

---

Diccionario Inglés-Español  
de  
Términos Técnicos  
en  
Inteligencia Artificial (IA)

---

Spanish-English Dictionary  
of  
Technical Terms  
in  
Artificial Intelligence (AI)

Omar U. Florez, PhD  
San Francisco, California  
USA



(c) 2018 Capital One Services, LLC This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/legalcode>.

# Contents

# Prefacio

Una gran parte del conocimiento relacionado a la **Inteligencia Artificial** esta disponible sólo en **Inglés**. Esto **previene que la información se compartan entre los hablantes de Espanol e Inglés**, y **llegue a todos con la misma igualdad de oportunidades**.

Con el fin de apoyar al entendimiento de la Inteligencia Artificial en la comunidad de Latinos y Latinas, se ha creado el **Primer Diccionario Inglés-Español de Términos Técnicos en Inteligencia Artificial (IA)**. Usuarios que pueden beneficiarse de este diccionario son estudiantes, periodistas, y divulgadores científicos que necesiten un acceso directo a conceptos de IA y que puedan compartirse fácilmente en un lenguaje simple.

**Por ese motivo, la primera definición de cada concepto corresponde a una descripción en alto nivel.** Las siguientes explicaciones proveen detalles técnicos y matemáticos para lectores interesados en profundizar su comprensión. El formato de cada entrada es el siguiente;

**Concepto** [Traducción] -*tipo* Explicación  
simple del término técnico. Segundo nivel  
de detalle de este concepto.

Espero os sea útil.

**Omar U. Florez, PhD**  
Diciembre del 2018

# Preface

A major part of knowledge related to **Artificial Intelligence** is only available in English. **This prevents information and recent discoveries from being equally shared among Spanish and non-Spanish speakers.**

With the goal of supporting the understanding of Artificial Intelligence in the LatinX community, it has been created the **First English-Spanish Dictionary of Technical Terms in Artificial Intelligence (AI)**. Users that may benefit from this dictionary are students, media professionals, and scientific disseminators who need quick and direct access to AI concepts that can be easily shared.

**The first definition of each concept corresponds to a high-level description.** Following explanations provide technical details and mathematical formulations behind the concepts for readers interested in obtaining an advanced understanding. The format of each entry follows;

**Concept** [Translation] -*type* Simple  
explanation of the technical term. Second  
level of detail for this concept.

Hope you find it useful.

**Omar U. Florez, PhD**  
December of 2018

# Diccionario

## A

**Accuracy** [Exactitud] -*n.* Es una métrica del rendimiento de un clasificador y se representa como el número de clasificaciones correctas (True Positives y True Negatives) dividido por el número total de elementos en el dataset de training, testing, o validación (True Positives, True Negatives, False Positives, False Negatives).

**Activation Function** [Función de Activación] -*n.* Es una transformación matemática que sucede a la salida de una neurona artificial. Esta operación sucede después de haber calculado la combinación lineal de los datos de entrada con los respectivos pesos de la neurona.

La función de activación esta inspirada en la influencia del campo eléctrico extracelular sobre un conjunto de neuronas biológicas.

Ejemplos populares de funciones de activación incluyen: Sigmoid, Tangente

hiperbólico (Tanh), Rectified linear unit (ReLU), y Exponential Linear Unit (ELU)

**Algorithm** [Algoritmo] -*n.* Es una secuencia de operaciones que resuelven una conjunto de problemas y que involucran el procesamiento de datos, el cálculo de operaciones matemáticas, o la predicción de resultados basados en evidencia. Un algoritmo puede ser definido en un lenguaje de programación y expresado como una función con datos de entrada y de salida.

### Artificial Intelligence (AI)

[Inteligencia Artificial] -*n.* Es la capacidad de las maquinas de demostrar inteligencia o imitar capacidades cognitivas propias de seres inteligentes. Estas capacidades incluyen razonar, representar conocimiento, aprender en base a evidencias, planear en base a objetivos, tener curiosidad, y entender el lenguaje natural.

La implementación de inteligencia artificial se basa en métodos de optimización

matemática, inferencia estadística, y abstracción computacional.

Actualmente la Inteligencia Artificial (IA) recibe influencia de diversos campos incluyendo lingüística, biología, psicología, economía, y muchos otros.

