

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Федеральное государственное образовательное бюджетное
учреждение высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА»

Ф.В. Филиппов

Интеллектуализация управления инфокоммуникационными системами и сетями

Лабораторный практикум

СПбГУТ)))

САНКТ-ПЕТЕРБУРГ
2017

Оглавление

Предисловие.....	2
Лабораторная работа № 1	
«Построение простейшей нейронной сети»	3
Лабораторная работа № 2	
«Кластеризация числовых векторов»	5
Лабораторная работа № 3	
«Построение многослойной полносвязной нейронной сети».....	6
Лабораторная работа № 4	
«Построение прогнозирующей нейронной сети ARTMAP».....	8
Источники информации	9

Предисловие

Лабораторные работы выполняются в среде RStudio [1]. Цель работ состоит в изучении способов разработки интеллектуальных средств анализа и систематизации данных. Основная задача заключается в приобретении практических навыков построения и использования нейронных сетей и кластеров. В качестве средств разработки используются программные пакеты с открытым кодом, построенные на базе языков программирования R, Python, C, C++ и других.

Для выполнения лабораторных работ необходимо предварительно установить интерпретатор языка R и среду разработки RStudio [1].

Лабораторная работа № 1

«Построение простейшей нейронной сети»

Цель работы.

Изучить пакет моделирования neuralnet [2]. Научиться формировать обучающую выборку и тестовый набор данных. Освоить способы создания, модификации и обучения простейшей нейронной сети.

Работа включает два задания.

Задание 1.

Создать нейронную сеть со структурой "персептрон", обучить ее логической функции по предложенным вариантам задания:

Номер варианта	Логическая функция (цифра вначале означает число входов функции)
1	"2И"
2	"2И-НЕ"
3	"2ИЛИ"
4	"2ИЛИ-НЕ"
5	исключающее "2ИЛИ"
6	исключающее "2ИЛИ-НЕ"
7	"2И" с инверсией по второму входу
8	"2И-НЕ" с инверсией по второму входу
9	"2ИЛИ" с инверсией по второму входу
10	"2ИЛИ-НЕ" с инверсией по второму входу
11	исключающее "2ИЛИ" с инверсией по второму входу
12	исключающее "2ИЛИ-НЕ" с инверсией по второму входу
13	"3И"
14	"3И-НЕ"
15	"3ИЛИ"
16	"3ИЛИ-НЕ"
17	исключающее "3ИЛИ"
18	исключающее "3ИЛИ-НЕ"
19	"3И" с инверсией по первому входу
20	"3И-НЕ" с инверсией по первому входу
21	"3ИЛИ" с инверсией по первому входу
22	"3ИЛИ-НЕ" с инверсией по первому входу
23	исключающее "3ИЛИ" с инверсией по первому входу
24	исключающее "3ИЛИ-НЕ" с инверсией по первому входу
25	"2И" с инверсией по первому входу

Отчет по работе №1 задание 1 (Отчет 1.1):

Оформить отчет в электронном виде (скрипт подготовки обучающих данных и обучения; топология сети; входные данные; эталонные выходные данные; полученные выходные данные, вывод по результатам работы).

Задание 2.

Построить нейронную сеть для определения направления двоичного сдвига.

Для решения поставленной задачи, следует построить и обучить нейронную сеть, которая должна будет определять направление циклического двоичного сдвига. Сущность операции циклического двоичного сдвига заключается в том, что число представляется в двоичном коде, а затем с полученной последовательностью производится операция циклического сдвига вправо или влево.

Если производится сдвиг влево, то самая старшая цифра переставляется в конец, а если сдвиг производится вправо, то самая младшая цифра переставляется в начало.

Для построения нейронной сети представим в четырехпозиционном двоичном коде числа от 0 до 15.

Число	Двоичный код	Сдвиг влево	Сдвиг вправо
1	0001	0010	1000
2	0010	0100	0001
3	0011	0110	1001
4	0100	1000	0010
5	0101	1010	1010
6	0110	1100	0011
7	0111	1110	1011
8	1000	0001	0100
9	1001	0011	1100
10	1010	0101	0101
11	1011	0111	1101
12	1100	1001	0110
13	1101	1011	1110
14	1110	1101	0111
15	1111	1111	1111

Очевидно, что для определения направления сдвига на входы нейронной сети необходимо представить исходную четырехпозиционную двоичную последовательность и четырехпозиционную двоичную последовательность, которая получилась в результате сдвига.

Выходной слой нейронной сети может состоять из одного нейрона. Его значение будет равно 0, если сдвиг произведен влево, и 1 – если сдвиг произведен вправо.

