Moisés Cantón Jara

Fundamentos de Matemáticas

Álgebra Lineal

Álgebra Lineal

1. Introducción al Álgebra Lineal

Álgebra Lineal es una rama de las matemáticas que se concentra en estudiar vectores, matrices y sistemas lineales desempeñando un papel central en el desarrollo y comprensión de los algoritmos en Machine Learning (ML) y la Inteligencia Artificial (IA).

En primer lugar, nos ofrece una estructura para representar datos, lo cual es muy útil en ML e IA donde es común trabajar con grandes conjuntos de datos. Estos datos se pueden representar de manera estructurada como vectores y matrices, por ejemplo, una imagen se puede representar por una o un conjunto de matrices donde cada entrada corresponde al valor de un píxel.

En segundo lugar, las operaciones fundamentales entre las estructuras de datos son fundamentales para la implementación de algoritmos de ML. Las redes neuronales, que son los pilares del Deep Learning (DL), dependen en gran medida de estas operaciones para el forward propagation y el backpropagation durante el entrenamiento.

En tercer lugar, las técnicas de optimización, que son cruciales para el entrenamiento y ajuste de modelos de IA, se apoyan en la comprensión de espacios vectoriales, proyecciones y transformaciones lineales.

2. Representación de Datos

A la hora de representar datos debemos considerar los diferentes constructos matemáticos que nos permiten describir magnitudes: “escalares”, “vectores”, “matrices” y “tensores”.

- Un escalar es una entidad matemática que se utiliza para representar una magnitud o cantidad en un espacio unidimensional. Un escalar se representa como números reales no tiene dirección ni orientación; simplemente describe cuánto de una cantidad específica tenemos.
  - Ejemplos de escalares incluyen valores numéricos como 5 metros, 30 grados Celsius o 70 kilogramos.

- Un vector es una entidad matemática que posee tanto magnitud como dirección. A diferencia de los escalares, los vectores representan cantidades que tienen una orientación específica en el espacio multidimensional. Se representan gráficamente mediante flechas, donde la longitud de la flecha indica la magnitud y la dirección de la flecha representa la dirección del vector.
  - Ejemplos comunes de vectores incluyen la velocidad de un automóvil, una fuerza aplicada en una dirección específica o las coordenadas espaciales de un punto en un sistema de coordenadas.

- Una matriz es una estructura matemática que consiste en una tabla rectangular de números organizados en filas y columnas. Las matrices se utilizan para representar datos y transformaciones lineales en álgebra lineal, y se denotan comúnmente con letras mayúsculas, como A, y sus elementos se identifican mediante subíndices, como Aij para el elemento en la fila i y la columna j.
  - Son fundamentales en la resolución de sistemas de ecuaciones, la representación de imágenes, ...

- Un tensor es una generalización de los conceptos de escalares, vectores y matrices, que puede representar datos multidimensionales en un espacio n-dimensional. En esencia, un tensor es un arreglo de números con más de dos ejes o dimensiones. Por ejemplo, un escalar se considera un tensor de orden 0, un vector un tensor de orden 1 y una matriz un tensor de orden 2.
  - Los tensores son esenciales en áreas avanzadas de matemáticas y ciencia de datos, especialmente en el aprendizaje profundo de redes neuronales, donde se manipulan tensores de alta dimensión para procesar datos complejos, como imágenes y texto.

3. Vectores y Matrices[iii]

Vectores

Un vector es una lista ordenada de números, que se conocen como componentes del vector y pueden representar diversas magnitudes en distintos contextos, desde simples listas de números hasta coordenadas en un espacio n-dimensional pasando, por ejemplo, por el número de habitaciones, el tamaño en metros cuadrados y el precio en un conjunto de datos sobre casas.[iii]

python:
```
import numpy as np

vector = np.array([1, 2, 3])
print(vector)
```

Matrices

Una matriz es un arreglo bidimensional de números, donde cada número se identifica por dos índices en lugar de uno, como en los vectores, y son útiles para representar sistemas de ecuaciones, transformaciones lineales, conjuntos de datos con múltiples instancias y características, …[iv]

python:
```
import numpy as np

matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print(matrix)
```