**Attributes** [Características] -  
*n.* Representan las propiedades de un objeto. Cuando son observables, se pueden medir de forma automática con sensores (e.g., los píxeles de un objeto, el espectro de frecuencias de un audio, o las palabras de un tweet) o de forma manual (e.g., el nombre de una persona o el tipo de música de una canción). Cuando no son observables, se les denomina latentes y se les representa como un vector numérico en cierto espacio matemático llamado *embedding* (e.g. la salida de una capa en una red neuronal)

## B

**Back Propagation** [Retro-propagación] -*n.* Es una técnica de optimización matemática que se utiliza para entrenar una red neuronal. Este algoritmo empieza por calcular el error en la capa de salida de la red y luego propaga esta información hacia las capas anteriores utilizando la regla de la cadena sobre cada capa (ref. Chain Rule).

La técnica Gradient Descent (ref. Gradient Descent) utiliza *back propagation* para actualizar los pesos de una red neuronal con un vector llamado gradiente, el cual comúnmente corresponde a la primera derivada de la función de costo con respecto a sus pesos. El valor y dirección de la gradiente guía la optimización de los pesos hacia el valor máximo de la función de costo. Debido a que *back propagation* tiene como objetivo reducir la noción de error (*loss function*) en el sistema, los pesos se actualizan con el valor negativo de la gradiente.

**Bias** [Sesgo/Prejuicio] -*n.* Un algoritmo de aprendizaje supervisado muestra un alto *bias* cuando predice de forma frecuente resultados incorrectos para cierta clase de observaciones.

Existe un balance entre el *bias* y el *variance* (ref. Variance) al momento de diseñar la solución a un problema de aprendizaje supervisado: un algoritmo con poca capacidad (ref. Capacity) puede ser poco flexible al aprender un número pequeño de interacciones en los datos de entrenamiento, mostrando así un alto *bias* y un menor *variance*.

Por el contrario, un algoritmo con alta capacidad puede llegar a ser demasiado flexible al aprender interacciones complejas en distintas regiones del espacio de los datos e incluso aprender el ruido pre-

sente en los datos de entrenamiento. El resultado es que el modelo se comporta de forma distinta con diferentes *datasets*, mostrando un bajo *bias* pero un alto *variance*.

## C

**Capacity** [Capacidad] *-n.* Es un valor numérico que mide la complejidad de un modelo para reconocer la presencia de distintas clases en los datos. Dicha complejidad obedece a las interacciones entre las variables de entrada, latentes, y de salida que componen un modelo. Mientras más grande sea la capacidad, el modelo puede aproximar funciones más complejas y no-lineales. En redes neuronales, la capacidad es comúnmente proporcional al número de pesos que exhibe su arquitectura y los cuales representan los parámetros entrenables del modelo.

Una medida más teórica de la capacidad es el VC *Dimension* (ref. VC *Dimension*), la cual mide el número máximo de observaciones que un clasificador puede asignarles etiquetas de forma correcta.

**Convergence** [Convergencia] *-n.* Es un estado de estabilidad dentro del proceso de entrenamiento de un modelo de aprendizaje automático. Podemos observar convergencia cuando la diferencia

entre los valores sucesivos de la función de costo es casi constante, el ciclo de entrenamiento alcanza un máximo número de pasos, o las funciones de costo de los datos de entrenamiento y validación dejan de disminuir de forma conjunta. Muchas veces seguir entrando a partir del punto de convergencia conduce al modelo a aprender el ruido presente en los datos de entrenamiento y generar *overfitting*.

**Convolution** [Convolución] *-n.* Es una operación matemática entre dos señales  $f$  y  $g$  que expresa la transformación de  $f$  cuando se le desplaza por encima  $g$ . Por ejemplo, imagine los píxeles de un imagen  $f$  que contiene el rostro de una persona y un conjunto de pesos  $g$  de una red neuronal que multiplica partes consecutivas de la imagen con el objetivo de calcular similitudes en toda la señal. El resultado es la convolución ( $f * g$ ) y consiste en una secuencia de valores que expresan que partes del rostro reaccionan más a la señal  $g$ , resaltando en este proceso esquinas, bordes, texturas, y otras características (ref. Feature).

