Lecture 14: Intro to Machine Learning

Economía Laboral

Junghanss, Juan Cruz

Universidad del CEMA

2nd Semester 2022

Contents

Today's lecture content:

- Intro to Supervised Machine Learning
- Intro to Unsupervised Machine Learning
- Intro to Deep Learning and Neural Networks
- Implementation in Python

Intro to Machine Learning

El Machine Learning ("aprendizaje automático" en español) es el término empleado para referirse a las técnicas estadísticas y herramientas computacionales que permiten a las computadoras aprender y adaptarse por sí solas.

$\Longrightarrow$ por eso el término aprendizaje *automático*.

Intro to Machine Learning

A su vez, es la rama de la IA que abarca los estudios de informática y estadística con el fin de desarrollar soluciones a problemas de cualquier disciplina o mercado y que permitan mejorar la eficiencia y/o eficacia de los actuales procesos, escalando en el tiempo.

Intro to Machine Learning

A su vez, es la rama de la IA que abarca los estudios de informática y estadística con el fin de desarrollar soluciones a problemas de cualquier disciplina o mercado y que permitan mejorar la eficiencia y/o eficacia de los actuales procesos, escalando en el tiempo.

En otras palabras, soluciones nuevas o mejores a las actuales en áreas donde antes no era posible llevar a cabo un proceso de tal forma.

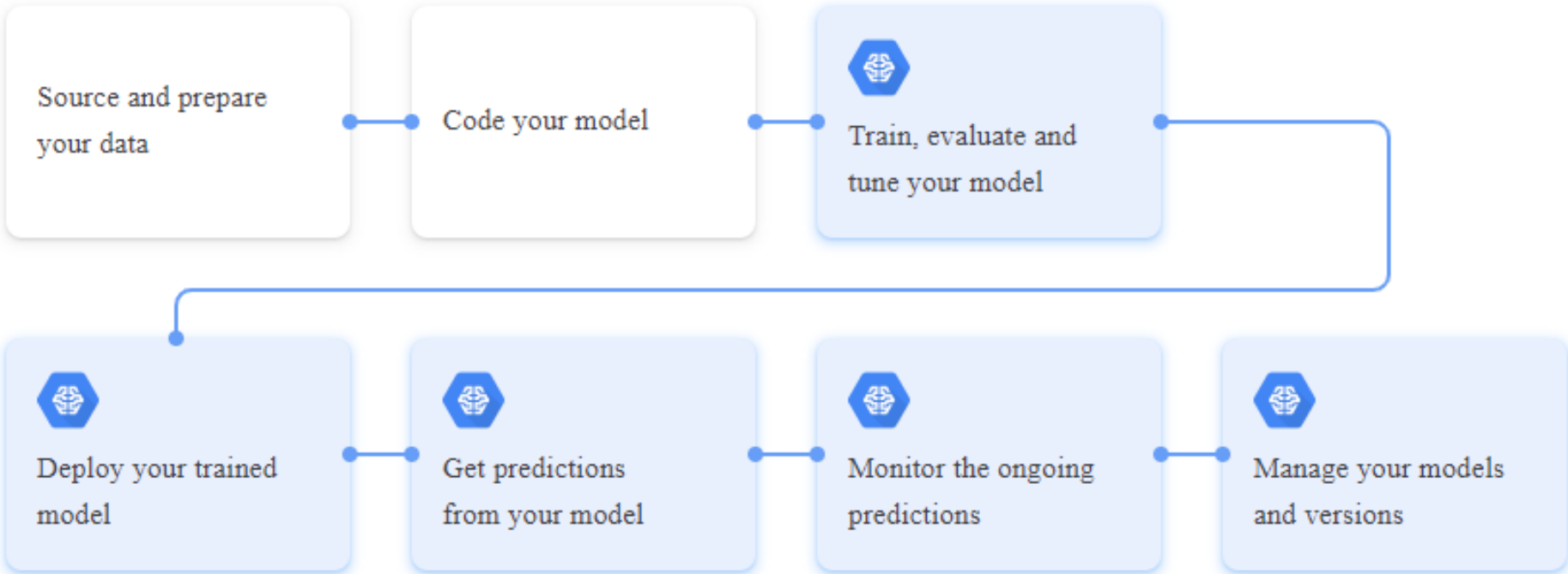
Intro to Machine Learning - Examples

Ejemplos de las aplicaciones más comúnes y tradicionales:

- Motor de recomendaciones
- Reconocimiento de imagenes
- Análisis de sentimiento (rama del procesamiento de lenguaje natural)
- Detección de fraudes
- etc.

Intro to Machine Learning - Workflow

ML workflow:

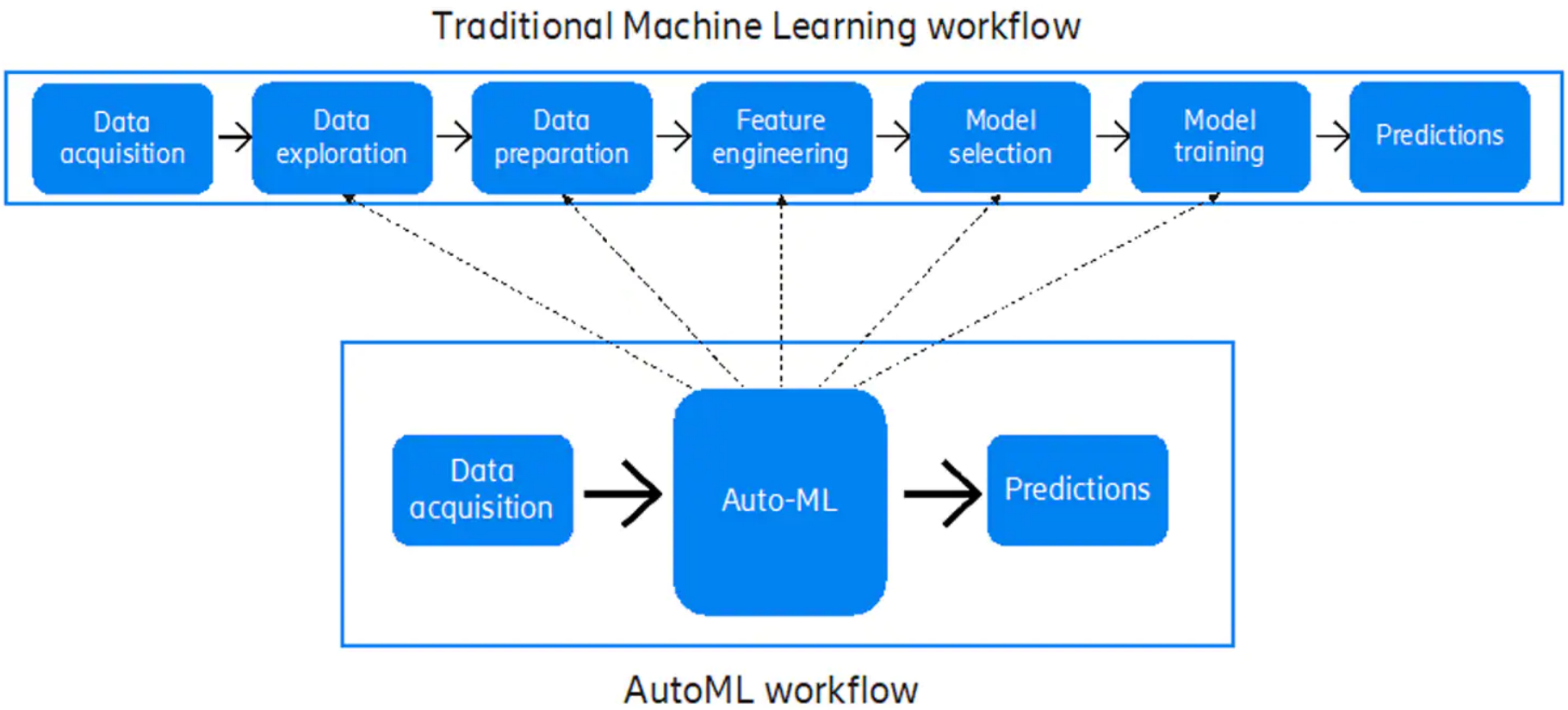


Intro to Machine Learning

Relevant steps of the model development:

1. **Training**: es la parte de optimización matemática del modelo (cuando se minimiza el error de predicción como función objetivo) y se ajusta a los datos del input (training dataset).
2. **Validation**: es el proceso de validar el modelo ya entrenado con datos nuevos para ver su performance (test set).
3. **Tuning**: el finetuning refiere a la optimización de los hiperparámetros del modelo luego de haberlo evaluado. El usuario busca corregir fallas y mejorar debilidades.
4. **Deployment**: una vez entrenado, evaluado y tuneado, el modelo se puede desplegar en la aplicación web, app, cloud, etc. y dejarlo operativo.

ML versus Auto-ML workflow:



Intro to Machine Learning - AutoML

What is Automated Machine Learning?

It's the process of automating the time-consuming and iterative tasks of machine learning model development. It allows data scientists, analysts, and developers to build ML models with high scale, efficiency, and productivity all while sustaining model quality.

The high degree of automation in AutoML aims to allow non-experts to make use of machine learning models and techniques without requiring them to become experts.

