

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Минцаев Магомед Шакалович

Должность: Ректор

Дата подписания: 25.12.2014 14:38:02

Уникальный программный ключ:

236bcc35c296f119d6aafdc22836b21db52dbc07971a86865a5825f9fa4304cc

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Уфимский государственный нефтяной технический университет»
Филиал ФГБОУ ВО УГНТУ в г. Стерлитамаке

Кафедра автоматизированных технологических и информационных систем

МОДЕЛИ И АЛГОРИТМЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ ПРИ ОБРАБОТКЕ ИНФОРМАЦИИ И УПРАВЛЕНИИ

Учебно-методическое пособие
к практическим работам
по дисциплине «Интеллектуальные системы управления и обеспечения
безопасности»

Уфа
2016

Учебно-методическое пособие предназначено для выполнения практических работ обучающимися направления 27.04.04 Управление в технических системах, изучающих дисциплину «Интеллектуальные системы управления и обеспечения безопасности».

Пособие формирует навыки самостоятельной работы в решении научных, прикладных и профессиональных задач по профилю будущей деятельности студента.

Составитель: Кадыров Р.Р., канд. техн. наук, доц. каф. АТИС

Рецензенты: Рахман П.А., канд. техн. наук, доц. каф. АТИС
Чариков П.Н., канд. техн. наук, доц. каф. АТИС

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Интеллектуальные системы	6
2 Практическая работа №1. Изучение программных пакетов современных интеллектуальных систем для обеспечения безопасности и управления технологическими процессами и дифференциально-модельная концепция в систематике базы макрофизических знаний	9
2.1 Теоретическая часть	9
2.1.1 Принятие решения на основе матричных игр в чистых стратегиях	9
2.1.2 Решение матричной игры в смешанных стратегиях.....	12
2.2 Описание базовых возможностей среды SciLab	13
2.2.1 Начало работы с пакетом.....	13
2.2.2 Редактирование и отладка файлов-сценариев.....	14
2.2.3 Массивы и матрицы в Scilab	15
2.2.4 Ввод и формирование массивов и матриц.....	16
2.2.5 Действия над матрицами	17
2.2.6 Специальные матричные функции	17
2.2.7 Принятие решений в смешанных стратегиях	18
2.2.8 Методика выполнения контрольной работы	19
2.2.9 Задание.....	19
2.2.10 Контрольные вопросы	20
3 Применение искусственных нейронных сетей в интеллектуальных системах высокой точности и надежности	21
4 Практическая работа №2. Построение и обучение нейросети	22
4.1 Теоретическое введение.....	22
4.2 Общие сведения об нейроэмуляторе NeuroPro	24
4.3 Описание меню.....	25
4.4 Нейросетевая парадигма	27
4.5 Подача и снятие сигналов.....	27
4.6 Точность решения задачи	28
4.7 Обучение нейронной сети	29
4.8 Упрощение нейронной сети.....	30
4.9 Вербализация нейронной сети	31

4.10	Этапы решения задач с использованием нейронных сетей.....	32
4.11	Практическая часть.....	34
4.12	Варианты заданий к контрольной работе.....	42
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	44

ВВЕДЕНИЕ

В данном учебном пособии приведены научные и методические материалы, посвященные проблемам исследования и создания интеллектуальных систем. Под интеллектуальной системой понимается объединенная информационным процессом совокупность технических средств и программного обеспечения, работающая во взаимосвязи с человеком (коллективом людей) или автономно, способная на основе сведений и знаний при наличии мотивации синтезировать цель, принимать решение к действию и находить рациональные способы достижения цели.

Рассмотрены также искусственные нейронные сети и возможности использования современных методов анализа в интеллектуальных системах.

Пособие может быть полезным также для аспирантов и специалистов в области системных исследований в задачах управления.

