

Московский государственный технический университет
имени Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Информационная безопасность»

М.А. Басараб, Н.С. Коннова

ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ НА ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

*Методические указания к лабораторным работам по дисциплинам
«Интеллектуальные технологии информационной безопасности»,
«Интеллектуальные системы информационной безопасности»
для студентов направлений «Информационная безопасность»,
«Информационная безопасность автоматизированных систем»*

Москва

(С) 2016 МГТУ им. Н.Э. БАУМАНА

УДК 004.8: 004:93

Рецензенты:

Басараб М.А., Коннова Н.С.

Интеллектуальные технологии на основе искусственных нейронных сетей.
М.: Изд-во МГТУ имени Н.Э. Баумана, 2016. 48 с.

Методические указания являются руководством для выполнения лабораторных работ по дисциплинам «Интеллектуальные технологии информационной безопасности», «Интеллектуальные системы информационной безопасности». Они охватывают различные разделы теории искусственных нейронных сетей на примерах решения задач классификации, кластеризации, прогноза, аппроксимации. Рассмотрены такие типы сетей, как персептрон с правилом обучения Видроу–Хоффа, многослойные с обратным распространением ошибки (BP), сети с радиальными базисными функциями (RBF), Хопфилда, двунаправленной ассоциативной памяти (Коско, ВАР), Кохонена.

Пособие предназначено для студентов МГТУ имени Н.Э. Баумана, обучающихся по специальностям «Информационная безопасность» (100401 - магистратура), «Информационная безопасность автоматизированных систем» (090303 - специалитет). Пособие может быть также полезно студентам и аспирантам других специальностей, интересующимся современными методами искусственного интеллекта на основе искусственных нейронных сетей.

Рекомендовано НМС МГТУ им. Н.Э. Баумана

Басараб Михаил Алексеевич

Коннова Наталья Сергеевна

ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ НА ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

© 2016 МГТУ имени Н.Э. Баумана

Оглавление

Введение	
1 Лабораторная работа № 1: <i>Исследование однослойных нейронных сетей на примере моделирования булевых выражений</i>	
2 Лабораторная работа № 2: <i>Применение однослойной нейронной сети для решения задач регрессии экспериментальных данных</i>	
3 Лабораторная работа № 3: <i>Применение однослойной нейронной сети с линейной функцией активации для прогнозирования временных рядов</i>	
4 Лабораторная работа № 4: <i>Исследование нейронных сетей с радиальными базисными функциями (RBF) на примере моделирования булевых выражений</i>	
5 Лабораторная работа № 5: <i>Исследование аппроксимативных свойств нейронных сетей RBF</i>	
6 Лабораторная работа № 6: <i>Изучение алгоритма обратного распространения ошибки (метод Back Propagation)</i>	
7 Лабораторная работа № 7: <i>Исследование рекуррентной нейронной сети Хопфилда на примере задачи распознавания образов</i>	
8 Лабораторная работа № 8: <i>Исследование рекуррентной нейронной сети Коско (BAM) на примере задачи распознавания образов</i>	
9 Лабораторная работа № 9: <i>Алгоритмы кластерного анализа данных</i>	
Список литературы	

Введение

Во многих современных задачах в области информационной безопасности часто приходится иметь дело с необходимостью осуществления автоматизированной поддержки принятия решений или реализации полностью автоматических экспертных оценок. Одним из наиболее эффективных и универсальных подходов здесь является так называемый метод машинного обучения, теоретическая база которого опирается на аппарат искусственных нейронных сетей (ИНС) [1,2].

В силу значительного многообразия типов решаемых проблем, с середины XX в. было предложено довольно большое число различных ИНС и их модификаций. В настоящее время направление ИНС непрерывно развивается как в области теории и создания новых типов сетей, так и в плане совершенствования инструментария их реализации (программного и аппаратного). Расширяется и диапазон задач, решаемых методами машинного обучения на основе ИНС.

С точки зрения информационной безопасности ИНС хорошо зарекомендовали себя при решении задач выявления аномалий и атак в компьютерных сетях [3], распознавания биометрических образов [4] (в частности, на нейросетевой парадигме базируется СВБА – техника сверхвысоконадежной биометрической аутентификации, опирающаяся на серию ГОСТов серии Р 52633), выявления спама, нейрокриптографии и многих других.

В данном пособии предлагается практическое ознакомление с наиболее известными типами ИНС на примерах решения ряда более простых математических задач, которые в дальнейшем смогут помочь и при решении реальных проблем в области информационной безопасности.

Лабораторный практикум включает в себя работы по решению задач моделирования булевых выражений (однослойный персептрон, сеть RBF), аппроксимации (однослойный персептрон, сеть RBF) и экстраполяции (прогнозирования) функций (сеть авторегрессии), распознавания образов (рекуррентные сети ассоциативной памяти Хопфилда и Коско), кластеризации (сеть Кохонена). В целом, как принято в теории машинного обучения, решение задач распадается на два этапа: обучения НС («с учителем» или ассоциативного) и непосредственно ее функционирования (рабочий режим).

При выполнении лабораторных работ целесообразно воспользоваться либо готовыми программными пакетами математического моделирования Matlab (за исключением расширения Neural Networks Toolbox), MathCAD и др., электронными таблицами (MS Excel) либо собственными программами, написанными на языке программирования высокого уровня (C++, C#, Python и др.). В последнем случае

программа может реализовать не полное решение задачи, а какие-либо вспомогательные и наиболее рутинные шаги всей процедуры.

При подготовке отчета по каждой лабораторной работе необходимо последовательно и полно представить все основные шаги метода (алгоритма) с выводом промежуточных результатов и необходимыми комментариями, демонстрирующими понимание сути процедуры. Студент должен быть знаком с такими понятиями, как вычислительная сложность и погрешность метода, уметь провести качественный и количественный анализ получаемых результатов, быть способным лаконично ответить на предложенные ему контрольные вопросы.

Методические указания содержат достаточный теоретический материал для выполнения каждой лабораторной работы. Вместе с тем, для более глубокого понимания нейросетевых моделей и алгоритмов следует использовать дополнительные источники, в том числе указанные в списке литературы.

Данные лабораторные работы являются частью программы дисциплин, «Интеллектуальные технологии информационной безопасности», «Интеллектуальные системы информационной безопасности», посвященной искусственным нейросетевым технологиям. Они были разработаны на основе практических и лабораторных занятий, которые проводились с 2006 г. на кафедрах «Информационная безопасность» и «Теоретическая информатика и компьютерные технологии» МГТУ им. Н.Э. Баумана.

Авторы выражают благодарность аспиранту кафедры «Информационная безопасность» Микову Д.А. за участие в практической апробации ряда работ.

1 Лабораторная работа № 1

Исследование однослойных нейронных сетей на примере моделирования булевых выражений

Цель работы

Исследовать функционирование простейшей нейронной сети (НС) на базе нейрона с нелинейной функцией активации и ее обучение по правилу Видроу-Хоффа.

Постановка задачи и сведения из теории

Постановка задачи. Получить модель булевой функции (БФ) на основе однослойной НС (единичный нейрон) с двоичными входами $x_1, x_2, x_3, x_4 \in \{0,1\}$, единичным входом смещения $x_0 = 1$, синаптическими весами w_0, w_1, w_2, w_3, w_4 , двоичным выходом $y \in \{0,1\}$ и заданной нелинейной функцией активации (ФА) $f: R \rightarrow (0,1)$ (рис. 1.1).

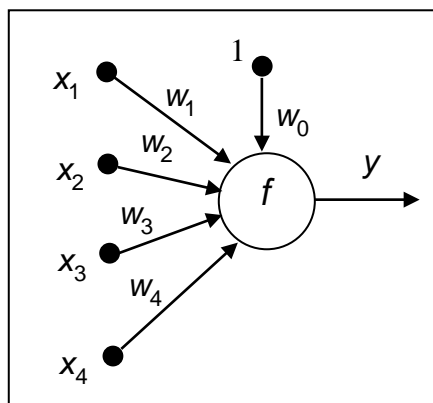


Рис. 1.1. Однослойная НС

Для заданной БФ (см. табл. 1.4) реализовать обучение НС для двух случаев:

- 1) с использованием всех комбинаций переменных x_1, x_2, x_3, x_4 ;
- 2) с использованием части возможных комбинаций переменных x_1, x_2, x_3, x_4 ;

остальные комбинации используются в качестве тестовых.

Рабочий режим НС. Алгоритм функционирования НС с пороговой ФА имеет вид:

$$\begin{aligned} \text{net} &= \sum_{i=1}^4 w_i x_i + w_0; \\ y(\text{net}) &= \begin{cases} 1, & \text{net} \geq 0, \\ 0, & \text{net} < 0, \end{cases} \end{aligned} \quad (1.1)$$

где net – сетевой (комбинированный) вход; y – реальный выход НС.

Алгоритм функционирования НС с логистической ФА имеет вид:

$$\begin{aligned}
\text{net} &= \sum_{i=1}^4 w_i x_i + w_0; \\
\text{out} &= f(\text{net}), \\
y(\text{out}) &= \begin{cases} 1, & \text{out} \geq 0.5, \\ 0, & \text{out} < 0.5, \end{cases}
\end{aligned} \tag{1.2}$$

где out – сетевой (недискретизированный) выход НС.

Режим обучения НС. Для необученной НС ее реальный выход y в общем случае отличается от целевого выхода t , представляющего собой значения заданной БФ нескольких переменных $F(x_1, x_2, x_3, x_4) : \{0,1\}^4 \rightarrow \{0,1\}$, т.е. имеется хотя бы один набор сигналов (x_1, x_2, x_3, x_4) , для которого ошибка

$$\delta = t - y \neq 0. \tag{1.3}$$

Каждая эпоха обучения $k = 1, 2, \dots$ включает в себя цикл последовательного предъявления всех образцов обучающей выборки на вход НС. Предъявление одного обучающего образца внутри эпохи является элементарным шагом обучения $l = 0, 1, 2, \dots$, во время которого вектор весовых коэффициентов $W = (w_i)_{i=0, \dots, 4}$ корректируется согласно правилу Видроу-Хоффа (дельта-правило):

$$\begin{aligned}
w_i^{(l+1)} &= w_i^{(l)} + \Delta w_i^{(l)}, \\
\Delta w_i^{(l)} &= \eta \delta^{(l)} \frac{df(\text{net})}{d \text{net}} x_i^{(l)},
\end{aligned} \tag{1.4}$$

где $x_i^{(l)}$ ($i = 0, \dots, 4$) – компоненты обучающего вектора на l -м шаге обучения (при этом $x_0^{(l)} \equiv 1$); $\delta^{(l)}$ – ошибка (1.3) на l -м шаге; $\Delta w_i^{(l)}$ – коррекция веса на l -м шаге; $\eta \in (0, 1]$ – норма обучения.

Для наиболее популярных сигмоидальных ФА производная $df(\text{net}) / d \text{net}$ легко выражается через значения функции $f(\text{net})$.

В случае НС с пороговой ФА (1.1) коррекцию веса в (1.4) следует брать в виде

$$\Delta w_i^{(k)} = \eta \delta^{(l)} x_i^{(l)}. \tag{1.5}$$

При этом реальный выход совпадает с сетевым: $y = \text{out}$.

На каждой эпохе k суммарная квадратичная ошибка $E(k)$ равна расстоянию Хемминга между векторами целевого и реального выхода (размерности $N = 2^4 = 16$) по всем входным векторам x_1, x_2, x_3, x_4 .

Пример выполнения работы

Получим нейросетевую модель БФ (табл. 1.1)

$$F(x_1, x_2, x_3, x_4) = \overline{x_1 x_2 x_3 x_4}.$$

Табл. 1.1. Таблица истинности БФ

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
F	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0

На начальном шаге $l = 0$ (эпоха $k = 0$) весовые коэффициенты возьмем в виде

$$w_0^{(0)} = w_1^{(0)} = w_2^{(0)} = w_3^{(0)} = w_4^{(0)} = 0.$$

1. Используем пороговую ФА (1.1). Динамика НС (при норме обучения $\eta = 0.3$) представлена в табл. 1.2. График ошибки приведен на рис. 1.2.

