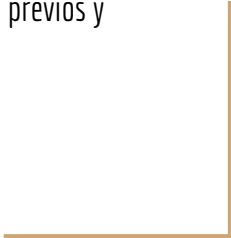




Clase 2

Organización del curso, conocimientos previos y
modalidad de trabajo



Organización del curso

Unidad 1: Aprendizaje automático

En la primera parte comenzaremos por comprender los conceptos y técnicas básicas del aprendizaje automático para poder terminar implementando un perceptrón multicapa funcional desde cero.

Unidad 3: Procesamiento de lenguaje natural

En esta unidad introduciremos las redes neuronales recurrentes (RNNs), modelos que explotan la estructura secuencial de los datos, usados comúnmente para el procesamiento de lenguaje natural. También le dedicaremos una buena parte de esta unidad a los mecanismos de atención que en los últimos años han desplazado a las RNNs en este tipo de tareas.

Unidad 2: Visión por computadora

A partir de la segunda unidad nos enfocaremos en técnicas modernas de aprendizaje profundo. Vamos a introducir las redes neuronales convolucionales (CNNs) para el procesamiento de imágenes, empezando por la simple LeNet hacia arquitecturas más recientes como ResNet.

Unidad 4: Sistemas de recomendación

Finalmente en una unidad breve, veremos sistemas de recomendación —necesidad frecuente de la industria— y de cómo el aprendizaje profundo habilita la factorización de matrices de grandes proporciones.

Organización del curso

Cada una de estas secciones constará de varias clases en donde trataremos de balancear:

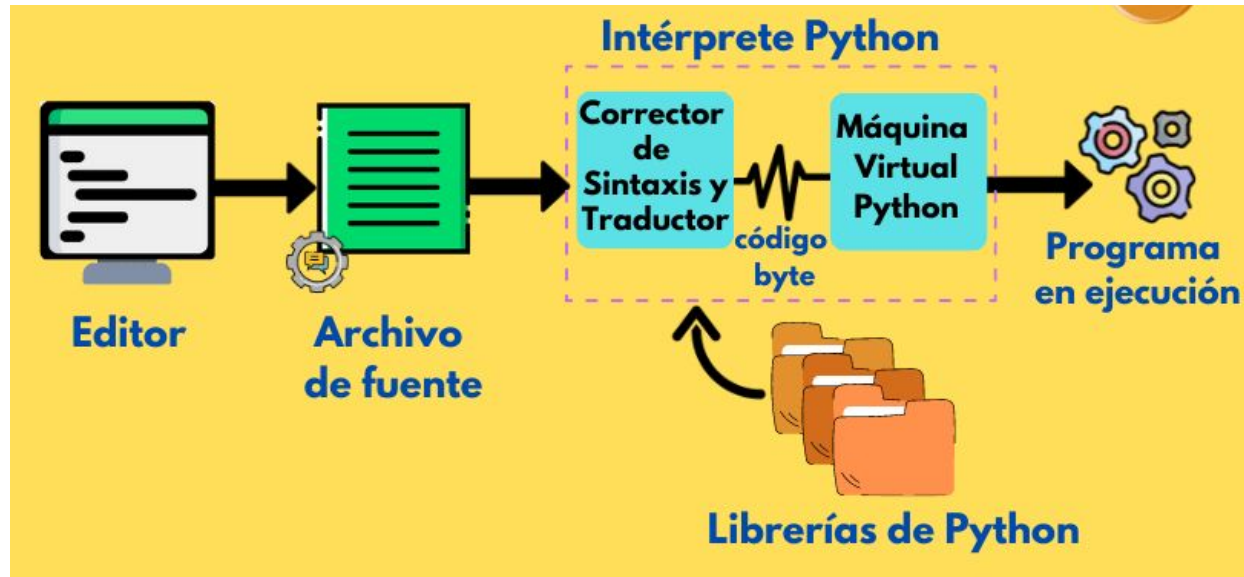
- **Conceptos teóricos:** abordados desde una forma intuitiva sin descuidar la explicación matemática que le da sustento.
- **Estado del arte:** descripción de las arquitecturas que supieron ser exitosas en cada disciplina
- **Implementación:** programación de los modelos en Pytorch

Requisitos para el curso

- Hardware:
 - Local
 - En la nube
- Software
 - Paquetes y librerías
 - Frameworks de Deep Learning
- Conocimientos
 - Previos: Python, Álgebra Lineal, Cálculo y Probabilidad.
 - Clase de hoy: Automatización de gradientes y Manipulación y Procesamiento de datos.

¿Qué es Python?

Es un lenguaje de programación **interpretado** de propósito general. Si bien al principio fue pensado como un lenguaje de scripting, ahora también se utiliza Python para programar grandes aplicaciones comerciales.



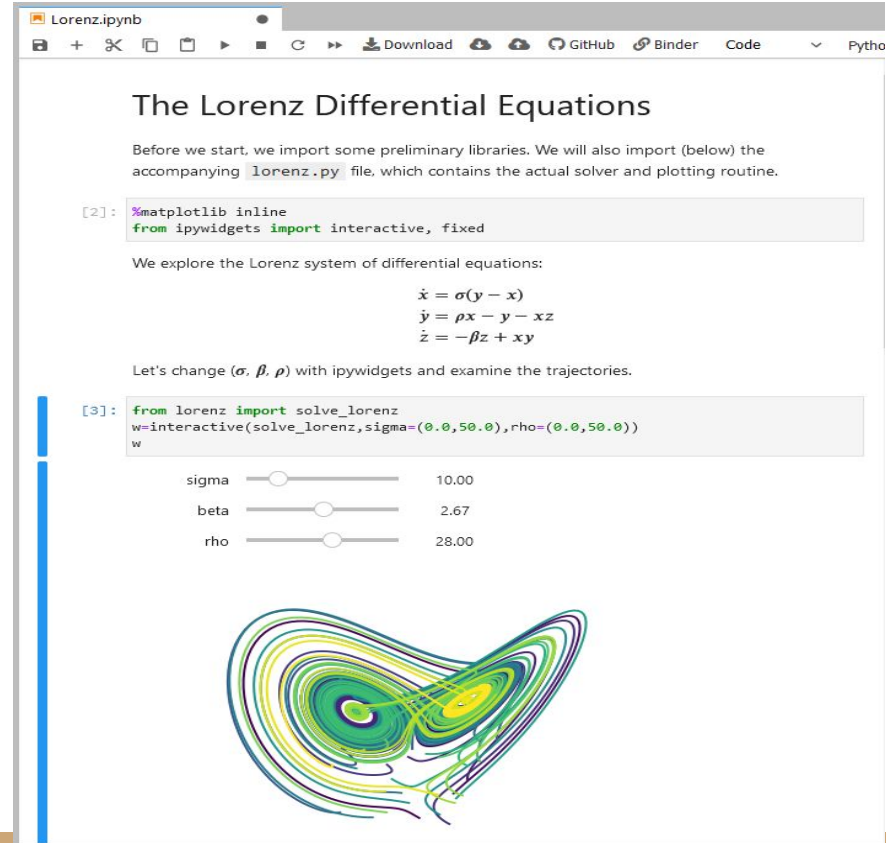
¿Por qué Python?

- Popularidad creciente
- Soporte de la comunidad
- Barreras de entrada bajas
- Legibilidad
- **Frameworks de Deep Learning**

Jupyter Notebook

Jupyter Notebook es una aplicación web de código abierto que nos va a permitir crear y compartir documentos con código en vivo, ecuaciones, visualizaciones y texto explicativo.

[Aquí](#) un instructivo de cómo instalarlo en tu máquina.



Google Colab

Google Colaboratory es un entorno gratuito de Jupyter Notebook que no requiere configuración y que se ejecuta completamente en la nube.

- Es **gratuito** y forma parte de la suite de aplicaciones de Google en la nube
- Los notebooks se conectan a una **VM** de Google Compute Engine que un inicio cuenta con 12 GB de RAM y 50 GB de almacenamiento en disco disponibles para el uso.
- El tiempo máximo que podemos estar conectados a una misma máquina desde un notebook es de **12 horas**.
- Nos permite cambiar los ajustes del entorno para utilizar una **GPU o TPU** de forma gratuita.
- Tiene integración con **Google Drive** para la carga y descarga de archivos de manera casi inmediata.

Frameworks de Deep Learning

Son interfaces, librerías o herramientas que nos permiten construir modelos de aprendizaje profundo de manera más fácil y rápida, sin entrar en los detalles de los algoritmos subyacentes. Proporcionan una forma clara y concisa de definir modelos utilizando una colección de componentes optimizados y prediseñados.

- Optimizado para el rendimiento
- Codificación de alto nivel
- Paralelización de los procesos
- Cálculo automático de los gradientes
- Soporte de la comunidad

Pytorch



PyTorch es una interfaz para el framework de deep learning Torch que se puede utilizar para construir redes neuronales profundas y ejecutar cálculos de tensores.

[Aquí](#) un instructivo de cómo instalarlo en tu máquina o en Colab.

Álgebra Lineal

Repasemos brevemente los conceptos básicos del álgebra lineal que se necesitarán para comprender e implementar la mayoría de los modelos cubiertos en este libro.

- Tensores: escalares, vectores y matrices.
- Productos escalares entre vectores.
- Productos entre matrices.
- Normas.

