MA1-Analyse numérique

J. Bahamou, E. Bretin et S. Thery

September 18, 2025

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Plan du cours

1. Introduction
2. Méthode des moindres carrés linéaires
3. Optimisation et méthode du gradient
4. Régression logistique
5. Introduction aux réseaux de neurones

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Plan du chapitre

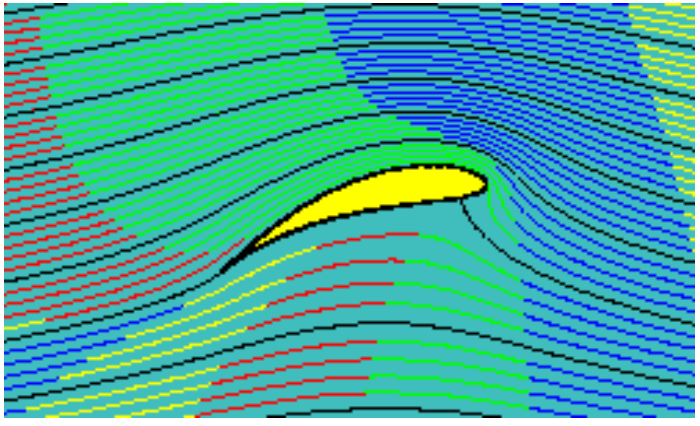
1. Introduction
   - Motivation
   - Représentation des réels et erreur numérique
   - Conditionnement, stabilité et complexité
2. Méthode des moindres carrés linéaires
3. Optimisation et méthode du gradient
4. Régression logistique
5. Introduction aux réseaux de neurones

Motivation

L'analyse numérique est une discipline des mathématiques et de l'informatique. Elle s'intéresse à la conception et l'étude d'algorithmes pour obtenir des solutions (approchées) à des équations venant de modèles issus de la physique, de la biologie, de la finance, etc.

Exemple: aérodynamique

Expérience



Simulation numérique

Objectifs:

- Proposer une solution approchée, calculée à l'aide de l'ordinateur
- Etudier la convergence et la stabilité de la méthode numérique
- Etudier le coût algorithmique

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Exemple: apprentissage de reconnaissance d'objets

Exemple réussi

Exemple d'échec

Objectifs:

- Proposer des algorithmes d'apprentissage efficaces
- Etudier la convergence, la stabilité et le coût algorithmique

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Principe de l'analyse numérique

Motivation

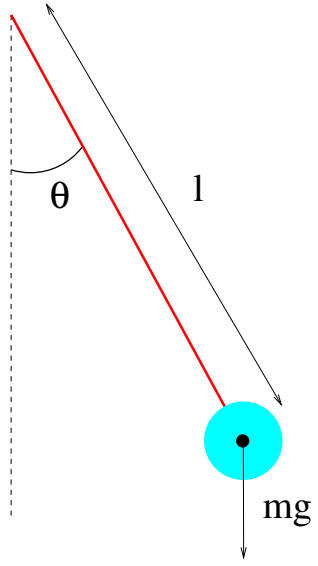
- Les modèles considérés sont composés d'ensemble d'équations dont on ne sait pas déterminer de solutions explicites
- Proposer une solution approchée, calculée à l'aide de l'ordinateur.

Développer des algorithmes efficaces

- Convergence et stabilité de la méthode numérique
- Coût algorithmique

Quelques exemples

Mouvement du pendule



- Equation

$$\begin{cases} \theta''(t) + \frac{g}{l}\sin(\theta(t)) = 0, \\ \theta(0) = \theta_0, \theta'(0) = \theta_1 \end{cases}$$

- Solution exacte ?

- Solution approchée pour $\theta << 1$

$$\begin{cases} \theta''(t) + \frac{g}{l}\theta(t) = 0, \\ \theta(t) = \theta_0 \cos\left(\sqrt{\frac{g}{l}}t\right) + \sqrt{\frac{l}{g}}\theta_1 \sin\left(\sqrt{\frac{g}{l}}t\right) \end{cases}$$

- Utiliser une approximation numérique de la solution !

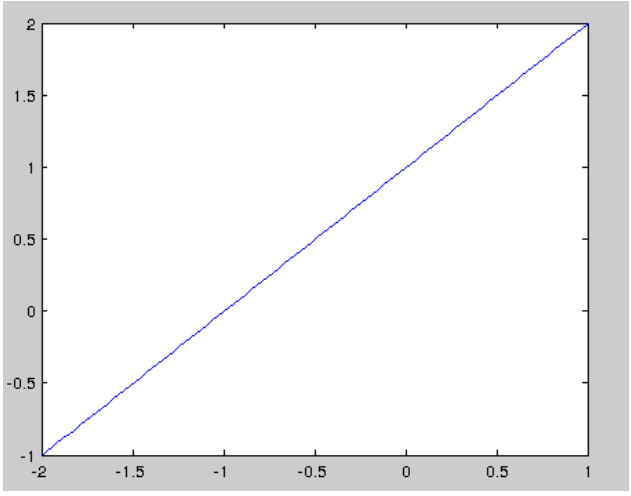
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Quelques exemples

Calculer les racines du polynôme $p(x) = ax^2 + bx + c$

```
>> a = 10^(-20);
>> b = 1;
>> c=1;
>> x1 = (-b + sqrt(b^2 - 4*a*c))/(2*a)

x1 =

     0

>> x2 = (-b - sqrt(b^2 - 4*a*c))/(2*a)

x2 =

  -1.0000e+20
```



INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Quelques exemples

Temps de calcul pour l'inversion d'une matrice

```
>>
>> N = 2^10

N =

        1024

>> N = 2^10;
>> A = rand(N,N);
>> tic;
A^(-1);
toc
Elapsed time is 1.033932 seconds.
>> B = rand(2*N,2*N);
>> tic;
B^(-1);
toc
Elapsed time is 9.313878 seconds.
>>
```

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Représentation des réels en base b

- Tout nombre réel x peut être représenté sous la forme

$$x = \pm m\, b^{\pm e} \quad \text{où} \quad b \in \mathbb{N}^*,\ b \geq 2.$$

  - avec la mantisse m :

$$m = m_1 b^{-1} + m_2 b^{-2} + ..., \quad \text{et} \quad m_i \in \{0, 1, ..., b-1\}$$

  - et l'exposant e :

$$e = e_0 b^0 + e_1 b^1 + ...e_{s-1} b^{s-1}, \quad \textit{avec} \quad s \in \mathbb{N}^*$$

- Représentation unique ?
- Ordinateur : mémoire finie !
  - La mantisse est tronquée au bout de r chiffres,
  - Valeur maximale pour s

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Le standard IEEE (754-1985)

- Float double précision : utilisation de 8 octets (64 bits), $b = 2$

- Pour la mantisse : utilisation de 52 bits

$$m = 2^{-1} + m_2 2^{-2} + m_3 2^{-3} + ... + m_{53} 2^{-53},$$

- Pour l'exposant : 11 bits : $e \in [-1022, 1025]$ avec

$$e = c_0 2^0 + c_1 2^1 + ... + c_{10} 2^{10} - 1022, \quad \text{avec} \quad c_i \in \{0, 1\}$$

- Les valeurs extrêmes $e = -1022$ et $e = 1025$ sont réservées à la représentation de 0, de Inf et de NAN (not a number)
- On appelle $\mathbb{F}$ l'ensemble fini de ces nombres

Exercices

Exercice 1

Calculer x_{max} le nombre le plus grand de $\mathbb{F}$ et x_{posmin}, le nombre positif le plus petit de $\mathbb{F}$

Exercice 2

Proposer deux algorithmes pour déterminer respectivement une approximation numérique de x_{max} et x_{posmin}

Erreur d'arrondi

- Sur l'ordinateur toutes les données initiales et tous les résultats de calcul sont automatiquement dans $\mathbb{F}$
- Un nombre réel $x = \pm m\, 2^e$, $m = 0.1 m_2 m_3 \ldots$ avec $x_{posmin} < |x| < x_{max}$ est arrondi de la manière suivante

$$\mathrm{fl}(x) = \mathrm{sign}(x) \begin{cases} 0.1 m_2 \ldots m_{53} 2^e & \text{si} \quad m_{54} = 0, \\ (0.1 m_2 \ldots m_{53} + 2^{-53}) 2^e & \text{si} \quad m_{54} = 1, \end{cases}$$

Exercice 3

Quel est le plus petit nombre de $\mathbb{F}$ supérieur à 1 ? Proposer un algorithme pour retrouver numériquement la différence entre ce nombre et 1.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Erreur d'arrondi

- Si $x \in \mathbb{R}$, $x_{posmin} < |x| < x_{max}$, alors
  - L'erreur d'arrondi absolue est

$$|x - \mathrm{fl}(x)| \leq \frac{1}{2} 2^{-53} 2^{e}.$$

  - L'erreur d'arrondi relative est

$$\left| \frac{x - \mathrm{fl}(x)}{x} \right| \leq \frac{1}{2} \frac{2^{-53} 2^{e}}{|m| 2^{e}} \leq 2^{-53} \simeq 10^{-16}$$

Conditionnement d'un problème numérique

Supposons qu'on souhaite calculer la solution y d'un problème à partir de données x. Le procédé est décrit par l'application

$$f : X \ni x \mapsto y = f(x) \in Y,$$

avec X, Y des espaces normés. Les données seront toujours erronées, en pratique on obtient un résultat $\tilde{y}$ à partir de données $\tilde{x}$.

Définition

Les erreurs absolues et relatives sont définies par

$$\delta x = \|\tilde{x} - x\|_X\,,\ \delta y = \|\tilde{y} - y\|_Y\,,\ \text{respectivement}\ \frac{\delta x}{\|x\|_X},\ \frac{\delta y}{\|y\|_Y}$$

Conditionnement d'un problème numérique

Définition

Le conditionnement représente la sensibilité du résultat par rapport à de petites variations des données ...
On dit qu'un problème est
- **bien conditionné** si une petite variation des données entraîne une petite variation du résultat
- **mal conditionné** si une petite variation des données entraîne une grande variation du résultat

Exemple : soit $f : \mathbb{R} \to \mathbb{R}$

- Développement de Taylor

$$f(x + \delta x) = f(x) + f'(x)\,\delta x + o(\delta x)$$

$$\delta f := f(x + \delta x) - f(x) \simeq f'(x)\delta x$$

- Conditionnement

$$k = \left| \frac{\delta f}{f(x)} \frac{x}{\delta x} \right| \simeq \left| \frac{x f'(x)}{f(x)} \right|$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Conditionnement des opérations arithmétiques élémentaires

- Soit $f \in \mathbb{C}^2(\mathbb{R}^n, \mathbb{R})$. Quel est le conditionnement de f ?
- D'après le développement de Taylor de f,

$$\delta f(\mathbf{x}) = f(\mathbf{x} + \delta\mathbf{x}) - f(\mathbf{x}) = \sum_{i=1}^{n} \partial_{x_i} f(\mathbf{x}) \delta x_i + 0(|\delta\mathbf{x}|^2).$$

- Le conditionnement est donc déterminé par les nombres

$$k_i = \left| \frac{\partial_{x_i} f(\mathbf{x}) \quad x_i}{f(\mathbf{x})} \right|.$$

Exercice 4

Calculer le conditionnement de l'addition et de la multiplication

Stabilité d'un algorithme

Définition

La stabilité d'un algorithme se réfère à la propagation des erreurs au cours des étapes du calcul, à la capacité de l'algorithme de ne pas trop amplifier d'éventuels écarts, à la précision des résultats obtenus.

Exemple : l'algorithme qui calcule les racines du polynôme $p(x) = ax^2 + bx + c$ basé sur les formules

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

s'avère instable en pratique : exemple avec $a = 10^{-20}, b = 1, c = 1$

Exercice 5

Proposer un autre algorithme plus stable

Complexité algorithmique

Définition (Coût algorithmique)

Nombre d'opérations élémentaires effectuées par l'algorithme

Exemple : Évaluation du polynôme $p(x) = \sum_{i=0}^{n} a_i x^i$:

- Nombre de multiplications : $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$
- Nombre d'additions : n
- Coût algorithmique de l'ordre $O(n^2)$

Définition

$$\begin{cases} O(n^k) \rightarrow \text{coût polynomial} \\ O(a^n) \rightarrow \text{coût exponentiel} \\ O(n!) \rightarrow \text{coût factoriel} \end{cases}$$

Complexité algorithmique

Exemple : Evalutation du polynôme $p(x) = \sum_{i=0}^{n} a_i x^i$:

- Schéma de Horner

$$p(x) = a_0 + x\,(a_1 + x\,(a_2 + x\,(...)))$$

- Coût algorithmique ?

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Exercice 6

Soit $n \in \mathbb{N}^*$. On s'intéresse à la fonction f définie pour tout $x \in [-1, \infty)$ par

$$f(x) = \begin{cases} \dfrac{(1+x)^{1/n}-1}{x} & \text{si } x \in [-1,\infty) \setminus \{0\} \\ 1/n & \text{si } x = 0 \end{cases}.$$

- Montrer que cette fonction est bien continue en $x = 0$.
- L'utilisation de cette expression pour déterminer la valeurs de f en $x \simeq 10^{-17}$ renvoie la valeur 0. Justifier un tel résultat?
- Lorsque que l'on trace plus précisément cette fonction au voisinage de $x = 0$, on observe de fortes oscillations. Déterminer une majoration de ces oscillations en fonction de x au voisinage de 0.
- En remarquant que $\sum_{k=0}^{n-1} y^k = (y^n - 1)/(y - 1)$, montrer que f vérifie

$$f(x) = \frac{1}{\sum_{k=0}^{n-1}(1+x)^{k/n}}.$$

- Justifier que l'utilisation d'un nouvel algorithme basé sur cette nouvelle expression sera mieux conditionné.

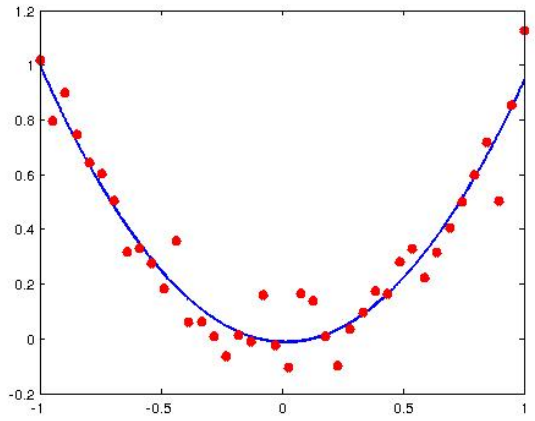
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Plan du chapitre

1. Introduction
2. Méthode des moindres carrés linéaires
   - Paramètres optimaux et calcul de gradient
   - Exemples d'exercices
3. Optimisation et méthode du gradient
4. Régression logistique
5. Introduction aux réseaux de neurones

Cadre mathématique et modèle

- Un échantillon de données $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_i$.
  Par exemple un nuage de points.



- Un modèle composant un ensemble de fonctions hypothèse $\mathcal{H} = \{h(x;\theta)\}_\theta$ paramétrées par le jeu de paramètres θ.
  Par exemple dans le cas d'une régression linéaire,

$$h(x;\theta) = \theta_0 + \theta_1 x.$$

Cadre mathématique et modèle

- Une mesure d'erreur (fonction de coût) $J(\theta; \mathcal{D})$ décrivant une distance du modèle par rapport au données.
  Par exemple, une erreur quadratique:

$$J(\theta) = \sum_{i=1}^{m} \left(h(x^{(i)}; \theta) - y^{(i)} \right)^2$$

- Le meilleur choix de paramètre θ minimisant la fonctionnelle J
  $\Rightarrow$ nécessite des algorithmes d'optimisation

Exemples de modèle dans le cas linéaire

Supposons pour l'instant $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m} \subset (\mathbb{R}^2)^m$

- régression linéaire, $\theta = (\theta_0, \theta_1)$

$$h(x; \theta) = \theta_0 + \theta_1 x$$

- cas général, $\theta = (\theta_0, \ldots, \theta_d)$

$$h(x; \theta) = \theta_0 \phi_0(x) + \ldots + \theta_d \phi_d(x)$$

avec certaines fonctions $\phi_0, \ldots, \phi_d$.

Exemple : cas de polynôme de degrés d

Le choix $\phi_0(x) = 1,\ \phi_1(x) = x, \ldots, \phi_d(x) = x^d$ permet d'utiliser un modèle polynomiale de degré d.

La fonction de coût

- Dans le cas de la régression linéaire, la fonction de coût est définie par

$$J(\theta) = \sum_{i=1}^{m} \left(h(x^{(i)}; \theta) - y^{(i)} \right)^2 = \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)^2$$

- Et de façon analogue dans le cas général, on utilisera

$$J(\theta) = \sum_{i=1}^{m} \left(h(x^{(i)}; \theta) - y^{(i)} \right)^2 = \sum_{i=1}^{m} \left(\sum_{k=0}^{d} \theta_k \phi_k(x^{(i)}) - y^{(i)} \right)^2$$

Paramètres optimaux et calcul de gradient

1 Introduction

2 Méthode des moindres carrés linéaires
- Paramètres optimaux et calcul de gradient
- Exemples d'exercices

3 Optimisation et méthode du gradient

4 Régression logistique

5 Introduction aux réseaux de neurones

Cas particulier d'une droite

Proposition

Le jeu de paramètres optimal (θ_0^*, θ_1^*) qui minimise

$$J(\theta) = \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)} \right)^2 .$$

vérifie

$$\begin{cases} \theta_0^* & = \frac{1}{D} \left(\sum_{i=1}^m y^{(i)} \sum_{i=1}^m (x^{(i)})^2 - \sum_{i=1}^m x^{(i)} \sum_{i=1}^m x^{(i)} y^{(i)} \right) \\ \theta_1^* & = \frac{1}{D} \left(m \sum_{i=1}^m x^{(i)} y^{(i)} - \sum_{i=1}^m x^{(i)} \sum_{i=1}^m y^{(i)} \right), \end{cases}$$

où $D = m \sum_{i=1}^m (x^{(i)})^2 - (\sum_{i=1}^m x^{(i)})^2 \neq 0$.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Cas général

On cherche le jeu de paramètre optimal $(\theta_0^*, \theta_1^*, \cdots, \theta_k^*)$ qui minimise

$$J(\theta) = \sum_{i=1}^{m} \left(\sum_{k=0}^{d} \theta_k \phi_k(x^{(i)}) - y^{(i)} \right)^2$$

Proposition (formule vectoriée)

La fonctionelle J s'identifie à

$$J(\theta) = \|X\theta - y\|_2^2,$$

où

$$X = \begin{pmatrix} \phi_0(x^{(1)}) & \cdots & \phi_d(x^{(1)}) \\ \vdots & & \vdots \\ \phi_0(x^{(m)}) & \cdots & \phi_d(x^{(m)}) \end{pmatrix} \text{ et } y = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

Définition du gradient d'une fonctionnelle

Définition (Différentielle de J)

La différentielle de $J : \mathbb{R}^p \to \mathbb{R}$ en θ dans la direction φ est noté $DJ(\theta)(\varphi)$

$$DJ(\theta)(\varphi) = \lim_{t \to 0} \frac{J(\theta + t\varphi) - J(\theta)}{t}.$$

Définition (Gradient de J)

Le gradient d'une fonctionnelle $J : \mathbb{R}^p \to \mathbb{R}$ en θ par rapport à un produit scalaire $\langle ., . \rangle$ donné est alors défini par

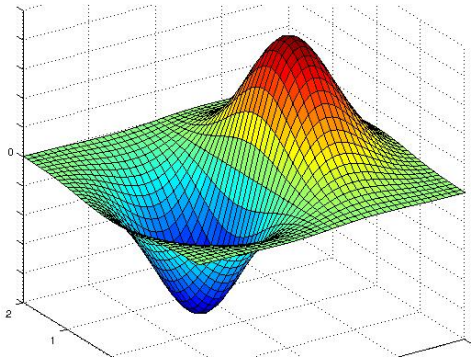
$$\langle \nabla J(\theta), \varphi \rangle = DJ(\theta)(\varphi), \quad \forall \varphi \in \mathbb{R}^p.$$

