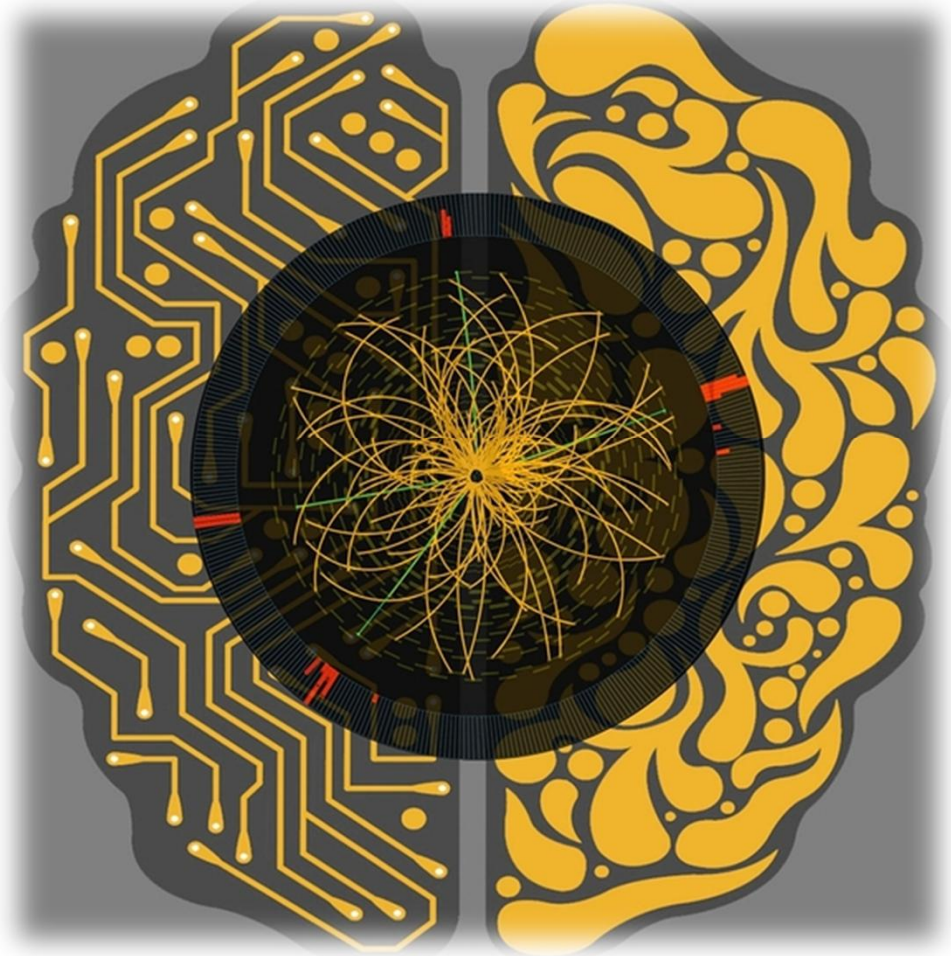


Machine Learning (Introducción)



Latin American alliance for
Capacity buildiNG in Advanced physics
LA-CoNGA physics



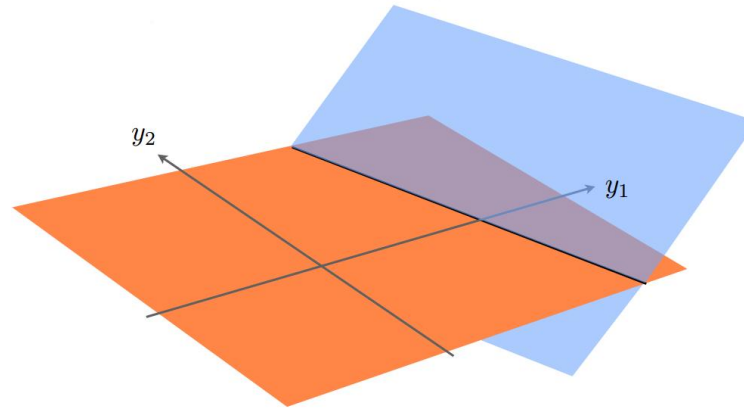
Gabriel Abellán (gabriel.abellan@ciens.ucv.ve)
Laboratorio de Física Teórica, Campos y Partículas
Universidad Central de Venezuela (UCV)



Machine Learning (Introducción)

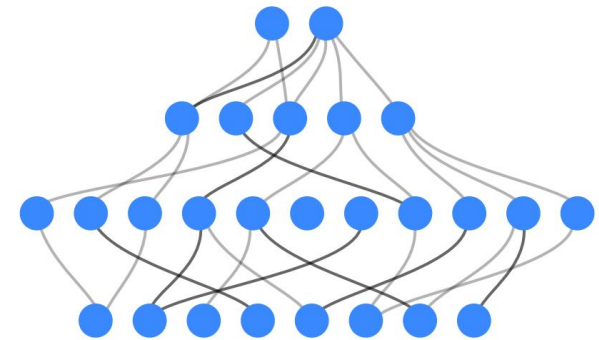
Machine Learning

- ¿Qué es?
- Tipos
- Metodología



Redes Neuronales

- ¿Qué son?
- Tipos
- Ejemplos (KERAS)



Machine Learning



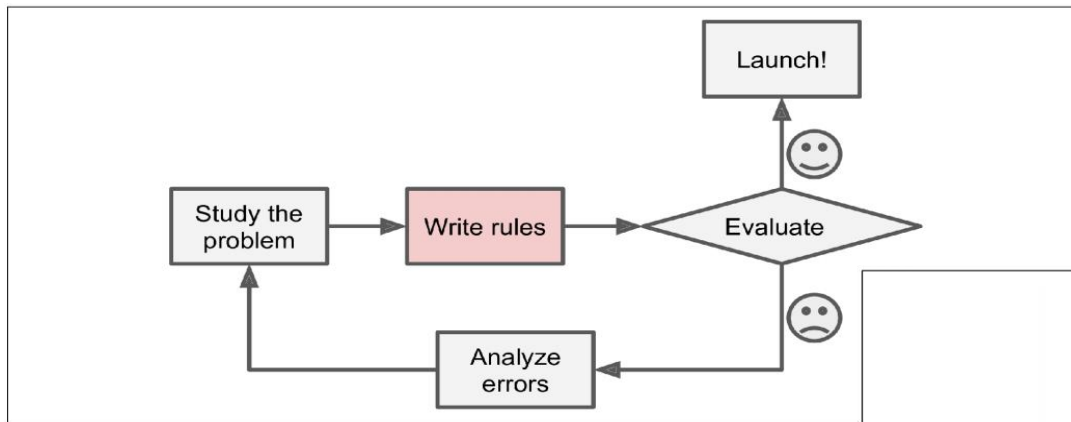
G. Abellán (gabriel.abellan@ciens.ucv.ve)

Machine Learning (Introducción)



