Fundamentos de Algebra Lineal

Vectores, Matrices y Transformaciones

Preparacion para Deep Learning

Guia de Estudio

1 de febrero de 2026

Índice

Parte I
Vectores

1. Que es un Vector

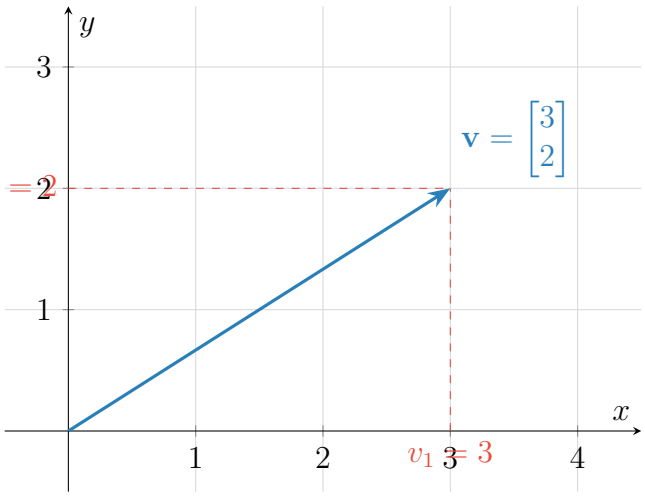
Vector

Un **vector** es una lista ordenada de numeros. En n dimensiones:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \in \mathbb{R}^n$$

Cada v_i se llama **componente** o **entrada** del vector.

Vector en $\mathbb{R}^2$



En dimensiones mayores ($\mathbb{R}^n$ con $n > 3$), no podemos visualizar los vectores, pero las operaciones funcionan igual. Por ejemplo, en $\mathbb{R}^{128}$: $\mathbf{v} = [v_1, v_2, \ldots, v_{128}]^T$.

Conexion con IA:

Representacion de Datos En Deep Learning, **todo son vectores**:

- Una imagen en escala de grises de 28×28 se "aplana" a un vector de $\mathbb{R}^{784}$.
- Un embedding de palabra es un vector de $\mathbb{R}^{300}$ (o similar).
- Los datos de un sensor (9 canales $\times$ 128 timesteps) forman $\mathbf{x} \in \mathbb{R}^{1152}$.

2. Operaciones Basicas con Vectores

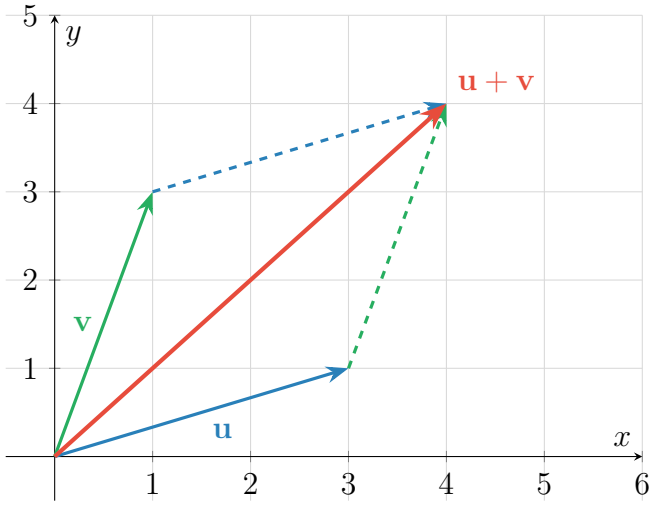
2.1. Suma de Vectores

Suma

Se suman componente a componente:

$$\mathbf{u} + \mathbf{v} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} u_1 + v_1 \\ u_2 + v_2 \end{bmatrix}$$

Suma de Vectores (Regla del Paralelogramo)