Operaciones

Al igual que para los escalares, estas estructuras matemáticas poseen operaciones básicas que nos ofrecen la capacidad de transformar, analizar y extraer información de estos datos de manera eficiente.[v] A continuación, describiremos aquellas para vectores:

- Suma

La suma de dos vectores se realiza componente a componente. Si tienes dos vectores v y w en un espacio n-dimensional, la suma $v + w$ se calcula tomando la suma de las correspondientes componentes en cada dimensión. Es decir, si $v = [v_1, v_2, \dots, v_n]$ y $w = [w_1, w_2, \dots, w_n]$, entonces la suma $v + w$ se define como $[v_1 + w_1, v_2 + w_2, \dots, v_n + w_n]$.

- Resta

La resta de dos vectores también se realiza componente a componente. Si tienes dos vectores v y w, la resta v−w se calcula restando las correspondientes componentes en cada dimensión. En otras palabras, si $v = [v_1, v_2, \dots, v_n]$ y $w = [w_1, w_2, \dots, w_n]$, entonces la suma $v - w$ se define como $[v_1 - w_1, v_2 - w_2, \dots, v_n - w_n]$.

- Multiplicación Escalar

La multiplicación de un vector por un escalar (un número) se realiza multiplicando cada componente del vector por ese escalar. Por ejemplo, si tienes un vector $v = [v_1, v_2, \dots, v_n]$ y un escalar k, la multiplicación escalar kv se calcula como $[kv_1, kv_2, \dots, kv_n]$.

- Producto Escalar

El producto punto, también conocido como producto escalar o producto interno, es una operación entre dos vectores que produce un escalar (un número real). Se denota como $v \cdot w$ o $\langle v, w \rangle$, donde v y w son los vectores involucrados.

La fórmula para el producto punto entre dos vectores v y w de igual dimensión n es: $v \cdot w = v_1 w_1 + v_2 w_2 + \cdots + v_n w_n$

El resultado del producto punto es un número real, y su valor depende de los ángulos y las magnitudes de los vectores. Si el resultado es cero, significa que los vectores son ortogonales (perpendiculares).

El producto punto satisface la propiedad conmutativa $(v \cdot w = w \cdot v)$ y la propiedad distributiva sobre la suma de vectores.

- Producto Vectorial

El producto cruz, también conocido como producto vectorial, es una operación entre dos vectores que produce un tercer vector que es perpendicular al plano definido por los dos vectores originales. El producto cruz se denota como $v \times w$.

La fórmula para el producto cruz entre dos vectores tridimensionales v y w es: $v \times w = [v2w3 - v3w2, v3w1 - v1w3, v1w2 - v2w1]$

El resultado del producto cruz es un vector que es perpendicular al plano formado por v y w. La magnitud de este vector resultante está relacionada con el área del paralelogramo definido por los dos vectores originales.

El producto cruz no es conmutativo $(v \times w \neq w \times v)$ y no satisface la propiedad conmutativa ni la propiedad distributiva. Sin embargo, sigue otras propiedades útiles relacionadas con la orientación y la magnitud del vector resultante.

- Norma

La norma de un vector es una medida de esa magnitud a través del cálculo de la longitud en un espacio euclidiano.

Normalmente se denota como $||v||$ y se calcula utilizando el teorema de Pitágoras para espacio bidimensionales o mediante una generalización para espacios de mayor dimensión, por ejemplo, dado un vector en dos dimensiones $v = [x, y]$, la norma se calcula como $||v|| = \sqrt{x^2 + y^2}$, mientras que para espacios de n-dimensiones tendremos: $||v|| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$.

Igualmente, para matrices tenemos:

- Suma

La suma de dos matrices se realiza sumando las correspondientes entradas de ambas matrices. Para que la suma sea posible, las matrices deben tener la misma dimensión, es decir, el mismo número de filas y columnas. La suma de matrices es conmutativa y asociativa.

