Rapport de stage

Rodrigue Lazarus

Stage de M1

Février-Juillet 2022

New York University

1 Vie et expérience à l'étranger

Au cours de mon stage, j'ai pu apprécier et découvrir un nouveau mode de vie - à savoir le mode de vie américain - ainsi qu'une multitude de lieux emblématiques, aussi bien à New York qu'en Amérique du Nord en général. Au-delà de la vie en totale indépendance et autonomie, propre à un séjour loin de chez soi, qui m'aura fait avancer davantage dans "l'âge adulte", j'ai pu constater certaines habitudes que l'on prend lorsque l'on vit dans une ville comme New York. J'ai de plus partagé mon logement avec des personnes de nationalités diverses, allant du Luxembourgeois à la Salvadorienne, et notamment des américains. J'ai alors pu voir concrètement les différences de mode de vie, aussi bien sur les rythmes alimentaires que les régimes ou encore la manière de faire la fête. Ce fut parfois enrichissant, parfois frustrant - je citerai en exemple une pizza "maison" constituée d'un curry étalé sur de la pâte - mais cela m'a aussi permis de comprendre à quel point les différences culturelles peuvent parfois rendre difficile des interactions. D'un point de vue politique et écologique, ce séjour m'aura donné une meilleure image de la France : lorsque l'on voit la mauvaise gestion des services publics - le métro en tête de gondole -, la criminalité et la violence qui font partie de notre quotidien et l'absence totale de conscience écologique, y compris chez les jeunes élites, on a de quoi se dire que la France se porte bien mieux sur certains points.

Pour ce qui est du cadre de travail, il était intéressant de travailler dans une université privée ayant beaucoup de moyens. Cela se ressentait sur l'apparence des locaux qui était dignes de ceux d'une start-up de finance quantitative. J'ai pu aussi constater un rapport au télétravail davantage assumé : toutes les sessions de groupes étaient en format hybride ou uniquement en ligne et pour le cours que je suivais, il n'y avait pas de tableau dans la salle : le professeur utilisait une tablette directement projetée par vidéo-projecteur et connectée à Zoom. J'ai pu d'ailleurs remarquer un rapport au covid presque inversé vis-à-vis de la France : alors que le pass vaccinal puis le port du masque étaient levés partout dans New York, l'université les conserva durant des mois encore. Bien que l'obligation de port du masque finit par être levé peu avant mon départ, le pass vaccinal version booster obligatoire reste encore aujourd'hui en vigueur - ce qui, pour des raisons de non-reconnaissance du document français, me força à faire mes dernières semaines de stage en télétravail.

En dehors du stage, j'ai pu faire du tourisme et visiter différents lieux à travers le continent Nord-Américain, entres autres :

- Différents lieux de New-York (Statue de la liberté, World Trade Center, le MET...) ;
- Montréal ;
- Washington (La Maison Blanche, le capitole, le Mall...) ;
- La Floride (Cap Cannaveral, Miami, Key West) ;
- La Californie (Los Angeles, Big Sur, San Francisco, Stanford) ;
- Boston et Harvard ;

En conclusion, les États-Unis sont un remarquable pays à visiter, offrant des paysages et expériences variés, mais y vivre dans mes conditions - hors des zones touristiques - offre une perspective différente et moins glorieuse où l'on ressent clairement les failles de cette société et ses différences vis-à-vis de notre modèle européen. Pour ce qui est de la recherche, j'ai apprécié la réflexion et le sentiment de travailler sur des projets concrets, mais j'ai été quelques peu déçu par la prise en charge trop légère et les longues périodes où j'étais livré à moi-même en attendant que mon tuteur trouve le temps de me parler.

2 Travaux Mathématiques

2.1 Introduction

L'objectif du projet de recherche est la compréhension d'une approche "dynamique" des réseaux de neurones à 3 couches. Plus précisément, il s'agissait d'étendre les résultats déjà connus pour les réseaux à 2 couches aux modèles avec une couche supplémentaire. Une première période du stage fut donc consacrée à l'étude des résultats connus sur les modèles à deux couches. Suite à cela, un premier travail fut l'extension d'un théorème de convergence démontré dans le cas à 2 couches au cas à 3 couches, qui consistait surtout à vérifier que l'on retrouvait bien les hypothèses nécessaires au théorème. Suite à cela, nous avons commencé la deuxième partie du projet, à savoir l'étude des modèles dits "synchronisés". L'objectif principal était de voir si l'on pouvait approcher un modèle "sparse" à 3 couches - difficile à appréhender - par un modèle à 3 couches régulier avec une fonction de loss différente. Pour cela, nous avons d'abord fait des essais numériques pour observer le comportement de ces différents modèles puis nous avons étudié des cas et arrangements particuliers afin de trouver un résultat à généraliser. Le projet est aujourd'hui arrêté à la démonstration rigoureuse du résultat trouvé, bien que nous ayons une intuition forte des méchanismes en jeu dans cette situation.

2.2 Modèles à 2 couches

2.2.1 Définitions

On considère l'espace paramétrique $\Theta = \mathbb{R} \times \mathbb{R}^d$ d'élements $\theta = (a, w)$. Un réseau de m neurones à 2 couches est alors une fonction

$$f(\theta_1, ..., \theta_m) : x \in \mathbb{R}^d \mapsto \frac{1}{m} \sum_{i=1}^{m} \varphi(\theta_i, x)$$

avec $(\theta_1, ..., \theta_m) \in \Theta$ et $\varphi(\theta_i, x) = a_i \sigma(\langle w_i, x \rangle)$; σ étant une fonction d'activation donnée.

