Numerical Probability: Stochastic Approximation from Finance to Data Science

Gilles Pagès

LPSM-Sorbonne-Université

(Labo. Proba., Stat. et Modélisation)

12th European Summer School in Financial Mathematics

Univ. of Padova

September 2-8, 2019

1. Optimization (deterministic, the origins)
2. Examples from Finance
   - Implicitation
   - Minimization
3. Learning procedures
   - Abstract Learning
   - Supervised Learning
   - Unsupervised Learning (clustering)
4. Stochastic algorithms/Approximation
   - From Robbins-Monro to Robbins-Siegmund
   - Stochastic Gradient Descent (*SGD*) and pseudo-*SGD*
5. Examples revisited by *SFD*
   - Numerical probability
   - Learning (supervised and unsupervised)
6. Application to Neural Networks and deep learning
   - Linear neural network
   - One hidden layer feedforward perceptron
   - Toward deep learning
   - Multilayer feedforward perceptron and Backpropagation

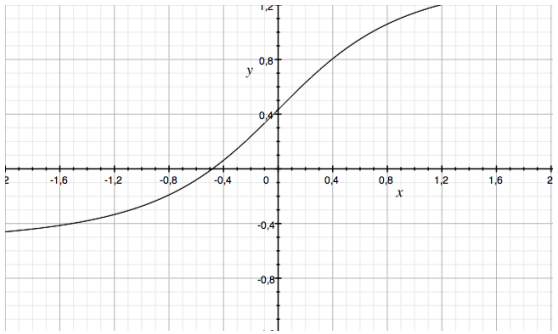
Table of Contents

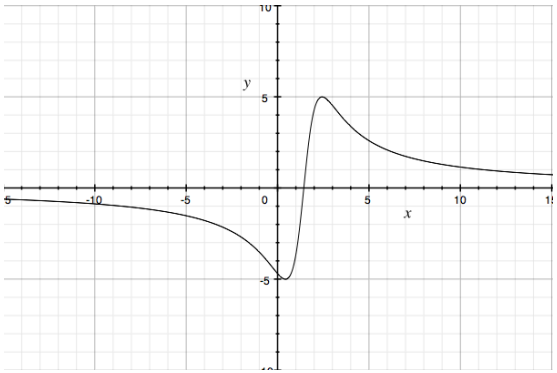
Deterministic zero search and optimization

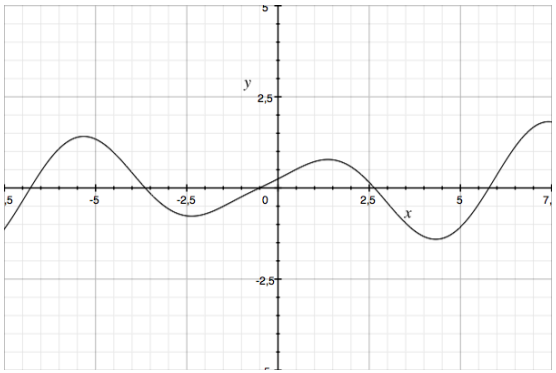
- Zero search: One aims at finding a zero θ^* of a function $h : \mathbb{R}^d \to \mathbb{R}^d$. In view of generic notations in stochastic approximation, we will denote

$$h(\theta),\ \theta \in \mathbb{R}^d$$

rather than $h(x)$.







($d = 1$ is mandatory just for graphs).

- Various methods:
  - Local recursive zero search (standard): θ_0 be fixed and let $\gamma > 0$ be small enough. Set

$$\theta_{n+1} = \theta_n - \gamma h(\theta_n), \quad n \geq 0$$

- Various methods (the sequel):
  - Local recursive zero search if h $\mathcal{C}^1$ (Newton-Raphson "false position" algoritm)

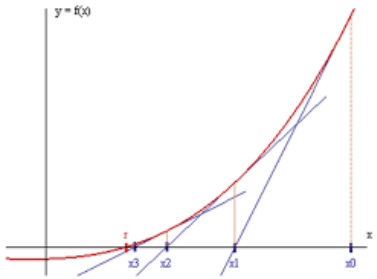
$$\theta_{n+1} = \theta_n - [J_h(\theta_n)]^{-1} h(\theta_n), \quad n \geq 0,$$

where $J_h(\theta)$ denotes the Jacobian of h at θ.

Idea: The tangent hyperplane is the best approximation of h (by an hyperplane)

$$h(\theta) \simeq h(\theta_n) + J_h(\theta_n)(\theta_n - \theta)$$

so θ_{n+1} is solution to $h(\theta_n) + J_h(\theta_n)(\theta_n - \theta) = 0$.



Very fast but also very unstable, especially when $J_h(\theta^*)$ is "small".

  - Yet another local recursive zero search if h $\mathcal{C}^1$ (Levenberg-Marquardt algorithm): Let $\lambda_n > 0$, $n \geq 1$,

$$\theta_{n+1} = \theta_n - \left[J_h(\theta_n) + \lambda_{n+1} I_d\right]^{-1} h(\theta_n), \quad n \geq 0.$$

turns out to be more stable.

- Global recursive zero search:

  – Idea: make the step decrease (not too fast) to "enlarge" in an adaptive way the convergence area of the algorithm...

  – Let γ_n, $n \geq 1$ satisfy

  $$\sum_n \gamma_n = +\infty \text{ and } \sum_n \gamma_n^2 < +\infty.$$

  – Set

  $$\theta_{n+1} = \theta_n - \gamma_{n+1} h(\theta_n), \ n \geq 0$$

- Etc.

- WARNING! All these methods require that

  h to be computed at a reasonable cost.

Minimizing a (potential function)

- Gradient descent (GD):
  Let $V : \mathbb{R}^d \to \mathbb{R}_+$, $\mathcal{C}^1$ with $\lim_{|x|\to+\infty} V(x) = +\infty$ so that $\operatorname{argmin} V \neq \varnothing$.

  How to compute argmin & $\min_{\mathbb{R}^d} V$???

  - If V is convex, then

    $$\operatorname{argmin} V = \{\nabla V = 0\} \quad \text{(is a convex set)}$$

    – Solution: set $h = \nabla V$,

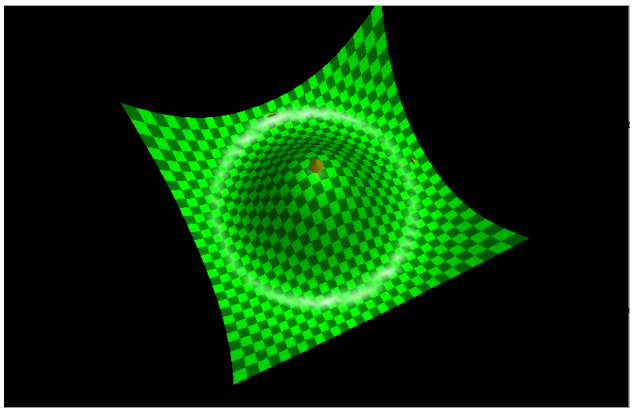
    – If ∇V Lipschitz, then (exercise)

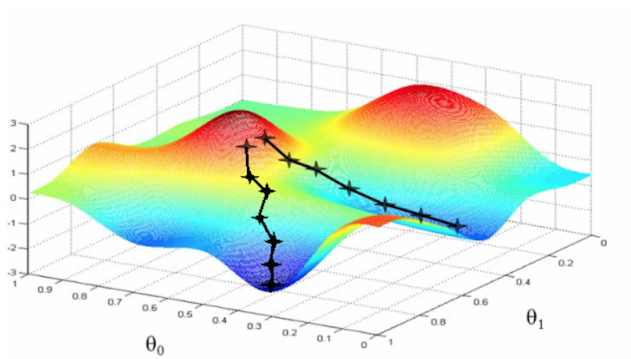
    $$\theta_n \to \theta^* \in \{\nabla V = 0\} = \operatorname{argmin} V \quad \text{as} \quad n \to +\infty.$$

- If V is not convex only

$$\operatorname{argmin} V \subsetneq \{\nabla V = 0\}.$$

Still set $h = \nabla V$ (what else?)





- Pseudo-gradient (back to zero search!):

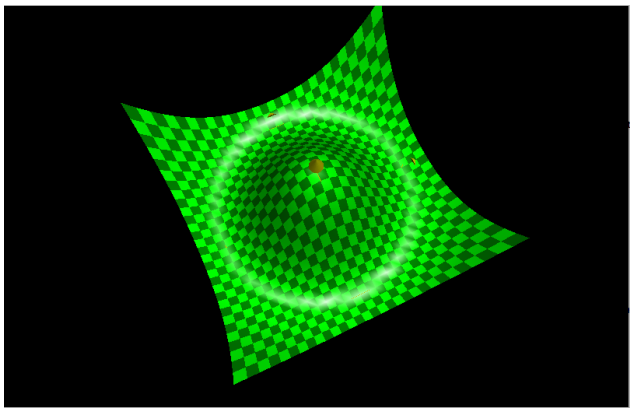
  The function h is often given (model) and (hopefully) there exists a Lyapunov function V s.t. $(h|\nabla V) \geq 0$ and

  $$\{h = 0\} \simeq \{(h|\nabla V) = 0\}.$$

  If $(d = 2)$, $\mathcal{H}(V)(x) = \begin{pmatrix} -\partial_{x_2} V \\ \partial_{x_1} V \end{pmatrix}$ (Hamiltonian of $\nabla V(x)$) and

  $$h(x) = \lambda \nabla V(x) + \mu \mathcal{H}(V)(x)$$

  then, the above conditions are satisfied and $|h|^2$ has V-linear growth so that $\theta_n \to C(0;1)$ (if $\theta_0 \neq 0$) but does not converge "pointwise".

  However, on this example, $V(\theta_n) \to \operatorname{argmin} V$.

- It may happen that $\{h = 0\} \neq \{(h|\nabla V) = 0\} \neq \{\nabla V = 0\} \neq \operatorname{argmin} V$!!.

Table of Contents

Table of Contents

Implicitation: Volatility

- Black-Scholes model: traded asset $X_t = x_0 e^{(r-\frac{\sigma^2}{2})t+\sigma W_t}$, x_0, volatility $\sigma > 0$, interest rate r, W standard Brownian motion.
- Call payoff $(X_T - K)_+ = \max(X_T - K, 0)$ with strike price K and maturity T.
- Mark-to-Market quoted price: $\mathrm{Call}_{M2Mkt} \in (0, x_0)$.
- Black-Scholes price at time 0

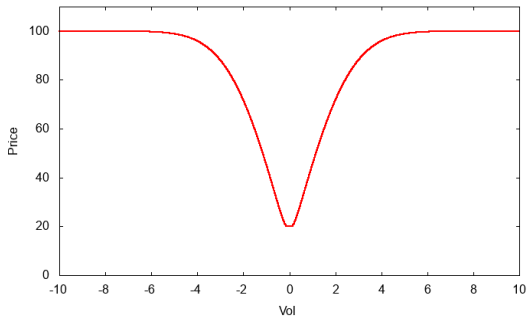
$$\begin{aligned}\mathrm{Call}_{BS}(x_0, K, r, \sigma) &= e^{-rT}\mathbb{E}\,(X_T - K)_+ \\ &= x_0\Phi_0(d_1) - Ke^{-rt}\Phi_0(d_2)\end{aligned}$$

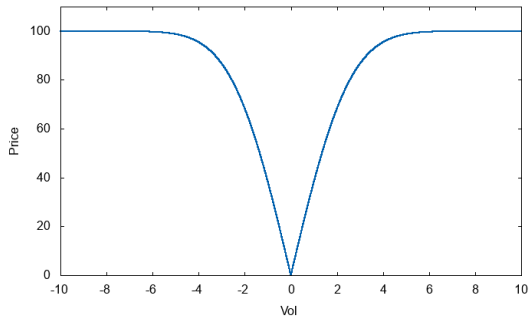
$$d_1 = \frac{\log(\frac{x_0}{K}) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T}.$$

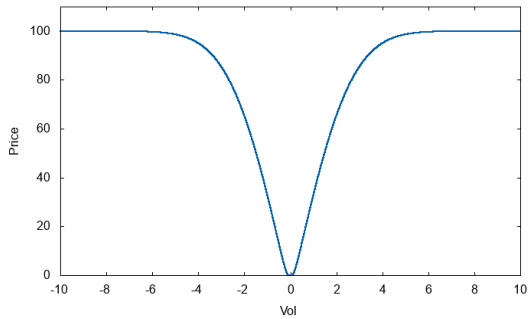
- Implicitation of the volatility: solve in σ the inverse problem

$$\mathrm{Call}_{BS}(\ldots, \sigma, \ldots) - \mathrm{Call}_{M2Mkt} = 0.$$

- Graphs of $\sigma \mapsto Call_{BS}(\sigma)$, $\sigma > 0$: In-, At and Out- the money.







- The function is even in σ and the equation has two opposite solutions.

- Algo$_1$:

$$\sigma_{n+1} = \sigma_n - \underbrace{\gamma_{n+1}\big(\mathrm{Call}_{BS}(\sigma_n) - \mathrm{Call}_{M2Mkt}\big)}_{=:h(\sigma_n)}, \quad \sigma_0 > 0.$$

with $\gamma_n = \gamma > 0$ or decreasing assumption.

- Algo$_2$:
  - The Vega:

$$\mathrm{Vega}_{BS}(\sigma) = x_0 \,\mathrm{sign}(\sigma)\sqrt{T}\,\frac{e^{-\frac{d_1(\sigma)^2}{2}}}{\sqrt{2\pi}}$$

  - Implicit volatility search reads:

$$\sigma_{n+1} = \sigma_n - \underbrace{\frac{\mathrm{Call}_{BS}(x_0, K, r, \sigma_n) - \mathrm{Call}_{M2Mkt}}{\mathrm{Vega}_{BS}(\sigma_n)}}_{=:h(\sigma_n)}, \quad \sigma_0 > 0.$$

[This is the actual algorithm with a "good choice" of σ_0]

Implicitation: Correlation I

- 2-dim (correlated) Black-Scholes model:

$$X_t^i = x_0^i e^{(r-\frac{\sigma_i^2}{2})t+\sigma_i W_t^i},\ x_0^i,\ \sigma_i > 0, i = 1, 2$$

with $\langle W^1, W^2 \rangle_t = \rho t$.

