Apuntes Semana 1- 20/02/2025

Elaborado por:
Yessenia Solano Retana-2021050802

***Abstract*—El presente documento recopila los apuntes de la primera semana de clases, donde se abordan conceptos básicos e introductorios al curso de Inteligencia Artificial. Se presentan definiciones fundamentales, un panorama general de las aplicaciones de la IA y una introducción a los principales enfoques utilizados en el campo. Además, se incluyen ejemplos ilustrativos para facilitar la comprensión de los temas tratados .**

Aspectos Administrativos

Se retomaron aspectos administrativos vistos en la primera clase, entre ellos:

- La clase presencial será los **martes** y la clase virtual los **jueves**.
- Todos los **martes habrá un quiz**, con la materia vista en la semana anterior.
- Existirá un sistema de **apuntadores**, habrá dos por clase y cada estudiante debe hacer dos apuntes por semestre. Este apunte debe subirse al **Tec Digital** en la carpeta de Apuntadores.
- El documento de los apuntes debe seguir el **formato IEEE** de doble columna e incluir nombre y carné del estudiante.

Herramientas de IA Utilizadas en el Curso

El curso hará uso de las siguientes herramientas de **Inteligencia Artificial**:

- **Lenguaje: Python (Python Cookbook)**
- Numpy
- Scikit-learn
- PyTorch-Lightning
- WandB
- Matplotlib
- Jupyter Notebooks (Se puede usar desde Visual Studio Code)
- Google Colab

Artículo de Interés

SWE-Lancer: Can Frontier LLMs Earn $1 Million from Real-World Freelance Software Engineering?

Este artículo es un **benchmark** lanzado por **OpenAI** que contiene varios códigos de software o trabajos que podrían ser realizados en modalidad de **freelance** y que están evaluados en **$1 millón**.

En este artículo se menciona que existen más de **1,400 proyectos de software freelance** valorados en **$1 millón**, con rangos de precios que varían desde **$50** hasta **$32,000**.

El **SWE-Lancer Benchmark** abarca una amplia gama de tareas, que incluyen desde implementaciones simples hasta desarrollos de funciones altamente complejas.

Introducción a la Inteligencia Artificial

Machine Learning (ML)

Arthur Samuel y el concepto de Machine Learning

En 1959, **Arthur Samuel** acuñó el término **Machine Learning (ML)** mientras trabajaba en IBM. Su tarea consistía en desarrollar una campaña de marketing para un nuevo producto, lo que lo llevó a crear un algoritmo capaz de simular inteligencia artificial en el juego de **damas chinas**.Este algoritmo permitió que la computadora aprendiera a jugar damas de manera autónoma, mejorando su desempeño con la práctica. Samuel presentó este avance como un

ejemplo de **Machine Learning**, aunque inicialmente lo promocionó más como una estrategia de marketing que como un desarrollo formal en Inteligencia Artificial.

¿Qué es Machine Learning?

El concepto de **Machine Learning** es construir máquinas que puedan hacer tareas sin ser explícitamente programadas, solamente infiriendo de los datos. Esto quiere decir, por ejemplo, que a la computadora no se le va a decir que es un perro y todas las reglas que implica ser un perro (tiene cola, ojos, raza, etc.), todo esto no se le va a pasar al algoritmo si no que solo se le va a dar una imagen y un feedback.

Por ejemplo, se tiene una imagen X y Y, con una gran cantidad de datos. Al modelo se le pasa toda la imagen como parámetro y el de forma implícita tiene que aprender a discernir que es una X o una Y implícitamente solo recibiendo la imagen, sin decirle cómo hacerlo. De forma resumida al modelo solo se le pasa el input y la etiqueta, con esto el modelo tiene que ser capaz de inferir o aproximar lo más posible el resultado. Para ilustrar lo anterior:

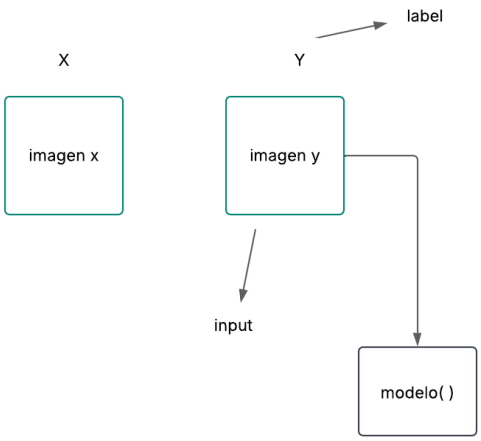

FIGURE 1. Machine Learning.

