ANN in R

2024-07-30

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Installing the packages

```
install.packages(c('neuralnet','keras','tensorflow'), dependencies = T)
```

```
## Installing packages into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(neuralnet)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::compute() masks neuralnet::compute()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
iris<-iris %>%mutate_if(is.character, as.factor)
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
```

```
##  versicolor:50
##  virginica :50
##
##
##
```

```
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
```

Train and test split

```
set.seed(254)
data_rows<-floor(0.80 * nrow(iris))
data_rows
```

```
## [1] 120
```

```
train_indices<-sample(c(1:nrow(iris)), data_rows)
train_indices
```

```
##   [1]  55  37 146  70  45 124  20  76 144   3  88  10 136 126 102 125  64 111
##  [19] 122  32 147 123  95 101 149 143  94 150  11  83  54  57  61  48  29  69
##  [37] 130 115 145  17  50  96  35  93  49  12  14  60  18  97 109 134  62 113
##  [55]  75 119  41  27  25  89 100  91  19 137  46 103  85   6  44  86  71  36
##  [73] 104  42 139 118 106   9  43  84  66  39   7  72 117 108   4  38 138  65
##  [91]   5   2  87  82  40  77 128  67  92 131  74  56  59 120  23  13  33 107
## [109] 127  24 116  34  68  58  73  80   8  99 121 133
```

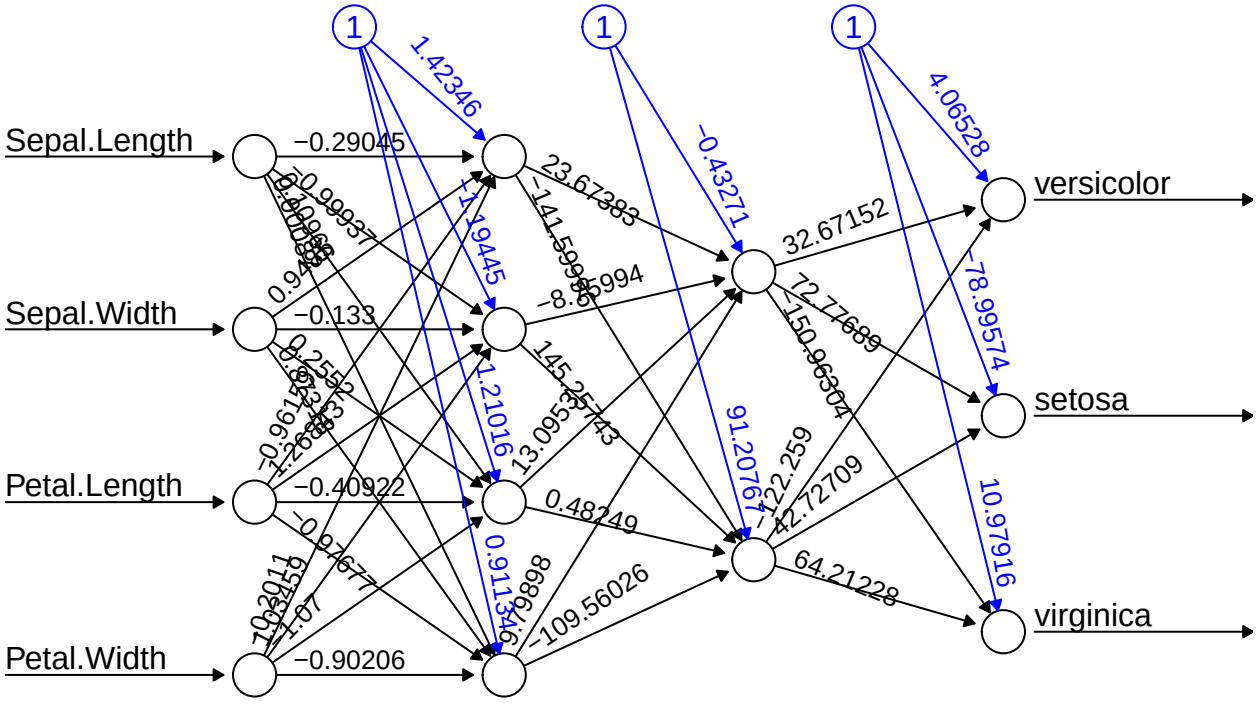
```
train_data<-iris[train_indices, ]
head(train_data)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 55           6.5         2.8          4.6         1.5 versicolor
## 37           5.5         3.5          1.3         0.2     setosa
## 146          6.7         3.0          5.2         2.3  virginica
## 70           5.6         2.5          3.9         1.1 versicolor
## 45           5.1         3.8          1.9         0.4     setosa
## 124          6.3         2.7          4.9         1.8  virginica
```

```
test_data<-iris[-train_indices,]
head(test_data)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1           5.1         3.5          1.4         0.2  setosa
## 15          5.8         4.0          1.2         0.2  setosa
## 16          5.7         4.4          1.5         0.4  setosa
## 21          5.4         3.4          1.7         0.2  setosa
## 22          5.1         3.7          1.5         0.4  setosa
## 26          5.0         3.0          1.6         0.2  setosa
```

```
model<-neuralnet( Species ~ Sepal.Length +Sepal.Width+Petal.Length +Petal.Width, data = train_data, hid
plot(model, rep = 'best')
```



