aivancity
SCHOOL FOR
TECHNOLOGY, BUSINESS & SOCIETY
PARIS-CACHAN

PGE 5

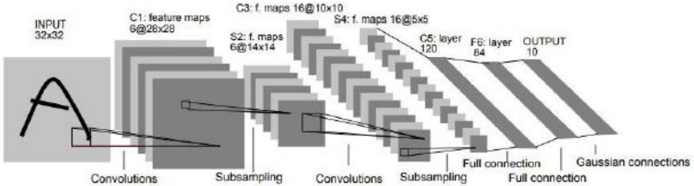
Deep Learning for Computer Vision

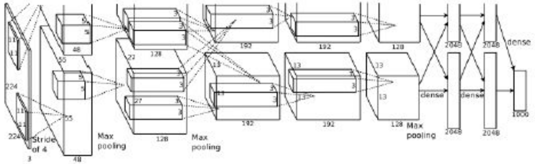
Dr. Y. Almehio
April, 2023

Review

Part 2

Brief history of Deep learning





LeCun, Bengio, 1998
LeNet-5
Gradient-based learning

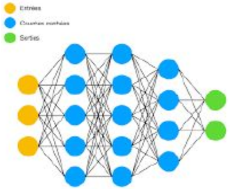
Minsky & Papert, 1969
perceptron

Krizhevsky, Hinton, 2012
AlexNet

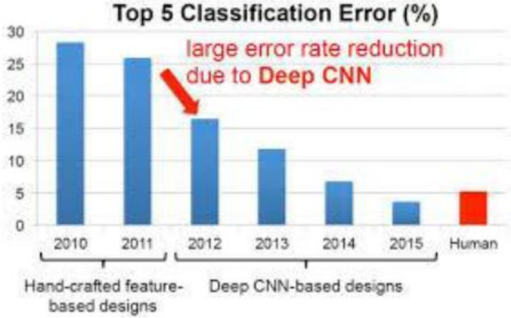
LeCun, 1990
convolutional networks

Li Fei-Fei, 2009
Image-net
22K categories and 15M images

Ross Girshick, 2016
Faster RCNN



ImageNet Large Scale Visual Recognition Challenge
Russakovsky et al. IJCV 2015



aivancity
PARIS-CACHAN

Part 3

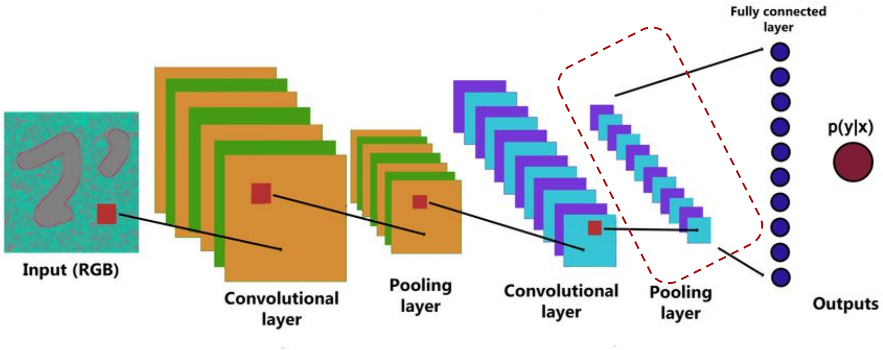
Deep learning Comes back !

Previous Course: DL: Methods & Applications

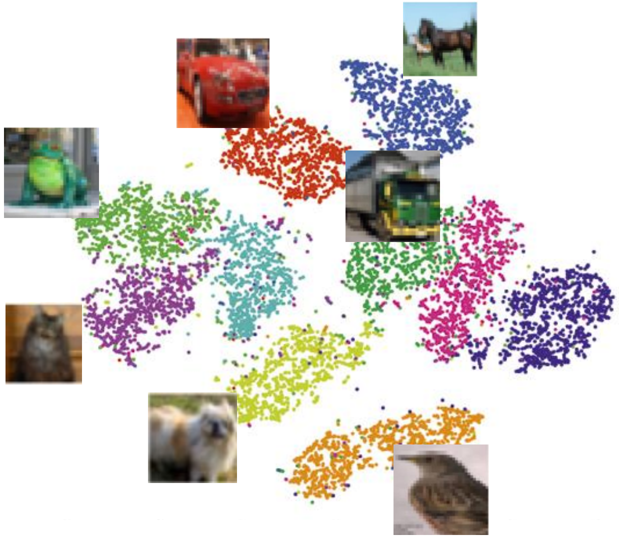
Part 3

Feature space !

