

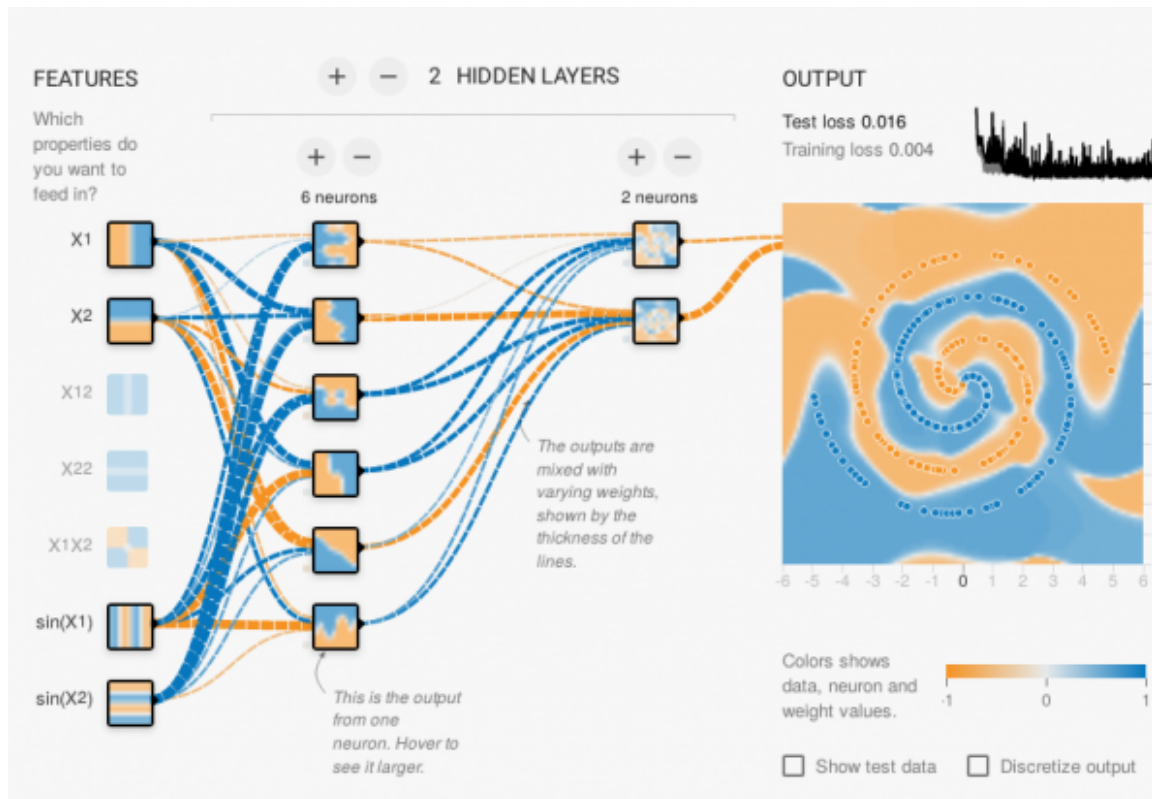


### Metadata

- Id: EU.AI4T.O1.M3.2.5a
- Title: 3.2.5 Activité : Jouez avec les neurones de la machine
- Type: activity
- Description: Comprendre comment fonctionne un réseau de neurones artificiel
- Subject: Artificial Intelligence for and by Teachers
- Authors:
  - AI4T
  - Site Web Pixees
- Licence: CC BY 4.0
- Date: 2022-11-15

## ACTIVITÉ : JOUEZ AVEC LES NEURONES DE LA MACHINE

Le logiciel en ligne [TensorFlow](#) permet de construire des réseaux de neurones artificiels et de tester leurs réponses pour différents types de problèmes et sur différents types de données. Dans le type de problème "Classification", l'objectif est de séparer les points de couleur bleue et orange. Une application de ce type de problème est, par exemple, un algorithme de classification de photos. Dans l'exemple ci-dessous, il y a une entrée (caractéristique) qui sépare les points horizontalement et une autre qui les sépare verticalement. En combinant ces deux entrées, on obtient une séparation oblique. Le résultat (sortie) est bien adapté au type de données choisi.



Vue de l'interface de TensorFlow

## TENSORFLOW : QUELQUES EXPLICATIONS AVANT D'ESSAYER LA SIMULATION D'UN RÉSEAU DE NEURONES

Source : [Site web Pixees](#)

**Qu'est-ce qu'un réseau de neurones et comment fonctionne-t-il?** Un réseau de neurones est un mécanisme générique composé de petites unités (pseudo-neurones) connectées les unes aux autres. Chaque unité effectue une opération très simple : elle prend des valeurs en entrée, les combine très simplement (une simple moyenne avec des coefficients) et applique une transformation sur le résultat (par exemple, ne garder 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 tels "neurones" est obtenu en accumulant plusieurs couches de ces unités. L'entrée est constituée des données que l'on veut traiter. Elles sont transformées à travers toutes les couches et la dernière couche donne en sortie une prédiction sur ces données, par exemple pour détecter s'il y a un visage dans une image. Le réseau de neurones est donc une fonction paramétrée avec de nombreux coefficients (appelés "poids") et c'est le choix de ces poids qui définit le traitement effectué.

