

**Московский государственный технический университет им. Н. Э. Баумана**

**Кафедра ИУ-4  
«Проектирование и технология производства ЭС»**

# **Журнал практических работ**

**по курсу: «Системы искусственного интеллекта»**

**Для студентов приборостроительных специальностей**

**20\_\_ / \_\_ учебный год**

Студент \_\_\_\_\_ (фамилия, и. о.)                      Группа \_\_\_\_\_

Преподаватель \_\_\_\_\_ Допуск к экзамену (зачету) \_\_\_\_\_ Подпись \_\_\_\_\_  
(фамилия, и. о.)                      (число)

**Москва  
2013**

Программа

к учебному плану направления подготовки 551100 (654300)

ПРОЕКТИРОВАНИЕ И ТЕХНОЛОГИЯ ЭЛЕКТРОННЫХ СРЕДСТВ,  
специальностям

220500 Проектирование и технология электронно-вычислительных средств и  
200800 Проектирование и технология радиоэлектронных средств.

№	Виды учебных работ	Объем работ в часах		
		Всего	9 сем.	
	На дисциплину	<b>68</b>	<b>68</b>	
1	Аудиторная работа	<b>51</b>	<b>51</b>	
1.1	- лекции	<b>34</b>	<b>34</b>	
1.2	- семинары	<b>17</b>	<b>17</b>	
1.3	- лабораторные занятия	-	-	
1.4	Самостоятельная работа:	<b>68</b>	<b>68</b>	
	Домашние задания:	17	-	
	• ДЗ1		17 (14)	
	Курсовая работа	-	-	
	Самостоятельное изучение раздела			
1.5	Виды отчетности по дисциплине			
	Контрольная работа			
	Рубежный контроль		РК (10)	
	Зачеты			
	Экзамены		<b>экзамен</b>	

**Отчет по лабораторной работе № 1**  
**«Разработка ЭС по ТП ЭС с использованием iu4.expert.lab»**

дата	Оценка (max 5)	Бонус за сложность	подпись
------	-------------------	-----------------------	---------

**Цели работы:**

Изучение принципов построения экспертных систем, знакомство с работой специализированной фреймовой экспертной системы iu4.expert.lab, создание экспертной системы синтеза ТП производства на основе комплексного задания

**Задачи работы:**

- знакомство с работой программы iu4.expert.lab на примере экспертной системы определения объекта (файл lab1.doc)
- создание экспертной системы синтеза ТП производства (файл expert\_var.dat, где var номер варианта по списку) с использованием системы iu4.expert.lab

**Задание повышенной сложности (бонус за сложность – 5 баллов):**

-разработка собственной экспертной программы на Си для синтеза технических решений с использованием фреймовых моделей представления знаний (пример таблица 1.1 lab1.doc)

**Краткий конспект теоретической части (ответы на контрольные вопросы lab1.doc)**

Понятие искусственный интеллект \_\_\_\_\_

\_\_\_\_\_

Определение экспертной системы \_\_\_\_\_

\_\_\_\_\_

Требования к экспертным системам \_\_\_\_\_

\_\_\_\_\_

Структура экспертной системы \_\_\_\_\_

\_\_\_\_\_

Модели представления знаний \_\_\_\_\_

\_\_\_\_\_

Понятие фрейма \_\_\_\_\_

\_\_\_\_\_

Фреймовая семантическая сеть \_\_\_\_\_

\_\_\_\_\_

Этапы разработки ЭС \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_





5. Работа с машиной вывода:

Зарисуйте (вклейте) структуру фреймового семантического дерева	Приведите пример диалога
	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

Зарисуйте результат синтеза	Сформулируйте выводы по работе
	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

**Контрольные вопросы**

1. Дайте определение понятия «искусственный интеллект»?
2. Дайте определение понятия «экспертная система»?
3. Какие задачи решаются экспертными системами?
4. Требования, предъявляемые к экспертным системам?
5. Обобщенная структура экспертных систем?
6. Модели представления знаний в экспертных системах?
7. Понятие «фрейма»?
8. Понятие «фреймовой семантической сети»?
9. Этапы разработки ЭС?
10. Фазы создания ЭС и их особенности?
11. Предназначение и принципы работы экспертной системы `iu4.expert.lab.`?
12. Понятие «слота» фрейма?
13. Понятие «фрейма прототипа»?
14. Понятие «конкретного фрейма»?
15. Понятие «целевого фрейма»?

## СПИСОК ЛИТЕРАТУРЫ

1. Норенков И.П. Системы автоматизированного проектирования. - М.: Изд-во МГТУ им.Н.Э.Баумана. 2001.
2. Власов А.И., Лыткин С.Л., Яковлев В.Л. Краткое практическое руководство по языку PL/SQL - М.: Машиностроение. 2000. 64 с.
3. Базы данных: интеллектуальная обработка информации - М.: Нолидж. 2001. - 496 с.
4. О.В.Герман Ведение в теорию экспертных систем и обработку знаний. - Минск. Дизайн-ПРО. 1995. - 255 с.
5. Информационные технологии в конструировании и технологии микроэлектроники/В.П.Григорьев, Ю.И.Нестеров, Д.В.Черепанов. - М.: Изд-во МГТУ им.Н.Э.Баумана. 2000. - 39 с.
6. Применение методов искусственного интеллекта в САПР технологического проектирования производства электронной аппаратуры/В.П.Григорьев, Э.Н.Камышная, Ю.И.Нестеров, С.А.Никитин - М.: Изд-во МГТУ им.Н.Э.Баумана. 1998. - 48 с.
7. Майкл Р. Oracle 7.3: Энциклопедия пользователя: Пер с англ. К.: Диасофт. 1997. 832 с.
8. Фаронов В.В., Шумаков П.В. Delphi 4: Руководство разработчика баз данных. М.: Нолидж, 1999. 560 с., ил.
9. Урман С. Oracle 8: Программирование на языке PL/SQL. М.: ЛОРИ, 1999. 608 с.
10. Луни К. Oracle 8: Настольная книга администратора. М.: ЛОРИ, 1999. 500 с.
11. Бобровски С. Oracle 8: Архитектура. М.: ЛОРИ, 1999. 208 с.
12. Хансен Г., Хансен Д. Базы данных: разработка и управление: Пер. с англ. М.: ЗАО "Издательство БИНОМ", 1999. 704 с.
13. Дунаев С. Доступ к базам данных и техника работы в сети. Практические приемы современного программирования. М.: ДИАЛОГ-МИФИ, 1999. 416 с.
14. Петерсен Р. Linux: Руководство по операционной системе. В 2 т: Пер. с англ. 2-е изд., перераб. и доп. К. Издательская группа BHV, 1998.
15. Скотт Урман Oracle 8. Программирование на языке PL/SQL. Руководство для программистов Издательство: Лори, 1999 г., 610 стр.
16. Кристофер Аллен 101 Oracle PL/SQL. Как писать мощные и гибкие программы на PL/SQL Издательство: Лори, 2001 г., 350 стр.
17. Марлен Терьо, Рэчел Кармайл, Джеймс Вискузи 101 Oracle. Настольная книга администратора Издательства: Лори, McGraw-HILL Inc, 2001 г., 490 стр.
18. Кевин Луни, Марлен Терьо Oracle 9i. Настольная книга администратора Oracle 9i DBA Handbook Издательство: Лори, 2003 г., 766 стр.
19. Толковый словарь по искусственному интеллекту <http://www.raai.org/library/tolk/aivoc.html#L74>

Отчет по лабораторной работе № 2			
«Разработка модулей ЭС с использованием хранимых процедур PL/SQL»			
дата	Оценка (max 5)	Бонус за сложность	подпись

**Цели работы:**

Изучение принципов построения модулей экспертных систем с использованием хранимых процедур на PL/SQL

**Задачи работы:**

-знакомство с возможным составом ЭС , реализованной в виде хранимых процедур на PL/SQL (файл lab2.doc)

-создание модуля экспертной системы синтеза ТП производства , исследованного в работе 1 и его реализация на сервере <http://oracle.iu4.bmstu.ru> в виде хранимых процедур с вызовом из SQL+.

**Задание повышенной сложности (бонус за сложность – 5 баллов):**

-разработка полнофункциональную экспертную систему на хранимых процедурах (PL/SQL) (прототип модули ЭС на Си, файл lab1.doc) для синтеза технических решений с использованием фреймовых моделей представления знаний (пример lab2.doc)

**Краткий конспект теоретической части (ответы на контрольные вопросы lab2.doc)**

Опишите актеров и прецедентов ЭС \_\_\_\_\_

\_\_\_\_\_

Опишите последовательность действий с базовыми модулями \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Опишите логическую модель модуля \_\_\_\_\_

\_\_\_\_\_

Опишите базовые пакеты и классы модуля \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

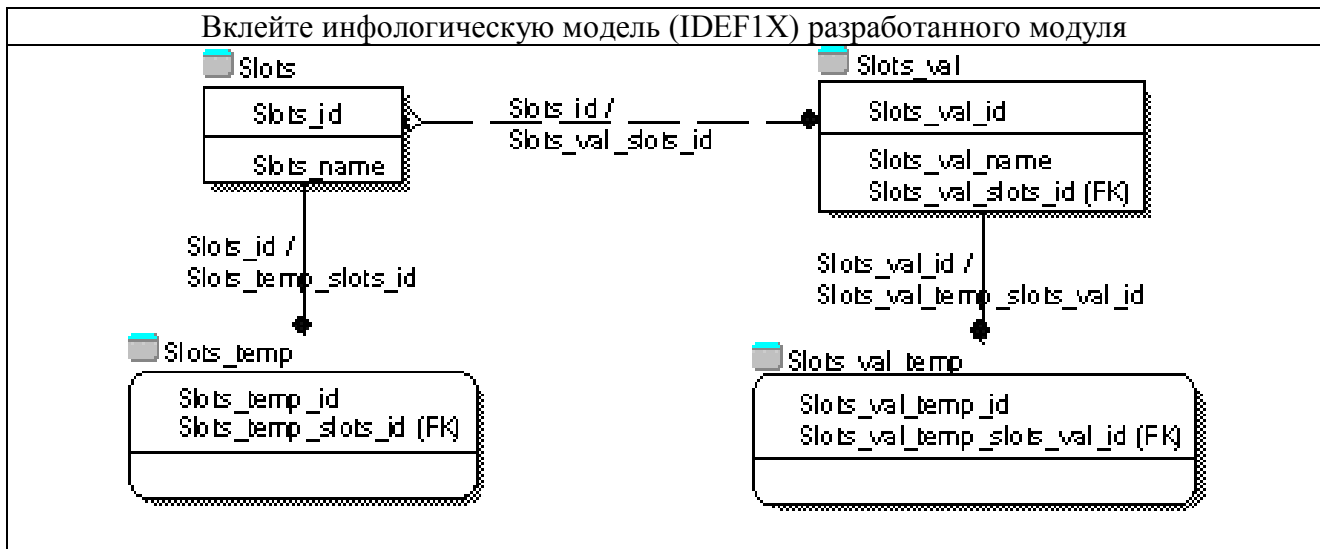
\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_



Разработайте один из модулей пакета (см. описание lab2.doc), разверните указанный модуль на сервере <http://oracle.iu4.bmstu.ru>, продемонстрируйте гео работу.



Пример модуля

**Задание:** В данном модуле реализовано управление слотами. Эксперт, владея администраторскими правами, может, как и добавлять новые слоты, так и вносить изменения в слоты, созданные другими экспертами.

Разработанный исходный код вклейте в табл.2.1., желательно, для хранения промежуточных результатов использовать «Курсоры», а не таблицы.

<p>Вклейте скриншот интерфейса SQL+ (или PHP) с вызовом разработанного модуля</p> <p><b>Ничего не найдено:</b></p> <pre> Choose Taste, please:   Sweet.  (&lt;@ + y/n&gt;)? SQL&gt; @n   Sour.   (&lt;@ + y/n&gt;)? SQL&gt; @y Ok! ----- Choose Color, please:   Yellow. (&lt;@ + y/n&gt;)? SQL&gt; @n   Red.    (&lt;@ + y/n&gt;)? SQL&gt; @n   Pink.   (&lt;@ + y/n&gt;)? SQL&gt; @n   Green.  (&lt;@ + y/n&gt;)? SQL&gt; @n Nothing found! SQL&gt;         </pre>	<p><b>Ответ найден:</b></p> <pre> Choose Taste, please:   Sweet.  (&lt;@ + y/n&gt;)? SQL&gt; @n   Sour.   (&lt;@ + y/n&gt;)? SQL&gt; @n   Bitter. (&lt;@ + y/n&gt;)? SQL&gt; @y Ok! ----- Choose Color, please:   Yellow. (&lt;@ + y/n&gt;)? SQL&gt; @n   Red.    (&lt;@ + y/n&gt;)? SQL&gt; @y Ok! ----- Results:   Color -- Red   Taste -- Bitter SQL&gt;         </pre>
--	--

Таблица 2.1 Листинг исходного кода программы

**Пример исходного кода (вместо кода примера вклеить свой):**

```

start.sql          // отсюда стартует экспертная система
CLEAR SCREEN;     --убочищаем экран
SET SERVEROUTPUT ON;      --включаем вывод информации с сервера
SET FEED OFF;       --отключаем вывод лишней информации
TRUNCATE TABLE slots_tmp;      --очищаем временную таблицу slots_tmp
TRUNCATE TABLE slots_val_tmp;  --очищаем временную таблицу slots_val_tmp

//Пользовательский интерфейс на sql+
DECLARE
print_tmp VARCHAR2(50);  --переменная, которая принимает значения
--функций, здесь она не особо нужна, введена
--для единообразия кода
BEGIN
    DBMS_OUTPUT.enable;
    print_tmp := EXPERT.Sel_Slots; --запрашиваем 1-ый слота [цвет/вкус]
    DBMS_OUTPUT.put_line('Choose ' || print_tmp || ', please:');
    print_tmp := EXPERT.Sel_Slots_val; --1-ое значение слота [синий/зеленый]
    DBMS_OUTPUT.put_line(chr(9) || print_tmp || '. (@ + y/n)?');

END;
/

//***** файлы инсталляционного комплекта *****
n.sql          // вызываем в случае отрицательного ответа
DECLARE
print_tmp VARCHAR2(50);
--переменная, которая принимает значения функций, по ней ещё
--определяется конец работы системы
BEGIN
    DBMS_OUTPUT.enable;
    print_tmp := EXPERT.Sel_Slots_val;      --выбираем следующее значение слота
    IF (print_tmp = '!no_found!') THEN      --значения кончились, а к результату
--не пришли. Значит пишем, что
--ничего не найдено:
        DBMS_OUTPUT.put_line('Nothing found!');
    ELSE                                     --иначе просто выводим на экран
--значение:
        DBMS_OUTPUT.put_line(chr(9) || print_tmp || '. (@ + y/n)?');
    END IF;
END;
/
y.sql          // вызываем в случае положительного ответа
DECLARE
print_tmp VARCHAR2(50);  --переменная, которая принимает значения
--функций; важна, так как по ней ещё
--определяется конец работы системы
CURSOR get_slots IS      --курсор, необходимый для построения итогового
    SELECT * FROM slots; --отчёта. По сути это список слотов
v_sl get_slots%ROWTYPE;  --переменная для работы с курсором
BEGIN
    DBMS_OUTPUT.enable;
    DBMS_OUTPUT.put_line('Ok!');           --Ok, т.к. ответ положительный
    DBMS_OUTPUT.put_line('----');
    print_tmp := EXPERT.Sel_Slots;         --выбираем следующий слот
    IF (print_tmp = '!no_found!') THEN     --слоты кончились, то есть,
--это конец программы
        DBMS_OUTPUT.put_line('Results:'); --выводим результаты
        FOR v_sl IN get_slots LOOP        --каждому слоту (v_sl.slots_name)
--ищем его значение
            SELECT slots_val_name INTO print_tmp FROM slots_val
            WHERE (slots_val_id = (
SELECT sl_val_id FROM slots_val_temp
            WHERE (sl_val_temp_id = (
SELECT MAX(sl_val_temp_id) FROM slots_val_temp
            WHERE sl_val_id IN (

