

Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Arquitectura de Software



Pontificia Universidad
JAVERIANA
Bogotá

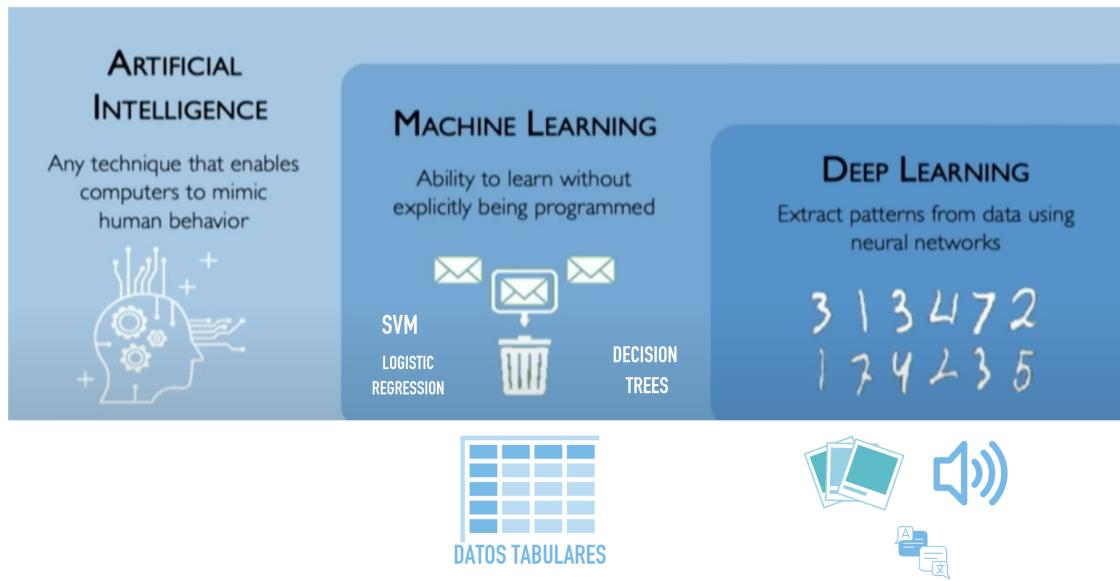
**Presentación Deep learning
y MLOPS**

Presentado por:
Juan Sebastián Vargas Torres-Ing. Sistemas

Docente:
Andres Armando Sanchez Martin

Mayo 09 del 2023
Bogotá, Colombia

1. Marco Teórico:



IA (Inteligencia Artificial) es un campo de la informática que busca desarrollar sistemas capaces de realizar tareas que requieren inteligencia humana, como la comprensión del lenguaje natural, el reconocimiento de patrones, la toma de decisiones y la resolución de problemas.

Machine Learning (Aprendizaje Automático) es una subárea de la IA que se centra en el desarrollo de algoritmos y modelos que permiten a las máquinas aprender a partir de datos sin necesidad de ser programadas explícitamente. En otras palabras, el objetivo del Machine Learning es entrenar a un modelo a través de datos y algoritmos para que pueda hacer predicciones o tomar decisiones en base a nuevos datos.

Deep Learning (Aprendizaje Profundo) es una técnica de Machine Learning que utiliza redes neuronales artificiales para aprender a partir de datos. Estas redes neuronales están formadas por múltiples capas de nodos interconectados que procesan los datos de forma jerárquica y progresivamente compleja. El Deep Learning ha demostrado ser muy efectivo en tareas como el reconocimiento de imágenes, el procesamiento de lenguaje natural y la toma de decisiones en juegos complejos.

¿Por qué hoy en día podemos trabajar con estas tecnologías?

BIG DATA

- Grandes volúmenes de datos
- Variedad de tipos de datos (Imágenes , audio, texto, etc)
- Grandes bases de datos.
- Altas capacidades de almacenamiento.

