Multiple face recognition in images

Presentation on Convolutional Neural Networks, 2018

Simone Caldarella
s.caldarella@studenti.unibs.it

IEEE Student Branch Brescia
University of Brescia

5/12/2018

Outline

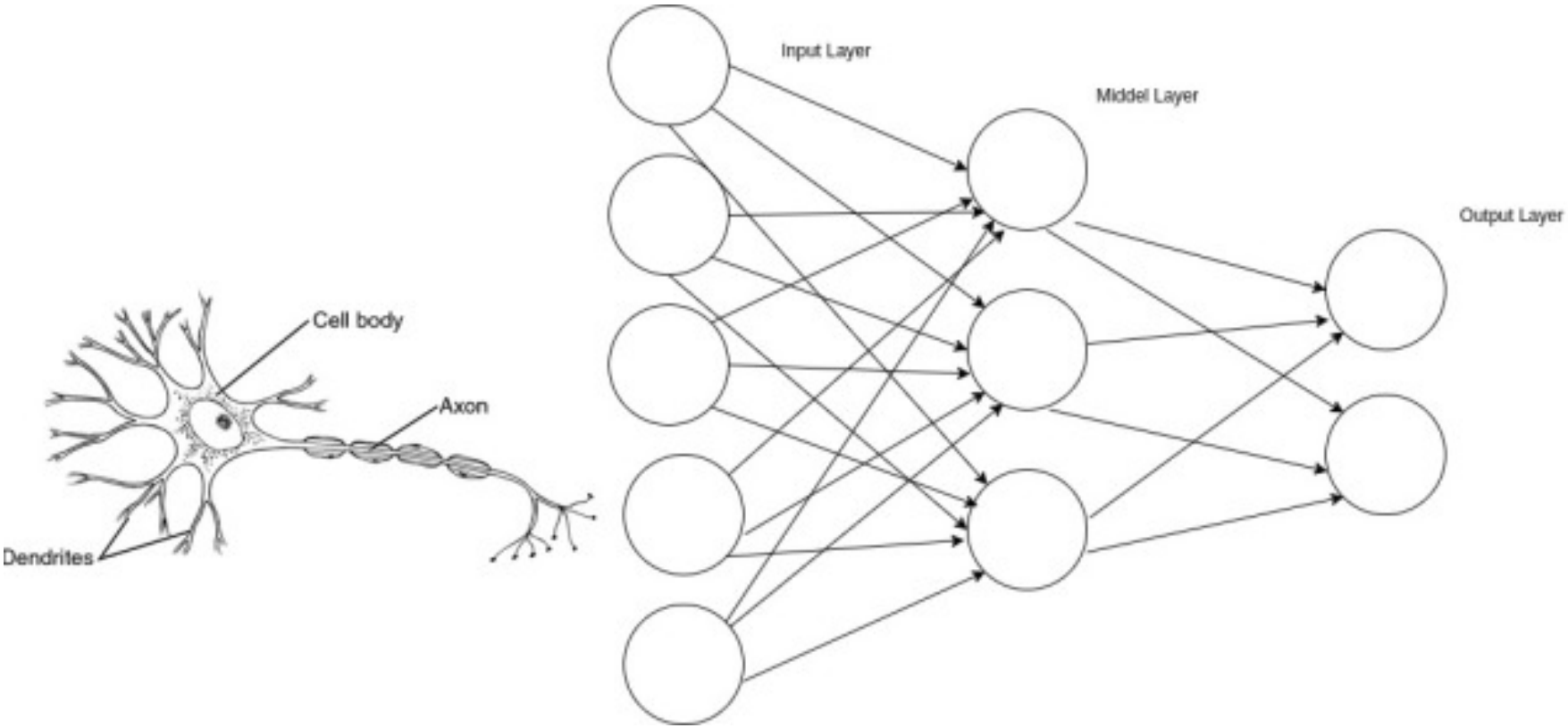
Motivation

- Machine learning is a field in continuos spreading.
- Problem: IEEE student branch's members' recognition using neural network.
- Create a ground for recognition problems.
- Goal: implementing a face recognition system for the student branch's members, expecially:
  - detect and recognize different faces in photos;
  - use a computer vision's framework (OpenCv) and a machine learning framework (Tensorflow).

