Inteligencia artificial

Camilo Menares
camilo.menares@uchile.cl
Departamento de Geofísica FCFM - Universidad de Chile
Centro de ciencia del clima y resiliencia

Francisco Gomez
frangomez@ug.uchile.cl
Departamento de Geofísica FCFM - Universidad de Chile

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

(CR)² Center for Climate and Resilience Research

Contenidos

NLHPC National Laboratory for High Performance Computing Chile

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

(CR)² Center for Climate and Resilience Research

Módulo 1

Motivación e Introducción a la IA

NLHPC National Laboratory for High Performance Computing Chile

Motivación

Go, juego de tablero Chino

- 5 veces más grandes que el ajedrez (37 jugadas por turno vs mínimo 61 jugadas por turno en GO)
- Anticipar 10 jugadas requiere 512 quintillones de combinaciones a predecir
- 512 quintillones de combinaciones toma a “Tianhe-2”, 4 horas de cálculo

Motivación

AlphaGo vs Lee Sedol

Deep learning vs Campeón mundial

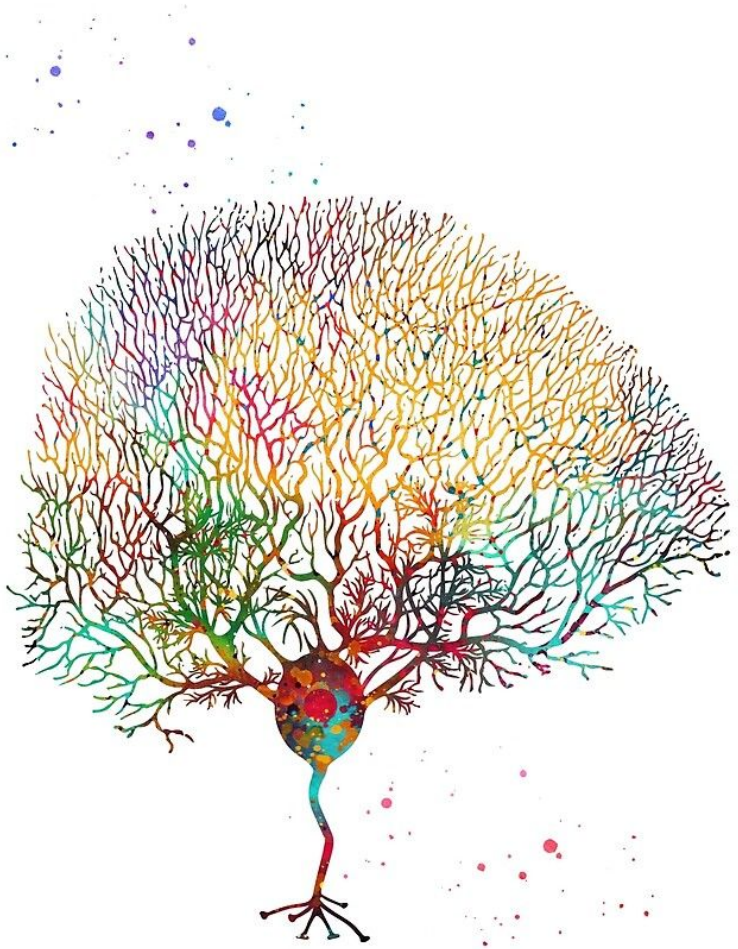
Resultado

AlphaGo 4 – 1 Lee Sedol

El árbitro del partido Toby Manning ha descrito el estilo del programa como «conservador»

Myungwan Kim «es como jugar contra un ser humano»

Machine Learning



www.redbubble.com

"Machine Learning es la ciencia que permite que las computadoras aprendan y actúen como lo hacen los humanos, mejorando su aprendizaje a lo largo del tiempo de una forma autónoma, alimentándose con datos e información en forma de observaciones e interacciones con el mundo real." — Dan Fagella

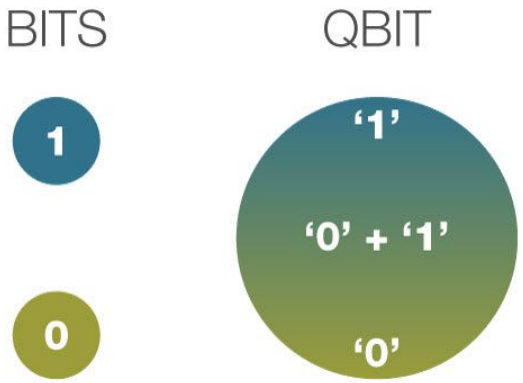
Supremacía cuántica de Google

Motivación

(CR)² Center for Climate and Resilience Research

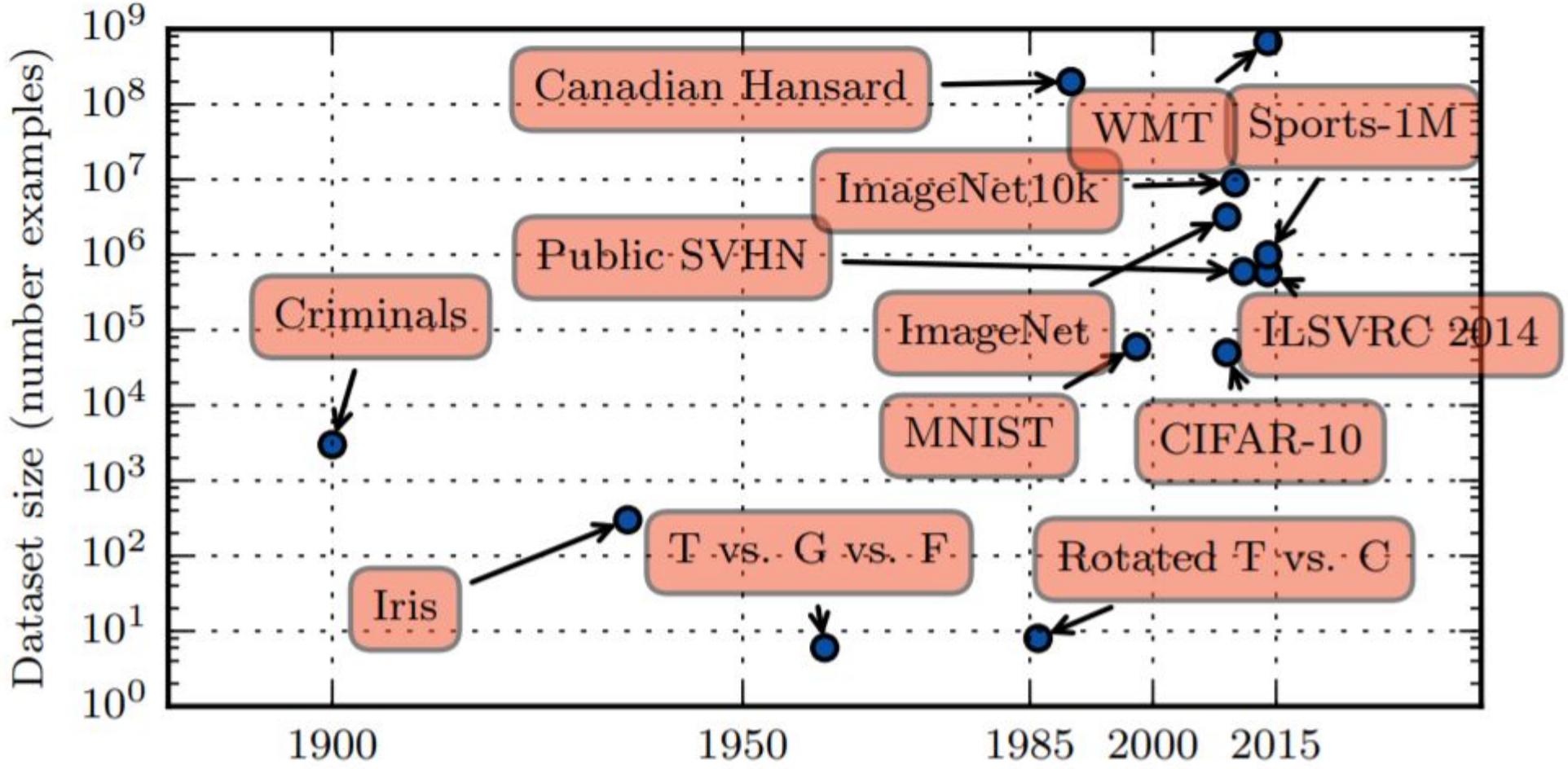
De los bits a los qubit

Bits dos estados (0 y 1)
Qubit tres estados (0, 1, <0,1>)

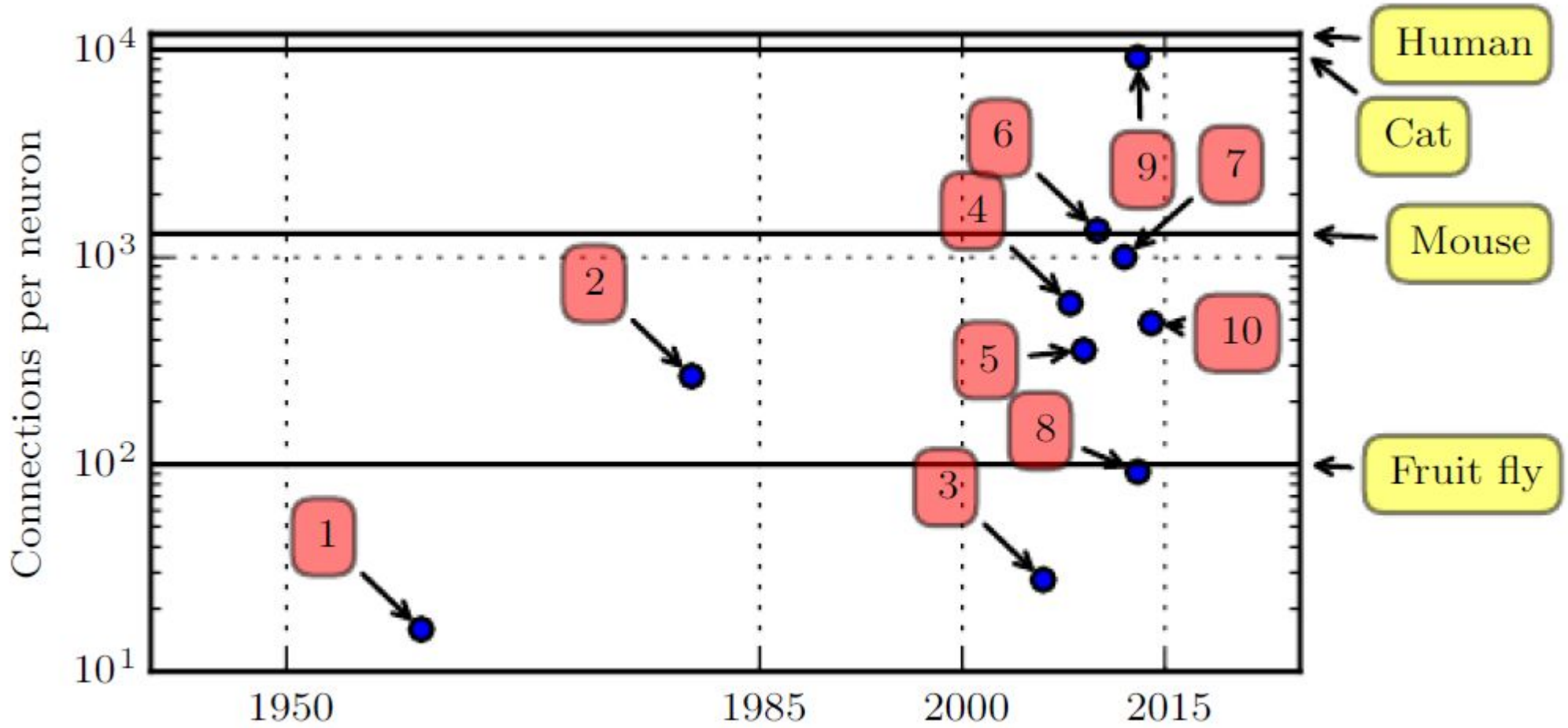


Google desarrolla su computadora cuántica para acelerar las tareas computacionales de Machine Learning

Más Neuronas, Más cerebro



Más Neuronas, Más cerebro



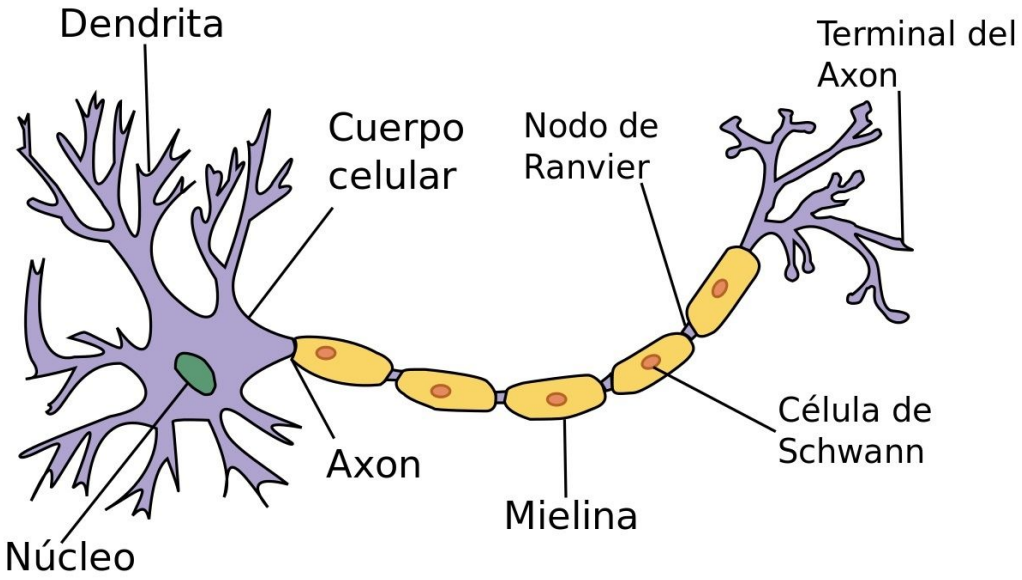
1. Adaptive linear element (Widrow and Hoff, 1960) ;
2. Neocognitron (Fukushima, 1980)
3. GPU-accelerated convolutional network (Chellapilla et al., 2006)
4. Deep Boltzmann machine (Salakhutdinov and Hinton, 2009a)
5. Unsupervised convolutional network (Jarrett et al., 2009)
6. GPU-accelerated multilayer perceptron (Ciresan et al., 2010)
7. Distributed autoencoder (Le et al., 2012)
8. Multi-GPU convolutional network (Krizhevsky et al., 2012)
9. COTS HPC unsupervised convolutional network (Coates et al., 2013)
10. GoogLeNet (Szegedy et al., 2014a)

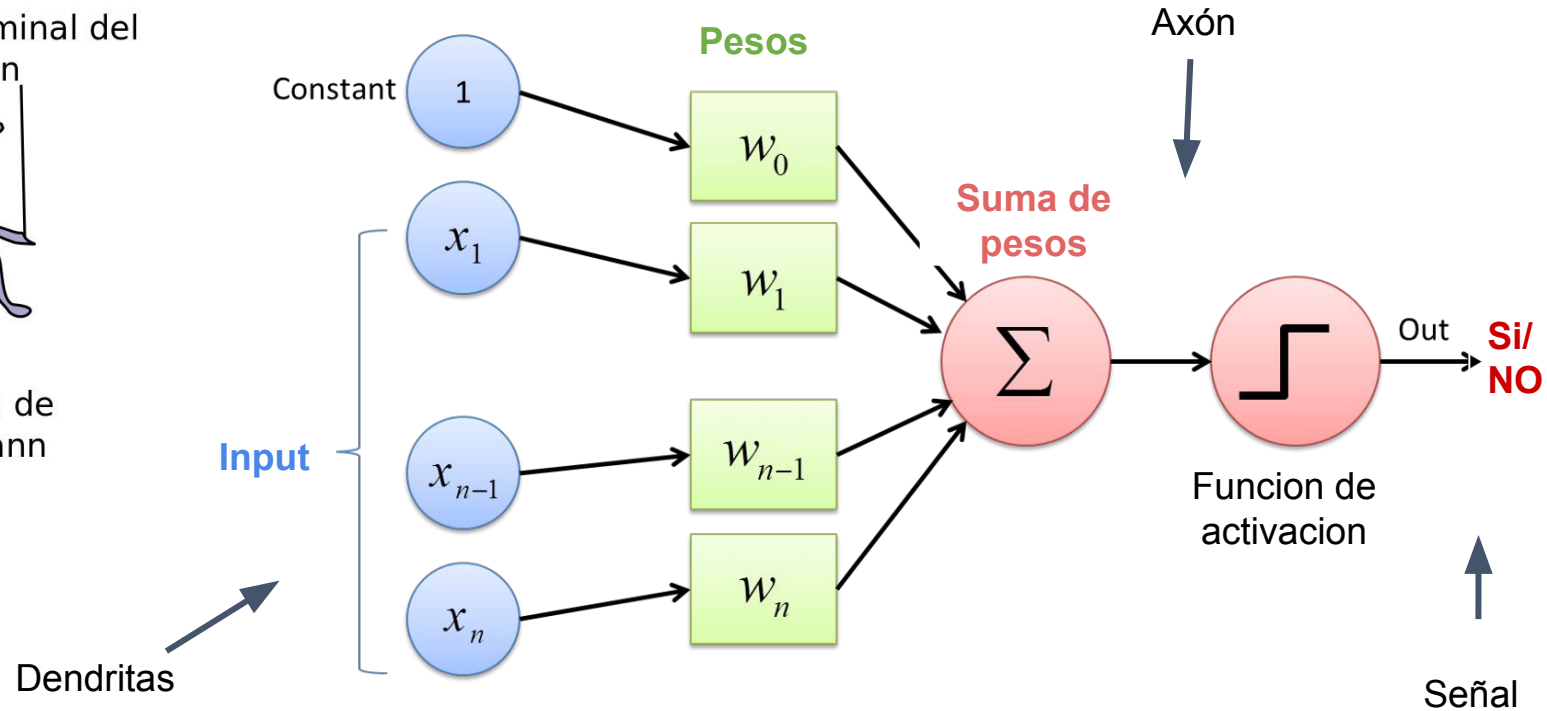
McCulloch & Pitts

Machine Learning

Perceptrón

Def. Unidad básica para la transmisión de información, la célula computacional.
McCulloch & Pitts / Rosenblatt 1954



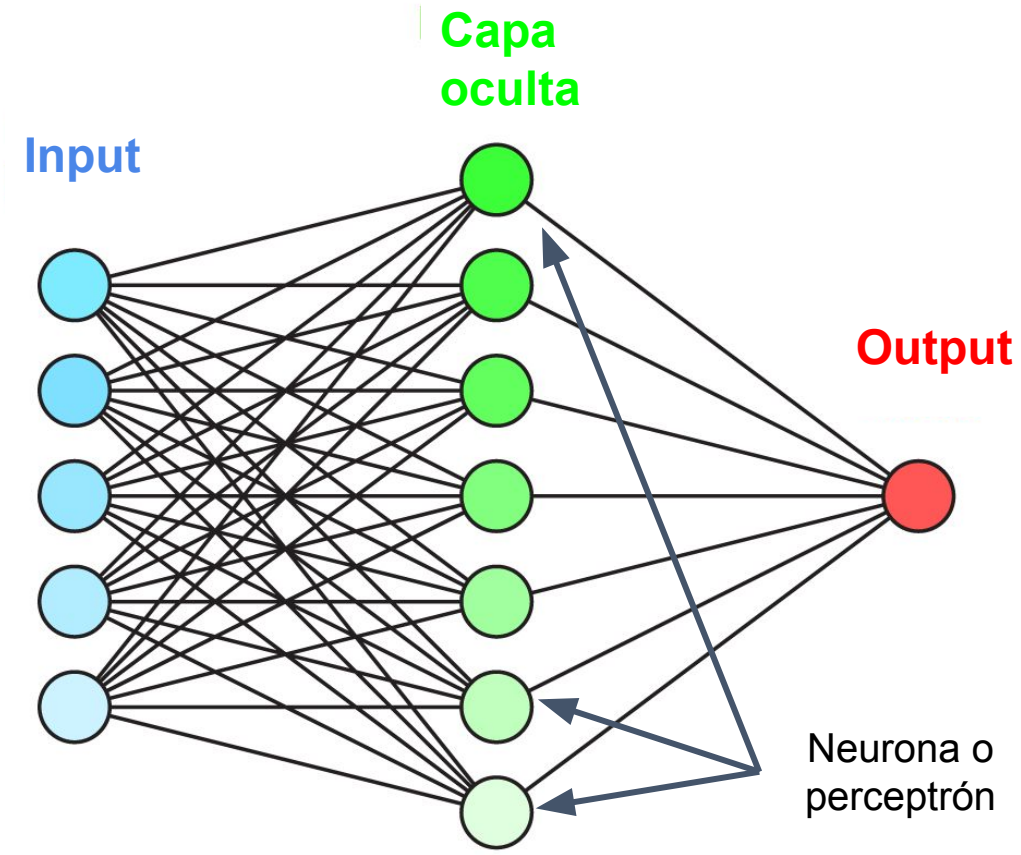


McCulloch & Pitts

Machine Learning

Perceptrón Multicapa

Def. Arquitectura neuronal, compuesta de varios o múltiples capas de neuronas (perceptrones) capaz de resolver problemas de no linealidad.

