Cours 1: Rappel d'apprentissage machine

Emmanuel Franck[*],

September 16-20, 2024
Master CMSI, M2, Strasbourg

[*] MACARON project-team, Université de Strasbourg, CNRS, Inria, IRMA, France

Outline

Introduction à l’apprentissage machine

Rappels

Apprentissage supervisé

Apprentissage supervisé profond

Introduction à l'apprentissage machine

Notion d'apprentissage machine

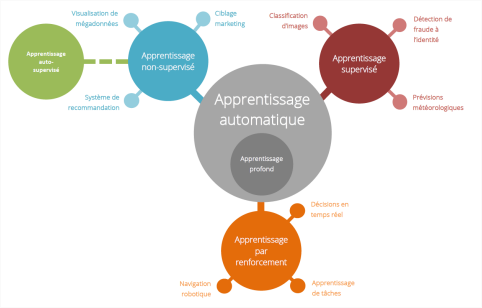
Apprentissage machine

Il s'agit de construire des modèles prédictifs ou explicatifs à partir de données (données numériques, textes, images, série temporelles etc).

- On sépare le processus en deux phases:
  - La phase d'entrainement ou l'on construit le modèle à partir de données d'entrainement.
  - La phase d'inférence ou l'on utilise le modèle sur de nouvelles données.
- Il s'agit d'un domain qui est une branche de l'IA et des statistiques.

Type d'apprentissage

- Trois principaux types d'apprentissage:
  - **Apprentissage supervisé**: il s'agit de construire des approximations de fonctions à partir d'exemple de données d'entrée et de sortie. **Applications**: classification et segmentation d'image, traduction de textes, ...
  - **Apprentissage non-supervisé**: il s'agit de construire des modèles expliquant des données à partir d'exemples. **Applications**: génération de texte, de molécules, compression.
  - **Apprentissage par renforcement**: il s'agit d'apprendre à contrôler un problème temporel par essai-erreur, ... **Applications**: robotique, jeux, ...



Modèles paramétriques

- En apprentissage on cherche à approcher des **fonctions** qui transforment nos données, des lois de probabilité qui expliquent les données etc.
- Par exemple, on peut chercher à déterminer une **fonction inconnue** de la forme:

$$\mathbf{y} = f(\mathbf{x}), \quad \mathbf{x} \in V \subset \mathbb{R}^d, \quad \mathbf{y} \in W \subset \mathbb{R}^p$$

- **Objectif**: trouver $f_h \in H$ une approximation de f avec H un espace de fonction.
- **Difficulté**: on cherche un objet de dimension infini.

Modèles paramétriques

On choisit une fonction paramétrique connue $f_\theta(x)$ avec des paramètres inconnus θ. Le problème d'approximation devient :

$$\text{Trouver} \quad \theta, \text{ tel que } \|f_\theta - f\|_H \leqslant \epsilon$$

Enjeux

- Enjeux:
  - Quels modèles paramétriques ?
  - Comment on détermine θ à partir des données ?
  - Sous quelles conditions un modèle appris est valide ?
  - Comment détermine-t-on la validité d'un modèle ?

Rappels

Moindres carrés

Définition: moindres carrés

Soit des matrices $A \in \mathcal{M}_{n,k}(\mathbb{R})$, $b \in \mathcal{M}_{n,1}(\mathbb{R})$ et $\theta \in \mathbb{R}^k$. Le problème des moindres carrés associé s'écrit

$$\min_{\theta \in \mathbb{R}^k} \mathcal{J}(\theta) = \min_{\theta \in \mathbb{R}^k} \| A\theta - b \|^2$$

Lemme: solution des moindres carrés

La fonction $\mathcal{J}(\theta)$ est strictement convexe et coercive. Les solutions du problème satisfont l'équation normale:

$$A^t A\theta = A^t b$$

Si la matrice A est de rang maximal, il existe une unique solution du problème:

$$\theta = \underbrace{(A^t A)^{-1} A^t}_{A^+} b$$

La matrice A^+ est appelée le pseudo-inverse de Moore-Penrose.

Idées de démonstration

- Principaux points:
  - Coercivité. Le point clé est de démontrer que

$$||A\theta||_2 \underset{||\theta||_2 \to +\infty}{\longrightarrow} +\infty.$$

  - Convexité sur $\mathbb{R}^k$. On utilise le coté quadratique de la fonctionnelle.
  - Stricte convexité sur $\chi = \mathbb{R}^k \backslash \text{Ker}(A)$.
  - On a donc l'existence et l'unicité de minimiseurs sur χ.
  - Soit $\mathcal{S}$ sur $\mathbb{R}^k$ on montre que

$$\mathcal{S} = \{\theta_2 + \omega, \text{ avec } \omega \in \text{Ker}(A)\}.$$

    avec θ_2 le minimiseur sur χ.
  - Calcul du gradient:

$$\text{On cherche: } \nabla_\theta \mathcal{J}(\theta) = 0.$$

$$\begin{aligned}\mathcal{J}(\theta + h) &= \langle A(\theta + h) - b, A(\theta + h) - b\rangle \\ &= \mathcal{J}(\theta) + 2\langle A\theta - b, Ah\rangle + o(h) \\ &= \mathcal{J}(\theta) + \langle 2A^t(A\theta - b), h\rangle + o(h)\end{aligned}$$

$$\text{donc } \nabla \mathcal{J}_r(x)h = \lim_{\varepsilon \to 0} \frac{\mathcal{J}_r(\theta + \varepsilon h) - \mathcal{J}(\theta)}{\varepsilon} = \langle 2A^t(A\theta - b), h\rangle$$

$$\text{ce qui donne par le Th de Riez: } \nabla_\theta \mathcal{J}(\theta) = 0 \Longleftrightarrow 2A^t(A\theta - b) = 0$$

  - $\text{Ker}(A^tA) = \text{Ker}(A)$ d'ou la condition de rang maximal.

Définition: loi de probabilité

Soit X une v.a.r. sur l'espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$. La loi de probabilité de la variable aléatoire X est la mesure de probabilité, notée $\mathbb{P}_X$, définie sur l'espace mesurable $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ par

$$\mathbb{P}_X(B) = \mathbb{P}\big(X^{-1}(B)\big) = \mathbb{P}(X \in B)$$

pour tout borélien $B \in \mathcal{B}(\mathbb{R})$. Autrement dit, $\mathbb{P}_X$ est la mesure image de $\mathbb{P}$ par X. Une v.a.r. discrète (resp. à densité) X est associée à une loi de probabilité dite discrète (resp. absolument continue).

- Probabilité:

$$\forall A \in \mathcal{B}(\mathbb{R}), \qquad \mathbb{P}(X \in A) = \int_{\mathbb{R}} f(x)\chi_A(x)\, dx.$$

- avec χ la fonction indicatrice.

Rappel probabilité II

Définition: loi de probabilité

Soit X, une v.a.r. définie sur $(\Omega, \mathcal{A}, \mathbb{P})$ et à valeurs dans $\overline{\mathbb{R}}$.

- Si X est $\mathbb{P}$-intégrable ou à valeurs dans $[0, +\infty]$ $\mathbb{P}$-presque sûrement, **l'espérance** de X, notée $\mathbb{E}(X)$ est définie par

$$\mathbb{E}(X) = \int_{\Omega} X(\omega) d\mathbb{P}(\omega).$$

- Si X est telle que $\mathbb{E}(|X|^2) < +\infty$ (on dit alors que X admet un moment d'ordre 2), la variance de X est définie par $\text{Var}(X) = \mathbb{E}\left[(X - \mathbb{E}(X))^2\right] = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$.

Définition: loi des grand nombres

Soit une variable aléatoire X de moyenne p tel que $\mathbb{E}[\mid X \mid] < \infty$. Soit $(X_n)_{n \in \mathbb{N}^*}$ une suite de v.a.r. i.i.d. (autrement dit, indépendantes et identiquement distribuées ou encore indépendantes et de même loi). Alors la **moyenne empirique** converge vers l'espérance $\bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i \xrightarrow{p.s.} \mathbb{E}[X]$

Rappel probabilité III: Estimateur

- La notion d'estimateur statistique est utile pour calculer numériquement les paramètres d'une loi de probabilité à partir d'échantillons. Considérons une loi de probabilité $\mathcal{L}_\theta$ dépendant d'un paramètre θ (nombre ou vecteur).
- En estimation on cherche a déterminer le paramètre à partir de réalisations d'échantillons de cette loi.

Estimateur

- *n*-échantillon d'une loi $\mathcal{L}_\theta$ est une famille $(X_k)_{1 \leqslant k \leqslant n}$ de variables aléatoires indépendantes et de même loi de probabilité $\mathcal{L}_\theta$.
- Une observation $(x_1, \ldots, x_n)$ est une réalisation du n-échantillon $(X_1, \ldots, X_n)$.
- Un estimateur de θ est une variable aléatoire $\hat{\theta}_n$ de la forme $F_n(X_1, ..., X_n)$ où $F_n : \mathbb{R}^n \to \mathbb{R}$.

Rappel probabilité IV: vraissemblance

- Estimateur classique : l'estimateur du maximum de vraissemblance.
- La fonction de vraisemblance quantifie la probabilité que les observations proviennent effectivement d'un échantillon (théorique) de la loi μ_θ.
- Considérons un n-échantillon $(X_1, \dots, X_n)$ de loi μ_θ discrète donnée par la probabilité P_θ. Soit $(x_1, \dots, x_n)$, une observation de cet échantillon. On montre aisément que

$$\mathbb{P}\left(\bigcap_{i=1}^{n}\{X_i = x_i\}\right) = \prod_{i=1}^{n} P_\theta(x_i)$$

en utilisant que les variables $(X_i)_{1 \leqslant i \leqslant n}$ sont indépendantes.

Principe

La méthode du maximum de vraisemblance revient à considérer qu'il est raisonnable d'estimer le paramètre θ comme celui maximisant la probabilité de réalisation de $(x_1, \dots, x_n)$ définie ci-dessus.

Rappel probabilité V: vraisemblance continue

- Dans le cas où la loi μ_θ est donnée par une densité continue f_θ, on se donne $\varepsilon > 0$ et on cherche à estimer $\mathbb{P}\big(\bigcap_{i=1}^n \{|X_i - x_i| \leqslant \varepsilon\}\big)$.
- On a
$$\mathbb{P}\left(\bigcap_{i=1}^n \{|X_i - x_i| \leqslant \varepsilon\}\right) = \prod_{i=1}^n \int_{x_i-\varepsilon}^{x_i+\varepsilon} f_\theta(x)\, dx$$
- Notons que $\int_{x_i-\varepsilon}^{x_i+\varepsilon} f_\theta(x)\, dx \sim 2\varepsilon f_\theta(x_i)$ lorsque $\varepsilon \to 0$ et par conséquent,
$$\mathbb{P}\left(\bigcap_{i=1}^n \{|X_i - x_i| \leqslant \varepsilon\}\right) \sim 2^n \varepsilon^n \prod_{i=1}^n f_\theta(x_i) \quad \text{lorsque } \varepsilon \to 0.$$
- Pour s'extraire de la dépendance en ε, on considère qu'il est raisonnable de chercher à maximiser $\lim_{\varepsilon\to 0} \mathbb{P}\big(\bigcap_{i=1}^n \{|X_i - x_i| \leqslant \varepsilon\}\big)/(2\varepsilon)^n$.

Rappel probabilité VI: fonction vraissemblance

Fonction vraissemblance

- Soit X, une v.a.r. discrète ou à densité sur l'espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$ de loi μ_θ définie par la probabilité P_θ dans le cas discret et par une densité f_θ sinon, dont on veut estimer le paramètre θ appartenant à un ensemble Θ.
- Soit

$$f : X(\Omega) \times \Theta \ni (x; \theta) \mapsto \begin{cases} f_\theta(x), \text{dans le cas continu} \\ P_\theta(X = x), \text{dans le cas discret} \end{cases}$$

- Soit $(x_1, \ldots, x_n)$, un n-échantillon de loi μ_θ.
- La fonction de vraisemblance $\mathcal{L}$ associée à cet échantillon est définie par

$$\mathcal{L} : X(\Omega)^n \times \Theta \ni (x_1, \ldots, x_n; \theta) \mapsto \prod_{i=1}^{n} f(x_i; \theta)$$

Max de vraissemblance

- Lorsque la fonction de vraisemblance admet un unique maximum, l'estimateur du maximum de vraisemblance $\hat{\theta}_n$ est défini par

$$\hat{\theta}_n = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \mathcal{L}(x_1, \ldots x_n; \theta)$$

- On définit également l'estimateur du maximum de log vraisemblance $\hat{\theta}_n$ par

$$\hat{\theta}_n = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \mathcal{L}_{\log}(x_1, \ldots x_n; \theta) \quad \text{avec } \mathcal{L}_{\log}(x_1, \ldots x_n; \theta) = \sum_{i=1}^{n} \log f(x_i; \theta)$$

- Exemple: On considère un échantillon $(x_1, \ldots, x_n) \in \{0,1\}^n$ (loi binomiale $\mathcal{B}(1,p)$). On estime $\theta = p \in [0,1]$.
- La fonction de vraisemblance s'écrit alors :

$$\mathcal{L}(x_1, \ldots x_n; \theta) = \prod_{i=1}^{n} \mathbb{P}_\theta(\{X_i = x_i\}) = \theta^{\sum_i x_i}(1-\theta)^{n-\sum_i x_i}, \quad \mathbb{P}_\theta(\{X_i = 1\}) = p = \theta,$$

- La solution θ^* est donnée par

$$\frac{\partial \mathcal{L}}{\partial \theta}(x_1, \ldots, x_n; \theta) = 0 \Leftrightarrow \theta^* = \frac{1}{n}\sum_i x_i.$$

Ainsi, on retrouve l'estimateur de l'espérance de la loi des grands nombres.

Rappel probabilité VIII: Théorie

- Théorie de convergence : https://sciml.gitlabpages.inria.fr/scimllectures/chapAS_sec2.html

Divergence de Kullback-Leibler

Soit P et Q deux distributions de probabilité. La divergence de Kullback Leibler est définie par

$$D_{KL}(P \parallel Q) = \sum_{i=1} P(i) \log\left(\frac{P(i)}{Q(i)}\right), \quad D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

Limite de l'estimateur de vraissemblance

Soit $\mathcal{X} = (x_1, ..., x_n)$ un échantillon issu de variables aléatoires de loi $p_{\theta_{ref}}(x)$. Soit $\hat{\theta}$ l'estimateur du maximum de vraisemblance. Lorsque $n \to \infty$ est solution du problème de minimisation

$$\hat{\theta} = \text{argmin}_\theta \, D_{KL}(p_{\theta_{ref}} \parallel p_\theta)$$

Démonstration

- Il s'agira une preuve très formelle, notamment sur le passage à la limite
- On considère l'estimateur du maximum de vraisemblance. Soit:

$$\begin{aligned}\hat{\theta}_n &= \text{argmax}_\theta \, \mathcal{L}_{\log}(x_1, \dots x_n, \theta) \\ &= \text{argmax}_\theta \sum_{i=1}^{n} \log f_\theta(x_i) = \text{argmax}_\theta \sum_{i=1}^{n} \Big(\log f_\theta(x_i) - \log f_{\theta_{\text{ref}}}(x_i) \Big) \\ &= \text{argmax}_\theta \sum_{i=1}^{n} \log \left(\frac{f_\theta(x_i)}{f_{\theta_{ref}}(x_i)} \right) = \text{argmin}_\theta \sum_{i=1}^{n} \log \left(\frac{f_{\theta_{ref}}(x_i)}{f_\theta(x_i)} \right) \\ &= \text{argmin}_\theta \, \frac{1}{n} \sum_{i=1}^{n} \log \left(\frac{f_{\theta_{ref}}(x_i)}{f_\theta(x_i)} \right) = \text{argmin}_\theta \, \frac{1}{n} \sum_{i=1}^{n} g_\theta(x_i)\end{aligned}$$

  avec $g_\theta(x) = \log \left(\frac{f_{\theta_{ref}}(x)}{f_\theta(x)} \right)$.
- On utilise ici le passage à la limite entre l'espérance empirique d'une fonction de variable aléatoire.

