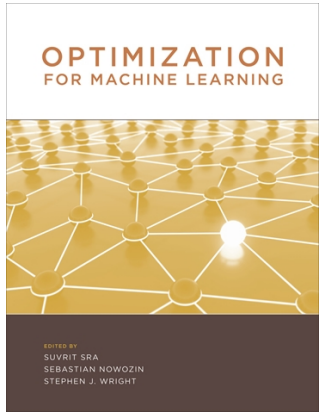
Optimisation et apprentissage

Stéphane Canu, Normandy University

asi.insa-rouen.fr/enseignants/~scanu

Journée Mathématiques/STIC Normande, Caen



Oct 20, 2017

Tentative de définition de l'apprentissage statistique

Machine Learning (T. Mitchell, 2006)

A computer program $\mathcal{A}$ learn from experience E with respect to some class of tasks T and performance measure L, if its performance at tasks in T, as measured by L, improves with experience E

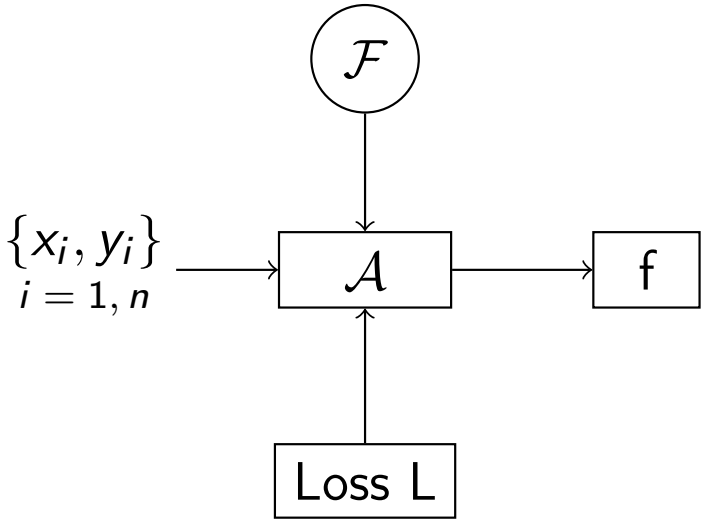
La bande des 4

- programme $\mathcal{A}$: algorithme $\mathcal{A}$
- experience E : données $\{x_i, y_i\}$
- performance measure L : cout L
- tasks T : modèle $f \in \mathcal{F}$
  - traduire
  - jouer aux échec ou au go
  - conduire
  - ...faire ce que les humains font

Apprentissage statistique et optimisation

Composantes :

- programme $\mathcal{A}$:
  algorithme $\mathcal{A}$.
- experience E :
  données $\{x_i, y_i\}$
- performance L :
  cout L .
- tâche - hypothèses T :
  modèle $f \in \mathcal{F}$

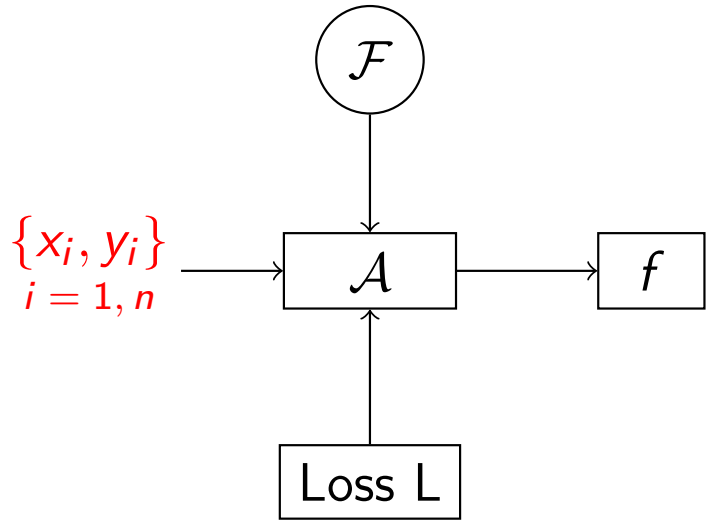


Optimisation

$$\min_{f \in \mathcal{F}} \; L(f, \{x_i, y_i\}, i = 1, n)$$

Plan de l'exposé

1. Apprendre a partir des données $\{x_i, y_i\}$
2. le Cout L : premier algorithme MRE
3. Régularisation et Pénalités : MRS et $\mathcal{F}$
4. Retour sur l'algorithme de minimisation $\mathcal{A}$
5. L'apprentissage à grande échelle : *deep learning*
6. Conclusion et perspectives



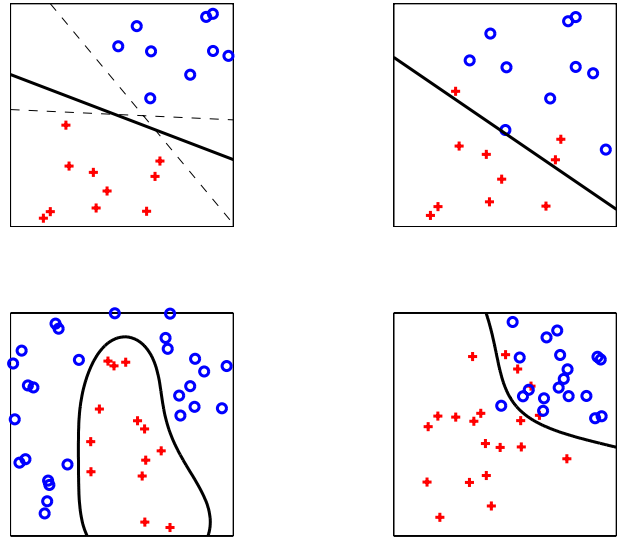
Classification à 2 classes

données
- observation : $x_i \in \mathbb{R}^p$
- étiquettes $y_i \in \{-1, 1\}$

individus $n = 20$

variables $p = 2$

classes $k = 2$



Trouver un classifieur ($f \in \mathcal{F}$) telle que

$\underbrace{f(x_i) > 0 \text{ si } x_i \in \mathit{Classe}1 \quad \text{et en même temps} \quad f(x) < 0 \text{ si } x_i \in \mathit{Classe}2}_{\text{Loss } L}$

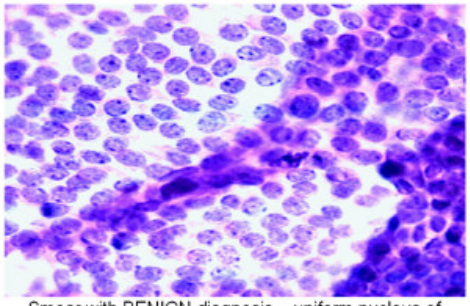
Wisconsin Diagnostic Breast Cancer (WDBC)

données
- observations : images
- caractéristiques : $x_i \in \mathbb{R}^p$
- étiquettes $y_i \in \{-1, 1\}$

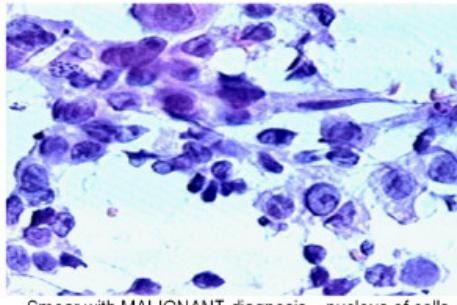
individus $n = 569$

variables $p = 30$

classes $k = 2$



Smear with BENIGN diagnosis – uniform nucleus of cells, symmetrical, homogeneous, with areas within normal size



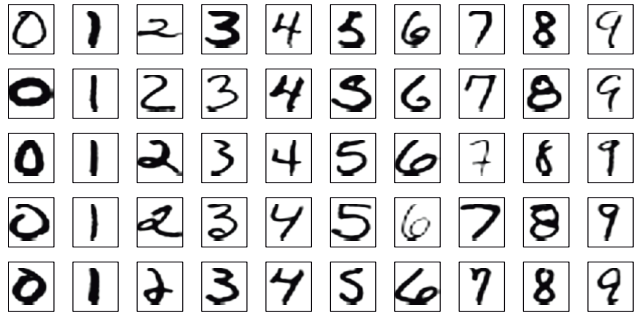
Smear with MALIGNANT diagnosis – nucleus of cells without uniformity, asymmetrical, not homogeneous (multiple sizes) and with areas above normal size

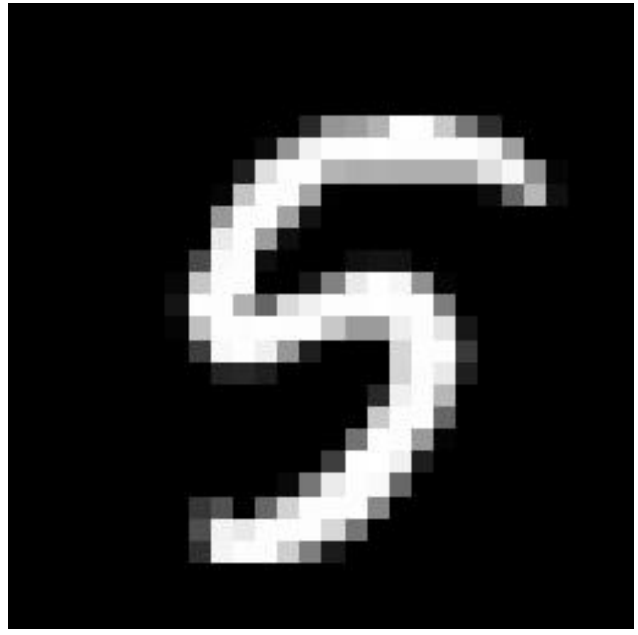
Caractéristiques extraites à la main

Image $\rightarrow$ caractéristiques $x \rightarrow$ classe (bénin ou malin) y

http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.names
ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer/

MNIST





données
- observations : $x_i \in [0, 255]^p$
  256 niveaux de gris
- étiquettes $y_i \in \{0, 1; \ldots, 9\}$

individus $n = 60\,000$

variables $p = 28 \times 28 = 784$

classes $k = 10$

Images normalisée

Image (presque brute, juste normalisée)

http://yann.lecun.com/exdb/mnist/

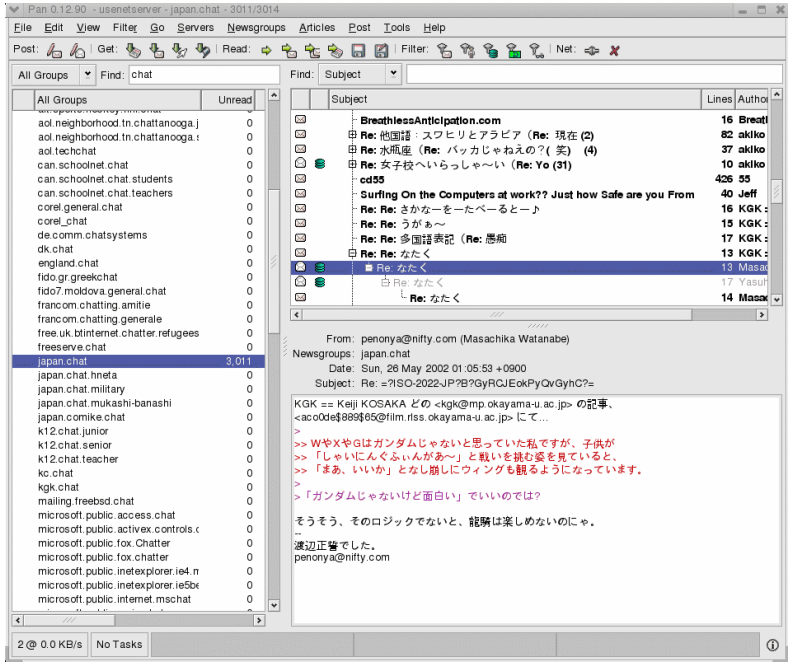
Newsgroup

données
- observations : textes
- étiquettes $y_i \in \{1; \ldots, 20\}$

individus $n = 11\,314$

variables $p = 4.8$ million

classes $k = 20 \to 2$



Texte: variables = mots

Données creuses (*sparse*)

http://www.image-net.org/challenges/LSVRC/2012/

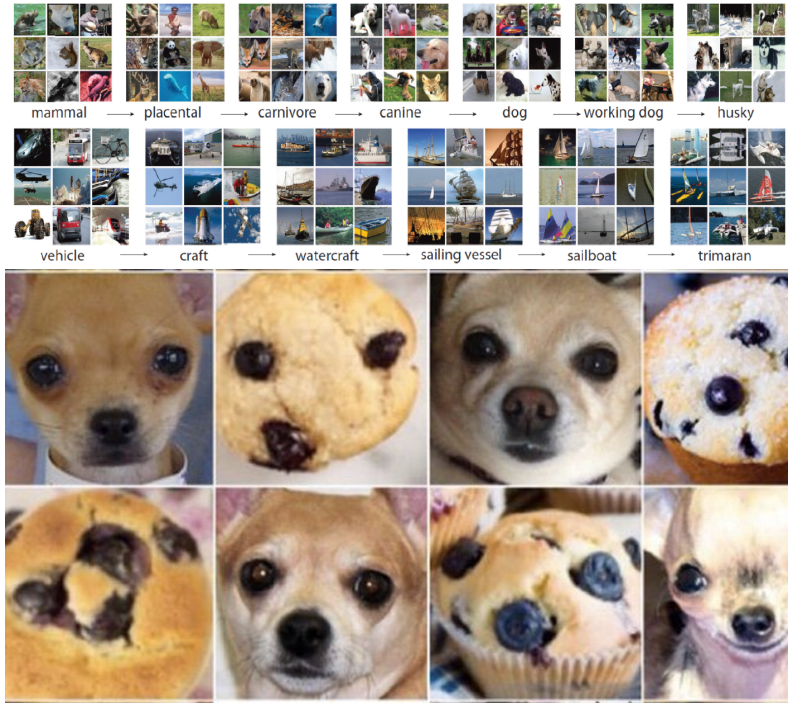
ImageNet Large Scale Visual Recognition Challenge 2012

