BUT
Métiers du Multimédia et de l'Internet
Troisième année

Projet (9h de TP en autonomie)
Web & IA – R602

1 Principe du projet

Vous avez deux exercices à réaliser pour ce projet. L'un est un problème de classification que vous devrez réaliser en python. Le second consiste à réaliser une application de démonstration pour la fête de la science. Le projet se déroule sur 3 séances de TP en autonomie pour un total de 9h, mais rien ne vous empêche bien sûr de travailler en dehors de ces heures affectées à l'emploi du temps.

2 Exercice de classification

Le taux de clics est le rapport entre le nombre d'utilisateurs qui ont cliqué sur une annonce (un ad) et le nombre total d'utilisateurs qui ont laissé des impressions sur l'annonce. En d'autres termes, clics ÷ impressions = CTR. L'analyse du taux de clics aide les entreprises à déterminer les types de personnes les plus susceptibles de cliquer sur leurs annonces. Un CTR élevé valide une stratégie publicitaire.

Vous disposez d'un dataset (issu de https://statso.io/click-through-rate-analysis-case-study/) nommé «ad_10000records.csv » qui contient pour chaque ligne du fichier, les informations suivantes :

1. Daily Time Spent on Site : la durée quotidienne de l'utilisation du site web par l'utilisateur,
2. Age : L'âge de l'utilisateur,
3. Area Income : le revenu moyen dans la région de l'utilisateur,
4. Daily Internet Usage : l'utilisation quotidienne d'internet par l'utilisateur,
5. Ad Topic Line : Le titre de l'ad,
6. City : la ville de l'utilisateur,
7. Gender : le genre de l'utilisateur,
8. Country : le pays de l'utilisateur,
9. Timestamp : la date de la visite du site par l'utilisateur,
10. Clicked on Ad : 1 si l'utilisateur a clique sur l'ad, 0 sinon.

Votre objectif est de concevoir un modèle de classification à base de réseau neuronal dense qui permettre de prédire si un utilisateur a cliqué sur un ad (variable Clicked on Ad) uniquement à partir des variables suivantes : Daily Time Spent on Site, Age, Area Income, Daily Internet Usage, et genre. Le fichier étant conséquent (environ 10000 instances), vous en construirez une version minimale nommée « ad_mini.csv » à l'aide d'un script python. Vous ne retiendrez que les colonnes nécessaires et ne garderez que 1000 instances pour chaque classe (Clicked on Ad à 1 ou 0) par un tirage aléatoire.

Concevez ensuite un script python qui entraine un réseau neuronal dense contenant plusieurs couches et permettant de prédire un click sur un ad. Écrivez un formulaire

permettant d'interroger le modèle entrainé afin de faire la prédiction de click à partir de valeurs fournies dans le formulaire. Utilisez flask.

3 Démonstrateur pour la fête de la science

On vous missionne pour réaliser une application de démonstration de l'usage de l'intelligence artificielle pour la fête de la science.

3.1 Étape 1

Vous devez concevoir une application (JS – mais pas de Node.js) qui permet :

- De capturer un flux vidéo de la webcam
- De prédire le squelette de la main lorsqu'elle apparait à l'écran
- Pour la main détectée :
    - D'afficher le squelette superposé sur le flux vidéo
    - De prédire le geste effectué par la main parmi un ensemble de gestes prédéfinis
    - D'afficher les gestes prédits

La détection des points clés de la main peut se faire avec les modèles pré-entrainés de MediaPipe :

https://github.com/tensorflow/tfjs-models/tree/master/hand-pose-detection

dont la documentation est disponible ici :

https://github.com/tensorflow/tfjs-models/tree/master/hand-pose-detection/src/tfjs

Des démonstrations sont disponibles ici :

https://storage.googleapis.com/tfjs-models/demos/hand-pose-detection/index.html?model=mediapipe_hands

https://codepen.io/mediapipe-preview/pen/zYamdVd

Il faut maintenant rendre ceci fonctionnel et attrayant dans l'application de démonstration qu'il vous est demandé de réaliser !

3.2 Étape 2

Les gestes prédits sont pour l'instant ceux prédéfinis dans MediaPipe. Il est possible d'en reconnaitre d'autres. Soit on définit son propre modèle personnalisé que l'on entraine :

https://ai.google.dev/edge/mediapipe/solutions/customization/gesture_recognizer

Cependant, cela nécessite de ré-entrainer un modèle. Alternativement on peut utiliser le package JS fingerpose qui permet de définir la structure des gestes que l'on souhaite reconnaître :

https://github.com/ErickWendel/fingerpose/tree/master?tab=readme-ov-file

A l'aide de *fingerpose*, faites en sorte de reconnaitre lorsque la main pointe vers les directions Haut/Bas/Gauche/Droite. Vous devrez créer une description pour chaque geste à reconnaître.

3.3 Étape 3

Exploiter les gestes de la main pour interagir dans un jeu peut se faire dans des jeux complexes :

https://github.com/chaitanya-chafale/Hand-Gesture-Gaming

Avec une démo fonctionnelle ici :

https://chaitanya-chafale.github.io/Hand-Gesture-Gaming/

On ne vous demande pas un jeu si complexe, mais le jeu du snake. Le principe du jeu est le suivant : On contrôle un serpent avec les gestes de la main :

- 👆 Haut → le serpent monte
- 👇 Bas → descend
- 👉 Droite → va à droite
- 👈 Gauche → va à gauche

Le serpent doit manger des pommes sans toucher les bords ou se toucher. Quand il mange une pomme, il grandit. Une démo fonctionne est visible ici : https://www.snake.fr/jeux/google-snake

Vous pensez que la reconnaissance des gestes pourrait être utile pour un autre jeu simple que le Snake ? Vous pouvez tout à fait le faire pour un autre jeu qui s'y prête, c'est comme vous voulez !