$$\lim_{n \to \infty} \hat{\theta}_n = \text{argmin}_\theta \, \mathbb{E}[g_\theta(x)]$$

- Ensuite, puisque x est distribué selon $f_{\theta_{ref}}$ on a $\text{argmin}_\theta \, \mathbb{E}[g_\theta(x)] = \text{argmin}_\theta \int_x f_{\theta_{ref}}(x) g_\theta(x) dx$ ce qui donne la définition de la divergence KL

$$\text{argmin}_\theta \, \mathbb{E}[g_\theta(x)] = \text{argmin}_\theta \int_x f_{\theta_{ref}}(x) \log \left(\frac{f_{\theta_{ref}}(x)}{f_\theta(x)} \right) dx$$

Apprentissage supervisé

Principe

On connait des données d'entrées et de sorties et on cherche un modèle reliant les entrées et sorties. On suppose qu'il existe une relation:

$$y = f(x)$$

On a des données $(x_1, ..., x_n) \in X$ et $(y_1, ..., y_n) \in Y$. On suppose que

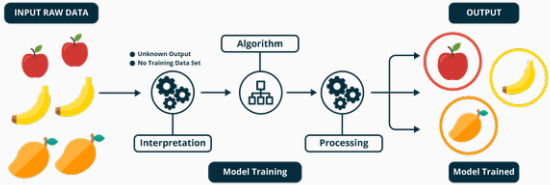
$$y_i = f(x_i) + \epsilon_i, \quad i \in \{1, .., n\}$$

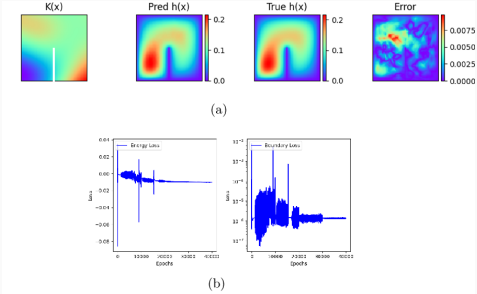
avec ϵ_i un bruit aléatoire.

- Régression: on est dans un cas X et Y sont des intervalles continue. On cherche une fonction numérique.
- Classification: $Y = \{0, 1, ..., q\}$. On cherche à classer les données en q catégories.

Exemples

- Exemples de classification et régression





: Flow in heterogeneous porous media: (a) The predicted $h(\boldsymbol{x})$ for a given conductivity field,

Première formulation

Principe

On propose une première formulation du problème de régression. On cherche donc a construire f_θ tel que $\| f - f_\theta \| < \epsilon$. Pour cela propose de minimiser le risque empirique:

$$\min_\theta \mathcal{J}(\theta) = \min_\theta \sum_{i=1}^{N} \| y_i - f_\theta(x_i) \|$$

- On cherche la meilleure fonction possible au moins sur les données.
- Si on suppose que les données sont des échantillons d'une loi de probabilité $dP(x, y)$ en pratique on cherche a déterminer:

$$\min_\theta \int \| y - f_\theta(x) \| \, dP(x, y)$$

- **Généralisation**: On minimise sur un nombre limité de données avec un bruit. Comment on peut espérer capturer la solution du problème continu ?

Formulation probabiliste

- On peut réécrire la dépendance entre les données sous la forme:

$$p_\theta(y \mid x) = \mathcal{N}(y \mid f(x), \sigma^2)$$

- On cherche donc a chercher une loi de probabilité conditionnelle entre x et y.
- La forme Gaussienne choisit revient à supposer $y = f(x) + \epsilon$ avec ϵ suivant $\mathcal{N}(0, \sigma^2)$.

Principe

On se ramène donc au problème d'estimation de paramètres θ de la loi

$$p_\theta(y \mid x) = \mathcal{N}(y \mid f_\theta(x), \sigma^2)$$

Modèle linéaire

On choisit comme modèle un modèle linéaire:

$$f_\theta(x) = (\omega, x) + b$$

avec les paramètres $\omega \in \mathbb{R}^d$, $b \in \mathbb{R}$.

Formulation probabiliste et vraissemblance

- On voit donc qu'on veut donc construire les paramètres du modèle $\theta = (\omega, b) \in \mathbb{R}^{d+1}$ de façon a maximiser la vraisemblance de $p_\theta(y \mid x)$ sur l'échantillon.

Modèle linéaire par maximum de vraisemblance

Les paramètres $\theta = (\omega, b)$ sont solutions du problème de moindre carré:

$$\min_\theta \parallel b_y - A_x\theta \parallel_2^2$$

avec $A_x \in \mathcal{M}_{n,d+1}(\mathbb{R})$ et $b_y \in \mathcal{M}_{n,1}(\mathbb{R})$ par

$$A_x = \begin{pmatrix} 1 & x_1^1 & \cdots & x_1^d \\ 1 & x_2^1 & \cdots & x_2^d \\ \vdots & & & \vdots \\ 1 & x_{n-1}^1 & \cdots & x_{n-1}^d \\ 1 & x_n^1 & \cdots & x_n^d \end{pmatrix}, \quad b_y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}$$

Pour un nouveau point x la prédiction est donné par $f(x) = \langle \theta, x \rangle$.

Formulation duale

- Comme ont le sait la solution est donné par l'équation normale

$$A^t A\theta = A^t b$$

- Supposons maintenant que notre matrice $A^t A$ soit inversible. On peut donc écrire

$$\theta = (A^t A)^{-1} A^t b = (A^t A)(A^t A)^{-2} A^t b = A^t \alpha$$

avec $\alpha = A(A^t A)^{-2} A^t b$. On voit donc qu'on peut toujours écrire le résultat de la régression sous la forme d'une combinaison linéaire des points d'échantillonnages:

$$\theta = \sum_{i=1}^{n} \langle \alpha_i, x_i \rangle$$

avec

$$\alpha = A(A^t A)^{-2} A^t b = A(A^{-1} A^{-t})(A^{-1} A^{-t}) A^t b = A^{-t} A^{-1} b = (AA^t)^{-1} b$$

Forme duale

Les paramètres $\theta = (\omega, b)$ du modèle de régression linéaire sont solution de

$$\theta = \sum_{i=1}^{n} \langle \alpha_i, \tilde{x}_i \rangle$$

avec $\tilde{x}_i = (x, 1)^t$ et α solution de $K\alpha = b$ ou $K \in \mathcal{M}_{n,n}$ définie par $K_{ij} = \langle \tilde{x}_i, \tilde{x}_j \rangle_{\mathbb{R}^{d+1}}$.

Résolution I

- **Régression polynomiale**: on crée des nouvelles données $z = P(x)$ avec P une transformation polynomiale puis on fait une régression: $f_\theta(x) = (\omega, z) + b$
- Méthodes de résolution:
  - On résout l'équation normale (inversion, SVD)
  - Méthode de gradient par mini lot. On veut minimiser:

$$\min_{\theta \in \mathbb{R}^{n_\theta}} \mathcal{J}(\theta)$$

avec

$$\mathcal{J}(\theta) = \sum_{i=1}^{N} j_i(\theta), \quad j_i(\theta) = \| f_\theta(x_i) - y_i \|^2$$

  - Itéré gradient stochastique:

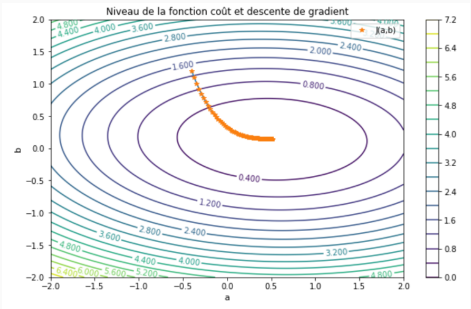
$$\theta_{k+1} = \theta_k - \eta(\frac{1}{m} \sum_{i \in I_k} \nabla_\theta j_i(\theta_k))$$

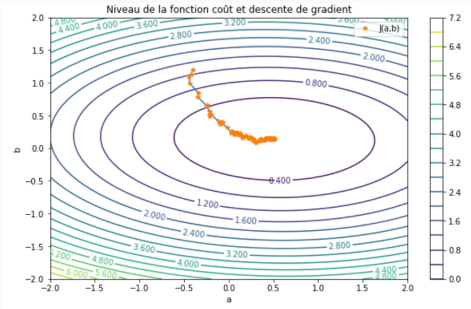
avec I_k un sous ensemble aléatoire d'indice et $m = Card(I_k)$

  - $N = m$ donne le gradient classique, $m = 1$ gradient stochastique.

