

Dans le cadre de l'évaluation des performances des systèmes de décision embarqués dans les véhicules autonomes, il semble pertinent de s'intéresser à un jeu de données de perception qui prend en compte un système de prise de décision. A ce titre, nous allons travailler sur un jeu de données robotiques intégrant un système de prise de décision.

1 Description du Robot SCITOS-G5 et dataset

Les données robotiques collectées sont recueillies à partir du robot SCITOS-G5. Ce dernier a été construit par MetraLabs Robotics Company basée en Allemagne et a par ailleurs été conçu de sorte à avoir une base circulaire sur laquelle repose 24 capteurs à ultrasons répartis en cercle autour de sa ceinture abdominale dont la précision est de l'ordre de 2 à 3 cm (<https://www.robotshop.com/media/files/PDF/datasheet-scitos-g5-advanced.pdf>). Cette dernière permet donc au robot de pouvoir se mouvoir dans toutes les directions grâce à un système de moteur différentiel. Un ordinateur de bord ainsi que d'autres capteurs sont mis en place sur SCITOS-G5, comme une caméra sur la partie supérieure du robot ainsi qu'un capteur laser.

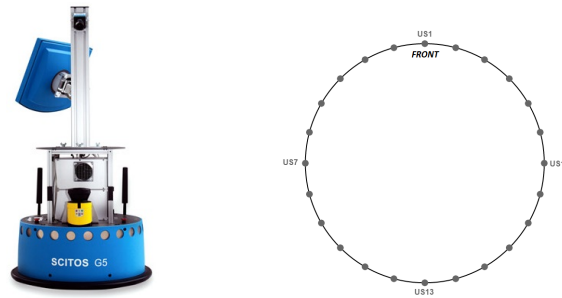


FIGURE 1 – Robot SCITOS-G5 et ses capteurs

La ceinture des 24 capteurs à ultrasons va permettre de récolter les données de perception avec un relevé des données toutes les 9 secondes, ce qui nous permettra d'avoir à notre disposition d'avoir un jeu de données composées de 5456 échantillons [1]. Avant tout, il faut prendre connaissance de la manière dont les données sont récoltées.

En effet, il faut savoir que le robot SCITOS-G5 a été conçu pour se déplacer en intérieur et plus précisément, ici il se déplace dans une pièce rectangulaire dans laquelle divers objets ont été placés. On peut observer la disposition de la pièce sur la figure 1 qui est extraite de [2]. Le robot sera alors chargé de faire quatre fois le tour de la pièce dans le sens des aiguilles d'une montre.

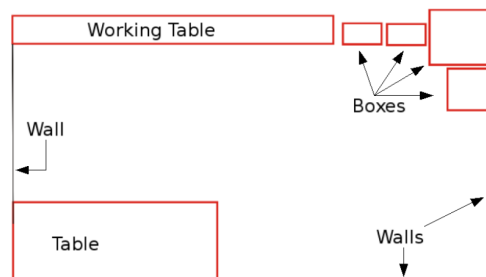


FIGURE 2 – Pièce dans laquelle SCITOS-G5 circule

Les données que l'on récolte sont extraites d'un système de prise de décision, il faut donc décrire la manière dont les décisions que le robot prend au fur et à mesure de sa navigation sont prises. A chaque étape, les 24 capteurs à ultrasons vont relever les distances respectives aux objets les plus proches, et à partir de ces dernières une nouvelle orientation pour le robot sera choisie. En effet, les directions sont choisies à partir d'un algorithme simple : on regroupe les distances des capteurs en 4 groupes (gauche, droite, devant, derrière) de sorte à prendre assez de capteurs pour couvrir à chaque fois un angle de 60° , comme sur la figure 3.

