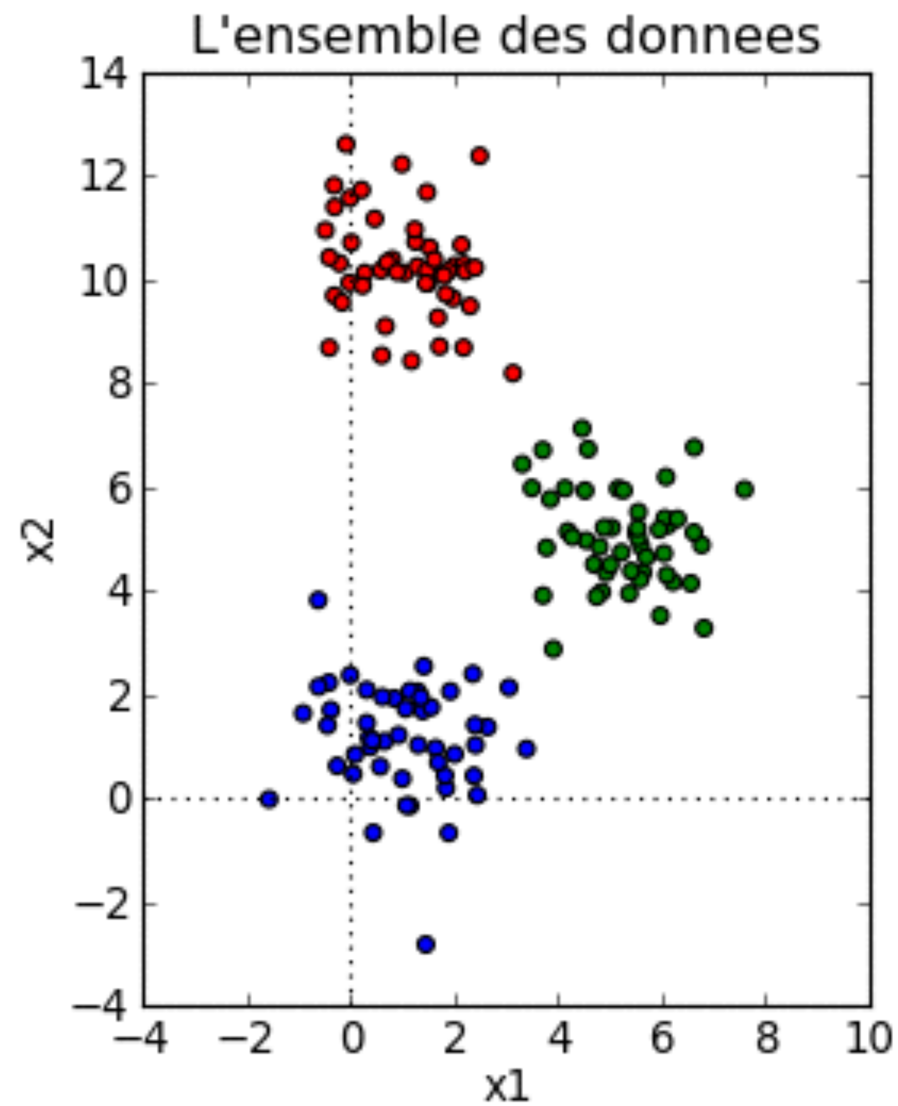
4I702 : TME 2

Apprentissage supervisé

Créer un ensemble de données en Python

- Nombre total des motifs : N
- Plusieurs “nuages” des points (par ex. gaussiens)
- $data\_x1 = \left[x_1^{(1)}, \ldots, x_1^{(N)} \right]$ tableau des abscisses de tous les motifs
- $data\_x2 = \left[x_2^{(1)}, \ldots, x_2^{(N)} \right]$ tableau des ordonnées de tous les motifs
- $labels = \left['r', 'g' \ldots, 'b' \right]$ liste des étiquettes (couleurs) de tous les motifs

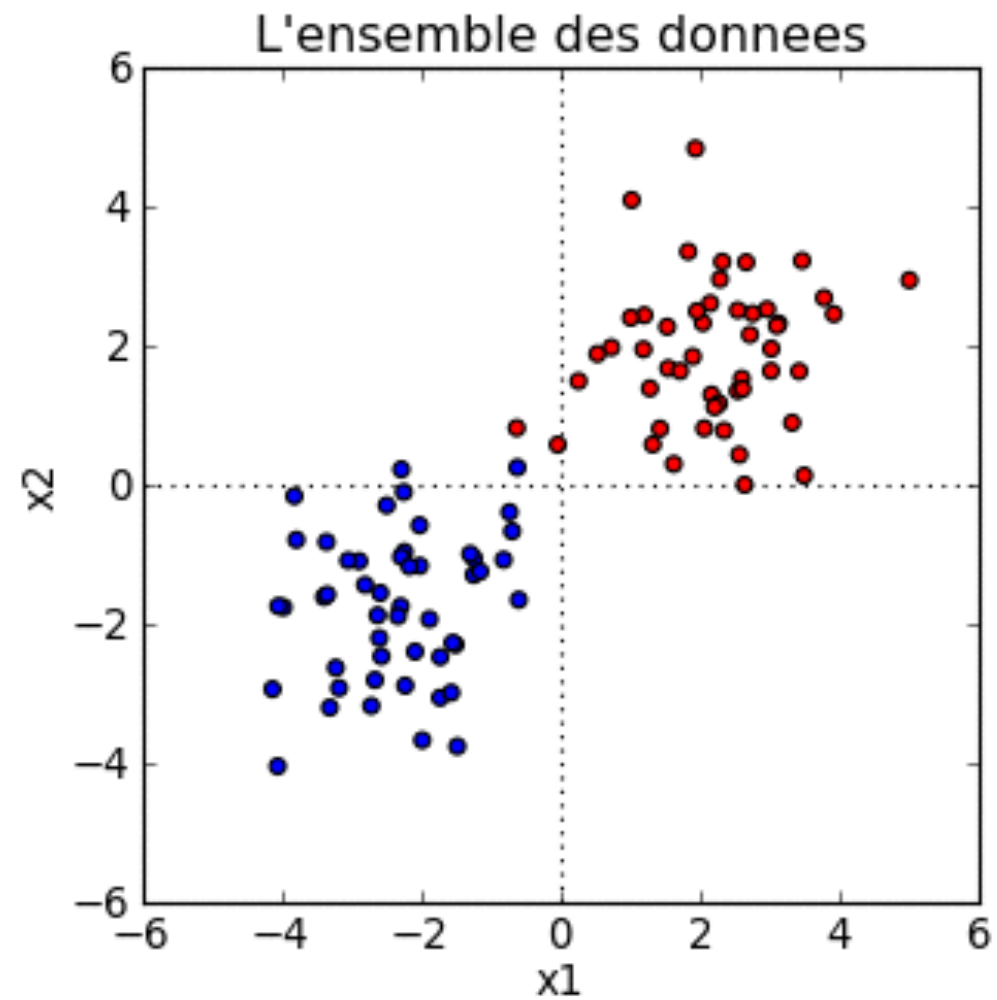


donnees.py

Exercice :

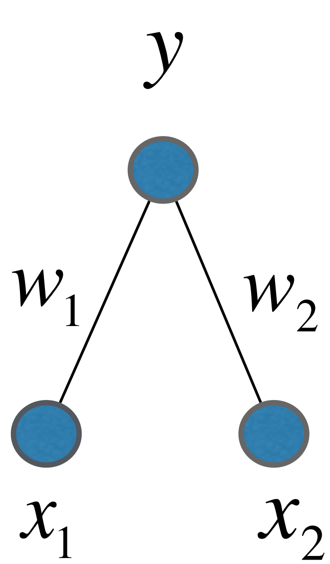
Écrire un script pour crées un ensemble composé de deux nuages gaussiens :

- N=100
- Classe 1 de taille N/2, étiquette ‘r’
  la moyenne **m1 = (2, 2)** et
  l’écart-type **s1 = 1**
- Classe 2, taille N/2, étiquette ‘b’
  la moyenne **m2 = (-2, -2)**,
  l’écart-type **s2 = 1**



Réseau à une couche : classification sans apprentissage

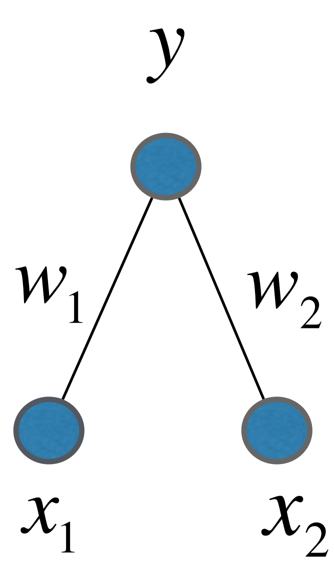
ex_classif.py



- Ajouter la génération de données (lignes 7-13)
- Ajouter l'initialisation aléatoire du vecteur des poids (ligne 19). Par exemple $\vec{w} = (1, -1)$

