Cours n° 3
Rappel DNN

I. Quelques résultats

1. Origine de la sigmoïde :

$$\begin{cases} P(X=1) = p \\ P(X=0) = 1-p \end{cases}$$

$$\begin{cases} p(y|x=1) = \mathcal{N}(y ; m_1 ; \sigma^2) \\ p(y|x=0) = \mathcal{N}(y ; m_2 ; \sigma^2) \end{cases}$$

$$\Rightarrow \quad p(X=1|y) = \frac{p(X=1) \cdot p(y|x=1)}{p(X=1)\,p(y|x=1) + P(X=0)\,p(y|x=0)}$$

$$\boxed{\ln \frac{p(X=1|y)}{1-p(X=1|y)} = ax+b}$$

$$= \frac{p \cdot \mathcal{N}(y ; m_1 ; \sigma^2)}{p\,\mathcal{N}(y ; m_1 ; \sigma^2) + (1-p)\,\mathcal{N}(y ; m_2 ; \sigma^2)}$$

$$= \frac{1}{1 + \frac{(1-p)}{p} \times e^{-\frac{1}{2\sigma^2}\left(y^2 - 2m_2 y + m_2^2 - y^2 + 2m_1 y - m_1^2\right)}}$$

$$= \frac{1}{1 + e^{\ln\left(\frac{1-p}{p}\right) - (ay+b)}} = \frac{1}{1 + e^{-(a'y+b')}}$$

$$= \mathrm{Sign}(a'y+b)$$

2. Dérivées

2

- $g(x) = \text{sign}(x)$

$$\frac{dg}{dx} = g(x)\,(1 - g(x))$$

- $f(z_1, \dots, z_K)$; $h(x) = f(g_1(x), \dots, g_K(x))$

$$\frac{dh}{dx} = \sum_{i=1}^{K} \frac{dg_i}{dx} \times \frac{df}{dg_i}$$

- $$f_i(z_1, \dots z_K) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} = \text{softmax}(z_1, \dots z$$

$$* \quad \frac{df_i}{dz_i} = \frac{e^{z_i} \times \sum_{k=1}^{K} e^{z_k} - e^{z_i} \times e^{z_i}}{\sum_k e^{z_k} \times \sum_k e^{z_k}}$$

$$= f_i - f_i^2 = f_i\,(1 - f_i)$$

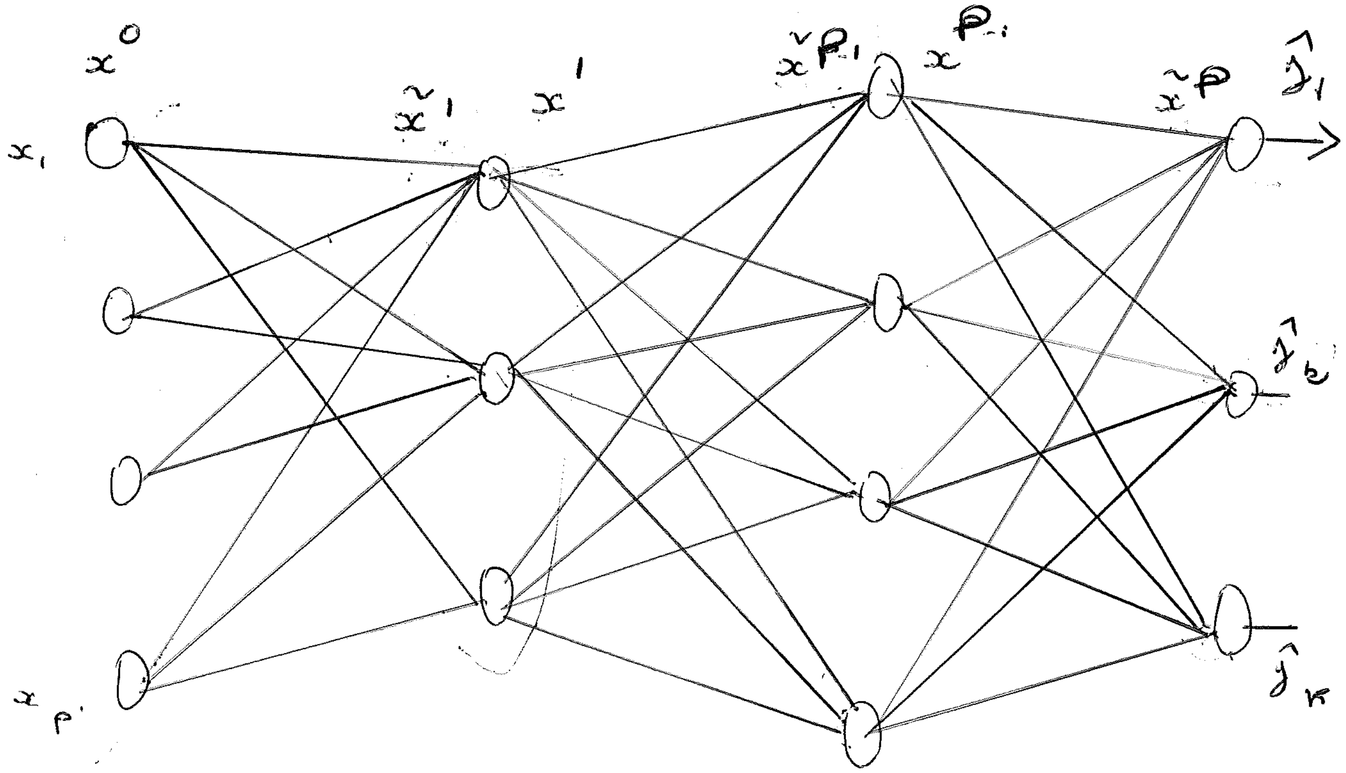
$$\frac{df_i}{dz_j} = \frac{-\,e^{z_i}\,e^{z_j}}{\sum_{k=1} e^{z_k} \times \sum_{k=1} e^{z_k}} = -f_i \times f_j$$

