



دانشکده مهندسی کامپیوتر

گزارش پایانی درس: مبانی هوش محاسباتی

عنوان پژوهش: تمرین دوم بخش دوم (طراحی سیستم منطق فازی برای توصیه تمرینات ورزشی)

ارائه دهنده:

نرگس سادات موسوی جد

شادی شاهی محمدی

استاد درس:

دکتر کارشناس

بهار ۱۴۰۴

فهرست:

- ۱- پیاده سازی سیستم فازی ۱
- ۱-۱- توابع عضویت : ۱
- ۲-۱- متغیرهای فازی: ۲
- ۳-۱- وزندهی به خروجی‌ها: ۷
- ۴-۱- توضیح قوانین: ۸
- ۴-۱- فازی سازی و فازی زدایی: ۱۳
- ۲- نتایج: ۱۴

۱- پیاده سازی سیستم فازی

۱-۱- توابع عضویت :

این توابع عضویت برای تبدیل مقادیر دقیق (مثلاً سطح تناسب اندام، انرژی) به مقادیر فازی (مقادیر بین ۰ و ۱) استفاده می‌شوند. نمونه‌هایی از این توابع شامل توابع مثلثی، دوزنقه‌ای، سیگموئید و گاوسی برای پارامترهایی مانند تناسب اندام، انرژی و هدف هستند.

```
def triangular(x, a, b, c):
    if x <= a or x >= c:
        return 0
    elif a < x < b:
        return (x - a) / (b - a)
    elif b <= x < c:
        return (c - x) / (c - b)
    elif x == b:
        return 1

def trapezoidal(x, a, b, c, d):
    if x <= a or x >= d:
        return 0
    elif a < x < b:
        return (x - a) / (b - a)
    elif b <= x <= c:
        return 1
    elif c < x < d:
        return (d - x) / (d - c)

def sigmoid(x, c, a):
    value = 1 / (1 + np.exp(-a * (x - c)))
    if value == np.inf:
        return 1
    return value

def gaussian(x, m, s):
    return math.exp(-((x - m) ** 2 / (s ** 2)))

def crisp(x, a):
    return 1 if x == a else 0
```

۱-۲- متغیرهای فازی:

در سیستم توصیه‌گر ورزشی، از متغیرهای فازی مختلف برای مدل‌سازی ویژگی‌های فردی استفاده می‌شود. این متغیرها به دسته‌های مختلف تقسیم می‌شوند که هر کدام ویژگی خاصی را مدل‌سازی می‌کنند. به عنوان مثال، متغیرهای تناسب اندام از نوع مثلثی و سیگموئید هستند و برای مدل‌سازی سطح انرژی از توابع مثلثی و گاوسی استفاده شده است. همچنین متغیرهایی برای مدل‌سازی سن، وزن، هدف و مدت زمان تمرین نیز تعریف شده‌اند.

متغیر تناسب اندام: (fitness_functions)

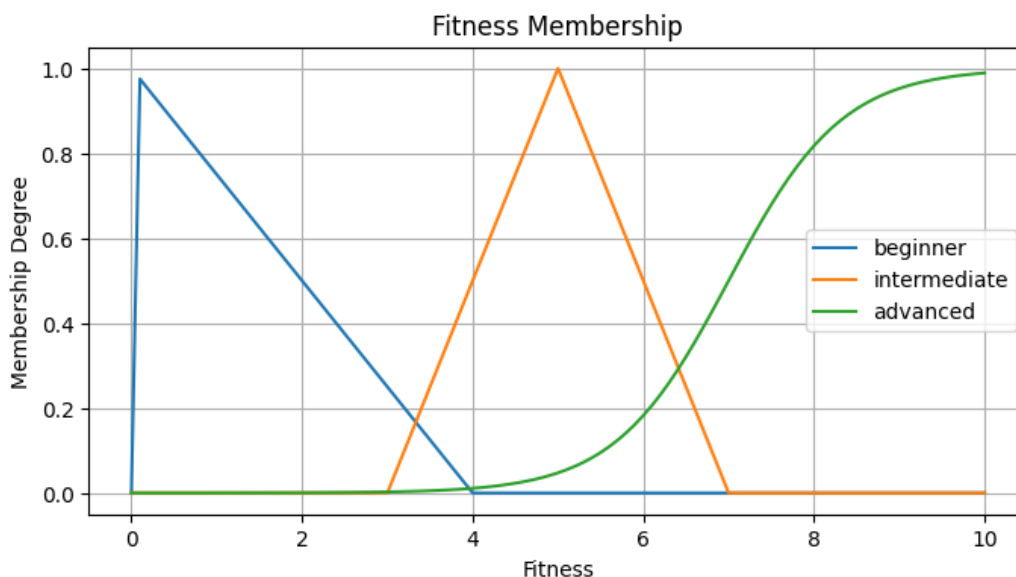
متغیر فازی تناسب اندام برای مدل‌سازی سه سطح مختلف تناسب اندام تعریف شده‌اند. این توابع برای تنظیم شدت تمرین بر اساس سطح تناسب اندام کاربر به کار می‌روند:

Beginner: برای افراد مبتدی که سطح تناسب اندام آنها از ۰ تا ۴ تغییر می‌کند.

Intermediate: برای افرادی که در سطح متوسط تناسب اندام هستند (بین ۳ تا ۷).

