



TECHNISCHE UNIVERSITÄT  
CHEMNITZ

# Neurocomputing

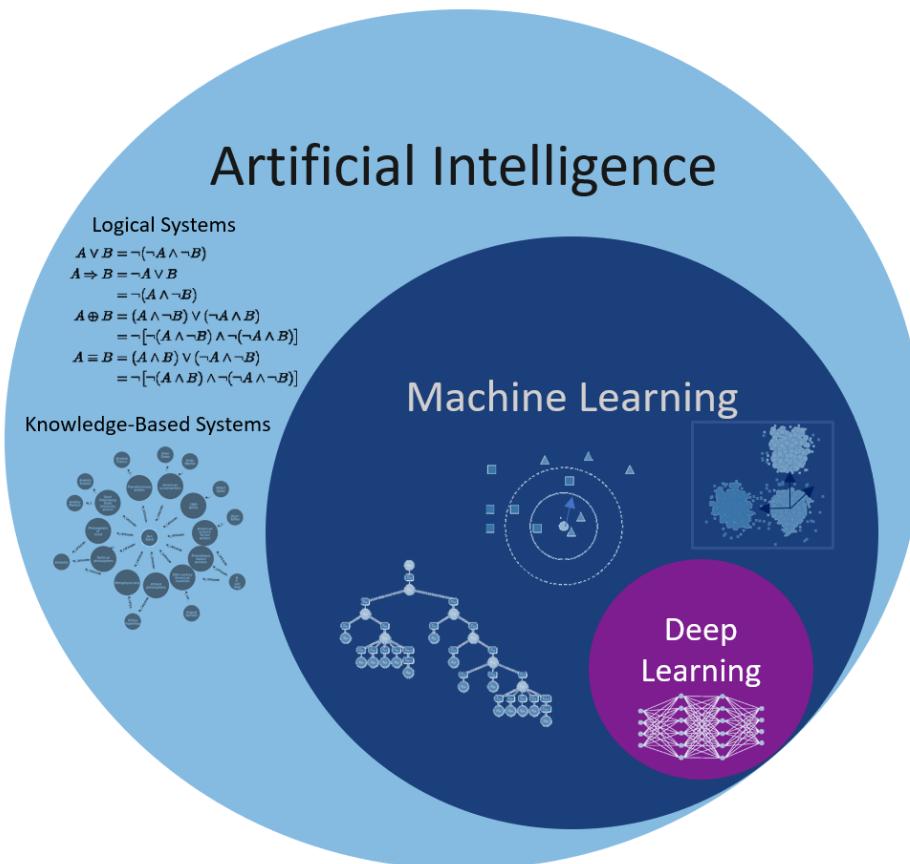
Introduction

Julien Vitay

Professur für Künstliche Intelligenz - Fakultät für Informatik

<https://tu-chemnitz.de/informatik/KI/edu/neurocomputing>

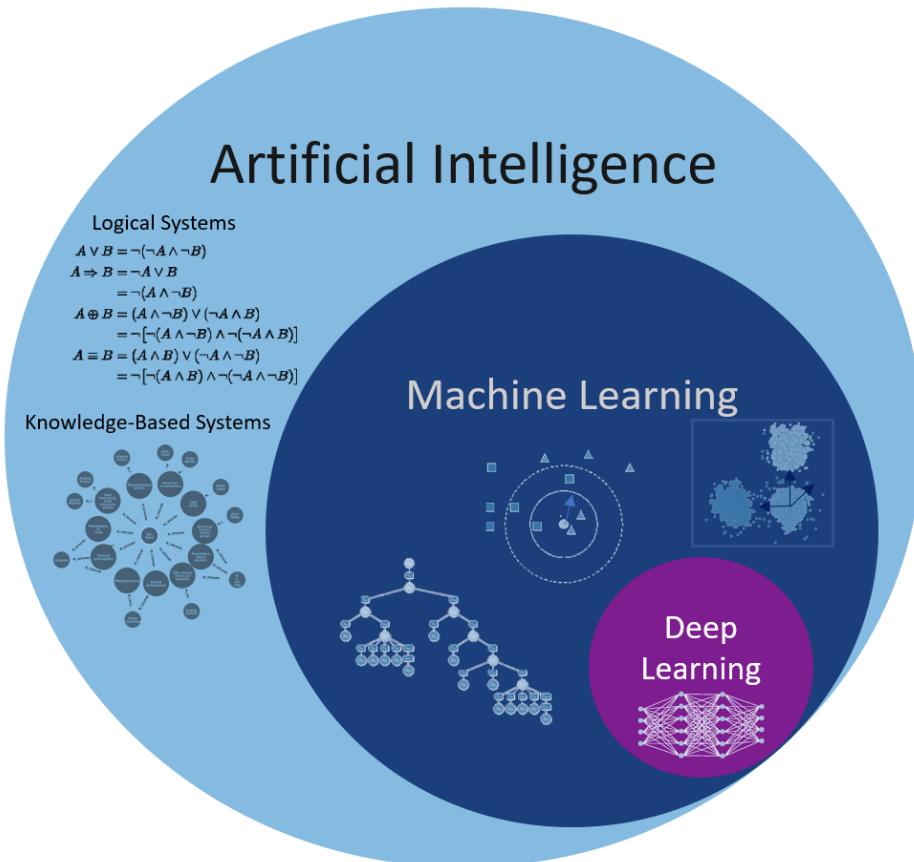
# Artificial Intelligence, Machine Learning, Deep Learning, Neurocomputing



- The term **Artificial Intelligence** was coined by John McCarthy at the Dartmouth Summer Research Project on Artificial Intelligence in **1956**.
- *"The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."*
- **Good old-fashion AI** approaches (GOFAI) were purely symbolic (logical systems, knowledge-based systems) or using linear neural networks.
- They were able to play checkers, prove mathematical theorems, make simple conversations (ELIZA), translate languages...

Source: <https://data-science-blog.com/blog/2018/05/14/machine-learning-vs-deep-learning-was-liegt-der-unterschied>

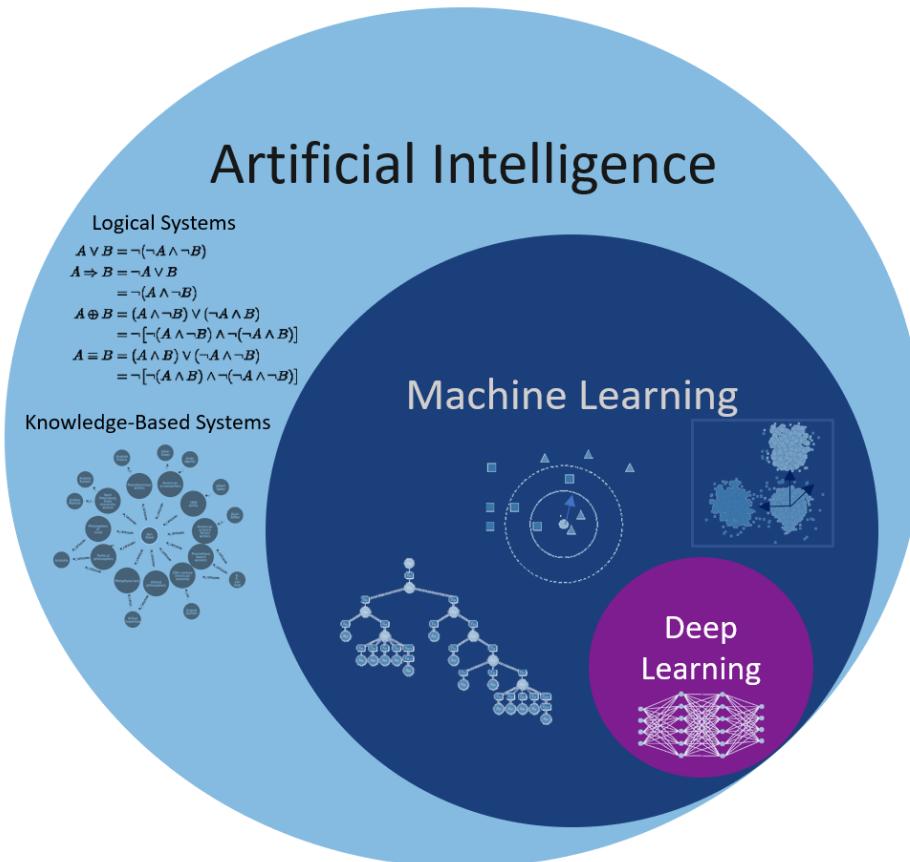
# Artificial Intelligence, Machine Learning, Deep Learning, Neurocomputing



- **Machine learning (ML)** is a branch of AI that focuses on learning from examples (data-driven).
- ML algorithms include:
  - Neural Networks (multi-layer perceptrons)
  - Statistical analysis (Bayesian modeling, PCA)
  - Clustering algorithms (k-means, GMM, spectral clustering)
  - Support vector machines
  - Decision trees, random forests
- Other names: big data, data science, operational research, pattern recognition...

Source: <https://data-science-blog.com/blog/2018/05/14/machine-learning-vs-deep-learning-wie-liegt-der-unterschied>

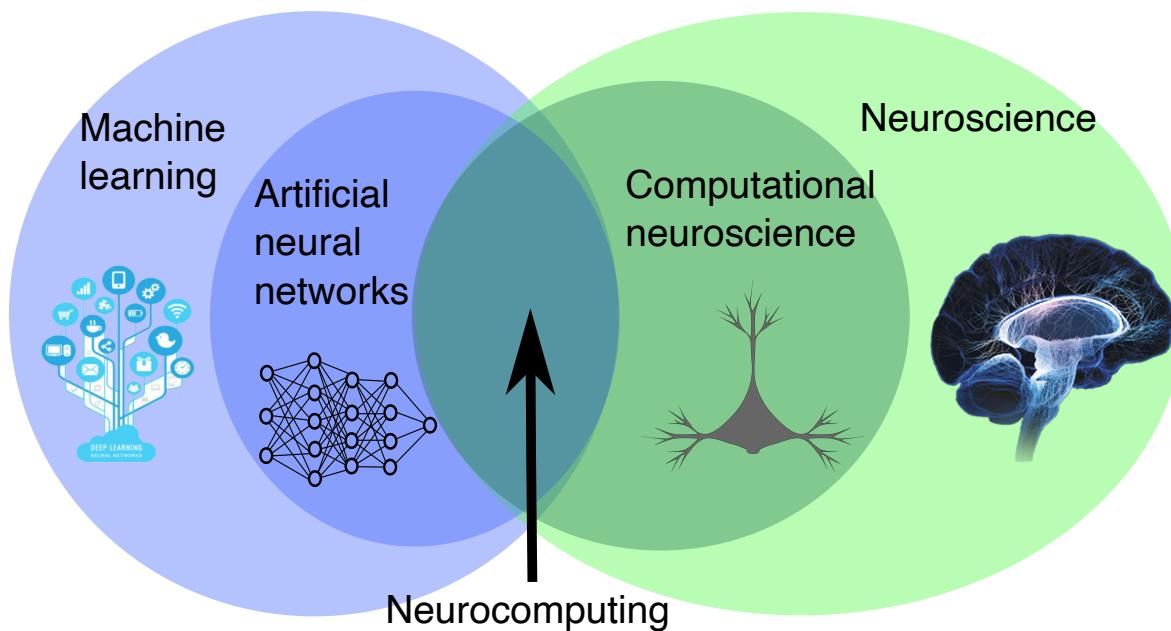
# Artificial Intelligence, Machine Learning, Deep Learning, Neurocomputing



- **Deep Learning** is a recent re-branding of neural networks.
- Deep learning focuses on learning high-level representations of the data, using:
  - Deep neural networks (DNN)
  - Convolutional neural networks (CNN)
  - Recurrent neural networks (RNN)
  - Generative models (GAN, VAE)
  - Deep reinforcement learning (DQN, PPO, AlphaGo)
  - Transformers
  - Graph neural networks

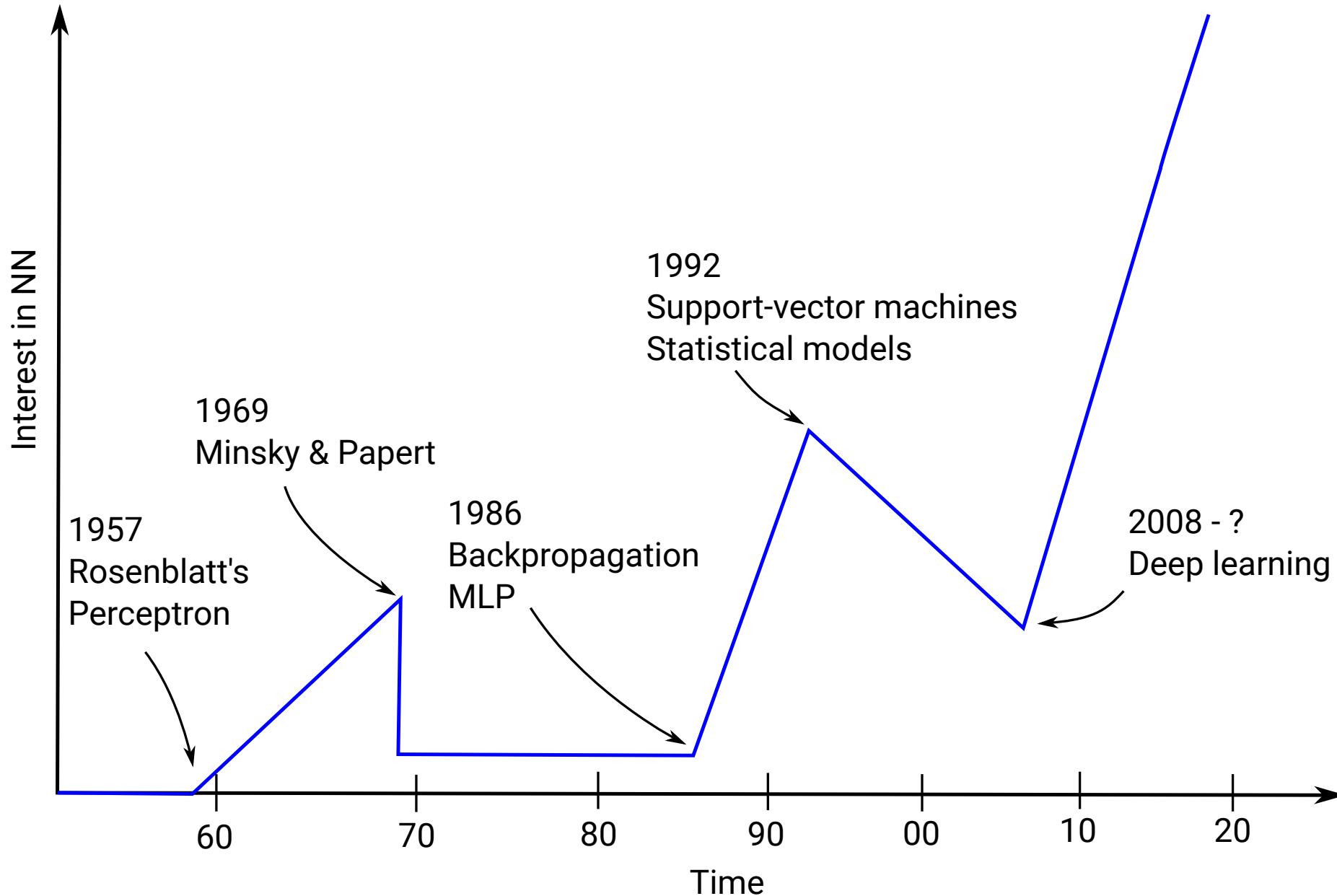
Source: <https://data-science-blog.com/blog/2018/05/14/machine-learning-vs-deep-learning-wie-liegt-der-unterschied>

# Artificial Intelligence, Machine Learning, Deep Learning, Neurocomputing



- **Neurocomputing** is at the intersection between computational neuroscience and artificial neural networks (deep learning).
- Computational neuroscience studies the functioning of the brain through detailed models.
- Neurocomputing aims at bringing the mechanisms underlying human cognition into artificial intelligence.

# AI hypes and AI winters

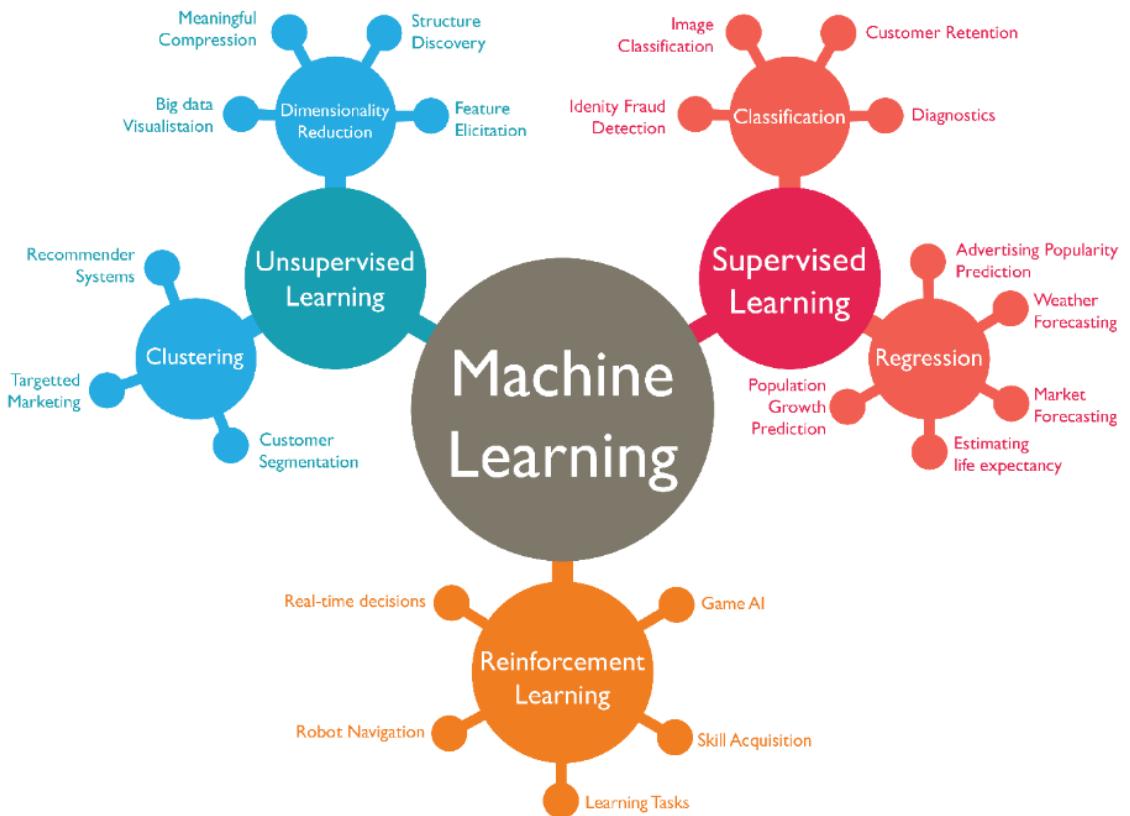


# Classification of ML techniques

- **Supervised learning:** The program is trained on a pre-defined set of training examples and used to make correct predictions when given new data.
- **Unsupervised learning:** The program is given a bunch of data and must find patterns and relationships therein.
- **Reinforcement learning:** The program explores its environment by producing actions and receiving rewards.

But also:

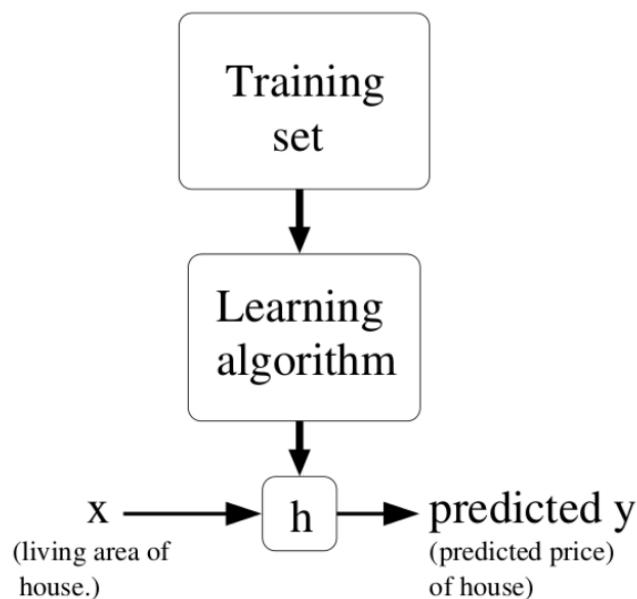
- Self-supervised learning, self-taught learning, developmental learning...



