

# Intelligence artificielle : de quoi parle-t-on ?

---

Thierry Dumont

Jeudi 6 novembre 2025

## Plan

- Historique.
- Les neurones (artificiels...).
- Apprentissage profond.  
Quelques exemples.
- Intelligence, vraiment ?

# Histoire (préhistoire)

---

- *Cerveaux électroniques*

- *Cerveaux électroniques*



(Dr. Follamour)

- *Cerveaux électroniques*



(Dr. Follamour)

- Fin des années 80 : le terme *Intelligence Artificielle* apparaît avec *les systèmes experts...* et disparaît.

- *Cerveaux électroniques*



(Dr. Follamour)

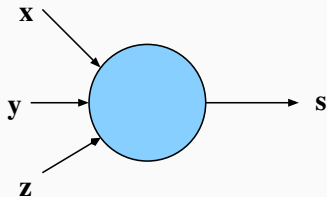
- Fin des années 80 : le terme *Intelligence Artificielle* apparaît avec *les systèmes experts...* et disparaît.
- Retour en force avec les réseaux de neurones profonds.

## « Neurones »

---

# Neurones artificiels

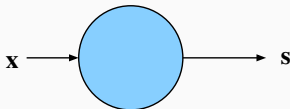
*What the frog's eye tells the frog's brain* (Warren McCulloch et Walter Pitts, 1959).



# Le Relu

## *Rectified Linear Unit*

Exemple (avec une seule entrée) :

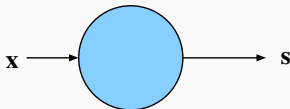


$$s = (a \times x + b)^+.$$

# Le Relu

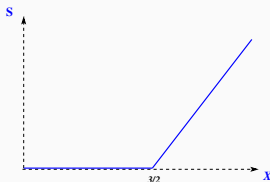
## *Rectified Linear Unit*

Exemple (avec une seule entrée) :



$$s = (a \times x + b)^+.$$

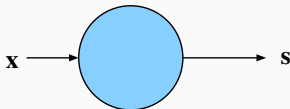
$$s = (2 \times x - 3)^+$$



# Le Relu

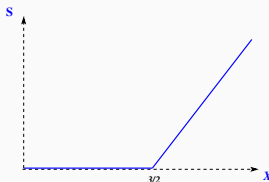
## *Rectified Linear Unit*

Exemple (avec une seule entrée) :



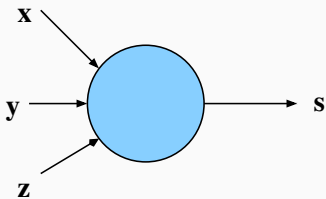
$$s = (a \times x + b)^+.$$

$$s = (2 \times x - 3)^+$$



2 paramètres.

Exemple, avec 3 entrées :

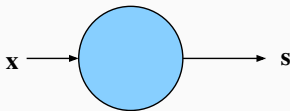


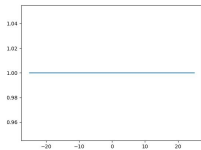
$$s = (2 \times x + 4 \times y - 3 \times z + 8)^+$$

C'est **LA** brique de base de tout ! Pourquoi ?

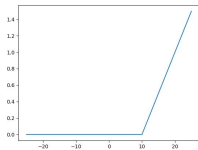
C'est **LA** brique de base de tout ! Pourquoi ?

Regardons toujours des neurones avec une seule entrée :

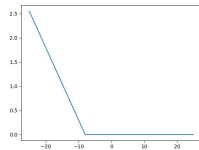




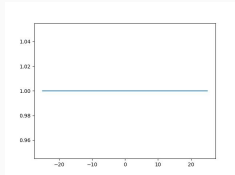
(A)  $(0x + 1)^+$



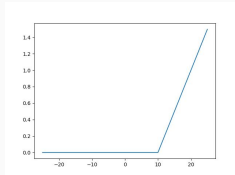
(B)  $(0.1x - 1)^+$



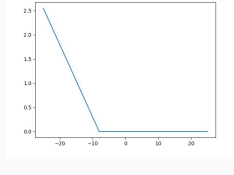
(C)  
 $(-0.15x - 1.2)^+$



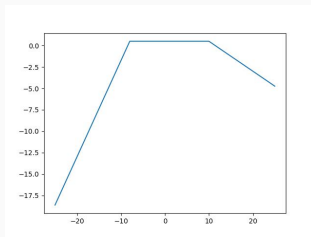
(A)  $(0x + 1)^+$



(B)  $(0.1x - 1)^+$



(C)  
 $(-0.15x - 1.2)^+$

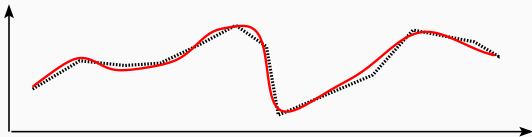


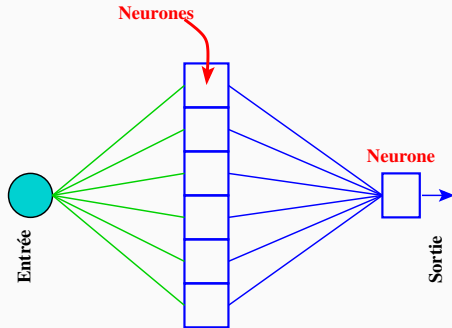
$$0.5 \times (A) - 3.5 \times (B) - 7.5 \times (C)$$

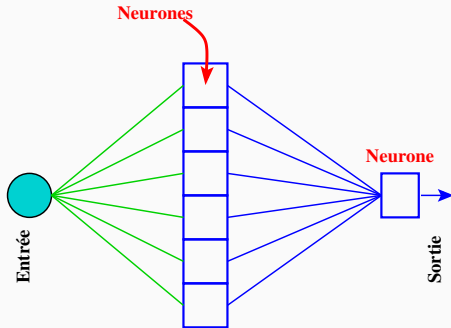
Difficile de fabriquer la courbe qu'on veut ! Et pourtant :

Combiner les neurones (ReLU) permet de fabriquer toutes les lignes brisées qu'on veut...

Combiner les neurones (ReLU) permet de fabriquer toutes les lignes brisées qu'on veut...