- Resta

La resta de dos matrices se realiza restando las correspondientes entradas de la segunda matriz de las de la primera matriz. Al igual que con la suma, las matrices deben tener la misma dimensión para que la resta sea posible.

- Multiplicación Escalar

La multiplicación de una matriz por un escalar (un número) se realiza multiplicando cada entrada de la matriz por ese escalar. La matriz resultante tiene las mismas dimensiones que la matriz original.

- Multiplicación de Matrices

La multiplicación de matrices es una operación más compleja que la suma y la multiplicación escalar. La multiplicación de dos matrices se realiza multiplicando filas de la primera matriz por columnas de la segunda matriz y sumando los productos. El número de columnas en la primera matriz debe ser igual al número de filas en la segunda matriz para que la multiplicación sea posible. La multiplicación de matrices no es conmutativa, es decir, AB no es necesariamente igual a BA.

Supongamos que tenemos las siguientes matrices:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

la multiplicación de estas dos matrices dará:

$$A\ x\ B = \begin{bmatrix} (1 * 5 + 2 * 7) & (1 * 6 + 2 * 8) \\ (3 * 5 + 4 * 7) & (3 * 6 + 4 * 8) \end{bmatrix}$$

- Transposición de Matrices

La transposición de una matriz implica cambiar sus filas por columnas (y viceversa). La matriz transpuesta de una matriz A se denota como A^T. Las matrices transpuestas son útiles en álgebra lineal y tienen aplicaciones en la resolución de sistemas de ecuaciones lineales y la representación de transformaciones lineales.

- Determinante

El determinante de una matriz es un número escalar que se calcula a partir de sus elementos y proporciona información importante sobre la matriz. Se denota comúnmente como $det(A)$ o $|A|$ y tiene diversos usos que incluyen:

- Invertibilidad: Si el determinante de una matriz es distinto de cero $(det(A) \neq 0)$, entonces la matriz es invertible y tiene una inversa. Si el determinante es cero $(det(A) = 0)$, la matriz es singular y no tiene una inversa.
- Área o Volumen de Transformaciones Lineales: En geometría, el valor absoluto del determinante de una matriz de transformación lineal representa el factor de escala por el cual el área o el volumen de una figura geométrica se transforma bajo esa transformación.
- Sistemas de Ecuaciones Lineales: El determinante de una matriz se utiliza en el método de Cramer para resolver sistemas de ecuaciones lineales. Si el determinante es distinto de cero, el sistema tiene una solución única.

- Inversa

La inversa de una matriz A es otra matriz A^{-1} que, cuando se multiplica por la matriz original, produce la matriz identidad, una matriz cuadrada en la que todos los elementos de la diagonal principal son 1 y todos los demás 0.

Si una matriz tiene inversa, se dice que es invertible o no singular, y para ello debe cumplir ciertas condiciones como que su determinante no debe ser igual a 0 y, por tanto, debe ser también una matriz cuadrada.

- Producto Escalar

El producto escalar es una operación que toma dos matrices y produce un escalar.

Supongamos que tenemos las siguientes matrices:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

el producto escalar resultará:

$$Producto\ Escalar(A, B) = (1 * 5 + 2 * 6) + (3 * 7 + 4 * 8) = 70$$

- Producto Kronecker

El producto de Kronecker es una operación que toma dos matrices y produce una matriz de mayor dimensión.

Supongamos que tenemos las siguientes matrices:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 0 & 6 \\ 7 & 8 \end{bmatrix}$$

el producto Kronecker resultará:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{bmatrix} = \begin{bmatrix} 1*0 & 1*6 & 2*0 & 2*6 \\ 1*7 & 1*8 & 2*7 & 2*8 \\ 3*0 & 3*6 & 4*0 & 4*6 \\ 3*7 & 3*8 & 4*7 & 4*8 \end{bmatrix}$$

- Norma

Igual que para las vectores, para el caso de las matrices, tenemos varias medidas que nos indican la magnitud de esta y su elección depende del contexto y el tipo de matriz, siendo la norma de Frobenius, que se calcula como sigue, una de las más comunes:

$$||A||_F = \sqrt{\sum_{i,j} |a_{i,j}|^2}$$

donde $a_{i,j}$ son los elementos de la matriz.