Табл. 1.2. Параметры НС на последовательных эпохах (пороговая ФА)

Номер эпохи, k	Вектор весов W , выходной сигнал Y , суммарная ошибка E
0	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),$ $W = (0, 0, 0, 0, 0), \quad E = 13$
1	$Y = (0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (-0.6, -0.3, -0.6, 0.3, 0.3), \quad E = 2$
...	...
24	$Y = (0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$ $W = (-2.1, -1.2, -0.9, 1.8, 1.5), \quad E = 0$

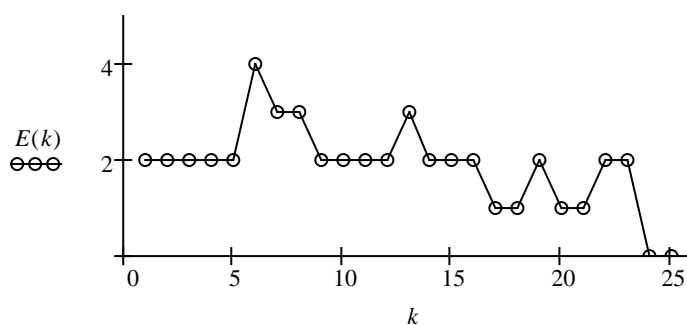


Рис. 1.2. График суммарной ошибки НС по эпохам обучения (пороговая ФА)

2. Используем сигмоидальную (логистическую) ФА

$$f(\text{net}) = \frac{1}{1 + \exp(-\text{net})},$$

производная которой выражается через значения самой функции как

$$\frac{df(\text{net})}{d \text{net}} = f(\text{net})(1 - f(\text{net})).$$

Для полного обучения (при нулевых начальных весах и норме обучения $\eta = 0.3$) потребовалось 34 эпохи (табл. 1.3). График ошибки приведен на рис. 1.3.

Табл. 1.3. Параметры НС на последовательных эпохах (логистическая ФА)

Номер эпохи, k	Вектор весов W , выходной вектор Y , суммарная ошибка E
0	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),$ $W = (0, 0, 0, 0, 0), \quad E = 13$
1	$Y = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (-0.302, -0.170, -0.174, -0.028, -0.033), \quad E = 3$
...	...
34	$Y = (0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0),$ $W = (-1.811, -0.918, -0.883, 1.379, 1.356), \quad E = 0$

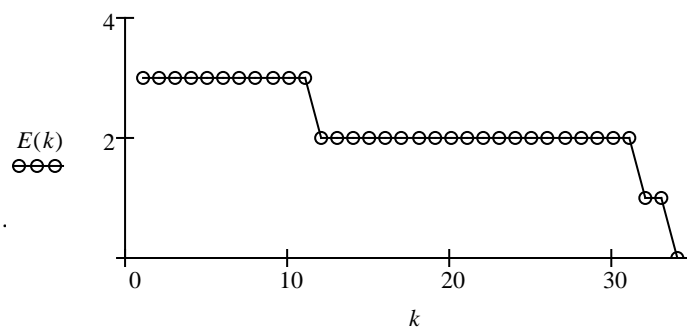


Рис. 1.3. График суммарной ошибки НС по эпохам обучения (логистическая ФА)

3. Рассматривая предыдущий случай, попытаемся последовательно уменьшать размер обучающей выборки, выбирая различные подмножества из 15, 14, 13 и т.д. обучающих векторов до тех пор, пока еще достижима нулевая ошибка. В рассматриваемом примере будет найден минимальный набор из пяти векторов

$$X^{(1)} = (0, 0, 0, 1), X^{(2)} = (0, 1, 1, 1), X^{(3)} = (1, 0, 1, 0), X^{(4)} = (1, 0, 1, 1), X^{(5)} = (1, 1, 1, 0),$$

дающий следующие синаптические коэффициенты:

$$W = (-0.225, -0.15, -0.075, 0.15, 0.224).$$

При этом для полного обучения потребовалось лишь 6 эпох.

Варианты работы

Табл. 1.4. Варианты БФ и ФА

№ пп	Моделируемая БФ	ФА*
1	$\overline{x_1 x_2 (x_3 + x_4)}$	1, 2
2	$(x_3 + x_4) \overline{x_1 + x_2}$	1, 3
3	$x_1 + x_2 + \overline{(x_3 + x_4)}$	1, 4
4	$(\overline{x_1 + x_3}) x_2 + x_2 x_4$	1, 2
5	$(\overline{x_1 + x_2 + x_3})(x_2 + x_3 + x_4)$	1, 3
6	$x_3 x_4 + \overline{x_1 + x_2}$	1, 4
7	$\overline{(x_1 + x_2)} + x_3 + x_4$	1, 2
8	$(x_1 + x_2 + x_4) x_3$	1, 3
9	$(x_1 + x_2 + x_3)(x_2 + x_3 + x_4)$	1, 4
10	$x_1 x_2 + x_3 + x_4$	1, 2
11	$(x_1 + x_2) x_3 x_4$	1, 3
12	$(x_1 + x_2) x_3 + x_4$	1, 4
13	$(\overline{x_1 + x_2 + x_3})(\overline{x_2 + x_3 + x_4})$	1, 2
14	$\overline{(x_1 + x_2)} + x_3 + x_4$	1, 3
15	$\overline{(x_1 + x_2) x_3 + x_3 x_4}$	1, 4
16	$\overline{(x_1 + x_2 + x_3)(x_2 + x_3 + x_4)}$	1, 2
17	$\overline{x_1 x_2 + x_3 + x_4}$	1, 3
18	$\overline{(x_1 + x_2) x_3 x_4}$	1, 4
19	$\overline{(x_1 + x_2) x_3 + x_4}$	1, 2
20	$\overline{x_3 x_4 (x_1 + x_2)}$	1, 3
21	$\overline{(x_1 + x_2) x_3 + x_4}$	1, 4
22	$\overline{x_3 + x_4 + (x_1 + x_2)}$	1, 2
23	$\overline{(x_2 + x_4) x_1 + x_1 x_3}$	1, 3
24	$\overline{(x_4 + x_3 + x_2)(x_3 + x_2 + x_1)}$	1, 4
25	$\overline{x_1 x_2 + x_3 + x_4}$	1, 2

* Функции активации:

$$\begin{aligned}
 1) f(\text{net}) &= \begin{cases} 1, & \text{net} \geq 0, \\ 0, & \text{net} < 0; \end{cases} & 3) f(\text{net}) &= \frac{1}{1 + \exp(-\text{net})}; \\
 2) f(\text{net}) &= \frac{1}{2} \left(\frac{\text{net}}{1 + |\text{net}|} + 1 \right); & 4) f(\text{net}) &= \frac{1}{2} (\tanh(\text{net}) + 1).
 \end{aligned}$$

Требования к отчету

Отчет должен содержать: титульный лист; цель работы; постановку задачи; таблицу истинности; график суммарной квадратичной ошибки в зависимости от эпохи обучения; целевые значения и реальный выход НС на каждой эпохе обучения; вектор синаптических коэффициентов на каждой эпохе обучения; минимальный набор обучающих векторов; выводы по результатам численного эксперимента.

Контрольные вопросы

1. Дайте определение персептрон и поясните алгоритм его функционирования.
2. Приведите функции активации НС и их производные.
3. Сформулируйте правило обучения Видроу-Хоффа.

2 ЛАБОРАТОРНАЯ РАБОТА № 2

Применение однослойной нейронной сети для решения задач регрессии экспериментальных данных

Цель работы

Изучение возможностей однослойных нейронных сетей с правилом обучения Видроу-Хоффа в задачах регрессии экспериментальных данных.

Постановка задачи и сведения из теории

Постановка задачи. В зависимости от варианта работы (табл. 2.1) с помощью НС с линейной функцией активации (рис. 2.1) аппроксимировать функцию $f(x)$ (найти коэффициенты c, d) по набору ее N дискретных значений, заданных равномерно на интервале $[a, b]$ со случайными ошибками $e_i = \text{Arnd}(-0.5; 0.5)$. Выполнить расчет параметров c, d методом наименьших квадратов (МНК) и определить среднеквадратичное отклонение (СКО). Провести обучение НС по правилу Видроу-Хоффа, экспериментальным путем подобрать оптимальный коэффициент обучения η .

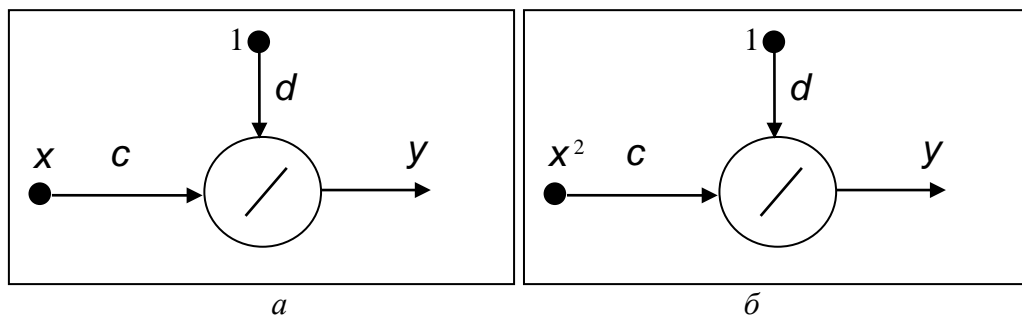


Рис. 2.1. Архитектура нейронной сети

Метод наименьших квадратов. Общее выражение для линейной регрессии имеет вид

$$y(x) = cx + d, \quad (2.1)$$

где параметры c, d должны быть выбраны таким образом, чтобы минимизировать сумму квадратов ошибок для всех точек набора экспериментальных данных (x_i, t_i) ($i = \overline{1, N}$):

$$E^2 = \sum_i [y(x_i) - t_i]^2 \rightarrow \min. \quad (2.2)$$

Подставляя (2.1) в (2.2) и приравнявая частные производные полученного выражения по c, d к нулю, получаем теоретические оценки коэффициентов регрессии:

$$\tilde{c} = \frac{N \sum_i x_i t_i - \sum_i x_i \sum_i t_i}{N \sum_i x_i^2 - \left(\sum_i x_i \right)^2}, \quad \tilde{d} = \frac{\sum_i t_i - \tilde{c} \sum_i x_i}{N}. \quad (2.3)$$

Если гипотеза о линейной зависимости между переменными x, y верна, необязательно ошибка (2.2) будет обращаться ноль, поскольку в процессе эксперимента реальные данные могут быть получены с погрешностями e_i :

$$t_i = cx_i + d + e_i. \quad (2.4)$$

При необходимости нахождения регрессии другого вида, например

$$y(x) = cx^2 + d, \quad (2.5)$$

можно воспользоваться заменой переменной $z = x^2$, и далее в (2.3) подставить вместо x_i значения $z_i = x_i^2$.

Нейросетевая регрессия. МНК дает нам уравнение прямой, наилучшим образом аппроксимирующей набор данных. Для нахождения такой прямой можно использовать другой подход, основанный на использовании нейронной сети с линейной функцией активации (см. рис. 2.1) и обучении, т.е. настройке ее весов c, d по некоторой выборке. Поскольку сеть однослойная, обучение можно реализовать по правилу Видроу–Хоффа.

Алгоритм функционирования НС с линейной ФА имеет вид:

$$\text{net} = cx + d; \quad y(\text{net}) = \text{net}. \quad (2.6)$$

где net – сетевой (комбинированный) вход; y – реальный выход НС.

Для необученной НС ее реальный выход $y_i = y(x_i)$ в общем случае отличается от целевого выхода t_i , представляющего собой данные, полученные с погрешностями, т.е. имеется ошибка

$$\delta_i = t_i - y_i \neq 0. \quad (2.7)$$

Каждая эпоха обучения $k = 1, 2, \dots$ включает в себя цикл последовательного предъявления всех образцов обучающей выборки, $i = \overline{1, N}$, на вход НС. Предъявление одного обучающего образца внутри эпохи является элементарным шагом обучения, во время которого вектор весовых коэффициентов $W = (c, d)$ корректируется согласно правилу Видроу-Хоффа (дельта-правило):

$$\begin{aligned} \hat{c}^{(i+1)} &= \hat{c}^{(i)} + \Delta \hat{c}^{(i)}, \quad \Delta \hat{c}^{(i)} = \eta \delta_i x_i, \\ \hat{d}^{(i+1)} &= \hat{d}^{(i)} + \Delta \hat{d}^{(i)}, \quad \Delta \hat{d}^{(i)} = \eta \delta_i, \end{aligned} \quad (2.8)$$

где $\eta \in (0, 1]$ – норма обучения.

Пример

Рассмотрим линейную регрессию (2.1) на интервале $a = 0, b = 3$ с известными параметрами

$$c = 0.5, \quad d = 2;$$

Набор данных содержит $N = 32$ отсчета.

В отсутствие шума ($A = 0$) МНК дает точные значения параметров регрессии (2.3):

$\tilde{c} = c, \tilde{d} = d$. В то же время нейронная сеть, обученная в течение $K = 17$ эпох с параметром $\eta = 0.05$, дает следующие результаты:

$$\hat{c} = 0.503, \quad \hat{d} = 1.991; \quad E^2 = 7.849 \cdot 10^{-4}$$

При наличии шума ($A = 0.5$) соответствующие результаты для МНК:

$$\tilde{c} = 0.485, \quad \tilde{d} = 2.024; \quad E^2 = 0.584;$$

для нейронной сети ($K = 50, \eta = 0.05$)

$$\tilde{c} = 0.496, \quad \tilde{d} = 2.021; \quad E^2 = 0.592.$$

Графики на рис. 2.2 иллюстрируют погрешности приближения в условиях шума (полужирная линия – точная зависимость, круглые маркеры – зашумленные отсчеты, тонкая сплошная линия – нейросетевая регрессия).

