# VC 19/20

# Introduction to Deep Learning and Convolutional Neural Networks

Mestrado em Ciência de Computadores

Mestrado Integrado em Engenharia de Redes e Sistemas Informáticos

***Francesco Renna***

# Outline

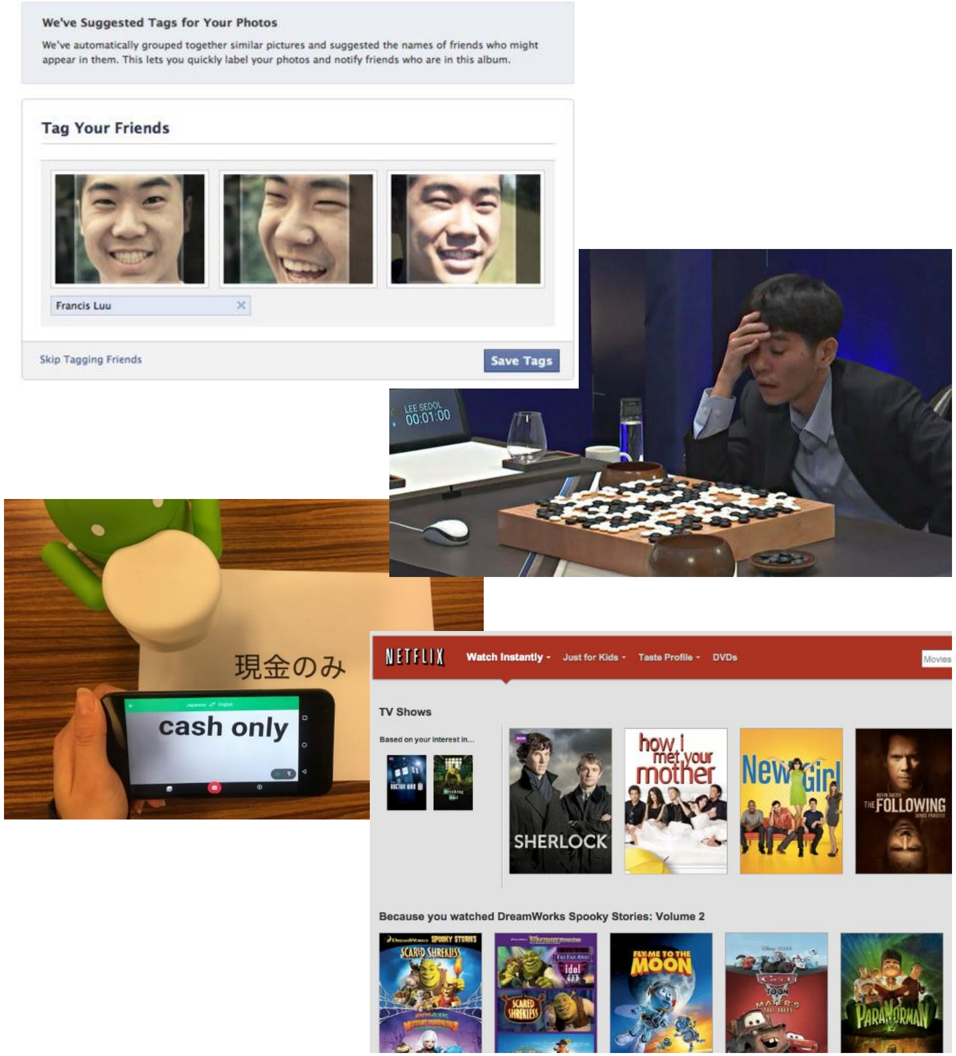
- What is deep learning?
- Artificial neural networks
- Convolutional neural networks
- Biomedical application examples
  - Image classification
  - Image segmentation
  - Image reconstruction
- Application challenges

# Outline

- What is deep learning?
- Artificial neural networks
- Convolutional neural networks
- Biomedical application examples
  - Image classification
  - Image segmentation
  - Image reconstruction
- Application challenges

# Deep learning:
# did you hear about that?

- Google image recognition
- Facebook face recognition
- Google translator
- DeepMind AlphaGo player
- Netflix, Amazon, Spotify recommendation engines
- Image colorization
- Image caption generation
- Sentiment analysis
- Etc…



**We've Suggested Tags for Your Photos**

We've automatically grouped together similar pictures and suggested the names of friends who might appear in them. This lets you quickly label your photos and notify friends who are in this album.

**Tag Your Friends**

# What is deep learning?

- It is a specific area of machine learning
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
- Idea (supervised learning): learn how to make decisions, perform a task, from examples

dog

cat

dog or cat?

# How to extract information from the raw data?

**Sensor**

Acquire the data, observations to be classified or described

**Feature Extraction**

Compute numeric or symbolic information starting from the data:
e.g., color, shape, texture, etc.

**Classifier**

Classify or describe the observation, relying on the extracted features

# More specifically

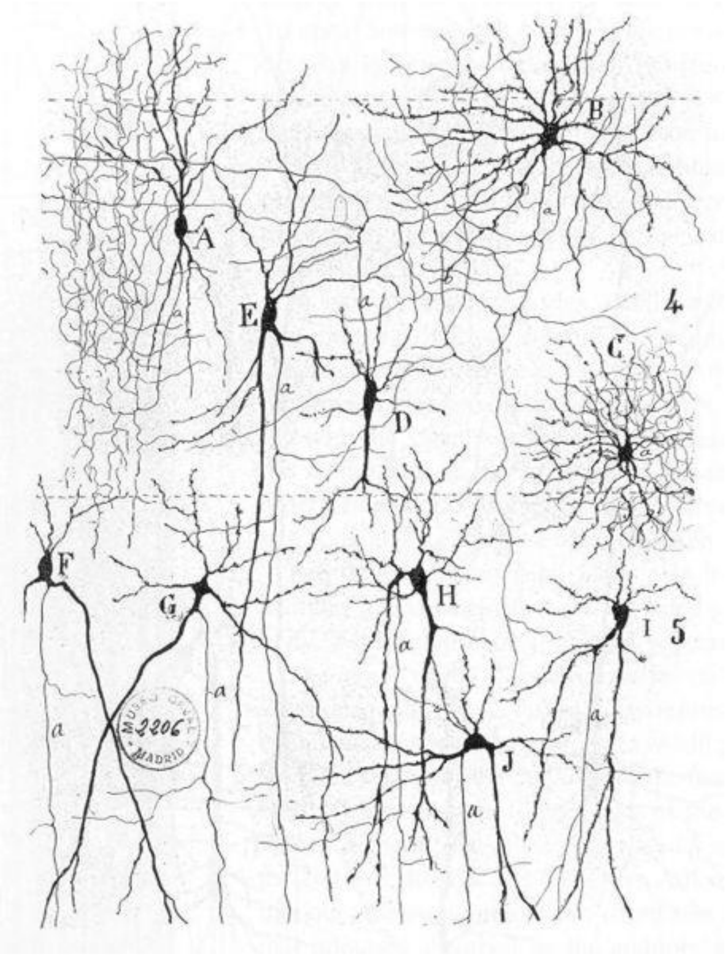
- Deep learning refers to a class of learning algorithms
- They are based on the use of a specific kind of classifiers: neural networks (NNs)

