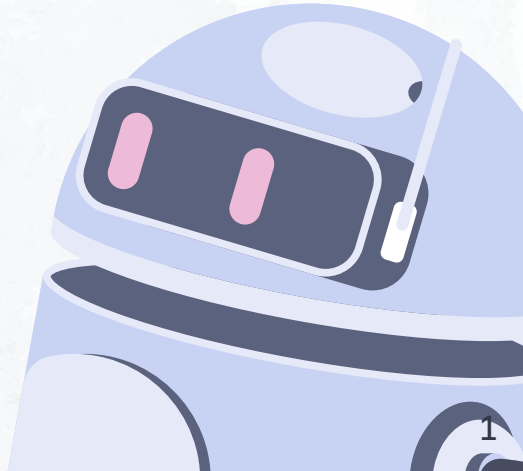
Quantum Reinforcement Learning to Solve Cart Pole Environment

Unidade curricular de ciência de dados quântica
14/06/2023

Maria Gabriela Jordão Oliveira, PG 50599
Miguel Caçador Peixoto, PG 50657



Conteúdos

01 →

Introdução e Motivação

(+) Interesse na Computação Quântica ⟶

Explorando as propriedades intrínsecas da mecânica quântica e com aplicações em diversas áreas.

(+) Aparecimento de paradigmas de ML ⟶

Aparecimento de paradigmas poderosos, tal como o reinforcement learning.

(+) Limitações de poder clássico ⟶

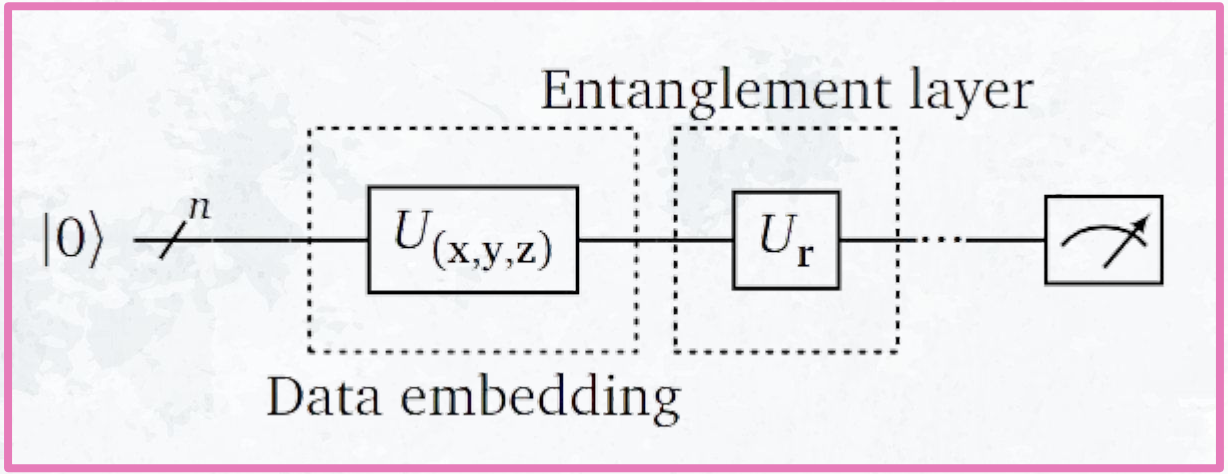
Problemas de escalabilidade e eficiência das máquinas clássicas ao resolver tarefas complexas.

