

Module 1 : Fondements mathématiques (fonctions, matrices, algèbre linéaire)

Présentée par : Tiebekabe Pagdame
Enseignant-chercheur - Université de Kara

Dates : 15-16 juillet 2025

Objectifs de la session

- Revoir les concepts fondamentaux de fonctions, matrices et algèbre linéaire.
- Comprendre les propriétés utiles en machine learning, deep learning, traitement du signal, etc.
- Développer l'intuition géométrique et l'agilité computationnelle.
- Préparer le terrain pour les réseaux de neurones et les transformations linéaires.

Public cible

- Étudiants en Mathématiques/Informatique et Science des Données
- Étudiants à la Faculté des Sciences et de la Santé
- Chercheurs en NLP
- Professionnels du secteur

- 1 Notions fondamentales sur les fonctions
- 2 Matrices et opérations matricielles
- 3 Algèbre linéaire avancée

Définition formelle d'une fonction

Fonction (définition)

Soient A et B deux ensembles. Une **fonction** f de A vers B , notée $f : A \rightarrow B$, est une application qui associe à chaque élément $x \in A$ un unique élément $f(x) \in B$.

- A est appelé le **domaine de définition** (ou ensemble de départ).
- B est appelé le **codomaine** (ou ensemble d'arrivée).
- L'ensemble des valeurs effectivement prises par f est l'**image** de f : $\text{Im}(f) = \{f(x) \mid x \in A\} \subseteq B$.

Exemple

$f : \mathbb{R} \rightarrow \mathbb{R}$ défini par $f(x) = x^2$:

- Domaine : \mathbb{R} , Codomaine : \mathbb{R}
- Image : $\mathbb{R}_+ = [0, +\infty)$

Définition formelle d'une fonction

Fonction (définition)

Soient A et B deux ensembles. Une **fonction** f de A vers B , notée $f : A \rightarrow B$, est une application qui associe à chaque élément $x \in A$ un unique élément $f(x) \in B$.

- A est appelé le **domaine de définition** (ou ensemble de départ).
- B est appelé le **codomaine** (ou ensemble d'arrivée).
- L'ensemble des valeurs effectivement prises par f est l'**image** de f : $\text{Im}(f) = \{f(x) \mid x \in A\} \subseteq B$.

Exemple

$f : \mathbb{R} \rightarrow \mathbb{R}$ défini par $f(x) = x^2$:

- Domaine : \mathbb{R} , Codomaine : \mathbb{R}
- Image : $\mathbb{R}_+ = [0, +\infty)$

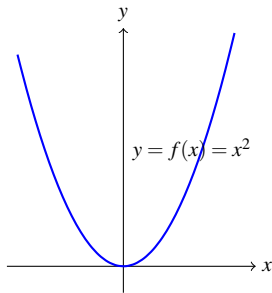
Graphe d'une fonction

Définition

Le **graphe** d'une fonction $f : A \rightarrow B$ est l'ensemble des couples :

$$\text{Graph}(f) = \{(x, f(x)) \mid x \in A\} \subseteq A \times B$$

- Chaque point du graphe représente un lien $x \mapsto f(x)$.
- En géométrie, pour $f : \mathbb{R} \rightarrow \mathbb{R}$, le graphe est une courbe dans le plan.



Remarque : Une courbe n'est le graphe d'une fonction que si toute verticale coupe la courbe en au plus un point.

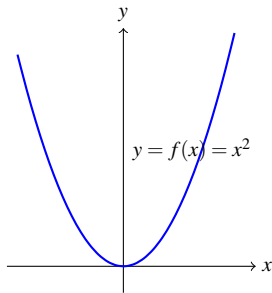
Graphe d'une fonction

Définition

Le **graphe** d'une fonction $f : A \rightarrow B$ est l'ensemble des couples :

$$\text{Graph}(f) = \{(x, f(x)) \mid x \in A\} \subseteq A \times B$$

- Chaque point du graphe représente un lien $x \mapsto f(x)$.
- En géométrie, pour $f : \mathbb{R} \rightarrow \mathbb{R}$, le graphe est une courbe dans le plan.



Remarque : Une courbe n'est le graphe d'une fonction que si toute verticale coupe la courbe en au plus un point.

Fonction injective (injection)

Définition

Une fonction $f : A \rightarrow B$ est dite **injective** si :

$$\forall x_1, x_2 \in A, \quad f(x_1) = f(x_2) \Rightarrow x_1 = x_2$$

- Autrement dit, deux éléments différents de A ont toujours des images différentes.
- Il n'y a pas de "collisions" dans l'image.

Exemple

$f : \mathbb{R} \rightarrow \mathbb{R}$ défini par $f(x) = 2x + 1$ est injective. Mais $f(x) = x^2$ ne l'est pas sur \mathbb{R} car $f(1) = f(-1)$.

Fonction injective (injection)

Définition

Une fonction $f : A \rightarrow B$ est dite **injective** si :

$$\forall x_1, x_2 \in A, \quad f(x_1) = f(x_2) \Rightarrow x_1 = x_2$$

- Autrement dit, deux éléments différents de A ont toujours des images différentes.
- Il n'y a pas de "collisions" dans l'image.

Exemple

$f : \mathbb{R} \rightarrow \mathbb{R}$ défini par $f(x) = 2x + 1$ est injective. Mais $f(x) = x^2$ ne l'est pas sur \mathbb{R} car $f(1) = f(-1)$.

Fonction surjective (surjection)

Définition

Une fonction $f : A \rightarrow B$ est dite **surjective** si :

$$\forall y \in B, \exists x \in A \text{ tel que } f(x) = y$$

- Autrement dit, l'image de f est exactement égale au codomaine : $\text{Im}(f) = B$.
- Tout élément du codomaine est atteint par la fonction.

Exemple

$f : \mathbb{R} \rightarrow \mathbb{R}$ défini par $f(x) = x^3$ est surjective. Mais $f(x) = e^x$ n'est pas surjective si $B = \mathbb{R}$.

Fonction surjective (surjection)

Définition

Une fonction $f : A \rightarrow B$ est dite **surjective** si :

$$\forall y \in B, \exists x \in A \text{ tel que } f(x) = y$$

- Autrement dit, l'image de f est exactement égale au codomaine : $\text{Im}(f) = B$.
- Tout élément du codomaine est atteint par la fonction.

Exemple

$f : \mathbb{R} \rightarrow \mathbb{R}$ défini par $f(x) = x^3$ est surjective. Mais $f(x) = e^x$ n'est pas surjective si $B = \mathbb{R}$.

Fonction bijective (bijection)

Définition

Une fonction $f : A \rightarrow B$ est dite **bijective** si elle est à la fois :

- injective : chaque valeur de B est atteinte par un seul x
- surjective : chaque $y \in B$ a un antécédent dans A

- Une bijection possède une **fonction réciproque** $f^{-1} : B \rightarrow A$ telle que $f^{-1}(f(x)) = x$.
- Les bijections permettent de faire des "changements de variables" ou des codages.

Exemple

$f : \mathbb{R} \rightarrow \mathbb{R}$ défini par $f(x) = x + 5$ est bijective.

Fonction bijective (bijection)

Définition

Une fonction $f : A \rightarrow B$ est dite **bijective** si elle est à la fois :

- injective : chaque valeur de B est atteinte par un seul x
- surjective : chaque $y \in B$ a un antécédent dans A

- Une bijection possède une **fonction réciproque** $f^{-1} : B \rightarrow A$ telle que $f^{-1}(f(x)) = x$.
- Les bijections permettent de faire des "changements de variables" ou des codages.

Exemple

$f : \mathbb{R} \rightarrow \mathbb{R}$ défini par $f(x) = x + 5$ est bijective.

Fonctions linéaires et affines

Fonction linéaire

Une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ est dite **linéaire** si $\exists a \in \mathbb{R}$ tel que $f(x) = ax$.

Fonction affine

Une fonction est **affine** si $f(x) = ax + b$ avec $a, b \in \mathbb{R}$.

- Les fonctions linéaires sont les transformations de type homothéties.
- Les fonctions affines incluent une translation (elles représentent des droites).

Applications

Les neurones artificiels combinent souvent une transformation affine $f(x) = w^T x + b$ suivie d'une non-linéarité.

Fonctions linéaires et affines

Fonction linéaire

Une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ est dite **linéaire** si $\exists a \in \mathbb{R}$ tel que $f(x) = ax$.

Fonction affine

Une fonction est **affine** si $f(x) = ax + b$ avec $a, b \in \mathbb{R}$.

- Les fonctions linéaires sont les transformations de type homothéties.
- Les fonctions affines incluent une translation (elles représentent des droites).

Applications

Les neurones artificiels combinent souvent une transformation affine $f(x) = w^T x + b$ suivie d'une non-linéarité.

Fonctions polynomiales

Définition

Une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ est polynomiale de degré n si :

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \quad \text{avec } a_n \neq 0$$

- Les polynômes modélisent des comportements courbes, sont dérivables partout.
- Leur étude s'appuie sur l'algèbre linéaire (espaces vectoriels de polynômes).

Utilisation

Les polynômes interviennent dans les séries de Taylor, les modèles de régression non-linéaire, etc.

Fonctions polynomiales

Définition

Une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ est polynomiale de degré n si :

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \quad \text{avec } a_n \neq 0$$

- Les polynômes modélisent des comportements courbes, sont dérivables partout.
- Leur étude s'appuie sur l'algèbre linéaire (espaces vectoriels de polynômes).

Utilisation

Les polynômes interviennent dans les séries de Taylor, les modèles de régression non-linéaire, etc.

Fonction exponentielle

Définition

La fonction exponentielle réelle est définie par :

$$f(x) = e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

- Strictement croissante, dérivable partout, $f'(x) = f(x)$.
- Image : $(0, +\infty)$; bijection entre \mathbb{R} et \mathbb{R}_+^* .

Application en IA

Intervient dans les fonctions d'activation comme la **sigmoïde** : $\sigma(x) = \frac{1}{1+e^{-x}}$.

Fonction exponentielle

Définition

La fonction exponentielle réelle est définie par :

$$f(x) = e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

- Strictement croissante, dérivable partout, $f'(x) = f(x)$.
- Image : $(0, +\infty)$; bijection entre \mathbb{R} et \mathbb{R}_+^* .

Application en IA

Intervient dans les fonctions d'activation comme la **sigmoïde** : $\sigma(x) = \frac{1}{1+e^{-x}}$.

Fonction logarithme népérien

Définition

La fonction logarithme népérien est la bijection réciproque de l'exponentielle :

$$\ln(x) = y \Leftrightarrow x = e^y, \quad \text{pour } x > 0$$

- Strictement croissante, dérivable sur $(0, +\infty)$.
- $\ln(ab) = \ln(a) + \ln(b)$, $\ln(a^r) = r \ln(a)$.

Utilisation

Très utilisé en backpropagation (log-loss), softmax, ou en normalisation des valeurs.

Fonction logarithme népérien

Définition

La fonction logarithme népérien est la bijection réciproque de l'exponentielle :

$$\ln(x) = y \Leftrightarrow x = e^y, \quad \text{pour } x > 0$$

- Strictement croissante, dérivable sur $(0, +\infty)$.
- $\ln(ab) = \ln(a) + \ln(b)$, $\ln(a^r) = r \ln(a)$.

Utilisation

Très utilisé en backpropagation (log-loss), softmax, ou en normalisation des valeurs.

Fonction sigmoïde

Définition

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- Image : $(0, 1)$
- Dérivable : $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
- Fonction non linéaire, à pente maximale en $x = 0$

Propriétés en Deep Learning

- Bonne interprétation probabiliste (utilisée en sortie pour des probabilités).
- Peut saturer : le gradient devient quasi nul pour $|x| \gg 0$.

Fonction sigmoïde

Définition

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- Image : $(0, 1)$
- Dérivable : $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
- Fonction non linéaire, à pente maximale en $x = 0$

Propriétés en Deep Learning

- Bonne interprétation probabiliste (utilisée en sortie pour des probabilités).
- Peut saturer : le gradient devient quasi nul pour $|x| \gg 0$.

Fonction tangente hyperbolique (Tanh)

Définition

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- Image : $(-1, 1)$, centrée sur zéro
- Dérivable : $\tanh'(x) = 1 - \tanh^2(x)$
- Courbe en forme de sigmoïde plus "centrée"

Avantages

- Zéro-centered meilleure convergence dans certains cas.
- Même inconvénient que $\sigma(x)$: saturation pour $|x|$ grand.