B. Réseau de neurones pour classif multiclasses

<u>objectif</u> : modéliser $P(Y = k \mid X = x) = f_k(x)$

$$\Rightarrow \quad f_k(x) \longrightarrow f_{k,\theta}(x) \quad tq \quad \sum_{k=1}^{K} f_{k,\theta}(x) = 1$$

Réseau de neurones



$$\tilde{x}_j^P = w_j^P \cdot x^{P-1} + b_j^P \qquad w_j^P = \text{vecteur de poids}$$

$$x_j^P = g(\tilde{x}_j^P) \quad \text{avec} \quad g(x) = \text{sign}(x)$$

$$\hat{y}_k = \text{softmax}_k(\tilde{x}_1^P, \dots, \tilde{x}_K^P)$$

. objectif : estimer $\theta = (W^p, b^p)$ pour $p = 1 : P$.

à partir de $\begin{pmatrix} x^{(1)}, \dots, x^{(n)} \\ y^{(1)}, \dots, y^{(n)} \end{pmatrix}$

. Démarche : . MAximisons la vraisemblance.

Pour une donnée (x, y) : $p(x, y) = p(x) . p(y|x)$

$$\log p(x, y) = \log p(x) + \log p_\theta(y|x)$$

$\Rightarrow$ maximisons $\sum_{i=1}^{n} \log p_\theta(y^{(i)} | x^{(i)})$

calculons $\nabla_\theta \log p(y|x)$. pour ? (x, y)

$y = i \iff y = [0, 0, 1, \dots 0]$

$\uparrow$ i ème composante

$$p_\theta(y|x) = \prod_{k=1}^{K} p_\theta(y = k | x)^{y_k}$$

$$\Rightarrow \log p_\theta(y|x) = \sum_{k=1}^{K} y_k \log p_\theta(y = k | x)$$

$$= \sum_{k=1}^{K} y_k \log \hat{y}_k$$

$$= -L(\hat{y}_1, \dots, \hat{y}_K)$$

Minimisons $\frac{\partial}{\partial \dots} - \sum_{k=1}^{K} y_k \log \hat{y}_k = \mathcal{L}$ /r à θ.

$$\frac{d\mathcal{L}}{dw_j^p} = \frac{d\mathcal{L}}{d\tilde{x}_j^p} \times \underbrace{\frac{d\tilde{x}_j^p}{dw_j^p}}_{\text{trivial}}$$

$$\frac{d\mathcal{L}}{d\tilde{x}_j^p} = \sum_{k=1}^{K} -\frac{y_k}{\hat{y}_k} \times \frac{d\hat{y}_k}{d\tilde{x}_j^p}$$

AVEC $\hat{y}_k = \text{softmax}_k(\tilde{x}_1^p, \dots \tilde{x}_K^p)$

$$= \sum_{k \neq j} -\frac{y_k}{\hat{y}_k} \cdot \frac{d\hat{y}_k}{d\tilde{x}_j^p} - \frac{y_j}{\hat{y}_j} \frac{d\hat{y}_j}{d\tilde{x}_j^p}$$

$$= \sum_{k \neq j} + \frac{y_k}{\cancel{\hat{y}_k}} \times \cancel{\hat{y}_k} \times \hat{y}_j - \frac{y_j}{\cancel{\hat{y}_j}} \cancel{\hat{y}_j}(1 - \hat{y}_j)$$

$$= \underbrace{\sum_{k \neq j} y_k \hat{y}_j} - y_j + \underbrace{y_j \hat{y}_j}$$

$$= \sum_{k=1}^{K} y_k \hat{y}_j - y_j = \hat{y}_j \sum_{k=1}^{K} y_k - y_j$$

$$\frac{d\mathcal{L}}{d\tilde{x}_j^p} = \hat{y}_j - y_j = c_j^p$$

$$\Rightarrow \quad \frac{d\mathcal{L}}{dw_j^p} = c_j^p \times x^{p-1}$$

De m̂

$$\frac{d\mathcal{L}}{db_j^p} = c_j^p$$

- $\dfrac{d\mathcal{L}}{dw_j^p}$, p quelconque ?

$$\frac{d\mathcal{L}}{d\tilde{x}_j^p} = \sum_p \frac{d\mathcal{L}}{d\tilde{x}_p^{p+1}} \times \frac{d\tilde{x}_p^{p+1}}{dx_j^p} \times \frac{dx_j^p}{d\tilde{x}_j^p}$$

$$c_j^p = \sum_p c_p^{p+1} \times w_{jp}^{p+1} \times x_j^p (1-x_j)^p$$

$$\Rightarrow \frac{d\mathcal{L}}{dw_j^p} = \frac{d\mathcal{L}}{d\tilde{x}_j^p} \times \frac{d\tilde{x}_j^p}{dw_j^p}$$

$$\begin{cases} \dfrac{d\mathcal{L}}{dw_j^p} = c_j^p \times x^{p-1} \\ \dfrac{d\mathcal{L}}{db_j^p} = c_j^p \end{cases}$$

Algorithme de rétropropagat° du gradient

- Ex: Écrire le pseudo code (matriciellement) avec les ≠ contraintes, que pour train RBM.

II Pré apprentissage de DNN

considérer les couches observées + cachées (couche de classif exclue) comme 1 DBN.

- Apprendre de manière non supervisée le DBN via Greedy Layer wise procedure. pour ~~initi~~ initialiser les paramètres du DNN

- Fine tuning : Apprentissage supervisé du DNN par rétropropagatio du gradient.