Les algorithmes d'optimisation : une modalité de l'intelligence ?

J. Bolte

Toulouse School of Economics (TSE)

&

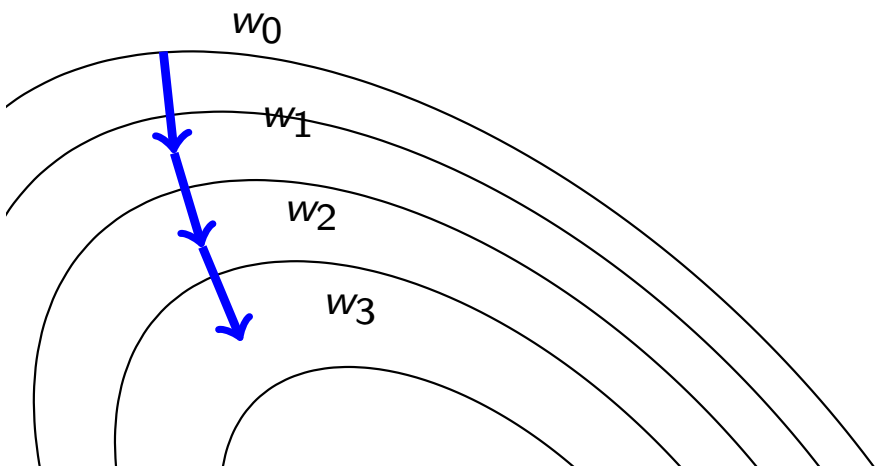
Artificial and Natural Intelligence Toulouse Institute (ANITI)

CNRS, Le 3 octobre 2024

L'algorithme central de l'optimisation de grande taille : méthode du gradient

Un principe d'amélioration local : $w_{k+1} = w_t - \gamma_k \nabla f(w_k)$ (avec $\gamma_k > 0$)

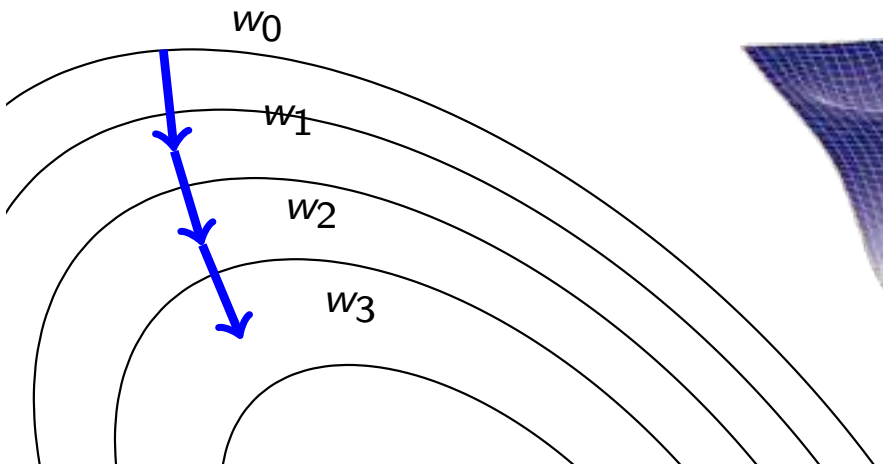
ou encore : $w'(t) = -\nabla f(w(t))$

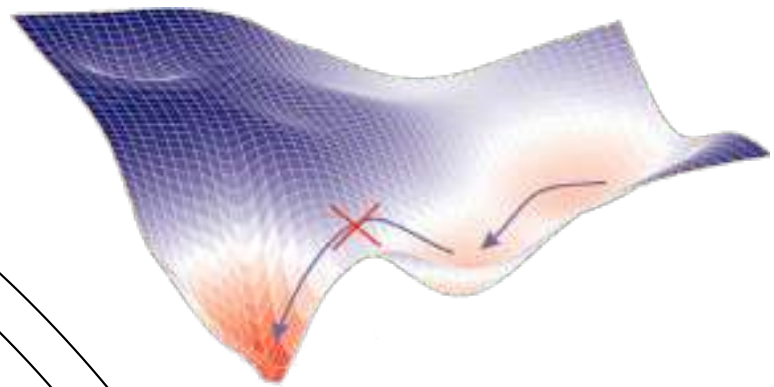


L'algorithme central de l'optimisation de grande taille : méthode du gradient

Un principe d'amélioration local : $w_{k+1} = w_t - \gamma_k \nabla f(w_k)$ (avec $\gamma_k > 0$)

ou encore : $w'(t) = -\nabla f(w(t))$

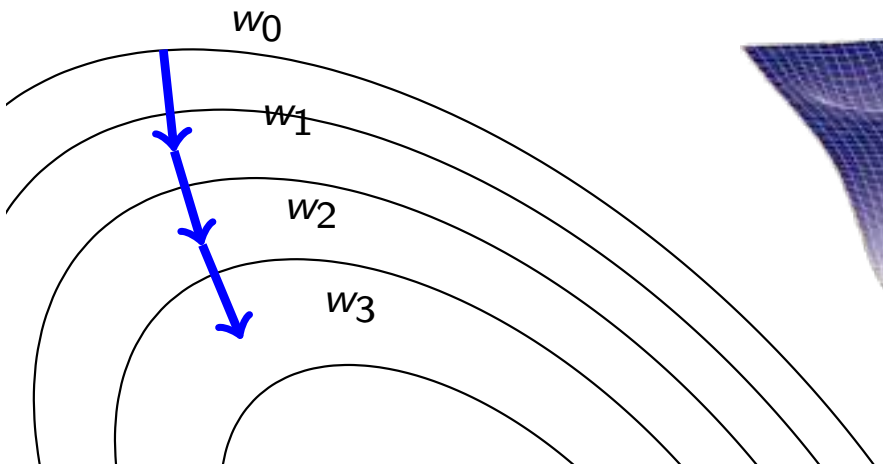


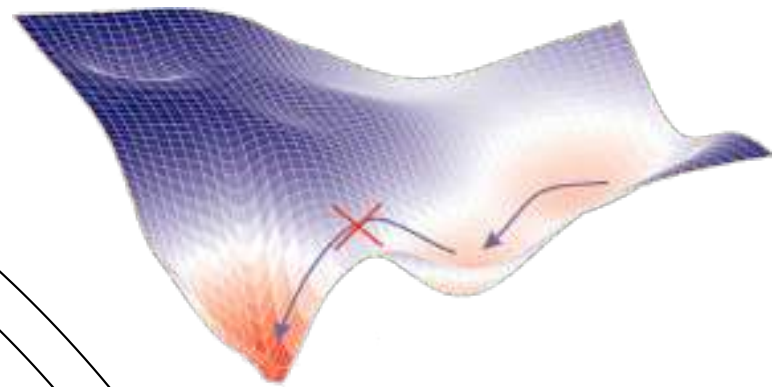


L'algorithme central de l'optimisation de grande taille : méthode du gradient

Un principe d'amélioration local : $w_{k+1} = w_t - \gamma_k \nabla f(w_k)$ (avec $\gamma_k > 0$)

ou encore : $w'(t) = -\nabla f(w(t))$





Centralité de l'algorithme du gradient en grande taille :

- Coût avantageux : une addition et le calcul de gradient
- accélérable
- gouverne une foule de phénomènes géométriques ou naturels : équation de la chaleur, boule pesante ou ondes amortie, évolution cinétique gaz, jeux de potentiels... → nombreuses intuitions

Deux faits majeurs

- **Le gradient est une notion métrique** Soit $F : M \to \mathbb{R}$.

