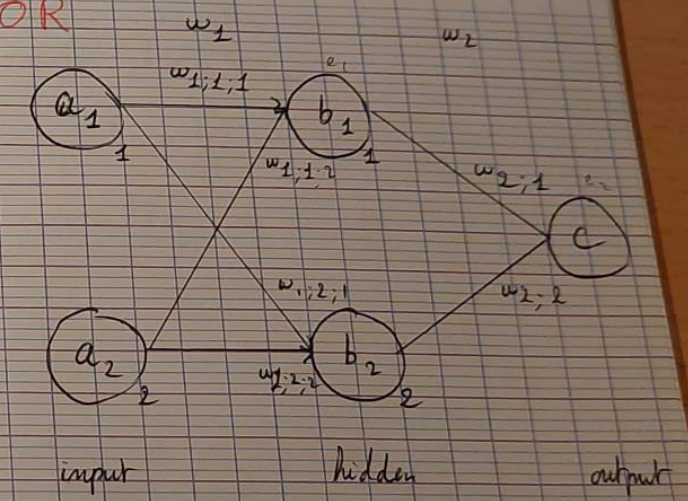


couche mince
parer et identi
igration le lon
d'un mélange
chéma a.
s apportées
ge M ?

XOR

Possibilités de rajouter
- Bias
- Activation "sigmoïde" $\sigma(a_2 \times w_2 + b_2)$



Propagation

$$a \times w_1 = \begin{pmatrix} a_1 & a_2 \end{pmatrix} \begin{pmatrix} w_{1,1;1} & w_{1,2;1} \\ w_{1,1;2} & w_{1,2;2} \end{pmatrix} = \begin{pmatrix} b_1 & b_2 \end{pmatrix}$$

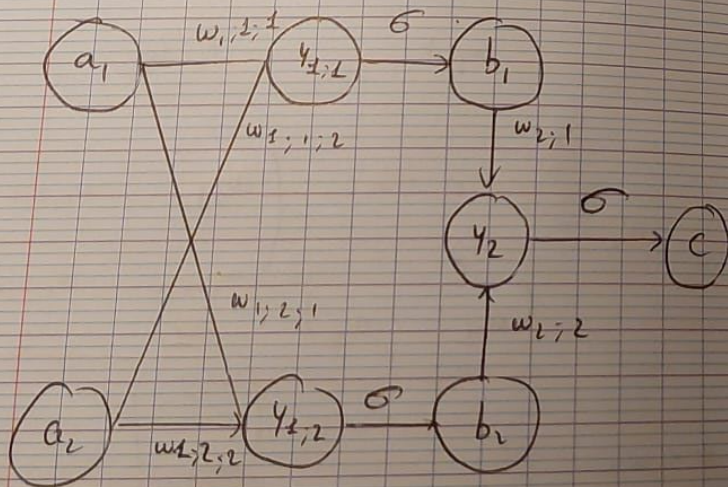
$$b \times w_2 = \begin{pmatrix} b_1 & b_2 \end{pmatrix} \begin{pmatrix} w_{2,1} \\ w_{2,2} \end{pmatrix} = \begin{pmatrix} c \end{pmatrix}$$

Cost x = training input
 y = training output

$$\text{cost} = (c - y_2)^2$$

plus c'est petit mieux c'est

Backward



On cherche à trouver

$$\rightarrow \frac{d\text{cost}}{dw_{2,1}} \quad \text{car } w_{2,1} = b_1 \times \frac{d\text{cost}}{dy_2}$$

De même pour

$$\boxed{w_{2,2}} \quad \frac{d\text{cost}}{dw_{2,2}} = \frac{d\text{cost}}{dy_2} \times \frac{dy_2}{db_2} \times \frac{db_2}{dw_{2,2}}$$

$$= \boxed{2(c-y) \times \sigma'(c) \times b_2}$$

$$= \sum_2 = \frac{d\text{cost}}{dy_2}$$

$$= \sum_2 \times b_2$$

On cherche à trouver

De même

pour

$$\boxed{w_{2,1,1}} \quad \frac{d\text{cost}}{dw_{2,1,1}} \quad \text{car } w_{2,1,1} = b_1 \times \frac{d\text{cost}}{dy_2}$$

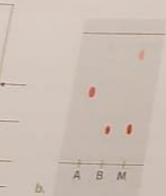
$$\frac{d\text{cost}}{dw_{2,1,1}} = \frac{d\text{cost}}{dy_2} \times \frac{dy_2}{db_1} \times \frac{db_1}{dy_{1,1}} \times \frac{dy_{1,1}}{dw_{2,1,1}}$$

$$= \boxed{\left(\sum_2 \times w_{2,1,1} \right) \times \sigma'(b_1) \times a_1}$$

ie sur couche mince (CCM) est ur
pour séparer et identifier les const
par migration le long d'une coust
la CCM d'un mélange M.

der le schéma a.

rmations apportées par cette CC
mélange M ?



DIFFÉRENCIATION

volume de dioxyde
avant expiration.

carbone provient uni-
culer le pourcentage
lume courant.

deux physiques

pour la plongée.

enu dans la bou-
de chaque gaz

ue du gaz à
phérique:

Conclusion

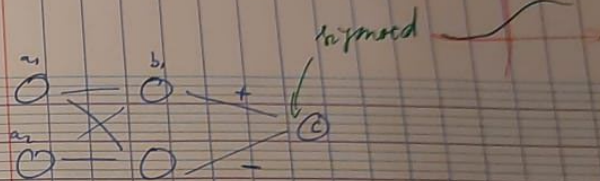
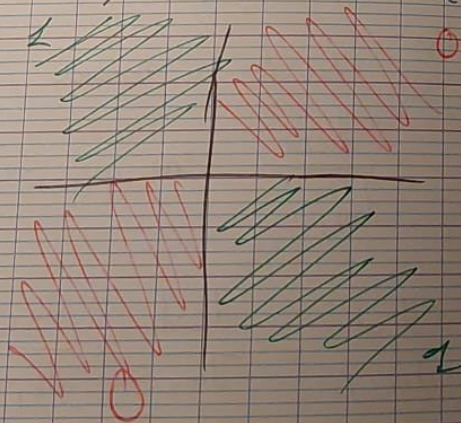
C'est peut se faire par matrice

Le modèle ne semble pas adéquat
pour le XOR
hidden - n ≥ 4 semble mieux
fonctionner.

$$b_1 = 0, 1$$

$$\approx 0, 01$$

Il faut mettre plus de donnée que
juste (0,1) (1,0) (0,0) et (1,1)



c doit parfois être égal à 0

G_1 peut poser $b_2 = a_1 - a_2$

$$b_2 = a_1 + a_2$$

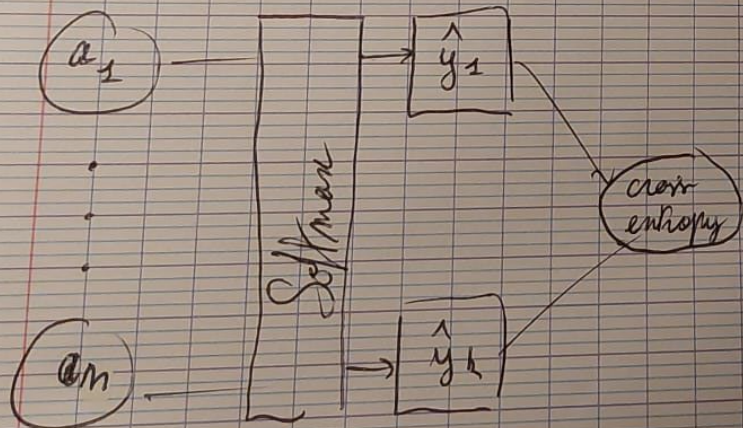
Maxis pour des valeurs max dans
les coins la réponse reste
approximative

Cross Entropy Loss

Utiliser avec Softmax

$$L_i = - \underbrace{y_i}_{\text{valeur souhaitée}} \log(\underbrace{\hat{y}_i}_{\text{valeur de sortie}}) \quad y_i \in]0, 1[$$

$$L_{\text{total}} = - \sum_i y_i \log(\hat{y}_i)$$



Softmax

$$i \in \llbracket 1; n \rrbracket$$

$$\hat{y}_i = \frac{e^{a_i}}{\sum_{j=1}^n e^{a_j}}$$

$$M_q = \frac{\partial L}{\partial a_k} = \hat{y}_m - y_m$$

Pour $i = k$

$$\frac{\partial \hat{y}_i}{\partial a_k} = \frac{e^{a_i} \sum_{j=1}^n e^{a_j} - e^{a_i} e^{a_k}}{\left(\sum_{j=1}^n e^{a_j} \right)^2}$$

$$= \frac{e^{a_i}}{\underbrace{\sum_{j=1}^n e^{a_j}}_{\hat{y}_i}} \times \left(1 - \frac{e^{a_k}}{\underbrace{\sum_{j=1}^n e^{a_j}}_{\hat{y}_k}} \right)$$

$$= \hat{y}_i (1 - \hat{y}_k) \quad (i=k)$$

$$= \hat{y}_k (1 - \hat{y}_k)$$

Pour $i \neq k$

$$\begin{aligned} \frac{\partial \hat{y}_i}{\partial \theta_k} &= \frac{0 \sum e^{a_j} - e^{a_i} e^{a_k}}{(\sum e^{a_j})^2} \\ &= - \frac{e^{a_i}}{(\sum e^{a_j})} \times \frac{e^{a_k}}{(\sum e^{a_j})} \\ &= - \hat{y}_i \hat{y}_k \end{aligned}$$

MNIST

Keras:

- Image que en 28×28
- Confusion entre $\{1, 4, 7\}$
- Problème lorsque je change de taille de stroke $\{2, 4, 8\}$ bon
 $\{16, 32\}$ mauvais
pour une image 128×128 .

