федеральное государственное бюджетное образовательное учреждение высшего образования "Волгоградский государственный технический университет"

Факультет электроники и вычислительной техники

Кафедра «Системы автоматизированного проектирования и поискового конструирования»

**Лабораторная работа № 5: Нейронные сети в R**

1. **Цель работы**

* Освоить основные черты по нейронным сетям.
* Приобрести основные навыки работы с нейронными сетями в R

1. **Задачи**

* Углубить и закрепить знания по нейронным сетям.
* Научить реализовать нейроннуые сети в R
* Совершенствовать навыки самостоятельной работы.

1. **План выполнения работы**

**3.1 Теория нейронной сети**

- Что такое нейронная сеть?

- Как бывают нейронные сети?

- Для чего нужны нейронные сети?

* Классификация
* Предсказание
* Распознавание

- Как работает нейронная сеть?

- ....

**3.2 Реализация нейронной сети в**

Для реализации нейронной сети в R используем пакет neuralnet

**Пример 1:** Рассмотрим простой набор данных (квадратов чисел), который использован для обучения в R и потом проверим точность построенной нейронной сети:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| output | 0 | 1 | 4 | 9 | 16 | 25 | 36 | 49 | 64 | 81 | 100 |

Наша задача состоит в том, чтобы модолировать функцию зависимости между входом и выходом, котоую можно использовать в будущем для определения выхода на основе входа.

**Решение:**

> library(neuralnet)

>

> # 1. creating the initial data

> mydata <- data.frame (

+ input = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10),

+ output = c(0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100)

+ )

>

> attach(mydata)

> names(mydata)

[1] "input" "output"

>

> # Train the model based on output from input

>

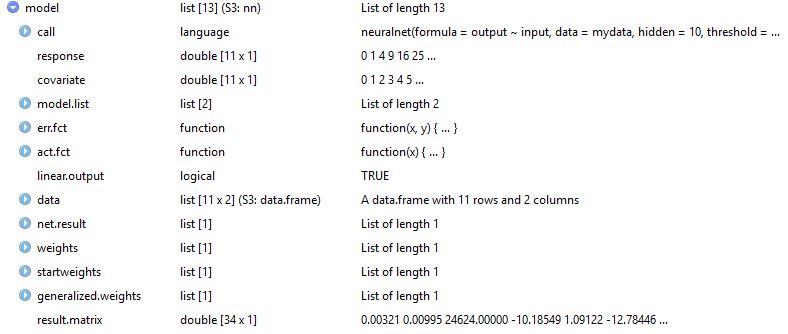
> model <- neuralnet(formula = output~input,

+ data = mydata, hidden = 10,

+ threshold = 0.01)

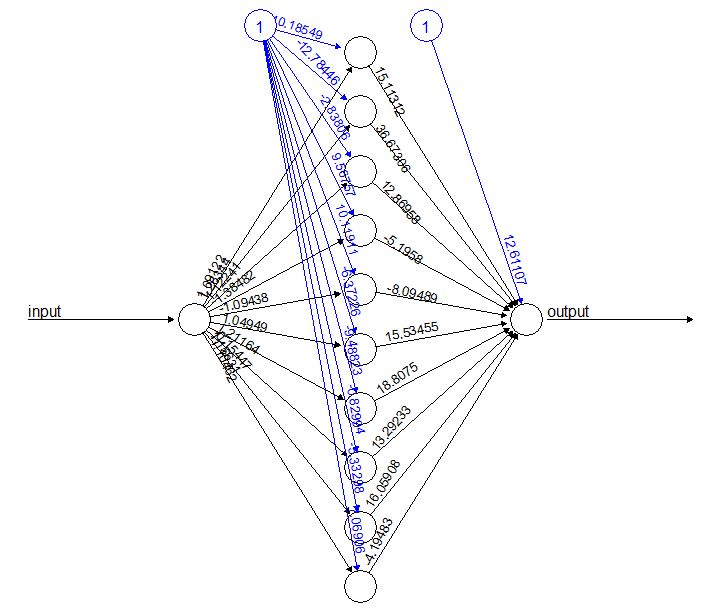
>

> print(model)

****

> # Lets plot and see layers

> plot(model)

****

> # Check the data - actual and predicted

>

> final\_output = cbind(input, output, as.data.frame(model$net.result))

> colnames(final\_output) = c("input", "expected output", "neural net output")

>

> print(final\_output)

input expected output neural net output

1 0 0 -0.08630961653

2 1 1 1.14131409009

3 2 4 3.93514119202

4 3 9 8.98723717501

5 4 16 16.04747794543

6 5 25 24.95984507457

7 6 36 36.01328041739

8 7 49 49.01081985054

9 8 64 63.98068535186

10 9 81 81.01430053277

11 10 100 99.99539110201

**Пример 2:** Прелагаю такую простую задачу: необходимо на языке R реализовать нейронную сеть для предсказания размера пенсии в зависимости средней зарплаты. У нас есть два ряда данных - средняя зарплата по городу за последние 10 лет и средняя пенсия за последние 10 лет. Эти данные используются для обучения нейронной сети:

**Решение:**

> # 1. creating the initial data, plotting

> data <- data.frame (

+ input = c(0.225, 690, 2313, 2931, 4061, 4937, 5809, 7096, 8803, 10095, 12229, 13572),

+ output = c(0.118, 274, 949, 1270, 1668, 2001, 2434, 3028, 3393, 4519, 5594, 7610)

+ )

> plot(data$output ~ data$input, main="Distribution of the pension relative to the salary", xlab="Salary", ylab="Pension")

>

> # 2. normalizing the data, plotting

> min.input <- min(data$input)

> min.output <- min(data$output)

> range.input <- diff(range(data$input))

> range.output <- diff(range(data$output))

> data.norm <- data.frame (

+ input = (data$input - min.input) / range.input,

+ output = (data$output - min.output) / range.output

+ )

> plot(data.norm$output ~ data.norm$input, main="Distribution of the pension relative to the salary (normalized)", xlab="Salary", ylab="Pension")

>

> # 3. neural network

> net <- neuralnet(output ~ input, data.norm)

>

> # 4. test the output

> testdata <- seq(0, 25000, by=500)

> testdata.norm <- (testdata - min.input) / range.input

> result <- round(compute(net, testdata.norm)$net.result \* range.output + min.output)

> plot(testdata, result, main="Predicred outcome", xlab="Salary", ylab="Pension")

