

Министерство образования и науки Российской Федерации

Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

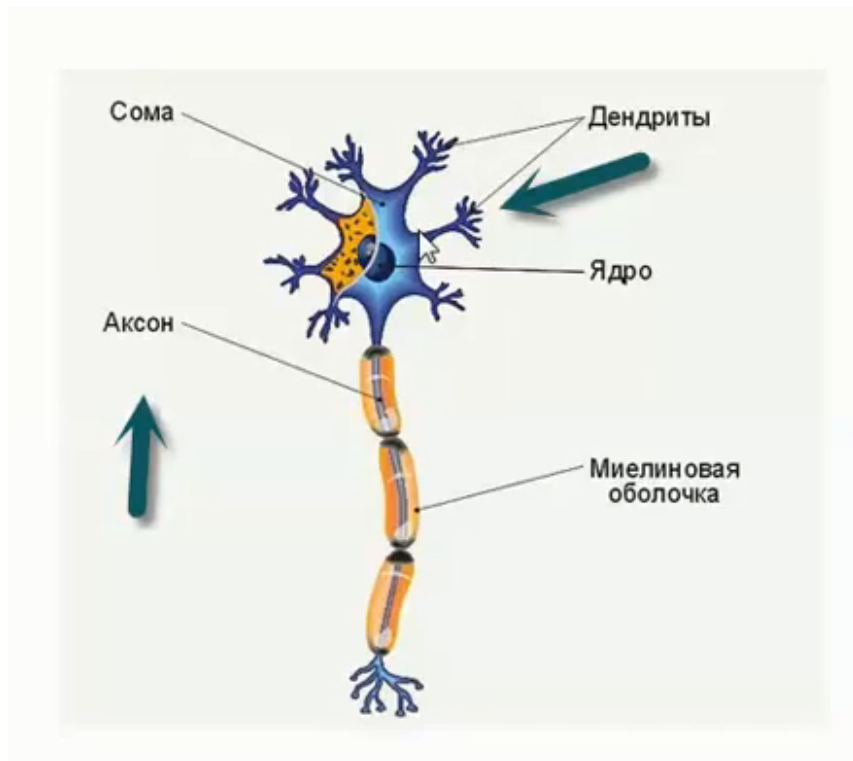
И.И. Кручинин
(к.т.н. доцент)

РАЗРАБОТКА НЕЙРОННЫХ СЕТЕЙ С ПОМОЩЬЮ ЯЗЫКА R
Методические указания по выполнению домашней работы
по курсу «Введение в машинное обучение»
Приложение.

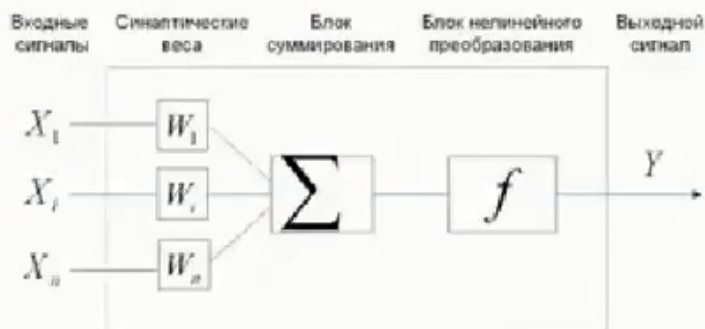
Калуга - 2018

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ. РАЗРАБОТКА НЕЙРОННЫХ СЕТЕЙ С ПОМОЩЬЮ ЯЗЫКА R

Биологический нейрон:



Искусственный нейрон



В качестве альтернативной возможности классификации исходных множеств в рамках данного домашнего задания предлагается использовать функцию языка статистического моделирования R: MLP (многослойный перцептрон). Данный алгоритм также основан на постулатах теории нейронных сетей. Используется обучение с учителем прямого и обратного распространения.

