Simulation de voiture autonome

Détails et Code

Les outils

Dans ce TP, je vous **conseille** d'utiliser la VM Développement Windows.

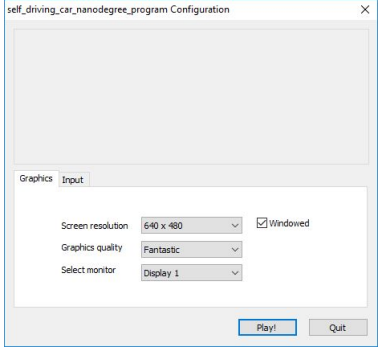
Nous aurons besoin de beaucoup de ressources (données, bibliothèques) déjà présentes dans la VM.

Dans la VM, vous avez tout pour travailler mais si vous le souhaitez vous pouvez télécharger les éléments un par un MAIS cela peut être très long.

Organisation du projet

- Le simulateur 3D
- modellinear.py
  - Regression linéaire, calculer les W
- drivelinear.py
  - Prédiction de l’angle du volant lors du mode autonome
- Aller dans c:\optioninfopeip2\voitureaunome
  - Nous y mettrons nos codes, nos données ….
-
- N’hésitez pas à revenir au snapshot original de la VM pour ne pas être pollué par le code des autres.

Le Simulateur 3D (dispo dans la VM)



- Le simulateur est disponible via ce lien :

  https://drive.google.com/open?id=1ZmK2b3A3efpPHVhV-wq12Jzs_RIHdueb

- Pour information il existe sous Windows, Mac et Linux
  - “https://d17h27t6h515a5.cloudfront.net/topher/2017/February/58ae46bb_linux-sim/linux-sim.zip">Linux
  - "https://d17h27t6h515a5.cloudfront.net/topher/2017/February/58ae4594_mac-sim.app/mac-sim.app.zip">macOS
  - “href="https://d17h27t6h515a5.cloudfront.net/topher/2017/February/58ae4419_windows-sim/windows-sim.zip">Windows
- Lancer le simulateur pour voir si il fonctionne
  - Play

Les données : (dispo dans la VM)

- Pour trouver les paramètres W il faut des images et des angles:
- Télécharger les données (320 Mo) :
  - https://drive.google.com/open?id=1crklqGi-36il2QLCjAgF9XVYGaJ1ZnHC
- Ouvrez le fichier driving_log.csv avec Notepad
  - Regarder les 2 premières lignes :
  - Déduire le caractère de séparation de colonne
  - Trouver les valeurs des angles ainsi que leur unit
- Regarder le contenu du répertoire IMG

Spyder en mode tfopencv2

- Démarrer Anaconda Navigator
- Dans le menu “applications on” passer de “base” à “tfopencv2”
- Lancer Spyder

Si vous n’avez pas la VM (solution 1)

Les principales bibliothèques dont vous avez besoin :

1. `numpy`
2. `Image`
3. `opencv-python`
4. `sklearn`
5. `jupyter`
6. `scipy`
7. `flask-socketio`

Vous pouvez les installer en lançant anaconda prompt

Et en tapant pip install “le nom de la bibliothèque”

Si vous n’avez pas la VM (solution 2)

Télécharger le fichier requirements.txt

https://drive.google.com/file/d/1mm-dth98sMx_uyR4sGPo8x1pZxq-X37_/view?usp=sharing

Lancer “anaconda prompt” dans et tapez :

pip install requirements.txt

Si vous n’avez pas la VM (solution 3)

Uniquement sous windows 10 :

Télécharger l’archive suivante : https://drive.google.com/file/d/1tsY2HnVBm1QRAbQDWW45YvhL3wJ2s48-/view?usp=sharing

Décompresser l’archive

Aller dans le répertoire obtenu lors de la décompression “tfopencv2” :

Aller dans le répertoire tfopencv2\Script et cliquer sur idle.exe

Idle.exe est un editeur simple de code python. Ouvrez vos fichiers et codez.

