1. Performance Summary

Each table shows performance of the best 2 models (sorted) for each value of the hyperparameters tested. A lower number of episodes (averaged over 5 trials) it takes for an agent to solve the environment means better performance. A graph is also included to compare performance between the best agents of DQN and REINFORCE.

1.1. Deep Q-Learning

Hidden Layer Size	Activat.	BN	Init.	Avg num of episodes
1	N/A	N/A	N/A	**N/A**
10	tanh	Yes	0.1 unif	**954.2**
	tanh	Yes	Xavier	**1062.4**
20	tanh	No	Default	**946.2**
	tanh	Yes	0.1 const	**1152.4**
80	tanh	No	0.1 const	**430.2**
	tanh	No	0.1 unif	**436.0**
256	tanh	No	Default	**392.8**
	tanh	No	0.1 unif	**447.8**

Table 1: DQN with varying hidden layer sizes. No models with hidden layer size 1 produced a solution, and only 1 model with hidden layer size 10 produced a solution.

Activat.	Hidden Layer Size	BN	Init.	Avg num of episodes
Identity	N/A	N/A	N/A	**N/A**
ReLU	80	No	0.1 unif	**469.0**
	256	No	0.1 unif	**478.2**
LeakyReLU	80	No	Default	**484.0**
	80	No	Xavier	**528.8**
tanh	256	No	Default	**392.8**
	80	No	0.1 const	**430.2**

Table 2: DQN with varying activation functions. Models with identity activation function did not produce a solution.

BN	Activat.	Hidden Layer Size	Init.	Avg num of episodes
No	tanh	256	Default	**392.8**
	tanh	80	0.1 const	**430.2**
Yes	tanh	10	0.1 unif	**954.2**
	tanh	10	Xavier	**1062.4**

Table 3: DQN without vs. with batchnorm

Init.	Hidden Layer Size	BN	Activat.	Avg num of episodes
0.1 const	80	No	tanh	**430.2**
	80	No	ReLU	**506.6**
30 const	256	No	tanh	**2480.0**
	80	No	tanh	**2842.4**
0.1 unif	80	No	tanh	**436.0**
	256	No	tanh	**447.8**
30 unif	256	No	tanh	**666.2**
	80	No	tanh	**1126.2**
Default	256	No	tanh	**392.8**
	80	No	tanh	**461.6**
Xavier	80	No	tanh	**465.4**
	256	No	tanh	**479.6**

Table 4: DQN with varying weight initializations. Only two models with initialization constant of 30 produced a solution.

1.2. REINFORCE Algorithm

Hidden Layer Size	Activat.	BN	Init.	Avg num of episodes
1	tanh	Yes	Default	**356.2**
	tanh	Yes	0.1 unif	**359.0**
10	tanh	No	0.1 const	**178.0**
	Identity	Yes	0.1 unif	**272.4**
20	tanh	No	0.1 unif	**271.2**
	tanh	No	Xavier	**278.4**
80	Identity	No	0.1 const	**330.2**
	LeakyReLU	No	Default	**352.6**
256	ReLU	No	Default	**1254.8**

Table 5: REINFORCE with varying hidden layer sizes. Only 1 model with hidden layer size 256 produced a solution.

Activat.	Hidden Layer Size	BN	Init.	Avg num of episodes
Identity	10	Yes	0.1 unif	**272.4**
	10	No	0.1 const	**285.8**
ReLU	20	No	0.1 const	**406.6**
	10	No	Default	**410.4**
LeakyReLU	10	No	0.1 unif	**336.0**
	20	Yes	0.1 const	**338.0**
tanh	10	No	0.1 const	**178.0**
	20	No	0.1 unif	**271.2**

Table 6: REINFORCE with varying activation functions.

BN	Activat.	Hidden Layer Size	Init.	Avg num of episodes
No	tanh	10	0.1 const	**178.0**
	tanh	20	0.1 unif	**271.2**
Yes	Identity	10	0.1 unif	**272.4**
	LeakyReLU	20	0.1 const	**338.0**

Table 7: REINFORCE without vs. with batchnorm

Init.	Hidden Layer Size	BN	Activat.	Avg num of episodes
0.1 const	10	No	tanh	**178.0**
	10	No	Identity	**285.8**
30 const	1	No	Identity	**2892.0**
	1	Yes	Identity	**3071.4**
0.1 unif	20	No	tanh	**271.2**
	10	Yes	Identity	**272.4**
30 unif	1	Yes	ReLU	**3580.6**
	1	No	ReLU	**3982.8**
Default	20	No	tanh	**281.0**
	80	No	LeakyReLU	**352.6**
Xavier	20	No	tanh	**278.4**
	10	Yes	LeakyReLU	**399.6**

Table 8: REINFORCE with varying weight initializations. Only 2 models with initialization of uniform distribution in range [-30, 30] produced a solution.

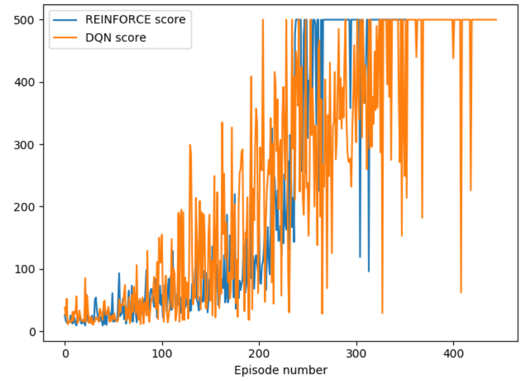

Figure 1: A single run comparison of the best performing agents of DQN & REINFORCE. REINFORCE solved the environment quicker than DQN.

2. Architectures Summary

This section summarizes the architecture of the DQN and REINFORCE agents' neural network models and the hyperparameters used.

2.1. Deep Q-Learning Algorithm

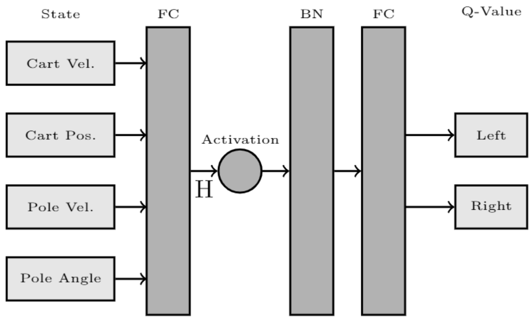

Figure 2: Architecture of Deep Q-Learning Agent Model

The architecture of the Deep Q-Learning Model is shown in Figure 2. It contains a hidden layer of variable size with variable weights initialization that accepts an input size of 4, a variable activation function, variable batch normalization layer, and a final hidden layer with output size of the number of possible actions, which is 2. Each output approximates the quality of the action it corresponds to.

In all DQN models, the following hyperparameters were held constant:

- Batch size 128.
- Replay buffer size = 1000000.
- $\epsilon_{start} = 1$ (notice that this is the ϵ for exploration-exploitation tradeoff and not learning rate).
- $\epsilon_{end} = .1$
- $\epsilon_{decay} = .996$

2.2. REINFORCE Algorithm

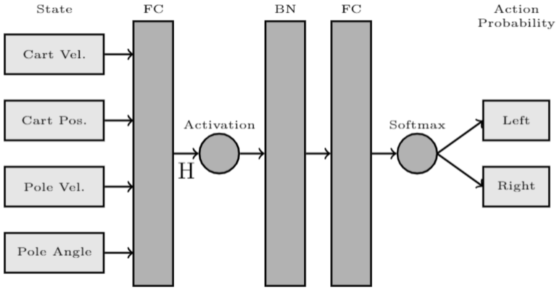

Figure 3: Architecture of REINFORCE Policy Agent Model

The architecture of the REINFORCE Policy model is shown in Figure 3. It is identical to DQN, with a softmax classifier as the output. Each output approximates the probability to choose the action it corresponds to.

3. Experiment Configuration

This section summarizes the configuration of the two experiments performed with the agents.

3.0.1 Experiment 1 Hyperparameters

- Hidden size∈[1, 10, 20, 80, 256].
- Initialization∈[const(0.1), const(30), U[-0.1,0.1], U[-30, 30], Xavier, PyTorch default].
- Activation∈[tanh, ReLU, LeakyReLU, Identity].
- Batch norm active or not.

In experiment 1, the following values of hyperparameters were used for both agents:

- $\gamma = 1$.
- Adam optimizer, with learning rate $\epsilon = 0.01$.
- $L2$ weights normalization, with $\lambda = 0.005$.
- Maximum number of episodes = 5000.
- Random seed=123.

3.0.2 Experiment 2 Hyperparameters

- $\gamma = [0.92, 0.96, 1]$, to confirm our intuition that $\gamma = 1$ is optimal.
- Adam optimizer, with learning rate $\epsilon = [0.5, 0.05, 0.005, 0.0005, 0.01]$.
- $L2$ weights regularization, with $\lambda = [0.5, 0.05, 0.005, 0.0005]$.
- Hidden sizes of $[8, 10, 12, 14, 16]$ (REINFORCE) and $[200, 250, 300]$ (DQN).
- tanh activation.
- No batch normalization.
- const(0.1) initialization for REINFORCE, and PyTorch Default initialization for DQN.

In experiment 2, the best configurations of activation, initialization, and batch norm, from experiment 1 were further optimized by testing different settings for γ, learning rate ϵ, λ, and hidden sizes.