Math642_HW10_FyonaSun

Fyona Sun

4/7/2020

Problem 1

Using the R code posted as an example, create a neural network that learns how to multiply 2 numbers from 1 to 10 together. Try different numbers of nodes per layer and number of hidden layers. Plot the best neural network and describe the model. Describe how accurate the results are in each case.

```
library(neuralnet)
#Generate 50 random numbers uniformly distributed between 0 and 10
#And store them as a dataframe
set.seed(1)
traininginput <-  data.frame(x1=runif(50, min=0, max=10), x2=runif(50, min=0, 
max=10))
trainingoutput <- traininginput$x1*traininginput$x2

#Column bind the data into one variable
trainingdata <- cbind(traininginput,trainingoutput)
head(trainingdata)
```

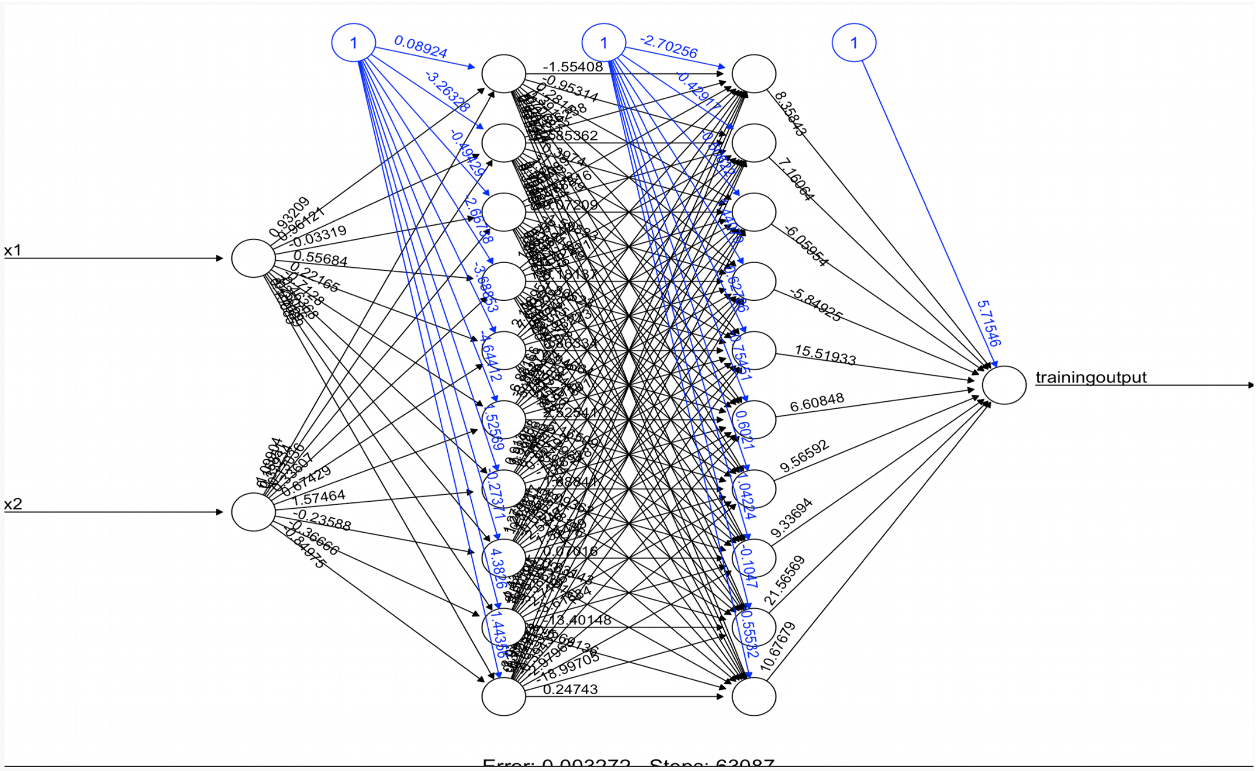
```
##         x1        x2 trainingoutput
## 1 2.655087 4.7761962      12.681215
## 2 3.721239 8.6120948      32.047663
## 3 5.728534 4.3809711      25.096540
## 4 9.082078 2.4479728      22.232679
## 5 2.016819 0.7067905       1.425469
## 6 8.983897 0.9946616       8.935937
```

```
net.multi <- neuralnet(trainingoutput~x1+x2,data=trainingdata, hidden=c(10,10
), threshold=0.01,stepmax = 1e16)
print(net.multi)
```

```
## $call
## neuralnet(formula = trainingoutput ~ x1 + x2, data = trainingdata, 
##     hidden = c(10, 10), threshold = 0.01, stepmax = 1e+16)
## 
## $response
##    trainingoutput
## 1       12.681215
## 2       32.047663
## 3       25.096540
## 4       22.232679
## 5        1.425469
## 6        8.935937
## 7       29.877406
## 8       34.271238
## 9       41.647669
## 10       2.513652
## 11      18.802923
## 12       5.183766
## 13      31.538864
## 14      12.767403
## 15      50.106704
## 16      12.841476
## 17      34.341293
## 18      76.010822
## 19       3.201679
## 20      68.051439
## 21      31.693325
## 22      17.808099
## 23      22.592454
## 24       4.190714
## 25      12.729090
## 26      34.449035
## 27       1.157379
## 28      14.912730
## 29      67.602870
## 30      32.694537
## 31      20.954069
## 32      42.719945
## 33      19.741374
## 34       6.058630
## 35      62.639371
## 36      13.549303
## 37      56.480082
## 38       1.313587
## 39      17.766272
## 40       5.893743
## 41      19.672288
## 42       3.813409
## 43      50.286852
## 44      48.460869
## 45      41.260636
## 46      62.936069
## 47       1.062210
## 48      19.570445
## 49      59.381142
## 50      41.905638
## 
## $covariate
##               x1        x2
##  [1,] 2.6550866 4.7761962
##  [2,] 3.7212390 8.6120948
##  [3,] 5.7285336 4.3809711
##  [4,] 9.0820779 2.4479728
##  [5,] 2.0168193 0.7067905
##  [6,] 8.9838968 0.9946616
##  [7,] 9.4467527 3.1627171
##  [8,] 6.6079779 5.1863426
##  [9,] 6.2911404 6.6200508
## [10,] 0.6178627 4.0683019
## [11,] 2.0597457 9.1287592
## [12,] 1.7655675 2.9360337
## [13,] 6.8702285 4.5906573
## [14,] 3.8410372 3.3239467
## [15,] 7.6984142 6.5087047
## [16,] 4.9769924 2.5801678
## [17,] 7.1761851 4.7854525
## [18,] 9.9190609 7.6631067
## [19,] 3.8003518 0.8424691
## [20,] 7.7744522 8.7532133
## [21,] 9.3470523 3.3907294
## [22,] 2.1214252 8.3944035
## [23,] 6.5167377 3.4668349
## [24,] 1.2555510 3.3377493
## [25,] 2.6722067 4.7635125
## [26,] 3.8611409 8.9219834
## [27,] 0.1339033 8.6433947
## [28,] 3.8238796 3.8998954
## [29,] 8.6969085 7.7732070
## [30,] 3.4034900 9.6061800
## [31,] 4.8208012 4.3465948
## [32,] 5.9956583 7.1251468
## [33,] 4.9354131 3.9999437
## [34,] 1.8621760 3.2535215
## [35,] 8.2737332 7.5708715
## [36,] 6.6846674 2.0269226
## [37,] 7.9423986 7.1112122
## [38,] 1.0794363 1.2169192
## [39,] 7.2371095 2.4548851
## [40,] 4.1127443 1.4330438
## [41,] 8.2094629 2.3962942
## [42,] 6.4706019 0.5893438
## [43,] 7.8293276 6.4228826
## [44,] 5.5303631 8.7626921
## [45,] 5.2971958 7.7891468
## [46,] 7.8935623 7.9730883
## [47,] 0.2333120 4.5527445
## [48,] 4.7723007 4.1008408
## [49,] 7.3231374 8.1087024
## [50,] 6.9273156 6.0493329
##
## $model.list
## $model.list$response
## [1] "trainingoutput"
## 
## $model.list$variables
## [1] "x1" "x2"
## 
## 
## $err.fct
## function (x, y) 
## {
##     1/2 * (y - x)^2
## }
## <bytecode: 0x7f87234874c8>
## <environment: 0x7f8723488968>
## attr(,"type")
## [1] "sse"
## 
## $act.fct
## function (x) 
## {
##     1/(1 + exp(-x))
## }
## <bytecode: 0x7f87262f1d48>
## <environment: 0x7f87262f44f0>
## attr(,"type")
## [1] "logistic"
## 
## $linear.output
## [1] TRUE
## 
## $data
##           x1        x2 trainingoutput
## 1  2.6550866 4.7761962      12.681215
## 2  3.7212390 8.6120948      32.047663
## 3  5.7285336 4.3809711      25.096540
## 4  9.0820779 2.4479728      22.232679
## 5  2.0168193 0.7067905       1.425469
## 6  8.9838968 0.9946616       8.935937
## 7  9.4467527 3.1627171      29.877406
## 8  6.6079779 5.1863426      34.271238
## 9  6.2911404 6.6200508      41.647669
## 10 0.6178627 4.0683019       2.513652
## 11 2.0597457 9.1287592      18.802923
## 12 1.7655675 2.9360337       5.183766
## 13 6.8702285 4.5906573      31.538864
## 14 3.8410372 3.3239467      12.767403
## 15 7.6984142 6.5087047      50.106704
## 16 4.9769924 2.5801678      12.841476
## 17 7.1761851 4.7854525      34.341293
## 18 9.9190609 7.6631067      76.010822
## 19 3.8003518 0.8424691        3.201679
## 20 7.7744522 8.7532133       68.051439
## 21 9.3470523 3.3907294       31.693325
## 22 2.1214252 8.3944035       17.808099
## 23 6.5167377 3.4668349       22.592454
## 24 1.2555510 3.3377493        4.190714
## 25 2.6722067 4.7635125       12.729090
## 26 3.8611409 8.9219834       34.449035
## 27 0.1339033 8.6433947        1.157379
## 28 3.8238796 3.8998954       14.912730
## 29 8.6969085 7.7732070       67.602870
## 30 3.4034900 9.6061800       32.694537
## 31 4.8208012 4.3465948       20.954069
## 32 5.9956583 7.1251468       42.719945
## 33 4.9354131 3.9999437       19.741374
## 34 1.8621760 3.2535215        6.058630
## 35 8.2737332 7.5708715       62.639371
## 36 6.6846674 2.0269226       13.549303
## 37 7.9423986 7.1112122       56.480082
## 38 1.0794363 1.2169192        1.313587
## 39 7.2371095 2.4548851       17.766272
## 40 4.1127443 1.4330438        5.893743
## 41 8.2094629 2.3962942       19.672288
## 42 6.4706019 0.5893438        3.813409
## 43 7.8293276 6.4228826       50.286852
## 44 5.5303631 8.7626921       48.460869
## 45 5.2971958 7.7891468       41.260636
## 46 7.8935623 7.9730883       62.936069
## 47 0.2333120 4.5527445        1.062210
## 48 4.7723007 4.1008408       19.570445
## 49 7.3231374 8.1087024       59.381142
## 50 6.9273156 6.0493329       41.905638
##
## $exclude
## NULL
##
## $net.result
## $net.result[[1]]
##            [,1]
##  [1,] 12.677509
##  [2,] 32.041960
##  [3,] 25.105071
##  [4,] 22.234862
##  [5,]  1.437681
##  [6,]  8.935643
##  [7,] 29.873445
##  [8,] 34.272699
##  [9,] 41.661421
## [10,]  2.507614
## [11,] 18.800990
## [12,]  5.202638
## [13,] 31.533856
## [14,] 12.746552
## [15,] 50.091394
## [16,] 12.846744
## [17,] 34.343091
## [18,] 76.010455
## [19,]  3.191216
## [20,] 68.049453
## [21,] 31.696108
## [22,] 17.811130
## [23,] 22.592204
## [24,]  4.201577
## [25,] 12.729411
## [26,] 34.452338
## [27,]  1.157485
## [28,] 14.939017
## [29,] 67.609544
## [30,] 32.695650
## [31,] 20.942900
## [32,] 42.705261
## [33,] 19.746188
## [34,]  6.036272
## [35,] 62.617290
## [36,] 13.545628
## [37,] 56.494479
## [38,]  1.301114
## [39,] 17.768024
## [40,]  5.900384
## [41,] 19.671112
## [42,]  3.815189
## [43,] 50.296796
## [44,] 48.460632
## [45,] 41.265970
## [46,] 62.946675
## [47,]  1.065884
## [48,] 19.562849
## [49,] 59.379147
## [50,] 41.901874
## 
## 
## $weights
## $weights[[1]]
## $weights[[1]][[1]]
##             [,1]       [,2]       [,3]       [,4]        [,5]       [,6]
## [1,]  3.7058464 -3.8726087 -0.7094619  2.1328045  3.90475826 -2.3981718
## [2,] -0.8121451  0.2974382 -0.1036156  1.7664235 -0.06502322 -0.6968875
## [3,]  0.1880711  0.4147811  0.4283282 -0.4471493 -0.48832626  1.2441251
##             [,7]       [,8]       [,9]      [,10]
## [1,] -2.4635835 -1.6013873 -0.8668011  1.11003980
## [2,]  0.2006309  1.1989368  0.2693704  0.07978545
## [3,]  0.4713748 -0.2940514 -1.0492305 -0.56572061
## 
## $weights[[1]][[2]]
##              [,1]        [,2]        [,3]         [,4]         [,5]       [
,6]
##  [1,] -0.9615342   0.6924705 -1.53893359 -0.121441587    0.0605988 -1.1863
790
##  [2,] -2.6176280  -5.3065824 -2.00689382 -2.219810056   -3.5193987 -4.4407
817
##  [3,]  3.1038306   2.9953922 -0.03595046  0.001099838    1.3209328  2.2686
169
##  [4,] -1.2855918   1.7988671  1.29024627  0.208235397    0.3141163  0.5587
601
##  [5,]  2.2403647   0.9147854  5.07818141  1.259452608    0.6678002  1.6734
567