  Si M est un espace muni d'une distance et d'espaces tangents, il est possible de définir des gradients et une *dynamique de gradient*

$$W'(t) = -\mathrm{grad}\, F(W(t)), t \geq 0$$

"Courbes de gradient = courbes de plus grande pente = famille d'orthogonales aux surfaces de niveau"

Deux faits majeurs

- **Le gradient est une notion métrique** Soit $F : M \to \mathbb{R}$.

  Si M est un espace muni d'une distance et d'espaces tangents, il est possible de définir des gradients et une *dynamique de gradient*

  $$W'(t) = -\operatorname{grad} F(W(t)), t \geq 0$$

  "Courbes de gradient = courbes de plus grande pente = famille d'orthogonales aux surfaces de niveau"

- **L'optimisation est un jeu de formulations** :
  On construit $F : M \to \mathbb{R}$ de sorte que

  $$\min_{\mathbb{R}^D} f = \min_{M} F$$

  avec si possible une bonne correspondance $\operatorname{argmin} f$ et $\operatorname{argmin} F$.

  Puis on étudie

  $$W'(t) = -\operatorname{grad} F(W(t)), t \geq 0$$

  ou des formes algorithmiques.

Exemples [1]

- **Exemple 1 : Courbes de gradients**

$$\boxed{w'(t) = -\nabla f(w(t))}$$

1. Merci Edouard (Pauwels)

Exemples [1]

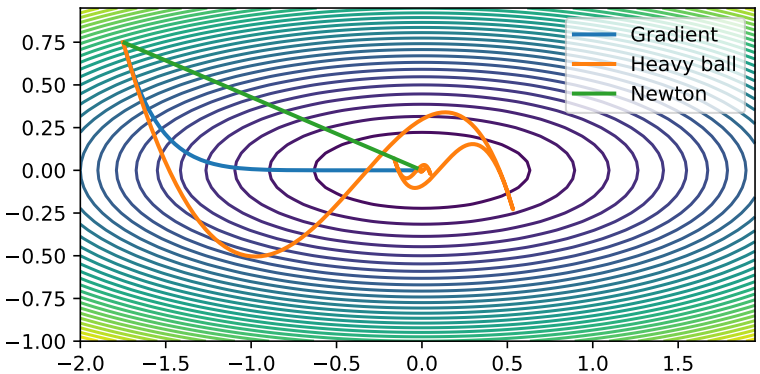
- **Exemple 1 : Courbes de gradients**

$$w'(t) = -\nabla f(w(t))$$

- **Exemple 2 : Méthode de Newton continue**

$M = \mathbb{R}^D$,
$F = f$
$d(w, w+h) \simeq \|\sqrt{\nabla^2 f(w)}(h)\|$.

$$w'(t) = -\nabla^2 f(w(t))^{-1} \nabla f(w(t))$$



1. Merci Edouard (Pauwels)

Exemples [1]

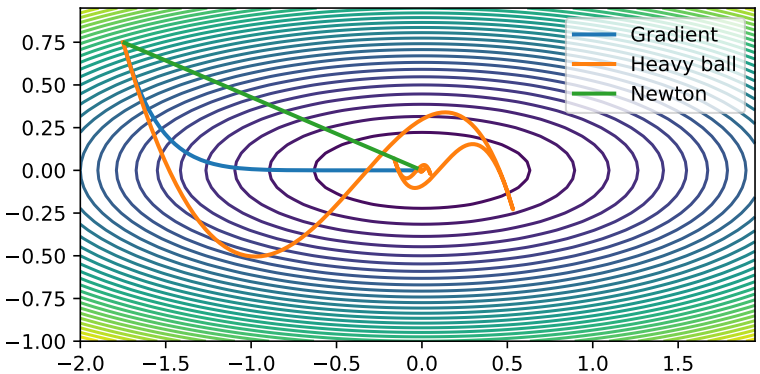
- **Exemple 1 : Courbes de gradients**

$$w'(t) = -\nabla f(w(t))$$

- **Exemple 2 : Méthode de Newton continue**

$M = \mathbb{R}^D$,
$F = f$
$d(w, w+h) \simeq \|\sqrt{\nabla^2 f(w)}(h)\|$.

$$w'(t) = -\nabla^2 f(w(t))^{-1} \nabla f(w(t))$$



- **Exemple 3 : Boule pesante avec frottements**

$M = \mathbb{R}^D \times \mathbb{R}^D$,
$F(w,u) = 1/2\|u\|^2 + f(w) + c\langle u, \nabla f(w)\rangle$
et une métrique bien choisie

$$w''(t) + \alpha w'(t) + \nabla f(w(t)) = 0.$$

1. Merci Edouard (Pauwels)

Le jeu des formulations et des changement de métriques

- Plongement dans un espace de probabilités : $\mathbb{R}^D \hookrightarrow \{\delta_w\}_{w\in\mathbb{R}^D} \subset \mathrm{Proba}(\mathbb{R}^D)$.

$$\min_{w\in\mathbb{R}^D} f = \min_{P\in\underbrace{\mathrm{Proba}(\mathbb{R}^D)}_{:=M}} \underbrace{\int f\,dP}_{:=F(P)}$$

Points de $\mathbb{R}^D$ sont vus comme des positions possibles pour une particule. Au lieu d'affecter une position à la particule on y affecte une probabilité de présence et on évalue le coût : $F(P) = \int_{\mathbb{R}^D} F dP$

Le jeu des formulations et des changement de métriques

- Plongement dans un espace de probabilités : $\mathbb{R}^D \hookrightarrow \{\delta_w\}_{w\in\mathbb{R}^D} \subset \mathrm{Proba}(\mathbb{R}^D)$.

