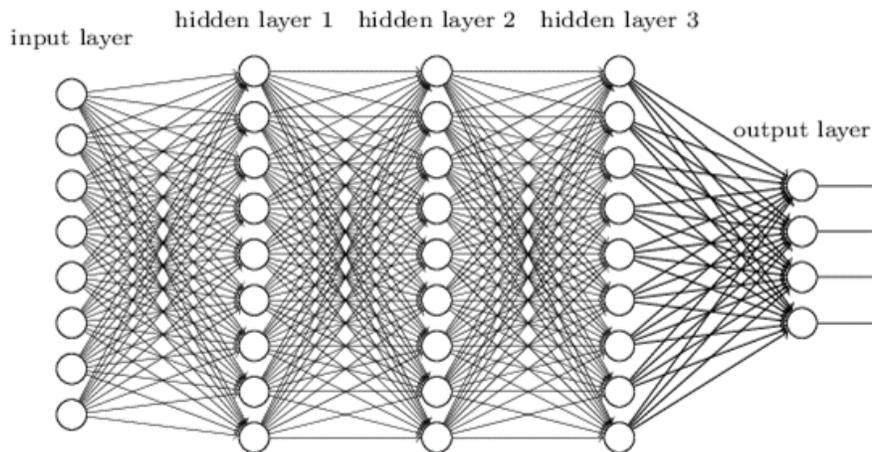


Chapitre 6.1 – L’algèbre linéaire du réseau de neurones

Deep neural network



<https://datawarrior.wordpress.com/2017/10/31/interpretability-of-neural-networks/>

DES CONCEPTS DE BASE EN ALGÈBRE LINÉAIRE	1
LA DROITE	1
LE GRADIENT	2
LA PSEUDO-DISTANCE À UNE DROITE	3
LA CLASSIFICATION	4
LES FONCTIONS DE LA CLASSIFICATION LINÉAIRE EN DEUX DIMENSIONS	5
LES TABLES DE VÉRITÉ	5
LA CLASSIFICATION À PLUSIEURS NEURONES	7
LE PROBLÈME DE LA CLASSIFICATION DE LA TABLE OU EXCLUSIF	9
LA FONCTION NON-LINÉAIRE	10
L’ESPACE DANS \mathbb{R}^N	12
L’HYPERPLAN	12
LA PSEUDO-DISTANCE À UN HYPERPLAN	12
BREAST CANCER WISCONSIN (DIAGNOSTIC) DATA SET	13

Des concepts de base en algèbre linéaire

La droite

Une droite dans \mathfrak{R}^2 peut être représentée sous les formes suivantes:

Notation en fonction $y = f(x)$	Notation en algèbre linéaire
$y = ax + b$	$ax + by + c = 0$
où a : Pente de la droite. b : Ordonnée à l’origine de la droite.	

Situation A : La droite. On désire **(a)** illustrer la droite $y = -x + 1,5$ et **(b)** obtenir sa représentation selon la notation en algèbre linéaire.

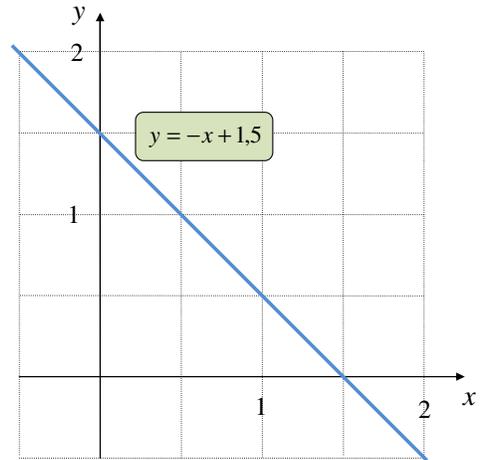
(a) Représentons dans la plan cartésien xy l'équation de la droite

$$y = -x + 1,5$$

Tel qu'illustré sur le schéma ci-contre.

(b) À partir de l'équation $y = -x + 1,5$, regroupons les termes du côté gauche de l'équation fin de représenter la droite selon la notation en algèbre linéaire $ax + bx + c = 0$ ce qui donnera l'équation

$$x + y - 1,5 = 0 .$$



Le gradient

Le gradient $\vec{\nabla}$ est un opérateur mathématique pouvant être appliquée sur une fonction scalaire f afin de la transformer sous une fonction vectorielle $\vec{g} = \vec{\nabla}f$. Le gradient peut être interprété comme étant la direction où la fonction f sera en croissance. Si la fonction f représente une géométrie (comme une droite), alors le gradient peut représenter la normale à la surface de la géométrie.

Le gradient $\vec{\nabla}f$ de la fonction f correspond au calcul suivant :

Pour une fonction $f(x, y)$ à deux dimensions	Pour une fonction $f(x_0, x_1, \dots, x_\mu, x_{N-1})$ à N dimensions
$\vec{\nabla}f(x, y) = \frac{\partial f}{\partial x} \vec{i} + \frac{\partial f}{\partial y} \vec{j}$	$\vec{\nabla}f(x, y) = \sum_{\mu=0}^{N-1} \frac{\partial f}{\partial x_\mu} \hat{x}_\mu$
où \vec{i} et \vec{j} sont les vecteurs unitaire des axes x et y .	où \hat{x}_μ est le vecteur unitaire de la dimension μ .

