به نام خدا

گزارش تمرین سری اول درس شبکه عصبی

زهرا دهقانیان

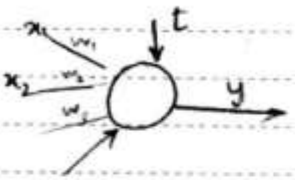
۹۸۱۳۱۰۵۹

سوال اول)

سوال 1)

پرسپترون ... واحد عصبی پرسپترون در سال 1957 توسط روزنبلت معرفی شد

ساختار این واحد به این صورت می باشد

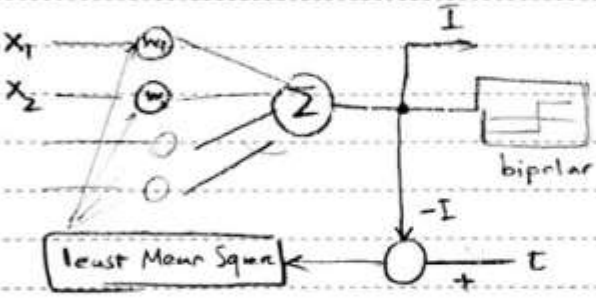


قانون یادگیری این واحد به این صورت می باشد

$$W_{new} = W_{old} + (t - y)\,X$$

X W

آدالاین : در سال 1959 توسط ویدرو معرفی شد. ساختار این واحد به این صورت می باشد



قانون یادگیری بدین صورت

$$W_{new} = W_{old} + \alpha\,(t - I)$$

$$0 < \alpha < \frac{2}{\lambda_{max}}$$

⚠ این دو مدل هر دو classifier های خطی هستند. تفاوت اصلی در قانون های یادگیری

این دو مدل پرسپترون برای آپدیت کردن وزن ها از خروجی دیجیتالی (y) استفاده میکند

(که یک مقدار 0 / 1 است) اما آدالاین از I یا net input - یا همان حاصل جمع وزن دار

ورودی ها استفاده میکند که یک مقدار پیوسته است. این استفاده باعث میشود که آپدیت

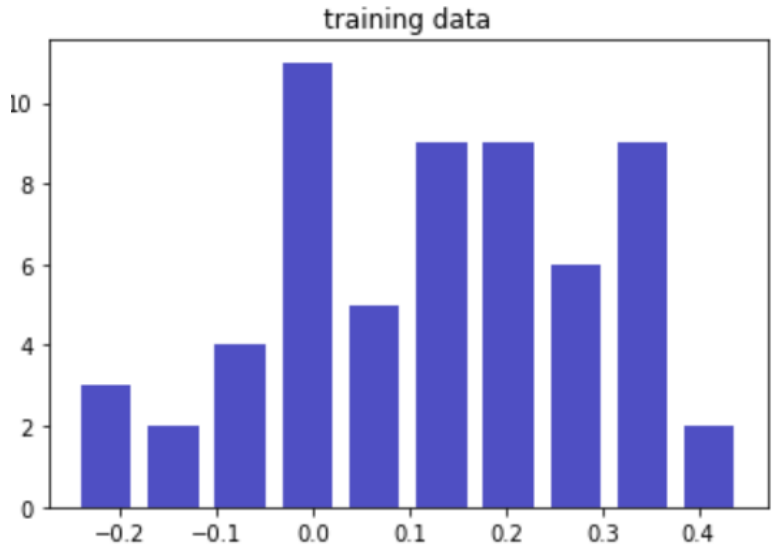
ملاقات های مهم :