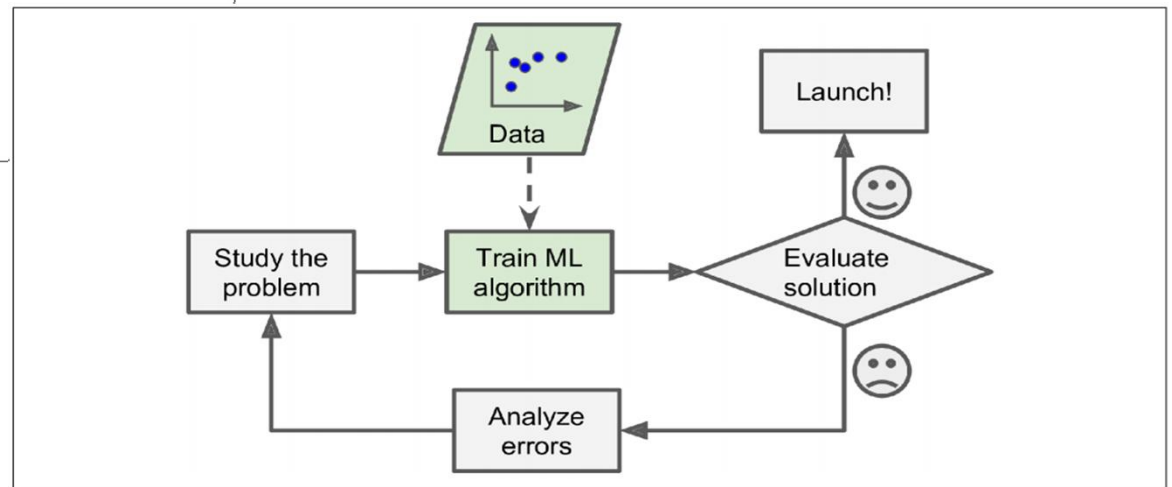
Machine Learning (Intro)

¿Qué es? Es la ciencia (¿arte?) de programar un ordenador y aprenda usando datos



En el approach tradicional debemos determinar reglas para programarlas

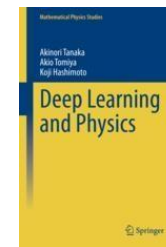
En machine learning se extraen las reglas de los datos



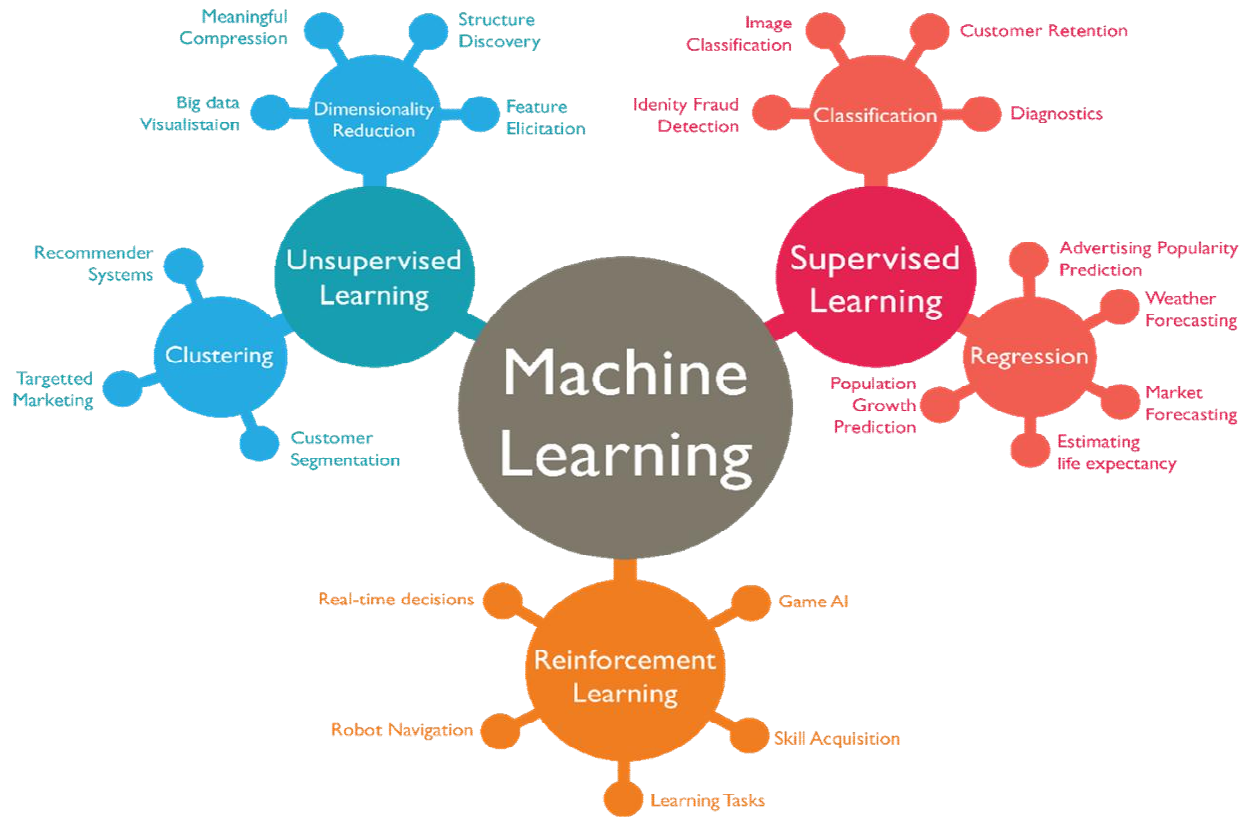
Machine Learning (Intro)

Algunas Aplicaciones

- Clasificación de imágenes
- Diagnóstico de enfermedades
- Clasificación de texto
- Creación de asistentes personales (chatbot)
- Predicción de eventos
- Detección de eventos anómalos
- Extracción de propiedades generales, segmentación de clases
- Sistemas de recomendación
- Bot inteligente para juegos
- Predecir energías de moléculas, orbitales...
- Desarrollo de drogas
- Desarrollo de materiales
- Predicción de estructuras cristalinas
- Descubrimiento de transiciones de fase
- Método para resolver ODEs y PDEs



Machine Learning (Intro)



Tipos de Machine Learning



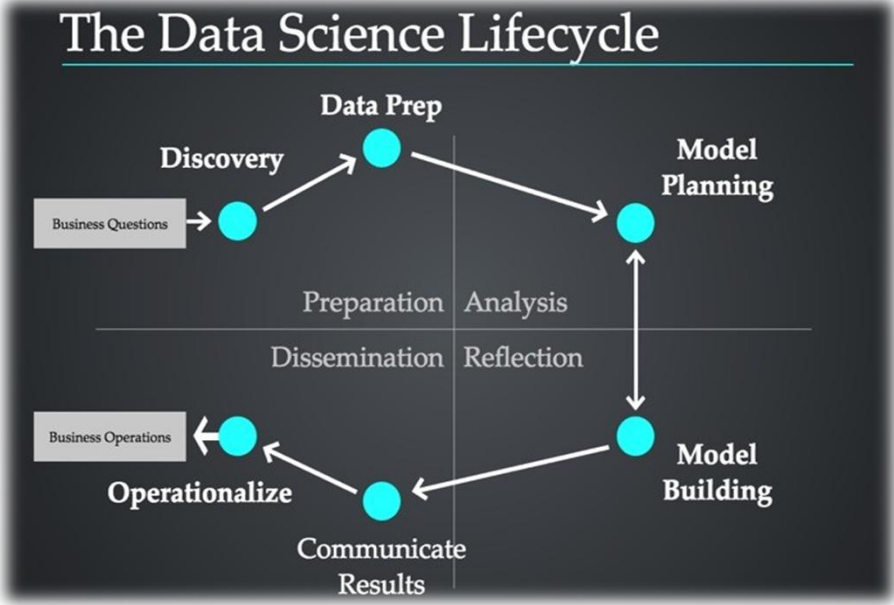
Machine Learning (Intro)