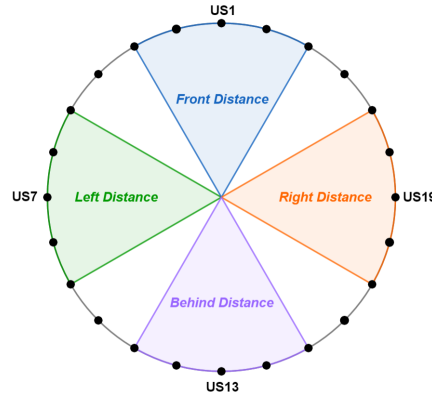


FIGURE 3 – Choix

Pour chacun de ses regroupements, on va alors conserver la distance minimale relevée par l'un des capteurs et en fonction de la valeur de celle-ci, le robot prendra alors l'une des 4 directions possibles :

- Move Forward
- Slight Right Turn
- Sharp Right Turn
- Slight Left Turn

On pourra se référer à [2] pour la description de l'algorithme des prises de décision par le robot en fonction des distances relevées par les capteurs.

On a donc à notre disposition un jeu de données robotiques où pour chaque relevé, 24 données de perception correspondent aux différentes distances et à chaque étape, le robot peut prendre 4 directions différentes. Le dataset obtenu est donc labellisé, autrement dit pour chaque relevé, on connaît la direction qui a été prise par le robot.

2 Data-augmentation et Réseaux de neurones utilisés

Dans le but d'étoffer notre jeu de données, nous avons fait de la data augmentation de sorte à avoir un dataset composé de 7500 données. Pour ce faire, nous avons choisi aléatoirement des échantillons du dataset auxquelles on a ajouté à chaque distance obtenue une perturbation aléatoire suivant une loi normale centrée d'écart-type suffisamment petit par rapport à la précision des capteurs pour ne pas fausser les données.

Nous avons fait le choix d'apprendre le dataset à notre disposition avec un réseau de neurones de type perceptron multi-couches et plus précisément, avec cinq couches cachées et la dernière couche composée de la fonction d'activation softmax nous permettant d'obtenir les probabilités des différentes

directions possibles. Dans le but de comparer nos résultats, nous avons décidé d'entraîner deux réseaux de neurones : un premier où les fonctions d'activation des couches cachées étaient TanH et un autre avec ReLu.

Pour l'entraînement des réseaux de neurones, nous avons pris un batch de 500 échantillons du dataset avec un paramètre de learning rate égal à 0.005 pour 50 epochs (respectivement 100 epochs) lorsque la fonction d'activation choisie était ReLu (respectivement TanH) . Ceci nous a permis d'avoir un taux d'accuracy pour ReLu de 97.58% pour le dataset d'apprentissage et 90.60% pour le dataset de test (respectivement pour TanH, 93.46% pour le dataset d'apprentissage et 90.34% pour le dataset de test).

On peut voir ci-dessous les courbes d'entraînement et de validation obtenues une fois l'apprentissage fini dans le cas de TanH, ce qui nous permet de confirmer la pertinence des résultats obtenus.

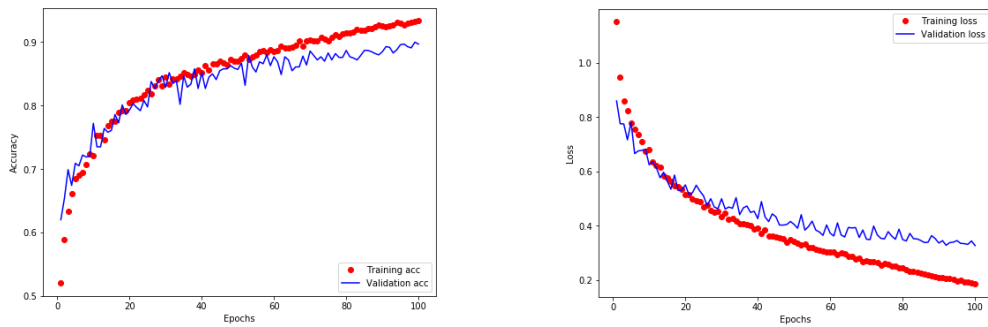


FIGURE 4 – Courbes Accuracy and Loss

3 Expériences et Résultats

On va commencer par choisir 4 échantillons au hasard dans le dataset qui correspondent aux 4 directions différentes choisies par le robot. Pour chacun d'entre eux, nous allons représenter d'une part les valeurs des distances des capteurs aux objets les plus proches et d'autre part, les probabilités associées aux quatre directions.

