P.F.E.
Projet de Fin d'Etude

Deep Learning

Etude de l'article :

Layer-wise training of deep generative models

Ludovic Arnold, Yann Ollivier

Auteur :
Simon Bussy

Encadrant :
Jérémie Jakubowicz
Enseignant-Chercheur Département RST

Table des matières

1 Introduction

Voici une courte introduction du contexte dans lequel se trouve la recherche dans le domaine de l'apprentissage statistique et en particulier en ce qui concerne le récent regain d'intérêt pour les architechture profondes.
Les termes techniques utilisés seront repris en détail et explicités proprement dans la suite si nécessaire.

On parle d'architecture profonde pour des modèles multicouches d'architecture classique, mais qui comportent plusieurs couches cachées.
Avant 2006, les tentatives d'entraînement de modèles profonds ont été tenues en échec. En effet les résultats étaient moins bons que ceux obtenus pour des réseaux moins profonds, tant sur l'erreur d'apprentissage que sur l'erreur de test.

Puis différentes équipes de chercheurs ont commencé à avoir des résultats surprenants car surpassants les performances des meilleurs algorithmes d'apprentissage automatique alors développés.
C'est en fait la manière de gérer leur apprentissage qui a donné un regain d'intérêt à leur étude.

Ces avancées sont basées sur les trois principes clés suivants :

- Un apprentissage non-supervisé de représentations est utilisé pour pré-entraîner chaque couche.

- L'entraînement se fait couche par couche (*layer-wise*), chaque couche étant entraînée après la couche du dessous. La représentation apprise par une couche est alors utilisée comme entrée de la couche suivante.

- On effectue un apprentissage supervisé pour raffiner (*fine-tune*) toutes les couches.

Bien que le même type de constatation empirique a commencé à apparaître dans différentes équipes de recherche du monde entier, une justification théorique restait alors à consolider.
C'est précisement ce que proposent Ludovic Arnold et Yann Ollivier dans leur article *Layer-wise training of deep generative models* paru en février 2013, que j'ai étudié tout au long de ce projet de fin d'étude.

J'ai alors entièrement repris les démonstrations de l'article en adaptant si nécessaire les notations pour une clareté optimale et en détaillant proprement les points passés rapidement, voire laissés sous silence, par les auteurs. Puis dans un second temps, j'ai tenté de vérifier expérimentalement les principaux résultats déduits de l'étude théorique.

2 Modèles génératifs profonds

Dans la pratique, un modèle est défini par un algorithme qui, avant l'apprentissage, dispose d'un ensemble de variables indéterminées appelées paramètres. Au cours de la procédure d'apprentissage, les données sont utilisées pour choisir les valeurs des paramètres qui maximisent la capacité du modèle pour effectuer une tâche voulue. Cette capacité à exécuter est mesurée par ce qu'on appelle une fonction cible ou fonction objectif. Pour ce faire, nous nous tournons vers l'optimisation qui est une branche des mathématiques consistant en l'étude de la façon de choisir les paramètres pour optimiser une fonction objectif.

En ce qui concerne les données à partir desquelles apprendre, elles doivent être informatives pour la tâche cible. Par exemple, dans le domaine de l'apprentissage supervisé, l'objectif est d'apprendre un modèle, étant donné des exemples de ce que le modèle devrait faire dans plusieurs situations. Les données consistent alors en une série d'exemples (feature x , label y) qui décrivent comment le système devrait idéalement répondre à plusieurs entrées ou features.

Dans l'apprentissage de représentations, un algorithme d'apprentissage est utilisé pour trouver des caractéristiques intéressantes des données. L'apprentissage de représentations utiles peut être fait en pratique avec de l'apprentissage non supervisé où l'apprentissage se fait sur un ensemble de données d'entraînement x sans les labels y correspondants.

Un aspect particulièrement important pour faire cela est la possibilité de considérer plusieurs couches de traitement, à savoir une architecture profonde.
Une nouvelle façon de procéder à donc été introduite récemment et a porté ses fruits expérimentalement : essayer d'apprendre les caractéristiques d'une couche à la fois au lieu d'essayer d'apprendre toutes les couches en même temps.

2.1 Contexte et notations

Nous allons alors mettre en place dans cette partie le formalisme des modèles profonds, pour progressivement tenter de comprendre les difficultés qu'ils entraînent puis tenter de montrer qu'il sera possible d'apprendre une couche inférieure optimale avant d'apprendre les couches supérieures et d'étudier un critère valide à optimiser pour l'entraînement couche par couche.

Soit X une variable aléatoire sur un espace $\mathcal{D}$ de loi $\mathbb{P}_{\mathcal{D}}$ inconnue qu'on appellera variable observée (souvent $\mathcal{D} = \mathbb{R}^p$).

Soit $\theta = (\theta_1, ..., \theta_r) \in \mathbb{R}^r$ un r-uplet constitué des paramètres d'une loi notée $\mathbb{P}_\theta$ sur $\mathcal{D}$.

De façon classique, on cherche à déterminer θ de façon à approximer au mieux la loi $\mathbb{P}_{\mathcal{D}}$ au sens du maximum de vraissemblance.

Ayant un échantillon $(X_1, ...,X_n)$ tel que $\forall i \in [\![1, n]\!]$, $X_i \overset{i.i.d.}{\sim} \mathbb{P}_{\mathcal{D}}$ pour n fixé,

on appelle vraissemblance associée la quantité :

$$\mathbb{P}_\theta(X_1, ...,X_n) = \prod_{i=1}^{n} \mathbb{P}_\theta(X_i) \qquad (i.i.d.)$$

Rque : On notera indifféremment $\mathbb{P}_\theta$ pour la probabilité ou la densité dans le cas où l'espace $\mathcal{D}$ est discret ou non respectivement.

On a alors le maximum de vraissemblance :

$$\begin{aligned}\max_{\theta\in\mathbb{R}^r} \mathbb{P}_\theta(X_1,...,X_n) &= \max_{\theta\in\mathbb{R}^r} log \prod_{i=1}^{n} \mathbb{P}_\theta(X_i) \qquad (log \text{ croissante}) \\ &= \max_{\theta\in\mathbb{R}^r} \sum_{i=1}^{n} log\, \mathbb{P}_\theta(X_i) \\ &= \max_{\theta\in\mathbb{R}^r} \sum_{i=1}^{n} \frac{log\, \mathbb{P}_\theta(X_i)}{n}\end{aligned}$$

Plus n est grand, plus on a d'information sur la loi $\mathbb{P}_\mathcal{D}$.

On pose alors

$$\begin{aligned}\theta^* &= arg \max_{\theta\in\mathbb{R}^r} \lim_{n\to+\infty} \sum_{i=1}^{n} \frac{log\, \mathbb{P}_\theta(X_i)}{n} \\ &= arg \max_{\theta\in\mathbb{R}^r} \mathbb{E}_{X\sim\mathbb{P}_\mathcal{D}}[log\, \mathbb{P}_\theta(X)] \quad \text{(Loi forte gd nbres)}\end{aligned}$$

On définit la divergence de Kullback-Leibler entre les deux distributions de probabilité $\mathbb{P}_\mathcal{D}$ et $\mathbb{P}_\theta$ par :

$$D_{\mathrm{KL}}(\mathbb{P}_\mathcal{D}||\mathbb{P}_\theta) = \mathbb{E}_{X\sim\mathbb{P}_\mathcal{D}}[log\, \tfrac{\mathbb{P}_\mathcal{D}(X)}{\mathbb{P}_\theta(X)}]$$

D'où

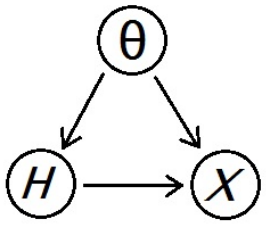
$$\theta^* = arg \min_{\theta\in\mathbb{R}^r} D_{\mathrm{KL}}(\mathbb{P}_\mathcal{D}||\mathbb{P}_\theta)$$

Approcher au mieux θ^* défini de ces deux façons équivalentes, à l'aide de la *log*-vraissemblance ou de la divergence de Kullback-Leibler, sera notre motivation première.

2.2 Modèles à variables latentes

Soit H une variable aléatoire sur un espace $\mathcal{H}$ de loi $\mathbb{P}_{\mathcal{H}}$, inobservable, qu'on appellera variable latente ou cachée.

On peut alors représenter graphiquement un modèle à variables latentes de la façon suivante :



On dit que H est une représentation de la variable aléatoire X.

On a alors

$$\begin{cases} \forall x \in \mathcal{D},\ \mathbb{P}_\theta(x) = \sum_{h \in \mathcal{H}} \mathbb{P}_\theta(x,h) \\ \forall h \in \mathcal{H},\ \mathbb{P}_\theta(h) = \sum_{x \in \mathcal{D}} \mathbb{P}_\theta(x,h) \end{cases}$$

où on note encore $\mathbb{P}_\theta$ la probabilité sur $\mathcal{D} \times \mathcal{H}$.

$$\text{et } \forall x \in \mathcal{D},\ \mathbb{P}_\theta(x) = \sum_{h \in \mathcal{H}} \mathbb{P}_\theta(x|h)\mathbb{P}_\theta(h)$$

Rque : En fait, on a $\forall h \in \mathcal{H},\ \mathbb{P}_\theta(h) = \mathbb{P}_\theta(\mathcal{D}, \{h\})$ et
$\forall x \in \mathcal{D},\ \mathbb{P}_\theta(x) = \mathbb{P}_\theta(\{x\}, \mathcal{H})$

Le principe du modèle en couche est alors de faire l'hypothèse suivante : la probabilité conditionnelle $\mathbb{P}_\theta(x|h)$ ne dépend que d'une partie des paramètres $\theta = (\theta_1, ..., \theta_r)$ et la probabilité $\mathbb{P}_\theta(h)$ de l'autre partie.

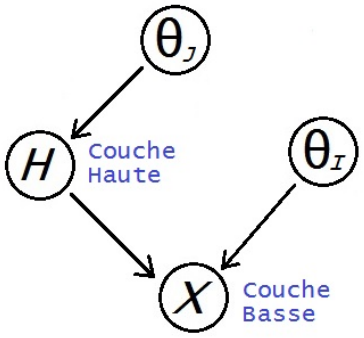
Autrement dit, on peut écrire :

$$\forall x \in \mathcal{D},\ \mathbb{P}_\theta(x) = \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_1,...,\theta_k}(x|h)\mathbb{P}_{\theta_{k+1},...,\theta_r}(h)$$

En notant I et J les ensembles $[\![1, k]\!]$ et $[\![k+1, r]\!]$ respectivement, on prend alors la notation suivante :

$$\forall x \in \mathcal{D},\ \mathbb{P}_\theta(x) = \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(x|h)\mathbb{P}_{\theta_J}(h)$$

On peut alors représenter graphiquement le modèle génératif de la façon suivante, qui est un cas particulier de la représentation précédente :



Le principe des modèles génératifs profonds est alors de répéter le même type de décomposition à la variable aléatoire H de façon récursive en définissant de la même façon de nouvelles variables latentes $H^{(1)}, H^{(2)}, ..., H^{(k_{max})}$ qui constitueront les différentes couches cachées du modèle.