Source: <http://www.isaziconsulting.co.za/machinelearning.html>

# **1- Supervised learning**

# Supervised Learning



- **Supervised learning** consists in presenting a dataset of input and output **samples** (or examples)  $(x_i, t_i)_{i=1}^N$  to a parameterized model.

$$y_i = f_\theta(x_i)$$

- The goal of learning is to adapt the parameters  $\theta$ , so that the model reduces its **prediction error** on the training data.

$$\theta^* = \operatorname{argmin} \sum_{i=1}^N ||t_i - y_i||$$

Source: Andrew Ng, Stanford CS229,

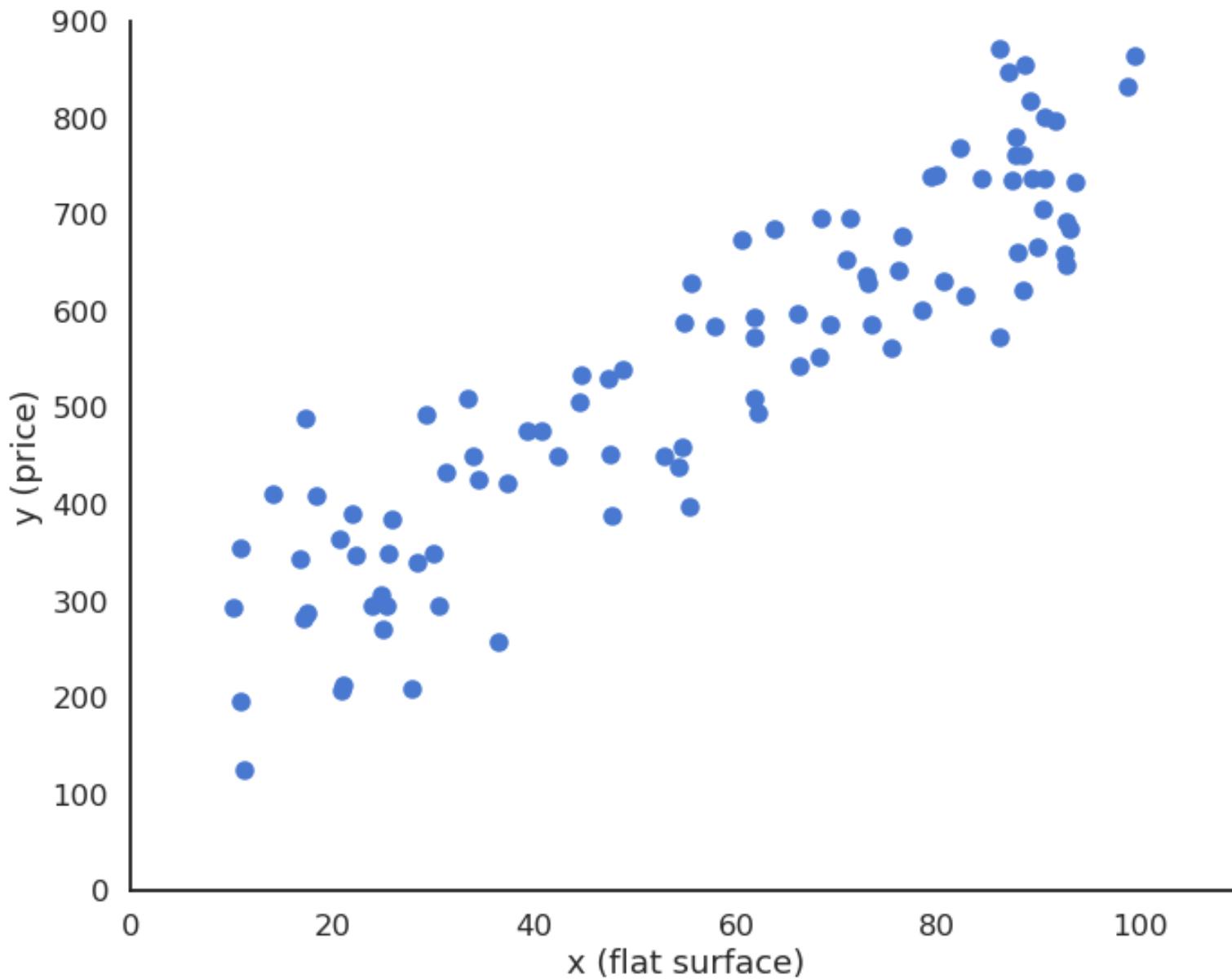
<https://see.stanford.edu/materials/aimlcs229/cs229-notes1.pdf>

- When learning is successful, the model can be used on novel examples (**generalisation**).

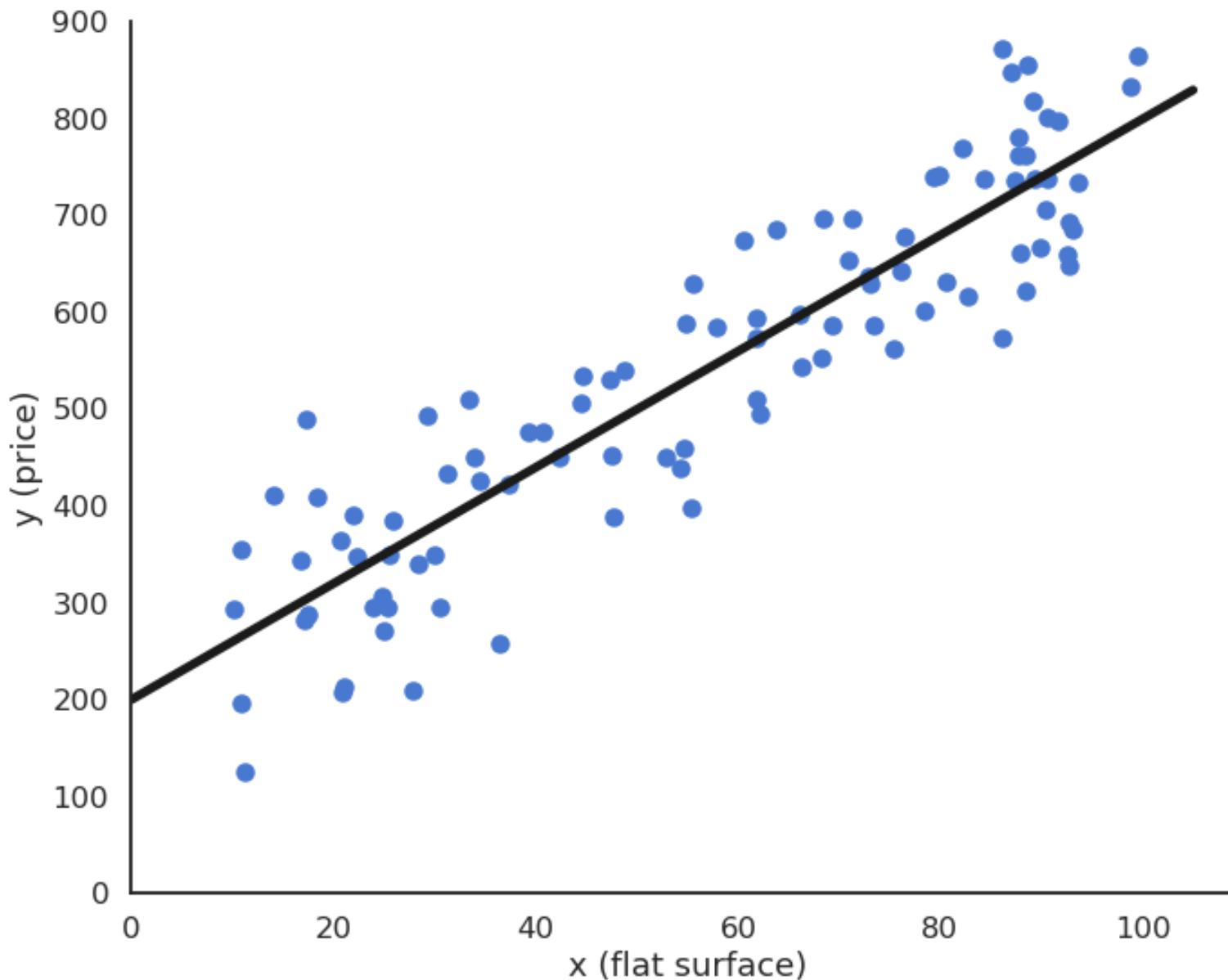
- The modality of the inputs and outputs does not really matter:

- Image → Label : **image classification**
- Image → Image : **semantic segmentation**
- Speech → Text : **speech recognition**
- Text → Speech : **speech synthesis**

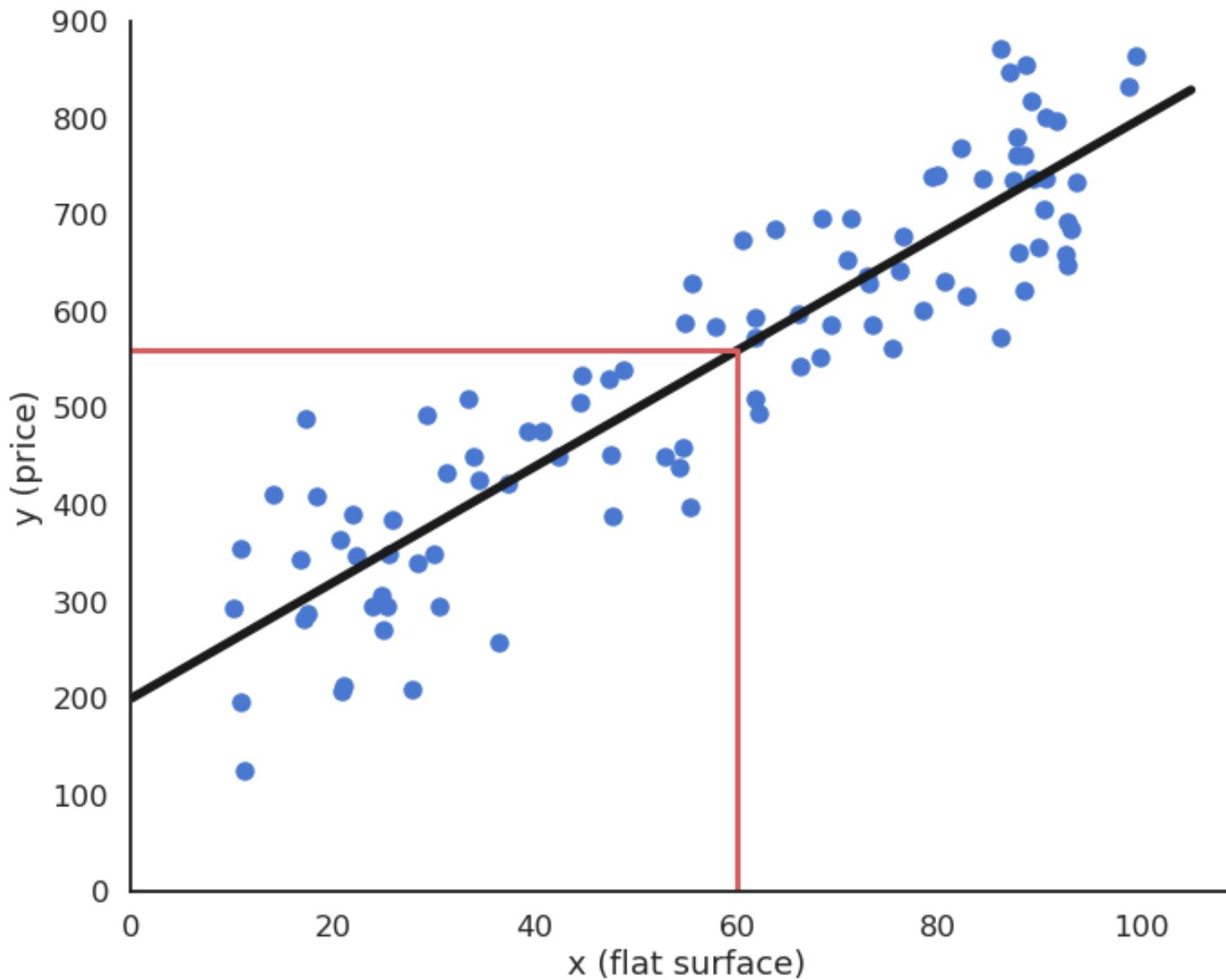
# Supervised learning : regression



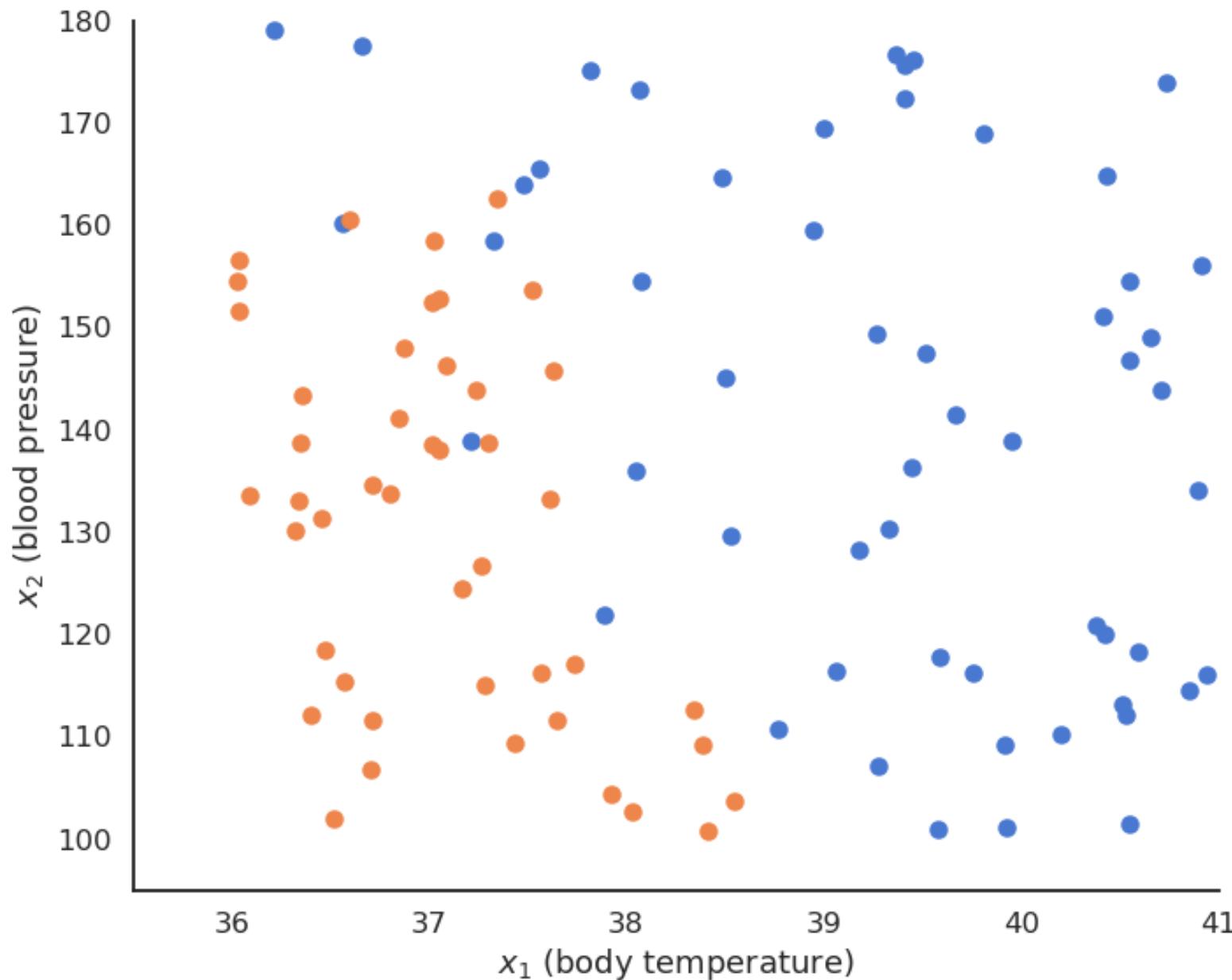
# Supervised learning : regression



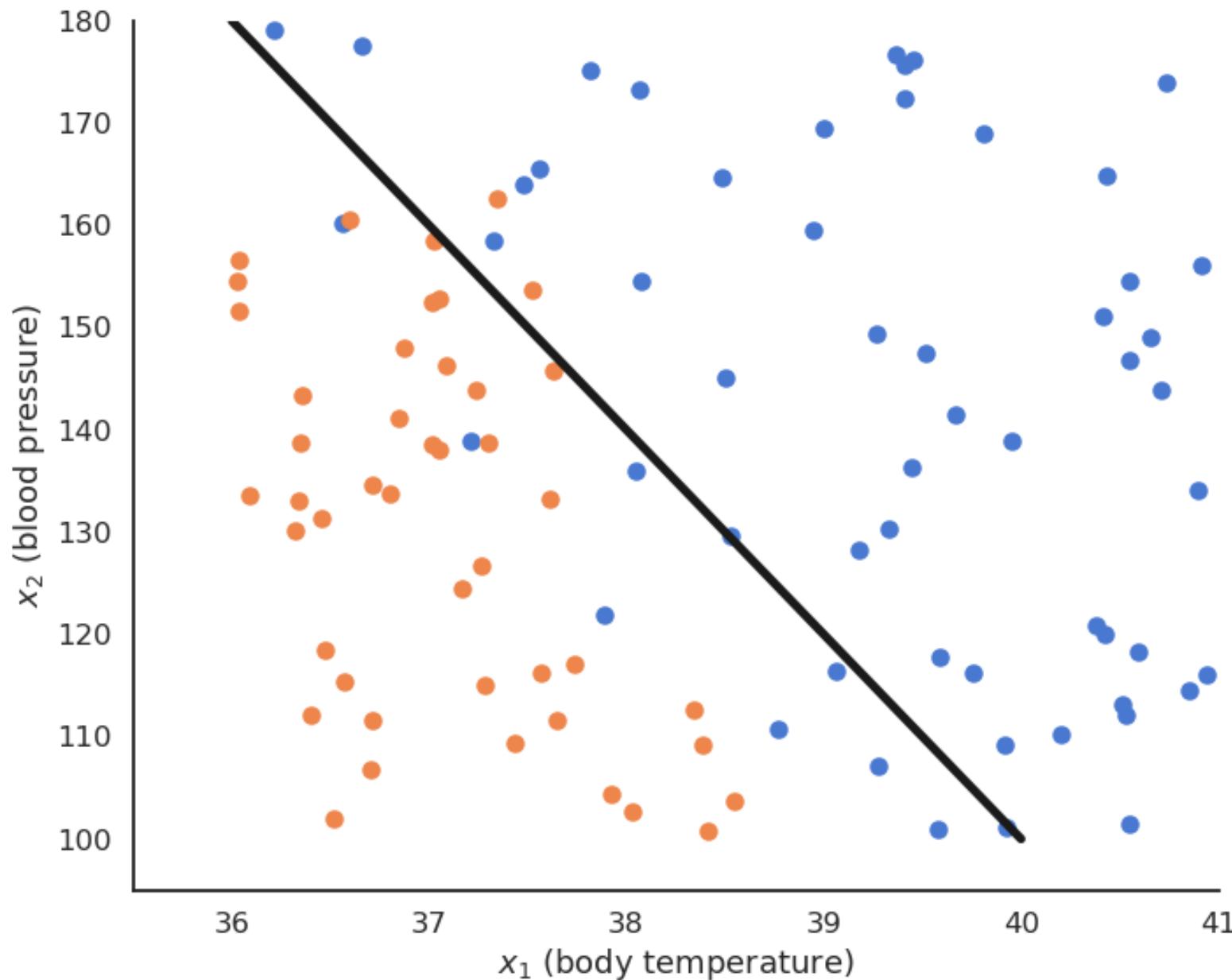
# Supervised learning : regression



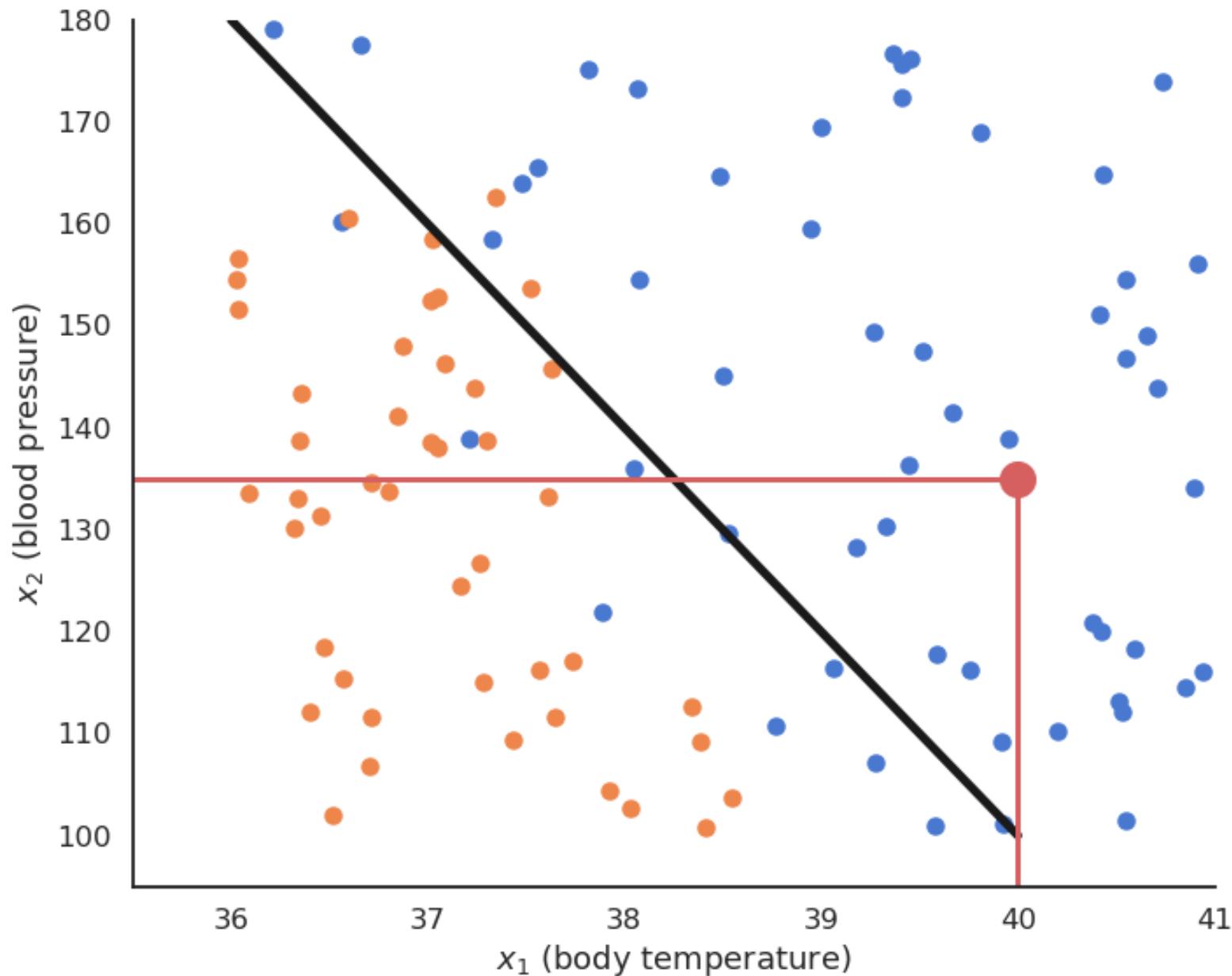
# Supervised learning : classification



## Supervised learning : classification

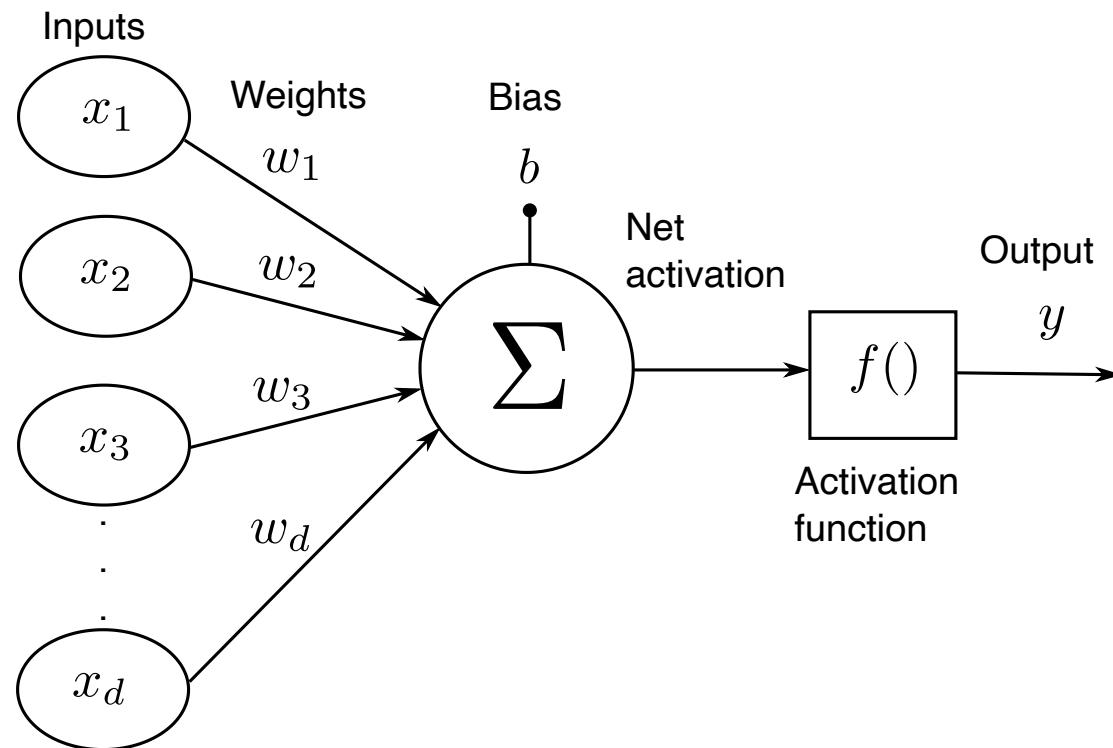


## Supervised learning : classification



# The artificial neuron

- A single artificial neuron is able to solve linear classification/regression problems:

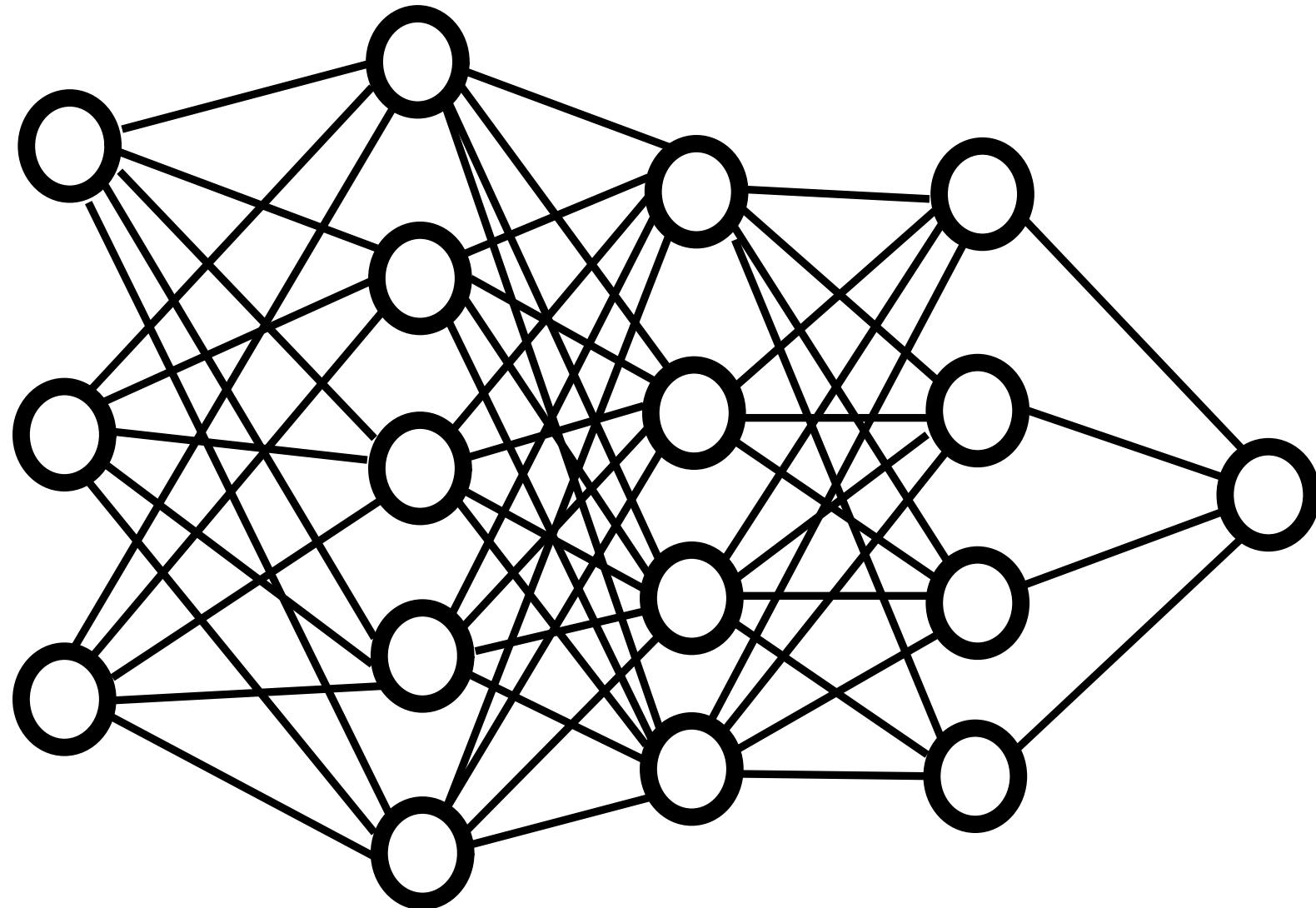


$$y = f\left(\sum_{i=1}^d w_i x_i + b\right)$$

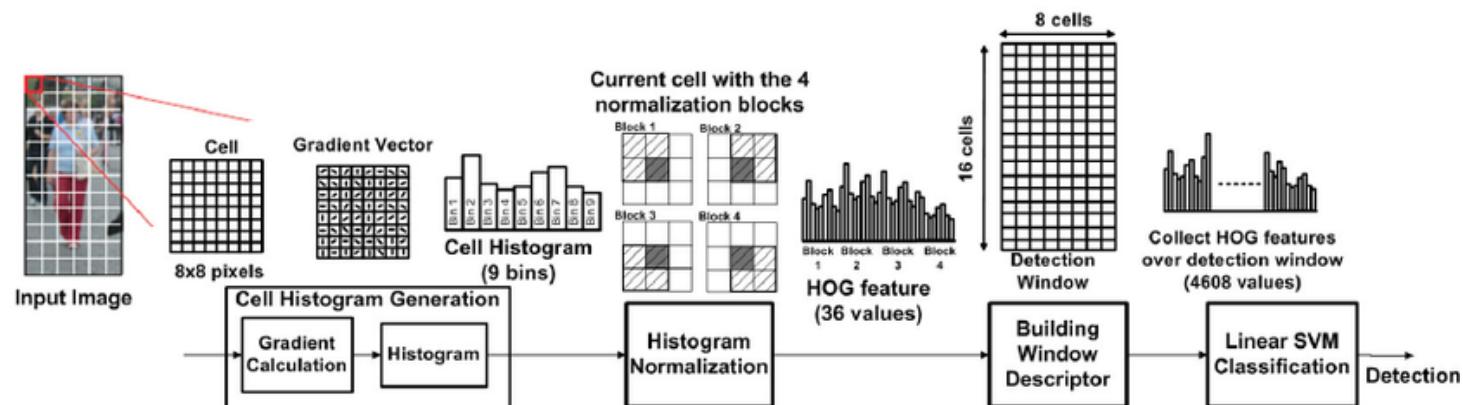
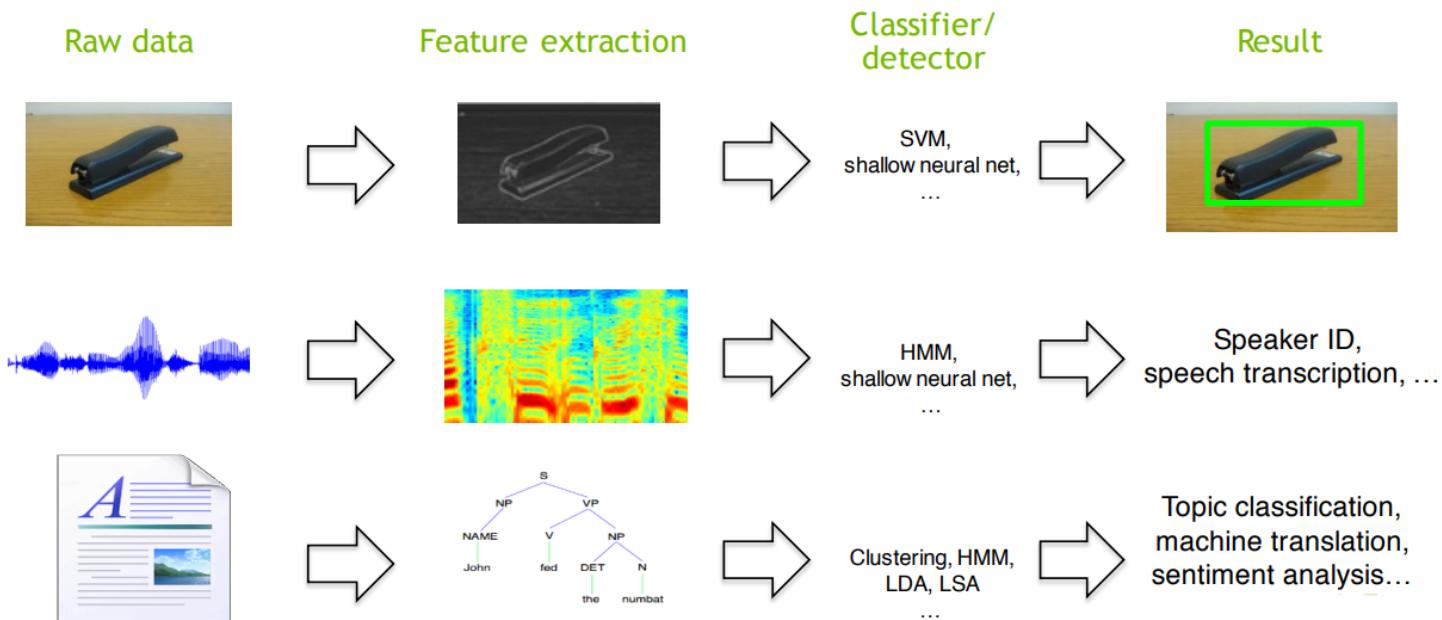
- A neuron integrates inputs  $x_i$  by multiplying them with weights  $w_i$ , adds a bias  $b$  and transforms the result into an output  $y$  using a transfer function (or activation function)  $f$ .

# Artificial Neural Network

- A **neural network** (NN) is able to solve non-linear classification/regression problems by combining many artificial neurons.



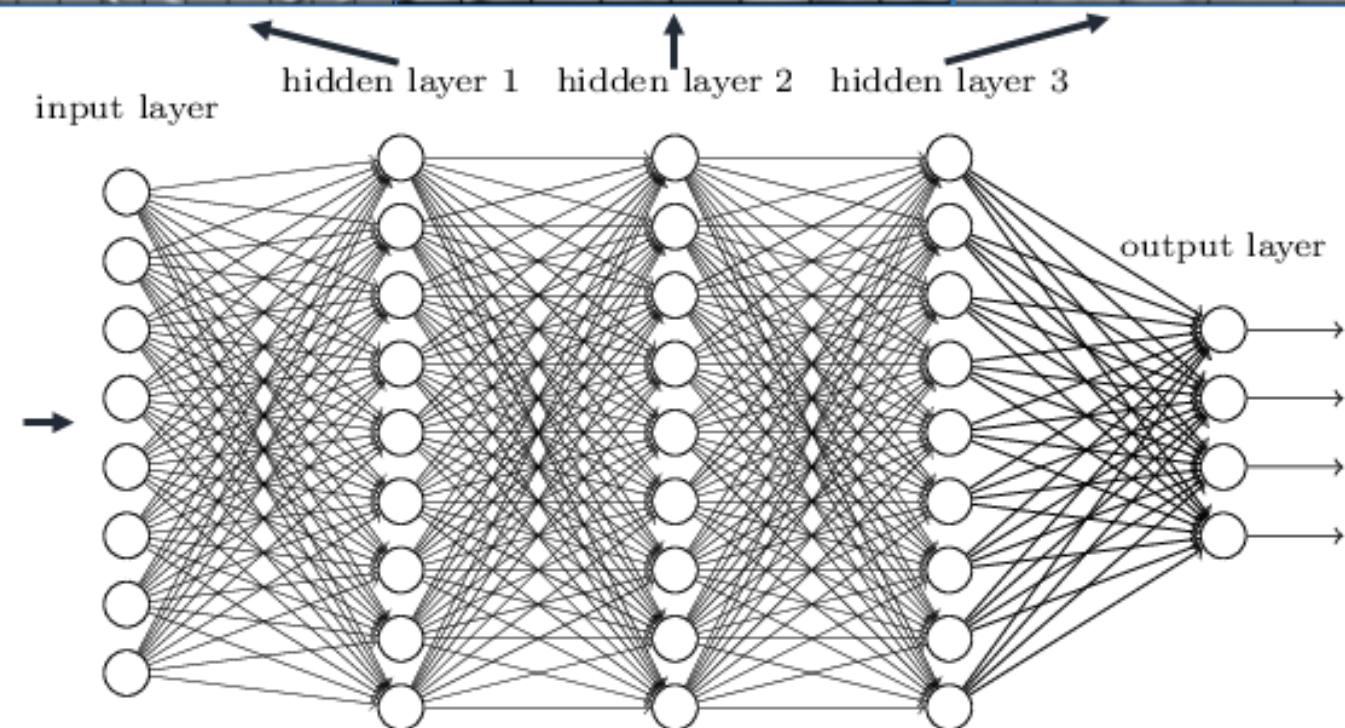
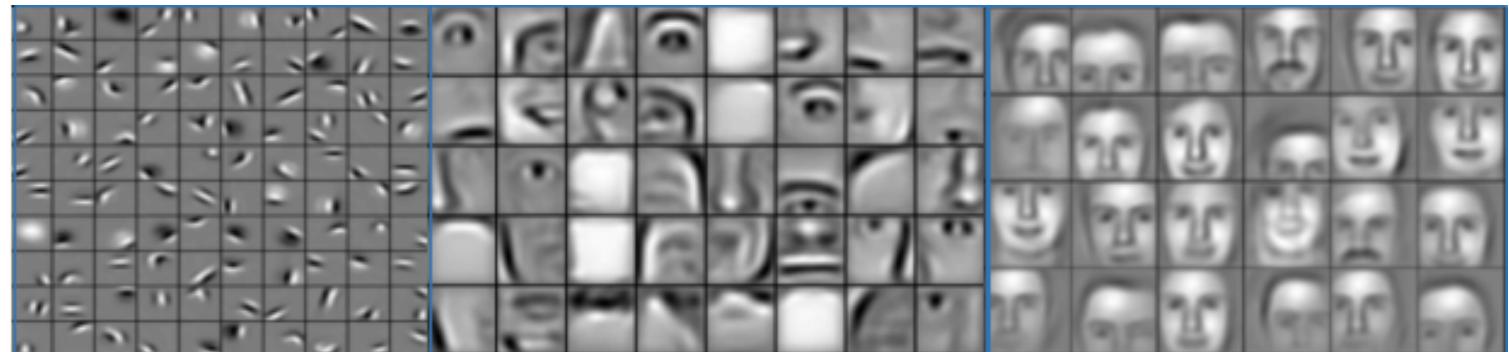
# Classical approach to pattern recognition



# Deep Learning approach to pattern recognition

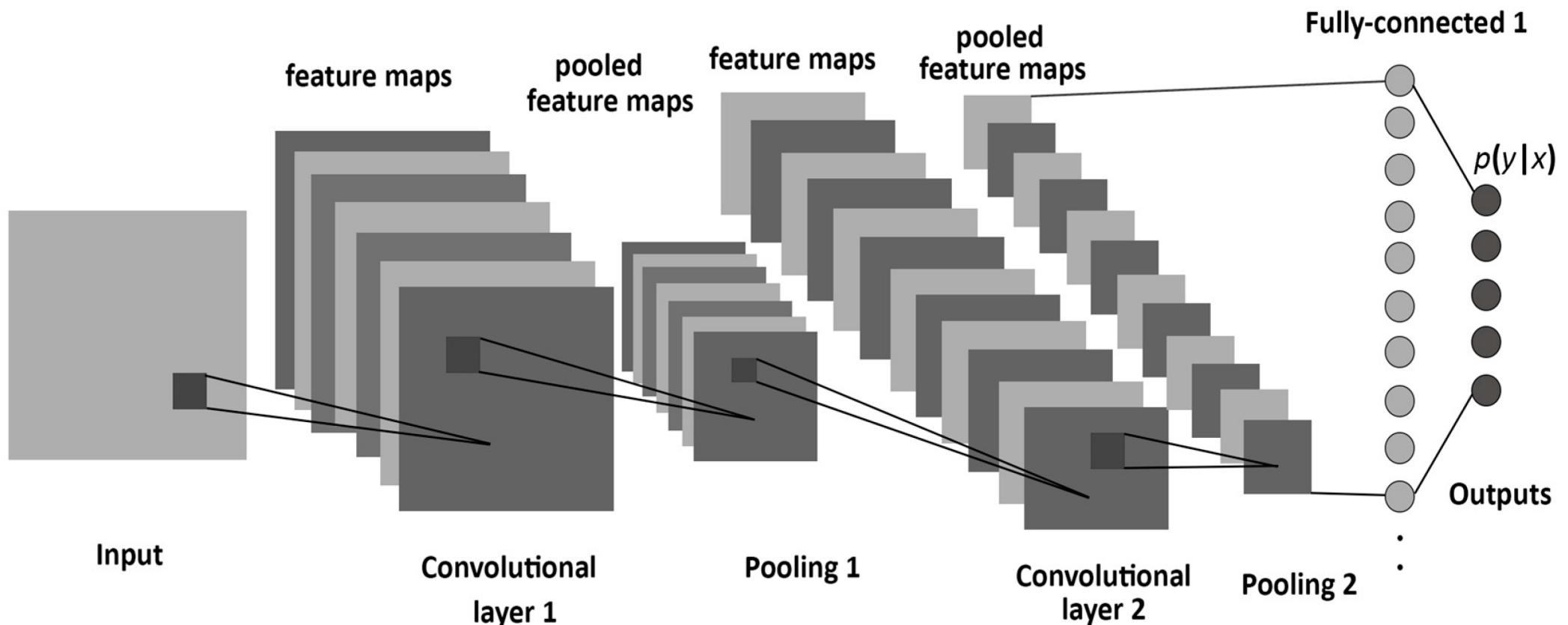
- **End-to-end** learning: the NN is trained directly on the raw data (pixels, sounds, text) and solves a non-linear classification/regression problem.

Deep neural networks learn hierarchical feature representations



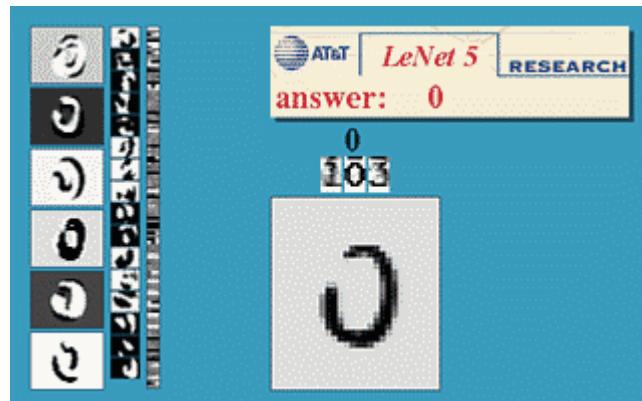
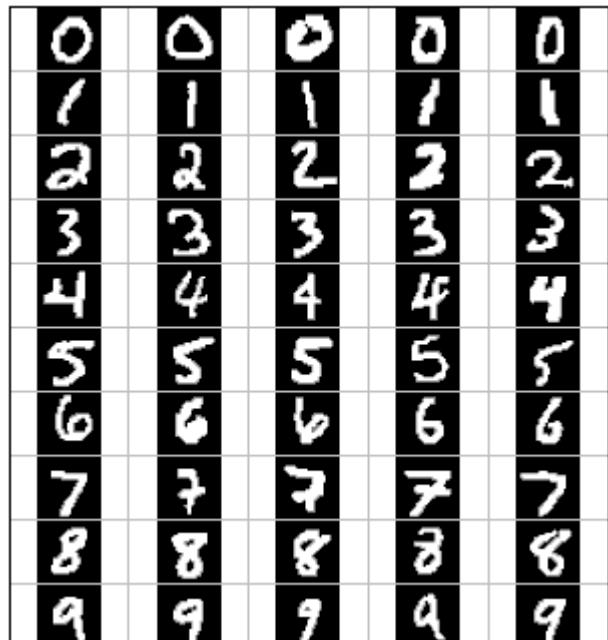
# Convolutional neural networks

- A **convolutional neural network** (CNN) is a cascade of convolution and pooling operations, extracting layer by layer increasingly complex features.
- It can be trained on huge datasets of annotated examples.



# Handwriting recognition

- The MNIST database is the simplest benchmark for object recognition (> 99.5 %).
- One of the early functional CNN was LeNet5, able to classify digits.



LeCun et al. (1998). Gradient-Based Learning Applied to Document Recognition (Proc. IEEE 1998)

# ImageNet recognition challenge

- The ImageNet challenge was a benchmark for computer vision algorithms, providing millions of annotated images for object recognition, detection and segmentation.