Intro to Machine Learning - AutoML

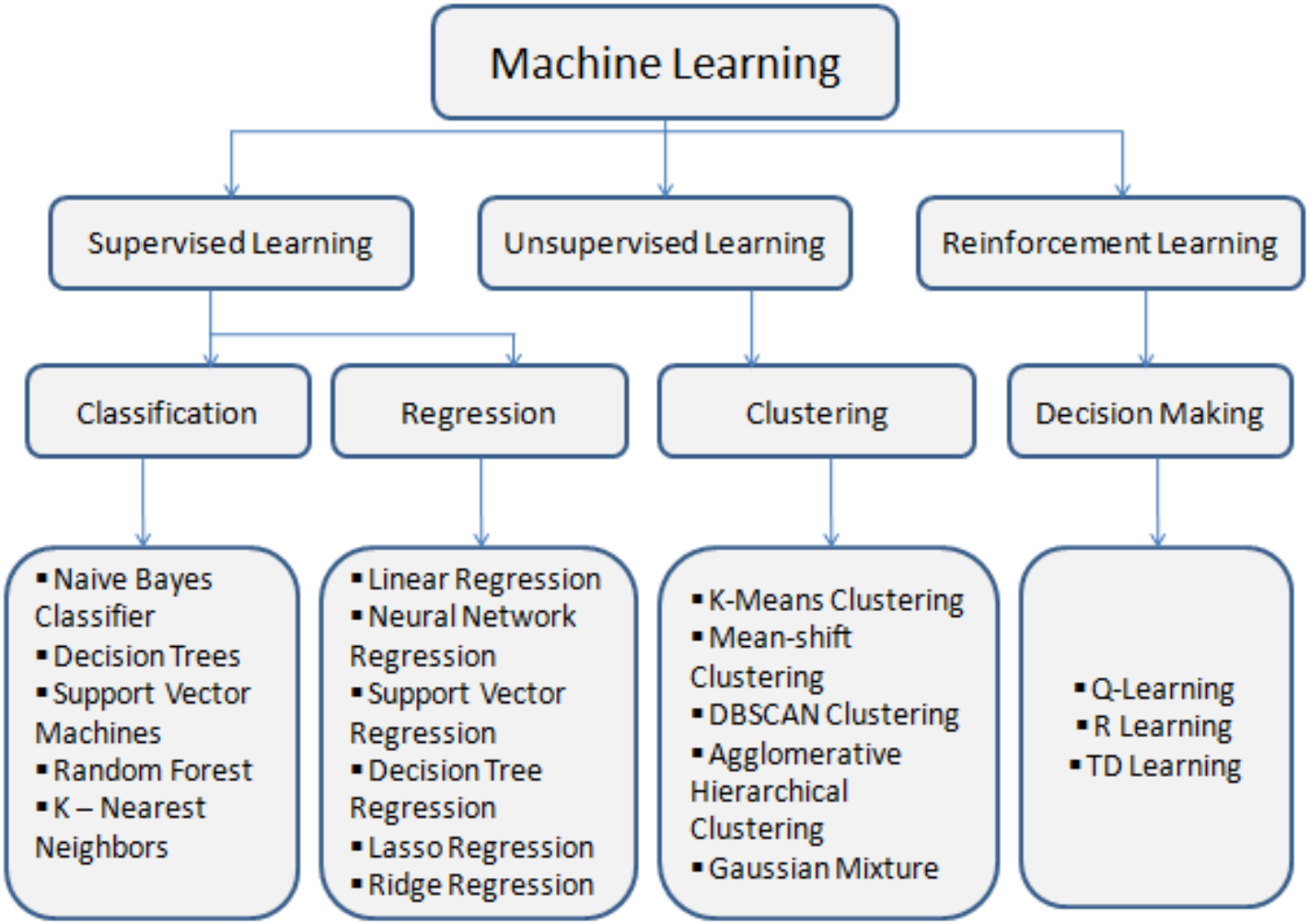
Some AutoML services platforms:

- Azure (Microsoft)
- Google Cloud AutoML
- AWS AutoML (Amazon)
- DataRobot
- BigML

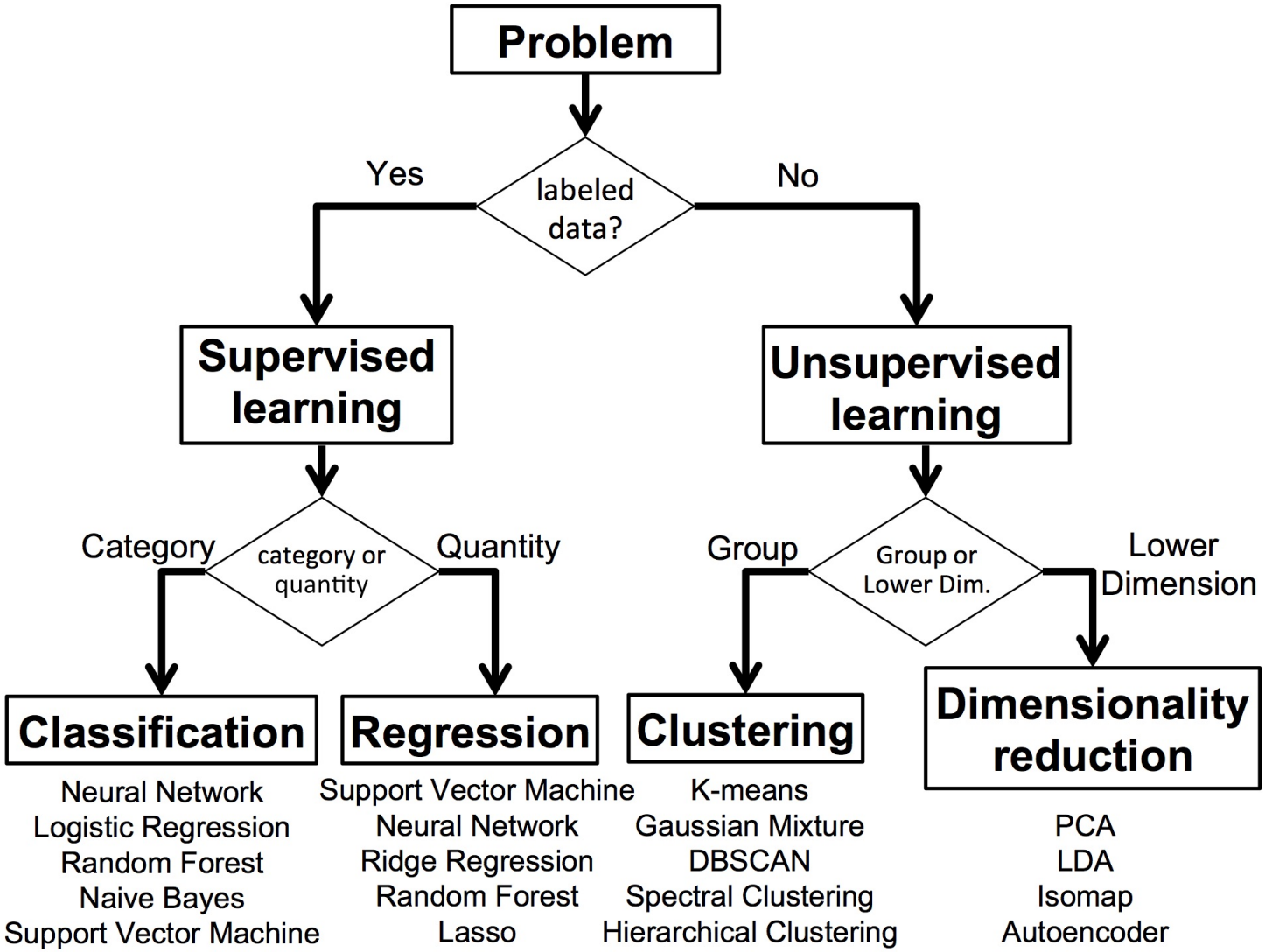
In Python:

- Auto-Sklearn
- Auto-Pytorch
- Auto-Keras
- H20
- Snorkel, MLBox, TPOT, etc.