Fonction tangente hyperbolique (Tanh)

Définition

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- Image : $(-1, 1)$, centrée sur zéro
- Dérivable : $\tanh'(x) = 1 - \tanh^2(x)$
- Courbe en forme de sigmoïde plus "centrée"

Avantages

- Zéro-centered meilleure convergence dans certains cas.
- Même inconvénient que $\sigma(x)$: saturation pour $|x|$ grand.

Fonction d'activation ReLU

Définition

$$\text{ReLU}(x) = \max(0, x)$$

- Non dérivable en $x = 0$, mais utilisée massivement en pratique.
- Simple, computationnellement efficace.
- Image : $[0, +\infty)$

Avantages / limites

- Accélère la convergence.
- Problème de "neurones morts" quand $x < 0$ de manière permanente.

Fonction d'activation ReLU

Définition

$$\text{ReLU}(x) = \max(0, x)$$

- Non dérivable en $x = 0$, mais utilisée massivement en pratique.
- Simple, computationnellement efficace.
- Image : $[0, +\infty)$

Avantages / limites

- Accélère la convergence.
- Problème de “neurones morts” quand $x < 0$ de manière permanente.

Fonction GELU

Définition

$$\text{GELU}(x) = x \cdot \Phi(x), \quad \text{où } \Phi(x) = \frac{1}{2} \left[1 + \text{erf} \left(\frac{x}{\sqrt{2}} \right) \right]$$

- $\Phi(x)$ est la fonction de répartition de la loi normale.
- Fonction lisse, proche de ReLU mais plus fine statistiquement.

Utilisation avancée

Adoptée dans les Transformers (BERT, GPT-2) car elle combine efficacité computationnelle et régularité du gradient.

Fonction GELU

Définition

$$\text{GELU}(x) = x \cdot \Phi(x), \quad \text{où } \Phi(x) = \frac{1}{2} \left[1 + \text{erf} \left(\frac{x}{\sqrt{2}} \right) \right]$$

- $\Phi(x)$ est la fonction de répartition de la loi normale.
- Fonction lisse, proche de ReLU mais plus fine statistiquement.

Utilisation avancée

Adoptée dans les Transformers (BERT, GPT-2) car elle combine efficacité computationnelle et régularité du gradient.

Continuité d'une fonction réelle

Définition

Une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ est **continue en** $x_0 \in \mathbb{R}$ si :

$$\lim_{x \rightarrow x_0} f(x) = f(x_0)$$

- Intuitivement : pas de "saut", ni de "trou".
- Toute fonction dérivable en un point est continue en ce point (mais la réciproque est fausse).

Conséquence

La continuité assure la stabilité du modèle : petites perturbations d'entrée \Rightarrow petites variations de sortie.

Continuité d'une fonction réelle

Définition

Une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ est **continue en** $x_0 \in \mathbb{R}$ si :

$$\lim_{x \rightarrow x_0} f(x) = f(x_0)$$

- Intuitivement : pas de "saut", ni de "trou".
- Toute fonction dérivable en un point est continue en ce point (mais la réciproque est fausse).

Conséquence

La continuité assure la stabilité du modèle : petites perturbations d'entrée \Rightarrow petites variations de sortie.

Continuité d'une fonction réelle

Définition

Une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ est **continue en** $x_0 \in \mathbb{R}$ si :

$$\lim_{x \rightarrow x_0} f(x) = f(x_0)$$

- Intuitivement : pas de "saut", ni de "trou".
- Toute fonction dérivable en un point est continue en ce point (mais la réciproque est fausse).

Conséquence

La continuité assure la stabilité du modèle : petites perturbations d'entrée \Rightarrow petites variations de sortie.

Dérivabilité : définition et intérêt

Définition

f est **dérivable en** x_0 si la limite suivante existe :

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

- Donne le taux de variation instantané.
- Fonctions usuelles (exp, ln, polynômes, sigmoïde, tanh, GELU) sont dérivables partout.
- ReLU n'est pas dérivable en $x = 0$, mais reste utilisée car presque partout dérivable.

Optimisation par descente de gradient

Idée clé

Minimiser une fonction de coût $J(\theta)$ en ajustant les paramètres θ dans le sens opposé au gradient :

$$\theta^{(t+1)} = \theta^{(t)} - \eta \cdot \nabla_{\theta} J(\theta^{(t)})$$

- $\nabla_{\theta} J$ n'existe que si J est dérivable.
- Les fonctions d'activation doivent donc être dérivables (ou presque partout dérivables).

Importance

La forme de f influence la vitesse et la stabilité de la convergence.

Optimisation par descente de gradient

Idée clé

Minimiser une fonction de coût $J(\theta)$ en ajustant les paramètres θ dans le sens opposé au gradient :

$$\theta^{(t+1)} = \theta^{(t)} - \eta \cdot \nabla_{\theta} J(\theta^{(t)})$$

- $\nabla_{\theta} J$ n'existe que si J est dérivable.
- Les fonctions d'activation doivent donc être dérivables (ou presque partout dérivables).

Importance

La forme de f influence la vitesse et la stabilité de la convergence.

Optimisation par descente de gradient

Idée clé

Minimiser une fonction de coût $J(\theta)$ en ajustant les paramètres θ dans le sens opposé au gradient :

$$\theta^{(t+1)} = \theta^{(t)} - \eta \cdot \nabla_{\theta} J(\theta^{(t)})$$

- $\nabla_{\theta} J$ n'existe que si J est dérivable.
- Les fonctions d'activation doivent donc être dérivables (ou presque partout dérivables).

Importance

La forme de f influence la vitesse et la stabilité de la convergence.

Rétropropagation (Backpropagation)

Principe

Algorithme qui applique la règle de la chaîne pour propager les gradients de la sortie vers l'entrée :

$$\frac{\partial J}{\partial \theta} = \frac{\partial J}{\partial z_n} \cdot \frac{\partial z_n}{\partial z_{n-1}} \cdots \frac{\partial z_1}{\partial \theta}$$

- Chaque fonction utilisée dans le réseau doit être différentiable pour propager l'information.
- Fonctions d'activation choisies pour leur dérivée simple à calculer.

Rétropropagation (Backpropagation)

Principe

Algorithme qui applique la règle de la chaîne pour propager les gradients de la sortie vers l'entrée :

$$\frac{\partial J}{\partial \theta} = \frac{\partial J}{\partial z_n} \cdot \frac{\partial z_n}{\partial z_{n-1}} \cdots \frac{\partial z_1}{\partial \theta}$$

- Chaque fonction utilisée dans le réseau doit être différentiable pour propager l'information.
- Fonctions d'activation choisies pour leur dérivée simple à calculer.

Dérivabilité, expressivité et efficacité

- $\sigma(x)$, $\tanh(x)$: dérivables partout mais saturent \Rightarrow gradients faibles.
- $\text{ReLU}(x)$: non dérivable en 0, mais simple et efficace, introduit de la sparsité.
- $\text{GELU}(x)$: dérivable partout, plus fluide que ReLU.

Compromis

Le choix repose sur un équilibre entre :

- Continuité/dérivabilité
- Coût de calcul
- Propriétés d'apprentissage (vitesse, stabilité)

Dérivabilité, expressivité et efficacité

- $\sigma(x)$, $\tanh(x)$: dérivables partout mais saturent \Rightarrow gradients faibles.
- $\text{ReLU}(x)$: non dérivable en 0, mais simple et efficace, introduit de la sparsité.
- $\text{GELU}(x)$: dérivable partout, plus fluide que ReLU.

Compromis

Le choix repose sur un équilibre entre :

- Continuité/dérivabilité
- Coût de calcul
- Propriétés d'apprentissage (vitesse, stabilité)

Objets de base en algèbre linéaire

- **Scalaire** : un seul nombre réel $a \in \mathbb{R}$.
- **Vecteur** : une liste ordonnée de scalaires

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

- **Matrice** : tableau de scalaires organisés en lignes et colonnes :

$$\mathbf{X} \in \mathbb{R}^{m \times n} \quad (\text{matrice à } m \text{ lignes et } n \text{ colonnes})$$

Notation conventionnelle

- \mathbf{X} : matrice
- \mathbf{x} : vecteur
- x : scalaire

Objets de base en algèbre linéaire

- **Scalaire** : un seul nombre réel $a \in \mathbb{R}$.
- **Vecteur** : une liste ordonnée de scalaires

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

- **Matrice** : tableau de scalaires organisés en lignes et colonnes :

$$\mathbf{X} \in \mathbb{R}^{m \times n} \quad (\text{matrice à } m \text{ lignes et } n \text{ colonnes})$$

Notation conventionnelle

- \mathbf{X} : matrice
- \mathbf{x} : vecteur
- x : scalaire

Scalars et vecteurs

Scalaire : un nombre réel $a \in \mathbb{R}$ (température, poids, coût...)

Vecteur colonne :

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

- Coordonnées : v_i pour $i = 1, \dots, n$
- \mathbb{R}^n est un espace vectoriel de dimension n
- Interprétation : points, directions, poids...

Vecteur ligne : $\mathbf{v}^\top = [v_1, v_2, \dots, v_n] \in \mathbb{R}^{1 \times n}$

Scalaire et vecteurs

Scalaire : un nombre réel $a \in \mathbb{R}$ (température, poids, coût...)

Vecteur colonne :

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

- Coordonnées : v_i pour $i = 1, \dots, n$
- \mathbb{R}^n est un espace vectoriel de dimension n
- Interprétation : points, directions, poids...

Vecteur ligne : $\mathbf{v}^\top = [v_1, v_2, \dots, v_n] \in \mathbb{R}^{1 \times n}$

Matrices : définitions et notations

Matrice $\mathbf{X} \in \mathbb{R}^{m \times n}$:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \dots & x_{m,n} \end{bmatrix}$$

- $x_{i,j}$: élément en ligne i , colonne j
- m = nombre de lignes (exemples)
- n = nombre de colonnes (features)

Exemple

\mathbf{X} peut représenter un batch de données : m exemples, chacun de n dimensions.

Matrices : définitions et notations

Matrice $\mathbf{X} \in \mathbb{R}^{m \times n}$:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \dots & x_{m,n} \end{bmatrix}$$

- $x_{i,j}$: élément en ligne i , colonne j
- m = nombre de lignes (exemples)
- n = nombre de colonnes (features)

Exemple

\mathbf{X} peut représenter un batch de données : m exemples, chacun de n dimensions.

Matrices : définitions et notations

Matrice $\mathbf{X} \in \mathbb{R}^{m \times n}$:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \dots & x_{m,n} \end{bmatrix}$$

- $x_{i,j}$: élément en ligne i , colonne j
- m = nombre de lignes (exemples)
- n = nombre de colonnes (features)

Exemple

\mathbf{X} peut représenter un batch de données : m exemples, chacun de n dimensions.

