Deep Learning Crash Course

By: Josue Cuevas

August-25th 2018

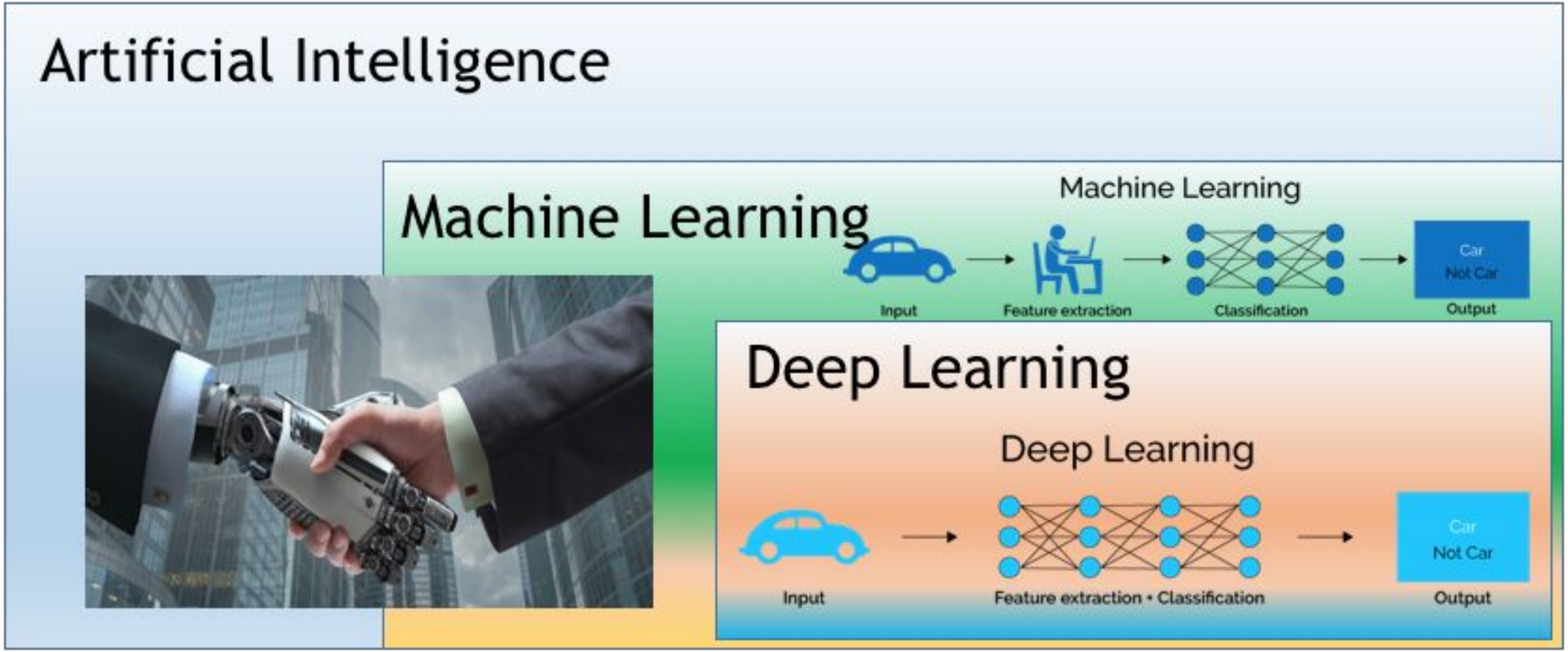
Outline

- **Basics**
  - Deep Learning and Artificial Intelligence: a general idea
  - The concept of learning
  - Basic types of learning approaches
- **Deep Learning Background**
  - Tensors and Tensor rank
  - Matrix and Matrix operations
  - Activation functions, Neurons, Layers
  - Neural Networks
  - Loss Function, gradient descent

Deep Learning Place in Artificial Intelligence

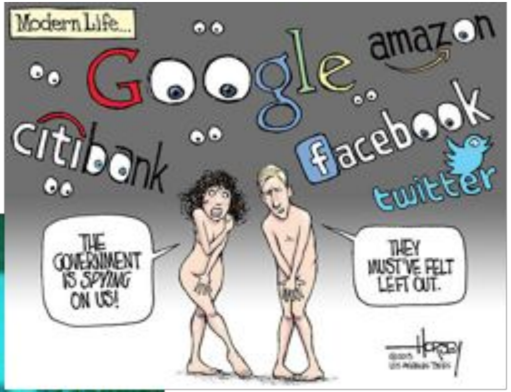
- **Artificial Intelligence (AI)**: focused towards mimicking humans and their constant process of learning. (What, When, How, Why). ***Gurus***
- **Machine Learning (ML)**: Subclass of AI where engineers and researchers are concerned not only on learning, but also on understanding what makes us learn. What Features? Why? ***Experts***
- **Deep Learning (DL)**: It doesn’t concern on the features that make humans learn, but the process in which those features are obtained. ***Silly***

Deep Learning Place in Artificial Intelligence

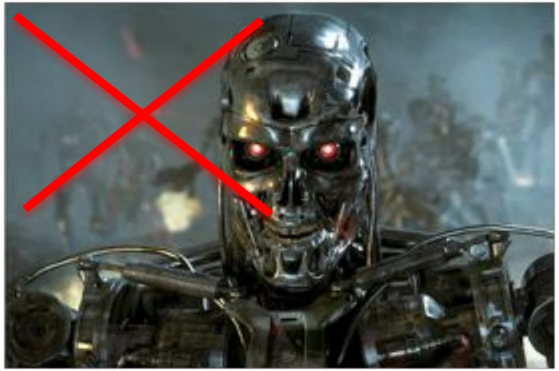


But, what is Learning, and what we do with it?

