

Enseignant(s)

VIDAL Nicolas

Email(s)

nvidal@myges.fr

2020-4A-IABD - Kaggle Competition and then History of Reinforcement

1 Matières, formations et groupes

Matière liée au projet :

Formations : -

Nombre d'étudiant
par groupe :

3 à 4

Règles de constitution des groupes: **Imposé**

Charge de travail
estimée par étudiant :

30,00 h

2 Sujet(s) du projet

Type de sujet : **Imposé**

3 Détails du projet

Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)

Implémenter et comprendre quand utiliser les algorithmes classiques de l'apprentissage par renforcement, qu'ils soient issus de la programmation dynamique ou de techniques à base de simulations Monte Carlo, comprendre l'impact de l'utilisation de techniques d'approximation de fonctions dans ce contexte.

Descriptif détaillé

Partie 1 :

Exceptionnellement cette année, la première partie de ce projet est une mise en application des algorithmes vus au premier semestre lors de la matière Deep Learning.

Il est demandé aux étudiants de participer à au moins une des compétitions Kaggle suivantes :

<https://www.kaggle.com/c/plant-pathology-2020-fgvc7/overview>

<https://www.kaggle.com/c/iwildcam-2020-fgvc7/overview>

<https://www.kaggle.com/c/herbarium-2020-fgvc7/overview>

<https://www.kaggle.com/c/imaterialist-fashion-2020-fgvc7/overview>

Après avoir formé les équipes correspondant aux groupes de ce projet et d'avoir respecté scrupuleusement les règles des compétitions, les étudiants seront amenés à entraîner des modèles selon les méthodologies vues au premier semestre :

Les étudiants devront procéder à l'étude de différents modèles sur des Datasets en faisant varier les différents hyper paramètres de ces derniers :

* Modèles précédents

- Modèle Linéaire
- Perceptron Multicouches

* Nouveaux modèles

- Conv Net(s)
- ResNets / HighwayNets
- RNN(s)

Il sera fortement conseillé aux étudiants de s'appuyer sur Tensorboard pour visualiser comparer et retranscrire les performances de ces modèles.

Il sera important de faire apparaître pour chaque compétition, pour chaque modèle :

L'influence de tous les hyperparamètres des modèles (structure, fonctions d'activation, ...), ainsi que des paramètres des algorithmes d'apprentissages (learning rate, momentum, ...)

Après avoir entraîné ces modèles, les étudiants devront soumettre sur les sites des compétitions leur meilleurs résultats pour figurer dans le classement public. Les étudiants sont fortement encouragés à enregistrer (screenshots) l'évolution de leur classement lors de ces compétitions.

Un post mortem de ces participations sera attendu lors du premier rendu intermédiaire correspondant à une petite soutenance (5 minutes par groupe) ainsi qu'un rapport interactif faisant figurer l'ensemble des résultats obtenus par les étudiants.

Partie 2 :

Dans ce projet, les étudiants devront en premier lieu implémenter les algorithmes suivants et s'assurer de leur bonne implémentation sur des cas de tests bien précis :

- Dynamic Programming

- Création des contrats liés à un MDP
- Création d'un environnement simple : Line World
- Implémentation de l'algorithme "Policy Evaluation"
- Implémentation de l'algorithme "Policy Iteration"
- Implémentation de l'algorithme "Value Iteration"
- Tests et vérifications de ces algorithmes sur l'environnement Line World
- Création d'un environnement un peu plus complexe : Grid World
- Tests et vérifications de ces algorithmes sur l'environnement Grid World

- Méthodes Monte Carlo

- Implémentation d'un TicTacToe 1 (joueur VS Random)

- Implémentation de l'algorithme "Monte Carlo ES"
- Implémentation de l'algorithme "On-policy first visit Monte Carlo Control"
- Implémentation de l'algorithme "Off-policy Monte Carlo Control"
- Tests et vérifications de ces algorithmes sur l'environnement TicTacToe 1

- Temporal Difference Learning
 - Implémentation de l'algorithme Sarsa
 - Implémentation de l'algorithme Q-Learning
 - Implémentation de l'algorithme Expected Sarsa
 - Tests et vérifications de ces algorithmes sur l'environnement Line World
 - Tests et vérifications de ces algorithmes sur l'environnement Grid World
 - Tests et vérifications de ces algorithmes sur l'environnement TicTacToe 1
 - Optionnel : Implémentation de l'algorithme "n-step Q-Learning"

