Estudo de caso de Deep Q-Learning

Vítor Kei Taira Tamada, Orientador: Denis Deratani Mauá

Departamento de Ciência da Computação, Instituto de Matemática e Estatística, Universidade de São Paulo

Introdução

A proposta do trabalho foi estudar a eficiência de *deep Q-learning* em três ambientes com características distintas.

Utilizando técnicas de aprendizado de máquina como **aprendizado por reforço** e **redes neurais convolucionais**, o programa enxerga apenas a tela (mapa em um dos domínios) e desenvolve um modelo a partir do qual busca obter sucesso no ambiente.

O objetivo do trabalho foi aprender mais sobre as ferramentas e técnicas de aprendizado de máquina, sobre os parâmetros e funções que uma arquitetura da inteligência artificial deve ter e quais são seus efeitos.

Ambientes

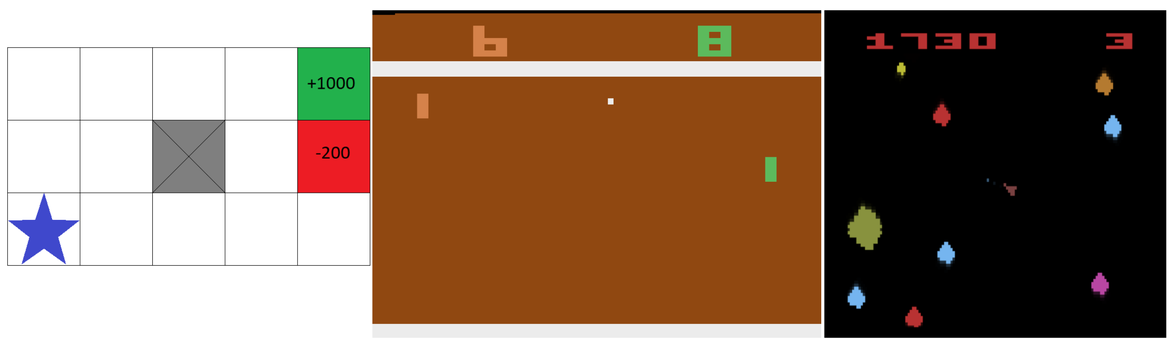

Figura: *Gridworld*, *Pong* e *Asteroids* respectivamente

- ***Gridworld***: Exemplo clássico em estudo e ensinamentos de aprendizado por reforço. Consiste em um mapa de espaços quadrados com objetivo e armadilhas. O agente começa em um espaço pré-determinado e deve tentar encontrar o caminho até o objetivo enquanto explora quais recompensas cada espaço gera.
- ***Pong***: Jogo em que o jogador controla uma das duas barras verticais presentes na tela e tem uma bola movendo-se de um lado para o outro. O objetivo é fazer a bola passar da barra adversária e chegar no final da tela para marcar ponto.
- ***Asteroids***: Jogo em que o jogador controla uma nave e deve destruir os asteroides que atravessam a tela para ganhar pontos enquanto evita contato físico com eles. O objetivo do jogo é alcançar a maior pontuação possível antes de perder todas as vidas.

Ferramentas

Figura: *Python*, *OpenAI*, *Gym*, *TensorFlow*, *GitHub* e *Stella* [7] respectivamente

- **Python3** [1]: Linguagem utilizada neste trabalho. Tem grande suporte para estudos em inteligência artificial e é usada pelas outras ferramentas.
- **Gym** [2]: Plataforma da companhia **OpenAI** [3] para estudo e pesquisa de aprendizado por reforço. Sua variante, **Gym-Retro** [4], é especializada em jogos antigos. Auxiliam na emulação dos ambientes deste trabalho.
- **TensorFlow** [5]: Arcabouço da Google que oferece grande suporte ao estudo de desenvolvimento de aprendizado de máquina e aprendizado profundo.
- **GitHub** [6]: Controlador de versão. Também ajudou a transferir arquivos do trabalho entre máquina pessoal, da Rede Linux e do servidor da Rede IME.