# Outline

- What is deep learning?
- Artificial neural networks
- Convolutional neural networks
- Biomedical application examples
  - Image classification
  - Image segmentation
  - Image reconstruction
- Application challenges

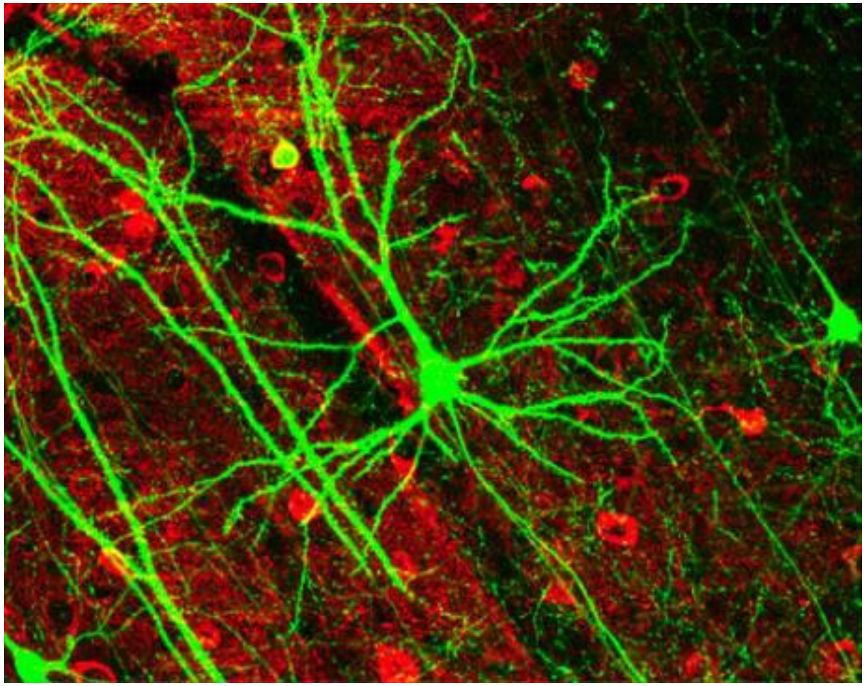
# Biological Neural Networks

- Neuroscience:
  - Population of physically inter-connected neurons.
- Includes:
  - Biological **Neurons**
  - Connecting **Synapses**
- The human brain:
  - 100 billion neurons
  - 100 trillion synapses



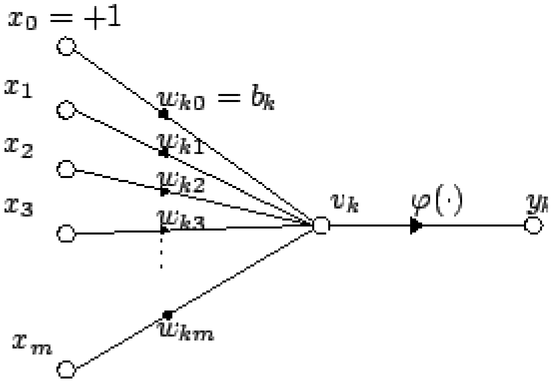
# Biological Neuron

- Neurons:
  - Have *K* inputs (*dendrites*).
  - Have 1 output (*axon*).
  - If the sum of the input signals surpasses a *threshold*, sends an *action potential* to the axon.
- Synapses
  - Transmit electrical signals between neurons.



# Artificial Neuron

- Also called the **McCulloch-Pitts neuron.**
- Passes a **weighted sum of inputs**, to an **activation function**, which produces an **output** value.



$$y_k = \varphi\left(\sum_{j=0}^{m} w_{kj} x_j\right)$$

W. McCulloch, W. Pitts, (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 7:115 - 133.

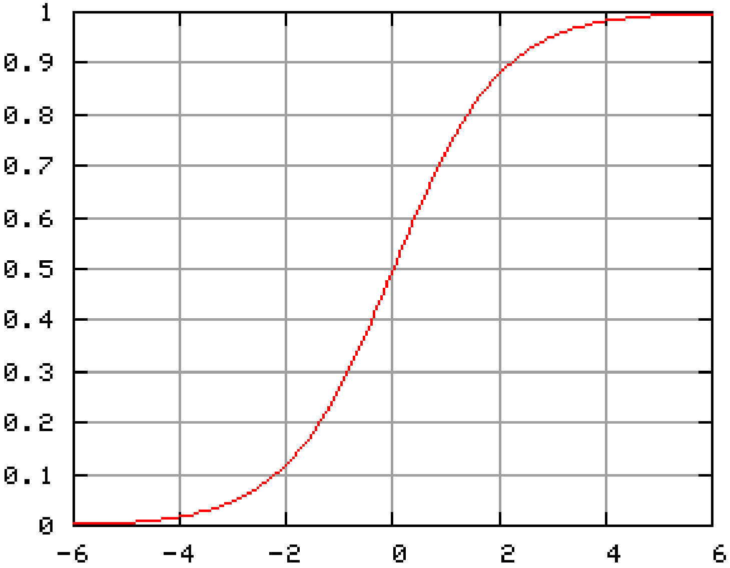
# Sample activation functions

- Rectified Linear Unit (ReLU)

$$y = \begin{cases} u, & \text{if } u \geq 0 \\ 0, & \text{if } u < 0 \end{cases}, \quad u = \sum_{i=1}^{n} w_i x_i$$

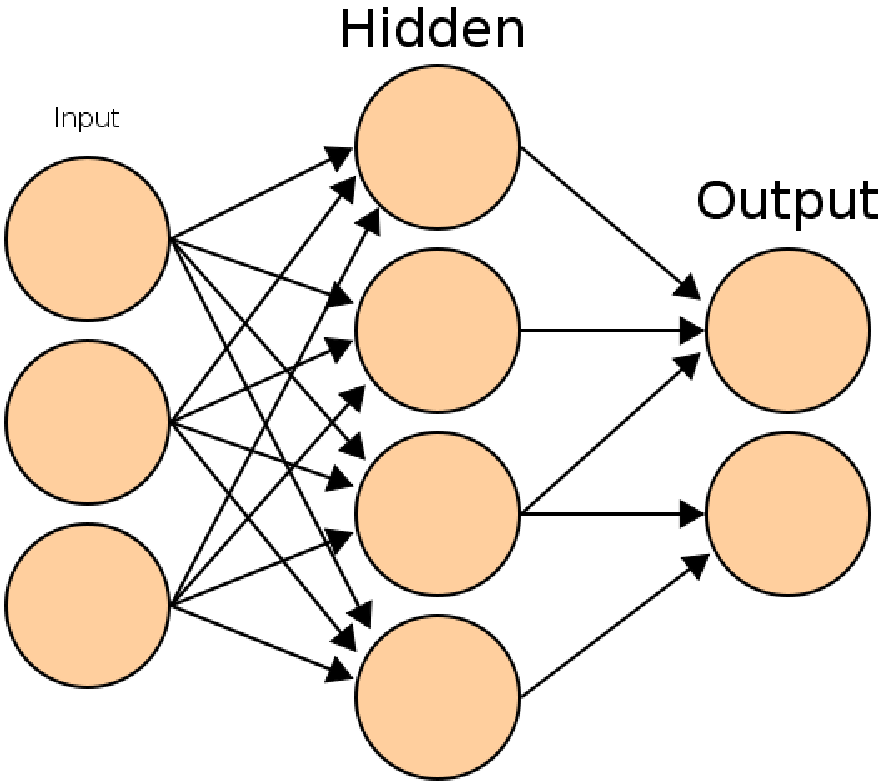
- Sigmoid function

$$y = \frac{1}{1+e^{-u}}$$



# Artificial Neural Network

- Commonly refered as **Neural Network**.
- Basic principles:
  - One neuron can perform a simple decision.
  - Many **connected** neurons can make more **complex decisions**.



