

這次作業使用Q learning也就是DQN，模板使用助教的cartpole，增加了一層hidden layer，那神經元個數如下圖所示：

```
self.fc1 = nn.Linear(num_state , 50 )  
self.fc2 = nn.Linear(50 , 50)  
self.out = nn.Linear(50 , num_action )
```

接著是reward shaping的部分，第一種用的是
 $\text{reward} - \text{abs}(y_position) * 2.3 - \text{abs}(x_position) * 1.2$ ，最後成功讓機器人登陸，
這個想法很直覺，就是盡量讓它越靠近landing pad越好。(result1.gif)

```
reward = reward - abs(state[1]) * 2.3 - abs(state[0]) * 1.2
```

接著是reward shaping的部分，第二種用的是
如果 $y_position \geq 1.5$ 的時候，reward就加上 y_speed ，這樣子做是想要讓它在比較高的時候，能夠更快的往下降，加速整個過程，接著就是跟第一種方法很類似的方法，
 $\text{reward} - \text{abs}(y_position) * 2 - \text{abs}(x_position)$ ，最後成功讓機器人登陸，讓它越靠近landing pad越好。(result2.gif)

```
if(state[1] >= 1.5):  
    reward = reward + abs(state[3])  
reward = reward - abs(state[1]) * 2 - abs(state[0])
```

以上兩種shaping方式，感謝助教！