données
- observations : images
- étiquettes $y_i \in \{1; \ldots, 1000\}$

individus $n = 1.2$ million

variables $p = 3 \times 224 \times 224 = 150\,000$

classes $k = 1000$



Images butes d'un seul objet

Étiquetées par des humains

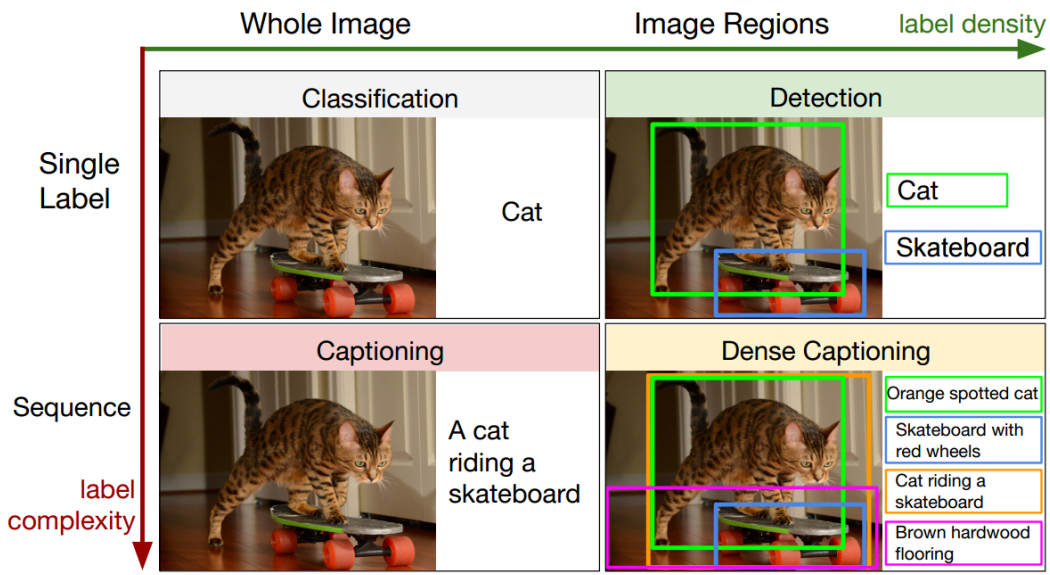
http://www.image-net.org/challenges/LSVRC/2012/

Les données

≈ chronologie	jouet	Cancer 1970	MNIST 1990	Newsgroup 2000	ImageNet 2010
individus $n =$	20	569	60 000	11 314	1.2 M $\rightarrow$ 14.2 M
variables $p =$	2	30	784	4.8 M	150 000
classes $k =$	2	2	10	20 $\rightarrow$ 2	1000

l'équation des données d'aujourd'hui (2020) : $n = p = k =$millions.

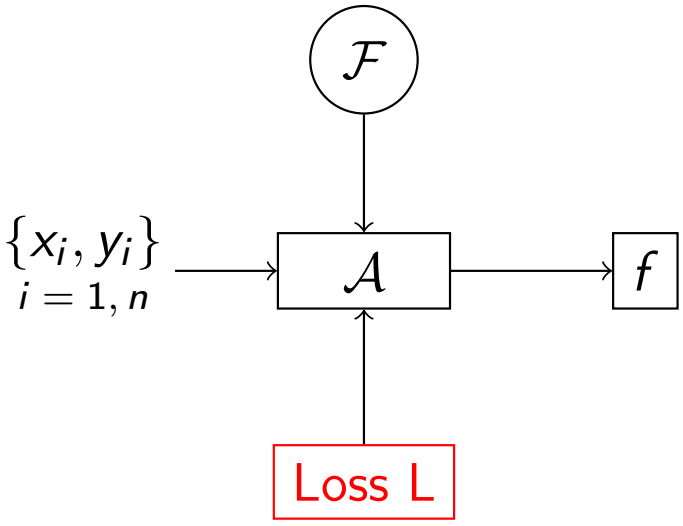
- Visual Genome `http://visualgenome.org/`
- genomique : n = 100 000 individus p = 30 millions



karpathy.github.io

Plan de l'exposé

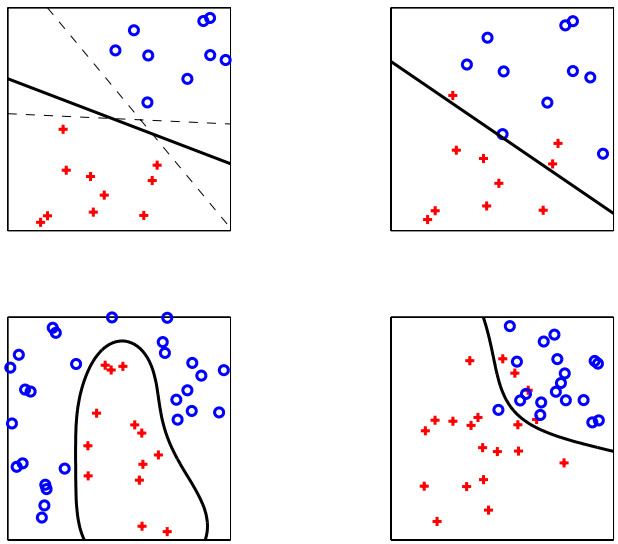
1. Apprendre a partir des données $\{x_i, y_i\}$
2. le Cout L : premier algorithme MRE
3. Régularisation et Pénalités : MRS et $\mathcal{F}$
4. Retour sur l'algorithme de minimisation $\mathcal{A}$
5. L'apprentissage à grande échelle : *deep learning*
6. Conclusion et perspectives

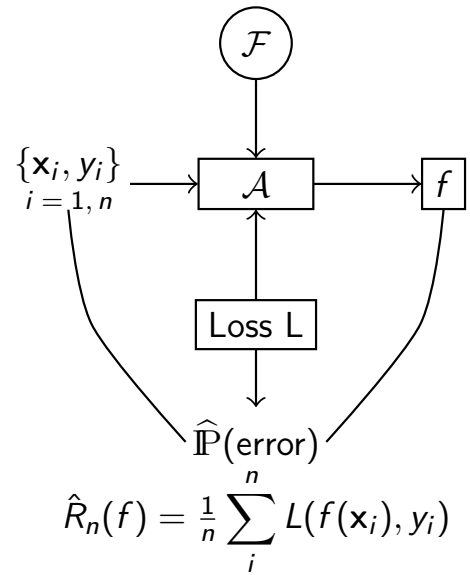


Cout et risque

Trouver un algorithme qui minimise le nombre d'erreur (le cout 0/1)

$$L(f(\mathbf{x}_i), y_i) = \begin{cases} 0 & \text{si} \quad f(\mathbf{x}_i) = y_i \\ 1 & \text{sinon} \end{cases}$$

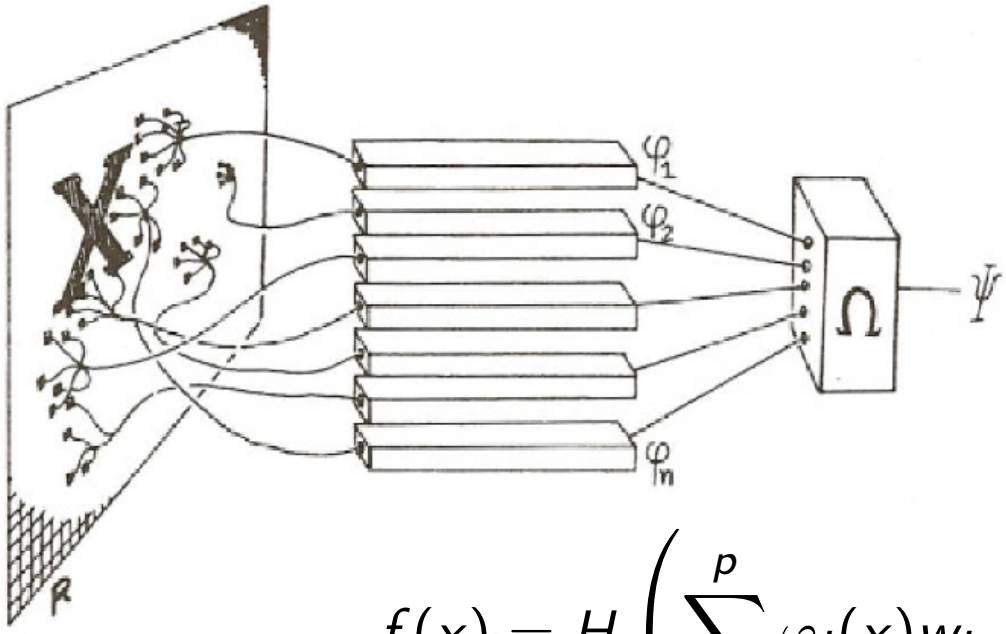


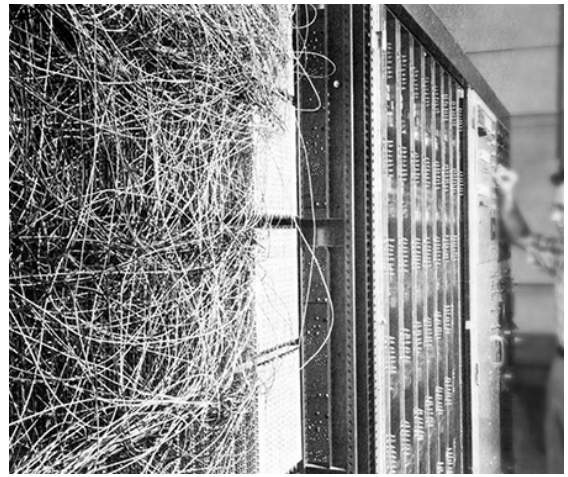


La minimisation du risque empirique $\hat{R}_n(f)$

$$\min_{f \in \mathcal{F}} \hat{R}_n(f) = \frac{1}{n}\sum_i^n L(f(\mathbf{x}_i), y_i)$$

Perceptron (Rosenblatt, 1957)





$$f(x) = H\left(\sum_{j=1}^{p} \varphi_j(x) w_j + w_0\right)$$

Règle du Perceptron

Tant que on n'a pas convergé :

1. tirer un exemple (x_i, y_i)
2. calculer la prédiction $f(x_i)$
3. adaptater les poids
   $w \leftarrow w + \frac{\rho}{2}(y_i - f(x_i))\varphi(x_i)$

Du point de vue optimisation

$$\hat{R}_n(w) = \sum_{i=1}^{n} \max(0, -y_i w^\top \varphi(x_i))$$

Méthode de gradient stochastique

La régression logistique (Cox, 1958)

Le modèle logistique : y_i réalisation d'une VA $Y \sim$ Bernouilli de paramètre

$$p(x_i) = \frac{\exp^{\varphi(\mathbf{x_i})^\top w}}{1 + \exp^{\varphi(\mathbf{x_i})^\top w}}$$

Le cout logistique : le max de vraisemblance

$$\hat{R}_n(w) = -log\mathcal{L}(w) = \sum_{i=1}^{n} -y_i \left(\varphi(\mathbf{x}_i)^\top w\right) + \log\left(1 + \exp^{\varphi(\mathbf{x}_i)^\top w}\right)$$

Cette fonction cout est non linéaire mais

- différentiable
- convexe
- consistante

Du point de vue de l'optimisation

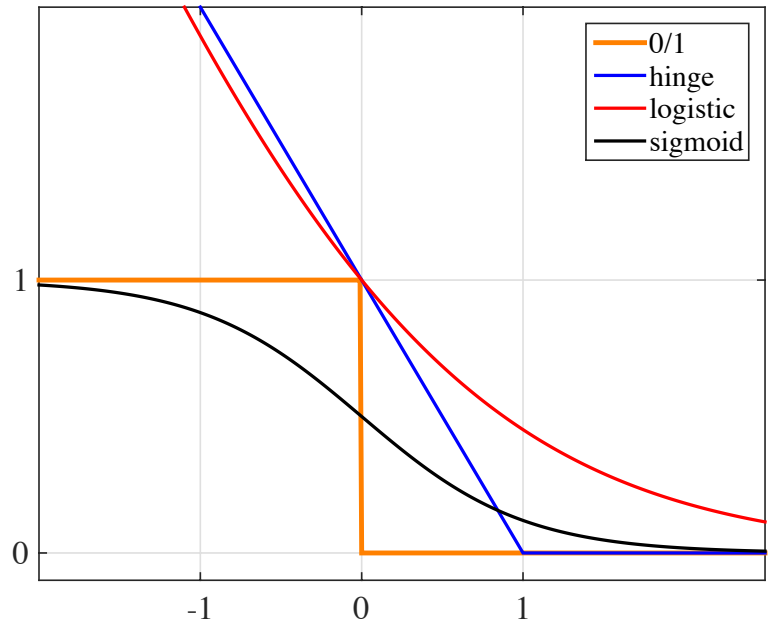
Méthode itérative du second ordre (Newton)