```

```

SELECT slots_val_id FROM slots_val
                                WHERE slots_val_slots_id = v_sl.slots_id)))));
                                --выводим всё красиво на экран:
    DBMS_OUTPUT.put_line(chr(9) || v_sl.slots_name || ' -- ' || print_tmp);
END LOOP;
ELSE
--слоты не кончились, о чем сообщаем пользователю и предлагаем первое
--значение слота:
    DBMS_OUTPUT.put_line('Choose ' || print_tmp || ', please:');
    print_tmp := EXPERT.Sel_Slots_val;
    DBMS_OUTPUT.put_line(chr(9) || print_tmp || '. (@ + y/n)?');
END IF;
END;
/
pkg.sql // пакет с необходимыми функциями экспертной системы
CREATE OR REPLACE PACKAGE Expert IS
FUNCTION Sel_slots RETURN VARCHAR2; --функция, возвращающая следующий слот
FUNCTION Sel_slots_val RETURN VARCHAR2; --функция, возвращающая
--следующее значение слота
END Expert;
/

CREATE OR REPLACE PACKAGE BODY Expert IS
FUNCTION Sel_slots RETURN VARCHAR2 IS --возвращает след. слот
temp NUMBER(5); --сюда скидываем id слота
print_tmp VARCHAR2(50); --сюда его название
BEGIN
--выбираем слот, при этом проверяя, что его не выбирали раньше:
SELECT slots_id, slots_name INTO temp, print_tmp FROM slots
WHERE (slots_name = (
SELECT MAX(slots_name) FROM slots
WHERE (slots_id NOT IN (
SELECT sl_id FROM slots_temp))))); --запоминаем, что слот выбран:
INSERT INTO slots_temp(sl_id) VALUES (temp);
COMMIT;
RETURN(print_tmp); --возвращаем его имя
EXCEPTION --слоты кончились, возвращаем спец. фразу
WHEN NO_DATA_FOUND THEN
RETURN('!no_found!');
END Sel_slots;

FUNCTION Sel_slots_val RETURN VARCHAR2 IS --возвращает след. значение
temp NUMBER(5); --сюда скидываем id значения
print_tmp VARCHAR2(50); --сюда его название
BEGIN
--выбираем значение, при этом проверяя, что его не выбирали раньше:
SELECT slots_val_id, slots_val_name
INTO temp, print_tmp
FROM slots_val
WHERE (slots_val_name = (
SELECT MAX(slots_val_name) FROM slots_val WHERE (slots_val_slots_id = (
SELECT sl_id FROM slots_temp WHERE (sl_temp_id = (
SELECT MAX(sl_temp_id) FROM slots_temp)))) AND (slots_val_id NOT IN (
SELECT sl_val_id FROM slots_val_temp)))));
--запоминаем, что значение выбрано:
INSERT INTO slots_val_temp(sl_val_id) VALUES (temp);
COMMIT;
RETURN(print_tmp); --возвращаем значение
EXCEPTION --значения кончились, возвращаем спец. фразу
WHEN NO_DATA_FOUND THEN
RETURN('!no_found!');
END Sel_slots_val;
END Expert;
/

```

Зарисуйте алгоритм работы ЭС

Выводы: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

### Контрольные вопросы

1. Какие основные модули входят в состав фреймовой ЭС, дайте их краткую характеристику?
2. Принципы реализации «движка» ЭС в виде хранимых процедур на PL/SQL?
3. Какие основные таблицы могут быть использованы для реализации фреймовой ЭС?
4. Требования, предъявляемые к экспертным системам?
5. Обобщенная структура фреймовой экспертных систем, реализуемой в виде хранимых процедур?
6. Прокомментируйте исходный код разработанной процедуры модуля ЭС?
7. Проиллюстрируйте работоспособность разработанного модуля ЭС на сервере <http://oracle.iu4.bmstu.ru>

### СПИСОК ЛИТЕРАТУРЫ

1. Урман С. Oracle 8: Программирование на языке PL/SQL. М.: ЛОРИ, 1999. 608 с.
2. Бобровски С. Oracle 8: Архитектура. М.: ЛОРИ, 1999. 208 с.
3. Скотт Урман Oracle 8. Программирование на языке PL/SQL. Руководство для программистов Издательство: Лори, 1999 г., 610 стр.
4. Кристофер Аллен 101 Oracle PL/SQL. Как писать мощные и гибкие программы на PL/SQL Издательство: Лори, 2001 г., 350 стр.

**Отчет по лабораторной работе № 3**  
**«Логическое программирование на языке GNU Prolog»**

дата	Оценка (max 5)	Бонус за сложность	подпись
------	-------------------	-----------------------	---------

**Цели работы:**

Получить практические навыки применения систем и языков логического программирования для построения систем, основанных на знаниях. Создать экспертную систему по диагностике и ремонту компьютеров с использованием среды GNU Prolog

**Задачи работы:**

На языке Пролог реализовать базу знаний на выбранную тему, с использованием прямого либо обратного логического вывода. База знаний должна обязательно включать несколько уровней рассуждений (т.е. окончательные выводы не должны напрямую следовать из комбинаций входных данных, необходимо использовать промежуточные выводы) и демонстрировать некоторую интеллектуальность в принятии решений. Ориентировочное количество правил если-то – около 50 шт.

**Задание повышенной сложности (бонус за сложность – 10 баллов):**

Разработать полнофункциональную экспертную систему на языке Пролог (прототип модули ЭС на Си, файл lab1.doc) для синтеза технических решений по варианту комплексного задания.

**Краткий конспект теоретической части (изучите теоретическую часть файл lab3.doc)**

Что такое язык ПРОЛОГ \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Типы данных языка Пролога и их свойства \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Синтаксис программ на языке Пролог \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Понятие списков и их использование \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**PROLOG** (PROgramming in LOGic) – это язык программирования, используемый для решения задач, в которых действуют объекты и отношения между этими объектами. Предложен в районе 1972 года Alain Colmerauer и Philippe Roussel и на данный момент не стандартизирован. Существует много реализаций данного языка: Visual Prolog, PDC Prolog, Turbo Prolog, Prova, InterProlog, GNU Prolog. В рамках занятий используется GNU Prolog (доступен для скачивания на <http://www.gprolog.org/>). Возможно выполнение заданий на любой другой реализации языка.

На рис. 1. Представлен внешний вид главного окна GNU Prolog.

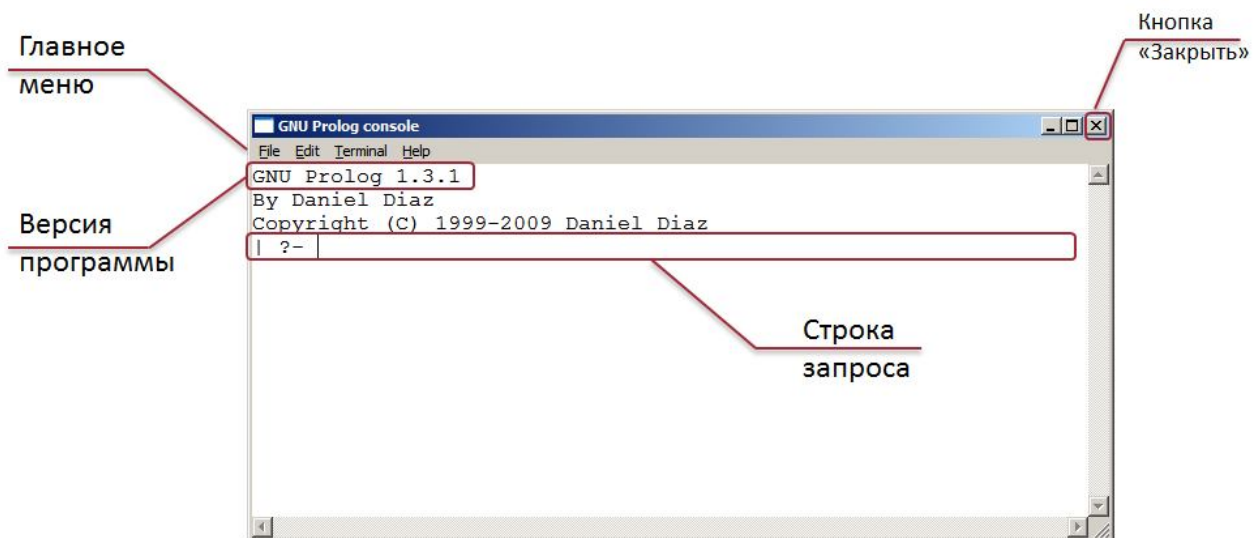


Рисунок 1 – Внешний вид главного окна GNU Prolog.

### Основные команды GNUProlog:

- [file\_name]. - загрузить файл с кодом программы (файл указывается без расширения «.pl»)
- **trace.** - включить режим трассировки программы.
- **notrace.** - выключить режим трассировки программы.
- **halt.** - завершить выполнение программы.
- [user]. - войти в режим интерактивного ввода программы. Control+D выход из режима интерактивного ввода программы

**Задание:** изучите теоретическую часть по ЛР, представленной на сервере <http://oracle.iu4.bmstu.ru>, скачайте и установите GNUProlog по следующей ссылке: <http://www.gprolog.org/setup-gprolog-1.3.1.exe>.

### Синтаксис языка Prolog

**Термы** — единица данных, объект данных в Prolog. Подразделяются на:

- Переменные
- Константы
- Числа
- Атомы
- Строки
- Списки
- Структуры
- Составные термы

**Переменными** в Прологе являются строки символов, цифр и символа подчеркивания, начинающиеся с заглавной буквы или символа подчеркивания. В случае, если имя переменной состоит из одного символа “\_”, переменная называется *анонимной*.

Область действия всех переменных, кроме анонимной, заключена в пределах высказывания. Анонимная переменная применяется, когда ее значение не используется в программе. Неоднократное употребление безымянной переменной в одном выражении применяется для того, чтобы подчеркнуть наличие переменных при отсутствии их специфической значимости.

Таким образом, в прологе переменные могут быть конкретизированы только один раз, например, выражение вида  $N = N + 1$  не имеет смысла. Подобные вещи в основном реализуются за счёт рекурсии.

**Числа** – в привычном понимании целые и действительные числа.

**Пример 1.** Вклейте результат выполнения произвольного запроса с объявлением чисел.

<pre>?- A = 12, B = 0.5e4. A = 12 B = 5000.0</pre>	
--	--

Пролог не производит вычисления автоматически, т.е. «2+3» будет восприниматься как терм. Для осуществления вычислений используется оператор «is».

**Пример 2.** Вклейте результаты выполнения нижеприведённых запросов

<pre>?- X = 2+3.  ?- X is 2+3.</pre>	
--	--

**Основные арифметические операторы:**

- + сложение
- вычитание
- \* умножение
- / деление
- // целочисленное деление
- mod** остаток от деления
- \*\* возведение в степень

**Операторы сравнения:**

- ==** равенство;
- =\=** неравенство;
- <** меньше;
- >** больше;
- >=** больше либо равно;
- =<** меньше либо равно.

**Задание 1.** Разработайте запрос для вычисления среднего арифметического двух чисел.

--

**Задание 2.** Разработайте запрос для вычисления дискриминанта квадратного уравнения.

--

**Атомы** – это любая последовательность символов, заключённая в одинарные кавычки (апострофы).

**Пример 3.** Вклейте результат выполнения приведённого запроса.

```
?- A = abc, B = 'Hello World'.  
   A = abc  
   B = 'Hello World'
```

**Строки** – списки кодов символов в кодировке ASCII, из которых состоит данное выражение. Таким образом, к строкам применимы все те же операции, что и к спискам.

**Пример 4.** Вклейте результат выполнения приведённого запроса.

```
?- S = "Hello World!".  
   S = [72,101,108,108,111,32,87,111,114,108,100,33]
```

**Список** - последовательность термов, заключенная в квадратные скобки.

**Пример 5.** Вклейте результат выполнения приведённого запроса.

```
?- A=[], B=[a, tro, 123, lolol, "hello!"], C=[A,B].  
   A = []  
   B = [1,tro,123,lolol,[72,101,108,108,111,33]]  
   C = [[],[1,tro,123,lolol,[72,101,108,108,111,33]]]
```

Это может быть любой пустой список, не содержащий ни одного элемента, либо структура, имеющая два компонента - **голову** и **хвост**. Пустой список не имеет ни головы, ни хвоста. Для расщепления списка в Прологе введена специальная форма - символ “|”:

Выражение	Голова	Хвост
[a, b, c]	a	[b, c]
[a]	A	[ ]
[ ]	Нет головы	Нет хвоста
[[a, b], c]	[a, b]	[c]
[a, [b,c]]	A	[[b, c]]

**Пример 6.** Вклейте результат выполнения приведённого запроса.

```
?- Head = iu4, Tail = [f,o,r,e,v,e,r], Fool = [Head | Tail].  
   Fool = [iu4,f,o,r,e,v,e,r]  
   Head = iu4  
   Tail = [f,o,r,e,v,e,r]
```



**Задание 3.** Разработайте запрос, выделяющий голову и хвост из списка *Fool* из примера 6.

**Задание 4.** Разработайте запрос, присваивающий переменной *T* последние 3 символа списка *Fool* из примера 6.

Знак "=" обозначает не императивное равенство (присвоение), а *унификацию* (что в других языках называется сопоставление с образцом), т.е. сопоставление левой и правой части и в случае удачного сопоставления конкретизация неизвестных значений.

**Пример 7.** Вклейте результат выполнения приведённого запроса.

```
?- tro=lolo.  
   no                                     %атомы несовместимы  
?- trololo=trololo.  
   yes                                    %сопоставление удачно  
?- 1=X.  
   X=1  
   yes                                    %атомы несовместимы  
?- [1,2| T] = [1,B,3,4].  
   T = [3,4]  
   B = 2
```

## Логические операторы.

Функтор “;” – логическое «И».

**Пример 8.** Вклейте результат выполнения приведённого запроса.

<pre>?- A = 2, B = 3, A = B. no</pre>	
---	--

Функтор “;” – логическое «ИЛИ».

**Пример 9.** Вклейте результат выполнения приведённого запроса.

<pre>?- A = 2; A = 5. A = 2 A = 5</pre>	
---	--

В данном случае система выдаёт неопределённый ответ. А может принимать как значение 2, так и 5.

Одной из особенностей языка Пролог является **backtracking** (перебор с возвратом). Добавим конкретики в предыдущий пример.

**Пример 10.** Вклейте результат выполнения приведённого запроса.

<pre>?- (A=2; A=5), A mod 2 =:= 0. A = 2 no</pre>	
---	--

Для данного примера система рассуждает следующим образом — допустим  $A=2$ , проведем проверку на  $A \bmod 2 =:= 0$  — это выполняется  $\Rightarrow$  сопоставление удачно — выполняется конкретизация ( $A=2$ ), далее происходит откат (возврат) к первой части утверждения и проверяется гипотеза  $A = 5$ , что является ложью и система возвращает «no». Возврат после первого удачного сопоставления можно отменить с помощью оператора отсечения “!”.

**Задание 5.** Модифицируйте пример 10 таким образом, чтобы система выходила из цикла сразу же после первого удачного сопоставления.

--

## Программа на языке Prolog

До этого мы имели дело с интерактивным режимом. Программа должна храниться в текстовом файле с расширением “.pl”. Для загрузки файла используется следующая команда:

```
[file_name].
```

где *file\_name* – имя файла программы **без расширения “.pl”**.

Программа на Прологе обычно представляет собой совокупность фактов и предикатов. Таким образом, рассмотренные далее структуры должны быть занесены в текстовый файл, после чего загружены для исполнения в оболочке.

