

# CAHIER DES CHARGES



Interface web - Machine Learning avec TensorFlow.JS



<i><b>Maître d'ouvrage</b></i>	Séverine AFFELDT & Lazhar LABIOD
<i><b>Version du document</b></i>	2.0
<i><b>Date du document</b></i>	22/01/2018
<i><b>Auteur(s)</b></i>	Naomie FOURNIE Lara LAFORGE Malik CHAIBDOUR Alexandre BARBOSA Amine SANAE
<i><b>Soumis le</b></i>	18/06/2019
<i><b>Type de diffusion</b></i>	.PDF
<i><b>Confidentialité</b></i>	Réservé aux étudiants et encadrants UFR Maths-Info de l'université Paris Descartes

# SOMMAIRE

I. Introduction	4
1. Contexte	4
2. Synthèse de l'existant	4
II. Description de la demande	5
1. Les objectifs	5
2. Produit du projet	5
3. Fonctions du produit	5
III. Contraintes	6
1. Contraintes de délais	6
2. Contraintes matérielles	6
3. Autres contraintes	6
IV. Déroulement du projet	7
1. Planification	7
2. Ressources	8
3. Organisation	8
Glossaire	9
Index	10

# I. Introduction

## 1. Contexte

Dans le cadre de notre formation par apprentissage en master MIAGE, nous avons l'opportunité de pouvoir gérer un projet dans son intégralité et ce en situation réelle. Plusieurs choix de sujets nous ont donc été soumis et nous avons choisi le sujet portant sur **le Machine Learning avec TensorFlow.js**.

TensorFlow est un outil open source d'apprentissage automatique développé par Google. Ce réseau permet notamment l'implémentation, l'entraînement et l'évaluation de réseaux de neurones profonds. Ce type de réseau est capable d'apprendre à priori n'importe quelle fonction.

Par exemple, imaginons que sur une photo il y a plusieurs choses dont un chat. Pour pouvoir identifier le chat sur la photo, l'algorithme doit être en mesure de distinguer le chat des autres éléments présents sur la photo, et de reconnaître un chat de manière précise quel que soit l'angle sous lequel il est photographié.

Afin d'y parvenir, le réseau de neurones doit être entraîné. Il est donc nécessaire de compiler un ensemble d'images d'entraînement pour pratiquer le Deep Learning. Cet ensemble va regrouper des milliers de photos de chats différents, mélangés avec des images d'objets qui ne sont pas des chats.

Pour ces images, ce sont les pixels qui seront insérés dans un réseau et sont ensuite convertis en données puis transférés sur le réseau. Les neurones artificiels assignent ensuite un poids (un certain nombre de paramètres ajustables) aux différents éléments.

La couche finale de neurones va alors rassembler les différentes informations pour déduire s'il s'agit ou non d'un chat.

Le réseau de neurones va ensuite comparer cette réponse aux bonnes réponses indiquées par les humains.

Dans le cas contraire, le réseau prend note de son erreur et ajuste le poids placé sur les différents neurones pour corriger son erreur. Le processus est répété sur chaque observation et le jeu de données peut être parcouru plusieurs fois (epoch) jusqu'à ce que le réseau soit capable de reconnaître un chat sur une photo dans toutes les circonstances.

TensorFlow est un système d'apprentissage automatique qui se présente comme une bibliothèque dédiée au calcul numérique.

Initialement, le but de TensorFlow était d'optimiser les calculs numériques complexes, mais aujourd'hui il est très connu pour résoudre des problèmes de Deep Learning. Toutefois, ce framework est assez général pour être utilisé à d'autres fins. Ce document présentera les objectifs que l'on devra atteindre avec le framework TensorFlow.

## 2. Synthèse de l'existant

Aucun existant. Le sujet a été proposé pour la première fois dans l'Université.

# II. Description de la demande

## 1. Les objectifs

L'objectif est de développer une application Web en utilisant la librairie TensorFlow.js permettant d'exécuter une tâche de machine learning pour l'analyse des données et la visualisation de l'information.

