

گزارش کار

صادق حایری

برای ساخت داده‌های مورد نیاز برای یادگیری شبکه عصبی از تابع گفته شده در صورت سوال استفاده می‌کنیم.

تابع آن را به صورت روبرو می‌نویسیم:

```
1 function [ res ] = myfunc(a, b, c)
2 -     res = (-b + sqrt(abs(b ^ 2 - 4 * a * c))) / (2*a * sin(a * pi / 180));
3 - end
```

```
1 P = zeros(1, size);
2 T = zeros(1, size);
3 for i = 1:size;
4     P(1, i) = rand*100;
5     P(2, i) = rand*100;
6     P(3, i) = rand*100;
7
8     T(i) = myfunc(P(1, i), P(2, i), P(3, i));
9 end;
```

حال برای اینکه داده‌های مورد نیاز

را بسازیم، از اعداد رندوم در بازه

صفر با ۱۰۰ استفاده می‌کنیم و به

ورودی تابع می‌دهیم:

آرایه‌ی P ورودی‌های تابع را ایجاد

می‌کند (۳ در تعداد داده‌ها) و آرایه

T آرایه target یا همان y هست (۱ در تعداد داده‌ها)

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---------|---------|---------|---------|---------|---------|---------|------|
| 1 | 54.0982 | 62.8023 | 8.6653 | 39.1897 | 34.7701 | 35.3677 | 73.9252 | 82.0 |
| 2 | 32.1411 | 88.6232 | 20.7203 | 20.2727 | 12.8664 | 20.1659 | 3.1063 | 18.1 |
| 3 | 74.0528 | 63.1849 | 60.8969 | 94.4893 | 88.9812 | 21.2378 | 6.9848 | 15.9 |

آرایه P

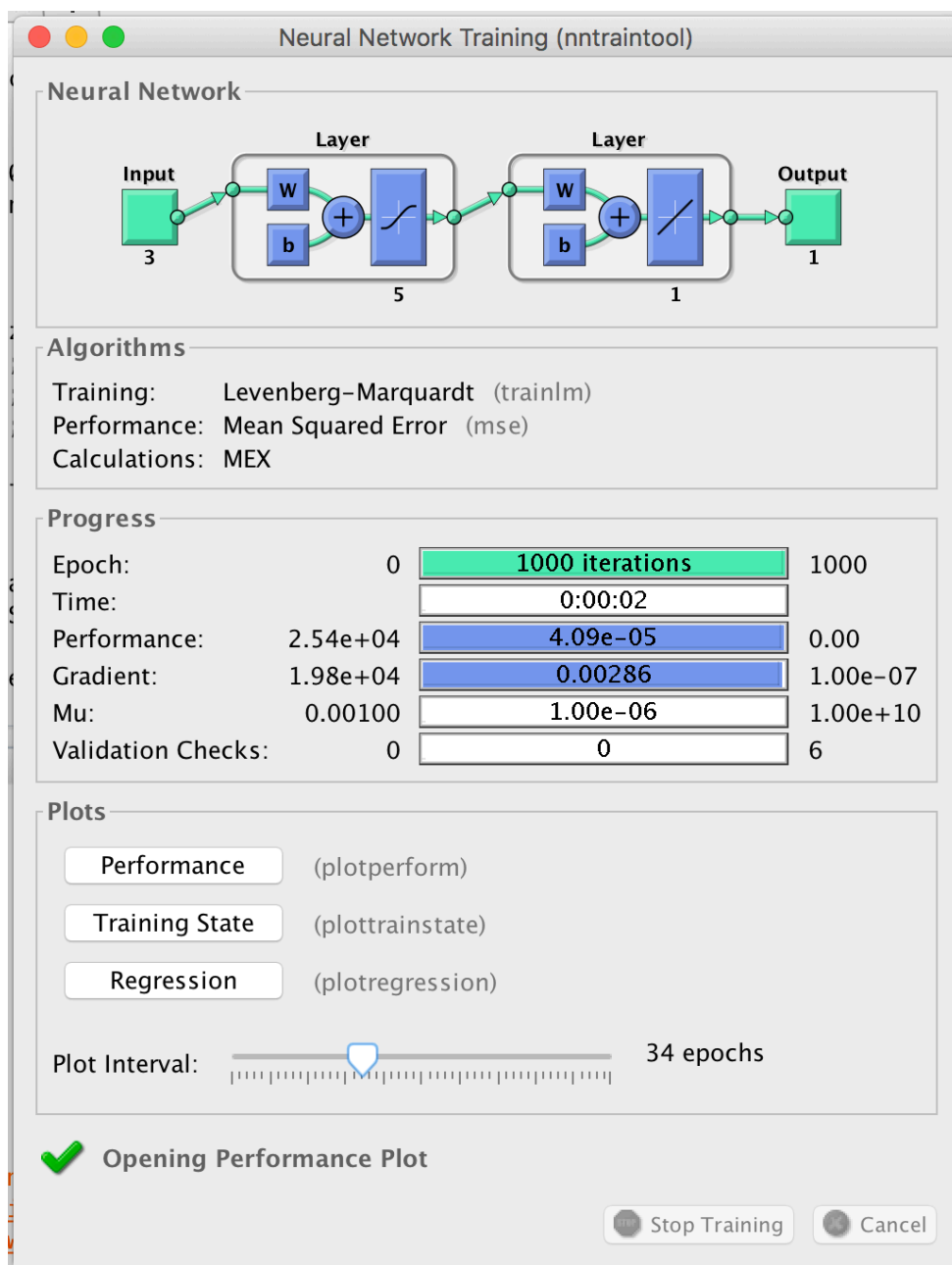
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|--------|--------|--------|--------|--------|--------|--------|-----|
| 1 | 1.0303 | 0.0083 | 7.7689 | 2.0137 | 2.4619 | 0.7523 | 0.2973 | 0.3 |