### Структуры

**Структура** – конструкция следующего вида:

```
food(fruit, banana).      %банан – это фрукт, он съедобен
food(fruit, kiwi).       %киви – это фрукт, он съедобен
food(vegetable, cabbage). %капуста – это овощ, она съедобна
```

Структура в прологе представляется **функтором** (имя структуры - food) и **параметрами** (fruit, banana). Число параметров называется **арностью** функтора.

**Утверждения** подразделяются на факты и предикаты (правила).

**Факт** - это одиночная цель, которая, безусловно, истинна:

```
primacy(man).            % человек – примат
primacy(monkey).         % обезьяна – примат
```

**Предикат (правило)** состоит из одной головной цели и одной или более хвостовых целей, которые истинны при некоторых условиях:

```
can_eat(X) :- primacy (X).
```

Т.е., чтобы что-то могло есть, оно должно быть приматом.

Примеры описания фактов и предикатов представлены см. на следующей странице.

## Задание 6. Разработайте базу знаний к Вашему примеру из Лабораторной работы №1

*% ТИПЫ ЖИВЫХ СУЩЕСТВ: человек, обезьяна*

primacy(man).

primacy(monkey).

*% ИМЕНА СУЩЕСТВ И ПРИНАДЛЕЖНОСТЬ ИХ К ВИДУ*

exemplar(monkey,chicky).

exemplar(man,mark).

exemplar(man,yud).

*% КТО И ГДЕ ЖИВЁТ*

living(yud,russia).

living(mark,america).

living(chicky,africa).

*% КТО ЧТО ЕСТ*

eating(chicky,banana).

eating(chicky,kiwi).

eating(yud,apple).

eating(yud,lemon).

eating(mark,onion).

eating(mark,potato).

eating(mark,cabbage).

*% ПРЕДИКАТ, ПОКАЗЫВАЮЩИЙ ИНФУ О ЖИВЫХ СУЩЕСТВАХ*

*% X - что за примат*

*% Y - его имя*

*% Z - где живёт*

*% E - что ест*

alive(X,Y,Z,E):-

primacy(X),

exemplar(X,Y),

living(Y,Z),

eating(Y,E).

*% еда: тип (фрукт/овощ), имя*

food(fruit,banana).

food(fruit,lemon).

food(fruit,apple).

food(fruit,kiwi).

food(vegetable,onion).

food(vegetable,potato).

food(vegetable,cabbage).

*% подробная инфа о еде: имя, вкус, цвет, форма.*

description(banana,sweet,yellow,long).

description(lemon,sour,yellow,round).

description(apple,sweet,red,round).

description(kiwi,sour,green,round).

description(onion,bitter,yellow,round).

description(potato,sweet,brown,round).

description(cabbage,sweet,green,round).

```
% предикат, показывающий инфу о хавчике
% X - тип еды (фрукт/овощ)
% Y - название еды
% A - вкус
% B - цвет
% C - форма
havchik(X,Y,A,B,C):-
    food(X,Y),
    description(Y,A,B,C).
```

Вставьте скриншоты результатов выполнения запросов к вашей базе знаний.

## Трассировка программы

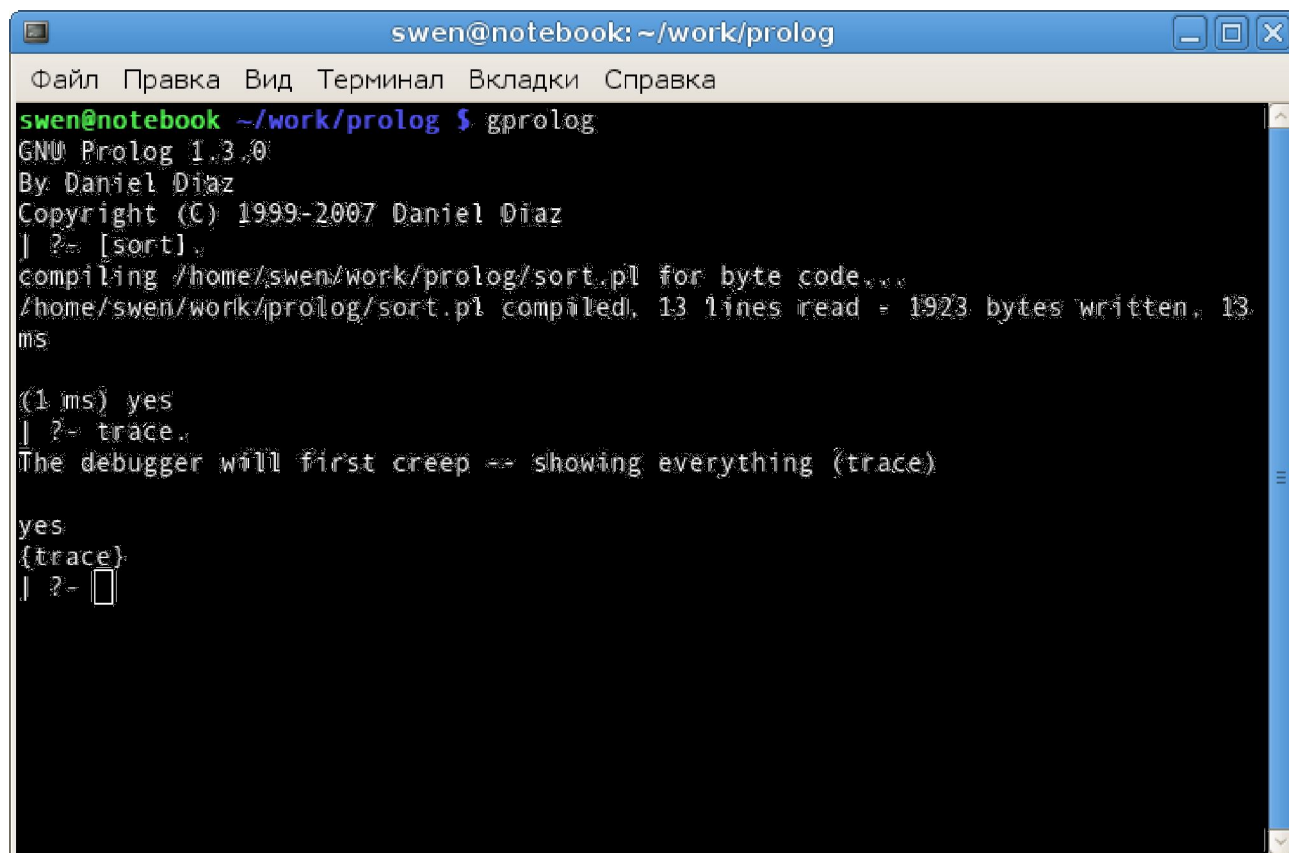
**Трассировка** – пошаговое выполнение программы. Для входа в режим трассировки необходимо выполнить одно из следующих действий:

- нажать “**Ctrl+C**”, затем ввести команду “**t**”;
- выполнить команду “**trace.**”

В качестве примера, рассмотрим трассировку программы быстрой сортировки.

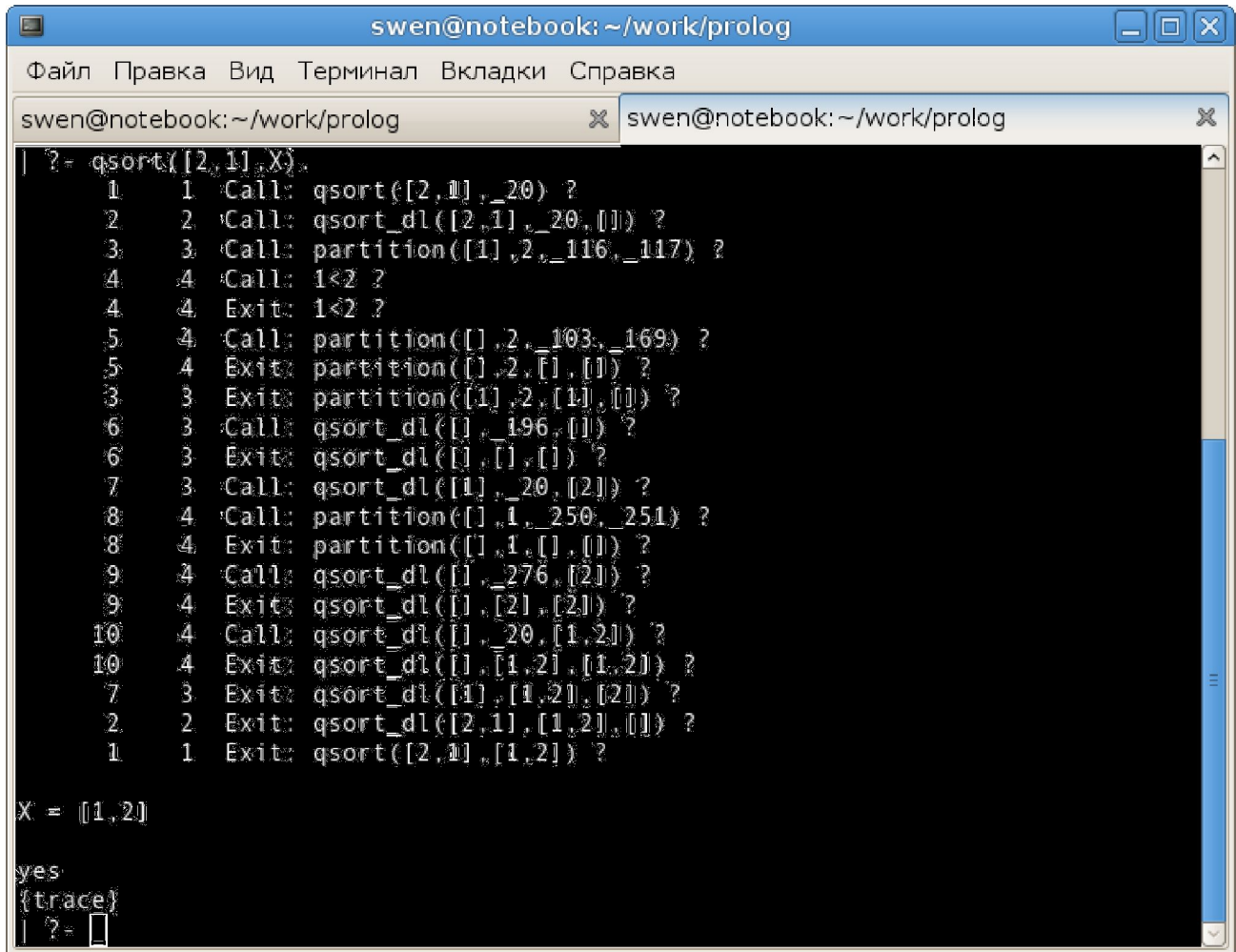
```
qsort( IN, OUT ) :- qsort_dl( IN, OUT, [] ).
qsort_dl( [X|M], R, R0 ) :-
    partition( M, X, M1, M2 ),
    qsort_dl( M2, R1, R0 ),
    qsort_dl( M1, R, [X|R1] ).
qsort_dl( [], R, R ).
partition( [X|M], Y, [X|M1], M2 ) :-
    X<Y, !, partition( M, Y, M1, M2 ).
partition( [X|M], Y, M1, [X|M2] ) :-
    partition( M, Y, M1, M2 ).
partition( [], _, [], [] ).
```

## Загрузка программы



```
swen@notebook: ~/work/prolog
Файл Правка Вид Терминал Вкладки Справка
swen@notebook ~/work/prolog $ gprolog
GNU Prolog 1.3.0
By: Daniel Diaz
Copyright (C) 1999-2007 Daniel Diaz
| ?- [sort].
compiling /home/swen/work/prolog/sort.pl for byte code...
/home/swen/work/prolog/sort.pl compiled, 13 lines read = 1923 bytes written, 13.
ms
(1 ms) yes
| ?- trace.
The debugger will first creep => showing everything (trace)
yes
{trace}
| ?- 
```

## Режим пошагового выполнения



```
swen@notebook: ~/work/prolog
Файл Правка Вид Терминал Вкладки Справка
swen@notebook: ~/work/prolog
| ?- qsort([2,1],X).
1 1 Call: qsort([2,1],_20) ?
2 2 Call: qsort_dl([2,1],_20,[1]) ?
3 3 Call: partition([1],2,_116,_117) ?
4 4 Call: 1<2 ?
4 4 Exit: 1<2 ?
5 4 Call: partition([],2,_103,_169) ?
5 4 Exit: partition([],2,[],[1]) ?
3 3 Exit: partition([1],2,[1],[1]) ?
6 3 Call: qsort_dl([],_196,[1]) ?
6 3 Exit: qsort_dl([],[],[1]) ?
7 3 Call: qsort_dl([1],_20,[2]) ?
8 4 Call: partition([],1,_250,_251) ?
8 4 Exit: partition([],1,[],[1]) ?
9 4 Call: qsort_dl([],_276,[2]) ?
9 4 Exit: qsort_dl([],[],[2]) ?
10 4 Call: qsort_dl([],_20,[1,2]) ?
10 4 Exit: qsort_dl([],[],[1,2]) ?
7 3 Exit: qsort_dl([1],[],[2]) ?
2 2 Exit: qsort_dl([2,1],[],[1]) ?
1 1 Exit: qsort([2,1],[],[1,2]) ?

X = [1,2]

yes
{trace}
| ?- 
```

**Задание 7.** Выполните трассировку программы к своему примеру и вставьте скриншот с результатом её выполнения.

## Контрольные вопросы

1. Формальная логика. Понятие. Суждение. Высказывание. Формальный язык, грамматика формального языка.
2. Логика предикатов. Интерпретация и унификация. Фразы Хорна. Принцип резолюции.
3. Парадигма декларативного программирования. Представление знаний с помощью фактов и правил. Управление вычислениями.
4. Простые и составные запросы. Понятие анонимной переменной.
5. Объекты данных. Структурирование множества объектов данных.
6. Рекурсивные вычисления.
7. Итерационные вычисления.
8. Рекурсивные структуры данных: списки. Способы обработки и примеры использования.
9. Рекурсивные структуры данных: деревья. Способы обработки и примеры использования.
10. Встроенные предикаты обработки символьных данных.
11. Встроенные предикаты управления вычислениями.
12. Способы представления баз знаний.
13. Представление баз знаний с использованием рекурсивных структур данных.
14. Создание графических изображений средствами языка Пролог.
15. Поиск на графах пространства состояний. Поиск в глубину и ширину.
16. Поиск на графах пространства состояний. Эвристический поиск.
17. Применение Пролога в естествознании.
18. Экспертные системы на правилах.
19. Экспертные системы на логике.

## Список литературы

1. Адаменко А. Логическое программирование и Visual Prolog (+CD-ROM). BHV.
2. Попов Э.В. Экспертные системы: решение информационных задач в диалоге с ЭВМ. М.: Наука, 1987. 283 с.
3. Астапкин Н.И., Матвеев М.Г. Синтез задач ситуационного управления перерабатывающим предприятием // Научно-техн. сб. мясной и холодильной промышленности РА сельскохоз. наук. 1994. № 2. С. 19-22.
4. Айзерман М.А., Алесекров Ф.Т. Выбор вариантов. Основы теории. М.: Наука, 1990. 240 с.
5. Месарович М., Такахара Я. Общая теория систем: математические основы. М.: Мир, 1978. 311с.
6. Шоломов Л.А. Логические методы исследования дискретных моделей выбора. М.: Наука, 1989. 288 с.
7. Матвеев М.Г., Сысоев В.В. Концепция информационных технологий управления перерабатывающими производствами // Информационная бионика и моделирование. М.: ГОСИФТП РАН, 1995. С. 25-31.
8. Минский М. Фреймы для представления знаний. М., 1979.
9. Робототехника и гибкие автоматизированные производства: В 9 кн. Кн. 6. Техническая имитация интеллекта / В.М. Назаретов, Д.П. Ким. Под ред. И.М. Макарова. М.: Высш. шк., 1986. 144 с.
10. Марселлус Д. Программирование экспертных систем на Турбо-ПРОЛОГЕ. М.: Финансы и статистика, 1994. 256 с.
- Иван Братко «Программирование на языке Пролог для систем искусственного интеллекта»
11. Марселлус «Программирование экспертных систем на Прологе»
12. Нейлор «Как построить свою экспертную систему»



Отчет по лабораторной работе № 4 «Примеры использования пакета NNTool»			
дата	Оценка (max 5)	Бонус за сложность	подпись

**Цель работы:** Знакомство с пакетом NNTols среды Matlab.

**Решаемые задачи:**

1. Исследование применения нейронной сети с прямой передачей сигнала для реализации логической функции согласно варианту задания.
2. Исследование применения нейронной сети с прямой передачей сигнала для решения задачи аппроксимации функции.
3. Исследование применения нейронной сети для решения задачи классификации входных векторов.
4. Изучение областей применения нейронных сетей по демопримерам пакета NNTools.
5. Задание повышенной сложности: реализация нейронной сети без использования пакета NNTools.