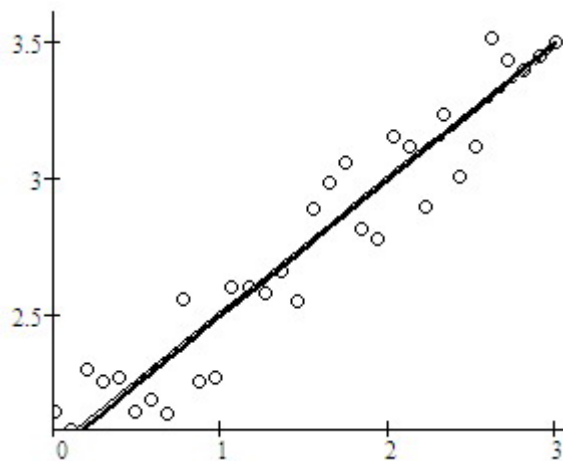


Рис. 2.2. Нейросетевая линейная регрессия экспериментальных данных

Варианты заданий

Табл. 2.1. Исходные данные

№ пп	$f(x)$	c	d	a	b	N	A
1	$cx + d$	3	1	-2	2	16	2
2	$cx^2 + d$	-4	2	-1	2	12	5
3	$cx + d$	0	3	-4	2	24	0.1
4	$cx^2 + d$	-0.5	0	-2	2	16	2
5	$cx + d$	-500	200	0	10	24	1000
6	$cx^2 + d$	0.5	0	-2	1	20	1
7	$cx + d$	8	0	-4	2	24	10
8	$cx^2 + d$	3	1	-1	3	16	3
9	$cx + d$	-4	2	-3	3	12	6
10	$cx^2 + d$	1	0	-2	2	24	2
11	$cx + d$	-10	0	-5	0	20	10
12	$cx^2 + d$	-1	3	0	3	10	3
13	$cx + d$	1000	0	-5	5	20	2000
14	$cx^2 + d$	-3	1	-3	0	8	3
15	$cx + d$	0.1	2	-5	0	32	0.2
16	$cx^2 + d$	-5	5	0	4	16	2
17	$cx + d$	-0.5	2	-2	1	20	0.4
18	$cx^2 + d$	-1	3	-1	2	8	1
19	$cx + d$	-3	-2	-2	2	20	3
20	$cx^2 + d$	1	3	-2	2	10	1
21	$cx + d$	-2	4	0	2	20	2
22	$cx^2 + d$	-3	3	-2	2	16	1.5
23	$cx + d$	1	3	1	2	16	0.2
24	$cx^2 + d$	2	2	-1	1	10	0.5
25	$cx + d$	-1	3	-1	2	8	2

Требования к отчету

Отчет должен содержать: титульный лист; цель работы; постановку задачи; результаты эксперимента (аппроксимируемая функция $f(x)$, зашумленная функция и их графики; график функции с параметрами, найденными по МНК, СКО; результат обучения – набор вычисленных синаптических весов \tilde{m}, \tilde{c} для функции без шума и с шумом, СКО, графики аппроксимирующей и аппроксимируемой функций); выводы по результатам численного эксперимента.

Контрольные вопросы

1. Объясните понятие регрессии экспериментальных данных.
2. Поясните суть метода наименьших квадратов.
3. Сформулируйте нейросетевой подход к задачам регрессии.

3 ЛАБОРАТОРНАЯ РАБОТА № 3

Применение однослойной нейронной сети с линейной функцией активации для прогнозирования временных рядов

Цель работы

Изучение возможностей однослойных нейронных сетей в задачах прогнозирования временных рядов методом скользящего окна (авторегрессия).

Постановка задачи и сведения из теории

Постановка задачи. На временном интервале $[a, b]$ задан дискретный набор значений функции $x(t)$ (см. табл. 3.1). Количество точек $N = 20$, расположение – равномерное. Методом «скользящего окна» спрогнозировать поведение функции $x(t)$ на N точках последующего интервала $(b, 2b - a]$. Для решения использовать однослойную нейронную сеть с количеством нейронов p и линейной функцией активации. Исходное количество нейронов (длина окна) $p = 4$. Обучение проводить методом Видроу–Хоффа (см. лаб. раб. 1, 2). Исследовать влияние количества эпох M обучения и коэффициента обучения η на среднеквадратичную погрешность приближения $\varepsilon = \sqrt{\sum_i [x(t_i) - \tilde{x}(t_i)]^2}$. Исследовать процесс прогнозирования при постепенном изменении (уменьшении/увеличении) количества нейронов p . Сделать выводы по результатам численного эксперимента.

Метод «скользящего окна». По временным рядом будем подразумевать функциональную зависимость $x = x(t)$, заданную на дискретном множестве отсчетов (моментов времени)

$$t = t_i \quad (i = \overline{1, m}),$$

так что

$$x_i = x(t_i) \quad (i = \overline{1, m}).$$

Одной из простейших моделей прогноза является авторегрессионная модель, когда прогнозируемое значение ряда в момент времени $n > m$ выражается через его известные значения в предыдущие моменты времени:

$$\tilde{x}_n = \sum_{k=1}^p w_k x_{n-p+k-1}, \quad (3.1)$$

где w_k – некоторые весовые коэффициенты; p – размер «окна» данных, по которому производится прогноз.

Выражение (3.1) часто используется для центрированных временных рядов, среднее значение которых равно 0. Если математическое ожидание временного ряда равно \bar{x} , то вместо (3.1) можно воспользоваться представлением

$$\tilde{x}_n = \sum_{k=1}^p w_k x_{n-p+k-1} + w_0, \quad (3.2)$$

где $w_0 = \bar{x}$.

Ошибка прогноза (локальная) равна апостериорной разнице спрогнозированного и реального значения временного ряда:

$$\delta_n = x_n - \tilde{x}_n. \quad (3.3)$$

Нейронная сеть прогноза. Нахождение неизвестных весовых коэффициентов w_k осуществляется по известной обучающей выборке значений временного ряда $x_i = x(t_i)$ ($i = \overline{1, m}$). Для этого необходимо определить окно длиной $p < m$. Далее, начиная с левого края временного ряда следует «прогнозировать» его значения в моменты, идущие непосредственно за окном (справа от него). Сравнивая этот «прогноз» с реальными значениями, можно оценить ошибку (3.3), а на ее основе – скорректировать весовые коэффициенты w_k , например, с помощью методов обучения нейронных сетей.

Простейшая архитектура нейронной сети показана на рис. 3.1. Функция активации – линейная, т.е.

$$f(\text{net}) = \text{net},$$

где net – комбинированный вход единственного нейрона:

$$\text{net} = \sum_{k=1}^p w_k x_{i+k-1} + w_0.$$

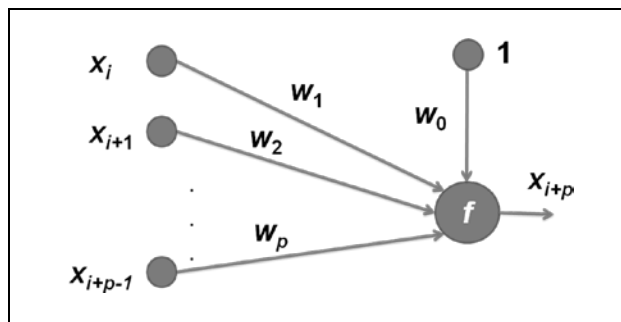


Рис. 3.1. Архитектура нейронной сети

В режиме обучения последовательно, от эпохи к эпохе, на вход сети подаются элементы векторов-столбцов обучающей выборки

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots & x_{m-p} \\ x_2 & x_3 & \dots & x_{m-p+1} \\ \dots & \dots & \dots & \dots \\ x_p & x_{p+1} & \dots & x_{m-1} \end{pmatrix}, \quad (3.4)$$

а на выходе соответственно получаются «прогнозные» значения

$$\tilde{x}_{p+1}, \tilde{x}_{p+2}, \dots, \tilde{x}_{m-1}.$$

Эти значения сравниваются с реальными

$$x_{p+1}, x_{p+2}, \dots, x_{m-1},$$

и по (3.3) оценивается ошибка. Коррекция весов на каждой эпохе производится по правилу Видроу–Хоффа:

$$\Delta w_k = \eta \delta x_k, \quad k = \overline{0, p},$$

где $\eta \in (0, 1]$ – норма обучения.

Если по достижении правого края выборки (3.4) суммарная среднеквадратичная ошибка $\varepsilon = \sqrt{\sum_i [x(t_i) - \tilde{x}(t_i)]^2}$ останется достаточно большой, следует продолжить обучение, снова вернувшись к первому столбцу (3.4) и т.д.

В рабочем режиме прогноз осуществляется путем сдвига окна далее вправо на 1, 2, 3 и т.д. отсчетов относительно последнего столбца (3.4). При этом, начиная со сдвига на 2, в прогнозе будут принимать значения, ранее в свою очередь спрогнозированные, что может привести к росту ошибки «долговременного» прогноза.

Пример выполнения работы

Рассмотрим прогноз функции $0.5 \sin 0.5t - 0.5$ по 20-ти равноотстоящим исходным значениям, заданным на интервале $t \in [-2, 4]$ (рис. 3.2). Выберем длину окна $p = 6$, норму обучения $\eta = 1$. Без ограничения общности будем считать вес смещения $w_0 = 0$. На рис. 3.3 показан график самой функции и ее прогноз (круглые маркеры) на интервале $t \in [5, 11]$ при различном количестве эпох обучения M .

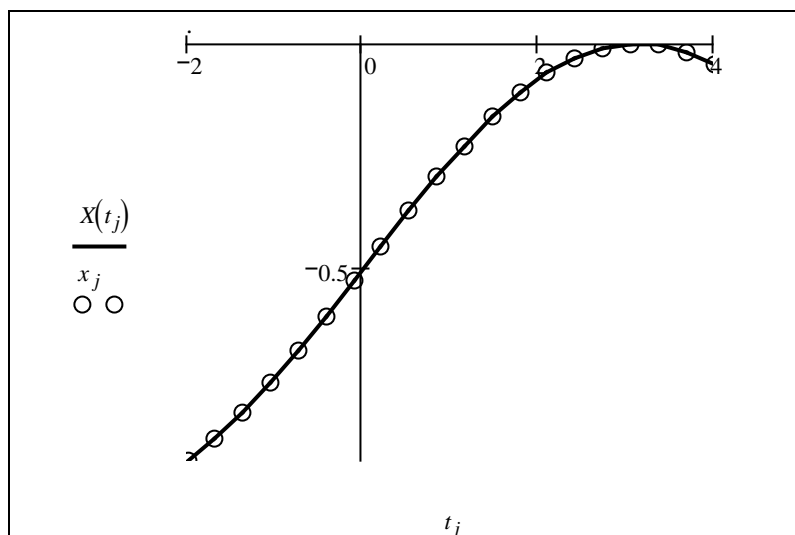
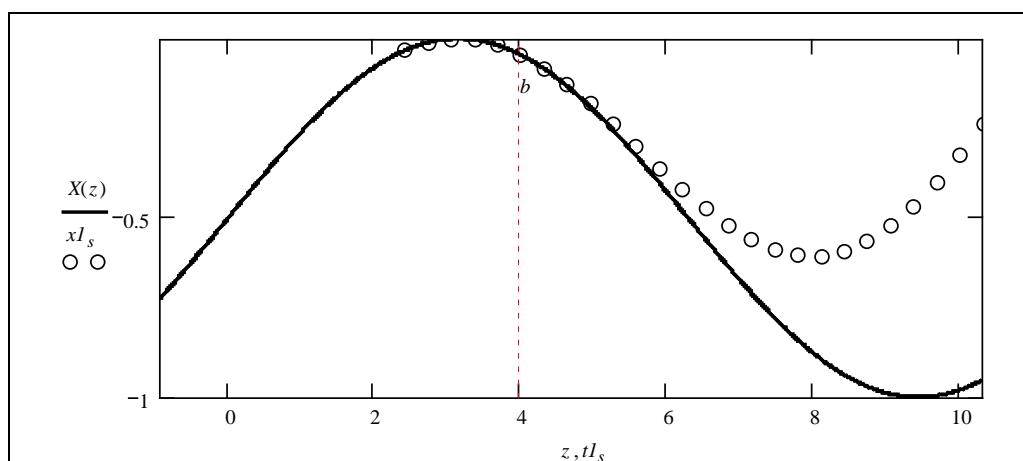
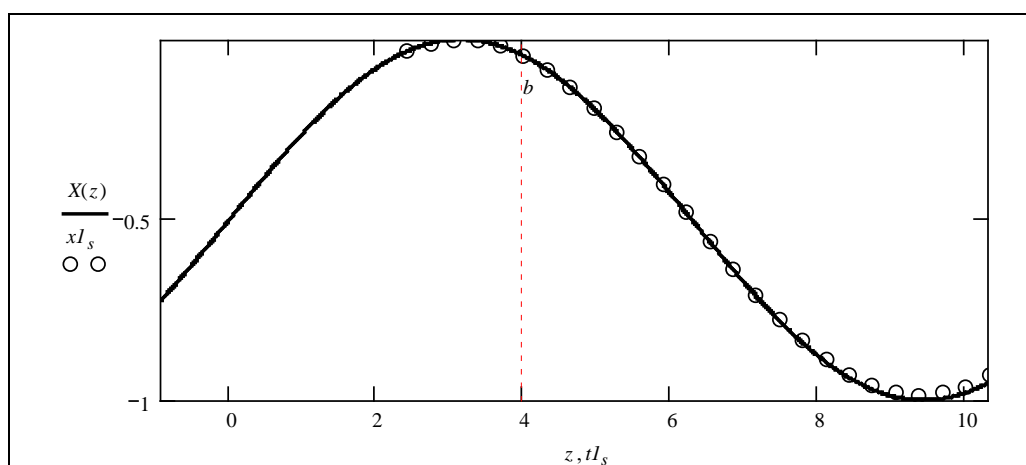


Рис. 3.2. Исходная функция



а)



б)

Рис. 3.3. Исходная функция и ее прогноз при различном количестве эпох обучения:
а) $M = 2000$ ($\varepsilon \approx 1.515$); б) $M = 4000$ ($\varepsilon \approx 0.040$)

Вектор весовых коэффициентов при $M = 4000$ равен

$$\mathbf{W} = (0 \ 0.503 \ -0.289 \ -0.590 \ -0.392 \ 0.300 \ 1.468).$$

Сравнивая результаты прогноза при различном количестве эпох, следует отметить, что его качество неудовлетворительно примерно до $M \approx 3500$, а затем быстро улучшается и при $M > 4000$ прогнозные значения практически совпадают с точными в пределах графического изображения.