##  [6,] -1.7889529  -6.2378807 -1.30477203 -2.763166458   -7.5932956 -9.1315
092
##  [7,]  1.7652559  -1.4261132 -1.68477629 -1.693317703    0.2956615  0.8085
698
##  [8,] -0.6970380  -1.7373463  0.60212889  0.021676841   -0.4413481 -1.3148
893
##  [9,]  1.4053151   0.3772688  1.61893759  3.093305160    3.4882757  1.2670
231
## [10,] -2.5058745 -30.4377830 -4.11509946 -2.659488998 -101.5853507 13.6249
820
## [11,] -3.3100273  -1.1829249 -2.92140425 -3.384839519   -1.6378489 -6.8158
828
##               [,7]       [,8]        [,9]       [,10]
##  [1,]   2.8311663 -1.4117078 -0.96579070   1.1390548
##  [2,]  -2.3957809 -4.2850710 -3.09471167 -12.8710567
##  [3,]  -1.3796190  3.6834309  5.22673545   2.1464851
##  [4,]  -0.8967128  0.5614032  0.01928677   0.1761856
##  [5,]   3.8140983  2.7944794  1.24609093  -0.9495265
##  [6,]  -4.6217867  0.1249645 -3.60997112  -2.5790619
##  [7,]   0.4044224  0.9789358  0.04605120  -2.9094045
##  [8,]  -0.3404017 -1.4099237 -0.62890351   1.3959018
##  [9,]   0.8516823  2.2356151  1.22884940  -0.4354296
## [10,] -26.1422991 -6.9695635  1.16004431   1.5755883
## [11,]  -2.8453588 -2.8683555 -3.37986531  -0.5912580
## 
## $weights[[1]][[3]]
##            [,1]
##  [1,] -1.133436
##  [2,]  9.831442
##  [3,]  9.242852
##  [4,]  9.948868
##  [5,]  6.281922
##  [6,] 12.894776
##  [7,] 19.134317
##  [8,]  5.841806
##  [9,]  7.587927
## [10,]  7.939912
## [11,] 15.394572
## 
## 
## 
## $generalized.weights
## $generalized.weights[[1]]
##               [,1]         [,2]
##  [1,]  -0.033780294 -0.017855079
##  [2,]  -0.008724807 -0.003805083
##  [3,]  -0.007324742 -0.009204953
##  [4,]  -0.004813561 -0.020775188
##  [5,]  -1.378174574 -3.479256374
##  [6,]  -0.023812760 -0.068813668
##  [7,]  -0.003401838 -0.010692140
##  [8,]  -0.004505276 -0.005835930
##  [9,]  -0.004034328 -0.003743663
## [10,]  -1.105211121 -0.122143070
## [11,]  -0.033578090 -0.006147455
## [12,]  -0.137499091 -0.073195536
## [13,]  -0.004784229 -0.007091517
## [14,]  -0.021449717 -0.025875894
## [15,]  -0.002715219 -0.003189579
## [16,]  -0.018249325 -0.032641218
## [17,]  -0.004292984 -0.006164744
## [18,]  -0.001163719 -0.001325312
## [19,]  -0.099740590 -0.540071680
## [20,]  -0.001913142 -0.001596568
## [21,]  -0.003205386 -0.009738885
## [22,]  -0.031017704 -0.007698044
## [23,]  -0.006471406 -0.013533344
## [24,]  -0.240215528 -0.095384645
## [25,]  -0.033444450 -0.017834804
## [26,]  -0.007812031 -0.003279045
## [27,] -11.700489063 -2.972126416
## [28,]  -0.018285799 -0.018980094
## [29,]  -0.001756904 -0.001852796
## [30,]  -0.008602935 -0.003052098
## [31,]  -0.010302109 -0.011536608
## [32,]  -0.004140975 -0.003438236
## [33,]  -0.011014443 -0.013151612
## [34,]  -0.108039074 -0.057285502
## [35,]  -0.001999466 -0.002111304
## [36,]  -0.010607618 -0.043083530
## [37,]  -0.002313522 -0.002477089
## [38,]  -3.428352169 -3.342608594
## [39,]  -0.007827604 -0.023765125
## [40,]  -0.042752866 -0.147324978
## [41,]  -0.006676239 -0.023204818
## [42,]  -0.068160800 -0.395512968
## [43,]  -0.002753102 -0.003205784
## [44,]  -0.003625349 -0.002259598
## [45,]  -0.004719782 -0.003344009
## [46,]  -0.002038431 -0.002073268
## [47,] -50.173393646 -0.228539513
## [48,]  -0.011343003 -0.013090232
## [49,]  -0.002368566 -0.002096193
## [50,]  -0.003418203 -0.004086458
## 
## 
## $startweights
## $startweights[[1]]
## $startweights[[1]][[1]]
##           [,1]      [,2]       [,3]       [,4]        [,5]       [,6]
## [1,]  0.3981059 -1.129363 -0.3672215 -0.1350546  0.68973936  0.1887923
## [2,] -0.6120264  1.433024 -1.0441346  2.4016178  0.02800216 -1.8049586
## [3,]  0.3411197  1.980400  0.5697196 -0.0392400 -0.74327321  1.4655549
##          [,7]       [,8]       [,9]        [,10]
## [1,] 0.1532533 -0.7099464 -1.2536334  0.001105352
## [2,] 2.1726117  0.6107264  0.2914462  0.074341324
## [3,] 0.4755095 -0.9340976 -0.4432919 -0.589520946
## 
## $startweights[[1]][[2]]
##             [,1]        [,2]       [,3]       [,4]        [,5]        [,6]
##  [1,] -0.5686687  1.20786781 -0.9109216 -0.6506964 -0.10019074 -1.53644982
##  [2,] -0.1351786  1.16040262  0.1580288 -0.2073807  0.71266631 -0.30097613
##  [3,]  1.1780870  0.70021365 -0.6545846 -0.3928079 -0.07356440 -0.52827990
##  [4,] -1.5235668  1.58683345  1.7672873 -0.3199929 -0.03763417 -0.65209478
##  [5,]  0.5939462  0.55848643  0.7167075 -0.2791133 -0.68166048 -0.05689678
##  [6,]  0.3329504 -1.27659221  0.9101742  0.4941883 -0.32427027 -1.91435943
##  [7,]  1.0630998 -0.57326541  0.3841854 -0.1773305  0.06016044  1.17658331
##  [8,] -0.3041839 -1.22461261  1.6821761 -0.5059575 -0.58889449 -1.66497244
##  [9,]  0.3700188 -0.47340064 -0.6357365  1.3430388  0.53149619 -0.46353040
## [10,]  0.2670988 -0.62036668 -0.4616447 -0.2145794 -1.51839408 -1.11592011
## [11,] -0.5425200  0.04211587  1.4322822 -0.1795565  0.30655786 -0.75081900
##              [,7]       [,8]        [,9]      [,10]
##  [1,]  2.08716655 -0.6212667 -0.14439960  1.2079084
##  [2,]  0.01739562 -1.3844268  0.20753834 -1.2313234
##  [3,] -1.28630053  1.8692906  2.30797840  0.9838956
##  [4,] -1.64060553  0.4251004  0.10580237  0.2199248
##  [5,]  0.45018710 -0.2386471  0.45699881 -1.4672500
##  [6,] -0.01855983  1.0584830 -0.07715294  0.5210227
##  [7,] -0.31806837  0.8864227 -0.33400084 -0.1587546
##  [8,] -0.92936215 -0.6192430 -0.03472603  1.4645873
##  [9,] -1.48746031  2.2061025  0.78763961 -0.7660820
## [10,] -1.07519230 -0.2550270  2.07524501 -0.4302118
## [11,]  1.00002880 -1.4244947  1.02739244 -0.9261095
## 
```

```
## $startweights[[1]][[3]]
##             [,1]
##  [1,] -0.1771040
##  [2,]  0.4020118
##  [3,] -0.7317482
##  [4,]  0.8303732
##  [5,] -1.2080828
##  [6,] -1.0479844
##  [7,]  1.4411577
##  [8,] -1.0158475
##  [9,]  0.4119747
## [10,] -0.3810761
## [11,]  0.4094018
## 
## 
## 
## $result.matrix
##                                      [,1]
## error                        2.353088e-03
## reached.threshold            9.762037e-03
## steps                        1.452060e+05
## Intercept.to.1layhid1        3.705846e+00
## x1.to.1layhid1              -8.121451e-01
## x2.to.1layhid1               1.880711e-01
## Intercept.to.1layhid2       -3.872609e+00
## x1.to.1layhid2               2.974382e-01
## x2.to.1layhid2               4.147811e-01
## Intercept.to.1layhid3       -7.094619e-01
## x1.to.1layhid3              -1.036156e-01
## x2.to.1layhid3               4.283282e-01
## Intercept.to.1layhid4        2.132804e+00
## x1.to.1layhid4               1.766423e+00
## x2.to.1layhid4              -4.471493e-01
## Intercept.to.1layhid5        3.904758e+00
## x1.to.1layhid5              -6.502322e-02
## x2.to.1layhid5              -4.883263e-01
## Intercept.to.1layhid6       -2.398172e+00
## x1.to.1layhid6              -6.968875e-01
## x2.to.1layhid6               1.244125e+00
## Intercept.to.1layhid7       -2.463583e+00
## x1.to.1layhid7               2.006309e-01
## x2.to.1layhid7               4.713748e-01
## Intercept.to.1layhid8       -1.601387e+00
## x1.to.1layhid8               1.198937e+00
## x2.to.1layhid8              -2.940514e-01
## Intercept.to.1layhid9       -8.668011e-01
## x1.to.1layhid9               2.693704e-01
## x2.to.1layhid9              -1.049230e+00
## Intercept.to.1layhid10       1.110040e+00
## x1.to.1layhid10              7.978545e-02
## x2.to.1layhid10            -5.657206e-01
## Intercept.to.2layhid1      -9.615342e-01
## 1layhid1.to.2layhid1       -2.617628e+00
## 1layhid2.to.2layhid1        3.103831e+00
## 1layhid3.to.2layhid1       -1.285592e+00
## 1layhid4.to.2layhid1        2.240365e+00
## 1layhid5.to.2layhid1       -1.788953e+00
## 1layhid6.to.2layhid1        1.765256e+00
## 1layhid7.to.2layhid1       -6.970380e-01
## 1layhid8.to.2layhid1        1.405315e+00
## 1layhid9.to.2layhid1       -2.505875e+00
## 1layhid10.to.2layhid1      -3.310027e+00
## Intercept.to.2layhid2       6.924705e-01
## 1layhid1.to.2layhid2       -5.306582e+00
## 1layhid2.to.2layhid2        2.995392e+00
## 1layhid3.to.2layhid2        1.798867e+00
## 1layhid4.to.2layhid2        9.147854e-01
## 1layhid5.to.2layhid2       -6.237881e+00
## 1layhid6.to.2layhid2       -1.426113e+00
## 1layhid7.to.2layhid2       -1.737346e+00
## 1layhid8.to.2layhid2        3.772688e-01
## 1layhid9.to.2layhid2       -3.043778e+01
## 1layhid10.to.2layhid2      -1.182925e+00
## Intercept.to.2layhid3      -1.538934e+00
## 1layhid1.to.2layhid3       -2.006894e+00
## 1layhid2.to.2layhid3       -3.595046e-02
## 1layhid3.to.2layhid3        1.290246e+00
## 1layhid4.to.2layhid3        5.078181e+00
## 1layhid5.to.2layhid3       -1.304772e+00
## 