**Краткий конспект теоретической части** (изучите теоретическую часть файл lab4.doc)

Что такое нейрон и нейронная сеть \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

Виды активационных функций \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

Топологии нейронных сетей \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

Архитектуры нейронных сетей \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

Алгоритмы обучения нейронных сетей \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_



### 1.6. Листинг М-файла:

Строка листинга	Комментарии (что задает данная строка)

Выводы: \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### 2. Нейронная сеть с прямой передачей сигнала. Аппроксимация функции.

**Задача:** создать и обучить нейронную сеть выполнению операции (функции) \_\_\_\_\_ если заданы последовательности входа \_\_\_\_\_ и цели \_\_\_\_\_, оценить погрешность обучения.

- 2.1. Задание входных векторов \_\_\_\_\_
- 2.2. Задание выходных векторов \_\_\_\_\_
- 2.3. Создание нейронной сети

	Тип сети (Network Type) - _____ Входные диапазоны (Input ranges) - _____ Количество нейронов (Number of neurons) - _____ Передаточная функция (Transfer function) – _____ Функция обучения (Learning function) – _____ Performance function – _____ _____
--	---

Номер слоя	Количество нейронов (Number of neurons)	Передаточная функция (Transfer function)

## 2.4. Обучение сети

Inputs	Targets	Количество эпох (epochs)	Период обновления (show)	Время обучения (time)	Достижение цели (goal)

## 1.5. График кривой обучения:

	Выводы: _____
	_____
	_____
	_____
	_____
	_____
	_____
	_____
	_____
	_____

### Веса и смещения:

---

---

---

---

---

---

## 1.6. Листинг М-файла:

Строка листинга	Комментарии (что задает данная строка)
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Выводы: \_\_\_\_\_

---

---

---

---



1.8. График классификации:

	Выводы: _____ _____ _____ _____ _____ _____ _____ _____ _____
--	---

**4. 4. Изучение областей применения нейронных сетей по демопримерам пакета NNTools**

**Задание 1:** Согласно своему варианту (номеру по списку) выберите из перечня демонстрационных примеров (файл lab4.doc или по команде help nndemos), включенных в ППП NNT. Проведите исследование выбранного примера, результаты исследования занесите в таблицы.

4.1. Описание выполняемой задачи

---

---

---

4.2. Задание входных векторов

4.3. Задание выходных векторов

4.4. Создание нейронной сети

	Тип сети (Network Type) - _____ Входные диапазоны (Input ranges) - _____ Количество нейронов (Number of neurons) - _____ Передаточная функция (Transfer function) – _____ Функция обучения (Learning function) – _____ Performance function – _____
--	--

Номер слоя	Количество нейронов (Number of neurons)	Передаточная функция (Transfer function)

#### 4.5. Обучение сети

Inputs	Targets	Количество эпох (epochs)	Период обновления (show)	Время обучения (time)	Достижение цели (goal)

#### 4.5. Графическая интерпретация результатов

	Выводы: _____
	_____
	_____
	_____
	_____
	_____
	_____
	_____
	_____
	_____

#### 4.6. Листинг М-файла:

Строка листинга	Комментарии (что задает данная строка)
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Выводы: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

---

---

Отчет по лабораторной работе № 5			
«Математические модели искусственных нейронных сетей: сеть Хемминга»			
дата	Оценка (max 5)	Бонус за сложность	подпись

### Цель работы:

Целью работы является изучение математической модели нейронной сети, наиболее часто используемой в технических приложениях. Для первого ознакомления с нейронными сетями предлагается практическая задача отнесения объекта к одному из заданных классов - задача распознавания образов. Исследуются особенности функционирования нейронных сетей двух разных типов - рекуррентной и прямого распространения - для решения поставленной задачи. Лабораторная работа выполняется на эмуляторах нейронных сетей, реализованных в среде MATLAB

### Решаемые задачи:

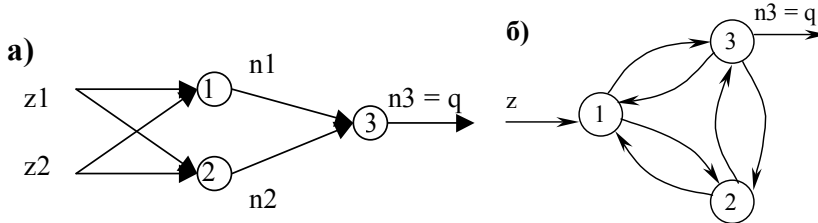
1. Подготовить данные и осуществить запуск сети Хемминга с использованием сети MAXNET прямого распространения.
    - 1.1. Ввести эталонные изображения представителей классов.
    - 1.2. Ввести изображение текущего классифицируемого объекта.
    - 1.3. Проверить правильность установки синаптических коэффициентов всех нейронов рабочего слоя сети Хемминга.
    - 1.4. Проконтролировать значения синаптических коэффициентов сети MAXNET прямого распространения.
    - 1.5. Осуществить запуск сети. Проанализировать результат классификации и значения выходов нейронов рабочего слоя и сети MAXNET.
    - 1.6. Повторить пункты 1.1-1.5 для разных вариантов задания текущего объекта.
  2. Провести исследование возможности классификации объекта в зависимости от степени отличия его признаков от эталонного представителя класса.
  3. Провести исследование устойчивости классификации объекта при изменениях значений синаптических коэффициентов рабочего слоя и сети MAXNET.
  4. Подготовить данные и осуществить запуск сети Хемминга с использованием рекуррентной сети MAXNET.
    - 4.1. Ввести эталонные изображения представителей классов.
    - 4.2. Ввести изображение текущего классифицируемого объекта.
    - 4.3. Проверить правильность установки синаптических коэффициентов всех нейронов рабочего слоя сети Хемминга.
    - 4.4. Проконтролировать значения синаптических коэффициентов рекуррентной сети MAXNET.
    - 4.5. Осуществить запуск сети. Проанализировать результат.
    - 4.6. Повторить пункты 4.1-4.5 для разных вариантов задания текущего объекта.
  5. Проанализировать время принятия решения о принадлежности объекта классу в зависимости от значения параметра  $\epsilon$  и различия значений потенциалов нейронов рабочего слоя. Для этого воспользоваться соответствующей модификацией синаптических коэффициентов рекуррентной сети MAXNET и текущих анализируемых объектов.
  6. Провести исследование, изложенное в пункте 2.
  7. Провести исследование, изложенное в пункте 3.
- Результаты экспериментов оформить в виде отчета.



**Краткий конспект теоретической части** (изучите теоретическую часть файл lab5.doc)

Какие особенности функционирования биологического нейрона учтены при построении математической модели технического нейрона \_\_\_\_\_

Постройте математическое описание нейронной сети, представленной на рисунке .



Какая нейронная сеть называется рекуррентной \_\_\_\_\_

Объясните особенности функционирования нейронной сети прямого распространения \_\_\_\_\_

Объясните принцип функционирования стохастического нейрона. \_\_\_\_\_

Какую задачу решает сеть Хемминга? \_\_\_\_\_

Объясните принцип построения сети MAXNET прямого распространения при произвольном числе входов с использованием нейросетевого компаратора на два входа. \_\_\_\_\_

**Подготовка к проведению лабораторной работы:**

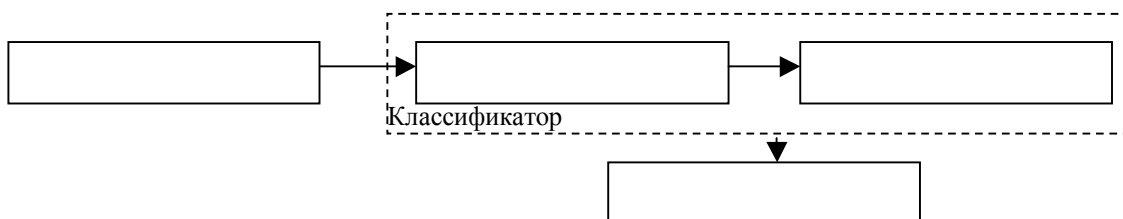
Программное обеспечение реализовано в среде MATLAB 5.0 в качестве прикладных пакетов системы MATLAB. Поэтому для установки и запуска этих программ на Вашем компьютере должна быть установлена система MATLAB.

1. Создайте каталог с именем hemming.
2. Скопируйте в созданный каталог hemming все файлы из архива.
3. Запустите систему MATLAB.
4. Добавьте полное имя созданного каталога `synaps` к пути поиска системы MATLAB, выполнив из командной строки MATLAB команду `addpath('имя_каталога')`. Например, если Вы создали каталог `C:\MATLAB\hemming`, команда будет выглядеть следующим образом: `addpath(C:\MATLAB\hemming)`.
5. Запустите пакет Hemming в симулинке набрав в командной строке `>>hemming`.
6. Выполните распознавание образов согласно варианту задания. В каждом варианте заданы эталонные представители трех классов (зарисуйте три класса образов согласно варианту задания).

Задание: три класса образов для распознавания


**В работе использовались нейронные сети:** сеть Хемминга, сеть MAXNET прямого распространения (feed-forward MAXNET), рекуррентная сеть MAXNET (recurrent MAXNET).

**Архитектура нейросетевого классификатора, используемого в работе:**



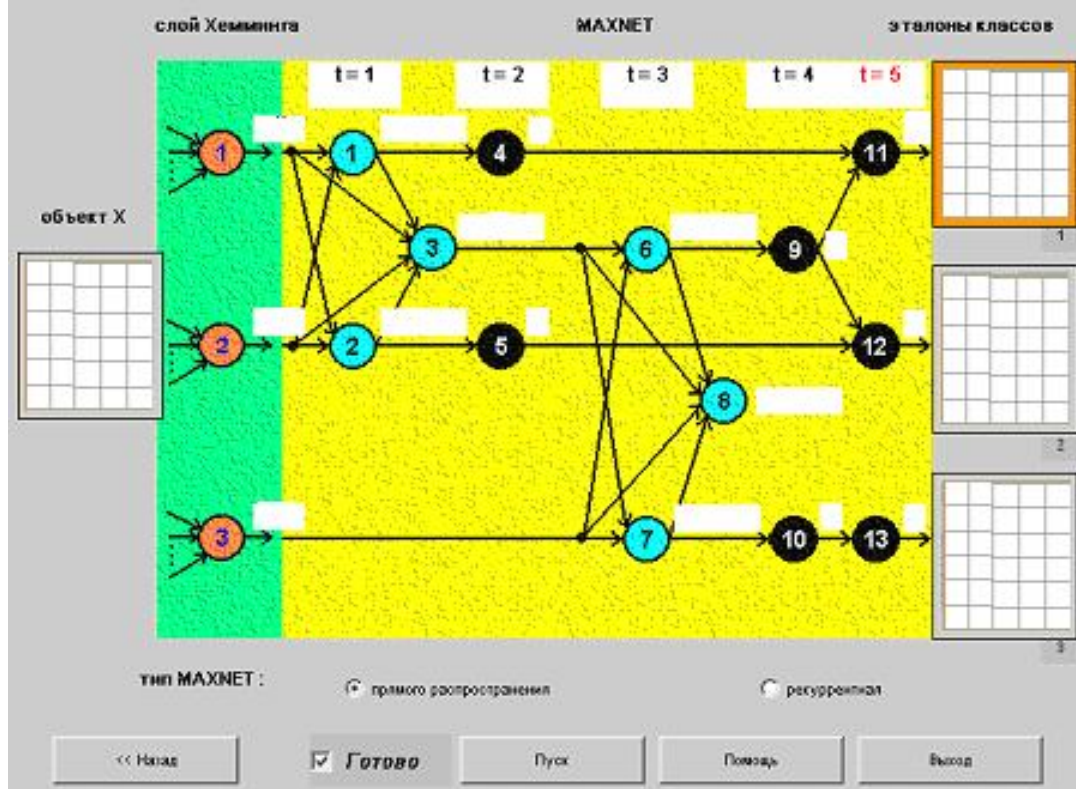
Зарисуйте модель искусственного нейрона, используемого в работе в слое Хемминга:

Зарисуйте схему компаратора, используемого в слое MAXNET (сеть прямого и обратного распространения):

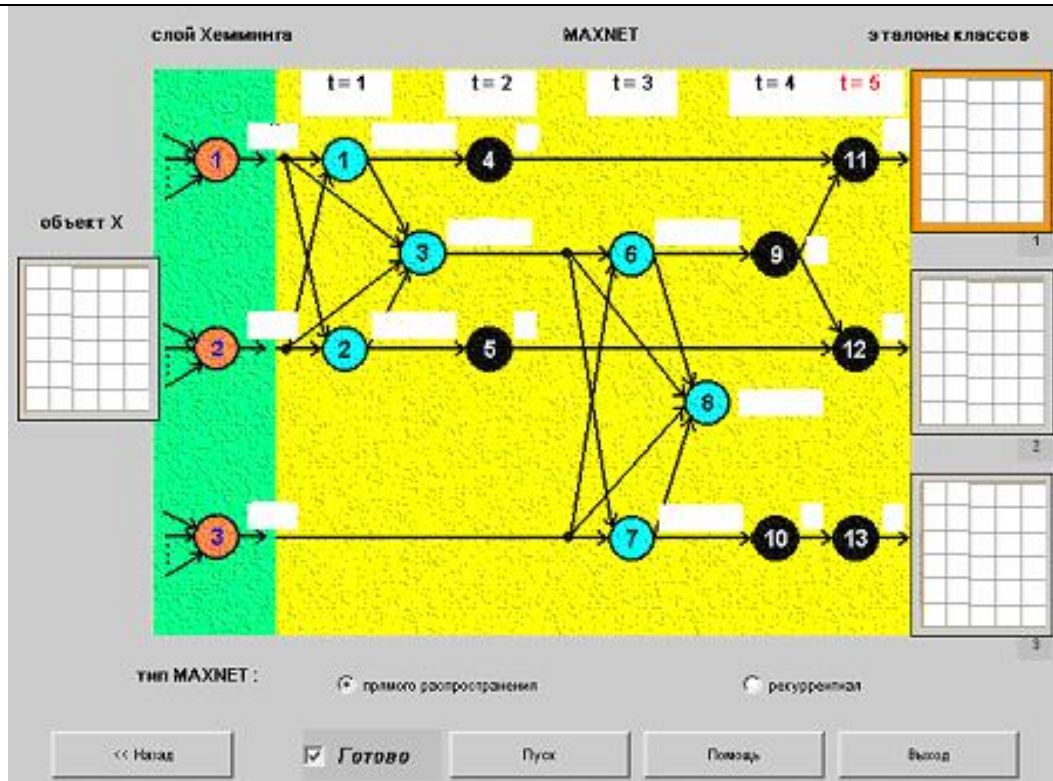
--	--

Описание принципов работы сети: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

