

# پیاده سازی یک سیستم فازی به منظور کنترل یک مساله با ورودیهای زبانی

سروش آریانا - ۹۷۱۳۰۳۴

دکتر مهدی قطعی

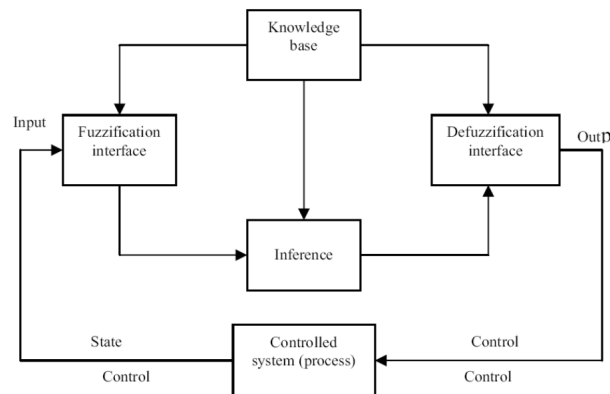
۲۰ تیر ۱۴۰۰

## چکیده

در حالی که همگان از قدرت ماشین ها در انجام محاسبات پیچیده مطلع هستند، پیچیدگی ذهن انسان اما هنوز در مقایسه با آن در هاله ای از ابهام قرار دارد. تحلیل بسیاری از امور پیش پا افتاده که برای انسان ها بسیار ساده است گاه میتواند برای ماشین ها بسیار چالش برانگیز باشد. از این رو دانشمندان حوزه علوم رایانه همواره در تلاش بوده اند که پدیده ها را نزدیک به ادراک انسان و در عین حال به شیوه ای قابل درک برای ماشین ها مدل سازی کنند. ظهور متودولوژی های فازی را میتوان یکی از این تلاش ها به حساب آورد. الگوریتم های فازی همان چیزی هستند که میتوانند ادراک ماشین ها را از دامنه صفر و یک، به بازه ای از اعداد صفر تا یک گسترش دهند تا استدلال های غیر قطعی در تصمیمات را برای رایانه ها مدل سازی کنند تا تفکر آن ها را یک گام دیگر به پیچیدگی ذهن انسان نزدیک کند.

حال ما در اینجا سعی داریم تا ضمن توصیف سیستم های کنترل فازی، مثالی برای نحوه کارکرد چنین سیستمی ارائه دهیم.

# ۱ معرفی سیستم فازی



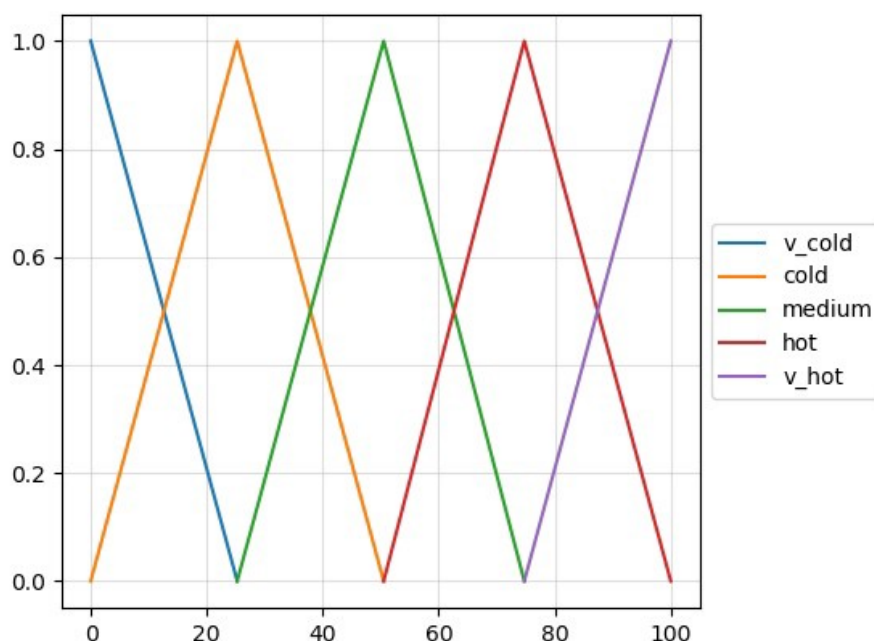
تصویر ۱: سیکل حل مسئله با روش فازی در کنترلر فازی

## ۱.۱ مراحل طراحی

ابتدا باید بیان کلی از مسئله داشته باشیم، از این رو قصد داریم مرحله‌ای که در ادامه به پیاده سازی آن‌ها می‌پردازیم را دوره کنیم.