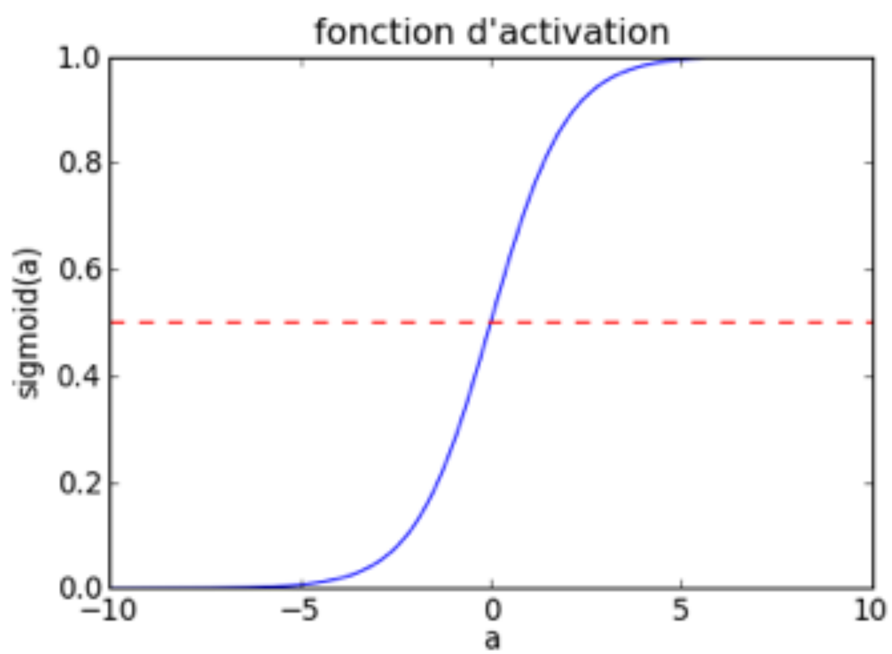
Réseau à une couche : classification sans apprentissage

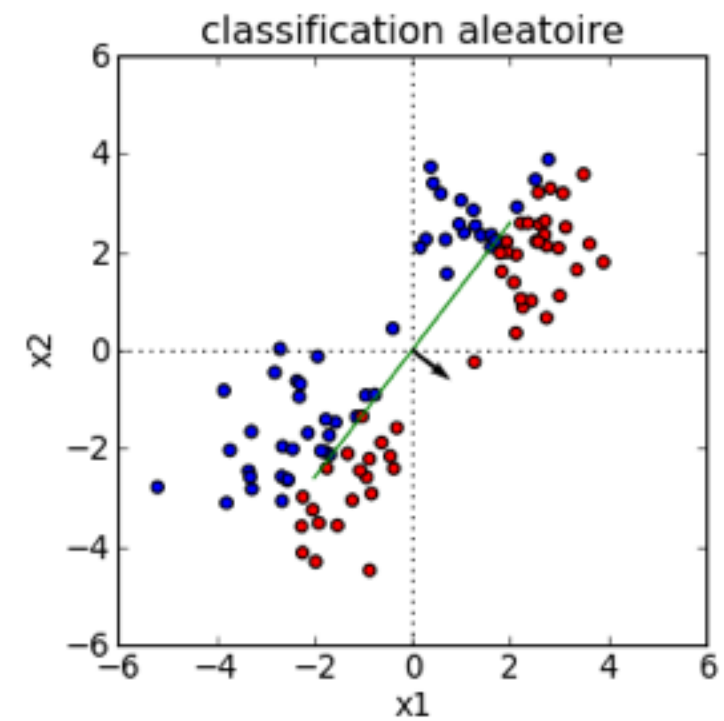
ex_classif.py



$$y = g(a) = g\left(\vec{w} \cdot \vec{x}\right)$$

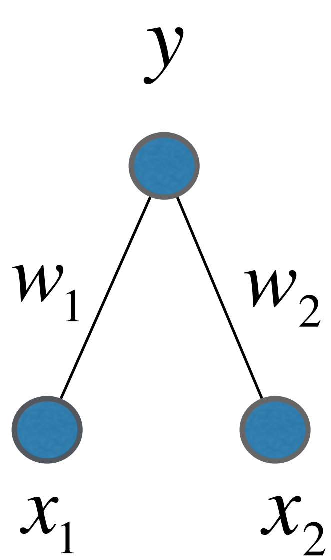
$$g(a) = \frac{1}{1+e^{-a}}$$





Le vecteur des poids $\vec{w}$ est perpendiculaire à la ligne de séparation

Réseau à une couche : apprentissage



$$y = g(a) = g\left(\vec{w}\cdot\vec{x}\right)$$

$$g(a) = \frac{1}{1+e^{-a}}$$

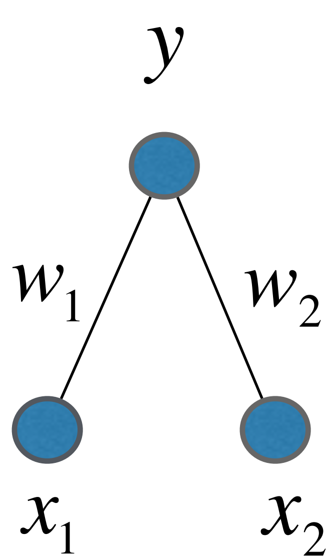
Règle d'apprentissage (delta rule)

$$\Delta w_i = -\eta\delta x_i^{(p)} \quad \Rightarrow \quad \Delta\vec{w} = -\eta\delta\vec{x}^{(p)}$$