IMAGENET

HARDWARE

- GPU'S

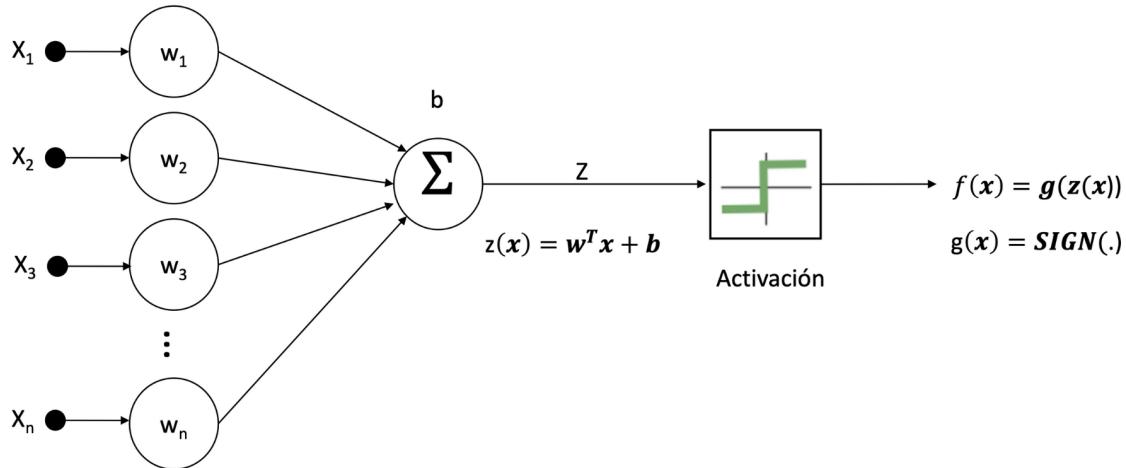


SOFTWARE

- Librerías optimizadas el entrenamiento de modelos



¿Qué es el perceptrón?



Un perceptrón es un tipo de algoritmo de aprendizaje automático utilizado en el campo del Machine Learning y la inteligencia artificial. Es una red neuronal artificial que se utiliza principalmente para la clasificación de patrones.

El perceptrón fue propuesto originalmente por el científico informático Frank Rosenblatt en 1957 y es uno de los algoritmos de aprendizaje supervisado más simples que existen.

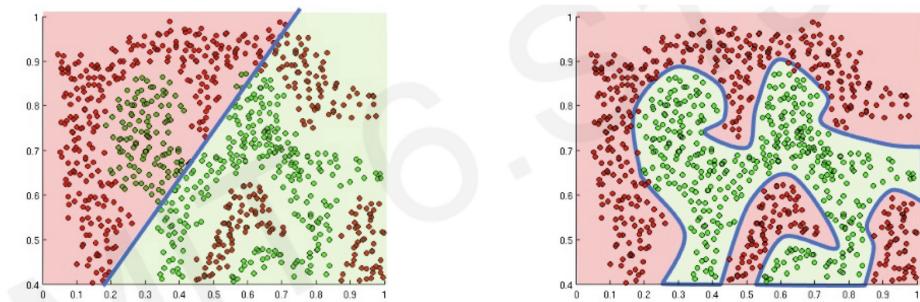
El perceptrón consiste en una o varias neuronas artificiales que reciben entradas numéricas y producen una salida binaria (1 o 0). La neurona combina las entradas con pesos numéricos y una función de activación para producir la salida.

Durante el entrenamiento del perceptrón, se ajustan los pesos de la neurona a través de un proceso iterativo, de manera que la salida del perceptrón se acerque lo más posible a la salida

deseada para un conjunto de ejemplos de entrenamiento. Este proceso se llama ajuste de los pesos, y se utiliza el algoritmo de aprendizaje del perceptrón para lograrlo.

El perceptrón es muy útil en la clasificación binaria y es utilizado en aplicaciones como la identificación de spam, la detección de fraude y la clasificación de imágenes y texto. Sin embargo, debido a su simplicidad, el perceptrón puede no ser capaz de resolver problemas más complejos, para los cuales se utilizan modelos más avanzados como las redes neuronales profundas.

