Computer Vision – TP14
Introduction to Deep Learning

Miguel Tavares Coimbra

U. PORTO FC

Outline

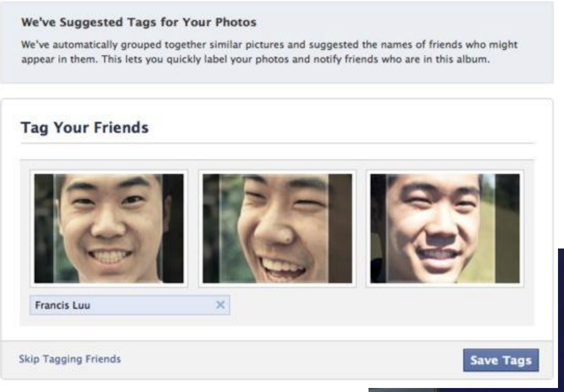
- What is deep learning?
- Convolutional neural networks
- Deep neural network architectures

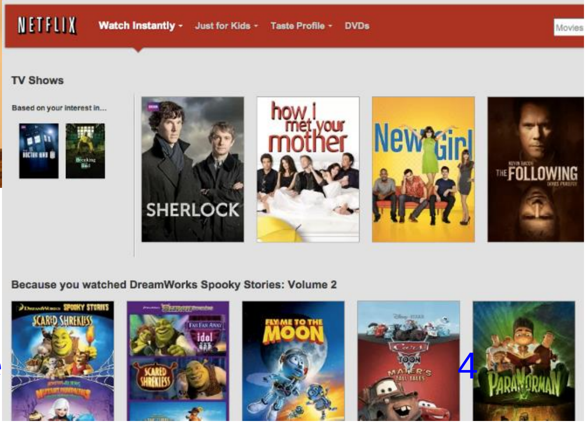
Topic: What is deep learning?

- What is deep learning?
- Convolutional neural networks
- Deep neural network architectures

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…



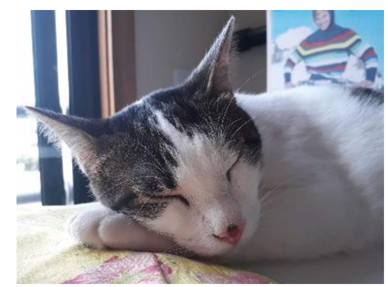


What is deep learning?

- 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?

Remember!

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.

