Introducción a la Inteligencia Artificial
Facultad de Ingeniería
Universidad de Buenos Aires

Ing. Lautaro Delgado
(lautarodc@unops.org)

Clase 1: Introduccion a AI. Python y Numpy para AI. Ejercicio de Aplicación (K-means).

Clase 2: Ejercicio Integrador Pipeline AI (Regresión Lineal).

Clase 3: Aprendizaje estadístico (Regresión Lineal). Esperanza Condicional, ECM, Máxima Verosimilitud.

Clase 4: Bayes y estimadores puntuales (bias, variance, etc).

Clase 5: Gradientes, Optimización, Hiperparámetros y Regularización.

Clase 6: Regresión Logística. Clasificación binaria. Softmax.

Clase 7: Algoritmos no supervisados. Expectation Maximization.

Clase 8: Examen.

El régimen de aprobación de la materia es simple:

- Trabajos prácticos a implementarse y entregar durante las clases.
- Un examen final online (teórico y práctico) en la clase 8.

Dinámica esperada para las clases:

- 50 minutos de teoría
- 10 minutos de descanso
- 50 minutos de ejemplos y ejercicios
- 10 minutos de descanso
- 60 minutos de programación

Durante la cursa vamos a utilizar las siguientes herramientas:

- Lenguaje de programación:
  - Python 3.8
  - Herramienta pip para instalar librerias de codigo y dependencias
- Librerias de codigo:
  - Numpy 1.18
  - SciPy 1.5
- Consola interactiva de Python:
  - iPython
- Herramientas:
  - PyTest para tests y GitHub para repositorios
- IDE recomendado:
  - PyCharm community edition: https://www.jetbrains.com/es-es/pycharm/

La bibliografía es solo a modo de sugerencia y no será obligatorio el uso de dicho material. El curso está diseñado para ser completamente autocontenido.

- The Elements of Statistical Learning | Trevor Hastie | Springer
- An Introduction to Statistical Learning | Gareth James | Springer
- Deep Learning | Ian Goodfellow | https://www.deeplearningbook.org/
- Stanford | CS229T/STATS231: Statistical Learning Theory
- Mathematics for Machine Learning | Deisenroth, Faisal, Ong

Índice

Definición de Artificial Intelligence (AI)

Pensar humanamente	Pensar racionalmente
"AI is the automation of activities that we associate with human thinking, activities such as decision-making, problem solving, and learning" (Bellman, 1978)	"AI is the study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)"
Actuar humanamente	**Actuar racionalmente**
"AI is the study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	"AI is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

proceso del pensamiento

proceso del comportamiento

comparación de éxito contra el resultado humano

comparación de éxito contra el resultado ideal

Definición de Artificial Intelligence (AI)

- Test de Turing propuesto por Alan Turing en 1950.
- “Una computadora pasa el Test de Turing si un interrogante humano no puede diferenciar si las respuestas provienen de otra persona o de una computadora”
- El Test de Turing prohíbe interacción física directa entre el interrogante y la computadora, ya que según el test simular físicamente un humano es innecesario para definir inteligencia.
- Pasar el Test de Turing rigurosamente requiere mucho trabajo:
    - Natural Language Processing
    - Knowledge Representation
    - Automated Reasoning
    - Machine Learning

Definición de Machine Learning (ML)

"Es la capacidad de un agente de Inteligencia Artificial de mejorar su rendimiento en futuras tareas después de hacer observaciones en el mundo donde el algoritmo se desempeña."

Porque el aprendizaje a partir de la observación es importante?

1. Es imposible anticipar todas las situaciones en la que se encontrara el agente.
2. Es imposible anticipar todos los cambios en el tiempo que se producirán.
3. Es imposible dar una solución sin usar al aprendizaje por observaciones.

Definición de Machine Learning supervisado

"El agente de AI observa ejemplo de datos de entrada-salida y aprende una función que mapea la la entrada a la salida."

Ejemplo: análisis de sentimiento de frases.