- Best-of-Call Payoff:

$$\left(\max(X_T^1, X_T^1) - K\right)_+$$

- Premium at time 0

$$\text{Best-of-Call}_{BS}(\dots, \rho, \dots) = e^{-rT}\mathbb{E}\left(\max(X_T^1, X_T^1) - K\right)_+.$$

- Organized markets on such options are market of the correlation ρ.
- The volatilities σ_i, $i = 1, 2$, are known from vanilla option markets on X^1 and X^2.

How to "extract" the correlation ρ?

- Deterministic algo(s):

$$\rho_{n+1} = \rho_n - \gamma_{n+1} \underbrace{\frac{\text{Best-of-Call}_{BS}(\rho_n) - \text{Best-of-Call}_{M2Mkt}}{\partial_\rho \text{Best-of-Call}_{BS}(\rho_n) + \lambda_n}}_{=:h(\rho_n)}.$$

  - Except that we have no (simple) closed form for the B-S price and its ρ-derivative
  - The correlation $\rho \in [-1, 1]$. Projections are possible but. . . .

- What to do?

Table of Contents

Minimization: Value-at-risk/Conditional Value-at-risk/I

- Let $X = \varphi(Z)$, $Z : (\Omega, \mathcal{A}, \mathbb{P}) \to \mathbb{R}^q$ be an integrable random variable representative of a loss and let $\alpha \in (0,1)$, $\alpha \simeq 1$.

$$\text{Value-at-Risk}_\alpha(X) = \alpha\text{-quantile} = \inf\left\{\xi : \mathbb{P}(X \le \xi) \ge \alpha\right\}.$$

- For simplicity, assume X has a density $f_x > 0$ on $\mathbb{R}$. Then $\xi_\alpha = \text{VaR}_\alpha(X)$ is the unique solution to

$$\mathbb{P}\big(X \le \xi_\alpha\big) = \alpha \Longleftrightarrow \mathbb{P}\big(X > \xi_\alpha\big) = 1 - \alpha.$$

- The conditional Value-at-Risk is defined by

$$\text{CVaR}_\alpha(X) = \mathbb{E}\big(X \,|\, X \ge \text{VaR}_\alpha(X)\big).$$

- Rockafellar-Uryasev Potential ([1]):

$$V(\xi) = \xi + \frac{1}{1-\alpha}\mathbb{E}\,(X - \xi)_+, \quad \xi \in \mathbb{R}.$$

[1] R.T. Rockafellar, S. Uryasev (2000). Optimization of Conditional Value-At-Risk, *The Journal of Risk*, **2**(3):21–41. `www.ise.ufl.edu/uryasev`.

- The function V is convex and $\lim_{|\xi|\to+\infty} V(\xi) = +\infty$ since

$$V(\xi) \geq \xi \quad \text{so that} \quad \lim_{\xi\to+\infty} V(\xi) = +\infty$$

and

$$\begin{aligned} V(\xi) &\geq \xi + \frac{1}{1-\alpha}(\mathbb{E}\, X - \xi)_+ \\ &= \xi + \frac{1}{1-\alpha}(\mathbb{E}\, X - \xi) \qquad \text{for } \xi \text{ low enough} \\ &= -\frac{\alpha}{1-\alpha}\xi + \frac{1}{1-\alpha}\mathbb{E}\, X \to +\infty \qquad \text{as} \quad \xi \to -\infty. \end{aligned}$$

- By differentiation under the $\mathbb{E}$, we get

$$V'(\xi) = 1 - \frac{1}{1-\alpha}\mathbb{P}(X > \xi).$$

- $V'(\xi) = 0$ iff $\mathbb{P}(X > \xi) = 1 - \alpha$ iff $\xi = \xi_\alpha$
- Moreover

$$V(\xi_\alpha) = \frac{\xi_\alpha + \mathbb{E}\,(X - \xi_\alpha)_+}{\mathbb{P}(X > \xi_\alpha)} = \frac{\mathbb{E}\,X\mathbf{1}_{\{X>\xi_\alpha\}}}{\mathbb{P}(X \geq \xi_\alpha)}$$
$$= \mathbb{E}\big(X \,|\, X \geq \mathrm{VaR}_\alpha(X)\big) = \mathrm{CVaR}_\alpha(X).$$

- (GD) pour la $\mathrm{VaR}_\alpha(X)$: $h(\xi) = V'(\xi)$. Let $\xi_0 \in \mathbb{R}$,

$$\xi_{n+1} = \xi_n - \gamma_{n+1}\big(1 - \frac{1}{1-\alpha}(1 - F_X(\xi_n))\big)$$
$$= \xi_n - \frac{\gamma_{n+1}}{1-\alpha}\big(F_X(\xi_n) - \alpha\big), \quad n \geq 0$$

- Newton/Levenberg-Marquardt algo: $\xi_0 \in \mathbb{R}$,

$$\xi_{n+1} = \xi_n - \frac{F_X(\xi_n) - \alpha}{(1-\alpha) f_X(\xi_n)}, \quad n \geq 0.$$

- Why not ! But $X = \varphi(Z)$ (the whole portfolio of a CIB Bank!) $\Rightarrow q$ large and no closed form for the c.d.f. $F_X(\xi) = \mathbb{P}(X \leq \xi)$ of X.

Table of Contents

Table of Contents

Abstract Learning

- Huge dataset $(z_k)_{k=1:N}$ with of possibly high dimension d: $N \simeq 10^6$, even 10^9, and $d \simeq 10^3$.
  [Image, profile, text, . . .]
- Set of parameters $\theta \in \Theta \subset \mathbb{R}^K$, K large (see later on).
- There exists a smooth local loss function/local predictor

$$v(\theta, z).$$

- Global loss function: $V(\theta) = \frac{1}{N} \sum_{k=1}^{N} v(\theta, z_k)$

  with gradient $\nabla V(\theta) = \frac{1}{N} \sum_{k=1}^{N} \nabla_\theta v(\theta, z_k)$.

- Solving the minimization problem

$$\min_{\theta\in\Theta} V(\theta).$$

- Suggests a (GD) i.e. $h = \nabla V$ [or others... if $\nabla^2_\theta v(\theta, z)$ exists]:

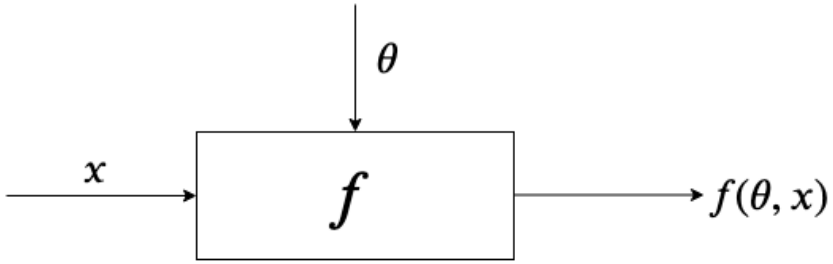
$$\begin{aligned}\theta_{n+1} &= \theta_n - \gamma_{n+1}\nabla V(\theta_n)\\ &= \theta_n - \frac{\gamma_{n+1}}{N}\sum_{k=1}^{N}\nabla_\theta v(\theta, z_k),\ n \geq 0,\end{aligned}$$

with the step sequence satisfying the (DS) assumption.

Table of Contents

Supervised learning

- Input x_k, output y_k. Data $z_k = (x_k, y_k) \in \mathbb{R}^{d_x+d_y}$, $k = 1 : n$.
- Transfer function $f : \Theta \times \mathbb{R}^{d_x} \longrightarrow \mathbb{R}^{d_y}$



- Prediction/loss function (local) $v(\theta, z) = \frac{1}{2}\left|f(\theta, x_k) - y_k\right|^2$, $k = 1 : n$ so that

$$\nabla_\theta v(\theta, z) = \nabla_\theta f(\theta, x)^\top \big(f(\theta, x) - y\big).$$

- Resulting loss function gradient

$$V(\theta) = \frac{1}{N}\sum_{k=1}^{N} \nabla_\theta f(\theta, x_k)^\top \big(f(\theta, x_k) - y_k\big).$$

Table of Contents

Unsupervised learning (clustering)

- Only input $z_k = x_k \in \mathbb{R}^d$, $k = 1 : N$.
- Prototype parameter set: $(\theta^1, \ldots, \theta^r) \in \Theta = (\mathbb{R}^d)^r$, $r \in \mathbb{N}$.
- Local loss function: $x \in \mathbb{R}^d$, $\theta \in \Theta$.

$$v(\theta, x) = \frac{1}{2} \min_{i=1:r} |\theta^i - x|^2 = \frac{1}{2} \mathrm{dist}\big(x, \{\theta^1, \ldots, \theta^r\}\big)^2$$

(minimal distance to prototypes).

- $v(\theta, x)$ is not convex in θ!
- Global loss function (Distortion):

$$V(\theta) = \frac{1}{2} \sum_{k=1}^{N} \min_{i=1:r} |\theta^i - x_k|^2 \quad \text{(mean minimal distance to prototypes)}.$$

Batch k-means/Forgy's algorithm

- Gradient at θ s.t. $\theta^i \neq \theta^j$: $\quad \nabla V(\theta) = \frac{1}{2}\sum_{k=1}^{N} \nabla_\theta v(\theta, x_k).$

  with, for $i = 1 : r$,

$$\partial_{\theta^i} v(\theta, x_k) = \left(\theta^i - x_k\right)\mathbf{1}_{\{|x_k-\theta^i|<\min_{j\neq i}|x_k-\theta^j|\}}.$$

- $\mathbf{1}_{\{|x_k-\theta^i|<\min_{j\neq i}|x_k-\theta^j|\}}$ = nearest neighbour search.
- Compute $\nabla V(\theta) = \frac{1}{N}\sum_{k=1}^{N} \nabla_\theta v(\theta, x_k)$
- $\Longrightarrow N\cdot$nearest neighbour searches among r ptotypes of dim d!
- Forgy's algorithm

$$\theta_{n+1} = \theta_n - \gamma_{n+1}\nabla V(\theta_n)$$

Table of Contents

Pros and cons: toward stochastic algorithm I

- Numerical Probability (for Finance): we do not know how to compute $h(\theta)$.
  - h always has a probabilistic presentation in our examples:

$$h(\theta) = \mathbb{E}\, H(\theta, Z) = \int_{\mathbb{R}^q} H(\theta, z)\mathbb{P}_Z(dz) = \int_{\mathbb{R}^q} H(\theta, z) f_Z(z) dz$$

    where $H : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^d$ is Borel, q often large...
  - Cons: ...which requires the computation of (often) high dimensional integrals on $\mathbb{R}^q$ at a reasonable cost: impossible.
  - Pros: The random vector Z can be simulated.
  - Pros: The function H is computable at a reasonable computational cost.
  - Pros: Regularizing effect of $\mathbb{E}$: h smoother than the functions $H(., z)$. (Think to $F_X(\xi) = \mathbb{E}\mathbf{1}_{\{X \leq \xi\}}$.)

Pros and Cons: toward stochastic algorithm I

- Data Science (usually V is given and $h = \nabla V$): but we cannot compute $h(\theta)$.
  - h still has probabilistic representation using the empirical measure.

$$h(\theta) = \frac{1}{N}\sum_{k=1}^{N} \nabla_\theta v(\theta, z_k) = \int_{\mathbb{R}^q} \nabla_\theta v(\theta, z_k)\mu_N(dz) \text{ with } \mu_N = \frac{1}{N}\sum_{k=1}^{N} \delta_{z_k}$$

  - Cons: But N huge $\Longrightarrow$ $h(\theta)$ cannot be computed at a reasonable cost.
  - Pros:

$$h(\theta) = \mathbb{E}\, \nabla_\theta v(\theta, Z)$$

    where Z can be simulated by picking up a datum (uniformly) at random since

$$Z \sim z_I, \quad I \sim \mathcal{U}\big(\{1, \ldots, N\}\big).$$

  - $v(\theta, z)$ and $\nabla_\theta v(\theta, z)$ both computable hence V and $h = \nabla V$ too.
  - Cons: No regularizing effect of $\mathbb{E}$: smoothness of $[h = \nabla V]$= smoothness of $H(., z)$.
  - Cons: Transfer of convexity in θ from $v(\cdot, z)$ to V.

Toward stochastic algorithm II

- Zero search of $h(\theta) = \mathbb{E}\, H(\theta, Z)$ as above.
- Idea 1: Use Monte Carlo simulation

$$\theta_{n+1} = \theta_n - \gamma_{n+1} \widehat{h}_{M_{n+1}}(\theta_n)$$

$$\widehat{h}_{M_{n+1}}(\theta_n) = \frac{1}{M_n} \sum_{k=1}^{M_n} H(\theta_n, Z_k^{(n+1)}), \quad \left(Z_k^{(n+1)}\right)_{k,n} \text{ i.i.d.} \sim Z$$

- Idea 2: Robbins-Monro, 1951 ([2])

$$\text{Set} \quad \forall\, n \geq 1, \quad M_n = 1 \text{ !!}$$

- Idea 1.5: Mini-batch i.e. $M_n = M > 2$. Successful among practitioners.

[2] H. Robbins, S. Monro (1951). A stochastic approximation method, *Ann. Math. Stat.*, **22**:400–407.

Table of Contents

Robbins-Monro framework (1951)

▷ Pactitioner's corner: Replace $h(\theta_n)$ by a $H(\theta_n, Z_{n+1})$.

