Let’s take a look at ANN’s in R. The ‘neuralnet’ package works nicely and has a nice visual representation of the ANN’s built.

We will start building a neural network for classifying Iris flowers.
Train our ANN

# load our data set
data(iris)

# make sure the ANN library is available
library(neuralnet)

# convert the labels into numeric labels and put them into a data frame
Species.numeric <- as.numeric(iris$Species)
iris.df <- data.frame(iris, Species.numeric)

# train a neural network with two hidden nodes
net <- neuralnet(
  Species.numeric ~ Sepal.Width + Sepal.Length + Petal.Width + Petal.Length, 
  iris.df, 
  threshold=0.01, 
  stepmax="10000", 
  lifesign="none", 
  hidden=2)
The data set:

```r
> iris.df[1:5,]
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species Species.numeric
 1       5.1       3.5         1.4       0.2  setosa          1
 2       4.9       3.0         1.4       0.2  setosa          1
 3       4.7       3.2         1.3       0.2  setosa          1
 4       4.6       3.1         1.5       0.2  setosa          1
 5       5.0       3.6         1.4       0.2  setosa          1
```

```r
> levels(iris.df$Species)
[1] "setosa" "versicolor" "virginica"
```
Evaluate our ANN

# display the ANN
plot(net)

# the training predictions from the ANN are numeric values,
# turn them into labels by rounding
predicted.labels <- round(net$net.result[[1]])

# plot the confusion matrix
print(table(iris.df$Species.numeric, predicted.labels))

The Confusion Matrix

<table>
<thead>
<tr>
<th>predicted.labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  2  3</td>
</tr>
<tr>
<td>1  50  0  0</td>
</tr>
<tr>
<td>2   0  49  1</td>
</tr>
<tr>
<td>3   0  1  49</td>
</tr>
</tbody>
</table>
ANN in R

Error: 0.839655  Steps: 90430