Apuntes Semana 3- 04/03/2025

Elaborado por:
Jeremy Chacón Beckford-2021131338

***Abstract*—El presente documento recopila los apuntes de la tercera semana de clases del curso de inteligencia artificial. Se presentan conceptos claves sobre el aprendizaje automático, la importancia de la calidad de los datos, los diferentes tipos de aprendizaje y el proceso de desarrollo y despliegue de modelos de Machine Learning. .**

Machine learning

El campo del Machine Learning (ML) se centra en la creación de modelos capaces de aprender patrones a partir de datos sin ser explícitamente programados para cada tarea. En su aplicación, se generan métricas que permiten evaluar y comparar diferentes enfoques de modelado, desde métodos estadísticos tradicionales hasta redes neuronales profundas. En este contexto, surgen dos roles fundamentales:

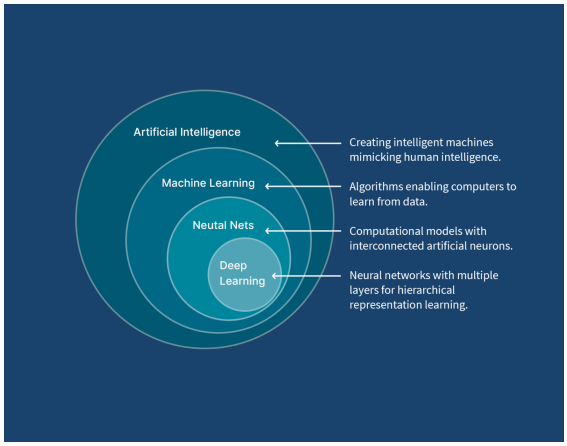

FIGURE 1. Machine Learning.

Dentro del Machine learning se cuenta con dos aristas: loa ciencia y la ingeniería.

Ciencia:

- **Generar conocimientos:** Para esto, a partir de artículos, papers o hipótesis, nosotros nos vamos a basar para poder construir más conocimiento o más ciencia. Desde aquí partimos el hecho de nuevos algoritmos, nuevas técnicas de optimización, como optimizar una función de manera adecuada, que no tome tantas iteraciones o tanto tiempo de entrenamiento, hacer ejecuciones, por ejemplo, en paralelo o con sistemas distribuidos, lo que fuera, que sea algo que nos pueda generar un poco más de conocimiento a partir de esto que ya tenemos.
- **Métricas:** se usan métricas para decidir qué modelo es mejor o peor, cuál me da mejores rendimientos, de acuerdo al problema que se esta atacando.
- **Data Scientist:** Se especializa en el análisis, limpieza y manipulación de datos para extraer conocimiento.
- **Research Scientist:**, enfocado en el desarrollo de nuevos modelos, algoritmos y teorías que mejoren las capacidades de ML.

Ingeniería:

Desde una perspectiva de ingeniería, el objetivo final no es solo entrenar modelos, sino desplegarlos en producción. Este proceso incluye la transformación del modelo a formatos más eficientes y compatibles con diferentes entornos. Tecnologías como ONNX permiten la interoperabilidad entre diferentes frameworks, mientras que MLOps introduce prácticas para la automatización y mantenimiento de modelos en producción.

Una vez implementados, los modelos requieren monitoreo constante para evaluar su rendimiento en en-

tornos reales. Problemas como el Shifted Data, donde la distribución de los datos cambia con el tiempo, pueden degradar la precisión del modelo, lo que hace fundamental una estrategia de revisión continua.

Tipos de aprendizaje

El aprendizaje automático puede clasificarse en diversas categorías según la manera en que el modelo adquiere conocimiento:

- **Supervisado:** Los datos de entrenamiento incluyen etiquetas que guían el aprendizaje del modelo, un ejemplo de esto es la clasificación de imágenes.
- **No supervisado:** No se proporcionan etiquetas y el modelo identifica patrones ocultos en los datos, un ejemplo de esto es el Clustering.
- **Semi-Supervisado:**Mezcla de datos etiquetados y no etiquetados, útil cuando etiquetar datos es costoso.
- **Auto-Supervisado:**El mismo input es la etiqueta, utilizado en autoencoders y modelos de representación, básicamente reducen el tamaño de su vector.
- **Aprendizaje por refuerzo:**: Basado en un sistema de recompensas, donde un agente aprende a maximizar su desempeño a través de la interacción con un entorno.
- **Few-Shot Learning:** El modelo necesita varios ejemplos para lograr aprender una tarea específica.
- **One-Shot Learning:**Se le tiene que enseñar almenos una vez al algoritmo como realizar la tarea.
- **Zero-Shot Learning:** Capacidad de un modelo para realizar tareas sin haber sido específicamente entrenado para ellas.

