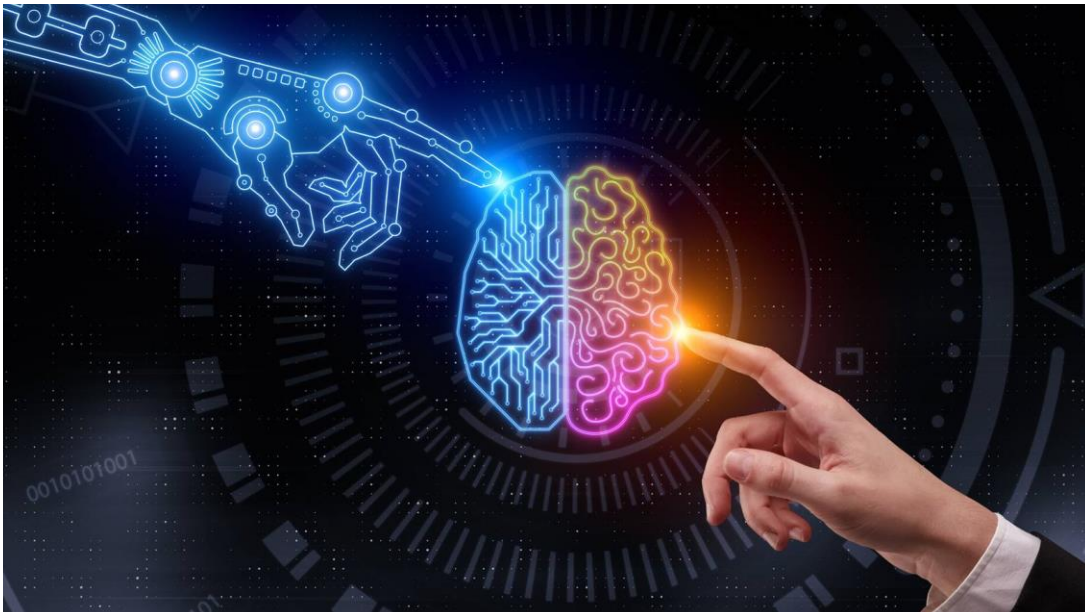
به نام خدا



هوش مصنوعی

سوال عملی 1 تمرین 2

استاد :

دکتر رهبان

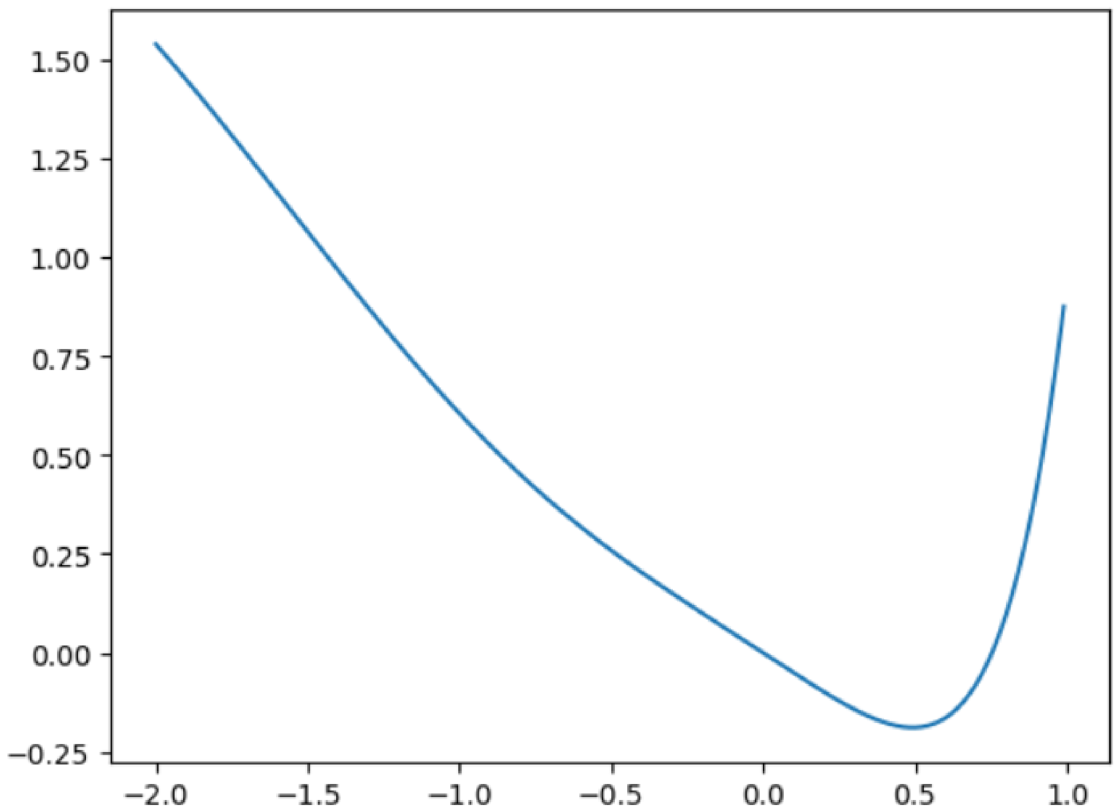
نویسنده :