Résolution II

- Exemples de descente de gradient sur le problème $y = ax + b$





- Le gradient classique converge plus vite mais chaque itération est plus lourde.
- Le coté stochastique permet de l'exploration dans le cas non convexe.

Sur-apprentissage

- **Cas linéaire**. En général on a $n >> d$. Ce qui revient a dire qu'on un nombre de données nettement plus grand que la dimension d'entrée. En général dans ce cas, A est de rang maximum et on a une unique solution.
- Si $d > n$, le rang est au mieux n (dimension de l'espace d'arrivée) dans par le Th du rang $dimKerA = dimKer(A^tA) > 0$ donc A^tA n'est pas inversible donc il n'y a pas unicité.

Remarque

Avec suffisamment de paramètres (ici $d+1$) il existe plusieurs façon passer exactement par les points de données. Exemple: on a une donnée x_1 et deux paramètres

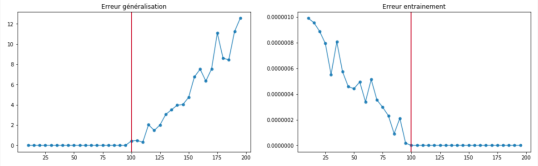
$$f_\theta(x) = ax + b$$

Il existe une infinité de couple (a, b) passant par x_1.

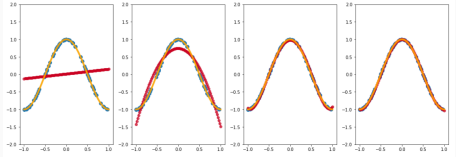
- On peut exactement approcher les données. Bonne nouvelle ?

Sur-apprentissage II

- Erreur sur les données et celle sur de nouvelles données en fonction de n/d.

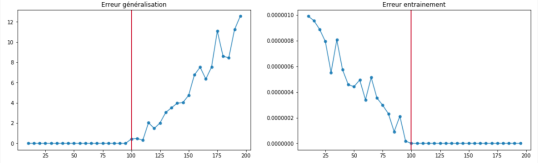


- Sur-apprentissage: on approche très bien les données et très mal de nouvelles données.
- Modèle trop riche (trop de paramètres). Approche bien les données en étant faux.
- Exemple polynomial: d = 1, nombre de paramètres= degré+1. 70 données

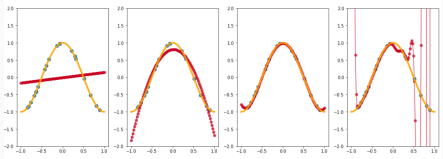


Sur-apprentissage II

- Erreur sur les données et celle sur de nouvelles données en fonction de n/d.



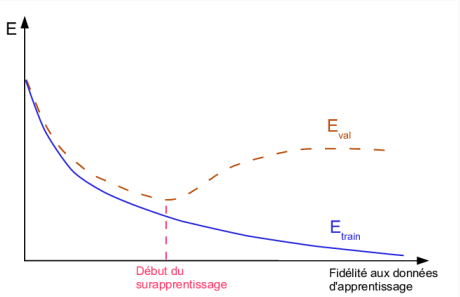
- Sur-apprentissage: on approche très bien les données et très mal de nouvelles données.
- Modèle trop riche (trop de paramètres). Approche bien les données en étant faux.
- Exemple polynomial: d = 1, nombre de paramètres= degré+1. 20 données



- Modèle trop riche (trop oscillant).

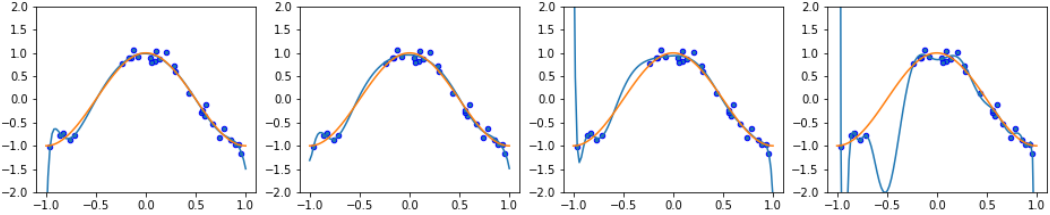
Première solution: ”earling stopping”

- Afin de déterminer ce sur-apprentissage on procède de la façon suivante:
  - On prend notre jeu de données $\mathcal{X}$ et on le divise en deux jeux de données $\mathcal{X}_{train}$ et $\mathcal{X}_{test}$
  - On procède à l'apprentissage avec le jeu de données d'entraînement $\mathcal{X}_{train}$
  - On vérifie les propriétés de généralisation du modèle avec le jeu de données test $\mathcal{X}_{train}$
- Le phénomène d'apprentissage d'un point de vue des erreurs ressemble souvent à la courbe



Première solution: ”earling stopping”

- On propose une régression polynomiale avec un polynôme de degré 30 et un échantillon de taille $n = 30$ sur une fonction sinus. On va utiliser un processus itératif (type gradient) et stopper a plusieurs moments.



- On voit qu’on limite le sur-apprentissage en arrêtant plus tôt.
- On peut utiliser les données tests pour monitorer l’apprentissage et arrêter à temps. On parle de ”early stopping”.

MAP et régularisation

- **Sur-apprentissage**: modèle avait trop de liberté pour approcher le jeu de données.
- **Autre solution**: ajouter un a priori sur les fonctions paramétriques qu'on recherche afin de contraindre l'apprentissage et éviter le sur-apprentissage.

Cadre Bayésien

L'**apprentissage bayésien** et une variante de l'apprentissage ou l'on peut aussi se donner une connaissance a priori sur le modèle à travers une loi de probabilité sur les paramètres.

- Jeux de données observées $\mathcal{X}$ et cibles $\mathcal{Y}$. Paramètres du modèle: θ.

Distribution d'échantillonnage

On appelle la **distribution d'échantillonnage** (ou aussi la fonction de vraisemblance):

$$p(\mathcal{Y} \mid \mathcal{X}, \theta)$$

si on a des données observées et cibles.

MAP et régularisation II

Distribution à priori

On appelle la **distribution à priori**: $p(\theta \mid \alpha)$ avec α les **hyperparamètres**. Cet a-priori est le coeur de l'apprentissage Bayésien.

Vraisemblance marginale

On appelle la **vraisemblance marginale**:

$$p(\mathcal{Y} \mid \mathcal{X}, \alpha) = \int_{\theta} p(\mathcal{Y} \mid \mathcal{X}, \theta) p(\theta \mid \alpha) d\theta$$

Distribution a posteriori

On appelle la distribution a posteriori et la distribution a posteriori prédictive:

$$p(\theta \mid \mathcal{X}, \mathcal{Y}, \alpha) = \frac{p(\mathcal{Y} \mid \mathcal{X}, \theta, \alpha) p(\theta \mid \alpha)}{p(\mathcal{Y} \mid, \mathcal{X}, \alpha)}, \quad p(y \mid \mathcal{Y}, \mathcal{X}, \alpha) = \int p(y \mid x, \theta) p(\theta \mid \mathcal{Y}, \mathcal{X}, \alpha) d\theta$$

MAP et régularisation III

- Pour construire le modèle, on souhaite appliquer une sorte de maximum de vraisemblance, mais en tenant compte de l'a priori.
- On introduit une généralisation de l'estimateur du maximum de vraisemblance.

