باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

**درس نوروساینس**

گزارش تمرین سری پنجم

علی محرابیان

تینا اسماعیل زاده

استاد

دکتر کربلایی آقاجان

## سوال اول:

جدول صحت AND و XOR به صورت زیر است.

XOR

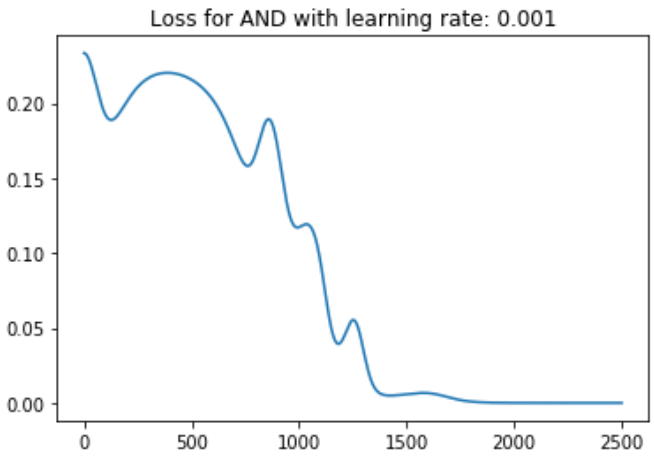
| X1 | X2 | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

AND

| X1 | X2 | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

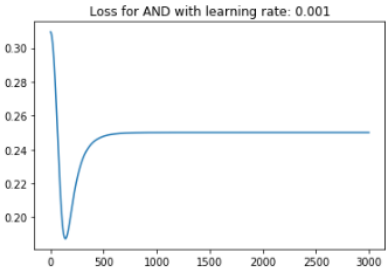
در قسمت اول کلاس Network را تعریف کردیم که ورودی آن تعداد ورودی،تعداد نورون های لایه میانی و تعداد خروجی است.trian نیز برای آموزش استفاده می شود که ورودی،خروجی،epoch، Learning rate و شبکه تعریف شده را می گیرد.در ادامه نیز network2 وtrain2 برای شبکه با تعداد لایه های میانی متغیر استفاده می شود.از تابع فعالسازی sigmoid استفاده کردیم.

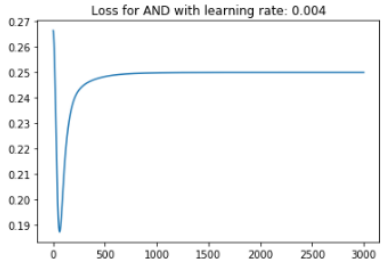
شبکه را برای AND آموزش داده و خطا را رسم می کنیم.

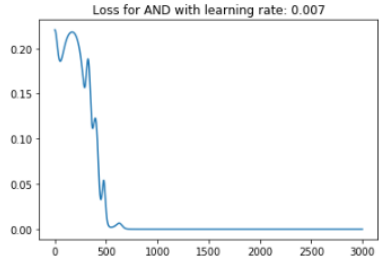


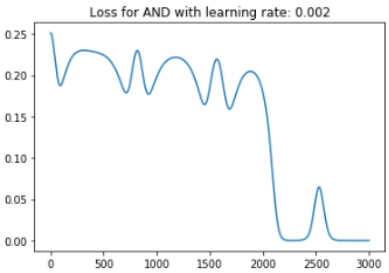

در این مورد تعداد epoch ها برابر 2500 انتخاب شده است.

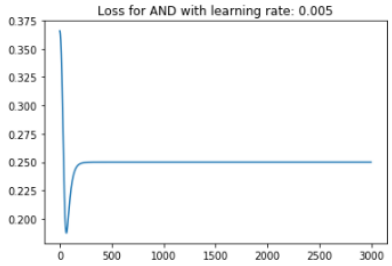
ابتدا منحنی خطا نسبت تغییرات learning rate را رسم می کنیم.همان طور که مشاهده می شود،هرچه Learning rate کمتر باشد،همگرایی دیرتر اتفاق می افتد.

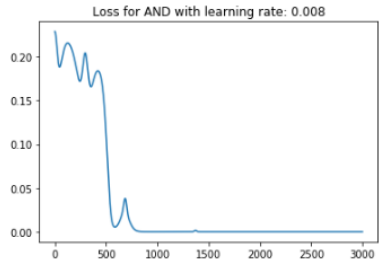


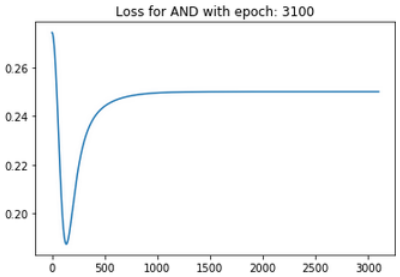



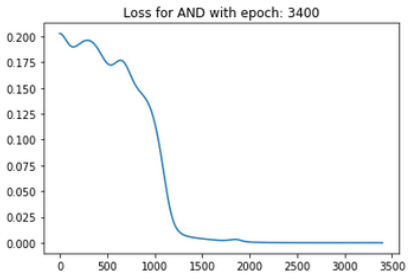



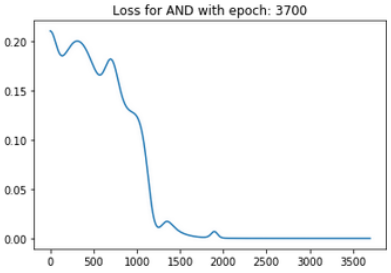



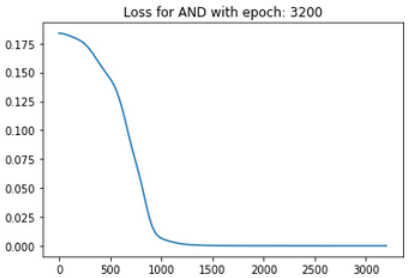



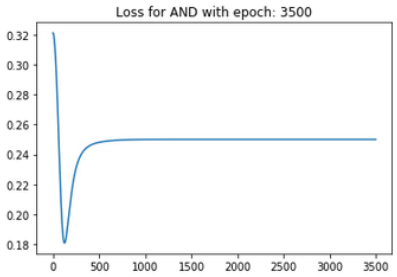

حال تغییرات epoch را رسم می کنیم.مشاهده می شود که نسبت به متغیرقبل،تغییرات شدید تر است.

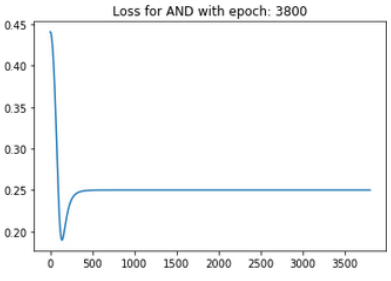


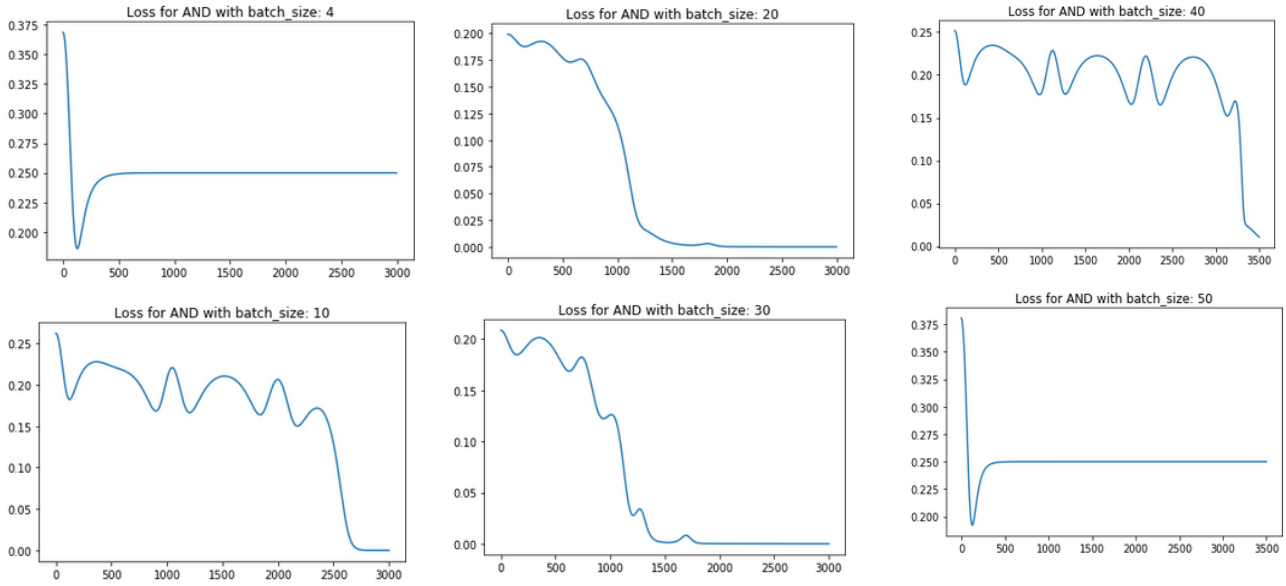



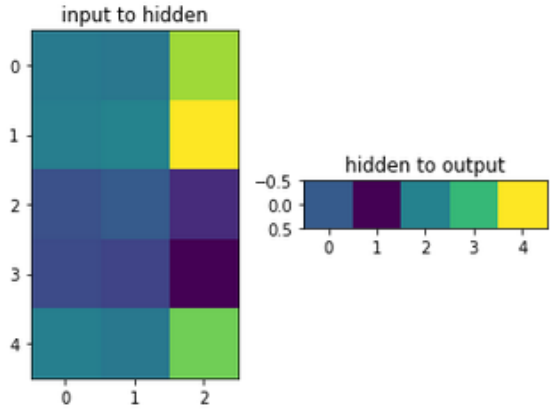



حال اثرات batch_size بررسی می کنیم.مشاهده می شود که با افزایش آن،یادگیری کندتر می شود چون که دسته بزرگ تری از داده ها بزرگتری باید train شود.



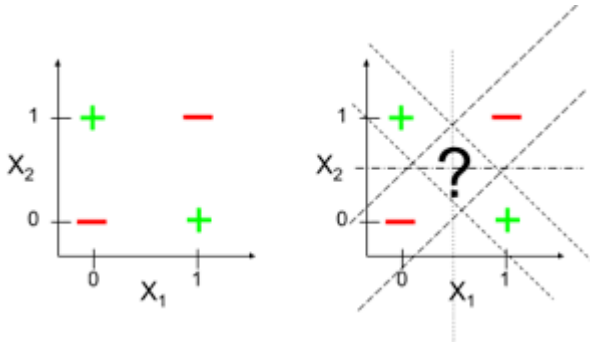

دراین قسمت فعالیت نورون ها را تصویر می کنیم.چون دارای 1 لایه میانی هستیم،دو تصویر نورون های ورودی به میانی و نورون های میانی به خروجی را تصویر می کنیم.




همان طور که مشاهده می شود،به ازای ورودی مشخص،خروجی ما وابسته به مقدار اولیه وزن ها می باشد.

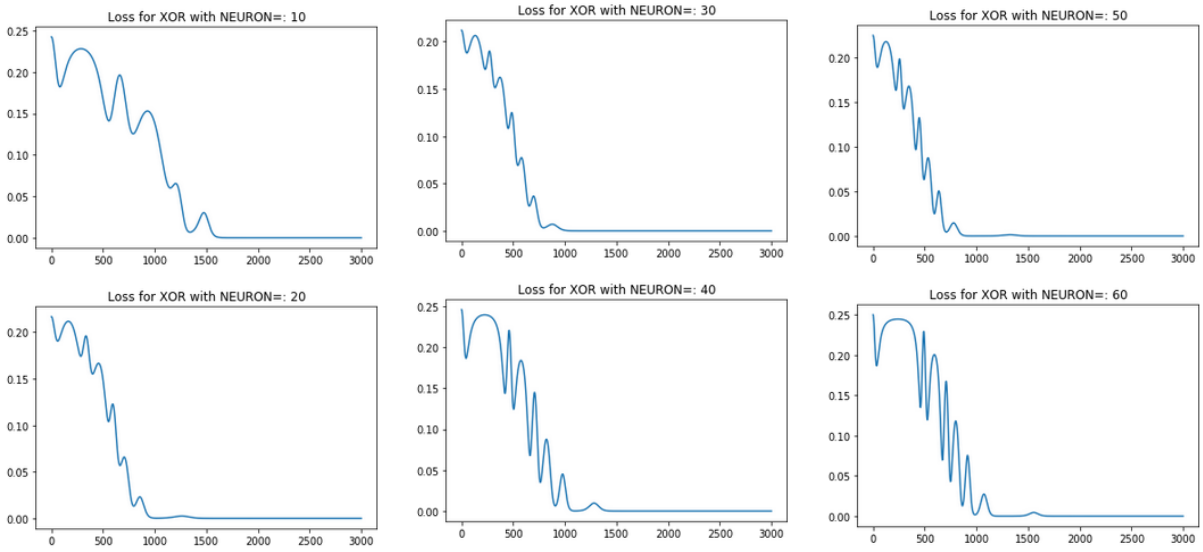
برای پیاده  سازی XOR،نیاز به نورون و لایه های بیشتری داریم چرا که هرچه کمتر باشند،همگرایی در این مورد کندتر می باشد.همان طورکه مشاهده می شود،دسته بندی کردن خروجی ها نسبت به حالت AND با یک خط امکان پذیر نیست.در نتیجه نیاز به لایه پنهان برای پیاده سازی داریم.

XOR



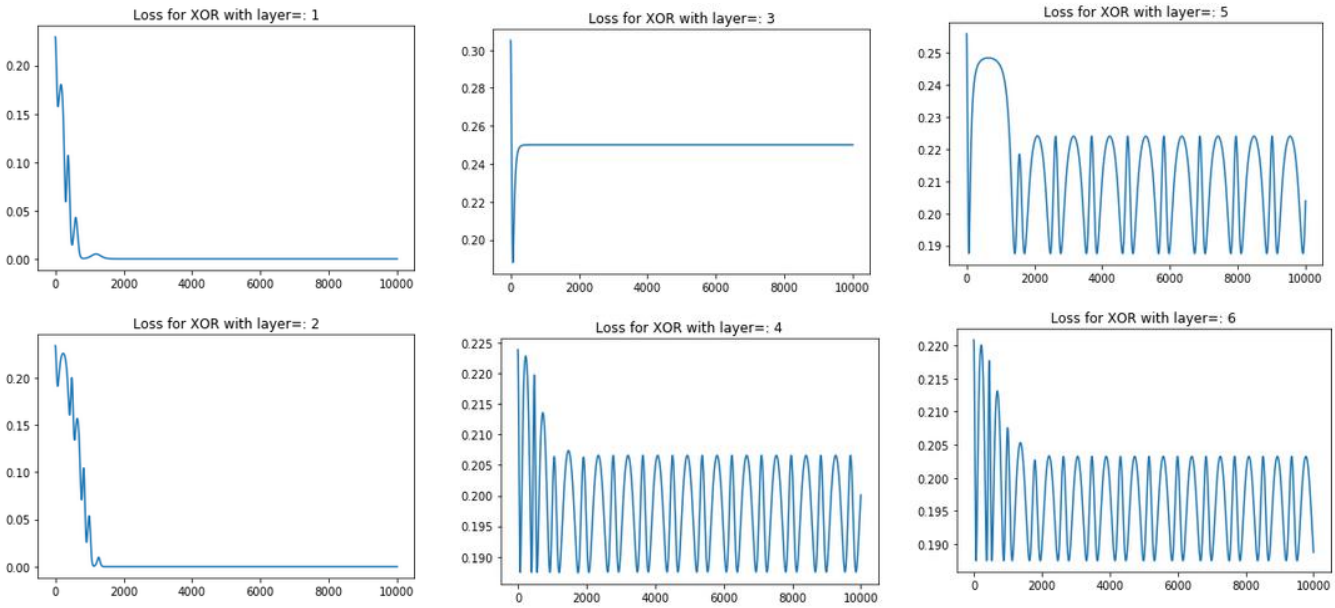

در این حالت تعداد ورودی هارا یک مورد افزایش می دهیم که مانند اضافه کردن یک ثابت و bias به لایه می باشد که از تغییرات ناگهانی وزن ها جلوگیری می کند.

در حالت اول،تغییرات با یک لایه و نورون های متفاوت را بررسی می کنیم.




مشاهده می شود که یادگیری با افزایش تعداد نورون ها سخت تر می شود که قابل انتظار است.

حال افزایش لایه های میانی را بررسی می کنیم.

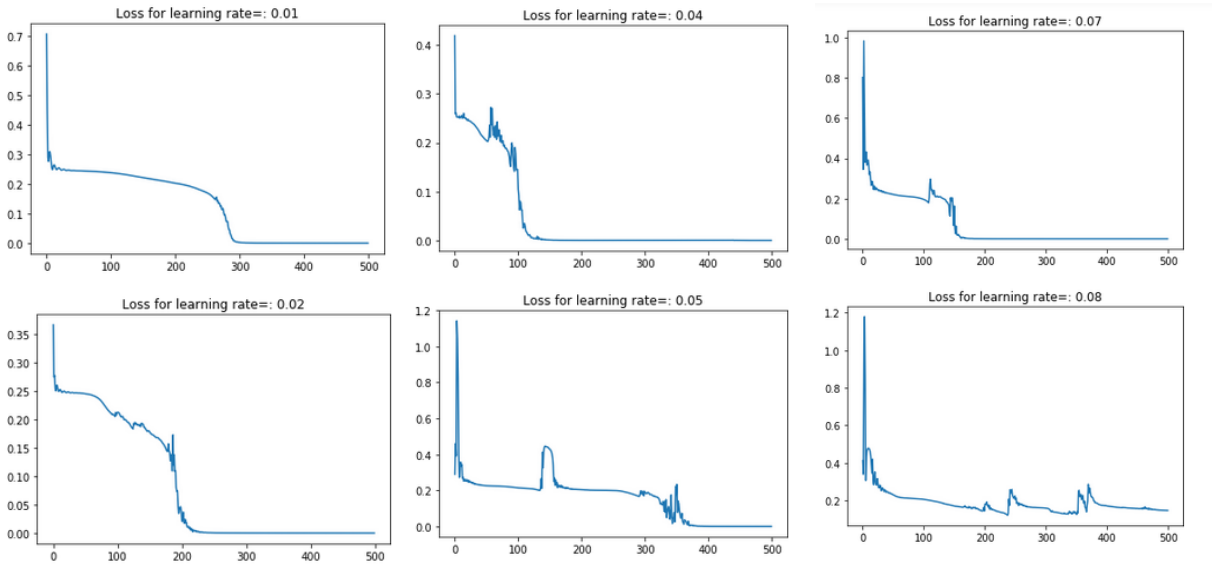



به وضوح مشاهده می شود که افزایش لایه ها،یادگیری را با اشکال مواجه می کند.که دلیل آن می تواند این باشد که که دریک min محلی افتاده باشیم.پس افزایش بیش از حد لایه ها کار خوبی نیست.به طور کلی تا حدامکان،شبکه با لایه و نورون های کمتری داشته باشیم تا خروجی بهتری بگیریم.

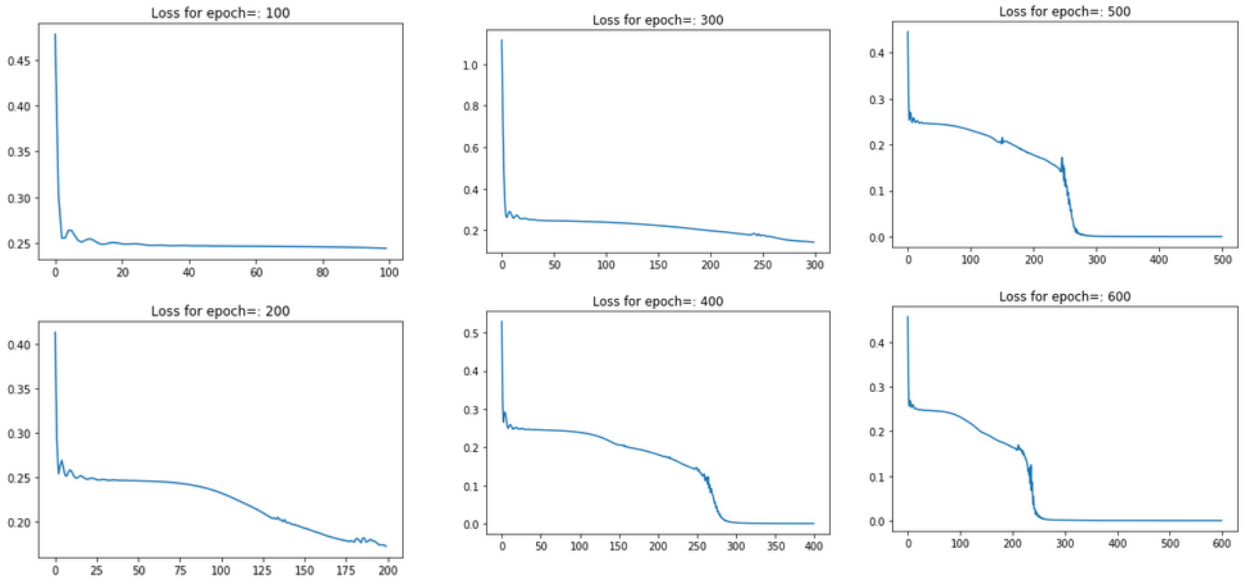
## سوال دوم:

شبکه بازگشتی با یک لایه طراحی کردیم.رشته مورد نظر را ساختیم.در این جا طول رشته بسیارمهم است به طوری که رشته با طول کوتاه می تواند باعث overfit شود و رشته با طول بلند به دلیل کم بودن لایه میانی و حجم کم حافظه،باعث خروجی نادرست می شود.در پیاده سازی شبکه بازگشتی RNNp،از تابع فعالسازی tanh استفاده شد.

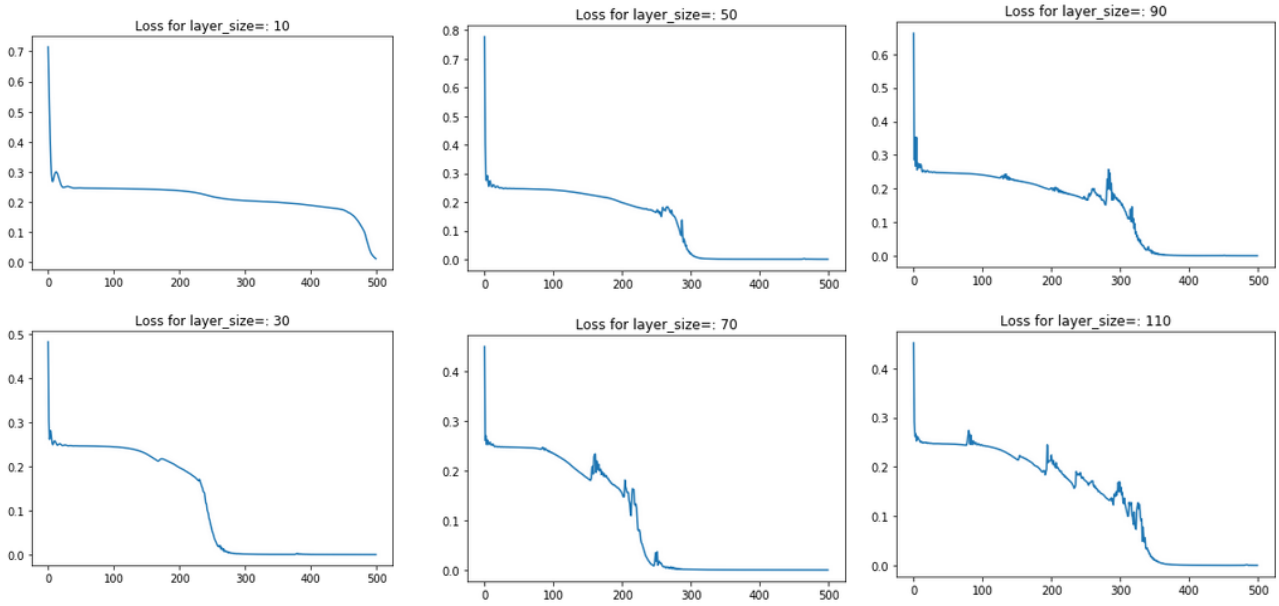
حال تغییرات خطا نسبت به متغیرهای مختلف را بررسی می کنیم.




مشاهده می شود که با افزایش learning rate،آموزش زودتر حاصل می شود.هرچند که در این سوال خاص،هر بار تغییر می تواند حالت خاص خود را داشته باشد.

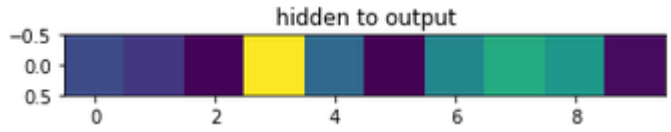



همان طور که می توان دید،با افزایش تعداد epoch ها،شبکه بهتر آموزش دیده و خطا زودتر به صفر می رسد.




در ابتدا مشاهده می شود که افزایش اندازه لایه میانی تاثیر بسیار مثبتی داشته است و خطا خیلی سریع به صفر میل می کند ولی اندازه که از 70 زیادتر می شود،خروجی مطلوبی نداریم که علت آن است که تعداد epoch های ما برای آموزش کم است که در صورت افزایش آن،زمان اجرای برنامه بسیار کند می شود.

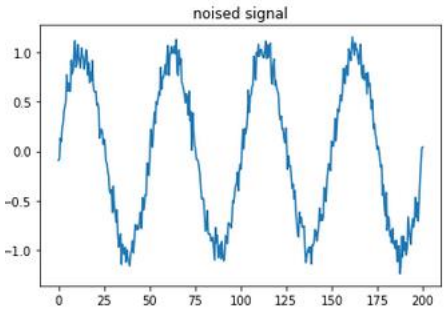
و حال تصویر فعالیت نورون ها



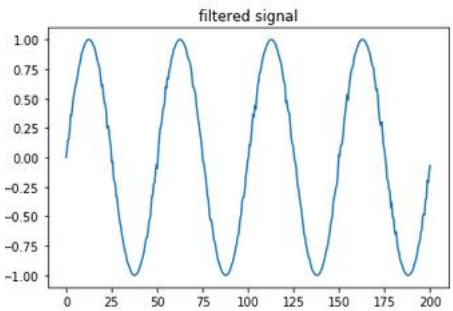

نورون های لایه میانی به خروجی تصویر شده اند که نشان می دهد که پس از مقدار معینی epoch، به حالت استقرار می رسند.

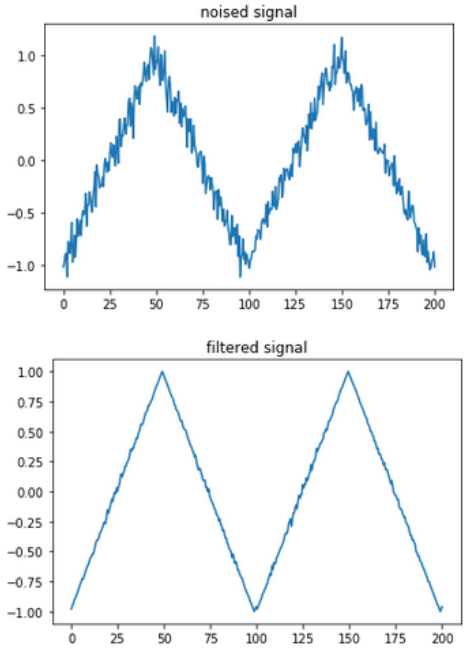
## سوال سوم:

ابتدا سیگنال های خواسته شده را می سازیم.سپس شبکه بازگشتی خود را تعریف کرده که در آن از تابع فعال سازی tanh استفاده شده است.سپس نویز normal و انتگرال آن،نویز Brownian را تعریف می کنیم.برای آموزش بهتر شبکه،مقادیر دو نویز را در کنار هم قرار داده و با هم به شبکه می دهیم. مقدار learning rate برابر با 0.1 و تعداد epochهارا 5000 در نظر گرفتیم و هر بار 400 نقطه (200 نقطه برای نویز نرمال و200 نقطه برای نویز براونی) در نظر گرفتیم.

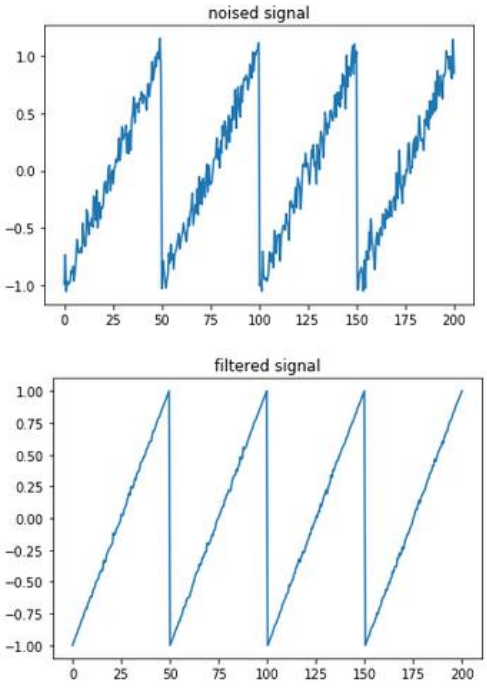



پس از آموزش شبکه،200 نقطه تست سیگنال با نویز نرمال به شبکه دادیم که خروجی آن برای موج سینوسی به صورت زیر است.



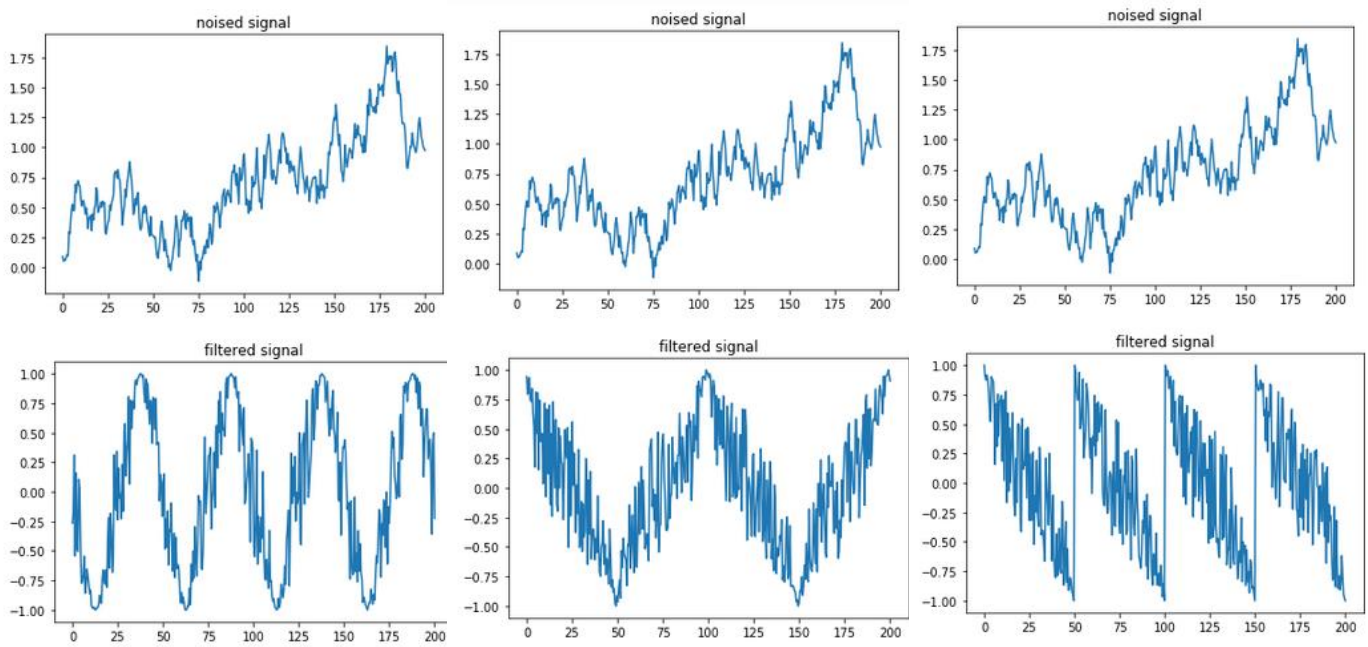



پس از آموزش شبکه،200 نقطه تست سیگنال با نویز نرمال به شبکه دادیم که خروجی آن برای موج مثلثی به صورت زیر است.

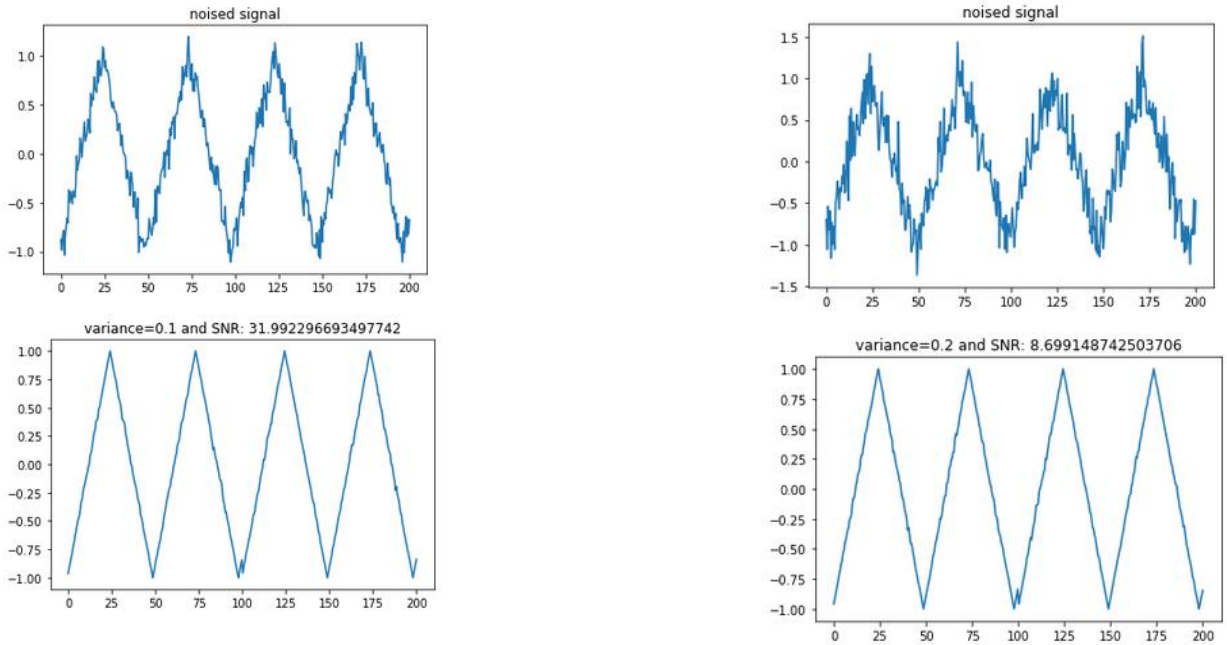


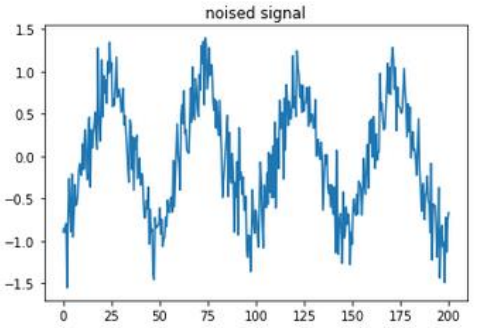

پس از آموزش شبکه،200 نقطه تست سیگنال با نویز نرمال به شبکه دادیم که خروجی آن برای موج اره ای به صورت زیر است.

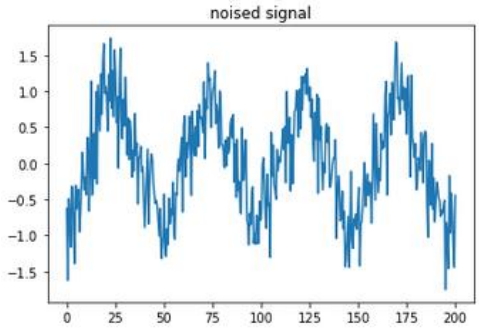
حال در قسمت فقط سیگنال Brownian را به هر شبکه می دهیم.نتایج جالب است.در خروجی می بینیم که سیگنال با نویز نرمال ظاهر می شود که در واقع می توان این طور تعبیر کرد که که ورودی یک مرحله عقب تر را داریم.

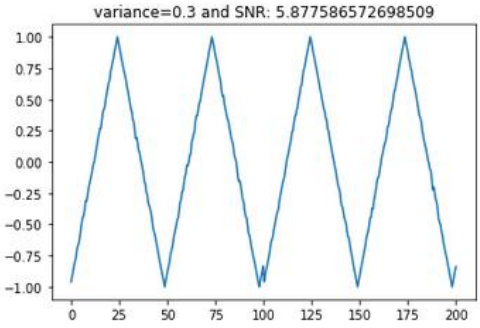


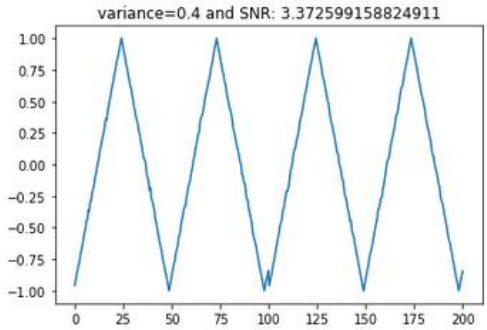

حال تاثیرات آستانه نویز را با تغییر واریانس بررسی می کنیم.ابتدا SNR که نسبت توان سیگنال به نویز است را تعریف می کنیم.در تمامی از سیگنال مثلثی برای برای نمایش نتایج استفاده کردیم.





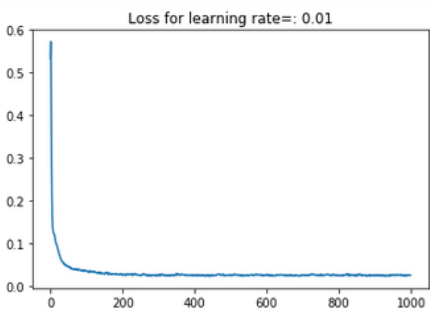



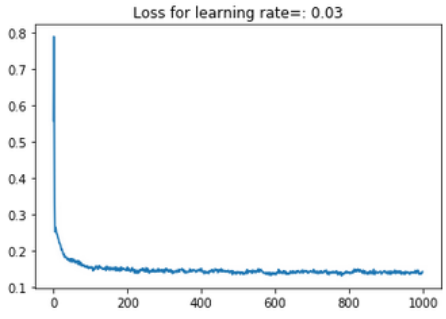



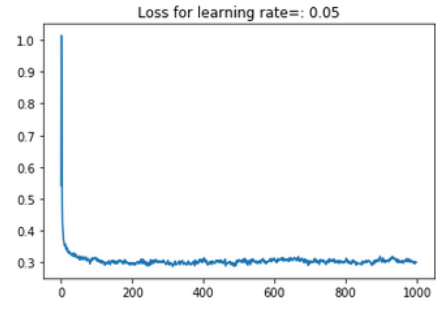



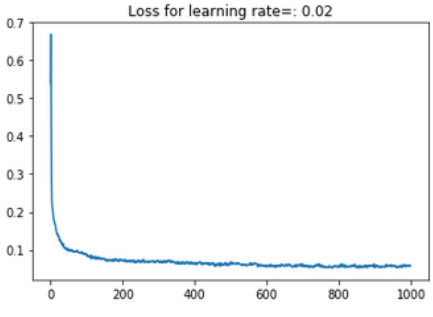

با افزایش واریانس خطا بیشتر شده ولی همان طور که مشاهده می شود،حساسیت به آستانه نویز خیلی زیاد نیست و با اینکه طی هر مرحله خطا بیشتر شده است،ولی خروجی مطلوبی با همان شبکه قبلی حاصل می شود.

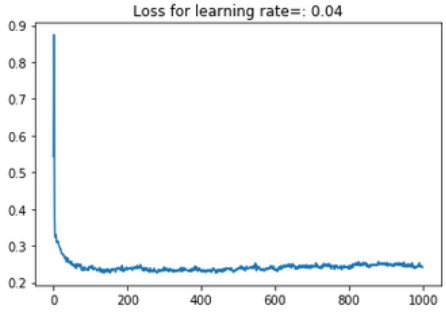
حال تغییرات خطا با تغییر پارامترهای مختلف را بررسی می کنیم.

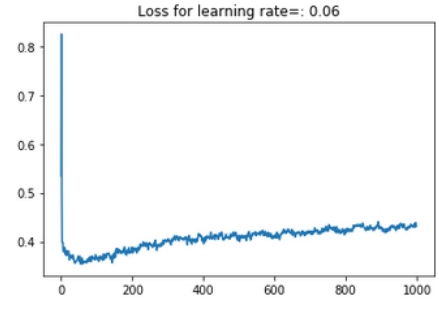


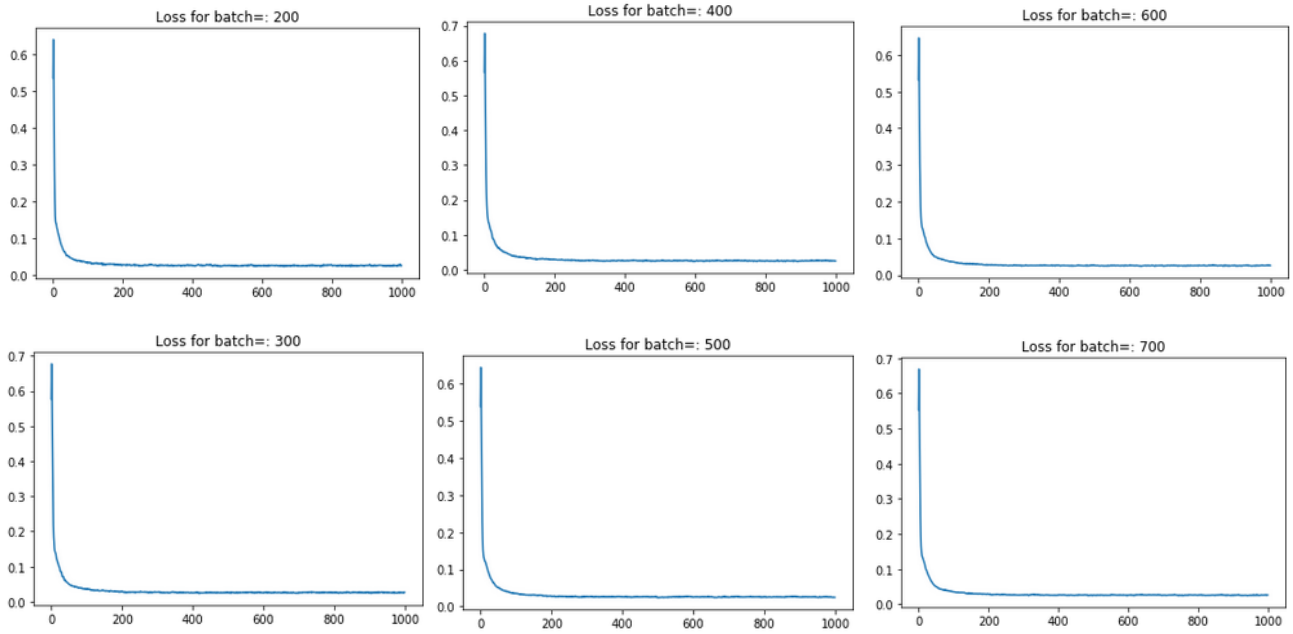



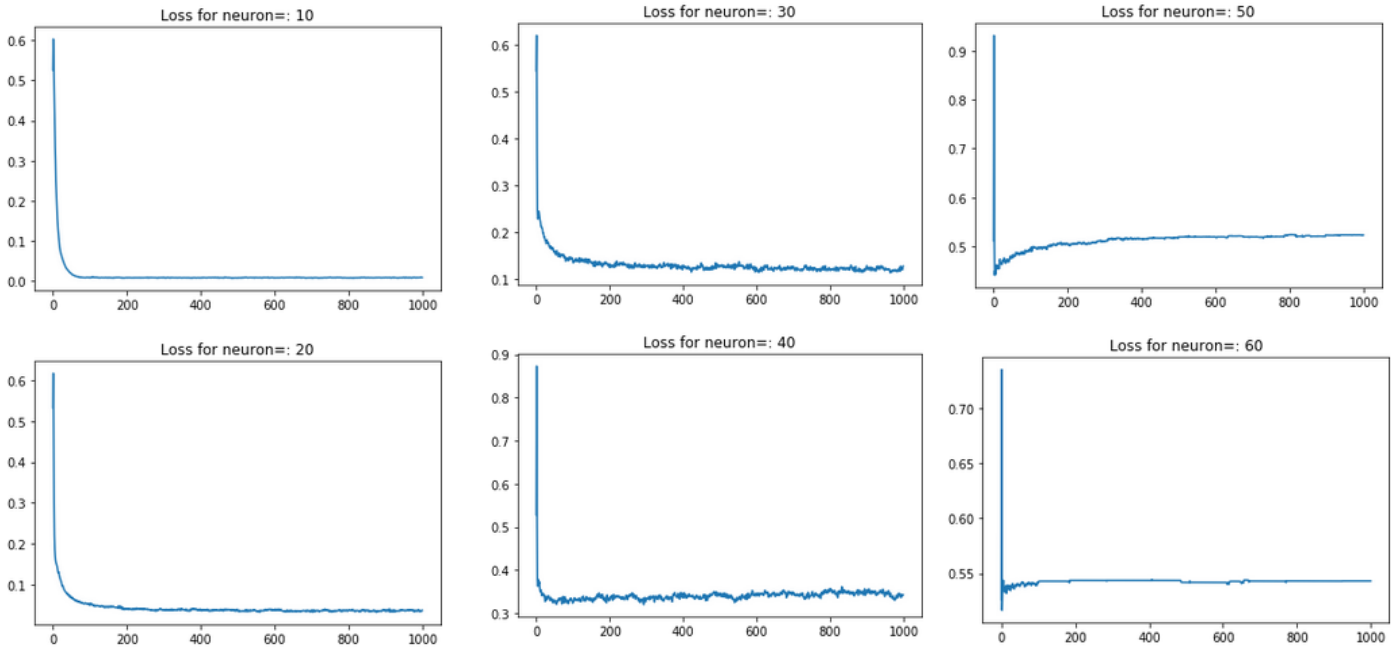



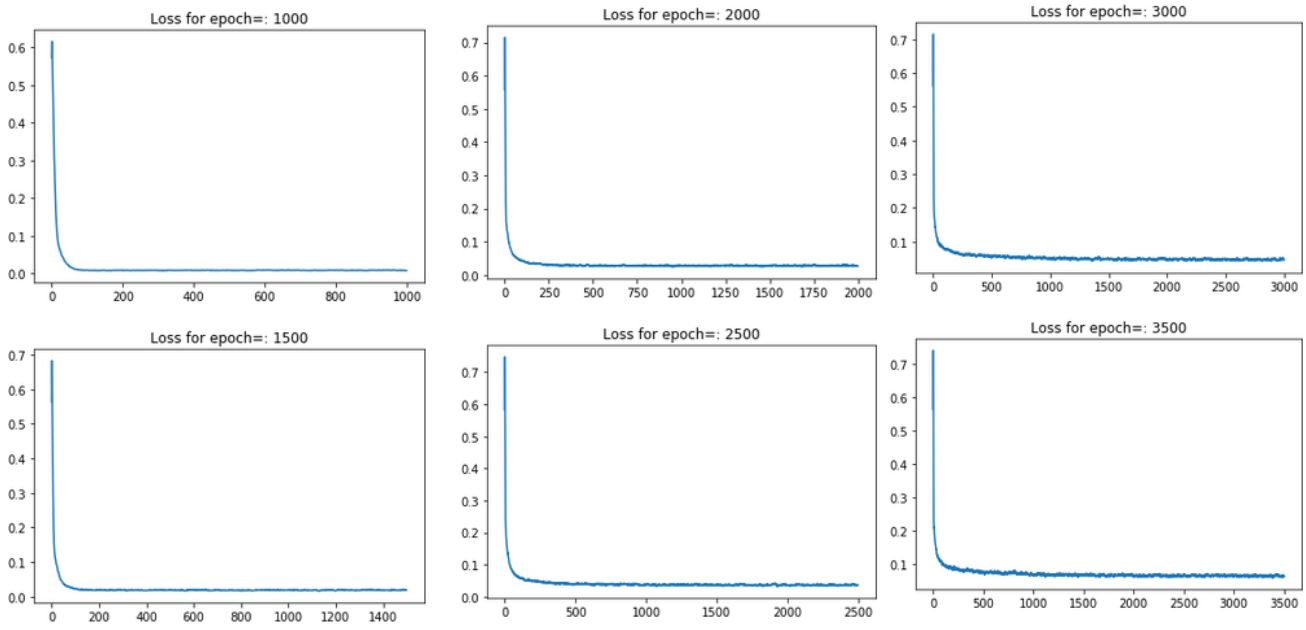
در ابتدا با learning rate های کم،خطا به صفر میل می کند ولی با افزایش آن،شاهد واگرایی هستیم.



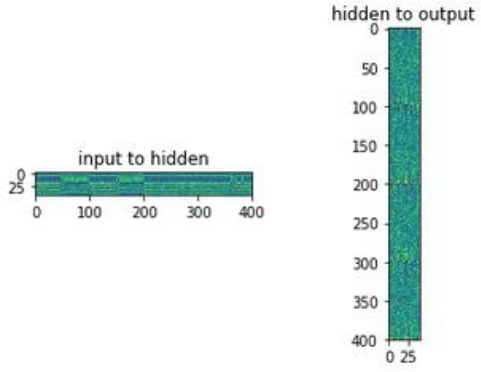

همان طور که مشاهده می شود،این مورد خیلی تاثیر گذار در خطا نمی باشد.



دراین قسمت مشاهده می شود که با نورون های لایه میانی کمترهم می توان خروجی مطلوب را گرفت و با زیاد شدن آنها می توان به وضوح دید که شبکه همگرا نمی شود.البته با افزایش تعداد epoch ها می توان آن را حل کرد ولی نیاز به این کار نبوده و با نورون های کمتر می توان به خروجی مطلوب رسید.



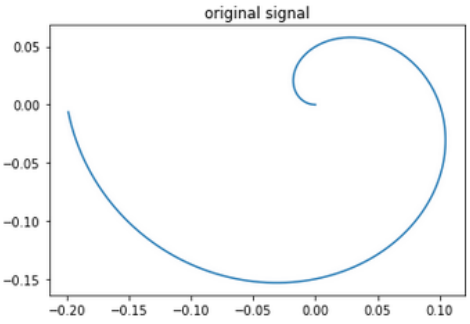

افزایش epoch،تاثیر مثبت دارد ولی دراین مورد،خیلی محسوس نیست.

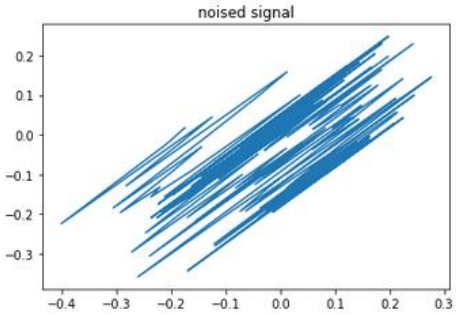


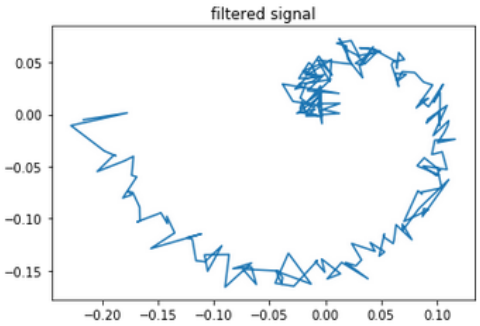

فعالیت نورون های ورودی به لایه میانی و لایه میانی به خروجی را تصویر کردیم.مشاهده می شود که فعالیت متوسطی دارند.

## سوال چهارم:

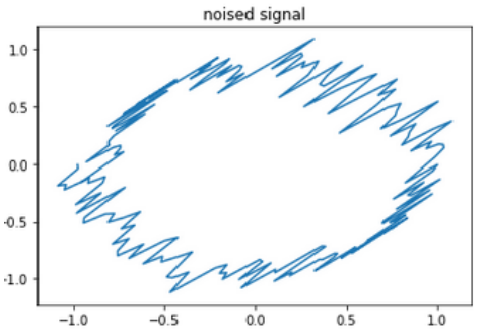
تقریبا تمام موارد این سوال مشابه سوال قبل است.خم مورد نظر را تعریف می کنیم.ابتدا برای 200 نقطه ابتدایی:

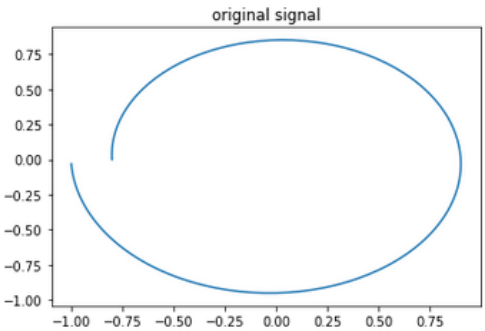


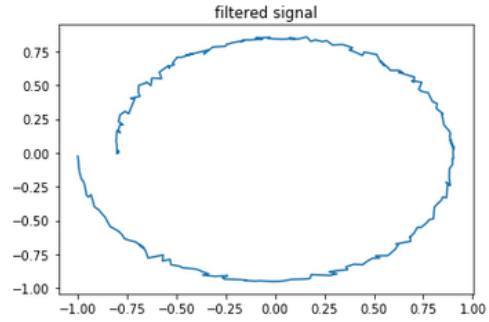



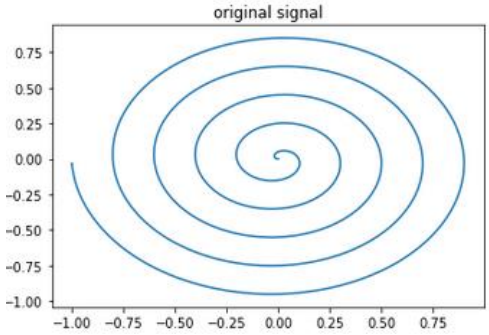

وبرای 200 نقطه انتهایی:

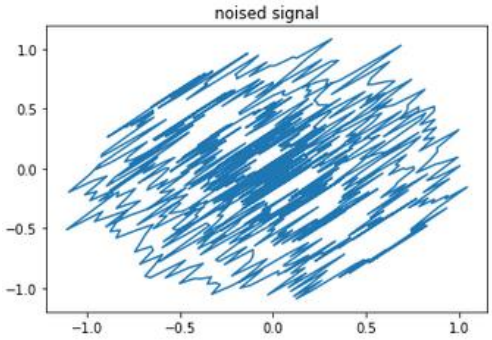


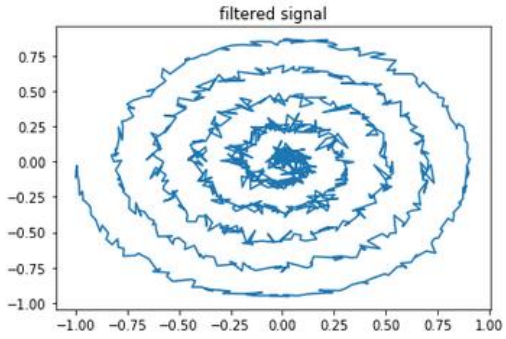



چون شبکه تا رسیدن به 200 نقطه انتهایی آموزش بهتری می بیند،در نتیجه خروجی بهتری دراین حالت حاصل می شود.

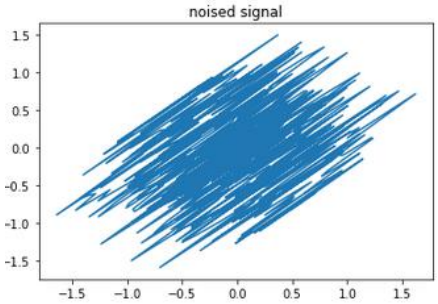


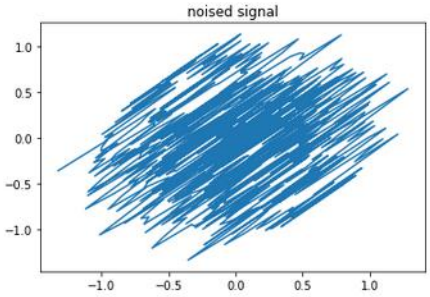



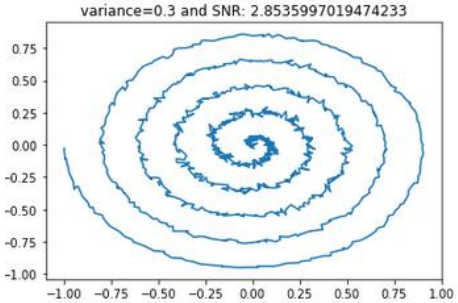



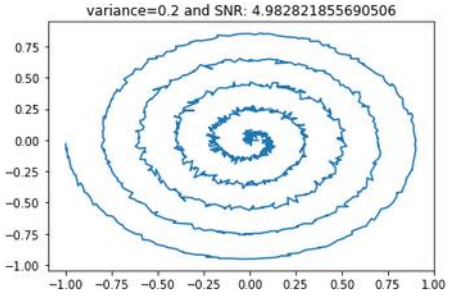

برای یک نمونه 1000 تایی ورودی و خروجی به صورت بالاست.learning rate را برابر با 0.01 وتعداد epoch ها را برابر با 7000 انتخاب کردیم.

مشابه قسمت قبل SNR را تعریف کردیم و مشاهدات زیر ثبت شد.



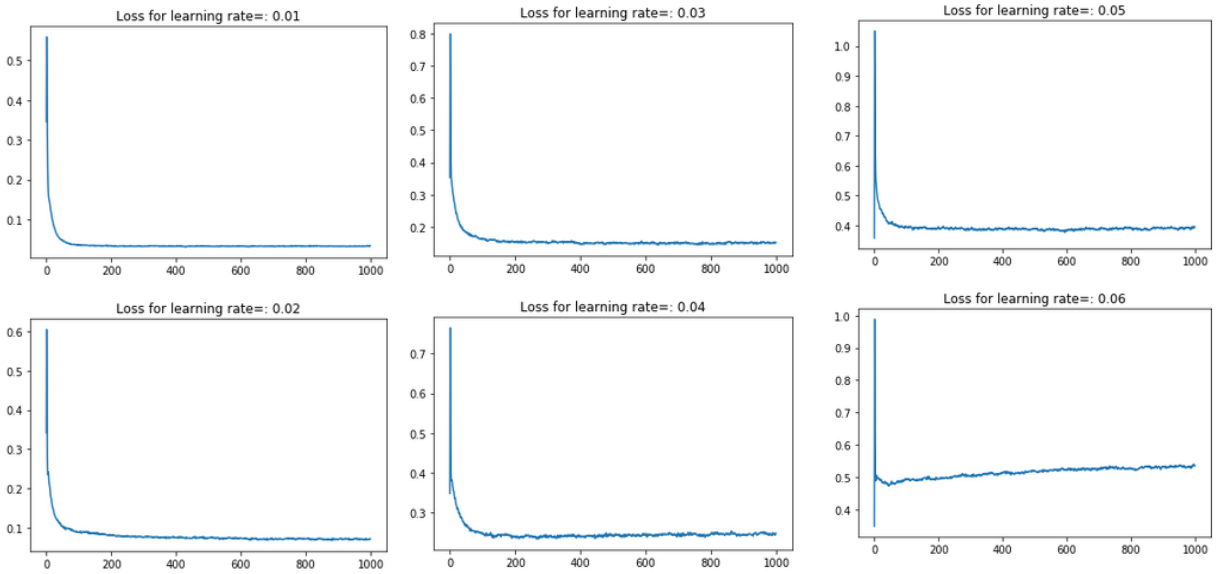



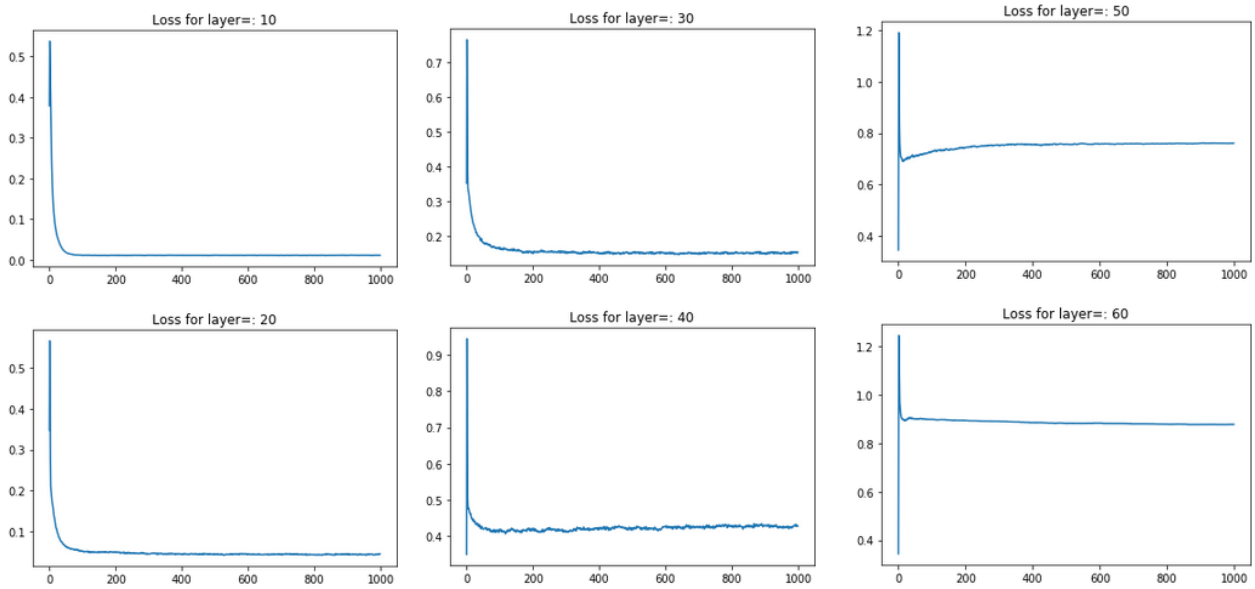



مشاهده می شود که دراین حالت نیز به آستانه نویز خیلی حساس نیستیم و خروجی قابل قبولی داریم.

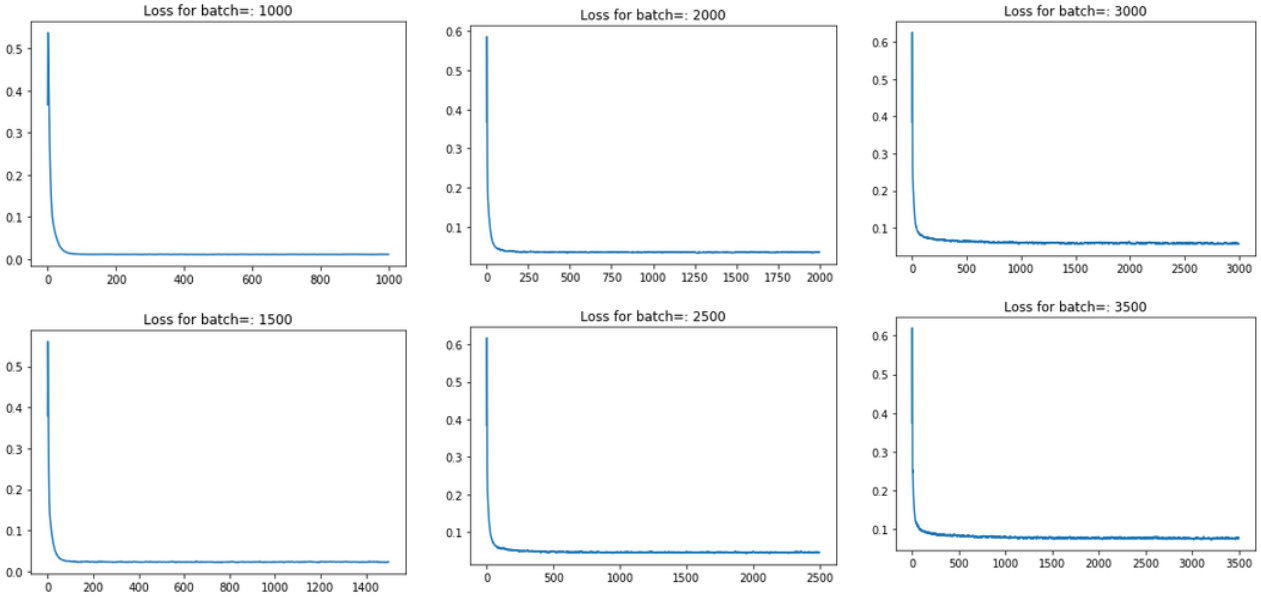
حال تغییرات خطا با تغییرات متغیرهای مختلف را بررسی می کنیم.



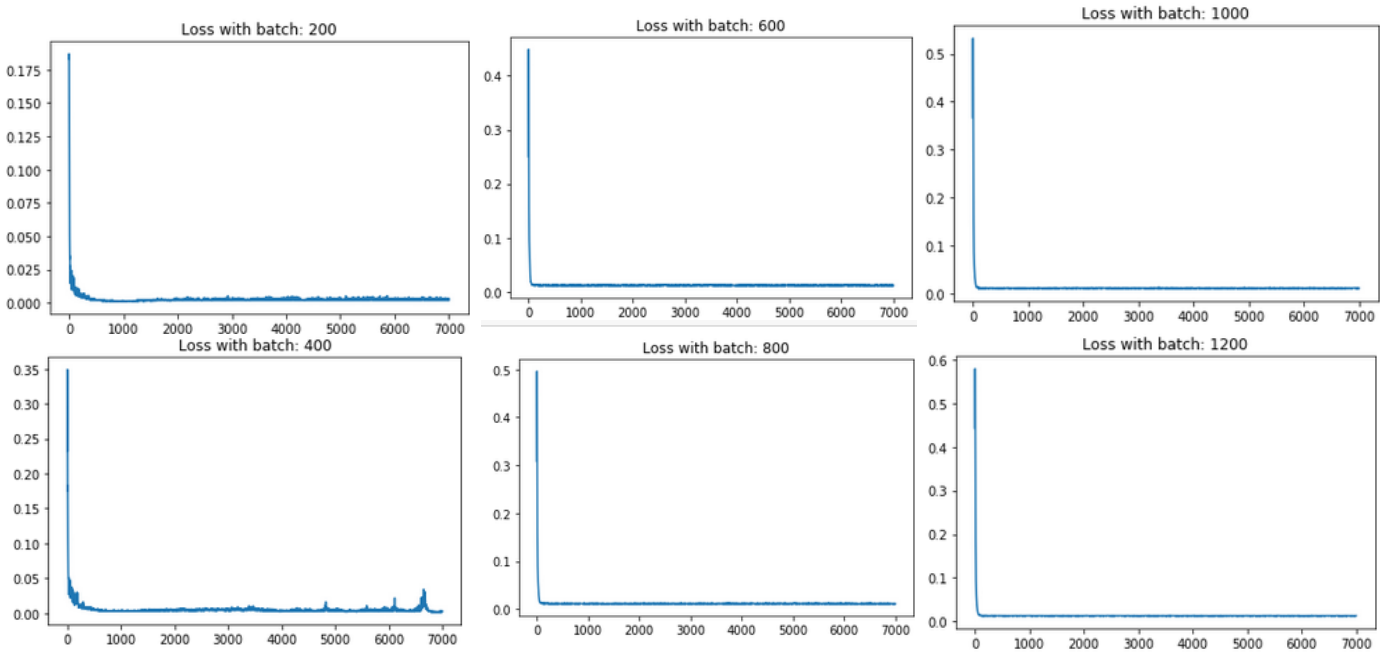

مشاهده می شود که برای learning rate های کمتر،شبکه بهتر آموزش می بیند ولی هرچه این مقدار بیشتر می شود،دچار خطای بیشتری می شویم.



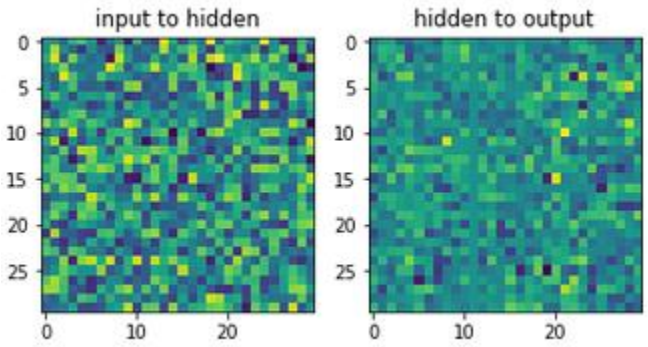
مشابه قسمت قبل،می بینیم که با نورون های لایه میانی کمترهم می توان خروجی مطلوب را گرفت و با زیاد شدن آنها می توان به وضوح دید که شبکه همگرا نمی شود.البته با افزایش تعداد epoch ها می توان آن را حل کرد ولی نیاز به این کار نبوده و با نورون های کمتر می توان به خروجی مطلوب رسید.




درحالت کلی،epoch ها بیشترین تاثیر را می گذارند ولی در دو سوال اخیر،با تغییر دادن این کمیت، چندان تغییر محسوسی مشاهده نمی شود.



مشاهده می شود که در همان اول،همگرایی خوبی داریم و تغییرات batch-size،خیلی محسوس نیست.




فعالیت نورون های ورودی به لایه میانی و لایه میانی به خروجی را تصویر کردیم.مشاهده می شود که نورون های آبی فعالیت متوسطی دارند و نورون های زرد رنگ که بیشتر در ابتدای کار هستند،فعالیت بیشتری دارند.