Варианты заданий

Табл. 3.1. Варианты функций

№пп	Функция $x(t)$	a	b
1	$0.5 \cos 0.5t - 0.5$	-5	5
2	$t^4 - 2t^3 + t$	-0.5	0.5
3	$\exp(t - 1)$	-2	2
4	$0.5 \exp(0.5 \cos 0.5t) + \sin 0.5t$	-5	3
5	$\sqrt{\lg(-t)}$	1.7	2
6	$\sqrt{0.1t} + 1$	1	5
7	$0.4 \sin 0.3t + 0.5$	-4	4
8	$\sin^2 t$	0	2
9	$\exp(t - 2) - \sin t$	-1	2
10	$\sin(t - 1)$	-2	2
11	$\lg t$	2	3
12	$\exp(-0.1t^2)$	-5	5
13	$\sin(0.1t^3 - 0.2t^2 + t - 1)$	0	1
14	$\sin(2\sqrt{\exp t})$	4.5	5
15	$0.5 \cos 0.5t - \sin t$	0	4
16	$\ln t - 1$	1	5
17	$\cos^2 t - 0.5$	-1	0.5
18	$\cos t$	1	1.5
19	t^2	-1	1
20	$t^2 \sin t$	-1	1
21	$0.1 \sin 3t + \cos t$	1	1.5
22	$0.2 \sin 4t$	1.2	1.5
23	$0.1 \cos 3t - 0.5$	-1	0
24	$\lg 3t$	-0.1	0
25	$t^2 \exp(t)$	-0.1	0.3

Требования к отчету

Отчет должен содержать: титульный лист; цель работы; постановку задачи; результаты эксперимента (в виде графиков функций, зависимостей погрешности от числа эпох и нормы обучения); выводы по результатам численного эксперимента.

Контрольные вопросы

1. В чем состоит принцип прогнозирования на основе авторегрессии?
2. Объясните методику обучения нейронной сети прогноза.
3. Поясните принцип функционирования нейронной сети прогноза.

4 Лабораторная работа № 4

Исследование нейронных сетей с радиальными базисными функциями (RBF) на примере моделирования булевых выражений

Цель работы

Исследовать функционирование нейронной сети (НС) с радиальными базисными функциями (RBF) и ее обучение по правилу Видроу-Хоффа.

Постановка задачи и сведения из теории

Постановка задачи. Получить модель булевой функции (БФ) на основе RBF-НС с двоичными входами $x_1, x_2, x_3, x_4 \in \{0,1\}$, единичным входом смещения $\varphi_0 = 1$, синаптическими весами v_0, v_1, v_2, v_3, v_4 , двоичным выходом $y \in \{0,1\}$ с пороговой ФА выходного нейрона, J скрытыми RBF-нейронами с гауссовой ФА $\varphi: R \rightarrow (0,1]$ и координатами центров $c_{j1}, c_{j2}, c_{j3}, c_{j4}$ ($j = \overline{1, J}$) (рис. 4.1).

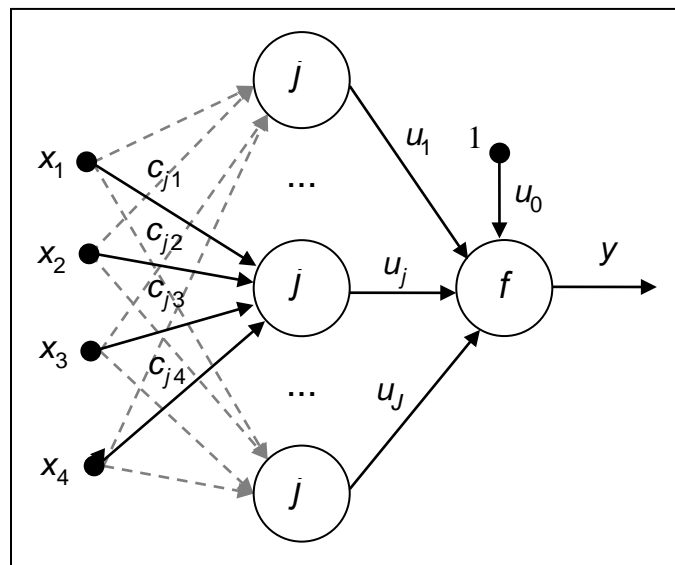


Рис. 4.1. Нейронная сеть RBF

Для заданной БФ (варианты см. в табл. 4.1) количество RBF-нейронов необходимо выбирать из соотношения $J = \min\{J_0, J_1\}$, где J_0, J_1 – количество векторов $X = (x_1, x_2, x_3, x_4)$, соответствующих значениям БФ “0” и “1” соответственно. Центры RBF $C^{(j)} = (c_{j1}, c_{j2}, c_{j3}, c_{j4})$ должны совпадать с концами этих векторов.

Требуется найти минимальный набор векторов X , используемых для обучения.

Рабочий режим НС. Алгоритм функционирования НС с гауссовой RBF имеет вид:

$$\begin{aligned}
\varphi_j(X) &= \exp\left(-\sum_{i=1}^4 (x_i - c_{ji})^2\right), \quad j = \overline{1, J}; \\
\text{net} &= \sum_{j=1}^J v_j \varphi_j(X) + v_0; \\
y(\text{net}) &= \begin{cases} 1, & \text{net} \geq 0, \\ 0, & \text{net} < 0, \end{cases}
\end{aligned} \tag{4.1}$$

где net – сетевой (комбинированный) вход; y – реальный выход НС.

Режим обучения НС. Для необученной НС ее реальный выход y в общем случае отличается от целевого выхода t , представляющего собой значения заданной БФ нескольких переменных $F(x_1, x_2, x_3, x_4) : \{0, 1\}^4 \rightarrow \{0, 1\}$, т.е. имеется хотя бы один набор сигналов (x_1, x_2, x_3, x_4) , для которого ошибка

$$\delta = t - y \neq 0. \tag{4.2}$$

Каждая эпоха обучения $k = 1, 2, \dots$ включает в себя цикл последовательного предъявления всех образцов обучающей выборки на вход НС. Предъявление одного обучающего образца внутри эпохи является элементарным шагом обучения $l = 0, 1, 2, \dots$, во время которого вектор весовых коэффициентов $V = (v_j)_{j=0, \dots, J}$ корректируется согласно правилу Видроу-Хоффа (дельта-правило):

$$\begin{aligned}
v_j^{(l+1)} &= v_j^{(l)} + \Delta v_j^{(l)}, \\
\Delta v_j^{(l)} &= \eta \delta^{(l)} \varphi_j^{(l)}(X),
\end{aligned} \tag{4.3}$$

где $\varphi_j^{(l)}$ ($j = \overline{0, J}$) – выходы RBF-нейронов на l -м шаге обучения (при этом $\varphi_0^{(l)} \equiv 1$); $\delta^{(l)}$ – ошибка (4.2) на l -м шаге; $\Delta v_j^{(l)}$ – коррекция веса на l -м шаге; $\eta \in (0, 1]$ – норма обучения.

На каждой эпохе k суммарная квадратичная ошибка $E(k)$ равна расстоянию Хемминга между векторами целевого и реального выхода (размерности $N = 2^4 = 16$) по всем входным векторам x_1, x_2, x_3, x_4 .

Пример выполнения работы

Построим RBF-модель для БФ из примера в лаб. раб. 1 (см. табл. 1.1)

$$F(x_1, x_2, x_3, x_4) = \overline{x_1 x_2 x_3 x_4}.$$

На начальном шаге $l = 0$ (эпоха $k = 0$) весовые коэффициенты возьмем в виде

$$v_0^{(0)} = v_1^{(0)} = v_2^{(0)} = v_3^{(0)} = v_4^{(0)} = 0,$$

а норму обучения: $\eta = 0.3$.

Находим количество RBF-нейронов: $J = 3$. Центры RBF-нейронов располагаем в точках

$$C^{(1)} = (0, 0, 1, 1), \quad C^{(2)} = (0, 1, 1, 1), \quad C^{(3)} = (1, 0, 1, 1).$$

В качестве обучающей выборки можно взять минимальный набор из пяти векторов, найденный в лаб. раб. 1:

$$X^{(1)} = (0, 0, 0, 1), \quad X^{(2)} = (0, 1, 1, 1), \quad X^{(3)} = (1, 0, 1, 0), \quad X^{(4)} = (1, 0, 1, 1), \quad X^{(5)} = (1, 1, 1, 0).$$

В итоге цикл полного обучения занял 85 эпох (рис. 4.2), а конечные значения синаптических весов имеют вид:

$$V = (16.5, -2.592, -2.158, -0.506).$$

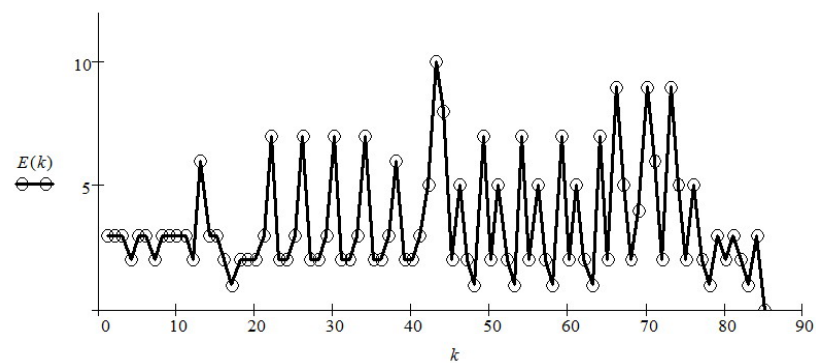


Рис. 4.2. График суммарной ошибки НС по эпохам обучения

Варианты работы

Табл. 4.1. Варианты БФ и ФА

№ пп	Моделируемая БФ	ФА*
1	$\overline{x_1 x_2 (x_3 + x_4)}$	1, 2
2	$(\overline{x_3 + x_4}) \overline{x_1} + x_2$	1, 3
3	$x_1 + x_2 + (\overline{x_3 + x_4})$	1, 4
4	$(\overline{x_1 + x_3}) x_2 + x_2 x_4$	1, 2
5	$(\overline{x_1 + x_2 + x_3})(\overline{x_2 + x_3 + x_4})$	1, 3
6	$x_3 x_4 + \overline{x_1} + x_2$	1, 4
7	$(\overline{x_1 + x_2}) + x_3 + x_4$	1, 2
8	$(x_1 + x_2 + x_4) x_3$	1, 3
9	$(x_1 + x_2 + x_3)(x_2 + x_3 + x_4)$	1, 4
10	$x_1 x_2 + x_3 + x_4$	1, 2
11	$(x_1 + x_2) x_3 x_4$	1, 3
12	$(x_1 + x_2) x_3 + x_4$	1, 4
13	$(\overline{x_1 + x_2 + x_3})(\overline{x_2 + x_3 + x_4})$	1, 2

14	$\overline{\overline{(x_1 + x_2)} + x_3 + x_4}$	1, 3
15	$\overline{(x_1 + x_2)x_3 + x_3x_4}$	1, 4
16	$\overline{(x_1 + x_2 + x_3)(x_2 + x_3 + x_4)}$	1, 2
17	$\overline{x_1x_2 + x_3 + x_4}$	1, 3
18	$\overline{(x_1 + x_2)x_3x_4}$	1, 4
19	$\overline{(x_1 + x_2)x_3 + x_4}$	1, 2
20	$\overline{x_3x_4(x_1 + x_2)}$	1, 3
21	$\overline{(x_1 + x_2)x_3 + x_4}$	1, 4
22	$\overline{x_3 + x_4 + (x_1 + x_2)}$	1, 2
23	$\overline{(x_2 + x_4)x_1 + x_1x_3}$	1, 3
24	$\overline{(x_4 + x_3 + x_2)(x_3 + x_2 + x_1)}$	1, 4
25	$\overline{x_1x_2 + x_3 + x_4}$	1, 2

Требования к отчету

Отчет должен содержать: титульный лист; цель работы; постановку задачи; таблицу истинности; параметры RBF; набор обучающих векторов; график суммарной квадратичной ошибки в зависимости от эпохи обучения; вектор найденных синаптических коэффициентов; выводы по результатам численного эксперимента.