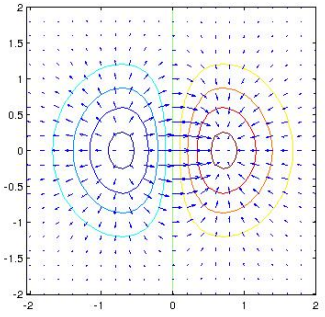
Proposition

Dans le cas du produit scalaire canonique $\langle x, y \rangle = \sum_{i=0}^{p} x_i y_i$,

$$(\nabla \phi(\theta))_i = \partial_{\theta_i} \phi(\theta).$$

Exemples





Proposition

Les gradients des fonctionnelles suivantes vérifient

$$\begin{cases} J_1(\theta) = <\theta, y> \quad \Rightarrow \quad \nabla J_1(\theta) = y \\ J_2(\theta) = <X\theta, y> \quad \Rightarrow \quad \nabla J_2(\theta) = X^T y \\ J_3(\theta) = \frac{1}{2} <X\theta, X\theta> \quad \Rightarrow \quad \nabla J_3(\theta) = X^T X\theta \\ J_4(\theta) = \frac{1}{2} <X\theta - y, X\theta - y> \quad \Rightarrow \quad \nabla J_4(\theta) = X^T(X\theta - y) \end{cases}$$

Solution: cas général, approche analytique

- Le gradient de $J(\theta)$

$$J(\theta) = \sum_{i=1}^{m} \left(\sum_{k=0}^{d} \theta_k \phi_k(x^{(i)}) - y^{(i)} \right)^2 = \|X\theta - y\|_2^2$$

s'identifie à

$$\nabla J(\theta) = 2(X^T X\theta - X^T y).$$

- Le jeu de paramètre optimal θ^* est donc solution du système d'équations

$$\boxed{X^T X\theta^* = X^T y}$$

Solution

- Si le rang de la $X \in \mathbb{R}^{m \times (d+1)}$ est égal à $\text{rang}(X) = d + 1$, alors la matrice $X^T X$ est inversible et la solution du problème θ^* vérifie

$$\theta^* = (X^T X)^{-1} X^T y$$

En `Matlab` la solution $A^{-1}b$ d'un système linéaire $Ax = b$ (A inversible) peut être calculée à l'aide de la commande $\backslash$:

$$\boxed{x = A \backslash b}$$

- Dans le cas contraire où $\text{rang}(X) < d + 1$, le problème d'optimisation admet plusieurs solutions et une des solutions θ^* peut être calculée à l'aide de la matrice pseudo-inverse de $X^T X$ disponible en `Matlab` avec la commande `pinv`

Exemples d'exercices

1 Introduction

2 Méthode des moindres carrés linéaires
- Paramètres optimaux et calcul de gradient
- Exemples d'exercices

3 Optimisation et méthode du gradient

4 Régression logistique

5 Introduction aux réseaux de neurones

Exercice 7

Montrer que dans le cas de la régression linéaire, on a bien

$$X^T X = \begin{pmatrix} m & \left(\sum_{i=1}^m x^{(i)}\right) \\ \left(\sum_{i=1}^m x^{(i)}\right) & \left(\sum_{i=1}^m (x^{(i)})^2\right) \end{pmatrix} \quad \text{et} \quad X^T y = \begin{pmatrix} \sum_{i=1}^m y^{(i)} \\ \sum_{i=1}^m x^{(i)} y^{(i)} \end{pmatrix}.$$

Exercice 8

Dans le cas d'un modèle de la forme

$$h(x;\theta) = \theta_0 + \theta_1 x^2,$$

donner l'expression de la matrice $X^T X$ et du vecteur $X^T y$.
En déduire la parabole paire qui approche au mieux les points

$$\{(x^{(i)}, y^{(i)})\}_i = \{(-2, 3.5), (-1, 1.5), (0, 0), (1, 1.5), (2, 3.5)\}$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Exercice 9

On s'intéresse à l'approximation d'une fonction f à partir d'un ensemble de mesures de f et de sa dérivée f' sur un échantillon $(x_i)_{i=1:n}$: $y_i = f(x_i)$ et $y_i' = f'(x_i)$. Pour cela on introduit un modèle à k paramètres $g_\theta(x) = \sum_{j=1}^{k} \theta_j \phi_j(x)$ et on cherche les coefficients $\theta = (\theta_1, \theta_2, \cdots \theta_k)^t$ qui minimisent l'erreur

$$J(\theta) = \sum_{i=1}^{n} (y_i - g_\theta(x_i))^2 + \sum_{i=1}^{n} (y_i' - g_\theta'(x_i))^2.$$

- Montrer que J s'identifie à $J(\theta) = \|X\theta - y\|^2 + \|M\theta - y'\|^2$, où $y = (y_1, y_2, \cdots, y_n)^t$, $y' = (y_1', y_2', \cdots, y_n')^t$, $M_{i,j} = \phi_j(x_i)$ et $X_{i,j} = \phi_j'(x_i)$.
- Déterminer l'expression du gradient de J
- En déduire que le meilleur jeu de coefficients θ est solution du système

$$(M'M + X'X)\theta = M'y + X'y'.$$

On souhaite maintenant approcher la fonction $\sin(x)$ avec un modèle à $k = 2$ paramètres $g_\theta(x) = \theta_1 x + \theta_2 x^3$ à partir de l'échantillon $\{x_i\}_{i=1:5} = \{0, \pi/6, \pi/4, \pi/3, \pi/2\}$. Montrer que le jeu optimal de coefficients $\theta^* = (\theta_1^*, \theta_2^*)^t$ est soluton du système

$$\begin{pmatrix} 9.45 & 21.1 \\ 21.11 & 86.3 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} 6.37 \\ 8.95 \end{pmatrix}$$

On pensera à préciser les valeurs des matrices M et X.

- Tracer sur un même graphique les points x_i et le modèle optimal obtenu $g_{\theta*}(x)$

Optimisation et méthode du gradient

1 Introduction

2 Méthode des moindres carrés linéaires

3 Optimisation et méthode du gradient
- Exitence, unicité et caractérisation d'un minimum de J
- Principe de la méthode du gradient
- Méthode du gradient à pas fixe et stabilité
- Cas quadratique avec pas de descente optimal
- Variantes
- Exemples d'exercices

4 Régression logistique

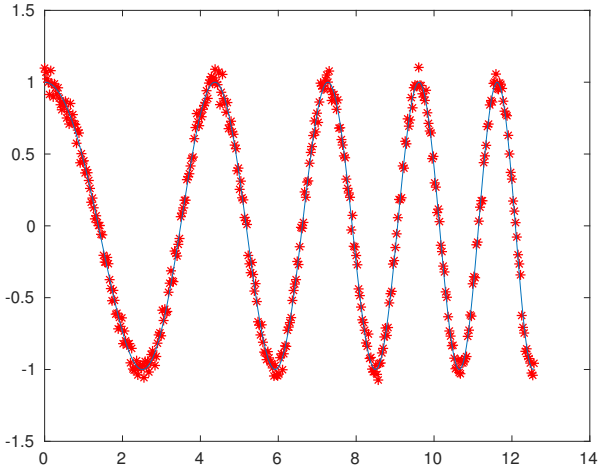
5 Introduction aux réseaux de neurones

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Motivation

- Un modèle de la forme $h(x, \theta) = \cos(\theta_0 x + \theta_1 x^2)$



- Trouver les meilleurs coefficients θ^* qui minimisent

$$J(\theta) = \sum_{i=1}^{m} (h(x^{(i)}, \theta) - y^{(i)})^2.$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Motivation

- Le gradient de J vérifie

$$\nabla J(\theta) = -2\sum_{i=1}^{m}(h(x^{(i)},\theta) - y^{(i)}) \begin{pmatrix} sin(\theta_0 x_{(i)} + \theta_1 x_{(i)}^2)x_{(i)} \\ sin(\theta_0 x_{(i)} + \theta_1 x_{(i)}^2)2x_{(i)}^2 \end{pmatrix}$$

- Trouver $\theta^* = (\theta_0^*, \theta_1^*)$ tels que

$$\nabla J(\theta^*) = 0,$$

nécessite **la résolution d'un système non linéaire !**

$\Longrightarrow$ Utiliser un algorithme itératif pour trouver un minimimum de J.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Optimisation et méthode du gradient

- Objectif: soit $J : E \mapsto \mathbb{R}$, une fonction continue définie sur un espace vectoriel E normé. Nous recherchons ici un **minimum global** de J, c'est à dire un point $x^* \in E$, qui vérifie

$$J(x^*) = \min_{x \in E} J(x).$$

- Question: Existence, unicité d'un tel minimum?

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Exitence, unicité et caractérisation d'un minimum de J

1 Introduction

2 Méthode des moindres carrés linéaires

3 Optimisation et méthode du gradient
- Exitence, unicité et caractérisation d'un minimum de J
- Principe de la méthode du gradient
- Méthode du gradient à pas fixe et stabilité
- Cas quadratique avec pas de descente optimal
- Variantes
- Exemples d'exercices

4 Régression logistique

5 Introduction aux réseaux de neurones

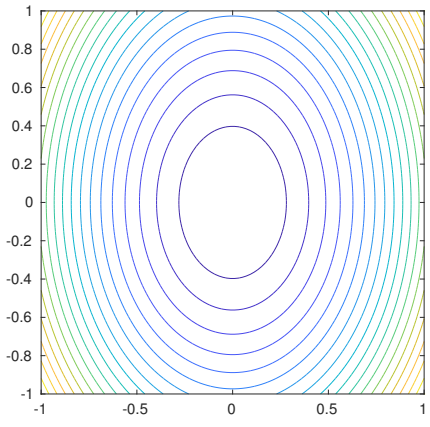
Exemple d'une fonction J convexe

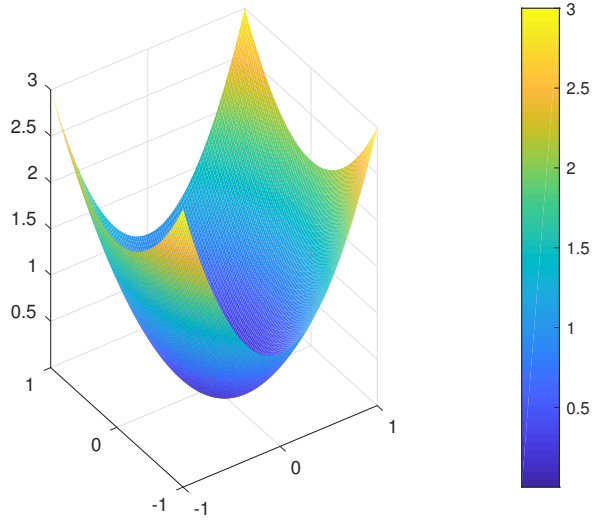
- Exemple où J est quadratique

$$J(x) = 2x_1^2 + x_2^2$$

- Son gradient est linéaire et s'identifie à

$$\nabla J(x) = \begin{pmatrix} 4x_1 \\ 2x_2 \end{pmatrix}$$





- Existence d'un minimum local ? global ?

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

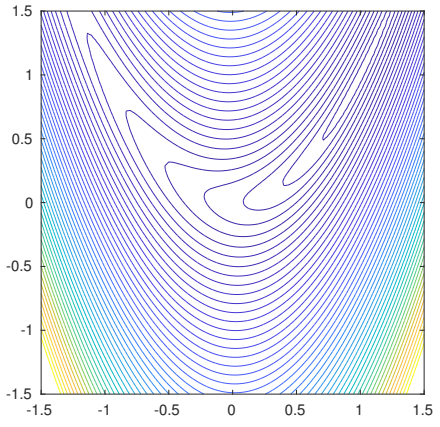
Exemple d'une fonction J non convexe

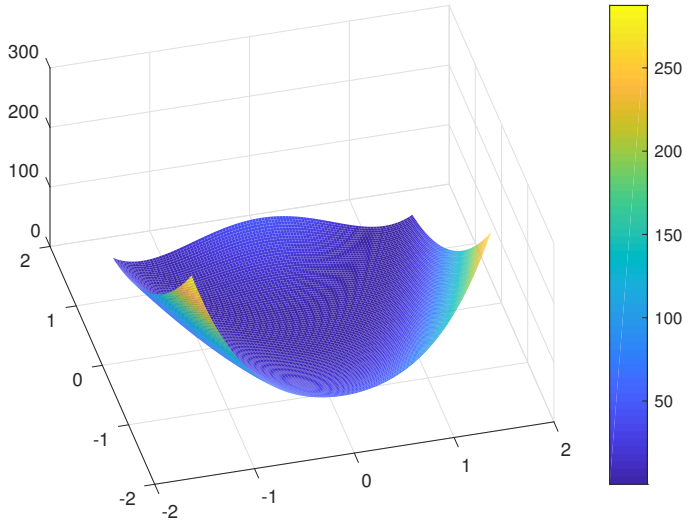
- Fonction de Rosenbrock

$$J(x) = (x_1 - 1)^2 + p(x_1^2 - x_2)^2$$

- Son gradient est non linéaire et s'identifie à

$$\nabla J(x) = \begin{pmatrix} 2(x_1 - 1) + 4p(x_1^2 - x_2)x_1 \\ -2p(x_1^2 - x_2) \end{pmatrix}$$





- Existence d'un minimum local ? global ?

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Existence, unicité et caractérisation

On suppose ici que E est une espace de dimension finie.

Proposition (Existence)

Soit $J : E \to \mathbb{R}$ une application continue telle que $J(x) \to \infty$ lorsque $\|x\| \to \infty$, alors il existe $x^* \in E$ solution de $J(x^*) = \min_{x \in E} J(x)$.
Si de plus J est différentiable en x^* alors $\nabla J(x^*) = 0$.

Proposition (unicité)

Soit $J : E \to \mathbb{R}$ une strictement convexe alors il existe au plus un $x^* \in E$ solution de $J(x^*) = \min_{x \in E} J(x)$.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Principe de la méthode du gradient

1 Introduction

2 Méthode des moindres carrés linéaires

3 Optimisation et méthode du gradient
- Exitence, unicité et caractérisation d'un minimum de J
- Principe de la méthode du gradient
- Méthode du gradient à pas fixe et stabilité
- Cas quadratique avec pas de descente optimal
- Variantes
- Exemples d'exercices

4 Régression logistique

5 Introduction aux réseaux de neurones

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Algorithme itératif et méthodes du gradient

L'algorithme : L'idée est d'approcher le minimum x^* d'une fonctionnelle J en utilisant un algorithme du gradient:

Définition (Algorithme du gradient)

$$\begin{cases} x_0 \in \mathbb{R}^n \\ x_{k+1} = x_k + \alpha_k d_k \quad \forall k \geq 1, \end{cases}$$

où d_k est une direction de descente et $\alpha_k > 0$ un pas de descente.

Critère d'arrêt : En pratique, l'algorithme s'arrête lorsque les x_k n'évoluent plus. On pourra par exemple utiliser avec un critère d'arrêt de la forme $\|x_{k+1} - x_k\| \leq \varepsilon$, où encore $\|\nabla J(x_k)\| \leq \varepsilon$.
Dans chaque cas, l'approximation sera d'autant plus précise que $\varepsilon \to 0$.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Choix de la direction de descente d_k

Questions:

- Comment choisir une direction de descente pour minimiser J ?
- Quelle est la meilleure direction de descente ?

Définition

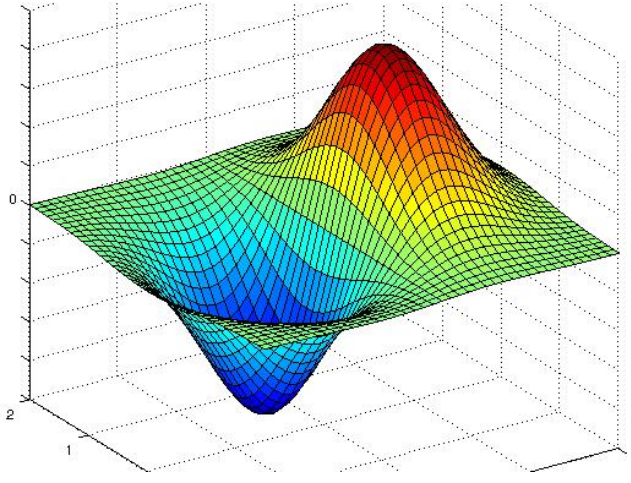
Un vecteur d est une direction de descente pour la fonction J en x si $J'(x)(d) \leq 0$ où $J'(x)(d)$ représente la dérivée directionnelle de J en x dans la direction d:

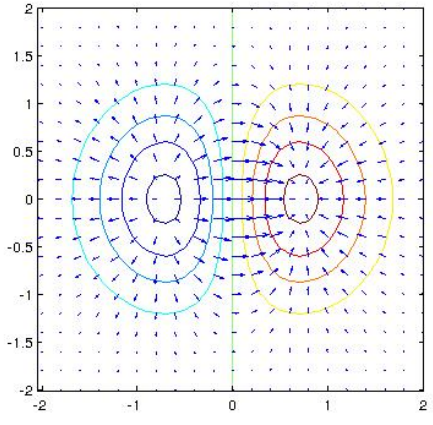
$$J'(x)(d) = \lim_{h \to 0} \frac{J(x + hd) - J(x)}{h} = \langle \nabla J(x), d \rangle.$$

Exemple avec $d = -\nabla J(x)$

Proposition

L'opposé du gradient $d = -\nabla J(x)$ représente la meilleure direction de descente.





Choix du pas de descente

- Méthode du gradient à pas fixe :

$$\alpha_k = \alpha > 0, \quad \forall k \in \mathbb{N}.$$

Comment s'assurer de la décroissance de J?

- Méthode du gradient à pas optimal : trouver α_k tel que

$$J(x_k + \alpha_k d_k) = \min_{\alpha > 0} J(x_k + \alpha d_k).$$

C'est un problème d'optimisation qui peut être compliqué !

- Variante de Wolfe : adaptation du pas de temps mais en relaxant son coté optimal.

Méthode du gradient à pas fixe et stabilité

1 Introduction

2 Méthode des moindres carrés linéaires

3 Optimisation et méthode du gradient
- Exitence, unicité et caractérisation d'un minimum de J
- Principe de la méthode du gradient
- Méthode du gradient à pas fixe et stabilité
- Cas quadratique avec pas de descente optimal
- Variantes
- Exemples d'exercices

4 Régression logistique

5 Introduction aux réseaux de neurones

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

En dimension 1

Proposition

Soit J est une fonction 2 fois dérivable telle qu'il existe une constante L telle que pour tout $x \in \mathbb{R}$,

$$|J''(x)| \leq L.$$

Si de plus $\alpha \leq \frac{1}{L}$, alors

$$J(x_{k+1}) \leq J(x_k) - \frac{\alpha}{2} J'(x_k)^2,$$

où $x_{k+1} = x_k - \alpha J'(x_k)$.

En dimension 1

Preuve : On vérifie tout d'abord que $\forall h \in \mathbb{R}$,

$$J(x+h) \leq J(x) + J'(x)h + \frac{L}{2}h^2.$$

Par ailleurs

$$\begin{aligned} J(x_{k+1}) &= J(x_k + \alpha d_k) \leq J(x_k) + \alpha J'(x_k)d_k + \frac{L}{2}\alpha^2 d_k^2 \\ &\leq J(x_k) - \alpha(1 - \frac{L\alpha}{2})J'(x_k)^2 \leq J(x_k) - \frac{\alpha}{2}(J'(x_k))^2 \end{aligned}$$

La contrainte de stabilité $\alpha \leq \frac{1}{L}$ permet donc d'assurer la décroissance de J au cours des itérations.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Cas général

Proposition

Si $J : E \to \mathbb{R}$ est 2 fois dérivable avec $\|D^2 J\| \leq L$, et si $\alpha \leq \frac{1}{L}$, alors

$$J(x_{k+1}) \leq J(x_k) - \frac{\alpha}{2}\|\nabla J(x_k)\|^2,$$

où $x_{k+1} = x_k - \alpha \nabla J(x_k)$.

Proposition

Si on suppose de plus que $J(x) \to \infty$ lorsque $\|x\| \to \infty$, alors

$$\lim_{k \to \infty} \|\nabla J(x_k)\|^2 = 0.$$

Si la suite minimisante $\{x_k\}_{k \geq 0}$ converge, alors elle converge vers un minimum local de J.

Convergence?