On aura donc :

$$\begin{cases} \forall x \in \mathcal{D}, \mathbb{P}_\theta(x) = \displaystyle\sum_{h^{(1)} \in \mathcal{H}_1} \mathbb{P}_{\theta_{I_0}}(x|h^{(1)})\mathbb{P}_{\theta_{J_0}}(h^{(1)}) \\ \forall k \in [\![1, k_{max}-1]\!], \forall h^{(k)} \in \mathcal{H}_k,\ \mathbb{P}_\theta(h^{(k)}) = \displaystyle\sum_{h^{(k+1)} \in \mathcal{H}_{k+1}} \mathbb{P}_{\theta_{I_k}}(h^{(k)}|h^{(k+1)})\mathbb{P}_{\theta_{J_k}}(h^{(k+1)}) \end{cases}$$

Par soucis d'écriture, on ne s'interessera dès lors qu'à un pas de cette décomposition où la variable observée sera notée X et la variable latente notée H.

On étendra alors les raisonnements à toutes les couches en renommant les variables aléatoires.

2.3 Le problème de tractabilité des modèles profonds

Revenons maintenant au problème initial :

$$\text{Déterminer } \theta^* = arg \max_{\theta \in \mathbb{R}^r} \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log\ \mathbb{P}_\theta(X)]$$

On cherche donc θ tel que $\frac{\partial}{\partial \theta}(\mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log\ \mathbb{P}_\theta(X)]) = 0$

C'est-à-dire qu'on annule le gradient. On admet alors que cela revient à résoudre cette équation :

$$\mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[\frac{\partial}{\partial \theta}(log\ \mathbb{P}_\theta(X))] = 0$$

Soit $x \in \mathcal{D}$.

$$\mathbb{P}_\theta(x) = \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(x|h)\mathbb{P}_{\theta_J}(h)$$

Donc

$$\begin{aligned}
\frac{\partial}{\partial \theta}(log\ \mathbb{P}_\theta(X)) &= \frac{1}{\mathbb{P}_\theta(X)} \frac{\partial}{\partial \theta}(\sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\mathbb{P}_{\theta_J}(h)) \\
&= \frac{1}{\mathbb{P}_\theta(X)} \left(\sum_{h \in \mathcal{H}} \frac{\partial}{\partial \theta}[\mathbb{P}_{\theta_I}(X|h)]\mathbb{P}_{\theta_J}(h) + \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h) \frac{\partial}{\partial \theta}[\mathbb{P}_{\theta_J}(h)] \right)
\end{aligned}$$

Or $\forall h \in \mathcal{H}, \mathbb{P}_\theta(h|X) = \frac{\mathbb{P}_\theta(X,h)}{\mathbb{P}_\theta(X)} = \frac{\mathbb{P}_{\theta_I}(X|h)\mathbb{P}_{\theta_J}(h)}{\mathbb{P}_\theta(X)}$

Soit $\forall h \in \mathcal{H}$, $\frac{\mathbb{P}_{\theta_J}(h)}{\mathbb{P}_\theta(X)} = \frac{\mathbb{P}_\theta(h|X)}{\mathbb{P}_{\theta_I}(X|h)}$

D'où

$$\begin{aligned}\frac{\partial}{\partial\theta}(log\ \mathbb{P}_\theta(X)) &= \sum_{h\in\mathcal{H}} \frac{\partial}{\partial\theta}[\mathbb{P}_{\theta_I}(X|h)]\frac{\mathbb{P}_\theta(h|X)}{\mathbb{P}_{\theta_I}(X|h)} + \sum_{h\in\mathcal{H}} \frac{\mathbb{P}_\theta(h|X)}{\mathbb{P}_{\theta_J}(h)}\frac{\partial}{\partial\theta}[\mathbb{P}_{\theta_J}(h)] \\ &= \sum_{h\in\mathcal{H}} \frac{\partial}{\partial\theta}[log(\mathbb{P}_{\theta_I}(X|h))]\mathbb{P}_\theta(h|X) + \sum_{h\in\mathcal{H}} \mathbb{P}_\theta(h|X)\frac{\partial}{\partial\theta}[log(\mathbb{P}_{\theta_J}(h))]\end{aligned}$$

Soit donc

$$\begin{cases} \forall i \in I, \frac{\partial}{\partial\theta_i}(log\ \mathbb{P}_\theta(X)) = \sum_{h\in\mathcal{H}} \frac{\partial}{\partial\theta_i}[log(\mathbb{P}_{\theta_I}(X|h))]\mathbb{P}_\theta(h|X) \\ \forall j \in J, \frac{\partial}{\partial\theta_j}(log\ \mathbb{P}_\theta(X)) = \sum_{h\in\mathcal{H}} \mathbb{P}_\theta(h|X)\frac{\partial}{\partial\theta_j}[log(\mathbb{P}_{\theta_J}(h))] \end{cases}$$

La descente de gradient est une méthode numérique que l'on peut utiliser pour déterminer θ tel que $\frac{\partial}{\partial\theta}(log\ \mathbb{P}_\theta(X)) = 0$. Cette méthode consiste à modifier itérativement θ_i^t en $\theta_i^{t+1} = \theta_i^t + \epsilon_t \frac{\partial}{\partial\theta_i^t}(log\ \mathbb{P}_{\theta^t}(X))$ pour tout i$\in [\![1, r]\!]$ et $t \geq 1$.

On appelle ϵ_t le *learning rate* (ou taux d'apprentissage).

Sous de bonnes conditions sur ϵ_t que nous ne préciserons pas ici, on approche un maximum local pour la fonction $\theta \mapsto log\ \mathbb{P}_\theta(X)$.
Donc pour chaque i et à chaque itération, il faut échantilloner selon $\mathrm{P}_\theta(h|X)$ pour mettre à jour θ_i, ce qui devient vite très lourd en terme de temps de calcul pour r grand.

L'idée de l'article est alors de proposer une justification théorique d'un entraînement couche par couche de notre modèle génératif profond, pour justement palier à ce problème de dimensionalité et de tractabilité des algorithmes.

3 Entraînement couche par couche

3.1 Théorème central de l'article

Revenons à notre problème initial, à savoir déterminer :

$$\theta^* = arg \max_{\theta \in \mathbb{R}^r} \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log\, \mathbb{P}_\theta(X)] = arg \min_{\theta \in \mathbb{R}^r} D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || \mathbb{P}_\theta)$$

$$\text{avec } \forall x \in \mathcal{D}, \mathbb{P}_\theta(x) = \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(x|h)\mathbb{P}_{\theta_J}(h)$$

Donc $\theta^* = (\theta_I^*, \theta_J^*) = arg \max_{(\theta_I, \theta_J) \in \mathbb{R}^k \times \mathbb{R}^{r-k}} \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\mathbb{P}_{\theta_J}(h)]$

Posons alors

$$(\hat{\theta_I}, \hat{q}) = arg \max_{(\theta_I, q)} \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h) q_{\mathcal{D}}(h)]$$

$$\text{avec } \forall h \in \mathcal{H}, q_{\mathcal{D}}(h) = \sum_{x \in \mathcal{D}} q(h|x)\mathbb{P}_{\mathcal{D}}(x)$$

Rque : En pratique, $q_{\mathcal{D}}$ est inconnue puisque $\mathbb{P}_{\mathcal{D}}$ l'est.

On appelle $\hat{\theta_I}$ la *Best Optimistic Lower Layer*, qu'on notera dans la suite *B.O.L.L*, et on note :

$$\hat{q_{\mathcal{D}}}(h) = \sum_{x \in \mathcal{D}} \hat{q}(h|x)\mathbb{P}_{\mathcal{D}}(x)$$

Notons dès lors qu'il faut prendre garde aux différents espaces probabilisés sur lesquels on travaille :

- $(\mathcal{D}, \mathcal{B}(\mathcal{D}), \mathbb{P}_{\mathcal{D}})$
- $(\mathcal{D}, \mathcal{B}(\mathcal{D}), \mathbb{P}_{\theta})_{\theta \in \mathbb{R}^r}$
- $(\mathcal{H}, \mathcal{B}(\mathcal{H}), \mathbb{P}_{\mathcal{H}})$
- $(\mathcal{D} \times \mathcal{H}, \mathcal{B}(\mathcal{D} \times \mathcal{H}), \mathbb{P}_{\theta})$

où $\mathcal{B}(\mathcal{D})$ est la tribu borélienne de $\mathcal{D}$.

On en arrive alors à l'énoncé du théorème central de l'article.

Théorème 1 :

(i) $\left(\exists \hat{\theta_J} \in \mathbb{R}^{r-k}, \forall h \in \mathcal{H}, \mathbb{P}_{\hat{\theta_J}}(h) = \hat{q}_{\mathcal{D}}(h)\right) \Rightarrow (\hat{\theta_I}, \hat{\theta_J}) = (\theta_I^*, \theta_J^*)$

(ii) $\forall \theta_J \in \mathbb{R}^{r-k}, 0 \leq D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || \mathbb{P}_{\hat{\theta_I}, \theta_J})\text{-}D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || \mathbb{P}_{\theta_I^*, \theta_J^*}) \leq D_{\mathrm{KL}}(\hat{q}_{\mathcal{D}} || \mathbb{P}_{\theta_J})$

(i) signifie que si on peut parvenir à entraîner la couche supérieure à reproduire $\hat{q}_{\mathcal{D}}$ de façon parfaite, alors les paramètres obtenus sont optimaux. *(ii)* signifie que si on utilise un paramètre θ_J quelconque pour la couche supérieure en conjonction avec la *B.O.L.L.* , la différence de performance entre $(\hat{\theta_I}, \theta_J)$ et l'optimum global (θ_I^*, θ_J^*) est majorée par la divergence de Kullback-Leibler entre $\hat{q}_{\mathcal{D}}$ et $\mathbb{P}_{\theta_J}$.

Rques :

1) Il est important de noter que cette borne ne dépend pas de l'optimum global.

2) On a :

$$\textit{(ii)} \Rightarrow \min_{\theta_J \in \mathbb{R}^{r-k}} D_{\mathrm{KL}}(\hat{q}_{\mathcal{D}} || \mathbb{P}_{\theta_J}) \geq \min_{\theta_J \in \mathbb{R}^{r-k}} [D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || \mathbb{P}_{\hat{\theta_I}, \theta_J})\text{-}D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || \mathbb{P}_{\theta_I^*, \theta_J^*})] \geq 0$$

et pour $\theta_J = \hat{\theta_J}$ défini par *(i)* (sous réserve d'existence), on a

$$\begin{cases} D_{\mathrm{KL}}(\hat{q}_{\mathcal{D}}||\mathbb{P}_{\theta_J}) = 0 \\ D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}}||\mathbb{P}_{\hat{\theta_I},\theta_J})\text{-}D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}}||\mathbb{P}_{\theta_I^*,\theta_J^*})=0 \end{cases}$$

donc $arg \min_{\theta_J \in \mathbb{R}^{r-k}} D_{\mathrm{KL}}(\hat{q}_{\mathcal{D}}||\mathbb{P}_{\theta_J}) = \hat{\theta_J}$

Le théorème suggère donc fortement d'entraîner la couche supérieure de la façon suivante :

$$\hat{\theta_J} = arg \min_{\theta_J \in \mathbb{R}^{r-k}} D_{\mathrm{KL}}(\hat{q}_{\mathcal{D}}||\mathbb{P}_{\theta_J})$$

On retrouve donc la formulation d'un problème à la couche supérieure similaire à celui de notre couche inférieure.

Passons alors à la preuve de ce théorème.

3.2 Preuve du théorème

Nous allons introduire dans cette partie différentes notions qui nous serviront par la suite.
Commençons par une définition.

