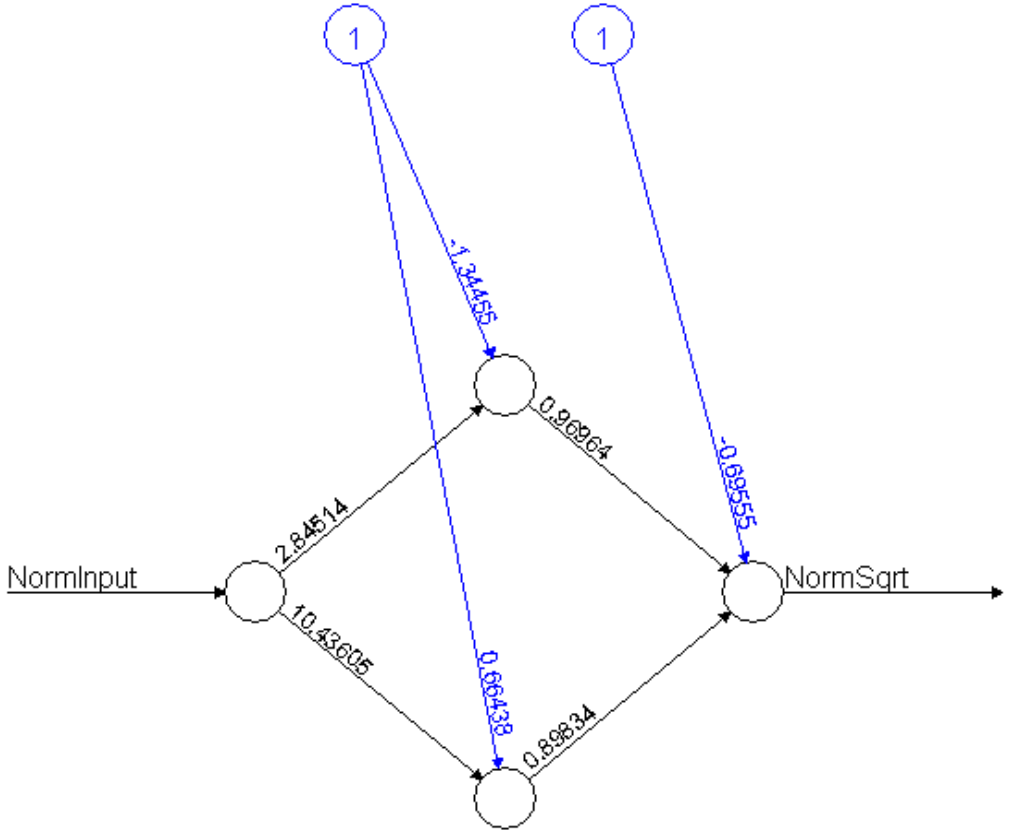
```
R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

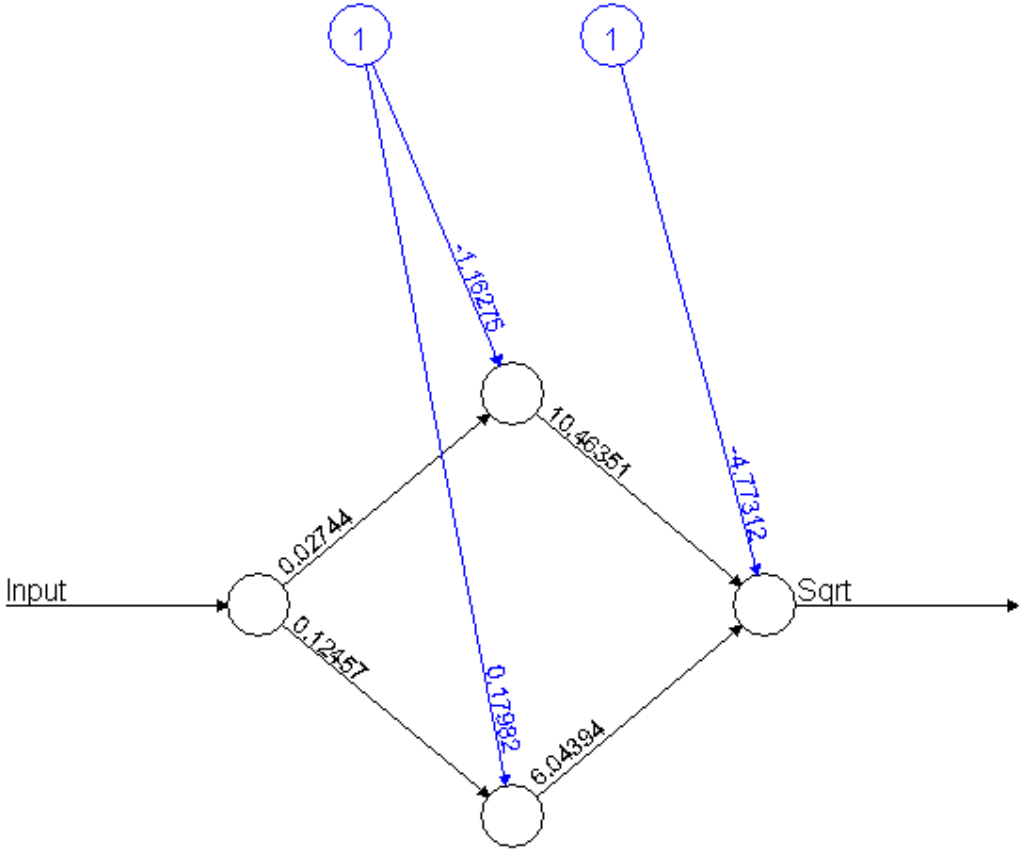
```
> library("neuralnet")
> #get random numbers between 0 and 100
> data <-  as.data.frame(runif(100, min=0, max=100))
> colnames(data)<-c("Input")
> data$Sqrt<-sqrt(data$Input)