- Let $(\theta_n)_{n\geq 0}$ be a sequence of $\mathbb{R}^d$-valued random vectors recursively defined on $(\Omega, \mathcal{A}, \mathbb{P})$ by

$$\theta_{n+1} = \theta_n - \gamma_{n+1} H(\theta_n, Z_{n+1}), \quad \theta_0 \in L^2(\mathbb{P}, \mathcal{A})$$

with

(i) $(Z_n)_{n\geq 1}$ is i.i.d. $\sim Z$, independent of θ_0

(ii) $\|H(\theta, Z)\|_2 \leq C(1 + |\theta|)$ $\quad (\Rightarrow h$ linear growth$)$

(iii) $(\gamma_n)_{n\geq 1}$ is a $(0, +\infty)$-valued deterministic step sequence

so that $(\theta_n)_{n\geq 0}$ is $(\mathcal{F}_n)$-adapted with $\mathcal{F}_n = \sigma(\theta_0, Z_1, \ldots, Z_n)$. Then

$$\theta_{n+1} = \theta_n - \gamma_{n+1} h(\theta_n) + \gamma_{n+1} \Delta M_{n+1}, \ n \geq 0.$$

- The key is simply: $\Delta M_{n+1} = H(\theta_n, Z_{n+1}) - h(\theta_n)$ since

$$\mathbb{E}\big(H(\theta_n, Z_{n+1}) \,|\, \mathcal{F}_n\big) \underbrace{=}_{Z_{n+1} \perp\!\!\!\perp \mathcal{F}_n} \Big[\mathbb{E}\, H(\theta, Z_{n+1})\Big]_{|\theta=\theta_n} = h(\theta_n), \ n \geq 0.$$

Interpretations

- Idea 1 (Robbins-Monro 1951, Robbins-Siegmund 1971):
  - Perturbed zero search procedure with decreasing step for h.
  - The perturbation is a martingale increment.

- Idea 2 (Ljung, 1977): Perturbed Euler scheme with decreasing step of the ODE

$$\dot{\theta} = -h(\theta).$$

(not exploited here)

Idea 1: Robbins-Siegmund Lemma, 1971

Theorem (Robbins-Siegmund Lemma, 1971)

- *Lyapunov function:* $V : \mathbb{R}^d \to \mathbb{R}_+, \mathcal{C}^1$, $\lim_\infty V = +\infty$, ∇V *Lipschitz,* $|\nabla V|^2 \leq c(1+V)$ *and (mean-reversion)*
$$(\nabla V | h) \geq 0 \quad \text{and} \quad \|H(\theta, Z)\|_2 \leq C\sqrt{1+V(\theta)}.$$
- *Decreasing Step assumption (DS):* $\sum_n \gamma_n = +\infty$ *and* $\sum_n \gamma_n^2 < +\infty$.
- $V(\theta_0) \in L^1$.

Then

(*i*) $V(\theta_n) \overset{a.s.}{\to} V_\infty \in L^1$ *[*$\Rightarrow (\theta_n)_{n\geq 0}$ *pathwise bounded] and* L^1*-bounded.*

(*ii*) $\sum_n \gamma_n (\nabla V | h)(\theta_{n-1}) \in L^1$ *a.s. , hence* $< +\infty$ *a.s.*

(*iii*) $\sum_n |\Delta\theta_n|^2 < +\infty$ *a.s. (so that* $\theta_n - \theta_{n-1} \to 0$ *a.s.).*

(*iv*) $\sum_n \gamma_n \Delta M_n$ *converges a.s. and in* L^2.

- Note that V is sub-quadratic i.e. $V(\theta) \leq \kappa(1+|\theta|^2)$ and h is sunlinear.

Proof

- Set $\mathcal{F}_n := \sigma(\theta_0, Z_1, \ldots, Z_n)$, $n \geq 1$ and $\Delta\theta_n := \theta_n - \theta_{n-1}$, $n \geq 1$.
- There exists $\xi_{n+1} \in (\theta_n, \theta_{n+1})$ s.t.

$$
\begin{aligned}
V(\theta_{n+1}) &= V(\theta_n) + (\nabla V(\xi_{n+1})|\Delta\theta_{n+1}) \\
&\leq V(\theta_n) + (\nabla V(\theta_n)|\Delta\theta_{n+1}) + [\nabla V]_{\mathrm{Lip}}|\Delta\theta_{n+1}|^2 \\
&= V(\theta_n) - \gamma_{n+1}(\nabla V(\theta_n)|H(\theta_n, Z_{n+1})) \\
&\qquad + [\nabla V]_{\mathrm{Lip}}\gamma_{n+1}^2|H(\theta_n, Z_{n+1})|^2 \\
&= V(\theta_n) - \gamma_{n+1}(\nabla V(\theta_n)|h(\theta_n)) - \gamma_{n+1}(\nabla V(\theta_n)|\Delta M_{n+1}) \\
(\star) \qquad &\qquad + [\nabla V]_{\mathrm{Lip}}\gamma_{n+1}^2|H(\theta_n, Z_{n+1})|^2,
\end{aligned}
$$

where

$$\Delta M_{n+1} = H(\theta_n, Z_{n+1}) - h(\theta_n).$$

- Show by induction that $V(\theta_n) \in L^1(\mathbb{P})$, given that $V(\theta_0) \in L^1(\mathbb{P})$.
- Key:

$$\mathbb{E}\left|\left(\nabla V(\theta_n)|H(\theta_n, Z_{n+1})\right)\right| \leq \frac{1}{2}\left(\mathbb{E}\,|\nabla V(\theta_n)|^2 + \mathbb{E}\,|H(\theta_n, Z_{n+1})|^2\right).$$

- So that $(\Delta M_n)_{n\geq 1}$ is a sequence of L^2 $(\mathcal{F}_n)$-martingale increments satisfying

$$\mathbb{E}\left(|\Delta M_{n+1}|^2 \,|\, \mathcal{F}_n\right) \leq \mathbb{E}\left(|H(\theta_n, Z_{n+1})|^2 \,|\, \mathcal{F}_n\right) \leq C(1 + V(\theta_n)).$$

- Coming back to

$$V(\theta_{n+1}) \le V(\theta_n) - \gamma_{n+1}(\nabla V(\theta_n)|h(\theta_n)) - \gamma_{n+1}(\nabla V(\theta_n)|\Delta M_{n+1}) + [\nabla V]_{\mathrm{Lip}}\gamma_{n+1}^2|H(\theta_n, Z_{n+1})|^2,$$

- conditioning given $\mathcal{F}_n$ yields
  - $\mathbb{E}\big[(\nabla V(\theta_n)|\Delta M_{n+1})\,|\,\mathcal{F}_n\big] = 0$
  - and, other terms in the RHS being $\mathcal{F}_n$-measurable,

$$\mathbb{E}\left(V(\theta_{n+1})\,|\,\mathcal{F}_n\right) + \gamma_{n+1}(\nabla V|h)(\theta_n) \le V(\theta_n) + C_V\gamma_{n+1}^2\big(1 + V(\theta_n)\big) = V(\theta_n)(1 + C_V\gamma_{n+1}^2) + C_V\gamma_{n+1}^2$$

  with $C_V = C^2[\nabla V]_{\mathrm{Lip}} > 0$.

- Add
  - the positive term $\sum_{k=1}^n \gamma_k(\nabla V|h)(\theta_{k-1}) + C_V\sum_{k\ge n+2}\gamma_k^2$ on the left-hand side of the above inequality,
  - $(1 + C_V\gamma_{n+1}^2)$ times this term on the right-hand side .

- Divide the resulting inequality by $\prod_{k=1}^{n+1}(1 + C_V\gamma_k^2)$ shows that (the $\mathcal{F}_n$-adapted sequence)

$$S_n = \frac{V(\theta_n) + \sum_{k=0}^{n-1} \gamma_{k+1}(\nabla V|h)(\theta_k) + C_V \sum_{k\geq n+1} \gamma_k^2}{\prod_{k=1}^{n}(1 + C_V\gamma_k^2)}, \ n \geq 1,$$

  is a (non-negative) super-martingale with $S_0 = V(\theta_0) \in L^1(\mathbb{P})$.

- The fact that the added term is positive follows from the mean-reverting inequality $(\nabla V|h) \geq 0$.

- Hence

$$S_n \xrightarrow{a.s.} S_\infty \in L^1_{\mathbb{R}_+}(\mathbb{P}).$$

- Consequently, using that $\sum_{k\geq n+1}\gamma_k^2 \to 0$, one gets

$$V(\theta_n)+\sum_{k=0}^{n-1}\gamma_{k+1}(\nabla V|h)(\theta_k)\xrightarrow{a.s.}\widetilde{S}_\infty = S_\infty\prod_{n\geq 1}(1+C_V\gamma_n^2)\in L^1(\mathbb{P}).$$

- $(i)_a$ The super-martingale

$$(S_n)_{n\geq 0} \text{ is } L^1(\mathbb{P})\text{-bounded by } \mathbb{E}\,S_0=\mathbb{E}\,V(\theta_0)<+\infty,$$

hence $(V(\theta_n))_{n\geq 0}$ is L^1-bounded since

$$V(\theta_n)\leq\left(\prod_{k=1}^{n}(1+C_V\gamma_k^2)\right)S_n,\quad n\geq 0,$$

and $\prod_{k\geq 1}(1+C_V\gamma_k^2)<+\infty$ by the (DS) assumption on $(\gamma_n)_{n\geq 1}$.

- (*ii*) Now, for the same reason, the series with non-negative terms $\sum_{0\le k\le n-1}\gamma_{k+1}(\nabla V|h)(\theta_k)$ satisfies for every $n\ge 1$,

$$\mathbb{E}\left(\sum_{k=0}^{n-1}\gamma_{k+1}(\nabla V|h)(\theta_k)\right)\le\prod_{k=1}^{n}(1+C_V\gamma_k^2)\mathbb{E}\,S_0$$

so that, by the Beppo Levi monotone convergence Theorem for series with non-negative terms,

$$\mathbb{E}\left(\sum_{n\ge 0}\gamma_{n+1}(\nabla V|h)(\theta_n)\right)<+\infty$$

so that, in particular,

$$\sum_{n\ge 0}\gamma_{n+1}(\nabla V|h)(\theta_n)<+\infty\qquad\mathbb{P}\text{-}a.s.$$

and the series converges in L^1 to its $a.s.$ limit.

- $(i)_b$ It follows that $V(\theta_n) \longrightarrow V_\infty$ a.s. as $n \to +\infty$. $V_\infty \in L^1$ by Fatou's Lemma since $\big(V(\theta_n)\big)_{n\geq 0}$ is L^1-bounded.

- (iii) Again by Beppo Levi's monotone convergence Theorem for series with non-negative terms,

$$\mathbb{E}\left(\sum_{n\geq 1}|\Delta\theta_n|^2\right) = \sum_{n\geq 1}\mathbb{E}\,|\Delta\theta_n|^2 \leq \sum_{n\geq 1}\gamma_n^2\,\mathbb{E}\,|H(\theta_{n-1},Z_n)|^2$$
$$\leq C\sum_{n\geq 1}\gamma_n^2(1+\mathbb{E}\,V(\theta_{n-1})) < +\infty$$

so that

$$\sum_{n\geq 1}|\Delta\theta_n|^2 \in L^1(\mathbb{P}) \quad \text{(hence } as. \text{ finite)}$$

which in turns yields

$$\Delta\theta_n = \theta_n - \theta_{n-1} \to 0 \qquad a.s. \text{ and in } \; L^2(\mathbb{P})$$

- (*iv*) We have $M_n^\gamma = \sum_{k=1}^n \gamma_k \Delta M_k$ so that M^γ is clearly an $(\mathcal{F}_n)$-martingale. Moreover,

$$\langle M^\gamma \rangle_n = \sum_{k=1}^n \gamma_k^2 \, \mathbb{E}\left(|\Delta M_k|^2 \, |\mathcal{F}_{k-1}\right) \leq \sum_{k=1}^n \gamma_k^2 \, \mathbb{E}\left(|H(\theta_{k-1}, Z_k)|^2 \, |\mathcal{F}_{k-1}\right)$$
$$\leq C \sum_{k=1}^n \gamma_k^2 \big(1 + \mathbb{E}\, V(\theta_{k-1})\big)$$

Consequently, owing to $(i)_a$,

$$\mathbb{E}\, \langle M^\gamma \rangle_\infty < +\infty,$$

Hence M_n^γ converges *a.s.* and in L^2. □

Table of Contents

Robbins-Monro (pathwise reasoning)

Theorem (Robbins-Monto algorithm)

Assume the mean function h is continuous and satisfies

$$\forall\, \theta \in \mathbb{R}^d,\ \theta \neq \theta_*, \quad \big(\theta - \theta_* | h(\theta)\big) > 0.$$

Suppose furthermore that $\theta_0 \in L^2$ and that H satisfies

$$\forall\, \theta \in \mathbb{R}^d, \quad \big\|H(\theta, Z)\big\|_2 \leq C(1 + |\theta|).$$

Finally, assume $(\gamma_n)_{n\geq 1}$ satisfies (DS). Then

$$\{h = 0\} = \{\theta_*\} \quad \textit{and} \quad \theta_n \xrightarrow{a.s.} \theta_*.$$

The convergence also holds in every L^p, $p \in (0, 2)$ (and $\big(|\theta_n - \theta_*|\big)_{n\geq 0}$ is L^2-bounded).

- If $H(\theta, z) = h(\theta)$, back to a deterministic zero search procedure!!!

Proof

- The function $V(\theta) = \frac{1}{2}|\theta - \theta_*|^2$ is a Lyapunov function.
- The quadratic linear growth assumption on H is satisfied too.
- Robbins-Siegmund's Lemma implies
  - $|\theta_n - \theta_*|^2 \longrightarrow V_\infty \in L^1$,
  - $\sum_{n\geq 1} \gamma_n \big(h(\theta_{n-1})|\theta_{n-1} - \theta_*\big) < +\infty \quad \mathbb{P}\text{-}a.s.$
  - $(|\theta_n - \theta_*|^2)_{n\geq 0}$ is L^1-bounded.
- We keep on reasoning pathwise: let ω be generic.