Définition: Estimateur du maximum a posteriori

Soit une distribution d'échantillonnage $f(x \mid \theta)$. Soit $(x_1, ..., x_n)$ un échantillon de vraisemblance

$$\mathcal{L}(x_1, ..., x_n \mid \theta)$$

On appelle **estimateur du maximum a posteriori** la solution de

$$\hat{\theta}_{map} = \text{argmax}_\theta \, \mathcal{L}(\theta \mid x_1, ..., x_n, \alpha) = \text{argmax}_\theta \, \mathcal{L}(x_1, ..., x_n \mid \theta, \alpha) p(\theta \mid \alpha)$$

avec $p(\theta \mid \alpha)$ distribution a-priori.

- La réécriture se fait par la formule de Bayes.

Régression Ridge

- Un a priori raisonnable (notamment si on normalise les données) et lorsqu'on est en grande dimension c'est que les poids θ soit petits.
- Cela ce traduit par: $p(\theta \mid \alpha) = \mathcal{N}(0, \frac{1}{\alpha})$ avec α un hyperparamètre de la loi normale.

Lemme: Modèle Ridge par maximum a posteriori

Les paramètres $\theta = (\omega, b)$ du modèle de régression linéaire probabiliste avec a-priori Gaussien construit par maximum a posteriori sont solutions du problème de régression Ridge:

$$\min_\theta \left(\| b_y - A_x \theta \|_2^2 + \lambda \| \theta \|_2^2 \right)$$

avec $A_x \in \mathcal{M}_{n,d+1}(\mathbb{R})$ et $b_y \in \mathcal{M}_{n,1}(\mathbb{R})$. Le problème au moindre carré admet une unique solution $\forall \lambda > 0$.

- Cela revient a minimiser la norme 2 des poids en même temps que l'erreur.
- On récupère de unicité et si $\lambda << 1$ on reste proche du problème initiale.

Démonstration

- On cherche donc a résoudre:

$$\text{argmax}_\theta \, \mathcal{L}(y_1, \ldots, y_n \mid \theta, \alpha) p(\theta \mid \alpha)$$

- Cela revient donc à maximiser

$$\mathcal{L}(y_1, \ldots, y_n, \theta\alpha) p(\theta \mid \alpha) = \left(\Pi_{i=1}^n \mathcal{N}(y_i \mid (\omega, x_i) + b, \sigma^2)\right) \mathcal{N}(0, \frac{1}{\alpha})$$

- Comme dans le cas du maximum de vraisemblance on passe au log ce qui revient à

$$\log(\mathcal{L}(y_1, \ldots., y_n, \theta, \alpha) p(\theta \mid \alpha)) = -\frac{1}{2\sigma^2} \sum_{i=1}^n ((\omega, x_i) + b - y_i)^2 + \frac{N}{2} \ln(\sigma^{-2}) - N \ln(2\pi) - \frac{\alpha}{2}(\omega, \omega) + b^2) + \frac{N}{2} \ln \alpha$$

- Résoudre ce problème revient donc au final a minimiser la fonctionnelle

$$\mathcal{J}(\theta) = \frac{1}{2} \sum_{i=1}^n ((\omega, x_i) + b - y_i)^2 + \frac{\sigma^2 \alpha}{2}(\omega, \omega) + b^2)$$

- On réécrit sous forme moindres carrés et on prend $\lambda = \sigma^2 \alpha$ pour conclure.
- Pour l'unicité on refait le calcul de gradient (idem que moindre carré)

$$\nabla_\theta \mathcal{J}(\theta) \leftrightarrow (A_x^t A_x + \lambda I_d)\theta = A_x^t b$$

- On démontre facilement que $A_x^t A_x$ est symétrique positive donc si $\lambda > 0$ la matrice globale est inversible.

Lasso

Parcimonie

L'hypothèse de Parcimonie revient a supposer que un grand nombre de paramètres sont nuls. Cela veut dire en régression linéaire qu'un grand nombre de variable ne contribue pas au modèle.

- Cela ce traduit par une loi a priori de Laplace $p(\theta \mid \alpha) = \mathcal{L}(0, \frac{1}{\alpha})$ avec α un hyper paramètre de la loi de Laplace.

Lemme: Modèle Lasso par maximum a posteriori

Les paramètres $\theta = (\omega, b)$ du modèle de régression linéaire probabiliste avec a priori de Laplace construit par maximum a posteriori sont solutions du problème de régression Lasso:

$$\min_\theta \left(\| b_y - A_x \theta \|_2^2 + \lambda \| \theta \|_1 \right)$$

avec $A_x \in \mathcal{M}_{n,d+1}(\mathbb{R})$ et $b_y \in \mathcal{M}_{n,1}(\mathbb{R})$.

Lasso

- Pourquoi lé norme l^1 favorise la parcimonie ?

Solution Lasso cas simple

Soit le problème de Régression Lasso, avec $A_x^t A_x = I_d <$. La solution est donnée par:

$$\theta_j^* = sign(\theta_j^{mc}) max(0, \mid \theta_j^{mc} \mid -\lambda), \quad \forall j \in \{1, .., d+1\}$$

avec θ^{mc} la solution du problème aux moindres carrés sans régularisation.

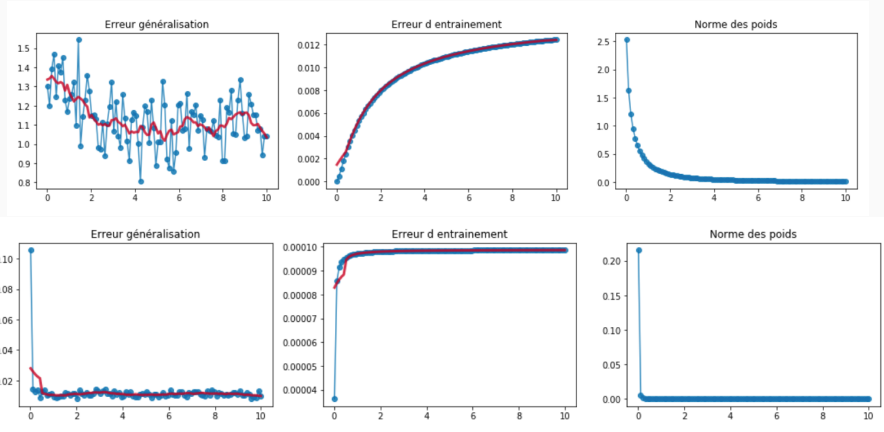
- En développant on voit que le problème de régression Lasso devient $\text{argmin}_\theta -(b, A\theta) + (\theta, \theta) + \lambda \sum_{i=1}^{d+1} \mid \theta_i \mid$
- On pose $\theta^{mc} = (A^t A)^{-1} A^t b = A^t b$. Par les propriétés du produit scalaire on obtient

$$\text{argmin}_\theta -(A^t b, \theta) + (\theta, \theta) + \lambda \sum_{i=1}^{d+1} \mid \theta_i \mid \longleftrightarrow \text{argmin}_\theta -(\theta^{mc}, \theta) + (\theta, \theta) + \lambda \sum_{i=1}^{m+1} \mid \theta_i \mid$$

- On remarque que si $\theta_j^{mc} > 0$ alors $\theta_j \geqslant 0$ et si $\theta_j^{mc} < 0$ alors $\theta_j \leqslant 0$. La fonction n'étant pas dérivable en zéro on sépare les cas:
  - $\theta_j^{mc} > 0$ dans ce cas la fonction minimisée est $f(\theta_j) = -\theta_j^{mc} \theta_j + \frac{1}{2}\theta_j^2 + \lambda\theta_j$ donc en résolvant $f'(\theta_j) = 0$ implique $\theta_j = \theta_j^{mc} - \lambda$. Comme $\theta_j \geqslant 0$ on a $\theta_j = max(0, \theta_j^{mc} - \lambda)$.
  - $\theta_j^{mc} < 0$: Par le même raisonnement on obtient : $\alpha_j = max(0, -\theta_j^{mc} - \lambda)$

