# Machine learning from scratch

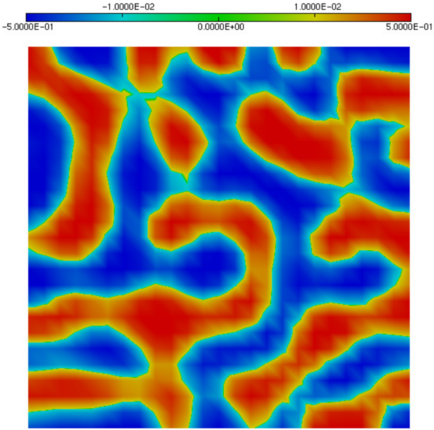
Ludovic Goudenège

CNRS, Fédération de Mathématiques de CentraleSupélec

April 20$^{th}$, 2023

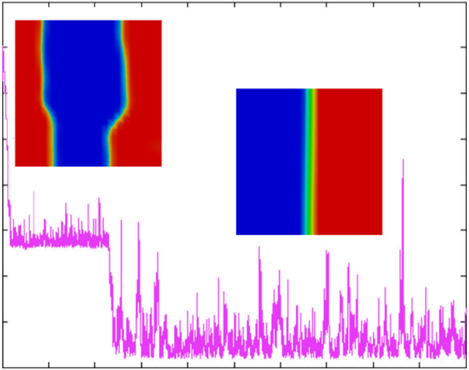
# Introduction

# Phase-field equation[1]



[1] Figures from L. Goudenège's PhD Thesis

# Phase-field equation[2]



Jump from one meta-stable well to ground state.

[2] Figures from L. Goudenège's PhD Thesis

# Convergence of random sampling

The starting point

$$S_M = \frac{1}{M} \sum_{m=1}^{M} X_m$$

where $X_m = f(Z_m)$ with $f : \mathcal{Z} \mapsto \mathbb{R}^d$.

## Theorem (Law of large numbers)

*If $\mathbb{E}[|X_1|] < +\infty$, with probability 1,*

$$\lim_{M \to +\infty} S_M = \mathbb{E}[X_1]$$

But what about the speed of convergence ?

# Speed of convergence

The convergence rate is in $1/\sqrt{M}$.
Therefore, for any $\varepsilon > 0$, with probability $1 - \varepsilon$, we have

$$|S_M - \mathbb{E}[X_1]| \leq \sqrt{Var(X_1)/(\varepsilon M)}$$

## Take-home message (First)

*The convergence rate of random estimators rate is linearly increasing with standard deviation, square-root decreasing with respect to probability and number of sampling.*

## Central Limit Theorem

We can ask the following question, is there something special in the following quantity?

$$\sqrt{M}\left(\frac{S_M - \mathbb{E}[X_1])}{\sqrt{Var(X_1)}}\right)$$

This does not seem bounded with probability 1.

Recall first take-home message: inversely proportional to the (square-root) of probability.

It is centered and normalized.

# Central Limit Theorem

## Theorem (Central Limit Theorem)

$\sqrt{M}\left(\frac{S_M-\mathbb{E}[X_1])}{\sqrt{Var(X_1)}}\right)$ *converges in distribution to a normal law.*

It is a universal theorem: it says that the limit does not depend on the law of randomness.

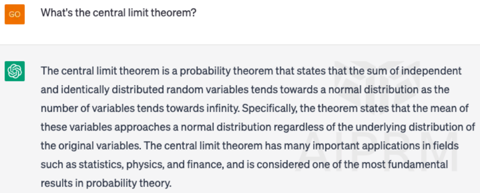
It says also that the limit is "a priori not" a random variable coming from the data.

It is a (misunderstood) fundamental theorem!