Le projet se découpe en 5 objectifs principaux :

- Importation des données ;
- Prétraitement et transformation de données ;
- Analyse exploratoire (statistiques et représentation graphique) ;
- Affichage interactif des résultats ;
- Application de méthodes avancées de Machine Learning.

Dans le cadre de notre apprentissage en Université, les objectifs du projet possèdent aussi des intérêts pédagogiques :

- Découverte des outils TensorFlow, tensorflow.js et le langage python ;
- Familiarisation avec les méthodes basiques de Machine Learning ;
- Définition d'une interface riche avec Javascript, HTML, CSS et tensorflow.js ;
- Application avec classification d'images.

## 2. Produit du projet

Le produit final choisi sera la **reconnaissance d'objets ou faciale avec la webcam**.

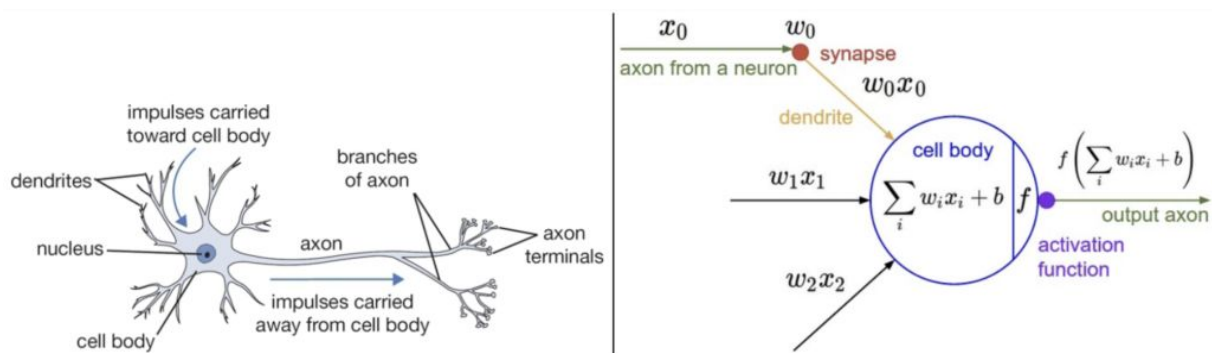
On a un modèle qui s'appelle Mobilenet de type CNN (Convolutional Neural Network). Le but serait d'avoir plusieurs classes, Objet1, Objet2 (le nombre d'objet n'est pas fixe)... On ré-entraîne et modifie dynamiquement le modèle avec les objets que l'on a choisi via la webcam.

Le but est que le modèle entraîné puisse reconnaître l'objet ou le visage sous différents angles, différente luminosité...

## 3. Fonctions du produit

Lister et justifier les principales fonctionnalités du produit.

- Importer des données ;
- Prétraitement et transformation de données ;
- Analyse exploratoire (statistiques et représentation graphique) ;
- Affichage interactive des résultats.



**Figure 1 :** Comparaison d'un neurone au point de vue biologique et mathématiques

- $x_0, x_1, x_2, \dots, x_n$  sont les entrées du neurone. Ce sont soit les données brutes (pour la couche d'entrée), soit les valeurs intermédiaires issues d'une couche cachée : couches comprises entre la couche d'entrée et la couche de sortie.
- $w_0, w_1, w_2, \dots, w_n$  sont les poids de chaque entrée (biais compris)
- $x_0$  est le biais du modèle. C'est une constante permettant de décrire un comportement initial du modèle. Il vaut 1.
- $f$  est la fonction d'activation.

Concernant notre projet, il y aura une partie modification dynamique du modèle et de l'entraînement.

Les paramètres pour l'entraînement seront :

- Le type d'optimizer et sa valeur : paramètres du modèle
- La taille du batch et les époques : paramètres de l'entraînement
- Les images obtenues via la webcam : paramètres de l'entraînement

L'application sera disponible en ligne, pour pouvoir faire participer des gens. L'échantillon utilisé pour nos tests sera à l'échelle de notre classe (pour ceux qui ont souhaité participer aux tests). Concernant le stockage, il y a deux options :

- On stocke les images brut une à une ;
- On stocke le modèle avec les nouvelles données.

Nous avons choisi la première solution.