بهتر و دقیق تر صورت بگیرد و مدل سریع تر به

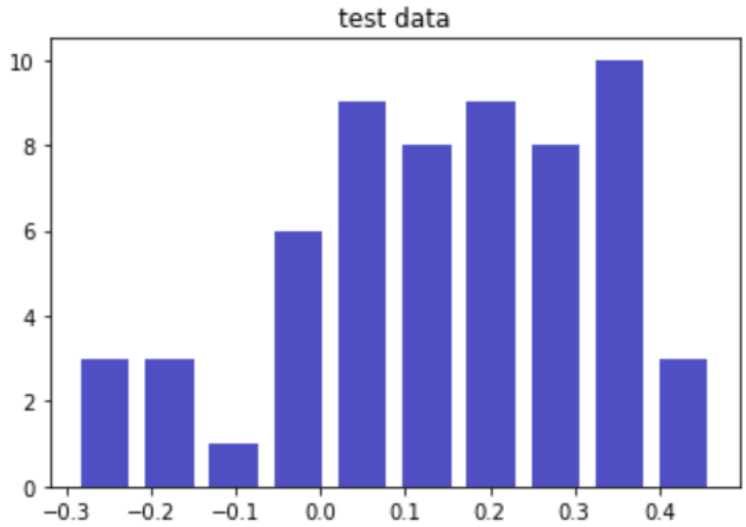
مدل نهایی همگرا شود

یکشنبه	دوشنبه	سه شنبه	چهارشنبه	پنجشنبه	جمعه
		۱	۲	۳	۴
۶	۷	۸	۹	۱۰	۱۱
۱۳	۱۴	۱۵	۱۶	۱۷	۱۸
۲۰	۲۱	۲۲	۲۳	۲۴	۲۵

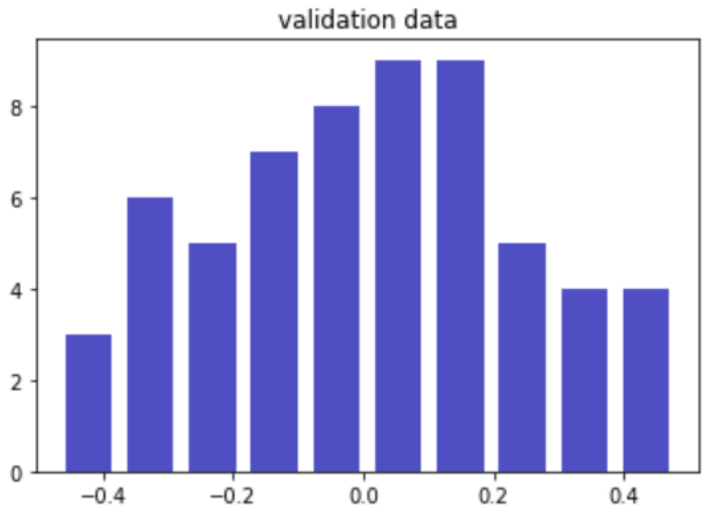
سوال دوم) در این بخش برای تقسیم داده ها از تابع np.split و np.random.shuffle استفاده کرده ایم. برای محاسبه ضریب همبستگی نیز، از تابع numpy.correlation بکار بردیم و برای نمایش هیستوگرام نیز از تابع plt.hist استفاده کردیم. هیستوگرام برای داده های اموزش به این صورت می باشد:



این نمودار برای داده ی تست به صورت زیر است :



و نمودار هیستوگرام برای داده validation به صورت زیر است :



در ادامه ویژگی هایی که در هر سه دسته همبستگی آن ها بیشتر از ۰,۳ (مثبت/منفی) می باشد را به عنوان ویژگی ها مطلوب انتخاب میکنیم. این ویژگی ها به همراه میزان همبستگی آن ها در هر کدام از مجموعه داده ها به صورت زیر است :