À chaque famille de paramètres (θ_i) choisie, on associe une mesure empirique μ_m par la formule :

$$\mu_m = \frac{1}{m} \sum_{i=1}^{m} \delta_{\theta_i}$$

Cette association nous permet notamment de réecrire un réseau de neurone sous la forme :

$$f(x) = \int_{\Theta} \varphi(\theta, x) \mu_m(d\theta)$$

Cette définition peut alors être généralisée à des mesures non-discrètes, sous certaines conditions, que nous verrons juste après, on appelle cela l'approximation des champs moyens. Pour l'apprentissage du réseau de neurone, on définit une fonction de loss $l : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}_+$ avec $\mathcal{Y}$ l'espace des labels. On définit de plus le risque à minimiser :

$\mathcal{L} : (\theta_1, ..., \theta_m) \mapsto \mathbb{E}_\nu(l(f_{\theta_1,...,\theta_m}, f^*))$ où f^* désigne la fonction optimale à approcher et ν une distribution sur l'espace des labels.

Dans le cadre de notre projet nous allons définir quelques objets supplémentaires :

On définit la fonction de régularisation $V : \Theta \longrightarrow \mathbb{R}_+$ (par exemple $V(\theta) = \|\theta\|$ ou $V(\theta) = |a|\|w\|$ (La "path norm")). L'espace des fonctions que nous allons étudier est alors l'espace :

$$\mathcal{F} = \{f = \int_{\Theta} \varphi(\theta) \mu(d\theta) | \ \int_{\Theta} V(\theta) \mu(d\theta) < +\infty\}$$

C'est naturellement un espace de Banach pour la norme :

$$\|f\| = \int_{\Theta} V(\theta) \mu(d\theta)$$

On dira qu'un élément de $\mathcal{F}$ est engendré par la mesure (ou la famille de paramètres) associée à sa décomposition.

2.2.2 Propriétés de convergence

Supposons que la fonction de risque est α-strictement convexe et L-lisse. On définit l'algorithme de descente de gradient de la manière suivante :

$\theta_0 \in \Theta$ et $\theta_{t+1} = \theta_t - \tau \nabla_\theta \mathcal{L}(\theta_t)$ avec $\tau = \frac{1}{L}$.

Proposition : Soit θ^* le minimiseur de $\mathcal{L}$. Alors :

$$f(\theta_t) - f(\theta^*) \leq (1 - \frac{\alpha}{L})^t (\theta_1 - \theta_0)$$

De plus, $\alpha \leq L$ donc on a bien la convergence de la descente de gradient.

Preuve :

Commençons par rappeler les définitions de fonction α-strictement convexe et fonctions L-lisses :

On dit que $f : \mathcal{X} \longrightarrow \mathcal{Y}$ est α-strictement convexe si :

$$\forall (x, y) \in \mathcal{X}, f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{\alpha}{2}\|x - y\|^2$$

On prendra pour α la constante optimale.

On dit que f est L-lisse si ∇f est L-lipschizienne.

Notons d'abord que si f est α-strictement convexe, on a aussi l'inégalité :

$$f(y) \leq f(x) + \langle \nabla f(x), y \rangle + \frac{1}{2\alpha}\|\nabla f(x) - \nabla f(y)\|^2$$

On en déduit pour $x = x^*$: $\|\nabla f(x)\|^2 \geq 2\alpha(f(x) - f(x^*)$ De plus, si f est L-lisse, alors :

$$\forall (x, y) \in \mathcal{X}, f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{\text{Ł}}{2}\|x - y\|^2$$

Soit $t \in \mathbb{N}$, alors :

$$\begin{aligned} f(\theta_t) &\leq f(\theta_{t-1}) + \langle \nabla f(\theta_{t-1}), \theta_t - \theta_{t-1} \rangle + \frac{L}{2}\|\theta_t - \theta_{t-1}\|^2 \\ &\leq f(\theta_{t-1}) - \frac{1}{L}\|\nabla f(\theta_{t-1})\|^2 + \frac{1}{2L}\|\nabla f(\theta_{t-1})\|^2 \\ &\leq f(\theta_{t-1}) - \frac{1}{2L}\|\nabla f(\theta_{t-1})\|^2 \end{aligned}$$

Donc :

$$\begin{aligned} f(\theta_t) - f(\theta^*) &\leq f(\theta_{t-1}) - f(\theta^*) - \frac{1}{2L}\|\nabla f(\theta_{t-1})\|^2 \\ &\leq f(\theta_{t-1}) - f(\theta^*) - \frac{\alpha}{L}(f(\theta_{t-1}) - f(\theta^*)) \\ &\leq (1 - \frac{\alpha}{L})(f(\theta_{t-1}) - f(\theta^*)) \end{aligned}$$

On conclut par récurrence immédiate.

D'un point de vue continu, la descente de gradient correspond à la dynamique suivante :

$$\dot{\theta}_t = -\nabla \mathcal{L}(\theta_t)$$

Nous allons maintenant être plus spécifiques sur le choix de la loss. On suppose désormais que la fonction de risque est de la forme :

$$\mathcal{L}[\mu] = \|f - f^*\|_\nu^2 + \lambda \int_\Theta V(\theta)\mu(d\theta)$$

Avec ν une distribution sur l'espace des features, f la fonction engendrée par μ et f^* la fonction optimale. On

notera que, par la correspondance entre un choix de famille de paramètres et un choix de mesure, on peut voir le risque comme une fonction sur l'espace des mesures de probabilité.