- On has
$$\underline{\lim}_n \left(\theta_{n-1}(\omega) - \theta_* | h(\theta_{n-1}(\omega))\right) = 0.$$
- If $\underline{\lim}_n \left(\theta_{n-1}(\omega) - \theta_* | h(\theta_{n-1}(\omega))\right) > 0$, the above convergence induces a contradiction with $\sum_{n\geq 1} \gamma_n = +\infty$.
- Let $\left(\phi(n,\omega)\right)_{n\geq 1}$ be a subsequence such that
$$\left(\theta_{\phi(n,\omega)}(\omega) - \theta_* | h(\theta_{\phi(n,\omega)}(\omega))\right) \longrightarrow 0 \quad \text{as} \quad n \to +\infty.$$
- Now, $(\theta_n(\omega))_{n\geq 0}$ being bounded, one may assume, up to one further extraction,
$$\theta_{\phi(n,\omega)}(\omega) \to \theta_\infty = \theta_\infty(\omega).$$
- By continuity of h, $\left(\theta_\infty - \theta_* | h(\theta_\infty)\right) = 0$ which implies $\theta_\infty = \theta_*$. Now, since we know that $V(\theta_n(\omega)) = \frac{1}{2}|\theta_n(\omega) - \theta_*|^2$ converges,
$$\lim_n \left|\theta_n(\omega) - \theta_*\right|^2 = \lim_n \left|\theta_{\phi(n,\omega)}(\omega) - \theta_*\right|^2 = 0.$$
- Convergence in L^p, $p \in (0,2)$ follows by uniform integrability. □

Theorem (Stochastic Gradient Descent)

◀ *Let* $V : \mathbb{R}^d \to \mathbb{R}_+$ *be a differentiable function* $\lim_{\infty} V(\theta) = +\infty$, ∇V *Lipschitz,* $|\nabla V|^2 \leq C(1+V)$ *and* $\{\nabla V = 0\} = \{\theta_*\}$.

◀ *Let* $h(\theta) = \mathbb{E}\, H(\theta, Z) = \nabla V$ *with* H *s.t.* $\|H(\theta, Z)\|_2 \leq C\sqrt{1+V(\theta)}$ *and that* $V(\theta_0) \in L^1(\mathbb{P})$. *Assume* $(\gamma_n)_{n\geq 1}$ *satisfies (DS).*

Then

$$V(\theta_*) = \min_{\mathbb{R}^d} V \quad \text{and} \quad \theta_n \xrightarrow{a.s.} \theta_* \quad \text{as} \quad n \to +\infty.$$

Moreover, $\nabla V(\theta_n)$ *converges to* 0 *in every* L^p, $p \in (0,2)$ *(and* $\left(V(\theta_n)\right)_{n\geq 0}$ *is* L^1*-bounded so that* $\left(\nabla V(\theta_n)\right)_{n\geq 0}$ *is* L^2*-bounded).*

- *Proof.* Use (almost) the same arguments as above but with $(\nabla V \,|\, h) = |\nabla V|^2$ instead of $(\theta - \theta_* \,|\, h(\theta))$. □
- If $H(\theta, z) = h(\theta) = \nabla V(\theta)$: Convergence thm for Gradient descent!!

Multi-target stochastic algorithms

Theorem (Multitarget Stochastic Gradient Descent)

(*a*) *If the former assumption* $\{\nabla V = 0\} = \{\theta_*\}$, *one has mutatis mutandis: a.s. there exists* $v_\infty \in \mathbb{R}_+$ *and a connected component* χ_∞ *of* $\{\nabla V = 0\} \cap \{V = v_\infty\}$ *such that*

$$\mathrm{dist}(\theta_n, \chi_\infty) \longrightarrow 0 \quad a.s.$$

(*b*) *In particular if* $\{\nabla V = 0\} \cap \{V = v\}$ *is locally finite for every* $v \geq 0$ *is finite, then there exists a r.v.* θ_∞ *such that*

$$\nabla V(\theta_\infty) = 0 \quad \text{and} \quad \theta_n \longrightarrow \theta_\infty.$$

(*c*) *Moreover,* $\nabla V(\theta_n)$ *converges to* 0 *in every* L^p, $p \in (0,2)$ *(and* $\big(V(\theta_n)\big)_{n\geq 0}$ *is* L^1*-bounded so that* $\big(\nabla V(\theta_n)\big)_{n\geq 0}$ *is* L^2*-bounded).*

- By the R.-S. Lemma, one has for free that, a.s., $(\theta_n)_{n\geq 0}$ is pathwise bounded and $\theta_n - \theta_{n-1} \longrightarrow 0$ pathwise. Hence its limiting values makes up a connected compact set Θ_∞, clearly included in some $\{V = v_\infty\}$.
- But this is not enough... Θ_∞ is also invariant under the flow of

$$ODE \equiv \dot{\theta} = -h(\theta)$$

  which converges toward $\{\nabla V = 0\}$.
- Still not enough : needs to make a transfer from *ODE* to algorithm.
- Needs further insights based on topology and the *ODE* method ([3]).

[3] G. Pagès (2018). *Introduction to Numerical Probability with application to Finance, Springer-Verlag, Berlin, 576p.*

Avoiding traps

Theorem (Traps (Pemantle 1984, Lazarev 1989, Brandière-Duflo 1996, Fort-P. 1997, Benaïm 1998s))

Let $\theta_* \in \{\nabla V = 0\}$. If there exists (λ, u) such that $D^2 V u = \lambda u$ such that

$$\lambda < 0 \quad \text{and} \quad \mathbb{E}\left(H(\theta_*, Z)|u\right)^2 > 0$$

then

$$\mathbb{P}\big(\theta_n \to \theta_*\big) = 0.$$

- This allows to eliminate noisy local maxima, saddle points, monkey saddle points, etc.

Table of Contents

Table of Contents

Numerical probability: Implicit volatility II

- Let ([4]) $h(\sigma) = \mathrm{Call}_{BS}(\sigma) - \mathrm{Call}_{M2Mkt} = \mathbb{E}H(\sigma, Z)$ with

$$H(\theta, z) = \left(x_0\, e^{-\frac{\sigma_+^2}{2}T + \sigma_+\sqrt{T}z} - e^{-rT}K \right)_+ - \mathrm{Call}_{M2Mkt}$$

  (σ_+ to ensure that h is increasing).
- Then the recursive stochastic zero search reads

$$\sigma_{n+1} = \sigma_n - \gamma_{n+1} H(\sigma_n, Z_{n+1}), \quad \sigma_0 > 0.$$

  with $(Z_n)_{n\geq 1}$ i.i.d.,$\sim \mathcal{N}(0,1)$ and $\sum_n \gamma_n = +\infty$, $\sum_{n\geq 1} \gamma_n^2 < +\infty$
- Try with $\gamma_n = \frac{a}{b+n}$ so that $\gamma_1 \times H(\sigma_0, Z_{+1}) \simeq$ few units.

[4] G. Pagès (2018). *Introduction to Numerical Probability with application to Finance, Springer-Verlag, Berlin, 576p.*

Numerical probability: correlation search II

- Let $h(\rho) = \text{Best-ofCall}_{BS}(\dots,\rho,\dots) - \text{Best-of-Call}_{M2Mkt}$
$= \mathbb{E}\, H(\rho, Z),\ Z = (Z^1, Z^2) \sim \mathcal{N}(0, I_2)$

with $H(\rho, Z) = \left(\max\left(x_0^1 e^{-\frac{\sigma_1^2 T}{2} + \sigma_1 \sqrt{T} z^1}, x_0^2 e^{-\frac{\sigma_2^2 T}{2} + \sqrt{T}\sigma_2(\rho z^1 + \sqrt{1-\rho^2} z^2)} \right) - e^{-rT} K \right)_+ - \text{Best-of-Call}_{M2Mkt}.$

- The naive algorithm (with $(\gamma_n)_{n\geq 1}$ satisfying the (DS) assumption)
$$\rho_{n+1} = \rho_n - \gamma_{n+1} H(\rho_n, Z_{n+1})$$
does not live inside $[-1,1]$!! ...
- What to do ? Project on $[-1,1]$ (theorems do exist) or change of variable ([5])
$$\rho = \tfrac{2}{\pi} \arctan(\theta) =: \varphi(\theta)$$
so that
$$\theta_{n+1} = \theta_n - \gamma_{n+1} H\big(\varphi(\theta_n), Z_{n+1}\big),\ \theta_0 \in \mathbb{R}.$$
- There exists a C^1_{Lip}-Lyapunov function $V : \mathbb{R} \to \mathbb{R}_+$ such that $V'(\theta) h\big(\varphi(\theta)\big) \geq 0$ and $= 0$ iff $h(\varphi(\theta)) = 0$ i.e. $\varphi(\theta) = \rho_*$.

[5] G. Pagès (2018). *Introduction to Numerical Probability with application to Finance, Springer-Verlag, Berlin, 576p.*

Higher dimensions

- In higher dimension a correlation matrix R whose Cholesky decomposition

$$R = TT^\top \quad \text{with } T \text{ lower triangular and } \quad \sum_{j=1}^{i} t_{ij}^2 = 1.$$

- (Hyper-)spherical parametrization

$$t_{11} = 1$$
$$t_{21} = \cos(\theta_2),\ t_{22} = \sin(\theta_2)$$
$$t_{31} = \cos(\theta_3)\sin(\phi_3); L_{32} = \cos(\theta_3)\cos(\phi_3)$$
$$t_{41} = \cos(\theta_4)\cos(\phi_4)\cos(\psi_4),\ t_{42} = \cos(\theta_4)\cos(\phi_4)\sin(\psi_4)$$
$$t_{43} = \cos(\theta_4)\sin(\phi_4),\ t_{44} = \cos(\theta_4).$$

  etc.

- Then $\theta = (\theta_2, , \theta_3, \phi_3, \theta_4, \phi_4, \psi_4)$.
- More involved problem: periodicity introduces multiple solutions.

Numerical probability: VaR_α-$CVaR_\alpha$ II

- Set $H(\xi, x) = \partial_\xi v(\xi, x) = 1 - \frac{1}{1-\alpha}\mathbf{1}_{\{x \geq \xi\}} = \frac{1}{1-\alpha}\left(\mathbf{1}_{\{x \leq \xi - \alpha\}}\right)$ so that

$$V'(\xi) = \mathbb{E}\, H(\xi, X)$$

- Set $\gamma_n = \frac{1}{n}$ and let X_n i.i.d., $\sim X$, then

$$\xi_{n+1} = \xi_n - \frac{\gamma_{n+1}}{1-\alpha}\left(\mathbf{1}_{\{X_{n+1} \leq \xi_n\}} - \alpha\right) \longrightarrow \xi_\alpha = \mathrm{VaR}_\alpha(X).$$

- What about $\mathrm{CVaR}_\alpha(X)$? Various solutions...

$$\Xi_n = \frac{v(\xi_0, X_1) + \cdots + v(\xi_{n-1}, X_n)}{n} \longrightarrow \mathbb{E}\, v(\xi_\alpha, X) = \mathrm{CVaR}_\alpha(X).$$

- Recursive form $\Xi_n = \Xi_{n-1} - \frac{1}{n}\Big(\Xi_{n-1} - v(\xi_{n-1}, X_n)\Big)$, $\Xi_0 = 0$.
- Warning ! Rare events phenomenon tends to freeze the algorithm $\Rightarrow$ adaptive Importance Sampling ([6]) !
- ...and try to slowly increase $\alpha = \alpha_n$ from $\alpha_0 = \frac{1}{2}$ to the target level.

[6] O. Bardou, N. Frikha, G. Pagès (2009). Computing VaR and CVaR using Stochastic Approximation and Adaptive Unconstrained Importance Sampling, *Monte Carlo and Applications Journal*, **15**(3):173–210.

First conclusions

- Low dimensional examples selected on purpose for expository.
- Not as automatic as (linear) Monte Carlo simulation: tuning of the step is mandatory.
- Many other examples : adaptive variance reduction (see Lemaire-P. 2007, AAP).
- Central-Limit Theorem, Averaging principle (Ruppert-Polyak).
- More details and results in ([7]) if interested and the references therein.

[7] G. Pagès (2018). *Introduction to Numerical Probability with application to Finance, Springer-Verlag, Berlin, 576p.*

Table of Contents

Learning

- Database $(z_k)_{k=1:N}$, parameters $\theta \in \Theta \subset \mathbb{R}^K$ and (local) loss function/predictor $v(\theta, z)$.
- Let $(I_k)_{k\geq 1}$ be an i.i.d. sequence $\mathcal{U}\big(\{1,\ldots,N\}\big)$-distributed.
- The stochastic gradient descent reads

$$\theta_{n+1} = \theta_n - \gamma_{n+1}\nabla_\theta v\big(\theta_n, z_{I_{n+1}}\big)$$

  where $z_{I_{n+1}}$ means that a datum has been picked up at random in the database uniformly in $\{1,\ldots,N\}$.
- Check that

$$\begin{aligned}\mathbb{E}\,\nabla_\theta v\big(\theta_n, z_I\big) &= \frac{1}{N}\sum_{k=1}^{N}\nabla_\theta v\big(\theta_n, z_k\big)\\ &= \int \nabla_\theta v\big(\theta_n, z\big)\mu_N(dz) = \nabla V(\theta_n)\end{aligned}$$

CLVQ/k-means (unsupervised learning)

- Aim:

$$\min_{(\theta^j)_{j=1:r}} \left[V(\theta) = \frac{1}{2} \sum_{k=1}^{N} \min_{i=1:r} |\theta^i - x_k|^2 \right]$$

  (mean minimal distance to prototypes).

- Competitive Learning Vector Quantization:

$$\theta^i_{n+1} = \begin{cases} \theta^i_n - \gamma_{n+1}\left(\theta^i_n - x_{n+1}\right) & \text{if } |x_{n+1} - \theta^i_n| < \min_{j \neq i} |x_{n+1} - \theta^j_n| \\ = 0 & \text{otherwise} \end{cases}$$

- In other words: $\longrightarrow n+1$ reads
  - Nearest neighbour searchto the datum among r prototypes of dimension d .
  - Moving the winner by a dilatation centered at the datum with ratio $1 - \gamma_{n+1} > 0$.