Intro to Machine Learning



Intro to Machine Learning



Intro to Supervised Learning

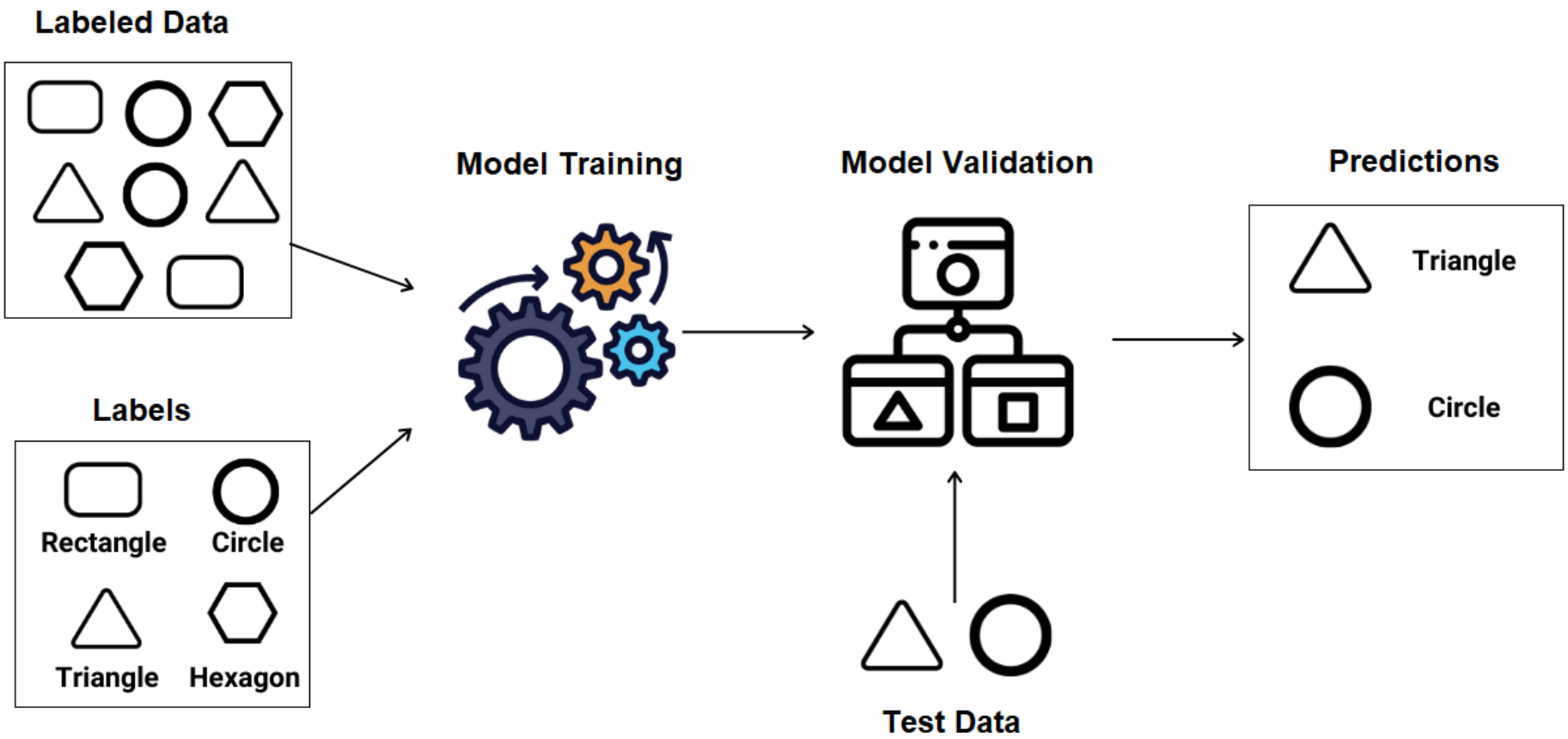
El aprendizaje supervisado (en inglés Supervised Learning) es una técnica que consta de proveer al modelo tanto el input, como el output deseado. En otras palabras, los datos están identificados tanto en las variables explicativas (parámetros o features), como en las variables explicadas (lo que será el output). Esto implica que nuestro modelo podrá interpretar que $X_i \Longrightarrow Y_i$ y encontrar patrones en la relación que hay entre los parámetros y las variables explicadas.

Intro to Supervised Learning

El aprendizaje supervisado (en inglés Supervised Learning) es una técnica que consta de proveer al modelo tanto el input, como el output deseado. En otras palabras, los datos están identificados tanto en las variables explicativas (parámetros o features), como en las variables explicadas (lo que será el output). Esto implica que nuestro modelo podrá interpretar que $X_i \Longrightarrow Y_i$ y encontrar patrones en la relación que hay entre los parámetros y las variables explicadas.

Un ejemplo simple para terminar de entenderlo: las regresiones lineales (o logísticas en el caso de clasificación) pueden basar modelos de supervised learning.

Intro to Supervised Learning



Intro to Supervised Learning

Las formas que toma el supervised learning son:

- **Regresión**: para problemas numéricamente continuos, es decir, cuando necesitamos modelar el patrón y ser capaces de explicar con un numéro específico como resultado el comportamiento de los datos. Se usa, por ejemplo, para las siguientes aplicaciones: estimar la demanda de un producto, predecir el retorno de inversión, los precios de viviendas, etc.

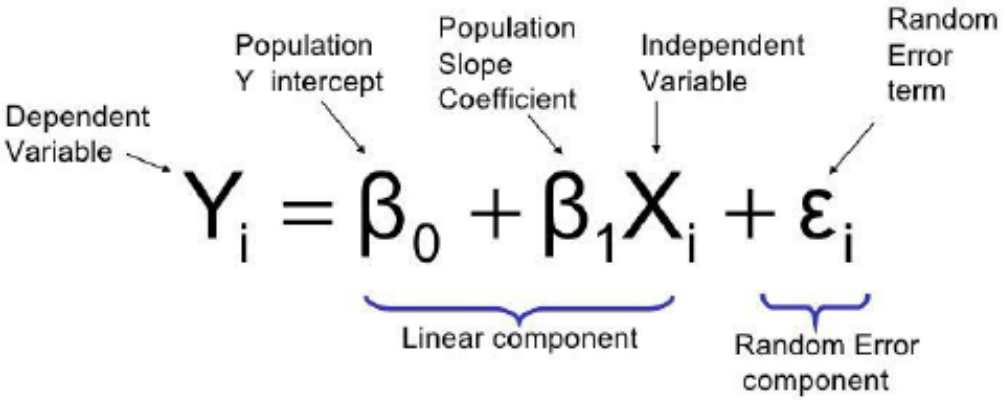
Intro to Supervised Learning

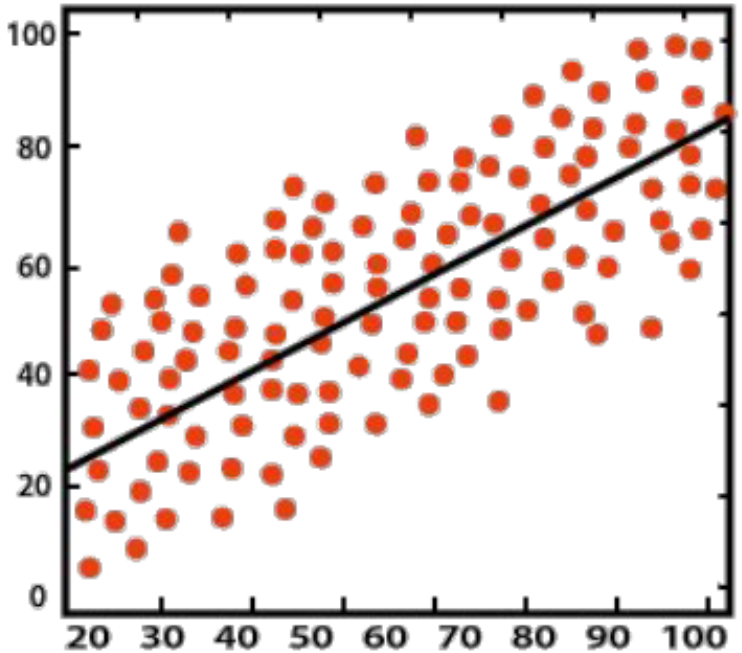
- **Clasificación**: para problemas discretos, es decir, cuando los datos en los cuales hay patrones se pueden clasificar en categorías. El algoritmo puede responder preguntas sencillas de dos opciones (i.e. binarias), como sí o no, verdadero o falso, por ejemplo: ¿Esta persona en la foto está feliz o no?

  Sin embargo, hay también clasificaciones de múltiples categorías, llamadas **multi-class classification**. Las preguntas que puede responder son un poco más complejas, por ejemplo en clasificación de sentimientos para reseñas/críticas de restaurantes de 1 a 5 estrellas. El algoritmo debe ser capaz de identificar si un comentario corresponde a una de cinco categorías.

Intro to Supervised Learning

Para el caso de regresión, ya sabemos que:





El vector óptimo de β_i lo cálculamos al minimizar la "loss function", que es la función que cuantifica cuán mala es una solución de regresión. Por ejemplo, el error cuadrático:

$$L(\beta) = \sum [y_i - (\beta_0 + \beta_1 x_i)]^2$$

Intro to Supervised Learning

¿Cómo minimizamos dicha loss function? Hay muchos algoritmos que iterativamente localizan el mínimo, el más simple es el "gradient descent" (descenso por el gradiente). Lo vamos a analizar luego.

Intro to Supervised Learning

¿Y para el caso de classification? Bastaría con cambiar:

- El modelo (i.e. tipo de regresión) para que el output sean valores binarios (0 y 1). Se logra teniendo una función que mapee el intervalo [0,1] y determinando un umbral de corte.