On peut alors le réécrire de la manière suivante :

$$\mathcal{L}[\mu] = \|f^*\|_\nu^2 + \|f\|_\nu^2 - 2\langle f, f^*\rangle_\nu + \lambda \int_\Theta V(\theta)\mu(d\theta)$$

Soit :

$$\mathcal{L}[\mu] = C - 2\int_\Theta F(\theta)\mu(d\theta) + \int\int_{\Theta^2} K(\theta,\theta')\mu(d\theta)\mu(d\theta)$$

Avec :

$$F(\theta) = \langle \varphi(\theta), f^*\rangle_\nu - \frac{\lambda}{2}V(\theta)$$

$$K(\theta,\theta') = \langle \varphi(\theta), \varphi(\theta')\rangle_\nu$$

Ce terme est analogue au terme d'énergie d'un système à m particules en interaction. On peut de plus négliger la constante.

Pour comprendre la dynamique d'une mesure continue, commençons par observer ce qu'il se passe dans le cas discret :

$$\mathcal{L}(\theta_1, ..., \theta_m) = -\frac{2}{m}\sum_{i=1}^m F(\theta_i) + \frac{1}{m^2}\sum_{i,j} K(\theta_i, \theta_j)$$

Donc :

$$\begin{aligned}\dot{\theta_j} &= -\nabla\mathcal{L}(\theta_j) \\ &= \nabla F(\theta_j) - \frac{1}{m}\sum_{i=1}^m \nabla K(\theta_j, \theta_i) \\ &= -\nabla G(\theta_j(t), \mu_m(t))\end{aligned}$$

Avec :

$$G(\theta, \mu) = F(\theta) - \int_\Theta K(\theta,\theta')\mu(d\theta')$$

On en déduit la variation suivante :

$$\begin{aligned}\frac{\delta\mathcal{L}}{\delta\mu}(\theta) &= -2F(\theta) + 2\int K(\theta,\theta')\mu(d\theta') \\ &= G(\theta,\mu)\end{aligned}$$

G correspond, en physique, au potentiel instantané du système. Déterminons maintenant l'équation d'évolution de μ_t :

Soit χ une fonction d'essai. Alors :

$$\int_\Theta \chi(\theta)\mu_t(d\theta) = \frac{1}{m}\sum_{i=1}^m \chi(\theta_i(t))$$

Donc :

$$\begin{aligned}\int_\Theta \chi(\theta)\partial_t\mu_t(d\theta) &= \frac{1}{m}\sum_{i=1}^{m}\langle\nabla\chi(\theta_i(t)),\dot{\theta}_i(t)\rangle \\ &= -\int_\Theta \langle\nabla\chi(\theta),\nabla G(\theta,\mu_t)\rangle\mu_t(d\theta)\end{aligned}$$

On en déduit l'équation de transport :

$$\partial_t\mu_t(\theta) = div(\nabla G(\theta,\mu_t)\mu_t)$$

On obtient alors le théorème suivant :

Théorème (Françis Bach, Lénaïc Chizat) : Prenons les hypothèses suivantes :

— L'ensemble paramétrique s'écrit $\mathbb{R}\times\Theta$ avec $\Theta\subset\mathbb{R}^{d-1}$;

— La fonction φ vérifie $\varphi(a,w) = a\phi(w)$, également $V(a,w) = a\tilde{V}(w)$ avec ϕ et $\tilde{V}$ des fonctions bornées et différentiables avec différentielles Lipschitziennes ;

— **(i)** La fonction de risque $\mathcal{L}$ est convexe et différentiable avec différentielle Lipschitzienne sur les ensembles bornés et bornée sur les sous-ensembles de niveau ;

— **(ii)** Pour tout $f\in\mathcal{F}$, l'ensemble de points réguliers de $g_f :\mapsto \langle f,\phi(w)\rangle + \tilde{V}(w)$ est dense dans son image ;

— **(iii)** Deux choix :

  — (a) $\Theta = \mathbb{R}^{d-1}$ et pour tout $f\in\mathcal{F}$, la fonction $w\in\mathbb{S}^{d-2}\mapsto g_f(rw)$ converge uniformément dans $\mathbb{C}^1(\mathbb{S}^{d-2})$, lorsque $r\to+\infty$, vers une fonction vérifiant la régularité du point **(ii)** ou :

  — (b) Θ est la fermeture d'un ouvert convexe borné et pour tout $f\in\mathcal{F}$, g_f vérifie :
  $\forall w\in\partial\Theta, dg_f(w)(n_\theta) = 0$ avec n_θ vecteur normal à $\partial\Theta$ en w.

Sous ces hypothèses, on a le résultat suivant :

Soit (μ_t) une famille de mesures évoluant selon l'équation de transport et tels que pour un certain $r_0>0$, le support de μ_0 est contenu dans $[-r_0,r_0]\times\Theta$ et sépare $\{-r_0\}\times\Theta$ de $\{r_0\}\times\Theta$. Si (μ_t) converge vers μ_∞ pour la distance de Wasserstein, alors μ_∞ est un minimiseur global de $\mathcal{L}$ sur l'espace $\mathcal{M}_+(\mathbb{R}\times\Theta)$. En particulier, si $(\mu_m(t))$ est une famille de mesures discrètes initialisées dans $[-r_0,r_0]\times\Theta$ telles que $\mu_m(0)$ converge vers $\mu(0)$ pour la distance de Wasserstein, alors :

$$\lim_{t,m\to\infty}\mathcal{L}[\mu_m(t)] = \min_{\mu\in\mathcal{M}_+(\mathbb{R}\times\Theta)}\mathcal{L}$$

