Sem. Test 2 - Memo

Dr. Niladri Chakraborty

2023-05-29

Q1. Write a while loop to find the first 10 Fibonacci numbers.

Ans.

```
# First two Fibonacci numbers
a <- 0
b <- 1

# counter to keep track of number of Fibonacci numbers printed
count <- 0

# while loop to print the first 10 Fibonacci numbers
while (count < 10 ) {
  print(a)

  # Calculate next Fibonacci number
  temp <- a + b
  a <- b
  b <- temp

  # Increase the counter
  count <- count + 1
}
```

```
## [1] 0
## [1] 1
## [1] 1
## [1] 2
## [1] 3
## [1] 5
## [1] 8
## [1] 13
## [1] 21
## [1] 34
```

Q2. Write a repeat loop to find the factorial of a given number (let's say 5). If the number is less than 0, it should print an error message.

Ans.

```r
# Number for which factorial is to be calculated
num <- 5

# Check if number is less than 0
if (num < 0) {
  print("Error! Factorial of a negative number doesn't exist.")
} else {
  # Initialize factorial
  factorial <- 1

  # repeat loop to calculate factorial
  repeat {
    # Break if number is 0
    if (num == 0) {
      break
    }

    # Multiply the number to factorial and decrease the number
    factorial <- factorial * num
    num <- num - 1
  }

  print(factorial)
}
```

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

Q3. The following code is meant to print the first five even numbers (2, 4, 6, 8, 10), but it contains a mistake. Identify the mistakes in the code and correct them.

```r
#count <- 0
#num <- 1
#while (count <= 5) {
#  if (num %% 2 == 0) {
#    print(num)
#  }
#  num <- num + 1
#}
```

Ans.

```r
# Initialize counter
count <- 0

# Initialize number
num <- 1

# While loop to print first five even numbers
```

```
while (count < 5) {
  if (num %% 2 == 0) {
    print(num)
    count <- count + 1
  }
  num <- num + 1
}
```

```
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
```

Q4. Write a R code to create an Artificial Neural Network (ANN) model for the mtcars data. The objective variable is 'mpg' and all other variables are input variables. Use the neuralnet and NeuralNetTools package. The ANN model fit should include the following steps:

- First, scale the dataset around mean and standard deviation. This implies subtracting the mean and dividing by the standard deviation for each variable (feature) in the dataset.
- Split the dataset into training and test data in 70:30 ratio.
- Fit an ANN model on the training data with 2 hidden layers with 4 neurons in each layer.
- Measure the accuracy of the predictions by the model in terms of the mean square error (MSE).
- Finally, create a graphical display of the ANN model.

Ans.

```
library(neuralnet)
```

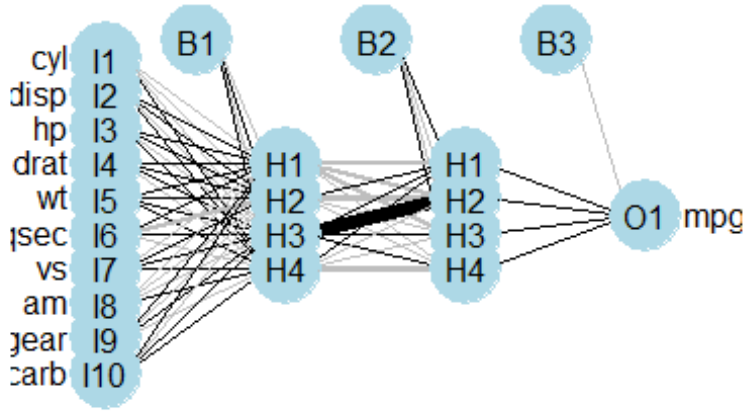
```
## Warning: package 'neuralnet' was built under R version 4.2.3
```

```
library(NeuralNetTools)
```

```
## Warning: package 'NeuralNetTools' was built under R version 4.2.3
```

```
# Load the mtcars dataset
data(mtcars)

# Scale the dataset
mtcars_scaled <- as.data.frame(scale(mtcars))
```

```r
# Split the data into training and testing sets
set.seed(123)
indices <- sample(1:nrow(mtcars_scaled), nrow(mtcars_scaled)*0.7)
train_data <- mtcars_scaled[indices, ]
test_data <- mtcars_scaled[-indices, ]

# Build the neural network
set.seed(123)
nn <- neuralnet(mpg ~ ., data = train_data, hidden = c(4, 4), linear.output =
TRUE)

# Predict on the test set
predictions_scaled <- predict(nn, test_data[,-1])

# Measure accuracy using Mean Squared Error (MSE)
mse <- mean((test_data$mpg - predictions_scaled)^2)
cat("The Mean Squared Error of the predictions is", mse)

## The Mean Squared Error of the predictions is 0.159159

# Plot the neural network
plotnet(nn)
```