# Characteristics of a NN

- Network configuration
  - – How are the neurons inter-connected?
  - – We typically use *layers* of neurons (input, output, hidden).
- Individual neuron parameters
  - – Weights associated with inputs.
  - – Activation function.
  - – Decision *thresholds*.

How do we find these values?

# Learning paradigms

- We can define the network configuration.
- How do we define neuron ***weights*** and ***decision thresholds***?
  - **Learning** step.
  - We **train** the NN to classify what we want.
  - (Supervised learning): We need to have access to a set of training data for which we know the correct class/answer

# Learning

- We want to obtain an **optimal solution** given a set of **observations**.
- A **cost function** measures how close our solution is to the **optimal solution**.
- Objective of our learning step:
  - Minimize the **cost function**.

Backpropagation Algorithm

# In formulas

Network output: $\mathrm{Out}(x) = \varphi(\sum_m w_{nm}^{(L)} \varphi(\dots \varphi(\sum_j w_{\ell j}^{(2)} \varphi(\sum_k w_{jk}^{(1)} x_k)))$

Training set: $\{(\overset{\text{input}}{x_i}, \overset{\text{label}}{y_i})\}_{i=1,\dots,N}$

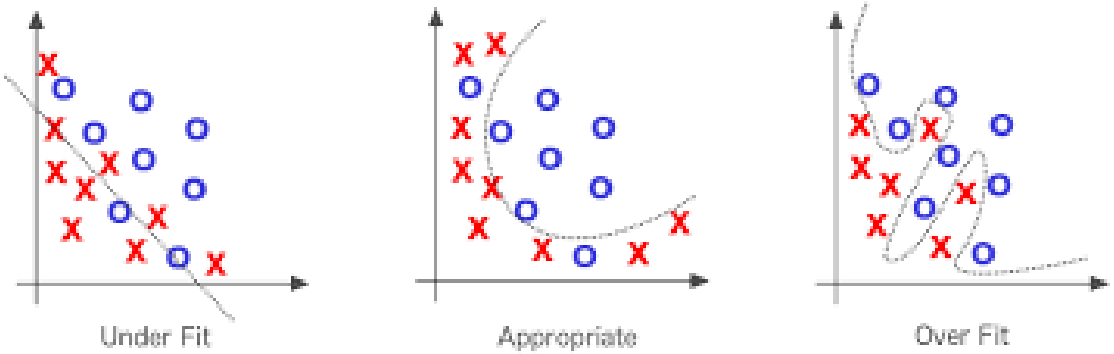
Optimization: find $[w_{jk}^{(1)}, w_{\ell j}^{(2)}, \dots, w_{nm}^{(L)}]$ such that

$$\text{minimize} \sum_{i=1}^{N} \mathrm{Loss}(\mathrm{Out}(x_i), y_i)$$

It is solved with (variants of) the gradient descent, where gradients are computed via the backpropagation algorithm

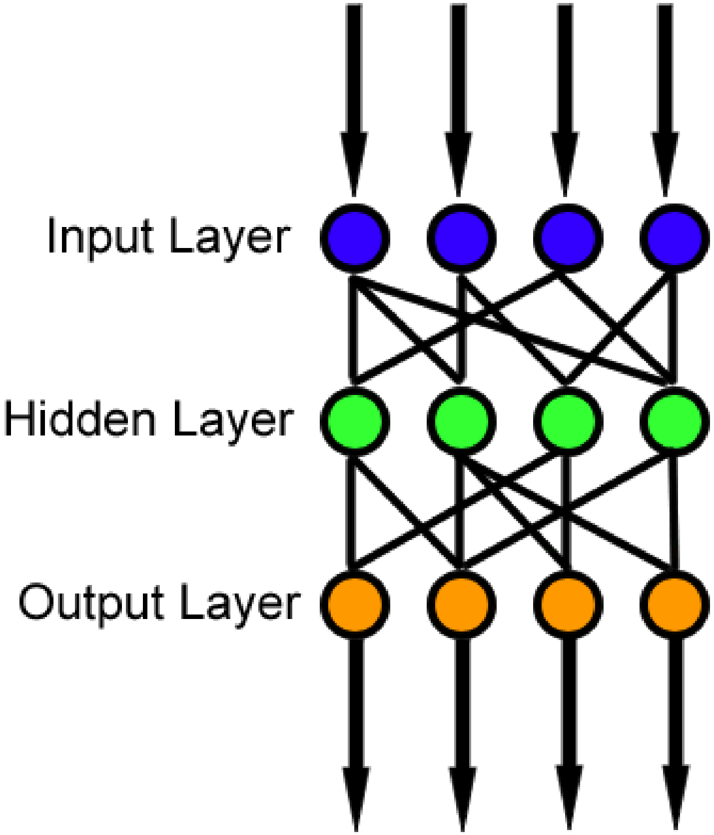
# Warnings!

- Is the NN too simple for the data?
  - Underfitting: cannot capture data behavior
- Is the NN too complex for the data?
  - Overfitting: fit perfectly training data, but will not generalize well on unseen data



# Feedforward neural network

- Simplest type of NN.
- Has no *cycles*.
- Input layer
  - Need as many neurons as coefficients of my *feature vector*.
- Hidden layers.
- Output layer
  - Classification results.



# Deep learning = Deep neural networks

- Deep = high number of hidden layers
  - Learn a larger number of parameters!
- It has been recently (~ in the last 6 years) possible since we have:
  - Access to big amounts of (training) data
  - Increased computational capabilities (e.g., GPUs)

# Outline

- What is deep learning?
- Artificial neural networks
- Convolutional neural networks
- Biomedical application examples
  - Image classification
  - Image segmentation
  - Image reconstruction
- Application challenges

# Convolutional neural networks (CNNs)

- Feedforward neural networks
- Weight multiplications are replaced by convolutions (filters)
- **Change of paradigm**: can be directly applied to the raw signal, without computing first *ad hoc* features
- Features are learnt automatically!!

