Mémoire de première année

Elric Angot, François Duhesme

18 juin 2018

Table des matières

1 Introduction

Les réseaux de neurones sont des modèles paramétriques qui font des opérations de manière séquentielle sur leurs données. Chacune de ces opérations est appelée "couche", et consiste en deux étapes : une transformation linéaire des données en entrée, puis l'application d'une fonction d'activation non linéaire. Le schéma ci-dessous est éclairant, bien que simplifié (dans les faits, il y a plusieurs vecteurs en entrée, plusieurs en sortie, et les vecteurs de sortie sont des combinaisons linéaires de tous les vecteurs en entrée) :

$$\begin{pmatrix} a_1 \\ a_2 \\ ... \\ a_{n-1} \\ a_n \end{pmatrix} \underset{(1)}{\rightarrow} \begin{pmatrix} b_1 \\ b_2 \\ ... \\ b_{m-1} \\ b_m \end{pmatrix} \underset{(2)}{\rightarrow} \begin{pmatrix} f_1(b_1) \\ f_2(b_2) \\ ... \\ f_{m-1}(b_{m-1}) \\ f_m(b_m) \end{pmatrix}$$

avec, dans (1), la multiplication du vecteur en entrée par $M \in \mathcal{M}_{m,n}(\mathbb{R})$, et, dans (2), l'application des fonctions d'activation à chaque élément du vecteur. La plupart du temps, les fonctions d'activations sont des sigmoïdes. La fonction $x \longrightarrow \frac{1}{1+e^{|x|}}$ en est un bon exemple. Les réseaux de neurones sont présents partout aujourd'hui, et l'augmentation de leur puissance est spectaculaire. Grâce à eux, les machines sont plus à même de traiter des images, comprendre un enregistrement audio, ou plus prosaïquement gagner au Go contre le champion du monde. Les réseaux de neurones les plus puissants actuellement sont dits "profonds". Ils tirent leurs efficacités de plusieurs couches, et des fonctions d'activations adaptées à ce grand nombre de couches. Elles sont nommées fonctions d'activations rectifiées. Par exemple, la fonction $x \longrightarrow max(0, x)$ en est une. Enfin, le grand nombre de données à leur disposition permet un entraînement efficace de ces algorithmes.
Il existe trois facteurs théoriques qui expliquent le fonctionnement de ces réseaux : leur architecture, les techniques de régularisation et les techniques d'optimisation algorithmique. Les comprendre est nécessaire afin de savoir pourquoi ces réseaux de neurones sont si puissants.

1.1 Architecture

Une propriété importante du design de l'architecture des réseaux de neurones est qu'il peut approximer l'évaluation de fonctions arbitraires sur les données. Mais en quoi cette capacité dépend de la profondeur (c'est-à-dire du nombre de couches) et de la largeur (c'est-à-dire le nombre maximum de vecteurs qu'on aura à une étape donnée) ?
Des premiers travaux ont montré qu'un réseau avec une seule couche peut approximer toutes fonctions. En revanche, un réseau avec peu de couches et ayant une grande largeur a des performances significativement plus élevées que celui disposant d'une unique couche. Cela est dû au fait que plus la largeur est élevée et plus le réseau est profond, plus il est capable de capturer les invariants. Par exemple, dans la reconnaissance d'images, un invariant serait le point de vue ou la luminosité (ces deux paramètres ne changeant pas la nature de l'objet, il ne faut donc pas les prendre en compte lorsqu'on désire nommer

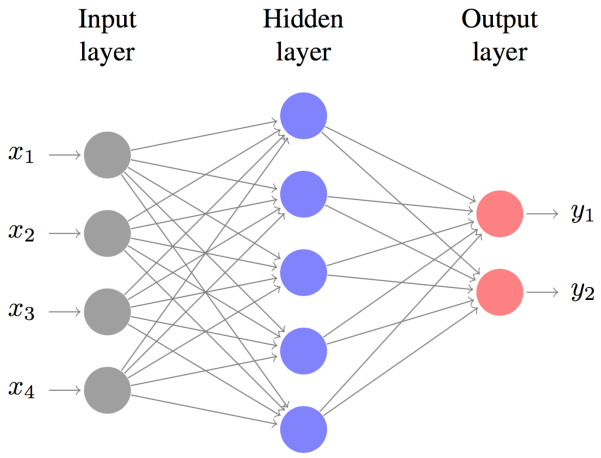


FIGURE 1 – Schéma d'un réseau typique

un objet d'après une image). Les preuves mathématiques résistent encore aux chercheurs, mais les faits sont là.

1.2 Propriétés de régularisation

Une autre propriété substantielle de l'architecture des réseaux de neurones est leur capacité à généraliser à partir d'un échantillon de taille réduite. A l'aide de méthodes statistiques, on peut montrer que la taille de l'échantillon évolue polynomialement avec la taille du réseau. En pratique, les réseaux de neurones profonds ont à leur disposition des échantillons trop petits par rapport aux normes données par les méthodes statistiques, mais, avec des techniques sophistiquées, les informaticiens arrivent à s'en sortir. Une explication heuristique de ce phénomène est que les réseaux de neurones profonds ont la capacité à augmenter la distance entre les objets de classes différentes, mais conservent (peu ou prou) la distance entre les objets de même classe.

1.3 Propriétés théoriques

Une autre propriété considérable de ces réseaux est leur capacité à donner une bonne représentation des données. Grossièrement, une représentation est une fonction qui est utile pour une tâche. Par exemple, filtrer les bruits parasites lors d'un enregistrement demande une fonction qui lisse les données : dans ce domaine, la fonction de lissage est une représentation.

1.4 Propriétés d'optimisation

Pour entraîner le réseau de neurone, on utilise souvent la technique de rétropropagation du gradient. Elle consiste à calculer le gradient de l'erreur entre la première couche et la dernière couche afin de la réduire au fur et à mesure qu'on entraîne l'algorithme.

1.5 Objectif de ce travail

Notre objectif sera d'étudier, à quel point un réseau de neurones aux poids aléatoires préserve la distance. Pour cela nous allons étudier le cas d'une seule couche dans un réseau.