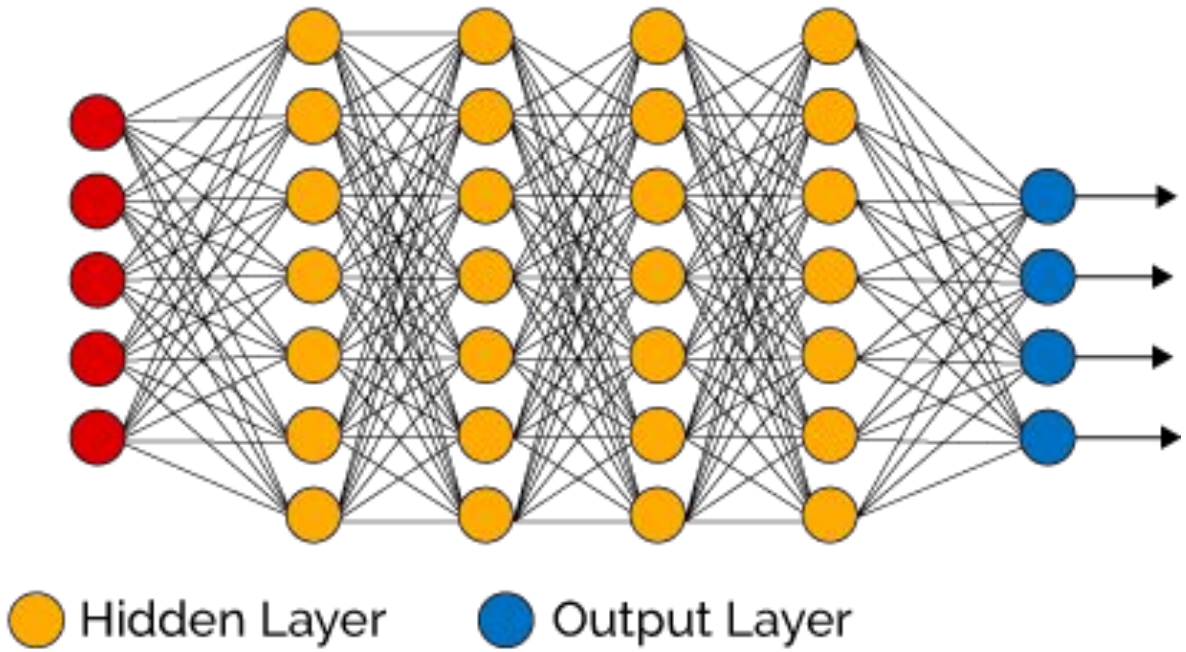


McCulloch & Pitts

Machine Learning

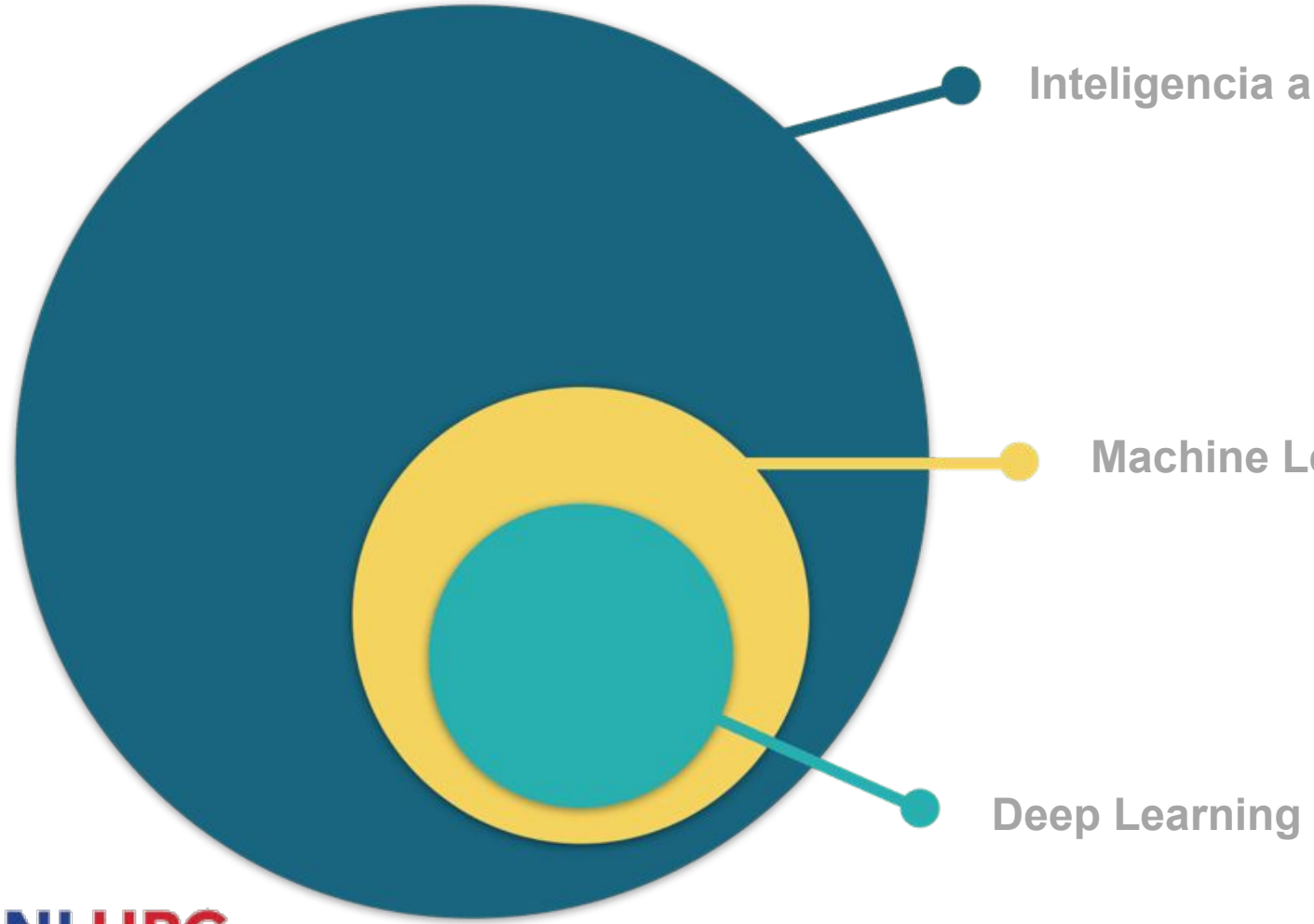
Perceptrón Multicapa Deep learning

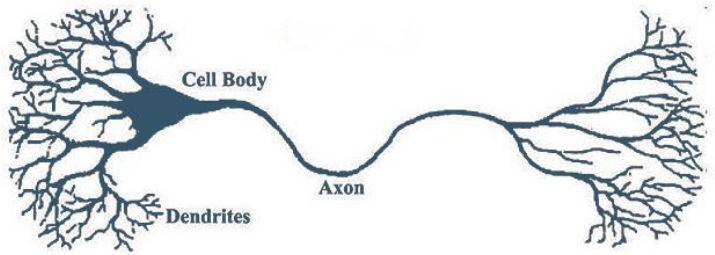
Def. Arquitectura neuronal, compuesta de varios o múltiples capas de neuronas (perceptrones) y muchas capas ocultas (profundas) capaces de resolver problemas de no linealidad.

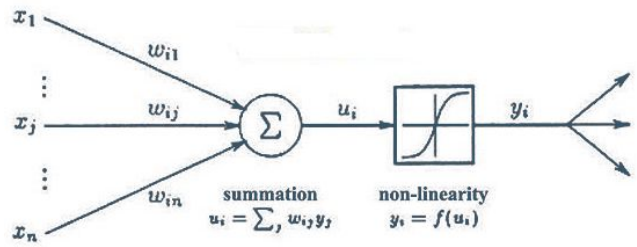


Machine Learning vs Deep Learning

(CR)² Center for Climate and Resilience Research







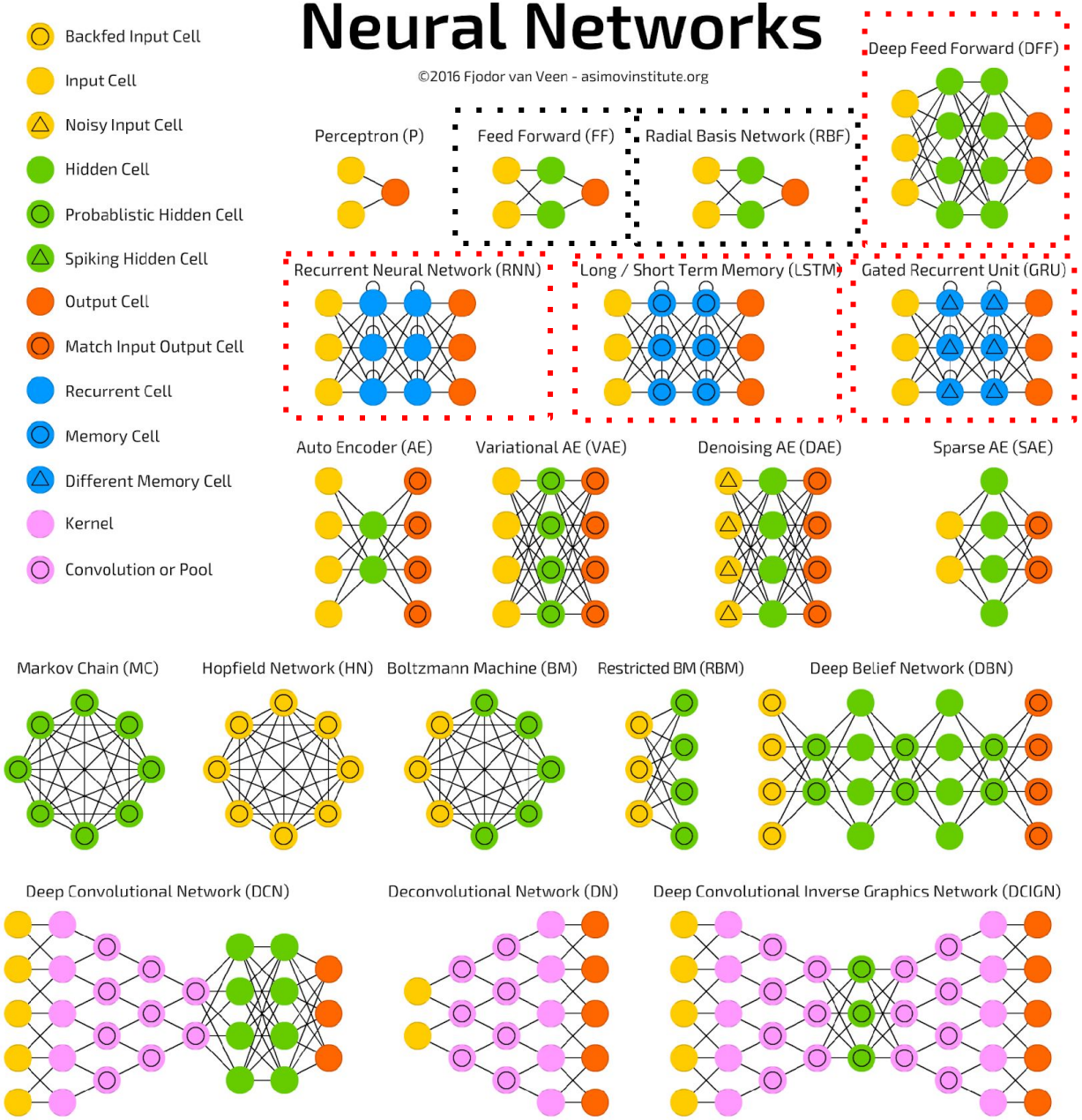
NLHPC National Laboratory for High Performance Computing Chile

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Center for Climate and Resilience Research

A mostly complete chart of

Neural Networks

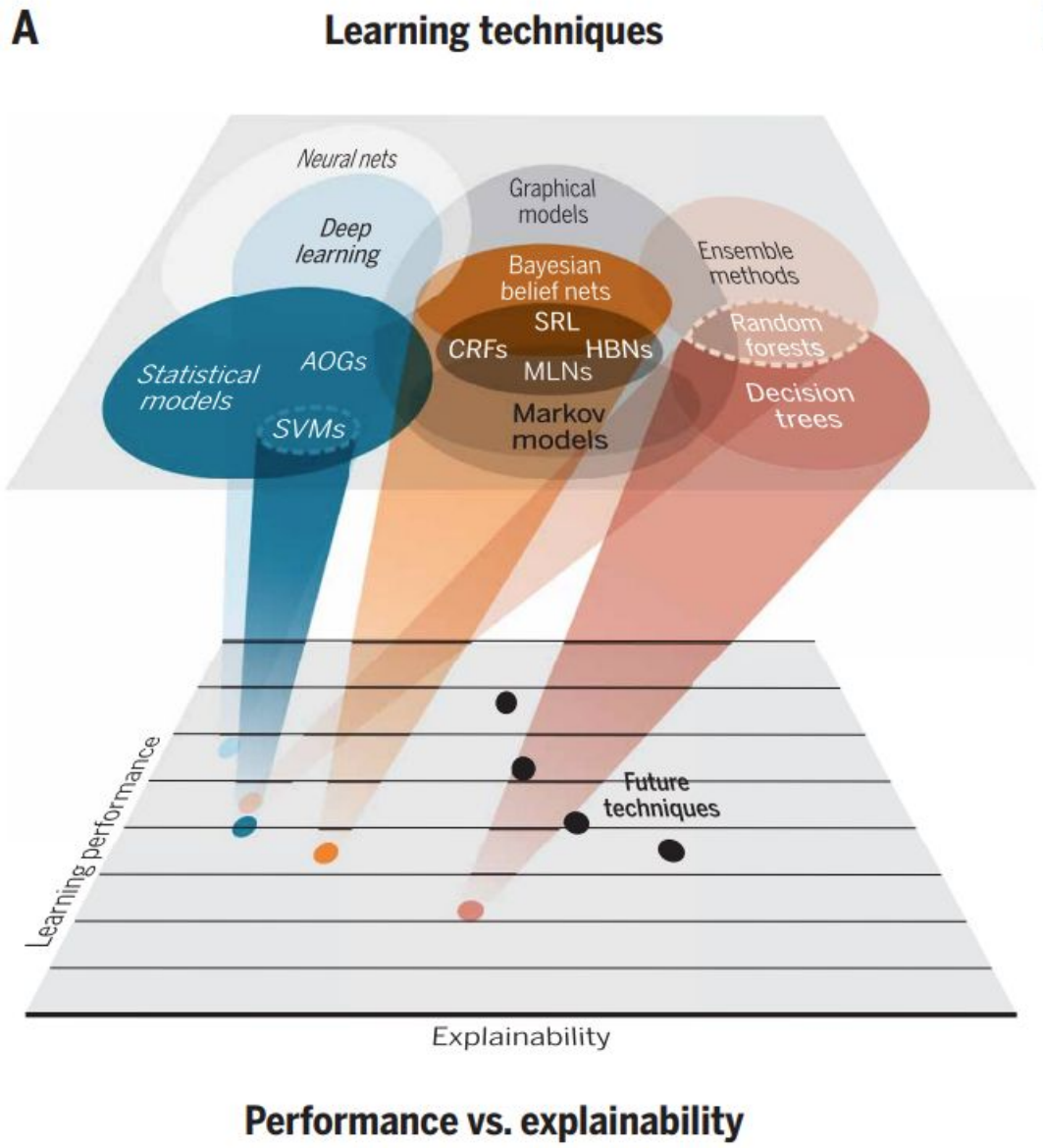


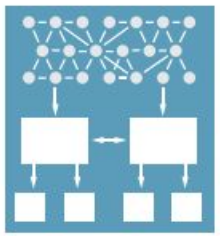
Traditional Machine Learning

Deep Learning

NLHPC
National Laboratory for High Performance Computing Chile

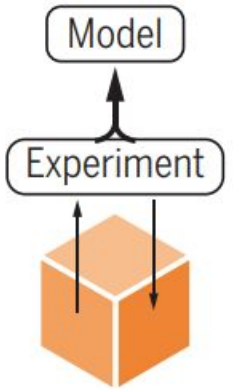
Interpretabilidad y precisión





Interpretable models
Techniques to learn more structured, interpretable, causal models

Deep learning
Improved deep learning techniques to learn explainable features

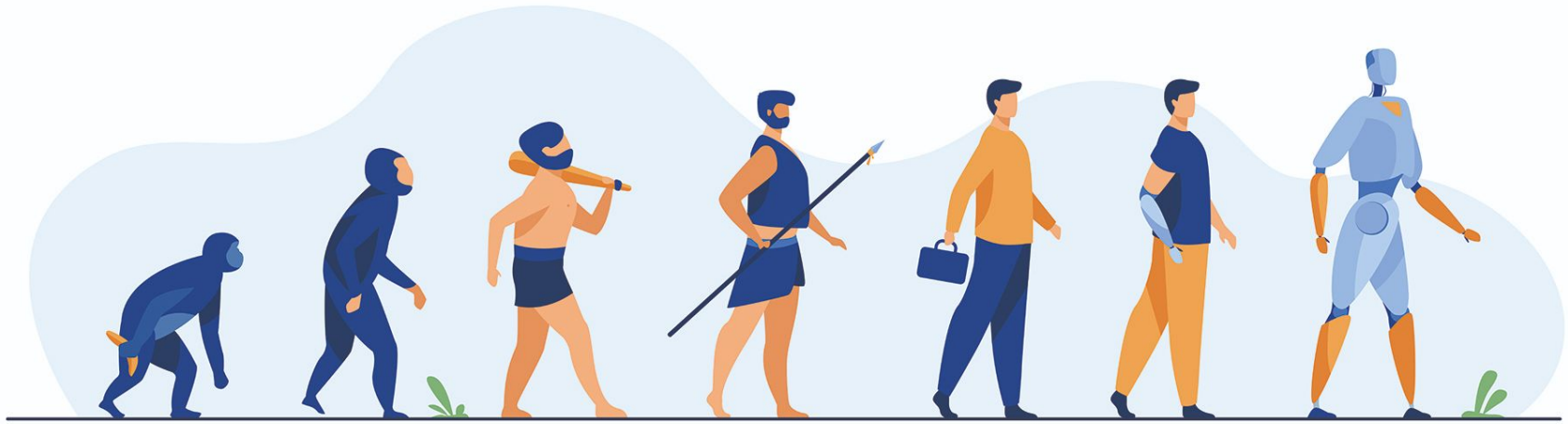


Model agnostic
Techniques to infer an explainable model from any model as a black box

Resumen primer módulo

Con la mano en la evolución de la capacidad de cómputo la Inteligencia artificial tiene una capacidad cada vez más potente.

Los modelos más complejos de IA “Deep Learning” aprenden de grandes cantidades de datos y generan estructuras complejas difíciles de interpretar.



FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

(CR)² Center for Climate and Resilience Research

Módulo 2

Arquitecturas de Redes Neuronales

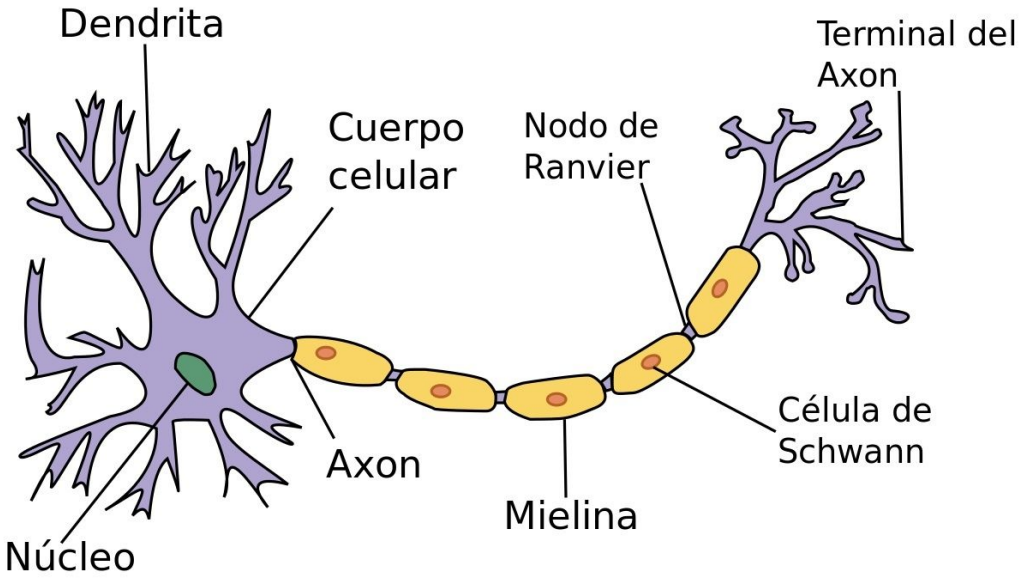
NLHPC
National Laboratory for High Performance Computing Chile

McCulloch & Pitts

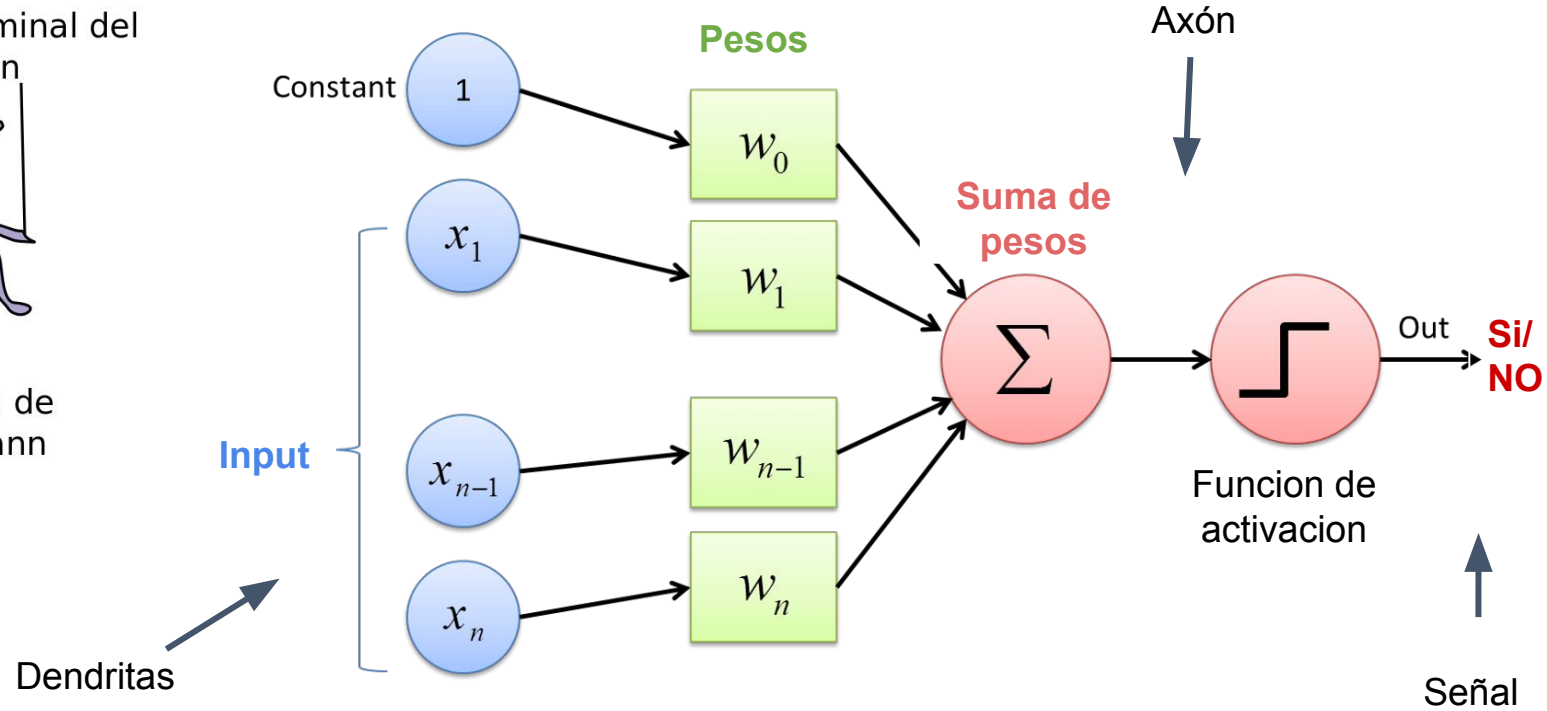
Machine Learning
Perceptrón

Center for Climate and Resilience Research

Def. Unidad básica para la transmisión de información, la célula computacional.
McCulloch & Pitts / Rosenblatt 1954



Desde : OpenStax College, Biología (CC BY 3.0)

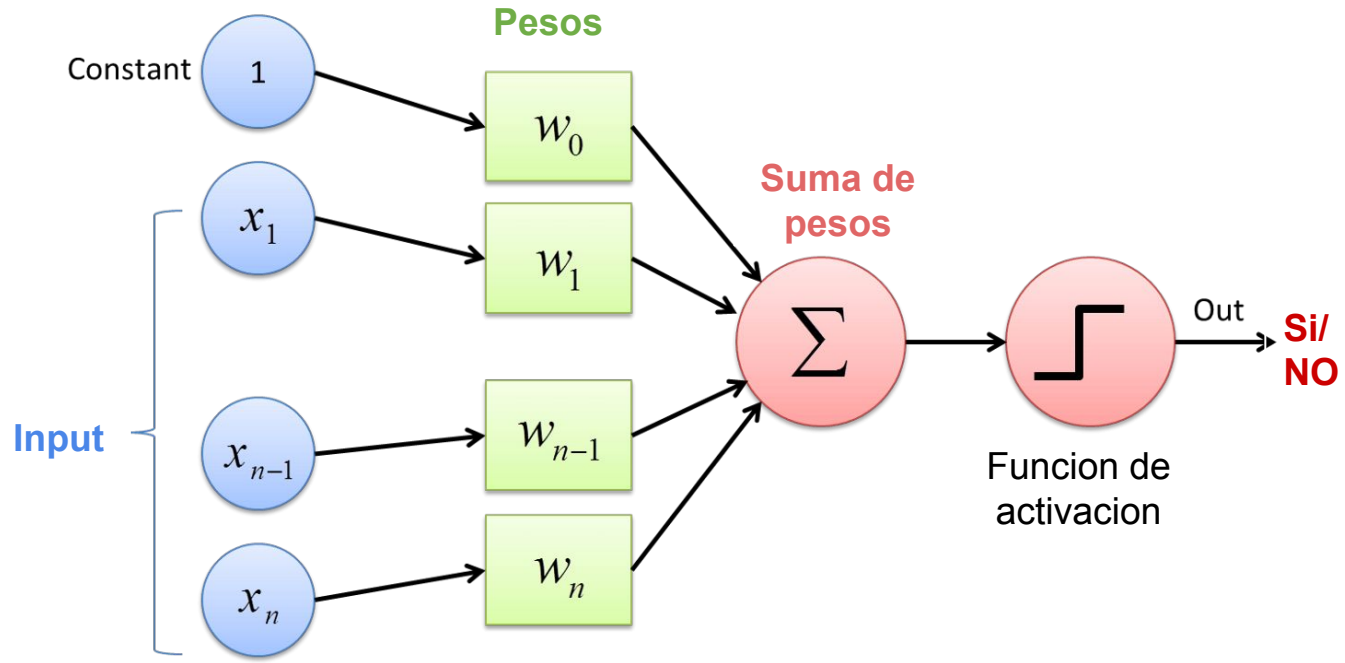


McCulloch & Pitts

Machine Learning
Perceptrón

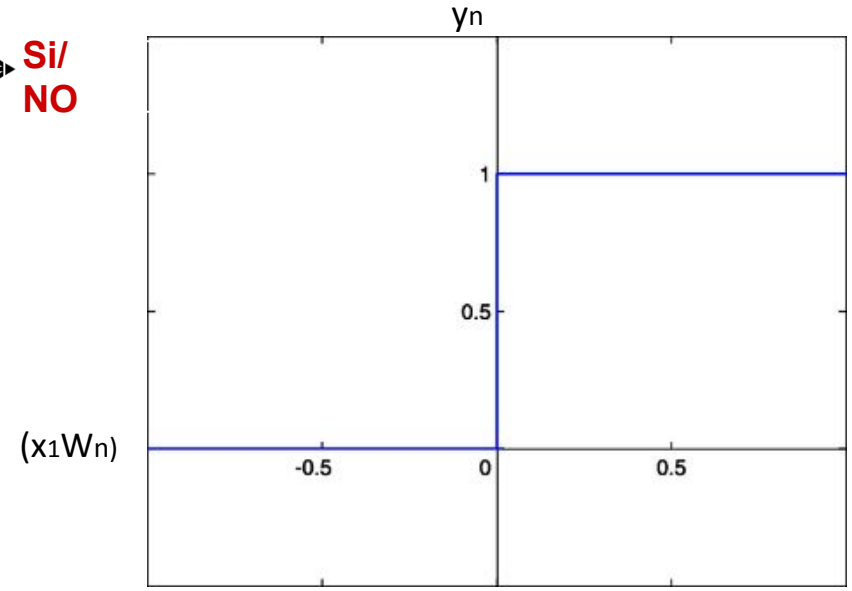
Center for Climate and Resilience Research

Pesos = Importancia información (coef numéricos)



Funcion de activacion= umbral de activación (función normalizadora)

Función umbral



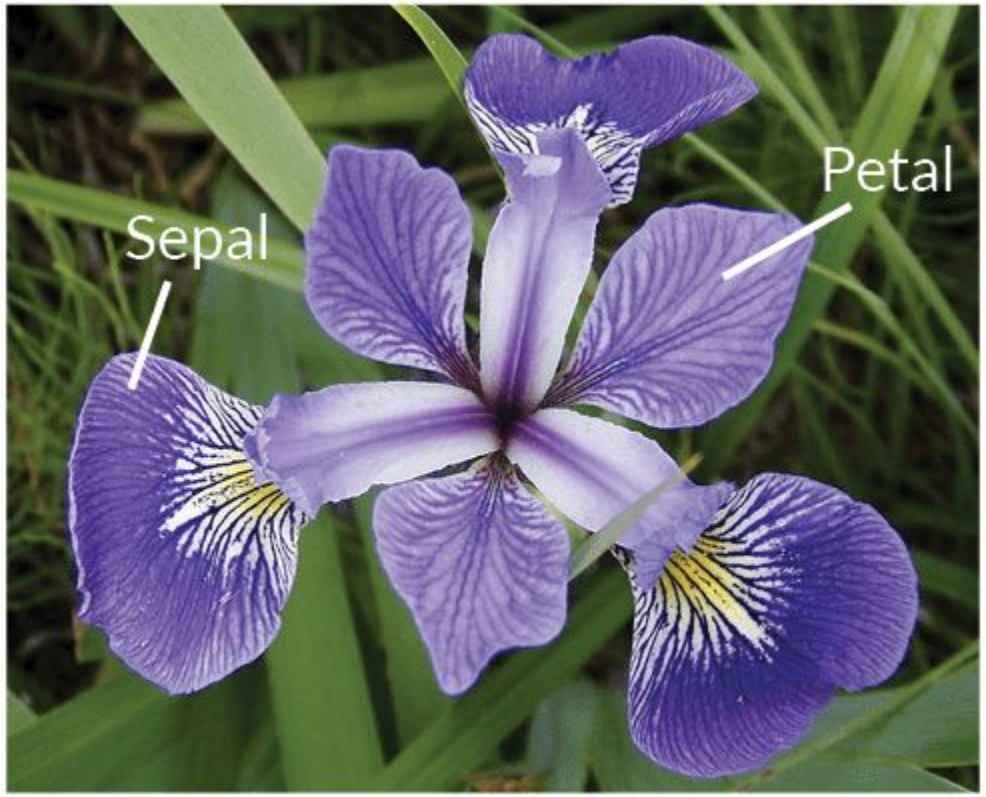
Input/Dendritas = Adquieren información de entrada, (set de datos o señales)