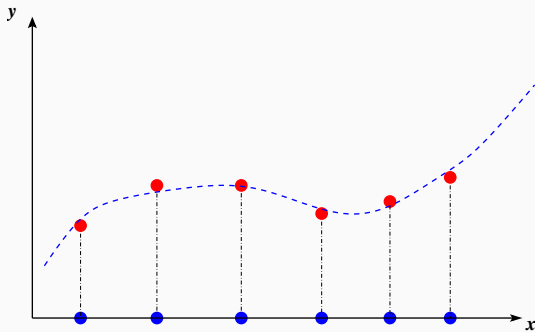


Nombre de coefficients :

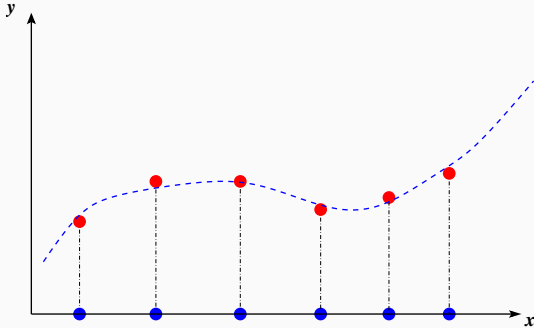
- Première couche :  $6 \times 2 = 12$ .
- Sortie : 6 (pas de ReLu).

soit en tout 18 coefficients.

## « Apprendre une courbe »

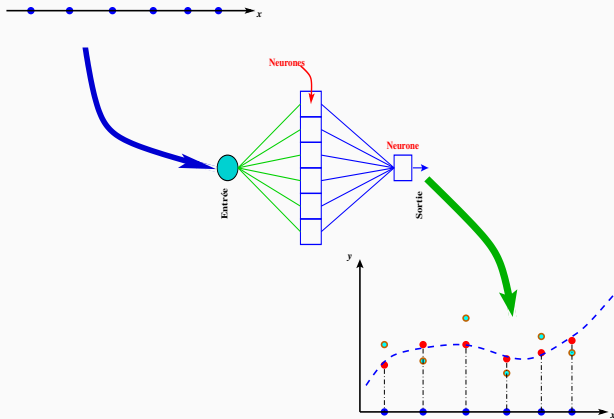


## « Apprendre une courbe »

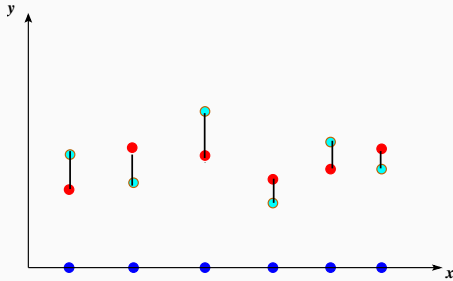


Données : des couples  $(x, y)$ .

Apprendre *par les données*.

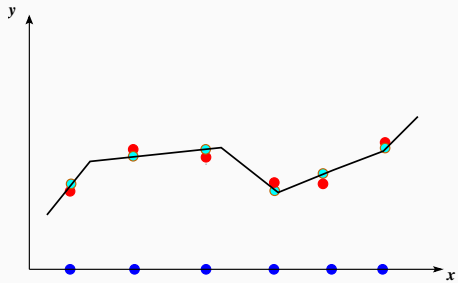


Points prévus, points observés

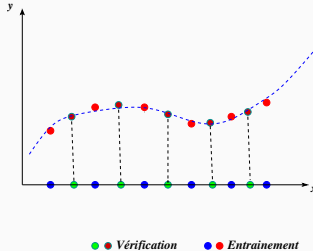


Apprendre =

- Choisir les paramètres des neurones pour minimiser une distance entre points prévus et points observés.



# Comment vérifier ?



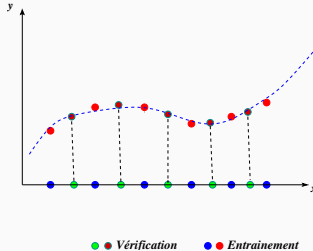
Il faut un ensemble de couples

$(x, y) = (\text{donnée}, \text{résultat}),$

partitionné en :

1. des couples pour l'apprentissage,
2. des couples pour vérifier.

# Comment vérifier ?



Il faut un ensemble de couples

$(x, y) = (\text{donnée}, \text{résultat}),$

partitionné en :

1. des couples pour l'apprentissage,
2. des couples pour vérifier.

Apprentissage **supervisé**.

Questions importantes :

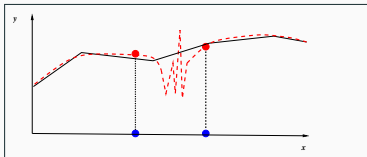
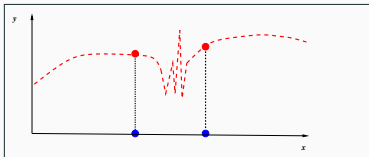
- Comment choisir le nombre de neurones ?

## Questions importantes :

- Comment choisir le nombre de neurones ?
- Fiabilité (de l'apprentissage et de la vérification) ? (question de régularité).

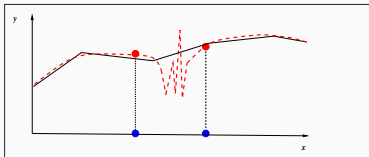
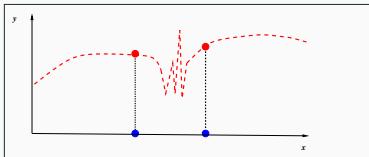
## Questions importantes :

- Comment choisir le nombre de neurones ?
- Fiabilité (de l'apprentissage et de la vérification) ? (question de régularité).



## Questions importantes :

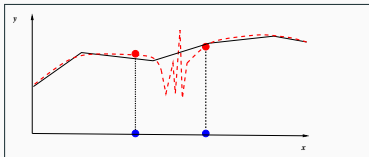
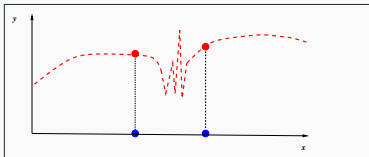
- Comment choisir le nombre de neurones ?
- Fiabilité (de l'apprentissage et de la vérification) ? (question de régularité).



- Extrapoler est sûrement dangereux !

## Questions importantes :

- Comment choisir le nombre de neurones ?
- Fiabilité (de l'apprentissage et de la vérification) ? (question de régularité).

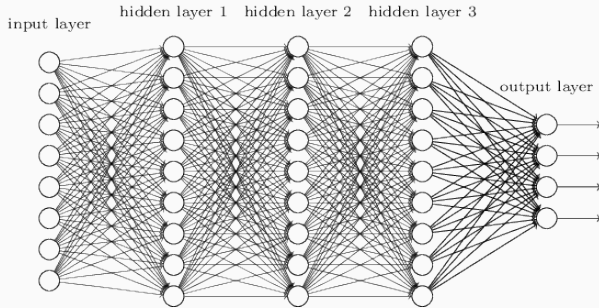


- Extrapoler est sûrement dangereux !
- Question pratique : comment calculer les paramètres des neurones qui minimisent la distance (calcul/observation) ?