1layhid6.to.2layhid3       -1.684776e+00
## 1layhid7.to.2layhid3        6.021289e-01
## 1layhid8.to.2layhid3        1.618938e+00
## 1layhid9.to.2layhid3       -4.115099e+00
## 1layhid10.to.2layhid3      -2.921404e+00
## Intercept.to.2layhid4      -1.214416e-01
## 1layhid1.to.2layhid4       -2.219810e+00
## 1layhid2.to.2layhid4        1.099838e-03
## 1layhid3.to.2layhid4        2.082354e-01
## 1layhid4.to.2layhid4        1.259453e+00
## 1layhid5.to.2layhid4       -2.763166e+00
## 1layhid6.to.2layhid4       -1.693318e+00
## 1layhid7.to.2layhid4        2.167684e-02
## 1layhid8.to.2layhid4        3.093305e+00
## 1layhid9.to.2layhid4       -2.659489e+00
## 1layhid10.to.2layhid4      -3.384840e+00
## Intercept.to.2layhid5       6.059880e-02
## 1layhid1.to.2layhid5       -3.519399e+00
## 1layhid2.to.2layhid5        1.320933e+00
## 1layhid3.to.2layhid5        3.141163e-01
## 1layhid4.to.2layhid5        6.678002e-01
## 1layhid5.to.2layhid5        -7.593296e+00
## 1layhid6.to.2layhid5         2.956615e-01
## 1layhid7.to.2layhid5        -4.413481e-01
## 1layhid8.to.2layhid5         3.488276e+00
## 1layhid9.to.2layhid5        -1.015854e+02
## 1layhid10.to.2layhid5       -1.637849e+00
## Intercept.to.2layhid6       -1.186379e+00
## 1layhid1.to.2layhid6        -4.440782e+00
## 1layhid2.to.2layhid6         2.268617e+00
## 1layhid3.to.2layhid6         5.587601e-01
## 1layhid4.to.2layhid6         1.673457e+00
## 1layhid5.to.2layhid6        -9.131509e+00
## 1layhid6.to.2layhid6         8.085698e-01
## 1layhid7.to.2layhid6        -1.314889e+00
## 1layhid8.to.2layhid6         1.267023e+00
## 1layhid9.to.2layhid6         1.362498e+01
## 1layhid10.to.2layhid6       -6.815883e+00
## Intercept.to.2layhid7        2.831166e+00
## 1layhid1.to.2layhid7        -2.395781e+00
## 1layhid2.to.2layhid7        -1.379619e+00
## 1layhid3.to.2layhid7        -8.967128e-01
## 1layhid4.to.2layhid7         3.814098e+00
## 1layhid5.to.2layhid7        -4.621787e+00
## 1layhid6.to.2layhid7         4.044224e-01
## 1layhid7.to.2layhid7        -3.404017e-01
## 1layhid8.to.2layhid7         8.516823e-01
## 1layhid9.to.2layhid7        -2.614230e+01
## 1layhid10.to.2layhid7       -2.845359e+00
## Intercept.to.2layhid8       -1.411708e+00
## 1layhid1.to.2layhid8        -4.285071e+00
## 1layhid2.to.2layhid8         3.683431e+00
## 1layhid3.to.2layhid8         5.614032e-01
## 1layhid4.to.2layhid8         2.794479e+00
## 1layhid5.to.2layhid8         1.249645e-01
## 1layhid6.to.2layhid8         9.789358e-01
## 1layhid7.to.2layhid8        -1.409924e+00
## 1layhid8.to.2layhid8         2.235615e+00
## 1layhid9.to.2layhid8        -6.969564e+00
## 1layhid10.to.2layhid8       -2.868356e+00
## Intercept.to.2layhid9       -9.657907e-01
## 1layhid1.to.2layhid9        -3.094712e+00
## 1layhid2.to.2layhid9         5.226735e+00
## 1layhid3.to.2layhid9         1.928677e-02
## 1layhid4.to.2layhid9         1.246091e+00
## 1layhid5.to.2layhid9        -3.609971e+00
## 1layhid6.to.2layhid9         4.605120e-02
## 1layhid7.to.2layhid9        -6.289035e-01
## 1layhid8.to.2layhid9         1.228849e+00
## 1layhid9.to.2layhid9         1.160044e+00
## 1layhid10.to.2layhid9       -3.379865e+00
## Intercept.to.2layhid10       1.139055e+00
## 1layhid1.to.2layhid10       -1.287106e+01
## 1layhid2.to.2layhid10        2.146485e+00
## 1layhid3.to.2layhid10        1.761856e-01
## 1layhid4.to.2layhid10       -9.495265e-01
## 1layhid5.to.2layhid10       -2.579062e+00
## 1layhid6.to.2layhid10       -2.909404e+00
## 1layhid7.to.2layhid10        1.395902e+00
## 1layhid8.to.2layhid10       -4.354296e-01
## 1layhid9.to.2layhid10        1.575588e+00
## 1layhid10.to.2layhid10      -5.912580e-01
## Intercept.to.trainingoutput -1.133436e+00
## 2layhid1.to.trainingoutput   9.831442e+00
## 2layhid2.to.trainingoutput   9.242852e+00
## 2layhid3.to.trainingoutput   9.948868e+00
## 2layhid4.to.trainingoutput   6.281922e+00
## 2layhid5.to.trainingoutput   1.289478e+01
## 2layhid6.to.trainingoutput   1.913432e+01
## 2layhid7.to.trainingoutput   5.841806e+00
## 2layhid8.to.trainingoutput   7.587927e+00
## 2layhid9.to.trainingoutput   7.939912e+00
## 2layhid10.to.trainingoutput  1.539457e+01
##
## attr(,"class")
## [1] "nn"
```

```
#Plot the neural network
plot(net.multi)
```



```r
#Test the neural network on some training data
set.seed(1)
testdata <- data.frame(x1=runif(20, min=0, max=10), x2=runif(20, min=0, max=1
0))
net.results <- predict(net.multi, testdata) #Run them through the neural netw
ork

#Lets see the results
print(net.results)

##            [,1]
##  [1,] 25.306635
##  [2,]  7.983956
##  [3,] 37.197326
##  [4,] 10.634707
##  [5,]  5.489751
##  [6,] 35.124971
##  [7,]  8.947969
##  [8,] 25.277294
##  [9,] 54.444994
## [10,]  2.141611
## [11,]  9.889186
## [12,] 10.520162
## [13,] 33.889668
## [14,]  7.239040
## [15,] 63.747313
## [16,] 33.061690
## [17,] 57.003164
## [18,] 11.052891
## [19,] 27.404187
## [20,] 32.181841
```

```
#Lets display a better version of the results
cleanoutput <- cbind(testdata,testdata$x1*testdata$x2,
                     as.data.frame(net.results))
colnames(cleanoutput) <- c("Input-x1","Input-x2","ExpectedOutput","NeuralNetO
utput")
print(cleanoutput)
```

```
##     Input-x1  Input-x2 ExpectedOutput NeuralNetOutput
## 1  2.6550866 9.3470523      24.817234       25.306635
## 2  3.7212390 2.1214252       7.894330        7.983956
## 3  5.7285336 6.5167377      37.331351       37.197326
## 4  9.0820779 1.2555510      11.403012       10.634707
## 5  2.0168193 2.6722067       5.389358        5.489751
## 6  8.9838968 3.8611409      34.688092       35.124971
## 7  9.4467527 0.1339033       1.264952        8.947969
## 8  6.6079779 3.8238796      25.268112       25.277294
## 9  6.2911404 8.6969085      54.713472       54.444994
## 10 0.6178627 3.4034900       2.102890        2.141611
## 11 2.0597457 4.8208012       9.929625        9.889186
## 12 1.7655675 5.9956583      10.585740       10.520162
## 13 6.8702285 4.9354131      33.907415       33.889668
## 14 3.8410372 1.8621760       7.152687        7.239040
## 15 7.6984142 8.2737332      63.694625       63.747313
## 16 4.9769924 6.6846674      33.269539       33.061690
## 17 7.1761851 7.9423986      56.996122       57.003164
## 18 9.9190609 1.0794363      10.706994       11.052891
## 19 3.8003518 7.2371095      27.503562       27.404187
## 20 7.7744522 4.1127443      31.974334       32.181841
```

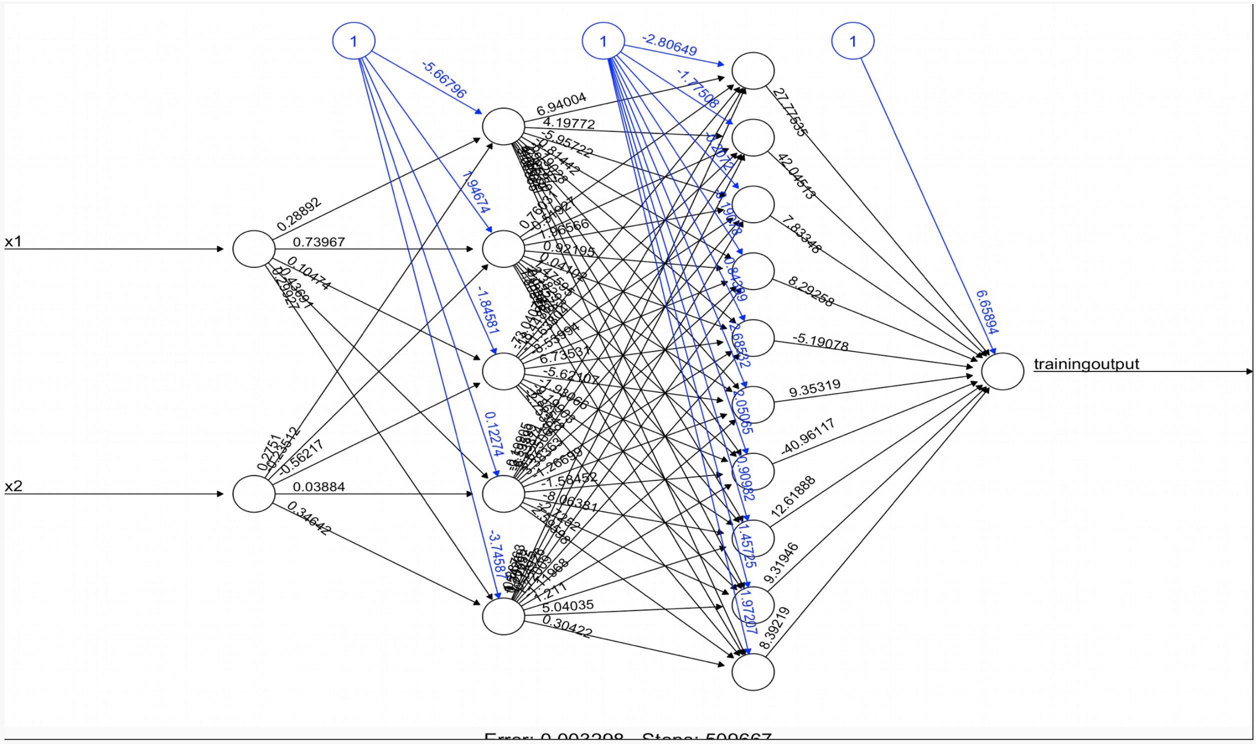
```
(sum((cleanoutput$NeuralNetOutput - cleanoutput$ExpectedOutput)^2) / nrow(cle
anoutput)) ^ 0.5
```

```
## [1] 1.737692
```

```
net.multi <- neuralnet(trainingoutput~x1+x2,data=trainingdata, hidden=c(5,10)
, threshold=0.01,stepmax = 1e16)
print(net.multi)
```

```
## $call
## neuralnet(formula = trainingoutput ~ x1 + x2, data = trainingdata,
##     hidden = c(5, 10), threshold = 0.01, stepmax = 1e+16)
##
## $response
##    trainingoutput
## 1       12.681215
## 2       32.047663
## 3       25.096540
## 4       22.232679
## 5        1.425469
## 6        8.935937
## 7       29.877406
## 8       34.271238
## 9       41.647669
## 10       2.513652
## 11      18.802923
## 12       5.183766
## 13      31.538864
## 14      12.767403
## 15      50.106704
## 16      12.841476
## 17      34.341293
## 18      76.010822
## 19       3.201679
## 20      68.051439
## 21      31.693325
## 22      17.808099
## 23      22.592454
## 24       4.190714
## 25      12.729090
## 26      34.449035
## 27       1.157379
## 28      14.912730
## 29      67.602870
## 30      32.694537
## 31      20.954069
## 32      42.719945
## 33      19.741374
## 34       6.058630
## 35      62.639371
## 36      13.549303
## 37      56.480082
## 38       1.313587
## 39      17.766272
## 40       5.893743
## 41      19.672288
## 42       3.813409
## 43      50.286852
## 44      48.460869
## 45      41.260636
## 46      62.936069
## 47       1.062210
## 48      19.570445
## 49      59.381142
## 50      41.905638
##
## $covariate
##              x1        x2
##  [1,] 2.6550866 4.7761962
##  [2,] 3.7212390 8.6120948
##  [3,] 5.7285336 4.3809711
##  [4,] 9.0820779 2.4479728
##  [5,] 2.0168193 0.7067905
##  [6,] 8.9838968 0.9946616
##  [7,] 9.4467527 3.1627171
##  [8,] 6.6079779 5.1863426
##  [9,] 6.2911404 6.6200508
## [10,] 0.6178627 4.0683019
## [11,] 2.0597457 9.1287592
## [12,] 1.7655675 2.9360337
## [13,] 6.8702285 4.5906573
## [14,] 3.8410372 3.3239467
## [15,] 7.6984142 6.5087047
## [16,] 4.9769924 2.5801678
## [17,] 7.1761851 4.7854525
## [18,] 9.9190609 7.6631067
## [19,] 3.8003518 0.8424691
## [20,] 7.7744522 8.7532133
## [21,] 9.3470523 3.3907294
## [22,] 2.1214252 8.3944035
## [23,] 6.5167377 3.4668349
