Tamburo ANN

2024-08-01

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

```
##
## The downloaded binary packages are in
##  /var/folders/bb/9352ds8s1g5cscthpcw8v4t40000gn/T//RtmpT7MLQq/downloaded_packages
```

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

```
##
## The downloaded binary packages are in
##  /var/folders/bb/9352ds8s1g5cscthpcw8v4t40000gn/T//RtmpT7MLQq/downloaded_packages
```

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

```
##
## The downloaded binary packages are in
##  /var/folders/bb/9352ds8s1g5cscthpcw8v4t40000gn/T//RtmpT7MLQq/downloaded_packages
```

```r
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
library(cowplot)
```

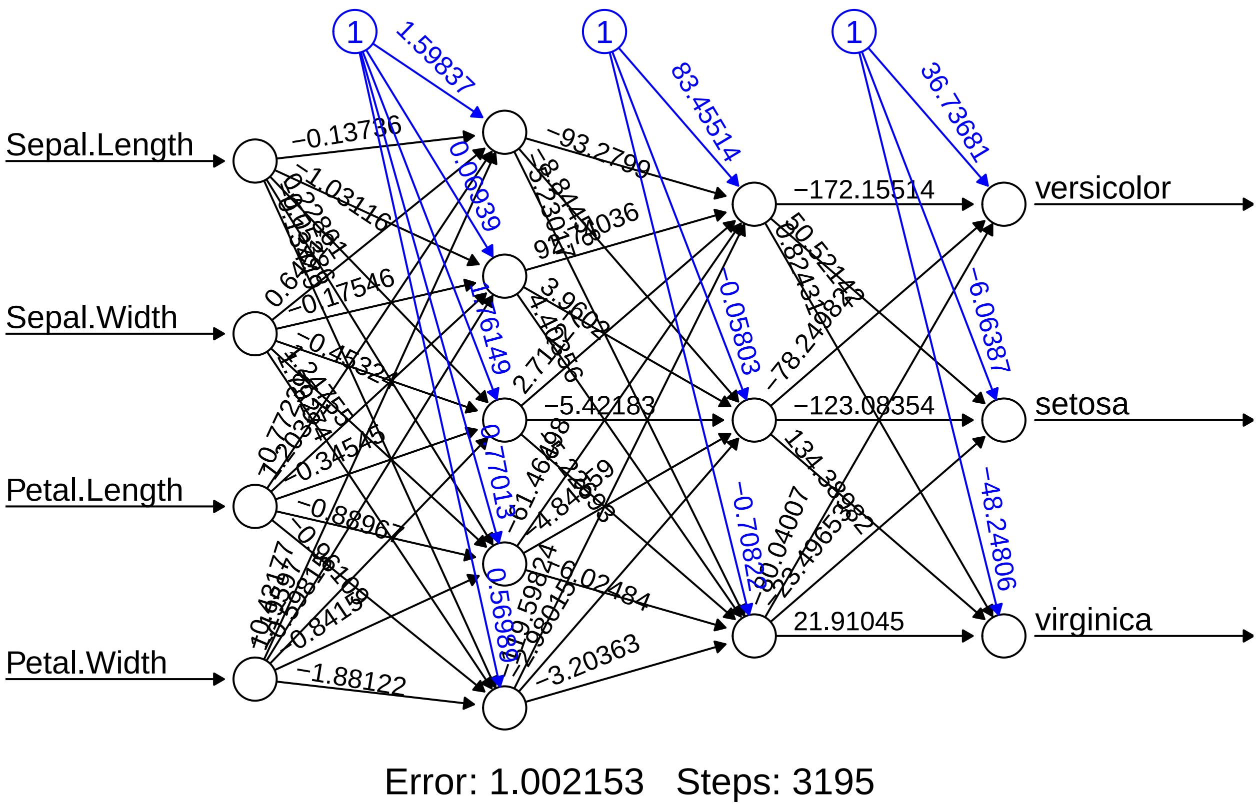
```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

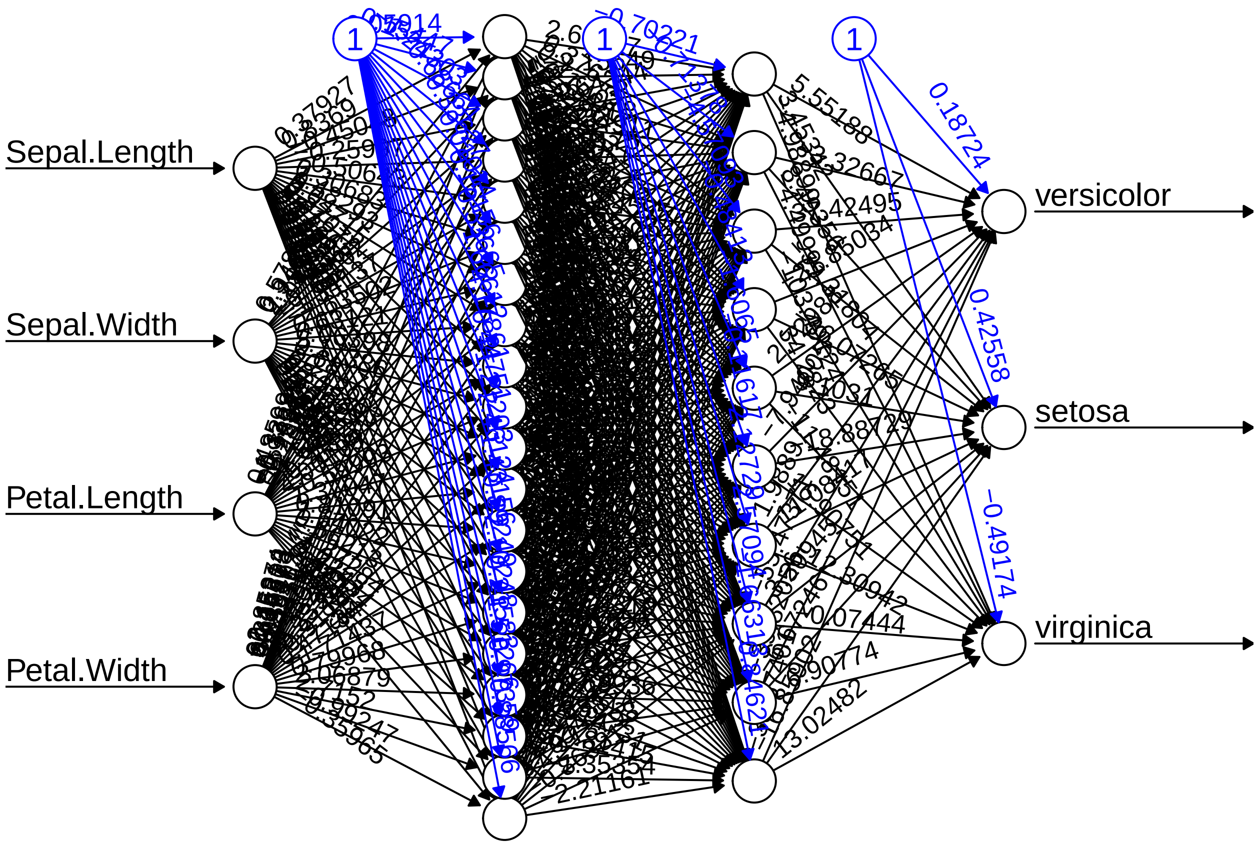
```r
# Prepare the iris dataset
iris <- iris %>% mutate_if(is.character, as.factor)
summary(iris)
```

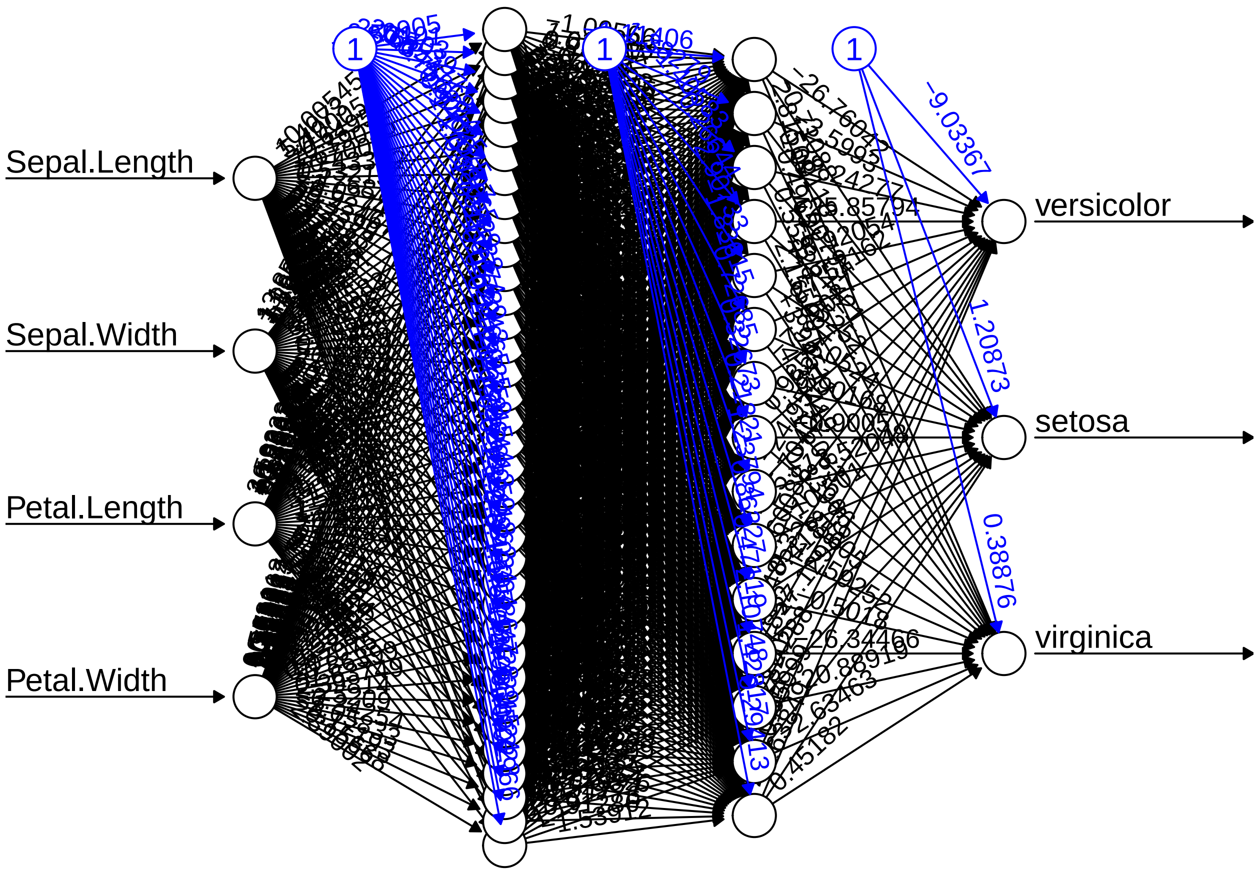
```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057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

```r
# Set the seed and split the data into training and test sets
set.seed(254)
data_rows <- floor(0.80 * nrow(iris))
train_indices <- sample(c(1:nrow(iris)), data_rows)
train_data <- iris[train_indices, ]
test_data <- iris[-train_indices, ]

