

**درس رایانش عصبی و یادگیری عمیق**

**استاد صفابخش**

**نیما پری فرد**

**402131017**

فهرست تمرین شماره 1

[تمرین 1 3](#_Toc162211479)

[1-a: 3](#_Toc162211480)

[1-b: 4](#_Toc162211481)

[1-c: 7](#_Toc162211482)

[تمرین 2 9](#_Toc162211483)

[2-a: 10](#_Toc162211484)

[2-b: 15](#_Toc162211485)

[2-c: 19](#_Toc162211486)

[تمرین 3 24](#_Toc162211487)

[3-a: 24](#_Toc162211488)

[3-b: 24](#_Toc162211489)

[3-c: 26](#_Toc162211490)

# تمرین 1

## 1-a:

A piece of paper with writing on it

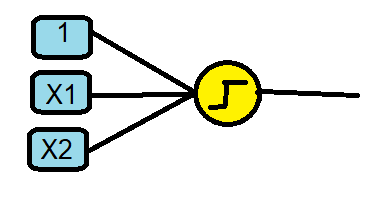
Description automatically generated

## 1-b:

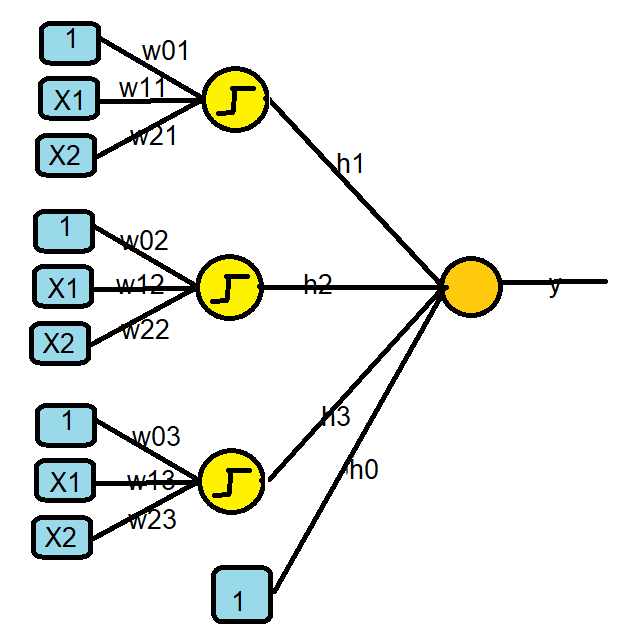
چون با یک خط نمی توان دو کلاس را از هم جدا کرد برای همین به صورت خطی جدایی پذیر نیست

برای حل این سوال نیاز به یک لایه مخفی داریم و در نهایت یک نورون در خروجی که خروجی این سه نوورن را ادغام کند.

برای حل این سوال به عنوان نورون پایه از نورون 1957 که دارای یک تابع خطی و تابع فعالیت پله بود استفاده می کنیم.



در نهایت معماری پیشنهاد شده به شکل زیر است.

