



## **Jouez avec les neurones de la machine**

Thalita F Drumond, Laurent Viennot, Thierry Viéville, Valérie François

### **► To cite this version:**

Thalita F Drumond, Laurent Viennot, Thierry Viéville, Valérie François. Jouez avec les neurones de la machine. 2017, pp.1-3. hal-01620451

**HAL Id: hal-01620451**

**<https://inria.hal.science/hal-01620451>**

Submitted on 23 Oct 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

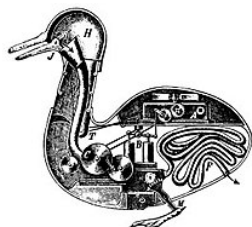
## binaire

L'informatique : science et technique au cœur du numérique

Avec la Société Informatique de France

20 octobre 2017

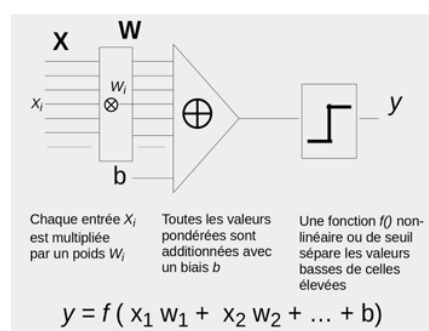
## Jouez avec les neurones de la machine



Le canard artificiel de Vaucanson (1738), qui « boit, mange, cancanne, barbote et digère comme un vrai canard »

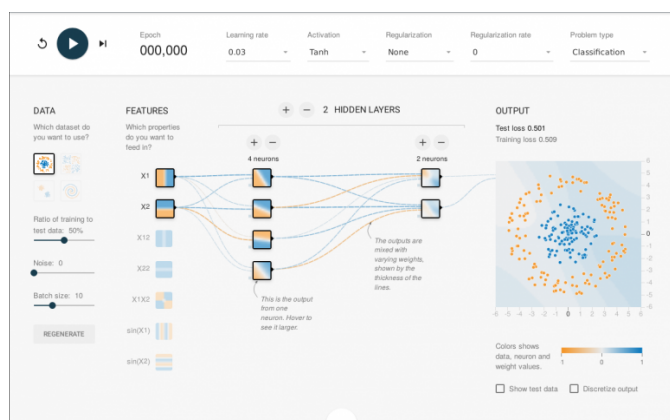
« L'intelligence artificielle est la science de faire faire à des machines des choses qui demanderaient de l'intelligence si elles étaient faites par des humains ». Tout est dit par le [fondateur de l'intelligence artificielle](#), [Marvin Minsky](#). Exit les fantasmes du genre\* de celui d'une « servante-robot, qui sert [le] café [au lit] le matin ». Et comme le mentionne Cédric Villani au [lancement de sa mission de réflexion](#) sur ces sujets, notre meilleure arme est « une grande qualité de formation » sur ce sujet qui est « l'affaire de [toutes et] tous ». Soit. Et si on commençait, là, maintenant ? Ça vous dirait de soulever le capot de l'intelligence artificielle ? Thierry Viéville.

Un réseau de neurones est un mécanisme générique composé de petites unités (des pseudo-neurones) connectées les unes aux autres. Chaque unité fait une opération très simple : elle prend des valeurs en entrée, les combine très simplement (un simple calcul de moyenne avec des coefficients comme les notes du bac) et applique une transformation sur le résultat (par exemple ne garde que les valeurs positives). Les coefficients utilisés pour pondérer la moyenne sont les paramètres de cet algorithme. C'est la combinaison d'un très, très grand nombre de ces unités qui permet de réaliser des opérations très complexes. Un réseau de telles « neurones » s'obtient en accumulant plusieurs couches de telles unités. En entrée il y a les données que l'on veut traiter. Elles se transforment à travers toutes les couches et la dernière couche donne en sortie une prédiction sur ces données, par exemple détecter s'il y a un visage dans une image. Le réseau de neurone constitue ainsi une fonction paramétrée par ces nombreux coefficients (on parle de « poids ») et c'est le choix de ces poids qui définit le traitement effectué.



Chaque « neurone » mélange les entrées  $X$  de manière proportionnelle à ses poids  $W$  et rectifie le résultat pour donner la sortie  $y$ . C'est la combinaison de miriades de tels calculs élémentaires qui génère un système complexe. Issu de [Interstices](#).

Sur l'interface web de [TensorFlow](#), on constitue facilement un réseau d'une douzaine de neurones possédant chacun entre 3 et 10 paramètres. La sortie calculée dépend donc d'une centaine de paramètres en plus des deux coordonnées  $(x,y)$  du point d'entrée. Sur l'interface, chaque carré représente un neurone et la couleur du pixel de coordonnées  $(x,y)$  dans le carré représente la sortie du neurone quand on met  $(x,y)$  en entrée du réseau. Il y a un seul neurone en sortie, il est représenté avec un carré plus grand sur la droite du réseau. Les paramètres du réseau sont initialisés au départ avec des valeurs aléatoires.