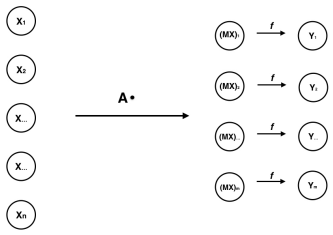


FIGURE 2 – La fonction que nous allons étudier

En particulier, nous avons la situation suivante. On a $\mathbb{R}^n \xrightarrow{A\cdot} \mathbb{R}^m \xrightarrow{f^m} \mathbb{R}^m$, où A est une matrice aléatoire gaussienne $m \times n$, et $f : \mathbb{R} \longrightarrow \mathbb{R}$ une fonction de troncature qui est linéaire sur un intervalle (possiblement semi-infini) et constant en dehors . Nous supposons en plus que f satisfait la propriété suivante :

$$f = \begin{cases} 0 < f(x) \leqslant x & \text{si } x > 0 \\ f(x) = 0 & \text{si } x = 0 \\ 0 > f(x) \geqslant x & \text{si } x < 0 \end{cases}$$

Le problème consiste alors à étudier la fonction $h = f^m(A\cdot)$. Nous supposerons de plus que notre ensemble de données appartient à une sous-variété compacte K de la sphère $\mathbb{S}^{n-1}$. La distance sur K sera alors celle induite par la métrique riemannienne de $\mathbb{S}^{n-1}$. Introduisons une première grandeur qui interviendra dans notre démonstration :

Définition 1.1 *La largeur moyenne gaussienne d'une sous-variété compacte K de $\mathbb{S}^{n-1}$ est définie par :*

$$\omega(K) := \mathbb{E}[\sup_{x,y \in K} \langle g, x - y \rangle] \tag{1}$$

où l'espérance est prise sur un vecteur aléatoire gaussien g avec des composantes i.i.d normales.

Il est a noter que $\sup_{x,y\in K}\langle g, x - y\rangle$ mesure la largeur de l'ensemble K dans la direction de g comme illustré dans la figure 3.

Ensuite, on moyenne sur g tiré de manière gaussienne, d'où la définition de largeur moyenne gaussienne.

Le résultat principal que nous allons démontrer consiste à montrer que nous pouvons à peu de choses près retrouver les distances initiales à partir des résultats.

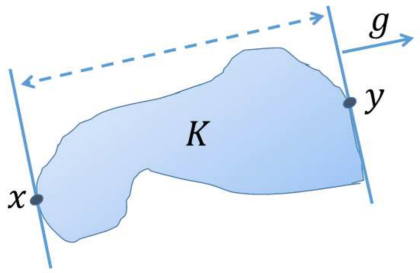


FIGURE 3 – largeur de l'ensemble K en direction de g.

Définition 1.2 *Une application entre deux espaces métriques*

$$g : (X, d_X) \longrightarrow (Y, d_Y)$$

est une δ-isométrie si

$$\forall x, y \in X : |d_X(x_1, x_2) - d_Y(g(x_1), g(x_2))| \leqslant \delta$$

Nous allons démontrer que la composition de fonctions suivante :

$$\mathbb{R}^n \xrightarrow{A\cdot} \mathbb{R}^m \xrightarrow{f^m} \mathbb{R}^m \xrightarrow{signe} \{-1, 1\}^m$$

est une δ -isométrie. Ici le cube de Hamming $\mathbb{H} := \{-1, 1\}$ est muni de la distance de Hamming $\frac{1}{m} d_H(u, v) := \sum_{i=1}^m \mathbf{1}_{u_i \neq v_i}$ qui correspond à la proportion des coordonnées qui sont distinctes.

Notons que comme $sgn(f(x)) = sgn(x)$ nous pouvons supposer dans la suite que f est l'identité.

Remarque 1 *Dans toute la suite de ce travail nous noterons par C des constantes quelconques (pas forcément les mêmes !).*

Théorème 1.3 *Soit $K \subset \mathbb{S}^{n-1}$, $\delta > 0$. Il existe une constante C tel que si*

$$m \geqslant C\delta^{-6}\omega(K)^2$$

Alors si A est une matrice gaussienne aléatoire, avec des entrées indépendantes $\mathbf{N}(0, 1)$. Alors avec une probabilité valant au moins $1 - 2\exp(-c\delta^2 m)$, la fonction

$$\begin{aligned} g : K &\longrightarrow \mathbb{H} = \{-1, 1\}^m \\ x &\longmapsto sgn^m(Ax) \end{aligned}$$

est une δ-isométrie.

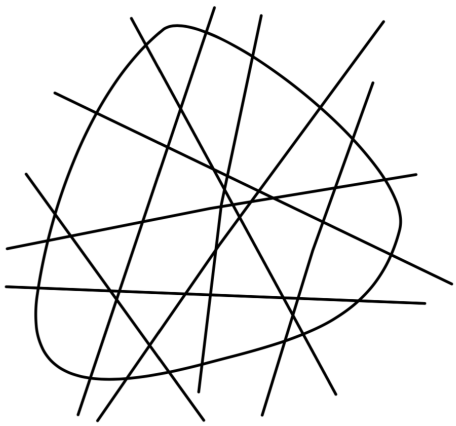


FIGURE 4 – Hyperplans induisant une distance

2 Démonstration du théorème principal

Pour montrer que g est une δ -isométrie, nous allons comparer la distance standard d sur K à la distance d_A qui est le tiré en arrière de la distance de Hamming d_H par g. En prenant les orthogonaux aux lignes de la matrice, A définit m hyperplans. La distance d_A entre deux points x et y consiste alors à compter la proportion d'hyperplans qui séparent x de y.

Nous allons procéder en 3 étapes pour la démonstration :

1. montrer que $d_A(x, y) \approx d(x, y)$, avec grande probabilité pour x et y fixés.
2. étendre l'approximation $d_A(x, y) \approx d(x, y)$ de manière uniforme sur un sous ensemble ε-dense N_ε.
3. Etendre l'estimation à tout K par approximation.

L'étape la plus difficile sera la troisième, la difficulté majeure étant la discontinuité de la distance de Hamming $d_A(x, y)$ en x, y.

De fait, pour certains recouvrements d'hyperplans, c'est même impossible. On voit dans la figure en contrebas qu'il existe des pavages qui sont non uniformes mais qui le sont pour certains ensembles ε-denses.