Vamos a crear un algoritmo que va a aproximar una función. Por ejemplo, todos tenemos "una función" implícita en nuestro cerebro que cuando vemos un perro sabemos que es un perro o cuando vemos un gato sabemos que es un gato, es decir visualmente el humano ve una imagen, esta pasa por una función y podemos identificar que lo que vimos es un perro o un gato.

Esta función matemáticamente también puede existir. Por ejemplo, una función con la siguiente forma:

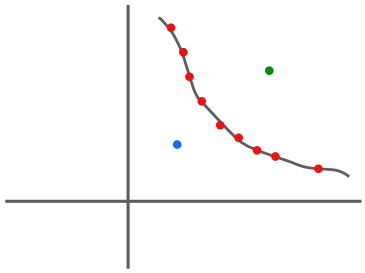

FIGURE 2. Ejemplo de una función matemática utilizada en Machine Learning.

Esta función sigue los datos que están en color rojo y cuando se vea un gato (punto color azul) se coloca o cuando se ve un perro (punto color verde) se coloca también. A partir de esto sabemos que dependiendo de la imagen que se este viendo (representado como punto) es un gato o un perro y existe. Lo que tratamos de hacer es aproximar lo más posible la función, estas serán muy útiles, pero no serán perfectas.

Por ejemplo, mi modelo puede hacer una función como la siguiente:

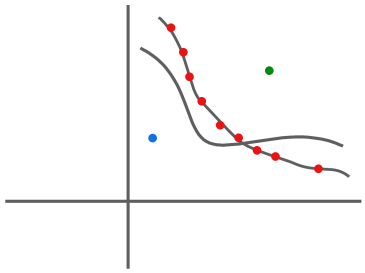

FIGURE 3. Ejemplo de aproximación de una función en Machine Learning.

La cual no necesariamente esta mal, si no que va a ser funcional para unos casos y para otros no. Por lo cual son aproximaciones de funciones que son útiles ya que nos van a ayudar a resolver la mayoría de los casos, pero puede presentar fallos.

En resumen, lo que queremos hacer es que a partir de los datos vamos a producir un programa que sea capaz de hacer una tarea sin programarla directamente, es decir solo infiriendo los datos.

En este existen dos artistas: Ciencia e Ingeniería

Machine Learning: Ciencia

Generar conocimientos: Con esto nos referimos a cuando se trata de crear modelos, dedicarse a técnicas de optimización de funciones, en como utilizar data que puede no estar etiquetada para entrenar un modelo o descubrir nuevas arquitecturas que puede simplificar o hacer más compleja una tarea, pero mejorando su rendimiento para lo cual se va a basar en gran medida en métricas.

Métricas:

Por ejemplo, si se produce ”n” cantidad de modelos a todos se les aplican las mismas métricas, en estos se puede ver que porcentaje de acierto tienen en los símbolos de clasificación, basados únicamente en el porcentaje de las métricas sin preocuparse por otros factores como el tiempo de rendimiento, capacidad de inferencia o cuanta GPU necesita correr el modelo, si no exclusivamente en algunas métricas establecidas. Por ejemplo, ilustrando lo anterior tenemos 3 modelos y escogemos el mejor, en este caso el B.

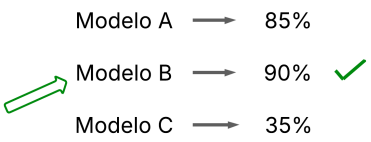

FIGURE 4. Comparación de modelos basados en métricas de precisión.

A partir de escoger el mejor modelo se hace un notebook y se construye el modelo desde 0, donde se utiliza la data que en caso de no estar limpia hay que limpiarla, para esto es común sentarse a explorar los datos, leer artículos, ver últimas tendencias en técnicas, además de analizar cuales arquitecturas son mejores para tratar de replicar y después a esto generalmente se hacen artículos. Por ejemplo, Google publica mucho de sus trabajos lo cual es de gran ayuda para todos.