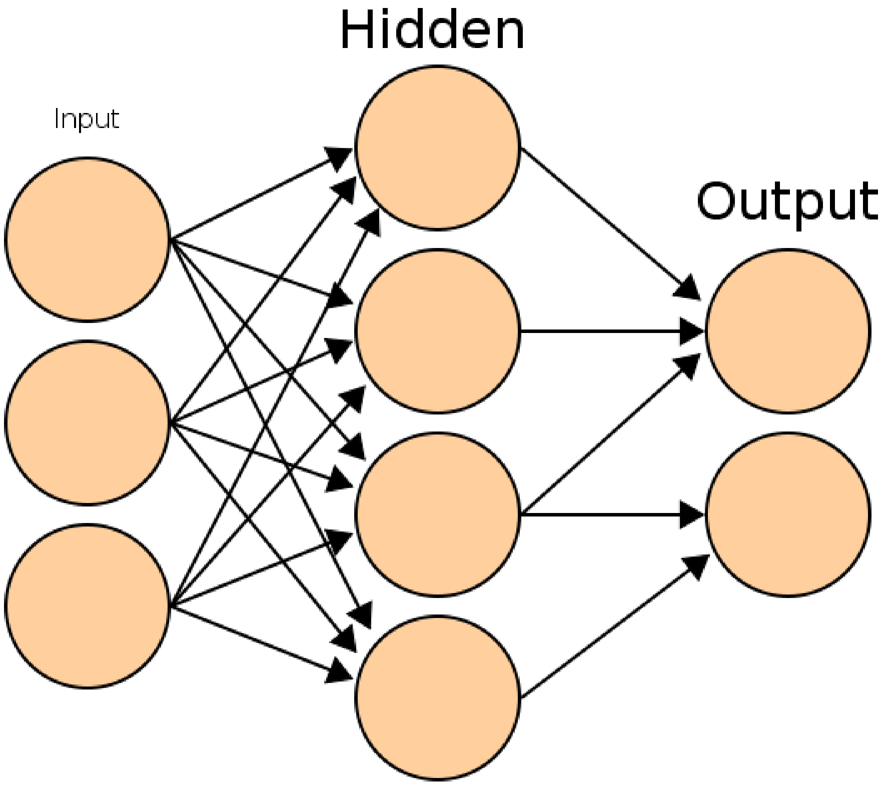
↓

Classifier

Classify or describe the observation, relying on the extracted features

Remember: Neural Networks

- Basic principles:
  – One neuron can perform a simple decision
  – Many **connected** neurons can make more **complex decisions**



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

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

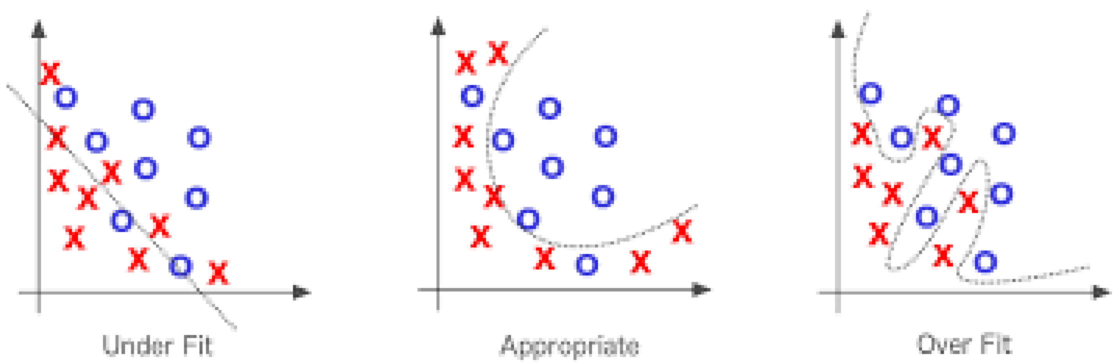
Optimization: find $[w_{jk}^{(1)}, w_{\ell j}^{(2)}, \ldots, w_{nm}^{(L)}]$ such that

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

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

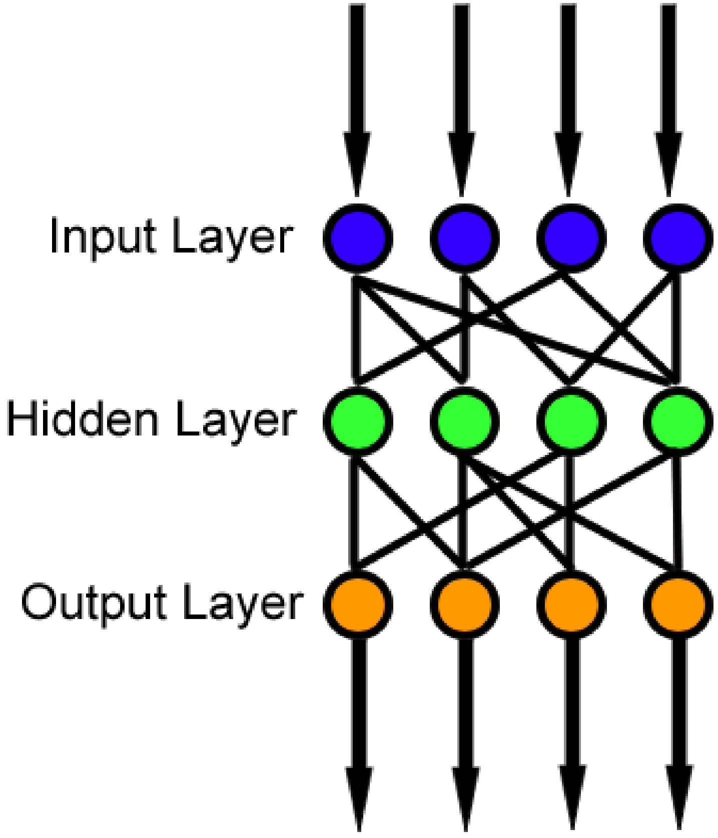
Warnings!