> #Normalize data so that inputs are between 0 and 1
> data$NormInput<-data$Input/100
> data$NormSqrt<-data$Sqrt/10


> #create a neural net with 2 hidden nodes using the first 50 numbers for training
> netSqrt<-neuralnet(NormSqrt~NormInput,data[1:50,], hidden=2, threshold=0.0001)
> #use first 50 rows for training and next 50 for testing
> netResults <- compute(netSqrt, data$NormInput[51:100])
```

```
>  test<-cbind(data$NormSqrt[51:100],netResults$net.result)
>  colnames(test)<-c("NormSqrt","Guess")
>  test <- as.data.frame(test)
>  test$Sqrt<-test$NormSqrt * 10
>  test$Guess10<-test$Guess*10
>  test$err<-abs(test$Guess10-test$Sqrt)
>  mean(test$err/test$Sqrt)
[1] 0.004719935694
>  #error is .5% compared with 4% for the model we built in Excel.
>  #Ours did not use an optimized learning rate and we only did 5000 iterations
>  #this one did 35,514 steps
>  plot(netSqrt)
```



Error: 0.000407 Steps: 35514

```
>  #let's see what happens if we don't normalize the data
>  #we shouldn't need to since, there is only one input variable
>  netSqrt<-neuralnet(Sqrt~Input,data[1:50,], hidden=2, threshold=0.01)
>  netResults<-compute(netSqrt,data[51:100,1:1])
>  test<-cbind(data$Sqrt[51:100],netResults$net.result)
>  colnames(test)<-c("Sqrt","Guess")
>  test <- as.data.frame(test)
>  test$err<-abs(test$Guess-test$Sqrt)
>  mean(test$err/test$Sqrt)
[1] 0.005361858356
>  #.5% error again; roughly same number of steps
>  plot(netSqrt)
```



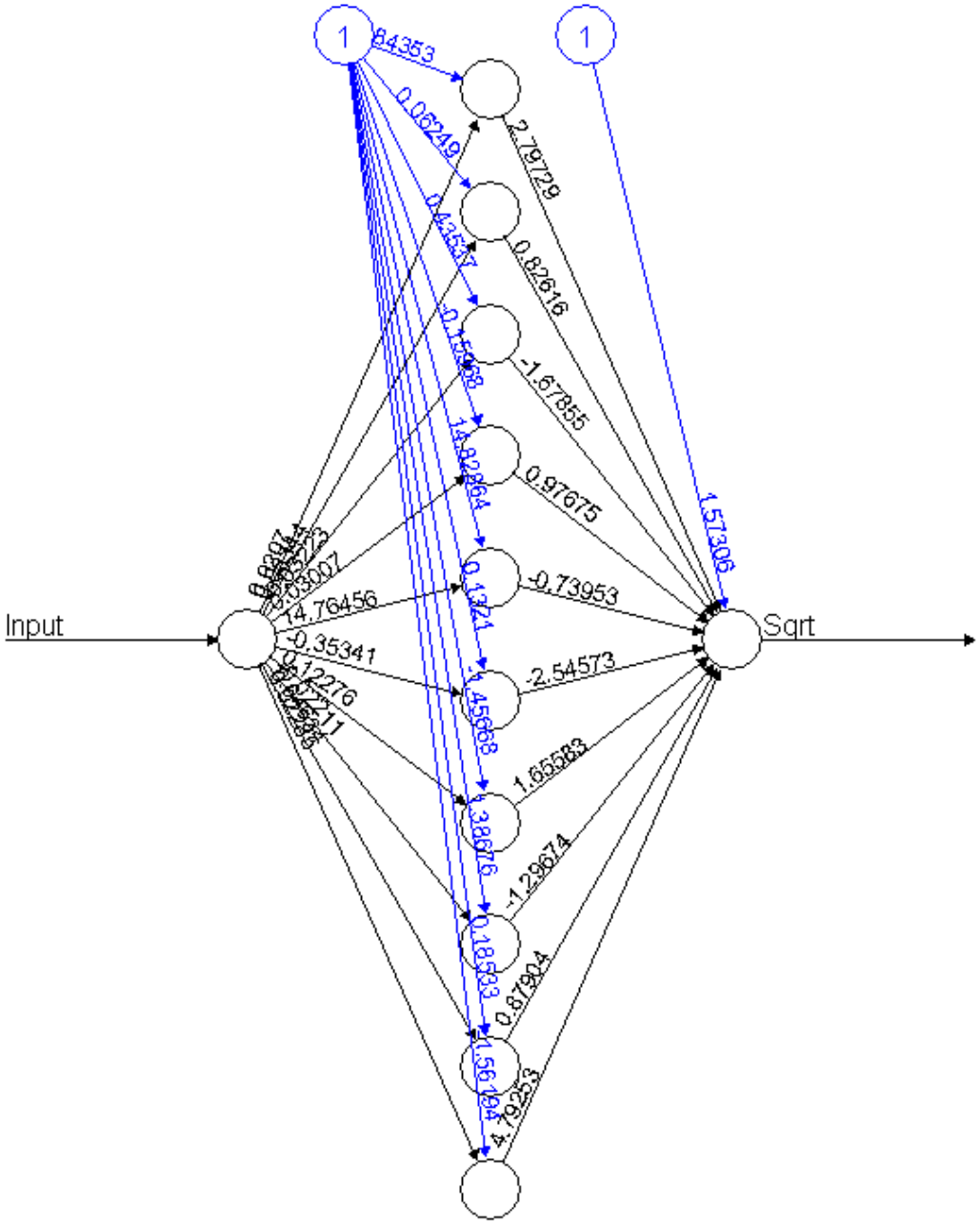
Error: 0.048737 Steps: 35056

```
>  #What happens if we print the neural net
> print(netSqrt)
Call: neuralnet(formula = Sqrt ~ Input, data = data[1:50, ], hidden = 2,     threshold = 0.01)

1 repetition was calculated.