## Take-home message (Second)

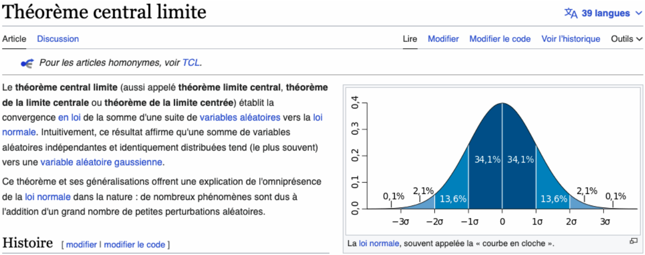
*Do not trust Wikipedia or ChatGPT (especially in maths).*

# Central Limit Theorem



GO What's the central limit theorem?

The central limit theorem is a probability theorem that states that the sum of independent and identically distributed random variables tends towards a normal distribution as the number of variables tends towards infinity. Specifically, the theorem states that the mean of these variables approaches a normal distribution regardless of the underlying distribution of the original variables. The central limit theorem has many important applications in fields such as statistics, physics, and finance, and is considered one of the most fundamental results in probability theory.



## Théorème central limite

*Pour les articles homonymes, voir TCL.*

Le **théorème central limite** (aussi appelé **théorème limite central**, **théorème de la limite centrale** ou **théorème de la limite centrée**) établit la convergence en loi de la somme d'une suite de variables aléatoires vers la loi normale. Intuitivement, ce résultat affirme qu'une somme de variables aléatoires indépendantes et identiquement distribuées tend (le plus souvent) vers une variable aléatoire gaussienne.

Ce théorème et ses généralisations offrent une explication de l'omniprésence de la loi normale dans la nature : de nombreux phénomènes sont dus à l'addition d'un grand nombre de petites perturbations aléatoires.

La loi normale, souvent appelée la « courbe en cloche ».

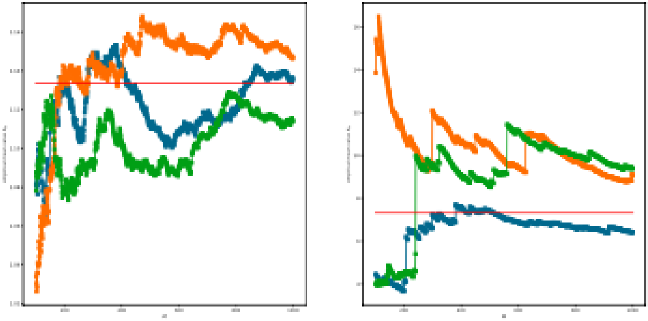
### Histoire [ modifier | modifier le code ]

# Central Limit Theorem

What's the truth?

- The convergence rate is in $1/\sqrt{M}$ is "independent" of the dimension $d$.
- The errors are random, and only characterized (asymptotically) by $\sigma^2 = Var[X_1]$ (which is unknown) or $Cov(X_1)$ in dimension $d$.
- The statistical error is larger when the variance/covariance matrix is large.
- The statistical properties (i.e. the law) of the estimator are closed to the properties of a/all/some random variable with normal law.

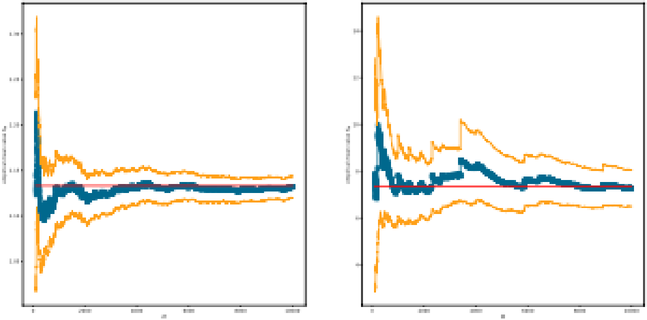
# Monte-Carlo estimations[3]



On the left, Monte-Carlo computations of $\mathbb{E}[e^{G/10}] \simeq 1.005$
On the right, Monte-Carlo computations of $\mathbb{E}[e^{2G}] \simeq 7.389$

[3] Figures from E. Gobet's lectures

# Monte-Carlo estimations[4]



One realization of the empirical mean and the corresponding 95% confidence interval.