Interprétation des dimensions en apprentissage

Exemple : classification supervisée

$$\mathbf{X} \in \mathbb{R}^{m \times n}, \quad \mathbf{y} \in \mathbb{R}^{m \times 1}$$

- m : nombre d'exemples (données d'entraînement)
- n : nombre de variables/features
- \mathbf{X} : matrice de design, chaque ligne = un vecteur d'entrée
- \mathbf{y} : vecteur des sorties/étiquettes

Réseaux de neurones

- Poids = matrices \mathbf{W}
- Inputs = vecteurs \mathbf{x}
- Opérations = produits matriciels, compositions non linéaires

Interprétation des dimensions en apprentissage

Exemple : classification supervisée

$$\mathbf{X} \in \mathbb{R}^{m \times n}, \quad \mathbf{y} \in \mathbb{R}^{m \times 1}$$

- m : nombre d'exemples (données d'entraînement)
- n : nombre de variables/features
- \mathbf{X} : matrice de design, chaque ligne = un vecteur d'entrée
- \mathbf{y} : vecteur des sorties/étiquettes

Réseaux de neurones

- Poids = matrices \mathbf{W}
- Inputs = vecteurs \mathbf{x}
- Opérations = produits matriciels, compositions non linéaires

Interprétation des dimensions en apprentissage

Exemple : classification supervisée

$$\mathbf{X} \in \mathbb{R}^{m \times n}, \quad \mathbf{y} \in \mathbb{R}^{m \times 1}$$

- m : nombre d'exemples (données d'entraînement)
- n : nombre de variables/features
- \mathbf{X} : matrice de design, chaque ligne = un vecteur d'entrée
- \mathbf{y} : vecteur des sorties/étiquettes

Réseaux de neurones

- Poids = matrices \mathbf{W}
- Inputs = vecteurs \mathbf{x}
- Opérations = produits matriciels, compositions non linéaires

Addition et transposition

Addition de matrices : $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$

$$\mathbf{C} = \mathbf{A} + \mathbf{B} \Rightarrow c_{ij} = a_{ij} + b_{ij}$$

Transposée d'une matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$:

$$\mathbf{A}^\top \in \mathbb{R}^{n \times m}, \quad (\mathbf{A}^\top)_{ij} = a_{ji}$$

Propriétés

- $(\mathbf{A} + \mathbf{B})^\top = \mathbf{A}^\top + \mathbf{B}^\top$
- $(\mathbf{A}^\top)^\top = \mathbf{A}$

Addition et transposition

Addition de matrices : $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$

$$\mathbf{C} = \mathbf{A} + \mathbf{B} \Rightarrow c_{ij} = a_{ij} + b_{ij}$$

Transposée d'une matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$:

$$\mathbf{A}^\top \in \mathbb{R}^{n \times m}, \quad (\mathbf{A}^\top)_{ij} = a_{ji}$$

Propriétés

- $(\mathbf{A} + \mathbf{B})^\top = \mathbf{A}^\top + \mathbf{B}^\top$
- $(\mathbf{A}^\top)^\top = \mathbf{A}$

Addition et transposition

Addition de matrices : $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$

$$\mathbf{C} = \mathbf{A} + \mathbf{B} \Rightarrow c_{ij} = a_{ij} + b_{ij}$$

Transposée d'une matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$:

$$\mathbf{A}^\top \in \mathbb{R}^{n \times m}, \quad (\mathbf{A}^\top)_{ij} = a_{ji}$$

Propriétés

- $(\mathbf{A} + \mathbf{B})^\top = \mathbf{A}^\top + \mathbf{B}^\top$
- $(\mathbf{A}^\top)^\top = \mathbf{A}$

Produit scalaire

Produit scalaire de deux vecteurs $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$:

$$\mathbf{u}^\top \mathbf{v} = \sum_{i=1}^n u_i v_i \in \mathbb{R}$$

- Résultat : scalaire
- Mesure l'alignement (cosinus de l'angle entre les vecteurs)

Propriétés

- Symétrie : $\mathbf{u}^\top \mathbf{v} = \mathbf{v}^\top \mathbf{u}$
- Linéarité : $\mathbf{u}^\top (a\mathbf{v} + b\mathbf{w}) = a\mathbf{u}^\top \mathbf{v} + b\mathbf{u}^\top \mathbf{w}$

Produit scalaire

Produit scalaire de deux vecteurs $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$:

$$\mathbf{u}^\top \mathbf{v} = \sum_{i=1}^n u_i v_i \in \mathbb{R}$$

- Résultat : scalaire
- Mesure l'alignement (cosinus de l'angle entre les vecteurs)

Propriétés

- Symétrie : $\mathbf{u}^\top \mathbf{v} = \mathbf{v}^\top \mathbf{u}$
- Linéarité : $\mathbf{u}^\top (a\mathbf{v} + b\mathbf{w}) = a\mathbf{u}^\top \mathbf{v} + b\mathbf{u}^\top \mathbf{w}$

Produit scalaire

Produit scalaire de deux vecteurs $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$:

$$\mathbf{u}^\top \mathbf{v} = \sum_{i=1}^n u_i v_i \in \mathbb{R}$$

- Résultat : scalaire
- Mesure l'alignement (cosinus de l'angle entre les vecteurs)

Propriétés

- Symétrie : $\mathbf{u}^\top \mathbf{v} = \mathbf{v}^\top \mathbf{u}$
- Linéarité : $\mathbf{u}^\top (a\mathbf{v} + b\mathbf{w}) = a\mathbf{u}^\top \mathbf{v} + b\mathbf{u}^\top \mathbf{w}$

Produit matriciel

Produit matriciel :

$$\mathbf{C} = \mathbf{AB}, \quad \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{B} \in \mathbb{R}^{n \times p} \Rightarrow \mathbf{C} \in \mathbb{R}^{m \times p}$$

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

Interprétation :

- Composition de transformations linéaires
- Produit de couches dans les réseaux de neurones

Produit matriciel

Produit matriciel :

$$\mathbf{C} = \mathbf{AB}, \quad \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{B} \in \mathbb{R}^{n \times p} \Rightarrow \mathbf{C} \in \mathbb{R}^{m \times p}$$

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

Interprétation :

- Composition de transformations linéaires
- Produit de couches dans les réseaux de neurones

Produit matriciel

Produit matriciel :

$$\mathbf{C} = \mathbf{AB}, \quad \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{B} \in \mathbb{R}^{n \times p} \Rightarrow \mathbf{C} \in \mathbb{R}^{m \times p}$$

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

Interprétation :

- Composition de transformations linéaires
- Produit de couches dans les réseaux de neurones

Inversibilité

Définition : Une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$ est **inversible** s'il existe \mathbf{A}^{-1} tel que :

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

- \mathbf{A} est alors dite **non singulière**
- Sinon, elle est **singulière** (non inversible)

Conditions d'inversibilité :

- $\det(\mathbf{A}) \neq 0$
- Les colonnes sont linéairement indépendantes

Inversibilité

Définition : Une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$ est **inversible** s'il existe \mathbf{A}^{-1} tel que :

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

- \mathbf{A} est alors dite **non singulière**
- Sinon, elle est **singulière** (non inversible)

Conditions d'inversibilité :

- $\det(\mathbf{A}) \neq 0$
- Les colonnes sont linéairement indépendantes

Inversibilité

Définition : Une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$ est **inversible** s'il existe \mathbf{A}^{-1} tel que :

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

- \mathbf{A} est alors dite **non singulière**
- Sinon, elle est **singulière** (non inversible)

Conditions d'inversibilité :

- $\det(\mathbf{A}) \neq 0$
- Les colonnes sont linéairement indépendantes

Trace d'une matrice

Trace d'une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$:

$$\text{tr}(\mathbf{A}) = \sum_{i=1}^n a_{ii}$$

- Somme des éléments diagonaux
- Invariante par changement de base
- $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$ si les produits sont définis

Applications :

- En statistiques : trace = somme des variances (matrice de covariance)
- En apprentissage : régularisation par la trace

Trace d'une matrice

Trace d'une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$:

$$\text{tr}(\mathbf{A}) = \sum_{i=1}^n a_{ii}$$

- Somme des éléments diagonaux
- Invariante par changement de base
- $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$ si les produits sont définis

Applications :

- En statistiques : trace = somme des variances (matrice de covariance)
- En apprentissage : régularisation par la trace

Trace d'une matrice

Trace d'une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$:

$$\text{tr}(\mathbf{A}) = \sum_{i=1}^n a_{ii}$$

- Somme des éléments diagonaux
- Invariante par changement de base
- $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$ si les produits sont définis

Applications :

- En statistiques : trace = somme des variances (matrice de covariance)
- En apprentissage : régularisation par la trace

Déterminant

Déterminant : $\det(\mathbf{A})$, pour une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$

- Donne une mesure de la "taille" du parallélépipède formé par les colonnes
- $\det(\mathbf{A}) = 0 \Leftrightarrow \mathbf{A}$ non inversible
- $\det(\mathbf{AB}) = \det(\mathbf{A}) \cdot \det(\mathbf{B})$
- $\det(\mathbf{A}^\top) = \det(\mathbf{A})$

Cas 2×2 :

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

Polynôme caractéristique, valeurs propres, vecteurs propres, etc..

Déterminant

Déterminant : $\det(\mathbf{A})$, pour une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$

- Donne une mesure de la "taille" du parallélépipède formé par les colonnes
- $\det(\mathbf{A}) = 0 \Leftrightarrow \mathbf{A}$ non inversible
- $\det(\mathbf{AB}) = \det(\mathbf{A}) \cdot \det(\mathbf{B})$
- $\det(\mathbf{A}^\top) = \det(\mathbf{A})$

Cas 2×2 :

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

Polynôme caractéristique, valeurs propres, vecteurs propres, etc..

Déterminant

Déterminant : $\det(\mathbf{A})$, pour une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$

- Donne une mesure de la "taille" du parallélépipède formé par les colonnes
- $\det(\mathbf{A}) = 0 \Leftrightarrow \mathbf{A}$ non inversible
- $\det(\mathbf{AB}) = \det(\mathbf{A}) \cdot \det(\mathbf{B})$
- $\det(\mathbf{A}^\top) = \det(\mathbf{A})$

Cas 2×2 :

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

Polynôme caractéristique, valeurs propres, vecteurs propres, etc..

Déterminant

Déterminant : $\det(\mathbf{A})$, pour une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$

- Donne une mesure de la "taille" du parallélépipède formé par les colonnes
- $\det(\mathbf{A}) = 0 \Leftrightarrow \mathbf{A}$ non inversible
- $\det(\mathbf{AB}) = \det(\mathbf{A}) \cdot \det(\mathbf{B})$
- $\det(\mathbf{A}^\top) = \det(\mathbf{A})$

Cas 2×2 :

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

Polynôme caractéristique, valeurs propres, vecteurs propres, etc..

Matrice identité

Définition : La matrice identité $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ est la matrice carrée telle que :

$$(\mathbf{I}_n)_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{sinon} \end{cases}$$

Propriété fondamentale :

$$\forall \mathbf{A} \in \mathbb{R}^{n \times n}, \quad \mathbf{A}\mathbf{I}_n = \mathbf{I}_n\mathbf{A} = \mathbf{A}$$

Rôle : élément neutre du produit matriciel

Matrice identité

Définition : La matrice identité $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ est la matrice carrée telle que :

$$(\mathbf{I}_n)_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{sinon} \end{cases}$$

Propriété fondamentale :

$$\forall \mathbf{A} \in \mathbb{R}^{n \times n}, \quad \mathbf{A}\mathbf{I}_n = \mathbf{I}_n\mathbf{A} = \mathbf{A}$$

Rôle : élément neutre du produit matriciel

Matrice identité

Définition : La matrice identité $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ est la matrice carrée telle que :

$$(\mathbf{I}_n)_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{sinon} \end{cases}$$

Propriété fondamentale :

$$\forall \mathbf{A} \in \mathbb{R}^{n \times n}, \quad \mathbf{A}\mathbf{I}_n = \mathbf{I}_n\mathbf{A} = \mathbf{A}$$

Rôle : élément neutre du produit matriciel

Matrice diagonale

Définition : Une matrice $\mathbf{D} \in \mathbb{R}^{n \times n}$ est dite **diagonale** si :

$$d_{ij} = 0 \quad \text{pour } i \neq j$$

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}$$

Propriétés :

- Facile à inverser si $d_i \neq 0$
- $\det(\mathbf{D}) = \prod_{i=1}^n d_i$
- $\mathbf{D}^k = \text{diag}(d_1^k, \dots, d_n^k)$

Matrice diagonale

Définition : Une matrice $\mathbf{D} \in \mathbb{R}^{n \times n}$ est dite **diagonale** si :

$$d_{ij} = 0 \quad \text{pour } i \neq j$$

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}$$

Propriétés :

- Facile à inverser si $d_i \neq 0$
- $\det(\mathbf{D}) = \prod_{i=1}^n d_i$
- $\mathbf{D}^k = \text{diag}(d_1^k, \dots, d_n^k)$

Matrice diagonale

Définition : Une matrice $\mathbf{D} \in \mathbb{R}^{n \times n}$ est dite **diagonale** si :

$$d_{ij} = 0 \quad \text{pour } i \neq j$$

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}$$

Propriétés :

- Facile à inverser si $d_i \neq 0$
- $\det(\mathbf{D}) = \prod_{i=1}^n d_i$
- $\mathbf{D}^k = \text{diag}(d_1^k, \dots, d_n^k)$

Matrice symétrique

Définition : Une matrice $\mathbf{A} \in \mathbb{R}^{n \times n}$ est **symétrique** si :

$$\mathbf{A}^\top = \mathbf{A}$$

Propriétés :

- Les éléments diagonaux sont réels.
- $\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x}^\top \mathbf{A} \mathbf{x} \in \mathbb{R}$
- Admet des valeurs propres réelles
- Diagonalisable dans une base orthonormale

Applications : matrices de covariance, Hessienne

Matrice symétrique

Définition : Une matrice $\mathbf{A} \in \mathbb{R}^{n \times n}$ est **symétrique** si :

$$\mathbf{A}^\top = \mathbf{A}$$

Propriétés :

