Sprawozdanie – zbiory rozmyte

Opracował: Szymon Ryś, nr 401471

Wybrana dziedzina

Stworzyłem sterownik rozmyty do obsługi pralki. W zależności od podanych na wejściu temperatury, twardości wody, zmiękczania środka zmiękczającego oraz czasu prania wprowadzonego przez użytkownika dobiera czas prania właściwie do danych wejściowych.

Środowisko i uruchamianie

Do stworzenia tego sterownika wybrałem język *Python* (v. 3.9.9) oraz bibliotekę *scikit-fuzzy (v. 0.4.2).*

Aby włączyć kod należy wykonać polecenie:

pip3 install -r requirements.cfg

w celu instalacji niezbędnych bibiotek, oraz polecenie:

python3 main.py

w celu odpalenia sterownika.

Wprowadzone są 3 przykładowe wartości dla sterownika, które opiszę w dalszej części sprawozdania.

Wejścia i wyjścia

Wejścia:

1. Temperatura prania - *temperature*
2. Współczynnik twardości wody – *hardness*
3. Współczynnik zmiękczania środka piorącego - *softening_factor*
4. Wprowadzony przez użytkownika czas – *user_duration*

```
# 4 inputs
temperature = ctrl.Antecedent(np.arange(0, 11, 1), 'temperature')             # water temperature
hardness = ctrl.Antecedent(np.arange(0, 11, 1), 'hardness')                   # water hardness factor
softening_factor = ctrl.Antecedent(np.arange(0, 11, 1), 'softening_factor')   # washing powder softerning factor
user_duration = ctrl.Antecedent(np.arange(0, 11, 1), 'user_duration')         # user length duration
```

Wszystkie zmienne wejściowe są ze skali rzeczywistej [0, 10] przetworzone na skalę 7-stopniową:

- dismal
- poor
- mediocre
- average
- decent
- good
- excellent

Wyjście:

1. Czas prania

```
# 1 output
output1_duration = ctrl.Consequent(np.arange(0, 16, 1), 'duration')                # output washing duration
```

Wyjście jest w skali rzeczywistej [0, 15] jest przetworzone na skalę 7-stopniową:

- lowest
- lower
- low
- medium
- high
- higher
- highest

Uwaga: zauważmy, że długość czasu prania jest proporcjonalna do twardości wody oraz czasu wprowadzonego przez użytkownika lecz odwrotnie proporcjonalna do temperatury i współczynnika zmiękczania.

```
temperature.automf(7)
hardness.automf(7)
softening_factor.automf(7)
user_duration.automf(7)

# define possible output states
output1_duration['lowest'] = fuzz.trimf(output1_duration.universe, [0, 0, 3])
output1_duration['lower'] = fuzz.trimf(output1_duration.universe, [0, 3, 5])
output1_duration['low'] = fuzz.trimf(output1_duration.universe, [3, 5, 8])
output1_duration['medium'] = fuzz.trimf(output1_duration.universe, [5, 8, 11])
output1_duration['high'] = fuzz.trimf(output1_duration.universe, [8, 11, 13])
output1_duration['higher'] = fuzz.trimf(output1_duration.universe, [11, 13, 15])
output1_duration['highest'] = fuzz.trimf(output1_duration.universe, [13, 15, 15])
```

Reguły

```
# most pesimistic case
r1 = ctrl.Rule(temperature['dismal'] | hardness['excellent'] | softening_factor['dismal'] | user_duration['excellent'], output1_duration['highest'])

# most optimistic case
r2 = ctrl.Rule(temperature['excellent'] | hardness['dismal'] | softening_factor['excellent'] | user_duration['dismal'], output1_duration['lowest'])

# medium cases
r3 = ctrl.Rule(temperature['dismal'] | hardness['poor'] | softening_factor['mediocre'] | user_duration['mediocre'], output1_duration['medium'])
r4 = ctrl.Rule(temperature['excellent'] | hardness['decent'] | softening_factor['decent'] | user_duration['excellent'], output1_duration['medium'])
r10 = ctrl.Rule(temperature['average'] | hardness['average'] | softening_factor['average'] | user_duration['average'], output1_duration['medium'])

r5 = ctrl.Rule(temperature['decent'] | hardness['mediocre'] | softening_factor['good'] | user_duration['mediocre'], output1_duration['lower'])
r6 = ctrl.Rule(temperature['average'] | hardness['good'] | softening_factor['average'] | user_duration['average'], output1_duration['medium'])
r7 = ctrl.Rule(temperature['poor'] | hardness['average'] | softening_factor['good'] | user_duration['average'], output1_duration['medium'])
r8 = ctrl.Rule(temperature['good'] | hardness['excellent'] | softening_factor['poor'] | user_duration['excellent'], output1_duration['high'])

r9 = ctrl.Rule(temperature['poor'] | hardness['mediocre'] | softening_factor['good'] | user_duration['average'], output1_duration['medium'])
r11 = ctrl.Rule(temperature['excellent'] | hardness['decent'] | softening_factor['good'] | user_duration['dismal'], output1_duration['lowest'])
r12 = ctrl.Rule(temperature['poor'] | hardness['mediocre'] | softening_factor['average'] | user_duration['poor'], output1_duration['higher'])
r13 = ctrl.Rule(temperature['excellent'] | hardness['good'] | softening_factor['decent'] | user_duration['mediocre'], output1_duration['low'])
r14 = ctrl.Rule(temperature['mediocre'] | hardness['dismal'] | softening_factor['average'] | user_duration['good'] , output1_duration['higher'])
r15 = ctrl.Rule(temperature['average'] | hardness['decent'] | softening_factor['excellent'] | user_duration['poor'], output1_duration['lower'])
```