امیرحسین عابدی

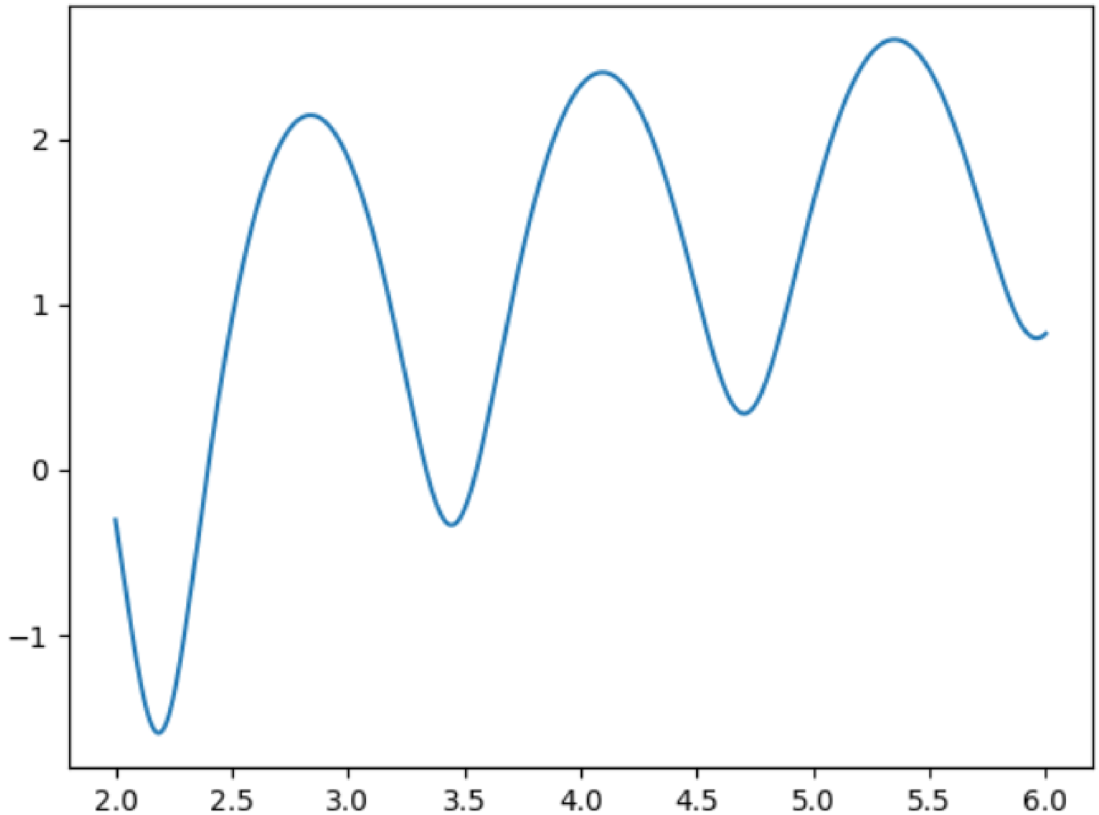
شماره دانشجویی : 99105594

۱ یافتن توابع محدب و پیدا کردن مینیمم توابع غیرمحدب

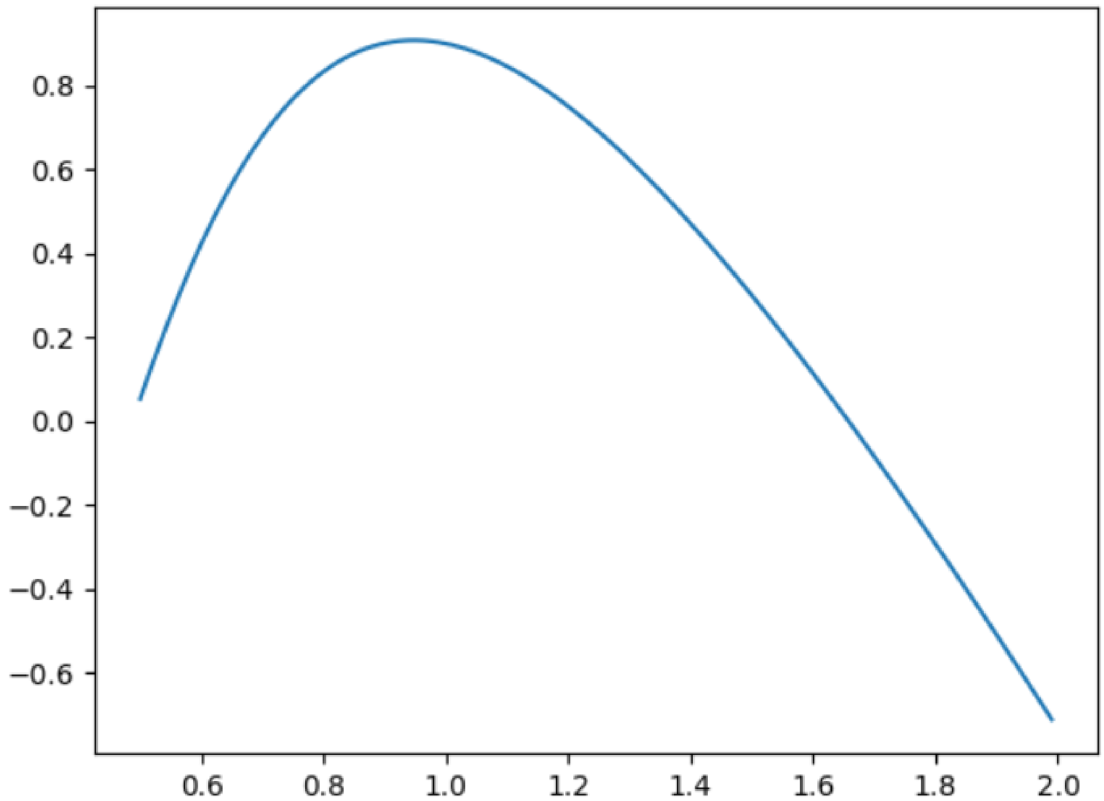
در ابتدا تابع draw_function را مینویسیم که در ورودی های خود یک تابع دریافت میکند و به ازای اول و آخر بازه آنرا رسم میکند. حال از این تابع استفاده میکنیم تا محدب بودن توابع خواسته شده را بررسی کنیم. هر 3 تابع را رسم میکنیم :



این نمودار تابع اول است که برای بازه داده شده محدب میباشد.



این نمودار مربوط به تابع دوم میباشد که محدب نیست.



این نمودار مربوط به تابع سوم است که محدب نیست.