En términos matemáticos, a grandes rasgos:

Observar ejemplos del vector aleatorio $\vec{X}$

Aprender a predecir Y a partir de observaciones del vector aleatorio X, estimando la función densidad de probabilidad de Y condicionada al vector aleatorio X.

$$f_{Y|\vec{X}}(y|\vec{x})$$

Definición de Machine Learning no supervisado

"El agente de AI observa datos y aprende patrones sin necesidad de utilizar feedback explícito."

Ejemplo: clusterización de clientes en un CRM.

En términos matemáticos, a grandes rasgos:

Observar ejemplos del vector aleatorio $\vec{X}$

Aprender implícitamente o explícitamente la función de densidad conjunta aproximada del vector aleatorio X:

$$f_{\vec{X}}(\vec{x})$$

Relación entre modelos supervisados y no supervisados (a grandes rasgos)

Podemos resolver el problema no-supervisado como n problemas supervisados.

$$f_{X_2,X_1}(x_2,x_1) = f_{X_2|X_1}(x_2|x_1)f_{X_1}(x_1)$$

$$f_{X_3,X_2,X_1}(x_3,x_2,x_1) = f_{X_3|X_2,X_1}(x_3|x_2,x_1)f_{X_2,X_1}(x_2,x_1)$$

$$f_{X_3,X_2,X_1}(x_3,x_2,x_1) = f_{X_3|X_2,X_1}(x_3|x_2,x_1)f_{X_2|X_1}(x_2|x_1)f_{X_1}(x_1)$$

$$f_{\vec{X}}(\vec{x}) = \prod_{i=1}^{n} f_{X_i|X_1,\ldots,X_{i-1}}(x_i|x_1,\ldots,x_{i-1})$$

Podemos resolver el problema supervisado a partir de uno no supervisado.

$$f_{Y|\vec{X}}(y|\vec{x}) = \frac{f_{Y,\vec{X}}(y,\vec{x})}{f_{\vec{X}}(\vec{x})}$$

Definición de Machine Learning semi-supervisado

"El agente de AI combina técnicas no-supervisadas y supervisadas para resolver problemas. Por ejemplo, comienza utilizando una técnica no-supervisada para clusterizar items. Luego asigna nombres a los clusters y utilizada esa información para entrenar un algoritmo supervisado que permita clasificar nuevos items".

clusterización -> asignación de nombres a los clusters -> clasificación

Definición de Reinforcement Learning (RL)

"El agente de AI aprende a partir de feedback explícito obtenido a través de una función de recompensas y castigos. El agente conoce las acciones que puede ejecutar y a medida que ejecuta las acciones aprende con el objetivo de maximizar la recompensa."

Ejemplo: un agente de AI que juega al ajedrez

Definición de Deep Learning

"Deep Learning es una técnica para problemas de ML supervisados, que agrega layers y parámetros para aprender complejos mapeos de entrada-salida, utilizando modelos no lineales."

"Ian Goodfellow"

(1) Resolución de problemas de ML supervisados por fuerza bruta.

(2) En general, requiere grandes volúmenes de datos.

(3) En general, requiere mucho poder de cómputo matricial mediante GPUs.

