DE VINCI INNOVATION CENTER

DEEP LEARNING

A Modern Approach to Artificial Intelligence

Ylierss **HATI**

PHD Student - Computer Science

yliess.hati@**devinci.fr**

DE VINCI HIGHER EDUCATION · ESILV ENGINEERING SCHOOL DE VINCI PARIS · IIM DIGITAL SCHOOL DE VINCI PARIS · EMLV BUSINESS SCHOOL DE VINCI PARIS

|00 INTRODUCTION

Perceptron
Rosenblatt
1958

Perceptrons
Minsky & Seymour
1958

Boltzmann Machine
Hinton
1985

CNN
LeCun
1989

Contrastive Divergence
Hinton
2002

GAN
Goodfellow
2014

1959
Hubel & Wiesel
Cat Visual Cortex

1979
Fukushima
NeoCognitron

1986
Smolenski
Harmonium
Hinton
RBM
Rumelhart, Hinton & Williams
MLP
Jordan
RNN

1997
Hochreiter & Schmidhuber
LSTM
Schuster & Paliwal
BRNN

2012
Hinton
Dropout

2017
Sabour, Frosst & Hinton
Capsule Network

|00 INTRODUCTION

AlexNet
Krizhevsky, Sutskever & Hinton
2012

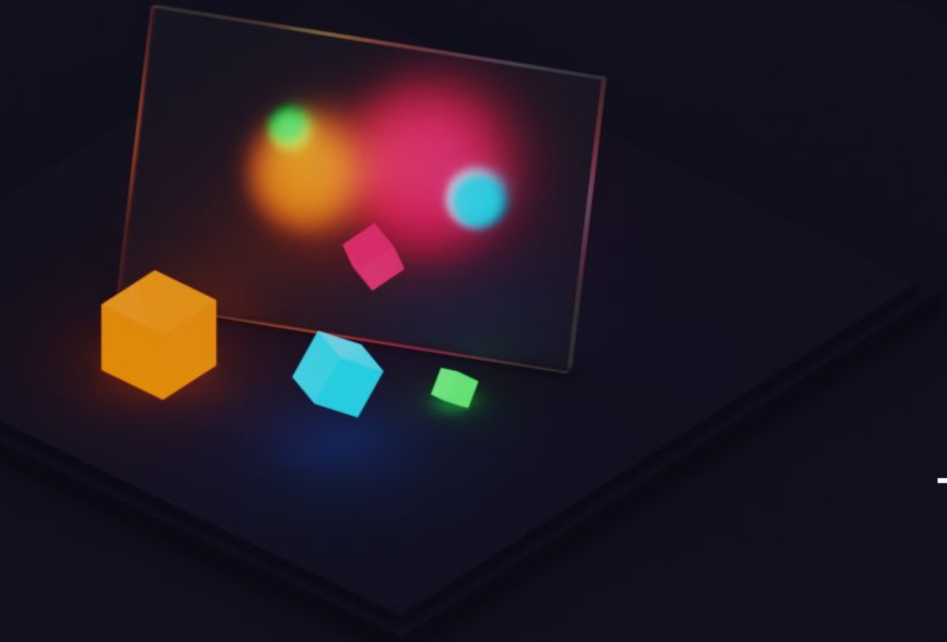
ResNet
He, Zhang, Ren & Sun
2015

ResNetXt
Xie, Girshick et al.
2019

2014
Simonyan & Zisserman
VGG
Google
Inception Network

2016
Huang et al.
DenseNet

DE VINCI INNOVATION CENTER



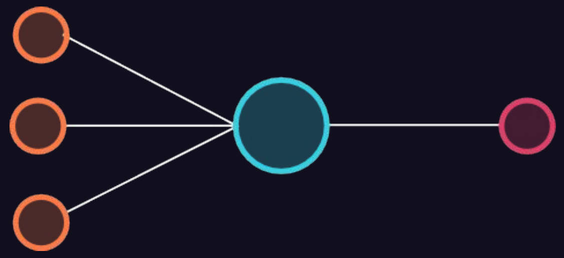
01|

PERCEPTRON

The Beginning and the End

DE VINCI INNOVATION CENTER

|01 PERCEPTRON



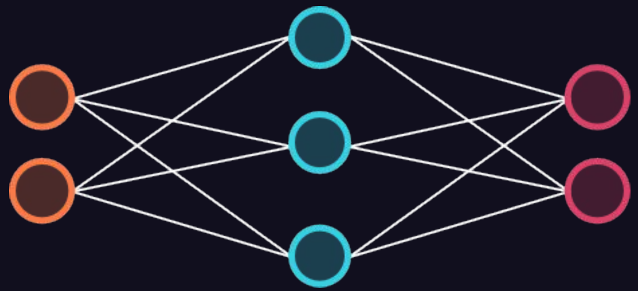
PERCEPTRON

$$\hat{y} = f(wx + b)$$

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

DE VINCI INNOVATION CENTER

|01 PERCEPTRON



MULTILAYER PERCEPTRON

$$\hat{y} = f(w_2 h + b_2)$$

$$h = f(w_1 x + b_1)$$

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

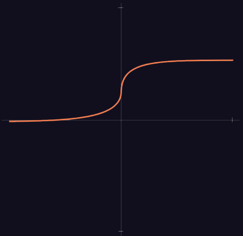
|01 PERCEPTRON

ACTIVATION FUNCTIONS

Step

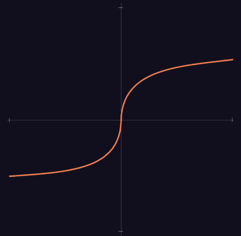
$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