Aplicación: Representación de Imágenes en Python

Tal y como comentábamos previamente, una de las aplicaciones de las matrices es la representación de imágenes que, a partir de su color, podemos dividir en imágenes en escala de grises e imágenes a color.

a) Imágenes en escala de grises: Se representan con una sola matriz, en la que cada elemento o píxel, contiene un valor que indica su intensidad luminosa y que normalmente suele oscilar entre 0 (negro) y 255 (blanco).[vi]

python:
```python
import matplotlib.pyplot as plt

# Cargar una imagen en escala de grises
image = plt.imread("path_to_image.jpg")[:, :, 0]

# Mostrar la matriz de la imagen
print(image)

# Visualizar la imagen
plt.imshow(image, cmap='gray')
plt.show()
```

b) Imágenes a color (RGB): Se suelen representar usando el formato RGB (Red, Green, Blue) que descompone la imagen en 3 matrices correspondientes a la intensidad luminosa en cada uno de los respectivos canales.

python:
```python
# Cargar una imagen a color
image_rgb = plt.imread("path_to_image.jpg")

# Descomposición en canales RGB
red_channel = image_rgb[:, :, 0]
green_channel = image_rgb[:, :, 1]
blue_channel = image_rgb[:, :, 2]

# Mostrar cada canal
plt.imshow(red_channel, cmap='Reds')
plt.show()
plt.imshow(green_channel, cmap='Greens')
plt.show()
plt.imshow(blue_channel, cmap='Blues')
plt.show()
```

Una vez dicho esto, representar una imagen a través de matrices nos permite realizar operaciones sobre ellas como, por ejemplo, rotación y escalado.

a) Rotación: Implica multiplicar la matriz / matrices de la imagen por una matriz de rotación que transforma las coordenadas originales de los píxeles.[vii]

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

python:

```
import math

def matrix_multiply(A, B):
    return [[sum(a*b for a, b in zip(row, col)) for col in zip(*B)] for row in A]

# Example rotation of a point (x, y) by 45 degrees
theta = math.radians(45)
rotation_matrix = [
    [math.cos(theta), -math.sin(theta)],
    [math.sin(theta), math.cos(theta)]
]

point = [1, 0]

rotated_point = matrix_multiply(rotation_matrix, [[point[0]], [point[1]]])
print(rotated_point)
```

or

python:

```
from scipy.ndimage import rotate

rotated_image = rotate(image_gray, 45)  # Rota la imagen 45 grados
plt.imshow(rotated_image, cmap='gray')
plt.show()
```

b) Escalado: Implica cambiar el tamaño de una imagen, ya sea ampliándola o reduciéndola.

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

python:

```
# Example scaling of a point (x, y) by factors of 2 and 0.5 in the x and y axes, respectively

scaling_matrix = [
    [2, 0],
    [0, 0.5]
]

scaled_point = matrix_multiply(scaling_matrix, [[point[0]], [point[1]]])
print(scaled_point)
```

or

```python
python:
from scipy.ndimage import zoom

scaled_image = zoom(image_gray, 1.5)  # Escala la imagen por un factor de 1.5
plt.imshow(scaled_image, cmap='gray')
plt.show()
```

4. Diagonalización de matrices. Valores y vectores propios. Descomposición en valores singulares

Valores y vectores propios

Se dice que un escalar λ en K es un valor propio o autovalor de A, si existe un vector X que cumple:

$$(A - \lambda I_n)X = 0$$

Ese escalar λ será valor propio si y sólo si el sistema homogéneo $(A - \lambda I_n) = 0$ admite soluciones no triviales, esto es, si y sólo si:

$$|A - \lambda I_n| = 0$$

El desarrollo de este determinante da lugar a un polinomio de grado n conocido como polinomio característico, cuyas soluciones serán todos los posibles valores propios.