1. Построение и обучение нейронной сети прямого распространения

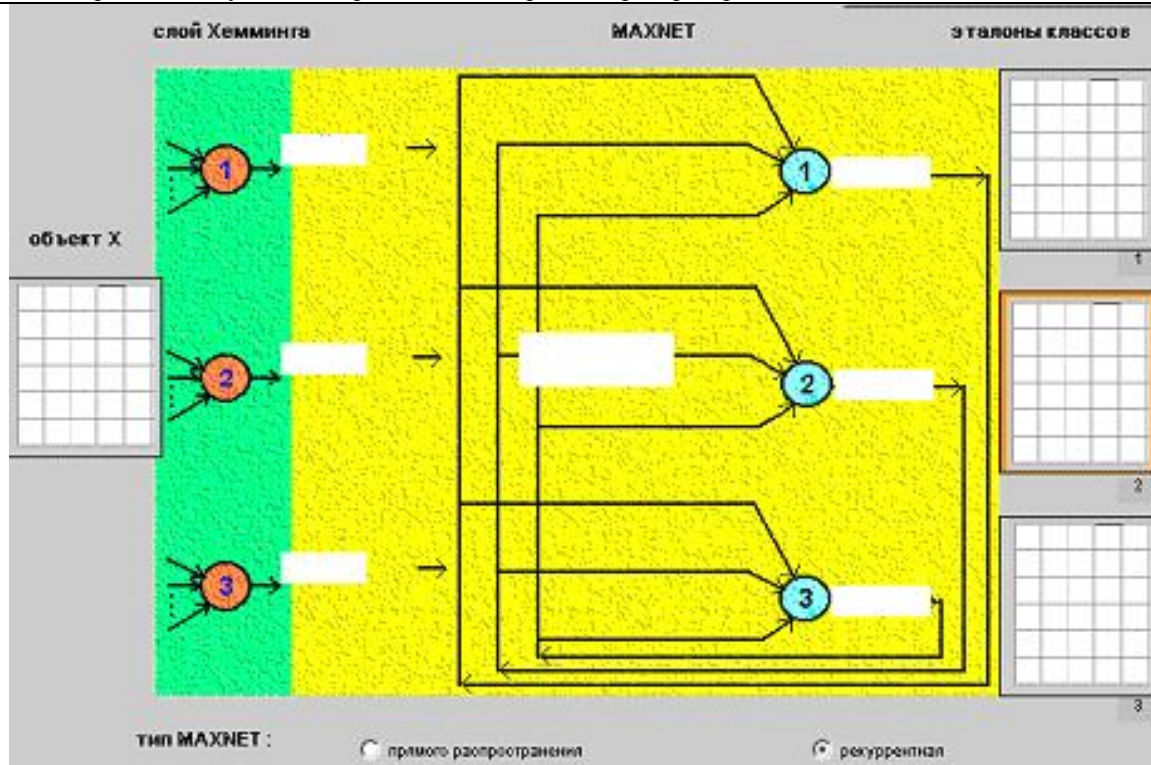


Выводы:

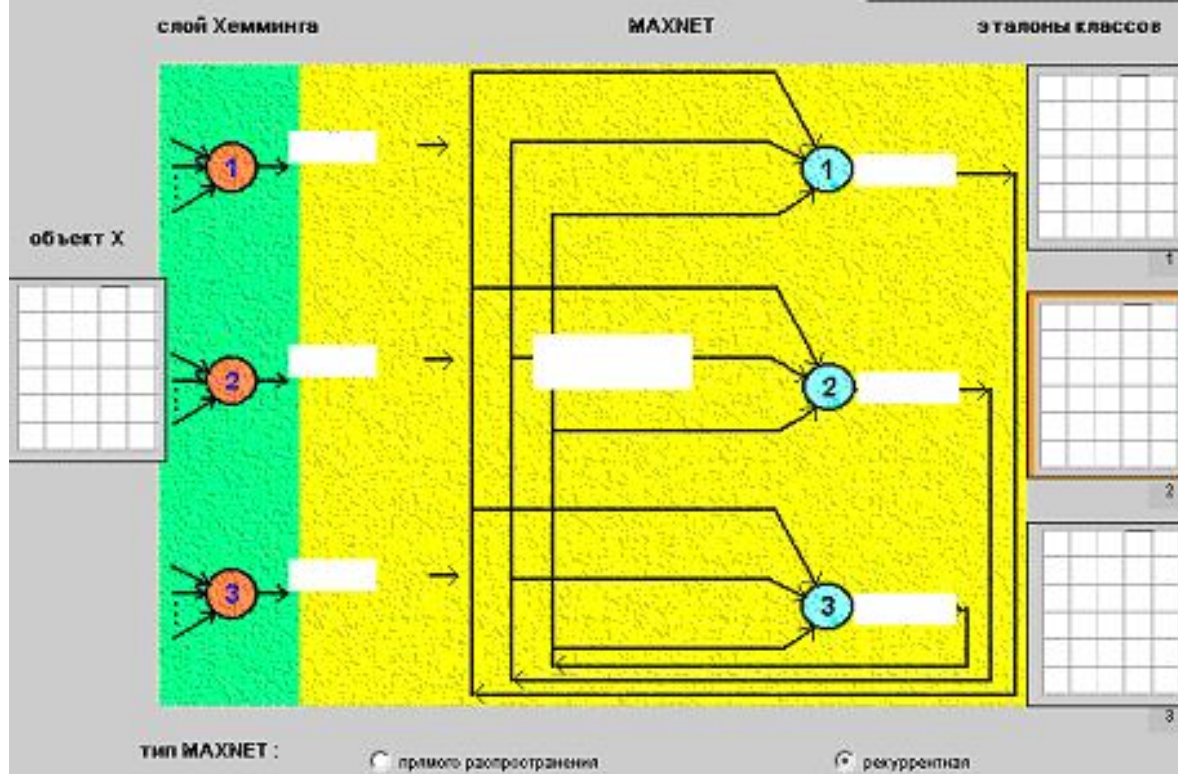


Выводы:

2. Построение и обучение нейронной сети обратного распространения



Выводы:



Выводы:



## Контрольные вопросы

1. Какие особенности функционирования биологического нейрона учтены при построении математической модели технического нейрона?
2. Напишите уравнение функционирования технического нейрона.
3. Что называется активационной характеристикой (передаточной функцией) нейрона? Приведите примеры.
4. Какими свойствами обладают сигмоидальные нелинейные преобразователи? Приведите примеры.
5. Что называется синаптическим коэффициентом  $w_{ij}$  математической модели нейронной сети?  $v_{ij}$ ?
6. Напишите уравнения, характеризующие динамику нейронной сети в дискретном времени (скалярная форма).
7. Напишите векторно-матричные уравнения, характеризующие динамику нейронной сети в дискретном времени.
8. Как записываются начальные условия для уравнений динамики нейронной сети?
11. Какая нейронная сеть называется рекуррентной?
12. Объясните особенности функционирования нейронной сети прямого распространения.
13. Какими особенностями обладает математическое описание многослойной нейронной сети?
14. Объясните принцип функционирования стохастического нейрона.
15. Какая активационная характеристика должна быть использована в детерминированном нейроне, который воспроизводит среднее значение выхода стохастического нейрона?
16. В чем состоит задача распознавания образов? Какая информация должна быть представлена об объекте при решении задачи распознавания образов?
17. Какую задачу решает сеть Хемминга?
18. В какой форме представлены характеристики объектов при решении задачи распознавания образов с помощью сети Хемминга?
19. Каким способом заданы классы при решении задачи распознавания образов с помощью сети Хемминга?
20. Сформулируйте критерий оптимальности отнесения объекта к одному из классов при применении сети Хемминга.
21. Какое содержание имеют выходные значения нейронов рабочего слоя сети Хемминга?
22. Сколько нейронов содержит рабочий слой сети Хемминга? Каковы их активационные характеристики?
23. Какими способами можно решать задачу поиска максимума среди  $K$  значений при построении сети Хемминга?
24. Нарисуйте схему нейросетевого компаратора на два входа и объясните его работу. Какие активационные характеристики нейронов используются в схеме нейросетевого компаратора?
25. Объясните принцип построения сети MAXNET прямого распространения при произвольном числе входов с использованием нейросетевого компаратора на два входа.
26. Нарисуйте схему рекуррентной сети MAXNET. Какие начальные условия устанавливаются на сети?
27. Какой параметр рекуррентной сети MAXNET может влиять на ее быстродействие?
28. Объясните принцип работы рекуррентной сети MAXNET.

Отчет по лабораторной работе № 6			
«Математические модели искусственных нейронных сетей»			
дата	Оценка (max 5)	Бонус за сложность	подпись

### Цель работы:

В этой лабораторной работе студент знакомится с математической моделью многослойной нейронной сети, постановкой задачи ее обучения, изучает метод обратного распространения ошибки для настройки параметров в процессе обучения. Теоретическое введение помогает понять технологию работы с эмулятором МНС в режимах обучения, тестирования и нормального функционирования.

### Решаемые задачи:

1. Разработать нейронную сеть осуществляющую преобразование координат на плоскости. Обучающая выборка хранится в текстовом файле `coord.dat`. Каждая строка файла содержит данные одного обучающего примера: координаты вектора в полярной системе координат (переменные  $\rho$  и  $\varphi$ ) и в декартовой системе (переменные  $x$  и  $y$ ):

$$y = \rho \cdot \cos(\varphi),$$

$$x = \rho \cdot \sin(\varphi).$$

Всего выборка содержит 100 примеров, в которых вектора расположены в первом квадрате и имеют длину от 0.05 до 1. Для удобства анализа файл содержит номера выборочных примеров (переменная `row`), которые при обучении нейронной сети не используются.

1. Создать задачу на базе выборочных данных `coord.dat`. Определить в процессе создания задачи сеть, содержащую 10-12 нейронов (например, 3 слоя с распределением нейронов по слоям 7, 3, 2, третий слой - выходной). Сохранить файл-задачу под уникальным именем. Создать файл для последующего тестирования сети (расширение `kgb`).

2. Обучить созданную сеть. В процессе обучения активно исследовать возможности варьирования значений параметров алгоритма обучения, фиксируя график изменения показателя ошибки ("Deviation") и отмечая качественное влияние выполненных настроек на процесс обучения. К числу варьируемых параметров относятся метод обучения, способ настройки (по примерам или по выборке в целом), скорость обучения.

По результатам проведенного исследования сделать выводы, которые должны быть включены в отчет по работе.

3. Изучить возможности режима анализа результата обработки данных нейронной сетью. Зарисовать и включить в отчет графики, полученные на разных стадиях обучения сети. Проанализировать динамику приближения графиков к желаемым зависимостям в процессе обучения сети.

4. Прервать обучение, когда число правильно обрабатываемых примеров составит примерно 80% от их общего числа. Провести тестирование сети на примерах, специально отобранных для этой цели при выполнении п.1. Оценить ошибку обобщения данных обученной сетью в каждом тестовом примере. Сделать выводы по результатам тестирования и включить их в отчет по работе.

5. Исследовать возможности нейроэмулятора SINAPS в режиме функционирования сети. Вывести результат обработки в выходной файл.

6. С учетом опыта, приобретенного в ходе выполнения п.п. 1-5, создать новый файл-задачу с целью получить сеть с улучшенными точностными показателями.

Провести обучение и тестирование сети. Результат исследования включить в отчет по работе.

**Краткий конспект теоретической части** (изучите теоретическую часть файл lab5.doc)

Что такое многослойная нейронная сеть \_\_\_\_\_

\_\_\_\_\_

Алгоритмы обучения многослойных нейронных сетей \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Топологии многослойных нейронных сетей \_\_\_\_\_

\_\_\_\_\_

Что называется «эпохой» в процессе обучения нейронной сети \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

В чем заключается этап подготовки данных для МНС \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

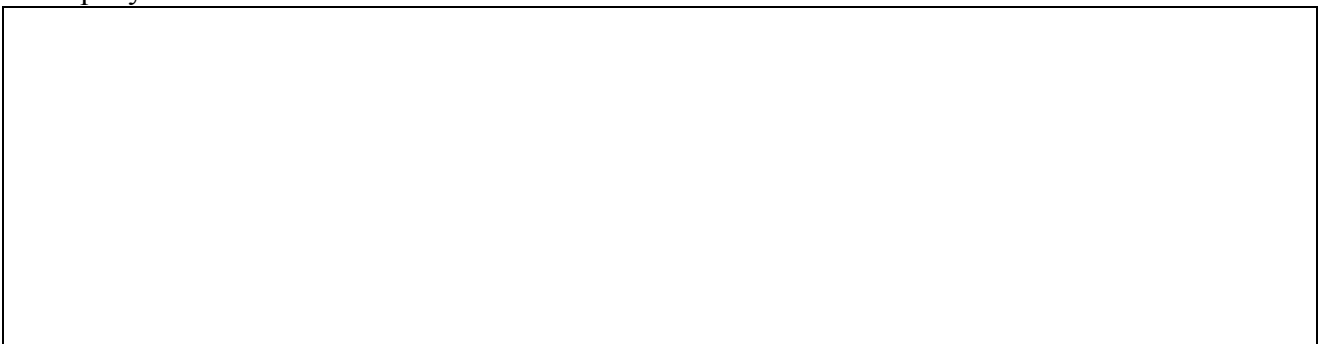
\_\_\_\_\_

**Подготовка к проведению лабораторной работы:**

Программное обеспечение реализовано в среде MATLAB в качестве прикладных пакетов системы MATLAB.

Загрузить MATLAB и указать путь к М-файлу лабораторной работы № 6, например, используя команду `addpath` (пример: `addpath c:\neuro\lab7`). Набрать в командной строке имя М-файла `mnpapps`. Указать тип решаемой задачи (аппроксимация функций, прогноз временных рядов или классификация). В трёх верхних полях указать пути и имена файлов с обучающей, тестовой и контрольной выборками последовательно.

1. Зарисуйте топологию МНС

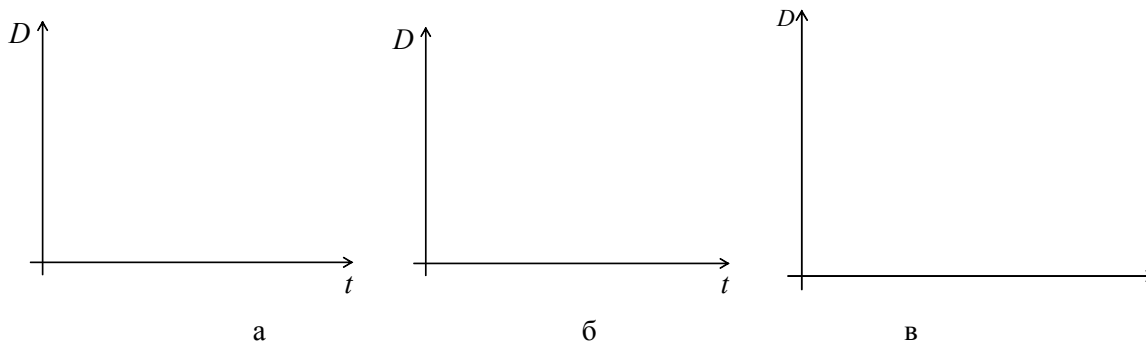




2. Построение и обучение многослойной нейронной сети, используя нейроэмулятор SINAPS.

Обучающая выборка для настраивания синаптических коэффициентов нейросети содержит \_\_\_\_\_ примеров, тестовая - \_\_\_\_\_. Обучение производилось с точностью: \_\_\_\_\_%.

График а) представляет зависимость критерия  $D$  от дискретного времени обучения, в ситуации, когда возникает «паралич» сети. Паралич наблюдается при \_\_\_\_\_. При выполнении исследования «паралич» *наблюдался / не наблюдался*. График  $D(t)$  б), соответствует слишком большой скорости обучения. В ходе выполнения лабораторной работы наблюдался график  $D(t)$  в).



