TD 8: Optimization for learning, Neural Networks.

Exercice 8.1. In this exercise, $\|x\|$ refers to the Euclidean norm of x, for $x \in \mathbb{R}^p$.

Let $f : \mathbb{R}^p \to \mathbb{R}$ be a twice continuously differentiable function (f is of class C^2). $\forall x \in \mathbb{R}^p$, we can thus calculate its gradient as :

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_p}(x) \end{pmatrix} ,$$

and its Hessian matrix :

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_p}(x) \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_p}(x) & \dots & \frac{\partial^2 f}{\partial x_p^2}(x) \end{pmatrix} .$$

Let $x, h \in \mathbb{R}^p$. We then have Taylor's formula of order 1 :

$$f(x+h) = f(x) + \nabla f(x)^T h + o(\|h\|)$$

and of order 2 :

$$f(x+h) = f(x) + \nabla f(x)^T h + \frac{1}{2} h^T \nabla^2 f(x) h + o\left(\|h\|^2\right).$$

f is called convex if

$$\forall x, h \in \mathbb{R}^p, \ f(x+h) \geq f(x) + \nabla f(x)^T h.$$

f is convex if and only if one of the following equivalent conditions is satisfied :

(i) $\forall x, h \in \mathbb{R}^p : \quad (\nabla f(x+h) - \nabla f(x))^T h \geq 0.$

(ii) $\forall x \in \mathbb{R}^p : \quad \nabla^2 f(x)$ is positive semi-definite (we denote $\nabla^2 f(x) \geq 0$).

f is said to be **strongly convex** if there is $\mu > 0$ such that :

$$\forall x, h \in \mathbb{R}^p, \ f(x+h) \geq f(x) + \nabla f(x)^T h + \frac{1}{2}\mu \|h\|^2.$$

μ is called parameter of strong convexity.

f is strongly convex with parameter μ if and only if one of the following equivalent conditions is satisfied:

(i) $\forall x, h \in \mathbb{R}^p : \quad (\nabla f(x+h) - \nabla f(x))^T h \geq \mu \|h\|^2.$

(ii) $\forall x \in \mathbb{R}^p : \quad \nabla^2 f(x) - \mu I_p$ is positive semi-definite (we denote $\nabla^2 f(x) \geq \mu I_p$).

In the following, we will assume that f is strongly convex with parameter μ. We will admit that as a consequence, f has a unique minimum $\bar{x}$ on $\mathbb{R}^p$ and that a necessary and sufficient condition for f to atteign its minimum in $\bar{x}$ is that $\nabla f(\bar{x}) = 0$.

1. a) Let $A \in M_{np}(\mathbb{R})$, $b \in \mathbb{R}^n$. Show that $f : f(x) = \|Ax - b\|^2$ is convex. Show that it is strongly convex if A is injective, and then give its strong convexity parameter.

b) Check that if f is convex, then $g(x) = f(x) + \lambda\|x\|^2$ is strongly convex. In which statistical learning situation, this property could be applied ?

2. a) Let $x \in \mathbb{R}^p$. If $\nabla f(x) \neq 0$, show that there exists $\rho > 0$ such that :

$$f(x - \rho\nabla f(x)) < f(x).$$

b) Show that if ∇f is M-Lipschitz continuous, then there is $\rho > 0$ such that the map $h_\rho : x \mapsto x - \rho\nabla f(x)$ is contractive. Deduce an algorithm to minimize the function f. What is its convergence rate ?

3. It is now assumed that f is written as :

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x) \ .$$

and we want to determine $\bar{x}$ that minimizes f.

a) Is this a common situation in statistical learning? What can then be a problem related to the use of the gradient algorithm proposed in question **1.)** ?

b) Let J be a random variable with uniform distribution defined on $\{1; 2; \ldots; N\}$. Show that $\forall x \in \mathbb{R}^p$:

$$\mathbb{E}_J\left(\nabla f_J(x)\right) = \nabla f(x).$$

Based on this observation, a so-called stochastic gradient algorithm is proposed. Let x_0 be given. At each iteration k :

(i) a random draw of the variable J is made, which is denoted by j_k.

(ii) we take : $x_{k+1} = x_k - \rho_k \nabla f_{j_k}(x_k)$.