- Les éléments diagonaux sont réels.
- $\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x}^\top \mathbf{A} \mathbf{x} \in \mathbb{R}$
- Admet des valeurs propres réelles
- Diagonalisable dans une base orthonormale

Applications : matrices de covariance, Hessienne

Matrice symétrique

Définition : Une matrice $\mathbf{A} \in \mathbb{R}^{n \times n}$ est **symétrique** si :

$$\mathbf{A}^\top = \mathbf{A}$$

Propriétés :

- Les éléments diagonaux sont réels.
- $\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x}^\top \mathbf{A} \mathbf{x} \in \mathbb{R}$
- Admet des valeurs propres réelles
- Diagonalisable dans une base orthonormale

Applications : matrices de covariance, Hessienne

Matrice orthogonale

Définition : $\mathbf{Q} \in \mathbb{R}^{n \times n}$ est **orthogonale** si :

$$\mathbf{Q}^T \mathbf{Q} = \mathbf{Q} \mathbf{Q}^T = \mathbf{I}_n$$

Propriétés :

- $\mathbf{Q}^{-1} = \mathbf{Q}^T$
- Conserve les normes : $\|\mathbf{Q}\mathbf{x}\| = \|\mathbf{x}\|$
- Produit de vecteurs orthonormés

Applications :

- Transformations orthogonales (rotations, réflexions)
- Décompositions QR, PCA

Matrice orthogonale

Définition : $\mathbf{Q} \in \mathbb{R}^{n \times n}$ est **orthogonale** si :

$$\mathbf{Q}^T \mathbf{Q} = \mathbf{Q} \mathbf{Q}^T = \mathbf{I}_n$$

Propriétés :

- $\mathbf{Q}^{-1} = \mathbf{Q}^T$
- Conserve les normes : $\|\mathbf{Q}\mathbf{x}\| = \|\mathbf{x}\|$
- Produit de vecteurs orthonormés

Applications :

- Transformations orthogonales (rotations, réflexions)
- Décompositions QR, PCA

Matrice orthogonale

Définition : $\mathbf{Q} \in \mathbb{R}^{n \times n}$ est **orthogonale** si :

$$\mathbf{Q}^\top \mathbf{Q} = \mathbf{Q} \mathbf{Q}^\top = \mathbf{I}_n$$

Propriétés :

- $\mathbf{Q}^{-1} = \mathbf{Q}^\top$
- Conserve les normes : $\|\mathbf{Q}\mathbf{x}\| = \|\mathbf{x}\|$
- Produit de vecteurs orthonormés

Applications :

- Transformations orthogonales (rotations, réflexions)
- Décompositions QR, PCA

Comparaison des matrices spéciales

Type	Définition	Propriétés clés
Identité	$I_{ij} = \delta_{ij}$	Neutre pour le produit
Diagonale	$a_{ij} = 0$ si $i \neq j$	Facile à inverser
Symétrique	$\mathbf{A}^\top = \mathbf{A}$	Valeurs propres réelles
Orthogonale	$\mathbf{Q}^\top = \mathbf{Q}^{-1}$	Norme conservée

Applications en apprentissage automatique

- **Identité** : poids initiaux, régularisation (ex : ridge \mathbf{I}_n)
- **Matrices diagonales** : simplifie le calcul des gradients, jacobiens diagonaux
- **Symétriques** : matrices de covariance, hessienne
- **Orthogonales** :
 - ▶ Initialisation des réseaux (orthogonal init)
 - ▶ RNNs stables (préservent norme des vecteurs)

Conclusion : La structure d'une matrice a un impact direct sur la stabilité numérique, l'interprétabilité, et la convergence des algorithmes.

Applications en apprentissage automatique

- **Identité** : poids initiaux, régularisation (ex : ridge \mathbf{I}_n)
- **Matrices diagonales** : simplifie le calcul des gradients, jacobiens diagonaux
- **Symétriques** : matrices de covariance, hessienne
- **Orthogonales** :
 - ▶ Initialisation des réseaux (orthogonal init)
 - ▶ RNNs stables (préservent norme des vecteurs)

Conclusion : La structure d'une matrice a un impact direct sur la stabilité numérique, l'interprétabilité, et la convergence des algorithmes.

Multiplication matrice \times vecteur

Soit une matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ et un vecteur colonne $\mathbf{x} \in \mathbb{R}^n$.

Produit :

$$\mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{R}^m$$

Formule explicite :

$$y_i = \sum_{j=1}^n a_{ij}x_j, \quad \text{pour } i = 1, \dots, m$$

Interprétation : combinaison linéaire des colonnes de \mathbf{A} pondérées par les coordonnées de \mathbf{x} .

Notation standard en Deep Learning : $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$

Multiplication matrice \times vecteur

Soit une matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ et un vecteur colonne $\mathbf{x} \in \mathbb{R}^n$.

Produit :

$$\mathbf{y} = \mathbf{Ax} \in \mathbb{R}^m$$

Formule explicite :

$$y_i = \sum_{j=1}^n a_{ij}x_j, \quad \text{pour } i = 1, \dots, m$$

Interprétation : combinaison linéaire des colonnes de \mathbf{A} pondérées par les coordonnées de \mathbf{x} .

Notation standard en Deep Learning : $\mathbf{y} = \mathbf{Wx} + \mathbf{b}$

Multiplication matrice \times vecteur

Soit une matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ et un vecteur colonne $\mathbf{x} \in \mathbb{R}^n$.

Produit :

$$\mathbf{y} = \mathbf{Ax} \in \mathbb{R}^m$$

Formule explicite :

$$y_i = \sum_{j=1}^n a_{ij}x_j, \quad \text{pour } i = 1, \dots, m$$

Interprétation : combinaison linéaire des colonnes de \mathbf{A} pondérées par les coordonnées de \mathbf{x} .

Notation standard en Deep Learning : $\mathbf{y} = \mathbf{Wx} + \mathbf{b}$

Multiplication matrice \times vecteur

Soit une matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ et un vecteur colonne $\mathbf{x} \in \mathbb{R}^n$.

Produit :

$$\mathbf{y} = \mathbf{Ax} \in \mathbb{R}^m$$

Formule explicite :

$$y_i = \sum_{j=1}^n a_{ij}x_j, \quad \text{pour } i = 1, \dots, m$$

Interprétation : combinaison linéaire des colonnes de \mathbf{A} pondérées par les coordonnées de \mathbf{x} .

Notation standard en Deep Learning : $\mathbf{y} = \mathbf{Wx} + \mathbf{b}$

Interprétation géométrique

L'application linéaire $x \mapsto Ax$ est une **transformation de l'espace**.

- $A \in \mathbb{R}^{n \times n}$ peut :
 - ▶ faire une **rotation**
 - ▶ une **dilatation**
 - ▶ une **réflexion**
 - ▶ une **projection**
- Si A n'est pas carrée : transformation entre espaces de dimension différente.

Exemple :

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \text{dilatation selon l'axe } x$$

Illustration visuelle : le vecteur x est "déformé" par A dans un nouvel espace.

Interprétation géométrique

L'application linéaire $x \mapsto Ax$ est une **transformation de l'espace**.

- $A \in \mathbb{R}^{n \times n}$ peut :
 - ▶ faire une **rotation**
 - ▶ une **dilatation**
 - ▶ une **réflexion**
 - ▶ une **projection**
- Si A n'est pas carrée : transformation entre espaces de dimension différente.

Exemple :

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \text{dilatation selon l'axe } x$$

Illustration visuelle : le vecteur x est "déformé" par A dans un nouvel espace.

Interprétation géométrique

L'application linéaire $x \mapsto Ax$ est une **transformation de l'espace**.

- $A \in \mathbb{R}^{n \times n}$ peut :
 - ▶ faire une **rotation**
 - ▶ une **dilatation**
 - ▶ une **réflexion**
 - ▶ une **projection**
- Si A n'est pas carrée : transformation entre espaces de dimension différente.

Exemple :

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \text{dilatation selon l'axe } x$$

Illustration visuelle : le vecteur x est "déformé" par A dans un nouvel espace.

Interprétation géométrique

L'application linéaire $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ est une **transformation de l'espace**.

- $\mathbf{A} \in \mathbb{R}^{n \times n}$ peut :
 - ▶ faire une **rotation**
 - ▶ une **dilatation**
 - ▶ une **réflexion**
 - ▶ une **projection**
- Si \mathbf{A} n'est pas carrée : transformation entre espaces de dimension différente.

Transformation d'un espace de dimension n vers m

Matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$: transforme $\mathbf{x} \in \mathbb{R}^n$ en un vecteur $\mathbf{y} \in \mathbb{R}^m$

Exemple 1 : compression ($n = 5, m = 2$) :

$\mathbf{A} \in \mathbb{R}^{2 \times 5}$, projection d'un espace de haute dimension vers un plan

Exemple 2 : expansion ($n = 2, m = 4$) :

$\mathbf{A} \in \mathbb{R}^{4 \times 2}$, immersion d'un plan dans un espace 4D

Usage : représentation des données, réduction de dimension, reconstruction.

Transformation d'un espace de dimension n vers m

Matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$: transforme $\mathbf{x} \in \mathbb{R}^n$ en un vecteur $\mathbf{y} \in \mathbb{R}^m$

Exemple 1 : compression ($n = 5, m = 2$) :

$\mathbf{A} \in \mathbb{R}^{2 \times 5}$, projection d'un espace de haute dimension vers un plan

Exemple 2 : expansion ($n = 2, m = 4$) :

$\mathbf{A} \in \mathbb{R}^{4 \times 2}$, immersion d'un plan dans un espace 4D

Usage : représentation des données, réduction de dimension, reconstruction.

Transformation d'un espace de dimension n vers m

Matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$: transforme $\mathbf{x} \in \mathbb{R}^n$ en un vecteur $\mathbf{y} \in \mathbb{R}^m$

Exemple 1 : compression ($n = 5, m = 2$) :

$\mathbf{A} \in \mathbb{R}^{2 \times 5}$, projection d'un espace de haute dimension vers un plan

Exemple 2 : expansion ($n = 2, m = 4$) :

$\mathbf{A} \in \mathbb{R}^{4 \times 2}$, immersion d'un plan dans un espace 4D

Usage : représentation des données, réduction de dimension, reconstruction.

Transformation d'un espace de dimension n vers m

Matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$: transforme $\mathbf{x} \in \mathbb{R}^n$ en un vecteur $\mathbf{y} \in \mathbb{R}^m$

Exemple 1 : compression ($n = 5, m = 2$) :

$\mathbf{A} \in \mathbb{R}^{2 \times 5}$, projection d'un espace de haute dimension vers un plan

Exemple 2 : expansion ($n = 2, m = 4$) :

$\mathbf{A} \in \mathbb{R}^{4 \times 2}$, immersion d'un plan dans un espace 4D

Usage : représentation des données, réduction de dimension, reconstruction.

Couches linéaires dans un réseau de neurones

Opération fondamentale :

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

Où :

- $\mathbf{x} \in \mathbb{R}^n$: entrée (features)
- $\mathbf{W} \in \mathbb{R}^{m \times n}$: poids de la couche
- $\mathbf{b} \in \mathbb{R}^m$: biais
- $\mathbf{y} \in \mathbb{R}^m$: sortie (logits ou activation)

But : apprendre \mathbf{W} , \mathbf{b} pour approximer des fonctions non linéaires via combinaisons linéaires + activation

Couches linéaires dans un réseau de neurones

Opération fondamentale :

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

Où :

- $\mathbf{x} \in \mathbb{R}^n$: entrée (features)
- $\mathbf{W} \in \mathbb{R}^{m \times n}$: poids de la couche
- $\mathbf{b} \in \mathbb{R}^m$: biais
- $\mathbf{y} \in \mathbb{R}^m$: sortie (logits ou activation)

But : apprendre \mathbf{W}, \mathbf{b} pour approximer des fonctions non linéaires via combinaisons linéaires + activation

Couches linéaires dans un réseau de neurones

Opération fondamentale :

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

Où :

- $\mathbf{x} \in \mathbb{R}^n$: entrée (features)
- $\mathbf{W} \in \mathbb{R}^{m \times n}$: poids de la couche
- $\mathbf{b} \in \mathbb{R}^m$: biais
- $\mathbf{y} \in \mathbb{R}^m$: sortie (logits ou activation)

But : apprendre \mathbf{W}, \mathbf{b} pour approximer des fonctions non linéaires via combinaisons linéaires + activation

Intuition en apprentissage profond

Chaque couche linéaire transforme les données :

$$\text{Input } \mathbf{x} \xrightarrow{\mathbf{W}\mathbf{x}+\mathbf{b}} \text{Espace latent } \mathbf{y}$$

Rôle :

- Encoder l'information dans un autre espace
- Préparer les données pour les non-linéarités (ReLU, Tanh, etc.)
- Construire progressivement des représentations complexes

Remarque :

- Sans multiplication matrice/vecteur, il n'y a pas de capacité d'apprentissage !
- Le gradient (via rétropropagation) est calculé directement sur \mathbf{W} et \mathbf{b} .

