Deep Reinforcement Learning : Breakout

7/jun/2021

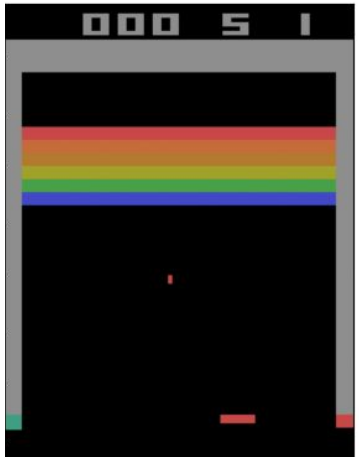
Grupo 7
Carolina Marques - PG42818
Constança Elias - PG42820
Maria Barbosa - PG42844
Renata Ribeiro - A86271

Sistemas Inteligentes
Computação Natural

Conteúdos

Breakout Deterministic

Descrição do Ambiente



(210, 160, 3)

Ações possíveis:

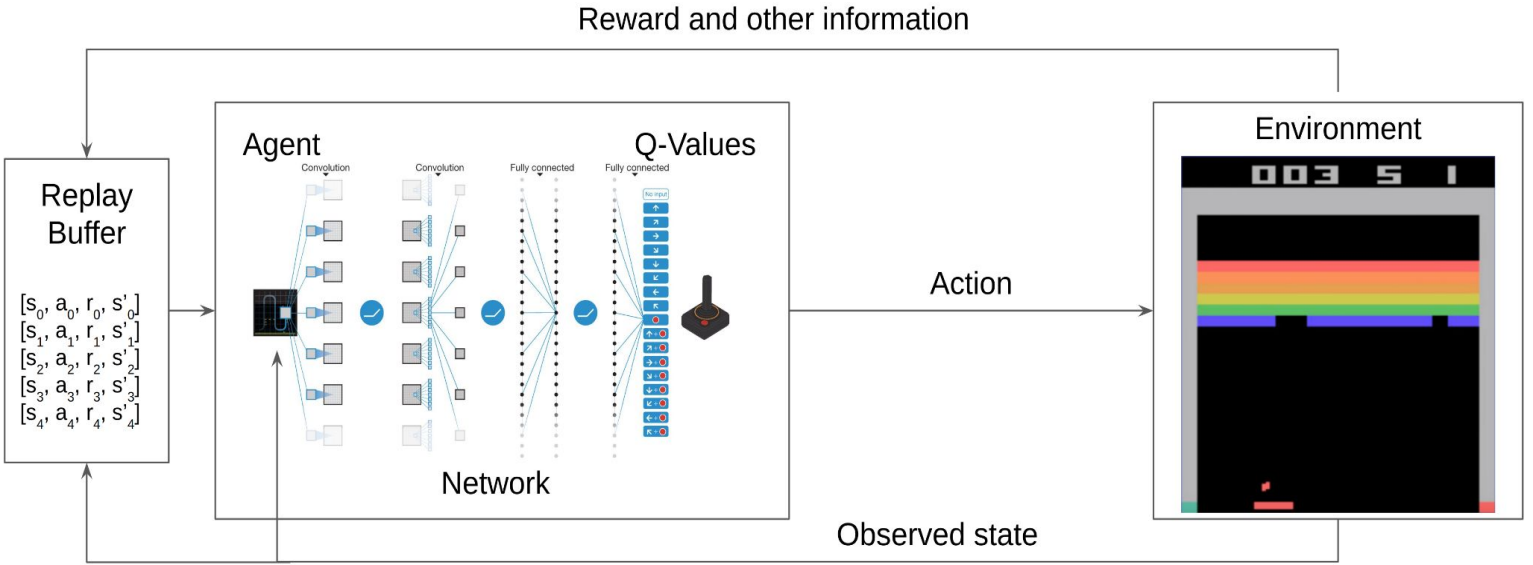
- **NOOP:** não fazer nada
- **FIRE:** disparar a bola no início do jogo e de cada vida
- **LEFT**: mover o tijolo para a esquerda
- **RIGHT**: mover o tijolo para a direita

Pré-Processamento

3 passos:

- RGB para escala de cinzentos;
- Redimensionar frame para 84 x 84;
- Cortar frame.