Note que la convolución anterior recibe datos de entrada en dos dimensiones en la función  $f$ , mas es muy frecuente usar también señales unidimensionales, como por ejemplo en el análisis de series de tiempo.

Si los datos son discretos, es posible acelerar el cálculo de la convolución mediante el cómputo del *Fast Fourier Transform* (FFT) para cada señal  $f$  y  $g$  independientemente, luego se multiplica ambas transformaciones elemento por elemento, y finalmente se calcula la inversa del FFT de este producto.

### Convolutional Neural Network (CNN)

[Red Neuronal Convolutional] -n. Un tipo de red neuronal profunda que utiliza operaciones de convolución (ref. Convolution) a través de una jerarquía de capas sobre los datos de entrada para imitar el efecto de los campos receptivos en la visión humana. Dicha transformación aprende una representación de los datos en cada capa generando características más complejas mientras más profunda sea la arquitectura de la red neuronal.

Este modelo ha sido inspirado por el trabajo de Hubel y Wiesel en procesamiento de información en el córtex visual, en donde se manifiesta los beneficios de explotar las correlaciones espaciales en imágenes. Esto añade robustez a las transformaciones tales como cambios de orientación y escala.

## D

**Dataset** -n. Es una colección de datos u observaciones

relacionadas a un problema determinado. Ejemplos conocidos incluyen una colección de imágenes que contienen objetos frecuentes (*ImageNet dataset*), las noticias de una agencia internacional (*Reuters-21578 dataset*), una lista de canciones (*1 Million Songs dataset*), o las preferencias de varios usuarios sobre determinadas películas (*Netflix dataset*). El *dataset* más conocido en algoritmos de aprendizaje profundo (ref. Deep Learning) es *ImageNet* y consiste de más de un millón de observaciones, categorizadas en 1000 tipos de objetos.

Cuando el dataset se almacena en una matriz, las columnas representan los diferentes atributos de un problema (ref. Feature) y las filas representan los vectores de atributos u observaciones de distintas instancias del problema (ref. Feature Vector).

Para datos textuales, al dataset también se le denomina *corpus*.

**Deep Neural Networks** [Redes Neuronales Profundas] -n. Son redes neuronales que contiene más de una capa escondida, lo cual incrementa la capacidad del modelo para aproximar funciones más complejas (Ref. Capacity).

El éxito actual de las redes neuronales profundas radica en aplicar optimización basada en gradientes (Ref. Gradients) a modelos pro-



fundos que tienen una gran capacidad para identificar distintas interacciones en los datos de entrada (patrones). De esta manera, la arquitectura del modelo es proporcional a su desempeño, si se le alimenta con una gran cantidad de información y muestra una capacidad adecuada.

**Deep Learning** [Aprendizaje Profundo] -n. Es una técnica de aprendizaje automático basado en redes neuronales profundas (ref. Deep Neural Networks) que tiene la propiedad de aprender características o *features* durante su proceso de entrenamiento. Esto la diferencia de otras técnicas de aprendizaje automático que requieren una selección de características manual o automatizada por propiedades estadísticas.

La clasificación realizada por un algoritmo de *deep learning* muestra un mejor rendimiento debido a que encuentra de forma iterativa el espacio matemático en donde la función de pérdida (ref. Loss Function) se minimiza.

La capacidad del algoritmo para recordar patrones se relaciona directamente con su arquitectura. Por ejemplo, patrones espaciales como texturas en imágenes pueden ser aprendidos por una red neuronal profunda llamada *Convolutional Neural Network* (CNN), patrones temporales que muestran dependencias

estadísticas en los datos de entrada como la voz humana o las palabras en un *tweet* suelen ser aprendidas con una red neuronal llamada *Long Short-Term Memory* (LSTM), y reconstrucciones de los datos de entrada con el fin de comprimir la información o segmentarla se representan con *Variational Auto Encoders* (VAE).

## E

**Entropy** [Entropía] -n. Es una medida de información que indica el grado de desorden en un conjunto o la cantidad de ruido en una señal. En teoría de la información, la entropía se mide en *bits* y representa el porcentaje de información generado por un proceso estocástico (ref. Stochastic).

