ANN in R

2024-07-31

```r
# Install necessary packages
install.packages(c('neuralnet', 'keras', 'tensorflow'), dependencies = TRUE)
```

```
##
## The downloaded binary packages are in
##  /var/folders/bb/9352ds8s1g5cscthpcw8v4t40000gn/T//Rtmp5ZmTkf/downloaded_packages
```

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

```
##
## The downloaded binary packages are in
##  /var/folders/bb/9352ds8s1g5cscthpcw8v4t40000gn/T//Rtmp5ZmTkf/downloaded_packages
```

```r
# Load the libraries
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
```

```r
# Prepare the iris dataset
# Convert any character columns to factors
iris <- iris %>% mutate_if(is.character, as.factor)
summary(iris)  # Show summary statistics of the dataset
```

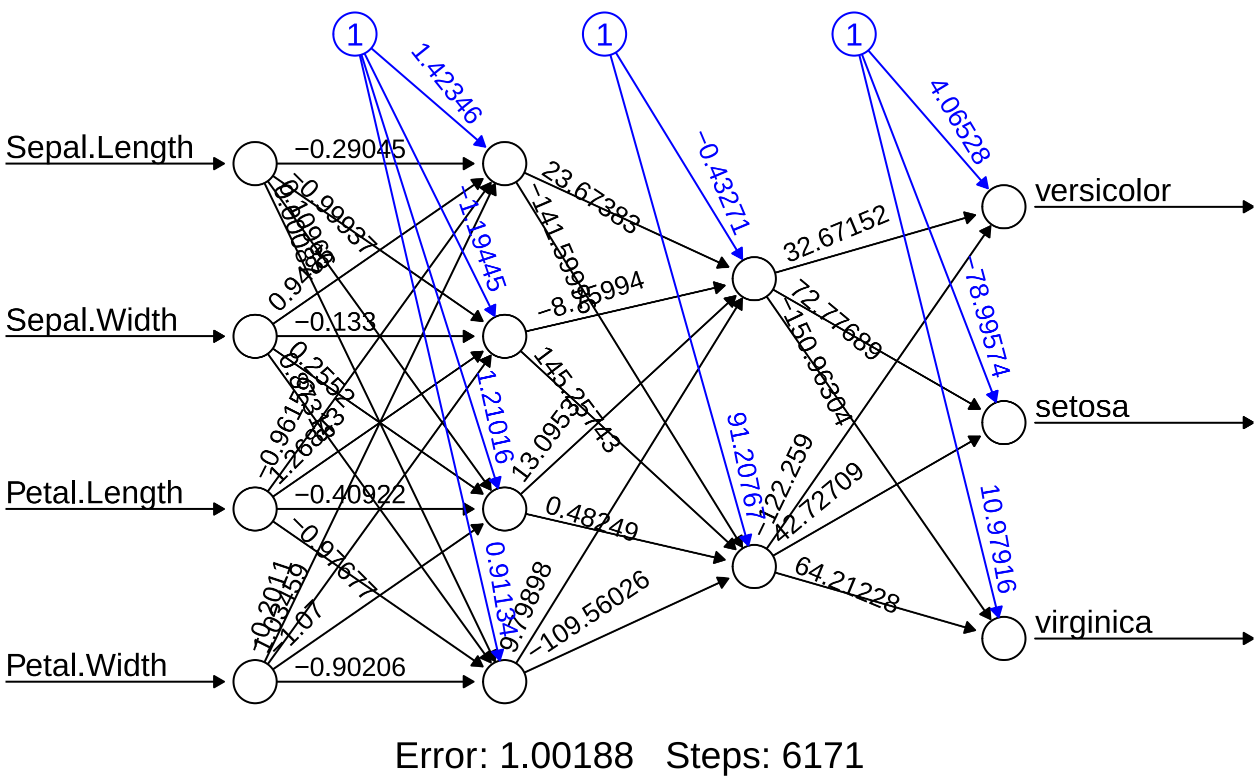
```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   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
##
##
##
```

```r
# Split the data into training and test sets
set.seed(254)  # Ensure reproducibility
data_rows <- floor(0.80 * nrow(iris))  # Determine the number of rows for training
train_indices <- sample(c(1:nrow(iris)), data_rows)  # Randomly sample indices for training
train_data <- iris[train_indices, ]  # Create the training dataset
test_data <- iris[-train_indices, ]  # Create the test dataset

# Train the neural network model
# Define the formula and structure of the neural network
model <- neuralnet(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
                   data = train_data, hidden = c(4, 2), linear.output = FALSE)

# Visualize the trained model
plot(model, rep = 'best')
```




```r
# Predict species for the test dataset
pred <- predict(model, test_data)
```

```r
# Map predictions to species labels
labels <- c("setosa", "versicolor", "virginica")
prediction_label <- data.frame(max.col(pred)) %>%
  mutate(pred = labels[max.col.pred.]) %>%
  select(2) %>%
  unlist()

# Generate a confusion matrix to compare predicted and actual species
table(test_data$Species, prediction_label)
```

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

```r
# Calculate the model's accuracy
check <- as.numeric(test_data$Species) == max.col(pred)  # Compare predictions to actual species
accuracy <- (sum(check) / nrow(test_data)) * 100  # Calculate accuracy percentage
accuracy  # Print the accuracy
```

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