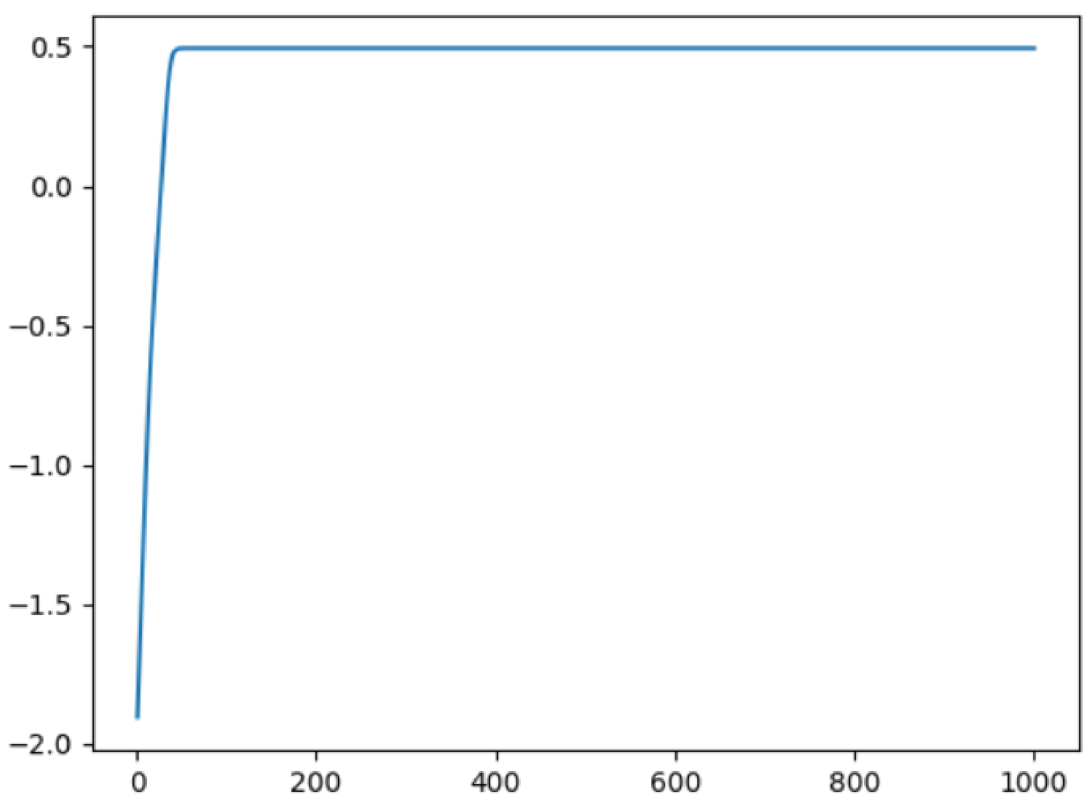
حال برای تابع دوم و سوم که محدب نیستند از روش بازه بندی روی بازه با انتها و ابتدای متناهی استفاده میکنیم. و همینطور میتوان دید که این توابع Lipschitz continuous هستند پس این روش برای آنها جواب میدهد. به طور مثال بازه بندی را با بازه های به طول 0.001 میتوانیم انجام دهیم که به وضوح در این بازه ها توابع تغییر زیادی ندارند.

2 الگوریتم GC برای پیدا کردن مینیمم در تابع اول (محدب)

در این قسمت میخواهیم از الگوریتم GC استفاده کنیم تا مینیمم گلوبال را برای تابع داده شده در بازه خواسته شده محاسبه کنیم. حال به ازای هرکدام از آلفا های خواسته شده روش GC و همگرا شدن آنرا مقایسه میکنیم.

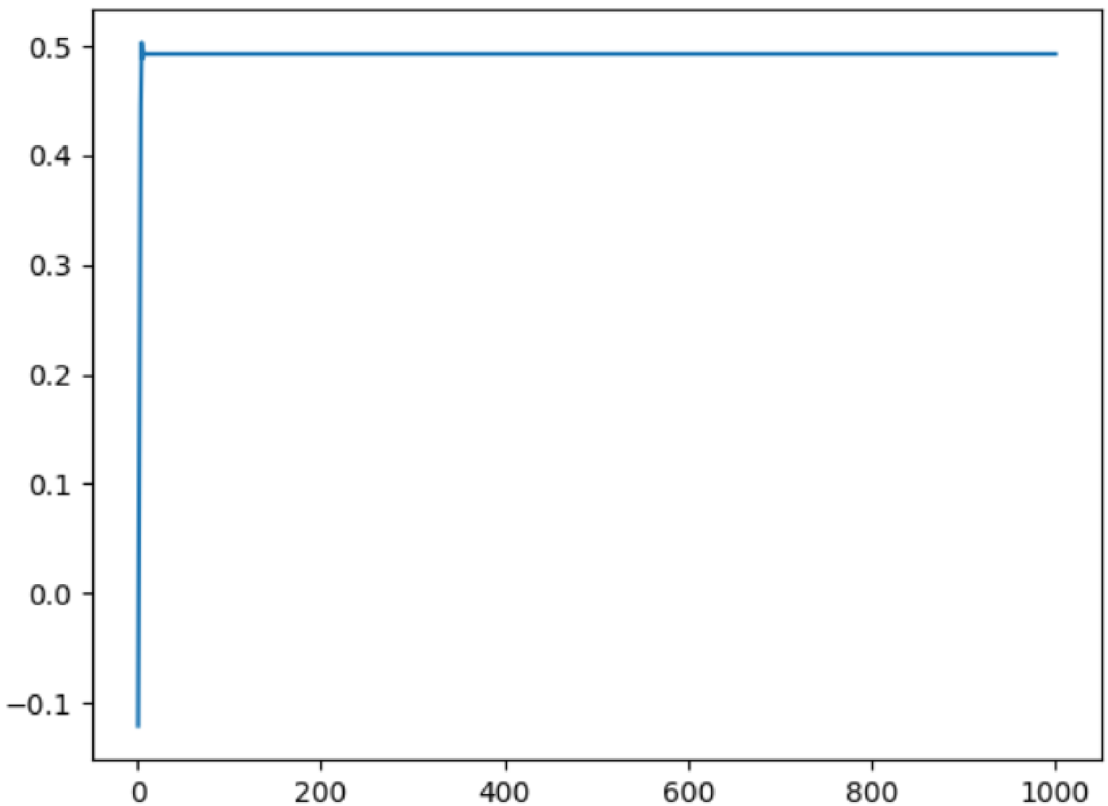
1.2 Alpha = 0.1

این نمودار x پیدا شده برحسب دفعه تلاش برای این مقدار از آلفا میباشد که معلوم است که همگرایی رخ داده است و مقدار 0.5 یافت شده است.