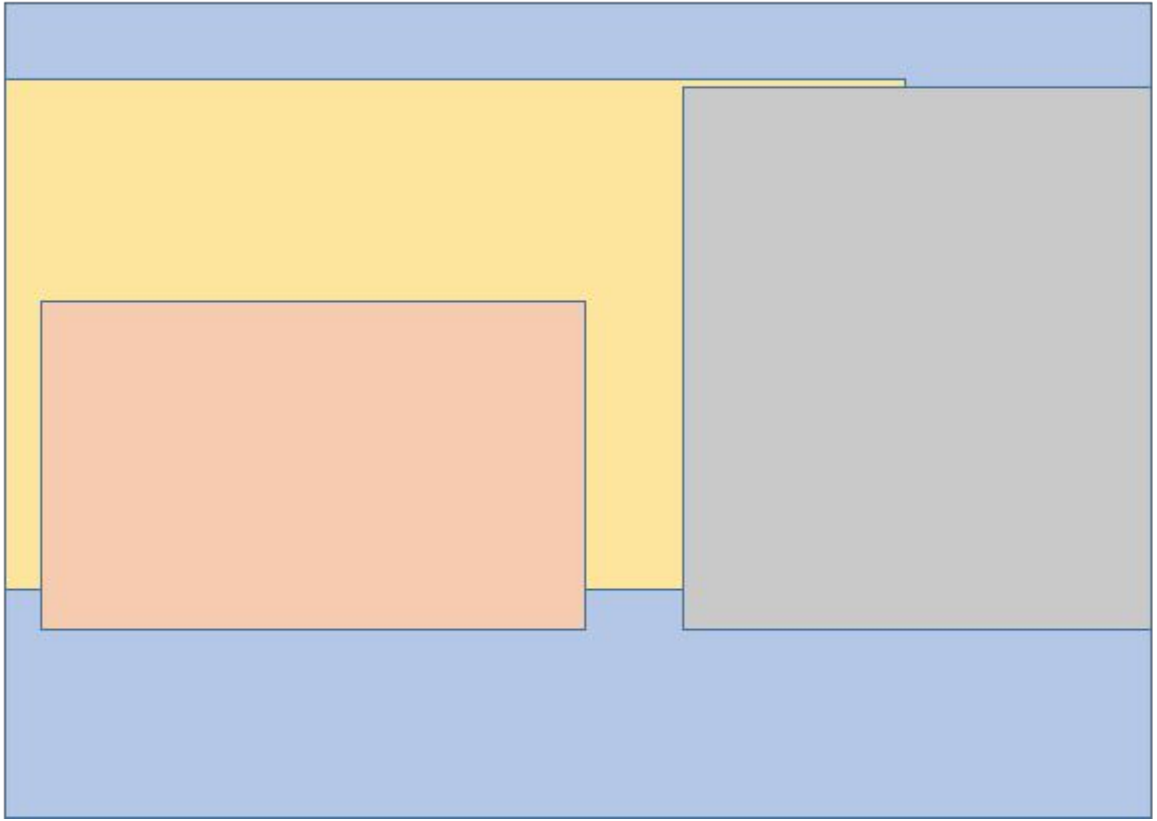
1. **Dictionary**: the acquisition of knowledge or skills through study, experience, or being taught. It helps us to make decisions throughout our lives.
2. **So in a simplistic** way we may approach the problem of learning by:
   a. Learn what is around us: detections.
   b. Understand what are the things around us: recognition/classification.
   c. Put things into context: interpretation of tasks “a” and “b”.
   d. Make decisions: what to do after we have performed task “c”.
3. After we have learned, **how do we use** that knowledge:
   a. GOOD
   b. BAD



This is Wrong!



Learning what is around (Detection)



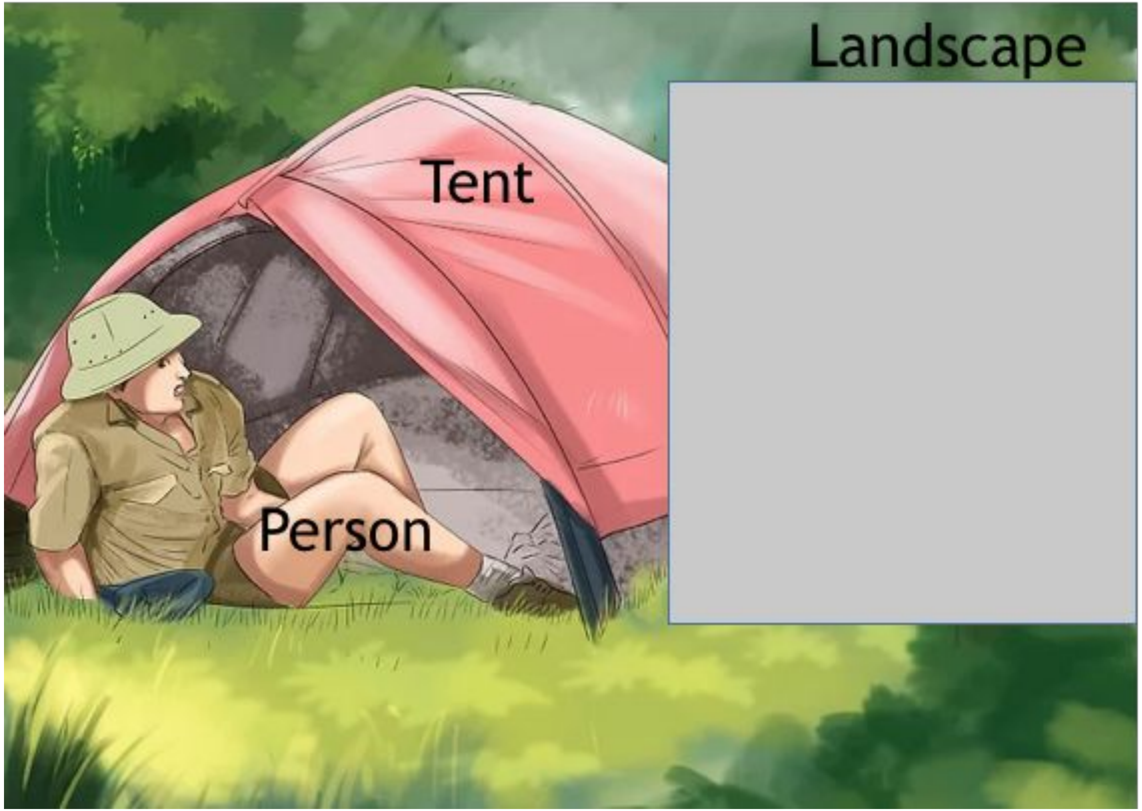
Detection Examples

1. Pedestrian detection
   a. Sample Video 1
2. Vehicle detection
   a. Sample Video 2
3. Lane detection
   a. Sample Video 3

Recognizing what we are detecting

Recognizing what we are detecting

Recognizing what we are detecting



Recognizing what we are detecting

Recognition / Classification examples

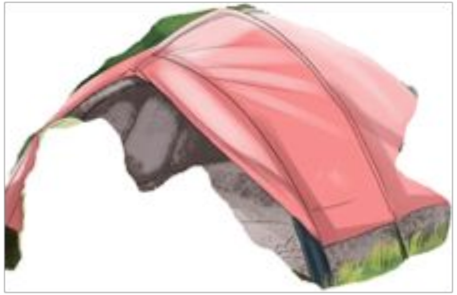
1. Vehicle detection and Speed estimation
   a. Video Example 1
2. Star Wars object detection and classification
   a. Video Example 2
3. Speech Recognition
   a. Video Example 3
4. Scene object detection, segmentation and classification
   a. Video Example 4

But wait ...