- 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)

Topic: Convolutional neural networks

- What is deep learning?
- Convolutional neural networks
- Deep neural network architectures

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

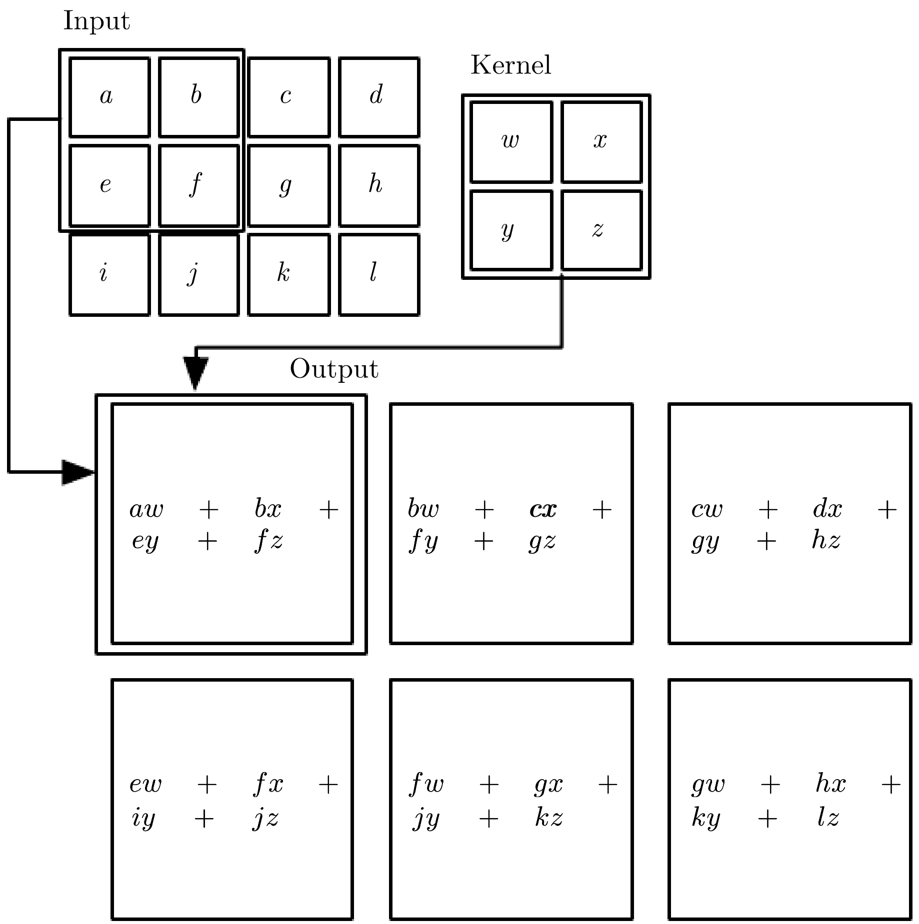
Sensor

