Neural nets

W. Grossmann

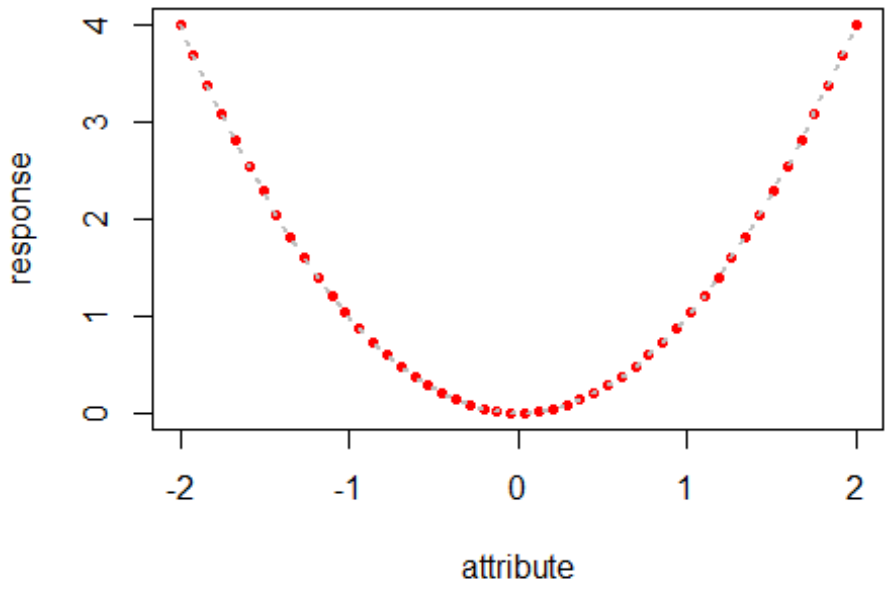
24 Mai 2017

```
#============================================================
# Neural nets with R
#============================================================

library(neuralnet)

#============================================================
# Example Regression y=x*x
#============================================================

#Generation of input data
#------------------------------------------------------------
set.seed(2016)

attribute<- as.data.frame(sample(seq(-2, 2, length=50), 50, replace=FALSE), n
col=1)

#------------------------------------------------------------
#Output
response<- attribute^2

#------------------------------------------------------------
# Function data
#------------------------------------------------------------
data<- cbind(attribute,response)
colnames(data) <- c('attribute', 'response')
head(data, 10)
```

```
##     attribute   response
## 1  -1.2653061 1.60099958
## 2  -1.4285714 2.04081633
## 3   1.2653061 1.60099958
## 4  -1.5102041 2.28071637
## 5  -0.2857143 0.08163265
## 6  -1.5918367 2.53394419
## 7   0.2040816 0.04164931
## 8   1.1020408 1.21449396
## 9  -2.0000000 4.00000000
## 10 -1.8367347 3.37359434
```

```
#Vsualisation
```

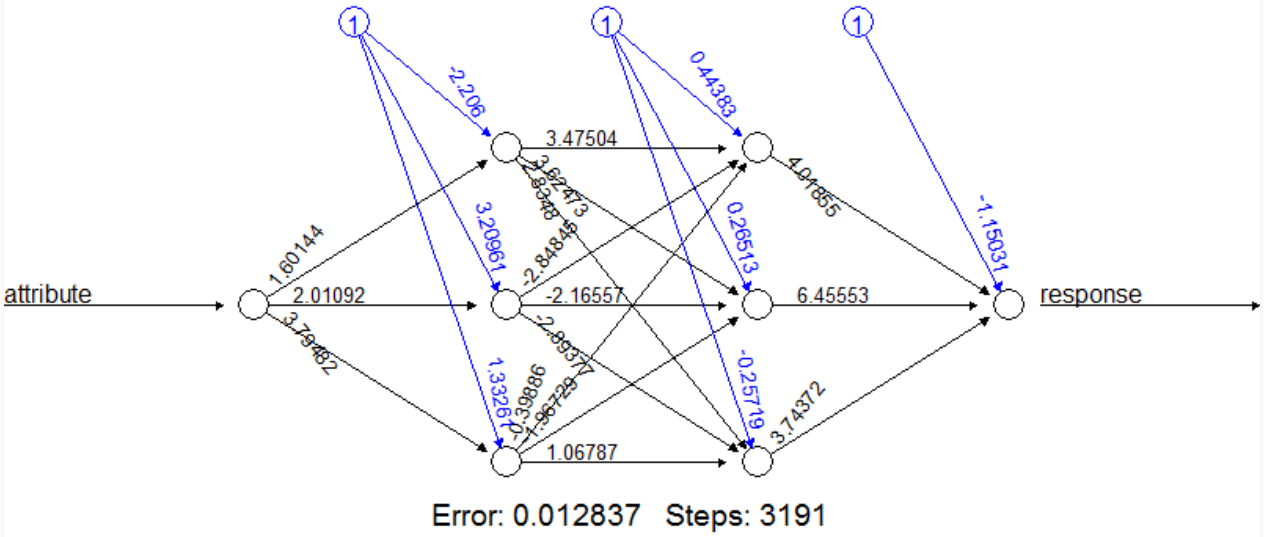
```r
plot(data, pch=20, col=2)

# draw lines
x <- data$attribute[order(data$attribute)]
y <- data$response[order(data$attribute)]
lines(x, y, col=8, lty=3, lwd=2)
```




```r
#-------------------------------------------------------------
# Neural net with 2 hidden layers and 3 nodes in each layer
#-------------------------------------------------------------

fit <- neuralnet(response~attribute, data=data, hidden=c(3, 3),
                 threshold =0.01, act.fct="logistic")

ls(fit)
```

```
##  [1] "act.fct"             "call"                "covariate"
##  [4] "data"                "err.fct"             "generalized.weights"
##  [7] "linear.output"       "model.list"          "net.result"
## [10] "response"            "result.matrix"       "startweights"
## [13] "weights"
```

