



دانشگاه صنعتی امیر کبیر
(بلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

پروژه دوم درس مبانی هوش محاسباتی

پیاده سازی سیستم خبره فازی تشخیص بیماری

نگارش:

محمدسپهر توکلی کرمانی - ۹۸۳۱۱۱۱

استاد درس:

دکتر عبادزاده

تیر ۱۴۰۱

این پروژه شامل ۳ بخش اصلی است که در ادامه به توضیح هر یک می پردازیم :

بخش اول : fuzzification

در این بخش با استفاده از نمودار های مقادیر تعلیق و هم چنین توضیحات ارائه شده در دستور کار ، با دریافت متغیر های crisp (که از طریق Gui به دست آمده) آنها را به مقادیر نادقیق فازی تبدیل می کنیم. برای این کار به ازای هر متغیر (در مجموع ۱۱ متغیر ورودی و یک متغیر خروجی داریم) ، یک کلاس در نظر گرفته و تابع های مربوطه را پیاده سازی می کنیم :

```
+ class age_fuzzification: ...  
+ class bloodPressure_fuzzification: ...  
+ class bloodSugar_fuzzification: ...  
+ class cholestrol_fuzzification: ...  
+ class heartRate_fuzzification: ...  
+ class ECG_fuzzification: ...  
+ class oldPeak_fuzzification: ...  
+ class outputsick_fuzzification: ...  
+ class chest_pain_fuzzification: ...  
+ class exercise_fuzzification: ...  
+ class thallium_fuzzification: ...  
+ class sex_fuzzification: ...
```

برای مثال کلاس متغیر سن، به این شکل خواهد بود :

```
class age_fuzzification:
    def __init__(self):
        pass

    def age_young(self,x):
        if x <= 29:
            return 1
        if 29 < x <= 38:
            return -0.111 * x + 4.222
        else :
            return 0

    def age_mild(self,x):
        if 33 < x < 38:
            return 0.2 * x - 6.600
        if x == 38:
            return 1
        if 38 < x < 45:
            return -0.143 * x + 6.429
        else :
            return 0

    def age_old(self,x):
        if 40 < x < 48:
            return 0.125 * x - 5.000
        if x == 48 :
            return 1
        if 48 < x < 58:
            return -0.1 * x + 5.800
        else :
            return 0

    def age_veryold(self,x):
        if x <= 52:
            return 0
        if 52 < x < 60:
            return 0.125 * x - 6.500
        else :
            return 1

    def calc_fuzzy_age(self,age):
        return dict (
            young=self.age_young(age),
            mild=self.age_mild(age),
            old=self.age_old(age),
            very_old=self.age_veryold(age)
        )
```

بخش دوم : inference

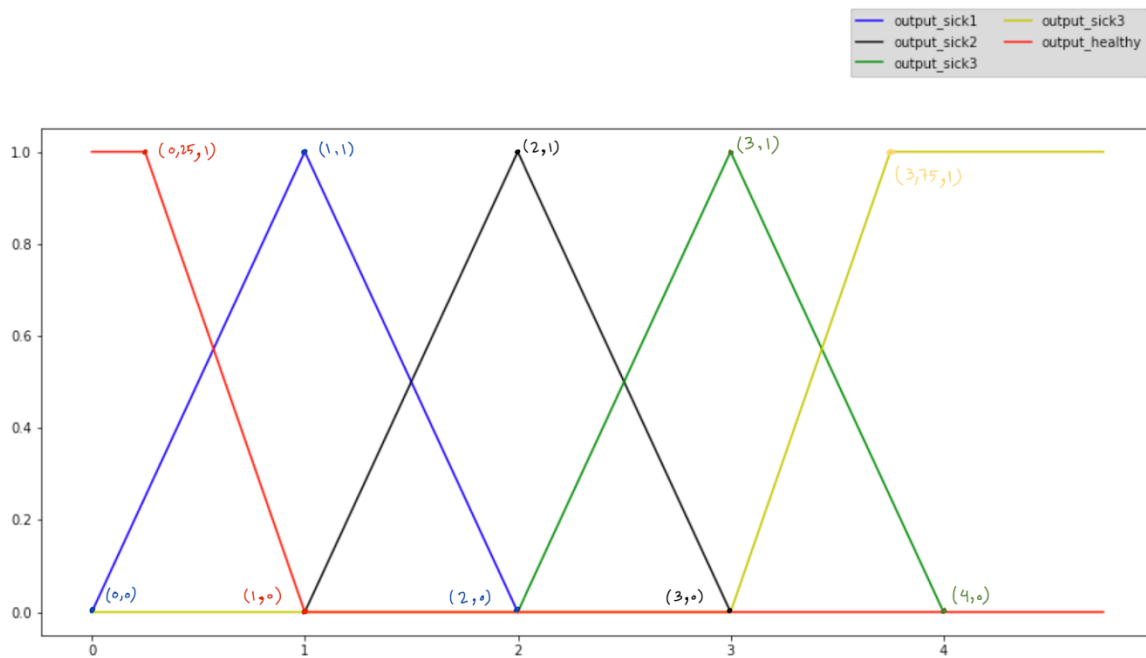
در بخش با استفاده از قوانین داده شده در فایل fcl و با مقادیر به دست آمده از قسمت قبلی ، میزان تعلق به هر یک از گروه های sick1 ، sick2 ، sick3 ، sick4 و healthy را محاسبه کرده و در نهایت در قالب یک دیکشنری پایتون آن را بر می گردانیم:

```
def inference(self,chest_pain,blood_pressure,cholesterol,blood_sugar,ECG,maximum_heart_rate,exercise,old_peak,thallium,sex,age):
    output_sick1,output_sick2,output_sick3,output_sick4,output_healthy = ([[] for i in range(5)])
    # 54 rules defined here ...
    output_sick4.append(min(age['very_old'],chest_pain['atypical_anginal']))
    output_sick4.append(min(maximum_heart_rate['high'],age['old']))
    output_sick3.append(min(sex['male'],maximum_heart_rate['medium']))
    output_sick2.append(min(sex['female'],maximum_heart_rate['medium']))
    output_sick3.append(min(chest_pain['non_aginal_pain'],blood_pressure['high']))
    output_sick2.append(min(chest_pain['typical_anginal'],maximum_heart_rate['medium']))
    output_sick3.append(min(blood_sugar['true'],age['mild']))
    output_sick2.append(min(blood_sugar['false'],blood_pressure['very_high']))
    output_sick1.append(max(chest_pain['asymptomatic'],age['very_old']))
    output_sick1.append(max(blood_pressure['high'],maximum_heart_rate['low']))
    output_healthy.append(chest_pain['typical_anginal'])
    output_sick1.append(chest_pain['atypical_anginal'])
    output_sick2.append(chest_pain['non_aginal_pain'])
    output_sick3.append(chest_pain['asymptomatic'])
    output_sick4.append(chest_pain['asymptomatic'])
    output_sick1.append(sex['female'])
    output_sick2.append(sex['male'])
    output_healthy.append(blood_pressure['low'])
    output_sick1.append(blood_pressure['medium'])
    output_sick2.append(blood_pressure['high'])
    output_sick3.append(blood_pressure['high'])
    output_sick4.append(blood_pressure['very_high'])
    output_healthy.append(cholesterol['low'])
    output_sick1.append(cholesterol['medium'])
    output_sick2.append(cholesterol['high'])
    output_sick3.append(cholesterol['high'])
    output_sick4.append(cholesterol['very_high'])
    output_sick2.append(blood_sugar['true'])
    output_healthy.append(ECG['normal'])
    output_sick1.append(ECG['normal'])
    output_sick2.append(ECG['abnormal'])
    output_sick3.append(ECG['hypertrophy'])
    output_sick4.append(ECG['hypertrophy'])
    output_healthy.append(maximum_heart_rate['low'])
    output_sick1.append(maximum_heart_rate['medium'])
    output_sick2.append(maximum_heart_rate['medium'])
    output_sick3.append(maximum_heart_rate['high'])
    output_sick4.append(maximum_heart_rate['high'])
    output_sick2.append(exercise['true'])
    output_healthy.append(old_peak['low'])
    output_sick1.append(old_peak['low'])
    output_sick2.append(old_peak['terrible'])
    output_sick3.append(old_peak['terrible'])
    output_sick4.append(old_peak['risk'])
    output_healthy.append(thallium['normal'])
    output_sick1.append(thallium['normal'])
    output_sick2.append(thallium['medium'])
    output_sick3.append(thallium['high'])
    output_sick4.append(thallium['high'])
    output_healthy.append(age['young'])
    output_sick1.append(age['mild'])
    output_sick2.append(age['old'])
    output_sick3.append(age['old'])
    output_sick4.append(age['very_old'])
    return dict(
        outputsick_sick1=max(output_sick1),
        outputsick_sick2=max(output_sick2),
        outputsick_sick3=max(output_sick3),
        outputsick_sick4=max(output_sick4),
        outputsick_healthy=max(output_healthy)
    )
```