1 Интеллектуальные системы

Развитие информационных технологий в XXI веке будет сопряжено с разработкой и созданием интеллектуальных систем обработки информации и управления в различных средах обитания и деятельности человека. Сегодня вычислительные средства значительно превзошли человека в таких областях, как вычисления, обработка текстов, а в последнее время — даже в области логического вывода. Тем не менее, им еще не хватает гибкости, и они отстают от человека во многих аспектах, например, в распознавании образов, решении задач при неполной информации, в способности к обучению, прогнозе результатов предполагаемого действия и выработке управления, с учетом динамики протекания процессов в реальном времени. Такая работа с информацией, свойственная человеку, характеризуется понятием "гибкой" обработки информации, в отличие от традиционной "жесткой" обработки информации и выработки управления вычислительной системой, которая предполагает наличие полностью заданной информации в априори оговоренном мире или проблемной области. Этот подход к обработке информации, который можно назвать ассоциативным или интуитивным в противовес логическому, еще совсем не развит в существующей ныне информационной технологии. Здесь уместно отметить, что развитие информационных технологий происходило во взаимосвязи с эволюцией вычислительных систем. Если такие системы первых поколений позволяли осуществлять цифровую обработку данных и текстов, создавать и использовать базы данных, то вычислительные системы пятого поколения уже дают возможность обрабатывать знания, осуществлять логический вывод и тем самым создают начала их интеллектуализации. Такие вычислительные системы представляли собой некоторые самостоятельные образования — инструмент, не входящий органически в "состав" естественных и общественных процессов, а лишь предназначались для выполнения некоторых весьма важных вычислительных операций, отображающих эти процессы. Взаимодействие же с человеком или коллектива людей с вычислительной системой состояло в необходимости разработки программы

вычислений, ее отладки и представлении результатов в удобной для понимания человеком форме и т.п. Однако ясно, что получение информации для обработки ее в вычислительных системах сопряжено с проведением различного рода измерений тех или иных характеристик окружающей среды, а результаты обработки данных должны использоваться для принятия решения о том или ином действии, в соответствии с управлением, выработанным вычислительной системой, с последующим контролем результатов управления.

Каждая система реального мира оперирует огромным количеством входной информации. Например, в качестве входных значений могут быть значения показателей датчиков окружающей среды: температура, давление, скорость ветра и т.д. При этом значения с вышеописанных датчиков постоянно меняются. И здесь сразу всплывает первая, и одна из основных проблем данных систем – обеспечение обработки постоянно меняющихся данных и способность устойчивой выработки такого управления, которое удовлетворяет мотивации системы. Другими словами, необходимо вырабатывать управление, учитывая постоянное изменение входных данных, и делать это за малый промежуток времени – приспосабливаться к окружающей среде, сродни тому, как это делает человек. Опишем работу узлов схемы ИС, представленной на рисунке 2.1. На основании активного оценивания лавинообразного потока информации от датчиков, представляющих тематическую информацию об окружающей среде, а также на основе знаний (выполненных на базе АП – ассоциативной памяти, которая, в свою очередь, представляет собой НС, описанную выше) и мотивации блок ЦИС (центральная интеллектуальная система) принимает решение к 33 действию. Изменчивость окружающей среды и собственного состояния системы может приводить к потребности в чем-либо (мотивации), а при наличии знания может синтезироваться цель (блок СЦ). Под целью понимается идеальное, мысленное предвосхищение результата деятельности. Продолжая активно оценивать информацию об окружающей среде и собственном состоянии системы, в том числе объекта управления, при сопоставлении вариантов достижения цели можно принять решение к действию. Отметим также блок вывода новых знаний, также реализующийся посредством НС. Параллельно с остальной

работой системы данный блок анализирует знания системы (АП) посредством блока ЦИС и, если это возможно – выводит новые знания на основании имеющихся, пополняя тем самым общее знание системы.