- Where is the intelligence in all this?
- They are just good at performing tasks, more like a microwave oven than a person. Wouldn’t you agree?

Let’s put things into context then

- A landscape
- A tent
- A person
- A lion
- A scared face emotion



Let's put things into context then

Okay, so what about it? … well,

- The Landscape is in the wilderness
- The tent is in the middle of the landscape
- The person is sitting next to the tent
- The lion is walking towards the tent
- The person has noticed there is a lion
- His face looks scared

Excellent interpretation of this scene!, We're done!

Wait, … that's it?

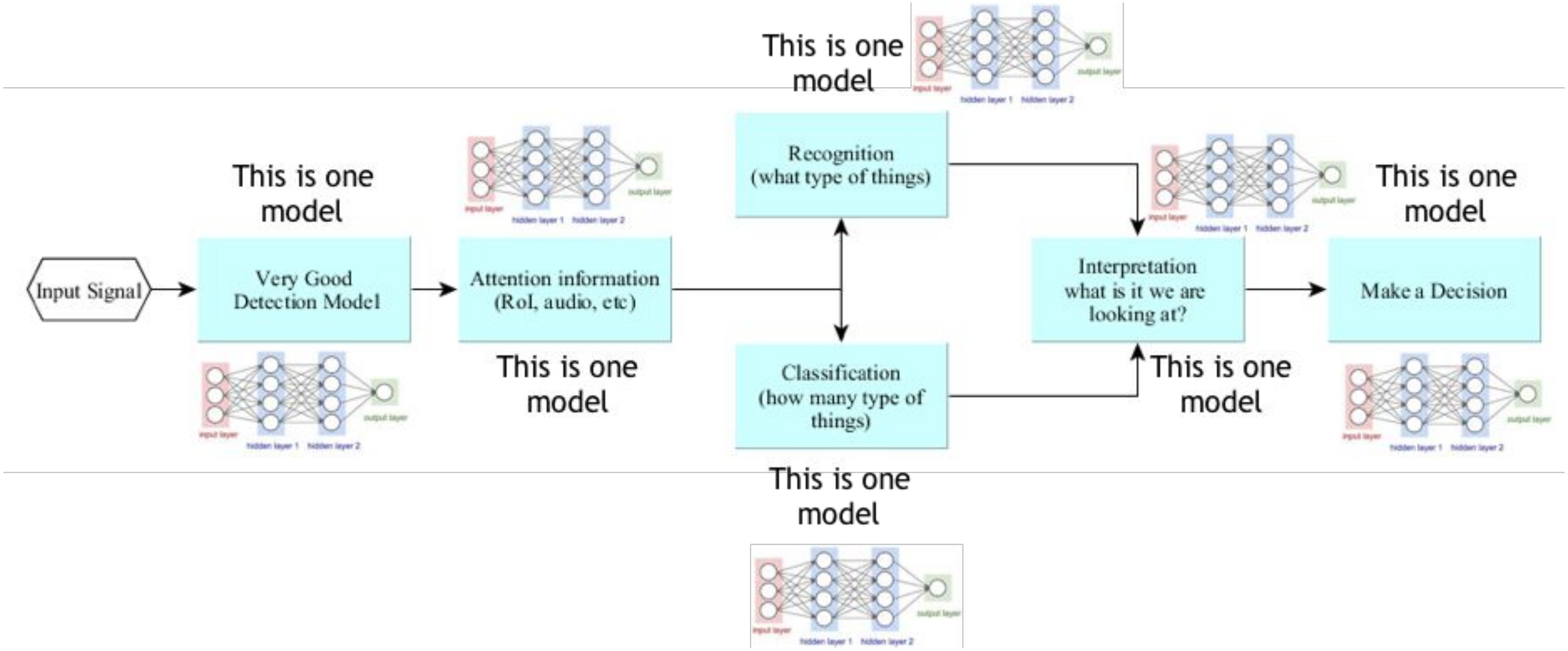
- It is just good at understanding things. That is not smart.
- It is impressive, BUT NOT SMART!
- There is a very important factor missing here, which is, "DECISION MAKING"
- With all the information we can get from analyzing the scenes, and object detection from each particular scene. How to use that in order to make smart decisions?

First let’s check a decision making process

Some examples of decision making process

- Autonomous drone navigation
  - Video Example 1
- Tesla automatic driving
  - Video Example 2
- Visual attention
  - Video Example 3

How are most people approaching the problem



Now, how do we learn all this? - Part 1

- In a supervised manner? what do we need?

How do we guide the learning process towards specializing on specific tasks? For instance,

1. We want to detect dogs? – we need bounding boxes of dogs
2. How about classifying type of flowers? – flowers types pictures
3. How about identifying people in a company? – subjects faces pictures
4. Voice command devices? – list of commands to be identified
5. Bug report email classification? – word classification and context recognition – emails with subjects to be classified
6. Music genre classification? – songs of different genres

Supervised Learning Examples

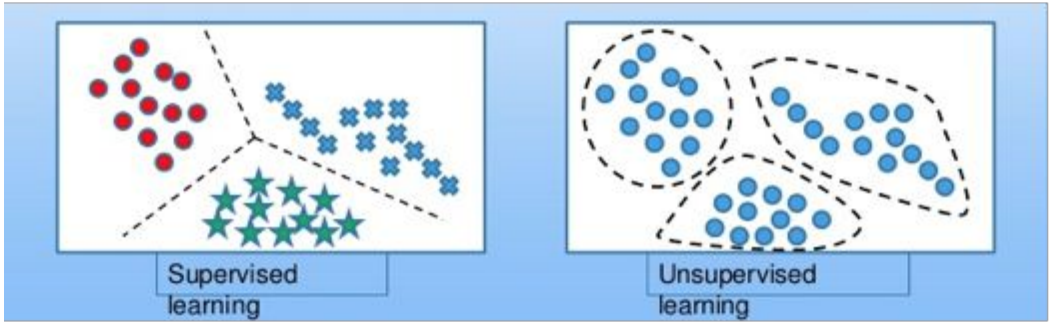
1. Face detection
   a. Video Example 1
   b. Video Example 2
2. Breathing detection
   a. Video Example 3
   b. Video Example 4
3. Defect classification (copper films)
   a. Video Example 5

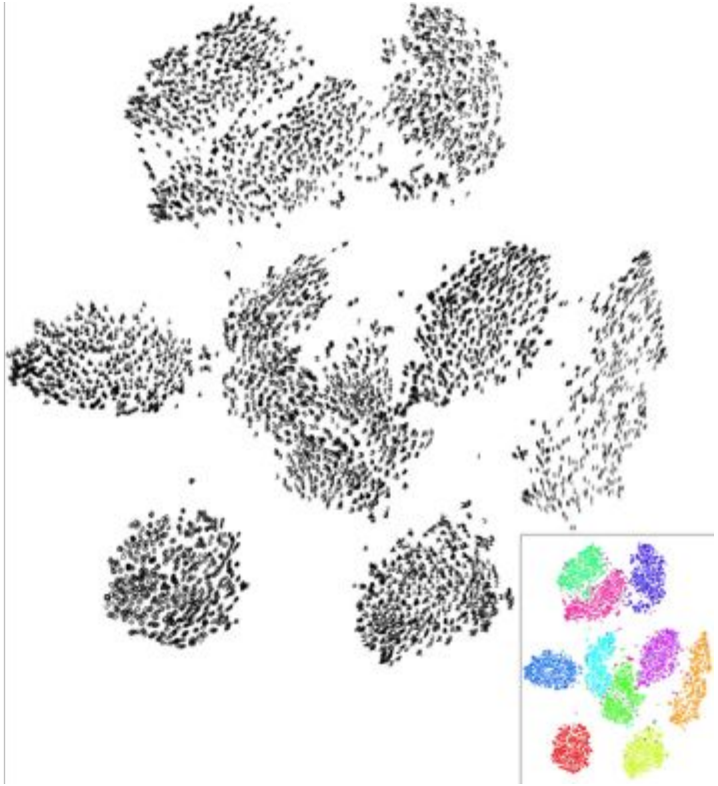
Now, how do we learn all this? - Part 2

- In a unsupervised manner? what do we need?

1. It is unsupervised in the sense that you don't have to provide labels or target answers which need to be learned by the model.
2. It is not unsupervised in the sense that you don't need to know the kind of data you are looking at.
3. You extract patterns, learn features from the input data.
4. However, you NEED to know what you are looking at, otherwise you are just playing with your data.
5. You should use this as a tool to help you to understand your data characteristics.

Unsupervised vs Supervised





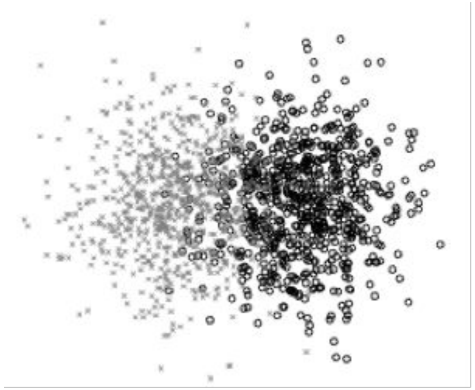
Unsupervised Learning Examples

1. Fabric defect localization
   a. Video Example 3
   b. Video Example 4
   c. Video Example 5
   d. Video Example 6
   e. Video Example 7

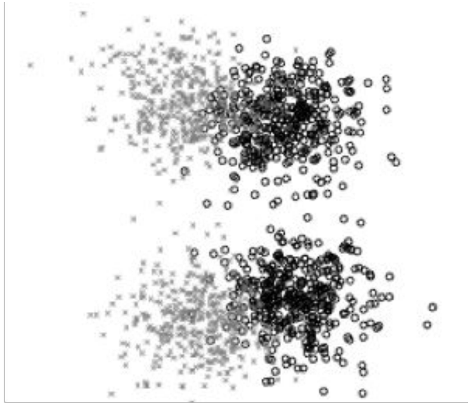
Now, how do we learn all this? - Part 3

How about semi-supervised learning?

There is lots of data in this category, and anything which is partially labeled or partially known could be considered as semi-supervised. For instance:



Supervised



Semi-supervised

Semi-supervised learning example

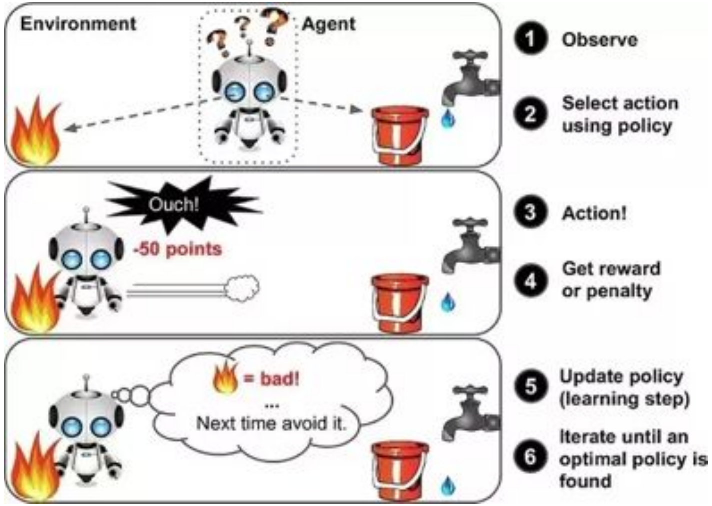
- Augmentation Data based on AC-GAN models
  - Bean Augmentation

Question:

Why do you think this could be considered a kind of unsupervised learning? (please read this paper for a better understanding, https://arxiv.org/pdf/1610.09585.pdf)

Now, how do we learn all this? - Part 4

We can also make use of reinforcement learning:
Where we penalize bad decision by the system, and give rewards for those good decisions. Think of it as a sort of pet training process. For instance:



Reinforcement Learning Examples

- A classic example you will see in every presentation, tutorial, courses, etc.
  - Cart-Pole
- A more realistic, drone navigation
  - Indoor navigation simulation
  - Multi-robot collision avoidance

Okay, enough of introductions, let’s take a look of the theoretical background you need to have.

Installing Pycharm

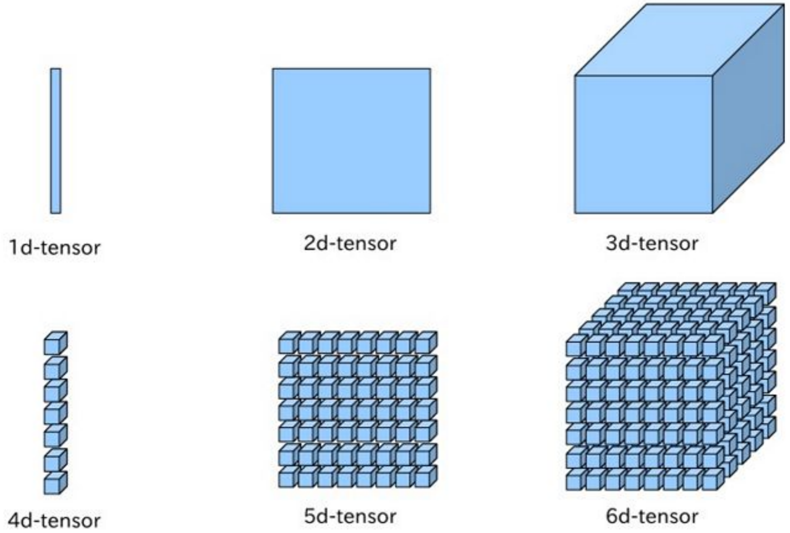
1. Main website
2. Download the community version of Pycharm
3. pycharm go to interpreter in file → settings → interpreter and install the package you want
4. Lets install Numpy and Tensorflow, which will be very helpful to understand tensors, matrices and other operations.
5. Test installation by typing the following command in a new pycharm file.

```
import numpy as np
import tensorflow as tf
```

6. Now go to Run → Run

Tensors and Matrices

A tensor is a generalization of vectors and matrices to potentially higher dimensions.



Tensors and Matrices in Python (Numpy)

1. [Tensor Creation]
2. [Tensor Addition]
3. [Tensor Subtraction]
4. [Tensor Multiplication]
5. [Tensor Division]
6. [Tensor Product]

Tensors and Matrices in Python (Tensorflow)

1. Tensor Creation
2. Tensor Addition
3. Tensor Subtraction
4. Tensor Multiplication
5. Tensor Division
6. Tensor Product

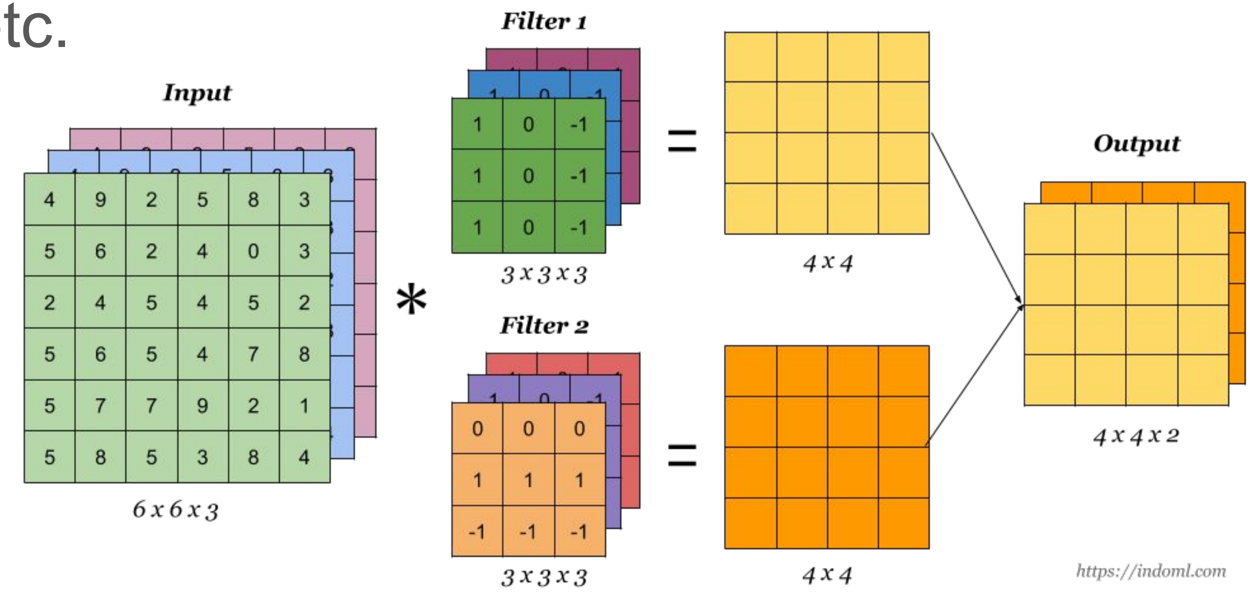
Full Tensorflow API

You can find all the information regarding all the supported operations in Tensorflow in the following links:

1. **Tensorflow APIs**: https://www.tensorflow.org/api_docs/
2. **Tensorflow Python API**: https://www.tensorflow.org/api_docs/python/
3. **Tensorflow C++ API:** https://www.tensorflow.org/api_docs/cc/

Filter

Filters: are “small” vector/matrices/etc. which are usually convolved (element-wise multiplied) with larger vectors/matrices/etc.



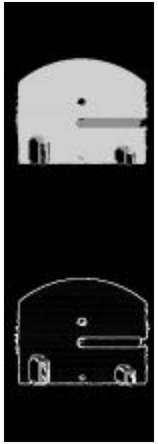
Examples of filters

-1	0	+1
-2	0	+2
-1	0	+1

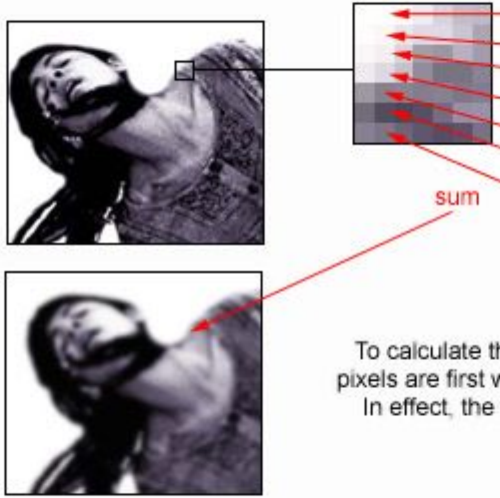
Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy



Edge detection filters



0	0	0	5	0	0	0
0	5	18	32	18	5	0
0	18	64	100	64	18	0
5	32	100	100	100	32	5
0	18	64	100	64	18	0
0	5	18	32	18	5	0
0	0	0	5	0	0	0

To calculate the shade of a single destination pixel, a group of source pixels are first weighted by the filter kernel, and then summed together. In effect, the filter kernel is "slid" across the source image, producing one destination pixel for each kernel position.

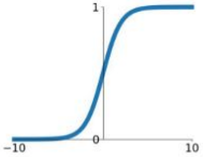
Gaussian Blurring filters

Activation Function commonly used

[Activation functions](): defines the output of a neuron given an input or a set of inputs.