Q5. For the mtcars dataset, fit a generalized linear model (glm) for the mpg dataset. Use gaussian() function for the family of the glm fit. Split the dataset into training and test data in 70:30 ratio and use the training data for model fitting. Then obtain predicted values based on the test data, and calculate prediction accuracy in terms of the mean square error (MSE).

```
# Load the mtcars dataset
data(mtcars)

# Split the data into training and testing sets
set.seed(123)
indices <- sample(1:nrow(mtcars), nrow(mtcars)*0.7)
train_data <- mtcars[indices, ]
test_data <- mtcars[-indices, ]

# Fit a generalized linear model
model <- glm(mpg ~ ., data = train_data, family = gaussian())

# Print the model summary
summary(model)

##
## Call:
## glm(formula = mpg ~ ., family = gaussian(), data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7715  -1.4518  -0.4919   1.1869   4.6713
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.130670  26.526254   0.797    0.443
## cyl         -0.888146   1.626880  -0.546    0.596
## disp         0.025947   0.030443   0.852    0.412
## hp          -0.031020   0.040364  -0.769    0.458
## drat         0.009658   2.491820   0.004    0.997
## wt          -4.881273   2.735728  -1.784    0.102
## qsec         0.929945   0.986188   0.943    0.366
## vs           0.542141   3.056351   0.177    0.862
## am           3.129479   3.322083   0.942    0.366
## gear        -0.310675   2.299091  -0.135    0.895
## carb         0.628132   1.262164   0.498    0.629
##
## (Dispersion parameter for gaussian family taken to be 10.00004)
##
##     Null deviance: 933.87  on 21  degrees of freedom
## Residual deviance: 110.00  on 11  degrees of freedom
```

```
## AIC: 121.84
##
## Number of Fisher Scoring iterations: 2

# Predict on the test set
predictions <- predict(model, newdata = test_data)

# Measure accuracy using Mean Squared Error (MSE)
mse <- mean((test_data$mpg - predictions)^2)
cat("The Mean Squared Error of the predictions is", mse)

## The Mean Squared Error of the predictions is 5.205016
```

Q6. Write a for loop in R that calculates and prints the remaining balance on a loan each year for 5 years. Assume the loan amount is $5000, the annual interest rate is 5%, and the annual payment is $1000. The formula for the remaining balance is $loan\ amount \times (1 + interest\ rate) - annual\ payment$.

Ans.

```
# Define the loan amount, interest rate, and annual payment
loan_amount <- 5000
interest_rate <- 0.05
annual_payment <- 1000

# Calculate and print the remaining balance each year
for(i in 1:5) {
  loan_amount <- loan_amount * (1 + interest_rate) - annual_payment
  cat("Year", i, ": $", round(loan_amount, 2), "\n")
}

## Year 1 : $ 4250
## Year 2 : $ 3462.5
## Year 3 : $ 2635.62
## Year 4 : $ 1767.41
## Year 5 : $ 855.78
```

Q7. Write a for loop in R that calculates and prints the balance in a bank account after each year for 10 years. Assume an initial deposit of $1000, an annual interest rate of 2%, and an annual deposit of $500. The formula for the account balance for the n^{th} year is $A_n = A_{n-1} \times (1 + interest\ rate) + annual\ deposit$.

Ans.

```
# Define the initial deposit, interest rate, and annual deposit
initial_deposit <- 1000
interest_rate <- 0.02
annual_deposit <- 500

# Initialize the account balance
account_balance <- initial_deposit

# Calculate and print the account balance each year
for(i in 1:10) {
  account_balance <- account_balance * (1 + interest_rate) + annual_deposit
  cat("Year", i, ": $", round(account_balance, 2), "\n")
}
```

```
## Year 1 : $ 1520
## Year 2 : $ 2050.4
## Year 3 : $ 2591.41
## Year 4 : $ 3143.24
## Year 5 : $ 3706.1
## Year 6 : $ 4280.22
## Year 7 : $ 4865.83
## Year 8 : $ 5463.14
## Year 9 : $ 6072.41
## Year 10 : $ 6693.85
```