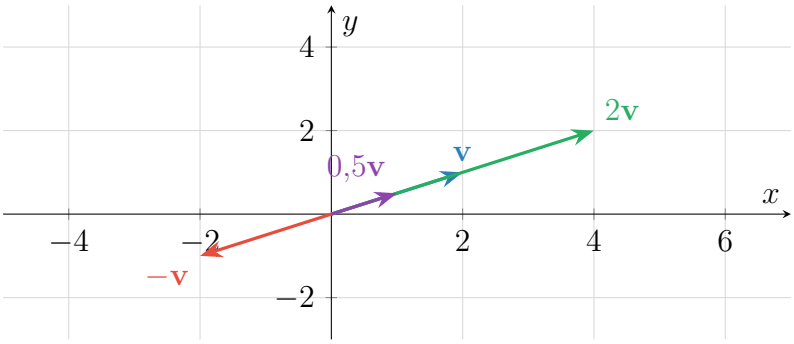
2.2. Multiplicacion por Escalar

Multiplicacion por Escalar

Multiplicar cada componente por un numero $c \in \mathbb{R}$:

$$c \cdot \mathbf{v} = c \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} c \cdot v_1 \\ c \cdot v_2 \end{bmatrix}$$

Efecto de Escalar un Vector



Conexion con IA:

Pesos en Redes Neuronales En una neurona, la operacion $w \cdot x$ es exactamente multiplicacion por escalar. Si $w > 1$, la senal se amplifica; si $0 < w < 1$, se atenua; si $w < 0$, se invierte.

3. Producto Punto (Producto Escalar)

Producto Punto

El producto punto de dos vectores produce un **escalar**:

$$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} u_i v_i = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n$$

Forma geometrica:

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

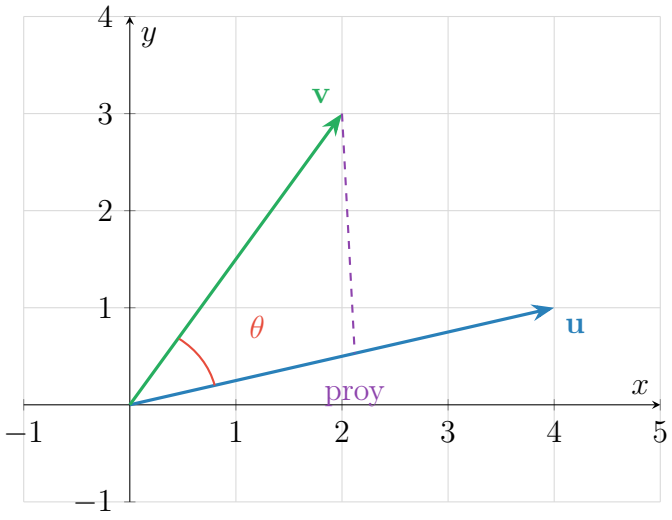
donde θ es el angulo entre los vectores.

Calculo de Producto Punto

$$\mathbf{u} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

$$\mathbf{u} \cdot \mathbf{v} = (1)(4) + (2)(5) + (3)(6) = 4 + 10 + 18 = 32$$

Interpretacion Geometrica del Producto Punto



Casos Especiales

- $\mathbf{u} \cdot \mathbf{v} > 0$: Angulo agudo ($\theta < 90$), vectores “apuntan similar”.
- $\mathbf{u} \cdot \mathbf{v} = 0$: Vectores **perpendiculares** (ortogonales).

- $\mathbf{u} \cdot \mathbf{v} < 0$: Angulo obtuso ($\theta > 90$), vectores "opuestos".

Conexion con IA:

Operacion Fundamental El producto punto es **LA** operacion basica de las redes neuronales:

$$z = \mathbf{w} \cdot \mathbf{x} + b = \sum_i w_i x_i + b$$

Una neurona calcula el producto punto entre pesos y entrada, suma el bias, y aplica la activacion.

4. Norma de un Vector

Norma Euclidiana (L2)

La **norma** o **magnitud** de un vector es su "longitud":

$$\|\mathbf{v}\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2} = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

Usando producto punto:

$$\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$$

Otras Normas Importantes

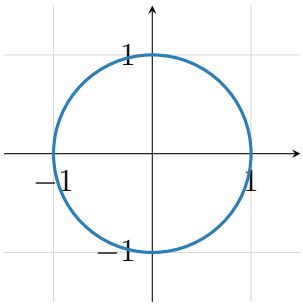
Norma L1 (Manhattan):

$$\|\mathbf{v}\|_1 = \sum_{i=1}^{n} |v_i|$$

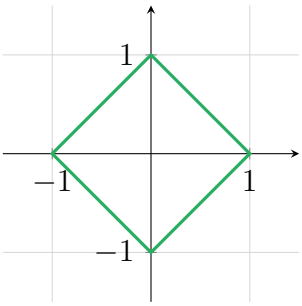
Norma L-infinito (Maximo):