Exemples de différents couts utilisées en apprentissage

Principe de minimisation du risque empirique

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i}^{n} L(f(\mathbf{x}_i), y_i)$$

Loss L	Convex	Diff.
logistic $\log(1 + \exp^{f(x_i)})$	✓	✓
sigmoïd $\frac{1}{1+\exp(y_i f(x_i))}$	✗	✓
hinge $\max(0, 1 - y_i f(x_i))$	✓	✗
cout 0/1 $\begin{cases} 0 & \text{si } f(x_i) = y_i \\ 1 & \text{sinon} \end{cases}$	✗	✗



Comparaison des méthodes (cas favorable)

$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i) \qquad \hat{f} = \operatorname*{argmin}_{f \in \mathcal{F}} \hat{R}_n(f)$$

méthodes	Gradient stochastique	1er ordre Gradient	2nd ordre Q Newton
pour une itération	p	np	$p(n+p)$
nombre d'itération	$\frac{\kappa\gamma}{\varepsilon}$	$\kappa \log \frac{1}{\varepsilon}$	$\log\log \frac{1}{\varepsilon}$
temps pour ε	$\frac{\kappa\gamma p}{\varepsilon}$	$\kappa np \log \frac{1}{\varepsilon}$	$p(n+p)\log\log \frac{1}{\varepsilon}$

ε : la précision de l'algorithme $\|f_{\mathcal{A}} - \hat{f}\|^2 \leq \varepsilon$

κ : le conditionnement de H la hessienne

$\gamma = \text{trace}(H^{-1}\nabla\hat{R}_n)$

Les méthodes du second ordre sont les plus efficaces

. . . si L est différentiable et si le critère est la précision ε

Large-scalemachinelearningwithstochasticgradientdescent LBottou - ProceedingsofCOMPSTAT'2010

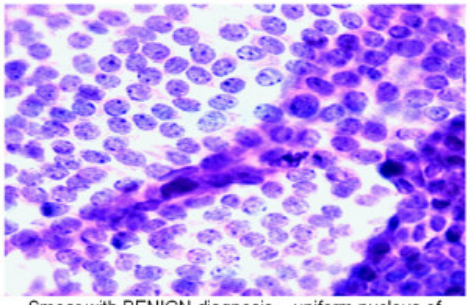
Wisconsin Diagnostic Breast Cancer (WDBC)

données
- observations : images
- caractéristiques : $x_i \in \mathbb{R}^p$
- étiquettes $y_i \in \{-1, 1\}$

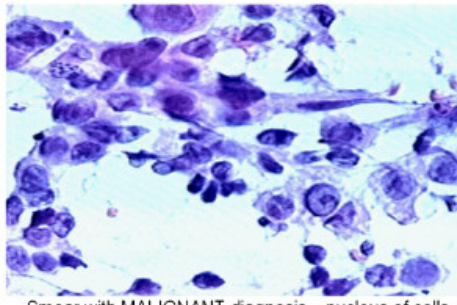
individus $n = 569$

variables $p = 30$

classes $k = 2$



Smear with BENIGN diagnosis – uniform nucleus of cells, symmetrical, homogeneous, with areas within normal size



Smear with MALIGNANT diagnosis – nucleus of cells without uniformity, asymmetrical, not homogeneous (multiple sizes) and with areas above normal size

Toutes les méthodes minimisent l'erreur en moins d'une seconde (meilleurs résultats avec des méthodes non linéaires)

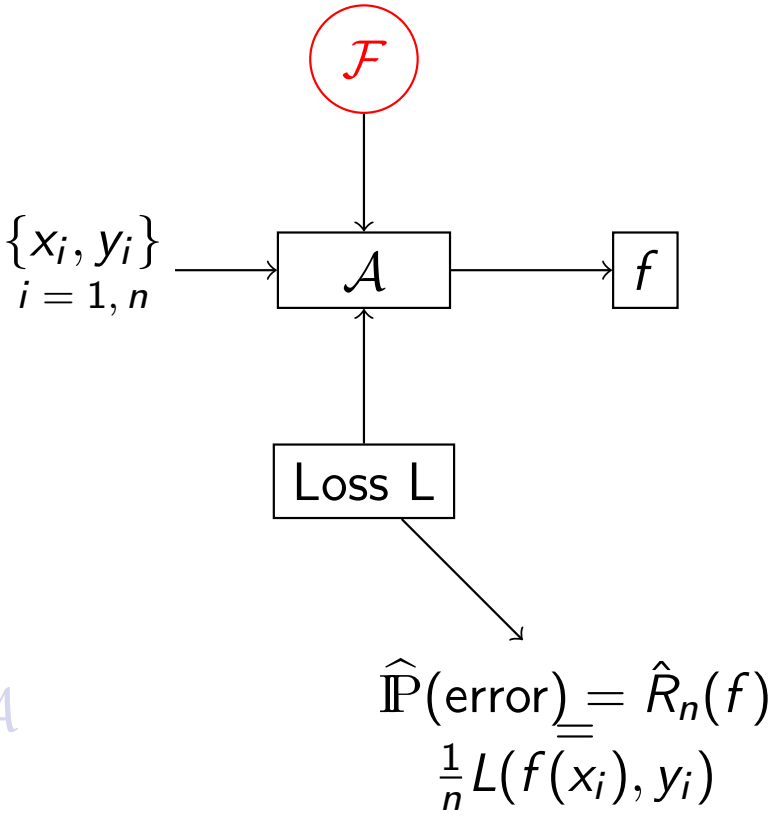
Propriété d'universalité de l'apprentissage

Que ce passe t'il lorsque p est grand ?

Breast Cancer Diagnosis and Prognosis via Linear Programming, O. Mangasarian et al., 1994

Plan de l'exposé

1. Apprendre a partir des données $\{x_i, y_i\}$
2. le Cout L : premier algorithme MRE
3. Régularisation et Pénalités : MRS et $\mathcal{F}$
4. Retour sur l'algorithme de minimisation $\mathcal{A}$
5. L'apprentissage à grande échelle : *deep learning*
6. Conclusion et perspectives



Que se passe t'il lorsque p augmente

Augmenter le nombre de caractéristiques pour obtenir l'universalité :

Les caractéristiques comme une fonction

$$
\begin{array}{rcll}
p & \longrightarrow & \infty & \\
\mathbf{x}_i & \longrightarrow & k(\mathbf{x}_i, \bullet) & \text{une fonction noyau} \\
\mathbb{R}^p & \longrightarrow & \mathcal{H} & \text{un RKHS} \\
\mathcal{F} & \longrightarrow & \mathcal{H} & \\
\mathrm{w}^\top \mathbf{x}_i & \longrightarrow & f(\mathbf{x}_i) = \langle f(\bullet), k(\mathbf{x}_i, \bullet) \rangle_{\mathcal{H}} &
\end{array}
$$

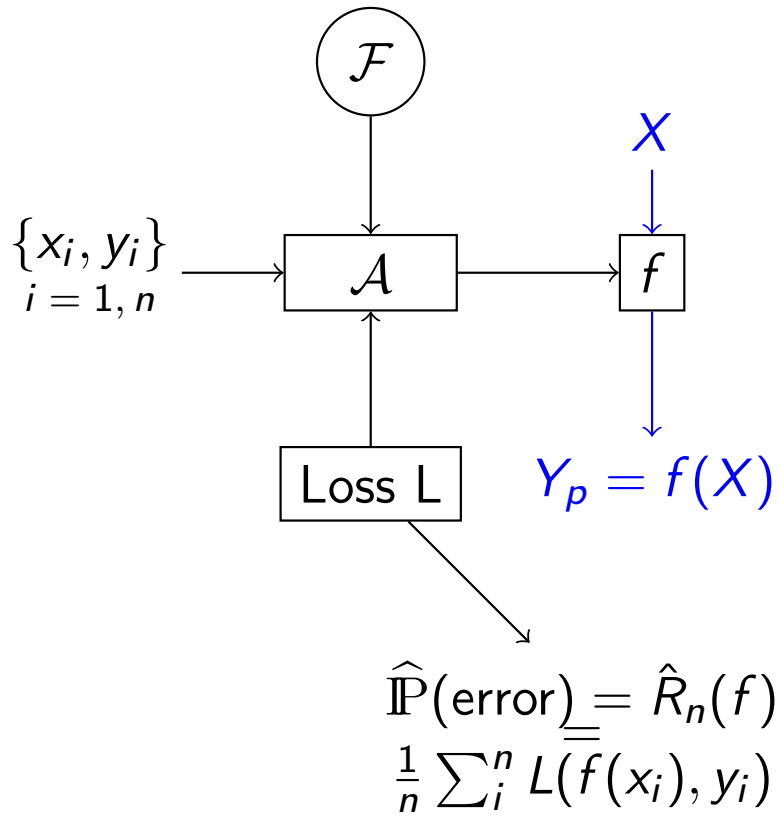
Mécaniquement :

$$\text{plus } p \text{ augmente, plus } \hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i) \text{ diminue}$$

Jusqu'à apprendre « par cœur », et mal généraliser : $\hat{R}_n(f) = 0$

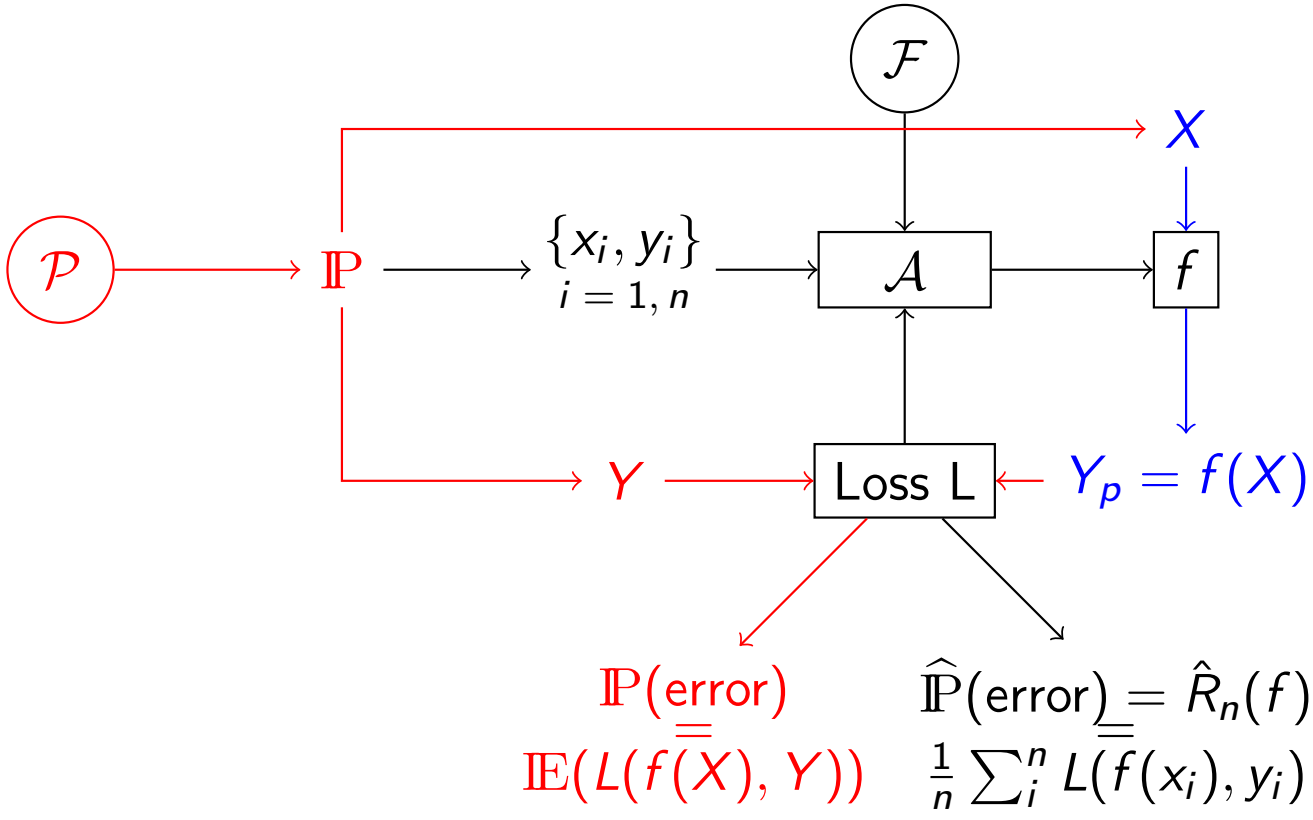
Apprentissage et généralisation

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i)$$



Apprentissage et généralisation

$$R(f) = \mathbb{E}\big(L(f(X), Y)\big) \qquad \hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i)$$



