

Introduction à l'intelligence artificielle

Olivier Canévet

24 mars 2023



Lien vers les ressources

Le contenu de cette présentation ainsi que le notebook Python pour les exemples de code est disponible à l'adresse suivante :



Plan de déroulement

Introduction

Applications de l'IA

Données

Classification

Réseaux de neurones artificiels

Apprentissage profond

Conclusion

Plan

Introduction

Applications de l'IA

Données

Classification

Réseaux de neurones artificiels

Apprentissage profond

Conclusion

Objectifs

La question principale à laquelle nous allons répondre est

Comment “la machine apprend-elle toute seule” ?

Le but de ce séminaire est de comprendre les points suivants :

- qu'est-ce qu'une donnée d'apprentissage ?
- qu'est-ce qu'un réseau de neurones artificiels ?
- qu'est-ce que la descente et la rétro-propagation du gradient ?
- comment entraîne-t-on un réseau de neurones avec Python et PyTorch ?

Intelligence artificielle (IA)

Ensemble des théories et des techniques développant des programmes informatiques complexes capables de reproduire certaines tâches nécessitant de l'intelligence humaine (raisonnement, apprentissage, prise de décision). (Wikipedia, 2023c)

¹On utilise l'orthographe rectifiée de 1990

Définitions

Intelligence artificielle (IA)

Ensemble des théories et des techniques développant des programmes informatiques complexes capables de reproduire certaines tâches nécessitant de l'intelligence humaine (raisonnement, apprentissage, prise de décision). (Wikipedia, 2023c)

Apprentissage automatique (*machine learning*)

Ensemble des théories et des techniques mathématiques et statistiques mises au point pour donner la possibilité aux ordinateurs d'apprendre à résoudre des problèmes et des tâches à partir de données, d'améliorer leurs performances, sans être explicitement programmés. (Wikipedia, 2023a)

¹On utilise l'orthographe rectifiée de 1990

Définitions

Intelligence artificielle (IA)

Ensemble des théories et des techniques développant des programmes informatiques complexes capables de reproduire certaines tâches nécessitant de l'intelligence humaine (raisonnement, apprentissage, prise de décision). (Wikipedia, 2023c)

Apprentissage automatique (*machine learning*)

Ensemble des théories et des techniques mathématiques et statistiques mises au point pour donner la possibilité aux ordinateurs d'apprendre à résoudre des problèmes et des tâches à partir de données, d'améliorer leurs performances, sans être explicitement programmés. (Wikipedia, 2023a)

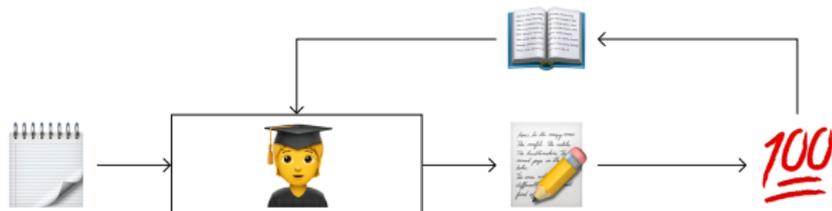
Entraîner un modèle

Entraîner¹ un modèle à effectuer une tâche signifie trouver les “bonnes valeurs” pour les paramètres du modèle, grâce à un algorithme d'optimisation, pour minimiser l'erreur que fait le modèle sur les données d'apprentissage.

¹On utilise l'orthographe rectifiée de 1990

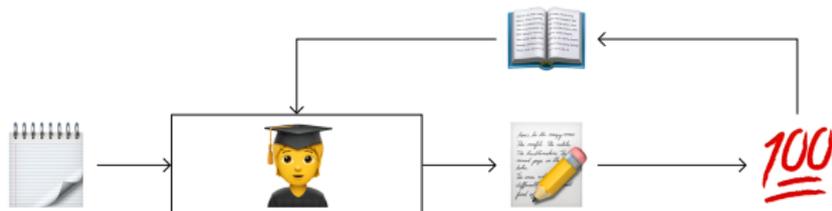
Analogie avec l'apprentissage scolaire

Durant l'année, les élèves apprennent au fil d'exercices notés qui leur permettent d'évaluer leurs performances.



Analogie avec l'apprentissage scolaire

Durant l'année, les élèves apprennent au fil d'exercices notés qui leur permettent d'évaluer leurs performances.

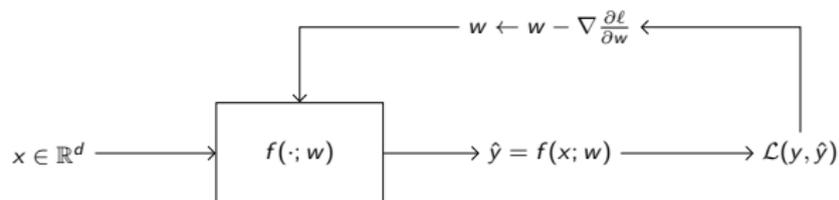


Le jour de l'examen, les élèves sont évalués sur des exercices similaires (mais pas identiques) pour évaluer leur capacité de généralisation.



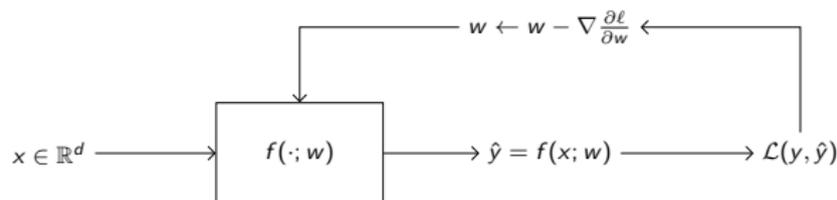
Étapes lors de l'apprentissage machine

En apprentissage automatique (*machine learning*), on donne au modèle des exemples d'apprentissage qui sont traités pour produire une réponse. Cette réponse est évaluée par une fonction de coût (*loss*) pour finalement mettre à jour le modèle, par exemple ici, avec la descente de gradient.

