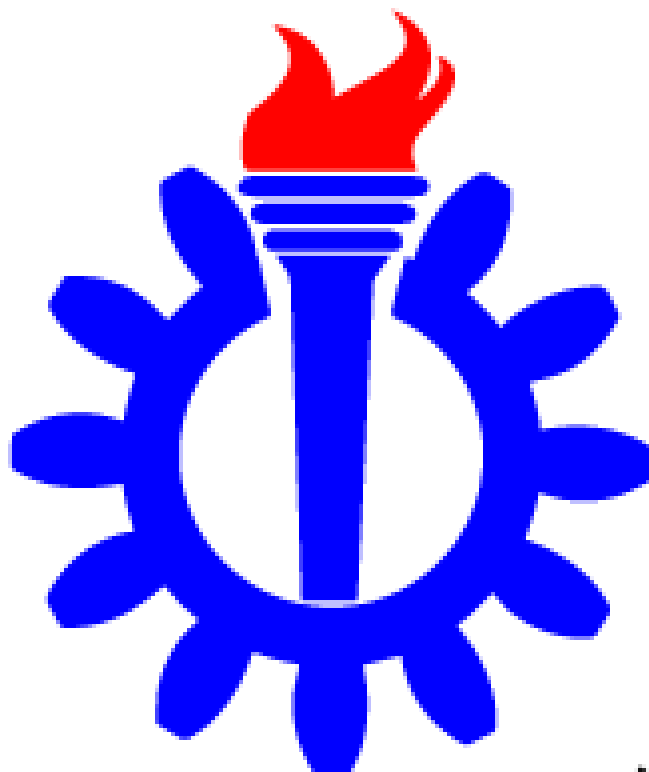


بسم الله الرحمن الرحيم



دانشگاه علم و صنعت ایران

مصدر عرفان زارع زردینر

تدریس سرر 3 در سرهوشربا سبتر

# سوال 1

```
import numpy as np
import matplotlib.pyplot as plt

class hopfield:
    def __init__(self, dimens):
        self.weights = np.zeros((dimens, dimens))
        self.stablebool=False
        self.num=0

    def savedata(self, list_inps):
        list_inps = np.array(list_inps)
        listinps_rshp=list_inps.reshape(-1, 1)
        nweight = list_inps * listinps_rshp
        np.fill_diagonal(nweight, 0)
        self.weights += nweight
        print("Weight when its updated is : ", list_inps, " is changed to :\n", self.weights)

    def survey(self, inps_1):
        inps_1 = np.array(inps_1)
        sign_inps_1 = np.sign(np.dot(self.weights, inps_1))
        self.stablebool=(sign_inps_1 == inps_1).all()
        if (self.stablebool):
            print(inps_1, "stable")
        else:
            print(inps_1, "not stable, The closest stable data that saved is: ", sign_inps_1)

    def inps_stablizer(self, inps_1):
        inps_1 = np.array(inps_1)
        inps_1_r = inps_1.reshape(-1, 1)
        while(self.stablebool != True):
            nweight = inps_1_r * inps_1
            np.fill_diagonal(nweight, 0)
            self.weights += nweight
            input = np.array(inps_1)
            input_sign = np.sign(np.dot(self.weights, input))
            self.stablebool = (input_sign == input).all()
            self.num += 1
        print(f"{self.num} times this while worked to inps_stablizer the input")
        print(self.weights)
```

در حل این سوال ابتدا کلاس هاپفیلد تعریف می کنیم و با توجه به داده و خواسته سوال ما آن را عضوی میگذاریم. حال ماتریس  $8 \times 8$  برای وزن ها تعریف می نماییم. در تابع

**savedata** یادگیری انجام می شود و وزن بدست آمده و به صورت قرینه آن نیز ذخیره

$$w(ij) = \sum_{k=1}^p x(i)^k \cdot x(j)^k$$

می شود.

در ادامه بررسی کردیم آیا در شبکه ورودیمان هست یا خیر که با خود و معکوس آن مقایسه نمودیم و در صورت عدم وجود شبیه آن را برگردانیدیم. تا به آخر هم برای پایداری هست و می اییم در آن یادگیری را تا پایدار شدن انجام میدهم تا وزن ها درست شوند.