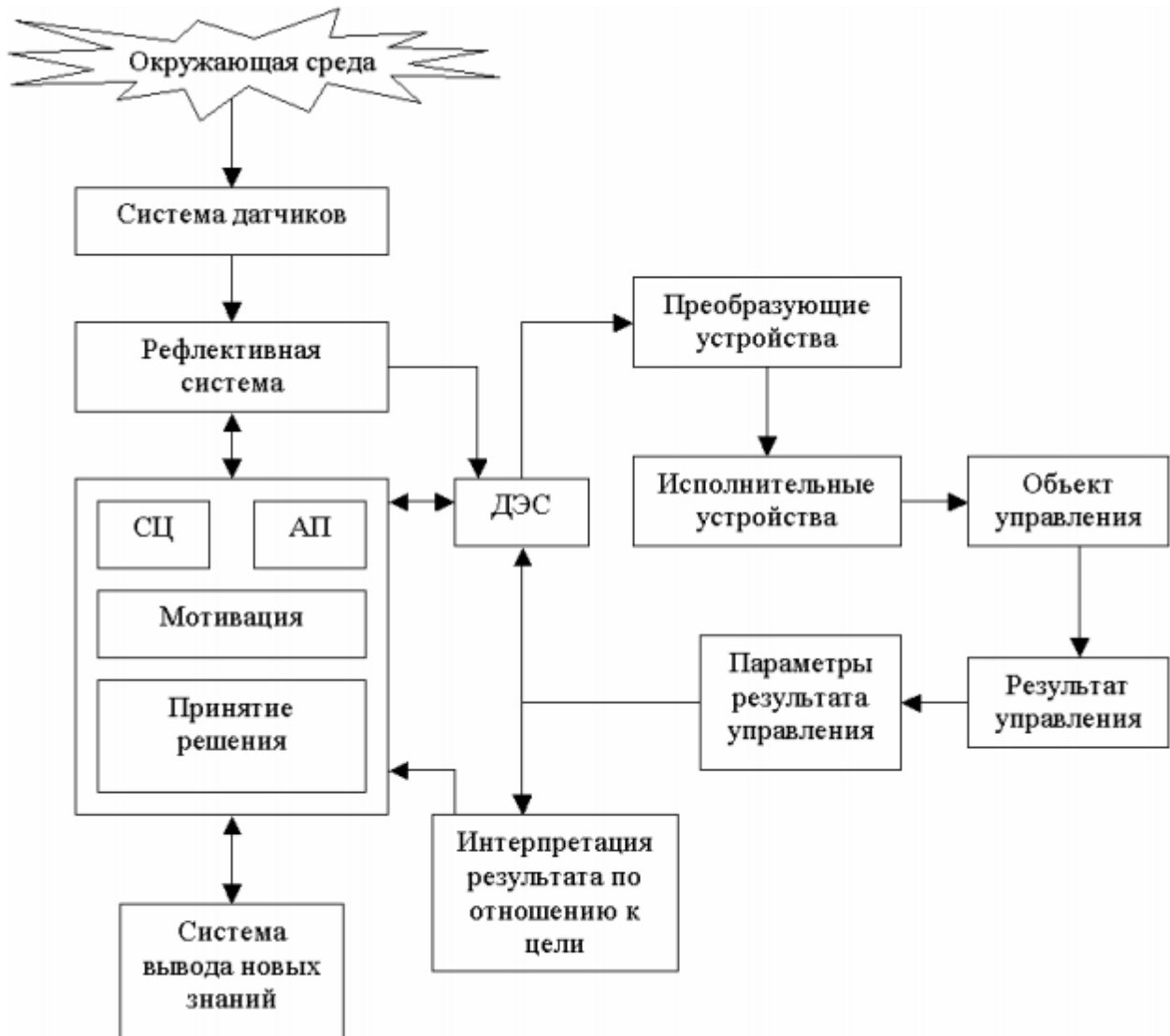


Рисунок 1.1 – Схема интеллектуальной системы

2 Практическая работа №1. Изучение программных пакетов современных интеллектуальных систем для обеспечения безопасности и управления технологическими процессами и дифференциально-модельная концепция в систематике базы макрофизических знаний

Цель работы:

Целью работы является изучение инструментального средства Scilab для решения задач поддержки принятия решения.

Задачи:

Задачами практической работы являются овладение навыками принятия решения на основе матричных игр и изучения пакета Scilab.

2.1 Теоретическая часть

Scilab – это система компьютерной математики, которая предназначена выполнения инженерных и научных вычислений, включающих в себя задачи принятия решения. Пакет Scilab содержит набор инструментов и интерактивную документацию. Он является некоммерческим аналогом пакета Matlab. Scilab состоит из трех частей:

- интерпретатор;
- библиотека функций;
- библиотека Fortran и C процедур.

2.1.1 Принятие решения на основе матричных игр в чистых стратегиях

Рассмотрим простейшую математическую модель конечной конфликтной ситуации, в которой имеется два участника и выигрыш одного равен проигрышу другого. Такая модель называется антагонистической игрой двух лиц с нулевой суммой. Игра состоит из двух ходов: игрок А выбирает одну из возможных

стратегий $a_i, i = \overline{1, m}$, а игрок В выбирает одну из возможных стратегий $b_j, j = \overline{1, n}$. Каждый выбор производится при полном незнании выбора соперника. В результате выигрыш игроков составит соответственно a_{ij} и $-a_{ij}$. Цель игрока А - максимизировать величину a_{ij} , а игрока В - минимизировать эту величину. Критерием принятия решения является функция, выражающая предпочтение лица принимающего решение и определяющая правило, по которому выбирается приемлемый или оптимальный вариант решения.