## Object recognition

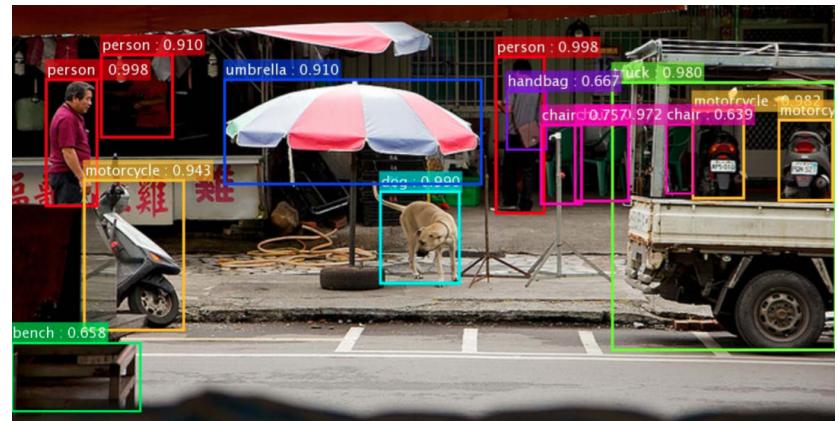
Easiest classes



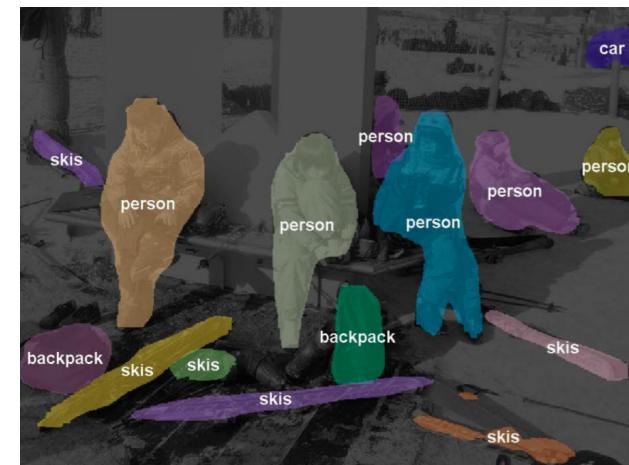
Hardest classes



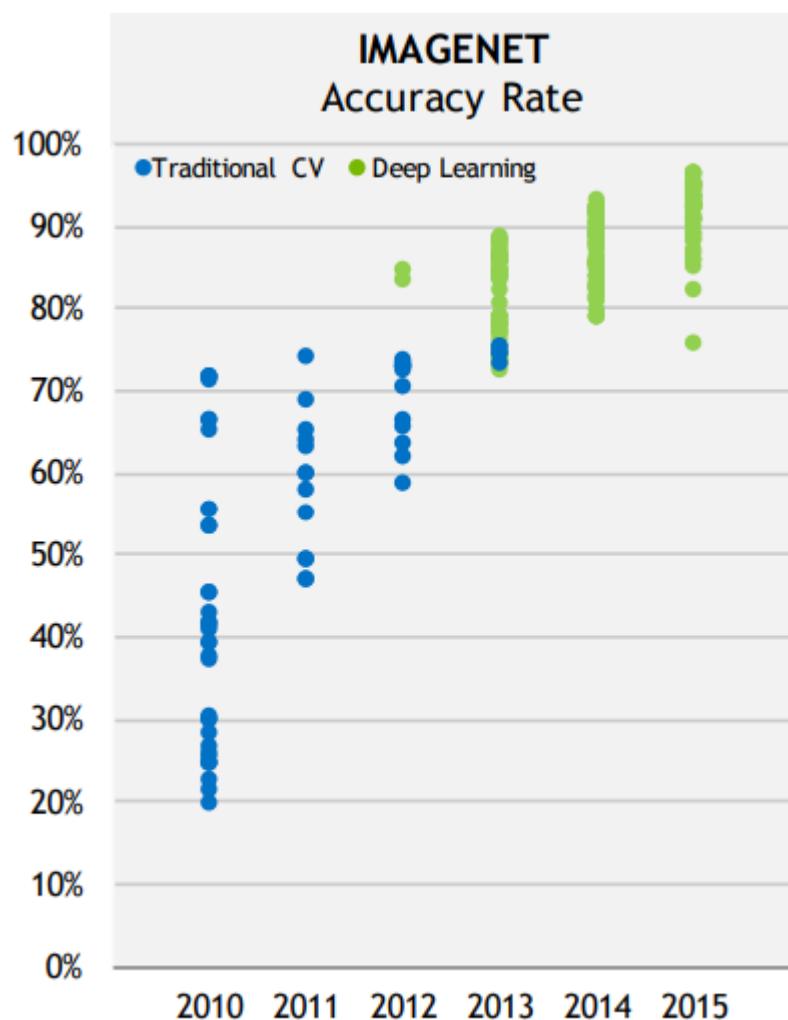
## Object detection



## Object segmentation



# AlexNet

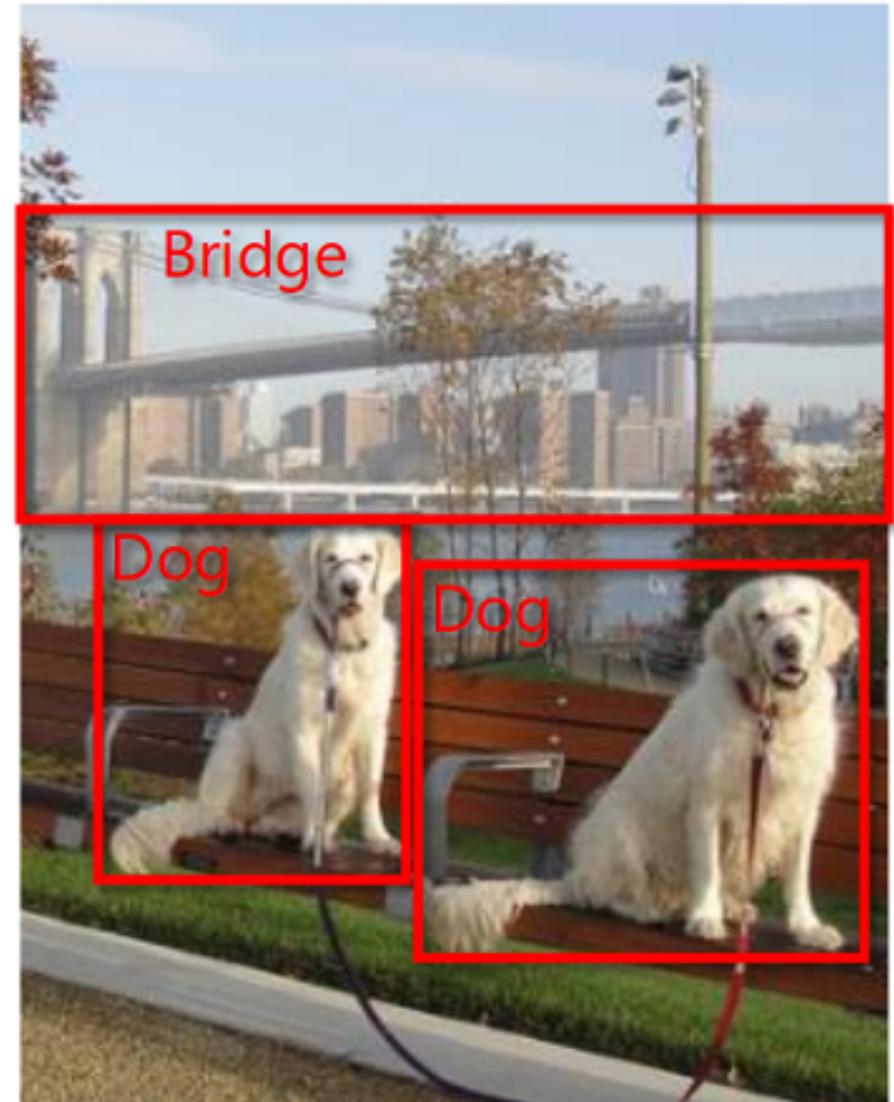


- Classical computer vision methods obtained moderate results, with error rates around 30%.
- In 2012, Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton (Uni Toronto) used a CNN (**AlexNet**) without any preprocessing, using directly images as inputs.
- To the big surprise of everybody, they won with an error rate of 15%, half of what other methods could achieve.
- Since then, everybody uses deep neural networks for object recognition.
- The deep learning hype had just begun...
  - Computer vision
  - Natural language processing
  - Speech processing
  - Robotics, control

# Object detection



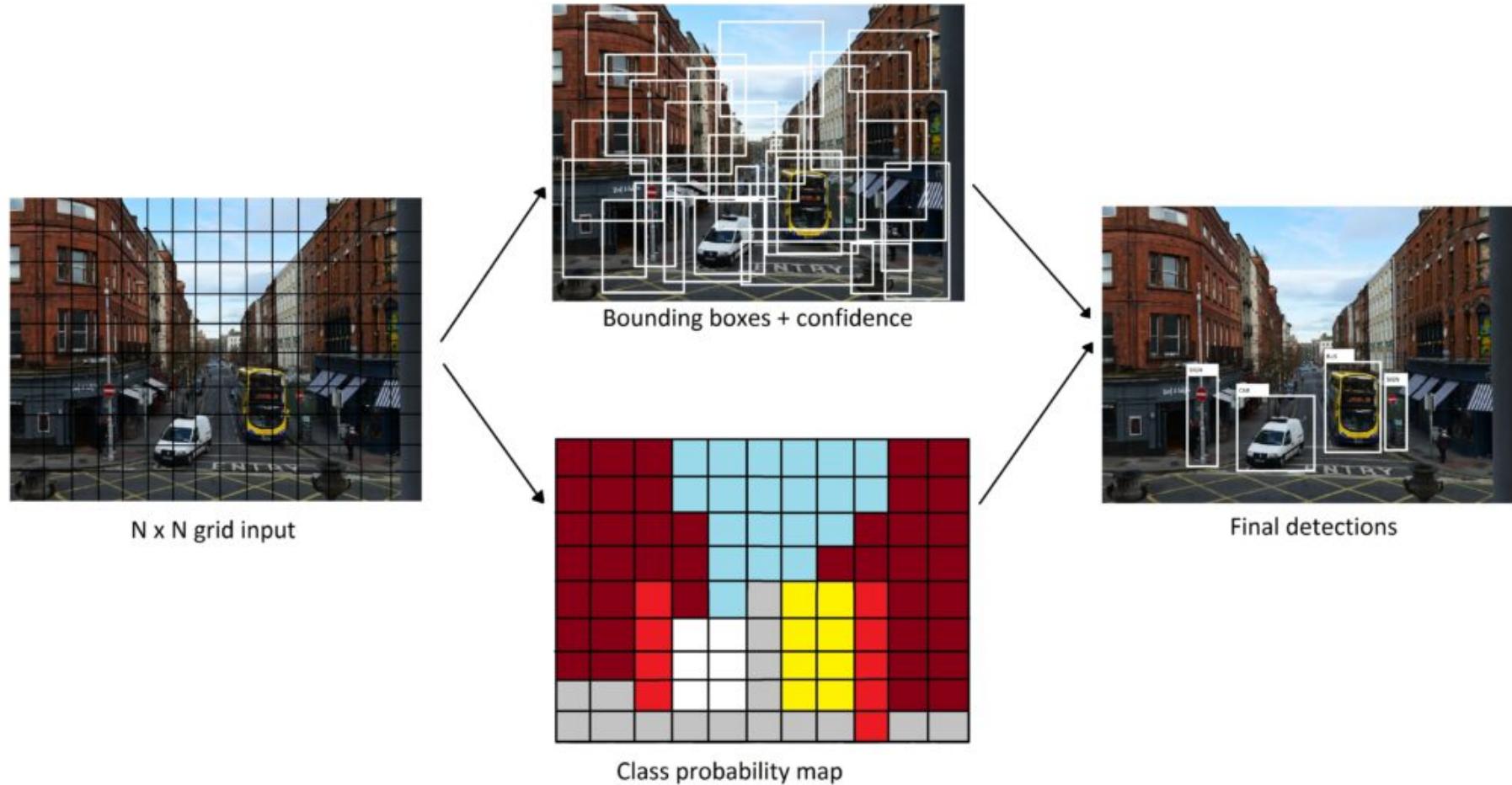
Classification, easy these days



Object detection, still a lot harder

# Object detection

- It turns out object detection is both a classification (what) and regression (where) problem.
- Neural networks can be trained to do it given enough annotated data.



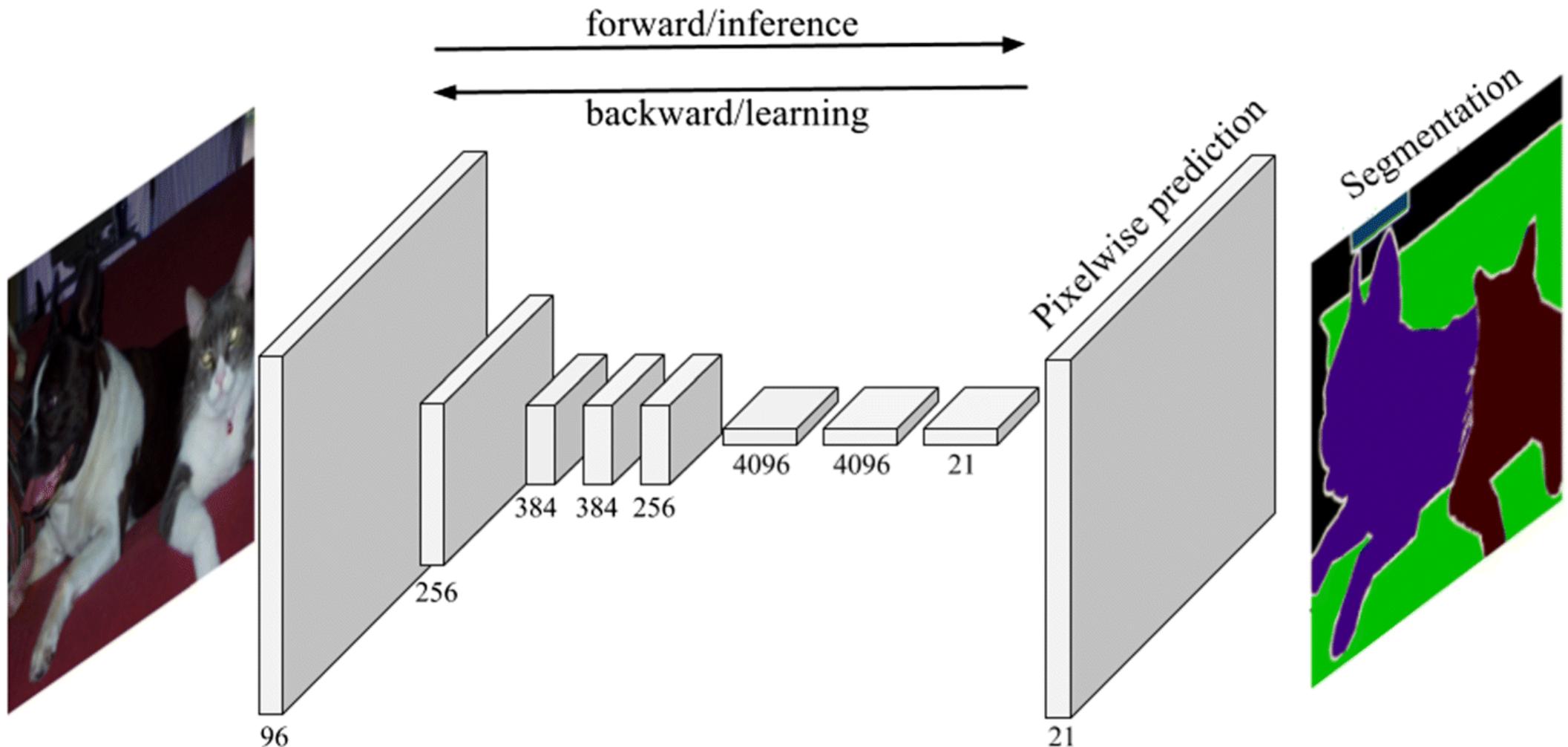
Source: <http://datahacker.rs/od1-yolo-object-detection/>

# Object detection



# Semantic segmentation

- Classes can be predicted at the pixel level, allowing **semantic segmentation**.



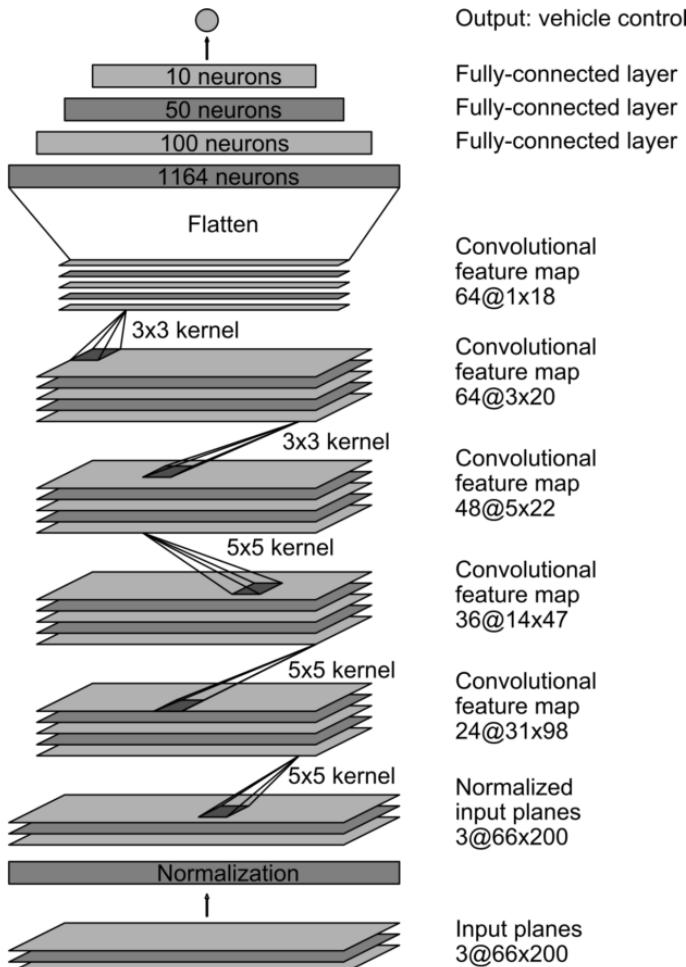
Badrinarayanan, Handa and Cipolla (2015). "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling." arXiv:1505.07293

# Semantic segmentation

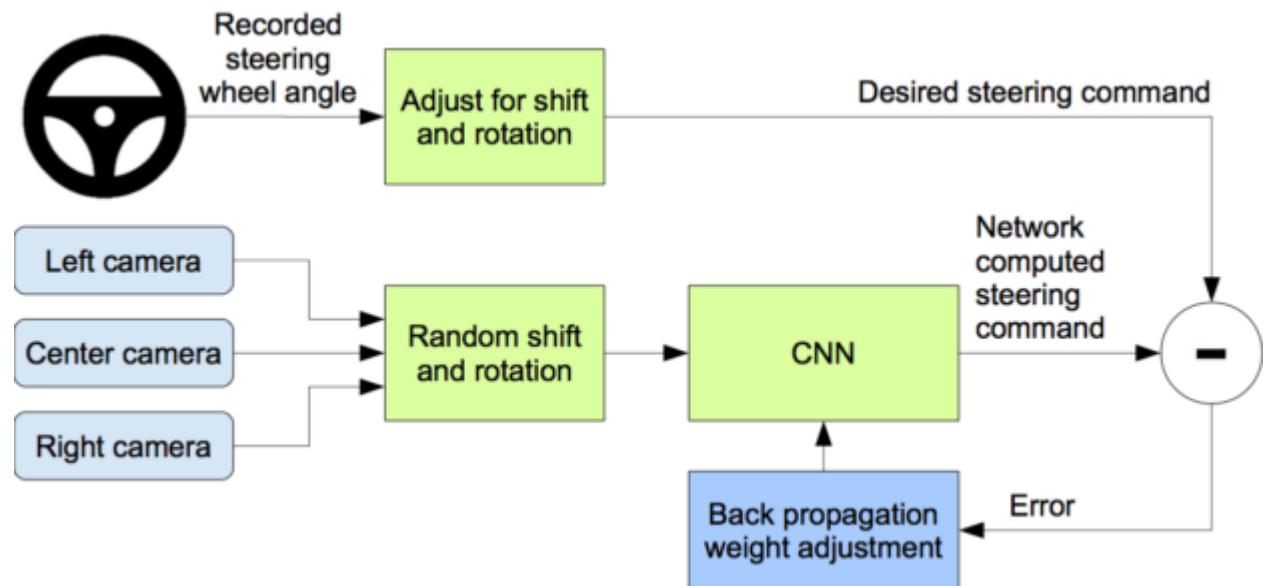
Mask RCNN - COCO - instance segmentation



# Dave2 : NVIDIA's self-driving car



- NVIDIA trained a CNN to reproduce wheel steerings from experienced drivers using only a front camera.
- After training, the CNN took control of the car.