Pour contourner cette difficulté, nous allons introduire une version 'faible' de la distance de Hamming.

L'idée va être d'introduire la notion d'hyperplans qui séparent presque deux points. Rappelons que la distance de Hamming normale $d_A(x, y)$ sur $\mathbb{R}^n$ par rapport à $A = (a_1, ..., a_m)$ s'écrit :

$$d_A(x, y) = \sum_{i=1}^{m} \frac{1}{m} \mathbf{1}_{\mathcal{E}_i}, \text{ avec } \mathcal{E}_i = \{(x, y) \mid signe\langle a_i, x\rangle \neq signe\langle a_i, y\rangle\}$$

De même on définit la distance de Hamming faible par :

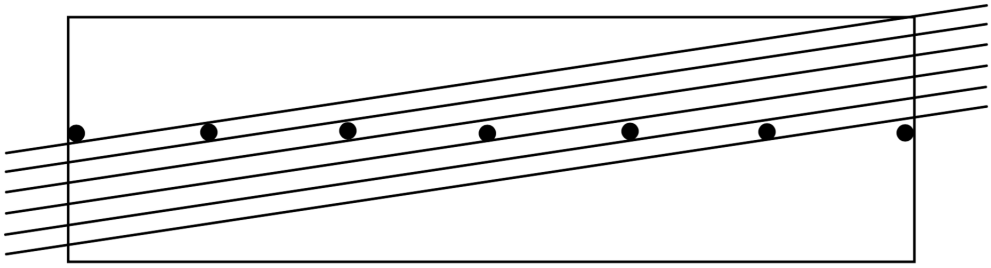


FIGURE 5 – Exemple d'un pavage non-uniforme pour l'ensemble $K = [-0.5, 0.5] \times [-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}]$, car chaque cellule a un diamètre > 1, alors que la distance induite sur le $\varepsilon-$filet $N_\varepsilon = \varepsilon\mathbb{Z} \cap K$ correspond à la distance euclidienne.

Définition 2.1 (distance de Hamming faible) *Soit A une matrice $m \times n$ avec les lignes $a_1, ..., a_m$ et $t \in \mathbb{R}$. La distance Hamming faible $d_A^t(x, y)$ sur $\mathbb{R}^n$ est définie par :*

$$d_A^t(x,y) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{\mathcal{F}_i},\ \textit{avec}\ \mathcal{F}_i = \{\langle a_i, x\rangle > t, \langle a_i, y\rangle < -t\} \cup \{-\langle a_i, x\rangle > t, -\langle a_i, y\rangle < -t\}$$

Pour un t positif, d_A^t compte les hyperplans séparant 'bien' x et y ; pour t négatif, elle compte les hyperplans qui les séparent 'presque'.

Remarque 2.2 *$d_A^t(x, y)$ est une fonction décroissante de t. On a donc les inégalités suivantes :*

$$d_A^{t^-}(x,y) \geqslant d_A^0(x,y) = d_A(x,y) \geqslant d_A^{t^+}(x,y), \quad \textit{avec}\ t^- \in \mathbb{R}^-\ \textit{et}\ t^+ \in \mathbb{R}^+$$

Pour un t fixé, la distance $d_A^t(x, y)$ est évidemment également discontinue en x, y. Cependant si nous nous donnons la liberté de varier t légèrement en fonction de x, y, la distance de Hamming faible est continue en x, y dans le sens suivant :

Lemme 2.3 (Continuité en x, y de la distance faible de Hamming) *Soient $x, y, x', y' \in \mathbb{R}^n$, tel que $\|Ax'\|_\infty \leqslant \varepsilon$ et $\|Ay'\|_\infty \leqslant \varepsilon$ pour un $\varepsilon > 0$. Alors pour tout $t \in \mathbb{R}$ on a :*

$$d_A^{t+\varepsilon}(x,y) \leqslant d_A^t(x + x', y + y') \leqslant d_A^{t-\varepsilon}(x,y).$$

démonstration. Reprenons les évènements

$$\mathcal{F}_i(x,y,t) = \{x, y, t \mid \langle a_i, x\rangle > t, \langle a_i, y\rangle < -t\} \cup \{-\langle a_i, x\rangle > t, -\langle a_i, y\rangle < -t\}$$

de la définition de la distance de Hamming faible 2.1. Par hypothèse, on a $|\langle a_i, x'\rangle| \leqslant \varepsilon, |\langle a_i, y'| \leqslant \varepsilon$ pour tout $i \in [\![1, m]\!]$. Ceci implique par l'inégalité triangulaire

$$\mathcal{F}_i(x,y,t+\varepsilon) \subseteq \mathcal{F}_i(x + x', y + y', t) \subseteq \mathcal{F}_i(x,y,t-\varepsilon)$$

En sommant sur les indicatrices, on obtient l'énoncé. □

Nous pouvons maintenant formuler une version plus forte du Théorème 1.3.

Théorème 2.4 (Pavages uniformes avec distance la faible) *Soit $K \subseteq S^{n-1}$ un compact, $\delta > 0$ suffisamment petit, et m assez grand pour que $m \geqslant C\delta^{-6}\omega(K)^2$. Alors, pour tout $t \in \mathbb{R}$, il existe une constante c tel que pour toute matrice aléatoire A aux lignes indépendantes $a_1, ..., a_m \sim \mathcal{N}(0, I_n)$ on a avec une probabilité d'au moins $1 - exp(-c\delta^2 m)$:*

$$|d_A^t(x, y) - d(x, y)| \leqslant \delta + 2|t|, x, y \in K$$

Si on prend $t = 0$, on retrouve le Théorème 1.3. Mais comme notre argument marche pour tout t, nous le formulons dans cette généralité.

Nous reprenons notre plan de démonstration en 3 étapes explicités au début de cette section.

2.1 Approximation pour deux points fixés.

Nous allons tout d'abord vérifier que pour x et y fixés, on a avec une grande probabilité que $d_A(x, y) \approx d(x, y)$. Vérifions en premier lieu que l'estimation est vraie en 'espérance', en d'autres termes que $\mathbb{E}d_A(x, y) \approx d(x, y)$.