Asimismo, se dice que X es un vector propio o autovector cuando mantiene la misma dirección después de la multiplicación por la matriz A y escalándose por el valor propio correspondiente λ.

$$(A - \lambda I_n)X = 0$$

Si consideramos la matriz 2x2:

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$$

Sus valores propios, vendrán dados por la ecuación característica $det(A - \lambda I) = 0$:

$$det\left(\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} - \lambda I\right) = (2-\lambda)(3-\lambda) - (1)(1) = 0$$

donde resolviendo la ecuación cuadrática, encontramos dos eigenvalores: $\lambda_1 = 2$ y $\lambda_2 = -1$.

- Para $\lambda = 2$:

$$(A - 2I)v = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} v = 0 \rightarrow \begin{cases} v_2 = 0 \\ v_1 + v_2 = 0 \end{cases}$$

Resolviendo este sistema, obtenemos el eigenvector asociado $v = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

- Para $\lambda = -1$:

$$(A + I)v = \begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix} v = 0 \rightarrow \begin{cases} 3v_1 + v_2 = 0 \\ v_1 + 4v_2 = 0 \end{cases}$$

Resolviendo este sistema, obtenemos el eigenvector asociado $v = \begin{bmatrix} -4 \\ 1 \end{bmatrix}$.

Diagonalización

La diagonalización de una matriz es un proceso mediante el cual se descompone una matriz cuadrada en tres componentes: una matriz diagonal, una matriz de autovectores y su inversa, cumpliendo:

$$A = P^{-1}DP$$

lo que permite simplificar cálculos y resolver sistemas de ecuaciones de forma eficiente.

A continuación, se presentan un ejemplo de los pasos necesarios para diagonalizar una matriz partiendo de sus valores y vectores propios:

1) Encuentre los valores propios de la matriz
2) Encuentre los vectores propios de la matriz
3) Organice los vectores propios en una matriz

$$P = [v_1\ v_2 \ldots v_n]$$

4) Organice los valores propios en una matriz

$$D = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}$$

5) Encuentre la matriz inversa de P
6) Compruebe la diagonalización

$$A = P^{-1}DP$$

Si queremos aplicar esto a la resolución de un sistema de ecuaciones lineales tendríamos, por ejemplo,

$$\begin{cases} 2x + y = 3 \\ x + 2y = 4 \end{cases}$$

Si representamos este sistema como una ecuación matriz tendríamos:

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

Si buscamos los autovalores y autovectores tendríamos:

$$\lambda_1 = 3 \ y \ \lambda_2 = 1, v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \ y \ v_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Ahora, formamos la matriz de autovectores P, utilizando los autovectores como columnas y calculamos su inversa.

$$P = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, P^{-1} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Si ahora diagonalizamos A, tenemos:

$$D = P^{-1}AP = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}$$

con lo que se comprueba la diagonalización y, ahora, el sistema de ecuaciones seguirá tal que:

$$\begin{cases} 3x = 3 \\ y = 4 \end{cases} \rightarrow \begin{cases} x = 1 \\ y = 4 \end{cases}$$

Descomposición en valores singulares (SVD)

La descomposición en valores singulares (SVD, por sus siglas en inglés) es un procedimiento fundamental en álgebra lineal y se utiliza para descomponer una matriz A en tres matrices: U, Σ, y V^T, donde U y V son matrices ortogonales (o unitarias en el caso complejo) y Σ es una matriz diagonal que contiene los valores singulares de A.

A continuación, se describen los pasos para descomponer A en valores singulares.