3. Исследуем зависимость качественных параметров сети от числа слоев и количества нейронов в слоях

Результаты моделирования занесены в таблицу, в которой GF – число примеров, удовлетворяющих заданной точности, BF – не удовлетворяющих заданной точности.

*Наблюдалось / не наблюдалось* переобучение многослойной нейронной сети, реализованной в нейроэмуляторе SINAPS.

Способность сети к генерализации *зависит / не зависит* от числа *нейронов / слоев* в сети. С увеличением числа *нейронов / слоев* в сети способность к генерализации *увеличивается / уменьшается*. Наилучшая архитектура, решающая задачу \_\_\_\_\_, которую удалось добиться в лабораторной работе: \_\_\_\_\_ на нейроэмуляторе \_\_\_\_\_. Выбор лучшей архитектуры сети производился по следующим показателям: \_\_\_\_\_

---

---

---

---

---

---

---

---



## Контрольные вопросы

1. Нарисуйте схему и объясните особенности архитектуры многослойной нейронной сети.
2. Напишите уравнения функционирования многослойной нейронной сети.
3. Объясните состав данных таблицы обучающей выборки, используемой для настройки параметров нейронной сети.
4. Какие параметры многослойной нейронной сети настраиваются в процессе ее обучения?
5. Какой критерий используется для организации обучения многослойной нейронной сети?
6. Какой метод применяется для обучения многослойной нейронной сети в используемом в работе нейроэмуляторе?
7. Чем характеризуется эффект «паралича» при обучении многослойной нейронной сети?
8. В чем состоит и как реализуется процесс инициализации при обучении нейронной сети?
9. В чем состоит и как проверяется эффект «генерализации данных» в нейронной сети?
10. Какие параметры режима обучения многослойной нейронной сети доступны пользователю для настройки?
11. В чем состоит процедура тестирования обученной многослойной нейронной сети?
12. Какие переменные вычисляются с помощью метода обратного распространения ошибки и как они используются в процессе обучения многослойной нейронной сети?
13. Приведите примеры активационных характеристик нейронов, используемых в многослойных нейронных сетях.
14. Что называется «эпохой» в процессе обучения нейронной сети?
15. Почему разные реализации процесса обучения многослойной нейронной сети из разных начальных условий не приводят к одному и тому же финальному результату?
16. В чем состоит формальное правило построения структурной схемы сети обратного распространения ошибки по заданной схеме прямого распространения сигнала?
17. По какому правилу производится модификация значений параметров многослойной нейронной сети при ее обучении методом обратного распространения ошибки?

<b>Отчет по лабораторной работе № 7</b> <b>«Применение многослойной нейронной сети</b> <b>для решения прикладных задач обработки данных»</b>			
дата	Оценка (max 5)	Бонус за сложность	подпись

**Цель работы:**

Отработать применение многослойной нейронной сети для решения задачи обработки данных одного из трех перечисленных выше типов:

- аппроксимация функции многих переменных,
- прогноз значений временного ряда,
- классификация векторных данных.

Целью работы является построение многослойной нейронной сети и исследование ее точностных характеристик. Работа выполняется на нейроэмуляторе NeuroPro (имя М-файла mnpapps).

**Решаемые задачи:**

1. Определение типа решаемой задачи.
2. Визуализация данных в пакете MATLAB.
3. Анализ данных обучающей выборки.
4. Создание и обучение МНС
5. Тестирование МНС.
6. Определение работы сети для контрольных примеров.
7. Визуализация результатов обработки данных нейросетью.
8. Упрощение МНС.
9. Выбор лучшей архитектуры для решения поставленной задачи.

**Краткий конспект теоретической части** (изучите теоретическую часть файл lab5.doc)

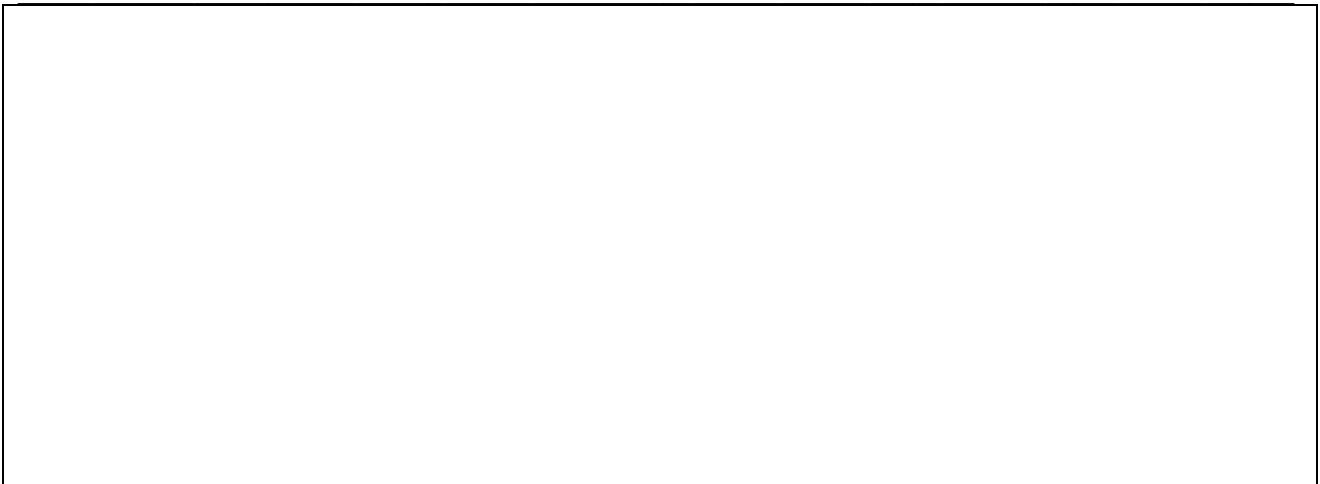
Постановка задачи аппроксимации функции многих переменных _____ _____ _____ _____
Постановка задачи прогноза значений временного ряда _____ _____ _____ _____ _____
Постановка задачи классификации векторных данных _____ _____ _____ _____ _____ _____
В чем заключается тестирование МНС _____ _____ _____ _____

**Исследуются следующие файлы с выборками:**

Выборка	Число примеров	Имя файла
Обучающая		Var_Train.txt
Тестовая		Var_Test.txt
Контрольная		Var_Control.txt

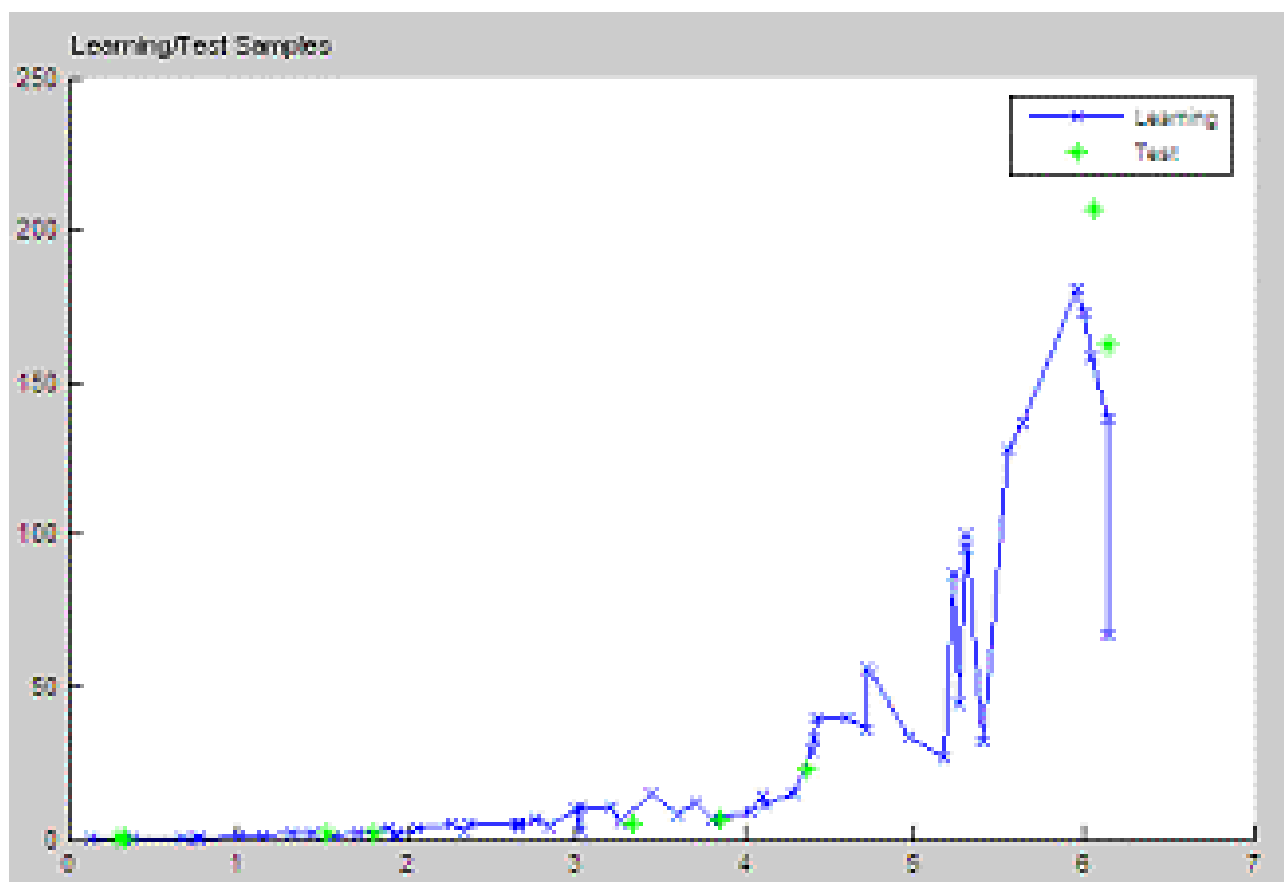
По структуре обучающей выборки удалось установить, что решается задача: \_\_\_\_\_

Зарисуйте топологию МНС для решения задачи типа \_\_\_\_\_



Визуализация результатов

На графике представлены обучающие и тестовые примеры:



Зарисуйте полученный график для вашей задачи

Анализ данных обучающей выборки.

- 1) \_\_\_\_\_
- 2) \_\_\_\_\_
- 3) \_\_\_\_\_
- 4) \_\_\_\_\_

Вывод: многослойная нейронная сеть, решающая поставленную задачу, должна содержать \_\_\_\_\_ слоев, т.к. \_\_\_\_\_.

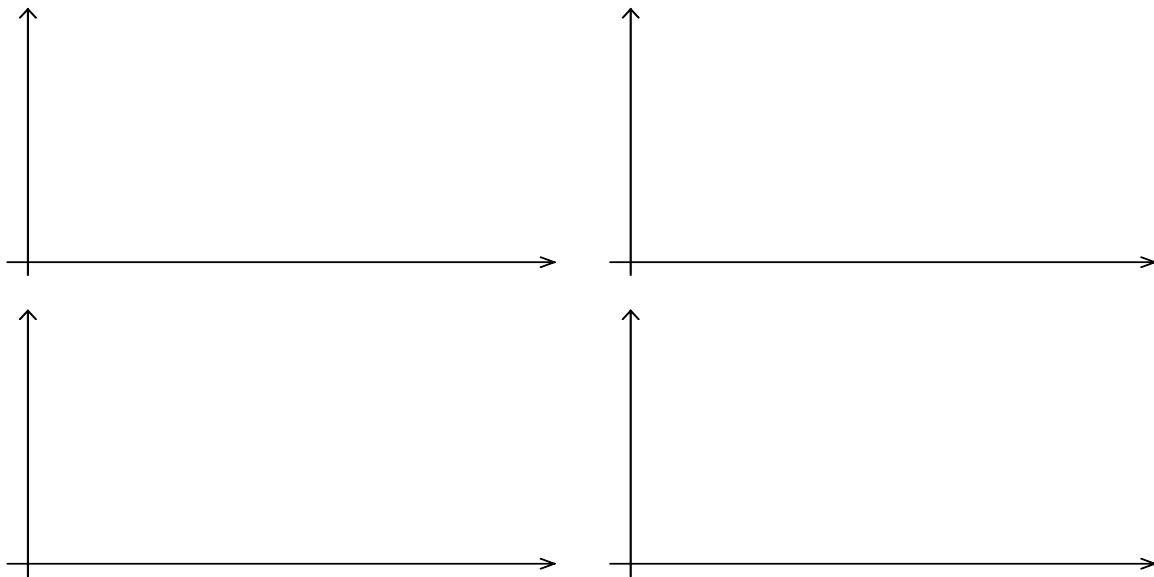
Т.к. исследуемые зависимости \_\_\_\_\_, то требуется сеть, содержащая около \_\_\_\_\_ нейронов по слоям для решения задачи с требуемой точностью \_\_\_\_\_%.

Построение и обучение многослойной нейронной сети, используя нейроэмулятор NEUROPRO.

Сеть обучалась в течение \_\_\_\_\_ тактов. Получены следующие результаты:

Параметр	Обучающая выборка	Тестовая выборка
GF		
BF		
Максимальная ошибка		
Средняя ошибка		

По результатам обработки сетью обучающей выборки построены следующие графики зависимости



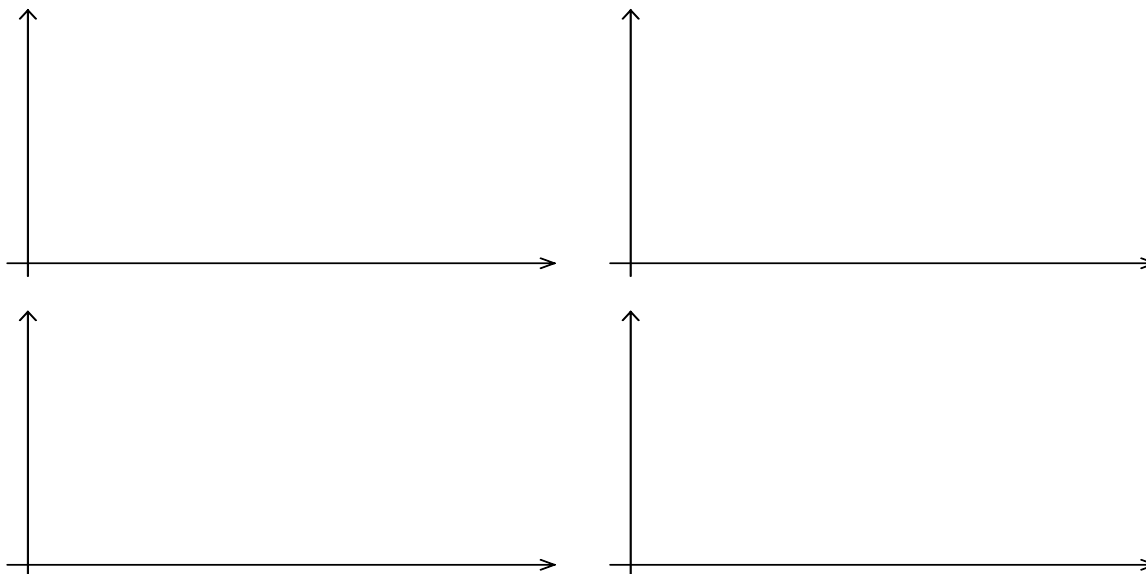
---

---

---

---

5. Обработка тестовой выборки позволила построить следующие графики зависимости\_



6. Результаты обработки сетью контрольных точек.

