

Univerzitet u Tuzli
Fakultet elektrotehnike
Predmet: Inteligentni sistemi
Akademska godina: 2021./2022.
Student: Indir Karabegović
Datum: 13.06.2022. god.

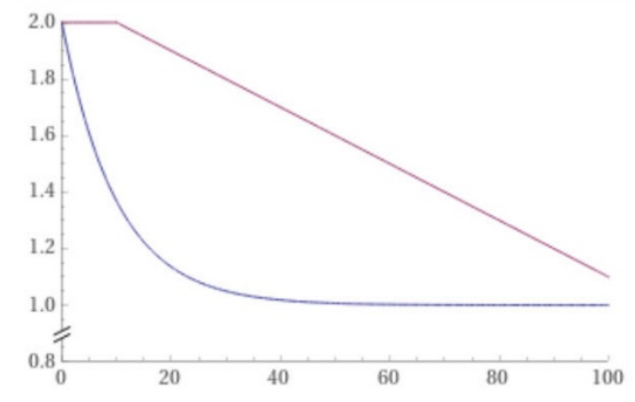
Izvještaj zadaje broj 4

Zadatak br. 1 (GL - Genetski algoritmi)

U fabrici je potrebno odrediti maksimalan profit pri proizvodnji određenog proizvoda. Najskuplje je proizvoditi male serije, tako da je proizvodna cijena proizvoda data formulom:

$$y = 1 + e^{\frac{-x}{m}}$$

gdje je x iznos dnevne proizvodnje (maksimalno n dnevno) (plava linija) Ujedno, prodajna cijena nije fiksna. Proizvođač mora dati rabate za veću količinu. Rabat je 0% za prvih k proizvoda, a nakon toga linearno pada do maksimalno 1% (ljubičasta linija). Dijagram je prikazan na narednoj slici:



Odabrati proizvoljno:

- Broj proizvoda n (u intervalu 100-1000)
- Strminu eksponencijalne funkcije preko parametra m (u intervalu 200-500)
- Broj proizvoda bez rabata (k) (u intervalu 10-300)
- Maksimalan rabat (l) (u intervalu 0-50%)

Nakon toga napisati *fitnes funkciju* koja omogućava izračun maksimalne razlike između troškova proizvodnje i prodajne cijene i pokrenuti genetski algoritam. Odrediti broj proizvoda s maksimalnim profitom x i iznos profita y (konkretno za primjer sa slike bi bio približno 22 proizvoda i profit je nešto ispod 0,8).

Rješenje:

Prvo ćemo prikazati matlab kod za ovaj zadatak:

```
import numpy as np
from matplotlib import pyplot as plt
from geneticalgorithm import geneticalgorithm as ga

m = 50
n = 100
k = 10
x = np.arange(0.0, k, 0.1)
a = np.linspace(10, n, n)
t = np.linspace(0, n, n)

def y(x):
    return 1+np.exp(-t/m)

def f(x):
    return -0.01*a+2.1

def g(x):
    return 2

def h(x):
    return g(x)+f(x)-y(x)

plt.plot(x, np.full(x.shape, g(x)), color='red')
plt.plot(a, f(x), color='red')
plt.plot(t, y(x), color='blue')
plt.show()
plt.plot(t, h(x), color='green')
plt.show()

def q(X):
    dim = len(X)
    OF = 0
    for i in range(0, dim):
        #OF += (1+np.exp(-X[i]/m)+0.01*X[i]-2.1-2)
        OF += (2-0.01*X[i]+2.1-1-np.exp(-X[i]/m))
    return -OF
```

```

varbound = np.array([[0, 98]])

algorithm_param = {'max_num_iteration': 250,
                   'population_size': 200,
                   'mutation_probability': 0.05,
                   'elit_ratio': 0.05,
                   'crossover_probability': 0.6,
                   'parents_portion': 0.3,
                   'crossover_type': 'uniform',
                   'max_iteration_without_improv': None}

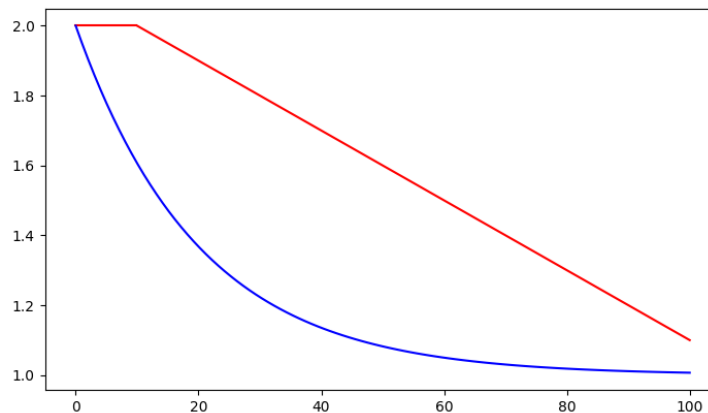
model = ga(function=q, dimension=1, variable_type='real',
           variable_boundaries=varbound, algorithm_parameters=algorithm_param)

model.run()

# convergence = model.report
# #solution = model.ouput_dict