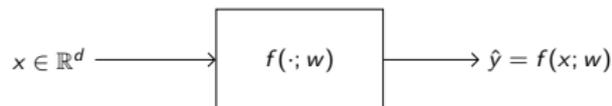


Étapes lors de l'apprentissage machine

En apprentissage automatique (*machine learning*), on donne au modèle des exemples d'apprentissage qui sont traités pour produire une réponse. Cette réponse est évaluée par une fonction de coût (*loss*) pour finalement mettre à jour le modèle, par exemple ici, avec la descente de gradient.



À la fin de l'entraînement, on évalue les performances finales du modèle sur des données de test qui n'ont pas été vues durant l'apprentissage.



Plan

Introduction

Applications de l'IA

Données

Classification

Réseaux de neurones artificiels

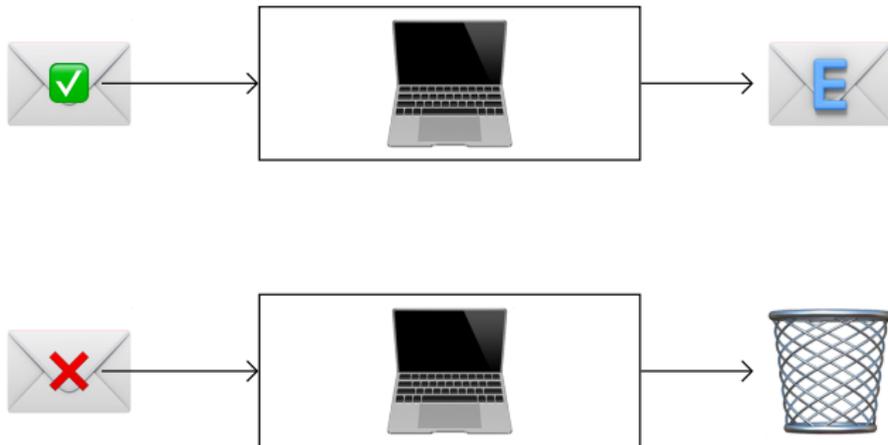
Apprentissage profond

Conclusion

Filtre anti-spam

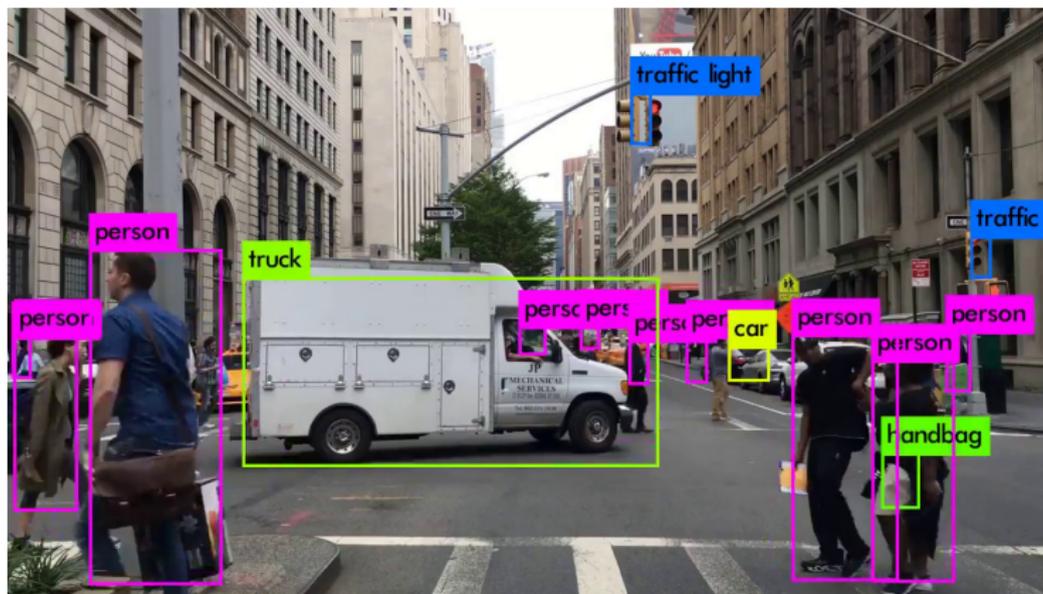
Lorsqu'on reçoit un e-mail, le service de messagerie applique un filtre pour décider

- si un e-mail est légitime et peut être placé dans la boîte de réception, ou
- s'il est un spam et peut être placé dans le dossier d'indésirables.