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Machine Learning

Clasificación de la flor iris

(CR)² Center for Climate and Resilience Research

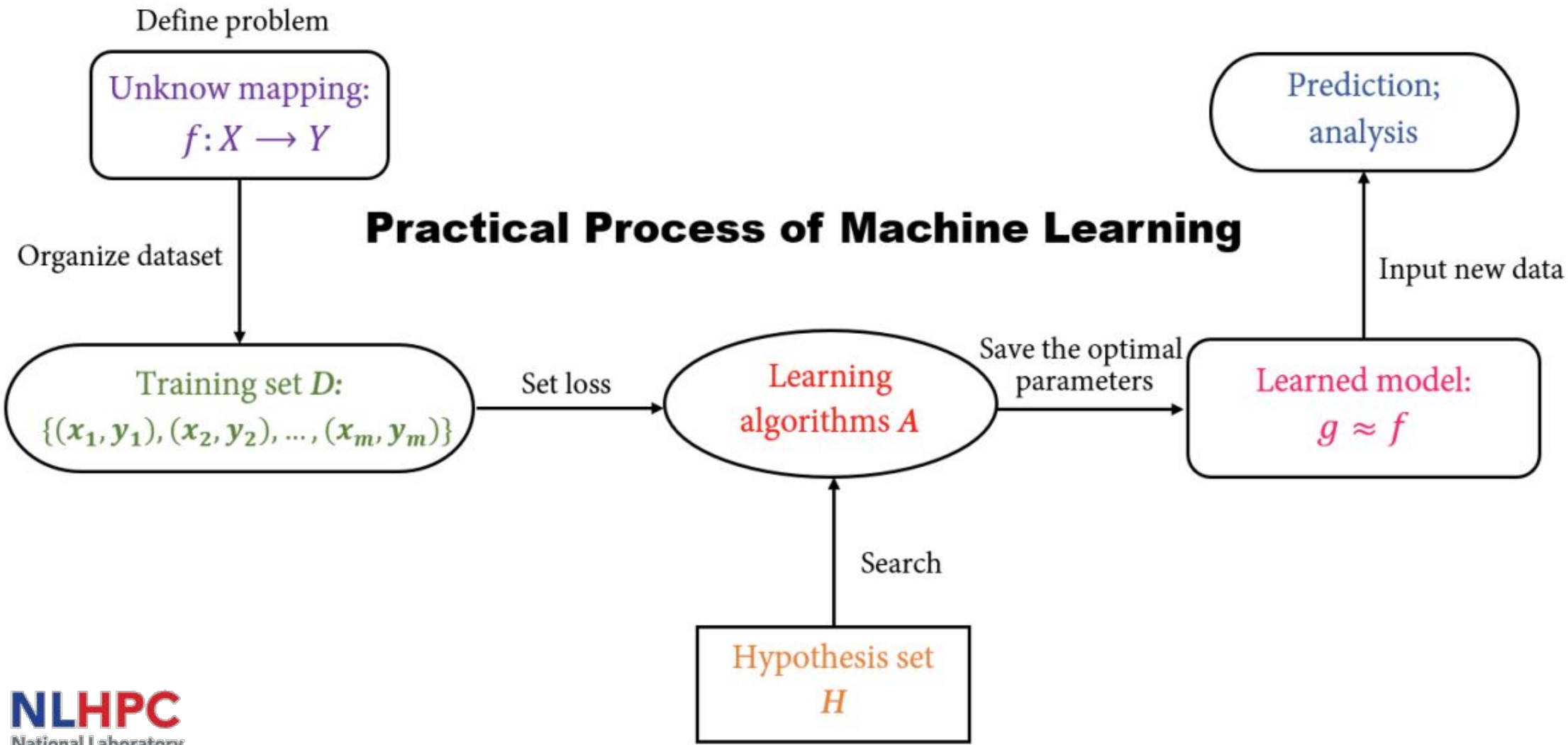


Iris Versicolor

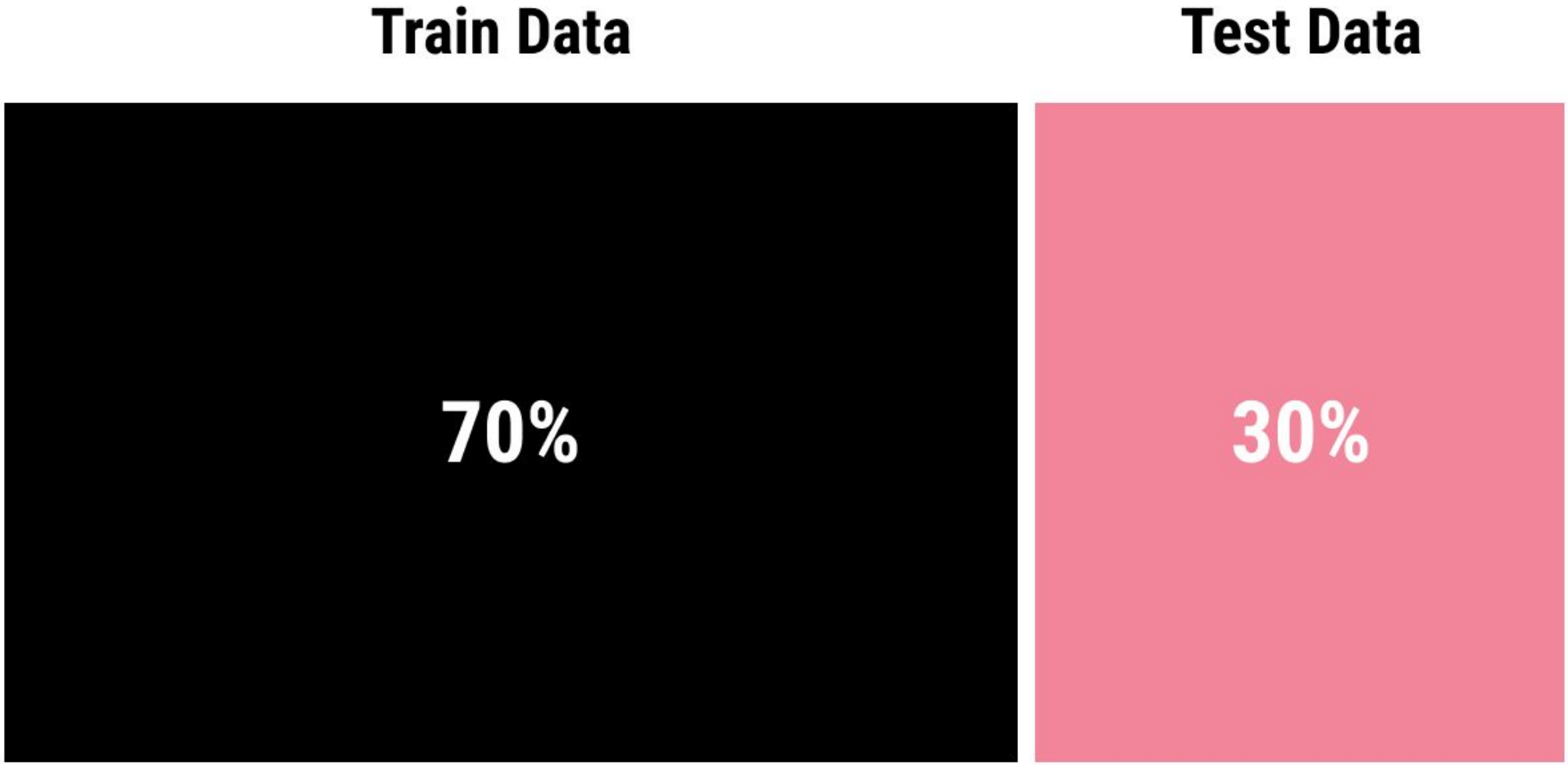
Iris Setosa

Iris Virginica

Proceso de modelos de Machine Learning

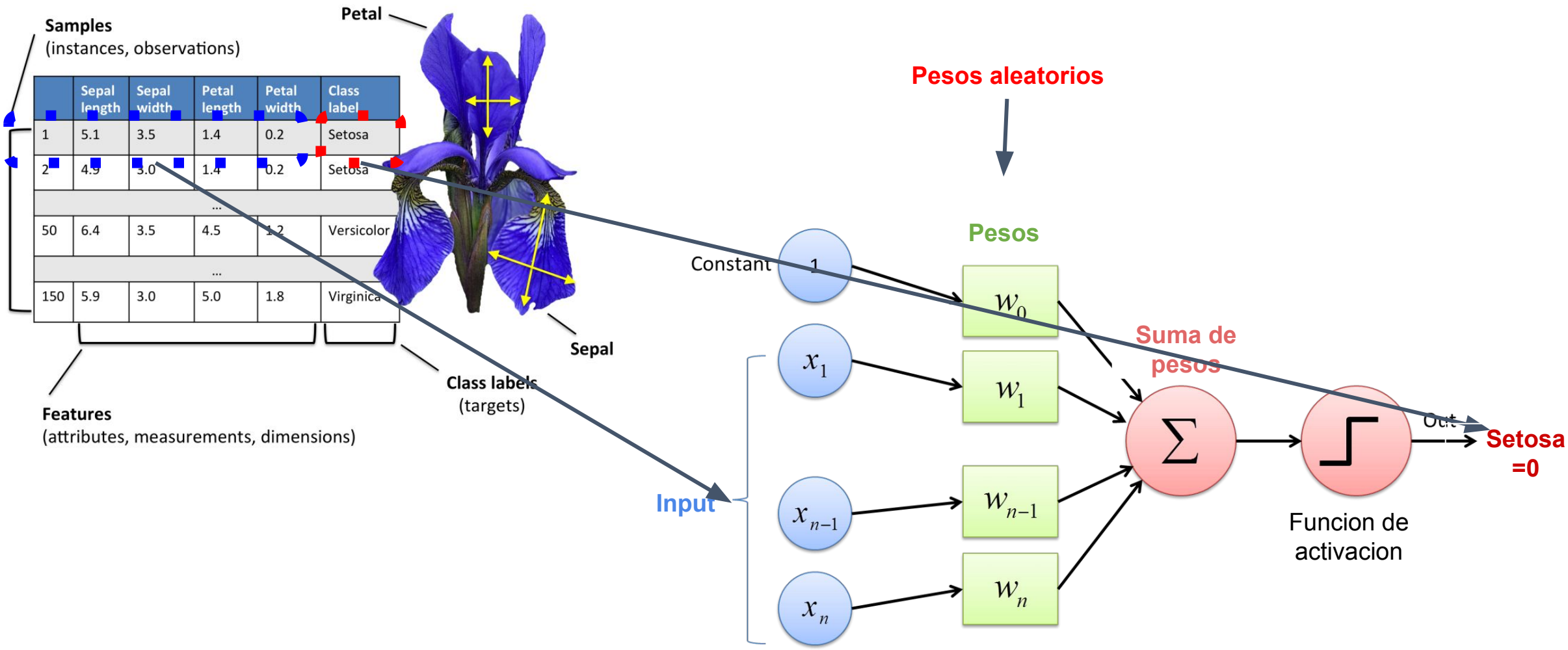


Rui Chen et al 2019


Train Data
Test Data
70%
30%

Machine Learning
Perceptrón

Paso 1

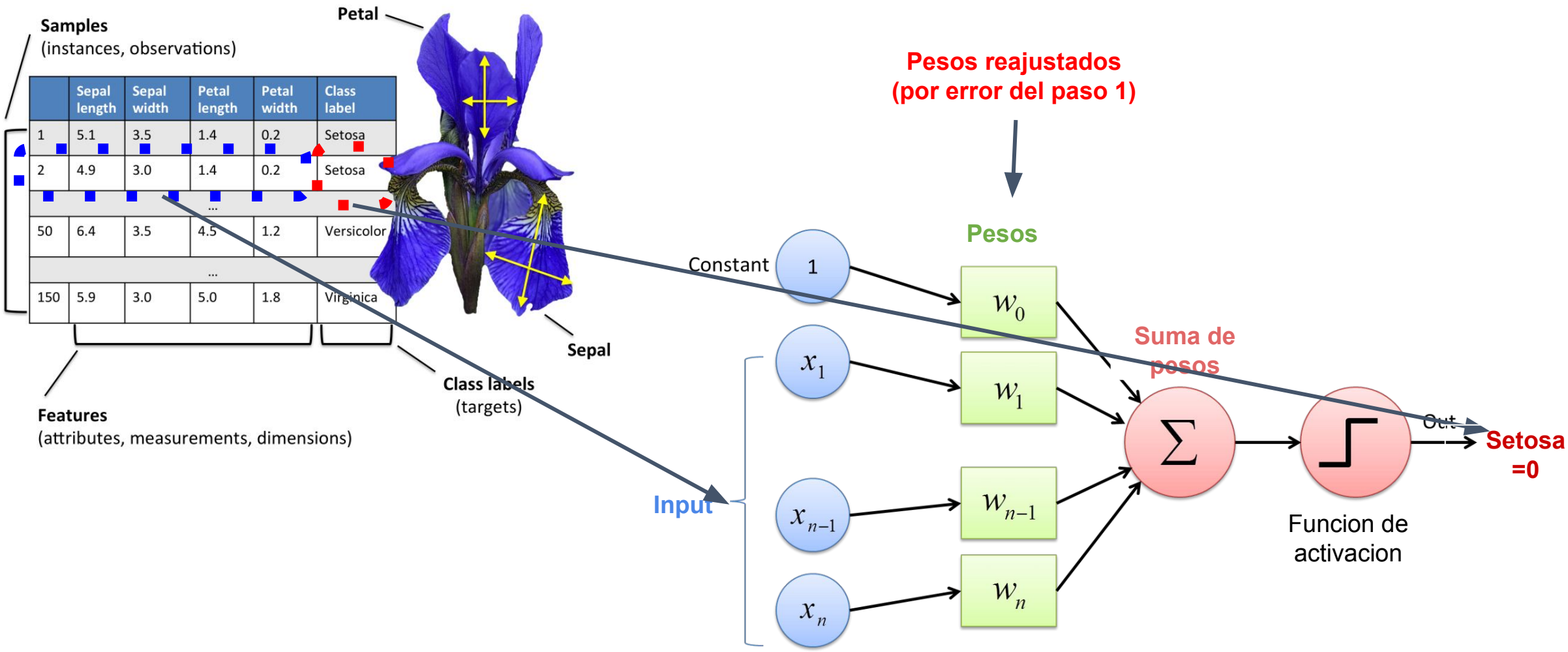


	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Machine Learning
Perceptrón

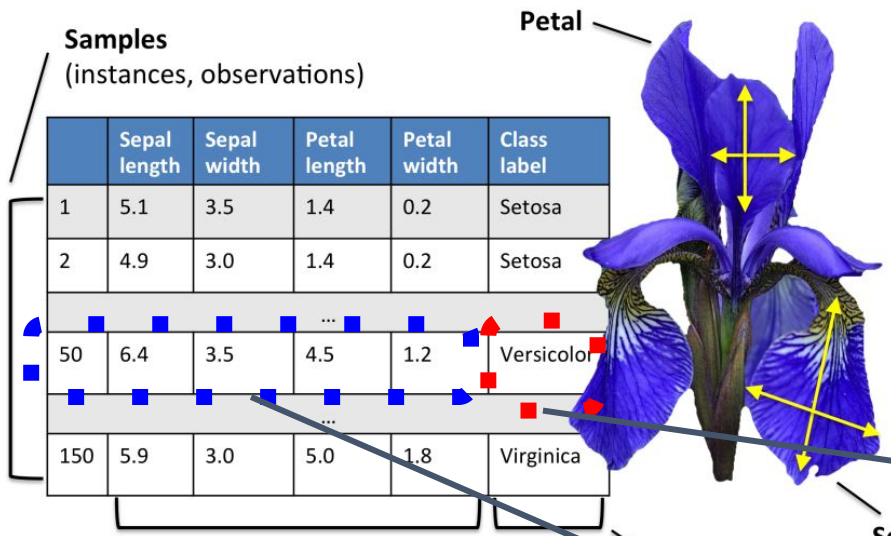
Center for Climate and Resilience Research

Paso 2



Machine Learning
Perceptrón

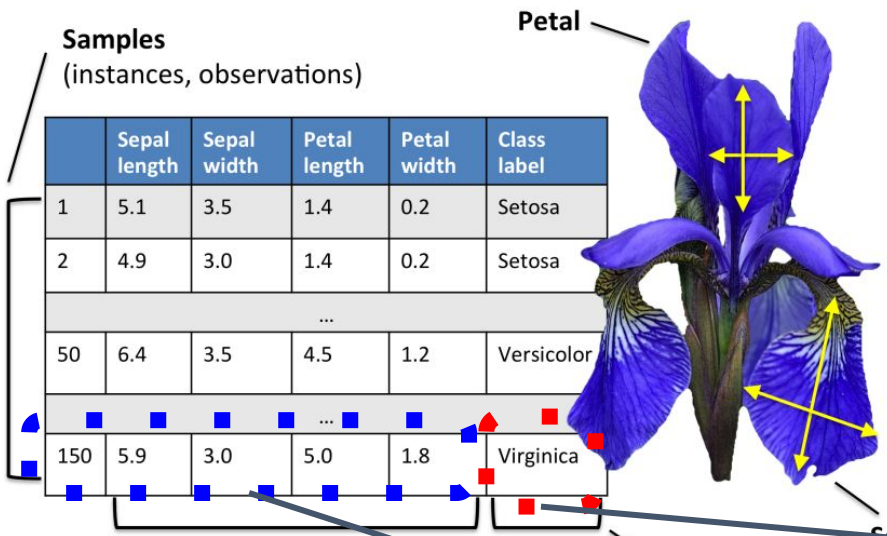
Paso 3



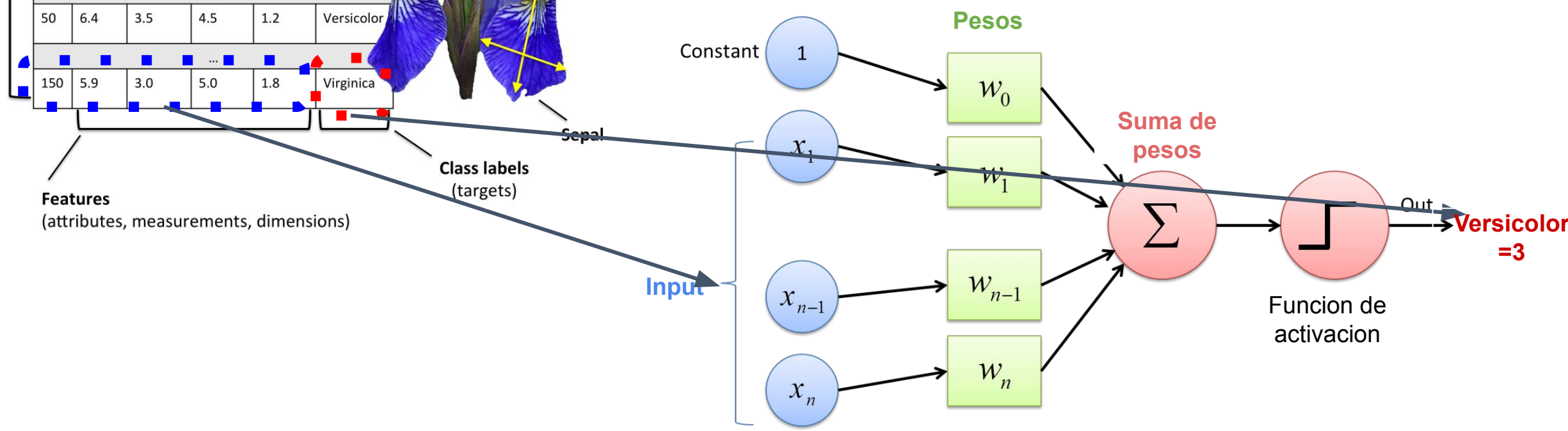
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Machine Learning
Perceptrón

Paso n



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

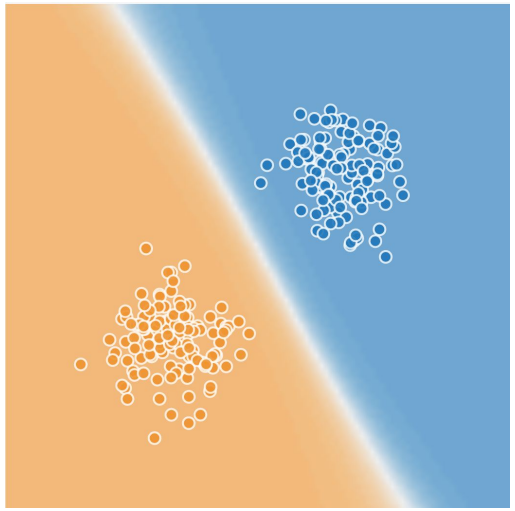


¿Límites del perceptrón?

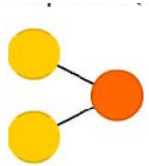
Machine Learning

(CR)² Center for Climate and Resilience Research

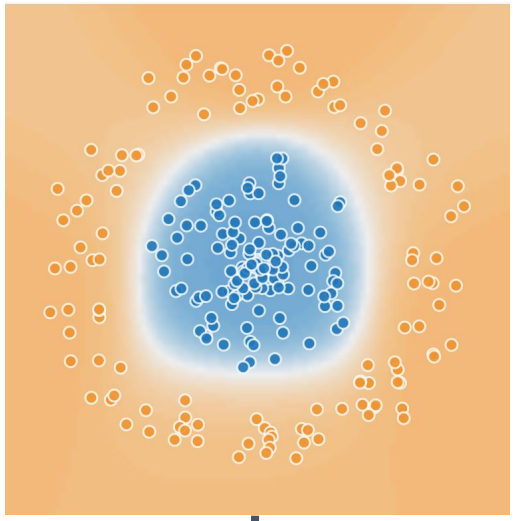
Problema lineal



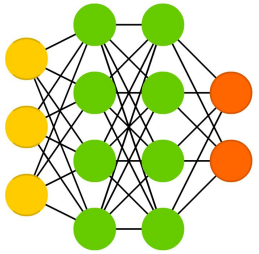
Perceptron



Problema no-lineal



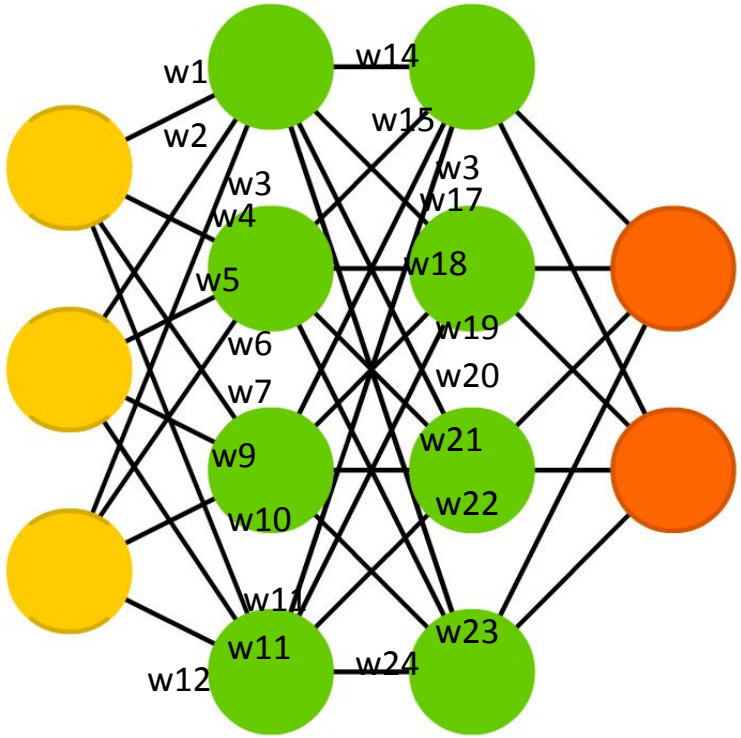
Red Multicapa



https://playground.tensorflow.org/

fuente: Tensorflow.org

¿Cómo determinamos tantos pesos?



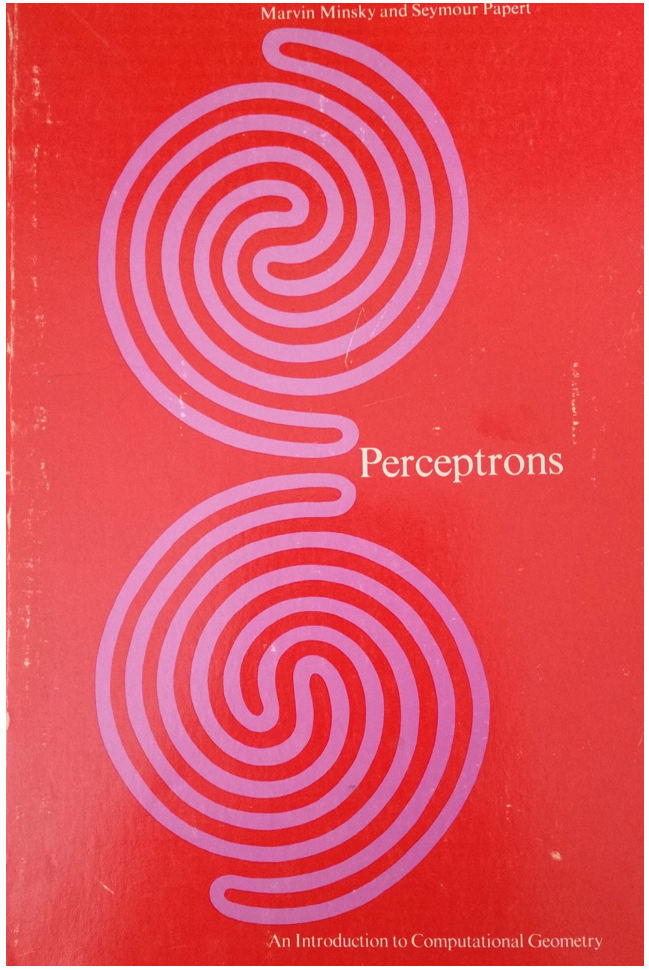
Fuente: The Asimov Institute

https://www.nature.com/articles/323533a0

El invierno IA

Center for Climate and Resilience Research

1969



14 años

1986

Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure[1].

There have been many attempts to design self-organizing neural networks. The aim is to find a powerful synaptic modification rule that will allow an arbitrarily connected neural network to develop an internal structure that is appropriate for a particular task domain. The task is specified by giving the more difficult when we introduce hidden units whose actual or desired states are not specified by the task. (In perceptrons, there are 'feature analysers' between the input and output that are not true hidden units because their input connections are fixed by hand, so their states are completely determined by the input vector: they do not learn representations.) The learning procedure must decide under what circumstances the hidden units should be active in order to help achieve the desired input–output behaviour. This amounts to deciding what these units should represent. We demonstrate that a general purpose and relatively simple procedure is powerful enough to construct appropriate internal representations.

The simplest form of the learning procedure is for layered networks which have a layer of input units at the bottom; any number of intermediate layers; and a layer of output units at the top. Connections within a layer or from higher to lower layers are forbidden, but connections can skip intermediate layers. An input vector is presented to the network by setting the states of the input units. Then the states of the units in each layer are determined by applying equations (1) and (2) to the connections coming from lower layers. All units within a layer have their states set in parallel, but different layers have their states set sequentially, starting at the bottom and working upwards until the states of the output units are determined.

The total input, x_j, to unit j is a linear function of the outputs, y_i, of the units that are connected to j and of the weights, w_{ji}, on these connections

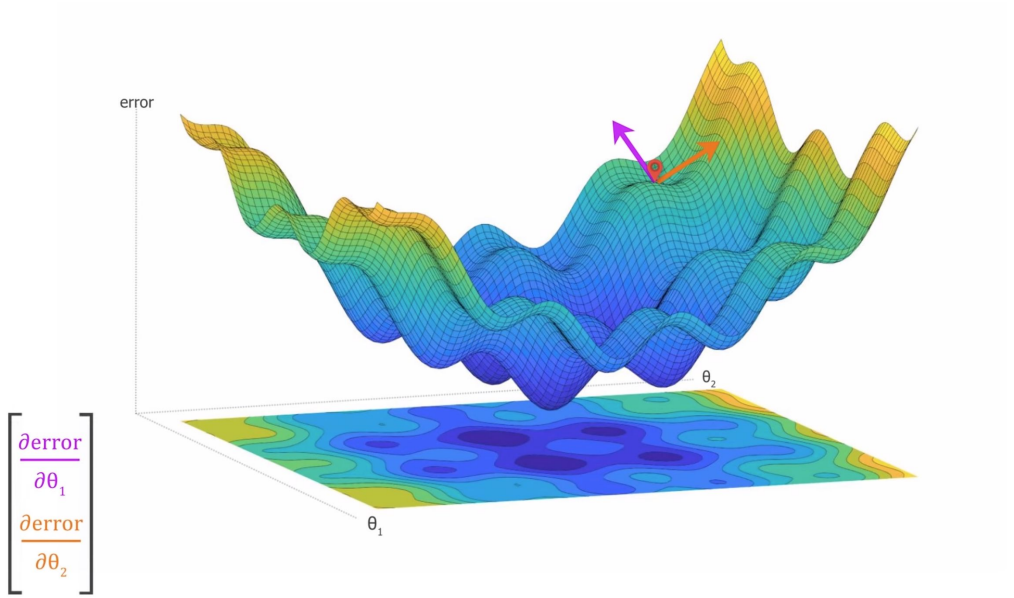
$$x_j = \sum_i y_i w_{ji} \qquad (1)$$

Problema severo para entrenar perceptrones con muchas conexiones

Back-propagation, Solución!

¿Entonces cómo entrenamos la red?

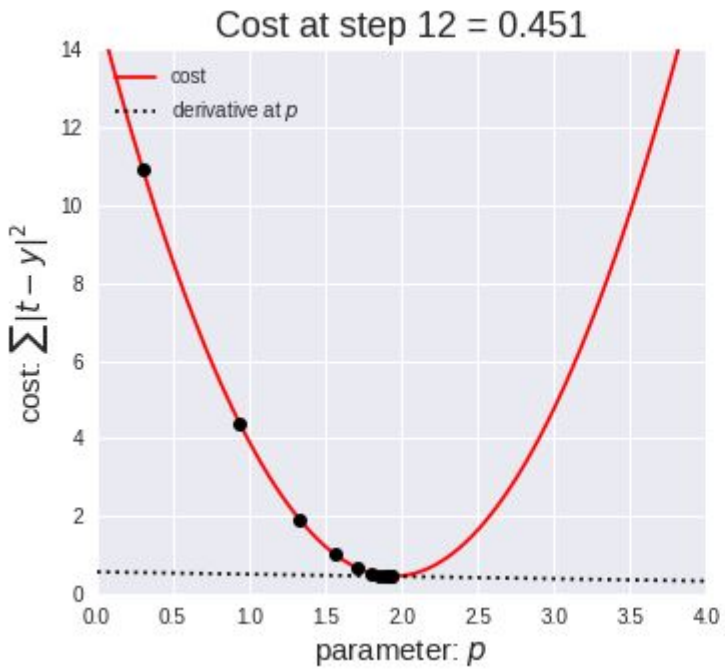
Computo del gradiente.