Situation B : La normale de la droite. On désire (a) évaluer la normale de la droite $y = -x + 1,5$ et (b) représenter la normale de la droite graphiquement.

(a) À partir de la droite $y = -x + 1,5$ que l'on peut représenter sous la forme $x + y - 1,5 = 0$, construisons la fonction

$$f(x, y) = x + y - 1,5 .$$

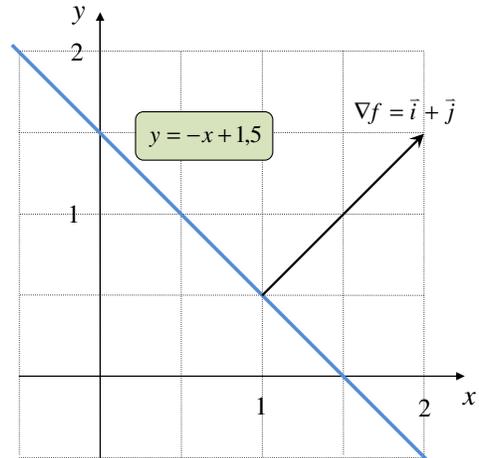
Évaluons le gradient de la fonction afin d'obtenir la normale de la droite :

$$\bar{\nabla}f(x, y) = \frac{\partial f}{\partial x} \bar{i} + \frac{\partial f}{\partial y} \bar{j}$$

$$\Rightarrow \bar{\nabla}f(x, y) = \frac{\partial}{\partial x} (x + y - 1,5) \bar{i} + \frac{\partial}{\partial y} (x + y - 1,5) \bar{j}$$

$$\Rightarrow \bar{\nabla}f(x, y) = (1) \bar{i} + (1) \bar{j}$$

$$\Rightarrow \boxed{\bar{\nabla}f(x, y) = \bar{i} + \bar{j}}$$



(b) La normale $\bar{\nabla}f$ est représentée sur le schéma ci-contre.

La pseudo-distance à une droite

Lorsque l'on évalue l'équation d'une droite

$$ax + by + c = 0 ,$$

avec un point (x, y) quelconque, il y a trois types de résultats possibles :

Scénario du point (x, y)	Interprétation
$ax + by + c = 0$	Le point est sur la droite
$ax + by + c > 0$	Le point est au-dessus de la droite
$ax + by + c < 0$	Le point est en dessous de la droite

Ainsi, on peut définir une fonction

$$z(x, y) = ax + by + c$$

afin d'évaluer une *pseudo-distance* entre le point (x, y) et la droite. La fonction $z(x, y)$ ne peut pas être considérée comme une distance euclidienne dans le plan xy , car il existe plusieurs fonctions $z(x, y)$ différentes correspondant à la même droite mathématique. Elles se distinguent tous pour une valeur de gradient $\bar{\nabla}z$ de modules différentes. Pour avoir une distance, il faudrait que le gradient $\bar{\nabla}z$ possède une longueur unitaire correspondant à un vecteur normale unitaire.

En intelligence artificielle, la fonction $z(x, y)$ porte le nom de fonction d'agrégation. Elle n'a pas besoin d'être « une distance », mais une « appréciation » de la distance entre un point (x, y) et la droite. Ainsi, la notion de *pseudo-distance* répondra adéquatement au besoin des calculs.

Situation C : La distance entre un point et une droite. On désire évaluer la *pseudo-distance* entre les points **(a)** $(0,0)$, **(b)** $(1,0)$ et **(c)** $(1,1)$ avec la droite $y = -x + 1,5$.

À partir de la droite $y = -x + 1,5$ représentée sous la fonction $z(x, y) = x + y - 1,5$, évaluons la distance entre trois points et cette droite :

(a) Point $(0,0)$:

$$z(x=0, y=0) = (0) + (0) - 1,5$$

$$\Rightarrow \boxed{z(x=0, y=0) = -1,5} \text{ (en-dessous)}$$

(b) Point $(1,0)$:

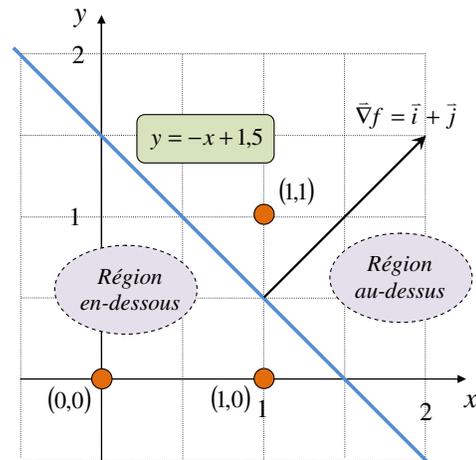
$$z(x=1, y=0) = (1) + (0) - 1,5$$

$$\Rightarrow \boxed{z(x=1, y=0) = -0,5} \text{ (en-dessous)}$$

(c) Point $(1,1)$:

$$z(x=1, y=1) = (1) + (1) - 1,5$$

$$\Rightarrow \boxed{z(x=1, y=1) = 0,5} \text{ (au-dessus)}$$



La classification

La classification consiste à regrouper des données différentes ayant une caractéristique commune. Par exemple, un ensemble d'images différentes contenant strictement des représentations de chats et de chiens peuvent être classifié en deux catégories : chat et chien. Bien que toutes les images de chats soient différentes, elle représente un même caractéristique, soit d'être un chat. Si un système est capable de bien classifier une image, la classification de cette image est alors un succès.