Outline

What does machine learning means?

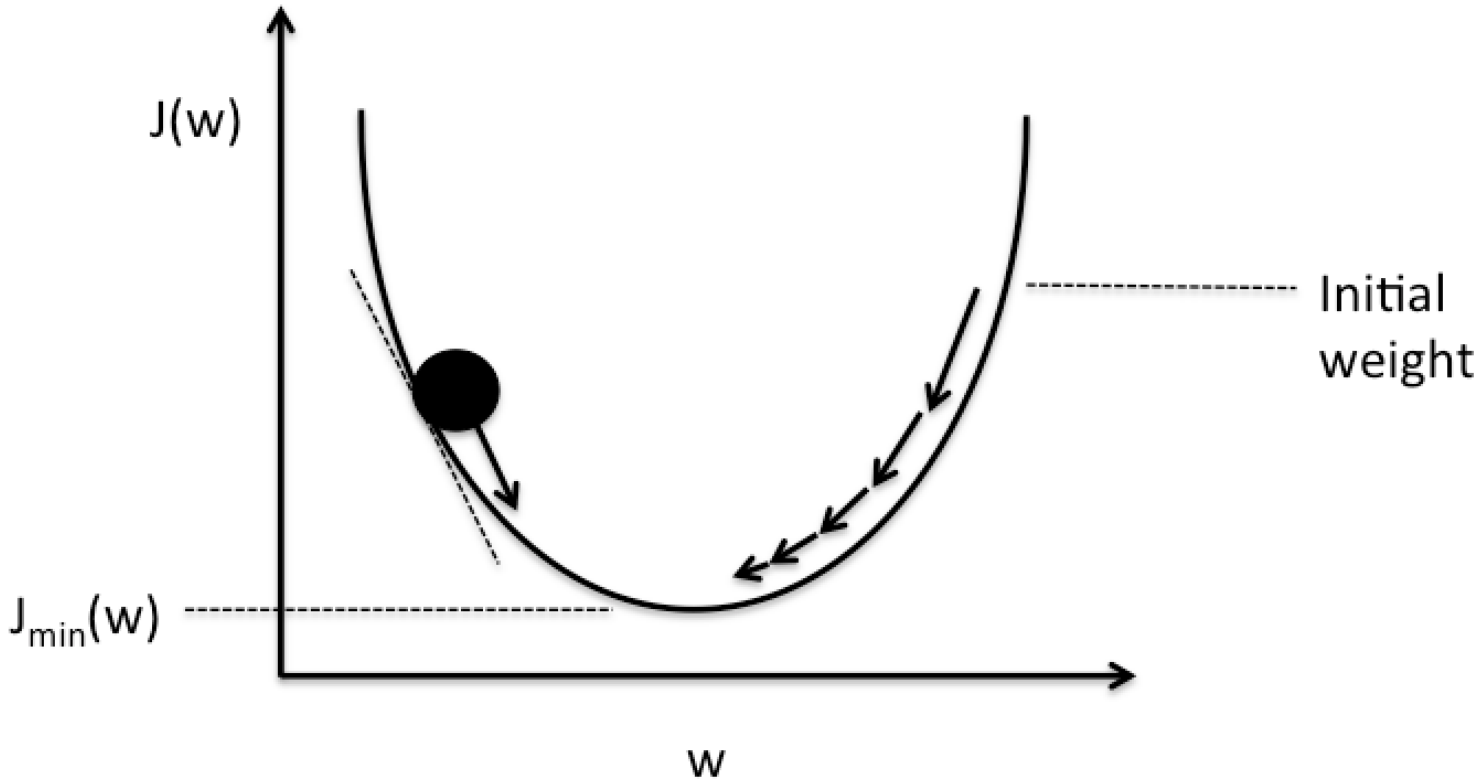
- Is this a **neural network** or a **graph**?