Acquire the data, observations to be classified or described

Convolutional neural network

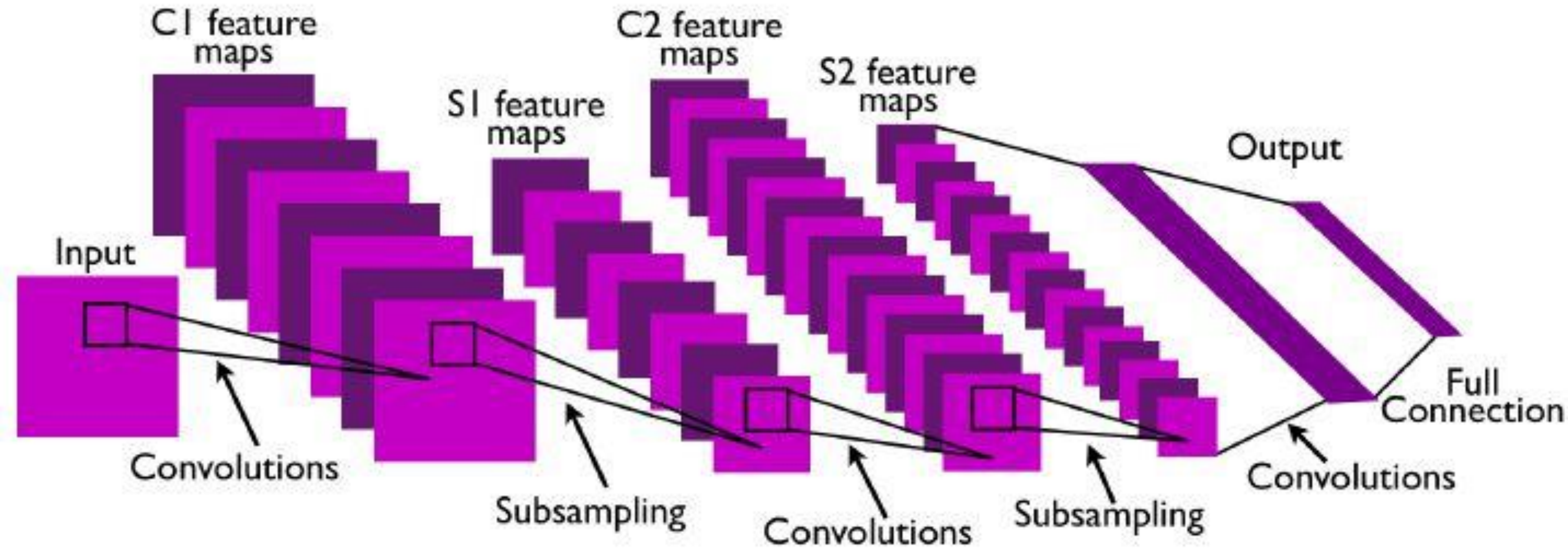
Classify or describe the observation, automatically extracting (learnt) features

Convolution



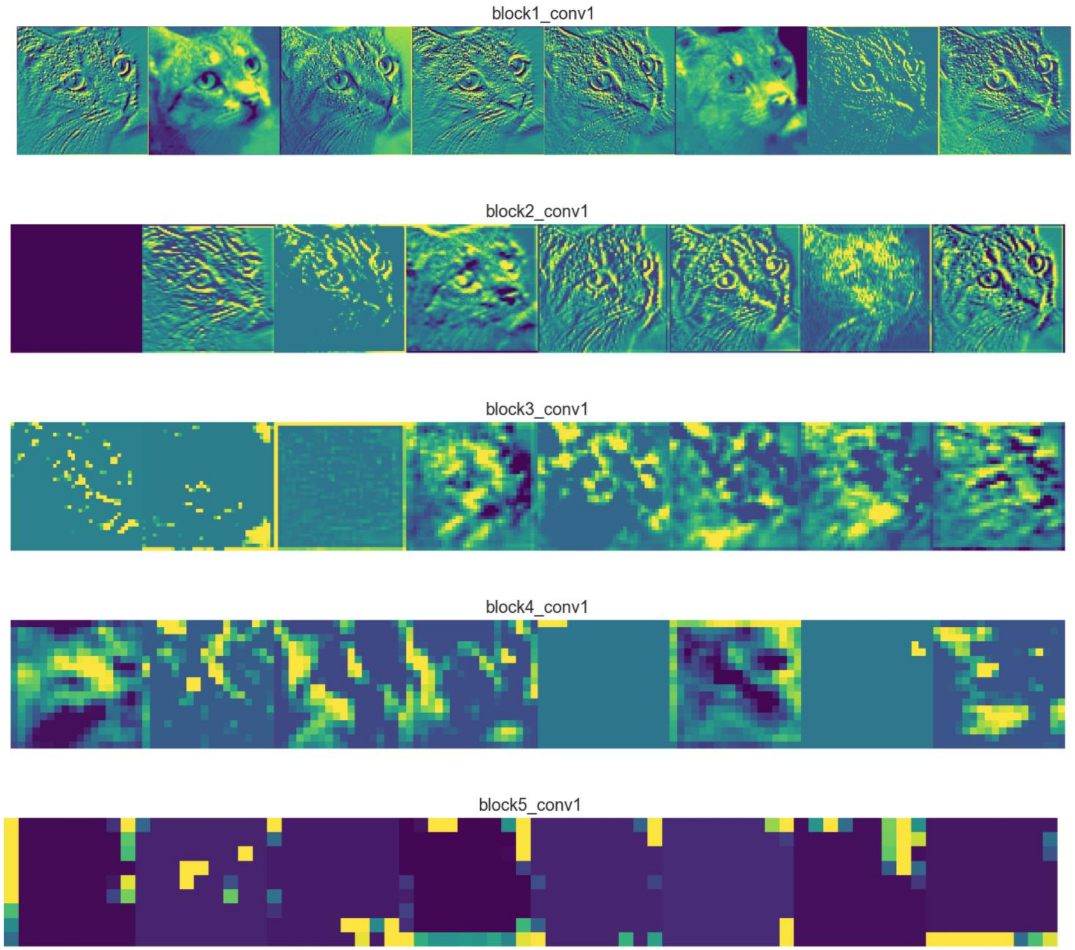
I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