Quantum Reinforcement Learning

02 →

Quantum Machine Learning

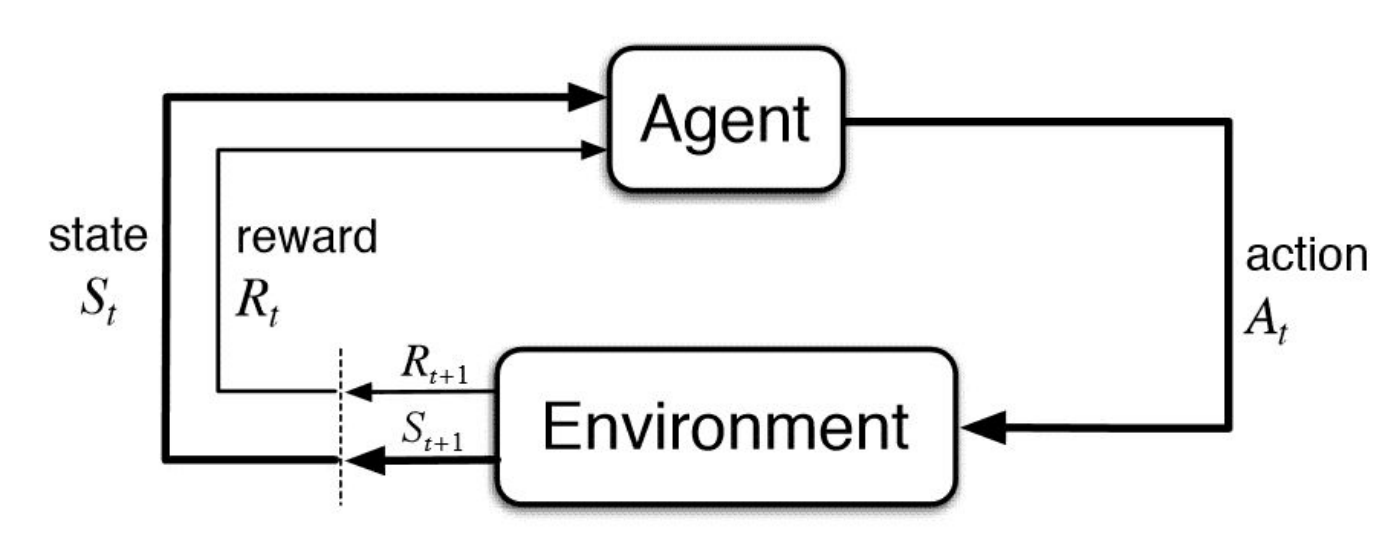
Circuitos Quânticos Variacionais



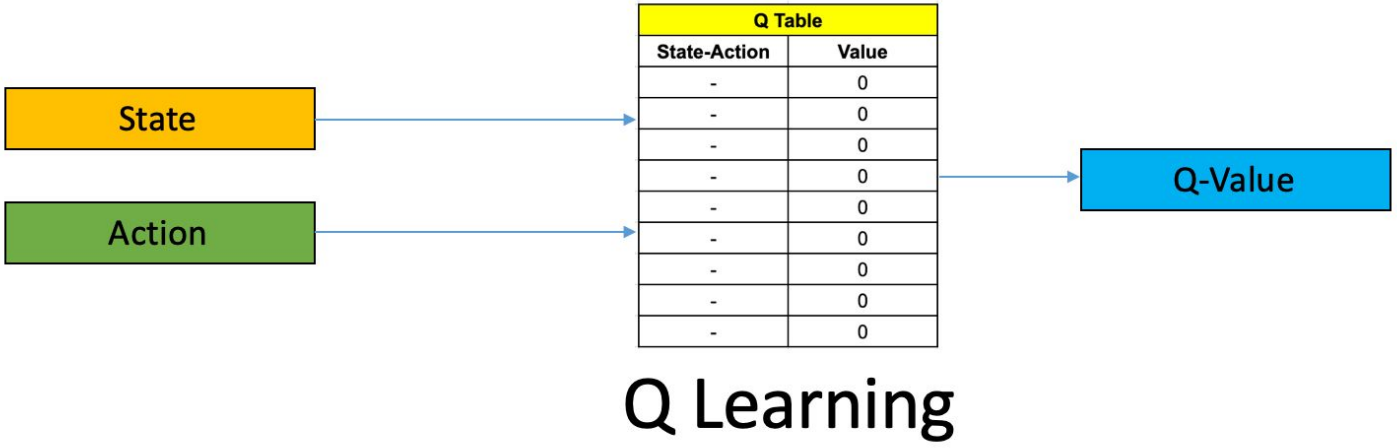
03 →

Reinforcement Learning

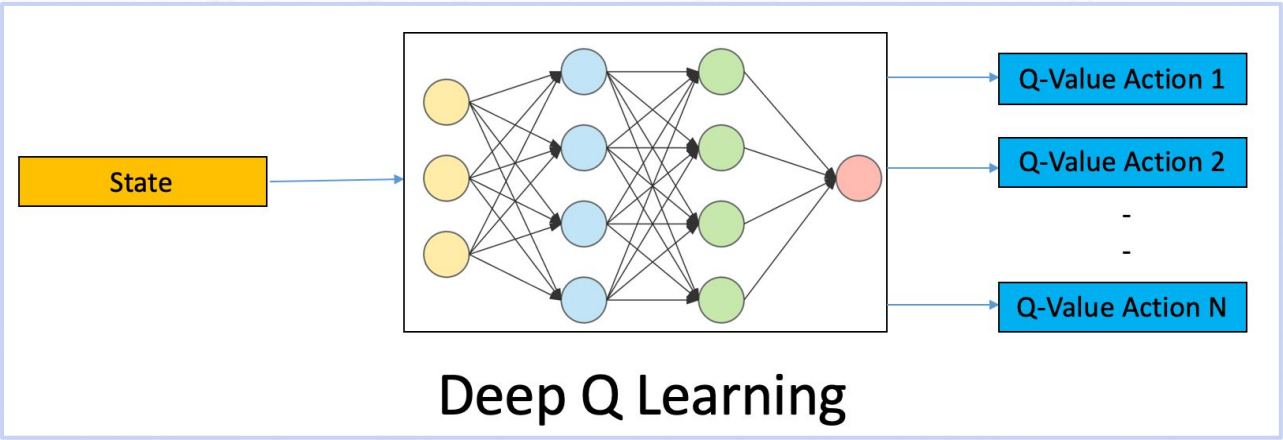
Q - Learning



Q - Learning



Policy Gradient



$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$$

New Q-value estimation

Former Q-value estimation

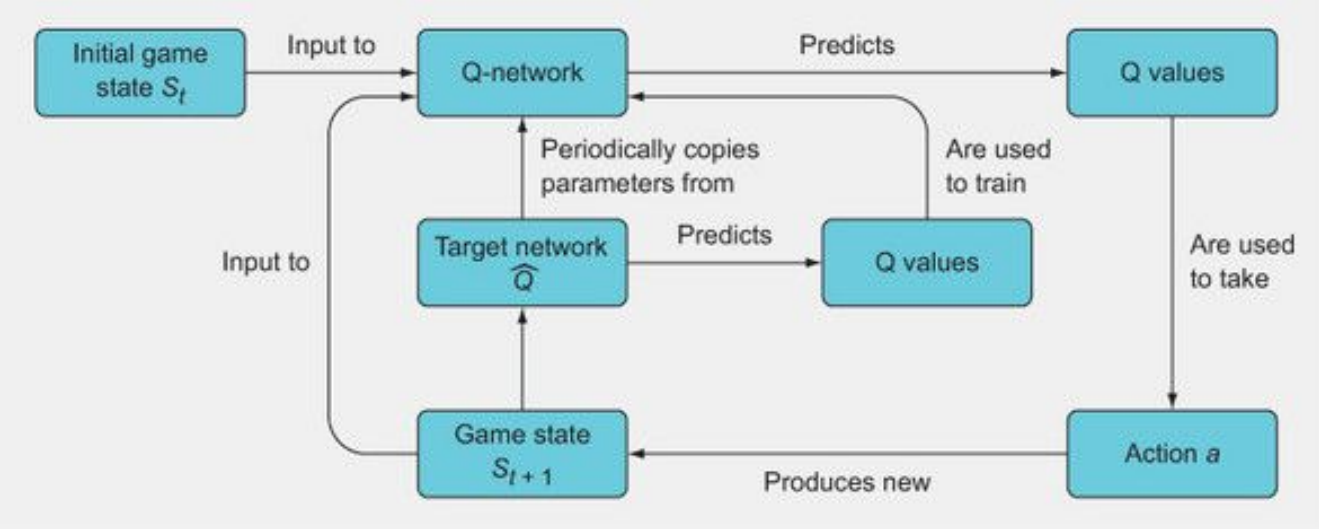
Learning Rate

Immediate Reward

Discounted Estimate optimal Q-value of next state

Former Q-value estimation

Policy Gradient



03 →

Implementação

Ambiente - Cart Pole (v1)

Action Space	Discrete(2)
Observation Shape	(4,)
Observation High	[4.8 inf 0.42 inf]
Observation Low	[-4.8 -inf -0.42 -inf]

Posição e velocidade do carrinho e ângulo e velocidade angular do poste

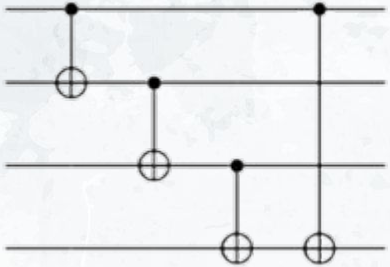
O jogo termina quando o poste ultrapassa um certo ângulo, o carrinho ultrapassa os limites do jogo ou se excede os 500 passos temporais.