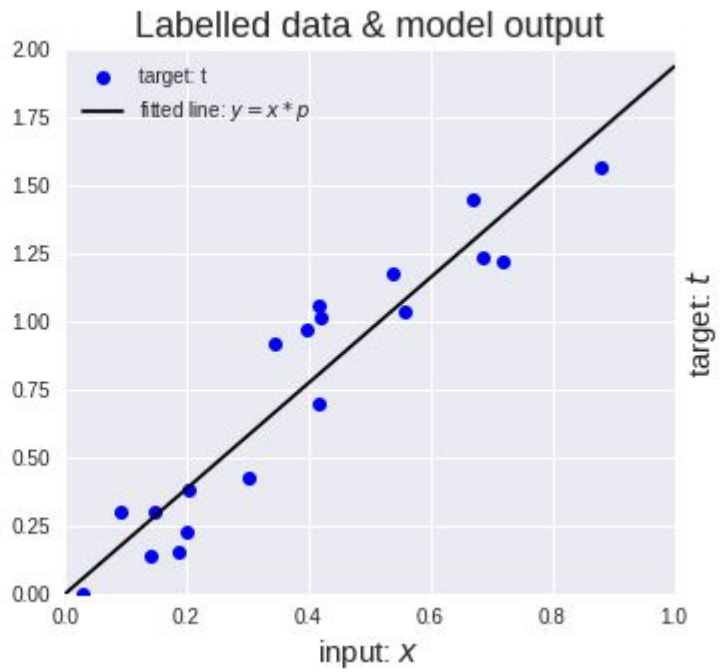


Descenso del Gradiente

(CR)² Center for Climate and Resilience Research

Caso 2D en un problema lineal

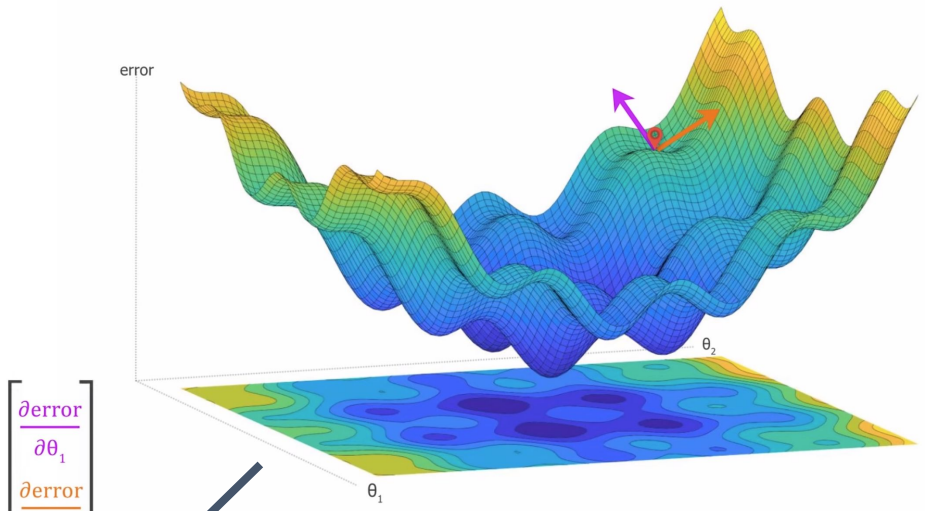




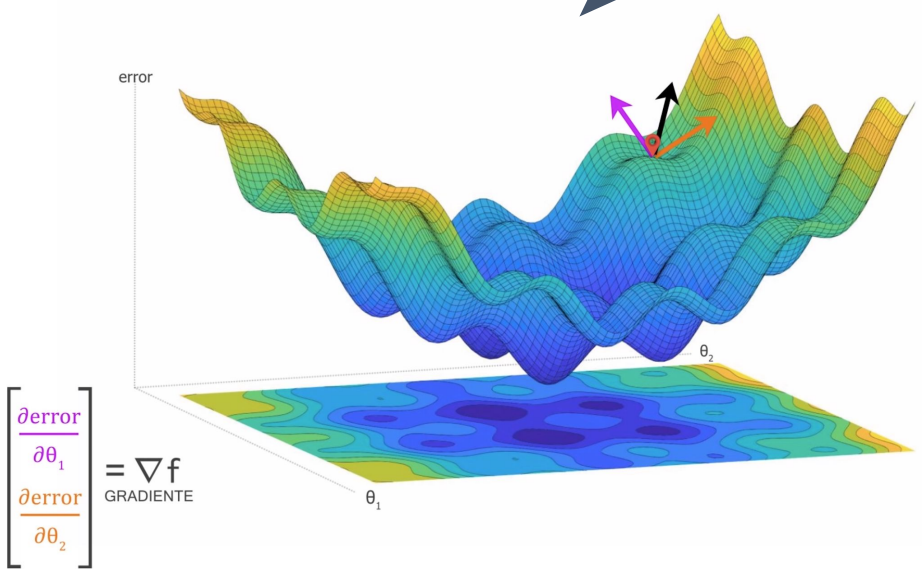
Fuente: [medium]

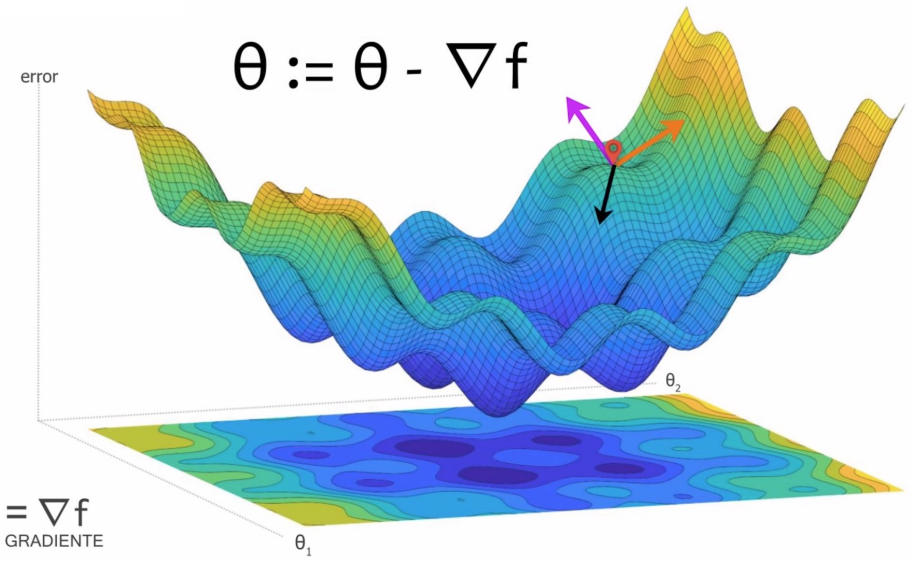
Descenso del Gradiente

Computo del gradiente.



Computo del gradiente.

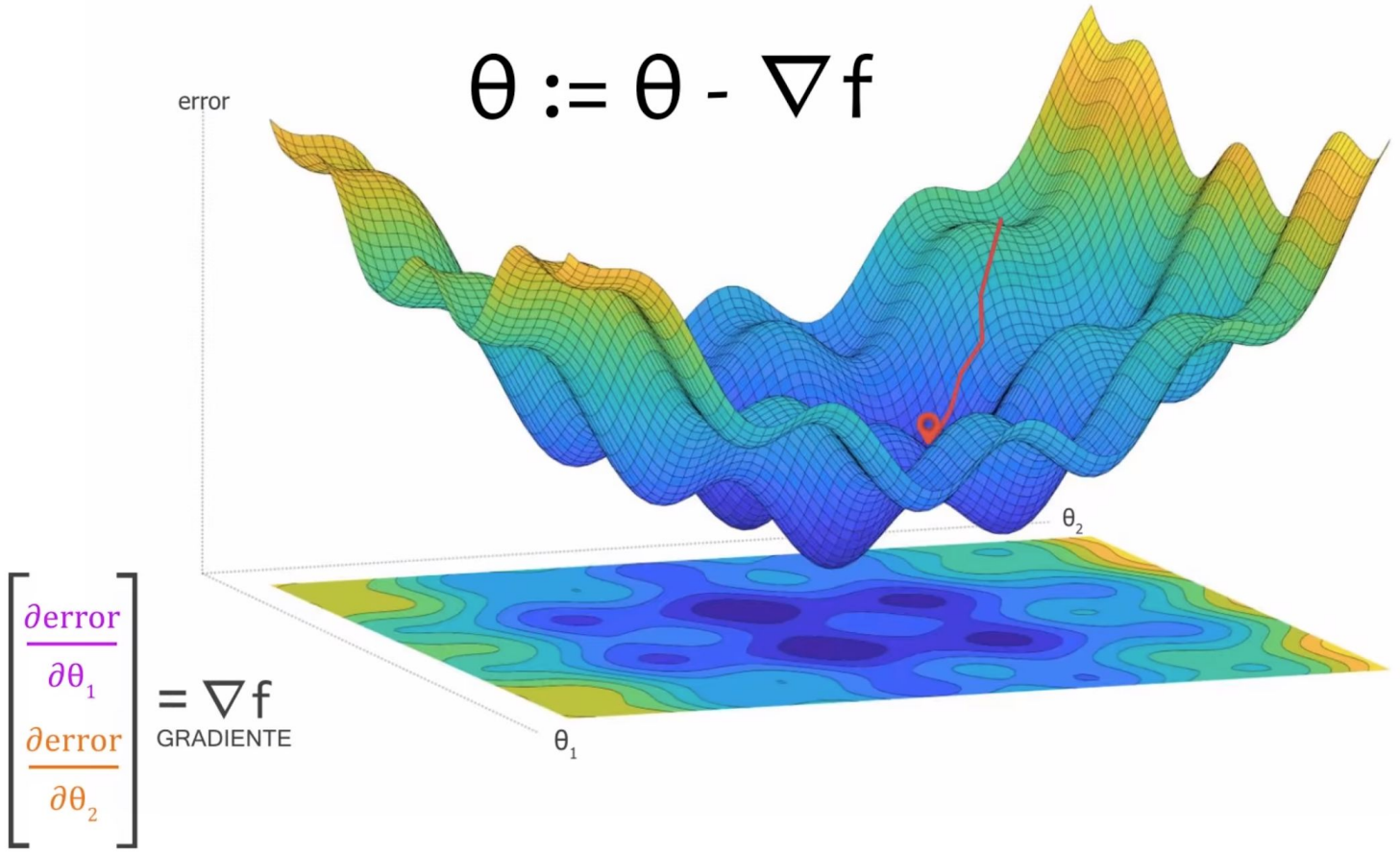




Fuente Dot CSV

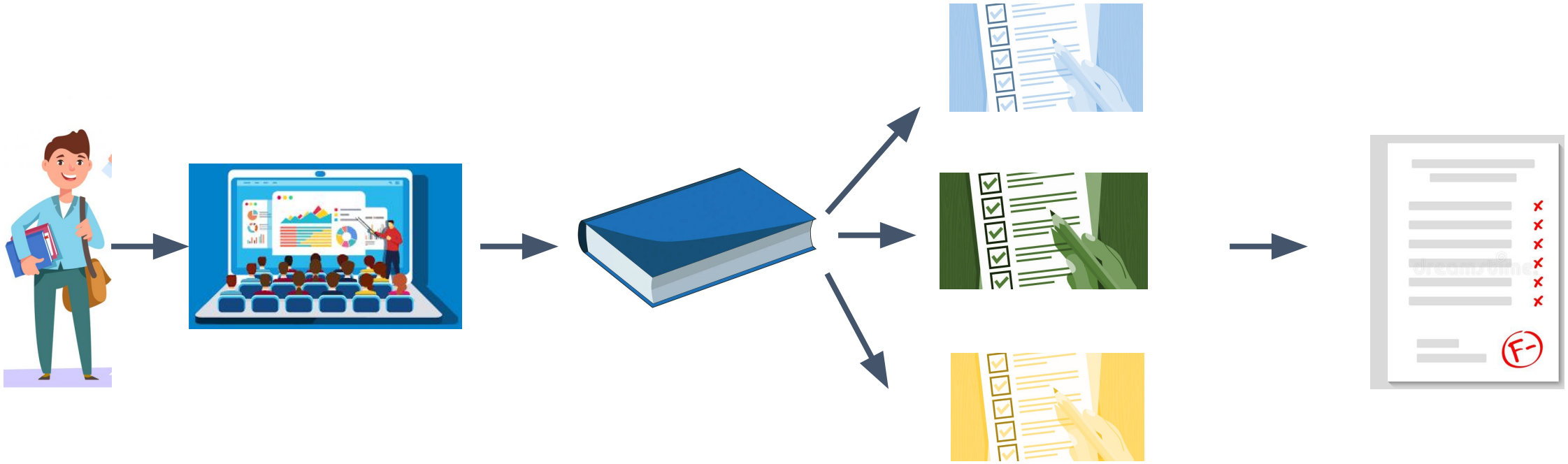
Descenso del Gradiente

Computo del gradiente.

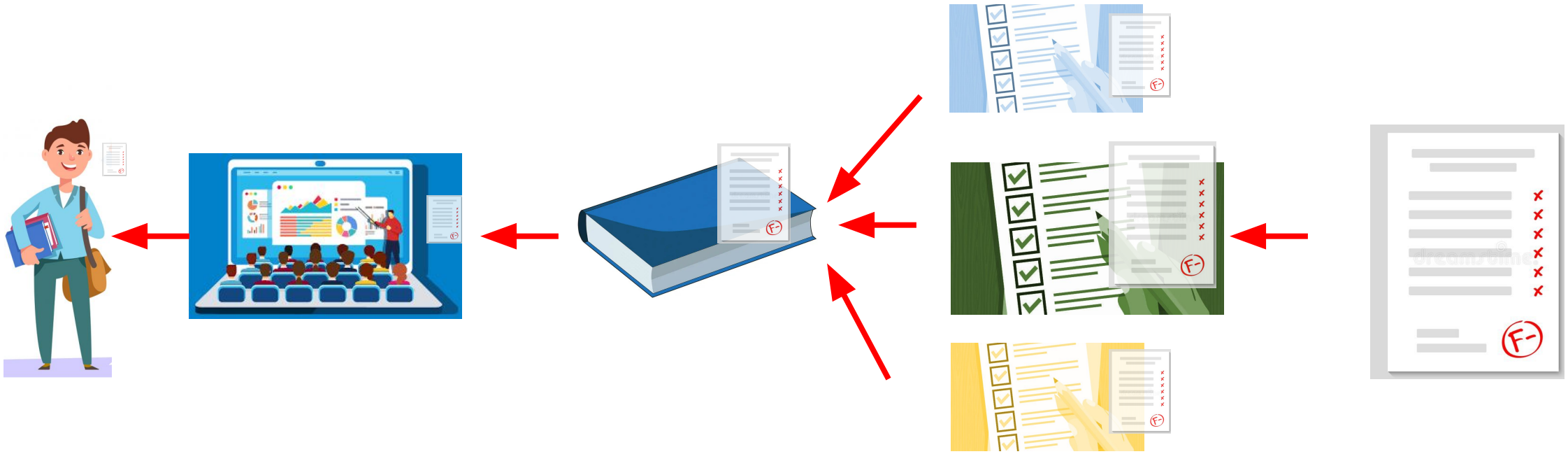


Fuente Dot CSV

Algoritmo: Backpropagation

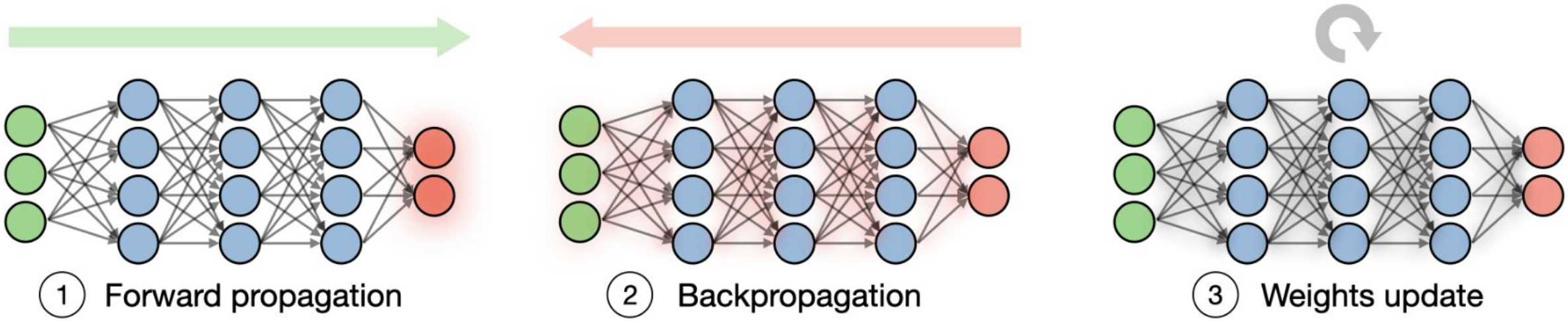


Algoritmo: Backpropagation



Algoritmo: Backpropagation

En resumen, propagamos los errores de las capas hacia atrás y luego actualizamos los pesos, para minimizar el error