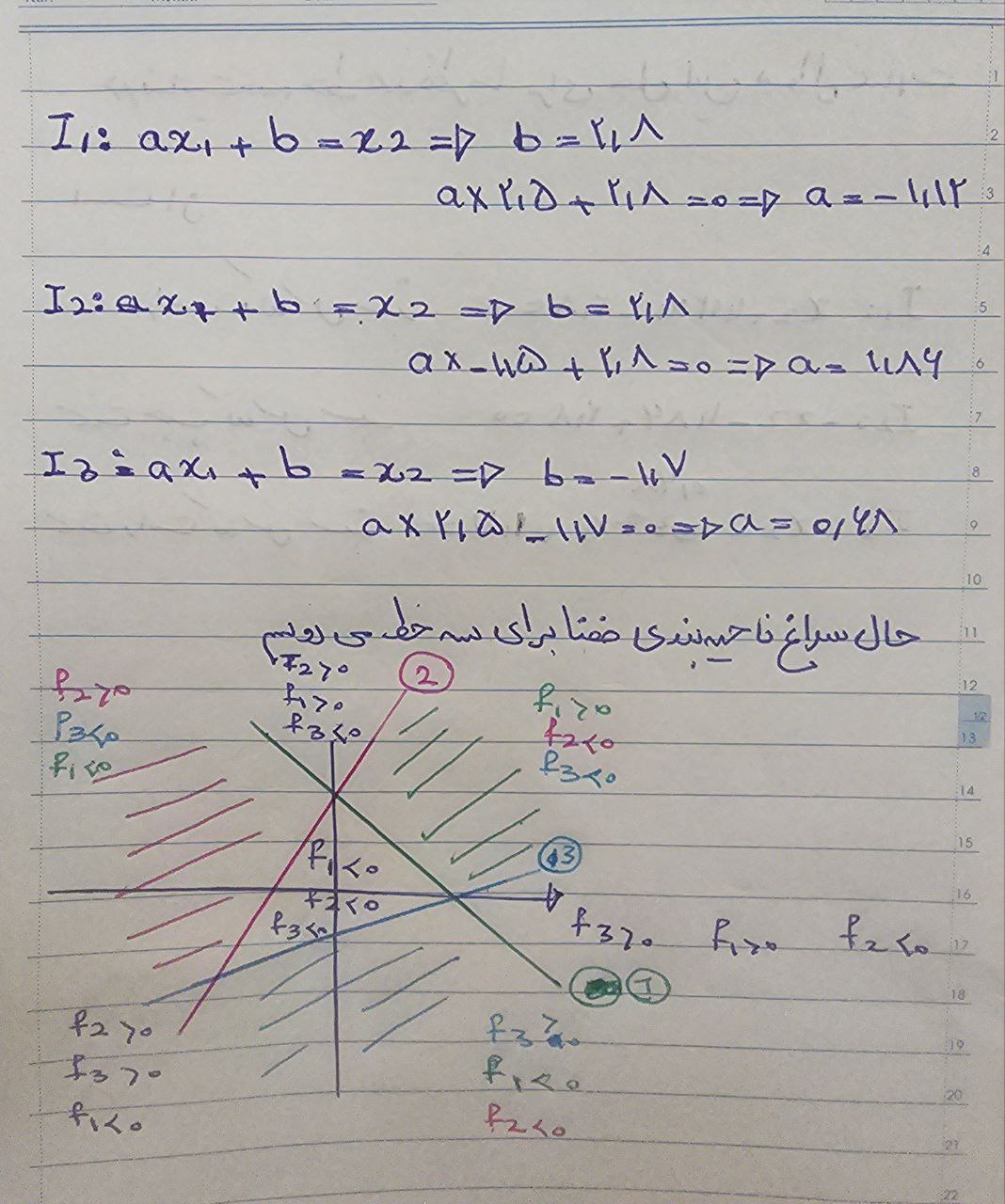


حال به دنبال پیدا کردن وزن های مناسب برای حل این سوال هستیم.

برای حل این سوال ایده بنده این است که اگر در محدوده کلاس آبی بیافتد هر سه خط کلاس مقدار -1 خروجی بده.

در نهایت وزن های h1 و h2 و h3 جوری مقدار دهی کنیم که در نهایت خروجی y اگر کلاس سبز بود +1 و اگر خروجی کلاس آبی -1 خروجی بدهد.

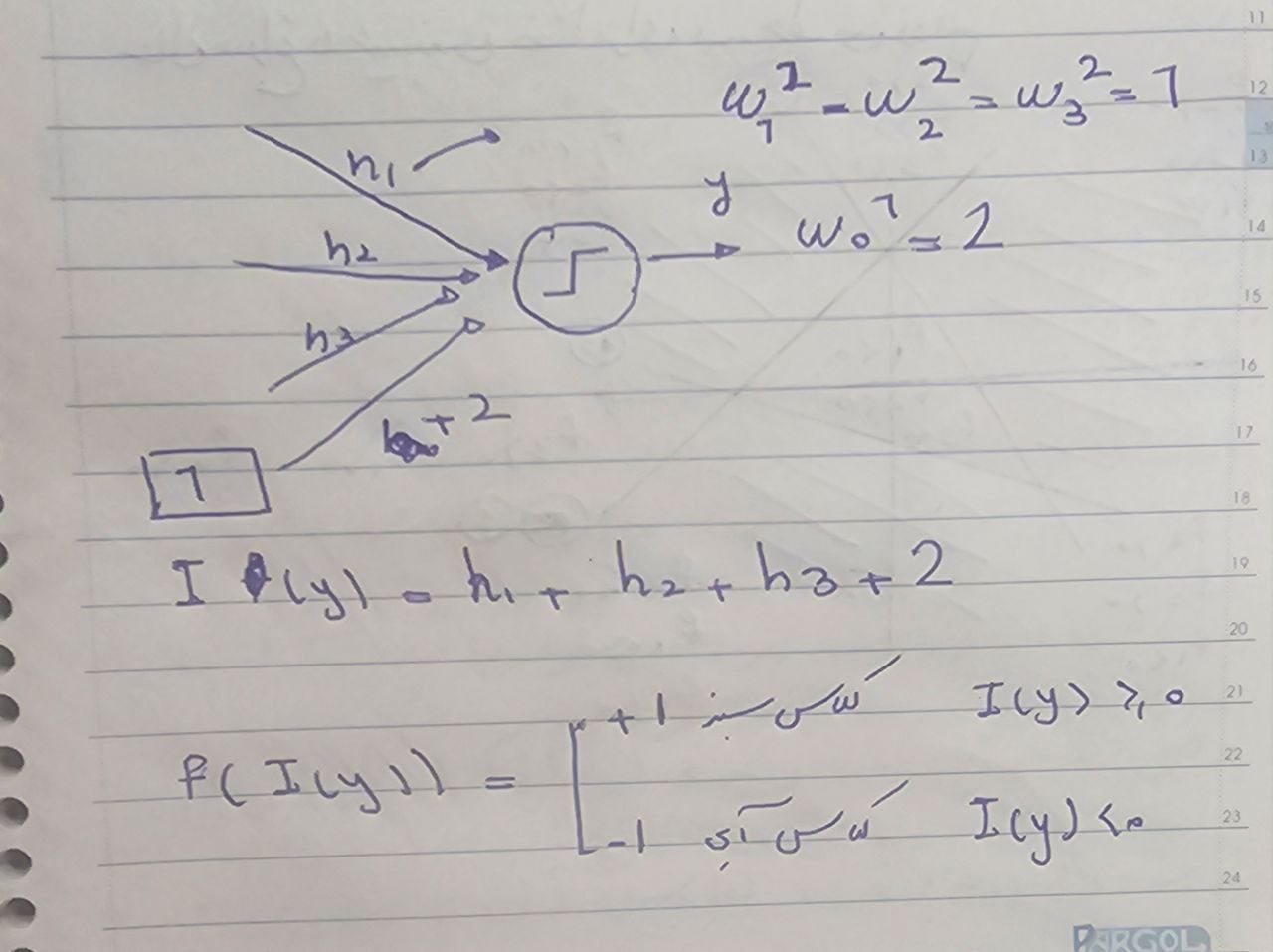
حال در ابتدا سه خط جدا کننده در نورون های لایه مخفی را به دست می آوریم.



