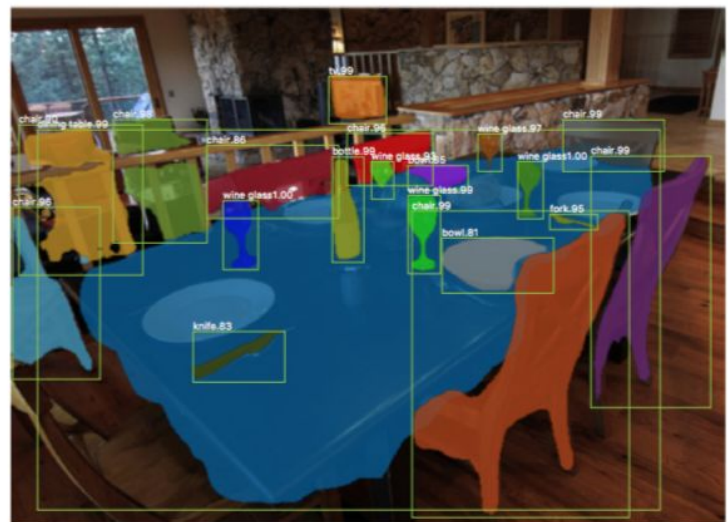
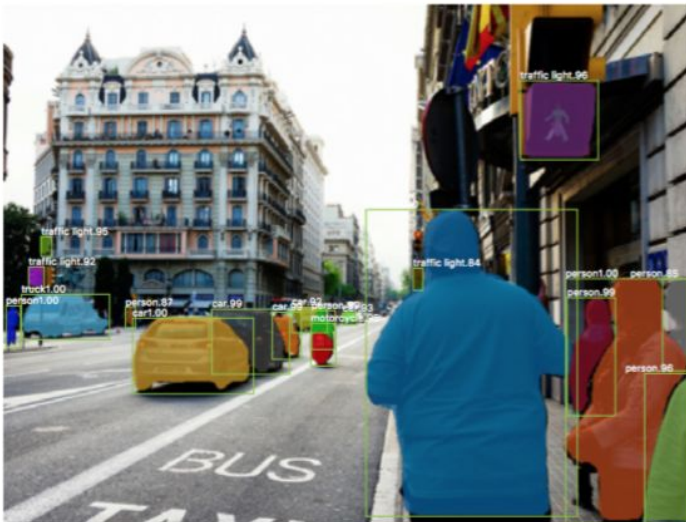


Systèmes Intelligents - IA TP

Afficheur 7 segments



Un exemple de réseau de neurones: Mask-RCNN. Il est constitué de plus de 90 millions de unités avec leurs poids, et est un assemblage de plusieurs MLP et CNN. Il est capable de détecter des objets et de tracer leurs contours au sein d'une image. Il nécessite environ 10Go de RAM GPU pour un fonctionnement par batch de 1 image. Toutefois, même si c'est cool, ça n'a rien à voir avec le TP.



14H rue Pierre de Coubertin
21000 Dijon
FRANCE

Tel : +33 380 371 795

23, rue Paul Bert
92100 Boulogne Billancourt
FRANCE

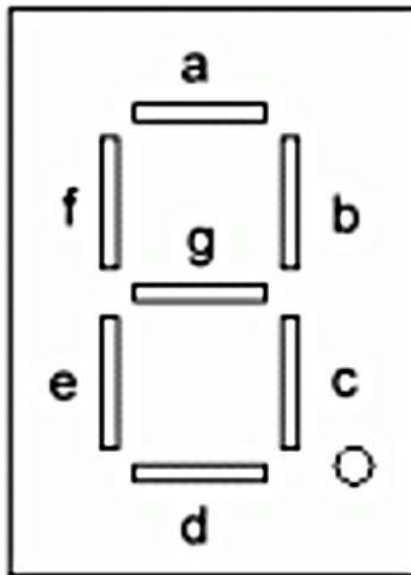
Tel : +33 146 032 660

Vous avez jusque là vu comment lire un fichier CSV, caster dans différents types en Python, écrire des classes, des fonctions et des méthodes, et avez presque terminé de développer votre premier perceptron assez générique pour résoudre différents problèmes assez simples.

