



**Universidad
Tecnológica
del Perú**

UNIVERSIDAD TECNOLÓGICA DEL PERÚ

CURSO:
**INNOVACIÓN Y TRANSFORMACIÓN
DIGITAL**

SECCIÓN:

34886

DOCENTE:

PALOMINO IRIARTE, Edwin

Integrantes:

ANTONIO HUAMANCHUMO, Jose	U21315042
GOSE BENAVENTE, Jhonny	U18301666
MALGAREJO CUENCIA Yaren	U21218077
MUÑOZ TASAYCO, Jesus	U21220711
VIVAR OCAÑA, Bryan	U20213725

**Lima, Perú
2023**

INDICE

INTRODUCCIÓN	1
CONTENIDO.....	2
I. Algoritmos de aprendizaje automático	2
Tipos de Algoritmos:.....	2
Algoritmos populares:.....	2
Aplicaciones en Campos Específicos:	2
Algoritmos de Búsqueda y Optimización:	3
II. Algoritmos de búsqueda y optimización:.....	3
¿Qué son los algoritmos de búsqueda?	3
¿Qué son los Algoritmos de Optimización?	7
III. Procesamiento de lenguaje natural	8
1. Tokenización	8
2. La lematización	8
3. Etiquetado Gramatical (POS tagging)	8
4. Algoritmos de clasificación	9
5. Algoritmos de agrupamiento	9
6. Modelos estadísticos del lenguaje.....	10
7. Visión Artificial	11

INTRODUCCIÓN

Un algoritmo es un conjunto ordenado y finito de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas, los algoritmos son utilizados en informática para procesar datos y automatizar soluciones.

Por otro lado, los sistemas inteligentes son sistemas software y hardware que imitan las capacidades cognitivas humanas para resolver problemas complejos. Entre estas capacidades se encuentran el aprendizaje automático, el razonamiento automático y la autocorrección. Algunos ejemplos de sistemas inteligentes son los asistentes virtuales, los vehículos autónomos y los sistemas de recomendación.

Los algoritmos son esenciales en el funcionamiento de los sistemas inteligentes, ya que les permiten aprender de datos, percibir el entorno, tomar decisiones racionales y mejorar con la experiencia. Sin algoritmos, los sistemas inteligentes serían incapaces de presentar un comportamiento que imite la inteligencia humana.

Algunas de las tareas más importantes que realizan los algoritmos en los sistemas inteligentes modernos son: clasificación de datos, agrupamiento de datos, procesamiento de lenguaje natural, reconocimiento de patrones, optimización de procesos y aprendizaje automático. Los algoritmos más populares incluyen redes neuronales, árboles de decisión, algoritmos genéticos, máquinas de vectores de soporte, entre muchos otros.

En esta monografía se analizarán algunos de los algoritmos más relevantes que sustentan el funcionamiento de diversos tipos de sistemas inteligentes en la actualidad, se estudiarán sus principios básicos, sus aplicaciones y ejemplos de uso en inteligencia artificial.

Entre los temas centrales que cubriré se encuentran: algoritmos de aprendizaje automático, algoritmos de procesamiento de lenguaje natural, algoritmos de visión artificial, algoritmos de búsqueda y optimización. Asimismo, se discutirá la importancia de estos algoritmos en el desarrollo de aplicaciones inteligentes.

CODIGO ALGORITMO:

<https://colab.research.google.com/drive/1GiQI2rWMM1DCDVZjaJuoLt1SZadkeEB4?usp=sharing>

CODIGO CRUD BANCARIO:

<https://colab.research.google.com/drive/1hGFFN7pg8fx5rVrF6wLodSualigJmBJt?usp=sharing>

CONTENIDO

I. Algoritmos de aprendizaje automático

Tipos de Algoritmos:

- ✓ Supervisados: Utilizan un conjunto de datos etiquetado para entrenar al algoritmo, donde se conocen las entradas y las salidas correspondientes. Ejemplo: Regresión Lineal y Logística, Máquinas de Soporte Vectorial (SVM), Redes neuronales.
- ✓ No supervisados: Trabajan con datos no etiquetados, buscando patrones y estructuras inherentes en los datos. Ejemplos: K-Means para agrupamiento, Análisis de Componentes Principales (PCA) para reducción de dimensionalidad, Algoritmos de asociación para descubrir relaciones entre variables.
- ✓ Por Refuerzo: El algoritmo aprende a través de interacción con un entorno, tomando decisiones y recibiendo retroalimentación en forma de recompensas o penalizaciones. Ejemplos: Q-Learning para aprendizaje por refuerzo, Algoritmos Genéticos para evolución y optimización, Algoritmos Populares.

