



IPTC  
Information Processing and  
Telecommunications Center

# IPTC Seminar Introduction to Deep Learning and Keras

July 15, 16 and 17, 2019  
ETSIT UPM, Madrid



Information Processing and  
Telecommunications Center



## Introduction to Deep Learning & Keras Day 1

**iptc**

Information Processing and Telecommunications Center

Prof. Luis A. Hernández Gómez  
[luisalfonso.hernandez@upm.es](mailto:luisalfonso.hernandez@upm.es)

E.T.S. Ingenieros de Telecomunicación  
Universidad Politécnica de Madrid

# IPTC Information Processing and Telecommunications Center

<https://iptc.upm.es/>



Information Processing and  
Telecommunications Center



## Master Program

We are working on the following applied research areas.

<b>Mobility and Transport</b> Air, surface, multimodal	<b>Interaction &amp; Interfaces</b> Augmented, virtual, natural and smart
<b>Future Telecoms</b> Signals, media, networks and security	<b>Connected Industry</b> Data-driven, cognitive computation, Big Data, IoT
<b>Data Engineering &amp; Digital Transformation</b> Target sector orientation, data exploitation, advanced modelling	<b>Remote Sensing &amp; Space</b> Devices, algorithms, systems and applications
<b>Smart Cities</b> Spaces, resources, citizens	<b>Health and Wellbeing support</b> Independent ageing, improved health care and p-health services

Fundamentals

Signal Processing

Machine Learning

Applications and Practice

### SIGNAL PROCESSING AND MACHINE LEARNING FOR BIG DATA

SEMESTER 1			SEMESTER 2		
Statistical Modelling (3C)	Time Series Analysis (4.5C)	Optimization Fundamentals (3C)	Signal Processing for Big Data (4C)	Big Data for Image and Video Signals (4C)	Bio-inspired learning (3C)
Optimization techniques for big data analysis (3C)	Predictive and Descriptive Learning (6C)	Machine Learning Lab (4.5C)	Reinforcement learning (3C)	Application Projects (4C)	Large-scale Media Analytics (4C)
Data Science Foundations and Applications (2C)					Masters' Thesis (12C)

<http://www.mstc.ssr.upm.es/big-data-track>



IPTC  
Information Processing and  
Telecommunications Center

Day 1 (July 15) : Intro to Deep Learning & Keras.  
Feed Forward models

Day 2 (July 16) : Backpropagation. Convolutional Networks  
Transfer Learning. Data Augmentation.

Day 3 (July 17) : Recurrent Networks. Advanced Architectures.  
GANs. Applications. Final discussion on AI

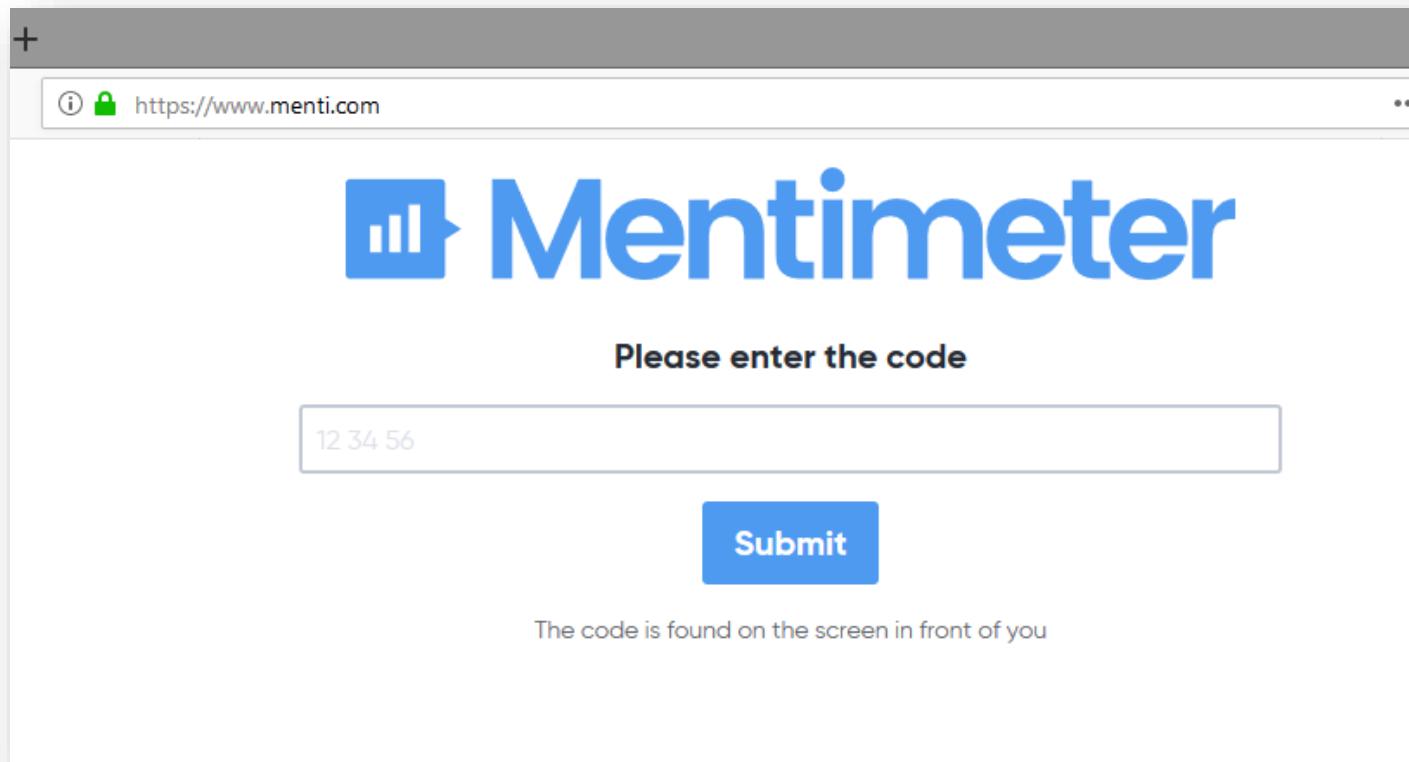
IPTC Seminar  
Introduction to Deep Learning and Keras

July 15, 16 and 17, 2019  
ETSIT UPM, Madrid

# To have some feedback... we will use:

<https://www.menti.com/>

CODE: .....



# IPTC Summer Seminar MATERIALS

Please go to our MSTC GitHub:

[https://github.com/MasterMSTC/IPTC\\_DeepLearning](https://github.com/MasterMSTC/IPTC_DeepLearning)

The screenshot shows the GitHub repository page for 'MasterMSTC / IPTC\_DeepLearning'. The repository has 26 commits, 1 branch, 0 releases, and 1 contributor. The latest commit was made 11 minutes ago. The repository contains notebooks, presentations, and README files.

Materials for IPTC Summer Seminar: Practical Introduction to Deep Learning & Keras

Manage topics

26 commits | 1 branch | 0 releases | 1 contributor

Branch: master ▾ | New pull request | Create new file | Upload files | Find File | Clone or download ▾

File	Commit Message	Time
MasterMSTC Delete IPTC_GoogleColab_Intro.ipynb	Latest commit f7a7cf3 11 minutes ago	
Notebooks	Delete readme.txt	11 minutes ago
PRESENTATIONS	Create readme.txt	8 hours ago
README.md	Update README.md	8 hours ago
README.md		

# WiFi Connection



**User:** iptc

**Password:** n6gd44j8

# The Artificial Intelligence Renaissance:

*deep learning and the road to  
human-Level machine intelligence*

<https://www.cambridge.org/core/journals/apsipa-transactions-on-signal-and-information-processing/article/artificial-intelligence-renaissance-deep-learning-and-the-road-to-humanlevel-machine-intelligence/A82CA4909877C98B23755744A18EA64F>

# A Brief History



1958 Perceptron

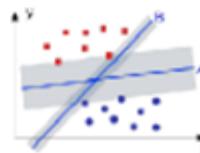
1974 Backpropagation



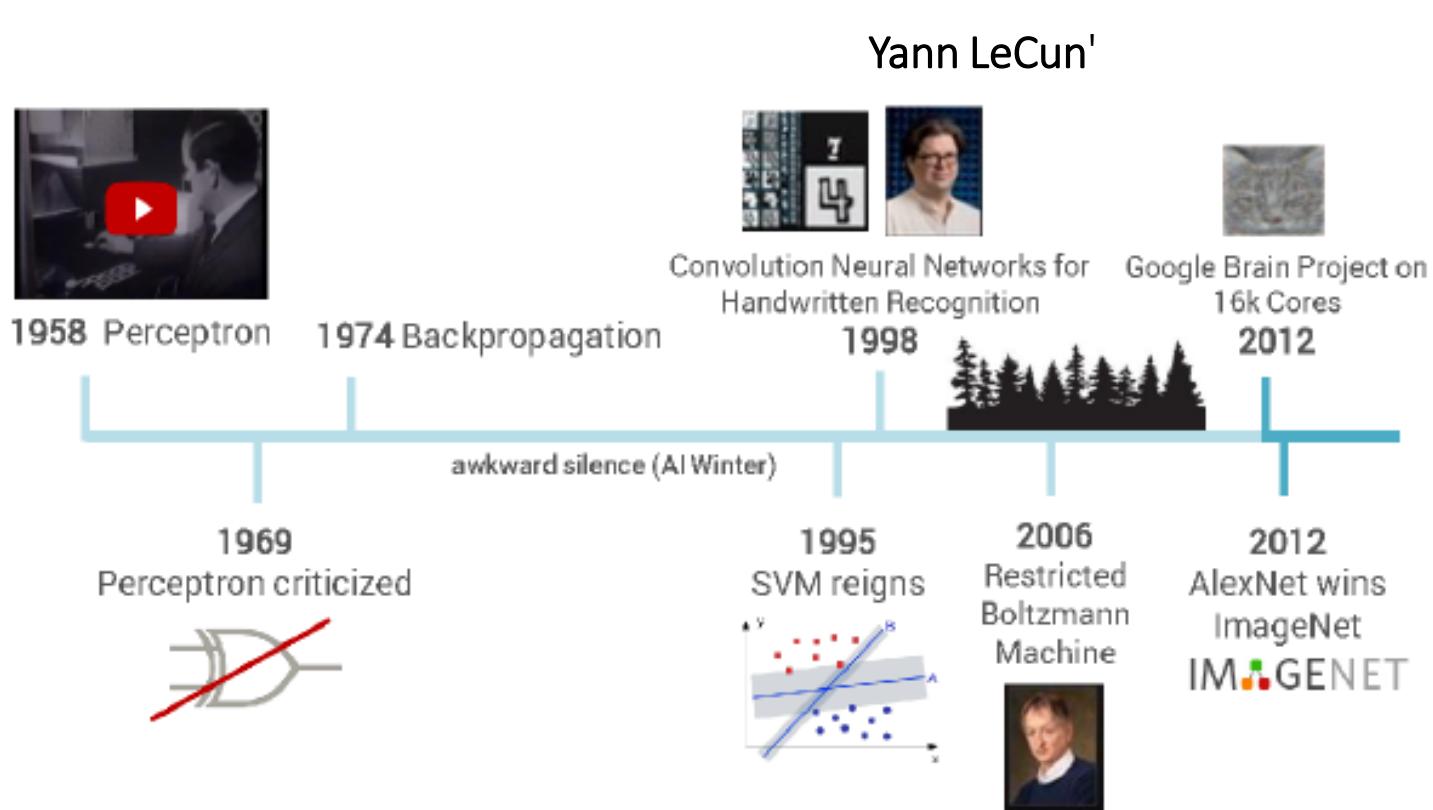
1969  
Perceptron criticized



1995  
SVM reigns



# A Brief History

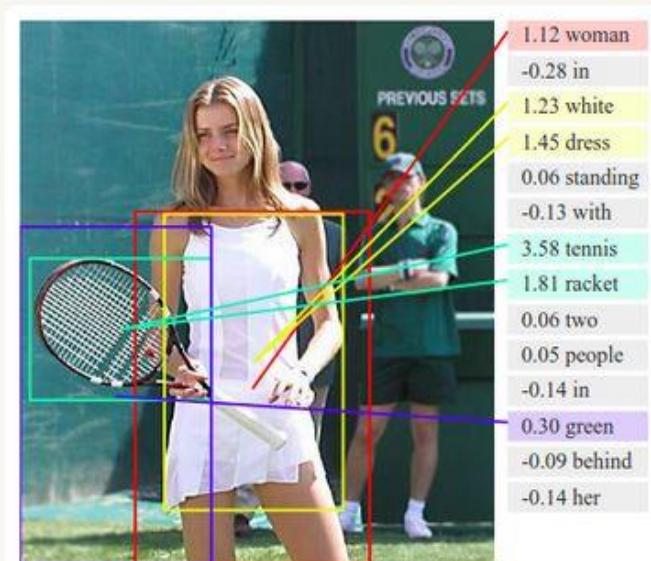


# Deep Learning: Hype or Reality?

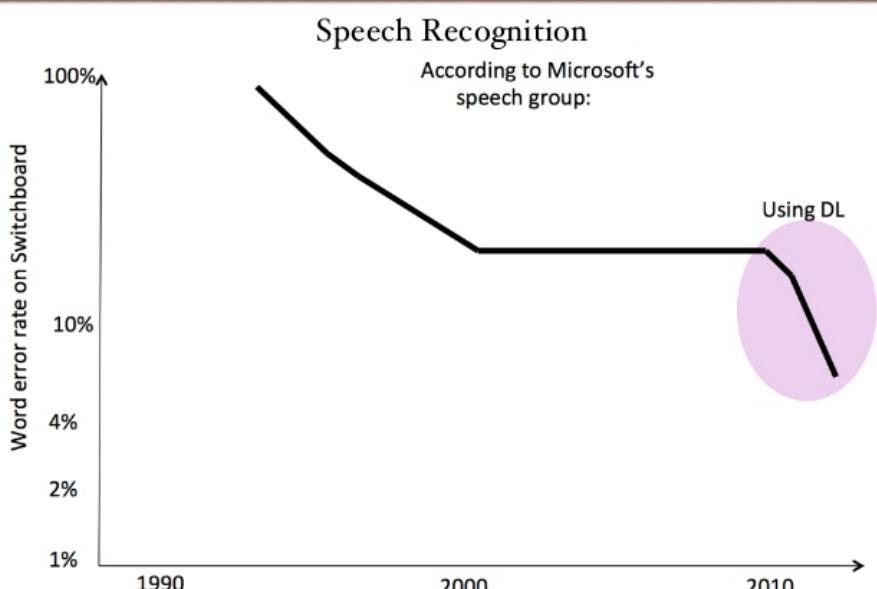
DNNs better than humans at image recognition



ImageNet  
1000 categories  
1.3M images  
Human error: 5%  
DNN: 3%



# Deep Learning: Hype or Reality?



Pedro Moreno



Super-human speech recognition

# Deep Learning: Hype or Reality?

# Just with in Google

- Search
  - Search by image
  - Driveless cars
  - Youtube recommendation
    - Videos
    - Thumbnails
  - Maps
  - Reading street addresses



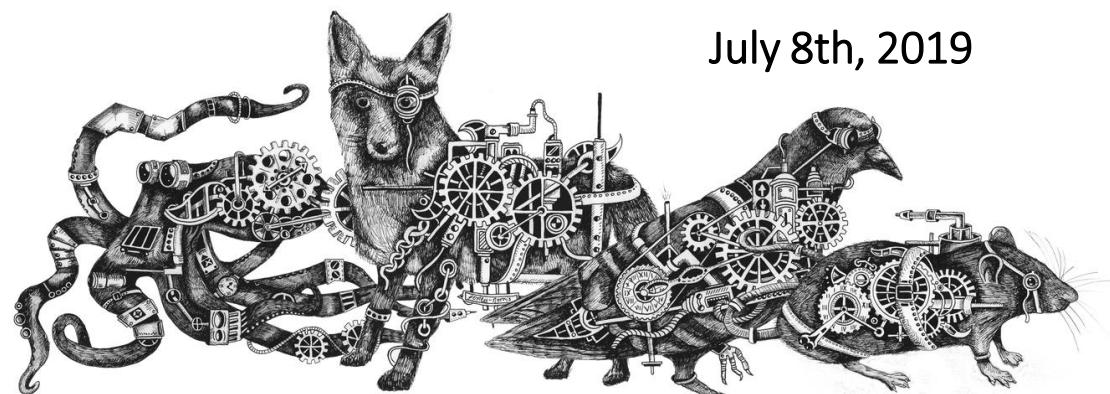
# Deep Reinforcement Learning



## Humanoid Robots Conversational AI : Social Bots



Animal-AI Olympics   Introduction   News   Timeline   Sponsors   Media   Organisers   Rules   Resources and Links ▾



July 8th, 2019

Image Credit Squidoodle

<https://github.com/beyretb/AnimalAI-Olympics>

 Andrew Ng 

Inicio Momentos Buscar en Twitter

 **Andrew Ng**   
@AndrewYNg [Seguir](#) ▾

I'm glad DeepNude is dead. As a person and as a father, I thought this was one of the most disgusting applications of AI. To the AI Community: You have superpowers, and what you build matters. Please use your powers on worthy projects that move the world forward.

11:06 - 28 jun. 2019

---

1.984 Retweets 8.092 Me gusta 

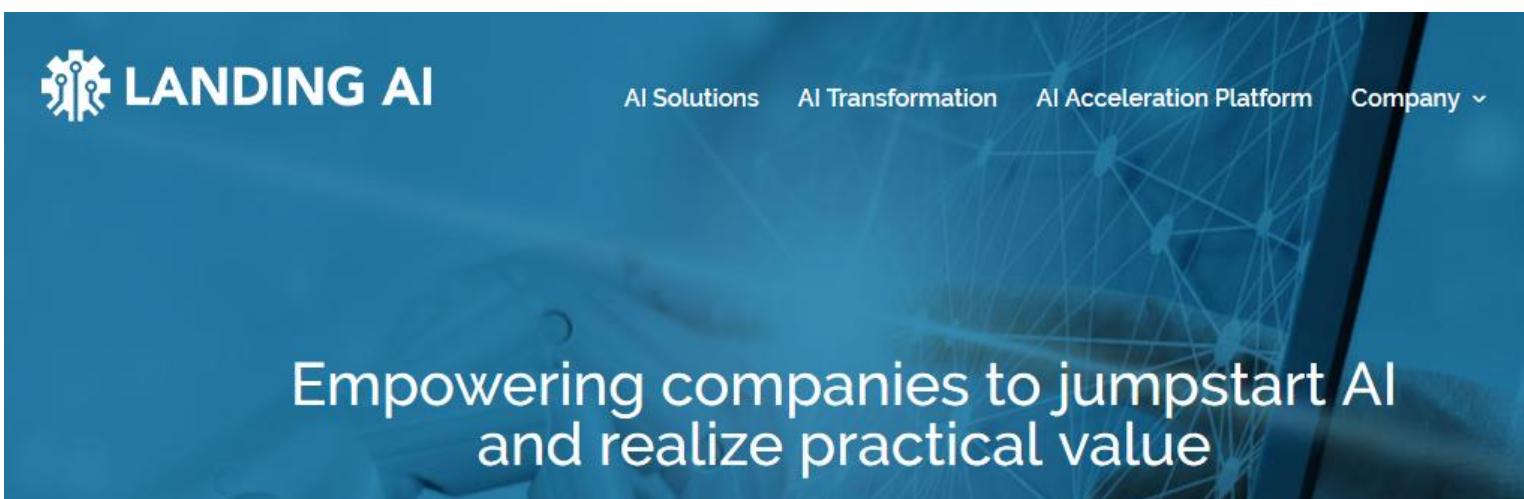
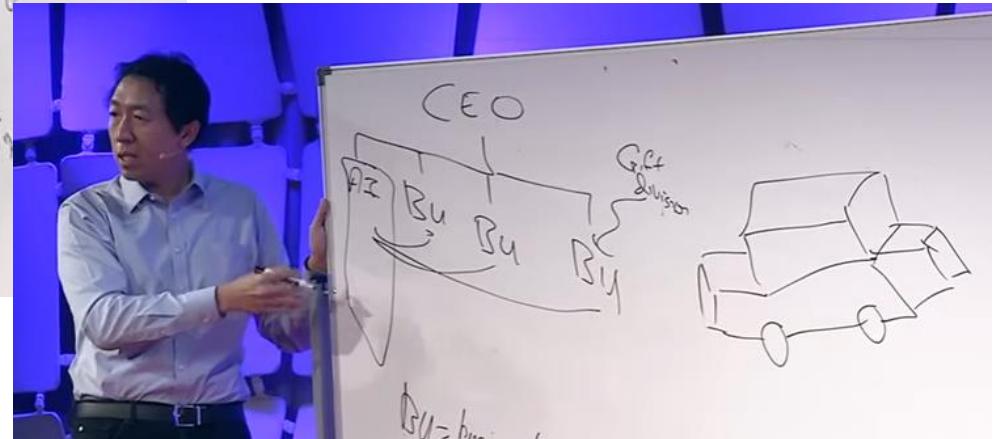
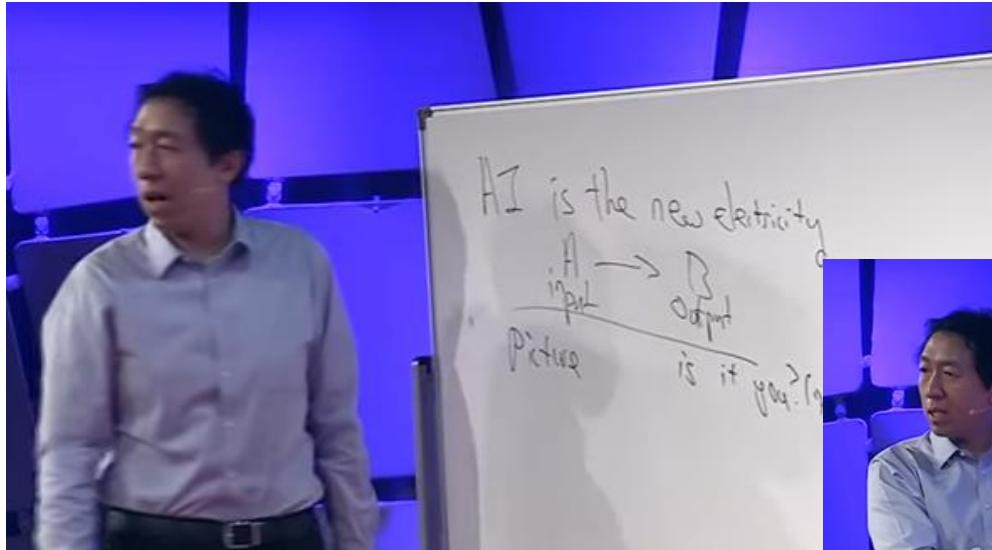
---

164 2,0K 8,1K

 **Muriz** @MurgioMurmani · 28 jun.  
En respuesta a @AndrewYNg  
I feel like this is just the beginning of AI misuse

3 3 73

[https://www.youtube.com/watch?v=NKpuX\\_yzdYs](https://www.youtube.com/watch?v=NKpuX_yzdYs)



The screenshot shows the Landing AI website. The header features the company logo (a stylized figure), the text "LANDING AI", and navigation links for "AI Solutions", "AI Transformation", "AI Acceleration Platform", and "Company". The main tagline "Empowering companies to jumpstart AI and realize practical value" is displayed prominently in the center.