Контрольные вопросы

1. Нейронная сеть RBF, алгоритм функционирования.
2. Типы радиальных базисных функций.
3. Нахождение параметров и обучение НС RBF.

5 Лабораторная работа № 5

Исследование аппроксимативных свойств нейронных сетей с радиальными базисными функциями (RBF)

Цель работы

Исследовать аппроксимацию одномерной функции с помощью нейронной сети (НС) с радиальными базисными функциями (RBF).

Постановка задачи и сведения из теории

Постановка задачи. Аппроксимировать функцию $F(t)$ на интервале $[a, b]$ (см. табл. 3.1) с помощью сети с гауссовыми RBF:

$$\varphi(t) = \exp\left[-\frac{1}{2\sigma^2}(c-t)^2\right]. \quad (5.1)$$

Архитектура сети (рис. 5.1): 1 входной нейрон; J нейронов слоя RBF (центры $c_j = a + j(b-a)/(J+1)$, $j = \overline{1, J}$); 1 выходной нейрон (линейная функция активации). Размер обучающей выборки Q . Весовые коэффициенты v_j определить по правилу Видроу–Хоффа. Исследовать влияние ширины RBF σ и коэффициента обучения $\eta \in (0, 1]$ на погрешность приближения $E = \sqrt{\sum_i (f(t_i) - y_i)^2}$, $i = \overline{1, N}$ ($N \gg J$) при количестве эпох обучения $K \gg 1$.

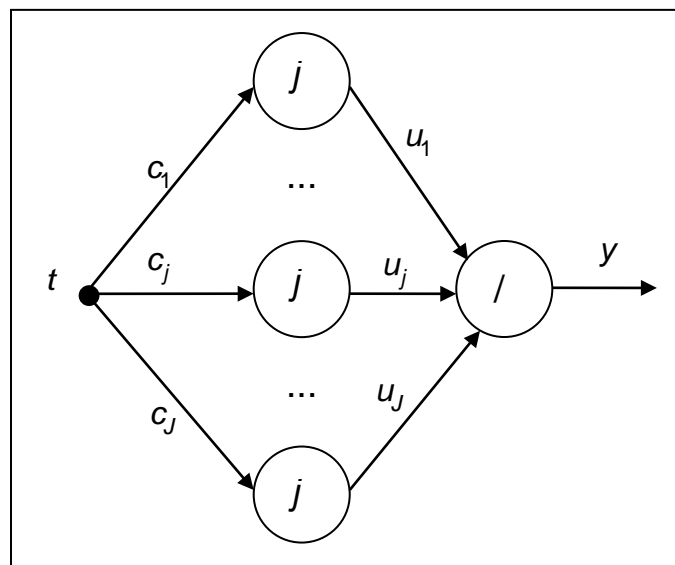


Рис. 5.1. Архитектура нейронной сети RBF

Рабочий режим НС. Алгоритм функционирования НС с RBF (5.1) имеет вид:

$$\begin{aligned}\varphi_j(t) &= \exp\left[-\frac{1}{2\sigma^2}(c_j - t)^2\right], \quad j = \overline{1, J}; \\ \text{net}(t) &= \sum_{j=1}^J v_j \varphi_j(t); \quad y(\text{net}) = \text{net}.\end{aligned}\quad (5.2)$$

где net – сетевой (комбинированный) вход; y – реальный выход НС.

Режим обучения НС. Для необученной НС ее реальный выход y в общем случае отличается от целевого выхода, представляющего собой значения заданной функции $f(t)$, т.е. имеется ошибка

$$\delta(t) = f(t) - y \neq 0. \quad (5.3)$$

Каждая эпоха обучения $k = 1, 2, \dots$ включает в себя цикл последовательного предъявления образцов обучающей выборки на вход НС. Предъявление одного обучающего образца t_i ($i = \overline{1, N}$) внутри эпохи является элементарным шагом обучения $l = 1, 2, 3, \dots$, во время которого вектор весовых коэффициентов $V = (v_j)_{j=\overline{1, J}}$ корректируется согласно правилу Видроу-Хоффа (дельта-правило):

$$\begin{aligned}v_j^{(l+1)} &= v_j^{(l)} + \Delta v_j^{(l)}, \\ \Delta v_j^{(l)} &= \eta \delta^{(l)} \varphi_j^{(l)},\end{aligned}$$

где $\varphi_j^{(l)}$ ($j = \overline{1, J}$) – выходы RBF-нейронов на l -м шаге обучения; $\delta^{(l)}$ – ошибка (5.3) на l -м шаге; $\Delta v_j^{(l)}$ – коррекция веса на l -м шаге; $\eta \in (0, 1]$ – норма обучения.

Пример выполнения работы

Выполним аппроксимацию функции

$$\exp(t - 2) + \cos 2t,$$

заданной на интервале с концами $a = -1, b = 2$.

Размер обучающей выборки $Q = 20$. Аппроксимация выполняется с помощью $J = 9$ нейронов RBF. На рис. 5.2 показаны графики исходной функции (полужирный), точек обучающей выборки (круглые маркеры) и аппроксимирующей кривой (5.2) с параметрами $\eta = 0.2$, $\sigma = 0.5$. Количество эпох обучения $K = 50$, результирующая среднеквадратичная ошибка $E \approx 0.456$. Вектор весовых коэффициентов:

$$\mathbf{v} = (-1.326 \ 0.224 \ 1.201 \ 1.150 \ 0.278 \ -0.694 \ -0.993 \ -0.312 \ 0.987).$$

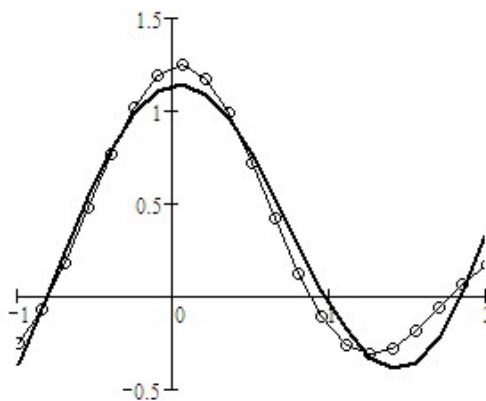


Рис. 5.2. Аппроксимация функции сетью RBF

Варианты заданий

См. табл. 3.1.

Требования к отчету

Отчет должен содержать: титульный лист; цель работы; постановку задачи; результаты эксперимента (в виде графиков функций, зависимостей погрешности от числа эпох и нормы обучения); вектор найденных синаптических коэффициентов; выводы по результатам численного эксперимента.

Контрольные вопросы

1. Нейронная сеть RBF, алгоритм функционирования.
2. Типы радиальных базисных функций.
3. Нахождение параметров и обучение НС RBF.

6 Лабораторная работа № 6

Изучение алгоритма обратного распространения ошибки (метод Back Propagation)

Цель работы

Исследовать функционирование многослойной нейронной сети (МНС) прямого распространения и ее обучение методом обратного распространения ошибки (англ. Back Propagation – BP).

Постановка задачи и сведения из теории

Постановка задачи. На примере МНС архитектуры $N - J - M$ (рис. 6.1) реализовать ее обучение методом BP, проведя настройку весов нейронов скрытого ($w_{ij}^{(1)}(k)$, $i = \overline{0, N}$, $j = \overline{1, J}$) и выходного ($w_{jm}^{(2)}(k)$, $j = \overline{0, J}$, $m = \overline{1, M}$) слоев, где индексы $i, j = 0$ соответствуют нейронам смещения; $k = 1, 2, \dots$ – номер эпохи обучения.

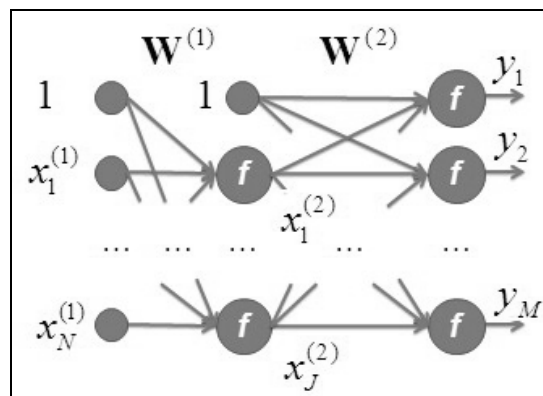


Рис. 6.1. Многослойная НС

Алгоритм обратного распространения ошибки.

Обозначения:

$x_i^{(1)}, x_j^{(2)}(k)$ – входные сигналы нейронов скрытого и выходного слоев;

$net_j^{(1)}(k), net_m^{(2)}(k)$ – комбинированные входы нейронов скрытого и выходного слоев;

$out_j^{(1)}(k), out_m^{(2)}(k)$ – выходные сигналы нейронов скрытого и выходного слоев;

$\delta_j^{(1)}(k), \delta_m^{(2)}(k)$ – ошибки скрытого и выходного слоев.

Начальные веса $w_{ij}^{(1)}(0), w_{jm}^{(2)}(0)$ принять произвольными.

Функция активации нейронов скрытого и выходного слоев:

$$f(\text{net}) = \frac{1 - \exp(-\text{net})}{1 + \exp(-\text{net})} \in (-1, 1).$$

Ее производная выражается через значения самой функции как

$$\frac{d f(\text{net})}{d \text{net}} = \frac{1}{2} [1 - f^2(\text{net})].$$

На первом этапе следует рассчитать по заданному входному сигналу x_i , $i = \overline{0, N}$, выход МНС $y_m(k)$:

$$\text{I.1 } x_i^{(1)} \equiv x_i, \quad i = \overline{0, N};$$

$$\text{I.2 } \text{net}_j^{(1)}(k) = \sum_{i=1}^N w_{ij}^{(1)}(k) x_i^{(1)} + w_{0j}^{(1)}(k), \quad j = \overline{1, J};$$

$$\text{I.3 } x_j^{(2)}(k) \equiv \text{out}_j^{(1)}(k) = f[\text{net}_j^{(1)}(k)], \quad j = \overline{1, J};$$

$$\text{I.4 } \text{net}_m^{(2)}(k) = \sum_{j=1}^J w_{jm}^{(2)}(k) x_j^{(2)}(k) + w_{0m}^{(2)}(k), \quad m = \overline{1, M};$$

$$\text{I.5 } y_m(k) \equiv \text{out}_m^{(2)}(k) = f[\text{net}_m^{(2)}(k)], \quad m = \overline{1, M};$$

На втором этапе по известному желаемому выходу t_m («обучение с учителем») оцениваем ошибки нейронов выходного и скрытого слоев (обратное распространение ошибки):

$$\text{II.1 } \delta_m(k) \equiv \delta_m^{(2)}(k) = \frac{d f[\text{net}_m^{(2)}(k)]}{d \text{net}_m^{(2)}(k)} [t_m - y_m(k)], \quad m = \overline{1, M};$$

$$\text{II.2 } \delta_j^{(1)}(k) = \frac{d f[\text{net}_j^{(1)}(k)]}{d \text{net}_j^{(1)}(k)} \sum_{m=1}^M w_{jm}^{(2)}(k) \delta_m(k), \quad j = \overline{1, J}.$$

После этого производится настройка весов

$$\text{III.1 } w_{ij}^{(1)}(k+1) = w_{ij}^{(1)}(k) + \Delta w_{ij}^{(1)}(k), \quad \Delta w_{ij}^{(1)}(k) = \eta x_i \delta_j^{(1)}(k);$$

$$\text{III.2 } w_{jm}^{(2)}(k+1) = w_{jm}^{(2)}(k) + \Delta w_{jm}^{(2)}(k), \quad \Delta w_{jm}^{(2)}(k) = \eta x_j^{(2)}(k) \delta_m(k).$$

Норму обучения следует принять $\eta \in (0, 1]$.