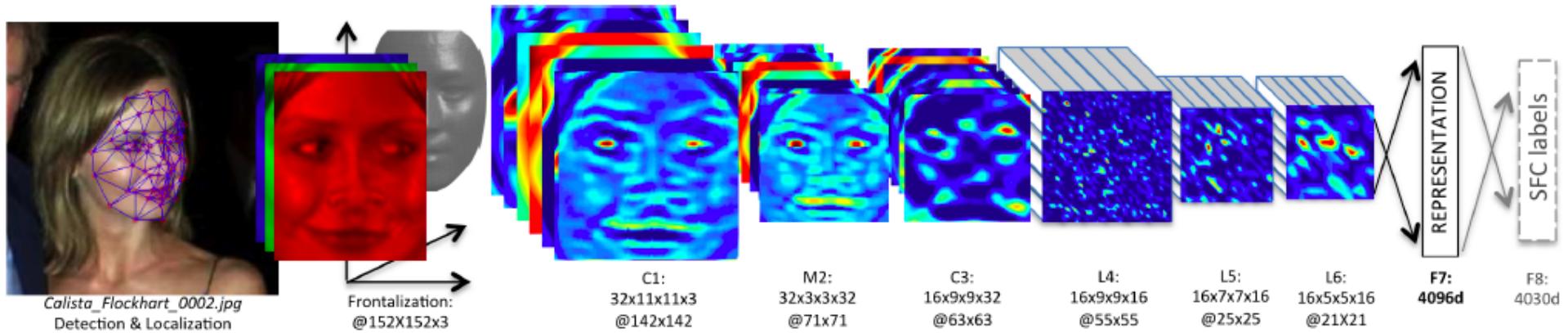


# Dave2 : NVIDIA's self-driving car

NVIDIA Autonomous Car

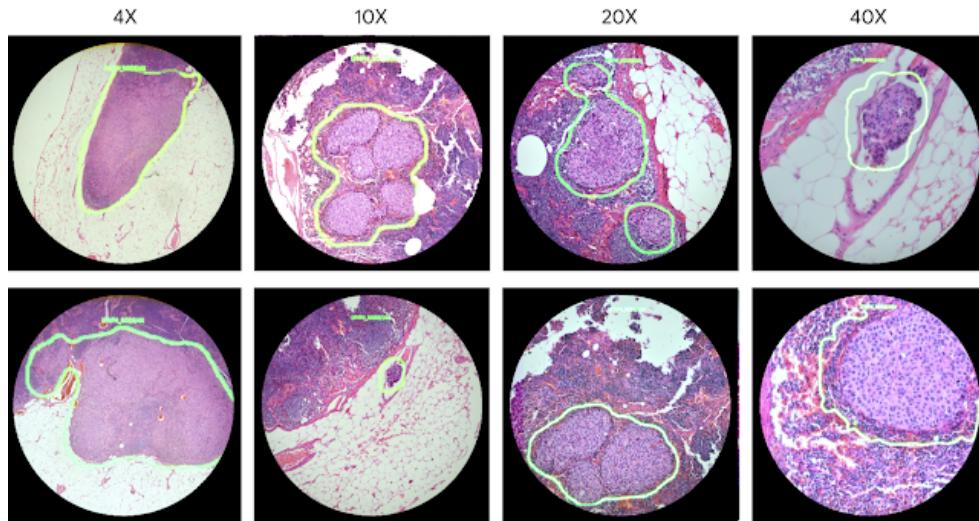


# Facial recognition

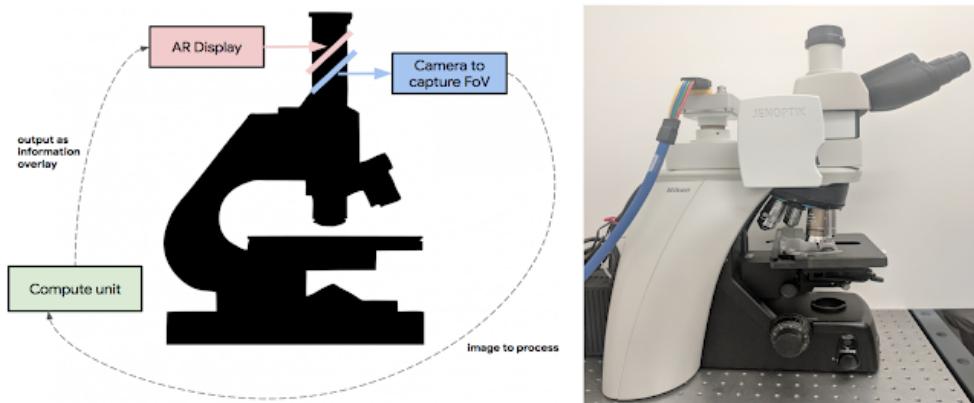


- Facebook used 4.4 million annotated faces from 4030 users to train **DeepFace**.
- Accuracy of 97.35% for recognizing faces, on par with humans.
- Used now to recognize new faces from single examples (transfer learning, one-shot learning).

# Augmented Reality Microscope for Real-time Automated Detection of Cancer

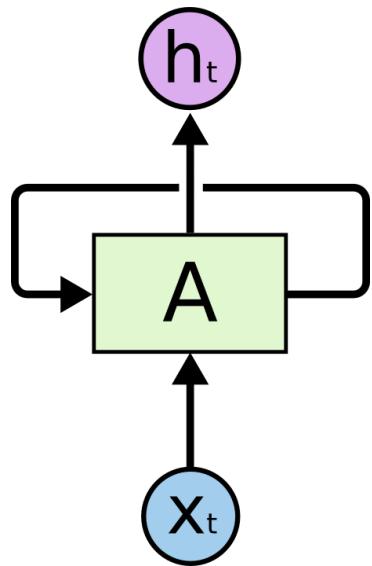


Detecting cancer in real-time with machine learning



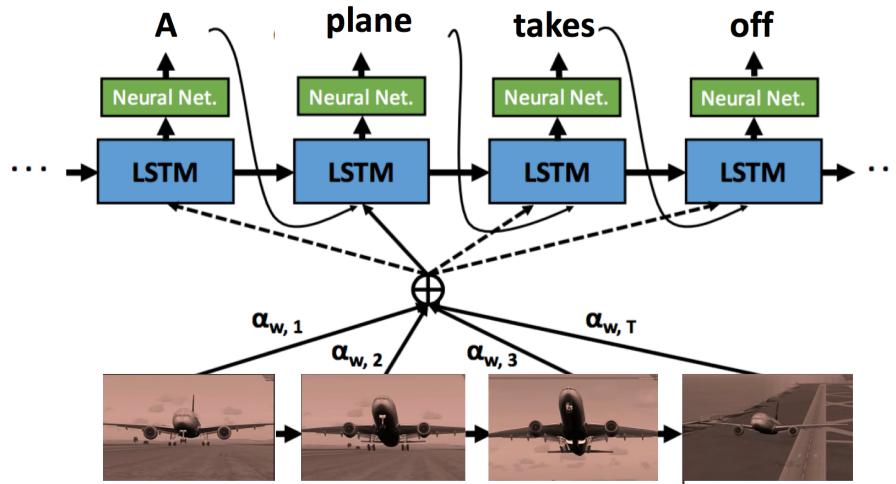
# **Recurrent neural networks**

# Recurrent neural networks



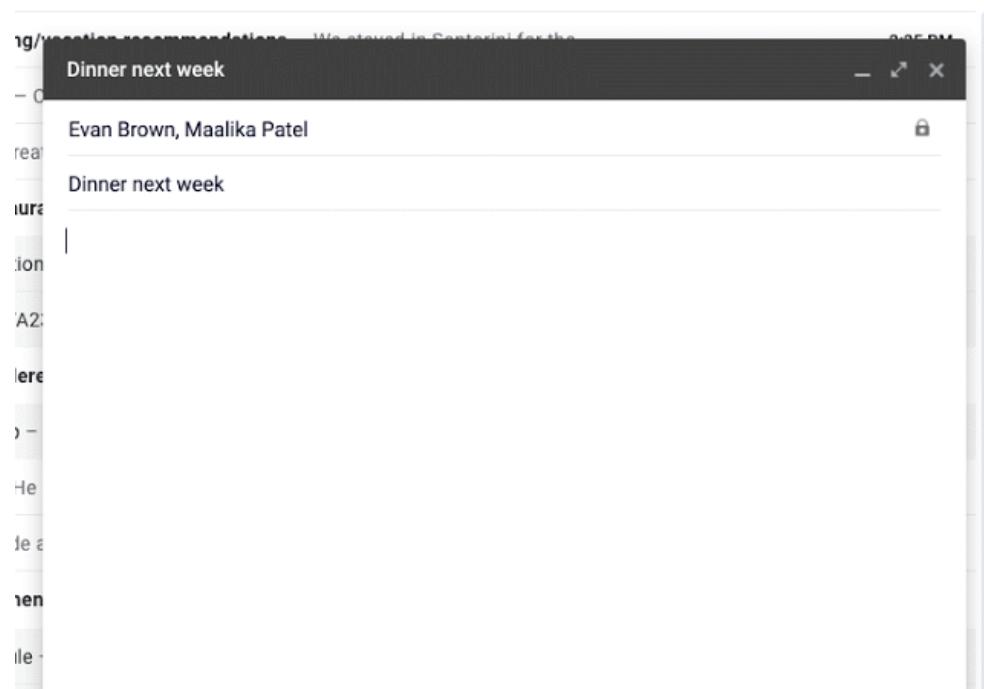
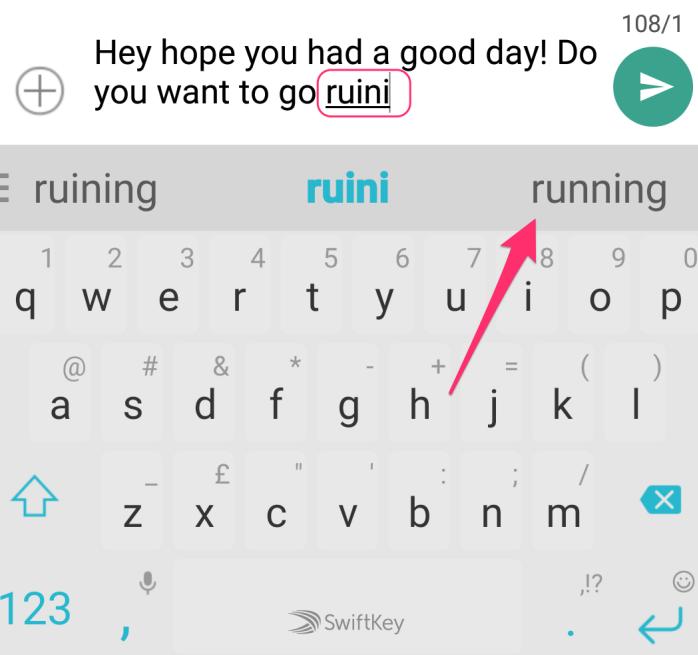
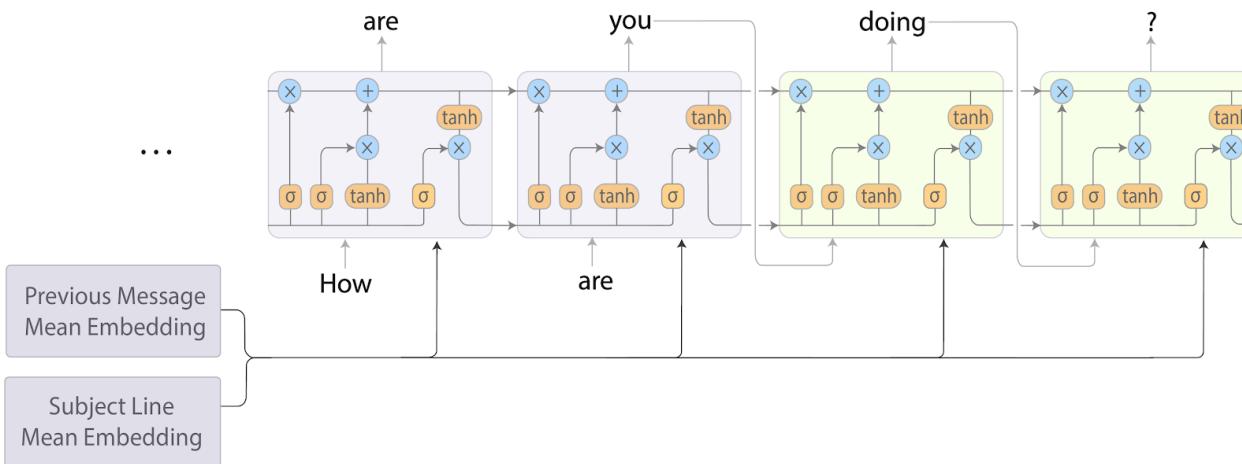
Source: C. Olah

- A **recurrent neural network (RNN)** uses its previous output as an additional input (context).
- The inputs are integrated over time to deliver a response at the correct moment.
- This allows to deal with time series (texts, videos) without increasing the input dimensions.
- The input to the RNN can even be the output of a pre-trained CNN.
- The most efficient RNN is called **LSTM** (Long short-term memory networks) (Hochreiter and Schmidhuber, 1997).





# Natural Language Processing : Automatic word/sentence completion





# Natural Language Processing : Text Generation

PANDARUS:

Alas, I think he shall be come approached and the day  
When little strain would be attain'd into being never fed,  
And who is but a chain and subjects of his death,  
I should not sleep.

Second Senator:

They are away this miseries, produced upon my soul,  
Breaking and strongly should be buried, when I perish  
The earth and thoughts of many states.

DUKE VINCENTIO:

Well, your wit is in the care of side and that.

Second Lord:

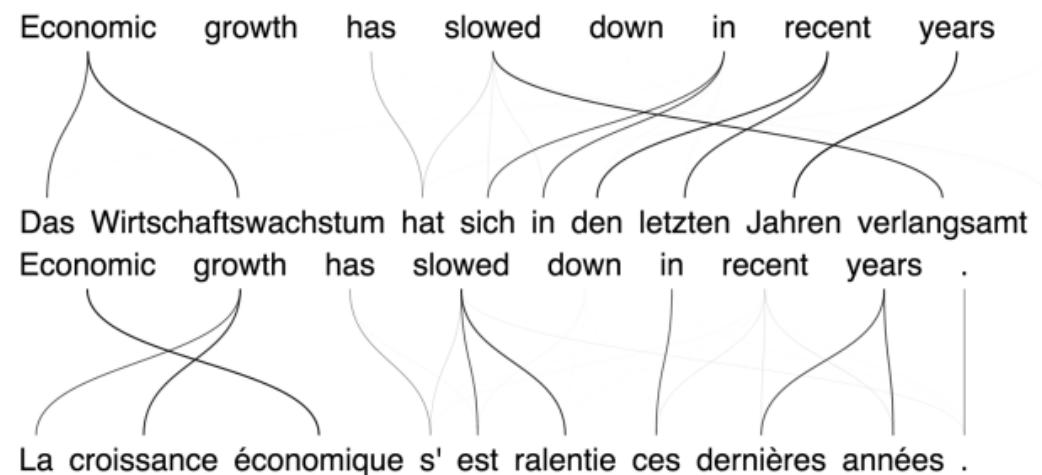
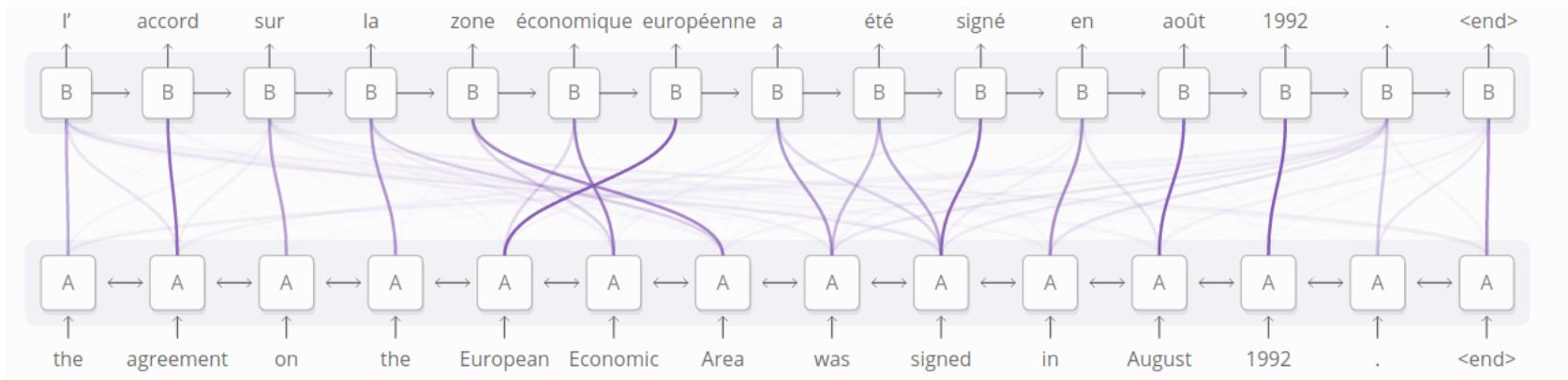
They would be ruled after this chamber, and  
my fair nues begun out of the fact, to be conveyed,  
Whose noble souls I'll have the heart of the wars.

Clown:

Come, sir, I will make did behold your worship.

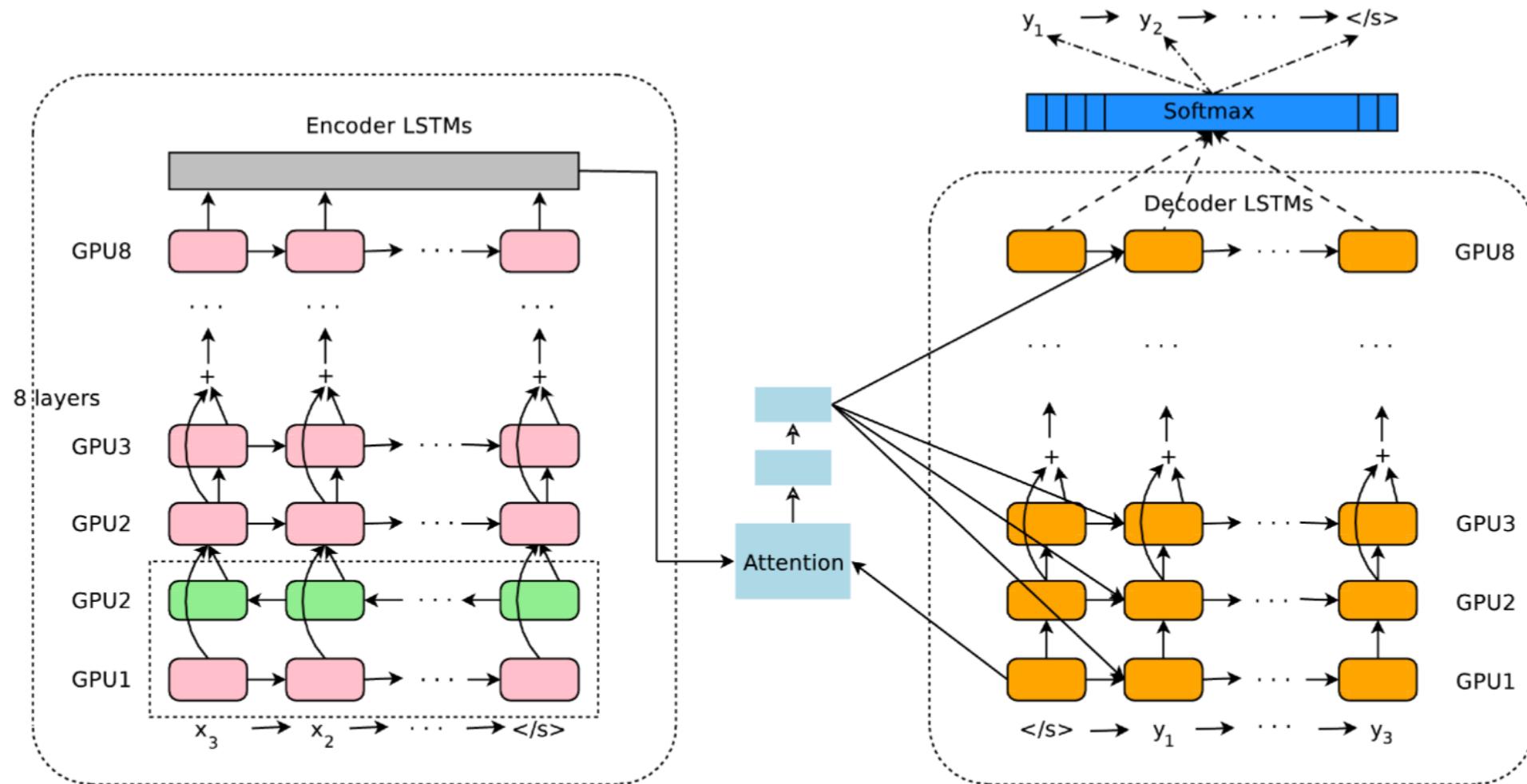
- Characters or words are fed one by one into a LSTM.
- The desired output is the next character or word in the text.
- Example:
  - Inputs: **To, be, or, not, to**
  - Output: **be**
- The text on the left was generated by a LSTM having read the entire writings of William Shakespeare.
- Each generated word is used as the next input.

# Natural Language Processing : text translation



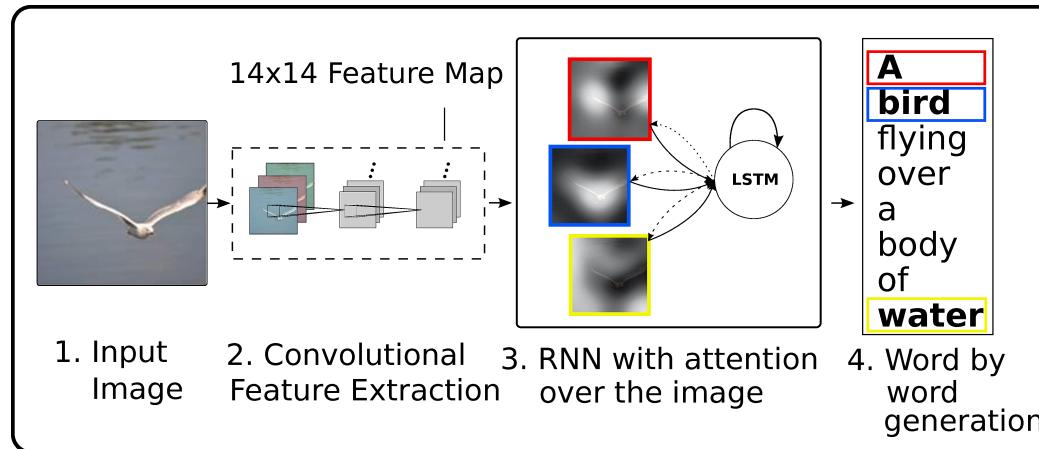
- Two LSTM can be stacked to perform sequence-to-sequence translation (**seq2seq**).
- One is the encoder, the other the decoder.

# Natural Language Processing : Google Neural Machine Translation



- Same idea, but with much more layers...
- Can translate any pair of languages!