# Define a function to train and evaluate the neural network model
train_and_evaluate <- function(hidden_layers) {
  model <- neuralnet(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
                     data = train_data, hidden = hidden_layers, linear.output = FALSE)
  pred <- predict(model, test_data)
  labels <- c("setosa", "versicolor", "virginica")
  prediction_label <- data.frame(max.col(pred)) %>%
    mutate(pred = labels[max.col(pred)]) %>%
    select(pred) %>%
    unlist()
  check <- as.numeric(test_data$Species) == max.col(pred)
  accuracy <- (sum(check) / nrow(test_data)) * 100
  list(model = model, accuracy = accuracy)
}

# Train and evaluate models with different hidden layer configurations
hidden_layers_list <- list(c(5, 3), c(20, 10), c(35, 15))
results <- data.frame(Hidden_Layers = character(), Accuracy = numeric(), stringsAsFactors = FALSE)
models <- list()

for (hidden_layers in hidden_layers_list) {
  result <- train_and_evaluate(hidden_layers)
  models <- c(models, list(result$model))
  results <- rbind(results, data.frame(Hidden_Layers = paste(hidden_layers, collapse = ", "), Accuracy =
}

# Create plots for the models
```

```
model_plots <- lapply(1:length(models), function(i) {
  plot(models[[i]], rep = 'best', main = paste("Model:", paste(hidden_layers_list[[i]], collapse = ", "
})
```



