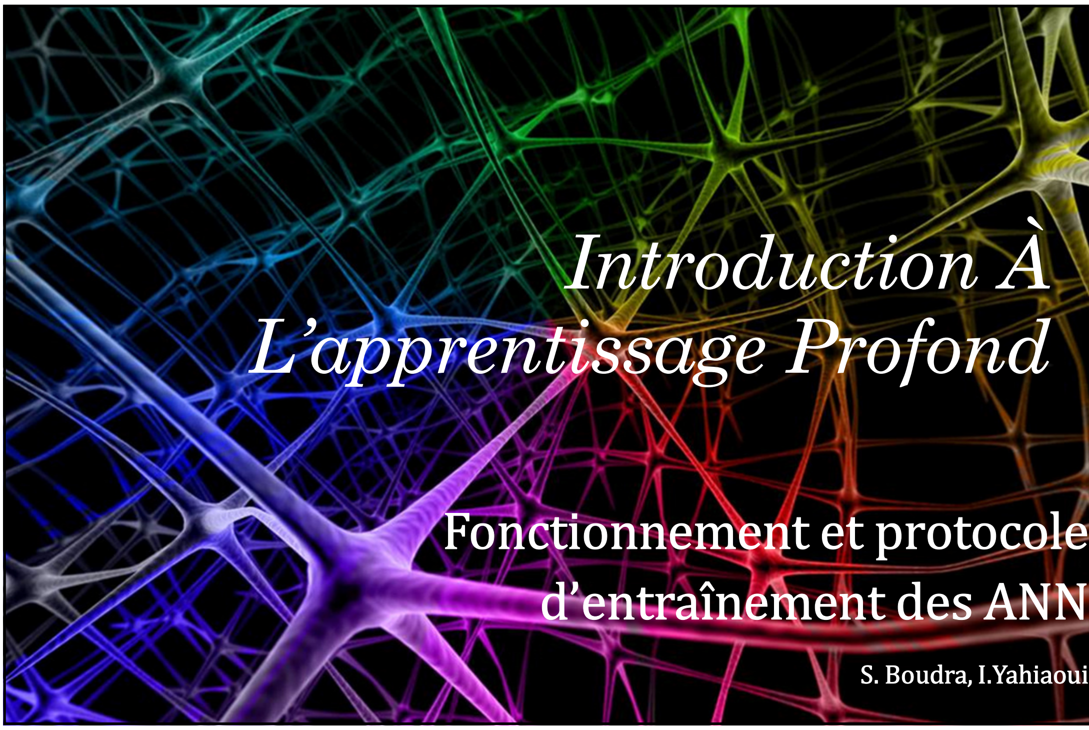


1

Ce qu’on va voir
Contenu

1. Propagation-Retro-propagation
2. Fonction d’activation
3. Fonction cout
4. Variantes du SGD

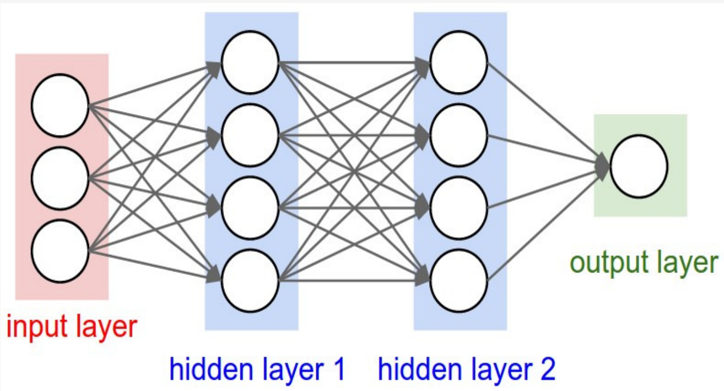
2

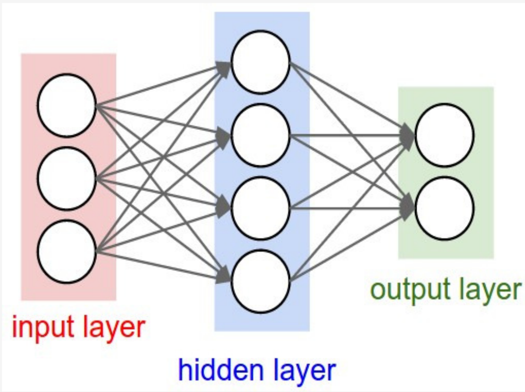
Rappel

1. Neurone biologique vs. Neurone artificielle
2. Quelques applications des réseaux de neurones
3. Classification binaire par perceptron
4. Problème XOR: besoin de couches cachées
5. Des réseaux aux réseaux profonds

3

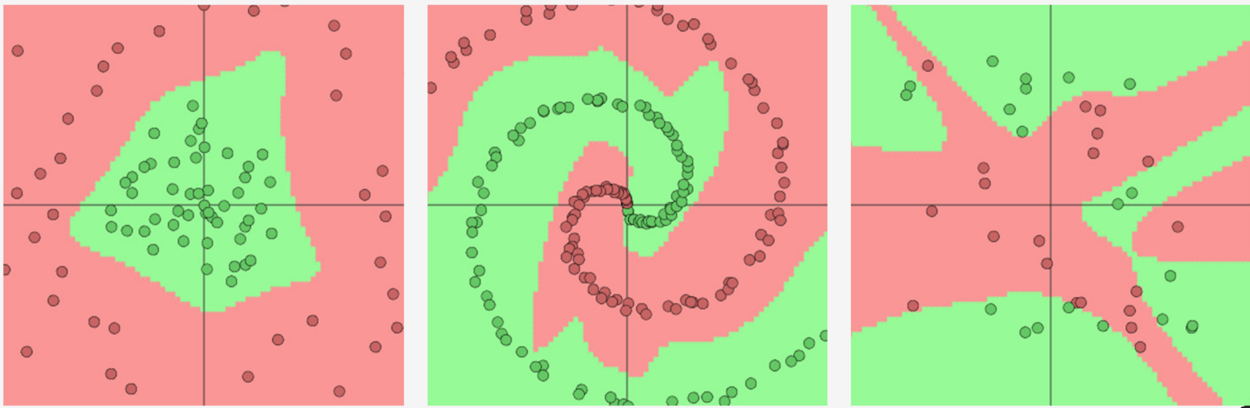
Réseaux de neurones





4

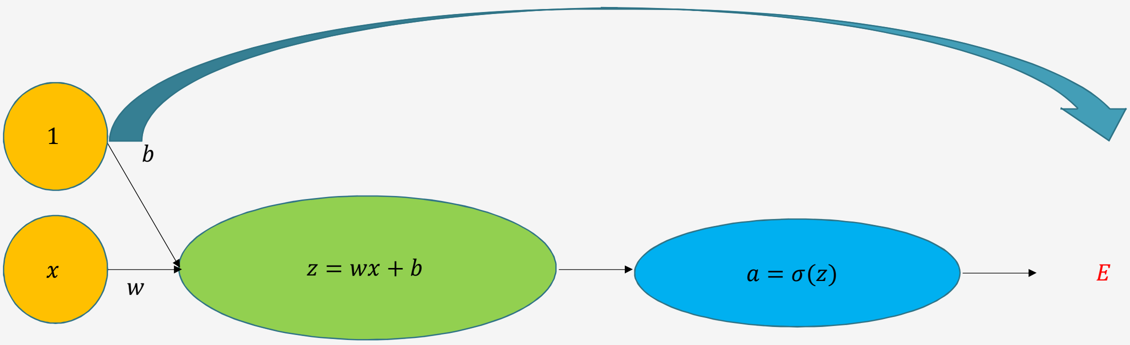
Séparation non-linéaire



- Les couches cachées permettent une séparation-non linéaire des données
- https://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html

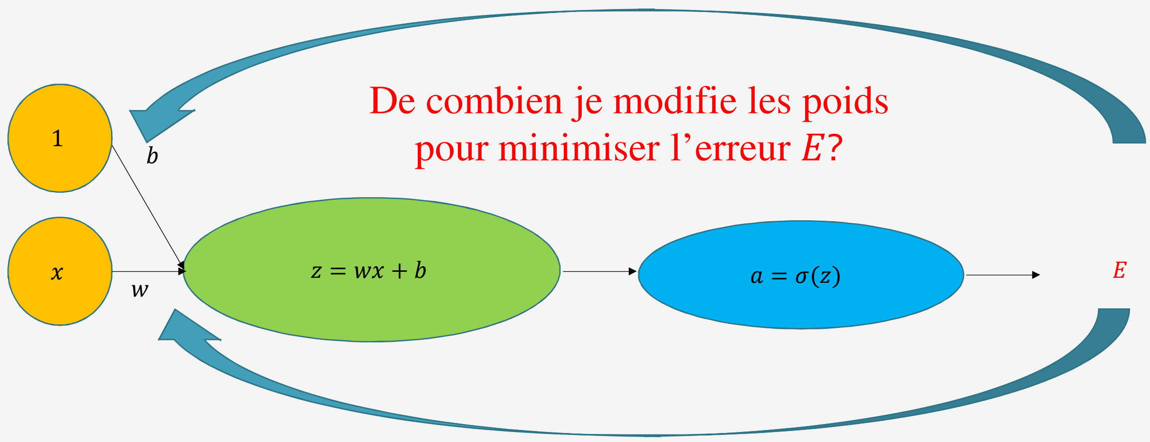
5

Propagation; Retro-propagation



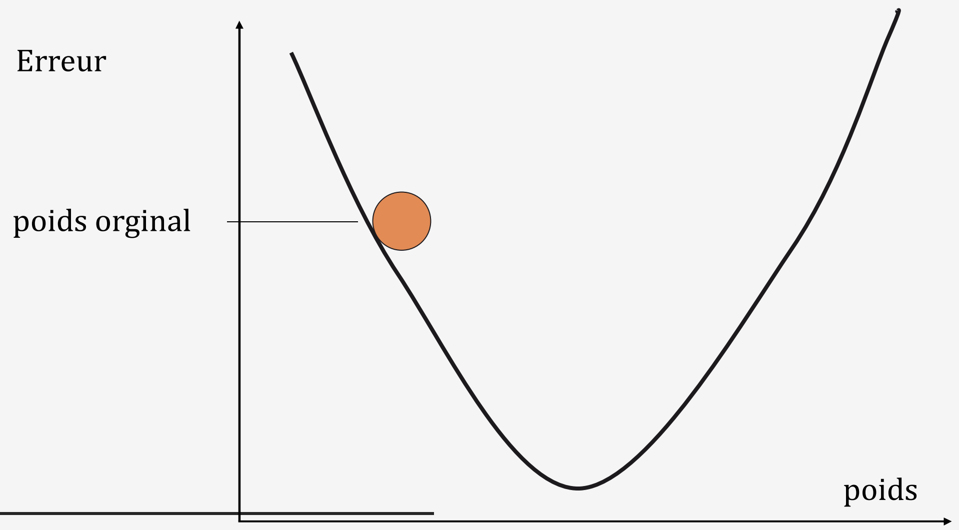
6

Propagation; Retro-propagation



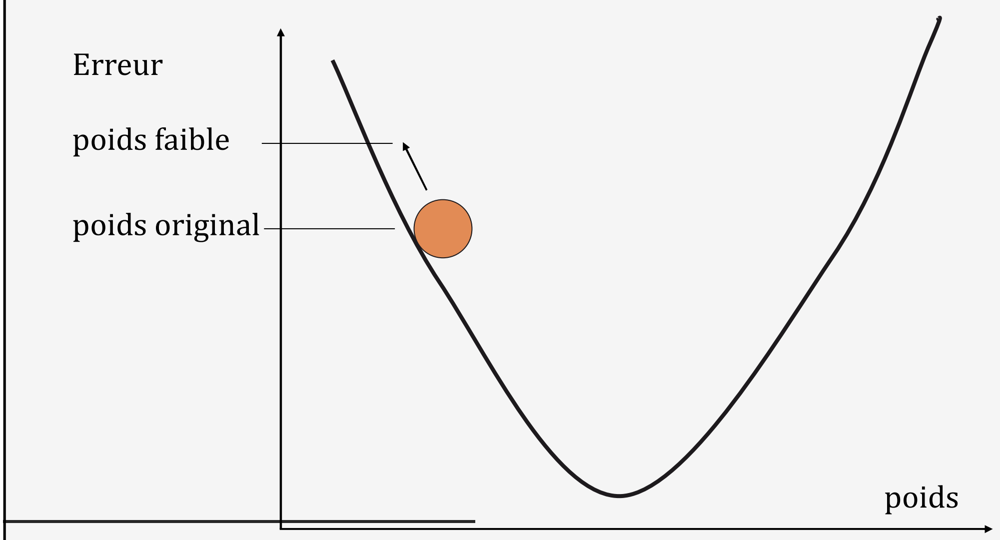
7

Gradient descent



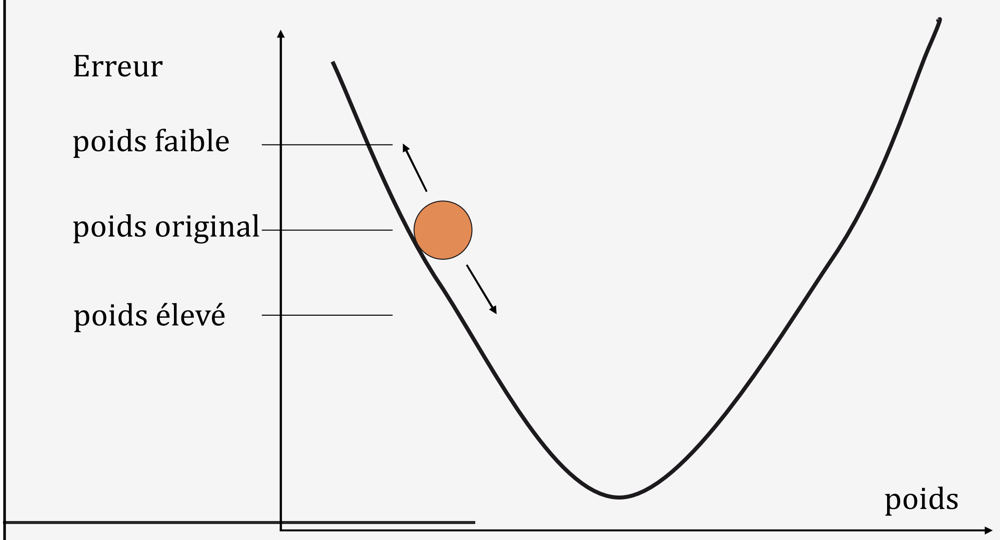
8

Gradient descent



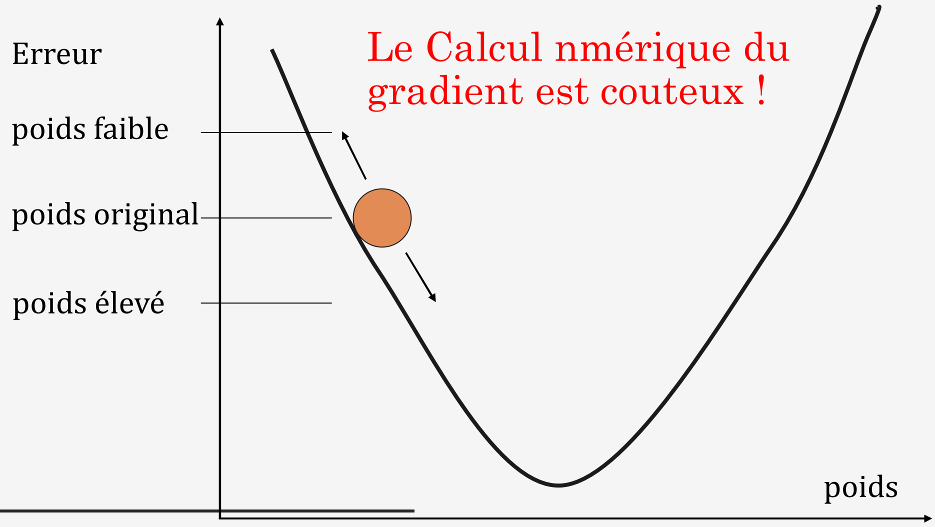
9

Gradient descent



10

Gradient descent



Le Calcul nmérique du gradient est couteux !

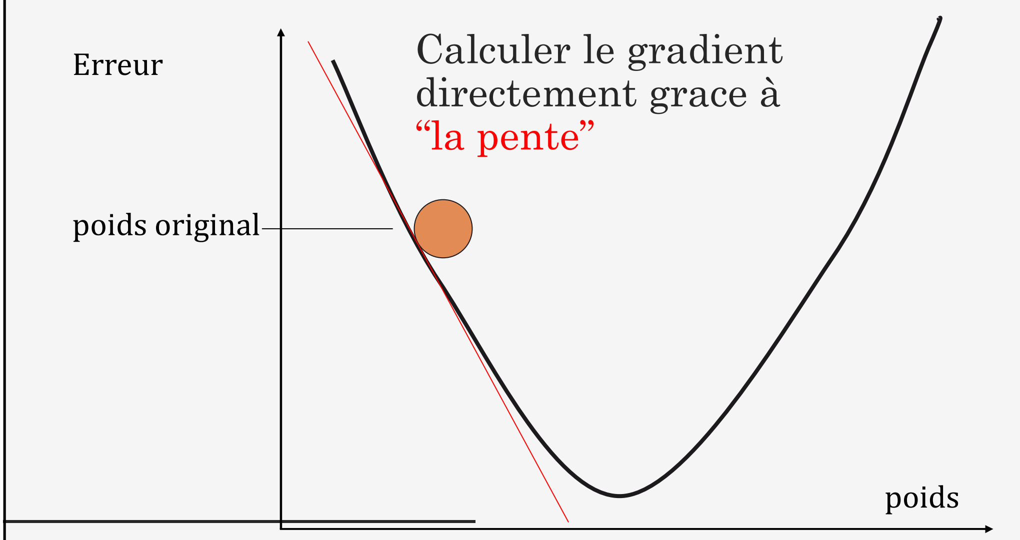
11

Gradient descent

Le Calcul numérique du gradient est « très » couteux!