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

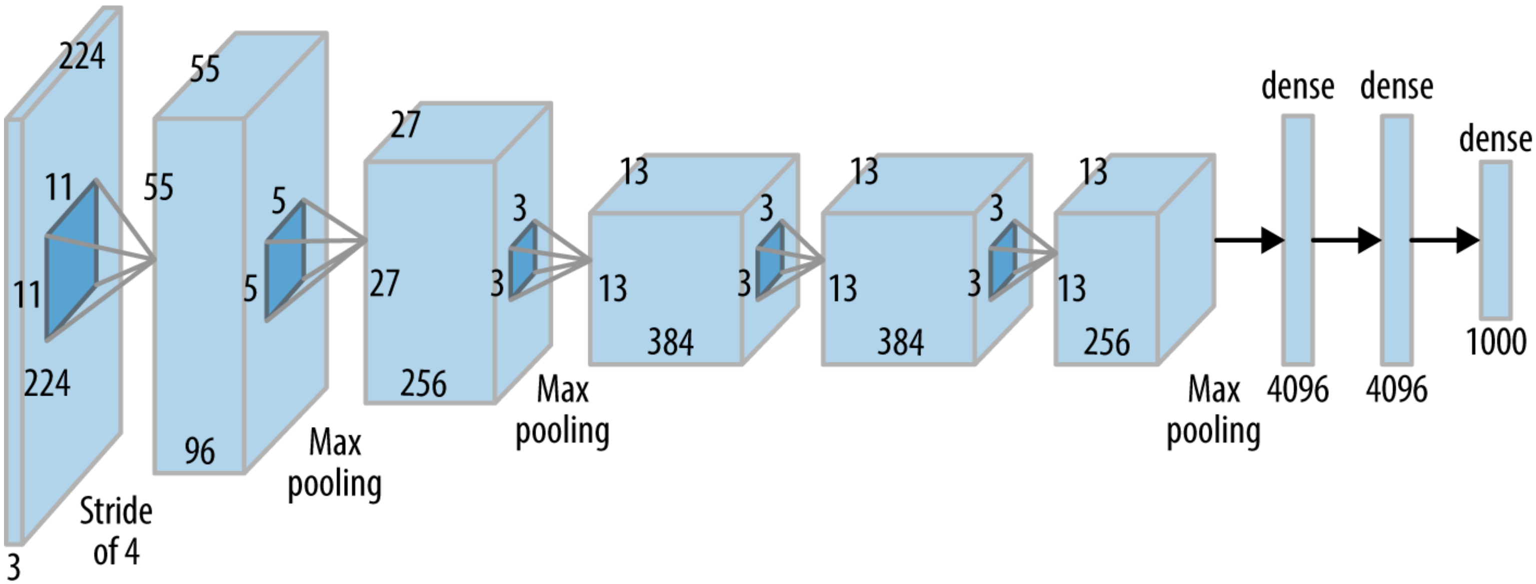
CNN - Properties

- 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

Topic: Deep neural network architectures

- What is deep learning?
- Convolutional neural networks
- Deep neural network architectures