$$f(x_i) = sigmoid\left(\beta_0 + \beta_1 x_i\right)$$

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

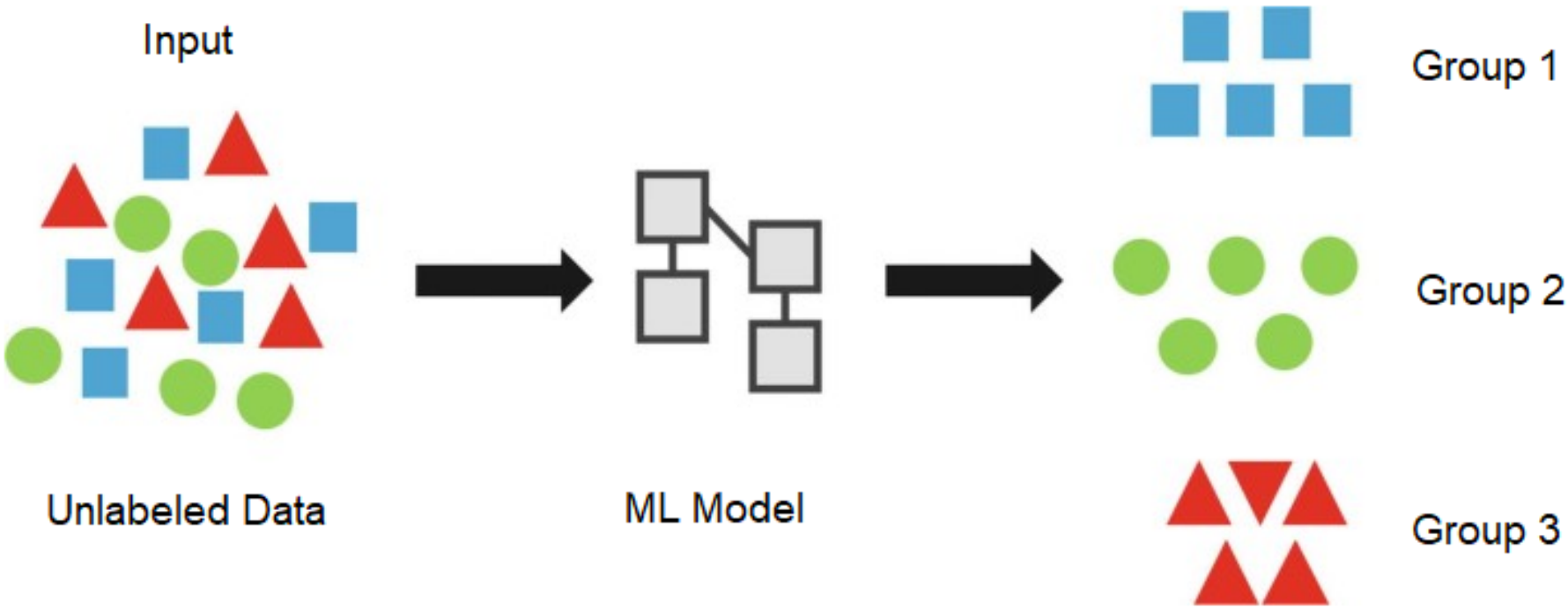
- La loss function para que represente mejor el problema:

$$\sum y_i \log\left(f(x_i)\right) + (1 - y_i) \log\left(1 - f(x_i)\right)$$

Intro to Unsupervised Learning

El Unsupervised Learning es una técnica que consume como input datos que solamente están identificados por las variables explicativas (es decir, las características), pero no hay un output en concreto para estos. El modelo devolverá como resultado agrupaciones por similitud que capturan la *información más importante* del conjunto de datos.

Intro to Unsupervised Learning



Intro to Unsupervised Learning

Hay distintos tipos de unsupervised learning, como el análisis por agrupaciones (clusters), detección de anomalías y analísis de componentes principales (PCA):

- **Cluster analysis**: es un tipo de aprendizaje no supervisado que separa puntos de datos similares en grupos intuitivamente identificables. Se suele usar, por ejemplo, para segmentar clientes en negocios, predecir o identificar las preferencias de los clientes, etc.

Intro to Unsupervised Learning

- **Anomaly detection**: la detección de anomalias implica identificar o predecir comportamientos raros o poco usuales, es decir, puntos de datos que no son "normales". Básicamente el enfoque es que el algoritmo aprenda cómo se ve la actividad normal (usando datos históricos de cierta variable) y luego así podrá identificar cuando un punto es significativamente diferente. Como ejemplos, se puede considerar la detección de fraude, la predicción de riesgo, etc.

- **Principal component analysis (PCA)**: por medio de métodos algebraicos reduce la dimensionalidad del espacio de características con menos variables (no correlacionadas), que se denominan componentes principales. Es completamente útil cuando se requiere responder preguntas sobre las relaciones entre variables.

Intro to Deep Learning

Definition:

Deep Learning is a type of machine learning based on **artificial neural networks** in which multiple layers of processing are used to extract progressively higher level features from data.

Intro to Deep Learning

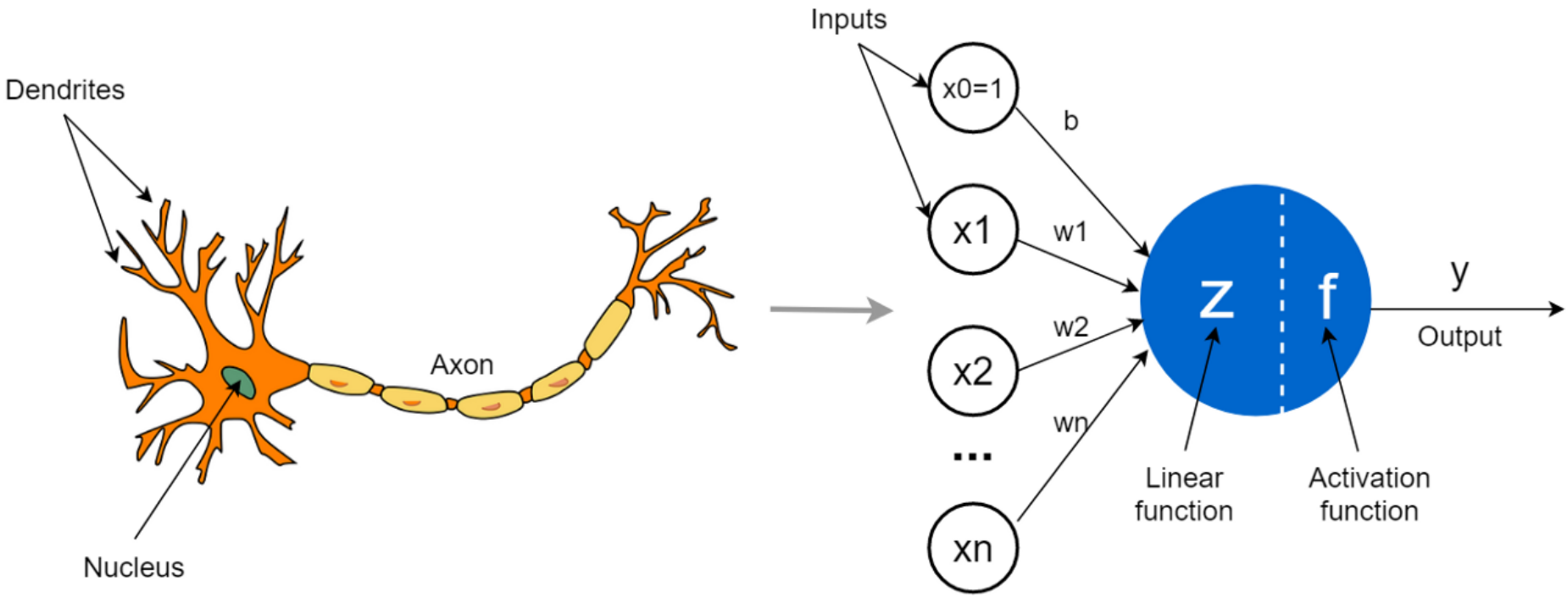
Cuando hablamos de Deep Learning o aprendizaje profundo nos referimos a una de las técnicas más populares en ML que abarca la utilización de modelos con arquitecturas más complejas. Pueden ser construidos con Neural Networks, de una dimension relativamente grande, es decir, que poseen muchas capas (layers) y neuronas.
A su vez, dentro de las redes neuronales es posible encontrar distintos tipos, como las Convolucionales (CNN) o Recurrentes (RNN).

Machine Learning versus Deep Learning

CHARACTERISTIC CATEGORY	ALL MACHINE LEARNING	ONLY DEEP LEARNING
Number of data points	Can use small amounts of data to make predictions.	Needs to use large amounts of training data to make predictions.
Hardware dependencies	Can work on low-end machines. It doesn't need a large amount of computational power.	Depends on high-end machines. It inherently does a large number of matrix multiplication operations. A GPU can efficiently optimize these operations.
Featurization process	Requires features to be accurately identified and created by users.	Learns high-level features from data and creates new features by itself.
Learning approach	Divides the learning process into smaller steps. It then combines the results from each step into one output.	Moves through the learning process by resolving the problem on an end-to-end basis.
Execution time	Takes comparatively little time to train, ranging from a few seconds to a few hours.	Usually takes a long time to train because a deep learning algorithm involves many layers.
Output	The output is usually a numerical value, like a score or a classification.	The output can have multiple formats, like a text, a score, or a sound.

Deep Learning: Intro to Artificial Neural Networks (ANN)

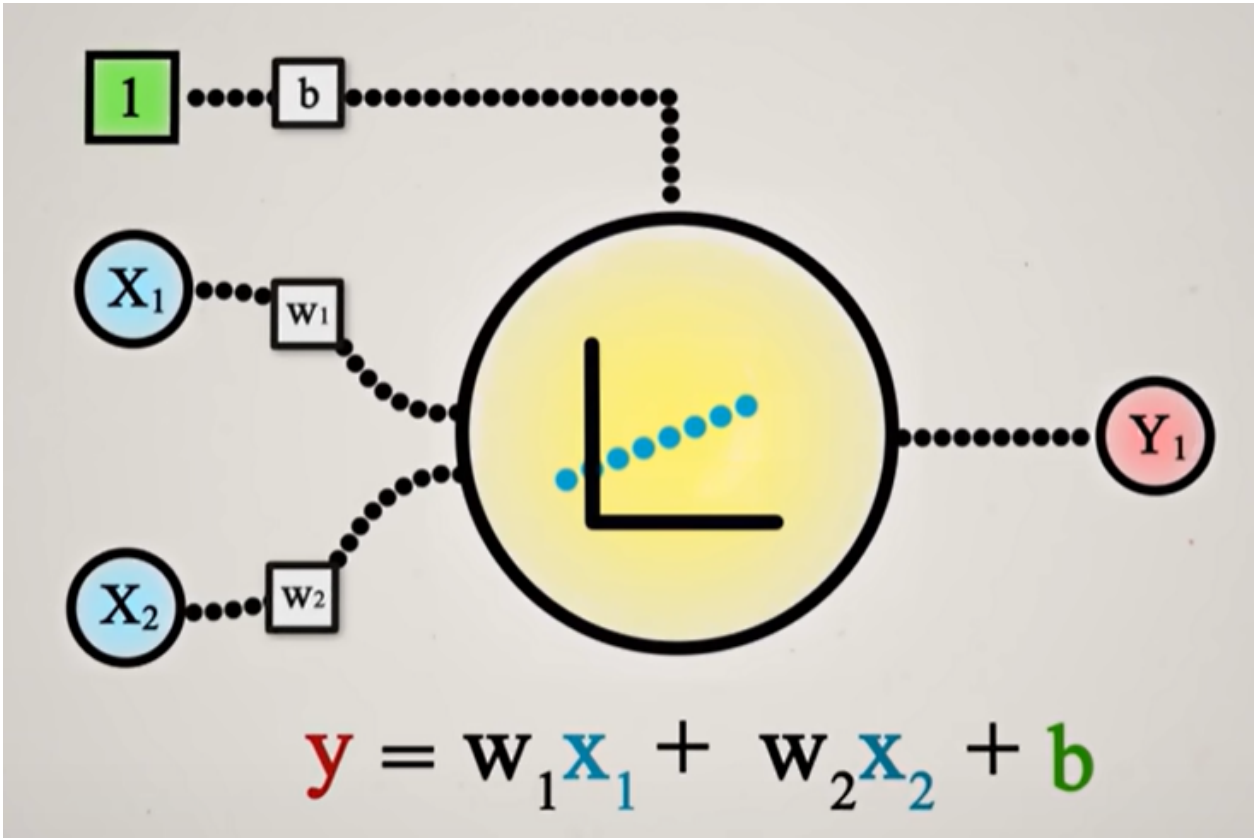
¿Qué es una red neuronal artificial?



Single neuron = linear regression without applying activation ("Perceptron")

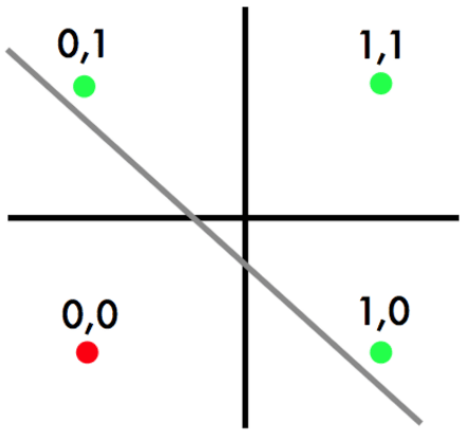
Deep Learning: Intro to Artificial Neural Networks (ANN)

Como ya indicamos anteriormente, una neurona no es ni más ni menos que una función de regresión lineal (pero distorsionada no linealmente por una "función de activación" como veremos luego).

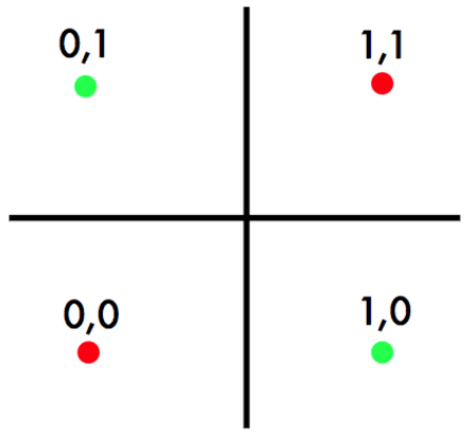


Deep Learning: Intro to Artificial Neural Networks (ANN)

¿Y por qué necesitaríamos múltiples regresiones interactuando entre sí en una red? La intuición yace en que si bien podremos resolver problemas con una regresión lineal, otros muchos no tendrán solución.



OR

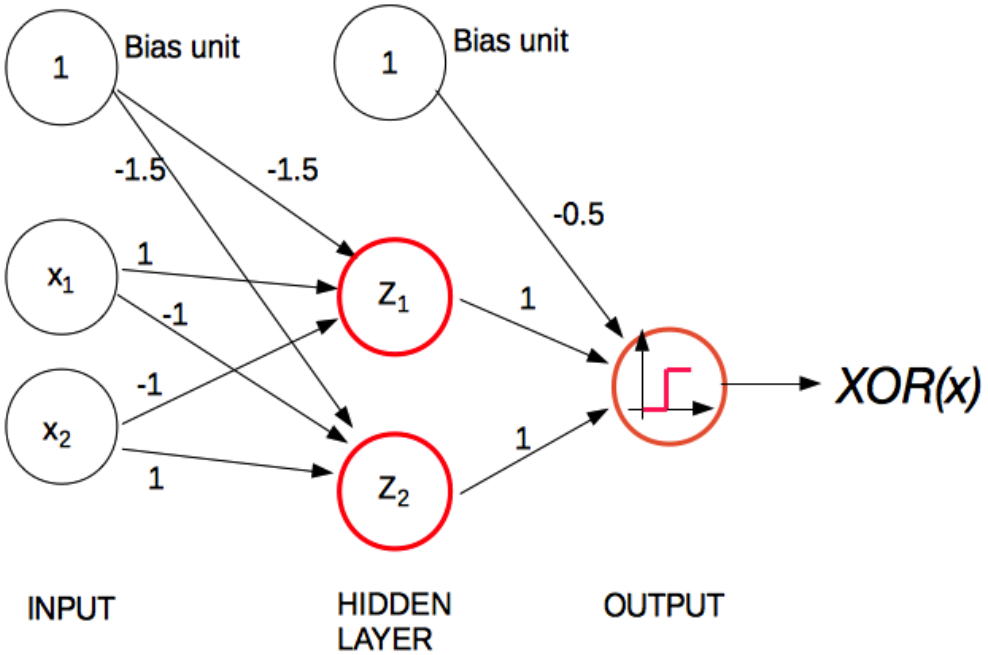


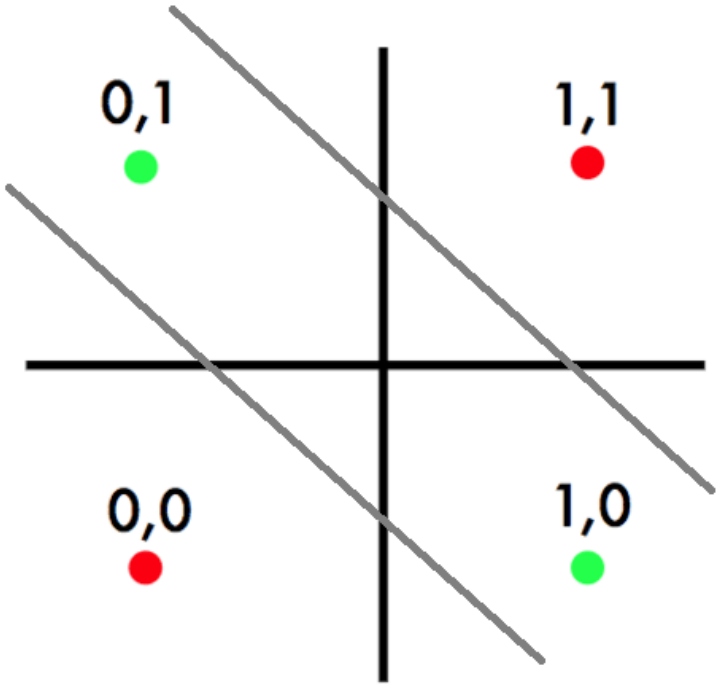
XOR

Un básico ejemplo son las puertas lógicas OR y XOR. La primera se puede resolver con una regresión (separar los 1 de 0), pero la segunda no. ¿Qué hacemos?

Deep Learning: Intro to Artificial Neural Networks (ANN)

Si usamos dos neuronas, i.e. dos regresiones, podemos llegar a la solución del problema:





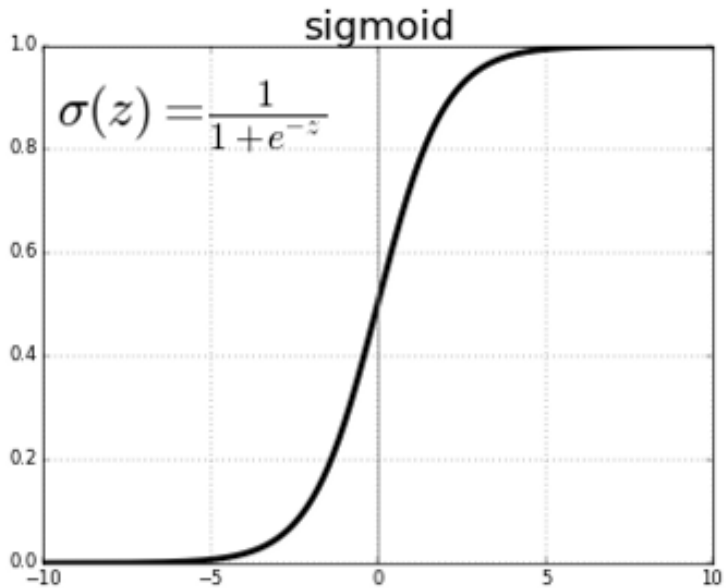
Un interrogante válido es ¿cómo podemos estar sumando N regresiones lineales si por propiedad matemática el resultado será igual a otra regresión lineal?

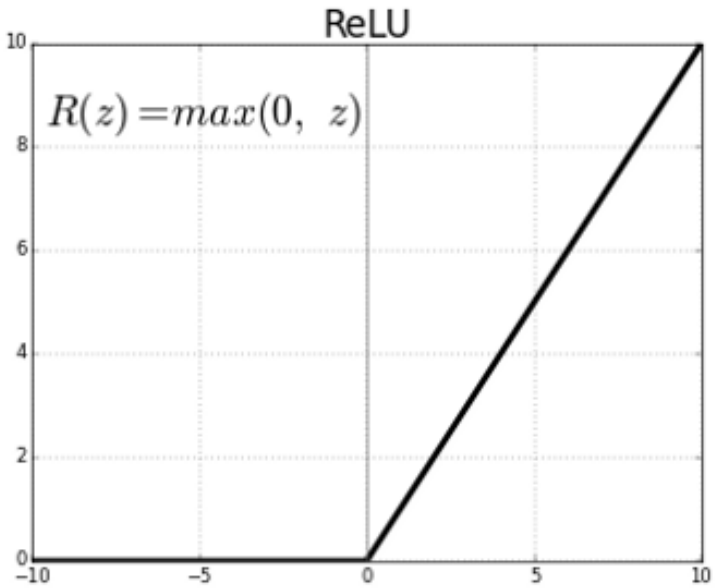
Aquí entran en juego las funciones de activación dentro de la neurona, también denominadas no-linearidades.

Deep Learning: Intro to Artificial Neural Networks (ANN)

Una función de activación mapea el resultado de la regresión en valores entre 0 y 1, -1 y 1, etc. El intervalo dependerá de la función elegida. Las más usadas son:

- Función logística o sigmoide
- Tangentehiperbolica (tanh)
- Rectified Lienar Unit (ReLU)

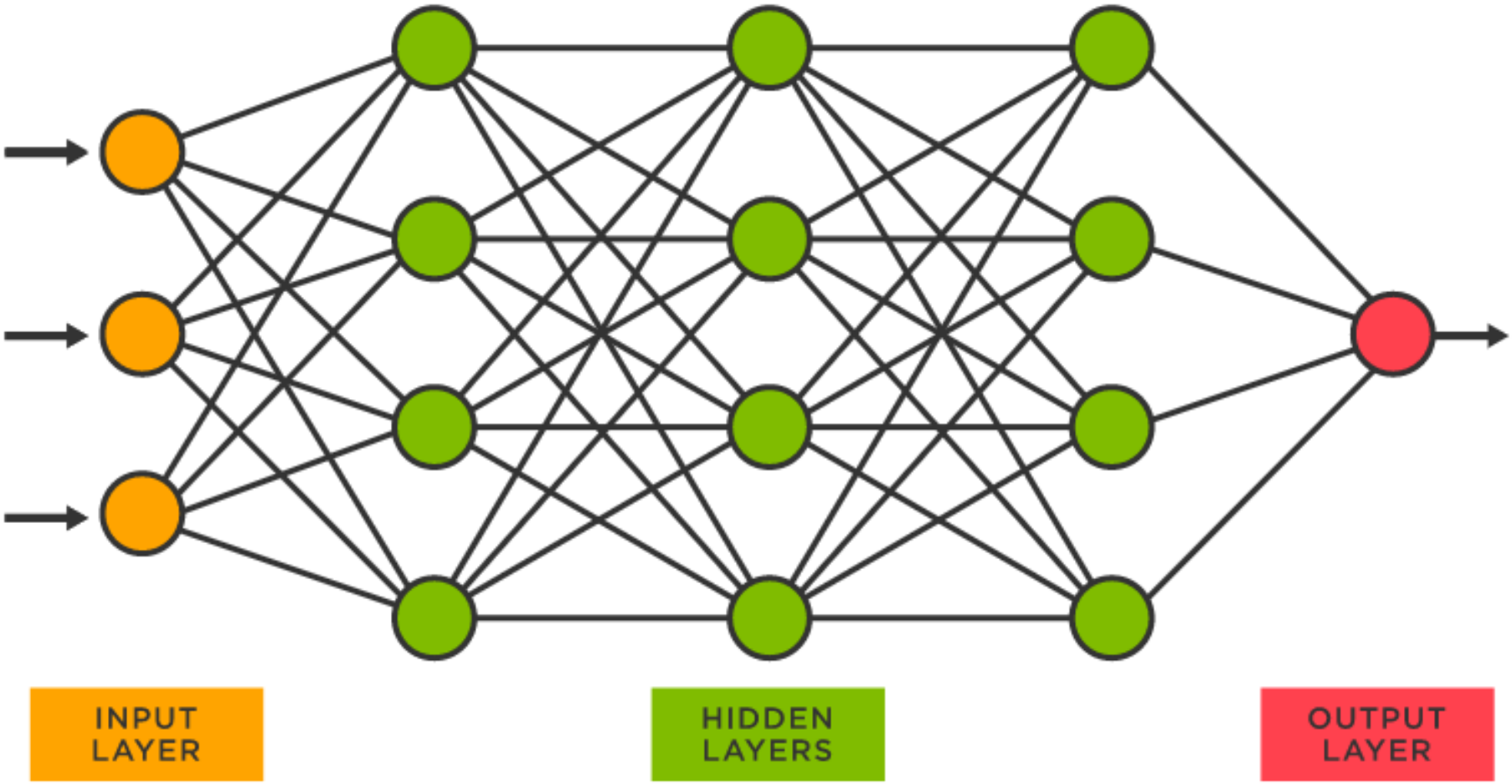




Listado de funciones

Deep Learning: Intro to Artificial Neural Networks (ANN)

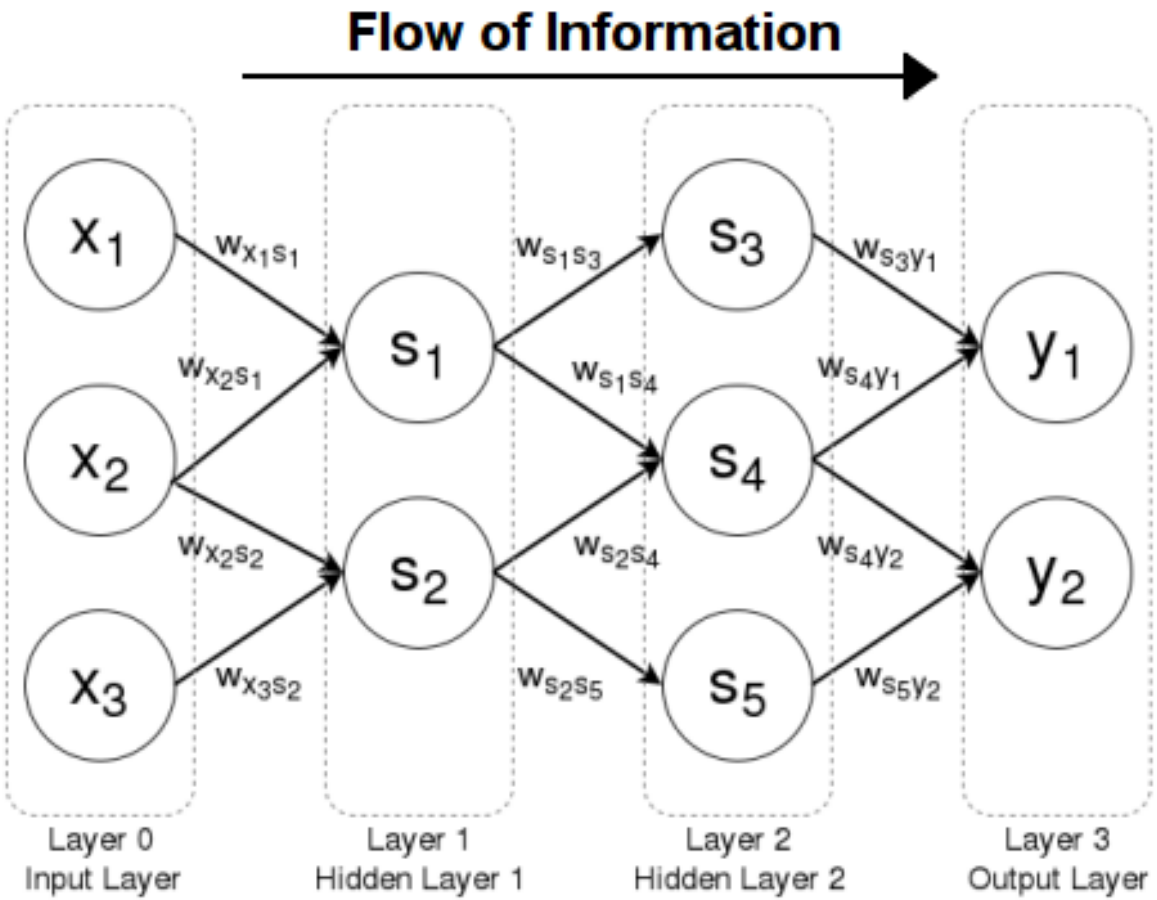
Entonces, generalizando el caso a N neuronas:



La estructura se divide en capas: la del input, la del output y las "hidden layers" que son múltiples neuronas.

Deep Learning: Intro to Artificial Neural Networks (ANN)

¿Cómo se ve una red neuronal cuando entra en funcionamiento? Desde la input layer, los subsiguientes outputs de las capas neuronales empiezan a fluir hacia delante (*feedforward propagation*).



Deep Learning: Intro to Artificial Neural Networks (ANN)

Nos queda un último punto por responder: ¿cómo aprenden las redes neuronales? En definitiva, es el propósito del machine learning, que haya aprendizaje y no solo eso, que sea automático.

El aprendizaje se lo puede pensar como un proceso de dos pasos:

1. Computar un error del resultado obtenido.
2. Propagar ese error calculado a cada neurona para "penalizarla" (i.e. actualizarla).

Para esto último hay diversos algoritmos, más específicamente para el cómputo del error o más generalmente para la minimización de la función de error (la loss function presentada anteriormente).

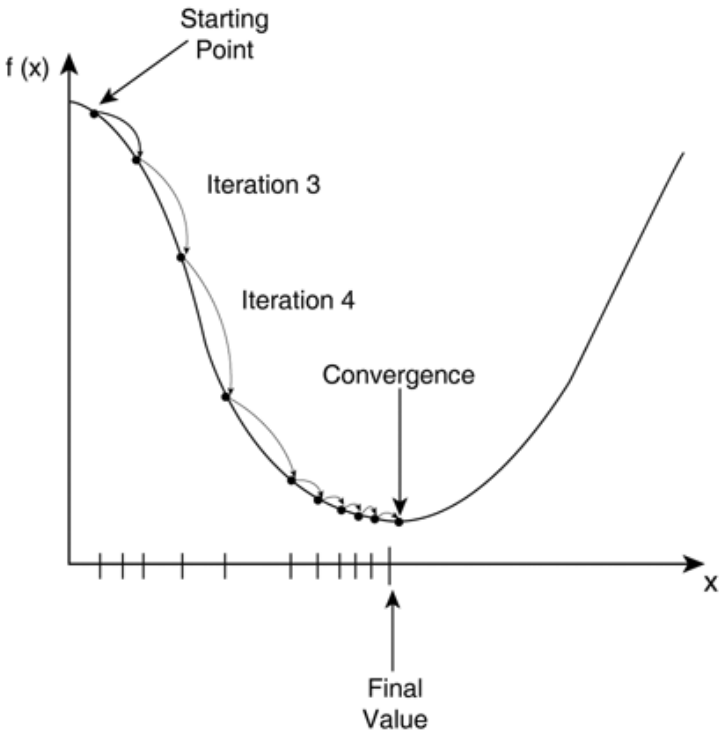
Analicemos entonces dos algoritmos importantes para este proceso de aprendizaje;

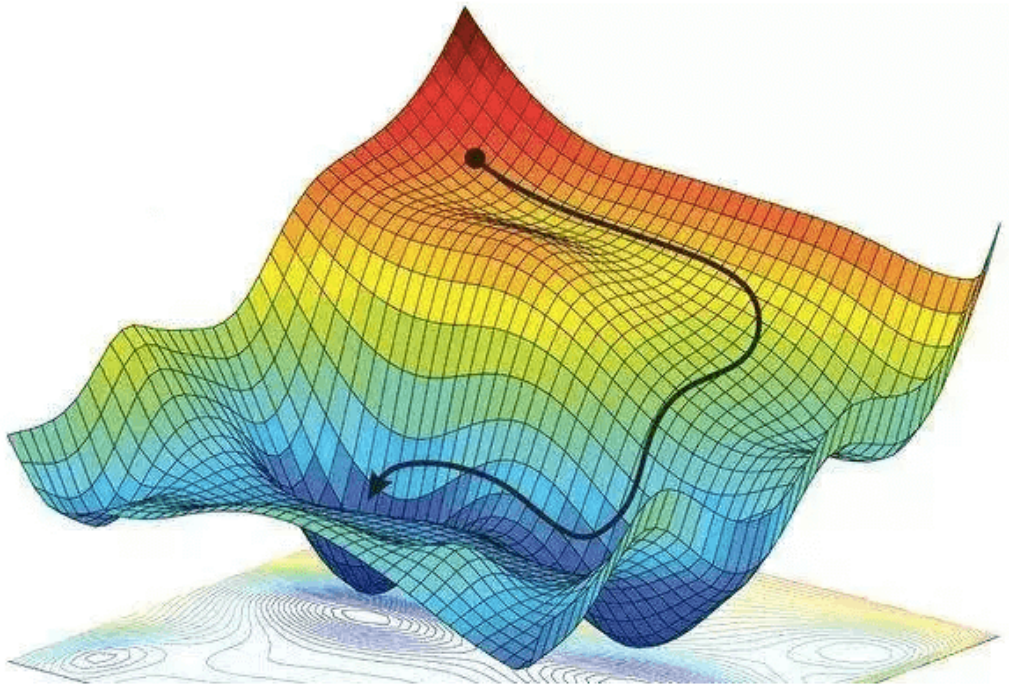
1. Gradient Descent (descenso del gradiente)
2. Backpropagation

Gradient Descent:

- Definición matemática: first-order iterative optimization algorithm for finding a local minimum of a differentiable function.
- Definición intuitiva: take repeated steps in the opposite direction of the gradient of the function at a current point, because this is the direction of steepest descent. Hence, we'll find a local or global minimum.

Deep Learning: Intro to Artificial Neural Networks (ANN)





Cost Function

$$J(\Theta_0, \Theta_1) = \frac{1}{2m} \sum_{i=1}^{m} [h_\Theta(x_i) - y_i]^2$$

$h_\Theta(x_i)$: Predicted Value

y_i: True Value

Gradient Descent

$$\Theta_j = \Theta_j - \alpha \frac{\partial}{\partial \Theta_j} J(\Theta_0, \Theta_1)$$

α: Learning Rate

Now,

$$\frac{\partial}{\partial \Theta} J_\Theta = \frac{\partial}{\partial \Theta} \frac{1}{2m} \sum_{i=1}^{m} [h_\Theta(x_i) - y]^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} (h_\Theta(x_i) - y) \frac{\partial}{\partial \Theta_j} (\Theta x_i - y)$$

$$= \frac{1}{m} (h_\Theta(x_i) - y) x_i$$

Therefore,

$$\Theta_j := \Theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} [(h_\Theta(x_i) - y) x_i]$$

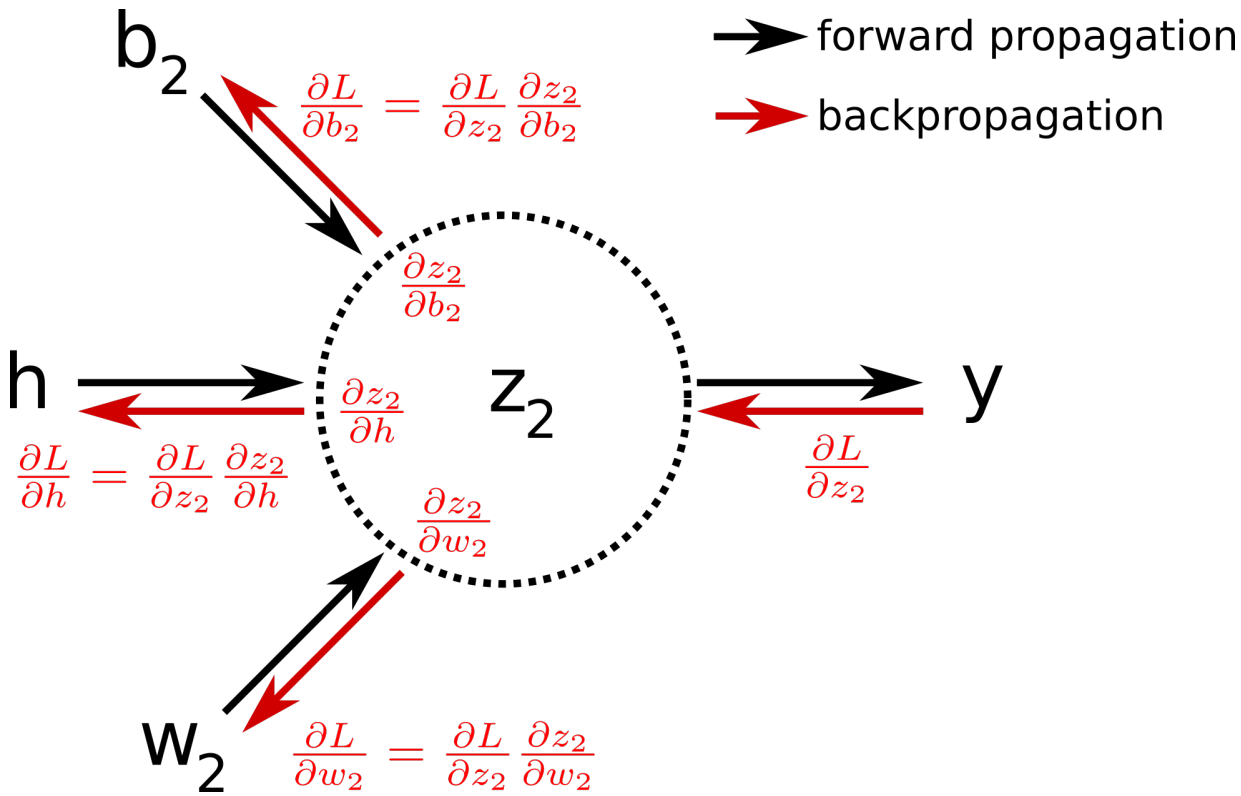
Deep Learning: Intro to Artificial Neural Networks (ANN)

Otros algoritmos de optimización:

- Stochastic Gradient Descent
- Minibatch Stochastic Gradient Descent
- Momentum
- Adagrad
- RMSProp
- Adadelta
- Adam

Deep Learning: Intro to Artificial Neural Networks (ANN)

Nos resta entender cómo las neuronas pueden actualizarse a partir de un error computado al final (luego de que el output haya sido calculado) con el Backpropagation algorithm;



Deep Learning: Intro to Artificial Neural Networks (ANN)

El aprendizaje no es ni más ni menos que un proceso de computar las derivadas respecto los parámetros, pero que será en cadena porque:

- En una neurona tenemos una activation function de una función lineal de parámetros, ergo tendremos que usar la regla de la cadena.
- En la red neuronal, tenemos neuronas que le transmiten el output a otras y así sucesivamente a lo largo de las capas. Para llegar a los parámetros iniciales debemos atravesar n neuronas, por lo que nuevamente tendremos que usar la regla de la cadena.

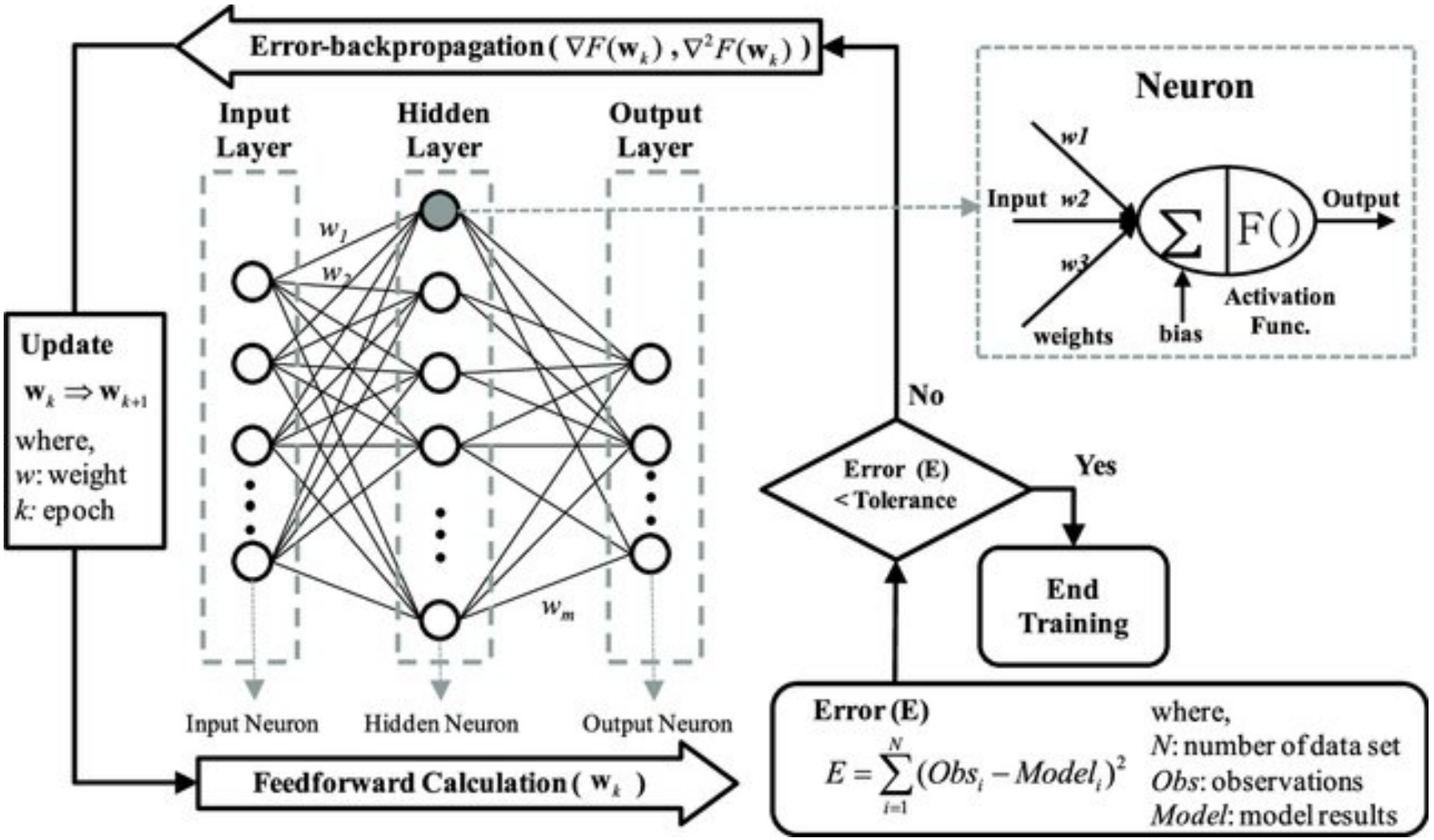
Entonces, calcular la derivada parcial del error respecto a un parámetro (weight) w_{ij} se hace utilizando la regla de la cadena dos veces:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial \text{net}_j}\frac{\partial \text{net}_j}{\partial w_{ij}}$$

Lo laborioso será a travesar la red, pero con álgebra matricial y programación se hace en solo un par de líneas de código.

Deep Learning: Intro to Artificial Neural Networks (ANN)

En resumen,



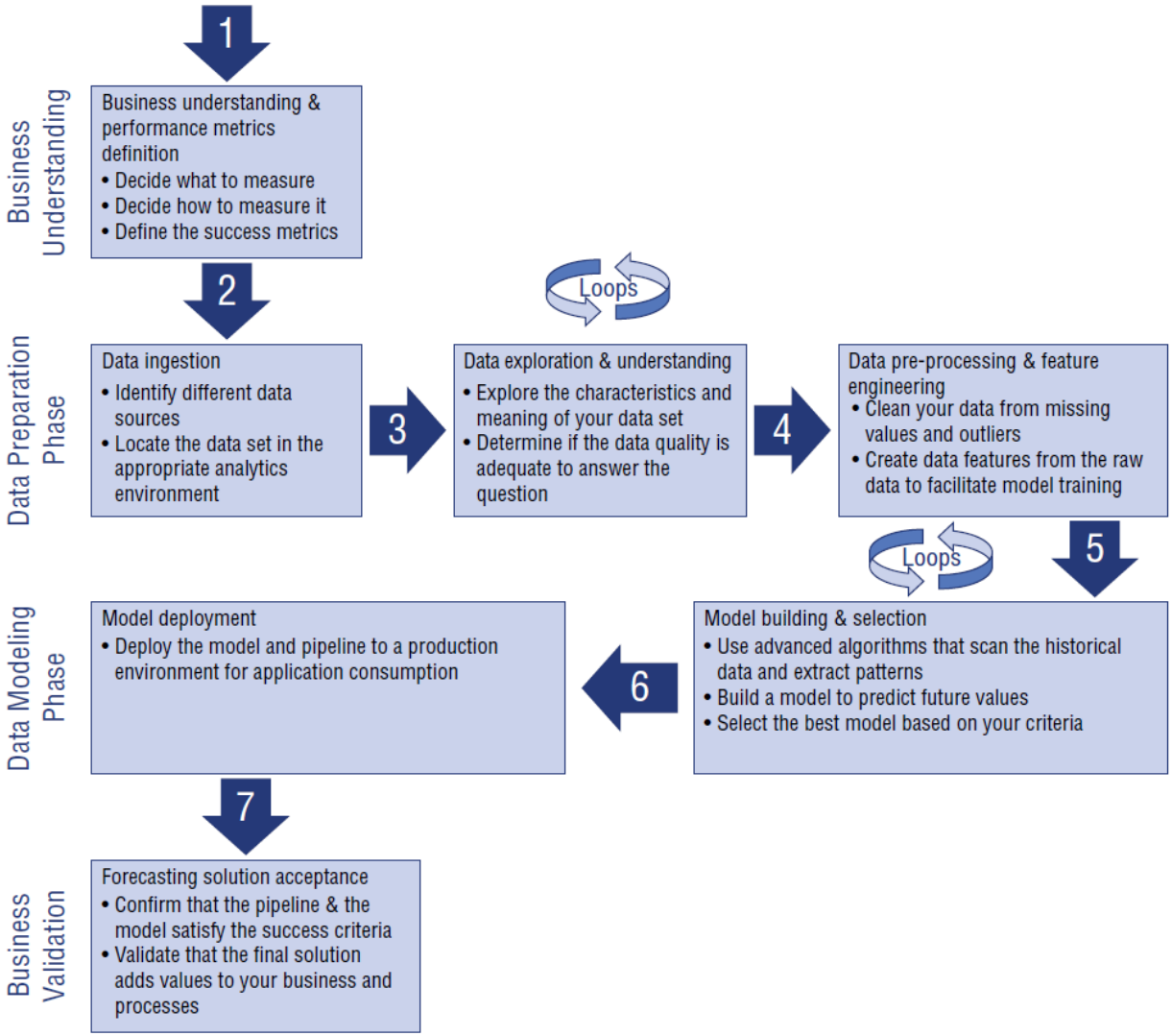
Some Practical Aspects

Cuestiones prácticas y estrategias para modelos de ML:

- Uso de datos y preparación de datasets
- Bias-Variance trade-off
- Regularization
- Elegir un buen Optimization algorithm
- Hyperparameter tuning

End-to-end solutions

Si quieren ver el picture completo de la implementación de ML dentro de un negocio:



Resources

Algunas recomendaciones si desean seguir aprendiendo ML;

- Self-paced:
  - Machine Learning Yearning (Andrew Ng 2018): excelente libro práctico
  - Dive into Deep Learning (2020): gran manual de redes neuronales
  - Kaggle: competencias y cursos
- Formal:
  - Cursos de Coursera (Deep Learning Specialization from Andrew Ng, etc.)
  - Cursos de Digital House, Coderhouse, etc.