Advanced: برای افرادی که در سطح پیشرفته تناسب اندام قرار دارند (با تابع سیگموئید که از ۷ به بالا تغییر می‌کند).

```
fitness_functions = {  
    'beginner': lambda x: triangular(x, 0, 0, 4),  
    'intermediate': lambda x: triangular(x, 3, 5, 7),  
    'advanced': lambda x: sigmoid(x, 7, 1.5)  
}
```



متغیر سطح انرژی: (energy_functions)

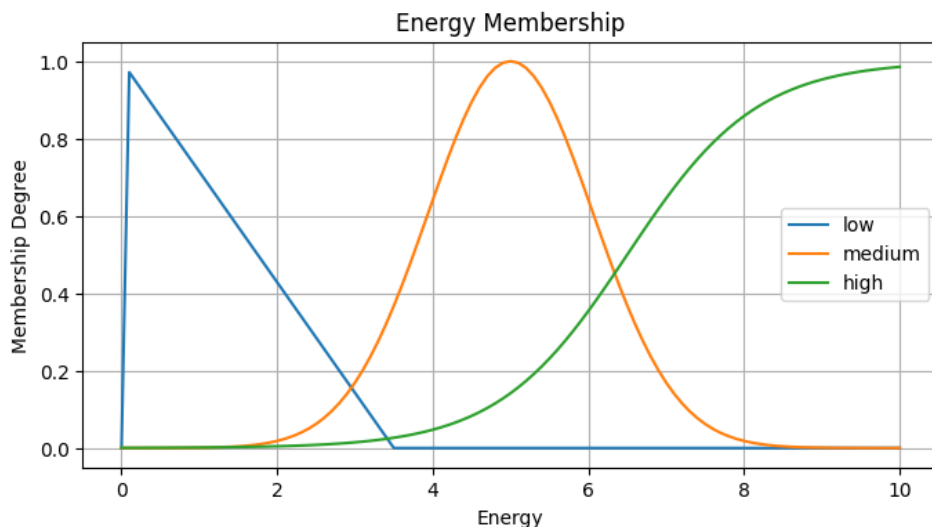
برای تعیین میزان انرژی فرد، سه سطح انرژی مختلف در نظر گرفته شده است. این متغیر بیشتر بر مدت زمان تمرین تاثیر دارد:

Low: سطح انرژی پایین که با تابع مثلثی از ۰ تا ۳.۵ مدل سازی می شود.

Medium: سطح انرژی متوسط که با تابع گاوسی مشخص می شود.

High: سطح انرژی بالا که با استفاده از تابع سیگموئید از ۶.۵ به بعد افزایش می یابد.

```
energy_functions = {  
    'low': lambda x: triangular(x, 0, 0, 3.5),  
    'medium': lambda x: gaussian(x, 5, 1.5),  
    'high': lambda x: sigmoid(x, 6.5, 1.2)  
}
```



متغیر سن: (age_functions)

برای مدل سازی سن، چهار دسته مختلف در نظر گرفته شده است:

Kid: برای کودکان که سن آنها بین ۰ تا ۱۸ سال است.

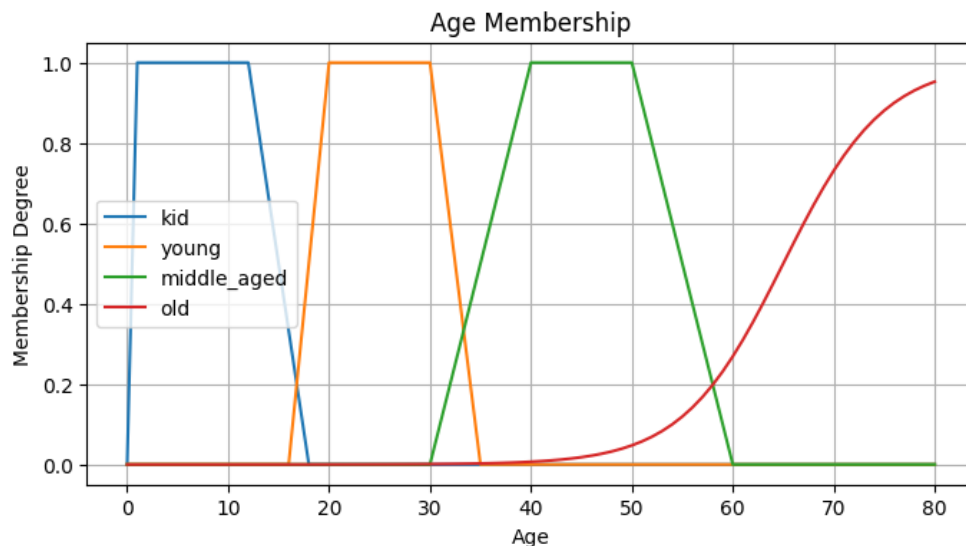
Young: برای افراد جوان با سن بین ۱۶ تا ۳۵ سال.

middle_aged: برای افراد میانه سالی که سن آنها بین ۳۰ تا ۶۰ سال است.

Old: برای افراد مسن که سن آنها ۶۵ سال به بالا است و تابع سیگموئید برای این دسته استفاده می شود.