A piece of paper with writing on it

Description automatically generated

در این حالت خروجی برای فضایی که داده آبی در ان وجود دارد برابر با -1 می شود و برای بقیه ناحیه بزرگتر از یک می شود.



## 1-c:

Kernel Trick مفهومی در یادگیری ماشین است، به‌ویژه در زمینه‌ی ماشین‌های بردار پشتیبان (SVM). این امکان را فراهم می‌کند که داده‌های ورودی به صورت غیرخطی به یک فضای بعد بالاتر تبدیل شوند بدون اینکه مانند روش های دیگر افزایش بعد هزینه محاسباتی افزایش پیدا کند و ما دچار نفرین بعد شویم.

Kernel trick یک مرز تصمیم غیرخطی را با تبدیل ضمنی داده‌های ورودی به یک فضای بعد بالاتر پیدا می کند که در آن یک مرز تصمیم خطی می‌تواند ایجاد شود. این کار با استفاده از یک تابع هسته (Kernel Function) انجام می‌شود که ضرب داخلی بین جفت داده‌ها در فضای بعد بالاتر را بدون محاسبه تبدیل به صورت صریح محاسبه می‌کند. توابع هسته محبوب شامل تابع هسته خطی، تابع هسته چندجمله‌ای، تابع هسته گاوسی (RBF) و تابع هسته سیگموئیدی هستند.

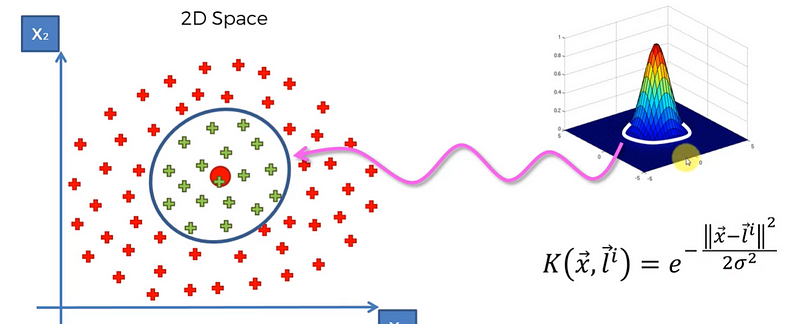
با استفاده از kernel trick، دسته بند می‌توانند به‌طور مؤثری داده‌ها را دسته‌بندی کنند که ممکن است در فضای ویژگی اصلی به صورت خطی جدایی‌پذیر نباشند و این باعث افزایش انعطاف‌پذیری و عملکرد آن‌ها در وظایف مختلف دسته‌بندی می‌شود.

شگرد کرنل روشی است که داده های ها را به بعد های بالاتر می برد بدون این که محاسبات ما بیش تر شود.

در این جا به نظر بنده بهترین کرنل rbf است چون می تواند به خوبی این شکل با استفاده از یک نورون و این کرنل از هم جدا کنیم.