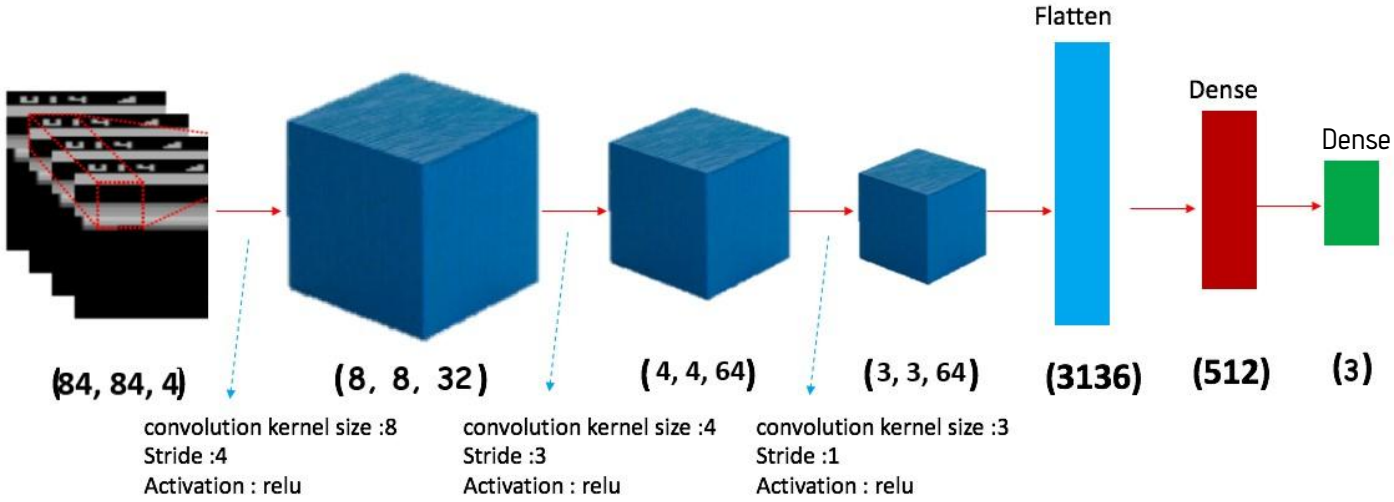
Algoritmo de DRL



***Fonte**: Building a Powerful DQN in TensorFlow2.0, Sebastian Theller*

Algoritmo de DRL

Rede Neuronal Convolucional



Otimização do Algoritmo

Rede Otimizada #1

- Convolucional com 32 *kernels* de 8x8 e *stride* 4x4 com ativação relu, inicialização *Variance Scaling* (com *scale* igual a dois);
- Convolucional com 64 *kernels* de 4x4 e *stride* 2x2 com ativação relu, inicialização *Variance Scaling* (com *scale* igual a dois);
- Convolucional com 64 *kernels* de 3x3 e *stride* 1x1 com ativação relu, inicialização *Variance Scaling* (com *scale* igual a dois);
- Convolucional com 1024 *kernels* de 7x7 e *stride* 1x1 com ativação relu, inicialização *Variance Scaling* (com *scale* igual a dois);
- Flatten;
- Dense com 1 *output* e inicialização *Variance Scaling* (com *scale* igual a dois);
- Flatten;
- Dense com 3 *outputs* e inicialização Variance Scaling (com *scale* igual a dois);
- Optimizador Adam com *learning rate* igual a 1e-4 e *Huber Loss function*.

Fonte: *Building a Powerful DQN in TensorFlow2.0, Sebastian Theller*

Otimização do Algoritmo

Rede Otimizada #1

Nesta arquitetura:

- Dois fluxos para estimar o valor de um estado:
  - Estimar o valor de um estado;
  - Estimar a vantagem de realizar uma ação num dado estado;

Permite aprender de forma intuitiva quais os estados mais valiosos sem necessitar de realizar ações nesse estado.