The algorithm stops when $\|x_{k+1} - x_k\| < \epsilon$.

We suppose that $\forall 1 \leq j \leq N, \forall x \in \mathbb{R}^p$, $\|\nabla f_j(x)\| \leq B$.

c) Show that :

$$\mathbb{E}_{J_k}\left[\|x_{k+1} - \overline{x}\|^2\right] \leq (1 - 2\rho_k\mu)\|x_k - \overline{x}\|^2 + \rho_k^2 B^2 \qquad (1)$$

where E_{J_K} corresponds to the expectation with respect to the k-th draw of the variable J.

d) When ρ_k is constant, $\forall k : \ \rho_k = \rho$, show that :

$$\mathbb{E}\left[\|x_{k+1} - \overline{x}\|^2\right] \leq (1 - 2\rho\mu)^{k+1}\|x_0 - \overline{x}\|^2 + \frac{\rho}{2\mu}B^2$$

where the expectation is now taken with respect to the random vector $(J_0, J_1, \ldots, J_K)$. Verify that this equality does not lead to the conclusion that the algorithm converges.

e) Using equation 1, show that quadratic convergence is well ensured if we choose : $\rho_k = \frac{1}{\mu(k+1)}$.
For this, we can prove by induction that $\forall k$:

$$\mathbb{E}\left[\|x_k - \overline{x}\|^2\right] \leq \frac{\max\left(\|x_0 - \overline{x}\|^2, \frac{B^2}{\mu^2}\right)}{k+1}.$$

f) What is the problem with this method ? Compare the convergence rate with the classical gradient. Show that we could have better performance if at each iteration we took : $x_{k+1} = x_k - \rho_k g_k$ with g_k a stochastic approximation of the gradient, such that $\mathbb{E}(g_k) = \nabla f(x_k)$ and such that its variance can be controlled by $\mathbb{E}\left(\|g_k - \nabla f(x_k)\|^2\right) \leq L\|x_k - \bar{x}\|^2$.

Solution: 1. a)

$f(x+h) = (Ax + Ah - b)^T(Ax + Ah - b) = f(x) + 2h^T A^T(Ax - b) + h^T A^T Ah$.

By the uniqueness of Taylor series formula :

$$\nabla f(x) = 2A^T(Ax - b) \quad \text{et} \quad \nabla^2 f(x) = 2A^T A \ .$$

$\nabla^2 f(x)$ is symmetric and positive, so f is convex.
If A is injective, then $A^T A$ is positive definite : $x^T A^T A^T Ax = \|Ax\|^2 \geq 0$, and zero only if $Ax = 0$ that leads to $x = 0$ since A is injective. So $\nabla^2 f(x)$ is symmetric and positive definite, and according to the spectral theorem, all eigenvalues are strictly positive. Hence, if λ_{min} is the smallest eigenvalues, then $\forall h \in \mathbb{R}^p$:

$$h^T \nabla^2 f(x) h \geq \lambda_{min}\|h\|^2 \ .$$

Therefore f is strongly convex with parameter λ_{min}.

b) We have $\nabla^2 g(x) = \nabla^2 f(x) + \lambda I_p$. So if f is convex, then g is strongly convex. This situation may correspond to learning problems with ridge penalty, when the loss function is convex. In the case of linear regression with non-injective A, we are in the situation where f is convex, but not strongly convex. The penalized criterion is then strongly convex.

2. a) By applying the Taylor series formula of order 1, we get :

$$f(x - \rho\nabla f(x)) = f(x) - \rho\nabla f(x)^T \nabla f(x) + o(\rho\|\nabla f(x)\|) \ .$$

We denote $\alpha(\rho) = o(\rho\|\nabla f(x)\|)$, we then have $\dfrac{\alpha(\rho)}{\rho} \underset{\rho\to 0}{\longrightarrow} 0$. Hence :

$$f(x - \rho\nabla f(x)) = f(x) - \rho\left(\nabla f(x)^T \nabla f(x) - \frac{\alpha(\rho)}{\rho}\right) \ .$$