**Où sont les neurones dans TensorFlow ?** Sur l'interface web de TensorFlow, on peut facilement créer un réseau d'une dizaine de neurones, chacun ayant entre 3 et 10 paramètres. La sortie calculée dépend donc de centaines 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 lorsqu'on met  $(x,y)$  en entrée du réseau. S'il n'y a qu'un seul neurone en sortie, il est représenté par un carré plus grand à droite du réseau. Les paramètres du réseau sont initialisés avec des valeurs aléatoires.

**Mais comment apprendre ces poids ?** L'apprentissage supervisé consiste à fournir des exemples de données avec la solution à trouver afin d'entraîner le réseau à ajuster ces poids selon les besoins. Dans l'exemple de la figure ci-dessus, il s'agit d'une série de points dans un carré, chacun avec une couleur attendue (bleu ou orange), dans le but de prédire la couleur du point à un endroit donné. Un algorithme classique d'ajustement progressif des poids est utilisé pour trouver les paramètres en question.

Le bouton "play" en haut à gauche de l'interface permet de lancer cet algorithme, et l'on voit alors la sortie du réseau de neurones évoluer au cours du processus "d'apprentissage" : la couleur de fond du neurone de sortie tend à prendre la couleur des points d'entraînement qui sont dessinés sur lui. Une autre partie de l'ensemble de données est ensuite utilisée pour tester la qualité de la fonction résultante du réseau. Une courbe en haut à droite montre le taux d'erreur des données utilisées pour l'apprentissage (pour vérifier que les poids ont été correctement ajustés) et le taux d'erreur des autres données de test (pour vérifier que ce qui a été appris se généralise bien à de nouvelles données). Les boutons sur le côté gauche permettent d'ajuster la distribution des données entre l'ensemble d'apprentissage et l'ensemble de test et aussi d'ajouter des erreurs aux données (données bruitées) pour voir si le mécanisme est robuste à ces erreurs.

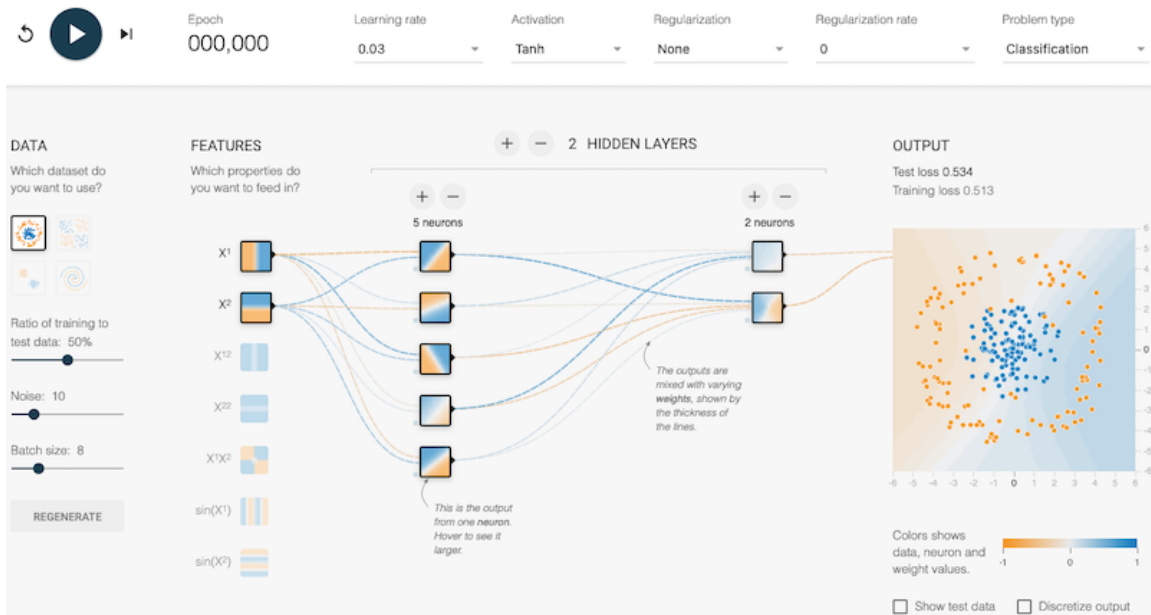
En pratique, on arrive à trouver des paramètres satisfaisants, mais il n'y a pas vraiment de cadre théorique pour formaliser tout cela, c'est une question d'expérimentation : choisir le bon nombre de neurones, le bon nombre de couches de neurones, quels calculs préliminaires ajouter en entrée (par exemple multiplier les entrées pour augmenter les degrés de liberté du calcul).

Ce type de techniques peut donner des résultats impressionnants dans la pratique, comme dans la reconnaissance de la voix ou des objets dans une image.

Toutefois, comprendre pourquoi (et comment) on obtient de si bons résultats reste une question scientifique assez ouverte.

## ESSAYEZ TENSORFLOW

*Cliquez sur l'image ci-dessous pour accéder à l'application TensorFlow dans une nouvelle fenêtre.*



TensorFlow playground view