Notons tout d'abord que

$$\mathbb{E}d_A(x, y) = \mathbb{E}d(x, y) \tag{2}$$

En effet, en dimension 2 on a :

$$\begin{aligned}\mathbb{E}d_A(x, y) &= \mathbb{E}\frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{\{sgn\langle a_i,x\rangle \neq sgn\langle a_i,y\rangle\}} \\ &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}\mathbf{1}_{\{sgn\langle a,x\rangle \neq sgn\langle a,y\rangle\}}, \text{ car les } a_i \text{ sont identiquement distribués}\end{aligned}$$

or

$$\begin{aligned}\mathbb{E}\mathbf{1}_{\{sgn\langle a,x\rangle \neq sgn\langle a,y\rangle\}} &= \frac{1}{2\pi}\int_{a\in S^1} \mathbf{1}_{\{sgn\langle a,x\rangle \neq sgn\langle a,y\rangle\}} da \\ &= \frac{1}{2\pi}\int_{a\in S^1} \mathbf{1}_{\{H(a) \text{ sépare } x \text{ et } y\}} da\end{aligned}$$

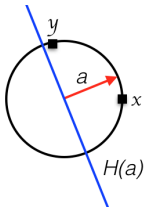


FIGURE 6 – H en fonction de a

or au lieu, d'intégrer autour du vecteur normal à H(a), on peut aussi intégrer par rapport à un vecteur tangent à H directement :

$$= \frac{1}{2\pi} \int_{b \in S^1} \mathbf{1}_{\{H(b) \text{ sépare } x \text{ et } y\}} db$$

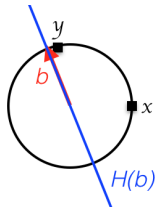


FIGURE 7 – H en fonction de b

$$= \frac{longeur_{\text{segment}}\,(x,y)}{\pi} = d(x,y)$$

Le cas à plus de dimensions se résout en utilisant que x et y se trouvent sur un plan en commun qui intersecte la sphère. Puis en intégrant on se ramène au cas de la dimension 2 via Fubini. Maintenant nous allons affiner notre comparaison entre $\mathbb{E}d_A^t(x,y)$ et $d(x,y)$:

Lemme 2.5 (Comparaison des distances de Hamming fortes et faibles en espérance) *Soit A une matrice aléatoire gaussienne comme dans les théorème 2.4. Alors, pour tout $t \in \mathbb{R}$ et tout $x, y \in \mathbb{S}^{n-1}$ on a :*

$$\left|\mathbb{E}d_A^t(x,y) - d(x,y)\right| \leqslant \mathbb{E}\left|d_A^t(x,y) - d_A(x,y)\right| \geqslant 2|t|$$

démonstration. La première inégalité est une application directe de l'inégalité de Jensen et de l'égalité 2 vue plus haut. Pour la deuxième inégalité, nous reprenons les évènements

$$\mathcal{E}_i = \{(x,y) \mid signe\langle a_i, x\rangle \neq signe\langle a_i, y\rangle\}$$
$$\mathcal{F}_i = \{\langle a_i, x\rangle > t, \langle a_i, y\rangle < -t\} \cup \{-\langle a_i, x\rangle > t, -\langle a_i, y\rangle < -t\}$$

utilisés précédemment dans la définition de la distance de Hamming. Il s'ensuit que

$$\begin{aligned}
\mathbb{E}\left|d_A^t(x,y) - d_A(x,y)\right| &= \mathbb{E}\left|\frac{1}{m}\sum_{i=1}^{m}(\mathbf{1}_{\mathcal{E}_i} - \mathbf{1}_{\mathcal{F}_i})\right| \\
&\leqslant \mathbb{E}|\mathbf{1}_{\mathcal{E}_1} - \mathbf{1}_{\mathcal{F}_1}| \qquad \text{par l'inégalité triangulaire et même loi} \\
&= \mathbb{P}\{\mathcal{E}_1 \triangle \mathcal{F}_1\} \quad \text{ici } \triangle \text{ désigne la différence symétrique ensembliste} \\
&\leqslant \mathbb{P}\{|\langle a_1, x\rangle| \leqslant |t|\} + \mathbb{P}\{|\langle a_1, y\rangle| \leqslant |t|\} \\
&= 2\mathbb{P}\{|g| \leqslant |t|\} \quad \text{où } g \sim \mathcal{N}(0,1), \text{ car toute combinaison linéaire} \\
&\qquad \text{des coordonnées d'un vecteur gaussien suit la loi normale.} \\
&\leqslant 2|t| \qquad \text{par estimation de la densité de la loi normale.}
\end{aligned}$$

□

Lemme 2.6 (comparaison des distances de Hamming fortes et faibles) *Soit A une matrice gaussienne. Alors :*

$$\forall t \in \mathbb{R}, \forall (x,y) \in (\mathbb{R}^n)^2, \qquad \mathbb{P}(|d_A^t(x,y) - d(x,y)| > 2 \cdot |t| + \delta) \geqslant 2 \cdot exp(-2m\delta^2)$$

Dans la démonstration nous utiliserons l'inégalité de Chernoff :
Soit X une variable aléatoire réelle qui vérifie :

$$\mathbb{E}(e^{tX}) < \infty \qquad \forall t \in \mathbb{R}$$

Alors, on a :

$$\forall a \geqslant 0, \qquad \mathbb{P}(X \geqslant a) \leqslant e^{-ta} \cdot \mathbb{E}(e^{tX})$$

démonstration. On a :

$$\begin{aligned} m \cdot d_A^t(x,y) &= \sum_{i=1}^{m} \mathbf{1}_{\{\langle a_i|x\rangle > t \wedge \langle a_i|y\rangle < -t\} \cup \{\langle a_i,x\rangle < -t \wedge \langle a_i,y\rangle > t\}} \\ &= B(m,p) \qquad \text{avec } p = \mathbb{E}(d_A^t(x,y)) \end{aligned} \tag{3}$$