1. **Calcular la matriz A^TA**: Primero, calcula el producto A^TA, que es una matriz cuadrada de tamaño n×n (si A es de tamaño m×n). Esta matriz es simétrica y semidefinida positiva.
2. **Calcular los autovalores de A^TA:** Encuentra los autovalores de la matriz A^TA. Estos autovalores serán no negativos y están relacionados con los valores singulares de A.
3. **Calcular los vectores propios de A^TA:** Encuentra los vectores propios correspondientes a los autovalores que calculaste en el paso anterior. Normaliza estos vectores propios para obtener las columnas de la matriz V.
4. **Ordenar los autovalores**: Ordena los autovalores en orden descendente: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$.
5. **Calcular las raíces cuadradas de los autovalores**: Calcula las raíces cuadradas de los autovalores ordenados: $\sigma_1 = \sqrt{\lambda_1}, \sigma_2 = \sqrt{\lambda_2}, \ldots, \sigma_n = \sqrt{\lambda_n}$. Estos valores singulares son todos no negativos.
6. **Construir la matriz Σ**: Construye una matriz diagonal Σ de tamaño m×n con los valores singulares en la diagonal y ceros en las demás posiciones. Es decir, Σ es una matriz diagonal con los valores singulares $\sigma_1, \sigma_2, \ldots, \sigma_r$ en las r primeras posiciones de la diagonal, donde r es el rango de la matriz A.
7. **Calcular las matrices U y V**: Calcula las matrices U y V utilizando los autovectores de A^TA y AA^T, respectivamente. Normaliza los autovectores para que formen bases ortogonales de U y V.
8. **Construir las matrices U y V^T**: Completa las matrices U y V^T para que tengan dimensiones m×m y n×n, respectivamente, rellenando las columnas adicionales con vectores ortogonales.
9. **La descomposición SVD**: La descomposición en valores singulares final de la matriz A es $A = U\Sigma V^T$.

5. Transformaciones lineales y Espacios Vectoriales

Espacios Vectoriales: Definición y propiedades

Un espacio vectorial, también conocido como espacio lineal, es un concepto fundamental del algebra lineal, que consta de dos conjuntos: un conjunto no vacío (V) de elementos llamados vectores y un conjunto de escalares (K) que se utilizan para combinar los vectores. Estos conjuntos deben cumplir las siguientes propiedades para que el espacio sea considerado espacio vectorial:

1) Cerradura bajo la suma: Sea dos vectores cualesquiera del espacio vectorial, el resultado de su suma debe producir otro vector en el mismo espacio vectorial.
2) Cerradura bajo la multiplicación por escalar: Sea un vector cualquier del espacio, su multiplicación por un escalar debe producir un vector dentro del mismo espacio vectorial

Además, al ser un cuerpo, también cumplirán las siguientes:

3) Asociatividad de la suma:

$$(u+v)+w=u+(v+w)$$

4) Existencia del elemento neutro de la suma: Siendo u un vector del espacio vectorial, debe existir un vector 0, tal que:

$$u+0=0$$

5) Existencia del elemento opuesto de la suma: Siendo u un vector del espacio vectorial, debe existir un vector -u, tal que:

$$u+(-u)=0$$

6) Asociatividad de la multiplicación por escalar:

$$(a*b)*u=a*(b*u)$$

7) Distributividad de la multiplicación por escalar respecto a la suma de vectores:

$$a*(u+v)=a*u+a*v$$

Subespacios Vectoriales

Sea V un espacio vectorial sobre K y U un subconjunto no vacío de V. Se dice que U es un subespacio vectorial de V si, con las mismas operaciones que V, tiene estructura de espacio vectorial sobre K.

Bases: Definición y propiedades

Una base de un espacio vectorial V de tipo finito es un sistema generador S de V que es además linealmente independiente. En otras palabras, una base es un conjunto de vectores tal que cualquier vector en el espacio vectorial puede expresarse de manera única como una combinación lineal de estos vectores.

Un ejemplo de base serían los vectores unitarios i, j y k son una base estándar del espacio tridimensional Euclidiano, ya que cualquier vector puede expresarse como combinación lineal de estos tres vectores.

Similitud del Coseno (Cosine Similarity)

Cosine Similarity es una medida de similitud empleada para comparar la similitud entre dos vectores de características, como vectores de términos en documentos de texto o vectores de características en representaciones de palabras. Esta medida de similitud calcula el ángulo entre los vectores en lugar de considerar la magnitud de estos.

