

تمرین چهارم شبکه عصبی

سوال اول

A) شبکه‌های Art زمانی که با داده‌های پر از نویز کار می‌کنند معمولاً عملکرد خوبی ندارند که دلایل آن به شرح زیر است:

1. گم شدن الگوهای اصلی:

وقتی نویز زیادی وجود دارد ویژگی‌های مهم و کلیدی داده‌ها به سختی شناسایی می‌شوند. انگار که یک تصویر را با گرد و غبار ببینید و در این شرایط جزئیات مهم به چشم نمی‌آید.

2. آموزش در شرایط متفاوت:

اگر شبکه با داده‌های تمیز و بدون نویز آموزش دیده باشد، در مواجهه با داده‌های پر از نویز غافلگیر می‌شود و نمی‌تواند به خوبی الگوها را تشخیص دهد. این مثل این است که کسی در محیطی آرام تمرین کند و بعد بخواهد همان کار را در جایی شلوغ و پر سر و صدا انجام دهد.

3. ایجاد اختلال در پردازش:

نویز شبیه یک مزاحم عمل می‌کند و کار شبکه را برای پردازش درست اطلاعات سخت می‌کند. این اختلال می‌تواند باعث شود که شبکه به جای الگوهای واقعی روی بخش‌های تصادفی یا اشتباه تمرکز کند.

4. پیچیدگی بیش از حد:

با افزایش نویز، شبکه مجبور است انرژی بیشتری صرف کند تا از میان این اشتباهات الگوهای مورد نظر را پیدا کند. این مسئله باعث کندی و کاهش دقت آن می‌شود.

B) مسئله Grandmother Node یک مفهوم جالب در دنیای هوش مصنوعی و شبکه‌های عصبی است. این ایده به این اشاره دارد که ممکن است یک نورون یا نود خاص در شبکه مسئول شناسایی یک مفهوم بسیار مشخص و خاص باشد.

حالا چرا این موضوع مشکل ساز است؟

1. عدم توانایی در تعمیم دادن:

اگر شبکه به نودهای خاصی برای شناسایی ویژگی‌های بسیار دقیق وابسته باشد دیگر نمی‌تواند این دانش را به موارد جدید و مواردی که تاکنون ندیده تعمیم دهد. برای مثال اگر نورورنی فقط بتواند عکس شما را در یک عکس خاص شناسایی کند، نمی‌تواند شما را در موقعیت‌های دیگر مثل عکس با نور زیاد یا زاویه‌ی دیگری از شما را تشخیص دهد.

2. بیش از حد خاص بودن:

شبکه‌های عصبی طراحی شده‌اند تا دانش را در بین نودهای مختلف پخش کنند نه اینکه یک نود را مسئول یک مفهوم خاص کنند. اگر مفهومی فقط به یک نود نسبت داده شود شبکه شکننده می‌شود و کوچک‌ترین تغییر در ورودی ممکن است باعث شود مفهوم شناسایی نشود.

3. کاهش انعطاف‌پذیری شبکه:

یکی از دلایل قدرتمند بودن شبکه‌های عصبی این است که یادگیری را در بین نودها تقسیم می‌کنند. اما اگر یک نود به تنهایی مسئول یک مفهوم باشد، شبکه انعطاف‌پذیری خود را از دست می‌دهد و در برابر خرابی آسیب‌پذیر می‌شود.

(C) در محیط‌های RCE دلیل نبود خطا در داده‌های آموزشی و احتمال بروز خطا در داده‌های آزمایشی را می‌توان با توجه به ماهیت کنترل‌شده‌ی آموزش و تنوع طبیعی داده‌های دنیای واقعی توضیح داد.

داده‌های train معمولاً با دقت بسیار زیاد انتخاب، پردازش و پاک‌سازی می‌شوند تا ورودی‌هایی با کیفیت بالا و کاملاً سازگار به دست آید. این فرآیند باعث می‌شود که نویز، ابهامات یا ناسازگاری‌هایی که می‌توانند مدل را هنگام آموزش دچار مشکل کنند، حذف شوند. به همین دلیل مدل می‌تواند الگوها و روابط داده‌ها را بدون سردرگمی و خطا یاد بگیرد.

