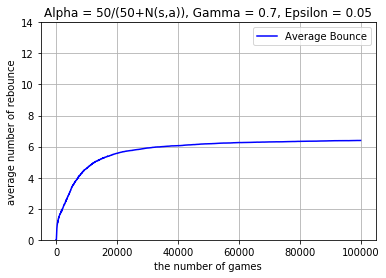
**MP 4.2 Part 1: Q-Learning (Pong)**

**Main Result**

Best result was found when

After training the agent on 100,000 training games:



**Discretization of continuous state**

A state of the pong game can defined by:

Where:

However, we can’t use Q-learning algorithm on a continuous state – there are infinitely many states. So, we discretize the states:

Conversion: Continuous State 🡪 Discrete State

Note: We chose to make if since it is a reasonably negligible speed.

The number of all possible discrete states is 12\*12\*2\*3\*12 +1 =10369.

We found that for larger number of discrete states, a larger number of training game is required to converge to a good policy.

(Number of all Q-values) = 10368\*3 = 31104, since there are 3 possible actions for each state.

**Exploration function**

We chose for good result (justification is discussed later.)

**Choosing an optimal discount rate, :**

We chose

Justification:

Suppose, in and , the ball is bounced. We don’t want to affect calculation of ’s Q value. Assuming that on average, and are separated by 24 steps, the discount factor for is .

|  |  |
| --- | --- |
|  |  |
| 0.9 | 0.0798 = 7.98% |
| 0.8 | 0.00472=0.472% |
| 0.7 | 0.0019 = 0.02% just small enough |
| 0.6 | 0.0000047= too small |

**Choosing an optimal learning rate, :**

Using epsilon = 0.05, gamma=0.7, we manually check C values {15, 50, 100, 500}.

We find that learning rate converges quickest when **C=50**.

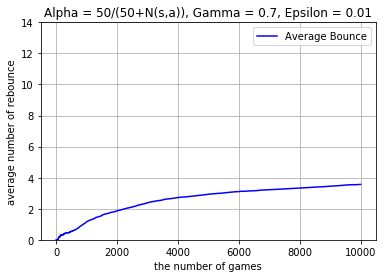
|  |  |
| --- | --- |
| C | number of training game to reach bounce average of 6 |
| 15 | 32108 |
| 50 | 19106 |
| 100 | 35872 |
| 500 | 62311 |
|  |  |

**Choosing an optimal epsilon,**

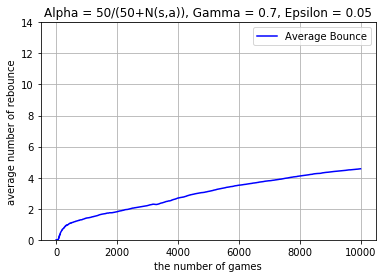
Using gamma = 0.7, C=50, and 10,000 training games, we analyze epsilon values {0.01, 0.05, 0.15, 0.3}. We find that **epsilon= 0.05** gives highest average number of bounces.

In the below plots, the average number of bounces is updated after each training game. (The old average is updated to new average.)

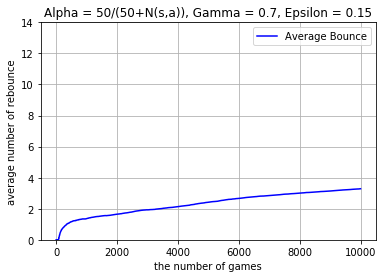
Epsilon =0.01



Epsilon=0.05



Epsilon= 0.15



Epsilon=0.3