$$\|\mathbf{v}\|_\infty = \max_i |v_i|$$

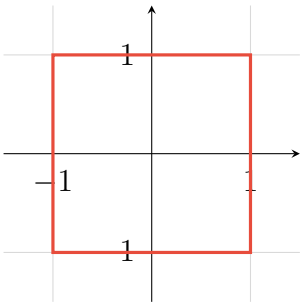
L2: $\|\mathbf{v}\|_2 = 1$



L1: $\|\mathbf{v}\|_1 = 1$



L∞: $\|\mathbf{v}\|_\infty = 1$



Conexion con IA:

Regularizacion

- **L2 Regularization:** Penaliza $\|\mathbf{w}\|_2^2 = \sum w_i^2$. Produce pesos pequenos.
- **L1 Regularization:** Penaliza $\|\mathbf{w}\|_1 = \sum |w_i|$. Produce pesos **sparse** (muchos ceros).

5. Vectores Unitarios y Normalizacion

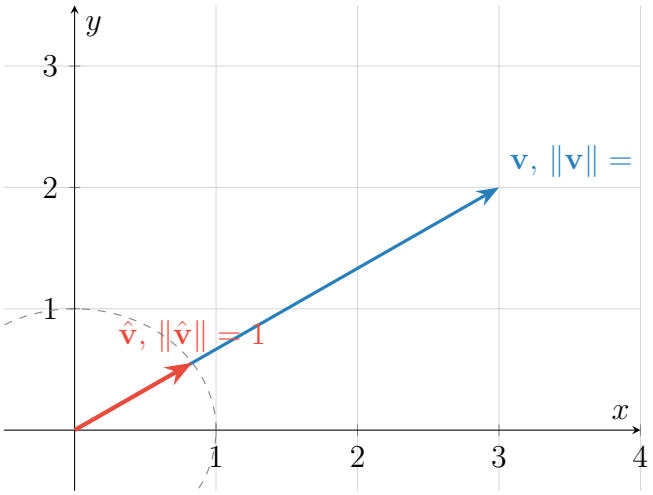
Vector Unitario

Un vector con norma 1. Para normalizar cualquier vector:

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

El vector $\hat{\mathbf{v}}$ apunta en la misma direccion que $\mathbf{v}$ pero tiene longitud 1.

Normalizacion de Vector



Conexion con IA:

Normalizacion en Deep Learning

- **Embeddings:** Frecuentemente se normalizan para usar similitud coseno.
- **Batch Normalization:** Normaliza activaciones para estabilizar el entrenamiento.
- **Layer Normalization:** Normaliza a traves de features (comun en Transformers).

Parte II
Matrices

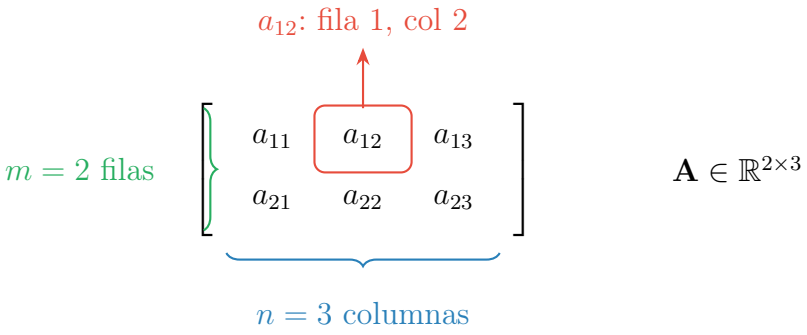
6. Que es una Matriz

Matriz

Una **matriz** es un arreglo rectangular de numeros con m filas y n columnas:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

Notacion: a_{ij} es el elemento en la fila i, columna j.



Conexion con IA:

Matrices en Deep Learning

- **Pesos de capa:** $\mathbf{W} \in \mathbb{R}^{\text{salida} \times \text{entrada}}$
- **Batch de datos:** $\mathbf{X} \in \mathbb{R}^{\text{batch} \times \text{features}}$
- **Imagen:** $\mathbf{I} \in \mathbb{R}^{\text{alto} \times \text{ancho}}$ (escala de grises)

7. Matrices Especiales

Matrices Especiales

Matriz Identidad $\mathbf{I}$: Unos en la diagonal, ceros fuera.

