Réseaux de neurones

Les réseaux de neurones, inspirés de la structure du cerveau humain, sont au cœur des progrès récents de l'intelligence artificielle. Dotés de capacités impressionnantes, ils arrivent à reconnaître des images avec grande précision et sont utilisés dans les voitures autonomes. Ils peuvent lire et écrire et même jouer à des jeux vidéos! Ces réseaux utilisent des principes mathématiques relativement simples pour représenter ces connaissances pourtant complexes. Survolons certains de ces concepts les plus importants.

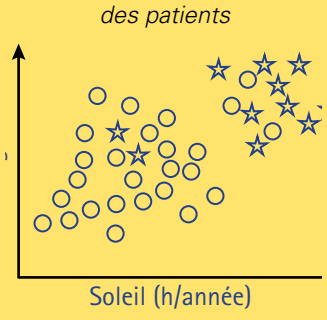
Massimo Caccia
Université de Montréal
Laurent Charlin
HEC Montréal

Détecter les cancers

La Dre Douggie est une médecin-oncologue de renommée mondiale. Chaque jour, elle diagnostique la présence de mélanomes, synonymes de cancer de la peau, chez ses patients. Son expertise est telle qu'elle ne suffit plus à la tâche et son hôpital aimerait pouvoir l'aider en créant une application, un programme informatique, pour identifier automatiquement les mélanomes avec le même taux de succès que la médecin. L'hôpital demande donc son aide pour développer une telle application.

Une première approche

Pour mieux formaliser le processus menant à un diagnostic, la Dre Douggie a une idée. En premier, elle va récolter les données médicales de ses anciens patients, mais pas n'importe quelles données: celles qui selon elle l'aident à prédire si un patient a un mélanome ou non.



Premier jeu de données récolté. L'axe horizontal représente le nombre moyen d'heures d'exposition au soleil par année et l'axe vertical représente l'âge de chaque individu. Les étoiles représentent les patients malades et les ronds ceux en santé.

Dénotons ces données, pour chaque patient, par le vecteur x. Ensuite, elle cherchera une formule mathématique qui, à partir de x, permet de déterminer si un patient est atteint d'un mélanome ou non. Une fois cette formule trouvée, elle pourra être utilisée pour prédire si de nouveaux patients ont un mélanome ou non.

Ce processus, et surtout les méthodes permettant de trouver automatiquement cette formule, se nomme *apprentissage automatique*. La formule elle-même constitue un modèle, c'est-à-dire une représentation mathématique simplifiée du processus de diagnostic. Plus précisément, on cherche une fonction f qui transforme les données x en prédiction y, soit $f(x) = y$. Dans notre cas, y est une variable avec deux valeurs possibles, soit une valeur indiquant la présence d'un mélanome et une autre indiquant son absence. On peut donner la valeur numérique de 1 au premier cas et de 0 au second.

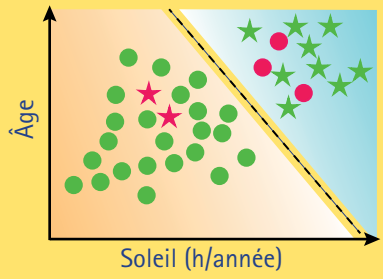
Le nombre d'heures d'exposition au soleil par année ainsi que l'âge du patient sont deux caractéristiques que la Dre Douggie utilise pour établir son diagnostic. Pour chacun de ses patients, elle utilise donc ces deux variables que l'on dénote x_1 et x_2. L'ensemble des caractéristiques de chaque patient constitue alors ce qu'on appelle un *jeu de données*: $\{(x_1, x_2, y)\}$. Ce jeu de données contenant 37 paires est illustré à la figure à gauche.

Après quelques minutes de réflexion, la Dre Douggie trouve une droite entre les variable x_1 et x_2:

$$ax_1 + bx_2 + c = 0$$

artificiels

La frontière de classification linéaire



La frontière de classification suggère de diagnostiquer les patients dans la zone bleue comme malades et déclarer sains ceux dans la zone orange. De plus, les patients en vert auraient bien été diagnostiqués selon la frontière, tandis que ceux en rouge ne le seraient pas.

qui semble bien séparer les patients cancéreux des autres. Par tâtonnement, elle détermine la valeur des paramètres a, b, c qui semblent le mieux séparer les patients selon leur pathologie. Les données et la droite trouvée sont illustrées à la figure ci-dessus.