Определение 1. Матрица, составленная из величин $a_{ij}, i = \overline{1, m}, j = \overline{1, n}$,

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}_{m \times n}$$

называется платежной матрицей, или матрицей игры. Каждый элемент платежной матрицы $a_{ij}, i = \overline{1, m}, j = \overline{1, n}$, равен выигрышу А (проигрышу В), если он выбрал стратегию $A_i, i = \overline{1, m}$, а игрок В выбрал стратегию $B_j, j = \overline{1, n}$.

Задача каждого из игроков – найти наилучшую стратегию игры, при этом предполагается, что противники одинаково разумны и каждый из них делает все, чтобы получить наибольший доход.

Найдем наилучшую стратегию первого игрока. Если игрок А выбрал стратегию $A_i, i = \overline{1, m}$, то в худшем случае (например, если его ход известен В) он получит выигрыш $a_i = \min_j a_{ij}$. Предвидя такую возможность, игрок А должен выбрать такую стратегию, чтобы максимизировать свой минимальный выигрыш.

$$\alpha = \max_i a_i = \max_i (\min_j a_{ij}). \quad (2.1)$$

Определение 2. Величина α - гарантированный выигрыш игрока А называется нижней ценой игры. Стратегия $A_{\text{юпт}}$, обеспечивающая получение выигрыша α , называется максиминной.

Если первый игрок будет придерживаться своей максиминной стратегии, то у него есть гарантия, что он в любом случае выиграет не меньше α .

Аналогично определяется наилучшая стратегия второго игрока. Игрок В при выборе стратегии $V_j, j = \overline{1, n}$ в худшем случае получит проигрыш $\beta_j = \min_i a_{ij}$. Он выбирает стратегию $V_{j_{opt}}$, при которой его проигрыш будет минимальным и составит

$$\beta = \max_j \beta_j = \min_j (\max_i a_{ij}). \quad (2.2)$$

Определение 3. Величина β – гарантированный проигрыш игрока В называется верхней ценой игры. Стратегия $V_{j_{opt}}$, обеспечивающая получение проигрыша β , называется минимаксной.

Если второй игрок будет придерживаться своей минимаксной стратегии, то у него есть гарантия, что он в любом случае проиграет не больше β .

Фактический выигрыш игрока А (проигрыш игрока В) при разумных действиях партнеров ограничен верхней и нижней ценой игры. Для матричной игры справедливо неравенство $\alpha \leq \beta$.

Определение 4. Если $\alpha = \beta = v$, т. е.

$$\max_i (\min_j a_{ij}) = \min_j (\max_i a_{ij}) = v, \quad (2.3)$$

то выигрыш игрока А (проигрыш игрока В) определяется числом v . Оно называется ценой игры.

Определение 5. Если $\alpha = \beta = v$, то такая игра называется игрой с седловой точкой, элемент матрицы $a_{i_{opt} j_{opt}} = v$, соответствующий паре оптимальных стратегий $(A_{i_{opt}}, V_{j_{opt}})$, называется седловой точкой матрицы. Этот элемент является ценой игры.

Седловой точке соответствуют оптимальные стратегии игроков. Их совокупность – решение игры, которое обладает свойством: если один из игроков придерживается оптимальной стратегии, то второму отклонение от своей оптимальной стратегии не может быть выгодным.

Определение 6. Если игра имеет седловую точку, то говорят, что она решается в чистых стратегиях.

2.1.2 Решение матричной игры в смешанных стратегиях

Если платежная матрица не имеет седловой точки, т.е. $\alpha < \beta$ и $\alpha \leq v \leq \beta$, то поиск решения игры приводит к применению сложной стратегии, состоящей в случайном применении двух и более стратегий с определенными частотами.

Определение 1. Сложная стратегия, состоящая в случайном применении всех стратегий с определенными частотами, называется смешанной.

Предположим, что цена игры $v > 0$. Если это не так, то прибавим ко всем элементам матрицы A такое число C , чтобы $v > 0$. Пусть имеется матрица A , и известны оптимальные смешанные стратегии 2-х игроков: $x = (x_1, \dots, x_m)$ – 1-го игрока, $y = (y_1, \dots, y_n)$ – 2-го игрока. Цена игры удовлетворяет соотношениям:

$$\left\{ \begin{array}{l} \sum_{i=1}^m a_{ij} * x_i \geq v, j = \overline{1, n} \\ \sum_{i=1}^m x_i = 1 \\ x_i \geq 0 \end{array} \right. , \left\{ \begin{array}{l} \sum_{j=1}^n a_{ij} * y_j \leq v, i = \overline{1, m} \\ \sum_{j=1}^n y_j = 1 \\ y_j \geq 0 \end{array} \right. \quad (2.4)$$