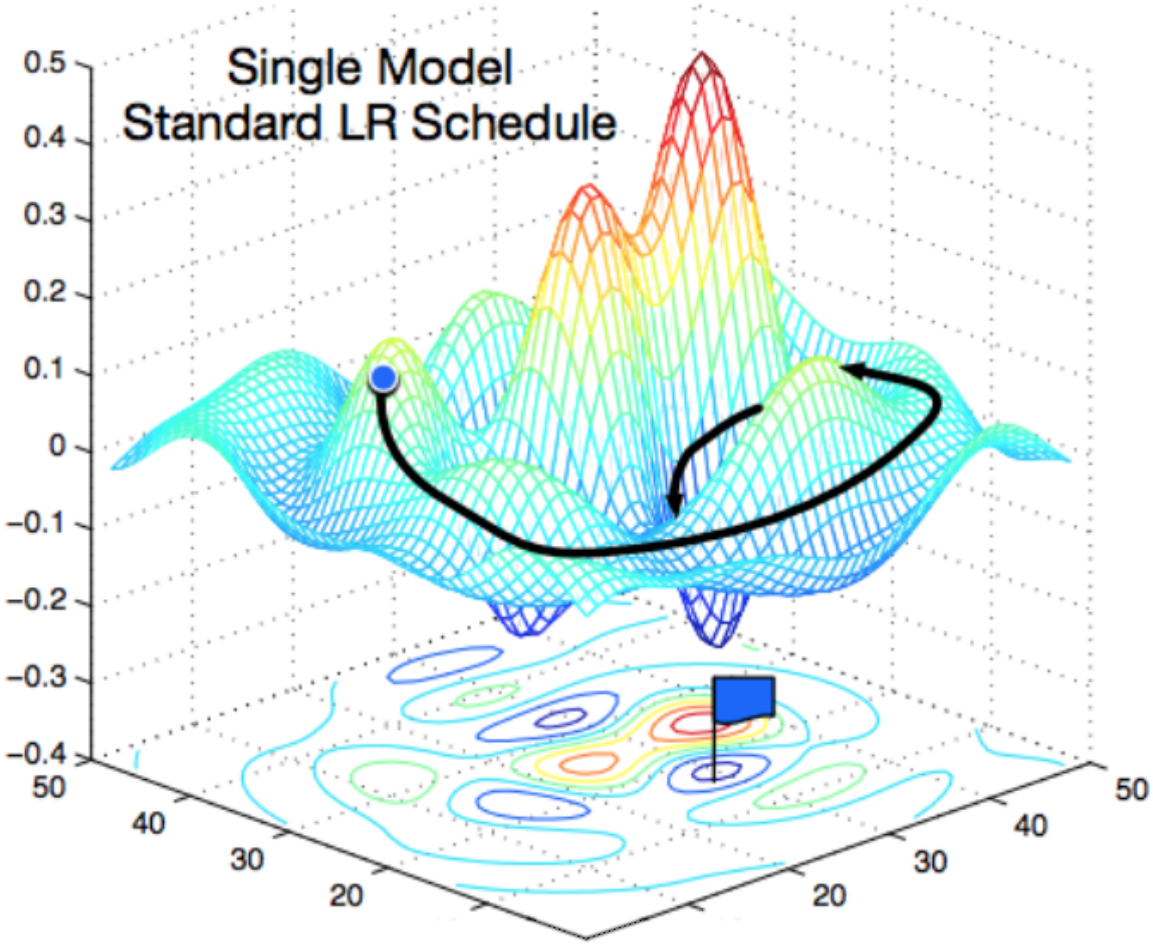
What does machine learning means?

- The concept of **training**:
  - input and output;
  - inference and loss function;
  - gradient descent and weights update.
- The importance of a large and well structured dataset:
  - common problems;
  - cognitive bias.