Otimização do Algoritmo

Rede Otimizada #2

- Convolucional com 32 *kernels* de 8x8 e *stride* 2x2 com ativação elu;
- *BatchNormalization*
- Convolucional com 64 *kernels* de 4x4 e *stride* 2x2 com ativação elu;
- *BatchNormalization*
- Convolucional com 128 *kernels* de 4x4 e *stride* 1x1 com ativação elu;
- *BatchNormalization*
- Flatten;
- Dense com 3 *outputs*;
- Optimizador Adam com *learning rate* igual a 1e-4;

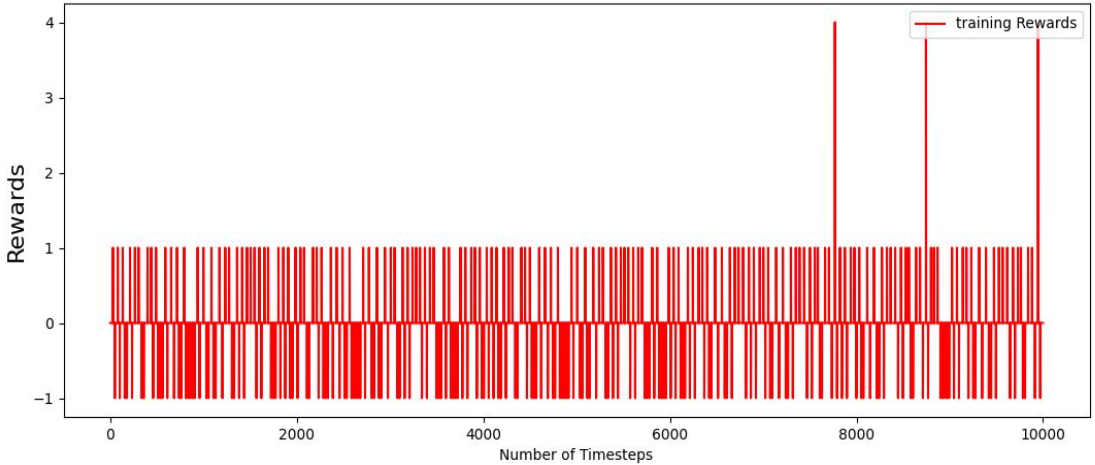
Hiper-Parâmetros

- *Gamma*: 0.85;
- *Observation*: 200;
- *Explore*: 30000;
- Final *Epsilon*: 0.1;
- *Initial Epsilon*: 1;
- *Replay Memory*: 50000;
- *Batch*: 32;
- *Frame per Action*: 1;
- *Learning Rate*: 1e-4;
- *Episodes*: 10000;

Resultados

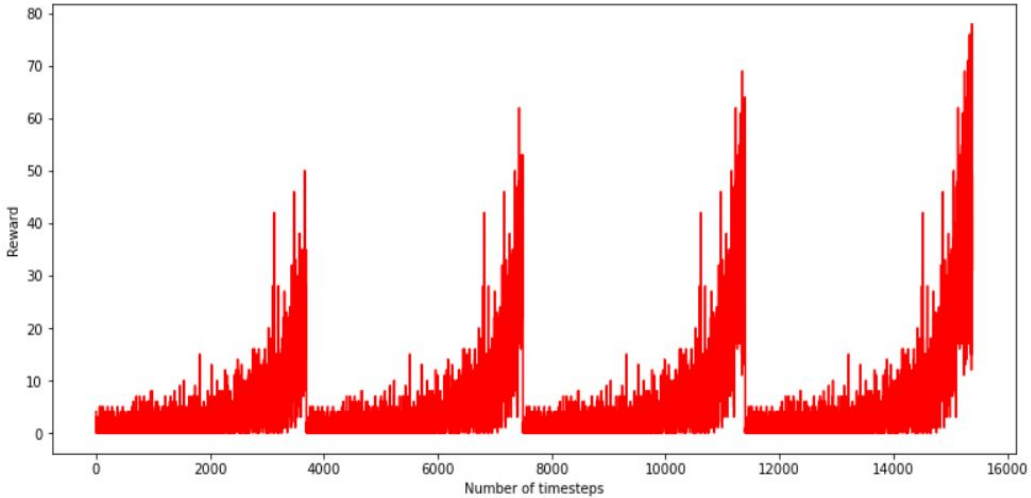
	Rede Inicial	Rede Otimizada #1	Rede Otimizada #2
Score (máximo)	27	64	0
Loss (máximo)	0.011	2 .7 (acumulado)	------
Q_Max (máximo)	4	7.5 (acumulado)	------