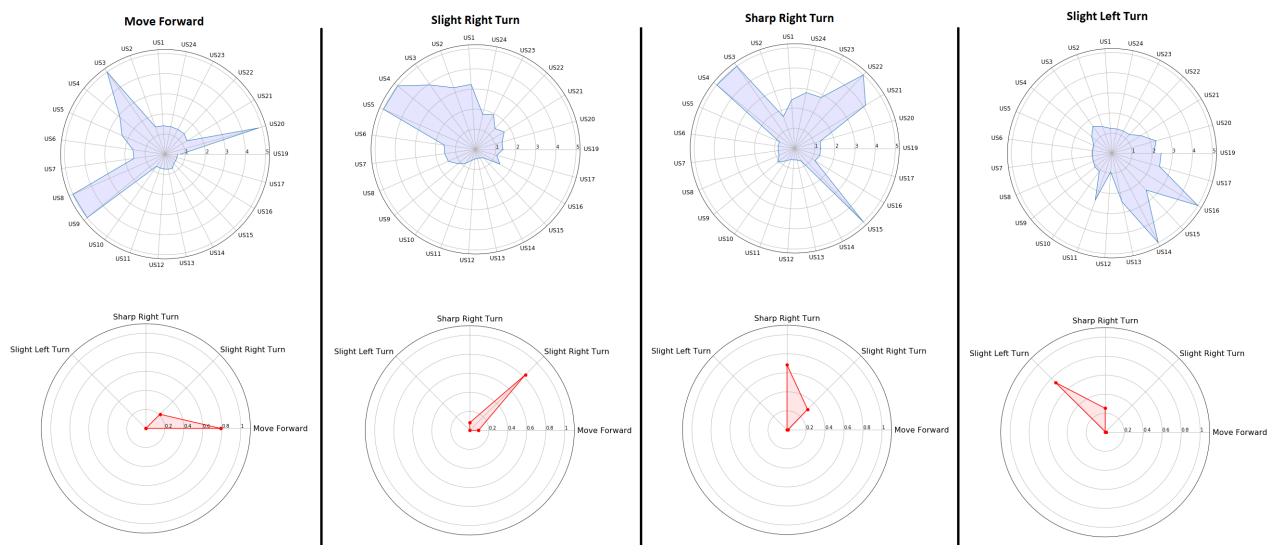


FIGURE 5 – Choix

Ces exemples permettent de confirmer que la direction qui a été prise par le robot est intrinsèquement liée aux valeurs des distances relevées par les capteurs, et surtout aux quatre regroupements qui ont été choisis pour les directions à prendre.

3.1 Non Pondéré

Activation Function	Sucess Rate		Average Number of Modified Sensors		Distance from Initial Perception (cm)	
nb rep = 10						
	Targeted	Untargeted	Targeted	Untargeted	Targeted	Untargeted
TanH	2.19 %	6.13 %	6.61	5.89	1.54	1.34
ReLu	2.11 %	5.99 %	7.70	4.63	1.77	1.08
nb rep = 20						
	Targeted	Untargeted	Targeted	Untargeted	Targeted	Untargeted
TanH	2.45 %	7.13 %	6.33	5.98	1.63	1.53
ReLu	2.89 %	7.88 %	8.33	4.52	2.11	1.19

3.2 Pondéré

Activation Function	Sucess Rate		Average Number of Modified Sensors		Distance from Initial Perception (cm)	
nb rep = 10						
	Targeted	Untargeted	Targeted	Untargeted	Targeted	Untargeted
TanH	2.30%	6.88 %	7.18	7.82	1.65	1.77
ReLu	3.26 %	6.22 %	10.1	4.25	2.21	0.966
nb rep = 20						
	Targeted	Untargeted	Targeted	Untargeted	Targeted	Untargeted
TanH	2.53 %	7.69 %	7.03	6.52	1.83	1.70
ReLu	3.70 %	7.88 %	9.44	4.63	2.29	1.20

Références

- [1] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [2] A L Freire, G A. Barreto, M Veloso, and A T. Varela. Short-term memory mechanisms in neural network learning of robot navigation tasks : A case study. 2009.