$\nabla f(x)^T \nabla f(x) > 0$, so we can find a ρ small enough such that :

$$f(x - \rho \nabla f(x)) < f(x) \ .$$

b) Let $x, y \in \mathbb{R}^p$:

$$\begin{aligned} \|h_\rho(y) - h_\rho(x)\|^2 &= \|y - x - \rho(\nabla f(y) - \nabla f(x))\|^2 \\ &= \|y - x\|^2 - 2\rho(\nabla f(y) - \nabla f(x))^T (y - x) + \rho^2 \|\nabla f(y) - \nabla f(x)\|^2 \end{aligned}$$

From the strong convexity, we have $(\nabla f(y) - \nabla f(x))^T (y - x) \geq \mu \|y - x\|^2$ and since the gradient is M-Lipschitz, then :

$$\|\nabla f(y) - \nabla f(x)\| \leq M \|y - x\|.$$

And finally :

$$\|h_\rho(y) - h_\rho(x)\|^2 \leq \|y - x\|^2 \left(1 - 2\rho\mu + \rho^2 M^2\right).$$

If $\rho < \dfrac{2\mu}{M^2}$, $|1 - 2\rho\mu + \rho^2 M^2| < 1$ then h_ρ is indeed a contraction. The minimum of the upper bound (supremum) is reached for $\rho = \frac{\mu}{M^2}$. It admits a fixed point $\bar{x}$ such that $\nabla f(\bar{x}) = 0$ which is thus the minimum of f.

We deduce the minimization algorithm as follows :

For any $x_0 \in \mathbb{R}^p$, we define : $x_{k+1} = x_k - \rho \nabla f(x_k)$, with $\rho = \frac{\mu}{M^2}$. We stop the algorithm when $\|x_{k+1} - x_k\|$ is small enough.

We then have :

$$\|x_{k+1} - \bar{x}\| \leq \left(1 - \frac{\mu^2}{M^2}\right)^{\frac{1}{2}} \|x_k - \bar{x}\|.$$

We define the convergence rate as linear (or geometric). And :

$$\|x_k - \bar{x}\| \leq \left(1 - \frac{\mu^2}{M^2}\right)^{\frac{k}{2}} \|x_0 - \bar{x}\|.$$

NB: $\frac{\mu}{M}$ corresponds to $\frac{\lambda_{min}}{\lambda_{max}}$ in the case of linear regression, which is actually related to the conditioning of the $A^T A$ matrix: the closer the ratio $\frac{\lambda_{min}}{\lambda_{max}}$ is to 1, the faster the convergence will be, corresponding to a good conditioning of the matrux. Reversely, for ill-conditione matrices, there is a big difference between the eigenvalues, and the convergence will be slow. This can be generalized to any situations by considering the conditioning of the hessian.

3. a) In supervised learning : for a data set $(x_i, y_i)_{1 \leq i \leq N}$, we generally have to determine the parameters θ that minimize the empirical risk of a loss function associated with a parametric decision function h. Empirical risk is then written as :

$$R(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, h(\theta, x_i))$$

The minimization of the opposite of log-likelihood is also often written in this way.

We are in the expected form by taking : $f_i(\theta) = L\left(y_i, h(\theta, x_i)\right)$

For example, for the linear regression with ridge penalty :

$$f_i(\beta_0, \beta) = \left(\beta_0 + \beta^T x_i - y_i\right)^2 + \lambda \|(\beta_0, \beta)\|_2^2$$

or for logistic regression :

$$f_i(\beta_0, \beta) = -y_i \left(\beta_0 + \beta^T x_i\right) + \ln\left(1 + \exp\left(\beta_0 + \beta^T x_i\right)\right) \ .$$

A difficulty is the potentially large amount of data (N large) and therefore the cost of calculating all the gradients $\nabla f_i(x^k)$ at each iteration of the algorithm.

b)

$$\mathbb{E}_J\left(\nabla f_J(x)\right) = \sum_{i=1}^{N} \mathbb{P}(J = i) \nabla f_i(x) = \sum_{i=1}^{N} \frac{1}{N} \nabla f_i(x) = \nabla f(x) \ .$$

Hence, we get the idea of using $\nabla f_J(x)$ as a stochastic approximation of the gradient at each iteration.

