Deep Belief Networks

I. Rappels : estimation de paramètres dans des modèles à variables latentes

Soit X v.a observée $\in \mathbb{R}$

H v.a cachée $\in \mathbb{R}$

- On se donne $p_\theta(x, h)$ dépend d'un paramètre θ qu'on aimerait apprendre à partir de $(x^{(1)}, \dots x^{(n)})$ iid

- $$\hat{\theta}_{MV} = \underset{\theta}{\text{Arg max}} \prod_{i=1}^{n} p_\theta(x^{(i)}) = \underset{\theta}{\text{Arg max}}\, p_\theta(x^{(1)}, \dots, x^{(n)})$$

$$= \underset{\theta}{\text{Arg max}} \sum_{i=1}^{n} \log p_\theta(x^{(i)})$$

Avec $p_\theta(x) = \int p_\theta(x, h)\, dh$

est l'estimateur du MV . bonnes propriétés asymptotiques

. minimise la DKL $\int p(x) \log \frac{p(x)}{p_\theta(x)} dx$

où $p(x)$ est la vraie loi des observations

ie $\hat{\theta}_{MV} \xrightarrow[n \to +\infty]{\text{proba}}$ vers $\underset{\theta}{\text{Argmin}}\, DKL(p, p_\theta)$

$\Rightarrow$ difficile à appliquer si $\int p_\theta(x, h)\, dh$ inconnue

- Algorithme EM : itératif (pour ! x)

Partant de $\theta^{(i)}$:

E : $E_{\theta^{(i)}} [\log p_\theta(x, h) \mid x] = Q(\theta^{(i)}, \theta)$

M : $\theta^{(i+1)} = \underset{\theta}{\text{Arg max}} \; Q(\theta^{(i)}, \theta)$

Rq : E : $= \int \log p_\theta(x, h) \times p_{\theta^{(i)}}(h \mid x) \, dh$

[Th de Transfert $E(f(X, Y)) = \iint f(x, y) \, p(x, y) \, dx \, dy$]

On a tout de même besoin de la loi à posteriori : $p_{\theta^{(i)}}(h \mid x) = \dfrac{p_{\theta^{(i)}}(x, h)}{p_{\theta^{(i)}}(x)}$

Vocab :
$p(h) \to$ prior
$p(x \mid h) \to$ likelihood conditionnelle
$p(x) \to$ likelihood
$p(h \mid x) \to$ posterior

Problème : on a besoin de $p(h \mid x)$ pas évident à calculer.

- Inférence Bayésienne Variationnelle. (IBV)

Soit $q_\xi(h|x)$ une loi quelconque donnée de paramètre ξ.

$$DKL(q_\xi(h|x), p_\theta(h|x)) = \int \log\left(\frac{q_\xi(h|x)}{p_\theta(h|x)}\right) q_\xi(h|x)\, dh$$

$$= \int \log\left(\frac{q_\xi(h|x)}{p_\theta(h,x)}\right) q_\xi(h|x) + \int \log(p_\theta(x))\, q_\xi(h|x)\, dh$$

$$= \int \log\left(\frac{q_\xi(h|x)}{p_\theta(h,x)}\right) q_\xi(h|x) + \log p_\theta(x)$$

$$\Rightarrow \quad \log p_\theta(x) = -\int \log\left(\frac{q_\xi(h|x)}{p_\theta(h,x)}\right) q_\xi(h|x) + \underbrace{DKL(q_\xi(h|x), p_\theta)}_{\geq 0}$$

$$\Rightarrow \quad \log p_\theta(x) \geq -\int \log\left(\frac{q_\xi(h|x)}{p_\theta(h|x)}\right) q_\xi(h|x) = \mathrm{elbo}(\theta, \xi)$$

Rq : à $\theta = \theta^{(i)}$ fixé

Alors $q_\xi(h|x) = p_{\theta^{(i)}}(h|x)$ maximise $\mathrm{elbo}(\theta^{(i)}, \xi)$ /t à ξ

à $q_\xi(h|x) = p_{\theta^{(i)}}(h|x)$

$$\mathrm{ELBO}(\theta) = \int \log p_\theta(h,x)\, p_{\theta^{(i)}}(h|x)$$

À maximiser /t à θ

On retrouve EM.

• On se donne une classe de q_ξ paramétrée par ξ (4)
en faisant en sorte $q_\xi(h|x)$ "ne soit pas trop éloignée de" $p_\theta(h|x)$ et on optimise
$ELBO(\theta, \xi)$ /r à (θ, ξ) (gradient)
de manière à pousser $\log p_\theta(x)$ vers la gauche.

