بسمه تعالی

تمرین ۲ درس هوش محاسباتی

علی صفرپوردهکردی
۹۵۵۲۱۲۷۹

MLP.۱

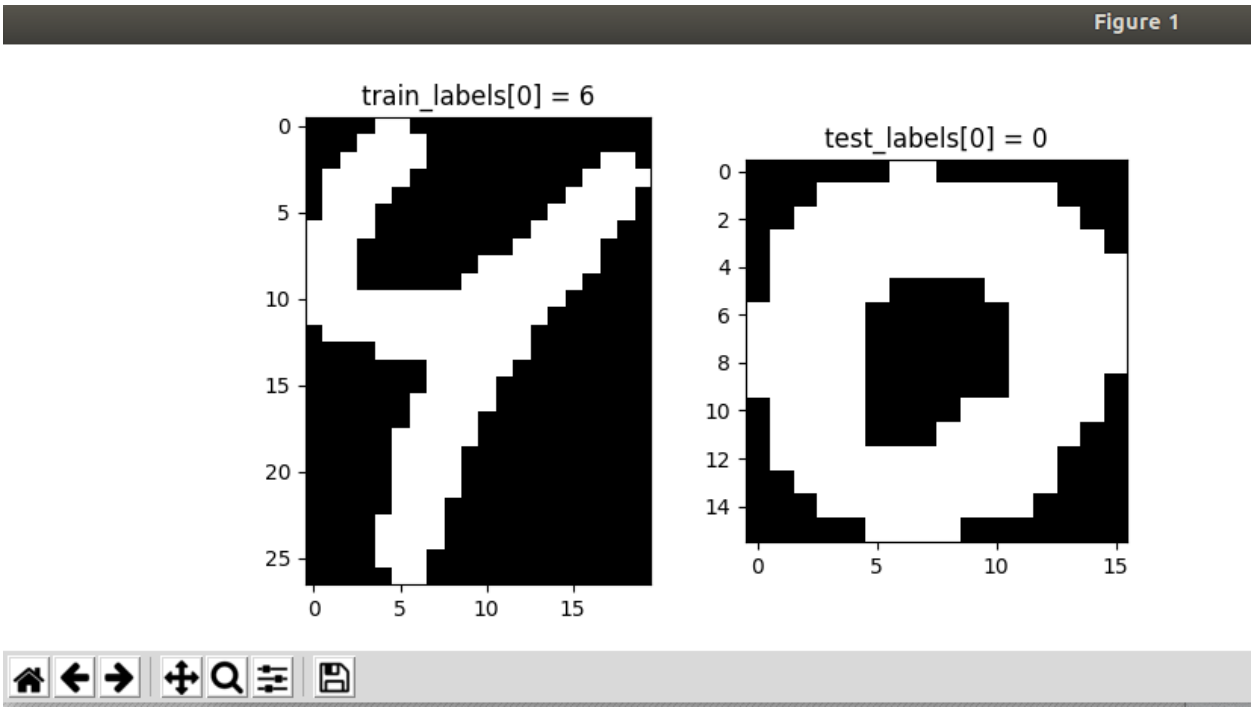
a.

در این قسمت صرفا کد مربوط به بخش اول را اجرا مینماییم.
توضیح کد: ابتدا باید HodaDatasetReader را دریافت نموده باشیم، سپس با خواندن اطلاعات موجود در آن و دسته بندی اطلاعات به دو دسته train و test و label گذاری هریک داده ها آماده خواهند بود.لازم به ذکر است دسته ای دیگر از داده ها به نام remainingنیز وجود دارد که در این تمرین از آن استفاده ای نمی نماییم.همچنین در طول تمرین ها از matplotlib برای نشان دادن شکل های مختلف استفاده میگردد.

فایل مربوط به کد : 1_a.py

برای اجرای کد ها از python3 استفاده گردد.

نمونه خروجی:



---

b.

در این قسمت ابتدا بدنه کامل کد مربوط به سوال را برای keras پیاده میکنیم .

اشاره به چند مورد:

لیبل ها به صورت one_hot هستند.

تصاویر تست و تمرین را به مقیاس ۱۶*۱۶ برده ایم تا هم یکسان باشند هم برای ترین اطلاعات کمتری را یاز باشد تفسیر نماییم.

لایه خروجی دارای ۱۰ نورون باید باشد چراکه خروجی به صورت one_hot برای ۱۰ کلاس تعیین میگردد که به ترتیب ارقام ۰ تا ۹ میباشند.

بخش های معماری که معرفی نشده اند بر اساس کد های رایج mnist نوشته شده اند.

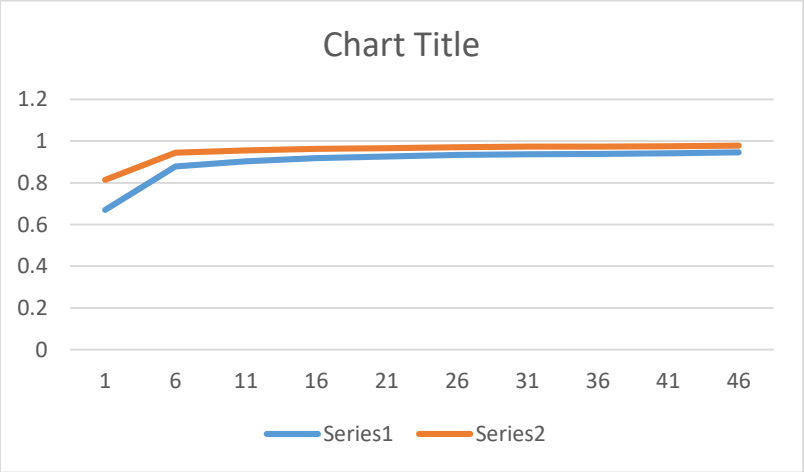
در کلاس sigmoid معرفی گردیده بود ، ولی به این علت که خروجی مطلوبی نداشت از آن استفاده نگردید.

ورودی به صورت یک بردار ۲۵۶ تایی میباشد.

فایل مربوط به کد: 1_b.py

البته فایل بالا صرفا به علت سادگی اولیه معرفی گردید و برای تست های زیر در این قسمت و قسمت های بعد از فایل 1_c.py استفاده گردیده است.البته کد مربوط به قسمت نهایی است و در هر قسمت به مرور تکمیل شده است.همچنین ۲ لایه معرفی گردیده است.

تعداد epoch	صحت برای دیتا تست شده	صحت برای دیتای تمرین شده
1	0.66945	0.814116667
6	0.87885	0.944766667
11	0.90335	0.955933333
16	0.918	0.963483333
21	0.9256	0.96665
26	0.9335	0.970233333
31	0.937	0.9734
36	0.93925	0.97455
41	0.9415	0.97605

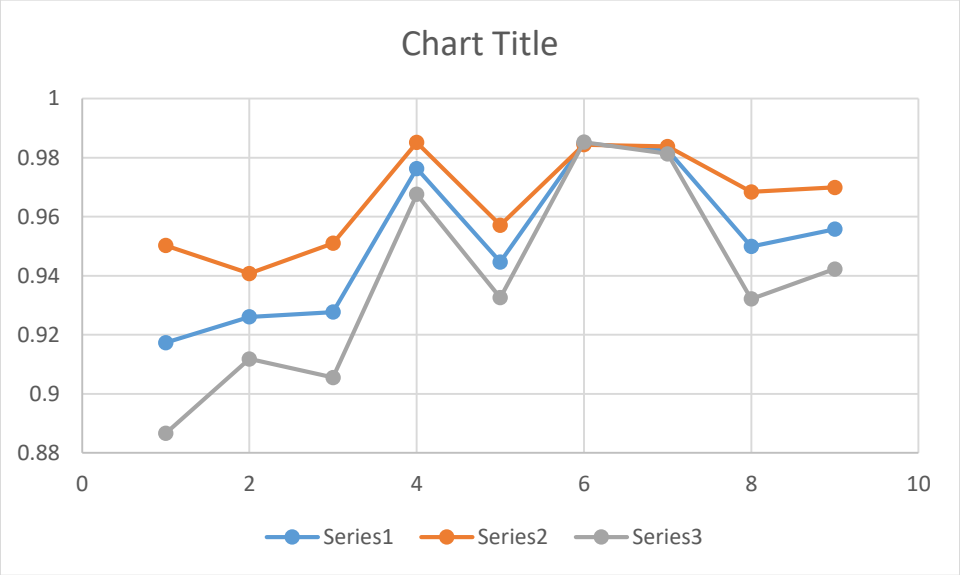


لازم به ذکر است در نمونه فوق batch_size برابر با ۱۰۰ بوده است.

همچنین precision ,f1, recall نیز محاسبه گردیده اند.

نحوه محاسبه: در پایان اجرا و تست کردن نمونه ورودی ها و نمونه تست ها با محاسبه tp,np,fp,np بر اساس روابط مربوطه و محاسبه فرمول های لازمه اطلاعات زیر کسب شده است.

		f1	recall	precision
class	2	0.9173404	0.95025	0.886634
class	3	0.926049	0.94075	0.9118003
class	4	0.9276917	0.951	0.9054987
class	5	0.9762775	0.985125	0.9675875
class	6	0.9446672	0.957125	0.9325295
class	7	0.9848059	0.984375	0.9852371
class	8	0.9825218	0.98375	0.9812968
class	9	0.9499111	0.968375	0.9321381
total	total	0.9557512	0.9699	0.9422494



.c

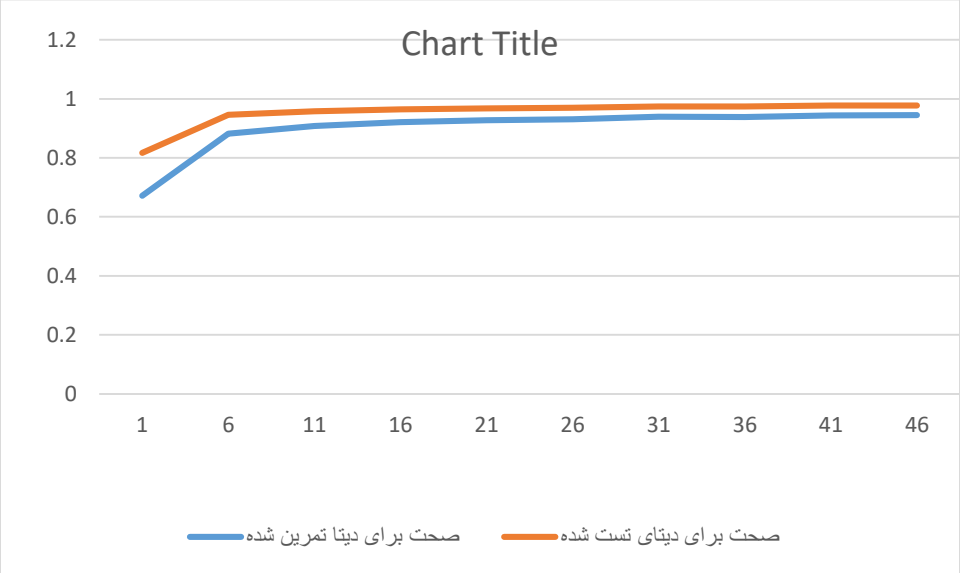
Drop-out : این اصطلاح به این معناست که در هر بار تعدادی از نورون ها را به صورت رندم از سیستم لرن و پیشبینی به صورت موقت خارج کنیم.در این صورت این مزیت ایجاد میگردد که در صورتی که سیستم دچار اخلال شد و همه نورون ها نتوانند کار کنند سیستم کارایی خود را از دست نمیدهد.به علاوه در راستای کاهش اوورفیت شدن نیز موثر است.

نمودارهایی که در ادامه مشاهده مینمایید مانند نمودار های قبلی ایجاد گردیده است.صرفا یک دراپ اوت در بین دو لایه استفاده نموده ایم.

دستور زیر در کد اضافه شده است:

Dropout(.05, noise_shape=None, seed=None)

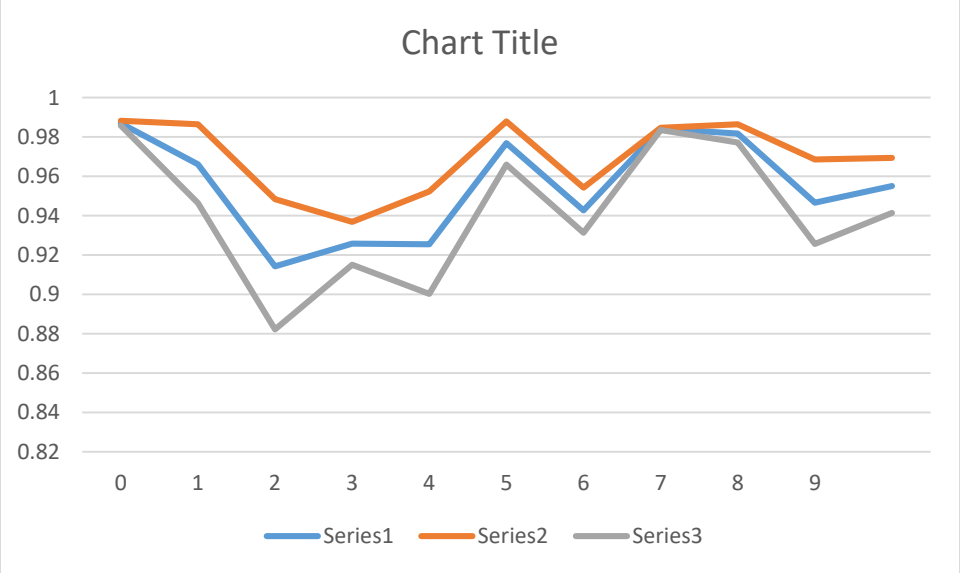
تعداد epoch	صحت برای دیتا تست شده	صحت برای دیتای تمرین شده
1	0.67165	0.8163
6	0.88175	0.945583333
11	0.9082	0.958083333
16	0.92065	0.963716667
21	0.92775	0.967533333
26	0.931	0.970166667
31	0.93885	0.973766667
36	0.9381	0.9738
41	0.94325	0.97685
46	0.9446	0.977716667



همچنین اطلاعات زیر بدست آمد:

Column1	Column2	Column3	Column4	Column5
		f1	recall	precision
class	0	0.9870162	0.98825	0.9857855
class	1	0.9660913	0.9865	0.94651
class	2	0.9142169	0.9485	0.8823256

class	3	0.925823	0.936875	0.9150287
class	4	0.9255255	0.95225	0.90026
class	5	0.9768247	0.987875	0.9660188
class	6	0.9427019	0.95425	0.93143
class	7	0.9841349	0.98475	0.9835206
class	8	0.9818363	0.9865	0.9772164
class	9	0.946674	0.968625	0.9256959
total		0.9550845	0.9694375	0.9413791



.d

Validation Set : دیتا را به صورت دسته بندی استفاده مینماید به این صورت که هر بار بخشی از دیتا را لرن کرده و بخشی از دیتای ترین را برای تست استفاده میکند.

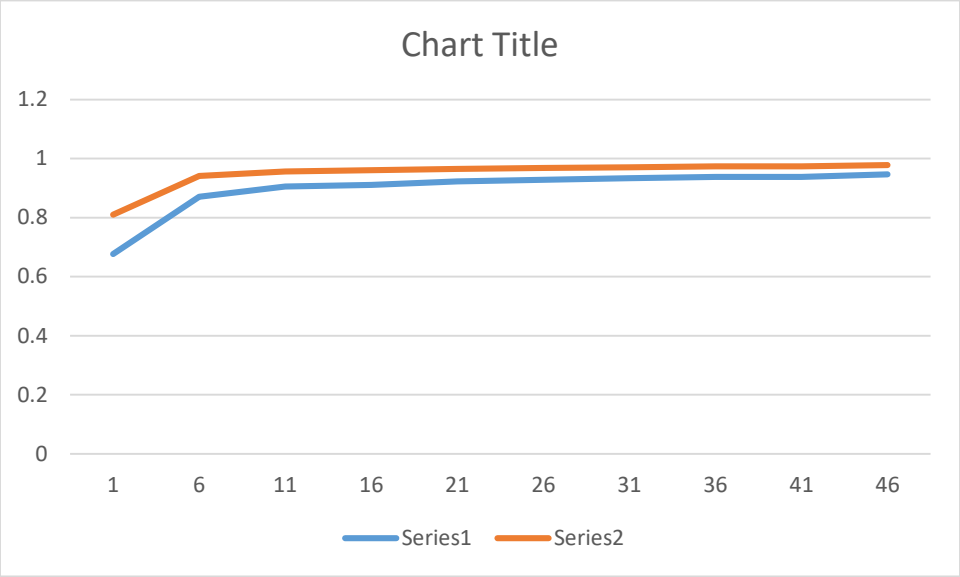
در نتیجه برای تست و همچنین فرمت کلی دیتا ترین اورفیت نمیکند.

در کد در قسمت fit این حالت را افزوده ایم و میزان آن ۰.۱ میباشد:

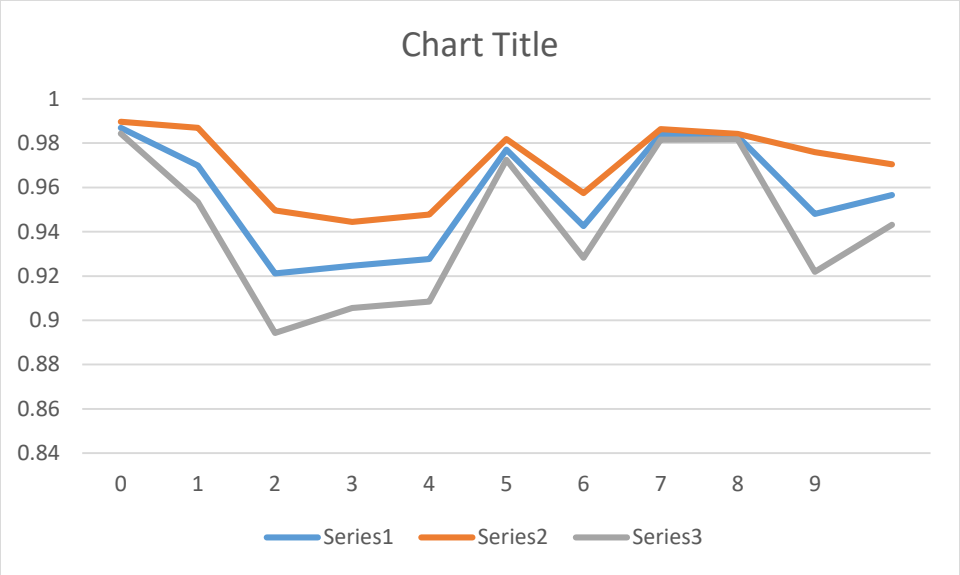
```
network.fit(x_train, y_train, epochs=epoch_num, batch_size=batch_num,validation_split=0.1)
```

تعداد epoch	صحت برای دیتا تست شده	صحت برای دیتای تمرین شده
1	0.6763	0.809366667
6	0.87045	0.94105
11	0.9048	0.956316667
16	0.91115	0.960133333
21	0.9227	0.964816667

26	0.9276	0.968183333
31	0.9331	0.970516667
36	0.9382	0.97385
41	0.9379	0.973383333
46	0.9468	0.9783



Column1	Column2	Column3	Column4	Column5
		f1	recall	
class	0	0.9869725	0.989625	0.9843342
class	1	0.9699054	0.987	0.9533929
class	2	0.9211276	0.949625	0.8942908
class	3	0.9246207	0.9445	0.9055609
class	4	0.9276887	0.94775	0.9084591
class	5	0.9771724	0.981875	0.9725145
class	6	0.9425881	0.957375	0.9282511
class	7	0.9839766	0.986375	0.9815897
class	8	0.9830212	0.98425	0.9817955
class	9	0.9480843	0.975875	0.9218326
total		0.9565157	0.970425	0.9432021



همانطور که مشاهده مینمایید در این حالت میزان صحت بالاتری بدست آمده اسن و در ایپاک های بالاتر نیز درصد صحت افتی نکرده است.

---

e.

Batch_size: : در هر مرحله از یادگیری تعدادی از دیتا ها که همزمان برای ورود به شبکه انتخاب میشوند را تعیین میکند.
این اندازه در صورتی که خیلی کوچک باشد یادگیری کند و در صورتی که خیلی بزرگ باشد میزان صحت کاهش می یابد.

Batch_mode : حالتی که کل دیتا را انتخاب کرده باشیم

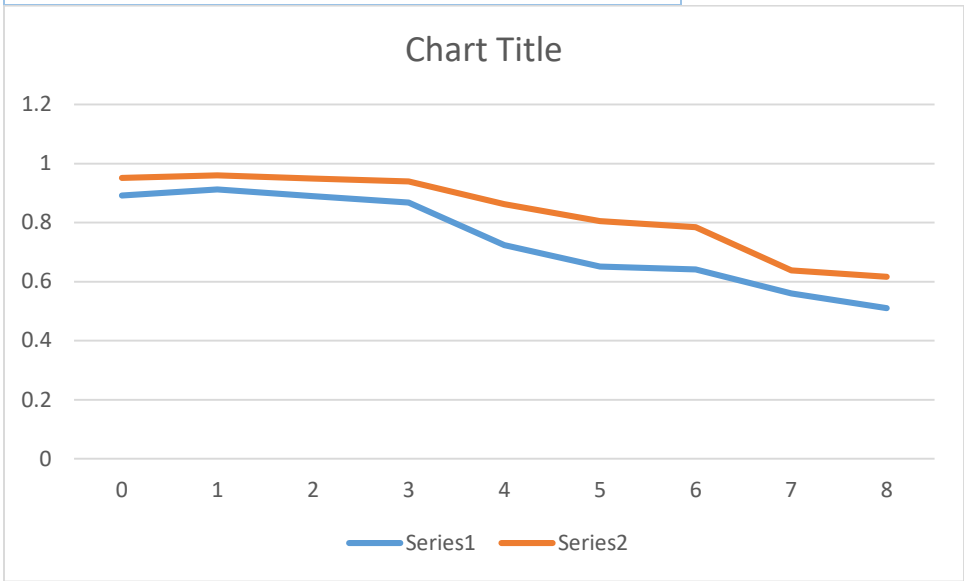
Mini-batch mode: حالت بینابینی است.

Stotastic mode: در صورتی که اندازه بچ ۱ باشد.

در زیر برای اندازه های مختلف بچ اطلاعات را مشاهده مینمایید.قابل ذکر است در این مورد تعداد ایپاک برابر ۶ بوده است.
همچنین کدی که در فایل رویت مینمایید کد بعد از تست این قسمت سوال است.

batch_size	صحت برای دیتا تست شده	صحت برای دیتای تست شده
1	0.89175	0.95085
10	0.9125	0.95988333
50	0.88955	0.94925
100	0.8678	0.93983333
500	0.7241	0.8629
1000	0.6509	0.80528333
10000	0.6414	0.78408333

30000	0.5601	0.63835
60000	0.51045	0.61636667



همانطور که مشاهده میگردد در ستون های اول و دوم وسوم بهترین عملکرد ها را مشاهده مینماییم و به طور خاص تر ستون دوم یعنی مقدار ۱۰ بهترین جواب است.

Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9
batch_size=1								
class	0	:	f1=	0.9499907	recall=	0.958125	precision=	0.9419934
class	1	:	f1=	0.9228305	recall=	0.972375	precision=	0.8780901
class	2	:	f1=	0.837877	recall=	0.86825	precision=	0.8095571
class	3	:	f1=	0.8774599	recall=	0.9085	precision=	0.8484707
class	4	:	f1=	0.8724977	recall=	0.907125	precision=	0.8404169
class	5	:	f1=	0.9418157	recall=	0.964125	precision=	0.9205156
class	6	:	f1=	0.8726164	recall=	0.912375	precision=	0.8361783
class	7	:	f1=	0.9597438	recall=	0.955125	precision=	0.9644074
class	8	:	f1=	0.9660273	recall=	0.96325	precision=	0.9688207
class	9	:	f1=	0.8773628	recall=	0.9515	precision=	0.8139435
total	:		f1=	0.9078222	recall=	0.936075	precision=	0.8822394
bs=10								
class	0	:	f1=	0.9619437	recall=	0.97	precision=	0.9540202
class	1	:	f1=	0.946459	recall=	0.97225	precision=	0.9220009
class	2	:	f1=	0.8481869	recall=	0.916625	precision=	0.7892584
class	3	:	f1=	0.8886442	recall=	0.90775	precision=	0.870326
class	4	:	f1=	0.8908196	recall=	0.916375	precision=	0.8666509
class	5	:	f1=	0.9574143	recall=	0.979375	precision=	0.9364169

class	6	:	f1=	0.8972986	recall=	0.92175	precision=	0.874111
class	7	:	f1=	0.9705643	recall=	0.970625	precision=	0.9705037
class	8	:	f1=	0.9709502	recall=	0.971375	precision=	0.9705258
class	9	:	f1=	0.915238	recall=	0.95425	precision=	0.8792905
total	:		f1=	0.9247519	recall=	0.9480375	precision=	0.9033104
bs=50								
class	0	:	f1=	0.9505024	recall=	0.95775	precision=	0.9433637
class	1	:	f1=	0.9274926	recall=	0.963375	precision=	0.8941873
class	2	:	f1=	0.7939931	recall=	0.902125	precision=	0.7090087
class	3	:	f1=	0.864018	recall=	0.86175	precision=	0.8662981
class	4	:	f1=	0.8613139	recall=	0.907125	precision=	0.8199074
class	5	:	f1=	0.9551945	recall=	0.97	precision=	0.9408341
class	6	:	f1=	0.8687448	recall=	0.90925	precision=	0.8316945
class	7	:	f1=	0.9644045	recall=	0.960125	precision=	0.9687224
class	8	:	f1=	0.9674761	recall=	0.968625	precision=	0.96633
class	9	:	f1=	0.909968	recall=	0.943125	precision=	0.8790633
total	:		f1=	0.9063108	recall=	0.934325	precision=	0.8819409
bs=100								
class	0	:	f1=	0.9300542	recall=	0.954875	precision=	0.906491
class	1	:	f1=	0.9049689	recall=	0.963	precision=	0.8535342
class	2	:	f1=	0.7620358	recall=	0.865625	precision=	0.6805897
class	3	:	f1=	0.8318584	recall=	0.863625	precision=	0.8023458
class	4	:	f1=	0.8529358	recall=	0.868875	precision=	0.8375708
class	5	:	f1=	0.9508683	recall=	0.961625	precision=	0.9403496
class	6	:	f1=	0.8450687	recall=	0.879875	precision=	0.8129114
class	7	:	f1=	0.9569664	recall=	0.959	precision=	0.9549415
class	8	:	f1=	0.9644645	recall=	0.9635	precision=	0.9654309
class	9	:	f1=	0.8904971	recall=	0.93825	precision=	0.8473696
total	:		f1=	0.8889718	recall=	0.921825	precision=	0.8601535
bs=500								
class	0	:	f1=	0.8813284	recall=	0.882375	precision=	0.8802843
class	1	:	f1=	0.7956189	recall=	0.9625	precision=	0.6780557
class	2	:	f1=	0.5457472	recall=	0.678125	precision=	0.4566114
class	3	:	f1=	0.6878893	recall=	0.7455	precision=	0.6385439
class	4	:	f1=	0.7242188	recall=	0.714125	precision=	0.734602
class	5	:	f1=	0.8406081	recall=	0.857	precision=	0.8248316
class	6	:	f1=	0.6864035	recall=	0.70425	precision=	0.6694392

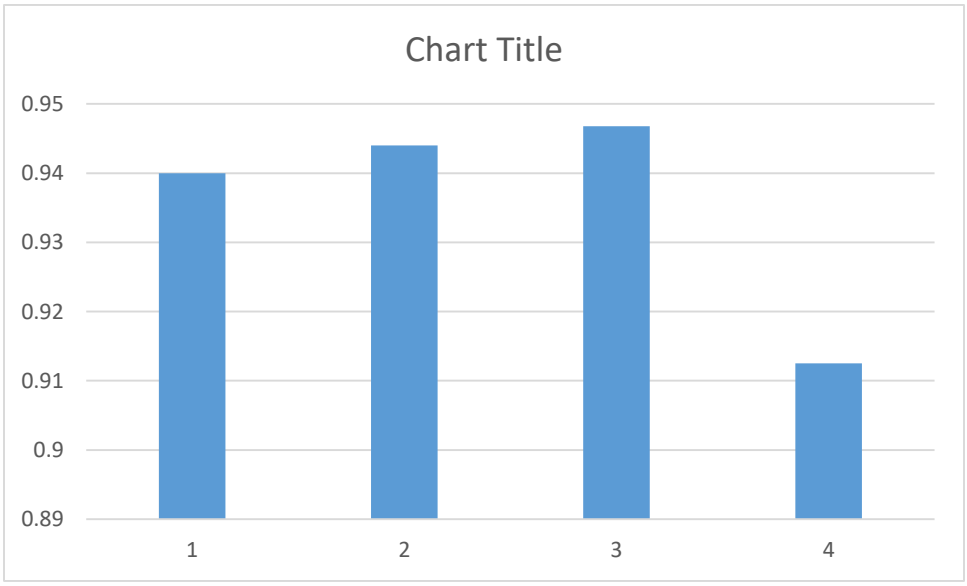
class	7	:	f1=	0.8853784	recall=	0.90375	precision=	0.8677388
class	8	:	f1=	0.8604555	recall=	0.909125	precision=	0.8167322
class	9	:	f1=	0.7628568	recall=	0.92525	precision=	0.6489567
total	:		f1=	0.7670505	recall=	0.8282	precision=	0.7215796
bs=1000								
class	0	:	f1=	0.847344	recall=	0.8215	precision=	0.8748669
class	1	:	f1=	0.6452675	recall=	0.98	precision=	0.4809816
class	2	:	f1=	0.4595868	recall=	0.634	precision=	0.3604321
class	3	:	f1=	0.6168026	recall=	0.6525	precision=	0.5848084
class	4	:	f1=	0.67112	recall=	0.565125	precision=	0.8260552
class	5	:	f1=	0.7625568	recall=	0.870125	precision=	0.6786585
class	6	:	f1=	0.6023305	recall=	0.481375	precision=	0.8044704
class	7	:	f1=	0.7945088	recall=	0.926	precision=	0.6957175
class	8	:	f1=	0.8070892	recall=	0.839625	precision=	0.7769809
class	9	:	f1=	0.7156897	recall=	0.896625	precision=	0.5955168
total	:		f1=	0.6922296	recall=	0.7666875	precision=	0.6678488
bs=10000								
class	0	:	f1=	0.6797497	recall=	0.848625	precision=	0.5669311
class	1	:	f1=	0.6789616	recall=	0.971	precision=	0.5219729
class	2	:	f1=	0.5017327	recall=	0.624375	precision=	0.4193603
class	3	:	f1=	0.6565854	recall=	0.607875	precision=	0.7137825
class	4	:	f1=	0.6041942	recall=	0.515	precision=	0.7307556
class	5	:	f1=	0.7385589	recall=	0.7585	precision=	0.7196395
class	6	:	f1=	0.6000858	recall=	0.437125	precision=	0.9567715
class	7	:	f1=	0.8356417	recall=	0.905	precision=	0.7761578
class	8	:	f1=	0.6168239	recall=	0.9725	precision=	0.4516429
class	9	:	f1=	0.7393661	recall=	0.844125	precision=	0.6577384
total	:		f1=	0.66517	recall=	0.7484125	precision=	0.6514752
bs=30000								
class	0	:	f1=	0.3666563	recall=	0.8885	precision=	0.2309892
class	1	:	f1=	0.7921968	recall=	0.934	precision=	0.6877761
class	2	:	f1=	0.5803438	recall=	0.559125	precision=	0.6032367
class	3	:	f1=	0.0047387	recall=	0.002375	precision=	1
class	4	:	f1=	0.2118661	recall=	0.129	precision=	0.5924225
class	5	:	f1=	0.7951531	recall=	0.77925	precision=	0.8117188
class	6	:	f1=	0.4819552	recall=	0.713625	precision=	0.3638391
class	7	:	f1=	0.8201183	recall=	0.86625	precision=	0.7786517

class	8	:	f1=	0.4810443	recall=	0.96275	precision=	0.3206228
class	9	:	f1=	0.4957865	recall=	0.353	precision=	0.8325472
total	:		f1=	0.5029859	recall=	0.6187875	precision=	0.6221804
bs=60000								
class	0	:	f1=	0.3493202	recall=	0.93775	precision=	0.2146372
class	1	:	f1=	0.6396023	recall=	0.972875	precision=	0.4764033
class	2	:	f1=	0.3995896	recall=	0.292125	precision=	0.6321342
class	3	:	f1=	0.0004999	recall=	0.00025	precision=	1
class	4	:	f1=	0.2863354	recall=	0.34575	precision=	0.2443463
class	5	:	f1=	0.7884279	recall=	0.815875	precision=	0.7627673
class	6	:	f1=	0.4548922	recall=	0.842375	precision=	0.3115724
class	7	:	f1=	0.7717151	recall=	0.74075	precision=	0.8053819
class	8	:	f1=	0.6197293	recall=	0.495125	precision=	0.8281413
class	9	:	f1=	0.59931	recall=	0.456	precision=	0.8739818
total	:		f1=	0.4909422	recall=	0.5898875	precision=	0.6149366

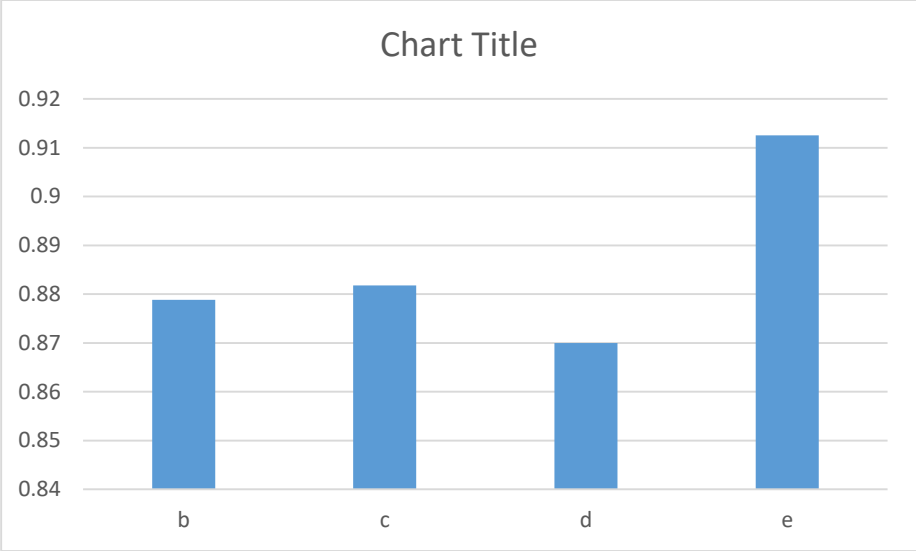
جمع بندی:

در کل بهترین خروجی بدست آمده ۹۵ درصد بود.

مقایسه کلی میزان موفقیت در جدول زیر موجود است ، در هر مورد بهترین عملکرد آورده شده است:



همچنین مشاهده نمودار در تعداد ایپاک مساوی ۶:



Kohonen .

در این سوال روابط مربوط به کوهونن پیاده سازی شده است.
برای انتخاب بهترین نورون فاصله اقلیدسی اندازه r,g,b را با ورودی میسنجیم.

برای فاصله نورون ها از مختصات استفاده مینماییم

انتخاب بهترین نورون بر اساس فاصله رنگی است.

R,k,tetta بر اساس روابط نوشته شده است و مقدار اولیه بر اساس تست نتیجه ها انتخاب شده است.

با انتخاب هر دیتا ، بهترین نورون را یافته فاصله را حساب مینماییم و سپس براساس میزان فاصله وزن های نورون ها را آپدیت مینماییم.

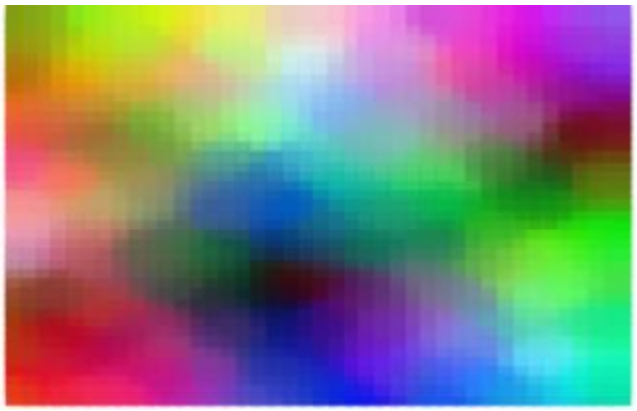
شکل های زیر نتایج خروجی هستند، عدد نوشته شده میزان epoch هر کدام از نتایج میباشد.

.0

.١

.٢٠

.۵۰



همانطور که مشاهده مینمایید دسته بندی رخ داده است.

در ادامه احتمال خطا اعمال شد که اندازه ۲ درصدی داشت اما پس از ۵۰ بار یادگیری نتیجه به صورت زیر شد.به عبارتی این میزان خطا برای این تعداد یادگیری , و این تعداد ورودی کافی نبوده است.

:Rbf

توابع پیاده سازی شده مطابق با روابط پی دی اف ها در اسلاید ها میباشند.

تابع اول همان رابطه صفحه ۱۱ میباشد.

تابع دوم صفحه ۱۲ میباشد.به این صورت که ماتریس ما بر اساس هر ورودی و هر مرکز منحنی دارای یک درایه باشد.

مراکز را ابتدا به صورت رندم انتخاب کرده ایم و شعاع اولیه داده ایم.

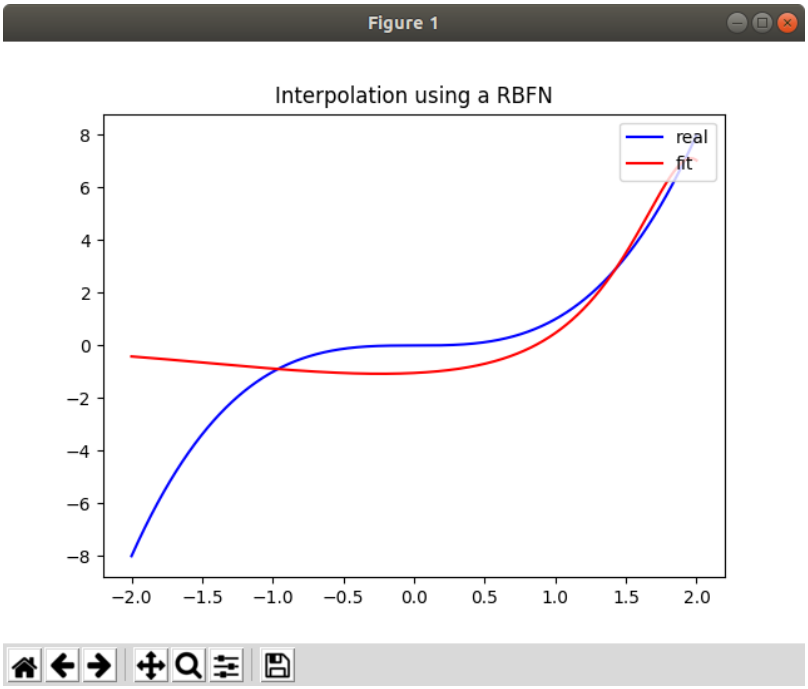
تابع فیت کننده دیتا تعریف شده که همان روابط صفحه ۱۳ میباشند.

تابع پیشبینی کننده در صفحه ۵ اسلاید ها موجود است.

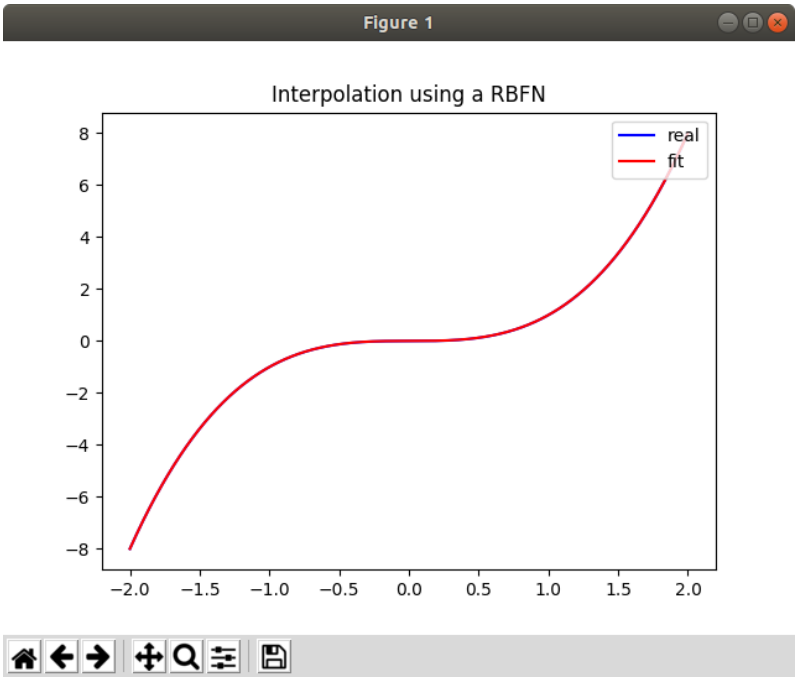
نقاط ورودی و خروجی هر دو در بازه [2,2-] قرار دارند و تابع x^3 استفاده گردیده است.

نمونه های خروجی زیر بر اساس تغییرات شعاع و تعداد مراکز منحنی (نورون ها) مشخص گردیده است.

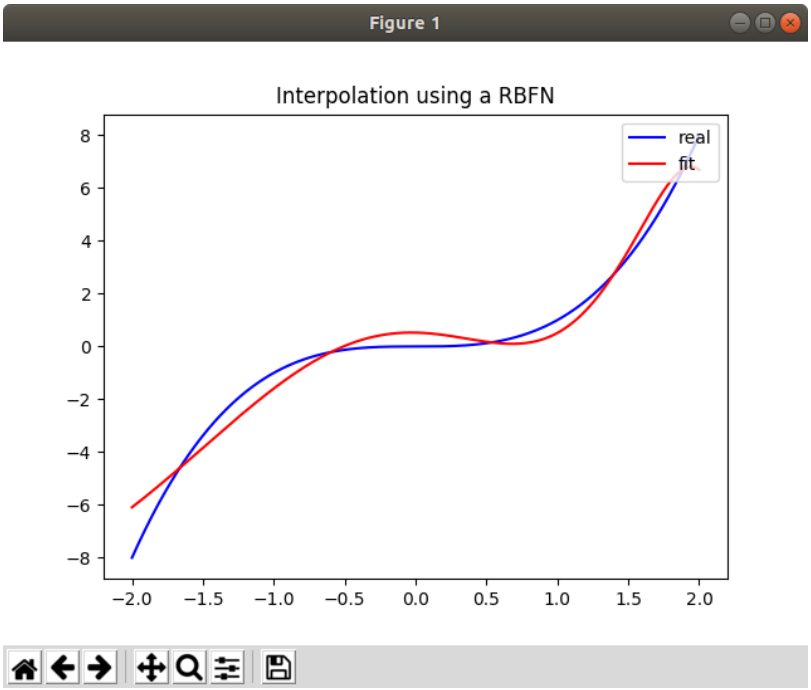
شعاع ۲ تعداد ۱۰۰:



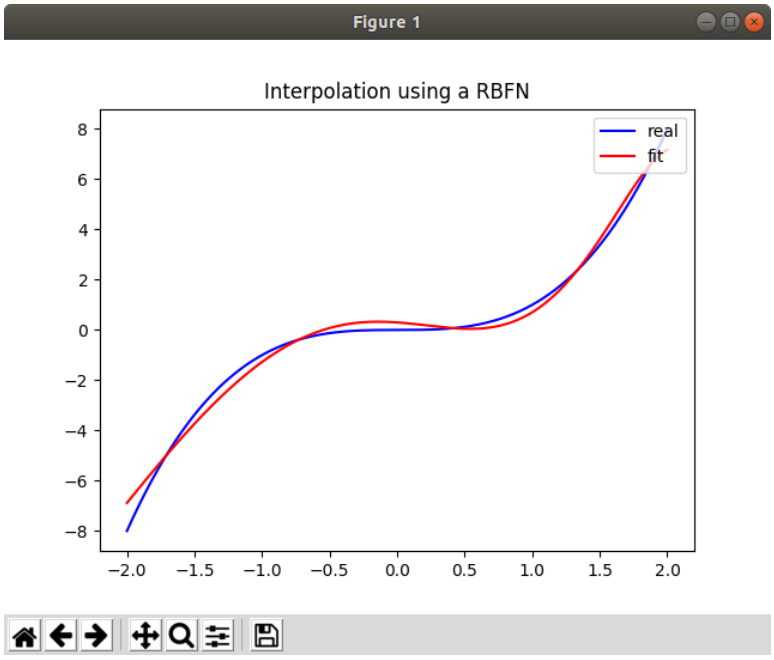
شعاع ۲۰ تداد ۱۰۰:



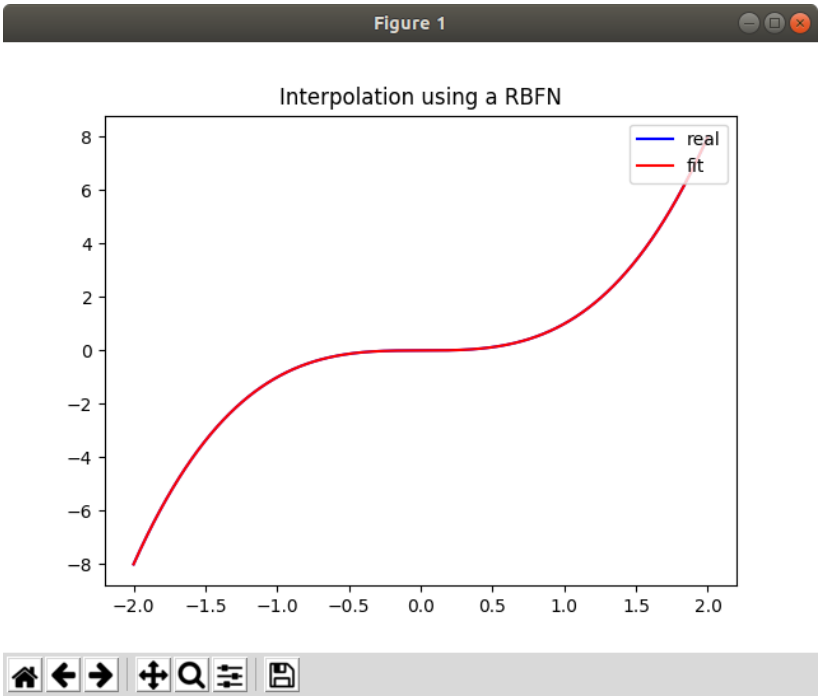
شعاع ۱۰ تعداد ۵۰:



شعاع ۱۰ تعداد ۱۰۰:



و در نهایت شعاع ۱۰ تعداد ۴۰۰:



قابل مشاهده است که در صورت انتخاب تعداد مناسب و شعاع مناسب میتوان به تخمین بسیار دقیقی دست یافت.