Sigmoid



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

Tanh



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

ReLU

$$relu(x) = max(0, x) = x^+$$

DE VINCI
INNOVATION
CENTER

ACTIVATION FUNCTIONS

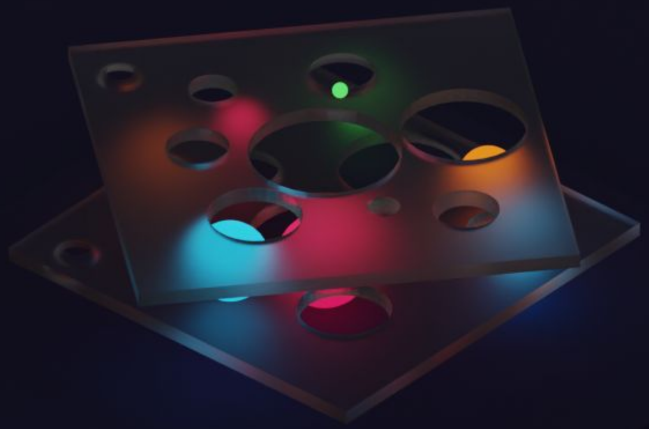
Softmax

$$p_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

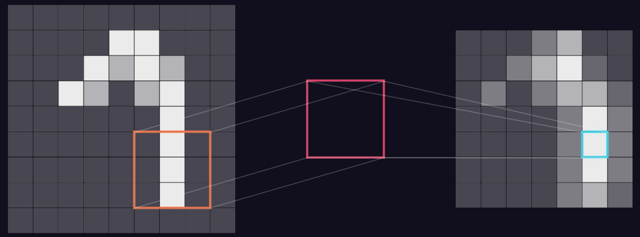
|02

CONVOLUTION

Signal Processing 101



DE VINCI
INNOVATION
CENTER



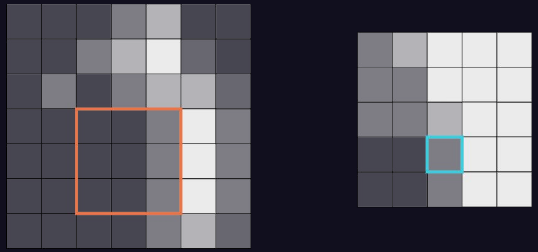
CONVOLUTION
CROSS CORRELATION

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(x)g(x-t)dt$$

Weight Sharing

|02 CONVOLUTION

POOLING



Dimensionality Reduction

DE VINCI INNOVATION CENTER

|02 CONVOLUTION

CONVOLUTIONAL NEURAL NETWORK

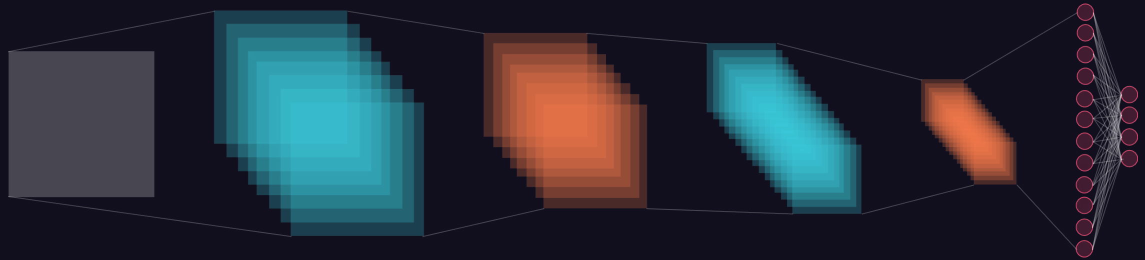


Image Input | Convolution & Pooling | MLP

DE VINCI INNOVATION CENTER

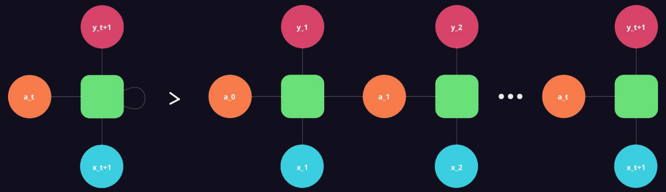
03|

RECURRENT

Backprop Through Time

DE VINCI INNOVATION CENTER

RECURRENT CELLS