Le taux de classification correspondra au pourcentage de succès de classification sur une grande collection de données à classifier. Par exemple, si un système est capable de séparer l'ensemble des images dans toutes les catégories disponibles sans faire d'erreur, le taux de classification sera de 100%. Un humain est capable sans trop de difficulté à réaliser cette tâche pour des images de chats et de chien (voir l'image ci-contre).



https://www.reddit.com/r/aww/comments/3pbnei/cuteness_couple_of_cat_and_dog_with_same_look/

Pour un humain, distinguer le chat du chien sur l'image est relativement simple ce qui n'est pas le cas pour un réseau de neurones.

Le défi mathématique en classification est de :

- Représenter la donnée à classifier sous une forme mathématique (vecteur d'entrée).
- Utiliser une structure mathématique (réseau de neurones) pour calculer à quelle catégorie appartient une donnée (vecteur de sortie).
- Vérifier si la classification de l'image est un succès (l'entraîneur).
- Améliorer la structure mathématique si la classification est inadéquate (l'apprentissage).

Pour réaliser cette tâche colossale mathématique, l'usage de l'informatique est nécessaire car elle permet l'accès à un vaste espace mémoire pour représenter le vecteur d'entrée et le réseau de neurones et dispose d'une puissance de calcul phénoménal pour réaliser les apprentissages.

Les fonctions de la classification linéaire en deux dimensions

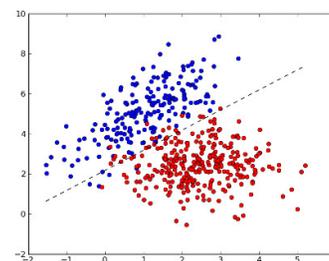
La classification linéaire en deux dimensions utilise l'équation d'agrégation

$$z(x, y) = ax + by + c$$

afin de déterminer si un point (x, y) est du bon côté de la droite. Cette fonction ne peut donc qu'admettre deux catégories pour le côté positif ou négatif de la droite. Pour définir numériquement les deux classes admissibles pour la fonction z , on peut utiliser une fonction $R(z)$ tel que

$$R(z) = \begin{cases} R = 0 & \text{si } z \leq 0 \\ R = 1 & \text{si } z > 0 \end{cases}$$

qui admettra la classe « 0 » ou « 1 ». Dans l'exemple ci-contre, on peut faire l'association « 0 » et **rouge** et « 1 » et **bleu**.



<http://interweave-consulting.blogspot.com/2016/06/logistic-discriminant-analysis-neural.html>

Exemple de classification linéaire à deux dimensions. Le taux de classification rouge/bleu ne peut être de 100 % même en choisissant la meilleure droite possible.

En intelligence artificielle, la fonction $R(z)$ porte le nom de fonction de classification. Il en existe plusieurs sortes et elle est choisie en fonction du « chiffre » que l'on veut générer. Par exemple, on peut prendre une fonction qui dit « 0 » si faux et « 1 » si vrai. Une donnée (x, y) sera alors bien classée si la fonction $R(z)$ génère la « bonne valeur ».

Les tables de vérité

Les tables de vérité sont des relations mathématiques en lien avec l'algèbre de Boole. Elle propose deux valeurs d'entrée possible et deux valeurs de sortie étant

$$\text{vrai} \equiv 1 \text{ (vert)} \quad \text{et} \quad \text{faux} \equiv 0 \text{ (rouge)} .$$

Voici quelques tables connues :

E1		E2		E1 ET E2 (E1 and E2)		E1 OU E2 (E1 or E2)		E1 OU exclusif E2 (E1 xor E2)	
Faux	0	Faux	0	Faux	0	Faux	0	Faux	0
Vrai	1	Faux	0	Faux	0	Vrai	1	Vrai	1
Faux	0	Vrai	1	Faux	0	Vrai	1	Vrai	1
Vrai	1	Vrai	1	Vrai	1	Vrai	1	Faux	0

Nous pouvons représenter ces tables dans un plan cartésien avec l'association

$$x \equiv E1 \text{ (vert)} \quad \text{et} \quad y \equiv E2 \text{ (rouge)}$$

où la couleur correspond à la valeur de la table à la coordonnée (x, y) :

Table ET	Table OU	Table OU exclusif

Situation D : La classification de la table de vérité ET. On désire vérifier que nous pouvons réaliser la classification de la table de vérité ET avec

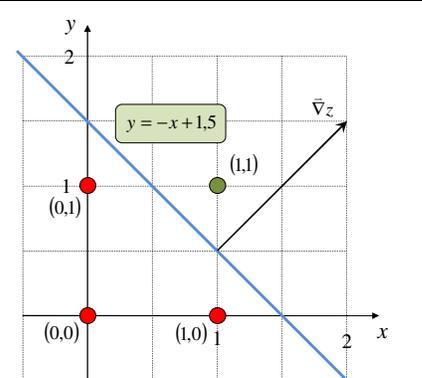
la droite $y = -x + 1,5$ et la fonction mathématique $R(z) = \begin{cases} R = 0 & \text{si } z \leq 0 \\ R = 1 & \text{si } z > 0 \end{cases}$.