D'après le lemme 1, on a $|p - d(x,y)| \leqslant 2|t|$. Alors, on a, $\forall \tau \in \mathbb{R}^+$:

$$\begin{aligned} &\mathbb{P}(|d_A^t(x,y) - p| > \delta) \\ &= \mathbb{P}(d_A^t(x,y) - p > \delta) + \mathbb{P}(|d_A^t(x,y) - p| < -\delta) \\ &\leqslant 2 \cdot e^{-\tau\delta} \cdot \mathbb{E}(e^{\tau(d_A^t(x,y) - p)}) \qquad \text{par l'inégalité de Chernoff} \\ &\leqslant 2 \cdot e^{-\tau\delta} \cdot e^{-\tau p} \cdot \mathbb{E}(e^{\tau d_A^t(x,y)}) \end{aligned}$$

$$\leqslant 2 \cdot e^{-\tau(\delta+p)} \cdot \prod_{i=1}^{m} \mathbb{E}(exp(\frac{\tau}{m} \mathbf{1}_{\{\langle a_i|x\rangle > t \wedge \langle a_i|y\rangle < -t\} \cup \{\langle a_i,x\rangle < -t \wedge \langle a_i,y\rangle > t\}})) \qquad \text{car variables indépendantes}$$

$$\leqslant 2 \cdot e^{-\tau(\delta+p)} \cdot \mathbb{E}(exp(\frac{\tau}{m} \mathbf{1}_{\{\langle a_1|x\rangle > t \wedge \langle a_1|y\rangle < -t\} \cup \{\langle a_1,x\rangle < -t \wedge \langle a_1,y\rangle > t\}}))^m \qquad \text{variables identiquement distribuées}$$

$$\begin{aligned} &\leqslant 2 \cdot e^{-\tau(\delta+p)} \cdot (pe^{\frac{\tau}{m}} + (1-p))^m \\ &\leqslant 2 \cdot e^{-\tau\delta} \cdot (pe^{\frac{\tau}{m}(1-p)} + (1-p)e^{-\frac{\tau}{m}(1-p)})^m \end{aligned}$$

Or, on a (après une analyse de monotonie) que :

$$\forall \vartheta \in [0,1], \forall \lambda \in \mathbb{R}, \qquad \vartheta \cdot e^{\lambda(1-\vartheta)} + (1-\vartheta)e^{-\lambda\vartheta} \leqslant e^{\frac{\lambda^2}{8}}.$$

Ici, on prend $\vartheta = p$ et $\lambda = \frac{\tau}{m}$
On a donc :

$$\begin{aligned} \mathbb{P}(|d_A^t(x,y) - p| > \delta) &\leqslant 2 \cdot e^{-\tau\delta} \cdot (e^{\frac{\tau^2}{8m^2}})^m \\ &\leqslant 2 \cdot e^{\frac{\tau^2}{8m} - \tau\delta} \\ &\leqslant 2 \cdot e^{-2m\delta^2} \qquad \text{en prenant } \tau = 4m\delta \end{aligned}$$

□

Définition 2.7 *Un ε-filet T est un ensemble de points de $\mathbb{R}^d$ tel que :*

$$\forall x \in \mathbb{R}^d, \exists y \in T, ||x - y|| < \varepsilon$$

2.2 Existence d'un filet uniforme

Lemme 2.8 (Estimation de la distance sur un filet) *Soit A une matrice gaussienne aléatoire définie comme dans le théorème 2.4.*
Soit N_ε un ε-filet de S^{n-1} tel que $log|N_\varepsilon| \leqslant C\varepsilon^{-2}\omega(K)^2$
Soit $\delta > 0$

Supposons que $m \geqslant 2 \cdot C \cdot \varepsilon^{-2} \cdot \delta^2 \cdot \omega(K)^2$
Soit $t \in \mathbb{R}$. Alors :

$$\mathbb{P}(\ \forall (x,y) \in N_\varepsilon^2 : |d_A^t(x,y) - d(x,y)| \leqslant 2|t| + \delta) \geqslant 1 - 2 \cdot e^{-\delta^2 m}$$

démonstration.

Montrons tout d'abord que $\mathbb{P}(\sup_{(x,y)\in N_\varepsilon^2} |d_A^t(x,y) - d(x,y)| \geqslant 2|t| + \delta) \leqslant |N_\varepsilon|^2 \cdot 2 \cdot e^{-2\delta^2 m}$.

Notons $\mathrm{P} = \mathbb{P}(\sup_{(x,y)\in N_\varepsilon^2} |d_A^t(x,y) - d(x,y)| \geqslant 2|t| + \delta)$.

On a :

$$\begin{aligned} P &= \mathbb{P}(\bigcup_{(x,y)\in N_\varepsilon^2} |d_A^t(x,y) - d(x,y)| \geqslant 2|t| + \delta) \\ &\leqslant \sum_{(x,y)\in N_\varepsilon^2} \mathbb{P}(|d_A^t(x,y) - d(x,y)| \geqslant 2|t| + \delta) \\ &\leqslant |N_\varepsilon|^2 \cdot 2 \cdot e^{-2\delta^2 m} \qquad \text{par le lemme 2.6} \end{aligned} \tag{4}$$

Montrons maintenant que $|N_\varepsilon|^2 \cdot 2 \cdot e^{-2\delta m} \leqslant 2e^{-\delta^2 m}$.

On a $|N_\varepsilon| \leqslant e^{C\omega(K)^2\varepsilon^{-2}} \leqslant e^{\frac{m\delta^2}{2}}$.

Ainsi, $2|N_\varepsilon|e^{-2\delta^2 m} \leqslant e^{-2\delta^2 m + 2\cdot\frac{\delta^2 m}{2}} \leqslant 2e^{-\delta^2 m}$

En recollant les deux inégalités obtenues, on conclut la preuve. □

2.3 Extension des estimations à l'espace tout entier

Lemme 2.9 (Contrôle des queues) *Soit $K \subseteq S^{n-1}$, $\varepsilon > 0$ et $m \leqslant \frac{C\omega(K)^2}{\varepsilon^2}$.*
Soit $a_1, ..., a_m \sim \mathcal{N}(0, I_n)$ m vecteurs indépentants.
On suppose que $C > 1346$ et que $c \geqslant \frac{\varepsilon^2}{128d(T)^2}$. Ce sont des hypothèses techniques. Alors,
$\forall x \in (K - K) \bigcap B_2^n \cdot \varepsilon, \qquad \mathbb{P}(\frac{1}{m}\sum_{i=1}^m |\langle a_i, x\rangle| \leqslant \varepsilon) \geqslant 1 - 2e^{-cm}$