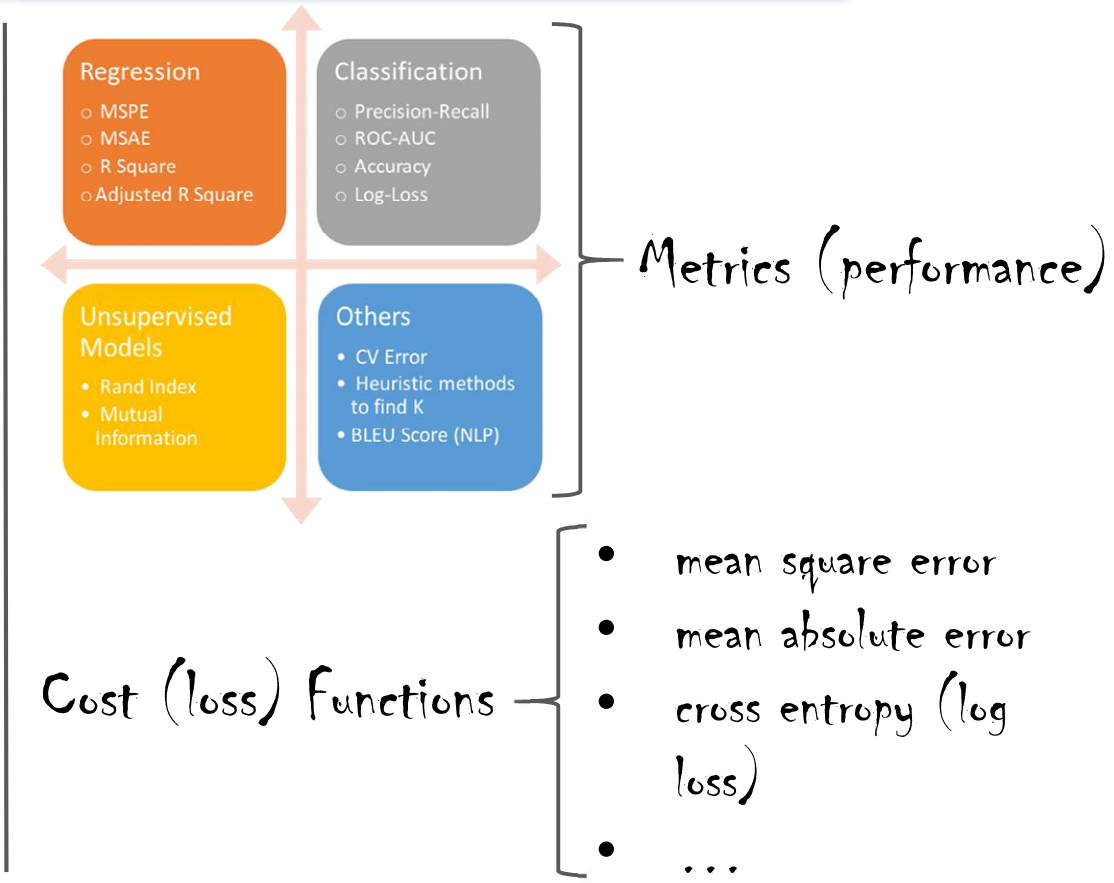
Algoritmos
de Machine
Learning



Machine Learning (Intro)



en cualquier etapa es posible retroceder y replantearse cosas



Machine Learning (Intro)

Para aprender se realiza un proceso de optimización usando la función de costo



Buscar los parámetros que extremen la función costo y maximizan el performance

$$\text{Repeat until convergence } \left\{ \begin{array}{l} \theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \end{array} \right.$$

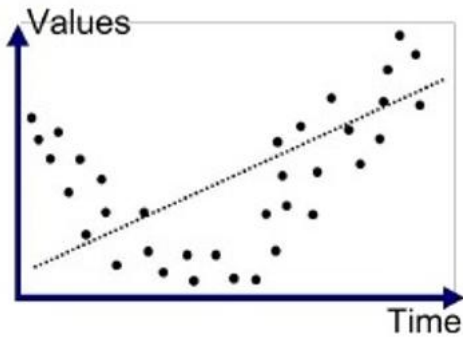
Los algoritmos más simples (lineales) tienen respuestas cerradas

Los algoritmos más poderosos son no lineales y usan métodos iterativos

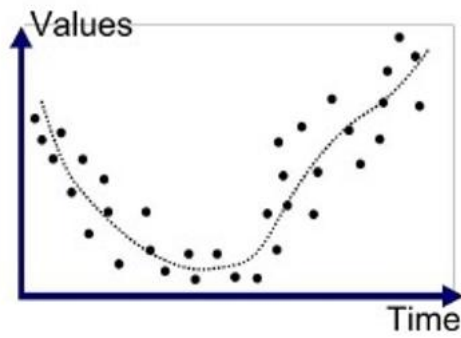
Gradiente Descendente (GD) $\left\{ \begin{array}{l} \bullet \text{ Batch GD} \\ \bullet \text{ Mini Batch GD} \\ \bullet \text{ Stochastic GD} \end{array} \right.$

Machine Learning (Intro)

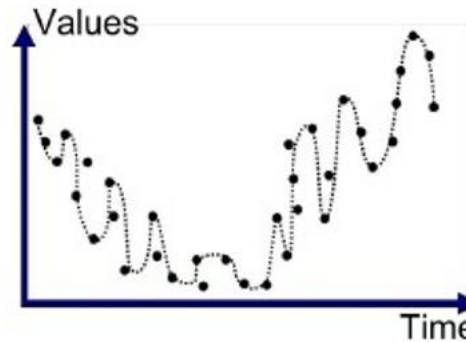
Memorizar no es aprender



Underfitted



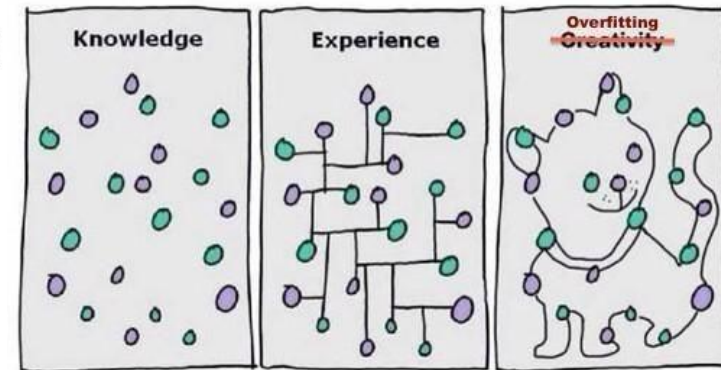
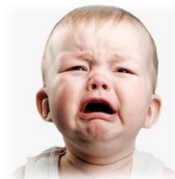
Good Fit/Robust



