[1]Deep Q Networks

Let us consider how to approximate Q by a function with another argument. To approximate means that there is a target function and an approximating function, and the task is to minimize the difference between them as the error. To minimize the error, we use the gradient descent method with the partial derivative of each variable of the approximate function. This is the same method used in machine learning. Now let's define the target function.

The target function

$$R + \gamma \max_a \hat{q}(S', a, \mathbf{w})$$

, $R$is the immediate reward.

$\max_a \hat{q}(S', a, \mathbf{w})$is an "approximate function" of the most valuable action value function in the following state

current function

$$\hat{q}(S, A, \mathbf{w})$$

$\Delta\mathbf{w}$is the column vector of variables for all approximate functions, and for each variable the error$\times \nabla_{\mathbf{w}} \hat{q}(S, A, \mathbf{w})$ to obtain the magnitude of the gradient of each parameter.

$$\Delta\mathbf{w} = \alpha \left(R + \gamma \max_a \hat{q}(S', a, \mathbf{w}) - \hat{q}(S, A, \mathbf{w}) \right) \nabla_{\mathbf{w}} \hat{q}(S, A, \mathbf{w})$$

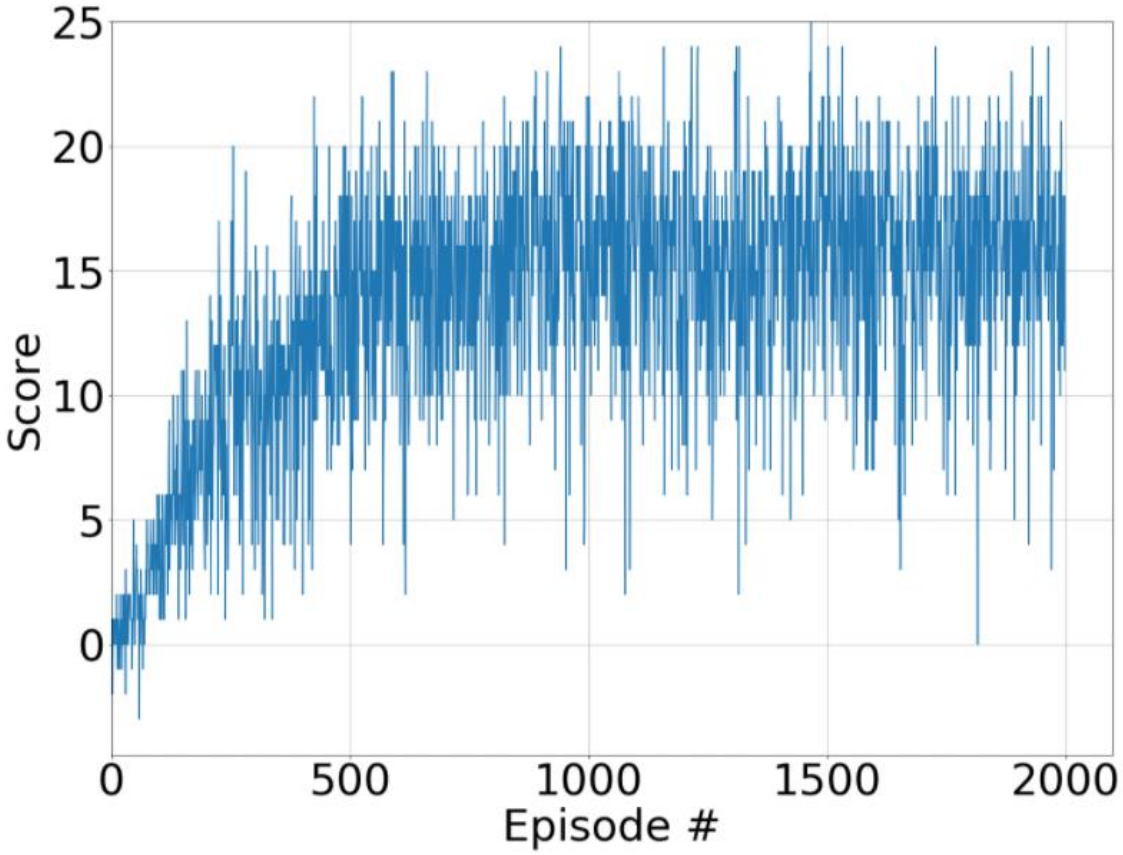
[2]Implement

The experiment was conducted in the environment shown in the README.

The neural network had three layers.

Input layer: 37-Hidden layer: 64-Output layer: 4

2000 episodes



Result: Maximum average 15.78

Discussion: The values were close to each other at about 900.