برای مثال شکل زیر شبیه شکل 1.b زیر است.

که با استفاده کرنل rbf همان طور که در شکل زیر مشاهده می کند شبیه شکل ماست و با افزایش به راحتی با استفاده از یک صفحه جدا شده است.



منبع این شکل و متن: <https://medium.com/@suvigya2001/the-gaussian-rbf-kernel-in-non-linear-svm-2fb1c822aae0>



A mathematical equation with a number of symbols

Description automatically generated with medium confidence

# تمرین 2

برای پیاده سازی این تمرین در فایل perveptron.py یک نورون پرسپترونی را از صفر تا صد طبق نورون معرفی شده در اسلاید ها و سال 1957 پیاده سازی کردم.

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

## 2-a:

مشخصات پرسپترون آموزش داده شده:

1. تابع خطا: پله
2. بهینه ساز: گرادیان کاهشی
3. معماری شبکه: از یک پرسپترون استفاده کردیم.
4. تعداد گام آموزش: 40
5. اندازه دسته: برای آموزش کل داده به اموزش می دادم در واقعا اندازه برابر است با کل داده بنده.
6. آمارگان تفکیک داده: طبق متن گفته شده به نسبت 80 10 10 داده را تقسیم کردم.

A screenshot of numbers

Description automatically generated

1. پیش پردازش اعمال شده: در مورد اول پیش پردازشی اعمال نشده.
2. نرخ یادگیری: 0.1

منحنی یادگیری و منحنی دقت:

A graph of a graph

Description automatically generated

A graph of a graph showing a line graph

Description automatically generated with medium confidence

اگر بخواهم تحلیلی در مورد این نمودار ها بدهم علت زیگزاگ بودن این مدل می تواند این باشد توانایی این مدل این قدر است و داده جدایی پذیر خطی نیست و نمی تواند صفحه ای مناسب پیدا کند که داده ها از هم جدا کند

دقت و معیار f1 به سه دسته داده مد نظر:

A screenshot of a computer

Description automatically generated

ماتریس confusion برای سه دسته داده:

داده آموزش:

A screenshot of a graph

Description automatically generated

داده validation:

A screenshot of a graph

Description automatically generated

داده تست:

A screenshot of a graph

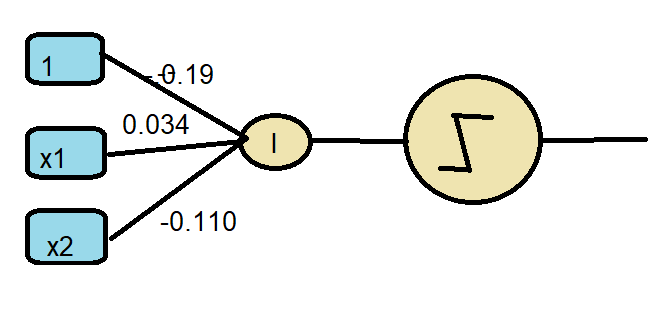
Description automatically generated

وزن های مدل آموزش داده شده:

A screenshot of a computer

Description automatically generated

طرحواره نورون پرسپترونی:



## 2-b:

1. تابع خطا: پله
2. بهینه ساز: گرادیان کاهشی
3. معماری شبکه: از یک پرسپترون استفاده کردیم.
4. تعداد گام آموزش: 130
5. اندازه دسته: برای آموزش کل داده به اموزش می دادم در واقعا اندازه برابر است با کل داده بنده.
6. آمارگان تفکیک داده: طبق متن گفته شده به نسبت 80 10 10 داده را تقسیم کردم.

A screenshot of a computer

Description automatically generated

1. پیش پردازش اعمال شده: تا توان سوم ویژگی ها اضافه شد.

A screenshot of a computer

Description automatically generated

1. نرخ یادگیری: 0.1

منحنی یادگیری دقت و صحت:

A graph of a graph showing the results of training error

Description automatically generated with medium confidence

A graph showing the performance of a performance

Description automatically generated

توان های ویژگی ها به مسئله اضافه شده و خب دقت بهتر شده است اما همچنان مسیر آموزش دارای حرکات زیگزاگی است که شاید به این خاطر است که ارتباط ویژگی ها را در نظر گرفته ایم.

بعد تا گام 130 پیش رفتم چن بعد آن اورفیت می شد و آموزش را همین جا نگه داشتم.