$$\min_{w\in\mathbb{R}^D} f = \min_{P\in \underbrace{\mathrm{Proba}(\mathbb{R}^D)}_{:=M}} \underbrace{\int f\,dP}_{:=F(P)}$$

  Points de $\mathbb{R}^D$ sont vus comme des positions possibles pour une particule. Au lieu d'affecter une position à la particule on y affecte une probabilité de présence et on évalue le coût : $F(P) = \int_{\mathbb{R}^D} F dP$

- Forcer l'"ubiquité" des particules :

$$F_\epsilon(\rho) := \int_{\mathbb{R}^D} f(w)\rho(w)dw + \epsilon \int_{\mathbb{R}^D} \rho(w)\log\rho(w)dw$$

- Avec la *métrique du transport optimal (cf G. Peyré)*, $\rho'(t) = -\mathrm{grad}\, F_\epsilon(\rho(t))$ s'écrit :

$$\frac{\partial}{\partial t}\rho = \mathrm{div}(\rho\nabla f) + \epsilon\Delta\rho, \qquad \text{(Equation de Fokker-Planck)} \qquad \text{(b)}$$

  qui est la loi d'une variable aléatoire X_t satisfaisant :

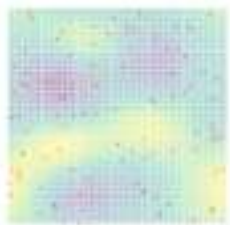
$$dX_t = -\nabla f(X_t)dt + \epsilon dB_t$$

  où dB_t est un Brownien, jouant le rôle d'"un terme exploratoire aléatoire"

Illustrations [2]

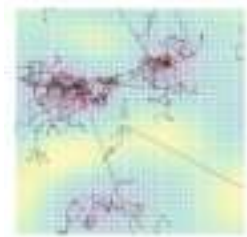
Si $\epsilon = \epsilon_t$, on obtient le recuit simulé : propriétés de convergence globale en *petite dimension.*

Un “essaim" de particules (indépendantes) $dX_t = -\nabla f(X_t)dt + \epsilon_t dB_t$



L'essaim à l'instant $t \simeq 0$

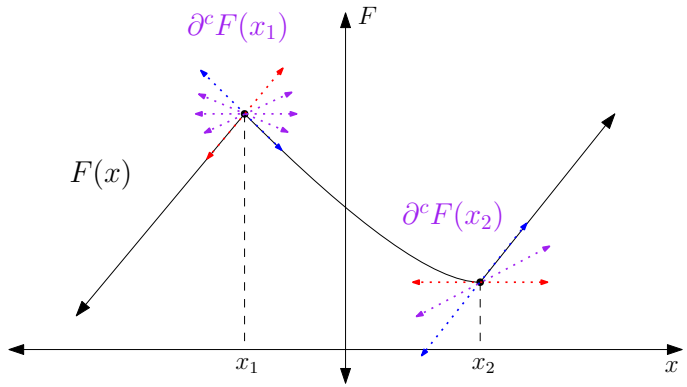
Instant $t \simeq 50$



Instant $t \simeq 100$.

2. Merci Lenaïc (Chizat)

Cas non-lisse ou non-différentiable [3]

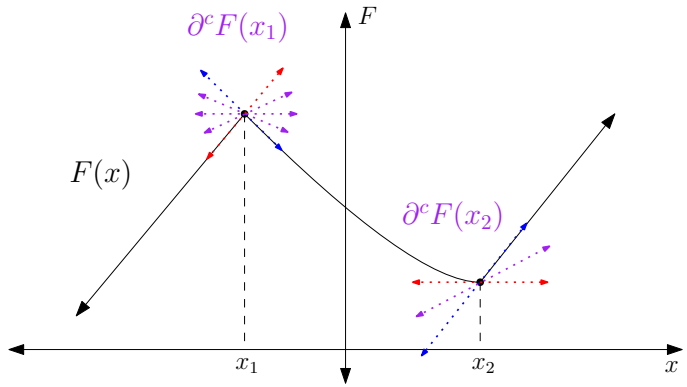


Soit $F : \mathbb{R}^p \to \mathbb{R}^q$ localement Lipschitz. Le **sous-gradient/jacobien de Clarke** est

$$\partial^c F(x) = \operatorname{conv}\left\{ \lim_{k \to +\infty} \operatorname{Jac} F(x_k) : x_k \in \operatorname{diff}_F, x_k \underset{k \to +\infty}{\to} x \right\}$$

3. Merci Tam (Le)

Cas non-lisse ou non-différentiable [3]



Soit $F : \mathbb{R}^p \to \mathbb{R}^q$ localement Lipschitz. Le **sous-gradient/jacobien de Clarke** est

$$\partial^c F(x) = \operatorname{conv}\left\{ \lim_{k \to +\infty} \operatorname{Jac} F(x_k) : x_k \in \operatorname{diff}_F, x_k \underset{k \to +\infty}{\to} x \right\}$$

La **méthode de sous gradient** s'écrit

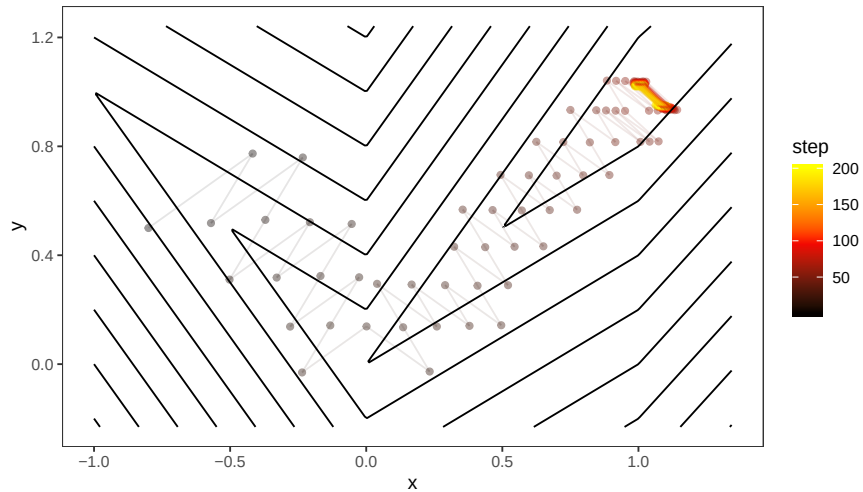
$$w_{k+1} \in w_k - \gamma_k \partial f(w_k),$$

avec $\gamma_k \to 0$.

3. Merci Tam (Le)

Illustration : méthode du sous-gradient

$$w_{k+1} \in w_k - \gamma_k \partial f(w_k) \text{ avec } \gamma_k \to 0.$$



Suite de sous-gradients.