[4] Figures from E. Gobet's lectures

# Size of Confidence Interval

If we only use the law of large number, the confidence interval are of size $O(1/\sqrt{\epsilon M})$.

This is much better with Central Limit Theorem, but this is only asymptotic.

$$\mathbb{P}(S_M - \mathbb{E}[X_1] \in CI_M) \simeq 1 - \varepsilon$$

with $CI_M = \sqrt{Var(X_1)/M}[\pm Normal\ Statistic]$.

The "Normal Statistic" is the size of the tail of a normal law, so typically $\sqrt{\log(1/\varepsilon)}$.

## Take-home message (Third)

*The Central Limit Theorem gives "asymptotic statistical bounds" for the exact value, not for $S_M$.*

# The problems

# Bias in substitution method

Suppose that you want to compute $\phi(\mathbb{E}[X_1])$ for some function $\phi$.

Since $S_M$ converges almost surely to $\mathbb{E}[X_1]$ we expect $\phi(S_M) \simeq \phi(\mathbb{E}[X_1])$.

But the first problem is that there is bias!

$$\mathbb{E}[\phi(S_M)] \neq \phi(\mathbb{E}[X_1]).$$

However, we can quantify the bias in the substitution method

$$\mathbb{E}[\phi(S_M)] - \phi(\mathbb{E}[X_1]) = \frac{c_1}{M} + \frac{c_2}{M^2} + o(M^{-2})$$

if $\phi$ is in $\mathcal{C}_b^4$, and $X_1$ satisfies some moment conditions.

## Remark

*If $\phi$ is convex (respectively concave), the substitution method gives an overestimation (respectively underestimation) of $f(\mathbb{E}[X_1])$.*

# Curse of dimensionality

You have to understand that stochastic optimization in a high dimensional space is a difficult problem for a very simple reason. Assume that you have a very large number $M$ of realizations of uniform random variables in a cube $[0,1]^d$, say $X_1, \ldots, X_M$.

Could you expect that the next random number $X_{M+1}$ is close to the others ? Let's define

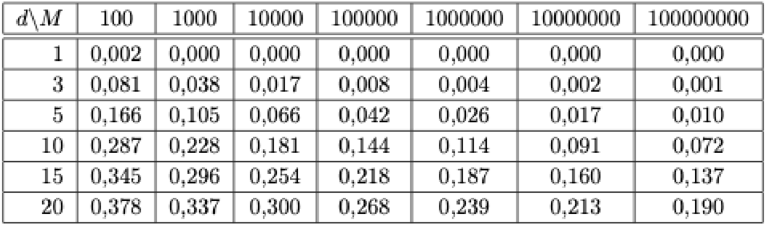
$$\mathcal{D}(d, M) = \mathbb{E}\left[\min_{m=1,\ldots,M} |X_{M+1} - X_M|_\infty\right]$$

which is the expected distance to the nearest neighbors.

## Take-home message (Fourth)

*In dimension 21 (7 atoms with 3 coordinates), with* $M = 100.000.000$*, then* $\mathcal{D}(21, 10^8) \simeq 20\%$*.*

# Curse of dimensionality

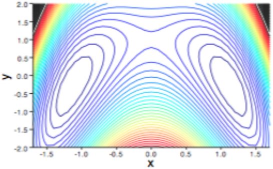


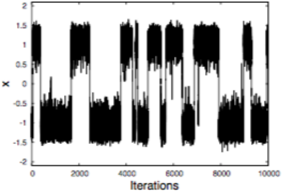
| $d\backslash M$ | 100 | 1000 | 10000 | 100000 | 1000000 | 10000000 | 100000000 |
|---|---|---|---|---|---|---|---|
| 1 | 0,002 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 |
| 3 | 0,081 | 0,038 | 0,017 | 0,008 | 0,004 | 0,002 | 0,001 |
| 5 | 0,166 | 0,105 | 0,066 | 0,042 | 0,026 | 0,017 | 0,010 |
| 10 | 0,287 | 0,228 | 0,181 | 0,144 | 0,114 | 0,091 | 0,072 |
| 15 | 0,345 | 0,296 | 0,254 | 0,218 | 0,187 | 0,160 | 0,137 |
| 20 | 0,378 | 0,337 | 0,300 | 0,268 | 0,239 | 0,213 | 0,190 |