Définition 1 : Soit $\theta_I \in \mathbb{R}^k$.
On définit la *Best Latent Marginal* associée à θ_I de la façon suivante :

$$BLM(\theta_I) = \hat{\mathcal{Q}}_{\theta_I,\mathcal{D}} = arg \max_{\mathcal{Q}} \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\mathcal{Q}(h)]$$
$$\text{où } \mathcal{Q} \in \{\text{probabilités sur } \mathcal{H}\}$$

En utilisant le paramètre θ_I pour la couche inférieure, la $BLM(\theta_I)$ est donc la marginale sur $\mathcal{H}$ qui donnera la meilleure performance en *log*-vraissemblance de notre modèle parmis toutes les probabilités possibles pour la couche supérieure.

On définit alors la *BLM upper bound* comme étant la valeur de la borne de *log*-vraissemblance correspondante :

$$\mathcal{U}_{\mathcal{D}}(\theta_I) = \max_{\mathcal{Q}} \ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\mathcal{Q}(h)]$$

On en arrive alors à une première propriété :

Propriété 1 : $\mathcal{U}_{\mathcal{D}}(\theta_I) = \max_{q} \ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h) q_{\mathcal{D}}(h)]$

Preuve : Soit $\mathcal{Q}$ une loi de probabilité quelconque sur $\mathcal{H}$.
On peut alors toujours définir une probabilité q sur $\mathcal{H}$ telle que

$$\forall h \in \mathcal{H}, q(h|X) = \mathcal{Q}(h)$$

Alors

$$\begin{aligned}
\hat{\mathcal{Q}}_{\theta_I,\mathcal{D}} &= arg \ \max_{\mathcal{Q}} \ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h) \sum_{x\in\mathcal{D}} \mathcal{Q}(h)\mathbb{P}_{\mathcal{D}}(x)] \\
&= arg \ \max_{q} \ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h) \sum_{x\in\mathcal{D}} q(h|x)\mathbb{P}_{\mathcal{D}}(x)] \\
&= arg \ \max_{q} \ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h) q_{\mathcal{D}}(h)]
\end{aligned}$$

□

On a donc en conséquence direct que la *B.O.L.L.* $\hat{\theta_I} = arg \ \max_{\theta_I} \mathcal{U}_{\mathcal{D}}(\theta_I)$, ce qui justifie au passage l'appelation de *Best Optimistic Lower Layer*.

Puis soit $h \in \mathcal{H}$.

$$\begin{aligned}
\hat{q}_{\mathcal{D}}(h) &= \sum_{x \in \mathcal{D}} \hat{q}(h|x)\mathbb{P}_{\mathcal{D}}(x) \\
&= \sum_{x \in \mathcal{D}} arg\ \max_{q} \left(\max_{\theta_I} \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h' \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h')q_{\mathcal{D}}(h')] \right) (h|x)\mathbb{P}_{\mathcal{D}}(x) \\
&= arg\ \max_{q}\ \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h' \in \mathcal{H}} \mathbb{P}_{\hat{\theta}_I}(X|h')q_{\mathcal{D}}(h')](h) \\
&= arg\ \max_{\mathcal{Q}}\ \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h' \in \mathcal{H}} \mathbb{P}_{\hat{\theta}_I}(X|h')\mathcal{Q}(h')](h) \quad (prop\ 1) \\
&= \hat{\mathcal{Q}}_{\hat{\theta_I},\mathcal{D}}(h)
\end{aligned}$$

$$\text{donc } \hat{q}_{\mathcal{D}} = \hat{\mathcal{Q}}_{\hat{\theta_I},\mathcal{D}}$$

Nous comprendrons alors plus tard (proposition 4) en quoi il est utile de déterminer $\hat{q}_{\mathcal{D}}$ en passant par une maximisation sur q, plutôt que $\mathcal{Q}$, en plus du fait que $\hat{q}_{\mathcal{D}}$ apparaisse dans la borne du *(ii)*.

Propriété 2 : Soit donc $\hat{\theta_I} = arg\ \max_{\theta_I} \mathcal{U}_{\mathcal{D}}(\theta_I)$ et $\hat{\mathcal{Q}}_{\hat{\theta_I},\mathcal{D}}$ la *BLM* correspondante.
Alors $\left(\exists \theta_J \in \mathbb{R}^{r-k}, \forall h \in \mathcal{H}, \mathbb{P}_{\theta_J}(h) = \hat{\mathcal{Q}}_{\hat{\theta_I},\mathcal{D}}(h)\right) \Rightarrow \hat{\theta_I} = \theta_I^*$

Preuve : On commence par définir la *BLM upper bound* du modèle pour $\theta_I \in \mathbb{R}^k$:

$$\mathcal{U}_{\mathcal{D}}^{model}(\theta_I) = \max_{\theta_J \in \mathbb{R}^{r-k}} \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\mathbb{P}_{\theta_J}(h)]$$

On a alors par définition $\theta_I^* = arg\ \max_{\theta_I \in \mathbb{R}^k} \mathcal{U}_{\mathcal{D}}^{model}(\theta_I)$

$$\text{puis } \forall \theta_I \in \mathbb{R}^k, \mathcal{U}_{\mathcal{D}}^{model}(\theta_I) \leq \mathcal{U}_{\mathcal{D}}(\theta_I)$$

car pour $\theta_I \in \mathbb{R}^k$,

$$\{ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\mathbb{P}_{\theta_J}(h)]/\theta_J \in \mathbb{R}^{r-k}\} \subset$$
$$\{\mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\mathcal{Q}(h)]/\mathcal{Q} \in \{\text{probabilités sur } \mathcal{H}\}\}$$

Ensuite, $\hat{\theta_I} = arg \max_{\theta_I} \mathcal{U}_{\mathcal{D}}(\theta_I)$

$$\text{donc } \forall \theta_I \in \mathbb{R}^k, \mathcal{U}_{\mathcal{D}}(\theta_I) \leq \mathcal{U}_{\mathcal{D}}(\hat{\theta_I})$$

Supposons qu'il existe $\theta_J \in \mathbb{R}^{r-k}$ tel que $\forall h \in \mathcal{H}, \mathbb{P}_{\theta_J}(h) = \hat{\mathcal{Q}}_{\hat{\theta_I},\mathcal{D}}(h)$

Alors

$$\begin{aligned}
\mathcal{U}_{\mathcal{D}}(\hat{\theta_I}) &= \max_{\mathcal{Q}} \ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\hat{\theta_I}}(X|h)\mathcal{Q}(h)] \\
&= \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\hat{\theta_I}}(X|h)\hat{\mathcal{Q}}_{\hat{\theta_I},\mathcal{D}}(h)] \\
&= \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\hat{\theta_I}}(X|h)\mathbb{P}_{\theta_J}(h)] \\
&\leq \mathcal{U}_{\mathcal{D}}^{model}(\hat{\theta_I})
\end{aligned}$$

Et comme $\mathcal{U}_{\mathcal{D}}(\hat{\theta_I}) \geq \mathcal{U}_{\mathcal{D}}^{model}(\hat{\theta_I})$, on a en fait $\mathcal{U}_{\mathcal{D}}(\hat{\theta_I}) = \mathcal{U}_{\mathcal{D}}^{model}(\hat{\theta_I})$

On a donc $\forall \theta_I \in \mathbb{R}^k, \mathcal{U}_{\mathcal{D}}^{model}(\theta_I) \leq \mathcal{U}_{\mathcal{D}}(\theta_I) \leq \mathcal{U}_{\mathcal{D}}(\hat{\theta_I}) \leq \mathcal{U}_{\mathcal{D}}^{model}(\hat{\theta_I})$

$$\text{Soit donc } \hat{\theta_I} = arg \max_{\theta_I \in \mathbb{R}^k} \mathcal{U}_{\mathcal{D}}^{model}(\theta_I) = \theta_I^*$$

Ce qui achève la preuve de la propriété 2. □

Ainsi

$$\left(\exists \hat{\theta_J} \in \mathbb{R}^{r-k}, \forall h \in \mathcal{H}, \mathbb{P}_{\hat{\theta_J}}(h) = \hat{q}_{\mathcal{D}}(h)\right) \Leftrightarrow \left(\exists \hat{\theta_J} \in \mathbb{R}^{r-k}, \forall h \in \mathcal{H}, \mathbb{P}_{\hat{\theta_J}}(h) = \hat{\mathcal{Q}}_{\hat{\theta_I},\mathcal{D}}(h)\right) \ (prop\ 1)$$
$$\Rightarrow \hat{\theta_I} = \theta_I^* \quad (prop\ 2)$$

Puis $\theta_J^* = arg \max_{\theta_J \in \mathbb{R}^{r-k}} \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h)\mathbb{P}_{\theta_J}(h)]$

Et $\forall \theta_J \in \mathbb{R}^{r-k}$,

$$\begin{aligned}
\mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h)\mathbb{P}_{\theta_J}(h)] &\leq \max_{\theta_J \in \mathbb{R}^{r-k}} \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h)\mathbb{P}_{\theta_J}(h)] \\
&\leq \max_{\mathcal{Q}} \ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h)\mathcal{Q}(h)] \\
&= \max_{q} \ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h)q_{\mathcal{D}}(h)] \quad (prop\ 1) \\
&= \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h)\hat{q}_{\mathcal{D}}(h)] \quad (par\ déf.) \\
&= \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h)\mathbb{P}_{\hat{\theta_J}}(h)] \quad (par\ hyp.)
\end{aligned}$$

D'où $\max_{\theta_J \in \mathbb{R}^{r-k}} \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h)\mathbb{P}_{\theta_J}(h)] = \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h)\mathbb{P}_{\hat{\theta_J}}(h)]$

et donc $\hat{\theta_J} = \theta_J^*$

D'où le *(i)* du théorème 1.

Nous allons maintenant nous concentrer sur la preuve du *(ii)*.

Soit $\theta_J \in \mathbb{R}^{r-k}$.

D'après notre problème initial, la différence de performance de *log*-vraissemblance entre deux distributions de probabilité p_1 et p_2 quelconques est donnée par :

$$\mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log\ p_1(X)] - \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log\ p_2(X)]$$

$$\text{ou par } D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}}||p_1) - D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}}||p_2)$$

On cherche ici à comparer les performances de $\mathbb{P}_{\hat{\theta_I},\theta_J}$ et $\mathbb{P}_{\theta_I^*,\theta_J^*}$.

La propriété suivante va justement nous permettre de faire cela.

Propriété 3 :

(1)
$D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}}||\mathbb{P}_{\hat{\theta_I},\theta_J}) - D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}}||\mathbb{P}_{\theta_I^*,\theta_J^*}) \leq D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}}||\mathbb{P}_{\hat{\theta_I},\theta_J}) - D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}}||\hat{q}_{\mathcal{D},\hat{\theta_I}}) = p$

$$\text{où } \forall x \in \mathcal{D}, \hat{q}_{\mathcal{D},\hat{\theta_I}}(x) = \sum_{h\in\mathcal{H}} \mathbb{P}_{\hat{\theta_I}}(x|h)\hat{q}_{\mathcal{D}}(h)$$

qui est donc la distribution obtenue en utilisant la $BLM(\hat{\theta}_I)$.

(2) $p \leq D_{\mathrm{KL}}(\hat{q}_{\mathcal{D}}||\mathbb{P}_{\theta_J})$

Preuve : Soit $x \in \mathcal{D}$.