Definición de AI | Deep Learning

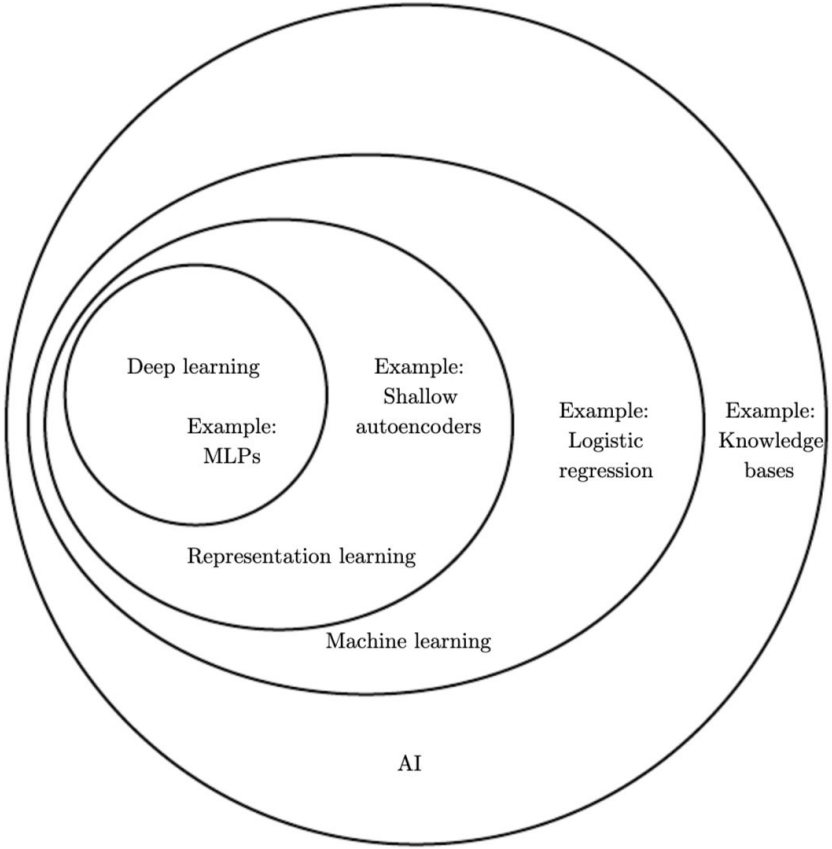
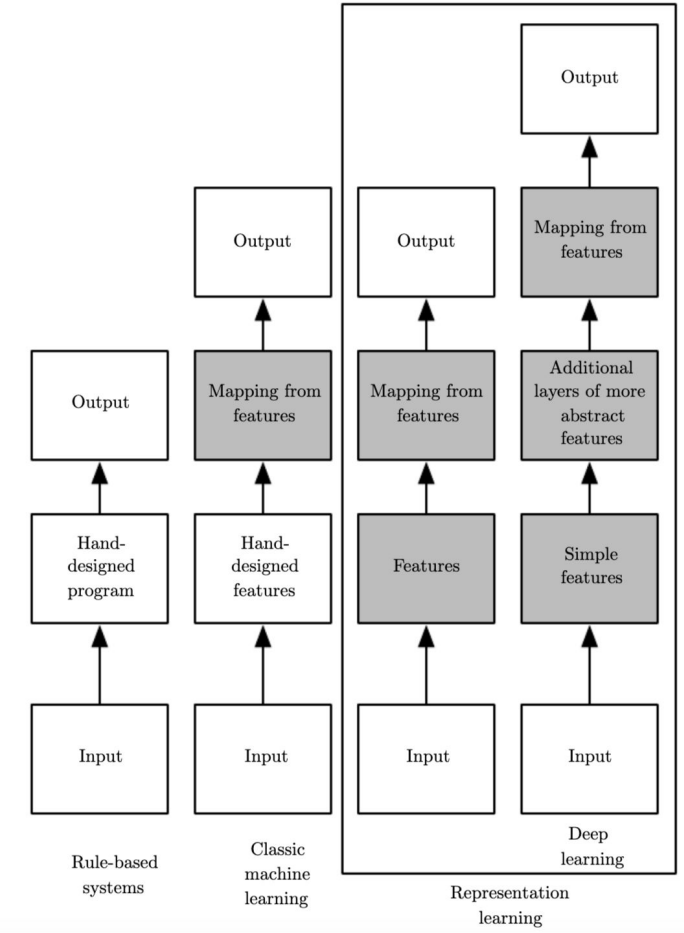
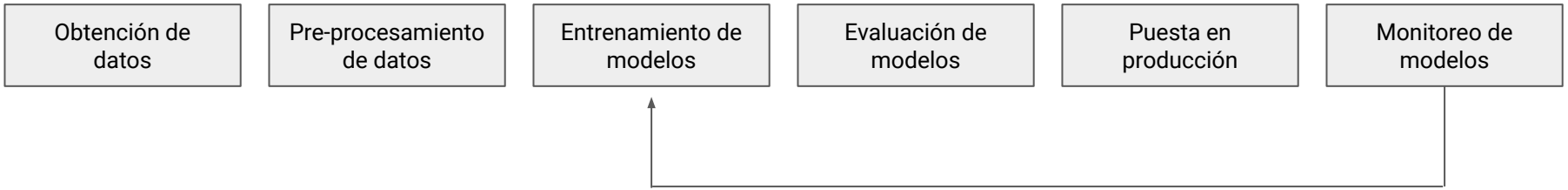