Algoritmos populares:

Aplicaciones en Campos Específicos:

1. Visión Artificial:

- Redes Neuronales Convolucionales (CNN) para detección de objetos.
- Segmentación semántica para identificar áreas específicas en imágenes.

2. Procesamiento de Lenguaje Natural (PLN):

- Modelos de Lenguaje como BERT y GPT para comprensión del lenguaje y generación de texto.
- Redes Neuronales Recurrentes para secuencias de texto.

3. Robótica:

- Algoritmos de Planificación de Trayectorias para movimientos robóticos precisos.
- Aprendizaje por Refuerzo para entrenar robots en entornos dinámicos.

Algoritmos de Búsqueda y Optimización:

- Algoritmos de Búsqueda: Utilizados para encontrar soluciones a un problema en un espacio de búsqueda. **Ejemplo:** Búsqueda en Profundidad (DFS) y Búsqueda en Amplitud (BFS) para explorar grafos.
- Algoritmos de Optimización: Buscan la mejor solución en un conjunto de posibles soluciones. Ejemplos: Algoritmos Genéticos para la optimización evolutiva, Optimización por Enjambre de Partículas para encontrar el óptimo global.

II. Algoritmos de búsqueda y optimización:

¿Qué son los algoritmos de búsqueda?

Un algoritmo de búsqueda, en el ámbito de la informática, es una herramienta diseñada para ubicar un elemento específico con ciertas propiedades dentro de una estructura de datos. Su aplicación abarca desde localizar registros en una base de datos hasta determinar el mejor movimiento en una partida de ajedrez. En su forma más básica, este problema se simplifica a la búsqueda de un número en un vector, donde el algoritmo sigue un conjunto ordenado de pasos lógicos para encontrar la posición o información deseada. Estos algoritmos son fundamentales en la resolución eficiente de problemas prácticos en la programación y la toma de decisiones computacionales, abarcando desde la gestión de datos hasta la optimización en situaciones estratégicas. Algunos ejemplos pueden ser los de:

A. Búsqueda Secuencial:

La búsqueda secuencial, también denominada búsqueda lineal, se aplica cuando se trabaja con un vector no ordenado o cuando no es factible realizar una ordenación previa. Este algoritmo aborda la tarea de localizar un elemento particular mediante una comparación secuencial con cada elemento del arreglo,

avanzando uno por uno, hasta que se encuentra el elemento buscado o se llega al final del conjunto de datos. Aunque este método garantiza la identificación del elemento en caso de existir, no proporciona la certeza de su no existencia hasta que se haya revisado exhaustivamente todo el arreglo.

Consideremos una situación en la que se maneja una colección de registros organizados en una estructura de lista lineal. La implementación básica del algoritmo de búsqueda secuencial implica iniciar desde el principio de la lista y examinar cada registro de manera sucesiva hasta hallar la llave indicada (k) o alcanzar el final de la lista. La situación más eficiente se da cuando el registro buscado es el primero examinado, mientras que el peor escenario implica comparar la llave (k) con todas las llaves de los n registros presentes en la colección. En términos promedio, se realizan aproximadamente $n/2$ comparaciones.

Aunque la búsqueda secuencial se caracteriza por su relativa lentitud, constituye el método preferido cuando los datos no siguen un patrón ordenado predefinido. Esta estrategia se torna esencial en situaciones en las que la información carece de una estructura predictiva. Adicionalmente, si las llaves no son únicas, resulta necesario realizar una búsqueda en toda la lista para identificar todos los registros asociados con una llave específica. En resumen, la búsqueda secuencial, a pesar de su eficiencia relativa, se convierte en una herramienta indispensable en contextos donde la organización de datos no sigue un patrón de ordenación preexistente.

B. Búsqueda Binaria:

El algoritmo de búsqueda binaria, también conocido como búsqueda dicotómica, se emplea en vectores ordenados para mejorar la eficiencia de la búsqueda. Su enfoque se asemeja a la búsqueda en un diccionario, donde la intuición es evitar revisar secuencialmente cada elemento. En lugar de eso, se inicia abriendo el diccionario por la mitad y se descarta una mitad basándose en si la letra inicial es mayor o menor que la buscada. Este proceso se repite hasta localizar el elemento deseado.