# Apprentissage profond

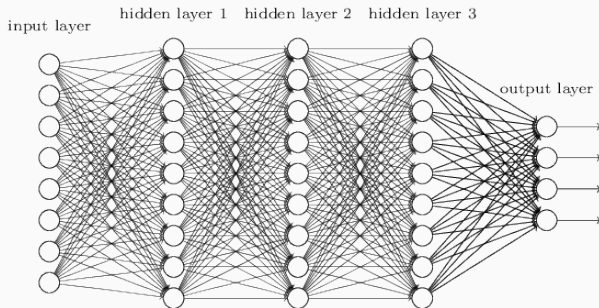
---

# Réseau de neurones profond



Réseau de neurones profond (3 couches cachées), *dense*.

# Réseau de neurones profond



Réseau de neurones profond (3 couches cachées), *dense*.

**C'est l'apprentissage profond (ajouter une ou plusieurs couches de neurones) qui a tout changé !**

## Exemple œnologique

- On fait tester des vins par des œnologues, qui donnent une note à chaque vin.
- On connaît la composition de chaque vin.

Peut-on prévoir la qualité d'un vin connaissant sa composition ?

# Un exemple œnologique

- 1598 vins sont testés et notés (Q) de 0 à 10.
- 12 paramètres sont mesurés.

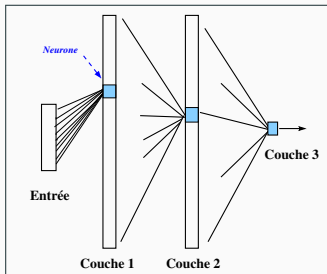
| n    | ac.  | v.ac. | cita. | sugar | chlor. | sulf. | totalsul. | dens. | pH   | sulph. | alc.  | Q    |
|------|------|-------|-------|-------|--------|-------|-----------|-------|------|--------|-------|------|
| 1589 | 6.60 | 0.72  | 0.20  | 7.80  | 0.07   | 29.00 | 79.00     | 1.00  | 3.29 | 0.54   | 9.20  | 5.00 |
| 1590 | 6.30 | 0.55  | 0.15  | 1.80  | 0.08   | 26.00 | 35.00     | 0.99  | 3.32 | 0.82   | 11.60 | 6.00 |
| 1591 | 5.40 | 0.74  | 0.09  | 1.70  | 0.09   | 16.00 | 26.00     | 0.99  | 3.67 | 0.56   | 11.60 | 6.00 |
| 1592 | 6.30 | 0.51  | 0.13  | 2.30  | 0.08   | 29.00 | 40.00     | 1.00  | 3.42 | 0.75   | 11.00 | 6.00 |
| 1593 | 6.80 | 0.62  | 0.08  | 1.90  | 0.07   | 28.00 | 38.00     | 1.00  | 3.42 | 0.82   | 9.50  | 6.00 |
| 1594 | 6.20 | 0.60  | 0.08  | 2.00  | 0.09   | 32.00 | 44.00     | 0.99  | 3.45 | 0.58   | 10.50 | 5.00 |
| 1595 | 5.90 | 0.55  | 0.10  | 2.20  | 0.06   | 39.00 | 51.00     | 1.00  | 3.52 | 0.76   | 11.20 | 6.00 |
| 1596 | 6.30 | 0.51  | 0.13  | 2.30  | 0.08   | 29.00 | 40.00     | 1.00  | 3.42 | 0.75   | 11.00 | 6.00 |
| 1597 | 5.90 | 0.65  | 0.12  | 2.00  | 0.07   | 32.00 | 44.00     | 1.00  | 3.57 | 0.71   | 10.20 | 5.00 |
| 1598 | 6.00 | 0.31  | 0.47  | 3.60  | 0.07   | 18.00 | 42.00     | 1.00  | 3.39 | 0.66   | 11.00 | 6.00 |

Peut-on réaliser un système qui prévoit la qualité (Q) du vin en fonction des 12 paramètres ?

# Le réseau

Réseau à 2 couches cachées, dense.

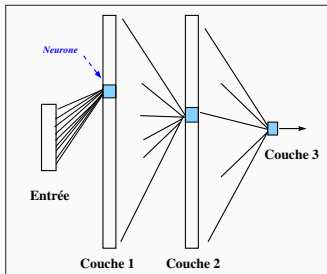
1. Entrée : 12 valeurs.
2. 1<sup>re</sup> couche : 64 neurones.
3. 2<sup>e</sup> couche : 64 neurones.
4. 3<sup>e</sup> couche : 1 neurone, pas de ReLu.



# Le réseau

Réseau à 2 couches cachées, dense.

1. Entrée : 12 valeurs.
2. 1<sup>re</sup> couche : 64 neurones.
3. 2<sup>e</sup> couche : 64 neurones.
4. 3<sup>e</sup> couche : 1 neurone,  
pas de ReLu.

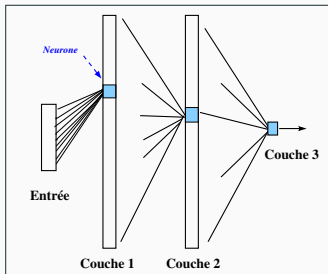


- Chaque neurone de la couche 1 : couplé aux 12 entrées.

# Le réseau

Réseau à 2 couches cachées, dense.

1. Entrée : 12 valeurs.
2. 1<sup>re</sup> couche : 64 neurones.
3. 2<sup>e</sup> couche : 64 neurones.
4. 3<sup>e</sup> couche : 1 neurone,  
pas de ReLu.

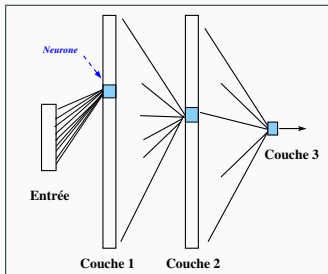


- Chaque neurone de la couche 1 : couplé aux 12 entrées.
- Chaque neurone de la couche 2 : couplé aux 64 sorties de la couche 1.

# Le réseau

Réseau à 2 couches cachées, dense.

1. Entrée : 12 valeurs.
2. 1<sup>re</sup> couche : 64 neurones.
3. 2<sup>e</sup> couche : 64 neurones.
4. 3<sup>e</sup> couche : 1 neurone,  
pas de ReLu.