# End-to-end learning

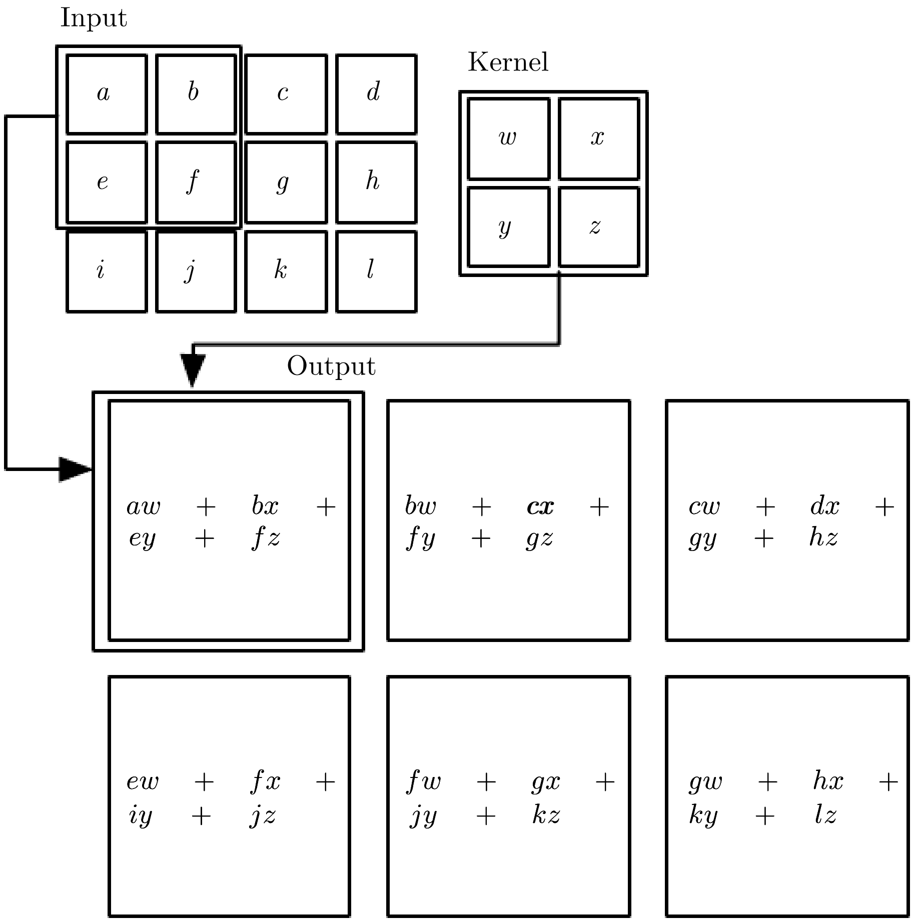
Sensor

Acquire the data, observations to be classified or described

Convolutional neural network

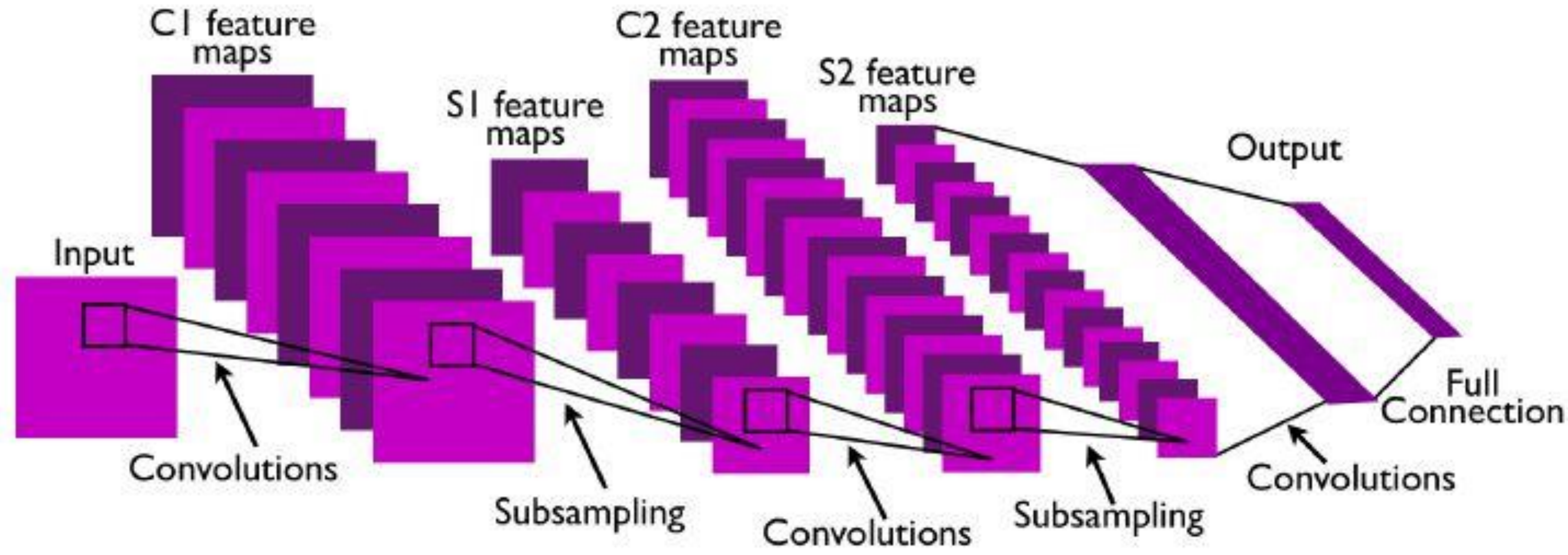
Classify or describe the observation, automatically extracting (learnt) features