بخش سوم : defuzzification

در این بخش در فایل defuzzification.py منطق غیرفازی سازی را پیاده می کنیم، که در آن با استفاده از نمودار زیر و فرمول مرکز جرم ، گروه بیماری فرد مورد نظر محاسبه می شود و در نهایت آنرا به همراه مرکز جرم محاسبه شده در gui داده شده نمایش می دهیم.

نمودار مقادیر تعلق خروجی :



فرمول محاسبه مرکز جرم :

$$x^* = \frac{\sum_{i=1}^n \mu_{\bar{C}}(x_i) \cdot x_i}{\sum_{i=1}^n \mu_{\bar{C}}(x_i)}$$

در نهایت کد ما به این شکل خواهد بود :

```
class defuzzification:
    def __init__(self):
        pass

    def defuzzify(self, x):
        n = 300
        delta = 1. / n
        points = [0. + i * delta for i in range(4*n)]

        numerator = 0.
        denominator = 0.
        for point in points:
            s1 = outputsick_fuzzification2.outputsick_sick1(point)
            if s1 > x['outputsick_sick1']:
                s1 = x['outputsick_sick1']
            s2 = outputsick_fuzzification2.outputsick_sick2(point)
            if s2 > x['outputsick_sick2']:
                s2 = x['outputsick_sick2']
            s3 = outputsick_fuzzification2.outputsick_sick3(point)
            if s3 > x['outputsick_sick3']:
                s3 = x['outputsick_sick3']
            s4 = outputsick_fuzzification2.outputsick_sick4(point)
            if s4 > x['outputsick_sick4']:
                s4 = x['outputsick_sick4']
            sh = outputsick_fuzzification2.outputsick_healthy(point)
            if sh > x['outputsick_healthy']:
                sh = x['outputsick_healthy']

            res = max(s1, s2, s3, s4, sh)
            numerator += res * point * delta
            denominator += res * delta

        try:
            centerOfGravity = numerator / denominator
        except ZeroDivisionError:
            centerOfGravity = 0

        result = []

        if centerOfGravity < 1.78 :
            result.append('healthy')

        if 1 <= centerOfGravity <= 2.51:
            result.append('sick1')

        if 1.78 <= centerOfGravity < 3.25:
            result.append('sick2')

        if 1.5 <= centerOfGravity <= 4.5:
            result.append('sick3')

        if 3.25 <= centerOfGravity:
            result.append('sick4')

        result.append(str( round(centerOfGravity,3)))

        return " & ".join(result)
```

نمونہ خروجی

Chest Pain	<input type="range"/>	3	Blood Pressure	<input type="range"/>	243
Cholestrol	<input type="range"/>	89	Blood Sugar	<input type="range"/>	166
ECG	<input type="range"/>	1	Heart Rate	<input type="range"/>	214
Exercise	<input type="range"/>	0	Old Peak	<input type="range"/>	1.5
Thallium Scan	<input type="range"/>	6	Sex	<input type="range"/>	1
Age	<input type="range"/>	52			

Show Result

Result is sick1 & sick2 & sick3 & 1.998