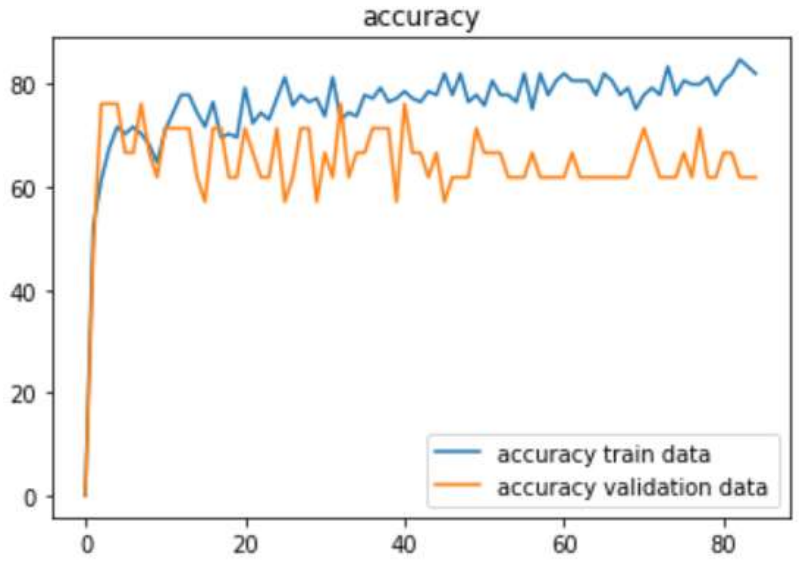
```
10.0
[0.4422181515023248, 0.4422181515023248, 0.4422181515023248]
48.0
[0.3889130654820476, 0.3889130654820476, 0.3889130654820476]
11.0
[0.3856028900883182, 0.3856028900883182, 0.3856028900883182]
44.0
[0.36831913106336134, 0.36831913106336134, 0.36831913106336134]
8.0
[0.3529229235272831, 0.3529229235272831, 0.3529229235272831]
9.0
[0.352636963430212, 0.352636963430212, 0.352636963430212]
45.0
[0.3414481833814232, 0.3414481833814232, 0.3414481833814232]
47.0
[0.3339525960767587, 0.3339525960767587, 0.3339525960767587]
12.0
[0.3217677480474085, 0.3217677480474085, 0.3217677480474085]
46.0
[0.3159814218284191, 0.3159814218284191, 0.3159814218284191]
```

همان طور که مشخص است ویژگی های همبسته تر با برچسب داده، ویژگی های شماره ۹،۸،۴۴،۱۱،۴۸،۱۰، ۴۵ ، ۴۷، ۱۲ و ۴۶ می باشند.

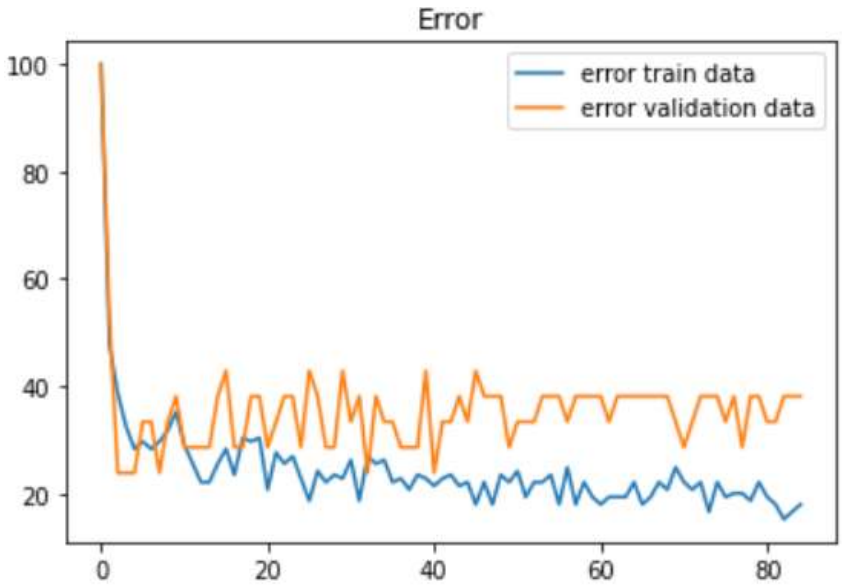
سوال سوم)در این بخش تابع پرسپترون را در دو بخش training_weights و predict پیاده سازی می کنیم. در بخش یادگیری همانند قانون مربوط به پرسپترون در صورتی که برچسب متفاوت باشد، وزن ها را اپدیت می کنیم. درابتدا برای پایان دادن اموزش، شرط early sttoping یعنی افزایش نیافتن دقت / بالا رفتن خطا برای داده validation را استفاده کردیم اما این شرط کارآمد نبود و در نهایتا ۵ ایپک اول اموزش را متوقف میکرد و ضرایب بدست امده تا جواب بهینه فاصله بسیار زیادی داشت به همین دلیل این شرط را غیرفعال کردیم و در کنار شرط تعداد ایپک از شرط همگرایی نیز استفاده کردیم. این شرط را به این صورت پیاده سازی کردیم که اگر اختلاف خطا بین دو ایپک در ۵۰ ایپک کمتر از ۲ تا تخمین زده شود، به همگرایی رسیدیم و از حلقه خارج شده و اموزش را پایان میدهیم. خروجی نهایی برای داده تست:

```
>>>>>>>>>>Convegence condition :
training finish at epoch 85
--------------------
Final Accuracy test data = 80.19047619047619
```