- Planning
 - Implémentation de l'algorithme "Dyna-Q"
 - Optionnel : Implémentation de l'algorithme "Dyna-Q+"

- Deep Reinforcement Learning
 - Création d'un environnement beaucoup plus complexe : PacMan
 - Implémentation de l'algorithme "Episodic Semi-gradient Sarsa"
 - Implémentation de l'algorithme "Deep Q Learning"
 - Tests et vérifications de ces algorithmes sur l'environnement Line World
 - Tests et vérifications de ces algorithmes sur l'environnement Grid World
 - Tests et vérifications de ces algorithmes sur l'environnement TicTacToe 1
 - Tests et vérifications de ces algorithmes sur l'environnement PacMan
 - Optionnel : Implémentation de l'algorithme "Double Deep Q Learning"

- Policy gradient Methods
 - Implémentation de l'algorithme "REINFORCE"
 - Implémentation de l'algorithme "REINFORCE with Baseline"
 - Tests et vérifications de ces algorithmes sur l'environnement Line World
 - Tests et vérifications de ces algorithmes sur l'environnement Grid World
 - Tests et vérifications de ces algorithmes sur l'environnement TicTacToe 1
 - Tests et vérifications de ces algorithmes sur l'environnement PacMan

Il sera également nécessaire de présenter une interface graphique permettant de regarder jouer chaque agent et également de mettre à disposition un agent 'humain'.

Pour chaque environnement, les étudiants devront étudier les performances de l'agent ou de chaque couple d'agents possible et retranscrire leur résultats.

Les étudiants devront fournir l'intégralité du code leur ayant permis d'obtenir leurs résultats ainsi que les modèles (keras) entraînés et sauvegardés prêts à être exécutés pour confirmer les résultats présentés.

Les étudiants devront présenter ces résultats lors d'une soutenance. Dans cette dernière, les étudiants devront faire valoir leur méthodologie de choix d'hyperparamètres, et proposer leur interprétation des résultats obtenus

Ouvrages de référence (livres, articles, revues, sites web...)

Reinforcement Learning: An Introduction :
<http://incompleteideas.net/book/the-book.html>
 Richard S. Sutton
 and Andrew G. Barto

Outils informatiques à installer

Unity / Unreal Engine / PyGame / ... ?
 Tensorflow / Keras

1	Etape intermédiaire	<p>Choix d'un ou plusieurs sujets de compétition parmi : https://www.kaggle.com/c/plant-pathology-2020-fgvc7/overview https://www.kaggle.com/c/iwildcam-2020-fgvc7/overview https://www.kaggle.com/c/herbarium-2020-fgvc7/overview https://www.kaggle.com/c/imaterialist-fashion-2020-fgvc7/overview</p> <p>Les étudiants doivent uploader un fichier texte dans lequel doivent figurer :</p> <ul style="list-style-type: none"> - leurs noms et prénoms tels qu'ils figurent sur Kaggle - le nom de leur équipe - le/les compétitions choisies 	dimanche 29/03/2020 23h59
2	Etape intermédiaire	<p>Présentation des résultats et du protocole suivi pour les compétitions Kaggle</p> <p>Livrables :</p> <ul style="list-style-type: none"> - rapport sous la forme de Notebook Jupyter + pdf - screenshots des résultats obtenus lors de la compétition (classement) - courte présentation (slides) de 5 minutes 	jeudi 28/05/2020 23h59
3	Rendu final	<p>Présentation des résultats et du protocole suivi pour les environnements de Deep Reinforcement Learning</p> <p>Livrables :</p> <ul style="list-style-type: none"> - rapport sous la forme de Notebook Jupyter + pdf - présentation (slides) de 10 minutes - sources complètes du projet - démonstration <p>(attention la date n'est pas complètement figée et n'est fixée qu'à titre indicatif)</p>	vendredi 17/07/2020 9h00

5 Soutenance

Durée de présentation par groupe :	15 min	Audience : A huis clos
Type de présentation :	Présentation / PowerPoint	
Précisions :		