این توابع به سیستم کمک می‌کنند تا شدت و مدت زمان تمرینات را با توجه به سن کاربر تنظیم کنند.

```
age_functions = {
    'kid': lambda x: trapezoidal(x, 0, 0, 12, 18),
    'young': lambda x: trapezoidal(x, 16, 20, 30, 35),
    'middle_aged': lambda x: trapezoidal(x, 30, 40, 50, 60),
    'old': lambda x: sigmoid(x, 65, 0.2)
}
```



متغیر وزن: (weight_functions)

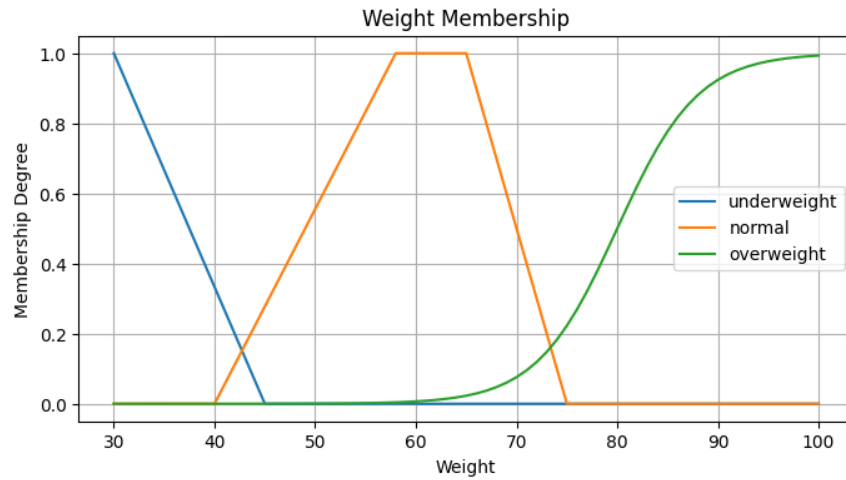
برای مدل‌سازی وزن، سه دسته اصلی در نظر گرفته شده است:

Underweight: برای افرادی که وزن آنها بین ۰ تا ۴۵ کیلوگرم است.

Normal: برای افرادی که وزن آنها بین ۴۰ تا ۷۵ کیلوگرم است.

Overweight: برای افرادی که وزن آنها از ۸۰ کیلوگرم به بالا است و از تابع سیگموئید استفاده می‌شود.

```
weight_functions = {
    'underweight': lambda x: trapezoidal(x, 0, 0, 30, 45),
    'normal': lambda x: trapezoidal(x, 40, 58, 65, 75),
    'overweight': lambda x: sigmoid(x, 80, 0.25)
}
```



متغیر هدف: (goal_functions):

هدف‌های مختلف ورزشی نیز به صورت قطعی مدل‌سازی شده‌اند. این اهداف شامل:

weight_loss: برای کاهش وزن.

Fitness: برای رسیدن به تناسب اندام.

muscle_gain: برای افزایش عضلات.

```
goal_functions = {
    'weight_loss': lambda x: crisp(x, 0),
    'fitness': lambda x: crisp(x, 1),
    'muscle_gain': lambda x: crisp(x, 2)
}
```



متغیر شدت تمرین: (intensity_functions):

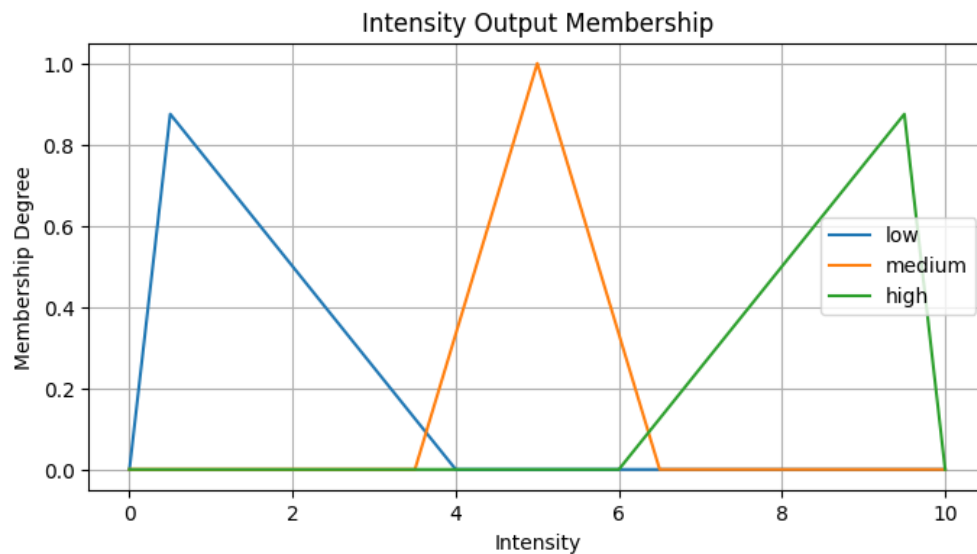
شدت تمرینات به سه سطح مختلف تقسیم شده است:

Low: برای شدت تمرین پایین (از ۰ تا ۴).

Medium: برای شدت تمرین متوسط (از ۳.۵ تا ۶.۵).