Entonces en resumen el Machine Learning en la ciencia:

FIGURE 5. Resumen de Machine Learning en la ciencia.

En línea con esto hay un artículo que es de Google, sobre que liberaron una Inteligencia Artificial que hace hipótesis científicas a partir de los datos que se están analizando, este se probó con la parte médica y se encontraron muchos avances médicos, el nombre de este artículo es: **Towards an AI co-scientist**

Marchine learning: Ingeniería

FIGURE 6. Resumen de Machine Learning en la Ingeniería.

Ahora en la **Puesta en Producción de un modelo**, aquí lo primero que hay que sentarse a pensar es que hay que hacer un reafactor al código, ya que hay que seguir monitoreando el modelo o tal vez para cierto tiempo volverlo a entrenar, entonces hay que cambiar la estructura del modelo, para que quede un modelo igual, por ejemplo, un modelo de 50 GB, que es un modelo muy pesado como puede ser servido para un cliente.

Para esto hay que **transformar el modelo**, para esto entran muchas herramientas que pueden ser útiles, por ejemplo, si se tiene **Onnx** con esta herramienta se puede agarrar un código que este escrito en un framework especifico y lo va a transformar a una optimización haciéndole un código mucho más pequeño y ligero. También se pueden usar otras técnicas como que un modelo más pequeño imite las acciones que haga un modelo mucho más grande, esto permite que resuelvan las mismas tareas solo que uno es mucho más grande que el otro. Además, también otras técnicas llamadas quantization que reducen el número de bits que tienen los parámetros de los modelos, estos parámetros son para ajustar su salida, generalmente estos parámetros son de punto flotante, esto dependiendo de estos tamaños se va a tener un mejor rendimiento, este método mejora el rendimiento que es mucho más pesado.

En caso del **MLOps** se dedican a hacer el modelo, más detallado, hay un concepto que se llama Data Shift donde se entrena un modelo y sigue una distribución normal, pero conforme pasa el tiempo los patrones de comportamiento que se están analizando cambien y se comporten se comporten diferente, lo que va a hacer que inicie a fallar. Por lo tanto, una de las tareas de los MLOps es monitorear que este comportamiento no suceda y si sucede deben volver a entrenar el modelo.

Por ejemplo, si existía un modelo en el 2020 que seguía el comportamiento de consumo en los Malls, pero inicio la pandemia y el patrón de comportamiento

cambio, así que ese modelo hay que rentrenarlo.

Ahora considerando el siguiente grafico:

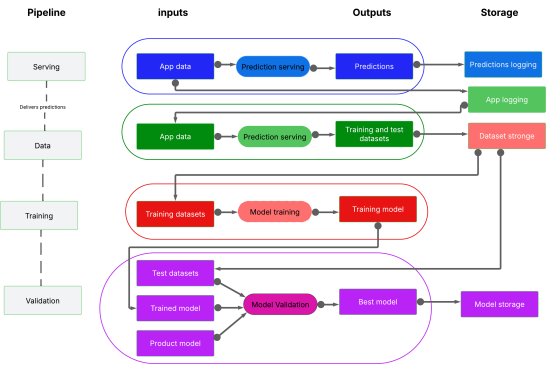

FIGURE 7. Resumen de Machine Learning en la Ingeniería.

Partimos de **DATA**, asumiendo el hecho de que tenemos los datos, a partir de esto vamos a producir un data set para ir al punto de **Training and test dataset**, esto con el objetivo de entrenar y hacer test, esto se guarda. Y así tenemos el **Dataset storage**.

El siguiente paso es **Training**, es decir vamos a entrenar nuestro modelo usando el **Training dataset** específicamente, en el proceso de entrenamiento esta la selección del modelo donde vemos que modelo vamos a utilizar y porque se va a usar.

Ahora una vez tenemos el modelo entrenado y funcionando bien vamos a pasar a **Validation** en esta etapa de la data que teníamos vamos a usar la data de testing, estos dataset uno es para entrenar el modelo y que aprenda el fenómeno natural y el otro son datos que el modelo jamás ha visto y sirven para simular datos de producción.