Intuition en apprentissage profond

Chaque couche linéaire transforme les données :

$$\text{Input } \mathbf{x} \xrightarrow{\mathbf{W}\mathbf{x}+\mathbf{b}} \text{Espace latent } \mathbf{y}$$

Rôle :

- Encoder l'information dans un autre espace
- Préparer les données pour les non-linéarités (ReLU, Tanh, etc.)
- Construire progressivement des représentations complexes

Remarque :

- Sans multiplication matrice/vecteur, il n'y a pas de capacité d'apprentissage !
- Le gradient (via rétropropagation) est calculé directement sur \mathbf{W} et \mathbf{b} .

Intuition en apprentissage profond

Chaque couche linéaire transforme les données :

$$\text{Input } \mathbf{x} \xrightarrow{\mathbf{W}\mathbf{x}+\mathbf{b}} \text{Espace latent } \mathbf{y}$$

Rôle :

- Encoder l'information dans un autre espace
- Préparer les données pour les non-linéarités (ReLU, Tanh, etc.)
- Construire progressivement des représentations complexes

Remarque :

- Sans multiplication matrice/vecteur, il n'y a pas de capacité d'apprentissage !
- Le gradient (via rétropropagation) est calculé directement sur \mathbf{W} et \mathbf{b} .

Espaces vectoriels : définition formelle

Définition : Un espace vectoriel V sur un corps \mathbb{K} (souvent \mathbb{R} ou \mathbb{C}) est un ensemble muni de deux opérations :

- Addition vectorielle : $+: V \times V \rightarrow V$
- Multiplication scalaire : $\cdot: \mathbb{K} \times V \rightarrow V$

Ces opérations doivent satisfaire les 8 axiomes suivants (associativité, commutativité, neutres, etc.).

Exemples :

- \mathbb{R}^n avec addition et multiplication scalaire usuelles
- Ensemble des fonctions continues sur $[a, b]$

Espaces vectoriels : définition formelle

Définition : Un espace vectoriel V sur un corps \mathbb{K} (souvent \mathbb{R} ou \mathbb{C}) est un ensemble muni de deux opérations :

- Addition vectorielle : $+: V \times V \rightarrow V$
- Multiplication scalaire : $\cdot: \mathbb{K} \times V \rightarrow V$

Ces opérations doivent satisfaire les 8 axiomes suivants (associativité, commutativité, neutres, etc.).

Exemples :

- \mathbb{R}^n avec addition et multiplication scalaire usuelles
- Ensemble des fonctions continues sur $[a, b]$

Espaces vectoriels : définition formelle

Définition : Un espace vectoriel V sur un corps \mathbb{K} (souvent \mathbb{R} ou \mathbb{C}) est un ensemble muni de deux opérations :

- Addition vectorielle : $+: V \times V \rightarrow V$
- Multiplication scalaire : $\cdot: \mathbb{K} \times V \rightarrow V$

Ces opérations doivent satisfaire les 8 axiomes suivants (associativité, commutativité, neutres, etc.).

Exemples :

- \mathbb{R}^n avec addition et multiplication scalaire usuelles
- Ensemble des fonctions continues sur $[a, b]$

Combinaisons linéaires

Définition : Soient $\mathbf{v}_1, \dots, \mathbf{v}_k \in V$ et $\lambda_1, \dots, \lambda_k \in \mathbb{K}$.

La combinaison linéaire :

$$\mathbf{w} = \lambda_1 \mathbf{v}_1 + \dots + \lambda_k \mathbf{v}_k$$

est un élément de V .

En Deep Learning :

- Les couches linéaires produisent des combinaisons linéaires d'entrées pondérées.
- L'espace engendré par un ensemble de vecteurs est l'ensemble de toutes les combinaisons linéaires possibles.

Combinaisons linéaires

Définition : Soient $\mathbf{v}_1, \dots, \mathbf{v}_k \in V$ et $\lambda_1, \dots, \lambda_k \in \mathbb{K}$.

La combinaison linéaire :

$$\mathbf{w} = \lambda_1 \mathbf{v}_1 + \dots + \lambda_k \mathbf{v}_k$$

est un élément de V .

En Deep Learning :

- Les couches linéaires produisent des combinaisons linéaires d'entrées pondérées.
- L'espace engendré par un ensemble de vecteurs est l'ensemble de toutes les combinaisons linéaires possibles.

Combinaisons linéaires

Définition : Soient $\mathbf{v}_1, \dots, \mathbf{v}_k \in V$ et $\lambda_1, \dots, \lambda_k \in \mathbb{K}$.

La combinaison linéaire :

$$\mathbf{w} = \lambda_1 \mathbf{v}_1 + \dots + \lambda_k \mathbf{v}_k$$

est un élément de V .

En Deep Learning :

- Les couches linéaires produisent des combinaisons linéaires d'entrées pondérées.
- L'espace engendré par un ensemble de vecteurs est l'ensemble de toutes les combinaisons linéaires possibles.

Familles libres et génératrices

Famille génératrice : Un ensemble $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ est générateur de V si tout vecteur de V est combinaison linéaire de ces vecteurs.

Famille libre : Aucune combinaison linéaire non triviale des vecteurs ne donne le vecteur nul :

$$\lambda_1 \mathbf{v}_1 + \dots + \lambda_k \mathbf{v}_k = \mathbf{0} \Rightarrow \lambda_i = 0 \ \forall i$$

Une base est une famille libre et génératrice de V .

Familles libres et génératrices

Famille génératrice : Un ensemble $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ est générateur de V si tout vecteur de V est combinaison linéaire de ces vecteurs.

Famille libre : Aucune combinaison linéaire non triviale des vecteurs ne donne le vecteur nul :

$$\lambda_1 \mathbf{v}_1 + \dots + \lambda_k \mathbf{v}_k = \mathbf{0} \Rightarrow \lambda_i = 0 \quad \forall i$$

Une **base** est une famille libre et génératrice de V .

Familles libres et génératrices

Famille génératrice : Un ensemble $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ est générateur de V si tout vecteur de V est combinaison linéaire de ces vecteurs.

Famille libre : Aucune combinaison linéaire non triviale des vecteurs ne donne le vecteur nul :

$$\lambda_1 \mathbf{v}_1 + \dots + \lambda_k \mathbf{v}_k = \mathbf{0} \Rightarrow \lambda_i = 0 \quad \forall i$$

Une base est une famille libre et génératrice de V .

Base d'un espace vectoriel

Définition : Une base \mathcal{B} de V est un ensemble de vecteurs tel que :

- \mathcal{B} est libre
- \mathcal{B} engendre V

Exemple dans \mathbb{R}^3 :

$$\mathcal{B} = \left\{ \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

Tout vecteur $\mathbf{v} \in \mathbb{R}^3$ peut s'écrire de manière unique :

$$\mathbf{v} = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$$

Base d'un espace vectoriel

Définition : Une base \mathcal{B} de V est un ensemble de vecteurs tel que :

- \mathcal{B} est libre
- \mathcal{B} engendre V

Exemple dans \mathbb{R}^3 :

$$\mathcal{B} = \left\{ \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

Tout vecteur $\mathbf{v} \in \mathbb{R}^3$ peut s'écrire de manière unique :

$$\mathbf{v} = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$$

Base d'un espace vectoriel

Définition : Une base \mathcal{B} de V est un ensemble de vecteurs tel que :

- \mathcal{B} est libre
- \mathcal{B} engendre V

Exemple dans \mathbb{R}^3 :

$$\mathcal{B} = \left\{ \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

Tout vecteur $\mathbf{v} \in \mathbb{R}^3$ peut s'écrire de manière unique :

$$\mathbf{v} = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$$

Dimension d'un espace vectoriel

Définition : La dimension de V , notée $\dim(V)$, est le nombre de vecteurs dans une base de V .

Exemples :

- $\dim(\mathbb{R}^n) = n$
- L'ensemble des polynômes de degré $\leq n$ a pour dimension $n + 1$
- L'espace des matrices $m \times n$ a pour dimension mn

Propriété : Toutes les bases d'un espace vectoriel ont le même nombre de vecteurs.

Dimension d'un espace vectoriel

Définition : La dimension de V , notée $\dim(V)$, est le nombre de vecteurs dans une base de V .

Exemples :

- $\dim(\mathbb{R}^n) = n$
- L'ensemble des polynômes de degré $\leq n$ a pour dimension $n + 1$
- L'espace des matrices $m \times n$ a pour dimension mn

Propriété : Toutes les bases d'un espace vectoriel ont le même nombre de vecteurs.

Dimension d'un espace vectoriel

Définition : La dimension de V , notée $\dim(V)$, est le nombre de vecteurs dans une base de V .

Exemples :

- $\dim(\mathbb{R}^n) = n$
- L'ensemble des polynômes de degré $\leq n$ a pour dimension $n + 1$
- L'espace des matrices $m \times n$ a pour dimension mn

Propriété : Toutes les bases d'un espace vectoriel ont le même nombre de vecteurs.

Importance des espaces vectoriels en IA

Pourquoi s'en soucier ?

- Les données (images, sons, textes) sont représentées comme des vecteurs dans \mathbb{R}^n
- Les couches de neurones réalisent des transformations linéaires entre espaces vectoriels
- Les dimensions déterminent la capacité de représentation d'un modèle

Remarque :

- La réduction de dimension (ex : PCA) s'appuie sur ces notions.
- Comprendre les bases permet de visualiser les changements de repères dans les embeddings.

Importance des espaces vectoriels en IA

Pourquoi s'en soucier ?

- Les données (images, sons, textes) sont représentées comme des vecteurs dans \mathbb{R}^n
- Les couches de neurones réalisent des transformations linéaires entre espaces vectoriels
- Les dimensions déterminent la capacité de représentation d'un modèle

Remarque :

- La réduction de dimension (ex : PCA) s'appuie sur ces notions.
- Comprendre les bases permet de visualiser les changements de repères dans les embeddings.

Rang d'une matrice : définition

Définition : Le **rang** d'une matrice $A \in \mathbb{R}^{m \times n}$ est :

- le nombre de lignes (ou colonnes) linéairement indépendantes ;
- la dimension de l'image de l'application linéaire associée à A ;
- le nombre de pivots non nuls dans la forme échelonnée de A .

Notation : $\text{rang}(A)$ ou $\text{rg}(A)$.

Exemple :

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \Rightarrow \text{rang}(A) = 2$$

Rang d'une matrice : définition

Définition : Le **rang** d'une matrice $A \in \mathbb{R}^{m \times n}$ est :

- le nombre de lignes (ou colonnes) linéairement indépendantes ;
- la dimension de l'image de l'application linéaire associée à A ;
- le nombre de pivots non nuls dans la forme échelonnée de A .

Notation : $\text{rang}(A)$ ou $\text{rg}(A)$.

Exemple :

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \Rightarrow \text{rang}(A) = 2$$

Rang d'une matrice : définition

Définition : Le **rang** d'une matrice $A \in \mathbb{R}^{m \times n}$ est :

- le nombre de lignes (ou colonnes) linéairement indépendantes ;
- la dimension de l'image de l'application linéaire associée à A ;
- le nombre de pivots non nuls dans la forme échelonnée de A .

Notation : $\text{rang}(A)$ ou $\text{rg}(A)$.