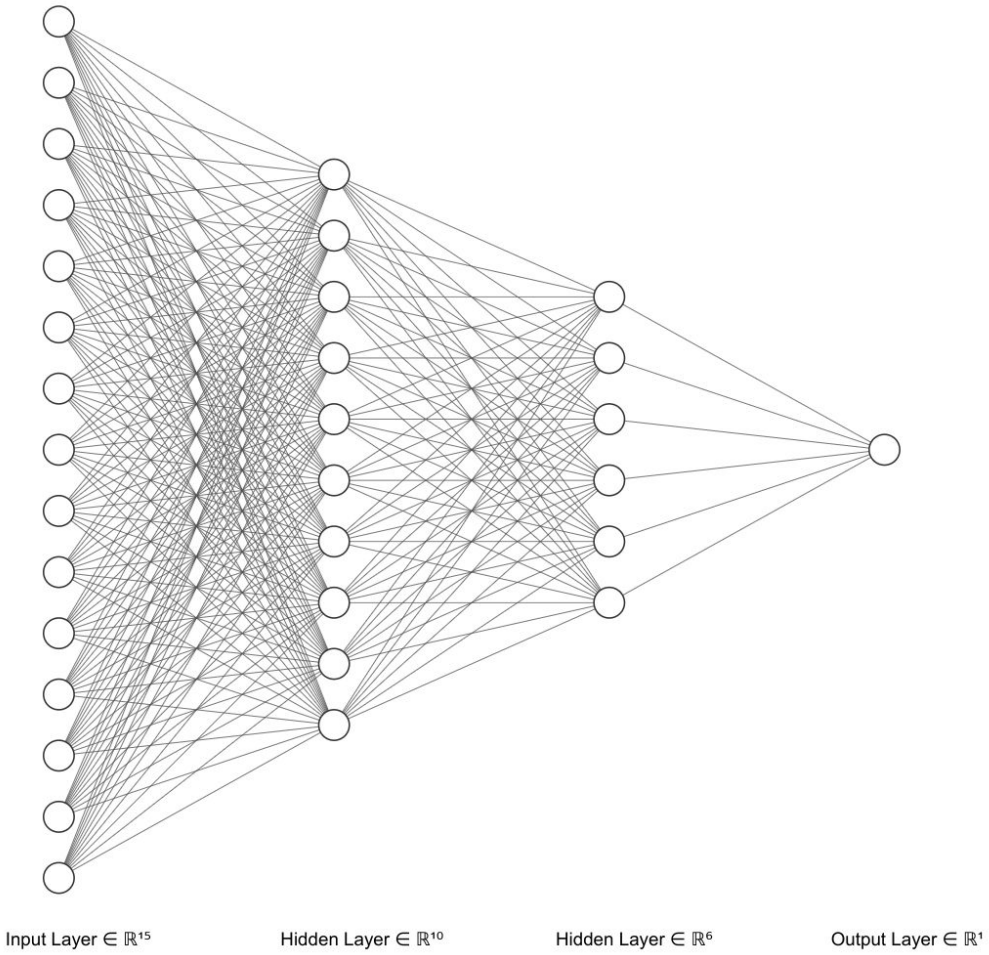


Fuente: stanford

Estructura e Hiperparametros

https://playground.tensorflow.org/

http://alexlenail.me/NN-SVG/index.html



Hiperparametros

Funciones de activación

Opciones

- Sigmoide
- ReLu
- Linear

Fuente: stanford

Hiperparametros

Funciones de pérdida

Opciones

- Erro cuadratico (MSE)
- Error medio (MAE)
- Cross Entropy

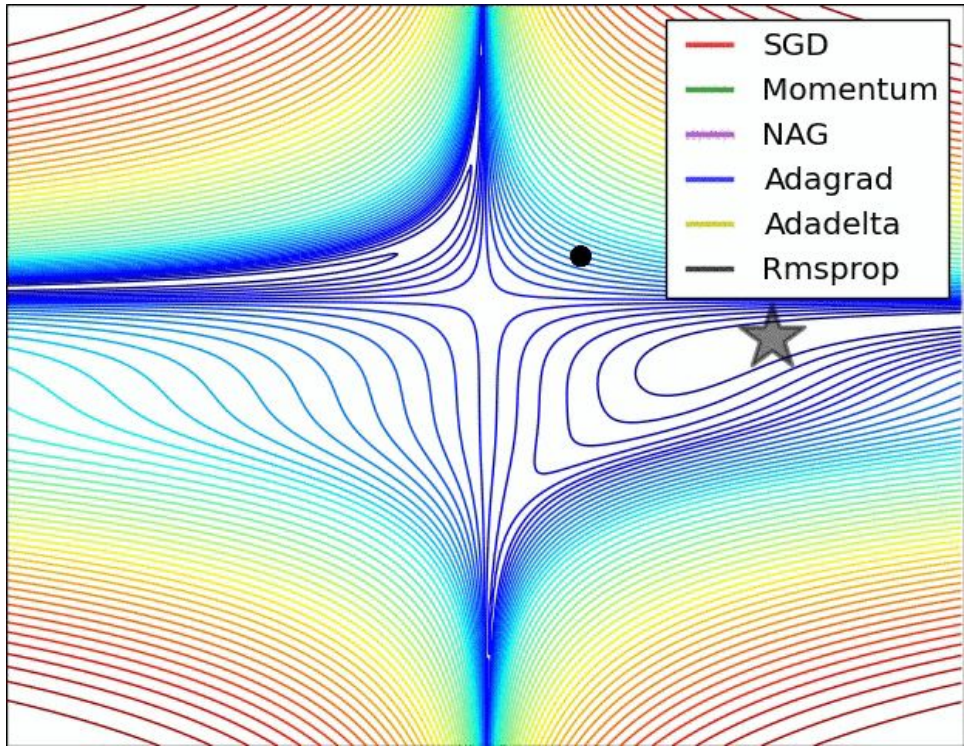
Fuente: stanford

Adam:
https://arxiv.org/pdf/1412.6980.pdf

(CR)² Center for Climate and Resilience Research

Hiperparametros

Gradiente descendente



Opciones

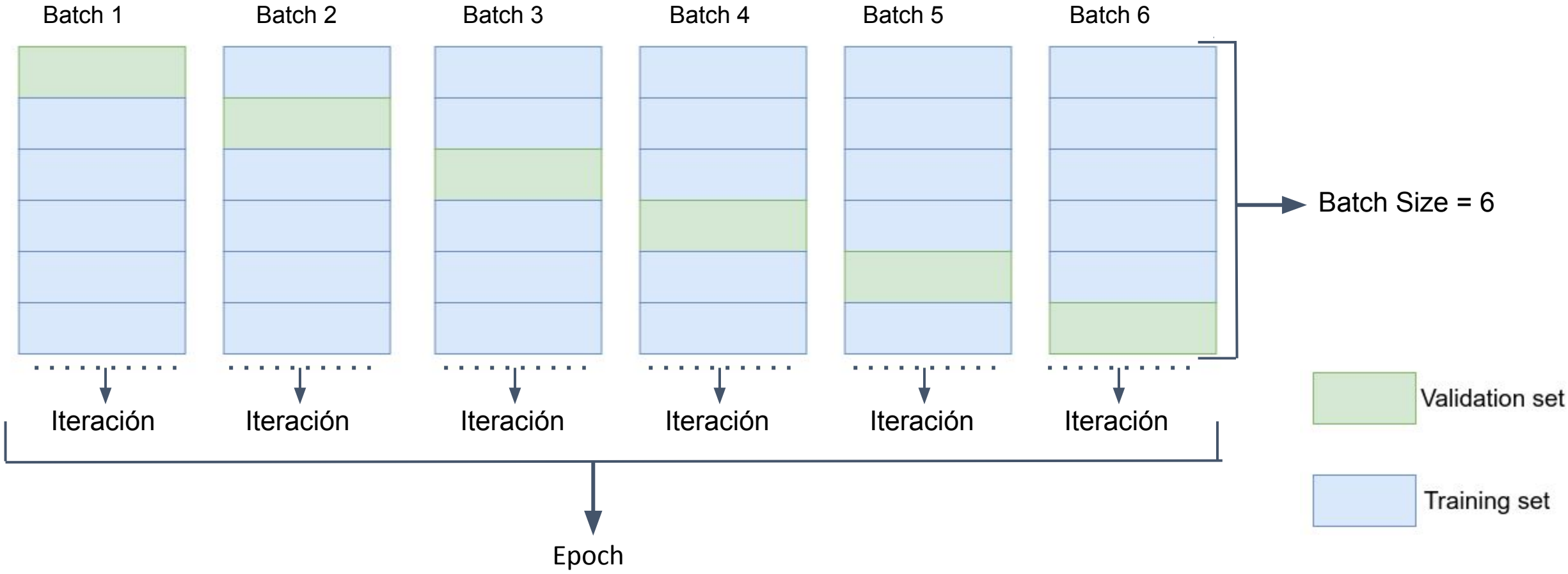
- Rmsprop
- Adagrad
- Adam

Method	Explanation	Update of w
Momentum	• Dampens oscillations • Improvement to SGD • 2 parameters to tune	$w - \alpha v_{dw}$ ← δ
RMSprop	• Root Mean Square propagation • Speeds up learning algorithm by controlling oscillations	$w - \alpha \frac{dw}{\sqrt{s_{dw}}}$ ← δ
Adam	• Adaptive Moment estimation • Most popular method • 4 parameters to tune	$w - \alpha \frac{v_{dw}}{\sqrt{s_{dw}} + \epsilon}$ ← δ

Fuente: stanford

Hiperparametros

Batch , Batch size, Epoch, iteración



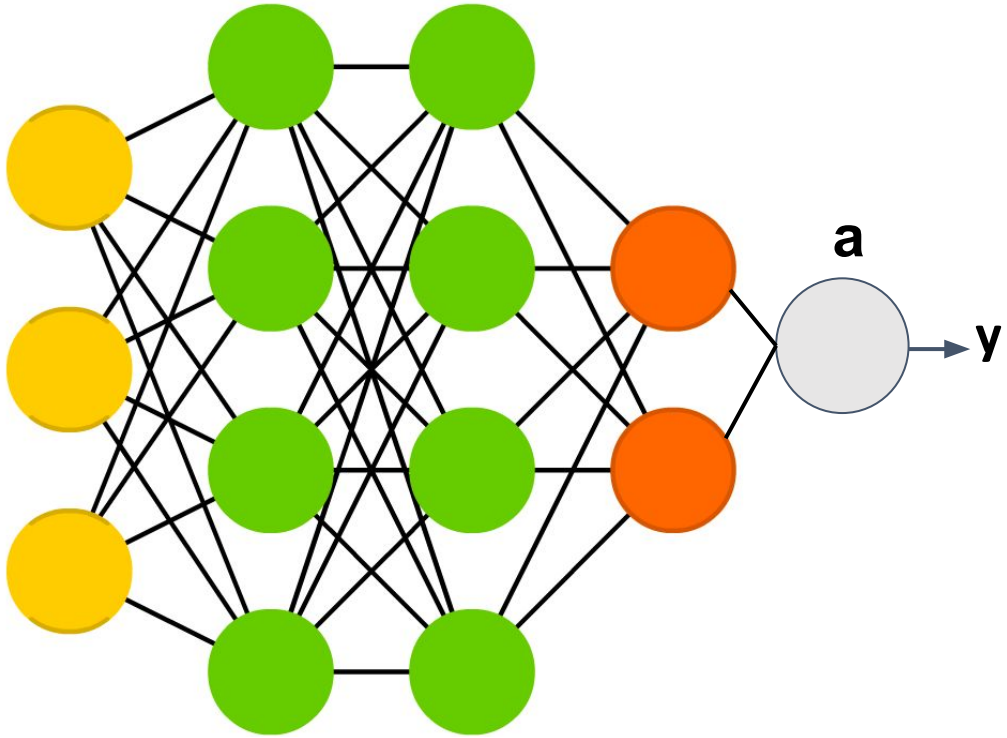
Resumen 2 Módulo; Hiperparametros

Funciones de activación

Funcion de perdida

Metodo de optimizacion

Batch size y Epoch



Pesos

Neuronas

Capas Ocultas

Input (datos)

Datos de entrenamientos (70%)

Datos de testeo (30%)

Resumen segundo módulo

Las redes neuronales profundas (Deep Learning) son estructuras complejas, las cuales tienen varios parámetros para ajustar.

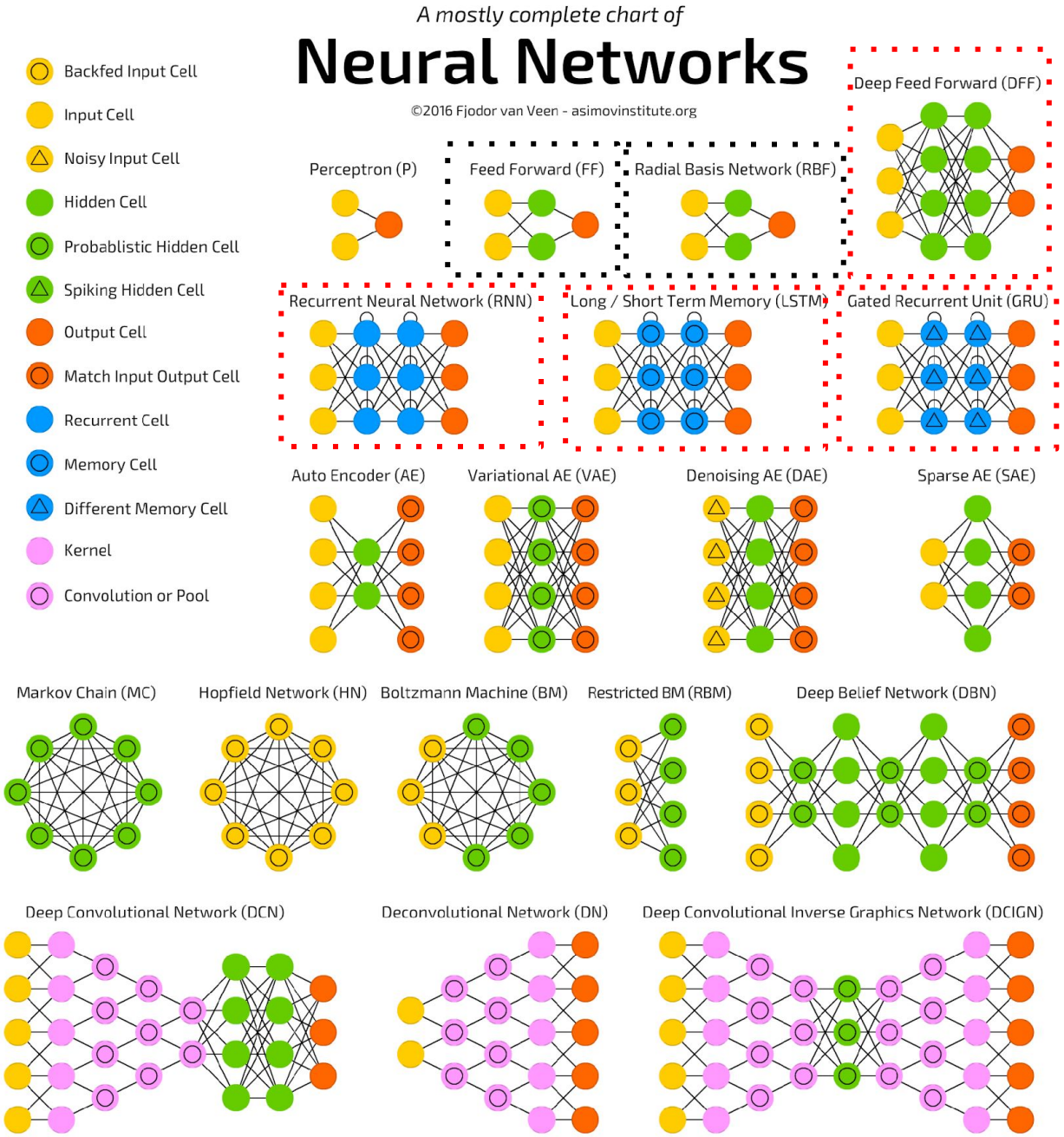
Para hacer uso de los modelos de Deep Learning necesitamos grandes cantidades de datos, los cuales debemos separarlos en el entrenamiento y el testing.

Cada iteración o epoch de la red, tiene un costo computacional que puede ser elevado, para entrenar las redes usamos HPC.

Módulo 3

Transfer Learning

Arquitecturas de Deep Learning



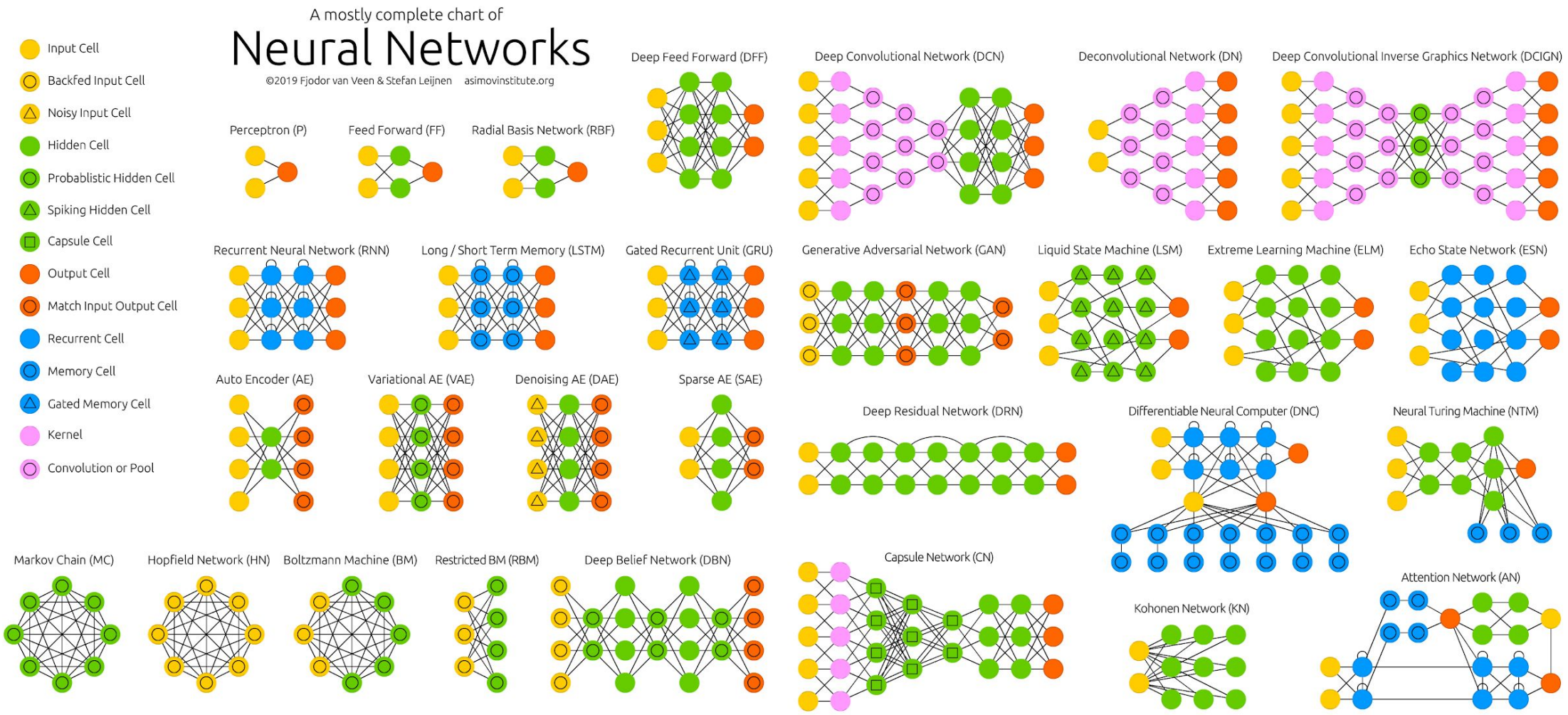
Traditional Machine Learning

Deep Learning

(CR)² Center for Climate and Resilience Research

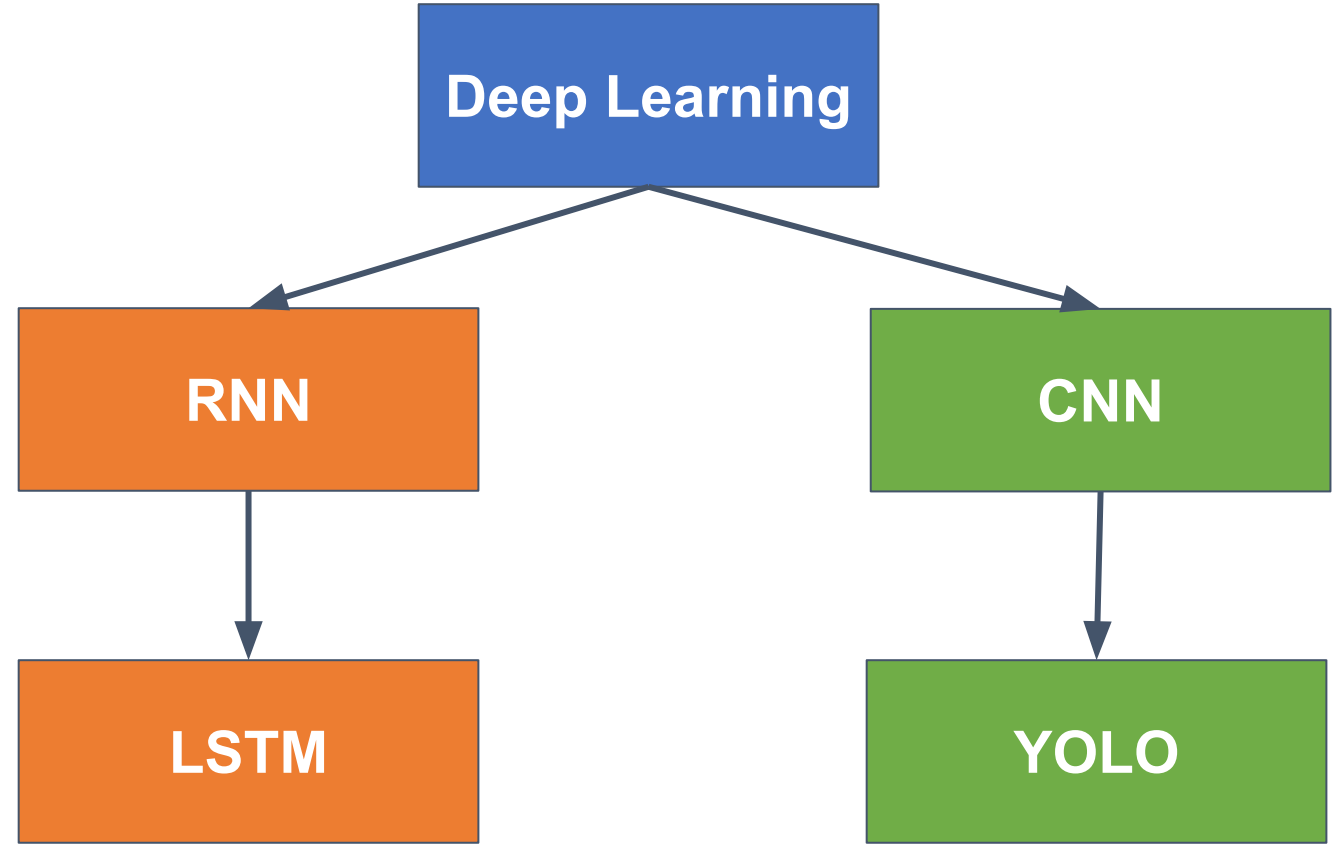
NLHPC National Laboratory for High Performance Computing Chile

Arquitecturas de Deep Learning



Fuente: [The Asimov Institute]