Параметр функции MLP – learnFunc (алгоритм обучения):

- 1) Std_Backpropagation,
- 2) BackpropBatch,
- 3) BackpropChunk,
- 4) BackpropMomentum,
- 5) BackpropWeightDecay,
- 6) Rprop,
- 7) Quickprop,
- 8) SCG (scaled conjugate gradient)

Параметры нейронных сетей можно оптимизировать средствами пакета **caret**

Функция `train()` из пакета `caret` перекрестной проверкой оценивает оптимальные значения числа скрытых нейронов `size` и параметр

“ослабления весов” decay, который осуществляет регуляризацию точности подстройки коэффициентов (при decay = 0 стремление к точности может перерасти в эффект переусложнения модели).

```
library(nnet)
library(caret)
```

```
load(file = "data/abalone.RData")
set.seed(123)
train.aba <- train(Возраст ~ ., data = abalone[, c(3:8, 10)],
  method = "nnet", trace = FALSE, linout = 1,
  tuneGrid = expand.grid(.decay = c(0, 0.05, 0.2), .size = 4:9),
  trControl = trainControl(method = "cv"))
train.aba
```

Функция avNNet() осуществляет обучение заданного множества моделей нейронной сети на одном и том же наборе данных. Для моделей классификации функция avNNet() оценивает среднее значение вероятностей классов на основе частных прогнозов каждой из моделей созданного ансамбля и далее производит заключительное предсказание класса.

Пример программного кода.

Нейронная сеть NNET

```
#data(iris)
library(kohonen)
library(RSNNS)
library(class)
library(gmodels)
library(modeest)
library(nnet)
library(e1071)
```

```
ramFo = data.frame(read.table("cleverK.txt", header = TRUE, sep =
""))
```

```
#print("Исходные данные")
```

```
#print(ramFo)
```

```
alg <- matrix(0:0, nrow=150, ncol=4)
```

```
for (i in 1:50) {
```

```
  for (j in 1:4) {
```

```
    alg[i,j] = sample(5:50,1)
```

```
  }
```

```
}
```

```
for (i in 51:100) {
```

```
  for (j in 1:4) {
```

```
    alg[i,j] = sample(51:95,1)
```

```
  }
```

```
}
```

```
for (i in 101:150) {
```

```
  for (j in 1:4) {
```

```
    alg[i,j] = sample(1:4,1)
```

```
  }
```

```
}
```

```
#alg
```

```
write.table(alg, file="GMB1.txt")
```

```
ramFo2 = data.frame(read.table("GMB1.txt", header = TRUE, sep =  
""))
```

```
#C1 <- c("Выгодная")
```

```
ramFoT = data.frame(read.table("org.txt", header = TRUE, sep = ""))
```

```
ramFo2 <- cbind(ramFo2, ramFoT$V1)
```

```
#ramFo2
```

```
ir <- rbind(ramFo2[,1],ramFo2[,2],ramFo2[,3])
```

```
targets <- class.ind( c(rep("V1", 50), rep("V2", 50), rep("V3", 50)) )
```

```

samp <- c(sample(1:50,25), sample(51:100,25), sample(101:150,25))
ir1 <- nnet(ir[samp,], targets[samp,], size = 2, rang = 0.1,
decay = 5e-4, maxit = 200)
test.cl <- function(true, pred) {
true <- max.col(true)
cres <- max.col(pred)
table(true, cres)
}
test.cl(targets[-samp,], predict(ir1, ir[-samp,]))
# or
ird <- data.frame(rbind(ramFo2[,1], ramFo2[,2], ramFo2[,3]),
topics = factor(c(rep("V1",50), rep("V2", 50), rep("V3", 50))))
ir.nn2 <- nnet(topics ~ ., data = ird, subset = samp, size = 2, rang = 0.1,
decay = 5e-4, maxit = 200)
table(ird$topics[-samp], predict(ir.nn2, ird[-samp,], type = "class"))
plot(ird)
print(ir.nn2)

```

Примеры нейронных сетей –многослойный персептрон

```

#data(ramFo2)
library(kohonen)
library(RSNNS)

ramFo = data.frame(read.table("cleverK.txt", header = TRUE, sep =
""))

#print("Исходные данные")
#print(ramFo)

alg <- matrix(0:0, nrow=150, ncol=4)
for (i in 1:50) {
  for (j in 1:4) {
    alg[i,j] = sample(5:50,1)
  }
}

for (i in 51:100) {