2.3 Modèles à 3 couches

2.3.1 Un résultat de convergence

Pour le modèle à 3 couches, nous ferons les hypothèses suivantes :

— la fonction de loss $l : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}_+$ est convexe par rapport au premier argument ;

— L'ensemble des labels $\mathcal{Y}$ est $\mathbb{R}$

— La fonction d'activation est :

$$\varphi : (\theta, x) \mapsto a\sigma(u^T\sigma(W^T x))$$

— L'espace des paramères est $\Theta = \mathbb{R} \times \mathbb{R}^s \times \mathbb{R}^{d\times s}$ avec $\theta = (a, u, W)$

— L'espace des fonctions étudiées est :

$$\mathcal{F} = \{f = \int_\Theta \varphi(\theta, x)\,\mu(d\theta) \mid \int_\Theta V(\theta)\,\mu(d\theta) < +\infty\}$$

avec $V : \Theta \longrightarrow \mathbb{R}_+$ une fonction de régularisation convexe ;

— Le risque à minimiser est :

$$\mathcal{L}[\mu] = \int_{\mathcal{X}\times\mathcal{Y}} l(\int_\Theta \varphi(\theta, x)\,\mu(d\theta), y)\,\nu(dx, dy) + \lambda \int_\Theta V(\theta)\,\mu(d\theta) \text{ pour } \nu \text{ une distribution sur } \mathcal{X} \times \mathcal{Y}$$

— La fonction σ est 1-homogène (exemple : RELU)

— $\forall\theta = (a, u, W) \in \Theta, V(\theta) = |a|V(1, u, W)$ (exemple : la "path norm")

2.4 Strategie

La statégie est d'adapter le théorème de convergence obtenu dans le cas d'un réseau à 2 couches. On considèrera de nouveau la mesure associée aux réseaux à nombre fini de neurones :

$$\mu = \frac{1}{m}\sum_{j=1}^{m} \delta_{\theta_j}$$

mais pour simplifier l'étude nous étendrons le résultat à l'espace des mesures de probabilités $\mathcal{M}_+(\Theta)$ La fonction de risque étant la même que dans le cas 2 couches, on aura la même dynamique pour l'évolution de (μ_t). Pour adapter le théorème, vérifions que nous retrouvons bien les hypothèses énoncées :

— Le domaine étant $\mathbb{R} \times \Theta$ avec $\Theta \subset \mathbb{R}^s \times \mathbb{R}^{s\times d} \approx \mathbb{R}^{\tilde{d}}$ (on suppose même $\Theta = \mathbb{S}^{\tilde{d}-1}$), on a bien $\varphi(a, \theta) = a\phi(\theta)$ avec ϕ une fonction bornée différentiable sur Θ avec une différentielle Lipschizienne ;

— On a aussi $V(a, \theta) = |a|\tilde{V}(\theta)$ avec $\tilde{V}$ une fonction bornée différentiable sur Θ avec une différentielle Lipschizienne ;

— Puisque $\mathcal{L}$ est la même fonction que précédemment, elle vérifie les mêmes hypothèses ;

— Si l'on définit $g_f : \theta \mapsto \langle f, \phi(\theta)\rangle + \tilde{V}(\theta)$ pour $f \in \mathcal{F}$, on voit que g_f est lisse pour tout $f \in \mathcal{F}$, donc par le théorème de Sard, l'ensemble des valeurs régulières de g_f est dense dans son image ;

— Puisque $\Theta = \mathbb{S}^{d-1}$, Θ est la fermeture d'un espace ouvert convexe bornée, $\partial\Theta = \varnothing$ donc on vérifie l'hypothèse (iii) ;

Donc, nous vérifions toutes les hypothèses du théorème énoncé précédemment. Le résultat peut donc être adapté au cas d'un réseau à 3 couches, la preuve étant similaire.

2.5 Modèles à 3 couches synchronisés

2.5.1 Définition

Notre dernier objectif est de comprendre le comportement de réseaux engendrés par des paramètres $\theta_i = (a_i, u_i, W_i) \in \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^d$ avec une contrainte additionelle de parcimonie sur les paramètres u_i : le support de chaque u_i sera désormais exactement de cardinal s. (voir 2.5.2.)

On appelle ce modèle un modèle synchronisé dur.

Malheureusement, ce type de modèle est difficile à appréhender alors nous allons considérer une version "douce" :

L'intuition est que chaque neurone de la seconde couche peut être dupliqué s fois, de manière à ce que chaque sous-neurone ne serait connecté qu'à un seul neurone de la première couche (voir 2.) On obtient alors un modèle à 3 couches avec une première couches ayant m neurones et une seconde ayant $m \times s$ neurones. De plus, on peut représenter la deuxième couche par un graphe, où chaque neurone devient un cluster de s sous-neurones, et deux sous-neurones venant d'un cluster différent sont reliés s'ils sont connectés au même neurone de la première couche. (voir 3.).