На входное воздействие контрольных примеров сеть выдала следующий результат:

№	
1	
2	
3	
4	
5	

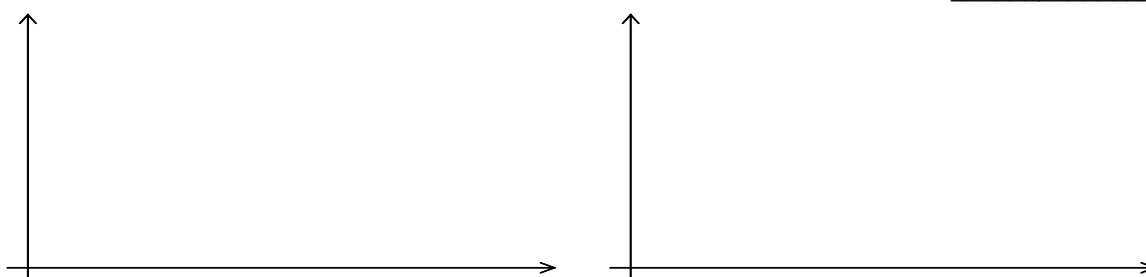
По расположению выходов сети на графиках обучающей выборки при воздействии контрольных примеров и по реакции сети на тестовую выборку можно сделать вывод о *слишком сильной генерализации сети / умеренной генерализации сети / слишком сильной подстройке коэффициентов сети под обучающую выборку*, что требует *построение более мощной сети / упрощения сети*.

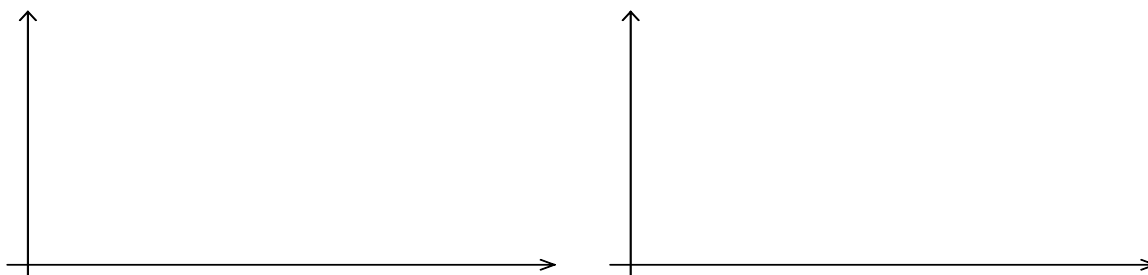
7. Построение МНС, более адекватной решаемой задаче.

В результате *упрощения сети, построенной в п.4 / построения и обучения более мощной сети*, которое длилось \_\_\_\_\_ тактов, сеть стала содержать \_\_\_\_\_ нейронов по слоям.

Параметр	Обучающая выборка	Тестовая выборка
GF		
BF		
Максимальная ошибка		
Средняя ошибка		

По результатам обработки сетью обучающей, тестовой и контрольной выборок построены следующие графики зависимости \_\_\_\_\_





На входное воздействие контрольных примеров сеть выдала следующий результат:

№	
1	
2	
3	
4	
5	

Выводы:


### Контрольные вопросы

1. Объясните содержание задачи аппроксимации функции многих переменных на многослойной нейронной сети.
2. В чем состоит этап подготовки данных для обучения многослойной нейронной сети в задаче аппроксимации функции многих переменных?
3. Объясните содержание задачи прогноза временных рядов на многослойной нейронной сети.
4. В чем состоит этап подготовки данных для обучения многослойной нейронной сети в задаче прогноза временных рядов?
5. Объясните содержание задачи классификации данных на многослойной нейронной сети.
6. В чем состоит этап подготовки данных для обучения многослойной нейронной сети в задаче классификации данных?
7. Какой критерий используется при решении задачи классификации данных с помощью многослойной нейронной сети?
8. Какие правила останова процесса обучения МНС применяются в практических приложениях?
9. В чем состоит функциональная задача дополнительного блока принятия решения на выходе МНС при использовании ее в качестве классификатора данных?
10. Почему чрезмерное увеличение числа нейронов скрытых слоев в МНС при решении задачи аппроксимации может ухудшить точностные показатели сети? Какое свойство нейронной сети имеется в виду?
11. Почему на задачи классификации данных и прогноза временного ряда распространяются те же рекомендации по выбору архитектуры МНС, что и в задаче аппроксимации функции многих переменных?
12. Какие Вы можете предложить нейросетевые решения задачи прогноза временного ряда одновременно на несколько последовательных временных тактов?
13. Предложите вариант построения нейросетевого классификатора при числе классов 4 с использованием многослойных нейросетевых классификаторов на 2 класса?



Отчет по лабораторной работе № 8 «Ассоциативная память. Сеть Хопфилда»			
дата	Оценка (max 5)	Бонус за сложность	подпись

### Цель работы:

В лабораторной работе исследуются свойства нейронной сети Хопфилда, которая рассматривается как модель ассоциативной памяти, позволяющей восстановить объект по ограниченному набору зашумленных признаков. Сеть Хопфилда относится к классу рекуррентных нейронных сетей. В асинхронном режиме работы нейронов сеть Хопфилда из произвольного начального состояния за конечное число тактов дискретного времени приходит в состояние устойчивого равновесия (аттрактор). Чисел различных аттракторов сети определяет ее объем «памяти» (число запомненных образов).

Экспериментальное исследование свойств сети Хопфилда выполняется в нейроэмуляторе, реализованном в системе MATLAB. Изучается динамика переходного процесса, методом статистического моделирования рассчитывается число аттракторов и размеры бассейнов аттракторов. Исследуется устойчивость решений при зашумлении данных.

### Решаемые задачи:

. Требуется исследовать четыре важных аспекта данной модели:

- динамика (процесс эволюции сети от начального состояния к устойчивому состоянию);
- аттракторы (множество устойчивых состояний сети, размер бассейнов аттракторов);
- ассоциативная память (способность сети сохранять в качестве аттракторов некоторое множество образов и релаксировать к сохраненному образу при подаче на вход этого же образа с искажениями);
- робастность (способность сети сохранять свойство ассоциативной памяти после искажения матрицы синаптических связей, т. е. образ с одинаковым уровнем шума правильно восстанавливается сетью до и после внесения искажений в синаптическую матрицу).

Сеть Хопфилда эмулируется на персональном компьютере с помощью программного обеспечения, специально разработанного для исследования указанных аспектов сети.

### Краткий конспект теоретической части (изучите теоретическую часть файл lab8.doc)

Что такое робастность модели \_\_\_\_\_

Дайте определение понятию «аттрактор» \_\_\_\_\_

Что такое ассоциативная память \_\_\_\_\_

Что такое динамика нейронной сети \_\_\_\_\_

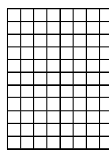
**Подготовка к выполнению работы:**

1. Создайте каталог с именем lab8.
2. Скопируйте в созданный каталог lab8 все файлы из архива hopfield.rar, находящегося в каталоге lab8\hopfield на дистрибутивном диске.
3. Запустите систему MATLAB.
3. Добавьте полное имя созданного каталога lab8 к пути поиска системы MATLAB, выполнив из командной строки MATLAB команду `addpath('имя_каталога')`.

В лабораторной работе изучается сеть Хопфилда, содержащая \_\_\_\_\_ нейронов. Ожидается, что эта сеть «запомнит» \_\_\_\_\_ входных образа.

1. Исследование сети Хопфилда, настроенной на один образ.

Нейронная сеть настраивается на образ:



Предположение (1): ожидается, что память данной сети будет представлена \_\_\_\_\_ позитивами, \_ негативами и *содержать/не содержать* ложную память.

1.1. Исследование динамики сети Хопфилда.

За один такт работы сети Хопфилда опрашивается \_\_\_\_\_ нейронов.

Начальное воздействие	Аттрактор	Кол-во тактов	Энергия	
			начало	конец
позитив «__»	позитив «__», негатив «__», ложн. память			
негатив «__»	позитив «__», негатив «__», ложн. память			
случ. вектор	позитив «__», негатив «__», ложн. память			

Вывод: значение энергии данной сети Хопфилда в процессе функционирования *увеличивается/уменьшается/остаётся неизменным* и *не достигает/достигает глобального/локального минимума/максимума* при попадании в аттрактор.

1.2. Исследование бассейна аттракторов сети Хопфилда.

Максимальное число случайных состояний, которые могут подаваться на сеть Хопфилда: \_  
 . В данном исследовании сеть стартовала из \_\_\_\_\_ случайных состояний.

Запоминаемый образ	Позитив, %	Негатив, %
Ложная память, суммарно, %		

Доли в бассейне аттракторов *соответствуют/не соответствуют* теоретическим, равным: «позитивы» - \_\_\_\_\_%, «негативы» - \_\_\_\_\_%, «ложная память» - \_\_\_\_\_%. Предположение (1) *подтвердилось/не подтвердилось*.

1.3. Исследование ассоциативной памяти, реализованной сетью Хопфилда.

Исследуемый образ	Макс. расстояние по Хеммингу	Вызываемый образ, при увеличении расстояния на 1

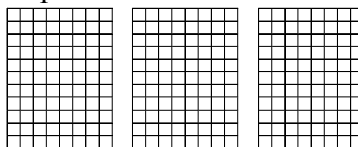
1.4. Исследование робастности сети Хопфилда.

Условие проведения эксперимента	Входной вектор (аттрактор)	Уровень шума	Аттрактор при увеличении шума
«шум» + «симметрия» + «нули»			
«шум» + «нули»			
«шум» + «симметрия»			
«шума»			

Вывод: было установлено, что максимальный уровень шума, при котором сеть верно выполняет свою функцию: \_\_\_\_%. Не соблюдение условия симметричности матрицы синаптических коэффициентов *влияет/не влияет* на робастность сети, а наличие нулей по диагоналям - *влияет/не влияет*.

2. Исследование сети Хопфилда, настроенной на три образа.

Нейронная сеть настраивается на образы:



Предположение (2): ожидается, что память данной сети будет представлена \_\_\_\_\_ позитивами, \_\_ негативами и *содержать/не содержать* ложную память.

2.1. Исследование динамики сети Хопфилда.

За один такт работы сети Хопфилда опрашивается \_\_\_\_\_ нейронов.

Начальное воздействие	Аттрактор	Кол-во тактов	Энергия	
			начало	конец
позитив «__»	позитив «__», негатив «__», ложн. память			
негатив «__»	позитив «__», негатив «__», ложн. память			
случ. вектор	позитив «__», негатив «__», ложн. память			

Вывод: значение энергии данной сети Хопфилда в процессе функционирования *увеличивается/уменьшается/остаётся неизменным* и *не достигает/достигает глобального/локального минимума/максимума* при попадании в аттрактор.

## 2.2. Исследование бассейна аттракторов сети Хопфилда.

Максимальный коэффициент корреляции между образами \_\_\_\_\_ и \_\_\_\_\_ равен \_\_\_\_\_, минимальный – между образами \_\_\_\_\_ и \_\_\_\_\_ равен \_\_\_\_\_.

Максимальное число случайных состояний, которые могут подаваться на сеть Хопфилда: \_\_\_\_\_ . В данном исследовании сеть стартовала из \_\_\_\_\_ случайных состояний.

Запоминаемый образ	Позитив, %	Негатив, %
Ложная память, суммарно, %		

Предположение (2) *подтвердилось/не подтвердилось*. Большой/небольшой объем ложной памяти связан с \_\_\_\_\_

## 2.3. Исследование ассоциативной памяти, реализованной сетью Хопфилда.

Исследуемый образ	Макс. расстояние по Хеммингу	Вызываемый образ, при увеличении расстояния на 1

При подаче на вход произвольного вектора сеть Хопфилда *исследует все «запомненные» образы/часть образов/вычисляет расстояние по Хеммингу/сравнивает потенциалы нейронов/функционирует некоторое время до любого устойчивого состояния*. В этом выражается ее свойство ассоциативной памяти. При увеличении количества «запоминаемых» образов, устойчивость сети вызывать образы по ассоциации при зашумлении «хранимого» образа *растет / уменьшается*.

## 2.4. Исследование робастности сети Хопфилда.

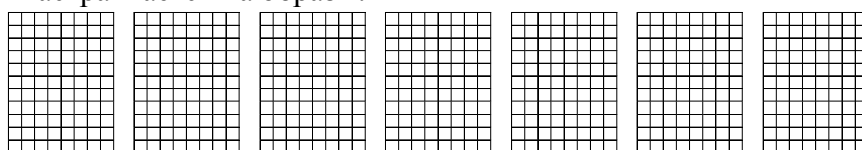
Условие проведения эксперимента	Входной вектор (аттрактор)	Уровень шума	Аттрактор при увеличении шума
«шум» + «симметрия» + «нули»			
«шум» + «нули»			
«шум» + «симметрия»			
«шума»			

Вывод: было установлено, что максимальный уровень шума, при котором сеть верно выполняет свою функцию: \_\_\_\_\_% *отличается / не отличается* от сети, настроенный на 1 образ. Не соблюдение условия симметричности матрицы синаптических коэффициентов *влияет/не влияет* на робастность сети, а наличие нулей по диагоналям - *влияет/не влияет*.

## 1. Исследование сети Хопфилда, настроенной на семь образов.

Предположение (3): ожидается, что память данной сети будет представлена \_\_\_\_\_ позитивами, \_\_\_\_\_ негативами и *содержать/не содержать* ложную память.

Нейронная сеть настраивается на образы:



## 3.1. Исследование бассейна аттракторов сети Хопфилда.

Максимальный коэффициент корреляции между образами \_\_\_\_\_ и \_\_\_\_\_ равен \_\_\_\_\_, минимальный – между образами \_\_\_\_\_ и \_\_\_\_\_ равен \_\_\_\_\_.

Максимальное число случайных состояний, которые могут подаваться на сеть Хопфилда: \_  
 \_ . В данном исследовании сеть стартовала из \_\_\_\_\_ случайных состояний.

Запоминаемый образ	Позитив, %	Негатив, %
Ложная память, суммарно, %		

Предположение (2) *подтвердилось/не подтвердилось*. Большой/небольшой объем ложной памяти связан с \_\_\_\_\_

### 3.2. Исследование робастности сети Хопфилда.

Условие проведения эксперимента	Входной вектор (аттрактор)	Уровень шума	Аттрактор при увеличении шума
«шум» + «симметрия» + «нули»			
«шум» + «нули»			
«шум» + «симметрия»			
«шума»			

Вывод: было установлено, что максимальный уровень шума, при котором сеть верно выполняет свою функцию: \_\_\_\_\_% *отличается / не отличается* от сети, настроенной на 3 образа. Не соблюдение условия симметричности матрицы синаптических коэффициентов *влияет/не влияет* на робастность сети, а наличие нулей по диагоналям - *влияет/не влияет*.

### Контрольные вопросы

1. Что называется автоассоциативной памятью? Приведите пример.
2. Что называется аттрактором динамической системы? Объясните принцип применения динамических систем с множеством аттракторов для построения ассоциативной памяти.
3. Напишите уравнения динамики сети Хопфилда.
4. Какова активационная характеристика нейронов сети Хопфилда?
5. Чему равно начальное состояние нейронов сети Хопфилда?
6. Объясните различие синхронного и асинхронного режимов функционирования рекуррентной нейронной сети. Какой из режимов функционирования используется в сети Хопфилда?
7. Каким выражением определен энергетический функционал в процессе работы сети Хопфилда?
8. Почему время достижения сетью Хопфилда одного из аттракторов из произвольного начального состояния конечно?
9. Как рассчитывается матрица синаптических коэффициентов сети Хопфилда? Какими свойствами она обладает?
10. Как приближенно оценивается объем памяти сети Хопфилда?
11. На какие типы можно разделить множество аттракторов сети Хопфилда? Что такое “ложная память”?
12. Как применяется сеть Хопфилда для кластеризации данных? Как интерпретируются кластеры?
13. Как устанавливается класс принадлежности вектора признаков при использовании нескольких (по числу классов) сетей Хопфилда, “настроенных” на разные классы?
14. Как организован в лабораторной работе процесс оценивания размеров бассейнов аттракторов сети Хопфилда?