Exemple :

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \Rightarrow \text{rang}(A) = 2$$

Systèmes d'équations linéaires

Un système linéaire s'écrit sous la forme :

$$A\mathbf{x} = \mathbf{b} \quad \text{où } A \in \mathbb{R}^{m \times n}, \mathbf{x} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m$$

Classification selon le rang :

- **Unique solution** : $\text{rang}(A) = \text{rang}([A \mid \mathbf{b}]) = n$
- **Infinité de solutions** : $\text{rang}(A) = \text{rang}([A \mid \mathbf{b}]) < n$
- **Aucune solution** : $\text{rang}(A) < \text{rang}([A \mid \mathbf{b}])$

Systèmes d'équations linéaires

Un système linéaire s'écrit sous la forme :

$$A\mathbf{x} = \mathbf{b} \quad \text{où } A \in \mathbb{R}^{m \times n}, \mathbf{x} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m$$

Classification selon le rang :

- **Unique solution** : $\text{rang}(A) = \text{rang}([A \mid \mathbf{b}]) = n$
- **Infinité de solutions** : $\text{rang}(A) = \text{rang}([A \mid \mathbf{b}]) < n$
- **Aucune solution** : $\text{rang}(A) < \text{rang}([A \mid \mathbf{b}])$

Méthode de Gauss : principe

La méthode de Gauss (ou élimination de Gauss) consiste à :

- ➊ Réduire le système à une forme triangulaire (forme échelonnée) ;
- ➋ Résoudre par substitution arrière.

Opérations autorisées (opérations élémentaires) :

- Permutation de lignes ;
- Multiplication d'une ligne par un scalaire non nul ;
- Ajout d'un multiple d'une ligne à une autre.

But : Identifier les pivots \Rightarrow déterminer le rang.

Méthode de Gauss : principe

La méthode de Gauss (ou élimination de Gauss) consiste à :

- ➊ Réduire le système à une forme triangulaire (forme échelonnée) ;
- ➋ Résoudre par substitution arrière.

Opérations autorisées (opérations élémentaires) :

- Permutation de lignes ;
- Multiplication d'une ligne par un scalaire non nul ;
- Ajout d'un multiple d'une ligne à une autre.

But : Identifier les pivots \Rightarrow déterminer le rang.

Méthode de Gauss : principe

La méthode de Gauss (ou élimination de Gauss) consiste à :

- ➊ Réduire le système à une forme triangulaire (forme échelonnée) ;
- ➋ Résoudre par substitution arrière.

Opérations autorisées (opérations élémentaires) :

- Permutation de lignes ;
- Multiplication d'une ligne par un scalaire non nul ;
- Ajout d'un multiple d'une ligne à une autre.

But : Identifier les pivots \Rightarrow déterminer le rang.

Méthode de Gauss : exemple

Réolvons :

$$\begin{cases} x + y + z = 6 \\ 2x + 3y + z = 14 \\ x + 2y + 3z = 14 \end{cases}$$

Forme matricielle augmentée :

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 2 & 3 & 1 & 14 \\ 1 & 2 & 3 & 14 \end{array} \right]$$

Après élimination (détails à faire au tableau ou en notes) :

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & -1 & 2 \\ 0 & 0 & 1 & 2 \end{array} \right] \Rightarrow (z = 2, y = 4, x = 0)$$

Méthode de Gauss : exemple

Réolvons :

$$\begin{cases} x + y + z = 6 \\ 2x + 3y + z = 14 \\ x + 2y + 3z = 14 \end{cases}$$

Forme matricielle augmentée :

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 2 & 3 & 1 & 14 \\ 1 & 2 & 3 & 14 \end{array} \right]$$

Après élimination (détails à faire au tableau ou en notes) :

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & -1 & 2 \\ 0 & 0 & 1 & 2 \end{array} \right] \Rightarrow (z = 2, y = 4, x = 0)$$

Méthode de Gauss : exemple

Réolvons :

$$\begin{cases} x + y + z = 6 \\ 2x + 3y + z = 14 \\ x + 2y + 3z = 14 \end{cases}$$

Forme matricielle augmentée :

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 2 & 3 & 1 & 14 \\ 1 & 2 & 3 & 14 \end{array} \right]$$

Après élimination (détails à faire au tableau ou en notes) :

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & -1 & 2 \\ 0 & 0 & 1 & 2 \end{array} \right] \Rightarrow (z = 2, y = 4, x = 0)$$

Interprétation géométrique du rang

Cas de 2 ou 3 équations à 2 ou 3 inconnues :

- Chaque équation représente un hyperplan ;
- Le rang représente le nombre de directions indépendantes ;
- **Rang 1** : plans parallèles (ou confondus) \Rightarrow intersection ligne ou vide ;
- **Rang 2 (en 3D)** : intersection en une droite ou un point ;
- **Rang 3 (en 3D)** : intersection unique (point).

En image : intersection de plans en 3D.

Interprétation géométrique du rang

Cas de 2 ou 3 équations à 2 ou 3 inconnues :

- Chaque équation représente un hyperplan ;
- Le rang représente le nombre de directions indépendantes ;
- **Rang 1** : plans parallèles (ou confondus) \Rightarrow intersection ligne ou vide ;
- **Rang 2 (en 3D)** : intersection en une droite ou un point ;
- **Rang 3 (en 3D)** : intersection unique (point).

En image : intersection de plans en 3D.

Interprétation géométrique du rang

Cas de 2 ou 3 équations à 2 ou 3 inconnues :

- Chaque équation représente un hyperplan ;
- Le rang représente le nombre de directions indépendantes ;
- **Rang 1** : plans parallèles (ou confondus) \Rightarrow intersection ligne ou vide ;
- **Rang 2 (en 3D)** : intersection en une droite ou un point ;
- **Rang 3 (en 3D)** : intersection unique (point).

En image : intersection de plans en 3D.

Pourquoi s'intéresser au rang ?

En apprentissage automatique :

- Les données sont représentées par des matrices (features \times échantillons) ;
- Un rang faible indique de la redondance \Rightarrow **réduction de dimension** utile ;
- Le rang est lié à la capacité à inverser ou pseudo-inverser une matrice (A^\dagger) ;
- En réseaux de neurones : vérifier la capacité des couches à capturer des représentations linéaires distinctes.

Conclusion : Le rang est fondamental pour comprendre la structure des données et la stabilité des solutions.

Pourquoi s'intéresser au rang ?

En apprentissage automatique :

- Les données sont représentées par des matrices (features \times échantillons) ;
- Un rang faible indique de la redondance \Rightarrow **réduction de dimension** utile ;
- Le rang est lié à la capacité à inverser ou pseudo-inverser une matrice (A^\dagger) ;
- En réseaux de neurones : vérifier la capacité des couches à capturer des représentations linéaires distinctes.

Conclusion : Le rang est fondamental pour comprendre la structure des données et la stabilité des solutions.

Changement de base : motivation

Pourquoi changer de base ?

- Pour simplifier les calculs (ex. base orthonormée) ;
- Pour exprimer un vecteur dans un nouveau repère plus adapté au problème ;
- Pour compresser l'information ou réduire la dimension.

Définition : Soit $\mathcal{B} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ une base de \mathbb{R}^n . Tout vecteur $\mathbf{x} \in \mathbb{R}^n$ s'écrit de manière unique :

$$\mathbf{x} = a_1 \mathbf{v}_1 + \dots + a_n \mathbf{v}_n$$

Les coordonnées (a_1, \dots, a_n) sont les **coordonnées de \mathbf{x} dans la base \mathcal{B}** .

Changement de base : motivation

Pourquoi changer de base ?

- Pour simplifier les calculs (ex. base orthonormée) ;
- Pour exprimer un vecteur dans un nouveau repère plus adapté au problème ;
- Pour compresser l'information ou réduire la dimension.

Définition : Soit $\mathcal{B} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ une base de \mathbb{R}^n . Tout vecteur $\mathbf{x} \in \mathbb{R}^n$ s'écrit de manière unique :

$$\mathbf{x} = a_1 \mathbf{v}_1 + \dots + a_n \mathbf{v}_n$$

Les coordonnées (a_1, \dots, a_n) sont les **coordonnées de \mathbf{x} dans la base \mathcal{B}** .

Formule de changement de base

Soient \mathcal{B} et \mathcal{B}' deux bases de \mathbb{R}^n .

Soit P la matrice de passage de \mathcal{B}' vers \mathcal{B} :

$$[\mathbf{x}]_{\mathcal{B}} = P \cdot [\mathbf{x}]_{\mathcal{B}'}$$

Interprétation : La matrice P est formée par les vecteurs de \mathcal{B}' exprimés dans la base \mathcal{B} :

$$P = \begin{bmatrix} | & & | \\ \mathbf{v}'_1 & \cdots & \mathbf{v}'_n \\ | & & | \end{bmatrix}$$

Changement de base très utile si \mathcal{B}' est orthonormale.

Formule de changement de base

Soient \mathcal{B} et \mathcal{B}' deux bases de \mathbb{R}^n .

Soit P la matrice de passage de \mathcal{B}' vers \mathcal{B} :

$$[\mathbf{x}]_{\mathcal{B}} = P \cdot [\mathbf{x}]_{\mathcal{B}'}$$

Interprétation : La matrice P est formée par les vecteurs de \mathcal{B}' exprimés dans la base \mathcal{B} :

$$P = \begin{bmatrix} | & & | \\ \mathbf{v}'_1 & \cdots & \mathbf{v}'_n \\ | & & | \end{bmatrix}$$

Changement de base très utile si \mathcal{B}' est orthonormale.

Formule de changement de base

Soient \mathcal{B} et \mathcal{B}' deux bases de \mathbb{R}^n .

Soit P la matrice de passage de \mathcal{B}' vers \mathcal{B} :

$$[\mathbf{x}]_{\mathcal{B}} = P \cdot [\mathbf{x}]_{\mathcal{B}'}$$

Interprétation : La matrice P est formée par les vecteurs de \mathcal{B}' exprimés dans la base \mathcal{B} :

$$P = \begin{bmatrix} | & & | \\ \mathbf{v}'_1 & \cdots & \mathbf{v}'_n \\ | & & | \end{bmatrix}$$

Changement de base très utile si \mathcal{B}' est orthonormale.

Formule de changement de base

Soient \mathcal{B} et \mathcal{B}' deux bases de \mathbb{R}^n .

Soit P la matrice de passage de \mathcal{B}' vers \mathcal{B} :

$$[\mathbf{x}]_{\mathcal{B}} = P \cdot [\mathbf{x}]_{\mathcal{B}'}$$

Interprétation : La matrice P est formée par les vecteurs de \mathcal{B}' exprimés dans la base \mathcal{B} :

$$P = \begin{bmatrix} | & & | \\ \mathbf{v}'_1 & \cdots & \mathbf{v}'_n \\ | & & | \end{bmatrix}$$

Changement de base très utile si \mathcal{B}' est orthonormale.

Procédé de Gram-Schmidt (1/2)

Objectif : Transformer une base quelconque $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ en une base **orthogonale**.

Formules : Soit

$$\mathbf{u}_1 = \mathbf{v}_1$$

$$\mathbf{u}_2 = \mathbf{v}_2 - \frac{\langle \mathbf{v}_2, \mathbf{u}_1 \rangle}{\langle \mathbf{u}_1, \mathbf{u}_1 \rangle} \mathbf{u}_1$$

$$\mathbf{u}_3 = \mathbf{v}_3 - \frac{\langle \mathbf{v}_3, \mathbf{u}_1 \rangle}{\langle \mathbf{u}_1, \mathbf{u}_1 \rangle} \mathbf{u}_1 - \frac{\langle \mathbf{v}_3, \mathbf{u}_2 \rangle}{\langle \mathbf{u}_2, \mathbf{u}_2 \rangle} \mathbf{u}_2$$

Et ainsi de suite...

Procédé de Gram-Schmidt (1/2)

Objectif : Transformer une base quelconque $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ en une base **orthogonale**.

Formules : Soit

$$\mathbf{u}_1 = \mathbf{v}_1$$

$$\mathbf{u}_2 = \mathbf{v}_2 - \frac{\langle \mathbf{v}_2, \mathbf{u}_1 \rangle}{\langle \mathbf{u}_1, \mathbf{u}_1 \rangle} \mathbf{u}_1$$

$$\mathbf{u}_3 = \mathbf{v}_3 - \frac{\langle \mathbf{v}_3, \mathbf{u}_1 \rangle}{\langle \mathbf{u}_1, \mathbf{u}_1 \rangle} \mathbf{u}_1 - \frac{\langle \mathbf{v}_3, \mathbf{u}_2 \rangle}{\langle \mathbf{u}_2, \mathbf{u}_2 \rangle} \mathbf{u}_2$$

Et ainsi de suite...

Procédé de Gram-Schmidt (1/2)

Objectif : Transformer une base quelconque $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ en une base **orthogonale**.