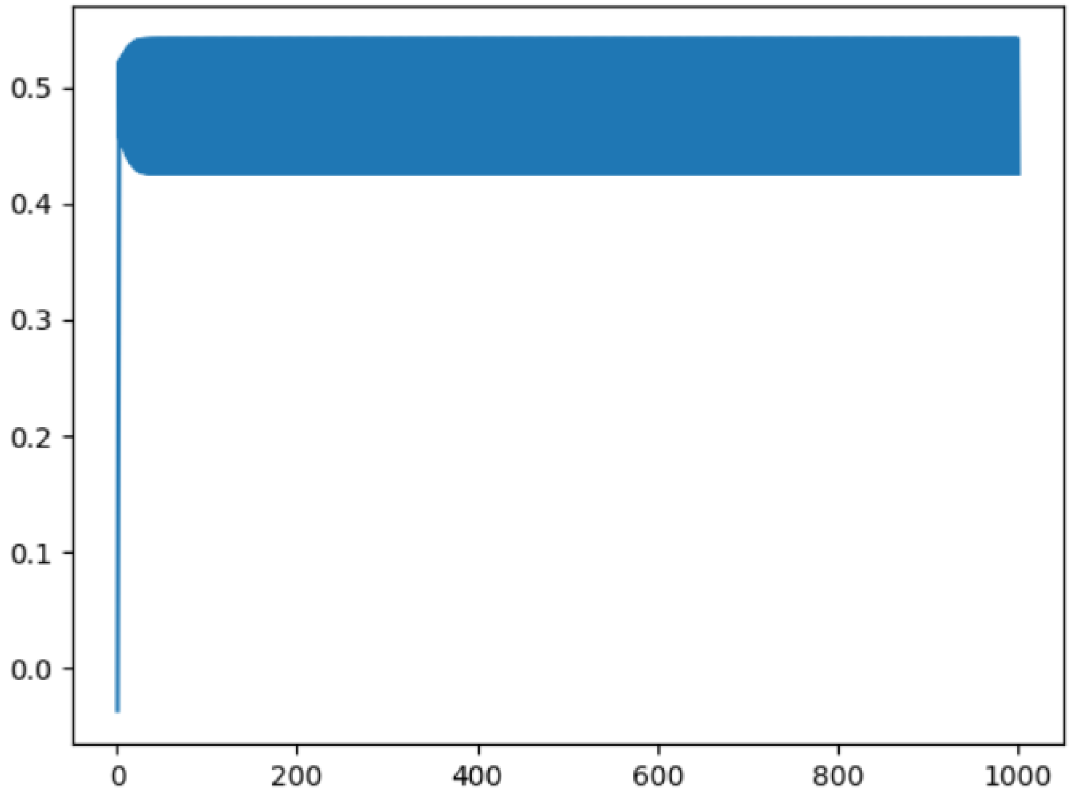


2.2 Alpha = 0.4

نمودار زیر مانند نمودار قبل اما برای alpha = 0.4 میباشد. همگرایی رخ داده است اما این همگرایی سریعتر از حالت قبل رخ داده است. مقدار یافت شده برای نقطه مینیمم تابع همان مقدار قبلی است.

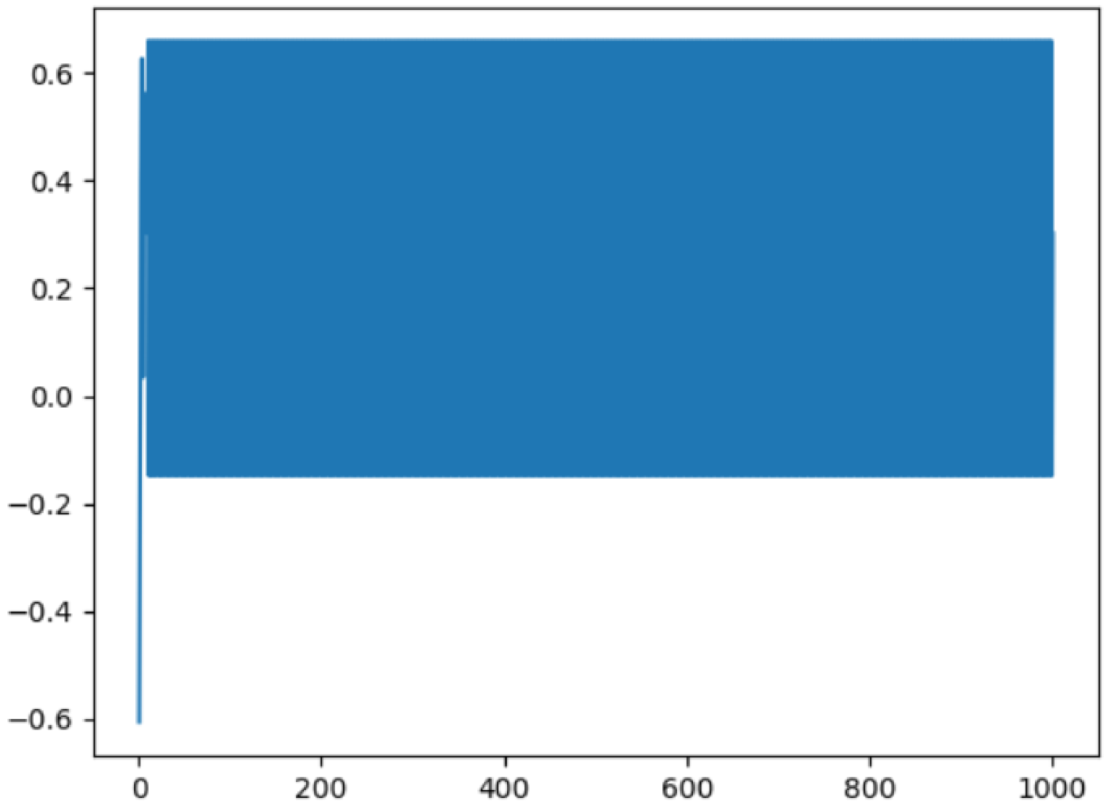


Alpha = 0.6   3.2



به ازای این مقدار از آلفا همگرایی رخ نداده است و نوسان های پی در پی در تابع مشاهده میشود.