$$\delta = g'(a)(y^{(p)} - t^{(p)})$$

$$g'(a) = g(a)(1 - g(a))$$

Réseau à une couche : apprentissage



$$y = g(a) = g\left(\vec{w}\cdot\vec{x}\right)$$

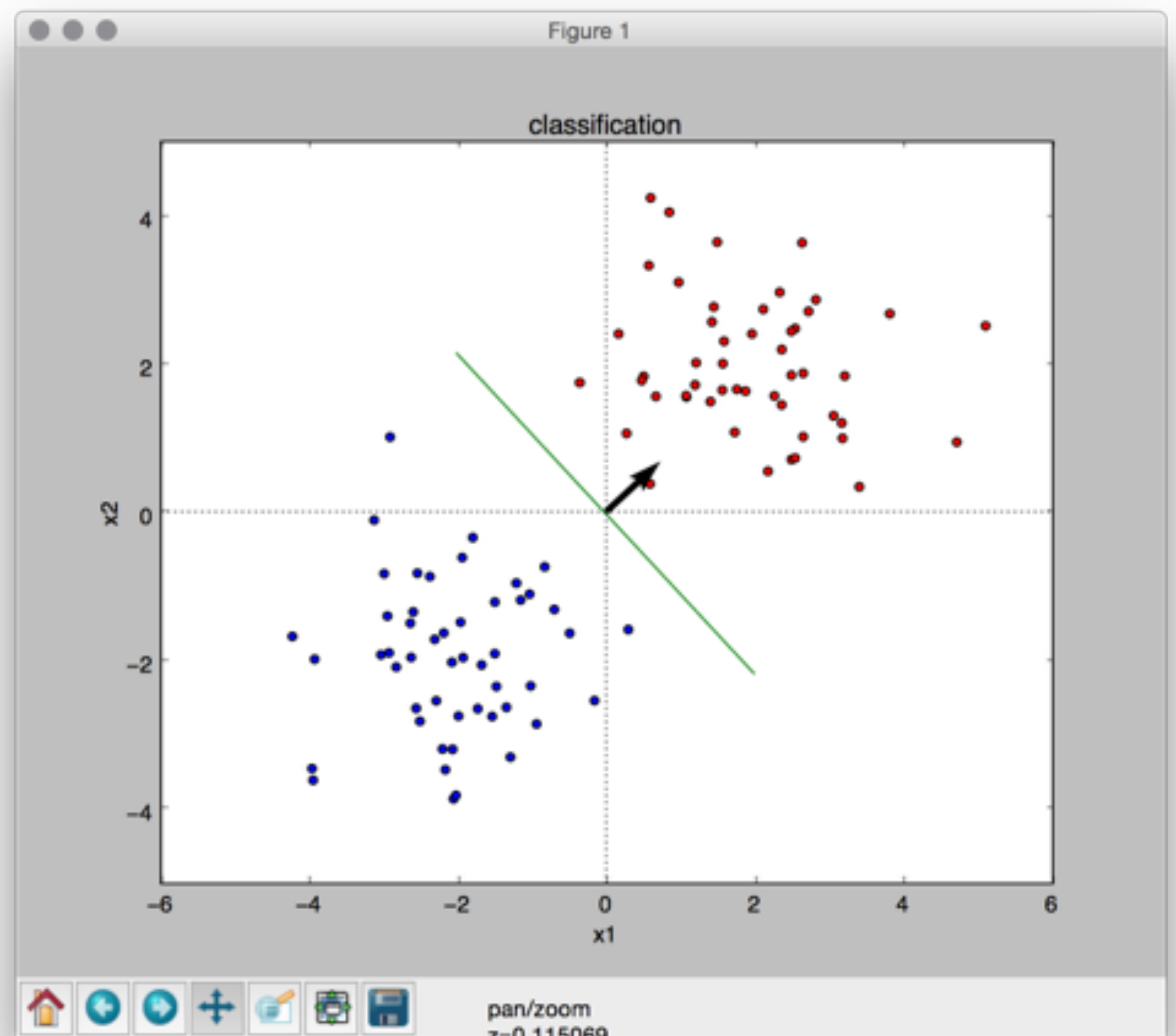
$$g(a) = \frac{1}{1+e^{-a}}$$

$$\Delta\vec{w} = -\eta\delta\vec{x}^{(p)}$$

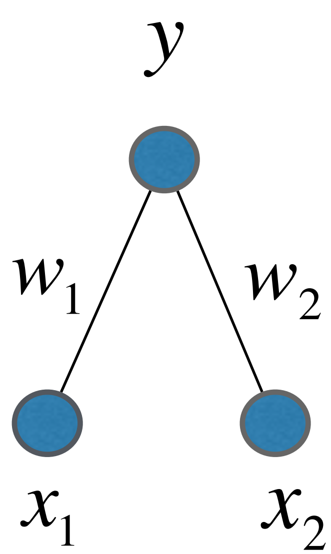
$$\delta = g'(a)(y^{(p)} - t^{(p)})$$

$$g'(a) = g(a)(1-g(a))$$

apprentis.py

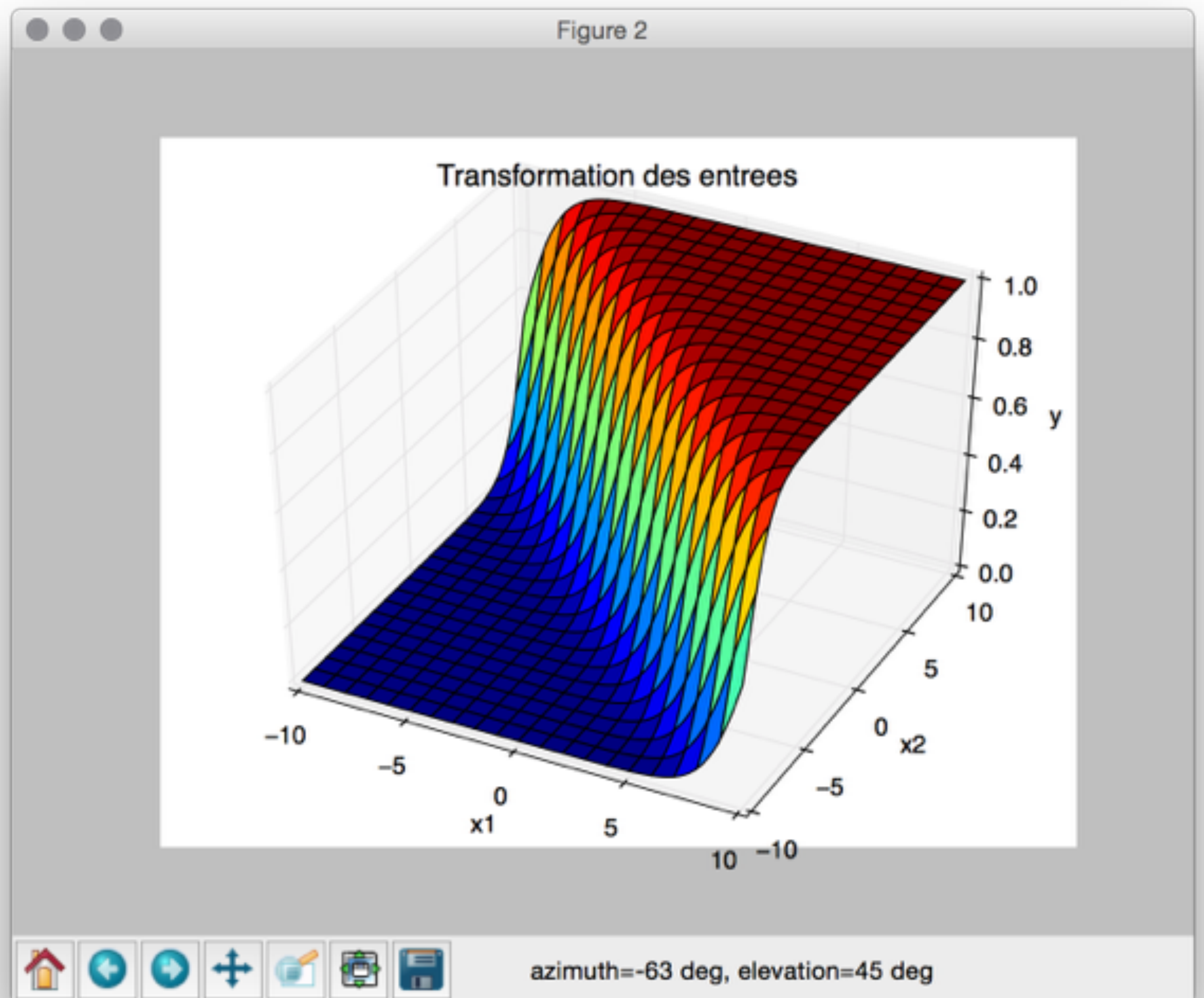


Réseau à une couche : apprentissage



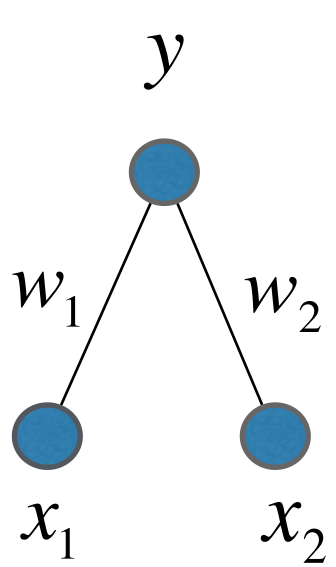
$$y = \frac{1}{1 + e^{-w_1 x_1 - w_2 x_2}}$$