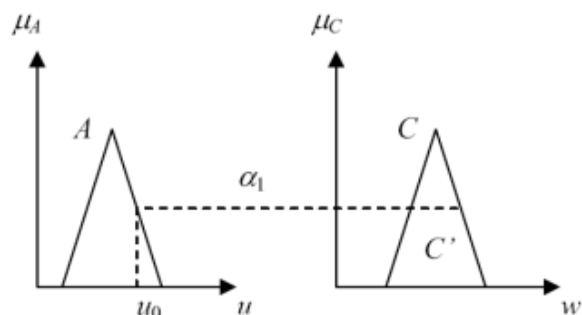
۱. معرفی ورودی‌ها و خروجی‌های مسئله در آغاز کار به معرفی متغیرهای ورودی مسئله می‌پردازیم و مقادیر ممکن برای آن را بیان می‌کنیم.

۲. فازی کردن متغیرهای ورودی در ادامه لازم است که بر روی دامنه متغیرهای خود چند حالت گسسته را در نظر بگیریم و برای آنها یک ضریب فازی موجودیت تعریف کنیم. برای اینکار از توابع عضویت استفاده می‌کنیم. این توابع می‌تواند با توجه به تعریف ما مثلی یا ذوزنقه ای و یا هر شکل دیگری باشد.



تصویر ۲: نمونه ای از روش فازی کردن و گسسته سازی مقادیر ورودی. دامنه این نمودار ها دامنه متغیر ما را بیان میکند و برد بیانگر مقدار تابع عضویت است. هر رنگ بیانگر مقادیر عضویت در یک دسته است.

۳. تعریف قوانین ارجاع و پایگاه دانش سپس باید تعریف کنیم که چه قوانین استدلالی در مدل ما موجود است. قوانین ارجاعی مجموعه گزاره هایی به شکل ”اگر... آنگاه” هستند که ما از ان ها برای رسیدن به نتایج جدید استفاده میکنیم. سیاست هایی که میتوانیم برای بدست آورد مقادیر خروجی توسط قوانین ارجاع استفاده کنیم میتواند روش ممدانی یا لارسن باشد. دقت کنید که قوانین ارجاع و پایگاه دانش ما باید کامل باشد. یعنی هیچ مقدار ورودی ای در دامنه ممکن نباشد که به ازای آن در مسئله گیر کنیم و نتوانیم پیشروی کنیم.



The inference result  $C'$  is defined by the following membership function

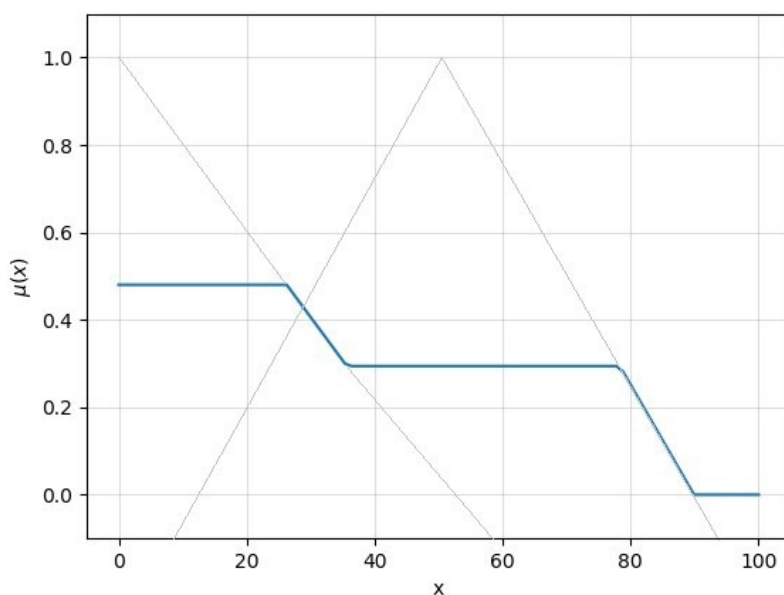
$$\mu_{C'}(w) = \alpha_1 \wedge \mu_C(w) \text{ for } R_C \text{ (Mamdani implication)}$$