آرایه T

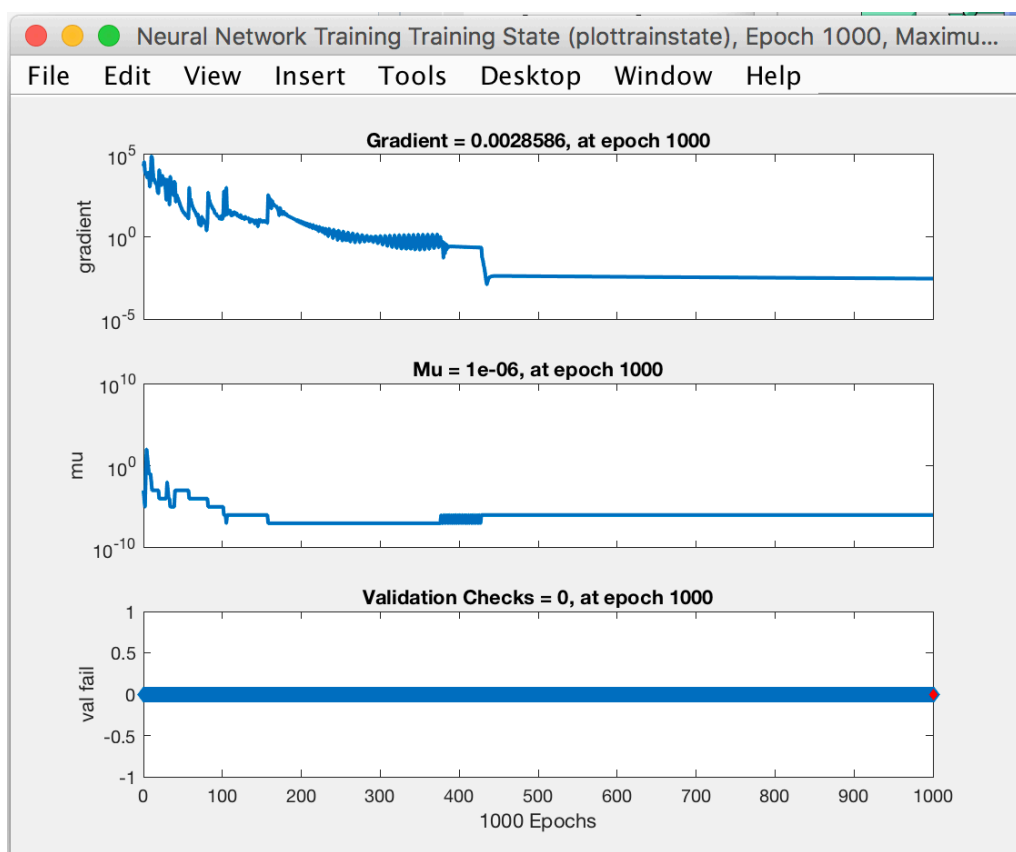
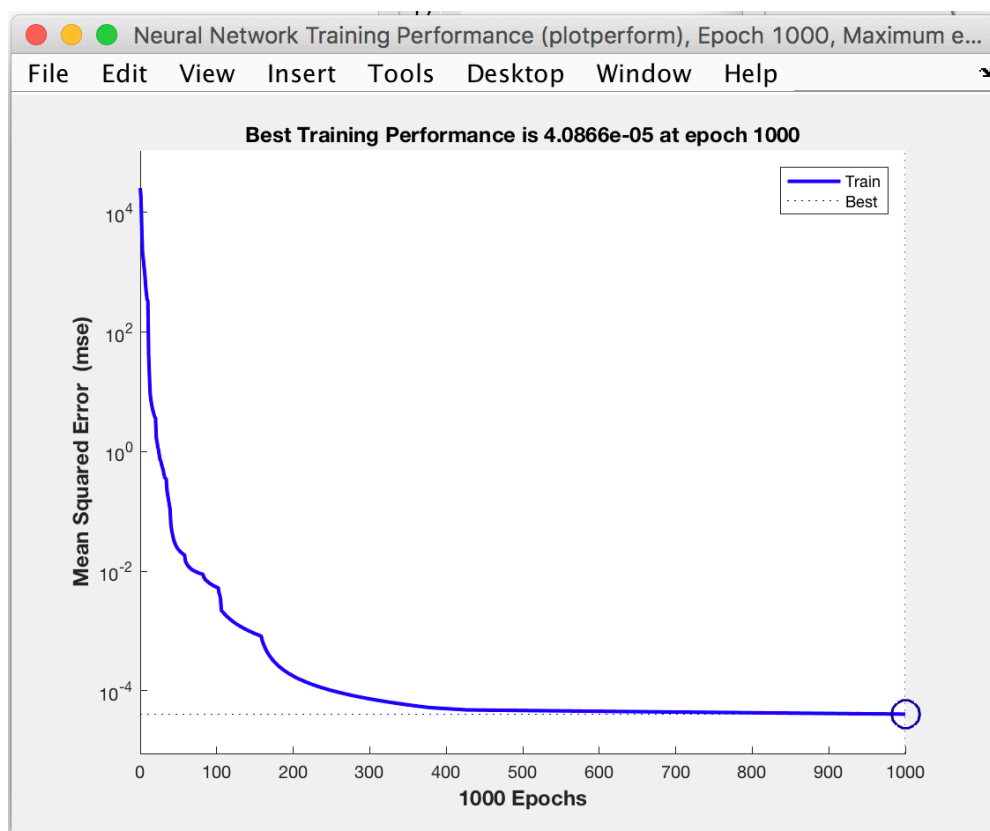
حال که دیتای ورودی و خروجی را داریم، می‌توانیم شبکه‌عصبی را تشکیل دهیم.
بهترین نتیجه‌ی گرفته شده با ۱۵ نود و یک لایه پنهان ایجاد شد:

```
1 net = newff([0 100; 0 100; 0 100],[15 1],{'tansig' 'purelin'});
2 [tnet,tr] = train(net, P, T);
3 plotperform(tr)
```

در زمان یادگیری تابع پنجره روبرو نشان داده می‌شود:



و نمودار Performance ما به شکل روبرو در می آید:



برای تست اینکه شبکه به خوبی کار می‌کند یا خیر دیتای تستی به سائز ۱۰۰ تا ایجاد می‌کنیم و میانگین اختلاف‌های آن‌ها را محاسبه می‌کنیم، و در نتیجه به عدد مناسبی برای خطا می‌رسیم:

```
1 - errSum = 0;
2 - for i = 1:testSize;
3 -     a = rand*100;
4 -     b = rand*100;
5 -     c = rand*100;
6
7 -     realAns = myfunc(a, b, c);
8 -     pridictAns = sim(tnet, [a; b; c;]);
9
10 -    err = abs(realAns - pridictAns / realAns);
11 -    errSum = errSum + err;
12 - end;
13 - err = errSum / testSize * 100;
14 - err
```

خطا ۳.۵ تا ۱۰.۶ درصد دریافت می‌شود.