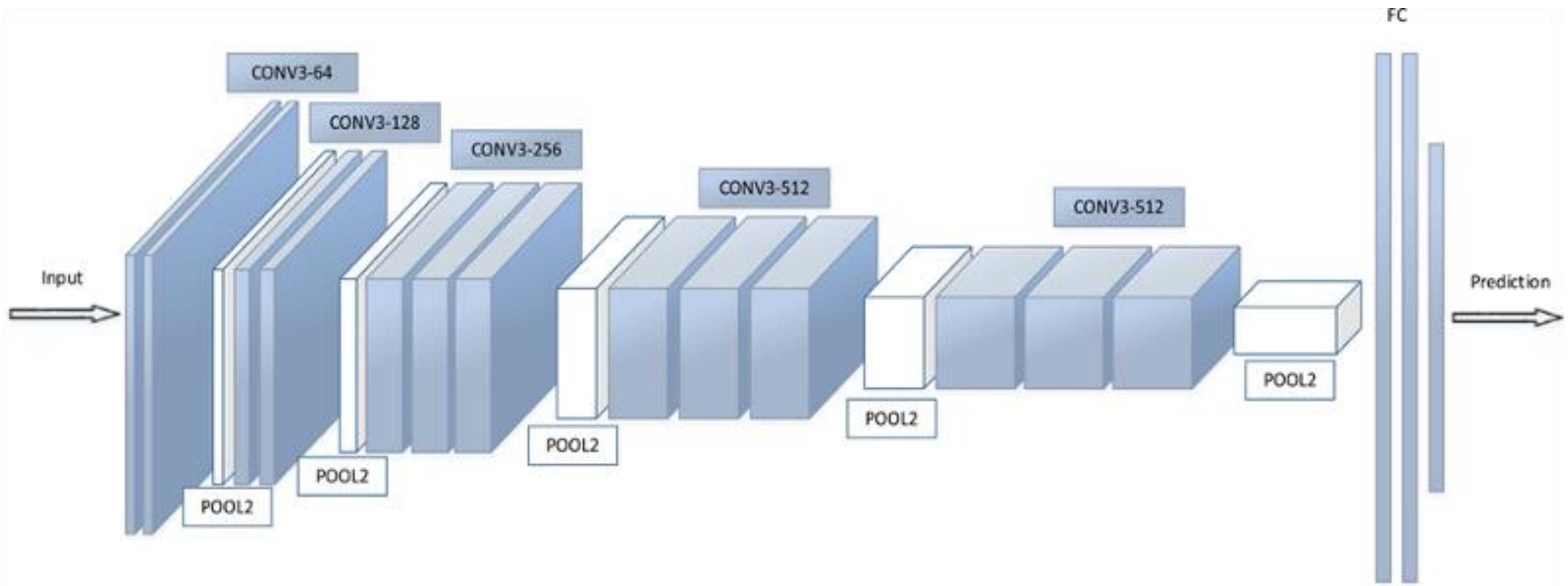
AlexNet



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

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

VGG-16

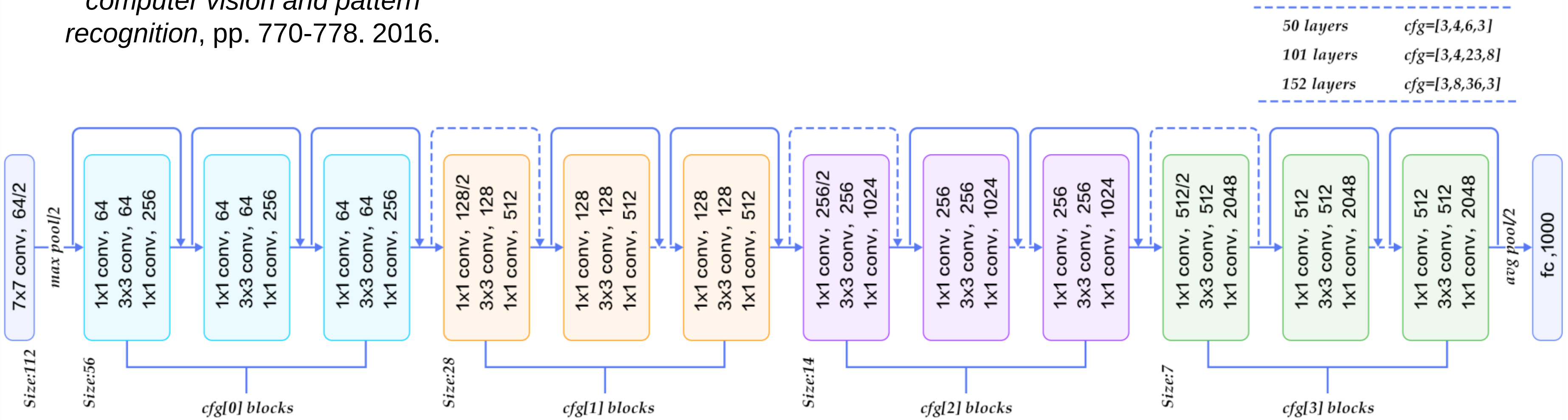


- Very small filters (3x3)
- Deeper than AlexNet:16 layers

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

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.