با توجه به این که خروجی باید برچسب کلاس داده باشد. در این جا از تابع فعال سازی پله استفاده میکنیم. برای مقادیر تتا و بتای این تابع ۱۲ نوع ترکیب پارامتر بررسی شد (تتا = ۰ و ½ و ۳/۱ و ¼ و ۱) (بتا = ۰ و -۱ و -۲/۱) از بین این مقادیر، تابع فعال سازی که بهترین جواب را داشت همان تابع پیشنهادی رزنبلت بود. نمودار دقت در حین آموزش به صورت زیر است:



نمودار خطا برای داده اموزش و validation در شکل زیرامده است:



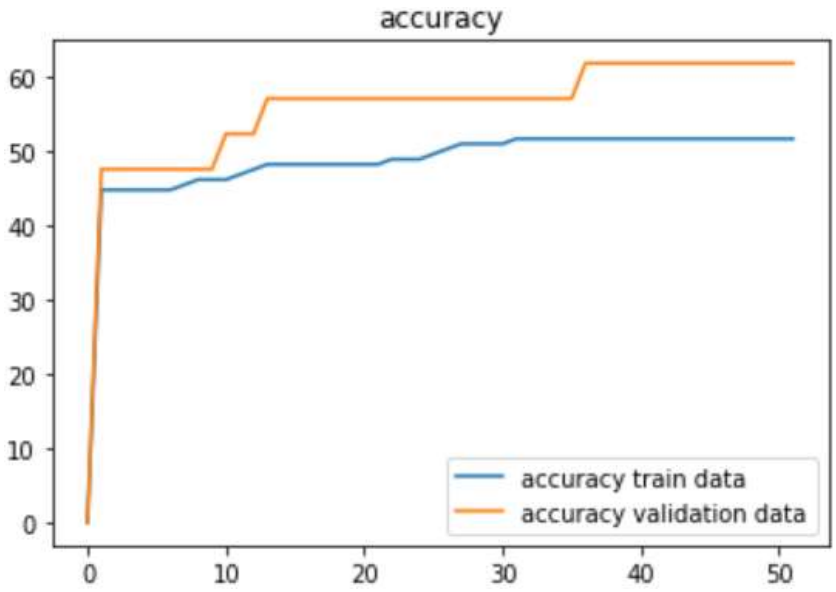
همان طور که دیده می شود شرط همگرایی رو داده های آموزش دقت نسبتا خوبی ۷۶٪ را دارد اما به بهترین جواب دست نمی یابد.

سوال چهارم) ساختار آدالین شباهت زیادی به پرسپترون دارد و تنها در قانون اپدیت کردن وزن ها تفاوت دارند. در اینجا به جای این که کیفیت خروجی نهایی شبکه (خروجی از تابع فعال سازی) را برای اپدیت کردن بکارگیریم، از net input استفاده میکنیم ، یعنی خطا به این صورت محاسبه می شود که برچسب اصلی داده منهای net input شده و در ضریب یادگیری ضرب شده و وزن جدید محاسبه می شود. خروجی این بخش به این صورت است:

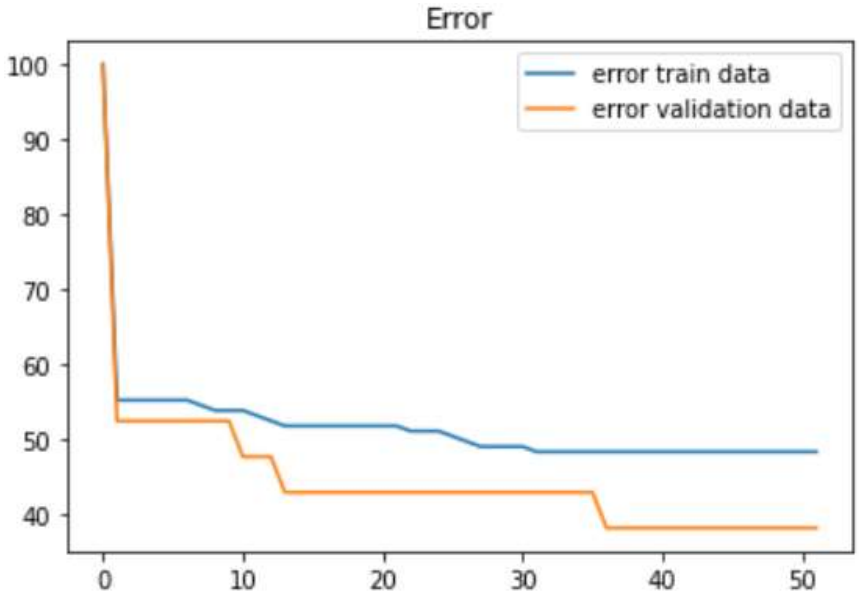
```
>>>>>>>>>>Convegence condition :
training finish at epoch 51
--------------------
Final Accuracy test data = 78.57142857142857
```

نمودار های دقت و خطا برای داده های تست و validation در ادامه امده است :

نمودار دقت :



نمودار خطا :



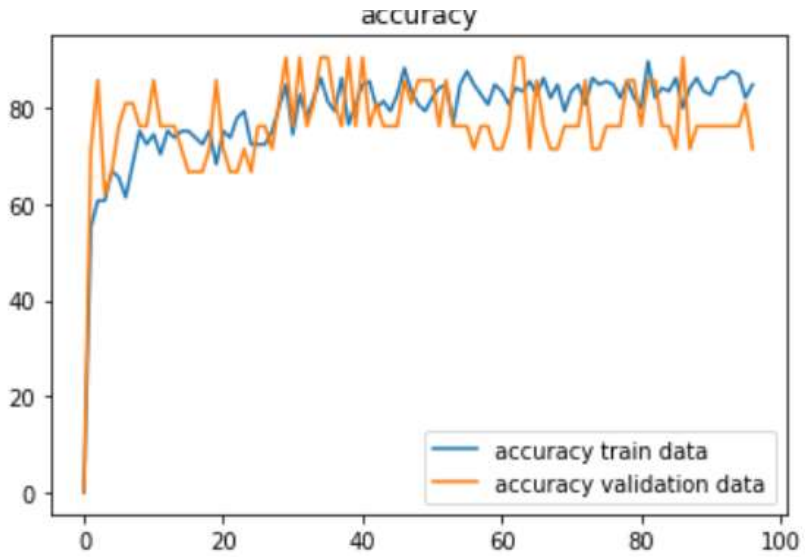
همانطور که در نمودار دیده می شود، نمودار ادالین کمتر تغییرات شدید دارد (spiky) و وضعیت نسبتا پایدار تری دارد . همچنین این واحد دقت مناسبی و با کمی بهبود (۷۸٪) دارد

سوال پنجم) در آدالین ، مدل درجه ۲ به این صورت است که علاوه بر ورودی داده خود x ها تمامی ترکیب های ۲تایی از هر دو x دلخواه را هم به عنوان ورودی میدهیم . به عنوان مثال اگر ورودی تنها ۲ بعد داشته باشد، تعداد نود های ورودی هادالین درجه ۲ ، ۵ تاست و آن ها این هستند: $x1, x2, x1x2, x1^2, x2^2$ و ادامه اموزش همانند ادالین درجه ۱ می باشد.