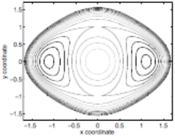
An approximated formula gives

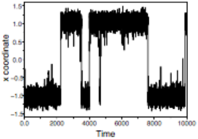
$$\mathcal{D}(d, M) \geq \frac{d}{2(d+1)} \frac{1}{M^{\frac{1}{d}}}$$

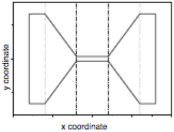
# Meta-stable dynamics in small dimension

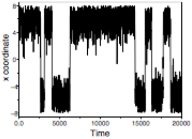












# Stochastic optimization

You want to minimize over $\theta \in \Theta \subset \mathbb{R}^p$ the function $\Phi(\theta) = \mathbb{E}[\phi(X_1, \theta)]$. You certainly have a Monte-Carlo approximation

$$\hat{\Phi}(\theta) = \frac{1}{M} \sum_{m=1}^{M} \phi(X_m, \theta)$$

The aim is to find a bound on the error

$$|\mathrm{argmin}_{\theta \in \Theta} \Phi(\theta) - \mathrm{argmin}_{\theta \in \Theta} \hat{\Phi}(\theta)|$$

as a function of the number of sampling size $M$.

Again, you cannot hope better than a bound $\frac{c}{\sqrt{M}}$. For instance

$$\Phi(\theta) = \lambda \|\theta\|^2 + \mathbb{E}[|Y - X^T\theta|] \text{ and } \hat{\Phi}(\theta) = \lambda \|\theta\|^2 + \frac{1}{M} \sum_{m=1}^{M} |Y_m - X_m^T \theta|.$$

# Regression

Assume the space of regression functions is of dimension $K$. $\Psi := Vect(\psi_1, \dots, \psi_K)$. Define

$$\mathcal{M}_M := \mathrm{argmin}_{\psi \in \Psi} \frac{1}{M} \sum_{m=1}^{M} |O^m - \psi(I^m)|^2$$

then the Mean Square Error satisfies

$$MSE = \mathbb{E}[|\mathcal{M}_M - \mathcal{M}|^2_{\mu_M}] \leq \inf_{\psi \in \Psi} |\psi - \mathcal{M}|^2_{\mu} + \frac{K}{M} \sup_{i \in \mathbb{R}^d} Var(O|I = i)$$

## Take-home message (Fifth)

*There is always two parts in a MSE. First one is about "structure", the second is a "Variance" term.*

# Kernel regression

Make the link with the talk of Arthur France-Lanord.
You want to approximate

$$s(\xi) = \frac{\sum_{i=1}^{K} \theta_i \ K(\xi, \xi_i)}{\sum K(\xi, \xi_i)}$$

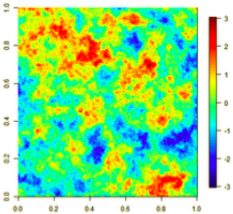
with

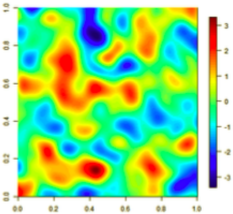
$$\hat{\theta} = \mathrm{argmin} \ \|y - K\theta\|^2 + \lambda \theta^T K \theta.$$

# Kernels

Exponential covariance function versus Gaussian covariance function

$$K(x,y) = \exp^{-\frac{|x-y|}{\ell}} \qquad K(x,y) = \exp^{-\frac{|x-y|^2}{2\ell^2}}$$





Interaction length : $\ell > 0$.

# Maximum likelihood estimation

# Maximum likelihood estimation

Consider a family $\mu_\theta$ of distributions with density (easy to sample for any $\theta$).

You have many observations sampled from the unknown distribution $\mu^*$.