```r
#Fitted net


plot(fit)
```




```
#------------------------------------------------------------------
# Testing the net with ten new inputs
#------------------------------------------------------------------

testdata <- as.matrix(sample(seq(-2, 2, length =10), 10, replace=FALSE), ncol
=1)

#Prediction in the neuralnet with compute function (gives the values for each
neuron)

pred <- compute(fit, testdata)

ls(pred)

## [1] "net.result" "neurons"

pred$net.result

##                [,1]
##  [1,] 3.93170046232
##  [2,] 3.96749483140
##  [3,] 0.43947303213
##  [4,] 0.45539845038
##  [5,] 2.45206397261
##  [6,] 2.42129930705
##  [7,] 0.03639675054
##  [8,] 0.07853868946
##  [9,] 1.22536423634
## [10,] 1.20132207689

pred$neurons
```

```
## [[1]]
##       [,1]          [,2]
##  [1,]    1  2.0000000000
##  [2,]    1 -2.0000000000
##  [3,]    1  0.6666666667
##  [4,]    1 -0.6666666667
##  [5,]    1  1.5555555556
##  [6,]    1 -1.5555555556
##  [7,]    1 -0.2222222222
##  [8,]    1  0.2222222222
##  [9,]    1 -1.1111111111
## [10,]    1  1.1111111111
##
## [[2]]
##       [,1]           [,2]         [,3]           [,4]
##  [1,]    1 0.730442467500 0.9992770491 0.999866628253
##  [2,]    1 0.004456732604 0.3074171899 0.001913271961
##  [3,]    1 0.242618198427 0.9895456043 0.979416814399
##  [4,]    1 0.036487153406 0.8663425995 0.231960998792
##  [5,]    1 0.570800910039 0.9982347658 0.999280081359
##  [6,]    1 0.009039026669 0.5203684303 0.010247302193
##  [7,]    1 0.071632648732 0.9406289227 0.619944678785
##  [8,]    1 0.135857926038 0.9748268624 0.898064177594
##  [9,]    1 0.018246363126 0.7261666315 0.052957415344
## [10,]    1 0.394931475490 0.9956962873 0.996123997806
##
## [[3]]
##       [,1]          [,2]          [,3]          [,4]
##  [1,]    1 0.43461224878 0.22824504635 0.49738055379
##  [2,]    1 0.39720850177 0.40414719855 0.24377245439
##  [3,]    1 0.12759543032 0.05092305916 0.19988106585
##  [4,]    1 0.12029574553 