Diagrama de Venn de algoritmos



Bloques que se aprenden en Deep Learning



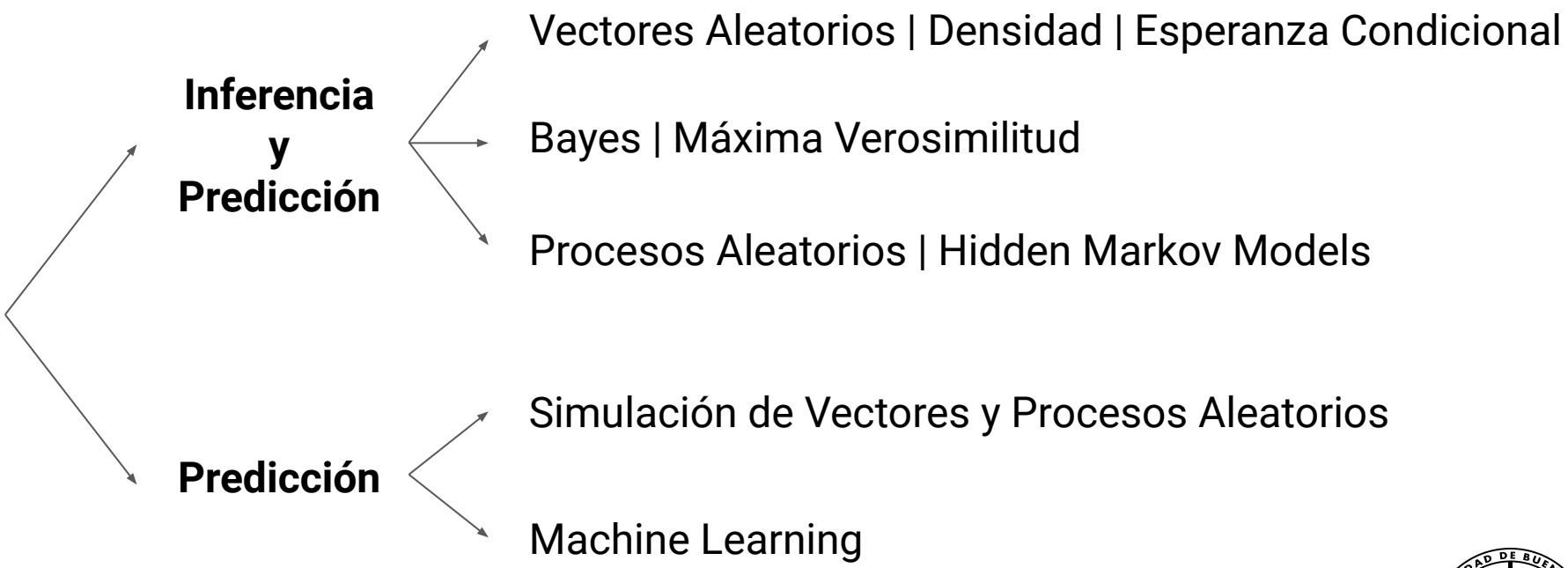
Data Pipeline Engineer

Data Scientist

AI Researcher

Machine Learning Engineer

Estrategias



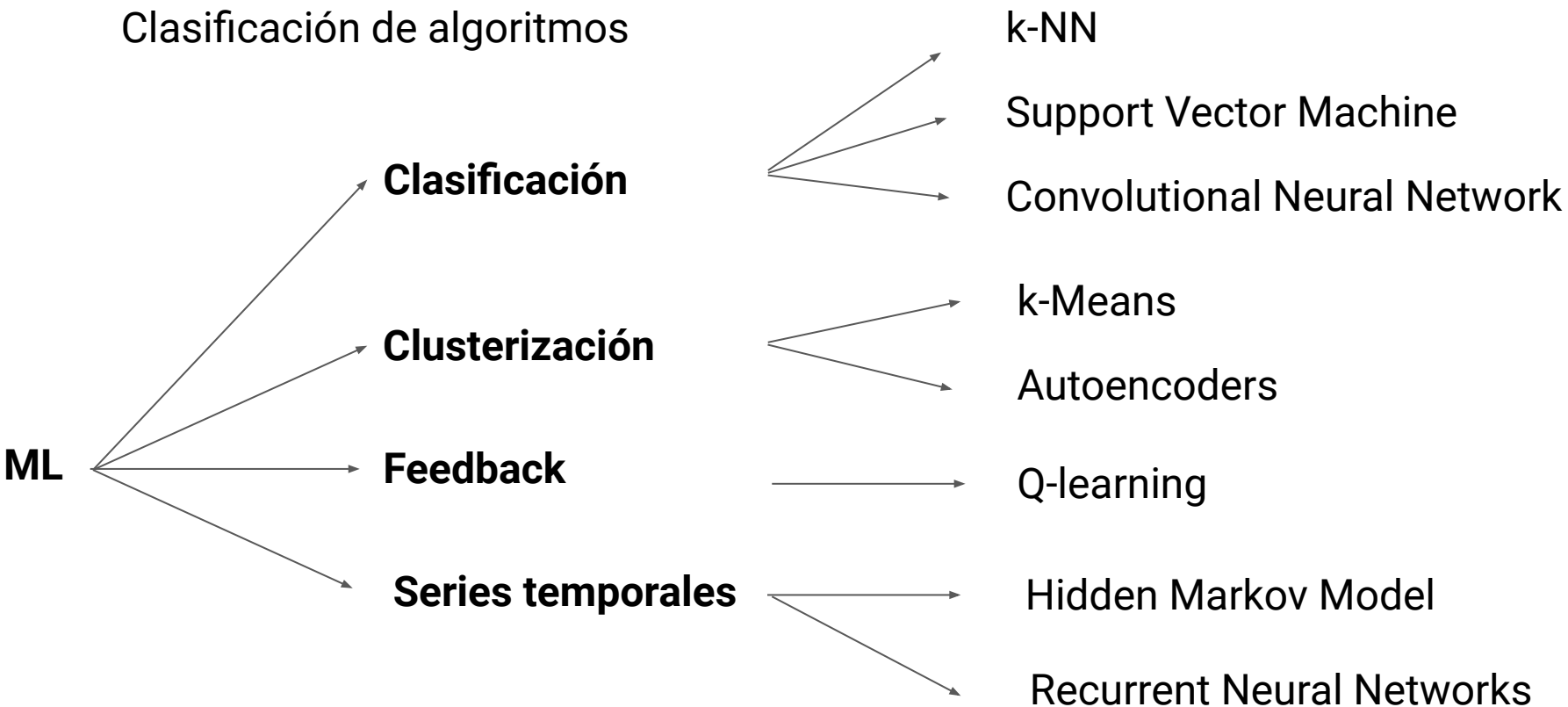
“When you’re fundraising, it’s AI. When you’re hiring, it’s ML. When you’re implementing, it’s linear regression. When you’re debugging, it’s printf().”

Baron Schwartz

Clasificación de algoritmos de AI | Cuando usar cada estrategia?