# Convolution



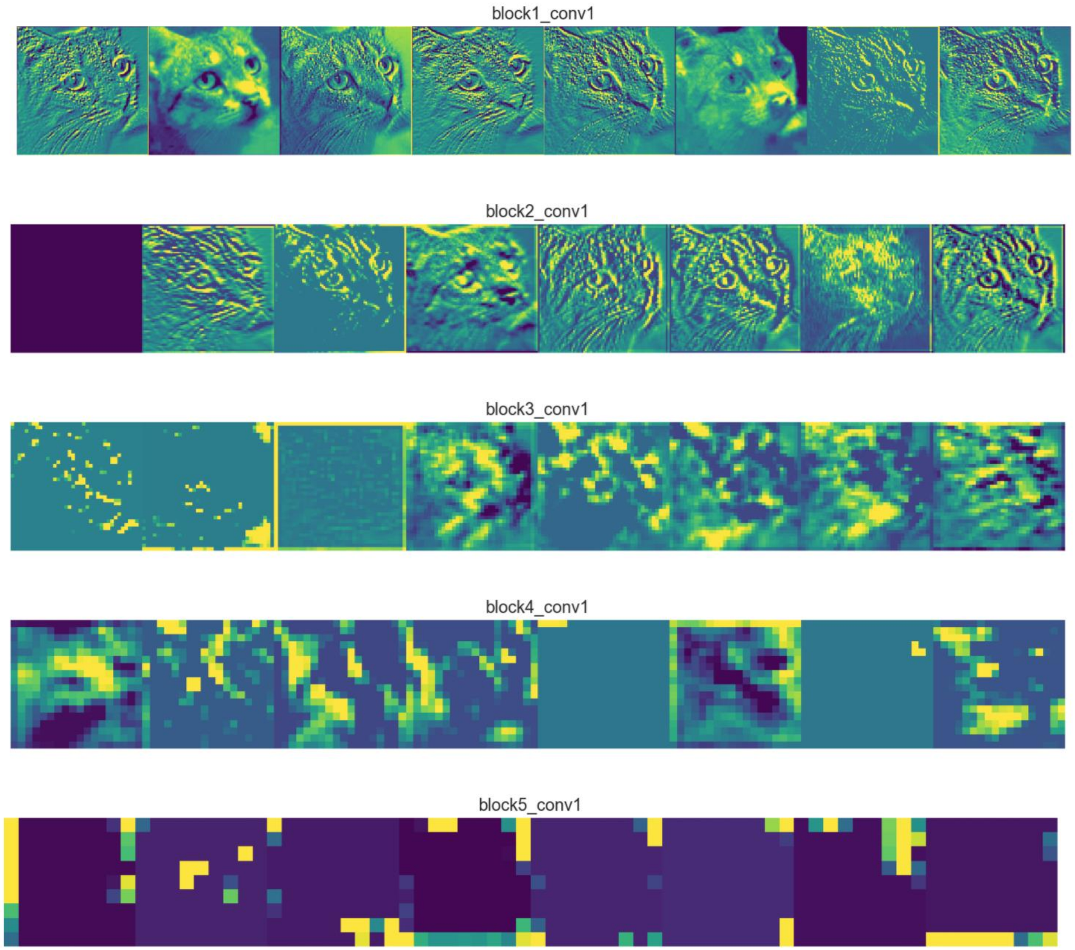
I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

# CNN example



- Convolutional layers, followed by nonlinear activation and subsampling
- Output of hidden layers (feature maps) = features learnt by the CNN
- Before classification, fully connected layers (as in “standard” NN)

# Automatically learnt features



Retain most information (edge detectors)

Towards more abstract representation

Encode high level concepts

Sparser representations:
Detect less (more abstract) features

https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

# CNN - Properties

- Reduced amount of parameters to learn (local features)
- More efficient than dense multiplication
- Specifically thought for images or data with grid-like topology
- Convolutional layers are equivariant to translation (useful for classification!)
- Currently state-of-the-art in several tasks

# Outline

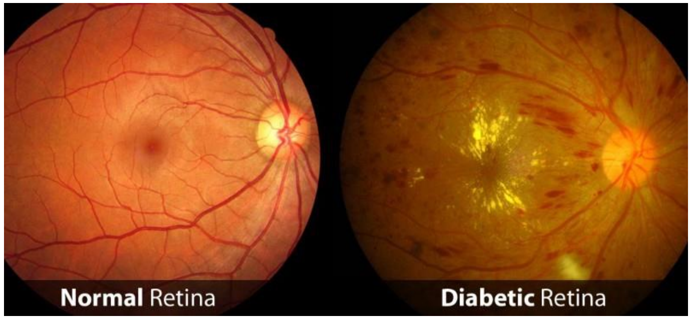
- What is deep learning?
- Artificial neural networks
- Convolutional neural networks
- Biomedical application examples
  - Image classification
  - Image segmentation
  - Image reconstruction
- Application challenges

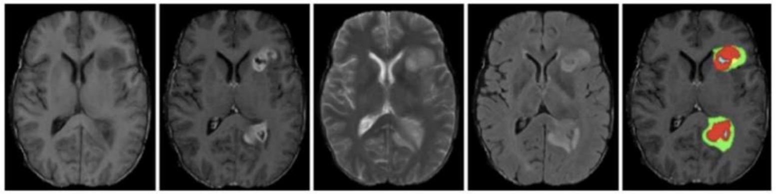
# Image/signal classification

- Objective: given an image/signal, produce a label
- Computer Aided Decision (CAD) systems:
  - Help human operator in taking decision
  - Continuous monitoring
  - Screening:
    - Reduce number of unnecessary exams
    - Reduce number of missed detections

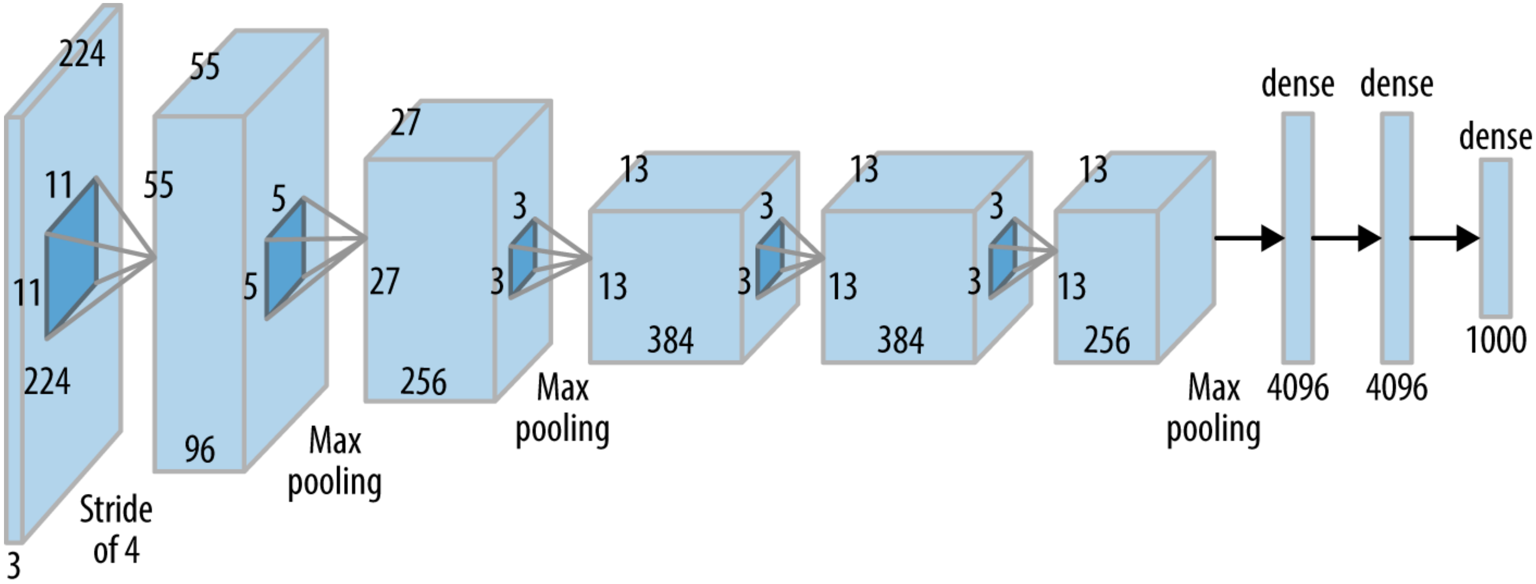
# Successful biomedical application

- Diabetic retinopathy detection
- Tumor detection from MRI, CT, X-rays, etc
- Skin lesion classification from clinical and dermoscopic images
- Heart sound classification: normal vs. abnormal, murmur classification
- Parkinson’s disease detection from voice recording