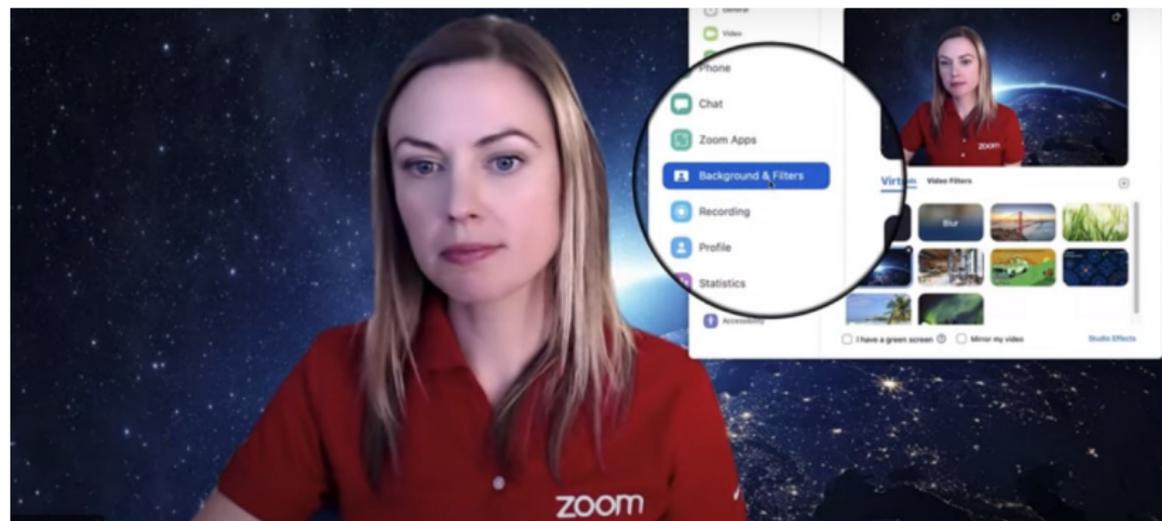
Détection d'objets

La détection d'objets consiste à trouver la localisation et la taille de toutes les instances d'un type d'objet donnée. Ceci peut servir pour l'assistance à la conduite, la détection d'anomalies sur les chaînes de production, l'analyse de trafic routier, la détection de plaque d'immatriculation.



Segmentation d'image

La segmentation d'image consiste à attribuer un label à chaque pixel. Par exemple, le type d'objet : personne ou arrière-plan. Cela permet de s'affranchir du besoin d'un "fond vert" dans le cas du remplacement d'arrière plan.



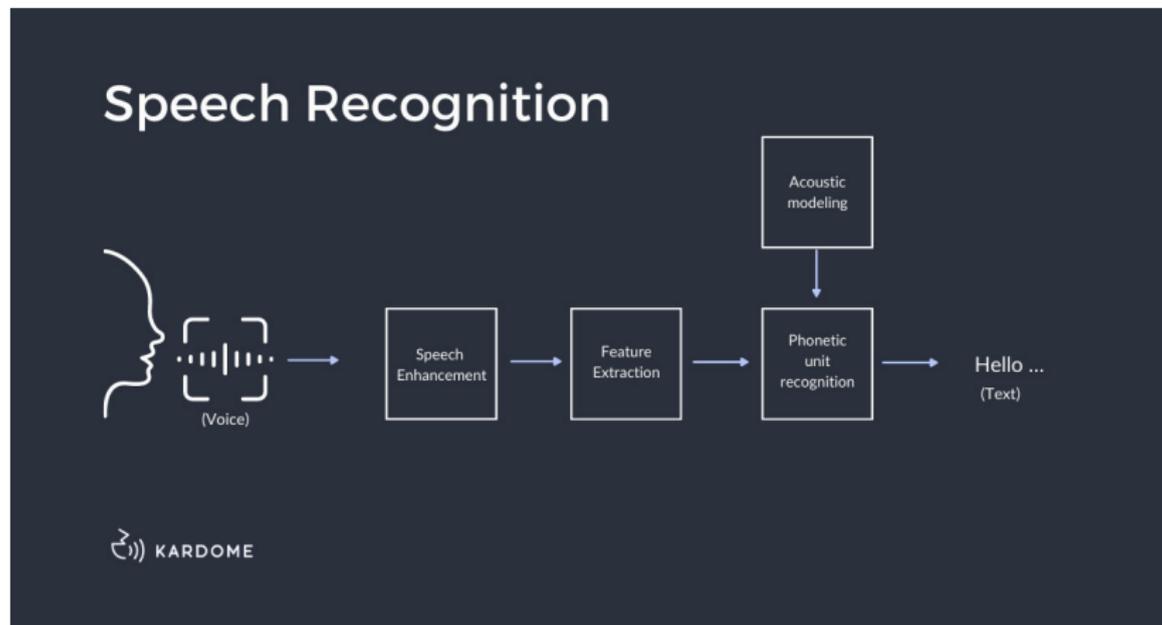
Super résolution et colorisation cinématographique

Les écrans de télévision modernes ont une résolution bien supérieure aux films tournés il y a longtemps.



Reconnaissance automatique de la parole

La reconnaissance de la parole (*speech recognition*) consiste à analyser la voix humaine captée par un microphone pour la transcrire sous la forme d'un texte exploitable par une machine. (Wikipedia, 2023d)



Apprentissage par renforcement

L'apprentissage par renforcement consiste, pour un agent autonome à apprendre les actions à prendre (aller à droite, à gauche, etc.), à partir d'expériences, de façon à optimiser une récompense quantitative au cours du temps. (Wikipedia, 2023b)



Plan

Introduction

Applications de l'IA

Données

Classification

Réseaux de neurones artificiels

Apprentissage profond

Conclusion

Données structurées

Les données les plus classiques à traiter sont sous la forme d'un tableau.

Par exemple, la base de données de fleurs "Iris" (Fisher, 1936) contient 150 entrées avec les longueurs et largeurs des sépales et des pétales de 3 espèces d'iris différentes :



Sepal length	Sepal width	Petal length	Petal width	Class
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
6.9	3.1	4.9	1.5	versicolor
5.5	2.3	4.0	1.3	versicolor
7.7	2.8	6.7	2.0	virginica
6.3	2.7	4.9	1.8	virginica

Chaque élément de la base de données est décrit par des caractéristiques (les *features*), ici il y a 4 valeurs et une étiquette (le *label*). Mathématiquement, on note