Training and gradient descent



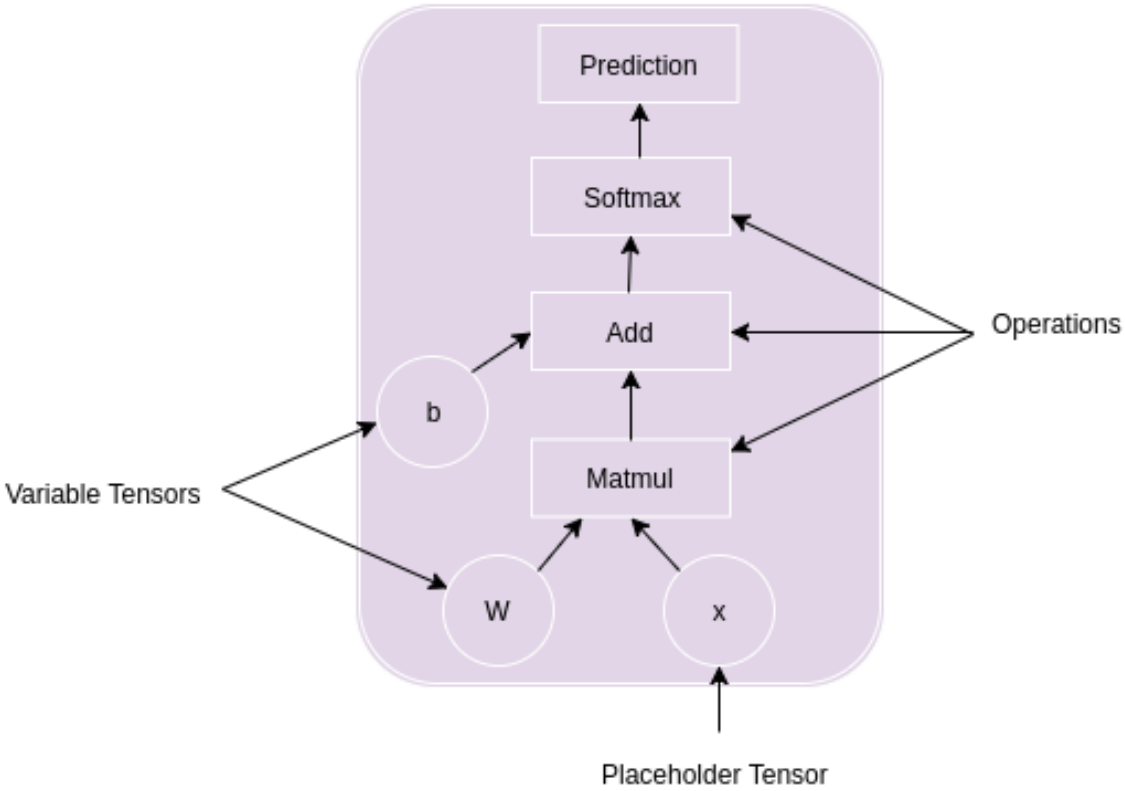
A more complex gradient descent loss



Outline

Tensorflow and OpenCv

- Tensorflow and computational graph concept



Computational Graph

Tensorflow

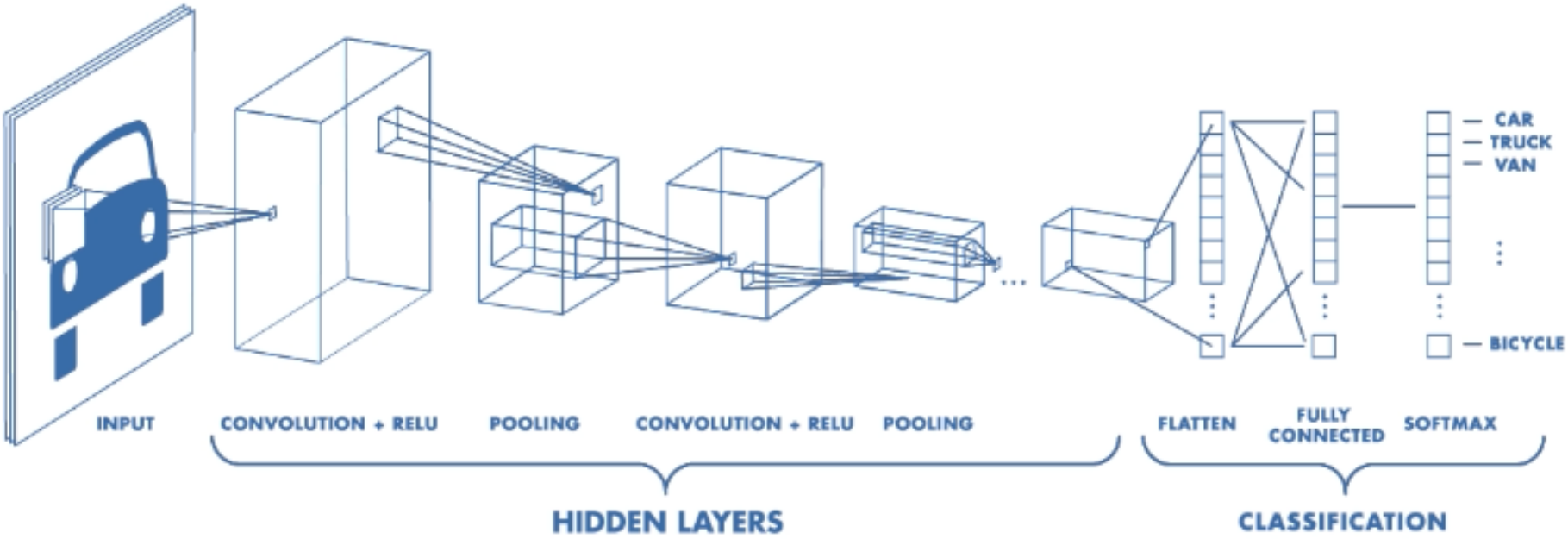
- Why Tensorflow?
  - Tensors and use of computational graph;
  - different levels of abstraction.
- Low level and high level API:
  - Tensorflow functions;
  - `Keras` and `tflearn`.

OpenCV

- Why OpenCV?
  - Very fast;
  - lots of library and algorithm at the state-of-art.
- OpenCV detection algorithm:
  - `HaarCascadeClassifier`;
  - `Dlib` library for face features detection.

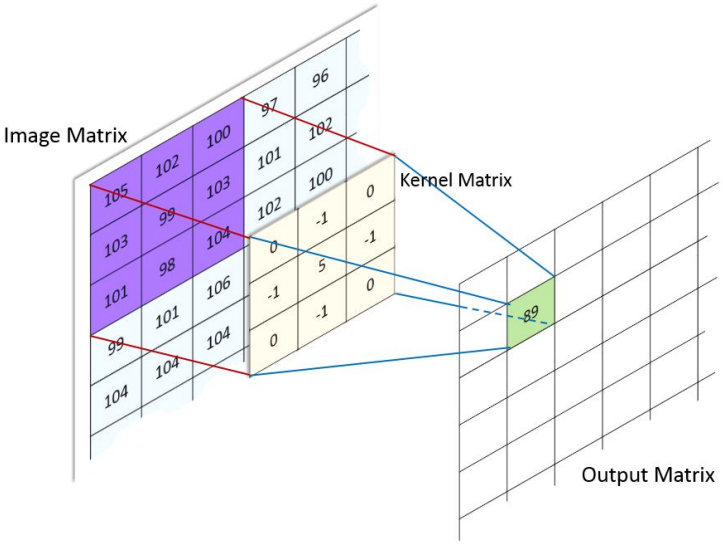
Outline

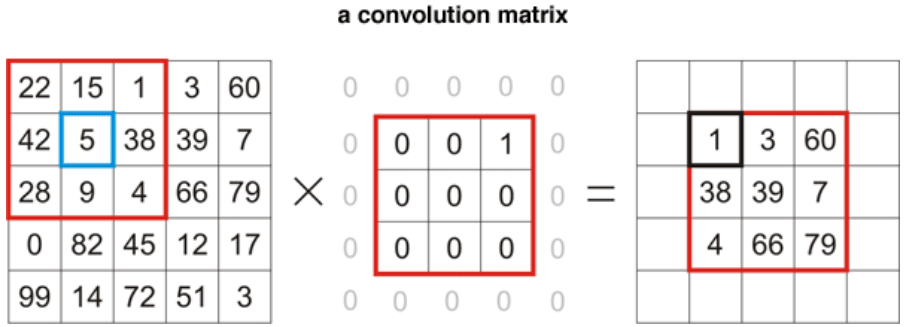
Convolutional Neural Network (CNN)



Convolutional layers

- Convolutional matrix (Kernel)