$$\mu_{C'}(w) = \alpha_1 \cdot \mu_C(w) \text{ for } R_P \text{ (Larsen implication)}$$

$$\text{where } \alpha_1 = \mu_A(u_0)$$

تصویر ۳: روش های ممدانی و لارسن در ارجاع.

۴. غیر فازی کردن مقادیر خروجی در ادامه میتوانیم بیان فازی متغیرهای خروجی را با کمک روش هایی همچون "ماکزیمم میانگین ها" و "مرکز مساحت" و "نیمساز مساحت" به بیان عددی آن ها بازگردانیم. این روش ها به طور مفصل در جزوه توضیح داده شده اند.



تصویر ۴: استفاده از روش مرکز ثقل در غیر فازی سازی

## ۲ مثال حل شده برای یک کنترل گرافازی

### ۱.۲ توصیف مسأله

مسابقات علمی در رشته های ترکیبی در سطح دبیرستان ها در آستانه برگزاری است، مدیر مدرسه دبیرستان شهید ممدانی باید بتواند از بین ۵۰ شاگرد ممتاز مدرسه افراد مناسب برای شرکت در هر رشته را انتخاب کند. هر کدام از رشته ها نیاز به دانش ترکیبی افراد در زمینه های مختلف ریاضی فیزیک و شیمی را دارد. همچنین محدودیتی برای ارسال دانش آموزان به هر رشته وجود دارد به طوری که از بین افراد مناسب برای هر رشته تنها ۳ نفر از آن ها که بیشترین امتیاز را آورده اند انتخاب میشوند.

نمرات تمام دانش آموزان در سه درس شیمی فیزیک و ریاضی در اختیار مدیر قرار گرفته است و وی میخواهد با توجه به معدل نمرات گذشته افراد در هر درس آنها را قضاوت کرده و برای شرکت در رشته های متناسب با توانایی هایشان برگزیند و ما نیز قصد داریم با طراحی یک سیستم کنترل فازی به او کمک کنیم.

### ۲.۲ متغیر های ورودی (Antecednets)

تمام متغیر های زیر اعدادی اعشاری در بازه ۰ تا ۲۰ و با دقت ۲۵.۰ هستند. مجموعه های فازی آن ها نیز شامل ۳ دسته خوب، متوسط و بد می شود.

۱. معدل نمرات درس ریاضی

۲. معدل نمرات درس فیزیک

۳. معدل نمرات درس شیمی

### ۳.۲ متغیر های خروجی (Consequents)

تمام متغیر های زیر اعدادی اعشاری در بازه ۰ تا ۱۰۰ و با دقت ۱ هستند. مجموعه های فازی آن ها نیز شامل ۳ دسته خوب، متوسط و بد می شود.

۱. سطح توانایی دانشجو برای شرکت در مسابقات علوم کامپیوتر

۲. سطح توانایی دانشجو برای شرکت در مسابقات سازه مستحکم

۳. سطح توانایی دانشجو برای شرکت در مسابقات خودرو سازی

در نهایت در هر رشته سه نفر از آنهایی که بیشترین امتیاز را آورده اند برگزیده میشوند.

## ۴.۲ پیاده سازی

برای پیاده سازی این روش در کامپیوتر از زبان پایتون و کتابخانه scikit-fuzzy استفاده شده است. همچنین به عنوان محیط پیاده سازی از محیط Google Colab استفاده شده است.