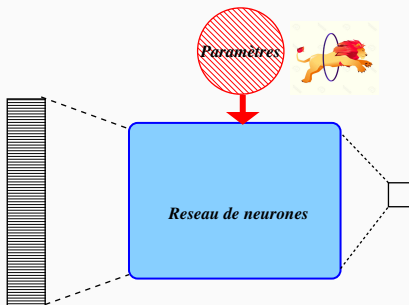
- Chaque neurone de la couche 1 : couplé aux 12 entrées.
- Chaque neurone de la couche 2 : couplé aux 64 sorties de la couche 1.
- Le neurone de la couche 3 est couplé aux 64 sorties de la couche 2.

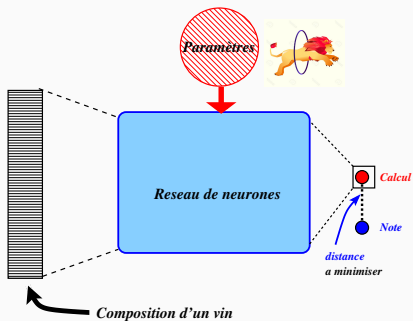
Soit :

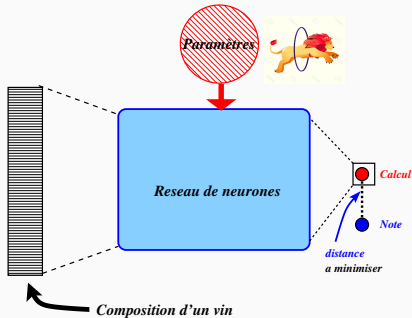
$$64 \times 12 + 64 \times 65 + 65 = 4993 \text{ paramètres à ajuster.}$$

Soit :

$64 \times 12 + 64 \times 65 + 65 = 4993$  paramètres à ajuster.







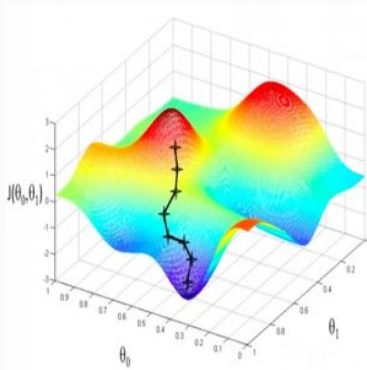
On calcule les valeurs des coefficients des neurones qui minimisent la **somme** des distances, **somme prise pour tous les vins** (qui servent à ajuster le modèle).

**Apprentissage = optimisation**

# Algorithme glouton : descente de gradient

# Algorithme glouton : descente de gradient

On progresse par pas dans la direction de la plus grande pente :



*Aucune garantie d'atteindre le vrai minimum.*

- Calcul du gradient (direction de la plus forte pente) : un problème d'une complexité redoutable ! (la quantité de calculs à effectuer est énorme).

- Calcul du gradient (direction de la plus forte pente) : un problème d'une complexité redoutable ! (la quantité de calculs à effectuer est énorme).

## Important ! Propriété fondamentale !

- L'entraînement (= optimisation) est une opération **extrêmement coûteuse**.
- **Mais** l'utilisation du réseau entraîné est **peu coûteuse**.

## Important ! Propriété fondamentale !

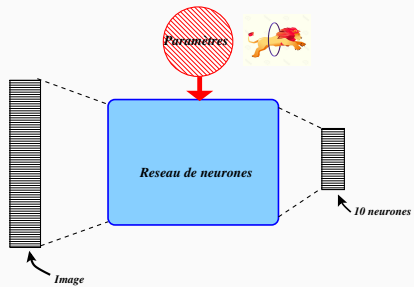
- L'entraînement (= optimisation) est une opération **extrêmement coûteuse**.
- **Mais** l'utilisation du réseau entraîné est **peu coûteuse**.

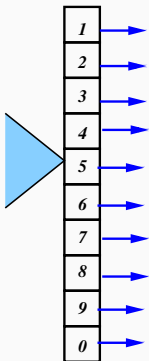
Le coût en nombre d'opérations de l'utilisation est de l'ordre du nombre de paramètres du réseau.

# Classification

## Un exemple : reconnaissance de chiffres manuscrits

- En entrée : une image d'un chiffre (1,...,9,0).
- En sortie : 10 neurones pour la dernière couche.





On voudrait entraîner le réseau de neurones pour que :

1. quand on entre une image de "5", le neurone 5 fournisse 1 et les autres 0.
2. ... et en tout cas que tous fournissent des valeurs entre 0 et 1 (et un score, maximal pour le neurone 5).

Exemple, en sortie :

Softmax donne

0.00575893, 0.00703398, 0.0063646, 0.00949487, 0.94459097,  
0.00526327, 0.00703398, 0.00426632, 0.00547807, 0.00471501.

## Attention !

0.00575893, 0.00703398, 0.0063646, 0.00949487, 0.94459097,  
0.00526327, 0.00703398, 0.00426632, 0.00547807, 0.00471501.

qu'on assimile à une probabilité.

## Attention !

0.00575893, 0.00703398, 0.0063646, 0.00949487, 0.94459097,  
0.00526327, 0.00703398, 0.00426632, 0.00547807, 0.00471501.

qu'on assimile à une probabilité.

**À priori, aucun résultat d'un processus de classification  
n'est sûr à 100 % !**

En vraie grandeur : l'OCR *Tesseract*. Une idée de la complexité des problèmes.

En vraie grandeur : l'OCR *Tesseract*. Une idée de la complexité des problèmes.

Principe :

- Caractère = nombre en machine  
(Exemple : LATIN CAPITAL LETTER A -> 65).

En vraie grandeur : l'OCR *Tesseract*. Une idée de la complexité des problèmes.

Principe :

- Caractère = nombre en machine  
(Exemple : LATIN CAPITAL LETTER A -> 65).
- On prend des textes en docx, pdf, html, etc.  
*Ces fichiers stockent les nombres correspondant aux lettres, et la police de caractères, indépendamment.*

En vraie grandeur : l'OCR *Tesseract*. Une idée de la complexité des problèmes.

Principe :

- Caractère = nombre en machine  
(Exemple : LATIN CAPITAL LETTER A -> 65).
- On prend des textes en docx, pdf, html, etc.  
*Ces fichiers stockent les nombres correspondant aux lettres, et la police de caractères, indépendamment.*

- On rend les caractères sous forme d'images avec différentes polices de caractères.