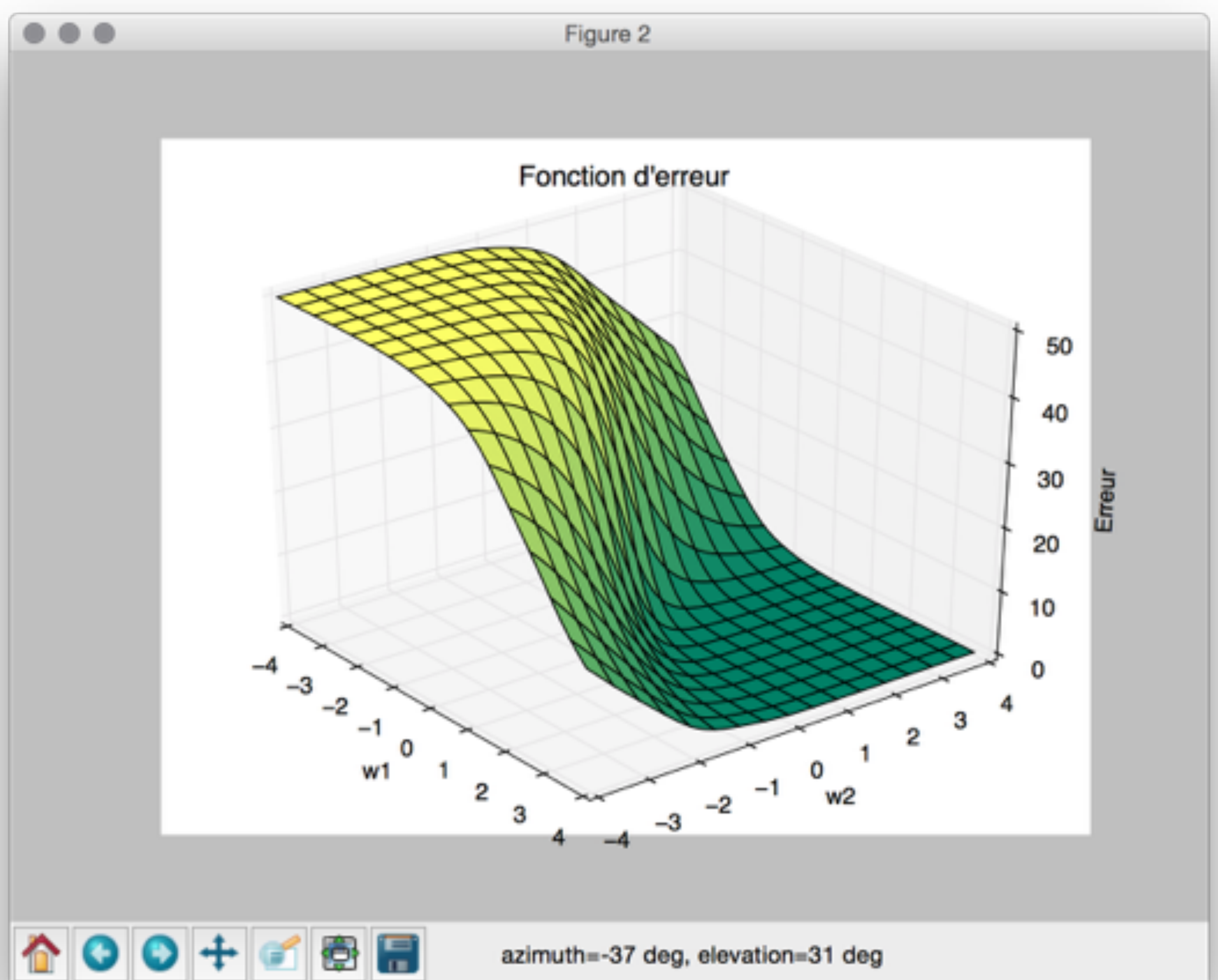
apprentis.py



Réseau à une couche : apprentissage

apprentis.py

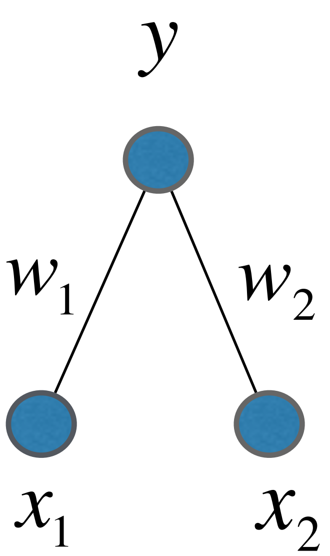




$$E(w_1, w_2) = \sum_{p=1}^{N} \frac{1}{2} (y^{(p)} - t^{(p)})^2$$

$$y^{(p)} = g(\vec{w} \cdot \vec{x})$$

Réseau à une couche : apprentissage

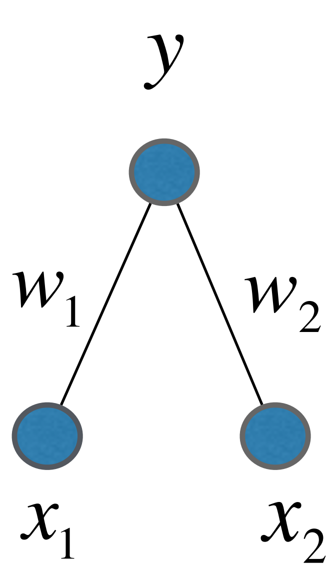


apprentis.py

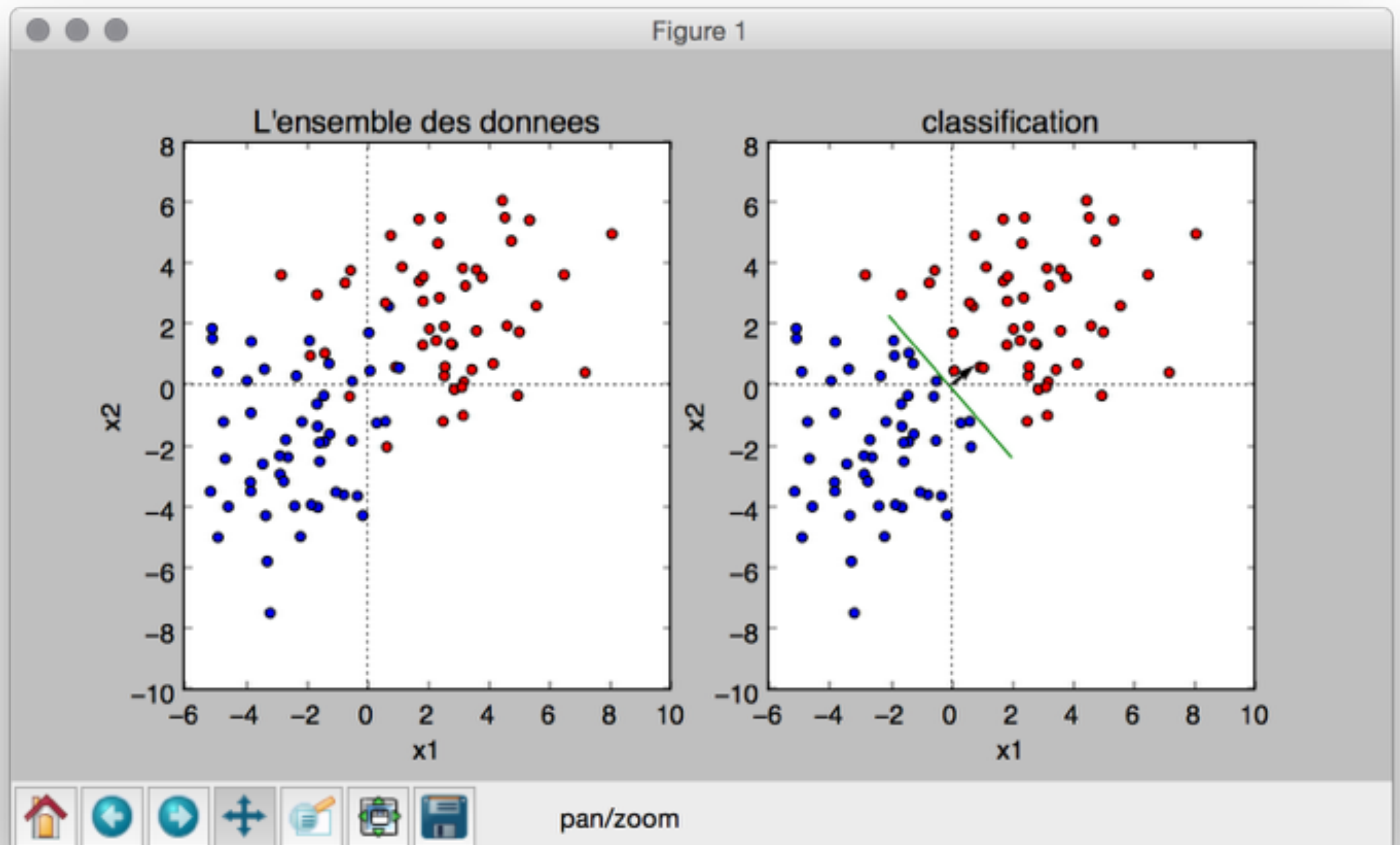
Question : Que se passe-t-il si les classes ne sont pas linéairement séparables ?

(par ex. si les écart-types des nuages sont trop larges)

Réseau à une couche : apprentissage



apprentis.py



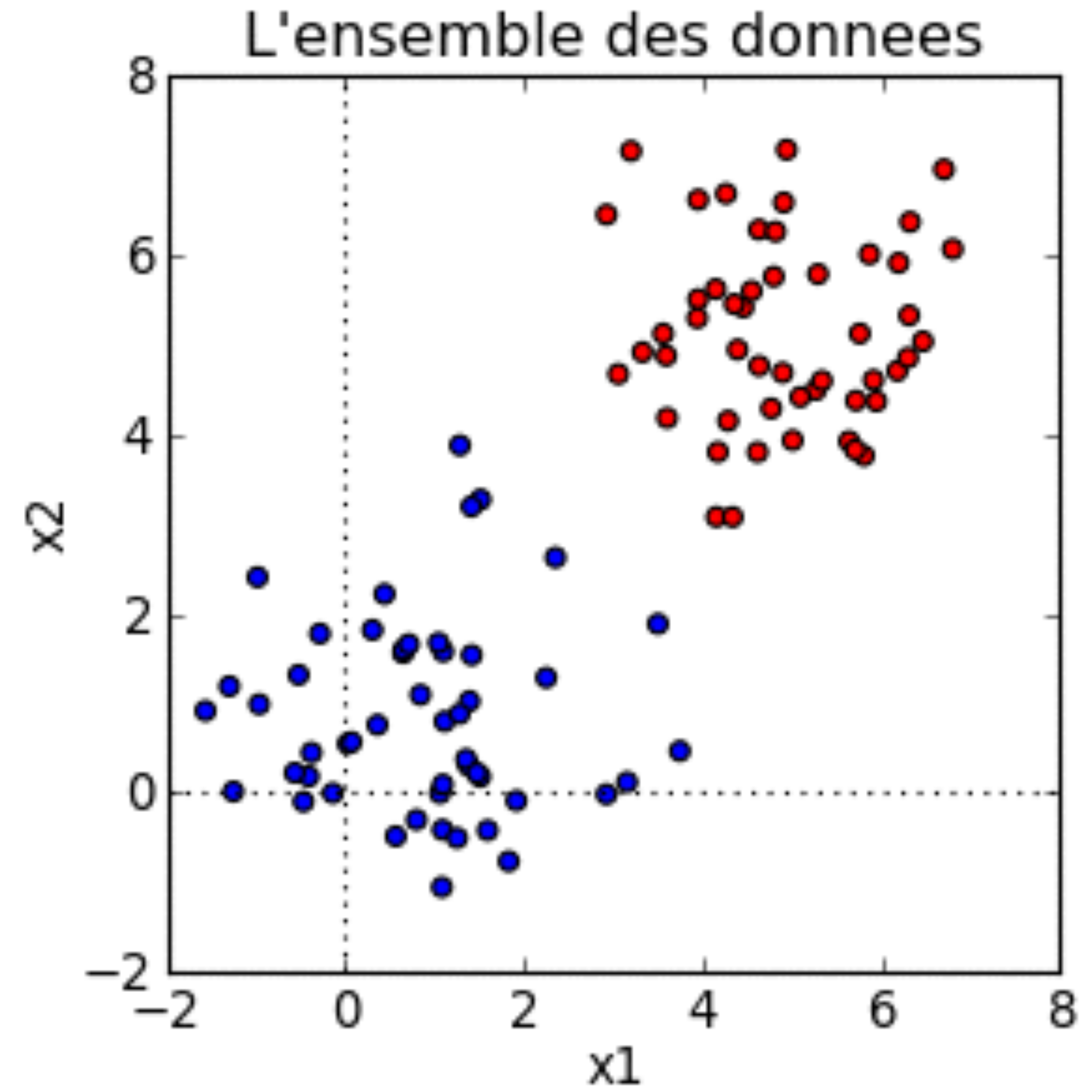
Exercice

- Modifier les données pour avoir

  m1 = (5, 5) pour classe 1
  m2 = (1, 1) pour classe 2

- Est-ce que l'algorithme marche pour ces données ? Pourquoi ?