Théorie de l'apprentissage statistique (COLT)

$$R(f) = \mathbb{E}\big(L(f(X), Y)\big) \qquad \hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i)$$

$$\begin{array}{lllllll} \text{L'idéal :} & f^\star = \operatorname{argmin}_f R(f) & \Rightarrow & R(f^\star) & \leq & R(\hat{f}) \\ \text{Ce qu'on a :} & \hat{f} = \operatorname{argmin}_f \hat{R}_n(f) & \Rightarrow & \hat{R}_n(\hat{f}) & \leq & \hat{R}_n(f^\star) \end{array}$$

$$R(\hat{f}) - R(f^\star) < \underbrace{|R(\hat{f}) - \hat{R}_n(\hat{f})|}_{\leq \varepsilon} + \underbrace{|\hat{R}_n(f^\star) - R(f^\star)|}_{\leq \varepsilon}$$

$$\mathbb{P}\Big(\sup_{f \in V(\mathcal{F})} |R(\hat{f}) - \hat{R}_n(\hat{f})| \leq \varepsilon\Big) \geq \delta$$

Convergence uniforme localisée (Mendelson, Bousquet, Tsybakov, Massart)

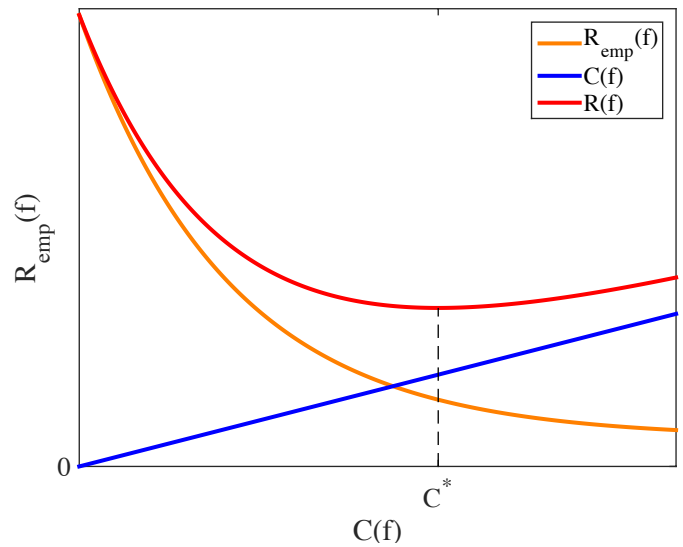
$$\varepsilon(n, C(\mathcal{F})) \propto \mathcal{O}\left(\sqrt{\frac{C(\mathcal{F})}{n} \log\left(\frac{n}{C(\mathcal{F})}\right)}\right)$$

$C(\mathcal{F})$ = mesure de la complexité de $\mathcal{F}$

Introduction~to~statistical~learning~theory,O.Bousquet,S.Boucheron,G.Lugosi.Advanced~lectures~on~ML~169-207,2004

L'apprentissage : un problème d'optimisation bi objectif

$$\begin{cases} \min\limits_{f \in \mathcal{F}} & \hat{R}_n(f) = \dfrac{1}{n} \sum\limits_{i=1}^{n} L(f(x_i), y_i) & \text{attache aux données} \\ \min\limits_{f \in \mathcal{F}} & C(\mathcal{F}) & \text{mesure de la complexité de } \mathcal{F} \end{cases}$$



Le principe de minimisation du risque structurel

Exemple de minimisation du risque structurel : les SVM

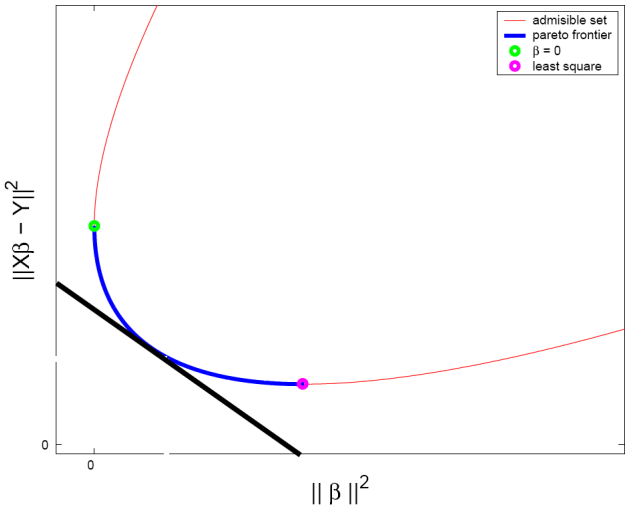
$C(\mathcal{F}) = \|\mathbf{f}\|^2$ et $\mathcal{F} = \mathcal{H}$ un RKHS

L'apprentissage comme un problème d'estimation fonctionnel

$$\begin{cases} \min\limits_{f \in \mathcal{H}} \quad \hat{R}_n(f) = \dfrac{1}{n} \sum\limits_{i=1}^{n} \max(0, 1 - y_i(f(x_i) + b)) & \text{attache aux données} \\ \min\limits_{f \in \mathcal{H}} \quad C(\mathcal{F}) = \|\mathbf{f}\|^2 & \text{mesure de la complexité} \end{cases}$$

3 formulations equivalentes

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i) + \lambda \|f\|^2$$

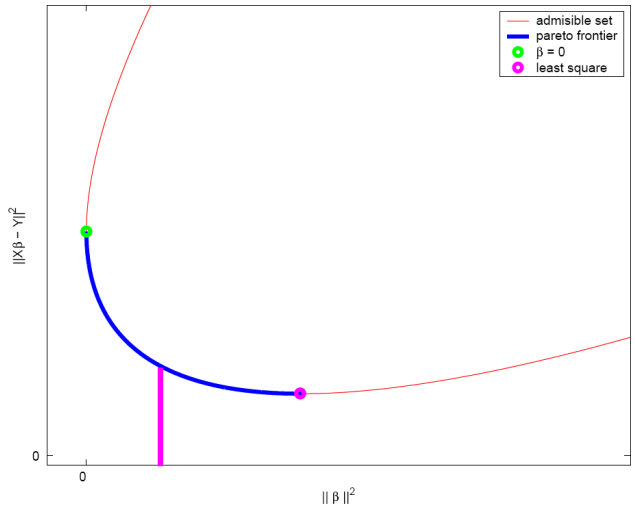


grâce à la convexité

3 formulations equivalentes

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i) + \lambda \|f\|^2$$

$$\begin{cases} \min\limits_{f} \dfrac{1}{n} \sum\limits_{i=1}^{n} L(f(x_i), y_i) \\ \text{avec } \|f\|^2 \leq k \end{cases}$$

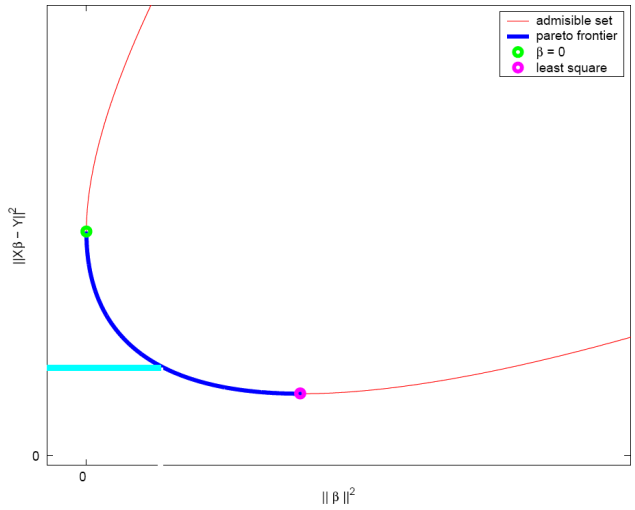


grâce à la convexité

3 formulations equivalentes

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i) + \lambda \|f\|^2$$

$$\begin{cases} \min\limits_{f} \frac{1}{n} \sum\limits_{i=1}^{n} L(f(x_i), y_i) \\ \text{avec } \|f\|^2 \leq k \end{cases}$$



$$\begin{cases} \min\limits_{f} \|f\|^2 \\ \text{avec } \frac{1}{n} \sum\limits_{i=1}^{n} L(f(x_i), y_i) \leq k' \end{cases}$$

grâce à la convexité

C'est un programme quadratique (QP)

Primal Dual

Primal

$$\begin{cases} \min\limits_{f,b,\xi\in\mathbb{R}^n} & \frac{1}{2}\|f\|^2 + C\sum\limits_{i=1}^{n}\xi_i \\ \text{avec} & y_i(f(\mathbf{x}_i)+b) \geq 1-\xi_i \\ & \xi_i \geq 0 \quad i=1,n \end{cases}$$

- QP à $\#f+n+1$ inconnues
- et $2n$ contraintes

Dual

$$\begin{cases} \min\limits_{\alpha\in\mathbb{R}^n} & \frac{1}{2}\alpha^\top G\alpha - \mathbf{e}^\top\alpha \\ \text{avec} & \mathbf{y}^\top\alpha = 0 \\ \text{ert} & 0 \leq \alpha_i \leq C \qquad i=1,n \end{cases}$$

- QP à n inconnues
- G matrice de Gram $\mathcal{O}(pn^2)$
- $2n$ contraintes de boite
- complexité empirique $\sim \mathcal{O}(n^{1.5})$

$$\sum_{j=1}^{d} f(x_j) + b = \sum_{i=1}^{n} \alpha_i\, y_i(\mathbf{x}^\top\mathbf{x}_i) + b$$

Primal Dual

Primal

$$\begin{cases} \min\limits_{f,b,\xi\in\mathbb{R}^n} & \frac{1}{2}\|f\|^2 + C\sum\limits_{i=1}^{n}\xi_i \\ \text{avec} & y_i(f(\mathbf{x}_i)+b) \geq 1-\xi_i \\ & \xi_i \geq 0 \quad i=1,n \end{cases}$$

- QP à $\#f+n+1$ inconnues
- et $2n$ contraintes

Dual

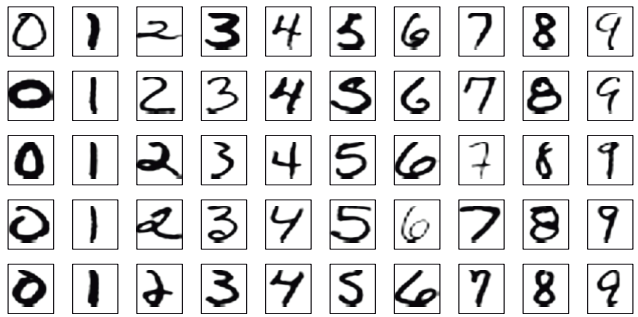
$$\begin{cases} \min\limits_{\alpha\in\mathbb{R}^n} & \frac{1}{2}\alpha^\top G\alpha - \mathbf{e}^\top\alpha \\ \text{avec} & \mathbf{y}^\top\alpha = 0 \\ \text{ert} & 0 \leq \alpha_i \leq C \qquad i=1,n \end{cases}$$

- QP à n inconnues
- G matrice de Gram $\mathcal{O}(pn^2)$
- $2n$ contraintes de boite
- complexité empirique $\sim \mathcal{O}(n^{1.5})$

$$\sum_{j=1}^{d} f(x_j) + b = \sum_{i=1}^{n} \alpha_i\, y_i(\mathbf{x}^\top\mathbf{x}_i) + b$$

Les résultats sur MNIST

- MNIST[1], data = « image-label »
- $n = 60\,000$; $p = 700$; k classes $= 10$
- Kernel error rate (2002) $= 0.56$ %,
- Best error rate (2012) $= 0.23$ % .