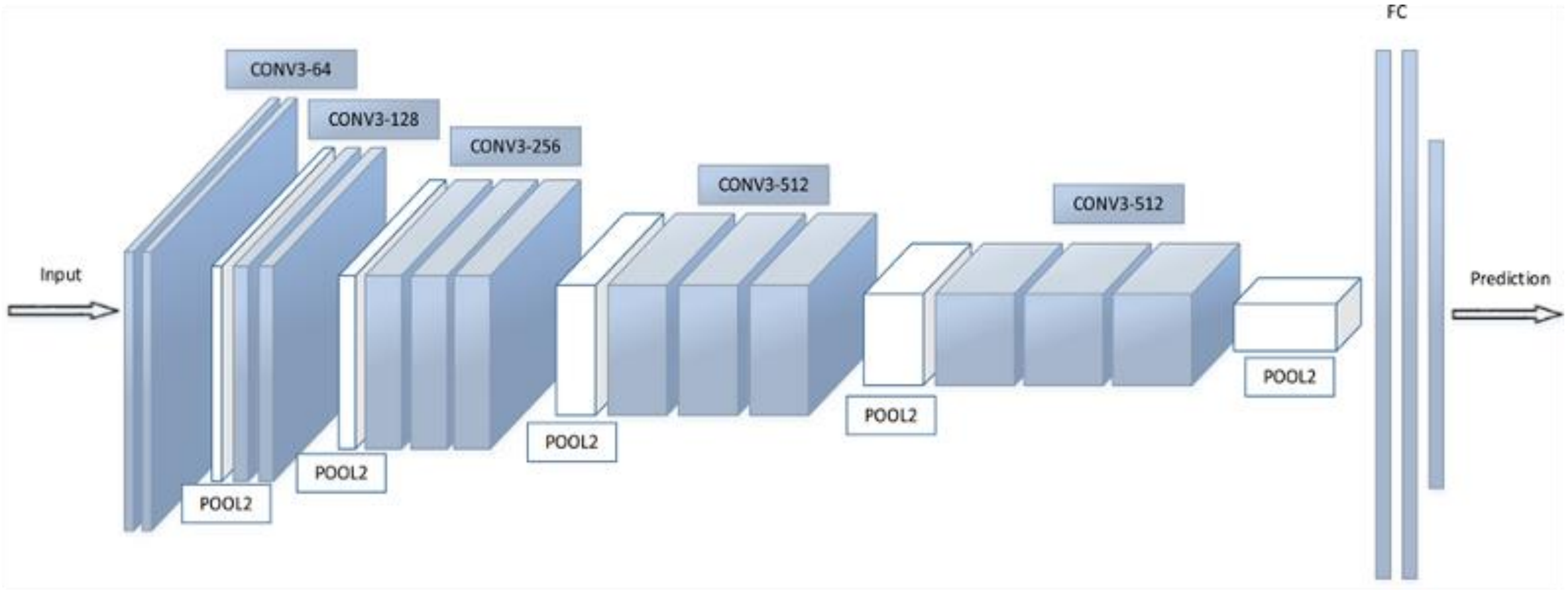
# AlexNet



- Winner of ILSVRC 2012
- Marked the beginning of recent deep learning revolution

A. Krizhevsky, I. Sutskever, and G. Hinton. "ImageNet Classification with Deep Convolutional Neural." In *NIPS*, pp. 1-9. 2014.

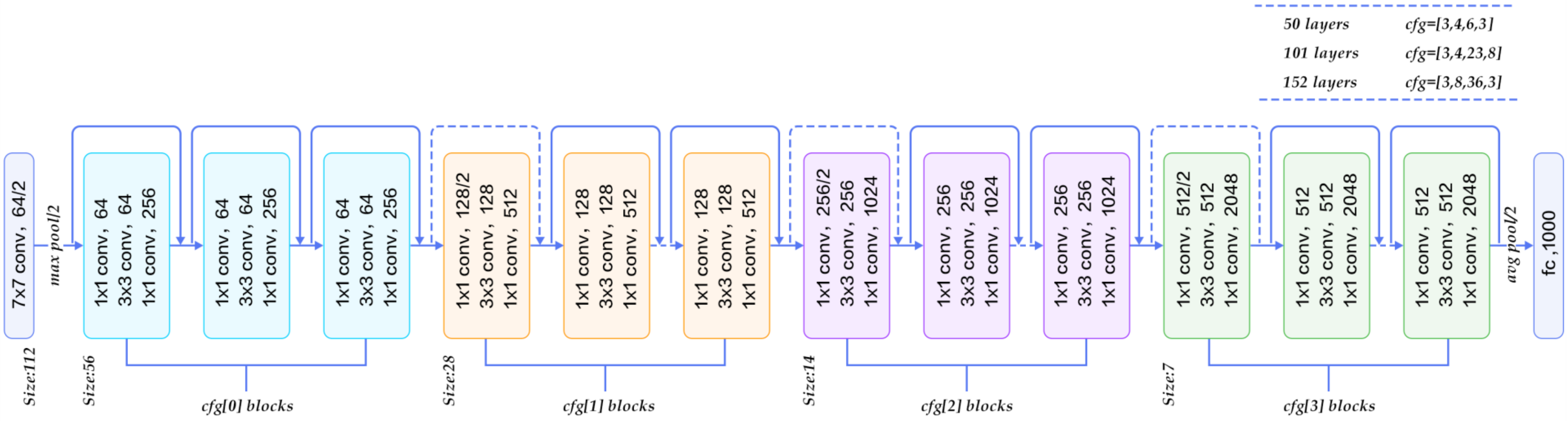
# VGG-16



- Very small filters (3x3)
- Deeper than AlexNet:16 layers

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Representations, 2015.

# ResNet



From: https://www.codeproject.com/Articles/1248963/Deep-Learning-using-Python-plus-Keras-Chapter-Re

- Increase the number of layers by introducing a residual connection
- Blocks are actually learning residual functions: easier!