- Modifier l'algorithme "apprentis.py" pour trouver l'hyperplan séparateur



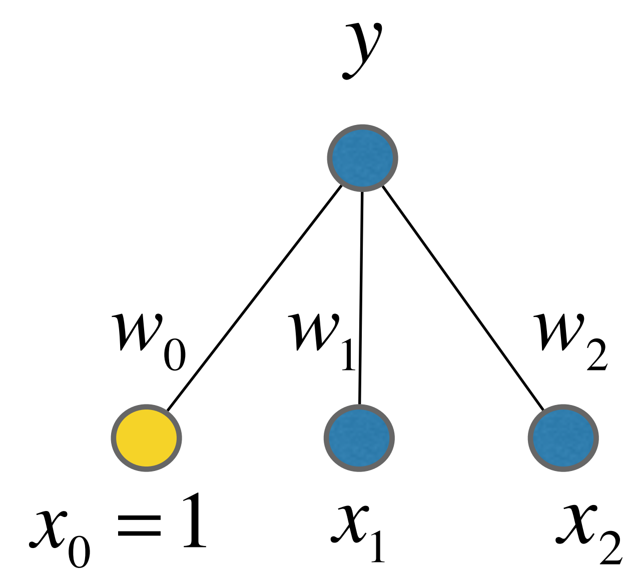
Exercice

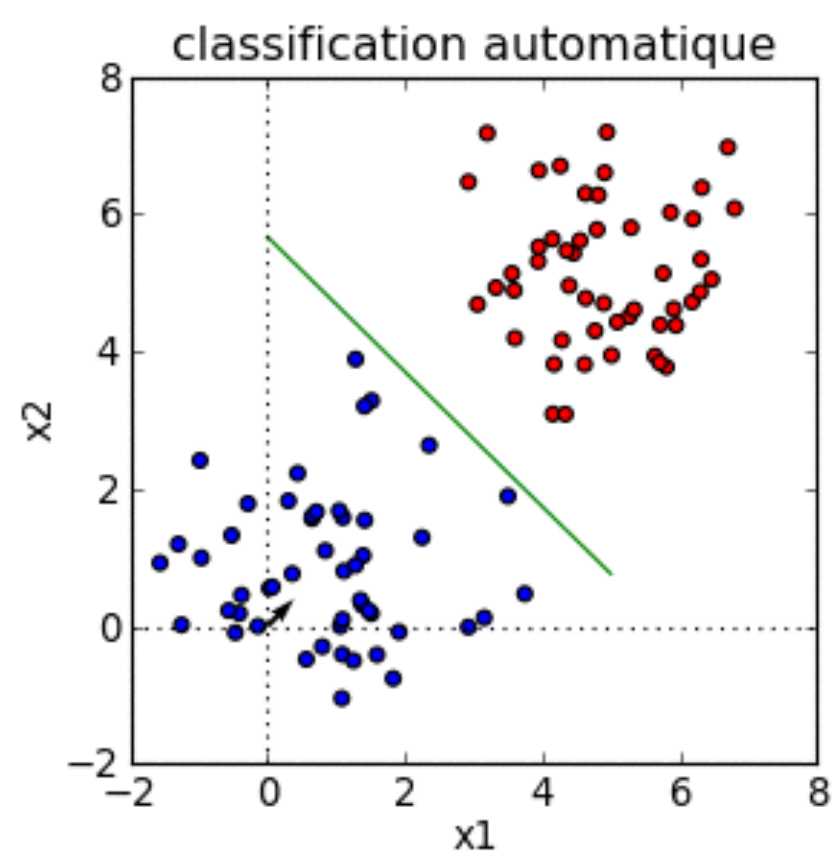
- Modifier les données pour avoir

  m1 = (5, 5) pour classe 1
  m2 = (1, 1) pour classe 2

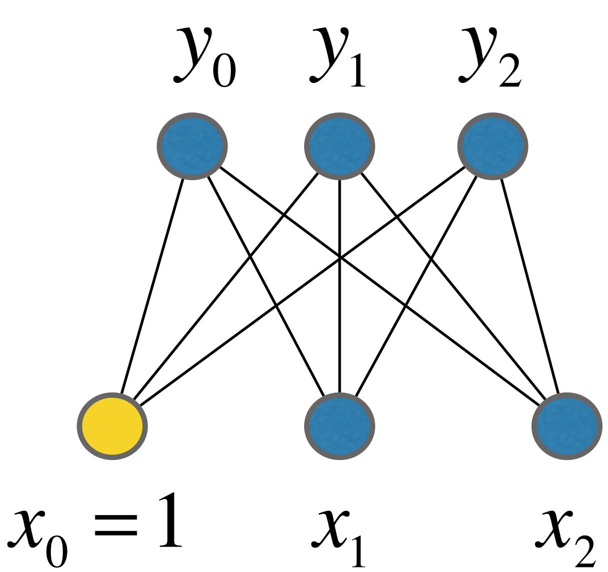
- Est-ce que l'algorithme marche pour ces données ? Pourquoi ?

- Modifier l'algorithme "apprentis.py" pour trouver l'hyperplan séparateur





Réseau à une couche - plusieurs classes



Produit scalaire

Produit dyadique (outer product)

$$y_k = g(a_k) = g\left(\sum_{i=1}^{D} w_{ki} x_i\right) \quad \Rightarrow \quad \vec{y} = g\left(W \cdot \vec{x}\right)$$

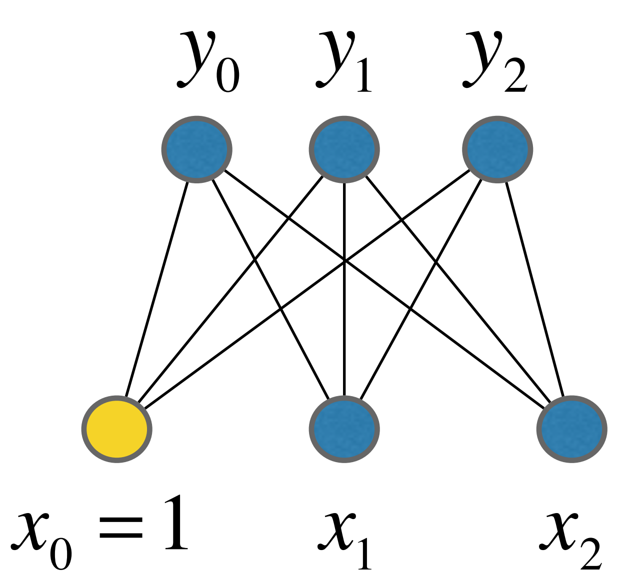
$$\Delta w_{ki} = -\eta \delta_k x_i \quad \Rightarrow \quad \Delta W = -\eta \begin{pmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \end{pmatrix} \times \begin{pmatrix} x_0 & x_1 & x_2 \end{pmatrix} = -\eta \vec{\delta} \otimes \vec{x}$$

$$\delta_k = g'(a_k)(y_k^{(p)} - t_k^{(p)}) \quad \Rightarrow \quad \vec{\delta} = \vec{g}(\vec{a}) \times (\vec{y}^{(p)} - \vec{t}^{(p)})$$