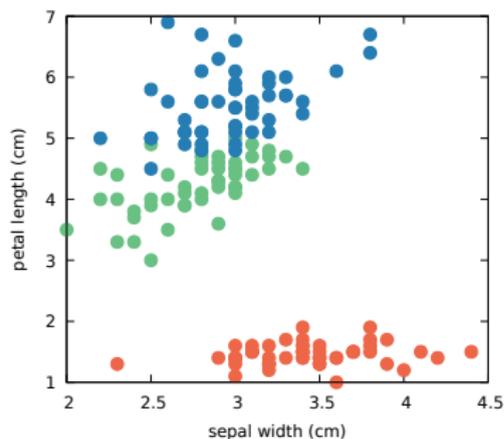
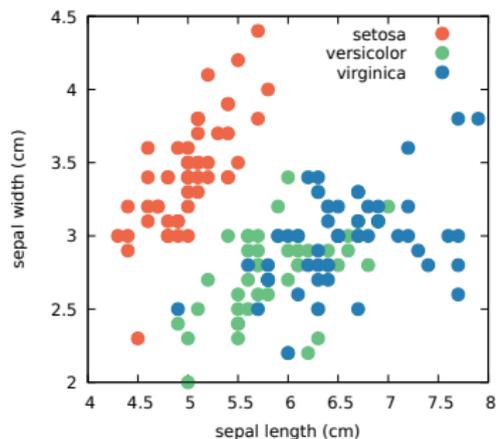
$$(x_n, y_n) \in \mathbb{R}^4 \times \{0, 1, 2\}$$

"Blue flag flower close-up" par Danielle Langlois (CC BY-SA 3.0)

"Iris virginica shrevei BLUE FLAG" par Frank Mayfield CC BY-SA 2.0

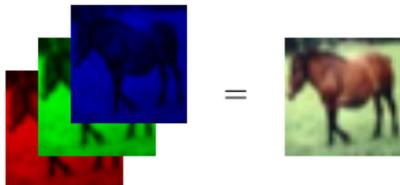
Visualisation de la base "Iris"

Comme les données sont dans \mathbb{R}^4 il est difficile de les représenter entièrement. On peut cependant les visualiser par paire de *features* : (x_1, x_2) , (x_1, x_3) , \dots



Images

Une image numérique est un tableau (ou matrice) à plusieurs dimensions : la hauteur, la largeur, et la profondeur (comme rouge, vert, bleu).



Images

Une image numérique est un tableau (ou matrice) à plusieurs dimensions : la hauteur, la largeur, et la profondeur (comme rouge, vert, bleu).



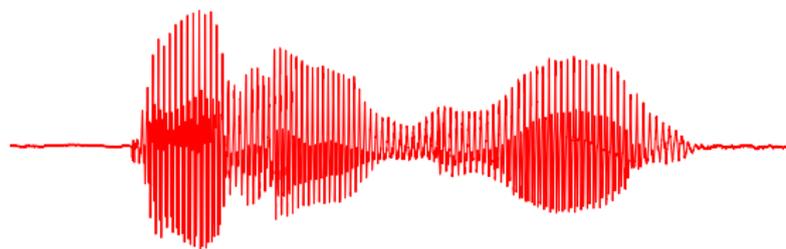
En mémoire, les pixels sont stockés de manière contigue, dans un tableau d'entiers, souvent codés sur 8 bits, de 0 à 255 :

```
28 30 33 62 63 31 29 42 55 67 92 76 57 75 69 57 74 98 86 71 59 62 57
42 51 46 41 38 37 43 52 46 27 27 21 38 60 39 41 47 48 72 120 103 66 75
110 134 146 153 146 139 125 130 109 76 82 85 82 91 99 112 117 115 34
33 24 38 82 52 43 55 65 113 157 142 108 108 152 185 179 178 180 184
182 177 171 165 166 168 165 167 169 175 177 176 45 49 46 55 83 62 44
45 54 91 108 100 119 144 177 194 191 198 206 205 200 200 193 192 194
194 203 199 192 191 189 187 77 84 80 53 44 46 37 38 47 58 61 57 77 102
155 199 218 228 235 229 217 211 188 174 168 159 179 181 189 190 190
192 149 138 81 56 34 31 40 55 66 85 101 101 101 104 107 124 188 232
...
```

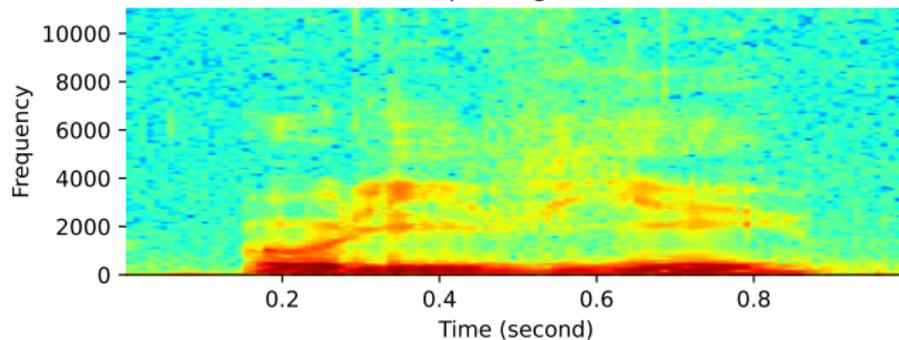
Audio

Un signal audio numérique est représenté par une suite de valeurs correspondant aux variations de pression qui se sont exercées sur le microphone.

Audio signal



Spectrogram



Plan

Introduction

Applications de l'IA

Données

Classification

Réseaux de neurones artificiels

Apprentissage profond

Conclusion

Classification

La classification consiste à attribuer une étiquette (un *label*) à une donnée d'entrée.