0.12618985737 0.08218337975
##  [5,]    1 0.30690793942 0.14264489773 0.38683412758
##  [6,]    1 0.26676621895 0.29960886898 0.15104243272
##  [7,]    1 0.09675136604 0.06112096518 0.10773735027
##  [8,]    1 0.09805627810 0.04228035764 0.15008134229
##  [9,]    1 0.17046464965 0.20661023606 0.09532616439
## [10,]    1 0.19508862748 0.08171523877 0.27783662063
```

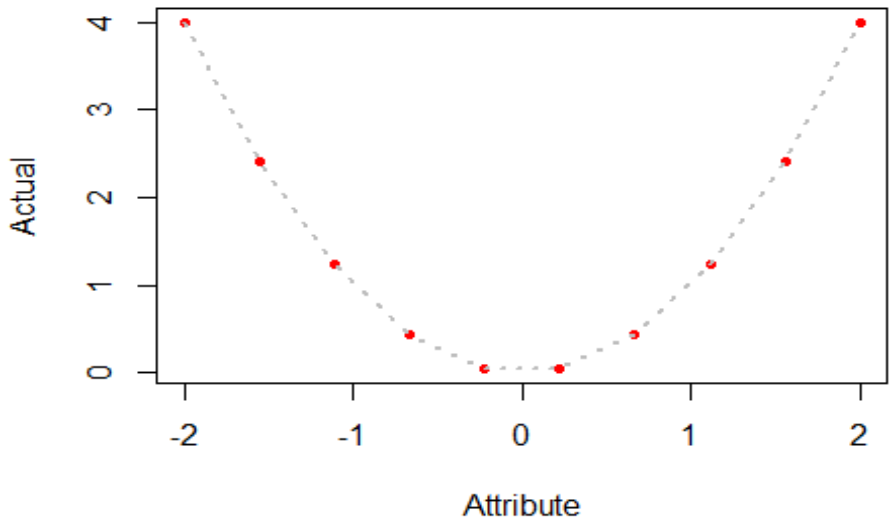
```
cor(pred$net.result, testdata^2)
```

```
##              [,1]
## [1,] 0.9998552216
```

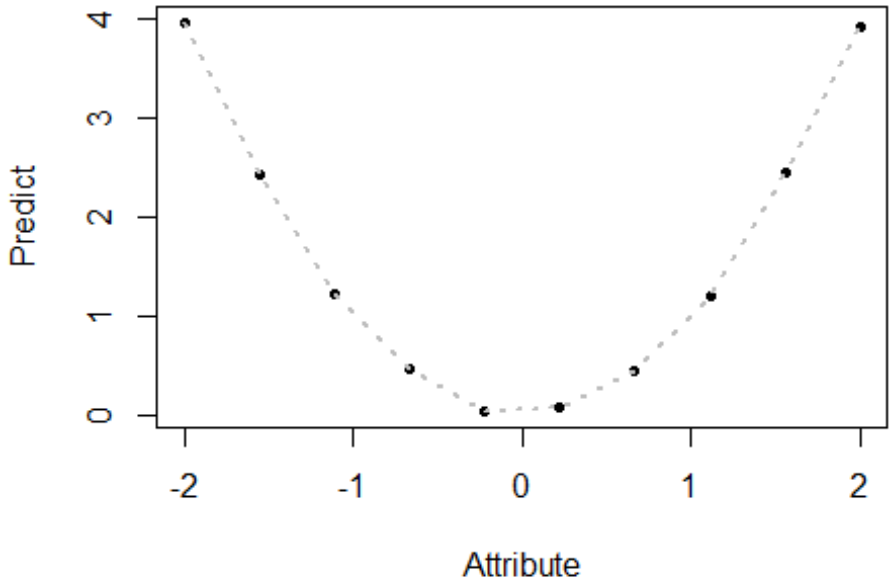
```
#-------------------------------------------------------
#Visualization of the predictions
#-------------------------------------------------------

result <- cbind(testdata, pred$net.result, testdata^2)
colnames(result) <- c('Attribute', 'Prediction', 'Actual')
round(result, 4)
```

```
##       Attribute Prediction Actual
##  [1,]    2.0000     3.9317 4.0000
##  [2,]   -2.0000     3.9675 4.0000
##  [3,]    0.6667     0.4395 0.4444
##  [4,]   -0.6667     0.4554 0.4444
##  [5,]    1.5556     2.4521 2.4198
##  [6,]   -1.5556     2.4213 2.4198
##  [7,]   -0.2222     0.0364 0.0494
##  [8,]    0.2222     0.0785 0.0494
##  [9,]   -1.1111     1.2254 1.2346
## [10,]    1.1111     1.2013 1.2346
```

```
x <- result[,"Attribute"][order(result[,"Attribute"])]
y_act <- result[,"Actual"][order(result[,"Attribute"])]
y_pred <- result[,"Prediction"][order(result[,"Attribute"])]

#par(mfrow=c(1,2))

# Actual data
plot(x, y_act, pch=20, col=2, xlab='Attribute', ylab="Actual")
lines(x, y_act, col=8, lty=3, lwd=2)
```




```
# Predictions
plot(x, y_pred, pch=20, col=1, xlab='Attribute', ylab="Predict")
lines(x, y_pred, col=8, lty=3, lwd=2)
```



```r
#=========================================================================
# Classification with neural nets using Iris data
#=========================================================================
dim(iris)
```

```
## [1] 150   5
```

```r
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```r
#-------------------------------------------------------------------------