c) We denote $\rho_k = \rho$

$$\begin{aligned} \|x_{k+1} - \bar{x}\|^2 &= \|x_k - \bar{x} - \rho \nabla f_{j_k}(x_k)\|^2 \\ &= \|x_k - \bar{x}\|^2 - 2\rho \left\langle \nabla f_{j_k}(x_k), x_k - \bar{x} \right\rangle + \rho^2 \left\|\nabla f_{j_k}(x_k)\right\|^2 \end{aligned}$$

We compute the expectation :

$$\begin{aligned} \mathbb{E}_{j_k}\left[\|x_{k+1} - \bar{x}\|^2\right] &= \|x_k - \bar{x}\|^2 - 2\rho \nabla f(x_k)^T (x_k - \bar{x}) + \rho^2 \mathbb{E}_{j_k}\left[\left\|\nabla f_{j_k}(x_k)\right\|^2\right] \\ &\leq \|x_k - \bar{x}\|^2 - 2\rho \nabla f(x_k)^T (x_k - \bar{x}) + \rho^2 B^2 \end{aligned}$$

from the strong convexity of f:

$$\nabla f(x_k)^T (x_k - \bar{x}) = \left(\nabla f(x_k) - \nabla f(\bar{x})\right)^T (x_k - \bar{x}) \geq \mu \|x_k - \bar{x}\|^2,$$

and so :

$$\mathbb{E}_{J_k}\left[\|x_{k+1} - \bar{x}\|^2\right] \leq (1 - 2\rho\mu) \|x_k - \bar{x}\|^2 + \rho^2 B^2$$

d) Now, by computing the expectation with respect to the random vector $(J_0, J_1, \ldots, J_K)$ and iterating :

$$\begin{aligned} \mathbb{E}\left[\|x_{k+1} - \bar{x}\|^2\right] &\leq (1 - 2\rho\mu) \mathbb{E}\left[\|x_k - \bar{x}\|^2\right] + \rho^2 B^2 \\ &\leq (1 - 2\rho\mu)^2 \mathbb{E}\left[\|x_{k-1} - \bar{x}\|^2\right] + (1 - 2\rho\mu)\rho^2 B^2 + \rho^2 B^2 \\ &= (1 - 2\rho\mu)^{k+1} \|x_0 - \bar{x}\|^2 + \sum_{i=0}^{k} (1 - 2\rho\mu)^i \rho^2 B^2 \end{aligned}$$

Since,

$$\sum_{i=0}^{k} (1 - 2\rho\mu)^i = \frac{1 - (1 - 2\rho\mu)^{k+1}}{2\rho\mu} \leq \frac{1}{2\rho\mu}.$$

Then finally :

$$\mathbb{E}\left[\|x_{k+1} - \bar{x}\|^2\right] \leq (1 - 2\rho\mu)^{k+1} \|x_0 - \bar{x}\|^2 + \frac{\rho}{2\mu} B^2$$

$\mathbb{E}\left[\|x_{k+1} - \bar{x}\|^2\right]$ does not tend a priori towards 0.

e) Suppose that $\rho_k = \frac{1}{\mu(k+1)}$.

$$\mathbb{E}_{J_k}\left[\|x_{k+1} - \overline{x}\|^2\right] \leq (1 - 2\rho_k\mu) \|x_k - \overline{x}\|^2 + \rho_k^2 B^2$$

Note that $L = \max\left(\|x_0 - \overline{x}\|^2, \frac{B^2}{\mu^2}\right)$. We then surely have

$$\mathbb{E}\left[\|x_0 - \overline{x}\|^2\right] = \|x_0 - \overline{x}\|^2 \leq L.$$

Suppose that

$$\mathbb{E}\left[\|x_k - \overline{x}\|^2\right] \leq \frac{L}{k+1} \ .$$

So :

$$\begin{aligned}
\mathbb{E}\left[\|x_{k+1} - \overline{x}\|^2\right] &\leq (1 - 2\rho_k\mu)\, \mathbb{E}\big[\, \|x_k - \overline{x}\|^2 \,\big] + \rho_k^2 B^2 \\
&\leq \left(1 - \frac{2}{k+1}\right) \mathbb{E}\big[\, \|x_k - \overline{x}\|^2 \,\big] + \frac{1}{(k+1)^2} \frac{B^2}{\mu^2} \\
&\leq \left(1 - \frac{2}{k+1}\right) \frac{L}{k+1} + \frac{L}{(k+1)^2} \\
&\leq \frac{L}{k+1}\left(1 - \frac{1}{k+1}\right) \\
&\leq \frac{L}{k+1}\left(1 - \frac{1}{k+2}\right) \\
&\leq \frac{L}{k+2}
\end{aligned}$$