- Set of all possible features or variables that can be used to describe a particular data point.



Z



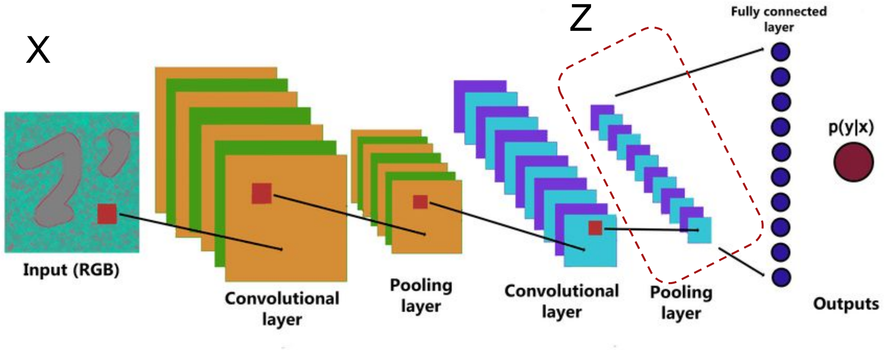
Part 3

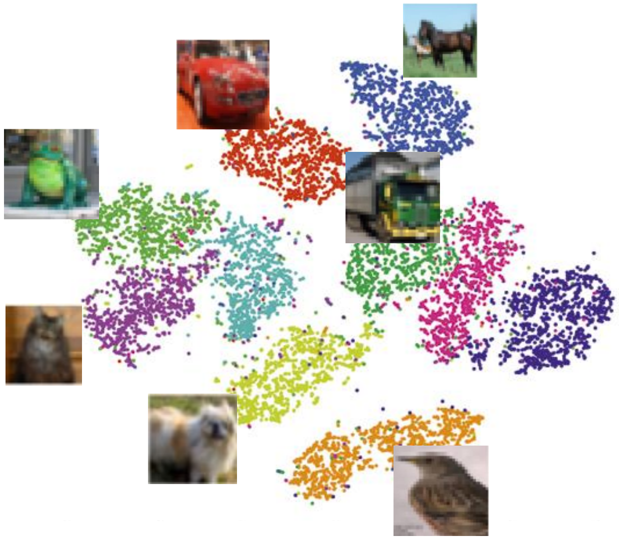
Feature space : encoder

- Set of all possible features or variables that can be used to describe a particular data point.

X

Z





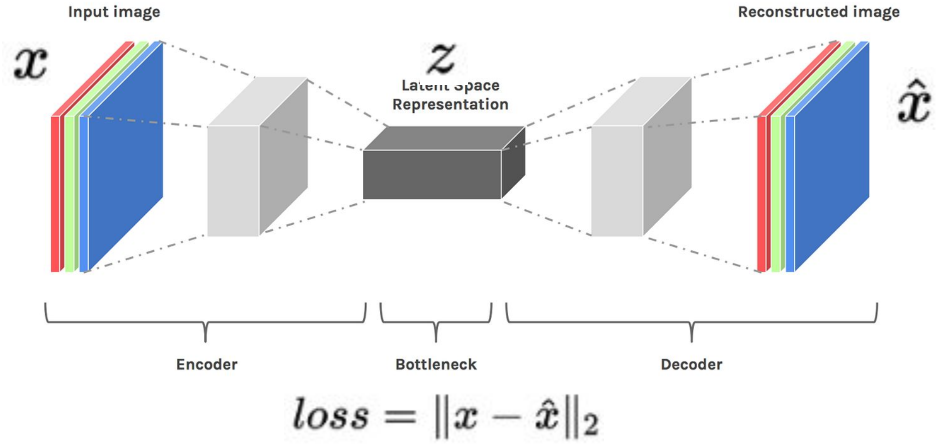
Z = f(w.X+b) ⇒ encoding X into Z (latent space)

Is there a function to decode Z into X^ ≃X ?

