Inteligência Artificial para Robótica – IAR (2024/2025)
Enunciado do projeto prático – Parte 2 (v1.0)

Informações gerais

- Este documento apresenta um conjunto de instruções relativas à realização da Parte 2 do Projeto Prático sob a forma de trabalho autónomo, por grupos de 3 ou 4 alunos, grupos esses definidos de acordo com as regras indicadas nas aulas.
- Apesar do enunciado do projeto ser disponibilizado por partes, o projeto será entregue como um todo pelo grupo num momento único.
- O Projeto Prático como um todo terá de ser defendido numa sessão de discussões orais, onde todos os elementos do grupo terão de estar presentes; todos os elementos do grupo terão de ser capazes de explicar todos os elementos entregues pelo grupo.
- Sempre que se considerar necessário, serão adicionadas dicas no Moodle junto a este documento, sendo que a realização e análise cuidada dos vários laboratórios é essencial para a realização deste trabalho.
- Algumas dicas poderão ser fornecidas durante as aulas teóricas, práticas e laboratoriais, pelo que se torna essencial que pelo menos um dos elementos do grupo esteja sempre presente em todas as aulas.
- Fontes disponíveis na internet (e.g., manuais, tutoriais, vídeos Youtube, ChatGPT) podem ser consultadas como fonte de aprendizagem e inspiração, mas todo o projeto (análise, desenho e implementação) terá de ser da autoria do grupo.
- Todo o projeto, desde a sua conceção, implementação, validação e reporte terá de ser realizado única e exclusivamente pelos elementos do grupo, ou seja, sem qualquer tipo de recurso a outras pessoas ou serviços.
- De forma a maximizar o processo de aprendizagem e a garantir que o grupo está a dirigir-se na direção certa, aconselha-se que os estudantes procurem obter feedback dos docentes durante o desenvolvimento do projeto, recorrendo às aulas e aos horários de dúvidas, mas não por e-mail.

Objetivos Gerais

Na Parte 2 do Projeto Prático, os grupos explorarão a utilização do algoritmo PPO para o treino de um controlador para o robô Thymio. O controlador terá de garantir que o robô, arrancando da zona central do ambiente com qualquer orientação, explorar o ambiente sem colidir com obstáculos ou cair de precipícios. Para tal, o controlador fará uso de uma rede neuronal que escolherá a velocidade a atribuir a cada roda do robô em função das observações recolhidas através dos sensores de proximidade do robô: os 5 sensores de proximidade frontais e os dois sensores de chão (virados para baixo).

Requisitos

Numa primeira fase, o grupo terá de explorar a utilização algoritmo base PPO[1]. De seguida, o grupo terá de explorar a utilização da versão recorrente do algoritmo PPO (i.e., com LSTM[2]), denominada RecurrentPPO[3]. A versão recorrente é essencial para que o robô possa realizar as suas escolhas com

[1] https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html
[2] https://developer.nvidia.com/discover/lstm
[3] https://sb3-contrib.readthedocs.io/en/master/modules/ppo_recurrent.html

base em memória de observações passadas, nomeadamente para que possa recuar e rodar após detetar um precipício.

A recompensa deverá ser definida de tal forma que o robô treinado:

- Evita cair em precipícios;
- Desvia-se dos obstáculos;
- Explora o espaço o mais possível;
- Evita revisitar regiões do espaço;
- Prefere velocidades lineares positivas.

O grupo terá também de ter em conta que:

- O espaço de ações inclui as velocidades angulares a aplicar nos motores do robô, podendo aplicar um fator de escala às saídas da rede neuronal;
- O espaço de observações inclui as leituras normalizadas dos sensores de proximidade e de chão (e.g., normalizadas para [0,1]);
- A queda do robô pode ser antecipada observando uma súbita mudança de inclinação;
- No início de cada episódio, o robô tem de ser posicionado no centro do mundo e a sua orientação escolhida aleatoriamente;
- No início de cada episódio é necessário colocar um conjunto de obstáculos (e.g., paralelepípedos) no ambiente, com dimensões, posições e orientações aleatórias (o grupo pode usar o código fornecido no Laboratório 1);
- O robô tem de ser capaz de desviar-se de obstáculos e evitar quedas em qualquer ambiente planar preenchido com caixas, ou seja, o robô tem de operar corretamente também em ambientes não observados durante o treino.

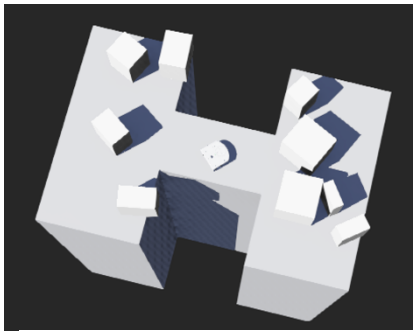

Figura 1 - Exemplo de condição inicial

Material de Suporte

O desenvolvimento do projeto deverá basear-se no *template* fornecido no ficheiro .zip anexo a este enunciado. Esse *template* inclui:

- thymio_rl_env.wbt:
  - versão base de ambiente Webots onde o robô simulado será treinado.
- thymio_rl_controller.py:
  - versão base de um controlador para o treino de um Thymio simulado.
- real_thymio_interaction_template.py:
  - versão base de um controlador para um robô Thymio real.
  - execução:
    1. inicie o ThymioSuite;
    2. ligue-se o ThymioSuite ao robô conectado por USB;
    3. execute: `$python3 real_thymio_interaction_template.py`

O grupo deverá ter em conta os seguintes elementos durante o desenvolvimento do projeto:

- Para treinar o agente o grupo utilizará o pacote *Stable Baselines3* (SB3)[4].
- Para utilizar o SB3, é necessário que o Webots seja abstraído como um OpenAI Gym[5].
- O *template* fornecido inclui o código base para abstrair o Webots como um OpenAI Gym.