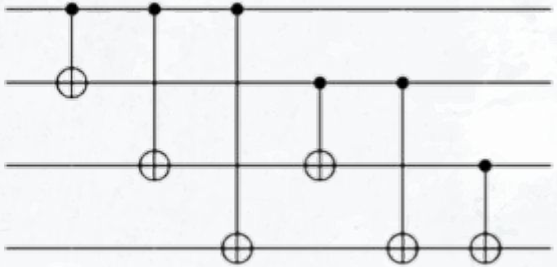
Arquitetura dos Circuitos Quânticos

Embedding dos dados

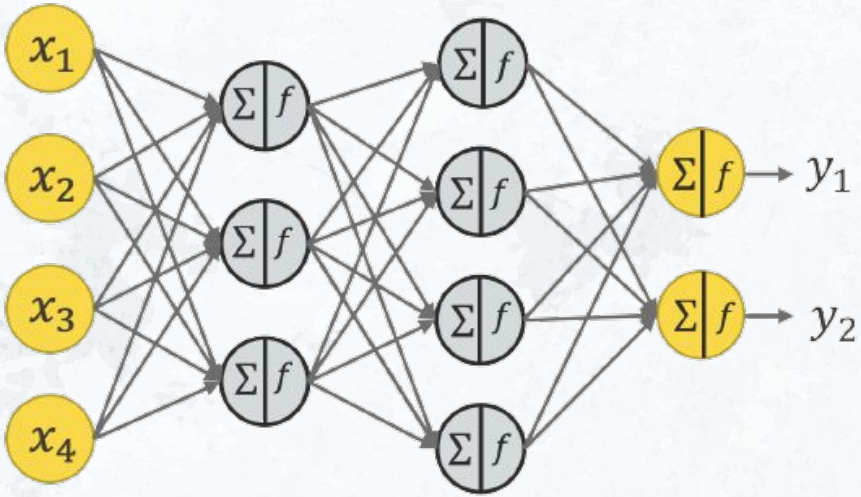
$$|\psi_X\rangle = \bigotimes_{i=1}^{N} R_X(x_i) = \bigotimes_{i=1}^{N} \left[\cos\left(\frac{x_i}{2}\right)|0\rangle - i \sin\left(\frac{x_i}{2}\right)|1\rangle \right]$$