Le code

- Télécharger la base de code
  - Modelinear.py
    - https://drive.google.com/open?id=1MajRN6Ips42wvLAOBZvvK0S_wi_rybE_
  - Drivelinear.py
    - https://drive.google.com/open?id=1zq8_dKXZhhWeZ71BuDFG0ir58enNwEkQ
- Mettre les fichiers dans le **répertoire “code”**
-

Modelinear.py

- C’est fourni :
  - La fonction getBatch
    - Retourne un lot de 64 images avec les angles associés
    - Appel de la fonction :
      - images, rotations = getBatch(content, train_index, BATCH_SIZE,True)
      - Warning : Comme dans le notebook, les vecteurs images contiennent déjà une entrée pour le biais
- A faire :
  - Modifier la fonction main
  - Modifier la fonction gradient
  - Le calcul de W par la méthode de la descente de gradient
    - Voir pseudo code de la slide suivante

Descente de Gradient (mini batch)

1. Récupérer un lot de données par la méthode getBatch
2. Appeler la méthode DescenteGradient
   a. Calculer le gradient de W par rapport à la sortie
   b. Mettre à jours les W pour un pas de gradient
3. Aller à l'étape 1 tant qu'on a pas traité toutes les données de train_index
4. Recommencer tant que le nombre d'itération n'est pas atteint

Fourni : A la fin sauvegarder W dans un fichier texte (modelW.txt)

Pseudo code fonction main : Descente de Gradient (mini batch)

```
tailleTrain  = nombre d'images total=len(train_index)
Nbiter = nombre d'itération de la descente de gradient (epoch)
pas= pas de descente de gradient
BATCH_SIZE = 64 dans notre exemple

pour j de 0 à nbiter:
      pour i de 0 à tailleTrain  par pas de BATCH_SIZE :
          images, rotations = getBatch(content, train_index,
BATCH_SIZE,True)

          W,error =DescenteGradient(images,rotations,W,pas)
```

Function DescenteGradient

1. Calcul du gradient : GradW=X^T (XW-Y) (je crois à vérifier)
2. Mise à jour des paramètres W --> W=W-alpha* GradW
3. Calcul et affichage de l’erreur

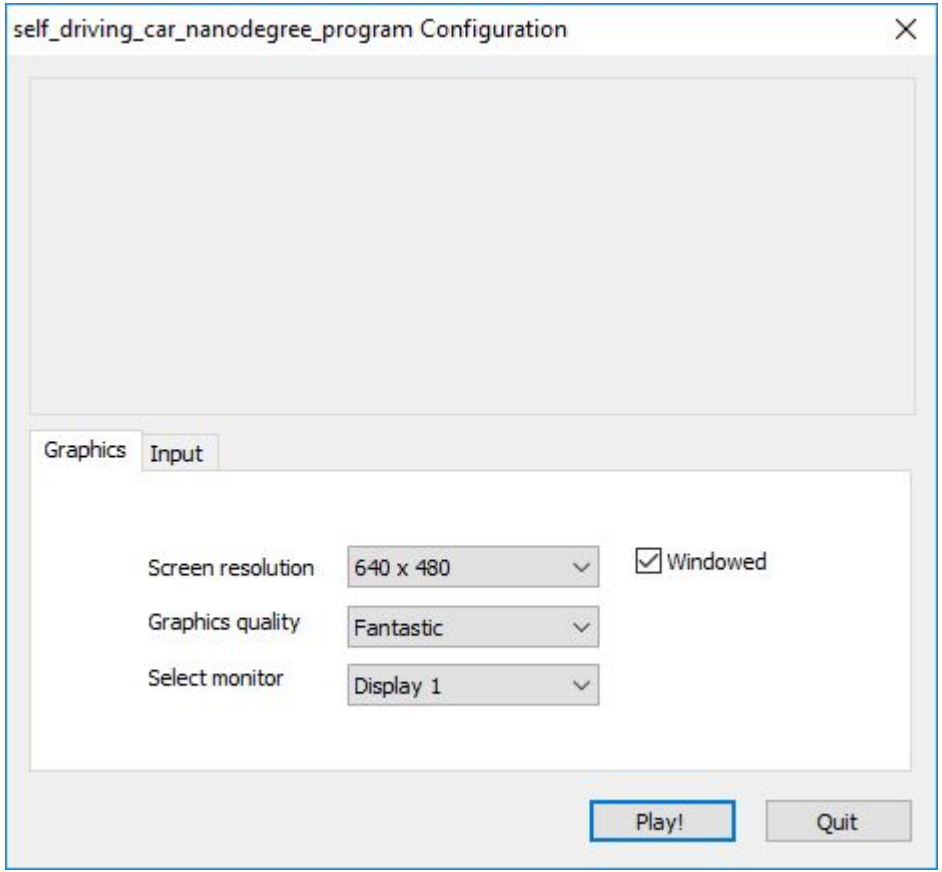
drivelinear.py

1. On conduit !!!!
2. Dans le main :
    a. Chargement des paramètres W
        i. W=np.loadtxt("modelW.txt")
3. Dans la fonction telemetry :
    a. On récupére l'image du simulateur
        i. Variable Image_array puis x
4. Dans la fonction telemetry : à faire
    a. Prédire la valeur de l'angle
    b. Stocker la valeur dans la variable angle
5. Exécuter le fichier et attendre …. La diapo suivante