Valores de *Reward*

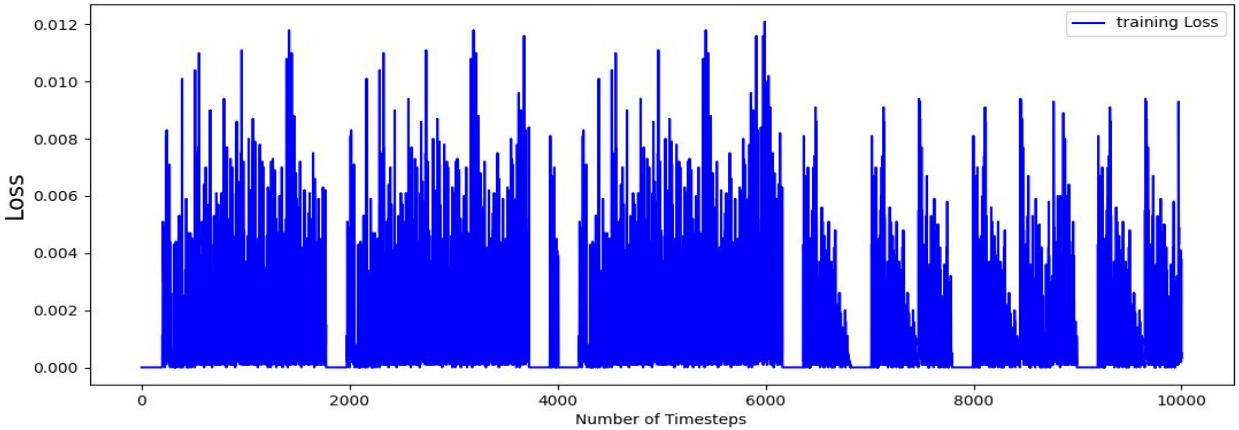


Sem otimização

Com otimização

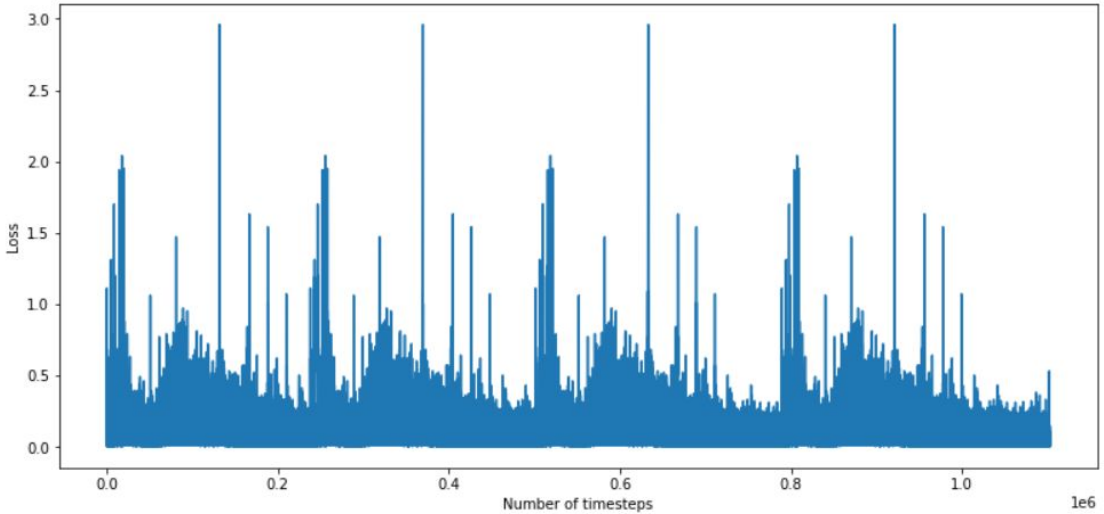


Valores de *Loss*

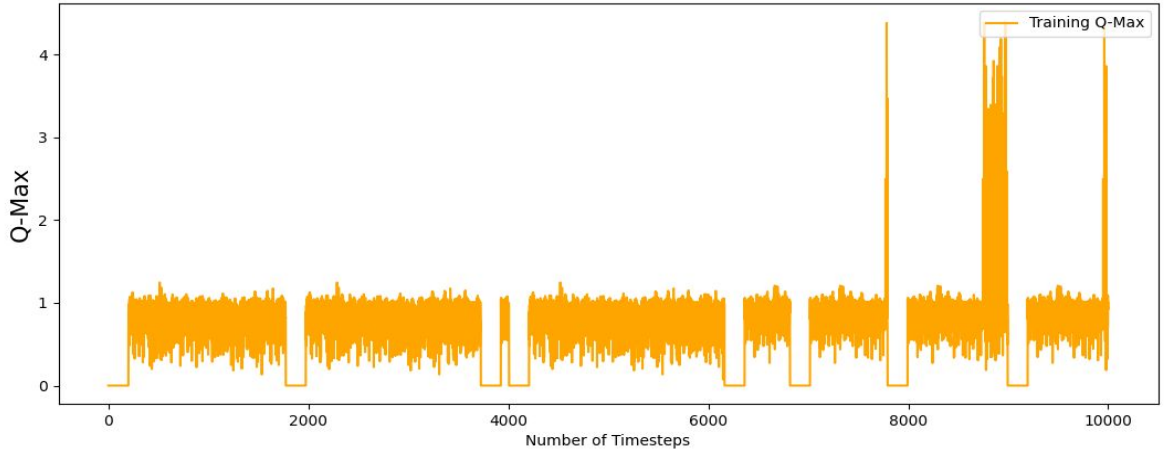


Sem otimização

Com otimização

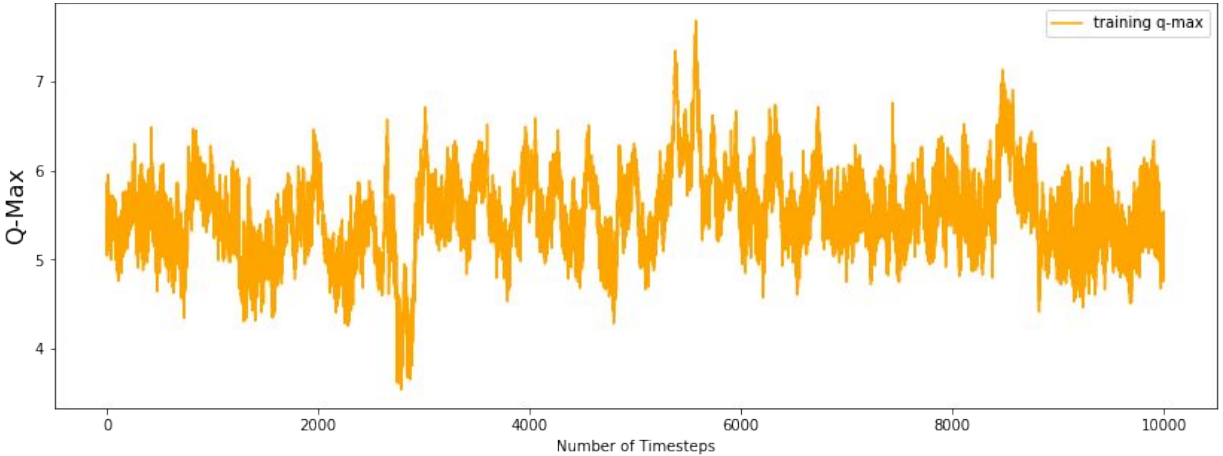


Valores de *Q_Max*

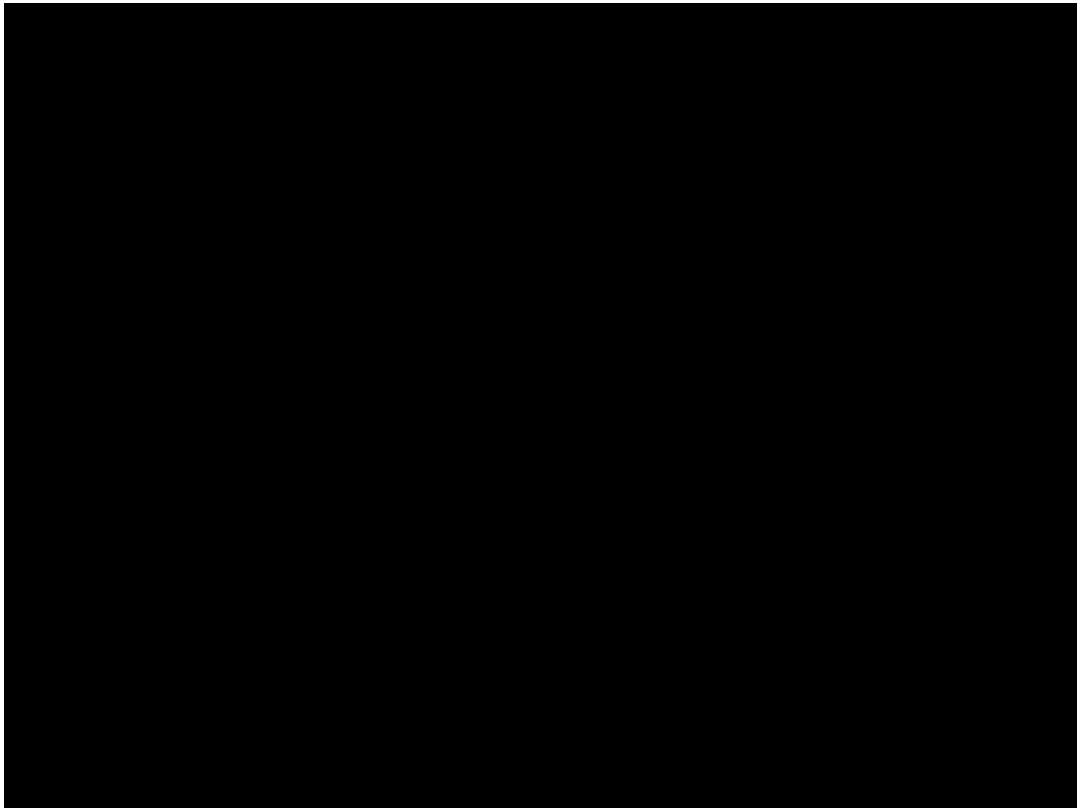


Sem otimização

Com otimização



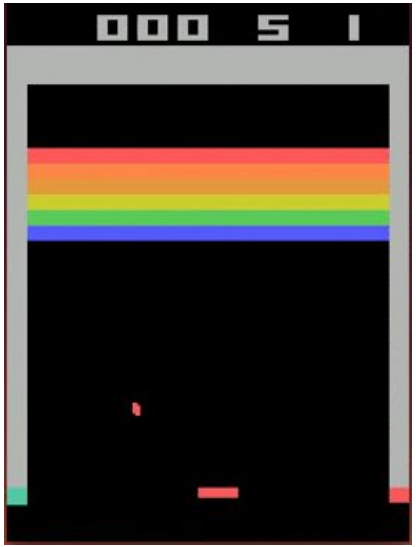
Demo



Conclusões

Resultados: Não foram os ideais

- Limitação de recursos computacionais, recorreu-se ao *Google Colab*;
- Benéfico aproveitar melhor as duas semanas seguintes ao lançamento do projeto, uma vez que seria mais tempo útil para treino;
- *Breakout Deterministic* : *frame skip* de 4 frames a cada *step*. Os modelos não observam tudo o que acontece e tendem a aprendem menos. Solução para este problema seria usar *NoFrameskip*, uma vez que este não avança frames de todo.



Trabalho Futuro:

- Maior exploração dos parâmetros a otimizar e uso de outros algoritmos (exemplo *Actor Critic*);
- Fazer o treino do modelo apenas com 1 vida.

Questões?

Grupo 7
Carolina Marques - PG42818
Constança Elias - PG42820
Maria Barbosa - PG42844
Renata Ribeiro - A86271

Sistemas Inteligentes
Computação Natural