

Sample Training Sessions

1. Build a 1-10-1 network to model the `simplefit_dataset` that comes with Matlab. That is, in Matlab type:

```
[x,t]=simplefit_dataset; %Data comes with Matlab
plot(x,t);
%Initialize the network using 10 nodes in the hidden layer.
net=feedforwardnet(10);
net=train(net,x,t);
```

At this point, a dialog box comes up to illustrate how the training is proceeding. Once done, you might take a look at some of the plots- In particular, the “Performance” plot and the “Error Histogram” are kind of interesting.

Try using a different form of the training command, and try reducing the number of hidden nodes.

```
net=feedforwardnet(5);
[net,tr]=train(net,x,t);
```

The output `tr` is a Matlab structure that contains a lot of information. To see the plot from the training, you can type:

```
h=length(tr.perf);
plot(1:h,log(tr.perf),1:h,log(tr.vperf),1:h,log(tr.tperf))
```

2. PROBLEM: Build an 8-15-15-2 feedforward neural network that uses `trainrp` for the training method, and uses a training goal of 0.05. Train the data on the Matlab file on our class website, `diabetes1.mat`

SOLUTION:

```
net=feedforwardnet([15 15]);
net.trainParam.goal=0.05;
net.trainFcn='trainrp';
load diabetes1; %You have to download this
net=train(net,P',T'); %The data needs to be transposed.
```

3. PROBLEM: Get some random data for your input and output- Say 350 points in $\mathbb{R}^3$ for the domain, and 350 points in $\mathbb{R}^2$ for the range. Build a 3 – 8 – 5 – 2 network and train it (output `tr` as well as `net`).

- (a) Plot the performance measures from the training output (not the dialog box).

- (b) There are three sets of weights. Where are they in the network structure?
 - (c) There are three bias vectors. Where are they in the network structure?
 - (d) Find where the transfer function is defined for each layer (Hint: **layers**) and change all of them to **logsig** (except input and output layers). Re-train.
 - (e) Investigate the “Plot Regression” option. In particular, the program finds the linear relationship between the “outputs” and the “targets” (the closer that R is to 1, the better). Use **plotregression** directly to see what relationship you get.
4. Use the **simpleclass_dataset** (a built in data set) to try a pattern classifier:

```
load simpleclass_dataset;
net=feedforwardnet(20);
net=train(net,simpleclassInputs, simpleclassTargets);
SampleOut=sim(net,simpleclassInputs);
plotconfusion(simpleclassTargets,SampleOut);
```

5. Did your previous network perform any pre-processing of the data? To see and/or change the option, go to:

```
net.inputs{1}
net.outputs{1}
```

The preprocessing options are: **mapminmax**, **mapstd**, **processpca**, **fixunknowns**, and **removeconstantrows**