El pseudocódigo del algoritmo de búsqueda binaria implica el mantenimiento de índices que definen los límites superior e inferior de la porción del vector que se está analizando. Se trata de una estrategia eficiente que aprovecha la propiedad de orden del vector para reducir la complejidad de búsqueda. En resumen, la búsqueda binaria destaca por su capacidad para encontrar rápidamente elementos en conjuntos de datos ordenados mediante iterativas divisiones y descartes, proporcionando así un enfoque optimizado para la búsqueda en vectores ordenados.

C. Fuerza Bruta:

Los algoritmos de fuerza bruta resuelven un problema explorando todas las posibles soluciones de manera exhaustiva, sin utilizar estrategias más sofisticadas o heurísticas que podrían reducir el número de operaciones. Esta estrategia es simple y directa, pero puede volverse ineficiente, especialmente cuando el espacio de búsqueda es grande.

La fuerza bruta considera cada opción posible y verifica si cumple con los requisitos del problema. Aunque garantiza encontrar la solución correcta, puede requerir un tiempo significativo, especialmente cuando el conjunto de soluciones posibles es extenso. En contraste, algunos algoritmos más avanzados, como los algoritmos de búsqueda y optimización que mencionamos anteriormente, buscan reducir el espacio de búsqueda y encontrar soluciones de manera más rápida y eficiente.

La elección entre un enfoque de fuerza bruta y otros métodos depende de la complejidad del problema y de la eficiencia necesaria para resolverlo. En muchos casos, se busca equilibrar la simplicidad del algoritmo con la eficiencia en términos de tiempo y recursos computacionales.

D. Divide y reinarás:

Divide y reinarás (Divide and Conquer) es una estrategia de diseño de algoritmos que divide un problema en subproblemas más pequeños, resolviéndolos de manera recursiva. Esta división continúa hasta que los

subproblemas sean lo suficientemente simples para resolverse directamente. Luego, las soluciones se combinan para obtener la solución completa. Ejemplos incluyen algoritmos de búsqueda binaria, ordenación como Merge Sort y QuickSort, y el algoritmo de multiplicación de matrices de Strassen. Esta técnica mejora la eficiencia al abordar problemas complejos de manera modular y estructurada.

E. Búsqueda Hashing:

Cuando el tiempo requerido para ordenar un arreglo o buscar un elemento se asemeja al de una búsqueda secuencial en un arreglo desordenado, surge el Hashing como una solución alternativa eficiente. La característica distintiva del Hashing es asignar a cada elemento una posición específica determinada por su módulo, lo que facilita las operaciones básicas como ingresar, eliminar y buscar elementos.

Existen dos enfoques principales de Hashing:

a) Hashing Cerrado:

- Es una estructura de datos estática que utiliza un tamaño fijo para el almacenamiento.
- La limitación en el tamaño del conjunto puede ser una desventaja, ya que podría no adaptarse bien a conjuntos de datos cambiantes o en crecimiento.
- A pesar de esto, proporciona un acceso rápido y eficiente para conjuntos de tamaño constante.

b) Hashing Abierto:

- Es una estructura de datos dinámica que no impone un límite al tamaño del conjunto.
- Ofrece mayor flexibilidad para conjuntos de datos cambiantes y en expansión.

- Sin embargo, puede requerir manejo adicional para abordar colisiones, situaciones donde dos elementos se asignan a la misma posición.

En el Hashing, el cálculo de la posición de un elemento se realiza mediante operaciones de módulo, lo que simplifica y agiliza las búsquedas. Esta técnica se destaca por su eficiencia en comparación con otros métodos cuando se trata de realizar operaciones básicas en conjuntos de datos. La elección entre Hashing Cerrado y Hashing Abierto depende de las características del conjunto de datos y los requisitos específicos del problema. En conjunto, el Hashing ofrece una alternativa valiosa y eficiente para abordar las operaciones fundamentales en la manipulación de datos.

¿Qué son los Algoritmos de Optimización?