Plotting the Data

```
# predict categories using test data
# create list of category name
# prediction dataframe
# create a table to display the actual and the predicted
labels<-c("setosa","versicolor","virginca")
labels
```

```
## [1] "setosa"     "versicolor" "virginca"
```

```
pred<- predict(model, test_data)
pred
```

```
##              [,1]         [,2]         [,3]
## 1   1.000000e+00 1.987582e-03 1.606099e-61
## 15  1.000000e+00 1.987582e-03 1.606099e-61
## 16  1.000000e+00 1.987582e-03 1.606099e-61
## 21  1.000000e+00 1.987582e-03 1.606099e-61
## 22  1.000000e+00 1.987582e-03 1.606099e-61
## 26  1.000000e+00 1.987582e-03 1.606099e-61
## 28  1.000000e+00 1.987582e-03 1.606099e-61
## 30  1.000000e+00 1.987582e-03 1.606099e-61
## 31  1.000000e+00 1.987582e-03 1.606099e-61
## 47  1.000000e+00 1.987582e-03 1.606099e-61
## 51  5.976903e-38 1.000000e+00 2.953469e-33
```

```
## 52  5.723452e-38 1.000000e+00 3.608146e-33
## 53  1.384220e-38 1.000000e+00 2.544987e-30
## 63  6.966252e-38 1.000000e+00 1.455306e-33
## 78  5.834333e-43 9.999693e-01 4.187287e-10
## 79  1.736209e-38 1.000000e+00 8.933657e-31
## 81  7.119429e-38 1.000000e+00 