Produit usuel

Réseau à une couche - plusieurs classes

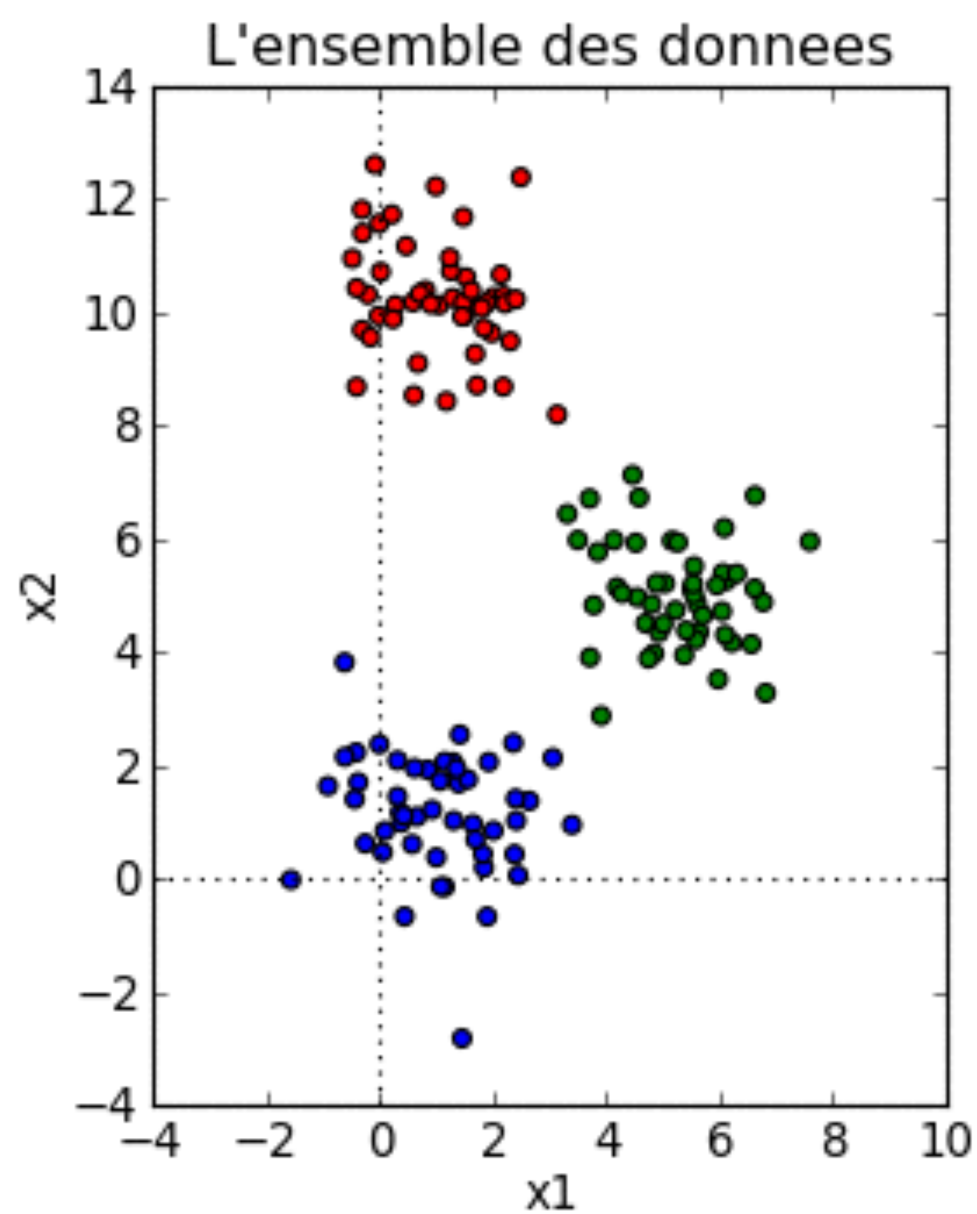
trois_classes.py



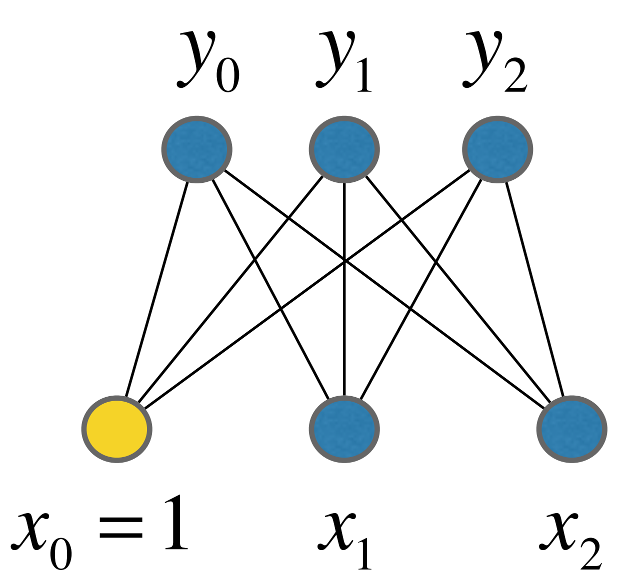
$$\vec{y} = g\left(W \cdot \vec{x}\right)$$

$$\Delta W = -\eta \vec{\delta} \otimes \vec{x}^{(p)}$$

$$\vec{\delta} = \vec{g}(\vec{a}) \times (\vec{y}^{(p)} - \vec{t}^{(p)})$$



Réseau à une couche - plusieurs classes



trois_classes.py

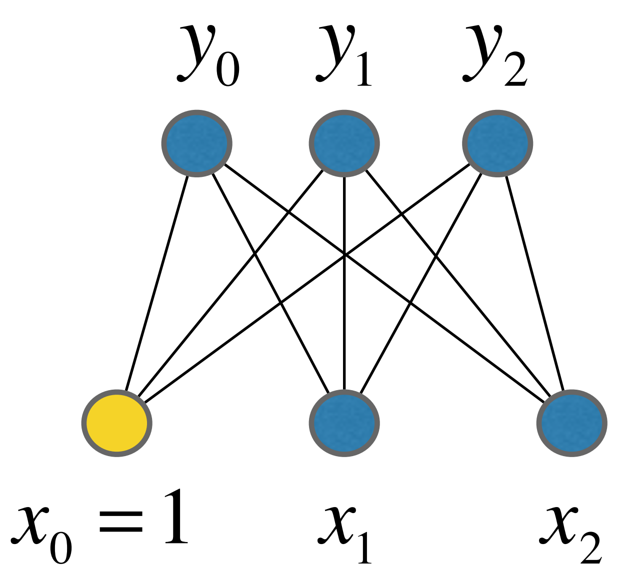
Question : Donner un exemple d'un ensemble de données qui ne peut pas être classifié correctement par ce réseau.

$$\vec{y} = g\left(W \cdot \vec{x}\right)$$

$$\Delta W = -\eta \vec{\delta} \otimes \vec{x}^{(p)}$$

$$\vec{\delta} = \vec{g}(\vec{a}) \times (\vec{y}^{(p)} - \vec{t}^{(p)})$$

Réseau à une couche - plusieurs classes



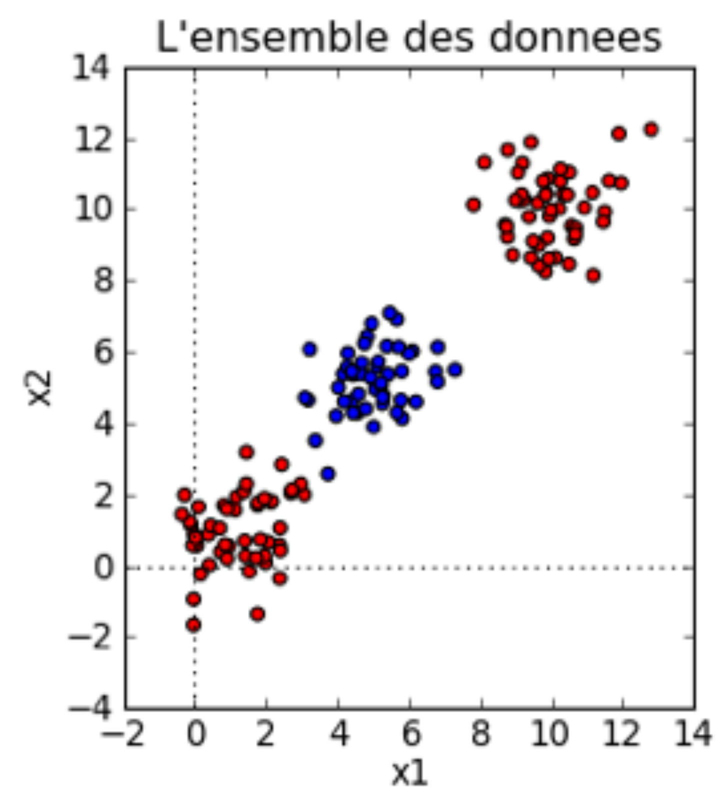
trois_classes.py

Question : Donner un exemple d'un ensemble de données qui ne peut pas être classifié correctement par ce réseau.

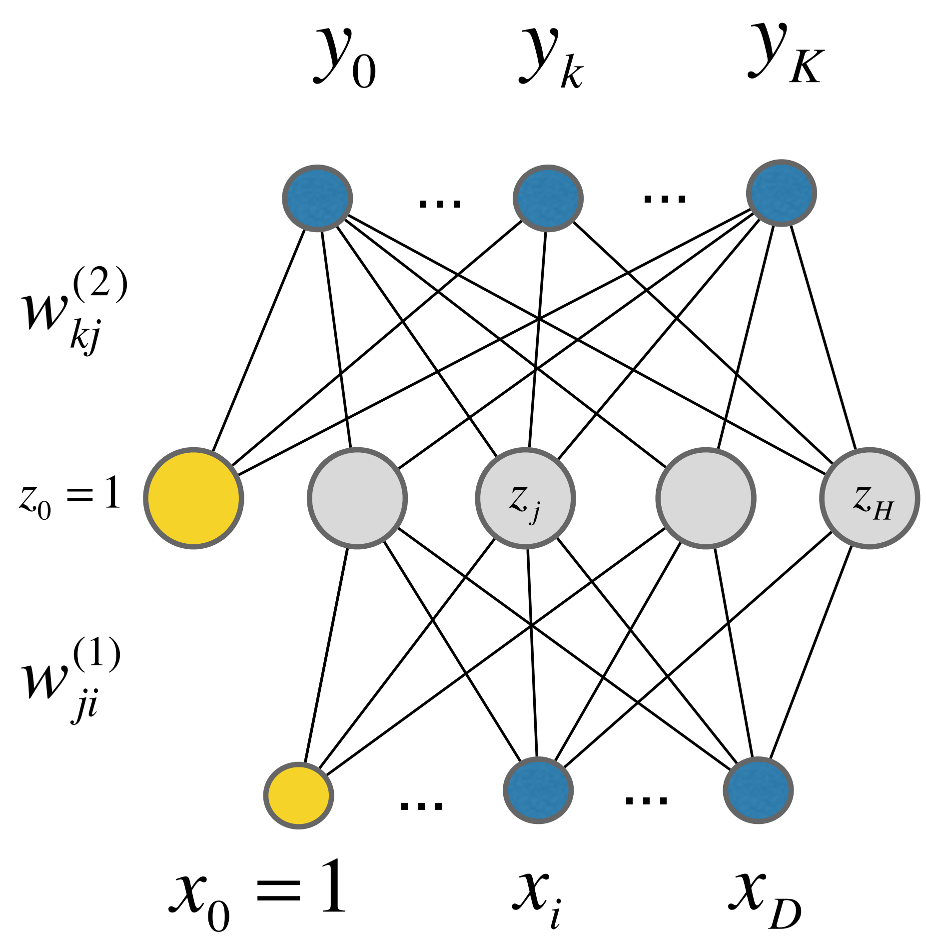
$$\vec{y} = g\left(W \cdot \vec{x}\right)$$