Создайте набор обучающих пар и обучите нейронную сеть.
Протестируйте получившуюся нейронную сеть.

Отчет по работе №1 задание 2 (Отчет 1.2):

Оформить отчет в электронном виде (скрипт подготовки обучающих данных и обучения; топология сети; входные данные; эталонные выходные данные; полученные выходные данные, вывод по результатам работы).

Лабораторная работа № 2 **«Кластеризация числовых векторов»**

Цель работы: изучить назначение и способы построения кластеров. Работа включает два задания.

Задание 1.

Цель: изучить приемы работы с функциями кластеризации и особенности их использования на примере функции `hclust()` [5].

Исходные данные для работы:

Отыскать статистические данные из сферы своих личных интересов в вебе. Загрузить их для анализа.

Пояснения:

Подготовить полученные данные для анализа с помощью функции кластеризации `hclust()`. Построить дендограмму. Представить дендограмму графически с помощью функции `plot()`. С помощью функции `rect.hclust()` выделить значимые кластеры.

Отчет по практической работе №2 задание 1 (Отчет 2.1):

Оформить отчет в электронном виде (скрипт подготовки данных и построения кластера, скрипт построения дендограмм, дендограммы, выводы по работе).

Задание 2.

Цель: изучить приемы работы с самообучающимися нейронными сетями и особенности их использования на примере самоорганизующейся сети Кохонена.

Исходные данные для работы:

В файле `vectors.txt` записаны 23 вектора, подлежащие автоматической классификации (кластеризации). Каждая колонка содержит 128 чисел и представляет собой один из векторов. Первые 8 векторов относятся к первому классу, следующие 4 – ко второму, следующие 5 – к третьему, последние 6 – к четвертому. В файле `initial.txt` записаны векторы минимальных и макси-

мальных значений для каждой из 128 переменных вектора входов. Минимальное значение везде принято равным 0, а максимальное – 1.

Пояснения:

Изучить назначение и особенности использования функции `som()` из пакета моделирования `RSSN` [2].

Используя исходные данные, построить сеть Кохонена. С помощью функции `names()` определить имена компонентов модели сети.

С помощью функции `plotActMap()` построить карты для следующих компонентов модели: `componentMaps[[i]]`, `map` и `labeledMap`.

Отчет по работе №2 задание 2 (Отчет 2.2):

Оформить отчет в электронном виде (скрипт подготовки исходных данных и формирования карты Кохонена, архитектура сети, скрипт построения карт, карты компонентов модели сети, выводы по работе)

Лабораторная работа № 3

«Построение многослойной полносвязной нейронной сети»

Цель работы: Научиться создавать и обучать многослойные нейронные сети. Работа включает два задания.

Задание 1.

Создать нейронную сеть со структурой "многослойный персептрон", и обучить ее распознаванию цифр, заданных пиксельной матрицей размером 5*7 клеток, согласно своего варианта задания (табл. 1). В каждом варианте задания указана своя дополнительная функция.

В качестве входных данных предлагается 5 файлов в формате `csv`, каждый из которых содержит по 10 «изображений» цифр в виде матрицы 5 на 7 пикселей, развернутых в векторы длиной по 35 элементов каждый:

- `set1.csv` - изображения 10 цифр (от 0 до 9) в нормальной ориентации.
- `set2.csv` - изображения 5 цифр (от 0 до 4) в нормальной ориентации (первые 5 векторов) и в перевернутом виде (следующие 5 векторов).
- `set3.csv` - изображения 5 цифр (от 5 до 9) в нормальной ориентации (первые 5 векторов) и в "зеркальном" виде по вертикали (следующие 5 векторов).
- `set4.csv` - изображения 10 цифр (от 0 до 9) в "зеркальном" виде по горизонтали.
- `set5.csv` - изображения 5 цифр (от 5 до 9) в перевернутом виде (первые 5 векторов) и в "зеркальном" виде по горизонтали (следующие 5 векторов).