Sin embargo se puede quedar corto ante casos como el siguiente:

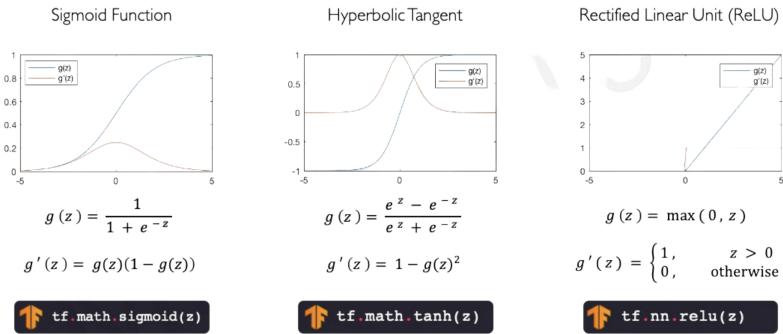


¿Qué son las funciones de activación?

En las redes neuronales artificiales, las funciones de activación son funciones matemáticas que se utilizan para introducir no linealidades en el modelo. En otras palabras, una función de activación se aplica a la salida de una capa de neuronas y produce una nueva salida que se utiliza como entrada para la siguiente capa.

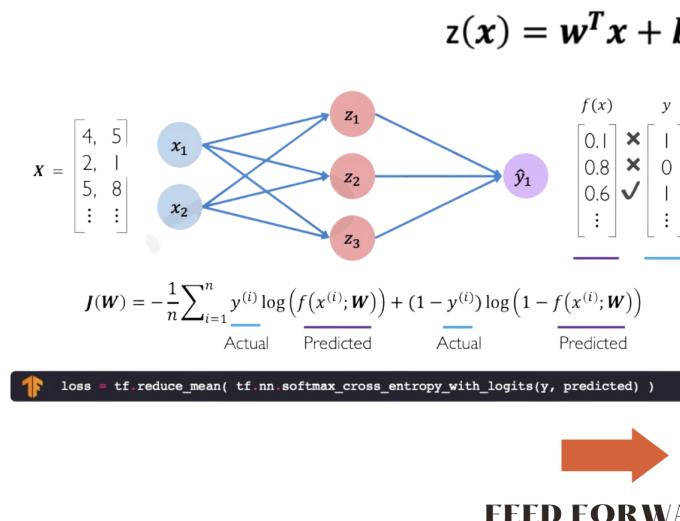
Las funciones de activación son importantes porque permiten a las redes neuronales modelar relaciones no lineales entre las entradas y las salidas. Si se utilizan únicamente funciones lineales en todas las capas, la red neuronal sería equivalente a una única capa, lo que limitaría su capacidad de modelado.

Algunos ejemplos de funciones de activación comunes son:



Existen muchas otras funciones de activación utilizadas en las redes neuronales, y la elección de la función de activación adecuada depende del problema que se esté resolviendo y de la estructura de la red neuronal.

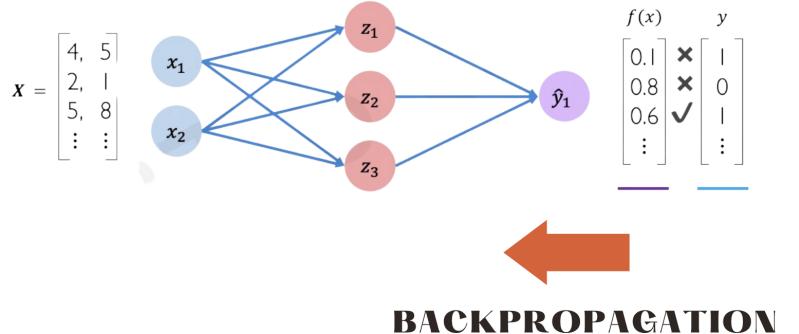
Feed forward:



Feedforward (alimentación directa) se refiere al proceso de propagar una entrada a través de una red neuronal para obtener una salida. En una red neuronal feedforward, las señales se propagan en una sola dirección, desde la capa de entrada hasta la capa de salida, sin ciclos o retroalimentación. La información fluye a través de las capas de la red neuronal, y cada capa procesa los datos y produce una salida que se utiliza como entrada para la siguiente capa.