Pipeline de Machine Learning

El proceso de construcción de un modelo de Machine Learning se compone de varias etapas:

- **Data adquisition:** La calidad de los datos determina el rendimiento del modelo. Es esencial asegurarse de que los datos sean representativos y libres de inconsistencias como valores faltantes o duplicados. Ejemplo: Un modelo entrenado con datos de chips AMD puede fallar al aplicarse a chips Intel debido a diferencias en sus características.
- **Data preparation:**
  - **Normalización:** Ajustar los datos para que tengan media cero.
  - **Escalado:** Uniformizar valores en una escala específica (por ejemplo, entre 0 y 1).
  - **Tratamiento de valores faltantes:** Sustitución por la media, eliminación o inferencia basada en otros datos.
- **Feature Engineering:**El feature engineering consiste en seleccionar o generar nuevas características relevantes para el modelo, asegurando que cada columna aporte información importante para resolver el problema. Dependiendo del tipo de dato, se aplican diferentes técnicas para optimizar su representación. Además, se realiza un análisis de correlación entre las distintas features con el objetivo de eliminar variables redundantes y mejorar la eficiencia del modelo.
- **Model selection:**Se elige el modelo más adecuado según el problema y los requisitos computacionales. Modelos simples como la regresión logística pueden ser suficientes para tareas básicas, mientras que problemas más complejos pueden requerir redes neuronales profundas.
- **Model training:**
  - División de datos en training set y validation set.
  - Ajuste de hiperparámetros mediante técnicas como Grid Search o Bayesian optimization.
- **Model deployment** Se implementa el modelo en producción como API o embebido en un sistema, asegurando su monitoreo y mantenimiento continuo.

Hiperparámetros y optimización

Los hiperparámetros son valores ajustados manualmente para optimizar el rendimiento del modelo. Algunos ejemplos incluyen:

- **Tasa de Aprendizaje (Learning Rate):** Controla la velocidad de aprendizaje.
- **Número de Epochs:** Número de veces que el modelo recorre los datos de entrenamiento.
- **Tamaño del Batch:**Cantidad de datos procesados en cada iteración.
- **Número de Vecinos en KNN:**Define cuántos vecinos se consideran en la clasificación.

Técnicas de ajuste:

- **Grid Search:** Explora combinaciones de hiperparámetros en un rango específico.
- **Random Search:**Esta técnica toma muestras aleatorias de una cuadrícula de hiperparámetros, en lugar de evaluar sistemáticamente todas las combinaciones. Para ello, se utiliza una función de número

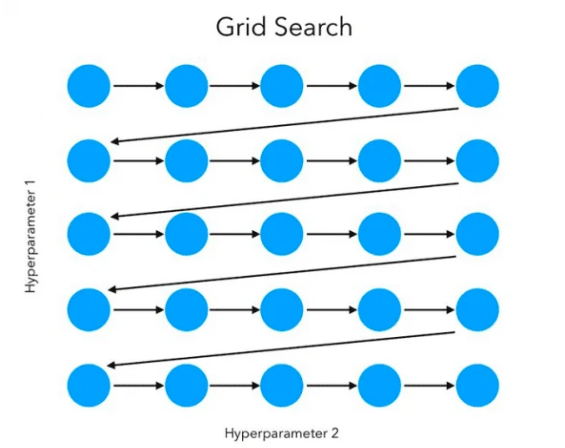

FIGURE 2. Grid Search.

entero aleatorio que selecciona aleatoriamente un valor dentro de cada parámetro de la cuadrícula, reduciendo así el tiempo .

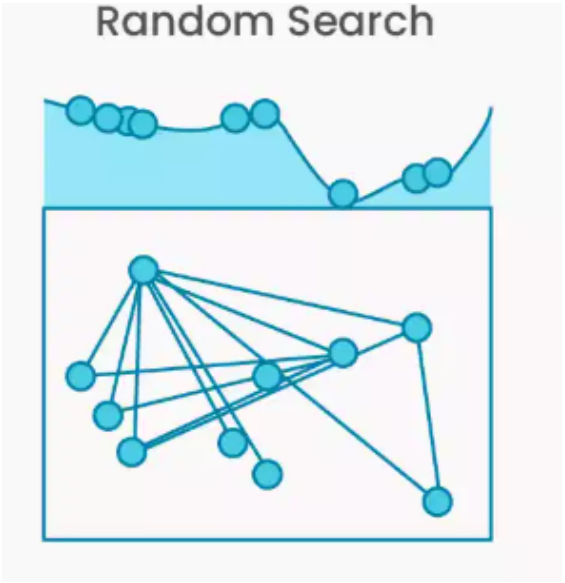

FIGURE 3. Random Search.