```

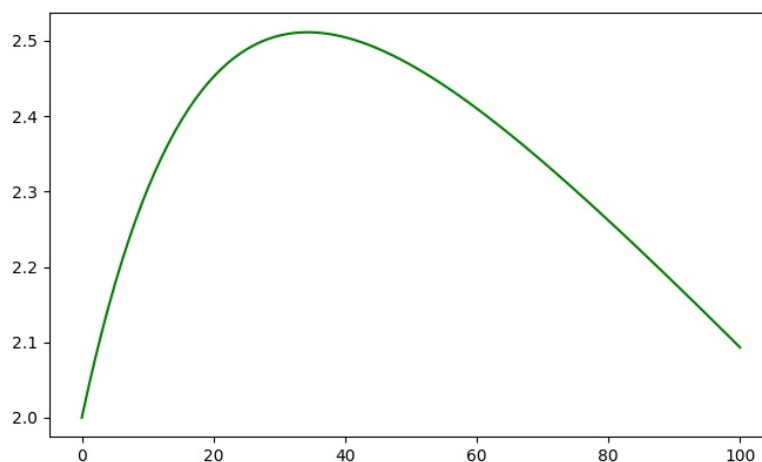
U ovom zadatku prvo ćemo plotati funkcije koje se odnose na cijenu proizvodnje i zaradu:



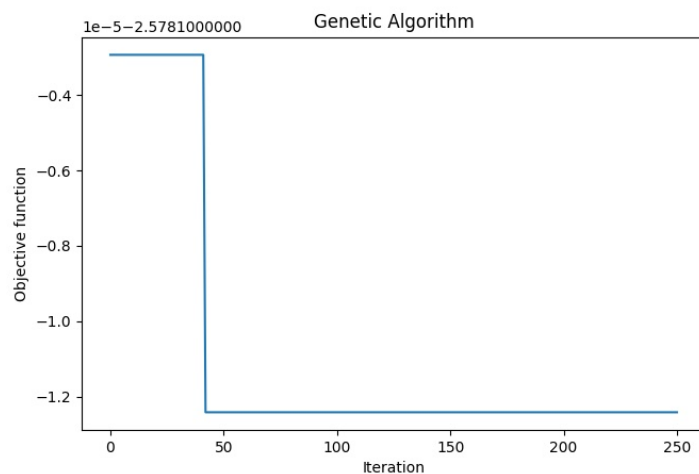
Parametri koje smo uzeli u ovom zadatku su:

- Broj proizvoda ($n = 100$)
- Strmina eksponencijalne funkcije ($m = 20$)
- Broj proizvoda bez rabata ($k = 10$)

Na prethodnom dijagramu plava funkcija predstavlja cijenu proizvodnje dok crvena funkcija predstavlja cijenu na osnovu koje je proizvod prodan (prodajna cijena). Kada smo dobili ove dvije funkcije, od funkcije zarade ćemo oduzeti funkciju cijene proizvodnje, te ćemo na navedenu funkciju primjeniti genetski algoritam. Kada oduzmemo dvije navedene funkcije, dobijamo sljedeću funkciju:



Nakon toga, potrebno je primijeniti genetski algoritam (s vježbi) na navedenu funkciju, te tražiti maksimum ove funkcije. Postavke genetskog algoritma kao i funkcija su date u matlab kodu iznad. Također, za razliku od slučaja s vježbi, moramo voditi računa o dimenzionalnosti. U ovom slučaju smo u genetski algoritam prosljedili dimenziju 1, te smo kao rješenje dobili jednu tačku na x osi. Pokretanjem prethodnog matlab koda smo dobili sljedeći dijagram (vezan za GA):



Tačka u kojoj smo dosegli maksimum, kao i vrijednost funkcije u toj tački su prikazani u narednom text box-u (bitno je napomenuti da smo dobili vrijednost sa negativnim predznakom iako je vrijednost funkcije u dobijenoj tački pozitivna i maksimalna, što možemo vidjeti i sa prethodnog grafika – grafik gdje je funkcija obojena zelenom bojom) :

The best solution found:

[32.18774846]

Objective function:

-2.5781124172582586

Možemo vidjeti da je u ovom slučaju optimalan broj proizvoda $x = 32$, a maksimalna zarada je $y = 2.57811$.

Zadatak br. 2 (RL – Učenje s podrškom)

Problem s labirintom sa prethodnih vježbi proširiti na sljedeći način:

- Polje postaviti na dimenzije 12x12
- Bilo gdje u labirintu postaviti jednu veliku otvorenu „prostoriju“ dimenzija 5x5 bez prepreka, od koje vodi uski hodnik (bez raskrsnica) prema cilju, minimalne dužine 5 polja (ne mora biti ravan hodnik, može imati L oblik)
- Trenirati izlazak iz labirinta - metoda reinforcement learninga je klasična Q-tabela
- Simulirati 4 kombinacije alpha parametra (learning rate) i gamma parametra (discount rate). Analizirati rezultate i napisati komentar kakav efekt se dobija s promjenom ovih parametara.
- Simulirati 3 varijante različitih kazni i nagrada i prikazati rezultate u tabeli

Ne treba prilagati kompletan kod u word fajlu, nego samo navesti u kojim linijama originalnog koda su izvršene izmjene.

Rješenje:

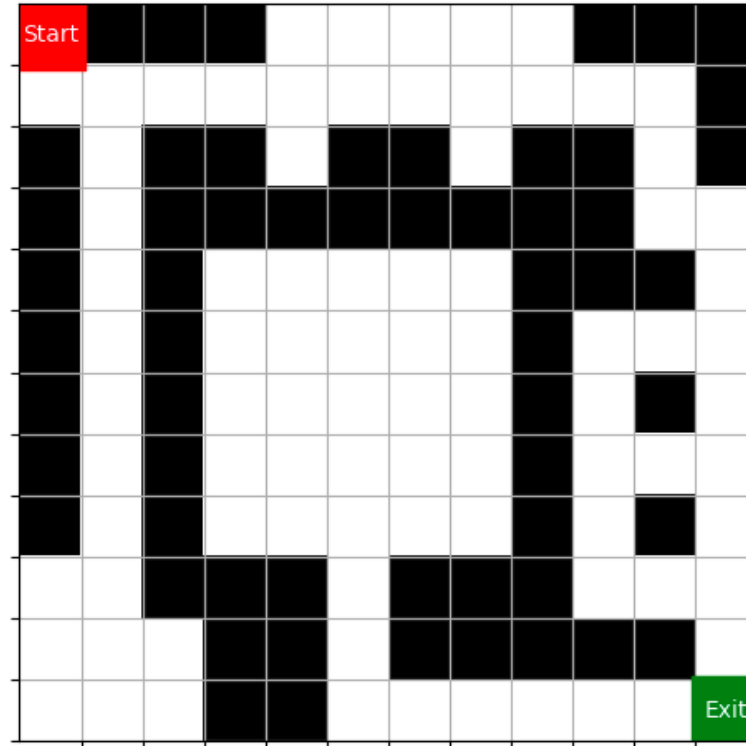
Prvo što smo morali uraditi u ovom zadatku jeste modifikacija labirinta kako bismo dobili labirint sa specifikacijama koje su tražene u zadatku. Također, morali smo postaviti metodu reinforcement learninga kao klasičnu Q tabelu. Kod za navedene postavke je prikazan u narednom frame-u:

```
# show maze ->
# test = Test.SHOW_MAZE_ONLY

test = Test.Q_LEARNING # first change
# test = Test.SARSA_ELIGIBILITY # which test to run

maze = np.array([
[0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1],
[1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0],
[1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0],
[1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0],
[1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0],
[1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0],
[1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0],
[0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0],
[0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0],
[0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0]
]) # 0 = free, 1 = occupied
```

Kada smo pokrenuli kod (pri čemu smo postavili test kao `Test.SHOW_MAZE_ONLY`) dobili smo sljedeći izgled labirinta:



Nakon što smo kreirali odgovarajući labirint i izabrali zadanu metodu učenja, potrebno je da simuliramo 4 slučaja učenja mijenjanjem parametara *learning rate* i *discount rate*. Navedene parametre mijanjamo sljedećim matlab kodom:

```
if test == Test.Q_LEARNING:
    game.render(Render.TRAINING)
    model = models.QTableModel(game)
    h, w, _, _ = model.train(discount=0.90, exploration_rate=0.10, learning_rate=0.10, episodes=200,
                             stop_at_convergence=True)
```

1. Prvi slučaj je sa sljedećim parametrima: ***discount = 0.90, learning_rate=0.10, episodes = 200***

U svakom od navedenih primjera presjek ćemo praviti na sljedećim epizodama (10, 50, 100, 150 itd.) Prvi presjek pravimo nakon 10 epizoda te možemo vidjeti sljedeći rezultat:

```
INFO : 2022-06-12 15:20:56: episode: 10/200 | status: LOSE | e: 0.09559
INFO : 2022-06-12 15:20:57: won: 11 | lost: 74 | win rate: 0.12941
```

Vidimo da smo nakon 10 epizoda imali izuzetno loš rezultat, odnosno, u 11 slučajeva je ostvarena pobjeda, dok je u 74 slučajeva ishod bio poraz. Win rate je iznosio: 0.12941.

INFO : 2022-06-12 15:27:46: episode: 50/200 | status: LOSE | e: 0.07822

INFO : 2022-06-12 15:27:47: won: 25 | lost: 60 | win rate: 0.29412

Nakon 50 epizoda, rezultat se neznatno popravio, te je u 25 slučajeva ostvarena pobjeda, dok je u 60 slučajeva ostvaren poraz. Win rate je iznosio: 0.29412 .

INFO : 2022-06-12 15:31:55: episode: 100/200 | status: WIN | e: 0.06088

INFO : 2022-06-12 15:31:56: won: 64 | lost: 21 | win rate: 0.75294

Nakon 100 epizoda, rezultat je mnogo bolji, te imamo 64 pobjede i 21 poraz. Win rate je iznosio: 0.75294.

INFO : 2022-06-12 15:34:28: episode: 150/200 | status: WIN | e: 0.04738

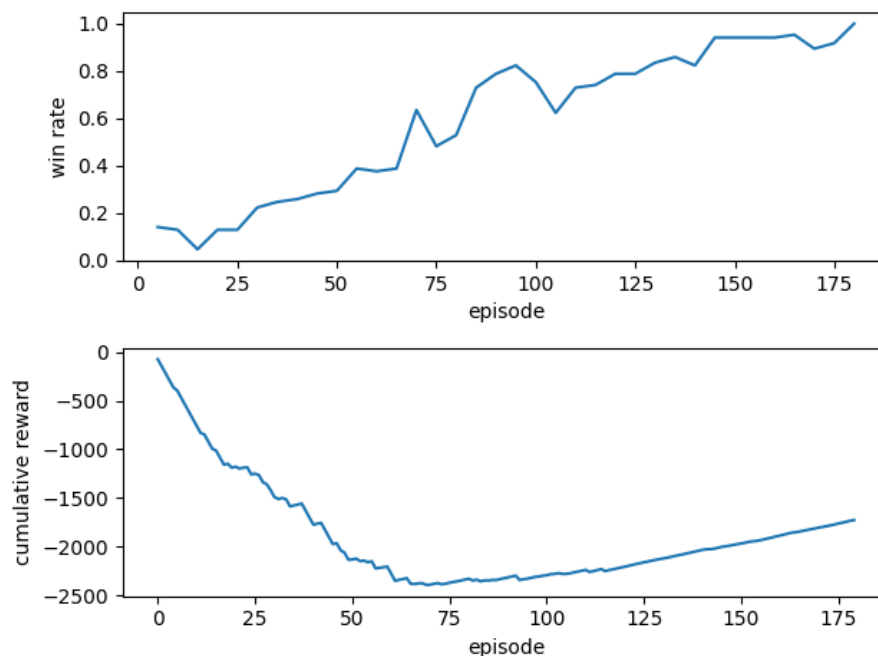
INFO : 2022-06-12 15:34:28: won: 80 | lost: 5 | win rate: 0.94118

Poslije 150 epizoda, rezultat je izuzetno dobar, te imamo 80 pobjeda i 5 poraza. Win rate je iznosio: 0.94118.