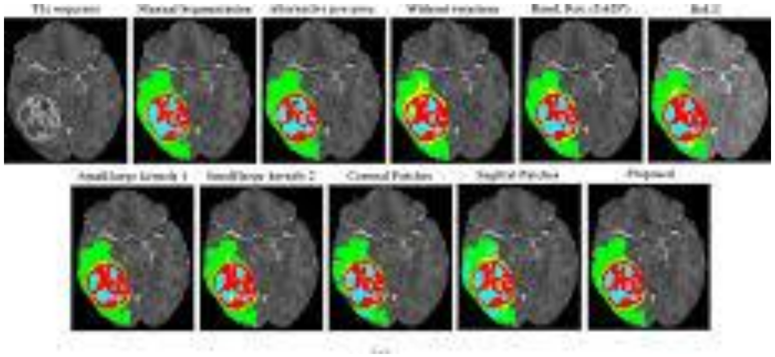
K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

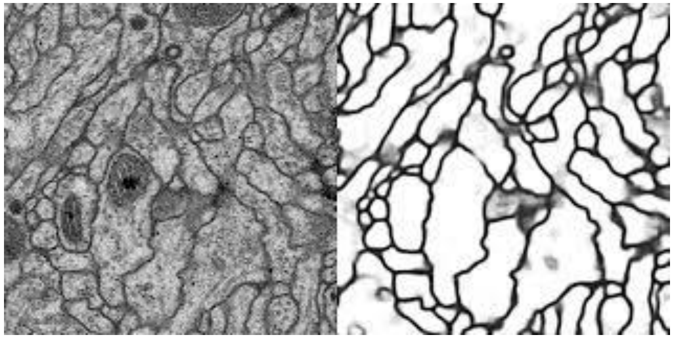
# Image/signal semantic segmentation

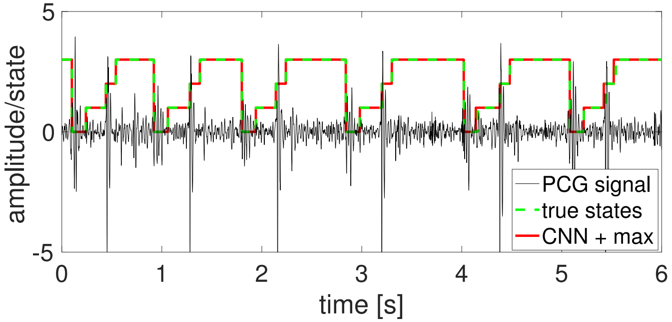
- Objective: partition an image/signal in multiple segments, sets of pixels/samples
- Similar to classification, but a label is assigned to each pixel of the image
- Used for understanding and interpretation:
  - Highlight region of interest
  - Compute volume
  - Surgery planning

# Successful biomedical applications

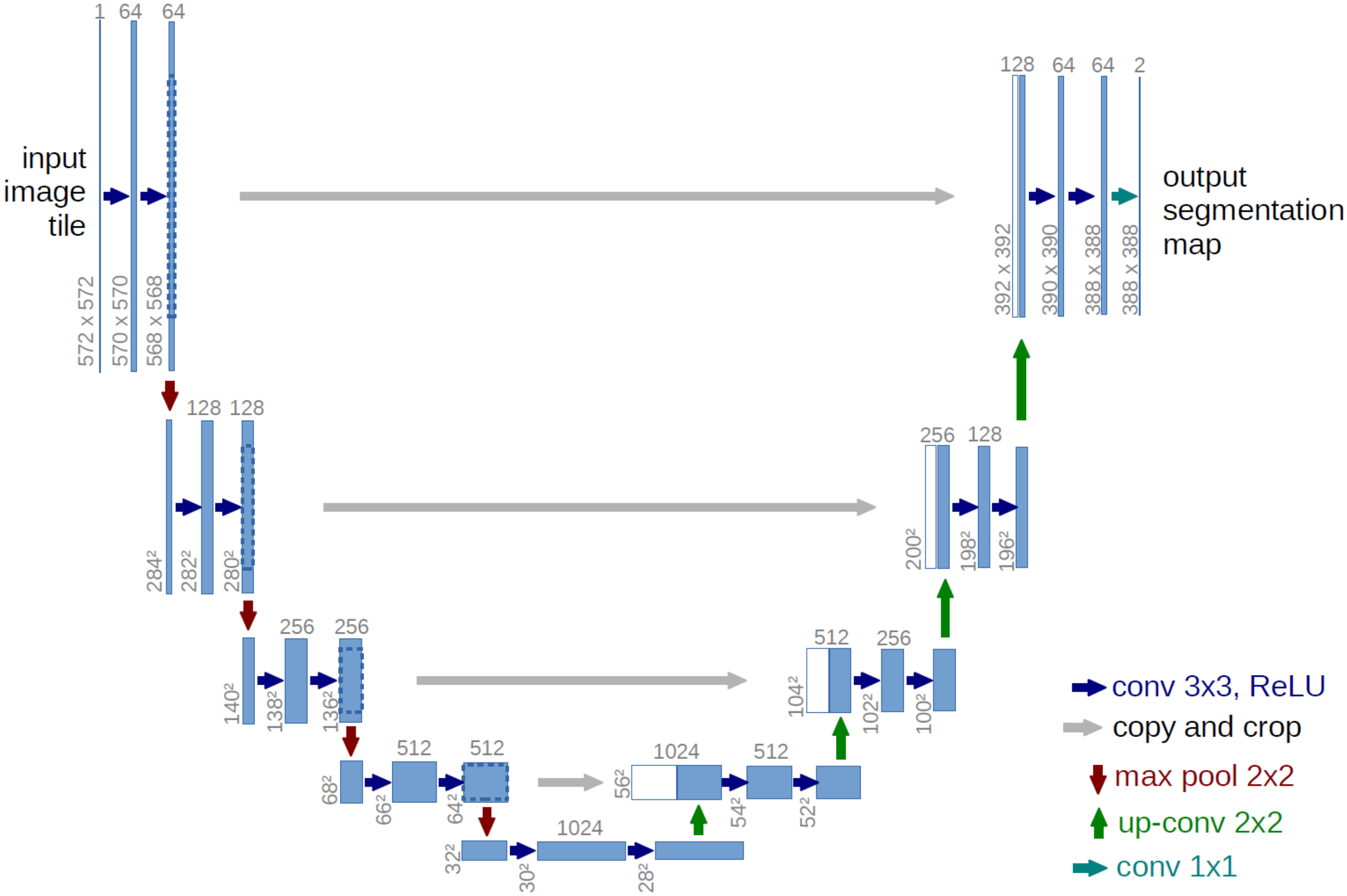
- MRI tumor segmentation
- X-Ray image segmentation
- Electron and light microscopy segmentation
- Heart sound segmentation
- Etc.







# U-Net



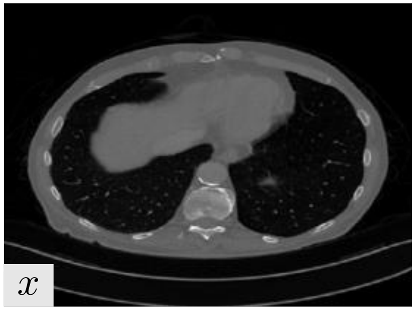
- Encoder-decoder structure

O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation." In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234-241. Springer, Cham, 2015.