- On rend les caractères sous forme d'images avec différentes polices de caractères.
- On a donc un grand nombre de correspondances :  
(image)  $\rightarrow$  (nombre = code du caractère)  
pour entraîner l'OCR.

## Entraînement :

- 400 000 lignes de texte,
- 4500 polices de caractères.
- *de l'ordre de 10 milliards d'entrées pour l'entraînement et la vérification !*
- *Plusieurs jours (semaines) de calcul !*

Entraînement :

- 400 000 lignes de texte,
- 4500 polices de caractères.
- *de l'ordre de 10 milliards d'entrées pour l'entraînement et la vérification !*
- *Plusieurs jours (semaines) de calcul !*

*Difficile d'entraîner le système pour des polices exotiques (exemple : polices du XVIII<sup>e</sup> siècle).*

# Calculus

---

**Calcul : ce qui rend l'entraînement possible :  
« heureuse coïncidence »**

Entraînement :

- énorme quantité de données,
- énorme quantité de calculs.
- Pas besoin d'une grande précision.

## Entraînement :

- énorme quantité de données,
- énorme quantité de calculs.
- Pas besoin d'une grande précision.
- Il **faut** paralléliser les calculs.
- Calculs aisément *parallélisables*.

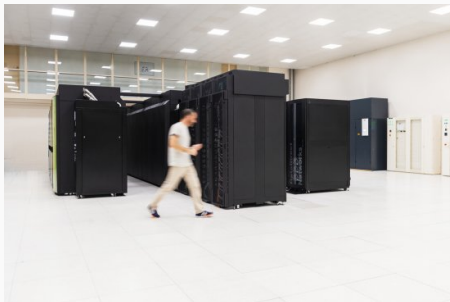
Entraînement :

- énorme quantité de données,
- énorme quantité de calculs.
- Pas besoin d'une grande précision.
- Il **faut** paralléliser les calculs.
- Calculs aisément *parallélisables*.

Heureuse coïncidence : le développement des processeurs graphiques (**GPU**).

## Les deux sortes de parallélisme :

- Processeurs à  $n$  cœurs,
- Super calculateurs (réseaux de machines classiques, avec processeurs à  $n$  cœurs). Exemple : TaiHu, 16 millions de cœurs.



Calculateur Jean Zay (CNRS/IDRIS, Orsay)

- SIMD (Single Instruction, Multiple Data) :
  - Exemple : additionner ou multiplier 2 tableaux en une seule instruction.

- **SIMD** (Single Instruction, Multiple Data) :
  - Exemple : additionner ou multiplier 2 tableaux en une seule instruction.

**GPU.**

Graphic Processing Unit.



- **SIMD** (Single Instruction, Multiple Data) :
  - Exemple : additionner ou multiplier 2 tableaux en une seule instruction.

**GPU.**

Graphic Processing Unit.



**Chance !** les calculs sont bien adaptés à ce type de parallélisme.

## Performances :

- CPU : 64 Gigaflops par cœur ( $= 64 \cdot 10^9$ ).
- NVIDIA H100 : 134 Teraflops ( $= 134 \cdot 10^{12}$ ) (et même 3000 teraflops dans certains cas) !

Performances :

- CPU : 64 Gigaflops par cœur ( $= 64 \cdot 10^9$ ).
- NVIDIA H100 : 134 Teraflops ( $= 134 \cdot 10^{12}$ ) (et même 3000 teraflops dans certains cas) !

Coût : 40 000 euros.

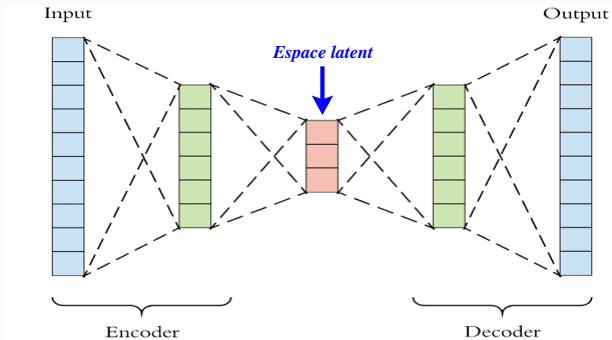
Nvidia peut les vendre par « cabinets » de 4608 GPU.

## **Quelques exemples d'applications curieuses et/ou intéressantes**

---

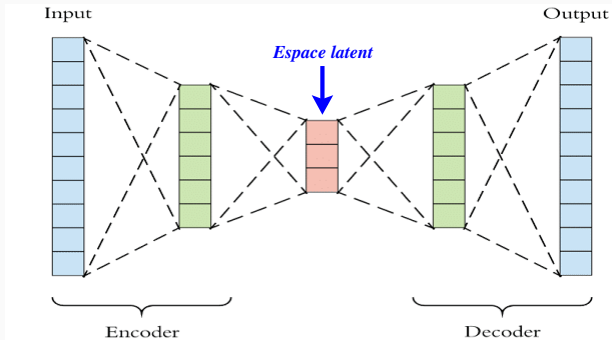
# Auto encodeurs

---

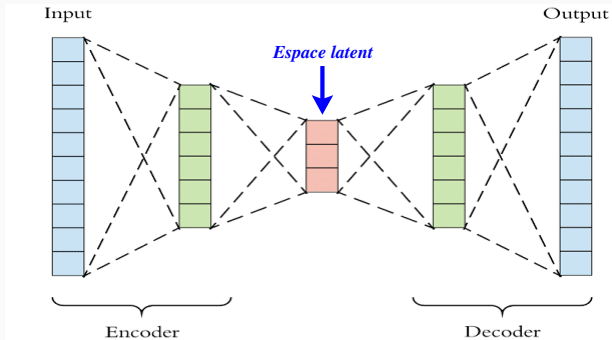


- En entrée : disons, une image.
- En sortie : retrouver *presque* la même image.

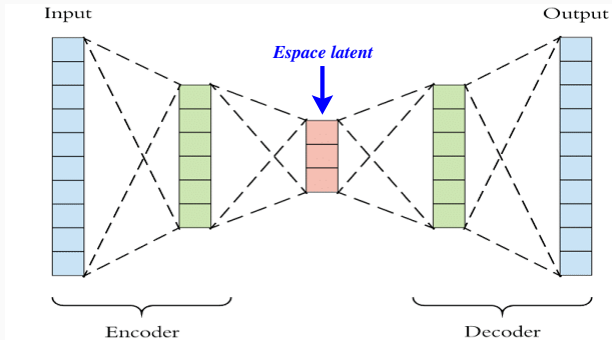
en passant par un espace de petite dimension.



