

# Report of training the agent

The model was trained for a maximum of 2,000 episodes but the agent was able to solve the environment (i.e. get atleast +13 average score over 100 adjacent episodes).

## Model

The model is called a Deep Q-Network which was first discovered by the DeepMind team. This uses a Deep Neural Network (or a Multi Layered Perceptron) for estimating the action-value function ( $Q_{\pi}$ ).

The Q-Network has three dense (or fully connected layers). The first two layers have 64 nodes activated with ReLU activation function. The final (output layer) has 4 nodes and is activated with linear activation (or no activation at all). This network takes in as input the 37 dimensional current state and gives as output 4 action-values corresponding to the possible actions that the agent can take.

The neural network used Adam optimizer and Mean Squared Error (MSE) as the loss function.

The following image provides a pictorial representation of the Q-Network model:

Pictorial representation of Q-Network

The following image provides the plot for score v/s episode number:

Plot for score v/s episode number

## Performance

The model was trained on MacBook Air 2017 with 8GB RAM and Intel Core i5 Processor.

- Average time per episode: 1.6s
- Total time for training agent: 892.9s
- Number of episodes required to solve the environment 411 episodes
- Final score of the agent: 13.10

## Hyperparameters used

Hyperparameter	Value	Description
Buffer size	100000	Maximum size of the replay buffer
Batch size	64	Batch size for sampling from replay buffer
Gamma ( $\gamma$ )	0.99	Discount factor for calculating return
Tau ( $\tau$ )	0.001	Hyperparameter for soft update of target parameters
Learning Rate ( $\alpha$ )	0.0005	Learning rate for the neural networks

Hyperparameter	Value	Description
Epsilon ( $\epsilon$ )	1.0	For epsilon-greedy action selection
Epsilon decay rate	0.995	Rate by which epsilon decays after every episode
Epsilon minimum	0.01	The minimum value of epsilon

## Future work

---

The following algorithms can be considered for further development of this agent:

- Double DQN
- Duelling DQN
- Prioritized Experience Replay DQN
- Rainbow