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Propiedad: $\mathbf{AI} = \mathbf{IA} = \mathbf{A}$

Matriz Diagonal: Solo tiene valores en la diagonal.

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix}$$

Matriz de Ceros: Todos los elementos son cero.
Matriz Transpuesta $\mathbf{A}^T$: Intercambia filas por columnas.

$$\text{Si } \mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \text{ entonces } \mathbf{A}^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

$$\underset{\mathbf{A}\ (3\times 2)}{\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}} \xrightarrow{\text{Transponer}} \underset{\mathbf{A}^T\ (2\times 3)}{\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}}$$

8. Multiplicacion de Matrices

Producto Matricial

Para $\mathbf{A} \in \mathbb{R}^{m\times n}$ y $\mathbf{B} \in \mathbb{R}^{n\times p}$, el producto $\mathbf{C} = \mathbf{AB}$ tiene dimensiones $m \times p$:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} = (\text{fila } i \text{ de } \mathbf{A}) \cdot (\text{columna } j \text{ de } \mathbf{B})$$

Requisito: El numero de columnas de $\mathbf{A}$ debe ser igual al numero de filas de $\mathbf{B}$.

$$\underset{\mathbf{A}\ (2\times 3)}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}} \times \underset{\mathbf{B}\ (3\times 2)}{\begin{bmatrix} 7 & 10 \\ 8 & 11 \\ 9 & 12 \end{bmatrix}} = \underset{\mathbf{C}\ (2\times 2)}{\begin{bmatrix} 50 & 68 \\ 122 & 167 \end{bmatrix}}$$

$$c_{11} = 1(7) + 2(8) + 3(9) = 7 + 16 + 27 = 50$$

Propiedades del Producto Matricial

- **No conmutativo:** $\mathbf{AB} \neq \mathbf{BA}$ en general.
- **Asociativo:** $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$

- **Distributivo:** $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB}+\mathbf{AC}$
- **Transpuesta:** $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$

Conexion con IA:

Forward Pass El paso hacia adelante en una capa densa es multiplicacion matricial:

$$\mathbf{Y} = \mathbf{X}\mathbf{W}^T + \mathbf{b}$$

Donde:

- $\mathbf{X} \in \mathbb{R}^{\text{batch}\times\text{features\_in}}$
- $\mathbf{W} \in \mathbb{R}^{\text{features\_out}\times\text{features\_in}}$
- $\mathbf{Y} \in \mathbb{R}^{\text{batch}\times\text{features\_out}}$

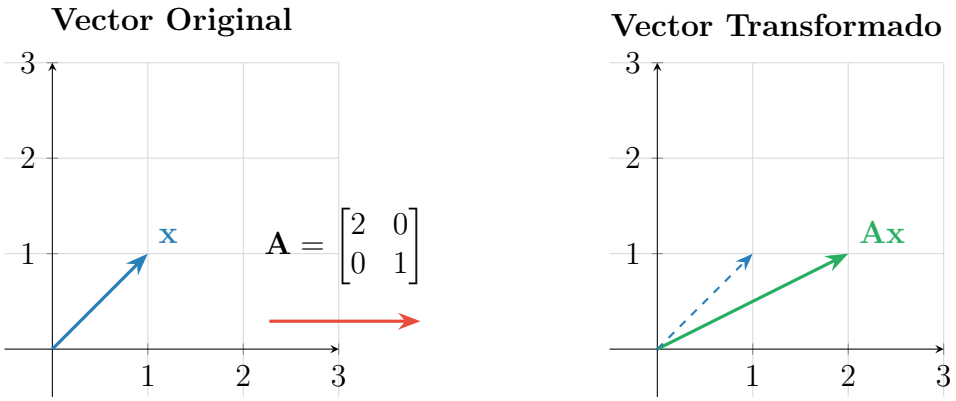
9. Producto Matriz-Vector

Matriz por Vector

Es un caso especial de multiplicacion matricial:

$$\mathbf{Ax} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{bmatrix}$$

Interpretacion: Es una **combinacion lineal** de las columnas de $\mathbf{A}$.