ResNet

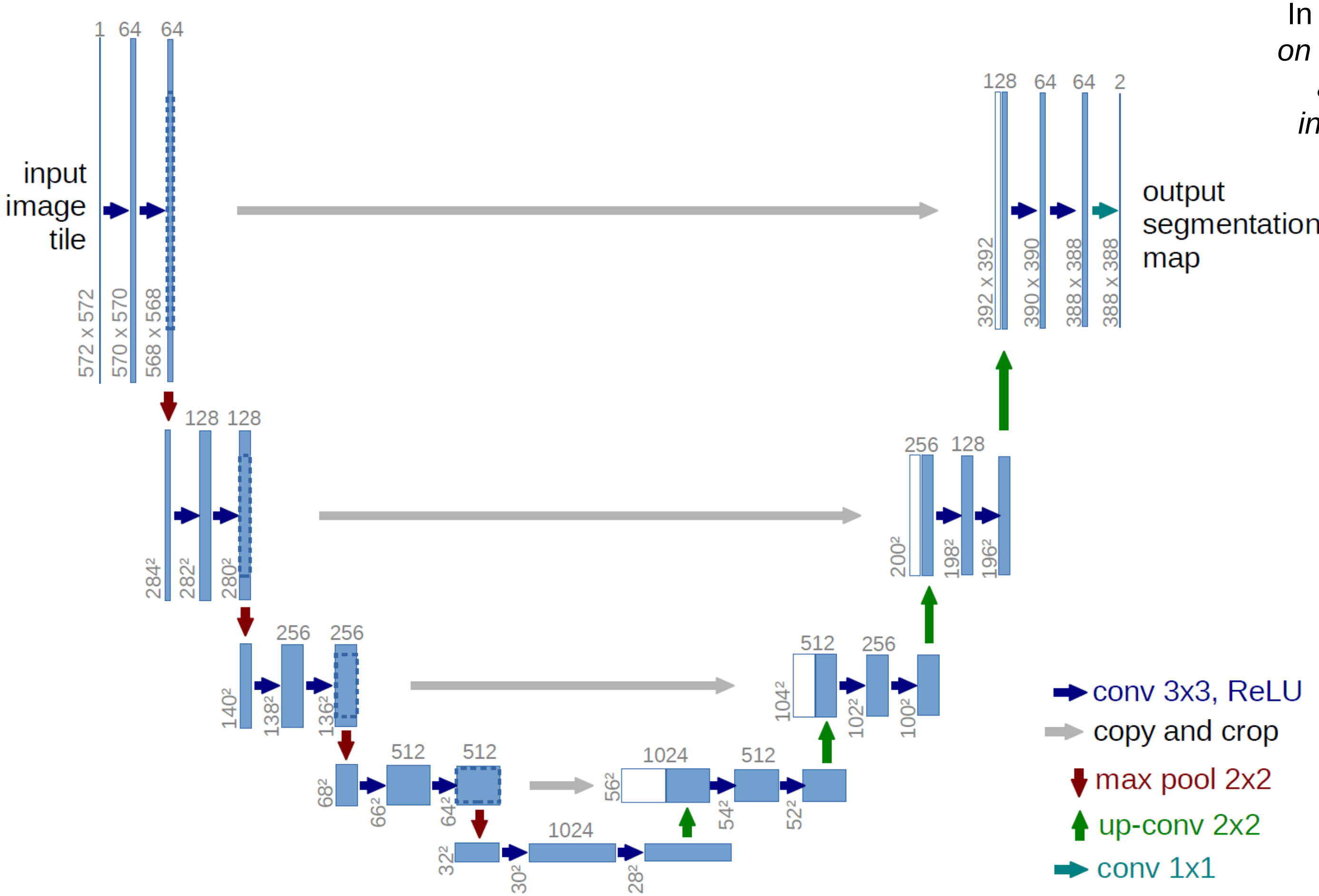


From: https://www.codeproject.com/Articles/1248963/Deep-Learning-using-Python-plus-Keras-Chapter-Re