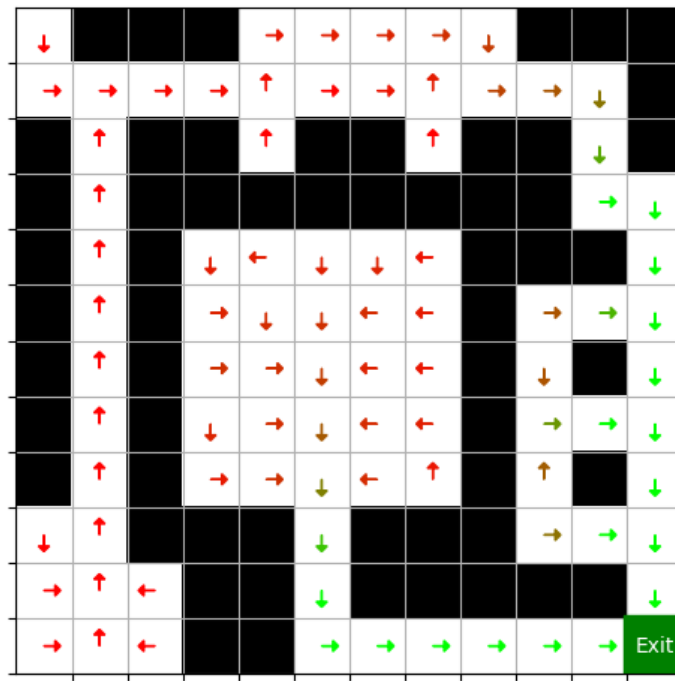
INFO : 2022-06-12 15:35:48: episode: 180/200 | status: WIN | e: 0.04077

INFO : 2022-06-12 15:35:48: won: 85 | lost: 0 | win rate: 1.00000

Vidimo da smo nakon 180 epizoda, dostigli Win rate od 1.00000 te imamo 85 pobjeda i 0 poraza. Dijagrami koje smo dobili po izvršenju ovog zadatka su prikazani na narednoj slici:



Sa slike možemo vidjeti kako se Win rate mijenjao tokom prolaska kroz epizode, te vidimo da je nakon 180 epizoda ovaj parametar dosegao maksimalnu vrijednost. Labirint je na kraju poprimio sljedeći oblik:



2. Drugi slučaj je sa sljedećim parametrima: ***discount = 0.70, learning_rate=0.10, episodes = 200***

U svakom od navedenih primjera presjek ćemo praviti na sljedećim epizodama (10, 50, 100, 150 itd.) Prvi presjek pravimo nakon 10 epizoda te možemo vidjeti sljedeći rezultat:

INFO : 2022-06-12 16:58:46: episode: 10/200 | status: LOSE | e: 0.09559

INFO : 2022-06-12 16:58:47: won: 3 | lost: 82 | win rate: 0.03529

Vidimo da smo nakon 10 epizoda imali izuzetno loš rezultat, odnosno, u 3 slučaja je ostvarena pobjeda, dok je u 82 slučaja ishod bio poraz. Win rate je iznosio: 0.03529.

INFO : 2022-06-12 17:12:27: episode: 50/200 | status: WIN | e: 0.07822

INFO : 2022-06-12 17:12:28: won: 19 | lost: 66 | win rate: 0.22353

Nakon 50 epizoda, rezultat se neznatno popravio, te je u 19 slučajeva ostvarena pobjeda, dok je u 66 slučaja ostvaren poraz. Win rate je iznosio: 0.22353 .

INFO : 2022-06-12 17:17:29: episode: 100/200 | status: WIN | e: 0.06088

INFO : 2022-06-12 17:17:29: won: 67 | lost: 18 | win rate: 0.78824

Nakon 100 epizoda, rezultat je mnogo bolji, te imamo 67 pobjeda i 18 poraza. Win rate je iznosio: 0.78824.

INFO : 2022-06-12 17:21:24: episode: 150/200 | status: WIN | e: 0.04738

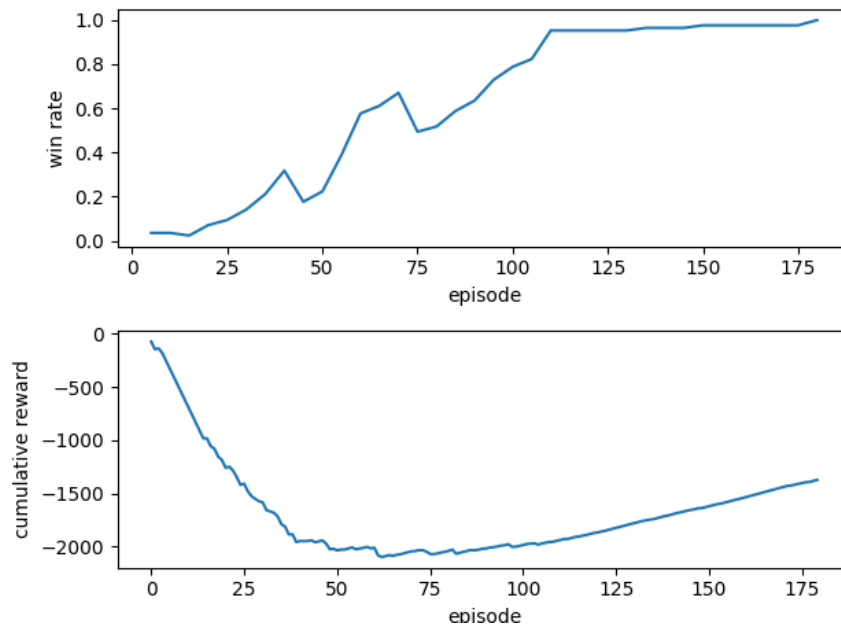
INFO : 2022-06-12 17:21:25: won: 83 | lost: 2 | win rate: 0.97647