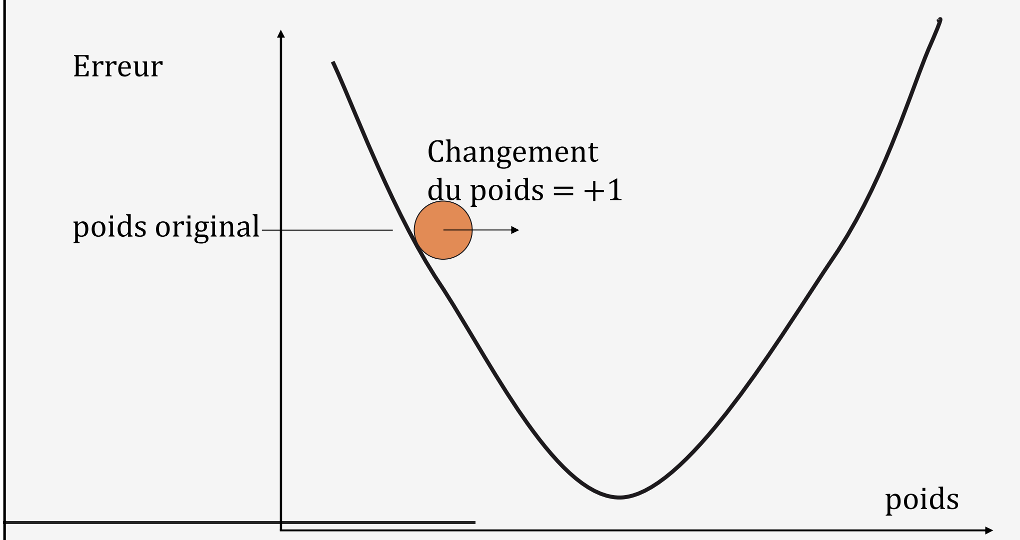
12

Gradient descent: la pente



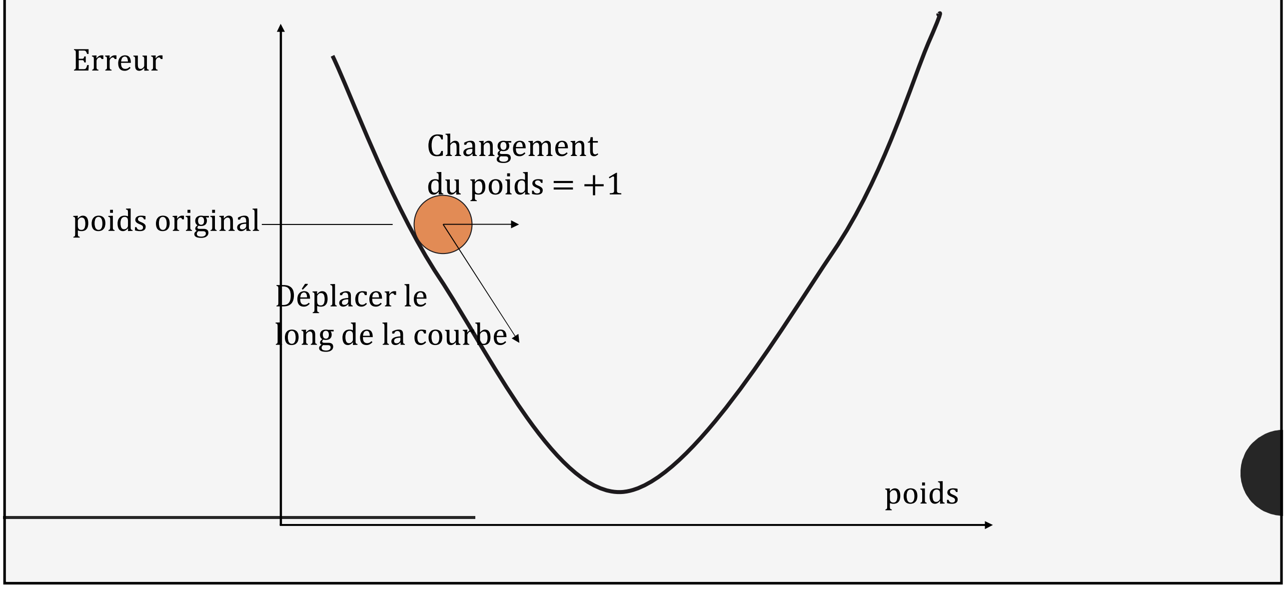
13

Gradient descent: la pente



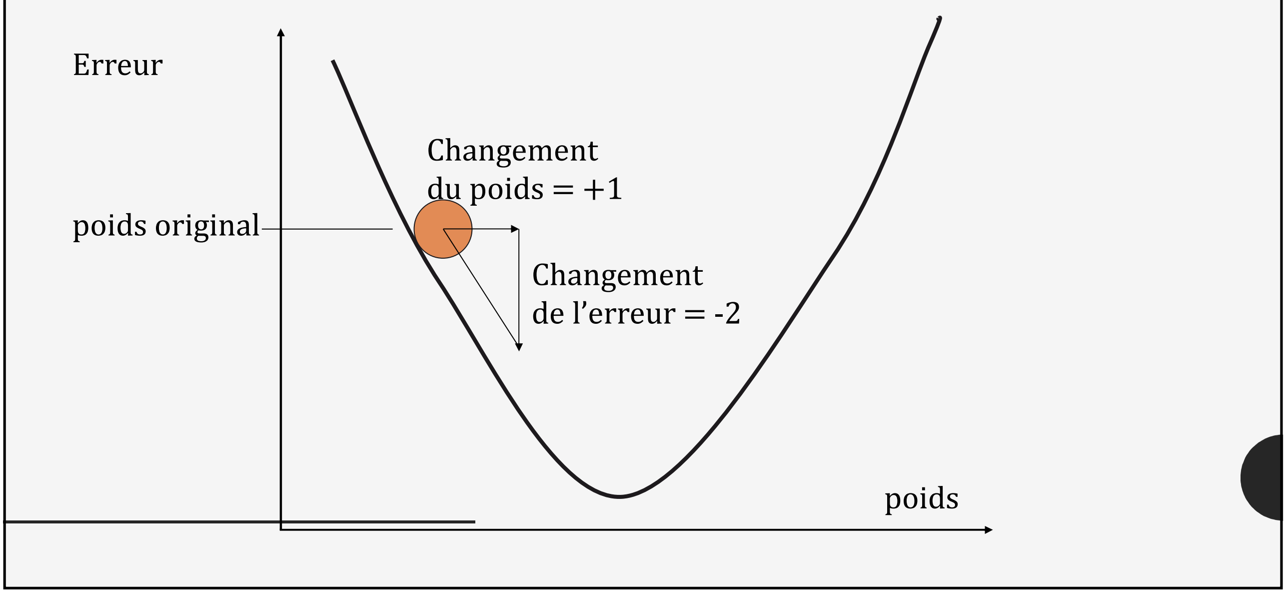
14

Gradient descent: la pente



15

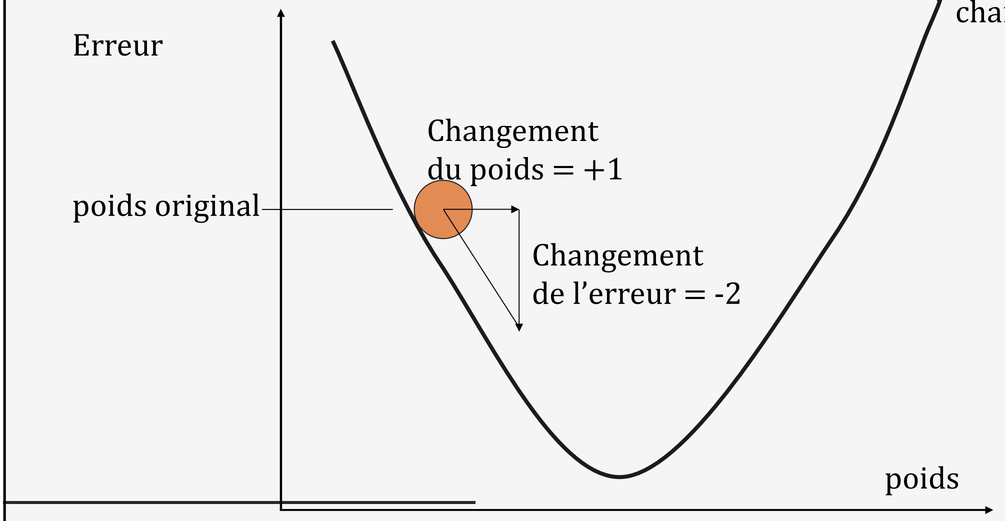
Gradient descent: la pente



16

Gradient descent: la pente

pente = $\dfrac{\text{changement de l'erreur}}{\text{changement du poids}}$



17

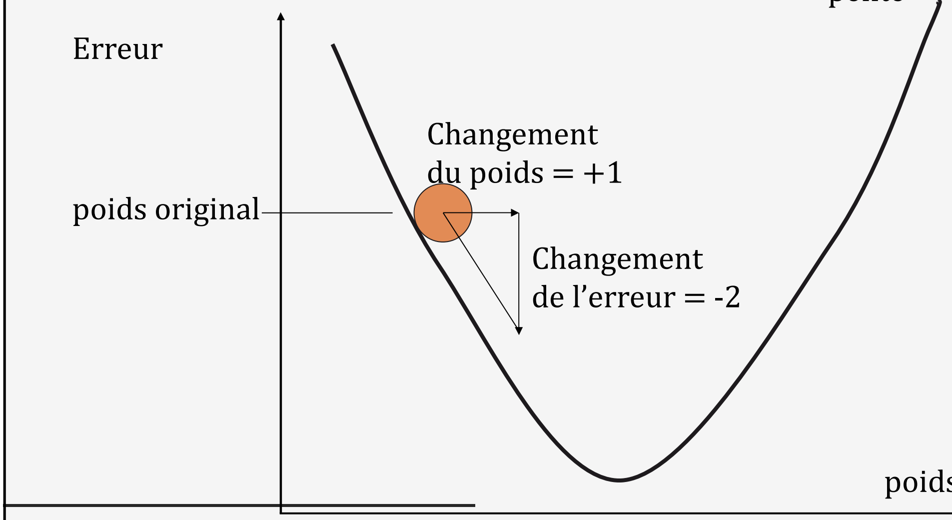
Gradient descent: la pente

pente = $\dfrac{\text{changement de l'erreur}}{\text{changement du poids}}$

$= \dfrac{\Delta \text{ erreur}}{\Delta \text{ poids}}$

$= \dfrac{d(\text{erreur})}{d(\text{poids})}$

$= \dfrac{\partial e}{\partial w}$



18

Gradient descent: la pente

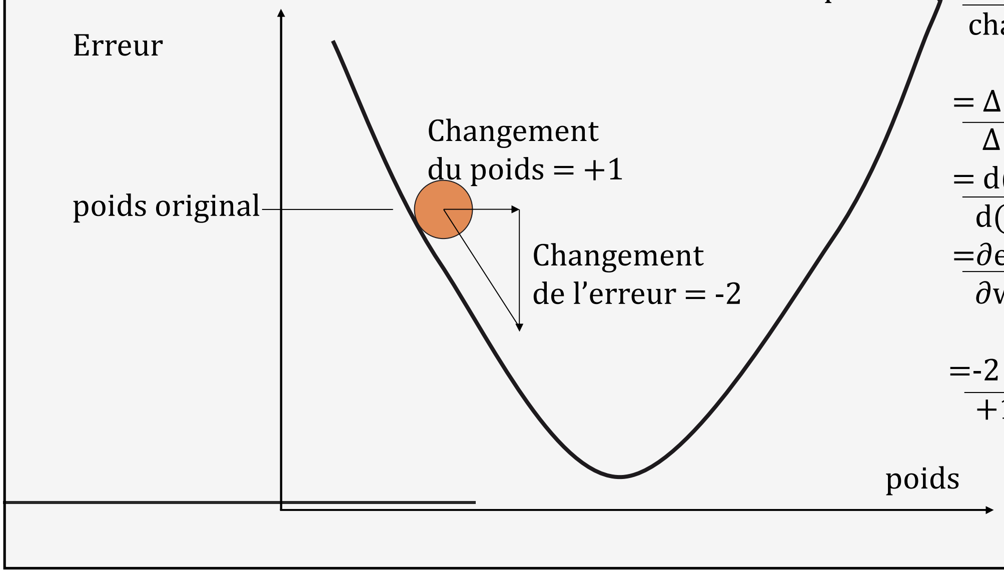
pente $= \dfrac{\text{changement de l'erreur}}{\text{changement du poids}}$

$= \dfrac{\Delta\ \text{erreur}}{\Delta\ \text{poids}}$

$= \dfrac{d(\text{erreur})}{d(\text{poids})}$

$= \dfrac{\partial e}{\partial w}$

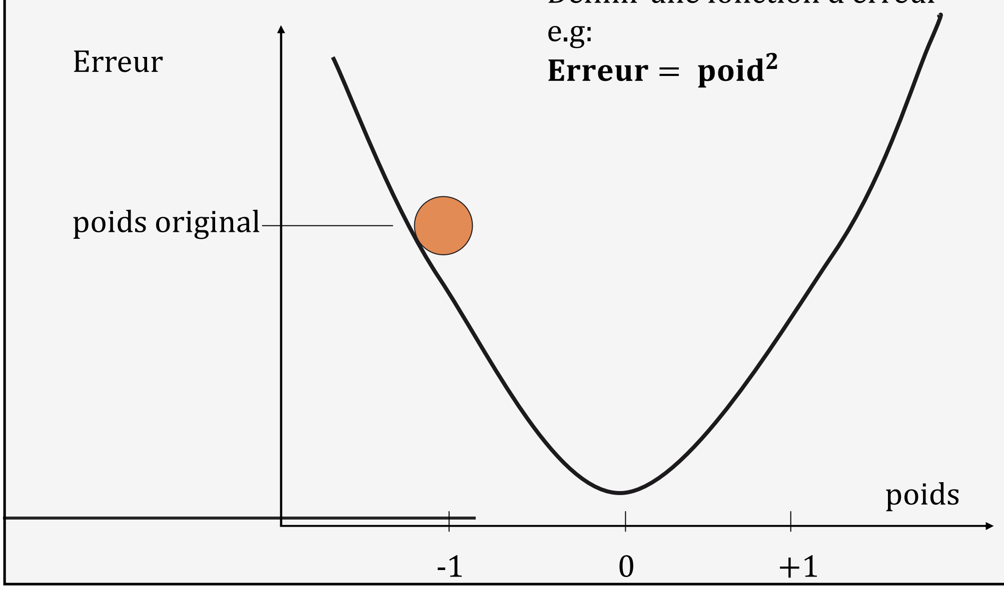
$= \dfrac{-2}{+1} = -2$



19

Gradient descent: la pente

Définir une fonction d'erreur
e.g:
Erreur = poid^2

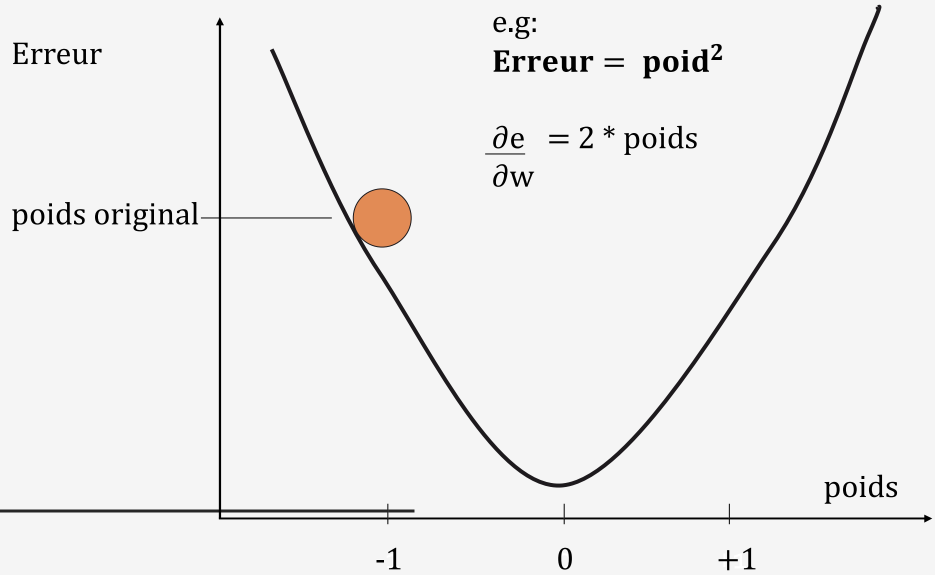


20

Gradient descent: la pente

Définir une fonction d'erreur
e.g:
Erreur $=$ **poid**^{2}

$$\frac{\partial e}{\partial w} = 2 * \text{poids}$$

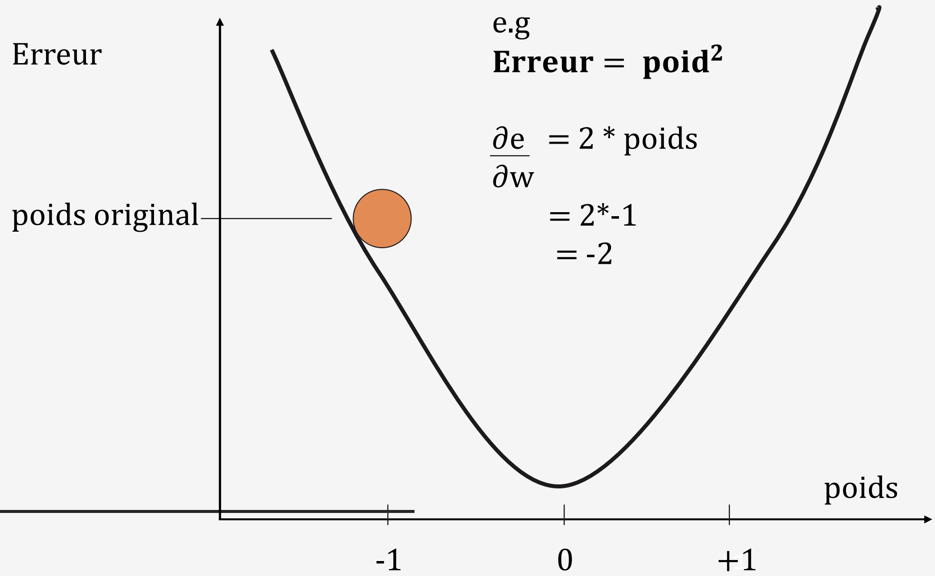


21

Gradient descent: la pente

Définir une fonction d'erreur
e.g
Erreur $=$ **poid**^{2}

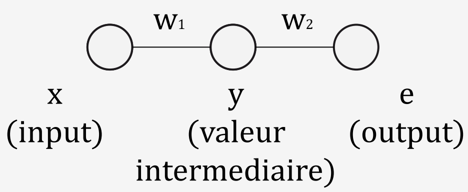
$$\frac{\partial e}{\partial w} = 2 * \text{poids}$$
$$= 2*-1$$
$$= -2$$



22

Chain rule

$$\frac{\partial e}{\partial w_1} = ??$$

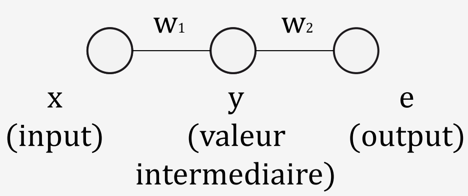


23

Chain rule

$$y = x * w_1$$

$$\frac{\partial y}{\partial w_1} = x$$



24

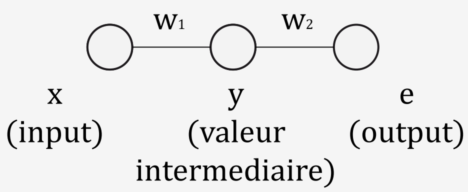
Chain rule

$y = x * w_1$

$\frac{\partial y}{\partial w_1} = x$

$e = y * w_2$

$\frac{\partial e}{\partial y} = w_2$



25

Chain rule

$y = x * w_1$

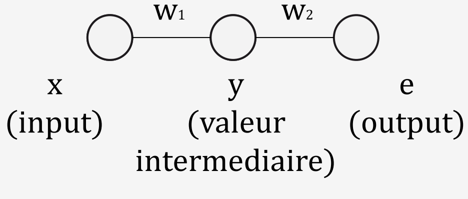
$\frac{\partial y}{\partial w_1} = x$

$e = y * w_2$

$\frac{\partial e}{\partial y} = w_2$

$e = x * w_1 * w_2$

$\frac{\partial e}{\partial w_1} = x * w_2$



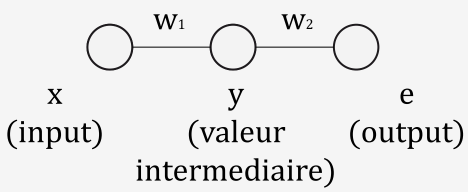
26

Chain rule

$$y = x * w_1$$
$$\frac{\partial y}{\partial w_1} = x$$

$$e = y * w_2$$
$$\frac{\partial e}{\partial y} = w_2$$

$$e = x * w_1 * w_2$$
$$\frac{\partial e}{\partial w_1} = x * w_2$$
$$\frac{\partial e}{\partial w_1} = \frac{\partial y}{\partial w_1} * \frac{\partial e}{\partial y}$$