Les SVM sélectionnent les individus mais :

n ne doit pas être trop grand

p grand et n grand, donc on repasse dans le primal

[1] http://yann.lecun.com/exdb/mnist/index.html

Une autre approche quand p est grand

Retour au modèle linéaire :

$$f(\mathbf{x}_i) = \mathrm{w}^\top \mathbf{x}_i$$

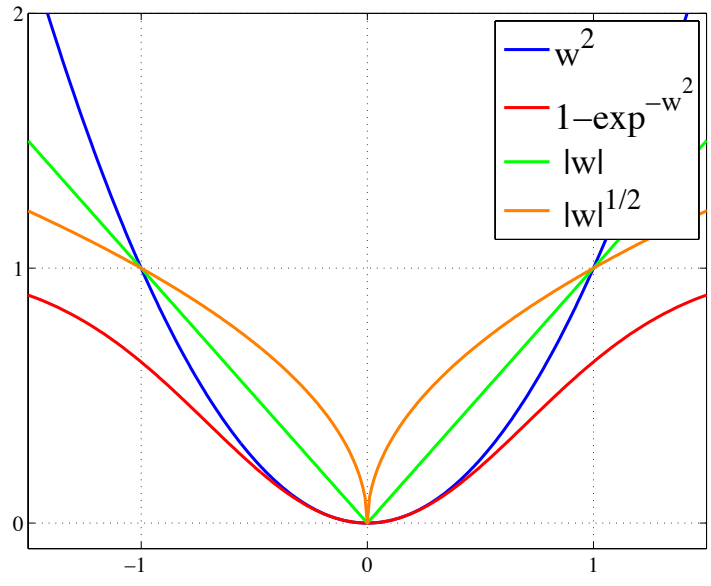
Dans le primal : sélection de variable

$$\begin{cases} \min\limits_{f \in \mathbb{R}^p} & \hat{R}_n(f) = \dfrac{1}{n} \sum\limits_{i=1}^{n} L(\mathrm{w}^\top \mathbf{x}_i), y_i) & \text{attache aux données} \\ & C(\mathbb{R}^p) = \|\mathrm{w}\|_0 \leq k & \text{nombre de variables } k < p \end{cases}$$

Comme pour le cout 0/1, on utilise une autre pénalité (relaxation)

Exemples de différentes pénalités utilisées en apprentissage

Pénalité	Convexe	Diff.
$\omega(w) = w^2$ $\Omega(\mathbf{x}) = \|\mathbf{x}\|^2$	✓	✓
$\omega(w) = 1 - \exp(-w^2)$	✗	✓
$\omega(w) = \|w\|$ $\Omega(\mathbf{x}) = \|\mathbf{x}\|_1$	✓	✗
$\omega(w) = \sqrt{\|w\|}$ $\Omega(\mathbf{x}) = \|\mathbf{x}\|_{1/2}$	✗	✗



Gradient proximal pour gérer la non différentiabilité

selon la différentiabilité des critères

$$\min_f L(f(x_i), y_y) + \lambda\Omega(f)$$

L et Ω différentiables -> gradient

$$f^{(t+1)} = f^{(t+1)} - \rho\nabla(L(f(x_i), y_y) + \lambda\Omega(f))$$

Ω non différentiable -> gradient proximal une méthode du premier ordre pour résoudre certains programme quadratiques

$$\begin{aligned} g^{(t+1)} &= f^{(t+1)} - \rho\nabla(L(f(x_i), y_y)) \\ f^{(t+1)} &= prox_{\rho\lambda\Omega}(g^{(t+1)}) \end{aligned}$$

Proximal algorithms N Parikh, S Boyd - Foundations and Trends in Optimization, 2014 (avec les travaux de J. Fadili)

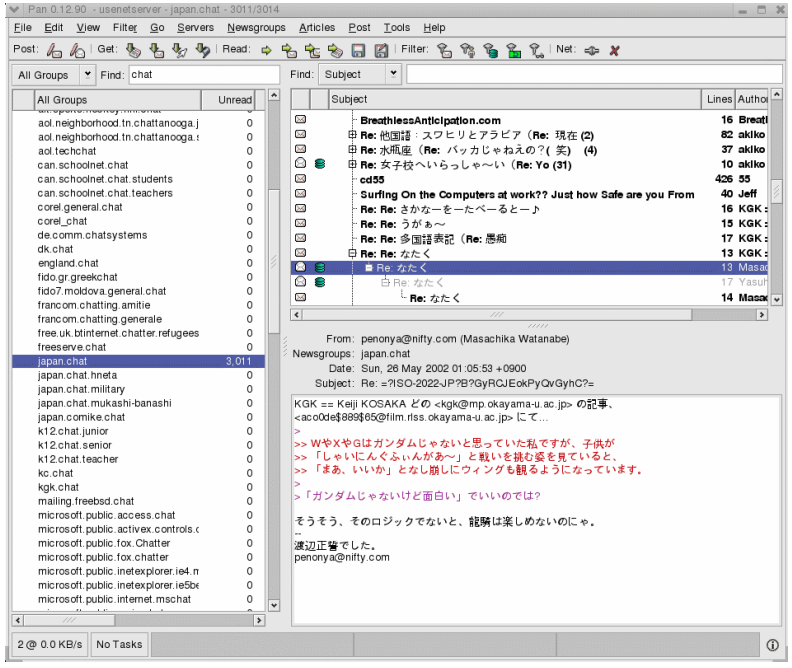
Newsgroup

données
- observations : textes
- étiquettes $y_i \in \{1; \ldots, 20\}$

individus $n = 11\,314$

variables $p = 4.8$ million

classes $k = 20 \rightarrow 2$



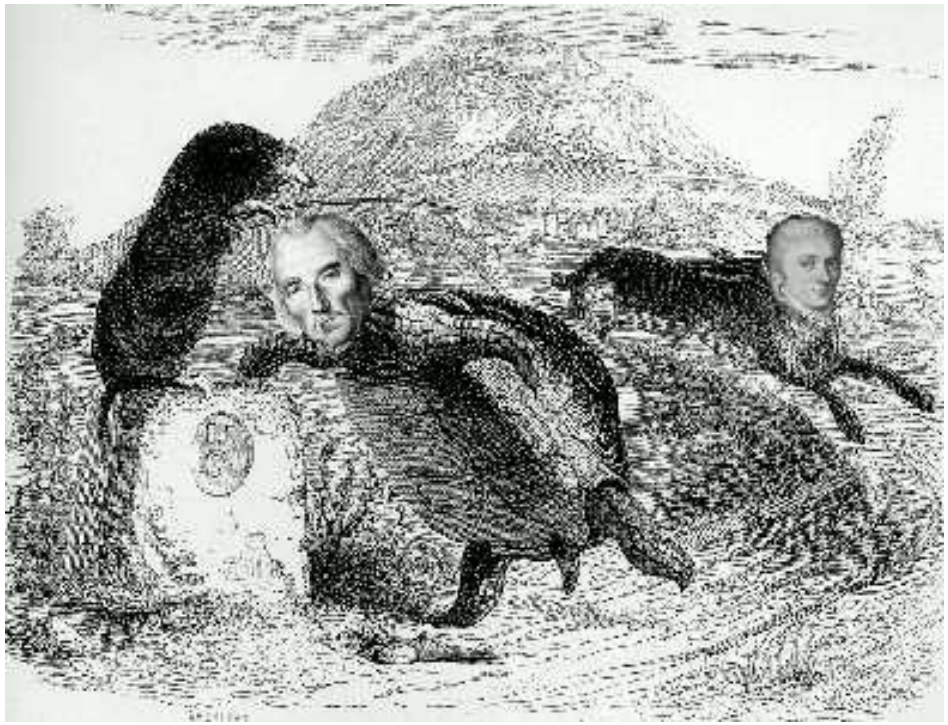
modèle linéaire, cout logistique, pénalité L1

- computing time (2010) = 2 minutes
- Best error rate = 2.73 % .

An interior-point method for large-scale l1-regularized logistic regression K Koh, SJ Kim, S Boyd - JMLR, 2007

Le lièvre gaussien et la tortue laplacienne

Loi de Gauss $\|w\|_2$ Loi de Laplace $\|w\|_1$



Gauss p^3 vs Laplace (LP, QP) $n^2 \times k^2$, k nombre de variables utiles

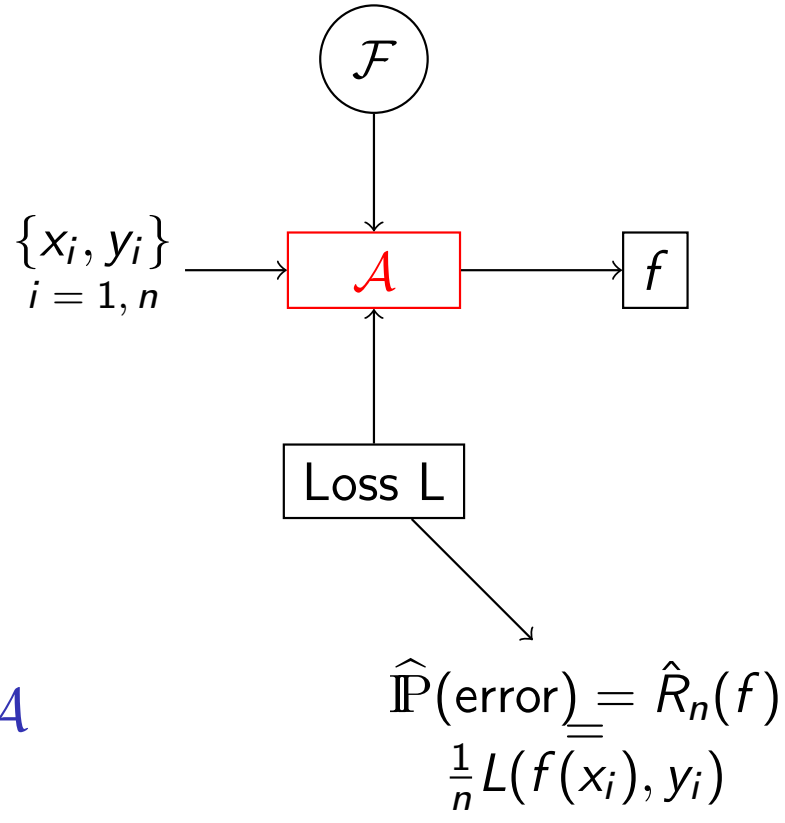
Problème : comment gérer

pour n et p grand.

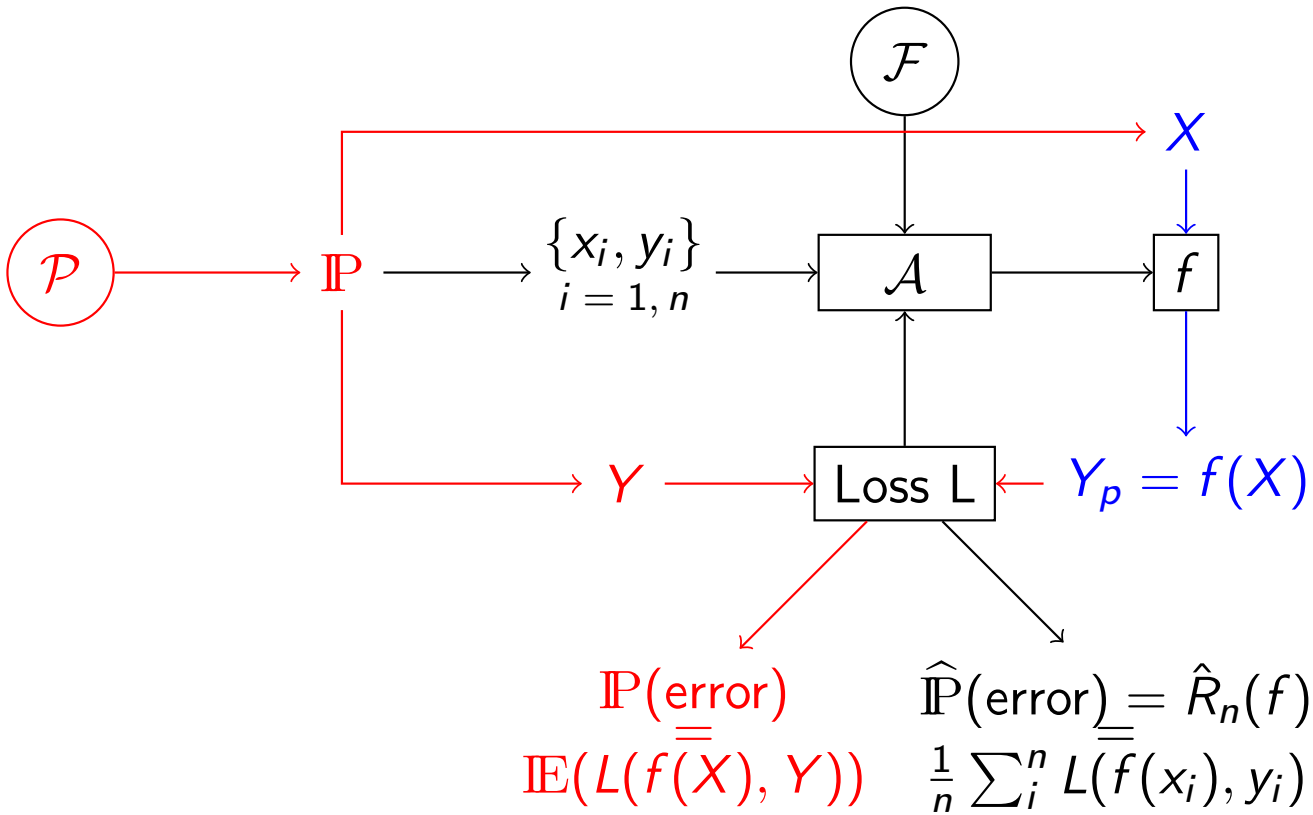
The Gaussian hare and the Laplacian tortoise: computability of squared-error versus absolute-error estimators S Portnoy, R Koenker - Statistical Science, 1997

Plan de l'exposé

1. Apprendre a partir des données $\{x_i, y_i\}$
2. le Cout L : premier algorithme MRE
3. Régularisation et Pénalités : MRS et $\mathcal{F}$
4. Retour sur l'algorithme de minimisation $\mathcal{A}$
5. L'apprentissage à grande échelle : *deep learning*
6. Conclusion et perspectives



Quel algorithme choisir lorsque n et p sont grands



$$\mathbb{P}(|R(\hat{f}) - R(f^\star)| \leq \varepsilon) \geq \delta$$

L'erreur statistique est de l'ordre de $\quad \varepsilon(n) \propto \mathcal{O}\sqrt{\frac{1}{n}} \quad \Rightarrow \quad n \propto \frac{1}{\varepsilon^2}$