La entropía se define Matemáticamente como el valor esperado del logaritmo negativo de los elementos de una distribución discreta.

$$H = - \sum_i P_i \ln P_i$$

Interesantemente, la entropía utiliza el logaritmo negativo para cuantificar la mayor cantidad de información presente en eventos menos probables y la multiplica por la probabilidad de su ocurrencia calculando así su valor esperado (ref. Expected Value).

El concepto de entropía en la información fue introducido por Claude Shannon en su trabajo titulado "A Mathematical Theory of Communication" en 1948.

**Expected Value** [Valor Esperado]

-n. Es el valor predicho de una variable y corresponde a la suma del valor de cada observación multiplicada por la probabilidad de su ocurrencia.

Matemáticamente, si  $x$  representa el valor de la variable  $X$  y  $p(x)$  es su probabilidad de ocurrencia, el valor esperado de  $X$  se define como,

$$E[X] = \sum_{i=1}^k x_i p(x_i)$$

La regla de los números largos (*law of large numbers*) establece que el promedio de los valores de una variable casi seguramente converge a su valor esperado si el número de repeticiones es casi infinito.

Cuando un algoritmo de aprendizaje automático produce predicciones cuyo valor esperado es igual al valor real de la variable, se dice que no muestra *bias* y es por lo tanto un *unbiased estimator*.

## F

**Feature** [Característica] -n. Es el valor de una propiedad

particular asociada a un fenómeno, el cual puede ser observable o escondido (latente). Ejemplos de características observables incluyen los píxeles de una imagen, las frecuencias de un audio, las palabras de un texto, o incluso las conexiones en una red social. Una característica latente es el valor de salida (activación) de una neurona dentro de una capa.

Al proceso de escoger o crear características que sean estadísticamente informativas, no muestren redundancia, y discriminen correctamente se le denomina *feature engineering* (ingeniería de características) y es muy importante para obtener modelos de *regresión*, *clasificación*, y *clustering* que funcionen de forma más exacta y sean robustos al ruido.

La salida de un modelo también se le puede usar como una característica y se le denomina *meta-data*.

**Feature Vector** [Vector de Características]

-n. Es el conjunto de distintas características (ref. Feature) asociadas al mismo fenómeno.

El número de dimensiones de este vector representa la cantidad de características que describen el estado de cierto fenómeno. La dimensionalidad o número de características de este vector no debería ser muy largo, debido al llamado *curse of dimensionality* (maldición de la alta dimensionalidad), la

cual representa la dificultad en separar vectores de alta dimensionalidad en distintas clases. Por otro lado, el número de vectores de características indica el tamaño del dataset con el que podemos entrenar y validar un modelo.

Los algoritmos de inteligencia artificial usualmente requieren de un vector de características para facilitar el proceso numérico de encontrar patrones en los datos. A menudo un vector de características puede contener valores faltantes (ref. Missing Values), lo cual suele indicar que no todas las características son observables al mismo tiempo.

## G

### Generalization [Generalización]

-n Es la propiedad de los seres humanos y animales de utilizar aprendizaje pasado para responder a situaciones presentes, si el contexto y los estímulos son similares.

El cerebro realiza constantemente generalización cuando extrae las propiedades comunes de múltiples observaciones y las abstrae en un concepto más general. Así, los píxeles de una imagen que corresponde a un *gato* pueden generalizarse bajo el concepto de un *animal*, a pesar de que otras instancias del mismo concepto luzcan muy

diferentes, por ejemplo un *perro*.

En el aprendizaje automático, se utiliza el término inferencia (ref. Inference) para referirse a esta propiedad.

**Gradient** [Gradiente] -n. La gradiente de una función es un vector que apunta en la dirección donde su función se maximiza. A la magnitud de la gradiente se le conoce como pendiente.

Matemáticamente, a la gradiente de la función diferenciable  $f$  en el punto  $x_0$  se le denota como  $\nabla f(x_0)$  y representa la tangente de la función en ese punto, es decir es la mejor aproximación lineal a  $f$  en  $x_0$ . Tal aproximación se le calcula como

$$f(x) = f(x_0) + \nabla f(x_0) \cdot (x - x_0)$$

para un  $x$  muy cercano a  $x_0$

**Gradient Ascent** -n. Es una técnica de optimización matemática que actualiza sucesivamente las variable entrenables de un modelo en la dirección de la gradiente de una función objetivo con el objetivo de encontrar el máximo de tal función.

