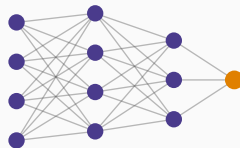


Apprentissage profond

Chapitre 1 : Du modèle linéaire au neurone



Jaouad Madkour

Faculté d'Économie et de Gestion de Tanger · 27 août 2026

Prérequis

Le fil de tout le cours

Ce qu'un seul neurone ne peut pas faire

La profondeur

Le cadre de travail

Ce qu'il faut retenir

Prérequis

Ce que ce cours suppose acquis

Les cours qui précèdent

Apprentissage statistique, ainsi que l'*Algèbre linéaire* et l'*Analyse mathématique* du niveau Licence.

Les notions mobilisées

Le compromis biais-variance, la validation croisée, la régularisation. Le produit matriciel et la dérivée composée — la rétropropagation n'est rien d'autre.

Python est ici la langue de travail; le cours suppose qu'on la lit.

Repris depuis le début

Le neurone est construit à partir du **modèle linéaire**, que le lecteur connaît déjà : un réseau n'est qu'un empilement de régressions séparées par une non-linéarité.

Aucune bibliothèque n'est traitée comme une boîte noire : ce que fait une ligne de code est établi avant d'être écrit.

Le fil de tout le cours

Un neurone est un modèle que vous connaissez

Le neurone artificiel

Un **neurone** calcule une combinaison linéaire de ses entrées, y ajoute une constante, puis applique une fonction non linéaire :

$$a = g \left(\underbrace{w_1 x_1 + \dots + w_p x_p + b}_z \right)$$

w sont les **poids**, b le **biais**, g la fonction d'**activation**.

La reconnaissance à faire immédiatement

Si g est la fonction logistique, cette formule **est** la régression logistique du chapitre 6 d'*Apprentissage statistique*.

Un réseau de neurones n'est donc pas un objet nouveau : c'est un **empilement** de régressions logistiques, entraîné par descente de gradient.

Ce cours tient ce fil du premier chapitre au dernier. C'est ce qui évite la boîte noire.

Le vocabulaire, traduit

Ce qu'un seul neurone ne peut
pas faire

La limite historique

Le problème du OU exclusif

Un neurone unique produit une frontière **linéaire**. Il ne peut donc pas représenter une relation qui exige deux frontières.

Le cas d'école est le **OU exclusif** : la sortie vaut 1 quand exactement une entrée vaut 1.

La table impossible

x_1	x_2	sortie
0	0	0
0	1	1
1	0	1
1	1	0

Aucune droite ne sépare les deux classes : les points de sortie 1 sont sur une diagonale, ceux de sortie 0 sur l'autre.

La solution, et le prix à payer

La profondeur

Pourquoi empiler

Le théorème d'approximation universelle

Un réseau à **une seule** couche cachée, suffisamment large, peut approcher n'importe quelle fonction continue avec la précision voulue.

Un théorème qui prouve moins qu'il n'y paraît

Il affirme qu'une solution **existe**. Il ne dit ni combien de neurones il faudrait, ni qu'un algorithme saura la trouver, ni qu'elle généralisera.

En pratique, la largeur requise peut être astronomique. **La profondeur est bien plus économique que la largeur** : plusieurs couches modestes représentent, avec peu de paramètres, des fonctions qu'une couche unique exigerait immense.

L'idée de représentations hiérarchiques

Chaque couche construit des descripteurs à partir de ceux de la couche précédente, du plus simple au plus abstrait.

Sur une image : contours, puis motifs, puis parties d'objets, puis objets. Sur une série financière : variations, puis régimes, puis configurations.

Ce que cela remplace

Le cadre de travail

Ce qu'il faut définir pour entraîner un réseau

1. Une **architecture** — combien de couches, de quelle largeur, avec quelles activations.
2. Une **fonction de perte** — ce que l'on cherche à minimiser.
3. Un **algorithme d'optimisation** — comment on la minimise.

Les chapitres 2 à 5 traitent ces trois pièces dans l'ordre.

Le fil rouge du cours

Le cours travaille sur des **séries financières** : rendements quotidiens d'un indice, à prévoir ou à classer en régimes.

Ce choix prolonge directement *Modélisation ARMA* et *GARCH*, et permettra au chapitre 11 de comparer honnêtement les approches — ce qui est plus instructif que de reprendre les images de chats de la littérature.

Intuition

L'apprentissage profond domine sur l'image, le texte et le son. Sur les **données tabulaires**, le boosting du chapitre 10 d'*Apprentissage statistique* reste généralement supérieur.

Le chapitre 12 y reviendra sans détour. Il vaut mieux le savoir en commençant qu'en terminant.

Ce qu'il faut retenir

Cinq points

1. Un **neurone** est une combinaison linéaire suivie d'une non-linéarité — la régression logistique en est un cas.
2. Un neurone seul ne produit qu'une frontière **linéaire** : le **OU exclusif** lui échappe.
3. Une couche cachée suffit à le résoudre, en apprenant une représentation nouvelle.
4. Le **théorème d'approximation universelle** garantit l'existence, non la trouvabilité.
5. La **profondeur** est plus économique que la largeur, et remplace l'ingénierie des variables.

La suite

Le chapitre 2 assemble ces neurones en réseau, et compte exactement ce que cela coûte en paramètres.