### ۱. نصب و Import لایبرری های مورد نیاز به محیط کولب

```
# uncomment in order to install scikit-fuzzy in colab enviroment.
!pip install scikit-fuzzy

Requirement already satisfied: scikit-fuzzy in /usr/local/lib/python3.7/dist-packages (0.4.2)
Requirement already satisfied: numpy>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from scikit-fuzzy) (1.19.5)
Requirement already satisfied: networkx>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from scikit-fuzzy) (2.5.1)
Requirement already satisfied: scipy>=0.9.0 in /usr/local/lib/python3.7/dist-packages (from scikit-fuzzy) (1.4.1)
Requirement already satisfied: decorator<5,>=4.3 in /usr/local/lib/python3.7/dist-packages (from networkx>=1.9.0->scikit-fuzzy) (4.4.2)

[2] import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
```

### تصویر ۵

### ۲. تعریف متغیر های ورودی و خروجی و بازه تعریف آن ها

#### Antecedent/Consequent objects

```
[3] Chemistry = ctrl.Antecedent(np.arange(0, 20.25, 0.25), 'Chemistry')
Physics = ctrl.Antecedent(np.arange(0, 20.25, 0.25), 'Physics')
Math = ctrl.Antecedent(np.arange(0, 20.25, 0.25), 'Math')
```

#### Auto-membership function population

```
[4] Computer_match = ctrl.Consequent(np.arange(0, 101, 1), 'Computer_match')
Construction_match = ctrl.Consequent(np.arange(0, 101, 1), 'Construction_match')
Automotive_match = ctrl.Consequent(np.arange(0, 101, 1), 'Automotive_match')
```

### تصویر ۶

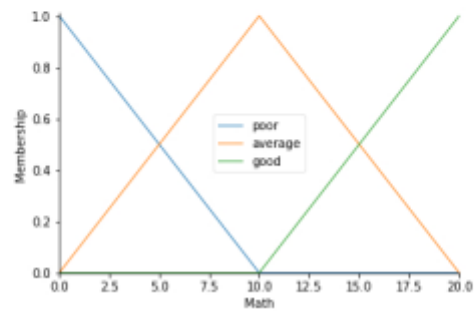
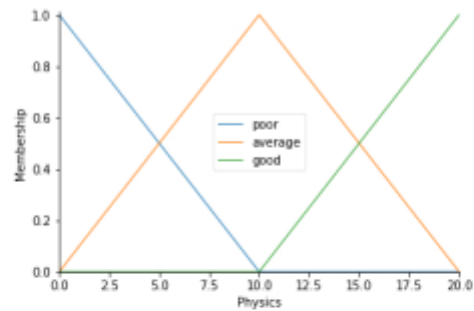
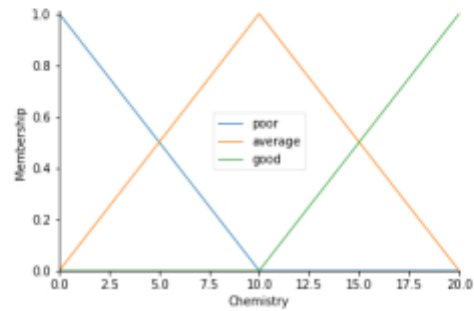
### ۳. تبدیل دامنه متغیر ها به توابع عضویت بر روی متغیر های فازی

## Membership function population

```
[5] Chemistry.automf(3)
Physics.automf(3)
Math.automf(3)

Computer_match.automf(3)
Construction_match.automf(3)
Automotive_match.automf(3)

Chemistry.view()
Physics.view()
Math.view()
```



تصویر ۷: شمای کلی توابع عضویت متغیرهای فازی که به صورت مثلثی استاندارد تعریف شده اند.