Cálculo

En deep learning, entrenamos modelos, actualizándolos sucesivamente para que sean cada vez mejores a medida que ven más y más datos. Esto no es más que un problema de optimización para lo cual necesitamos entender [algunos conceptos de cálculo](#).

- Derivadas y diferenciación.
- Derivadas parciales.
- Gradientes.
- Regla de la cadena.

Probabilidad

De una forma u otra, el aprendizaje automático se trata de hacer predicciones. Para ello, necesitamos utilizar el [lenguaje de la probabilidad](#).

- Axiomas de probabilidad.
- Variables Aleatorias.
- Probabilidad conjunta, condicional y Teorema de Bayes.
- Probabilidad Marginal.
- Independencia de sucesos.
- Esperanza, varianza y desviación estándar.

Manipulación de Datos

Para lograr algo en ciencia de datos, necesitamos alguna forma de almacenar y manipular los datos. Generalmente, hay dos cosas importantes que debemos hacer con los datos:

1. adquirirlos
2. procesarlos una vez que estén dentro de la computadora

No tiene sentido adquirir datos sin alguna forma de almacenarlos, así que ensuciemos nuestras manos primero [jugando con datos sintéticos](#).

Preprocesamiento de Datos

Hasta ahora hemos introducido una variedad de técnicas para manipular datos que ya están almacenados en tensores.

Para aplicar el aprendizaje profundo a la resolución de problemas del mundo real, a menudo comenzamos con el preprocesamiento de datos “crudos” en lugar de aquellos datos bien preparados en formato tensorial.

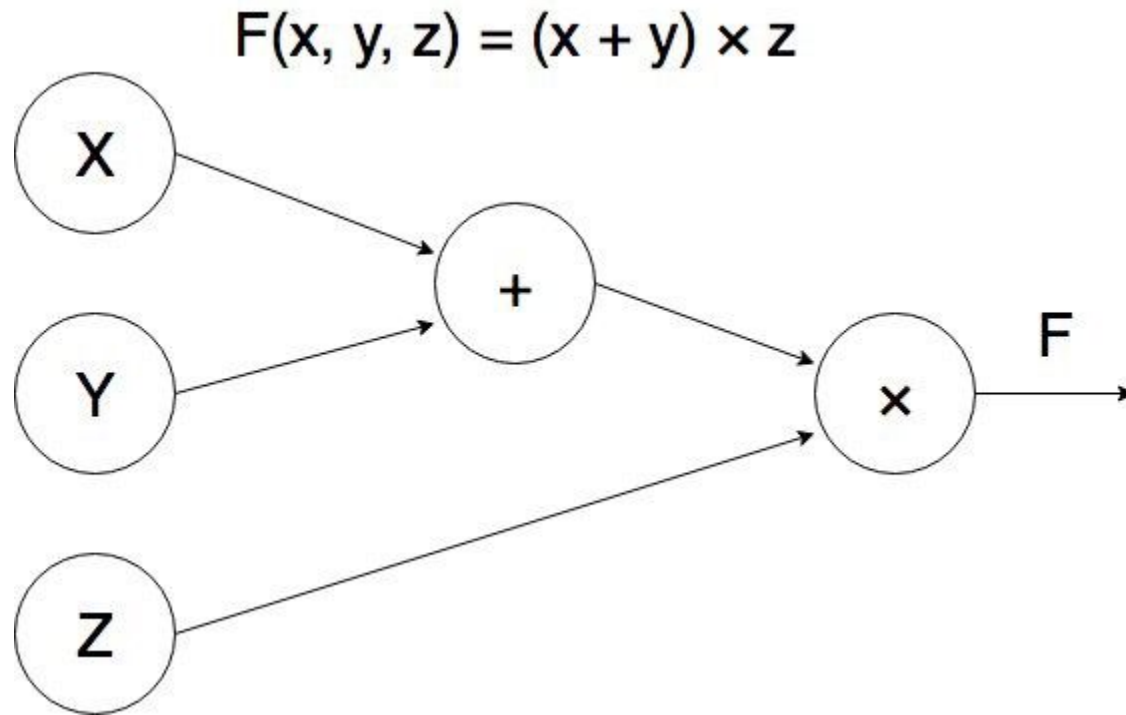
Por lo tanto, analizaremos brevemente los pasos para [preprocesar datos con pandas](#) y convertirlos al formato tensorial.

Diferenciación automática

La diferenciación es un paso crucial en casi todos los algoritmos de optimización de aprendizaje profundo. Si bien los cálculos para tomar estas derivadas son sencillos para modelos complejos, trabajar las actualizaciones “a mano” puede ser una molestia y propenso a errores.

En la práctica, los frameworks de aprendizaje profundo aceleran este trabajo calculando automáticamente las derivadas.

Diferenciación automática: Grafos Computacionales



Diferenciación automática: Grafos Computacionales