High: برای شدت تمرین بالا (از ۶ تا ۱۰).

```
intensity_functions = {  
    'low': lambda x: triangular(x, 0, 0, 4),  
    'medium': lambda x: triangular(x, 3.5, 5, 6.5),  
    'high': lambda x: triangular(x, 6, 10, 10)  
}
```



متغیر مدت زمان تمرین: (duration_functions):

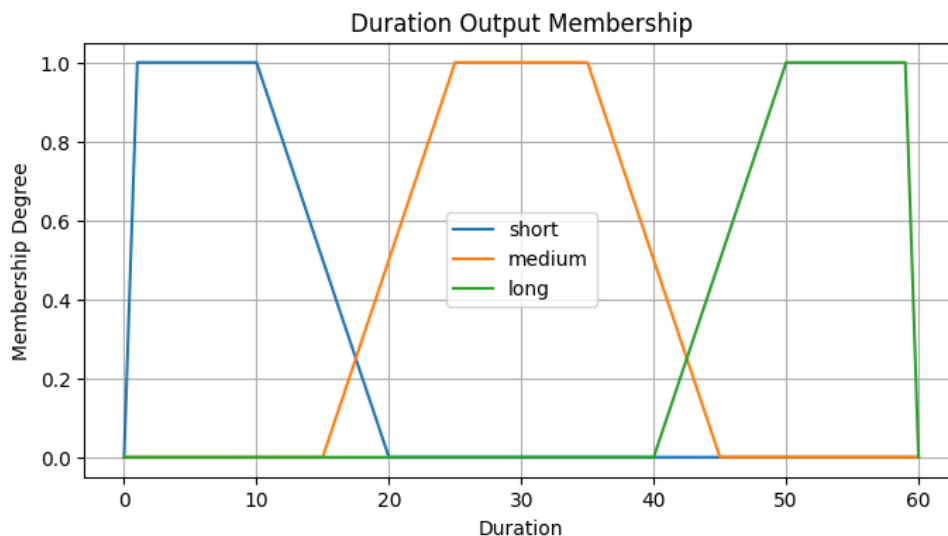
مدت زمان تمرین نیز به سه دسته تقسیم شده است:

Short: برای مدت زمان کوتاه تمرین (از ۰ تا ۲۰ دقیقه).

Medium: برای مدت زمان متوسط تمرین (از ۱۵ تا ۴۵ دقیقه).

Long: برای مدت زمان طولانی تمرین (از ۴۰ تا ۶۰ دقیقه).


```
duration_functions = {
    'short': lambda x: trapezoidal(x, 0, 0, 10, 20),
    'medium': lambda x: trapezoidal(x, 15, 25, 35, 45),
    'long': lambda x: trapezoidal(x, 40, 50, 60, 60)
}
```



۱-۳- وزن دهی به خروجی‌ها:

برای اعمال وزن به خروجی‌ها، دو متغیر دیکشنری مورد نظر قرار گرفته‌اند: شدت تمرین و مدت زمان تمرین. برای شدت تمرین، شدت پایین و متوسط هرکدام وزن برابر ۱.۰ دارند، در حالی که شدت بالا وزن کمتری (۰.۹) دارد. برای مدت زمان تمرین نیز، مدت زمان کوتاه و متوسط وزن برابر ۱.۰ دارند، در حالی که مدت زمان طولانی وزن کمتری (۰.۹) دارد. این وزن‌ها به تنظیم دقیق‌تر توصیه‌ها کمک می‌کنند.

```
INTENSITY_WEIGHTS = {
    'low': 1.0,
    'medium': 1.0,
    'high': 0.9
}

DURATION_WEIGHTS = {
    'short': 1.0,
    'medium': 1.0,
    'long': 0.9
}
```

۱-۴- توضیح قوانین:

در این قسمت، قوانین فازی تعریف شده‌اند تا بر اساس ویژگی‌های فردی، شدت و مدت زمان تمرینات بهینه را محاسبه کنند. این قوانین به تناسب ویژگی‌هایی مانند سن، سطح تناسب اندام، انرژی، وزن و هدف فرد تغییر می‌کنند.

همچنین براساس شرایط سنی و وزن فرد، شدت و مدت زمان تمرین‌ها به کمک وزن‌های تعریف شده کاهش می‌یابد. بدین صورت که اگر سن فرد مسن یا کودک باشد، شدت و مدت زمان تمرین کاهش می‌یابد. همچنین اگر فرد کمبود وزن داشته باشد نیز از شدت و مدت تمرین او کاسته می‌شود. در ادامه تمایلات فرد در مورد مدت زمان تمرین خود نیز در نظر گرفته می‌شود.

```
• def fuzzy_rules(fitness, energy, goal, age, weight, duration_wanted):
    intensity_result = {}
    duration_result = {}
    intensity_weights = INTENSITY_WEIGHTS.copy()
    duration_weights = DURATION_WEIGHTS.copy()

    if age['old'] > 0.4 or age['kid'] > 0.5:
        intensity_weights['high'] = 0.8
        duration_weights['long'] = 0.9
    if weight['underweight'] > 0.5 and age['kid'] < 0.2:
        intensity_weights['high'] = 0.8
        duration_weights['long'] = 0.8
    if duration_wanted == 0:
        duration_weights['medium'] = 0.9
        duration_weights['long'] = 0.7
    elif duration_wanted == 1:
        duration_weights['short'] = 0.9
        duration_weights['long'] = 0.9
    elif duration_wanted == 2:
        duration_weights['short'] = 0.8
        duration_weights['medium'] = 0.9
```

در ادامه با توجه به شرایط افراد قوانینی ارائه شده است:

آمادگی جسمانی در سطح مبتدی و انرژی کم: در این حالت، شدت تمرین و مدت زمان تمرین به شدت پایین و کوتاه تنظیم می‌شود.

```
# Beginner fitness & low energy → low intensity and short duration
if fitness['beginner'] > 0 and energy['low'] > 0:
    intensity_result['low'] = max(
        intensity_result.get('low', 0),
        min(fitness['beginner'], energy['low'])
    )
```

```

duration_result['short'] = max(
    duration_result.get('short', 0),
    min(fitness['beginner'], energy['low'])
)