Si on a C catégories (ou classes), les données que l'on utilise pour l'entraînement sont

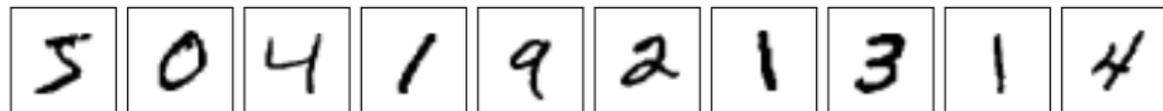
$$(x_n, y_n) \in \mathbb{R}^d \times \{1, \dots, C\}$$

et on cherche à entraîner un *classifieur* f de telle sorte que $f(x_n) = y_n$.

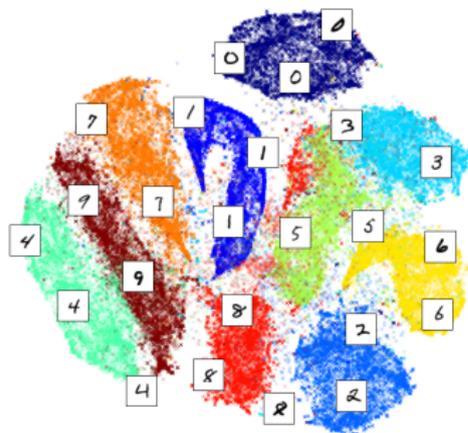
On évalue généralement un *classifieur* en comptant le nombre d'erreurs qu'il fait sur les données de test.

Base de données de chiffres manuscrits MNIST

MNIST est une base de données d'images en niveau de gris, de taille 28×28 représentant des chiffres de 0 à 9. Il y a 60000 pour l'entraînement, et 10000 pour l'évaluation. On est donc "en dimension 784" (28×28).



Entraîner un classifieur consiste à trouver une fonction qui soit capable de délimiter les classes dans l'espace des données.



Données accessibles à l'adresse <http://yann.lecun.com/exdb/mnist/>

Représentation t-SNE tiré de https://bigsnarf.files.wordpress.com/2016/11/tsne_mnist_all.png

Plan

Introduction

Applications de l'IA

Données

Classification

Réseaux de neurones artificiels

Apprentissage profond

Conclusion

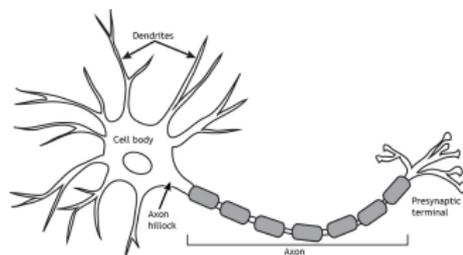
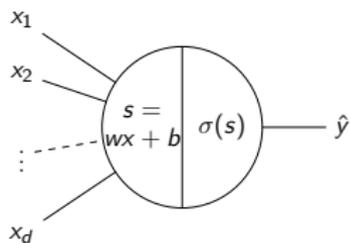
Composant de base d'un réseau de neurones

Le composant essentiel d'un réseau de neurones est le "neurone artificiel" qui calcule une somme pondérée suivie d'une non-linéarité :

$$\mathbb{R}^d \rightarrow \mathbb{R}$$

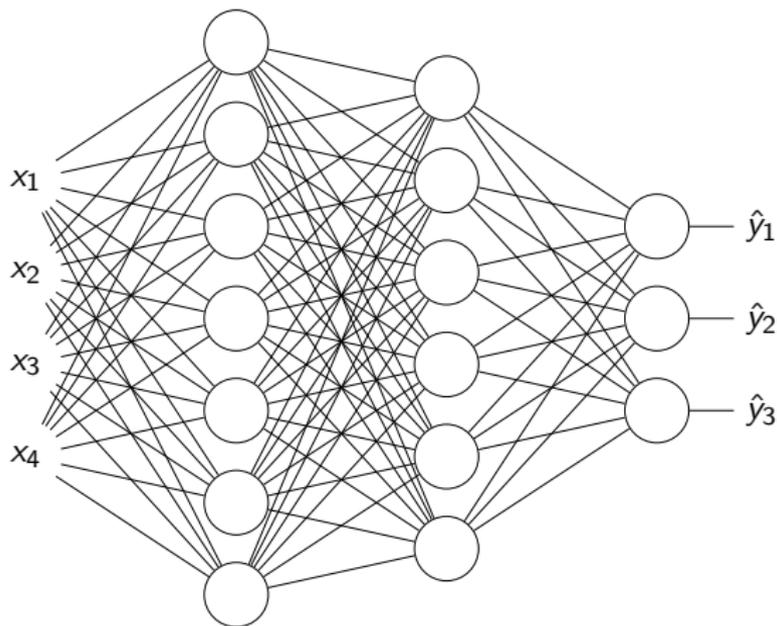
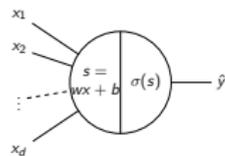
$$x \mapsto \sigma(w^\top x + b) = \sigma\left(b + \sum_i w_i x_i\right)$$

où $x \in \mathbb{R}^d$ est la donnée d'entrée (de dimension d), $w \in \mathbb{R}^d$ les poids du neurone, et $b \in \mathbb{R}$ l'ordonnée à l'origine, appelé *bias* en *machine learning*.