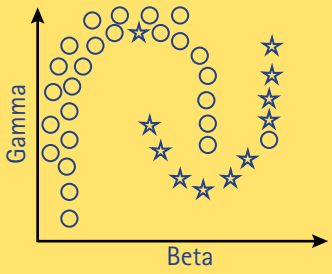
La droite indique ce que l'on nomme la *frontière de classification* : d'un côté, les patients sont présumés avoir un mélanome et de l'autre, non. On peut donc utiliser cette droite pour classifier les patients en suivant la règle suivante:

$$y = \begin{cases} 1 & \text{si } ax_1 + bx_2 + c \geq 0 \\ 0 & \text{sinon} \end{cases}.$$

Dans cet exemple, a et b sont des valeurs positives, puisque les variables x_1 et x_2 sont toutes les deux corrélées positivement avec la présence d'un mélanome. De plus, l'équation est en fait un plan à deux dimensions dans un espace tridimensionnel.

Même si, à première vue, cette fonction semble poser un bon diagnostic, elle se trompe en fait cinq fois. Trois patients en santé auraient été diagnostiqués comme ayant un mélanome (cercles rouges) tandis que deux patients avec mélanome auraient été manqués (étoiles rouges). Puisque la base de données contient 37 patients, l'erreur de classification est de 5/37=13,5%. Encouragée par ces résultats, la Dre Douggie croit cependant qu'il y a lieu d'améliorer ce modèle. Elle décide donc de poursuivre la recherche.

Les patients et leur concentration sanguine



Deuxième jeu de données récolté. L'axe vertical indique la concentration en protéines Gamma des patients, tandis que l'axe horizontal indique celle des protéines Beta.

Un autre essai

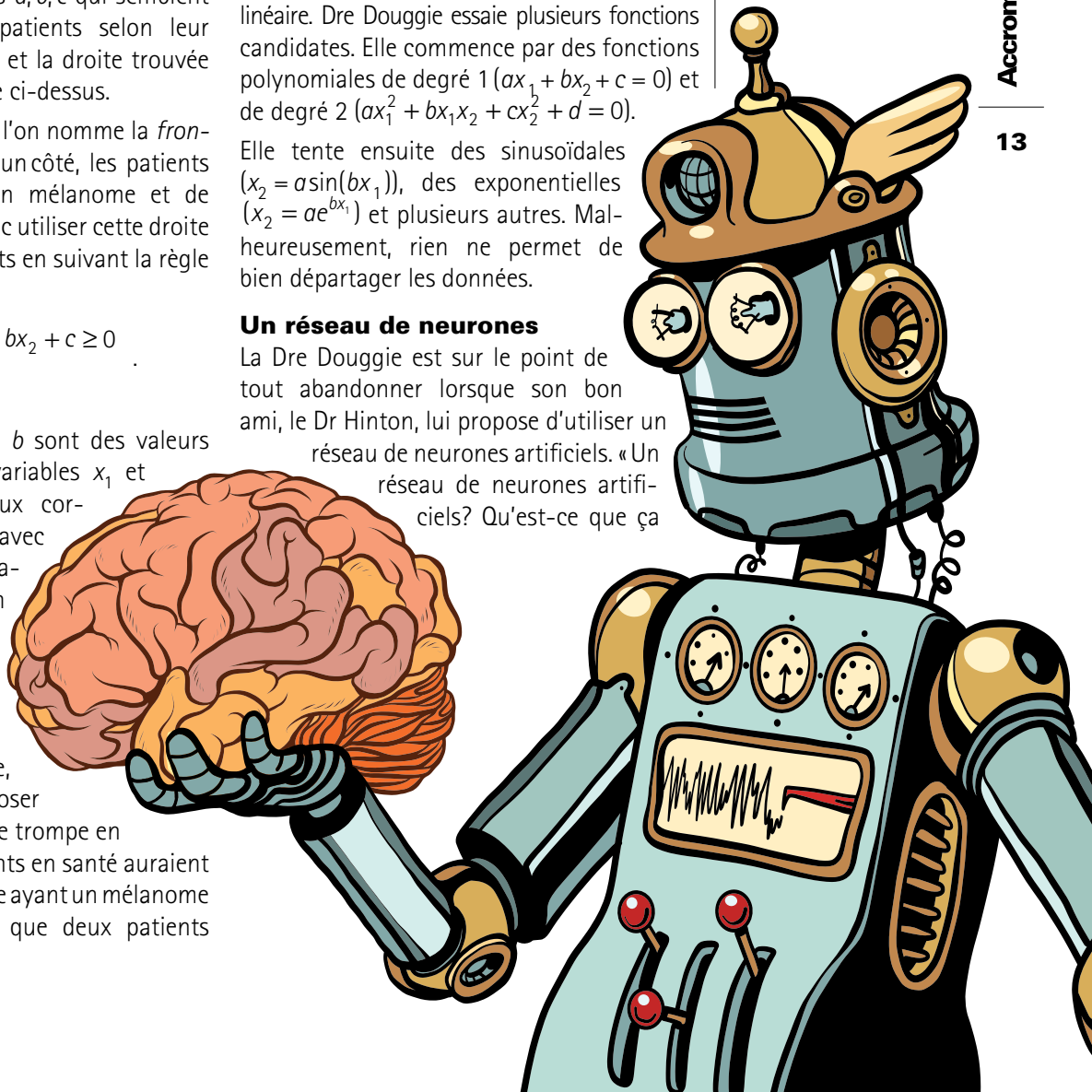
Les concentrations sanguines de deux protéines, Gamma et Beta, l'ont toujours aidée à prédire la présence de la maladie. Cependant, la Dre Douggie n'a jamais réussi à expliquer comment elle interprétait leur niveau pour réaliser son diagnostic. Elle s'est toujours fiée à son intuition, qui est devenue de plus en plus précise avec le temps. La médecin crée donc un deuxième jeu de données, illustré à la figure ci-dessus.