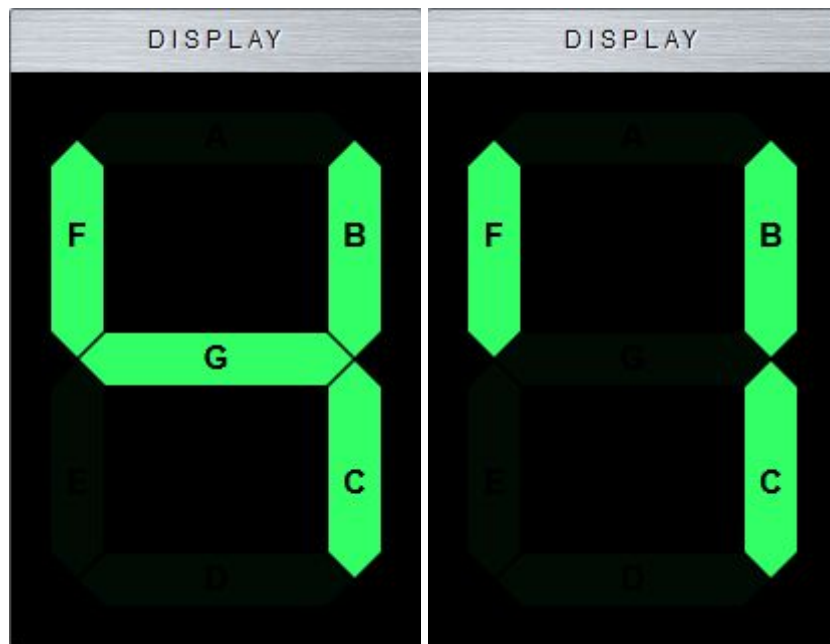
Afin de terminer votre classe perceptron, vous allez devoir y ajouter quelques fonctionnalités:

1. Tout d'abord, une méthode `save_weights()` qui sauvegarde les poids du perceptron dans un fichier CSV
2. Ensuite, une méthode `load_weights()` qui prend en paramètre le chemin vers un fichier CSV de poids et qui charge lesdits poids dans le perceptron.
3. A chaque fin d'époque, calculez la valeur d'erreur totale sur l'époque. Retenez cette valeur dans une liste, que vous afficherez au sein d'un graphe avec matplotlib.
4. Calculez la convergence et stoppez l'apprentissage dès que celle-ci est atteinte. Autrement dit, regardez la valeur d'erreur totale epoch après epoch. Si la valeur d'erreur ne change pas suffisamment (à vous de décider ce que "suffisamment" signifie), alors cela signifie que vous avez atteint un point de convergence.

Vous allez désormais travailler sur le problème de l'afficheur 7 segments. Je vous conseille FORTEMENT de lire tout l'énoncé avant d'entreprendre quoi que ce soit.



Vous avez à votre disposition, dans votre imaginaire tout du moins, un afficheur 7 segment, dont chaque segment peut être alimenté afin de s'allumer. Nous souhaitons faire en sorte que, lorsque certains segments sont allumés, une IA soit capable de déterminer le chiffre qui est censé s'afficher. La raison pour laquelle cela *pourrait* être intéressant est que, comme les réseaux de neurones sont capables de généraliser ce qu'ils apprennent, un segment qui viendrait à ne pas s'allumer ne nous poserait pas de problème pour détecter le chiffre que l'on souhaitait afficher. Aux "yeux" de l'IA, les deux affichages ci-dessous sont des 4.



Première étape: sans prendre en compte le 0, cherchez un moyen de représenter votre modèle de données. Comment les inputs et outputs sont-ils formés ? Combien d'entrées possèdera le perceptron ?

Deuxième étape: Manuellement, stocker vos données d'inputs dans un fichier CSV, et vos données d'output dans un autre fichier CSV. Ne créez aucune entrée pour la valeur 0.

Troisième étape: Pour ce problème, une sortie vous donnant 0 ou 1 en suivant la règle du perceptron ne sera pas suffisante. Il faut mettre en place une nouvelle fonction d'activation: la fonction ReLU. Cette-fois ci, au lieu d'avoir $y = 0$ si $y' \leq 0$ et $y = 1$ si $y' > 0$ nous allons avoir

$$y = 0 \text{ si } y' \leq 0$$

$$y = y' \text{ si } y' > 0$$

Faites en sorte que nous puissions choisir, via un paramètre du constructeur du perceptron, si l'on souhaite utiliser la règle du perceptron ou ReLU. **La fonction d'activation ne peut pas changer une fois qu'elle a été choisie.**



Quatrième étape: Récupérez vos inputs et outputs correspondant via le code en ouvrant les fichiers CSV que vous avez créé manuellement. Jouez avec les paramètres de votre perceptron jusqu'à obtenir au moins 6 prédictions correctes. Normalement, vous devriez vous retrouver avec toujours au moins 2 chiffres qui ne peuvent pas être détectés convenablement. Expliquez la raison.

Cinquième étape: Remplissez bien votre compte-rendu de TD/TP. Vous devrez en 5 minutes, à une date ultérieure, faire une présentation à l'oral sur comment vous avez mis en place vos données, pourquoi un tel choix de paramètres, que faire pour améliorer ce dernier réseau de neurones, etc. Soyez concis. Préparez-vous également à recevoir des questions sur le cours et sur les questions qui ont été posées en cours.

Pour rappel, le rapport est à rendre pour le 22 décembre au soir, à 23h59 au maximum. Il y aura -4 points par jour de retard. L'examen est toujours fixé au 15 janvier, et la date de la présentation orale reste à définir.