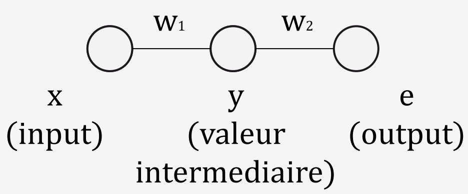
27

Chain rule

$$y = x * w_1$$
$$\frac{\partial y}{\partial w_1} = x$$

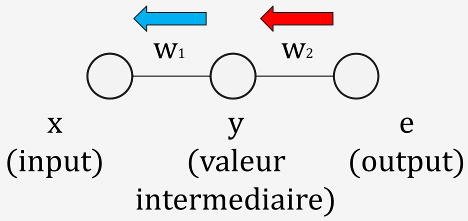
$$e = y * w_2$$
$$\frac{\partial e}{\partial y} = w_2$$

$$e = x * w_1 * w_2$$
$$\frac{\partial e}{\partial w_1} = x * w_2$$
$$\frac{\partial e}{\partial w_1} = \frac{\partial y}{\partial w_1} * \frac{\partial e}{\partial y}$$



28

Chain rule

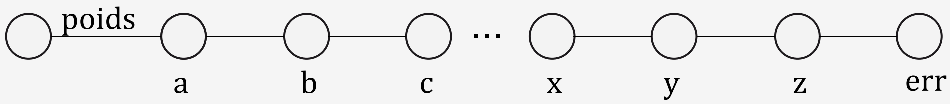


$$\frac{\partial e}{\partial w_1} = \frac{\partial y}{\partial w_1} * \frac{\partial e}{\partial y}$$

29

Chain rule

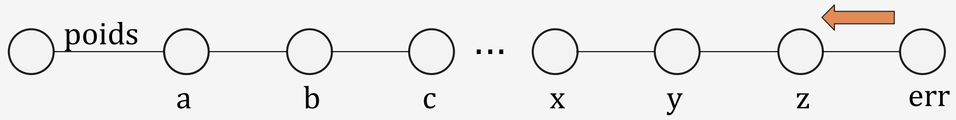
$$\frac{\partial err}{\partial poids} = \frac{\partial a}{\partial poids} * \frac{\partial b}{\partial a} * \frac{\partial c}{\partial b} * \frac{\partial d}{\partial c} * \ldots * \frac{\partial y}{\partial x} * \frac{\partial z}{\partial y} * \frac{\partial err}{\partial z}$$



30

Rétropropagation

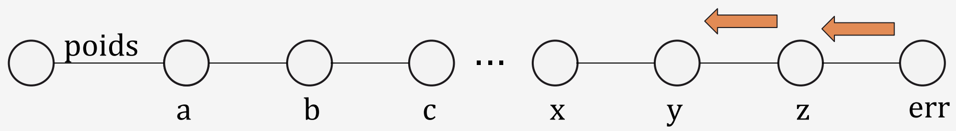
$$\frac{\partial err}{\partial poids} = \frac{\partial a}{\partial poids} * \frac{\partial b}{\partial a} * \frac{\partial c}{\partial b} * \frac{\partial d}{\partial c} * \ldots * \frac{\partial y}{\partial x} * \frac{\partial z}{\partial y} * \frac{\partial err}{\partial z}$$



31

Rétropropagation

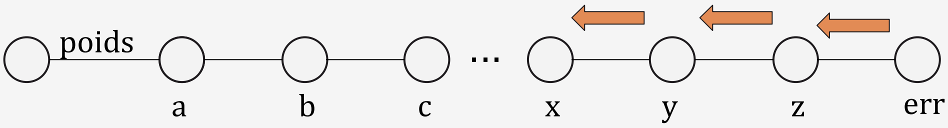
$$\frac{\partial err}{\partial poids} = \frac{\partial a}{\partial poids} * \frac{\partial b}{\partial a} * \frac{\partial c}{\partial b} * \frac{\partial d}{\partial c} * \ldots * \frac{\partial y}{\partial x} * \frac{\partial z}{\partial y} * \frac{\partial err}{\partial z}$$



32

Rétropropagation

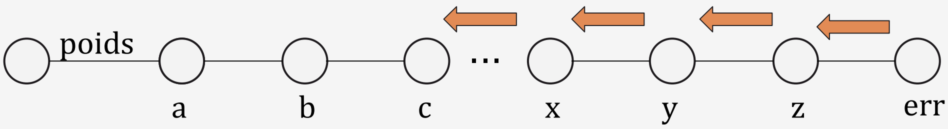
$$\frac{\partial err}{\partial poids} = \frac{\partial a}{\partial poids} * \frac{\partial b}{\partial a} * \frac{\partial c}{\partial b} * \frac{\partial d}{\partial c} * \ldots * \frac{\partial y}{\partial x} * \frac{\partial z}{\partial y} * \frac{\partial err}{\partial z}$$



33

Rétropropagation

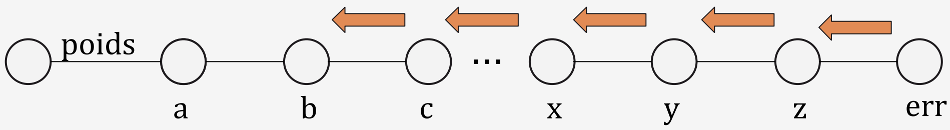
$$\frac{\partial err}{\partial poids} = \frac{\partial a}{\partial poids} * \frac{\partial b}{\partial a} * \frac{\partial c}{\partial b} * \frac{\partial d}{\partial c} * \ldots * \frac{\partial y}{\partial x} * \frac{\partial z}{\partial y} * \frac{\partial err}{\partial z}$$



34

Rétropropagation

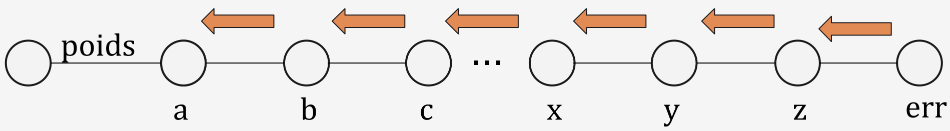
$$\frac{\partial err}{\partial poids} = \frac{\partial a}{\partial poids} * \frac{\partial b}{\partial a} * \frac{\partial c}{\partial b} * \frac{\partial d}{\partial c} * \ldots * \frac{\partial y}{\partial x} * \frac{\partial z}{\partial y} * \frac{\partial err}{\partial z}$$



35

Rétropropagation

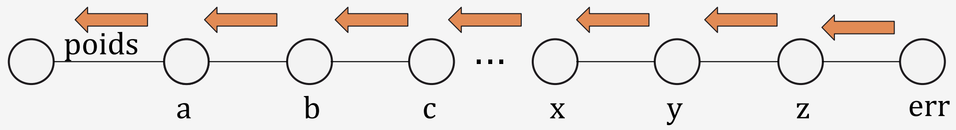
$$\frac{\partial err}{\partial poids} = \frac{\partial a}{\partial poids} * \frac{\partial b}{\partial a} * \frac{\partial c}{\partial b} * \frac{\partial d}{\partial c} * \ldots * \frac{\partial y}{\partial x} * \frac{\partial z}{\partial y} * \frac{\partial err}{\partial z}$$



36

Rétropropagation

$$\frac{\partial \text{err}}{\partial \text{poids}} = \frac{\partial \text{a}}{\partial \text{poids}} * \frac{\partial \text{b}}{\partial \text{a}} * \frac{\partial \text{c}}{\partial \text{b}} * \frac{\partial \text{d}}{\partial \text{c}} * \ldots * \frac{\partial \text{y}}{\partial \text{x}} * \frac{\partial \text{z}}{\partial \text{y}} * \frac{\partial \text{err}}{\partial \text{z}}$$



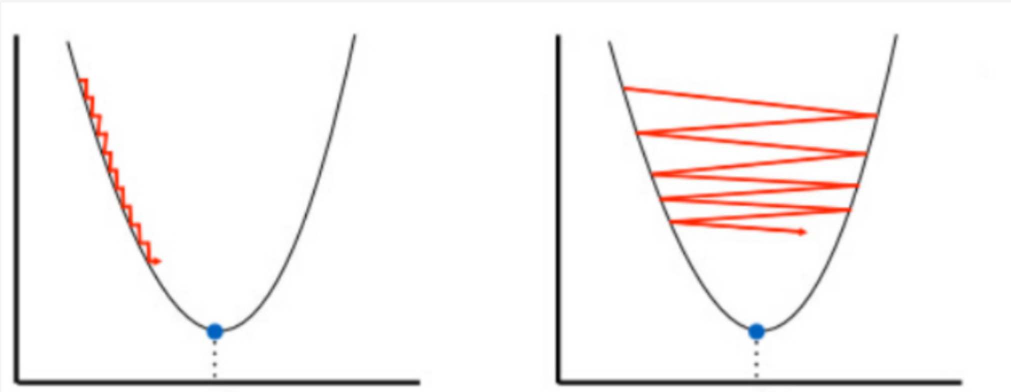
37

1. Initialiser aléatoirement les poids $\boldsymbol{w}$ et les biais $\boldsymbol{b}$
2. Définir une fonction d'activation
3. Calculer la valeur de sortie prédite $\hat{\boldsymbol{y}}$ avec l'opération de propagation:
4. Définir une fonction d'erreur $\boldsymbol{E}(\boldsymbol{y}, \hat{\boldsymbol{y}})$
5. Calculer les dérivées partielles (chain rule) $\frac{\boldsymbol{\delta E}}{\boldsymbol{\delta w}}; \frac{\boldsymbol{\delta E}}{\boldsymbol{\delta b}}$
6. Définir un taux d'apprentissage $\boldsymbol{\eta}$
7. Mettre à jour les poids $\boldsymbol{w}^{+} = \boldsymbol{w} - \boldsymbol{\eta}\frac{\boldsymbol{\delta E}}{\boldsymbol{\delta w}}$
8. Mettre à jours les biais $\boldsymbol{b}^{+} = \boldsymbol{b} - \boldsymbol{\eta}\frac{\boldsymbol{\delta E}}{\boldsymbol{\delta b}}$

Algorithme de rétro-propagation

38

Taux d'erreur



- $\boldsymbol{\eta}$ très petit => converge très lentement

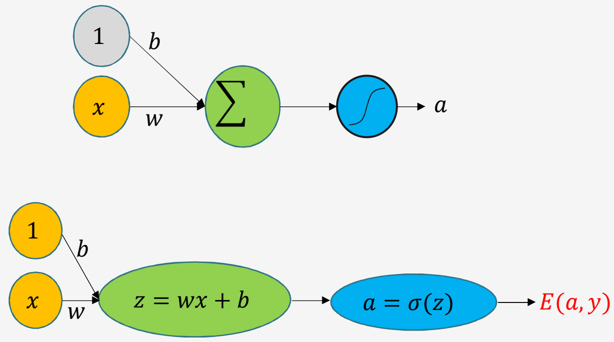
$\boldsymbol{\eta}$ très grand => risque de divergence

39

Exemple numérique

40

Descente de gradient

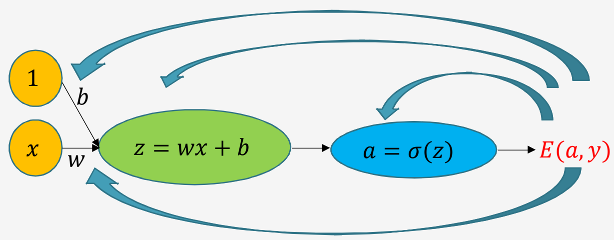


1. Initialiser aléatoirement les poids w et les biais b

2. Calculer la valeur de sortie prédite avec l'opération de propagation:
$$z = wx + b$$
3. Définir une fonction d'activation:
$$\sigma = \frac{1}{1+e^{-z}}$$

4. Définir une fonction d'erreur (loss):
$$E = \frac{1}{2}(y - a)^2$$

41



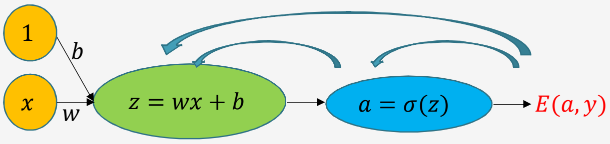
1. De combien je modifie les poids pour minimiser l'erreur E

=> Calculer les changements partiels

⇒ Les dérivées partielles

$$\frac{\partial E}{\partial a} \quad \frac{\partial E}{\partial z} \quad \frac{\partial E}{\partial w} \quad \frac{\partial E}{\partial b}$$

42



1. Dérivées partielles

$$\frac{\partial E}{\partial a} \quad \frac{\partial E}{\partial z} \quad \frac{\partial E}{\partial w} \quad \frac{\partial E}{\partial b}$$

$$\frac{\partial E}{\partial a}$$

$$E = \frac{1}{2}(y-a)^2$$

$$\frac{\partial E}{\partial a} = \frac{\partial(\frac{1}{2}(y-a)^2)}{\partial a} \Rightarrow \frac{\partial E}{\partial a} = a - y$$

$$\frac{\partial E}{\partial z} = \frac{\partial E}{\partial a} \times \frac{\partial a}{\partial z}$$

$$\frac{\partial a}{\partial z} = \frac{\partial(\frac{1}{1+e^{-z}})}{\partial z} \Rightarrow \frac{\partial a}{\partial z} = a(1-a)$$

$$\Rightarrow \frac{\partial E}{\partial z} = (a-y)a(1-a)$$

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w} \Rightarrow \frac{\partial E}{\partial w} = \frac{\partial E}{\partial z} \times \frac{\partial z}{\partial w}$$

$$\frac{\partial z}{\partial w} = \frac{\partial(wx+b)}{\partial w} \Rightarrow \frac{\partial z}{\partial w} = x$$

$$\Rightarrow \frac{\partial E}{\partial w} = (a-y)a(1-y)x$$

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial b} \Rightarrow \frac{\partial E}{\partial w} = \frac{\partial E}{\partial z} \times \frac{\partial z}{\partial b}$$

$$\frac{\partial z}{\partial b} = \frac{\partial(wx+b)}{\partial b} \Rightarrow \frac{\partial z}{\partial b} = 1$$

$$\Rightarrow \frac{\partial E}{\partial b} = (a-y)a(1-y)$$

$$\frac{\partial E}{\partial w} = (a-y)a(1-y)x$$

$$w^+ = w - \eta \times \frac{\partial E}{\partial w}$$

$$\frac{\partial E}{\partial b} = (a-y)a(1-y)$$

$$b^+ = b - \eta \times \frac{\partial E}{\partial b}$$

43

1. **Initialiser** aléatoirement les poids w et les biais b.
2. **Propagation**:

$z_{h_1} = w_1 x_1 + w_2 x_2 + b_1$

$z_{h_1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 = 0.3775$

$a_{h_1} = \frac{1}{1+e^{-z_{h_1}}} = \frac{1}{1+e^{-0.3775}} = 0.593269992$

$a_{h_2} = 0.596884378$

……

$z_{a_1} = w_5 a_{h_1} + w_6 a_{h_2} + b_2$

$z_{a_1} = 0.4 * 0.593269992 + 0.45 * 0.596884378 + 0.6 = 1.105905967$

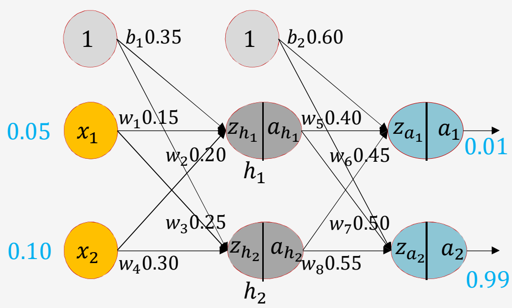
$a_1 = \frac{1}{1+e^{-z_{a_1}}} = \frac{1}{1+e^{-1.105905967}} = 0.75136507$

$a_2 = 0.772928465$

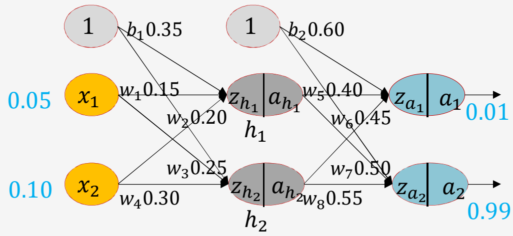
3. **Fonction d'erreur:**

$E_{a_1} = \frac{1}{2}(y_1 - a_1)^2 = \frac{1}{2}(0.01 - 0.75136507)^2 = 0.274811083;$ $\quad E_{a_2} = 0.023560026$

$E_t = E_{a_1} + E_{a_2} = 0.274811083 + 0.023560026 = 0.298371109$



44



1. Retropropagation:

Couche de sortie

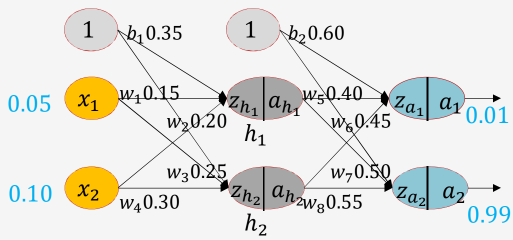
$$\frac{\partial E_t}{\partial w_5} = \frac{\partial E_t}{\partial a_1} \times \frac{\partial a_1}{\partial z_{a_1}} \times \frac{\partial z_{a_1}}{\partial w_5}$$

$$\frac{\partial E_t}{\partial a_1} = \frac{\partial(\frac{1}{2}(y_1 - a_1)^2 + \frac{1}{2}(y_2 - a_2)^2)}{\partial a_1} = a_1 - y_1 = 0.75136507 - 0.01 \Rightarrow \frac{\partial E_t}{\partial a_1} = 0.74136507$$

$$\frac{\partial a_1}{\partial z_{a_1}} = \frac{\partial(\frac{1}{1+e^{-z_{a_1}}})}{\partial z_{a_1}} = a_1(1 - a_1) = 0.75136507(1-0.75136507) \Rightarrow \frac{\partial a_1}{\partial z_{a_1}} = 0.186815602$$

$$\frac{\partial z_{a_1}}{\partial w_5} = \frac{\partial(w_5 a_{h_1} + w_6 a_{h_2} + b_2)}{\partial w_5} = a_{h_1} = 0.593269992$$

45



1. Retropropagation:

Couche de sortie

$$\frac{\partial E_t}{\partial w_5} = \frac{\partial E_t}{\partial a_1} \times \frac{\partial a_1}{\partial z_{a_1}} \times \frac{\partial z_{a_1}}{\partial w_5}$$

$$\frac{\partial E_t}{\partial w_5} = \frac{\partial E_t}{\partial a_1} \times \frac{\partial a_1}{\partial z_{a_1}} \times \frac{\partial z_{a_1}}{\partial w_5} = 0.74136507 \times 0.186815602 \times 0.593269992 = 0.082167041$$

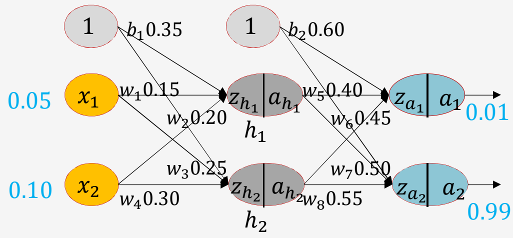
$$w_5^+ = w_5 - \eta * \frac{\partial E_t}{\partial w_5} = 0.4 - 0.5 \times 0.082167041 = 0.35891648$$

$w_6^+ = 0.408666186;$
$w_7^+ = 0.511301270\ ;$
$w_8^+ = 0.561370121$

46

1. Retropropagation:

Biais b_2



$$\frac{\partial E_t}{\partial b_2} = \frac{\partial E_t}{\partial a_1} \times \frac{\partial a_1}{\partial z_{a_1}} \times \frac{\partial z_{a_1}}{\partial b_2} + \frac{\partial E_t}{\partial a_2} \times \frac{\partial a_2}{\partial z_{a_2}} \times \frac{\partial z_{a_2}}{\partial b_2}$$

$$\frac{\partial E_t}{\partial a_2} = a_2 - y_2 = 0.772928465 - 0.99 \Rightarrow \frac{\partial E_t}{\partial a_2} = -0.217071535$$