Los algoritmos de optimización representan una piedra angular en la ciencia de la computación y las matemáticas aplicadas, con la finalidad de descubrir la solución más efectiva para problemas específicos. Estos algoritmos están diseñados para maximizar o minimizar funciones objetivo, considerando restricciones particulares, y su aplicabilidad abarca disciplinas diversas como logística, ingeniería, economía y aprendizaje automático. Estos procedimientos no solo buscan eficiencia, sino que también toman en cuenta restricciones, proporcionando herramientas fundamentales para la toma de decisiones en planificación estratégica, diseño de sistemas y análisis económico. Su aplicación abarca desde la optimización de rutas logísticas hasta la mejora de la eficiencia operativa en sistemas complejos, y su papel crítico se extiende al diseño de modelos en el aprendizaje automático y la toma de decisiones económicas estratégicas.

A. Optimización Lineal:

B. Optimización no Lineal:

C. Optimización multiobjetivo:

D. Cuáles son las tendencias y avances:

III. Procesamiento de lenguaje natural

1. Tokenización

La tokenización es el proceso de dividir un texto en unidades más pequeñas llamadas "tokens". Estos tokens pueden ser palabras, frases, o incluso caracteres, dependiendo de la granularidad deseada. La tokenización es el primer paso crucial en el procesamiento del lenguaje natural. Divide el texto en unidades que pueden ser procesadas más fácilmente y proporciona la base para análisis lingüísticos más profundos.

2. La lematización

La lematización es el proceso de reducir las palabras a sus formas base o lemas. Un lema es la forma canónica de una palabra que representa su significado central. La lematización es esencial para normalizar las palabras y reducir la dimensionalidad del vocabulario. Ayuda a capturar la esencia semántica de las palabras, facilitando tareas como la búsqueda, clasificación y análisis de sentimientos.

3. Etiquetado Gramatical (POS tagging)

El etiquetado gramatical asigna a cada token en un texto una etiqueta que indica su categoría gramatical, como sustantivo, verbo, adjetivo, etc. El etiquetado gramatical es crucial para comprender la estructura gramatical de una oración. Ayuda a identificar las relaciones entre las palabras y permite un análisis más profundo del significado. Es fundamental en tareas como el análisis sintáctico y la extracción de información.

4. Algoritmos de clasificación

4.1 Regresión Logística:

Se utiliza para problemas de clasificación binaria. Establece una relación logarítmica entre las variables de entrada y la probabilidad de pertenencia a una clase.

4.2 Máquinas de Soporte Vectorial (SVM)

Óptimo para problemas de clasificación binaria y también puede extenderse a problemas de clasificación multiclase. Busca encontrar el hiperplano que mejor separa las clases en el espacio de características.

4.3 Árboles de Decisión

Efectivos para problemas de clasificación y regresión. Representan decisiones en forma de un árbol, dividiendo el conjunto de datos en base a las características más informativas.

4.4 K Vecinos más Cercanos (K-NN)

Clasificación basada en la mayoría de votos de los "k" vecinos más cercanos en el espacio de características. Funciona bien para datos con patrones locales.

5. Algoritmos de agrupamiento

5.1 K-Means:

Agrupar datos en "k" clusters basándose en la similitud entre observaciones. Los centroides de los clusters representan los puntos medios de los grupos.

5.2 Agrupamiento Jerárquico

Construye una jerarquía de clusters en la que los clusters más pequeños están anidados en clusters más grandes. Puede ser aglomerativo (comienza con clusters individuales) o divisivo (comienza con un cluster único).

5.3 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Identifica clusters basándose en la densidad de puntos en el espacio de características. Puede identificar clusters de formas arbitrarias y es robusto frente a ruido.

5.4 Mean Shift

Encuentra los modos de densidad de un conjunto de datos. Es particularmente útil cuando los clusters tienen formas no esféricas y pueden variar en tamaño.

6. Modelos estadísticos del lenguaje

Los modelos estadísticos del lenguaje son herramientas fundamentales en el procesamiento del lenguaje natural. Dos tipos principales son:

6.1 Modelos de Lenguaje N-gram:

- ✓ Utilizan estadísticas y probabilidades.
- ✓ Aproximan la probabilidad de una secuencia de palabras basándose en la probabilidad condicional de ocurrencia de palabras individuales dado un contexto anterior.
- ✓ Se dividen en unigramas, bigramas, trigramas, etc.

6.2 Modelos de Lenguaje Basados en Máquinas de Aprendizaje:

- ✓ Utilizan algoritmos de aprendizaje automático.
- ✓ Pueden ser N-gram con suavizado para manejar secuencias no observadas.
- ✓ Incluyen modelos neuronales (como GPT y BERT), HMM (Modelos Ocultos de Markov), entre otros.