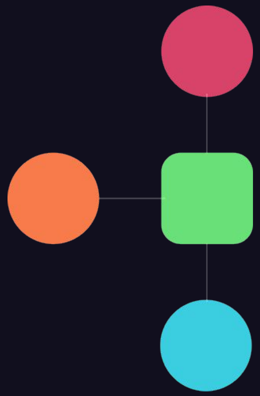


Weight Sharing & Backprop Through Time

$$a_t = g_1(W_{aa} a_{t-1} + W_{ax} x_t + b_a)$$

$$y_t = g_2(W_{ya} a_t + b_y)$$

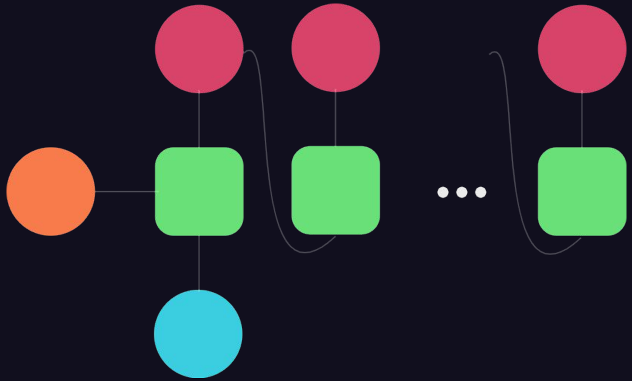
DE VINCI INNOVATION CENTER



ARCHITECTURES

One to One

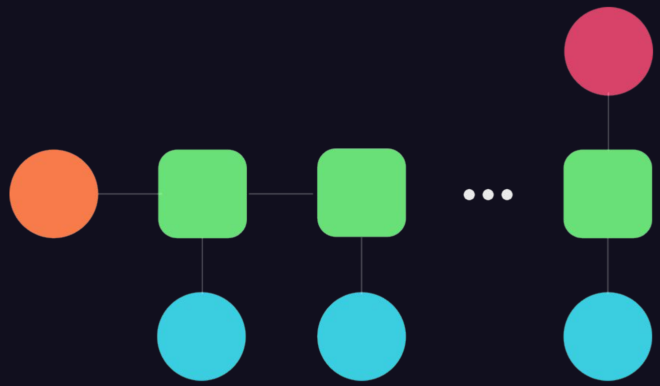
Traditional Neural Network



ARCHITECTURES

One to Many

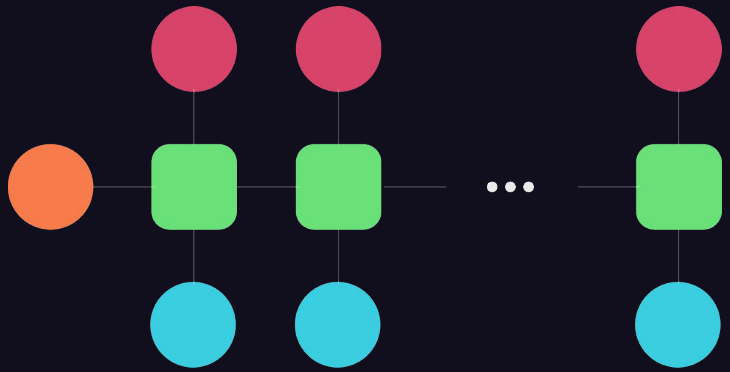
Music Generation



ARCHITECTURES

Many to One

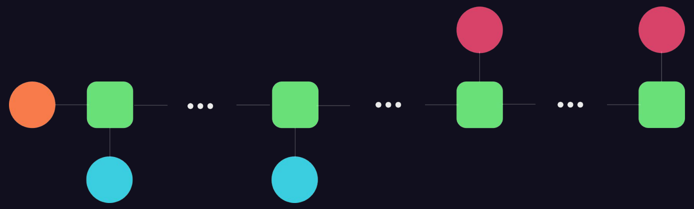
Sentiment Classification



ARCHITECTURES

Many to Many

Name Entity Recognition



ARCHITECTURES

Many to Many

Machine Translation

DE VINCI INNOVATION CENTER

|03 RECURRENT

ADVANTAGES

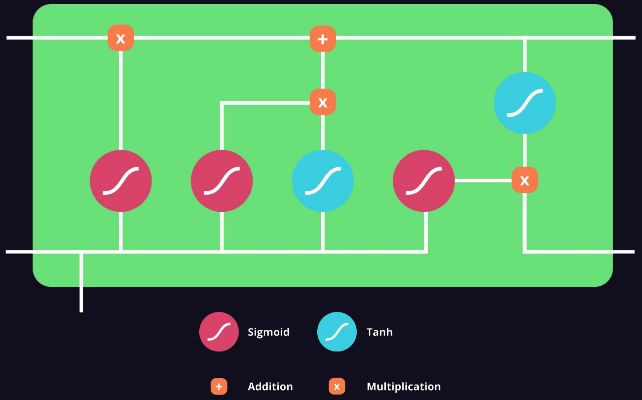
Infinite Input Length
Model **Size Invariant**
Historical Information
Weight Sharing Through Time

DRAWBACKS

Computationally **Slow**
Long Time **Dependency Lost** Over Time
Future Input not Considered
Vanishing/Exploding Gradient