À première vue, il semble exister une fonction (un peu serpentine) qui ne ferait que deux erreurs! Cependant, elle serait loin d'être linéaire. Dre Douggie essaie plusieurs fonctions candidates. Elle commence par des fonctions polynomiales de degré 1 ($ax_1 + bx_2 + c = 0$) et de degré 2 ($ax_1^2 + bx_1x_2 + cx_2^2 + d = 0$).

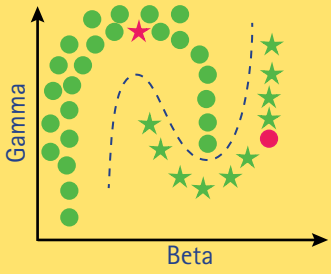
Elle tente ensuite des sinusoïdales ($x_2 = a\sin(bx_1)$), des exponentielles ($x_2 = ae^{bx_1}$) et plusieurs autres. Malheureusement, rien ne permet de bien départager les données.

Un réseau de neurones

La Dre Douggie est sur le point de tout abandonner lorsque son bon ami, le Dr Hinton, lui propose d'utiliser un réseau de neurones artificiels. « Un réseau de neurones artificiels? Qu'est-ce que ça



La frontière de classification d'un réseau de neurones



La frontière identifiée par le réseau de neurones du Dr Hinton

mange en hiver, ça?» demande Dre Douggie. «Eh bien, c'est un type de fonction qui peut lui-même modéliser toutes les fonctions existantes. Tu n'auras plus à émettre et à tester des hypothèses, comme: est-ce qu'une fonction cubique collerait bien à mon jeu de données? Le réseau de neurones s'en occupe, et il n'y a pas grand-chose à son épreuve!» réplique le Dr Hinton.

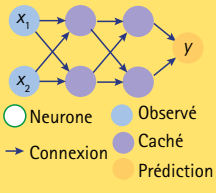
La Dre Douggie reste sceptique; ça lui semble un peu trop beau pour être vrai. Tout de même, elle laisse le Dr Hinton essayer ses réseaux de neurones sur son jeu de données. Quelques minutes plus tard, le Dr Hinton aboutit à la fonction illustrée à la figure ci-dessus.

La Dre Douggie est bouche bée. Elle a passé une semaine à essayer d'identifier **LA** fonction parfaite et n'a abouti à rien. Le Dr Hinton l'a trouvée quasi instantanément grâce à ses réseaux de neurones. L'erreur de classification, qui était 13,5 %, descend donc à 2/37 = 5,4 %.

Comment ça fonctionne?