Parte III
Transformaciones Lineales

10. Que es una Transformacion Lineal

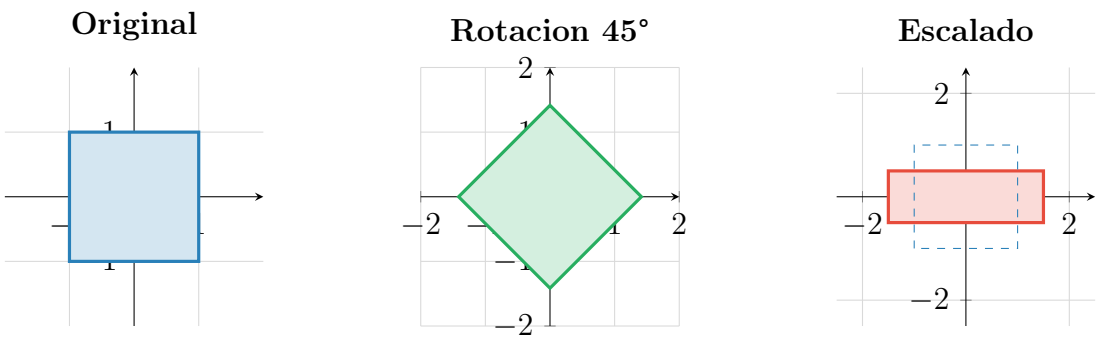
Transformacion Lineal

Una funcion $T : \mathbb{R}^n \to \mathbb{R}^m$ es **lineal** si cumple:

1. **Aditividad:** $T(\mathbf{u} + \mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$
2. **Homogeneidad:** $T(c\mathbf{v}) = cT(\mathbf{v})$

Toda transformacion lineal puede representarse como multiplicacion por una matriz:

$$T(\mathbf{x}) = \mathbf{A}\mathbf{x}$$



11. Tipos de Transformaciones

11.1. Rotacion

Matriz de Rotacion 2D

Rotar por un angulo θ (en sentido antihorario):

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

11.2. Escalado

Matriz de Escalado

Escalar por factores s_x y s_y en cada eje:

$$\mathbf{S} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

11.3. Reflexion

Matrices de Reflexion

Reflexion respecto al eje x:

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

Reflexion respecto al eje y:

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

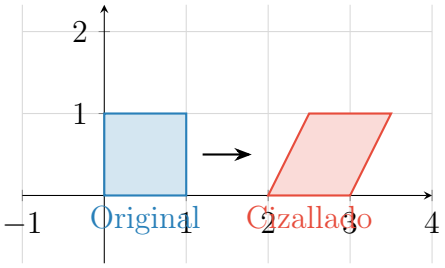
11.4. Cizallamiento (Shear)

Matriz de Cizalla

Cizalla horizontal:

$$\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$$

Cizallamiento (Shear)



Conexion con IA:

Transformaciones en Redes Neuronales Cada capa de una red neuronal aplica una transformacion:

$$\mathbf{h} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

- $\mathbf{W}$: Transformacion lineal (rotacion, escalado, proyeccion).
- $\mathbf{b}$: Traslacion (desplazamiento del origen).
- σ: No linealidad ("dobla" el espacio).

Sin σ, multiples capas colapsan en una sola transformacion lineal.

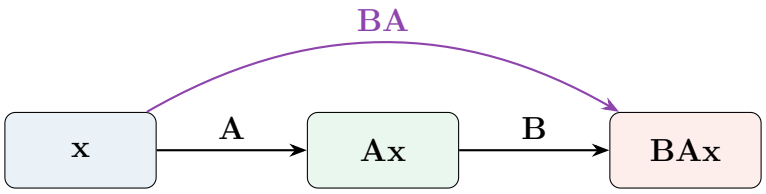
12. Composicion de Transformaciones

Composicion

Aplicar varias transformaciones en secuencia equivale a multiplicar sus matrices:

$$T_2(T_1(\mathbf{x})) = \mathbf{B}(\mathbf{A}\mathbf{x}) = (\mathbf{B}\mathbf{A})\mathbf{x}$$

Importante: El orden importa. Primero se aplica la matriz mas cercana al vector.



Conexion con IA:

Redes Profundas Una red con L capas (sin activaciones) es:

$$\mathbf{W}_L\mathbf{W}_{L-1}\cdots\mathbf{W}_2\mathbf{W}_1\mathbf{x} = \mathbf{W}_{equiv}\mathbf{x}$$

Esto colapsa a una sola matriz. Por eso necesitamos activaciones no lineales entre capas.