## [24,] 1.2555510 3.3377493
## [25,] 2.6722067 4.7635125
## [26,] 3.8611409 8.9219834
## [27,] 0.1339033 8.6433947
## [28,] 3.8238796 3.8998954
## [29,] 8.6969085 7.7732070
## [30,] 3.4034900 9.6061800
## [31,] 4.8208012 4.3465948
## [32,] 5.9956583 7.1251468
## [33,] 4.9354131 3.9999437
## [34,] 1.8621760 3.2535215
## [35,] 8.2737332 7.5708715
## [36,] 6.6846674 2.0269226
## [37,] 7.9423986 7.1112122
## [38,] 1.0794363 1.2169192
## [39,] 7.2371095 2.4548851
## [40,] 4.1127443 1.4330438
## [41,] 8.2094629 2.3962942
## [42,] 6.4706019 0.5893438
## [43,] 7.8293276 6.4228826
## [44,] 5.5303631 8.7626921
## [45,] 5.2971958 7.7891468
## [46,] 7.8935623 7.9730883
## [47,] 0.2333120 4.5527445
## [48,] 4.7723007 4.1008408
## [49,] 7.3231374 8.1087024
## [50,] 6.9273156 6.0493329
##
## $model.list
## $model.list$response
## [1] "trainingoutput"
##
## $model.list$variables
## [1] "x1" "x2"
##
##
## $err.fct
## function (x, y)
## {
##     1/2 * (y - x)^2
## }
## <bytecode: 0x7f87234874c8>
## <environment: 0x7f872768ca10>
## attr(,"type")
## [1] "sse"
##
## $act.fct
## function (x)
## {
##     1/(1 + exp(-x))
## }
## <bytecode: 0x7f87262f1d48>
## <environment: 0x7f872768c578>
## attr(,"type")
## [1] "logistic"
##
## $linear.output
## [1] TRUE
##
## $data
##           x1        x2 trainingoutput
## 1  2.6550866 4.7761962      12.681215
## 2  3.7212390 8.6120948      32.047663
## 3  5.7285336 4.3809711      25.096540
## 4  9.0820779 2.4479728      22.232679
## 5  2.0168193 0.7067905       1.425469
## 6  8.9838968 0.9946616       8.935937
## 7  9.4467527 3.1627171      29.877406
## 8  6.6079779 5.1863426      34.271238
## 9  6.2911404 6.6200508      41.647669
## 10 0.6178627 4.0683019       2.513652
## 11 2.0597457 9.1287592      18.802923
## 12 1.7655675 2.9360337       5.183766
## 13 6.8702285 4.5906573      31.538864
## 14 3.8410372 3.3239467      12.767403
## 15 7.6984142 6.5087047      50.106704
## 16 4.9769924 2.5801678      12.841476
## 17 7.1761851 4.7854525      34.341293
## 18 9.9190609 7.6631067      76.010822
## 19 3.8003518 0.8424691       3.201679
## 20 7.7744522 8.7532133      68.051439
## 21 9.3470523 3.3907294      31.693325
## 22 2.1214252 8.3944035      17.808099
## 23 6.5167377 3.4668349      22.592454
## 24 1.2555510 3.3377493       4.190714
## 25 2.6722067 4.7635125      12.729090
## 26 3.8611409 8.9219834      34.449035
## 27 0.1339033 8.6433947       1.157379
## 28 3.8238796 3.8998954      14.912730
## 29 8.6969085 7.7732070      67.602870
## 30 3.4034900 9.6061800      32.694537
## 31 4.8208012 4.3465948      20.954069
## 32 5.9956583 7.1251468      42.719945
## 33 4.9354131 3.9999437      19.741374
## 34 1.8621760 3.2535215       6.058630
## 35 8.2737332 7.5708715      62.639371
## 36 6.6846674 2.0269226      13.549303
## 37 7.9423986 7.1112122      56.480082
## 38 1.0794363 1.2169192       1.313587
## 39 7.2371095 2.4548851      17.766272
## 40 4.1127443 1.4330438       5.893743
## 41 8.2094629 2.3962942      19.672288
## 42 6.4706019 0.5893438       3.813409
## 43 7.8293276 6.4228826      50.286852
## 44 5.5303631 8.7626921      48.460869
## 45 5.2971958 7.7891468      41.260636
## 46 7.8935623 7.9730883      62.936069
## 47 0.2333120 4.5527445       1.062210
## 48 4.7723007 4.1008408      19.570445
## 49 7.3231374 8.1087024      59.381142
## 50 6.9273156 6.0493329      41.905638
##
## $exclude
## NULL
##
## $net.result
## $net.result[[1]]
##             [,1]
##  [1,] 12.686287
##  [2,] 32.065525
##  [3,] 25.084766
##  [4,] 22.240260
##  [5,]  1.409380
##  [6,]  8.935376
##  [7,] 29.876789
##  [8,] 34.261128
##  [9,] 41.639447
## [10,]  2.477577
## [11,] 18.810054
## [12,]  5.216648
## [13,] 31.551060
## [14,] 12.748226
## [15,] 50.121339
## [16,] 12.846365
## [17,] 34.336871
## [18,] 76.009174
## [19,]  3.221296
## [20,] 68.050588
## [21,] 31.691904
## [22,] 17.797652
## [23,] 22.594596
## [24,]  4.200781
## [25,] 12.735174
## [26,] 34.430910
## [27,]  1.157772
## [28,] 14.908533
## [29,] 67.614475
## [30,] 32.693877
## [31,] 20.959841
## [32,] 42.700720
## [33,] 19.743169
## [34,]  6.042092
## [35,] 62.619676
## [36,] 13.553142
## [37,] 56.508483
## [38,]  1.316765
## [39,] 17.776273
## [40,]  5.879254
## [41,] 19.655496
## [42,]  3.812604
## [43,] 50.252157
## [44,] 48.471857
## [45,] 41.267033
## [46,] 62.941656
## [47,]  1.081730
## [48,] 19.574201
## [49,] 59.366393
## [50,] 41.935848
##
##
## $weights
## $weights[[1]]
## $weights[[1]][[1]]
##             [,1]       [,2]       [,3]       [,4]       [,5]
## [1,]  5.3069854 -4.7438254  3.4937482  2.1011582 -1.9125571
## [2,]  0.3165482  0.3623006 -0.3077353  0.4667796  0.3315504
## [3,] -0.6603752  0.3422437 -0.2800632 -0.2745539 -0.9914733
## 
## $weights[[1]][[2]]
##             [,1]         [,2]         [,3]        [,4]       [,5]       [,6
]
## [1,] -0.3552167   -0.4221027    0.2475597   -1.672978 -1.3370053 -0.224735
2
## [2,]  1.0443770    2.0902610    1.6513336    0.887525  3.3250467 -1.097387
7
## [3,]  4.4788689    3.0588821    1.5886920    4.541940  2.5920818  3.685164
8
## [4,] -5.2357096  -10.8977263   -5.5531356  -16.667975 -3.2702856 -7.270582
5
## [5,]  3.6748218   -1.0595832   -0.4137183   -1.732491 -0.5604287  1.848324
4
## [6,] -4.3185629 -181.7862310 -127.8871385 -284.874134 -6.2336951 -0.136629
7
##             [,7]        [,8]       [,9]       [,10]
## [1,] -0.3035726  -0.3521843 -1.5223237 -0.88287571
## [2,]  1.8858590   1.3781835  1.9186154 -0.03701397
## [3,]  1.5041994   3.1110069  3.6679984 12.32191368
## [4,] -5.5349722  -4.3638728 -2.7553186 -4.95893995
## [5,]  1.1795108   1.0565827 -0.2898827  5.84569537
## [6,] -6.4242755 -35.0639327 -3.3943725  1.18384849
## 
## $weights[[1]][[3]]
##             [,1]
##  [1,] -8.654180
##  [2,] 15.151625
##  [3,] 14.826133
##  [4,]  7.702594
##  [5,] 66.637047
##  [6,]  6.731100
##  [7,]  9.002911
##  [8,]  8.670619
##  [9,]  6.430165
## [10,]  6.599582
## [11,]  8.685164
## 
## 
## 
## $generalized.weights
## $generalized.weights[[1]]
##                 [,1]         [,2]
##  [1,]  -0.033218159 -0.018705020
##  [2,]  -0.008619457 -0.003570914
##  [3,]  -0.007253447 -0.009440249
##  [4,]  -0.005415317 -0.019393477
##  [5,]  -1.432932397 -3.873916651
##  [6,]  -0.022758986 -0.116748553
##  [7,]  -0.003907888 -0.010725918
##  [8,]  -0.004555857 -0.005815026
##  [9,]  -0.003963918 -0.003723018
## [10,]  -1.028683064 -0.102007028
## [11,]  -0.027429232 -0.006846006
## [12,]  -0.130351091 -0.074040819
## [13,]  -0.004806207 -0.007025962
## [14,]  -0.022364315 -0.025683239
## [15,]  -0.002551614 -0.003179654
## [16,]  -0.017105679 -0.032760710
## [17,]  -0.004165462 -0.006162177
## [18,]  -0.001353777 -0.002401711
## [19,]  -0.109908854 -0.515460443
## [20,]  -0.002057623 -0.001754877
## [21,]  -0.003524887 -0.009960595
## [22,]  -0.029159838 -0.006795874
## [23,]  -0.007067910 -0.013541168
## [24,]  -0.240850718 -0.079986887
## [25,]  -0.032841445 -0.018670859
## [26,]  -0.007791127 -0.003289482
## [27,] -39.084548848 -5.906254617
## [28,]  -0.018904414 -0.018684249
## [29,]  -0.001746191 -0.002076249
## [30,]  -0.009230159 -0.003507291
## [31,]  -0.010316801 -0.011562163
## [32,]  -0.004036201 -0.003390179
## [33,]  -0.010787676 -0.013322500
## [34,]  -0.105409866 -0.056661712
## [35,]  -0.001914265 -0.002169666
## [36,]  -0.011863202 -0.040630780
## [37,]  -0.002193083 -0.002556821
## [38,]  -3.246268446 -3.321698691
## [39,]  -0.008117273 -0.023771954
## [40,]  -0.047587463 -0.144401671
## [41,]  -0.006465672 -0.022136029
## [42,]  -0.052784325 -0.463885053
## [43,]  -0.002458681 -0.003229827
## [44,]  -0.003759454 -0.002340948
## [45,]  -0.004679365 -0.003226317
## [46,]  -0.002050683 -0.002004052
## [47,] -44.050615165  0.160274760
## [48,]  -0.011240378 -0.013150411
## [49,]  -0.002331754 -0.002047000
## [50,]  -0.003543788 -0.004063574
## 
## 
## $startweights
## $startweights[[1]]
## $startweights[[1]][[1]]
##            [,1]        [,2]       [,3]       [,4]        [,5]
## [1,] 0.91897737 -1.98935170 -0.1557955  0.4179416  0.38767161
## [2,] 0.78213630  0.61982575 -1.4707524  1.3586796 -0.05380504
## [3,] 0.07456498 -0.05612874 -0.4781501 -0.1027877 -1.37705956
## 
## $startweights[[1]][[2]]
##            [,1]       [,2]       [,3]       [,4]        [,5]       [,6]
## [1,] -0.4149946 -0.2533617  0.7685329 -1.1293631 -0.13505460  0.1887923
## [2,] -0.3942900  0.6969634 -0.1123462  1.4330237  2.40161776 -1.8049586
## [3,] -0.0593134  0.5566632  0.8811077  1.9803999 -0.03924000  1.4655549
## [4,]  1.1000254 -0.6887557  0.3981059 -0.3672215  0.68973936  0.1532533
## [5,]  0.7631757 -0.7074952 -0.6120264 -1.0441346  0.02800216  2.1726117