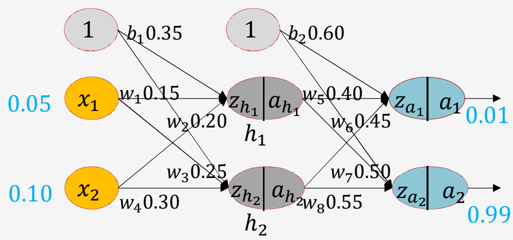
$$\frac{\partial a_2}{\partial z_{a_2}} = a_2(1 - a_2) = 0.772928465(1 - 0.772928465) \Rightarrow \frac{\partial a_2}{\partial z_{a_2}} = 0.175510053$$

$$\frac{\partial z_{a_1}}{\partial b_2} = 1; \frac{\partial z_{a_2}}{\partial b_2} = 1$$

47

1. Retropropagation:

Biais b_2



$$\frac{\partial E_t}{\partial b_2} = \frac{\partial E_t}{\partial a_1} \times \frac{\partial a_1}{\partial z_{a_1}} \times \frac{\partial z_{a_1}}{\partial b_2} + \frac{\partial E_t}{\partial a_2} \times \frac{\partial a_2}{\partial z_{a_2}} \times \frac{\partial z_{a_2}}{\partial b_2}$$

$$\frac{\partial E_t}{\partial b_2} = 0.74136507 \times 0.186815602 + -0.217071535 \times 0.175510053 = -\ 0.005276551$$

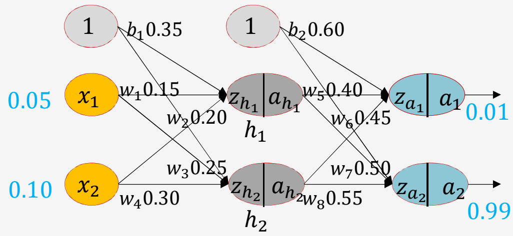
$$b_2^+ = b_2 - \eta \times \frac{\partial E_t}{\partial b_2} = 0.6 + 0.5 \times 0.005276551{=}0.602638275$$

48

1. Retropropagation:

Couche cachée

$$\frac{\partial E_t}{\partial w_1} = \frac{\partial E_t}{\partial a_{h_1}} \times \frac{\partial a_{h_1}}{\partial z_{h_1}} \times \frac{\partial z_{h_1}}{\partial w_1}$$



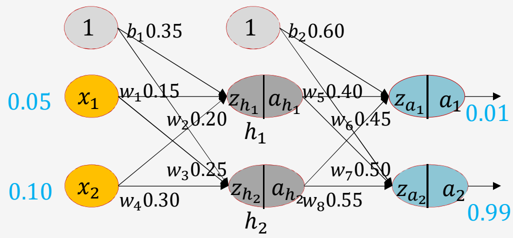
49

1. Retropropagation:

Couche cachée

$$\frac{\partial E_t}{\partial w_1} = \frac{\partial E_t}{\partial a_{h_1}} \times \frac{\partial a_{h_1}}{\partial z_{h_1}} \times \frac{\partial z_{h_1}}{\partial w_1}$$

$$\frac{\partial E_t}{\partial a_{h_1}} = \frac{\partial E_{a_1}}{\partial a_{h_1}} + \frac{\partial E_{a_2}}{\partial a_{h_1}}$$



50

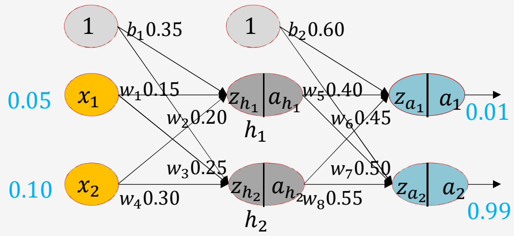
1. Retropropagation:

Couche cachée

$$\frac{\partial E_t}{\partial w_1} = \frac{\partial E_t}{\partial a_{h_1}} \times \frac{\partial a_{h_1}}{\partial z_{h_1}} \times \frac{\partial z_{h_1}}{\partial w_1}$$

$$\frac{\partial E_t}{\partial a_{h_1}} = \frac{\partial E_{a_1}}{\partial a_{h_1}} + \frac{\partial E_{a_2}}{\partial a_{h_1}}$$

$$\frac{\partial E_{a_1}}{\partial a_{h_1}} = \frac{\partial E_{a_1}}{\partial z_{a_1}} \times \frac{\partial z_{a_1}}{\partial a_{h_1}}$$



51

1. Retropropagation:

Couche cachée

$$\frac{\partial E_t}{\partial w_1} = \frac{\partial E_t}{\partial a_{h_1}} \times \frac{\partial a_{h_1}}{\partial z_{h_1}} \times \frac{\partial z_{h_1}}{\partial w_1}$$

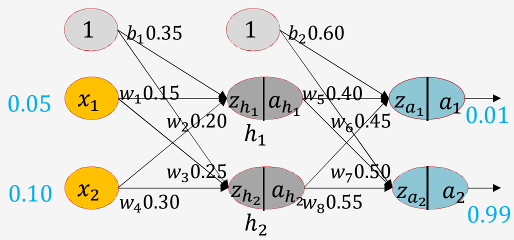
$$\frac{\partial E_t}{\partial a_{h_1}} = \frac{\partial E_{a_1}}{\partial a_{h_1}} + \frac{\partial E_{a_2}}{\partial a_{h_1}}$$

$$\frac{\partial E_{a_1}}{\partial a_{h_1}} = \frac{\partial E_{a_1}}{\partial z_{a_1}} \times \frac{\partial z_{a_1}}{\partial a_{h_1}}$$

$$\frac{\partial E_{a_1}}{\partial z_{a_1}} = \frac{\partial E_{a_1}}{\partial a_1} \times \frac{\partial a_1}{\partial z_{a_1}} = 0.74136507 \times 0.186815602 = 0.138498562$$

$$\frac{\partial z_{a_1}}{\partial a_{h_1}} = \frac{\partial (w_5 a_{h_1} + w_6 a_{h_2} + b_2)}{\partial a_{h_1}} = w_5 = 0.40$$

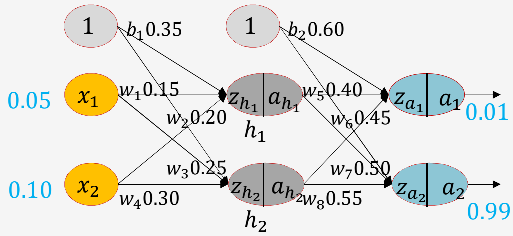
$$\frac{\partial E_{a_1}}{\partial a_{h_1}} = \frac{\partial E_{a_1}}{\partial z_{a_1}} \times \frac{\partial z_{a_1}}{\partial a_{h_1}} = 0.138498562 \times 0.4 = 0.055399425$$



52

1. Retropropagation:

Couche cachée



$$\frac{\partial E_t}{\partial w_1} = \frac{\partial E_t}{\partial a_{h_1}} \times \frac{\partial a_{h_1}}{\partial z_{h_1}} \times \frac{\partial z_{h_1}}{\partial w_1}$$

$$\frac{\partial E_t}{\partial a_{h_1}} = \frac{\partial E_{a_1}}{\partial a_{h_1}} + \frac{\partial E_{a_2}}{\partial a_{h_1}}$$

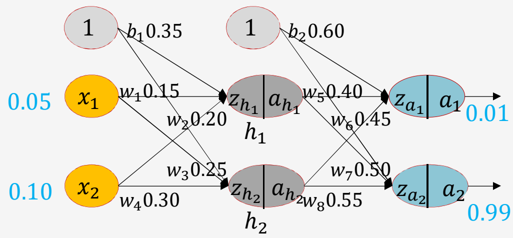
$$\frac{\partial E_{a_1}}{\partial a_{h_1}} = 0.055399425; \quad \frac{\partial E_{a_2}}{\partial a_{h_1}} = -0.019049119$$

$$\frac{\partial E_t}{\partial a_{h_1}} = \frac{\partial E_{a_1}}{\partial a_{h_1}} + \frac{\partial E_{a_2}}{\partial a_{h_1}} = 0.055399425 - 0.019049119 = 0.036350306$$

53

1. Retropropagation:

Couche cachée



$$\frac{\partial E_t}{\partial w_1} = \frac{\partial E_t}{\partial a_{h_1}} \times \frac{\partial a_{h_1}}{\partial z_{h_1}} \times \frac{\partial z_{h_1}}{\partial w_1}$$

$$\frac{\partial E_t}{\partial a_{h_1}} = 0.036350306$$

$$\frac{\partial a_{h_1}}{\partial z_{h_1}} = \frac{\partial\left(\frac{1}{1+e^{-z_{h_1}}}\right)}{\partial z_{h_1}} = a_{h_1}\left(1 - a_{h_1}\right) = 0.593269992(1-0.593269992)=0.241300709$$

$$\frac{\partial z_{h_1}}{\partial w_1} = \frac{\partial(w_1 x_1 + w_2 x_2 + b_1)}{\partial w_1} = x_1 = 0.05$$

$$\frac{\partial E_t}{\partial w_1} = \frac{\partial E_t}{\partial a_{h_1}} \times \frac{\partial a_{h_1}}{\partial z_{h_1}} \times \frac{\partial z_{h_1}}{\partial w_1} = 0.036350306 \times 0.241300709 \times 0.05 = 0.000438568$$

$$w_1^+ = w_1 - \eta \times \frac{\partial E_t}{\partial w_1} = 0.15 - 0.5 \times 0.000438568 = 0.149780716$$

$$w_2^+ = 0.19956143; \quad w_3^+ = 0.249751140; \quad w_4^+ = 0.299502290$$

54

1. Retropropagation:

Biais b_1

$$\frac{\partial E_t}{\partial b_1}=\frac{\partial E_t}{\partial a_1}\times\frac{\partial a_1}{\partial z_{a_1}}\times\frac{\partial z_{a_1}}{\partial a_{h_1}}\times\frac{\partial a_{h_1}}{\partial z_{h_1}}\times\frac{\partial z_{h_1}}{\partial b_1}+\frac{\partial E_t}{\partial a_2}\times\frac{\partial a_2}{\partial z_{a_2}}\times\frac{\partial z_{a_2}}{\partial a_{h_2}}\times\frac{\partial a_{h_2}}{\partial z_{h_2}}\times\frac{\partial z_{h_2}}{\partial b_1}$$

$$\frac{\partial E_t}{\partial b_1}=0.74136507\times 0.186815602\times 0.40+-0.217071535\times 0.175510053\times 0.55$$

$$\frac{\partial E_t}{\partial b_1}=-\ 0.00116084$$

$$b_1^+=\ b_1-\eta\times\frac{\partial E_t}{\partial b_1}=0.35+0.5\times 0.00116084{=}0.350580420$$

55

Initialisation:Epoch0:

Poids

$w_1\ =0.15;$ $w_2\ =0.20;$ $w_3\ =0.25;$ $w_4\ =0.30;$

$w_5\ =0.40;$ $w_6\ =0.45;$ $w_7\ =0.50;$ $w_8\ =0.55;$

Biais

$b_1\ =0.35;$ $b_2\ =0.60;$

Mise à jour: Epoch1

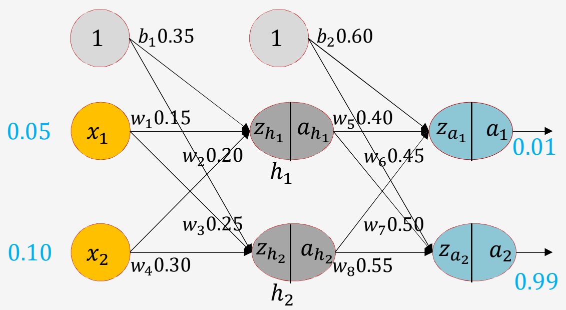
$w_1^+\ =0.149780716;$ $w_2^+\ =0.199561430;$ $w_3^+\ =0.249751140;$ $w_4^+\ =0.299502290;$

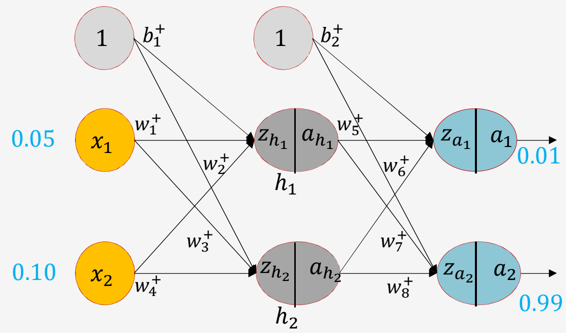
$w_5^+\ =0.358916480;$ $w_6^+\ =0.408666186;$ $w_7^+\ =0.511301270;$ $w_8^+\ =0.561370121;$

$b_1^+\ =0.350580420;$ $b_2^+\ =0.602638275;$

$y_1^+\ =0.015912196;$ $y_2^+\ =0.984065734;$

$E_T^+\ =0.291027924$



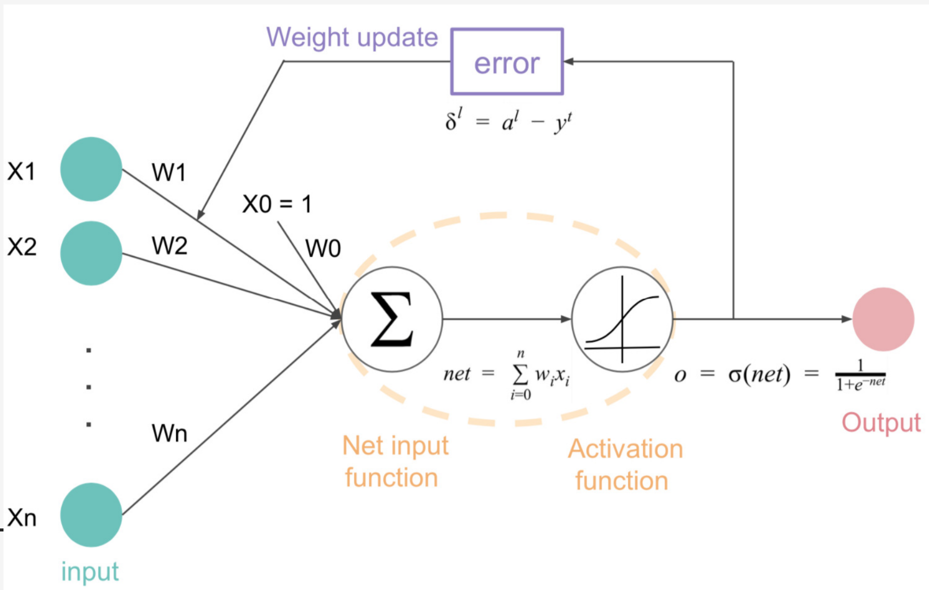


56

Plus de détails

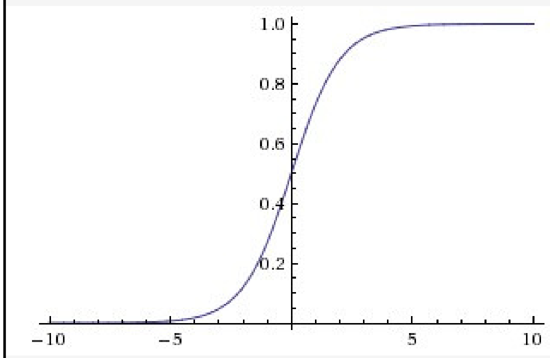
57

Fonctions d'activation



58

Fonction d'activation



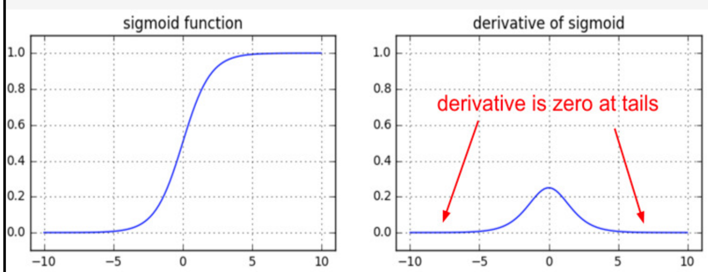
- **Sigmoid:**

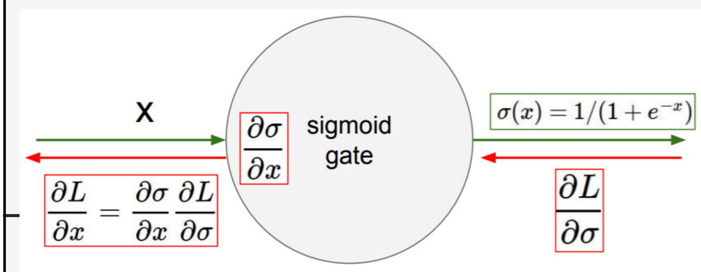
$$f(v) = \frac{1}{1+e^{-v}}$$

- Retourne des valeurs entre [0,1].
- Problèmes avec sigmoid
- => Calcul exp très couteux
- => Gradient vanishing

59

Fonction d'activation

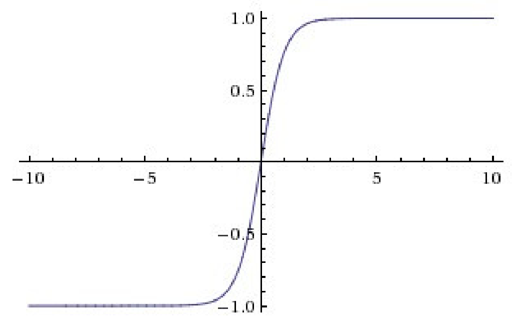




- **Sigmoid:** $f(v) = \frac{1}{1+e^{-v}}$
- Quand v est très grand ou très petits,
- Sa dérivée tend vers **0.**
- Lors de la propagation avec la règle de la chaine, vers la première couche, on multiplie par des valeurs très faibles,
- ➔ le réseau n'apprend plus, càd pas de mise à jour des poids.

60

Fonction d'activation



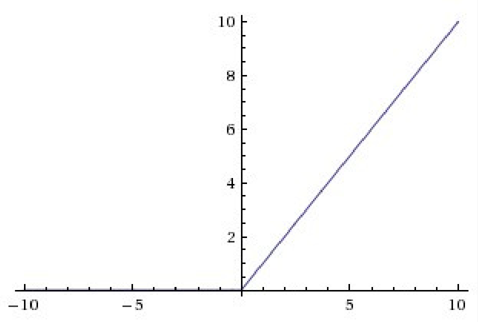
- **Tanh**

$$f(v) = 2\sigma(2x) - 1$$

- Retourne des valeurs entre [-1,1].
- Variante de la sigmoid.
- Même problème de gradient vanishing

61

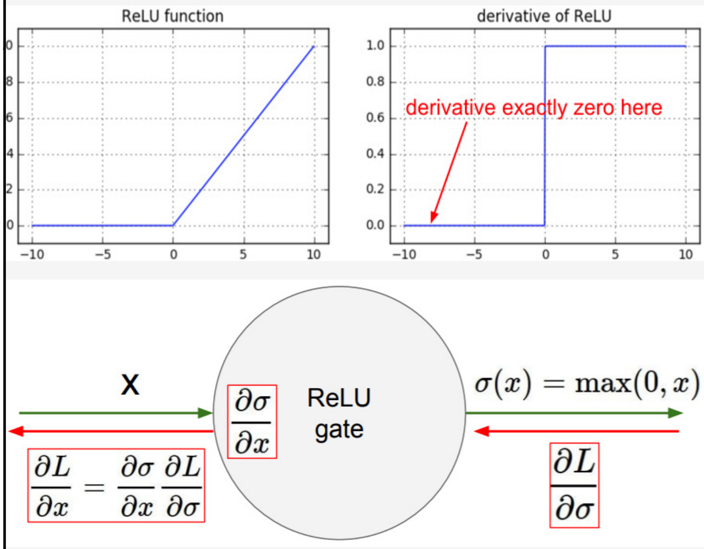
Fonction d'activation



- **ReLu** $f(v) = max(v, 0)$
- (Rectified Linear Unit)
- Largement utilisée.
- Pas de pbm de vanishing pour des v positif.
- Accélère la convergence du réseau par rapport à sigmoid/tanh (6fois plus rapide)
- Simple à calculer
- Problème:
- => 'Dead gradient': gradient égal zéro pour v <0

62

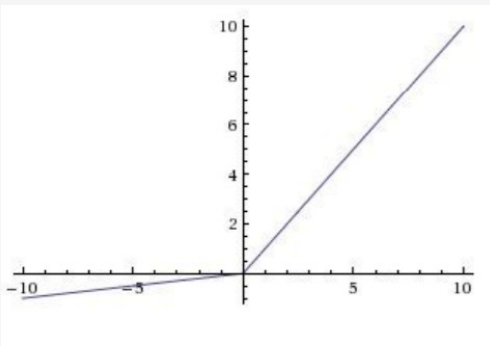
Fonction d'activation



- ReLu
  $\boldsymbol{f(v) = max(v, 0)}$
- Si $\boldsymbol{v}$ >0 => neurone activé, son gradient = 1, il est possible de faire la rétropropagation.
- Si $\boldsymbol{v}$ <0 => neurone non-activé, son gradient = 0, pas possible de mettre à jour les poids.