```

```

    for (j in 1:4)  {
      alg[i,j] = sample(51:95,1)
    }
  }

  for (i in 101:150) {
    for (j in 1:4)  {
      alg[i,j] = sample(1:4,1)
    }
  }

  #alg
  write.table(alg, file="GMB1.txt")
  ramFo2 = data.frame(read.table("GMB1.txt", header = TRUE, sep =
  ""))

  #C1 <- c("Выгодная")
  ramFoT = data.frame(read.table("org.txt", header = TRUE, sep =
  ""))

  ramFo2 <- cbind(ramFo2, ramFoT$V1)
  #ramFo2

  #shuffle the vector
  ramFo2 <-
ramFo2[sample(1:nrow(ramFo2),length(1:nrow(ramFo2))),1:ncol(ramFo2)]
  ramFo2Values <- ramFo2[,1:4]
  ramFo2Targets <- decodeClassLabels(ramFo2[,5])
  #ramFo2Targets <- decodeClassLabels(ramFo2[,5], valTrue=0.9,
valFalse=0.1)
  ramFo2 <- splitForTrainingAndTest(ramFo2Values, ramFo2Targets,
ratio=0.15)
  ramFo2 <- normTrainingAndTestSet(ramFo2)

```

```

model <- mlp(ramFo2$inputsTrain, ramFo2$targetsTrain, size=5,
learnFuncParams=c(0.1), learnFunc = "Std_Backpropagation",
maxit=50,
inputsTest=ramFo2$inputsTest,
targetsTest=ramFo2$targetsTest)

```

```

summary(model)
model
weightMatrix(model)
extractNetInfo(model)
par(mfrow=c(2,2))
plotIterativeError(model)

```

```

predictions <- predict(model,ramFo2$inputsTest)
plotRegressionError(predictions[,2], ramFo2$targetsTest[,2])

```

```

confusionMatrix(ramFo2$targetsTrain,fitted.values(model))
confusionMatrix(ramFo2$targetsTest,predictions)
plotROC(fitted.values(model)[,2], ramFo2$targetsTrain[,2])

```

```

plotROC(predictions[,2], ramFo2$targetsTest[,2])

```

```

j=3
for (i in 1:20000000) {
  j=j+1
}

```

```

#confusion matrix with 402040-method
confusionMatrix(ramFo2$targetsTrain,
encodeClassLabels(fitted.values(model),
method="402040", l=0.4, h=0.6))

```


Структура NeuralNet

```
#data(ramFo2)
library(kohonen)
library(RSNNS)
library(neuralnet)

ramFo = data.frame(read.table("cleverK.txt", header = TRUE, sep =
""))

#print("Исходные данные")
#print(ramFo)

alg <- matrix(0:0, nrow=150, ncol=4)
for (i in 1:50) {
  for (j in 1:4) {
    alg[i,j] = sample(5:50,1)
  }
}

for (i in 51:100) {
  for (j in 1:4) {
    alg[i,j] = sample(51:95,1)
  }
}

for (i in 101:150) {
  for (j in 1:4) {
    alg[i,j] = sample(1:4,1)
  }
}

#alg
write.table(alg, file="GMB1.txt")
ramFo2 = data.frame(read.table("GMB1.txt", header = TRUE, sep =
""))
```

```

#C1 <- c("Выгодная")
ramFoT = data.frame(read.table("org.txt", header = TRUE, sep =
""))
ramFo2 <- cbind(ramFo2, ramFoT$V1)
#ramFo2
ramFo3 <- cbind(ramFo2, ramFoT$V1)

#write.table(ramFo2, file="geni.txt")
#k = data.frame(read.table("geni.txt", header = TRUE, sep = ""))
gens <- matrix(0:0, nrow=150, ncol=3)
for (i in 1:50) {
  gens[i,1] = 1
  gens[i,2] = 0
  gens[i,3] = 0
}
for (i in 51:100) {
  gens[i,1] = 0
  gens[i,2] = 1
  gens[i,3] = 0
}
for (i in 101:150) {
  gens[i,1] = 0
  gens[i,2] = 0
  gens[i,3] = 1
}
write.table(gens, file="geni.txt")
ramFoR = data.frame(read.table("geni.txt", header = TRUE, sep =
""))
ramFo2 <- cbind(ramFo2, ramFoR$V1,ramFoR$V2,ramFoR$V3)
ramFo2

```

```

nn=neuralnet(ramFoR$V1+ramFoR$V2+ramFoR$V3~V1+V2+V3+
V4, data=ramFo2, hidden=2, err.fct="ce", algorithm = "backprop",
linear.output = FALSE)
?neuralnet
plot(nn)

j=3
for (i in 1:50000000) {
  j=j+1
}

print(nn)

nn$net.result
nn$weights
nn$result.matrix

par(mfrow=c(2,2))
gplot(nn,selected.covariate ="V1", min=-2.5,max=6)
gplot(nn,selected.covariate ="V2", min=-2.5,max=6)
gplot(nn,selected.covariate ="V3", min=-2.5,max=6)
gplot(nn,selected.covariate ="V4", min=-2.5,max=6)

#compute(nn, (1:10)^2)$net.result

```

ЗАДАЧИ И ПОРЯДОК ВЫПОЛНЕНИЯ ДОМАШНЕЙ РАБОТЫ

Разработать нейронную модель предметной области, указанной в варианте задания – проанализировать полученные знания.