1.316157e-33
## 90  6.249596e-38 1.000000e+00 2.403280e-33
## 98  6.688873e-38 1.000000e+00 1.755865e-33
## 105 5.423696e-52 2.476923e-16 1.000000e+00
## 110 5.316714e-52 2.369408e-16 1.000000e+00
## 112 1.893062e-51 4.010254e-15 1.000000e+00
## 114 9.329015e-52 8.290613e-16 1.000000e+00
## 129 6.037474e-52 3.145041e-16 1.000000e+00
## 132 1.404842e-51 2.063591e-15 1.000000e+00
## 135 2.891381e-51 1.030162e-14 1.000000e+00
## 140 3.342740e-51 1.423096e-14 1.000000e+00
## 141 5.820653e-52 2.898980e-16 1.000000e+00
## 142 1.001202e-50 1.638601e-13 1.000000e+00
## 148 7.647401e-51 8.991549e-14 1.000000e+00
```

```
prediction_label <- data.frame(max.col(pred)) %>%
mutate(pred=labels[max.col.pred.]) %>%
select(2) %>%
unlist()
table(test_data$Species, prediction_label)
```

```
##             prediction_label
##              setosa versicolor virginca
##   setosa         10          0        0
##   versicolor      0          9        0
##   virginica       0          0       11
```

```
check= as.numeric(test_data$Species) == max.col(pred)
check
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

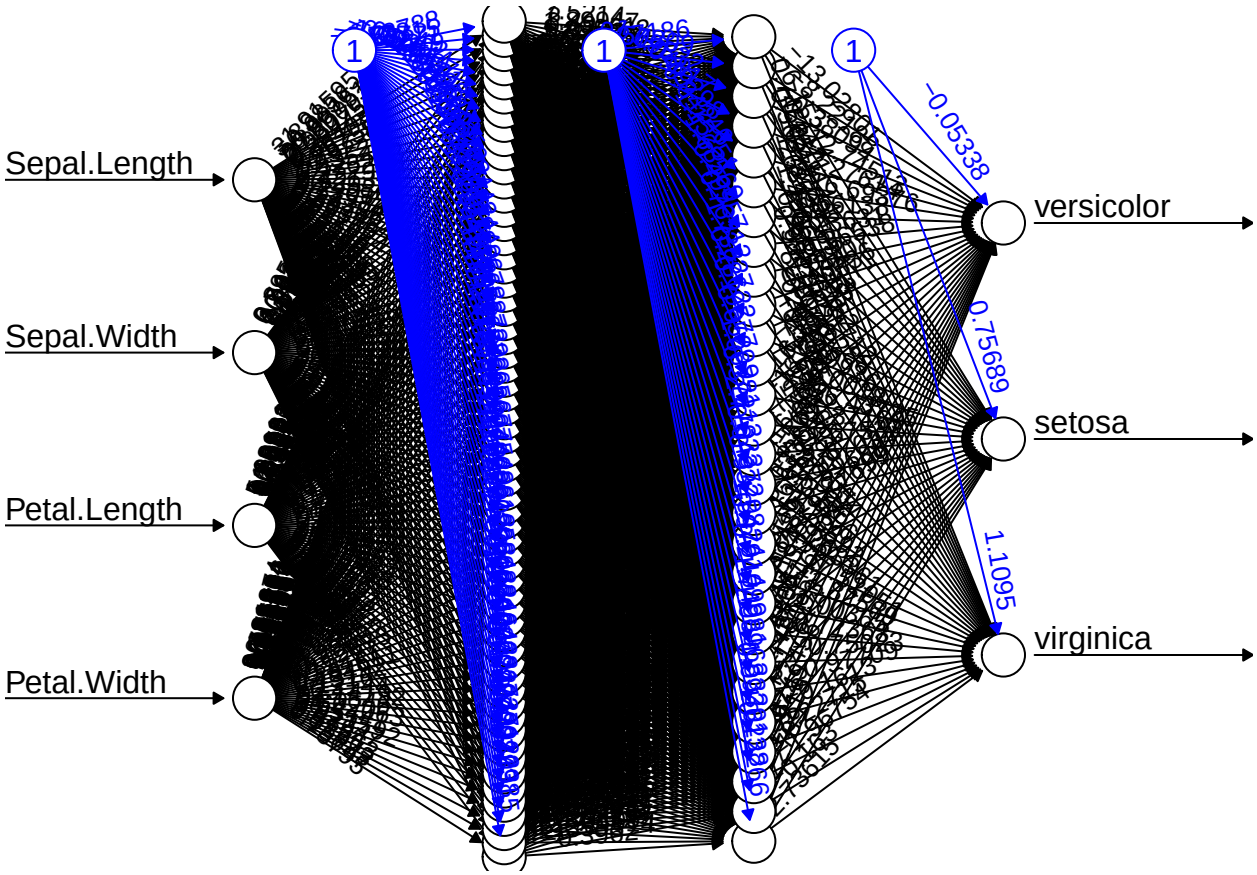
Accuracy