X es un vector aleatorio, Y es un vector aleatorio, n es cantidad features de X

Datos	Modelo	Problema	Estrategia
Conozco densidad de X, n chico	Conozco relación entre X e Y	Inferencia y predicción	Teoría de probabilidad
Conozco densidad de X, n chico	Conozco Y si condiciono a X	Inferencia y predicción	Esperanza condicional
Conozco familia de X, n chico	Conozco Y si condiciono a X	Inferencia y predicción	Teoria de Bayes
Conozco familia de X(t), n chico	Conozco modelo del proceso	Inferencia y predicción	Procesos aleatorios
Conozco familia de X(t), n grande	Conozco modelo del proceso	Predicción	Simulación
No conozco X, n grande	No conozco relación entre X e Y	Predicción	Machine Learning supervisado
No conozco X, n grande	-	Predicción	Machine Learning no supervisado


Clasificación de algoritmos
ML
Clasificación
Clusterización
Feedback
Series temporales
k-NN
Support Vector Machine
Convolutional Neural Network
k-Means
Autoencoders
Q-learning
Hidden Markov Model
Recurrent Neural Networks

Herramientas de Análisis Matemático

- Optimización
  - Gradiente
  - Gradiente estocástico
  - Gradiente mini-batch
  - Optimización convexa y no-convexa
  - Mínimo/Máximo locales y absolutos

- Algebra lineal
  - Normas y regularización
  - Factorización de matrices
  - Descomposición en Valores Singulares
  - Análisis de Componentes Principales

- Pre-procesamiento
  - Transformada de Fourier

Herramientas de Probabilidad y Estadística

- Aprendizaje estadístico
  - Variables Aleatorias, Función Densidad y Función Distribución de Probabilidad
  - Vectores Aleatorios, Densidad Conjunta
  - Esperanza Condicional
  - Enfoque Bayesiano

- Simulación
  - Procesos estocásticos
  - Hidden Markov Model

- Análisis de datos
  - Estimadores Puntuales
  - Estimadores por Intervalos
  - Test de Hipótesis
  - Test de Bondad de Ajuste
  - Teorema Central del Límite

Herramientas de programación para AI

- Análisis y desarrollo de modelos de ML:
  - Scikit-learn
  - Deep Learning:
    - PyTorch | TensorFlow | Keras
- Modelos de ML en producción:
  - Numpy
- ML Distribuido:
  - Apache Spark

Python


Python
Features
Easy to Write
Easy to Understand
Object-Oriented
Robust Standard Libraries
Supports Various Programming Paradigms
Support for Interactive Mode
Dynamically Typed and Type Checking
Databases and GUI Programming
Extensible
Portable
Scalable
Integrated
Automatic Garbage Collection
Free and Open Source

