

# DEEP LEARNING

A Modern Approach to  
Artificial Intelligence

Yliess **HATI**

**PHD Student** - Computer Science

[yliess.hati@devinci.fr](mailto:yliess.hati@devinci.fr)



# |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

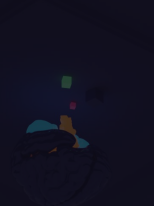
Hinton

## 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

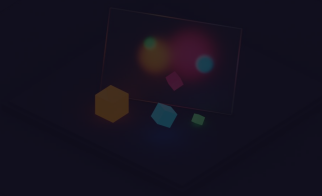


# 01|

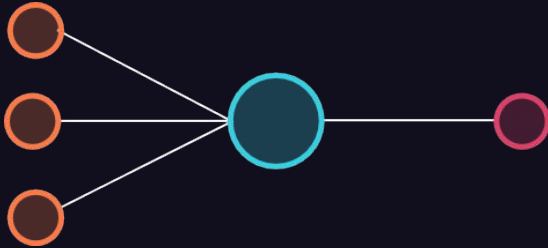
## PERCEPTRON

The Beginning and the End

# |01 PERCEPTRON

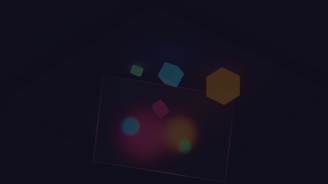


## PERCEPTRON

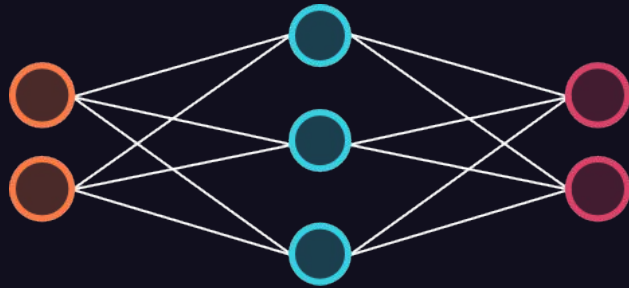
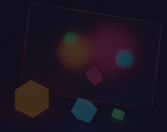


$$\hat{y} = f(wx + b)$$

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



# |01 PERCEPTRON

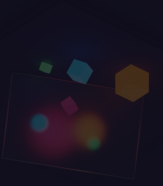


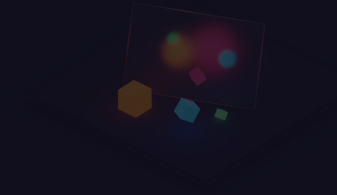
## MULTILAYER PERCEPTRON

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

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

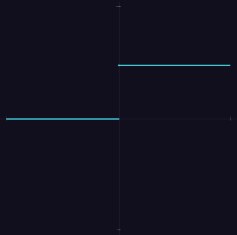
$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$





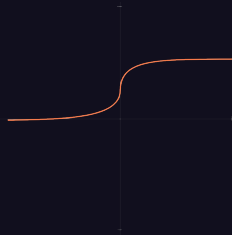
## ACTIVATION FUNCTIONS

Step



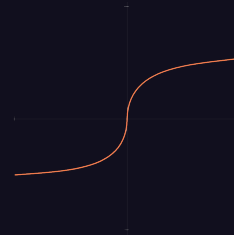
$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

Sigmoid



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

Tanh

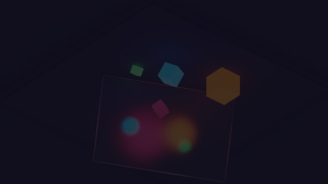


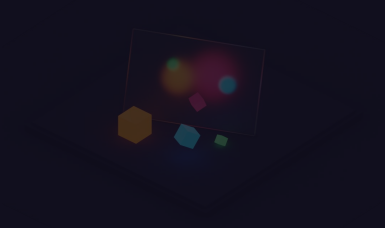
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

ReLU



$$\text{relu}(x) = \max(0, x) = x^+$$

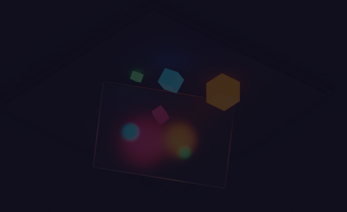




## ACTIVATION FUNCTIONS

**Softmax**

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

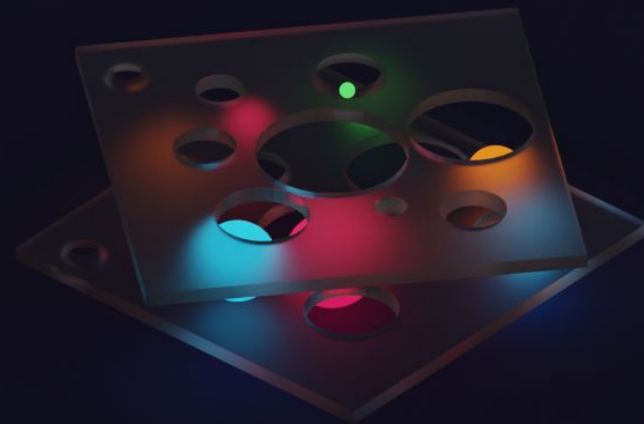




# |02

## CONVOLUTION

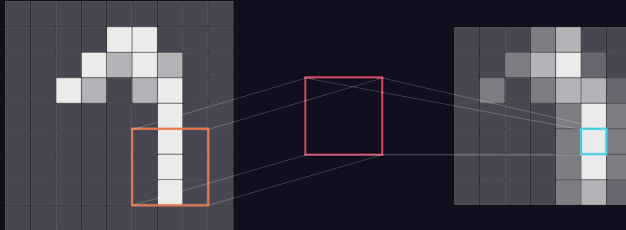
Signal Processing 101



## |02 CONVOLUTION

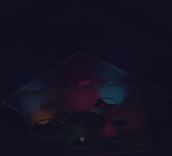


### CONVOLUTION CROSS CORRELATION



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

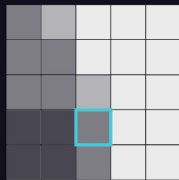
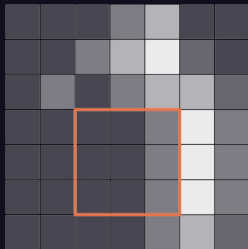
**Weight Sharing**



## |02 CONVOLUTION



### POOLING

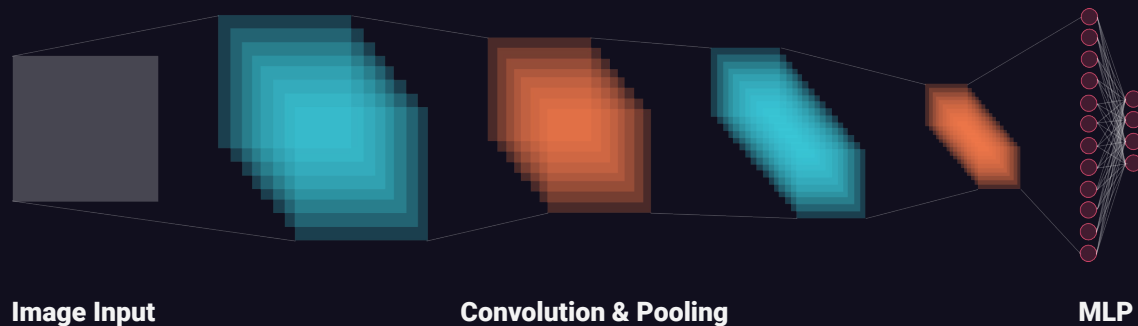


Dimensionality Reduction



## |02 CONVOLUTION

### CONVOLUTIONAL NEURAL NETWORK



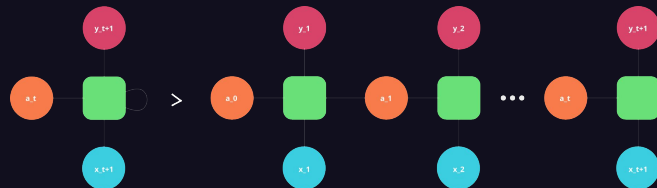


# 03|

## **RECURRENT**

Backprop Through Time

## |03 RECURRENT



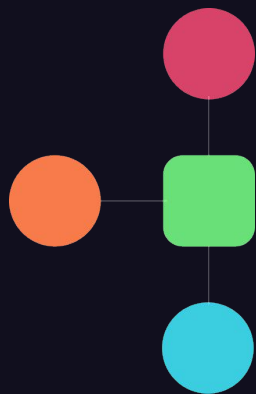
## RECURRENT CELLS

Weight Sharing & Backprop Through Time

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

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

## |03 RECURRENT

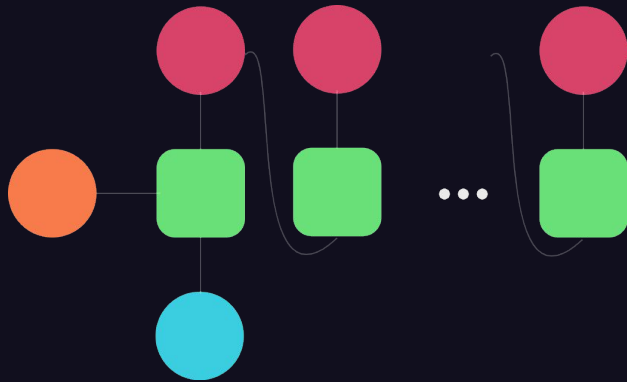


## ARCHITECTURES

**One to One**

Traditional Neural Network

## |03 RECURRENT



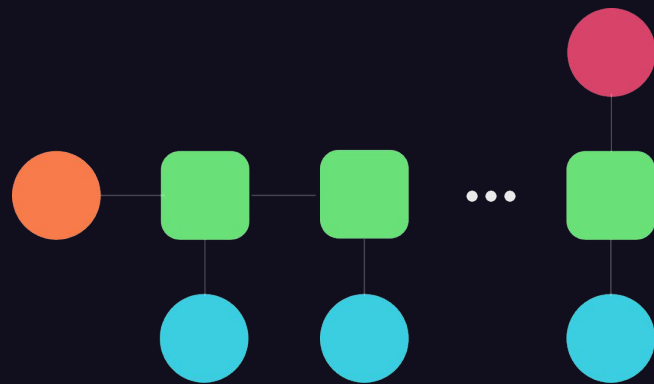
## ARCHITECTURES

**One to Many**

Music Generation



## |03 RECURRENT

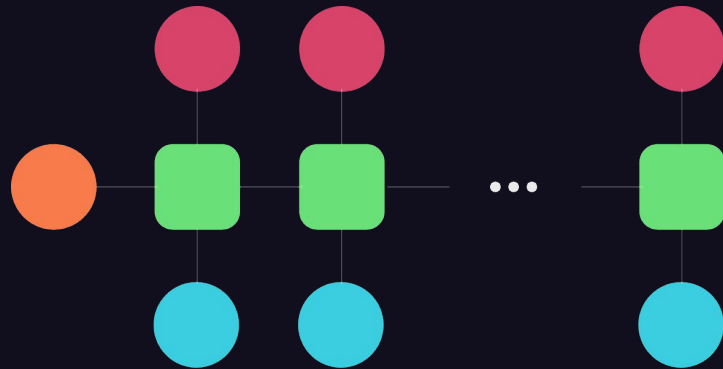


## ARCHITECTURES

**Many to One**

Sentiment Classification

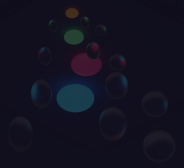
## |03 RECURRENT



## ARCHITECTURES

**Many to Many**

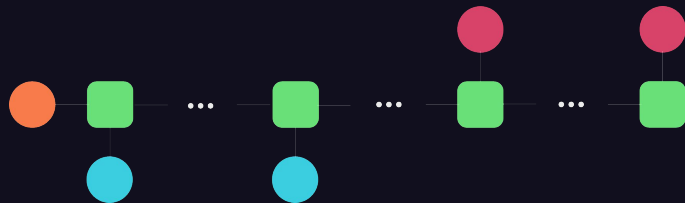
Name Entity Recognition



## |03 RECURRENT



### ARCHITECTURES



**Many to Many**

Machine Translation

## |03 RECURRENT



### ADVANTAGES

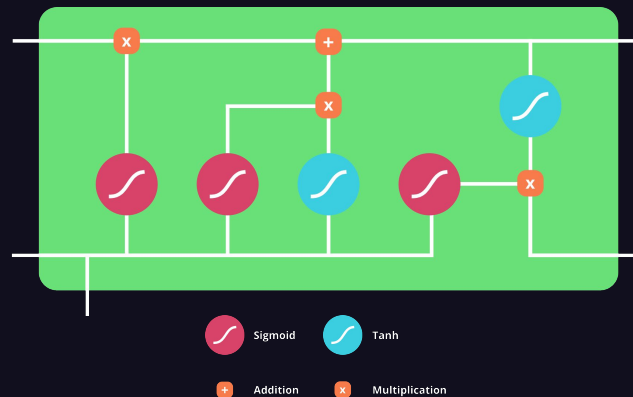
**Infinite** Input Length  
Model **Size Invariant**  
**Historical** Information  
**Weight Sharing** Through Time

### DRAWBACKS

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



## |03 RECURRENT



## LSTM

### Gates

**Forget** Gate  
**Update** Gate  
**Output** Gate

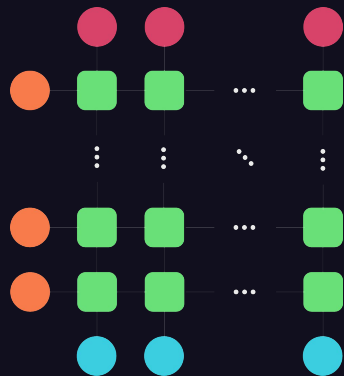
### I/O

**Previous** Input  
**Cell State**  
**Output** State

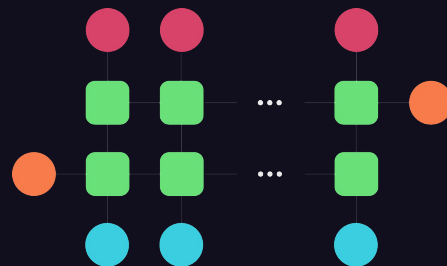
Still **Suffers** from **Exploding Gradient**

## |03 RECURRENT

STACKED



BIDIRECTIONAL



# |04

## AUTO-ENCODER

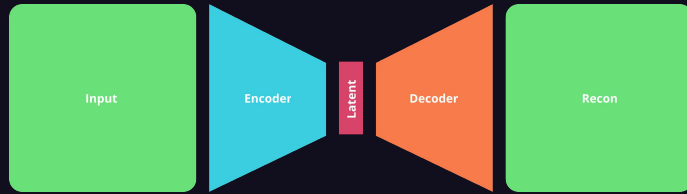
Hierarchical Compression is Key



## |04 AUTO-ENCODER



### AUTO-ENCODER



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

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

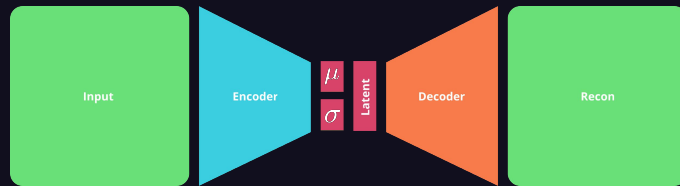




## |04 AUTO-ENCODER



### VARIATIONAL AUTO-ENCODER



$$\langle \mu, \sigma \rangle = e(x)$$

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

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

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

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

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



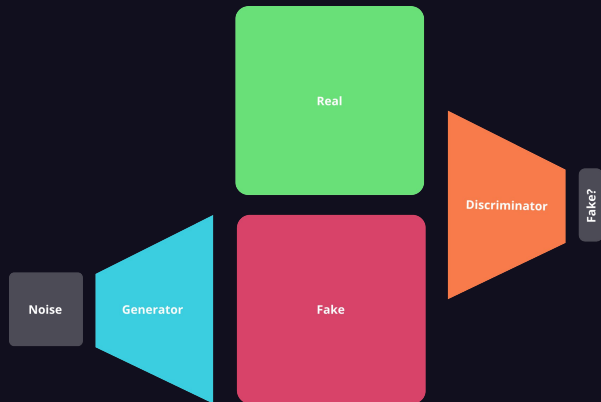


05|

# GENERATIVE ADVERSARIAL NETWORK

Find the Bandit

# |05 GENERATIVE ADVERSARIAL NETWORK



## GENERATIVE ADVERSARIAL NETWORK

$$fake = G(z)$$

$$loss_G = BCE(D(fake), 1)$$

$$loss_D = \frac{BCE(D(real), 1) + BCE(D(fake), 0)}{2}$$



# |06

## ATTENTION

It is All You Need