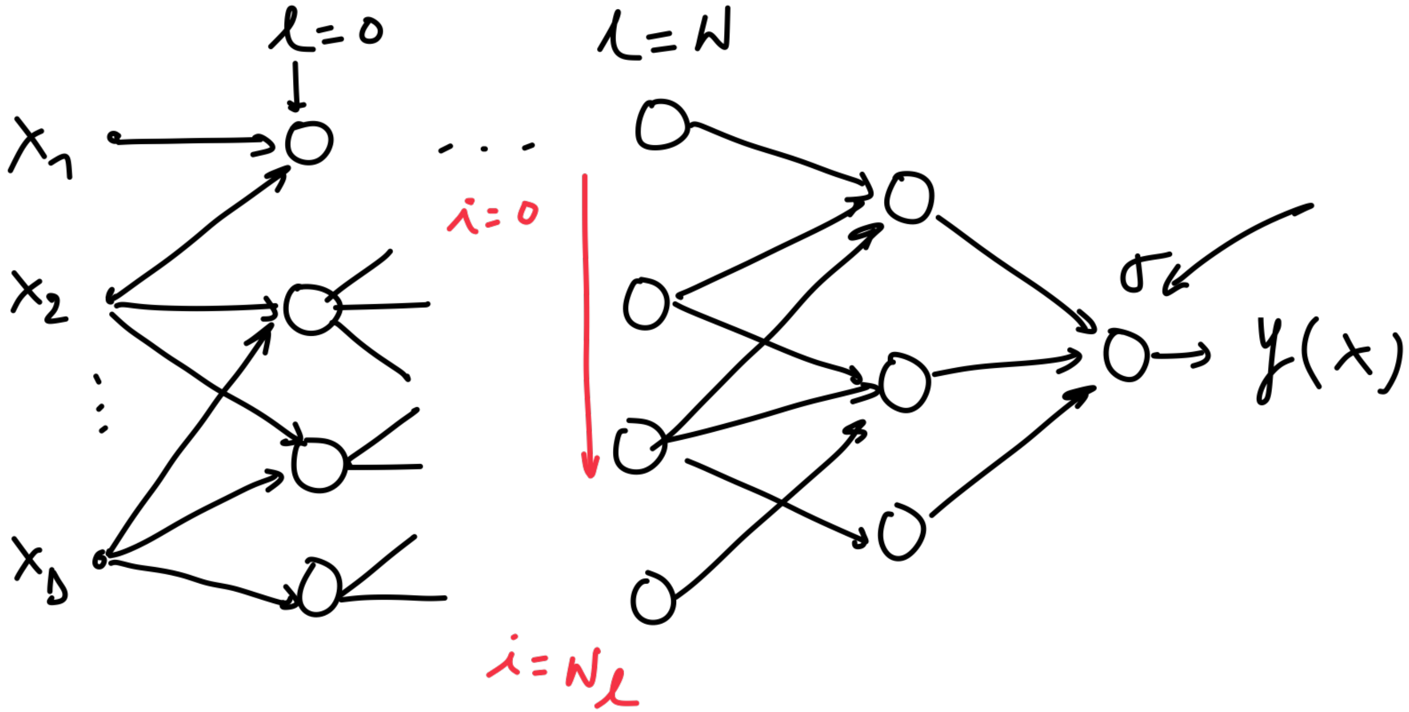
Réseau de Neurones



$\longrightarrow$ Apprentissage du réseau via Maximum de vraisemblance

$$p\left(\{t(x^{(i)}) = t^{(i)}\}_{i=1}^{N} \mid \{x^{(i)}\}\right) = \prod_{i=1}^{N} p\left(t(x^{(i)}) = t^{(i)} \mid x^{(i)}\right)$$

Cette fois-ci c'est le réseau lui même qui est utilisé pour représenter $p(t(x^{(i)}) = t^{(i)} \mid x^{(i)})$

$$p(t(x^{(i)}) = 1 \mid x^{(i)}) = y(x^{(i)})$$
$$p(t(x^{(i)}) = 0 \mid x^{(i)}) = 1 - y(x^{(i)})$$
$$p(t(x^{(i)}) = t^{(i)} \mid x^{(i)}) = y(x^{(i)})^{t^{(i)}} (1 - y(x^{(i)}))^{1 - t^{(i)}}$$