Quel algorithme choisir lorsque n et p sont grands

L'erreur statistique est de l'ordre de $n \propto \frac{1}{\varepsilon^2}$

Principe : l'erreur statistique $\approx$ l'erreur algorithmique

méthodes	Gradient stochastique	1er ordre Gradient	2nd ordre Q Newton	QP dual
pour une itération	p	np	$p(n+p)$	
nombre d'itération	$\frac{\kappa\gamma}{\varepsilon}$	$\kappa \log \frac{1}{\varepsilon}$	$\log\log \frac{1}{\varepsilon}$	
temps pour ε	$\frac{\kappa\gamma p}{\varepsilon}$	$\kappa np \log \frac{1}{\varepsilon}$	$pn \log\log \frac{1}{\varepsilon}$	n^2
temps pour ε				

Pour une comparaison détaillée voir pr exemple `http://leon.bottou.org/projects/sgd`

Quel algorithme choisir lorsque n et p sont grands

L'erreur statistique est de l'ordre de $n \propto \frac{1}{\varepsilon^2}$

Principe : l'erreur statistique $\approx$ l'erreur algorithmique

méthodes	Gradient stochastique	1er ordre Gradient	2nd ordre Q Newton	QP dual
pour une itération	p	np	$p(n+p)$	
nombre d'itération	$\frac{\kappa\gamma}{\varepsilon}$	$\kappa \log \frac{1}{\varepsilon}$	$\log\log \frac{1}{\varepsilon}$	
temps pour ε	$\frac{\kappa\gamma p}{\varepsilon}$	$\kappa np \log \frac{1}{\varepsilon}$	$pn \log\log \frac{1}{\varepsilon}$	n^2
temps pour ε	$\frac{\kappa\gamma p}{\varepsilon}$	$\kappa \frac{p}{\varepsilon^2} \log \frac{1}{\varepsilon}$	$\frac{p}{\varepsilon^2} \log\log \frac{1}{\varepsilon}$	$\frac{1}{\varepsilon^4}$

L'algorithme le plus rapide est

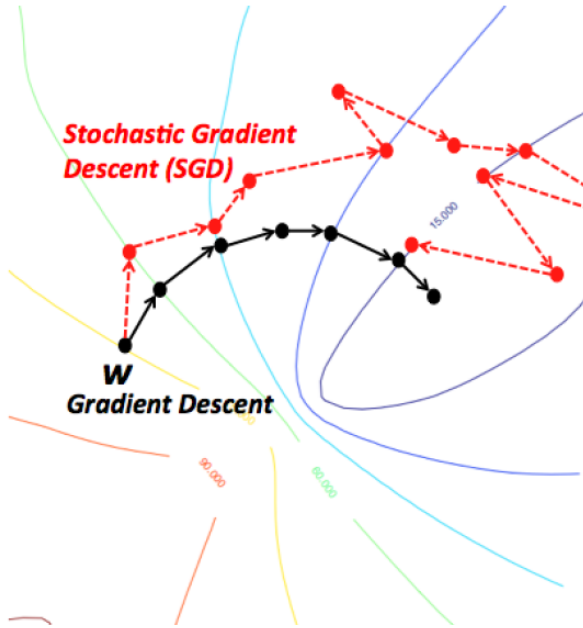
le gradient stochastique

Pour une comparaison détaillée voir pr exemple `http://leon.bottou.org/projects/sgd`

Le beurre et l'argent du beurre

- Accélération de la convergence
  - méthode de Nesterov (1983)
  - momentum (heuristique)

- Moyenner
  - les exemples (mini batch)
  - les paramètres (Polyak and Juditsky, 1992)
  - les gradients (stochastic average gradient (SAG-A), Le Roux et al 2012)
  - reduction de la variance (Johnson, Zhang, 13)



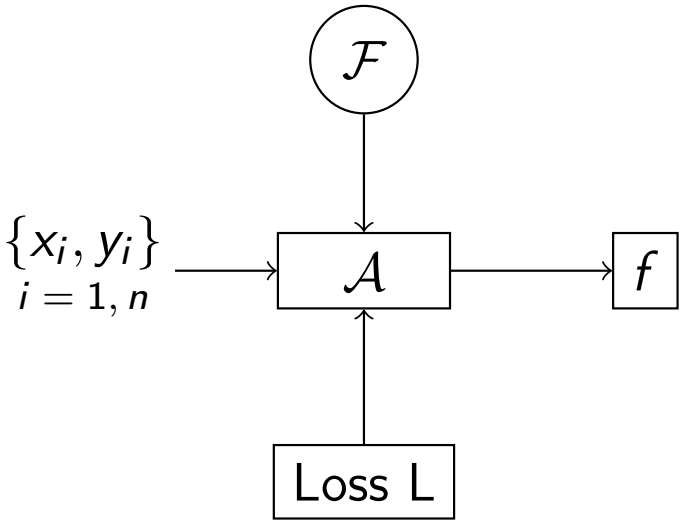
- Moyenner et accélérer
  - (Dieuleveut, Flammarion & Bach, 2016)

- Adapter les pas
  - Adaptive Moment Estimation – ADAM (Kingma & Ba, 2015)

http://ruder.io/optimizing-gradient-descent/ https://wikidocs.net/3413

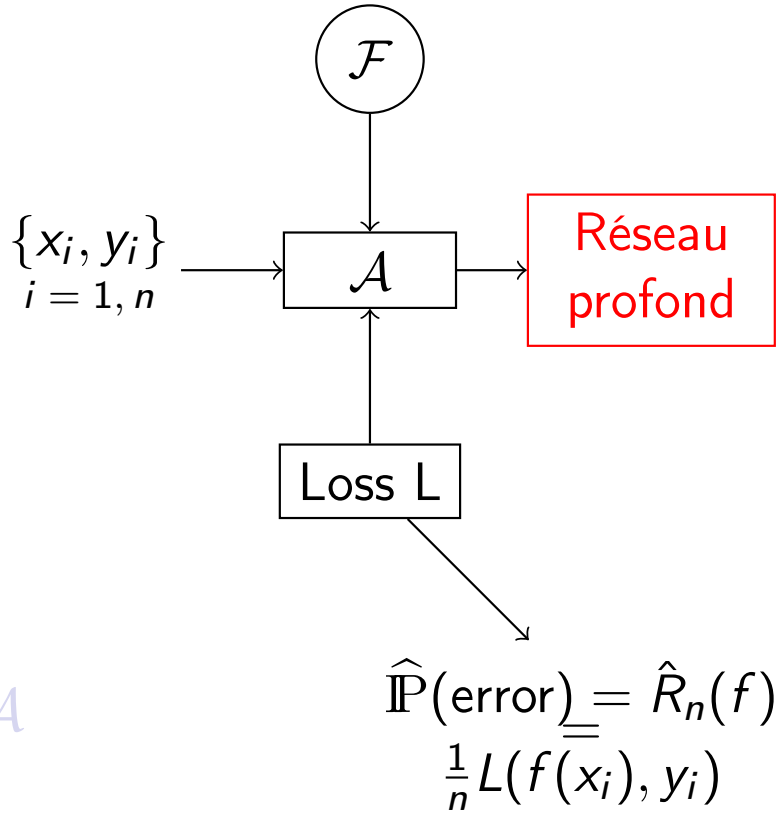
Revenir au non linéaire avec n et p grands

- cout L
- regulariser $C(\mathcal{F})$
- gradient stochastique (++) $\mathcal{A}$
- Modèle non linéaire à taille fixe : apprendre les caractéristiques f



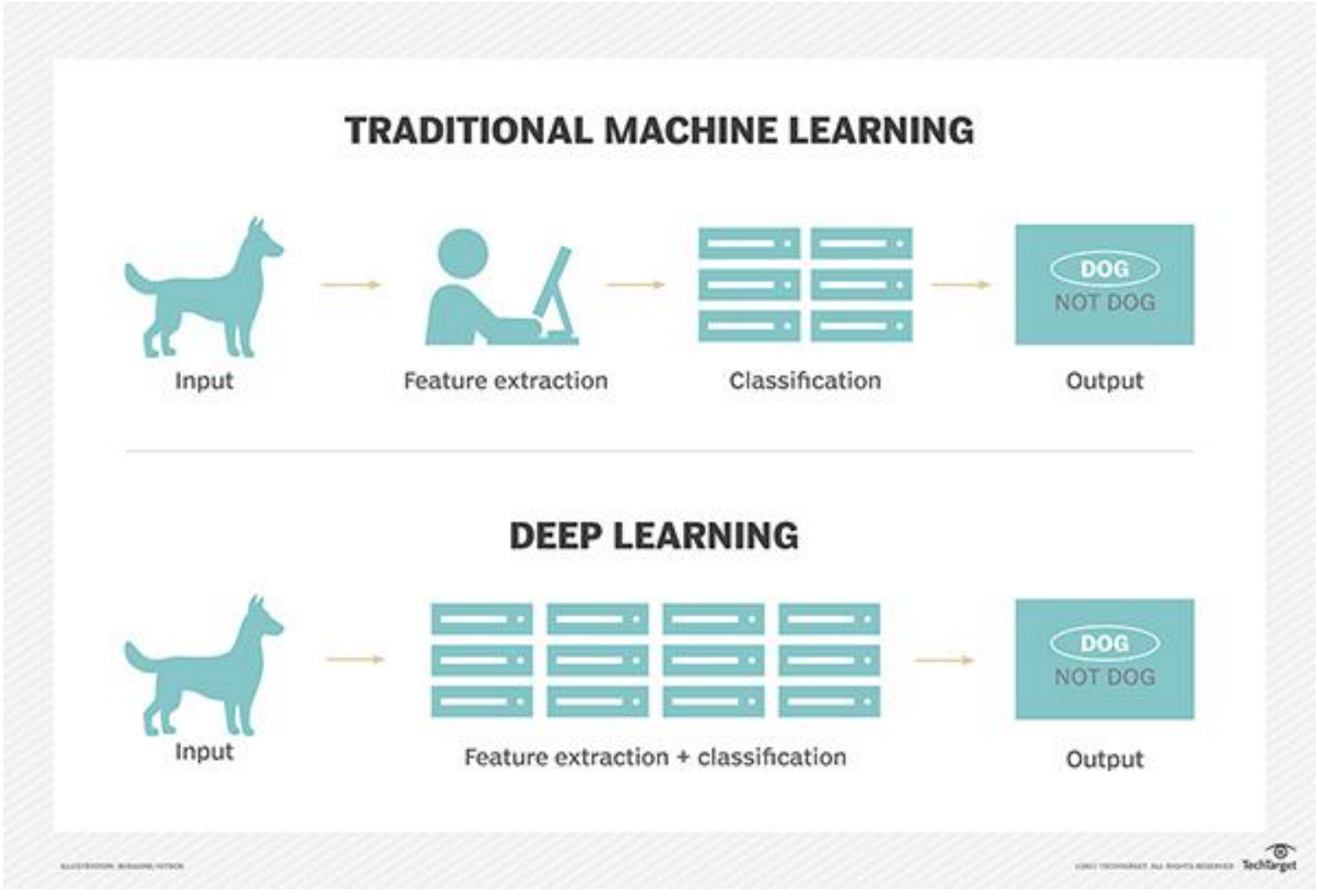
Plan de l'exposé

1. Apprendre a partir des données $\{x_i, y_i\}$
2. le Cout L : premier algorithme MRE
3. Régularisation et Pénalités : MRS et $\mathcal{F}$
4. Retour sur l'algorithme de minimisation $\mathcal{A}$
5. L'apprentissage à grande échelle : *deep learning*
6. Conclusion et perspectives



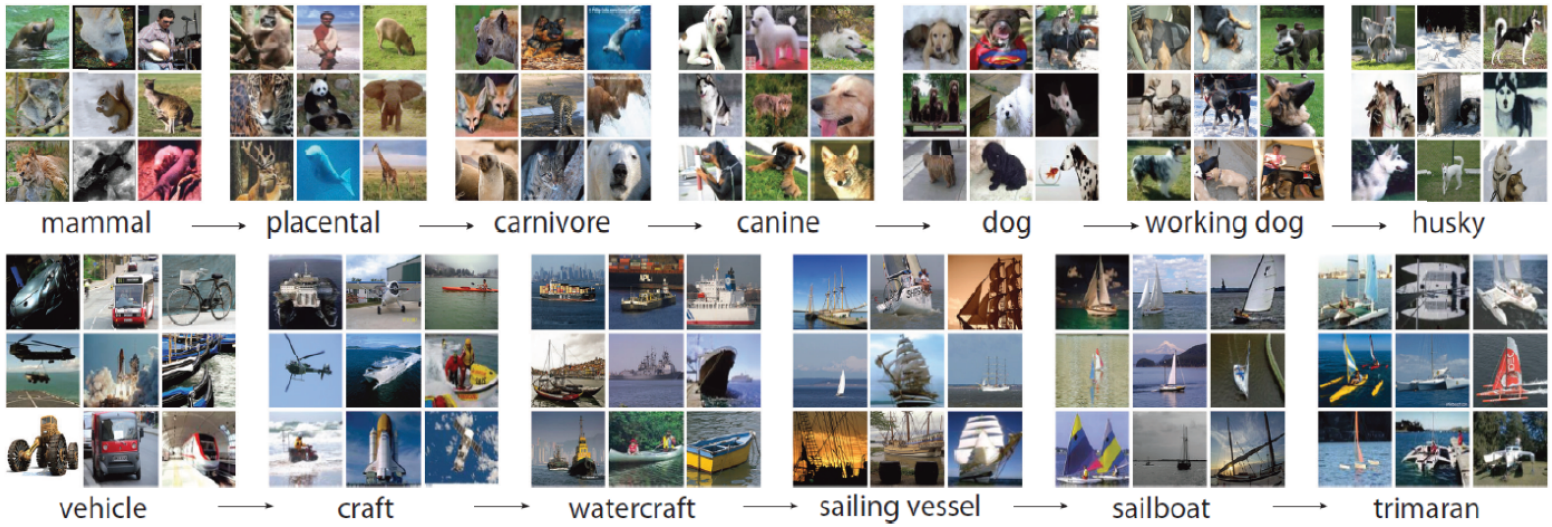
Que faire quand n et p et k sont grands

Apprendre les caractéristiques



http://searchbusinessanalytics.techtarget.com/definition/deep-learning

The image net database (Deng et al., 2012)