(الف)

```
Weight when its updated is : [-1 -1 1 -1 1 -1 -1 1] is changed to :
[[ 0.  1. -1.  1. -1.  1.  1. -1.]
 [ 1.  0. -1.  1. -1.  1.  1. -1.]
 [-1. -1.  0. -1.  1. -1. -1.  1.]
 [ 1.  1. -1.  0. -1.  1.  1. -1.]
 [-1. -1.  1. -1.  0. -1. -1.  1.]
 [ 1.  1. -1.  1. -1.  0.  1. -1.]
 [ 1.  1. -1.  1. -1.  1.  0. -1.]
 [-1. -1.  1. -1.  1. -1. -1.  0.]]
Weight when its updated is : [-1 -1 -1 -1 -1 1 -1 -1] is changed to :
[[ 0.  2.  0.  2.  0.  0.  2.  0.]
 [ 2.  0.  0.  2.  0.  0.  2.  0.]
 [ 0.  0.  0.  0.  2. -2.  0.  2.]
 [ 2.  2.  0.  0.  0.  0.  2.  0.]
 [ 0.  0.  2.  0.  0. -2.  0.  2.]
 [ 0.  0. -2.  0. -2.  0.  0. -2.]
 [ 2.  2.  0.  2.  0.  0.  0.  0.]
 [ 0.  0.  2.  0.  2. -2.  0.  0.]]
Weight when its updated is : [-1 1 1 -1 -1 1 -1 1] is changed to :
[[ 0.  1. -1.  3.  1. -1.  3. -1.]
 [ 1.  0.  1.  1. -1.  1.  1.  1.]
 [-1.  1.  0. -1.  1. -1. -1.  3.]
 [ 3.  1. -1.  0.  1. -1.  3. -1.]
 [ 1. -1.  1.  1.  0. -3.  1.  1.]
 [-1.  1. -1. -1. -3.  0. -1. -1.]
 [ 3.  1. -1.  3.  1. -1.  0. -1.]
 [-1.  1.  3. -1.  1. -1. -1.  0.]]
[-1 -1 1 -1 1 -1 -1 1] stable
[-1 -1 -1 -1 -1 1 -1 -1] stable
[-1 1 1 -1 -1 1 -1 1] stable
```

(ب)

```

[ 1 -1 1 -1 1 -1 -1 1] not stable, The closest stable data that saved is: [-1. -1.  1. -1.  1. -1. -1.  1.]
[ 1 1 -1 -1 -1 1 -1 -1] not stable, The closest stable data that saved is: [-1. -1. -1.  1. -1.  1.  1. -1.]
[ 1 1 1 -1 1 1 -1 1] not stable, The closest stable data that saved is: [-1.  1.  1. -1. -1. -1. -1.  1.]
[ 1 -1 1 -1 1 -1 -1 1] not stable, The closest stable data that saved is: [-1. -1.  1. -1.  1. -1. -1.  1.]
[ 1 1 -1 -1 -1 1 -1 -1] not stable, The closest stable data that saved is: [-1. -1. -1.  1. -1.  1.  1. -1.]
[ 1 1 1 -1 1 1 -1 1] not stable, The closest stable data that saved is: [-1.  1.  1. -1. -1. -1. -1.  1.]

```

با توجه به عدم پایداری، تابع پایدارکننده برآشون صدا زده می شود.

```

2 times this while worked to inps_stablizer the input
[[ 0. -1.  1.  1.  3. -3.  1.  1.]
 [-1.  0. -1.  3. -3.  3.  3. -1.]
 [ 1. -1.  0. -3.  3. -3. -3.  5.]
 [ 1.  3. -3.  0. -1.  1.  5. -3.]
 [ 3. -3.  3. -1.  0. -5. -1.  3.]
 [-3.  3. -3.  1. -5.  0.  1. -3.]
 [ 1.  3. -3.  5. -1.  1.  0. -3.]
 [ 1. -1.  5. -3.  3. -3. -3.  0.]]
[ 1 -1  1 -1  1 -1 -1  1] stable
2 times this while worked to inps_stablizer the input
[[ 0. -1.  1.  1.  3. -3.  1.  1.]
 [-1.  0. -1.  3. -3.  3.  3. -1.]
 [ 1. -1.  0. -3.  3. -3. -3.  5.]
 [ 1.  3. -3.  0. -1.  1.  5. -3.]
 [ 3. -3.  3. -1.  0. -5. -1.  3.]
 [-3.  3. -3.  1. -5.  0.  1. -3.]
 [ 1.  3. -3.  5. -1.  1.  0. -3.]
 [ 1. -1.  5. -3.  3. -3. -3.  0.]]
[ 1  1 -1 -1 -1  1 -1 -1] not stable, The closest stable data that saved is: [-1.  1. -1.  1. -1.  1.  1. -1.]
4 times this while worked to inps_stablizer the input
[[ 0.  1.  3. -1.  5. -1. -1.  3.]
 [ 1.  0.  1.  1. -1.  5.  1.  1.]
 [ 3.  1.  0. -5.  5. -1. -5.  7.]
 [-1.  1. -5.  0. -3. -1.  7. -5.]
 [ 5. -1.  5. -3.  0. -3. -3.  5.]
 [-1.  5. -1. -1. -3.  0. -1. -1.]
 [-1.  1. -5.  7. -3. -1.  0. -5.]
 [ 3.  1.  7. -5.  5. -1. -5.  0.]]
[ 1  1  1 -1  1  1 -1  1] stable

```

همانطور که مشنص شده تعداد دفعات فراخوانی و پایدار شدن نهایی نشان داده شده است.

