

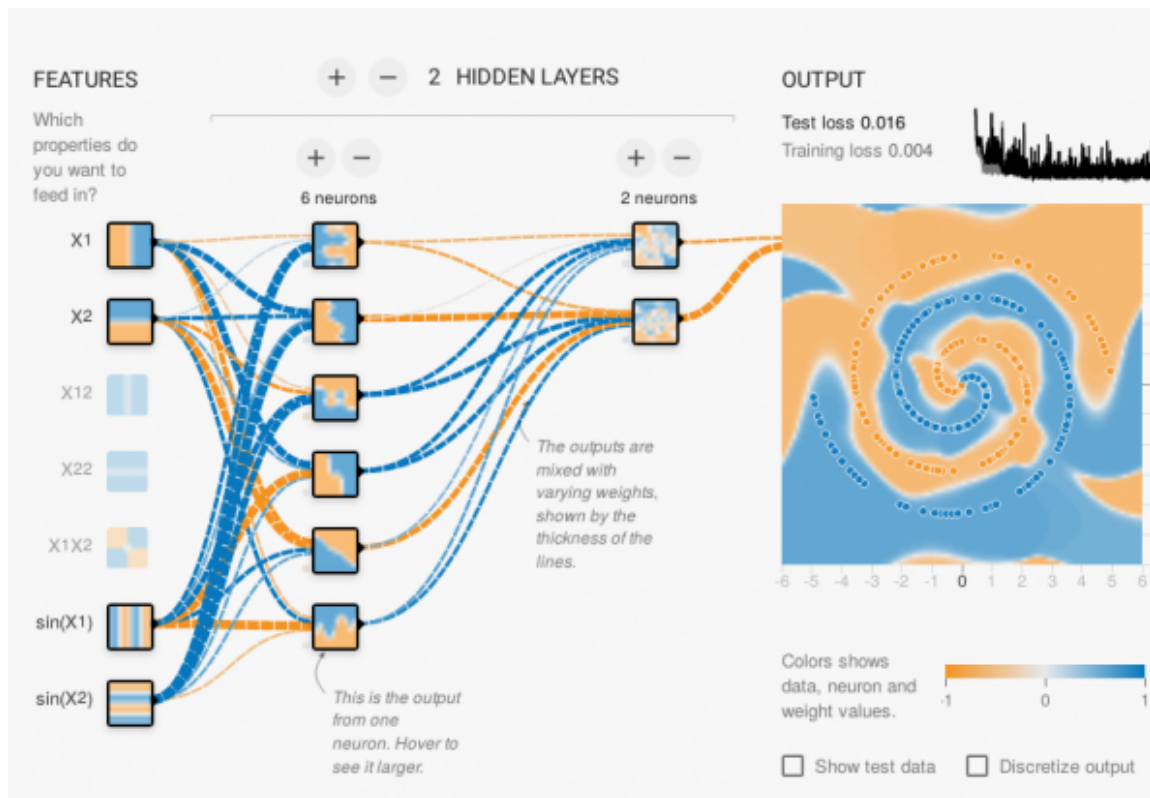


#### Metadata

- Id: EU.AI4T.O1.M3.2.5a
- Title: 3.2.5 Aktivität: Mit den Neuronen der Maschine spielen
- Type: activity
- Description: Verstehen, wie ein künstliches neuronales Netz funktioniert
- Subject: Artificial Intelligence for and by Teachers
- Authors:
  - AI4T
  - Pixees Französische Website
- Licence: CC BY 4.0
- Date: 2022-11-15

## AKTIVITÄT: SPIELEN SIE MIT DEN NEURONEN DER MASCHINE

Die Online-Software **TensorFlow** ermöglicht es, künstliche neuronale Netze zu erstellen und ihre Antworten für verschiedene Problemtypen und Datenarten zu testen. Bei der Problemart "Klassifizierung" besteht das Ziel darin, blaue und orangefarbene Punkte zu trennen. Eine Anwendung dieses Problemtyps ist z. B. ein Algorithmus zur Klassifizierung von Fotos. Im folgenden Beispiel gibt es eine Eingabe (Merkmal), die die Punkte horizontal trennt, und eine andere, die sie vertikal trennt. Kombiniert man diese beiden Eingaben, erhält man eine schräge Trennung. Das Ergebnis (Output) ist gut an die Art der gewählten Daten angepasst.