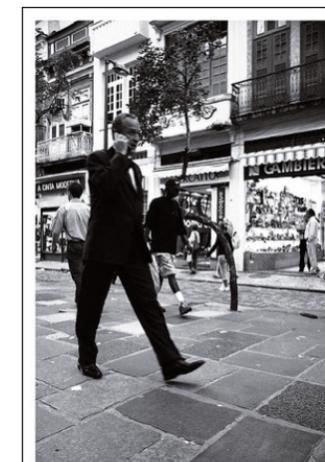
# Caption Generation



↑ a living room with a couch and a television

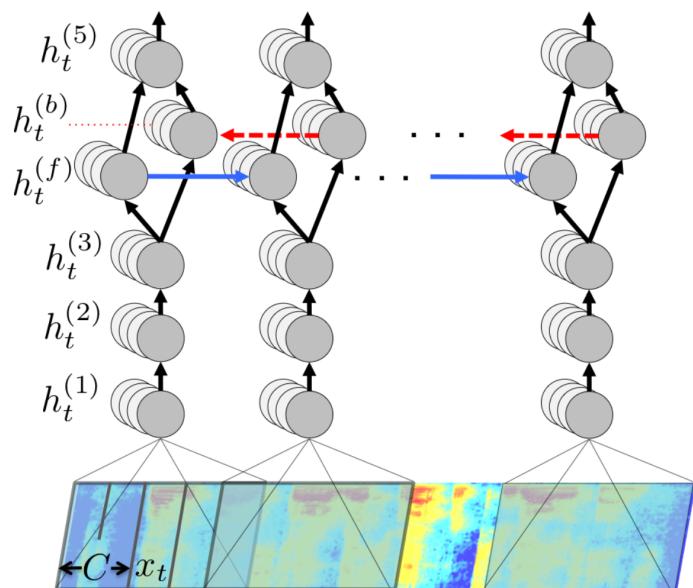
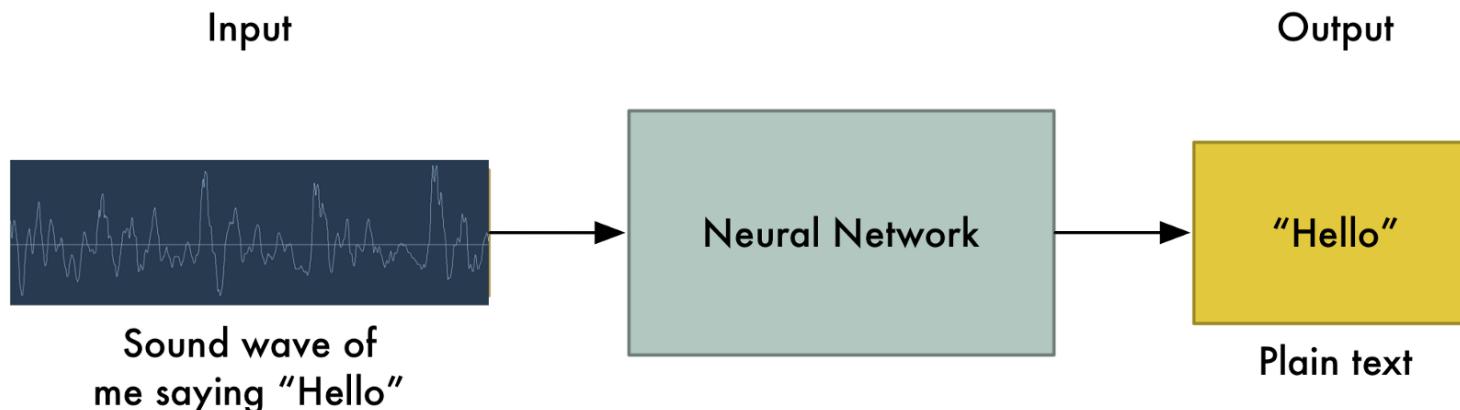


↑ a man riding a bike on a beach



a man is walking down the street with a suitcase ↗

# Voice recognition

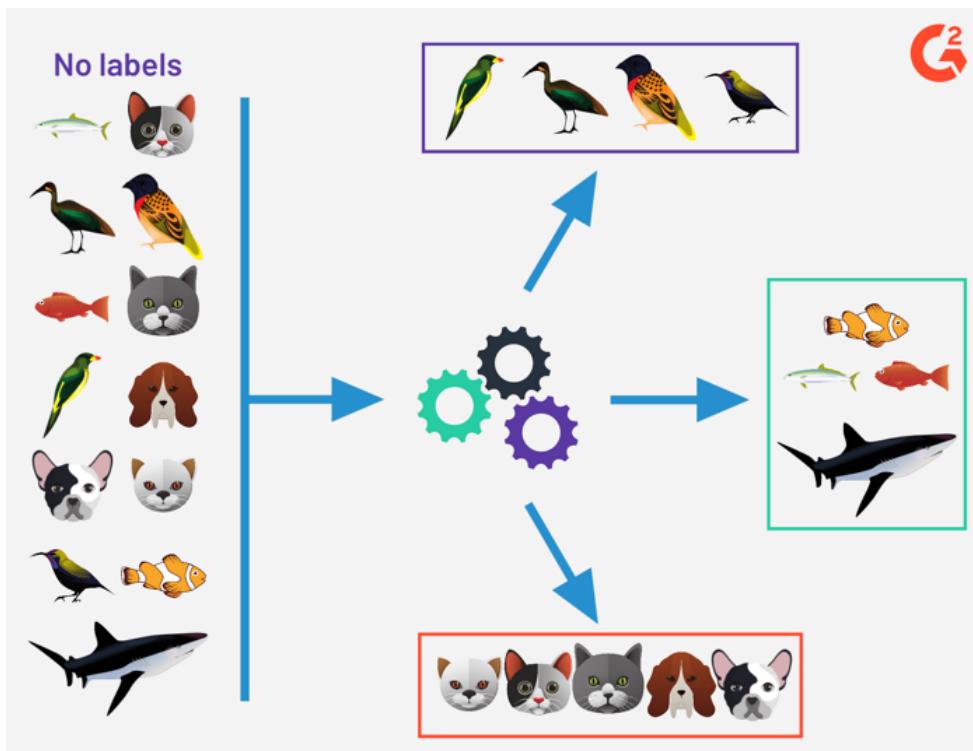


- CNNs are not limited to images, voice signals can also be recognized using their mel-spectrum.
- Siri, Alexa, Google now, etc. use recurrent CNNs to recognize vocal commands and respond.
- **DeepSpeech** from Baidu is one of the state-of-the-art approach.

## **2 - Unsupervised learning**

# Unsupervised learning

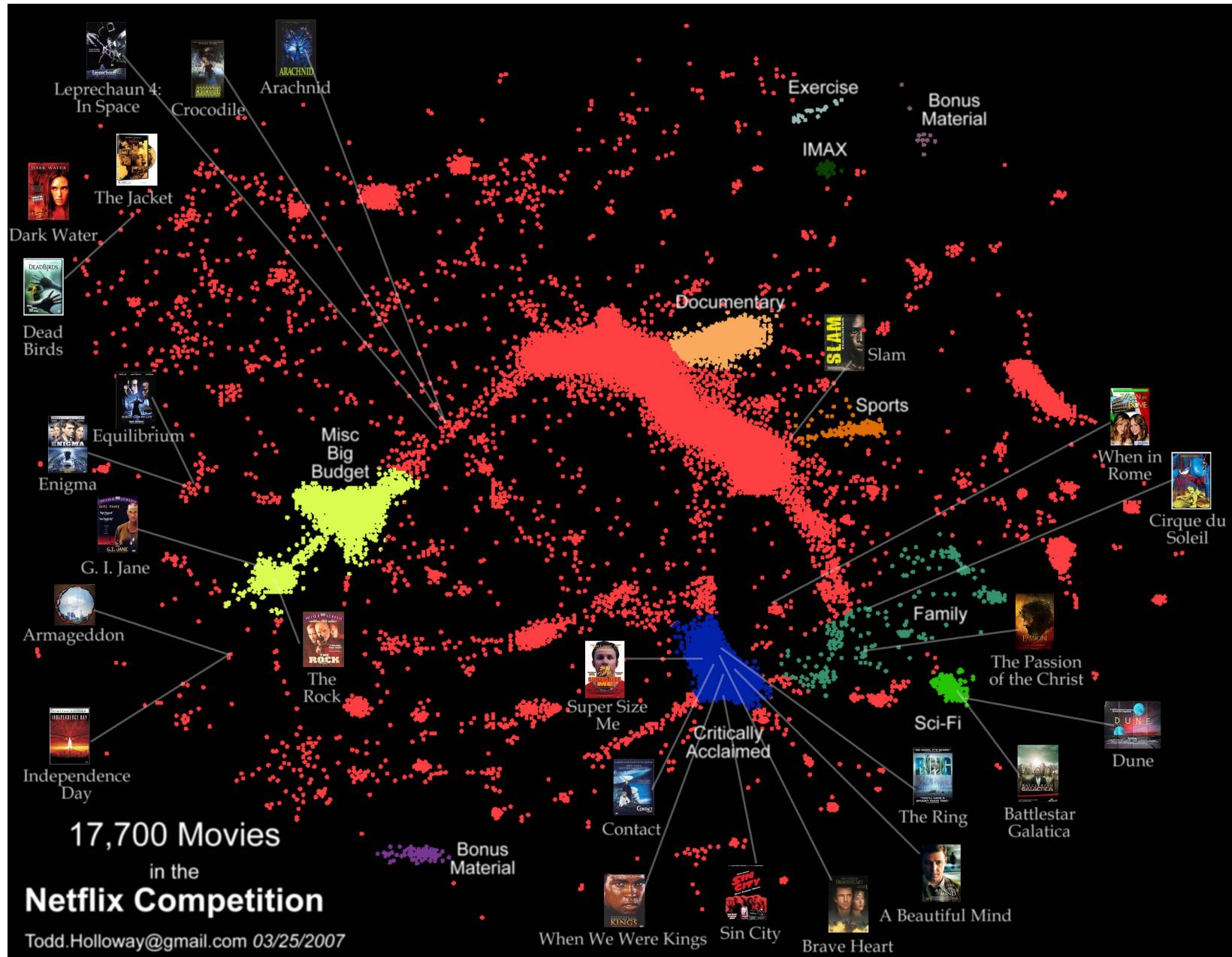
- In unsupervised learning, only raw input data is provided to the algorithm, which has to analyze the statistical properties of the data.



- The goal of **unsupervised learning** is to build a model or find useful representations of the data, for example:
  - finding groups of similar data and model their density (**clustering**).
  - reduce the redundancy of the input dimensions (**dimensionality reduction**).
  - finding good explanations / representations of the data (**latent data modeling**).
  - generate new data (**generative models**).

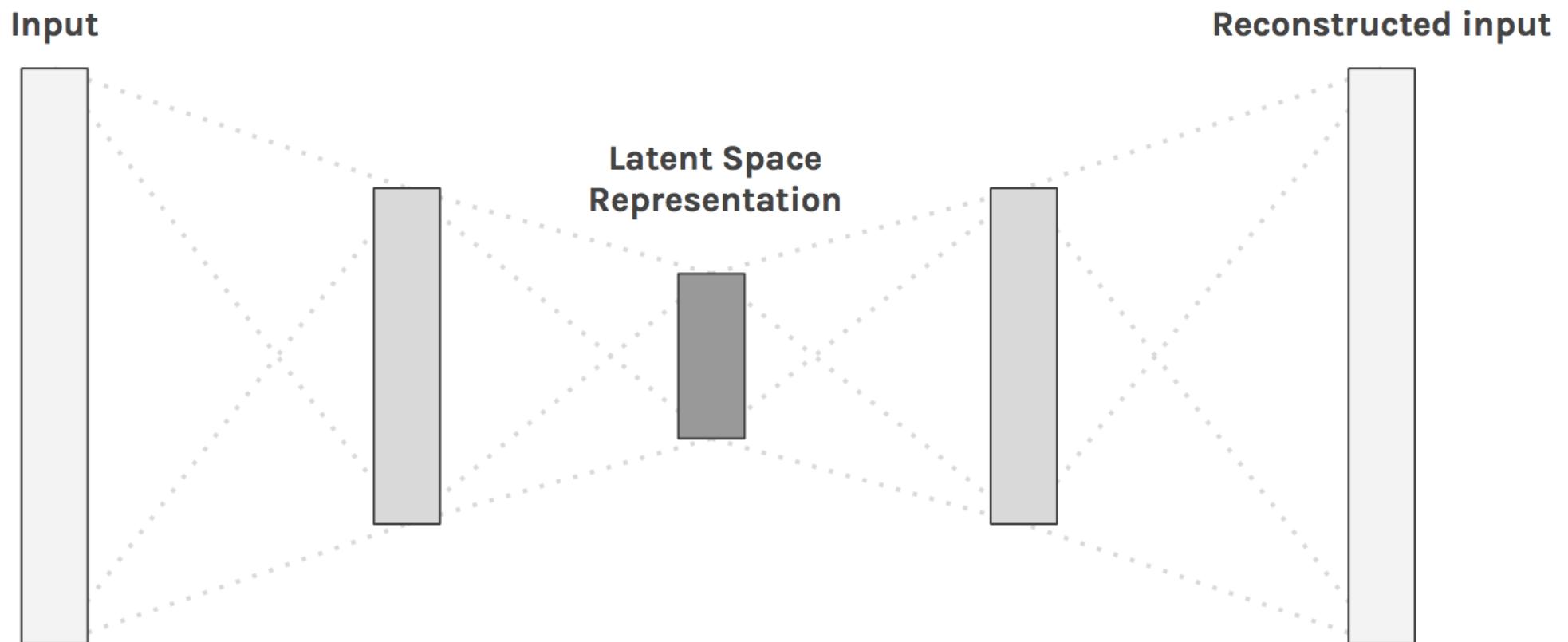
<https://learn.g2.com/supervised-vs-unsupervised-learning>

# Clustering: learning topologies in film preferences



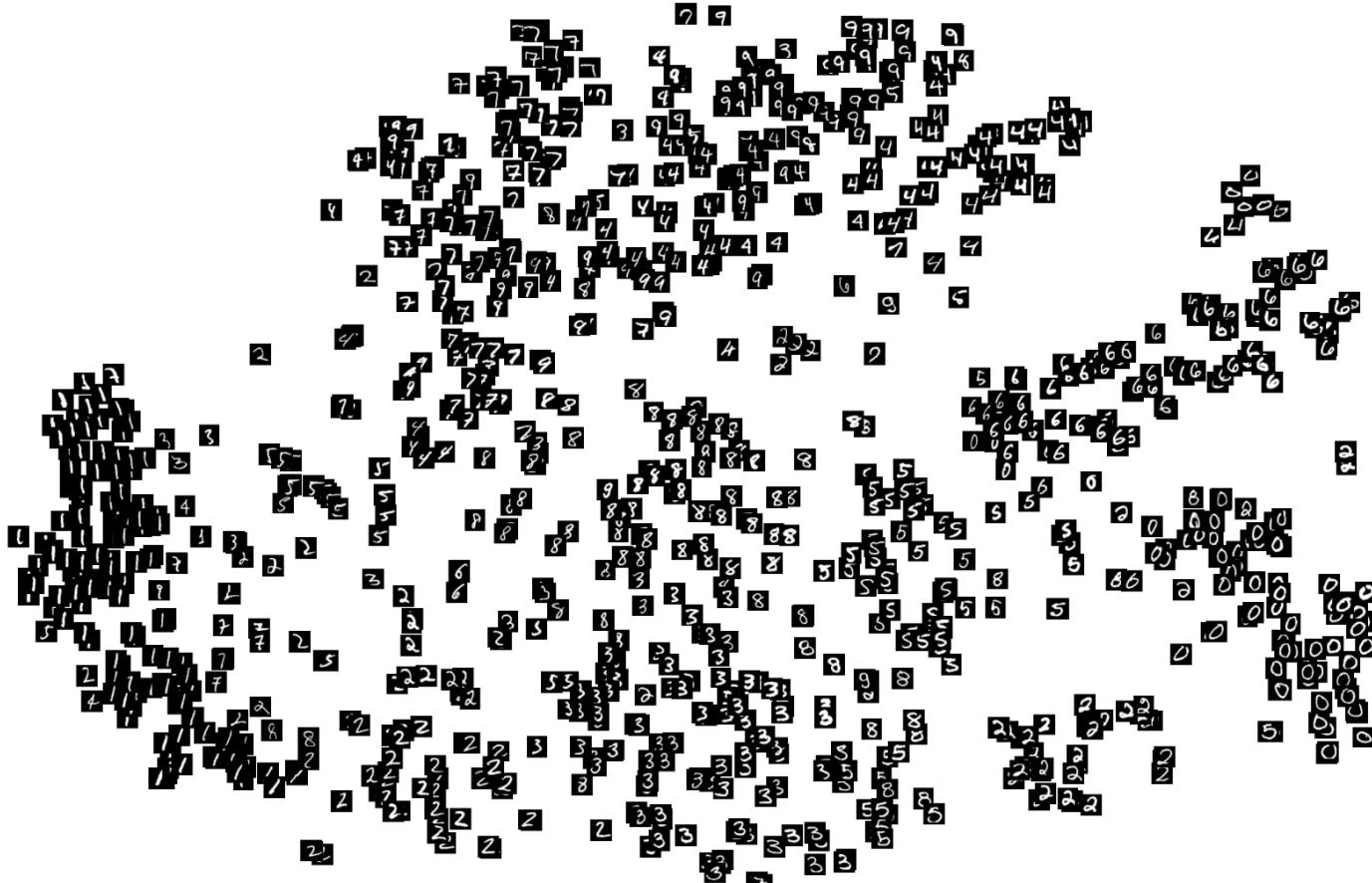
# Dimensionality reduction: finding the right latent space

- Images have a lot of dimensions (pixels), most of which are redundant.
- Dimensionality reduction techniques allow to reduce this number of dimensions by projecting the data into a **latent space**.
- **Autoencoders** are NN that learn to reproduce their inputs by compressing information through a bottleneck.



# Dimensionality reduction: visualization

- If the latent space has two or three dimensions, you can use dimensionality reduction to **visualize** your data.

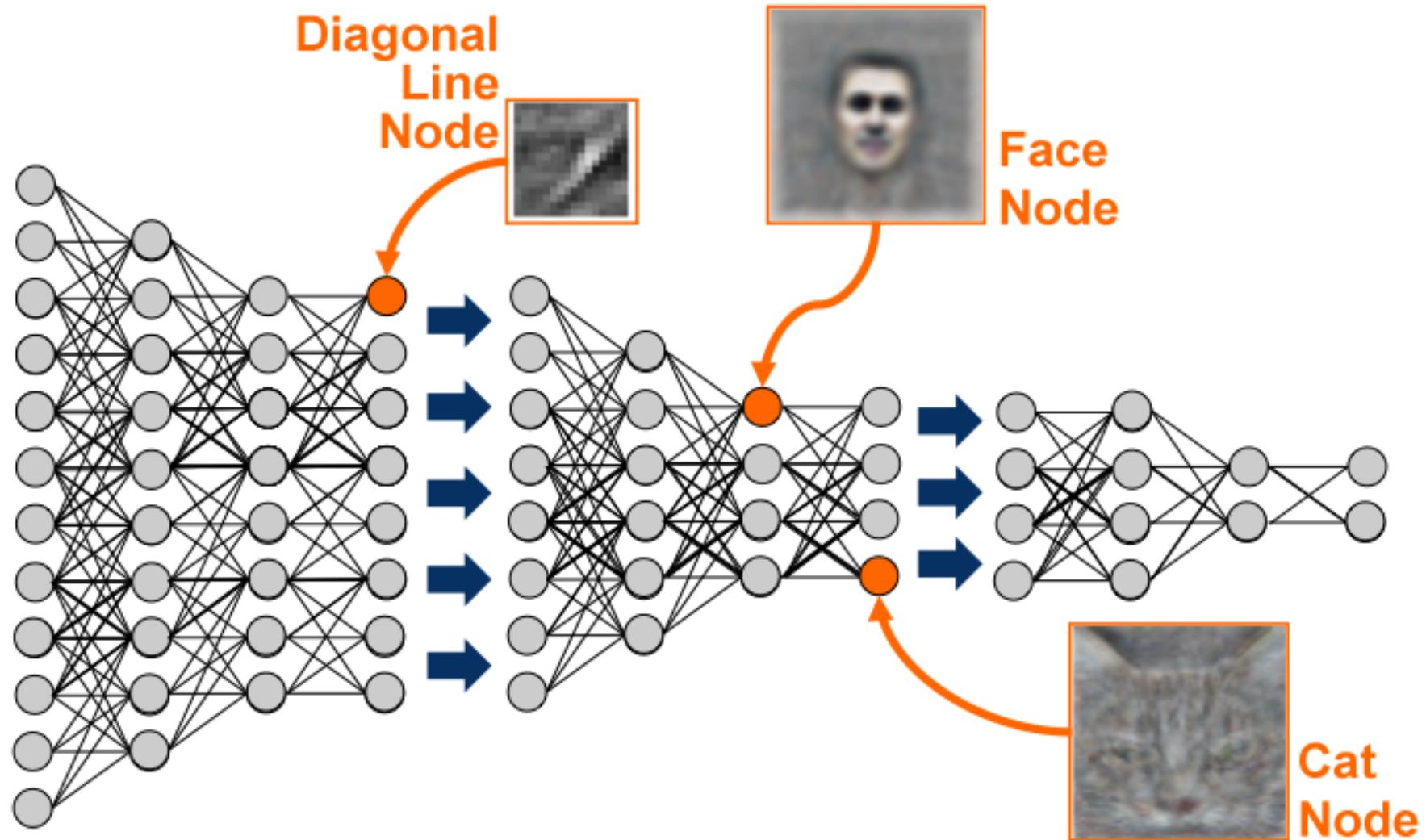


<https://hackernoon.com/latent-space-visualization-deep-learning-bits-2-bd09a46920df>

- Classical machine learning algorithms include PCA (principal component analysis) or t-SNE.
- NN autoencoders can also be used for visualization, e.g. UMAP.

# Feature extraction: self-taught learning

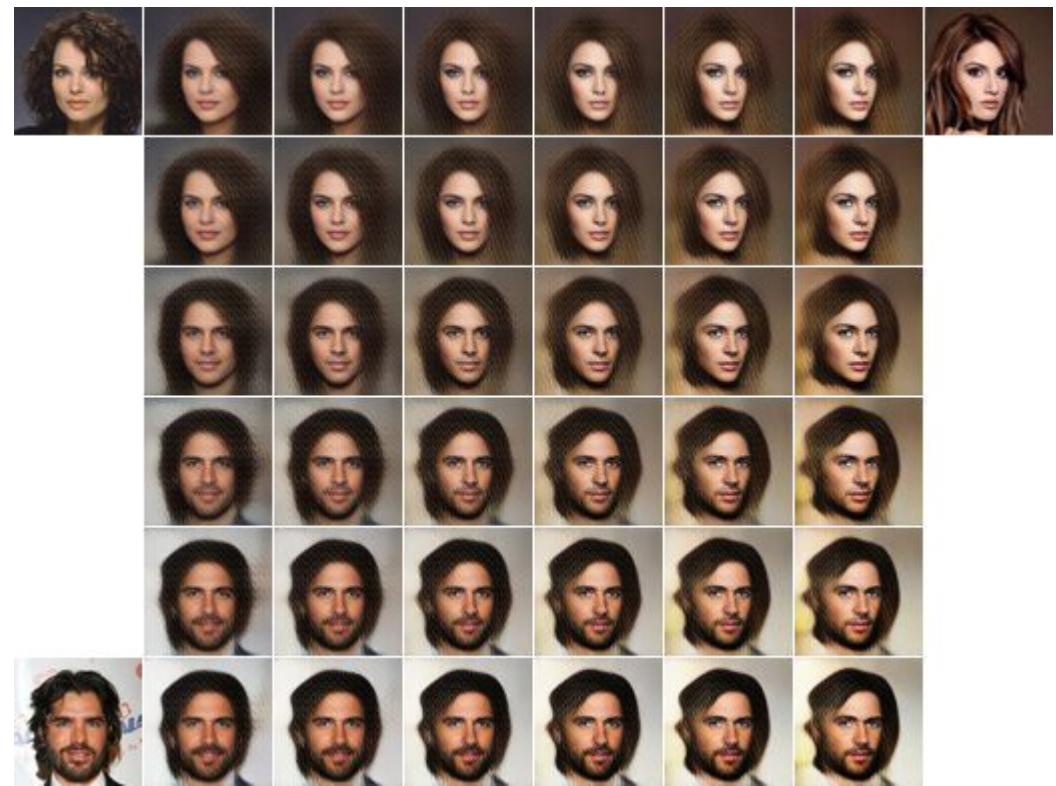
- Pretrain a neural network on huge unlabeled datasets (e.g. Youtube videos) before applying it to small-data supervised problems.