Entrelaçamento





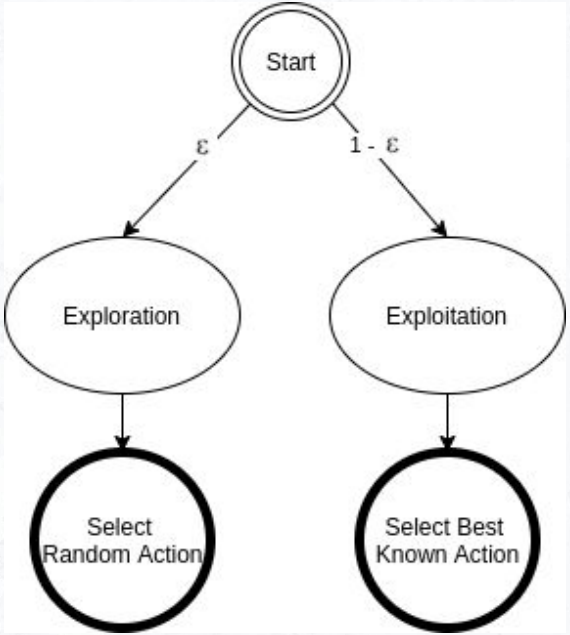
Modelo clássico

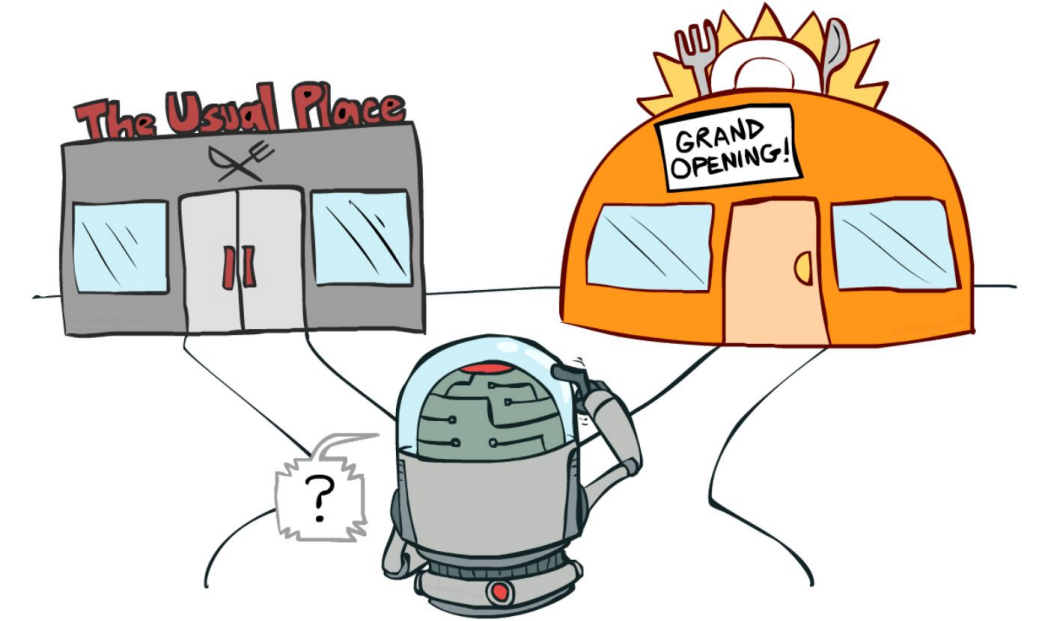


DNN convencional com 10 neurônios por camada escondida, 4 na camada de input e 2 na de saída.

Treino

Q-Learning ✚ Policy Gradient ✚ Epsilon Greedy





Treino

Variable HP	Values
Data Re-Uploading	{0, 1}
Entanglement Type	{CX, CZ}
Entanglement Format	{Ladder, Circular}
Number of Layers	[1, 8]
Fixed HP	**Values**
Batch Size	16
Learning Rate (LR)	0.001
Learning Rate (IO Scaling)	0.01
ϵ_0	1
ϵ_{decay}	0.99
ϵ_{min}	0.01
Buffer Size	0.01
Target Update Frequency (Steps)	5
Online Train Frequency (Steps)	1
Win Threshold (Episodes)	100

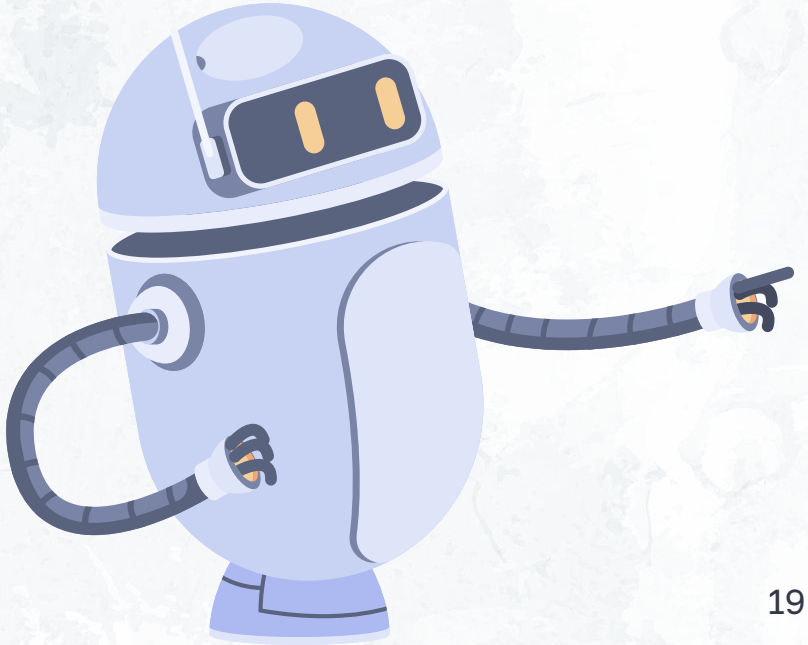
As diferentes arquiteturas são treinadas ao longo de 5000 episódios, com condições de 'early stopping' de ganhar 100 episódios consecutivos.