Overfitted

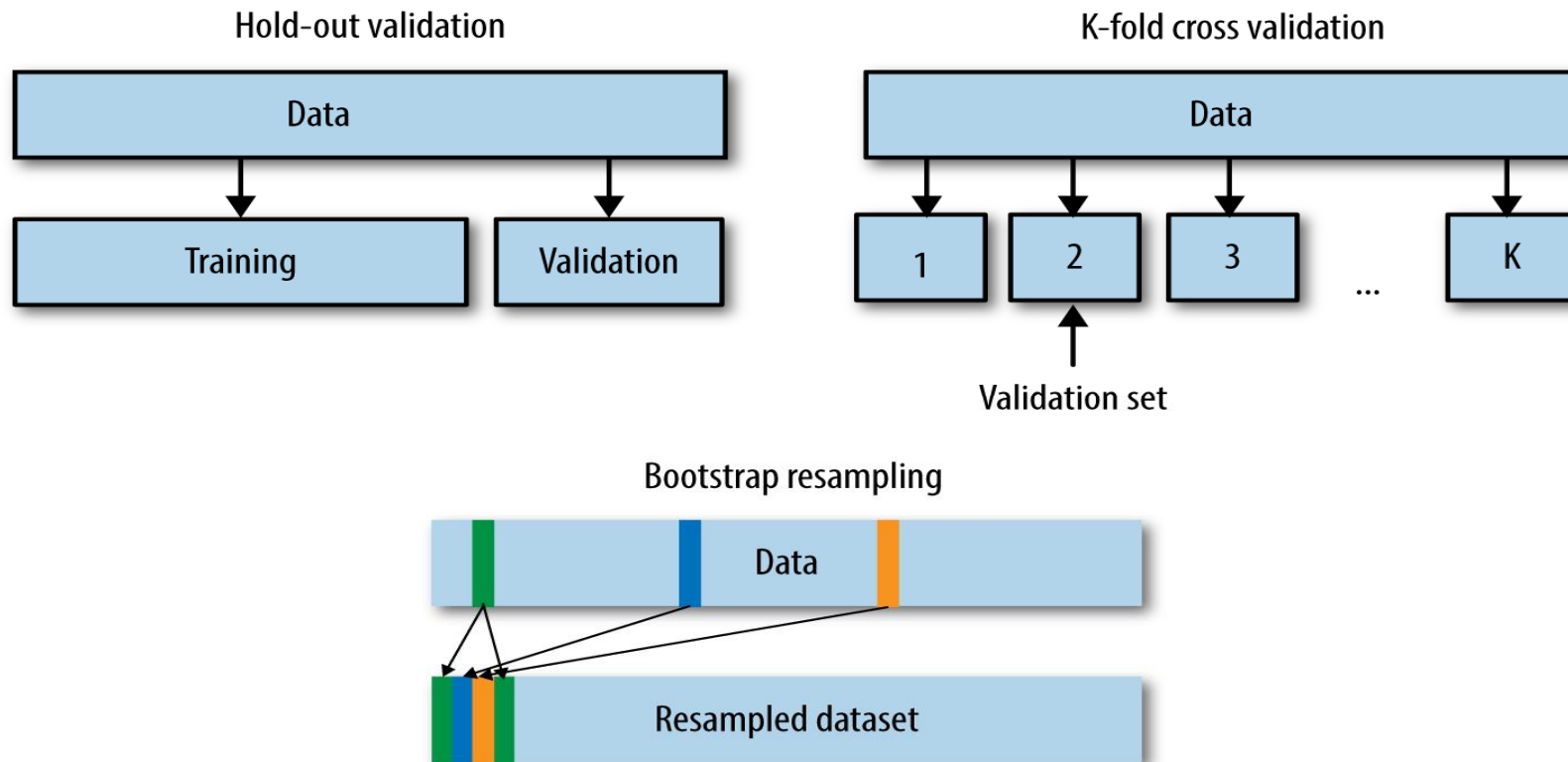
Existe el riesgo de que el modelo se ajuste perfectamente a la data

¡Estos modelos no generalizan bien!



Machine Learning (Intro)

Para evitar ser demasiado optimista, se utiliza un procedimiento para validar el performance del algoritmo




Machine Learning (Intro)

Algunas expresiones e ideas que aparecen en la literatura

Occam's Razor

Si dos modelos tienen igual performance, elegimos el más simple

The Curse of Dimensionality

Al aumentar las variables, la data necesaria para entrenar también aumenta (pero exponencialmente )

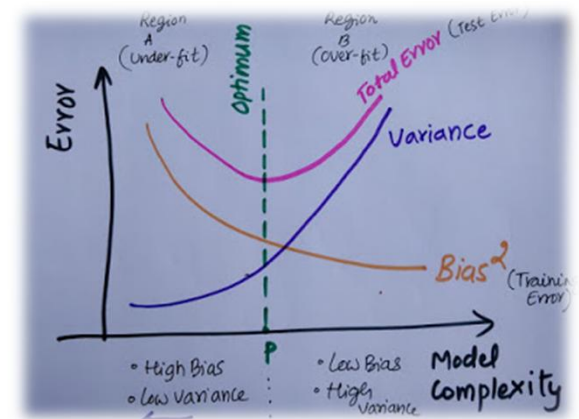
No Free Lunch Theorem

Dos algoritmos son equivalentes cuando se promedia su performance sobre todos los problemas posibles

Es necesario seleccionar el algoritmo para cada problema

Bias-Variance Tradeoff

Es necesario encontrar un compromiso entre la complejidad del modelo y la consistencia en las predicciones



Artificial Neural Network (ANN)



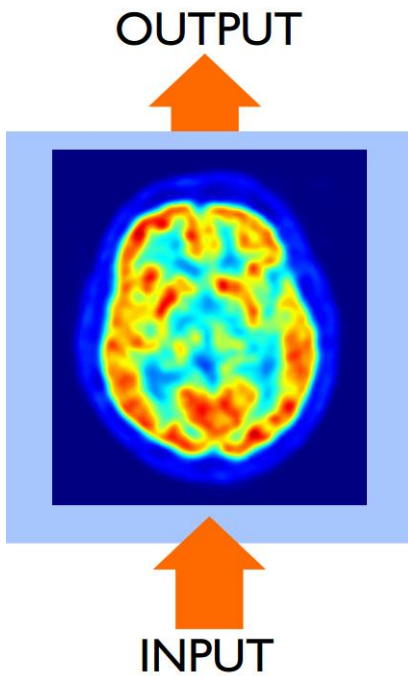
G. Abellán (gabriel.abellan@ciens.ucv.ve)

Machine Learning (Introducción)

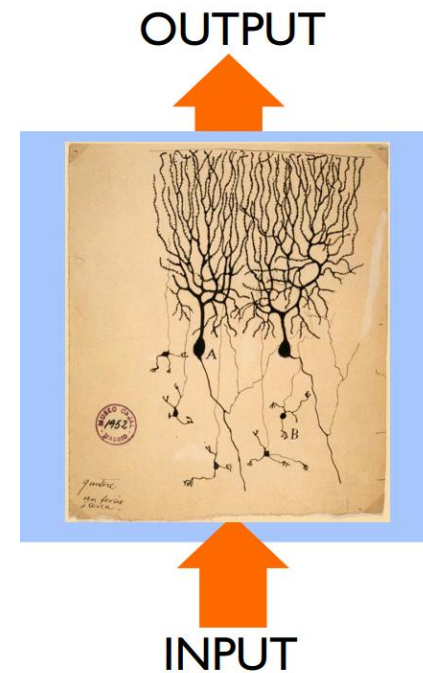


Artificial Neural Network (ANN)

En nuestro cerebro

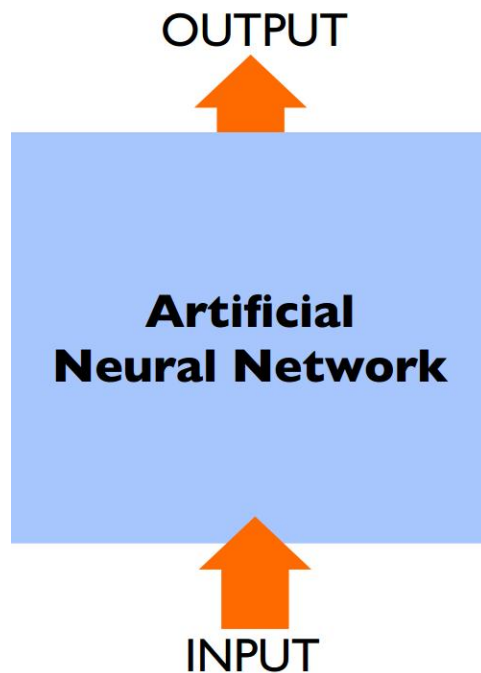


Dibujo de Ramón y Cajal
~1900



Artificial Neural Network (ANN)

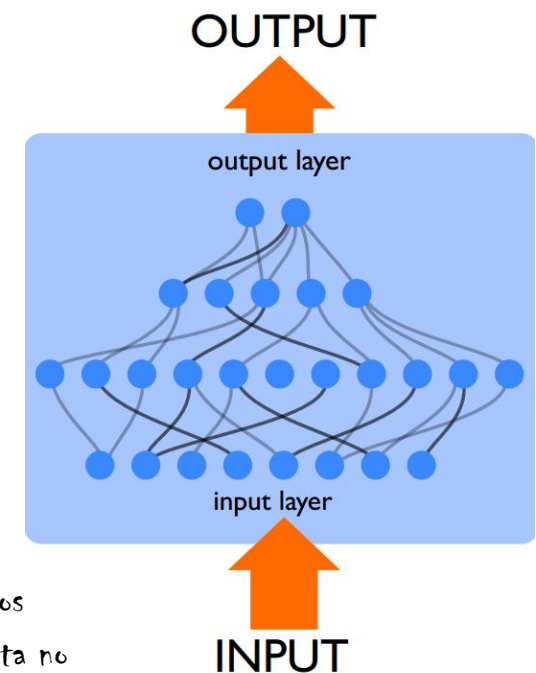
En el ordenador



Inspirada en la forma en que se conectan las neuronas en el cerebro

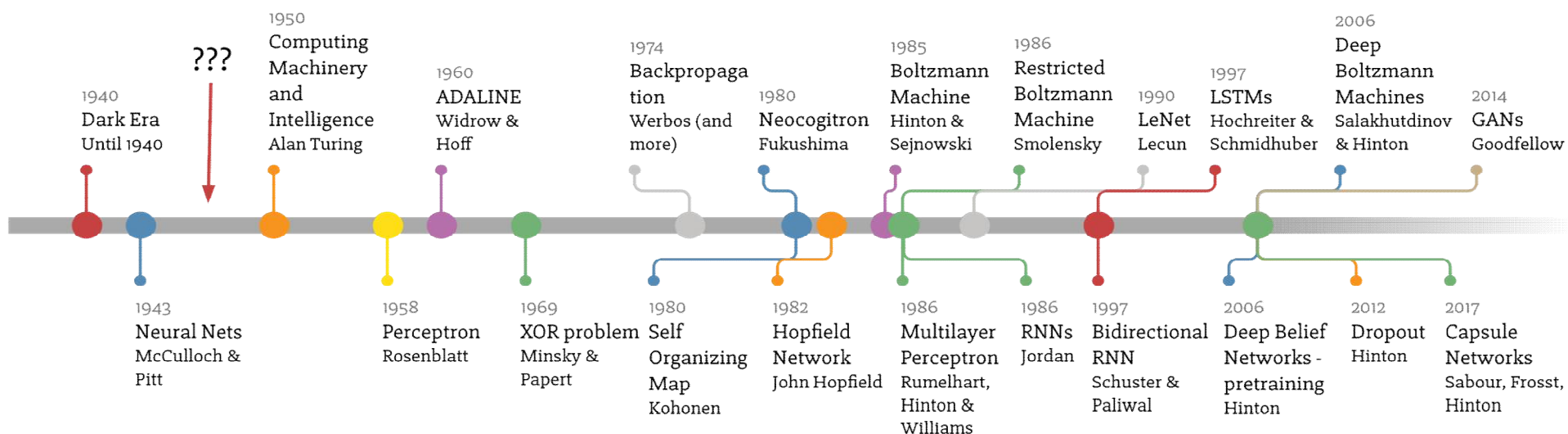
Aprender ~

- extraer características generales
- predecir eventos dados unos datos
- ser flexible para enfrentarse a data no vista con anterioridad
- ...



Artificial Neural Network (ANN)

Deep Learning Timeline



Made by Favio Vázquez



G. Abellán (gabriel.abellan@ciens.ucv.ve)

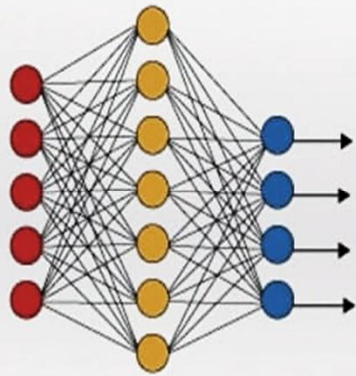
Machine Learning (Introducción)



Artificial Neural Network (ANN)

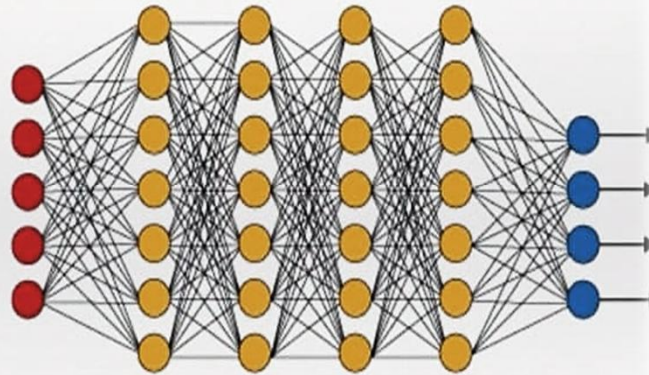
¿Qué es? Una función no lineal que depende de muchos parámetros

Simple Neural Network



● Input Layer ● Hidden Layer ● Output Layer

Deep Learning Neural Network



- Cada conexión tiene un peso w
- Cada neurona tiene un bias b
- Cada neurona tiene una función - no lineal- asociada f

Los valores en la entrada son provistos a la red

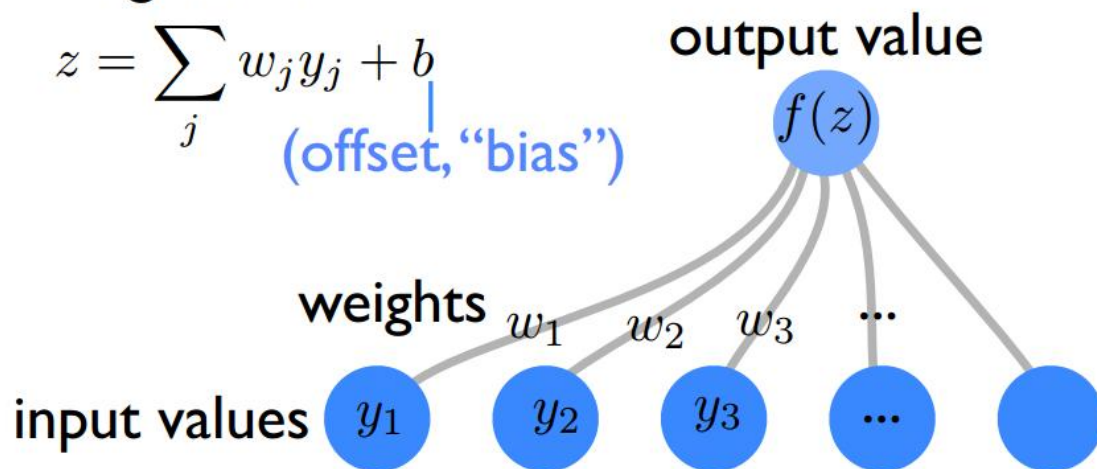
Artificial Neural Network (ANN)

¿Qué es? Una función no lineal que depende de muchos parámetros

weighted sum

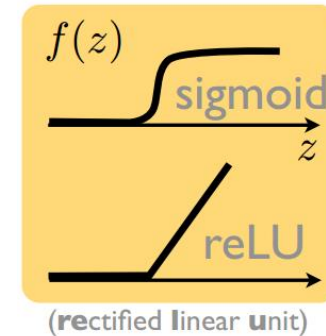
$$z = \sum_j w_j y_j + b$$

(offset, "bias")



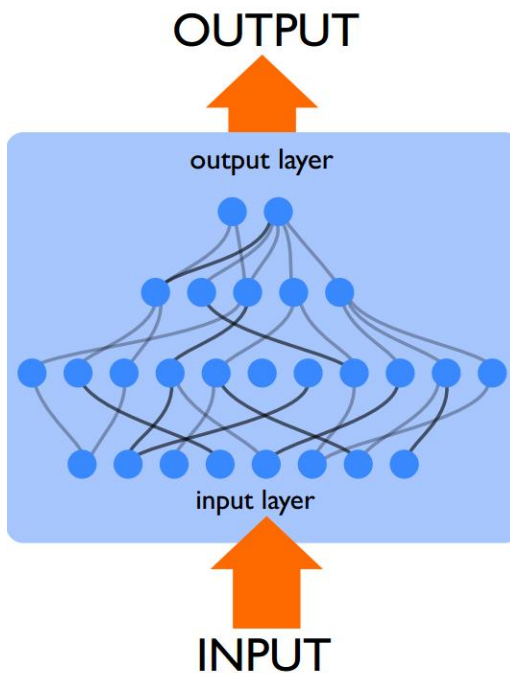
Los valores en la entrada son provistos a la red

- Cada conexión tiene un peso w
- Cada neurona tiene un bias b
- Cada neurona tiene una función - no lineal- asociada f



Artificial Neural Network (ANN)

Neural Networks: Ingredientes



General: feedforward y backpropagation

Específico:

- elegir arquitectura de la red (capas, # de neuronas, funciones de activación, función costo, métricas)
- generar data de bases de datos o colectarla a mano o simularla
- entrenamiento, elegir learning rate y batch size, probar combinaciones

Artificial Neural Network (ANN)

Herramientas de alto nivel

Keras

From the website keras.io

“Keras is a high-level neural networks API, written in Python and capable of running on top of either [TensorFlow](https://www.tensorflow.org/) or [Theano](https://www.theano.org/). It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.”

Sequential: la red usual con capas sucesivas

Dense: capa totalmente conectada

input_shape: neuronas de entrada

loss: cost

SGD: stochastic gradient descent

lr: learning rate (stepsize)

```
from keras import *
from keras.models import Sequential
from keras.layers import Dense
```

Defining a network

layers with 2,150,150,100,1 neurons

```
net=Sequential()
net.add(Dense(150, input_shape=(2,), activation='relu'))
net.add(Dense(150, activation='relu'))
net.add(Dense(100, activation='relu'))
net.add(Dense(1, activation='relu'))
```

'Compiling' the network

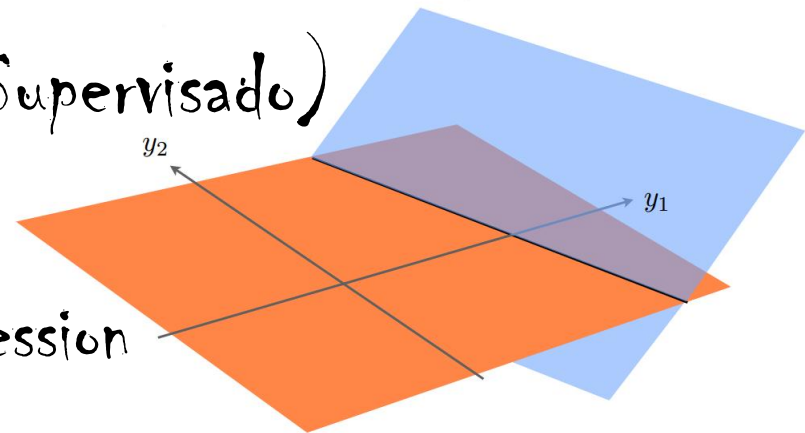
```
net.compile(loss='mean_squared_error',
            optimizer=optimizers.SGD(lr=0.1),
            metrics=['accuracy'])
```



Artificial Neural Network (ANN)

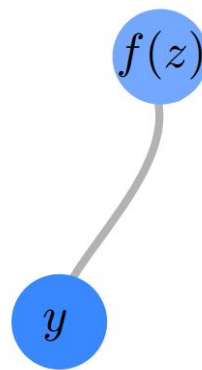
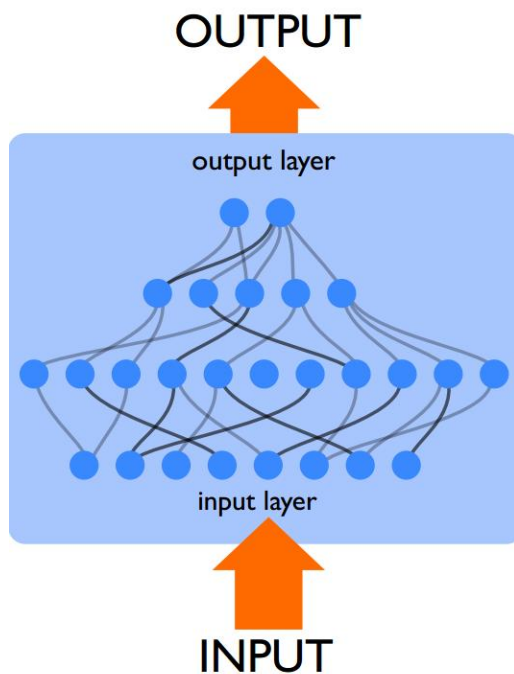
Linear Examples (Supervisado)

Linear Regression



Artificial Neural Network (ANN)

Neural Networks: para regresión lineal



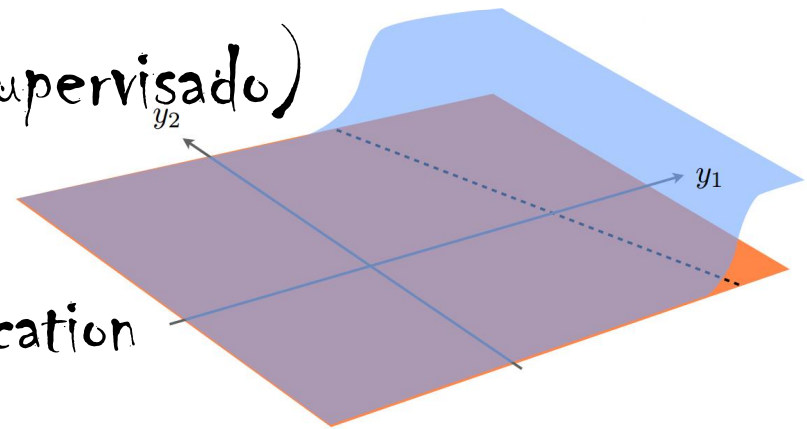
$$z = wy + b$$

Si f es la función identidad se tiene un problema de regresión

Artificial Neural Network (ANN)

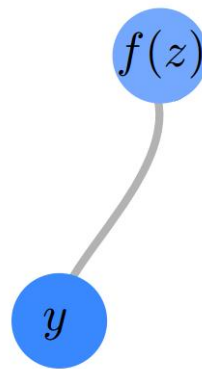
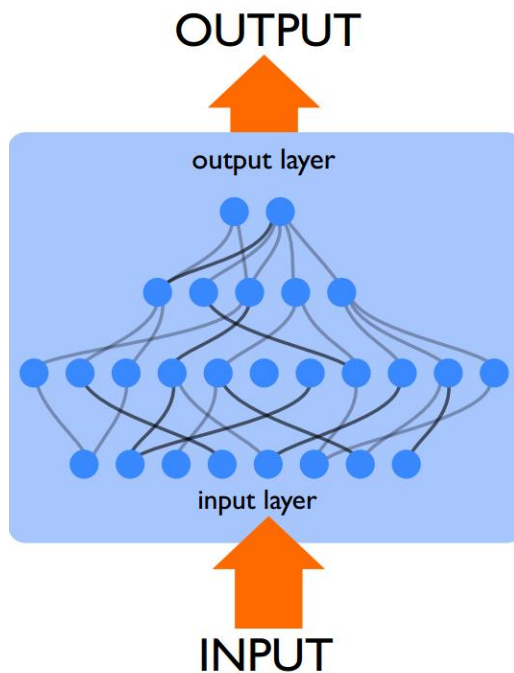
Linear Examples (Supervisado)

Binary Classification



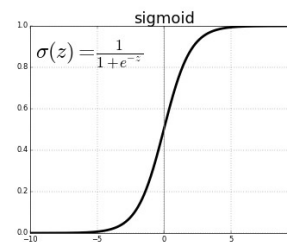
Artificial Neural Network (ANN)

Neural Networks: para clasificación



$$z = wy + b$$

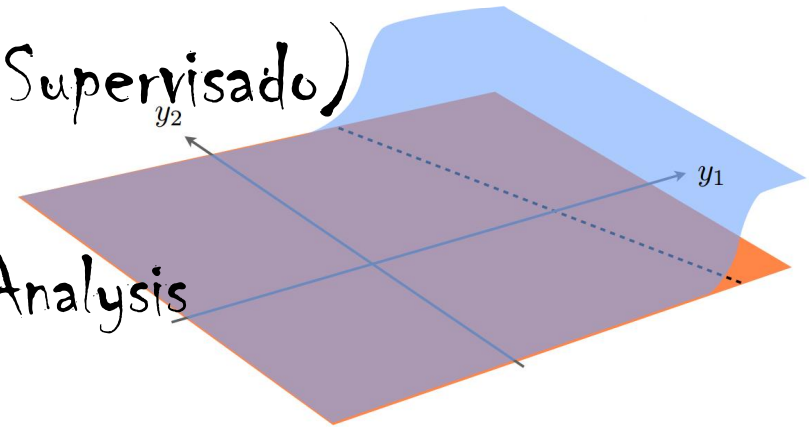
Si f es una sigmoide se tiene un problema de clasificación



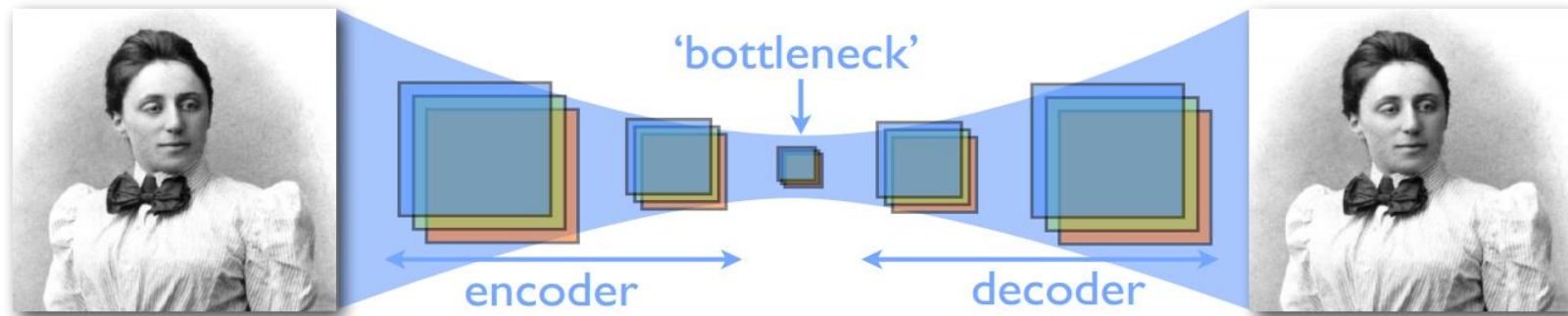
Artificial Neural Network (ANN)

Linear Examples (No Supervisado)

Principal Component Analysis
(PCA)

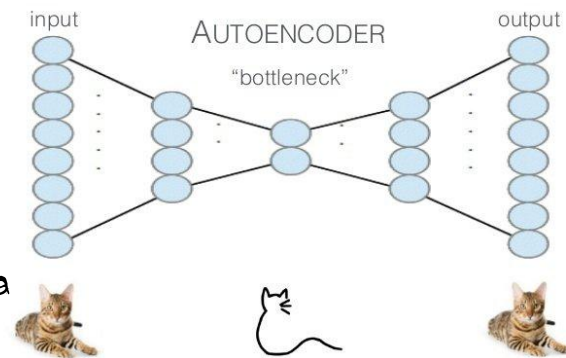


Artificial Neural Network (ANN)

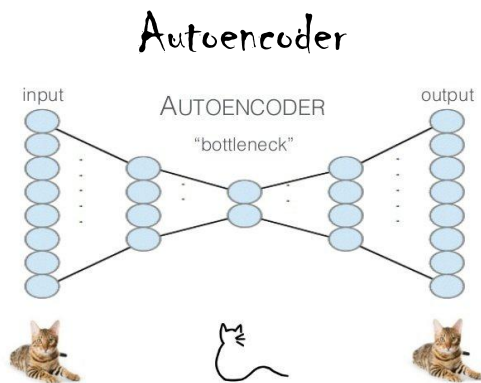


Autoencoder

- reproduce la entrada (input) en la salida (output)
- no requiere ser etiquetado ('respuestas correctas')
- reduce la info de entrada haciéndola pasar por el bottleneck intermedio
- esto funciona si la red aprende a extraer las características esenciales de la data
- funciona como una forma de comprimir data

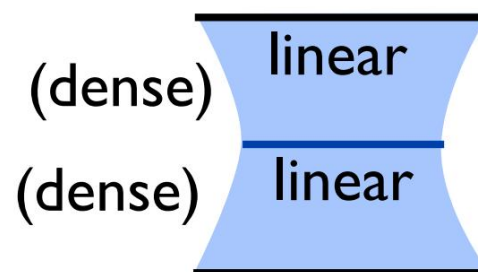


Artificial Neural Network (ANN)



Linear →

Principal Component Analysis



$$|\psi\rangle \approx \hat{P} |\psi\rangle$$

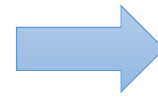
Entrada ↗ ↖ Proyector

$$\hat{P} = \sum_{j=1}^M |v_j\rangle \langle v_j|$$

Proyectar el input en un espacio de menor dimensionalidad y que sea lo más parecido al input mismo

Artificial Neural Network (ANN)

Proyectar el input en un espacio de menor dimensionalidad y que sea lo más parecido al input mismo



En física podemos resolver este problema usando la matriz densidad

$$\hat{\rho} = \langle |\psi\rangle \langle \psi| \rangle = \sum_j p_j \left| \psi^{(j)} \right\rangle \left\langle \psi^{(j)} \right|$$

$$\rho_{mn} = \langle \psi_m \psi_n^* \rangle$$

La matriz densidad es cuadrática en la función de onda, por lo tanto está relacionada con la matriz de covarianza

- Diagonalizar
- Elegir los m autovalores más grandes
- Los autovectores asociados son la base del subespacio buscado



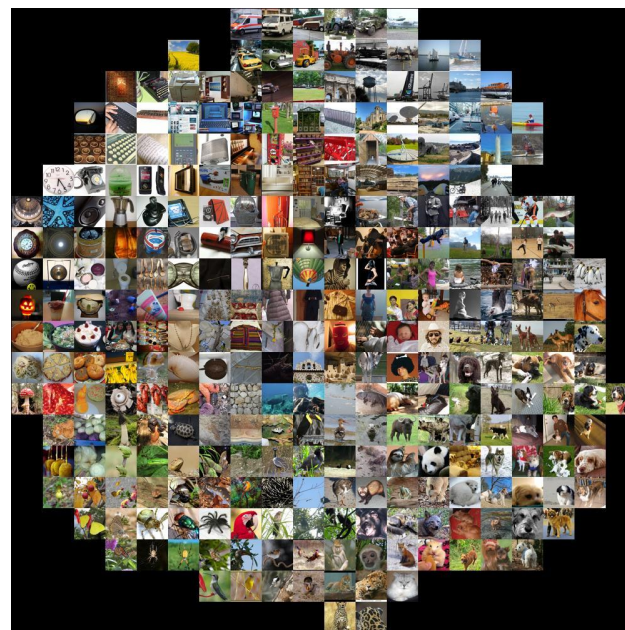
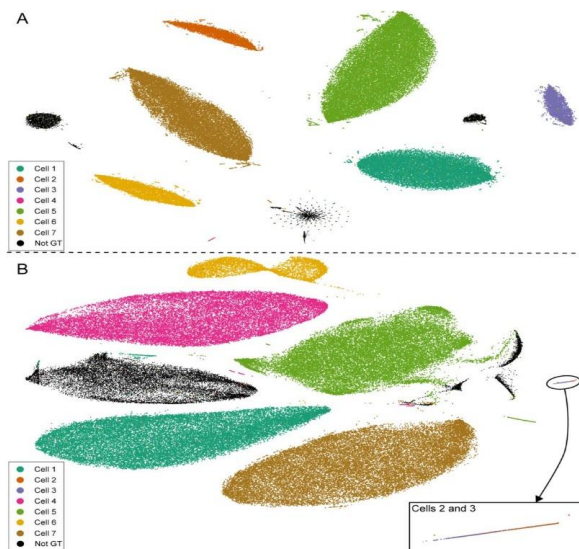
Artificial Neural Network (ANN)

t-SNE (t-distributed Stochastic Neighbor Embedding)



al proyectar, busca preservar las distancias

Algunos ejemplos curiosos



<https://www.biorxiv.org/content/10.1101/087395v1>

<https://cs.stanford.edu/people/karpathy/cnnembed/>



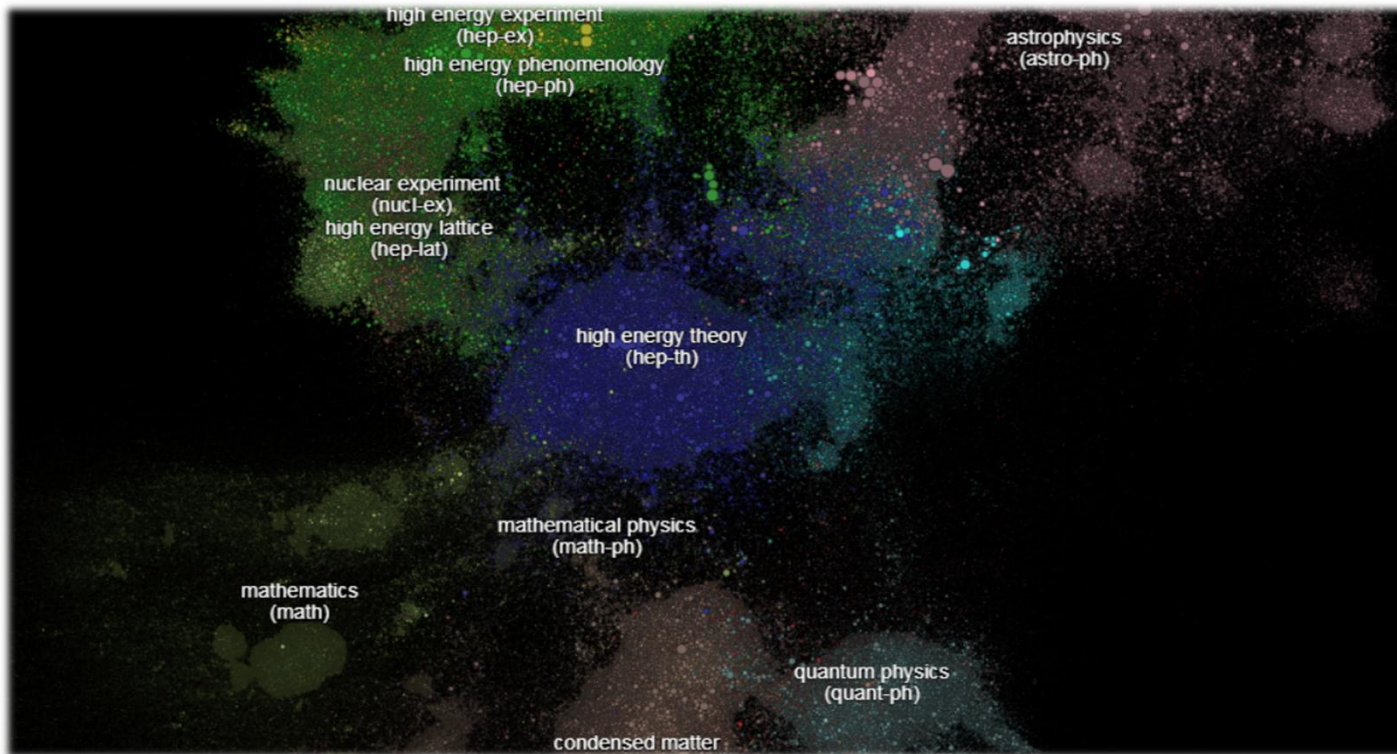
G. Abellán (gabriel.abellan@ciens.ucv.ve)

Machine Learning (Introducción)



Artificial Neural Network (ANN)

t-SNE (t-distributed Stochastic Neighbor Embedding) → al proyectar, busca preservar las distancias



<https://paperscape.org>



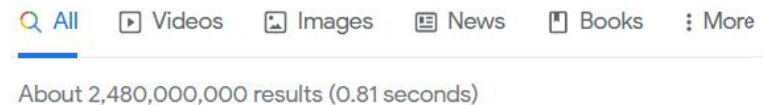
G. Abellán (gabriel.abellan@ciens.ucv.ve)

Machine Learning (Introducción)



Machine Learning (Introducción)

Referencias (en línea) Útiles



Clases en **YouTube** hay muchas, aún siguen siendo muy útiles las clases del prof. **Yaser Abu-Mostafa** en *Caltech* <https://www.youtube.com/playlist?list=PLD63A284B7615313A>

Como lectura puedes revisar

- A high-bias, low-variance introduction to Machine Learning for physicists (excelente introducción en nuestro *argot* <https://arxiv.org/abs/1803.08823>)
- An overview of gradient descent optimization algorithms (<https://arxiv.org/abs/1609.04747>)
- Neural Networks and Deep Learning (<https://neuralnetworksanddeeplearning.com/>)
- Deep Learning (<https://www.deeplearningbook.org>)
- Machine Learning for Physicist (<https://machine-learning-for-physicists.org/>)
- Machine Learning Forum, Training, Competitions (<https://www.kaggle.com/>)