The algorithm therefore converges, however the rate of convergence is sublinear in $O\left(\frac{1}{k}\right)$.

f) We can write :

$$x_{k+1} = x_0 - \sum_{i=0}^{k} \frac{1}{\mu(i+1)} \nabla f_{j_i}(x_i) \ .$$

It seems counter-intuitive that the gradients at the new points are less important in the decomposition when they are closer to the target.

So: for the classical gradient, we have the linear or geometric convergence rate $\left(1 - \frac{\mu^2}{M^2}\right)$, while for the stochastic gradient, we get sublinear convergence in $O\left(\frac{1}{k}\right)$.

We take : $x_{k+1} = x_k - \rho g_k$. The same calculations as for the c) are done here :

$$\|x_{k+1} - \bar{x}\|^2 = \|x_k - \bar{x} - \rho g_k\|^2 = \|x_k - \bar{x}\|^2 - 2\rho g_k^T (x_k - \bar{x}) + \rho^2 \|g_k\|^2$$

We compute the expectation (with respect to the stochastic construction of the gradient at k-th iteration):

$$\begin{aligned}\mathbb{E}\left[\|x_{k+1}-\bar{x}\|^2\right] &= \|x_k-\bar{x}\|^2 - 2\rho\nabla f\left(x_k\right)^T\left(x_k-\bar{x}\right) + \rho^2\mathbb{E}\left[\|g_k\|^2\right] \\ &\leq \|x_k-\bar{x}\|^2 - 2\rho\nabla f\left(x_k\right)^T\left(x_k-\bar{x}\right) + \rho^2\mathbb{E}\left(\|\nabla f(x_k)\|^2\right) + \rho^2\mathbb{E}\left(\|g_k-\nabla f(x_k)\|^2\right) \\ &\leq \|x_k-\bar{x}\|^2 - 2\rho\nabla f\left(x_k\right)^T\left(x_k-\bar{x}\right) + \rho^2\|\nabla f(x_k)-\nabla f(\bar{x})\|^2 + \rho^2\mathbb{E}\left(\|g_k-\nabla f(x_k)\|^2\right) \\ &\leq \|x_k-\bar{x}\|^2\left(1-2\rho\mu+\rho^2M^2+\rho^2L\right)\end{aligned}$$

We then find a linear convergence of the sequence towards $\bar{x}$. Several methods have been proposed to achieve these conditions (for example stochastic average gradient descent by Le Roux et al., NIPS 2012).

Exercice 8.2. We consider a neural network defined from $\mathbb{R}^I$ to $\mathbb{R}$, with a hidden layer of H neurons and an activation function given by $\tanh$.

Let $x=(x_1,\ldots,x_I)\in\mathbb{R}^I$. The output of the network is given by the following equations.

$$\begin{aligned}f(x;\theta) &= b+\textstyle\sum_{j=1}^H v_j h_j(x) \\ h_j(x) &= \tanh\left(a_j+\textstyle\sum_{i=1}^I u_{ij}x_i\right)\end{aligned}$$

We denote by $\theta=(a,U,b,V)$ the vector of the network parameters corresponding respectively to the biases and weights of the hidden layer, the bias and weights of the output layer.

We want to compute the Bayesian estimation of network parameters based on a sample of independent samples $\mathcal{T}=(x^{(i)},y^{(i)})_{1\leq i\leq n}$.

We take the Gaussian distribution centered on the network output $f(x;\theta)$ as conditional likelihood of the network :

$$p(y|x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(f(x;\theta)-y\right)^2}{2\sigma^2}\right) \tag{2}$$

1.) We consider a prior distribution on the parameters $\pi(\theta)$. Let $x^{(n+1)}$ be a new network input. Write the formulas of the predictive distribution and a forecast $\hat{y}$ for the quadratic loss function. Suppose that by an MCMC (Markov Chain Monte-Carlo) algorithm, we generate a Markov chain $\theta_1,\ldots,\theta_N$ that has converged towards the stationary distribution $p(\theta|\mathcal{T})$ (to be exact, we also assume that it is Harris-ergodic). How would you estimate $\hat{y}$?