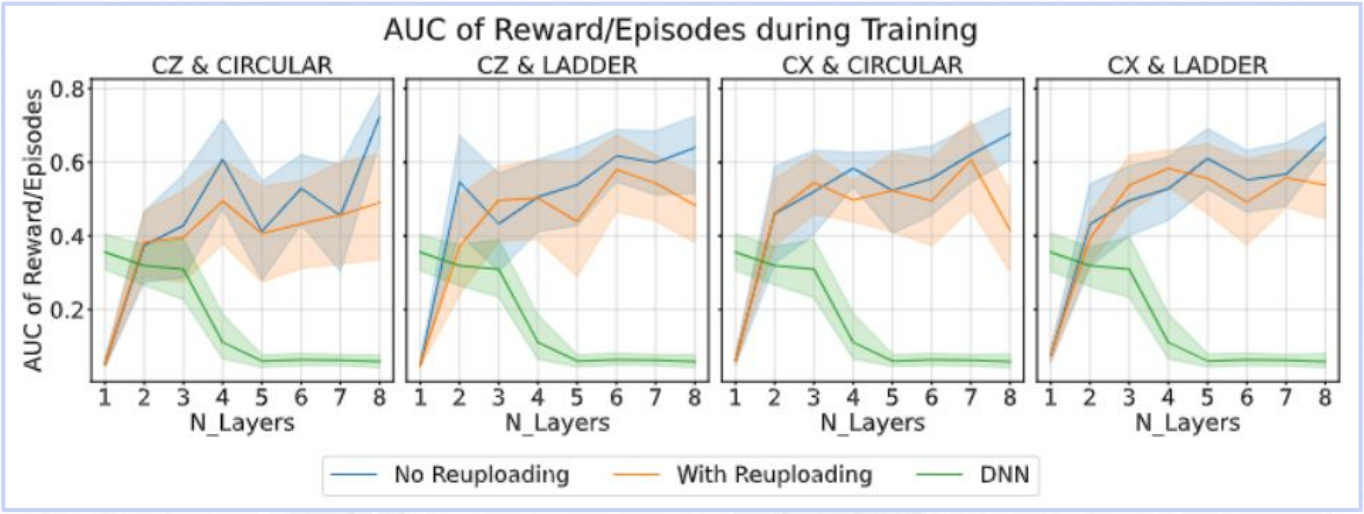
Teste

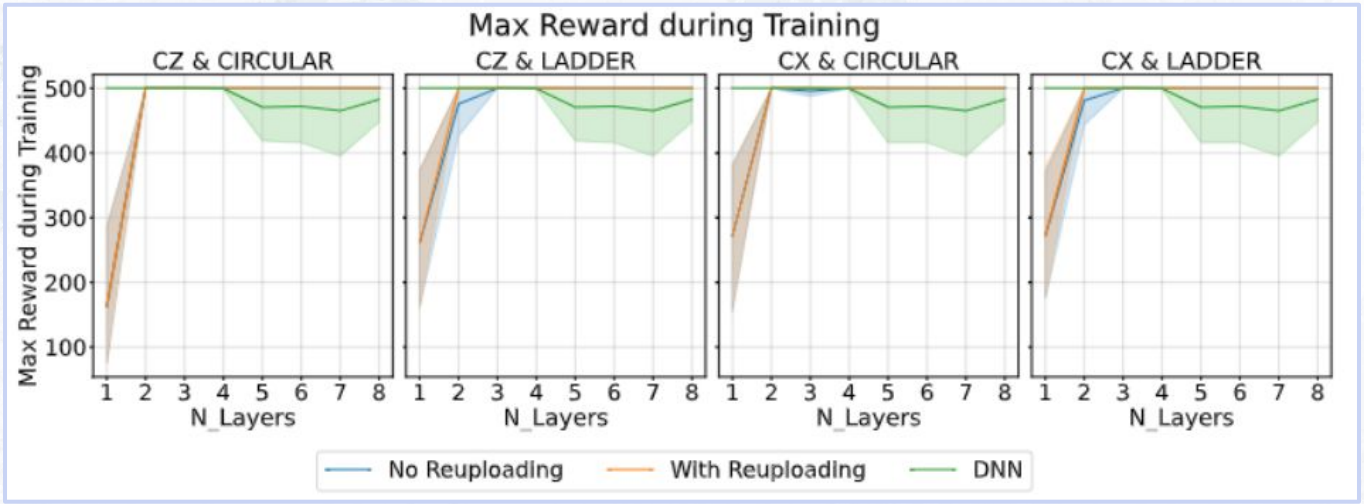
100 jogos tendo em conta os modelos previamente treinados.