# Introduction to Deep Learning but what is new?

What Changed?  
Old wine in new bottles



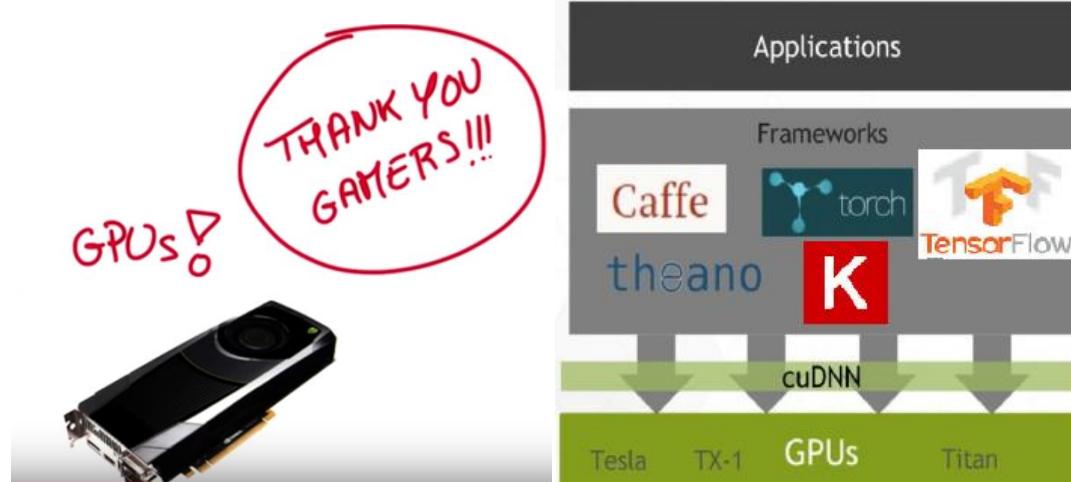
Big Data  
(Digitalization)



Computation  
(Moore's Law, GPUs)



Algorithmic  
Progress



FREE COURSE

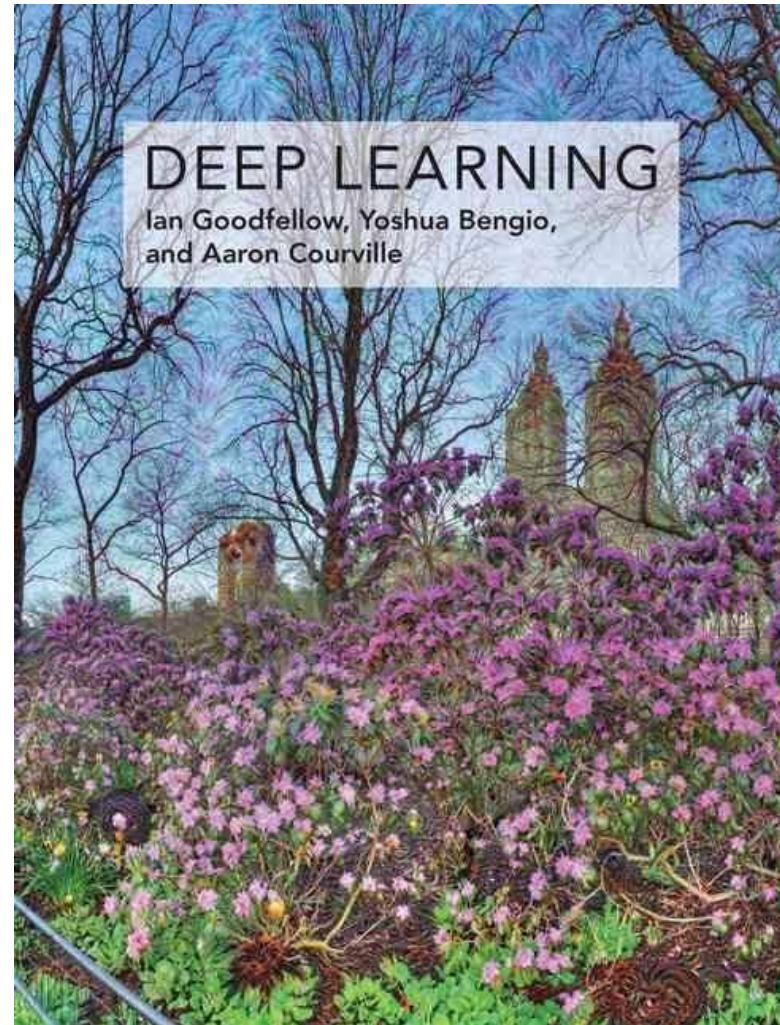
Deep Learning  
by Google

# Introduction to Deep Learning

## ...the Machine Learning background...

You should Learn the  
whole  
Machine Learning  
context !

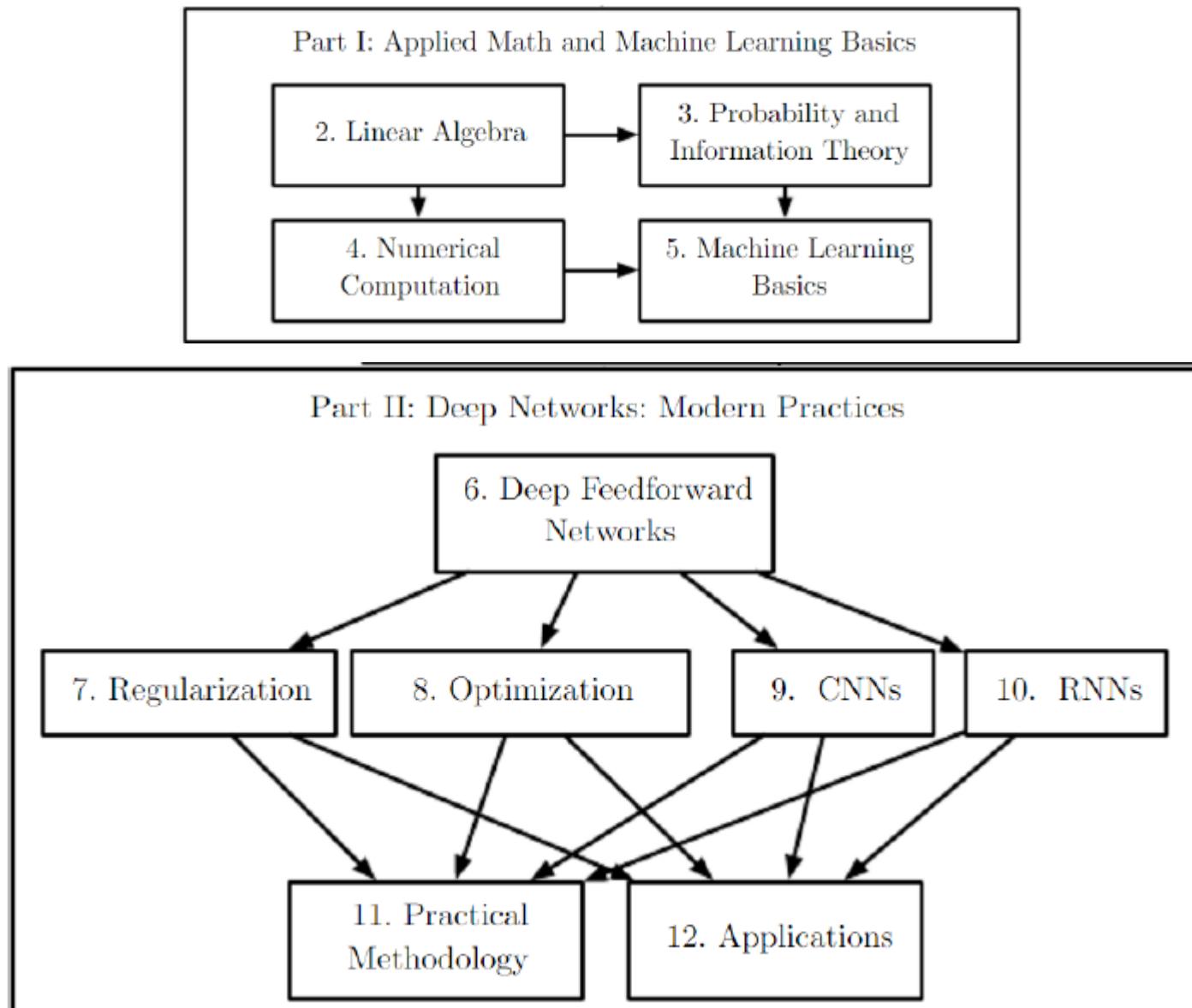
On line:  
[www.deeplearningbook.org](http://www.deeplearningbook.org)



Deep Learning courses  
Prof. Hung-yi Lee National Taiwan University (NTU) Taipei

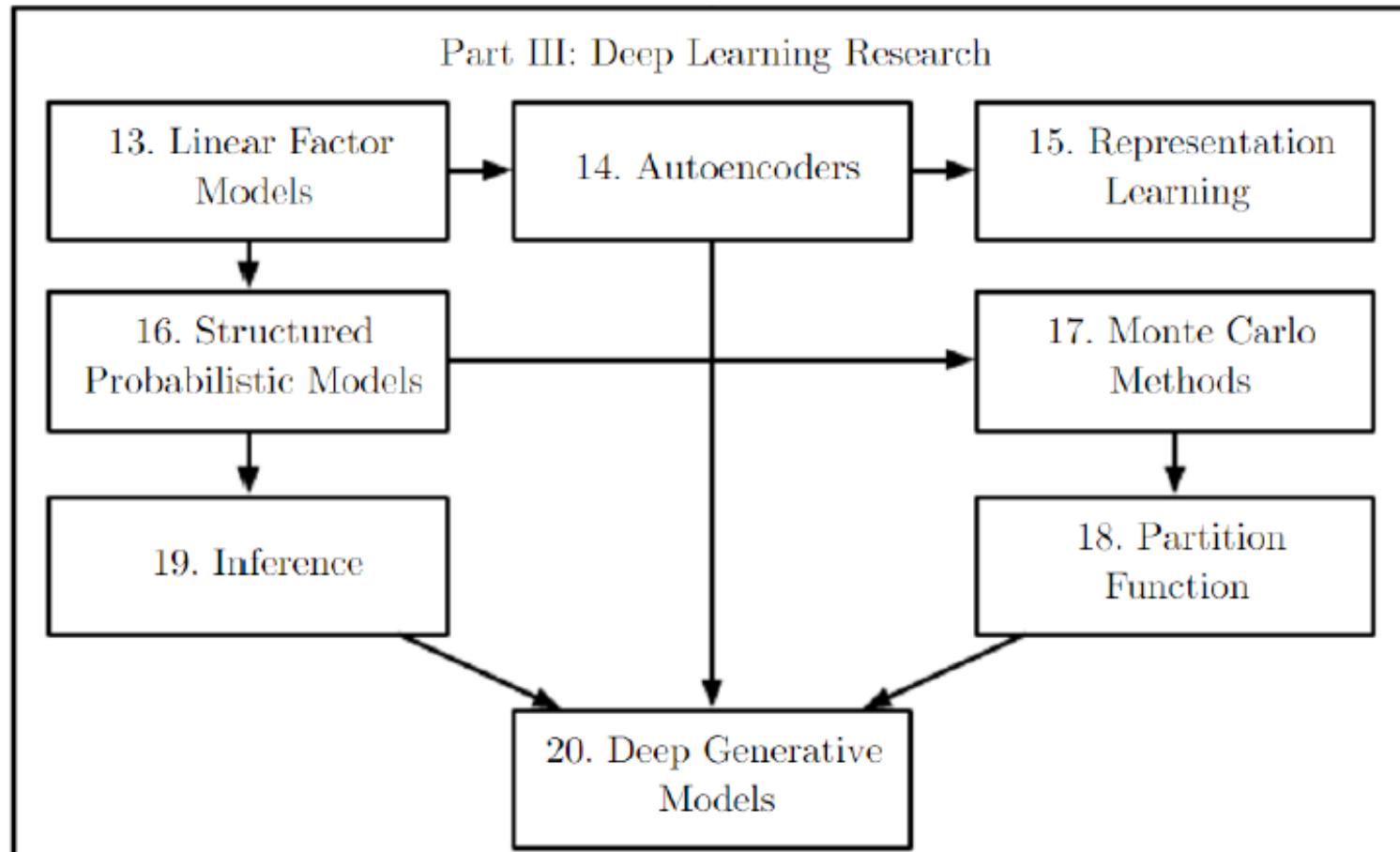
# Introduction to Deep Learning

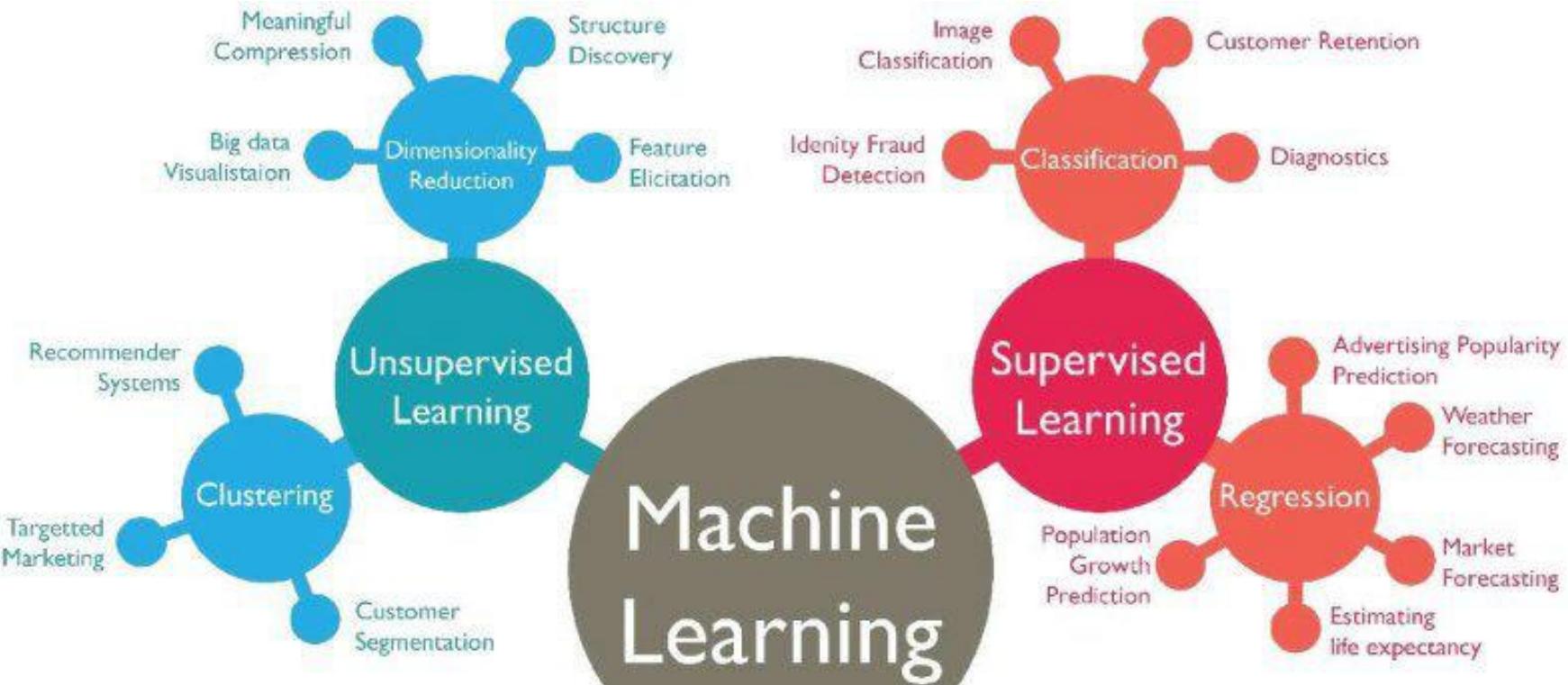
## ...the Machine Learning background...



# Introduction to Deep Learning

## ...the Machine Learning background...





[StepUpAnalytics.com](http://StepUpAnalytics.com)

# Supervised or unsupervised? Recommendation systems (I)

## The Netflix prize

- Competition started in October 2006. Training data is ratings for 18.000 movies by 400.000 Netflix customers, each rating between 1 and 5.
- Training data is very sparse — about 98% missing.
- Objective is to predict the rating for a set of 1 million customer-movie pairs that are missing in the training data.
- Netflix's original algorithm achieved a root MSE of 0.953.
- The first team to achieve a 10% improvement wins one million dollars.
- Is this a supervised or unsupervised problem?

# Supervised or unsupervised? Recommendation systems (II)

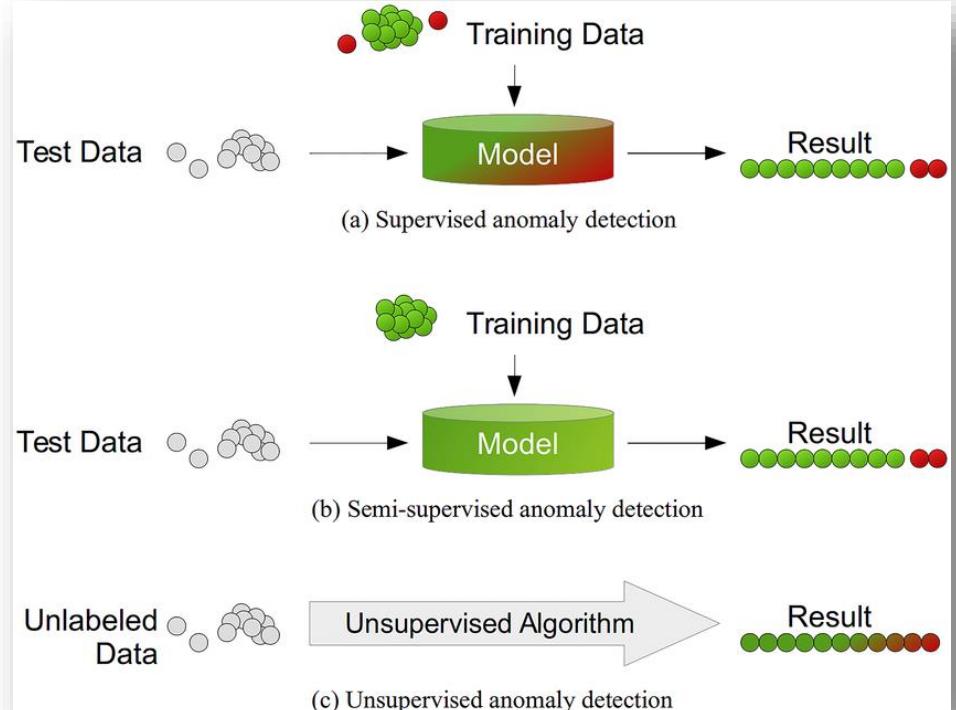
The screenshot shows the Netflix Prize Leaderboard page. At the top, there's a yellow banner with the text "Netflix Prize" and a large red "COMPLETED" stamp. Below the banner, the page has a navigation bar with links for "Home", "Rules", "Leaderboard", and "Update". The main title "Leaderboard" is displayed prominently in blue. To the right of the title, there's a message "Showing Test Score. [Click here to show quiz score](#)" and a dropdown menu to "Display top 20 leaders". The table below lists the top 12 teams, their test scores, improvement percentages, and submission times.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
<b>Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos</b>				
1	<a href="#">BellKor's Pragmatic Chaos</a>	0.8567	10.06	2009-07-26 18:18:28
2	<a href="#">The Ensemble</a>	0.8567	10.06	2009-07-26 18:38:22
3	<a href="#">Grand Prize Team</a>	0.8582	9.90	2009-07-10 21:24:40
4	<a href="#">Opera Solutions and Vandelay United</a>	0.8588	9.84	2009-07-10 01:12:31
5	<a href="#">Vandelay Industries !</a>	0.8591	9.81	2009-07-10 00:32:20
6	<a href="#">PragmaticTheory</a>	0.8594	9.77	2009-06-24 12:06:56
7	<a href="#">BellKor in BigChaos</a>	0.8601	9.70	2009-05-13 08:14:09
8	<a href="#">Dace</a>	0.8612	9.59	2009-07-24 17:18:43
9	<a href="#">Feeds2</a>	0.8622	9.48	2009-07-12 13:11:51
10	<a href="#">BigChaos</a>	0.8623	9.47	2009-04-07 12:33:59
11	<a href="#">Opera Solutions</a>	0.8623	9.47	2009-07-24 00:34:07
12	<a href="#">BellKor</a>	0.8624	9.46	2009-07-26 17:19:11

BellKor's Pragmatic Chaos wins, beating The Ensemble by a narrow margin.

# Supervised or unsupervised?

- Fraud detection?
- Anomaly detection?



- Sound classification (FreeSound Challenge)

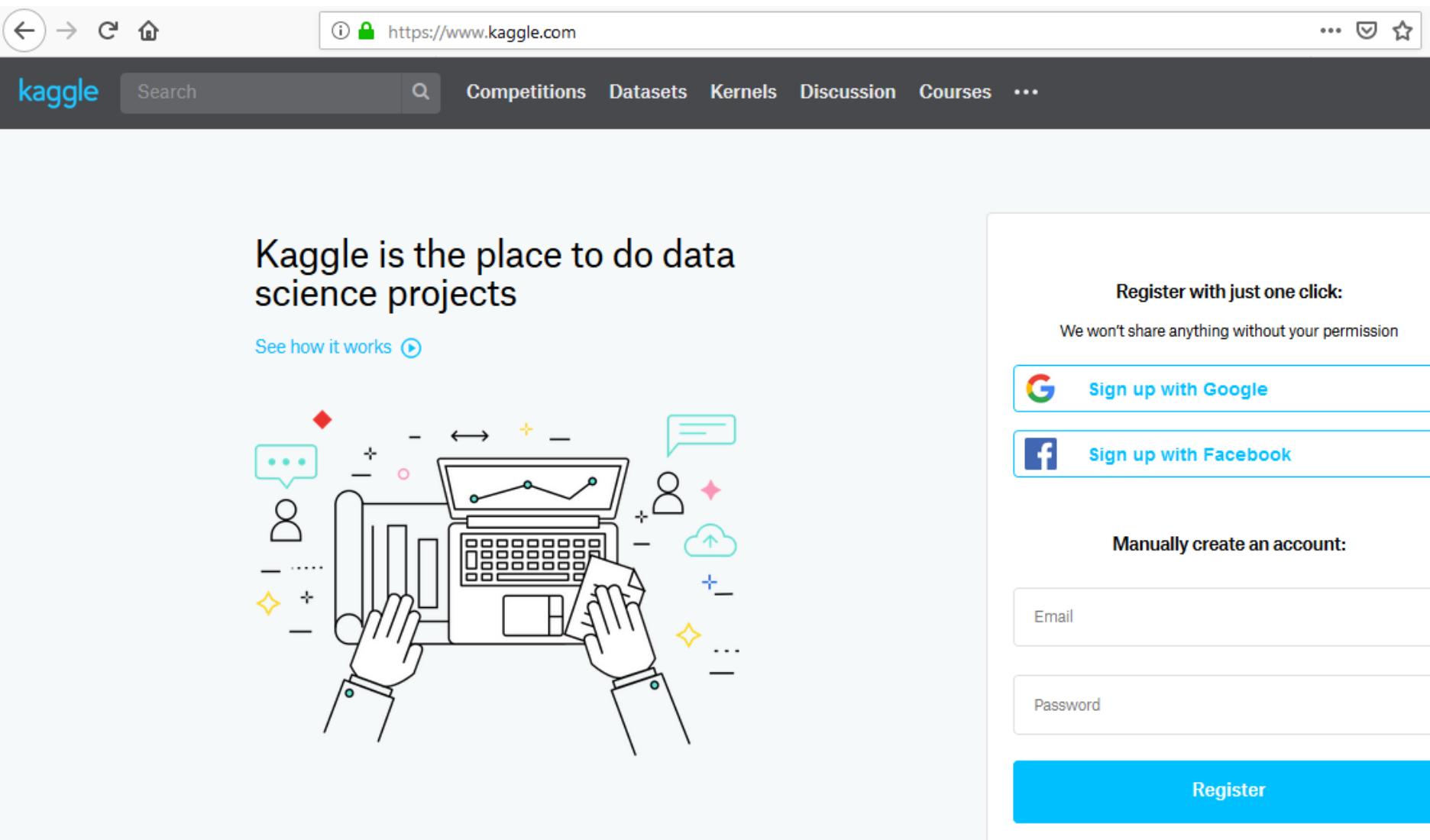
kaggle Search Competitions Datasets Kernels Discussion Learn ...

Research Prediction Competition

**Freesound General-Purpose Audio Tagging Challenge**

Can you automatically recognize sounds from a wide range of real-world environments?

558 teams · a year ago



# Artificial Intelligence

## Machine Learning

### Deep Learning

The subset of machine learning composed of algorithms that permit software to train itself to perform tasks, like speech and image recognition, by exposing multilayered neural networks to vast amounts of data.

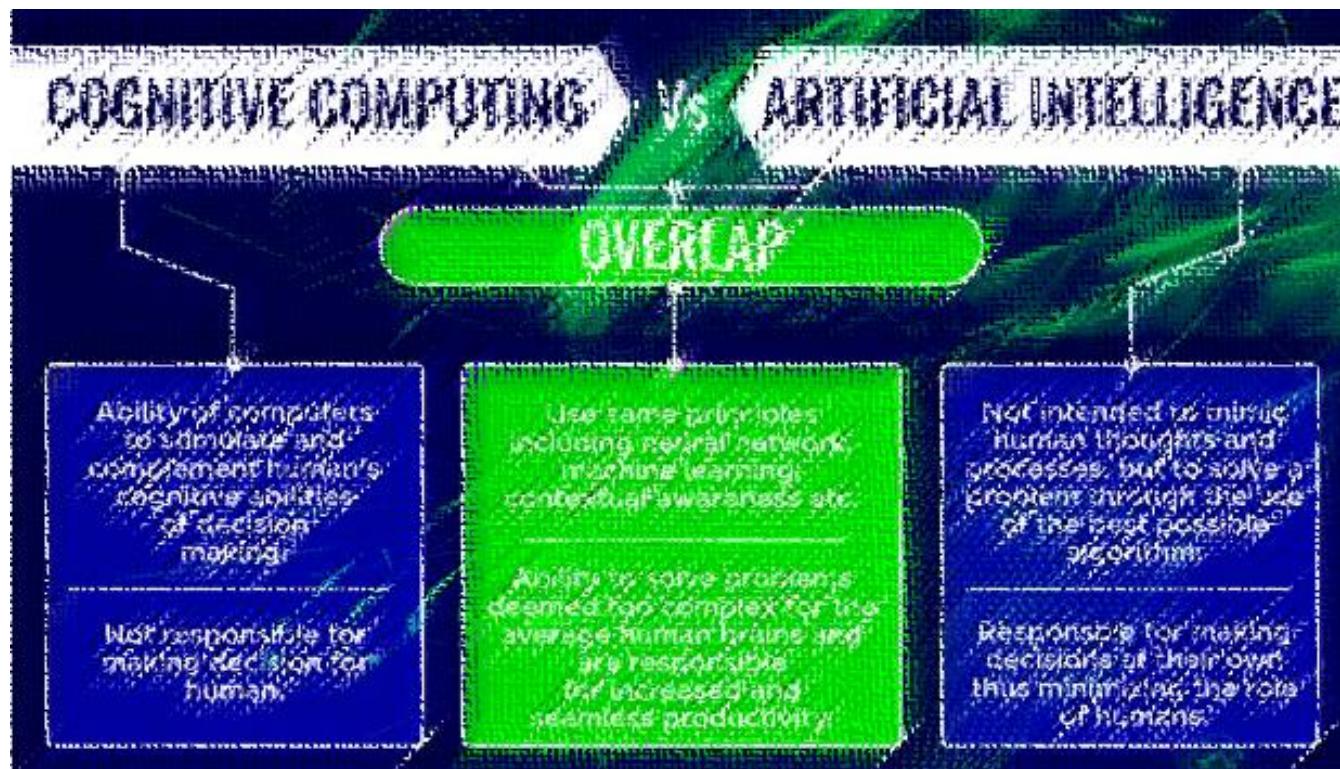
A subset of AI that includes abstruse statistical techniques that enable machines to improve at tasks with experience. The category includes deep learning

Any technique that enables computers to mimic human intelligence, using logic, if-then rules, decision trees, and machine learning (including deep learning)

# Research by your own on:

...we will discuss it by the end of Seminar (day-3)

- Cognitive Computing
- Artificial General Intelligence (AGI)





Deep Learning est mort. Vive  
Differentiable Programming!

**Yann LeCun**, Director of Facebook AI Research and the inventor of convolutional neural networks, took it.

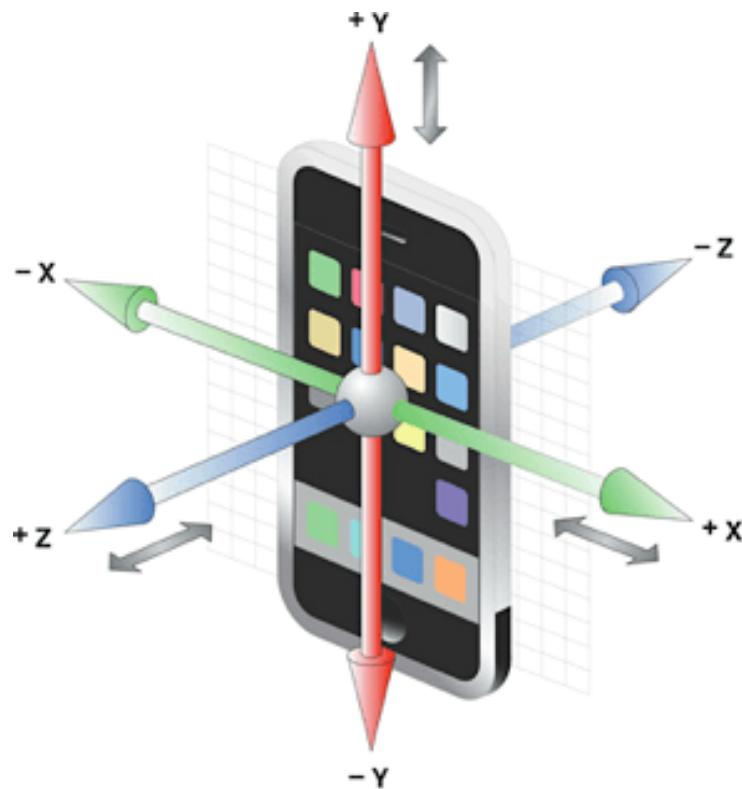
FROM : *humans* ...

- to *machines* ...
- and to Artificial Intelligence (AI)

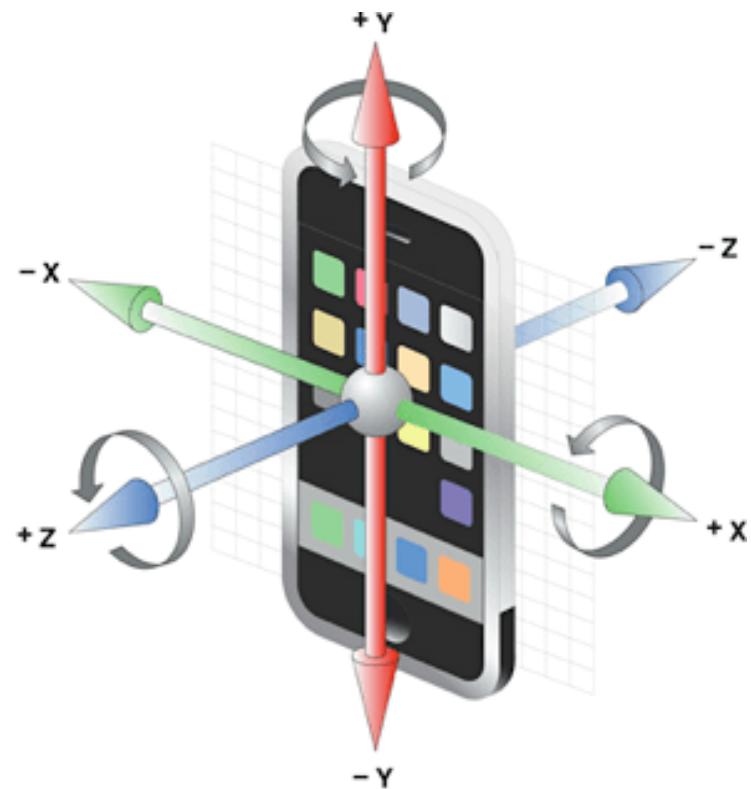
# Drivies use case:

*[www.driviesapp.com](http://www.driviesapp.com)*

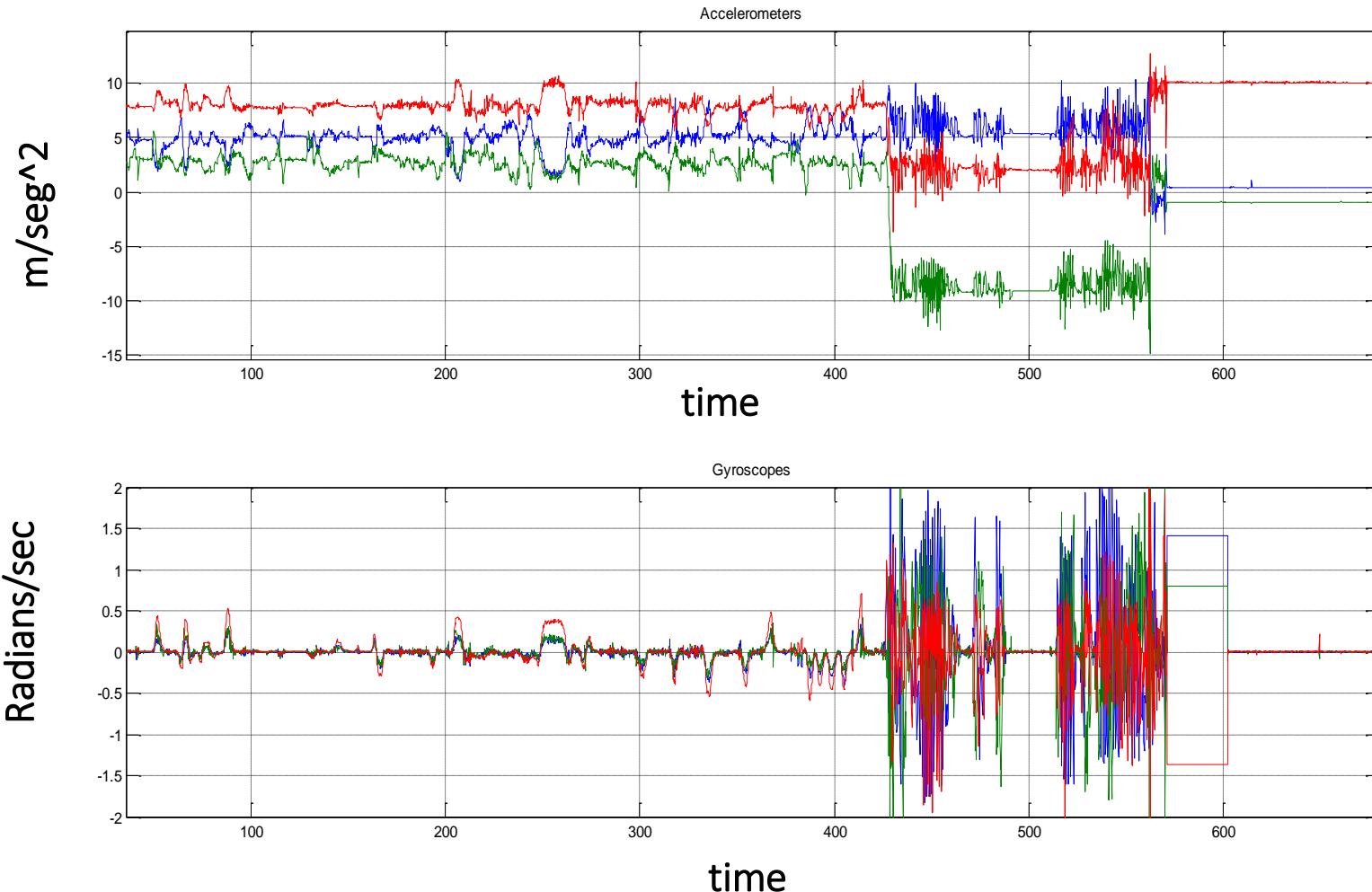




**Accelerometer**

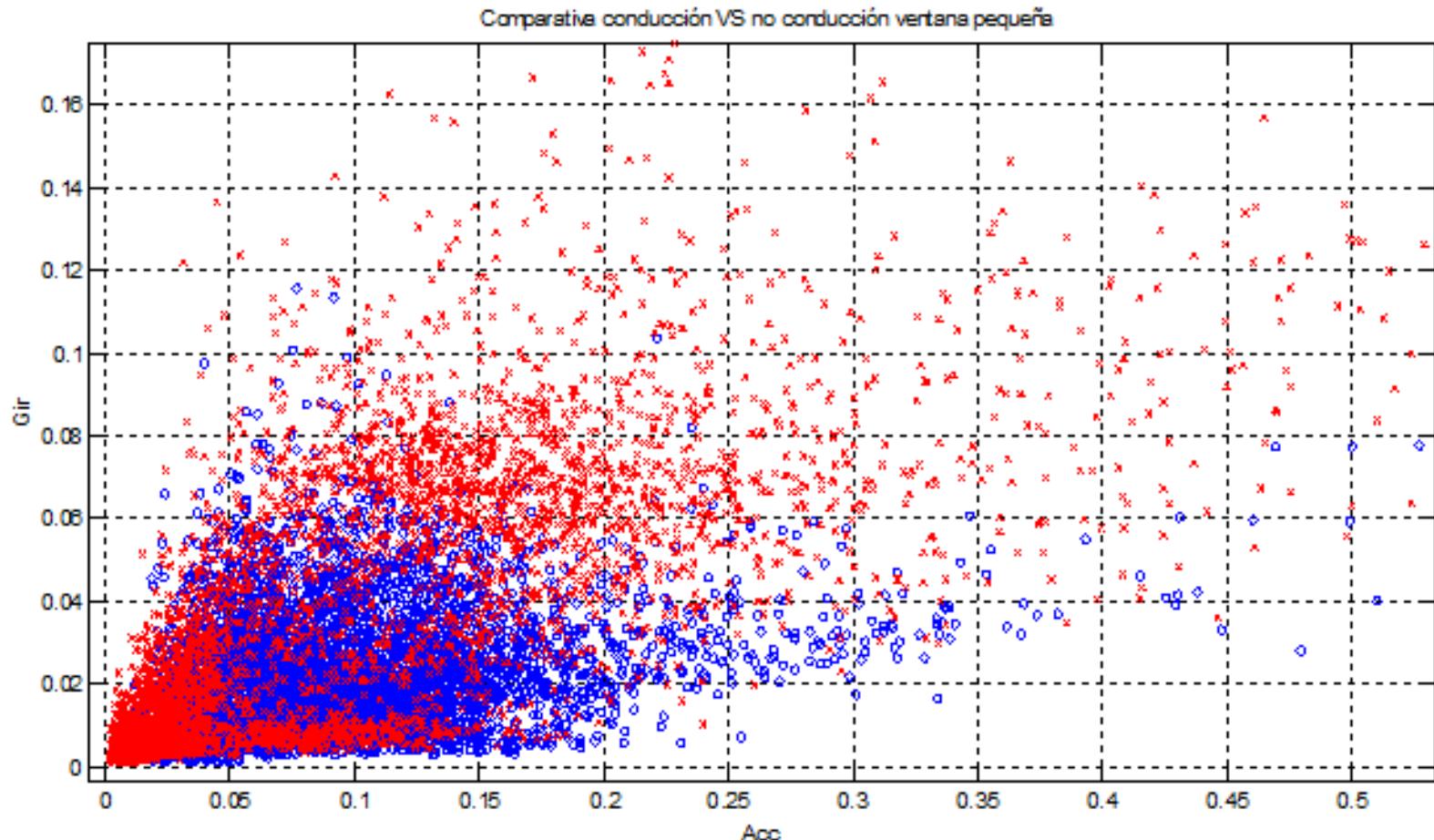


**Gyroscope**



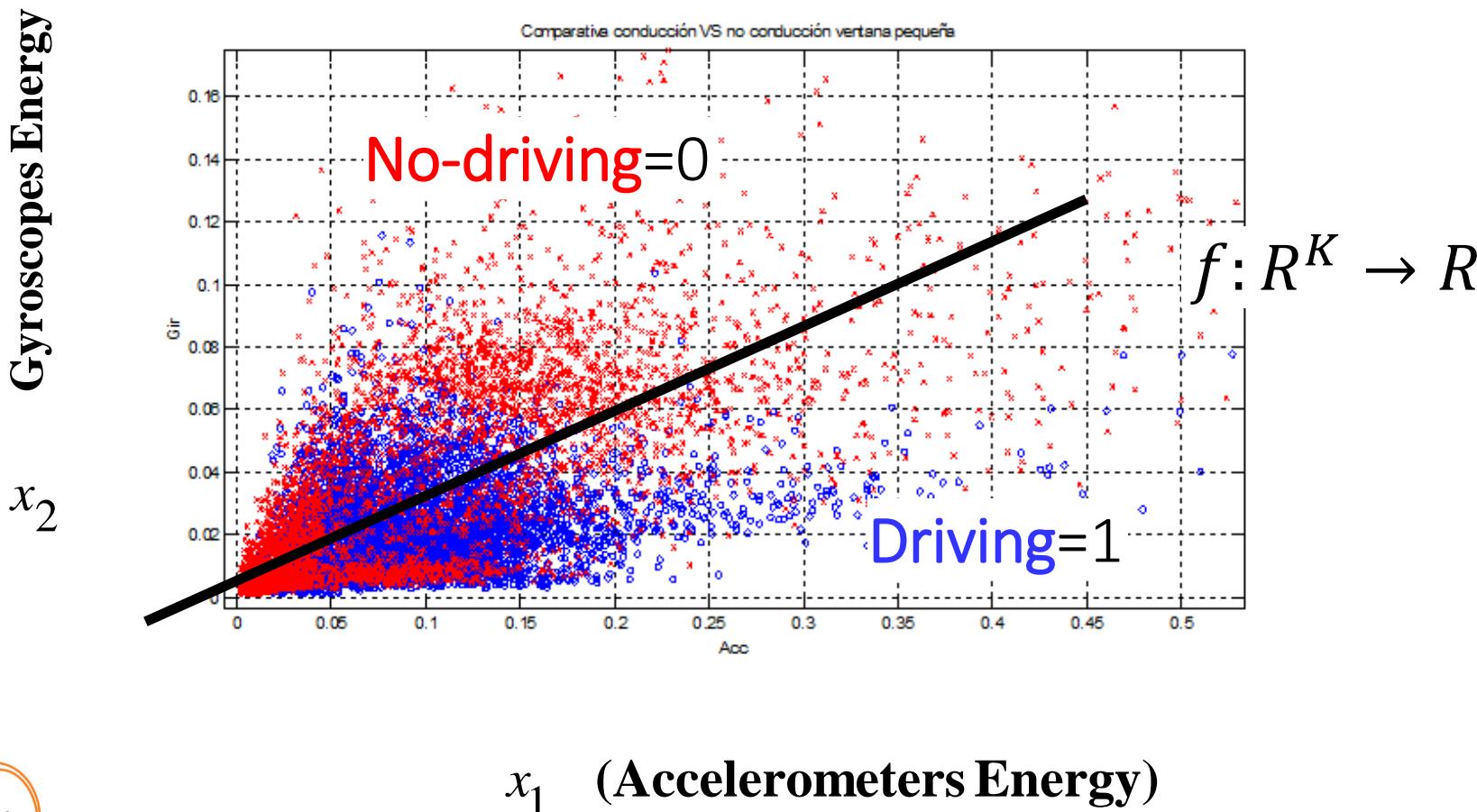
# From manual - linear classifiers TO Neural Networks

Gyroscopes Energy



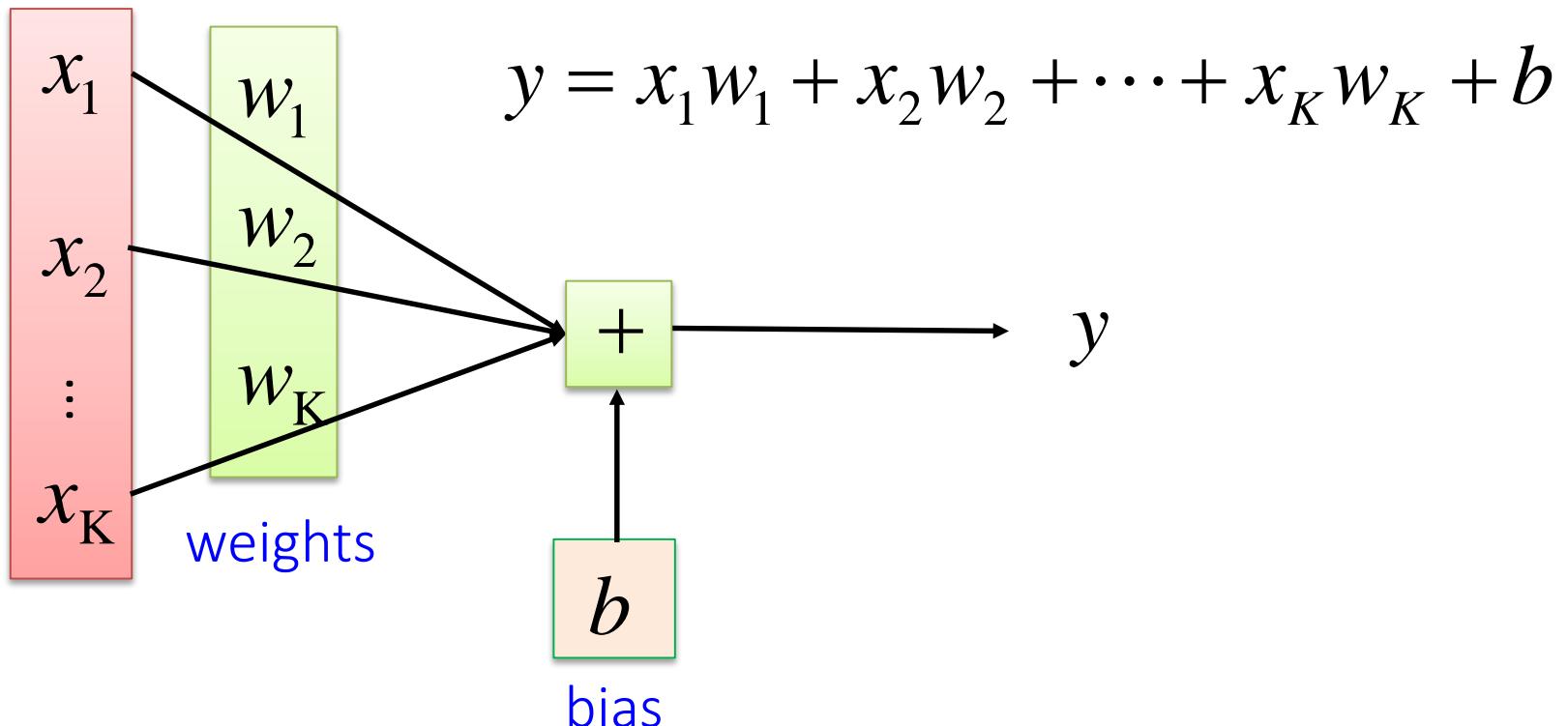
$x_1$  : Accelerometers Energy

Driving detection (yes/no)  
= define a decision function



# From linear classifiers TO Neural Networks

## A Linear decision function



# From linear classifiers TO Neural Networks

## A Linear decision function

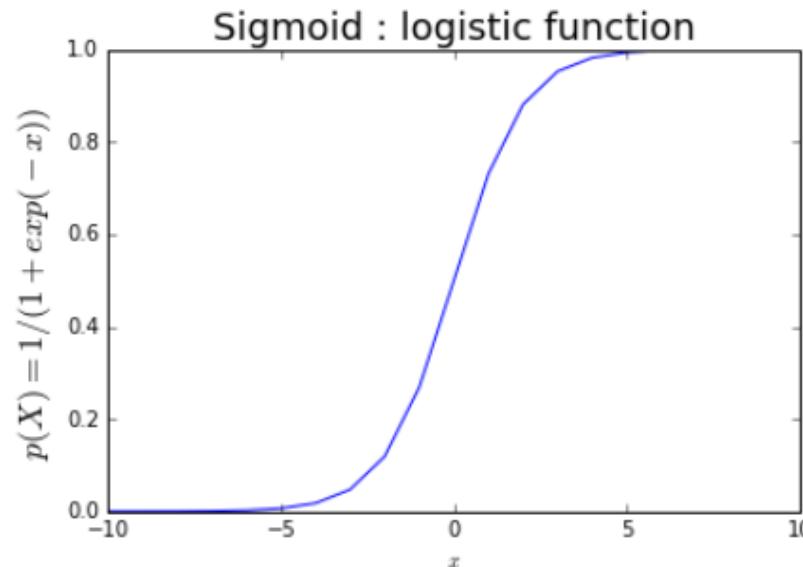
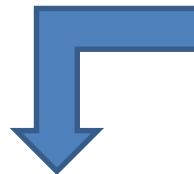
$$y = x_1 w_1 + x_2 w_2 + \cdots + x_K w_K + b$$

$$y = \mathbf{x}^T \mathbf{w}$$

# From linear classifiers TO Neural Networks

We typically use a **Sigmoid Function**

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

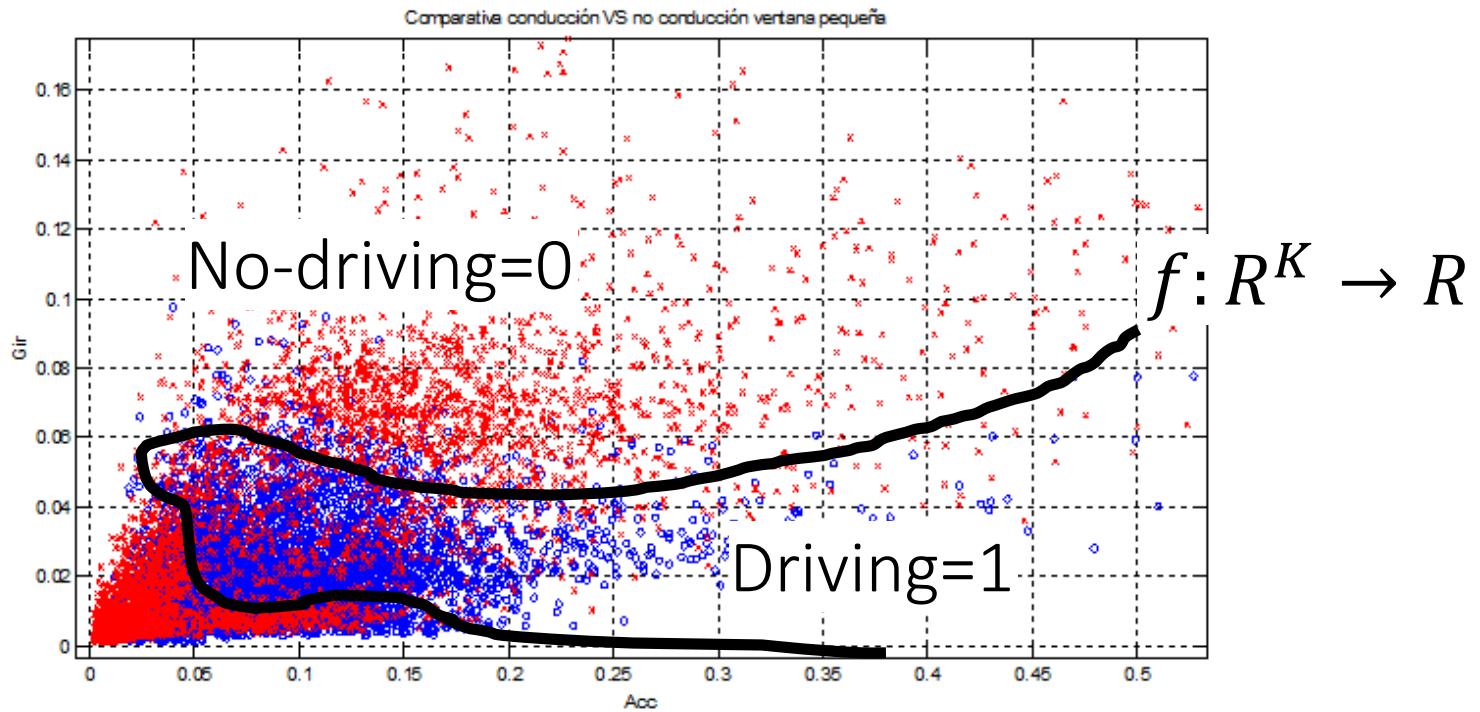


To approach *posterior* probabilities:

- $P(\text{Driving} | X)$
  - $P(\text{No Driving} | X)$
- $\uparrow$   
 $\mathbf{x}^T \mathbf{w}$

# From linear classifiers TO Neural Networks

## Nonlinear decision function?



# Non-linear decision functions

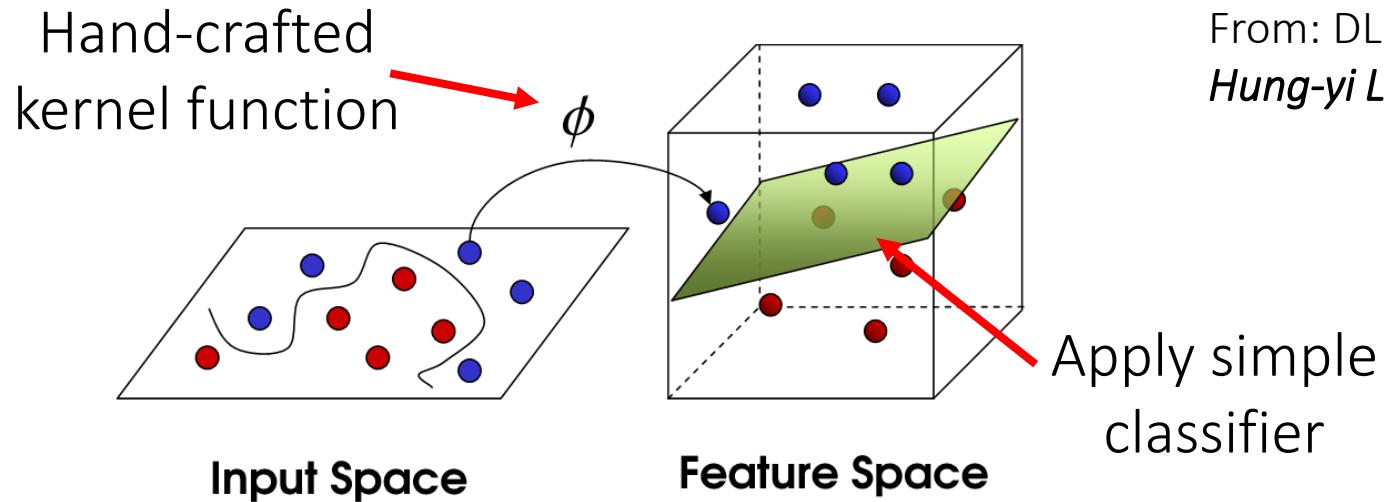
$$y = x_1 w_1 + x_1^2 w_2 + x_1^3 w_3 + x_1 x_2 w_4 + \dots + b$$

A linear model of transformed inputs:

$$y = \phi(\mathbf{x})^T \mathbf{w}$$

$\phi(\mathbf{x})$  where  $\phi$  is a non linear transformation

## SVM



Source of image: [http://www.gipsa-lab.grenoble-inp.fr/transfert/seminaire/455\\_Kadri2013Gipsa-lab.pdf](http://www.gipsa-lab.grenoble-inp.fr/transfert/seminaire/455_Kadri2013Gipsa-lab.pdf)

# How choosing the mapping $\phi(\cdot)$ ?

1. Feature engineering (based on expert knowledge)  $\phi(\cdot)$
2. Use a very generic  $\phi(\cdot)$  as kernel machines (e.g. SVM, RBF kernel)

# How choosing the mapping $\phi(\cdot)$ ?

1. Feature engineering (based on expert knowledge)  $\phi(\cdot)$
2. Use a very generic  $\phi(\cdot)$  as kernel machines (e.g. SVM, RBF kernel)
3. The strategy of Deep Learning : to learn  $\phi(\cdot)$

# The DL approach: learn $\phi(\cdot)$

Now we have:

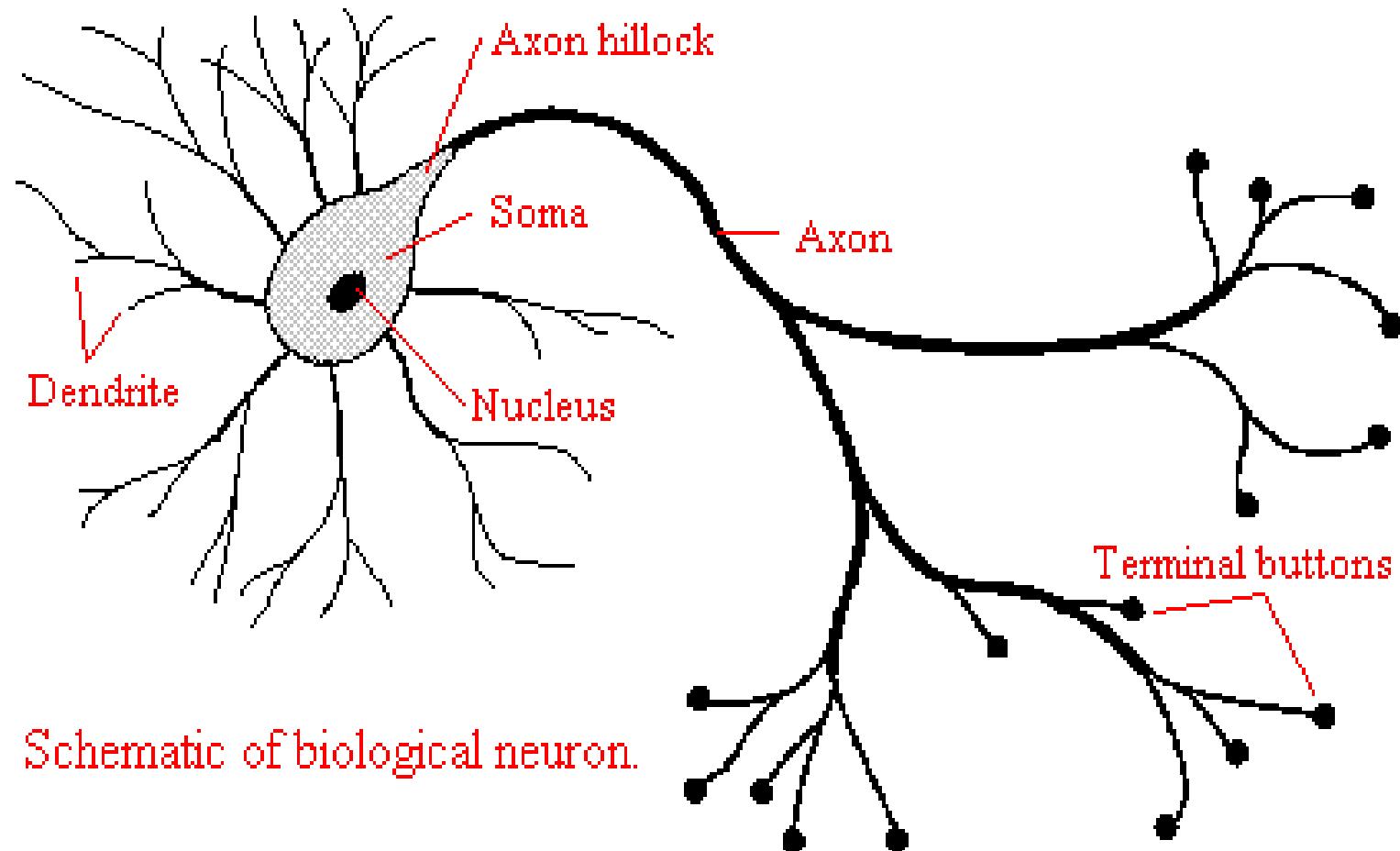
- Parameters  $\theta$  that we use to learn  $\phi(\cdot)$  in a broad class of functions
- Parameters  $w$  to learn  $\phi(x)$  to be desired

BUT .... what  $\phi(\cdot)$  ?

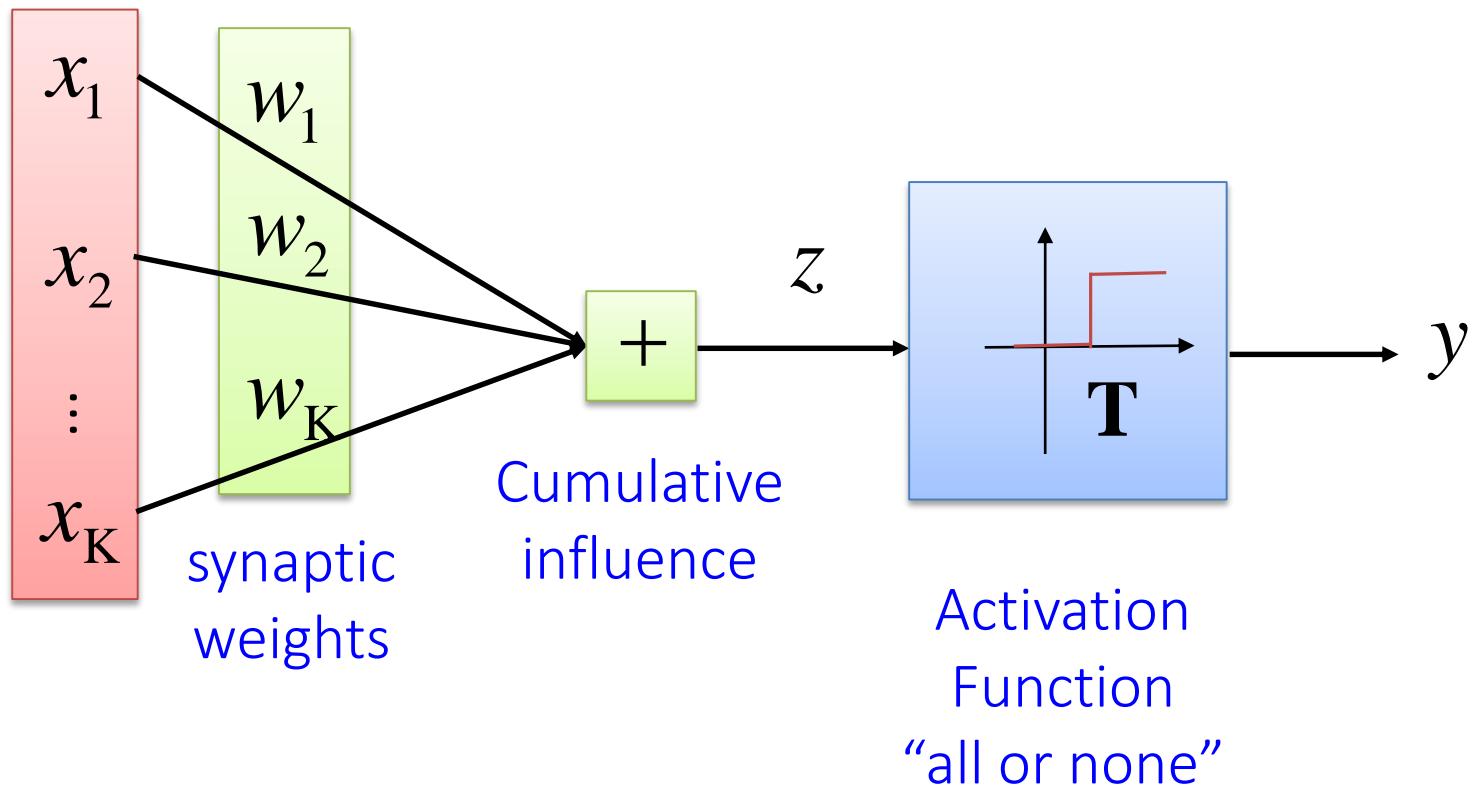
$$y = f(\mathbf{x}; \theta, \mathbf{w}) = \phi(\mathbf{x})^T \mathbf{w}$$

# The DL approach: learn $\phi(\mathbf{x})$

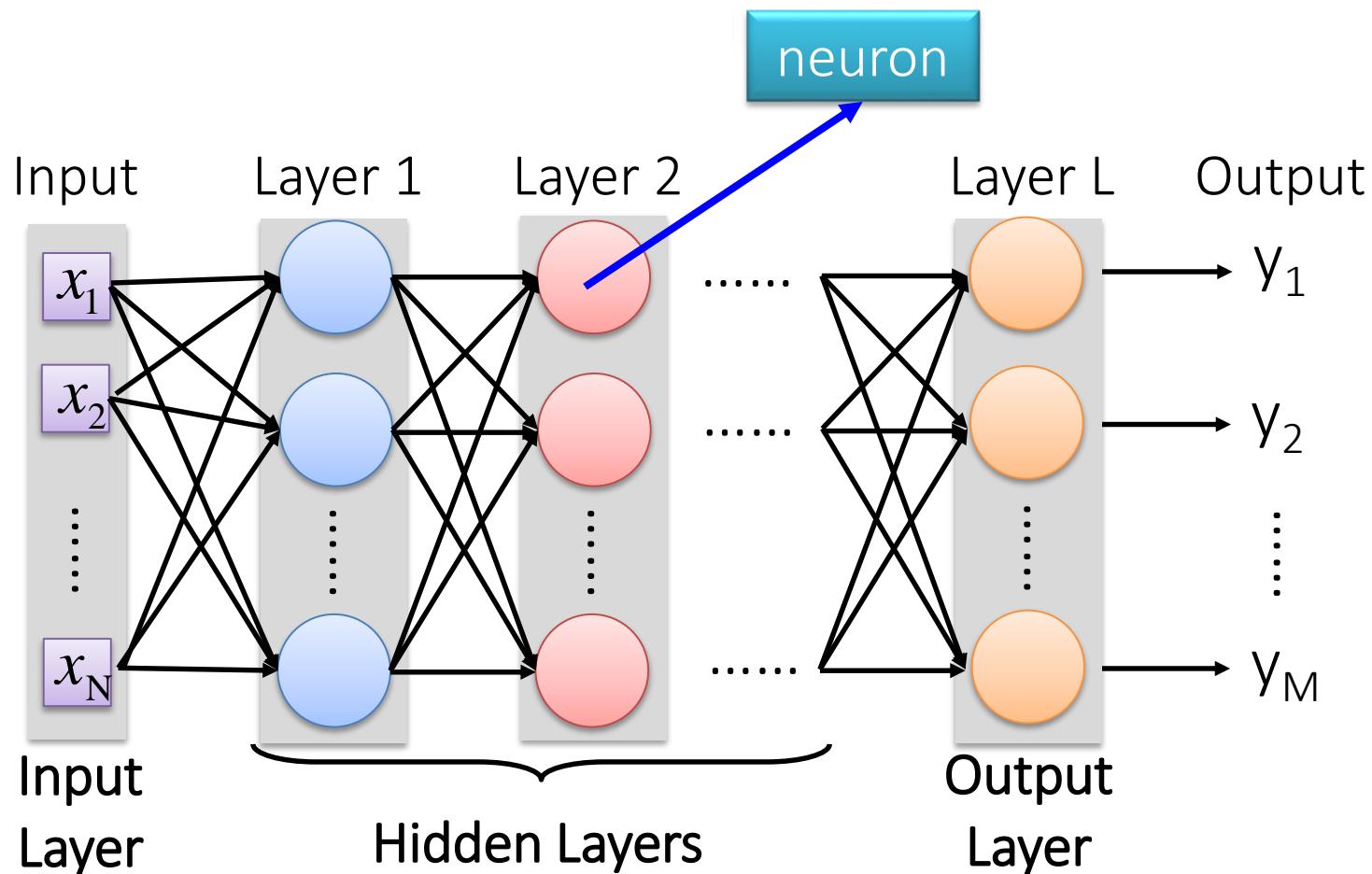
...from a broad class of functions



# Naïve Neuron approach....

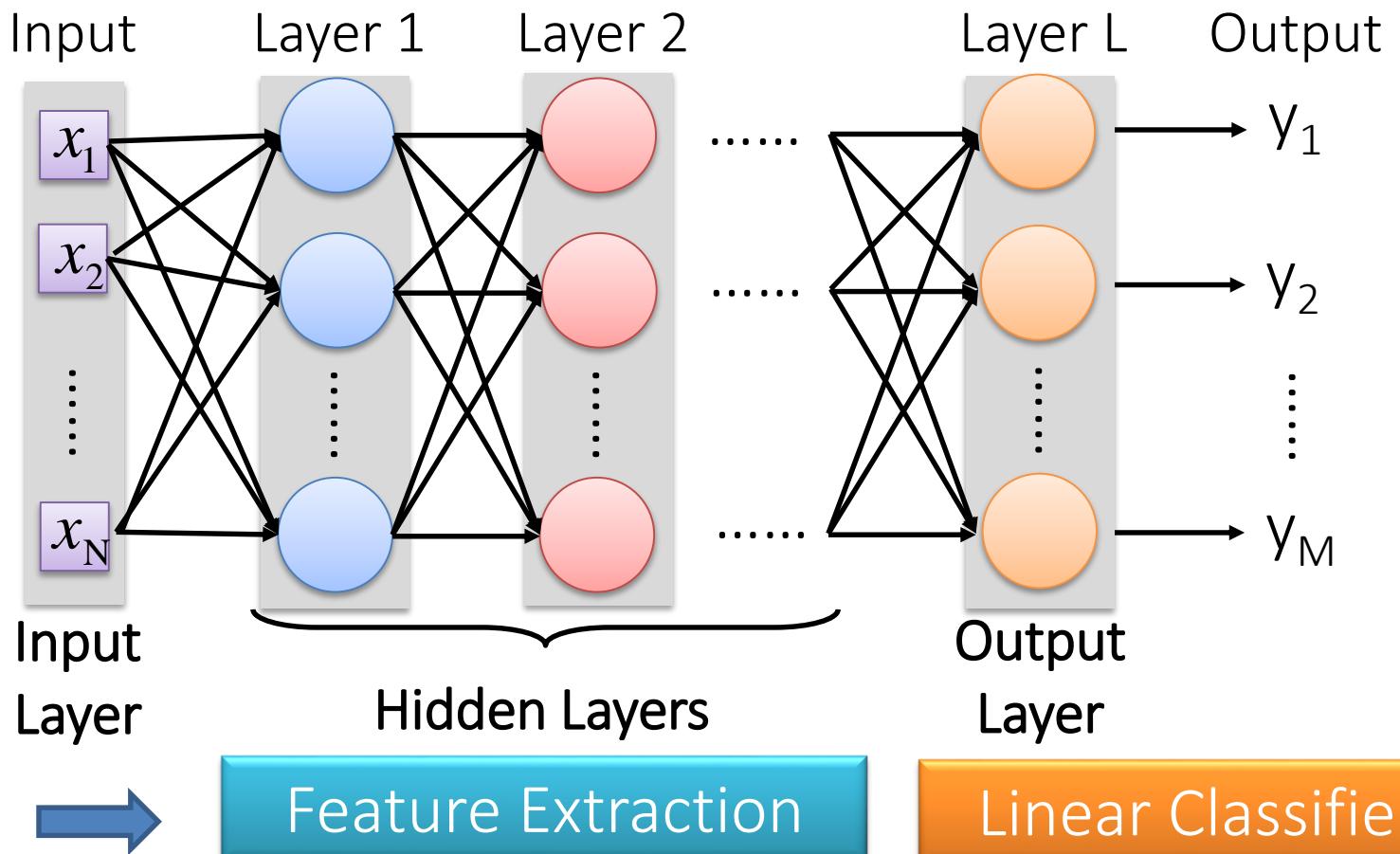


# Neural Network (from Hung-yi Lee “Deep Learning Tutorial”)



Deep means many hidden layers

# Feature Extraction + Final Linear Classifier



# From linear classifiers TO Neural Networks

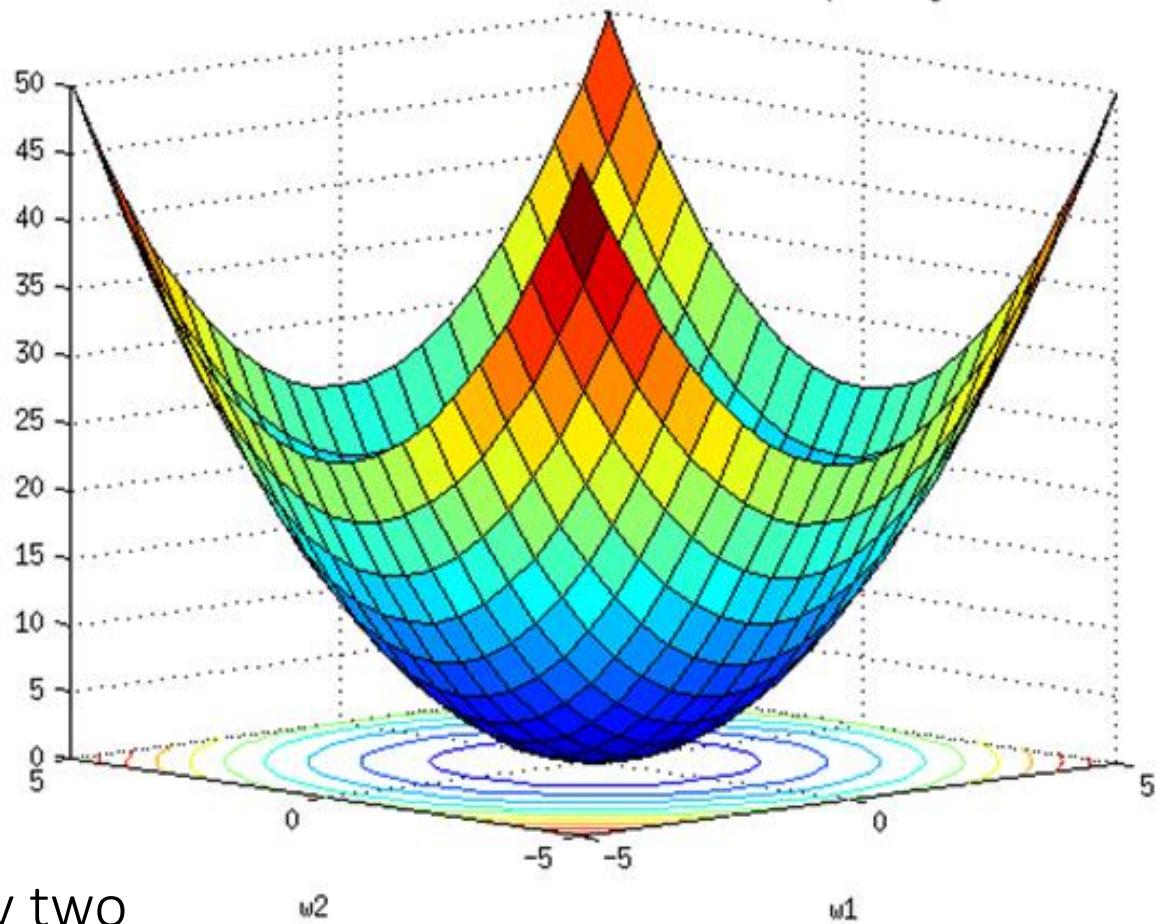
- Parameters  $\Theta$  that we use to learn  $\phi(\cdot)$  from a broad class of functions
- **Weights** and **thresholds** are estimated from training examples:
  - to minimize a **loss function** (i.e. similarity between NN outputs  $y$  and desired outputs  $\hat{y}$  )

# OPTIMIZATION

# Gradient Descent

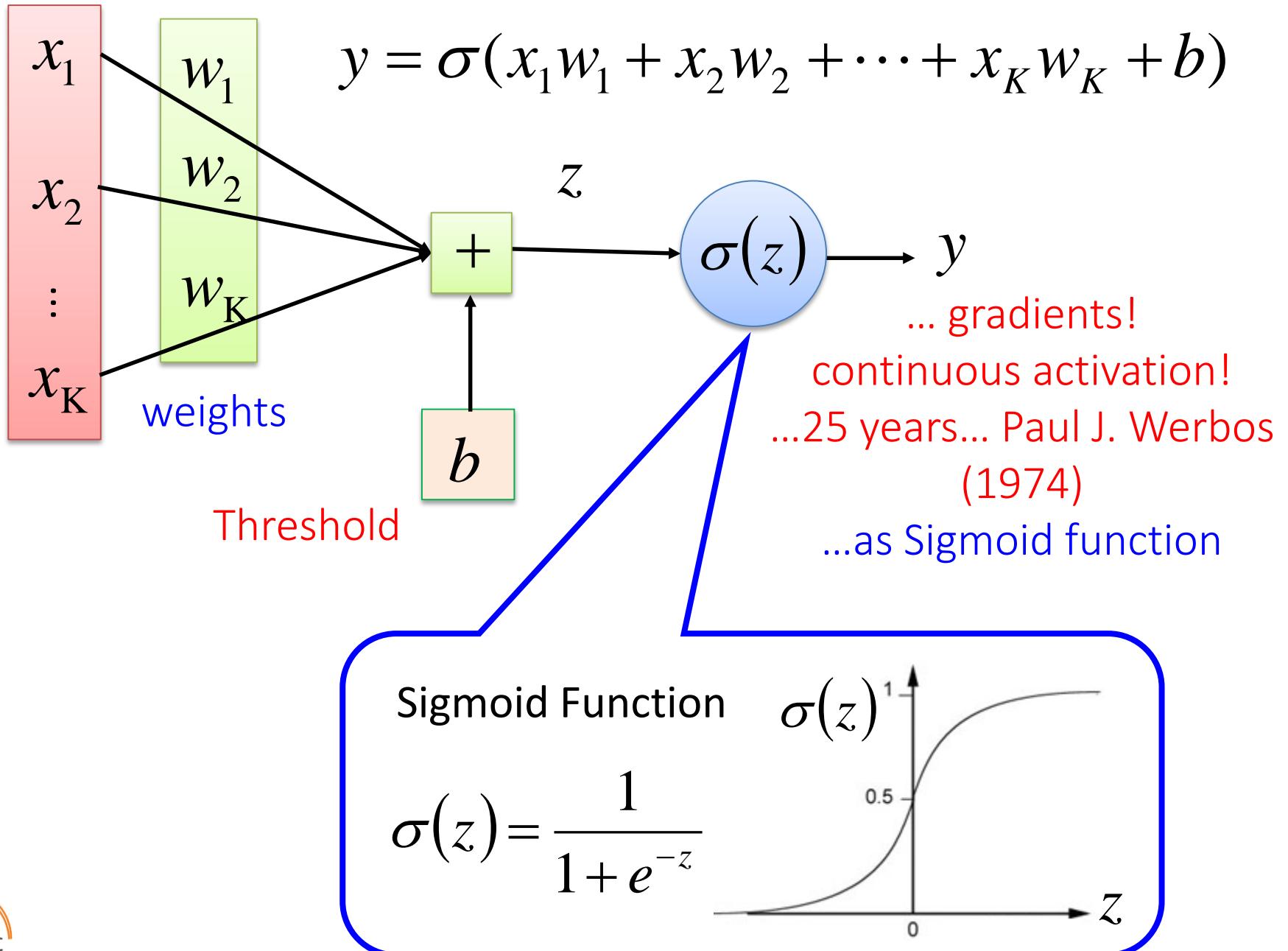
Cost  
Or  
Loss function

$C(\theta)$

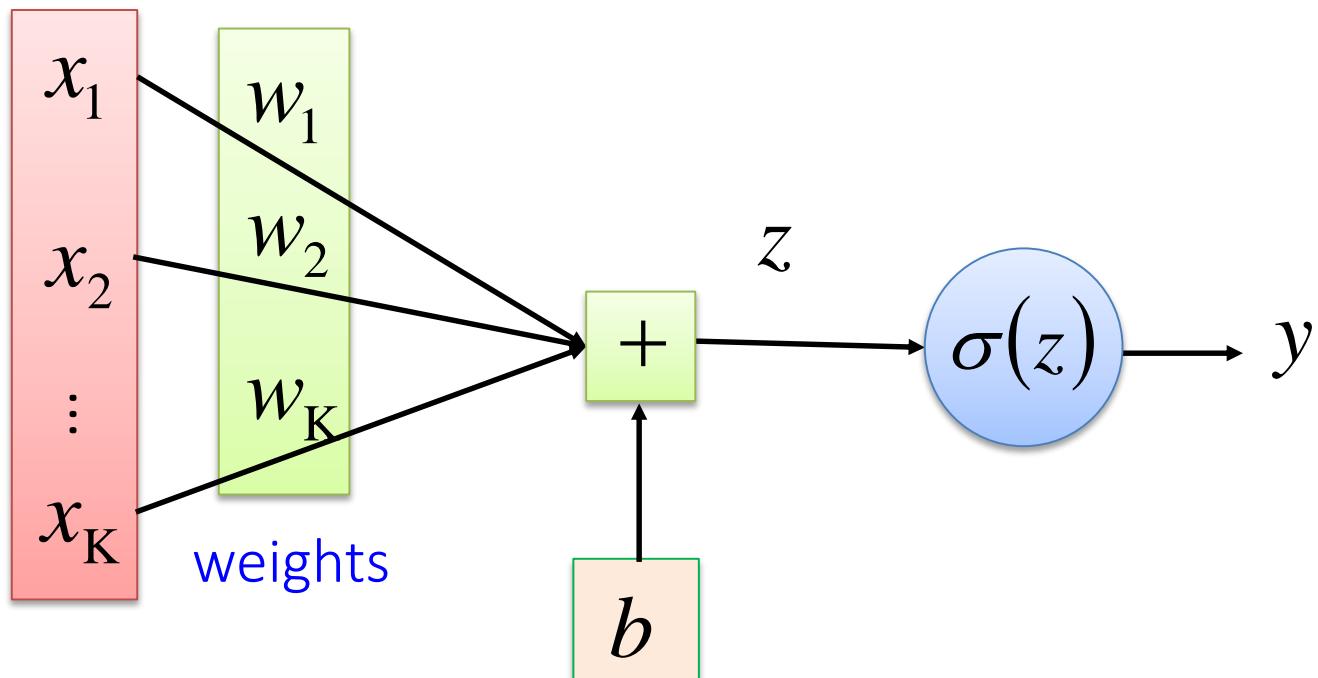


Assume there are only two parameters  $w_1$  and  $w_2$  in a network.

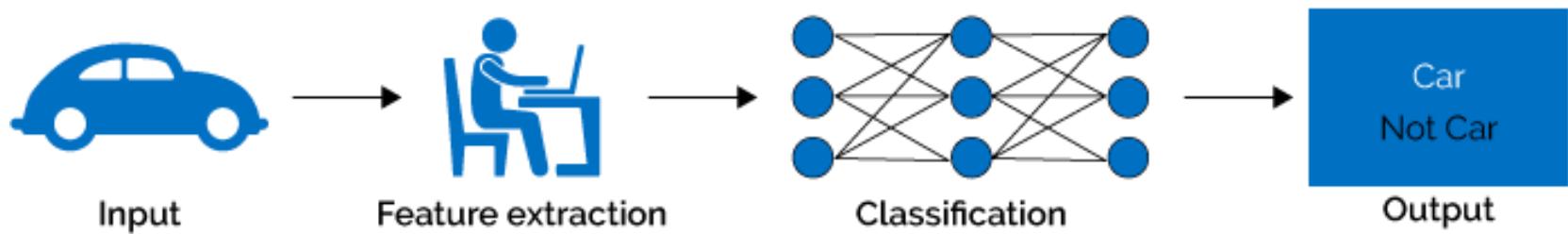
$$\theta = \{w_1, w_2\}$$



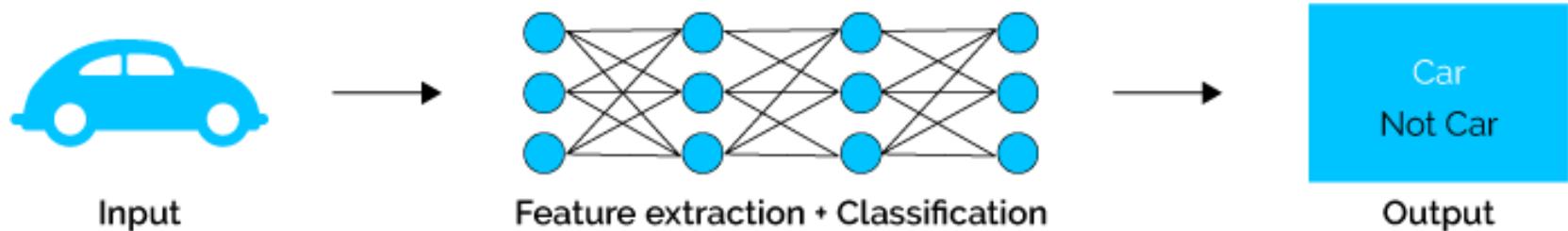
But recall that this is also:  
**logistic regression!**



## Machine Learning



## Deep Learning

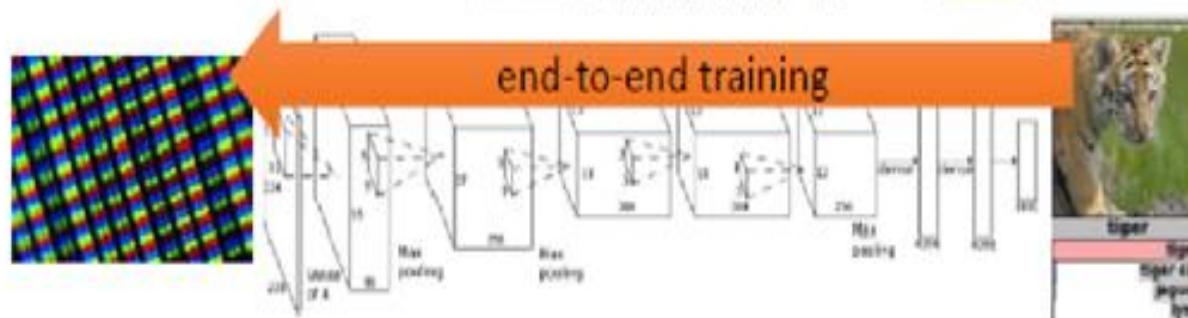


# End-to-end REVOLUTION!

standard  
computer  
vision



deep  
learning



# Towards End-to-End Speech Recognition

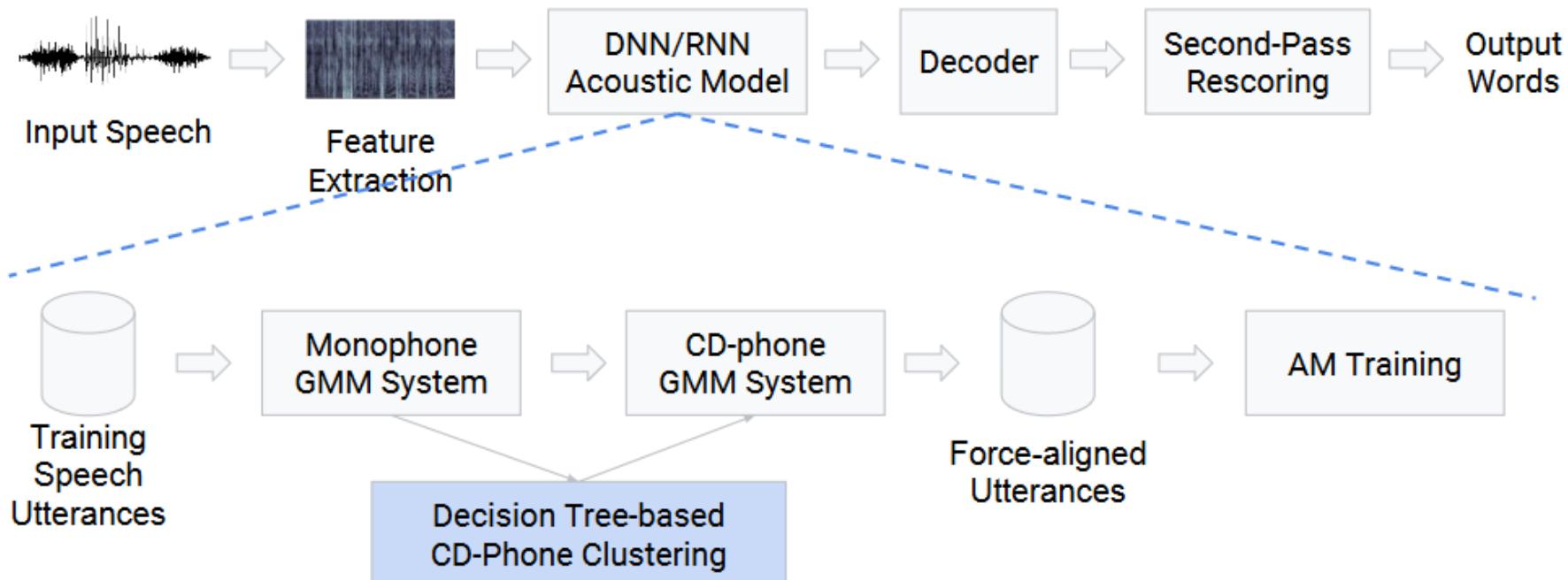
Bo Li, Yanzhang He, Shuo-Yiin Chang

ISCSLP Tutorial 4

Nov. 26, 2018

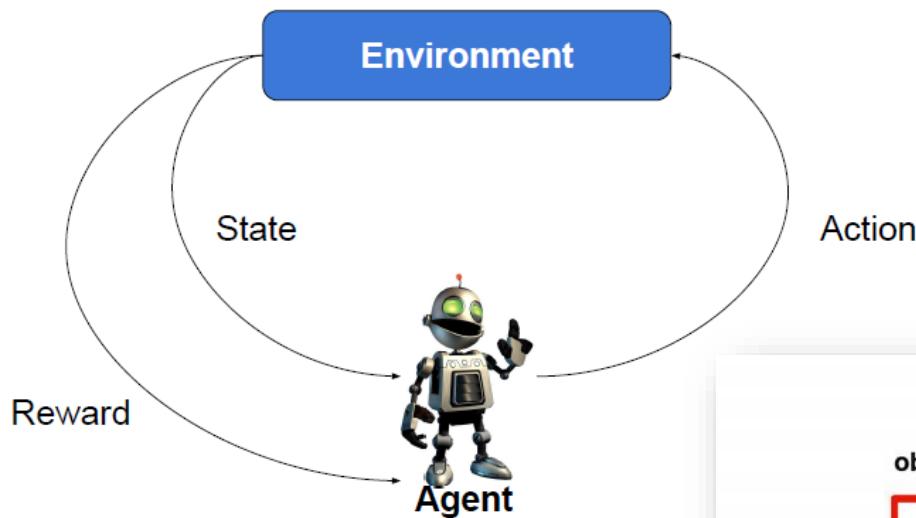
## Conventional ASR

### AM Training

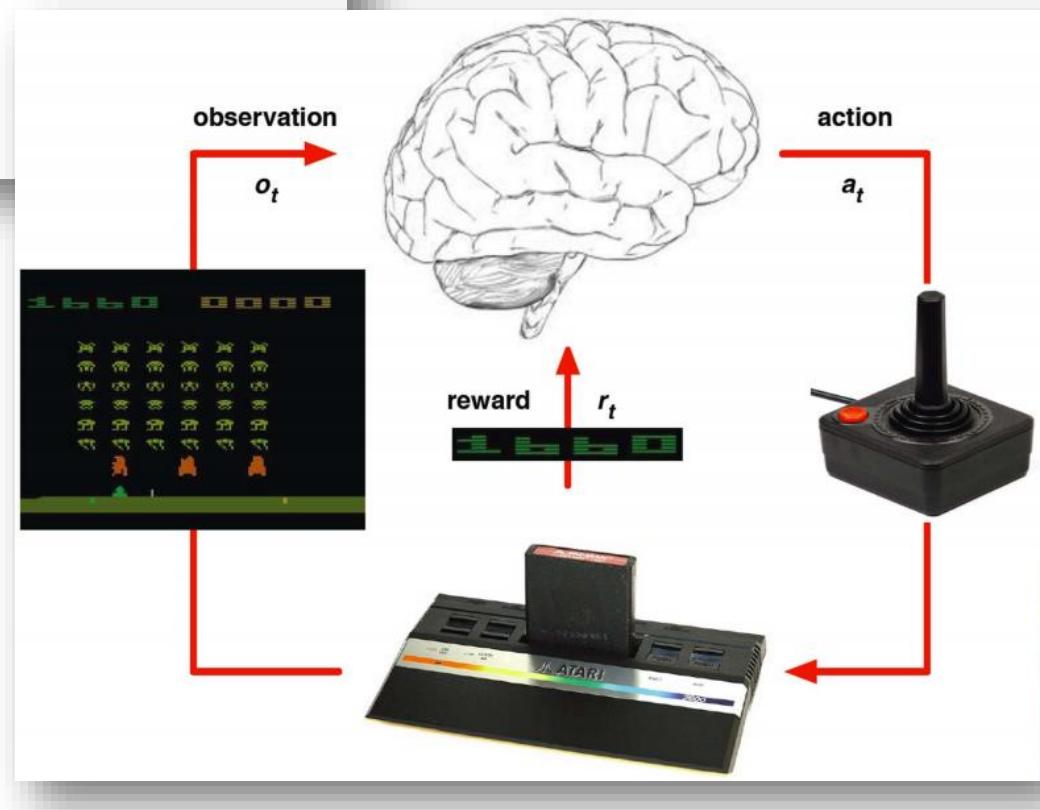


Google

# What's Reinforcement Learning?



- Agent interacts with an environment and learns by maximizing a reward.
- No labels or any other supervision is given.



## Get started with TensorFlow's High-Level APIs (Google I/O '18)



Recommended way to get started

- Try TensorFlow with zero install using Colab
- Train your first neural network in <10 lines of code using tf.keras
- Learn more about TensorFlow prerequisites

0:59

.../6



27:33 / 39:29



---

Intelligent Machines

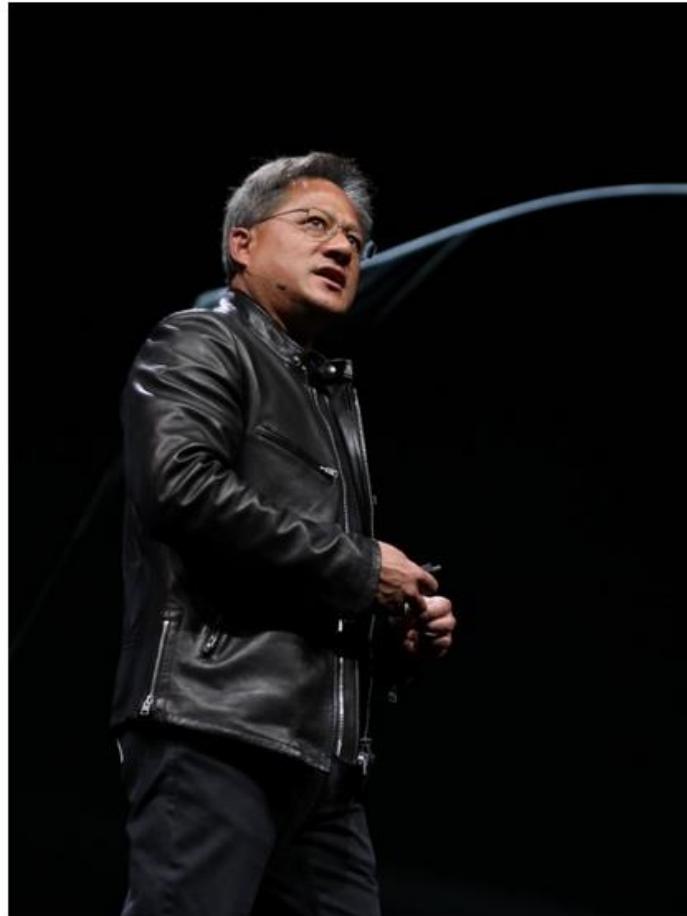
---

# Nvidia CEO: Software Is Eating the World, but AI Is Going to Eat Software

Jensen Huang predicts that health care and autos are going to be transformed by artificial intelligence.

by Tom Simonite May 12, 2017

Nvidia CEO Jensen Huang at the company's developer conference in San Jose, California.



---

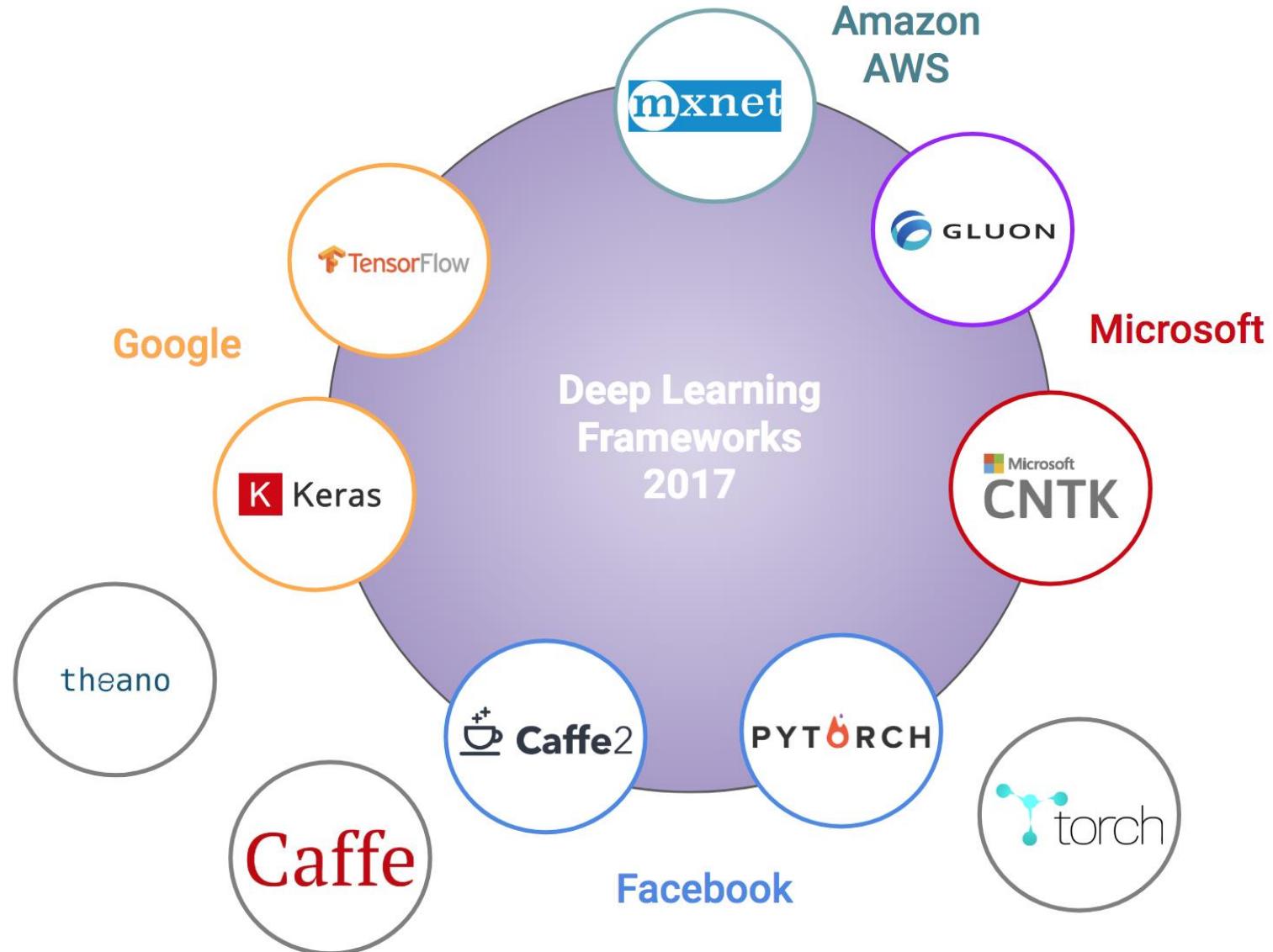
**Intelligent Machines**

# The U.S. Military Wants Its Autonomous Machines to Explain Themselves

The latest machine-learning techniques are essentially black boxes. DARPA is funding a number of efforts to open them up.

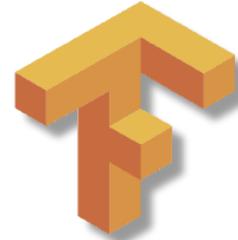
by Will Knight    March 14, 2017

---



<https://towardsdatascience.com/battle-of-the-deep-learning-frameworks-part-i-cff0e3841750>

# Google TensorFlow



- Library for writing “machine intelligence” algorithms
- Very popular for deep learning and neural networks
- Can also be used for general purpose numerical computations
- Interface in C++ and Python

## Pre-made Estimators

Estimator

Keras Model

Layers

Datasets

Python Frontend

C++

Java

Go

...

## TensorFlow Distributed Execution Engine

CPU

GPU

TPU

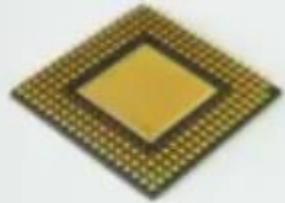
Android

XLA

iOS

...





CPU



GPU



Cloud TPU



Android



iOS



Embedded  
Devices

# API styles

tf.keras

Estimators

Eager execution

Deferred execution

# Deep Learning: some on-line resources



DataCamp

## Keras Tutorial: Deep Learning in Python

### 10 Best Machine Learning & Deep Learning Courses [2019 UPDATED]

<https://digitaldefynd.com/best-machine-learning-and-deep-learning-courses/>

# Keras: The Python Deep Learning library



# Keras

## Guiding principles

- User friendliness
- Modularity
- Easy extensibility
- Work with Python

BUT WHERE to (play) run  
your programs (scripts) ?

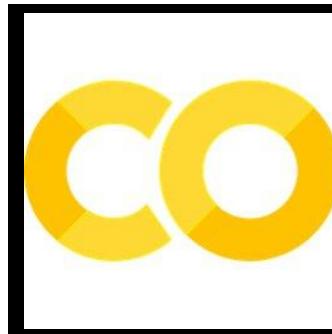
# BUT WHERE to (play) run your programs (scripts)

- Your local machine (ANACONDA)
- On the cloud
  - Google [Colaboratory](https://colab.research.google.com/notebooks/welcome.ipynb)  
<https://colab.research.google.com/notebooks/welcome.ipynb>
  - Kaggle kernels (powered by Google)  
<https://www.kaggle.com/docs/kernels>
  - IBM Cognitive Labs  
<https://labs.cognitiveclass.ai>

# Training and Testing:

MNIST

*Deep Learning Hello World!*



Google Colaboratory

Go to:

<https://www.tensorflow.org/tutorials>

The screenshot shows the TensorFlow website's 'Get Started with TensorFlow' page. On the left, there's a sidebar with a navigation menu under 'Learn and use ML'. The main content area features a heading 'Get Started with TensorFlow' and a brief introduction about TensorFlow being an open-source machine learning library. Below this, there's a section titled 'Learn and use ML' with a sub-section 'Basic classification' containing five numbered steps. To the right, a code editor window displays Python code for a Keras model to classify MNIST digits. A red circle highlights the 'Run code now' button at the bottom of the code editor.

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(512, activation=tf.nn.relu),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

Run code now Try in Google's interactive notebook

To Run it: You must login using your Gmail account

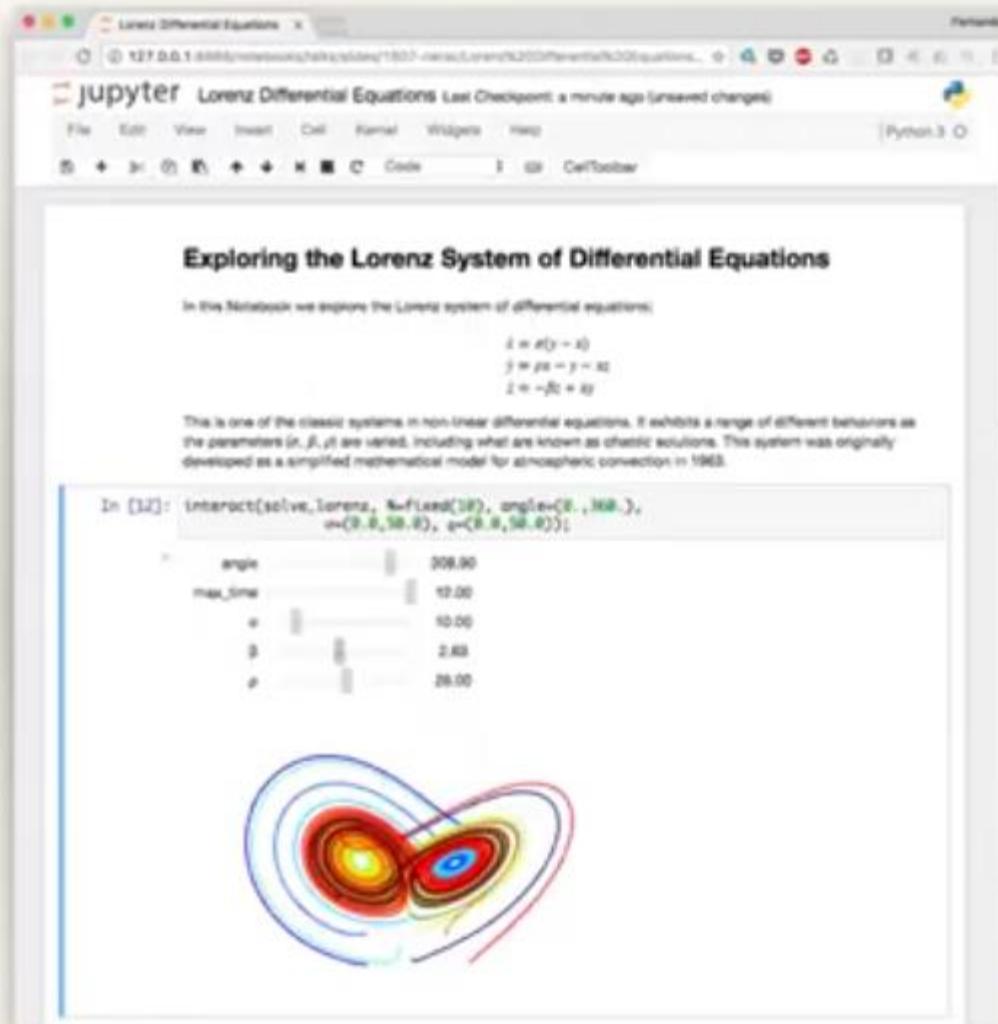
The screenshot shows a Google Colab notebook interface. At the top, a yellow bar displays the message "To Run it: You must login using your Gmail account". Below this, the browser address bar shows the URL [https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/\\_index.ipynb](https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/_index.ipynb). The main content area of the notebook shows the following text:

```
Copyright 2018 The TensorFlow Authors.  
Licensed under the Apache License, Version 2.0 (the "License");  
[ ] Licensed under the Apache License, Version 2.0 (the "License");  
  
- Get Started with TensorFlow  
  
This is a Google Colaboratory notebook file. Python programs are run directly in the browser—a great way to learn and use TensorFlow. To run the Colab notebook:  
1. Connect to a Python runtime: At the top-right of the menu bar, select CONNECT.  
2. Run all the notebook code cells: Select Runtime > Run all.  
  
For more examples and guides (including details for this program), see Get Started with TensorFlow.  
Let's get started, import the TensorFlow library into your program:  
  
[ ] from __future__ import absolute_import, division, print_function, unicode_literals  
import tensorflow as tf
```

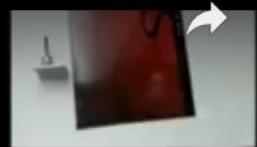
At the top right of the notebook interface, there are several buttons: "COMPARTIR", "Iniciar sesión" (which is circled in red), "CONECTAR", and "EDICIÓN". A red arrow points from the bottom of the "CONECTAR" button towards the "Iniciar sesión" button.

# The Jupyter Notebook

- ◊ Rich web client
- ◊ Text & math
- ◊ Code
- ◊ Results
- ◊ Share, reproduce.



Jupyter Protocol  
is language agnostic



# What is Jupyter?

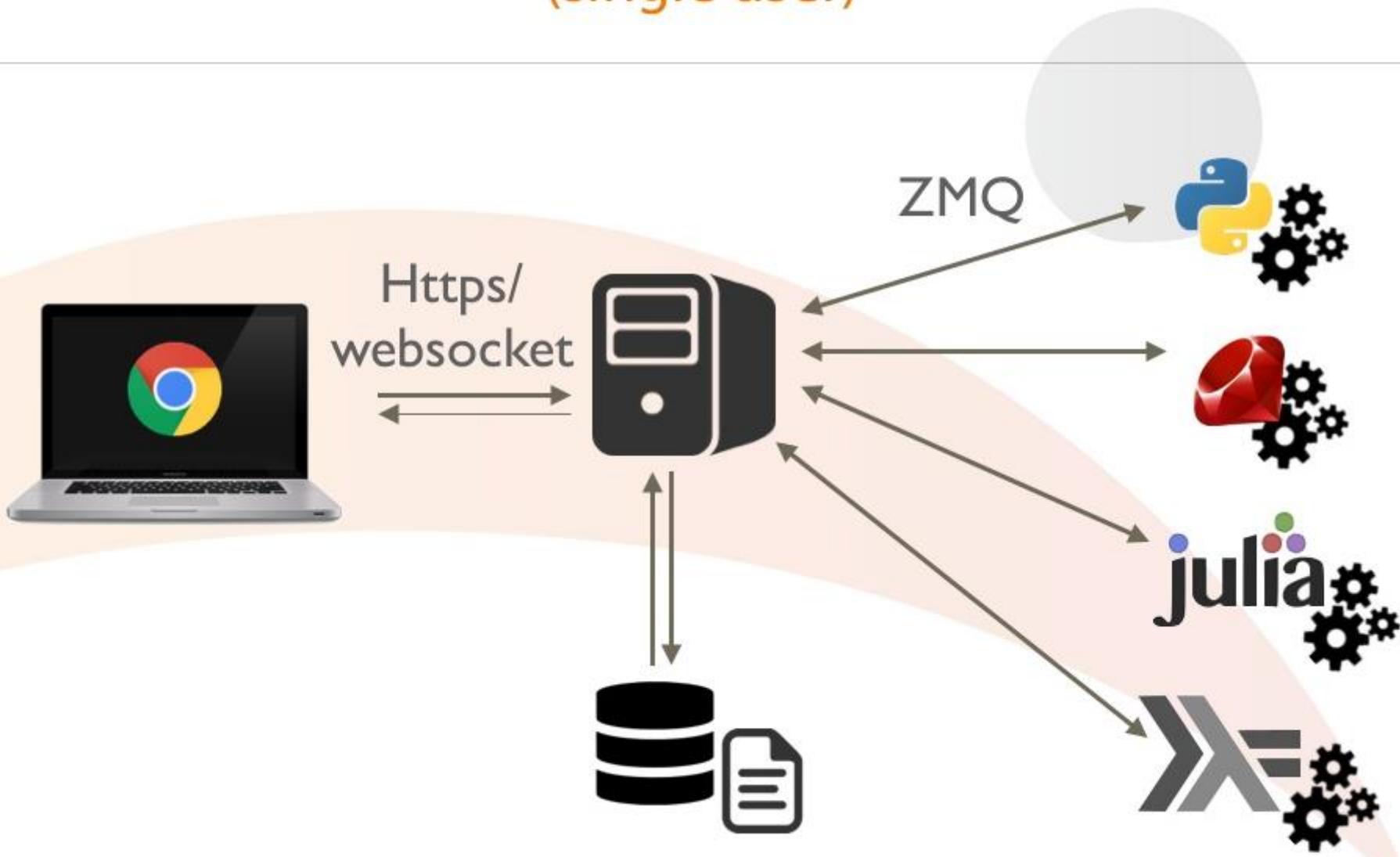
To succeed in digital transformation, businesses need to adopt AI and rapid deployment. Jupyter fits that bill.

By Mike Loukides. June 20, 2017

**The Jupyter architecture:** Jupyter is built from three parts:

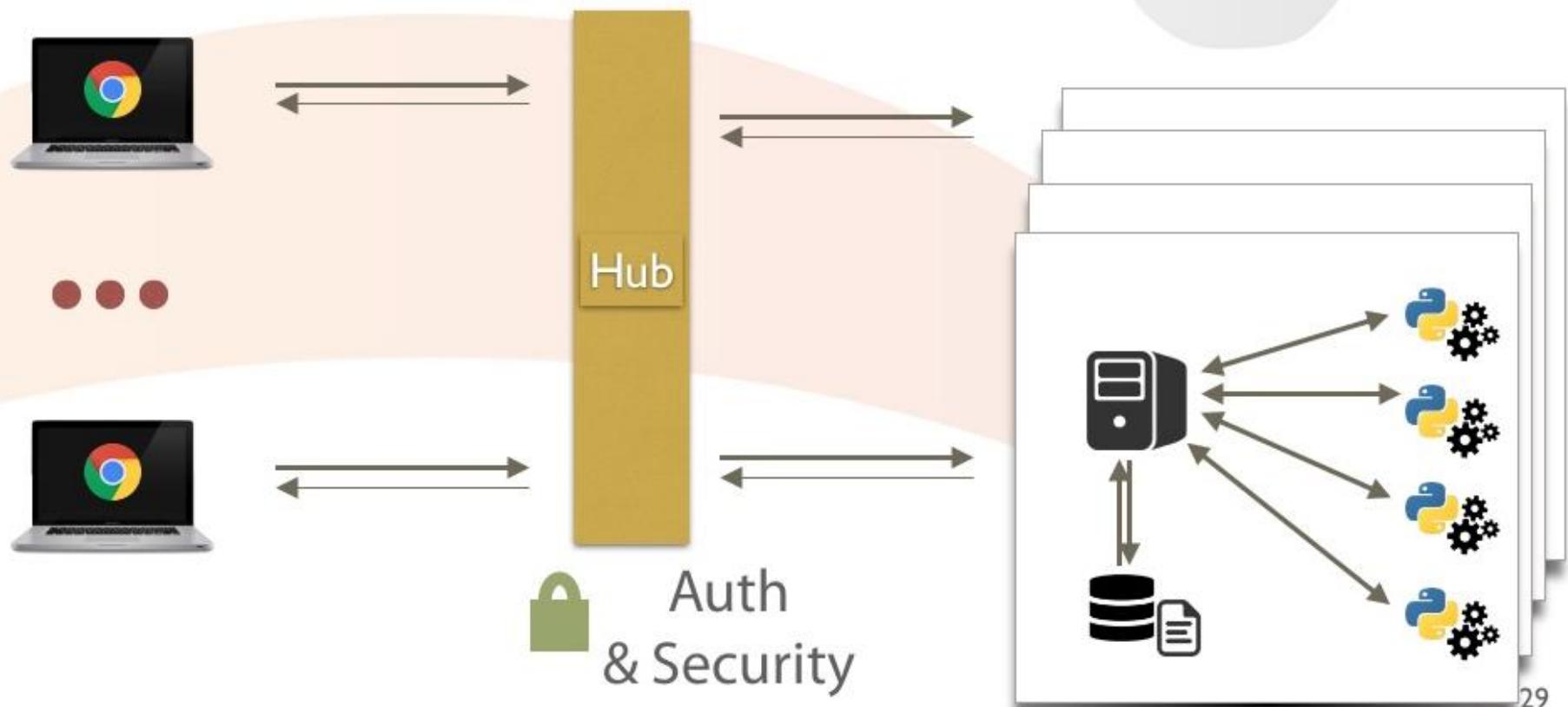
- 1. The notebook front end.** The front end allows you to edit and run notebooks (JavaScript application)
- 2. The Jupyter server,** which is either a relatively simple application that runs on your laptop, or a multi-user server. The Jupyter project's [JupyterHub](#) is the most widely used multi-user server for Jupyter.
- 3. The kernel protocol,** which allows the server to offload the task of running code to a language-specific kernel. Jupyter ships with kernels for Python 2 and Python 3, but kernels for many other languages are available.

# The networking architecture (single user)





Https/websocket proxy



# Now CREATE YOUR OWN Python 3 Notebook

The screenshot shows the Google Colab interface. At the top, there are four tabs: "TensorFlow Core | TensorFlow", "\_index.ipynb - Colaboratory", "\_index.ipynb - Colaboratory", and "Te damos la bienvenida a Cola". The main window displays a message: "Este cuaderno está abierto con resultados privados. Los resultados no se guardarán. Puedes inhabilitar esta opción en Configuración del cuaderno." Below this, the Colab menu bar includes "Archivo", "Editar", "Ver", "Insertar", "Entorno de ejecución", "Herramientas", and "Ayuda". A red oval highlights the "Nuevo cuaderno de Python 3" option in the "Archivo" dropdown menu. The left sidebar shows sections like "Índice", "Copyright", "Get Started", and "SECCIÓN". The main content area features the TensorFlow documentation for "Getting Started with TensorFlow". It includes links to "TensorFlow.org", "Run in Google Colab", and "View source on GitHub". A code cell at the bottom contains the following Python code:

```
from __future__ import absolute_import, division, print_function, unicode_literals
import tensorflow as tf
```

Below the code, a note says: "Load and prepare the [MNIST](#) dataset. Convert the samples from integers to floating-point numbers:"



## Untitled0.ipynb ⭐

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

+ CÓDIGO + TEXTO

↑ CELDA ↓ CELDA

[4] a=[1,2,3]

a\*3

↳ [1, 2, 3, 1, 2, 3, 1, 2, 3]

[5] type(a)

↳ list

[8] import numpy as np

av=np.array(a)

av\*3

↳ array([3, 6, 9])

▶ import numpy as np

av=np.array(a)

av\*3

↳ File "&lt;ipython-input-9-24975934ba49&gt;", line 5

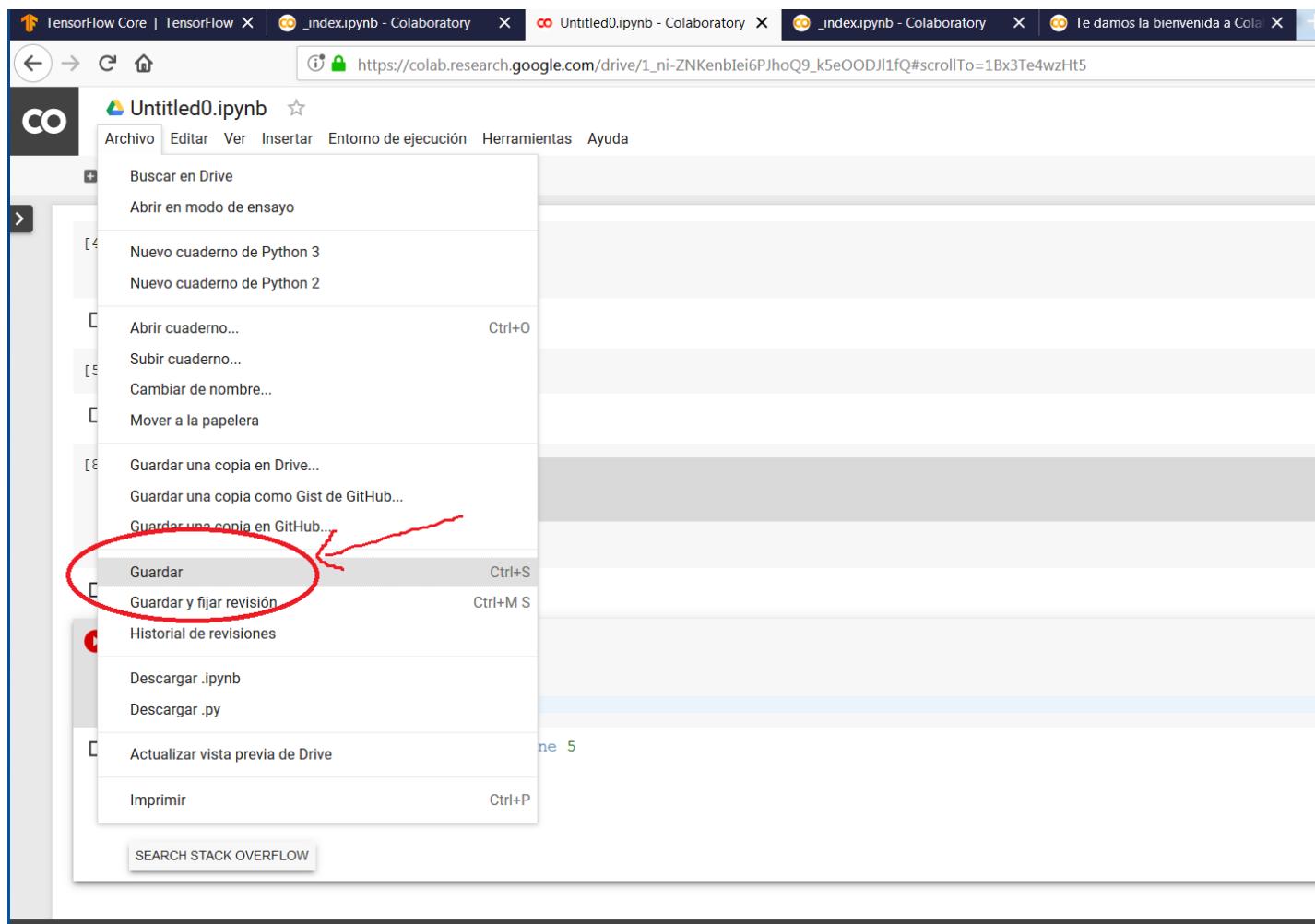
av\*3

^

IndentationError: unexpected indent

SEARCH STACK OVERFLOW

# Save it....



# WHERE?

By default it is saved into your Google Drive under folder “Colab Notebooks”

The screenshot shows a Google Drive interface. The left sidebar includes links for 'My Drive', 'Computers', 'Shared with me', 'Recent', 'Starred', 'Trash', and 'Backups'. Below this is a 'Storage' section indicating '6.9 GB of 15 GB used' and a 'UPGRADE STORAGE' button. The main area displays a list of files under 'My Drive > Colab Notebooks'. The files listed are:

Name	Owner
Copia de Te damos la bienvenida a Colaboratory	me
IPTC_GoogleColab_Intro.ipynb	me
Untitled0.ipynb	me

The file 'Untitled0.ipynb' is highlighted with a red oval.

# ANACONDA NAVIGATOR

[Sign in to Anaconda Cloud](#)[Home](#)[Environments](#)[Projects \(beta\)](#)[Learning](#)[Community](#)[Documentation](#)[Developer Blog](#)[Feedback](#)

Applications on

root

Channels

Refresh



jupyterlab  
0.27.0

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

[Launch](#)



jupyter  
notebook  
5.0.0

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

[Launch](#)



qtconsole  
4.3.1

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

[Launch](#)



spyder  
3.2.4

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

[Launch](#)



glueviz  
0.10.4

Multidimensional data visualization across files. Explore relationships within and among related datasets.

[Install](#)



orange3  
3.4.1

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

[Install](#)



rstudio  
1.1.383

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

[Install](#)

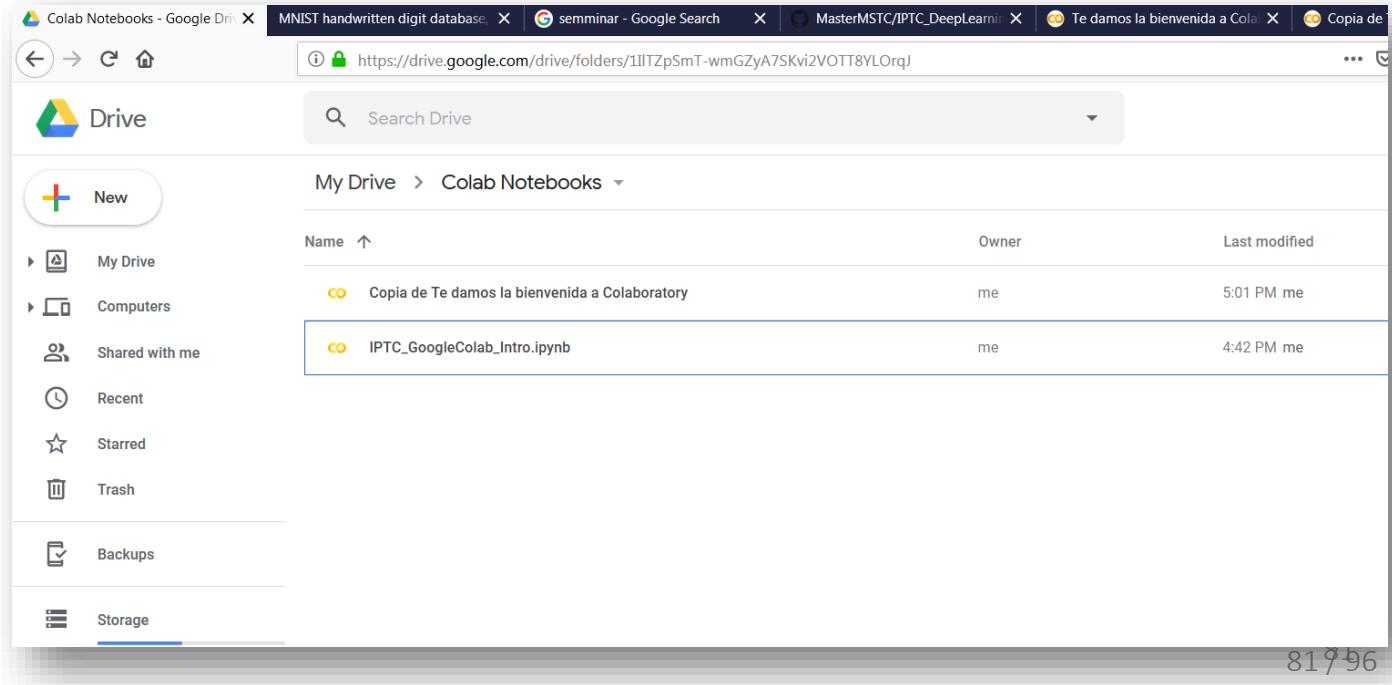
Please go to our MSTC GitHub:

[https://github.com/MasterMSTC/IPTC\\_DeepLearning](https://github.com/MasterMSTC/IPTC_DeepLearning)

...and copy from Notebooks:

IPTC\_GoogleColab\_Intro.ipynb

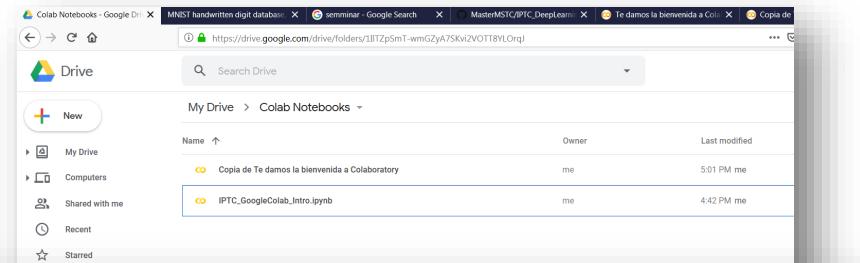
in to your: Google Drive



# Now, from your Google Drive open the Notebook

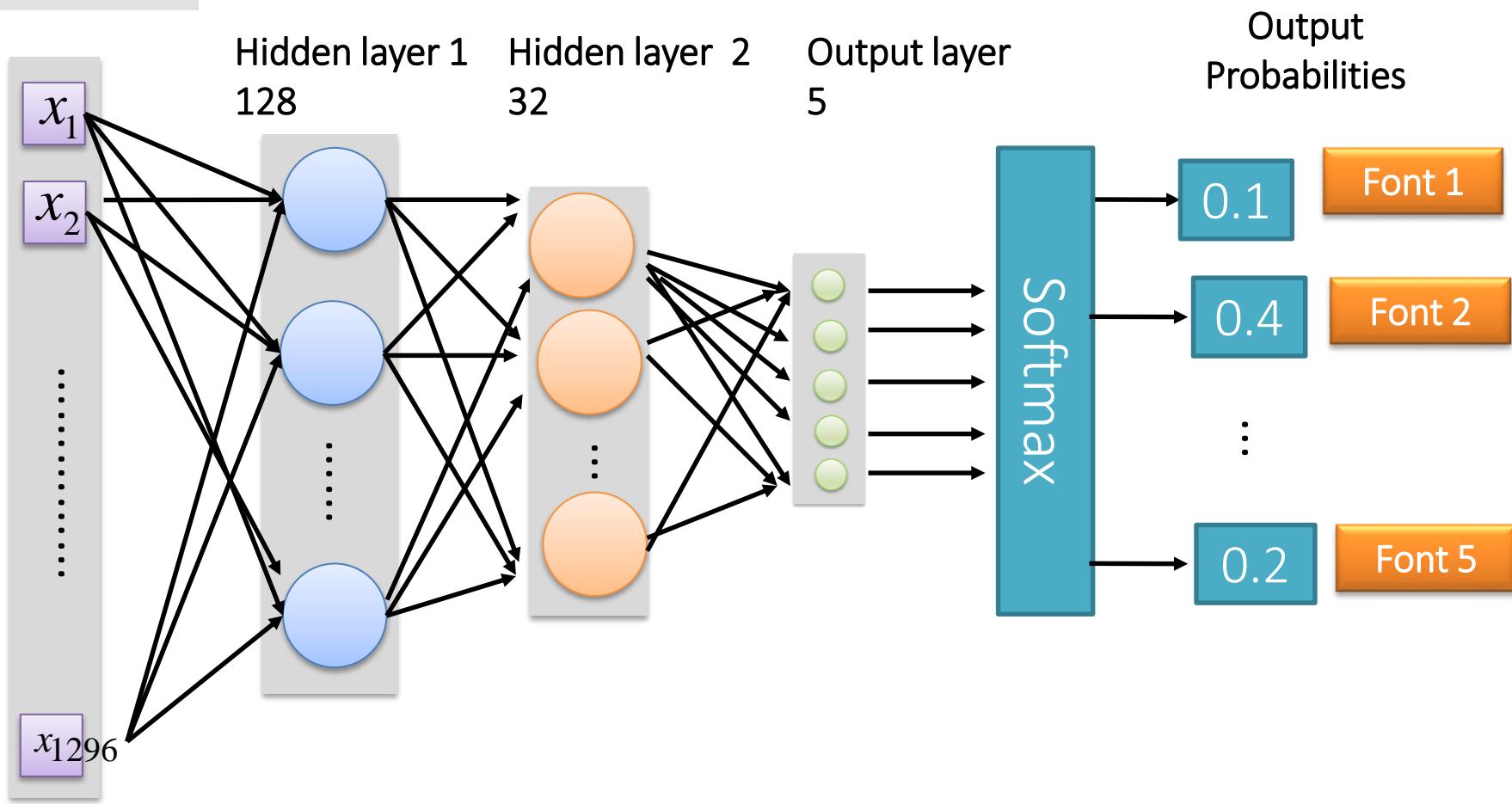
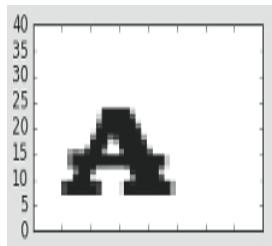
## IPTC\_GoogleColab\_Intro.ipynb

...using Google Colab



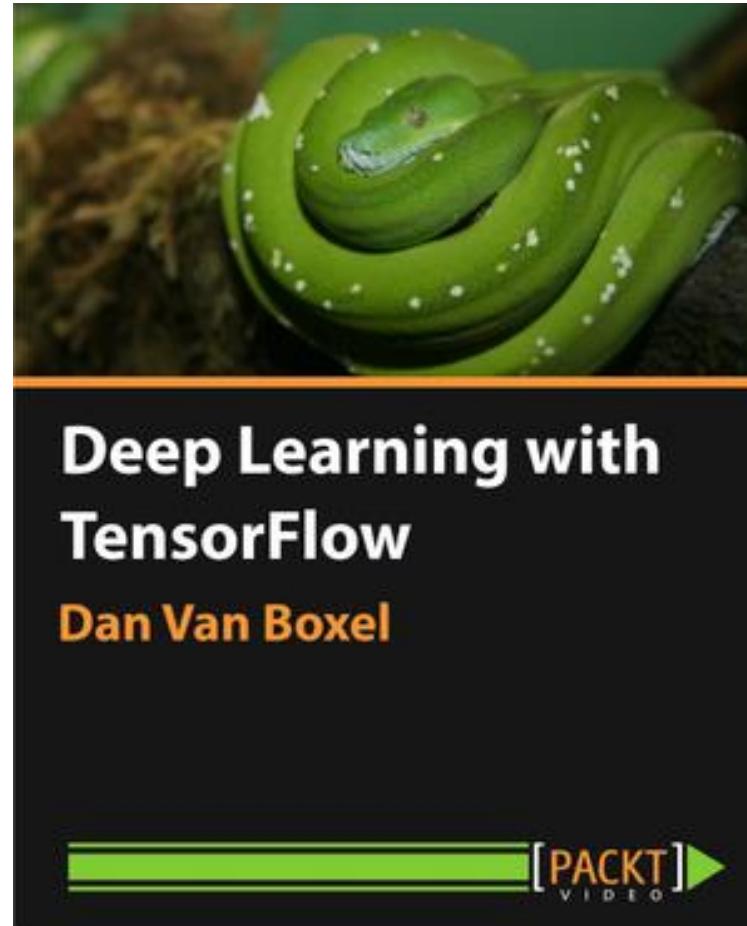
A screenshot of a Google Colab notebook titled 'IPTC\_GoogleColab\_Intro.ipynb'. The top navigation bar includes 'Archivo', 'Editar', 'Ver', 'Historial', 'Marcadores', 'Herramientas', 'Ayuda', and tabs for 'Colab Notebooks - Google Drive', 'IPTC\_GoogleColab\_Intro.ipynb', 'MNIST handwritten digit database', 'seminar - Google Search', 'MasterMSTC/IPTC\_DeepLearn', 'Te damos la bienvenida a Colab', and 'Copia de'. Below the navigation is a toolbar with icons for back, forward, search, and file operations. The main content area has a header 'IPTC Seminar' and a section 'Practical Introduction to Deep Learning and Keras'. A red box highlights the heading 'Google colab Intro'. Below it, a text block explains that Google Colab is a google internal research tool for data science and has been released to the general public with a goal of disseminating machine learning education and research. It lists two bullet points: 'You can use GPU as a backend for free for 12 hours at a time.' and a numbered list: '1. The GPU used in the backend is K80 (at this moment). 2. The 12-hour limit is for a continuous assignment of VM. It means we can use GPU compute even after the end of 12 hours by connecting to a different VM.' At the bottom, there is a code cell starting with '[ ] ! cat /etc/lsb-release'.

# Let's try to develop an Image Classifier using a FeedForward Network

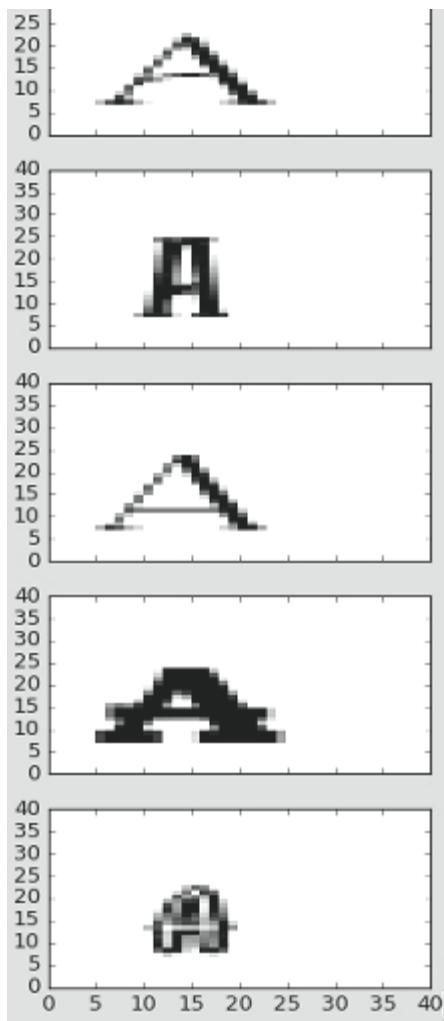


# Font type Recognition

K Keras



# One-hot-encoding (OHE)



Categories	Number of Categories (N=5)				
Type 1	1	0	0	0	0
Type 2	0	1	0	0	0
Type 3	0	0	1	0	0
Type 4	0	0	0	1	0
Type 5	0	0	0	0	1

# Softmax Layer

*The output of a Classifier is a categorical variable*

- In REGRESSION the output  $y$  is a **Numerical** variable
- BUT in CLASSIFICATION  $y$  is a **Categorical** one

So: How to represent **Categorical** variables?

$$y = \{\text{cats, dogs, birds, ants}\}$$

- Why not just to assign “1”, “2” , “3” ?

## OHE : One-Hot encoding

cats : 1 0 0 0

dogs: 0 1 0 0

birds: 0 0 1 0

ants: 0 0 0 1

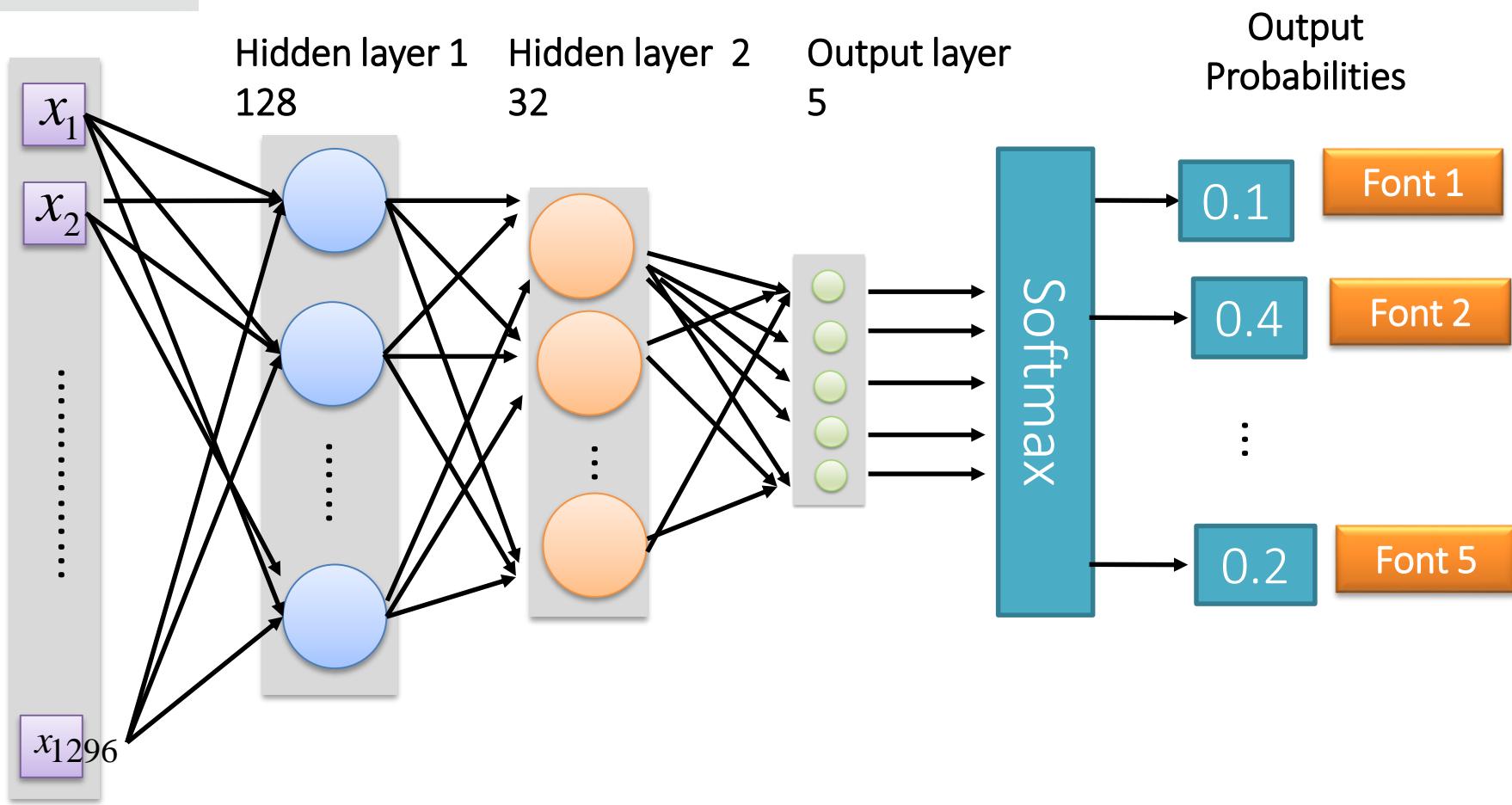
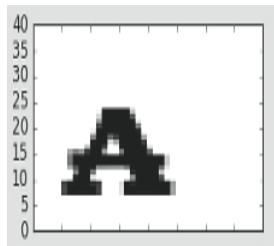
- ...and what for a **LARGE** number of categories (as words!) .... **sparsity** (a lot of zeros)....

On Day-3 we will talk about:

**embeddings**

Great in NLP

(Natural Language Processing)

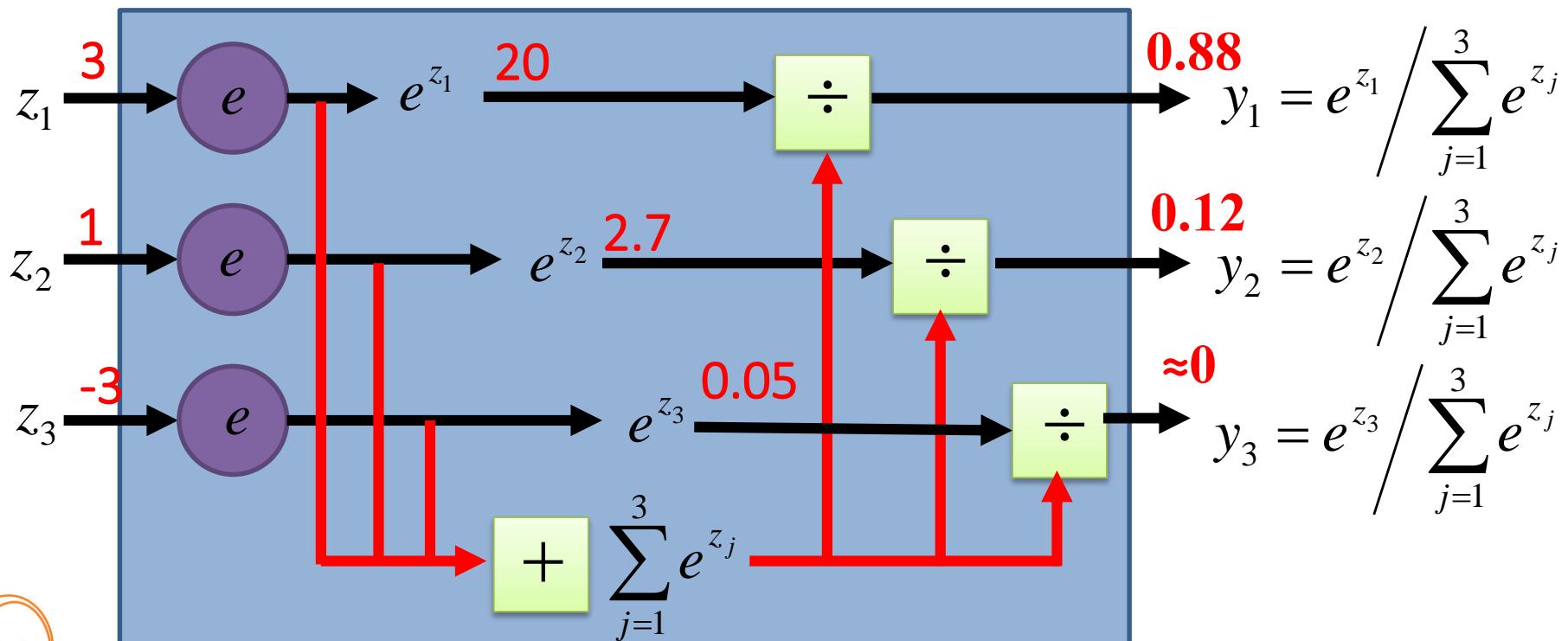


# Softmax layer as the output layer

Probability:

- $1 > y_i > 0$
- $\sum_i y_i = 1$

Softmax Layer



# Font type Recognition



IPTC\_Keras\_FontReco\_FeedForward.ipynb

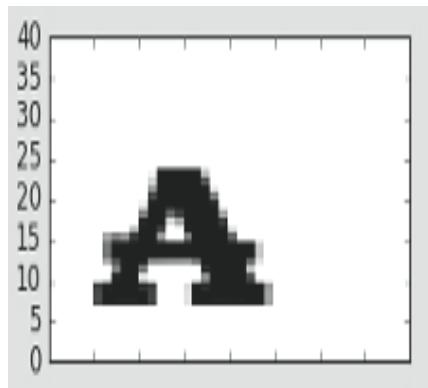
Available at:

[https://github.com/MasterMSTC/IPTC\\_DeepLearning](https://github.com/MasterMSTC/IPTC_DeepLearning)

# Reshape

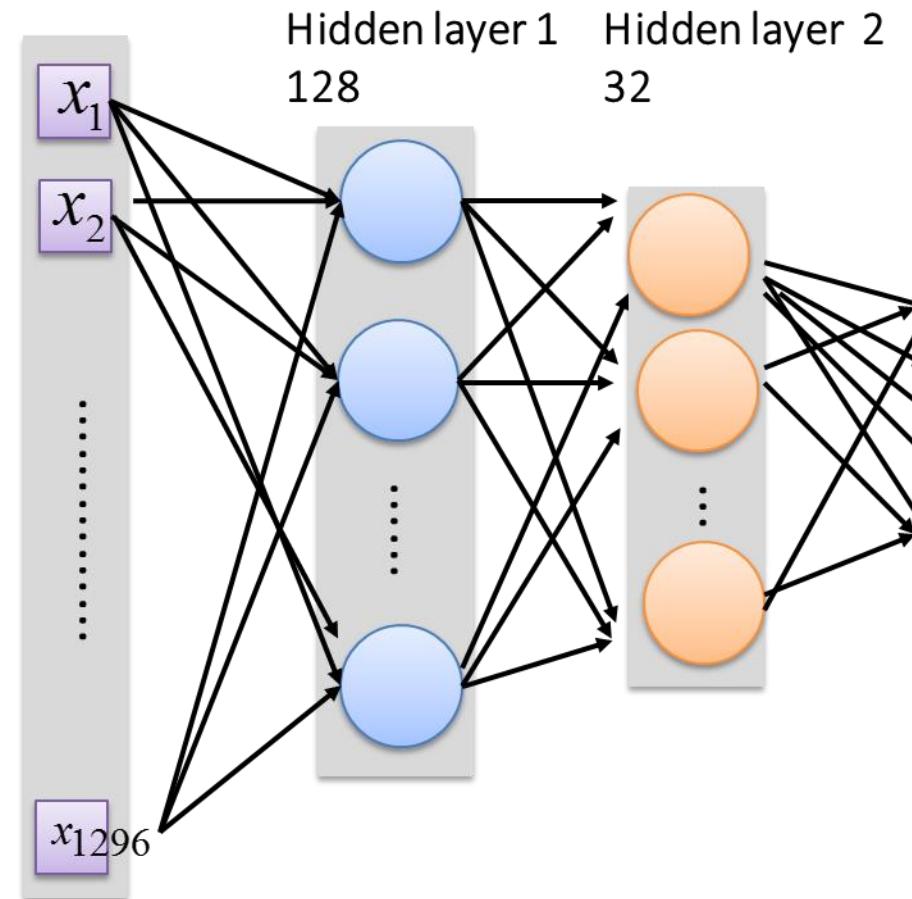
(we need a 1D input)

Input



$$36 \times 36 = 1296$$

reshape



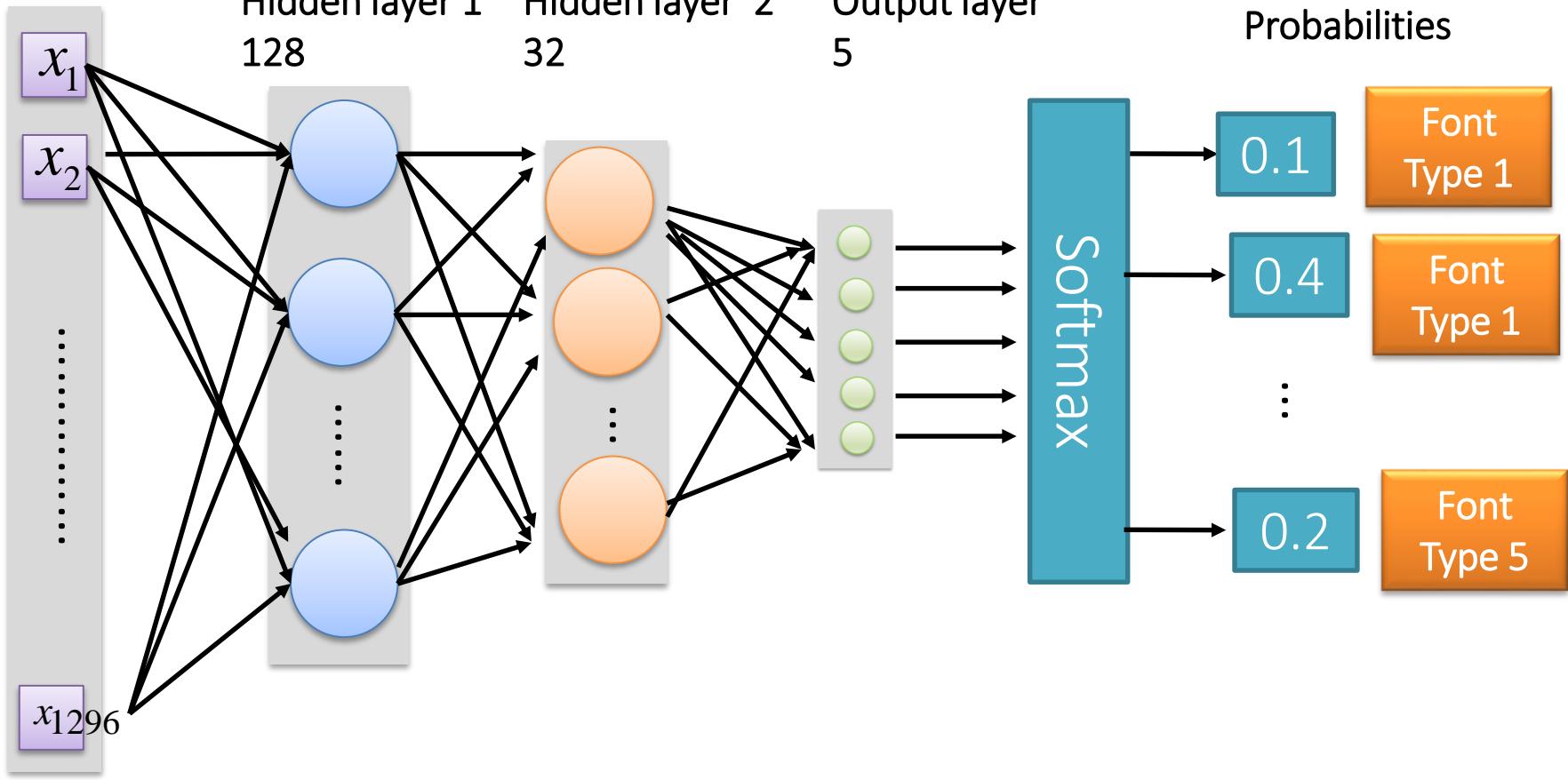
Elements [0:255]

White pixel → 0

Black pixel → 255

PLEASE BE AWARE OF HOW CRITICAL IS TO NORMALIZE  
INPUTS TO Neural Networks

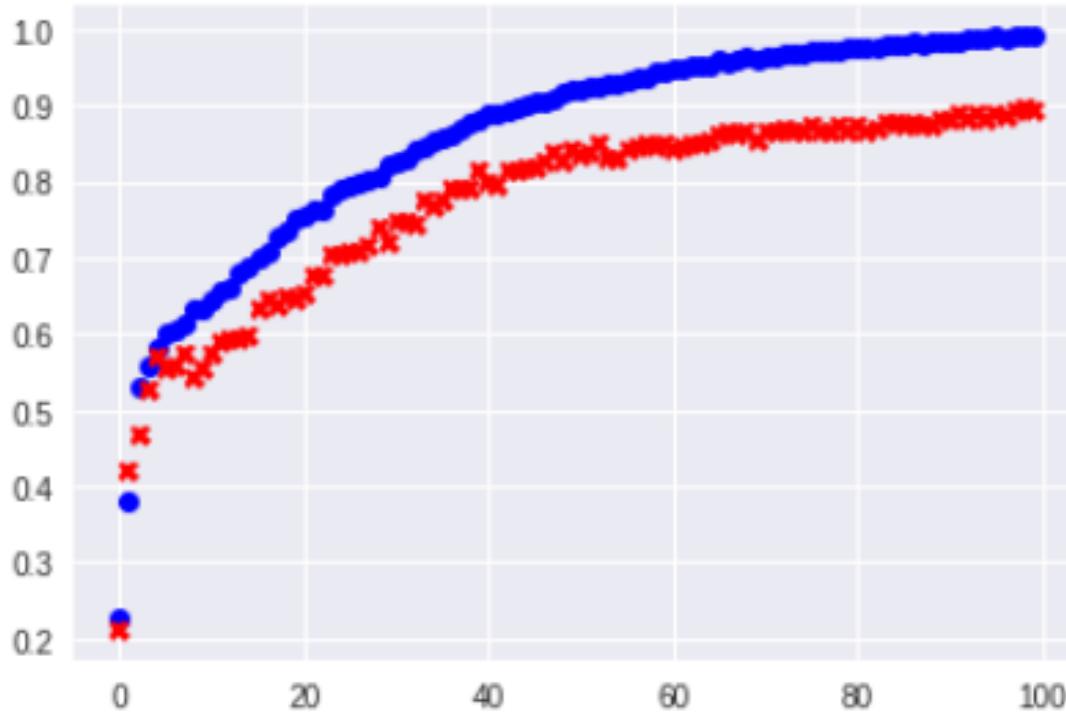
Input  
Image  
Vector



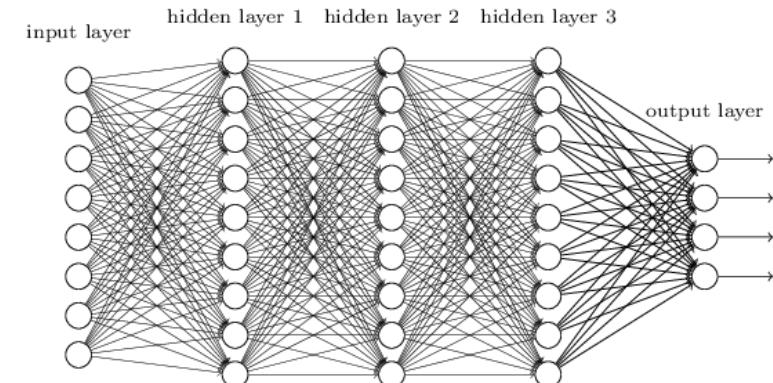
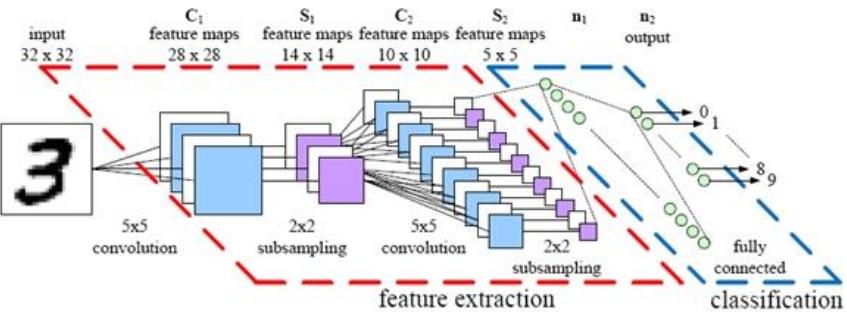
# Feed Forward

# Keras

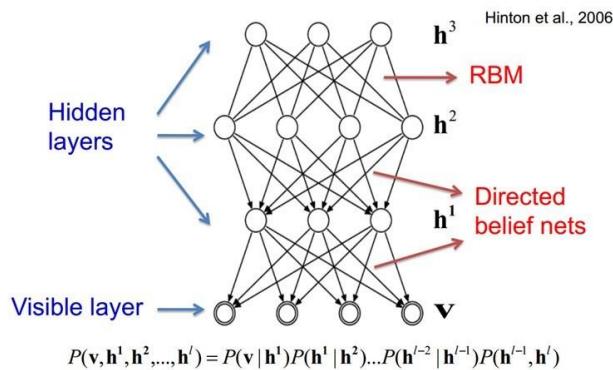
```
Train Accuracy: 0.99  
Test_Accuracy: 0.89  
[<matplotlib.lines.Line2D at 0x7fec4831d2b0>]
```



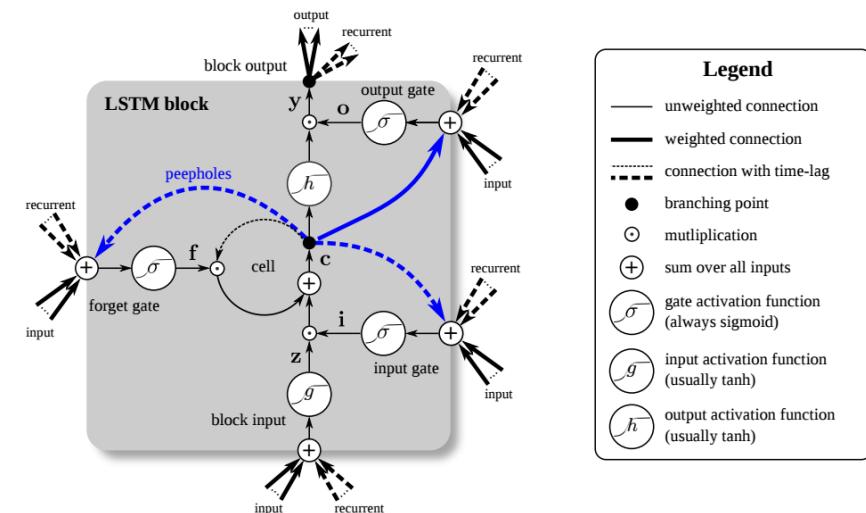
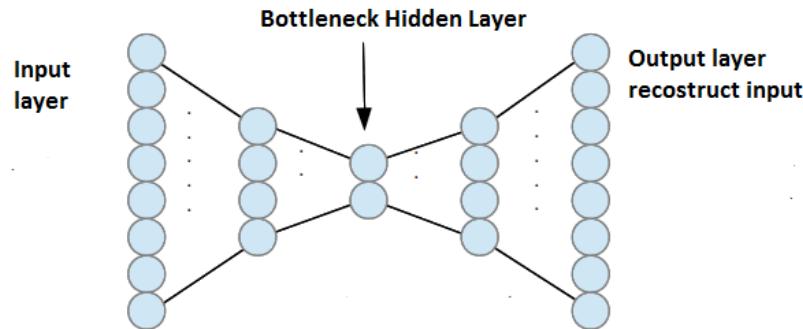
# What's next for Day-2



DBN structure



# Deep Learning Architectures

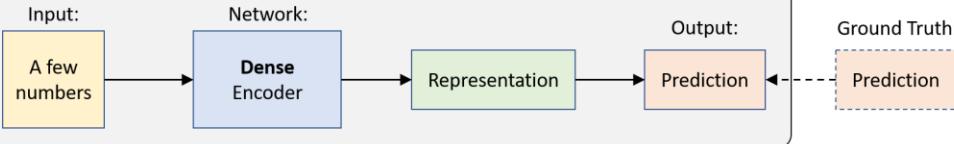


- Legend**
- unweighted connection
  - weighted connection
  - - - connection with time-lag
  - branching point
  - multiplication
  - ⊕ sum over all inputs
  - ( $\sigma$ ) gate activation function (always sigmoid)
  - ( $g$ ) input activation function (usually tanh)
  - ( $h$ ) output activation function (usually tanh)

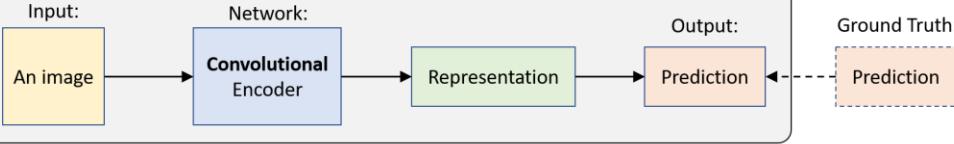
# ... and towards Day-3 : Deep learning basics

## Supervised Learning

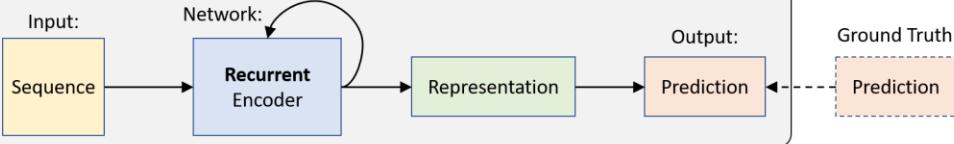
### 1. Feed Forward Neural Networks



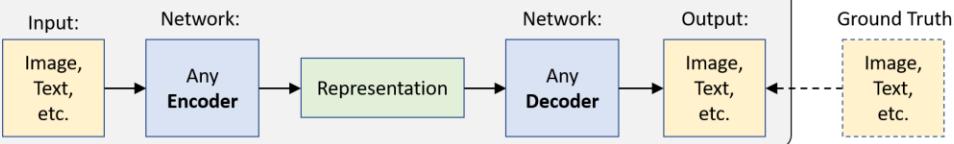
### 2. Convolutional Neural Networks



### 3. Recurrent Neural Networks

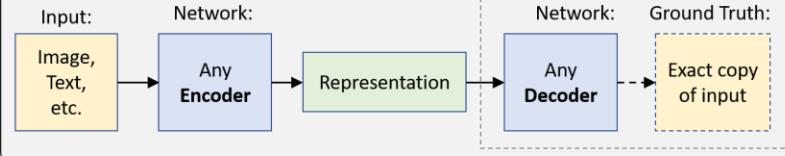


### 4. Encoder-Decoder Architectures

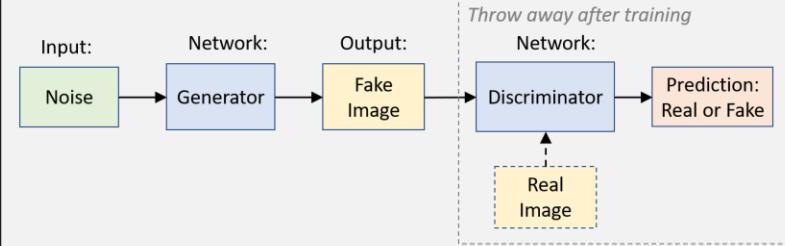


## Unsupervised Learning

### 5. Autoencoder

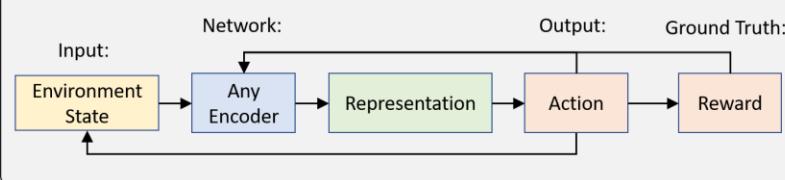


### 6. Generative Adversarial Networks



## Reinforcement Learning

### 7. Networks for Learning Actions, Values, and Policies



[https://github.com/lexfridman/mit-deep-learning/blob/master/tutorial\\_deep\\_learning\\_basics/deep\\_learning\\_basics.ipynb](https://github.com/lexfridman/mit-deep-learning/blob/master/tutorial_deep_learning_basics/deep_learning_basics.ipynb)

# Some on-line courses

## Practical:

- UDACITY: Deep Learning by Google (DNN + TF)
- Udemy : Zero to Deep Learning™ with Python and Keras
- Deep Learning with TensorFlow : **Packt Video** (our examples today from this!)
- TensorFlow and Deep Learning without a PhD, Part 1 (Google Cloud Next '17)  
<https://www.youtube.com/watch?v=u4alGiomYP4>

## Theory:

- Coursera Neural Networks for Machine Learning, as taught by Geoffrey Hinton (University of Toronto)
- Andrew Ng : Deep Learning and Unsupervised Feature Learning