```

آمادگی جسمانی در سطح مبتدی و انرژی زیاد: در این حالت، شدت تمرین پایین و مدت زمان تمرین در حد متوسط تنظیم می‌شود.

```

# Beginner fitness & high energy → low intensity and medium duration
if fitness['beginner'] > 0 and energy['high'] > 0:
    intensity_result['low'] = max(
        intensity_result.get('low', 0),
        min(fitness['beginner'], energy['high'])
    )
    duration_result['medium'] = max(
        duration_result.get('medium', 0),
        min(fitness['beginner'], energy['high'])
    )

```

آمادگی جسمانی در سطح پیشرفته و انرژی بالا: در این حالت، شدت تمرین و مدت زمان تمرین زیاد و طولانی تنظیم می‌شود.

```

# Advanced fitness & high energy → high intensity and long duration
if fitness['advanced'] > 0 and energy['high'] > 0:
    intensity_result['high'] = max(
        intensity_result.get('high', 0),
        min(fitness['advanced'], energy['high'])
    )
    duration_result['long'] = max(
        duration_result.get('long', 0),
        min(fitness['advanced'], energy['high'])
    )

```

آمادگی جسمانی در سطح پیشرفته و انرژی کم: در این حالت، شدت تمرین متوسط و مدت زمان تمرین کوتاه تنظیم می‌شود.

```

# Advanced fitness but low energy → medium intensity and short duration
if fitness['advanced'] > 0 and energy['low'] > 0:
    intensity_result['medium'] = max(
        intensity_result.get('medium', 0),
        min(fitness['advanced'], energy['low'])
    )
    duration_result['short'] = max(
        duration_result.get('short', 0),
        min(fitness['advanced'], energy['low'])
    )

```

آمادگی جسمانی و سطح انرژی متوسط: در این حالت، شدت تمرین و مدت زمان تمرین نیز متوسط تنظیم می‌شود.

```

# 'intermediate' fitness and 'medium' energy → medium intensity and 'medium' duration
if fitness['intermediate'] > 0 and energy['medium'] > 0:

```

```

intensity_result['medium'] = max(
    intensity_result.get('medium', 0),
    min(fitness['intermediate'], energy['medium']))
)
duration_result['medium'] = max(
    duration_result.get('medium', 0),
    min(fitness['intermediate'], energy['medium']))
)

```

اگر آمادگی جسمانی در سطح متوسط و هدف عضله سازی باشد : در این حالت، شدت تمرین متوسط

تنظیم می شود.

```

# Goal is muscle gain & fitness is intermediate → medium intensity
if goal['muscle_gain'] > 0 and fitness['intermediate'] > 0:
    intensity_result['medium'] = max(
        intensity_result.get('medium', 0),
        min(goal['muscle_gain'], fitness['intermediate']))
)

```

اگر آمادگی جسمانی در سطح متوسط و هدف کم کردن وزن باشد : در این حالت، شدت تمرین بالا

تنظیم می شود.

```

# Goal is weight_loss & fitness is intermediate → high intensity
if goal['weight_loss'] > 0 and fitness['intermediate'] > 0:
    intensity_result['high'] = max(
        intensity_result.get('high', 0),
        min(goal['weight_loss'], fitness['intermediate']))
)

```

اگر آمادگی جسمانی در سطح متوسط و هدف تناسب اندام باشد : در این حالت، شدت تمرین متوسط

تنظیم می شود.

```

# Goal is fitness & fitness is intermediate → medium intensity
if goal['fitness'] > 0 and fitness['intermediate'] > 0:
    intensity_result['medium'] = max(
        intensity_result.get('medium', 0),
        min(goal['fitness'], fitness['intermediate']))
)

```

اگر شخص پیر باشد با هر هدفی: شدت تمرین و مدت زمان تمرین به مقدار کم تنظیم می شود.

```

# Elderly with any goal → reduce intensity and duration
if age['old'] > 0.4:
    intensity_result['low'] = max(
        intensity_result.get('low', 0),
        age['old'] * 0.5
    )
    duration_result['short'] = max(
        duration_result.get('short', 0),
        age['old'] * 0.7
    )
)

```

اگر شخص کودک باشد با هر هدفی: شدت تمرین و مدت زمان تمرین به مقدار کم تنظیم می‌شود.

```
# kid with any goal → reduce intensity and duration
if age['kid'] > 0.4:
    intensity_result['low'] = max(
        intensity_result.get('low', 0),
        age['kid'] * 0.7
    )
    duration_result['short'] = max(
        duration_result.get('short', 0),
        age['kid'] * 0.8
    )
```

اگر آمادگی جسمانی شخص پیشرفته ولی آن فرد پیر باشد: در این حالت، شدت تمرین متوسط و مدت زمان تمرین کوتاه تنظیم می‌شود.

```
# Rule 13: 'advanced' fitness & old → 'medium' intensity and short duration
if fitness['advanced'] > 0 and age['old'] > 0:
    intensity_result['medium'] = max(
        intensity_result.get('medium', 0),
        min(fitness['advanced'], age['old'])
    )
    duration_result['short'] = max(
        duration_result.get('short', 0),
        min(fitness['advanced'], age['old'])
    )
```

اگر شخص اضافه وزن دارد و هدف کم کردن وزن باشد: در این حالت، شدت تمرین بالا و مدت زمان تمرین طولانی تنظیم می‌شود.

```
# Goal is weight_loss & weight is overweight → high intensity and long duration
if goal['weight_loss'] > 0 and weight['overweight'] > 0:
    intensity_result['high'] = max(
        intensity_result.get('high', 0),
        min(goal['weight_loss'], weight['overweight'])
    )
    duration_result['long'] = max(
        duration_result.get('long', 0),
        min(goal['weight_loss'], weight['overweight'])
    )
```