          Error Reached Threshold Steps
1 0.04873716842    0.009318378238 35056

>  #multiply err column by 100 to get % err
>  test$err<-100*abs(test$Guess-test$Sqrt)
>  mean(test$err/test$Sqrt)
[1] 0.5361858356
```

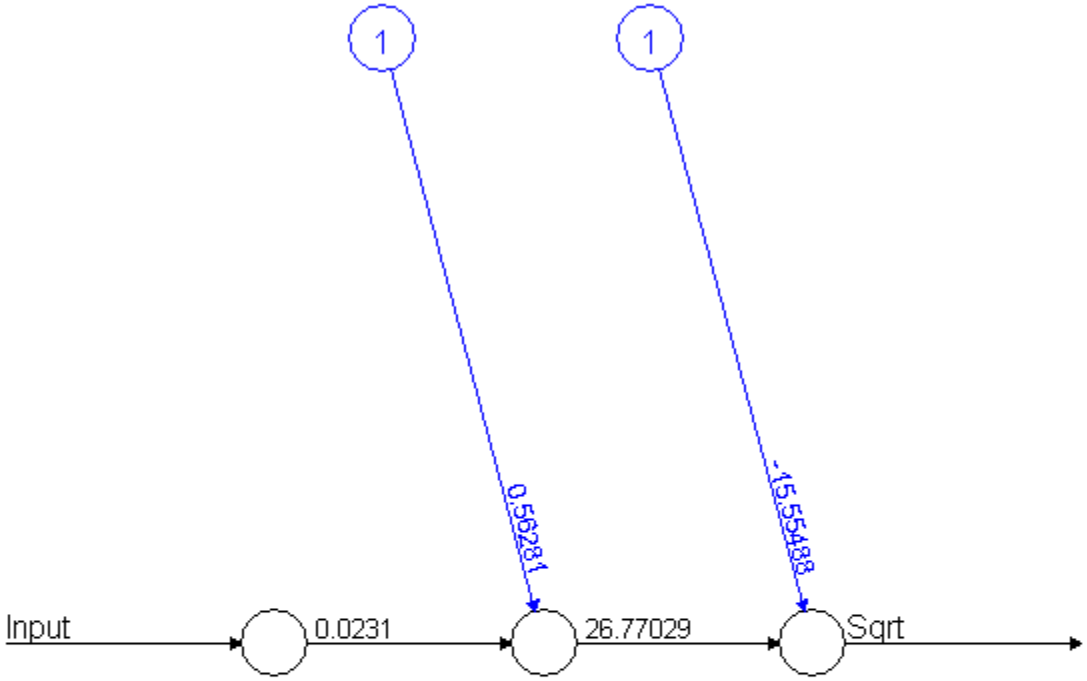
```
> #let's look at a few records
> test$input<-test$Sqrt*test$Sqrt
> test$GuessSqr<-test$Guess*test$Guess
> test[1:10,]
          Sqrt       Guess          err       input    GuessSqr
1  8.899407981 8.941720098 4.2312117461 79.19946240 79.95435831
2  7.292895577 7.266500651 2.6394926229 53.18632589 52.80203170
3  5.974000885 5.965487750 0.8513135148 35.68868658 35.58704410
4  8.298297160 8.323590013 2.5292852518 68.86173576 69.28215070
5  6.081867503 6.066896486 1.4971017326 36.98911233 36.80723297
6  8.941043124 8.983272157 4.2229032186 79.94225215 80.69917864
7  5.314329879 5.354647662 4.0317782766 28.24210207 28.67225158
8  7.420931002 7.399665499 2.1265502983 55.07021693 54.75504949
9  6.076708054 6.062029551 1.4678502734 36.92638077 36.74820228
10 8.518179411 8.552935992 3.4756581451 72.55938047 73.15271408
```