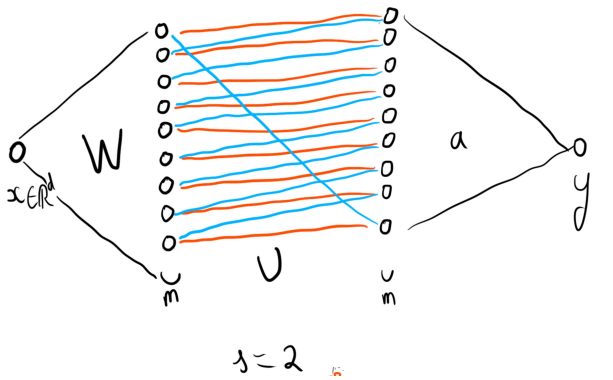

FIGURE 1 – Modèle dur avec chaque neurone connecté à exactement 2 autres neurones

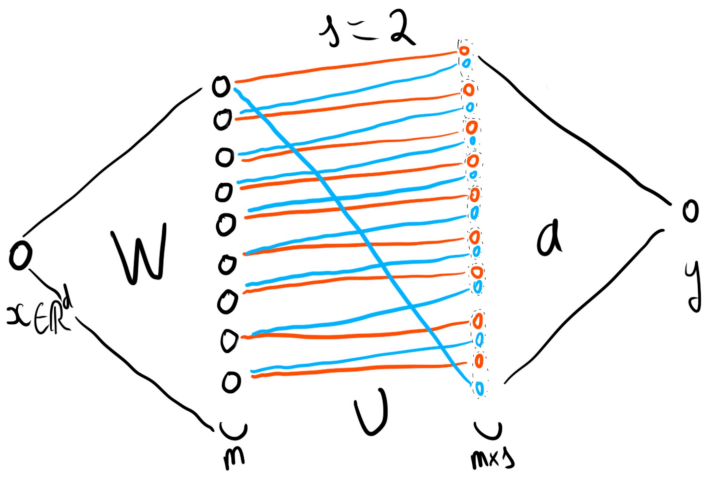

FIGURE 2 – On sépare les neurones de la 2e couche en sous-neurones

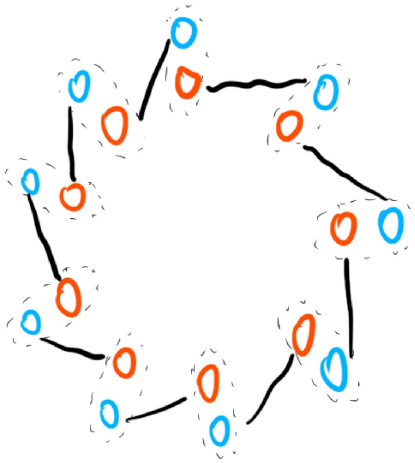

FIGURE 3 – Le graphe de synchronisation obtenu dans ce cas

2.5.2 Étude numérique

Une partie du projet a été consacrée à l'implémentation numérique des modèles présentés au-dessus, afin de comparer leurs performances. Les modèles ont été conçus avec le module Pytorch et les configurations suivantes :

— Les valeurs pertinentes du modèle sont : $n = 4000$ (nombre de données) ; $d = 100$ (dimension des données) ; $m_t = 10$ (nombre de neurone de la fonction génératrice) ; $s_t = 4$ (nombre de parcimonie de la fonction génératrice) ; $m = 5m_t$ et $s = 4s_t$ (mêmes nombres mais pour le modèle entraîné) ;

— Les données X sont générées par loi gaussienne ;

— Les labels Y sont générés en tirant au hasard une famille de paramètres (θ_i^*) (loi gaussienne) et en faisant $Y = f^*(X)$ où f désigne naturellement la fonction engendrée par les θ_i^* ;

— En fonction de la tâche à effectuer, la fonction "teacher" f^* peut être à 2 ou 3 couches ;

— Dans le cas de modèles à 3 couches, un "masque de parcimonie" est généré aléatoirement : il s'agit d'une matrice dont chaque entrée suit une loi de Bernoulli de paramètre $\frac{s}{m}$ (ou $\frac{s_t}{m_t}$ pour le teacher) et qui est multipliée terme à terme avec la matrice de seconde couche. Notons que ce masque est généré avant l'apprentissage et ne change pas par la suite ;

— $\lambda = 0,1$, $\tau = \frac{1}{10^5}$ et on effectue $n_{iter} = 1000$ itérations de l'algorithme de descente de gradient ;

— La mesure de performance se fait selon ce critère :
On génère un nouveau jeu aléatoire de données X_{test} et on regarde la valeur $\frac{\|f(X_{test}) - f^*(X_{test})\|^2}{\|f^*(X_{test})\|^2}$. Cette valeur est comprise entre 0 et 1 et devrait décroître à mesure que l'apprentissage a lieu.

Voici le code utilisé pour le modèle synchronisé dur :

Entrée [3]:
```
d = 100
n = 4000
m_t = 10
m = 5*m_t
s_t = 4
s = 16

mask_star = np.random.binomial(1,s_t/m_t,size=(m_t))
mask = torch.from_numpy(np.random.binomial(1,s/m,size = (m,m)))

X = np.random.randn(n,d)
a = np.random.randn(m_t)
u_tilde = np.random.randn(m_t,m_t)
W = np.random.randn(m_t,d)
u = np.array([u_tilde[k]*mask_star for k in range(m_t)])
Labelisation_3 = lambda x : np.array([(1/m_t)*np.sum([a[j]*relu(np.dot(u[j],np.array([relu(np.dot(W[k],x[i])) for k
X_t = torch.from_numpy(X)
Y_3 = torch.from_numpy(Labelisation_3(X))
```