Отчет по лабораторной работе № 9 «Самообучение нейронных сетей. Кластеризация данных»			
дата	Оценка (max 5)	Бонус за сложность	подпись

**Цель работы:**

В лабораторной работе исследуется соревновательный слой нейронов, изучаются способы его самонастройки и анализируется качество решения задачи кластеризации данных. Исследуется топографическая карта Кохонена, позволяющая построить удобную для визуального анализа многомерных данных графическую модель в плоскости. В ходе выполнения лабораторной работы студент ознакомится с демонстрационными примерами решения задач анализа данных с применением самообучающихся нейронных сетей.

**Решаемые задачи:**

Лабораторная работа выполняется с применением пакета прикладных программ “Нейронные сети” в программной среде MATLAB и включает следующие исследования:

- анализ режима самообучения соревновательного слоя нейронов и качества кластеризации двумерных данных,
- изучение на демонстрационном примере особенностей кластеризации одномерных данных с неравномерной плотностью распределения точек на прямой,
- ознакомление с демонстрационным примером построения самоорганизующейся карты Кохонена,
- построение одномерной карты Кохонена для совокупности точек в плоскости и анализ качества полученного отображения.

**Краткий конспект теоретической части** (изучите теоретическую часть файл lab9.doc)

<p>Что такое Карта Кохонена _____</p> <p>_____</p> <p>_____</p> <p>В чем смысл кластеризации данных _____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>Что такое самообучение _____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>Дайте определение одномерной карты Кохонена _____</p> <p>_____</p> <p>_____</p> <p>_____</p>
---

**Подготовка к выполнению работы:**

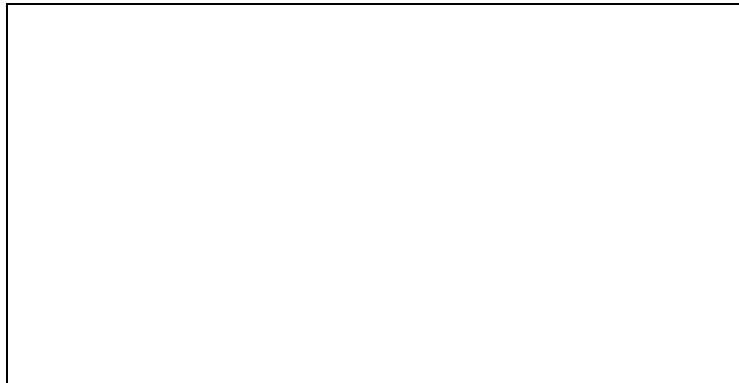
1. Создайте каталог с именем lab9.
2. Скопируйте в созданный каталог lab9 все файлы из архива kohonen.rar, находящегося в каталоге lab9 \kohonen на дистрибутивном диске.
3. Запустите систему MATLAB.
4. Добавьте полное имя созданного каталога lab9 к пути поиска системы MATLAB, выполнив из командной строки MATLAB команду addpath('имя\_каталога').

1. Кластеризация двумерных данных.

Параметры расположения двумерных данных:

Количество эллипсов рассеяния	Количество точек в кластере	Внутрикластерная дисперсия	Оптимальное число кластеров
7	10	0.1	

Для кластеризации представленных данных применяется сеть Кохонена. Входы, выходы и структура сети представлены на схеме:



Сеть содержит \_\_\_\_\_ входов, \_\_\_\_\_ выходов, \_\_\_\_\_ нейронов. Самообучение сети продолжалось в течение \_\_\_\_\_ эпох. Начальное значение параметра скорости самообучения  $\lambda_0 =$  \_\_\_\_\_.  
На графике зависимости \_\_\_\_\_ представлено визуальное расположение двумерных данных и центров кластеров после завершения процесса самообучения.



Указать на графике номера кластеров

Координаты центров кластеров представлены в таблице:

№ кластера	1	2	3	4	5	6	7
<i>x</i>							
<i>y</i>							

Близким по номеру кластерам *соответствуют / не соответствуют* близкие кластеры в пространстве признаков.

2. Негативные эффекты процедуры самообучения сети Кохонена.

Обучающая выборка имеет параметры:

Количество эллипсов рассеяния	Количество точек в кластере	Внутрикластерная дисперсия	Оптимальное число кластеров
5	10	0.2	

Визуальное расположение данных и центров кластеров после завершения процесса самообучения представлено на графике *a*.

**Укрупнение** кластеров может наблюдаться при *небольшом / слишком большом / произвольном* количестве нейронов. Условие возникновения укрупнённых кластеров *зависит / не зависит* от скорости самообучения и *зависит / не зависит* от начального расположения нейронов.

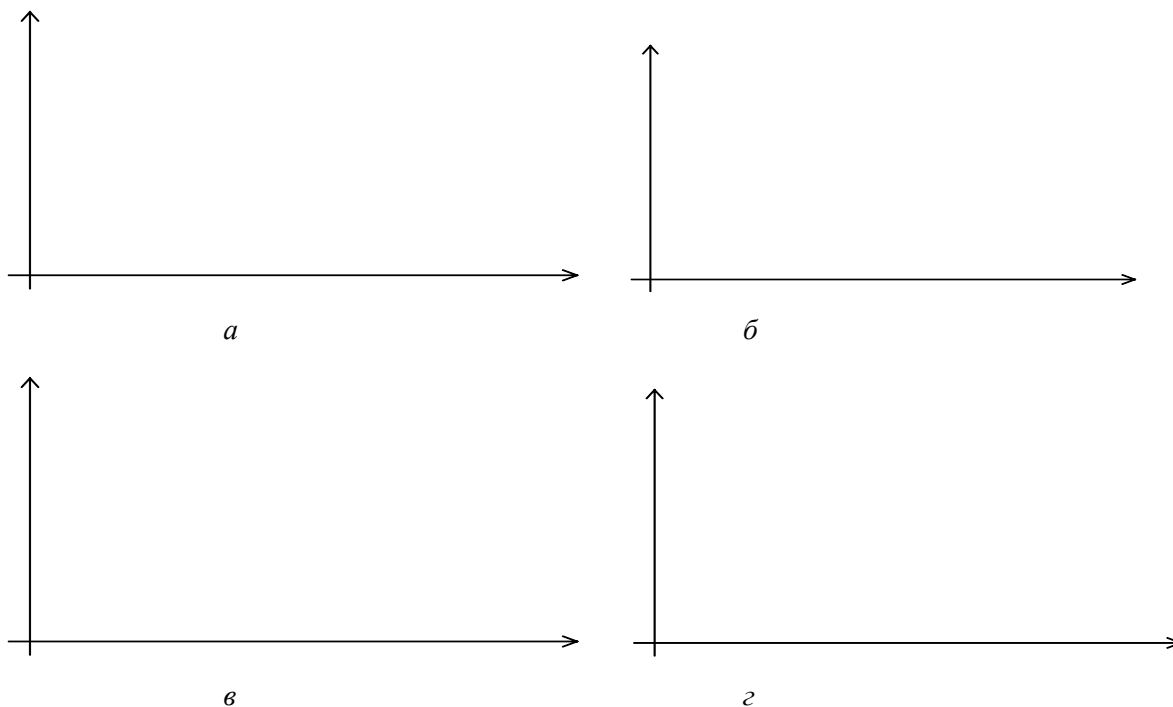
**Детализация** кластеров может наблюдаться при *небольшом /слишком большом /произвольном* количестве нейронов. Условие возникновения детализации *зависит /не зависит* от скорости самообучения и *зависит /не зависит* от начального расположения нейронов.

Возникновение **мёртвых нейронов** (пустых кластеров) может наблюдаться при *небольшом /слишком большом /произвольном* количестве нейронов. Условие возникновения пустых кластеров *зависит /не зависит* от скорости самообучения и *зависит /не зависит* от начального расположения нейронов.

Наблюдаемая ситуация	Число кластеров	Начальное значение параметра скорости самообучения $\lambda_0$	Число эпох самообучения
Укрупнение кластеров			
Детализация кластеров			
Мёртвые нейроны			

После завершения процесса самообучения визуальное расположение исходных данных и центров укрупнённых кластеров представлено на рисунке *б*, центров детализированных кластеров – на рисунке *в*, центров пустых кластеров – на рисунке *г*.

**Колебательный характер** процесса самообучения может наблюдаться при *небольшом /слишком большом /произвольном* количестве нейронов. Условие возникновения колебательного процесса самообучения *зависит /не зависит* от скорости самообучения и *зависит /не зависит* от начального расположения нейронов.



### 3. Проверка качества самообучения сети Кохонена.

Координаты примеров контрольной выборки и результаты их обработки сетью Кохонена:

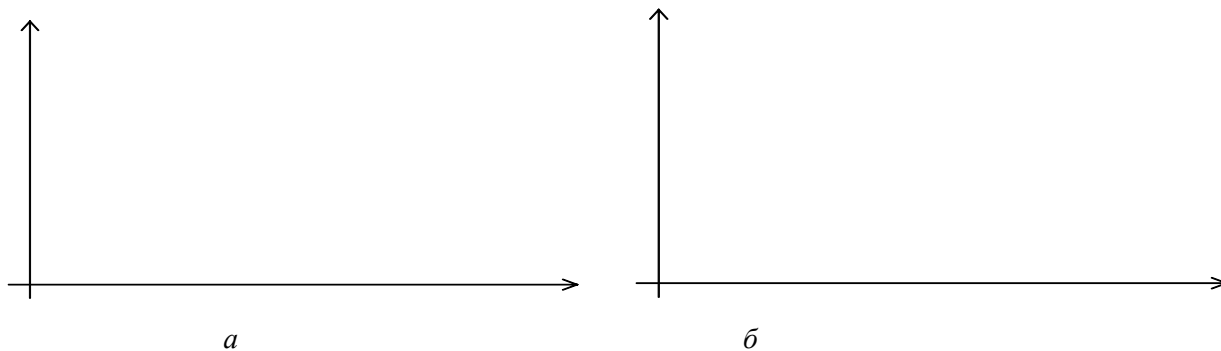
№ примера	1	2	3	4	5	6	7
<i>x</i>							
<i>y</i>							
<i>N</i> кластера							

**Нанести контрольные точки на график.** Результаты обработки контрольных примеров сетью Кохонена *согласуются /не согласуются* с визуальным анализом расположения кластеров и контрольных точек.



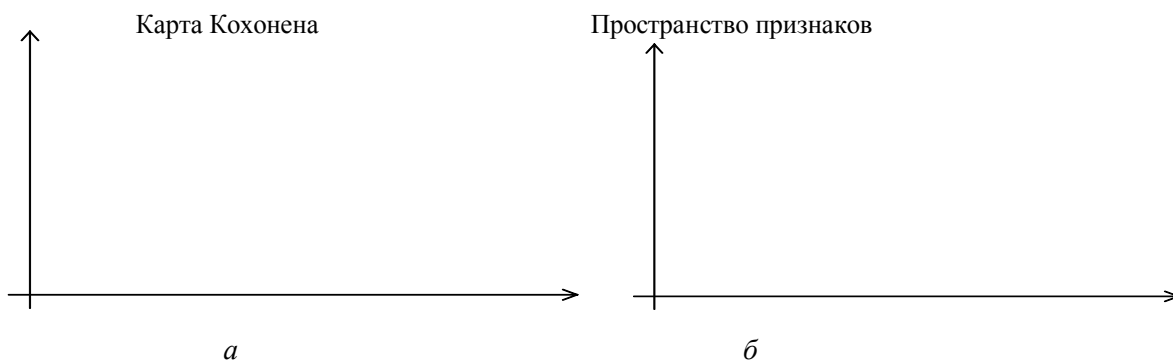
4. Кластеризация одномерных данных.

Сеть Кохонена содержит \_\_\_\_ входов, \_\_\_\_ выходов, \_\_\_\_ нейронов. Самообучение сети продолжалось в течение \_\_\_\_ эпох. Функция плотности распределения точек на прямой приведена на графике *a*. График зависимости синаптических коэффициентов нейронов слоя Кохонена от номера эпохи обучения приведён на рис. *б*.



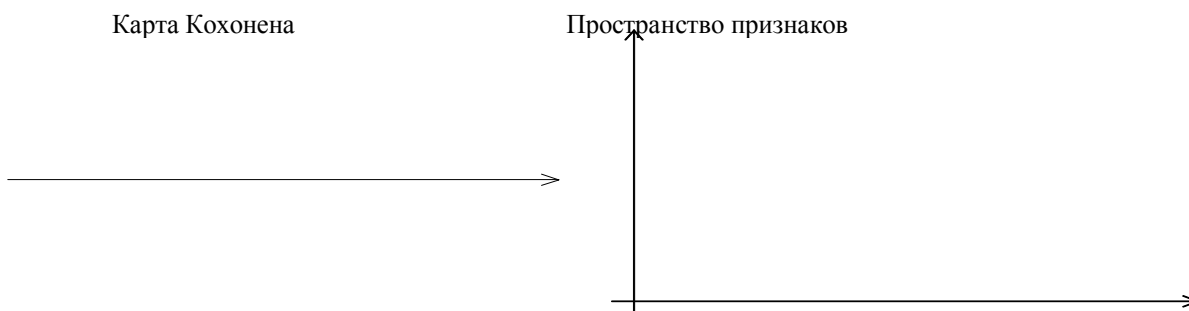
5. Самообучение двумерной карты Кохонена на двумерных данных.

Сеть Кохонена содержит \_\_\_\_ входов, \_\_\_\_ выходов, \_\_\_\_ нейронов. Расположение нейронов на карте Кохонена показано на графике *a*. Расположение данных, центров кластеров и отображение карты Кохонена в пространство признаков после завершения процесса самообучения – на графике *б*.



6. Самообучение одномерной карты Кохонена на двумерных данных.

Для кластеризации представленных данных выбрано число кластеров  $k = \underline{\hspace{2cm}}$ . Сеть Кохонена содержит \_\_\_\_ входов, \_\_\_\_ выходов, \_\_\_\_ нейронов. Самообучение карты продолжалось в течение \_\_\_\_ эпох,  $\lambda_0 = \underline{\hspace{2cm}}$ .



Близким по номеру кластерам (по метрике на карте) *соответствуют / не соответствуют* близкие кластеры в пространстве признаков.

## Контрольные вопросы

1. В чем состоит проблема кластеризации данных? Какие выборочные данные необходимы для решения этой задачи?
2. Где в соревновательном слое нейронов хранится информация о прототипах кластеров?
3. Дайте описание работы нейронов соревновательного слоя.
4. В чем состоит избирательность нейронов соревновательного слоя по отношению к данным обучающей выборки?
5. Как оценивается расстояние между выборочным примером и вектором синаптических коэффициентов нейрона?
6. Почему именно потенциалы нейронов соревновательного слоя используются для выявления нейрона-победителя?
7. Объясните правило самообучения нейронов соревновательного слоя.
8. Что называется “эпохой” самообучения соревновательного слоя нейронов?
9. Как влияет выбор параметра скорости самообучения на качество процесса настройки синаптических коэффициентов? Какие практические рекомендации можно дать для выбора этого параметра?
10. Что такое “соревновательная” активационная характеристика?
11. Какие негативные особенности можно ожидать при реализации самообучения соревновательного слоя нейронов?
12. Каким способом можно избежать появления “мертвых” нейронов соревновательного слоя?
13. По каким причинам может возникать укрупнение кластеров при обучении соревновательного слоя нейронов?
14. Как практически можно оценить качество обучения соревновательного слоя нейронов? Каким свойством обладает оптимально обученный соревновательный слой?
15. Какое основное качество отличает карту Кохонена от соревновательного слоя нейронов?
16. Приведите примеры оценки расстояния между нейронами в топографической карте. какими в этих примерах являются линии точек, равноудаленных от фиксированного нейрона?
17. В чем состоит правило обучения нейронов карты Кохонена? Каково его отличие от правила обучения нейронов соревновательного слоя?
18. В чем состоит “раскраска” топографической карты отдельными признаками?
19. Что представляет собой одномерная карта Кохонена?
20. Какой Вы можете привести пример практического применения карты Кохонена?

Для заметок

Для заметок