



Universidade do Minho
Departamento de Informática

Perfil Sistemas Inteligentes
Computação Natural
Edição 2018/2019

2º Trabalho prático

Tema	Reinforcement Learning aplicado a agentes
Objectivos de aprendizagem	<p>Com a realização deste trabalho prático, pretende-se que os alunos aprendam os seguintes procedimentos:</p> <ul style="list-style-type: none">• Preparação de um modelo de Reinforcement Learning• Análise e validação da performance de um modelo Reinforcement Learning• Optimização de métodos de pesquisa de política aplicados no modelo de Reinforcement Learning
Enunciado	<p>Este enunciado pretende ser o ponto de partida para o desenvolvimento de um algoritmo de Reinforcement Learning (RL) utilizando o ambiente de desenvolvimento Python. Para isso, será necessário o desenvolvimento de uma solução para o seguinte problema:</p> <p><i>“Preparação de um modelo RL capaz de aprender a jogar o famoso jogo: Pacman”</i></p> <p>Neste projeto, tenciona-se aplicar a matéria lecionada durante a unidade curricular de Computação Natural para a criação de um agente informático capaz de aprender a jogar (e ganhar!) o jogo Pacman, através da implementação de algoritmos RL. Para isso, serão disponibilizados códigos padrão do jogo, possibilitando aos grupos focar-se no desenvolvimento do modelo RL. Tendo em conta este objetivo, o projeto será dividido em 2 fases:</p> <p>(1) uma primeira fase dedicada ao treino do modelo, onde o agente começará por aprender sobre os valores das posições e ações. Tendo em conta o largo período de tempo necessário a aprender os Q-value, será disponibilizado um código que possibilitará treinar o agente a aprender os mecanismos do jogo no modo silencioso, sem exibir a GUI/terminal;</p> <p>(2) uma vez que o treino se encontre completo, passamos para o modo de teste (onde os valores de ϵ e α são alterados para o valor 0, interrompendo efetivamente o processo de Q-learning e desativando a exploração, a fim de permitir que o agente explore a política aprendida). Durante este processo, a GUI padrão será apresentada ao utilizador. Para a execução de todo este processo para mapas pequenos, o grupo deverá executar o seguinte comando (outros esquemas de mapas encontram-se disponíveis na pasta layouts):</p> <pre>python pacman.py -p PacmanQAgent -x 2000 -n 2010 -l smallGrid</pre> <p>Além disso, os grupos deverão otimizar o modelo através da adoção de outros parâmetros de aprendizagem (epsilon, alpha, gamma). Para isso, deverão utilizar o argumento “-a epsilon=X,alpha=Y,gamma=Z”. Os valores a otimizar deverão ser selecionados usando para isso o algoritmo de Particle Swarm Optimization. Outros métodos poderão ser utilizados, justificando o seu uso através da comparação dos diferentes resultados.</p> <p>Outro comando útil para fins de visualização do processo de treino, p.e., para possibilitar a análise visual de 10 jogos de treino, apresenta-se disponível o seguinte comando:</p> <pre>python pacman.py -p PacmanQAgent -n 10 -l smallGrid -a numTraining=10</pre>

Como ponto de referência, no final do treino o agente deverá ser capaz de vencer pelo menos 80%-90% dos jogos de teste em mapas pequenos.

(+2 valores) Tarefa extra: análise do treino/teste do algoritmo RL desenvolvido para um mapa de tamanho médio. Nesta tarefa, deverão:

- analisar e comparar os resultados provenientes da utilização de diferentes mapas/ambientes;
- listar o conjunto de limitações provenientes de mais estados e acções;
- definir um conjunto de medidas necessárias para a optimização do algoritmo de RL.

Ficheiro a editar:

- `qlearningAgents.py` - agentes Q-learning a utilizar no jogo Pacman.

Ficheiros a ter em conta:

- `learningAgents.py` - define a base da classe `QLearningAgent`.
- `util.py` - apresenta algumas variáveis importantes para o método Q-learning.
- Ficheiros associados ao jogo Pacman, p.e., `pacman.py`.

O trabalho prático compreende a entrega do código desenvolvido e do relatório, definindo todos os procedimentos aplicados e respetiva justificação da sua utilização, apoiando-se na demonstração dos resultados adquiridos.

Poderão utilizar quaisquer bibliotecas que o grupo entender úteis para a resolução do problema (tendo em conta os ambientes de desenvolvimento definidos).

Entrega

O código resultante da realização do trabalho prático e o respetivo relatório em formato digital .PDF deverão ser enviados por correio eletrónico para cesar.analide@di.uminho.pt e fgoncalves@algoritmi.uminho.pt, em ficheiros compactados (formato ZIP). Tanto o assunto da mensagem como o ficheiro deverão ser identificados na forma “[CN: F2GXX]”, em que [XX] designa o número do grupo de trabalho.

A sessão de apresentação do trabalho prático terá lugar no dia 20 de maio de 2019, em formato a anunciar oportunamente.

Referências

Sutton, Richard S., and Andrew G. Barto. Introduction to reinforcement learning. Vol. 135. Cambridge: MIT press, 1998.

WATKINS, Christopher JCH; DAYAN, Peter. Q-learning. *Machine learning*, 1992, 8.3-4: 279-292.

<http://ai.berkeley.edu/reinforcement.html>

<https://pyswarms.readthedocs.io/en/latest/examples/tutorials.html>