DQN in VizDoom Code Report

Learning algorithm, hyperparameters and network size:

I have used DQN learning algorithm in vizdoom environment (‘defend_the_center’), whereas the network used is keras sequential model with three Conv2D layers having the same activation function, relu, but different parameters (strides and kernel_size) values, then followed by flattened and two dense layers. Where the last dense layer has ‘out_dim’ of action size. The loss function I have used is mean squared error, ‘mse’.

Hyperparameters:

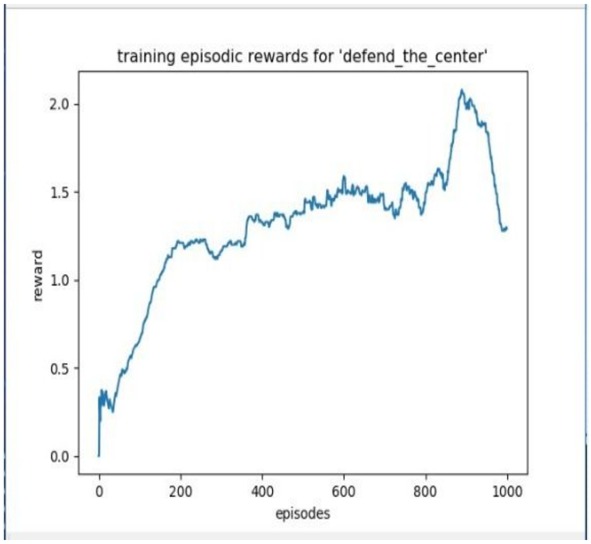
LEARNING_RATE = 0.0001
BATCH_SIZE = 32
EPSILON = 0.05
GAMMA = 0.99(Discount factor)
MAX_EPISODES = 2000(Number of episodes for training)

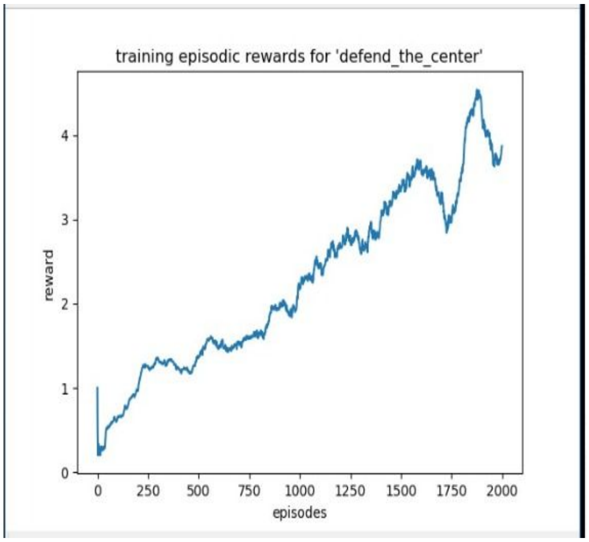
Environment used:

The config file I have used for doom game is defend_the_center.cfg, and hence the action space is size of three different actions TURN_LEFT, TURN_RIGHT, and ATTACK. While the observation is frame of different size image,(in my case it is 160X120). The task is to kill the monster with one shot. And possible rewards are +101 for killing the monster and -5 for missing a shot.

Agent Performance:

The performance of the agent can be measured based on the following plots of rewards in number of episodes:





Conclusion:

In my learning algorithm I have used single model to predict the Q values from the current state(s1) but also from the new state(s2). So training on Q values which are both predicted from same model might sound good, and would try to have another model where I could build a Dueling DQN algorithm to train the network. That way the performance of agent could increase.