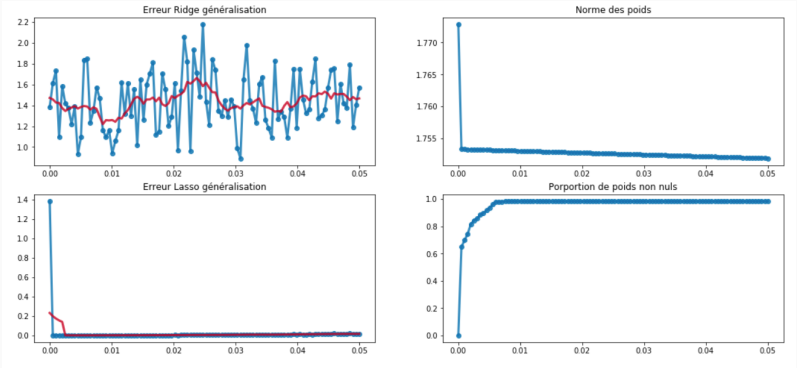
Exemple Ridge et Lasso

- Deux problème Ridge avec $d = 250$. Les données sont générés avec des θ^* associés à $\alpha = 6$ et $\alpha = 45$.
- On fait varier λ



Exemple Ridge et Lasso

- On regarde le problème $y = x_2 - 2x_4 + 1.5x_{d-2}$.
- on prend $n = 100$ et on fait varié λ



Régression Bayesienne

- Jusqu'a présent on a toujours construit le meilleur modèle dans un certain sens.
- **Régression linéaire Bayesienne**. Idée: plusieurs modèles sont très bon (plusieurs choix de θ), certains moins bons mais plus robustes au sur-apprentissage. Pour cela on va donc construire la loi des paramètres et donc des modèles et pas juste le meilleur.

Objectif

Par rapport à l'approche précédente cela va revenir à ne pas chercher un jeu unique de poids $\theta = (\omega, b)$ qui maximise la loi de probabilité a posteriori $p(\theta \mid x, y)$ mais la loi en elle-même. On a donc

- Loi a priori $p(\theta \mid \alpha) = \mathcal{N}(0, \frac{1}{\alpha})$.
- On a supposé $p_\theta(y \mid x, \theta) = \mathcal{N}((\omega, x) + b, \sigma^2)$, on a donc la loi d'échantillonnage:

$$p_\theta(\mathcal{Y} \mid \mathcal{X}, \theta, b) = \Pi_{i=1}^{n} \mathcal{N}(y_i \mid (\theta, x_i) + b, \sigma^2)$$

L'objectif est donc de calculer la loi a posteriori.

Proposition

La loi a posteriori est donnée par

$$p_\theta(\theta \mid \mathcal{Y}, \mathcal{X}) = \mathcal{N}(\mu_{pos}, \Sigma_{pos})$$

de paramètres

$$\mu_{pos} = \frac{b^t}{\alpha}\left(\frac{AA^t}{\alpha} + \sigma^2 I_d\right)^{-1} b, \quad \Sigma_{pos} = \frac{I_d}{\alpha} - \frac{A^t}{\alpha^2}\left(\frac{AA^t}{\alpha} + \sigma^2 I_d\right)^{-1} A$$

- Pour faire une prédiction il faut calculer la loi a posteriori de prédiction:

$$p(x \mid \mathcal{X}, \alpha) = \int p(x \mid \theta) p(\theta \mid \mathcal{X}, \alpha) d\theta$$

- Moyenne les prédictions de tout les modèles pondérés par leurs probabilités.
- Les calculs et preuves sont basés sur l'algèbre des lois normales.
- Pour d'autre a-priori pas de analytique. On doit estimer les intégrales numériquement (MCMC, Gibbs etc).

Régression Noyau I

- Autre **modèle nonlinéaire en** x: les méthodes à noyaux.
- **Idée**: Puisqu'on sait traiter le cas linéaire peut-on se ramener à un problème linéaire ?

Fonction de re-description

Soit $f(x) : \mathbb{R}^d \to \mathbb{R}$. Soit $\phi(x) : \mathbb{R}^d \to H$ avec $(H, \langle . \rangle)$ un espace de Hilbert (de dimension finie ou infinie). On dit que ϕ est une une fonction de re-description et H l' espace de re-description associé si $f(x) = \langle \theta, \phi(x) \rangle_H$

- Si on connaît ϕ on voit donc qu'on peut proposer une régression linéaire pour construire une approximation de f
- Régression linéaire vs nonlinéaire
  - **Régression linéaire**: $f(x) = \langle w, x \rangle_{\mathbb{R}^{d+1}}$
  - **Théorème**: Il existe $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ tels que $\theta = \sum_{i=1}^n \alpha_i x_i$
  - **Prédiction**: $f(x) = \sum_{i=1}^n \alpha_i \langle x_i, x \rangle_{\mathbb{R}^{d+1}}$
  - **Régression à noyau**: $f(x) = \langle w, \phi(x) \rangle_{\mathbb{R}^{d+1}}$
  - **Théorème**: Il existe $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ tels que $\theta = \sum_{i=1}^n \alpha_i \phi(x_i)$
  - **Prédiction**: $f(x) = \sum_{i=1}^n \alpha_i \langle \phi(x_i), \phi(x) \rangle_H$

Régression Noyau II

- Comment choisir ϕ et calculer $\langle ., . \rangle_H$?

RKHS

Soit V un espace métrique. Soit H un espace de Hilbert de fonctions réelles définies sur V. Une fonction $k : V \times V \to \mathbb{R}$ est appelé un noyau reproduisant si

- Contient toutes les fonctions de la forme: $\forall x \in H, \quad k_x(y) \to k(x, y)$
- $\forall x \in V, f \in H$ on a: $f(x) = \langle f, k_x \rangle_H$

Si ce noyau existe alors H est appelé espace de Hilbert a noyau reproduisant.

Théorème d'Aronszajn

- Si on a une fonction k semi-defini positive, alors il existe un rkhs ayant k pour un unique noyau reproduisant.
- Si k est un noyau reproduisant il vérifie la propriété reproduisante:

$$\forall (x, y) \in V^2, \quad k(x, y) = \langle k(., x), k(., y) \rangle_H$$

Régression Noyau III

Théorème de représentation

Soit V un ensemble muni d'un noyau semi-définie positif k, on nomme H_k le RKHS associé et $\mathcal{X} = (x_1, ..., x_n) \in V$ un sous-ensemble fini. Soit $(y_1, ..., y_n)$ avec $y_i \in \mathbb{R}, \quad \forall 1 \leqslant i \leqslant n$. Soit $L(x, y) \in \mathbb{R} \in \mathbb{R}^2 \to \mathbb{R}$ et $\lambda > 0$. Alors, toute solution au problème :

$$\min_{f \in H_k} \left(\sum_{i=1}^{n} L(f(x_i), y_i) + \lambda \parallel f \parallel_{H_k} \right)$$

admet une représentation de la forme $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$

- Si on prend L la norme 2 on retrouve la régression classique.
- Si on prend $\phi(x) = k(., x)$ on une fonction de re-description qui nous plonge dans un espace de dimension infini: le RKHS.
- Cependant le produit scalaire dans cet espace est facile a calculer.
- Le dernier résultat nous montre que la régression $f(x) = \sum_{i=1}^{n} \alpha_i \langle \phi(x_i), \phi(x) \rangle_H$ qui utilise $\phi(x) = k(., x)$ est la meilleure solution pour approcher les fonctions de H_k.