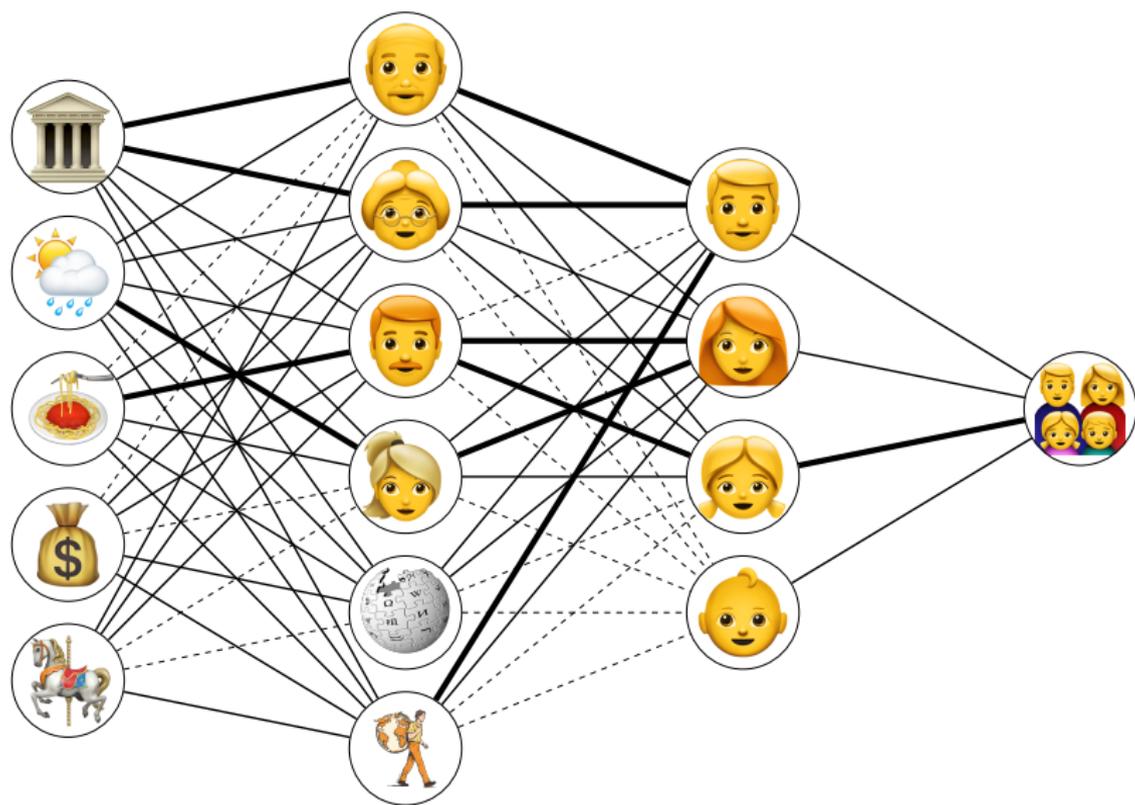


Réseau de neurones multi-couches

Chaque neurone d'une couche traite les sorties de la couche précédente.



Analogie avec un repas de famille



Descente de gradient

Pour trouver le minimum local d'une fonction, minimum pour lequel il n'existe pas de forme analytique, on peut utiliser la descente de gradient. Cet algorithme utilise une information locale à la fonction pour se déplacer dans la direction de valeurs plus petite.

Si on considère la fonction

$$\mathcal{L} : \mathbb{R}^d \rightarrow \mathbb{R}$$

on peut l'approximer au voisinage de $w_0 \in \mathbb{R}^d$ à l'ordre 2

$$\tilde{\mathcal{L}}_{w_0}(w) = \mathcal{L}(w_0) + \nabla \mathcal{L}(w_0)^\top (w - w_0) + \frac{1}{2\eta} \|w - w_0\|^2$$

le minimum de cette approximation est atteint pour $\nabla \tilde{\mathcal{L}}_{w_0}(w) = 0$:

$$\nabla \tilde{\mathcal{L}}_{w_0}(w) = \nabla \mathcal{L}(w_0) + \frac{1}{\eta} (w - w_0) = 0$$

soit lorsque

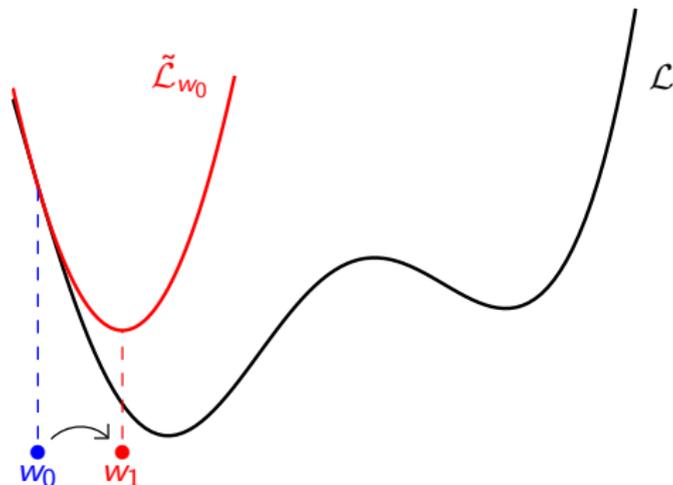
$$w = w_0 - \eta \nabla \mathcal{L}(w_0)$$

Descente de gradient

La descente de gradient consiste à “déplacer” la variable à optimiser (ici w) dans le sens contraire au gradient de manière à diminuer la fonction de coût.

Cet algorithme converge vers un minimum local de la fonction. Le choix de la vitesse d'apprentissage η et de la valeur de départ w_0 sont importants. Ici, on a

$$w_1 = w_0 - \eta \nabla \mathcal{L}(w_0)$$

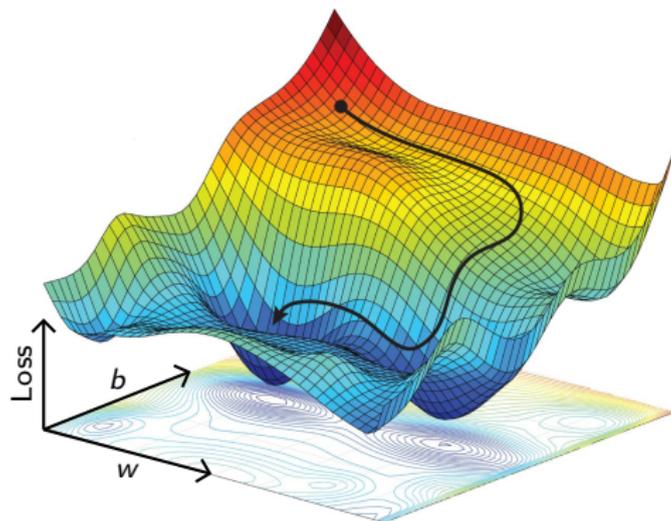


Descente de gradient

On en déduit une règle de mise à jour des paramètres du modèle, qui s'applique de manière itérative :

$$w^{(t+1)} = w^{(t)} - \eta \nabla \mathcal{L}(w^{(t)})$$

Cet algorithme converge en général vers un minimum local de la fonction.