Una vez se valida el modelo, escogemos cual es el mejor de todos de acuerdo a una métrica o atributo de evaluación. A este punto pasamos al **Model storage** donde se escogerá la plataforma que se va a utilizar para el deployment.

Y finalmente vamos a poner el modelo a servir en **Perdiction serving** y todo el proceso de **Serving**. Esto en resumen es un diagrama con el flujo que se sigue naturalmente.

Inteligencia artificial

IA básicamente son programas que simulan la capacidad de razonar del ser humano o su comportamiento, aquí entran todos los algoritmos generativos y técnicas con simbolismo.

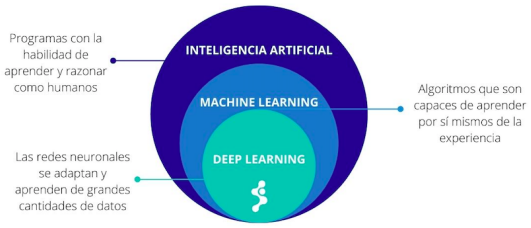

FIGURE 8. Relación entre Inteligencia Artificial, Machine Learning y Deep Learning.

Machine learning

Algoritmos que ocupan datos para entrenarse y siguen algoritmos estadísticos, regresión lineal, regresión logística, arboles de decisión entre otros. Pero independiente de estos el Machine Learnging ocupa datos para aprender.

Deep Learning

Este es un subconjunto del Machine Learning, este va a tener algoritmos multicapa de redes neuronales anidadas. Este aprende a resolver tareas con muy buen rendimiento, pero con grandes cantidades de datos. La explicación matemática de esto es que las redes neuronales son capaces de emular cualquier función no lineal cuando se anidan.

La diferencia entre estas 3 es que la inteligencia artificial es todo el concepto de comportamiento y el machine learning son los algoritmos que podemos llamar “clásicos” que ocupan data para poder aprender y el Deep learnign es diseñado para el consumo gigante de datos para tareas más específicas.

Tipos de aprendizaje en machine learning

Tenemos varias categorías, donde se encuentran las más generales, que se explican a continuación:

Supervisado: en este caso se puede tomar un data set que tiene datos por ejemplo X que representa un valor y puede tener un x1, x2 x3. Además, también hay un Y. Para entender bien este caso, se plantea un ejemplo, para esto pensemos en la predicción del precio de una casa, considerando factores como la cantidad de cuartos, la cantidad de ventanas cantidad de puertas, ect. Generalmente es lo que comprendería el vector. Mientras que el Y lo que representa es una etiqueta, esta diría cuanto vale que una casa tenga esas características, por ejemplo, asumamos que el valor de la casa con las características x1, x2, x3 puede valer

$200.000.

Entonces el aprendizaje supervisado usa ese X y el Y. Es decir, tenemos el modelo y le pasamos el X y le hacemos la comparación con un "Y" aproximado y también se compara con el Y que tenemos que representa la etiqueta. Si los valores del Y aproximado y el Y de la etiqueta son distintos, estos resultados van a ser el feedback que necesitamos para nuestro modelo, es decir le podemos indicar que se equivocó, esto con el fin de que mejore sus predicciones. Para ilustrar lo anterior tenemos:

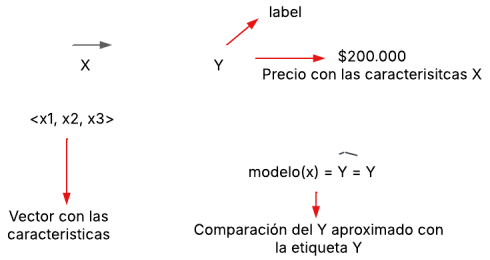

FIGURE 9. Ejemplo de aprendizaje supervisado.

En resumen, el aprendizaje supervisado siempre requiere de una etiqueta, esto permite algo que supervise el trabajo, es decir permite que se le indique al algoritmo que tan bien está haciendo su tarea.

Por otra parte, los Y que representan las etiquetas no siempre serán números si no que pueden ser la clase de un problema de clasificación, ect. Estos se pueden representar de diferentes formas.