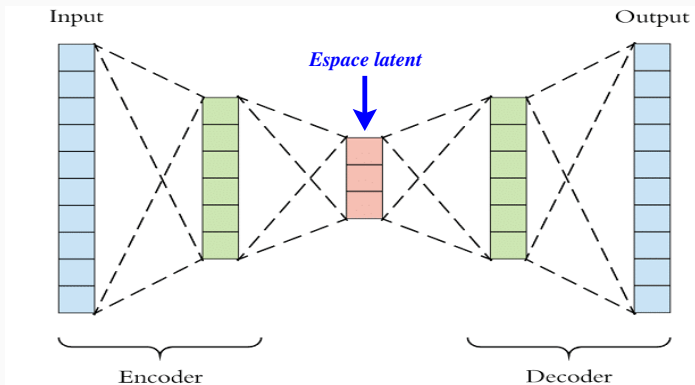
- Image : millions de pixels = millions de valeurs.



- Image : millions de pixels = millions de valeurs.
- Image = un point dans un espace à quelques millions de dimensions.



- Image : millions de pixels = millions de valeurs.
- Image = un point dans un espace à quelques millions de dimensions.
- Espace latent : petite dimension (2...10).



De manière étonnante, ça fonctionne !

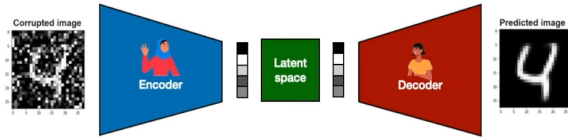


De manière étonnante, ça fonctionne !



Espace latent : représentation *parcimonieuse* des images.

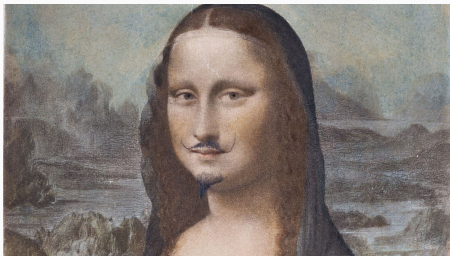
## Plus fort ! Auto-encodeurs débruiteurs





Apprentissage non supervisé.

## Autoencodeurs variationnels : ou comment mettre des moustaches à la Joconde



En parcourant l'espace latent :



IA *généraliste*.

En parcourant l'espace latent :



IA *générative*.

On peut sûrement mettre des moustaches à la Joconde...

# Le langage naturel

---

Le bas de gamme, genre SMS :

« *Salut Anatole, je vais passer ???* »

Le bas de gamme, genre SMS :

« *Salut Anatole, je vais passer ???* »

1. Découper la phrase en *jetons* (tokens).

Le bas de gamme, genre SMS :

« *Salut Anatole, je vais passer ???* »

1. Découper la phrase en *jetons* (tokens).
2. Associer à chaque *jeton* un ou plusieurs nombres.

Le bas de gamme, genre SMS :

« *Salut Anatole, je vais passer ???* »

1. Découper la phrase en *jetons* (tokens).
2. Associer à chaque *jeton* un ou plusieurs nombres.
3. Retour aux méthodes précédentes.

Le bas de gamme, genre SMS :

« *Salut Anatole, je vais passer ???* »

1. Découper la phrase en *jetons* (tokens).
2. Associer à chaque *jeton* un ou plusieurs nombres.
3. Retour aux méthodes précédentes.

*Ce mécanisme ne regarde qu'en arrière.*

Innovation : *le mécanisme d'attention.*

Regarder en avant et en arrière.

Exemple de logiciel : BERT (Bidirectional Encoder Representations from Transformers).

Entraînement sur Wikipédia.

- J'ai mangé un *avocat* au restaurant.
- Au restaurant, j'ai rencontré un *avocat*.

- J'ai mangé un *avocat* au restaurant.
- Au restaurant, j'ai rencontré un *avocat*.

Entraînement :

- J'ai mangé un ???? au restaurant.
- J'ai rencontré un ???? au restaurant.

Il faut un moyen de coder le couplage entre ???? et les mots de la phrase.

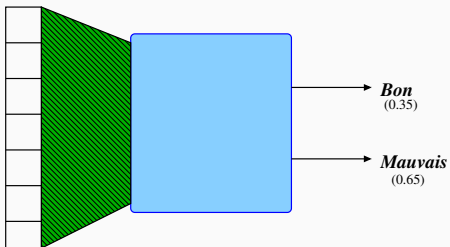
BERT entraîné sur un corpus français pour donner :  
FLAUBERT et

BERT entraîné sur un corpus français pour donner :  
FLAUBERT et CAMEMBERT.

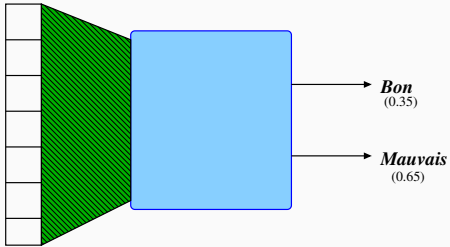
# Dangers

---





Dites bonjour au crédit social !



Dites bonjour au crédit social !

**Rappel : à priori, aucun résultat d'un processus de classification n'est sûr à 100 % !**

Un exemple donné par Stéphane Mallat (Collège de France,  
Médaille d'or du CNRS 2025) :

*système d'aide à la décision pour la libération des prisonniers aux  
États Unis.*

Un exemple donné par Stéphane Mallat (Collège de France, Médaille d'or du CNRS 2025) :

*système d'aide à la décision pour la libération des prisonniers aux États Unis.*

Pose la question : « Pourquoi et comment ça fonctionne ? ».

- Confondre corrélation et causalité.

- Confondre corrélation et causalité.
- On ne peut prévoir au mieux que ce qui est contenu dans les données.

... et ses conséquences pour la vie privée.

**Mais pourquoi est-ce que ça  
fonctionne ?**

---

Pourquoi est-ce que ça fonctionne (assez  
correctement) ?

On ne sait pas trop !

Nécessité de comprendre (voir ci-dessus le problème des prisonniers).

*Voir par exemple l'exposé de Stéphane Mallat  
(Collège de France) à la MMI.*



Pistes :

- Régularité.

*Voir par exemple l'exposé de Stéphane Mallat  
(Collège de France) à la MMI.*



Pistes :

- Régularité.
- Parcimonie.

*Voir par exemple l'exposé de Stéphane Mallat  
(Collège de France) à la MMI.*



Pistes :

- Régularité.
- Parcimonie.
- Liens avec différentes branches des maths et de la physique (invariance).
- ...

Tout ou presque reste à faire.