X^ = h(Z)

Part 3

Encoder + Decoder = Autoencoder

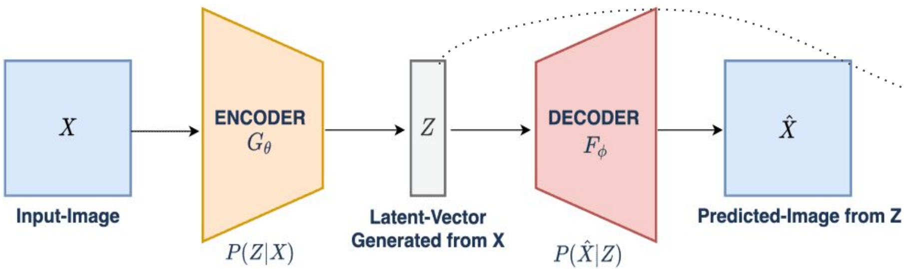


Since the bottleneck is designed in such a way that the maximum information possessed by an image is captured in it, we can say that the bottleneck helps us form a *knowledge-representation* of the input.

The aim of an autoencoder is to learn a lower-dimensional representation (encoding) for a higher-dimensional data, typically for dimensionality reduction, by training the network to capture the most important parts of the input image.

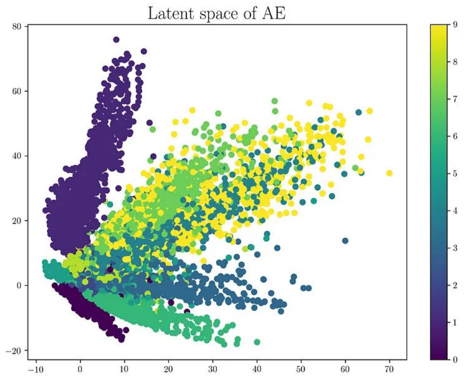
A good comparison PCA & AE

Encoder + Decoder = Autoencoder



ref

$$loss = \|x - \hat{x}\|_2$$



ref

- It can be seen that the same digits tend to cluster themselves in the latent space
- the latent space **is not regularized** ?

Parts of the latent space that doesn't correspond to any data point. Using those as inputs to the encoder will result in an output that doesn't look like any image of training dataset

Part 3

Autoencoder: AE
Types, Applications

Types:

- Denoising autoencoders
- Contractive autoencoder
- Sparse autoencoder
- Variational autoencoder (generative model)

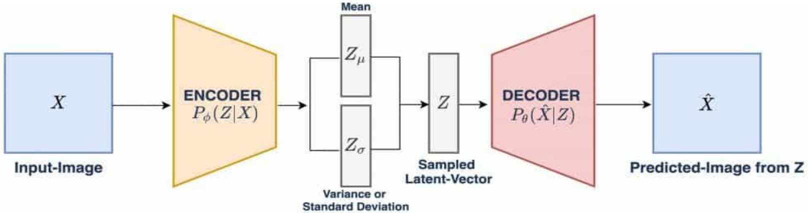
Application

- Image denoising
- Semantic segmentation
- Dimensionality reduction / Image compression
- Feature extraction
- Image search

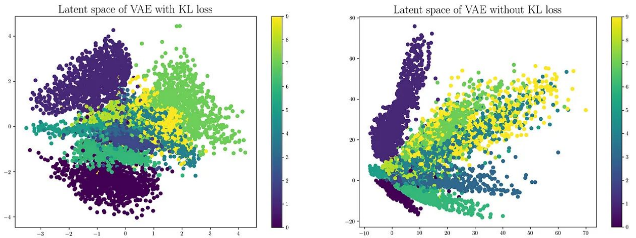
Part 3

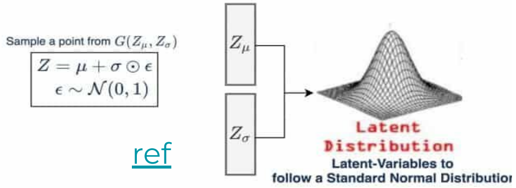
Variational Autoencoder

VAE



ref





ref

- It is a solution to tackle the problem of non-regularized latent vector provides the generative capability to the entire space
- VAE's encoder outputs parameters of a pre-defined distribution in the latent space for every input
- VAE then imposes a constraint on this latent distribution forcing it **to be a normal distribution ⇒ regularized**
- **Loss Function**
  a. Minimize the reconstruction error
  b. Latent space should follow a gaussian/normal distribution

$$L(\phi, \theta, x) = \frac{1}{N}\sum_{i=1}^{N}(X_i - \hat{X}_i)^2 + KL[G(Z_\mu, Z_\sigma), \mathcal{N}(0,1)]$$