63

Fonction d'activation



- LeakyReLu

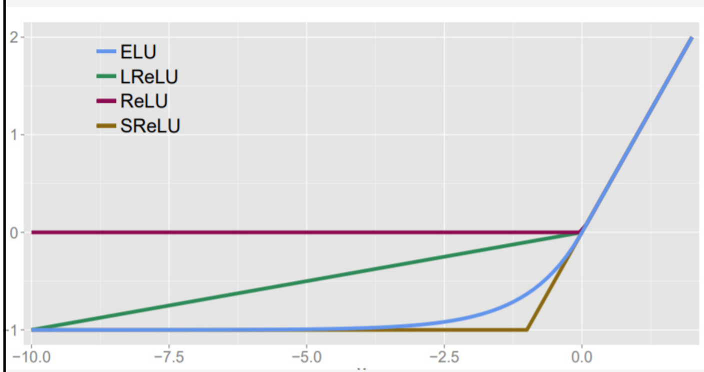
$$\boldsymbol{f(v) = max(v, 0.01v)}$$

- Pas de pbm de vanishing pour des $\boldsymbol{v}$ positif.
- Accélère la convergence du réseau par rapport à sigmoid/tanh (6fois plus rapide)
- Simple à calculer
- Pas de pbm de 'dead gradient'

- Autre variante: Parametric Rectifier (PRelu)
  $\boldsymbol{f(v) = max(v, \alpha v)}$ | $\boldsymbol{\alpha}$ ***est un paramètre***

64

Fonction d'activation



- ELu

$$f(v) = \begin{cases} v & si\ v > 0 \\ \alpha(e^v - 1) & si\ v \leq 0 \end{cases}$$

- Pas de pbm de vanishing pour des v positif.
- Accélère la convergence du réseau par rapport à sigmoid/tanh (6fois plus rapide)
- Simple à calculer
- Pas de pbm de 'dead gradient'
- Calcul exponentiel!

65

Fonction d'activation

Exemple numérique:

$$y_i = \begin{bmatrix} 2.0 \\ 1.0 \\ 0.1 \end{bmatrix} \Rightarrow f(y_i) = \frac{e^{y_i}}{\sum_k e^{y_k}} \Rightarrow p_i = \begin{bmatrix} 0.7 \\ 0.2 \\ 0.1 \end{bmatrix}$$

- $k = 3\ (classes)$
- $p_1 = \frac{e^{2.0}}{e^{2.0}+e^{1.0}+e^{0.1}} = 0.7$
- $p_2 = \frac{e^{1.0}}{e^{2.0}+e^{1.0}+e^{0.1}} = 0.2$
- $p_3 = \frac{e^{0.1}}{e^{2.0}+e^{1.0}+e^{0.1}} = 0.1$

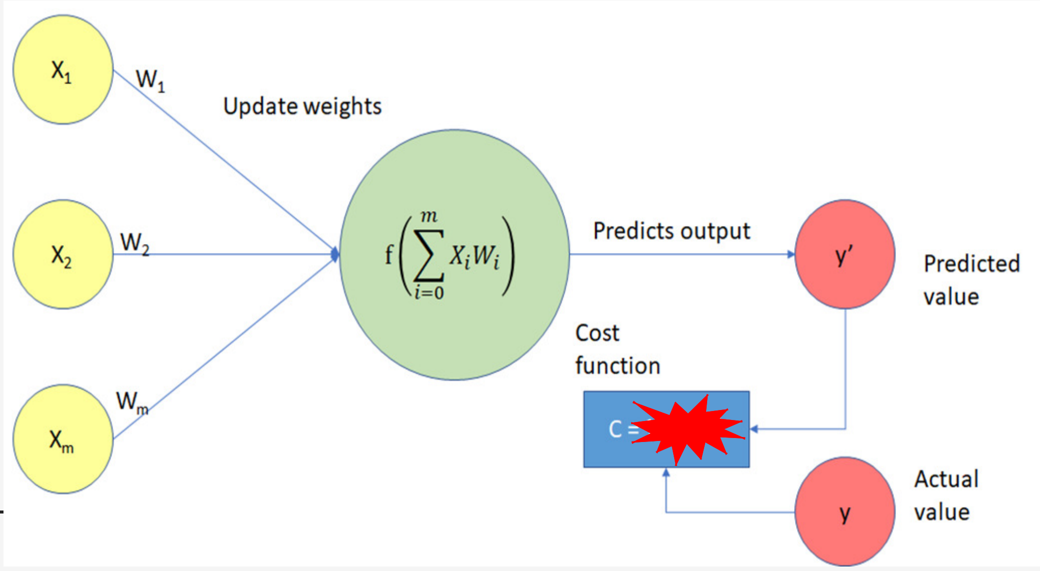
- Softmax

$$f(y_i) = \frac{e^{y_i}}{\sum_k e^{y_k}}$$

- Contexte de classification multi-classes
- Retourne des valeurs de probabilité.
- La somme des valeurs=1.
- La valeur max est celle de la classe prédite

66

Fonctions cout



67

Fonction cout:

- Régression
- Approximer une valeur numérique (e.g. prix d'un produit)
- **Mean Squared Error (MSE)**

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Classification Binaire
- Prédire une catégorie binaire (1/0) (e.g. spam emails)
- **Binary Cross Entropy**

$$(\boldsymbol{ylog}(\hat{\boldsymbol{y}}) + (\mathbf{1} - \boldsymbol{y})\boldsymbol{log}(\mathbf{1} - \hat{\boldsymbol{y}}))$$

- Classification multi-classes (un seul label)
- Prédire un seul label à partir de plusieurs classes (e.g. classification d'objets)
- **Categorical cross-entropy**

$$\sum_{i=1}^{K} \boldsymbol{y_i log}(\hat{\boldsymbol{y}}_i)$$

- Classification multi-classes (multi-labels)
- Prédire plusieurs labels à partir de plusieurs classes (e.g. la présence d'un animal dans l'image)
- **Binary Cross Entropy**

$$(\boldsymbol{ylog}(\hat{\boldsymbol{y}}) + (\mathbf{1} - \boldsymbol{y})\boldsymbol{log}(\mathbf{1} - \hat{\boldsymbol{y}}))$$

68

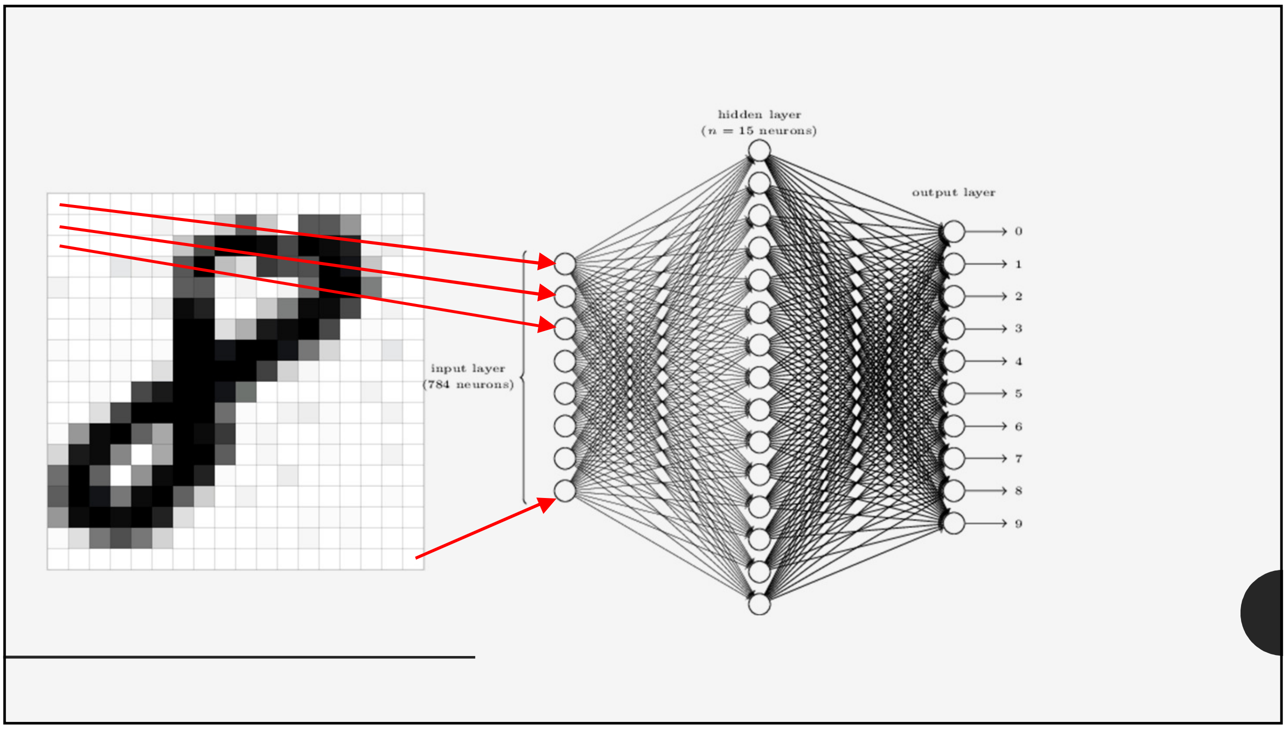
Quelle fonction d'activation (en dernière couche) et fonction cout choisir?

Problem Type	Output Type	Final Activation Function	Loss Function
Regression	Numerical value	Linear	Mean Squared Error (MSE)
Classification	Binary outcome	Sigmoid	Binary Cross Entropy
Classification	Single label, multiple classes	Softmax	Cross Entropy
Classification	Multiple labels, multiple classes	Sigmoid	Binary Cross Entropy

69

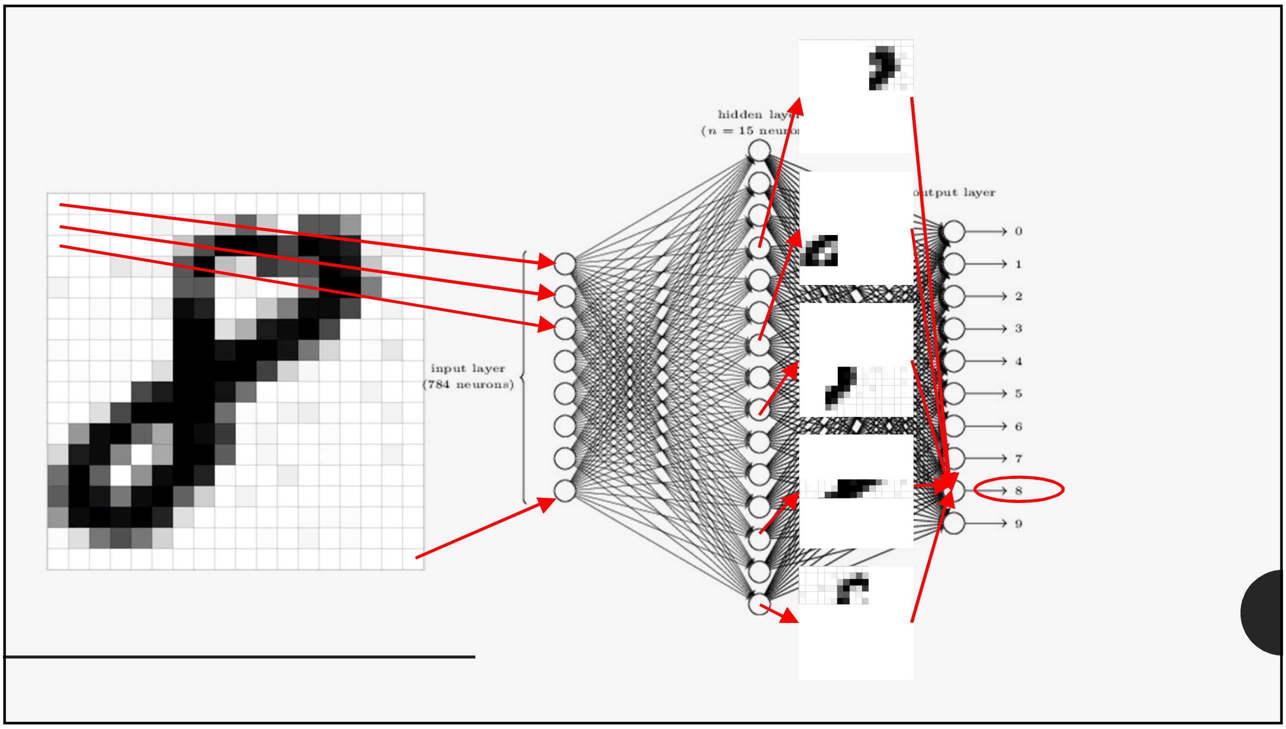
Les ANN pour apprendre des motifs (pattern)

70



hidden layer
(n = 15 neurons)
output layer
input layer
(784 neurons)
0
1
2
3
4
5
6
7
8
9

