

Le calcul matriciel

Un réseau de neurones n'est, au fond, qu'une suite de produits matriciels

À savoir avant de commencer

Ce document réutilise le produit scalaire (document 2) et la fonction sigmoïde (document 3) : une couche entière de neurones se calcule en une seule opération, le produit d'une matrice par un vecteur.

Niveau : Terminale (spécialité + option mathématiques expertes)

Introduction : d'un neurone à toute une couche

Dans le document 3, un neurone calculait $z = ax + b$ à partir d'une seule caractéristique x . Plus généralement (document 2), un neurone calcule un produit scalaire entre un vecteur de poids et un vecteur d'entrée : $z = w \cdot x + b$.

Ce document répond à une question simple : comment calcule-t-on, en une seule fois, la sortie de plusieurs neurones à la fois ? Réponse : avec un produit matriciel.

Partie A — Rappels de calcul matriciel

Niveau : Terminale (option mathématiques expertes)

Rappels de cours

Une matrice de dimension $m \times n$ est un tableau de nombres à m lignes et n colonnes.

Le produit d'une matrice M (de dimension $m \times n$) par un vecteur colonne x (de dimension $n \times 1$) est un vecteur colonne $z = Mx$ de dimension $m \times 1$, où chaque coordonnée de z est le produit scalaire d'une ligne de M avec x .

Le produit de deux matrices M ($m \times n$) et N ($n \times p$) est une matrice MN de dimension $m \times p$ — cela ne fonctionne que si le nombre de colonnes de M est égal au nombre de lignes de N .

On donne $M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 1 \end{bmatrix}$ (une matrice 3×2) et $x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ (un vecteur colonne).

1. Quelle est la dimension du vecteur Mx ? Calcule Mx .
2. Peut-on calculer xM ? Justifie à partir des règles de dimension.

Partie B — Un neurone comme produit scalaire (rappel)

Niveau : Terminale

Un neurone à 3 entrées x_1, x_2, x_3 calcule $z = w_1x_1 + w_2x_2 + w_3x_3 + b$, qu'on peut écrire $z = w \cdot x + b$ où $w = (w_1, w_2, w_3)$ est le vecteur des poids du neurone (document 2).

3. Un neurone a pour poids $w = (0,2 ; -0,1 ; 0,4)$ et pour biais $b = 0,1$. Calcule sa sortie z pour l'entrée $x = (1 ; 2 ; 1)$.

Partie C — Une couche entière de neurones = un produit matriciel

Niveau : Terminale (option mathématiques expertes)

Une couche de neurones, ce n'est rien d'autre que plusieurs neurones (comme celui de la partie B) appliqués en parallèle à la même entrée. Si une couche a 2 neurones de sortie recevant chacun 3 entrées, on peut ranger les poids des 2 neurones dans une matrice W à 2 lignes (une par neurone) et 3 colonnes (une par entrée) :



Une couche = un produit matriciel

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

où $\mathbf{W}$ est la matrice des poids (une ligne par neurone de sortie), $\mathbf{x}$ le vecteur d'entrée, et $\mathbf{b}$ le vecteur des biais (un par neurone).

On donne $\mathbf{W} = \begin{bmatrix} 0,2 & -0,1 & 0,4 \\ 0,5 & 0,3 & -0,2 \end{bmatrix}$, $\mathbf{x} = (1 ; 2 ; 1)$ et $\mathbf{b} = (0,1 ; -0,2)$, correspondant exactement au schéma ci-dessus.

4. Calcule $\mathbf{z} = \mathbf{W} \mathbf{x} + \mathbf{b}$ (tu dois retrouver, pour la première coordonnée, le calcul de la partie B).
5. On applique ensuite la fonction sigmoïde (document 3) à chaque coordonnée de $\mathbf{z}$, séparément. Calcule la sortie finale de la couche (arrondis à 10^{-3} près).

Cette opération — un produit matriciel, suivi d'une fonction appliquée coordonnée par coordonnée — est exactement ce qui se répète, couche après couche, dans un réseau de neurones.