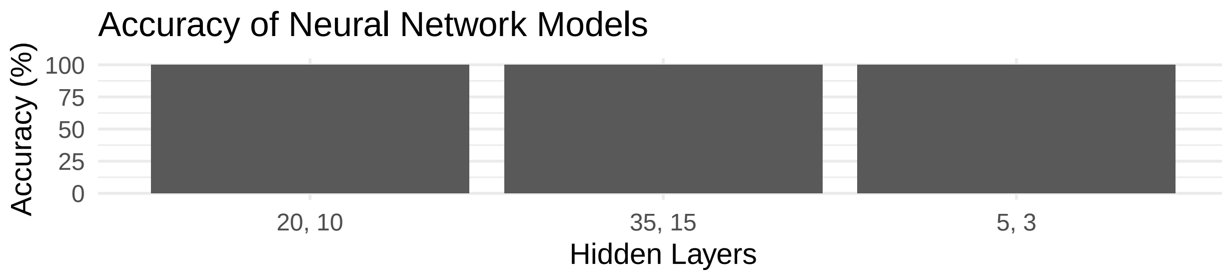

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
versicolor
setosa
virginica
0.18724
0.42558
−0.49174
5.55188
13.02482
0.07444




```r
# Plot the accuracy bar graph
accuracy_plot <- ggplot(results, aes(x = Hidden_Layers, y = Accuracy)) +
  geom_bar(stat = "identity") +
  labs(title = "Accuracy of Neural Network Models", x = "Hidden Layers", y = "Accuracy (%)") +
  theme_minimal()

# Print the accuracy of the models
print(results)
```

```
##   Hidden_Layers Accuracy
## 1          5, 3      100
## 2        20, 10      100
## 3        35, 15      100
```

```r
# Combine the model plots and accuracy plot using cowplot
top_row <- plot_grid(plotlist = model_plots, nrow = 1)
bottom_row <- plot_grid(accuracy_plot)
final_plot <- plot_grid(top_row, bottom_row, ncol = 1, rel_heights = c(2, 1))

# Display the final combined plot
print(final_plot)
```


Accuracy of Neural Network Models
Accuracy (%)
100
75
50
25
0
20, 10
35, 15
5, 3
Hidden Layers