

**1 Présentation**

2 Se lancer

3 Charger les données

4 Conceptualiser la tâche

5 Définir l'architecture du modèle

6 Entraîner le modèle

7 Évaluer le modèle

8 Points à retenir

Signaler une erreur

TensorFlow.js :

Reconnaître des chiffres manuscrits à l'aide de réseaux de neurones convolutifs

À propos de cet atelier de programmation

Dernière mise à jour : avr. 22, 2021

Rédigé par Yannick Assogba

1. Présentation

Dans ce tutoriel, nous allons concevoir un modèle TensorFlow.js capable de reconnaître des chiffres manuscrits à l'aide d'un réseau de neurones convolutif. Nous entraînerons tout d'abord le classificateur en lui présentant des milliers d'images de chiffres manuscrits ainsi que leurs étiquettes. Nous évaluerons ensuite la justesse du classificateur à l'aide de données de test qu'il n'a jamais vues auparavant.

Dans la mesure où nous allons entraîner le modèle pour qu'il attribue une catégorie (le chiffre présent sur une image) à une image d'entrée, cette tâche est considérée comme une tâche de classification. Pour entraîner le modèle, nous lui fournirons de nombreux exemples d'entrées ainsi que leur résultat correct. Ce type de méthodologie est appelé [apprentissage supervisé](#).

Objectifs du tutoriel

Vous allez créer une page Web qui exploite TensorFlow.js pour entraîner un modèle dans le navigateur. Le modèle devra déterminer le chiffre apparaissant dans une image en noir et blanc d'une taille donnée. Cette démarche inclut les étapes suivantes :

- Chargement des données
- Définition de l'architecture du modèle
- Entraînement du modèle et surveillance des performances pendant l'entraînement
- Évaluation du modèle entraîné via des prédictions

Points abordés

- ✓ Syntaxe TensorFlow.js servant à créer des modèles convolutifs à l'aide de l'API Layers TensorFlow.js
- ✓ Création de tâches de classification dans TensorFlow.js
- ✓ Surveillance de l'entraînement dans le navigateur à l'aide de la bibliothèque tfjs-vis

Prérequis

- Une version récente de [Chrome](#) ou un autre navigateur moderne compatible avec les modules ES6.
- Un éditeur de texte, s'exécutant localement sur votre ordinateur ou sur le Web via un environnement tel que [CodePen](#) ou [Glitch](#).
- Une connaissance de HTML, CSS, JavaScript et [Chrome DevTools](#) (ou vos outils de développement de navigateur préférés).
- Une compréhension des concepts généraux des réseaux de neurones. Si vous avez besoin d'une présentation ou d'un rappel, regardez [cette vidéo de 3blue1brown](#) ou [cette vidéo sur le deep learning en JavaScript publiée par Ashi Krishnan](#).

Vous devez également maîtriser les éléments abordés dans [notre premier tutoriel sur l'entraînement](#).

Suivant

Suivant