

Hendro Susanto

1301160284

IF-40-12

Laporan Tugas 3 Machine Learning

A. Analisis Masalah

Terdapat file DataTugas3ML2019.txt yang isinya menggambarkan sebuah *grid world*. Grid world ini memiliki ukuran 15X15 yang setiap cell nya memiliki reward. Akan dipilih sebuah rute terbaik yang memiliki total reward maksimum untuk membantu agent yang berada di posisi start [14][0] (dalam array) untuk menemukan goal yang berada di [0][14] dengan menerapkan metode Q learning dimana agent hanya bisa melakukan 4 aksi yaitu : north, south, west, dan east.

B. Strategi Penyelesaian Masalah

Strategi yang dilakukan untuk membantu agent mencapai goal dengan mendapatkan total reward maksimum adalah menerapkan metode Q-learning pada program dengan menggunakan bahasa pemrograman java.

Langkah-langkahnya sebagai berikut :

- Menginisialisasi nilai north=0, south=1, west=2, dan east=3

```
static int north = 0;
static int south = 1;
static int west = 2;
static int east = 3;
```

- Menentukan nilai parameter learning_rate = 0.1 dan discount_rate = 0.8

```
double learning_rate = 0.1;
double discount_rate = 0.8;
```

- Membuat array state untuk menampung state yang dilalui, array data untuk menampung reward setiap cell pada grid world, dan array Q untuk menentukan setiap aksi pada cell

```
int[] state = new int[29];
String[][] data = new String[15][15];
double[][] Q = new double[225][4];
```

- Melakukan perulangan sampai 1000 episode untuk menentukan aksi terbaik setiap cell yang disimpan pada array Q

```
while (n<1000){
    int a = 14; int b = 0;
    episode(a,b,learning_rate,discount_rate,Q,data);
    System.out.println("episode "+(n+1)+" done");
    n++;
}
```

Setiap aksi cell akan dihitung dengan rumus :

$Q[\text{cell}] = \text{reward_cell} + \text{learning_rate} * (\text{reward_cell_tujuan} + \text{discount_rate} * \max(\text{semua_aksi_cell}) - \text{reward_cell})$

```
Q[(15*a)+b][north] = Double.parseDouble(data[a][b]) +
    learning_rate*(Double.parseDouble(data[a-1][b]) +
    discount_rate*maxQ(Q, ((15*(a-1))+b)) - Double.parseDouble(data[a][b]));
```

Untuk menomori cell pada grid word dibuat rumus 15*baris+kolom. Setiap episode dimulai dari posisi start ke goal.

- Menentukan rute yang harus dilalui sampai goal (goal adalah cell 14) dan menyimpannya ke array state

```
static void nextstate(int a, double[][] Q, int[] state){
    inputstate(a, state);
    if (a!=14){
        int b = indexMax(Q,a);
        if (b==north) a=a-15;
        else a=a+1;
        nextstate(a,Q,state);
    }
}
```

- Mengoutputkan isi array state yang berisi cell-cell dilalui dengan mengubahnya ke dalam bentuk indeks pada grid beserta total reward yang didapatkan.

```
for (int j = 0; j < state.length; j++) {
    System.out.println("indeks : "+(state[j]/15)+" "+(state[j]%15)+" "+
        " -> Reward : "+Integer.parseInt(data[state[j]/15][state[j]%15]));
    total_reward = total_reward + Integer.parseInt(data[state[j]/15][state[j]%15]);
}
System.out.println("
Total Reward : "+total_reward);
```

Output :

```
run:
episode 1 done      episode 997 done
episode 2 done      episode 998 done
episode 3 done      episode 999 done
.....             episode 1000 done
```

```
indeks : [14][0] -> Reward : -2
indeks : [13][0] -> Reward : -2
indeks : [13][1] -> Reward : -1
indeks : [12][1] -> Reward : -2
indeks : [12][2] -> Reward : -1
indeks : [12][3] -> Reward : -2
indeks : [12][4] -> Reward : -2
indeks : [12][5] -> Reward : -2
indeks : [11][5] -> Reward : -3
indeks : [11][6] -> Reward : -3
indeks : [11][7] -> Reward : -3
indeks : [11][8] -> Reward : -4
indeks : [11][9] -> Reward : -1
indeks : [10][9] -> Reward : -1
indeks : [9][9] -> Reward : -2
indeks : [9][10] -> Reward : -3
indeks : [8][10] -> Reward : -2
indeks : [7][10] -> Reward : -1
indeks : [6][10] -> Reward : -1
indeks : [5][10] -> Reward : -1
indeks : [4][10] -> Reward : -1
indeks : [4][11] -> Reward : -3
indeks : [3][11] -> Reward : -1
indeks : [2][11] -> Reward : -1
indeks : [1][11] -> Reward : -2
indeks : [1][12] -> Reward : -2
indeks : [1][13] -> Reward : -2
indeks : [1][14] -> Reward : -1
indeks : [0][14] -> Reward : 500
----- +
Total Reward : 448
BUILD SUCCESSFUL (total time: 3 seconds)
```

Rute yang dilalui