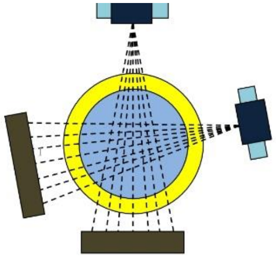
# Image reconstruction/acquisition

- Recover a full image of interest from partial measurements/observations
- Increase de quality/resolution of acquired image
- Reduce the impact of reconstruction artifacts
- Reduce acquisition time/dose

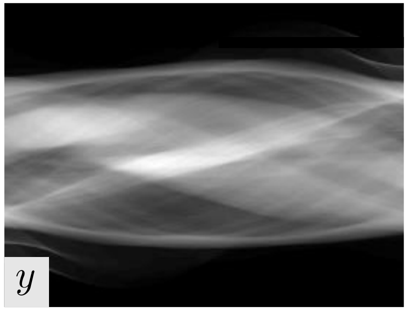
# Example: Computer Tomography



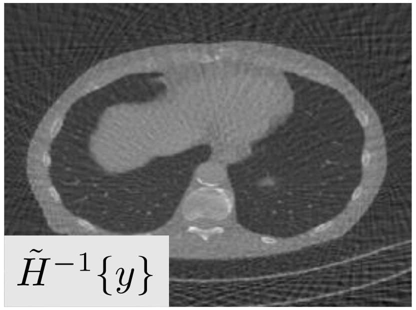
Fully sampled image



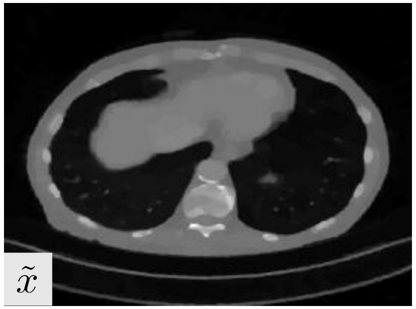
CT measurements



Sinogram



Direct reconstruction
from downsampled sinogram



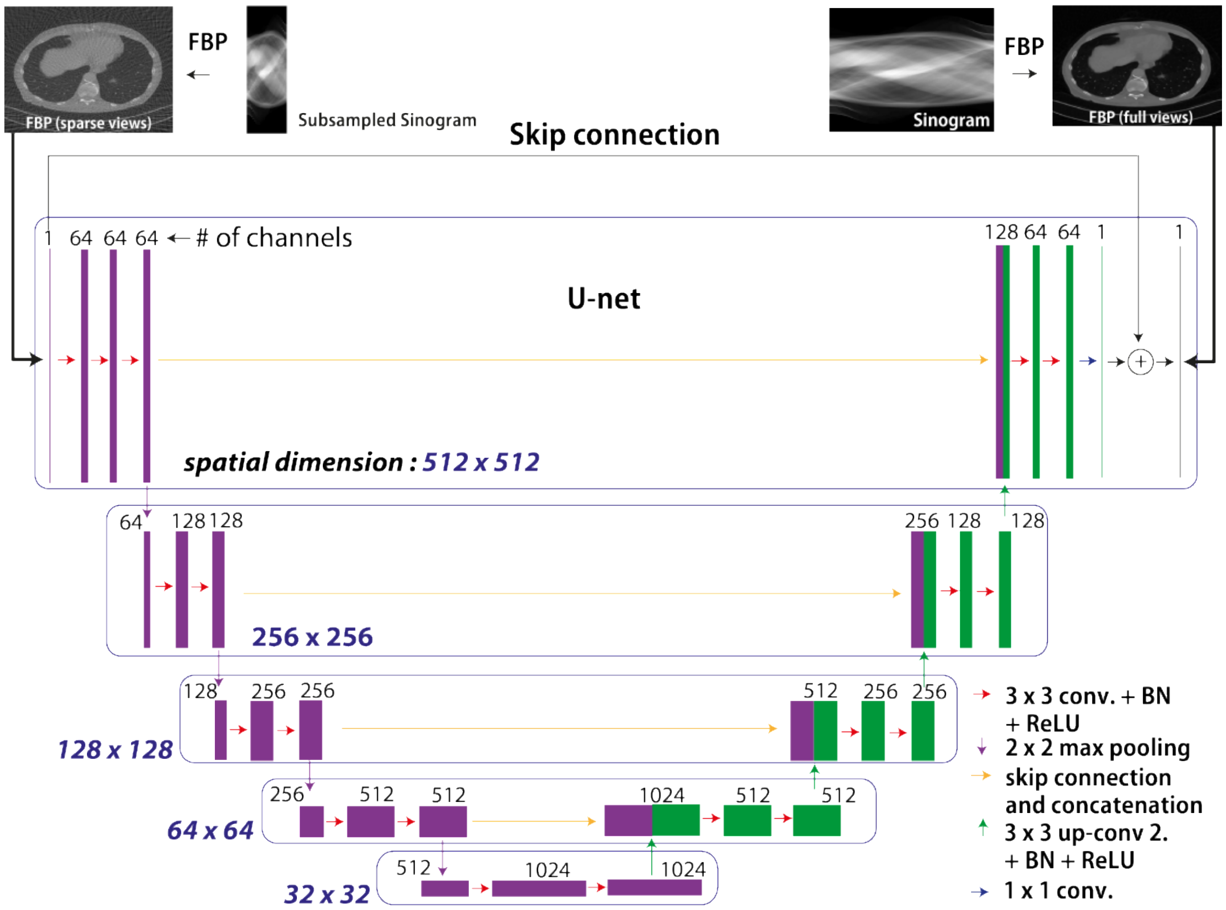
Reconstruction with CNN

M. McCann, K. Jin, and M. Unser. "A review of convolutional neural networks for inverse problems in imaging." *arXiv preprint arXiv:1710.04011* (2017).

# Remarks

- It is a regression problem, not a classification problem
  - The CNN output is not a class label, but a collection of real numbers (the recovered image)
- Loss function: usually different from classification problems (e.g., L2-norm, in space or frequency domain)
- Training set: pairs of ground truth images (fully sampled) and downsampled measurements

# Modified U-Net



K. Jin, M. McCann, E. Froustey, and M. Unser. "Deep convolutional neural network for inverse problems in imaging." *IEEE Transactions on Image Processing* 26, no. 9 (2017): 4509-4522.

# Application challenges

- Great results! But…
  - Difficult to select best architecture for a problem
  - Require new training for each task/configuration
  - (Most commonly) require a large training dataset to generalize well
    - Data augmentation, weight regularization, transfer learning, etc.
  - Still not fully understood why it works so well
    - Robustness against adversarial examples
    - Approval from government agencies (ex. FDA)?

# To know more…

- Theory
  - I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Vol. 1. Cambridge: MIT press, 2016. (https://www.deeplearningbook.org/)
- Survey papers
  - "Deep Learning for Visual Understanding," in IEEE Signal Processing Magazine, vol. 34, no. 6, Nov. 2017.
  - A. Lucas, M. Iliadis, R. Molina and A. K. Katsaggelos, "Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods," in IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 20-36, Jan. 2018.
- Tutorial
  - Oxford Visual Geometry Group: VGG Convolutional Neural Networks Practical (http://www.robots.ox.ac.uk/~vgg/practicals/cnn/)

# To start coding

- Coding frameworks for deep learning
  - TensorFlow (https://www.tensorflow.org/),
    PyTorch (https://pytorch.org/),
    Theano (http://deeplearning.net/software/theano/),
    MatConNet (http://www.vlfeat.org/matconvnet/),
    etc.
- High-level wrappers
  - Keras (https://keras.io/),
    TensorLayer (https://tensorlayer.readthedocs.io/en/stable/),
    Lasagne (https://lasagne.readthedocs.io/en/latest/),
    etc.
- GPU strongly recommended!