Apprentissage par renforcement

Alexandre NOURY et Antoine GERMAIN

Q-learning

Matrice Q, initialisation, itération

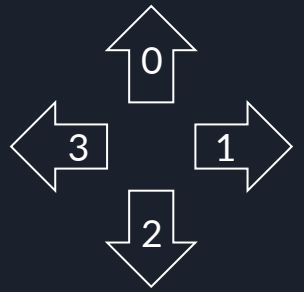
Q-learning

La matrice Q(s,a) est une matrice de cols*rows lignes et 4 colonnes : à chaque paire (case du labyrinthe,action possible depuis cette case) sera associée une récompense.

s est le numéro de la case dans le sens de lecture

a est la direction de déplacement

0	1	2	3
4	5	6	7
…	…	…	…



Q-learning

On initialise notre matrice comme matrice nulle.

Puis on l'entraîne sur le labyrinthe étudié en utilisant une politique ε-greedy : à chaque instant, on se déplace vers la case apportant la plus grande récompense dans l'état actuel des choses avec une probabilité 1-ε, et vers une case au hasard avec une probabilité ε.

On met ainsi à jour la matrice Q grâce aux récompenses en données du problème : r = -1 pour un déplacement, -1 000 pour un mur et 100 pour l'arrivée.

$Q(s, a) \leftarrow \alpha * Q(s, a) + (1-\alpha) * [r + \gamma * \max\{Q(s', a') / a'=0,1,2,3\}]$

Implémentation et obtention de la solution

Main, initialisation de Q, itération, obtention de la solution

Implémentation

```c
int main() {
    char mot[20]; // Déclaration d'une variable pour stocker le mot (20 est la taille maximale du mot)
    char nomLabyrinthe[100]; // Déclaration d'une variable pour stocker le chemin vers le labyrinthe
    printf("Veuillez renseigner le labyrinthe : ");
    scanf("%s", mot);
    sprintf(nomLabyrinthe, "../data/%s.txt", mot);

    mazeEnv_make(nomLabyrinthe);
    mazeEnv_render();
    float** Q;
    matriceQ(&Q);
    for (int iteration = 0;iteration<100;iteration++) {
        iterationQ(Q);
    }

    solutionQ(Q);
```

Implémentation

```
/*Données*/
float alpha=0.8; //facteur d'apprentissage
float mazegamma=1; //facteur d'actualisation
float epsilon=0.1; //chance de choisir aléatoirement une direction dans la politique epsilon-greedy
/*Récompenses*/
int arrivée = 100;
int mur = -1000;
int deplacement=-1;
int piege = -5;
```

Implémentation

```
/*Choix de l'action suivant politique epsilon-greedy*/
while (test==1) {
float p = rand()/RAND_MAX;
if (p>epsilon){
    a = argmaxQ(current_col, current_row, Q);
}
else {
    a = rand()%4;
}
new_row=newrow(a, current_row);
new_col=newcol(a, current_col);
/*Si on n'est pas sortis du labyrinthe on peut continuer l'exploration*/
if (new_row != -1 && new_row != rows && new_col != -1 && new_col != cols) {
    test = 0;
}
}
```

Implémentation

```
/*Constitution du chemin grâce à la matrice Q*/
void solutionQ(float** Q) {
    int current_row=start_row;
    int current_col=start_col;
    int a;
    /*Q étant construite il suffit de chercher la récompense maximale pour savoir où aller*/
    while (current_row!=goal_row || current_col!=goal_col) {
        a = argmaxQ(current_row, current_col, Q);
        current_row=newrow(a,current_row);
        current_col=newcol(a,current_col);
        mazeEnv[current_row][current_col]='.';
    }
    mazeEnv[current_row][current_col]='g';
}
```

Difficultés rencontrées

Compréhension du Q-learning, fonctions auxiliaires, tests des étapes

Difficultés rencontrées

- Compréhension du fonctionnement du Q-learning
- Erreurs dans les nombreuses fonctions auxiliaires (max, argmax, ...)
- Tests de codes et visualisation des étapes
- Difficulté pour afficher les labyrinthes (code du retour à la ligne)

Résultats

Différents labyrinthe, scanf du nom du fichier texte, pièges

Résultats

```
antoine.germain@byod:~/in104/ApprentissageRenforcement$ make clean
rm -f src/qlearning.o src/qlearning src/dfs src/dfs.o src/functions.o src/mazeEnv.o
antoine.germain@byod:~/in104/ApprentissageRenforcement$ make
gcc -Wall -Werror -Wfatal-errors -I include/ -g -o src/qlearning.o -c src/qlearning.c
gcc -Wall -Werror -Wfatal-errors -I include/ -g -o src/functions.o -c src/functions.c
gcc -Wall -Werror -Wfatal-errors -I include/ -g -o src/mazeEnv.o -c src/mazeEnv.c
gcc -Wall -Werror -Wfatal-errors -I include/ -g -o src/qlearning src/qlearning.o src/functions.o src/mazeEnv.o
antoine.germain@byod:~/in104/ApprentissageRenforcement$ cd src
antoine.germain@byod:~/in104/ApprentissageRenforcement/src$ ./qlearning
Veuillez renseigner le labyrinthe : maze1
```

Résultats

Résultats

Conclusion

Limites, pistes d’amélioration

Conclusion

On a implémenté notre propre résolution en se basant uniquement sur les fonctions de stockage et d'affichage du labyrinthe de MazeEnv. On a pu ajouter une fonctionnalité personnalisée avec les pièges sur la route. Ayant codé la majorité de l'environnement, nous pouvons facilement le modifier ou ajouter des fonctionnalités.

Prolongements à faire :

- implémenter d'autres politiques
- implémenter des bonus sur les routes du labyrinthe
- adapter à d'autres jeux