Le schéma ci-contre illustre le réseau de neurones utilisé par le Dr Hinton. Les deux éléments au cœur de ce schéma sont d'une part les *neurones* (cercles) et de l'autre les *connexions* (flèches). Les connexions transmettent l'information entre les neurones qui, pour leur part, procèdent à des calculs à partir des informations reçues. L'information se transmet dans le sens des connexions, à partir des neurones observés qui correspondent aux données jusqu'à la prédiction.

Un réseau de neurones



La force de ces modèles est de combiner les neurones, chacun représentant une fonction relativement simple, pour obtenir des fonctions très complexes.

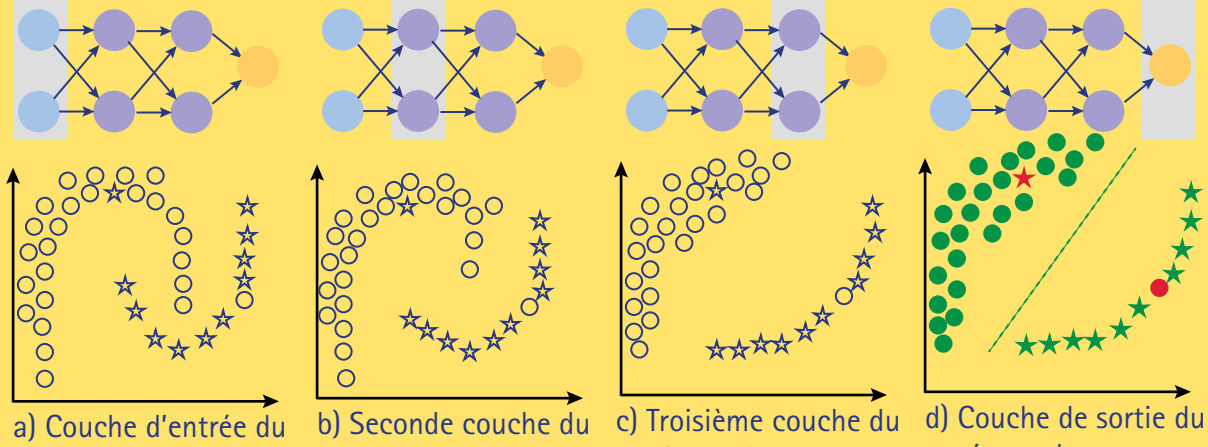
Décortiquons ce qui se passe lorsqu'un réseau de neurones fait une prédiction. Rappelons-nous qu'un réseau de neurones est une fonction $f(x) = y$, où x représente les caractéristiques d'un patient du jeu de données et où y est la prédiction.

La première étape est plutôt simple: on insère les données x dans le réseau de neurones f. Dans notre cas, c'est la paire de concentrations (Gamma, Beta) que nous insérons dans les neurones observés (en bleu) du réseau. La figure illustre cette étape.

Les deuxième et troisième étapes sont très intéressantes. Une par une, les couches transforment le jeu de données de façon à ce qu'on puisse linéairement séparer les ronds des étoiles. Plus précisément, les connexions vont créer un nouvel *espace vectoriel* et y transporter le jeu de données de façon à ce que les cercles soient séparés des étoiles.

Ces calculs – qui utilisent des opérations simples comme des multiplications et des additions – ont lieu dans chaque neurone. On peut visualiser le procédé dans les parties b) et c) de la figure en bas de page. Le jeu de données transformé est donc «stocké» dans les neurones cachés (en mauve) du réseau. Nous les appelons ainsi puisque ce sont des représentations du jeu de données que nous n'observons pas dans le monde réel. D'un autre point de vue, les axes qui

Les réseaux de neurones transforment les données



a) Couche d'entrée du réseau de neurones

b) Seconde couche du réseau de neurones

c) Troisième couche du réseau de neurones

d) Couche de sortie du réseau de neurones

correspondaient aux protéines Gamma et Beta en a) deviennent l'intensité de l'activation des deux neurones en b) et c). Pour en savoir plus sur l'activation des neurones, voir l'encadré « Activation des neurones ».

La dernière étape peaufine le diagnostic: on retrouvera dans le neurone jaune la probabilité qu'un patient ait une tumeur cancéreuse. Le réseau de neurones trace une frontière linéaire pour séparer les patients. S'il estime qu'un patient est en santé, les étapes 2 et 3 le déplaceront bien à la gauche de la frontière, que l'on peut voir à la partie d) de la figure. S'il estime que le patient est malade, les étapes 2 et 3 le déplaceront bien à la droite. Par contre, si le réseau est moins confiant en son diagnostic, il déplacera le patient plus près de la frontière. C'est pourquoi on peut interpréter la distance entre la frontière et un patient comme la confiance du réseau de neurones en son diagnostic. Par exemple, si un patient est exactement sur la frontière, c'est que le réseau est indécis.