Débutons par représenter la droite sous la forme d'une fonction d'agrégation $z(x, y)$:

$$y = -x + 1,5 \quad \Rightarrow \quad x + y - 1,5 = 0 \quad \Rightarrow \quad \boxed{z(x, y) = x + y - 1,5}$$

Évaluons nos points (x, y) dans l'équation $z(x, y)$ suivit de la fonction $R(z)$ afin de vérifier que la droite $y = -x + 1,5$ possède un taux de classification est de 100 % ce qui signifie que la fonction $a(z)$ donne exactement les résultats de la table de vérité ET :

Point (x, y)	$z(x, y)$	$R(z)$	Table ET
(0,0)	-1,5	0	0
(1,0)	-0,5	0	0
(0,1)	-0,5	0	0
(1,1)	0,5	1	1



Nous constatons que la classification est de 100 %, car la fonction $R(z)$ donne toutes les valeurs de la table ET pour l'ensemble des points (x, y) .

Situation E : La classification de la table de vérité OU. On désire vérifier que nous pouvons réaliser la classification de la table de vérité OU avec

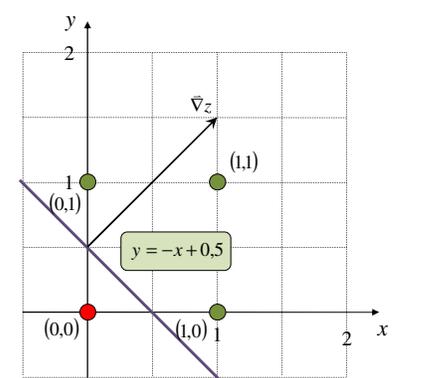
la droite $y = -x + 0,5$ et la fonction mathématique $R(z) = \begin{cases} R = 0 & \text{si } z \leq 0 \\ R = 1 & \text{si } z > 0 \end{cases}$.

Débutons par représenter la droite sous la forme d'une fonction d'agrégation $z(x, y)$:

$$y = -x + 0,5 \quad \Rightarrow \quad x + y - 0,5 = 0 \quad \Rightarrow \quad \boxed{z(x, y) = x + y - 0,5}$$

Évaluons nos points (x, y) dans l'équation $z(x, y)$ suivit de la fonction $R(z)$ afin de vérifier que la droite $y = -x + 0,5$ possède un taux de classification est de 100 % ce qui signifie que la fonction $a(z)$ donne exactement les résultats de la table de vérité OU :

Point (x, y)	$z(x, y)$	$R(z)$	Table OU
(0,0)	-0,5	0	0
(1,0)	0,5	1	1
(0,1)	0,5	1	1
(1,1)	1,5	1	1



Nous constatons que la classification est de 100 %, car la fonction $R(z)$ donne toutes les valeurs de la table OU pour l'ensemble des points (x, y) .

La classification à plusieurs neurones

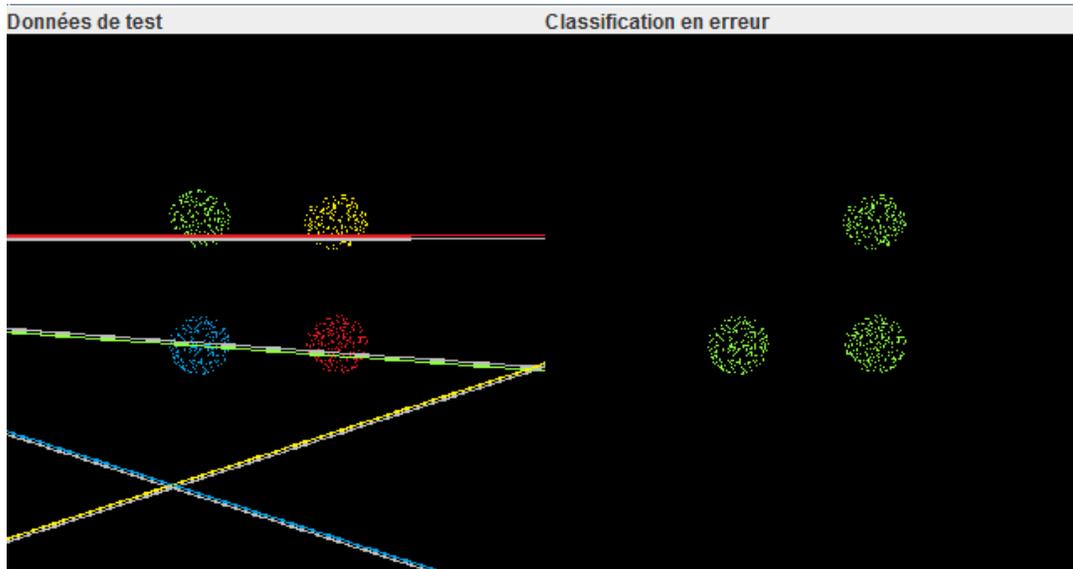
Pour réaliser une classification à plusieurs classes en deux dimensions, il nous faudra utiliser plusieurs fonctions d'agrégation $z_u(x, y)$. En intelligence artificielle, on utilise le terme neurone pour représenter la collection de fonctions qui aura pour but de donner un « signal fort » pour bien identifier une classe particulière. Un peu comme dans l'analogie du cerveau, un neurone émettra un « signal fort » s'il détecte ce pour quoi il est sensible.