Parte IV
Conceptos Avanzados

13. Determinante

Determinante

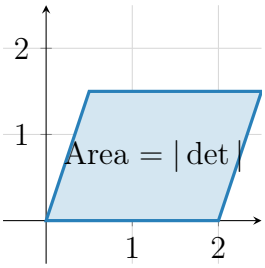
El **determinante** de una matriz cuadrada es un escalar que indica:

- El **factor de escala** del area/volumen bajo la transformacion.
- Si la matriz es **invertible** ($\det \neq 0$).

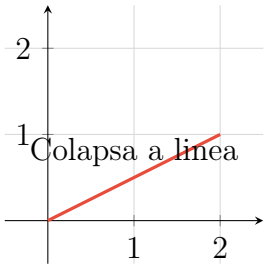
Para 2×2:

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

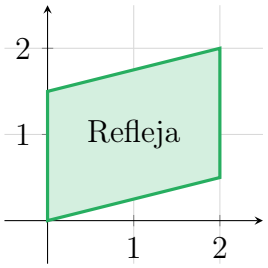
det > 0



det $= 0$



det < 0



14. Matriz Inversa

Inversa

La **inversa** $\mathbf{A}^{-1}$ de una matriz $\mathbf{A}$ cumple:

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

"Deshace" la transformacion aplicada por $\mathbf{A}$.
Una matriz tiene inversa si y solo si $\det(\mathbf{A}) \neq 0$.
Para 2×2:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

15. Eigenvalores y Eigenvectores

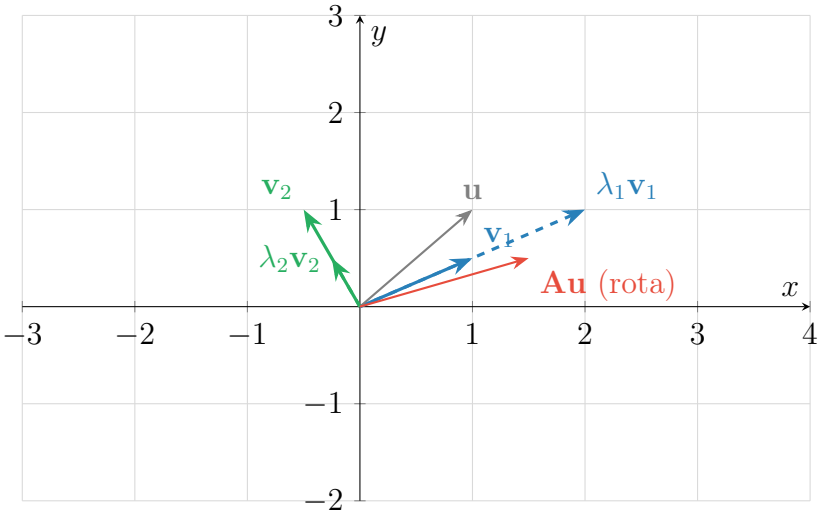
Eigenvector y Eigenvalor

Un vector $\mathbf{v} \neq \mathbf{0}$ es **eigenvector** de $\mathbf{A}$ si al aplicar $\mathbf{A}$, el vector solo se escala (no cambia direccion):

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

donde λ es el **eigenvalor** correspondiente.

Eigenvectores: Direcciones Especiales



Conexion con IA:

PCA y Reduccion de Dimensionalidad **PCA** (Analisis de Componentes Principales) usa eigenvectores:

1. Calcula la matriz de covarianza de los datos.
2. Encuentra sus eigenvectores y eigenvalores.
3. Los eigenvectores con mayores eigenvalores son las "direcciones principales".
4. Proyecta los datos en estas direcciones para reducir dimensionalidad.

16. Descomposicion en Valores Singulares (SVD)

SVD

Cualquier matriz $\mathbf{A} \in \mathbb{R}^{m\times n}$ se puede descomponer como:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

Donde:

- $\mathbf{U} \in \mathbb{R}^{m\times m}$: Matriz ortogonal (columnas = vectores singulares izquierdos).
- $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$: Matriz diagonal con valores singulares $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$.