2.) We consider a Gaussian prior distribution, independent for each individual parameter. The variances are given by :

- σ_a^2 for bias terms a_j pour $1\leq j\leq J$;
- σ_u^2 for weights u_{ij}, $1\leq i\leq I$, $1\leq j\leq H$;
- σ_b^2 for the bias b;

- σ_v^2 for the weights v_j, $1 \leq j \leq H$.

a) Determine the expectation and variance of the contribution of each neuron in the hidden layer for the output $f(x^{(1)})$, that is to say $\mathbb{E}(v_j h_j(x^{(1)}))$ and $\mathbb{V}\left(v_j h_j(x^{(1)})\right)$, as functions of $\gamma(x^{(1)}) = \mathbb{E}\left(h_j(x^{(1)})^2\right)$.

Deduce the a priori limit distribution of $f(x^{(1)})$ when the number of neurons in the hidden layer tends towards infinity. How to choose σ_v to ensure that the variance of this limit law remains finite ?

b) Let $x^{(p)}$, $x^{(q)}$ be two inputs. Show in the same way that $\left(f\left(x^{(p)}\right), f\left(x^{(q)}\right)\right)$ tends towards a Gaussian vector whose covariance $Cov\left(f\left(x^{(p)}\right), f\left(x^{(q)}\right)\right)$ can be expressed as:

$$\sigma_b^2 + w_v^2 \kappa(x^{(p)}, x^{(q)}) \ ,$$

with $\kappa(x^{(p)}, x^{(p)}) = \gamma(x^{(p)})$.

Perform a random simulation for a network with 1 input and 1 output and different numbers of neurons in the hidden layer. Visualize the obtained points to confirm the Gaussian nature when H is large.

Remark:

The neural network tends towards a Gaussian process when the number of neurons in the hidden layer increases.

If $x^{(p)}$ is close to $x^{(q)}$), $\gamma(x^{(p)}) \approx \gamma(x^{(q)}) \equiv \tilde{\gamma}$, we can approximate $\kappa(x^{(p)}, x^{(q)})$:

$$\begin{aligned}\kappa\left(x^{(p)}, x^{(q)}\right) &= \frac{1}{2}\left(\gamma(x^{(p)}) + \gamma(x^{(q)}) - \mathbb{E}\left[\left(h(x^{(p)}) - h(x^{(q)})\right)^2\right]\right) \\ &\approx \tilde{\gamma} - \frac{1}{2}\delta\left(x^{(p)}, x^{(q)}\right)\end{aligned}$$

We could show that for the activation function tanh, $\delta\left(x^{(p)}, x^{(q)}\right)$ is of order $\|x^{(p)} - x^{(q)}\|^2$, which corresponds to a regular process.

If the activation function is a step function :

$$\forall x \in \mathbb{R} : h(x) = \begin{cases} -1 & \text{si} \quad x < 0 \\ 1 & \text{si} \quad x \geq 0 \end{cases}$$

then $\delta\left(x^{(p)}, x^{(q)}\right)$ is of order $\|x^{(p)} - x^{(q)}\|$, which corresponds to a Brownian.

Obviously, we can thus play on the regularity of the response of the neural network. The same would be true by using different types of prior distributions on the parameters.

Solution: 1)

So we are looking for

$$\begin{aligned}p(y|x^{(n+1)}, \mathcal{T}) &= \int_\Theta p(y, \theta|x^{(n+1)}, \mathcal{T})\, d\theta \\ &= \int_\Theta p(y|x^{(n+1)}, \theta, \mathcal{T}) p(\theta|x^{(n+1)}, \mathcal{T})\, d\theta \\ &= \int_\Theta p(y|x^{(n+1)}, \theta) p(\theta|\mathcal{T})\, d\theta\end{aligned}$$

According to the Bayes' theorem, we have :

$$p(\theta|\mathcal{T}) \propto \pi(\theta) \prod_{i=1}^{n} p(y^{(i)}|x^{(i)}, \theta).$$