No-supervisado: En este tipo de aprendizaje no se tienen etiquetas, si no solo se tienen los datos, y a partir de este empieza a inferir cual es el fenómeno que está ocurriendo, esto es necesario diseñar tecinas para identificar si los datos se parecen, por ejemplo, para esto se usan algoritmos de distancias y para ampliar un poco sobre esto pueden leer más sobre el algoritmo **K-means**.

En resumen, para este tipo de aprendizaje, solo tenemos datos.

Semi-supervisado: Este tipo de aprendizaje es, por ejemplo, cuando se tiene un data set que solo un porcentaje esta etiquetado así que usan técnicas de análisis de datos para aprovechar el data set al máximo, con la data que ya se tiene. Por ejemplo, se puede hacer análisis de distribución para ver si los datos que no están etiquetados pertenecen a la misma distribución o no.

Auto-supervisado: En este tipo de aprendizaje, por ejemplo, tenemos el concepto de autoencoders en este hay una imagen X y se trata de procesar esta imagen para producir un vector con 256 espacios, donde ese vector representara esa imagen, a esto se le conoce como espacio latente y es simplemente la imagen comprimida en un formato de vector, luego va a hacer la reconstrucción a partir de ese vector. De esta forma el autoencoder debería reconstruir la imagen y se obtiene el input como etiqueta y esto es lo que nos va a servir para supervisar el trabajo. Para ejemplificar lo anterior:

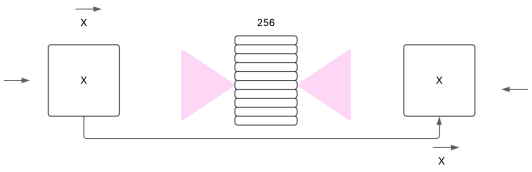

FIGURE 10. Ejemplo de aprendizaje auto-supervisado con autoencoders.

Refuerzo: En este es un tipo de aprendizaje se obtiene rewards de las acciones que se van haciendo, con cada acción que realiza el agente o la Inteligencia Artificial va obteniendo puntaje y va a tratar de obtener el mayor puntaje posible. Esta es la forma de indicar si la tarea se esta haciendo bien o se está haciendo mal.

Few-shot: Este tipo de aprendizaje es cuando se requiere unas pocas iteraciones para que realice bien las tareas. Por ejemplo, se le explica a la inteligencia artificial como sacar la raíz cuadrada de un número y por ejemplo se le pasan 10 ejemplos para que pueda iterar y a partir de este n cantidad de iteraciones la IA ya es capaz de iterar correctamente.

One-shot: Este tipo de aprendizaje es cuando con un intento le basta a la inteligencia artificial para cumplir con la tarea, por ejemplo, reconocer un rostro con solo una foto

Zero-shot: este tipo de aprendizaje es cuando le toma 0 intentos hacer bien la tarea que debe hacer, es decir no es entrenada, pero es capaz de hacer la tarea.

Machine Learning Pipeline

El proceso de desarrollo de modelos de Machine Learning sigue los siguientes pasos:

1) **Data adquisition:**: Es muy importante ya que se necesita saber de dónde se sacaron los datos o

si se necesita consentimiento, por ejemplo, todo esto es sentarse a analizar de donde se van a obtener los datos para resolver el problema.

2) **Data preparation:** : En esta fase hay que preparar los datos que se tienen para tener datos o un data set de calidad, por ejemplo, se eliminan datos duplicados o se descartan datos que no son de utilidad.
3) **Feature engineering:**: A partir de los datos que ya se consiguieron, se puede plantear hacer datos nuevos, por ejemplo, si hay datos tabulares, aquí se va a construir nuevos datos a partir de los anteriores o también se puede eliminar datos que no son útiles.
4) **Model selection:**: Aquí se va a seleccionar el modelo y se piensa en que modelo sirve para resolver el problema que se tiene. Se pueden aplicar muchas técnicas que no necesariamente serán siempre Machine Learning o Deeap Learning.
5) **Model training:**: En el proceso de entrenamiento del modelo, se van a escoger ciertos parámetros que van a ser mas lento o mas rápido que van a variar el resultado de las métricas.
6) **Model deployment**: En esta fase de deployment, se evalúa como se va a presentar.