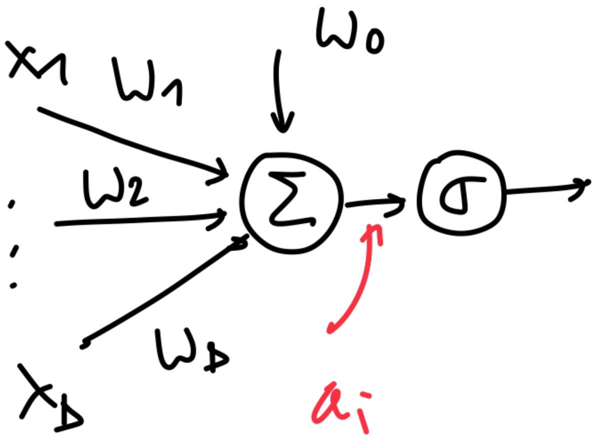
→ On peut ensuite entraîner le réseau en maximisant la log vraissemblance,

$$\max \quad \log \prod_{i=1}^{N} p(t(x^{(i)}) = t^{(i)} \mid x^{(i)})$$

$$\max \log \prod_{i=1}^{N} y(x^{(i)})^{t^{(i)}} \left(1 - y(x^{(i)})\right)^{1-t^{(i)}}$$

$$\min J = -\sum_{i=1}^{N} t^{(i)} \log \underbrace{y(x^{(i)})}_{\sigma(a_{out})} + (1-t^{(i)}) \log\left(1 - \underbrace{y(x^{(i)})}_{\sigma(a_{out})}\right)$$

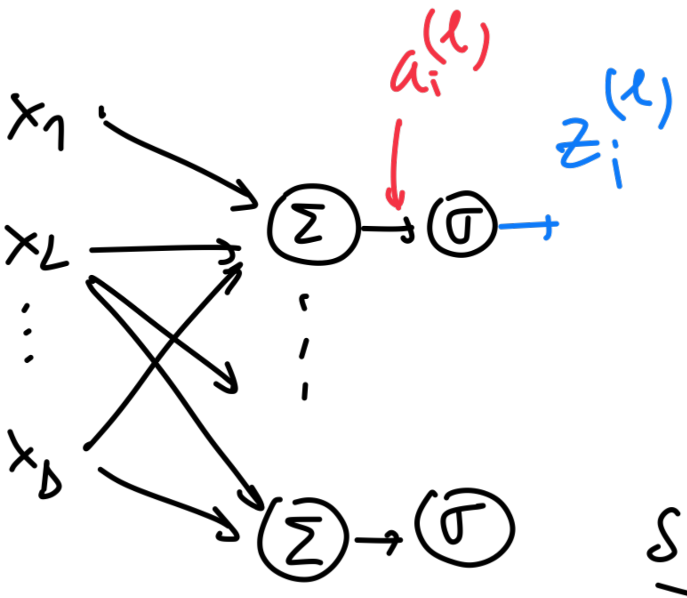
ENTROPIE BINAIRE CROISÉE $= J$



$a_i^{(l)} =$ préactivation associée au neurone i de la couche l

$$a_i^{(l)} = \sum_{i=1}^{N_l} w_{ij}^{(l)} z_j^{(l-1)} + w_{i0}^{(l)}$$

$$z_j^{(l-1)} = \sigma(a_j^{(l-1)})$$



→ Objectif : obtenir les gradients

$$\frac{\partial J}{\partial w_{ij}^{(l)}}$$

Step 1

$$\frac{\partial J}{\partial w_{ij}^{(l)}} = \frac{\partial J}{\partial a_i^{(l)}} \cdot \frac{\partial a_i^{(l)}}{\partial w_{ij}^{(l)}}$$

$$= \delta_i^{(l)} \quad z_j^{(l-1)}$$

gradient par rapport aux paramètres du neurone i de la couche l

Step 2

$$\delta_{out} = \frac{\partial J}{\partial a_{out}}$$

$$\delta_{out} = \frac{\partial}{\partial a_{out}} \left\{ -t^{(i)} \log\left(\sigma(a_{out})\right) - (1 - t^{(i)}) \log\left(1 - \sigma(a_{out})\right) \right\}$$