CLVQ

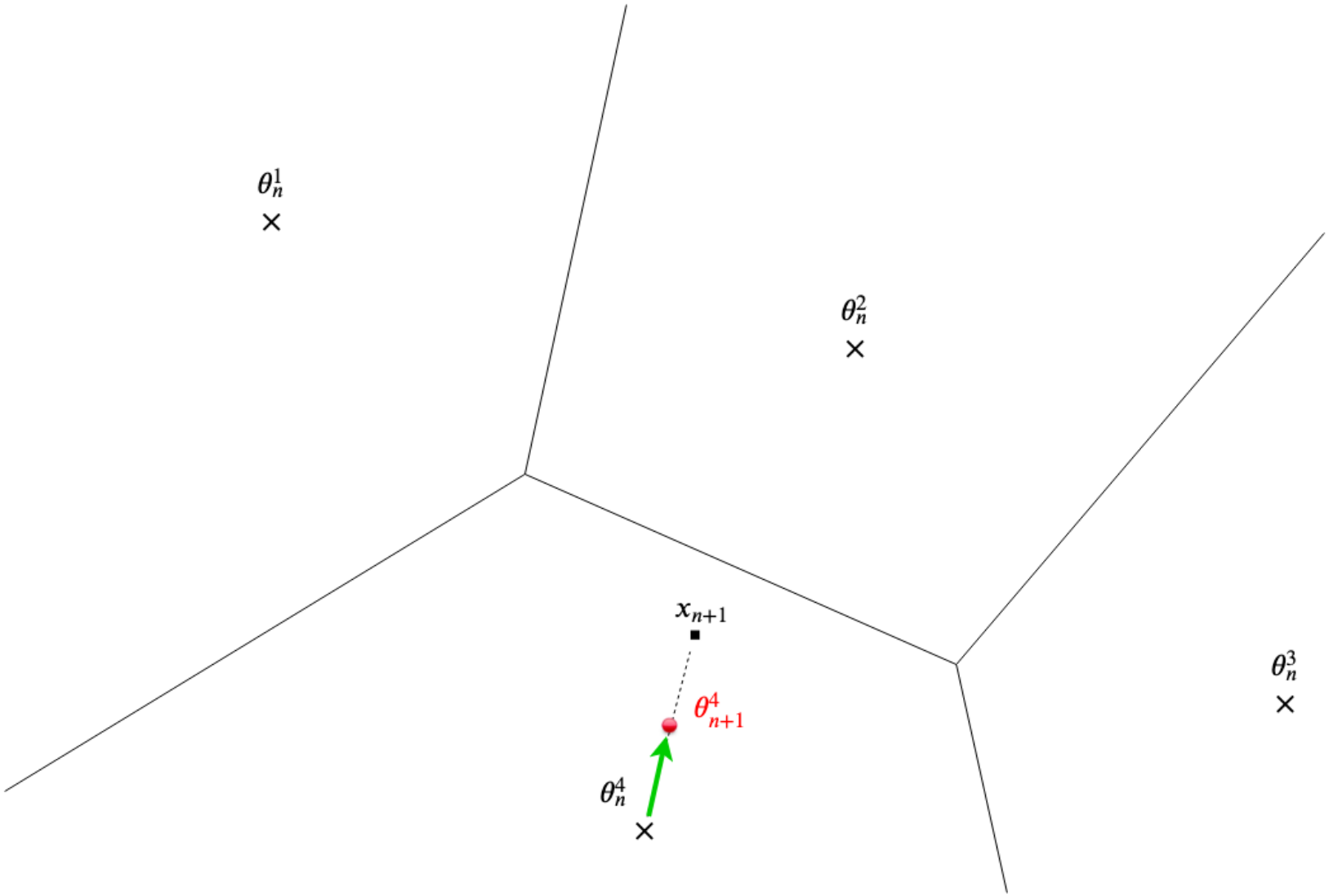


Table of Contents

Table of Contents

Linear artificial neuron

- Mc Cullogh & Pitts in 1943 : linear "neuron": linear partitioner of two classes of date (if any...).
- Learning by Hebb's rule also known as reinforcement rule.
- The first perceptron: 1957 by Rosenblatt ([8]) as a binary classifier.
- Only able to classify linearly separable classes of data.

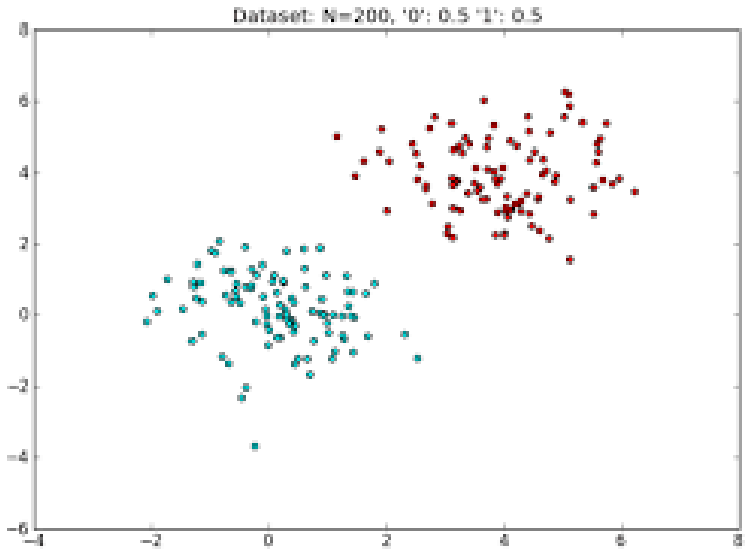


Figure: The Rosenblatt neural network performances.

[8] Rosenblatt, Frank (1957), The Perceptron: a perceiving and recognizing automaton. Report 85-460-1, Cornell Aeronautical Laboratory.

No hidden layer: ADALINE (Adaptive Linear Neuron (B. Widrow & T. Hoff, 1960))

- Input data $x_1, \ldots, x_N$ of the form $x_k = (1, x_k^1, \ldots, x_N^d) \in \mathbb{R}^{d+1}$.
- Output (true) data are real numbers $y_1, \ldots, y_N$.
- Data set: $z_k = (x_k, y_k)$, $k = 1 : N$ (supervised learning).
- Let $\Theta = \mathbb{R}^{d+1}$ the parameter set..
- The answer of the ADALINE neuron/network for $\theta \in \Theta$ "fed" with an imput datum $x = (1, x^1, \ldots, x^d)$ is

$$\theta^\top x = \sum_{i=0}^{d} \theta^i x^i = \theta^0 + \sum_{i=1}^{d} \theta^i x^i$$

  where $\theta^\top$ denote the transpose of θ.
- (Convex) local prediction/loss function: $v\big(\theta, (x, y)\big) = \frac{1}{2}(y - \theta^\top x)^2$

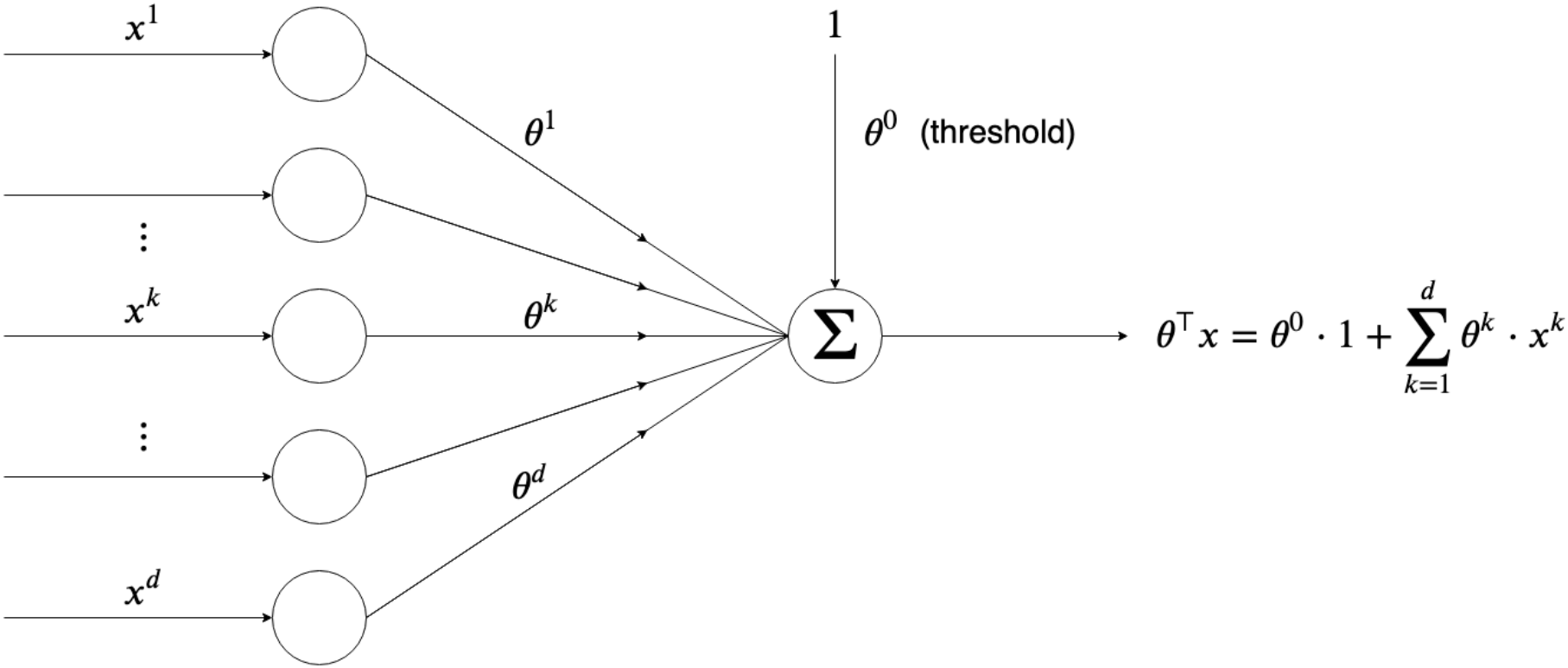


Figure: The ADALINE neural network (with T. Montes)

- $V(\theta) = \frac{1}{2}\int (\theta^\top x - y)^2 \mu_N\big(d(x,y)\big) = \mathbb{E}\,(\theta^\top X_I - Y_I)^2$ (convex!)

  i.e. Global prediction/loss function = Ordinary Least squares !

- $\nabla V(\theta) = \int (\theta^\top x - y) x \, \mu_N\big(d(x,y)\big)\big).$

- The target is

$$\nabla V(\theta) = 0 \Longleftrightarrow \int \underbrace{(\theta^\top x)x}_{=(xx^\top)\theta} \mu_N\big(d(x,y)\big)\big) = \int y\, x\, \mu_N\big(d(x,y)\big)\big)$$

$$\Longleftrightarrow \theta_{opt}^\top = \left(\int x\, x^\top \mu_N\big(d(x,y)\big)\right)^{-1} \left(\int y\, x\, \mu_N\big(d(x,y)\big)\right).$$

- It is linear regression (as expected).

- Resulting (SGD):

$$\theta_{n+1} = \theta_n - \gamma_{n+1}\big(\theta_n^\top x_{I_{n+1}} - y_{I_{n+1}}\big)x_{I_{n+1}}$$

  All assumptions are satisfied $\Longrightarrow \theta_n \to \theta^*$ $a.s.$

- Conclusion : the ADALINE network performs linear regression without matrix inversion.

- Extension to q-dimensional (true) output data y_k with $\Theta = \mathbb{M}(d, q)$ and

$$v\big(\theta, (x, y)\big) = \tfrac{1}{2}|\theta^\top x - y|^2.$$

"Non-linear" ADALINE

- Assume the output data y_k are e.g. $(0,1)$-valued.
- Let $\Psi \in C\big(\mathbb{R},(0,1)\big)$, Ψ increasing homeomorphism from $\mathbb{R}$ onto $(0,1)$.
- $V(\theta) = \frac{1}{2}\mathbb{E}\left(\Psi(\theta^\top X_I) - Y_I\right)^2$.
- $\nabla V(\theta) = \int \Big(\big(\Psi(\theta^\top x) - y\big)\Psi'(\theta^\top x)x\Big)\mu_N\big(d(x,y)\big)$.
- Loss of convexity since

$$\nabla^2 V(\theta) = \int \left[\Psi'(\theta^\top x)^2 + \big(\Psi(\theta^\top x) - y\big)\underbrace{\Psi''(\theta^\top x)}_{\gtrless 0} xx^\top\right]\mu_N\big(d(x,y)\big)$$