**Gradient Descent** -n. Es una técnica de optimización matemática que itera en la dirección opuesta a la gradiente para encontrar mínimos locales de una función de costo. La actualización de los pesos de una red neuronal  $\theta$  usando *gradient descent* tiene la siguiente forma:

$$\theta = \theta - \alpha \nabla J(\theta)$$

donde la función  $L(\theta)$  es una medida de error de predicción asociada a la red neuronal (ref. Loss function) y  $\alpha$  es la velocidad de aprendizaje (ref. Learning rate).

## H

**Hidden Layer** [Capa Escondida] -*n.* Una capa de neuronas dentro de la arquitectura de una red neuronal que no está expuesta a los datos de entrada ni a los datos de salida.

**Hill Climbing** -*n.* Es un algoritmo de optimización que estima los valores de los parámetros entrenables de un modelo, por ejemplo los pesos de una red neuronal. *Hill Climbing* añade sucesivamente una pequeña cantidad de ruido con el fin de proponer un modelo que de un mejor rendimiento y optimice una función objetivo. Si tal cambio produce una mejor solución, otro cambio incremental se produce encima de la nueva solución hasta que no se encuentren mejoras sucesivas.

**Hyper-Parameter** [Híper-Parámetro] -*n.* Son los parámetros externos a un modelo de aprendizaje automático cuyos valores no se calculan mediante un proceso de optimización matemática sino a través de una búsqueda manual o heurística. Ejemplos de híper-parámetros incluyen

el número de capas escondidas, la velocidad de aprendizaje, y el rango de los valores de inicialización de los pesos de una red neuronal.

Usualmente se ajusta los híper-parámetros calculando los valores que resultan en un rendimiento óptimo del modelo en un subconjunto de los datos llamado el *dataset de validación*.

## I

**Inference** [Inferencia] -*n.* Es el proceso de obtener hipótesis en base a evidencia o conclusiones lógicas.

En redes neuronales, este paso corresponde a la predicción realizada por un modelo entrenado para saber a que clase le pertenece una observación en los datasets de *testing* o validación. Los tipos de inferencia incluyen: deducción, inducción, y abducción.

## J

## K

**Kernel** -*n.* Es una función que pondera los datos de entrada de una señal durante la operación de convolución (ref. Convolution).

Estadísticamente, es un función de densidad probabilística que normaliza los valores de una variable.

## L

**Label** [Etiqueta] *-n.* Es el valor real asignado a una observación en un dataset (ref. dataset). A menudo cada observación posee una etiqueta la cual ha sido otorgada por una persona después de observar sus características (ref. Feature). Por ejemplo, a los píxeles de una imágenes se les puede otorgar una etiqueta que indica el objeto que representan. Para reducir la subjetividad en su definición, se suele pedir a varias personas que definan una etiqueta para la misma observación y así obtener más robustez en su definición. La clasificación de observaciones que contienen más de una etiqueta al mismo tiempo, se le denomina *multi-label classification*.

**Learning** [Aprendizaje] *-n.* Es el proceso de actualizar los parámetros entrenables de un modelo matemático o estadístico con el fin de optimizar una función objetivo (Ref. Loss Function) y de esa manera resolver una tarea determinada (e.g., clasificación, regresión, *clustering*). En el caso de las redes neuronales, los parámetros entrenables son los pesos en

tre las neuronas y el aprendizaje se da a través de un proceso iterativo que actualiza los pesos con el objetivo de minimizar una métrica de error o *loss function*.

**Learning Rate** [Velocidad de Aprendizaje] *-n.* Es un hiper-parámetro (ref. Hyper-Parameter) que representa el ratio con el cual se modifican los pesos de una red neuronal. Mientras un *learning rate* actualiza rápidamente los pesos, puede no converger en un mínimo local adecuado. Por otro lado, un valor muy bajo puede hacer que el entrenamiento converja lentamente, pero también dejar de explorar otras regiones del espacio de solución que podrían minimizar la función de costo (ref. Loss Function).