## [6,] -0.1645236  0.3645820  0.3411197  0.5697196 -0.74327321  0.4755095
##            [,7]         [,8]       [,9]      [,10]
## [1,] -0.7099464  0.001105352 -1.5235668  0.2670988
## [2,]  0.6107264  0.074341324  0.5939462 -0.5425200
## [3,] -0.9340976 -0.589520946  0.3329504  1.2078678
## [4,] -1.2536334 -0.568668733  1.0630998  1.1604026
## [5,]  0.2914462 -0.135178615 -0.3041839  0.7002136
## [6,] -0.4432919  1.178086997  0.3700188  1.5868335
## 
## $startweights[[1]][[3]]
##              [,1]
##  [1,]  0.55848643
##  [2,] -1.27659221
##  [3,] -0.57326541
##  [4,] -1.22461261
##  [5,] -0.47340064
##  [6,] -0.62036668
##  [7,]  0.04211587
##  [8,] -0.91092165
##  [9,]  0.15802877
## [10,] -0.65458464
## [11,]  1.76728727
## 
## 
## 
## $result.matrix
##                                      [,1]
## error                        5.357529e-03
## reached.threshold            9.485465e-03
## steps                        3.872330e+05
## Intercept.to.1layhid1        5.306985e+00
## x1.to.1layhid1               3.165482e-01
## x2.to.1layhid1              -6.603752e-01
## Intercept.to.1layhid2       -4.743825e+00
## x1.to.1layhid2               3.623006e-01
## x2.to.1layhid2               3.422437e-01
## Intercept.to.1layhid3        3.493748e+00
## x1.to.1layhid3              -3.077353e-01
## x2.to.1layhid3              -2.800632e-01
## Intercept.to.1layhid4        2.101158e+00
## x1.to.1layhid4               4.667796e-01
## x2.to.1layhid4              -2.745539e-01
## Intercept.to.1layhid5       -1.912557e+00
## x1.to.1layhid5               3.315504e-01
## x2.to.1layhid5              -9.914733e-01
## Intercept.to.2layhid1       -3.552167e-01
## 1layhid1.to.2layhid1         1.044377e+00
## 1layhid2.to.2layhid1         4.478869e+00
## 1layhid3.to.2layhid1        -5.235710e+00
## 1layhid4.to.2layhid1         3.674822e+00
## 1layhid5.to.2layhid1        -4.318563e+00
## Intercept.to.2layhid2       -4.221027e-01
## 1layhid1.to.2layhid2         2.090261e+00
## 1layhid2.to.2layhid2         3.058882e+00
## 1layhid3.to.2layhid2        -1.089773e+01
## 1layhid4.to.2layhid2        -1.059583e+00
## 1layhid5.to.2layhid2        -1.817862e+02
## Intercept.to.2layhid3        2.475597e-01
## 1layhid1.to.2layhid3         1.651334e+00
## 1layhid2.to.2layhid3         1.588692e+00
## 1layhid3.to.2layhid3        -5.553136e+00
## 1layhid4.to.2layhid3        -4.137183e-01
## 1layhid5.to.2layhid3        -1.278871e+02
## Intercept.to.2layhid4       -1.672978e+00
## 1layhid1.to.2layhid4         8.875250e-01
## 1layhid2.to.2layhid4         4.541940e+00
## 1layhid3.to.2layhid4        -1.666798e+01
## 1layhid4.to.2layhid4        -1.732491e+00
## 1layhid5.to.2layhid4        -2.848741e+02
## Intercept.to.2layhid5       -1.337005e+00
## 1layhid1.to.2layhid5         3.325047e+00
## 1layhid2.to.2layhid5         2.592082e+00
## 1layhid3.to.2layhid5        -3.270286e+00
## 1layhid4.to.2layhid5        -5.604287e-01
## 1layhid5.to.2layhid5        -6.233695e+00
## Intercept.to.2layhid6       -2.247352e-01
## 1layhid1.to.2layhid6        -1.097388e+00
## 1layhid2.to.2layhid6         3.685165e+00
## 1layhid3.to.2layhid6        -7.270583e+00
## 1layhid4.to.2layhid6         1.848324e+00
## 1layhid5.to.2layhid6        -1.366297e-01
## Intercept.to.2layhid7       -3.035726e-01
## 1layhid1.to.2layhid7         1.885859e+00
## 1layhid2.to.2layhid7         1.504199e+00
## 1layhid3.to.2layhid7        -5.534972e+00
## 1layhid4.to.2layhid7         1.179511e+00
## 1layhid5.to.2layhid7        -6.424275e+00
## Intercept.to.2layhid8       -3.521843e-01
## 1layhid1.to.2layhid8         1.378183e+00
## 1layhid2.to.2layhid8         3.111007e+00
## 1layhid3.to.2layhid8        -4.363873e+00
## 1layhid4.to.2layhid8         1.056583e+00
## 1layhid5.to.2layhid8        -3.506393e+01
## Intercept.to.2layhid9       -1.522324e+00
## 1layhid1.to.2layhid9         1.918615e+00
## 1layhid2.to.2layhid9         3.667998e+00
## 1layhid3.to.2layhid9        -2.755319e+00
## 1layhid4.to.2layhid9        -2.898827e-01
## 1layhid5.to.2layhid9        -3.394372e+00
## Intercept.to.2layhid10      -8.828757e-01
## 1layhid1.to.2layhid10       -3.701397e-02
## 1layhid2.to.2layhid10        1.232191e+01
## 1layhid3.to.2layhid10       -4.958940e+00
## 1layhid4.to.2layhid10        5.845695e+00
## 1layhid5.to.2layhid10        1.183848e+00
## Intercept.to.trainingoutput -8.654180e+00
## 2layhid1.to.trainingoutput   1.515162e+01
## 2layhid2.to.trainingoutput   1.482613e+01
## 2layhid3.to.trainingoutput   7.702594e+00
## 2layhid4.to.trainingoutput   6.663705e+01
## 2layhid5.to.trainingoutput   6.731100e+00
## 2layhid6.to.trainingoutput   9.002911e+00
## 2layhid7.to.trainingoutput   8.670619e+00
## 2layhid8.to.trainingoutput   6.430165e+00
## 2layhid9.to.trainingoutput   6.599582e+00
## 2layhid10.to.trainingoutput  8.685164e+00
##
## attr(,"class")
## [1] "nn"
```

```
#Plot the neural network
plot(net.multi)
```



```
#Test the neural network on some training data
testdata <- data.frame(x1=runif(20, min=0, max=10), x2=runif(20, min=0, max=1
0))
net.results <- predict(net.multi, testdata) #Run them through the neural netw
ork

#Lets see the results
print(net.results)

##              [,1]
##  [1,] 28.6837155
##  [2,] 57.6084736
##  [3,] 32.1724326
##  [4,] 21.5925439
##  [5,] 44.8006998
##  [6,] 50.0122773
##  [7,] 40.9470385
##  [8,] 15.4294007
##  [9,]  8.0233810
## [10,]  9.3448784
## [11,] 29.2949060
## [12,]  7.1837493
## [13,] 38.6326382
## [14,] 10.7750740
## [15,] 10.9987670
## [16,]  0.5560325
## [17,] 19.2657840
## [18,] 82.7270966
## [19,] 26.6963424
## [20,] 12.3637368
```

```
#Lets display a better version of the results
cleanoutput <- cbind(testdata,testdata$x1*testdata$x2,
                    as.data.frame(net.results))
colnames(cleanoutput) <- c("Input-x1","Input-x2","ExpectedOutput","NeuralNetO
utput")
print(cleanoutput)
```

```
##     Input-x1 Input-x2 ExpectedOutput NeuralNetOutput
## 1  7.6322269 3.744869     28.5816872      28.6837155
## 2  9.4796635 6.314202     59.8565132      57.6084736
## 3  8.1863469 3.900789     31.9332146      32.1724326
## 4  3.0829233 6.896278     21.2606977      21.5925439
## 5  6.4957946 6.894134     44.7828793      44.8006998
## 6  9.5335545 5.549006     52.9017534      50.0122773
## 7  9.5373265 4.296244     40.9746825      40.9470385
## 8  3.3997920 4.527201     15.3915406      15.4294007
## 9  2.6247411 3.064433      8.0433422       8.0233810
## 10 1.6545393 5.783539      9.5690935       9.3448784
## 11 3.2216806 9.103703     29.3292232      29.2949060
## 12 5.1012521 1.426041      7.2745937       7.1837493
## 13 9.2396847 4.150476     38.3490920      38.6326382
## 14 5.1095970 2.109258     10.7774558      10.7750740
## 15 2.5762126 4.287504     11.0455211      10.9987670
## 16 0.4646089 1.326900      0.6164894       0.5560325
## 17 4.1785626 4.600964     19.2254179      19.2657840
## 18 8.5400150 9.429571     80.5286745      82.7270966
## 19 3.4723068 7.619739     26.4580700      26.6963424
## 20 1.3144232 9.329098     12.2623833      12.3637368
```

```
(sum((cleanoutput$NeuralNetOutput - cleanoutput$ExpectedOutput)^2) / nrow(cle
anoutput)) ^ 0.5
```

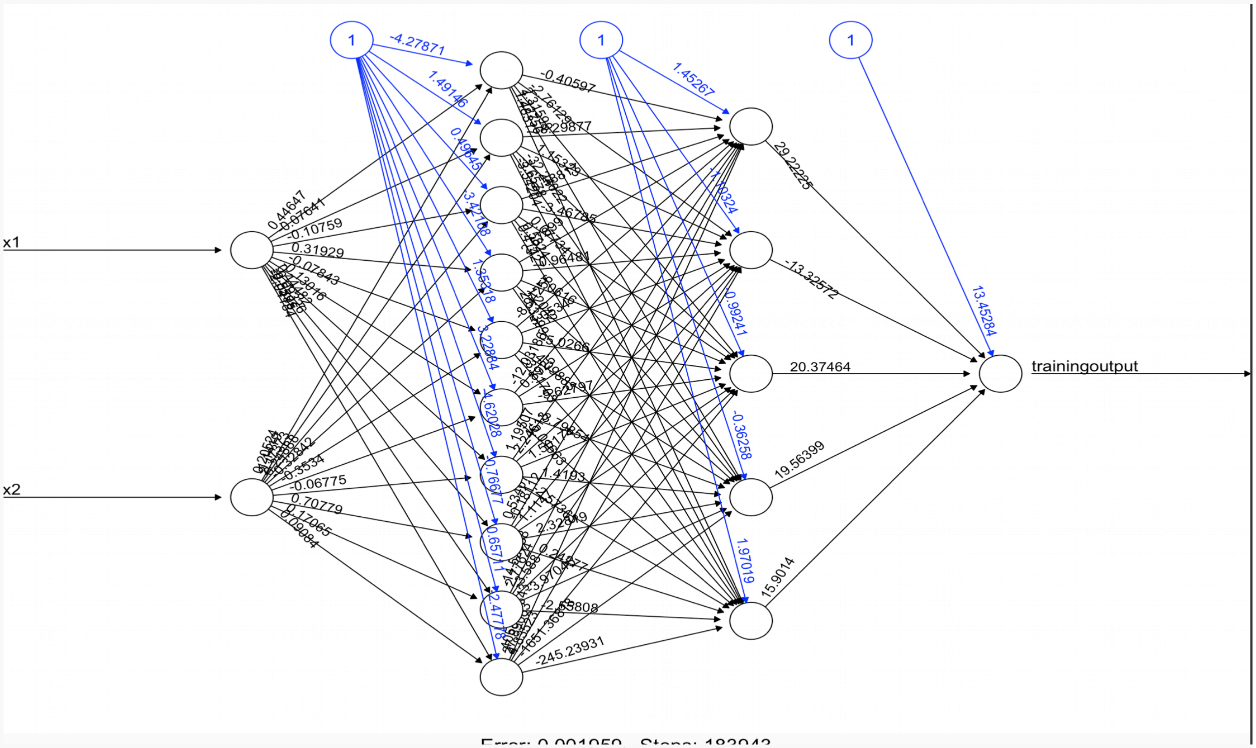
```
## [1] 0.9651712
```

```
net.multi <- neuralnet(trainingoutput~x1+x2,data=trainingdata, hidden=c(10,5)
, threshold=0.01,stepmax = 1e16)
print(net.multi)
```

```
## $call
## neuralnet(formula = trainingoutput ~ x1 + x2, data = trainingdata,
##     hidden = c(10, 5), threshold = 0.01, stepmax = 1e+16)
##
## $response
##    trainingoutput
## 1       12.681215
## 2       32.047663
## 3       25.096540
## 4       22.232679
## 5        1.425469
## 6        8.935937
## 7       29.877406
## 8       34.271238
## 9       41.647669
## 10       2.513652
## 11      18.802923
## 12       5.183766
## 13      31.538864
## 14      12.767403
## 15      50.106704
## 16      12.841476
## 17      34.341293
## 18      76.010822
## 19       3.201679
## 20      68.051439
## 21      31.693325
## 22      17.808099
## 23      22.592454
## 24       4.190714
## 25      12.729090
## 26      34.449035
## 27       1.157379
## 28      14.912730
## 29      67.602870
## 30      32.694537
## 31      20.954069
## 32      42.719945
## 33      19.741374
## 34       6.058630
## 35      62.639371
## 36      13.549303
## 37      56.480082
## 38       1.313587
## 39      17.766272
## 40       5.893743
## 41      19.672288
## 42       3.813409
## 43      50.286852
## 44      48.460869
## 45      41.260636
## 46      62.936069
## 47       1.062210
## 48      19.570445
## 49      59.381142
## 50      41.905638
##
```

```