#Splitting in training and test data
#-------------------------------------------------------------------------

indexesTrain <- sample(1:150, 70)
indexesValidate <- setdiff(1:150, indexesTrain)

train <- iris[indexesTrain,]

# classes labeling
train$setosa = c(train$Species == "setosa")
train$versicolor = c(train$Species == "versicolor")
train$virginica = c(train$Species == "virginica")

#-------------------------------------------------------------------------
# Classification with a neural net with 2 hidden layers with 3 and 4 nodes
#-------------------------------------------------------------------------
-
hidd = c(3, 4)

fit1 <- neuralnet(setosa + versicolor + virginica ~
           Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, train,
           hidden=hidd, lifesign="full")
```

```
## hidden: 3, 4    thresh: 0.01    rep: 1/1    steps:    1000   min thresh: 0
.104268603
##                                                     1903   error: 0.0035
3  time: 0.45 secs
```

```r
#-------------------------------------------------------------------------
-
#Validation of the network
#-------------------------------------------------------------------------
-
```

```r
predict1 <- compute(fit1, iris[1:4])

head(predict1$net.result)
```

```
##              [,1]             [,2]            [,3]
## [1,] 1.0035739828  0.0006833288176 -0.004920109234
## [2,] 0.9990292622 -0.0007082486559 -0.000943893136
## [3,] 1.0028991305 -0.0046127354590  0.003043297021
## [4,] 0.9984722182 -0.0041688613856  0.009163250287
## [5,] 1.0042655060 -0.0005405407394 -0.002950816614
## [6,] 1.0016978836  0.0022112597009 -0.005037442371
```

```r
maxidx = function(arr) {
  return(which(arr == max(arr)))
}

idx1<-apply(predict1$net.result, 1, maxidx)
prediction1<-c("setosa", "versicolor", "virginica")[idx1]
table(prediction1,iris$Species)
```

```
##
## prediction1  setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         49         2
##   virginica       0          1        48
```