$$\Delta W = -\eta \vec{\delta} \otimes \vec{x}^{(p)}$$

$$\vec{\delta} = \vec{g}(\vec{a}) \times (\vec{y}^{(p)} - \vec{t}^{(p)})$$

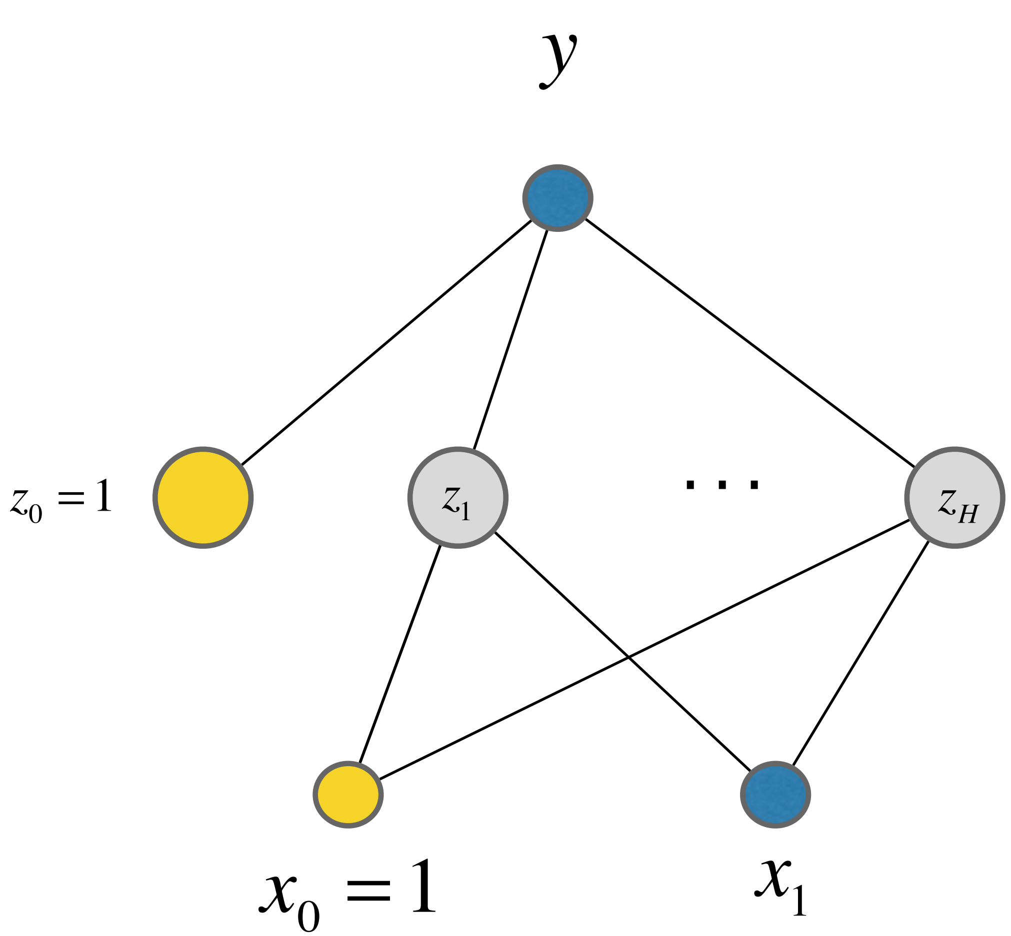


Réseau multicouche

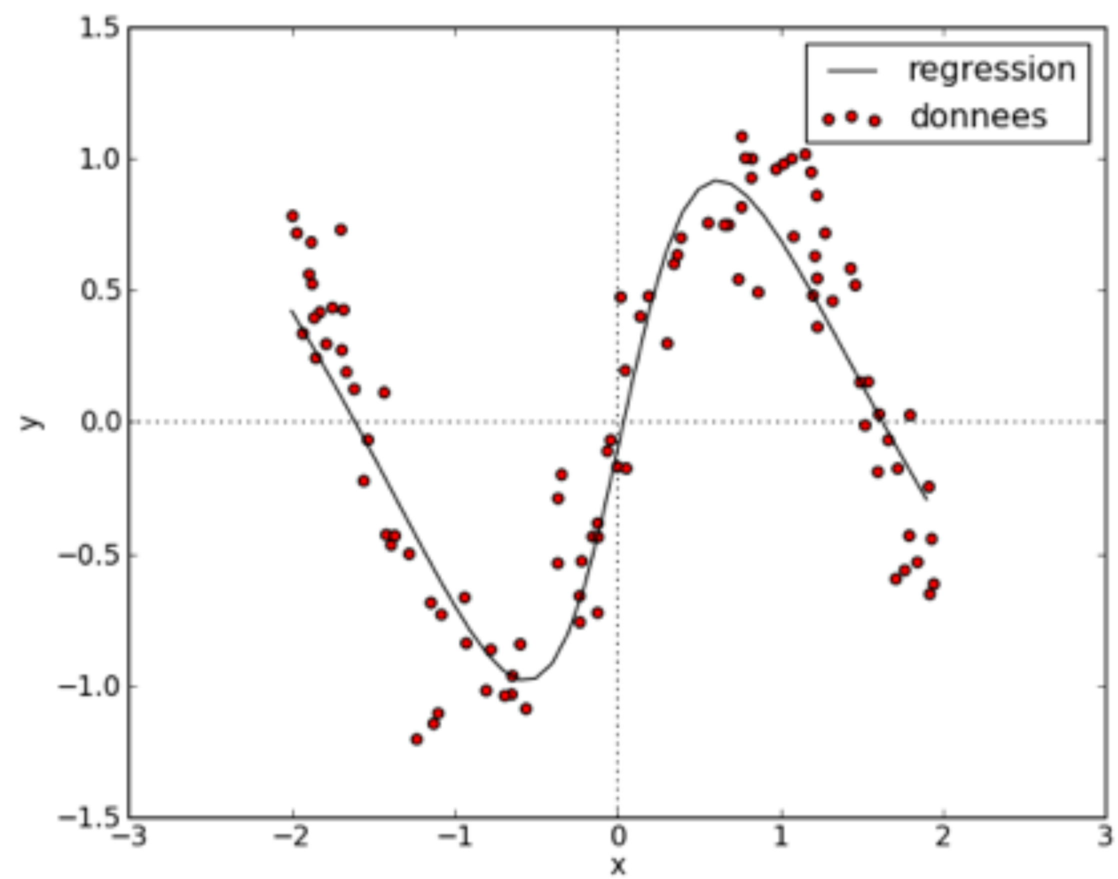


$$y_k = \sum_{j=0}^{H} w_{kj}^{(2)} z_j = w_{k0} + \sum_{j=1}^{H} w_{kj}^{(2)} g(a_j) = w_{k0} + \sum_{j=1}^{H} w_{kj}^{(2)} g\left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i \right)$$