Régression Noyau IV

Remarque

Utiliser le modèle paramétrique $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$ est optimal pour approcher une fonction $f \in H_k$ si on a n points. Approche intéressante si H_k est assez riche

Exemple de noyaux:

- **Gaussien**: $K(x, y) = K(x - y) = e^{\frac{\|x-y\|_2^2}{2\sigma^2}}$. RKHS associé est

$$H_k = \left\{ f \in L^2(\mathbb{R}^d), \quad \text{tel que} \quad \int |\hat{f}(\omega)|^2 \, e^{\frac{\sigma^2 \omega}{2}} \leqslant \infty \right\}$$

  Cela correspond aux fonctions $C^\infty(\mathbb{R}^d)$
- **Laplace**: $K(x, y) = K(x - y) = e^{\frac{\|x-y\|_1}{2\sigma^2}}$. RKHS associé est

$$H_k = \left\{ f \in L^2(\mathbb{R}), \quad \text{tel que} \quad \int |\hat{f}(\omega)|^2 \, \frac{\gamma + \omega^2}{\gamma} \leqslant \infty \right\}$$

  Cela correspond aux fonctions $H^1(\mathbb{R}^d)$
- **Matern** (α, h) génèrant les Sobolev $H^{\alpha + \frac{d}{2}}(\mathbb{R}^d)$

Processus Gaussien

Processus Gaussien

Un processus stochastique est une famille de variables aléatoires définies sur le même espace de probabilité indexée par T un ensemble. Un processus stochastique $\{X_t\}_{t\in T}$ est dit **Gaussien** si toute combinaison linéaire

$$a_1 X_{t_1} + \cdots + a_n X_{t_n}$$

suit une loi Gaussienne (pour tout $n \in \mathbb{N}, t_1, \ldots, t_n \in T$ et $a_1, \ldots, a_n \in \mathbb{R}$).

- Si T est un ensemble indénombrable, une réalisation de ce processus donne donc une fonction $f : T \to M$.
- Les processus Gaussien (GP) permettent de d'échantillonner des fonctions.
- GP déterminés par une fonction moyenne $\mu(x)$ et un noyau de covariance $k(x_1, x_2)$.

A priori

Principe: régression faisant l'a priori que la fonction cible peut être générée par une processus Gaussien.

Processus Gaussien II

Idée

On cherche f tel que $y_i = f(x_i) + \epsilon$ et on suppose que

$$f(x) \sim \mathcal{GP}(\mu(x), \Sigma(x_1, x_2))$$

- Sur les données cela revient a supposer que:

$$Y = (y_1, ..., y_n) \sim \mathcal{GP}(V_\mu, M_\Sigma)$$

avec V_μ $(\mu(x_1), ...\mu(x_n))$ et $M_{\Sigma,ij} = k(x_i, x_j)$

- On veut maintenant prédire les $Y^* = (f(x_1^*), ..., f(x_m^*))$. Pour ca on va calculer

$$p(Y^* \mid Y, X^*)$$

- Le choix du noyau permet de choisir un a priori sur la classe de fonction (même noyaux qu'en régression à noyau). Ensuite on calcule les nouvelles valeurs en conditionnant le résultat avec les valeurs connues.
- Exemple: `http://www.infinitecuriosity.org/vizgp`.

Introduction à l'apprentissage machine

Rappels

Apprentissage supervisé

Apprentissage supervisé profond

Apprentissage supervisé profond

Réseaux de neurones I

- Les réseaux de neurones sont un autre type de modèles paramétriques. Ils sont non-linéaire par rapports aux entrées, mais aussi aux paramètres.

Couche

On appelle une couche la fonction $L : \mathbf{x} \in \mathbb{R}^{d_i} \to \mathbf{y} \in \mathbb{R}^{d_{i+1}}$ définie par

$$L_{i,i+1}(\mathbf{x}) = \sigma(A\mathbf{x} + \mathbf{b})$$

avec $A \in \mathcal{M}_{d_i,d_{i+1}}(\mathbb{R})$, $\mathbf{b} \in \mathbb{R}^{d_{i+1}}$ et $\sigma()$ une fonction nonlinéaire appliquée terme à terme. On nomme $\sigma()$ la fonction d'activation. Les coefficients de A et $\mathbf{b}$ sont appelés paramètres à déterminer (entrainable).

Réseau de neurones

On appelle un réseau de neurones une fonction paramétrique $N_\theta : \mathbf{x} \in \mathbb{R}^{d_{in}} \to \mathbf{y} \in \mathbb{R}^{d_o}$ qui est défini par

$$N_\theta(\mathbf{x}) = L_{o,n} \circ ... \circ L_{i+1,i} \circ ... \circ L_{1,in}(x)$$

avec θ l'ensemble des paramètres entraînables.

Réseaux de neurones II

- Exemple de fonctions d'activation:
  - **ReLu**: $\sigma(x) = \max(0, x)$
  - **Softplus**: $\sigma(x) = \ln(1 + e^x))$
  - **Tangente hyperbolique**: $\sigma(x) = \tanh(x)$
- **Approximation**: Il existe des théorèmes de densité des réseaux de neurones dans les fonctions continues.
- **Apprentissage**: on utilise des méthodes de gradients.
- Comment calcul t'on le gradient d'un réseau de neurones: formule de dérivation des fonctions composées.
- Exemple:
  - On prend $f_\theta(\mathbf{x}) = f^3_\theta \circ f^2_\theta \circ f^1_\theta(\mathbf{x})$
  - On appelle $\mathbf{h}_i$ la sortie de f_i et θ le vecteur de tous les paramètres du réseau.

$$\nabla_\theta \mathcal{J}(\theta) = \left(\nabla_{\mathbf{h}_3} \mathcal{L}\right) \frac{\partial f_\theta(\mathbf{h}_3)}{\partial \theta}$$

$$\nabla_\theta \mathcal{J}(\theta) = \left(\nabla_{\mathbf{h}_3} \mathcal{L}\right) \left(\frac{\partial f^3_\theta}{\partial \theta} + \frac{\partial f^3_\theta}{\partial \mathbf{h}_2} \frac{\partial f^2_\theta}{\partial \theta} + \frac{\partial f^3_\theta}{\partial \mathbf{h}_2} \frac{\partial f^2_\theta}{\partial \mathbf{h}_1} \frac{\partial f^1_\theta}{\partial \theta} \right)$$

$$\nabla_\theta \mathcal{J}(\theta) = \left(\nabla_{\mathbf{h}_3} \mathcal{L}\right) \frac{\partial f^3_\theta}{\partial \theta} + \left(\nabla_{\mathbf{h}_3} \mathcal{L} \frac{\partial f^3_\theta}{\partial \mathbf{h}_2} \right) \frac{\partial f^2_\theta}{\partial \theta} + \left(\nabla_{\mathbf{h}_3} \mathcal{L} \frac{\partial f^3_\theta}{\partial \mathbf{h}_2} \right) \frac{\partial f^2_\theta}{\partial \mathbf{h}_1} \frac{\partial f^1_\theta}{\partial \theta}$$

Gradient d'un réseau profond

On se donne un réseau n couches pour commencer:

$$f_\theta(\mathbf{x}) = f^n_{\theta_n} \circ ..... \circ f^1_{\theta_1}(\mathbf{x})$$

avec $\mathbf{h}_i$ la sortie de f_i. En utilisant de la même façon le gradient d'une composition de fonction on obtient les deux inégalités suivantes:

$$\nabla_\theta \mathcal{J}(\theta) = \sum_{i=1}^{N} \nabla_{\mathbf{h}_i} \mathcal{J}(\theta) \frac{\partial f^i_\theta}{\partial \theta}$$

et le plus important:

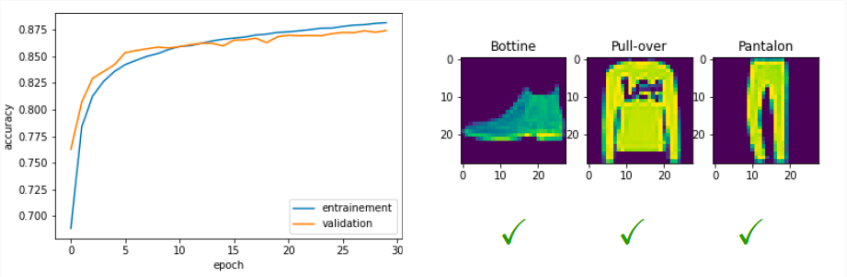
$$\nabla_{\mathbf{h}_i} \mathcal{J}(\theta) = \nabla_{\mathbf{h}_{i+1}} \mathcal{J}(\theta) \frac{\partial f^{i+1}_\theta}{\partial \theta}$$

On parle de rétropropagation.