اگر شخص کمبود وزن دارد و هدف کم کردن وزن باشد: در این حالت، شدت تمرین کم و مدت زمان تمرین نیز کم تنظیم می‌شود.

```
# Goal is weight_loss & weight is 'underweight' → low intensity and short duration
if goal['weight_loss'] > 0 and weight['underweight'] > 0:
    intensity_result['low'] = max(
        intensity_result.get('low', 0),
        min(goal['weight_loss'], weight['underweight'])
    )
    duration_result['short'] = max(
```

```

        duration_result.get('short', 0),
        min(goal['weight_loss'], weight['underweight']))
    )

```

اگر شخص وزن طبیعی دارد و هدف تناسب اندام باشد: در این حالت مدت زمان تمرین متوسط تنظیم

می‌شود.

```

# Normal weight & goal is fitness → 'medium' intensity and medium duration
if weight['normal'] > 0 and goal['fitness'] > 0:
    duration_result['medium'] = max(
        duration_result.get('medium', 0),
        min(weight['normal'], goal['fitness']))
    )

```

اگر شخص اضافه وزن دارد و هدف ساخت عضله باشد: در این حالت شدت تمرین زیاد و مدت زمان

تمرین کم تنظیم می‌شود.

```

# 'overweight' and muscle gain → 'high' intensity and short duration
if weight['overweight'] > 0 and goal['muscle_gain'] > 0:
    intensity_result['high'] = max(
        intensity_result.get('high', 0),
        min(goal['muscle_gain'], weight['overweight']))
    )
    duration_result['short'] = max(
        duration_result.get('short', 0),
        min(weight['overweight'], goal['muscle_gain']))
    )

```

اگر شخص کمبود وزن و انرژی پایین دارد و هدف ساخت عضله باشد: در این حالت شدت تمرین و مدت

زمان تمرین متوسط تنظیم می‌شود.

```

# Underweight + muscle gain → medium intensity & duration
if weight['underweight'] > 0 and goal['muscle_gain'] > 0 and energy['low'] > 0:
    intensity_result['medium'] = max(
        intensity_result.get('medium', 0),
        min(weight['underweight'], goal['muscle_gain'], energy['low']))
    )
    duration_result['medium'] = max(
        duration_result.get('medium', 0),
        min(weight['underweight'], goal['muscle_gain'], energy['low']))
    )

```

در نهایت وزن‌های تعریف شده در قسمت قبل به خروجی اعمال می‌شوند:

```

for label in intensity_result:
    intensity_result[label] *= intensity_weights.get(label, 1.0)

for label in duration_result:
    duration_result[label] *= duration_weights.get(label, 1.0)

```

۱-۴- فازی سازی و فازی زدایی:

تابع fuzzify:

این تابع برای تبدیل ورودی‌های دقیق (عددی) به مقادیر فازی استفاده می‌شود. ورودی‌ها از طریق دیکشنری توابع فازی (func_dict) برای هر برچسب (label) مانند "beginner" یا "low" پردازش می‌شوند. در اینجا، تابع فازی مرتبط با هر برچسب از دیکشنری تابع به ورودی اعمال می‌شود و نتیجه به صورت یک دیکشنری از برچسب‌ها و مقادیر فازی برگشت داده می‌شود.

```
def fuzzify(x, func_dict):  
    return {label: func(x) for label, func in func_dict.items() }
```

تابع defuzzify:

یک روش Defuzzification به کار می‌برد تا از مقادیر عضویت فازی (Fuzzy Memberships) یک مقدار دقیق برای خروجی به دست آورد. این روش معمولاً در سیستم‌های فازی برای تبدیل نتایج فازی به مقادیر عددی دقیق استفاده می‌شود.

روش کار این تابع به اینصورت است که برای هر نقطه x در دامنه خروجی (output_range)، مقدار عضویت آن در توابع مختلف محاسبه می‌شود. این مقادیر برای هر برچسب فازی (low, medium, high) محاسبه می‌شود و سپس حداقل مقدار عضویت (min) از بین مقادیر مختلف انتخاب می‌شود. در صورتی که مقادیر عضویت برای یک نقطه وجود نداشته باشند، مقدار آن ۰ فرض می‌شود.

در ادامه متغیر numerator که مجموع نقاط x ضرب شده در مقادیر عضویت (μ) را نشان می‌دهد و متغیر denominator که مجموع مقادیر عضویت (μ) را مشخص می‌کند، محاسبه شده و با تقسیم آن‌ها مرکز ثقل محاسبه می‌شود.

```
def defuzzify(output_membership, output_range, functions):  
    numerator = 0.0  
    denominator = 0.0  
    for x in output_range:  
        memberships = [min(value, functions[label](x)) for label, value in  
            output_membership.items()]  
        mu = max(memberships) if memberships else 0  
        numerator += x * mu  
        denominator += mu  
    return numerator / denominator if denominator != 0 else 0
```

۲-نتایج:

در ادامه ۳ خروجی از برنامه خواهیم داشت:

ورودی اول:

Enter your personal information:

Fitness level (0 to 10): 1

Energy level (0 to 10): 2

Choose your goal: [0] Weight Loss, [1] Fitness, [2] Muscle Gain: 1

Age: 20

Weight: 70

Choose duration: [0] short, [1] medium, [2] long 1

شخصی جوان با وزن ۷۰ و تناسب اندام ضعیف و انرژی پایین با هدف تناسب اندام خروجی زیر را خواهیم داشت.