La couleur reflète le nombre d'itérations.

La suite oscille perpétuellement ; on peut montrer un phénomène de compensation des oscillations.

La convergence est une question ouverte.

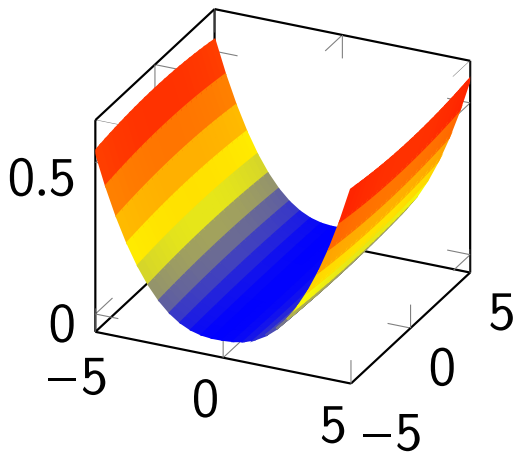
Clés de convergence vers un “équilibre" : courbure et conditionnement

Géométrie du graphe : convexité, courbure, mais aussi "amenabilité au cas linéaire"

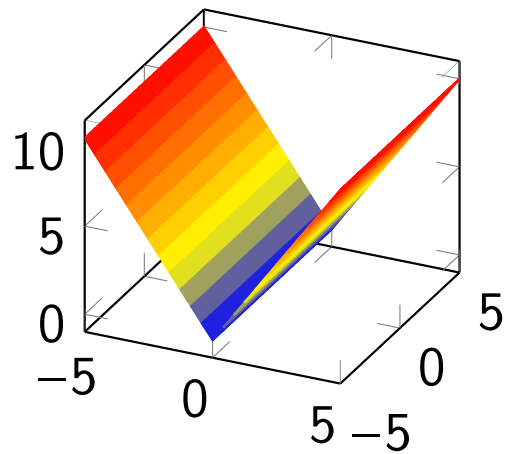
Pour toute valeur $f^* = f(x_0)$ la fonction est *affutable*

$$\|\nabla(f - f^*)^\theta(x)\| \geq 1 \text{ au voisinage de } x_0$$

$\swarrow\ f - f^*$ et $(f - f^*)^\theta\ \searrow$



Reparamétrisation de $f - f^*$ via $s \to s^\theta$ affute la fonction !



Fonctions fortement convexes, Inégalités dans les cas lisses ou non lisse semi-algébriques, Inégalité Log-Sobolev, Inégalité Gagliardo-Nirenberg etc ...

Foule de résultats de convergence et de complexité

Les fondements de la raison au 17ème siècle ?

Trois grands penseurs de l'"Aube des Lumières" :

- René Descartes, Le Discours de la Méthode, 1637,

  «*Ces longues chaînes de raisons, toutes simples et faciles, dont les géomètres ont coutume de se servir pour parvenir à leurs plus difficiles démonstrations, m'avaient donné occasion de m'imaginer que toutes les choses qui peuvent tomber sous la connaissance des hommes s'entresuivent en même façon*»

- Thomas Hobbes, Leviathan, 1651,

  «*Car la raison (...) n'est rien d'autre que le fait de calculer, c'est-à-dire additionner et soustraire, les consécutions des dénominations générales admises pour marquer et signifier nos pensées*»

- Gottfried Wilhelm Leibniz, « Nova methodus pro maximis et minimis » in Acta Eruditorum, 1684 Characteristica Universalis,

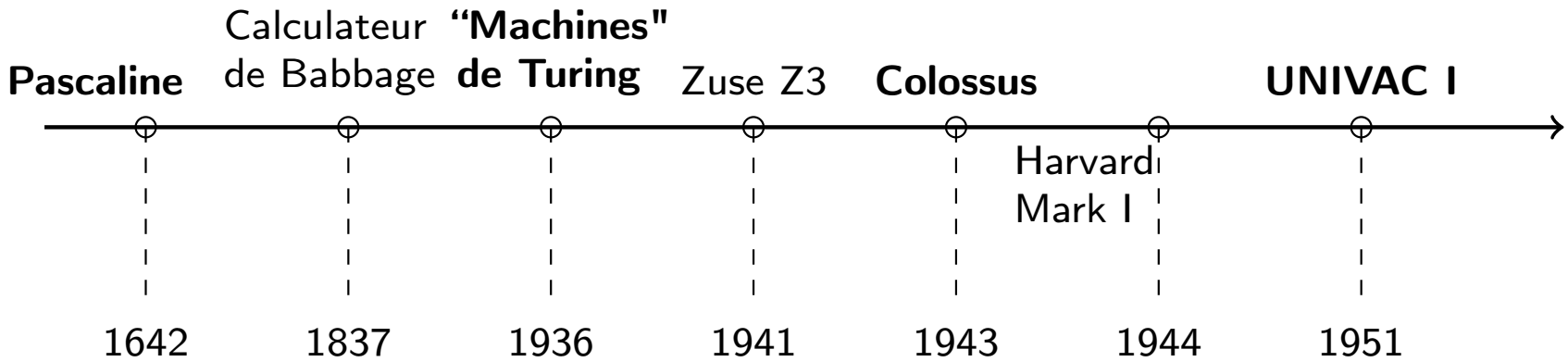
  « *Alors, il ne sera plus besoin entre deux philosophes de discussions plus longues qu'entre deux mathématiciens, puisqu'il suffira qu'ils saisissent leur plume, qu'ils s'asseyent à leur table de calcul (en faisant appel, s'ils le souhaitent, à un ami) et qu'ils se disent l'un à l'autre : « Calculons ! »* »