4.2 Alpha = 0.9



به ازای این مقدار از آلفا هم همگرایی رخ نداده است و نوسان های آن از حالت قبلی هم بیشتر است.

3 استفاده از GC برای توابع غیرمحدب

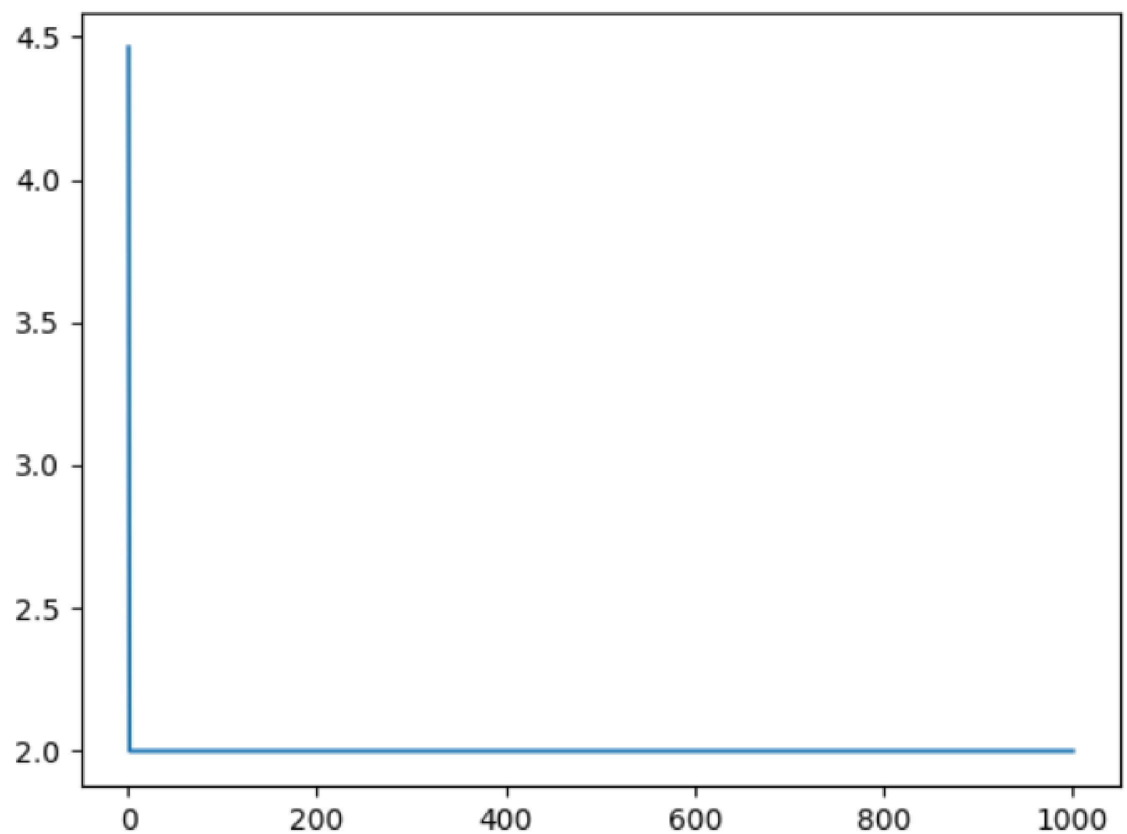
در ابتدا لازم است که در تابع دوم و سوم ، مینیمم گلوبال را بدانیم تا بتوانیم در ادامه با استفاده از الگوریتم ، ببینیم که در چند درصد از مواقع به این نقطه میرسیم.

در تابع دوم مینیمم گلوبال در نقطه (2.186, -1.591) بدست می آید و همینطور در تابع سوم این نقطه (2, -0.734) بدست می آید.

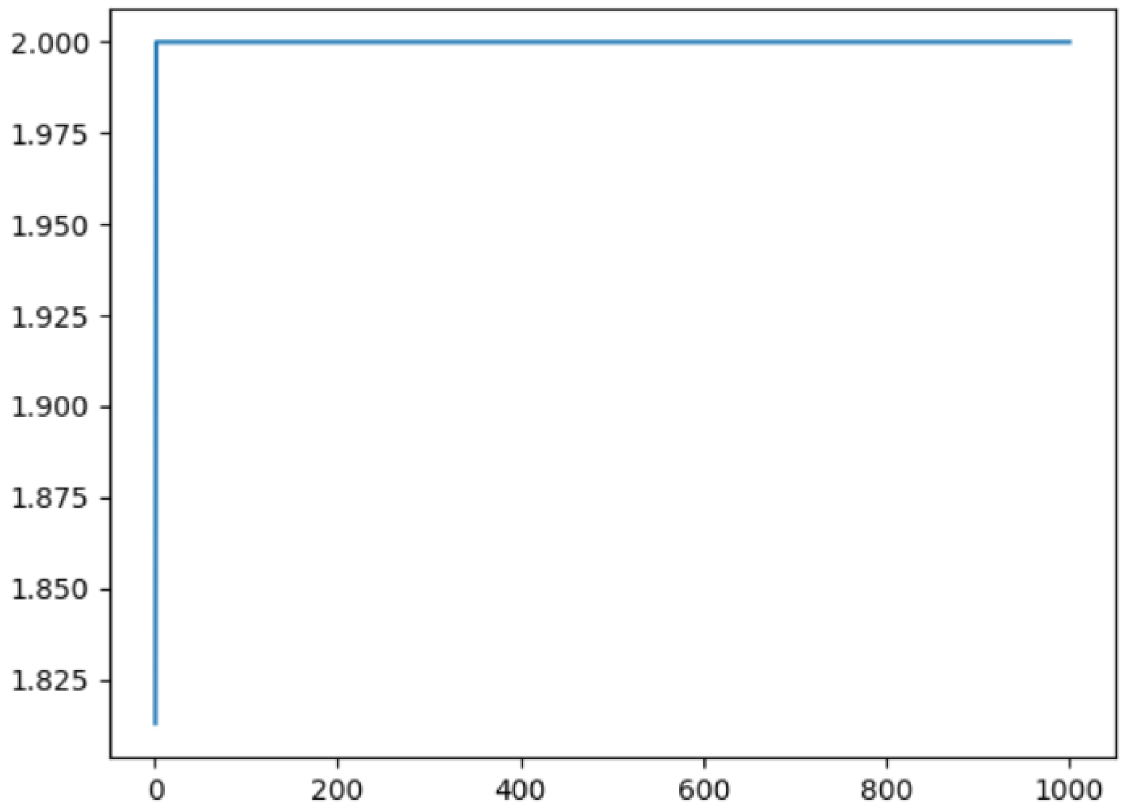
حال در ابتدا برای هرکدام از این توابع GC را اجرا میکنیم (دقت کنید که چون توابع محدب نیستند لزومی برای پیدا کردن جواب درست برای الگوریتم وجود ندارد.)

