

Starting with neural network in matlab

The neural networks is a way to model any input to output relations based on some input output data when nothing is known about the model. This example shows you a very simple example and its modelling through neural network using MATLAB.

Actual Model

Let us take that our model has three inputs a,b and c and generates an output y. For data generation purposes, let us take this model as

$$y=5a+bc+7c;$$

we are taking this model for data generation. In actual cases, you dont have the mathematical model and you generate the data by running the real system.

Let us first write a small script to generate the data

```
a= rand(1,1000);  
b=rand(1,1000);  
c=rand(1,1000);  
n=rand(1,1000)*0.05;  
  
y=a*5+b.*c+7*c+n;
```

n is the noise, we added deliberately to make it more like a real data. The magnitude of the noise is 0.1 and is uniform noise.

So our input is set of a,b and c and output is y.

```
I=[a; b; c];  
T=y;
```

Understanding Neural Networks

Neural network is like brain full of nerons and made of different layers.

The first layer which takes input and put into internal layers or hidden layers are known as input layer.

The outer layer which takes the output from inner layers and gives it to outer world is known as output layer.

The internal layers can be any number of layers.

Each layer is a basically a function which takes some variables (in the form of vector u) and transforms it to another variable(another vector v) by multiplying it with coefficients and adding some biases b . These coefficient is known as weight matrix w . Size of the v vector is known as v -size of the layer.

$$v=sum(w.*u)+b$$

So we will make a very simple neural network for our case- 1 input and 1 output layer. We will take the input layer v -size as 5. Since we have three input , our input layer will take u with three values and transform it to a vector v of size 5. and our output layer now take this 5 element vector as input u and transforms it to a vector of size 1 because we have only on output.

Creating a simple Neural FF Network

We will use matlab inbuilt function newff for generation of model.

First we will make a matrix R which is of 3 * 2 size. First column will show the minimum of all three inputs and second will show the maximum of three inputs. In our case all three inputs are from 0 to 1 range, So

```
R=[0 1; 0 1 ; 0 1];
```

Now We make a Size matrix which has the v-size of all the layers.

```
S=[5 1];
```

Now call the newff function as following

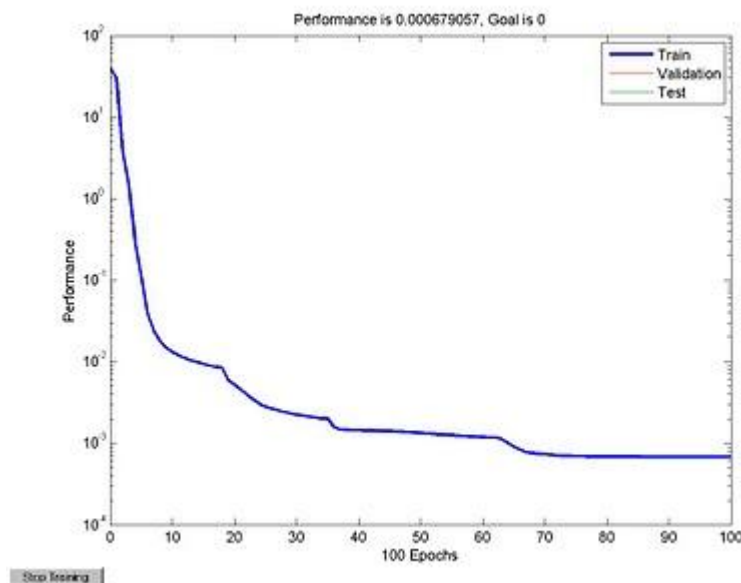
```
net = newff([0 1;0 1 ;0 1],[4 1],{'tansig','purelin'});
```

net is the neural model. {'tansig','purelin'} shows the mapping function of the two layers. Let us not waste time on this.

Now as each brain need training, this neural network too need it. We will train this neural network with the data we generated earlier.

```
net=train(net,I,O);
```

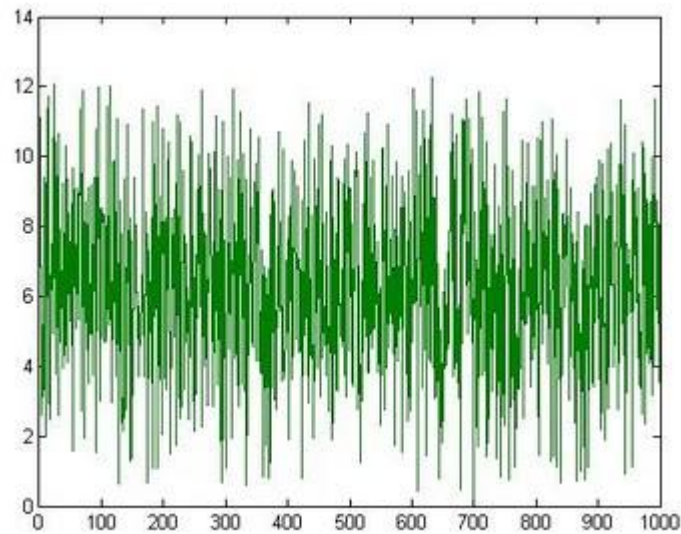
Now net is trained. You can see the performance curve, as it gets trained.



So now simulate our neural network again on the same data and compare the out.puts.

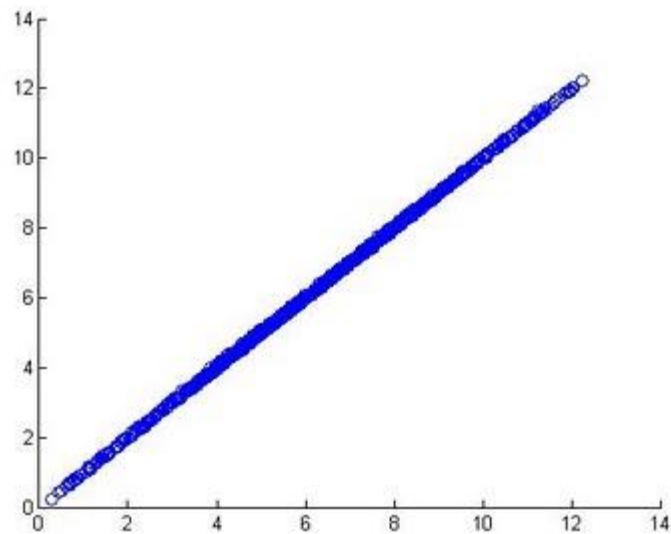
```
O1=sim(net,I);
```

```
plot(1:1000,O,1:1000,O1);
```



You can observe how closely the the two data green and blue follow each other.
Let us try scatter plot between simulated output and actual target output.

```
scatter(O,O1);
```



Let us observe the weight matrix of the trained model.

```
net.IW{1}
-0.3684    0.0308   -0.5402
    0.4640    0.2340    0.5875
    1.9569   -1.6887    1.5403
    1.1138    1.0841    0.2439
net.LW{2,1}
-11.1990    9.4589   -1.0006   -0.9138
```

Now test it again on some other data. What about $a=1, b=1$ and $c=1$;

So input matrix will be $[1 \ 1 \ 1]'$;

```
y1=sim(net,[1 1 1]');
```

you will see 13.0279. which is close to 13 the actual output ($5*1+1*1+12*1$);