Partie D — Pourquoi les couches ne peuvent pas être seulement matricielles

Niveau : Terminale (option mathématiques expertes)

Que se passerait-il si on enchaînait plusieurs couches sans jamais appliquer de fonction non linéaire (comme la sigmoïde) entre elles ? On donne deux couches successives, sans sigmoïde : $\mathbf{z}_1 = \mathbf{W}_1 \mathbf{x}$, puis $\mathbf{z}_2 = \mathbf{W}_2 \mathbf{z}_1$.

6. Exprime $\mathbf{z}_2$ en fonction de $\mathbf{x}$ uniquement (remplace $\mathbf{z}_1$ par son expression). Quelle propriété du produit matriciel utilises-tu ?

On donne $\mathbf{W}_1 = \begin{bmatrix} 1,2 \\ 0,1 \end{bmatrix}$, $\mathbf{W}_2 = \begin{bmatrix} 2,0 \\ 1,1 \end{bmatrix}$ et $\mathbf{x} = (1 ; 1)$.

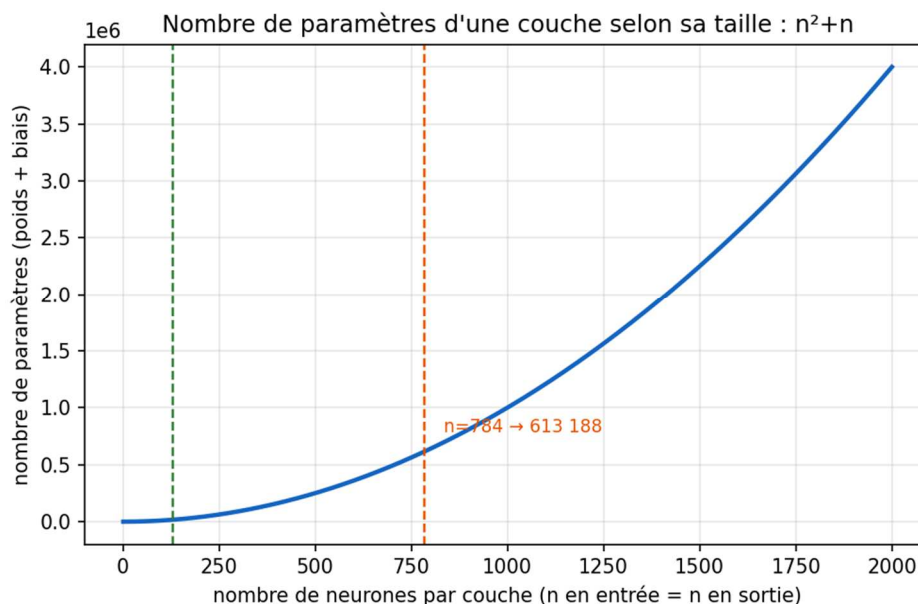
7. Calcule $\mathbf{z}_1 = \mathbf{W}_1 \mathbf{x}$, puis $\mathbf{z}_2 = \mathbf{W}_2 \mathbf{z}_1$.
8. Calcule la matrice produit $\mathbf{W}_2 \mathbf{W}_1$, puis calcule $(\mathbf{W}_2 \mathbf{W}_1) \mathbf{x}$. Compare avec le résultat de la question précédente.
9. D'après ces calculs, deux couches successives **sans fonction non linéaire entre elles** équivalent-elles à une seule couche (avec une seule matrice) ? Pourquoi est-ce un problème pour un réseau qui est censé gagner en puissance de calcul à chaque couche ajoutée ?

C'est précisément pour cette raison que la sigmoïde (document 3) — ou une autre fonction non linéaire — est indispensable entre les couches : sans elle, empiler 100 couches ne serait, mathématiquement, pas plus puissant qu'une seule.

Partie E — Compter les paramètres d'un réseau

Niveau : Terminale

Une couche qui reçoit n entrées et produit m sorties a besoin d'une matrice de poids de dimension $m \times n$ (donc $m \times n$ nombres), plus m biais (un par neurone de sortie) : soit $m \times n + m$ paramètres au total pour cette seule couche.



- Un réseau destiné à reconnaître des chiffres manuscrits (comme dans une image de $28 \times 28 = 784$ pixels) a une première couche de 784 entrées vers 128 neurones cachés, puis une seconde couche de 128 neurones cachés vers 10 sorties (une par chiffre de 0 à 9). Calcule le nombre de paramètres (poids + biais) de la première couche.
- Calcule le nombre de paramètres de la seconde couche.
- Calcule le nombre total de paramètres de ce réseau, en additionnant les deux couches.
- Ce réseau est en réalité un tout petit modèle. Sachant que les plus grands modèles de langage actuels comptent plusieurs centaines de milliards de paramètres, par quel ordre de grandeur ce nombre est-il supérieur au tien ?

Ce que ce modèle simplifie, et ce que fait réellement une IA comme Claude

Ce que ce modèle simplifie	Ce que fait réellement un modèle comme Claude
Un réseau à seulement 2 couches, avec des dizaines à des centaines de neurones. Des matrices de poids que l'on peut écrire et manipuler entièrement à la main. Un calcul présenté comme se faisant « étape par étape », neurone par neurone.	Des réseaux à des dizaines, voire des centaines de couches (on parle d'apprentissage profond, deep learning), avec des milliers de neurones par couche. Des matrices contenant des milliards d'entrées, stockées et multipliées sur du matériel spécialisé (processeurs graphiques, GPU, ou puces dédiées, TPU), justement conçu pour paralléliser des produits matriciels à très grande échelle. Tous les neurones d'une couche sont calculés simultanément (en parallèle), grâce à cette écriture matricielle : c'est précisément parce qu'un produit matriciel se prête bien au calcul parallèle que l'IA moderne repose sur des matrices plutôt que sur des calculs neurone par neurone.

Le principe central — une couche de neurones se résume à un produit matriciel suivi d'une fonction non linéaire — reste, lui, exactement la description mathématique de tout réseau de neurones existant, du plus petit au plus grand.

Exercices d'entraînement

Exercice 1 — Calcul d'une couche

Niveau : Terminale

On donne $W = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$, $x = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}$ et $b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

- Calcule $z = Wx + b$.
- Quelle est la dimension de W ? Combien de neurones d'entrée et de sortie cette couche a-t-elle ?

Exercice 2 — Composition de deux couches linéaires

Niveau : Terminale (option mathématiques expertes)

On reprend $W_1 = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$ et $W_2 = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$, mais avec $x = (2 ; 3)$ cette fois.

16. Calcule $z_1 = W_1 x$, puis $z_2 = W_2 z_1$.
17. Calcule le produit matriciel $W_2 W_1$, puis $(W_2 W_1)x$. Vérifie que tu retrouves le même résultat qu'à la question précédente.

Exercice 3 — Compter les paramètres d'un autre réseau

Niveau : Terminale

Un réseau a une couche d'entrée de 500 neurones, une couche cachée de 64 neurones, puis une couche de sortie de 5 neurones.

18. Calcule le nombre de paramètres de la première couche ($500 \rightarrow 64$).
19. Calcule le nombre de paramètres de la seconde couche ($64 \rightarrow 5$).
20. Calcule le nombre total de paramètres du réseau.

Corrigé

Partie A

A.1. M est de dimension 3×2 , x de dimension 2×1 , donc Mx est de dimension 3×1 . $Mx = \begin{bmatrix} 1 \times 2 + 2 \times 1 \\ 3 \times 2 + 4 \times 1 \\ 0 \times 2 + 1 \times 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 10 \\ 1 \end{bmatrix}$.

A.2. Non : pour calculer xM , il faudrait que le nombre de colonnes de x (ici, x a 1 colonne en tant que vecteur colonne 2×1) soit égal au nombre de lignes de M (ici 3) : $1 \neq 3$, donc xM n'est pas définie avec ces dimensions.

Partie B

B.1. $z = 0,2 \times 1 + (-0,1) \times 2 + 0,4 \times 1 + 0,1 = 0,2 - 0,2 + 0,4 + 0,1 = 0,5$.

Partie C

C.1. $z = Wx + b$: première coordonnée $= 0,2 \times 1 + (-0,1) \times 2 + 0,4 \times 1 + 0,1 = 0,5$ (on retrouve B.1) ; deuxième coordonnée $= 0,5 \times 1 + 0,3 \times 2 + (-0,2) \times 1 + (-0,2) = 0,5 + 0,6 - 0,2 - 0,2 = 0,7$. Donc $z = (0,5 ; 0,7)$.