- $\mathbf{V} \in \mathbb{R}^{n \times n}$: Matriz ortogonal (columnas = vectores singulares derechos).

$$\underset{m \times n}{\begin{bmatrix} \\ \mathbf{A} \\ \\ \end{bmatrix}} = \underset{m \times m}{\begin{bmatrix} \\ \mathbf{U} \\ \\ \end{bmatrix}} \underset{m \times n}{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \\ & & \end{bmatrix}} \underset{n \times n}{\begin{bmatrix} \\ \mathbf{V}^T \\ \\ \end{bmatrix}}$$

Conexion con IA:

Aplicaciones de SVD

- **Compresion:** Aproximar $\mathbf{A}$ usando solo los k valores singulares mas grandes.
- **Regularizacion:** Truncar valores singulares pequenos reduce ruido.
- **Inicializacion de pesos:** Algunas tecnicas usan SVD para inicializar.

Parte V
Operaciones para Deep Learning

17. Operaciones Elemento a Elemento

Operaciones Element-wise

Operaciones que se aplican **componente por componente**:

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

(Diferente del producto matricial **AB**)

Producto Matricial vs Element-wise

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix} \quad \text{(matricial)}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \odot \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 5 & 12 \\ 21 & 32 \end{bmatrix} \quad \text{(element-wise)}$$

18. Broadcasting

Broadcasting

Mecanismo que permite operar con arrays de diferentes dimensiones, "expandiendo" automaticamente el mas pequeno.
Reglas:

1. Comparar dimensiones de derecha a izquierda.
2. Dimensiones compatibles si son iguales o una es 1.
3. La dimension 1 se "estira" para igualar.

$$\underset{(2\times 3)}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}} + \underset{(2\times 1)}{\begin{bmatrix} 10 \\ 20 \end{bmatrix}} = \underset{(2\times 3)}{\begin{bmatrix} 11 & 12 & 13 \\ 24 & 25 & 26 \end{bmatrix}}$$

Se "expande" a cada columna

Conexion con IA:

Bias en Redes Neuronales El bias se suma usando broadcasting:

$$\mathbf{Z} = \mathbf{X}\mathbf{W}^T + \mathbf{b}$$

Donde $\mathbf{X}\mathbf{W}^T$ es $(batch, features)$ y $\mathbf{b}$ es $(1, features)$ o $(features,)$.

A. Tabla Resumen: Algebra Lineal a Deep Learning

Concepto	**Notacion/Formula**	**Uso en Deep Learning**
Vector	$\mathbf{v} \in \mathbb{R}^n$	Representacion de datos, embeddings
Producto punto	$\mathbf{u} \cdot \mathbf{v} = \sum u_i v_i$	Operacion de neurona, similitud
Norma L2	$\|\mathbf{v}\|_2 = \sqrt{\sum v_i^2}$	Regularizacion, normalizacion
Matriz	$\mathbf{A} \in \mathbb{R}^{m \times n}$	Pesos de capas, batches
Producto matricial	$\mathbf{C} = \mathbf{AB}$	Forward pass, capas densas
Transpuesta	$\mathbf{A}^T$	Calcular gradientes, matmul
Transformacion lineal	$T(\mathbf{x}) = \mathbf{Ax}$	Cada capa aplica una
Composicion	$\mathbf{BA}$	Red profunda = composicion
Eigenvalores	$\mathbf{Av} = \lambda\mathbf{v}$	PCA, analisis de estabilidad
SVD	$\mathbf{A} = \mathbf{U\Sigma V}^T$	Compresion, inicializacion
Broadcasting	Expansion automatica	Sumar bias, operaciones batch

Cuadro 1: Mapeo de Algebra Lineal a Deep Learning

B. Formulas Clave para el Proyecto HAR

Resumen de Formulas de Algebra Lineal

1. Capa Densa (Linear):

$$\mathbf{Y} = \mathbf{X}\mathbf{W}^T + \mathbf{b}$$

$\mathbf{X}$: $(batch, in)$, $\mathbf{W}$: (out, in), $\mathbf{Y}$: $(batch, out)$

2. Convolucion 1D (simplificada):

$$y[t] = \sum_k x[t+k] \cdot w[k]$$

Producto punto deslizante con kernel $\mathbf{w}$

3. Regularizacion L2:

$$\mathcal{L}_{total} = \mathcal{L}_{data} + \lambda \|\mathbf{W}\|_2^2$$

4. Normalizacion de vector:

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