```
> #by the way, neural nets are good for interpolating; lousy at extrapolating
> test2 <-  as.data.frame(runif(100, min=101, max=200))
> test2Results<-compute(netSqrt,test2)
> colnames(test2)=c("Input")
> test2$Sqrt<-sqrt(test2$Input)
> test2$err<-100*abs(test2$Sqrt-test2$Guess)/test2$Sqrt
> test2[1:15,]
         Input         Sqrt        Guess           err
1  193.4182718 13.90748977 11.57111966 16.799366035
2  177.3041215 13.31555938 11.48254093 13.765989050
3  144.0705429 12.00293893 11.12925557  7.278911947
4  108.8815507 10.43463227 10.28174823  1.465159820
5  130.1204288 11.40703418 10.87035196  4.704835757
6  139.3692023 11.80547341 11.05137376  6.387712050
7  112.5954978 10.61110257 10.40461052  1.946000000
8  151.9827041 12.32812654 11.24178685  8.811879849
9  197.8639884 14.06641349 11.58960459 17.607963125
10 141.4628515 11.89381568 11.08714686  6.782254246
11 151.6691175 12.31540164 11.23773215  8.750583369
12 190.7420231 13.81093853 11.55889112 16.306259023
13 142.5891396 11.94106945 11.10565985  6.996103666
14 107.2981910 10.35848401 10.22653713  1.273804985
15 111.6751427 10.56764603 10.37501770  1.822812100
> mean(test2$err)
[1] 7.803518772
> #7.8% error for extrapolation compared with .5% for interpolation



> #lets create a model with 10 nodes in the hidden layer and see if it works better
>  netSqrt<-neuralnet(Sqrt~Input,data[1:50,], hidden=10, threshold=0.01)
>  netResults<-compute(netSqrt,data[51:100,1:1])
>  test<-cbind(data$Sqrt[51:100],netResults$net.result)
>  colnames(test)<-c("Sqrt","Guess")
>  test <- as.data.frame(test)
>  test$err<-abs(test$Guess-test$Sqrt)
>  mean(test$err/test$Sqrt)
[1] 0.001441720024
> #.14% compared with .5% for 2 nodes; and it took only 3902 steps to converge
```

```
> plot(netSqrt)
```



```
> print(netSqrt)
Call: neuralnet(formula = Sqrt ~ Input, data = data[1:50, ], hidden = 10,     threshold = 0.01)

1 repetition was calculated.

           Error Reached Threshold Steps
1 0.002974978503    0.008345310477  3902

>  #what if we only use one hidden node?
>  netSqrt<-neuralnet(Sqrt~Input,data[1:50,], hidden=1, threshold=0.01)
>  netResults<-compute(netSqrt,data[51:100,1:1])
>  test<-cbind(data$Sqrt[51:100],netResults$net.result)
>  colnames(test)<-c("Sqrt","Guess")
>  test <- as.data.frame(test)
>  test$err<-abs(test$Guess-test$Sqrt)
>  mean(test$err/test$Sqrt)
[1] 0.02664707248
>  plot(netSqrt)
```



Error: 0.755658 Steps: 39627

```
> test[1:15,]
          Sqrt       Guess           err
1  8.899407981 8.972481192 0.07307321137
2  7.292895577 7.389268118 0.09637254094
3  5.974000885 5.864715560 0.10928532486
4  8.298297160 8.430542609 0.13224544845
5  6.081867503 5.992142625 0.08972487830
6  8.941043124 9.007485104 0.06644197957
7  5.314329879 5.090338336 0.22399154285
8  7.420931002 7.529746762 0.10881576067
9  6.076708054 5.986048775 0.09065927834
10 8.518179411 8.636540712 0.11836130181
11 4.958163631 4.682068199 0.27609543266
12 9.442216539 9.402017870 0.04019866848
13 9.801737451 9.654200943 0.14753650821
14 8.280617067 8.413605418 0.13298835145
15 3.367950552 3.057240885 0.31070966635
> #We get a 2% error, which is not awful; took 39,627 iterations
> #Go to Excel to see how to implement finished model, outside of R
```