Se le representa con la letra  $\alpha$  y usualmente se define como una función del tiempo con la idea de disminuir su valor mientras el entrenamiento se desarrolla.

**Linear Kernel** [Kernel Lineal] *-n.*

Un *kernel* lineal es la simple suma de la multiplicación de cada una de las entradas de dos vectores de igual tamaño. El término matemático para esto se le llama producto punto y también se le define como el coseno del ángulo de dos vectores multiplicado por el producto de sus longitudes.

**Long Short-Term Memory (LSTM)**

*-n.* Es un tipo de red neuronal recurrente (ref. Re-

current Neural Network) que resuelve explícitamente el problema del desvanecimiento de gradientes (*vanishing gradients*) mediante el uso de compuertas entrenables que controlan el flujo de gradientes dentro de una unidad de procesamiento. Esto se realiza mediante un conjunto de operaciones sobre la memoria interna de cada unidad. Por ejemplo, LSTM puede aprender a escribir, leer, y sobrescribir patrones en la memoria utilizando compuertas llamadas *input* (entrada), *output* (salida), and *forget* (olvido), respectivamente.

A diferencia de otros modelos como RNN y HMM, que también representan dependencias temporales, LSTM no suele ser sensible a la presencia de intervalos entre patrones dentro de largas señales de entrada, de ahí el término *long-term* (largo-plazo).

**Loss Function** [Función de Costo] -n. Es un valor numérico que representa el costo o error asociado a una predicción. En redes neuronales, una observación genera una distribución de clases en la capa de salida, este valor representa la diferencia entre tal distribución y la clase asignada a esta observación. Un método común para medir esta discrepancia es el denominado error cuadrado:

$$J(\theta) = (y - f(x, \theta))^2$$

## M

**Machine Learning** [Aprendizaje Automático] -n. Es la predicción del futuro con datos, evidencia, y patrones del pasado usando una computadora.

**Mapping** -n. Transformación matemática que consiste en llevar los datos a una espacio en donde ciertas propiedades se cumplen. Por ejemplo, que cada dimensión sea ortogonal o que la separación entre clases sea más larga.

**Memory** [Memoria] -n. Conjunto de pesos de una red neuronal que se activan de forma similar en presencia de la misma observación.

**Multi-Layer Perceptron (MLP)** [Red Neuronal Multi-capa] -n. Es un tipo de red neuronal que esta organizada en una capa de entrada, una o mas capas escondidas, y una capa de salida. Las capas de esta red neuronal se conectan a través de sus sinapsis, cada una asociada a un valor numérico llamado peso y que representa su intensidad.

Un MLP se utiliza principalmente como un clasificador con el fin de aprender un espacio matemático donde la representación de los datos de entrada es fácilmente separable en clases. Debido a su capacidad de aproximar funciones muy complejas, se les denomina aproximadores universal de funciones (*universal function*

*approximators*)

Cada capa de un MLP es un conjunto de neuronas que propagan la señal hacia la siguiente capa, creando nuevas representaciones, y finalmente proyectándolas a la capa de salida, la cual tiene un número de neuronas igual al número de clases a aprender. El valor óptimo de los pesos de una de un MLP se realiza ajustando los pesos mediante la técnica llamada *back propagation* (ref. Back Propagation).

## N

**Neural Networks** [Redes Neuronales] -*n.* Un modelo matemático cuya arquitectura contiene varias capas de neuronas las cuales construyen progresivamente representaciones más abstractas de información directamente desde los datos de entrada.

## O

**Objective function** [Función Objetivo] -*n.* Ref. Loss Function.

**Occam's Razor** -*n.* Es una heurística utilizada en ciencia que aconseja la elección de modelos más simples sobre modelos complejos o con mayor capacidad (ref. Capacity).

La lógica de esta heurística es que si la optimización de modelos de aprendizaje suele ser no-convexa, entonces siempre existirán modelos más complejos, y menos interpretables, que provean resultados similares. Ante la existencia de alternativas más complejas, se elige los modelos más simples debido a que sus desempeños son más fáciles de evaluar o consumen una menor cantidad de recursos.