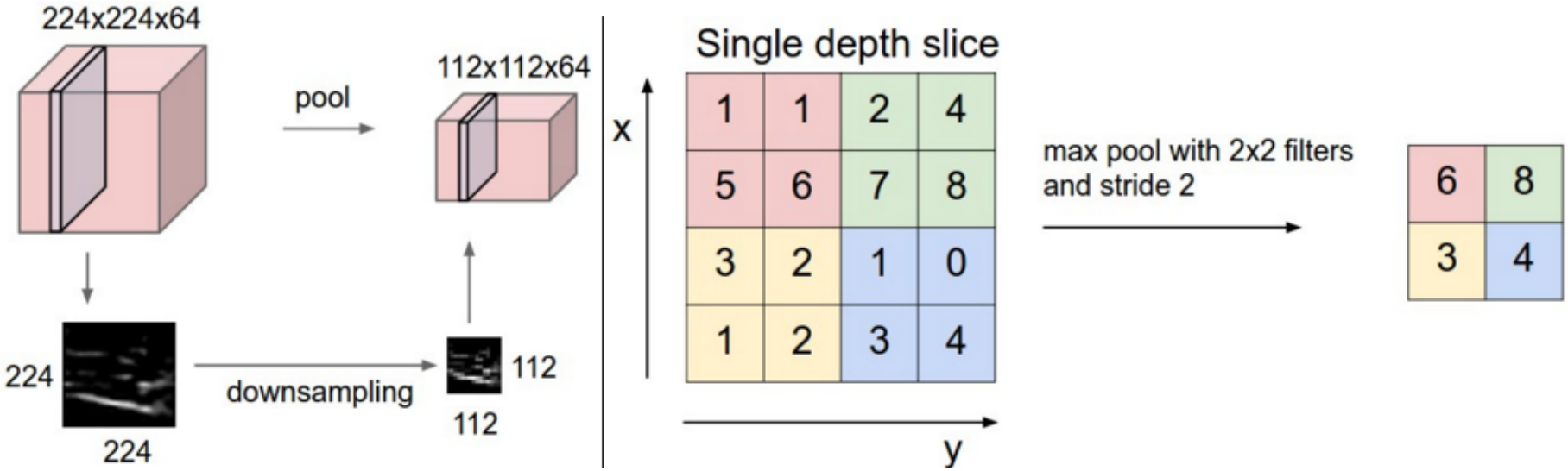




- 3x3, 5x5, or 7x7, why only odd numbers?
- Edge detection:
  - Similarity with human vision;
  - From simple to complex forms.

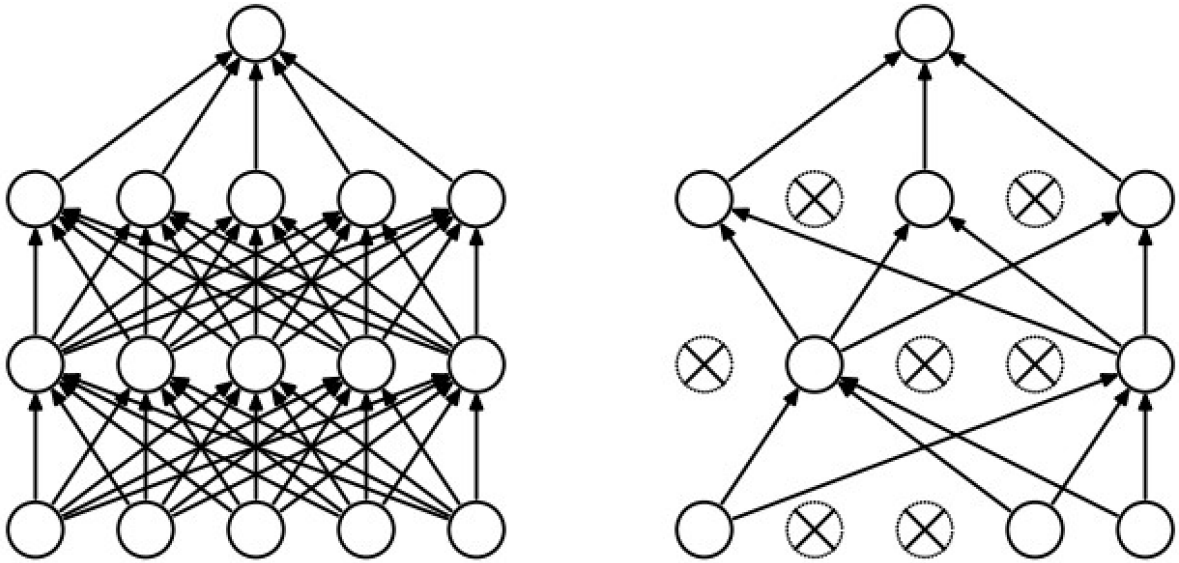
Pooling layers

- Reducing number of information: best way to avoiding **overfitting** and decreasing computation complexity