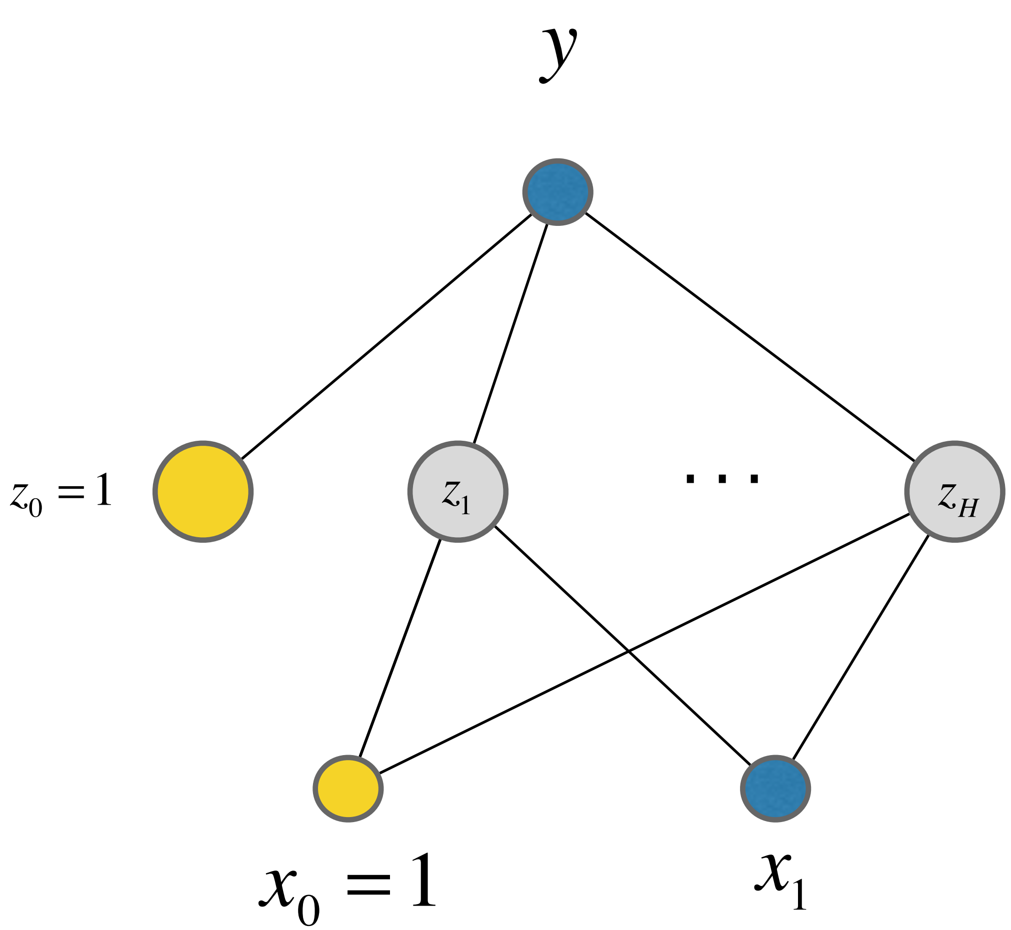
Réseau multicouche : régression



ml_regres.py



Réseau multicouche : régression

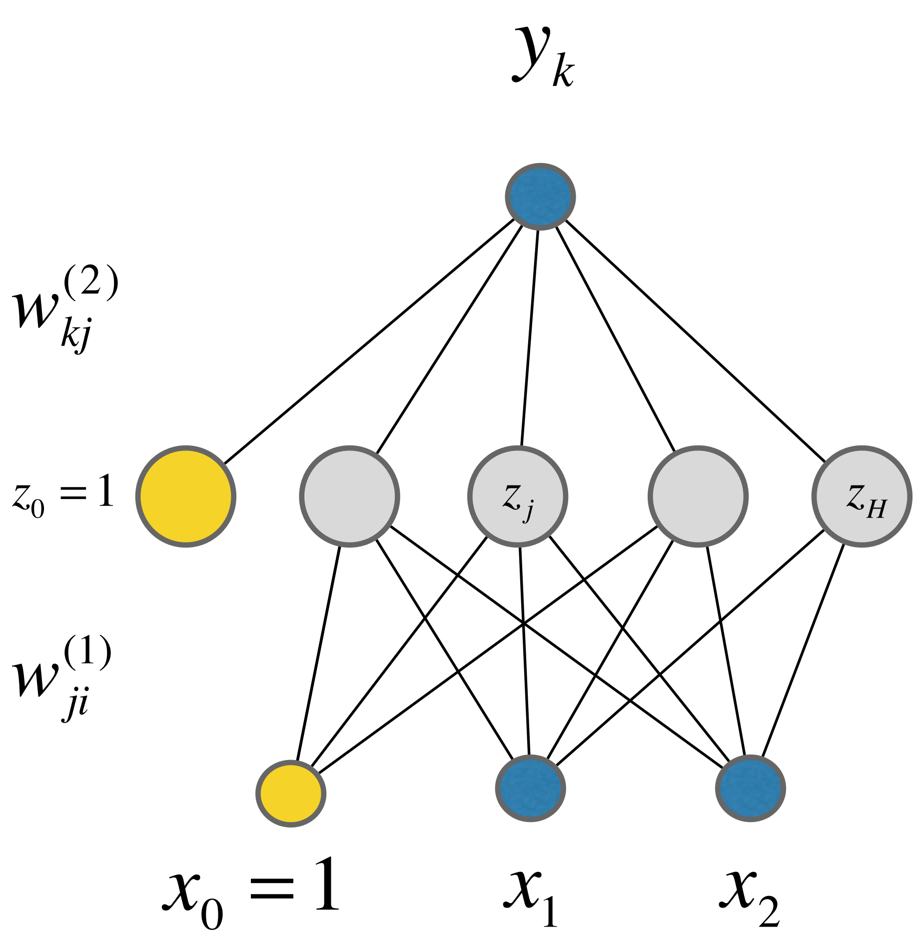


ml_regres.py

Questions :

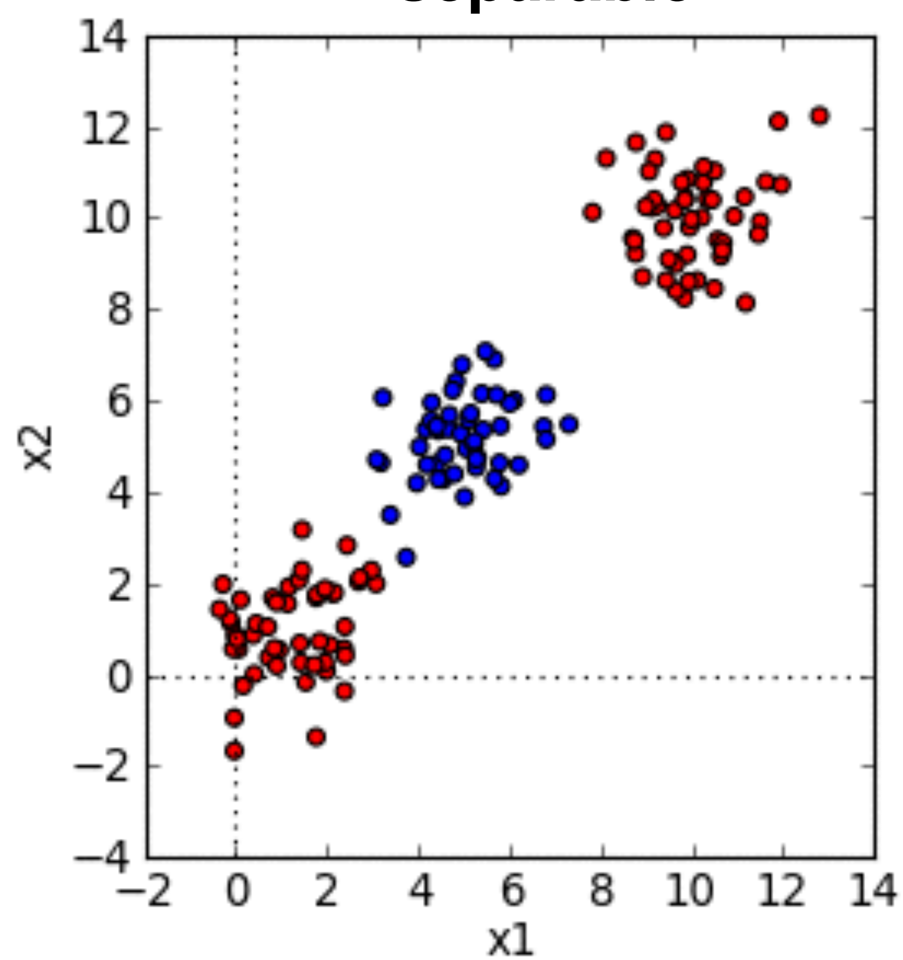
- Quel paramètre du réseau détermine la flexibilité de la courbe de régression ?
- Que se passe-t-il si ce paramètre soit égal à 1 ?

Réseau multicouche : classification

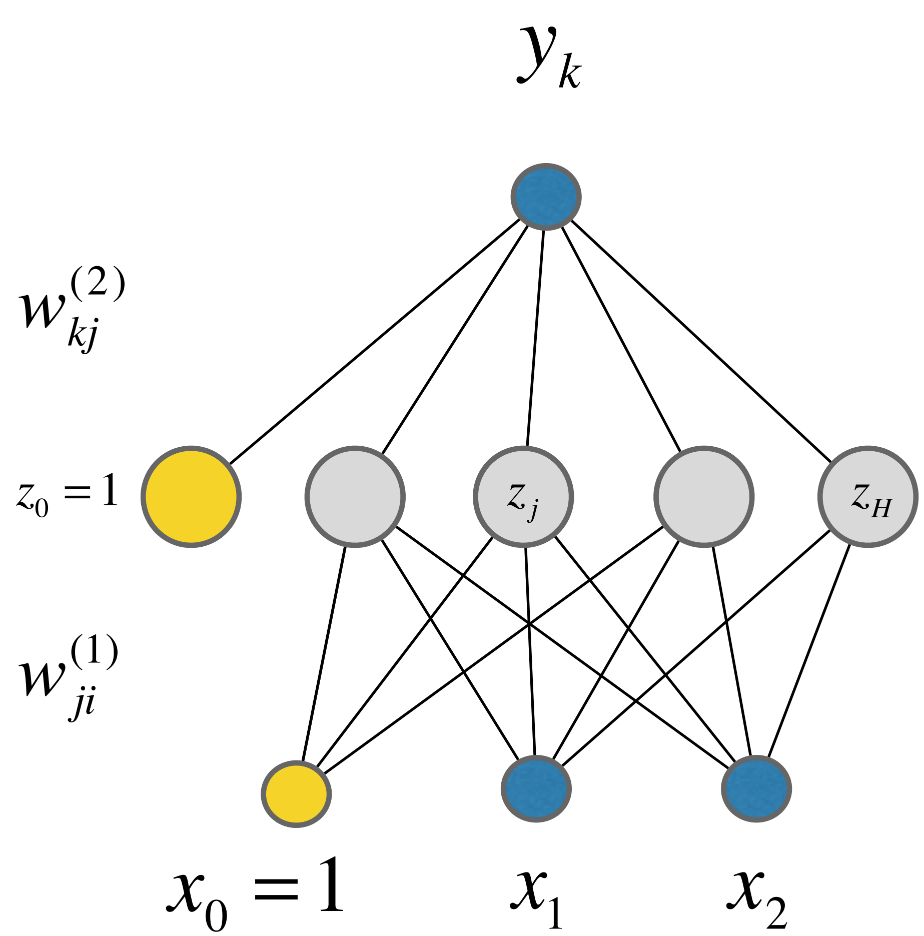


ml_classif.py

Ensemble de données non linéairement séparable



Réseau multicouche : classification



ml_classif.py