<https://phys.org/news/2019-08-all-optical-neural-network-deep.html>

De façon traditionnelle, pour un réseau de neurones où il n'y a qu'une seule couche de neurone, nous attribuons pour chaque neurone une fonction d'agrégation $z_u(x, y)$ et une fonction de classification $R_u(z)$ qui aura pour but de générer $R_u(z) = 1$ si une donnée correspond à sa classe désirée et généré $R_u(z) = 0$ autrement.

Voici un exemple de classification pour un réseau à 4 neurones non entraîné pour 4 classes à classier :



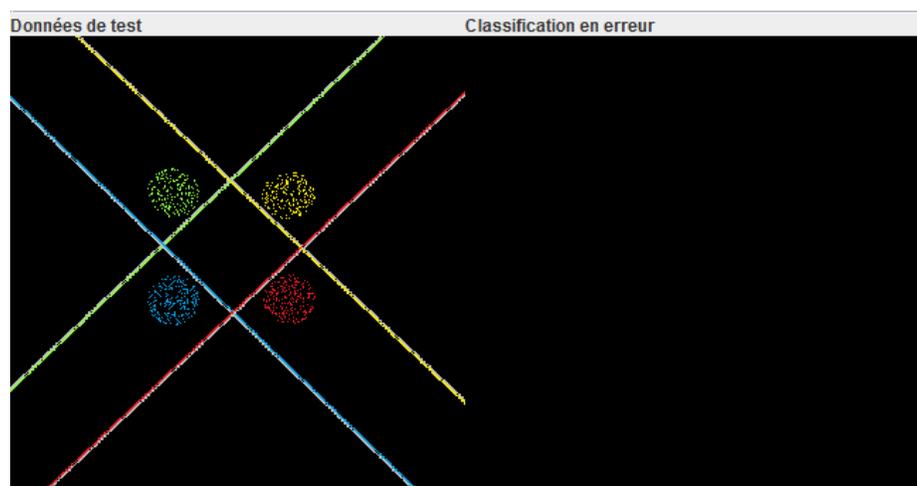
Réseau sans entraînement avec un taux de classification de 25%.

On peut visualiser graphiquement le comportement des 4 neurones à l'aide des 4 lignes (4 équations linéaires). Chaque ligne de couleur représente un neurone qui aura pour but de donner un signal fort (près de 1) s'il reconnaît sa couleur et un signal faible (près de 0) s'il ne la reconnaît pas. La ligne grise associée à chacune des droites représente le côté positif de la droite (côté pour avoir un signal supérieur à 0.5 selon le choix de la fonction de classification).

Présentement, ce réseau affirme que toutes les points sont verts, car les points « vert » et « jaune » sont du côté positif de la droite avec un signal fort (près de 1.0, car très loin) et les points « bleu » et « rouge » sont très près de celle-ci donnant un signal moyen (près de 0.5), mais quand même beaucoup plus élevé que les signaux des autres neurones, car ils sont du mauvais sens (donc signaux près de 0.0). Ainsi, ce réseau est très mal calibré !

Après un entraînement qui a eu pour but de calibrer les neurones, donc déplacer et réorienter les droites, nous obtenons le résultat ci-dessous :

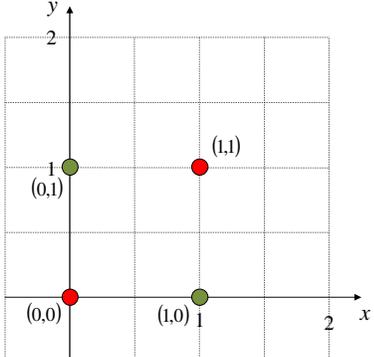
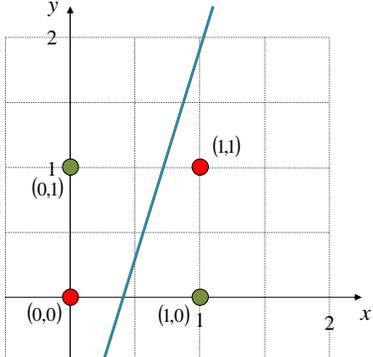
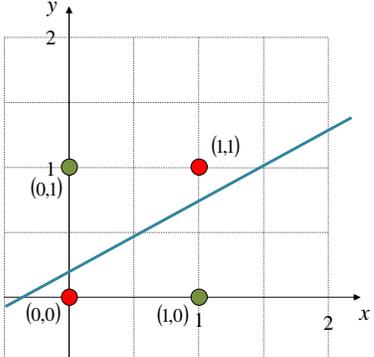
On remarque maintenant que tous les points sont maintenant bien classifiés. Tous les points de couleur sont maintenant du « côté positif » de leur droite correspondant. Ceci fait en sorte que les signaux des neurones sont près de 1.0 pour la bonne couleur d'un neurone et 0.0 pour la mauvaise couleur d'un neurone. C'est exactement le comportement recherché !



Le réseau est entraîné avec un taux de classification de 100%.

Le problème de la classification de la table OU exclusif

La table de vérité OU exclusif ne peut pas être classifiée uniquement que par des droites. On réalise en premier temps qu'il est impossible de réaliser la classification avec uniquement une droite :

Classification à réaliser (Table OU exclusif)	Exemple de mauvais choix #1	Exemple de mauvais choix #2
		

On réalise rapidement qu'on ne peut pas isoler d'un même côté les points **verts** et **rouges**. On peut alors tenter notre chance à l'aide de deux droites. Par exemple, nous pouvons utiliser les deux droites

$$y = -x + 1,5$$

(1^{re} droite, classification du ET)

$$y = -x + 0,5$$

(2^e droite, classification du OU)

ce qui nous donnera deux fonctions d'agrégation

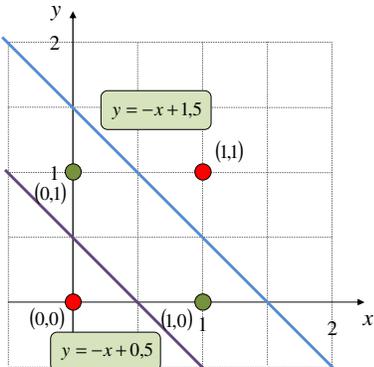
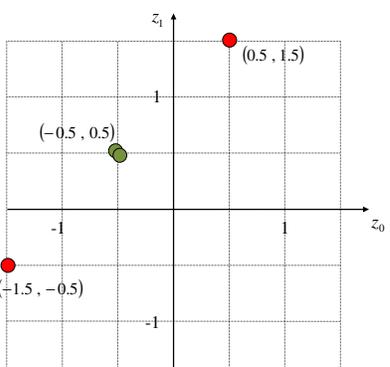
$$z_0(x, y) = x + y - 1,5$$

(1^{re} droite, classification du ET)

$$z_1(x, y) = x + y - 0,5$$

(2^e droite, classification du OU)

Évaluons nos points dans notre équation $z_0(x, y)$ et $z_1(x, y)$ afin de vérifier si une classification dans l'espace des coordonnées (z_0, z_1) est possible :

Point	Droite ET	Droite OU	Point dans l'espace (x, y)	Point dans l'espace (z_0, z_1)
(x, y)	$z_0(x, y)$	$z_1(x, y)$		
(0,0)	-1,5	-0,5		
(1,0)	-0,5	0,5		
(0,1)	-0,5	0,5		
(1,1)	0,5	1,5		

Nous remarquons que les points (x, y) exprimés dans l'espace (z_0, z_1) sont tous alignés sur une droite

$$z_1 = z_0 + 1$$

ce qui nous empêche de les classifier en « **vrai** » et « **faux** » selon la table de vérité OU exclusif. Nous pourrions utiliser d'autres droites, mais le problème de la disposition des points (z_0, z_1) serait le même qu'à l'origine. La raison est simple :

La table de vérité OU exclusif correspond à une fonction non-linéaire.

Pour régler la situation, nous devons introduire dans le calcul une fonction d'activation $a(z)$ non-linéaire qui aura pour but de pondérer le résultat de l'agrégation z de façon non-linéaire.

La fonction non-linéaire

Afin de réaliser la classification d'un problème non-linéaire, il est possible de pondérer la *pseudo-distance* $z = ax + by + c$ entre la droite et le point (x, y) à l'aide d'une fonction non-linéaire.

Par exemple, la fonction tangente hyperbolique

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

admet des valeurs dans l'intervalle

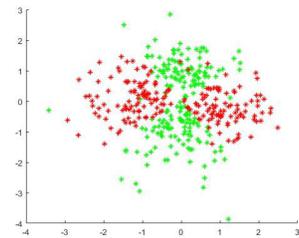
$$]-1, 1[$$

avec

$$\tanh(0) = 0, \quad \lim_{x \rightarrow \infty} \tanh(x) = 1 \quad \text{et} \quad \lim_{x \rightarrow -\infty} \tanh(x) = -1$$

Avec cette fonction, les signes de la fonction d'agrégation $z = (x, y)$ restent inchangés, mais est « recadré » dans un intervalle restreint.

En intelligence artificielle, la fonction $a(z)$ porte le nom de fonction d'activation et elle permet de recadrer la *pseudo-distance* sur un intervalle de valeur bien précis ce qui donnera un comportement non-linéaire à la fonction d'agrégation.



<https://it.mathworks.com/matlabcentral/fileexchange/63024-svm-for-nonlinear-classification>

Illustration d'une distribution de points dont la classification est non-linéaire.

Situation F : La classification de la table de vérité OU exclusif. On désire vérifier que nous pouvons réaliser la classification de la table de vérité OU exclusif avec

la droite $y = -x + 1,5$ et la droite $y = -x + 0,5$ en utilisant la fonction d'activation $a(z) = \tanh(z)$

$$\text{et la fonction mathématique } R(z) = \begin{cases} a = 0 & \text{si } z < 0 \\ a = 0,5 & \text{si } z = 0 \\ a = 1 & \text{si } z > 0 \end{cases} .$$

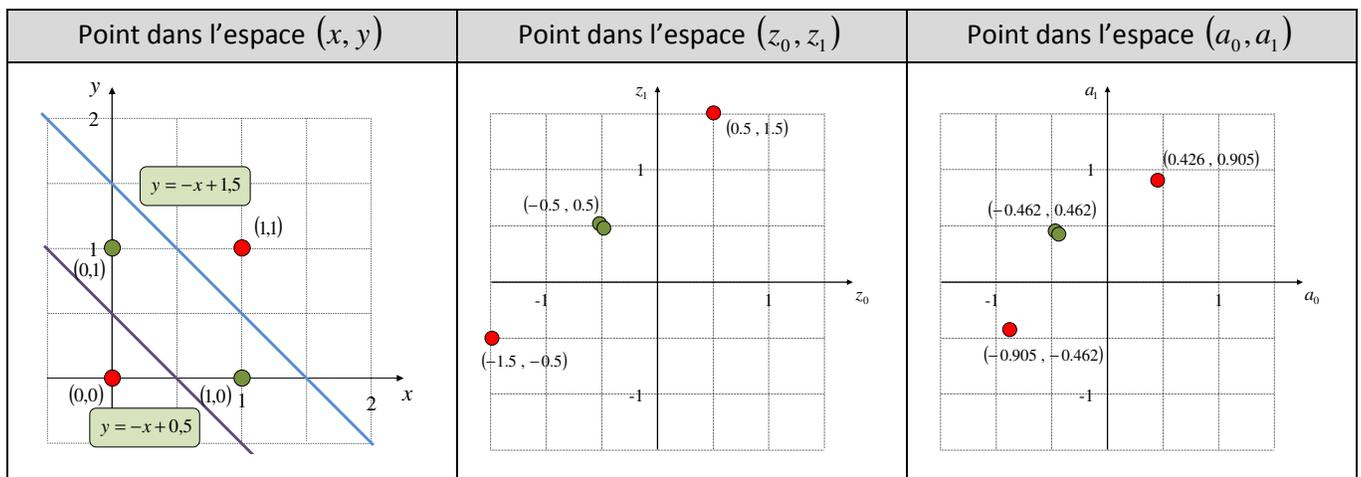
Débutons par représenter nos deux droites sous la forme d'une fonction d'agrégation $z(x, y)$:

$$y = -x + 1,5 \quad \Rightarrow \quad z_0(x, y) = x + y - 1,5$$

$$y = -x + 0,5 \quad \Rightarrow \quad z_1(x, y) = x + y - 0,5$$

Évaluons nos points (x, y) dans l'équation $z_0(x, y)$ et $z_1(x, y)$ suivit de la fonction $a(z) = \tanh(z)$ afin de former des nouveaux points (a_0, a_1) associés à l'ensemble de nos points (x, y) :

Point (x, y)	$z_0(x, y)$	$a_0(z_0)$	$z_1(x, y)$	$a_1(z_1)$	Point (a_0, a_1)
(0,0)	-1,5	-0,90515	-0,5	-0,46212	(-0,905 , -0,462)
(1,0)	-0,5	-0,46212	0,5	0,46212	(-0,462 , 0,462)
(0,1)	-0,5	-0,46212	0,5	0,46212	(-0,462 , 0,462)
(1,1)	0,5	0,46212	1,5	0,90515	(0,462 , 0,905)



Nous constatons que nous pouvons réaliser la classification en introduisant une 3^e droite dans l'espace (a_0, a_1) tel que

$$a_1 = a_0 + 0,5$$

ce qui donne la fonction d'agrégation

$$z_3^*(a_0, a_1) = a_0 - a_1 + 0,5 \text{ .}$$

Afin d'aligner le sens du gradient de la fonction d'agrégation pour réaliser la classification avec la fonction

$$R(z) = \begin{cases} a = 0 & \text{si } z \leq 0 \\ a = 1 & \text{si } z > 0 \end{cases} \text{ ,}$$

nous allons utiliser la fonction d'agrégation $z_3 = -z_3^*$ ce qui donne la fonction suivante :

$$z_3(a_0, a_1) = -a_0 + a_1 - 0,5$$

Évaluons nos points (a_0, a_1) dans l'équation $z_3(a_0, a_1)$ suivit de la fonction $R(z_3)$ afin de vérifier que notre séquence de fonction possède un taux de classification est de 100 % :

Point (a_0, a_1)	$z_3(a_0, a_1)$	$R(z_3)$	Table OU exclusif
$(-0,905, -0,462)$	$-0,057$	0	0
$(-0,462, 0,462)$	$0,424$	1	1
$(-0,462, 0,462)$	$0,424$	1	1
$(0,462, 0,905)$	$-0,057$	0	0

Nous constatons que la classification est de 100 %, car la fonction $R(z)$ donne toutes les valeurs de la table OU pour l'ensemble des points (x, y) .

L'espace dans \mathbb{R}^N

L'hyperplan

Un hyperplan dans \mathfrak{R}^N peut être représenté sous les formes suivantes:

Notation en algèbre linéaire	Notation avec produit scalaire
$\sum_{i=1}^N A_i x_i + B = 0$	$\bar{w} \cdot \bar{x} + b = 0$ <p>où $\bar{w} = \{w_0, w_1, \dots, w_u, \dots, w_{N-1}\}$ $\bar{x} = \{x_0, x_1, \dots, x_u, \dots, x_{N-1}\}$</p>

La pseudo-distance à un hyperplan

La *pseudo-distance* entre un hyperplan dans \mathfrak{R}^N et un point dans \mathfrak{R}^N représenté par le vecteur \bar{x} peut être évaluée grâce à l'équation suivante :

$$z(\bar{x}) = \bar{w} \cdot \bar{x} + b$$

où $z(\bar{x})$: *Pseudo-distance* entre le point et l'hyperplan.

\bar{x} : La position du point.

\bar{w}, b : Les caractéristiques de l'hyperplan.