Entrée [4]:
```
class NeuralNetwork_3(nn.Module):
    def __init__(self,d_in,s_neur,m_neur):
        super(NeuralNetwork_3,self).__init__()
        self.W = nn.Linear(d_in,m_neur)
        self.u = nn.Linear(m_neur,m_neur)
        self.a = nn.Linear(m_neur,1)

        def initialize():
            nn.init.normal(self.a)
            nn.init.kaiming_normal(self.W)
            nn.init.kaiming_normal(self.u)
            initialize()

    def forward(self,x):
        x = F.relu(torch.matmul(self.W.weight,x.t()))
        x = F.relu(torch.matmul(self.u.weight*mask,x))
        x = torch.matmul(self.a.weight,x)
        return x
model_3 = NeuralNetwork_3(d,s,m)
```

Entrée [5]:
```
eta = 0.00001
tau = 0.00001
n_step = 200
batch_size = 64
lamba = 0.1
iterations = [i for i in range(n_step)]
```

Entrée [6]:
```
optimizer_3 = torch.optim.Adam(model_3.parameters(),lr=eta)
optimizer = torch.optim.Adam(model_3.parameters(),lr=tau)
criterion = nn.MSELoss()
GE = lambda x,y : torch.norm(x-y).pow(2)/torch.norm(y).pow(2)
```

Entrée [7]:
```
X = np.random.randn(n,d)
a = np.random.randn(m)
u = np.random.randn(m,s)
W = np.random.randn(m,s,d)
Labelisation_3 = lambda x : np.array([(1/m)*np.sum([a[j]*relu(np.dot(u[j],np.array([relu(np.dot(W[j][k],x[i])) for k
X_t = torch.from_numpy(X)
Y_3 = torch.from_numpy(Labelisation_3(X))

model_3 = NeuralNetwork_3(d,s,m)
eta = 0.001
n_step = 1000
optimizer_3 = torch.optim.Adam(model_3.parameters(),lr=eta)
iterations = [i for i in range(n_step)]
X_test = torch.randn(n,d)
Y_test = torch.from_numpy(Labelisation_3(np.array(X_test)))
Error = np.zeros(n_step)
for epoch in range(n_step) :
    Error[epoch] = GE(model_3(X_test),Y_test)
    optimizer_3.zero_grad()
    pred = torch.flatten(model_3(X_t.float()))
    reg = 1/m*torch.sum(abs(model_3.a.weight)*(torch.norm(model_3.W.weight).pow(2) + torch.norm(model_3.u.weight).po
    loss = criterion(pred,Y_3.float()) + lamba*reg
    loss.backward()
    optimizer_3.step()
```

Voici les différents résultats de performance obtenus :