ImageNet = 15 million labeled high-resolution images of 22,000 categories.
Large-Scale Visual Recognition Challenge (a subset of ImageNet)

- 1000 categories.
- 1.2 million training images,
- 50,000 validation images,
- 150,000 testing images.

www.image-net.org/challenges/LSVRC/2012/

A new fashion in image processing

2012 Teams	%error
Supervision (Toronto)	15.3
ISI (Tokyo)	26.1
VGG (Oxford)	26.9
XRCE/INRIA	27.0
UvA (Amsterdam)	29.6
INRIA/LEAR	33.4

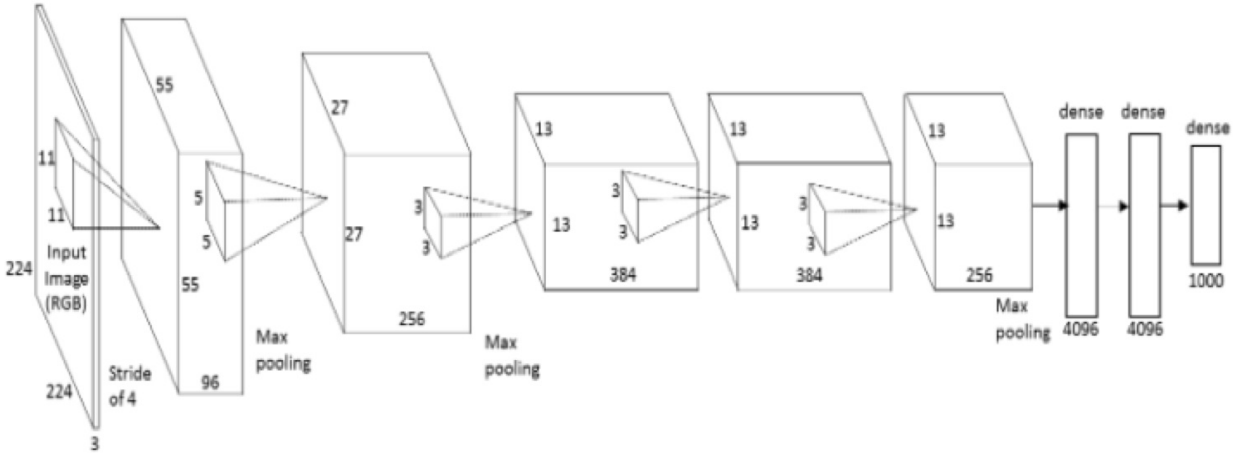
2013 Teams	%error
Clarifai (NYU spinoff)	11.7
NUS (singapore)	12.9
Zeiler-Fergus (NYU)	13.5
A. Howard	13.5
OverFeat (NYU)	14.1
UvA (Amsterdam)	14.2
Adobe	15.2
VGG (Oxford)	15.2
VGG (Oxford)	23.0

2014 Teams	%error
GoogLeNet	6.6
VGG (Oxford)	7.3
MSRA	8.0
A. Howard	8.1
DeeperVision	9.5
NUS-BST	9.7
TTIC-ECP	10.2
XYZ	11.2
UvA	12.1

shallow approaches

deep learning

Deep architecture and the image net (15%)

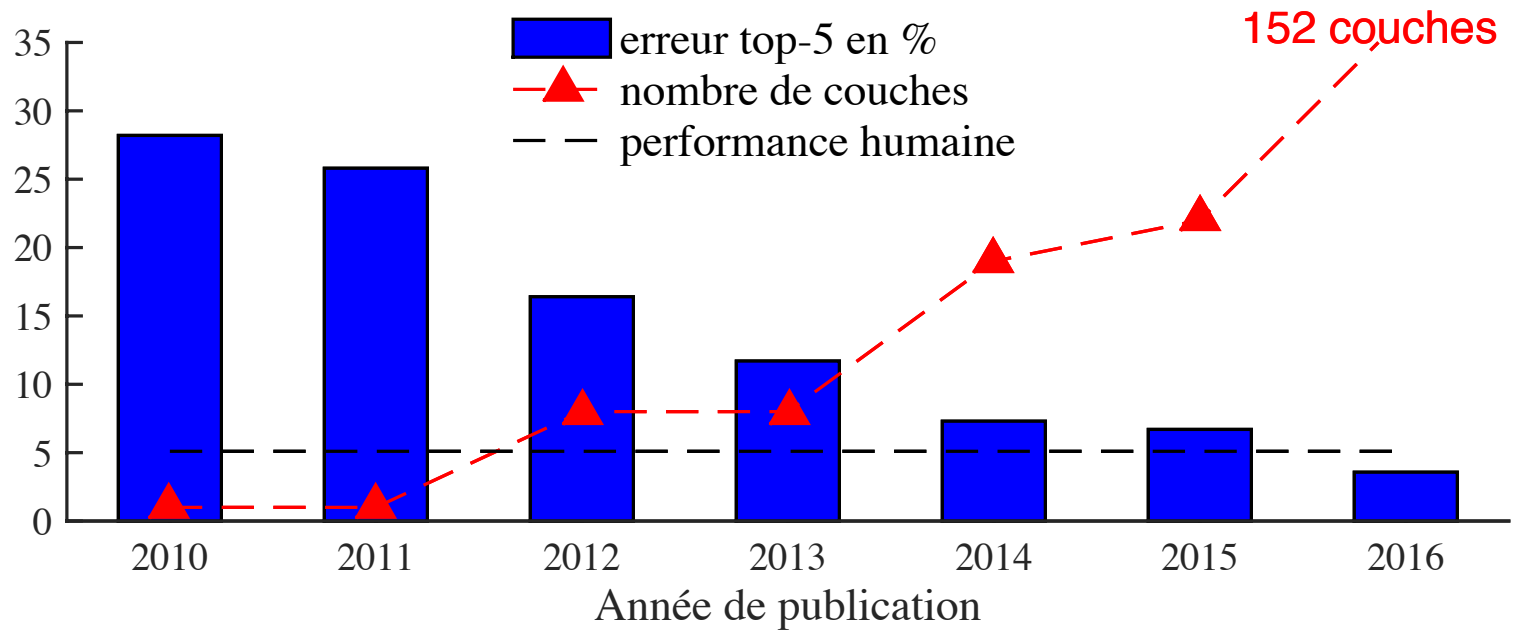


The *Alex Net* architecture [Krizhevsky, Sutskever, Hinton, 2012]

Convolution + Recitification (ReLU) + Normalization + Pooling

- 60 million parameters
- using 2 GPU – 6 days
- regularization
  - data augmentation
  - dropout
  - weight decay

ImageNet results



2012 Alex Net

2013 ZFNet

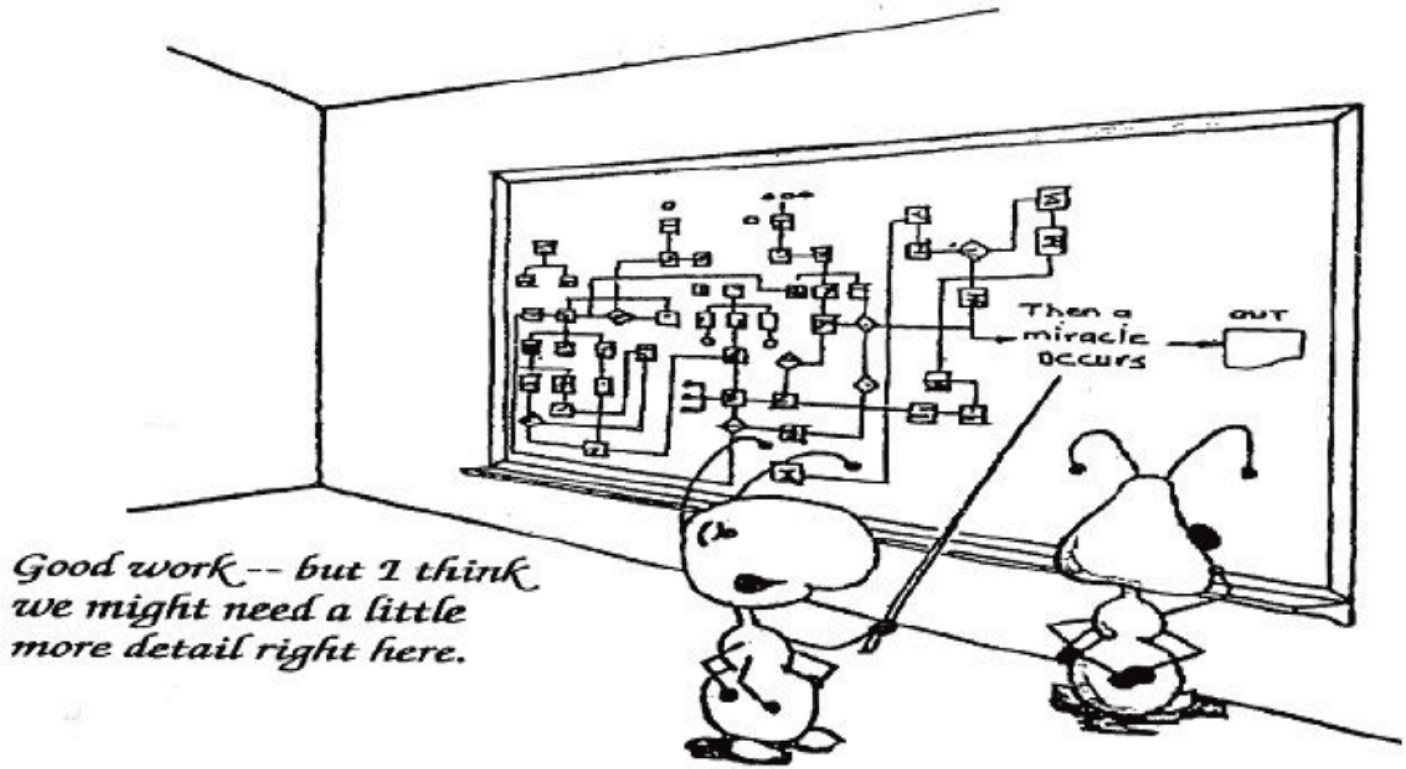
2014 VGG

2015 GoogLeNet / Inception

2016 Residual Network

karpathy's blog: karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/

Learning Deep architecture

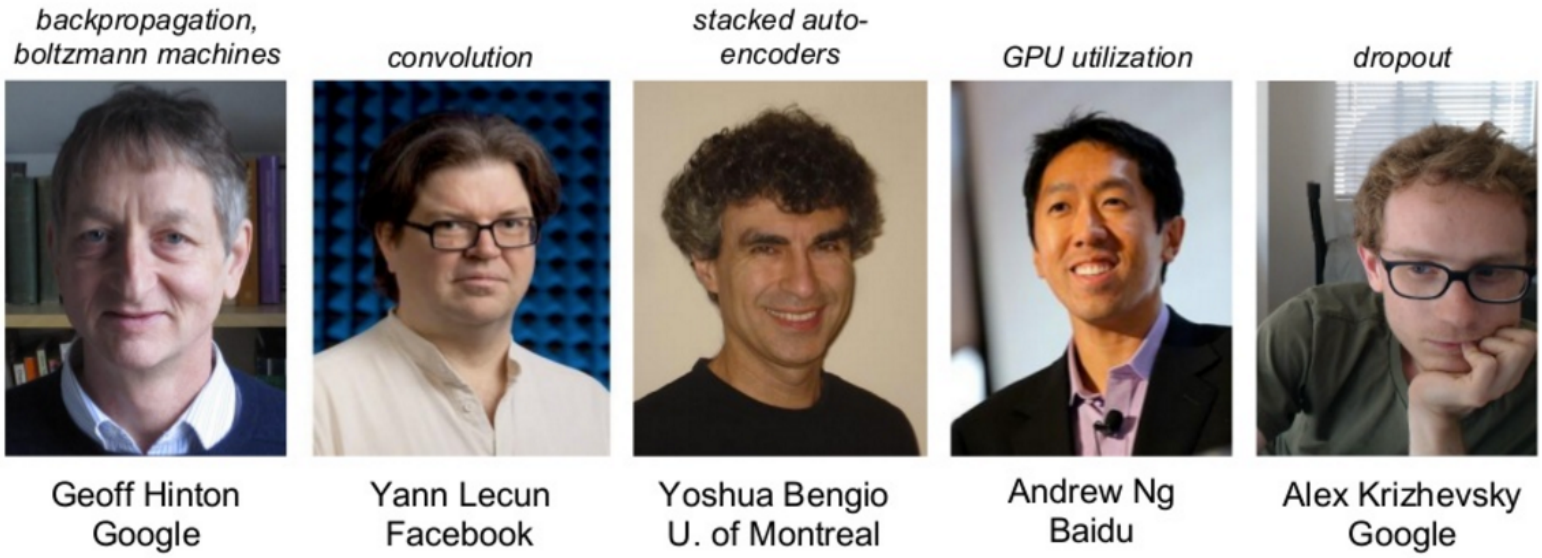


$$\min_{W \in \mathbb{R}^d} \sum_{i=1}^{n} \|f(x_i, W) - y_i\|^2 + \lambda \|w\|^2$$