**Optimization** [Optimización] -*n.*

Es la elección del mejor conjunto de parámetros entrenables de un modelo con el fin de maximizar su función objetivo (ref. Objective Function).

## P

**Parameters** [Parámetros] -*n.* Valores que influyen en el comportamiento y desempeño de un modelo entrenable. Por ejemplo, los parámetros de una red neuronal son sus pesos.

**Perceptron** [Perceptrón] -*n.* Es un clasificador que aprende a categorizar entre dos clases (0 y 1) multiplicando un peso por cada dimensión de los datos de entrada y le suma a esta operación una constante llamada *bias* que mueve la decisión lejos del origen. Si los datos de entrada son  $x$ , los pesos del *perceptron* son  $w$  y el término *bias* es  $b$ , el *percep-*

tron retornará 1 si

$$w \cdot x + b > 0$$

y 0 en caso contrario.

**Precision** [Precisión] -n.

Aunque se suele utilizar comúnmente como sinónimo de exactitud, su definición es diferente en el contexto del método científico. La precisión es el grado de similaridad entre las predicciones correctas otorgadas por un modelo de aprendizaje automático. Si estas predicciones muestran variabilidad entre ellas, el modelo no será preciso.

Se le suele definir como el número de predicciones correctas (*True Positive*) dividido por el número total de predicciones (*True Positive* y *False Positive*). Un modelo puede ser preciso, mas no exacto y también ser poco preciso y exacto simultáneamente.

**Policy** -n. Es una función que define el comportamiento de un agente que interactúa a través de acciones con un ambiente determinado. El *policy*  $\pi(a|s)$  describe la probabilidad de tomar la acción  $a$  cuando el agente se encuentra en el estado  $s$ .

Q

R

**Recurrence** [Recurrencia] -n. Ref. Recurrence

**Recurrent Neural Network (RNN)**

[Red Neuronal Recurrente] -n. Es un tipo de red neuronal profunda (ref. Deep Neural Network) que presenta sinapsis y pesos entre cada unidad interna de procesamiento (memoria), cada cual alimentada por un dato de entrada dentro de una secuencia. Esta propiedad las hace adecuadas para modelar datos temporales como la voz humana, música, documentos de texto, y videos.

Las RNNs reciben el nombre de *recurrentes* por su capacidad de definir su memoria en términos de estados de memoria anteriores. Es decir, la salida de cada unidad  $c_{t+1}$  es una función de la entrada actual  $x_t$  y el valor actual de su memoria  $c_t$ ,

$$c_{t+1} = f(c_t, x_t)$$

La forma mas común de entrenar una red neuronal recurrente es usando gradientes con la técnica llamada *Back Propagation Through Time* (BPTT), la cual es similar a la técnica llamada *Back Propagation* (ref. Back Propagation) usada para entrenar modelos de aprendizaje profundo.

La principal diferencia es que BPTT desenvuelve la estructura temporal de la RNN en una secuencia donde todas las unidades comparten los mismos parámetros y memoria. Luego, se calcula la señal de error (*error signal*) después de proyectar la salida de la ultima unidad



—a manera de predicción— y compararla con la correspondiente etiqueta de los datos de entrada. Este es el inicio del proceso de retro propagación y va en el sentido opuesto a la secuencia, actualizando los pesos de toda la red neuronal en ese orden.

Entrenar este tipo de modelos puede presentar complicaciones cuando se memorizan patrones en señales de larga duración. La multiplicación de gradientes en una secuencia larga hace que el producto final converja rápidamente a 0 si las gradientes son menores a 1, o se incrementa rápidamente si las gradientes son mayores a 1. Ambos problemas reciben el nombre de desvanecimiento (*vanishing*) o explosión (*exploding*), respectivamente; y su estudio ha conducido a diseñar redes recurrentes que controlan el flujo de gradientes en base a compuertas entrenables llamadas redes LSTM (ref. Long Short Term Memory).

## Reinforcement Learning

[Aprendizaje por Refuerzo] -n. Conjunto de algoritmos que entrenan a un agente a interactuar con un ambiente a través de una secuencia de estados, acciones, y premios (*rewards*). El agente es entrenado con el objetivo de maximizar el valor acumulado de futuros *rewards* durante la secuencia de acciones que suceden durante un episodio.