Dérivée d'une composition de fonction

$$x \xrightarrow{f} f(x) = y \xrightarrow{g} g(y) = g(f(x)) = z$$

Dérivée d'une composition de fonction

$$x \xrightarrow{f} f(x) = y \xrightarrow{g} g(y) = g(f(x)) = z$$

La dérivée d'une composition de fonctions est :

$$(g \circ f)' = (g' \circ f)f'$$

ou autrement

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x}$$

Dérivée d'une composition de fonction

$$x \xrightarrow{f} f(x) = y \xrightarrow{g} g(y) = g(f(x)) = z$$

La dérivée d'une composition de fonctions est :

$$(g \circ f)' = (g' \circ f)f'$$

ou autrement

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x}$$

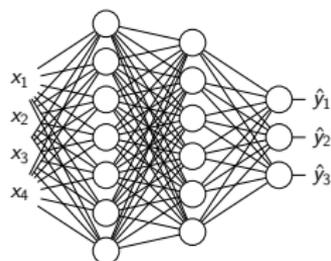
Par exemple, la fonction $h: x \mapsto (x^3 + 2)^2$ est la composition de 2 fonctions :

$$h: x \xrightarrow{f} x^3 + 2 \xrightarrow{g} (x^3 + 2)^2$$

Les dérivées de f et g étant $f'(t) = 3t^2$ et $g'(u) = 2u$, on a

$$h'(x) = \underbrace{2(x^3 + 2)}_{g'(f(x))} \underbrace{3x^2}_{f'(x)} = 6x^2(x^3 + 2)$$

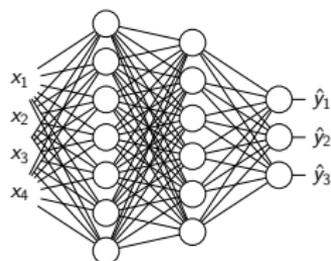
Rétro-propagation du gradient (*Backpropagation*)



Un réseau de neurones est la composition d'une succession d'opérations :

$$x \xrightarrow{f_1(\cdot; w_1)} a^{(1)} \xrightarrow{\sigma} x^{(1)} \xrightarrow{f_2(\cdot; w_2)} a^{(2)} \xrightarrow{\sigma} x^{(2)} \xrightarrow{f_3(\cdot; w_3)} a^{(3)} \xrightarrow{\sigma} x^{(3)} = \hat{y} \xrightarrow{\mathcal{L}} \mathcal{L}(\hat{y}, y)$$

Rétro-propagation du gradient (*Backpropagation*)



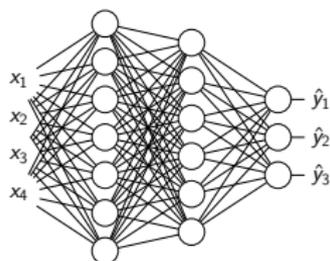
Un réseau de neurones est la composition d'une succession d'opérations :

$$x \xrightarrow{f_1(\cdot; w_1)} a^{(1)} \xrightarrow{\sigma} x^{(1)} \xrightarrow{f_2(\cdot; w_2)} a^{(2)} \xrightarrow{\sigma} x^{(2)} \xrightarrow{f_3(\cdot; w_3)} a^{(3)} \xrightarrow{\sigma} x^{(3)} = \hat{y} \xrightarrow{\mathcal{L}} \mathcal{L}(\hat{y}, y)$$

On souhaite optimiser les paramètres du modèle w_1, w_2, w_3 , en minimisant la fonction de coût \mathcal{L} , à l'aide de la descente de gradient. On met à jour chaque paramètre w_i de la façon suivante

$$w_i^{(t+1)} = w_i^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial w_i^{(t)}}$$

Rétro-propagation du gradient (*Backpropagation*)



Un réseau de neurones est la composition d'une succession d'opérations :

$$x \xrightarrow{f_1(\cdot; w_1)} a^{(1)} \xrightarrow{\sigma} x^{(1)} \xrightarrow{f_2(\cdot; w_2)} a^{(2)} \xrightarrow{\sigma} x^{(2)} \xrightarrow{f_3(\cdot; w_3)} a^{(3)} \xrightarrow{\sigma} x^{(3)} = \hat{y} \xrightarrow{\mathcal{L}} \mathcal{L}(\hat{y}, y)$$

On souhaite optimiser les paramètres du modèle w_1, w_2, w_3 , en minimisant la fonction de coût \mathcal{L} , à l'aide de la descente de gradient. On met à jour chaque paramètre w_i de la façon suivante

$$w_i^{(t+1)} = w_i^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial w_i^{(t)}}$$

On a donc

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_3} &= \frac{\partial \mathcal{L}}{\partial x_4} \frac{\partial x_4}{\partial a_3} \frac{\partial a_3}{\partial w_3} \\ \frac{\partial \mathcal{L}}{\partial w_2} &= \frac{\partial \mathcal{L}}{\partial x_4} \frac{\partial x_4}{\partial a_3} \frac{\partial a_3}{\partial x_3} \frac{\partial x_3}{\partial a_2} \frac{\partial a_2}{\partial w_2} \end{aligned}$$