II Deep. Belief Network

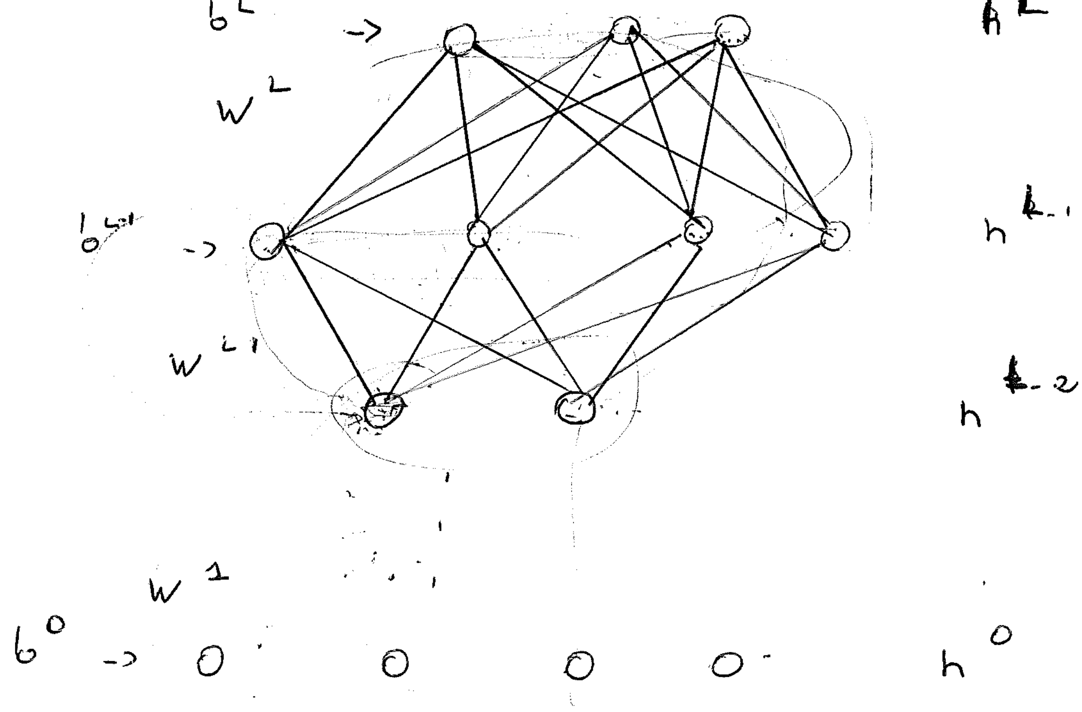
Soit ~~$V = H^0$~~ $V = H^0 = (V_1, \dots, V_p) \in \{0,1\}^p$
$H^p = (H_1^p, \dots H_{q_p}^p) \in \{0,1\}^{q_p}$
pour $p = 1 : L$

Déf : $p(h^0, h^1, \dots h^L) = p(h^{L-1}, h^L) \prod_{i=1}^{L-1} p(h^{i-1} | h^i)$

où i) $p(h^{L-1}, h^L)$ est un RBM
$= \frac{1}{Z} e^{-E(h^{L-1}, h^L)}$

ii) $p(h^{i-1} | h^i) = \prod_{j=1}^{q_{i-1}} p(h_j^{i-1} | h^i)$

iii) $p(h_j^{i-1} = 1 | h^i) = \text{sign}(b_j^{i-1} + \sum_k w_{jk}^i h_k^i)$



On a fabriqué un modèle génératif

$$p(v) = p(h^0) = \sum_{h_1, \dots h^L} p(h^0, h^1, \dots h^L)$$

et qui généralise le RBM (on prend $L = 1$ $\Rightarrow$ RBM)

<u>Rq</u>: $p(h^{l-1} \mid h^l) = p_{RBM}(h^{l-1} \mid h^l)$

mais (h^{l-1}, h^l) n'est pas un RBM (sauf pour $l = L$)

car $p(h^l \mid h^{l-1}) \neq p_{RBM}(h^l \mid h^{l-1})$

- Estimer les paramètres d'un DBN à partir $(x^{(1)}, \dots x^n)$ iid

i) Estimateur du MV : $p_\theta(x) = \sum_{h^1,\dots h^L} p_\theta(x, h^1, \dots h^L)$

$\theta = (b^i, W^i)_{i=1:L} + b^0$

$\Rightarrow$ Somme sur $h^1, \dots h^L$ ...

ii) EM $\Rightarrow$ $p(h^1, \dots, h^L \mid x) = \dfrac{p(x, \dots h^L)}{\sum_{h^1 \dots h^L} p(x, \dots, h^L)}$

iii) Variational inference.

Idée : DBN comme un empilement de RBM.

- $q(h_j^1 \mid x) = \text{sign}\left(\tilde{b}_j^1 + \sum_i w_{ij}^1 x_i\right)$
- $q(h_j^p \mid x) = q(h_j^p \mid h^{p-1}) = \text{Sign}\left(\tilde{b}_j^p + \sum_i w_{ij}^p h_i^{p-1}\right)$

où h^{p-1} est $h^{p-1} = f(x)$ où on a appliqué les transformations précédentes.

Une fois cette loi q variationnelle donnée, on mq l'optimisation de la ELBO se déduit :

Algorithme Greedy ~~LAyer wise~~ procédure

x : utiliser CD-1 pour apprendre $b^0, w^1, \tilde{b}^1$

calculer la sortée $\tilde{x}$ de x avec ce RBM

$\tilde{x}$ utiliser CD 1 pour apprendre $b^1, w^2, \tilde{b}^2$

⋮

(Learning Deep Architectures for AI.)

Pseudo - code x, DBN -> liste de RBM

```
train_DBN(                    )
for i = 1 : nb_couches
    DBN[i] = train_RBM(x, ......,)
    x = entree_sortie (x, DBN[i])
fin
```