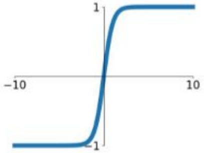
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



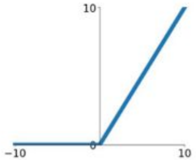
tanh

$\tanh(x)$



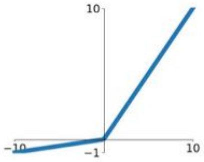
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

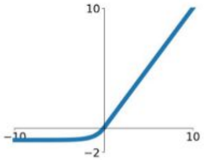


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$



Softmax

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

Example of activation functions

1. Softmax Function
2. Tanh Function
3. Relu Function
4. Leaky Relu

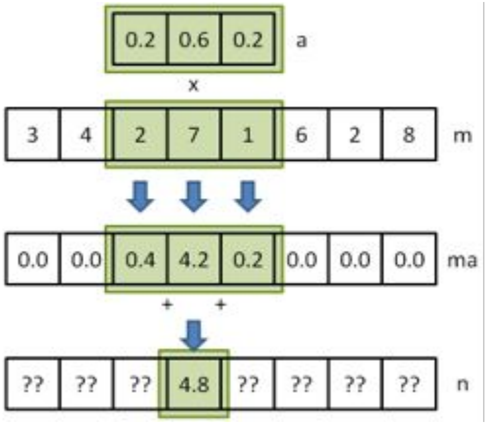
How do we use them?

Why these functions?

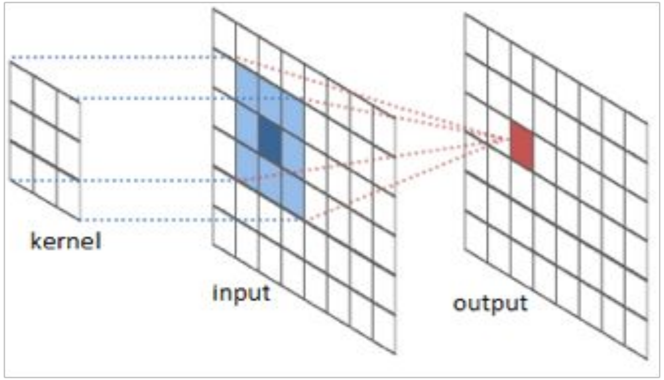
Advantages and Disadvantages.

Convolution

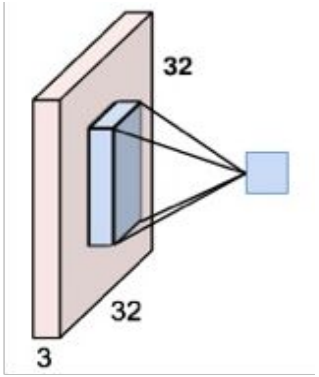
convolution is a mathematical “***OPERATION***” on two functions (f and g) to produce a third function that expresses how the shape of one is modified by the other. Visualize