Plan

Introduction

Applications de l'IA

Données

Classification

Réseaux de neurones artificiels

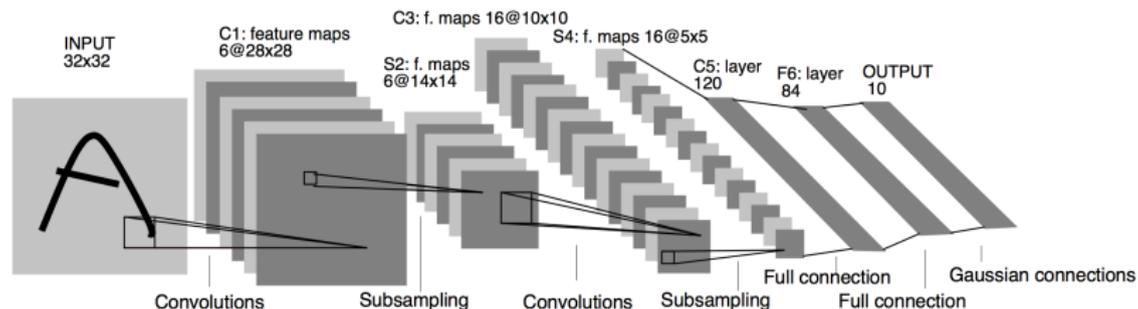
Apprentissage profond

Conclusion

Apprentissage profond

L'apprentissage profond (*deep learning*) est une classe de réseaux de neurones qui se caractérisent par un nombre important de couches de neurones, donc un grand nombre de paramètres à optimiser. Ils sont entraînés sur de grandes bases de données.

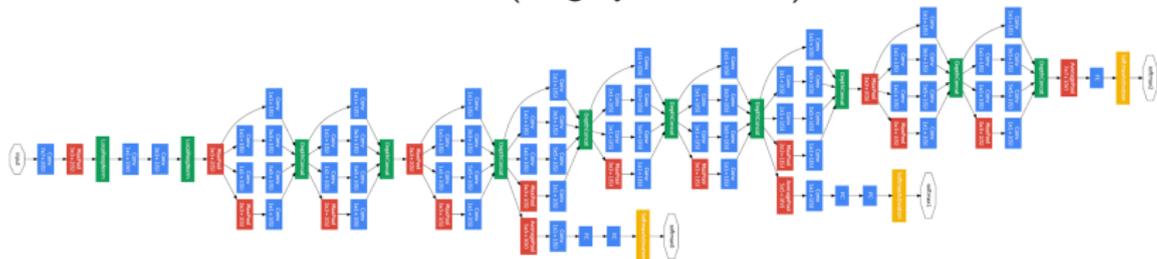
Le réseau LeNet5 (LeCun et al., 1998) a été l'un des premiers réseaux convolutifs utilisés pour lire automatiquement les chèques à la fin des années 1990.



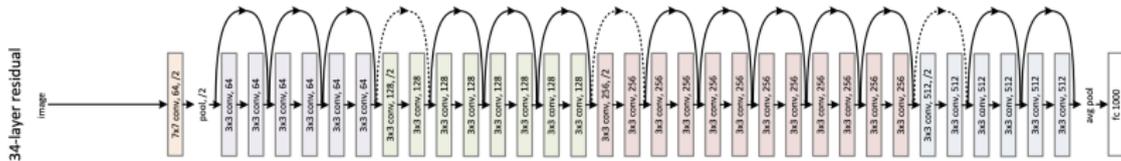
Des réseaux très profonds

Le réseau GooLeNet et le réseau ResNet utilisés pour la classification d'images ont respectivement 60 million et 21 million de paramètres.

GooLeNet (Szegedy et al., 2015)



ResNet (He et al., 2016)



Ils peuvent être entraînés avec la rétro-propagation du gradient sur la base ImageNet (Deng et al., 2009) qui contient un million d'images.

Plan

Introduction

Applications de l'IA

Données

Classification

Réseaux de neurones artificiels

Apprentissage profond

Conclusion

Conclusion

L'apprentissage automatique (*machine learning*) optimise des modèles à partir de données d'apprentissage pour effectuer une tâche spécifique.

Un réseau de neurones est un ensemble d'opérations matricielles organisées sous la forme d'un graphe de calcul.

Les réseaux de neurones sont optimisés grâce à la descente de gradient et à la rétro-propagation du gradient.

Bibliographie I

- Deng, Jia et al. (2009). "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- Fisher, Ronald A (1936). "The use of multiple measurements in taxonomic problems". In: *Annals of eugenics* 7.2, pp. 179–188.
- He, K. et al. (2016). "Deep residual learning for image recognition". In: *CVPR*, pp. 770–778.
- LeCun, Y. et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.
- Szegedy, C. et al. (2015). "Going deeper with convolutions". In: *CVPR*, pp. 1–9.
- Wikipedia (2023a). *Apprentissage automatique*.
https://fr.wikipedia.org/wiki/Apprentissage_automatique. (Consulté le 2 mars 2023).
- (2023b). *Apprentissage par renforcement*.
https://fr.wikipedia.org/wiki/Apprentissage_par_renforcement. (Consulté le 2 mars 2023).
- (2023c). *Intelligence artificielle*.
https://fr.wikipedia.org/wiki/Intelligence_artificielle. (Consulté le 2 mars 2023).
- (2023d). *Reconnaissance automatique de la parole*.
https://fr.wikipedia.org/wiki/Reconnaissance_automatique_de_la_parole. (Consulté le 2 mars 2023).