Ideally, you want to minimize a distance or divergence (typically Kullback-Leibler)

$$KL(\mu^* || \mu_\theta) = -\int \log\left(\frac{d\mu_\theta}{d\mu^*}\right) d\mu^*$$

# Maximum likelihood estimation

If $\mu^*$ has also a density with respect to Lebesgue measure, this is equivalent to maximize

$$\theta \mapsto \int \log\left(\frac{d\mu_\theta}{dLeb}\right) d\mu^*.$$

A classical approach is to replace the integral by an empirical one

$$\theta \mapsto \sum_{m=1}^{M} \log(p_\theta(X_m)).$$

# Density estimation

First solution

$$\mu_\theta = (T_\theta)_\# \nu_0$$

where $\nu_0$ is a reference probability with density $q_0$, and $T_\theta$ is a $\mathcal{C}^1$ diffeomorphism.

In that case, we have

$$p_\theta(x) = q_0(T_\theta^{-1}(x)) Jac_x[T_\theta^{-1}](x)$$

Example: $T_\theta(x) = m + \Sigma x$ with $\theta = (m, \Sigma)$.

Approximating the function $p(x)$ is called the density estimation problem.

# Stochastic gradient descent

But, how to optimize?

Stochastic gradient descent:

$$\theta^{(n+1)} = \theta^{(n)} - \gamma^{(n+1)} \sum_{X_m \in B^{(n+1)}} \nabla_\theta \log p_\theta(X_m)$$

where

- $B^{(n+1)}$ is a sequence of random batch of data points,
- $\gamma^{(n+1)}$ is a sequence of stepsizes/learning rates.

# In the past...

People thought it was hard to solve the following problem:

$$\text{Find } \{\, T_\theta : \theta \in \Theta \}$$

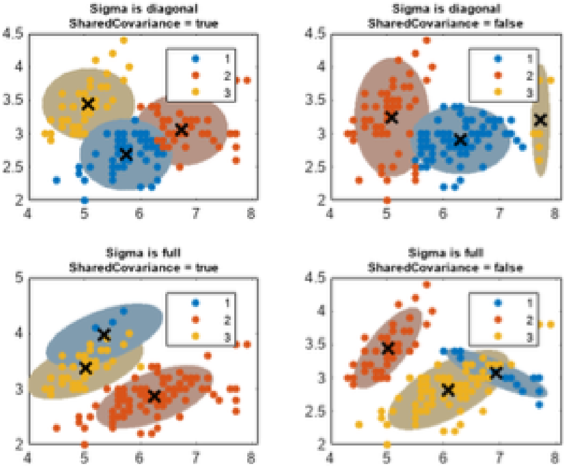
such that

$$\theta \mapsto \sum_{m=1}^{M} \log(p_\theta(X_m))$$

is easy to optimize.

Today we know that such construction are possible using neural networks, but this is not the only ones.

# Gaussian Mixture Models



# Gaussian Mixture Models

Latent variable models :
We consider that the data $(X_m)_{m=1}^M$ are i.i.d. from

$$C \sim Cate(\omega_1, \dots, \omega_K), \text{ with } \sum_{k=1}^{K} \omega_k = 1$$

$$X|C \sim \mathcal{N}(\mathrm{m}_C, \Sigma_C)$$

with $\mathrm{m}_1, \dots, \mathrm{m}_K \in \mathbb{R}^d$ and $\Sigma_1, \dots, \Sigma_K$ are squared positive definite matrix.

Here the parameters are $\theta = \{\omega_k, \mathrm{m}_k, \Sigma_k\}_{k=1}^K$.

# Likelihood

The likelihood for one observation is then the marginal

$$\begin{aligned} p_\theta(X_m) &= \sum_{k=1}^K p_\theta(X_m, k) = \sum_{k=1}^K p_\theta(X_m|k) p_\theta(k) \\ &= \sum_{k=1}^K \omega_k \phi(X_m|m_k, \Sigma_k) \end{aligned}$$

where $\phi(X_m|m_k, \Sigma_k)$ is th density of $\mathcal{N}(m_k, \Sigma_k)$.