داده‌های test معمولاً برای ارزیابی توانایی مدل در تطبیق و تعمیم طراحی می‌شوند، بنابراین

تنوع، نويز و گاهی داده‌های ناقص يا پر از اشکالاتی که نمایانگر شرایط واقعی هستند، در آنها بیشتر است. این خطاها يا ناسازگاری‌ها می‌توانند ناشی از موارد زیر باشند:

- نويز (مانند تصاویر تار يا دارای اختلال)
- اشتباهات در برچسب‌گذاری (مثلاً برچسب‌های نادرست يا ذهنی).
- تغییرات غیرمنتظره در داده‌ها (مانند سبک‌های هنری جدید يا شرایط متفاوتی که در آموزش وجود نداشته‌اند).
- تفاوت در منبع داده‌ها (مثلاً داده‌های test ممکن است از منابع متفاوتی نسبت به داده‌های train تهیه شده باشند).

سوال 2

(A) این توابع عضویت به خوبی نمایانگر چاقی و جوانی هستند، زیرا با درک شهودی این مفاهیم مطابقت دارند:

تابع چاقی:

برای وزن‌های کوچکتر از 50 مقدار عضویت صفر است، به این معنا که این افراد چاق در نظر گرفته نمی‌شوند.

با افزایش وزن از 50 تا 150، مقدار عضویت به تدریج افزایش می‌یابد، که نشان می‌دهد هرچه وزن بیشتر شود، احتمال چاق در نظر گرفته شدن نیز بیشتر می‌شود.

برای وزن‌های بیشتر از 150 مقدار عضویت به 1 می‌رسد، یعنی این افراد به طور کامل چاق محسوب می‌شوند.

تابع جوانی:

برای سن‌های کوچک تر از 25 مقدار عضویت برابر 1 است، به این معنا که این افراد کاملاً "جوان" در نظر گرفته می‌شوند.

با افزایش سن از 25 تا 100، مقدار عضویت به تدریج کاهش می‌یابد، که نشان‌دهنده این است که "جوان بودن" با افزایش سن کمرنگ‌تر می‌شود.

برای سن های بیشتر از 100 مقدار عضویت صفر است، یعنی دیگر هیچ ارتباطی با "جوان" بودن ندارند.

(B) تعریف توابع عضویت برای مسن، لاغر و میانگین وزن

تابع مسن (پیری) :

$$\mu_{old}(u) = \begin{cases} 0, & u \in (-\infty, 40] \\ \frac{u-40}{30}, & u \in (40, 70] \\ 1, & u \in (70, +\infty) \end{cases}$$

برای سن های کمتر از 40 مقدار عضویت برابر صفر است، چون این افراد مسن در نظر گرفته نمی شوند.

بین سنین 40 تا 70 مقدار عضویت به صورت خطی افزایش می یابد، که نشان می دهد افراد به تدریج مسن در نظر گرفته می شوند.

برای سنین بالاتر از 70 مقدار عضویت برابر 1 است، یعنی این افراد کاملاً مسن محسوب می شوند.

تابع لاغری :

$$\mu_{slim}(u) = \begin{cases} 1, & u \in (-\infty, 50] \\ 1 - \left(\frac{u-50}{30}\right)^2, & u \in (50, 80] \\ 0, & u \in (80, +\infty) \end{cases}$$

برای وزن های کمتر از 50 مقدار عضویت برابر 1 است چون این افراد کاملاً "لاغر" محسوب می شوند.

بین وزن های 50 تا 80 مقدار عضویت به تدریج و به صورت سهمی کاهش می یابد، که نشان دهنده کاهش "لاغر بودن" با افزایش وزن است.

برای وزن های بالاتر از 80 مقدار عضویت صفر است، یعنی این افراد دیگر "لاغر" در نظر گرفته نمی شوند.

تابع میانگین وزن:

$$\mu_{avg_weight}(u) = \begin{cases} 0, & u \in (-\infty, 40] \\ \frac{u-40}{30}, & u \in (40, 70] \\ 1 - \frac{u-70}{30}, & u \in (70, 100] \\ 0, & u \in (100, +\infty) \end{cases}$$

برای وزن‌های کمتر از 40 یا بیشتر از 100 مقدار عضویت برابر صفر است زیرا این مقادیر خارج از محدوده "میانگین وزن" قرار دارند.

بین 40 تا 70 مقدار عضویت به صورت خطی افزایش می‌یابد که نشان‌دهنده وزن‌های پایین‌تر از "متوسط" است.

بین 70 تا 100 مقدار عضویت به صورت خطی کاهش می‌یابد که نشان‌دهنده وزن‌های بالاتر از "متوسط" است.

(C) بررسی صحت جمله زیر:

If the second person is a very fat young person, then they are not relatively slimmer than the first person.

نفر اول :

$$W_1 = 70, a_1 = 60$$

نفر دوم:

$$W_2 = 110, a_2 = 30$$

محاسبه مقادیر عضویت چاقی برای هر فرد:

$$\mu_{fat}(u) = \begin{cases} 0, & u \in (-\infty, 50] \\ 1 - \left(\frac{u-150}{100}\right)^2, & u \in (50, 150] \\ 1, & u \in (150, +\infty) \end{cases} \quad \text{فرمول:}$$

نفر اول:

$$\mu_{fat}(70) = 1 - \left(\frac{70 - 150}{100}\right)^2 = 1 - \left(\frac{-80}{100}\right)^2 = 1 - 0.64 = 0.36$$

نفر دوم:

$$\mu_{fat}(110) = 1 - \left(\frac{110 - 150}{100}\right)^2 = 1 - \left(\frac{-40}{100}\right)^2 = 1 - 0.16 = 0.84$$

محاسبه مقادیر عضویت جوانی برای هر فرد:

$$\mu_{slim}(u) = \begin{cases} 1, & u \in (-\infty, 50] \\ 1 - \left(\frac{u-50}{30}\right)^2, & u \in (50, 80] \\ 0, & u \in (80, +\infty) \end{cases} \quad \text{فرمول:}$$

نفر اول:

$$\mu_{young}(60) = \left(1 + \left(\frac{60 - 25}{5}\right)^2\right)^{-1} = \left(1 + \left(\frac{35}{5}\right)^2\right)^{-1} = (1 + 49)^{-1} = \frac{1}{50} = 0.02$$

نفر دوم:

$$\mu_{young}(30) = \left(1 + \left(\frac{30 - 25}{5}\right)^2\right)^{-1} = \left(1 + \left(\frac{5}{5}\right)^2\right)^{-1} = (1 + 1)^{-1} = \frac{1}{2} = 0.5$$

محاسبه مقادیر عضویت لاغری برای هر فرد:

$$\mu_{slim}(u) = \begin{cases} 1, & u \in (-\infty, 50] \\ 1 - \left(\frac{u-50}{30}\right)^2, & u \in (50, 80] \\ 0, & u \in (80, +\infty) \end{cases} \quad \text{فرمول:}$$

نفر اول:

$$\mu_{slim}(70) = 1 - \left(\frac{70-50}{30}\right)^2 = 1 - \left(\frac{20}{30}\right)^2 = 1 - 0.44 = 0.56$$

نفر دوم:

$$\mu_{slim}(110) = 0 \quad (\text{since } 110 > 80).$$

تفسیر جمله:

If the second person is a very fat young person, then they are not relatively slimmer than the first person.

بخش اول "If the second person is a very fat young person"

مفهوم very یعنی به مقدار آن را به توان 2 برسانیم:

$$\mu_{very\ fat}(w_2) = (\mu_{fat}(110))^2 = (0.84)^2 = 0.7056$$

با عبارت young ترکیب میشود که به معنای AND است ما در اینجا AND را MIN می گیریم:

$$\mu_{very\ fat\ AND\ young}(w_2, a_2) = \min(0.7056, 0.5) = 0.5$$

بخش دوم "They are not relatively slimmer than the first person"

مفهوم Not slim به معنای $1 - \mu_{slim}$ است:

$$\mu_{not\ slim}(w_2) = 1 - \mu_{slim}(110) = 1 - 0 = 1$$

عبارت not slimmer در فازی را به صورت زیر تعریف می کنیم:

$$A = \mu_{very\ fat\ AND\ young}(w_2, a_2) = 0.5$$

$$B = \mu_{not\ slim}(w_2) = 1$$

$$\mu_{final} = \max(1 - 0.5, 1) = \max(0.5, 1) = 1$$

این نتیجه نشان می‌دهد که طبق منطق فازی اگر فرد دوم هم "جوان" و هم "بسیار چاق" باشد، او به طور قطع لاغرتر از فرد اول محسوب نمی‌شود.

سوال 3

مراحل طراحی یک سیستم فازی:

1. تعریف متغیرهای ورودی و خروجی:

- متغیرهای ورودی را مشخص کنید (x_1, x_2, \dots)
- بازه آنها را تعریف کنید $[-1, 1]$
- متغیر خروجی را تعریف کنید $g(x_1, x_2)$

2. فازی‌سازی متغیرهای ورودی:

- برای متغیرهای ورودی، مجموعه‌های فازی تعریف کنید. برای مثال: کم، متوسط، زیاد.
- توابع عضویت (مانند مثلثی یا ذوزنقه‌ای) را برای این مجموعه‌ها انتخاب کنید.

3. تعریف قوانین فازی:

- مجموعه‌ای از قوانین اگر... آنگاه... بنویسید که ورودی‌های فازی را به خروجی فازی تبدیل کنند.
- مثال: اگر x_1 کم باشد و x_2 متوسط باشد، آنگاه خروجی زیاد است.

4. انتخاب روش استنتاج:

- تصمیم بگیرید چگونه قوانین فازی را با هم ترکیب کنید و نتیجه‌گیری انجام دهید (روش‌های ممدانی یا سوگنو).

5. دی‌فازی‌سازی:

- خروجی فازی را به یک عدد قطعی تبدیل کنید (روش مرکز ثقل یا میانگین وزنی).

6. تست و بهبود سیستم:

- عملکرد سیستم را ارزیابی کنید و در صورت نیاز قوانین یا توابع عضویت را اصلاح کنید.

طراحی یک سیستم فازی برای $g(x_1, x_2)$

تابع مورد نظر:

$$g(x_1, x_2) = \frac{1}{3 + x_1 + x_2}$$

متغیرهای ورودی:

$$x_1 \in [-1, 1]$$

$$x_2 \in [-1, 1]$$

متغیر خروجی:

$$g(x_1, x_2)$$

فازی‌سازی ورودی‌ها:

برای هر ورودی x_1 و x_2 سه مجموعه فازی تعریف کنید:

کم: در نزدیکی مقدار پایین مثلا -1

متوسط: در اطراف صفر

زیاد: در نزدیکی مقدار بالا مثلا $1+$

توابع عضویت برای این مجموعه‌ها می‌تواند مثلثی باشد:

$$\mu_{Low}(x) = \max(0, 1 - |x - (-1)|), \quad \mu_{Medium}(x) = \max(0, 1 - |x|), \quad \mu_{High}(x) = \max(0, 1 - |x - 1|)$$

تعریف قوانین فازی:

قوانین فازی را به گونه‌ای تنظیم کنید که تابع $g(x_1, x_2)$ به صورت تقریبی بازسازی شود:

- قانون 1: اگر x_1 کم و x_2 کم باشد، آنگاه $g(x_1, x_2)$ زیاد است.
- قانون 2: اگر x_1 متوسط و x_2 کم باشد آنگاه $g(x_1, x_2)$ متوسط است.

- تعداد کل قوانین: $3 \times 3 = 9$ قانون

استنتاج : (Inference)

از روش Sugeno استفاده می کنیم که هر قانون خروجی ای به شکل خطی یا ثابت تولید می کند

$$g = \frac{\sum w_i z_i}{\sum w_i}$$

دی فازی سازی : (Defuzzification)

با استفاده از روش وزن دهی قوانین، مقدار عددی خروجی را محاسبه کنید

سوال 4

پس از وارد کردن کتابخانه های لازم یک تابع برای ایجاد سیستم فازی ایجاد کردیم

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

def create_fuzzy_system():
    temperature = ctrl.Antecedent(np.arange(-40, 41, 1), 'temperature')
    humidity = ctrl.Antecedent(np.arange(0, 101, 1), 'humidity')
    motor_speed = ctrl.Consequent(np.arange(0, 21, 1), 'motor_speed')
```

برای تعریف ورودی های سیستم فازی (دما و رطوبت) از ctrl.Antecedent استفاده می شود.

دما: (temperature) مقادیر بین -40 تا 40 با گام های ۱ تعریف شده است.

رطوبت: (humidity) مقادیر بین 0 تا 100 با گام های ۱.

برای تعریف خروجی سیستم (سرعت موتور) از ctrl.Consequent استفاده می شود.

سرعت موتور: (motor_speed) مقادیر بین 0 تا 20

```
temperature['low'] = fuzz.trimf(temperature.universe, [-40, -40, 0])
temperature['moderate'] = fuzz.trimf(temperature.universe, [-10, 0, 30])
temperature['high'] = fuzz.trimf(temperature.universe, [20, 40, 40])
```

سه مجموعه برای دما تعریف شده است:

Low: مقدارهای پایین (از -40 شروع و به 0 ختم می شود).

Moderate: مقدارهای متوسط (بین 10 تا 30).

high: مقدارهای بالا (بین 20 تا 40).

```
humidity['low'] = fuzz.trimf(humidity.universe, [0, 0, 40])
humidity['moderate'] = fuzz.trimf(humidity.universe, [30, 50, 70])
humidity['high'] = fuzz.trimf(humidity.universe, [60, 100, 100])
```

مشابه دما، سه مجموعه برای رطوبت تعریف شده است:

Low: مقدارهای پایین (بین 0 و 40).

Moderate: مقدارهای متوسط. (بین 30 و 70)

High: مقدارهای بالا. (بین 60 و 100).

```
motor_speed['low'] = fuzz.trimf(motor_speed.universe, [0, 0, 10])
motor_speed['high'] = fuzz.trimf(motor_speed.universe, [10, 20, 20])
```

دو مجموعه برای سرعت موتور تعریف شده است:

Low: مقدار کم. (بین 0 و 10)

High: مقدار زیاد. (بین 10 و 20)

```
rule1 = ctrl.Rule(temperature['high'] & humidity['low'], motor_speed['low'])
rule2 = ctrl.Rule(temperature['moderate'] & humidity['low'], motor_speed['low'])
rule3 = ctrl.Rule(temperature['low'] & humidity['low'], motor_speed['high'])
rule4 = ctrl.Rule(humidity['moderate'], motor_speed['high'])
rule5 = ctrl.Rule(temperature['low'] & humidity['high'], motor_speed['high'])
rule6 = ctrl.Rule(humidity['high'], motor_speed['high'])
```

این قوانین رفتار سیستم را مشخص می‌کنند:

- اگر دما بالا و رطوبت کم باشد، سرعت موتور کم باشد.
- اگر دما متوسط و رطوبت کم باشد، سرعت موتور کم باشد.
- اگر دما پایین و رطوبت کم باشد، سرعت موتور زیاد باشد.
- اگر رطوبت متوسط باشد، سرعت موتور زیاد باشد.

- اگر دما پایین و رطوبت زیاد باشد، سرعت موتور **زیاد** باشد.
- اگر رطوبت زیاد باشد، سرعت موتور **زیاد** باشد.

```
control_system = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6])
return control_system
```

در ادامه یک سیستم کنترلی ایجاد می کنیم.

با استفاده از `ctrl.ControlSystemSimulation` یک شی شبیه ساز از سیستم کنترلی ساخته شده ایجاد می کنیم که وظیفه دارد که با دریافت مقادیر ورودی (دما و رطوبت) قوانین فازی را اعمال کند و خروجی (سرعت موتور) را محاسبه نماید.

```
def get_motor_speed(control_system, temp, hum):
    simulator = ctrl.ControlSystemSimulation(control_system)
    simulator.input['temperature'] = temp
    simulator.input['humidity'] = hum
    simulator.compute()
    return simulator.output['motor_speed']
```

ورودی ها : دما و رطوبت به شبیه ساز داده می شوند.

`simulator.compute()`: سیستم را بر اساس قوانین ارزیابی می کند.

خروجی : سرعت موتور از طریق `simulator.output['motor_speed']` بازایی می شود.

```

control_system = create_fuzzy_system()

print("\ntest 1:")
temp, hum = 20, 90
print(f"Temperature: {temp}°C, Humidity: {hum}%")
motor_speed = get_motor_speed(control_system, temp, hum)
print(f"Motor Speed: {motor_speed:.2f}")

test_cases = [
    (-30, 20),
    (0, 50),
    (35, 30),
]

for i, (temp, hum) in enumerate(test_cases, 2):
    print(f"\ntest {i}:")
    print(f"Temperature: {temp}°C, Humidity: {hum}%")
    motor_speed = get_motor_speed(control_system, temp, hum)
    print(f"Motor Speed: {motor_speed:.2f}")

```

دمای ورودی و رطوبت با استفاده از تابع `get_motor_speed` مورد آزمایش قرار می‌گیرند.
 هر ورودی براساس قوانین سیستم فازی ارزیابی شده و سرعت موتور محاسبه می‌شود.

نتیجه نهایی:

```

test 1:
Temperature: 20°C, Humidity: 90%
Motor Speed: 16.50

test 2:
Temperature: -30°C, Humidity: 20%
Motor Speed: 16.11

test 3:
Temperature: 0°C, Humidity: 50%
Motor Speed: 16.67

test 4:
Temperature: 35°C, Humidity: 30%
Motor Speed: 4.40

```