در ابتدا این الگوریتم را برای تابع دوم اجرا میکنیم. (فرض کنید در این قسمت آلفا را برابر با 0.1 در نظر بگیریم.)

در تابع دوم نمودار x برحسب تعداد بار تکرار به این صورت خواهد بود :



حال این تابع را با همین learning rate و به همین تعداد تکرار برای تابع سوم اجرا میکنیم.



همانطور که مشاهده میشود در اینجا هم همگرایی به سمت x=2 را داریم.

حال GC را برای این توابع اجرا میکنیم و میبینیم که در چند درصد از این مواقع به این نقطه میرسیم. (فرض کنید که اگر 0.001 نزدیک به مینیمم شدیم ، آنرا قبول کنیم.)

```
f2 with 0.1 learning rate : 0.0%
f2 with 0.4 learning rate : 0.0%
f2 with 0.6 learning rate : 0.0%
f2 with 0.9 learning rate : 0.0%
f3 with 0.1 learning rate : 70.0%
f3 with 0.4 learning rate : 71.8%
f3 with 0.6 learning rate : 72.1%
f3 with 0.9 learning rate : 70.2%
```

هنگام اجرای این الگوریتم برای این دو تابع ، در تابع اول در هیچ موقع به همگرایی و جواب درست نمیرسیم. و در تابع سوم به طور میانگین در 70٪ مواقع به جواب درست دست میابیم. (علت دقت صفر رو هم میتوان به این صورت بیان کرد که ، تابع دوم مشتق زیادی دارد ، حتی در بعضی نقاط این مقدار از اردر 10 هست ، حال فرض کنید با لرنینگ ریت 0.1 داریم ادامه میدهیم ، در اینجا ممکن است پرش هایی به طول 1 داشته باشیم به این معنی که ممکن است در تلاش های خود به گلوبال مینیمم نرسیم.)

4 پیدا کردن مینیمم با روش نیوتون-رافسون

در این قسمت میخواهیم با روش نیوتون-رافسون نقطه صفر مشتق تابع را بدست آوریم. در ابتدا مانند قسمت قبل یکبار این تابع را اجرا میکنیم تا ببینیم به چه جوابی میرسیم. دقت کنید که این روش نقطه ای را پیدا میکند که مشتق در آن صفر است. این نقطه میتواند مینیمم/گلوبال/یا هیچ کدام نباشد.

```
f2 with newton-raphson , x found : 4.093958340533443
f3 with newton-raphson , x found : 0.9472406902882855
```

این شکل نشان میدهد که این روش برای توابع داده شده برای یک بار اجرا چه نقاطی را پیدا کرده است. که این نقاط درمورد هر دو تابع نقاط ماکزیمم میباشد.

حال میخواهیم ببینیم که با اجرای 1000 بار این الگوریتم در چه درصدی از دفعات ، این روش قادر خواهد بود که نقطه درست را پیدا کند. نقاط جواب را در قسمت قبل مشخص کردیم. (فرض کنید اگر جواب به جواب اصلی به اندازه 0.001 اختلاف داشت ، آنرا بپذیریم.)

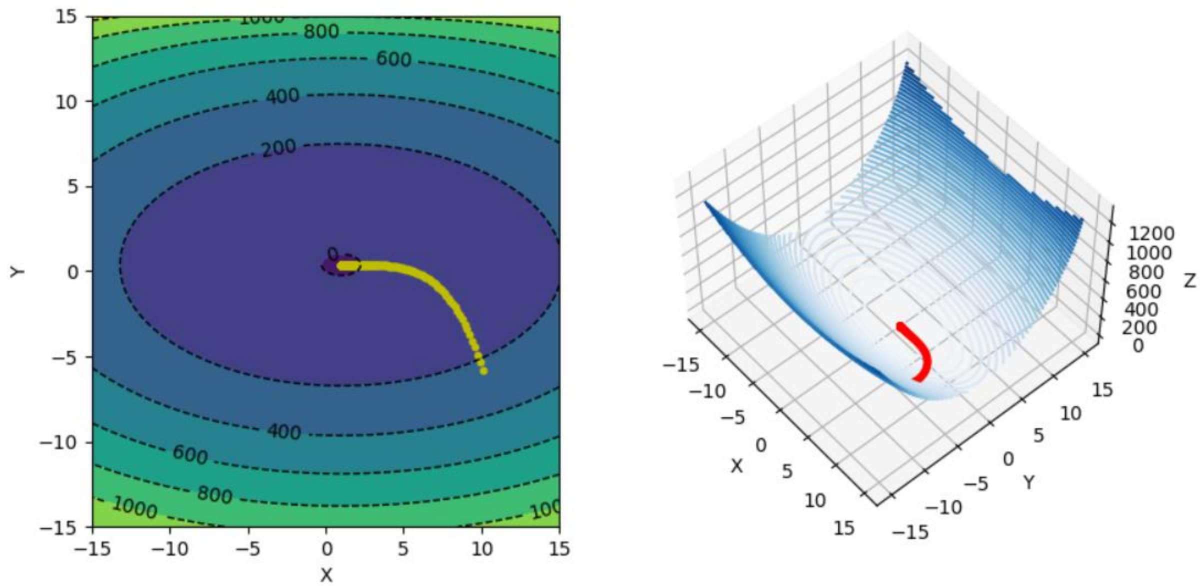
```
f2 newton-raphson percentage : 5.2
f3 newton-raphson percentage : 0.0
```