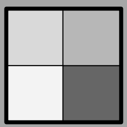
71



input layer
(784 neurons)
0
1
2
3
4
5
6
7
8
9

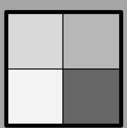
72

Une image de 4 pixels



73

Classifier l’image



solid

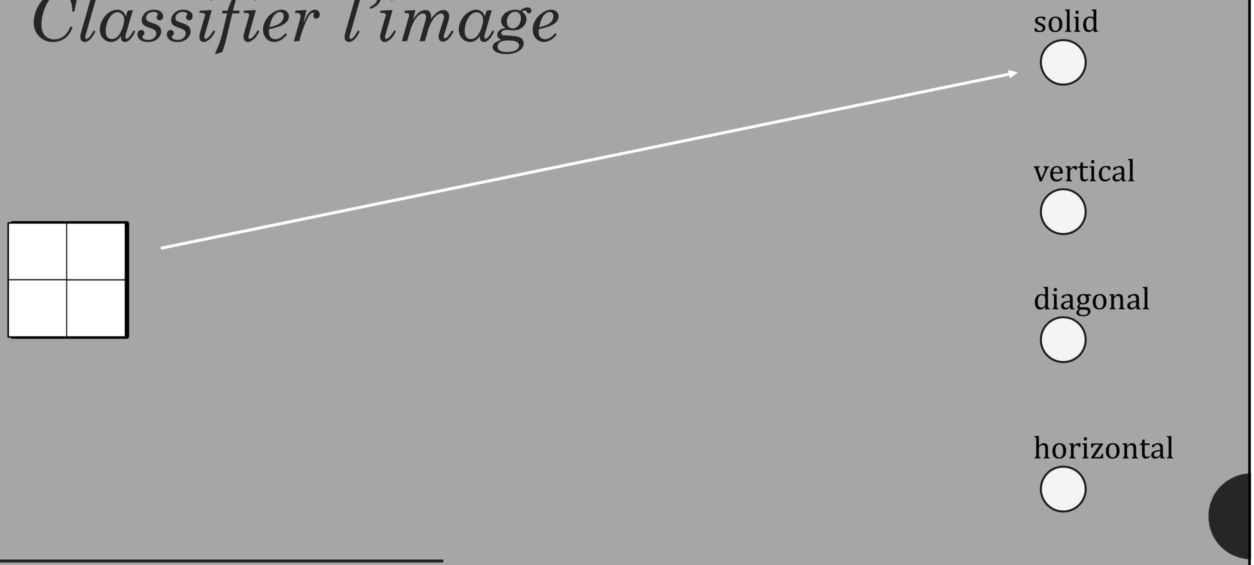
vertical

diagonal

horizontal

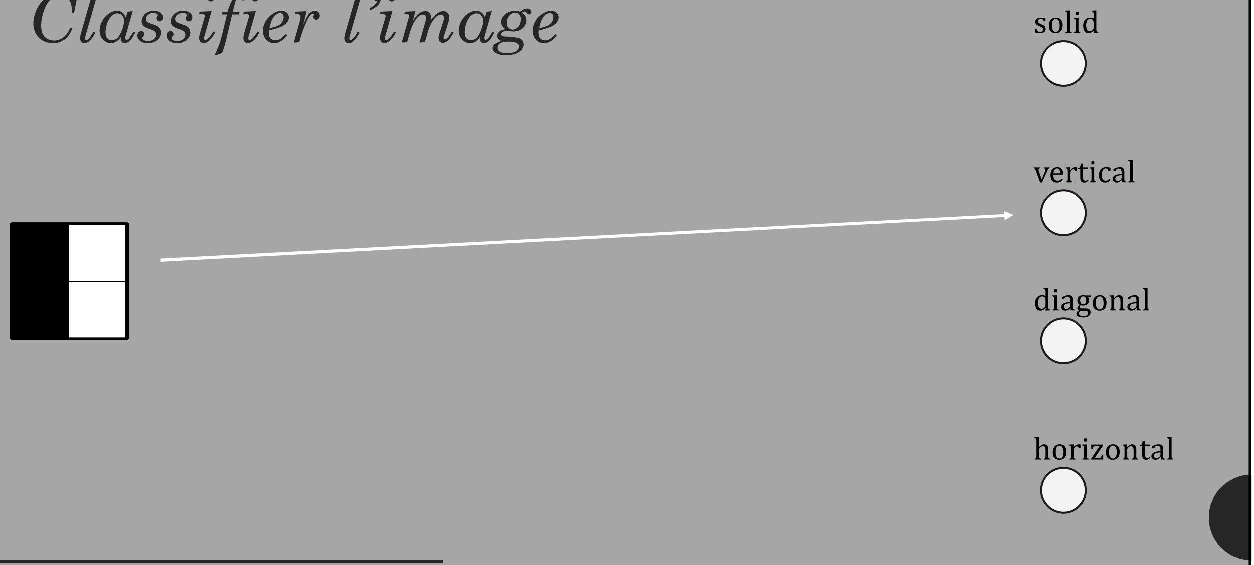
74

Classifier l’image



75

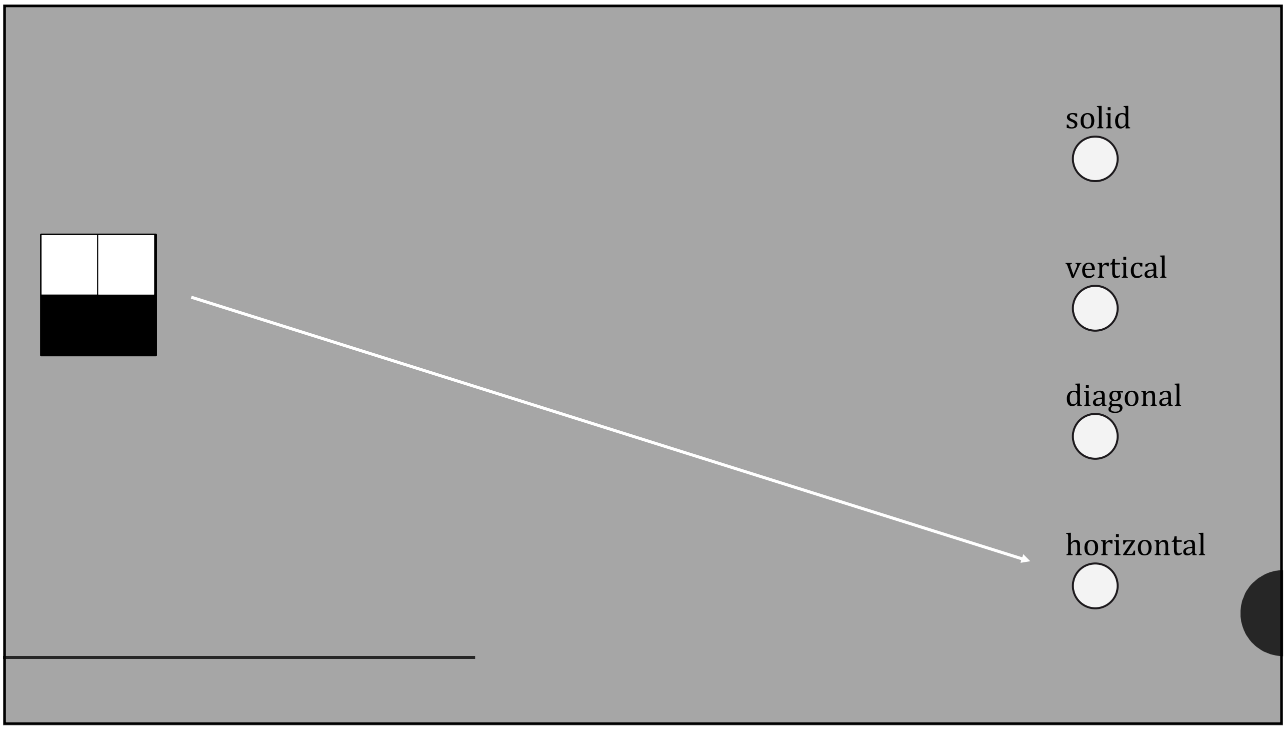
Classifier l’image



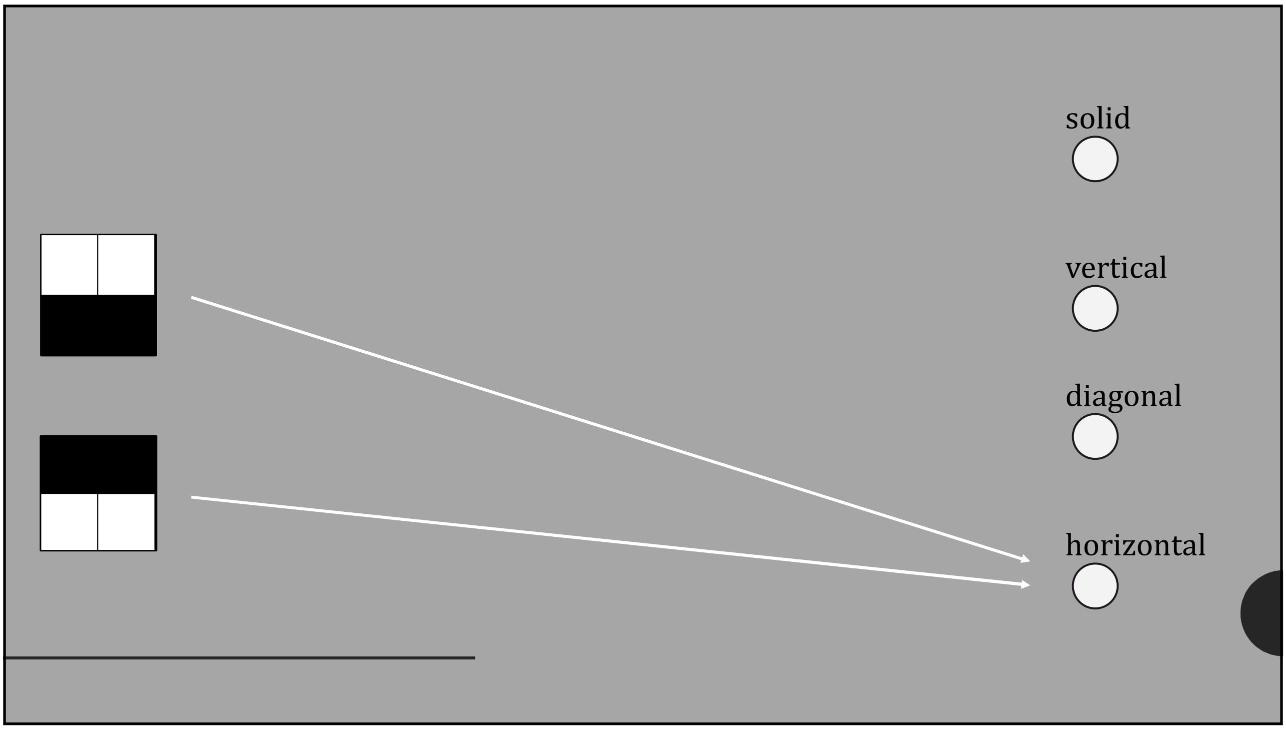
76


Classifier l'image
solid
vertical
diagonal
horizontal

77


solid
vertical
diagonal
horizontal

78

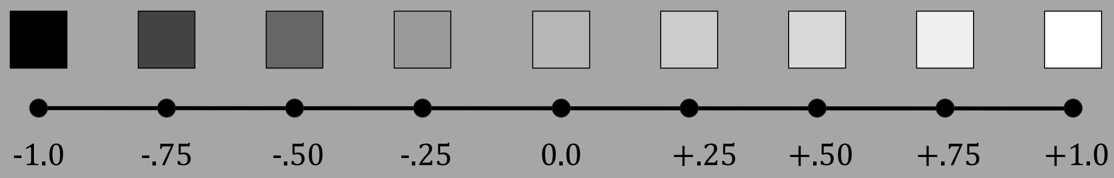


79

Neurones d'entrée

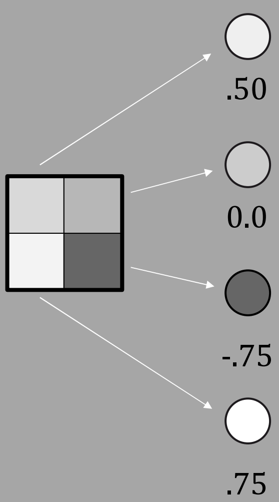
80

Intensité des pixels



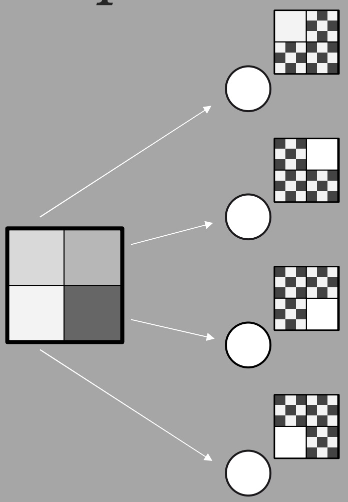
81

Vecteur d'entrée

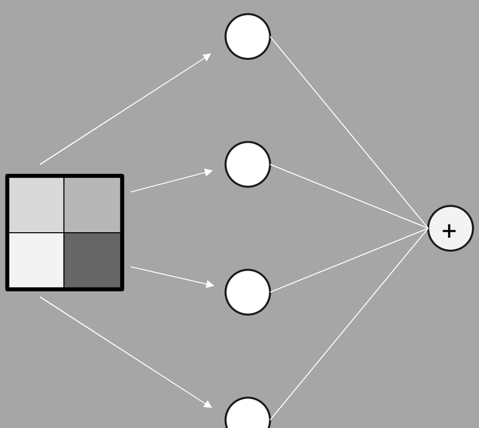


82

Réponses

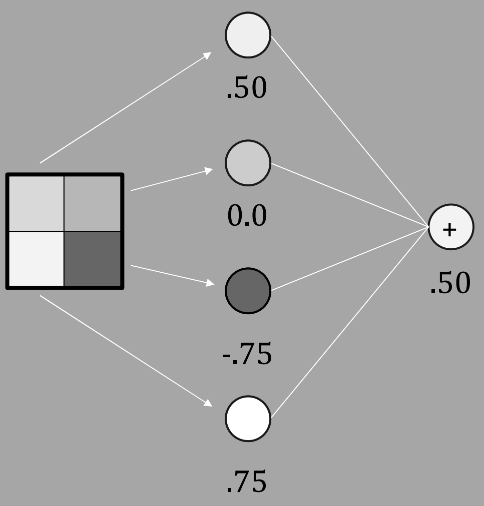


83



84

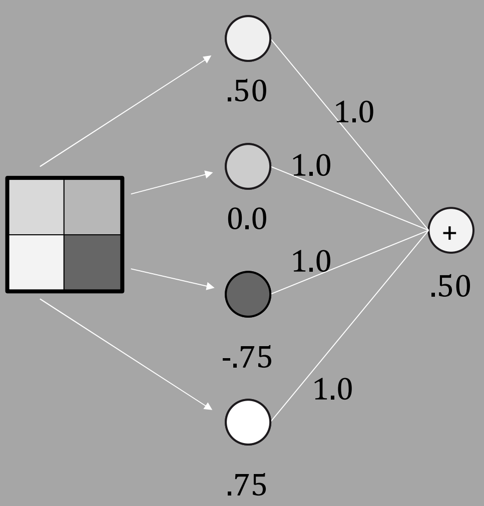
Somme des entrées



$$
\begin{array}{r}
.50 \\
0.00 \\
-.75 \\
+ \quad .75 \\
\hline
.50
\end{array}
$$

85

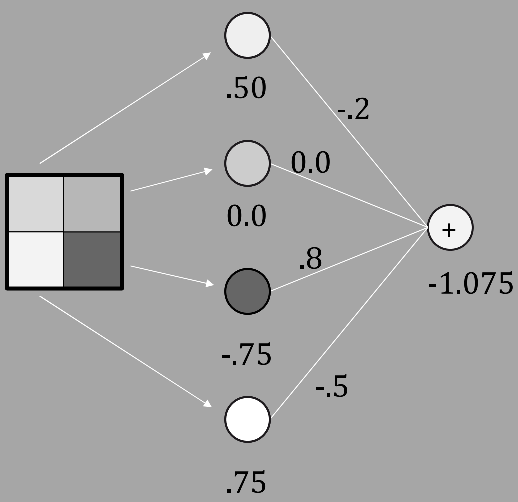
Les poids



$$
\begin{array}{r}
.50 \text{ x } 1.0 \\
0.00 \text{ x } 1.0 \\
-.75 \text{ x } 1.0 \\
+ \quad .75 \text{ x } 1.0 \\
\hline
.50
\end{array}
$$

86

Les poids

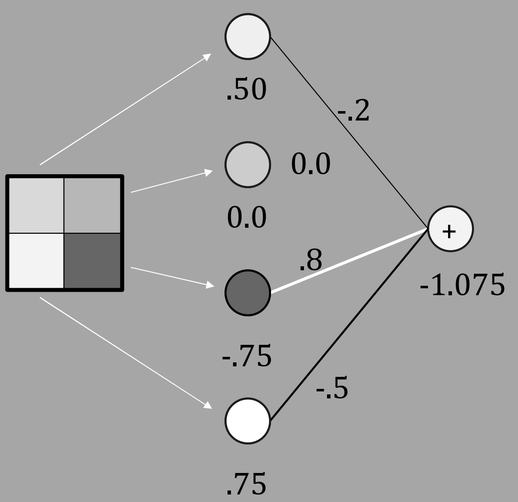


.50 x -.2
0.00 x0.0
-.75 x .8
+ .75 x -.5

-1.075

87

Les poids

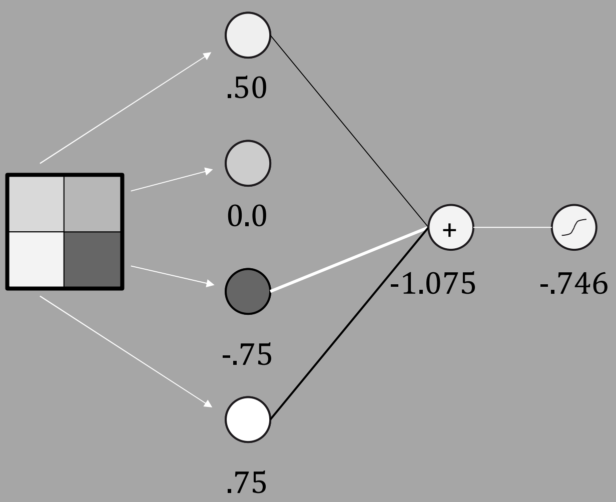


.50 x -.2
0.00 x0.0
-.75 x .8
+ .75 x -.5

-1.075

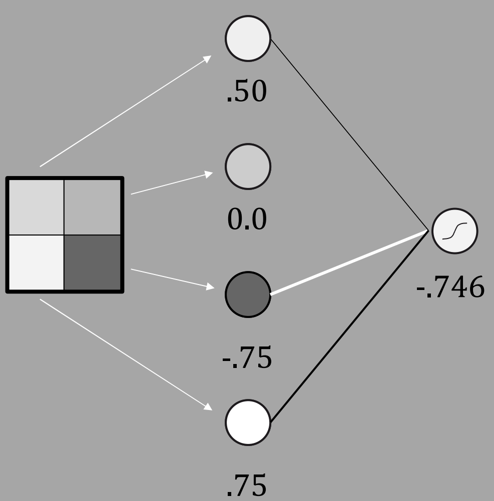
88

Appliquer l'activation sigmoid



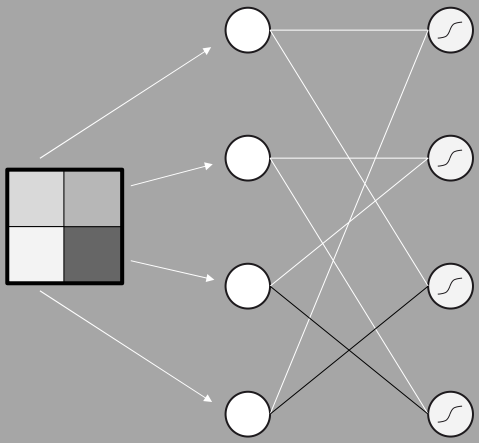
89

Activation du neurone



90

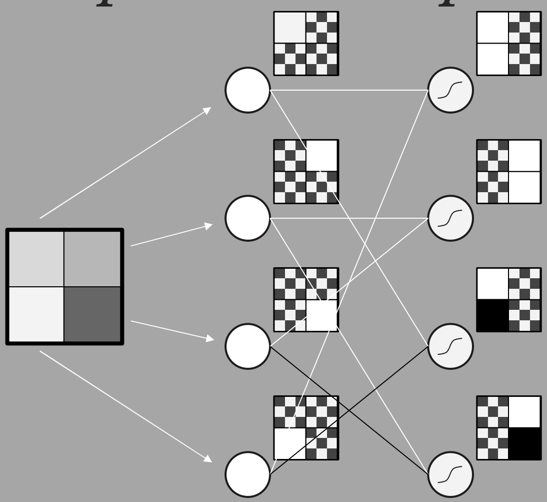
Ajouter d'autres neurones



Poids:
+1.0 en blanc
-1.0 en noir

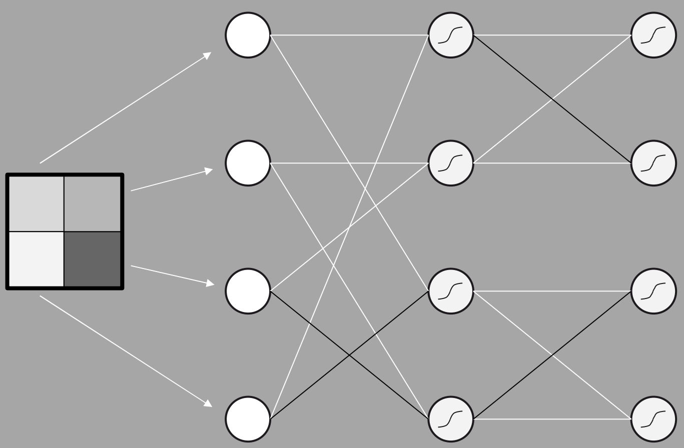
91

Réponses complexes



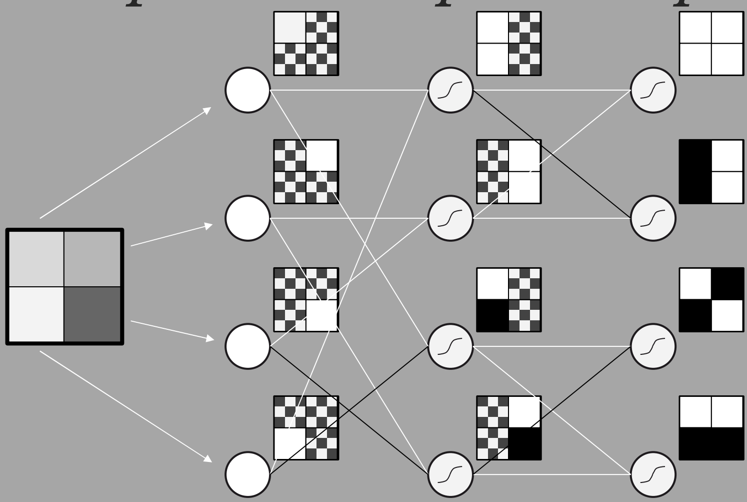
92

Ajouter une autre couche cachée



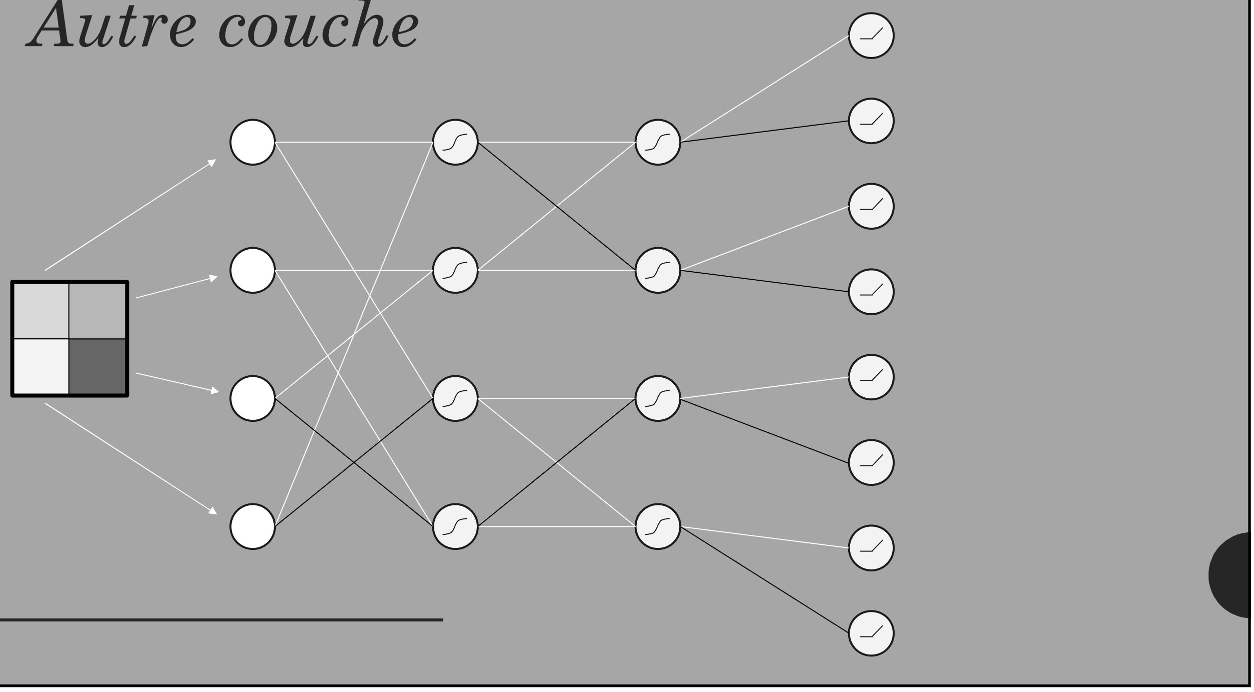
93

Résponses de plus en plus complexes

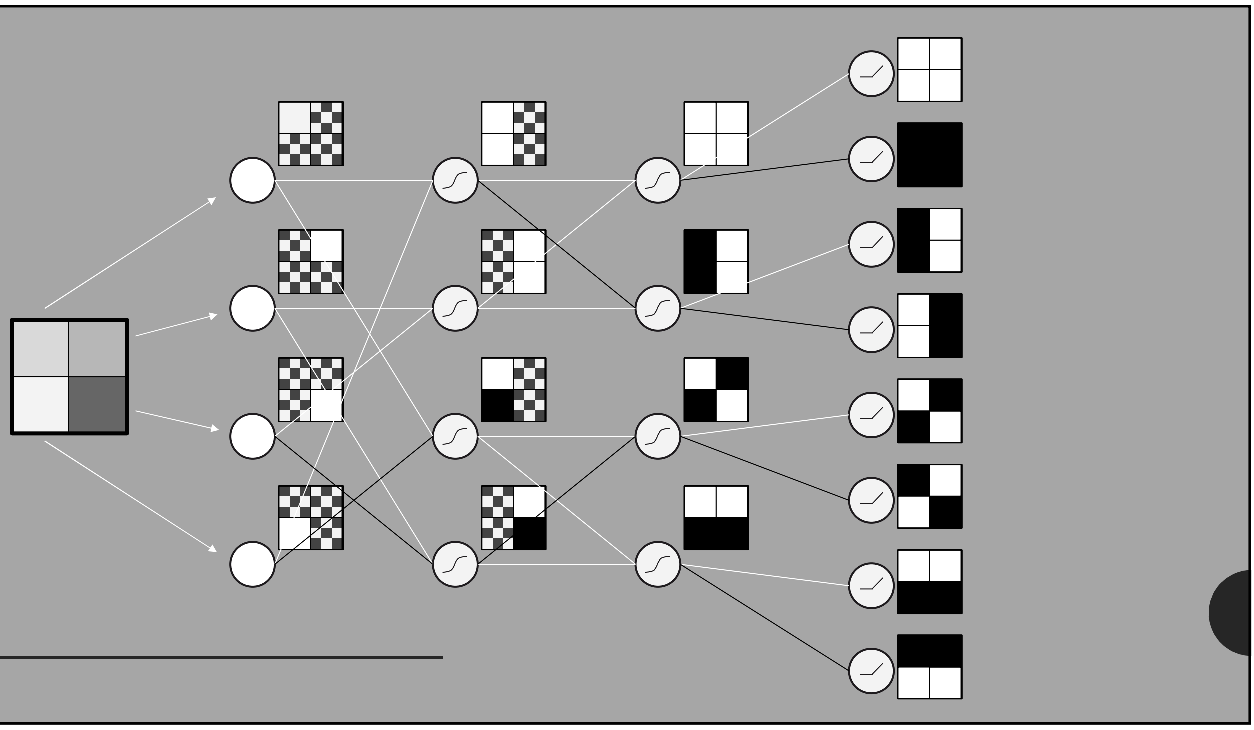


94

Autre couche

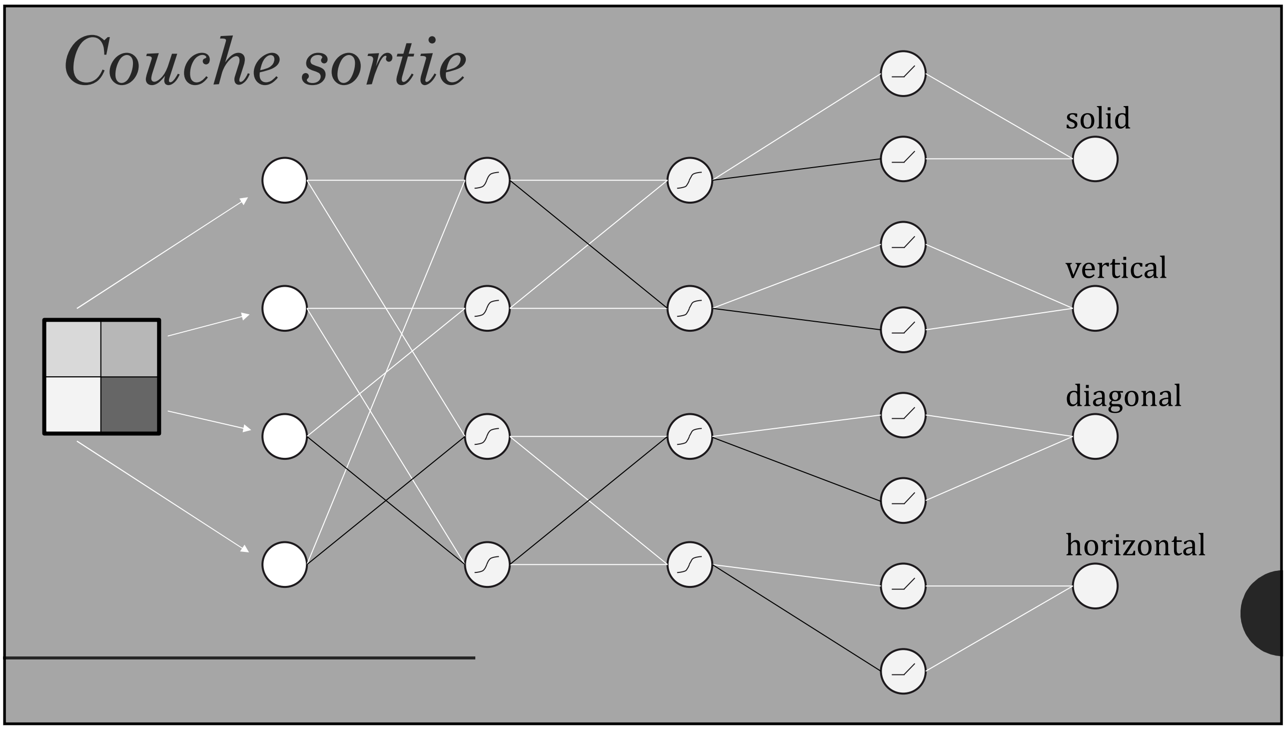


95

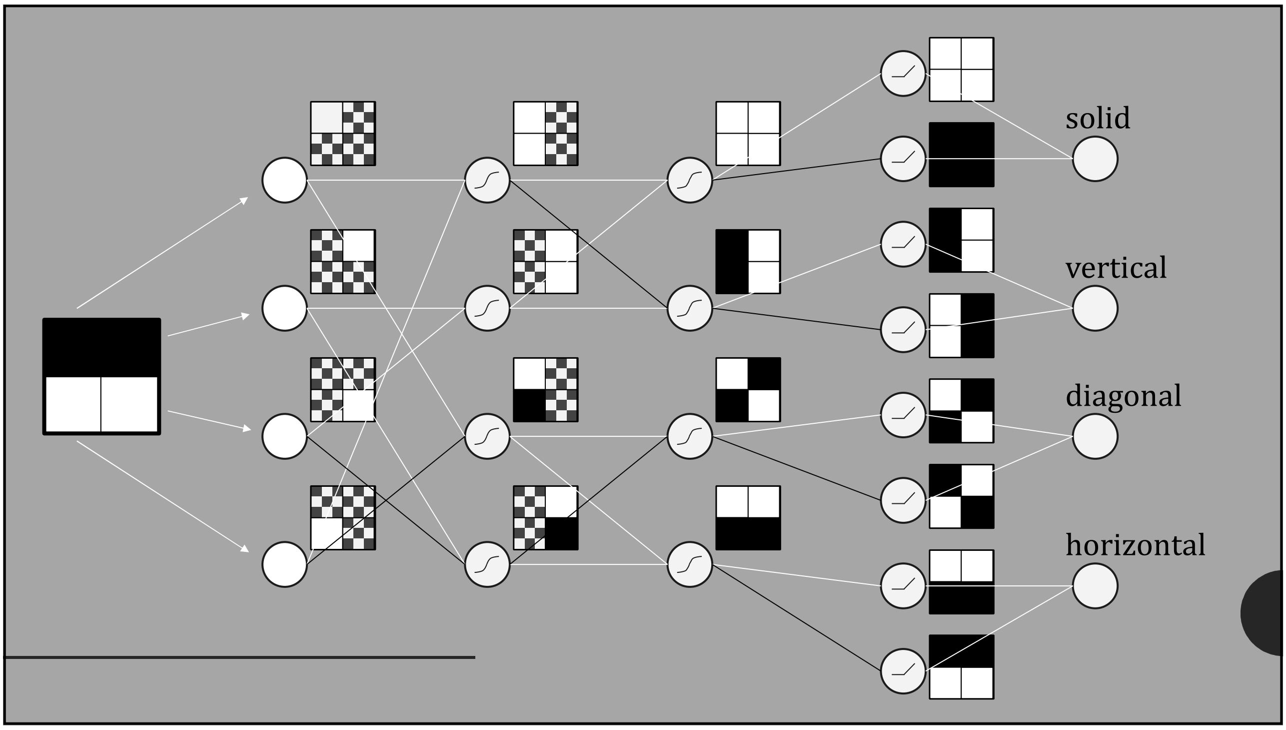


96

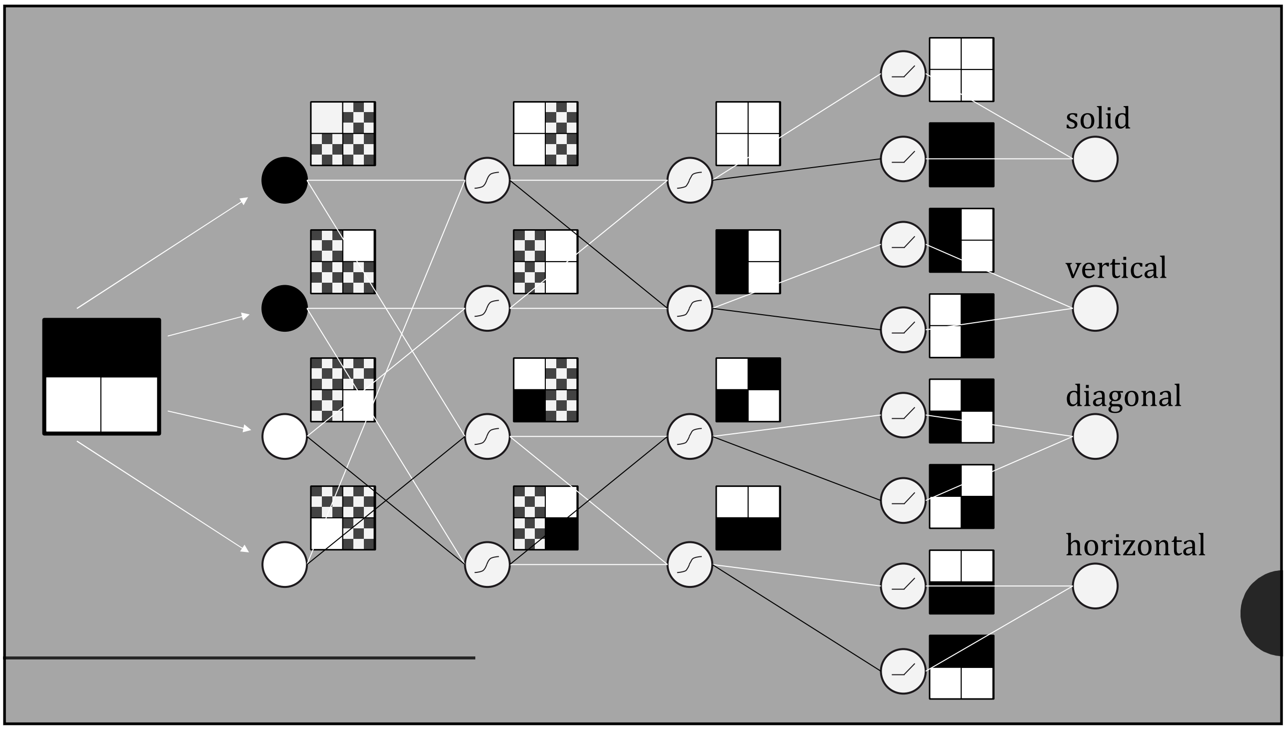
Couche sortie



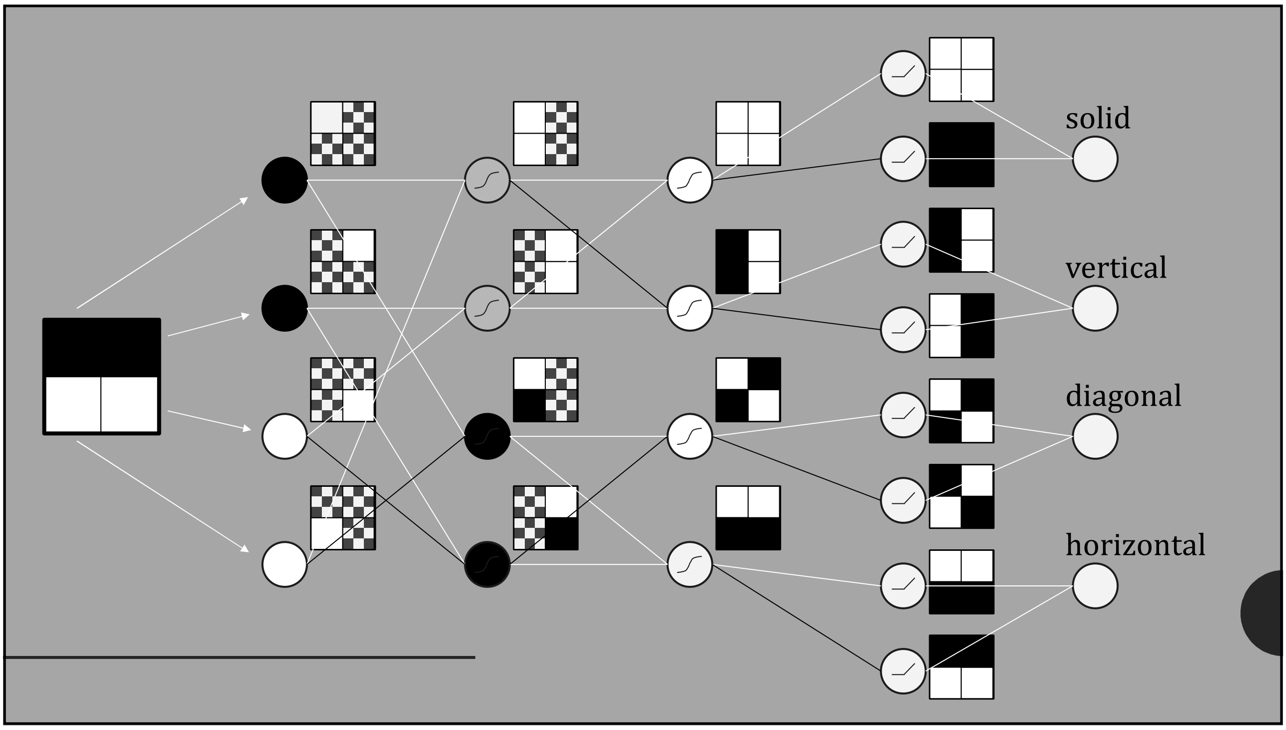
97



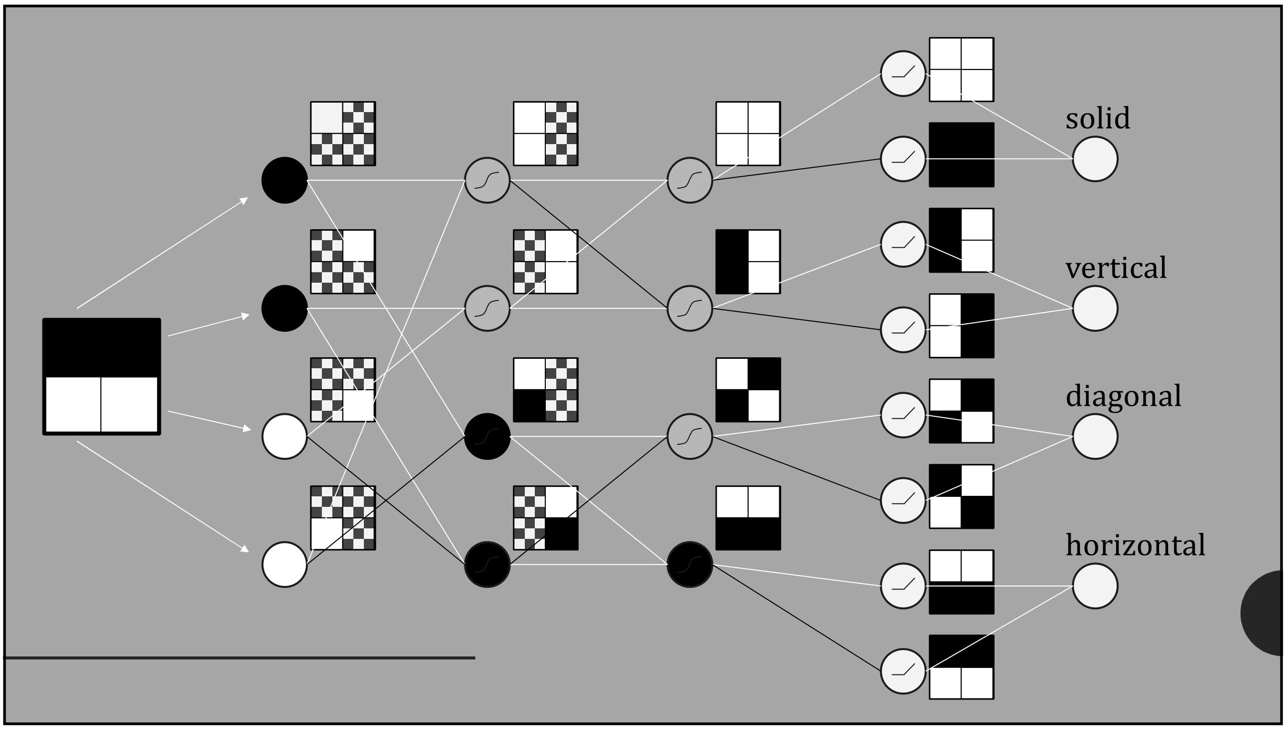
98


solid
vertical
diagonal
horizontal

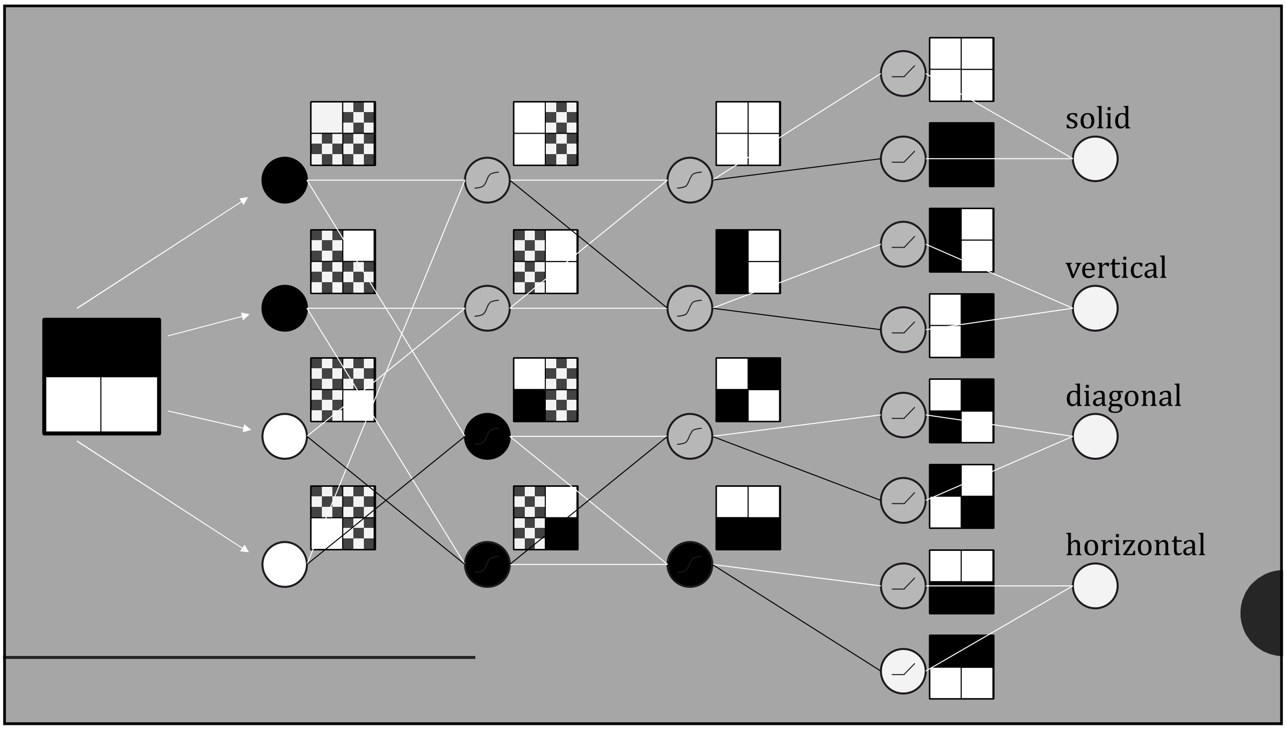
99


solid
vertical
diagonal
horizontal

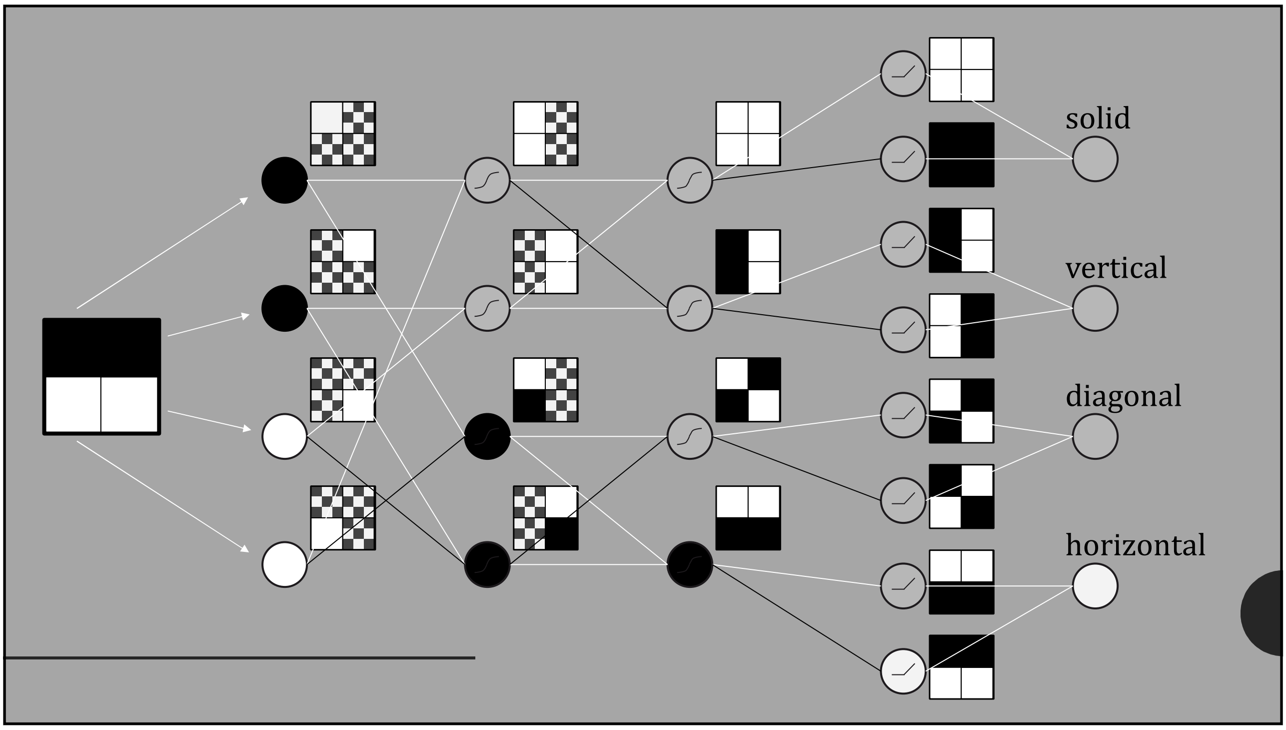
100


solid
vertical
diagonal
horizontal

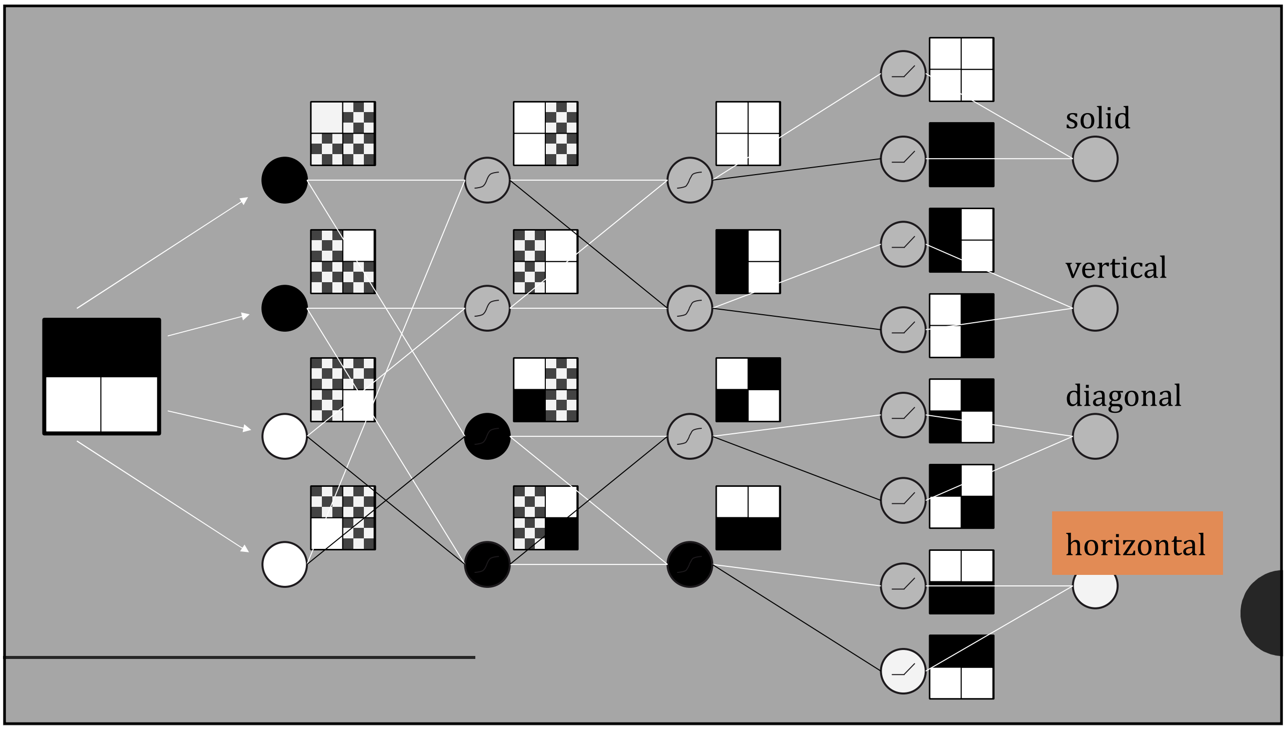
101


solid
vertical
diagonal
horizontal

102

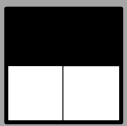

solid
vertical
diagonal
horizontal

103


solid
vertical
diagonal
horizontal

104

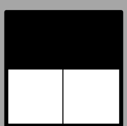
Calcul d'érreur



réelle	
0.	solid
0.	vertical
0.	diagonal
1.	horizontal

105

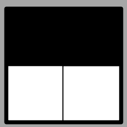
Calcul d'érreur



réelle	prédite	
0.	.5	solid
0.	.75	vertical
0.	-.25	diagonal
1.	-.75	horizontal

106

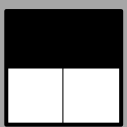
Calcul d’érreur



erreur	réelle	prédite	
.5	0.	.5	solid
.75	0.	.75	vertical
.25	0.	-.25	diagonal
1.75	1.	-.75	horizontal

107

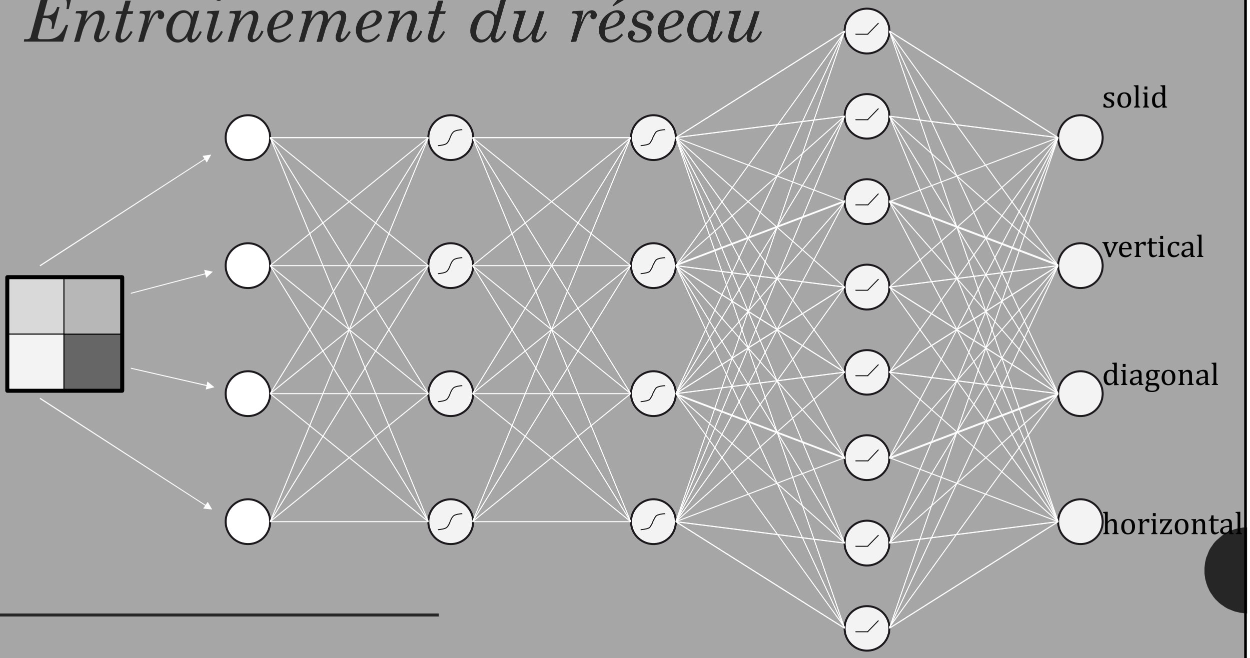
Calcul d’érreur



	erreur	réelle	prédite	
	.5	0.	.5	solid
	.75	0.	.75	vertical
	.25	0.	-.25	diagonal
	1.75	1.	-.75	horizontal
total	3.25			

108

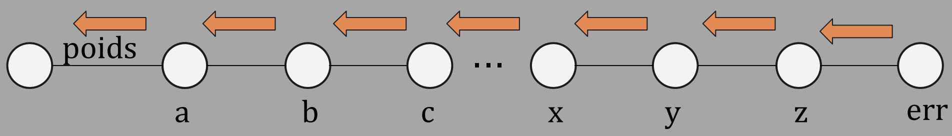
Entrainement du réseau



109

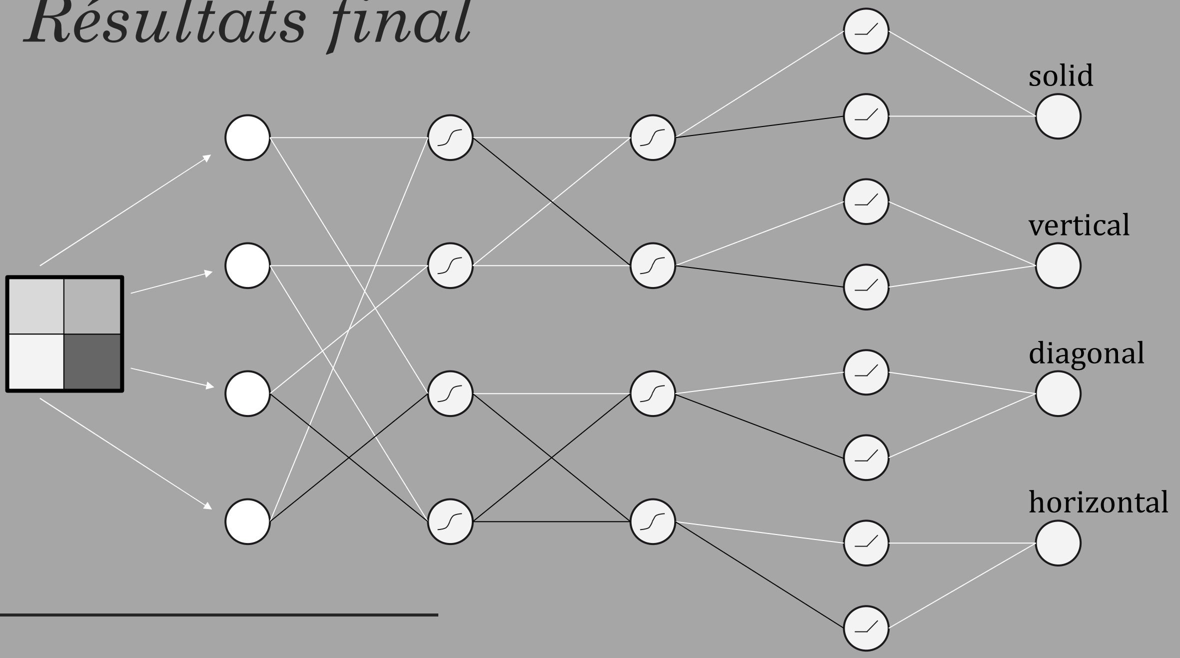
Rétropropagation

$$\frac{\partial err}{\partial poids} = \frac{\partial a}{\partial poids} * \frac{\partial b}{\partial a} * \frac{\partial c}{\partial b} * \frac{\partial d}{\partial c} * \ldots * \frac{\partial y}{\partial x} * \frac{\partial z}{\partial y} * \frac{\partial err}{\partial z}$$



110

Résultats final



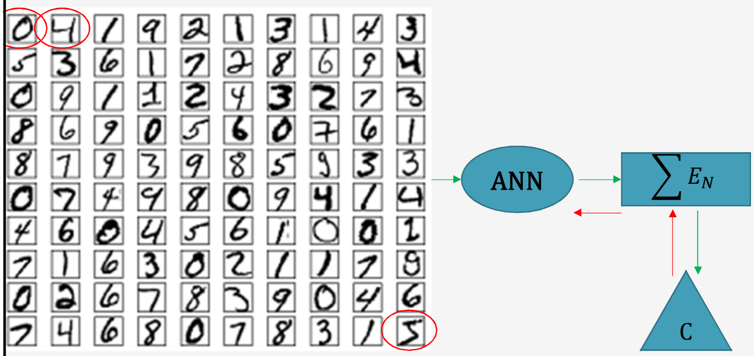
111

De façon générale

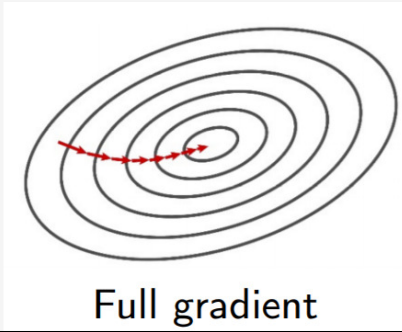
- On a une ensemble de données d'apprentissage
- e.g. la base MNIST: image de 28×28pixels
- On définit un ANN
- Problème de classification multi-classes (10 classes de chiffres à prédire 0,1,...9)

112

Variantes du gradient descent

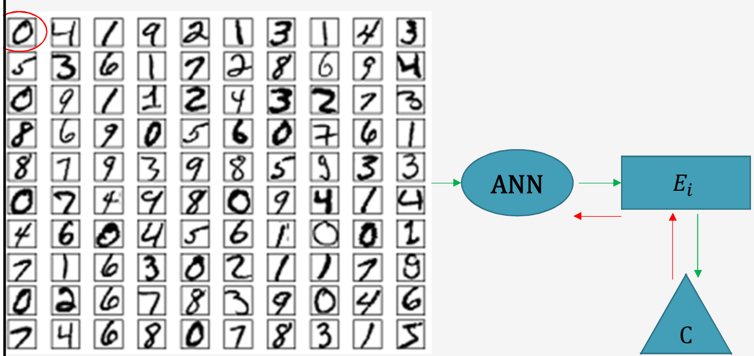


- **Full gradient**
- On applique la màj des poids/biais après avoir passer TOUT l'ensemble des données d'apprentissage.

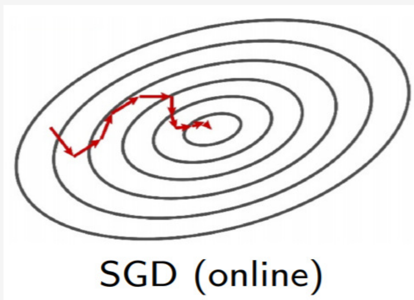


113

Variantes du gradient descent

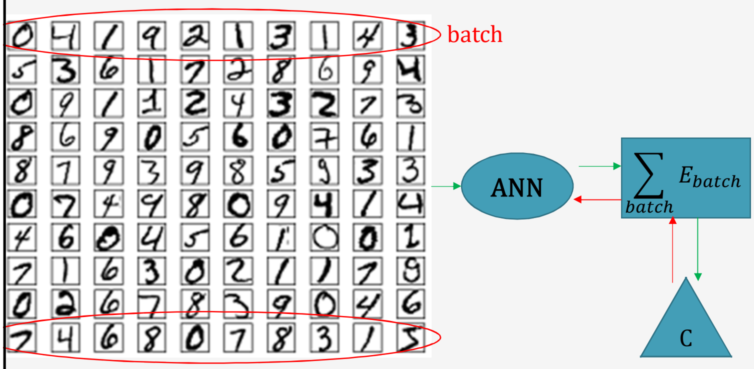


- **Stochastic gradient descent (SGD)(online)**
- On applique la màj des poids/biais après avoir passer CHAQUE donnée d'apprentissage

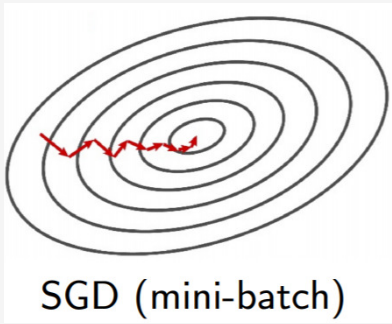


114

Variantes du gradient descent



- **Stochastic gradient descent (SGD) (mini-batch)**
- On applique la màj des poids/biais après avoir passer UN LOT de données d'apprentissage



115

Références

- https://medium.com/@pdquant/all-the-Rétropropagation-derivatives-d5275f727f60
- https://www.anotsorandomwalk.com/Rétropropagation-example-with-numbers-step-by-step/
- https://mattmazur.com/2015/03/17/a-step-by-step-Rétropropagation-example/
- https://google-developers.appspot.com/machine-learning/crash-course/backprop-scroll/
- https://bit.ly/35rYWeI
- https://towardsdatascience.com/deep-learning-which-loss-and-activation-functions-should-i-use-ac02f1c56aa8

116