Formules : Soit

$$\mathbf{u}_1 = \mathbf{v}_1$$

$$\mathbf{u}_2 = \mathbf{v}_2 - \frac{\langle \mathbf{v}_2, \mathbf{u}_1 \rangle}{\langle \mathbf{u}_1, \mathbf{u}_1 \rangle} \mathbf{u}_1$$

$$\mathbf{u}_3 = \mathbf{v}_3 - \frac{\langle \mathbf{v}_3, \mathbf{u}_1 \rangle}{\langle \mathbf{u}_1, \mathbf{u}_1 \rangle} \mathbf{u}_1 - \frac{\langle \mathbf{v}_3, \mathbf{u}_2 \rangle}{\langle \mathbf{u}_2, \mathbf{u}_2 \rangle} \mathbf{u}_2$$

Et ainsi de suite...

Procédé de Gram-Schmidt (2/2)

Remarque : On obtient une base $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ **orthogonale**. Pour obtenir une base orthonormée, on normalise :

$$\mathbf{e}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|}$$

Utilité :

- Diagonalisation plus facile (matrices symétriques) ;
- Calculs simplifiés avec $\langle \mathbf{e}_i, \mathbf{e}_j \rangle = \delta_{ij}$;
- Étape de base dans la décomposition QR et dans l'ACP.

Procédé de Gram-Schmidt (2/2)

Remarque : On obtient une base $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ **orthogonale**. Pour obtenir une base orthonormée, on normalise :

$$\mathbf{e}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|}$$

Utilité :

- Diagonalisation plus facile (matrices symétriques) ;
- Calculs simplifiés avec $\langle \mathbf{e}_i, \mathbf{e}_j \rangle = \delta_{ij}$;
- Étape de base dans la décomposition QR et dans l'ACP.

Application : réduction de dimension (PCA)

ACP (Analyse en Composantes Principales) :

- Objectif : trouver une base orthogonale où les données sont projetées avec variance maximale ;
- Basée sur les vecteurs propres de la matrice de covariance ;
- Retourne une nouvelle base $\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$ (avec $k < n$).

Interprétation géométrique :

- Nouvelle base = directions principales de la distribution ;
- Les données sont « compressées » dans ce nouveau repère.

Application : réduction de dimension (PCA)

ACP (Analyse en Composantes Principales) :

- Objectif : trouver une base orthogonale où les données sont projetées avec variance maximale ;
- Basée sur les vecteurs propres de la matrice de covariance ;
- Retourne une nouvelle base $\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$ (avec $k < n$).

Interprétation géométrique :

- Nouvelle base = directions principales de la distribution ;
- Les données sont « compressées » dans ce nouveau repère.

ACP : résumé mathématique

Étapes de la PCA :

- 1 Centrer les données : $X \leftarrow X - \mu$
- 2 Calculer la matrice de covariance : $C = \frac{1}{n} X^T X$
- 3 Calculer les vecteurs propres $\{\mathbf{v}_i\}$ et valeurs propres $\{\lambda_i\}$ de C
- 4 Choisir les k plus grandes λ_i et construire la base projetée U_k
- 5 Nouvelle représentation : $Z = XU_k$

Avantages :

- Réduction de dimension ;
- Compression avec perte minimale d'information ;
- Très utilisé en Machine Learning pour la pré-analyse.

ACP : résumé mathématique

Étapes de la PCA :

- ❶ Centrer les données : $X \leftarrow X - \mu$
- ❷ Calculer la matrice de covariance : $C = \frac{1}{n} X^T X$
- ❸ Calculer les vecteurs propres $\{\mathbf{v}_i\}$ et valeurs propres $\{\lambda_i\}$ de C
- ❹ Choisir les k plus grandes λ_i et construire la base projetée U_k
- ❺ Nouvelle représentation : $Z = XU_k$

Avantages :

- Réduction de dimension ;
- Compression avec perte minimale d'information ;
- Très utilisé en Machine Learning pour la pré-analyse.

Valeurs propres et vecteurs propres

Définitions :

Soit $A \in \mathbb{R}^{n \times n}$. Un vecteur $\mathbf{v} \neq \mathbf{0}$ est un **vecteur propre** de A s'il existe un scalaire λ tel que :

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

λ est alors une **valeur propre** de A .

Interprétation géométrique : \mathbf{v} est une direction invariante par A ; A étire (ou contracte ou inverse) \mathbf{v} sans changer sa direction.

Valeurs propres et vecteurs propres

Définitions :

Soit $A \in \mathbb{R}^{n \times n}$. Un vecteur $\mathbf{v} \neq \mathbf{0}$ est un **vecteur propre** de A s'il existe un scalaire λ tel que :

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

λ est alors une **valeur propre** de A .

Interprétation géométrique : \mathbf{v} est une direction invariante par A ; A étire (ou contracte ou inverse) \mathbf{v} sans changer sa direction.

Comment calculer les valeurs propres ?

Équation caractéristique :

$$A\mathbf{v} = \lambda\mathbf{v} \Rightarrow (A - \lambda I)\mathbf{v} = \mathbf{0}$$

$$\Rightarrow \det(A - \lambda I) = 0$$

Cette équation donne un polynôme de degré n (appelé polynôme caractéristique), dont les racines sont les valeurs propres λ .

Exemple :

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad \det(A - \lambda I) = (\lambda - 3)(\lambda - 1) \Rightarrow \lambda = 3, 1$$

Comment calculer les valeurs propres ?

Équation caractéristique :

$$A\mathbf{v} = \lambda\mathbf{v} \Rightarrow (A - \lambda I)\mathbf{v} = \mathbf{0}$$

$$\Rightarrow \det(A - \lambda I) = 0$$

Cette équation donne un polynôme de degré n (appelé polynôme caractéristique), dont les racines sont les valeurs propres λ .

Exemple :

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad \det(A - \lambda I) = (\lambda - 3)(\lambda - 1) \Rightarrow \lambda = 3, 1$$

Comment calculer les valeurs propres ?

Équation caractéristique :

$$A\mathbf{v} = \lambda\mathbf{v} \Rightarrow (A - \lambda I)\mathbf{v} = \mathbf{0}$$

$$\Rightarrow \det(A - \lambda I) = 0$$

Cette équation donne un polynôme de degré n (appelé polynôme caractéristique), dont les racines sont les valeurs propres λ .

Exemple :

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad \det(A - \lambda I) = (\lambda - 3)(\lambda - 1) \Rightarrow \lambda = 3, 1$$

Diagonalisation

Définition : Une matrice $A \in \mathbb{R}^{n \times n}$ est diagonalisable s'il existe une matrice inversible P et une matrice diagonale D telles que :

$$A = PDP^{-1}$$

Condition nécessaire : A possède n vecteurs propres linéairement indépendants.

Intérêt : La puissance k d'une matrice devient facile à calculer :

$$A^k = PD^kP^{-1} \quad \text{avec} \quad D^k = \text{diag}(\lambda_1^k, \dots, \lambda_n^k)$$

Diagonalisation

Définition : Une matrice $A \in \mathbb{R}^{n \times n}$ est diagonalisable s'il existe une matrice inversible P et une matrice diagonale D telles que :

$$A = PDP^{-1}$$

Condition nécessaire : A possède n vecteurs propres linéairement indépendants.

Intérêt : La puissance k d'une matrice devient facile à calculer :

$$A^k = PD^kP^{-1} \quad \text{avec} \quad D^k = \text{diag}(\lambda_1^k, \dots, \lambda_n^k)$$

Diagonalisation

Définition : Une matrice $A \in \mathbb{R}^{n \times n}$ est diagonalisable s'il existe une matrice inversible P et une matrice diagonale D telles que :

$$A = PDP^{-1}$$

Condition nécessaire : A possède n vecteurs propres linéairement indépendants.

Intérêt : La puissance k d'une matrice devient facile à calculer :

$$A^k = PD^kP^{-1} \quad \text{avec} \quad D^k = \text{diag}(\lambda_1^k, \dots, \lambda_n^k)$$

Application : stabilité

Considérons un système dynamique linéaire discret :

$$\mathbf{x}_{t+1} = A\mathbf{x}_t$$

Si \mathbf{x}_0 est combinaison des vecteurs propres de A :

$$\mathbf{x}_t = A^t \mathbf{x}_0 = P D^t P^{-1} \mathbf{x}_0$$

Stabilité :

- Si $|\lambda_i| < 1$ pour tout i , alors $\mathbf{x}_t \rightarrow 0$;
- Si certains $|\lambda_i| > 1$, alors le système diverge.

Les valeurs propres déterminent le comportement à long terme.

Application : stabilité

Considérons un système dynamique linéaire discret :

$$\mathbf{x}_{t+1} = A\mathbf{x}_t$$

Si \mathbf{x}_0 est combinaison des vecteurs propres de A :

$$\mathbf{x}_t = A^t \mathbf{x}_0 = P D^t P^{-1} \mathbf{x}_0$$

Stabilité :

- Si $|\lambda_i| < 1$ pour tout i , alors $\mathbf{x}_t \rightarrow 0$;
- Si certains $|\lambda_i| > 1$, alors le système diverge.

Les valeurs propres déterminent le comportement à long terme.

Application : stabilité

Considérons un système dynamique linéaire discret :

$$\mathbf{x}_{t+1} = A\mathbf{x}_t$$

Si \mathbf{x}_0 est combinaison des vecteurs propres de A :

$$\mathbf{x}_t = A^t \mathbf{x}_0 = P D^t P^{-1} \mathbf{x}_0$$

Stabilité :

- Si $|\lambda_i| < 1$ pour tout i , alors $\mathbf{x}_t \rightarrow 0$;
- Si certains $|\lambda_i| > 1$, alors le système diverge.

Les valeurs propres déterminent le comportement à long terme.

Application : stabilité

Considérons un système dynamique linéaire discret :

$$\mathbf{x}_{t+1} = A\mathbf{x}_t$$

Si \mathbf{x}_0 est combinaison des vecteurs propres de A :

$$\mathbf{x}_t = A^t \mathbf{x}_0 = P D^t P^{-1} \mathbf{x}_0$$

Stabilité :

- Si $|\lambda_i| < 1$ pour tout i , alors $\mathbf{x}_t \rightarrow 0$;
- Si certains $|\lambda_i| > 1$, alors le système diverge.

Les valeurs propres déterminent le comportement à long terme.

Lien avec les réseaux dynamiques

Soit A la matrice d'adjacence pondérée d'un graphe. L'état du réseau à l'instant t peut être modélisé par :

$$\mathbf{x}_{t+1} = A\mathbf{x}_t$$

Exemples d'applications :

- Diffusion de l'information (ou virus) ;
- Synchronisation dans un réseau ;
- Réseaux de neurones récurrents linéarisés.

La dynamique est contrôlée par les valeurs propres de A .

Lien avec les réseaux dynamiques

Soit A la matrice d'adjacence pondérée d'un graphe. L'état du réseau à l'instant t peut être modélisé par :

$$\mathbf{x}_{t+1} = A\mathbf{x}_t$$

Exemples d'applications :

- Diffusion de l'information (ou virus) ;
- Synchronisation dans un réseau ;
- Réseaux de neurones récurrents linéarisés.

La dynamique est contrôlée par les valeurs propres de A .

Lien avec les réseaux dynamiques

Soit A la matrice d'adjacence pondérée d'un graphe. L'état du réseau à l'instant t peut être modélisé par :

$$\mathbf{x}_{t+1} = A\mathbf{x}_t$$

Exemples d'applications :

- Diffusion de l'information (ou virus) ;
- Synchronisation dans un réseau ;
- Réseaux de neurones récurrents linéarisés.

La dynamique est contrôlée par les valeurs propres de A .