On notera

$$\begin{aligned} p_1(x) = \mathbb{P}_{\hat{\theta_I},\theta_J}(x) &= \sum_{h\in\mathcal{H}} \mathbb{P}_{\hat{\theta_I}}(x|h)\mathbb{P}_{\theta_J}(h) \\ p_2(x) = \mathbb{P}_{\theta_I^*,\theta_J^*}(x) &= \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I^*}(x|h)\mathbb{P}_{\theta_J^*}(h) \\ \text{et } p_3(x) = \sum_{h\in\mathcal{H}} p_3(x,h) &= \sum_{h\in\mathcal{H}} \mathbb{P}_{\hat{\theta_I}}(x|h)\hat{q}_{\mathcal{D}}(h) = \hat{q}_{\mathcal{D},\hat{\theta_I}}(x) \end{aligned}$$

En reprenant les mêmes notations que précedemment, on a

$$\theta_I^* = arg \max_{\theta_I \in \mathbb{R}^k} \mathcal{U}_{\mathcal{D}}^{model}(\theta_I) \text{ et } \hat{\theta_I} = arg \max_{\theta_I \in \mathbb{R}^k} \mathcal{U}_{\mathcal{D}}(\theta_I)$$

puis on a déjà vu que

$$\begin{aligned}
\mathcal{U}_{\mathcal{D}}^{model} \leq \mathcal{U}_{\mathcal{D}} &\Rightarrow \max_{\theta_I \in \mathbb{R}^k} \mathcal{U}_{\mathcal{D}}^{model}(\theta_I) \leq \max_{\theta_I \in \mathbb{R}^k} \mathcal{U}_{\mathcal{D}}(\theta_I) \\
&\Leftrightarrow \mathcal{U}_{\mathcal{D}}^{model}(\theta_I^*) \leq \mathcal{U}_{\mathcal{D}}(\hat{\theta_I}) \\
&\Leftrightarrow \max_{\theta_J \in \mathbb{R}^{r-k}} \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h) \mathbb{P}_{\theta_J}(h)] \leq \max_{q} \ \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\hat{\theta}_I}(X|h) q_{\mathcal{D}}(h)] \\
&\Leftrightarrow \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I^*}(X|h) \mathbb{P}_{\theta_J^*}(h)] \leq \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\hat{\theta}_I}(X|h) \hat{q}_{\mathcal{D}}(h)] \\
&\Leftrightarrow \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log\ p_2(X)] \leq \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log\ p_3(X)] \\
&\Leftrightarrow D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_2) \geq D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_3)
\end{aligned}$$

et par définition, $D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_2) = \min_{\theta \in \mathbb{R}^r} D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || \mathbb{P}_{\theta}) \leq D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_1)$

Donc $0 \leq D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_3) \leq D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_2) \leq D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_1)$

et $0 \leq D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_1) - D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_2) \leq D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_1) - D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}} || p_3) = p$

d'où le *(1)*.

Passons à la preuve du *(2)*.

Pour cela, nous aurons besoin du lemme suivant.

Lemme : Pour $n \in \mathbb{N}$,

$$\forall((a_i),(b_i)) \in (\mathbb{R}^{+*})^n \times (\mathbb{R}^{+*})^n, \quad log\ \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \leq \frac{1}{\sum_{i=1}^{n} a_i} \sum_{i=1}^{n}\ a_i log\ \frac{a_i}{b_i}$$

Preuve : Soit $n \in \mathbb{N}$.

Posons pour $t \in \mathbb{R}^{+*}, f(t) = t\ log(t)$.

$f" > 0$ donc f est strictement convexe.

Donc d'après l'inégalité de Jensen,

$$\forall(\alpha_i) \in (\mathbb{R}^{+})^n, \left(\sum_{i=1}^{n}\ \alpha_i = 1 \Rightarrow \forall(t_i) \in (\mathbb{R}^{+*})^n, \sum_{i=1}^{n}\ \alpha_i f(t_i) \geq f(\sum_{i=1}^{n}\ \alpha_i t_i)\right)$$

Posons alors pour i $\in [\![1, n]\!]$,

$$\alpha_i = \frac{b_i}{\sum_{j=1}^{n}\ b_j} \in \mathbb{R}^{+*} \text{ et } t_i = \frac{a_i}{b_i} \in \mathbb{R}^{+*}$$

$$\text{Alors} \quad \sum_{i=1}^{n}\ \alpha_i = 1$$

Et donc :

$$\sum_{i=1}^{n} \frac{b_i}{\sum_{j=1}^{n} b_j} \frac{a_i}{b_i} log\ \frac{a_i}{b_i} = \sum_{i=1}^{n} \frac{a_i}{\sum_{j=1}^{n} b_j} log\ \frac{a_i}{b_i}$$
$$\geq \frac{\sum_{i=1}^{n} a_i}{\sum_{j=1}^{n} b_j} log\ \frac{\sum_{i=1}^{n} a_i}{\sum_{j=1}^{n} b_j}$$

En simplifiant par $\sum_{j=1}^{n} b_j$ des deux côtés de l'inégalité, on obtient le résultat attendu. □

On peut alors écrire

$$p = \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log\ p_3(X)] - \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log\ p_1(X)]$$
$$= \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}\left[log\ \frac{\sum_{h\in\mathcal{H}} \mathbb{P}_{\hat{\theta_I}}(X|h)\hat{q}_{\mathcal{D}}(h)}{\sum_{h\in\mathcal{H}} \mathbb{P}_{\hat{\theta_I}}(X|h)\mathbb{P}_{\theta_J}(h)} \right]$$

Donc d'après le lemme et en supposant $\mathcal{H}$ fini,

$$
\begin{aligned}
p &\leq \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}\left[\frac{1}{\sum_{h\in\mathcal{H}}\mathbb{P}_{\hat{\theta}_I}(X|h)\hat{q}_{\mathcal{D}}(h)}\ \sum_{h\in\mathcal{H}}\mathbb{P}_{\hat{\theta}_I}(X|h)\hat{q}_{\mathcal{D}}(h)log\ \frac{\mathbb{P}_{\hat{\theta}_I}(X|h)\hat{q}_{\mathcal{D}}(h)}{\mathbb{P}_{\hat{\theta}_I}(X|h)\mathbb{P}_{\theta_J}(h)}\right] \quad (croissance\ de\ \mathbb{E}) \\
&= \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}\left[\frac{1}{p_3(x)}\ \sum_{h\in\mathcal{H}}p_3(x,h)\ log\ \frac{\hat{q}_{\mathcal{D}}(h)}{\mathbb{P}_{\theta_J}(h)}\right] \\
&= \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[\sum_{h\in\mathcal{H}}p_3(h|x)\ log\ \frac{\hat{q}_{\mathcal{D}}(h)}{\mathbb{P}_{\theta_J}(h)}] \\
&= \sum_{x\in\mathcal{D}}\ \mathbb{P}_{\mathcal{D}}(x)\sum_{h\in\mathcal{H}}p_3(h|x)\ log\ \frac{\hat{q}_{\mathcal{D}}(h)}{\mathbb{P}_{\theta_J}(h)} \\
&= \sum_{h\in\mathcal{H}}\hat{q}_{\mathcal{D}}(h)\ log\ \frac{\hat{q}_{\mathcal{D}}(h)}{\mathbb{P}_{\theta_J}(h)} \quad (voir\ (*)) \\
&= D_{\mathrm{KL}}(\hat{q}_{\mathcal{D}}||\mathbb{P}_{\theta_J})
\end{aligned}
$$

(*) car $\sum_{x\in\mathcal{D}}\mathbb{P}_{\mathcal{D}}(x)p_3(h|x) = \sum_{x\in\mathcal{D}}\mathbb{P}_{\mathcal{D}}(x)\hat{q}(h|x) = \hat{q}_{\mathcal{D}}(h)$ qui est vrai en vertu de la proposition 4 énoncée ci-après.

Cela achève donc la preuve de *(2)*.

□

D'où le *(ii)* du théorème 1 qui donne donc une borne de la perte de performance dans le cas où l'entraînement des couches supérieures ne parvient pas à atteindre la *BLM*.

Autrement dit, lorsque la formation des couches supérieures est imparfaite, comme cela peut être le cas en pratique, l'erreur globale admet une borne supérieure qui est exactement le critère optimisé pour les couches supérieures.

On en a donc fini avec la preuve du théorème 1.

3.3 Incorporation des données

Comme nous l'avons déjà mentionné plusieurs fois, il n'est pas évident de prétendre qu'il serait théoriquement plus intéressant de travailler avec la probabilité conditionnelle $q(h|x)$ et de définir alors une probabilité $q_{\mathcal{D}}$ sur $\mathcal{H}$, plutôt que de travailler directement sur les probabilités Q sur $\mathcal{H}$ (ici "travailler" est assez flou mais on a déjà détaillé les maximisations auxquelles on fait implicitement référence).

On dira qu'on *incorpore les données grâce à* $q_{\mathcal{D}}$ dans le sens où cette distribution dépend de $\mathcal{D}$, ce qui n'est pas le cas lorsqu'on travaille avec Q quelconque.

C'est justement l'objet de cette sous-partie que de montrer l'intérêt théorique de cette manière de procéder.

On énonce pour cela la proposition suivante :

Propriété 4 : Soit $\theta_I \in \mathbb{R}^k$ et soit Q une probabilité quelconque sur $\mathcal{H}$.

On définit alors la fonction

$$\phi_{\theta_I} : Q \mapsto Q_{\mathcal{D}}^{cond}$$

où

$$\begin{cases} \forall h \in \mathcal{H},\, Q_{\mathcal{D}}^{cond}(h) = \displaystyle\sum_{x \in \mathcal{D}} Q^{cond}(h|x)\mathbb{P}_{\mathcal{D}}(x) \\ \forall (x,h) \in \mathcal{D} \times \mathcal{H},\, Q^{cond}(h|x) = \dfrac{\mathbb{P}_{\theta_I}(x|h)\,Q(h)}{\displaystyle\sum_{h' \in \mathcal{H}} \mathbb{P}_{\theta_I}(x|h')\,Q(h')} \end{cases}$$

Alors

(1) $\mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\,Q(h)] \leq \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\phi_{\theta_I}(Q)(h)]$

(2)
$BLM(\theta_I) \in \{Q, \phi_{\theta_I}(Q) = Q\} = \{Q, \delta\ \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\,Q(h)] = 0\}$

(3) Appliquer la fonction ϕ_{θ_I} revient à effectuer un pas de l'algorithme *EM* (*Expectation Maximisation*) dans l'optimisation de la $BLM(\theta_I)$.

Preuve :

Nous allons commencer par prouver le *(3)* dans la mesure où *(3)* $\Rightarrow$ *(1)*. En effet, en rappelant que

$$BLM(\theta_I) = arg \max_Q \ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\,Q(h)]$$

et sachant que l'algorithme *EM* utilisé dans la maximisation que définit la $BLM(\theta_I)$ va améliorer la *log*-vraissemblance $\mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\,Q(h)]$ à chaque pas (par définition même de cet algorithme), on a alors clairement que *(1)* est une simple conséquense de *(3)*.

Pour simplifier, considérons que $(X_1, ..., X_n)$ est un échantillon *i.i.d.* tiré suivant $\mathbb{P}_{\mathcal{D}}$ uniformément, *i.e.* que $\forall i \in [\![1, n]\!], \mathbb{P}_{\mathcal{D}}(X_i) = \frac{1}{n}$

De plus, nous noterons $\mathbb{P}_Q$ la probabilité sur $\mathcal{D} \times \mathcal{H}$ définie par $\mathbb{P}_Q(X, H) = \mathbb{P}_{\theta_I}(X|H)\,Q(H)$.

Et en notant $\vec{x} = (x_1, ..., x_n)$ puis $\vec{h} = (h_1, ..., h_n)$, on étend cette probabilité de la façon suivante :

$$\mathbb{P}_Q(\vec{x}, \vec{h}) = \prod_{i=1}^{n} \mathbb{P}_Q(x_i, h_i) = \prod_{i=1}^{n} \mathbb{P}_{\theta_I}(x_i|h_i)\,Q(h_i)$$

Appliquons alors l'algorithme *EM* dans les conditions que nous venons d'introduire et avec les notations prises.