- **Bayesian Optimization:** Ajusta hiperparámetros basándose en modelos probabilísticos.

Feature

Es una propiedad o atributo medible de una entidad, la mayoría de casos son representadas de forma numérica para procesados por algoritmos, un ejemplo de esto es la altura de una casa o el peso de un individuo.

- **Feature vector:** Un vector de **D** Dimensiones donde cada dimensión representa una caracteristica de un objeto
- **Label:** Refuerzo del feature vector información que se desea predecir
- **Dataset:** Coleción de instancias de tamaño N.

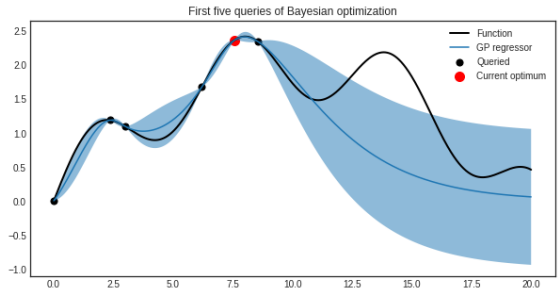

FIGURE 4. Bayesian Optimization.

Resumen de notebook vectores:

Conceptos:

- Vector: Un vector es un arreglo N dimensional que tiene tanto dirección como magnitud asociado, se compone de un punto de origen y un punto final
- Magnitud de vector o norma: Hace referencia de la distancia de origen desde un punto A a un punto B.
- La distancia Manhattan entre dos puntos $A = (x_1, y_1, z_1, \ldots, n_1)$ y $B = (x_2, y_2, z_2, \ldots, n_2)$ en un espacio n-dimensional se calcula mediante la fórmula:

$$\sum_{i=1}^{n} |x_i - y_i|$$

- Vector unidad
  - Es un vector que tiene una longitud o magnitud 1.
  - Se utiliza generalmente para indicar la dirección de un vector sin considerar su longitud.
  - Básicamente ayuda a simplificar cálculos porque son más simples.

  Se puede obtener el vector unidad de un vector $\mathbf{u}$ dividiendo el vector $\mathbf{u}$ entre su norma.
  El vector unitario $\hat{\mathbf{u}}$ de un vector $\mathbf{u}$ se calcula como:

$$\hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$$

- Producto punto: también conocido como producto escalar o producto interno, es una operación entre dos vectores en el cual su resultado es un escalar.
- Propiedad del producto punto: la identidad del coseno para el producto punto entre dos vectores

$\mathbf{u}$ y $\mathbf{v}$ en función de sus magnitudes y el ángulo θ es:

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \cdot \|\mathbf{v}\| \cdot \cos(\theta)$$

Donde $\|\mathbf{u}\|$ y $\|\mathbf{v}\|$ son las magnitudes de los vectores $\mathbf{u}$ y $\mathbf{v}$, respectivamente, y θ es el ángulo entre los vectores.

- Vector co-direccional: dos vectores son co-direccionales si tienen la misma dirección aunque tengan diferente magnitud. En otras palabras, si existe un escalar K, no igual a cero, tal que uno de los vectores es producto del escalar por el vector:

$$\mathbf{u} = K \cdot \mathbf{v}$$

- Vectores ortogonales: dos vectores son ortogonales si los vectores son perpendiculares, es decir, tienen un ángulo de 90 grados entre ellos.

Notas importantes:

- Se recomienda descargar anaconda el cual es un gestor de ambientes.
- Instalar Jupyter
- Se hizo un repaso de conceptos básicos de álgebra lineal que serán subidos mediante notebooks a Tec digital.

Tarea 01 enunciado

Mencione un aporte de:

- Yann LeCun
- Yoshua Bengio
- Sam Altman
- Geoffrey Hinton
- Timnit Gebru
- Ian Goodfellow

- Incluya un resumen del funcionamiento de las siguientes herramientas:
  - Onnx
  - MLFlow
  - Vertex
  - N8N

Fecha de entrega martes 11 de marzo.