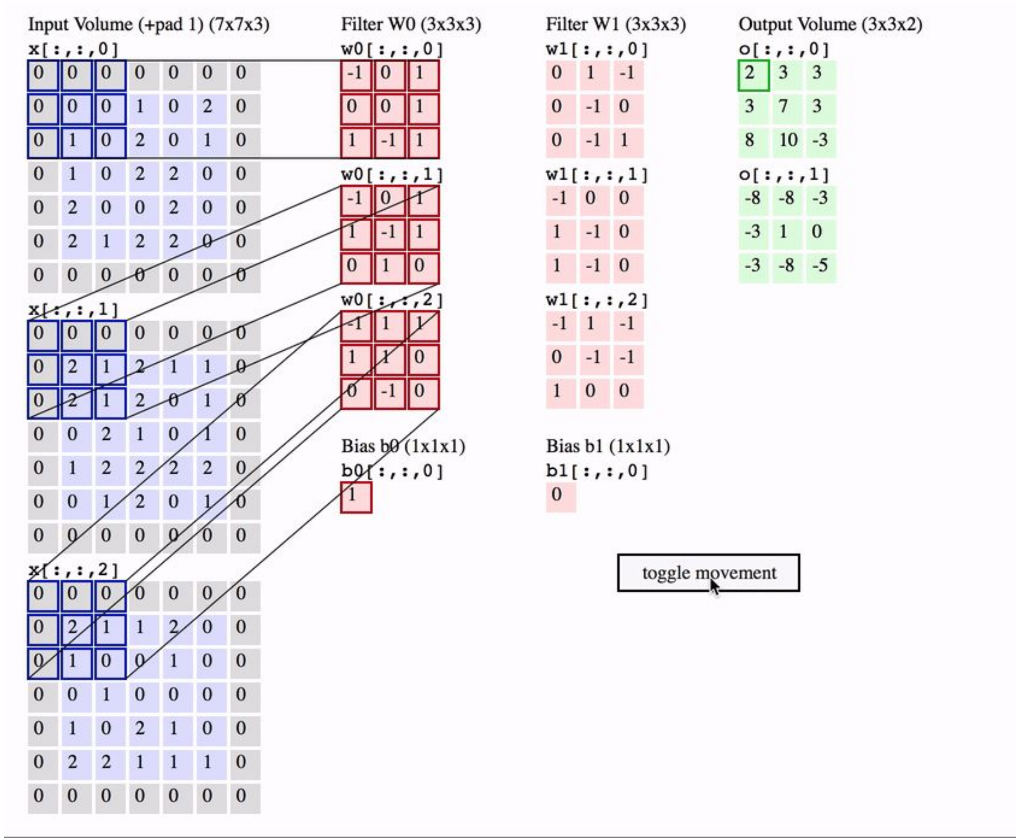
Code sample



Code Sample

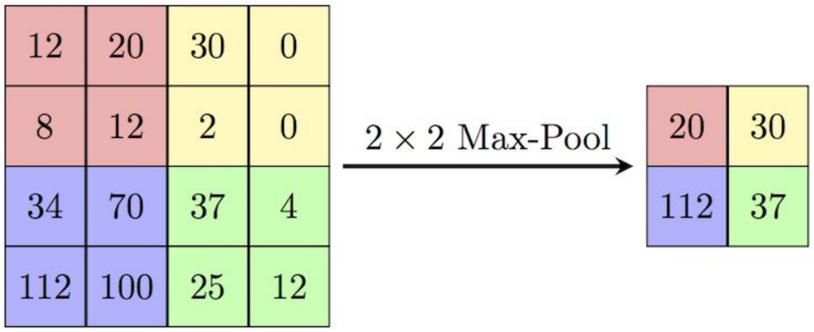


Convolution Illustration

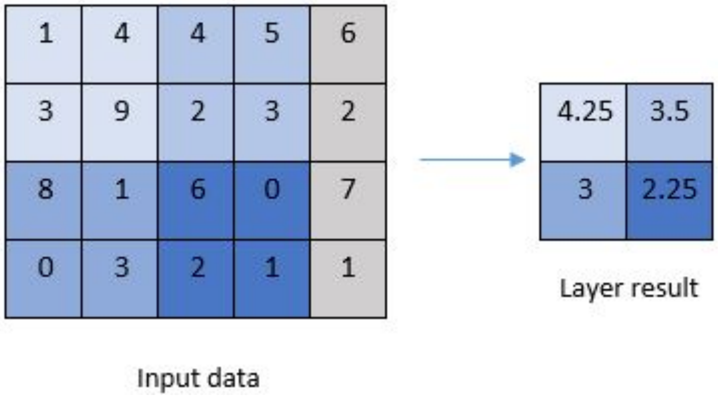


Pooling

Pooling: is the subsampling of an input signal, the two most common types are, MaxPooling, AveragePooling.



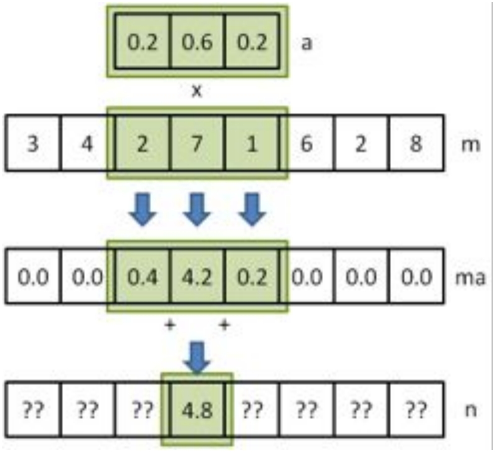
Max Pooling

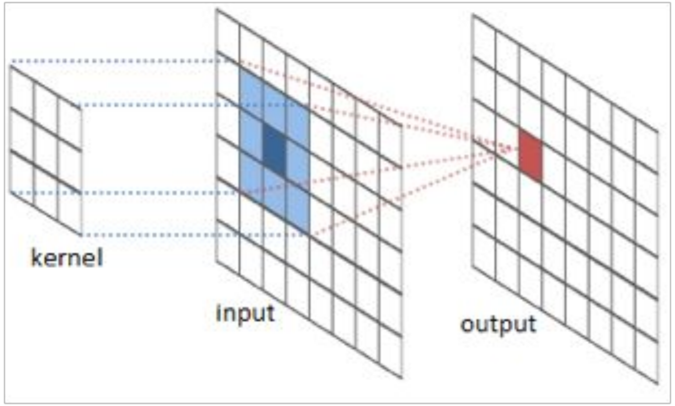


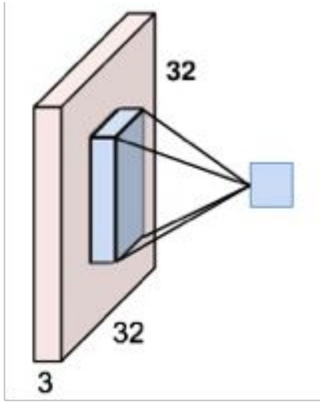
Average Pooling

Neurons

A neuron: it is the combination of a 1-D, 2-D, 3-D "***FILTER***" which is to be multiplied or convolved through an input signal (image, audio, video) and an activation function.

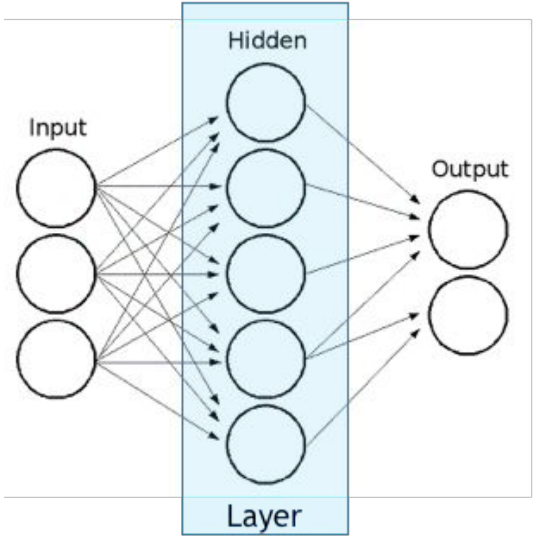


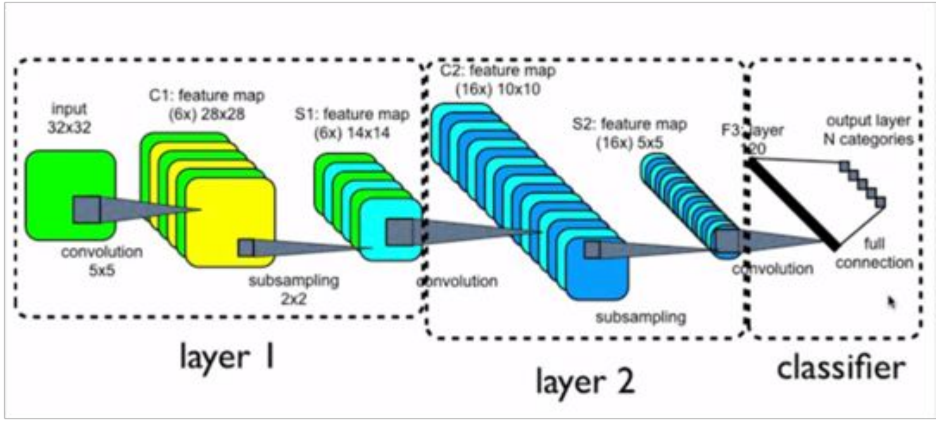




Layer

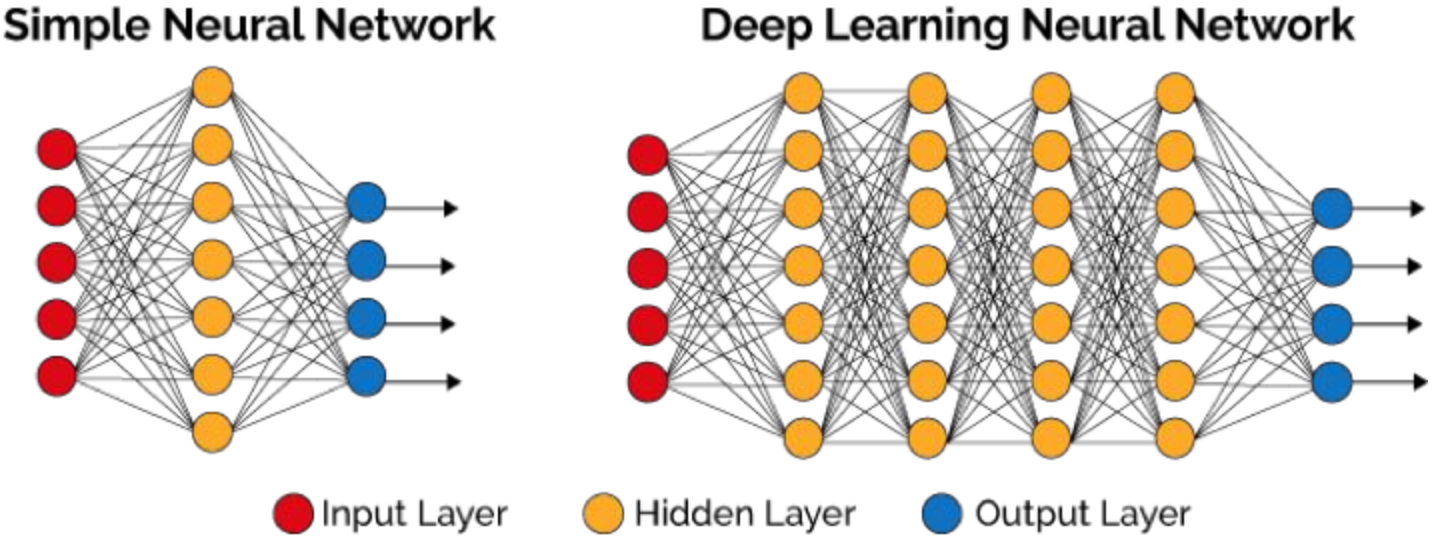
A Layer: in ***most cases*** it is just a bunch of **neurons/filters** extracting features from an input signal, by implementing **convolutions**, **pooling**, **activations**, and other operations.





Network

A Network: think of a it as a **bunch of layers** which primary objective is to extract important features from an input signal in order to perform an specific task.



Loss Function

Loss Function: it is what we are trying to optimize during the process of training our neural network. It basically tells you how good the output of the network is with respect to what you expect from it. For example:

1. Classification: cross entropy loss function
2. Signal reconstruction: PSNR, MSE(L2), MAE(L1) loss function
3. Object detection: Regression loss function
4. Object segmentation: Intersection over Union loss function
5. **What else?** Basically anything that will help you to measure the network performance during training.

Loss Functions

$$\text{cross-Entropy} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

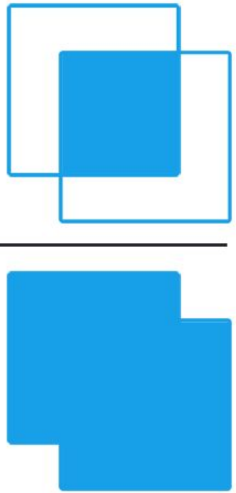
$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$$

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{e_t}{y_t}\right|$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}|e_t|$$

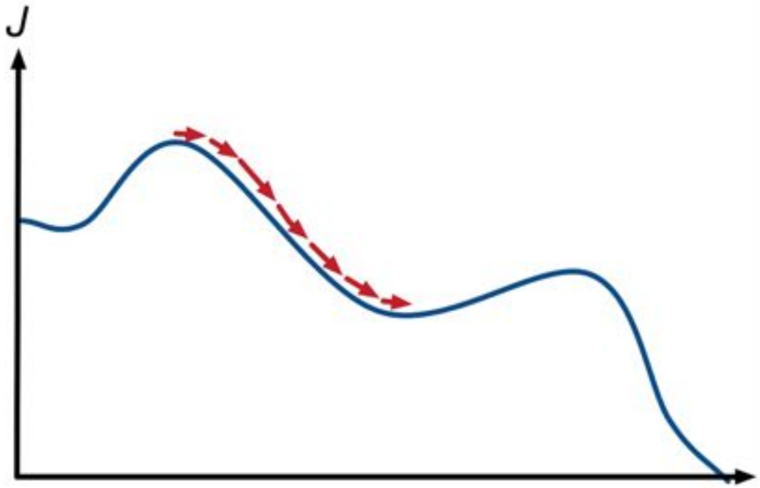
$$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2$$

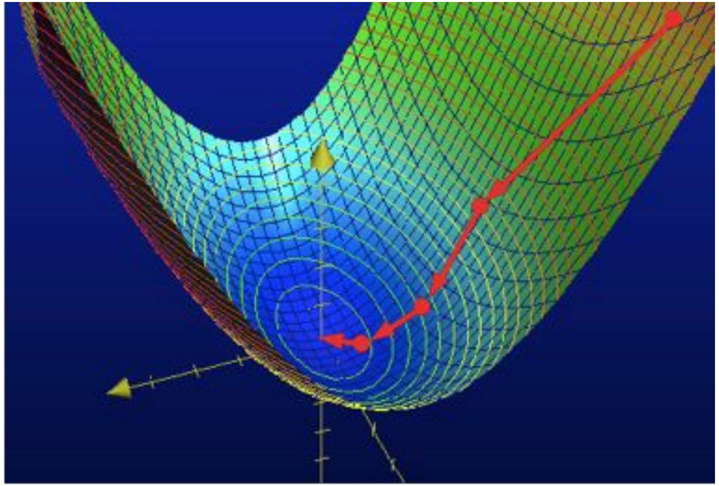
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Gradient Descent

This is what we use in order to optimize the loss function. What is does basically is to modify the filter values so the Loss function is minimized according to the task at hand.





Why do we use this?

- It is easy to compute for differentiable functions, and relatively inexpensive. Just a bunch of derivatives
- BUT, it is not exactly the “Gradient Descent” we used to implement, since this one uses the whole dataset in order to compute the gradients. It is the “Stochastic Gradient Descent” that we used, because the gradients can be computed per sample(s).
- Can we use other algorithms? Let’s discuss some of them!

Any Example?

Reference Literature

Book:

https://books.google.com.tw/books/about/Deep_Learning.html?id=Np9SDQAAQBAJ&redir_esc=y&hl=en

Deep learning and toolkits survey:

https://arxiv.org/pdf/1803.04818.pdf