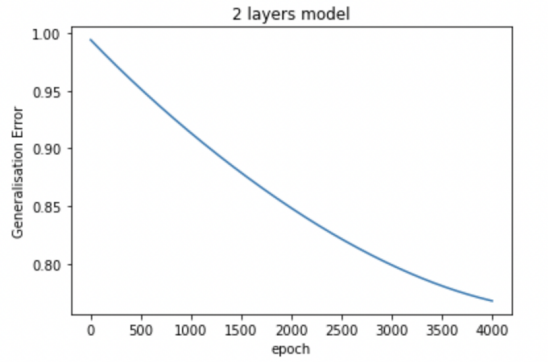

FIGURE 4 – Modèle à 2 couches

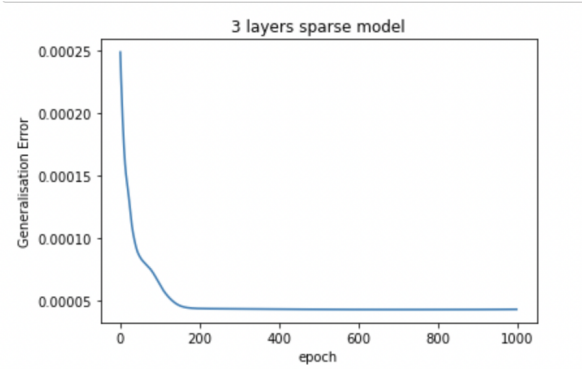

FIGURE 5 – Modèle à 3 couches non synchronisé

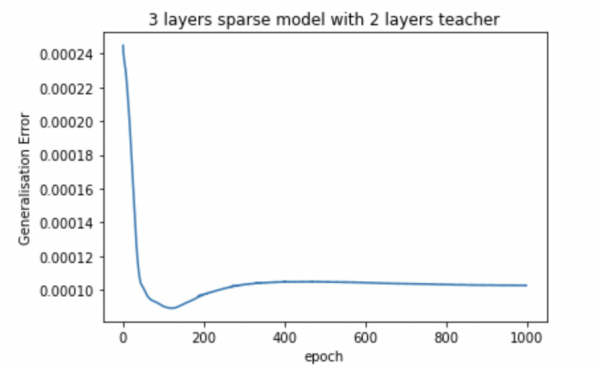

FIGURE 6 – Modèle à 3 couches non-synchronisé avec modèle teacher à 2 couches

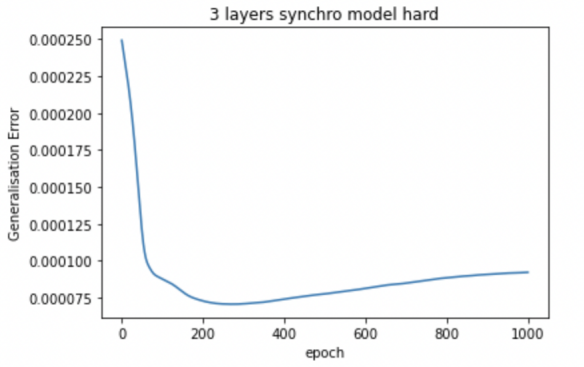

FIGURE 7 – Modèle à 3 couches synchronisé dur

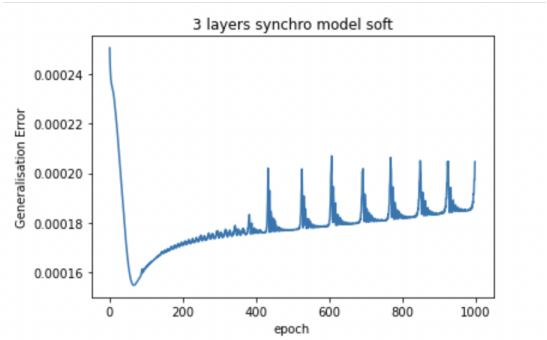

FIGURE 8 – Modèle à 3 couches synchronisé doux

On voit donc une efficacité claire du modèle à 3 couches synchonisé dur, tandis que la version douce est proche du modèle non-synchronisé en terme de performances.

Enfin, pour estimer l'effet du terme de synchronisation sur la dynamique d'apprentissage, nous avons mesuré l'évolution de la loss "usuelle" et de ce terme :

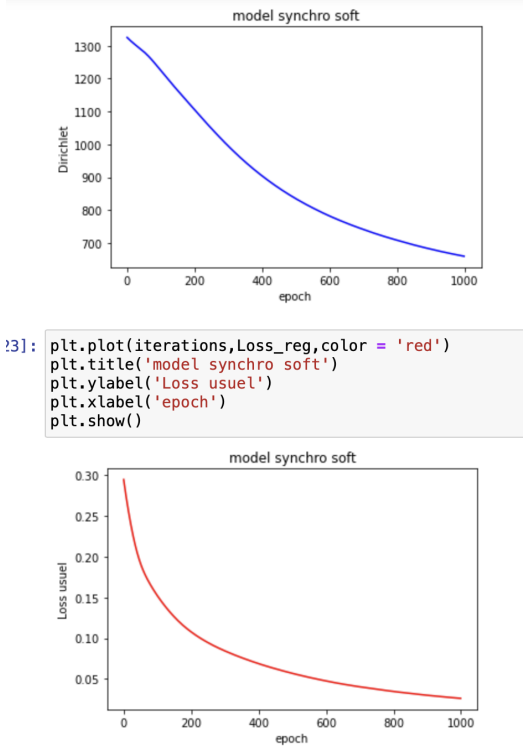

FIGURE 9 – La variation du terme de synchronisation est d'environ -46% alors que celle du terme de loss usuelle est de -83%

2.5.3 Cas s = 2

Pour mieux comprendre le modèle, nous allons d'abord poursuivre l'étude du cas représenté sur les graphiques.

On considère un réseau à 3 couches de paramètres $\theta_i \in \mathbb{R} \times \mathbb{R}^{m\times s} \times \mathbb{R}^d$. En général, on peut choisir les connections imposées entre les couches, afin que le graphe de synchronisation présente des symétries. Dans notre cas, on voit clairement que l'ordre des neurones ne change pas l'aspect *visuel* du graphe.

L'intervention de la synchronisation dans le modèle se fait par l'ajout d'un terme à la fonction de loss :

on veut que les neurones synchronisés soient aussi près les uns des autres que possible, mais on veut aussi avoir une organisation (au sens d'une permutation) optimale, ce terme vaut donc :

$$\mathcal{D}(u_1, ..., u_m) = \frac{1}{m^2} \min_{\sigma\in\varsigma(m)} \sum_{i=1}^{m} \|u^{(1)}_{\sigma(i)} - u^{(2)}_{\sigma(i+1)}\|^2$$

Ici, $u_i^{(k)}$ correspond à la k-eme ligne de la matrice u_i. De plus, la somme est prise modulo m, donc $m+1 = 1$.

On peut commencer par effectuer le changement de variable $\tilde{\sigma} = \sigma \circ s \circ \sigma^{-1}$ avec $s : k \mapsto k + 1$ mod m. Le terme devient alors :

$$\mathcal{D}(u_1, ..., u_m) = \frac{1}{m^2} \min_{\tilde{\sigma}\in\varsigma(m)} \sum_{i=1}^{m} \|u^{(1)}_{i} - u^{(2)}_{\tilde{\sigma}(i)}\|^2$$

Le problème d'optimisation est appellé le "Linear assignement problem" est peut être résolu. De plus, si on

nomme μ_m la mesure de probabilité associée aux u_i *uniquement* : $\mu_m = \frac{1}{m}\sum \delta_{u_i}$; on obtient :

$$\mathcal{D}(\mu_m) = \min_{T_{\#}\mu_1=\mu_2} \int\int_{\mathbb{R}^m \times R^m} || \, u - T(u') \, ||^2 \, d\mu_1 d\mu_2$$

Avec μ_1 et μ_2 les mesures marginales respectivement à la 1ère et 2de colonne. Donc, on obtient $\mathcal{D}(\mu_m) = \mathcal{W}_2^2(\mu_1, \mu_2)$ avec $\mathcal{W}_2$ la distance de Wasserstein entre deux mesures.

On peut alors prendre la limite en m pour obtenir une expression de champs moyens.