Реализовать разработанную нейронную модель на высокоуровневом языке программирования - R. Созданная нейронная сеть должна предоставить ответы согласно выбранному варианту задания.

ВАРИАНТЫ ЗАДАНИЙ ДЛЯ ДОМАШНЕГО ПРОЕКТИРОВАНИЯ:

Задания могут выполняться группой студентов (численностью до 2-ух человек)

1. Необходимо на языке R реализовать нейронную сеть для предсказания размера пенсии военнослужащего в зависимости средней зарплаты в звании капитана, майора и подполковника и мест прохождения службы (Дальний Восток, Таджикистан, Красноярский край). Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать Std_Backpropagation). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

2. Необходимо на языке R реализовать нейронную сеть для предсказания размера квартплаты в регионах: Хабаровский край, Калининградская область, Московская область в зависимости от изменений цен на нефть в течение 2015- 2017 годов. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать BackpropBatch). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

3. Разработать нейросетевой классификатор для распознавания видов водостойких растений (гидатофиты, гидрофиты, гигрофиты). Разработать набор признаков характеризующих каждый из трех заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать BackpropChunk). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

4. Разработать нейросетевой классификатор для распознавания видов засухоустойчивых растений (мезофиты, ксерофиты, склерофиты). Разработать набор признаков характеризующих каждый из трех заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать BackpropMomentum). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

5. Разработать нейросетевой классификатор для распознавания видов растений Ксерофитов (суккуленты и склерофиты). Разработать набор признаков характеризующих каждый из двух заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать BackpropWeightDecay). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

6. Разработать нейросетевой классификатор для распознавания видов растений Склерофитов (зуксерофиты и стипаксерофиты). Разработать набор признаков характеризующих каждый из двух заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать Rprop). Оптимизировать параметры нейронных сетей с

помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

7. Разработать нейросетевой классификатор для распознавания видов растений по способу регулирования воды внутри организма (пойкилогидридные и гомогидридные). Разработать набор признаков характеризующих каждый из двух заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать Quickrpgor). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

8. Разработать нейросетевой классификатор для распознавания видов растений по потребности во влажной среде (эвригигробионты и стеногигробионты). Разработать набор признаков характеризующих каждый из двух заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать SCG). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

9. Разработать нейросетевой классификатор для распознавания видов растений по потребности поглощать свет (гелиофиты и сциофиты). Разработать набор признаков характеризующих каждый из двух заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать Std_Backpropagation). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

10. Разработать нейросетевой классификатор для распознавания видов теневыносливых растений (лиственница, ясень и липа). Разработать набор признаков характеризующих каждый из трех заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать BackpropBatch). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

11. Разработать нейросетевой классификатор для распознавания видов жаростойких растений (мегатермофиты, мезотермофиты). Разработать набор признаков характеризующих каждый из двух заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc

(алгоритм обучения) выбрать BackpropChunk). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

12. Разработать нейросетевой классификатор для распознавания видов холодостойких растений (микротермофиты, гекистотермофиты). Разработать набор признаков характеризующих каждый из двух заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать BackpropMomentum). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

13. Разработать нейросетевой классификатор для распознавания видов растений по отношению к механическому составу почвы(литофиты, хасмофиты). Разработать набор признаков характеризующих каждый из двух заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать BackpropWeightDecay). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

14. Разработать нейросетевой классификатор для распознавания видов растений по отношению к механическому составу песчаных почв (пелитофиты, псаммофиты). Разработать набор признаков характеризующих каждый из двух заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать Rprop). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

15. Разработать нейросетевой классификатор для распознавания видов растений по отношению к содержанию питательных веществ в почве (эутрофные, олиготрофные). Разработать набор признаков характеризующих каждый из двух заданных классов растений. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать Quickprop). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

16. Необходимо на языке R реализовать нейронную сеть для предсказания уровня инфляции (виды инфляции – ползучая, галопирующая, гиперинфляция) на основании выбранных признаков: индекс потребительских цен, индекс цен производителей, дефлятор ВВП, паритет покупательной способности, индекс Пааше. Использовать функции NeuralNet и MLP (параметр learnFunc (алгоритм обучения) выбрать SCG). Оптимизировать параметры нейронных сетей с помощью пакета caret и сравнит полученные результаты. Полный список с исходными данными взять из приложения к домашнему заданию.