The complete log-likelihood is

$$\frac{1}{M} \sum_{m=1}^M \log(p_\theta(X_m)) = \frac{1}{M} \sum_{m=1}^M \log \left( \sum_{k=1}^K \omega_k \phi(X_m|m_k, \Sigma_k) \right)$$

# Expectation-Maximization algorithm

By Jensen inequality, for any observation $X_m$, and $\theta, \theta'$

$$
\begin{aligned}
\log(p_\theta(X_m)) &= \log\left(\sum_{k=1}^{K} p_\theta(X_m, z)\right) \\
&\geq \sum_{k=1}^{K} p_{\theta'}(k|X_m) \log\left(\frac{p_\theta(X_m, k)}{p_{\theta'}(k|X_m)}\right).
\end{aligned}
$$

This suggests a fixed-point algorithm. Given $\theta^{(n)}$ at step $n$

E step: Compute $\psi^{(n)}(\theta) = \sum_{k=1}^{K} p_{\theta^{(n)}}(k|x_m) \log\left(\frac{p_\theta(X_m,k)}{p_{\theta^{(n)}}(k|X_m)}\right)$.

M step: Maximize $\psi^{(n)}(\theta)$ and define $\theta^{(n+1)} = \mathrm{argmax}_\theta \psi^{(n)}(\theta)$.

In a Gaussian Mixture Model, these two steps are fully explicit. The likelihood for one observation is

$$L(\theta; X_m) = p_\theta(X_m) = \sum_{k=1}^{K} \omega_k \phi(X_m | \mathrm{m}_k, \Sigma_k),$$

and the complete log-likelihood of the model is

$$\begin{aligned} \log L(\theta; x, k) &= \log \left( \prod_{m=1}^{M} \sum_{k=1}^{K} \omega_k \phi(X_m | \mathrm{m}_k, \Sigma_k) \right) \\ &= \sum_{m=1}^{M} \log \left( \sum_{k=1}^{K} \omega_k \phi(X_m | \mathrm{m}_k, \Sigma_k) \right). \end{aligned}$$

- In the E-step, we have to compute

$$\psi^{(n)}(\theta) = \sum_{m=1}^{M} \sum_{k=1}^{K} p_{\theta^{(n)}}(k|X_m) \log \frac{p_\theta(X_m, k)}{p_{\theta^{(n)}}(k|X_m)}$$

- In the M-step, we have to maximize $\psi^{(n)}$ over $\theta$.

### Take-home message (Sixth)

*You can write the code to apply EM algorithm on a Gaussian Mixture Model.*

Thanks to the form of the function $\psi^{(n)}$, the resulting expressions for the new parameters are

$$\omega_k^{(n+1)} = \frac{1}{M} \sum_{m=1}^{M} \frac{\omega_k^{(n)} \phi(X_m | \mathrm{m}_k^{(n)}, \Sigma_k^{(n)})}{\mathcal{Z}_m},$$

$$\mathrm{m}_k^{(n+1)} = \frac{1}{M\omega_k^{(n+1)}} \sum_{m=1}^{M} \frac{\omega_k^{(n)} \phi(X_m | \mathrm{m}_k^{(n)}, \Sigma_k^{(n)}) X_i}{\mathcal{Z}_m},$$

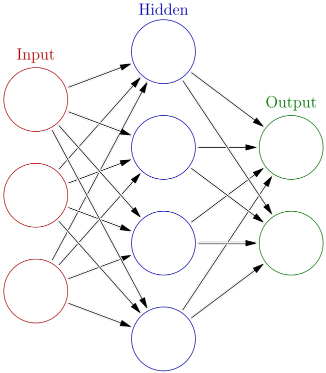
and

$$\Sigma_k^{(n+1)} = \sum_{m=1}^{M} \frac{\omega_k^{(n)} \phi(X_m | \mathrm{m}_k^{(n)}, \Sigma_k^{(n)})(X_m - \mathrm{m}_k^{(n+1)})(X_m - \mathrm{m}_k^{(n+1)})^T}{M\omega_k^{(n+1)} \mathcal{Z}_m},$$

where

$$\mathcal{Z}_m = \sum_{k=1}^{K} \omega_k^{(n)} \phi(X_m | \mathrm{m}_k^{(n)}, \Sigma_k^{(n)}).$$