#### Bilan des fonctionnalités :

- Utilisation de Tensorflow js avec le modèle Mobilenet
- Sauvegarde des images prises par rapport à la webcam
- Entraîner un modèle à partir des images prises directement par la webcam (pas depuis les images sauvegardées pour l'instant)
- Paramétrages possibles du modèle
- Les résultats de l'entraînement sous forme de graphique de suivi
- Tableau des comparaisons entre résultats attendus et résultats prédits

- Prédiction de l'appartenance des visages

### III. Contraintes

#### 1. Contraintes de délais

Le délais imposé par l'Université est de 7 mois. La deadline est planifiée pour la date du : 24/06/2019. De plus, les séances de Projet Pluridisciplinaire ont une durée de 3h30.

#### 2. Contraintes matérielles

Si on veut utiliser juste TensorFlow, il sera nécessaire de posséder un ordinateur assez puissant et la nécessité d'avoir installé Python, les librairies reliées à TensorFlow, PyCharm. Une webcam est nécessaire.

#### 3. Autres contraintes

Nous sommes 5 étudiants, ce qui peut être assez compliqué pour l'organisation. De plus, notre niveau en développement n'est pas homogène au sein du groupe.

### IV. Déroulement du projet

#### 1. Planification

<u>Semaine</u>	<u>Date</u>	<u>Tâche</u>
<u>1</u>	<u>27 novembre</u>	<u>Analyse des besoins</u>
<u>2</u>	<u>4 décembre</u>	<u>Installation des outils</u>
<u>3</u>	<u>11 décembre</u>	<u>Analyse des besoins</u>
<u>4</u>	<u>17 et 18 décembre</u>	<u>Conception</u>
<u>5</u>	<u>7 et 8 janvier</u>	<u>Spécification</u>
<u>6</u>	<u>15 janvier</u>	<u>Développement</u>



<b><u>7</u></b>	<u>21 et 22 janvier</u>	<u>Développement</u>
<b><u>8</u></b>	<u>29 janvier</u>	<u>Développement</u>
	<u>4 et 5 février</u>	<u>Développement</u>
<b><u>9</u></b>	<u>11 et 12 février</u>	<u>Développement</u>
<b><u>10</u></b>	<u>18 et 19 février</u>	<u>Développement</u>
<b><u>11</u></b>	<u>4 et 5 mars</u>	<u>Développement</u>
<b><u>12</u></b>	<u>18 et 19 mars</u>	<u>Développement</u>
<b><u>13</u></b>	<u>25 et 26 mars</u>	<u>Développement</u>
<b><u>14</u></b>	<u>1 et 2 avril</u>	<u>Développement</u>
<b><u>15</u></b>	<u>8 et 9 avril</u>	<u>Développement</u>
<b><u>16</u></b>	<u>15 avril</u>	<u>Développement - Démonstration</u>
<b><u>17</u></b>	<u>29 et 30 avril</u>	<u>Développement</u>
<b><u>18</u></b>	<u>6 et 7 mai</u>	<u>Développement</u>
<b><u>19</u></b>	<u>13 et 14 mai</u>	<u>Développement</u>
<b><u>20</u></b>	<u>20 et 21 mai</u>	<u>Développement</u>
<b><u>21</u></b>	<u>27 et 28 mai</u>	<u>Développement</u>
<b><u>22</u></b>	<u>3 et 4 juin</u>	<u>Développement</u>
<b><u>23</u></b>	<u>11 juin</u>	<u>Rédaction des documents</u>
<b><u>24</u></b>	<u>17 et 18 juin</u>	<u>Rédaction des documents</u>
<b><u>25</u></b>	<u>24 juin</u>	<u>Soutenance</u>

**Figure 2 :** *Planning à respecter (modifié selon les indications de l'encadrant)*



## 2. Ressources

Nous sommes constitués d'un groupe de cinq personnes. Nous disposons chacun d'un ordinateur portable personnel.

Nous serons amenés à utiliser MsProject pour le diagramme de Gantt, Word pour la rédaction des documents.