Таблица 1. Варианты заданий

Номер варианта	Набор данных	Дополнительная функция распознавания
1	set1.csv	Распознать четность цифры
2	set1.csv	Распознать нечетность цифры
3	set2.csv	Распознать нечетность цифры
4	set1.csv	Распознать все степени числа 2
5	set4.csv	Распознать все степени числа 2
6	set4.csv	Распознать нечетность цифры
7	set4.csv	Распознать нечетность цифры
8	set3.csv	Распознать простые числа
9	set4.csv	Распознать простые числа
10	set1.csv	Распознать простые числа
11	set2.csv	Распознать числа, делящиеся на 3 без остатка
12	set2.csv	Распознать простые числа
13	set3.csv	Распознать нечетность цифры
14	set3.csv	Распознать все степени числа 2
15	set5.csv	Распознать четность цифры
16	set5.csv	Распознать простые числа
17	set2.csv	Распознать четность цифры
18	set5.csv	Распознать числа, делящиеся на 3 без остатка
19	set1.csv	Распознать числа, делящиеся на 3 без остатка
20	set3.csv	Распознать числа, делящиеся на 3 без остатка
21	set4.csv	Распознать четность цифры
22	set4.csv	Распознать числа, делящиеся на 3 без остатка
23	set2.csv	Распознать все степени числа 2
24	set5.csv	Распознать все степени числа 2
25	set3.csv	Распознать четность цифры

Отчет по работе №3 задание 1 (Отчет 3.1):

Оформить отчет в электронном виде.

Состав отчета: задание на лабораторную работу; описание топологии сети; входные данные; эталонные выходные данные; полученные выходные данные; выводы.

Задание 2.

Ознакомьтесь с наборами данных, содержащими рукописные цифры на сайте [3].

Загрузите тренировочные наборы mnist_train.csv (mnist_train_100.csv) и тестовые наборы mnist_test.csv (mnist_test_10.csv) в рабочий директорию.

Используя функцию read.csv() загрузите файл mnist_test.csv в переменную train, а файл mnist_test_10.csv в переменную test [2].

Для того, чтобы визуализировать цифры из загруженных наборов используйте функцию displayDigit() [3]:

```
displayDigit <- function(X){
  m <- matrix(unlist(X),nrow = 28,byrow = T)
  m <- t(apply(m, 2, rev))
  image(m,col=grey.colors(255))}
```

Изучите назначение и особенности всех функций, использованных для реализации функции `displayDigit()`: `matrix()`, `unlist()`, `t()`, `apply()`, `rev()`, `image()` и `grey.colors()`. Для этого используйте правую нижнюю панель RStudio. Название функции набирайте в окне поиска в закладке Help. Отчет о работе должен содержать примеры использования всех перечисленных выше функций.

Используя функцию `displayDigit()` получите изображение 7-ой и 8-ой цифры из набора `mnist_test_10`. Прокомментируйте полученные изображения.

Создайте нейронную сеть со структурой "многослойный персептрон" на базе функции `MLP()` пакета `RSNNS` [2], и обучите ее распознаванию цифр, заданных набором `mnist_train.csv`. Воспользуйтесь информацией из [4].

Отчет по работе №3 задание 2 (Отчет 3.2):

Оформить отчет в электронном виде.

Состав отчета: задание на лабораторную работу; описание топологии сети; полученные выходные данные для тестовых наборов `mnist_test.csv`; выводы.

Лабораторная работа № 4 **«Построение прогнозирующей нейронной сети ARTMAP»**

Цель работы: Научиться создавать и обучать нейронные сети, реализующие принципы ассоциативной памяти. Работа включает два задания.

Задание 1.

Ознакомиться с особенностями работы функции `art1()`, загрузить и изучить набор данных `snnsData` примера 21 из [2]. Построить функцию отображения букв латинского алфавита, представленных входными матрицами.

Распечатать все гласные буквы.

Отчет по работе №4 задание 1 (Отчет 4.1):

Оформить отчет в электронном виде.

Состав отчета: задание на лабораторную работу; описание функции отображения букв; полученные образцы гласных букв.

Задание 2.

Ознакомиться с особенностями работы прогнозирующей сети ARTMAP. Используя набор данных подготовленных в задании 1 и функцию `artmap()`, обучить сеть ARTMAP надежному распознаванию букв латинского алфавита.

Продemonстрировать качество обучения на тестовом наборе «HELLO ART».

Отчет по работе №4 задание 2 (Отчет 4.2):

Оформить отчет в электронном виде.

Состав отчета: задание на лабораторную работу; описание топологии сети ARTMAP; полученные выходные данные для тестового набора; выводы.

Источники информации

1. Филиппов Ф.В., Губин А.Н. Обработка информации в среде RStudio : учебное пособие / СПбГУТ, – СПб., 2015. – 99 с.

2. Филиппов Ф.В. Моделирование нейронных сетей на R: учебное пособие, СПбГУТ, – СПб., 2016. – 86 с.

3. The MNIST Dataset of Handwritten Digits:
<http://makeyourownneuralnetwork.blogspot.ru/2015/03/the-mnist-dataset-of-handwritten-digits.html>

4. Build your own neural network classifier in R:
<http://junma5.weebly.com/data-blog/build-your-own-neural-network-classifier-in-r>

5. <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/hclust.html>