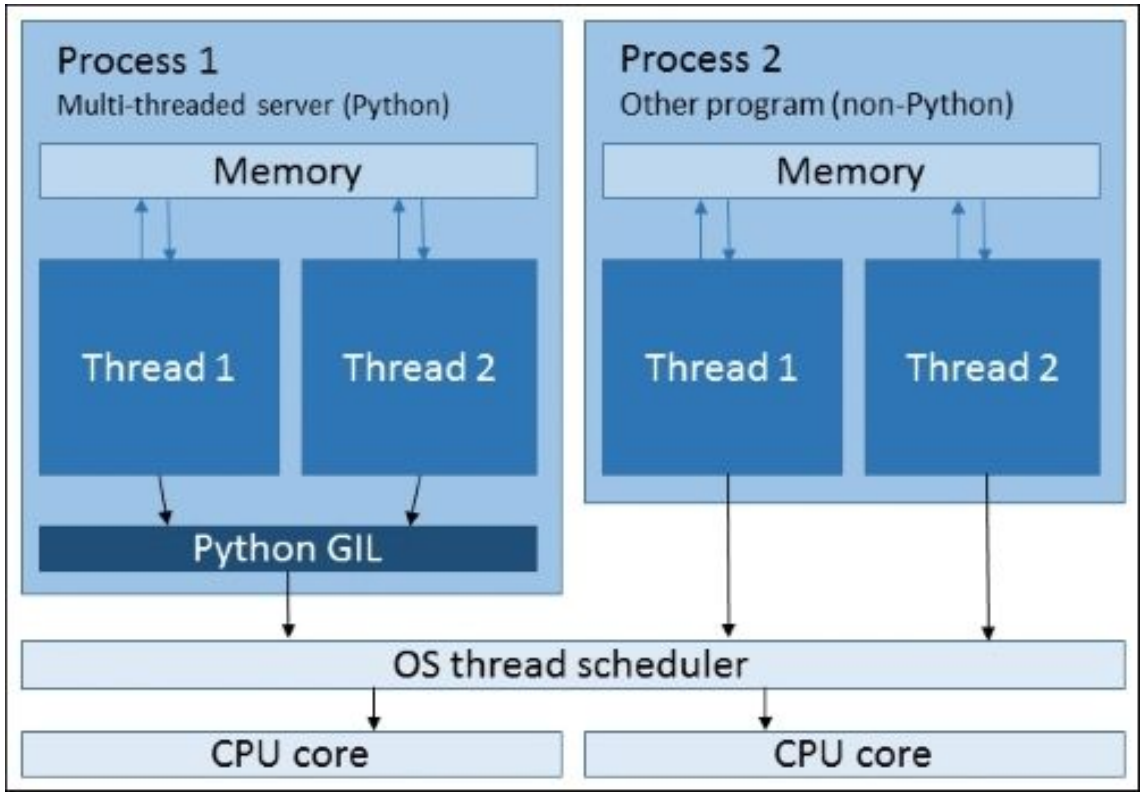
GIL, Threads y Procesos

- **GIL: Global Interpreter Lock.** Es un mutex/lock que permite que un solo thread de python mantenga el control del intérprete. Esto significa que por más que escribamos un programa en Python que sea multi-thread, si el programa es CPU-intensivo, va a comportarse como si fuese single-thread. Por ejemplo, si estamos haciendo una operación intensiva de procesamiento de imagen, el thread ejecutando dicha tarea no permitirá que otros threads se ejecuten. En programas del tipo IO-intensivo, no es un problema porque los threads estan la mayoria del tiempo esperando datos (base de datos, protocolos de comunicación, etc.).

- **Multi-threading:** Python usa un único procesador, pero hay muchos threads ejecutándose concurrentemente. Debido al GIL, threading es útil en programas IO-intensivos, pero no es útil en programas CPU-intensivos. Esto va a tener un impacto interesante en problemas de Machine Learning y servidores HTTP/HTTPs como uWSGI. En general, los problemas de Machine Learning son CPU-intensivos. **Como podemos hacer escalar un servidor con modelos de Machine Learning?**

- **Multi-processing:** Python usa múltiples procesadores del servidor, y cada procesador responde a su propio GIL. Esto soluciona en gran medida el problema de programas CPU-intensivos, pero no es simple compartir la memoria. En general, los modelos de Machine Learning están representados por grandes matrices en memoria. **Como podemos escalar la lectura y escritura de estas matrices?**

GIL, Threads y Procesos



GIL, Threads y Procesos

- **Como podemos hacer escalar un servidor con modelos de Machine Learning?**
  - Podemos agregar más procesos.

- **Como podemos escalar la lectura de matrices para modelos de Machine Learning?**
  - En sistemas operativos basados en Linux, la creación de procesos hijos desde un proceso padre, utiliza la metodología copy-on-write. Esto significa, que la memoria entre procesos no va a ser copiada, siempre y cuando no editemos la memoria desde el thread hijo.

- **Como podemos escalar la escritura de matrices para modelos de Machine Learning?**
  - Es muy común necesitar editar las matrices en tiempo-real para actualizar y mejorar los modelos. Esta operación suele ser bastante compleja, y la mayoría de las soluciones en el mercado solucionan este problema a través de solo entrenar los modelos en producción, cada un periodo determinado (entrenar nuevo modelo -> eliminar modelo viejo -> subir modelo nuevo).
    - Alternativa 1: copiar la memoria y usar patrones pub/sub.
    - Alternativa 2: utilizar caché como Redis.
    - Alternativa 3: machine-learning distribuido.