Backpropagation:

$$z(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \mathbf{b}$$



Backpropagation (retropropagación) es un algoritmo utilizado en el entrenamiento de redes neuronales que ajusta los pesos de las conexiones entre las neuronas para minimizar la diferencia entre la salida deseada y la salida producida por la red neuronal. El algoritmo utiliza el gradiente descendente para calcular la tasa de cambio de la función de pérdida con respecto a los pesos de la red, y ajusta los pesos de forma iterativa para reducir la pérdida.

Gradient Descent:

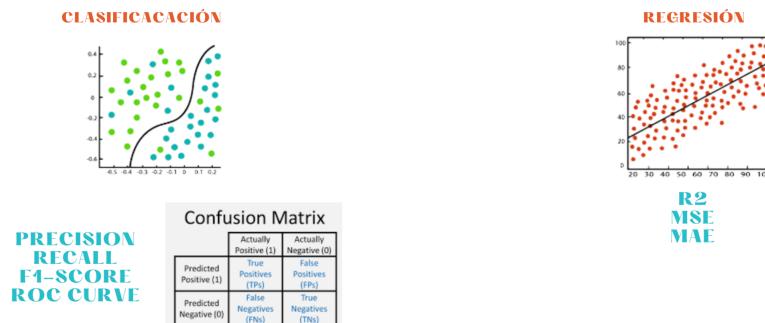
El Gradient Descent (descenso de gradiente) es un algoritmo de optimización utilizado en el aprendizaje automático y la inteligencia artificial para minimizar una función de pérdida. La idea básica del descenso de gradiente es ajustar iterativamente los parámetros de un modelo de tal manera que la función de pérdida se minimice.

En esencia, el descenso de gradiente es un algoritmo de búsqueda de la mejor solución para un problema. Comienza por elegir aleatoriamente un conjunto de valores para los parámetros del modelo y calcula la tasa de cambio de la función de pérdida con respecto a cada parámetro utilizando el cálculo del gradiente. A continuación, ajusta los valores de los parámetros en la dirección opuesta al gradiente para reducir la función de pérdida. Este proceso se repite iterativamente hasta que se alcanza un mínimo de la función de pérdida, o hasta que se alcanza un límite predefinido en el número de iteraciones.

El descenso de gradiente se utiliza ampliamente en el aprendizaje automático para entrenar modelos, como las redes neuronales, que tienen muchos parámetros y cuya función de pérdida es difícil o imposible de optimizar analíticamente. El algoritmo de backpropagation, que se utiliza para entrenar redes neuronales, se basa en el descenso de gradiente para ajustar los pesos de las conexiones entre las neuronas durante el entrenamiento.

Existen varias variantes del descenso de gradiente, como el descenso de gradiente estocástico y el descenso de gradiente por lotes, que se utilizan para mejorar el rendimiento y la eficiencia del algoritmo en diferentes situaciones.

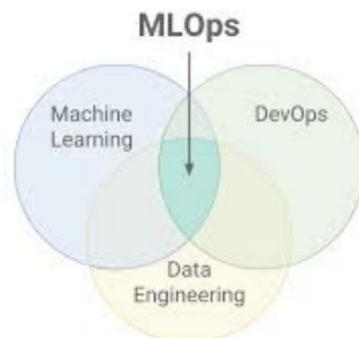
Métricas:



MLOPS:x

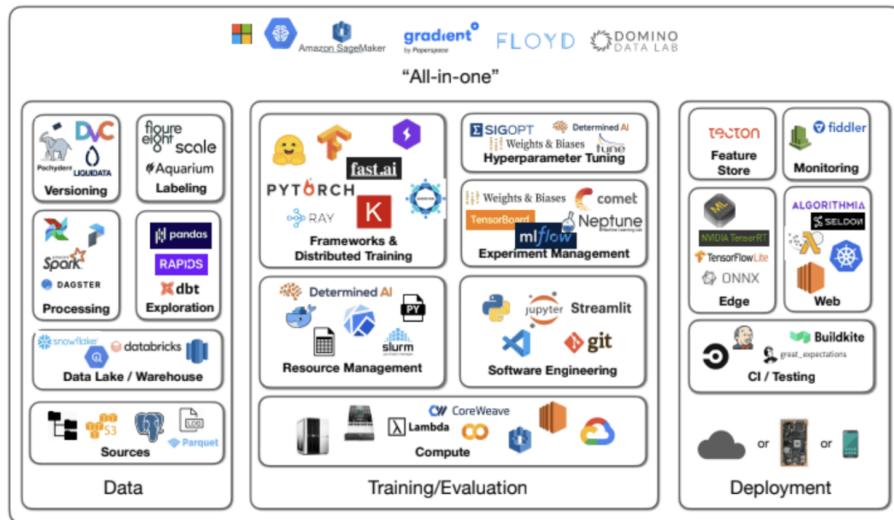
MLOps es un término que se refiere a las prácticas y herramientas utilizadas para la implementación, el despliegue, la gestión y el monitoreo de modelos de aprendizaje automático (machine learning) en producción. MLOps es una combinación de las palabras "machine learning" y "operaciones".

El objetivo de MLOps es mejorar la eficiencia y la eficacia de la implementación de modelos de aprendizaje automático, y garantizar que los modelos funcionen correctamente en el entorno de producción. Las prácticas y herramientas de MLOps se utilizan para automatizar el ciclo de vida de los modelos de aprendizaje automático, desde el desarrollo y el entrenamiento hasta la implementación y el monitoreo en producción.

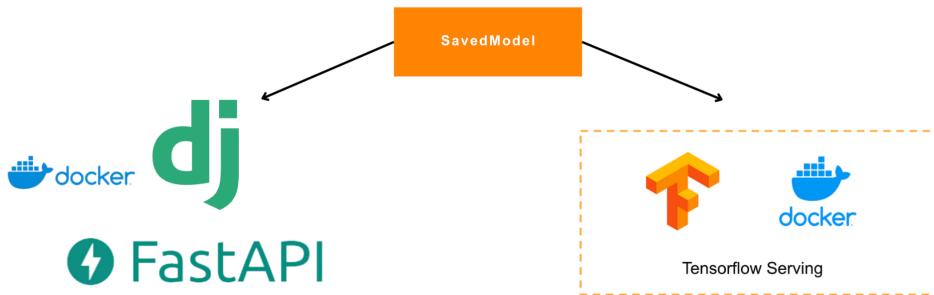


Algunas de las prácticas y herramientas de MLOps incluyen:

- Gestión de versiones de modelos: Controlar y gestionar las diferentes versiones de los modelos, sus dependencias y configuraciones, para asegurar su reproductibilidad y trazabilidad.
- Automatización del flujo de trabajo: Automatizar las etapas del proceso de implementación de los modelos, incluyendo la integración continua, el entrenamiento automático y el despliegue continuo.
- Monitoreo y gestión de modelos: Establecer métricas y umbrales para monitorear el rendimiento de los modelos en producción, y tomar medidas correctivas en caso de fallas.
- Gestión de datos: Gestionar el acceso y la seguridad de los datos utilizados por los modelos, y asegurar su calidad y consistencia.
- Gestión de infraestructura: Gestionar y monitorear la infraestructura de cómputo y almacenamiento utilizada para implementar y ejecutar los modelos.



TensorFlow Serving:



TensorFlow Serving es una plataforma de servidor de modelos de aprendizaje automático (machine learning) de código abierto desarrollada por Google. Permite la implementación de modelos de aprendizaje profundo en producción de manera escalable, eficiente y flexible.

TensorFlow Serving está diseñado para admitir una variedad de modelos de aprendizaje automático, incluyendo redes neuronales profundas, modelos de regresión y clasificación, y modelos de agrupamiento y de series de tiempo. También admite diferentes formatos de modelo, como TensorFlow, Keras y SavedModel.