Proposition : $\mathcal{D}$ est convexe en μ.
Preuve :
Soit μ et ν deux mesures de probabilité.
On définit $f : (u, v) \mapsto || \, u - T(v) \, ||^2$ avec T une fonction telle que $T_{\#}\mu_1 = \mu_2$.
Soit P le polynome vérifiant $P(\lambda) = \mathcal{D}(\lambda\mu + (1-\lambda)\nu) - (\lambda\mathcal{D}(\mu) + (1-\lambda)\mathcal{D}(\nu))$
On sait que P s'annule en $\lambda = 0$ et en $\lambda = 1$, de plus $degP = 2$ puisque D es 2-homogènes. De ce fait :

$$P(\lambda) = \alpha(\lambda - 1)\lambda$$

Par définition, on a :

$$\begin{aligned}
\alpha &= \int\int_{\mathbb{R}^m \times R^m} f(u,v)\, \mu_1(du)\, \mu_2(dv) + \int\int_{\mathbb{R}^m \times R^m} f(u,v)\, \nu_1(du)\, \nu_2(dv) \\
&- \int\int_{\mathbb{R}^m \times R^m} f(u,v)\, \mu_1(du)\, \nu_2(dv) - \int\int_{\mathbb{R}^m \times R^m} f(u,v)\, \nu_1(du)\, \mu_2(dv) \\
&= \int\int_{\mathbb{R}^m \times R^m} f(u,v)\, (\mu_1 - \nu_1)(du)\, \mu_2(dv) + \int\int_{\mathbb{R}^m \times R^m} f(u,v)\, (\nu_1 - \mu_1)(du)\, \nu_2(dv) \\
&= \mathcal{D}(\mu - \nu)
\end{aligned}$$

Or, on peut échanger μ et ν dans la seconde égalité et obtenir le même résultat. On doit donc avoir $\mathcal{D}(\mu - \nu) \geq 0$. En faisant ainsi pour n'importe quel choix de T et en prenant le minimum, ceci complète la preuve.

2.5.4 Cas $s = 3$

Le cas $s = 2$ est particulier car il n'y a que 2 catégories de sous-neurones donc le couplage est unique. Pour $s \geq 3$, les neurones peuvent intervenir dans plusieurs couplages et on perd l'indépendance entre les lignes qui permettait de se ramener à effectuer une seule permutation. Pour comprendre comment la synchronisation fonctionne dans le cas $s \geq 3$, commençons par regarder le cas $s = 3$.

En suivant une logique similaire au cas précédent sur le choix des connexions (chaque neurone sera connecté à celui face à lui, celui en-dessous et celui au-dessus), on obtient un terme de la forme :

$$\mathcal{D}(u_1, ..., u_m) = \alpha(m) \min_{\sigma \in \varsigma(m)} \sum_{i=1}^{m} \|u^{(1)}_{\sigma(i)} - u^{(2)}_{\sigma(i+1)}\|^2 + \|u^{(2)}_{\sigma(i)} - u^{(3)}_{\sigma(i-2)}\|^2 + \|u^{(3)}_{\sigma(i)} - u^{(1)}_{\sigma(i+1)}\|^2$$

Soit après changement de variable :

$$\mathcal{D}(u_1,...,u_m) = \alpha(m) \min_{\sigma\in\varsigma(m)} \sum_{i=1}^{m} \|u_i^{(1)} - u_{\sigma(i)}^{(2)}\|^2 + \|u_i^{(2)} - u_{\sigma^{-2}(i)}^{(3)}\|^2 + \|u_i^{(3)} - u_{\sigma(i)}^{(1)}\|^2$$

Ce problème est bien plus difficile à résoudre. Cependant, la dépendance des termes entre eux peut nous donner une intuition du comportement :

En effet, si l'on suppose par exemple que $\forall i \in [[1;m]], u_{\sigma(i)}^{(2)} = u_i^{(1)}$, un rapide calcul nous montre que $\forall i \in [[1;m]], u_i^{(3)} = u_{\sigma(i)}^{(1)}$ et finalement le terme de synchronisation ne comporte plus qu'un seul terme : on retrouve le cas du Linear Assignement Problem. La proximité de deux lignes entre elles nous ramène donc à un probléme résolvable. Il semblerait donc que ce terme de synchronisation soit contrôlé par la proximité entre les mesures marginales des lignes, mesurée par la distance de Wasserstein.

2.5.5 Cas général et intuition

Les deux cas particuliers étudiés précédemment nous permettent d'établir une expression pour le terme de synchronisation :

$$\mathcal{D}(u_1,...,u_m) = \alpha(m) \min_{\sigma\in\varsigma(m)} \sum_{i=1}^{m} \sum_{1\le k<l\le s} \|u_i^{(k)} - u_{\sigma^{j(k,l)}(i)}^{(l)}\|^2$$

Notre intuition, que nous n'avons pas eu le temps de prouver, est la suivante :

$$\mathcal{D}(u_1,...,u_m) \le \sum_{1\le k<l\le s} \mathcal{W}_2^2(\mu^k,\mu^l)$$

On pourrait donc choisir le terme à droite de l'inégalité comme terme de synchronisation dans l'approximation des champs moyens.

Notons finalement une dernière inégalité, prise ici dans le cas $s = 3$:

$$\mathcal{W}_2^2(\mu^1,\mu^2) + \mathcal{W}_2^2(\mu^2,\mu^3) + \mathcal{W}_2^2(\mu^1,\mu^3) \le \mathcal{W}_2^2(\mu^{(1,2)},\mu^{(1,3)}) + \mathcal{W}_2^2(\mu^2,\mu^3)$$

2.6 References

— **Gradient Descent on Infinitely Wide Neural Networks : Global Convergence and Generalization**
- *Françis Bach, Lénaïc Chizat*