Tensor Flow Spielplatz Ansicht

## TENSORFLOW: EINIGE ERKLÄRUNGEN, BEVOR SIE DIE SIMULATION EINES NEURONALEN NETZES VERSUCHEN

Quelle: Übersetzung von [Pixees französische Website](#)

**Was ist ein neuronales Netzwerk und wie funktioniert es?** Ein neuronales Netz ist ein allgemeiner Mechanismus, der aus kleinen Einheiten (Pseudoneuronen) besteht, die miteinander verbunden sind. Jede Einheit führt eine sehr einfache Operation durch: Sie nimmt Eingabewerte entgegen, kombiniert sie sehr einfach (eine einfache Mittelwertbildung mit Koeffizienten) und wendet eine Transformation auf das Ergebnis an (z. B. behält sie nur positive Werte).

Die Koeffizienten, die zur Gewichtung des Durchschnitts verwendet werden, sind die Parameter dieses Algorithmus. Erst die Kombination einer sehr, sehr großen Anzahl dieser Einheiten ermöglicht die Durchführung sehr komplexer Operationen. Ein Netz solcher "Neuronen" erhält man durch die Anhäufung mehrerer Schichten solcher Einheiten. Die Eingabe sind die Daten, die wir verarbeiten wollen. Sie werden durch alle Schichten transformiert, und die letzte Schicht gibt als Ausgabe eine Vorhersage über diese Daten ab, zum Beispiel um zu erkennen, ob ein Gesicht in einem Bild zu sehen ist. Das neuronale Netz ist also eine parametrisierte Funktion mit vielen Koeffizienten (den sogenannten "Gewichten"), und die Wahl dieser Gewichte bestimmt die durchgeführte Verarbeitung.



**Wo sind die Neuronen in TensorFlow?** Auf der TensorFlow-Weboberfläche kann ein Netzwerk mit einem Dutzend Neuronen, jedes mit 3 bis 10 Parametern, leicht erstellt werden. Die berechnete Ausgabe hängt also von hunderten von Parametern zusätzlich zu den beiden Koordinaten  $(x,y)$  des Eingabepunktes ab. Auf der Oberfläche stellt jedes Quadrat ein Neuron dar, und die Farbe des Pixels mit den Koordinaten  $(x,y)$  im Quadrat entspricht der Ausgabe des Neurons, wenn wir  $(x,y)$  in den Eingang des Netzes geben. Wenn es nur ein Neuron am Ausgang gibt, wird es durch ein größeres Quadrat auf der rechten Seite des Netzes dargestellt. Die Parameter des Netzes werden mit Zufallswerten initialisiert.

**Aber wie lernt man diese Gewichte?** Das überwachte Lernen besteht darin, Datenbeispiele mit der zu findenden Lösung zu versehen, um das Netz zu trainieren, diese Gewichte nach Bedarf anzupassen. In dem Beispiel in der obigen Abbildung handelt es sich um eine Reihe von Punkten in einem Quadrat mit jeweils einer erwarteten Farbe (blau oder orange), mit dem Ziel, die Farbe des Punktes an einer bestimmten Stelle vorherzusagen. Ein klassischer Algorithmus zur schrittweisen Anpassung der Gewichte wird verwendet, um die fraglichen Parameter zu finden.

