

mée suite de Fibonacci). Toutefois on peut aussi bien prétendre que le nombre suivant est n'importe quel autre nombre, car une infinité de fonctions f prennent les valeurs 1, 1, 2, 3, 5, 8 pour les arguments 1, 2, 3, 4, 5, 6. La réponse

$f(7) = 13$ nous paraît seulement plus «naturelle», plus «raisonnable» : on s'attend à ce que la règle cachée soit «simple». Peut-on comprendre cette faculté que nous avons à généraliser, autrement dit à inférer une règle à

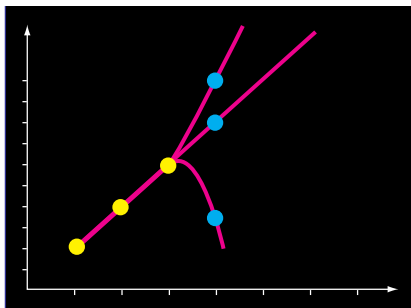
partir d'un nombre limité d'exemples? Peut-on comprendre pourquoi certaines généralisations paraissent plus naturelles, plus simples, que d'autres?

L'analyse des réseaux de neurones a éclairé cette question. Mieux encore, ces réseaux de neurones ont été généralisés en «machines à apprendre» avec lesquelles on cherche à comprendre le fonctionnement du cerveau. De telles machines, réelles ou simulées sur ordinateur, sont décrites par quelques équations ; elles sont spécialisées dans la tentative de résolution de tâches analogues à la recherche de termes d'une suite. On admet qu'elles peuvent lire des questions, c'est-à-dire saisir des données dans un format prédéterminé ; puis elles répondent aux questions qu'on leur pose.

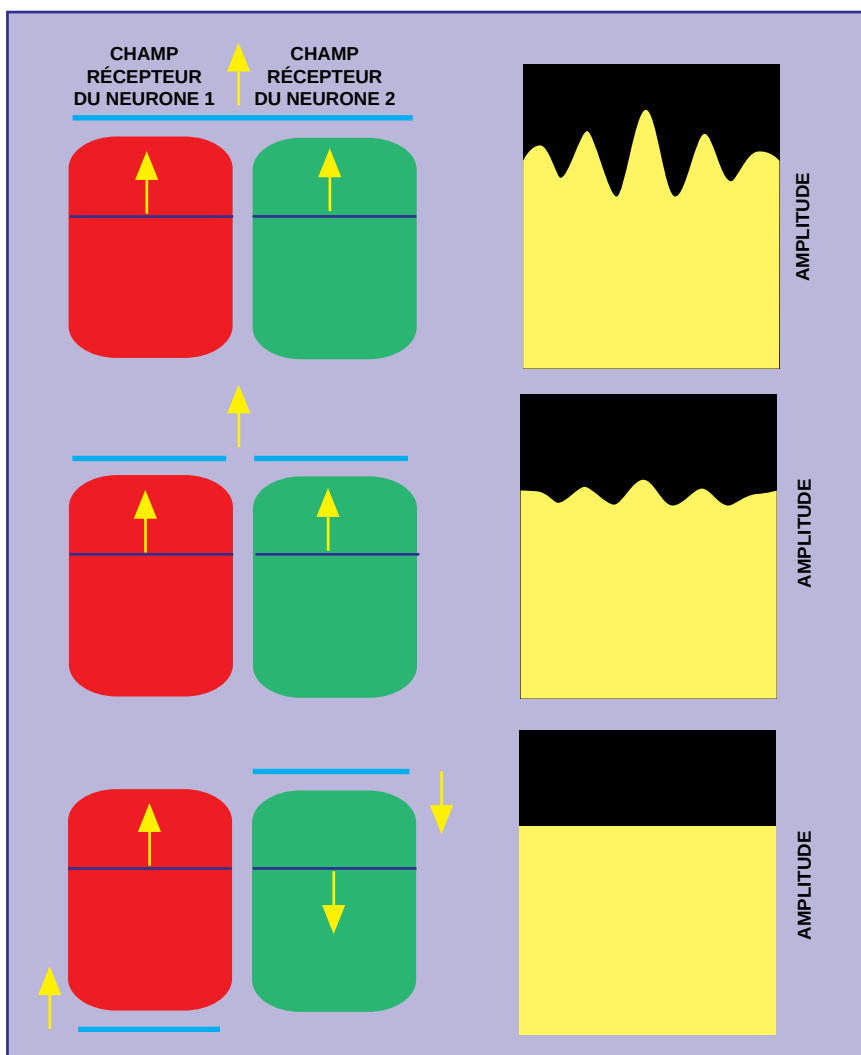
Par exemple, vers la fin des années 1980, une équipe des Laboratoires Bell de la Société AT&T a mis au point des réseaux de neurones pour la reconnaissance de chiffres manuscrits de codes postaux. L'équipe présentait des photographies numérisées de caractères manuscrits en entrée d'un réseau à trois couches cachées, et elle faisait apprendre au réseau à produire la sortie appropriée : le premier neurone de sortie devait être actif quand le réseau reconnaissait le chiffre 0, le deuxième neurone devait être actif quand le chiffre 1 était reconnu, etc.