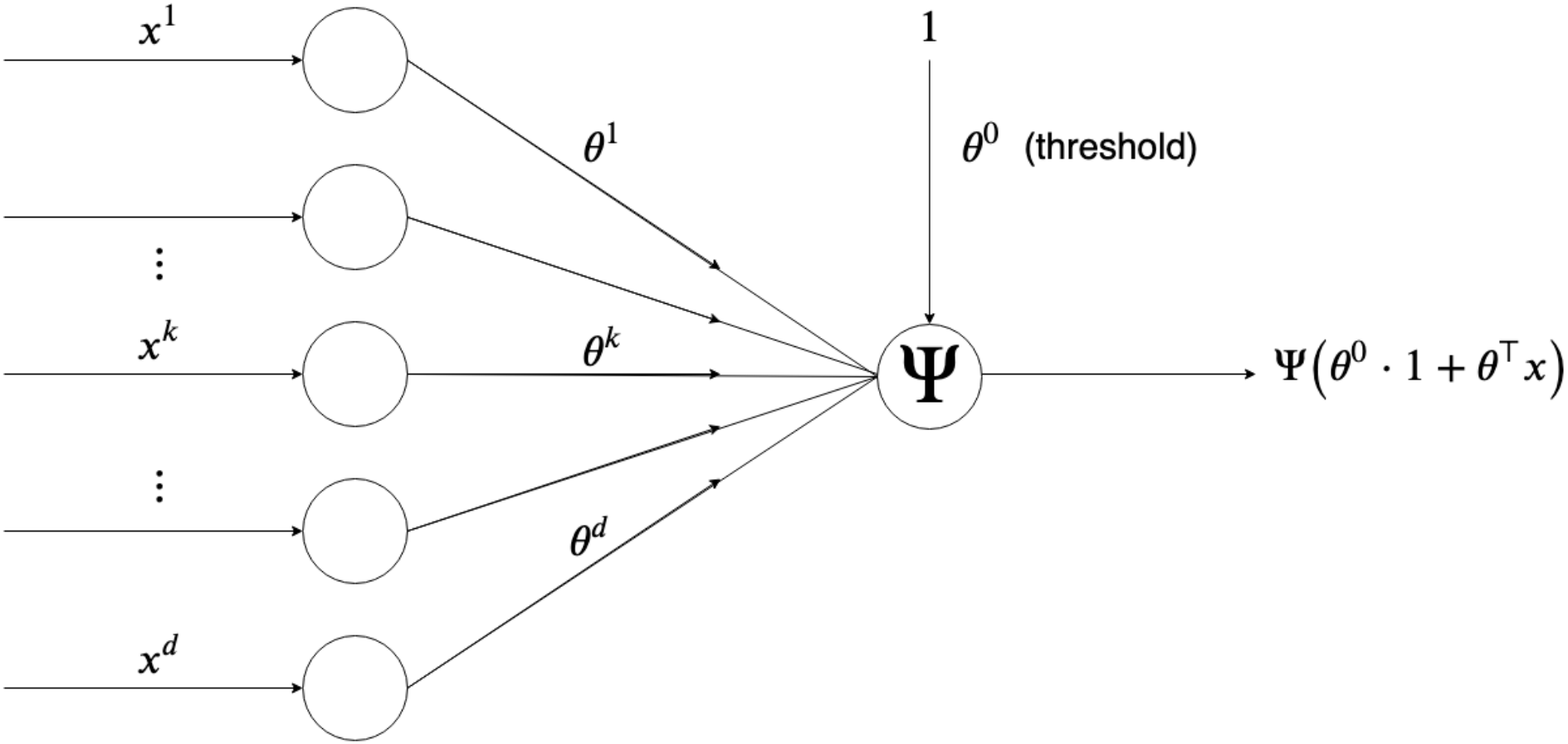


Figure: The non-linear ADALINE neural network (with T. Montes).

- Historical Rosenblatt's "linear perceptron"' (1957): $\Psi(u) = \mathbf{1}_{u \geq 0}$ is a linear classifier.

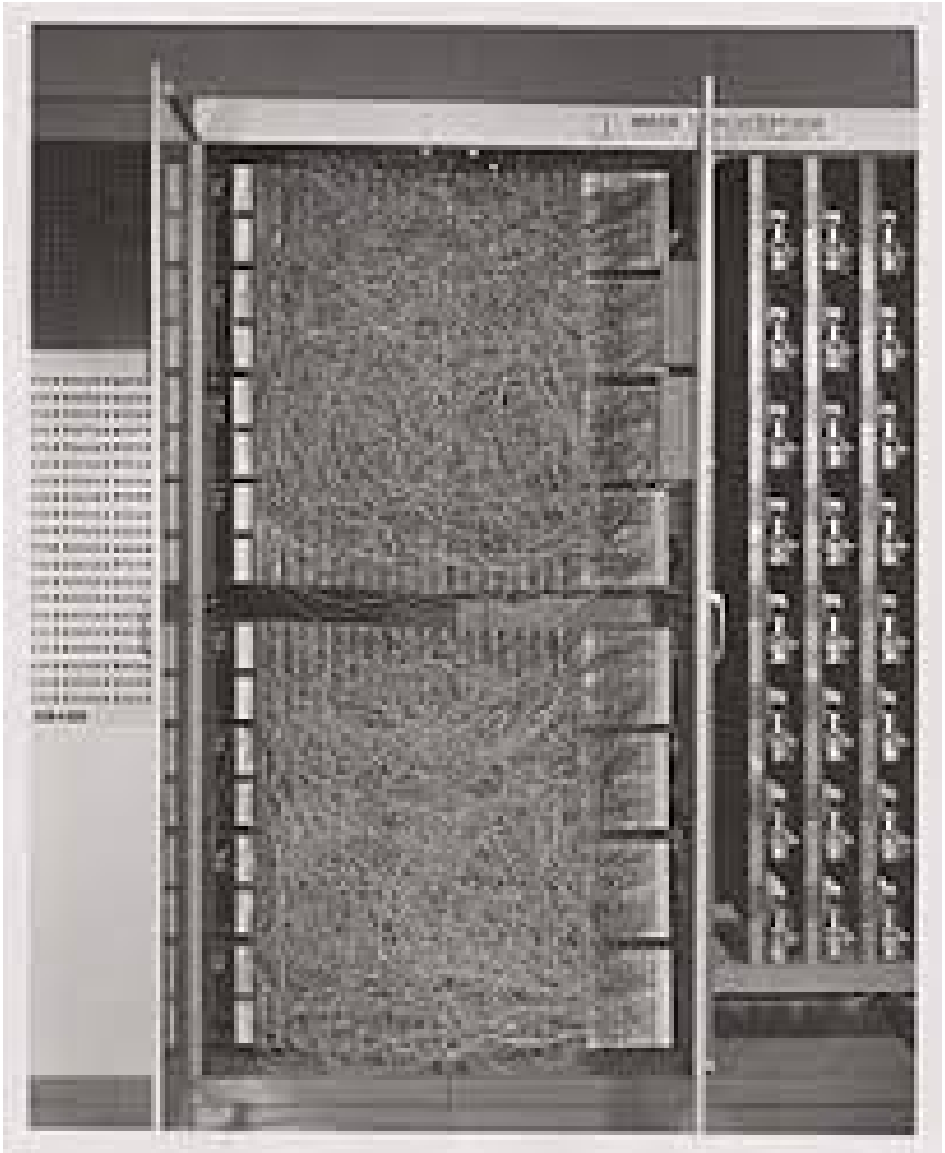


Figure: Historical Rosenblatt's perceptron: the hardware (without T. Montes !).

Table of Contents

One hidden layer: universal approximation property

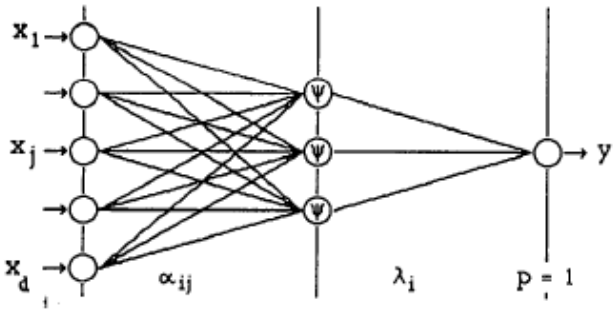


Figure: *One hidden layer MLP* *(below: convention $\alpha_{ij} = w_{ij}$).*

- $d = d_x$-dimensional inputs. Switch to $x \rightsquigarrow \begin{pmatrix} 1 \\ x \end{pmatrix}$.
- L units $i = 1 : L$ on the hidden layer, with an *activation function*

$$\Psi \in C(\mathbb{R}, \mathbb{R}).$$

- Unit i of the hidden layer receives $x = (x^j)_{j=1:d}$ and emits

$$\Psi\big((w_{i\cdot}|x)\big) = \Psi\Big(w_{i0} + \sum_{1 \le j \le d} w_{ij}\, x^j\Big).$$

- The output layer receives $\big[\Psi\big((w_{i\cdot}|x)\big)\big]_{i=1:L}$ and emits

$$\sum_{1 \le i \le L} \lambda_\ell \Psi\big((w_{i\cdot}|x)\big).$$

Universal approximation property

Theorem (Cybenko, 1989)

([a]) If the activation function Ψ satisfies

$$(CL_\Psi) \equiv \Psi \in C_b(\mathbb{R},\mathbb{R}), \quad \lim_{\xi\to-\infty}\Psi(\xi)=0, \quad \lim_{\xi\to+\infty}\Psi(\xi)=1,$$

then

$$F=\left\{x\longmapsto\sum_{\ell=1}^{L}\lambda_\ell\Psi\big((w_{i\cdot}|x)\big),\ L\in\mathbb{N}^*,\ \lambda\in\mathbb{R}^L,\ w\in\mathbb{R}^{L\times(d+1)}\right\}$$

is $\|\cdot\|_{\sup}$-dense in $C([0,1]^d,\mathbb{R})$.

[a] G. Cybenko, Approximation by Superpositions of a Sigmoidal Function. Mathematics of Control, Signals, and Systems, 2:303-314, 1989.

- (CL_Ψ) can be slightly relaxed into Hornik's condition ([9], 1989)

$$(CS_\Psi) \equiv \Psi \in C_b(\mathbb{R},\mathbb{R}), \quad \exists\, x_1,\, x_2\in\mathbb{R}^d \text{ such that } \Psi(x_1)\neq\Psi(x_2).$$

- Extension to vector-valued outputs is straightforward.

[9] Hornik K., Stinchcombe M. & White H., Multilayer feedforward networks are universal approximators. *Neural Networks*, **2**, 359–366

Proof I

Assume $d = 1$ (for simplicity).

- Step 1. If $\overline{F}^{\|\cdot\|_{\sup}} \subsetneq C([0,1],\mathbb{R})$, linear form $\Lambda \in C([0,1],\mathbb{R})^*$ s.t.

$$\Lambda \not\equiv 0 \quad \text{and} \quad \Lambda_{|F} \equiv 0 \quad \text{(by Hahn-Banach's Theorem)}.$$

By Riesz's representation Theorem, there exists a signed measure μ s.t.

$$\Lambda g = \int_{\mathbb{R}} g \, d\mu.$$

Hence, with for every weight $w = (\alpha, \beta)$,

$$\forall\, w = (\alpha,\beta) \in \mathbb{R}^2, \quad \int_{\mathbb{R}} \Psi(\alpha x + \beta)\mu(dx) = 0.$$

- Step 2. Let $\mu = \mu^+ - \mu^-$, denote the/a decomposition of μ.
  - Let $\beta \to +\infty$. By Lebesgue's Dominated Convergence (*LDC*)

$$\int \mathbf{1}\mu(dx) = 0 \quad \text{so that} \quad \mu^+(\mathbb{R}) = \mu^-(\mathbb{R}).$$

Proof II

- Now setting $\beta = -\alpha u$ yields

$$\forall\, \alpha,\, u \in \mathbb{R}, \quad \int \Psi(\alpha(x-u))\mu(dx) = 0.$$

Letting $\alpha \to +\infty$ implies, still by *LDC* theorem

$$\forall\, u \in \mathbb{R}, \quad \mu\big(\{x : x > u\}\big) + \Psi(0)\mu(\{u\}) = 0.$$

- The set $D = \big\{u : \mu^+(\{u\}) > 0 \text{ or } \mu^-(\{u\}) > 0\big\}$ is at most countable and, for every $u \in D^c$,

$$\forall\, u \in D^c, \quad \mu\big(\{x : x > u\}\big) = 0.$$

▶ Step 3. Combined with $\mu^+(\mathbb{R}) = \mu^-(\mathbb{R})$, one has

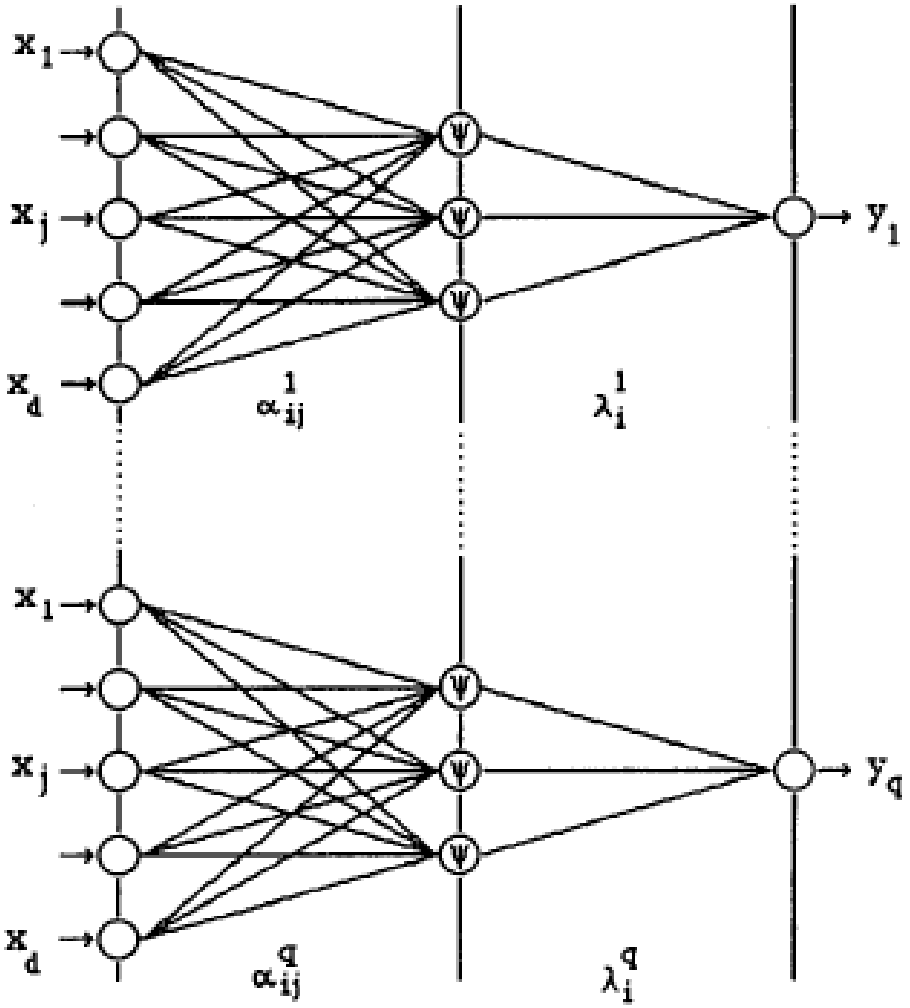
$$\forall\, u \in \mathbb{R}, \quad \mu^+\big(\{x : x > u\}\big) = \mu^-\big(\{x : x > u\}\big).$$

Hence $\mu^+ = \mu^-$ since D^c is everywhere dense in $\mathbb{R}$ i.e. $\mu \equiv 0$.