دقت و معیار f1 به سه دسته داده مد نظر:

A screenshot of a computer

Description automatically generated

ماتریس confusion برای سه دسته داده:

داده آموزش:

A screenshot of a graph

Description automatically generated

داده validation:

A screenshot of a graph

Description automatically generated

داده تست:

A screenshot of a graph

Description automatically generated

وزن های مدل آموزش داده شده:

A screenshot of a computer program

Description automatically generated

## 2-c:

1. تابع خطا: پله
2. بهینه ساز: گرادیان کاهشی
3. معماری شبکه: از یک پرسپترون استفاده کردیم.
4. تعداد گام آموزش: 170
5. اندازه دسته: برای آموزش کل داده به اموزش می دادم در واقعا اندازه برابر است با کل داده بنده.
6. آمارگان تفکیک داده: طبق متن گفته شده به نسبت 80 10 10 داده را تقسیم کردم.

A screenshot of numbers

Description automatically generated

1. پیش پردازش اعمال شده: تا توان سوم ویژگی ها اضافه شد.

A screenshot of a computer screen

Description automatically generated

1. نرخ یادگیری: 0.0001

منحنی یادگیری دقت و صحت:

A graph of error

Description automatically generated

A graph of a graph showing the performance of a performance

Description automatically generated with medium confidence

اگر بخوهیم کمی نمودار را تحلیل کنیم واضح است که فرم بهتری گرفته اما مشکلی دارد این است مسئله داده کافی برای اضافه کردن این تعداد ویژگی ندارد و با اضافه کردند ویژگی ها دچار نفرین بعد می شود.

دقت و معیار f1 به سه دسته داده مد نظر:

A screenshot of a computer screen

Description automatically generated

ماتریس confusion برای سه دسته داده:

داده آموزش:

A screenshot of a graph

Description automatically generated

داده validation:

A screenshot of a graph

Description automatically generated

داده تست:

A screenshot of a graph

Description automatically generated

وزن های مدل آموزش داده شده:

A screenshot of a computer

Description automatically generated

با افزودن ضرب دو به دو این ویژگی‌ها، علاوه بر افزایش بعد، ویژگی‌هایی را نیز به مدل اضافه کردیم که نشان‌دهنده تعامل بین ورودی‌های اولیه ما است. به عبارت دیگر، با این کار، ما ارتباط و وابستگی بین ویژگی‌ها را بهتر مدل می‌کنیم. این اصطلاح را به گونه‌ای بیان کنیم که برداشت‌پذیری آن تشویق به ادامه کار کند.

می توان گفت در قسمت قبل ما توان های ویژگی ها را افزایش داده تا با استفاده از آن دقت بالا ببریم و در این جا دنبال ارتباط ویژگی های ساخته شده جهت افزایش دقت هستیم.در واقع این جا ما بد مسئله تا 3 افزایش دادیم تا بررسی کنیم ایا در بعد 3 جداپذیر خطی هست یا خیر.

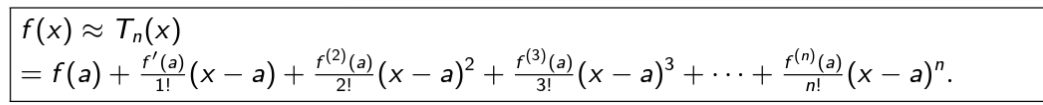
# تمرین 3

## 3-a:

سری تیلور یک ابزار ریاضی قدرتمند است که برای تقریب توابع با چند جمله‌ای استفاده می‌شود. گسترش سری تیلور یک تقریب چند جمله‌ای از رفتار تابع در اطراف یک نقطه مشخص فراهم می‌کند. با اضافه کردن جملات بالاتر در سری، می‌توان دقت بیشتری در تقریب تابع به دست آورد.

برای استفاده از سری تیلور برای تخمین یک تابع، معمولاً یک نقطه را انتخاب می‌کنیم که در آن تابع خوب رفتار کند، و سری را به یک نقطه معین قطع کرده وابسته به دقت مورد نظر و پیچیدگی تابع.

در شکل زیر از سری تیلور به این گونه استفاده می کنیم که میخواهیم مقدار تابع در نقطه x که نزدیک نقطه a است را حساب کنیم.

ثابت شده است که زمانی که n به سمت به نهایت می رود تابع ما همگرا می شود به مقدار واقعی.

## 3-b:

سری تیلور تقریبی از نقاط اطراف یک نقطه به ما می دهد.