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

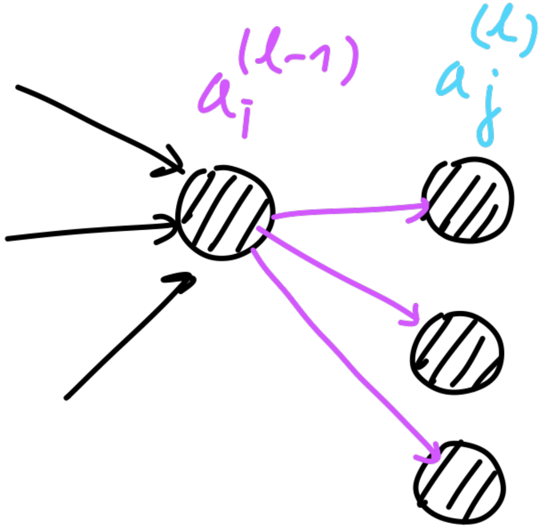
$$\delta_{out} = -\,t^{(i)} \frac{\sigma'(a_{out})}{\sigma(a_{out})} - (1-t^{(i)}) \frac{(-\sigma'(a_{out}))}{1-\sigma(a_{out})}$$

$$\sigma'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1}{1+e^{-x}}\right)\left(1 - \frac{1}{1+e^{-x}}\right) = \sigma(x)(1-\sigma(x))$$

$$\delta_{out} = -t^{(i)}(1-\sigma(a_{out})) + (1-t^{(i)})\,\sigma(a_{out})$$

$$= \sigma(a_{out}) - t^{(i)}$$

<u>Step 3</u>



chaque $a_i^{(l-1)}$ est utilisé par tous les neurones de la couche suivante (couche l)

$$\frac{\partial J}{\partial a_i^{(\ell-1)}} = \delta_i^{(\ell-1)} = \sum_{j=1}^{N_\ell} \underbrace{\frac{\partial J}{\partial a_j^{(\ell)}}} \cdot \frac{\partial a_j^{(\ell)}}{\partial a_i^{(\ell-1)}}$$

pour le second facteur, en utilisant la définition des préactivations

$$a_j^{(\ell)} = \sum_{i=1}^{N_\ell} w_{ji}^{(\ell)} \sigma(\underline{a_i^{(\ell-1)}}) + w_{i0}^{(\ell)}$$

$$\delta_i^{(\ell-1)} = \sum_{j=1}^{N_\ell} \delta_j^{(\ell)} w_{ji}^{(\ell)} \sigma'(a_i^{(\ell-1)})$$

Step 3 : à partir des $\delta_i^{(\ell)}$ obtenir les $\delta_i^{(\ell-1)}$ via

$$\delta_i^{(\ell-1)} = \sum_{j=1}^{N_\ell} \delta_j^{(\ell)} w_{ji}^{(\ell)} \sigma'(a_i^{(\ell-1)}) \quad (*)$$

__Backpropagation :__ (démarrer avec $w_{ij}^{(\ell)}$ aléatoires)

- Pour un exemple $\{x^{(i)}, t^{(i)}\}$, propager l'exemple à travers tout le réseau et générer toutes les valeurs de pré et post activations $a_i^{(\ell)}$, $z_j^{(\ell)}$ pour toute couche ℓ et tout neurone i

→ Calculer $\delta_{out} = y(x^{(i)}) - t^{(i)}$

→ Dériver tous les $\delta_i^{(\ell)}$ à partir de δ_{out} via (*)

→ Dériver les gradients via

$$\frac{\partial J}{\partial w_{ij}^{(\ell)}} = \delta_i^{(\ell)} z_j^{(\ell-1)}$$

→ Réaliser un pas dans la direction opposée au gradient

$$\boxed{w_{ij}^{(\ell)} \leftarrow w_{ij}^{(\ell)} - \eta \frac{\partial J}{\partial w_{ij}^{(\ell)}}}$$