# Generative models

- If the latent space is well organized, you can even sample from it to generate new images using **variational autoencoders** (VAE).

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



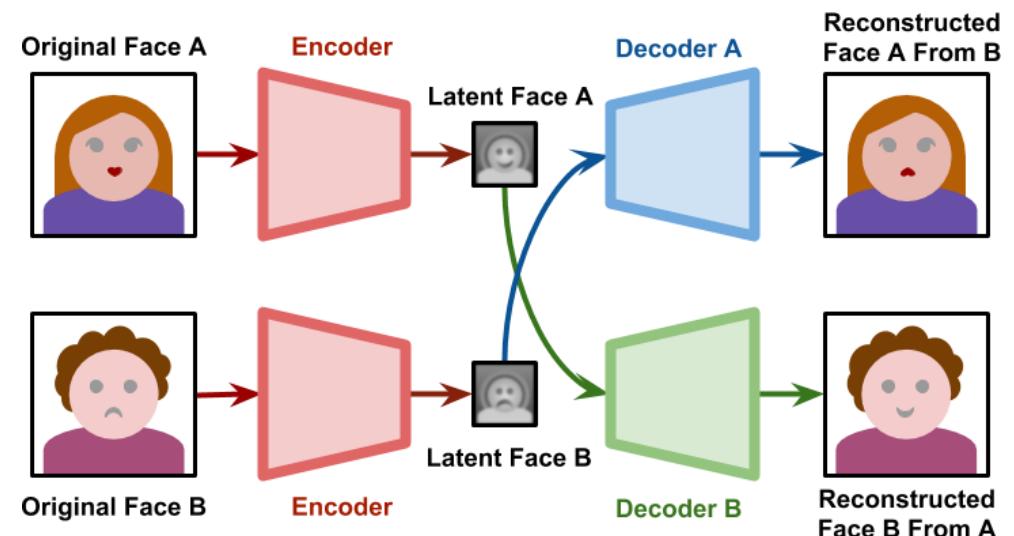
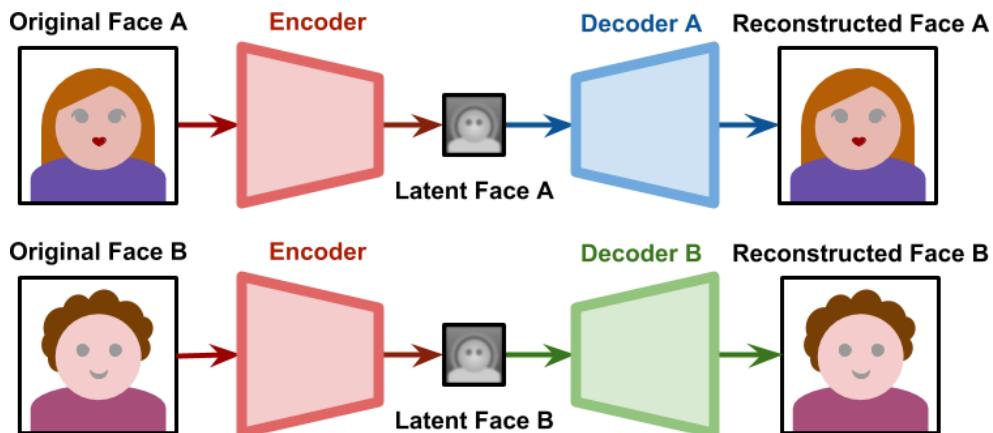
# DeepFake

Jim Carrey DeepFake [VFX Comparison]

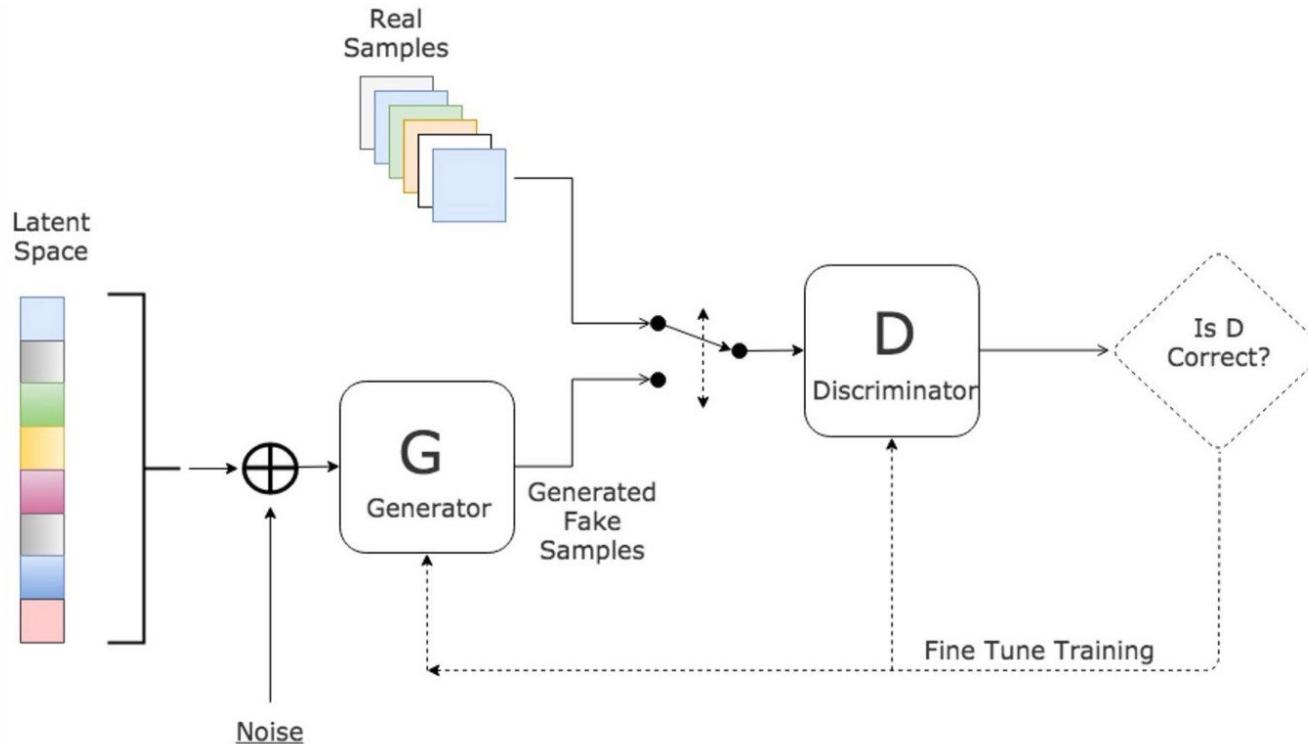


# DeepFake

- During training, each autoencoder learns to reproduce the face of one person.
- When generating the deepfake, the decoder of person B is used on the encoder of person A.



# Generative Adversarial Networks

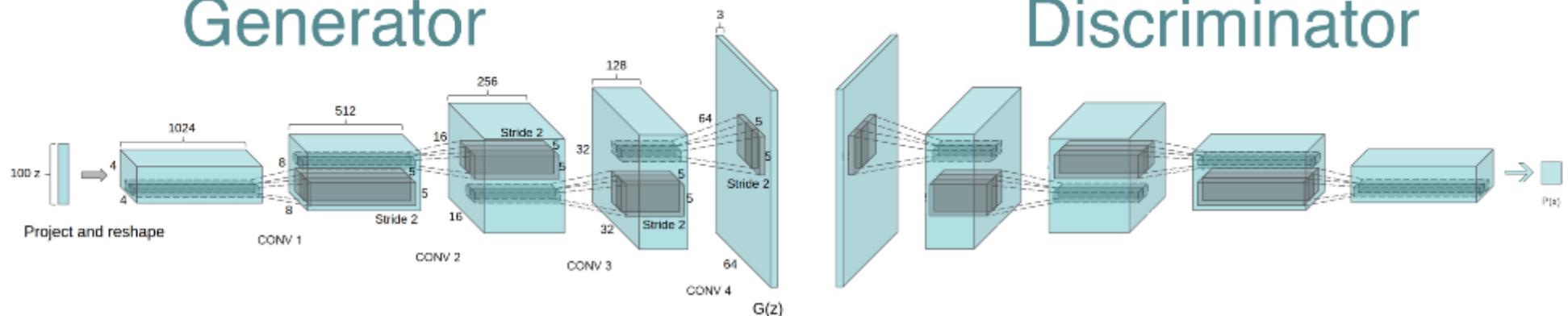


- A **Generative Adversarial Network (GAN)** is composed of two networks:
  - The **generator** learns to produce realistic images.
  - The **discriminator** learn to differentiate real data from generated data.
- Both compete to reach a Nash equilibrium:

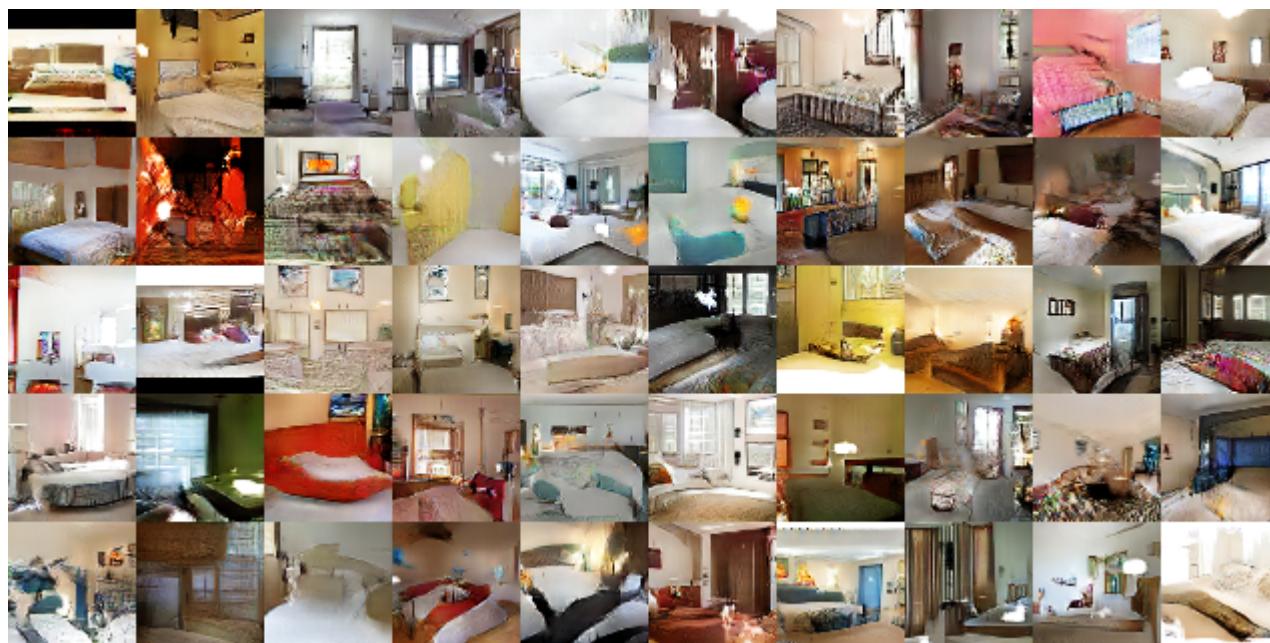
$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim P_z(z)} [\log(1 - D(G(z)))]$$

# DCGAN : Deep convolutional GAN

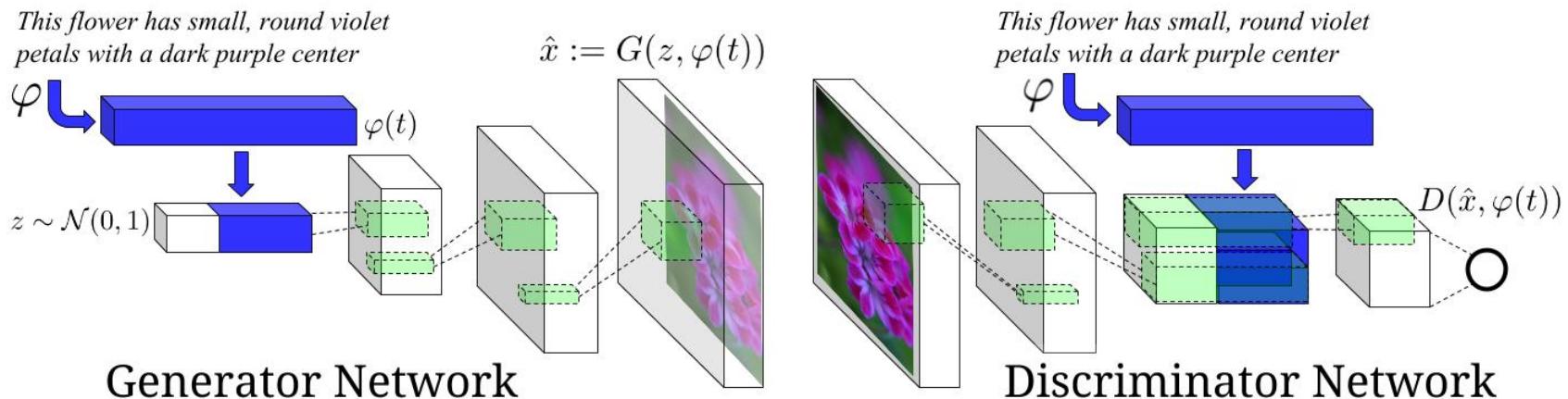
## Generator



## Discriminator

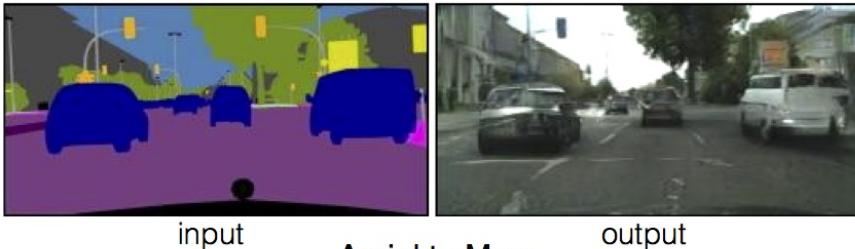


# cGAN : conditional GAN for image synthesis



# pix2pix : Image translation

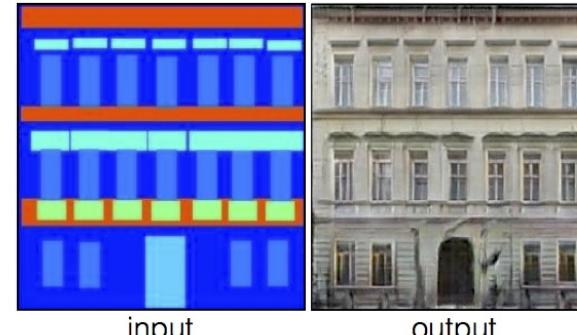
Labels to Street Scene



input

output

Labels to Facade



input

output

BW to Color



input

output

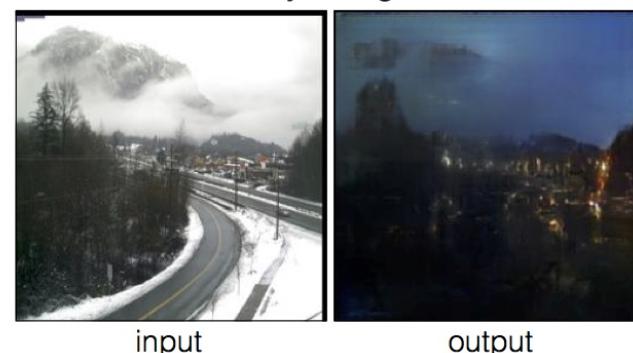
Aerial to Map



input

output

Day to Night



input

output

Edges to Photo

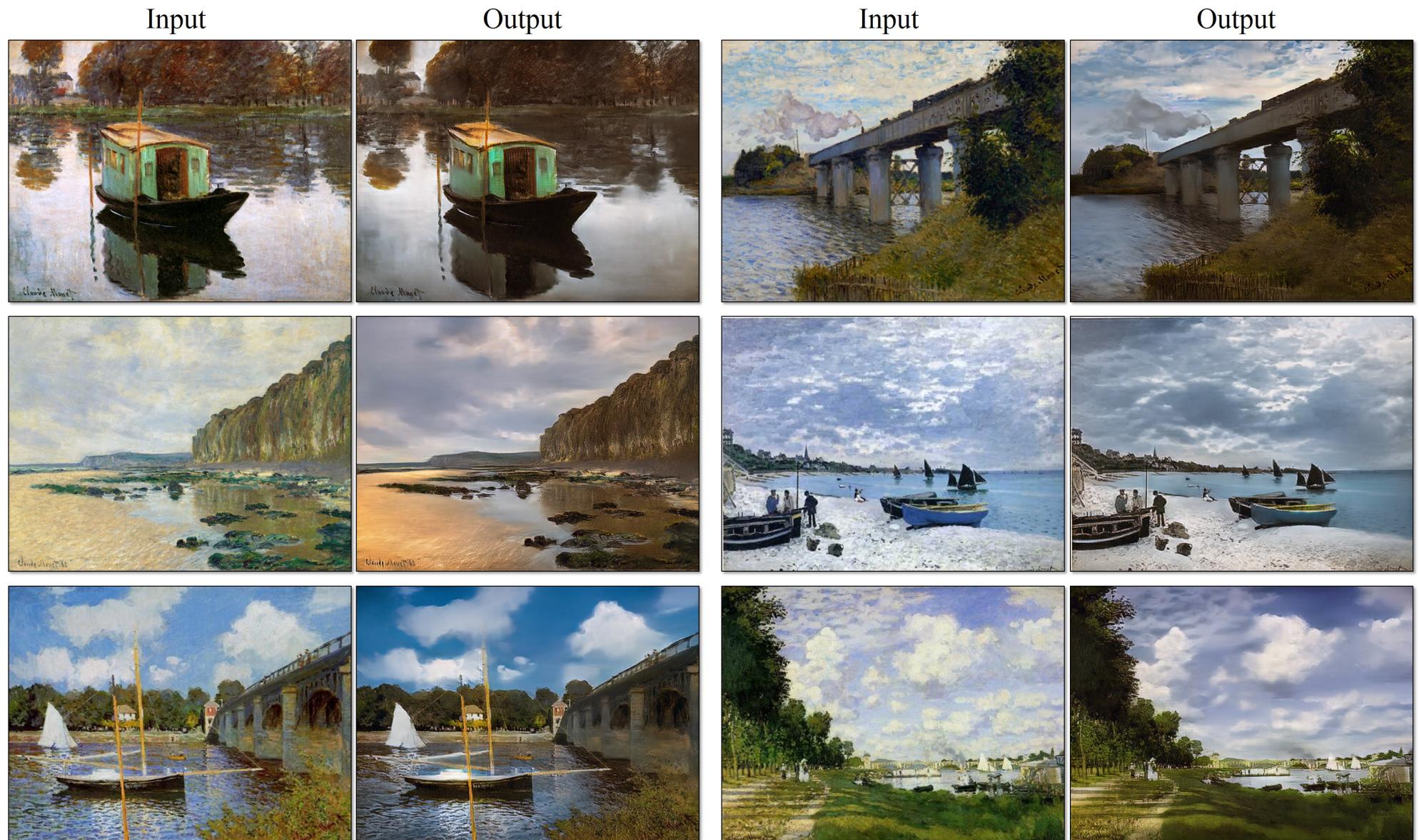


input

output

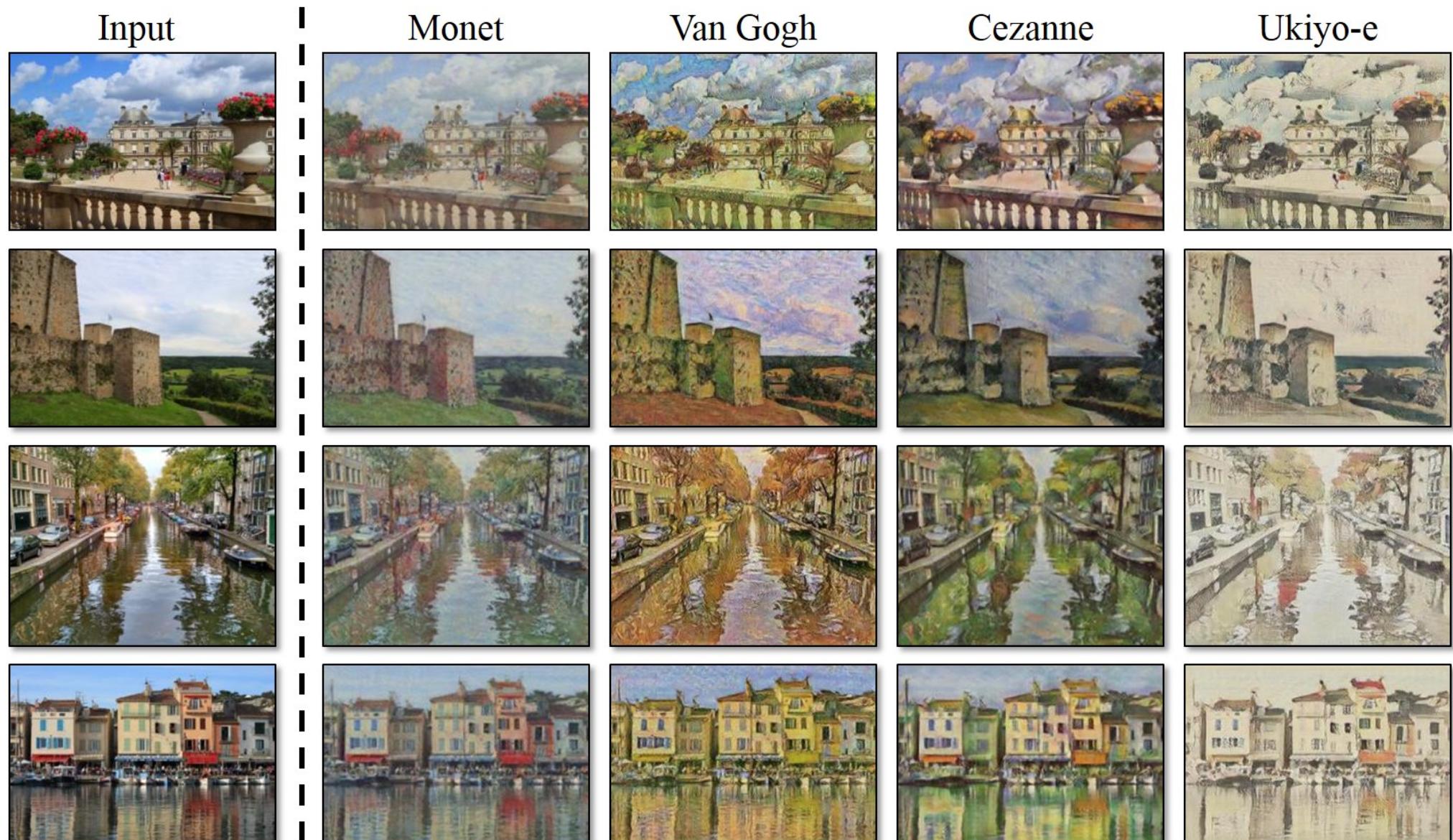
Source: <https://phillipi.github.io/pix2pix/>