▶ Step 4. Contradiction! Since $\Lambda \not\equiv 0$. □

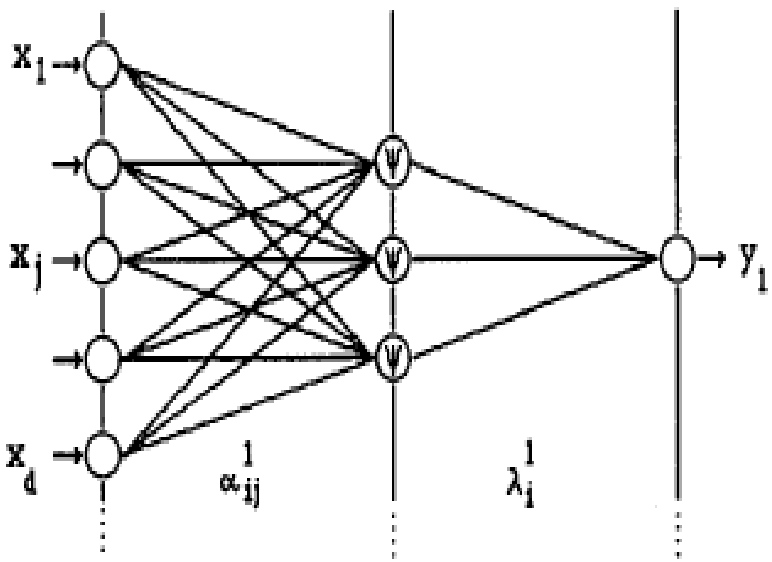
Higher dimensional outputs I

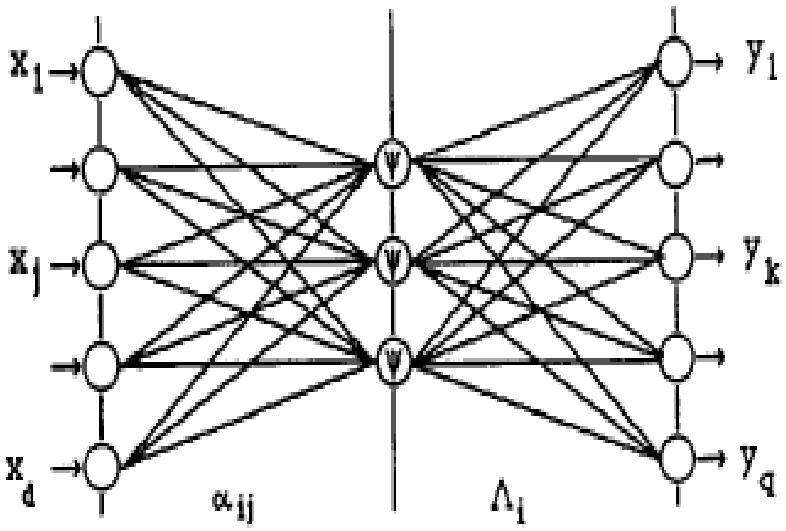
Instant result by concatenating networks



Higher dimensional outputs II

Although this is better in practice





Constructive proof/smoothness

- Both proofs are not constructive (Hornik's relies on a distribution based argument).
- Constructive approach are possible including rates for higher order derivatives like...

Theorem (Attali-Pagès, 1997)

([a]) Assume the activation function $\Psi \in C_b^r(\mathbb{R},\mathbb{R})$ *and satisfies*

ψ *is non-polynomial on any open interval and* $\exists\, \xi_0$ *s.t.* $\psi^{(s)}(\xi_0) \neq 0,\ s = 0,\dots,r.$

Let $f \in C^\infty(\mathbb{R}^d,\mathbb{R})$. *For every* $L \in \mathbb{N}$, *there exists weights* $(w_{ij}^{(L)})_{i=1:d,j=1:L}$, $\lambda_{i=1:L}^{(L)}$ *such that (with* $\|g\|_{[0,1]^d} = \sup_{\xi\in[0,1]^d} |g(\xi)|$*)*

$$\max_{s=0:r} \left\| \partial^s \Big(\sum_{i=1}^{L} \lambda_i^{(L)} \psi\big((w_i^{(L)}|x)\big) \Big) - \partial^s f \right\|_{[0,1]^d} \leq C_f \cdot L^{-1/d}$$

[a] J.-G. Attali, G. Pagès, Approximations of Functions by a Multilayer a New Approach, *Neural Networks*, **10**(6):1069-10811,1997.

- The proof heavily relies on multi-variable Bernstein polynomials and Vandermonde determinants.
- Our 1997 paper does not avoid the curse of dimensionality and came too late: Vapnik's *SVM* were coming at the front owing to a nice mathematical framework.
- Y. Le Cun et al. stroke back in the early 2000's with outstanding performances classification results at a yearly challenge using deep learning...
- New networks with more layers, not fully connected and new type of units (convolutive units, recurrent units).
- and even more recently Generative Adversarial Networks (2014) seem to get rid of curse of dimensionality,
- but, so far, no theoretical evidence.

Table of Contents

From learning to deep learning

- Feedforward multilayer perceptron.
- 1998 (Y. Le Cun): MNIST database of handwritten figures.
  - Input: Image $= 28 \times 28$ pixels $\times$ 256 grey levels $d_x \simeq 2 \times 10^5$.
  - Output : probability y_i for each figure 10 figures : $[0,1]^{10}$ i.e. $d_y \simeq 9$.
  - Predictor : Multilayer perceptron (MLP) with 2 hidden layers with 200 units $\Longrightarrow K = 200 \times 10 = 22 \times 10^3$ parameters
  - Size of the database: $N = 50\,000$ for learning, $10\,000$ for testing.
- 2010: Same database
  - Predictor: MLP with convolutional units on *GPU* with 7 hidden layers

    $K = 2\,500 \times 2\,000 \times 1\,500 \times 1\,000 \times 500 \times 10 \simeq 122 \times 10^6$ parameters
- 2013 (Google) : ImageNet database of $N = 162 \times 10^6$ images of 100×100 pixels and 256 grey levels: $d_x \simeq 2.562 \times 10^6$.
  - $d_y = 21$ (classifier across 21 classes).
  - Predictor: Convolutional MLP network $\Longrightarrow K \simeq 1.72 \times 10^9$ parameters.
  - Error rate $< 1/1\,000$.

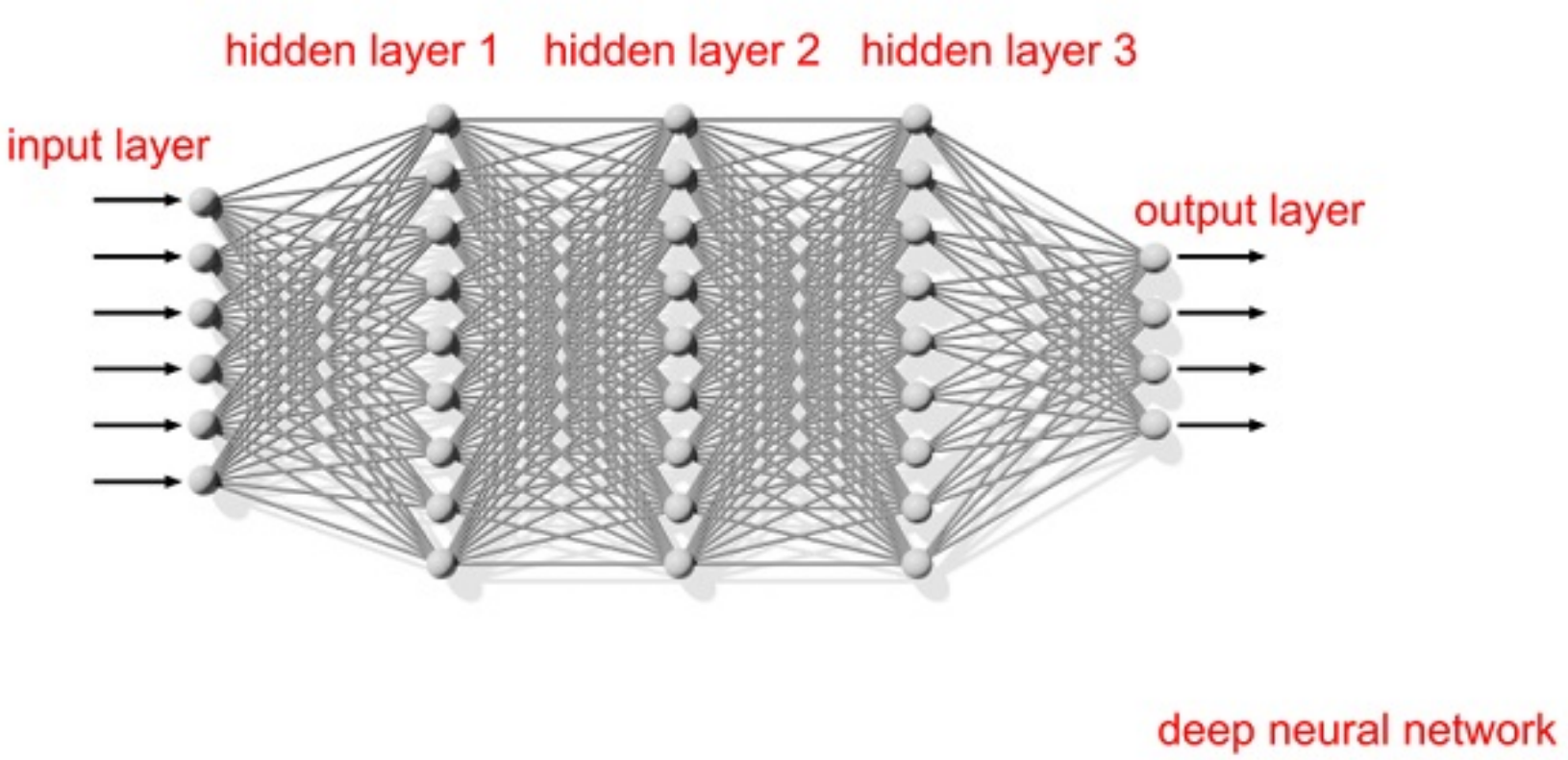


Figure: *A three-hidden layer feedforward perceptron.*

Table of Contents

Multilayer : toward Back propagation

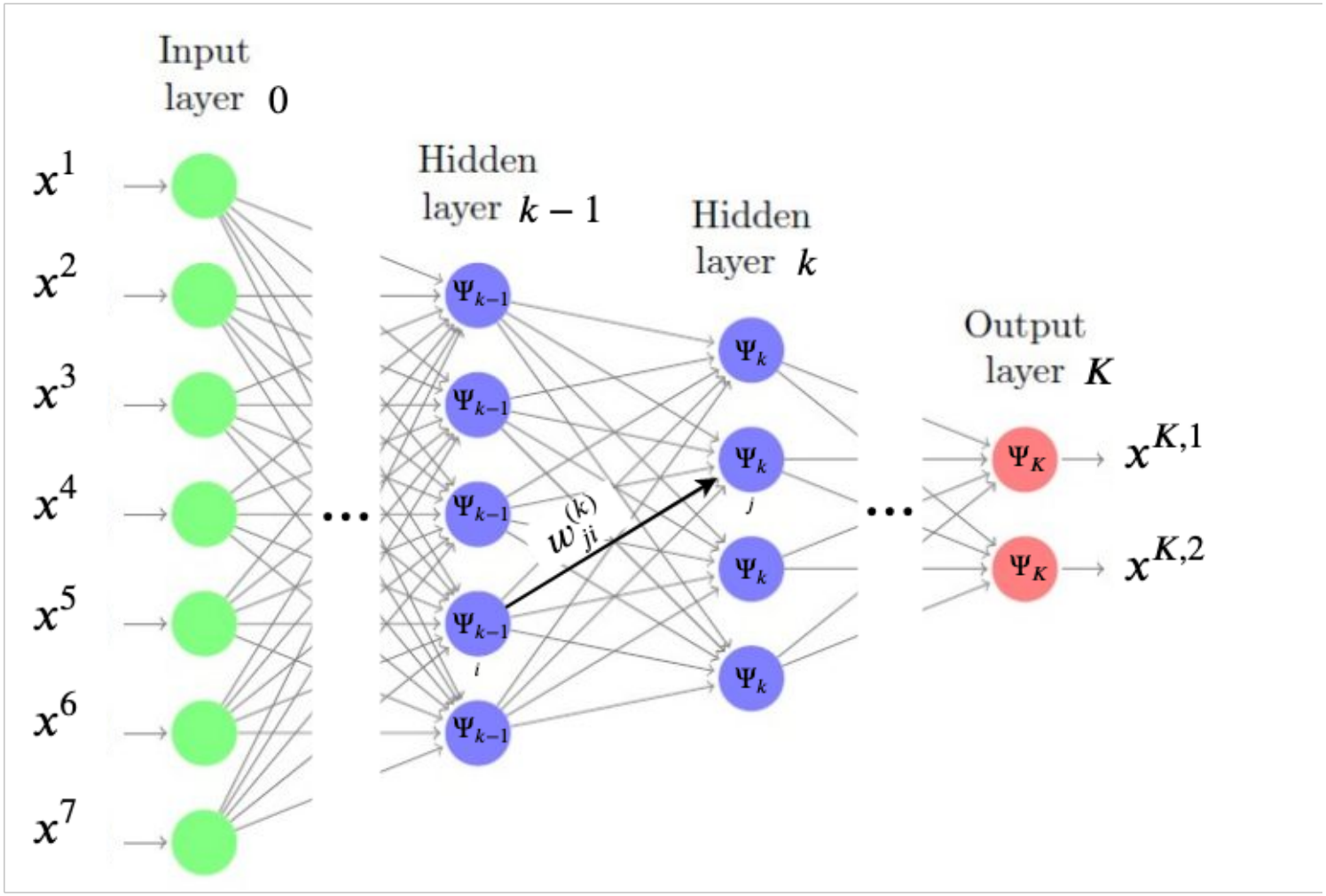


Figure: *A $K-1$ hidden layer feedforward perceptron.* *Warning! New notation $w_{ji}^{(k)}$ instead of w_{ij}.*

Multilayer and backpropagtion in one slide

- Input data (layer 0): $x^0 = x$ i.e. $x_j^0 = x_j$, $j = 1 : d_0 = d + 1$,
- $K - 1$ hidden layers ($k = 1 : K - 1$) and $K + 1$ layers (0 and K)
- Output of unit i of layer $k - 1$: $x^{k-1,i}$.
- Input of unit j of layer k:

$$\sum_{1 \le i \le d_{k-1}} w_{ji}^{(k)} x^{k-1,i} = (w_{j\cdot}^{(k)} \,|\, x^{k-1,\cdot})$$

so that, after passing through the activation function

$$x^{k,j} = \Psi_k\big((w_{j\cdot}^{(k)} \,|\, x^{k-1,\cdot})\big).$$

- Typical activation functions

$$\Psi_k(\xi) = c \cdot \frac{e^\xi}{e^\xi + 1}, \quad \Psi_k(\xi) = c \cdot \arctan(\xi), \quad \Psi_k(\xi) = c \cdot \xi_+$$

In fact two slides

- Parameter to be calibrated to perform the learning of the network:

$$\mathbf{w} = (w^{(0)}, w^{(1)}, \ldots, w^{(K)})$$

with $w^{(0)} = Id$, $w^{(k)} \in \mathbb{M}_{d_k, d_{k+1}}(\mathbb{R})$.

