Assignment 5 Memo

Dr. Niladri Chakraborty

2025-05-19

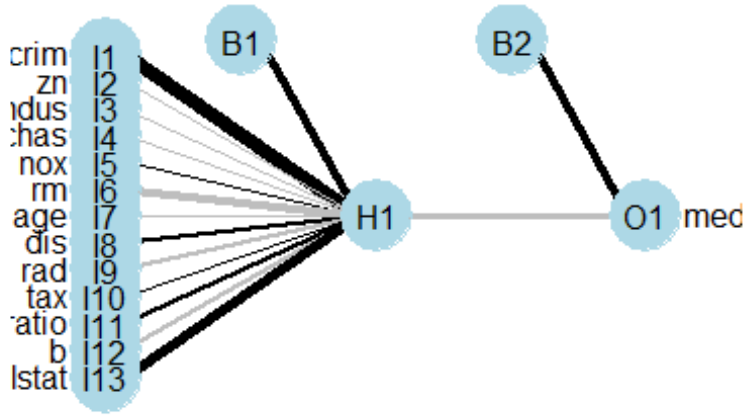
Q1.

```r
# Step 1: Load the necessary libraries
library(mlbench)       # For the dataset
library(neuralnet)     # For neural network modeling
library(NeuralNetTools) # For neural network visualization
library(dplyr)         # For data manipulation
library(caret)         # For data splitting and pre-processing

# Step 2: Load the dataset
data("BostonHousing")
dataset <- BostonHousing

# Step 3: Convert the 'chas' variable to numeric
dataset$chas <- as.numeric(dataset$chas)

# Step 4: Split the dataset into training and test sets
set.seed(123) # For reproducibility
training_rows <- createDataPartition(dataset$medv, p = 0.7, list = FALSE)
training_data <- dataset[training_rows, ]
test_data <- dataset[-training_rows, ]

# Step 5: Scale the dataset
preprocess_params <- preProcess(training_data, method = c("range"))
training_data_scaled <- predict(preprocess_params, training_data)
test_data_scaled <- predict(preprocess_params, test_data)

# Step 6: Create a neural network model
vars <- colnames(training_data_scaled)
formula <- as.formula(paste("medv ~", paste(vars[!vars %in% "medv"], collapse
= " + ")))
nn_model <- neuralnet(formula, training_data_scaled, linear.output = TRUE)

# Step 7: Print the summary of the model
summary(nn_model)
```

```
##                   Length Class      Mode
## call                 4   -none-     call
## response           356   -none-     numeric
## covariate         4628   -none-     numeric
## model.list           2   -none-     list
## err.fct              1   -none-     function
## act.fct              1   -none-     function
```

```
## linear.output          1   -none-     logical
## data                  14   data.frame list
## exclude                0   -none-     NULL
## net.result             1   -none-     list
## weights                1   -none-     list
## generalized.weights    1   -none-     list
## startweights           1   -none-     list
## result.matrix         19   -none-     numeric
```

```r
# Step 8: Predict 'medv' using the neural network model
predictions <- predict(nn_model, test_data_scaled)

# Step 9: Plot the neural network
plotnet(nn_model)
```




```r
# Step 10: Calculate Mean Squared Error (MSE)
actual <- test_data_scaled$medv
mse <- mean((predictions - actual)^2)
print(paste("Mean Squared Error (MSE):", mse))
```

```
## [1] "Mean Squared Error (MSE): 0.0111148801165191"
```

Q2.

```r
# Continue from the previous setup (assuming all libraries are loaded and
data is prepared)
```

```r
# Define the different configurations for the neural networks
configurations <- list(
  c(10, 10),
  c(5, 5, 5),
  c(10, 10, 10)
)

# Initialize a list to store results
results <- list()

# Loop through each configuration
for (config in configurations) {
  # Train the neural network model
  nn_model <- neuralnet(formula, training_data_scaled, hidden = config,
linear.output = TRUE)

  # Print the summary of the model
  summary(nn_model)

  # Predict 'medv' using the neural network model
  predictions <- predict(nn_model, test_data_scaled)

  # Calculate Mean Squared Error (MSE)
  actual <- test_data_scaled$medv
  mse <- mean((predictions - actual)^2)

  # Store the results
  results[[paste("Hidden layers:", paste(config, collapse = "-"))]] <- mse
}

# Print all MSE results
print(results)
```

```
## $`Hidden layers: 10-10`
## [1] 0.008600956
##
## $`Hidden layers: 5-5-5`
## [1] 0.006295382
##
## $`Hidden layers: 10-10-10`
## [1] 0.005964214
```