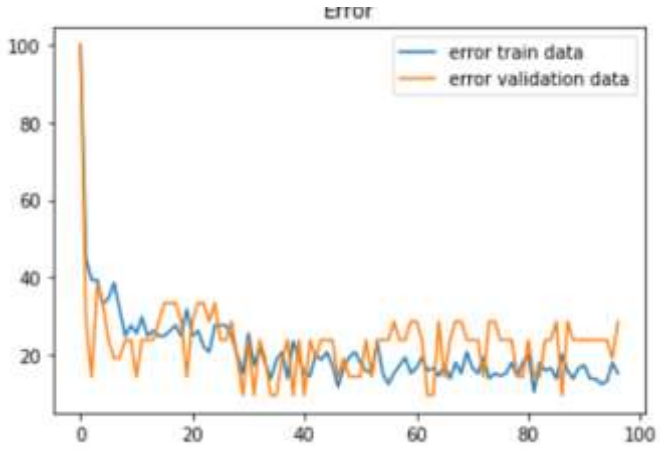
در پرسپترون درجه ۲ به این صورت می باشد که به ازای هر ورودی ، خودش و توان ۲ ان را هم به شبکه می دهیم و ادامه کار شبیه پرسپترون معمولی می باشد. در این بخش به پیاده سازی پرسپترون درجه۲ پرداخته ایم، خروجی این شبکه به ازای داده های مورد ازمایش در بخش قبل به صورت زیر است :

```
>>>>>>>>>>Convegence condition :
training finish at epoch 96
--------------------
Final Accuracy test data = 64.28571428571429
```

همان طور که در خروجی کد آمده است، این واحد دقت کمتری به نسبت واحد های قبلی بدست اورده است و بر خلاف انتظار ما کیفیت دسته بندی را بهبود زیادی نداده است.نمودار دقت به صورت زیر است:



نمودار خطا به صورت زیر است:



از نظر سرعت همگرایی پرسپترون درجه ۲ ، کندتر همگرا شده و سرعت یادگیری و اپدیت کردن در هر ایپک کمتر بوده که در نهایت منجر به کاهش سرعت آموزش شده است. در ادامه ماتریس درهم ریختگی پرسپترون درجه یک و دو را با هم می بینیم:

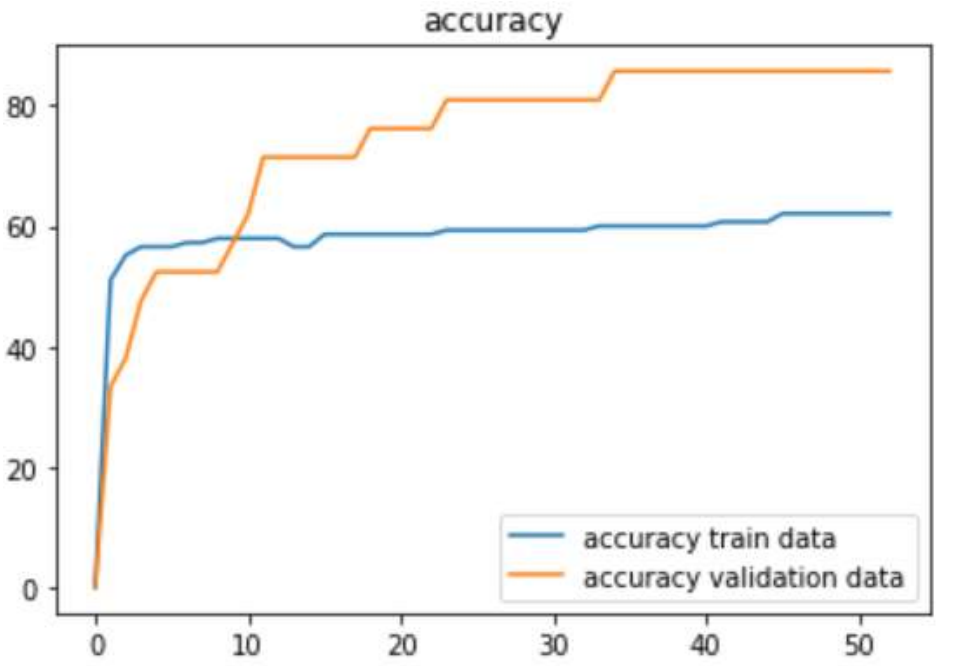
```
Confusion matrix Perceptron :
[[18  0]
 [ 8 16]]
----------------------------
Confusion matrix Seccond order Perceptron :
[[18  0]
 [15  9]]
```