[4] https://stable-baselines3.readthedocs.io/en/master/
[5] https://www.gymlibrary.dev/index.html

- O grupo pode analisar o exemplo incluído no Webots em:
  - **samples->howto->openai_gym->openai_gym.wbt**.
- A documentação do OpenAI Gym contém exemplos de como criar ginásios *custom*[6], úteis para construir um ambiente de treino para o robô Thymio.
- A documentação do PPO[7] e do RecurrentPPO[8] contém exemplos de como usá-los para treinar e executar controladores, incluindo como configurar o treino e as redes que implementam a *policy* e a *value function*.
- É importante estudar na documentação do PPO[9] e do RecurrentPPO[10] que parâmetros de entrada podem ser utilizados na criação dos modelos (ex.: *learning rate*, *batch size*).
- O código fornecido nos **Laboratórios 1** e **7** (ver Moodle) também poderá ser útil.

Instruções de instalação

Para o desenvolvimento no robô simulado, é necessário realizar os seguintes passos de instalação:

1. Instalar a versão **3.7.9** do python, fazendo (necessário adaptar):
```
$ pyenv install 3.7.9
$ [path]/.pyenv/versions/3.7.9/bin/python3.7 -m pip install --upgrade pip
```

2. Instalar o gym e a StableBaselines3 (necessário adaptar):
```
$ [path]/.pyenv/versions/3.7.9/bin/python3.7 -m pip install gym
$ [path]/.pyenv/versions/3.7.9/bin/python3.7 -m pip install 'stable-baselines3[extra]'
```

3. No Webots, definer o *python command* nas preferências como (necessário adaptar):
```
[path]/.pyenv/versions/3.7.9/bin/python3.7
```

Para o desenvolvimento no robô real, é necessário realizar os seguintes passos de instalação:

1. Instalar o ThymioSuite de https://www.thymio.org/download-thymio-suite/
2. Instalar o tdmclient (necessário adaptar):
```
$ [path]/.pyenv/versions/3.7.9/bin/python3.7 -m pip install tdmclient
```

Entregáveis obrigatórios para projeto ser aceite para avaliação

1. Código fonte:
   a. Controladores dos robôs (real e simulado);
   b. Ficheiros com parâmetros das redes neuronais treinadas;
   c. Ficheiro de configuração do Webots.
2. Relatório técnico:
   a. Descrição detalhada da implementação do ambiente de treino;
   b. Descrição detalhada da implementação dos controladores;
   c. Análise comparativa, suportada por tabelas e gráficos, do impacto dos vários elementos do ambiente de aprendizagem no desempenho do robô e do próprio processo de aprendizagem, nomeadamente:

[6] https://www.gymlibrary.dev/content/environment_creation/
[7] https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html
[8] https://sb3-contrib.readthedocs.io/en/master/modules/ppo_recurrent.html
[9] https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html
[10] https://sb3-contrib.readthedocs.io/en/master/modules/ppo_recurrent.html

i. Avaliar a importância de cada elemento da função de recompensa através de estudos de ablação (i.e., isolar os vários elementos da função de recompensa);
ii. Quantificar a vantagem de se usar uma LSTM ao invés de um MLP não-recorrente;
iii. Comparar o desempenho quando se usa *tanh* e ReLU como funções de ativação;
iv. Quantificar a vantagem de se usar geração aleatória do ambiente versus utilizar um ambiente constante ao longo de todo o treino.

Critérios de avaliação

1. Desempenho Geral (40%):
    a. Média de área coberta nos ambientes de teste;
    b. Nº de colisões/quedas nos ambientes de testes;
    c. Os ambientes de teste incluirão ambientes desconhecidos do robô e do grupo;
    d. Haverá ambientes de teste reais e simulados, sendo que será dado mais peso ao desempenho do robô simulado do que ao do robô real.
2. Eficácia e robustez do treino (30%):
    a. Melhoria da recompensa média por episódio;
    b. Originalidade da função de recompensa;
    c. Originalidade da geração automática dos ambientes de treino;
    d. Diversidade dos ambientes de treino.
3. Qualidade do relatório técnico (20%):
    a. Detalhe, clareza, qualidade e rigor da descrição da implementação;
    b. Detalhe, clareza, qualidade e rigor da análise comparativa.
4. Qualidade do Código (10%):
    a. Modularidade, documentação e reprodutibilidade.

Data-limite para entrega final

- Os entregáveis obrigatórios da Parte 1 e da Parte 2 terão de ser enviados por e-mail para sancho.oliveira@iscte-iul.pt até às 14:00 h de 6 de junho 2025.
- As discussões orais decorrerão na data atribuída para avaliação de 1ª época, a definir pelos serviços do Iscte.
- Assim que os serviços do Iscte definam a data de avaliação de 1ª época, será produzida uma nova versão deste documento com a respetiva informação.

Prática de plágio

- O projeto é para ser desenvolvido na sua totalidade pelos elementos do grupo, sem acesso ou partilha de informação específica ao projeto com elementos externos ao grupo, com exceção do docente;
- A deteção da prática de plágio resultará no desencadear de um processo disciplinar, como recomendado pelo Conselho Pedagógico do ISCTE-IUL.