```
accuracy<-(sum(check)/nrow(test_data))*100
print(accuracy)
```

```
## [1] 100
```

Model 2

```
model<-neuralnet( Species ~ Sepal.Length +Sepal.Width+Petal.Length +Petal.Width, data = train_data, hid
plot(model, rep = 'best')
```

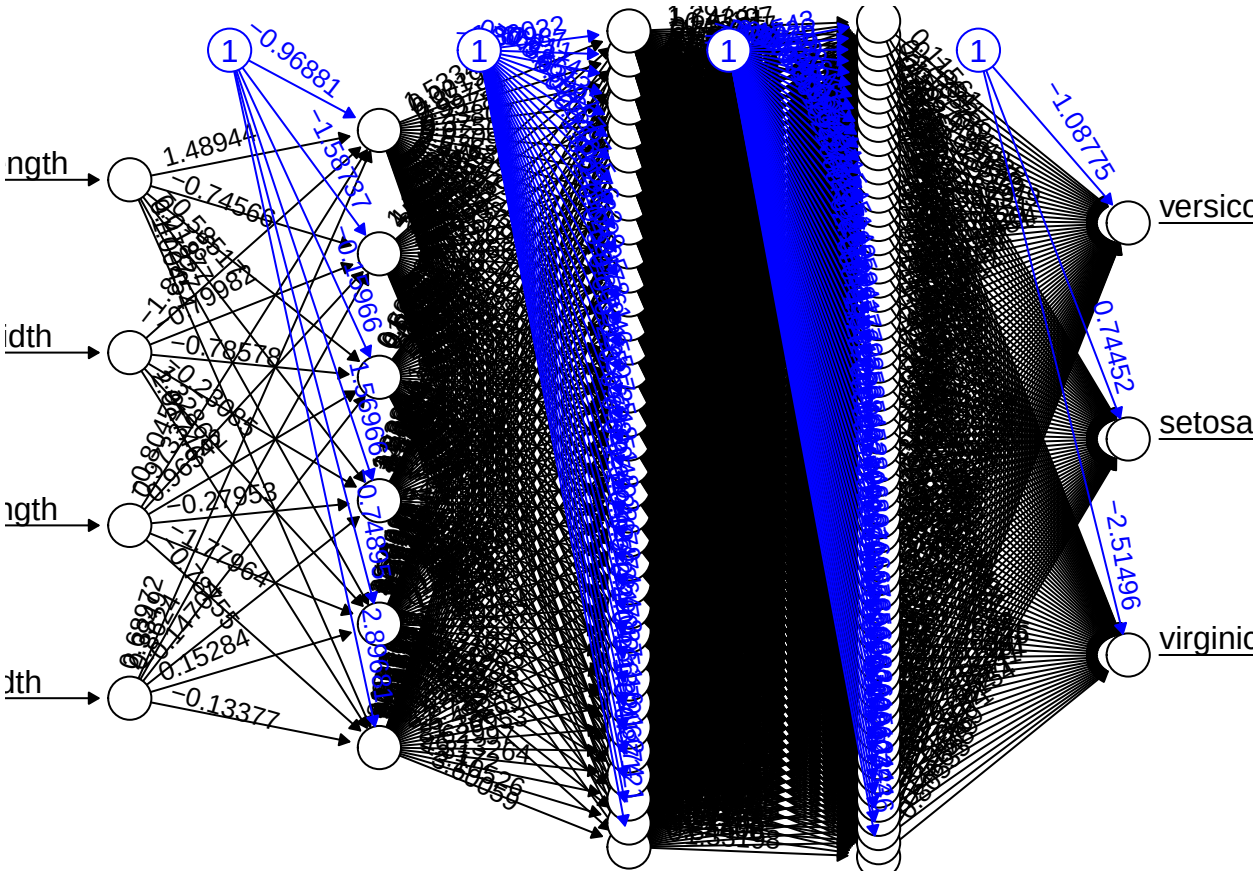



```
accuracy<-(sum(check)/nrow(test_data))*100
print(accuracy)
```

```
## [1] 100
```

Model 3

```
model<-neuralnet( Species ~ Sepal.Length +Sepal.Width+Petal.Length +Petal.Width, data = train_data, hid
plot(model, rep = 'best')
```




```
accuracy<-(sum(check)/nrow(test_data))*100
print(accuracy)
```

```
## [1] 100
```

Creating The Table

```
install.packages("knitr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(knitr)
model_accuracy <- data.frame(
  Model = c("(4, 2)", "(60, 28)", "(6, 35, 60)"),
  Accuracy = c("100%", "96%", "96%")
)
# Print the table using knitr
kable(model_accuracy, caption = "Model Accuracy Comparison")
```

Table 1: Model Accuracy Comparison

Model	Accuracy
(4, 2)	100%
(60, 28)	96%
(6, 35, 60)	96%