Técnicas de aprendizado

- **Aprendizado por reforço:** Técnica de aprendizado de máquina que deixa o agente descobrir um ambiente interagindo com ele e recebendo recompensas e penalidades pelas ações. Assemelha-se a forma como seres humanos descobrem uma ferramenta ou um jogo quando não leem um manual.

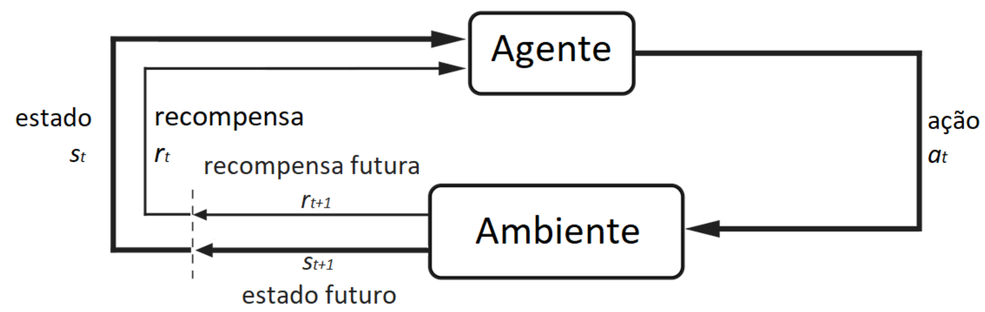

Figura: Esquema da interação agente-ambiente[8]. Tradução e adaptação feitas pelo autor.

As recompensas e penalidades determinam a **qualidade das ações nos estados**:

$$Q^*(S,A) = \sum_{S'} P(S'|S,A)[R(S,A,S') + \gamma \max_{A'} Q^*(S',A')] \quad (1)$$

- **Rede neural convolucional:** Variante de rede neural profunda muito utilizada em análise de imagens. Consiste em quebrar uma imagem (matriz) de entrada em várias menores e tentar detectar características em cada uma delas. Precisa que os exemplos de treinamento estejam rotulados.

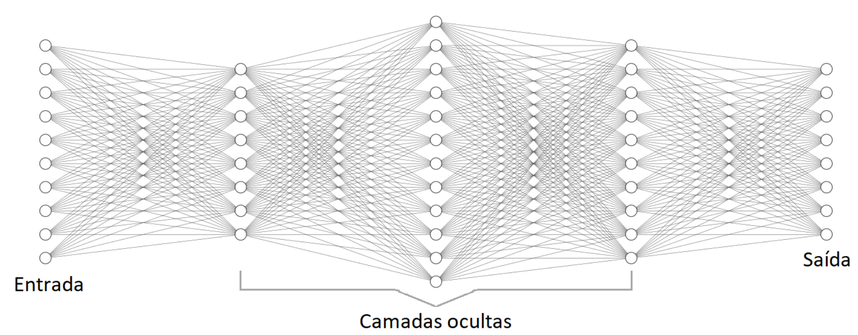

Figura: Esquema de uma rede neural profunda. [9]

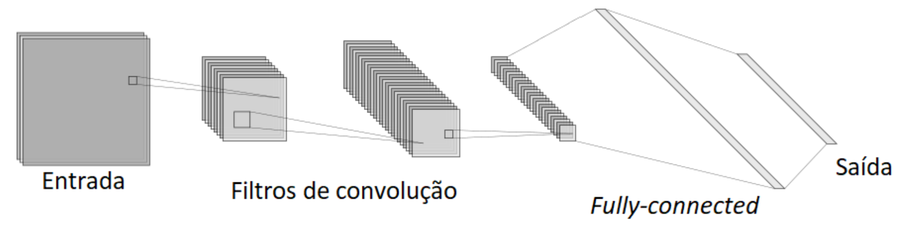

Figura: Esquema do funcionamento de uma rede neural convolucional. [10]

- **Deep Q-Learning:** União de aprendizado por reforço com rede neural convolucional. As vantagens de uma técnica compensam as desvantagens da outra para se ter uma estrutura estável que aprende com imagens sem precisar de exemplos rotulados.