## $covariate
##              x1        x2
##  [1,] 2.6550866 4.7761962
##  [2,] 3.7212390 8.6120948
##  [3,] 5.7285336 4.3809711
##  [4,] 9.0820779 2.4479728
##  [5,] 2.0168193 0.7067905
##  [6,] 8.9838968 0.9946616
##  [7,] 9.4467527 3.1627171
##  [8,] 6.6079779 5.1863426
##  [9,] 6.2911404 6.6200508
## [10,] 0.6178627 4.0683019
## [11,] 2.0597457 9.1287592
## [12,] 1.7655675 2.9360337
## [13,] 6.8702285 4.5906573
## [14,] 3.8410372 3.3239467
## [15,] 7.6984142 6.5087047
## [16,] 4.9769924 2.5801678
## [17,] 7.1761851 4.7854525
## [18,] 9.9190609 7.6631067
## [19,] 3.8003518 0.8424691
## [20,] 7.7744522 8.7532133
## [21,] 9.3470523 3.3907294
## [22,] 2.1214252 8.3944035
## [23,] 6.5167377 3.4668349
## [24,] 1.2555510 3.3377493
## [25,] 2.6722067 4.7635125
## [26,] 3.8611409 8.9219834
## [27,] 0.1339033 8.6433947
## [28,] 3.8238796 3.8998954
## [29,] 8.6969085 7.7732070
## [30,] 3.4034900 9.6061800
## [31,] 4.8208012 4.3465948
## [32,] 5.9956583 7.1251468
## [33,] 4.9354131 3.9999437
## [34,] 1.8621760 3.2535215
## [35,] 8.2737332 7.5708715
## [36,] 6.6846674 2.0269226
## [37,] 7.9423986 7.1112122
## [38,] 1.0794363 1.2169192
## [39,] 7.2371095 2.4548851
## [40,] 4.1127443 1.4330438
## [41,] 8.2094629 2.3962942
## [42,] 6.4706019 0.5893438
## [43,] 7.8293276 6.4228826
## [44,] 5.5303631 8.7626921
## [45,] 5.2971958 7.7891468
## [46,] 7.8935623 7.9730883
## [47,] 0.2333120 4.5527445
## [48,] 4.7723007 4.1008408
## [49,] 7.3231374 8.1087024
## [50,] 6.9273156 6.0493329
##
## $model.list
## $model.list$response
## [1] "trainingoutput"
##
## $model.list$variables
## [1] "x1" "x2"
##
##
## $err.fct
## function (x, y)
## {
##     1/2 * (y - x)^2
## }
## <bytecode: 0x7f87234874c8>
## <environment: 0x7f87248b46d0>
## attr(,"type")
## [1] "sse"
##
## $act.fct
## function (x)
## {
##     1/(1 + exp(-x))
## }
## <bytecode: 0x7f87262f1d48>
## <environment: 0x7f87248b3fd0>
## attr(,"type")
## [1] "logistic"
##
## $linear.output
## [1] TRUE
##
## $data
##           x1        x2 trainingoutput
## 1  2.6550866 4.7761962      12.681215
## 2  3.7212390 8.6120948      32.047663
## 3  5.7285336 4.3809711      25.096540
## 4  9.0820779 2.4479728      22.232679
## 5  2.0168193 0.7067905       1.425469
## 6  8.9838968 0.9946616       8.935937
## 7  9.4467527 3.1627171      29.877406
## 8  6.6079779 5.1863426      34.271238
## 9  6.2911404 6.6200508      41.647669
## 10 0.6178627 4.0683019       2.513652
## 11 2.0597457 9.1287592      18.802923
## 12 1.7655675 2.9360337       5.183766
## 13 6.8702285 4.5906573      31.538864
## 14 3.8410372 3.3239467      12.767403
## 15 7.6984142 6.5087047      50.106704
## 16 4.9769924 2.5801678      12.841476
## 17 7.1761851 4.7854525      34.341293
## 18 9.9190609 7.6631067      76.010822
## 19 3.8003518 0.8424691       3.201679
## 20 7.7744522 8.7532133      68.051439
## 21 9.3470523 3.3907294      31.693325
## 22 2.1214252 8.3944035      17.808099
## 23 6.5167377 3.4668349      22.592454
## 24 1.2555510 3.3377493       4.190714
## 25 2.6722067 4.7635125      12.729090
## 26 3.8611409 8.9219834      34.449035
## 27 0.1339033 8.6433947       1.157379
## 28 3.8238796 3.8998954      14.912730
## 29 8.6969085 7.7732070      67.602870
## 30 3.4034900 9.6061800      32.694537
## 31 4.8208012 4.3465948      20.954069
## 32 5.9956583 7.1251468      42.719945
## 33 4.9354131 3.9999437      19.741374
## 34 1.8621760 3.2535215       6.058630
## 35 8.2737332 7.5708715      62.639371
## 36 6.6846674 2.0269226      13.549303
## 37 7.9423986 7.1112122      56.480082
## 38 1.0794363 1.2169192       1.313587
## 39 7.2371095 2.4548851      17.766272
## 40 4.1127443 1.4330438       5.893743
## 41 8.2094629 2.3962942      19.672288
## 42 6.4706019 0.5893438       3.813409
## 43 7.8293276 6.4228826      50.286852
## 44 5.5303631 8.7626921      48.460869
## 45 5.2971958 7.7891468      41.260636
## 46 7.8935623 7.9730883      62.936069
## 47 0.2333120 4.5527445       1.062210
## 48 4.7723007 4.1008408      19.570445
## 49 7.3231374 8.1087024      59.381142
## 50 6.9273156 6.0493329      41.905638
##
## $exclude
## NULL
##
## $net.result
## $net.result[[1]]
##             [,1]
##  [1,] 12.673759
##  [2,] 32.035390
##  [3,] 25.096213
##  [4,] 22.245438
##  [5,]  1.473321
##  [6,]  8.934203
##  [7,] 29.872594
##  [8,] 34.270888
##  [9,] 41.646933
## [10,]  2.513399
## [11,] 18.779142
## [12,]  5.164633
## [13,] 31.533408
## [14,] 12.768903
## [15,] 50.103698
## [16,] 12.855362
## [17,] 34.347619
## [18,] 76.006658
## [19,]  3.172000
## [20,] 68.054435
## [21,] 31.695210
## [22,] 17.837519
## [23,] 22.586564
## [24,]  4.205031
## [25,] 12.720730
## [26,] 34.451060
## [27,]  1.158756
## [28,] 14.902423
## [29,] 67.626860
## [30,] 32.702748
## [31,] 20.966807
## [32,] 42.711427
## [33,] 19.740239
## [34,]  6.072580
## [35,] 62.620474
## [36,] 13.545587
## [37,] 56.499234
## [38,]  1.276660
## [39,] 17.764045
## [40,]  5.903563
## [41,] 19.663311
## [42,]  3.818786
## [43,] 50.280633
## [44,] 48.458217
## [45,] 41.267471
## [46,] 62.904148
## [47,]  1.062390
## [48,] 19.572699
## [49,] 59.398275
## [50,] 41.910615
##
##
## $weights
## $weights[[1]]
## $weights[[1]][[1]]
##            [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
## [1,] -1.3659654 -0.5287725 -3.5486671 -2.6923578  1.4755660  6.1693998
## [2,]  1.1583527  0.1539680 -1.4635407  0.4066105  0.1762529 -0.4210939
## [3,] -0.1214929  0.5832828  0.3532801  0.2637745 -0.9599647 -0.3170062
##            [,7]       [,8]       [,9]       [,10]
## [1,]  1.4669839  2.7194705 -1.4275043  1.68093708
## [2,] -0.4109916 -0.1610761  0.2575187 -0.43961823
## [3,]  1.2272652 -0.4390666 -0.1853609 -0.04220407
##
## $weights[[1]][[2]]
##             [,1]       [,2]         [,3]          [,4]        [,5]
##  [1,]   1.336118 -1.9839815  2.023526696 -7.875792e-02  0.01700331
##  [2,]   0.843029 -1.6860662  2.374793011  8.885473e-01  2.31609850
##  [3,]  -1.017594 -0.4291021  0.006362132  2.722507e-01  2.39885163
##  [4,] -16.270114 -3.4753687  5.529294715 -2.616650e+03 -0.08631115
##  [5,]   1.207864 -1.0528923  1.538815869  5.126486e-01  0.48167700
##  [6,]  -1.418668 12.2576180 -2.534647395 -1.445947e+01 -1.77646207
##  [7,]  -1.224285  8.0642423 -5.279574277 -8.490440e+00 -3.29526109
##  [8,]   3.970987 -1.6855173  0.984107075  2.557805e+00  1.00422956
##  [9,]  -3.780219  3.6657003 -4.251714877 -8.414310e+00 -3.60454873
## [10,]   2.669028 -0.3536503 -0.116615710  1.092318e+00  1.29072057
## [11,]  -2.120577  3.7642934 -3.912978300  2.856890e+00 -2.99582838
##
## $weights[[1]][[3]]
##          [,1]
## [1,]  13.51623
## [2,]  13.33814
## [3,] -14.29800
## [4,]  18.20367
## [5,]  26.11620
## [6,]  16.61165
##
##
##
## $generalized.weights
## $generalized.weights[[1]]
##               [,1]         [,2]
##  [1,]  -0.031694244 -0.017657729
##  [2,]  -0.008701219 -0.003782826
##  [3,]  -0.007207587 -0.009391084
##  [4,]  -0.005293473 -0.019255897
##  [5,]  -1.090669637 -2.513056682
##  [6,]  -0.021910552 -0.102044967
##  [7,]  -0.003800670 -0.011006149
##  [8,]  -0.004598002 -0.005826322
##  [9,]  -0.003899047 -0.003692996
## [10,]  -1.076261368 -0.130063163
## [11,]  -0.025560324 -0.005760028
## [12,]  -0.138417214 -0.086432001
## [13,]  -0.004798987 -0.007157270
## [14,]  -0.021689552 -0.025208014
## [15,]  -0.002667878 -0.003167398
## [16,]  -0.016649890 -0.031536816
## [17,]  -0.004181107 -0.006276607
## [18,]  -0.001198778 -0.001393784
## [19,]  -0.107805742 -0.526893019
## [20,]  -0.001830319 -0.001691401
## [21,]  -0.003598772 -0.009707093
## [22,]  -0.025996141 -0.006716608
## [23,]  -0.007115883 -0.013435829
## [24,]  -0.246106157 -0.098583087
## [25,]  -0.031397846 -0.017633247
## [26,]  -0.007695078 -0.003393022
## [27,]  -7.601672278 -3.997252236
## [28,]  -0.018724734 -0.018480896
## [29,]  -0.001764125 -0.001908526
## [30,]  -0.009555297 -0.003392121
## [31,]  -0.010385748 -0.011568413
## [32,]  -0.003971533 -0.003349706
## [33,]  -0.010860015 -0.013511672
## [34,]  -0.106932795 -0.063258110
## [35,]  -0.001973350 -0.002129627
## [36,]  -0.011519071 -0.040109853
## [37,]  -0.002274413 -0.002508208
## [38,]  -3.233556764 -2.926160948
## [39,]  -0.008236163 -0.024310813
## [40,]  -0.048632827 -0.150223338
## [41,]  -0.006529321 -0.022613426
## [42,]  -0.066418655 -0.430298049
## [43,]  -0.002611194 -0.003215933
## [44,]  -0.003921593 -0.002409396
## [45,]  -0.004688394 -0.003173415
## [46,]  -0.002035538 -0.002029190
## [47,] -57.043091386 -0.118485764
## [48,]  -0.011323091 -0.013285101
## [49,]  -0.002317155 -0.002145847
## [50,]  -0.003496584 -0.004021453
## 
## 
## $startweights
## $startweights[[1]]
## $startweights[[1]][[1]]
##             [,1]        [,2]       [,3]       [,4]        [,5]       [,6]
## [1,] -0.07356440 -0.32427027  0.5314962 -1.5364498 -0.65209478  1.1765833
## [2,] -0.03763417  0.06016044 -1.5183941 -0.3009761 -0.05689678 -1.6649724
## [3,] -0.68166048 -0.58889449  0.3065579 -0.5282799 -1.91435943 -0.4635304
##           [,7]        [,8]        [,9]      [,10]
## [1,] -1.115920  0.01739562  0.45018710 -0.9293621
## [2,] -0.750819 -1.28630053 -0.01855983 -1.4874603
## [3,]  2.087167 -1.64060553 -0.31806837 -1.0751923
## 
## $startweights[[1]][[2]]
##             [,1]        [,2]       [,3]       [,4]        [,5]
##  [1,]  1.0000288 -1.42449465  1.0273924 -0.9261095  0.40940184
##  [2,] -0.6212667 -0.14439960  1.2079084 -0.1771040  1.68887329
##  [3,] -1.3844268  0.20753834 -1.2313234  0.4020118  1.58658843
##  [4,]  1.8692906  2.30797840  0.9838956 -0.7317482 -0.33090780
##  [5,]  0.4251004  0.10580237  0.2199248  0.8303732 -2.28523554
##  [6,] -0.2386471  0.45699881 -1.4672500 -1.2080828  2.49766159
##  [7,]  1.0584830 -0.07715294  0.5210227 -1.0479844  0.66706617
##  [8,]  0.8864227 -0.33400084 -0.1587546  1.4411577  0.54132734
##  [9,] -0.6192430 -0.03472603  1.4645873 -1.0158475 -0.01339952
## [10,]  2.2061025  0.78763961 -0.7660820  0.4119747  0.51010842
## [11,] -0.2550270  2.07524501 -0.4302118 -0.3810761 -0.16437583
## 
## $startweights[[1]][[3]]
##            [,1]
## [1,]  0.4206946
## [2,] -0.4002467
## [3,] -1.3702079
## [4,]  0.9878383
## [5,]  1.5197450
## [6,] -0.3087406
## 
## 
## 
## $result.matrix
##                                       [,1]
## error                         5.455025e-03
## reached.threshold             9.881563e-03
## steps                         7.715600e+04
## Intercept.to.1layhid1        -1.365965e+00
## x1.to.1layhid1                1.158353e+00
## x2.to.1layhid1               -1.214929e-01
## Intercept.to.1layhid2        -5.287725e-01
## x1.to.1layhid2                1.539680e-01
## x2.to.1layhid2                5.832828e-01
## Intercept.to.1layhid3        -3.548667e+00
## x1.to.1layhid3               -1.463541e+00
## x2.to.1layhid3                3.532801e-01
## Intercept.to.1layhid4        -2.692358e+00
## x1.to.1layhid4                4.066105e-01
## x2.to.1layhid4                2.637745e-01
## Intercept.to.1layhid5         1.475566e+00
## x1.to.1layhid5                1.762529e-01
## x2.to.1layhid5               -9.599647e-01
## Intercept.to.1layhid6         6.169400e+00
## x1.to.1layhid6               -4.210939e-01
## x2.to.1layhid6               -3.170062e-01
## Intercept.to.1layhid7         1.466984e+00
## x1.to.1layhid7               -4.109916e-01
## x2.to.1layhid7                1.227265e+00
## Intercept.to.1layhid8        2.719470e+00
## x1.to.1layhid8              -1.610761e-01
## x2.to.1layhid8              -4.390666e-01
## Intercept.to.1layhid9       -1.427504e+00
## x1.to.1layhid9               2.575187e-01
## x2.to.1layhid9              -1.853609e-01
## Intercept.to.1layhid10       1.680937e+00
## x1.to.1layhid10             -4.396182e-01
## x2.to.1layhid10             -4.220407e-02
## Intercept.to.2layhid1        1.336118e+00
## 1layhid1.to.2layhid1         8.430290e-01
## 1layhid2.to.2layhid1        -1.017594e+00
## 1layhid3.to.2layhid1        -1.627011e+01
## 1layhid4.to.2layhid1         1.207864e+00
## 1layhid5.to.2layhid1        -1.418668e+00
## 1layhid6.to.2layhid1        -1.224285e+00
## 1layhid7.to.2layhid1         3.970987e+00
## 1layhid8.to.2layhid1        -3.780219e+00
## 1layhid9.to.2layhid1         2.669028e+00
## 1layhid10.to.2layhid1       -2.120577e+00
## Intercept.to.2layhid2       -1.983981e+00
## 1layhid1.to.2layhid2        -1.686066e+00
## 1layhid2.to.2layhid2        -4.291021e-01
## 1layhid3.to.2layhid2        -3.475369e+00
## 1layhid4.to.2layhid2        -1.052892e+00
## 1layhid5.to.2layhid2         1.225762e+01
## 1layhid6.to.2layhid2         8.064242e+00
## 1layhid7.to.2layhid2        -1.685517e+00
## 1layhid8.to.2layhid2         3.665700e+00
## 1layhid9.to.2layhid2        -3.536503e-01
## 1layhid10.to.2layhid2        3.764293e+00
## Intercept.to.2layhid3        2.023527e+00
## 1layhid1.to.2layhid3         2.374793e+00
## 1layhid2.to.2layhid3         6.362132e-03
## 1layhid3.to.2layhid3         5.529295e+00
## 1layhid4.to.2layhid3         1.538816e+00
## 1layhid5.to.2layhid3        -2.534647e+00
## 1layhid6.to.2layhid3        -5.279574e+00
## 1layhid7.to.2layhid3         9.841071e-01
## 1layhid8.to.2layhid3        -4.251715e+00
## 1layhid9.to.2layhid3        -1.166157e-01
## 1layhid10.to.2layhid3       -3.912978e+00
## Intercept.to.2layhid4       -7.875792e-02
## 1layhid1.to.2layhid4         8.885473e-01
## 1layhid2.to.2layhid4         2.722507e-01
## 1layhid3.to.2layhid4        -2.616650e+03
## 1layhid4.to.2layhid4         5.126486e-01
## 1layhid5.to.2layhid4        -1.445947e+01
## 1layhid6.to.2layhid4        -8.490440e+00
## 1layhid7.to.2layhid4         2.557805e+00
## 1layhid8.to.2layhid4        -8.414310e+00
## 1layhid9.to.2layhid4         1.092318e+00
## 1layhid10.to.2layhid4        2.856890e+00
## Intercept.to.2layhid5        1.700331e-02
## 1layhid1.to.2layhid5         2.316098e+00
## 1layhid2.to.2layhid5         2.398852e+00
## 1layhid3.to.2layhid5        -8.631115e-02
## 1layhid4.to.2layhid5         4.816770e-01
## 1layhid5.to.2layhid5        -1.776462e+00
## 1layhid6.to.2layhid5        -3.295261e+00
## 1layhid7.to.2layhid5         1.004230e+00
## 1layhid8.to.2layhid5        -3.604549e+00
## 1layhid9.to.2layhid5         1.290721e+00
## 1layhid10.to.2layhid5       -2.995828e+00
## Intercept.to.trainingoutput  1.351623e+01
## 2layhid1.to.trainingoutput   1.333814e+01
## 2layhid2.to.trainingoutput  -1.429800e+01
## 2layhid3.to.trainingoutput   1.820367e+01
## 2layhid4.to.trainingoutput   2.611620e+01
## 2layhid5.to.trainingoutput   1.661165e+01
##
## attr(,"class")
## [1] "nn"
```

```r
#Plot the neural network
plot(net.multi)
```



```r
#Test the neural network on some training data
testdata <- data.frame(x1=runif(20, min=0, max=10), x2=runif(20, min=0, max=1
0))
net.results <- predict(net.multi, testdata) #Run them through the neural netw
ork

#Lets see the results
print(net.results)
```

```
##             [,1]
##  [1,]  0.5814599
##  [2,] 52.7617099
##  [3,] 42.7278387
##  [4,]  3.1598949
##  [5,] 29.0568854
##  [6,]  7.0976135
##  [7,]  0.7532833
##  [8,] 13.9555120
##  [9,]  1.1950277
## [10,] 26.9085013
## [11,]  6.9210177
## [12,] 39.6182959
## [13,] 26.2224819
## [14,] 15.2019665
## [15,]  6.0200677
## [16,]  4.1583250
## [17,] 43.3236728
## [18,] 42.0593237
## [19,] 26.3655355
## [20,]  6.8334377
```

```
#Lets display a better version of the results
cleanoutput <- cbind(testdata,testdata$x1*testdata$x2,
                    as.data.frame(net.results))
colnames(cleanoutput) <- c("Input-x1","Input-x2","ExpectedOutput","NeuralNetO
utput")
print(cleanoutput)
```

```
##     Input-x1   Input-x2 ExpectedOutput NeuralNetOutput
## 1  1.0505014 0.54190429     0.56927121       0.5814599
## 2  8.0168771 6.57828069    52.73726755      52.7617099
## 3  7.3964175 5.78161917    42.76326894      42.7278387
## 4  0.5214901 9.87101764     5.14763831       3.1598949
## 5  4.8216957 6.03792401    29.11303245      29.0568854
## 6  9.2051784 0.64949919     5.97875590       7.0976135
## 7  0.4152843 1.62109082     0.67321355       0.7532833
## 8  2.9399180 4.75397920    13.97630899      13.9555120
## 9  5.0085049 0.01932835     0.09680612       1.1950277
## 10 6.0974894 4.41459143    26.91792424      26.9085013
## 11 2.6424905 2.60929737     6.89504351       6.9210177
## 12 4.2309861 9.38413745    39.70415509      39.6182959
## 13 3.6656362 7.15833284    26.23984372      26.2224819
## 14 9.4250532 1.63085478    15.37089314      15.2019665
## 15 1.2372357 4.76188018     5.89156793       6.0200677
## 16 0.7003268 6.90256723     4.83405277       4.1583250
## 17 9.6431704 4.60895180    44.44490743      43.3236728
## 18 4.4251011 9.55146738    42.26620866      42.0593237
## 19 3.7027238 7.12540122    26.38339272      26.3655355
## 20 1.7024358 3.97147933     6.76118860       6.8334377
```

```
(sum((cleanoutput$NeuralNetOutput - cleanoutput$ExpectedOutput)^2) / nrow(cle
anoutput)) ^ 0.5
```

```
## [1] 0.6417419
```

The model with 4 layers and 10 nodes in each layer has the smallest error rate with an accuracy 99.7828%. The input layer has 2 nodes, x1 and x2, the second layer and the third layer has 10 nodes and the output layer has one node.

Problem 2

Create a neural network that predicts a hitters salary will be above the median or below using the Hitters Dataset. You will have to create a variable “above median salary” and “below median salary”.