Далее по тому же входному сигналу x_i выполняется расчет нового выходного вектора $y_j(k+1)$ и далее аналогично вплоть до достижения эпохи $k = K$, на которой суммарная среднеквадратичная ошибка не превысит некоторого порога $0 < \varepsilon \ll 1$:

$$E(k) = \sqrt{\sum_{j=1}^M [t_j - y_j(k)]^2} \leq \varepsilon.$$

Пример выполнения работы

Выберем в качестве примера МНС архитектуру 3–3–4 ($N = 3$, $J = 3$, $M = 4$).

Пусть требуется обучить МНС на восстановление по входному вектору

$$\mathbf{x} = (1 \ 0.3 \ -0.1 \ 0.9)$$

целевого вектора

$$\mathbf{t} = (0.1 \ -0.6 \ 0.2 \ 0.7)$$

с погрешностью не больше $\varepsilon = 1 \cdot 10^{-3}$.

Исходные веса примем нулевыми:

$$w_{ij}^{(1)}(0) = 0, \ i = \overline{0, N}, \ j = \overline{1, J};$$

$$w_{jm}^{(2)}(0) = 0, \ j = \overline{0, J}, \ m = \overline{1, M}.$$

В табл. 6.1 приведены некоторые результаты обучения МНС методом ВР при значении параметра $\eta = 1$.

Табл. 6.1. Параметры НС на последовательных эпохах

Номер эпохи, k	Выходной сигнал \mathbf{y} , суммарная ошибка $E(k)$
0	$\mathbf{y} = (0 \ 0 \ 0 \ 0)$, $E(0) = 0.949$
1	$\mathbf{y} = (0.025 \ -0.149 \ 0.050 \ 0.173)$, $E(1) = 0.714$
...	...
$K = 73$	$\mathbf{y} = (0.100 \ -0.600 \ 0.200 \ 0.699)$, $E(73) = 9.816 \cdot 10^{-4}$

Требования к отчету

Отчет должен содержать: титульный лист; цель работы; постановку задачи; график или таблицу суммарной квадратичной ошибки в зависимости от эпохи обучения; целевые значения и реальный выход НС на каждой эпохе обучения; наборы синаптических коэффициентов (весов) на каждой эпохе обучения; выводы по результатам численного эксперимента.

Варианты заданий

Табл. 6.2. Варианты многослойных нейронных сетей

№пп	Архитектура	\mathbf{x}	$10\mathbf{t}$	№пп	Архитектура	\mathbf{x}	$10\mathbf{t}$
1	1–2–1	(1 2)	4	14	1–2–1	(1 2)	–3
2	1–1–3	(1 –1)	(2 –3 1)	15	1–1–3	(1 –3)	(1 1 1)
3	2–1–2	(1 2 1)	(2 1)	16	2–1–2	(1 2 1)	(–1 –1)
4	1–2–1	(1 3)	1	17	1–2–1	(1 2)	3
5	1–1–3	(1 2)	(–1 0 2)	18	1–1–3	(1 –1)	(–1 2 2)

6	2 – 1 – 2	(1 1 –1)	(2 –1)
7	1 – 2 – 1	(1 4)	–2
8	1 – 1 – 3	(1 –2)	(2 1 3)
9	2 – 1 – 2	(1 2 2)	(3 1)
10	1 – 2 – 1	(1 3)	–4
11	1 – 1 – 3	(1 –3)	(–3 1 1)
12	2 – 1 – 2	(1 1 2)	(2 2)
13	1 – 2 – 1	(1 –3)	–1

19	2 – 1 – 2	(1 2 –1)	(2 –2)
20	1 – 2 – 1	(1 –3)	1
21	1 – 1 – 3	(1 –1)	(–3 2 1)
22	2 – 1 – 2	(1 2 3)	(1 2)
23	1 – 2 – 1	(1 3)	4
24	1 – 1 – 3	(1 –2)	(–1 2 3)
25	2 – 1 – 2	(1 2 2)	(2 3)
26	1 – 2 – 1	(1 –2)	3

Контрольные вопросы

1. Дайте определение многослойной нейронной сети и объясните ее принципиальное отличие от однослойной с точки зрения нелинейной классификации.
2. Объясните основную идею метода обратного распространения ошибки.
3. Сформулируйте теорему Колмогорова об аппроксимации.

7 Лабораторная работа № 7

Исследование рекуррентной нейронной сети Хопфилда на примере задачи распознавания образов

Цель работы

Исследовать процедуры обучения и функционирования рекуррентной нейронной сети (РНС) Хопфилда в качестве устройства автоассоциативной памяти.

Постановка задачи и сведения из теории

Постановка задачи. Закодировать запоминаемые образы (табл. 7.1) в виде биполярных матриц-паттернов размерности $I \times J$ (рекомендуемое число пикселей $I \cdot J \approx 20...40$). Произвести векторизацию матриц. Провести настройку весов РНС Хопфилда (рис. 7.1) согласно правилу ассоциативного обучения (Хебба). Задать функцию активации и реализовать алгоритм функционирования РНС Хопфилда в синхронном или асинхронном режиме. Протестировать РНС на запомненных эталонных образах. Проверить функционирование РНС Хопфилда на искаженных паттернах (изменены порядка 10% пикселей).

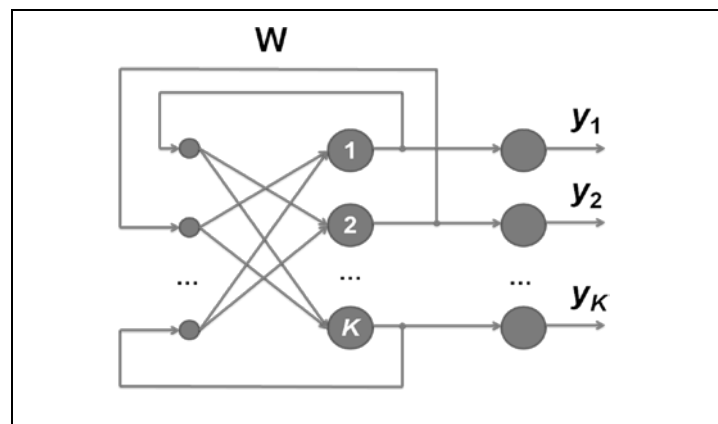


Рис. 7.1. Рекуррентная нейронная сеть Хопфилда

Рабочий режим. Рекуррентная нейронная сеть Хопфилда является автоассоциативной памятью, которая в ответ на исходное входное воздействие-сигнал

$$X = (x_1, x_2, \dots, x_K), x_k \in \{-1, 1\}, k = 1, 2, \dots, K,$$

формирует отклик

$$Y = (y_1, y_2, \dots, y_K), y_k \in \{-1, 1\}, k = 1, 2, \dots, K,$$

структурно соответствующий прототипу.

В синхронном режиме каждая эпоха с номером $n=1,2,\dots$ включает в себя следующие вычисления:

$$\text{net}_k^{(n)} = \sum_{\substack{j=1 \\ (j \neq k)}}^K w_{jk} y_j^{(n-1)}, \quad y_k^{(n)} = f(\text{net}_k^{(n)}), \quad k=1,2,\dots,K. \quad (7.1)$$

Здесь функция активации каждого нейрона

$$f(\text{net}_k^{(n)}) = \begin{cases} 1, & \text{net}_k^{(n)} > 0, \\ f(\text{net}_k^{(n-1)}), & \text{net}_k^{(n)} = 0, \\ -1, & \text{net}_k^{(n)} < 0. \end{cases}$$

В асинхронном режиме вместо (Н.1) имеем

$$\text{net}_k^{(n)} = \sum_{j=1}^{k-1} w_{jk} y_j^{(n)} + \sum_{j=k+1}^K w_{jk} y_j^{(n-1)}, \quad y_k^{(n)} = f(\text{net}_k^{(n)}), \quad k=1,2,\dots,K. \quad (7.2)$$

На каждой эпохе вместо прямого порядка (7.2) можно реализовать любой другой, в том числе случайный, порядок обновления нейронов.

Инициализация. Для начала работы РНС Хопфилда необходимо задать начальные условия:

$$y_k^{(0)} = x_k, \quad k=1,2,\dots,K,$$

а также вычислить компоненты матрицы весов:

$$w_{jk} = \begin{cases} \sum_{l=1}^L x_j^{(l)} x_k^{(l)}, & j \neq k, \\ 0, & j = k, \end{cases} \quad (7.3)$$

где l – номера запоминаемых образов-паттернов, а L – «емкость» ассоциативной памяти (количество запоминаемых образов).

Пример выполнения работы

Пусть требуется запомнить 3 образа, представляющих собой графические изображения цифр «1», «2», «3». В качестве примера закодируем их в виде биполярных матриц-паттернов размерности 5×3 (рис. 7.2). Векторизуем матрицы по столбцам для получения рабочих векторов длины 15:

$$X^{(1)} = (-1, 1, -1, -1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, 1),$$

$$X^{(2)} = (1, -1, 1, 1, 1, 1, -1, 1, -1, 1, 1, 1, 1, -1, 1),$$

$$X^{(3)} = (1, -1, 1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1).$$

«1»

-1	1	-1
1	1	-1
-1	1	-1
-1	1	-1

«2»

1	1	1
-1	-1	1
1	1	1
1	-1	-1

«3»

1	1	1
-1	-1	1
1	1	1
-1	-1	1

1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---

Рис. 7.2. Биполярный код образов

Настроим веса РНС Хопфилда, записав (7.3) в векторно-матричном виде:

$$W = \left(\sum_{l=1}^3 X^{(l)T} X^{(l)} \right)',$$

где штрихом обозначена операция обнуления главной диагонали. В итоге получим:

$$W = \begin{pmatrix} 0 & -3 & 3 & 1 & 1 & 1 & -3 & 1 & -3 & 1 & 3 & 3 & 3 & 1 & 1 \\ -3 & 0 & -3 & -1 & -1 & -1 & 3 & -1 & 3 & -1 & -3 & -3 & -3 & -1 & -1 \\ 3 & -3 & 0 & 1 & 1 & 1 & -3 & 1 & -3 & 1 & 3 & 3 & 3 & 1 & 1 \\ 1 & -1 & 1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 0 & 3 & -1 & 3 & -1 & 3 & 1 & 1 & 1 & -1 & 3 \\ 1 & -1 & 1 & -1 & 3 & 0 & -1 & 3 & -1 & 3 & 1 & 1 & 1 & -1 & 3 \\ -3 & 3 & -3 & -1 & -1 & -1 & 0 & -1 & 3 & -1 & -3 & -3 & -3 & -1 & -1 \\ 1 & -1 & 1 & -1 & 3 & 3 & -1 & 0 & -1 & 3 & 1 & 1 & 1 & -1 & 3 \\ -3 & 3 & -3 & -1 & -1 & -1 & 3 & -1 & 0 & -1 & -3 & -3 & -3 & -1 & -1 \\ 1 & -1 & 1 & -1 & 3 & 3 & -1 & 3 & -1 & 0 & 1 & 1 & 1 & -1 & 3 \\ 3 & -3 & 3 & 1 & 1 & 1 & -3 & 1 & -3 & 1 & 0 & 3 & 3 & 1 & 1 \\ 3 & -3 & 3 & 1 & 1 & 1 & -3 & 1 & -3 & 1 & 3 & 0 & 3 & 1 & 1 \\ 3 & -3 & 3 & 1 & 1 & 1 & -3 & 1 & -3 & 1 & 3 & 3 & 0 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 0 & -1 \\ 1 & -1 & 1 & -1 & 3 & 3 & -1 & 3 & -1 & 3 & 1 & 1 & 1 & -1 & 0 \end{pmatrix}$$

Для тестирования сети подадим поочередно все 3 рабочих вектора на вход. Можно убедиться, что в синхронном режиме, записывая (7.1) в векторно-матричном виде, результат будет следующий:

$$f(X^{(l)}W) = X^{(l)}, \quad l = 1, 2, 3.$$

Попробуем исказить, например, входной образ $X^{(1)}$, инвертировав биты 2 и 3:

$$\tilde{X}^{(1)} = (-1, -1, 1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1, 1).$$

Результатом работы РНС Хопфилда будет восстановление исходного эталона:

$$f(\tilde{X}^{(1)}W) = X^{(1)}.$$

Варианты работы

Табл. 7.1. Варианты заданий

№ пп	Режим работы РНС Хопфилда	Запоминаемые образы
1	синхронный	0 1 8
2	синхронный	A I F
3	асинхронный	1 3 7
4	асинхронный	3 4 5
5	синхронный	G H I
6	синхронный	6 7 4
7	асинхронный	1 3 5
8	асинхронный	J K L
9	синхронный	2 4 6
10	синхронный	0 6 1
11	асинхронный	S T U
12	асинхронный	3 5 7
13	синхронный	2 4 8
14	синхронный	X Y Z
15	асинхронный	0 1 9
16	асинхронный	I V X
17	синхронный	B C I
18	синхронный	D E X
19	асинхронный	G H L
20	асинхронный	0 2 7
21	синхронный	F H J
22	синхронный	1 4 7
23	асинхронный	N O P
24	асинхронный	S V Z
25	синхронный	1 3 5

Требования к отчету

Отчет должен содержать: титульный лист; цель работы; постановку задачи; закодированные эталонные образы (паттерны); рассчитанную матрицу весов обратных связей; результаты тестирования РНС для эталонных паттернов; результаты работы РНС при искаженных паттернах; выводы по результатам численного эксперимента.