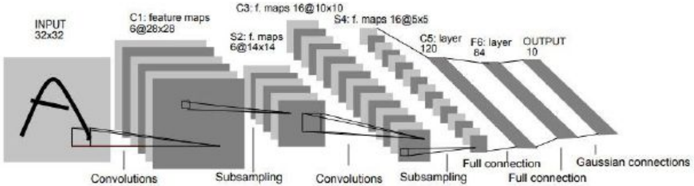
[Kullback−Leibler Divergence](): measure how one probability distribution differ from another

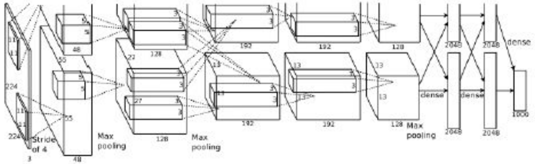
Part 3

Autoencoders Project

- Colab notebook: Generative models: Autoencoder

Brief history of Deep learning





Minsky & Papert, 1969
perceptron

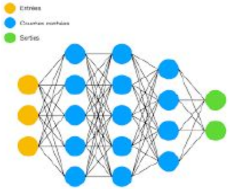
LeCun, 1990
convolutional networks

LeCun, Bengio, 1998
LeNet-5
Gradient-based learning

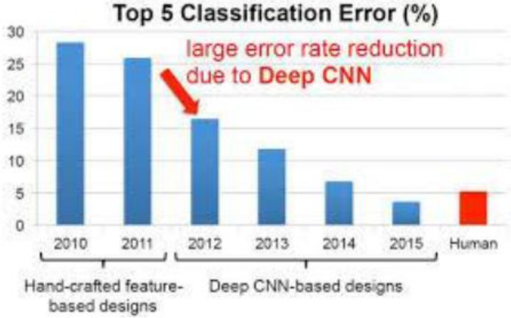
Li Fei-Fei, 2009
Image-net
22K categories and 15M images

Krizhevsky, Hinton, 2012
AlexNet

Ross Girshick, 2016
Faster RCNN



ImageNet Large Scale Visual Recognition Challenge
Russakovsky et al. IJCV 2015



Inria
aivancity
PARIS-CACHAN

Part 4

Deep learning Comes back !

Previous Course: DL: Methods & Applications

Part 3

Autoencoders Project

Types, Applications

Types:

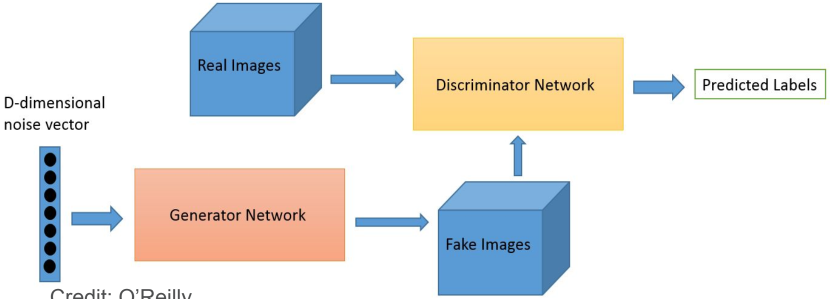
- Classic autoencoders
- Denoising autoencoders
- Variational autoencoder (generative model)

[Colab notebook]

Part4

GAN
Definition

- Generative adversarial networks (GANs): architectures that use two neural networks, competing one against the other (thus the “adversarial”) in order to generate new, synthetic instances of data that can pass for real data. They are used widely in image generation, video generation and voice generation.



Credit: O’Reilly

- The two models will work against each other, with the generator trying to trick the discriminator, and the discriminator trying to accurately classify the real vs. generated images.