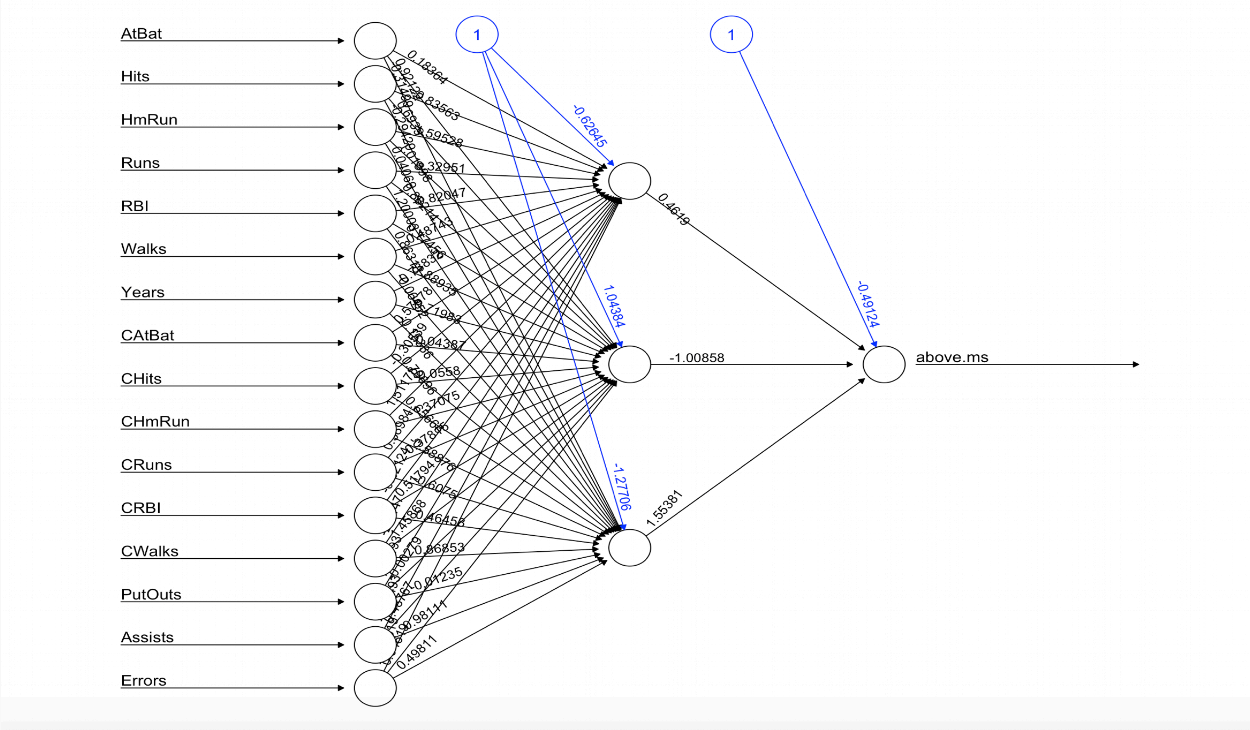
```
library(ISLR)
data("Hitters")
Hitters <-  Hitters[-which(is.na(Hitters$Salary)), ]
sum(is.na(Hitters$Salary))
```

```
## [1] 0
```

```
Hitters$above.ms<- ifelse(Hitters$Salary>=median(Hitters$Salary),1,0)
head(Hitters)
```

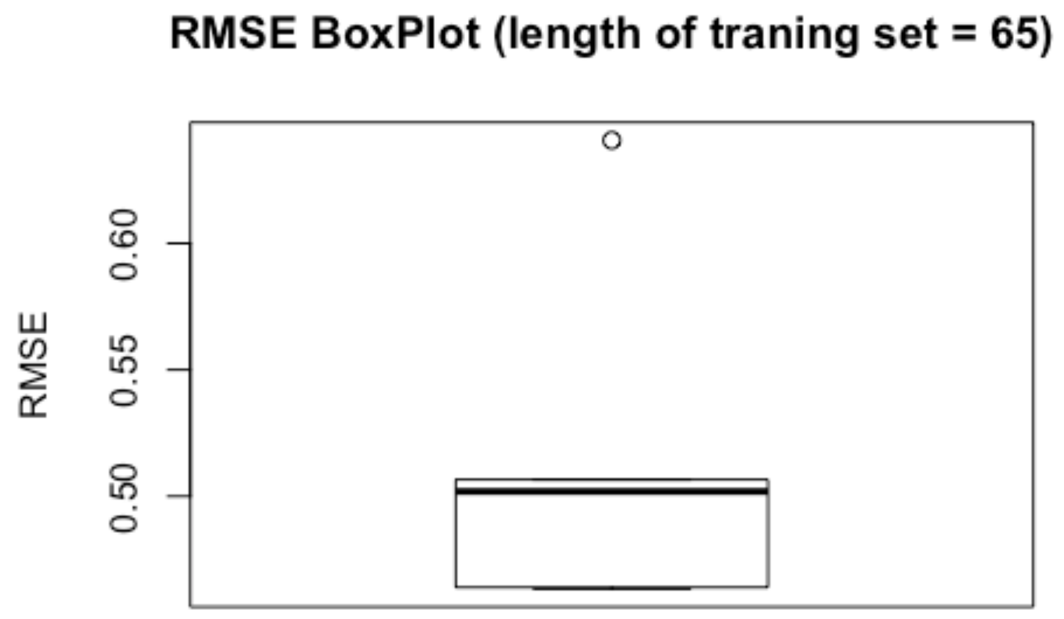
```
##                   AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRu
n
## -Alan Ashby         315   81     7   24  38    39    14   3449   835     6
9
## -Alvin Davis        479  130    18   66  72    76     3   1624   457     6
3
## -Andre Dawson       496  141    20   65  78    37    11   5628  1575    22
5
## -Andres Galarraga   321   87    10   39  42    30     2    396   101     1
2
## -Alfredo Griffin    594  169     4   74  51    35    11   4408  1133     1
9
## -Al Newman          185   37     1   23   8    21     2    214    42      
1
##                   CRuns CRBI CWalks League Division PutOuts Assists Errors
## -Alan Ashby         321  414    375      N        W     632      43     10
## -Alvin Davis        224  266    263      A        W     880      82     14
## -Andre Dawson       828  838    354      N        E     200      11      3
## -Andres Galarraga    48    46     33      N        E     805      40      4
## -Alfredo Griffin    501  336    194      A        W     282     421     25
## -Al Newman           30    9     24      N        E      76     127      7
##                   Salary NewLeague above.ms
## -Alan Ashby        475.0         N        1
## -Alvin Davis       480.0         A        1
## -Andre Dawson      500.0         N        1
## -Andres Galarraga   91.5         N        0
## -Alfredo Griffin   750.0         A        1
## -Al Newman          70.0         A        0

# Random sampling
samplesize = 0.60 * nrow(Hitters)
set.seed(80)
index = sample( seq_len ( nrow ( Hitters ) ), size = samplesize )

# Create training and test set
datatrain = Hitters[ index, ]
datatest = Hitters[ -index, ]

## Fit neural network
library(neuralnet)

# fit neural network
set.seed(1)
NN = neuralnet(above.ms ~AtBat+Hits+HmRun +Runs+ RBI+ Walks+ Years+ CAtBat+ C
Hits+ CHmRun+ CRuns+ CRBI+ CWalks+ PutOuts+ Assists+ Errors, data= datatrain,
hidden = 3 , linear.output = F,stepmax = 1e16)

# plot neural network
```

```r
plot(NN)
```

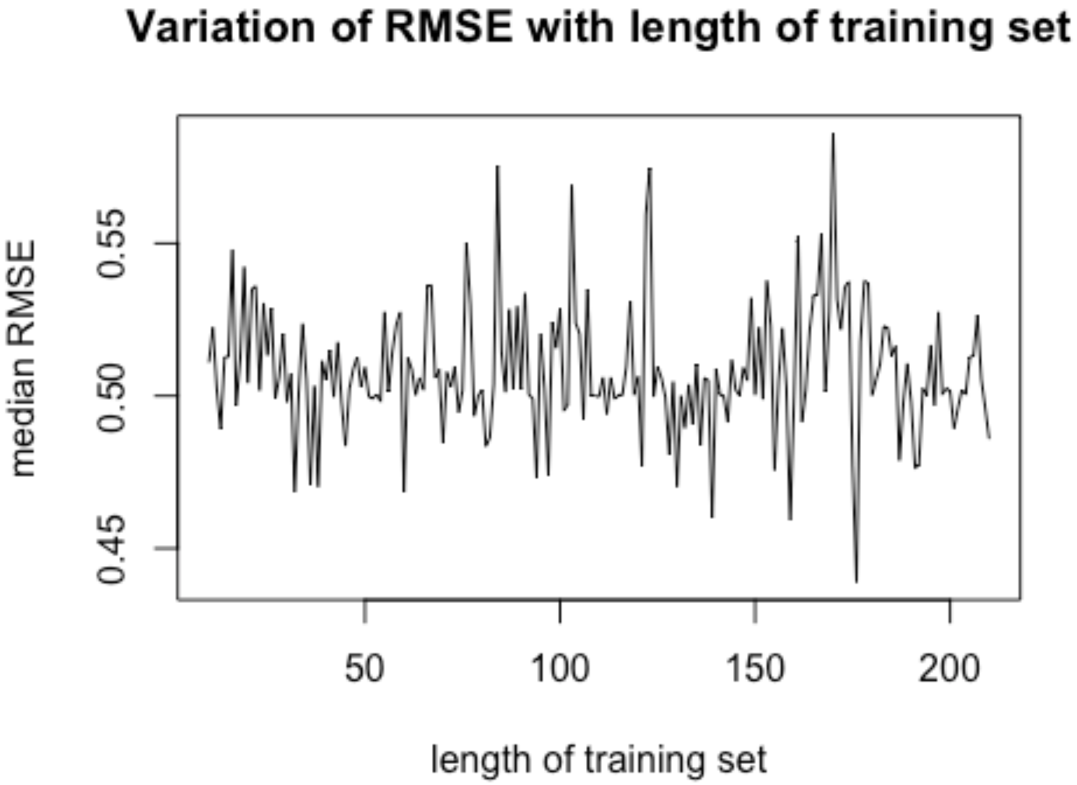


```r
# Prediction using neural network

predict_testNN = compute(NN, datatest[,-c(14,15,10,20,21)])
predict_testNN = (predict_testNN$net.result * (max(Hitters$above.ms) - min(Hitters$above.ms))) + min(Hitters$above.ms)

# Calculate Root Mean Square Error (RMSE)
RMSE.NN = (sum((datatest$above.ms - predict_testNN)^2) / nrow(datatest)) ^ 0.5
RMSE.NN

## [1] 0.436304

## Cross validation of neural network model
library(boot)
library(plyr)

# Initialize variables
```

```
set.seed(50)
k = 5
RMSE.NN = NULL

List = list( )

# Fit neural network model within nested for loop
for(j in 10:210){
  for (i in 1:k) {
    index = sample(1:nrow(Hitters),j )

    trainNN = Hitters[index,]
    testNN = Hitters[-index,]

    NN = neuralnet(above.ms ~AtBat+Hits+HmRun +Runs+ RBI+ Walks+ Years+ CAtBat+ CHits+ CHmRun+ CRuns+ CRBI+ CWalks+ PutOuts+ Assists+ Errors, trainNN, hidden = 3, linear.output= F)
    predict_testNN = compute(NN,testNN[,-c(14,15,10,20,21)])
    predict_testNN = (predict_testNN$net.result*(max(Hitters$above.ms) - min(Hitters$above.ms))) + min(Hitters$above.ms)

    RMSE.NN [i]<- (sum((testNN$above.ms - predict_testNN)^2)/nrow(testNN))^0.5
  }
  List[[j]] = RMSE.NN
}

Matrix.RMSE = do.call(cbind, List)

## Prepare boxplot
boxplot(Matrix.RMSE[,56], ylab = "RMSE", main = "RMSE BoxPlot (length of traning set = 65)")
```



```
library(matrixStats)

##
## Attaching package: 'matrixStats'

## The following object is masked from 'package:plyr':
##
##     count

med = colMedians(Matrix.RMSE)

X = seq(10,210)

plot (med~X, type = "l", xlab = "length of training set", ylab = "median RMSE
", main = "Variation of RMSE with length of training set")
```


Variation of RMSE with length of training set
median RMSE
0.45
0.50
0.55
50
100
150
200
length of training set