Nous aurons alors pour un pas de cet algorithme, de t à t+1, l'expression classique suivante.

$$Q_{t+1} = arg\ \max_{Q} \sum_{\vec{h}} \mathbb{P}_t(\vec{h}|\vec{x}) log\ \mathbb{P}_Q(\vec{x},\vec{h})$$
$$\text{où} \quad \mathbb{P}_t(\vec{x},\vec{h}) = \mathbb{P}_{\theta_I}(\vec{x}|\vec{h})\, Q_t(\vec{h})$$

Et

$$\begin{aligned}
\sum_{\vec{h}} \mathbb{P}_t(\vec{h}|\vec{x}) log\ \mathbb{P}_Q(\vec{x},\vec{h}) &= \sum_{\vec{h}} \mathbb{P}_t(\vec{h}|\vec{x}) log \prod_{i=1}^{n}\ \mathbb{P}_Q(x_i,h_i) \\
&= \sum_i \sum_{\vec{h}} \mathbb{P}_t(\vec{h}|\vec{x}) log\ \mathbb{P}_Q(x_i,h_i) \\
&= \sum_i \sum_{h_1,\dots,h_n} log\ \mathbb{P}_Q(x_i,h_i) \prod_j \mathbb{P}_t(h_j|x_j) \quad (i.i.d.) \\
&= \sum_i \sum_{h_i} log\ \mathbb{P}_Q(x_i,h_i)\ \mathbb{P}_t(h_i|x_i) \prod_{j\neq i} \sum_{h_j} \mathbb{P}_t(h_j|x_j) \\
&= \sum_i \sum_{h_i} log\ \mathbb{P}_Q(x_i,h_i)\ \mathbb{P}_t(h_i|x_i) \quad (car\ \sum_{h_j} \mathbb{P}_t(h_j|x_j) = 1) \\
&= \sum_h \sum_i log\ \mathbb{P}_Q(x_i,h)\ \mathbb{P}_t(h|x_i) \\
&= \sum_h \sum_i (log\ \mathbb{P}_{\theta_I}(x_i|h) + log\ Q(h))\ \mathbb{P}_t(h|x_i)
\end{aligned}$$

Donc $Q_{t+1} = arg\ \max_{Q} \sum_h \sum_i (log\ Q(h))\ \mathbb{P}_t(h|x_i)$

qui est une quantité concave en Q, il suffit donc de déterminer la distribution Q qui annule la dérivée (par rapport à Q) pour exiber l'*argmax*. En remplaçant Q par $Q + \delta Q$ avec δQ infinitésimal, la variation de cette quantité est donnée par :

$$\begin{aligned}
\delta \sum_h \sum_i (log\ Q(h))\ \mathbb{P}_t(h|x_i) &= \sum_h \sum_i (\delta log\ Q(h))\ \mathbb{P}_t(h|x_i) \\
&= \sum_h \frac{\delta Q(h)}{Q(h)} \sum_i \mathbb{P}_t(h|x_i)
\end{aligned}$$

Si on prend alors

$$\begin{aligned}
Q(h) &= (Q_t)^{cond}_{\mathcal{D}}(h) \\
&= \sum_{x \in \mathcal{D}} Q_t^{cond}(h|x)\mathbb{P}_{\mathcal{D}}(x) \\
&= \sum_i \frac{1}{n} Q_t^{cond}(h|x_i) \quad (par\ hyp.) \\
&= \sum_i \frac{1}{n} \frac{\mathbb{P}_{\theta_I}(x_i|h)\, Q_t(h)}{\sum_{h' \in \mathcal{H}} \mathbb{P}_{\theta_I}(x_i|h')\, Q_t(h')} \\
&= \sum_i \frac{1}{n} \frac{\mathbb{P}_t(x_i, h)}{\mathbb{P}_t(x_i)} \\
&= \frac{1}{n} \sum_i \mathbb{P}_t(h|x_i)
\end{aligned}$$

On a donc

$$\sum_h \frac{\delta Q(h)}{Q(h)} \sum_i \mathbb{P}_t(h|x_i) = n \sum_h \delta Q(h)$$

Or

$$\begin{aligned}
\sum_h Q(h) + \delta Q(h) &= 1 \qquad (car\ Q + \delta Q\ est\ une\ proba.) \\
&= \sum_h Q(h) + \sum_h \delta Q(h) \\
&= 1 + \sum_h \delta Q(h) \qquad (car\ Q\ est\ une\ proba.) \\
&\Rightarrow \sum_h \delta Q(h) = 0
\end{aligned}$$

D'où $Q_{t+1} = (Q_t)^{cond}_{\mathcal{D}} = \phi_{\theta_I}(Q_t)$

On a donc prouvé le *(3)*, et donc également le *(1)*.

Reste à prouver le *(2)*.

Montrons d'abord que $BLM(\theta_I) \in \{Q, \phi_{\theta_I}(Q) = Q\}$

On a $BLM(\theta_I) = \hat{\mathcal{Q}}_{\theta_I,\mathcal{D}} = arg \max_Q \mathbb{E}_{X\sim\mathbb{P}_\mathcal{D}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)Q(h)]$

Supposont alors que $\phi_{\theta_I}(\hat{\mathcal{Q}}_{\theta_I,\mathcal{D}}) \neq \hat{\mathcal{Q}}_{\theta_I,\mathcal{D}}$.

D'après *(1)*,

$$\mathbb{E}_{X\sim\mathbb{P}_\mathcal{D}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\hat{\mathcal{Q}}_{\theta_I,\mathcal{D}}(h)] \leq \mathbb{E}_{X\sim\mathbb{P}_\mathcal{D}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\phi_{\theta_I}(\hat{\mathcal{Q}}_{\theta_I,\mathcal{D}})(h)]$$

Et

$\mathbb{E}_{X\sim\mathbb{P}_\mathcal{D}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\hat{\mathcal{Q}}_{\theta_I,\mathcal{D}}(h)] < \mathbb{E}_{X\sim\mathbb{P}_\mathcal{D}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\phi_{\theta_I}(\hat{\mathcal{Q}}_{\theta_I,\mathcal{D}})(h)]$ est absurde par définition de la $BLM(\theta_I)$.

D'où

$$\mathbb{E}_{X\sim\mathbb{P}_\mathcal{D}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\hat{\mathcal{Q}}_{\theta_I,\mathcal{D}}(h)] = \mathbb{E}_{X\sim\mathbb{P}_\mathcal{D}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\phi_{\theta_I}(\hat{\mathcal{Q}}_{\theta_I,\mathcal{D}})(h)]$$

ce qui est absurde d'après notre hypothèse, qui s'avère donc fausse.

Soit

$$\phi_{\theta_I}(\hat{\mathcal{Q}}_{\theta_I,\mathcal{D}}) = \hat{\mathcal{Q}}_{\theta_I,\mathcal{D}}$$

$$i.e. \quad BLM(\theta_I) \in \{Q, \phi_{\theta_I}(Q) = Q\}$$

Reste maintenant à montrer que les points fixes de la fonction ϕ_{θ_I} coïncident avec les points critiques de la *log*-vraissemblance $\mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log\sum_{h\in\mathcal{H}}\mathbb{P}_{\theta_I}(X|h)\,Q(h)]$.

Les points critiques en question sont ceux pour lesquels

$$\delta\ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log\sum_{h\in\mathcal{H}}\mathbb{P}_{\theta_I}(X|h)\,Q(h)] = 0$$

Et

$$\begin{aligned}
\delta\ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log\sum_{h\in\mathcal{H}}\mathbb{P}_{\theta_I}(X|h)\,Q(h)] &= \frac{1}{n}\delta\sum_i\ log\sum_{h\in\mathcal{H}}\mathbb{P}_{\theta_I}(x_i|h)\,Q(h) \\
&= \frac{1}{n}\sum_i\ \frac{\delta\sum_{h\in\mathcal{H}}\mathbb{P}_{\theta_I}(x_i|h)\,Q(h)}{\sum_{h\in\mathcal{H}}\mathbb{P}_{\theta_I}(x_i|h)\,Q(h)} \\
&= \frac{1}{n}\sum_i\ \frac{\sum_{h\in\mathcal{H}}\mathbb{P}_{\theta_I}(x_i|h)\delta\,Q(h)}{\mathbb{P}_Q(x_i)} \\
&= \frac{1}{n}\sum_{h\in\mathcal{H}}\delta\,Q(h)\sum_i\frac{\mathbb{P}_{\theta_I}(x_i|h)}{\mathbb{P}_Q(x_i)} \\
&= \frac{1}{n}\sum_{h\in\mathcal{H}}\delta\,Q(h)\sum_i\frac{\mathbb{P}_Q(x_i,h)}{\mathbb{P}_Q(x_i)\,Q(h)} \\
&= \frac{1}{n}\sum_{h\in\mathcal{H}}\delta\,Q(h)\sum_i\frac{\mathbb{P}_Q(h|x_i)}{Q(h)}
\end{aligned}$$

Et donc

$$\begin{aligned}\delta\ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log\sum_{h\in\mathcal{H}}\mathbb{P}_{\theta_I}(X|h)\,Q(h)]=0 &\Leftrightarrow \sum_{h\in\mathcal{H}}\delta\,Q(h)\sum_i\frac{\mathbb{P}_Q(h|x_i)}{Q(h)}=0\\ &\Leftrightarrow \sum_i\frac{\mathbb{P}_Q(h|x_i)}{Q(h)}\ \textit{indépendant de } h \quad (car\ \sum_{h\in\mathcal{H}}\delta\,Q(h)=0)\\ &\Leftrightarrow \exists C\in\mathbb{R},\forall h\in\mathcal{H},\,Q(h)=C\sum_i\mathbb{P}_Q(h|x_i)\end{aligned}$$

Puis $\displaystyle\sum_{h\in\mathcal{H}}Q(h)=1=C\sum_{h\in\mathcal{H}}\sum_{i=1}^{n}\mathbb{P}_Q(h|x_i)=Cn\Rightarrow C=\frac{1}{n}$

Enfin, par définition, on a $\mathbb{P}_Q(h|x_i)=\dfrac{\mathbb{P}_{\theta_I}(x|h)\,Q(h)}{\displaystyle\sum_{h'\in\mathcal{H}}\mathbb{P}_{\theta_I}(x|h')\,Q(h')}=Q^{cond}(h|x_i)$

$$\begin{aligned}\text{D'où } \forall h\in\mathcal{H},\,Q(h)&=\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}_Q(h|x_i)\\ &=\frac{1}{n}\sum_{i=1}^{n}Q^{cond}(h|x_i)\\ &=\sum_{x\in\mathcal{D}}Q^{cond}(h|x)\mathbb{P}_{\mathcal{D}}(x)\\ &=Q^{cond}_{\mathcal{D}}(h)\end{aligned}$$

Soit

$$\{\ Q,\,\delta\ \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log\sum_{h\in\mathcal{H}}\mathbb{P}_{\theta_I}(X|h)\,Q(h)]=0\}=\{Q,\,Q=Q^{cond}_{\mathcal{D}}=\phi_{\theta_I}(Q)\}$$

On a donc prouvé le *(2)*, ce qui achève la preuve du théorème 4. □

4 Relation avec certains modèles préexistants

Nous allons introduire dans cette partie deux modèles déjà existants qui vont s'avérer être des cas particuliers, dans le sens que nous allons voir ci-dessous, du modèle que nous venont de décrire dans la partie 3 précédente.