Les réseaux de neurones excellent dans la vision par ordinateur qui cible la compréhension automatique d'informations visuelles. Par exemple, un réseau de neurones pourrait reconnaître la race d'un chien à partir de sa photo. Un réseau de neurones pourrait également décrire une image comme celle ci-contre.

Un réseau de neurones annote les entités dans une photo.

Photo parue dans l'article AutoML for Large Scale Image ClassifAication *de Zoph* et al.

Les réseaux de neurones peuvent aussi générer des images on ne peut plus réalistes. Sauriez-vous discerner la vraie photo de celle générée par un réseau de neurones dans la figure en haut de la page suivante ?

Les réseaux utilisés dans de telles applications sont de très grande taille. Dans le problème étudié par la Dre Douggie, les données x qui entrent dans le réseau de neurones ont deux dimensions (concentration des protéines Gamma et Beta). Une image d'une télévision haute définition contient plus de 900 000 pixels, donc 900 000 dimensions! De plus, la

Activation des neurones

Un peu comme dans nos cerveaux, dans un réseau de neurones artificiels, les neurones s'activent ou non en fonction des signaux reçus. Plus précisément, chaque neurone calcule une transformation affine de ses entrées. Chaque entrée est multipliée par un coefficient unique aussi appelé *poids*. Par exemple un neurone ayant deux entrées calcule donc :

$$p = w_0 + w_1 x_1 + w_2 x_2$$

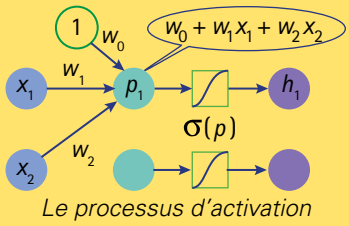
où l'ordonnée à l'origine w_0 est aussi appelée *biais*.

Ensuite, une fonction non linéaire transforme la sortie o de chaque neurone. Cette fonction non linéaire, souvent appelée *fonction d'activation*, est primordiale pour le réseau de neurones puisqu'elle permet au réseau d'apprendre une frontière de classification non linéaire (comme à la figure à droite par exemple). Un exemple d'une telle fonction souvent utilisée est la fonction sigmoïde :

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

Maintenant que nous avons compris le fonctionnement d'un neurone, reprenons le réseau du Dr Hinton. La figure montre comment a été calculé le premier neurone de la première couche cachée.

Les connexions w_1 et w_2 ainsi que le biais w_0 combinent linéairement x_1 et x_2 en une sortie préactivation p_1. Ensuite, la fonction sigmoïde calcule l'activation du neurone h_1. Si la valeur de p_1 est suffisamment petite, le neurone ne sera pas activé ($h_1 \sim 0$). Par contre, si la valeur de p_1 est grande, le neurone h_1 sera actif est sa valeur de sortie se rapprochera de 1.



Le processus d'activation d'un neurone

Dans un réseau, les signaux se déplacent une couche à la fois en partant des données. Les neurones de la première couche cachée calculent donc une transformation des données x_i. Les neurones de la seconde couche cachée utilisent les sorties des neurones de la première couche. Et ainsi de suite jusqu'à la sortie du réseau...

Photos véritables ou générées par un réseau de neurones?[1]

Paru dans l'article Progressive Growing of GANs for Improved Quality, Stability, and Variation *de Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen.*

profondeur des réseaux (nombre de couches cachées) peut aller jusqu'à 150 couches! Le nombre total de paramètres de ces réseaux peut être gigantesque.

Les réseaux de neurones réalisent aussi des performances impressionnantes en reconnaissance vocale. Lorsque vous parlez avec Alexa (Amazon), Siri (Apple), Cortana (Microsoft) ou à l'Assistant Google, les réseaux jouent un rôle clé quant à la reconnaissance des mots. Et comment toutes ces technologies produisent-elles une réponse à la question ou à la demande formulée par un utilisateur? Vous l'aurez deviné, avec d'autres réseaux de neurones!