Poslije 150 epizoda, rezultat je izuzetno dobar, te imamo 83 pobjede i 2 poraza. Win rate je iznosio: 0.97647.

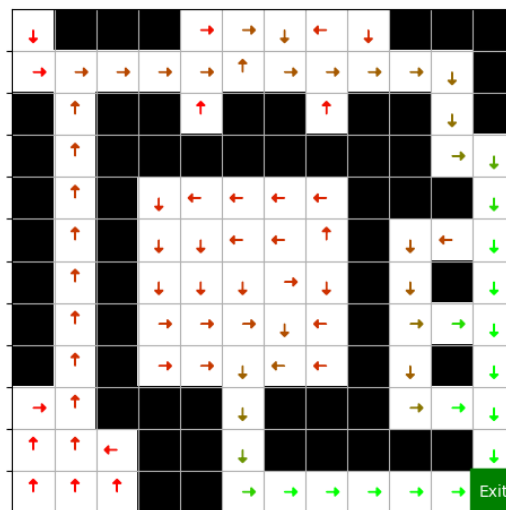
INFO : 2022-06-12 17:23:20: episode: 180/200 | status: WIN | e: 0.04077

INFO : 2022-06-12 17:23:20: won: 85 | lost: 0 | win rate: 1.00000

Vidimo da smo nakon 180 epizoda, dostigli Win rate od 1.00000 te imamo 85 pobjeda i 0 poraza. Dijagrami koje smo dobili po izvršenju ovog zadatka su prikazani na narednoj slici:



Sa slike možemo vidjeti kako se Win rate mijenjao tokom prolaska kroz epizode, te vidimo da je nakon 180 epizoda ovaj parametar dosegao maksimalnu vrijednost. Labirint je na kraju poprimio sljedeći oblik:



3. Treći slučaj je sa sljedećim parametrima: **discount = 0.60, learning_rate=0.30, episodes = 200**

U svakom od navedenih primjera presjek ćemo praviti na sljedećim epizodama (10, 50, 100, 150 itd.) Prvi presjek pravimo nakon 10 epizoda te možemo vidjeti sljedeći rezultat:

```
INFO : 2022-06-12 17:35:38: episode: 10/200 | status: WIN | e: 0.09559
```

```
INFO : 2022-06-12 17:35:39: won: 4 | lost: 81 | win rate: 0.04706
```

Vidimo da smo nakon 10 epizoda imali izuzetno loš rezultat, odnosno, imali smo samo 4 pobjede, dok je u 81 slučaju ishod bio poraz. Win rate je iznosio: 0.04706.

```
INFO : 2022-06-12 17:44:11: episode: 50/200 | status: WIN | e: 0.07822
```

```
INFO : 2022-06-12 17:44:11: won: 61 | lost: 24 | win rate: 0.71765
```

Nakon 50 epizoda, rezultat se znatno popravio, te je u 61 slučaju ostvarena pobjeda, dok je u 24 slučaja ostvaren poraz. Win rate je iznosio: 0.71765 .

```
INFO : 2022-06-12 17:47:15: episode: 100/200 | status: WIN | e: 0.06088
```

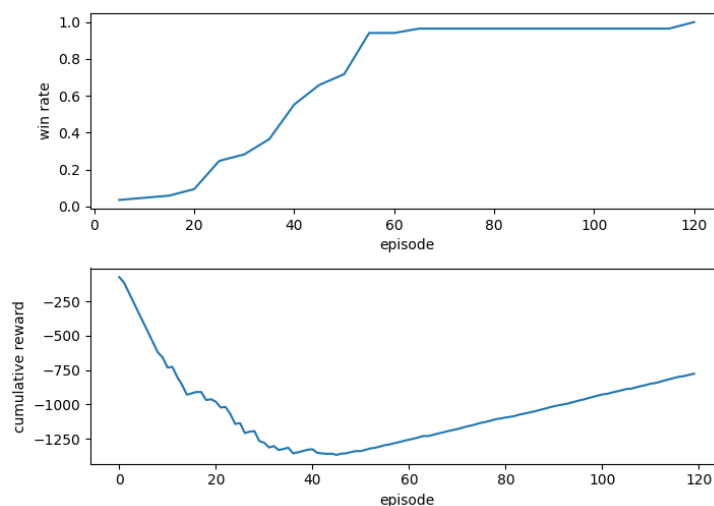
```
INFO : 2022-06-12 17:47:15: won: 82 | lost: 3 | win rate: 0.96471
```

Nakon 100 epizoda, rezultat je odličan, te imamo 82 pobjede i 3 poraza. Win rate je iznosio: 0.96471.

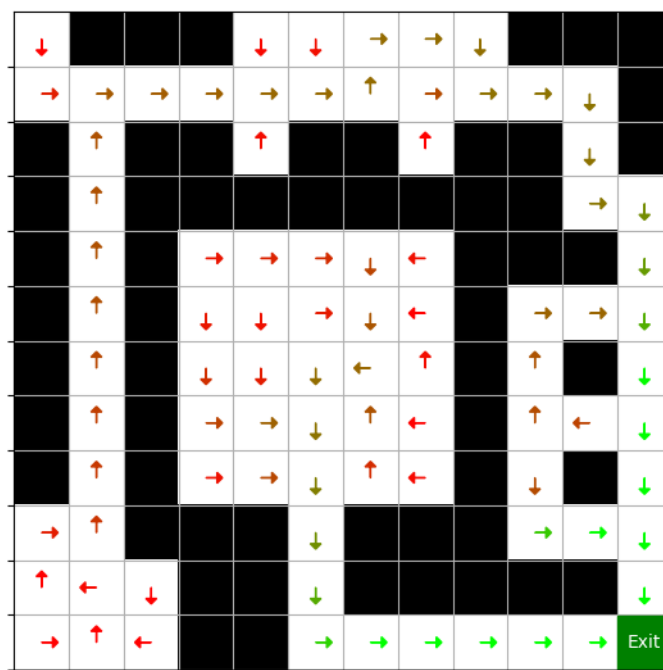
```
INFO : 2022-06-12 17:48:43: episode: 120/200 | status: WIN | e: 0.05507
```

```
INFO : 2022-06-12 17:48:43: won: 85 | lost: 0 | win rate: 1.00000
```

Vidimo da smo u ovom slučaju, već nakon 120 epizoda, dostigli Win rate od 1.00000 te imamo 85 pobjeda i 0 poraza. Dijagrami koje smo dobili po izvršenju ovog zadatka su prikazani na narednoj slici:



Sa slike možemo vidjeti kako se Win rate mijenjao tokom prolaska kroz epizode, te vidimo da je nakon 120 epizoda ovaj parametar dosegao maksimalnu vrijednost. U poređenju s prethodna dva slučaja, vidimo da imamo znatno strmije povećanje win rate parametra, te možemo vidjeti da se već nakon 50 epizoda ima izuzetno dobra vrijednost ovog parametra. Labirint je na kraju poprimio sljedeći oblik:



4. Četvrti slučaj je sa sljedećim parametrima: ***discount = 0.80, learning_rate=0.80, episodes = 200***

U svakom od navedenih primjera presjek ćemo praviti na sljedećim epizodama (10, 50, 100, 150 itd.) Prvi presjek pravimo nakon 10 epizoda te možemo vidjeti sljedeći rezultat:

INFO : 2022-06-12 18:03:49: episode: 10/200 | status: WIN | e: 0.09559

INFO : 2022-06-12 18:03:50: won: 7 | lost: 78 | win rate: 0.08235

Vidimo da smo nakon 10 epizoda imali izuzetno loš rezultat, odnosno, u 7 slučajeva je ostvarena pobjeda, dok je u 78 slučajeva ishod bio poraz. Win rate je iznosio: 0.08235.

INFO : 2022-06-12 18:15:33: episode: 50/200 | status: WIN | e: 0.07822

INFO : 2022-06-12 18:15:34: won: 33 | lost: 52 | win rate: 0.38824

Nakon 50 epizoda, rezultat se neznatno popravio, te je u 33 slučajeva ostvarena pobjeda, dok je u 52 slučajeva ostvaren poraz. Win rate je iznosio: 0.38824 .

INFO : 2022-06-12 18:23:52: episode: 100/200 | status: LOSE | e: 0.06088

INFO : 2022-06-12 18:23:52: won: 65 | lost: 20 | win rate: 0.76471

Nakon 100 epizoda, rezultat je mnogo bolji, te imamo 65 pobjeda i 20 poraza. Win rate je iznosio: 0.76471.

INFO : 2022-06-12 18:28:57: episode: 150/200 | status: WIN | e: 0.04738

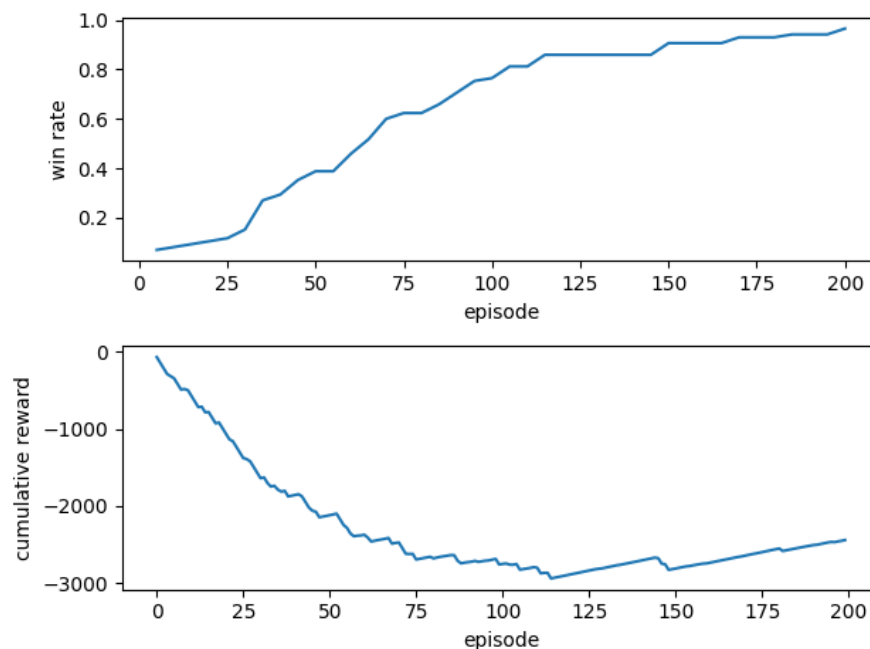
INFO : 2022-06-12 18:28:57: won: 77 | lost: 8 | win rate: 0.90588

Poslije 150 epizoda, rezultat je izuzetno dobar, te imamo 77 pobjeda i 5 poraza. Win rate je iznosio: 0.90588.

INFO : 2022-06-12 18:32:23: episode: 200/200 | status: WIN | e: 0.03688

INFO : 2022-06-12 18:32:23: won: 82 | lost: 3 | win rate: 0.96471

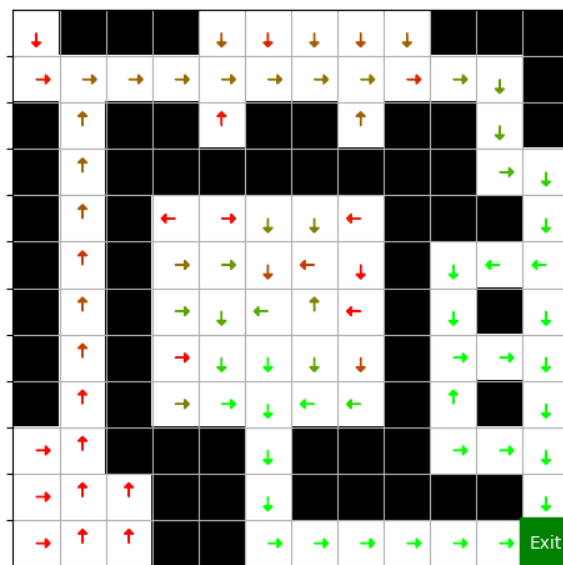
Vidimo da se nakon 200 epizoda trening prekinuo (taj broj smo postavili kao maksimalan broj epizoda), te da nismo dostigli Win rate od 1.00000, odnosno imamo Win rate od 0.96471. Dijagrami koje smo dobili po izvršenju ovog zadatka su prikazani na narednoj slici:



Sa slike možemo vidjeti kako se Win rate mijenjao tokom prolaska kroz epizode, te vidimo da je nakon 200 epizoda ovaj parametar dosegao win rate blizu maksimalnog, odnosno, win rate oko 0.96. Jedini slučaj kada nismo imali maksimalan win rate je ovaj. Odnosno, možemo zaključiti sljedeće:

U prvom slučaju (parametri: **discount = 0.90**, **learning_rate=0.10**, **episodes = 200**) smo nakon 180 epizoda dostigli maksimalnu vrijednost win rate-a. Isto je bilo u drugom slučaju (parametri: **discount = 0.70**, **learning_rate=0.10**, **episodes = 200**). Ako uporedimo ova dva slučaja, vidimo da je drugi slučaj nešto bolji, odnosno, da smo se nakon manje epizoda približili maksimalnoj vrijednosti parametra od značaja (iako je u oba slučaja do maksimalne vrijednosti trebao isti broj epizoda). U trećem slučaju (parametri: **discount = 0.60**, **learning_rate=0.30**, **episodes = 200**) smo imali najmanji broj epizoda (120) i najstrimiji dijagram za parametar

win rate. U posljednjem slučaju (parametri: *discount* = 0.80, *learning_rate*=0.80, *episodes* = 200) smo imali najlošije rješenje, odnosno ni nakon 200 epizoda nismo postigli željeni rezultat. Labirint u ovom (četvrtom) slučaju ima sljedeći oblik:



Zadnje što ćemo uraditi u ovom zadatku jeste mijenjati broj nagrada i kazni, te na osnovu toga formirati odgovarajuću tabelu. Dio koda koji je potrebno mijenjati je prikazan u narednom box-u:

```
reward_exit = 10.0 # reward for reaching the exit cell
penalty_move = -0.05 # penalty for a move which did not result in finding the exit cell
# mijenjali smo ovo
penalty_visited = -0.95 # penalty for returning to a cell which was visited earlier
# penalty for trying to enter an occupied cell or moving out of the maze
penalty_impossible_move = -1.75
```

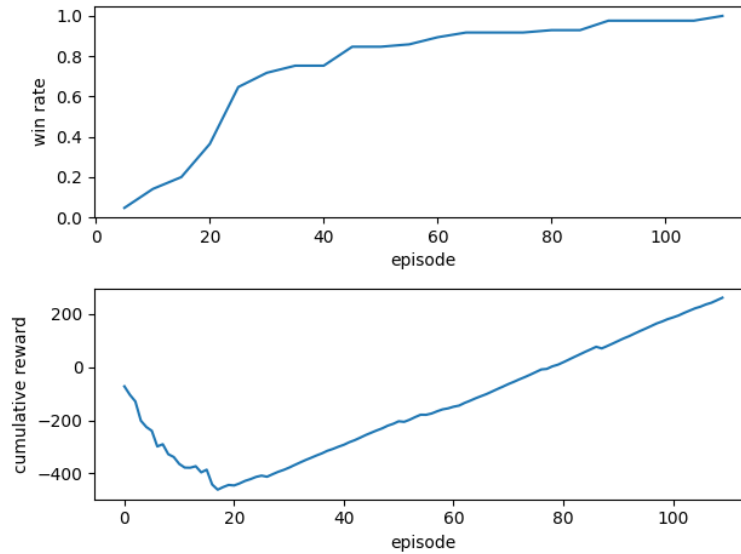
1. Slučaj (*reward_exit* =10, *penalty_move* = -0.05, *penalty_visited* = -0.3, *penalty_impossible_move* = -0.9)

Nakon pokretanja ovog koda smo dobili sljedeći rezultat:

```
INFO : 2022-06-13 11:29:11: episode: 110/200 | status: WIN | e: 0.05790
INFO : 2022-06-13 11:29:11: won: 85 | lost: 0 | win rate: 1.00000
```

Možemo primjetiti da smo u ovom slučaju već nakon 110 epizoda dobili maksimalan win rate. Bitno je napomenuti da smo u ova 3 slučaja, za discount i learning rate uzeli najbolje podatke iz prethodnog slučaja te ćemo na taj način pokušati da poboljšamo dodatno rezultat prethodna četiri slučaja. Pod navedenim poboljšanjem mislimo na pokušaj smanjivanja broja epizoda i bržeg dolaženja do samog rezultata.

Dijagram:



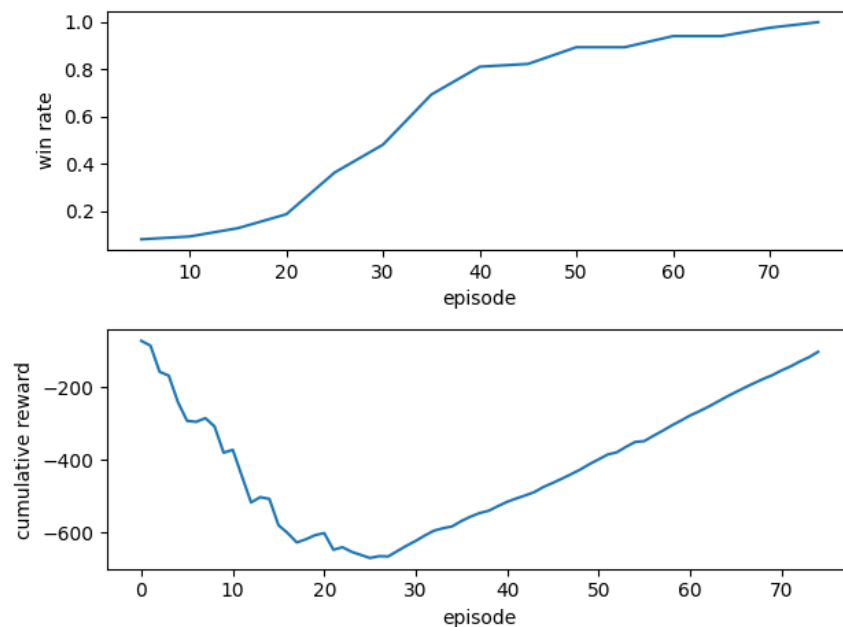
2. Slučaj ($reward_exit = 15$, $penalty_move = -0.1$, $penalty_visited = -0.5$, $penalty_impossible_move = -1.1$)

Nakon pokretanja ovog koda smo dobili sljedeći rezultat:

```
INFO : 2022-06-13 11:45:19: episode: 75/200 | status: WIN | e: 0.06901
INFO : 2022-06-13 11:45:19: won: 85 | lost: 0 | win rate: 1.00000
```

Možemo primjetiti da smo u ovom slučaju dostigli maksimalan win rate nakon samo 75 epizoda.

Dijagram:



3. Slučaj ($reward_exit = 20$, $penalty_move = -0.2$, $penalty_visited = -0.4$, $penalty_impossible_move = -1.6$)

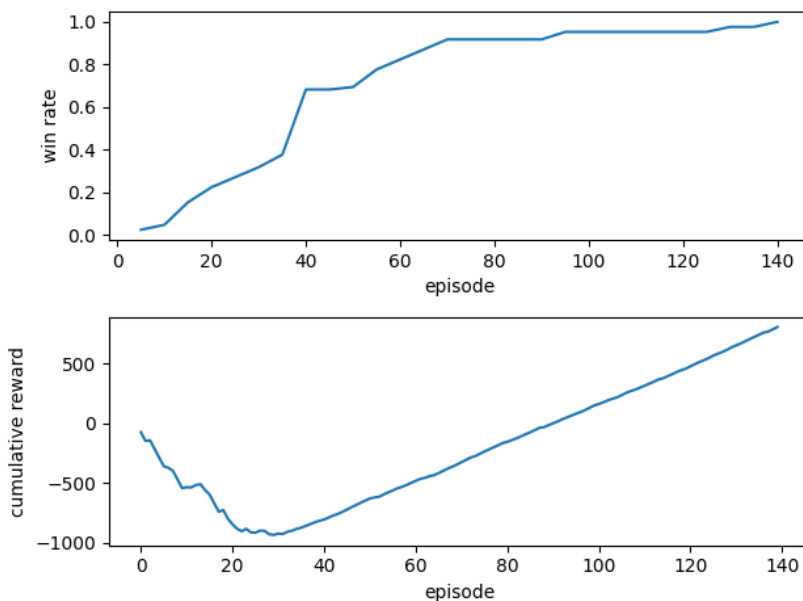
Nakon pokretanja ovog koda smo dobili sljedeći rezultat:

INFO : 2022-06-13 12:10:42: episode: 140/200 | status: WIN | e: 0.04982

INFO : 2022-06-13 12:10:43: won: 85 | lost: 0 | win rate: 1.00000

Možemo primjetiti da smo u ovom slučaju dostigli maksimalan win rate nakon samo 140 epizoda.

Dijagram:



Konačna tabela je prikazana u nastavku:

Discount	Learning rate	Reward exit	Penalty move	Penalty visited	Penalty impossible move	Episodes	Win rate
0.6	0.3	10	-0.05	-0.3	-0.9	110	100%
0.6	0.3	15	-0.1	-0.5	-1.1	75	100%
0.6	0.3	20	-0.2	-0.4	-1.6	140	100%