Pour prouver le lemme ci-dessus, nous admettons le résultat suivant, dont la démonstration est technique :

Lemme 2.10 *Soit $K \subset \mathbb{R}^n$ et soient $a_1, ..., a_m \sim \mathcal{N}(0, I_n)$ m vecteurs indépendants.*
On pose $Z = \sup_{x \in K} |\frac{1}{m} \sum_{i=1}^m | \langle a_i, x \rangle | - \sqrt{\frac{2}{\pi}} ||x||_2$.
Alors, on a :
. $\mathbb{E}(Z) \leqslant \frac{4\omega(K)}{\sqrt{m}}$
. $P(Z > \frac{4\omega(K)}{\sqrt{m}} + u) \leqslant 2e^{-\frac{mu^2}{2d(K)^2}}$, *où* $u > 0$ *et* $d(K) = \max_{x \in K} ||x||_2$

démonstration.

$$\begin{aligned}
&\mathbb{P}\left(\sup_{x \in T} \sum_{i=1}^m | \langle a_i, x \rangle | \leqslant \frac{4\omega(T)}{\sqrt{m}} + \frac{\varepsilon}{8} + \sqrt{\frac{2}{\pi}}\varepsilon \right) \\
&\geqslant \mathbb{P}\left(\sup_{x \in T} \sum_{i=1}^m | \langle a_i, x \rangle - \sqrt{\frac{2}{\pi}}\varepsilon | + \sqrt{\frac{2}{\pi}}\varepsilon \leqslant \frac{4\omega(T)}{\sqrt{m}} + \frac{\varepsilon}{8} + \sqrt{\frac{2}{\pi}}\varepsilon \right) \qquad \text{car } ||x|| \leqslant \varepsilon, \text{ puisque } x \in \varepsilon B_2^n \\
&\geqslant \mathbb{P}\left(\sup_{x \in T} \sum_{i=1}^m | \langle a_i, x \rangle - \sqrt{\frac{2}{\pi}} | \leqslant \frac{4\omega(T)}{\sqrt{m}} + \frac{\varepsilon}{8} \right) \\
&= 1 - \mathbb{P}\left(\sup_{x \in T} \sum_{i=1}^m | \langle a_i, x \rangle - \sqrt{\frac{2}{\pi}} | > \frac{4\omega(T)}{\sqrt{m}} + \frac{\varepsilon}{8} \right) \\
&\geqslant 1 - 2exp\left(-\frac{m(\frac{\varepsilon}{8})^2}{2d(T)^2} \right) \\
&\geqslant 1 - 2exp(-mc)
\end{aligned}$$

Or,

$$\frac{4\omega(T)}{\sqrt{m}} + \frac{\varepsilon}{8} + \varepsilon\sqrt{\frac{2}{\pi}} \leqslant 8\sqrt{\frac{\omega(K)^2}{m}} + \frac{\varepsilon}{8} + \varepsilon\sqrt{\frac{2}{\pi}} \leqslant \frac{8\varepsilon}{\sqrt{C}} + \frac{\varepsilon}{8} + \varepsilon\sqrt{\frac{2}{\pi}} \leqslant \varepsilon$$

la première inégalité venant du fait que $\omega(T) \leqslant \omega(K - K) \leqslant 2 \cdot \omega(K)$. On utilise aussi le fait que $C > 1346$
D'où :

$$\begin{aligned}
\mathbb{P}\left(\frac{1}{m} \sum_{i=1}^m | \langle x, a_i \rangle | \leqslant \varepsilon \right) &\geqslant \mathbb{P}\left(\frac{1}{m} \sum_{i=1}^m | \langle x, a_i \rangle | \leqslant \frac{\varepsilon}{8} + \varepsilon\sqrt{\frac{2}{\pi}} + \frac{4\omega(T)}{\sqrt{m}} \right) \\
&\geqslant 1 - 2exp(-cm) \qquad (5)
\end{aligned}$$

□

Lemme 2.11 (Continuité par rapport à des perturbations L_1) *Soient $(x, y, x', y') \in (\mathbb{R}^n)^4$. On suppose que $||Ax'|| \leqslant \varepsilon m$ et que $||Ay'|| \leqslant \varepsilon m$ pour un certain $\varepsilon > 0$.*
Alors, on a :

$$\forall t \in \mathbb{R}, \forall M \geqslant 1, \qquad d_A^{t+M\varepsilon}(x, y) - \frac{2}{M} \leqslant d_A^t(x + x', y + y') \leqslant d_A^{t-M\varepsilon}(x, y) + \frac{2}{M}$$

démonstration.

D'après nos hypothèses, on a $\sum_{i=1}^m |\langle a_i, x'\rangle| \leqslant \varepsilon m$ et $\sum_{i=1}^m |\langle a_i, y'\rangle| \leqslant \varepsilon m$.

Ainsi, l'ensemble $T = \{[\![0, m]\!] / |\langle a_i, x'\rangle| \leqslant M\varepsilon \wedge |\langle a_i, y'\rangle| \leqslant M\varepsilon\}$ vérifie $|T^C| \leqslant \frac{2m}{M}$. En effet :

$$\sum_{i=1}^m |\langle a_i, x'\rangle| \geqslant \sum_{i\in T^C} |M\varepsilon| \geqslant |T^C| M\varepsilon,$$

d'où une majoration de la taille de T^C.