So :

$$p(y|x^{(n+1)}, \mathcal{T}) \propto \int_{\Theta} p(y|x^{(n+1)}, \theta)\pi(\theta) \prod_{i=1}^{n} p(y^{(i)}|x^{(i)}, \theta)\, d\theta$$

If we choose the quadratic loss function, the best estimator is given by the expectation of the predictive distribution. So :

$$\hat{y} = \int_{\mathcal{Y}} yp(y|x^{(n+1)}, \mathcal{T})\, dy = \int_{\Theta} \left(\int_{\mathcal{Y}} yp(y|x^{(n+1)}, \theta)\, dy \right) p(\theta|\mathcal{T})\, d\theta \ .$$

From equation (2):

$$\left(\int_{\mathcal{Y}} yp(y|x^{(n+1)}, \theta)\, dy \right) = f(x^{(n+1)}; \theta)$$

Then finally :

$$\hat{y} = \int_{\Theta} f(x^{(n+1)}; \theta) p(\theta|\mathcal{T})\, d\theta \ .$$

Suppose that we have generated a Markov chain $\theta_1, \ldots, \theta_N$ which converged to the stationary distribution $p(\theta|\mathcal{T})$. We recall that, for example, the Metropolis-Hastings algorithm is interesting because it does not require calculating the normalizing constant in the density.

We can apply the Ergodic theorem, which tells us that $\forall\ g$ integrable with respect to $p(\theta|\mathcal{T})$, we have :

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} g\left(\theta_i\right) = \int_{\Theta} g(\theta) p(\theta|\mathcal{T}) d\theta \quad \text{almost surely.}$$

So here :

$$\frac{1}{n} \sum_{i=1}^{n} f(x^{(n+1)}; \theta_i) \underset{n\to+\infty}{\longrightarrow} \hat{y} \ .$$

In concrete terms, even if N is large, it is finite, and in order to avoid disruptions related to the beginning of the chain when it has not yet converged, we consider a period called "burn-in" that describes the practice of throwing away some iterations (often the first 20%) at the beginning of an MCMC run.

Let for example N_0 be the number of elements that we eliminate, we then have :

$$\hat{y} \approx \frac{1}{N - N_0} \sum_{i=N_0+1}^{N} f(x^{(n+1)}; \theta_i) \ .$$

2. Since the v_js are independent of the a_js and the u_{ij}s, they are also independent of $h_j\left(x^{(1)}\right)$s. Therefore

$$\mathbb{E}\left(v_j h_j\left(x^{(1)}\right)\right) = \mathbb{E}\left(v_j\right)\mathbb{E}\left(h_j\left(x^{(1)}\right)\right) = 0$$

and

$$\mathbb{V}\left(v_j h_j\left(x^{(1)}\right)\right) = \mathbb{E}\left(v_j^2 h_j\left(x^{(1)}\right)^2\right) = \mathbb{E}\left(v_j^2\right)\mathbb{E}\left(h_j\left(x^{(1)}\right)^2\right) = \sigma_v^2\gamma\left(x^{(1)}\right).$$

The previous result is true for all j, $1 \leq j \leq H$. The central limit theorem can then be applied :

$$\frac{1}{\sqrt{H\sigma_v^2\gamma\left(x^{(1)}\right)}}\sum_{j=1}^{H} v_j h_j\left(x^{(1)}\right) \xrightarrow{\mathcal{L}} \mathcal{N}(0,1).$$

If we consider $\sigma_v^2 = \omega_v^2/H$, then : $\sum_{j=1}^{H} v_j h_j\left(x^{(1)}\right)$ has finite variance, and therefore:

$$\sum_{j=1}^{H} v_j h_j\left(x^{(1)}\right) \xrightarrow{\mathcal{L}} \mathcal{N}\left(0, \omega_v^2\gamma\left(x^{(1)}\right)\right).$$

b has also a centered Gaussian distribution independent of $v_j h_j\left(x^{(1)}\right)$, so we have :

$$b + \sum_{j=1}^{H} v_j h_j\left(x^{(1)}\right) = f\left(x^{(1)}\right) \xrightarrow{\mathcal{L}} \mathcal{N}\left(0, \sigma_b^2 + \omega_v^2\gamma\left(x^{(1)}\right)\right).$$