Preuve : On rappelle que J est continue et vérifie $J(x) \to \infty$ lorsque $\|x\| \to \infty$. En particulier, on en déduit que J est minorée, c'est à dire qu'il existe $m \in \mathbb{R}$ telle $m \leq J(x)$, $\forall x \in E$.
La suite $\{J(x_k)\}_k$ est donc décroissante et minorée par m, il existe donc $\ell \in \mathbb{R}$ telle que $\lim_{k\to\infty} J(x_k) = \ell$. Au final,

$$\|\nabla J(x_k)\|^2 \leq \frac{2}{\alpha}(J(x_k) - J(x_{k+1})) \to 0$$

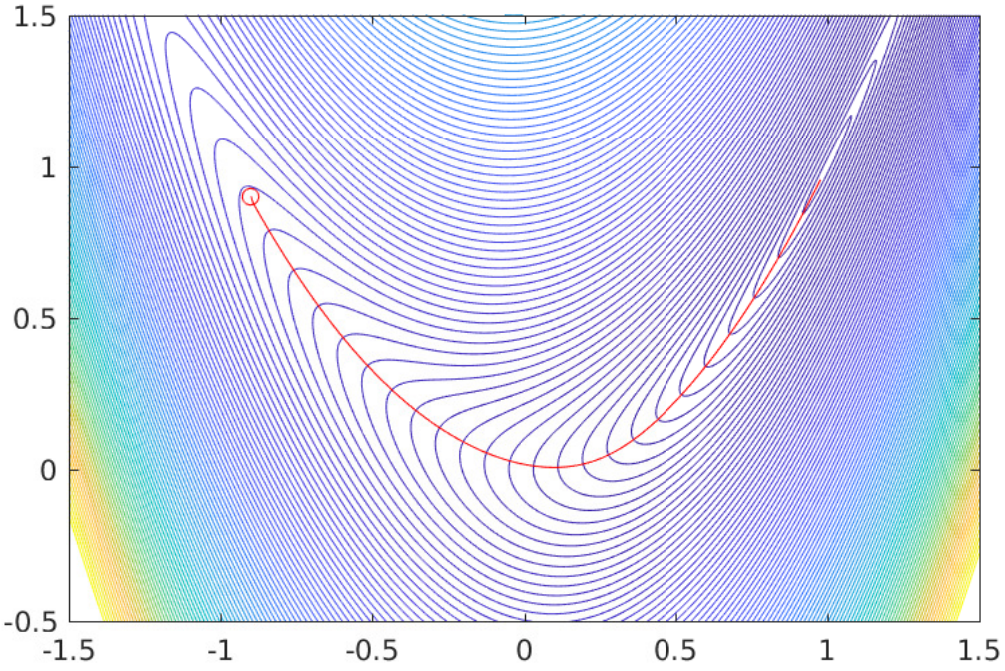
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Optimisation avec la méthode du gradient à pas fixe

- Fonction de Rosenbrock :

$$J(x) = (x_1 - 1)^2 + 10(x_1^2 - x_2)^2$$

- Méthode du gradient à pas fixe : $\alpha = 0.01$
- Critère d'arrêt : $\varepsilon = 10^{-3}$
- Nombre d'itérations nécessaire $N_\varepsilon = 907$



ISA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Cas quadratique avec pas de descente optimal

1 Introduction

2 Méthode des moindres carrés linéaires

3 Optimisation et méthode du gradient
- Exitence, unicité et caractérisation d'un minimum de J
- Principe de la méthode du gradient
- Méthode du gradient à pas fixe et stabilité
- Cas quadratique avec pas de descente optimal
- Variantes
- Exemples d'exercices

4 Régression logistique

5 Introduction aux réseaux de neurones

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Cas quadratique et méthode du gradient à pas optimal

On s'intéresse maintenant au cas quadratique où J s'exprime sous la forme

$$J(x) = \frac{1}{2}\langle Ax, x\rangle - \langle x, b\rangle$$

où A est une matrice symétrique définie positive et $b \in \mathbb{R}^N$.

Proposition

Le gradient de J vérifie

$$\nabla J(x) = Ax - b$$

La direction de descente d_k s'identifie donc

$$d_k = -\nabla J(x_k) = -(Ax_k - b) = b - Ax_k$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Cas quadratique et choix de α_k

On choisit α_k comme le pas qui minimise J dans la direction d_k :

$$\tilde{J}(\alpha) = J(x_k + \alpha d_k).$$

Avec

$$\begin{aligned} \tilde{J}'(\alpha) &= < \nabla J(x_k + \alpha d_k), d_k > \\ &= < A(x_k + \alpha d_k) - b, d_k > \\ &= \alpha < A d_k, d_k > - < d_k, d_k >, \end{aligned}$$

on en déduit que

$$\alpha_k = \frac{< d_k, d_k >}{< A d_k, d_k >}$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Algorithme (Méthode du gradient cas quadratique et pas de temps optimal)

Donnée : $x_0 \in \mathbb{R}^n$, $\epsilon > 0$

While $\|Ax_k - b\| > \epsilon$

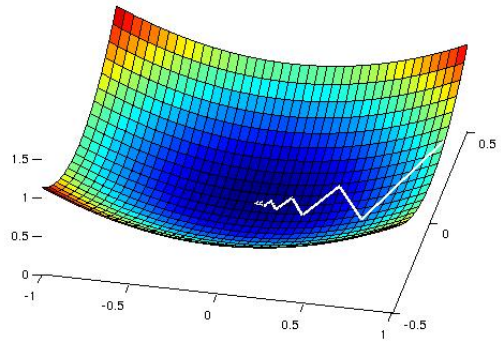
$d_k = b - Ax_k$

$\alpha_k =< d_k, d_k > / < Ad_k, dk >$

$x_{k+1} = x_k + \alpha_k d_k$

End

Exemple avec $A = (1,0; 0,5)$, $b = (0,0)'$ et $x_0 = (1, 0.2)'$



En pratique et dans le cas non quadratique, il ne sera pas toujours facile d'obtenir une expression explicite du pas de temps optimal α_k !

Variantes

1 Introduction

2 Méthode des moindres carrés linéaires

3 Optimisation et méthode du gradient
- Exitence, unicité et caractérisation d'un minimum de J
- Principe de la méthode du gradient
- Méthode du gradient à pas fixe et stabilité
- Cas quadratique avec pas de descente optimal
- Variantes
- Exemples d'exercices

4 Régression logistique

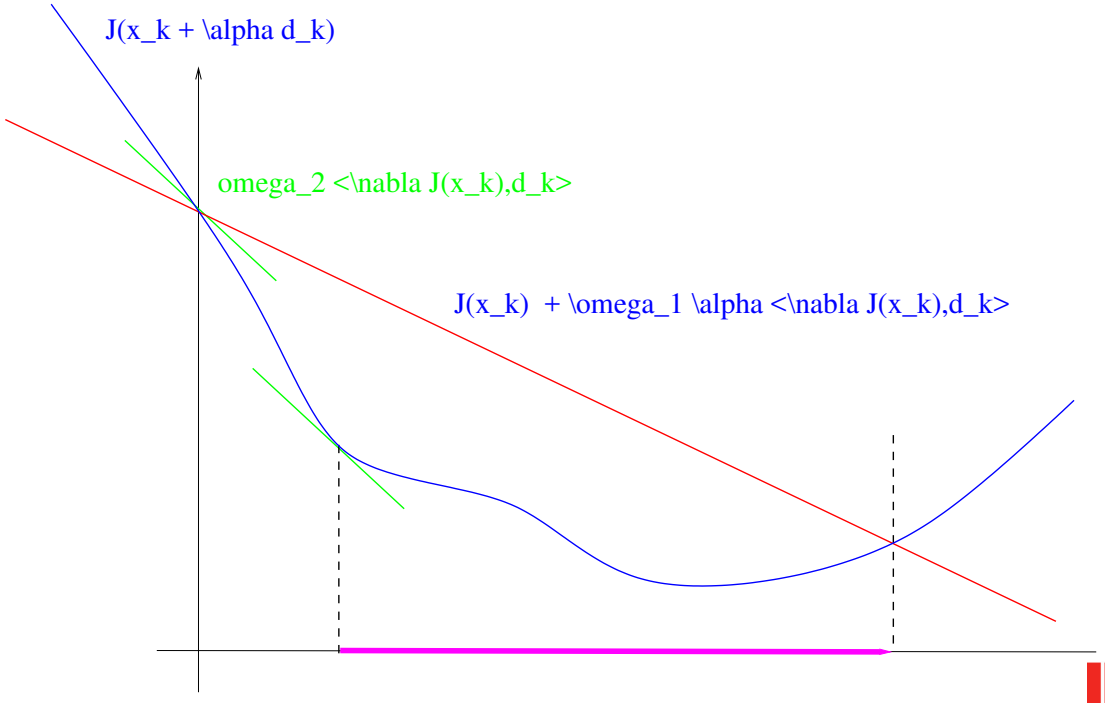
5 Introduction aux réseaux de neurones

Pas de descente optimisé et Règle de Wolfe

Trouver un paramètre α qui vérifie les règles de Wolfe :
avec $0 < \omega_1 < \omega_2 < 1$,

$$J(x_k + \alpha d_k) \leq J(x_k) + \omega_1 \alpha \nabla J(x_k) \cdot d_k, \tag{1}$$

$$\nabla J(x_k + \alpha d_k) \cdot d_k \geq \omega_2 \nabla J(x_k) \cdot d_k. \tag{2}$$



INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Algorithme de Fletchert-Lemaréchal

- Soit $\alpha_{min} = 0$ and $\alpha_{max} = +\infty$, $\tau_i \in]0, 1/2[$ et $\tau_e > 1$,
- Tant que (1) et (2) ne sont pas vérifiées :
  - Si (1) n'est pas vérifiée, alors poser $\alpha_{max} = \alpha$ et $\alpha = (1 - \tau_i)\alpha_{min} + \tau_i \alpha_{max}$
  - Sinon et si (2) n'est pas vérifié, alors poser $\alpha_{min} = \alpha$ et

$$\alpha = \begin{cases} \alpha = (1 - \tau_i)\alpha_{min} + \tau_i \alpha_{max}, \text{ si } \alpha_{max} < +\infty, \\ \alpha = \tau_e \alpha. \end{cases}$$

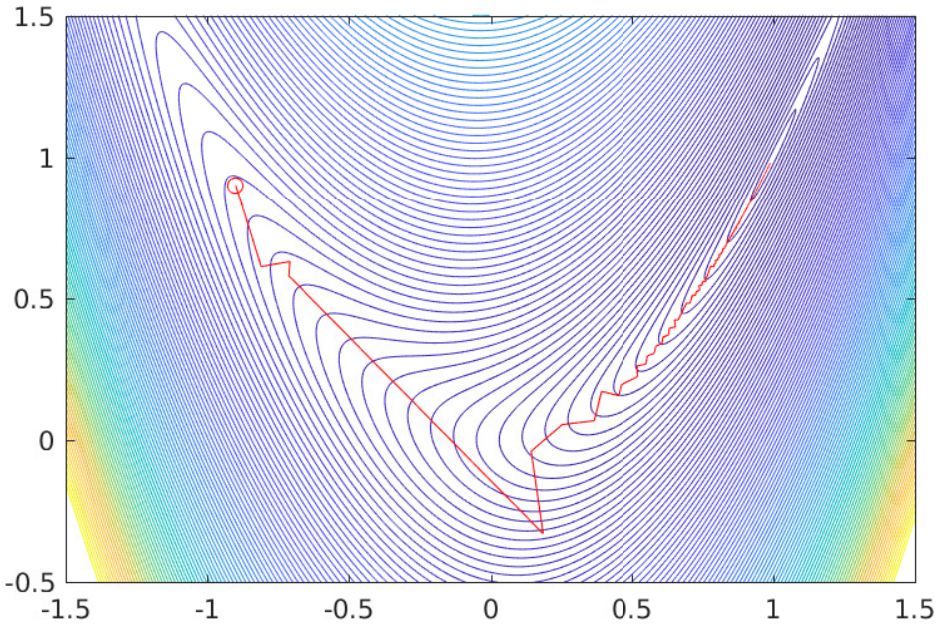
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Méthode du gradient avec les pas de descente optimisé

- Fonction de Rosenbrock :

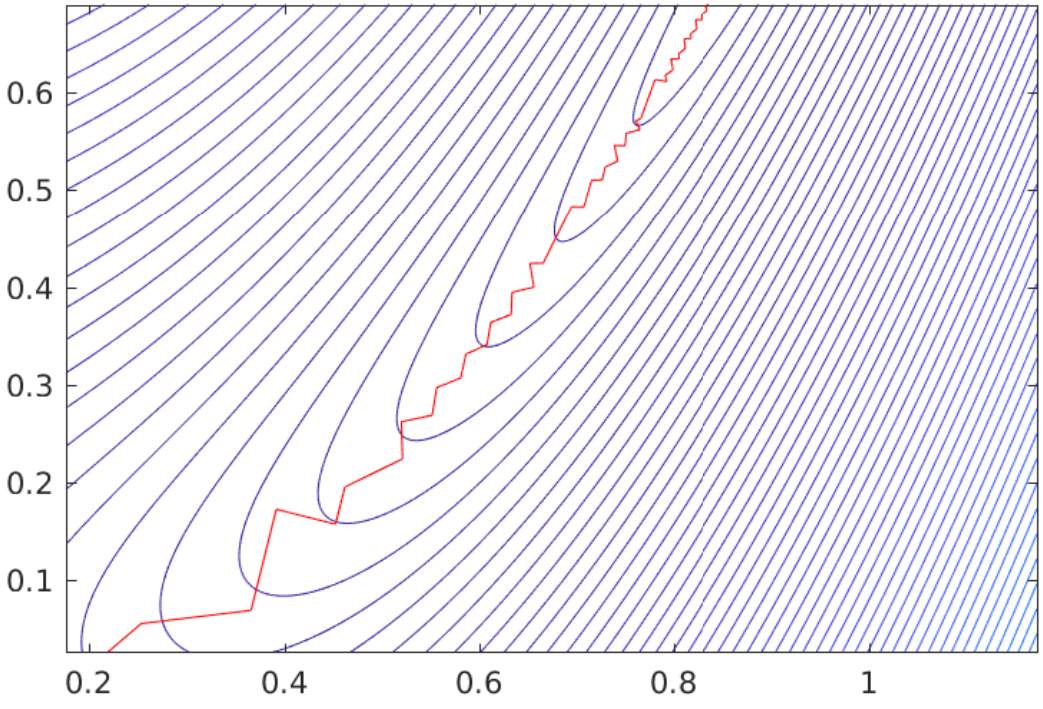
$$J(x) = (x_1 - 1)^2 + 10(x_1^2 - x_2)^2$$

- Choix du pas de descente α_k : critères de Wolfe
- Critère d'arrêt : $\varepsilon = 10^{-3}$
- Nombre d'itérations nécessaire $N_\varepsilon = 347$



INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Phénomène de zigzag



$\Longrightarrow$ Rajouter de l'inertie sur les directions de descente d_k !

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Variantes : méthodes du gradient conjugué non linéaires

- Méthode du gradient conjugué

$$d_k = \begin{cases} -\nabla J(x_k) & \text{si } k = 1 \\ -\nabla J(x_k) + \beta_k d_{k-1} & \text{si } k \geq 2 \end{cases}$$

- Choix du coefficient β_k :

Définition (Approche Fletcher-Reeves)

$$\beta_k = \frac{\|\nabla J(x_k)\|^2}{\|\nabla J(x_{k-1})\|^2}$$

Définition (Approche Polack-Ribière)

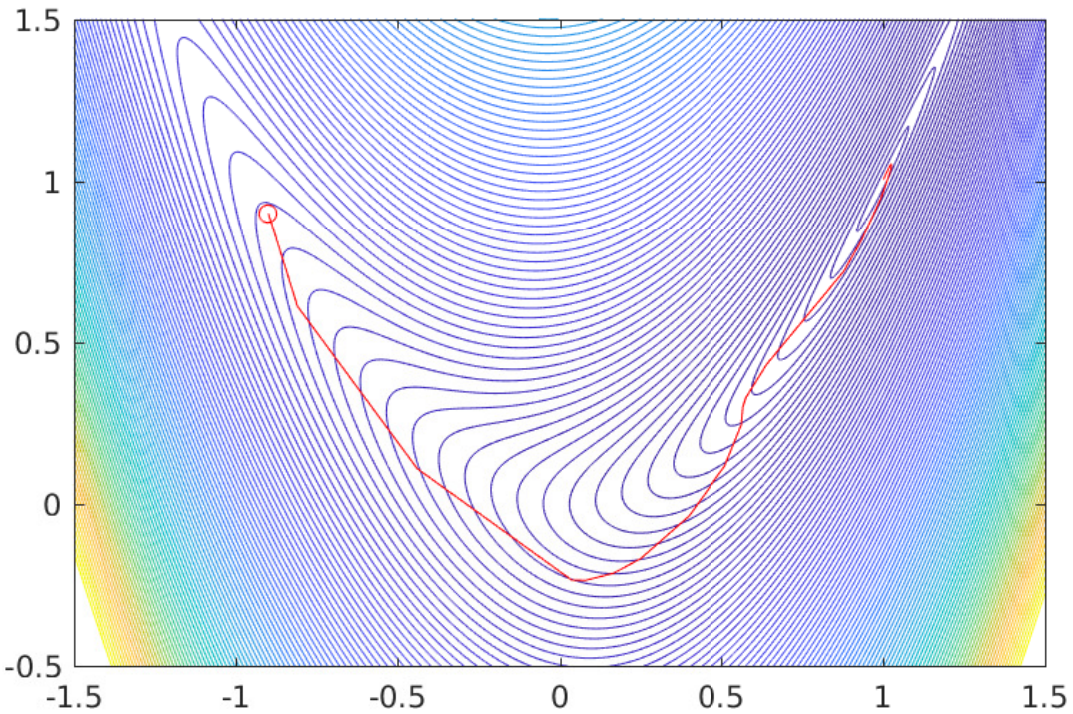
$$\beta_k = \frac{\nabla J(x_k) \cdot (\nabla J(x_k) - \nabla J(x_{k-1}))}{\|\nabla J(x_{k-1})\|^2}$$

Méthode du gradient conjugué de Fletcher-Reeves :

- Fonction de Rosenbrock :

$$J(x) = (x_1 - 1)^2 + 10(x_1^2 - x_2)^2$$

- Choix du pas de descente α_k : critères de Wolfe
- Critère d'arrêt : $\varepsilon = 10^{-3}$
- Nombre d'itérations nécessaire $N_\varepsilon = 28$



INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Calcul des gradients et formule de rétropropagation

1 Introduction

2 Méthode des moindres carrés linéaires

3 Optimisation et méthode du gradient
- Exitence, unicité et caractérisation d'un minimum de J
- Principe de la méthode du gradient
- Méthode du gradient à pas fixe et stabilité
- Cas quadratique avec pas de descente optimal
- Variantes
- Exemples d'exercices

4 Régression logistique

5 Introduction aux réseaux de neurones

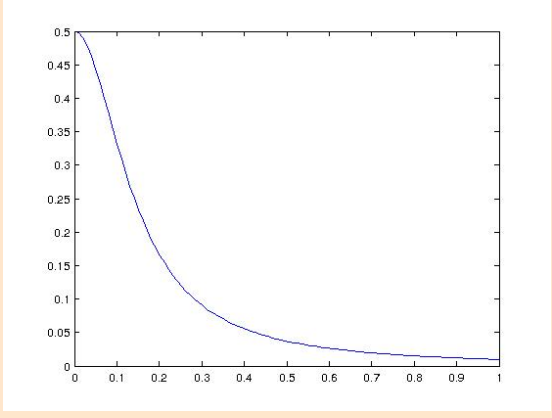
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

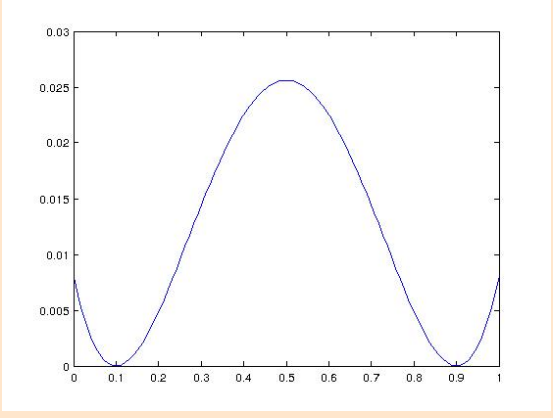
Pole Math

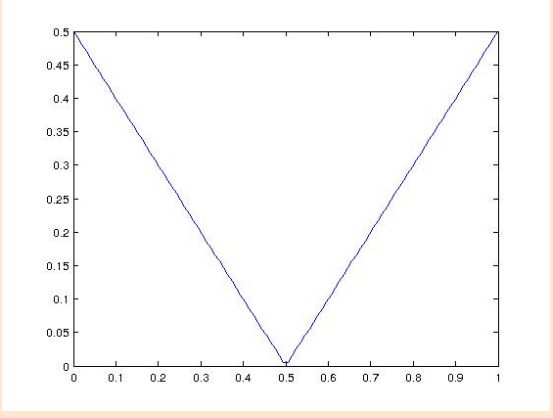
Existence, unicité et caractérisation

Exercice 10

Que peut-on dire de l'existence, de l'unicité et de la caractérisation de minimum de J dans les 3 cas $1D$ suivants ?







INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Exercice 11

On souhaite apliquer minimiser la fonction $f : \mathbb{R}^2 \mapsto \mathbb{R}$ définie pour $X = (x_1, x_2) \in \mathbb{R}^2$ par

$$f(X) = 2x_1^2 + x_2^2 - x_1 x_2 - 3x_1 - x_2 + 4$$

- Montrer que cette fonction admet bien un minimum X^* et expliciter la valeur de celui-ci.
- Appliquer un pas de l'algorithme du gradient avec un pas fixe $\alpha = 0.5$ et en partant de $X_0 = (0, 0)$
- Appliquer un pas de l'algorithme du gradient à pas optimal en partant de $X_0 = (0, 0)$.

Exercice 12

Dans cet exercice, on s'intéresse à la minimisation de la fonction f définie pour tout $x = (x_1, x_2)^t \in \mathbb{R}^2$ par

$$f(x) = \frac{1}{2}\left(\lambda_1 x_1^2 + \lambda_2 x_2^2\right).$$

- Quelles conditions sur les paramètres λ_1, λ_2 assurent l'existence d'un unique minimiseur pour f identifié à $X^* = (0,0)^t$. Par la suite, on suppose que $\lambda_1 = 1$ et $\lambda_2 = 1/2$.
- Montrer que la k-ième itérée de l'algorithme du gradient avec un pas fixe $\alpha > 0$, en partant de la condition initiale $X^0 = (x_1^0, x_2^0)^t$, vérifie
$$X^k = ((1-\alpha\lambda_1)^k x_1^0, (1-\alpha\lambda_2)^k x_2^0)^t.$$
- En déduire les conditions de stabilité sur le pas de descente α pour assurer la décroissance de f.
- On souhaite maintenant déterminer un pas de descente fixe α qui optimise la vitesse de convergence de l'algorithme du gradient. Montrer que
$$\max(|1-\alpha\lambda_1|, |1-\alpha\lambda_2|)^{2k} \min(|x_1^0|, |x_2^0|)^2 \leq \left\|X^k\right\|^2 \leq \max(|1-\alpha\lambda_1|, |1-\alpha\lambda_2|)^{2k} \left\|X^0\right\|^2.$$
- On suppose par la suite que $\min(|x_1^0|, |x_2^0|)^2 > 0$. Que peut-on dire de la vitesse de convergence asymptotique de l'algorithme ?
- Sur l'intervalle $[1/\lambda_1, 1/\lambda_2]$, tracer les fonctions $\alpha \mapsto |1-\alpha\lambda_1|$, $\alpha \mapsto |1-\alpha\lambda_2|$ et $\alpha \mapsto \max(|1-\alpha\lambda_1|, |1-\alpha\lambda_2|)$. En déduire que le pas optimal α vérifie $\alpha = \frac{2}{\lambda_1+\lambda_2}$.
- Dans ce cas, en déduire que $\|X^k\| = \left|\frac{\lambda_1-\lambda_2}{\lambda_1+\lambda_2}\right|^k \|X^0\|$.
- Si $X^0 = (1,1)^t$, combien doit-on effectuer d'itérations de l'algorithme du gradient afin d'obtenir une erreur sur la solution optimale X^* inférieur à 10^{-4} ? Même question dans le cas où $\lambda_1 = 1$ et $\lambda_2 = 10^{-4}$.

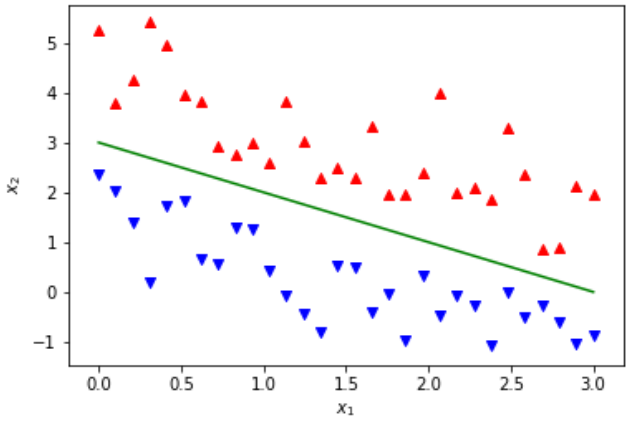
Plan du chapitre

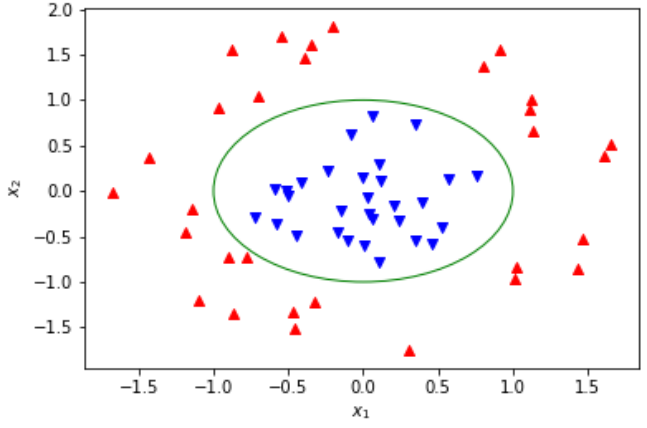
1. Introduction
2. Méthode des moindres carrés linéaires
3. Optimisation et méthode du gradient
4. Régression logistique
   - Introduction
   - Expression du gradient de la fonctionnelle J
   - Exemples numériques
   - Exemples d'exercices
5. Introduction aux réseaux de neurones

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Motivation

- Dans cette partie nous allons développer un algorithme d'apprentissage pour des problèmes de classification:
  - e-mails: spam (oui/non)
  - transactions en ligne: frauduleuse (oui/non)
  - tumeurs: bénigne/maligne





- Exemples: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_i \subset \mathbb{R}^{n+1}$ où

$x^{(i)} \in \mathbb{R}^n$ vecteur de caractéristiques

$y^{(i)} \in \{0, 1\}$ 0: classe négative

1: classe positive

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Introduction

Objectif: Optimiser un modèle $h(x;\theta)$ dépendant de paramètres θ pour approcher la classe de x:

$$h(x;\theta) \simeq P(y=1|x)$$

La classification d'un nouvel élément x s'effectuera à l'aide de h avec

- Si $h(x;\theta) < 0.5$, prédire $y = 0$
- Si $h(x;\theta) \geq 0.5$, prédire $y = 1$

Remarque 1

Un modèle linéaire comme pour la méthode des moindres carrés de la forme

$$h(x;\theta) = \sum_{k=1}^{d} \theta_k \phi_k(x)$$

ne sera pas vraiment adapté ici car on souhaiterait que le modèle $h(x;\theta)$ vérifie

$$0 \leq h(x;\theta) \leq 1$$

Fonction logistique (sigmoïde)

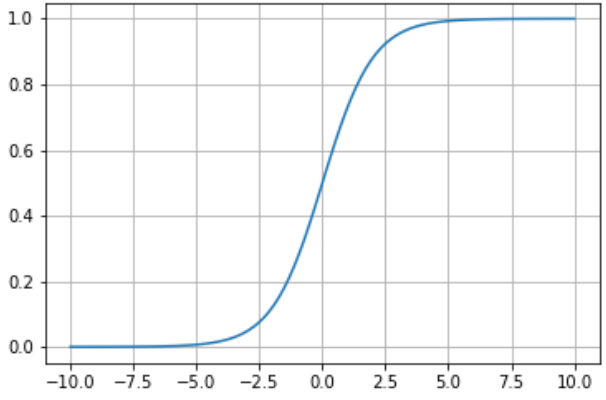
Définition (modèle de classification)

Pour les problèmes de classification il est plus efficace d'utiliser un modèle de la forme

$$h(x;\theta) = g\left(\sum_{k=1}^{d} \theta_k \phi_k(x)\right)$$

où g est une sigmoïde définie par $g(z) = \frac{1}{1+e^{-z}}$

Graphe de g:

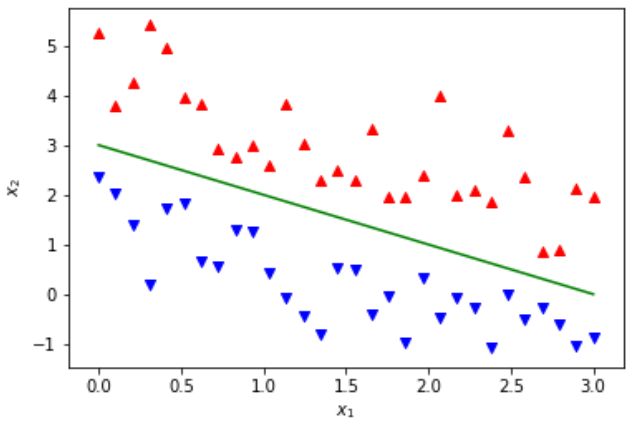


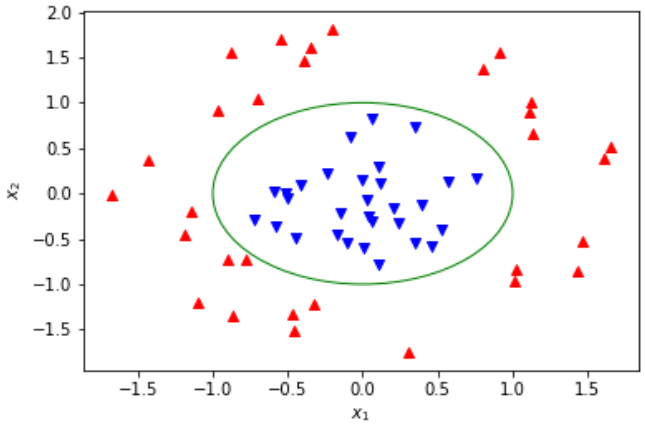
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Exemple de choix de modèles

Une fois le modèle fixé (choix de fonctions ϕ_k), le problème de classification revient à déterminer le meilleur jeu de coefficients θ^* qui permet de séparer-classifier au mieux les données :

- Première exemple (à gauche) $\phi_1(x) = 1$, $\phi_2(x) = x_1$ et $\phi_3(x) = x_2$ et en utilisant $\theta = (-3, 1, 1)$.
- Deuxième exemple (à droite) avec $\phi_1(x) = 1$, $\phi_2(x) = x_1^2$ et $\phi_3(x) = x_2^2$ et en utilisant $\theta = (-1, 1, 1)$.





INSA UT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Frontière de décision

Définition

A θ fixé, la frontière Γ (en vert) entre la décision pour $y = 0$ et celle pour $y = 1$ est définie comme la ligne de niveau $1/2$ de h :

$$\Gamma = \{x; h(x; \theta) = 0.5\}.$$

Proposition

En remarquant de plus que

$$h(x; \theta) = 0.5 \Leftrightarrow g\Big(\sum_k \theta_k \phi_k(x)\Big) = 0.5 \Leftrightarrow \sum_k \theta_k \phi_k(x) = 0,$$

on en déduit que la frontière de décision est définie implicitement par l'ensemble vérifiant

$$\Gamma = \left\{ x; \sum_k \theta_k \phi_k(x) = 0 \right\}$$

Optimisation du jeu de paramètres θ

Idée: minimiser une fonctionnelle d'erreur dans l'esprit de la méthode des moindres carrés

$$J(\theta) = \sum_{i=1}^{m} \left(h(x^{(i)}; \theta) - y^{(i)} \right)^2$$

Difficultés :

- Les erreurs de classification ne sont pas assez pénalisées!
- cette fonctionnelle n'est à priori **pas convexe** et peut donc avoir **des minimas locaux**.

Modèle entropique

On rappelle que $h(x;\theta)$ est un modèle pour $P(y=1|x)$.

- Si $y=1$, les prévisions $h(x;\theta)$ proche de zéro doivent être fortement pénalisées en utilisant par exemple un poids de la forme

$$-\log(h(x;\theta))$$

- Si $y=0$, les prévisions $h(x;\theta)$ proche de 1 doivent être fortement pénalisées en utilisant par exemple un poids de la forme

$$-\log(1-h(x;\theta))$$

Définition (Fonction de régression logistique)

La fonction de coût de la régression logistique s'écrit

$$J(\theta) = -\sum_{i=1}^{m} \left(y^{(i)} \log(h(x^{(i)};\theta)) + (1-y^{(i)}) \log(1-h(x^{(i)};\theta)) \right)$$

En résumé

- Le choix du jeu de paramètre θ^* optimal et permettant d'obtenir le "meilleur" modèle $h(x;\theta^*)$ est définie comme le minimum de la fonctionnelle J :

$$\theta^* = argmin_\theta\{J(\theta)\}$$

- La classification d'un nouveau vecteur de caractéristiques x s'effectue alors simplement en regardant la valeur de la fonction logistique

$$h(x;\theta^*) = g(\sum_k \theta_k \phi_k(x))$$

Calcul des gradients et formule de rétropropagation

1. Introduction
2. Méthode des moindres carrés linéaires
3. Optimisation et méthode du gradient
4. Régression logistique
   - Introduction
   - Expression du gradient de la fonctionnelle J
   - Exemples numériques
   - Exemples d'exercices
5. Introduction aux réseaux de neurones

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Optimisation et méthode du gradient

La fonctionnelle J étant non linéaire, nous utilisons par la suite une méthode du gradient pour trouver θ^*. Nous avons donc besoin de l'expression du gradient de J

$$J(\theta) = -\sum_{i=1}^{m} y^{(i)} \log(h_i) + (1 - y^{(i)}) \log(1 - h_i)$$

où $h_i(\theta) = g(\sum_k \theta_k \phi_k(x^{(i)}))$.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Expression du gradient de J

Proposition

Le gradient de J vérifie

$$(\nabla J(\theta))_k = \sum_{i=1}^{m} (h_i(\theta) - y^{(i)}) \phi_k(x^{(i)})$$

Preuve: Il suffit de montrer que

$$\nabla J(\theta) = -\sum_{i=1}^{m} \frac{y^{(i)}}{h_i} \nabla h_i - \frac{1 - y^{(i)}}{1 - h_i} \nabla h_i = -\sum_{i=1}^{m} \left(\frac{y^{(i)}}{h_i} - \frac{1 - y^{(i)}}{1 - h_i} \right) \nabla h_i.$$

En remarquant ensuite que la sigmoid vérifie $g'(z) = g(z)(1 - g(z))$, on en déduit que

$$(\nabla h_i(\theta))_k = h_i(\theta)(1 - h_i(\theta)) \phi_k(x^{(i)}).$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Vectorisation

En pratique, nous utiliserons la formulation vectorisée suivante :

$$J(\theta) = X^T(h - y),$$

où

$$X = \begin{pmatrix} \phi_1(x^{(1)}) & \cdots & \phi_d(x^{(1)}) \\ \vdots & & \vdots \\ \phi_1(x^{(m)}) & \cdots & \phi_d(x^{(m)}) \end{pmatrix}, \text{ et } y = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{pmatrix},$$

et

$$h = g(X\theta).$$

Dérivée seconde et convexité

Proposition

La dérivée seconde (Hessienne) de J vérifie

$$(\nabla^2 J(\theta))_{k,m} = \sum_{i=1}^{m} h_i(\theta)(1 - h_i(\theta))\phi_k(x^{(i)})\phi_m(x^{(i)}),$$

En particulier, on en déduit que J est convexe avec

$$z^T \nabla^2 J(\theta) z \geq 0, \text{pour tout} \quad z \in \mathbb{R}^d$$

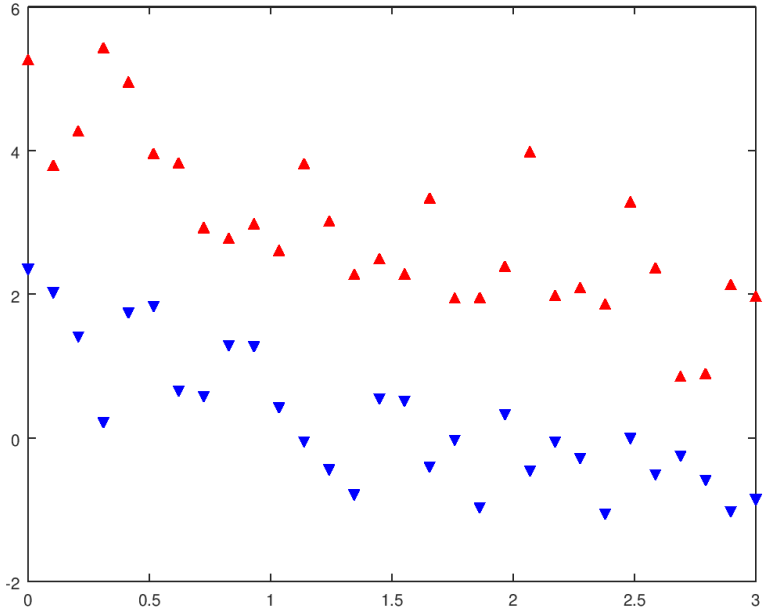
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Exemples numériques

1. Introduction
2. Méthode des moindres carrés linéaires
3. Optimisation et méthode du gradient
4. Régression logistique
   - Introduction
   - Expression du gradient de la fonctionnelle J
   - Exemples numériques
   - Exemples d'exercices
5. Introduction aux réseaux de neurones

Exemple

Régression logistique avec les données



Modèle avec $d = 3$, et $\phi_1(x) = 1$, $\phi_2(x) = x_1$ et $\phi_3(x) = x_2$:

$$\sum_k \theta_k \phi_k(x) = \theta_1 + \theta_2 x_1 + \theta_3 x_2$$

Question: existence d'un jeu de paramètres θ^* optimal ?

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Existence, unicité et Régularisation

Remarque 2 (Problème d'existence de solution)

Si le modèle permet de séparer exactement les deux nuages de points, bien que convexe, la fonctionnelle J n'est pas coercive et n'admet pas de solution bornée.

On peut éviter ce problème, on ajoute en général un terme de régularisation à la fonctionnelle J

$$J(\theta) = -\sum_{i=1}^{m} \left(y^{(i)} \log(h(x^{(i)};\theta)) + (1 - y^{(i)}) \log(1 - h(x^{(i)};\theta)) \right) + \frac{\lambda}{2} \|\theta\|^2,$$

ce qui l'a rend coercive et convexe !
L'expression du gradient J vérifie alors

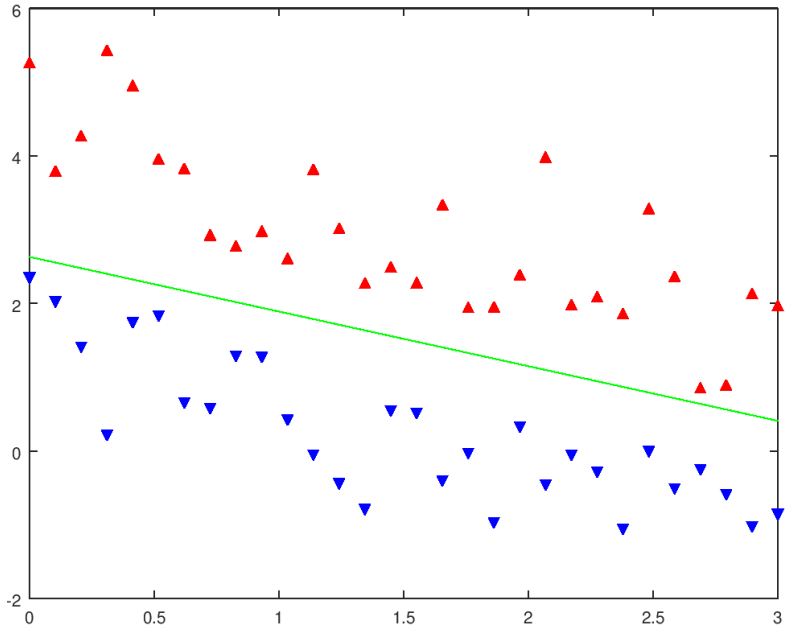
$$\nabla J(\theta) = X^T(h - y) + \lambda\theta$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Exemple

Résultat avec régularisation ($\lambda = 1$):



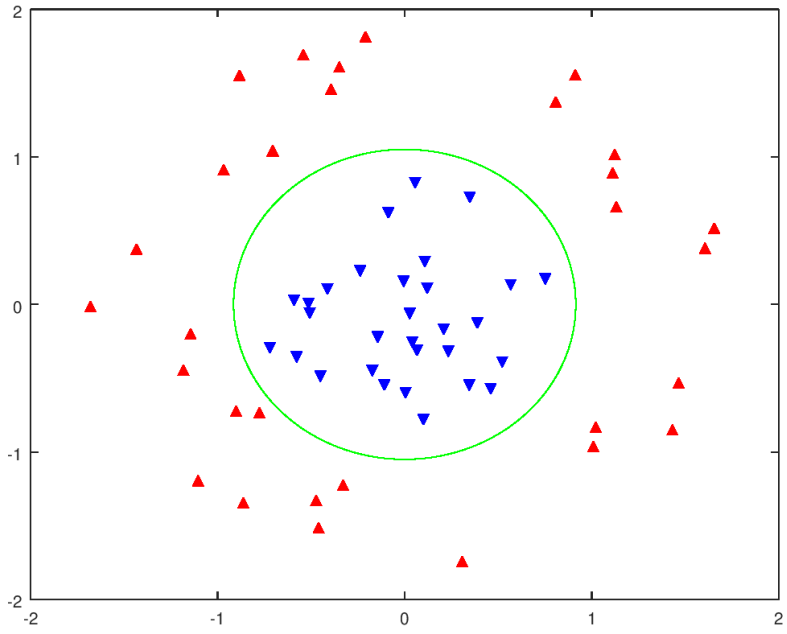
Maintenant $\theta_0 \approx -6.3,\ \theta_1 \approx 1.8,\ \theta_2 \approx 2.4$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Exemple

Exemple avec le modèle $d = 3$ et $\phi_1(x) = 1$, $\phi_2(x) = x_1^2$ et $\phi_3(x) = x_2^2$,

$$\theta^T \phi(x) = \theta_0 + \theta_1 x_1^2 + \theta_2 x_2^2$$



Sans régularisation on obtient $\theta_0 \approx -14.6$, $\theta_1 \approx 17.4$, $\theta_2 \approx 13.2$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Calcul des gradients et formule de rétropropagation

1. Introduction
2. Méthode des moindres carrés linéaires
3. Optimisation et méthode du gradient
4. Régression logistique
   - Introduction
   - Expression du gradient de la fonctionnelle J
   - Exemples numériques
   - Exemples d'exercices
5. Introduction aux réseaux de neurones

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Exercice 13

On considère les modèles suivant:

$$h(x, y; \theta) = g(\theta_0 + \theta_1 x - y) \tag{3}$$

$$h(x, y; \theta) = g((x^2 - \theta_0) + (y^2 - \theta_1) - 1) \tag{4}$$

$$h(x, y; \theta) = g(\theta_0 + \theta_1 x + \theta_2 \sin(\theta_3 x) - y) \tag{5}$$

où x, y et les θ_i sont des réels, et $g(x)$ est la fonction sigmoïde vue en cours.