1. Reguły *r1* i *r2* wprowadzają najbardziej optymistyczne i pesymistyczne wartości jeśli chodzi o długość czasu prania
2. Reguły *r3, r4 i r10* wprowadzają wartości średnie
3. Reguły *r5-8* przetwarzają wyjścia traktując je trochę jak średnią arytmetyczną przypadków
4. Pozostałe reguły działają podobnie do reguł *r5-8* jednak są przeszacowane a nie wyliczona dokładnie

Reszta kodu

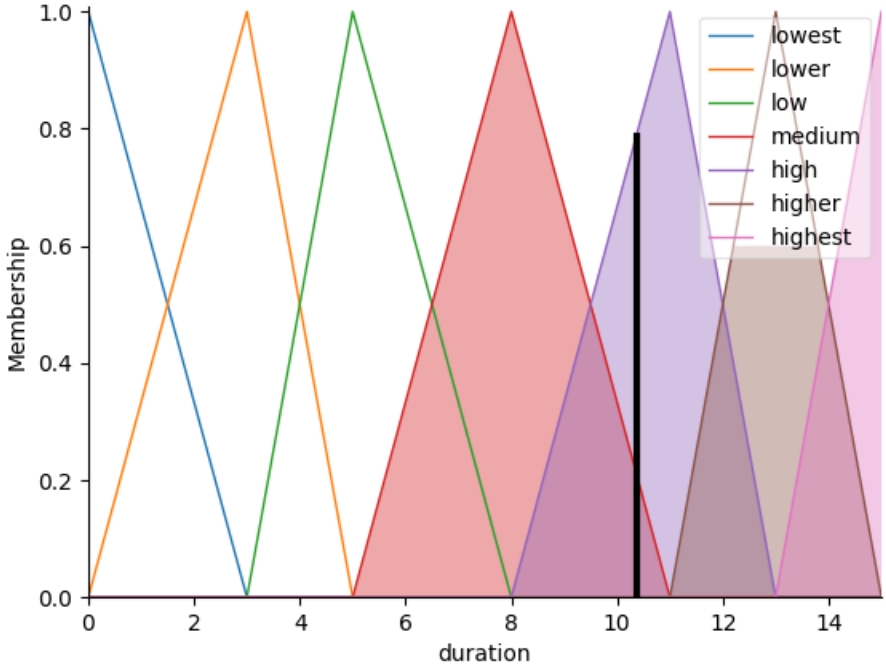
Reszta kodu to odpowiednio wybór reguł z których korzysta sterownik oraz dla każdego z 5 przypadków obliczenie wyniku

Test 1 –przypadek pesymistyczny - długie pranie:

```
# Set up arguments
outputt.input['temperature'] = 1.0
outputt.input['hardness'] = 10.0
outputt.input['softening_factor'] = 1.0
outputt.input['user_duration'] = 10.0

outputt.compute()

print(outputt.output['duration'])
output1_duration.view(sim=outputt)
plt.show()
```