Разделим все неравенства и уравнения на v и обозначим:

$$p_i = \frac{x_i}{v}, q_j = \frac{y_j}{v} \quad (2.5)$$

Получим:

$$\sum_{i=1}^m a_{ij} * p_i \geq 1, \sum_{i=1}^m p_i = \frac{1}{v}, p_i \geq 0 \quad (2.6)$$

$$\sum_{j=1}^n a_{ij} * q_j \leq 1, \sum_{j=1}^n q_j = \frac{1}{v}, q_j \geq 0 \quad (2.7)$$

Т.к. 1-й игрок стремится найти значения x_i и p_i , чтобы $v \rightarrow \max$, то необходимо найти p_i , при которых

$$\sum_{i=1}^m p_i \rightarrow \min, \sum_{i=1}^m a_{ij} * p_i \geq 1. \quad (2.8)$$

Аналогично

$$\sum_{j=1}^n q_j \rightarrow \max, \sum_{j=1}^n a_{ij} * q_j \leq 1. \quad (2.9)$$

Решив, найдем p_i , q_j .

Смешанной стратегией игроков является решение

$$x_i = v p_i, y_j = v q_j. \quad (2.10)$$

2.2 Описание базовых возможностей среды SciLab

2.2.1 Начало работы с пакетом

Программа SCILAB представляет собой пакет прикладных математических программ, представляющий мощное открытое окружение для инженерных и технических отчетов.

После запуска Scilab на экране появится основное окно приложения. Окно содержит меню, панель инструментов и рабочую область. Признаком того, что система готова к выполнению команды, является наличие знака приглашения $-->$, после которого расположен мигающий курсор. Рабочую область называют командной строкой. Ввод команд осуществляется с клавиатуры. Нажатие клавиши Enter заставляет систему выполнить команду и вывести результат (рисунок 2.1).

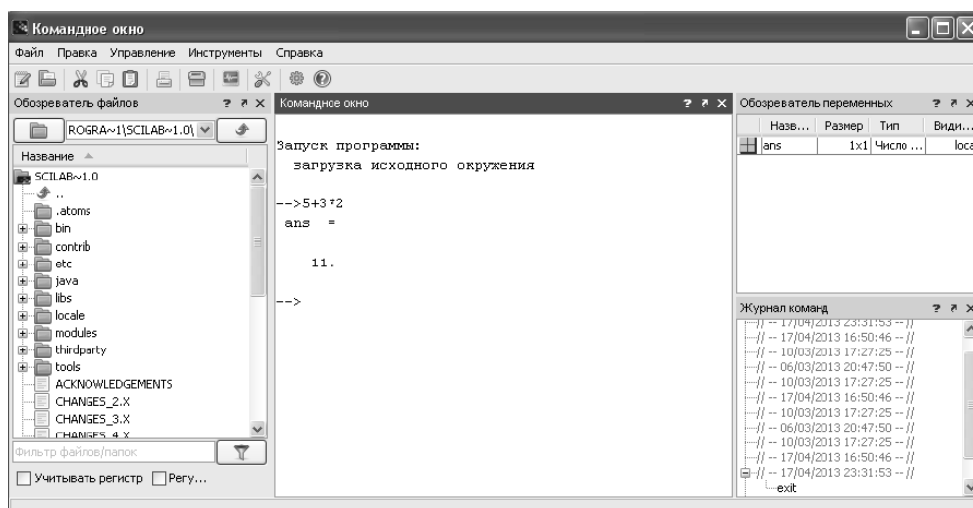


Рисунок 2.1 – Выполнение команд в Scilab

2.2.2 Редактирование и отладка файлов-сценариев

Файл-сценарий - это список команд Scilab, сохраненный на диске. Для подготовки, редактирования и отладки файлов- сценариев служит специальный редактор SciNotes, который можно вызвать из меню Инструменты. Окно редактора файлов-сценариев выглядит стандартно, т.е. имеет заголовок, меню, панели инструментов, строку состояния.

Ввод текста в окне редактора файла-сценария осуществляется по правилам, принятым для команд Scilab. Рис. 2 содержит пример ввода команд для решения квадратного уравнения $x^2+2x+1=0$. Точка с запятой «;» ставится после тех команд, которые не требуют вывода значений.