Experimentos

Os experimentos consumiram a maior parte do tempo do trabalho por conta dos treinamentos que podem levar alguns minutos, como foi o caso do *Gridworld*, ou até mesmo horas, passando de um dia para o outro, como foi o caso do *Asteroids* e do *Pong* em alguns casos. No caso do *Gridworld*, avaliou-se a capacidade do agente encontrar um caminho até o objetivo do mapa. Para o *Pong* e *Asteroids*, foram avaliadas as respectivas pontuações atingidas ao final dos episódios e do modelo construído ao final do treinamento.

Resultados

Os resultados se mantiveram, na maior parte, dentro do esperado.

- ***Gridworld***: Sucessos consistentes em encontrar o caminho até o objetivo, mesmo com variações nos hiper-parâmetros.
- ***Pong***: Resultados promissores, mas não tão bem sucedidos quanto no *Gridworld*. A arquitetura mostrou-se sensível a mudanças e o aprendizado foi lento, mas positivo.
- ***Asteroids***: Resultados pouco promissores. A pontuação obtida ao longo do treinamento e do modelo desenvolvido ficou abaixo da faixa de pontuação aleatória.

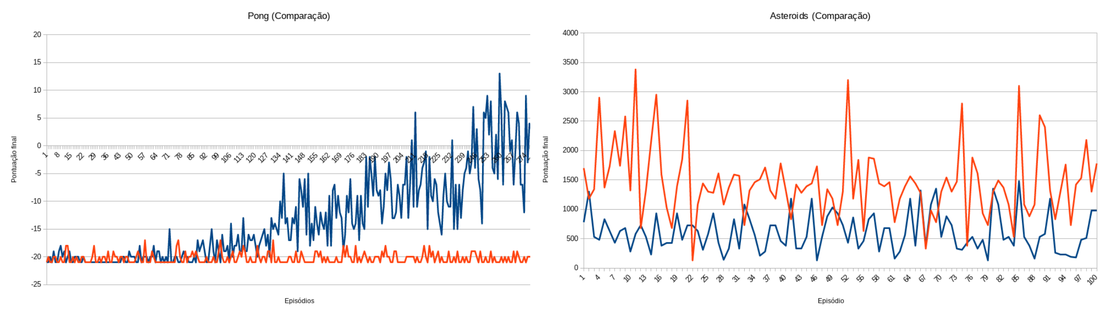

Figura: As linhas azuis são a pontuação do agente treinado. As linhas vermelhas são a pontuação do agente aleatório. 298 episódios de Pong e 100 episódios de *Asteroids* correspondem a aproximadamente 1 milhão de *frames* cada.

	Gridworld	*Pong*	*Asteroids*
Humano	-	7	1943.3
Aleatório	1.74%	-20.27	1458.1
DQL	66.15%	21	607.8

Tabela: ***Gridworld***: % média de chegada no objetivo ao longo do treinamento; pontuação de humano não aplicável. ***Pong***: Pontuação média; humano experiente após cerca de 1 hora de jogo. ***Asteroids***: Pontuação média; humano amador após cerca de 1 hora de jogo.

Conclusão

A técnica *deep Q-learning* mostrou-se difícil de usar por conta do impacto que os hiper-parâmetros têm no aprendizado, principalmente em ambientes complexos. Porém, isso é compensado pela sua capacidade de criar modelos capazes de solucionar diversos problemas que as técnicas base não conseguem individualmente ou são custosas demais para isso.

Referências

[1] Página oficial da *Python Software Foundation*: **https://www.python.org/**

[2] Página oficial da plataforma Gym: **https://gym.openai.com/**

[3] Página oficial da companhia OpenAI: **https://openai.com/**

[4] Página oficial da plataforma Gym-Retro: **https://blog.openai.com/gym-retro/**

[5] Página oficial do arcabouço TensorFlow: **https://www.tensorflow.org/**

[6] Página oficial da ferramente GitHub: **https://github.com/**

[7] Página do emulador Stella: **https://stella-emu.github.io/**

[8] Sutton, R. S., and Barto, A. G., *Reinforcement learning: an introduction*.

[9] Diagrama construído pela ferramenta: **http://alexlenail.me/NN-SVG/index.html**

[10] Diagrama construído pela ferramenta: **http://alexlenail.me/NN-SVG/LeNet.html**

IME - Instituto de Matemática e Estatística