Reprenons la première définition énoncée dans cette partie, celle où la *B.O.L.L.* a été introduite :

$$(\hat{\theta_I}, \hat{q}) = arg \max_{(\theta_I, q)} \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\, q_{\mathcal{D}}(h)] \quad [déf]$$
$$\text{avec } q_{\mathcal{D}}(h) = \sum_{x \in \mathcal{D}} q(h|x) \mathbb{P}_{\mathcal{D}}(x)$$

Les modèles introduits ci-dessous seront alors des cas particuliers du modèle précédent dès lors que le max dans cette définition, que nous noterons dans la suite $[déf]$, ne sera plus pris sur l'ensemble des probabilités conditionnelles $q(h|x)$ (ce qui est bien utopiste et intractable en pratique), mais sur un ensemble plus restreint, inclus dans celui-ci.

Ainsi, la borne de performance obtenue grâce au théorème 1 sera de fait plus grande car le modèle est donc plus restrictif (nous avons déjà détaillé mathématiquement cet argument).

4.1 Stacked RBMs

Dans ce modèle, une couche est appelée une *RBM* (*Restricted Boltzmann Machine*) et on emplile (*stacked*) plusieurs *RBMs* pour former un modèle génératif profond appelé *SRBMs* (*Stacked Restricted Boltzmann Machines*).

Nous les présentons dans le cadre précédent, où $[déf]$ est alors remplacée par :

$$(\hat{\theta_I}, \hat{\mathbb{P}'}_{\mathcal{D},\theta_I}) = arg \max_{(\theta_I, \mathbb{P}'_{\mathcal{D},\theta_I})} \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\mathbb{P}'_{\mathcal{D},\theta_I}(h)]$$
$$\text{avec } \forall h \in \mathcal{H}, \mathbb{P}'_{\mathcal{D},\theta_I}(h) = \sum_{x\in\mathcal{D}} \mathbb{P}_{\theta_I}(h|x)\mathbb{P}_{\mathcal{D}}(x) = \mathbb{E}_{X\sim\mathbb{P}_{\mathcal{D}}}[\mathbb{P}_{\theta_I}(h|X)]$$

En notant $\mathbb{P}^{(1)}_{\theta_I}(X) = \sum_{h\in\mathcal{H}} \mathbb{P}^{(1)}_{\theta_I}(X|h)\mathbb{P}'_{\mathcal{D},\theta_I}(h) = \mathbb{E}_{H\sim\mathbb{P}'_{\mathcal{D},\theta_I}}[\mathbb{P}_{\theta_I}(X|H)]$,

On a donc

$$\hat{\theta_I} = arg \min_{\theta_I} D_{\mathrm{KL}}(\mathbb{P}_{\mathcal{D}}||\mathbb{P}^{(1)}_{\theta_I})$$

On ignore donc dans ce modèle la dépendance en θ_J de la couche supérieure. On optimise donc ici les paramètres couche par couche et on n'a plus de résultat similaire au théorème 1 sur l'optimisation globale du modèle profond.

Rque : Nous avons donc présenté les *RBMs* vis à vis du contexte et du modèle construit précédemment. Mais les *RBMs* existaient bien avant, elles sont très largement décrites dans la littérature et utilisées en pratique. En quelques mots, une *RBM* est un modèle de réseau de neurones à deux couches, comportant des unités visibles et des unités cachées, entraîné par un algorithme de backpropagation. Elle peut être vue comme un graphe biparti, non orienté et pondéré, où les deux parties sont justement appelées couche visible et couche cachée.

4.2 Auto-Encodeurs

Dans ce modèle, une couche est appelée auto-associateur et on empile plusieurs auto-associateurs pour former un modèle génératif profond appelé auto-encodeur. Tout comme les *SRBMs*, nous présentons ici les auto-encodeurs dans le cadre qui est le nôtre, de façon à montrer qu'il s'agit là encore d'un cas particulier du modèle construit en partie 3. [*déf*] est alors ici remplacée par :

$$(\hat{\theta_I}, \hat{q}_\xi) = arg \max_{(\theta_I, q_\xi)} \mathbb{E}_{X\sim\mathbb{P}_\mathcal{D}}[log \sum_{h\in\mathcal{H}} \mathbb{P}_{\theta_I}(X|h) q_\mathcal{D}(h)]$$
$$\text{avec } q_\mathcal{D}(h) = \sum_{x\in\mathcal{D}} q_\xi(h|x)\mathbb{P}_\mathcal{D}(x)$$

Et en notant pour $(x,h) \in \mathcal{D}\times\mathcal{H}, x = (x_1, ..., x_p)$ et $h=(h_1, ..., h_{p'})$

Alors

$$\begin{cases} \forall (x,h) \in \mathcal{D}\times\mathcal{H}, q_\xi(h|x) = \prod_{j=1}^{p'} q_\xi(h_j|x) \qquad (i) \\ \forall j \in [\![1, p']\!] \;\; q_\xi(h_j|x) = sigm(\sum_{i=1}^{p} x_i w_{ij} + b_j) \\ \forall x \in \mathbb{R}, sigm(x) = \frac{1}{1+e^{-x}} \text{ est la fonction sigmoïde, et } \xi = \{W = (w_{ij}), b = (b_j)\} \end{cases}$$

(i) $\Rightarrow$ les h_j pour j$\in [\![1, p']\!]$ sont indépendants conditionnellement à x.

On fait de plus en général l'hypothèse suivante dans le modèle des auto-encodeurs :

Pour toute probabilité conditionnelle q,

$$\forall (x, x') \in \mathcal{D}^2, (x \neq x' \Rightarrow \mathbb{P}_{\theta_I}(x|h) q(h|x') = 0)$$

Ce qui signifie que deux observations distinctes de la variable observée X ne peuvent pas résulter d'une même réalisation de la variable latente H.

Rques :

De la même façon que pour les *RBMs*, les auto-encodeurs ont été introduits bien avant ce formalisme.
Un auto-associateur est en fait un réseau de neurones entraîné par un algorithme de backpropagation à reproduire sa propre entrée, en passant par une couche cachée qui sert de représentation intermédiaire.

On peut ainsi le voir comme un réseau à trois couches : $x \mapsto h^{(1)} \mapsto x$ qui donne deux distributions conditionnelles $\mathbb{P}(h^{(1)}|x)$ et $\mathbb{P}(x|h^{(1)})$.
L'auto-associateur formant la couche supérieure sera alors entraîné de la même façon, suivant $h^{(1)} \mapsto h^{(2)} \mapsto h^{(1)}$.
La dernière couche $h^{(k_{max})}$ est quant à elle souvent entraînée à l'aide d'une *RBM*.

Le critère pour entraîner les auto-encodeur est alors appelé assez naturellement "l'erreur de reconstruction", et nous venons de voir que la maximisation de ce critère peut être considéré comme une maximisation d'une borne inférieure de la borne supérieure de la *BLM*, où chaque exemple correspond à une seule représentation cachée.

4.3 Fine-tuning

Reprenons de nouveau la définition suivante.

$$(\hat{\theta_I}, \hat{q}) = arg \max_{(\theta_I, q)} \mathbb{E}_{X \sim \mathbb{P}_{\mathcal{D}}}[log \sum_{h \in \mathcal{H}} \mathbb{P}_{\theta_I}(X|h)\, q_{\mathcal{D}}(h)] \quad [déf\,]$$

Ce problème d'optimisation ne pourra pas être résolu de façon parfaite en pratique. D'une part, et nous venons de le voir, parce que le max ne portera pas sur l'ensemble des probabilités conditionnelles q ; et d'autre part car le

critère utilisé dans l'entraînement du modèle génératif profond est par définition optimiste : chaque couche est entraînée en faisant la supposition que la couche supèrieure sera capable d'atteindre exactement la *BLM*, ce qui en pratique n'est bien sûr jamais le cas.

Les paramètres globaux obtenus après l'entraînement de toutes les couches du modèle ne sont donc, en pratique, pas optimaux. Pour se rapprocher davantage de cet optimum global, on pourra alors faire un entraînement supervisé classique, dit de peaufinage (*fine-tuning*).

C'est d'ailleurs là tout l'intérêt de l'entraînement couche par couche décrit dans ce travail. En effet, entraîner par des méthodes classiques un modèle génératif profond s'avère très difficile d'une part pour des raisons computationnelles (nous l'avons vu à la partie 2), et d'autre part car l'entraînement est "coincé" dans de nombreux minima locaux, sans entrer ici dans les détails.

Faire un pré-entraînement couche par couche comme celui décrit à la section 3 permet alors d'initialiser un entraînement supervisé dans une "bonne région" de l'espace des paramètres, soit proche de l'optimum global, à partir de laquelle une descente de gradient classique trouve une meilleure solution que celle trouvée à partir d'une initialisation aléatoire, et plus rapidement.

Nous passons maintenant à la partie expérimentation du projet, à savoir la vérification empirique des résultats prouvés dans les partie précédentes.

5 Applications et expérimentations

Les résultats primordiaux qu'implique le théorème 1 énoncé à la partie 3 sont d'une part l'importance d'avoir des paramètres différents pour les parties génératives et l'inférence du modèle, et d'autre part de permettre à l'inférence du modèle d'être aussi riche que possible, c'est à dire que la probabilité conditionnelle q sous laquelle on optimise dans $[déf]$ vive dans un espace aussi vaste et complexe que possible.

C'est précisément ce que nous tenterons de vérifier expérimentalement dans cette partie.

Commençons par présenter notre façon de procéder.

5.1 Présentation du dispositif

L'idée va ici être de comparer la performance de deux modèles profonds après un pré-entraînement tel que celui décrit dans la partie théorique de ce rapport.
Le premier modèle sera un auto-encodeur profond, dit "classique".
Le second modèle se basera sur le premier, mais en le modifiant pour que la partie inférence du modèle soit plus riche, comme nous allons le voir.
Les deux modèles auront donc la même "partie générative", et c'est bien la procédure d'entraînement que nous comparerons.

Présentons tout d'abord la base de données qui sera utilisée en pratique.

Nous prendrons dans cette partie $\mathcal{D} = [\![0, 255]\!]^{28\times 28}$.

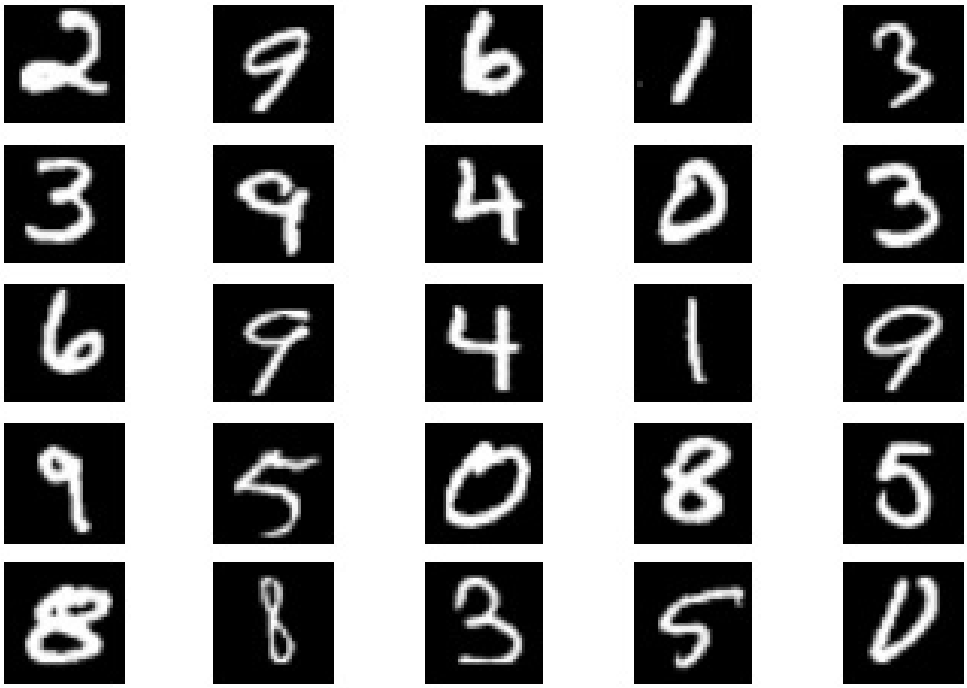
Les $x \in \mathcal{D}$ représentent alors ici des images carrées de 28 pixels de côté. Chaque pixel prend une valeur dans $[\![0, 255]\!]$ qui correspond à son niveau de gris, 0 étant le noir et 255 le blanc.