# Neural networks



# Neural networks

A Neural Network $T_\theta$ is a composition of $L$ transformations $(T^\ell)_{\ell=1,\dots,L}$

$$T_\theta = T^L \circ \dots \circ T^1,$$

where, for each $\ell$, $T^\ell : \mathbb{R}^{d^\ell} \to \mathbb{R}^{d^{\ell+1}}$ is given by

$$T^\ell(x) = \sigma^\ell\left(W^\ell x + b^\ell\right).$$

The matrices $W^\ell \in \mathbb{R}^{d^{\ell+1} \times d^\ell}$ are called the network weights, the vectors $b^\ell \in \mathbb{R}^{d^{\ell+1}}$ the network biases.

The activation function $\sigma^\ell$ acts componentwise,

$$\sigma^\ell(a) = (\sigma^\ell(a_1), \dots, \sigma^l(a_{d^{\ell+1}})).$$

# Neural networks

We will construct a Neural Network $T_\theta(x)$ and then, given some synthetic data $(X_m, Y_m)_{m=1,\dots,M}$, solve the fitting (or training) problem

$$\min_\theta R_M(\theta) \quad \text{where} \quad R_M(\theta) = \frac{1}{M}\sum_{m=1}^{M}(Y_m - T_\theta(X_m))^2$$

applying a gradient descent algorithm

$$\theta^{(n+1)} = \theta^{(n)} - \gamma^{(n+1)} \frac{1}{\left|B^{(n+1)}\right|} \sum_{m\in B^{(n+1)}} \nabla_\theta \left(Y_m - T_{\theta^{(n)}}(X_m)\right)^2$$

with step (or "learning rate") $\gamma$, where $(B^{(n)})_n$ is a sequence of batches.

# Back propagation

This procedure will require to implement the computation of the gradient $\nabla_\theta T(X_m)$, which can be done by backpropagation through the network.

# Proof of the theorem of Hornick, Cybenko et al.

**Theorem (Universal approximation theorems)**

*"Any reasonable function $f$ can be approximated with an arbitrary accuracy by a neural network with some number of hidden layers and for some activation function $\sigma$".*

- Gallant, White - 1988: $f$ is square integrable on $[0, 2\pi]^d$, and the activation function is the cosine squasher
- Cybenko - 1989: uniform convergence for any continuous function on a compact set, $\sigma$ is the general sigmoid squasher
- Hornik - 1990: the same but assuming that $\sigma$ is continuous, bounded and non-constant. Does not include the ReLU case.
- Pinkus - 1993: similar result, for any activation function $\sigma$ that is not a polynomial. Includes the ReLU case.

# Proof of the theorem of Hornick, Cybenko et al.

Is it a primitive function ?

$$f(x) = \int_{-\infty}^{x} f'(y)dy = \int_{\mathbb{R}} H(x-y)f'(y)dy$$

Can we approximated the Heaviside function $H$ at point $(x-y)$?

$$\int_{\mathbb{R}} H(x-y)f'(y)dy \simeq \sum_{j=-J}^{J} \sigma\left(\frac{x}{\varepsilon} - \frac{j\Delta x}{\varepsilon}\right) f'(j\Delta x)$$

$$f(x) \simeq \sum_{j=-J}^{J} \omega_j \sigma(a_j x + b_j).$$

## Take-home message (Seventh and last one)

*You know how to prove an easy version of the Universal Approximation Theorem.*

Thanks for your attention.

Do not forget the take-home messages.

- Cours d'Emmanuel Gobet. Méthodes de Monte-Carlo. Polytechnique.
- Cours d'Alain Durmus. Méthodes de Monte-Carlo. Polytechnique.
- Communications with Tony Lelièvre.