## S

**Softmax** -n. Es una función de activación (ref. Activation Function) que a menudo se coloca en la capa de salida de una red neuronal. Dada una capa de una red neuronal, la función *Softmax* normaliza cada valor entre 0.0 y 1.0 con la condición que la suma de estos valores sea 1.0. El resultado es un vector que representa la probabilidad de que un dato de entrada pertenezca a cada clase disponible con cierto nivel de probabilidad. Matemáticamente, la ecuación de esta función se define como:

$$S(f_{y_i}) = \frac{e^{f_{y_i}}}{\sum_j e^{f_{y_j}}}$$

donde  $f_{y_i}$  representa el vector que contiene los datos de la capa de salida de una red neuronal.

**Stochastic** [Estocástico] -n. Es la cualidad de un evento de ser determinado aleatoriamente.

Cuando un proceso es estocástico, se dice que es un proceso aleatorio (*random process*) en el cual los valores de sus variables están especificados por una función de distribución probabilística. Por ejemplo, el proceso de tirar un dado está definido por una función que asigna una probabilidad a cada resultado del experimento. Después de un gran número de

intentos, dicha distribución mostrará la misma probabilidad para cada valor del dado ( $\frac{1}{6}$ ).

**Supervised Learning** [Aprendizaje Supervisado] -*n.* Es un problema del aprendizaje automático que consiste en aprender una función que transforme los datos de entrada (nuestra observación del fenómeno) hacia una etiqueta, la cual representa la clase a la que pertenece dicha observación y que ha sido anotada manualmente.

Los parámetros entrenables  $\theta$  de dicha función  $f(x, \theta) = y$  son ajustados en un proceso llamado entrenamiento (ref. Training), el cual calcula la correspondencia matemática entre los elementos de entrada  $x$  y las etiquetas  $y$ . Estadísticamente, en algoritmos como las Redes Neuronales, esta relación toma la forma de la probabilidad condicional de las etiquetas dado los valores de entrada y parámetros entrenados,  $P(y|x, \theta^*)$  también llamado *likelihood*; mientras que en el caso de las Redes Bayesianas, dicha relación se aproxima mediante la probabilidad conjunta entre estos elementos  $P(x, y, \theta^*)$ .

La predicción de nuevos elementos de entrada  $x'$  es simplemente la ejecución de la función  $f(x', \theta^*)$ , ya con sus parámetros entrenados  $\theta^*$ . El resultado es una distribución sobre el conjunto de etiquetas, cuyo valor máximo representa la

predicción de que clase le corresponde a  $x'$ . Este proceso de inferencia en base a evidencias tiene similitud al proceso de generalización que sucede en el cerebro humano (ref. Generalization)

## T

**Training** [Entrenamiento] -*n.* Actualización iterativa de los parámetros entrenables de un modelo de aprendizaje automático en la dirección en la cual la función de error se minimiza.

**Trainable Parameters** [Parámetros Entrenables] -*n.* Conjunto de parámetros que pueden cambiar durante un proceso de entrenamiento con el fin de optimizar una función de costo (ref. Loss Function)

## U

### Unsupervised Learning

[Aprendizaje No Supervisado] -*n.* Es un tipo de aprendizaje que no requiere el etiquetado de observaciones y que funciona en base a identificar la presencia de relaciones entre grupos de observaciones. Tales relaciones pueden incluir las medidas de similaridad, densidad, asociación, o jerarquía entre las observaciones.

## V

**Variance** [Varianza] -*n*. Un algoritmo de aprendizaje supervisado tiene un alto *variance* cuando predice distintos resultados para diferentes *datasets*.

Por ejemplo, si un modelo entrenado exhibe una exactitud muy diferente para *datasets* de entrenamiento y validación, entonces se dice que muestra un alto *variance*.

## W

**Weights** [Pesos] -*n*. En una red neuronal artificial, un

peso corresponde al valor de la sinapsis entre dos neuronales y representa el nivel de activación de la neurona en presencia de sus datos de entrada.

Los pesos son usualmente los valores entrenables de una red neuronal.

## X

## Y

## Z