Ces images seront celles de la base de donnée MNIST qui est composée de chiffres manuscrits, et nous disposerons d'un échantillon d'apprentissage de 60 000 images et d'un échantillon de test de 10 000 images.

Nous disposerons également des labels correspondant à chaque image et qui sont simplement les chiffres correspondants, donc ici $\mathcal{Y} = [\![0, 9]\!]$.

Cette base de données est gratuite et disponible à cette adresse *http ://yann.lecun.com/exdb/mnist/*.

Voici quelques exemples de ce à quoi ressemblent ces images :

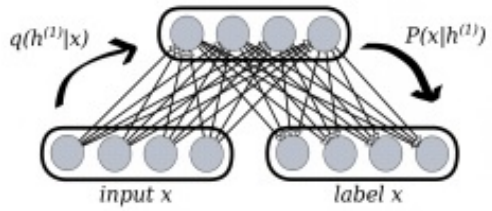


J'ai alors travaillé sous Matlab en partant des codes proposés par Ruslan Salakhutdinov et Geoff Hinton pour entraîner un auto-encodeur profond, disponibles gratuitement et en libre accés à cette adresse :
http ://www.cs.toronto.edu/ hinton/MatlabForSciencePaper.html .

Ces chercheurs ont donc implémenté un auto-encodeur, en empilant quatre auto-associateurs.
On a ainsi affaire à un modèle génératif profond à quatre couches cachées, soit un modèle de profondeur quatre, et qui constituera donc notre premier modèle.

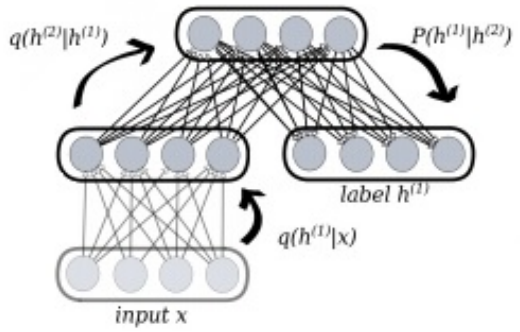
Sans revenir sur son formalisme théorique, mais plutôt pour compléter la partie 4.2 et pour mieux comprendre la suite, nous allons donner une représentation graphique de l'entraînement d'un auto-encodeur (à deux couches cachées seulement pour une meilleure lisibilité).

On commence alors à entraîner la première couche à reproduire les données elles-mêmes :



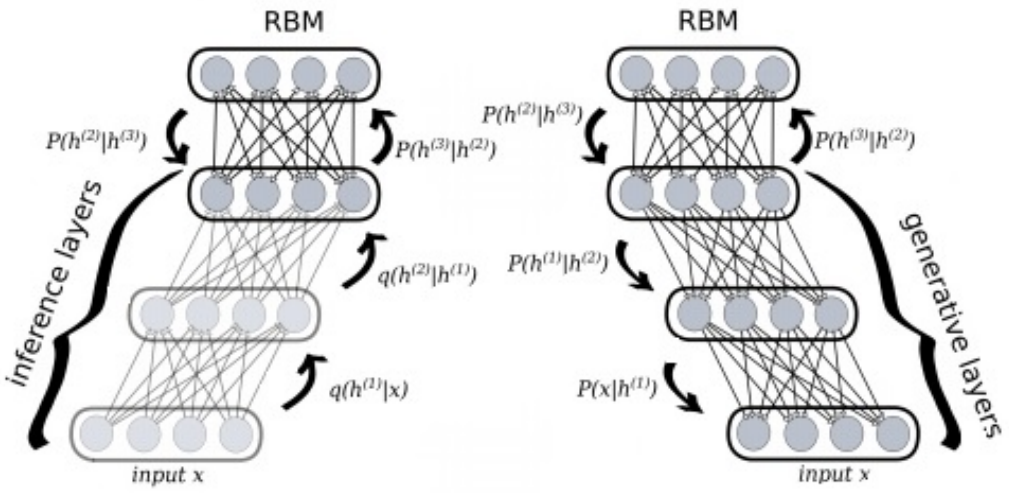
Sans plus entrer dans les détails, on utilise pour cela un algorithme de backpropagation du gradient de l'erreur pour optimiser la version de la *BLM* explicitée à la partie 4.2.

On entraîne ensuite la couche supérieure de la même façon, mais en prenant cette fois en entrée la première couche cachée :



On utilise alors une *RBM* pour entraîner la dernière couche cachée. En effet, en empilant des auto-associateurs, il se pose toujours la question de l'entraînement de la couche la plus haute. Une *RBM* est souvent utilisée.

On distingue alors bien la partie inférence et la partie générative du modèle à l'aide du schéma suivant :



Le code que j'ai récupéré entraîne un auto-encodeur comportant donc non pas deux, mais quatre couches cachées, mais il procède exactement de cette façon.

Pour être un peu plus précis, voici les différentes étapes de traitement (quelque peu simplifiées) de l'algorithme :

- On commence par convertir les 60 000 images de l'échantillon d'entraînement ainsi que les 10 000 images de l'échantillon de test en données exploitables par Matlab, à savoir en vecteurs à $28 \times 28 = 784$ composantes correspondant aux pixels de chaque image. On les place ensuite dans deux matrices, une comportant les données d'entraînement et l'autre les données de test.

- On normalise les valeurs en divisant tout par 255. On obtient donc deux matrices E=(e_{ij}) et T=(t_{ij}) avec $\forall i \in [\![1, 784]\!], \forall j \in [\![1, 60000]\!], (e_{ij}, t_{ij}) \in [0, 1]^2$ (E pour entraînement et T pour test).

- On lance l'entraînement de l'auto-encodeur à quatre couches cachées. On apprend donc les poids du réseau de neurones multi-couches.

On obtient une fois les poids appris et lorsqu'on présente la matrice de la base de test au système une matrice $\hat{T} = (\hat{t}_{ij})$ et on mesure l'erreur de reconstruction de la façon suivante :

$$erreur = \frac{1}{784 \times 60000} \sum_{i,j} (t_{ij} - \hat{t}_{ij})^2.$$

Nous calculerons alors l'erreur sur la matrice de test après avoir appris le modèle avec la matrice d'entraînement.

Rques :

1) En réalité, l'algorithme est plus compliqué car il sélectionne aléatoirement les images par lots de plus petite dimension que 60000 (le passage aléatoire des données est une nécéssité pour l'algorithme de backpropagation), et pour ensuite moyenner l'erreur et en donner une grandeur plus fiable.

2) On ne fera donc pas ici de *fine-tuning*, bien que le code de Salakhutdinov et Hinton en propose un. En effet, le temps de calcul devient vite long avec un *fine-tuning* si l'on souhaite faire de nombreuses comparaisons. Et surtout, ce n'est pas necessaire pour ce que l'on cherche à vérifier ici.

Explicitons maintenant le second modèle, qui se base sur l'auto-encodeur qui vient d'être décrit, mais pour lequel j'ai modifié le code de façon à ce que la partie inférence du modèle soit plus riche.

Je me suis alors basé sur l'expérience de Ludovic Arnold et Yann Ollivier, et sur ce qu'ils ont appelé les *AERIes* (*Auto-Encoders with Rich Inference*). Ils ont alors utilisé le fait que la complexité de l'inférence du modèle, à travers $q(h|x)$, pouvait être élevée sans risquer un sur-apprentissage puisque q ne fait pas partie du modèle génératif final, mais est seulement comme un "outil" pour l'optimisation des paramètres θ du modèle génératif.

Les conséquences d'une partie inférence plus riche ne peuvent alors être, semble-t-il, que positives.

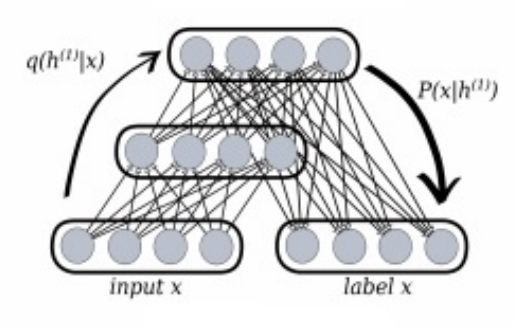
L'idée est alors d'utiliser des auto-associateurs à non pas une, mais deux couches cachées :

$$x \mapsto h' \mapsto h \mapsto x$$

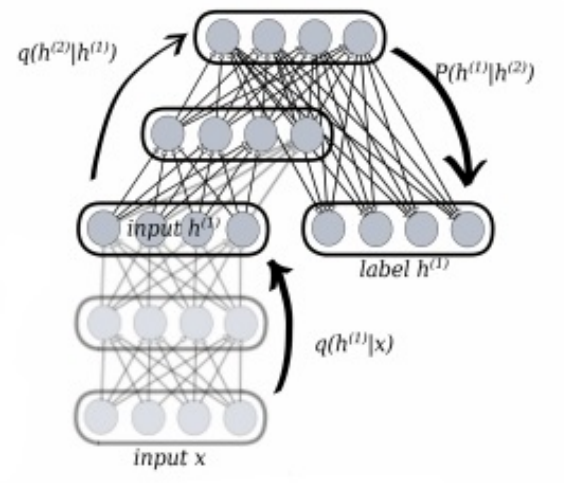
La partie générative de ce modèle ($\mathbb{P}_{\theta_I}(x|h)$) sera donc bien équivalente à celle du premier modèle, mais la partie inférence sera plus riche.

De la même façon que pour le modèle précédent et pour mieux saisir l'idée, voici une représentation graphique de l'entraînement du second modèle (toujours pour deux couches cachées au lieu de quatre).