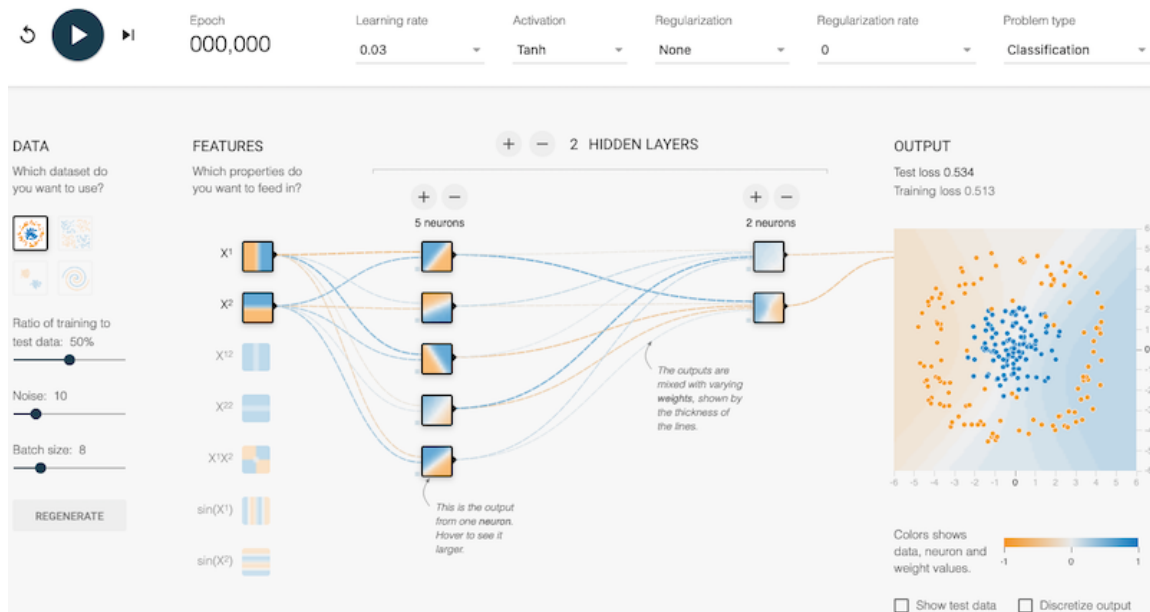
Über die Schaltfläche "Play" oben links auf der Benutzeroberfläche wird dieser Algorithmus gestartet, und die Ausgabe des neuronalen Netzes entwickelt sich im Laufe des "Lernprozesses": Die Hintergrundfarbe des Ausgangsneurons nimmt die Farbe der Trainingspunkte an, die über ihm gezeichnet werden. Ein anderer Teil des Datensatzes wird dann verwendet, um die Qualität der resultierenden Funktion des Netzes zu testen. Eine Kurve oben rechts zeigt die Fehlerrate der zum Lernen verwendeten Daten (um zu überprüfen, ob die Gewichte richtig angepasst wurden) und die Fehlerrate der anderen Testdaten (um zu überprüfen, ob das Gelernte gut auf neue Daten verallgemeinert werden kann). Mit den Schaltflächen auf der linken Seite können Sie die Verteilung der Daten zwischen dem Trainings- und dem Testdatensatz anpassen und auch Fehler zu den Daten hinzufügen (verrauschte Daten), um zu sehen, ob der Mechanismus gegenüber diesen Fehlern robust ist.

In der Praxis gelingt es uns, zufriedenstellende Parameter zu finden, aber es gibt keinen wirklichen theoretischen Rahmen, um all dies zu formalisieren, es ist eine Frage des Experimentierens: die richtige Anzahl von Neuronen, die richtige Anzahl von Neuronenschichten, welche Vorberechnungen als Eingabe hinzugefügt werden sollen (z. B. Multiplikation der Eingaben, um die Freiheitsgrade für die Berechnung zu erhöhen). Solche Techniken können in der Praxis zu beeindruckenden Ergebnissen führen, z. B. bei der Sprach- oder Objekterkennung in einem Bild.

Die Frage, warum (und wie) solch gute Ergebnisse erzielt werden, ist jedoch noch eine ziemlich offene wissenschaftliche Frage.

## TENSORFLOW AUSPROBIEREN

*Klicken Sie auf das Bild unten, um die TensorFlow-Anwendung in einem neuen Fenster zu öffnen.*



TensorFlow playground view