Chaînes de Markov : rappel

Soit P une matrice de transition de probabilité (stochastique) :

$$P_{ij} = \mathbb{P}[\text{état } j \text{ à } t+1 \mid \text{état } i \text{ à } t]$$

L'état du système à l'instant t est :

$$\mathbf{p}^{(t)} = P^t \mathbf{p}^{(0)}$$

Objectif : étudier le comportement à long terme $\lim_{t \rightarrow \infty} P^t \mathbf{p}^{(0)}$

Chaînes de Markov : rappel

Soit P une matrice de transition de probabilité (stochastique) :

$$P_{ij} = \mathbb{P}[\text{état } j \text{ à } t+1 \mid \text{état } i \text{ à } t]$$

L'état du système à l'instant t est :

$$\mathbf{p}^{(t)} = P^t \mathbf{p}^{(0)}$$

Objectif : étudier le comportement à long terme $\lim_{t \rightarrow \infty} P^t \mathbf{p}^{(0)}$

Chaînes de Markov : rappel

Soit P une matrice de transition de probabilité (stochastique) :

$$P_{ij} = \mathbb{P}[\text{état } j \text{ à } t+1 \mid \text{état } i \text{ à } t]$$

L'état du système à l'instant t est :

$$\mathbf{p}^{(t)} = P^t \mathbf{p}^{(0)}$$

Objectif : étudier le comportement à long terme $\lim_{t \rightarrow \infty} P^t \mathbf{p}^{(0)}$

Chaînes de Markov et valeurs propres

Fait : P a toujours une valeur propre $\lambda_1 = 1$. Si la chaîne est irréductible et apériodique :

$$\lim_{t \rightarrow \infty} P^t = \mathbf{1}\pi^T \quad \text{où } \pi \text{ est la distribution stationnaire}$$

Rôle des autres valeurs propres :

- Les λ_i avec $|\lambda_i| < 1$ contrôlent la vitesse de convergence vers l'équilibre ;
- Plus $|\lambda_2|$ est petit, plus la chaîne converge vite.

Outils utilisés en théorie des graphes, random walk, PageRank, etc.

Chaînes de Markov et valeurs propres

Fait : P a toujours une valeur propre $\lambda_1 = 1$. Si la chaîne est irréductible et apériodique :

$$\lim_{t \rightarrow \infty} P^t = \mathbf{1}\pi^T \quad \text{où } \pi \text{ est la distribution stationnaire}$$

Rôle des autres valeurs propres :

- Les λ_i avec $|\lambda_i| < 1$ contrôlent la vitesse de convergence vers l'équilibre ;
- Plus $|\lambda_2|$ est petit, plus la chaîne converge vite.

Outils utilisés en théorie des graphes, random walk, PageRank, etc.

Chaînes de Markov et valeurs propres

Fait : P a toujours une valeur propre $\lambda_1 = 1$. Si la chaîne est irréductible et apériodique :

$$\lim_{t \rightarrow \infty} P^t = \mathbf{1}\pi^T \quad \text{où } \pi \text{ est la distribution stationnaire}$$

Rôle des autres valeurs propres :

- Les λ_i avec $|\lambda_i| < 1$ contrôlent la vitesse de convergence vers l'équilibre ;
- Plus $|\lambda_2|$ est petit, plus la chaîne converge vite.

Outils utilisés en théorie des graphes, random walk, PageRank, etc.

Introduction à la décomposition

Objectif : Écrire une matrice $A \in \mathbb{R}^{m \times n}$ comme produit de trois matrices plus simples.

Décomposition en valeurs singulières (SVD) :

$$A = U \Sigma V^T$$

- $U \in \mathbb{R}^{m \times m}$: matrice orthogonale ($U^T U = I$)
- $V \in \mathbb{R}^{n \times n}$: matrice orthogonale
- $\Sigma \in \mathbb{R}^{m \times n}$: matrice diagonale avec valeurs singulières $\sigma_1 \geq \sigma_2 \geq \dots \geq 0$

Introduction à la décomposition

Objectif : Écrire une matrice $A \in \mathbb{R}^{m \times n}$ comme produit de trois matrices plus simples.

Décomposition en valeurs singulières (SVD) :

$$A = U \Sigma V^T$$

- $U \in \mathbb{R}^{m \times m}$: matrice orthogonale ($U^T U = I$)
- $V \in \mathbb{R}^{n \times n}$: matrice orthogonale
- $\Sigma \in \mathbb{R}^{m \times n}$: matrice diagonale avec valeurs singulières $\sigma_1 \geq \sigma_2 \geq \dots \geq 0$

Interprétation géométrique de la SVD

La SVD transforme un vecteur \mathbf{x} en trois étapes :

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

- $V^T \mathbf{x}$: rotation (changement de base) dans l'espace des colonnes
- Σ : mise à l'échelle des composantes
- U : rotation finale dans l'espace des lignes

La SVD donne une description optimale de A en termes de directions principales.

Interprétation géométrique de la SVD

La SVD transforme un vecteur \mathbf{x} en trois étapes :

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

- $V^T \mathbf{x}$: rotation (changement de base) dans l'espace des colonnes
- Σ : mise à l'échelle des composantes
- U : rotation finale dans l'espace des lignes

La SVD donne une description optimale de A en termes de directions principales.

Interprétation géométrique de la SVD

La SVD transforme un vecteur \mathbf{x} en trois étapes :

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

- $V^T \mathbf{x}$: rotation (changement de base) dans l'espace des colonnes
- Σ : mise à l'échelle des composantes
- U : rotation finale dans l'espace des lignes

La SVD donne une description optimale de A en termes de directions principales.

Exemple : petite matrice

Soit :

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

On peut écrire :

$$A = U\Sigma V^T \quad \text{où} \quad \Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

Valeurs singulières :

$$\sigma_1 = 4, \quad \sigma_2 = 2$$

A étire plus dans une direction que dans une autre.

Exemple : petite matrice

Soit :

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

On peut écrire :

$$A = U\Sigma V^T \quad \text{où} \quad \Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

Valeurs singulières :

$$\sigma_1 = 4, \quad \sigma_2 = 2$$

A étire plus dans une direction que dans une autre.

Exemple : petite matrice

Soit :

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

On peut écrire :

$$A = U\Sigma V^T \quad \text{où} \quad \Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

Valeurs singulières :

$$\sigma_1 = 4, \quad \sigma_2 = 2$$

A étire plus dans une direction que dans une autre.

Exemple : petite matrice

Soit :

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

On peut écrire :

$$A = U\Sigma V^T \quad \text{où} \quad \Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

Valeurs singulières :

$$\sigma_1 = 4, \quad \sigma_2 = 2$$

A étire plus dans une direction que dans une autre.

SVD pour réduction de dimension

La SVD permet une **approximation de rang k** :

$$A \approx A_k = U_k \Sigma_k V_k^T$$

où :

- U_k : les k premières colonnes de U
- Σ_k : matrice $k \times k$ avec les k plus grandes valeurs singulières
- V_k : les k premières colonnes de V

Théorème d'Eckart-Young : A_k est la meilleure approximation de A de rang k (en norme Frobenius).

SVD pour réduction de dimension

La SVD permet une **approximation de rang k** :

$$A \approx A_k = U_k \Sigma_k V_k^T$$

où :

- U_k : les k premières colonnes de U
- Σ_k : matrice $k \times k$ avec les k plus grandes valeurs singulières
- V_k : les k premières colonnes de V

Théorème d'Eckart-Young : A_k est la meilleure approximation de A de rang k (en norme Frobenius).

SVD pour réduction de dimension

La SVD permet une **approximation de rang k** :

$$A \approx A_k = U_k \Sigma_k V_k^T$$

où :

- U_k : les k premières colonnes de U
- Σ_k : matrice $k \times k$ avec les k plus grandes valeurs singulières
- V_k : les k premières colonnes de V

Théorème d'Eckart-Young : A_k est la meilleure approximation de A de rang k (en norme Frobenius).

Application : compression d'image

Une image (niveau de gris) est une matrice $A \in \mathbb{R}^{m \times n}$.

On applique la SVD : $A = U\Sigma V^T$

On garde les k plus grandes valeurs singulières :

$$A_k = U_k \Sigma_k V_k^T$$

Compression :

- $k = 50 \ll m, n$: image approximée avec peu de stockage ;
- Visualisation proche de l'original si σ_k suffisamment grands.

Application : compression d'image

Une image (niveau de gris) est une matrice $A \in \mathbb{R}^{m \times n}$.

On applique la SVD : $A = U\Sigma V^T$

On garde les k plus grandes valeurs singulières :

$$A_k = U_k \Sigma_k V_k^T$$

Compression :

- $k = 50 \ll m, n$: image approximée avec peu de stockage ;
- Visualisation proche de l'original si σ_k suffisamment grands.

Application : compression d'image

Une image (niveau de gris) est une matrice $A \in \mathbb{R}^{m \times n}$.

On applique la SVD : $A = U\Sigma V^T$

On garde les k plus grandes valeurs singulières :

$$A_k = U_k \Sigma_k V_k^T$$

Compression :

- $k = 50 \ll m, n$: image approximée avec peu de stockage ;
- Visualisation proche de l'original si σ_k suffisamment grands.

Application : compression d'image

Une image (niveau de gris) est une matrice $A \in \mathbb{R}^{m \times n}$.

On applique la SVD : $A = U\Sigma V^T$

On garde les k plus grandes valeurs singulières :

$$A_k = U_k \Sigma_k V_k^T$$

Compression :

- $k = 50 \ll m, n$: image approximée avec peu de stockage ;
- Visualisation proche de l'original si σ_k suffisamment grands.

SVD en NLP : analyse sémantique latente (LSA)

Matrice document-terme $A \in \mathbb{R}^{m \times n}$:

A_{ij} = pondération TF-IDF du terme j dans le document i

SVD :

$$A \approx U_k \Sigma_k V_k^T$$

- U_k : représentation des documents dans l'espace latent
- V_k : représentation des termes
- Σ_k : importance des dimensions sémantiques

Réduction de bruit, extraction de thèmes.

SVD en NLP : analyse sémantique latente (LSA)

Matrice document-terme $A \in \mathbb{R}^{m \times n}$:

A_{ij} = pondération TF-IDF du terme j dans le document i

SVD :

$$A \approx U_k \Sigma_k V_k^T$$

- U_k : représentation des documents dans l'espace latent
- V_k : représentation des termes
- Σ_k : importance des dimensions sémantiques

Réduction de bruit, extraction de thèmes.

SVD en NLP : analyse sémantique latente (LSA)

Matrice document-terme $A \in \mathbb{R}^{m \times n}$:

A_{ij} = pondération TF-IDF du terme j dans le document i

SVD :

$$A \approx U_k \Sigma_k V_k^T$$

- U_k : représentation des documents dans l'espace latent
- V_k : représentation des termes
- Σ_k : importance des dimensions sémantiques

Réduction de bruit, extraction de thèmes.

SVD en NLP : analyse sémantique latente (LSA)

Matrice document-terme $A \in \mathbb{R}^{m \times n}$:

A_{ij} = pondération TF-IDF du terme j dans le document i

SVD :

$$A \approx U_k \Sigma_k V_k^T$$

- U_k : représentation des documents dans l'espace latent
- V_k : représentation des termes
- Σ_k : importance des dimensions sémantiques

Réduction de bruit, extraction de thèmes.

Interprétation sémantique

La SVD regroupe les termes co-occurents dans des dimensions principales.

Exemple :

- termes : "chat", "animal", "chien", "voiture", "camion"
- SVD les regroupe par sémantique : (animaux vs véhicules)

Cela améliore la recherche sémantique :

- Interrogation "chien" \rightarrow document contenant "chat"
- Grâce à la proximité dans l'espace latent

Interprétation sémantique

La SVD regroupe les termes co-occurents dans des dimensions principales.

Exemple :

- termes : “chat”, “animal”, “chien”, “voiture”, “camion”
- SVD les regroupe par sémantique : (animaux vs véhicules)

Cela améliore la recherche sémantique :

- Interrogation “chien” \rightarrow document contenant “chat”
- Grâce à la proximité dans l'espace latent

Interprétation sémantique

La SVD regroupe les termes co-occurents dans des dimensions principales.

Exemple :

- termes : “chat”, “animal”, “chien”, “voiture”, “camion”
- SVD les regroupe par sémantique : (animaux vs véhicules)

Cela améliore la recherche sémantique :

- Interrogation “chien” \rightarrow document contenant “chat”
- Grâce à la proximité dans l'espace latent

Résumé et perspectives

Résumé :

- SVD décompose toute matrice A en $U\Sigma V^T$
- Donne une base orthonormée optimale (compression, approximation)
- Clé en réduction de dimension, visualisation, NLP

Perspectives :

- Intégration dans l'apprentissage automatique (Truncated SVD)
- Lien avec PCA (analyse en composantes principales)
- Alternatives : NMF, autoencodeurs

Résumé et perspectives

Résumé :

- SVD décompose toute matrice A en $U\Sigma V^T$
- Donne une base orthonormée optimale (compression, approximation)
- Clé en réduction de dimension, visualisation, NLP

Perspectives :

- Intégration dans l'apprentissage automatique (Truncated SVD)
- Lien avec PCA (analyse en composantes principales)
- Alternatives : NMF, autoencodeurs