7. Visión Artificial

7.1 Detección y reconocimiento de objetos

La detección de objetos es una rama de la visión artificial que se enfoca en detectar la presencia y ubicación de objetos en imágenes. Se utiliza en una amplia gama de aplicaciones, desde sistemas de vigilancia hasta análisis de imágenes médicas y conducción autónoma.

¿Cómo funciona la detección de objetos?

Partimos de una red convolucional (CNN), que nos permite dada una imagen obtener una clasificación de la misma; mediante un algoritmo, buscamos regiones que pueden ser de interés, por ejemplo, zonas contiguas de tonos parecidos y bordes, para introducirlas en nuestra CNN y determinar su clasificación. Finalmente, podemos determinar cuáles de las regiones contiene a nuestro objeto y poder indicar su posición.

Si bien esta forma de detectar objetos parece sencilla y eficaz, plantea un problema relacionado con las anteriormente mencionadas regiones de interés. ¿Qué ocurre si distintas áreas solapan al mismo objeto?

Para solucionar esto, se utilizan en conjunto las técnicas de IoU y NMS:

- **Intersection over Union (IoU)**: Calcula la relación entre el área de intersección de dos regiones y el área de unión de las mismas
- **Non-maximum Suppression (NMS)**: Elimina detecciones superpuestas, manteniendo solo la detección con la puntuación más alta.

7.2 Seguimiento de Objetos

El seguimiento es más rápido que la detección. Mientras que el clasificador previamente entrenado necesita detectar un objeto en cada fotograma del video para utilizar un rastreador de objetos especificamos el cuadro delimitador de un objeto una vez, el proceso de seguimiento es más rápido. El seguimiento es más estable. En los casos en que el objeto rastreado se superpone parcialmente con otro objeto, el algoritmo de detección puede "perderlo", mientras que los algoritmos de seguimiento son más robustos a la oclusión parcial. El seguimiento proporciona más información. Si no estamos interesados en la pertenencia de un objeto a una clase

específica, el algoritmo de seguimiento nos permite rastrear la trayectoria de movimiento de un objeto específico, mientras que el algoritmo de detección no puede.

7.3 Clasificación y segmentación semántica

La segmentación semántica es un algoritmo de aprendizaje profundo que asocia una etiqueta o categoría con cada píxel de una imagen. Se utiliza para identificar un grupo de píxeles que conforman diferentes categorías. Por ejemplo, los vehículos autónomos deben reconocer vehículos, peatones, señales de tráfico, aceras y otras características de la carretera. La segmentación semántica se utiliza en muchas aplicaciones, como la conducción autónoma, las imágenes médicas y la inspección industrial.

El proceso de entrenamiento de una red de segmentación semántica para clasificar imágenes consta de 4 pasos:

- Analizar conjuntos de imágenes con píxeles etiquetados
- Crear una red de segmentación semántica
- Entrenar la red para clasificar imágenes en categorías de píxeles
- Evaluar precisión de la red

7.4 Reconstrucción 3D realidad aumentada

La Realidad Aumentada es un área de interés creciente dado el gran número de aplicaciones potenciales. El reto principal consiste en ampliar la información que nos ofrece el mundo real con objetos virtuales y datos que coexistan en el mismo espacio y nos permitan mejorar la percepción que el usuario tiene de su entorno. Partimos de la hipótesis de que los sistemas visuales artificiales y, en particular, aquellos basados en visión estereoscópica o 3D pueden ayudarnos de manera importante a reconstruir la escena, detectar los objetos que en ella se encuentran, con el objetivo de etiquetar la realidad, añadir información de su estructura, disposición, color o identificación, e introducir objetos virtuales. De hecho, la visión estereoscópica es uno de los mecanismos que utilizan los seres vivos para calcular distancias a los objetos, evitar obstáculos, identificar elementos, etc.

Objetivos:

- Construcción de un mapa de profundidades a partir del mapa de disparidad proveniente del sistema de visión estereoscópica.
- Reconstrucción 3D del entorno a partir del mapa de profundidades, aplicando un esquema de enumeración de la ocupación espacial.
- Corrección de la reconstrucción aplicando los conceptos de la geometría perspectiva, coherencia espacial y otras restricciones geométricas.
- Integración de la información 3D en un sistema de realidad aumentada.
- Generación de objetos virtuales e inclusión en la imagen real, resolviendo el problema de la alineación de los elementos reales y virtuales.