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

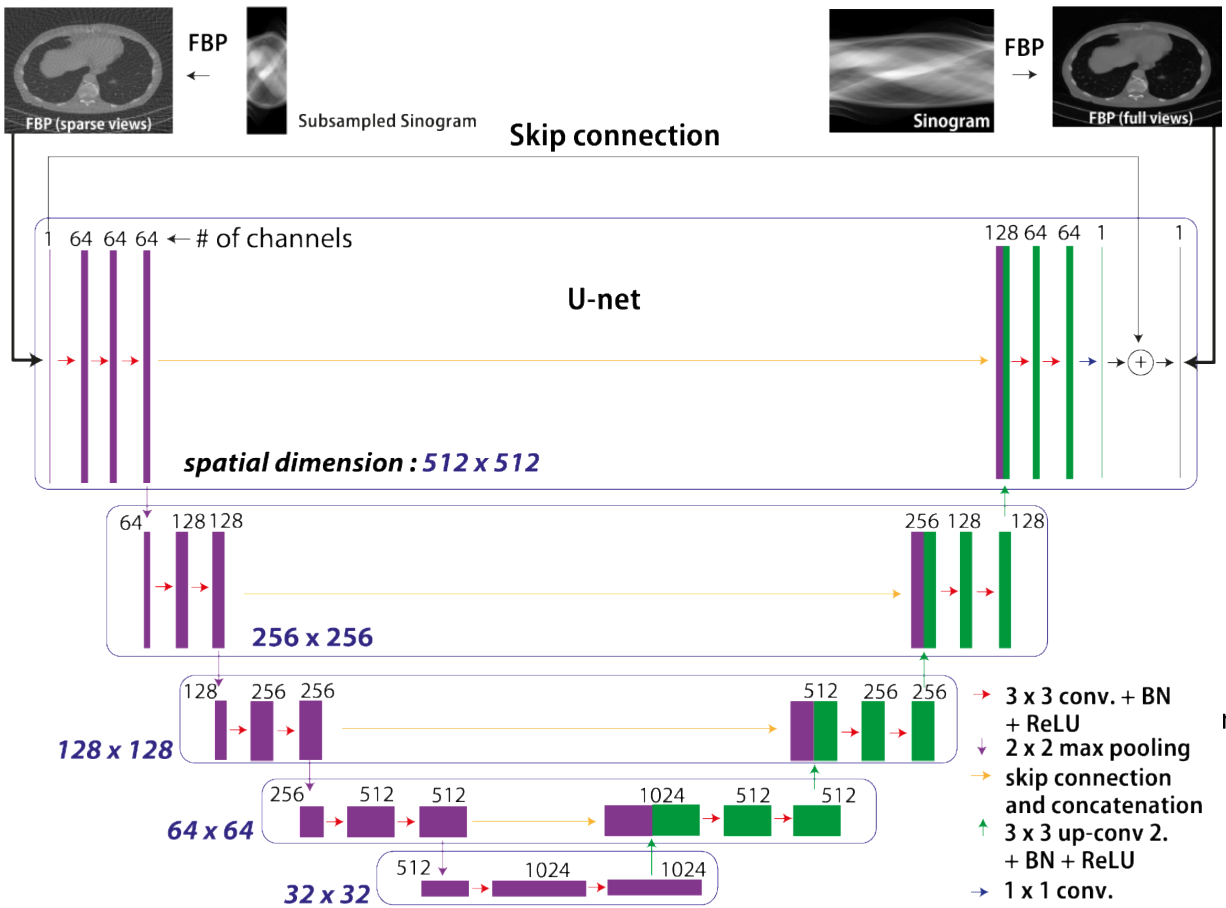
U-Net

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.



- Encoder-decoder structure

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)?

Resources

- 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/)

Coding Resources

- 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!