Plus de 7 300 chiffres étaient utilisés pour l'apprentissage. Après cette phase d'apprentissage, l'équipe présentait de nouveaux chiffres (environ 2 000) pour tester les performances en généralisation : le réseau afficherait-il la bonne sortie pour ces exemples non vus au cours de l'apprentissage? Le résultat fut assez satisfaisant : la proportion d'erreurs sur ces exemples tests était seulement de cinq pour cent (sur les mêmes données, un être humain fait moitié moins d'erreurs seulement). Plus récemment, avec un nombre de chiffres manuscrits bien supérieur (60 000 pour l'apprentissage et 10 000 pour le test), et avec des architectures et des algorithmes d'apprentissage plus complexes, la même équipe a obtenu un taux d'erreurs inférieur à un pour cent.

Dans ce type d'application, on modifie les poids synaptiques à l'aide d'algorithmes comparables à l'algorithme d'apprentissage précédemment évoqué. Toutefois, dans le cadre des neurosciences computationnelles, on considère



5. LA CAPACITÉ DE GÉNÉRALISER est utilisée dans les tests de QI. On présente une série de nombres telle que 2, 4, 6, et l'on demande le nombre suivant. Nous avons tendance à penser que ce nombre doit être 8, parce que les premiers termes sont les premiers nombres pairs de la suite. Pourtant, en théorie, n'importe quel nombre est possible, parce que plusieurs courbes peuvent passer par les points imposés. Par exemple, si l'on observe que 6 est égal à la somme des deux termes précédents, on peut ainsi poursuivre la suite par le nombre 10.



6. LA DYNAMIQUE des réseaux de neurones est une caractéristique importante de leur fonctionnement. Wolf Singer, à Francfort, a enregistré l'activité des neurones d'un singe à qui il montrait une barre (en bleu clair) qui se déplaçait. Il enregistrait alors des réactions synchrones de certains neurones (la courbe de corrélation à droite est accentuée quand les décharges des neurones sont synchrones) (a). Deux barres mobiles excitant indépendamment les deux neurones suscitent une réaction inférieure (b). Enfin, deux barres indépendantes qui se déplacent dans des directions opposées n'engendrent aucune réaction (c).

plus particulièrement des algorithmes d'apprentissage ayant une certaine plausibilité biologique : les poids synaptique sont changés selon un critère hebbien, c'est-à-dire en fonction des activités pré- et postsynaptiques.

Ainsi, après une phase d'apprentissage qui consiste à modifier les paramètres, une machine généralise – elle a compris la règle – si elle donne la réponse appropriée pour tout exemple qui n'a pas été considéré lors de l'apprentissage.