C.2. $\sigma(0,5) \approx 0,622$ et $\sigma(0,7) \approx 0,668$. La sortie finale de la couche est donc environ $(0,622 ; 0,668)$.

Partie D

D.1. $z_2 = W_2 z_1 = W_2 (W_1 x) = (W_2 W_1)x$, en utilisant l'associativité du produit matriciel : $(W_2 W_1)x = W_2 (W_1 x)$.

D.2. $z_1 = W_1 x = (1 \times 1 + 2 \times 1 ; 0 \times 1 + 1 \times 1) = (3 ; 1)$. $z_2 = W_2 z_1 = (2 \times 3 + 0 \times 1 ; 1 \times 3 + 1 \times 1) = (6 ; 4)$.

D.3. $W_2 W_1$: première ligne $= (2 \times 1 + 0 \times 0 ; 2 \times 2 + 0 \times 1) = (2 ; 4)$; deuxième ligne $= (1 \times 1 + 1 \times 0 ; 1 \times 2 + 1 \times 1) = (1 ; 3)$. Donc $W_2 W_1 = \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix}$. $(W_2 W_1)x = (2 \times 1 + 4 \times 1 ; 1 \times 1 + 3 \times 1) = (6 ; 4)$: on retrouve exactement le même résultat qu'en D.2.

D.4. Oui : deux couches linéaires successives, sans fonction non linéaire entre elles, équivalent mathématiquement à une seule couche dont la matrice est le produit des deux (ici $W_2 W_1$). C'est un problème car un réseau à 100 couches purement linéaires n'aurait, dans les faits, pas plus de puissance de calcul qu'un réseau à une seule couche : toute la profondeur ajoutée serait mathématiquement inutile.

Partie E

E.1. Première couche ($784 \rightarrow 128$) : $784 \times 128 + 128 = 100\,352 + 128 = 100\,480$ paramètres.

E.2. Seconde couche ($128 \rightarrow 10$) : $128 \times 10 + 10 = 1\,280 + 10 = 1\,290$ paramètres.

E.3. Total : $100\,480 + 1\,290 = 101\,770$ paramètres.

E.4. Plusieurs centaines de milliards, c'est-à-dire environ 10^{11} à 10^{12} : par rapport à environ 10^5 ($101\,770 \approx 10^5$) pour ce petit réseau, l'écart est de l'ordre de 10^6 à 10^7 , soit un facteur d'un million à dix millions de fois plus de paramètres.

Exercice 1

1. $z = (1 \times 3 + 0 \times 1 + (-1) \times 2 + 0 ; 2 \times 3 + 1 \times 1 + 0 \times 2 + 1) = (3 - 2 ; 6 + 1 + 0 + 1) = (1 ; 8)$.

2. W est de dimension 2×3 : la couche a 3 neurones d'entrée et 2 neurones de sortie.

Exercice 2

1. $z_1 = W_1 x = (1 \times 2 + 2 \times 3 ; 0 \times 2 + 1 \times 3) = (8 ; 3)$. $z_2 = W_2 z_1 = (2 \times 8 + 0 \times 3 ; 1 \times 8 + 1 \times 3) = (16 ; 11)$.

2. $W_2 W_1 = \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix}$ (calculé en partie D). $(W_2 W_1)x = (2 \times 2 + 4 \times 3 ; 1 \times 2 + 3 \times 3) = (4 + 12 ; 2 + 9) = (16 ; 11)$: on retrouve bien le même résultat.

Exercice 3

1. $500 \times 64 + 64 = 32\,000 + 64 = 32\,064$ paramètres.
2. $64 \times 5 + 5 = 320 + 5 = 325$ paramètres.
3. Total : $32\,064 + 325 = 32\,389$ paramètres.

Mathématiques du programme mobilisées dans ce document

Terminale (option mathématiques expertes)

Matrices : produit matrice-vecteur, produit matrice-matrice, règles de dimension.
Associativité du produit matriciel.

Terminale (spécialité mathématiques)

Produit scalaire (lien avec le document 2), fonction sigmoïde (lien avec le document 3).
Ordres de grandeur, puissances de 10.