Le Calculus Ratiocinator de Leibniz

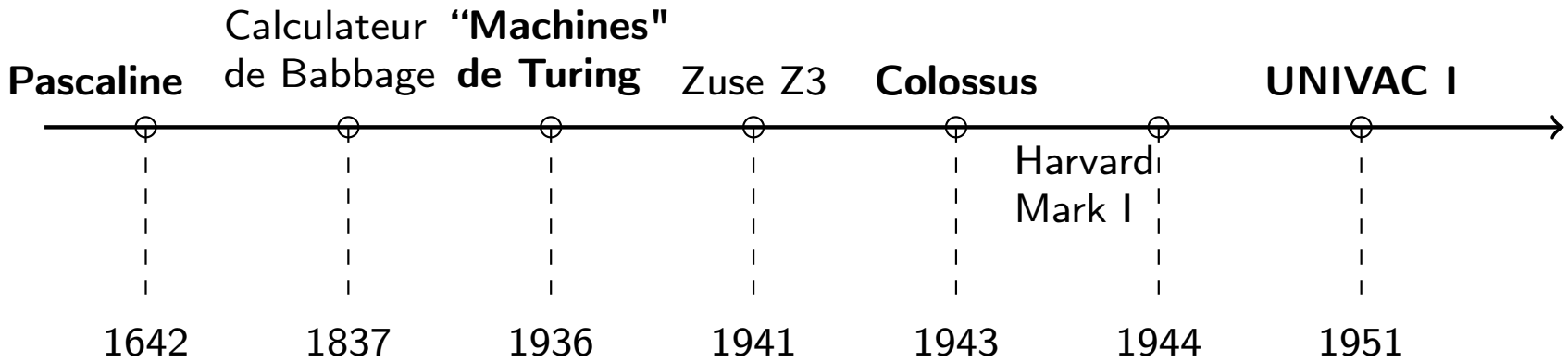
Leibniz, De arte combinatoria, 1666

Idée du “Calculus Ratiocinator" **une méthode, un algorithme,** ou une machine, **qui permettrait de démêler le vrai du faux dans toute discussion** dont les termes seraient exprimés dans une langue philosophique universelle, que Leibniz appelait la Caractéristique Universelle.

Le grand projet d'automatisation du calcul



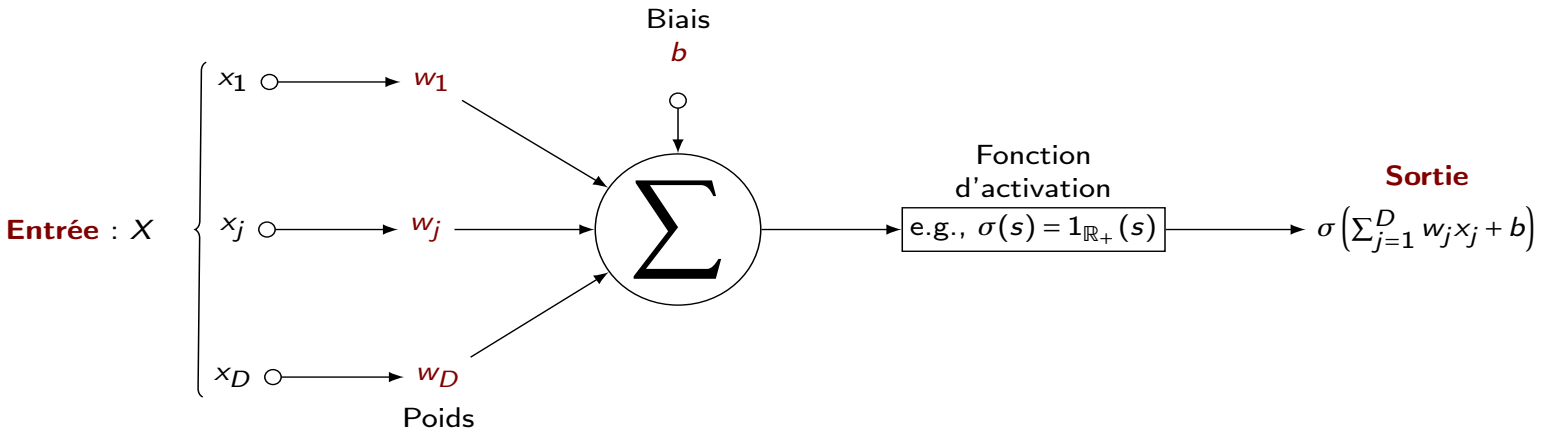
Le grand projet d'automatisation du calcul



Les calculs élémentaires étant augmentés d'algorithmes d'optimisation : division, $x^2 = a$, fonctions élémentaires, et toujours plus de "solveurs"

Avènement des neuro-sciences : le neurone de McCulloch-Pitts

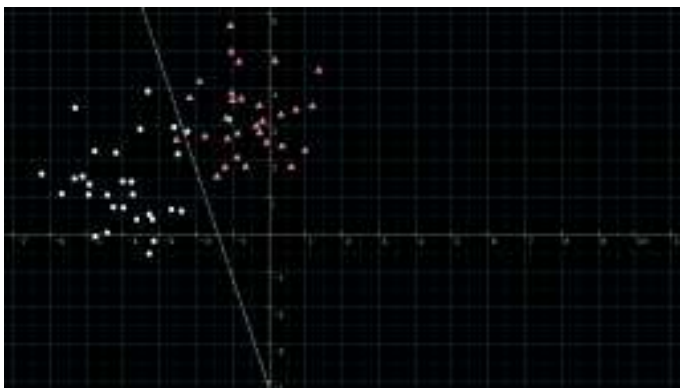
Entrée : $X \in \mathbb{R}^D$
Sortie : 0 ou 1 $\sigma(s) = 0$ si $s < 0$, $\sigma(s) = 1$ sinon



Objectif : classer des données de deux types possibles (X_j, y_j) où $y_j = +1$ ou $y_j = -1$

Perfectionnement de Rosenblatt, psychométricien à Cornell

Avec $\sigma(s) = \frac{\exp s}{1+\exp s}$ et un algorithme d'entrainement sur un IBM-704



Rebaptisé perceptron par Rosenblatt et premiers *entrainements* informatiques

F. Rosenblatt affirme dans le New-York Times :

"*A l'avenir, les perceptrons pourront reconnaître des personnes et les appeler par leur nom. Ils pourront écrire du texte à la dictée ou le traduire instantanément dans une autre langue.*"

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

1958 New York Times...

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

Des formules de décisions pour le perceptron

$S=$ nombre d'échantillons

$$\min_{W=(w,b)\in\mathbb{R}^{D+1}} f(w,b)$$

avec

$$f(w,b) = \sum_{i=1}^{S} \max\left\{0, 1 - y_i(w^T X_i - b)\right\}$$

Des formules de décisions pour le perceptron

$S=$ nombre d'échantillons

$$\min_{W=(w,b)\in\mathbb{R}^{D+1}} f(w,b)$$

avec

$$f(w,b)=\sum_{i=1}^{S}\max\left\{0,1-y_i(w^T X_i-b)\right\}$$

Formellement nous pouvons écrire que l'unique choix optimal est donné par le développement infini

$$W^*=W_0-\sum_{k=1}^{\infty}\frac{1}{k}g_k$$

avec

$$g_k\in\partial f(W_k)=\left\{\sum_{i\in I}\lambda_i(-y_iX_i,y_i)\in\mathbb{R}^{D+1}:\mu+\sum_{i\in I(W_k)}\lambda_i=1 \text{ et } \mu\geq 0,\lambda_i\geq 0,\forall i\in I(W_k)\right\}$$

où

$$I(W_k)=\left\{j:1-y_j(w_k^T X_j-b)=\max_i\left\{0,1-y_i(w_k^T X_i-b)\right\}\right\}$$

$$\mu\max_i\left\{0,1-y_i(w_k^T X_i-b)\right\}=0$$

Des formules de décisions pour le perceptron

$S=$ nombre d'échantillons

$$\min_{W=(w,b)\in\mathbb{R}^{D+1}} f(w,b)$$

avec

$$f(w,b)=\sum_{i=1}^{S}\max\left\{0,1-y_i(w^T X_i-b)\right\}$$

Formellement nous pouvons écrire que l'unique choix optimal est donné par le développement infini

$$W^*=W_0-\sum_{k=1}^{\infty}\frac{1}{k}g_k$$

avec

$$g_k\in\partial f(W_k)=\left\{\sum_{i\in I}\lambda_i(-y_iX_i,y_i)\in\mathbb{R}^{D+1}:\mu+\sum_{i\in I(W_k)}\lambda_i=1 \text{ et } \mu\geq 0,\lambda_i\geq 0,\forall i\in I(W_k)\right\}$$

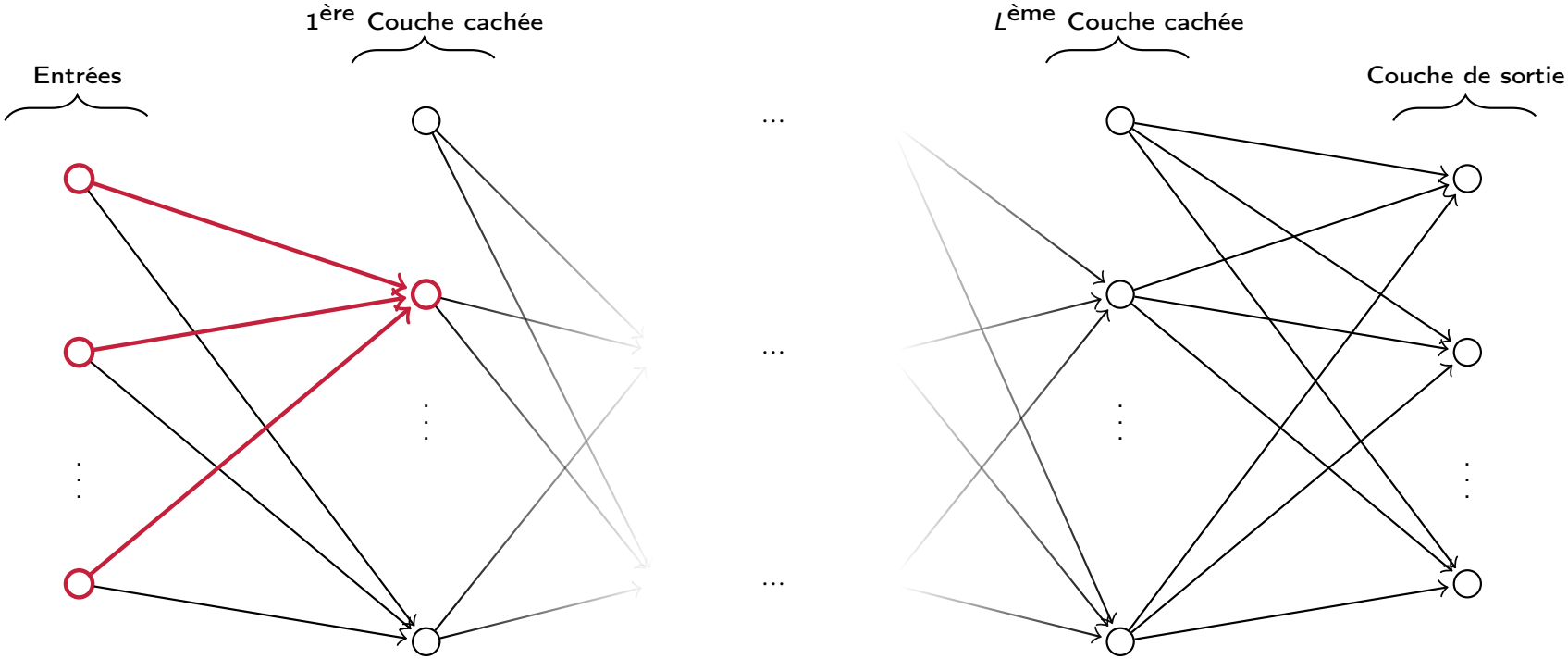
où

$$I(W_k)=\left\{j:1-y_j(w_k^T X_j-b)=\max_i\left\{0,1-y_i(w_k^T X_i-b)\right\}\right\}$$

$$\mu\max_i\left\{0,1-y_i(w_k^T X_i-b)\right\}=0$$

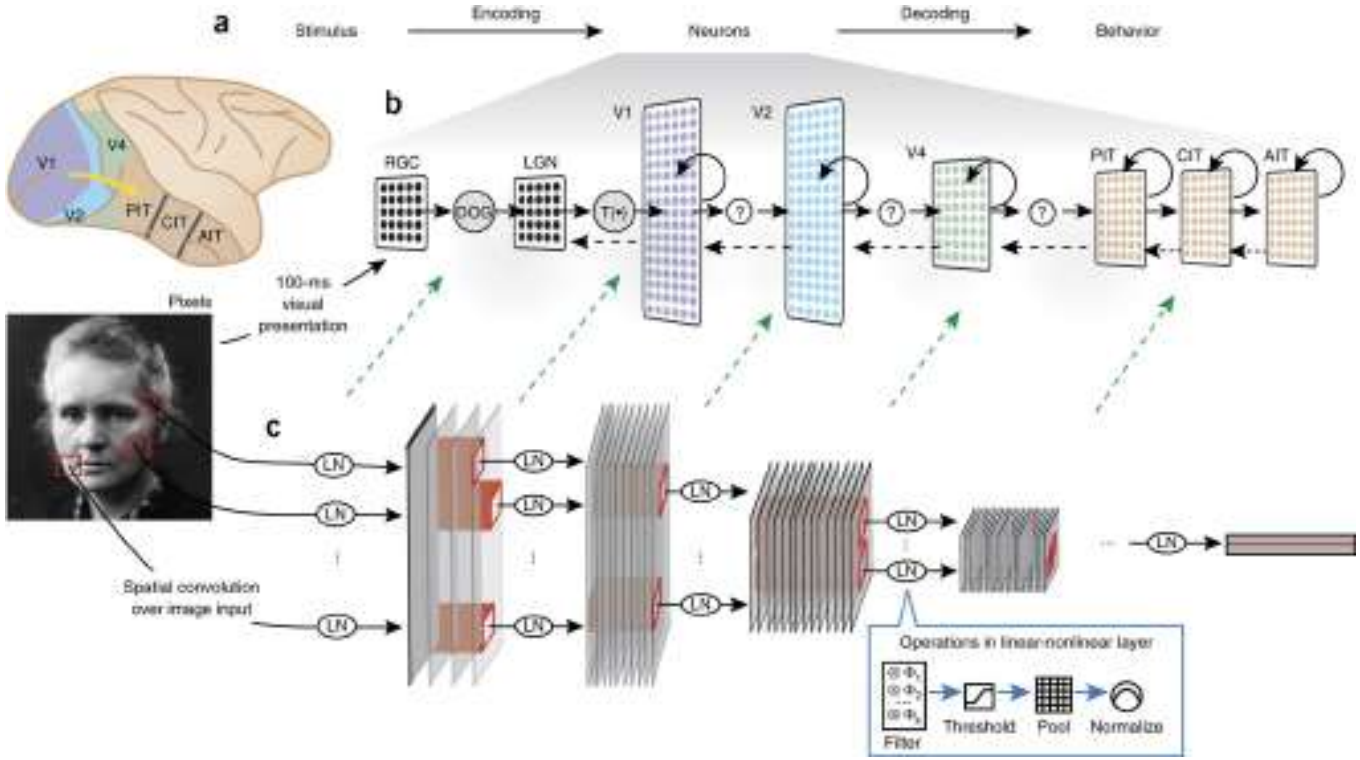
"Asseyons nous à notre table de calcul et (...) calculons !"

Perceptron multi-couches, premiers réseaux de neurones...



Un empilement de perceptrons : un “réseau de neurones".

Eléments empiriques : processus visuels corticaux



Le processus visuel ventral : un réseau de neurone prograde

Tiré de *Using goal-driven deep learning models to understand sensory cortex D. L K Yamins & J. J DiCarlo, Nature Neuroscience, 2016*

La dynamique modèle de l'apprentissage automatique

- Apprendre des données labellisées (X_i, Y_i) via un prédicteur $F_w(X_i) \simeq Y_i$. où w doit être réglé :

$$\text{Minimiser } f(w) = \sum_{i=1}^{S} f_i(w) = \sum_{i=1}^{S} \|F_w(X_i) - Y_i\|^2 \text{ pour } w \in \mathbb{R}^D$$

- $f = \sum_{i=1}^{S} f_i$ est optimisé par une méthode de gradient

$$w_{k+1} = w_k - \gamma \nabla f(w_k) \quad k \leq E$$

D dimension, S nombre de tâches, E nombre de pas

La dynamique modèle de l'apprentissage automatique

- Apprendre des données labellisées (X_i, Y_i) via un prédicteur $F_w(X_i) \simeq Y_i$. où w doit être réglé :

$$\text{Minimiser } f(w) = \sum_{i=1}^{S} f_i(w) = \sum_{i=1}^{S} \|F_w(X_i) - Y_i\|^2 \text{ pour } w \in \mathbb{R}^D$$

- $f = \sum_{i=1}^{S} f_i$ est optimisé par une méthode de gradient

$$w_{k+1} = w_k - \gamma \nabla f(w_k) \quad k \leq E$$

D dimension, S nombre de tâches, E nombre de pas

- Le nombre de dérivées partielles à évaluer $= D.S.E$.
  Par exemple pour ImageNet : $E = O(10^3)$, $S = O(10^6)$, $E := O(10^9)$:

$$DSE = O(10^{18}) \text{ dérivées partielles}$$

La dynamique modèle de l'apprentissage automatique

- Apprendre des données labellisées (X_i, Y_i) via un prédicteur $F_w(X_i) \simeq Y_i$. où w doit être réglé :

$$\text{Minimiser } f(w) = \sum_{i=1}^{S} f_i(w) = \sum_{i=1}^{S} \|F_w(X_i) - Y_i\|^2 \text{ pour } w \in \mathbb{R}^D$$

- $f = \sum_{i=1}^{S} f_i$ est optimisé par une méthode de gradient

$$w_{k+1} = w_k - \gamma \nabla f(w_k) \quad k \leq E$$

D dimension, S nombre de tâches, E nombre de pas

- Le nombre de dérivées partielles à évaluer $= D.S.E$.
  Par exemple pour ImageNet : $E = O(10^3)$, $S = O(10^6)$, $E := O(10^9)$:

$$DSE = O(10^{18}) \text{ dérivées partielles}$$

- $10^{18} \simeq$ est l'age de l'univers en seconde ...

La dynamique modèle de l'apprentissage automatique

- Apprendre des données labellisées (X_i, Y_i) via un prédicteur $F_w(X_i) \simeq Y_i$. où w doit être réglé :

$$\text{Minimiser } f(w) = \sum_{i=1}^{S} f_i(w) = \sum_{i=1}^{S} \|F_w(X_i) - Y_i\|^2 \text{ pour } w \in \mathbb{R}^D$$

- $f = \sum_{i=1}^{S} f_i$ est optimisé par une méthode de gradient

$$w_{k+1} = w_k - \gamma \nabla f(w_k) \quad k \leq E$$

D dimension, S nombre de tâches, E nombre de pas

- Le nombre de dérivées partielles à évaluer $= D.S.E$.
  Par exemple pour ImageNet : $E = O(10^3)$, $S = O(10^6)$, $E := O(10^9)$:

$$DSE = O(10^{18}) \text{ dérivées partielles}$$

- $10^{18} \simeq$ est l'age de l'univers en seconde ...
- **Historiquement : un bloquage considérable à la fois théorique, technologique et .... "empirique"**

Un algorithme rapide : la rétropropagation

- L'algorithme de rétropropagation est un algorithme rapide "exact" pour calculer le gradient de (Linnainmaa, 70) redécouvert en 86 par le monde de l'IA...

  Il est basé sur la règle des dérivée composées
  A ce jour apprendre en IA est principalement accumuler des applications de la rétropropagation.

- Pour toute fonction polynomiale réelle $Q : \mathbb{R}^p \to \mathbb{R}^m$, on définit :

  $\text{coutcalcul}(Q) =$ nombre minimal de calculs arithmétiques $+, \times$ pour évaluer Q.

  On a $\text{coutcalcul}(\nabla f) \leq O(\text{dimension} \times \text{coutcalcul}(f))$ pour $f : \mathbb{R}^p \to \mathbb{R}$ polynôme

Un algorithme rapide : la rétropropagation

- L'algorithme de rétropropagation est un algorithme rapide "exact" pour calculer le gradient de (Linnainmaa, 70) redécouvert en 86 par le monde de l'IA...

  Il est basé sur la règle des dérivée composées
  A ce jour apprendre en IA est principalement accumuler des applications de la rétropropagation.

- Pour toute fonction polynomiale réelle $Q : \mathbb{R}^p \to \mathbb{R}^m$, on définit :

  $$\text{coutcalcul}(Q) = \text{nombre minimal de calculs arithmétiques } +, \times \text{ pour évaluer } Q.$$

  On a $\text{coutcalcul}(\nabla f) \leq O(\text{dimension} \times \text{coutcalcul}(f))$ pour $f : \mathbb{R}^p \to \mathbb{R}$ polynôme

Théorème [Baur-Strassen — 1983] Il existe un algorithme qui permet d'obtenir

$$\text{coutcalcul}(f, \nabla f) \leq 5\, \text{coutcalcul}(f).$$

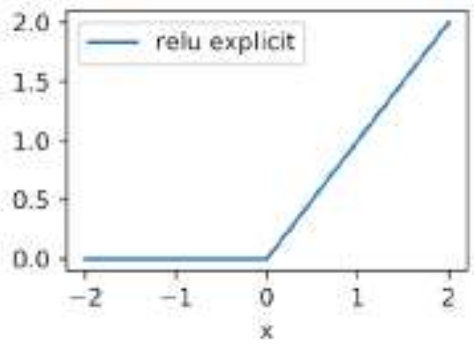
La rétropropagation, ou plus généralement la différentiation automatique, est une application de la chain rule et agit en réalité sur des programmes numériques

Les programmes modernes sont par nature non différentiables

La non-lissité des programmes numériques provient principalement de

- **Instructions conditionnelles** (if, then, else)
- **Solveurs dans les domaines appliqués** (IA, Physique, Robotique...)
- **Régularisation** (Statistiques ou problèmes inverses)

```
def myRelu(x):
    if x<=0:
        return 0
    else:
        return x
```



Les instructions conditionnelles produisent de la non-lissité

Exemples

- Fonctions de tri (ex. Classement)
- Contraintes unilatérales (ex. Robotique)
- Bang-bang et chocs (ex. Contrôle, EDP)
- Norme ℓ^1 (ex. parcimonie, régularisation statistique)

Le théorème du gradient et la convergence des réseaux de neurones

Cadre des structures o-minimales : les fonctions et ensembles sont définis par des fonctions élémentaires : $\times, +, \exp, \log$ ou des fonctions analytiques élémentaires restreintes à des compacts ...

Deux archétypes : les semi-algébriques et les linéaires par morceaux

Théorème (B.-Pauwels) Si f est Lipschitz et définissable dans une structure o-minimale :

$$\text{rétroprop}(f)(w) = \nabla f(w) \text{ pour presque tout } w \tag{1}$$

$$\text{coutcalcul}(f, \text{rétroprop } f) \leq 5\,\text{coutcalcul}(f) \tag{2}$$

En conséquence

Théorème (B.-Pauwels) L'entrainement de réseaux de neurones mène généralement à des points stationnaires.

Généralisations de résultats de Schechtman, Davis et al : “on atteint "génériquement" des minima locaux""

Comprendre la non-convexité et le la méthode du gradient en IA ?

Fait empirique majeur : très fréquemment les réseaux de neurones donnent lieu à une minimisation globale (ou quasi-globale) de la fonction de perte

- Quelle(s) structure(s) fonctionnelle(s) induisent les réseaux de neurones ?
- L'implémentation stochastique confère t elle une intelligence globale à la méthode ?

Comprendre la non-convexité et le la méthode du gradient en IA ?

Fait empirique majeur : très fréquemment les réseaux de neurones donnent lieu à une minimisation globale (ou quasi-globale) de la fonction de perte

- Quelle(s) structure(s) fonctionnelle(s) induisent les réseaux de neurones ?
- L'implémentation stochastique confère t elle une intelligence globale à la méthode ?

Dans le régime "surparamétré" (infinité de couches, couches infinies ...) divers résultats

- Convexité (une couche cachée) [Chizat et Bach, ...]
- Les valeurs des minima locaux se concentrent [Ben Arous et al., Pennington et al., ...]
- Ubiquité des minima globaux et bon conditionnement [S. Du et al, Belkin et al., Kawaguchi, ...]
- Mise en oeuvre très bruité "des méthodes de gradient" et "recuit" [Zou, Li, ...]

Comprendre la non-convexité et le la méthode du gradient en IA ?

Fait empirique majeur : très fréquemment les réseaux de neurones donnent lieu à une minimisation globale (ou quasi-globale) de la fonction de perte

- Quelle(s) structure(s) fonctionnelle(s) induisent les réseaux de neurones ?
- L'implémentation stochastique confère t elle une intelligence globale à la méthode ?

Dans le régime "surparamétré" (infinité de couches, couches infinies ...) divers résultats

- Convexité (une couche cachée) [Chizat et Bach, ...]
- Les valeurs des minima locaux se concentrent [Ben Arous et al., Pennington et al., ...]
- Ubiquité des minima globaux et bon conditionnement [S. Du et al, Belkin et al., Kawaguchi, ...]
- Mise en oeuvre très bruité "des méthodes de gradient" et "recuit" [Zou, Li, ...]

Loin des réalités empiriques ....

Conclusions

Automatisation du raisonnement

- "La raison mise en équation (?)" : Pascaline, Ratiocinator, Ordinateurs, Perceptron, Micro-ordinateurs, Réseaux de neurones, reconnaissance visuelle/vocale, chatGPT, le programme de Descartes/Hobbes/Leibniz continue d'avancer... jusqu'à où ?
- *Les critères et calculs sous-jacents sont orchestrés par l'optimisation* : évaluation de fonctions élémentaires, calcul de gradient non réguliers, méthodes de gradient (AdamW) et nombreux raffinements ...

Des objets d'études issus des neurosciences

- Les fonctions sous-jacentes sont des modèles produits par les neuro-sciences ; ils nous ont révélé la puissance des "prédicteurs compositionnels "
- Les fonctions pertes ne devraient pas être minimisables par des méthodes locales mais le sont.
  Quelles propriétés structurelles fondamentales des pertes se cachent derrière les réseaux de neurones ?
- Y a-t-il des algorithmes d'optimisation biologiques à découvrir ?
  Rétro-propagation et cerveau ?