A piece of paper with writing on it

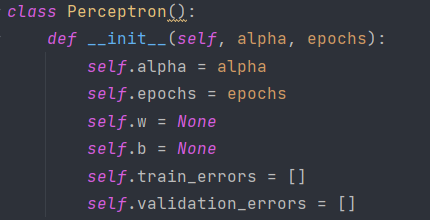
Description automatically generated

A piece of paper with writing on it

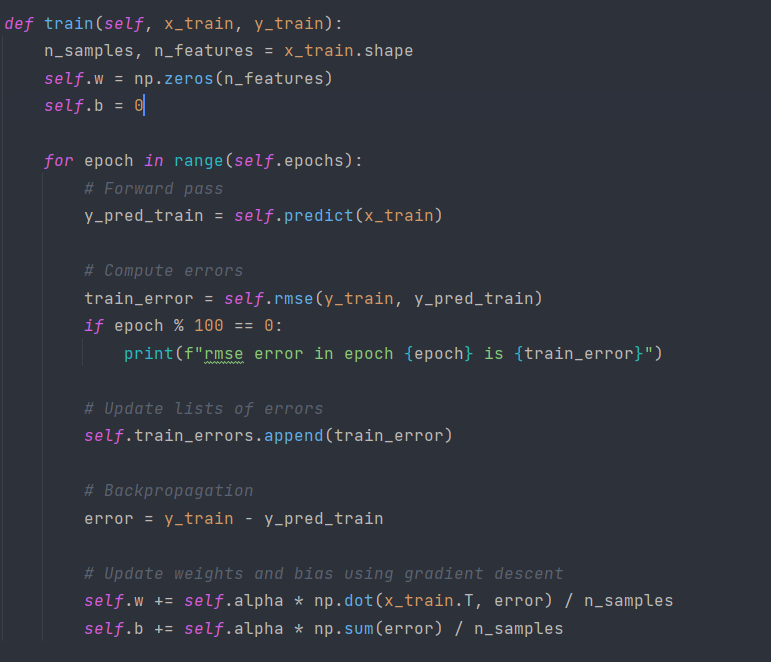
Description automatically generated

## 3-c:

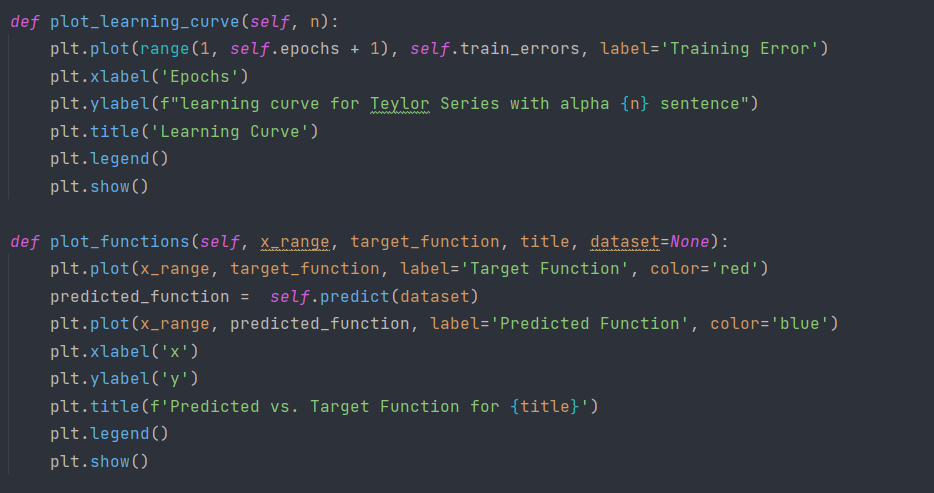
برای پیاده سازی صفر تا صد پرسپترون برای مسئله رگرسیون را انجام دادم.



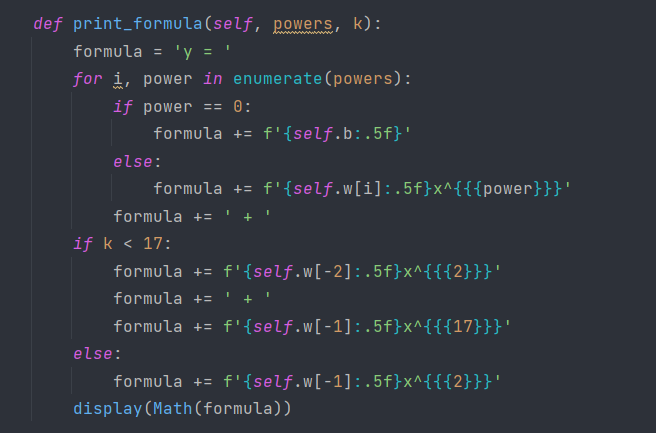
در تابع train آموزش پرسپترون انجام می شود گرادیان محاسبه می شود و backpropagate می شود تا وزن ها به روز شوند.



با استفاده از دو تابع زیر منحنی خطا و منحنی مقایسه تابع به دست امده از سری تیلور و تابع اصلی انجام می شود.



این تابع فرمول به دست آمده به شکل بهتری بر می گرداند که در خروجی ها مشاهده می کنید.