Part 3

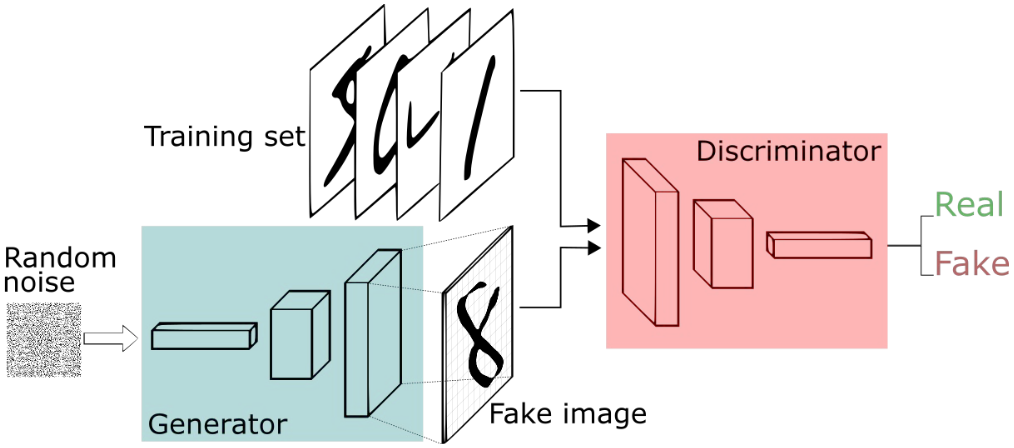
GAN
Definition

1. The discriminator takes in both real and fake images and returns probabilities, a number between 0 and 1, with 1 representing a prediction of authenticity and 0 representing fake.
2. The generator takes in random numbers and returns an image.
3. This generated image is fed into the discriminator alongside a stream of images taken from the actual, ground-truth dataset.

The Different Types of GANs

- Vanilla GAN
- Conditional Gan (CGAN)
- Deep Convolutional GAN (DCGAN)
- CycleGAN
- Generative Adversarial Text to Image Synthesis
- Style GAN
- Super Resolution GAN (SRGAN)

[Types of GANs] to read



Credit: *Thalles Silva*

Final Project: Cartoonize images (1/3)

Your project consists of two main parts

1. Traditional computer vision: to develop an algorithm to create you dataset (ground truth)
2. Deep CV : to develop a GAN network to transform real images into non-real (cartoonized images)

Final Project: Cartoonize images (2/3)

1. **Create your dataset**

In order to train GAN you need to collect dataset that have a complete ground truth ⇒ paired images (real <> cartoonized). So you have to follow the steps:

- Collect any public dataset for the real world (faces, animals, nature, roads.....)
- Develop a script (opencv) to stylish images, to transform each image into the desired style (cartoon)
- Opencv has several built-in functions that can help you. There is no only one possible, try to give the best of you to make the image render as beautiful as possible. For this step, you need to check these blogs:
  - Non-Photorealistic Rendering
  - Cartoonify reality
  - Cartoonize your image
- Now, you get paired image, for each input (real), you have its corresponding output (cartoonized), thus you have a dataset
- Split datasets into train and test parts

Final Project: Cartoonize images (3/3)

2. **Train Pix2Pix GAN (know as conditional GAN: cGAN)**

This model learn to map input images to output images, in our case real to non-real, this problem is known as well as image to image translation. You may get benefits from the available GAN:

- Image to sketch translation
- Style transfer
- Some useful tutorials:
  - How to Develop a Pix2Pix GAN
- Show the validation of model on test part
- Show the validation result on new real images
- Show a demo on a video
- Explain well your code, steps, readme
- The final project must be pushed in a github repo
- Deadline: 07/05, 23:59

Final Project: hints

- Generator takes as input a real image and produces a cartoon version
- Discriminator tries to distinguish between the generated cartoon images and the real cartoon images from the dataset.
- Loss functions: could be a combination of adversarial loss and L1 loss.
  - Adversarial loss is used to train the generator to produce images that are realistic and match the cartoon style
  - L1 loss is used to ensure that the generated images are similar to the real cartoon images.
- After training: try your new real images to cartoonify it
- Some other types of GAN is possible as well!