On définit $\mathcal{F}_i(x, y, t) = \{\langle a_i, x\rangle > t \wedge \langle a_i, y\rangle < -t\} \bigcup \{-\langle a_i, x\rangle > t \wedge -\langle a_i, y\rangle < -t\}$

Ainsi, on a, par inégalité triangulaire :

$$\mathcal{F}_i(x, y, t + M\varepsilon) \subseteq \mathcal{F}_i(x + x', y + y', t) \subseteq \mathcal{F}_i(x, y, t - M\varepsilon) \qquad i \in T$$

Ainsi, on a :

$$\begin{aligned} d_A^{t+M\varepsilon} &= \frac{1}{m}\sum_{i=1}^m \mathbf{1}_{\mathcal{F}_i}(x, y, t + M\varepsilon) \\ &\leqslant \frac{|T^C|}{m} + \frac{1}{m}\sum_{i\in T} \mathbf{1}_{\mathcal{F}_i}(x, y, t + M\varepsilon) \\ &\leqslant \frac{2}{M} + \sum_{i\in T} \mathbf{1}_{\mathcal{F}_i}(x + x', y + y', t) \\ &\leqslant \frac{2}{M} + d_A^t(x + x', y + y') \qquad (6) \end{aligned}$$

Cela prouve la première inégalité. La deuxième se fait de manière similaire. □

2.4 Démonstration du Théorème 2.4

Nous pouvons maintenant procéder à la démonstration du théorème 2.4. Soient l'ensemble K, les nombres δ, m, t et la matrice aléatoire A comme dans le théorème. Nous posons $\varepsilon = \delta/100$ et $M = 10/\delta$. Soit N_ε un ε-filet sur K définie comme dans la section précédente (c'est à dire que $\log(|N_\varepsilon|) \leqslant C\varepsilon^{-2}\omega(K)^2$). Le lemme 2.8 nous permet de contrôler les distances des éléments de N_ε. Le lemme, 2.9 quant à lui, nous permet d'estimer les queues. Par les hypothèses sur m et notre choix de ε nous pouvons appliquer ces deux lemmes. En utilisant l'estimation triviale : $\mathbb{P}(A \cap B) \geqslant \mathbb{P}(A) + \mathbb{P}(B) - 1$ on obtient donc qu'avec probabilité au moins $1 - 2\exp(-\delta^2 m) + 1 - 2\exp(-cm) - 1 \geqslant 1 - 4\exp(-c\delta^2 m)$ les résultats suivants sont vrais : Pour $x_0, y_0 \in N_\varepsilon$ et $x', y' \in (K - K) \cap \varepsilon B_2^n$, on a :

$$\begin{aligned} \left|d_A^{t-M\varepsilon}(x_0, y_0) - d(x_0, y_0)\right| &\overset{\text{lemme 2.8}}{\leqslant} 2|t - M\varepsilon| + \delta/2 \\ \left|d_A^{t+M\varepsilon}(x_0, y_0) - d(x_0, y_0)\right| &\overset{\text{lemme 2.8}}{\leqslant} 2|t + M\varepsilon| + \delta/2 \end{aligned} \qquad (7)$$

(8) $$\|Ax'\|_1 \overset{\text{lemme 2.9}}{\leqslant} \varepsilon m, \qquad \|Ay'\|_1 \overset{\text{lemme 2.9}}{\leqslant} \varepsilon m.$$

Soient $x, y \in K$. On peut les écrire de la manière suivante :

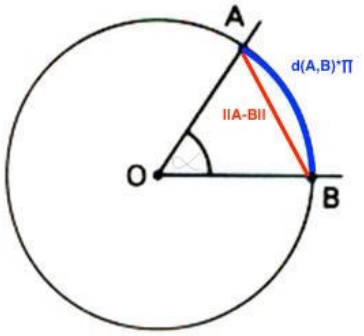
(9) $$x = x_0 + x', y = y_0 + y', \quad \text{où } x_0, y_0 \in N_\varepsilon, x', y' \in (K-K) \cap \varepsilon B_2^n.$$

La borne en 8 nous permet d'appliquer le Lemme 2.11. Nous parvenons à :

$$d_A^t(x,y) \leqslant d_A^{t-\varepsilon M}(x_0, y_0) + \frac{2}{M} \leqslant d(x_0, y_0) + 2|t| + 2M\varepsilon + \frac{\delta}{M}$$

grâce à l'inégalité triangulaire et l'inégalité 7. Par ailleurs, on peut remarquer que pour deux éléments $a, b \in K \subset S^n$, on a $d(a,b) \leqslant \|a-b\|_2$. Cela s'obtient facilement en se ramenant au cas $n = 2$, où on trouve

$$d(a,b)^2 = \frac{\alpha^2}{\pi^2} \leqslant \cos^2(\alpha) + (1 - \sin(\alpha))^2 = \|a-b\|_2^2.$$



En utilisant d'abord 9 puis l'inégalité sur les distances, nous obtenons :

$$|d(x_0, y_0) - d(x,y)| \leqslant d(x_0, x) + d(y_0, y) \leqslant \|x_0 + x\|_2 + \|y_0 - y\|_2 \leqslant 2\varepsilon.$$

En combinant les deux, nous avons

$$d_A^t(x,y) \leqslant d(x,y) + 2|t| + 2M\varepsilon + \frac{\delta}{2} + \frac{2}{M} + 2\varepsilon.$$