**Intelligence, vraiment ?**

---

N'oublions pas qu'on utilise des ordinateurs...

N'oublions pas qu'on utilise des ordinateurs... qui exécutent des programmes.

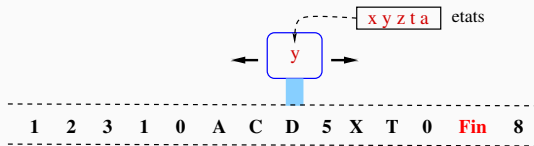
N'oublions pas qu'on utilise des ordinateurs... qui exécutent des programmes. Les programmes implantent des algorithmes.

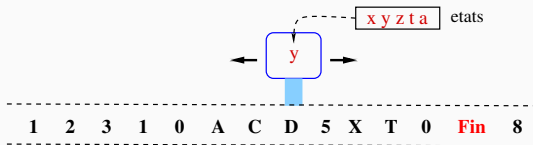
N'oublions pas qu'on utilise des ordinateurs... qui exécutent des programmes. Les programmes implantent des algorithmes.

**Que peut-on (ou ne peut-on pas) faire dans ce cadre ?**

## Machines de Turing (Alan Turing, 1936) et calculabilité

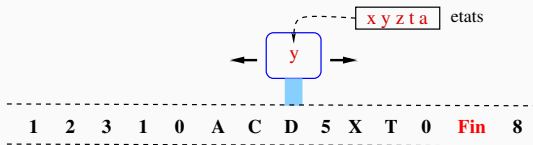
Preuve par Turing de l'impossibilité du *Calculus Ratiocinator* de Leibnitz (et de l'*Entscheidungsproblem*).





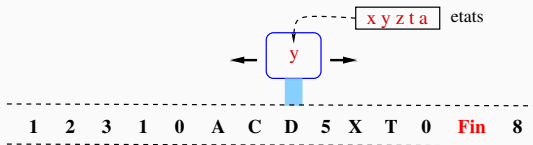
Un pas de la machine :

- lecture d'un symbole sur la bande.



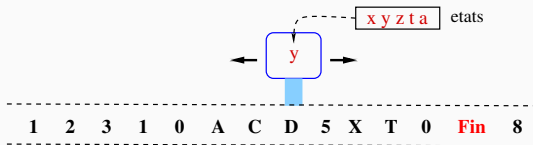
Un pas de la machine :

- lecture d'un symbole sur la bande.
- **table de transition** : en fonction de l'état de la tête et du symbole lu :



Un pas de la machine :

- lecture d'un symbole sur la bande.
- **table de transition** : en fonction de l'état de la tête et du symbole lu :
  - écriture éventuelle sur la bande,
  - déplacement éventuel à gauche ou à droite,
  - modification éventuelle de l'état de la tête.
  - arrêt si le symbole lu est **Fin**.



Un pas de la machine :

- lecture d'un symbole sur la bande.
- **table de transition** : en fonction de l'état de la tête et du symbole lu :
  - écriture éventuelle sur la bande,
  - déplacement éventuel à gauche ou à droite,
  - modification éventuelle de l'état de la tête.
  - arrêt si le symbole lu est **Fin**.

**Exemple de transition** :  $\{ y, \mathbf{D} \} \Rightarrow \{ t, \mathbf{X}, \leftarrow \}$

- Les machines de Turing sont « universelles » (base de la *calculabilité*).

- Les machines de Turing sont « universelles » (base de la *calculabilité*).
- Les ordinateurs, munis de langages de programmation sont « Turing équivalents ».

- Les machines de Turing sont « universelles » (base de la *calculabilité*).
- Les ordinateurs, munis de langages de programmation sont « Turing équivalents ».
- Donc, ce que peut faire l'IA (l'apprentissage profond), c'est **au plus** ce que peut faire une machine de Turing.

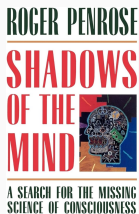
- Les machines de Turing sont « universelles » (base de la *calculabilité*).
- Les ordinateurs, munis de langages de programmation sont « Turing équivalents ».
- Donc, ce que peut faire l'IA (l'apprentissage profond), c'est **au plus** ce que peut faire une machine de Turing.
- Le cerveau est-il « Turing équivalent » ?
- La pensée, la conscience, l'intelligence ne sont-elles que des algorithmes ?

### Les idées de Roger Penrose

## Les idées de Roger Penrose



- Arguments basés sur le *théorème de l'arrêt* (Turing) et les résultats de Gödel.



## Les idées de Roger Penrose



- Arguments basés sur le *théorème de l'arrêt* (Turing) et les résultats de Gödel.
- Si conscience = algorithmes, quel peut avoir été le rôle de l'évolution ?



## Les idées de Roger Penrose



- Arguments basés sur le *théorème de l'arrêt* (Turing) et les résultats de Gödel.
- Si conscience = algorithmes, quel peut avoir été le rôle de l'évolution ?
- Roger Penrose s'intéresse aux *éclairs de compréhension*.

## Les idées de Roger Penrose



- Arguments basés sur le *théorème de l'arrêt* (Turing) et les résultats de Gödel.
- Si conscience = algorithmes, quel peut avoir été le rôle de l'évolution ?
- Roger Penrose s'intéresse aux *éclairs de compréhension*.
- Existence de phénomènes quantiques dans le cerveau ? ( $\Rightarrow$  *Quantum mind*).

## « Quantum mind »

Penrose, Hameroff et pas mal d'autres (ainsi qu'E. Wigner auparavant).

## « Quantum mind »

Penrose, Hameroff et pas mal d'autres (ainsi qu'E. Wigner auparavant).

- Existence de phénomènes quantiques *non calculables*.
- Candidats : micro tubules.

Beaucoup de réfutations, mais une expérience plutôt positive en avril 2022.

## Du côté de Noam Chomsky

*The New York Times* mars 2023.