Conduire tout seul

Lancer le simulateur

Avec les paramètres suivants :



Conduire tout seul

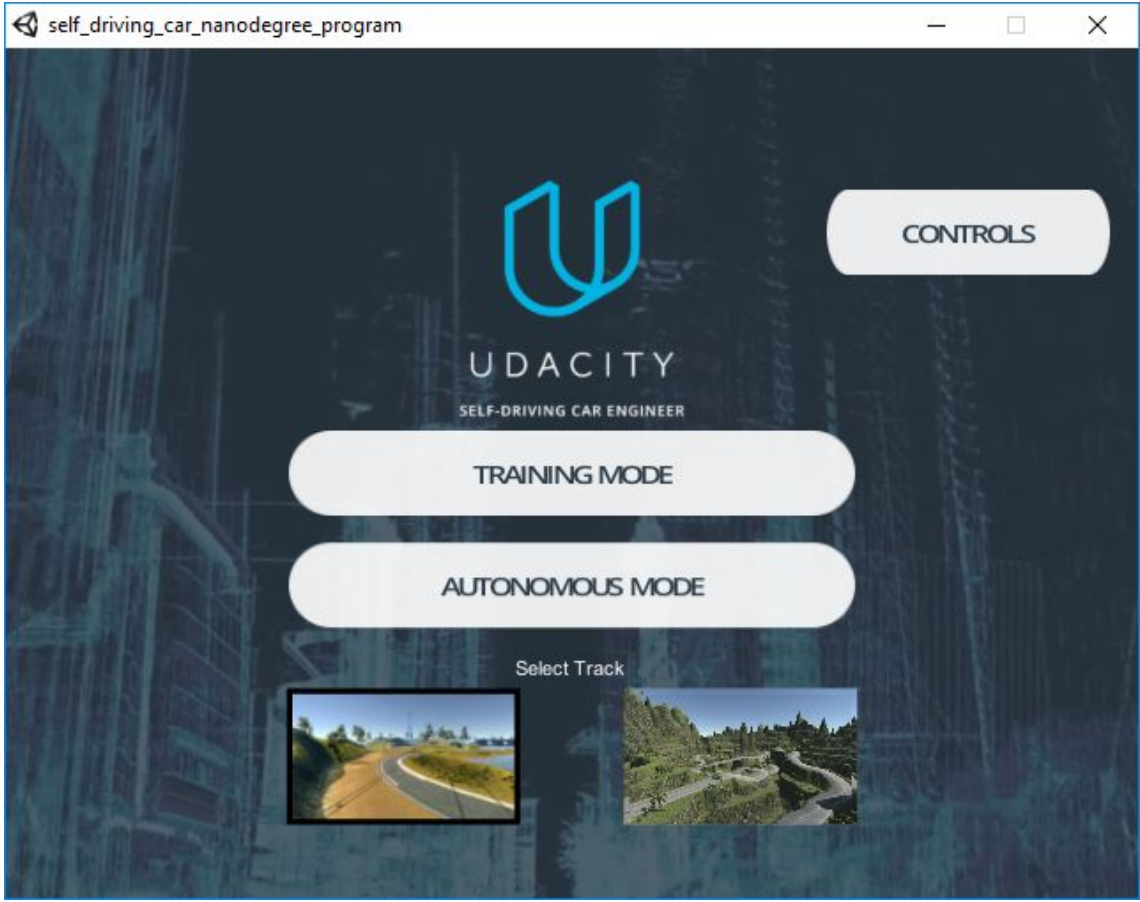
Choisir la course de gauche

Lancer le mode autonome

C’est parti !!! ça roule

Défi Fast and Furious :

- Essayer de changer la
- Vitesse de la voiture
- (20 MPH)



Bonus

Modelinear.py : Calculer l’erreur sur la base de validation

Modelinear.py : Afficher la courbe d’erreur en fonction des itérations sur la base d’apprentissage

Modelinear.py : Afficher la courbe d’erreur en fonction des itérations sur la base de validation

Modelinear.py : Sauvegarder W pour la plus faible erreur sur la base de validation. Tester en mode conduite autonome

Modelinear.py : Calculer et afficher le coefficient de détermination en plus de l’erreur