اخرین وزن ذخیره شده هم میشود همانی که در عکس بالا گذاشتم و دوباره عکس آن را در پایین نیز میگذارم.

```

[[ 0.  1.  3. -1.  5. -1. -1.  3.]
 [ 1.  0.  1.  1. -1.  5.  1.  1.]
 [ 3.  1.  0. -5.  5. -1. -5.  7.]
 [-1.  1. -5.  0. -3. -1.  7. -5.]
 [ 5. -1.  5. -3.  0. -3. -3.  5.]
 [-1.  5. -1. -1. -3.  0. -1. -1.]
 [-1.  1. -5.  7. -3. -1.  0. -5.]
 [ 3.  1.  7. -5.  5. -1. -5.  0.]]

```

(ت)

برای حل آن می توان حلقه **for** پیاده سازی نمود بدین گونه که تعداد دور های آن 2 به توان 8 هست که تعدادی با معکوس و تعدادی با خود توابع **train** شده برابر می شوند.

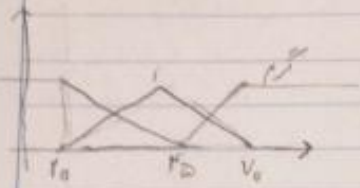
ث)

حال اگر نصف یا بیشتر بیت ها نویزی باشند، ممکن است به معکوس بیت تمرین داده شده یا ورودی دیگر نزدیک شده باشد زیرا ما قبلا آن را به شبکه مان آموزش داده باشیم. پس همیشه همگراست.

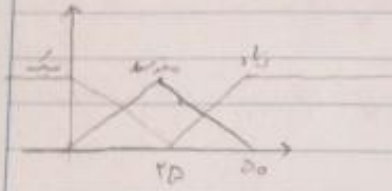
## سوال 2

سؤال ۱۲

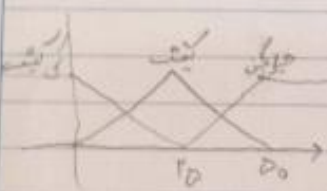
دمای آب:



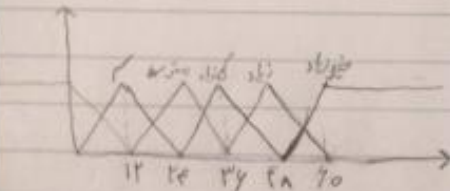
درین نمودار:



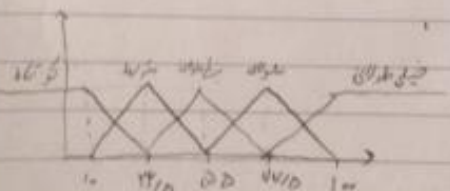
درجه چربی:

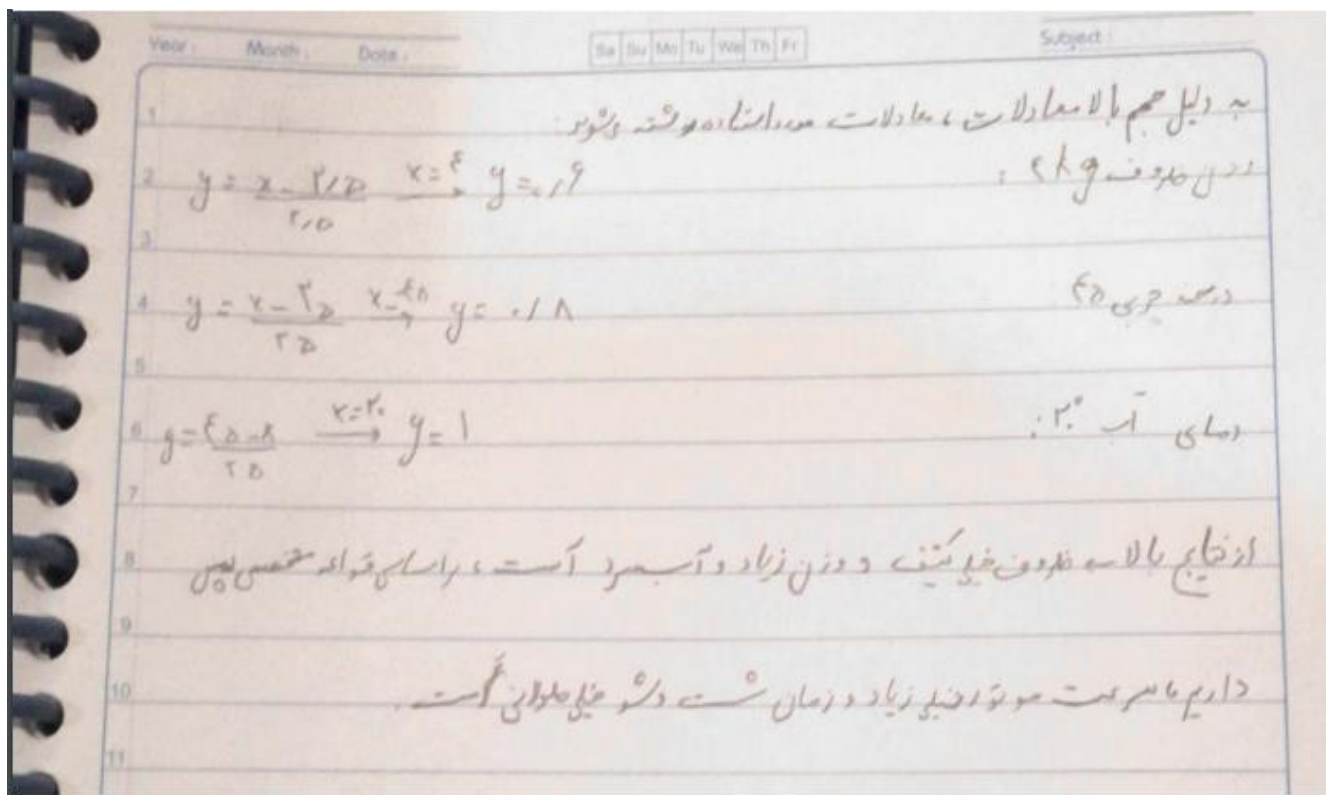


سرعت چرخش موتور:



زمان تست موتور:





## سوال 3

```
tmprte = AutoTriangle(5, terms=['very_cold', 'cold', 'moderate', 'hot', 'very_hot'], universe_of_discourse=[-20,100])
hmdt = AutoTriangle(3, terms=['dry', 'normal', 'wet'], universe_of_discourse=[0, 100])
Rainfall_amount = AutoTriangle(5, terms=['very_low', 'low', 'moderate', 'high', 'very_high'], universe_of_discourse=[0,90])
height = AutoTriangle(5, terms=['very_low', 'low', 'moderate', 'high', 'very_high'], universe_of_discourse=[0,5500])
```

```
fuzzsys.add_linguistic_variable("temperature_3", tmprte)
fuzzsys.add_linguistic_variable("temperature_2", tmprte)
fuzzsys.add_linguistic_variable("temperature_1", tmprte)
fuzzsys.add_linguistic_variable("humidity", hmdt)
fuzzsys.add_linguistic_variable("rain_amount", Rainfall_amount)
fuzzsys.add_linguistic_variable("altitude", height)
fuzzsys.add_linguistic_variable("today_temperature", tmprte)
```

```
rules = [
```

```
"IF (temperature_3 IS very_cold) AND (temperature_2 IS cold) AND (temperature_1 IS cold) THEN (today_temperature IS cold)",
"IF (altitude IS very_high) OR (humidity IS dry) THEN (today_temperature IS hot)",
"IF (humidity IS dry) AND (rain_amount IS high) THEN (today_temperature IS moderate)",
"IF (altitude IS low) AND (temperature_1 IS hot) THEN (today_temperature IS hot)",
"IF (rain_amount IS low) AND (humidity IS wet) THEN (today_temperature IS very_hot)",
"IF (temperature_3 IS hot) AND (temperature_2 IS moderate) AND (temperature_1 IS hot) THEN (today_temperature IS very_hot)"
]
```

برای حل سوال داده شده ، کنترلر فازی که دارای کنترلی با 4 پارامتر ورودی هست . به جز پارامتر دمای سه روز گذشته که 5 ترم دارد که "خیلی سرد، سرد، متوسط، گرم و خیلی سرد"

گرم" هست، بقیه متغیر ها یعنی **رطوبت و میزان بارش و ارتفاع** شهر همگی **3** ترم دارند که شامل کم، متوسط، زیاد می باشد. همچنین این کنترلر یک پارامتر فروبی هم دارد که دما می باشد و **5** ترم دارد که شامل "خیلی سرد، سرد، متوسط، گرم و خیلی گرم" می باشد. در تعریف این متغیر، بازه هر کدام نیز مشخص شده و سپس هر متغیر تولید شده است که عکس کدهای آن در بالا موجود هست. حال **8** قانون برای این کنترلر تعریف میکنیم در نهایت به عنوان مثال برای بررسی آن، **4** ورودی مقدار دهی شده و با استفاده از روش ممدانی فروبی دما برای ما توسط کنترلر محاسبه شده است. فروبی دمای بدست آمده به ازای متغیر های داده شده است.