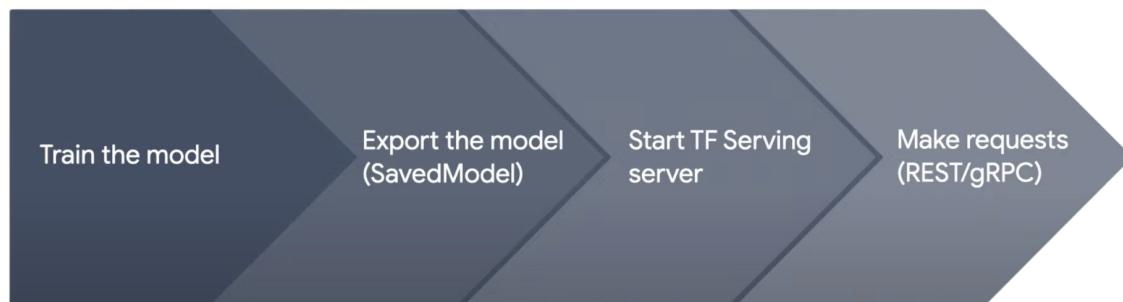
Los modelos se pueden implementar en TensorFlow Serving a través de una API RESTful o gRPC, lo que permite una integración sencilla con aplicaciones y sistemas existentes. El servidor de TensorFlow también proporciona una serie de herramientas de monitoreo y gestión de modelos, incluyendo estadísticas de uso y rendimiento, y herramientas para actualizar y gestionar diferentes versiones de los modelos.



TensorFlow Serving se utiliza comúnmente en aplicaciones empresariales y de producción que requieren implementar modelos de aprendizaje automático a gran escala y en tiempo real. Algunos ejemplos de casos de uso de TensorFlow Serving incluyen la detección de fraude en

tiempo real, la optimización de motores de búsqueda, la personalización de recomendaciones de productos y la identificación de objetos en imágenes.

El pipeline al usar tensorflow Serving se puede ver a continuación:



Aprendizajes:

- Hay que entender a un modelo de ML como un componente más en nuestro sistema.
- Tenemos dos opciones de despliegue con Tensorflow, sin embargo, es mejor usar TF Serving debido a que está optimizado y cuenta con las dependencias necesarias para el uso del modelo.
- MLOPS es un conjunto de mejores prácticas para administrar todo el ciclo de vida del aprendizaje automático.
- El objetivo de MLOps es optimizar el proceso de aprendizaje automático y hacerlo más confiable, escalable y eficiente.
- Los componentes de MLOPS son el control de versiones, la integración y el despliegue continuos, la gestión de datos, la validación de modelos, la supervisión y el registro, y los bucles de retroalimentación e iteración.
- MLOps ayuda a los científicos e ingenieros de datos a trabajar de manera más eficiente, colaborar de manera más efectiva y ofrecer modelos y aplicaciones de aprendizaje automático de mejor calidad a los usuarios finales.

Referencias:

[1] http://introtodeeplearning.com/slides/6S191/MIT_DeepLearning_L1.pdf Accessed: 2023-05-05

[2] <https://www.youtube.com/watch?v=tIeHLnjs5U8>

[3] "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville. This is a comprehensive textbook on deep learning, covering topics such as feedforward networks, convolutional networks, recurrent networks, and deep generative models. The book is available for free online and is widely regarded as one of the most authoritative references on the subject.

[4] "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron. This is a practical guide to machine learning and deep learning, covering both theory and implementation using the popular Python libraries Scikit-Learn, Keras, and TensorFlow. The book includes numerous examples and exercises, making it a great resource for those looking to gain hands-on experience with deep learning.

[5] "Deep Learning for Computer Vision" by Rajalingappaa Shanmugamani. This book focuses specifically on deep learning techniques for computer vision, covering topics such as image classification, object detection, and semantic segmentation. The book provides a mix of theory and practical implementation, with code examples using the popular Python library PyTorch. It is a good reference for those interested in applying deep learning to computer vision problems.