→ Overfitting solutions:

- Dropout
- Entraînement avec des données normalisées "normaliser les données"

Xuanyu

- Trop lent à entraîner :

Optimizers

- Conv / Max pooling

- Poids de base trop grand (Initialisato)

↳ Glorot (Xavier) initialization

- Internal Covariate Shift

↳ Batch normalization

(sur des réseaux trop profonds)

Optimizers

- Algorithme de Métropolis

- Opérateur Laplacien

↳ Reçoit simule

- Momentum

↳ part gradient

- Adagrad

↳ Large gradient : faster

Small gradient : slower

- RMS PROP

↳ Moving average of gradients

- ADAM

Laplace

$$f: \mathbb{R}^3 \rightarrow \mathbb{R}$$

$$\Delta f(x, y, z) = \frac{\partial^2 f}{\partial x^2}(x, y, z) + \frac{\partial^2 f}{\partial y^2}(x, y, z) + \frac{\partial^2 f}{\partial z^2}(x, y, z)$$

Si $f: \mathbb{R}^2 \rightarrow \mathbb{R}$

$$\Delta f(x, y) = \frac{\partial^2 f}{\partial x^2}(x, y) + \frac{\partial^2 f}{\partial y^2}(x, y)$$

Timeline

1952

1958

Stochastic Gradient Descent

Perceptron

- learnable weights

1986

Backpropagation

- Multi-layer Perceptron

1999

Deep Convolutional Neural Networks

- Digit Recognition

EXERCICE

Exemple de test positif avec le réactif soude

le cours
ne mince (CCM) est
et identifier les com
on le long d'une cou
mélange M.
ma a.
reportées par cette
M?

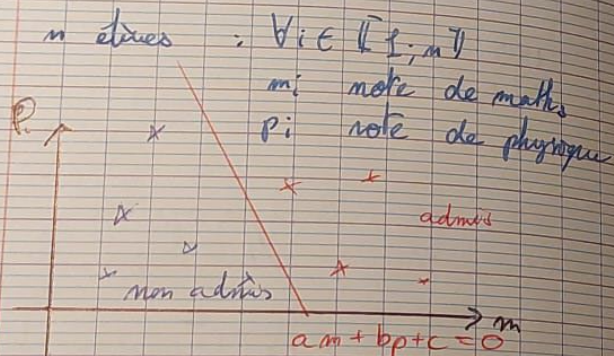


RENCIATIO

de dioxyde
piration.
e provient un
e pourcentage
ourant.

physiques
la plongée.
dans la bou
chaque gai

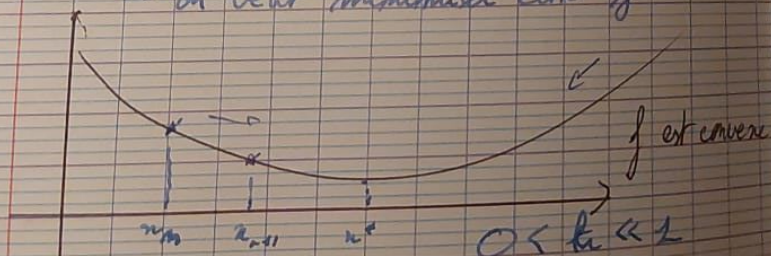
D° 2 Paramètres



Si $am + bp + c > 0$ l'élève est admis
 $am + bp + c < 0$ l'élève est non admis

$$E(a, b, c) = \sum_{i=1}^n (c_i - \sigma(am_i + bp_i + c))^2$$

on veut minimiser cette J^0



$$x_{m+1} = x_m - h f'(x_m)$$

H_1 : $x_m \xrightarrow{+ \infty} x^*$ (pour h assez petit)

H_2 : $f(x_m)$ est décroissante pour h assez petit

$$\forall m \in \mathbb{N}; \exists y_m \in h \text{ tq}$$

Lagrange

avec $h = -h f'(x_m)$

$$f(x_{m+1}) = f(x_m) - h f'(x_m) + \frac{h^2}{2} f''(y_m)$$

demande au prof

$$f(x_{m+1}) - f(x_m) = -h f'(x_m) + \frac{h^2}{2} f''(y_m)$$

$$= -h f'(x_m) \left[1 - \frac{h f''(y_m)}{2 f'(x_m)} \right]$$

> 1
 < 1

pour h assez petit
Donc $f(x_{m+1}) - f(x_m) < 0$

$$a \leftarrow a - h \frac{\partial E}{\partial a}(a, b, c)$$

$$b \leftarrow b - h \frac{\partial E}{\partial b}(a, b, c)$$

$$c \leftarrow c - h \frac{\partial E}{\partial c}(a, b, c)$$

$= -\sigma(am + bp + c)$
 $(c_i - \sigma(am_i + bp_i + c)) a_i$