Pour chaque modèle, tracer un exemple de la frontière de décision définie par $h(x, y; \theta) = 0.5$ dans un plan (x, y) et faire apparaitre les paramètres θ_i (on les choisira non nuls).

Exercice 14

On souhaite utiliser un modèle de régression logistique à partir de données $\{X^i, Y^i\}_{i=1:6}$ composées de 6 points dans $\mathbb{R}^2$ associés à 2 classes différentes. Plus précisément, les points $X^1 = (-0.5, -1.5)$, $X^2 = (1, -1)$ et $X^3 = (1.5, 0.5)$ sont de classe 1 ($Y^1 = Y^2 = Y^3 = 1$) alors que $X^4 = (0.5, 1.5)$, $X^5 = (-1, 1)$ et $X^6 = (-1.5, -0.5)$ sont de classe 0 ($Y^4 = Y^5 = Y^6 = 0$).

- Est-ce qu'un modèle de la forme $h_\theta(X) = g(\theta_1 x_1 + \theta_2 x_2)$ où $X = (x_1, x_2)$ serait efficace pour distinguer ces données ? Justification. On rappelle que g correspond ici à la fonction sigmoïde.
- Par la suite, on propose d'utiliser un modèle linéaire de la forme $h_\theta(X) = g(\theta_1 + \theta_2 x_1 + \theta_3 x_2)$. On propose de plus d'utiliser l'algorithme du gradient en partant de la condition initiale $\theta^{(0)} = (0, 0, 0)^T$. Déterminer les valeurs du vecteur $h^{(0)}$ qui évalue le modèle sur chacun des points X^i en utilisant $\theta^{(0)}$. En déduire le gradient de la fonction de coût d'entropie croisée J en $\theta^{(0)}$.
- Déterminer, en fonction de α, les valeurs du jeu de paramètres $\theta^{(1)}$ obtenu en effectuant une itération de l'algorithme du gradient avec un pas de descente α. En déduire la courbe de démarcation des deux classes obtenue avec ce modèle que vous tracerez sur un graphique.
- Déterminer, en fonction de α, le vecteur $h^{(1)}$ qui évalue le modèle sur chacun des points X^i en utilisant $\theta^{(1)}$. En déduire la limite de $J(\theta^{(1)})$ quand $\alpha \to \infty$. Remarque ?

1 Introduction

2 Méthode des moindres carrés linéaires

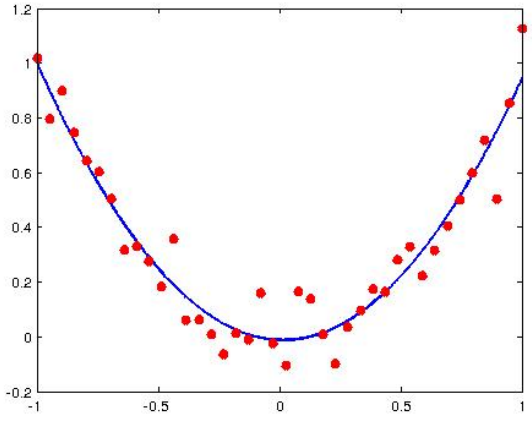
3 Optimisation et méthode du gradient

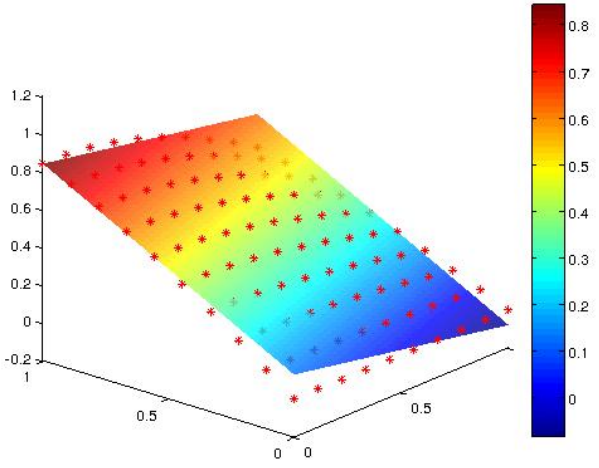
4 Régression logistique

5 Introduction aux réseaux de neurones
- Introduction
- Application pour un problème de classification
- Formule de propagation et rétropropagation
- Expériences numériques
- Exemples d'exercices

Problème d'approximation et modèle

- Un échantillon de données $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i \in \{1 \cdots m\}}$.





- Choix d'un modèle

$$h(x;\theta) = \theta_1 \phi_1(x) + \ldots + \theta_d \phi_d(x)$$

- Optimisation des paramètres θ en minimisant

$$J(\theta) = \sum_{i=1}^{m} \left(h(x^{(i)};\theta) - y^{(i)} \right)^2$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Problème d'approximation et optimisation

- Minimisation d'une fonctionnelle quadratique

$$J(\theta) = \sum_{i=1}^{m} \left(h(x^{(i)}; \theta) - y^{(i)} \right)^2$$

- Equation d'Euler,

$$\nabla J(\theta) = 0 \Rightarrow X^T X \theta = X^T y,$$

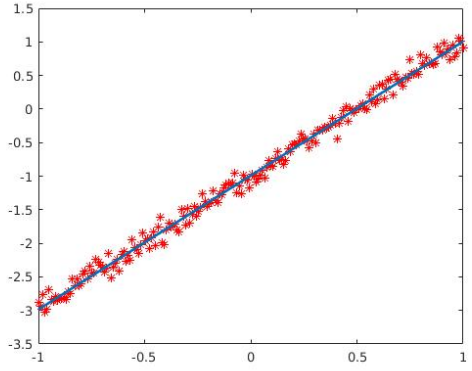
avec

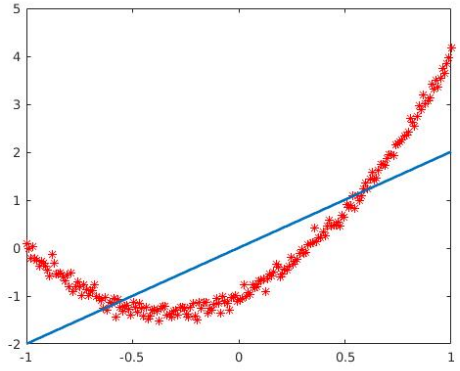
$$X = \begin{pmatrix} \phi_1(x^{(1)}) & \cdots & \phi_d(x^{(1)}) \\ \vdots & & \vdots \\ \phi_1(x^{(m)}) & \cdots & \phi_d(x^{(m)}) \end{pmatrix}, \theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_d \end{pmatrix}, \; y = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

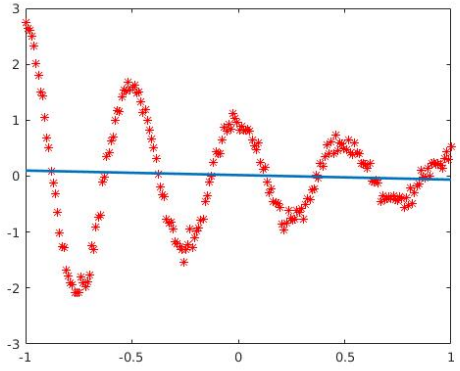
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Problème d'approximation,exemples numériques

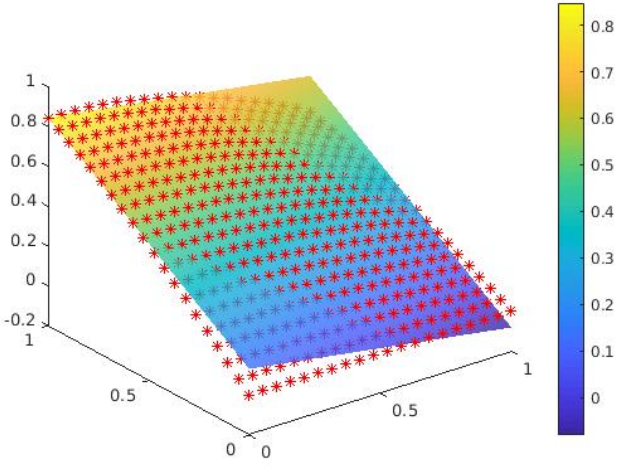
- Cas 1D

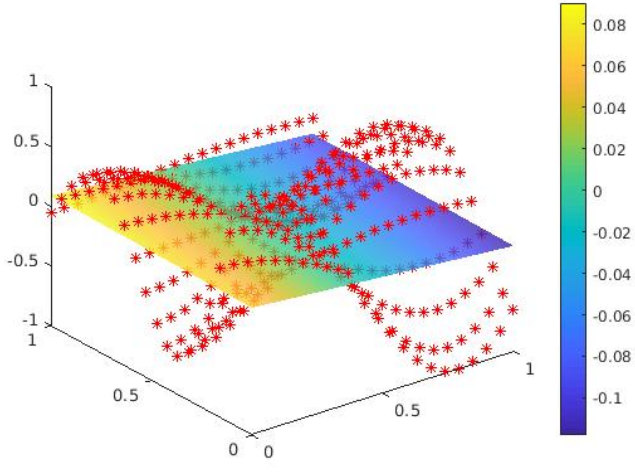






- Cas 2D



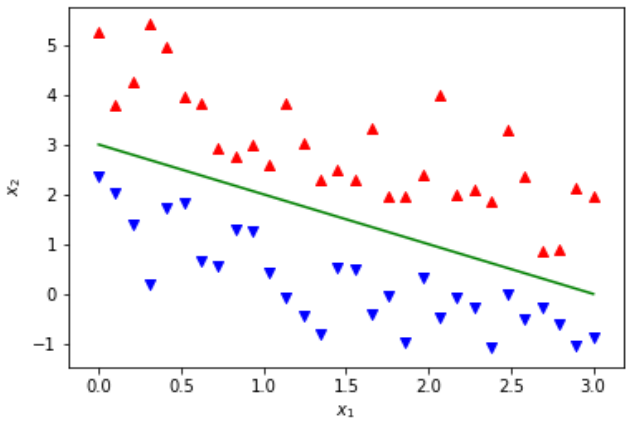


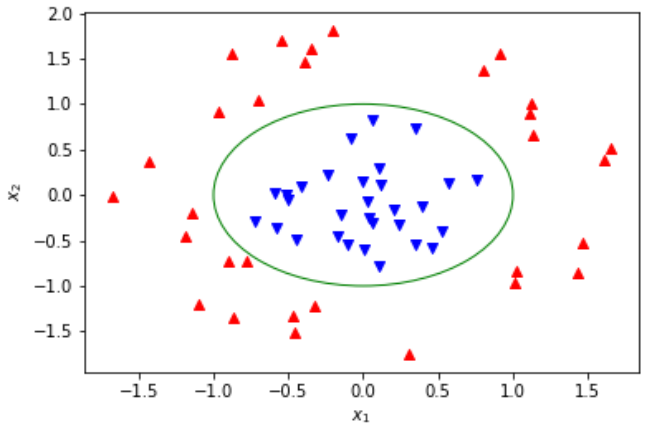
INSA DES SCIENCES APPLIQUÉES LYON

Pole Math

Problème de classification

- Exemples d'applications
  - e-mails: spam (oui/non)
  - transactions en ligne: frauduleuse (oui/non)
  - tumeurs: bénigne/maligne





- Données: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i \in \{1,\cdots,m\}} \subset \mathbb{R}^{n+1}$ où

$x^{(i)} \in \mathbb{R}^n$ vecteur de caractéristiques

$y^{(i)} \in \{0, 1\}$ 0: classe négative

1: classe positive

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Problème de classification, modèle mathématiques

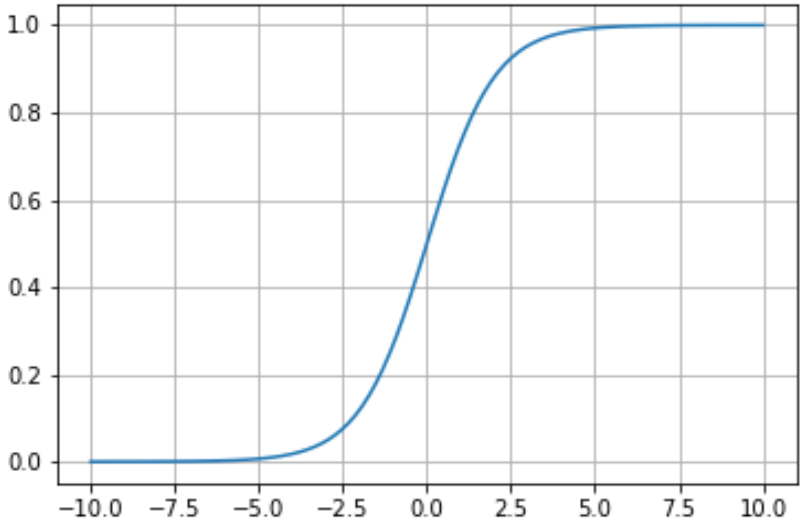
Trouver une approximation de

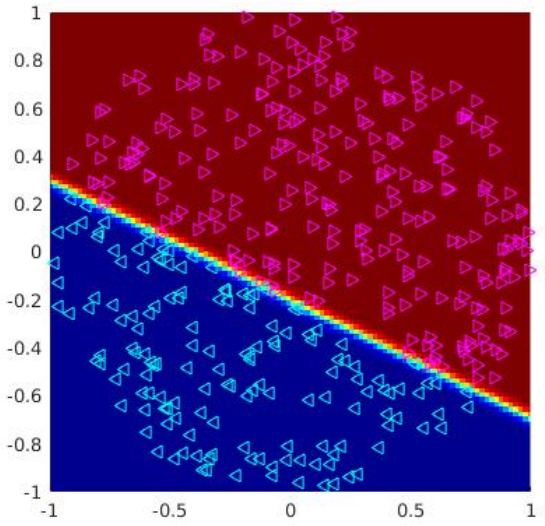
$$P(y = 1|x),$$

avec un modèle de la forme

$$h(x;\theta) = g\left(\sum_{k=1}^{d} \theta_k \phi_k(x)\right)$$

où g est une sigmoïde définie par $g(z) = \frac{1}{1+e^{-z}}$





Problème de classification, régression logistique

Trouver les meilleurs paramètres θ qui minimisent

$$J(\theta) = -\sum_{i=1}^{m} \left(y^{(i)} \log(h(x^{(i)};\theta)) + (1 - y^{(i)}) \log(1 - h(x^{(i)};\theta)) \right)$$

- Si $y^{(i)} = 1$, les prévisions $h(x^{(i)};\theta)$ proche de zéro doivent être fortement pénalisées en utilisant par exemple un poids de la forme

$$-\log(h(x^{(i)};\theta))$$

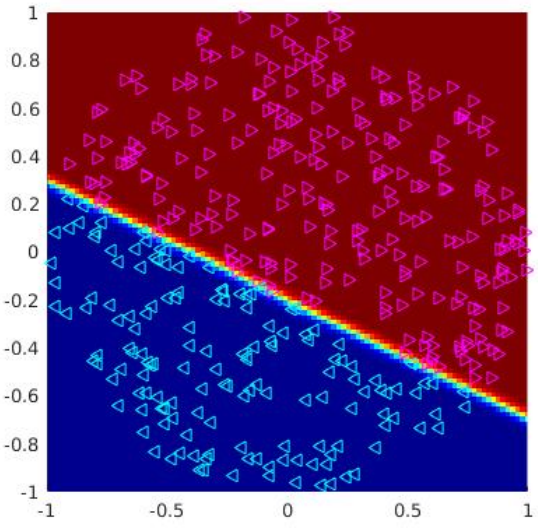
- Si $y^{(i)} = 0$, les prévisions $h(x^{(i)};\theta)$ proche de 1 doivent être fortement pénalisées en utilisant par exemple un poids de la forme

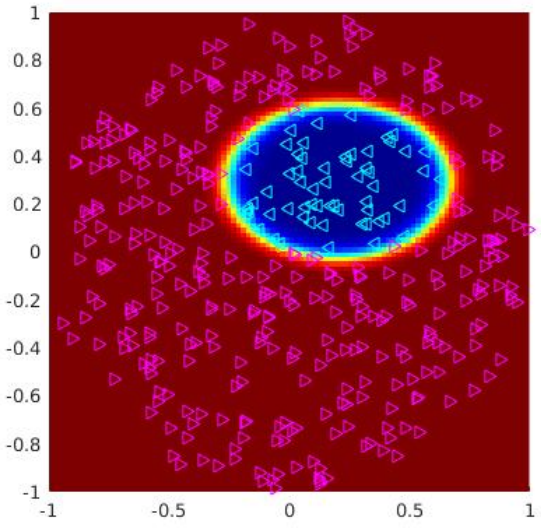
$$-\log(1 - h(x;\theta))$$

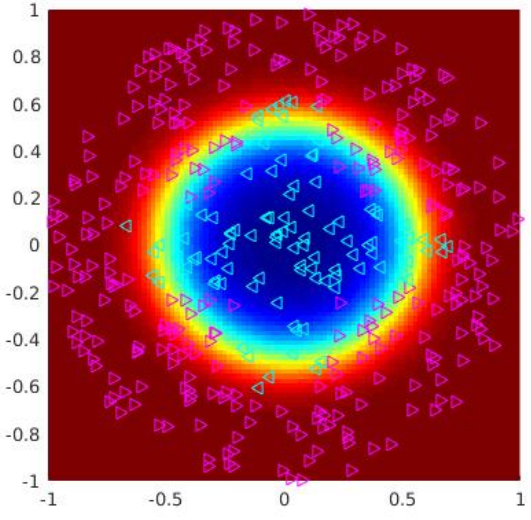
- La fonctionnelle J est alors convexe, donc facile à minimiser avec une approche itérative.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pôle Math

Problème de classification, exemples numériques







INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

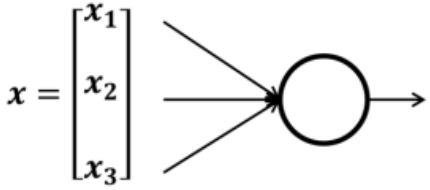
Pole Math

Réseaux de Neurones

- Remplacer le modèle linéaire

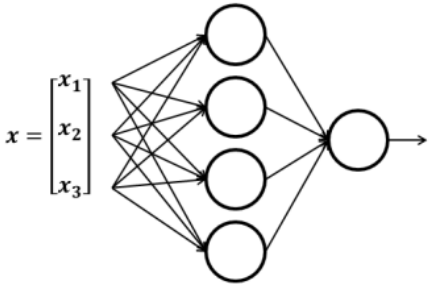
$$\phi(x, \theta) = \theta_1 \phi_1(x) + \ldots + \theta_d \phi_d(x),$$

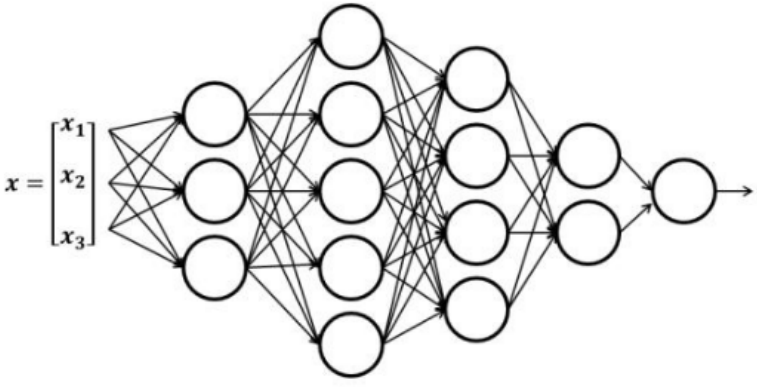
  par un réseau de neurones
- Un Neurone



$$\phi(x) = ReLu(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)$$

  où $g(s) = max(s, 0)$.
- Exemple de réseaux de neurones





NSTITUT NATIONAL
)ES SCIENCES
APPLIQUÉES
.YON

Pole Math

Réseaux de Neurones: Optimisation

- Modèle non linéaire !

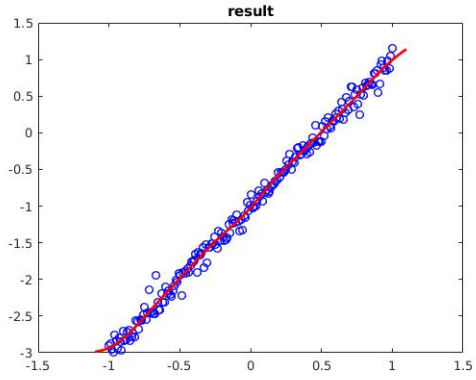
$$\phi(x) = g\left(w_1^{[2]} g\left(W_1^{[1]} x + b_1^{[1]}\right) + w_2^{[2]} g\left(W_2^{[1]} x + b_2^{[1]}\right) + b^{[2]}\right)$$

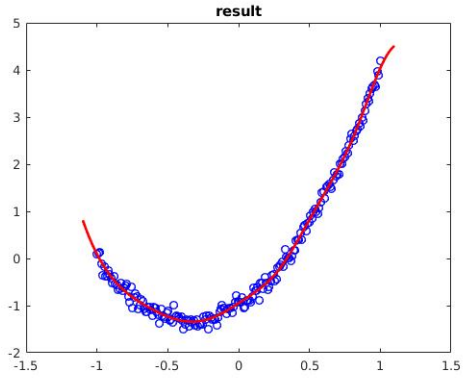
- La fonctionnelle à minimiser est non convexe $\Rightarrow$ éviter les minimums locaux
- Algorithme d'optimisation avec une méthode du gradient
- Accélération
  $\Rightarrow$ inertie + approche stochastique
- Calcul du gradient? couche par couche
- Beaucoup de degrés de liberté mais cela fonctionne en pratique !

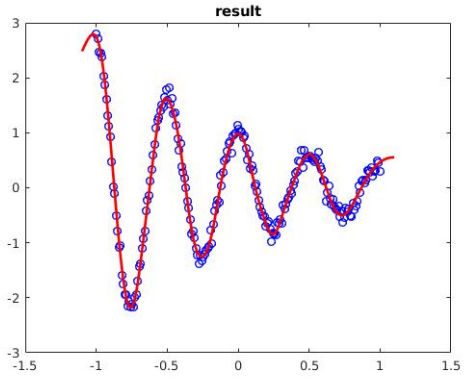
Exemple numérique, cas approximation

Exemples avec un réseau à 4 couches (16,8,4,1)

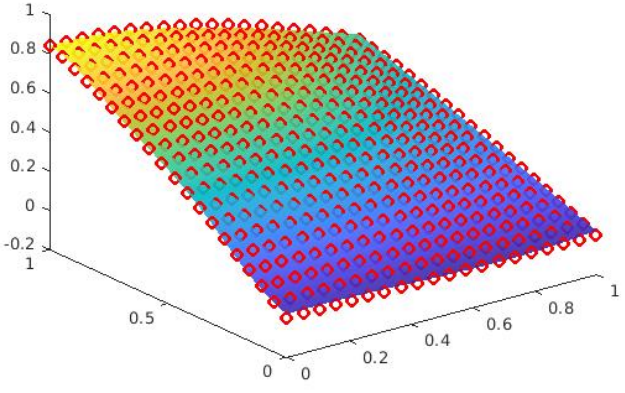
- Cas 1D

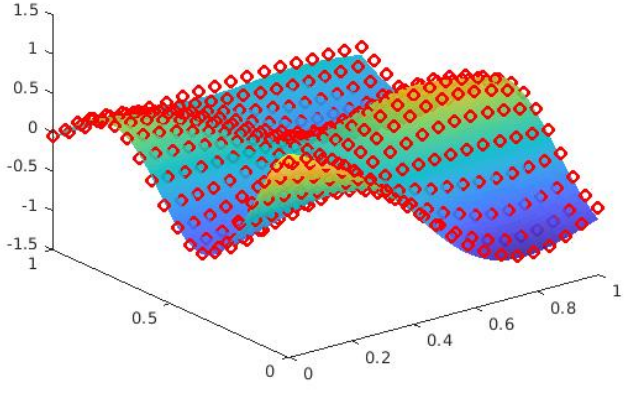






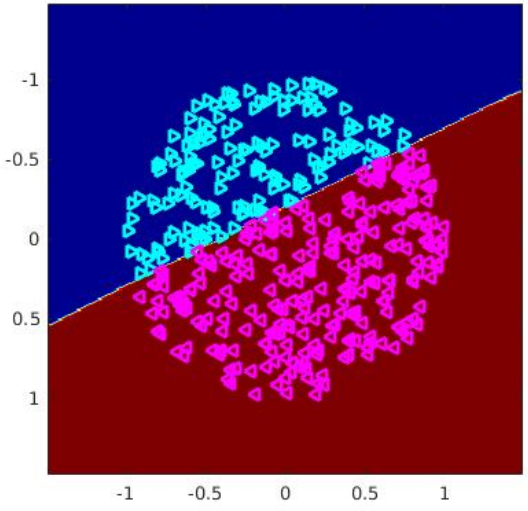
- Cas 2D

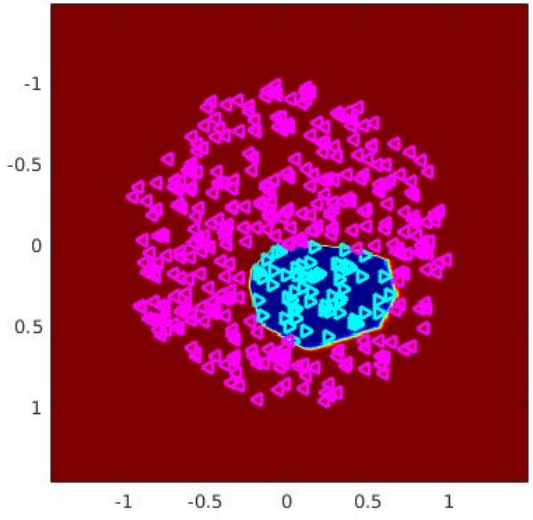


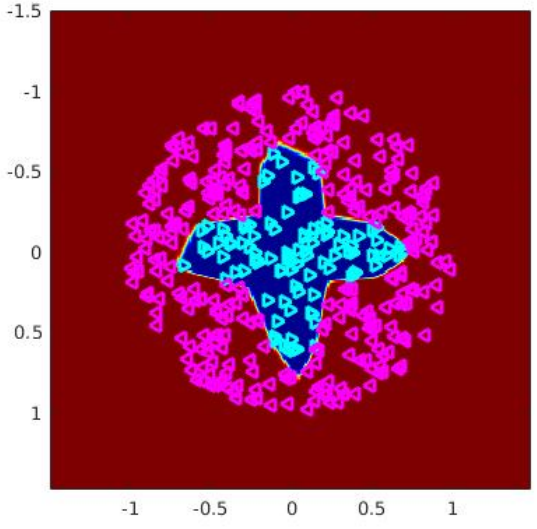


Exemple numérique, cas classification

Exemples avec un réseau à 4 couches (16,8,4,1)



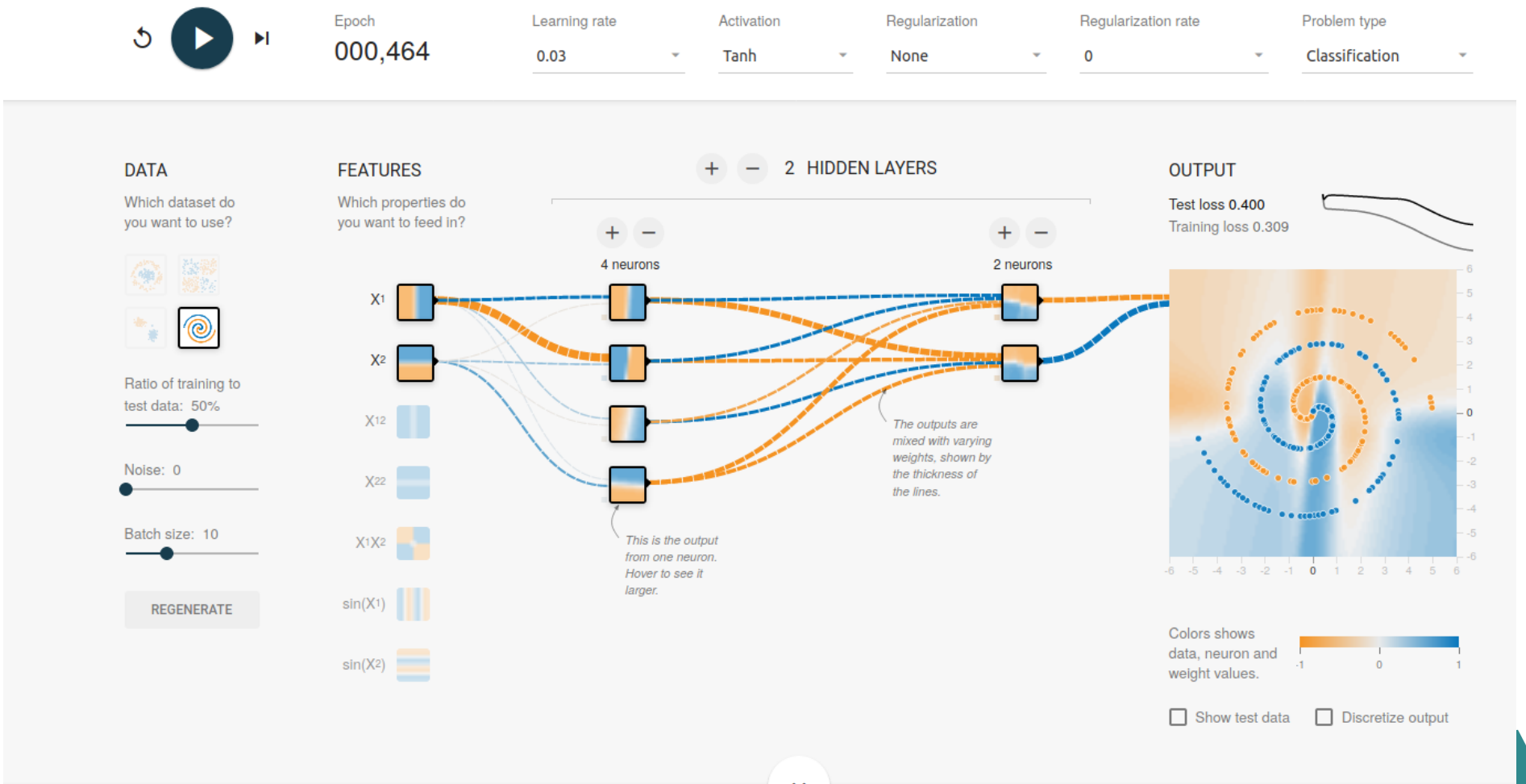




INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

https://playground.tensorflow.org



1 Introduction

2 Méthode des moindres carrés linéaires

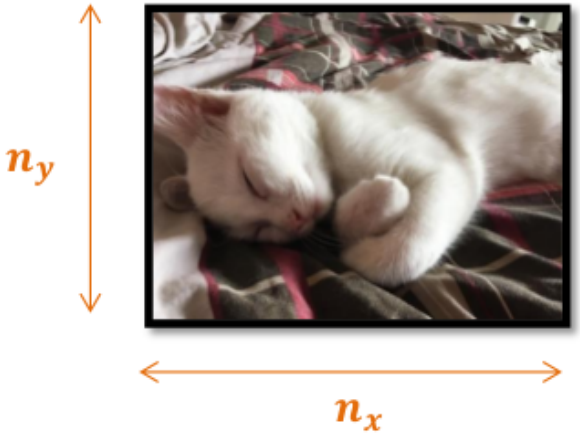
3 Optimisation et méthode du gradient

4 Régression logistique

5 Introduction aux réseaux de neurones
- Introduction
- Application pour un problème de classification
- Formule de propagation et rétropropagation
- Expériences numériques
- Exemples d'exercices

Concepts de base

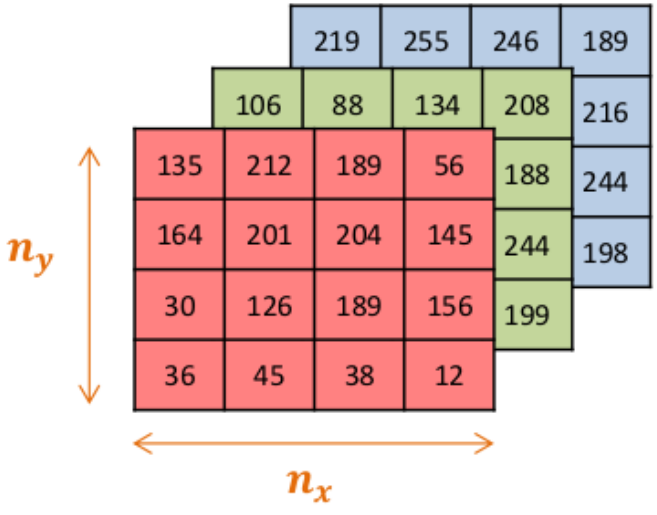
▶ Classification binaire



Présence d'un chat ?

1 (chat)
0 (pas de chat)

y



$$x = \begin{bmatrix} 135 \\ 212 \\ 189 \\ \vdots \\ 198 \end{bmatrix} \in \mathbb{R}^{[n_X \times 1]}$$

avec $n_X = n_x \times n_y \times 3$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Math

Concepts de base

▶ Base de données

Base de données composées de m échantillons

- **Introduction de la matrice de données X**

$$X = \begin{bmatrix} | & & | & & | \\ x^{(1)} & \dots & x^{(i)} & \dots & x^{(m)} \\ | & & | & & | \end{bmatrix} \quad n_X \quad m$$

$$X \in \mathbb{R}^{[n_X \times m]}$$

Concepts de base

▶ Base de données

Base de données composées de m échantillons

- **Introduction du vecteur de classe y**

$$\mathbf{y} = [\boldsymbol{y}^{(1)} \quad \dots \quad \boldsymbol{y}^{(i)} \quad \dots \quad \boldsymbol{y}^{(m)}] \updownarrow 1$$

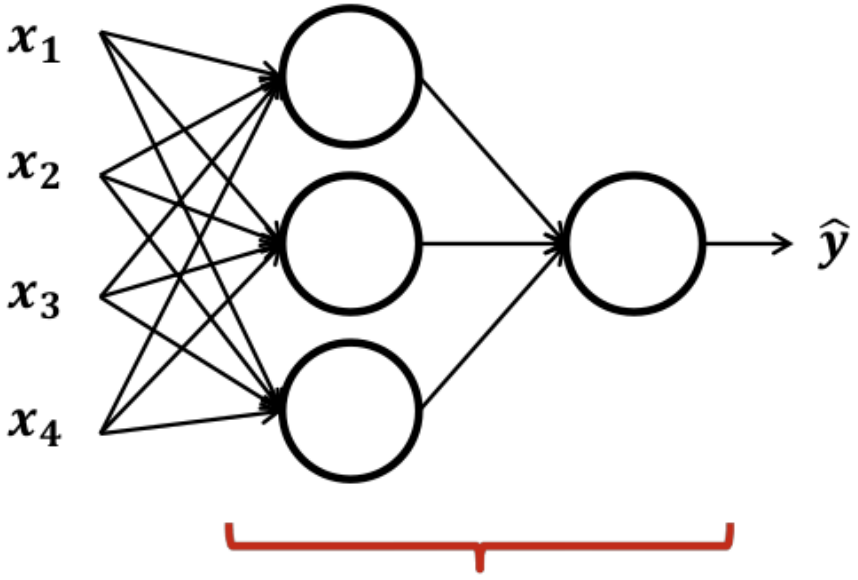
$$\longleftrightarrow$$

$$\boldsymbol{m}$$

$$\boldsymbol{y} \in \mathbb{R}^{[1 \times m]}$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Utilisation d'un réseau de neurones



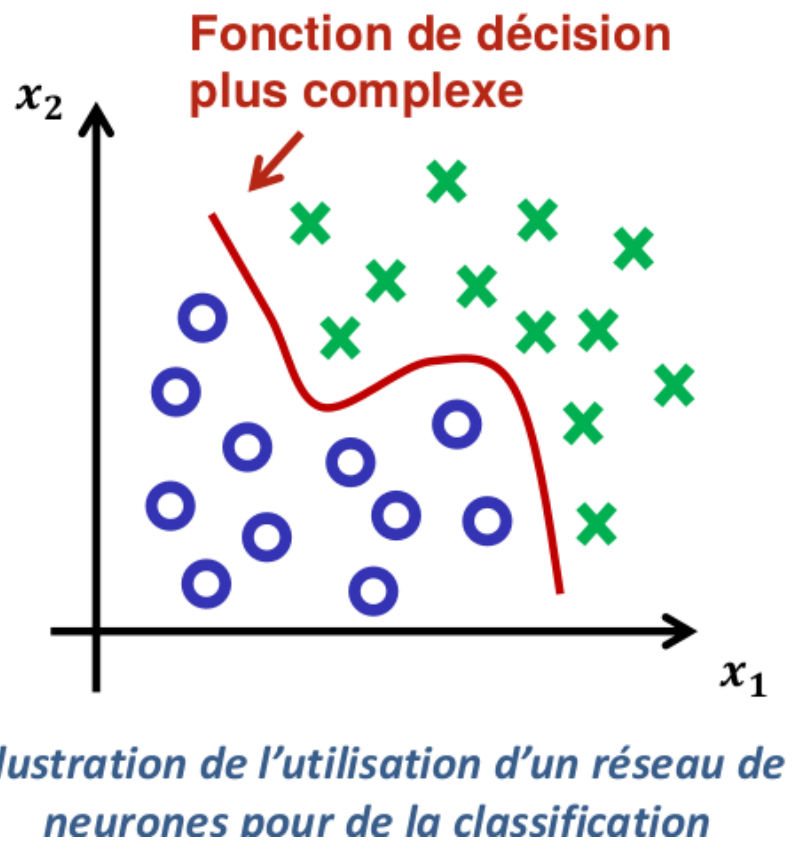


Illustration de l'utilisation d'un réseau de neurones pour de la classification

- La dernière couche utilise une fonction d'activation de type sigmoide

Apprentissage de la fonction de décision

- **Définition d'une fonction d'erreur à minimiser**
  - **Lorsque l'on a $\{x, y\}$ en entrée d'un neurone, on souhaite que sa sortie $\hat{y}$ soit la plus proche de y**
  - **Fonction d'erreur classiquement utilisée**

$$\mathcal{L}(\hat{y}, y) = -[y\, log(\hat{y}) + (1 - y)\, log(1 - \hat{y})]$$

Entropie croisée

Apprentissage de la fonction de décision

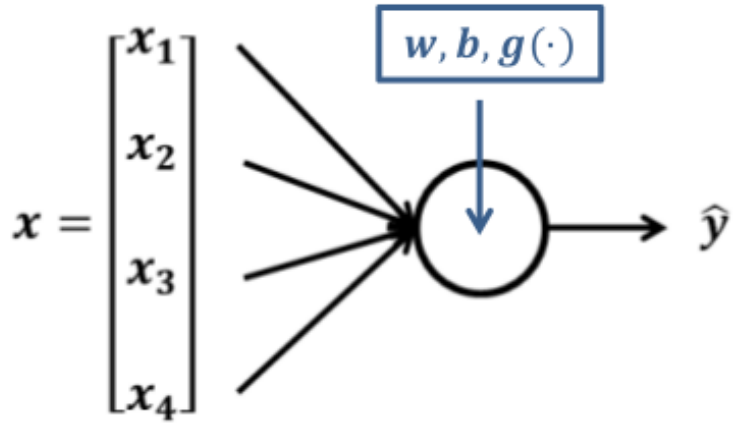
- **Fonction d'erreur à minimiser**
  - **Application à l'ensemble de la base de données**

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

  - **La minimisation de $J(w,b)$ a pour but de trouver les paramètres du neurone $\{w,b\}$ qui permettent de reconnaitre pour chaque échantillon $x^{(i)}$ de la base de données la classe $y^{(i)}$ associée**

Modélisation d'un neurone

▶ Un neurone en résumé



- **Entrée**

  $x \in \mathbb{R}^{[n_X \times 1]}$

- **Comportement**

  Ayant x en entrée, estimation de $\hat{y}$

- **Paramétrage**

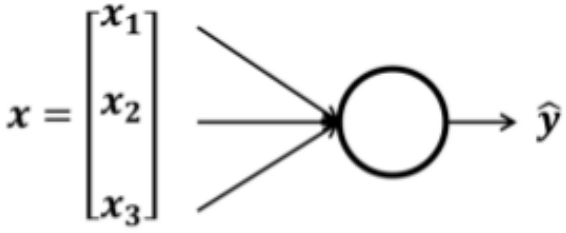
  $w \in \mathbb{R}^{[n_X \times 1]}$, $b \in \mathbb{R}$, **fonction d'activation** $g(\cdot)$

- **Sortie**

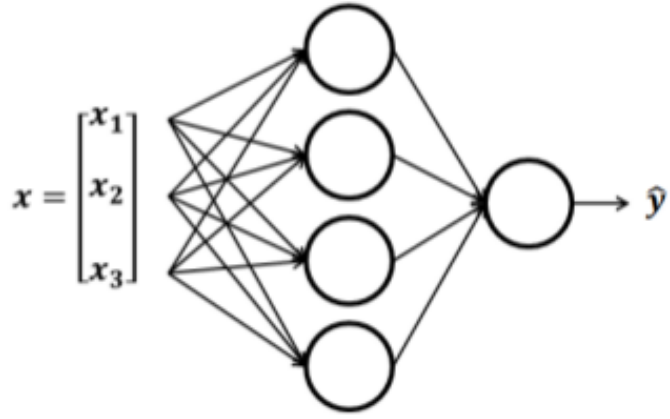
  $\hat{y} = g(w^T x + b) \in \mathbb{R}$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

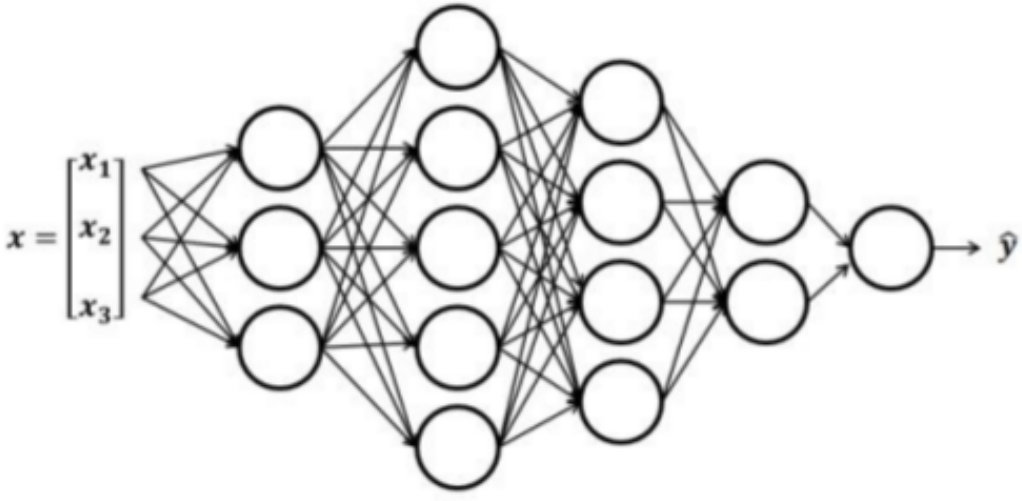
D'un neurone à un réseau



Un neurone = régression logistique



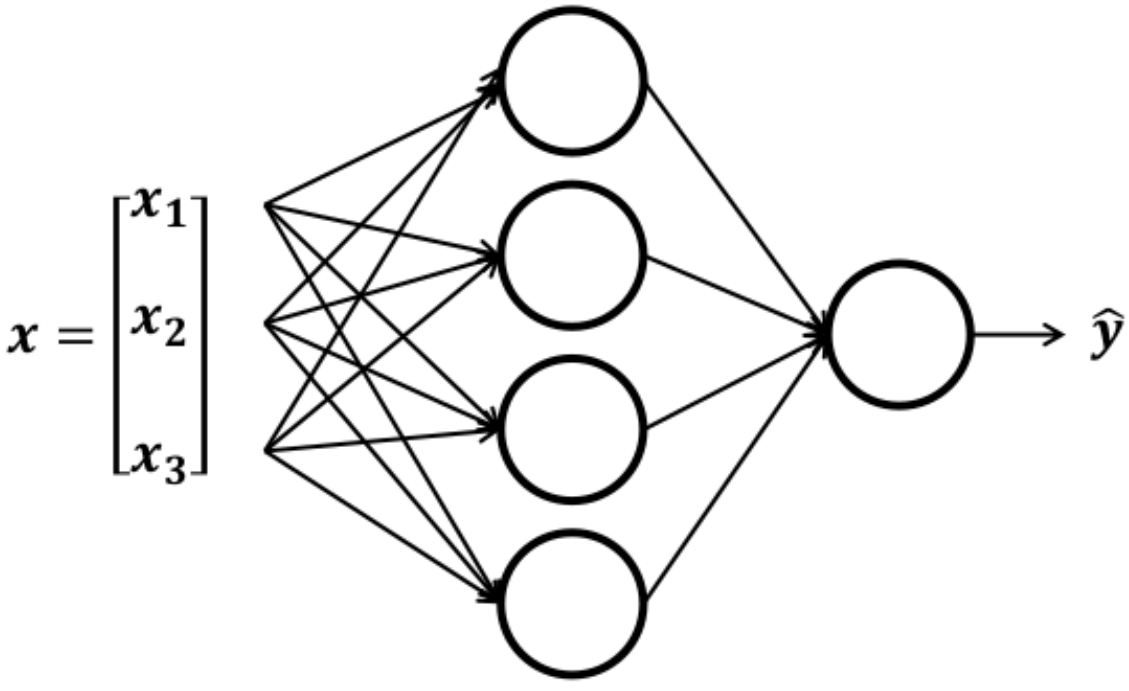
Réseau de neurones à 1 couche cachée



Réseau de neurones profond

Exemple du réseau 3-4-1

- **Les différentes couches d'un réseau de neurones**



Couche d'entrée $l = 0$

Couche cachée $l = 1$

Couche de sortie $l = 2$

L nombre de couches

$L = 2$

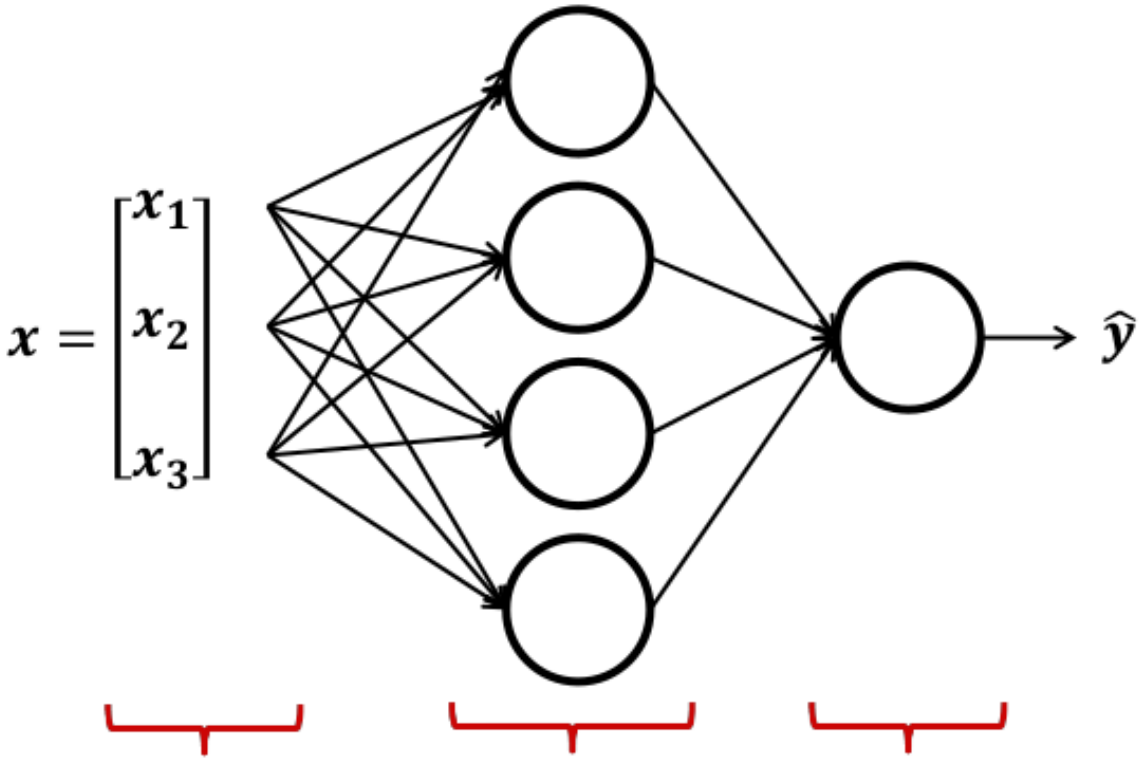
l numéro de couche

$l \in [0, 2]$

INSA DES SCIENCES APPLIQUÉES LYON
Math

Exemple du réseau 3-4-1

Les différentes couches d'un réseau de neurones



Couche d'entrée

$n^{[0]} = 3$

Couche cachée

$n^{[1]} = 4$

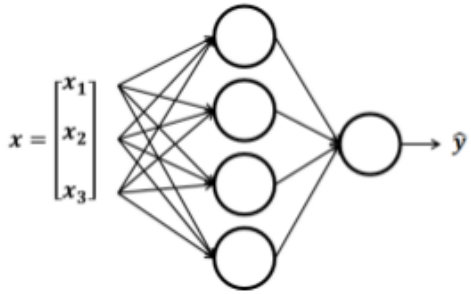
Couche de sortie

$n^{[2]} = 1$

$n^{[l]}$ nombre de neurones pour la couche l

$[l]$ indique que la variable est rattachée à la couche l

Apprentissage de la fonction de décision



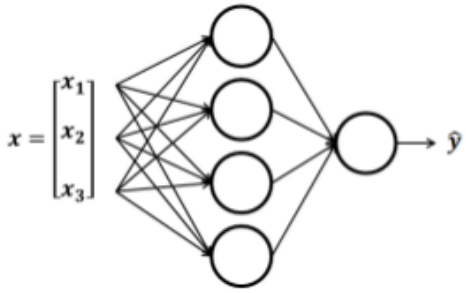
- **Fonction d'erreur à minimiser**
  - **Application à l'ensemble de la base de données**

$$J\left(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}\right) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

  - **La minimisation de $J\left(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}\right)$ a pour but de trouver les paramètres du réseau de neurones $\left\{W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}\right\}$ qui permettent de reconnaitre pour chaque échantillon $x^{(i)}$ de la base de données la classe $y^{(i)}$ associée**

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Apprentissage de la fonction de décision



- **Fonction d'erreur à minimiser**

3) Calcule des dérivées suivantes

$$\frac{\partial J}{\partial b^{[1]}}, \quad \frac{\partial J}{\partial W^{[1]}}, \quad \frac{\partial J}{\partial b^{[2]}}, \quad \frac{\partial J}{\partial W^{[2]}}$$

4) Mise à jour des paramètres du réseau

$$b^{[1]}, \; W^{[1]}, b^{[2]}, \; W^{[2]}$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Formule de rétropropagation et calcul des dérivées partielles

1. Introduction
2. Méthode des moindres carrés linéaires
3. Optimisation et méthode du gradient
4. Régression logistique
5. Introduction aux réseaux de neurones
   - Introduction
   - Application pour un problème de classification
   - Formule de propagation et rétropropagation
   - Expériences numériques
   - Exemples d'exercices

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Cas d'un réseau 1-1-1

Nous nous intéressons tout d'abord au cas simple d'un réseau 1-1-1 à 2 couches qui conduit à une fonction de décision y de la forme

$$y(x) = g^{(2)}(W^{[2]}g^{(1)}(W^{[1]}x + b^{[1]}) + b^{[2]}).$$

On minimise alors la fonctionnelle

$$J(W, b) = \mathcal{L}(y, \hat{y}).$$

avec

$$\mathcal{L}(s, \hat{y}) = -(\hat{y}log(s) + (1 - \hat{y})log(1 - s)),$$

et

$$g^{(2)}(s) = 1/(1 + exp(-s)).$$

Formule de propagation dans le cas d'un réseau 1-1-1

- Première couche :

$$z^{[1]} = W^{[1]}x + b^{[1]} \text{ et } a^{[1]} = g^{(1)}(W^{[1]}x + b^{[1]}) = g^{(1)}(z^{[1]})$$

- Deuxième couche :

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]} \text{ et } a^{[2]} = g^{(2)}(W^{[2]}a^{[1]} + b^{[2]}) = g^{(2)}(z^{[2]}),$$

- Calcul de l'erreur

$$J = \mathcal{L}(a^{[2]}, \hat{y})$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Formule de propagation

Proposition

Avec la condition d'initialisation $a^0 = x$, on en déduit l'expression de

$$J = \mathcal{L}(a^{[L]}, \hat{y})$$

avec les formules de récurrences suivantes

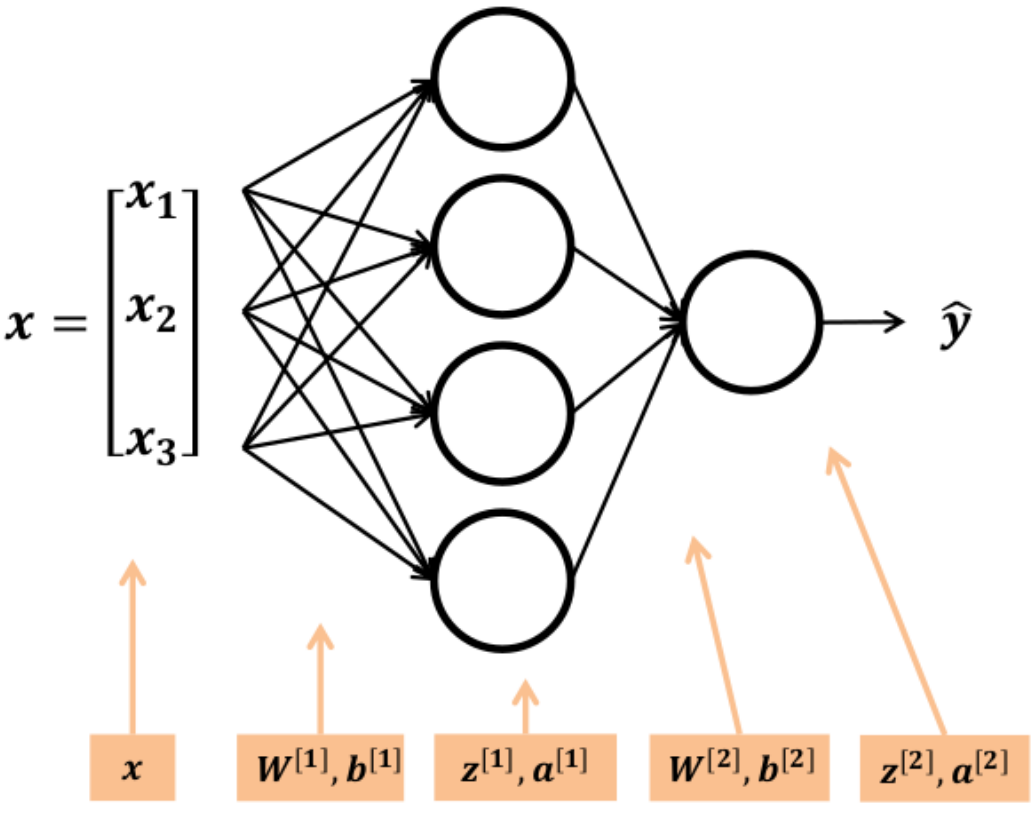
$$\begin{cases} z^{[\ell]} &= W^{[\ell]} a^{[\ell-1]} + b^{[\ell]} \\ a^{[\ell]} &= g^{(\ell)}(z^{\ell}) \end{cases}$$

où L représente le nombre de couche, c'est à dire $L = 2$ dans ce cas précis.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Cas du réseau 3-4-1

En résumé



Ayant x en entrée

$$z^{[1]} = W^{[1]}x + b^{[1]}$$
$$\mathbb{R}^{[4\times1]} \quad \mathbb{R}^{[4\times3]} \quad \mathbb{R}^{[3\times1]} \quad \mathbb{R}^{[4\times1]}$$

$$a^{[1]} = g^{[1]}\left(z^{[1]}\right)$$
$$\mathbb{R}^{[4\times1]} \quad \mathbb{R}^{[4\times1]}$$

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$
$$\mathbb{R}^{[1\times1]} \quad \mathbb{R}^{[1\times4]} \quad \mathbb{R}^{[4\times1]} \quad \mathbb{R}^{[1\times1]}$$

$$\hat{y} = a^{[2]} = g^{[2]}\left(z^{[2]}\right)$$
$$\mathbb{R}^{[1\times1]} \quad \mathbb{R}^{[1\times1]} \quad \mathbb{R}^{[1\times1]}$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Formule de rétropropagation dans le cas d'un réseau 1-1-1

Cas $\ell = 2$

Avec $J = \mathcal{L}(a^{[2]}, \hat{y})$, $a^{[2]} = g^{(2)}(z^{[2]})$ et $z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$, on en déduit que

$$\begin{cases} \partial_{b^{[2]}} J &= \left[\partial_s \mathcal{L}(a^{[2]}, \hat{y})\right] g^{(2)'}(z^{[2]}) = dz^{[2]} \\ \partial_{W^{[2]}} J &= \left[\partial_s \mathcal{L}(a^{[2]}, \hat{y})\right] g^{(2)'}(z^{[2]}) a^{[1]} = dz^{[2]} a^{[1]} \end{cases}$$

avec

$$da^{[2]} = \partial_s \mathcal{L}(a^{[2]}, \hat{y}), \text{ et } dz^{[\ell]} = da^{[\ell]} g^{(\ell)'}(z^{[\ell]}).$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Formule de rétropropagation dans le cas d'un réseau 1-1-1

Cas $\ell = 1$

Avec $J = \mathcal{L}(a^{[2]}, \hat{y})$, $a^{[2]} = g^{(2)}(z^{[2]})$, $z^{[2]} = g^{(2)}(W^{[2]}a^{[1]} + b^{[2]})$, $a^{[1]} = g^{(1)}(z^{[1]})$, $a^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$ et $a^{[0]} = x$,
on en déduit que

$$\begin{cases} \partial_{b^{[1]}} J &= \left[\partial_s \mathcal{L}(a^{[2]}, \hat{y})\right] g^{(2)'}(z^{[2]}) W^{[2]} g^{(1)'}(z^{[1]}) = dz^{[1]} \\ \partial_{W^{[1]}} J &= \left[\partial_s \mathcal{L}(a^{[2]}, \hat{y})\right] g^{(2)'}(z^{[2]}) W^{[2]} g^{(1)'}(z^{[1]}) a^{[0]} = dz^{[1]} a^{[0]} \end{cases}$$

avec

$$da^{[\ell-1]} = da^{[\ell]} W^{[\ell]}, \text{ et } dz^{[\ell]} = da^{[\ell]} g^{(\ell)'}(z^{[\ell]}).$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Formule de rétropropagation

Proposition

Avec la condition d'initialisation

$$da^{[L]} = \partial_s \mathcal{L}(a^{[L]}, \hat{y}) = (-\hat{y}/a^{[L]} + (1-\hat{y})/(1-a^{[L]})),$$

on en déduit les expressions des dérivées de J par rapport aux paramètres $b^{(\ell)}$ et $W^{(\ell)}$ à l'aide des formules de récurrences suivantes

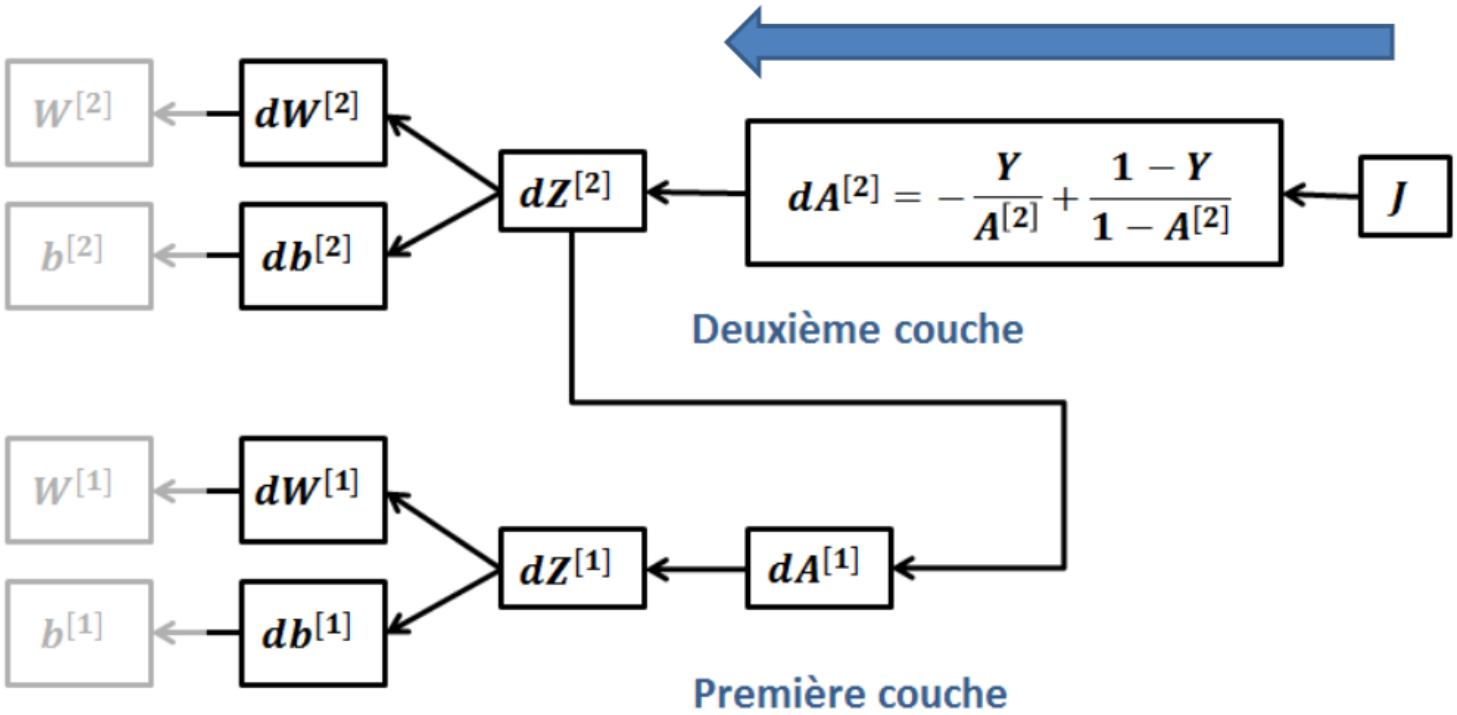
$$\begin{cases} \partial_{b^{[\ell]}} J &= dz^{[\ell]}, \\ \partial_{W^{[\ell]}} J &= dz^{[\ell]} a^{[\ell-1]}, \\ da^{[\ell]} &= da^{[\ell]} W^{[\ell]}, \\ dz^{[\ell]} &= da^{[\ell]} g^{(\ell)'}(z^{[\ell]}), \end{cases}$$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Exemple du réseau 3-4-1

Propagation arrière

- 1ère étape: calcul des $db^{[l]}$ et $dW^{[l]}$ pour $l \in [1, 2]$



INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Calcul des gradients et formule de rétropropagation

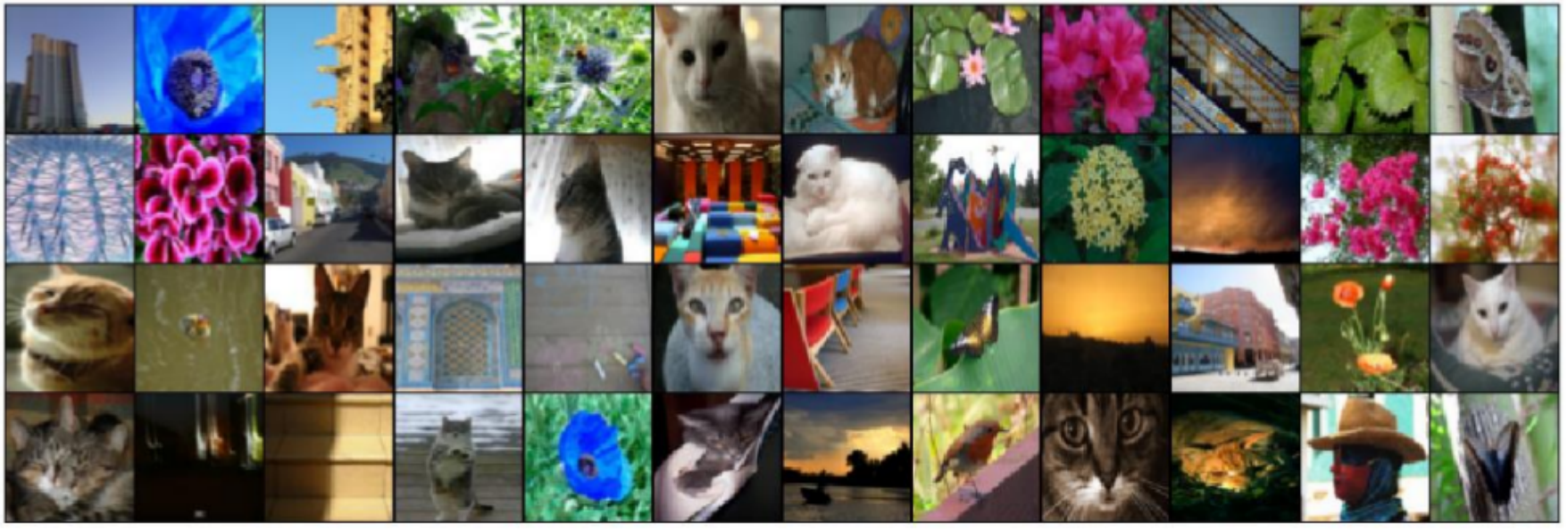
1. Introduction
2. Méthode des moindres carrés linéaires
3. Optimisation et méthode du gradient
4. Régression logistique
5. Introduction aux réseaux de neurones
   - Introduction
   - Application pour un problème de classification
   - Formule de propagation et rétropropagation
   - Expériences numériques
   - Exemples d'exercices

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Cas d'un réseau à 1 neurone

- **Classification pour la reconnaissance de chat dans une image**



Base de données d'entrainement (209 images)

Cas d'un réseau à 1 neurone

- **Classification pour la reconnaissance de chat dans une image**

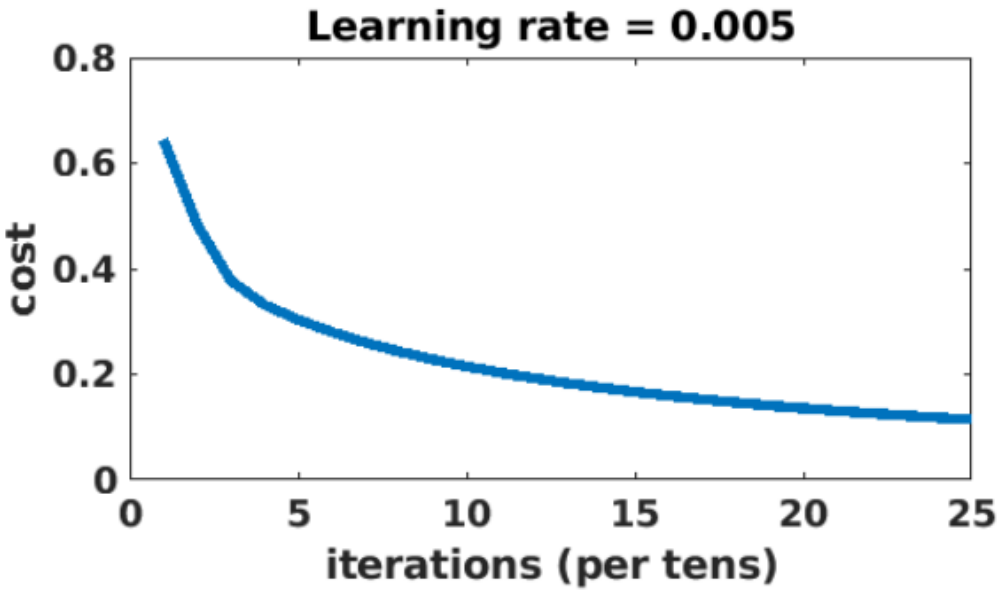
Base de données de test (50 images)

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Cas d'un réseau à 1 neurone

- **Classification pour la reconnaissance de chat dans une image**



Précision: 68 %

Quand même !

Evolution de la fonction d'énergie lors de la minimisation par descente de gradient

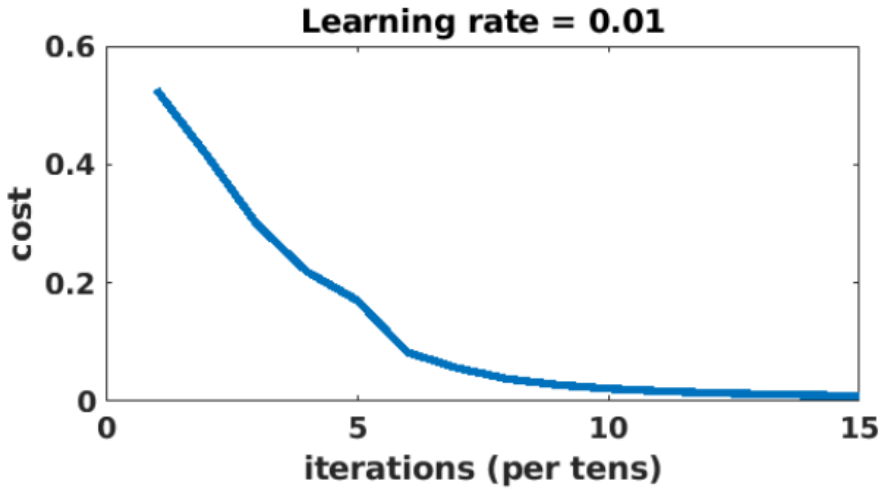
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Cas d'un réseau de neurones profond

Résultats

- **Classification pour la reconnaissance de chat dans une image**



Evolution de la fonction d'énergie lors de la minimisation par descente de gradient

Précision: 78 %

Quand même !

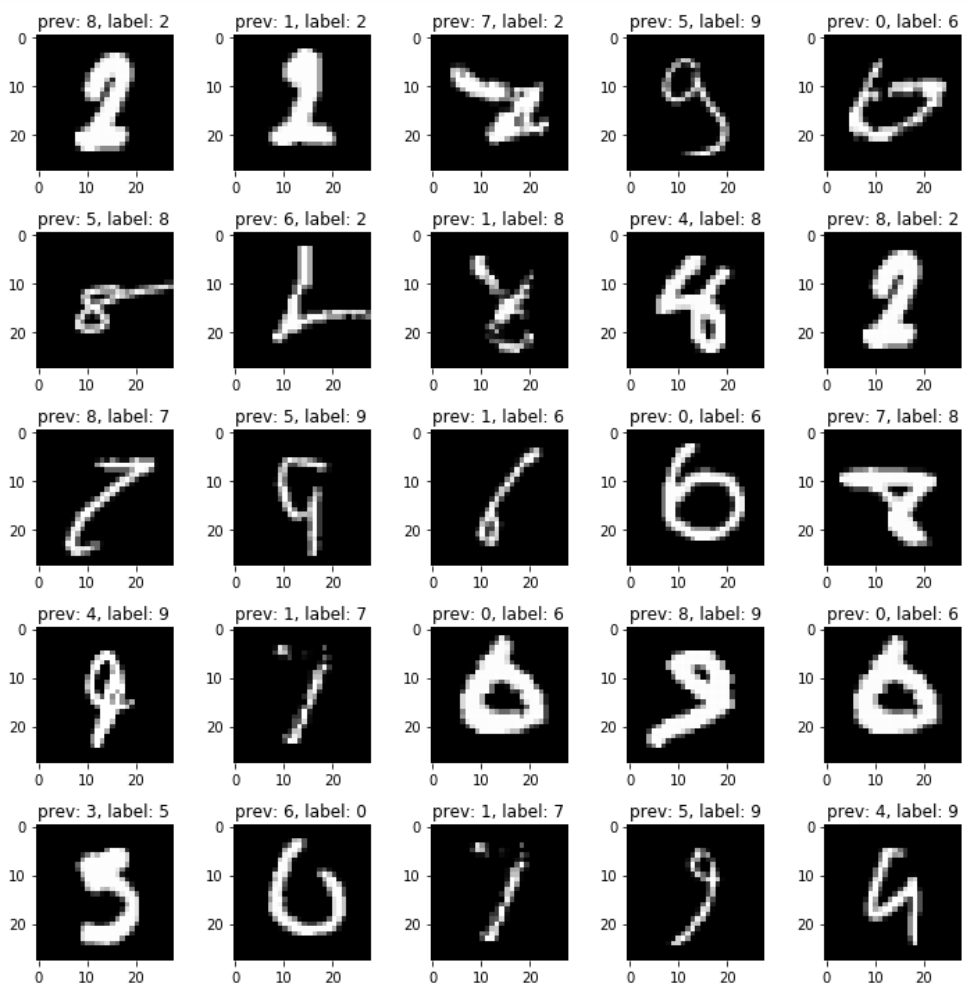
Pourquoi pas plus ?

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Exemple de réseaux de neurones CNN avec Keras sur Python

99,38 % de réussite!



INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Calcul des gradients et formule de rétropropagation

1. Introduction
2. Méthode des moindres carrés linéaires
3. Optimisation et méthode du gradient
4. Régression logistique
5. Introduction aux réseaux de neurones
   - Introduction
   - Application pour un problème de classification
   - Formule de propagation et rétropropagation
   - Expériences numériques
   - Exemples d'exercices

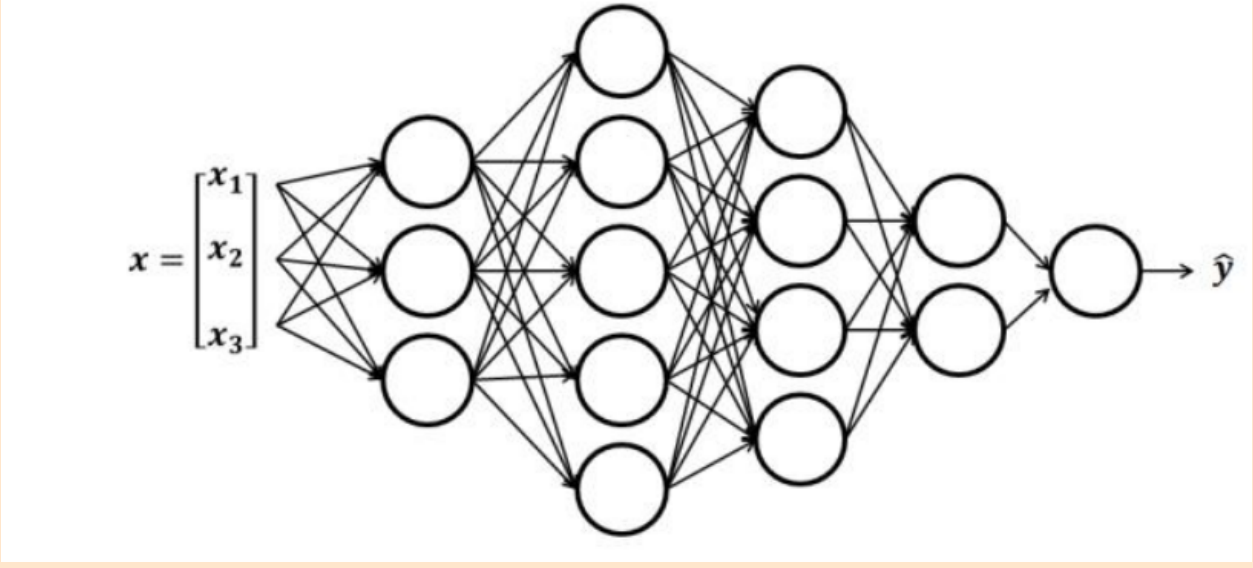
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Pole Math

Réseau de neurones et paramètres

Exercice 15

Déterminer le nombre de paramètres du réseau suivant :



Approximation d'une fonction triangle avec un réseau 2-1

Exercice 16

En 1D, expliciter la fonction triangle entre $[-1, 1]$:

$$f(x) = \begin{cases} x + 1 & \text{si } x \in [-1, 0] \\ 1 - x & \text{si } x \in [0, 1] \\ 0 & \text{sinon} \end{cases}$$

sur un réseau de Neurones à 2 couches avec respectivement 2 et 1 neurones sur les différentes couches et en utilisant l'activation non linéaire ReLu. Cette décomposition est-elle unique ?

Exercice 17

Les données d'entrée correspondent maintenant à des éléments $x \in \mathbb{R}$. On s'intéresse à un réseau à 1 couche cachée contenant 3 neurones et dont la fonction d'activation correspond à la fonction ReLu, et on met un neurone d'activation linéaire en sortie.
On suppose que $W^{[1]} = [2, 1, 2]^T$ et $b^{[1]} = [2, 0, -2]^T$ et $W^{[2]} = [1, -1, -1]$ et $b^{[2]} = 0$. Tracer la réponse de ce réseau pour $x \in [-2, 3]$.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Pole Math

Exercice 18

L'objectif de cet exercice est de mettre au point une architecture de réseau de neurones permettant de classifier des points dans le plan selon un damier.

- On s'intéresse tout d'abord à l'architecture d'un réseau $h_\theta^{(0,0)} : \mathbb{R}^2 \to [0,1]$ qui permet de classifier le plan $\mathbb{R}^2$ en deux classes: $y = 1$ si $x \in \mathcal{C}^{(0,0)}$ et $y = 0$ sinon, où

$$\mathcal{C}^{(0,0)} = \left\{ x = (x_1, x_2) \in \mathbb{R}^2 \text{ tel que } -1 \leq x_1 \leq 1 \text{ et } -1 \leq x_2 \leq 1 \right\}.$$

  - En suivant la convention du cours, quelles valeurs doit prendre le réseau $h_\theta^{(0,0)}$ en fonction de x pour que la classe 1 soit exactement associée au carré $\mathcal{C}^{(0,0)}$.
  - Soit φ la fonction définie pour tous $x \in \mathbb{R}^2$ par $\varphi^{(0,0)}(x) = 1 - |\frac{x_1+x_2}{2}| - |\frac{x_1-x_2}{2}|$. Vérifier que φ est bien positive pour $x \in \mathcal{C}^{(0,0)}$ et négative si $x \neq \mathcal{C}^{(0,0)}$.
  - En déduire des paramètres du réseau $h_\theta^{(0,0)}$ qui permet une classification exacte du problème et dont la structure comporte deux couches, 2 neurones sur la première couche (avec la fonction $\sigma : z \mapsto |z|$ comme fonction d'activation) et une couche de sortie (avec la sigmoïde comme fonction d'activation).

- On souhaite maintenant construire un deuxième réseau $h_\theta^{(z_1,z_2)}$ classifiant $y = 1$ si $x \in \mathcal{C}^{(z_1,z_2)}$ et $y = 0$ sinon. Le carré $\mathcal{C}^{(z_1,z_2)}$ est défini par

$$\mathcal{C}^{(z_1,z_2)} = \left\{ x \in \mathbb{R}^2 \text{ tel que } z_1 - 1 \leq x_1 \leq z_1 + 1 \text{ et } z_2 - 1 \leq x_2 \leq z_2 + 1 \right\}$$

  En reprenant la méthodologie précédente, proposer une structure de réseau $h_\theta^{(z_1,z_2)}$ et les valeurs de ses paramètres qui permettent une classification exacte du problème considéré.

- On souhaite maintenant construire un réseau $h_\theta^{(0,0),(2,2)}$ qui permet la classification des deux classes noir/blanc d'un damier à 4 cases dans $[-1,3]^2$. Proposer une achitecture de réseau à 3 couches qui permet une telle classification en spécifiant ses paramètres θ qui permettent une classification exacte de ce damier à $2 \times 2 = 4$ cases.