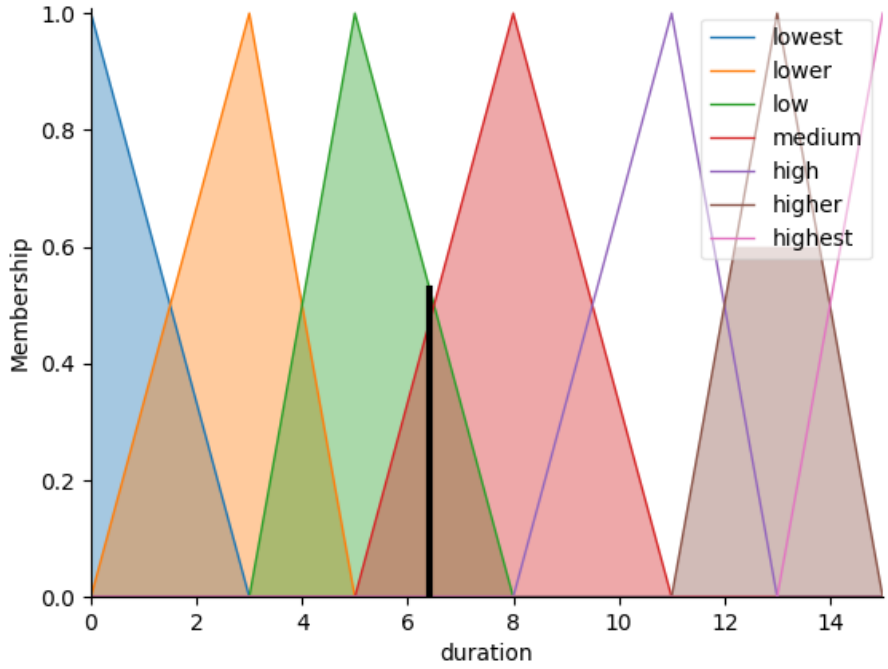


Test 2 –przypadek optymistyczny - krótkie pranie:

```
outputt.input['temperature'] = 10.0
outputt.input['hardness'] = 1.0
outputt.input['softening_factor'] = 10.0
outputt.input['user_duration'] = 1.0

outputt.compute()

print(outputt.output['duration'])
output1_duration.view(sim=outputt)
plt.show()
```

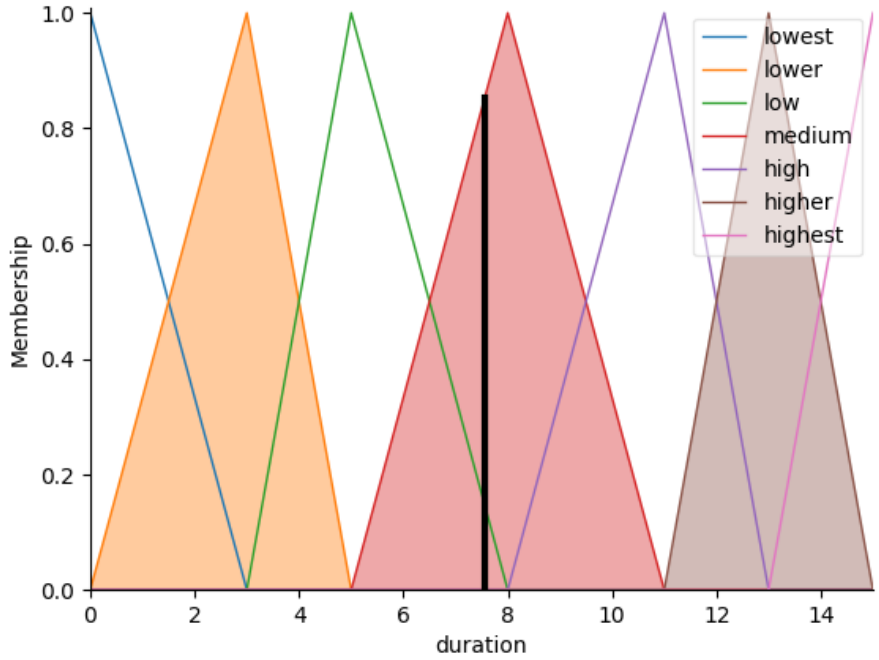


Test 3 –przypadek pośredni:

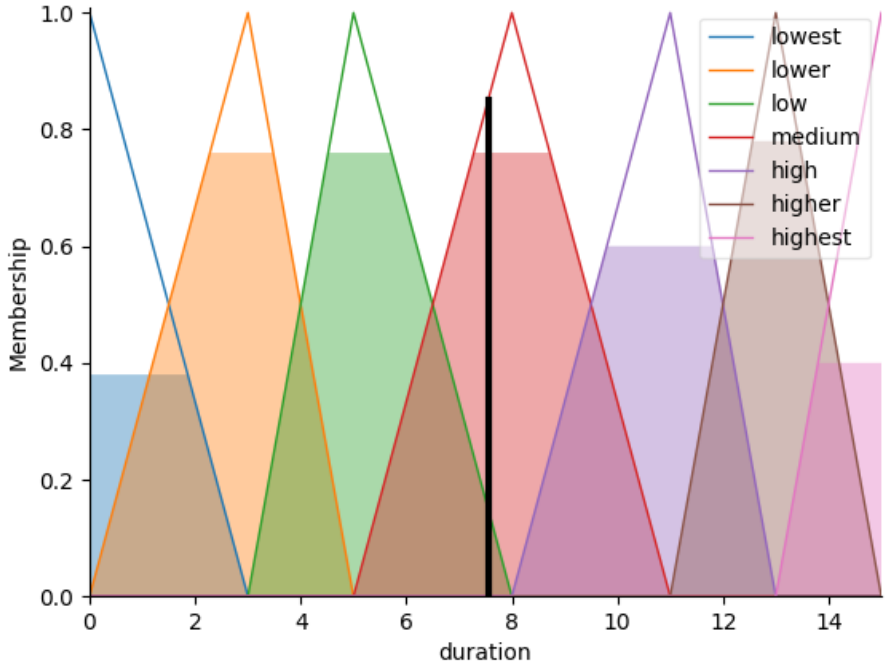
```
outputt.input['temperature'] = 5.0
outputt.input['hardness'] = 5.0
outputt.input['softening_factor'] = 5.0
outputt.input['user_duration'] = 5.0

outputt.compute()

print(outputt.output['duration'])
output1_duration.view(sim=outputt)
plt.show()
```



Test 4 wartości rozrzucone:



Test 5 wartości rozrzucone: