شبکه‌های عصبی

NNH

نگارش

حامد زارعی

نیمسال دوم ۹۷-۹۶

۱. تعریف مساله

هدف از این بخش بررسی کارایی شبکه های عصبی در شناسایی توابع و سیستم های غیر خطی می باشد. که در این راستا چند شبکه بررسی خواهد شد:

- شبکه عصبی پرسپترون با دو لایه میانی
- شبکه عصبی پایه شعاعی (RBF)
- شبکه بازگشتی جردن

۲. حل مساله

به صورت کلی ۳ نوع از شبکه های عصبی بررسی خواهد شد:

- پرسپترون با یادگیری Gradient Descent
- RBF
- Recurrent

در ادامه به بررسی هرکدام خواهیم پرداخت.

۲,۱. پرسپترون با یادگیری Gradient Descent

به صـورت کلی Gradient Descent که پایه برای انواع دیگر مورد بررسـی نیز هسـت به دو بخش کلی تقسیم می شود:

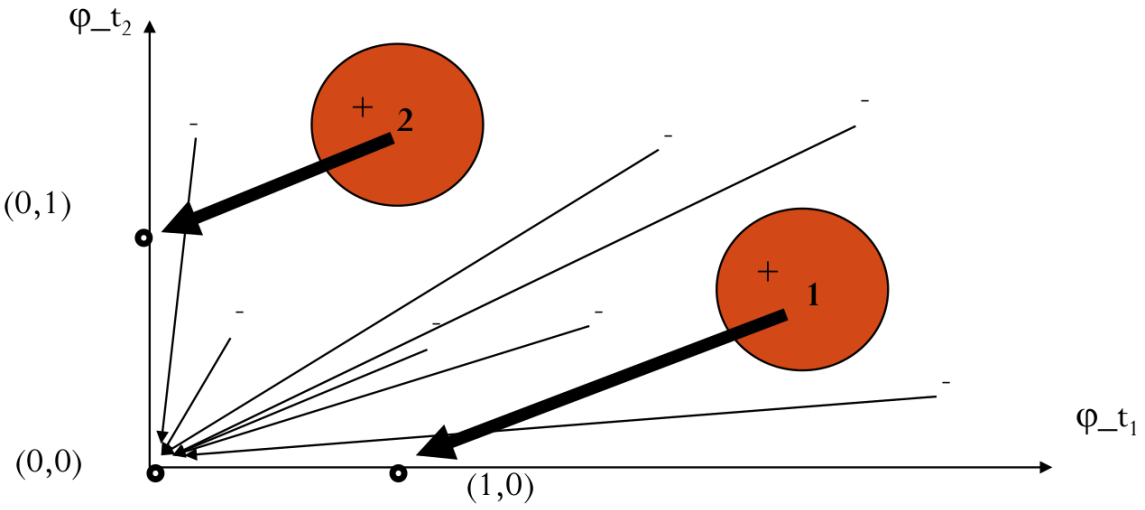
- Feed Forward
- Back Propagation

در بخش Feed Forward یکبـار بـا اعمـال Activation Function ها بر روی ورودی ها در هر لایه، خروجی هر قسمت بدست می آید.

در بخش بعدی که Back Propagation اسـت با اسـتفاده از مشتق توابع و اعداد بدست آمده از بخش FF، محاسبات مربوطه برای به روز رسانی وزن ها انجام می شود.

۲,۲. RBF

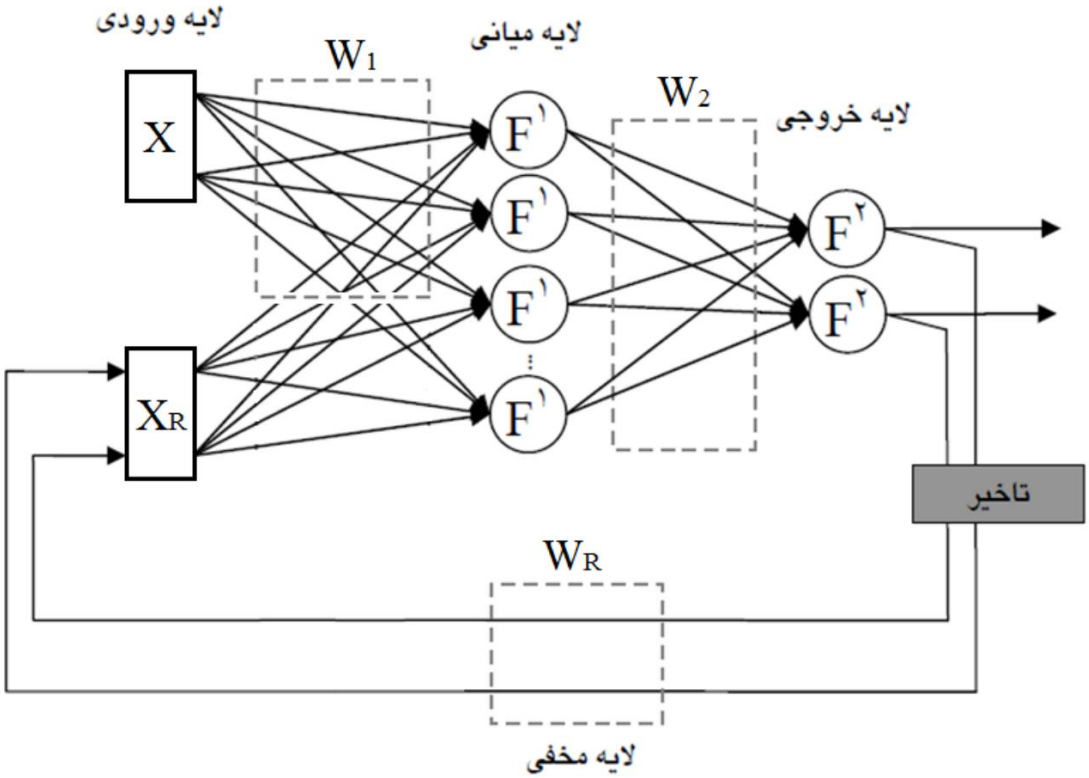
این شـبکه با اسـتفاده از ساده سازی خط جدا کننده و انتقال داده به ابعادی که جدا کردن داده ها آسان تر خواهد بود، عمل می کند.



شکل ۱- RBF [1]

۲,۳. Recurrent

عمده تفاوت در این شـبکه ها اسـتفاده از خروجی لایه های متفاوت شـبکه در زمانی های متفاوت، به عنوان ورود برای لایه ها می باشد. از جمله این شبکه ها شبکه Jordan است.



شکل ۲- Jordan [2]

در شبکه جردن از خروجی به ورودی وصل شده است.

۳. توابع آزمون – مجموعه داده

به منظور مقایسه عملکرد الگوریتم‌های مختلف از Mean Squared Error(MSE) استفاده می شود.

$$e = t - a$$

$$mse = \sum e^2.$$

داده های برای آزمون به صـورت 40 – 60 اسـت که ٤٠ درصـد از کل داده برای تست شبکه به کار رفته است.

۴. شرح عملکرد برنامه

حل مسـاله فوق، محیط شبیه سازی شامل یک دستگاه٤ GB حافظه اصلی ، 4 core cpuو نرم افزار Matlab 2016 می باشد.

۴,۱ آموزش شبکه عصبی پرسپترون

```
function mlpTrainingFunction( epochs, trainX, trainY, testX, testY,
trainFunc )
```

وظیفه این زیر برنامه، آموزش شـبکه عصبی با داده های داده شده و الگوریتم یادگیرنده مورد نظر می باشد. که خروجی آن mse و خروجی شبکه در آخرین مرحله از ۵ مرحله آموزش و تست است.

```
function rbTrainingFunction( epochs, trainX, trainY, testX, testY, goal,
spread, neuron )
```

وظیفه این زیر برنامه، آموزش شـبکه عصبی RBF با داده های داده شده و goal برای خطای کمینه و spread و تعداد نورون های لایه میانی مورد نظر می باشد. که خروجی آن mse و خروجی شبکه در آخرین مرحله از ۵ مرحله آموزش و تست است.

```
function rbeTrainingFunction( epochs, trainX, trainY, testX, testY,
spread )
```

وظیفه این زیر برنامه، آموزش شبکه عصبی RBF با داده های داده شده و spread مورد نظر می باشد. که خروجی آن mse و خروجی شبکه در آخرین مرحله از ۵ مرحله آموزش و تست است.

```
function myF1(x,y)
```

وظیفه این زیر برنامه، تولید داده برای تابع اول داده شـده در سـوال می باشد که با استفاده از ورودی خروجی آن را براساس تابع داده شده محاسبه می کند.

. function myF2(x,y,z)

وظیفه این زیر برنامه، تولید داده برای تابع دوم داده شـده در سـوال می باشد که با استفاده از ورودی خروجی آن را براساس تابع داده شده محاسبه می کند.

. function newrb(trainX, trainY, goal, spread, neuron)

وظیفه این زیر برنامه، آموزش براســـاس RBF اســـت. که مقدار خطای خروجی مورد نظر و تعداد نورون و spread را به عنوان ورودی میگیرد.

. function newrbe(trainX, trainY, spread)

وظیفه این زیر برنامه، آموزش براساس RBF است که مقدار خطای خروجی را ۰ در نظر میگیرد ولی همچنان spread را به عنوان یک پارامتر در نظر میگیرد.

. file train_test_f1

وظیفه این زیر برنامه، ساختن داده با استفاده از myF1 و ذخیره سازی آن می باشد.

. file train_test_f۲

وظیفه این زیر برنامه، ساختن داده با استفاده از myF2 و ذخیره سازی آن می باشد.

۵. شبیه سازی ها و نتایج

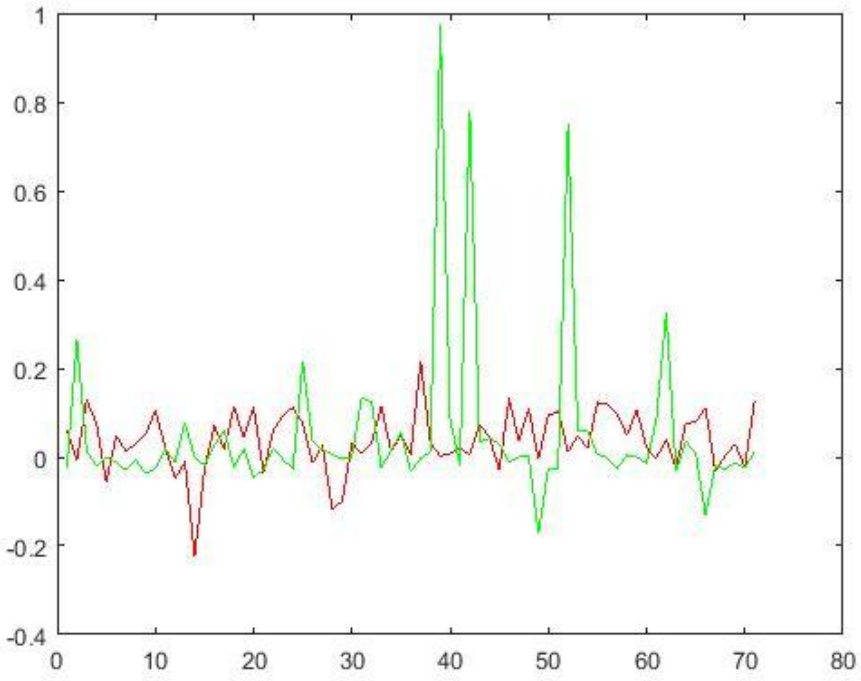
باید توجه شود که در rb خطا را ۰,۰۰۱ و تعداد نورون ها را در تمام جاها ۵ در نظر گرفته شده است.

۵,۱ سوال اول – تابع اول

$$y = \text{sinc}(x,y) = \frac{\sin(x)}{x} \times \frac{\sin(y)}{y}$$

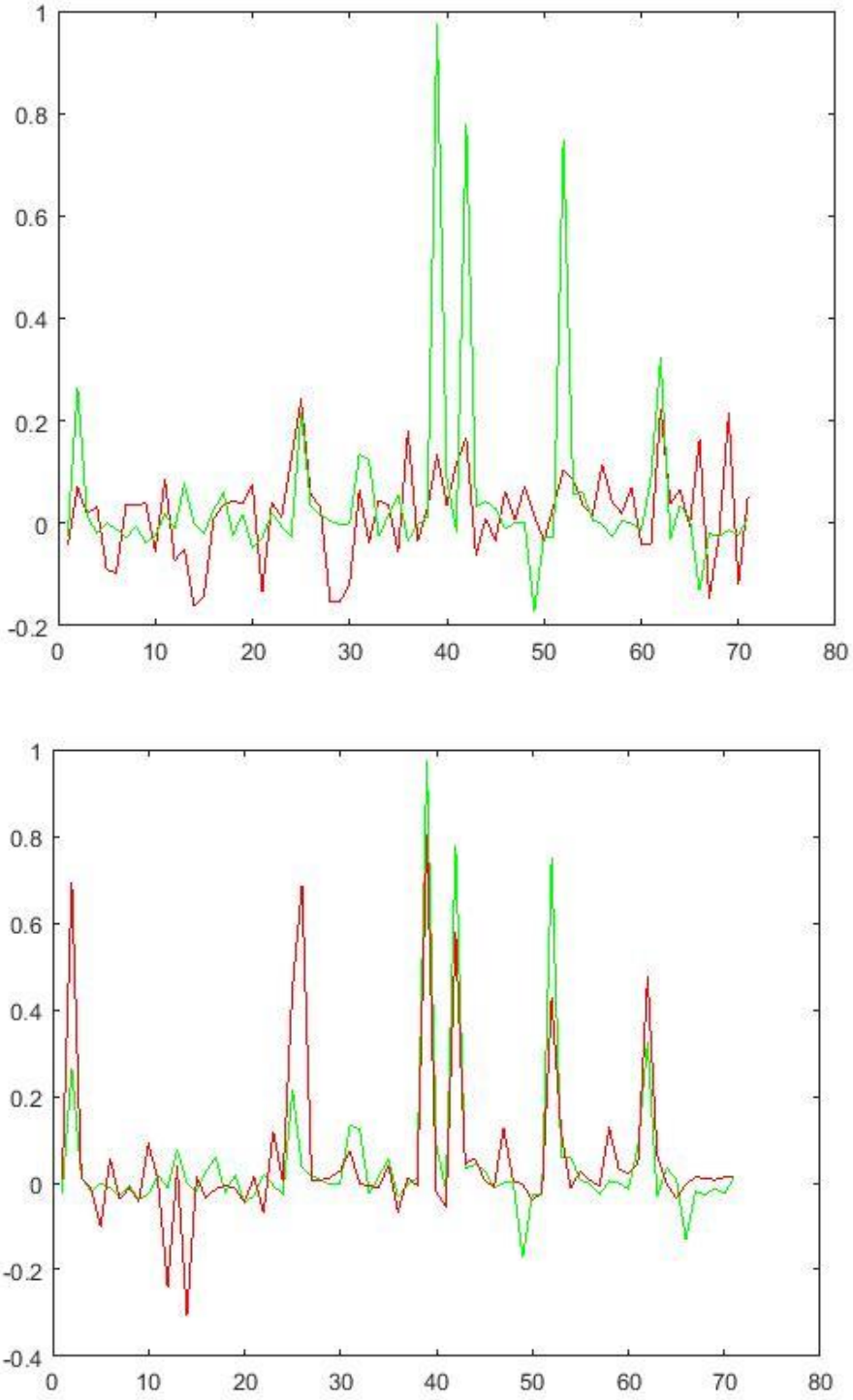
جدول ۱ - نتایج حاصل از شناسایی تابع f1 از روش های مختلف

Method	Mean	Best	Worst	STD	Time(s)
MLP-GD	0.0428	0.0391	0.0515	0.0050	4.7068
MLP-GDM	0.0391	0.0302	0.0725	0.0187	4.7058
MLP-LM	0.0245	0.0186	0.0297	0.0044	0.8381
NEWRB	0.0023	0.0023	0.0023	0	2.0877
NEWRBE	0.0016	0.0016	0.0016	0	0.6312

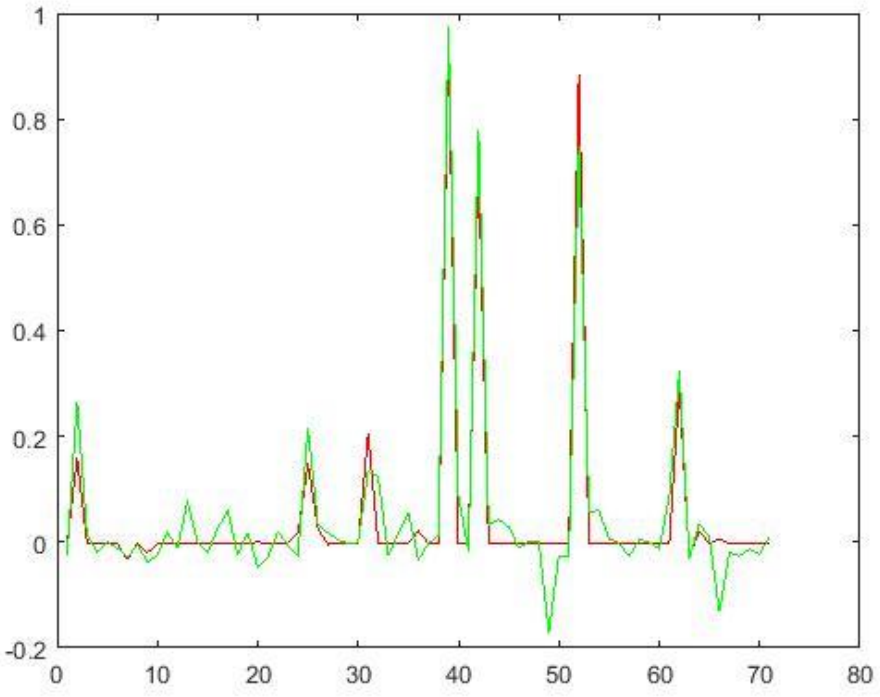


شکل ۳ - مقایسه خروجی شبکه با خروجی واقعی در الگوریتم MLP-GD در تابع f1

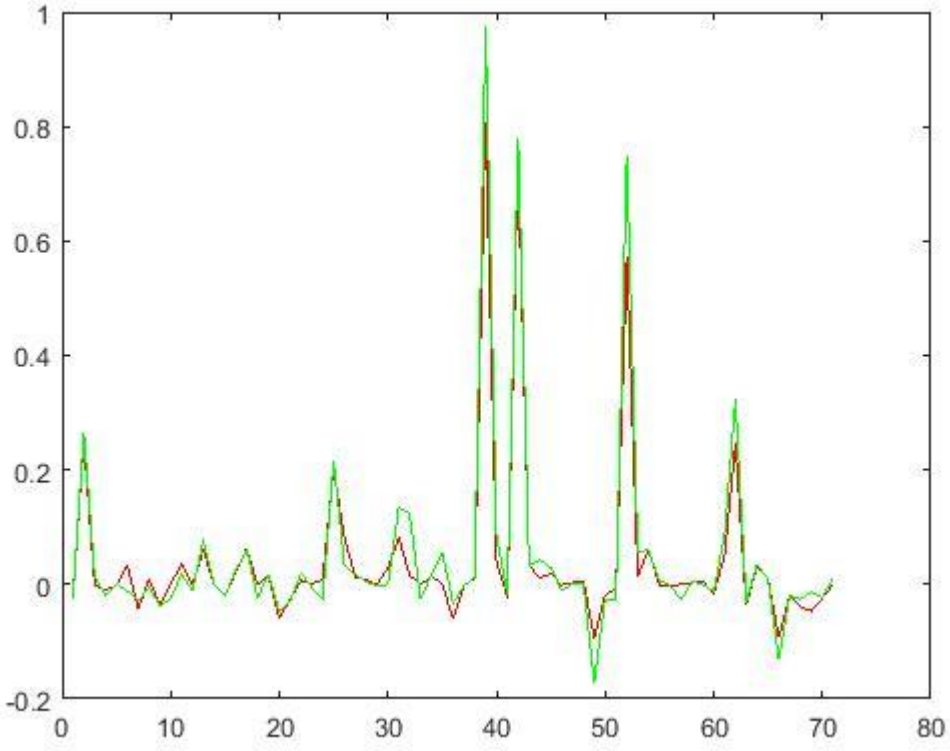
قرمز خروجی شبکه و سبز خروجی واقعی است.



شکل ۵ - مقایسه خروجی شبکه با خروجی واقعی در الگوریتم MLP-LM در تابع f1



شکل ۶- مقایسه خروجی شبکه با خروجی واقعی در الگوریتم NEWRB در تابع f1



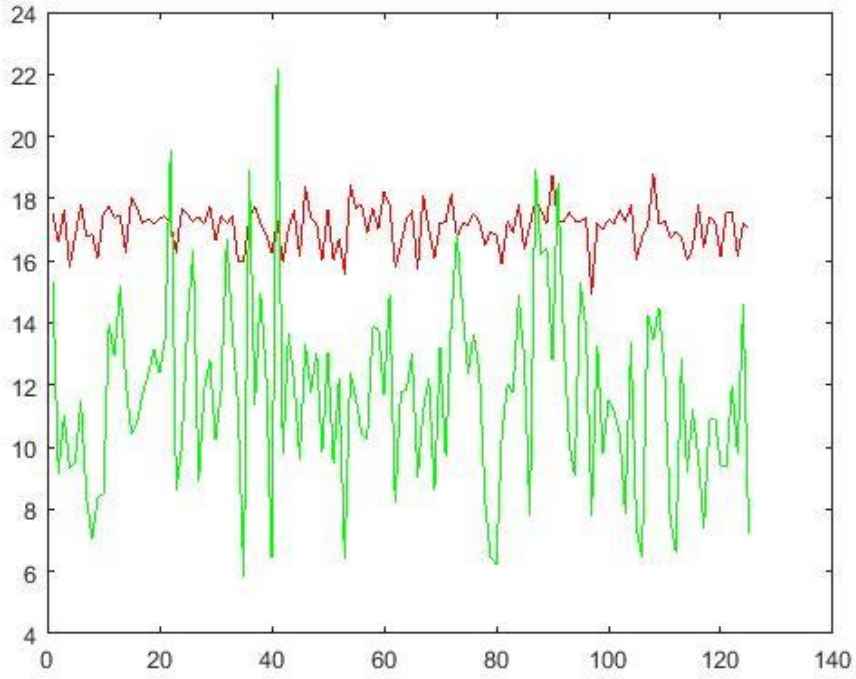
شکل ۷ - مقایسه خروجی شبکه با خروجی واقعی در الگوریتم NEWRBE در تابع f1

۵,۲. سوال اول – تابع دوم

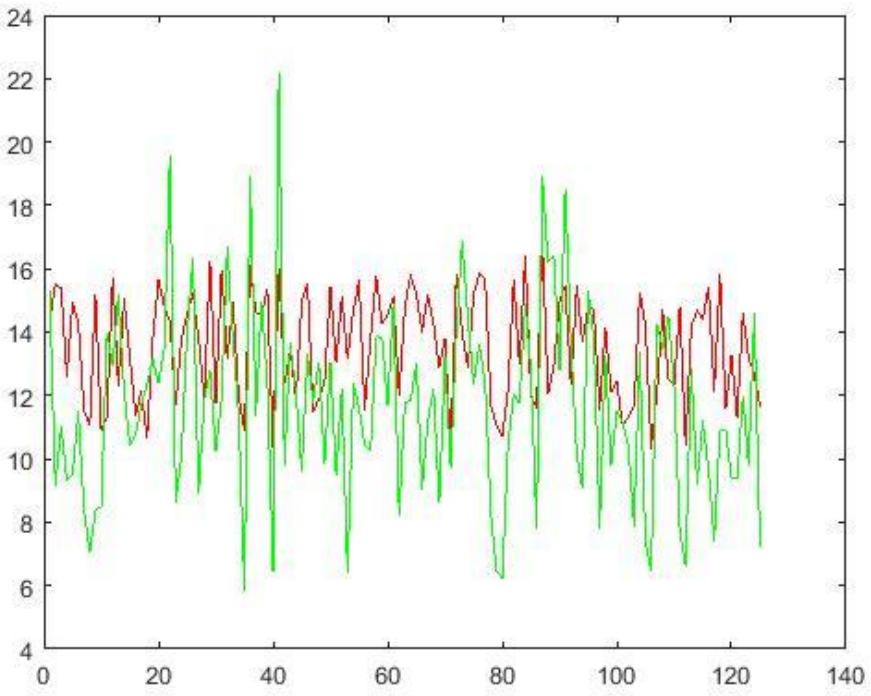
$$y = (1 + x^{0.5} + y^{-1} + z^{-1.5})^2$$

جدول ۲ - نتایج حاصل از شناسایی تابع $f1$ با استفاده از روش های مختلف

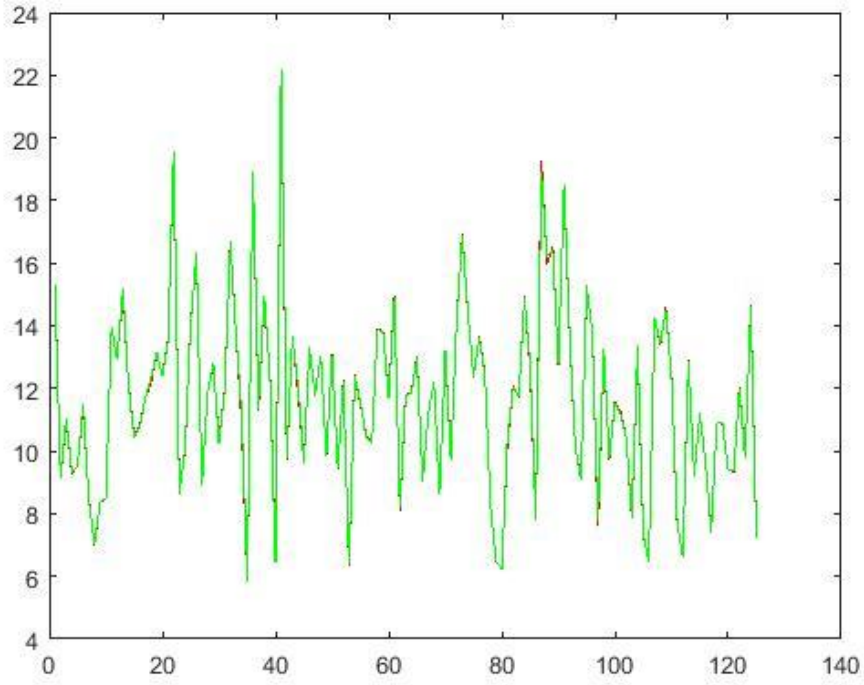
Method	Mean	Best	Worst	STD	Time(s)
MLP-GD	37.2479	37.2479	37.2479	0	0.8209
MLP-GDM	10.6741	9.2988	12.5779	1.1955	0.7711
MLP-LM	0.0031	6.3823e-05	0.0062	0.0030	4.1826
NEWRB	0.0421	0.0421	0.0421	0	26.4861
NEWRBE	0.4724	0.4724	0.4724	0	0.6501



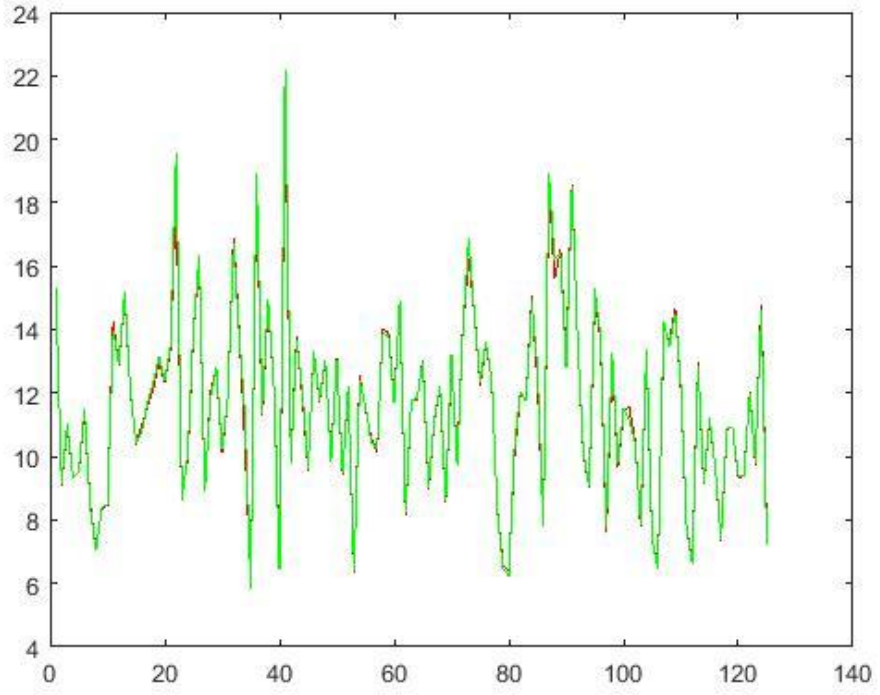
شکل ۸ - مقایسه خروجی شبکه با خروجی واقعی در الگوریتم MLP-GD در تابع f2



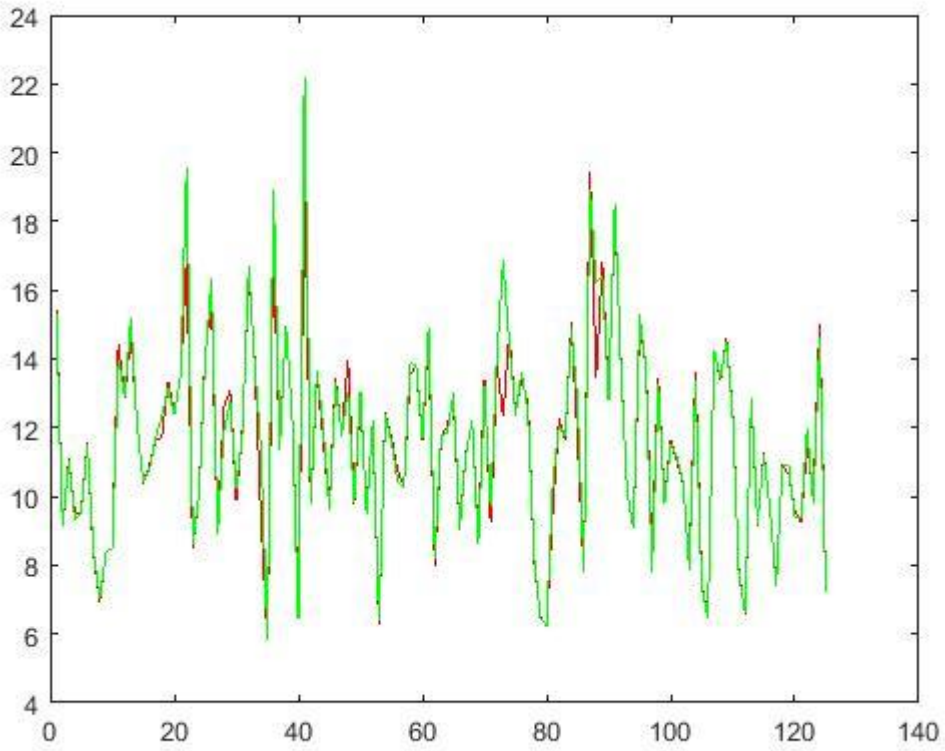
شکل ۹- مقایسه خروجی شبکه با خروجی واقعی در الگوریتم MLP-GDM در تابع f2



شکل ۱۰ - مقایسه خروجی شبکه با خروجی واقعی در الگوریتم MLP-LM در تابع f2



شکل ۱۱ - مقایسه خروجی شبکه با خروجی واقعی در الگوریتم NEWRB در تابع f2



شکل ۱۲ - مقایسه خروجی شبکه با خروجی واقعی در الگوریتم NEWRBE در تابع f2

همانطور که از نتایج بالا قابل مشاهده است در حالت توابع سینوسی rbe جواب بهتری داده است و در توابع چند جمله ای نیز در MLP-LM جواب بهتری بدست آمده است.

۵,۳. **سوال دوم** – CD Player Arm

Method	Mean Square Error	
	Y1	Y2
MLP-GD	0.0327	0.0263
MLP-GDM	0.3123	0.7941
MLP-LM	0.0131	0.0120
NEWRB	0.0131	0.0119
NEWRBE	0.0157	0.0144
Jordan		

۵,۴. سوال دوم – Steam Generator

Method	Mean Square Error			
	Y1	Y2	Y3	Y4
MLP-GD	1.6298e+04	326.3685	51.8944	102.7651
MLP-GDM	1.8024e+04	111.7177	90.0275	667.0146
MLP-LM	3.7132e+03	9.0445	4.6414	16.6834
NEWRB	4.6821e+03	11.8172	4.7462	21.7263
NEWRBE	50.6273	1.8734e+03	9.2758e+03	159.0960
Jordan				

ما بقیه شکلها در فولدر مربوطه است.

برای تولید نمودارها از دستورهای زیر استفاده شده است:

- plot(out, ‘r’)
- hold on
- plot(y , ‘g’)

مراجع

[1] A. Sharifi. Radial Basis Function.

[2] A. Sharifi. Recurrent Neural Network.