همانطور که مشاهده میشود در 5.2 درصد مواقع این روش تابع دوم را دست پیدا میکند و برای تابع سوم در هیچ موقعی نمیتواند مینیمم را به درستی پیدا کند. به نظر می آید که این روش به خوبی روش قبل عمل نمیکند. و روش کاهش گرادیان به طور میانگین بسیار بهتر عمل میکرد. (چرا که این تابع ماکزیمم هارا هم میتواند پیدا کند و در بعضی جاها پرش های زیادی میکند.)

5 پیدا کردن مینیمم تابع دو متغیره به کمک GC

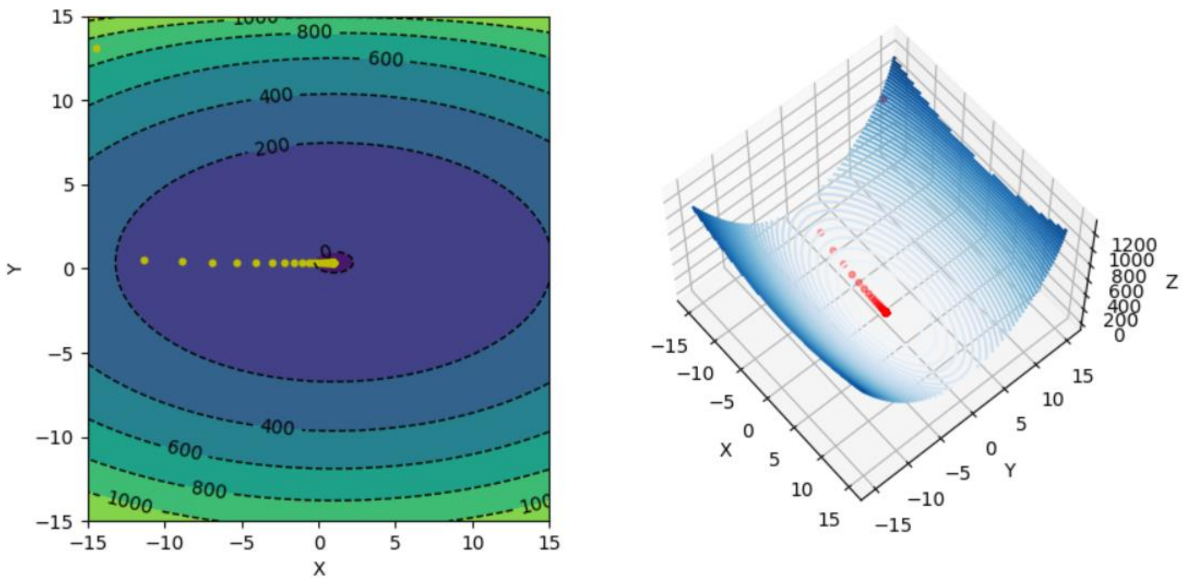
در اینجا میخواهیم با استفاده از GC مینیمم یک تابع دو متغیره را پیدا کنیم. در ابتدا از learning rate برابر با 0.01 استفاده میکنیم و روند اجرای الگوریتم را بررسی میکنیم.

روند اجرا به این صورت خواهد بود :

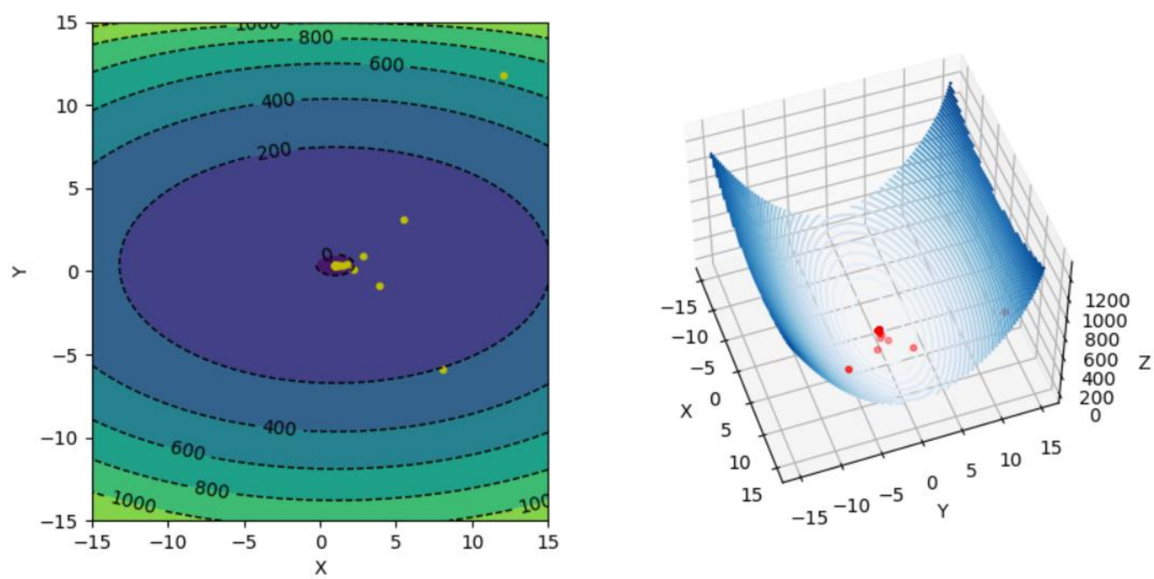


در اینجا الگوریتم خیلی نرم به سوی مینیمم حرکت کرده است و حرکت پیوسته ای دارد.