- Dimension of $\mathbf{w} = D = d_0 \cdot d_1 = \cdots + d_{K-1} \cdot d_K$.
- Local Predictor:

$$v\big(\mathbf{w}, (x^0, y)\big) = \tfrac{1}{2}|y - x^K(\mathbf{w})|^2$$

- Global Predictor $V(w) = \mathbb{E}\, v\big(w, (x_I^0, y_I)\big)$, $I \sim \mathcal{U}nif(\{1, \ldots, N\})$.
- Learning phase = Calibration

$$\min_{w \in \mathbb{R}^D} V$$

- (SGD) ? Needs to differentiate V in $\mathbf{w}$ i.e. $v\big(\mathbf{w}, (x^0, y)\big)$!!

In fact three slides

- Big Question: How to differentiate (and compute!) $v\big(\mathbf{w},(x^0,y)\big)$?
- Easy part: from $v\big(\mathbf{w},(x^0,y)\big)=\frac{1}{2}|y-x^K(\mathbf{w})|^2$

$$(^\top \text{ for transpose}) \Downarrow (J \text{ for Jacobian matrix})$$

$$J_{\mathbf{w}}v\big(\mathbf{w},(x^0,y)\big)=\big(J_{\mathbf{w}}x^K(\mathbf{w})\big)^\top(x^K(\mathbf{w})-y),$$

- Use x^k as auxiliary variables and the induction

$$x^k=\varphi_k(w^{(k)},x^{k-1}),\ k=1:K.$$

- Recursive formula: note that $J_{\mathbf{w}}x^k(\mathbf{w})^\top=J_{\mathbf{w}^{(1:k)}}x^k(\mathbf{w}^{(1:k)})^\top$ and

$$\begin{aligned}J_{\mathbf{w}}x^k(\mathbf{w})^\top &= J_{w^{(k)}}\varphi_K(w^{(k)},x^{k-1})^\top + J_{\mathbf{w}}x^{k-1}(\mathbf{w})^\top\, J_x\varphi_k(w^{(k)},x^{k-1})^\top\\ &= J_{w^{(k)}}\varphi_K(w^{(k)},x^{k-1})^\top\\ &\quad + J_{\mathbf{w}^{(1:k-1)}}x^{k-1}(\mathbf{w}^{(1:k-1)})^\top\, J_x\varphi_k(w^{(k)},x^{k-1})^\top.\end{aligned}$$

Finally . . . in four slides

Lemma

In a ring $(A, +, \cdot)$ *with regular multiplication, if*

$$a_k = b_k + a_{k-1} c_k, \ k = 1 : K$$

then: $$a_K = b_K + b_{K-1} c_K + b_{K-2} c_{K-1} c_K + \cdots + b_1 c_2 \cdots c_k + a_0 c_1 \cdots c_k$$

which can be computed in a backward way.

The backpropagation algorithm ([10]) is two-fold.

- A forward step: compute the values x_j^k at each unit j of each layer k.
- A backward step: Apply the lemma to the recursion
  - Compute

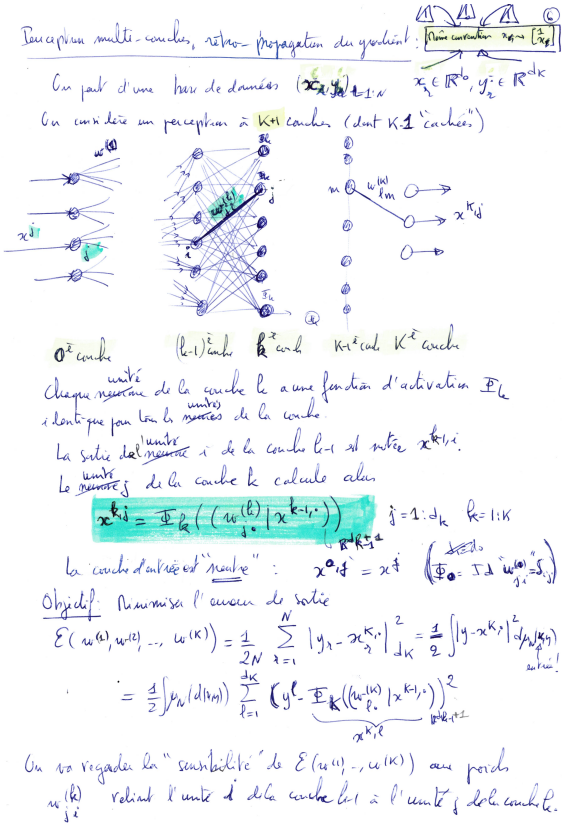
$$J_{w^{(k)}} \varphi_k(w^{(k)}, x^{k-1})^\top \quad \text{and} \quad J_{\mathbf{w}^{(1:k-1)}} x^{k-1}(\mathbf{w}^{(1:k-1)})^\top$$

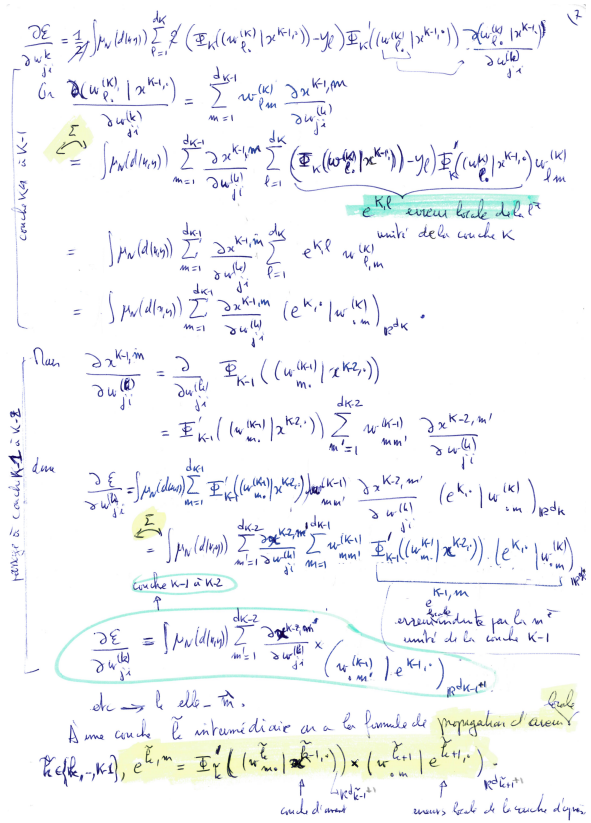
- And that's it!!

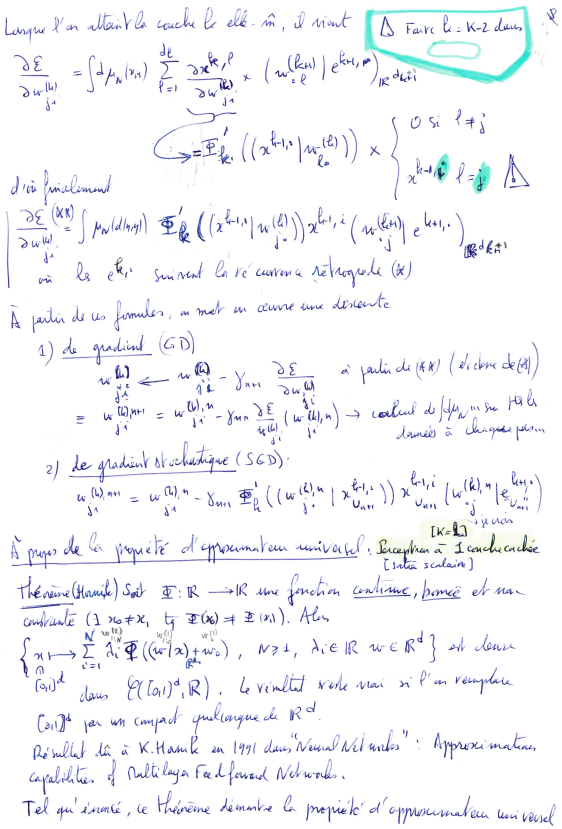
[10] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323, 533–536. But, goes back to Paul Werbos en the 1970's for AD.

The truth. . . about Backpropagation of gradient

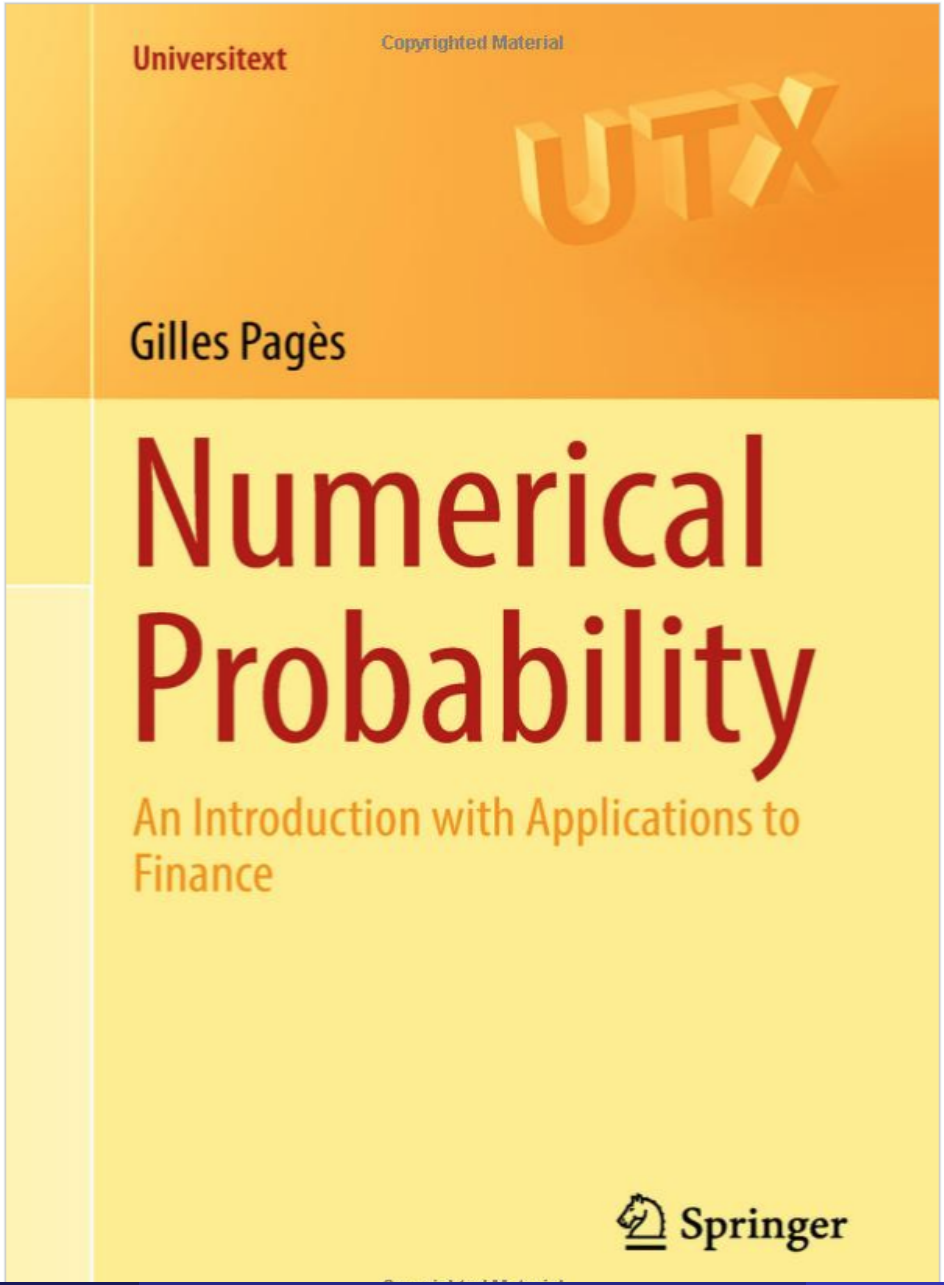
- Backpropagation of gradient is similar to the reverse mode of automatic differentiation developed independently!!
- True formulas







Ads. . .


Universitext
Copyrighted Material
UTX
Gilles Pagès
Numerical Probability
An Introduction with Applications to Finance
Springer

Grazie per l'attenzione !

Merci de votre attention !

Thank you for your attention !