# CycleGAN : Monet Paintings to Photo



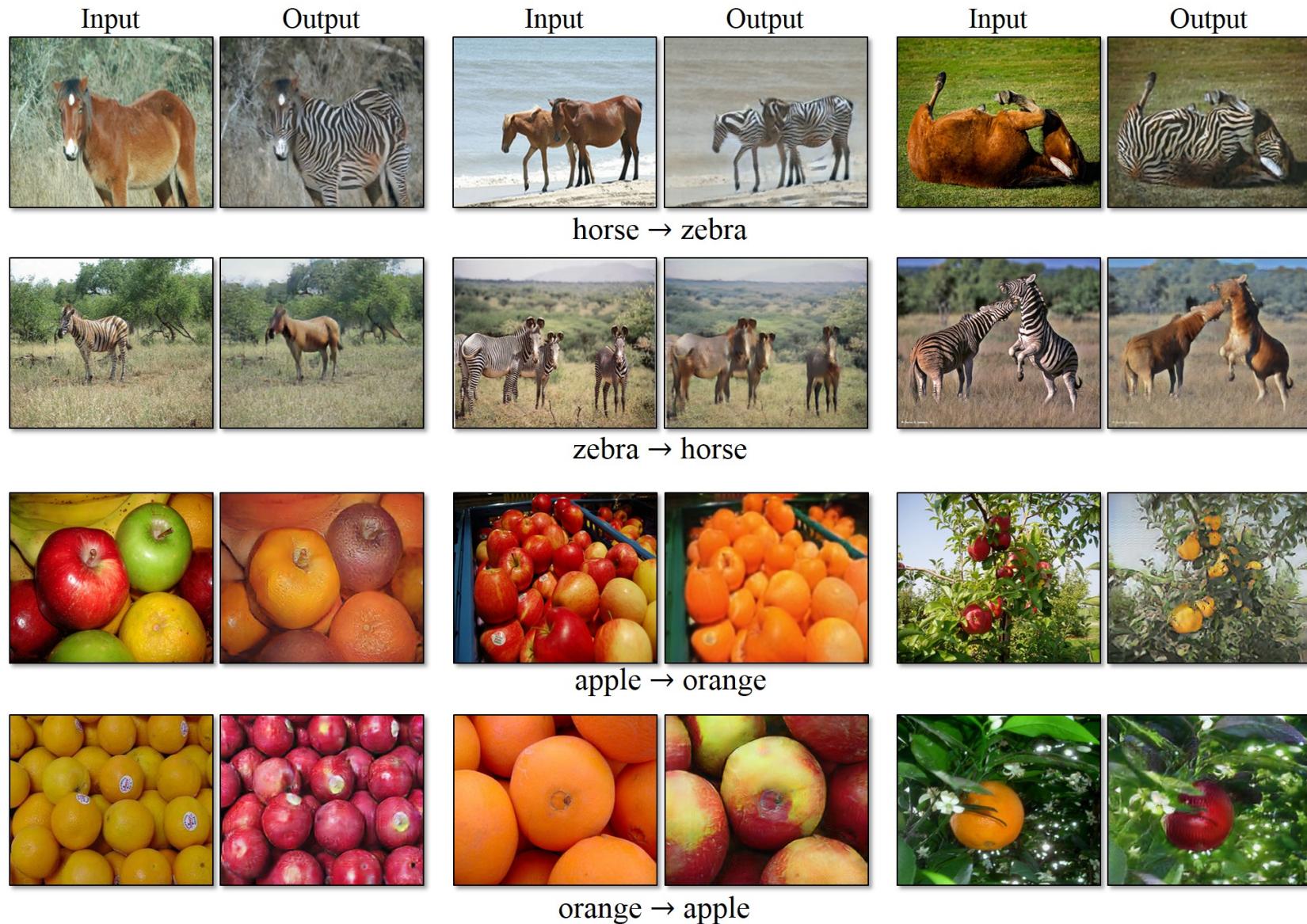
Source: <https://github.com/junyanz/CycleGAN>

# CycleGAN : Neural Style Transfer



Source: <https://github.com/junyanz/CycleGAN>

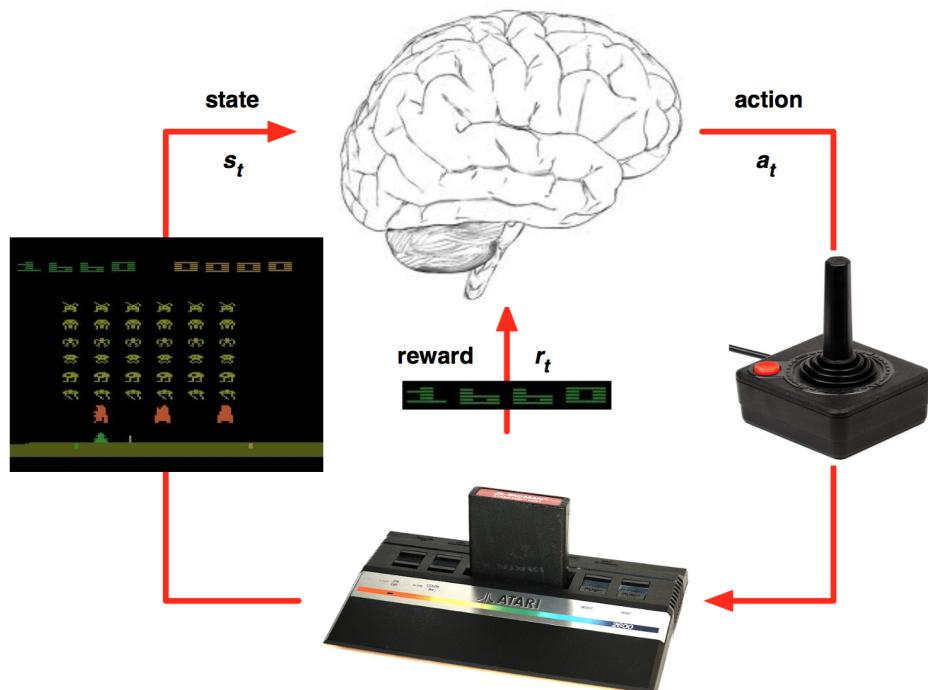
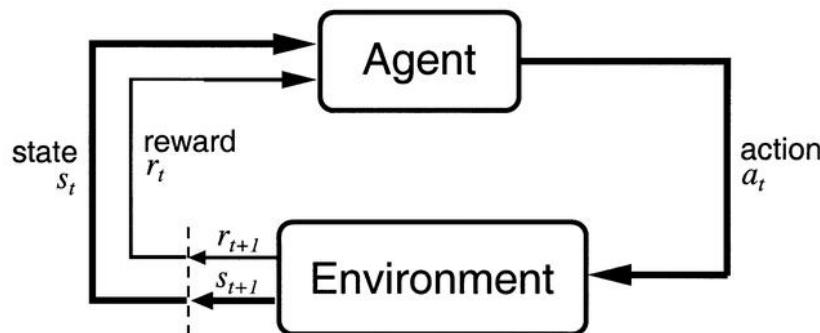
# CycleGAN : Object Transfiguration



Source: <https://github.com/junyanz/CycleGAN>

## **3 - Deep Reinforcement Learning**

# Reinforcement learning



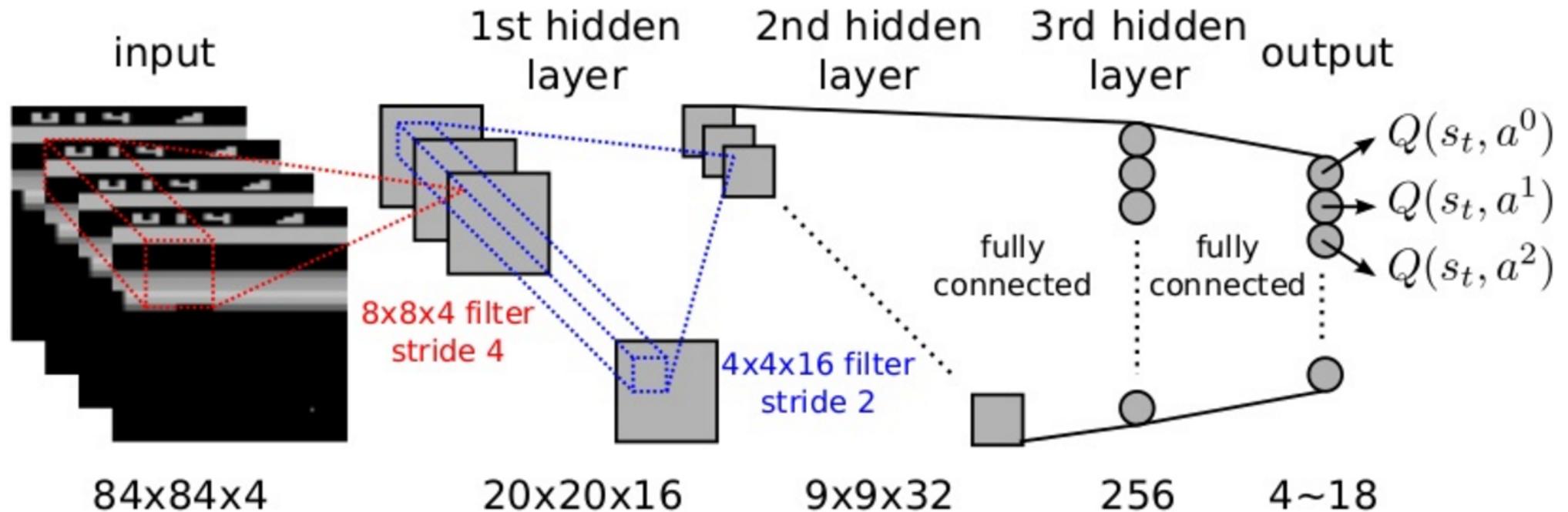
- **Supervised learning** allows to learn complex input/output mappings, given there is enough data.
- Sometimes we do not know the correct output, only whether the proposed output is correct or not (*partial feedback*).
- **Reinforcement Learning (RL)** can be used to learn by **trial and error** an optimal policy  $\pi(s, a)$ .
- Each action (=output) is associated to a **reward**.
- The goal of the system is to find a policy that maximizes the sum of the rewards on the **long-term** (return).

$$R(s_t, a_t) = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

- See the deep reinforcement learning course:

<https://www.tu-chemnitz.de/informatik/KI/edu/deeprl/>

## DQN : learning to play Atari games



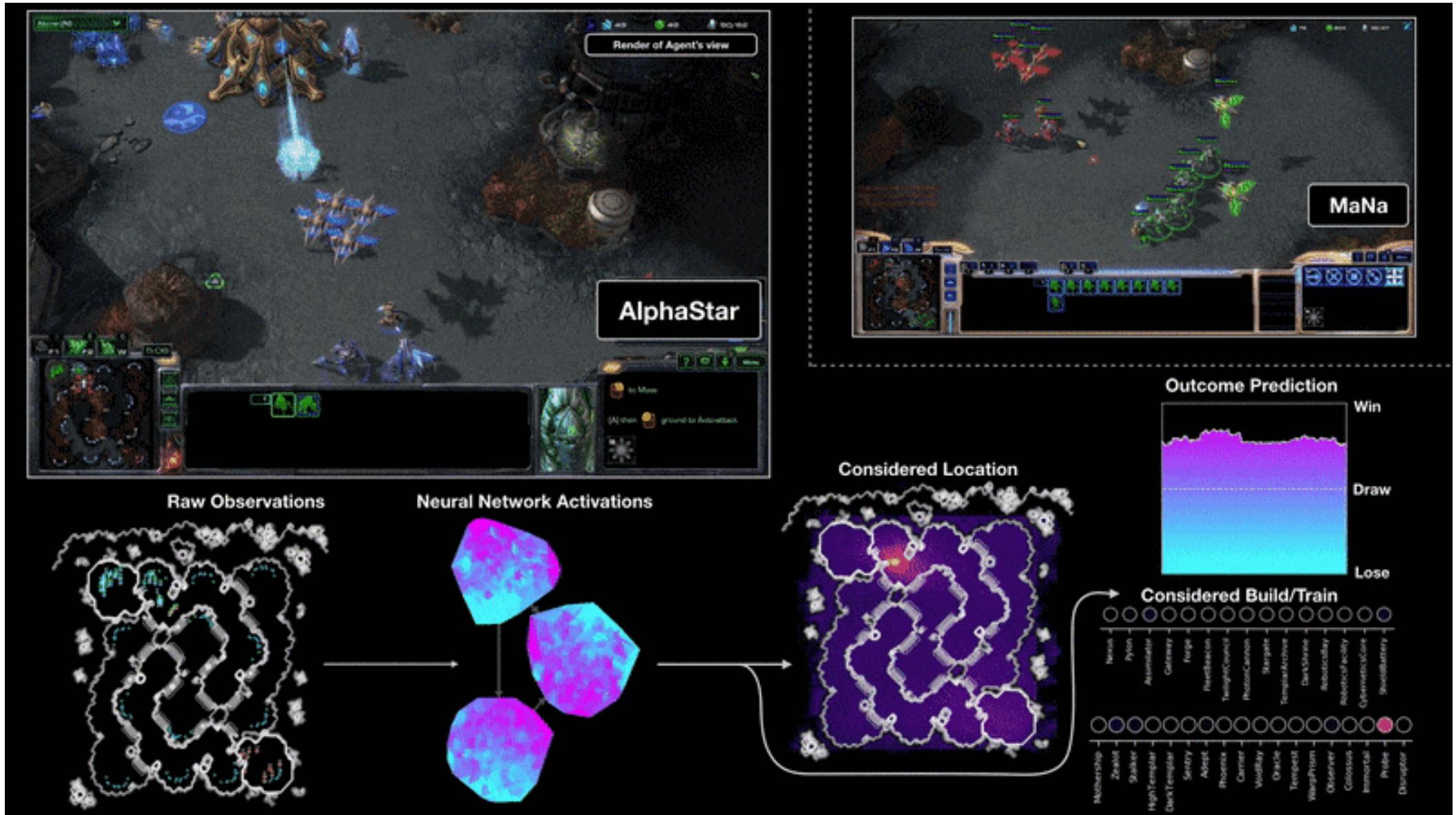
- A CNN takes raw images as inputs and outputs the probabilities of taking particular actions.
- Learning is only based on **trial and error**: what happens if I do that?
- The goal is simply to maximize the final score.

# DQN : learning to play Atari games

atari - DQN reinforcement learning experiments

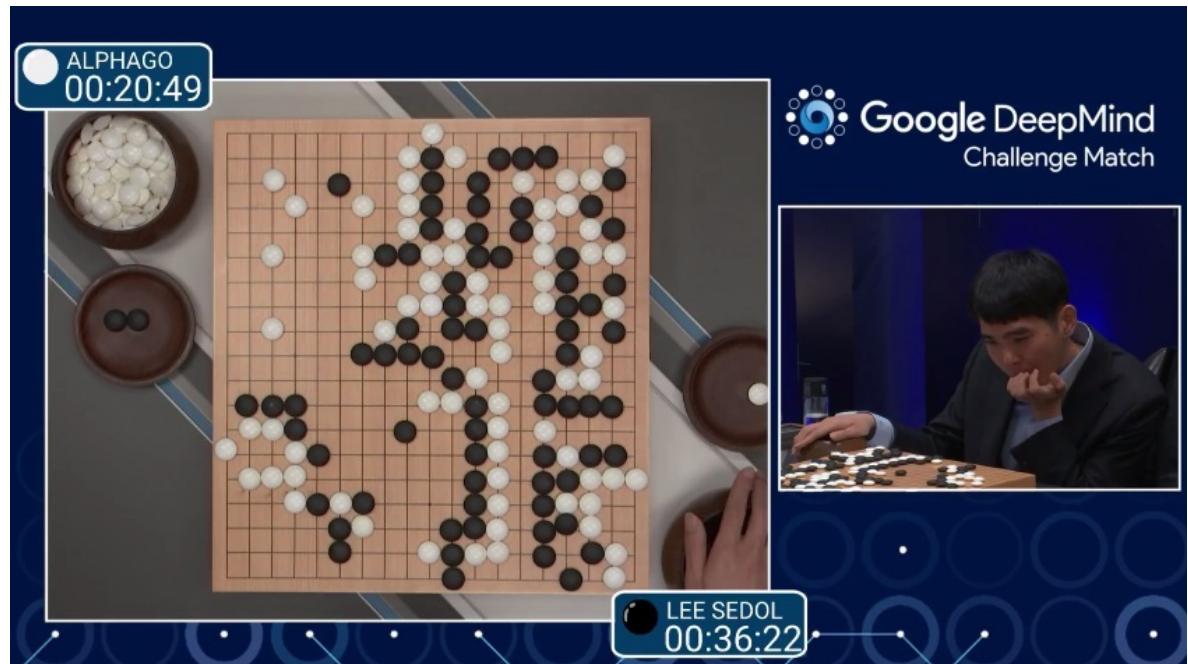
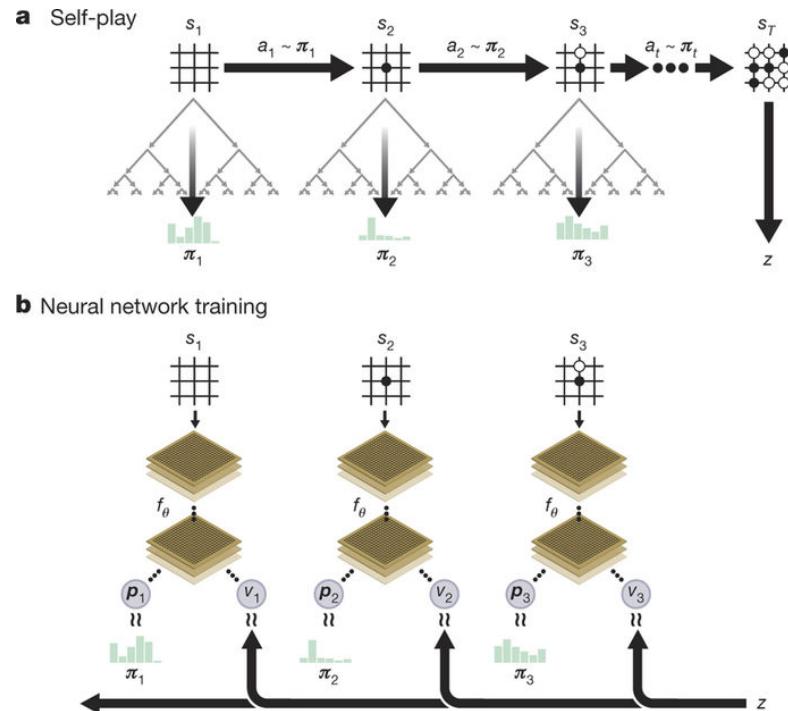


# AlphaStar : learning to play Starcraft II



<https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>

# Google Deepmind - AlphaGo



- In 2015, Google Deepmind surprised everyone by publishing **AlphaGo**, a Go AI able to beat the world's best players, including **Lee Sedol** in 2016, 19 times world champion.
- The RL agent discovers new strategies by using self-play: during the games against Lee Sedol, it was able to use **novel** moves which were never played before and surprised its opponent.
- The new version **AlphaZero** also plays chess and sokoban at the master level.

# Parkour

DeepMind Learns Parkour



# Dexterity

Learning Dexterity



# Autonomous driving

Learning to drive in a day



# Neurocomputing syllabus

## 1. Linear learning machines

- Optimization, Gradient Descent
- Linear regression and classification
- Multi-class classification
- Learning theory, Cross-validation

## 2. Neural networks

- Multi-layer perceptron
- Backpropagation algorithm
- Regularization, Batch Normalization

## 3. Convolutional neural networks

- Convolutional layer, pooling
- Transfer learning
- Object detection (Fast-RCNN, YOLO)
- Semantic segmentation

## 4. Autoencoders and generative models

- Auto-encoders
- Variational autoencoders
- Restricted Boltzmann machines
- Generative adversarial networks

## 5. Recurrent Neural Networks

- RNN
- LSTM / GRU
- Attention-gated networks

## 6. Self-supervised learning

- Transformers
- Contrastive learning

## 7. Outlook

# Literature

- **Deep Learning.** *Ian Goodfellow, Yoshua Bengio & Aaron Courville*, MIT press.

<http://www.deeplearningbook.org>

- **Neural Networks and Learning Machines.** *Simon Haykin*, Pearson International Edition.

<http://www.pearsonhighered.com/haykin>

- **Deep Learning with Python.** *Francois Chollet*, Manning.

<https://www.manning.com/books/deep-learning-with-python>

- **The Elements of Statistical Learning: Data Mining, Inference, and Prediction,** *Trevor Hastie, Robert Tibshirani & Jerome Friedman*, Springer.

[https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII\\_print12.pdf](https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf)

## But also

- The machine learning course of Andrew Ng (Stanford at the time) hosted on Coursera is great for beginners:

<https://www.coursera.org/learn/machine-learning>

- His advanced course on deep learning allows to go further:

<https://www.coursera.org/specializations/deep-learning>

- The machine learning course on EdX focuses on classical ML methods and is a good complement to this course:

<https://www.edx.org/course/machine-learning>

- <https://medium.com> has a lot of excellent blog posts explaining AI-related topics, especially:

<https://towardsdatascience.com/>

- The d2l.ai online book is a great resource, including programming exercises:

<http://d2l.ai/index.html>