La fórmula para calcularla entre dos vectores A y B se define de la siguiente manera>

$$similarity(A, B) = \frac{A \cdot B}{||A||||B||}$$

donde

- $A \cdot B$ es el producto escalar de los vectores A y B.
- $||A||$ es la norma (longitud) del vector A.
- $||B||$es la norma (longitud) del vector B.

y produce un valor en el rango de -1 a 1, donde:

- Si la similitud del coseno es 1, significa que los vectores son idénticos y apuntan en la misma dirección.
- Si la similitud del coseno es 0, significa que los vectores son ortogonales (no tienen similitud).
- Si la similitud del coseno es -1, significa que los vectores son diametralmente opuestos y apuntan en direcciones opuestas.

En aplicaciones prácticas, cuanto más cercano a 1 sea el valor de similitud del coseno, mayor será la similitud entre los vectores y, por lo tanto, se consideran más similares. Esta medida es especialmente útil en la recuperación de información, donde se pueden comparar documentos basados en la similitud de sus vectores de términos, o en tareas de búsqueda de información donde se pueden encontrar documentos similares a una consulta de búsqueda en función de la similitud del coseno.

Transformaciones Lineales

Se dice que, dados dos espacios vectoriales V y W, la relación $f: V \rightarrow W$ es una aplicación lineal u homomorfismo si verifica:

- Aditividad. Dado cualquier par de vectores, u y v en V,

$$T(u + v) = T(u) + T(v)$$

- Homegeneidad por Escalar. Dado cualquier vector u en V y cualquier escalar c,

$$T(cu) = c * T(u)$$

lo que implica que la transformación lineal conserva la estructura lineal de los vectores, lo que significa que preserva las operaciones de suma y multiplicación por escalar.

Algunos conceptos fundamentales de las aplicaciones lineales son: núcleo, rango (o imagen), composición de transformaciones e inversa.

- El núcleo denotado como $Nul(T)$ es el conjunto de todos los vectores en V que se mapean al vector cero en W.

- El rango de T, denotado como $Im(T)$, representa el conjunto de todos los vectores en W que son imágenes de algún vector en V bajo T.
- Dos o más transformaciones lineales se pueden componer para formar una nueva transformación lineal multiplicando sus matrices correspondientes.
- Una transformación lineal T tiene una inversa, denotada T^{-1}, si y solo si es una transformación biyectiva, es decir, si es una correspondencia uno a uno entre los elementos de V y W, y esta deshará la transformación original.

Cualquier transformación lineal T se representar mediante una matriz. Si V y W son espacios vectoriales con bases $B = \{v_1, v_2, \dots, v_n\}$ y $B' = \{w_1, w_2, \dots, w_n\}$, respectivamente, entonces la matriz de transformación A relaciona los vectores transformados w a través de la ecuación $AX = Y$.

Matemática, quedaría:

$$f: \begin{matrix} V & \rightarrow & W \\ \vec{x} = (x_1, x_2, \dots, x_n)_B & \rightarrow & f(\vec{x}) = \vec{y} = (y_1, y_2, \dots, y_n)_{B'} \end{matrix}$$

tal que:

$$\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_m \end{pmatrix}$$

donde cada columna i de la matriz A representa las componentes de los vectores imagen de B respecto de la base B'.

Si tomamos, como ejemplo, la transformación que rota puntos en el plano 90º en sentido antihorario. Su matriz de transformación seria:

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Así un vector, $v = [x, y]$ después de la transformación quedaría tal que Av

Aplicación: Uso Transformaciones Lineales para Reducción de Dimensionalidad a través de Análisis de Componentes Principales (PCA)

Algunas de las aplicaciones de las Transformaciones Lineales en IA son:

a) Reducción de Dimensionalidad que se emplea para proyectar datos en un espacio de menor dimensión conservando la máxima varianza.
b) Transformación de Características que nos permite modificar los datos para resaltar determinadas propiedades.
c) Redes Neuronales donde las capas realizan transformaciones lineales (seguidas por una activación no lineal) en los datos de entrada.