DE VINCI INNOVATION CENTER

|03 RECURRENT



LSTM

Gates

Forget Gate
Update Gate
Output Gate

I/O

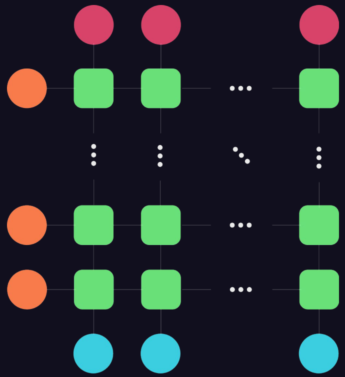
Previous Input
Cell State
Output State

Still **Suffers** from **Exploding Gradient**

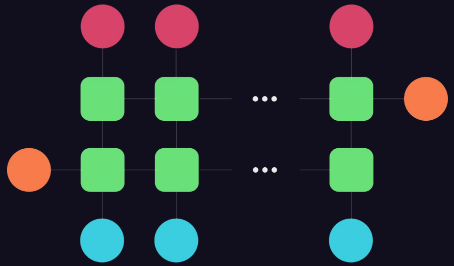
DE VINCI INNOVATION CENTER

|03 RECURRENT

STACKED



BIDIRECTIONAL



DE VINCI
INNOVATION
CENTER

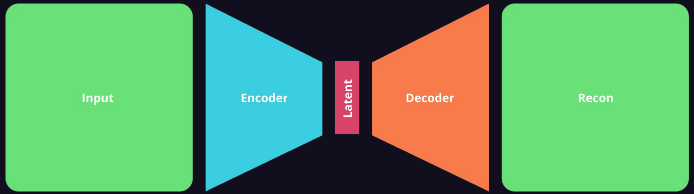
|04

AUTO-ENCODER

Hierarchical Compression is Key

DE VINCI
INNOVATION
CENTER

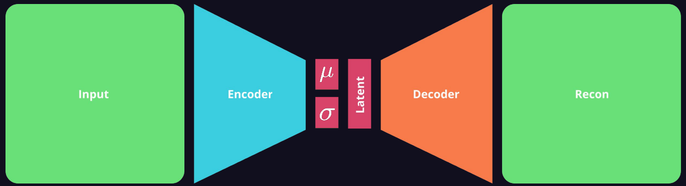
AUTO-ENCODER



$$z = e(x) \qquad \hat{y} = d(z)$$

$$loss = \frac{1}{N} \sum_i^N (\hat{y}_i - y_i)^2$$

VARIATIONAL AUTO-ENCODER



$$< \mu, \sigma >= e(x)$$

$$z = \mu \cdot \epsilon + \sigma$$

$$\hat{y} = d(z)$$

$$\epsilon \sim \mathcal{N}(0,1)$$

$$z \sim \mathcal{N}(\mu, \sigma)$$

$$loss = \frac{1}{N}\sum_i^N (\hat{y}_i - y_i)^2 + KL(\mathcal{N}(\mu_i, \sigma_i)||\mathcal{N}(0,1))$$