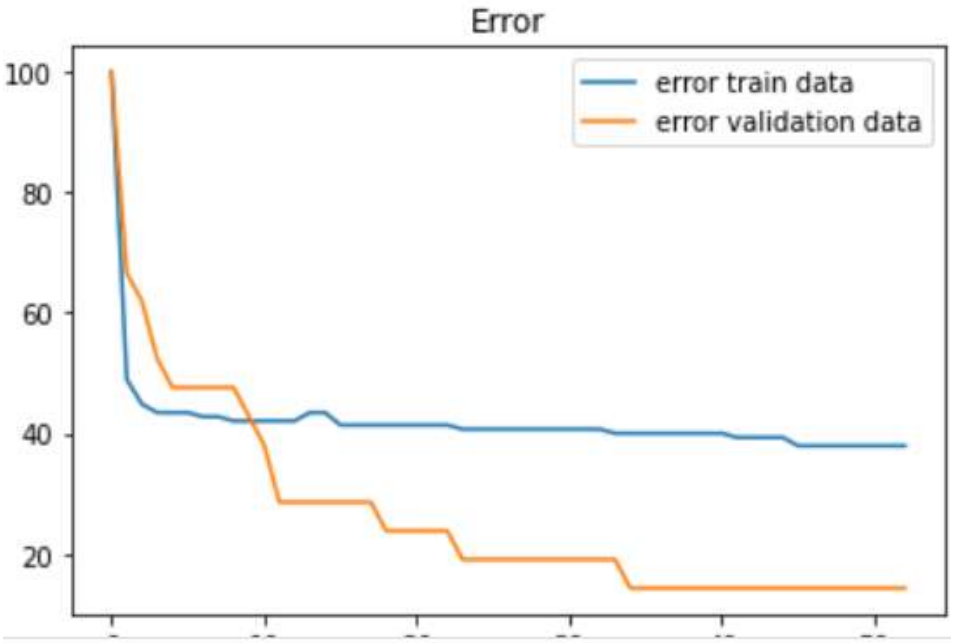
همان طور که می بینید، نرخ false negetive در پرسپترون درجه دو بسیار که این به این معناست که اگر که تشخیص دهنده بگوید سنگ معدنی است حتما معدنی می باشد، اما اگر عادی تلقی شود خیلی قابل اعتماد نیست.

اما برای پیاده سازی ادالین درجه ۲ دقت بسیار بالاتر است. دقت بدست آمده برای ادالین درجه ۲ برابر با ۹۲ درصد می باشد که دقت بسیار بالایی می باشد. خروجی به صورت زیر است :

```
>>>>>>>>>>Convegence condition :
training finish at epoch 52
--------------------
Final Accuracy test data = 92.85714285714286
```

در اینجا به صورت عمل میکنیم که ابتدا حاصل ضرب های دوتایی و تکی هر کدام از ویژگی ها را حساب میکنیم (تعداد = ویژگی۱۸۳۱) و سپس ادالین درجه ۱ فراخوانی می شود. سرعت همگرایی در این نود در تعداد ایپک کمتری (۵۲ ایپک می باشد) اما زمان اجرا برای این عنصر بسیار بیشتر از ادالین درجه ۱ بود(برای ادالین درجه۲ ، ۲۱ ثانیه اما برای درجه ۱، کمتر از ۲ ثانیه). نمودار دقت و خطا به صورت زیر است :





ماتریس درهم ریختگی برای ادالین درجه ۱ و ۲ به صورت زیر می باشد:

```
Confusion matrix adaline :
[[ 3 15]
 [ 0 24]]
----------------------------
Confusion matrix Seccond order adaline:
[[16  2]
 [ 1 23]]
```

همان طور که می بینید، ادالین درجه ۱ false positive خیلی بالایی دارد و اگر بگوید که یک سنگ معدنی است خیلی قابل اعتماد نیست. اما ادالین درجه ۲ کیفیتی بسیار مناسب داشته و با دقت خوبی توانسته جداسازی انجام دهد.

****قابل توجه است که کد این تمرین و به همراه نتیایج و روال پیشرفت آن در گیت هاب بنده به آدرس زیر موجود است :**

https://github.com/zahraDehghanian97/Neural_net_HW