Mais comment apprendre ces poids ? L'apprentissage supervisé consiste à fournir des exemples de données accompagnés de la solution à trouver, pour entraîner le réseau à ajuster ces poids comme il faut. Ici, il s'agit d'une série de points dans un carré avec pour chacun une couleur attendue (bleu ou orange), avec comme but de prédire la couleur du point à un endroit donné. C'est un algorithme classique d'ajustement progressif des poids (on parle de « descente de gradient ») qui permet de trouver les paramètres en question. Le bouton « play » en haut à gauche de l'interface permet de lancer cet algorithme, on voit alors la sortie du réseau de neurones évoluer au cours de l'« apprentissage » : la couleur du fond du neurone de sortie tend à prendre la couleur des points d'entraînement qui sont dessinés par-dessus. Une autre partie du jeu de données est ensuite utilisée pour tester la qualité de la fonction obtenue par le réseau. Une courbe en haut à droite affiche le taux d'erreurs liées aux données utilisées pour apprendre (pour vérifier que les poids se sont bien ajustés) et le taux d'erreurs liées aux autres données de test (pour vérifier que ce qui a été appris se généralise bien à de nouvelles données). Des boutons sur la gauche permettent de régler la répartition des données entre le jeu d'apprentissage et celui de test et aussi d'ajouter des erreurs aux données (les brouter) pour voir si le mécanisme est robuste face à ces erreurs.

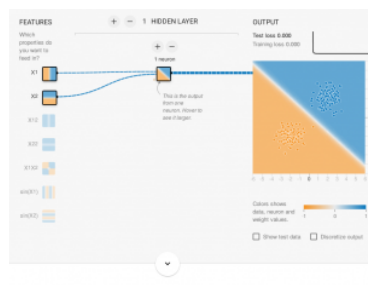


© Casterman

En pratique, on arrive à trouver des paramètres satisfaisants, mais il n'y a pas vraiment de cadre théorique pour formaliser tout cela, c'est affaire d'expérience : choisir le bon nombre de neurones, le bon nombre de couches de neurones, quels calculs préliminaires ajouter en entrée (par exemple multiplier les entrées pour augmenter les degrés de liberté permettant de faire le calcul). Ce genre de techniques permet d'obtenir des résultats impressionnants en pratique, comme en reconnaissance de la voix ou d'objets dans une image (voir les [vidéos](#) du cours de Yann Le Cun au Collège de France). Cependant, comprendre pourquoi (et comment) on obtient de si bons résultats reste une question scientifique encore assez ouverte. En attendant voulez-vous essayer une application réelle sur des données réelles ? C'est par ici <https://www.clarifai.com/demo> (il suffit de jouer avec les options).

A vous de tester votre petite cuisine neuronale ! Arriverez-vous à mettre les bonnes couleurs sur le jeu de données en spirale ?

## Quelques exemples

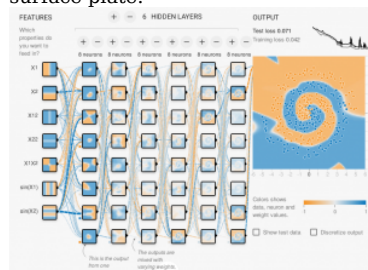


Si on choisit de simplement classer deux populations, ici orange et bleue, qui sont déjà groupées en deux blocs facilement séparés par une ligne, alors c'est très facile : il suffit de trois neurones. En cliquant sur l'image on peut lancer la solution et observer l'ajustement des paramètres de chaque unités de calcul, ces pseudo neurones.

Les deux premiers neurones de la couche d'entrée calculent pour la position horizontale et verticale et le neurone de sortie combine ses informations pour faire une sortie oblique.

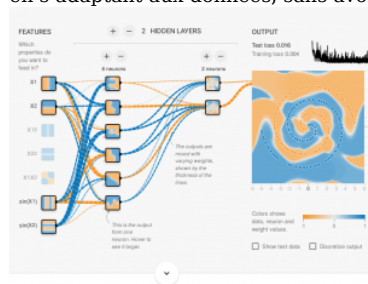
On parle de problème linéaire pour décrire un tel problème de classification dont la solution est une simple séparation par une

surface plate.



Mais si les données sont complètement intermélées comme dans le cas de ces données en spirale alors il faut beaucoup plus de couches de calcul avec plus de neurones.

On voit alors comment au fil du calcul dans les couches de l'architecture, chaque neurone code pour un aspect de la forme à trouver, basique dans les couches basses, plus sophistiqué dans les couches plus hautes. Beaucoup de calculs pour une simple paire de spirales ! Mais ce qui est remarquable c'est que cette foule de calculs élémentaires a pu coder approximativement un objet non trivial, en s'adaptant aux données, sans avoir eu à donner des informations à priori sur ces formes spiralées.



Faut-il tous ces neurones pour résoudre ce problème de classification de données ? Pas forcément : lors d'un atelier lycéen où des jeunes expérimentaient ce logiciel à travers l'interface Web, en expérimentant numériquement les solutions possibles, deux d'entre eux Akmal et Amandine ont obtenu une solution plus longue à obtenir et moins stable qu'une solution gourmande en nombre de neurones, mais qui marche plutôt bien.

Et vous avez vous une solution encore plus astucieuse à proposer ?

---

**Thalita Firmo-Drumond à partir d'une proposition de Laurent Viennot éditée par Valérie François et publiée conjointement sur [pixees.fr](http://pixees.fr).**

(\*) On trouve hélas cette métaphore sexiste et anti-pédagogique au possible (sans aucune explication du propos qui serait au deuxième degré), en introduction du [livre blanc sur l'intelligence artificielle](#) produit par un institut de recherche de référence sur ces sujets.