De manière analogue nous parvenons à

$$d_A^t(x,y) \geqslant d(x,y) - 2|t| - \delta$$

Nous concluons ainsi que

$$\left|d_A^t(x,y) - d(x,y)\right| \leqslant \delta + 2|t|.$$

□

3 Résultats empiriques

3.1 Code du programme

Ce programme fait le calcul du théorème 1.3 avec plusieurs variables (m et n) afin de voir que si m augmente, la distance de hamming se rapproche de la distance géodésique.

```
# 1 : avoir une matrice aléatoire.

from random import gauss
from math import sqrt
from math import acos
from numpy import random

#1.1 : f est une fonction suivant une loi centrée réduite.

def aleat_gauss(variance) :
"""
...
"""
return (gauss(0,variance))

#1.2 : créer la matrice

def creer_matrice_gaussienne ( m , n , variance ) :
"""
m est le nombre de lignes, et n le nombre de colonnes.
Normalement, la variance vaut (1/m)^1/2
"""
l = [[0 for k in range (n)] for l in range (m)]
for i in range (m) :
for j in range (n) :
l[i][j] = aleat_gauss(variance)
return (l)

#2 : avoir le produit de matrice

def prod(vect1,vect2) :
"""
calcule le produit de la transposée de vect 1 par vect 2
"""
res = 0
```

```
for i in range (len(vect1)) :
res += vect1[i]*vect2[i]
return res

def produit_matriciel(matrice,vecteur) :
"""
matrice est une matrice de taille m*n
vecteur est un vecteur de taille n
"""
nb_lgn = len(matrice)
nb_colonnes = len(matrice[0])
res = [0 for k in range (nb_lgn)]
for i in range (nb_lgn) :
res[i] = prod(vecteur,matrice[i])
return res

#3 : appliquer la fonction d'activation au résultat

def fonction_activation (x) :
if x<-1 :
return -1
if x>1 :
return 1
else :
return x




def appliquer_fct_activation(f,vect) :
"""
f est une fonction d'activation, et n est la taille de vect
"""
n = len(vect)
res = [0 for k in range (n)]
for k in range (n) :
res[k]= f(vect[k])
return res



#4 : assembler le tout
```

```
def reseau_neurones_simple (vect,matrice,f) :
"""
vect est le vecteur à traiter
matrice est la matrice aléatoire
t est une finction d'activation
"""
res = produit_matriciel(matrice,vect)
res = appliquer_fct_activation(f,res)
return res



#5 : distance euclidienne

def dist_eucl(vect1,vect2) :
res = 0
for k in range (len(vect1)) :
res += (vect1[k] - vect2[k])*(vect1[k] - vect2[k])
return sqrt(res)



# distance de Hamming

def hamming_distance(a,b) :
"""
a et b sont deux vecteurs
"""
res = 0
for k in range (len(a)):
if a[k] != b[k] :
res +=1
return res

#définition de g

def sgn(x) :
if x>0:
return (1)
else :
return (-1)

def g(x,M,f) :
```

```
"""
Ce g représente la fonction définie dans le théorème 1 du papier
Deep Neuron Network with Random Gaussian Weights.
Il correspond à appliquer la fonction sgn à chaque élément du vecteur suivant : f(Mx)
M est une matrice gaussienne aléatoire de taille m*n
x est un vecteur de la boule unité
f est une fonction de troncature
"""
a = produit_matriciel(M,x)
b =  appliquer_fct_activation(f,a)
c = appliquer_fct_activation(sgn,a)
return(c)

# distance géodésique sur la sphère unité

def dist_geodesique(x,y):
a = prod(x,y)
return acos(a)/3.1415




#Batterie de test

"il faut pouvoir prendre des vecteurs aléatoirement avec une loi uniforme"

def vect_aleat(n) :
x = random.rand(n)
#On fait en sorte que le vecteur puisse avoir des valeurs négatives
x = 2*(x-1/2)
#On le renormalise
x = x / sqrt(prod(x,x))
return(x)



def batterie_de_test(m,n,taille) :
"""
taille est le nombre de test. m est le nombre de ligne de la matrice,
et n le nombre de colonnes.
On retourne un tableau ayant une longueur valant taille.
```

```
Chaque élément du tableau est un réel.
Ce réel vaut le sup des valeurs absolues de la distance géodésique entre x et y
moins la distance de hamming entre g(x) et g(y).
Puisqu'on ne peut pas avoir le sup comme ça, on premdra le max sur 50
paires de vecteurs pris aléatoirement.
"""
variance = 1/sqrt(m)
f = fonction_activation
tabl = [0 for k in range (taille)]
for k in range (taille) :
M = creer_matrice_gaussienne ( m , n , variance )
maxi = 0
for z in range (50) :
x1 = vect_aleat(n)
y1 = vect_aleat(n)
c = abs( dist_geodesique(x1,y1) - 1/m* hamming_distance(g(x1,M,f),g(y1,M,f)))
maxi = max (maxi,c)
tabl[k] = maxi
return tabl


# Calcul. On va faire tourner le programme longtemps, pour avoir des beaux graphes.

def moyenne(t) :
n = len(t)
res = 0
for k in range (n):
res += t[k]
return (res/n)

def calcul() :
"""
la variable intéressante est le m. Il va donc prendre les valeurs:
5, 10, 50, 100, 500, 1000, 5000, 10 000, 50 000, 100 000.
n prendra les valeurs 2,5,8,11.
"""
t1 = [5,10,50,100,500,1000,5000,10000,50000,100000]
t2 = [2,5,8,11]
res = []
for k in range (len(t1)) :
for j in range (len(t2)) :
m = t1[k]
n = t2[j]
```

```
t = batterie_de_test(m,n,100)
moy = moyenne(t)
res += [(t,moy,m,n)]
return res
```

3.2 Résultats

On constate que δ évolue en $\frac{1}{\sqrt{m}}$ et non pas en $\frac{1}{m^{\frac{1}{6}}}$. Le résultat empirique est donc beaucoup plus puissant que celui du théorème. Cela vient du fait que nous avons fait beaucoup d'approximation sont venues dans la preuve des différents lemmes, on a fait beaucoup d'approximation, que ce soit dans l'approximation de l'exponentiel ou dans les différentes inégalités utilisées dans les probabilités.

m\n	2	5	8	11
5	0.4016585933968252	0.4916243354054183	0.4958625979571248 4	(0.509580680652715
10	0.2933531322000822	0.3611473057823082 5	0.3719537284696408	0.3750487612242083
50	0.1239045779107536	0.1663173761975906 8	0.1754721263420523 4	0.1734375818070781 3
100	0.0880458483795866 5	0.1152519190349680 4	0.1184106884519446 3	0.1227779073165577 6
500	0.0390332254042404 1	0.0530919761188479 65	0.0539531052178561 8	0.0547922197135791 2
1000	0.0283119696807758 45	0.0383392717965685 76	0.0385009779202724 4	(0.037306094970844125
5000	0.0132743188524826 82	0.0160249387015800 4	0.0176723800501717 54	0.0167887198108739 66
10000	0.0093061928672312 37	0.0118272259669314 97	0.0122051209717580 1	0.0125425943835385 6
50000	0.0039075122187386 04	0.0053408492554417 13	0.0053841654421900 47	0.0055066991695631 624
100000	0.0028698862306409 307	0.0037009896068153 493	0.0037807607480060 063	0.0038273033605631 91

Références

[1] R. Vidal, J. Bruna, R. Giryes and S. Soatto. 2017 . *Mathematics of Deep Learning. Technical Report* Arxiv

[2] Giryes, Raja and Sapiro, Guillermo and M. Bronstein, Alex, (2015), *Deep Neural Networks with Random Gaussian Weights : A Universal Classification Strategy ?* IEEE Transactions on Signal Processing, volume 64 ;

[3] Y. Plan, R. Versahynin ; (2014) *Dimension Reduction by random hyperplane tesselations* Discrete and computational Geometry, volume 51, pages 438-461