Контрольные вопросы

1. Обучение ассоциативного типа (без учителя). Правило Хебба.
2. Ассоциативная память.
3. Нейронная сеть Хопфилда, алгоритмы обучения и функционирования.

8 Лабораторная работа № 8

Исследование рекуррентной нейронной сети Коско (ВАМ) на примере задачи распознавания образов

Цель работы

Исследовать процедуру обучения и функционирования рекуррентной нейронной сети (РНС) Коско как модели двунаправленной гетероассоциативной памяти (англ. Bidirectional Associative Memory – ВАМ).

Постановка задачи и сведения из теории

Постановка задачи. Закодировать запоминаемые образы (см. табл. 7.1) в виде биполярных матриц-паттернов размерности $I \times J$ (рекомендуемое число пикселей $I \cdot J \approx 20...40$). Произвести векторизацию матриц. Установить каждому образу его пару-ассоциацию как биполярный вектор длины 3. Провести настройку весов РНС Коско (рис. 8.1) согласно правилу ассоциативного обучения (Хебба). Задать функцию активации и реализовать алгоритм функционирования РНС Коско в двунаправленном режиме. Протестировать РНС на запомненных эталонных образах в прямом и обратном режимах. Проверить функционирование РНС Коско в прямом режиме на искаженных паттернах (изменены порядка 10% пикселей).

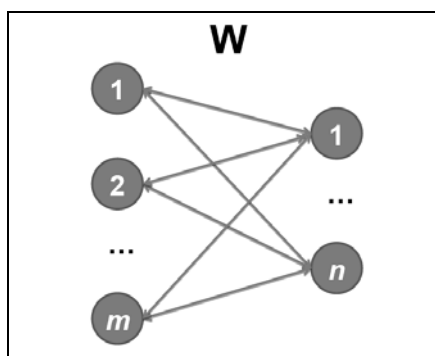


Рис. 8.1. Рекуррентная нейронная сеть Коско (ВАМ)

Рабочий режим. Рекуррентная нейронная сеть Коско является гетероассоциативной памятью, которая в ответ на исходное входное воздействие-сигнал

$$S = (s_1, s_2, \dots, s_m), s_i \in \{-1, 1\} (i = \overline{1, m}),$$

формирует отклик

$$T = (t_1, t_2, \dots, t_n), t_j \in \{-1, 1\} (j = \overline{1, n}),$$

и обратно, по сигналу T в качестве прообраза восстанавливает образ S (в общем случае $m \neq n$).

В рабочем режиме производятся следующие вычисления.

1. Инициализация выхода нейронов первого слоя:

$$y_i(0) = s_i \quad (i = \overline{1, m}).$$

2. Расчет сигналов комбинированных входов нейронов второго слоя:

$$\text{net}_j = \sum_{i=1}^m s_i w_{ij} \quad (j = \overline{1, n}).$$

3. Расчет выходов нейронов второго слоя:

$$t_j = f(\text{net}_j) \quad (j = \overline{1, n}).$$

4. Расчет сигналов комбинированных входов нейронов первого слоя:

$$\text{net}_i = \sum_{j=1}^n t_j w_{ji} \quad (i = \overline{1, m}).$$

5. Расчет выходов нейронов первого слоя:

$$s_i = f(\text{net}_i) \quad (i = \overline{1, m}).$$

6. Повторение шагов 2-5 до стабилизации обоих выходов РНС Коско.

Порядок вычислений может быть другим, т.е. начинаться с активации нейронов второго слоя:

$$y_j(0) = t_j \quad (j = \overline{1, n}),$$

после чего выполняются сначала шаги 4, 5, а затем – шаги 2, 3.

Функция активации каждого нейрона на k -й эпохе имеет вид

$$f(\text{net}_i^{(k)}) = \begin{cases} 1, & \text{net}_i^{(k)} > 0, \\ f(\text{net}_i^{(k-1)}), & \text{net}_i^{(k)} = 0, \\ -1, & \text{net}_i^{(k)} < 0. \end{cases}$$

Инициализация. Для начала работы РНС Коско необходимо по правилу Хебба вычислить компоненты матрицы весов:

$$w_{ij} = \sum_{l=1}^L s_i^{(l)} t_j^{(l)} \quad (i = \overline{1, m}, j = \overline{1, n}), \quad (8.1)$$

где l – номера запоминаемых пар образов, а L – «емкость» гетероассоциативной памяти (количество запоминаемых пар образов).

Пример выполнения работы

Пусть требуется запомнить три образа, представляющих собой графические изображения цифр «1», «2», «3». В качестве примера закодируем их в виде биполярных матриц-паттернов размерности 5×3 (см. рис. 7.2). Векторизуем матрицы по столбцам для получения рабочих векторов длины $m = 15$:

$$\begin{aligned} S^{(1)} &= (-1, 1, -1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1, 1), \\ S^{(2)} &= (1, -1, 1, 1, 1, 1, -1, 1, -1, 1, 1, 1, -1, 1), \\ S^{(3)} &= (1, -1, 1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1). \end{aligned}$$

В качестве пар-ассоциаций возьмем биполярные векторы размерности $n = 3$:

$$\begin{aligned} T^{(1)} &= (-1, -1, 1), \\ T^{(2)} &= (-1, 1, -1), \\ T^{(3)} &= (-1, 1, 1). \end{aligned}$$

Настроим веса РНС Коско, записав (8.1) в векторно-матричном виде:

$$W = \sum_{l=1}^3 S^{(l)T} T^{(l)}.$$

В итоге получим

$$W = \begin{pmatrix} -1 & 1 & -1 & 1 & -3 & -3 & 1 & -3 & 1 & -3 & -1 & -1 & -11 & -3 \\ 3 & -3 & 3 & 1 & 1 & 1 & -3 & 1 & -3 & 1 & 3 & 3 & 3 & 1 \\ -1 & 1 & -1 & -3 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -11 & 1 \end{pmatrix}^T.$$

Для тестирования сети активируем поочередно всеми тремя образами $S^{(l)}$ ее первый слой и выполним шаги 2, 3 рабочего режима. В итоге на выходе нейронов второго слоя получим соответствующие пары-ассоциации $T^{(l)}$. Аналогично, при активации нейронов второго слоя поочередно всеми тремя образами $T^{(l)}$ и выполнения шагов 4, 5 рабочего режима, получим на выходе нейронов первого слоя соответствующие пары $S^{(l)}$.

Попробуем исказить, например, входной образ $S^{(1)}$, инвертировав бит 4:

$$\tilde{S}^{(1)} = (-1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, 1)$$

Результатом работы РНС Коско после шагов 2, 3 будет восстановление на выходе второго слоя исходного парного образа $T^{(1)} = (-1, -1, 1)$. Реализуя далее шаги 4, 5 рабочего режима, на выходе нейронов первого слоя восстановится неискаженный образ

$$S^{(1)} = (-1, 1, -1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1, 1).$$

Аналогичная ситуация будет иметь место при внесении небольших искажений в любой из векторов $S^{(l)}, T^{(l)}$ ($l = 1, 2, 3$).

Варианты работы

См. табл. 7.1.

Требования к отчету

Отчет должен содержать: титульный лист; цель работы; постановку задачи; закодированные эталонные образы (паттерны) и их пары-ассоциации; рассчитанную матрицу весов; результаты тестирования РНС Коско для эталонных паттернов в прямом и обратном направлении; результаты работы РНС Коско при искаженных паттернах; выводы по результатам численного эксперимента.

Контрольные вопросы

4. Обучение ассоциативного типа (без учителя). Правило Хебба.
5. Ассоциативная память.
6. Нейронная сеть Коско (ВАН), алгоритмы обучения и функционирования.

Лабораторная работа № 9

Алгоритмы кластерного анализа данных

Цель работы

Исследовать применение основных алгоритмов кластерного анализа, включая их модификации, на примере различных типов данных.

Постановка задачи и сведения из теории

Постановка задачи. Выполнить разбиение предложенных данных при помощи указанного в варианте (см. табл. 9.1) алгоритма с использованием соответствующих метрик.

Кластерный анализ – процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы на основе какого-либо признака(-ов). Формально: Пусть X – множество объектов, Y – множество кластеров. Задана функция расстояния между объектами $\rho(x, x')$. Имеется конечная обучающая выборка объектов $X^m = \{x_1, \dots, x_m\} \subset X$. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике ρ , а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X^m$ приписывается номер кластера y_i . Алгоритм кластеризации – это функция $\varphi: X \rightarrow Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$. Множество Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации.

Существует множество методов кластерного анализа, наиболее известные из них:

- вероятностные алгоритмы (k-средних),
- подходы на основе применения искусственного интеллекта (нейронная сеть Кохонена),
- теоретико-графовый подход и др.

Алгоритм k-средних (k-means) стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров:

$$S = \sum_{i=1}^k \sum_{x_j \in Y_i} (x_j - \mu_i)^2,$$

где k – количество кластеров (задано заранее); Y_i – полученные кластеры, $i = 1, \dots, k$; μ_i – центры масс $x_j \in Y_i$. Суть алгоритма заключается в следующем: на каждой итерации переычисляется центр масс для каждого кластера, полученного на предыдущем шаге,

затем кластеризуемые точки $x_j \in R^d$ разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на l -той итерации не происходит изменения центра масс кластеров. Это происходит за конечное число итераций, так как количество возможных разбиений конечного множества конечно, и на каждом шаге суммарное квадратичное отклонение S не увеличивается, поэтому заикливание невозможно.

Нейронные сети Кохонена – класс нейронных сетей, основным элементом которых является слой Кохонена, состоящий из k адаптивных линейных сумматоров. Они имеют одинаковое число входов m и получают на свои входы вектор входных сигналов $x = (x_1, \dots, x_m)$. На выходе j -го линейного элемента получаем сигнал

$$y_j = w_{j0} + \sum_{i=1}^m w_{ji} x_i$$

где j – номер нейрона, w_{j0} – пороговый коэффициент, i – номер входа, w_{ji} – весовой коэффициент i -го входа j -го нейрона.

Выходные сигналы слоя Кохонена обрабатываются по правилу «победитель получает всё»: наибольший сигнал превращается в единичный, остальные обращаются в ноль. Таким образом, применительно к задаче кластеризации каждому j -му нейрону ставятся в соответствие точки-центры кластеров, для входного вектора $x = (x_1, \dots, x_m)$ вычисляются расстояния $\rho_j(x)$, и тот нейрон, до которого это расстояние минимально, выдает единицу, остальные – ноль.

Пример выполнения работы

Рассмотрим пример выполнения кластеризации данных на примере разбивки на кластере координатных точек на форме приложения по методу k -средних. Пусть пользователь ввел 7 координатных точек:

$$x_1 = (143, 213); x_2 = (180, 220); x_3 = (183, 249); \\ x_4 = (271, 253); x_5 = (226, 253); x_6 = (315, 275); x_7 = (266, 297),$$

а также задал 2 кластера точками

$$Y_1 = (159, 238); Y_2 = (270, 278)$$

(см. рис. 9.1, где точками обозначены исходные координатные точки, а квадратами – начальные центры кластеров). Теперь в цикле по всем исходным точкам нужно рассчитать расстояние ρ по заданной метрике (мы рассмотрим обычное, евклидово, расстояние) от каждой точки до центра каждого из заданных кластеров и отнести точку к тому или иному кластеру из соображений минимальности расстояния до центра кластера.

Мы получаем следующие значения:

$$\begin{array}{ccccccc} \rho_{11} = 29.68 & \rho_{21} = 27.66 & \rho_{31} = 26.40 & \rho_{41} = 113.0 & \rho_{51} = 68.66 & \rho_{61} = 160.33 & \rho_{71} = 122.19 \\ \rho_{12} = 142.67 & \rho_{22} = 107.07 & \rho_{32} = 91.71 & \rho_{42} = 25.02 & \rho_{52} = 50.61 & \rho_{62} = 45.10 & \rho_{72} = 19.42 \end{array}$$

Здесь ρ_{ij} – расстояние от i -той точки до центра j -го кластера. Таким образом, к первому кластеру будут отнесены точки $Y_1 \in \{x_1, x_2, x_3\}$, а ко второму $Y_2 \in \{x_4, x_5, x_6, x_7\}$.

На следующем шаге алгоритма перерасчитываются новые центры кластеров (как центры масс точек, отнесенных к j -му кластеру):

$$Y_1' = (168, 67; 227, 34); Y_2' = (269, 5; 269, 5).$$

После чего процедура расчета расстояний ρ_{ij} до новых центров кластеров и отнесение к тому или иному кластеру повторяется до тех пор, пока на очередном шаге алгоритма больше не произойдет изменений центров кластеров относительно предыдущего шага:

$$Y_j^{l+1} = Y_j^l, j = 1, \dots, k.$$

Тогда работу алгоритма можно считать законченной. Пример можно видеть на рис. 9.2.