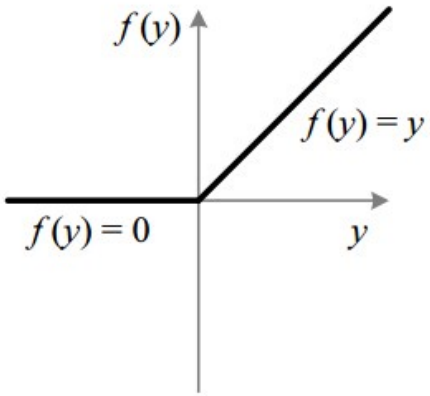
Fully connected layers and dropout

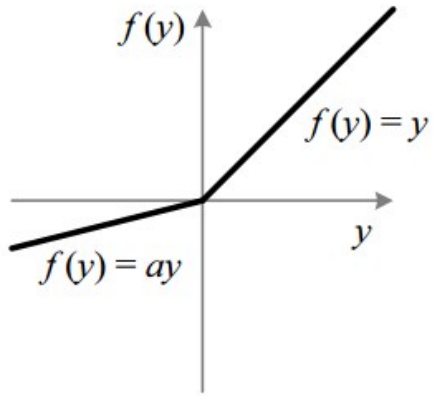
- Fully connected layers are the last layer of the CNN.
- Once the high-level features are recognized, they deal with classifications.
- Dropout regolarization.



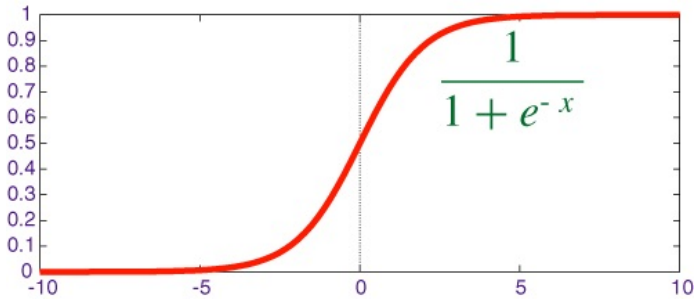
Activation functions

- ReLU:





- Softmax (sigmoid and generical softmax):