Questions de ressources

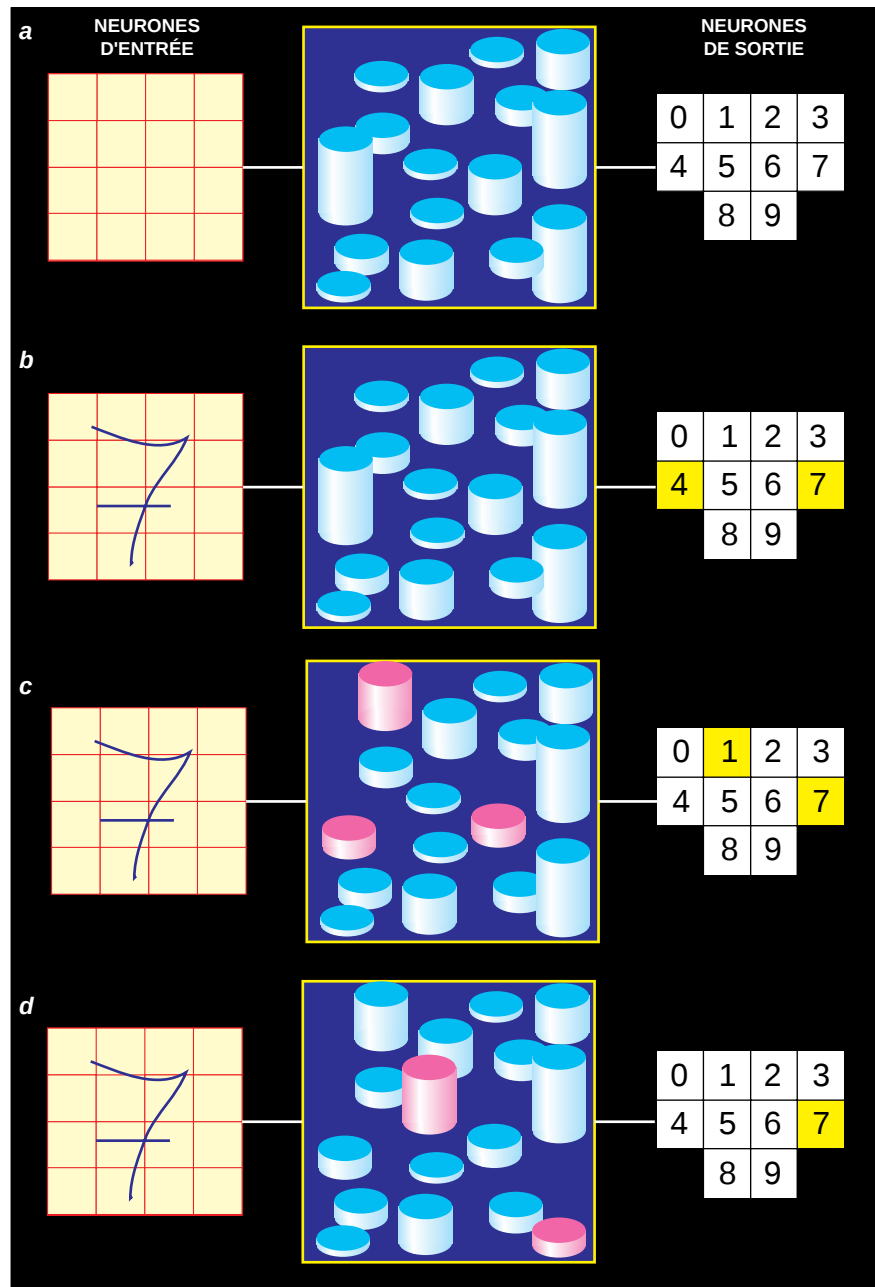
Supposons que l'on arrive effectivement à adapter les paramètres, de sorte que la machine produise la sortie désirée pour chaque exemple donné lors de l'apprentissage : ce serait la preuve que, pour chaque exemple, la machine a mémorisé l'association entrée-sortie désirée. Une telle mémorisation est possible si les ressources de la machine sont suffisantes, c'est-à-dire si l'architecture est appropriée et si le nombre de paramètres adaptables est suffisamment grand. Cependant, si ces ressources sont vraiment très grandes, la machine apprend par cœur tout nouvel exemple donné : quelle que soit la sortie désirée, il existera toujours un choix des paramètres qui permet d'obtenir cette sortie, tout en gardant les bonnes sorties pour tous les exemples déjà appris. La machine ne généralise alors pas, puisque à chaque instant elle peut mémoriser un nouvel exemple.

En revanche, si les ressources de la machine sont limitées, la mémorisation d'un grand nombre d'exemples risque d'encombrer la mémoire, de sorte que, lors de la présentation d'un nouvel exemple, on ne peut plus modifier les paramètres sans remettre en cause l'apprentissage déjà effectué. Autrement dit, avant que soit donnée la réponse à reproduire (la sortie désirée), la machine a déjà une réponse prête, la seule possible compte tenu de ce qu'elle a déjà appris. En ce sens, la machine généralise.

Plus on donne d'informations sur la règle à découvrir, plus on restreint le domaine des valeurs admissibles pour les paramètres, mais plus on limite le nombre de règles nouvelles que la machine peut encore apprendre. Si l'architecture est adaptée, la machine peut apprendre la règle cachée : il existe au moins un choix des paramètres qui per-

met à la machine de reproduire la règle ; dans ce cas, la généralisation sera correcte. Sinon, la machine généralise, mais infère une mauvaise règle : la règle trouvée ne coïncide que partiellement avec celle qu'il fallait trouver.

Ainsi la capacité à généraliser, à catégoriser, commence là où s'arrête la capacité à apprendre par cœur, et un système aux ressources illimitées ne peut qu'apprendre bêtement. Cette notion de ressources est subtile, car on



7. LA RÉTROPROPAGATION DU GRADIENT est un algorithme d'apprentissage très utilisé. Son principe est schématisé ici. Après le choix de l'architecture du réseau, par exemple un réseau en couches avec une couche cachée (*non représenté*), on modifie les connexions entre les neurones (*au centre, on a représenté les poids synaptiques par des plots de différentes hauteurs*) pour que le réseau affiche une sortie désirée. On définit une fonction dite de coût, qui décrit la différence entre l'activité de sortie et l'activité désirée. Cette fonction est nulle si la sortie est celle que l'on désire. On présente alors des exemples au réseau, et, à chaque présentation, on modifie les connexions, de sorte que la sortie se rapproche de la sortie désirée. Par exemple, on présente ici un chiffre 7 ; le réseau réagit d'abord en affichant un 4 et un 7. On modifie alors les poids synaptiques (*en rouge, c*) et l'on présente à nouveau le chiffre 7 : le réseau réagit en affichant 7 et 1. Puis, quand on modifie encore les poids (*d*), le réseau détecte le 7. De proche en proche, la sortie se rapproche de la sortie voulue.