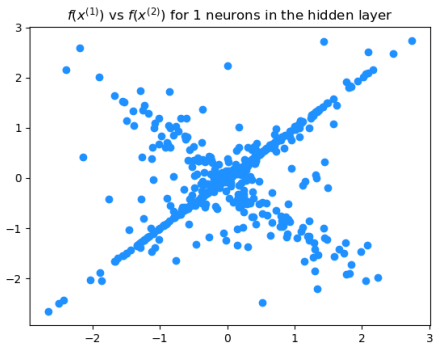
b) $\mathbb{E}\left(f\left(x^{(i)}\right)\right) = 0$, for all j. So:

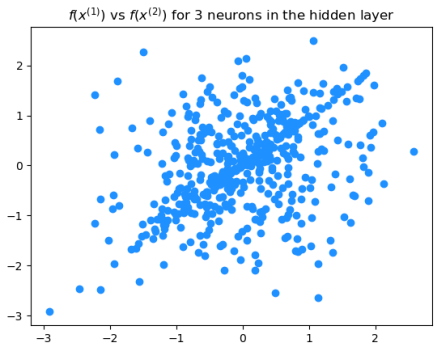
$$\mathrm{Cov}\left(f\left(x^{(p)}\right), f\left(x^{(q)}\right)\right) = \mathbb{E}\left(f\left(x^{(p)}\right) f\left(x^{(q)}\right)\right) = \sigma_b^2 + \sum_{j=1}^{H}\sigma_v^2\mathbb{E}\left(h_j\left(x^{(1)}\right) h_j\left(x^{(p)}\right)\right)$$

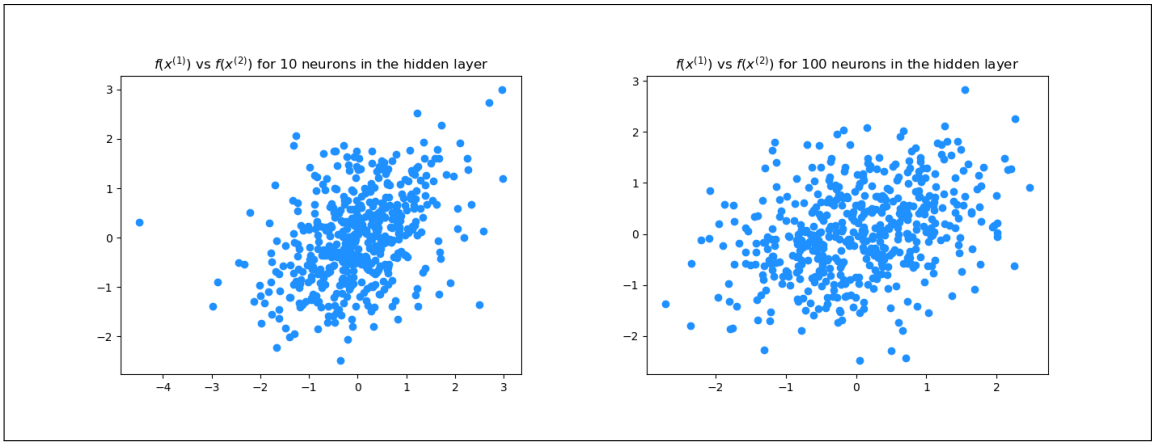
The term : $\mathbb{E}\left(h_j\left(x^{(1)}\right) h_j\left(x^{(p)}\right)\right)$ does not depend on j, we denote it then by $\kappa\left(x^{(p)}, x^{(q)}\right)$.

The multidimensional central limit theorem is then applied to conclude.

We present the results of 500 simulations for : $\sigma_a = 5$, $\sigma_v = 1/\sqrt{H}$, $\sigma_b = 0.1$, $\sigma_u = 10$ and we take $x^{(1)} = -0.2$ et $x^{(2)} = 0.4$. While the outputs do not correspond at all to a Gaussian process for $H = 1$ or $H = 3$, the Gaussian character appears clearly for $H = 100$.






$f(x^{(1)})$ vs $f(x^{(2)})$ for 10 neurons in the hidden layer
$f(x^{(1)})$ vs $f(x^{(2)})$ for 100 neurons in the hidden layer