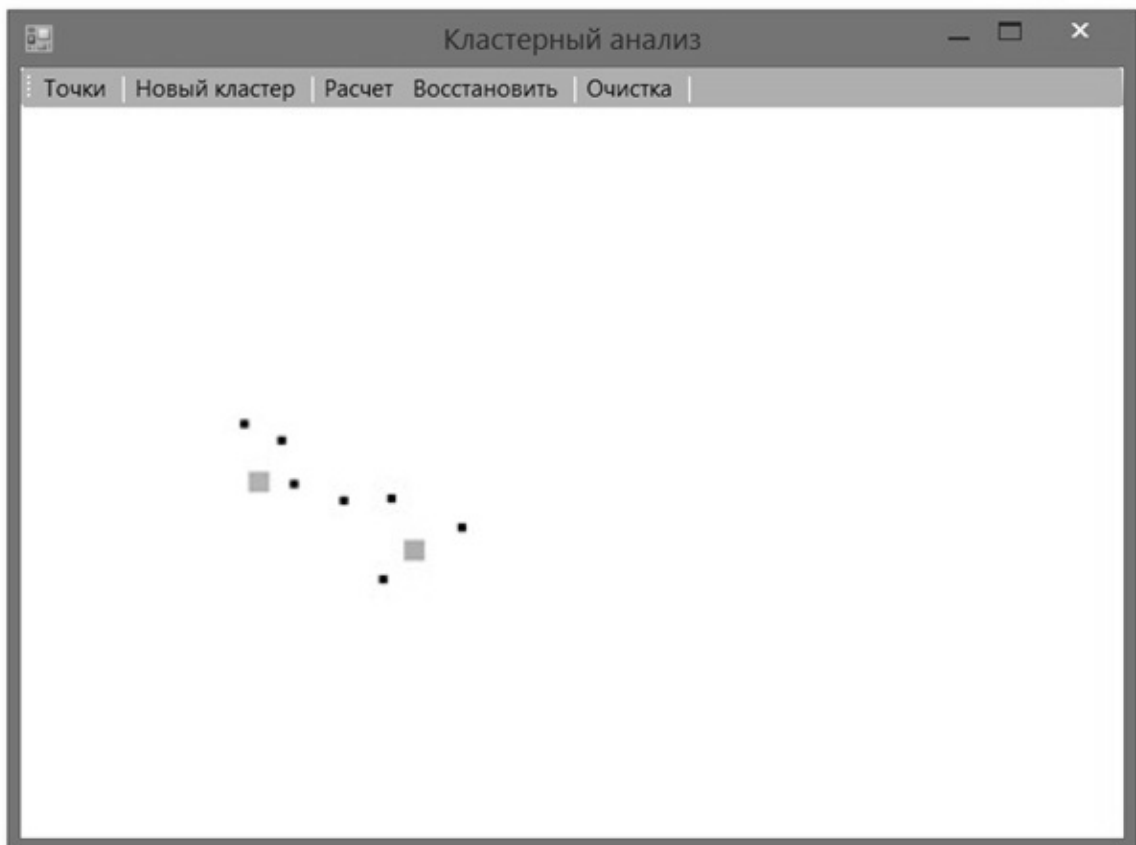


Рис. 9.1. Пример работы программы: введены исходные точки и начальные центры кластеров

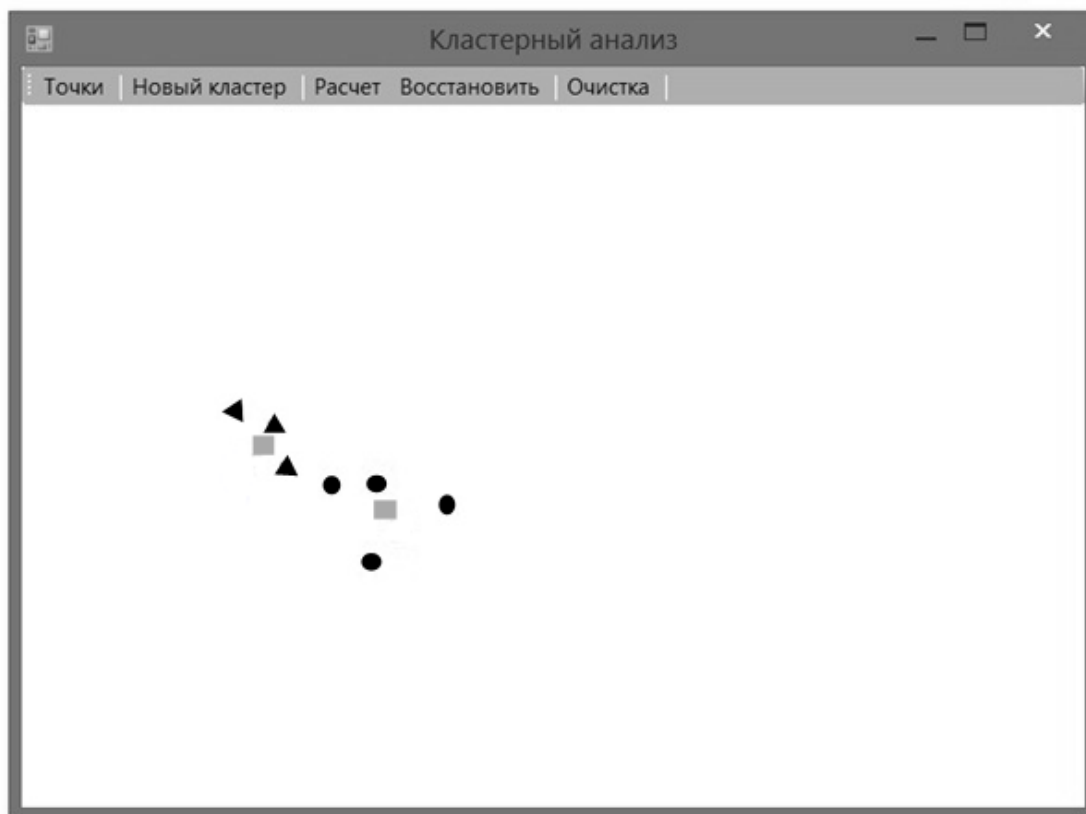


Рис. 9.2. Пример работы программы: результат разбиения точек на кластеры

Варианты работы

Выполнить разбиение предложенных в табл. 9.1 данных на кластеры при помощи указанного алгоритма (с обеими предложенными в таблице метриками) согласно варианту. При этом:

-*в нечетных вариантах* задания должен быть предусмотрен пользовательский GUI для ввода данных (координатных точек на форме приложения) и вывода результата кластеризации; результат кластеризации должен отображаться с цветовым кодированием по принадлежности к кластеру; приветствуется пошаговое отображение результатов работы алгоритма;

-*в четных вариантах* задания приложение может быть выполнено как консольное с выводом в консоль или выходной файл; в этих вариантах данные берутся с портала открытых данных правительства Москвы (<http://data.mos.ru/>) по варианту. При кластеризации данных из коллекций портала открытых данных должны использоваться только указанные в задании поля (координаты (X, Y) или количество мест), а остальные поля (название округа и т.п.) должны использоваться только в проверке результатов кластеризации.

Табл. 9.1. Варианты заданий

№ пп	Алгоритм	Исходные кластеризуемые данные	ρ
1	k-средних	Координатные точки на форме приложения	Евклида, Чебышева
2	НС Кохонена	Выборка колледжей г. Москвы, http://data.mos.ru/datasets/546 координаты местоположения (X, Y)	Принадлежность округу Москвы (Евклидово расстояние до координат (X_0, Y_0) центра округа)
3	k-средних	Координатные точки на форме приложения	Евклида, Манхеттен
4	НС Кохонена	Выборка лицеев г. Москвы, http://data.mos.ru/datasets/552 координаты местоположения (X, Y)	Принадлежность округу Москвы (Евклидово расстояние до координат (X_0, Y_0) центра округа)
5	k-средних	Координатные точки на форме приложения	Манхеттен, Чебышева
6	k-средних	Выборка велопарковок г. Москвы, http://data.mos.ru/datasets/916 координаты местоположения (X, Y)	Принадлежность округу Москвы (Евклидово расстояние до координат (X_0, Y_0) центра округа)
7	k-медиан	Координатные точки на форме приложения	Евклида, Чебышева
8	НС Кохонена	Выборка платных парковок г. Москвы, http://data.mos.ru/datasets/623 вместительность, машин, достаточный размер выборки $N = 200$ объектов	Вместительность, количество машин (Кластеры предложить самостоятельно)
9	k-медиан	Координатные точки на форме приложения	Евклида, Манхеттен
10	НС Кохонена	Выборка кинотеатров г. Москвы, http://data.mos.ru/datasets/495 координаты местоположения (X, Y)	Принадлежность округу Москвы (Евклидово расстояние до координат (X_0, Y_0) центра округа)
11	k-медиан	Координатные точки на форме приложения	Манхеттен, Чебышева

Продолжение табл. 9.1

№ пп	Алгоритм	Исходные кластеризуемые данные	ρ
12	НС Кохонена	Выборка взрослых больниц г. Москвы, http://data.mos.ru/datasets/517 координаты местоположения (X, Y) или количество коек для больных	Принадлежность округу Москвы (Евклидово расстояние до координат (X_0, Y_0) центра округа) или количество коек для больных
13	k-средних	Координатные точки на форме приложения	Евклида, Чебышева
14	НС Кохонена	Выборка взрослых поликлиник г. Москвы, http://data.mos.ru/datasets/503 координаты местоположения (X, Y)	Принадлежность округу Москвы (Евклидово расстояние до координат (X_0, Y_0) центра округа)
15	k-средних	Координатные точки на форме приложения	Евклида, Манхеттен
16	НС Кохонена	Выборка библиотек г. Москвы, http://data.mos.ru/datasets/526 координаты местоположения (X, Y) достаточный размер выборки $N = 200$ объектов	Принадлежность округу Москвы (Евклидово расстояние до координат (X_0, Y_0) центра округа)
17	k-средних	Координатные точки на форме приложения	Манхеттен, Чебышева
18	НС Кохонена	Выборка музеев г. Москвы, http://data.mos.ru/datasets/529 Занимаемая музеем площадь (LandArea)	Занимаемая музеем площадь (LandArea)
19	НС Кохонена	Координатные точки на форме приложения	Евклида, Чебышева
20	НС Кохонена	Выборка театров г. Москвы, http://data.mos.ru/datasets/531 вместительность, мест в залах	Вместительность, мест в залах (Кластеры предложить самостоятельно)
21	НС Кохонена	Координатные точки на форме приложения	Евклида, Манхеттен
22	НС Кохонена	Выборка детских садов г. Москвы, http://data.mos.ru/datasets/540 координаты местоположения (X, Y) ; выбрать случайным образом (не подряд) $N = 200$ объектов	Принадлежность округу Москвы (Евклидово расстояние до координат (X_0, Y_0) центра округа)

Продолжение табл. 9.1

№ пп	Алгоритм	Исходные кластеризуемые данные	ρ
23	НС Кохонена	Координатные точки на форме приложения	Манхеттен, Чебышева
24	НС Кохонена	Выборка детских садов г. Москвы, http://data.mos.ru/datasets/540 координаты местоположения (X, Y) выбрать случайным образом (не подряд) $N = 200$ объектов	Принадлежность округу Москвы (Евклидово расстояние до координат (X_0, Y_0) центра округа)
25	k-средних	Координатные точки на форме приложения	Евклида, Манхеттен

Требования к отчету

Отчет должен содержать: титульный лист; цель работы; постановку задачи; примеры работы программы; результат решения поставленной задачи (итог работы программы), а также выводы по результатам численного эксперимента.

Контрольные вопросы

1. Дайте классификацию алгоритмов кластерного анализа данных.
2. Сформулируйте основные шаги алгоритма функционирования НС Кохонена.
3. Сформулируйте основные шаги алгоритма k-средних.
4. Перечислите основные преимущества и недостатки алгоритма k-средних.

Список литературы

1. Тадеусевич Р., Боровик Б., Гончаж Т., Леппер Б. *Элементарное введение в технологию нейронных сетей с примерами программ.* – М.: Горячая линия – Телеком, 2011. – 408 с.
2. Каллан Р. *Основные концепции нейронных сетей.* – М.: Изд. дом «Вильямс». – 2003. – 288 с.
3. Шелухин О.И., Сакалема Д.Ж., Филинова А.С. *Обнаружение вторжений в компьютерные сети (сетевые аномалии).* – М.: Горячая линия – Телеком, 2013. – 220 с.
4. Волчихин В.И., Иванов А.И., Назаров И.Г., Фунтиков В.А., Язов Ю.К. *Нейросетевая защита персональных биометрических данных.* – М.: Радиотехника, 2012. – 160 с.