$$p_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)}$$

Outline

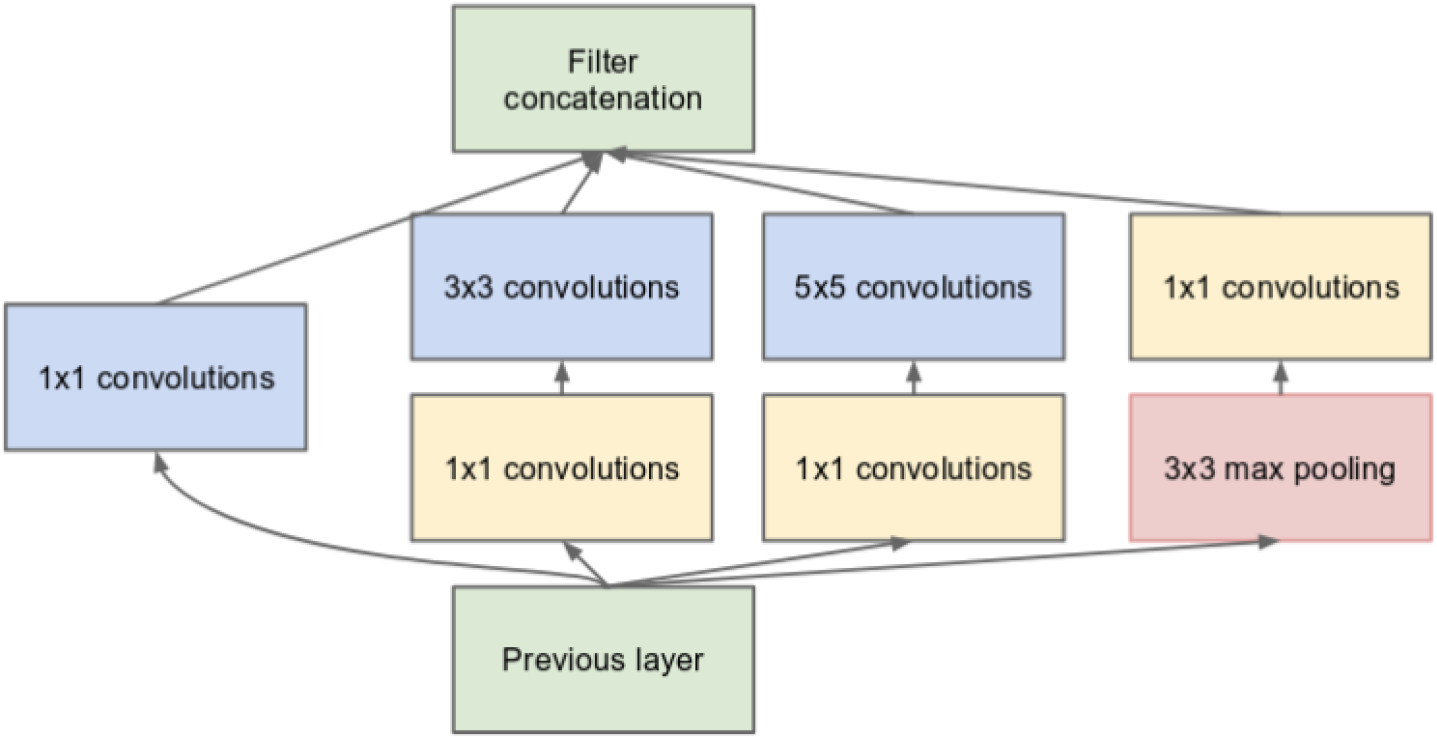
Dataset

- The perfect dataset should be:
  - made with thousands of images;
  - different images with different colors to help the network classify them better.

Outline

Inception network (V3 - Example)

- Inception network analizes images with different kernel size (in the same convolutional layer)



Outline

Tensorflow and retraining

- Concept of transfer learning (or retraining): taking a piece of a model that has already been trained on a related task and reusing it in a new model.
- Tensorflow-hub: the key to create your own classifier with good result and without a Tesla k80.

Outline

inserisci immagine divisa in tre con i tre "momenti"

Outline

Obtain images

- It uses OpenCv functions to get hundreds of photos in less than 30 seconds.
- It crops photos by keeping only the faces(using haarcascadeclassifier) and it saves them.
- This is made to avoid the recognition of unwanted features as background color without the need of several images taken in different places.

Outline

Training

- The algorithm takes the cropped images and use them as dataset for the training.
- Only the fully connected layers are trained.
- More than 10x faster than a complete training.

Outline

Inference

- Choose a photo with multiple faces that you want to recognize/classify.
- Select the already trained network you want to use for inference.
- Obtain the original photo with multiple boxes around all the faces and, under each of them, the name of the most probable person.

Outline

Conclusion

- Creating your own machine learning application using another pre-trained model can help you build something useful without the need of high performance cluster or cloud computing.
- The project, albeit simple, shows the potential of using Tensorflow and machine learning approaches in applications.
- Future works:
  - let the users choose beetwen more pre-trained models;
  - find the best way to recognize an unknown person (i.e. new student branch members).

Best bugs

- OpenCv `imshow` freezing bug on unix like system.
- Tensorflow-hub requires a tensorflow version that could not work with some processors(precompiled with AVX activation) (Link to issue).

Useful links

- Project repository: Link to repo.
- Tensorflow: Link to Tensorflow page.
- OpenCv: Link to OpenCv project.
- Convolutional Neural Network example: GoogleNet.
- Tensorflow retrain: tensorflow retraining.