- « Il est à la fois comique et tragique que tant d'argent et d'attention se concentrent sur si peu – quelque chose de si trivial comparé à l'esprit humain qui, par le biais du langage, pour reprendre les mots de Wilhelm von Humboldt, peut faire « un usage infini de moyens finis », élaborant des idées et des théories ayant une portée universelle. »

- « J'ai fini par demander à ChatGPT s'il connaissait l'idée de banalité du mal et s'il se sentait concerné. Voilà sa réponse :

- « J'ai fini par demander à ChatGPT s'il connaissait l'idée de banalité du mal et s'il se sentait concerné. Voilà sa réponse :  
*«Il est vrai que je suis un outil créé par des humains, et par conséquent, je peux refléter les limites et les biais de mes créateurs et des données sur lesquelles je suis entraîné. »*

- « J'ai fini par demander à ChatGPT s'il connaissait l'idée de banalité du mal et s'il se sentait concerné. Voilà sa réponse :  
*«Il est vrai que je suis un outil créé par des humains, et par conséquent, je peux refléter les limites et les biais de mes créateurs et des données sur lesquelles je suis entraîné. »*  
Une intelligence servile et sans pensée, c'est en effet une bonne définition de ce que Hannah Arendt cherchait à saisir à travers l'idée de la **banalité du mal.** »

# Questions de complexité

---

## Un petit détour

## Un petit détour

|       |   |   |   |   |  |
|-------|---|---|---|---|--|
|       |   | 2 | 3 | 1 |  |
|       | × | 1 | 2 | 4 |  |
| <hr/> |   |   |   |   |  |
|       |   | 9 | 2 | 4 |  |
|       | 4 | 6 | 2 |   |  |
| 2     | 3 | 1 |   |   |  |
| <hr/> |   |   |   |   |  |
| 2     | 8 | 6 | 4 | 4 |  |

Opérations élémentaires : les multiplications entre chiffres.

## Un petit détour

|   |   |   |   |   |  |
|---|---|---|---|---|--|
|   |   | 2 | 3 | 1 |  |
|   | × | 1 | 2 | 4 |  |
|   |   | 9 | 2 | 4 |  |
|   | 4 | 6 | 2 |   |  |
| 2 | 3 | 1 |   |   |  |
| 2 | 8 | 6 | 4 | 4 |  |

Opérations élémentaires : les multiplications entre chiffres.

- 3 multiplications par ligne,
- 3 lignes.

Soit en tout 9 opérations pour des nombres de 3 chiffres.

## Un petit détour

|       |   |   |   |   |  |
|-------|---|---|---|---|--|
|       |   | 2 | 3 | 1 |  |
|       | × | 1 | 2 | 4 |  |
| <hr/> |   |   |   |   |  |
|       |   | 9 | 2 | 4 |  |
|       | 4 | 6 | 2 |   |  |
| 2     | 3 | 1 |   |   |  |
| <hr/> |   |   |   |   |  |
| 2     | 8 | 6 | 4 | 4 |  |

Opérations élémentaires : les multiplications entre chiffres.

- 3 multiplications par ligne,
- 3 lignes.

Soit en tout 9 opérations pour des nombres de 3 chiffres.

- avec des nombres de 4 chiffres :  $4 \times 4 = 16$  opérations,
- avec des nombres de  $n$  chiffres :  $n \times n = n^2$  opérations.

Certes, on n'a pas compté les additions ni les retenues, mais c'est du même ordre :  $n^2$ .

Certes, on n'a pas compté les additions ni les retenues, mais c'est du même ordre :  $n^2$ .

Finalement, la complexité (le coût) de l'algorithme est proportionnelle à  $n^2$  :

$$C_n = K \times n^2.$$

.

Certes, on n'a pas compté les additions ni les retenues, mais c'est du même ordre :  $n^2$ .

Finalement, la complexité (le coût) de l'algorithme est proportionnelle à  $n^2$  :

$$C_n = K \times n^2.$$

.

### Implication

Si on **double la taille** des nombres à multiplier, le coût devient :

$$C_{2n} = K \times (2 \times n)^2$$

Certes, on n'a pas compté les additions ni les retenues, mais c'est du même ordre :  $n^2$ .

Finalement, la complexité (le coût) de l'algorithme est proportionnelle à  $n^2$  :

$$C_n = K \times n^2.$$

.

### Implication

Si on **double la taille** des nombres à multiplier, le coût devient :

$$C_{2n} = K \times (2 \times n)^2 = 4 \times K \times n^2,$$

Certes, on n'a pas compté les additions ni les retenues, mais c'est du même ordre :  $n^2$ .

Finalement, la complexité (le coût) de l'algorithme est proportionnelle à  $n^2$  :

$$C_n = K \times n^2.$$

.

### Implication

Si on **double la taille** des nombres à multiplier, le coût devient :

$$C_{2n} = K \times (2 \times n)^2 = 4 \times K \times n^2,$$

$$C_{2n} = 4 \times C_n.$$

Certes, on n'a pas compté les additions ni les retenues, mais c'est du même ordre :  $n^2$ .

Finalement, la complexité (le coût) de l'algorithme est proportionnelle à  $n^2$  :

$$C_n = K \times n^2.$$

.

### Implication

Si on **double la taille** des nombres à multiplier, le coût devient :

$$C_{2n} = K \times (2 \times n)^2 = 4 \times K \times n^2,$$

$$C_{2n} = 4 \times C_n.$$

**Le coût (la complexité) est multiplié par 4.**

Forcément, il existe une valeur de  $n$  au delà de laquelle on ne pourra pas effectuer la multiplication !

Forcément, il existe une valeur de  $n$  au delà de laquelle on ne pourra pas effectuer la multiplication !

Il existe des problèmes dont la *complexité algorithmique* est telle qu'on ne peut pas les résoudre !

Forcément, il existe une valeur de  $n$  au delà de laquelle on ne pourra pas effectuer la multiplication !

Il existe des problèmes dont la *complexité algorithmique* est telle qu'on ne peut pas les résoudre !

Qu'en est-il des problèmes d'IA ? On ne sait pas !

Que penser des centres de calcul prévus qui consommeront 1 GW ? (CCIN2P3 à Villeurbanne : 15 MW).

## Remarques finales

---

Comment avoir des crédits ?

Comment avoir des crédits ?

Dites « cerveau électronique » plutôt qu'ordinateur, « Intelligence Artificielle » plutôt qu'Apprentissage profond.

Comment avoir des crédits ?

Dites « cerveau électronique » plutôt qu'ordinateur, « Intelligence Artificielle » plutôt qu'Apprentissage profond.

L'histoire de la traduction automatique.

# Perles catastrophiques

## Perles catastrophiques

- De la ministre du Numérique et de l'IA : *des cours d'IA obligatoires en quatrième et en seconde.*

## Perles catastrophiques

- De la ministre du Numérique et de l'IA : *des cours d'IA obligatoires en quatrième et en seconde.*
- D'un député, juriste : « J'ai proposé que chaque plateforme soit obligée d'intégrer une IA qui coupe immédiatement tous les contenus illégaux ».

Thierry Dumont :

mail : `thierry@thierry-dumont.fr`

Diapos disponibles sur : `https://thierry-dumont.fr`