- On va donc calculer le gradient en partant de la dernière fonction puis remonter dans le sens inverse des compositions.

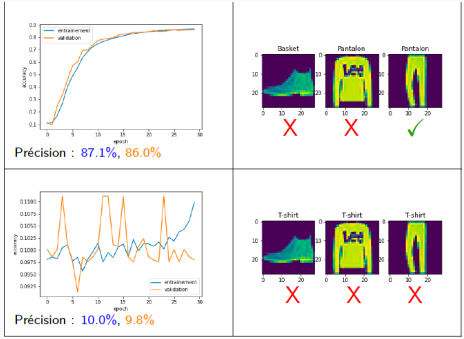
Instabilité de gradient

- La théorie semblent montrer que la profondeur semble être une solution pour améliorer la performance. En pratique c'est plus difficile.
- Jeux de données: vêtements. Résultat d'un entraînement avec un réseau à une couche cachée sigmoïde (30 neurones par couche).



Instabilité de gradient

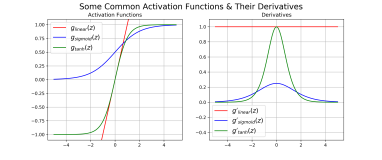
- La théorie semblent montrer que la profondeur semble être une solution pour améliorer la performance. En pratique c'est plus difficile.
- Jeux de données: vêtements. Résultat d'entraînements avec des réseaux à 4 et 6 couches cachées sigmoïdes (30 neurones par couche).



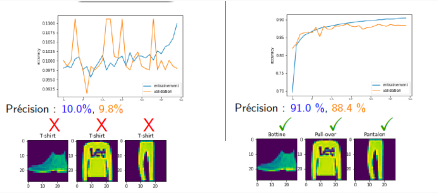
- On a $\nabla_{\mathbf{h}_i} \mathcal{J} = \nabla_{\mathbf{h}_n} \mathcal{L} \frac{\partial f_\theta^n(\mathbf{h}_{n-1})}{\partial \theta} \circ \ldots \circ \frac{\partial f_\theta^i(\mathbf{h}_i)}{\partial \mathbf{h}_i}$ qui tend vers zéro ou l'infini.

Instabilité de gradient II

- Solution 1: l'initialisation aléatoire des poids en ajustant la variance de la loi.
- Solution 2: La seconde solution a été de modifier la fonction d'activation. Dérivée activation:



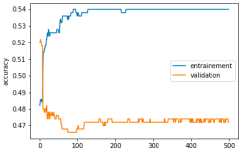
- Augmente les risques de disparition de gradients.



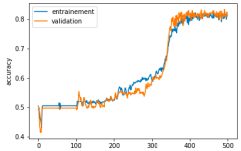
- Jeux de données: vêtements. Résultat avec réseaux à 6 couches caché, sigmoïde (gauche) ou ReLu (droite).

Instabilité de gradient III

- Ca suffit pas. Exemple de classification:

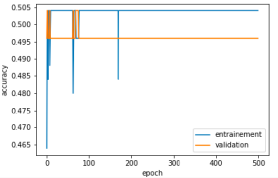


Précision : 54.0%, 47.2%



Précision : 81.4%, 81.4%

Précision : 50.4%, 49.6%



- Jeux de données: cercle. Résultat d'entraînements avec des réseaux à 6 couches caché avec des fonctions d'activation Tanh (gauche) ou ReLu (droite). En bas réseau Relu à 20 couches.

Instabilité de gradient IV

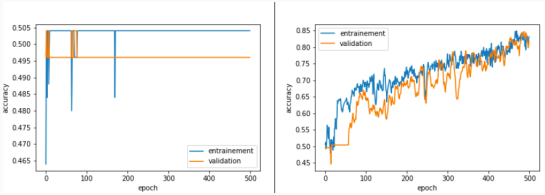
- Autre solutions: Ajout de normalisation par lot

Couche de normalisation par lot

On se donne un batch $\mathcal{B} = \{x_1, ..., x_{n_\mathcal{B}}\}$. On définit les moments:

$$\mu_\mathcal{B} = \frac{1}{n_\mathcal{B}} \sum_{i=1}^{n_\mathcal{B}} x_i, \quad \sigma^2_\mathcal{B} = \frac{1}{n_\mathcal{B}} \sum_{i=1}^{n_\mathcal{B}} (x_i - \mu_\mathcal{B})^2$$

La couche de normalisation par mini-lot est donnée par $y_i = \gamma \otimes \hat{x}_i + \beta$ avec γ et β des paramètres entraînables et $\hat{x}_i = \frac{x_i - \sigma^2_\mathcal{B}}{\sqrt{\sigma^2_\mathcal{B} + \epsilon}}$.



- Comparaison avec et sans batchnorm

Instabilité de gradient V

- Autre solutions: Resnet.

Couche résiduelle

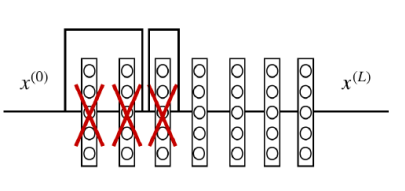
On appelle une couche de neurones résiduelle une fonction $L : \mathbf{x} \in \mathbb{R}^{d_i} \to \mathbf{y} \in \mathbb{R}^{d_{i+1}}$ définie par

$$L_{i,i+1}(\mathbf{x}) = \sigma(A\mathbf{x} + \mathbf{b} + \mathbf{x})$$

On appelle un bloc (séries de couches répétées) résiduellé un bloc de la forme:

$$B_k = \mathbf{x} + L_{n-1,n} \circ .... \circ L_{0,1}(x)$$

avec $L_{i,i+1}$ des couches.



Apprentissage I

- Pour un problème de régression on minimise:

$$\min_\theta \mathcal{J}(\theta) = \min_\theta \sum_{i=1}^{N} \| y_i - f_\theta(x_i) \|^2$$

  par des méthodes de type gradient stochastique.
- Une première solution pour améliorer les méthodes de gradient: ajouter de l'inertie.
  - La vitesse est donnée par: $v_{t+1} = -\eta \nabla_\theta \mathcal{J}(\theta_t) + \alpha v_t$.
  - Les nouveaux poids sont donnés par $\theta_{t+1} = \theta_t + v_t$.
  - Cela peut permettre de sortir des minimums locaux mais cela peut survoler le minimum global.
- Une deuxième solution: correction de Nesterov.
  - la vitesse est donnée par: $v_{t+1} = -\eta \nabla_\theta \mathcal{J}(\theta_t + \alpha v_t) + \alpha v_t$

Apprentissage II

- Troisième solution:
- Méthode adaptatif: Adagrad, RSMprop or ADAM.
- On adapte la direction de gradient.
- Pour ADAM on va estimer un gradient comme une combinaison du précédent et l'actuel gradient, de normaliser la direction de descente par une estimation de la variance de ce gradient.
  - $s_{t+1} = \rho_1 s_t + (1-\rho_1)\nabla_\theta \mathcal{J}(\theta_t)$
  - $r_{t+1} = \rho_2 r_t + (1-\rho_2)\nabla_\theta \mathcal{J}(\theta_t) \odot \nabla_\theta \mathcal{J}(\theta_t)$
  - correction $\hat{s}_{t+1} = \frac{s_{t+1}}{1-\rho_1}$, $\hat{r}_{t+1} = \frac{r_{t+1}}{1-\rho_2}$
  - update: $\theta_{t+1} = \theta_t - \epsilon \frac{\hat{s}_{t+1}}{\sqrt{\hat{r}_{t+1}+\delta}}$
- Adam est la méthode la plus fréquemment utilisée pour les réseaux.