DE VINCI INNOVATION CENTER

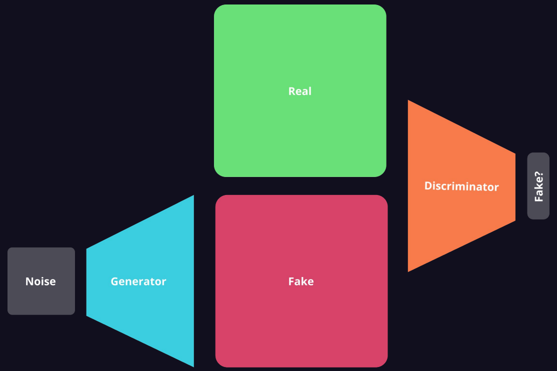
05|

GENERATIVE ADVERSARIAL NETWORK

Min Max for the Win

DE VINCI INNOVATION CENTER

|05 GENERATIVE ADVERSARIAL NETWORK



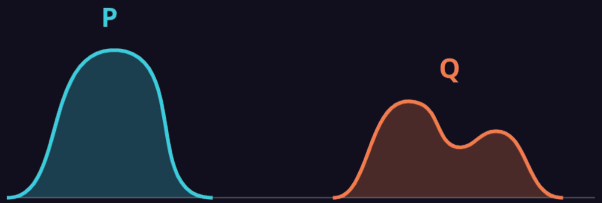
GENERATIVE ADVERSARIAL NETWORK

$$\underset{G}{min}\,\underset{D}{max} = \mathbb{E}_{x\sim p_r}[log(D(x))] + \mathbb{E}_{x\sim p_g}[1 - log(D(x))]$$

DE VINCI INNOVATION CENTER

|05 GENERATIVE ADVERSARIAL NETWORK

WASSERSTEIN



$$W_{(p_r,p_g)} = \underset{\gamma \sim \pi(p_r,p_g)}{inf} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$$

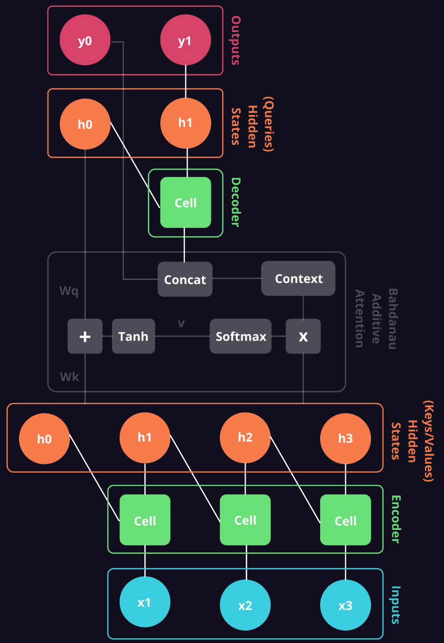
|06

ATTENTION

It is All You Need

DE VINCI
INNOVATION
CENTER

|06 ATTENTION



BAHDANAU ATTENTION

$$score(k,q) = V^T tanh(W_k k + W_q q)$$

$$attention(k,q) = softmax(score(k,q))$$

$$context = v \cdot attention(k,q)$$

DE VINCI INNOVATION CENTER

SELF ATTENTION