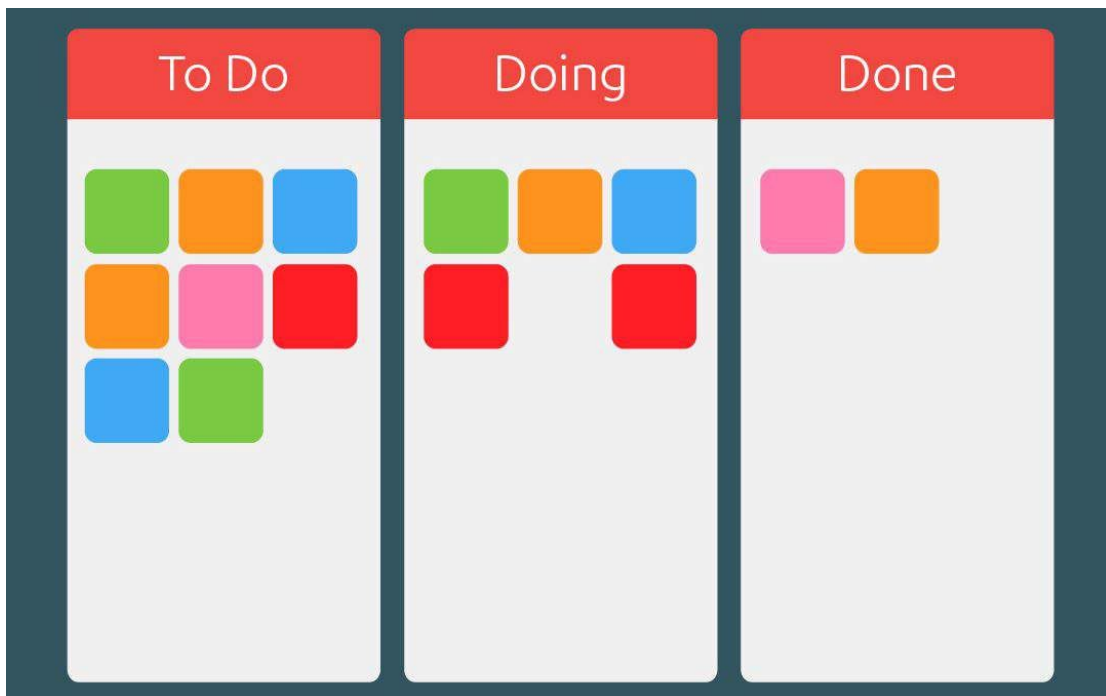
Google Drive permettra le dépôt de document, notamment les comptes rendus, des documents utiles au projet.

## 3. Organisation

La méthode Kanban est une méthode de gestion des connaissances relatives au travail, qui met l'accent sur une organisation de type Juste-à-temps en fournissant l'information ponctuellement aux membres de l'équipe afin de ne pas les surcharger. (cf Figure 1)

Dans cette approche, le processus complet de l'analyse des tâches jusqu'à leur livraison au client est consultable par tous les participants, chacun prenant ses tâches depuis une file d'attente.

C'est un moyen de répartir les tâches de façon agile.



**Figure 3 :** Modèle de Kanban

## Glossaire

**TensorFlow** : TensorFlow est un outil open source d'apprentissage automatique développé par Google.

**Machine Learning** : L'apprentissage automatique (ou machine learning) désigne une famille de méthode qui se base sur des approches statistiques pour donner aux ordinateurs la capacité d'«apprendre» à partir de données.

**MIAGE** : Méthodes Informatiques Appliquées à la Gestion des Entreprises.

**HTML** : HyperText Markup Language.

**CNN** : Conventioneel Neutral Network est un algorithme de Deep Learning.

**CSS** : Cascading Style Sheets.

**Javascript** : JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.js.

**Gitlab** : GitLab est un gestionnaire Web de référentiels Git fournissant des fonctionnalités de wiki, de suivi des problèmes et de pipeline CI / CD, utilisant une licence open source, développée par GitLab Inc.

**MsProject** : Microsoft Project est un logiciel de gestion de projets édité par Microsoft. Il permet aux chefs de projet et aux planificateurs de planifier et piloter les projets, de gérer les ressources et le budget, ainsi que d'analyser et communiquer les données des projets.

## Index