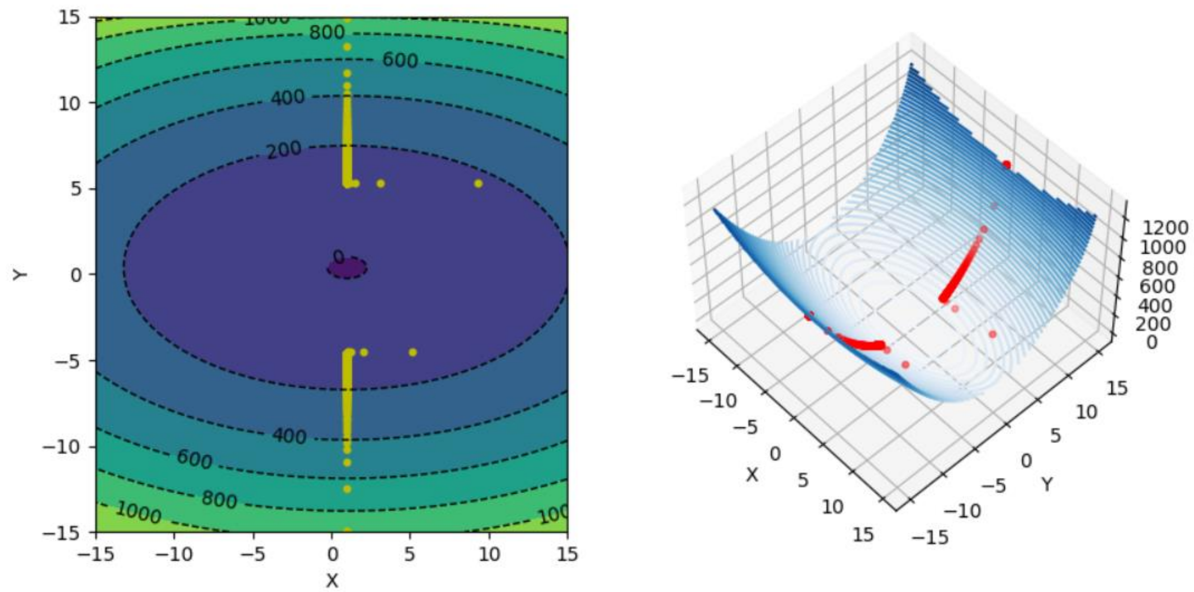
حال آلفا را کمی افزایش میدهیم تا سریعتر به سوی مینیمم محلی حرکت کنیم. به ازای alpha = 0.1 داریم :



که همانطور که مشاهده میشود ، باز همان نقطه یافت شده است اما حرکت به سوی آن بسیار سریعتر رخ داده است. حال باری دیگر آلفا را افزایش میدهیم. به ازای alpha = 0.18 داریم :



در اینجا پرش رخ داده است اما پرش ها به آن بزرگی نبوده اند که مانع یافت نشدن گلوبال مینیمم شوند. بنابراین به ازای این آلفا هم گلوبال مینیمم یافت شده است. به ازای alpha = 0.25 داریم :



همانطور که مشاهده میشود به ازای این آلفا ، پرش های بسیار بزرگی رخ داده است و الگوریتم نتوانسته است گلوبال مینیمم را پیدا کند.

6 Simulated Annealing

در این قسمت میخواهیم تابع ۲ را به کمک simulated annealing مینیمم کنیم. (تابع 1 محدب و تابع 3 مقعر است.) این تابع را به ازای مقدار زیر اجرا میکنیم.

Stopping temp = 0.00001

Stopping iter = 100000

Initial T = 100

Gamma = 0.99

با استفاده از پارامتر های داده شده میتوانیم به این دقت ها برسیم.

```
SA percentage using alpha = 0.1 : 18.8
SA percentage using alpha = 0.2 : 24.7
SA percentage using alpha = 0.3 : 20.8
SA percentage using alpha = 0.4 : 23.9
SA percentage using alpha = 0.5 : 21.8
SA percentage using alpha = 0.6 : 24.3
SA percentage using alpha = 0.7 : 26.7
SA percentage using alpha = 0.8 : 27.2
SA percentage using alpha = 0.9 : 29.2
SA percentage using alpha = 1.0 : 32.2
```

که با توجه به اینکه تابع نه مقعر میباشد و نه محدب ، مقادیر قابل قبولی هستند.

۷ مقایسه

در کل برای مقایسه میتوان گفت که بین تمام روش ها روش GC از خود عملکرد بسیار بهتری نشان داده است. مخصوصا برای توابعی که مشتق تقریبا کمی دارند و یا بازه قابل قبولی از آنها در مجاورت مینیمم گلوبال هستند. در اینجور توابع میتوانیم به راحتی از GC استفاده کنیم و یا چند بار آنرا اجرا کنیم و بین آنها مینیمم بگیریم.

بعد از آن الگوریتم SA خودش را بسیار نشان داد ، مخصوصا درمورد توابعی که لوکال مینیمم های زیادی دارند و نیاز داریم که از آنها به بیرون بپریم. با انتخاب پارامتر های درست

میتوانیم از این الگوریتم بهره ببریم و بعد از اجرای چندباره الگوریتم این اطمینان را داشته باشیم که بین این نقاط بدست آمده ، حداقل یکبار به گلوبال مینیمم رسیده ایم.

در آخر الگوریتم نیوتون-رافسون قرار دارد که عملکرد خیلی خوبی از خودش نشان نداد. این الگوریتم توانایی پیدا کردن نقاط مینیمم بر روی انتهای بازه را ندارد و همینطور نقاط ماکزیمم را هم پیدا میکند. همینطور ممکن است در نقاطی که مشتق دوم به صفر میل میکند ، پرش های بسیار بلندی بکند و به کل سرچ را به چالش بکشد.