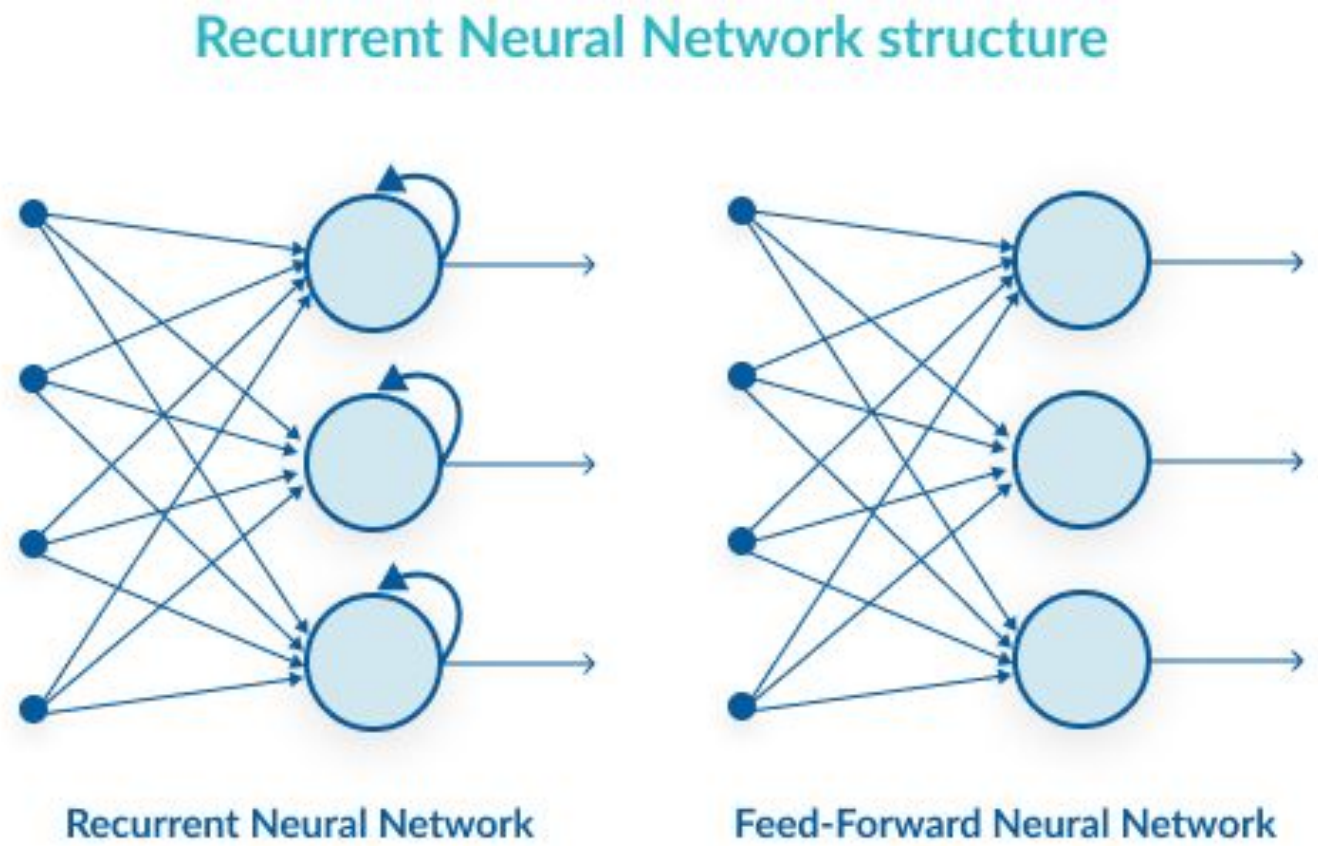
Arquitecturas de Deep Learning



Arquitectura	RNN	CNN
Tipo	LSTM	YOLO
Aplicación	Aplicadas en time series y NLP	Aplicadas en Visual Computing

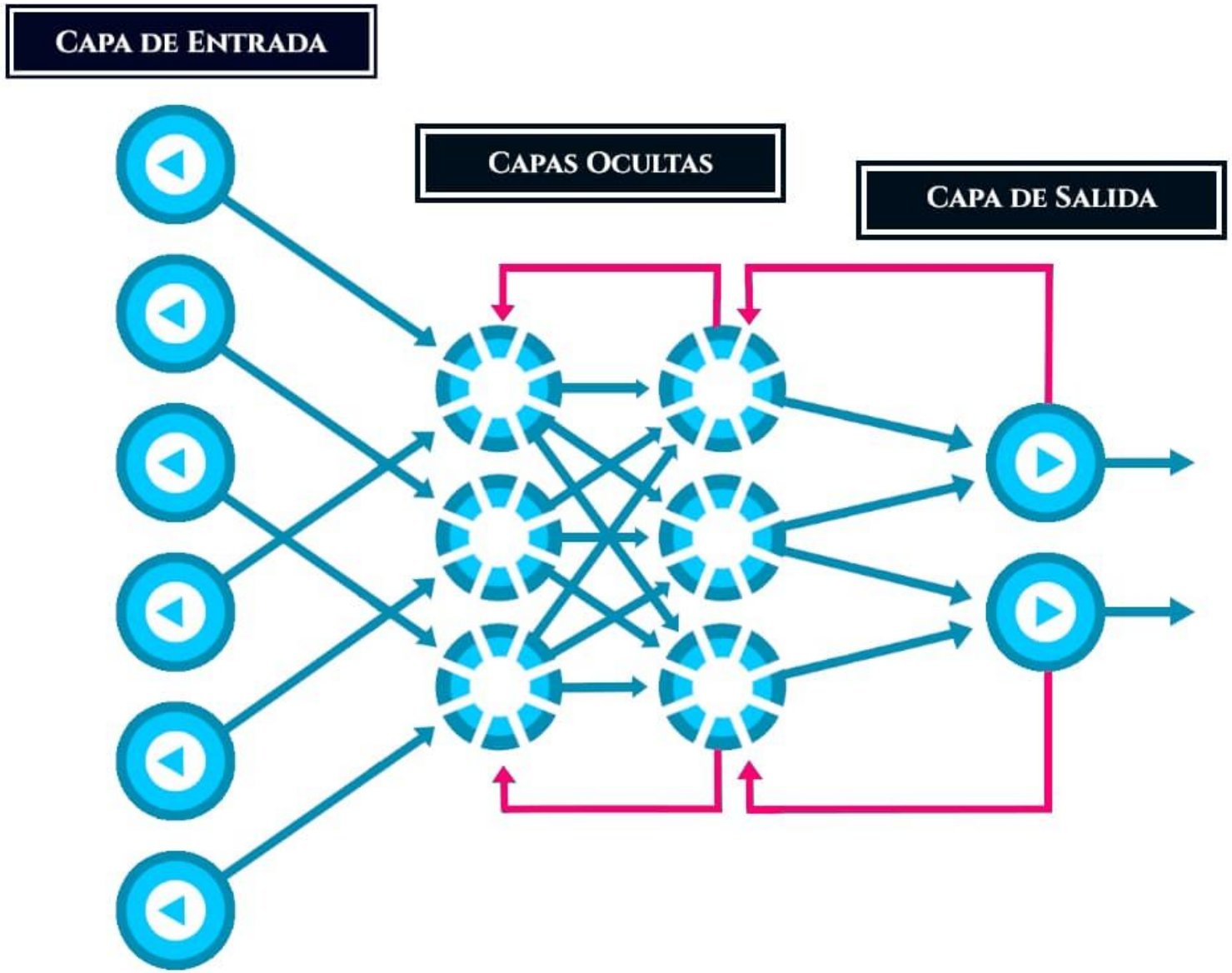
RNN

Redes Neuronales Recurrentes (RNN)



RNN

Redes Neuronales Recurrentes (RNN)

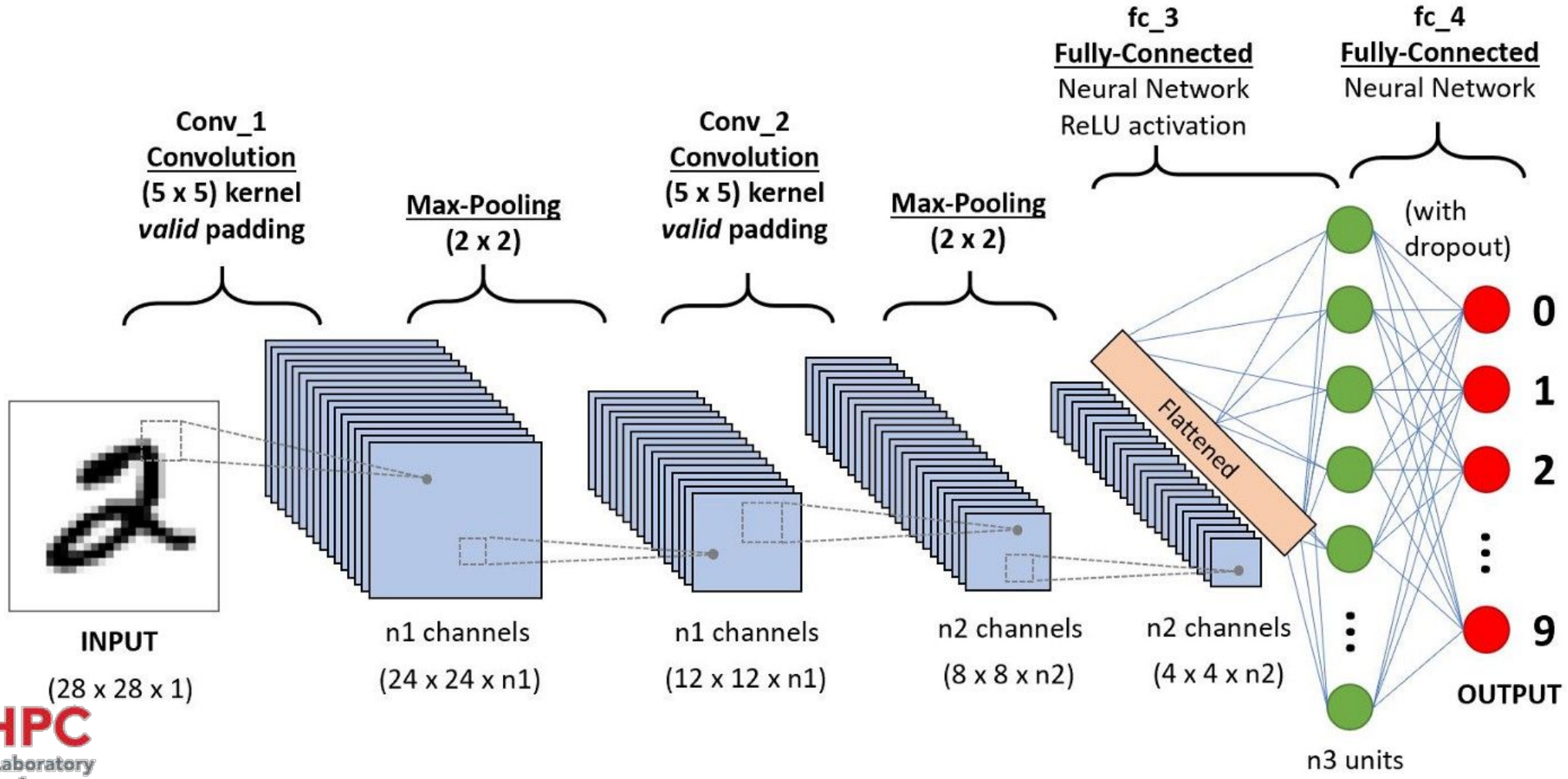


(CR)² Center for Climate and Resilience Research

NLHPC National Laboratory for High Performance Computing Chile

CNN

Redes Neuronales Convolucionales (CNN)

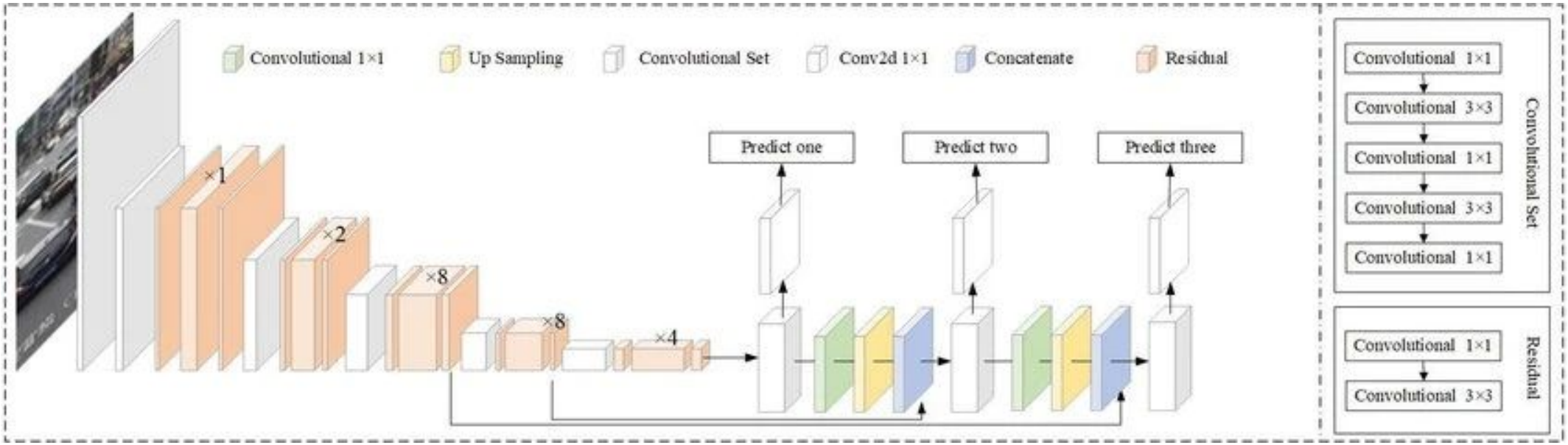


NLHPC
National Laboratory for High Performance Computing Chile

(CR)² Center for Climate and Resilience Research

CNN

CNN: YOLO

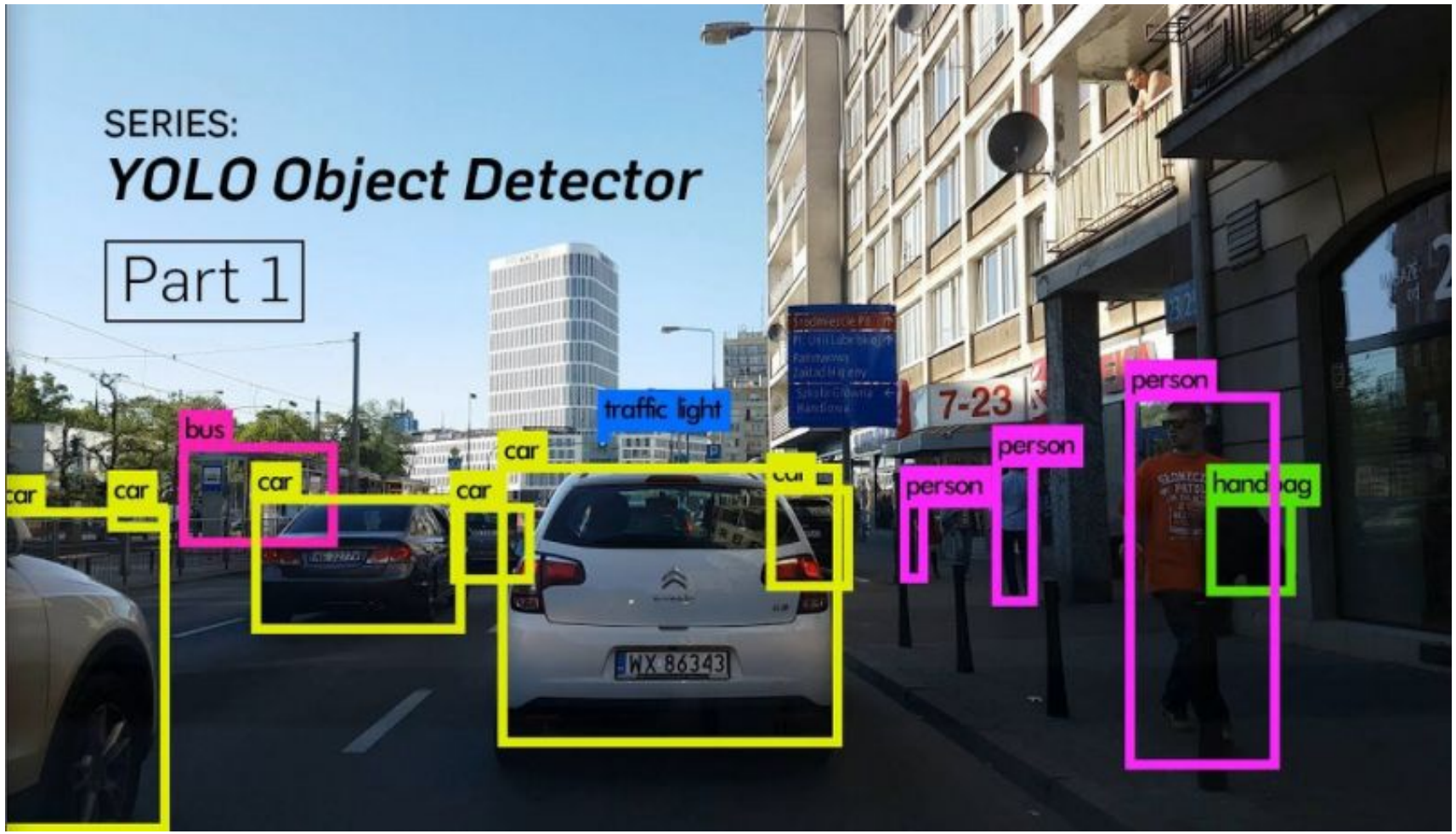


(CR)² Center for Climate and Resilience Research

NLHPC National Laboratory for High Performance Computing Chile

CNN

CNN: YOLO



Center for Climate and Resilience Research

NLHPC
National Laboratory for High Performance Computing Chile

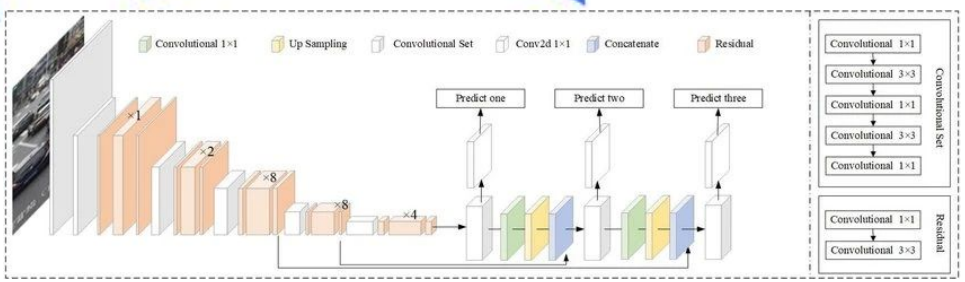
Transfer Learning

Data

Algorithm

Model

$$f(\mathbf{x})$$



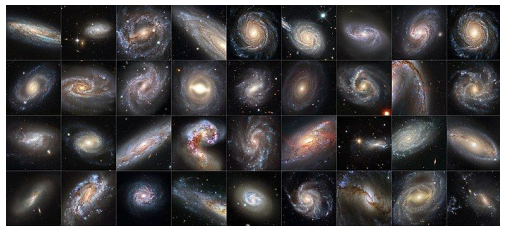
Transfer Learning

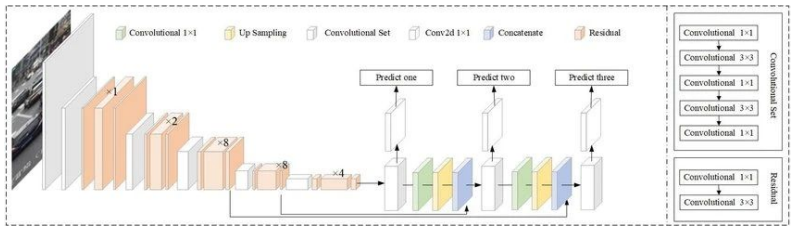
Data

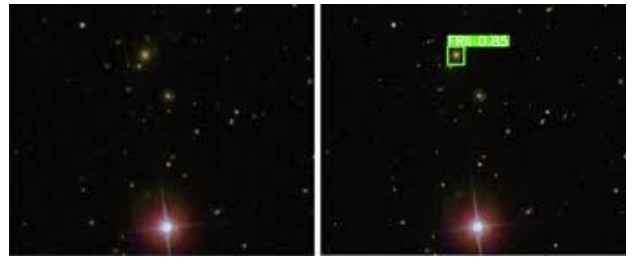
Algorithm

Model

$$f(\mathbf{x})$$







Outline II

- Repaso de Feed Forward Neural Network
- ¿Qué es una LSTM?
- Ejemplo concreto
- Código y actividades

Contenidos

A Repaso de Feed Forward Neural Network

B ¿Qué es una LSTM?