۴. تعریف قوانین ارجاع و پایگاه دانش و ساخت سیستم کنترلر

## ▾ Reference rules

```
[6] r1 = ctrl.Rule(Math['good'], Computer_match['good'])
    r2 = ctrl.Rule(Math['poor'], Computer_match['average'])
    r3 = ctrl.Rule(Math['good'], Construction_match['average'])
    r4 = ctrl.Rule(Physics['good'], Construction_match['good'])
    r5 = ctrl.Rule(Chemistry['good'], Construction_match['good'])
    r6 = ctrl.Rule(Physics['good'], Automotive_match['good'])
    match_ctrl = ctrl.ControlSystem([r1, r2, r3, r4, r5, r6])
```

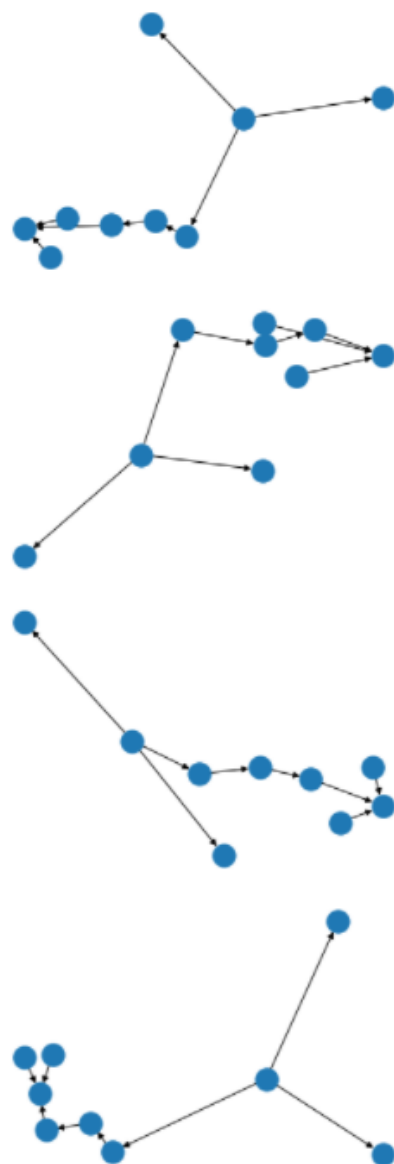
```
[7] r1.view()
    r2.view()
    r3.view()
    r4.view()
    r5.view()
    r6.view()
```

```
(<Figure size 432x288 with 1 Axes>,
 <matplotlib.axes._subplots.AxesSubplot at 0x7eff9fcfc250>)
```

## تصویر ۸

۵. مشاهده شمای کلی از درخت ارجاع متغیرهای خروجی از ورودی





تصویر ۹

۶. وارد کردن یک مثال به عنوان ورودی و بررسی خروجی ها

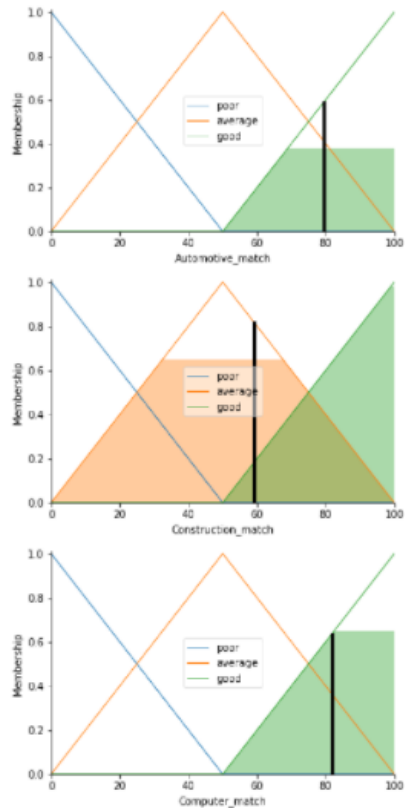
```
[8] match = ctrl.ControlSystemSimulation(match_ctrl)
```

```
match.input['Math'] = 16.5
match.input['Chemistry'] = 19.8
match.input['Physics'] = 13.8
```

```
match.compute()
```

```
[9] print(match.output)
Automotive_match.view(sim=match)
Construction_match.view(sim=match)
Computer_match.view(sim=match)
```

```
OrderedDict([('Computer_match', 81.82098765432103), ('Construction_match', 59.231556225790264), ('Automotive_match', 79.3786008230452)])
```



تصویر ۱۰

۳ منابع

منابع در فایل Source.txt موجود است.