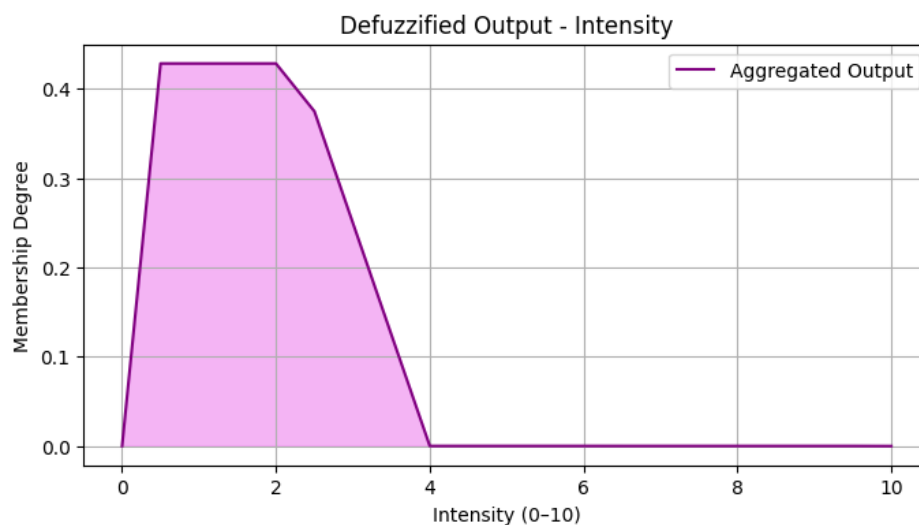
Final Recommendation:

Intensity: 1.73 / 10

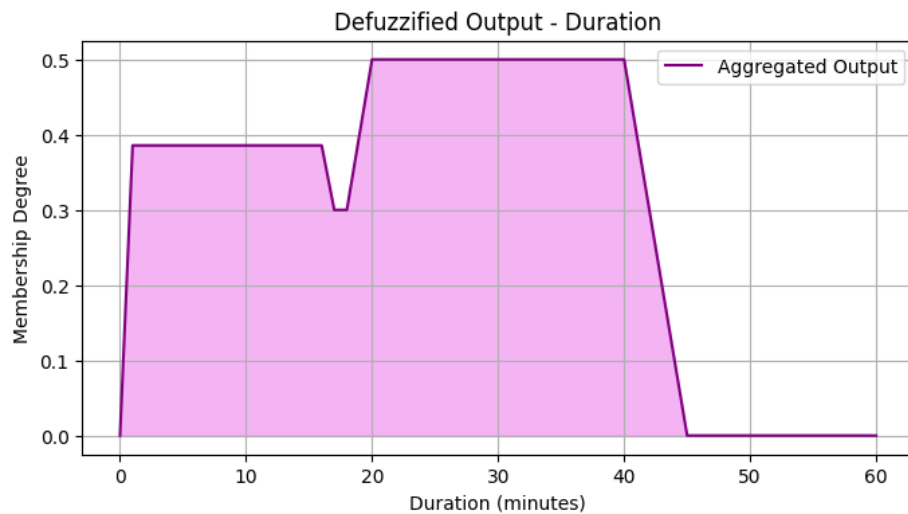
Duration: 22.9 minutes

مقادیر مشخص شده مراکز ثقل شدت تمرین و مدت زمان تمرین می باشند. در ادامه نمودار فازی خروجی ارائه می گردد.

نمودار فازی شدت تمرین:



نمودار فازی مدت تمرین:



ورودی دوم:

Enter your personal information:

Fitness level (0 to 10): 9

Energy level (0 to 10): 9

Choose your goal: [0] Weight Loss, [1] Fitness, [2] Muscle Gain: 0

Age: 79

Weight: 95

Choose duration: [0] short, [1] medium, [2] long 1

شخصی پیر با وزن ۹۵ و تناسب اندام خوب و انرژی بالا با هدف کم کردن وزن خروجی زیر را خواهیم داشت.

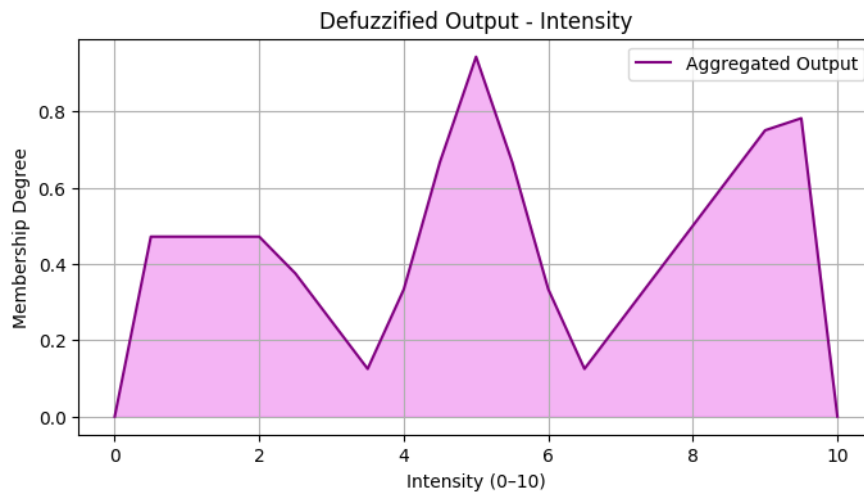
Final Recommendation:

Intensity: 5.35 / 10

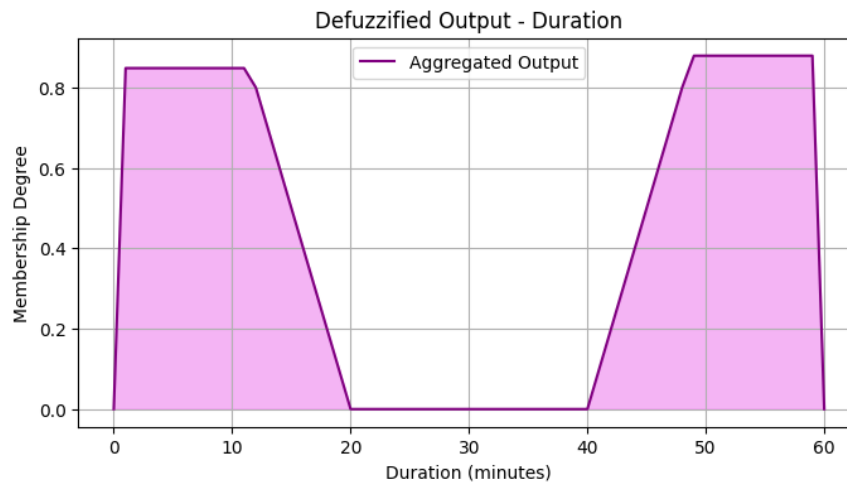
Duration: 30.31 minutes

مقادیر مشخص شده مراکز ثقل شدت تمرین و مدت زمان تمرین می باشند. در ادامه نمودار فازی خروجی ارائه می گردد.

نمودار فازی شدت تمرین:



نمودار فازی مدت تمرین:



ورودی سوم:

Enter your personal information:

Fitness level (0 to 10): 5

Energy level (0 to 10): 6

Choose your goal: [0] Weight Loss, [1] Fitness, [2] Muscle Gain: 1

Age: 46

Weight: 74

Choose duration: [0] short, [1] medium, [2] long 1

شخصی میانسال با وزن ۷۴ و تناسب اندام و انرژی متوسط با هدف تناسب اندام خروجی زیر را خواهیم داشت.

Final Recommendation:

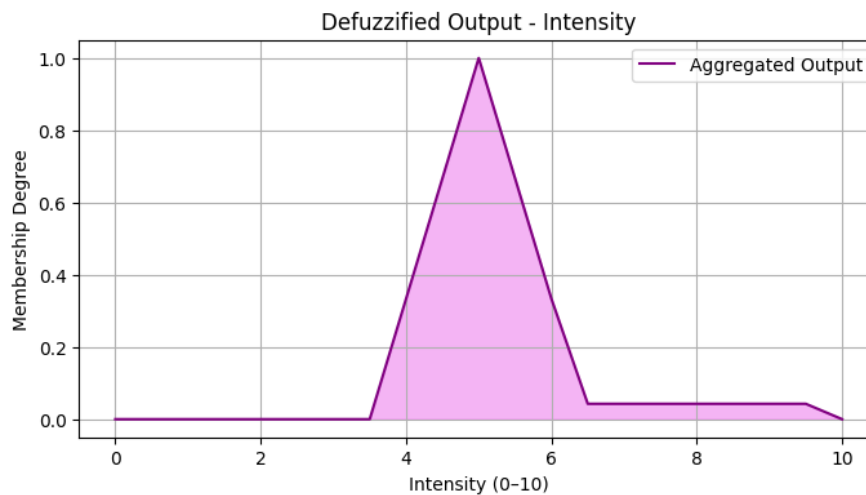
Intensity: 5.27 / 10

Duration: 31.98 minutes

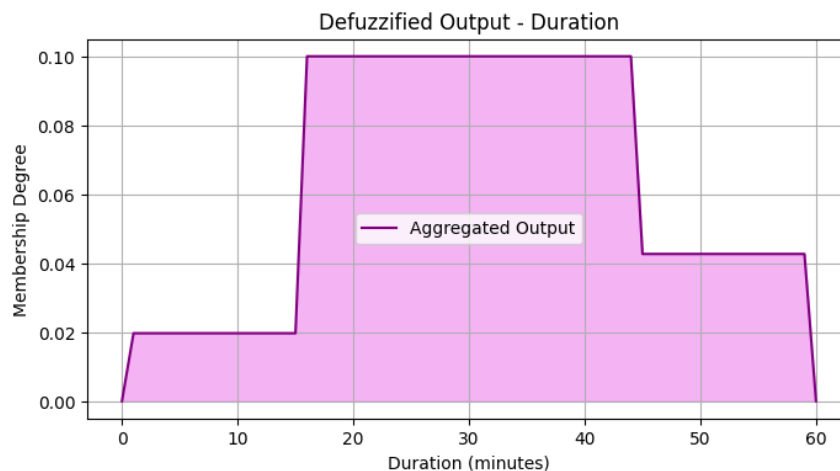
مقادیر مشخص شده مراکز ثقل شدت تمرین و مدت زمان تمرین می باشند. در ادامه نمودار فازی خروجی

ارائه می گردد.

نمودار فازی شدت تمرین:



نمودار فازی مدت تمرین:



لینک کد:

<https://colab.research.google.com/drive/1DTncY0zF4xun3h5Kpak66V6Qpr0DzIgS?usp=sharing#scrollTo=wq4x08fdhhJn>