C Ejemplo concreto: Aire en Coyhaique

D Código y actividades

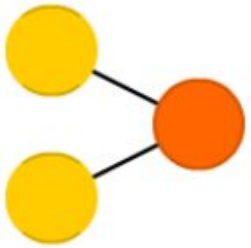
Outline II

- Repaso de FFNN (5 - 10 min)
- FFNN vs LSTM (20 - 30 min)
- Ejemplo concreto (5 - 10 min)
- Codigos (40 - 50 min)

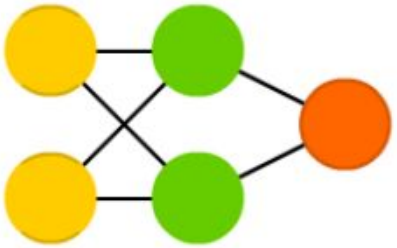
Feed Forward Neural Network (FFNN)

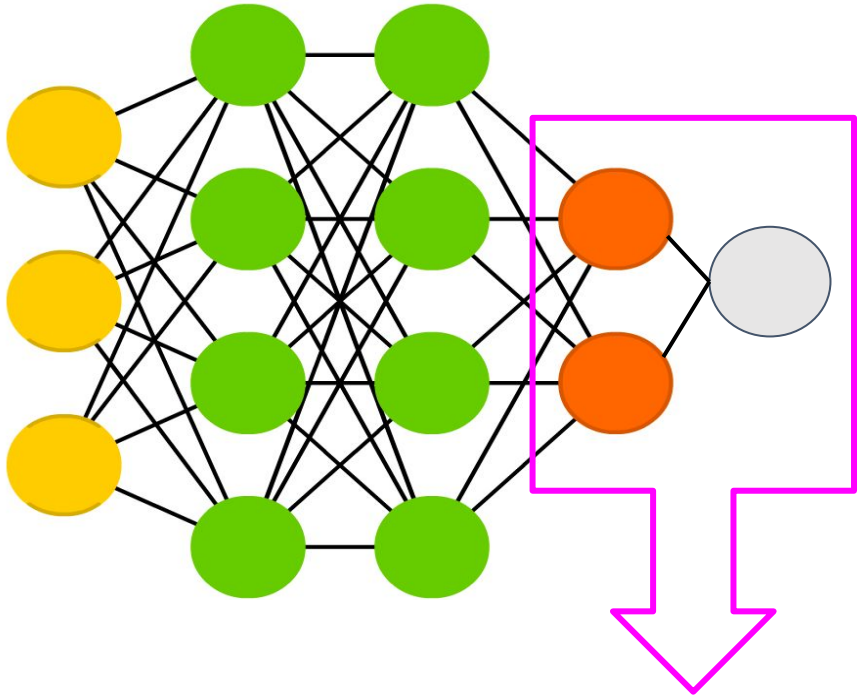
Abstract—*This paper rigorously establishes that standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. In this sense, multilayer feedforward networks are a class of universal approximators.*

Perceptron (P)



Feed Forward (FF)



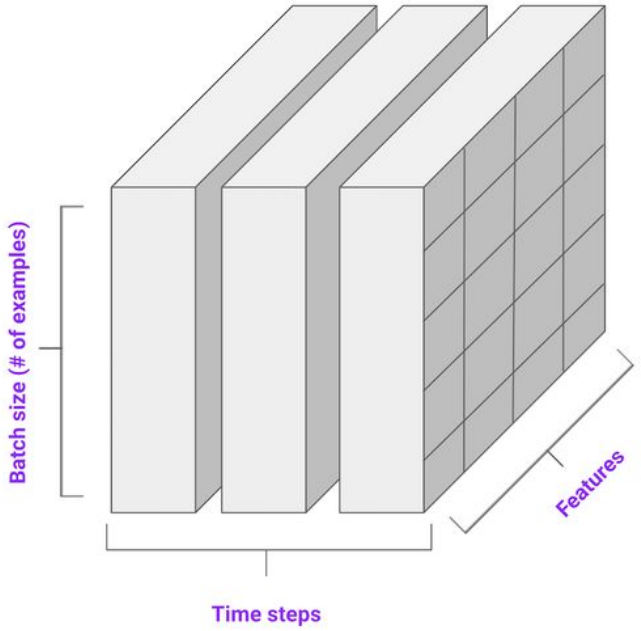


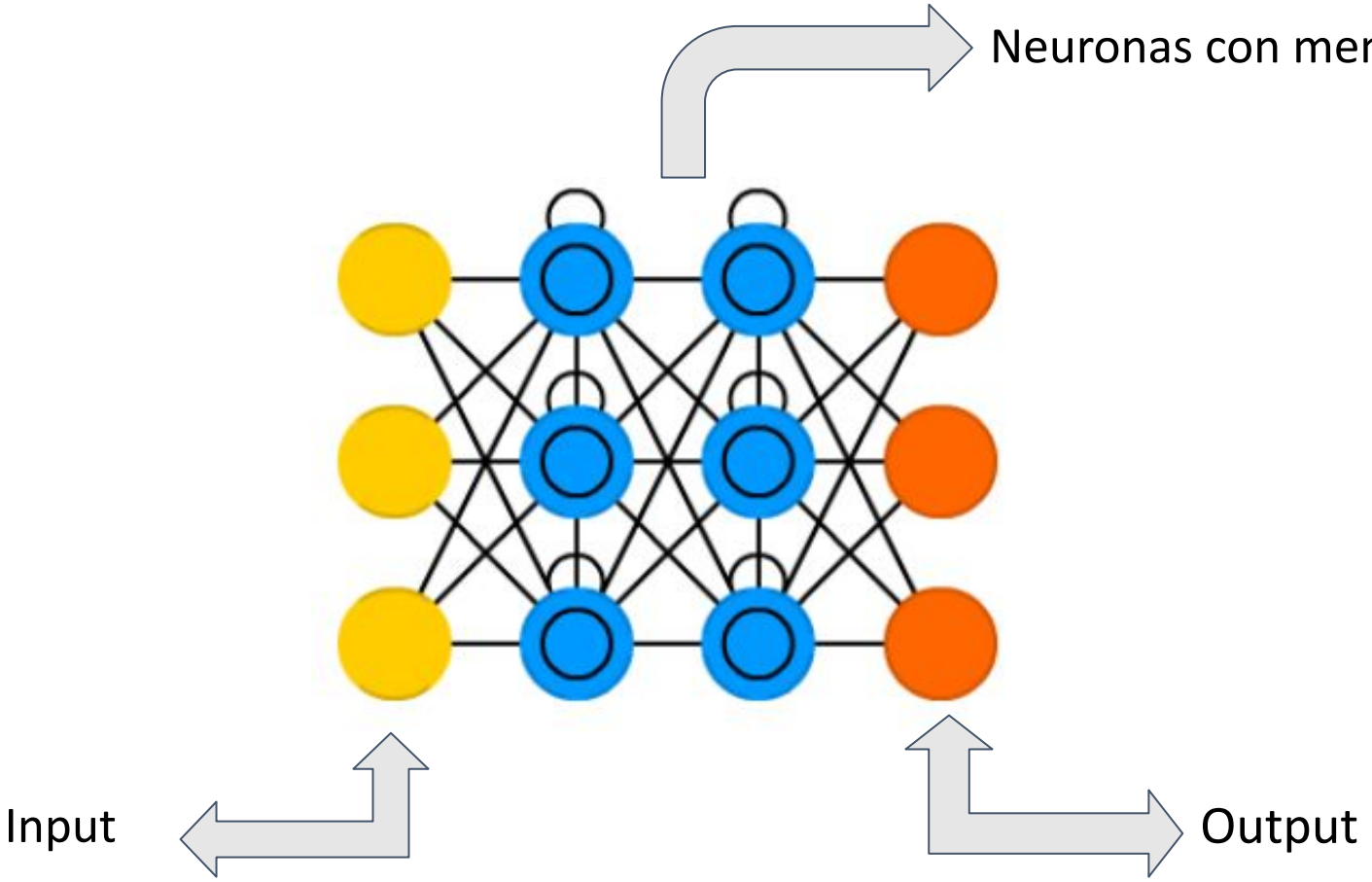
Define tarea de la red

Fuente: Hornik, K., Stinchcombe, M. and White, H. 1989 & The Asimov Institute

¿Qué es una LSTM?

LSTM: Long-short term memory





Fuente: Bpesquet.fr & The Asimov Institute

LSTM

- Cuestionario para adivinar siguiente término (FFNN vs LSTM)

EJ:

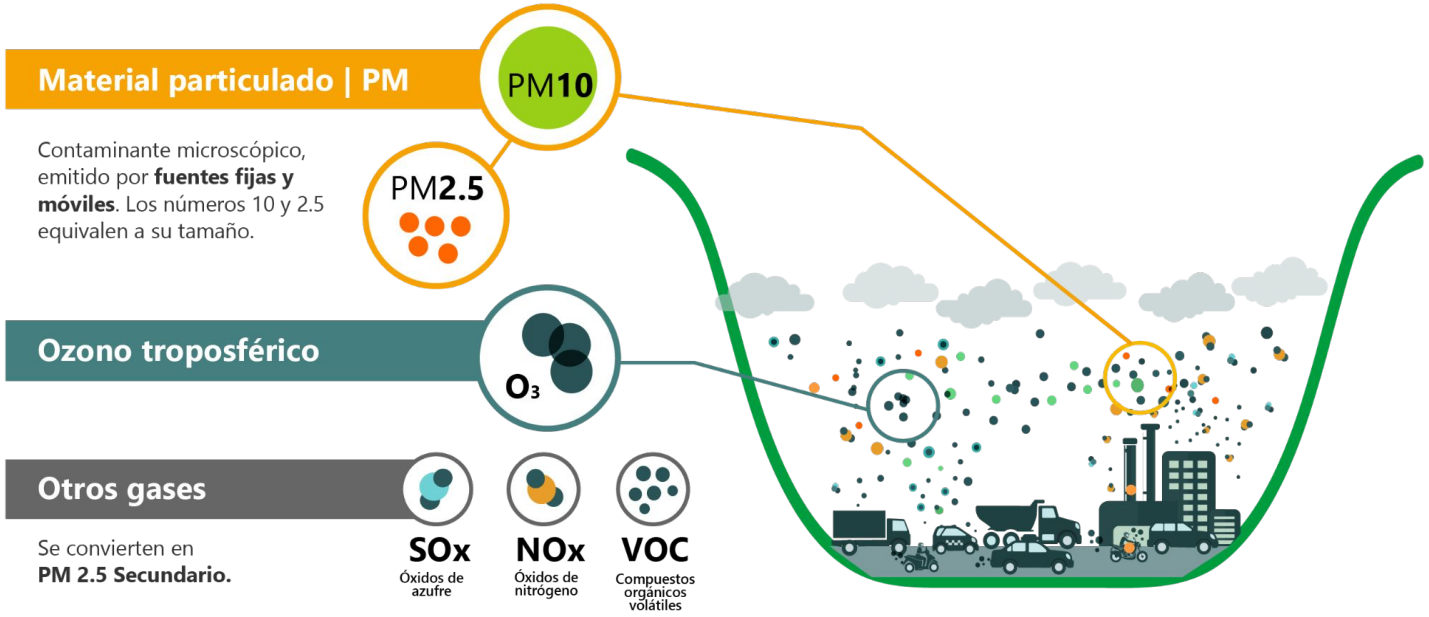
1.- el valor de pm25 hoy es 50, cuál de los siguientes será el valor de pm25 mañana ?: 51,50,49

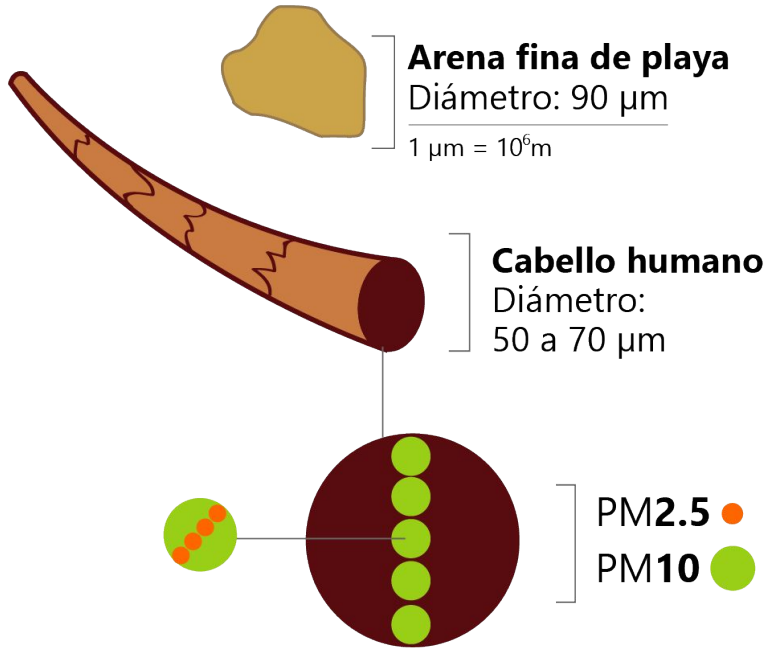
2- el valor de pm 25 de los últimos 3 días ha sido 47, 48, 49, 50, cuál de los siguientes es el valor de pm25 mañana?: 51, 50, 49

- Que es una LSTM

Ejemplo concreto:

Predicción de calidad de aire





Fuente: Metropol.gov.co

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

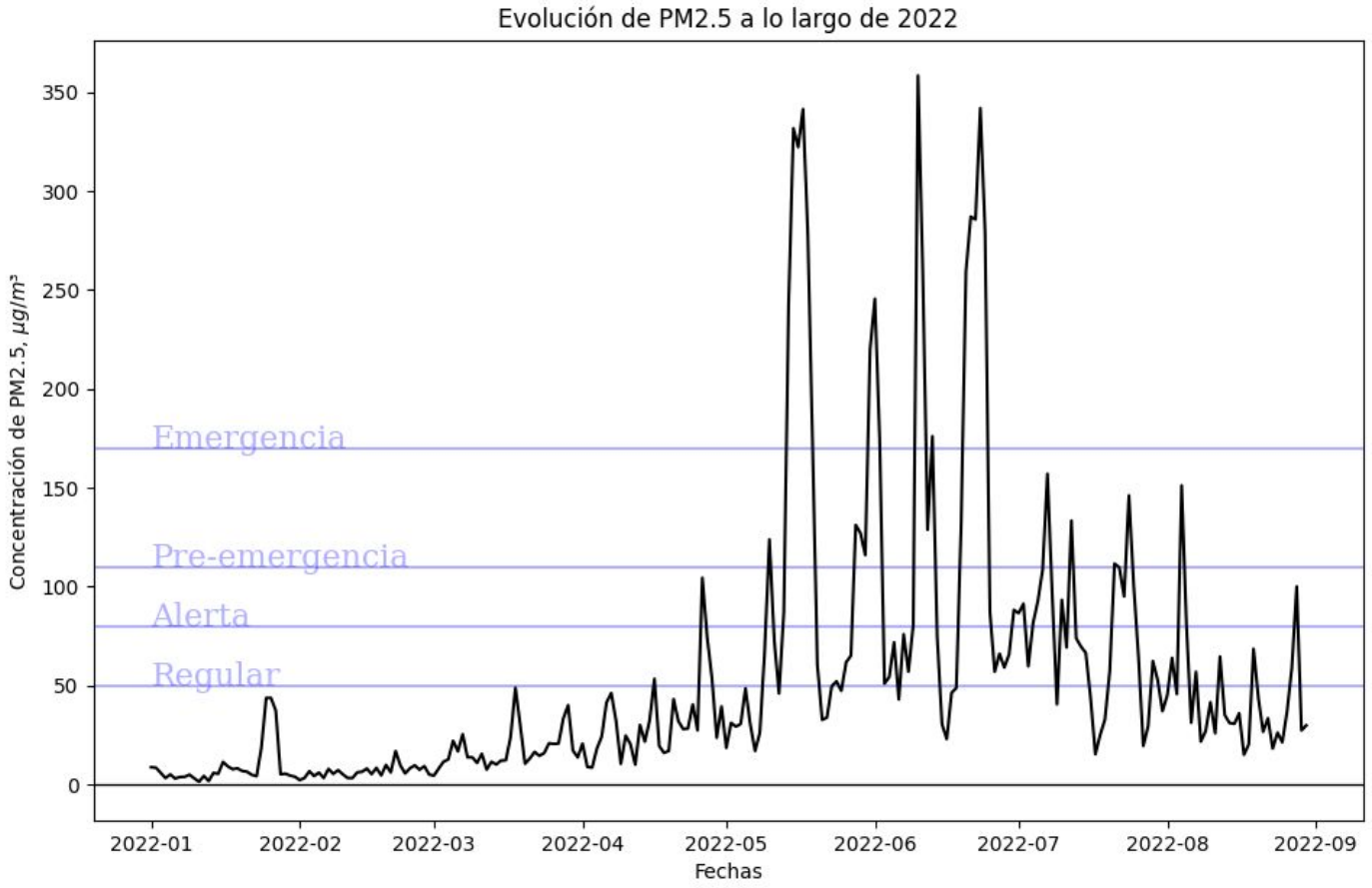
(CR)² Center for Climate and Resilience Research

Ciudades más contaminadas en SA

Rank	City	2021	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	2020	2019	2018	2017
1	Angol, Chile	47.7	7.2	7	3.2	-	56.2	125.3	89.9	67.8	52.9	22.1	8.8	12	-	-	-	-
2	Padre las Casas, C…	37.2	5.2	5.8	11.4	29.6	70.4	83.2	88.3	58.4	38.3	25.4	12.8	10.7	28.6	32.5	43.3	38.8
3	Coyhaique, Chile	36.1	6.5	4.5	11.4	27	83.2	127.3	41.3	60.2	33.8	17.3	11.2	5.1	33.3	41.5	34.2	39.3
4	San Juan de Luriga…	34	23.5	22.3	21.8	28.4	40.8	32.7	39.9	37.6	42.3	34.7	36.1	46.9	22.4	-	-	-
5	Coronel, Chile	33.3	14.6	13.3	17.4	31.8	70	65.4	76.1	46.9	25	15.2	8.7	13.3	20.7	11.8	19	22.9
6	Temuco, Chile	32.7	4.5	6.2	10.4	26.2	63.9	81	80.3	50.6	33.5	19.8	8.1	6.5	20.6	20.2	30.4	28.7
7	Lima, Peru	31.5	12.5	14.6	12.6	19.2	43.9	40.6	46.1	42.9	45.9	37.3	30.8	25.1	18	23.7	28	27.7
8	Traiguen, Chile	30.1	4	5.7	12.6	22.5	46.3	64.1	64	51.3	40	26.9	8.9	8.7	-	-	-	-
9	Nacimiento, Chile	26.2	8.8	8.6	12.8	27.7	52.7	106	65.9	39.3	38.8	14.9	6.8	11.3	27.3	-	-	-
10	Santiago, Chile	25.8	12.3	16.1	15	28	43.8	41	51.1	41.9	21.8	13.9	11	13.3	23.6	27.7	29.4	23.1

Fuente: IQAir

PM2.5 en Coyhaique



Fuentes: sustentable.cl & sinca.mma

Predicción de calidad del aire

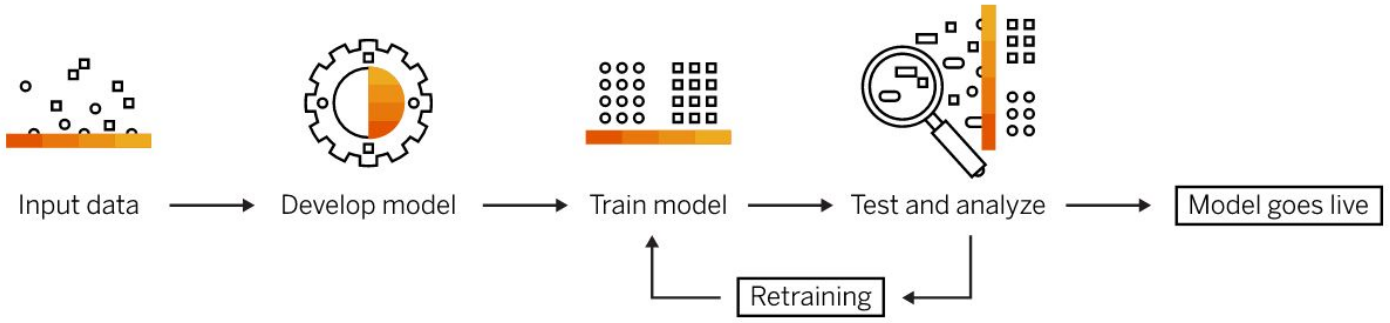
Modelo actual en Coyhaique sólo predice:

59%

BUENO	0-50µm
REGULAR	51-79µm
ALERTA	89-109µm
PRE-EMER.	110-169µm
EMERGENCIA	170µm >

No captura los máximos!

Códigos



Fuente: sap.com

Generar modelo de Deep Learning

1. Creación de arquitectura
2. Entrenamiento
3. Evaluación
4. Predicción

```python
#1.- Arquitectura
model = tf.keras.models.Sequential()

model.add(tf.keras.layers.Dense(1, activation = tf.keras.activations.linear))
model.add(tf.keras.layers.LSTM(1, activation = tf.keras.activations.linear))
model.add...
model.add...
model.add...

#2.- Entrenamiento
model.fit(X_train, Y_train, epochs = 250, batch_size = 100)

#3.- Evaluación
loss, mae, mse, mape = model.evaluate(X_test,Y_test)

#4.- Predicción
pred_never_seen = model.predict(X_never_seen)
```