Les réseaux de neurones peuvent conduire des automobiles, jouer à des jeux vidéos, découvrir de nouveaux médicaments, écrire de la musique, peindre des tableaux, traduire entre plusieurs langues, contrôler des robots, etc. Bref, les réseaux de neurones ne cessent de nous épater par leur polyvalence et le nombre de scientifiques s'y intéressant en divers domaines progresse à toute allure. D'ailleurs, bien malin qui sait tout ce qu'ils pourront nous aider à résoudre. Les réseaux de neurones ont révolutionné le domaine de la vision informatique.

Les réseaux de neurones, c'est un peu canadien!

L'idée des réseaux de neurones artificiels a vu le jour en 1943. Leur popularité a fluctué avec le temps, mais ce n'est qu'au début des années 2000, après des années de recherche, que Geoff Hinton, de la University of Toronto, Yoshua Bengio, de l'Université de Montréal, et Yann LeCun de la New York University ont pu démontrer les capacités de ces réseaux à apprendre des tâches complexes. L'Institut canadien de recherches avancées (ICRA) les a subventionnés, lorsque les entreprises et les autres pays avaient baissé les bras. Aujourd'hui, le terme *apprentissage profond* désigne le champ portant sur les réseaux de neurones.

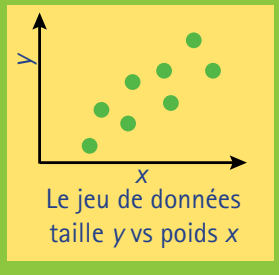
*Les « pères » de l'*apprentissage profond *: Yan LeCun, Geoffrey Hinton et Yoshua Bengio (tiré de la page Facebook de Andrew Ng)*

1. Les deux photos ont été imaginées par un réseau de neurones.

Comment le réseau apprend-il à transformer les données ?

Tout ça est bien beau, mais comment le réseau de neurones *apprend*-il une tâche ? Plus précisément, comment le réseau trouve-t-il les connexions pour correctement déplacer les patients en santé d'un côté de la frontière de décision, et les patients malades de l'autre. La réponse est : avec l'aide du puissant algorithme de *descente du gradient*.

Prenons un exemple très simple pour expliquer cet algorithme. Disons que l'on veut prédire la taille d'un individu à partir de son poids (voir le jeu de données à la figure suivante). Un modèle très simple pourrait être $y = f(x) = ax$ où y représente la taille d'un individu et x son poids.



Le jeu de données taille y vs poids x

Notons que a est la pente de la droite. Nous voulons donc trouver la pente qui nous permettra de prédire la taille d'un individu le plus précisément possible. Ce problème, bien qu'un peu différent de la détection du cancer, peut aussi être résolu par un réseau de neurones qui transforme les données. On peut utiliser la *descente de gradient* pour trouver la valeur de a qui donne les prédictions les plus justes. La technique est illustrée à la figure « Descente de gradient » (en bas). Sur la rangée du haut, on montre le jeu de données (en vert) en plus de différentes droites possibles (en mauve). Les lignes pointillées en mauve montrent l'écart entre la prédiction et le poids réel d'un individu. L'erreur commise par chaque droite est la somme de toutes ces distances. Sur la rangée du bas, on montre l'erreur totale de chaque droite. En d'autres mots, la courbe verte représente l'erreur totale en fonction de la pente de la droite.

De plus, cette figure montre l'évolution de la droite tout au long de l'entraînement. Au début de l'*apprentissage* (coin gauche supérieur), on essaie une pente au hasard. L'erreur (en mauve) du modèle est donc relativement élevée. Le but est de changer la pente pour minimiser l'erreur. Pour trouver ce minimum, nous utilisons la dérivée de la fonction d'erreur (flèche noire). La dérivée est une fonction qui pointe vers le minimum d'une autre fonction. Si la dérivée est négative, c'est que le minimum se situe vers la droite, et vice-versa. Dans notre cas, elle pointe vers la gauche. Pour cette raison, on essaie une pente plus petite et on répète le processus. On arrête lorsque le minimum est trouvé, soit lorsque la dérivée ne pointe ni vers la gauche, ni vers la droite (elle est égale à 0).

Cet exemple est très simple puisqu'on n'entraîne qu'un seul paramètre a. Pour trouver les bonnes connexions dans un réseau de neurones, des milliers, voire des millions de paramètres sont entraînés simultanément en utilisant cette méthode de descente de gradient !

Descente de gradient