

اگر شبکه ایجاد شده پاسخ خوبی ارائه نداده است راه هایی برای حل آن وجود دارد. میتوانید با دستور زیر پارامترهای شبکه را دوباره تنظیم کنید:

```
net = init(net);
```

```
net = train(net,houseInput,houseTargets);
```

الگوریتم یادگیری این شبکه به صورت پیش فرض **Levenberg-Marquardt** است. این الگوریتم بسیار سریع است اما نیاز به حافظه ی زیادی دارد. می توانید الگوریتم یادگیری را تغییر دهید تا هم حافظه کمتری استفاده شود و هم ممکن است عملکرد شبکه افزایش یابد.

همچنین می توانید تقسیم بندی داده ها را برای آموزش، اعتبارسنجی و تست تغییر دهید. چهار تابع برای انجام این کار وجود دارد. تابع **dividerand** که حالت پیش فرض هم از همین تابع استفاده می کند. همچنین توابع **divideblock**، **divideint** و **divideind** نیز برای تقسیم بندی استفاده می شوند. با استفاده از ویژگی زیر می توانید به توابع تقسیم بندی دسترسی داشته باشید و آن ها را تغییر دهید:

```
net.divideFcn
```

:Index Data Division (divideind)

در ابتدا داده ای تولید کنید. برای این کار دستورات زیر را اجرا کنید:

```
p = [-1:0.01:1];
```

این دستور ۲۰۱ داده از -۱ تا ۱ با گام های ۰.۱ تولید می کند. همچنین خروجی را با دستور زیر می توانید تولید کنید:

```
t = sin(2*pi*p)+0.1*randn(size(p));
```

حال ایندکس هایی را تولید می کنیم تا مشخص کنیم کدام نمونه برای کدام بخش آموزش، اعتبارسنجی و یا تست استفاده شود:

```
trainInd = 1:3:201
```

```
valInd = 2:3:201;
```

testInd = 3:3:201;

حال از این ایندکس ها می توانید به صورت زیر استفاده کنید:

[trainP, valP, testP] = divideind(p, trainInd, valInd, testInd);

با دستور بالا ورودی ها را تقسیم بندی کردید. با دستور زیر خروجی ها را تقسیم بندی کنید:

[trainT, valT, testT] = divideind(t, trainInd, valInd, testInd);

در هر کدام از پارامترهای trainP, valP, testP مقادیر نمونه ها متناظر با ایندکس های مشخص شده وجود دارند.

Random Data Division (dividerand)

همچنین می توانید داده ها را به صورت تصادفی به نسبت ۶۰٪، ۲۰٪ و ۲۰٪ تقسیم کنید برای این کار دستور زیر را اجرا کنید:

[trainP, valP, testP, trainInd, valInd, testInd] = dividerand(p);

با این دستور ایندکس ها هم تولید می شوند و به راحتی از ایندکس های تولید شده می توانید خروجی را نیز گروه بندی کنید:

[trainT, valT, testT] = divideind(t, trainInd, valInd, testInd);

Block Data Division (divideblock)

با این دستور ۶۰٪، ۲۰٪ بعدی و ۲۰٪ نهایی به ترتیب برای آموزش، اعتبار سنجی و تست استفاده می شود.

[trainP, valP, testP, trainInd, valInd, testInd] = divideblock(p);

[trainT, valT, testT] = divideind(t, trainInd, valInd, testInd);

Interleaved Data Division (divideint)

با استفاده از این دستور، داده ها به صورت پند در میان برای بخش های مختلف تقسیم می شوند.

```
[trainP,valP,testP,trainInd,valInd,testInd] = divideint(p);
```

```
[trainT,valT,testT] = divideind(t,trainInd,valInd,testInd);
```

نسبت این تقسیم نیز، به صورت ۶۰، ۲۰ و ۲۰ می باشد. اگر دستور بالا را اجرا کنید می توانید در صفحه ی **workspace** می توانید ایندکس ها را مشاهده کنید تا متوجه شوید تقسیم بندی به چه صورتی اجرا می شود.

www.matlabproject.ir