ФОРМА ОТЧЕТА ПО ДОМАШНЕЙ РАБОТЕ

На выполнение домашней работы отводится 8 академических часов: 7 часов на выполнение и сдачу домашней работы и 1 час на подготовку отчета.

Номер варианта студенту выдается преподавателем.

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания (вариант), описание формы представления знаний, этапы обработки данных системой, результаты выполнения работы выводы.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Jesse, Russell Искусственная нейронная сеть / Jesse Russell. - М.: VSD, 2012. - 0 с.
2. Jesse, Russell Нейрон / Jesse Russell. - М.: VSD, 2012. - 0 с.
3. Барский, А. Б. Логические нейронные сети / А.Б. Барский. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2007. - 352 с.
4. Барский, А.Б. Логические нейронные сети / А.Б. Барский. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2013. - 0 с.
5. Бунаков, В. Е. Нейронная физика. Учебное пособие: моногр. / В.Е. Бунаков, Л.В. Краснов. - М.: Издательство Санкт-Петербургского университета, 2015. - 200 с.
6. Головинский, П. А. Математические модели. Теоретическая физика и анализ сложных систем. Книга 2. От нелинейных колебаний до искусственных нейронов и сложных систем / П.А. Головинский. - М.: Либроком, 2012. - 234 с.
7. Денис, Хусаинов Механизмы ритмической активности нейронов виноградной улитки / Хусаинов Денис , Иван Коренюк und Татьяна Гамма. - М.: LAP Lambert Academic Publishing, 2012. - 108 с.
8. Как устроено тело человека. Выпуск 25. Нейроны. - М.: DeAgostini, 2007. - 30 с.
9. Катехоламинергические нейроны. - М.: Наука, 1979. - 296 с.
10. Круглов, В.В. Искусственные нейронные сети. Теория и практика: моногр. / В.В. Круглов, В.В. Борисов. - М.: Горячая линия - Телеком; Издание 2-е, стер., 2002. - 382 с.
11. Мандельштам, Ю. Е. Нейрон и мышца насекомого: моногр. / Ю.Е. Мандельштам. - М.: Наука, 1983. - 168 с.
12. Нейронные сети. Statistica Neural Networks. Методология и технологии современного анализа данных. - М.: Горячая линия - Телеком, 2008. - 392 с.
13. Парвин, Манучер Из серого. Концерт для нейронов и синапсов / Манучер Парвин. - М.: Страта, 2015. - 408 с.
14. Позин, Н. В. Моделирование нейронных структур / Н.В. Позин. - М.: Наука, 1970. - 264 с.
15. Рассел, Джесси Вербализация нейронных сетей / Джесси Рассел. - М.: VSD, 2013. - 0 с.
16. Рассел, Джесси Искусственный нейрон / Джесси Рассел. - М.: VSD, 2013.

- 0 с.

17. Татузов, А. Л. Нейронные сети в задачах радиолокации / А.Л. Татузов. - М.: Радиотехника, 2009. - 432 с.
18. Толкачев, С. Нейронное программирование диалоговых систем / С. Толкачев. - Москва: **РГГУ**, 2016. - 192 с.
19. Шибзухов, З. М. Конструктивные методы обучения сигма-пи нейронных сетей / З.М. Шибзухов. - М.: Наука, 2006. - 160 с.
20. Юревич, Артур Нейронные сети в экономике / Артур Юревич. - М.: LAP Lambert Academic Publishing, 2014. - 80 с.

Электронные ресурсы:

- 1) <http://alexanderdyakonov.narod.ru/upR.pdf>
- 2) <http://cran.gis-lab.info/web/packages/nnet/nnet.pdf>
- 3) <http://www.faqs.org/faqs/ai-faq/neural-nets/>
- 4) <http://r-analytics.blogspot.com/>