اطلاعات مورد نیاز که در مورد شبکه در اولین صفحه سوال خواسته شده:

1. تابع خطا: mse
2. بهینه ساز: گرادیان کاهشی
3. معماری شبکه: از یک پرسپترون استفاده کردیم.
4. تعداد گام آموزش: 100
5. اندازه دسته: برای آموزش کل داده به اموزش می دادم در واقعا اندازه برابر است با کل داده بنده.
6. آمارگان تفکیک داده: چون می خواستیم ظابطه یک تابع از طریق آموزش پرسپترون به دست آوریم همه داده ها داده آموزش در نظر گرفتم
7. پیش پردازش اعمال شده: هر دفعه یکی از توان ها فرد را به داده اضافه می کردم.
8. دیتاست: دیتاست ساخته شده به این صورت است داخل خود همیشه توان های 2 و 17 دارد و بعد از وارد حلقه می شود و هر یکی از درجه های فرد را به چند جمله ای اضافه می کند. دیتاست از -1 تا 1 با طول گام های 0.0001 ساخته شده است.
9. نرخ یادگیری: 1

A computer screen shot of a program

Description automatically generated

A screen shot of a computer code

Description automatically generated

تقریب سری تیلور به دست آمده بعد از آموزش پرسپترون با استفاده از یک جمله تقریب:

فرمول به دست آمده:

A number on a black background

Description automatically generated

مشاهده فرآیند آموزش:

A screenshot of a computer program

Description automatically generated

منحنی خطا:

A graph on a screen

Description automatically generated

نمودار به دست آمده با استفاده از تقریب و مقایسه آن با نمودار اصلی:

A graph with a curved line

Description automatically generated

تقریب سری تیلور به دست آمده بعد از آموزش پرسپترون با استفاده از دو جمله تقریب:

فرمول به دست آمده:

A black background with white text and a plus and a white line

Description automatically generated

مشاهده فرآیند آموزش:

A screenshot of a computer

Description automatically generated

منحنی خطا:

A graph with a line

Description automatically generated

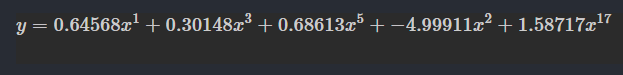
نمودار به دست آمده با استفاده از تقریب و مقایسه آن با نمودار اصلی:

A graph of a function

Description automatically generated

تقریب سری تیلور به دست آمده بعد از آموزش پرسپترون با استفاده از سه جمله تقریب:

فرمول به دست آمده:



مشاهده فرآیند آموزش:

A screenshot of a computer screen

Description automatically generated

منحنی خطا:

A graph of a training curve

Description automatically generated

نمودار به دست آمده با استفاده از تقریب و مقایسه آن با نمودار اصلی:

A screen shot of a graph

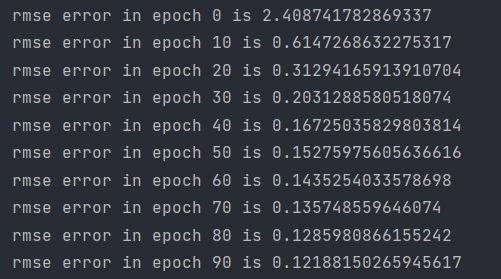
Description automatically generated

تقریب سری تیلور به دست آمده بعد از آموزش پرسپترون با استفاده از چهار جمله تقریب:

فرمول به دست آمده:



مشاهده فرآیند آموزش:



منحنی خطا:

A graph of a training curve

Description automatically generated

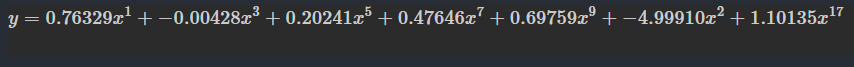
نمودار به دست آمده با استفاده از تقریب و مقایسه آن با نمودار اصلی:

A graph of a function

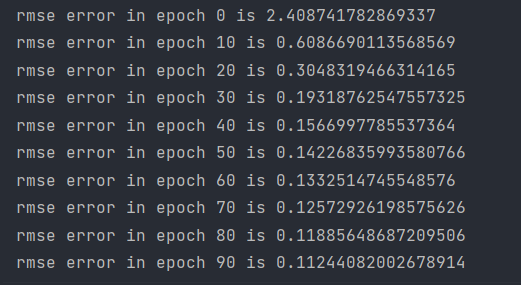
Description automatically generated

تقریب سری تیلور به دست آمده بعد از آموزش پرسپترون با استفاده از پنج جمله تقریب:

فرمول به دست آمده:



مشاهده فرآیند آموزش:



منحنی خطا:

A graph on a black background

Description automatically generated

نمودار به دست آمده با استفاده از تقریب و مقایسه آن با نمودار اصلی:

A graph on a screen

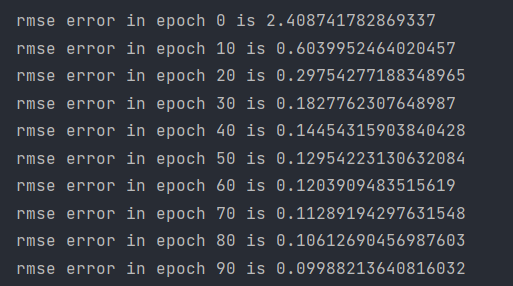
Description automatically generated

تقریب سری تیلور به دست آمده بعد از آموزش پرسپترون با استفاده از شش جمله تقریب:

فرمول به دست آمده:



مشاهده فرآیند آموزش:



منحنی خطا:

A graph of a training

Description automatically generated

نمودار به دست آمده با استفاده از تقریب و مقایسه آن با نمودار اصلی:

A graph with a red and blue line

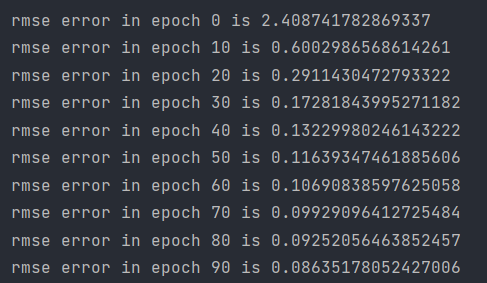
Description automatically generated

تقریب سری تیلور به دست آمده بعد از آموزش پرسپترون با استفاده از هفت جمله تقریب:

فرمول به دست آمده:



مشاهده فرآیند آموزش:



منحنی خطا:

A graph of a training curve

Description automatically generated

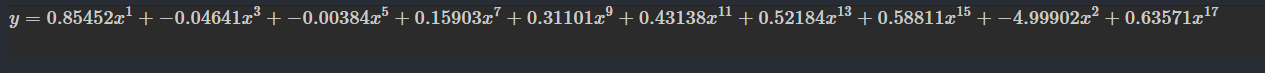
نمودار به دست آمده با استفاده از تقریب و مقایسه آن با نمودار اصلی:

A graph of a function

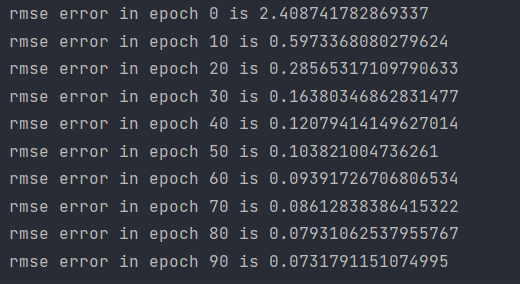
Description automatically generated

تقریب سری تیلور به دست آمده بعد از آموزش پرسپترون با استفاده از هشت جمله تقریب:

فرمول به دست آمده:



مشاهده فرآیند آموزش:



منحنی خطا:

A graph with a line

Description automatically generated

نمودار به دست آمده با استفاده از تقریب و مقایسه آن با نمودار اصلی:

A graph with a curved line

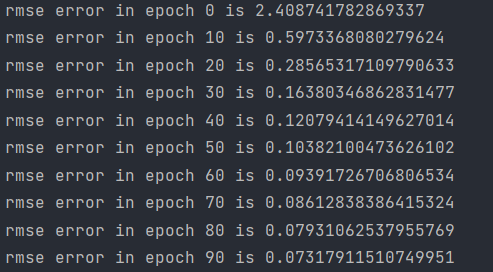
Description automatically generated

تقریب سری تیلور به دست آمده بعد از آموزش پرسپترون با استفاده از نه جمله تقریب:

فرمول به دست آمده:



مشاهده فرآیند آموزش:



منحنی خطا:

A graph on a black background

Description automatically generated

نمودار به دست آمده با استفاده از تقریب و مقایسه آن با نمودار اصلی:

A graph of a function

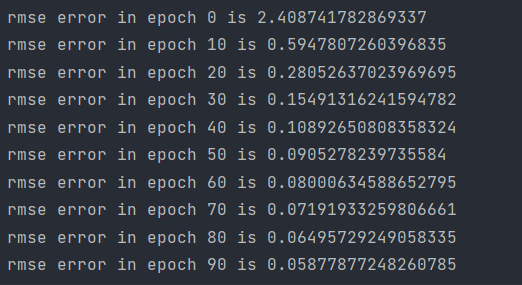
Description automatically generated

تقریب سری تیلور به دست آمده بعد از آموزش پرسپترون با استفاده از ده جمله تقریب:

فرمول به دست آمده:



مشاهده فرآیند آموزش:



منحنی خطا:

A graph of a training curve

Description automatically generated

نمودار به دست آمده با استفاده از تقریب و مقایسه آن با نمودار اصلی:

A graph with a curved line

Description automatically generated

در نهایت تجمیع تمام شکل های به دست امده بر روی یک نمودار:

A screen shot of a black screen

Description automatically generated