- $d = 60 \times 10^6$
- $n = 1,200\,000$ ++
- $\lambda = 0.0005$

f is a deep NN

Deep learning and the industry



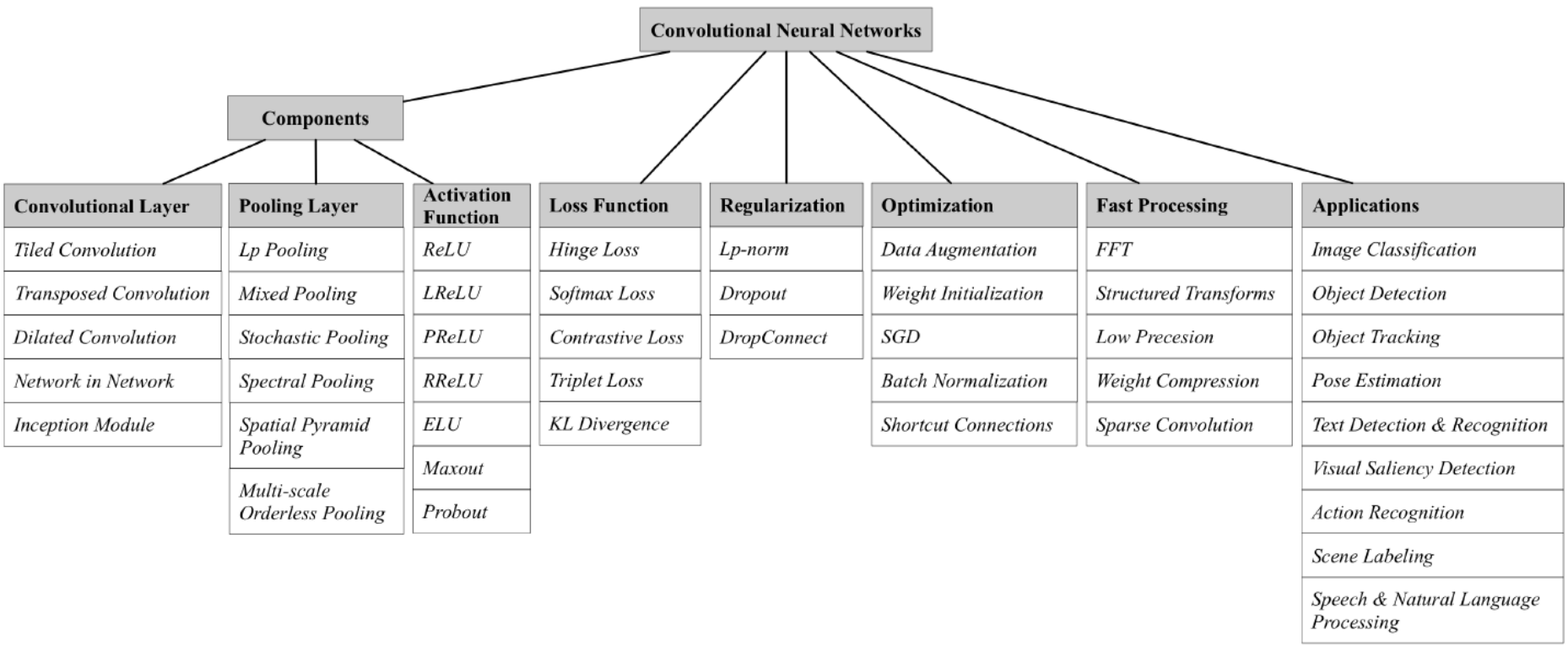
- data science, Artificial intelligence and deep learning
- les GAFA
  - they got the infrastructure (hard+software)
  - they got the data
  - deep learning bridges the gap between applications and ML

Deep Learning As A Service

a lot of available API (Google, Microsoft, Nvidia, Amazon...)

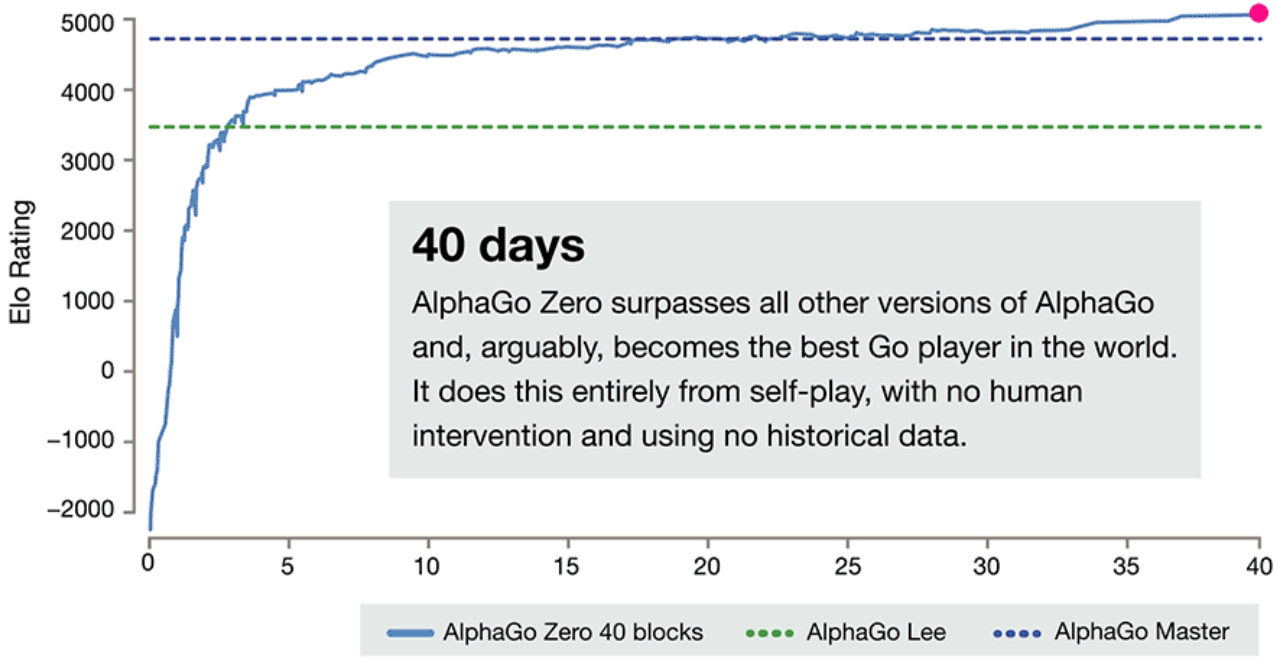
What's new with deep learning

- a lot of data (big data)
- big computing resources (hardware & software),
- big model (deep vs. shalow)
  - → new architectures
  - → new learning tricks



from Recent advances in convolutional neural networks Gu et al. Pattern Recognition, 2017

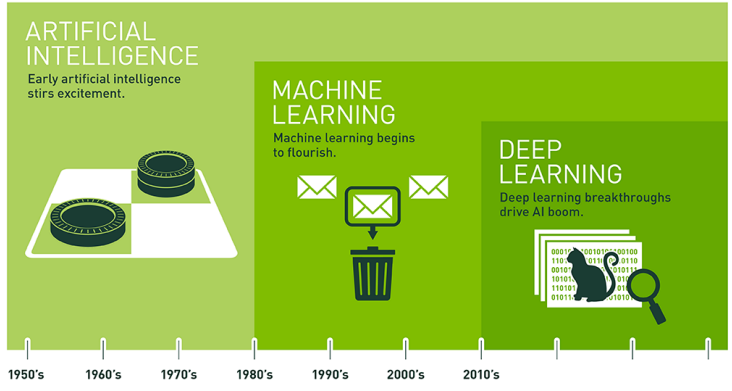
What's new with deep learning



from Mastering the game of Go without human knowledge D. Silver et al. Nature, 550, 2017

Plan de l'exposé

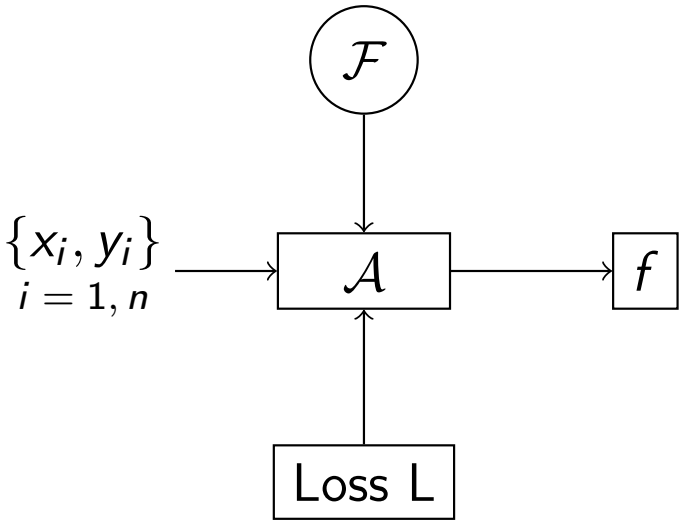
1. Apprendre a partir des données $\{x_i, y_i\}$
2. le Cout L : premier algorithme MRE
3. Régularisation et Pénalités : MRS et $\mathcal{F}$
4. Retour sur l'algorithme de minimisation $\mathcal{A}$
5. L'apprentissage à grande échelle : *deep learning*
6. Conclusion et perspectives



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Pendant ce temps au LITIS

- Applications $\{x_i, y_i\}$
  - BCI
  - Audio
  - détection de piétons...
- Noyaux $\mathcal{F}$
  - Boites à outil
  - MKL
  - Non positive kernels
- Sélection de variables p grand $\mathcal{A}$
  - dans les SVM
  - pénalités non convexes : DC, proximal
  - Modèles pour les distributions à symétrie elliptique



Conclusions

- Résumons nous
  - l'apprentissage : programmation à base d'exemples
  - optimisation grande taille
  - problème de représentation (caractéritiques)
  - optimisation non convexe et non différentiable
  - améliore certaines performances humaines
    - ★ perception bas niveau
  - on ne comprend pas bien pourquoi...

- le futur de l'apprentissage
  - apprentissage non supervisé : apprendre à raisonner
  - apprendre à apprendre
  - green learning
  - optimisation non convexe et non différentiable

Apprentissage en Normandie

Directions de recherche actuelles

- Apprentissage profond : image, audio et document (texte) $\mathcal{F}$
  - Normandeep (RIN 2017),
  - Deep in France (ANR 2016)
  - ICUB (ANR 2017) Imagerie Non Conventionnelle pour une Mobilité sécurisée en Milieu Urbain
- Transport optimal L
  - OATMIL (ANR 2017) Apprentissage statistique avec transport optimal
- Programmation mixte en nombre entiers (MIP) $\mathcal{A}$
  - ANR 2018 ???
- Conférence à Rouen : CAP 2018 en juin

Pour aller plus loin

- books
  - I. Goodfellow, Y. Bengio & A. Courville, *Deep Learning*, MIT Press book, 2016 `http://www.deeplearningbook.org/`
  - Gitbook `leonardoaraujosantos.gitbooks.io/artificial-inteligence/`
- conferences
  - NIPS, ICLR, xCML, AIStats,
- Journals
  - JMLR, Machine Learning, Foundations and Trends in Machine Learning, machine learning survey `http://www.mlsurveys.com/`
- lectures
  - Deep Learning: Course by Yann LeCun at Collège de France in 2016 `college-de-france.fr/site/en-yann-lecun/inaugural-lecture-2016-02-04-18h00.htm`
  - Convolutional Neural Networks for Visual Recognition (Stanford)
  - deep mind (`https://deepmind.com/blog/`)
  - CS 229: Machine Learning at stanford Andrew Ng
- Blogs
  - Andrej Karpathy blog (`http://karpathy.github.io/`)
  - `http://deeplearning.net/blog/`
  - `https://computervisionblog.wordpress.com/category/computer-vision/`