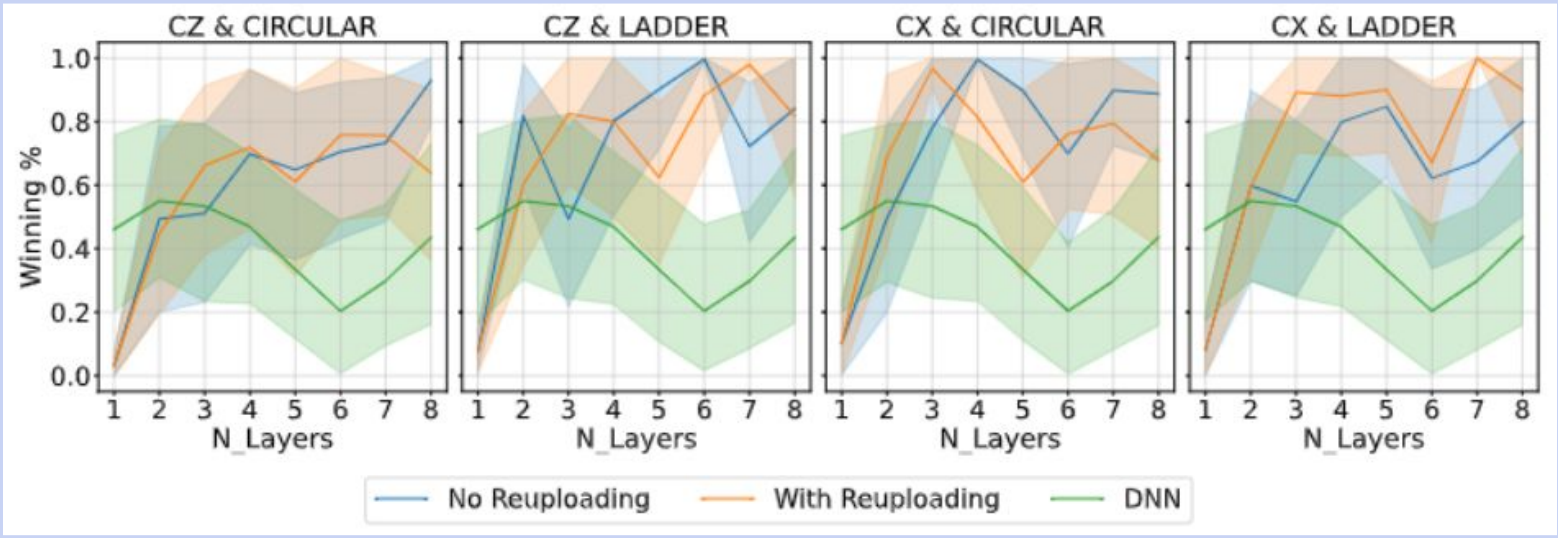
05 →

Resultados




AUC of Reward/Episodes during Training
CZ & CIRCULAR
CZ & LADDER
CX & CIRCULAR
CX & LADDER
AUC of Reward/Episodes
0.8
0.6
0.4
0.2
1 2 3 4 5 6 7 8
N_Layers
No Reuploading
With Reuploading
DNN


Max Reward during Training
CZ & CIRCULAR
CZ & LADDER
CX & CIRCULAR
CX & LADDER
Max Reward during Training
500
400
300
200
100
1 2 3 4 5 6 7 8
N_Layers
No Reuploading
With Reuploading
DNN


CZ & CIRCULAR
CZ & LADDER
CX & CIRCULAR
CX & LADDER
Winning %
1.0
0.8
0.6
0.4
0.2
0.0
1 2 3 4 5 6 7 8
N_Layers
No Reuploading
With Reuploading
DNN

05 →

Conclusão