En esta sección de aplicación, nos centraremos en el Análisis de Componentes Principales (PCA). Esta técnica basada en Algebra Lineal es útil para la visualización de datos de alta dimensión como para acelerar algoritmos de ML manteniendo al máximo la información relevante.

PCA trata de buscar la base en la cual la varianza de los datos es maximizada y normalmente sigue los siguientes pasos:

a) **Centralizar los datos**: Calcule la media de cada característica y reste esta media de la característica.

$$X_{centrada} = X - \mu$$

b) **Calcular la matriz de covarianza,** Σ:

$$\Sigma = \frac{1}{N} X_{centrada}^{T} X_{centrada}$$

c) **Calcular los eigenvectores y eigenvalores de** Σ. Los eigenvectores representan las direcciones de máxima varianza, y los eigenvalores indican la magnitud de esa varianza.

d) **Ordenar los eigenvalores en orden descendente** y elegir los k eigenvectores asociados a los k eigenvalores más grandes. Esto forma la matriz W de dimensiones d×k.

e) **Transformar los datos originales**:

$$X_{reducida} = X_{centrada} W$$

python:

```
import numpy as np
import matplotlib.pyplot as plt

# Generando un conjunto de datos bidimensional
np.random.seed(0)
X = np.dot(np.random.rand(2, 2), np.random.randn(2, 200)).T

# Visualizar datos originales
plt.scatter(X[:, 0], X[:, 1])
plt.title('Datos originales')
plt.xlabel('x1')
plt.ylabel('x2')
plt.axis('equal')
plt.show()

# 1. Estandarizar los datos
mean = np.mean(X, axis=0)
std_dev = np.std(X, axis=0)
X_normalized = (X - mean) / std_dev

# 2. Calcular la matriz de covarianza
covariance_matrix = np.cov(X_normalized.T)

# 3. Obtener los eigenvectores y eigenvalores
eigenvalues, eigenvectors = np.linalg.eigh(covariance_matrix)

# 4. Ordenar los eigenvectores por eigenvalores descendentes
sorted_index = np.argsort(eigenvalues)[::-1]
sorted_eigenvectors = eigenvectors[:, sorted_index]

# 5. Seleccionar los componentes principales (solo el primer componente en este caso)
component = sorted_eigenvectors[:, 0]

# 6. Transformar el conjunto de datos original
```

```
X_pca = X_normalized.dot(component)

# Visualizar los datos transformados
plt.scatter(X_pca, [1] * len(X_pca), alpha=0.5)
plt.title('Datos transformados con PCA')
plt.xlabel('Principal Component')
plt.show()
```

or

```
python:
from sklearn.decomposition import PCA

# Datos ficticios: 10 puntos en un espacio 3D
data = [[2, 3, 1], [3, 4, 2], [4, 5, 3], [5, 6, 1], [2, 2, 2],
        [3, 3, 3], [4, 4, 4], [5, 5, 5], [1, 2, 3], [6, 7, 8]]

# Aplicar PCA para reducir a 2D
pca = PCA(n_components=2)
transformed_data = pca.fit_transform(data)
print(transformed_data)
```

[i] Burgos, J. (2006). Álgebra Lineal y Geometría Cartesiana. Ed. McGraw Hill
[ii] Larson, R., Edwards, B. H., Falvo, D. (2004). Álgebra Lineal. Ed. Pirámide
[iii] Strang, G. (2006). Linear algebra and its applications. Thomson, Brooks/Cole.
[iv] Gilbert, J. R., & Strang, G. (2014). Linear algebra and learning from data. Wellesley-Cambridge Press.5
[v] Lay, D. C. (2015). Linear algebra and its applications. Pearson.
[vi] Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing (3rd ed.) Pearson/Prentice Hall.
[vii] Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer Science & Business Media.