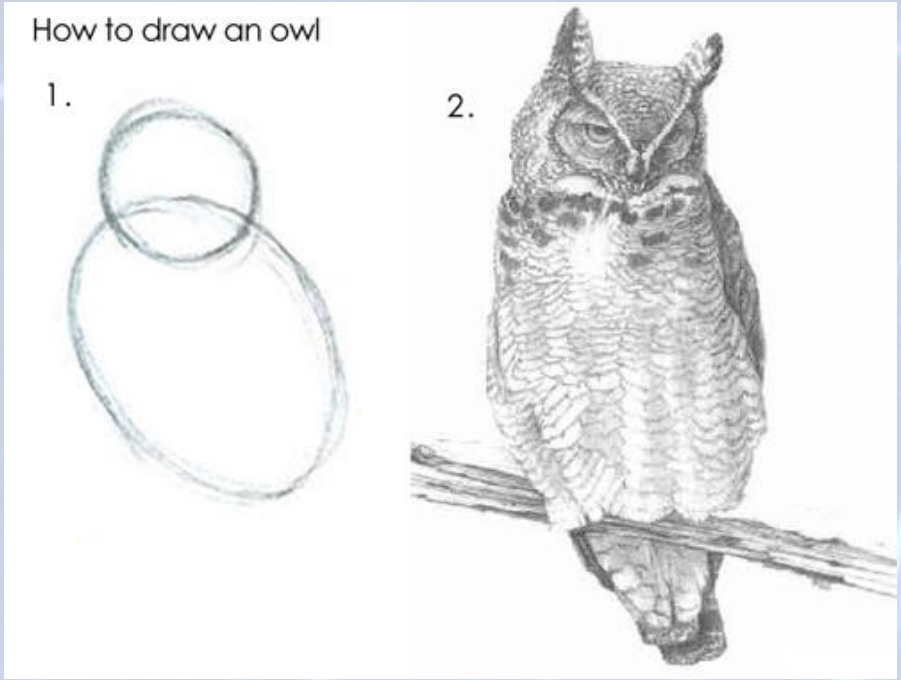
Acknowledgement: OTKA K135515, NKFIH 2019-2.1.6-NEMZKI-2019-00011, 2020-2.1.1-ED-2021-00179, the Wigner Scientific Computing Laboratory and MILAB RRF-2.3.1-21-2022-00004.

Hands-on

Get the Jupyter notebook download links from the indico page:

https://indico.wigner.hu/event/1409/contributions/3284/

Or: https://gitlab.wigner.hu/biro.gabor/modernlectures22



https://www.youtube.com/watch?v=WhvMvbqWNYo

Acknowledgement: OTKA K135515, NKFIH 2019-2.1.6-NEMZKI-2019-00011, 2020-2.1.1-ED-2021-00179, the **Wigner Scientific Computing Laboratory** and MILAB RRF-2.3.1-21-2022-00004.