Nous avons les interprétations suivantes sur le signe de la fonction $z(\bar{x})$:

Scénario du point \bar{x}	Interprétation
$z(\bar{x}) = 0$	Le point est sur l'hyperplan
$z(\bar{x}) > 0$	Le point est au-dessus de l'hyperplan
$z(\bar{x}) < 0$	Le point est en dessous de l'hyperplan

Breast Cancer Wisconsin (Diagnostic) Data Set

Au début des années 90, un groupe de chercheurs ont réussi à établir une corrélation entre des mesures effectuées sur des tumeurs en lien avec le dépistage du cancer du sein chez des patientes et la nature de la tumeur : **bénigne** ou **cancéreuse**.

Afin d'établir un lien entre une tumeur et l'algèbre linéaire, les données correspondant à des caractéristiques de la tumeur (l'épaisseur de la masse, l'uniformité de la taille des cellules, ...) ont été représentées sous la forme d'un vecteur \vec{x} à 9 dimensions contenant des valeurs entre 0 et 1.

Voici un tableau représentant quelques dossiers de patientes ainsi que le diagnostic de la tumeur :

$$y = 0 \text{ si la tumeur est « bénigne »} \qquad y = 1 \text{ si la tumeur est « cancéreuse ».}$$

ID ¹	Vecteur $\vec{x} = \{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$									Diagnostic y
	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
...										
1043999	0,1	0,1	0,1	0,1	0,2	0,3	0,3	0,1	0,1	0
1044572	0,8	0,7	0,5	1	0,7	0,9	0,5	0,5	0,4	1
1047630	0,7	0,4	0,6	0,4	0,6	0,1	0,4	0,3	0,1	1
1048672	0,4	0,1	0,1	0,1	0,2	0,1	0,2	0,1	0,1	0
1049815	0,4	0,1	0,1	0,1	0,2	0,1	0,3	0,1	0,1	0
1050670	1	0,7	0,7	0,6	0,4	1	0,4	0,1	0,2	1
1050718	0,6	0,1	0,1	0,1	0,2	0,1	0,3	0,1	0,1	0
1054590	0,7	0,3	0,2	1	0,5	1	0,5	0,4	0,4	1
...										

Référence : [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

À l'aide d'un algorithme itératif², le groupe de chercheurs a réussi à déterminer les caractéristiques d'un hyperplan dans \mathcal{R}^9 permettant de classer les diagnostics avec un taux de classification de plus de 97%. Ce succès permet d'affirmer que les 9 dimensions sont suffisantes pour bien diagnostiquer³ (peu de faux négatif ou faux positif) la tumeur sans avoir recours à une biopsie⁴.

¹ Le numéro ID correspond au numéro d'identification de la patiente.

² Cet algorithme porte le nom de la *descente du gradient*.

³ La qualité d'un diagnostic dépend du nombre de faux positifs (affirmation d'une tumeur cancéreuse, mais qui ne l'est pas) ou de faux négatifs (affirmation d'une tumeur bénigne, mais qui est en réalité cancéreuse).

⁴ Prélèvement d'un fragment de tissu sur un être vivant en vue d'un examen microscopique (référence : GOOGLE).

Situation G : La classification WDBC. On désire vérifier que nous pouvons réaliser la classification des données de la collection *Breast Cancer Wisconsin Data Set (WDBC)* avec la fonction d'agrégation

$$z(\bar{x}) = \bar{w} \cdot \bar{x} + b$$

où $\bar{w} = \{1.661, 0.772, 1.741, 1.098, 0.0263, 2.076, 0.490, 1.062, 1.017\}$ et $b = -3.575$

en utilisant la fonction mathématique $R(z) = \begin{cases} a = 0 & \text{si } z \leq 0 \\ a = 1 & \text{si } z > 0 \end{cases}$.

À partir d'une liste réduite des données de la collection *WDBC*, évaluons nos vecteurs \bar{x} dans l'équation

$$z(\bar{x}) = \bar{w} \cdot \bar{x} + b$$

suivit de la fonction $R(z)$ afin de vérifier que notre séquence de fonction possède un taux de classification de de 100 % pour les patients:

	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	b		
	1,661	0,772	1,741	1,098	0,0263	2,076	0,490	1,062	1,017	-3,575		
ID	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	$z(\bar{x})$	$R(z)$	y
1043999	0,1	0,1	0,1	0,1	0,2	0,3	0,3	0,1	0,1	-2,065	0	0
1044572	0,8	0,7	0,5	1	0,7	0,9	0,5	0,5	0,4	3,332	1	1
1047630	0,7	0,4	0,6	0,4	0,6	0,1	0,4	0,3	0,1	0,220	1	1
1048672	0,4	0,1	0,1	0,1	0,2	0,1	0,2	0,1	0,1	-2,031	0	0
1049815	0,4	0,1	0,1	0,1	0,2	0,1	0,3	0,1	0,1	-1,982	0	0
1050670	1	0,7	0,7	0,6	0,4	1	0,4	0,1	0,2	3,096	1	1
1050718	0,6	0,1	0,1	0,1	0,2	0,1	0,3	0,1	0,1	-1,650	0	0
1054590	0,7	0,3	0,2	1	0,5	1	0,5	0,4	0,4	2,431	1	1

Nous constatons que le taux de classification est de 100 % pour ce sous-ensemble, car la fonction $R(z)$ donne toutes les valeurs y attendues. Cependant, le taux de classification exacte pour l'ensemble de la base de données est de

95.60761346998537 %

ce qui peut occasionner des erreurs de diagnostic pour certaines patientes.