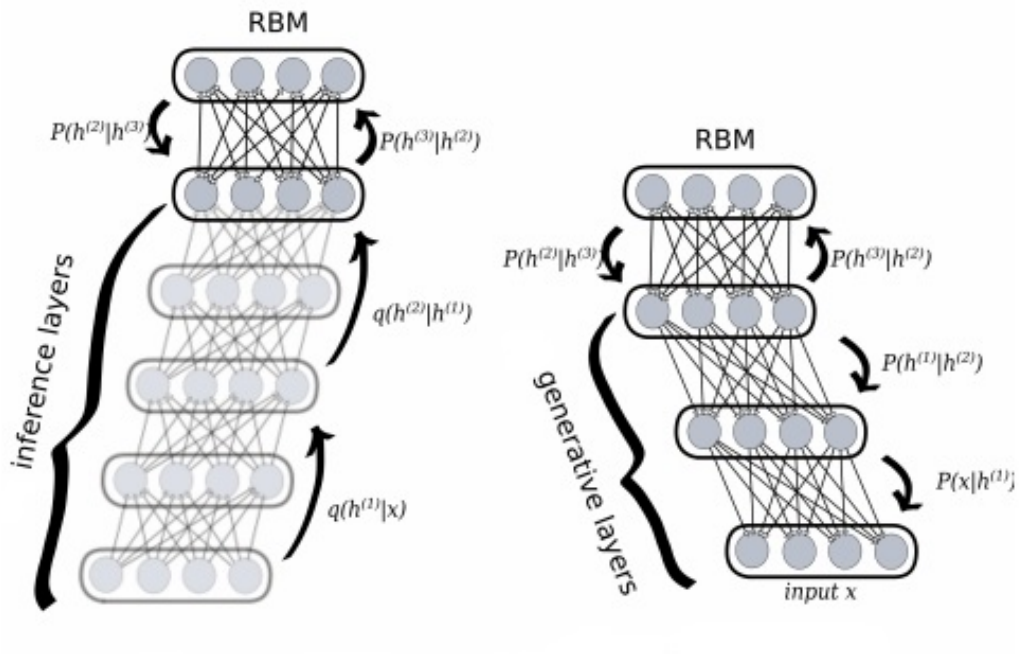
Entraînement de la première couche :



Entraînement de la seconde couche :



Entraînement de la dernière couche à l'aide d'une *RBM*.



Avant de passer aux résultats, fixons les paramètres choisis pour les différents modèles.

5.2 Paramètres choisis

Tout d'abord, voici les tableaux récapitulatifs des différentes tailles (nombre de neurones) des couches testées pour chaque modèle, que j'ai classé pas "cas".

cas: / tailles:	1	2	3	4	5
Modèle 1					
couche cachée 1	100	200	300	400	500
couche cachée 2	50	100	150	200	250
couche cachée 3	25	50	75	100	125
couche cachée 4	10	10	10	10	10
Modèle 2					
couche cachée 1	100	200	300	400	500
couche cachée 1'	75	150	225	300	375
couche cachée 2	50	100	150	200	250
couche cachée 2'	37	75	112	150	187
couche cachée 3	25	50	75	100	125
couche cachée 3'	18	37	56	75	93
couche cachée 4	10	10	10	10	10

cas: / tailles:	6	7	8	9	10
Modèle 1					
couche cachée 1	600	700	800	900	1000
couche cachée 2	300	350	400	450	500
couche cachée 3	150	175	200	225	250
couche cachée 4	10	10	10	10	10
Modèle 2					
couche cachée 1	600	700	800	900	1000
couche cachée 1'	450	525	600	675	750
couche cachée 2	300	350	400	450	500
couche cachée 2'	225	262	300	337	375
couche cachée 3	150	175	200	225	250
couche cachée 3'	112	131	150	168	187
couche cachée 4	20	20	20	20	20

Rque : Précisons qu'en notant n_i la taille d'une couche du premier modèle et n_{i+1} la taille de la couche supèrieure, on choisit d'ajouter simplement une couche intermédiaire pour le second modèle de taille $n'_i = E(\frac{n_i+n_{i+1}}{2})$ où E est la fonction partie entière, de façon a pouvoir comparer la performance des deux modèles.

Ensuite, voici les paramètres choisis dans l'algorithme de backpropagation lors du préapprentissage de chacunes des couches.

- nombre d'itération : N=10 pour les 5 premiers cas, N=20 pour les 3 suivants, et N=50 pour les deux derniers.
- taux d'apprentissage : $\epsilon = 0.1$
- taille des lots : 200
- nombre de lots : 120 pour l'apprentissage et 20 pour le test.

Rque : On comprend intuitivement que le nombre d'itérations nécessaires au modèle dans l'algorithme de backpropagation lors de l'entraînement de chacunes des couches doit augmenter lorsque la taille des couches augmente. D'où le choix pour N, avec la volontée de ne pas changer sa valeur à chaque cas pour observer la conséquence de l'augmentation de la taille des couches pour un N constant.

Passons enfin aux résultats obtenus pour la comparaison en performance des deux modèles que nous venons de décrire, en utilisant donc la base de données MNIST.

5.3 Résultats et interprétations

Les résultats obtenus dans les différents cas sont résumés dans le tableau suivant :

cas: / erreur de test:	1	2	3	4	5	6	7	8	9	10
Modèle 1	36,18	34,204	33,91	33,242	31,426	24,873	22,147	20,076	13,782	12,34
Modèle 2	33,271	32,13	30,645	28,584	26,91	22,355	19,289	17,105	11,924	10,856

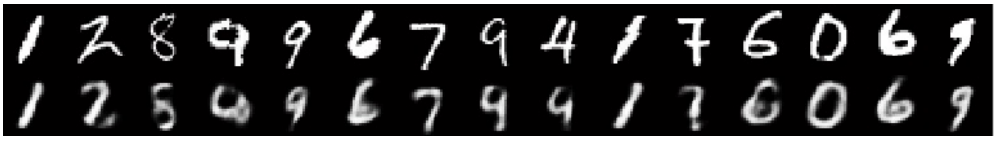
Rque : Les erreurs calculées ont été moyennées sur 20 expériences dans chacun des cas.

Pour bien visualiser ce que l'algorithme a fait, voici maintenant pour les cas 1 et 10 uniquement 15 images parmis les 10 000 de test (prises aléatoirement) avec en dessous de chacune, l'image reconstruite par l'algorithme après le pré-apprentissage.

$Cas1$:

Modèle 1 :

Modèle 2 :



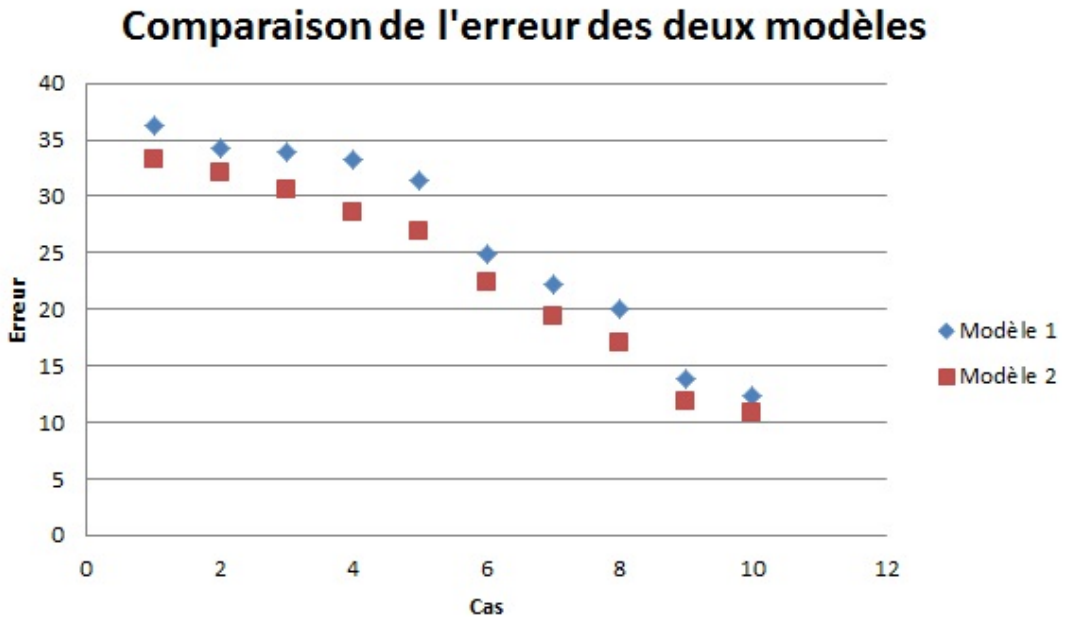
$Cas10$:

Modèle 1 :

Modèle 2 :

On contacte bien visuellement les différences entre les deux cas d'abord, et une performance qui paraît en effet meilleure pour le modèle 2.

Voici maintenant le graphe de performance des deux modèles correspondant aux réultats du tableau précédent.



On constate en effet que le second modèle a une meilleure performance, peu importe les hyper-paramètres choisis.

Cela confirme bien ce à quoi on s'attendait.

Remarquons enfin que l'entraînement non-supervisé a l'avantage d'être très peu couteux, car disposer de larges bases de données avec les labels correspondants nécessite souvent un travail manuel, long, pénible et qui coûte cher.

6 Conclusion

Après avoir étudié le principe de l'algorithme de backpropagation au tout début du projet, je l'ai implémenté sous R pour le voir opérer en pratique. J'ai ensuite étudié le modèle général des réseaux de neurones, puis codé depuis zéro un perceptron multi-couche, toujours sous R, pour mieux appréhender ces modèles profonds qui étaient alors nouveaux pour moi. J'étais dès lors armé pour me concentrer sur le point central de mon projet de fin d'étude, qui a donc été l'étude de cet article récent.

Un travail théorique important a été nécessaire pour bien comprendre les concepts nouveaux introduits par Ludovic Arnold et Yann Ollivier, et également pour comprendre tous les concepts et modèles associés aux modèles génératifs profonds de façon générale, afin de ne pas être noyé sous une quantité importante de nouveautés, ce qui n'a pas été facile au début.

Cela m'a néanmoins permi de percevoir tout à fait le contexte exact dans lequel est paru cet article, ainsi que son apport réel dans la recherche dans ce domaine.
Ce qui a d'ailleurs, à mon sens, été très important pour comprendre en profondeur l'article et pour prendre de la hauteur quant aux résultats qui en découlent.

Ainsi, j'ai entièrement repris cet article qui donne une nouvelle approche d'apprentissage couche par couche dans les modèles génératifs profonds, basé sur une hypothèse optimiste quant au succés des couches supérieures. Car en supposant que cet optimisme se vérifie et qu'un bon modèle est trouvé pour les couches hautes, alors les paramètres du modèle final seront proches des paramètres optimaux. Et lorsque l'optimisme ne se vérifie pas, l'article donne une borne de la perte de performance.
Cette nouvelle façon de voir les choses souligne l'importance d'utiliser une partie inférence du modèle plus riche que la partie générative.

La partie vérification empirique du projet a été également très intéressante car d'une part, elle a nécessité de bien comprendre les codes utilisés pour pouvoir ensuite les modifier, et d'autre part il est toujours agréable de

vérifier que les résultats théoriques fonctionnent bien en pratique.
A noter que je n'ai pas comparé expérimentalement le modèle auto-encodeur à 4 couches avec un modèle non profond, ce qui aurait été la première chose à faire pour justifier la pertinence de l'expérimentation, tout simplement parce que cela a déjà été fait longuement, comme cela est rappelé notamment dans l'introduction du rapport.

Il serait alors intéressant d'incorporer à la tâche d'apprentissage l'optimisation des hyper-paramètres tels que la taille des couches cachées ou la profondeur du réseau. Il faudrait alors trouver un critère non-supervisé pour faire cela. La *BLM* pourrait d'ailleurs être utilisée à cet effet, chose évoquée par l'auteur dans l'article, mais des confirmations empiriques restent à donner. En effet d'autres critères sont plus en vogue sur cette question au coeur de la recherche actuelle concernant les modèles génératifs profonds, comme l'erreur de reconstruction ou l'énergie induite.

Enfin, je terminerai en insistant sur le plaisir que j'ai eu à mener ce projet à bien, dans un domaine qui me plaît particulièrement. Ce projet s'inscrit d'ailleurs tout à fait dans la continuité de mon parcours scolaire puisque mon stage de fin d'étude portera sur des problèmatiques d'apprentissage statistique, et le deep learning fait beaucoup parler à l'heure actuelle. Je suis désormais plus familier avec ce sous-domaine du data-mining, chose qui me servira sans aucun doute, et ce projet m'aura donné envie de continuer la recherche dans ce domaine.