Для сохранения введенной информации необходимо выполнить команду Сохранить из меню Файл. Ввод имени в поле File Name и щелчок по кнопке Save приведет к сохранению информации, находящейся в окне редактора. Файлы-сценарии сохраняют с расширением .sce.

Выполнить операторы файла сценария можно двумя способами:

- 1 из меню редактора SciNotes вызвать команду Выполнить;
- 2 из главного меню Scilab вызвать команду Выполнить и указать имя файла-сценария.

Все эти действия приведут к появлению в рабочей области результатов вычислений команд файла сценария (рисунок 2.2).

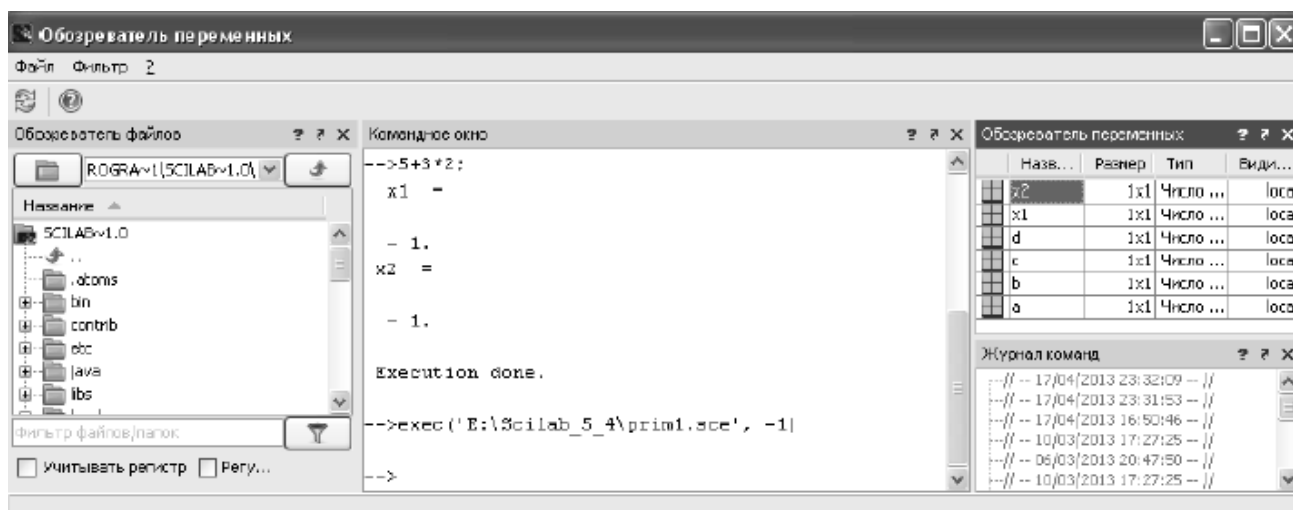


Рисунок 2.2 – Выполнение файла-сценария Scilab

2.2.3 Массивы и матрицы в Scilab

Для работы с множеством данных удобно использовать массивы. Массив - это множественный тип данных, состоящий из фиксированного числа элементов. Массиву должно быть присвоено имя.

Переменную, представляющую собой просто список данных, называют одномерным массивом или вектором. Для доступа к данным, хранящимся в определенном элементе массива, необходимо указать имя массива и порядковый номер этого элемента, называемый индексом.

Если возникает необходимость хранения данных в виде таблиц, то необходимо использовать матрицы или двумерные массивы. Для доступа к данным, хранящимся в матрице, необходимо указать имя матрицы и два индекса: первый должен соответствовать номеру строки, а второй - номеру столбца, в которых хранится необходимый элемент.

2.2.4 Ввод и формирование массивов и матриц

Задать массивы и матрицы в Scilab можно с помощью их поэлементного ввода.

Для определения вектора-строки следует ввести имя массива, а затем после знака присваивания, в квадратных скобках через пробел или запятую перечислить элементы массива: $\text{name}=[x1\ x2\ \dots\ xn]$ или $\text{name}=[x1, x2, \dots, xn]$.

Элементы вектора-столбца вводятся через точку с запятой: $\text{name}=[x1; x2; \dots; xn]$.

Ввод элементов матрицы осуществляется в квадратных скобках, при этом элементы строки отделяются друг от друга пробелом или запятой, а строки разделяются между собой точкой с запятой: $\text{name}=[x11, x12, \dots, x1n; x21, x22, \dots, x2n; \dots; xm1, xm2, \dots, xmn]$.

Обратиться к элементу матрицы можно, указав после имени матрицы, в круглых скобках через запятую, номер строки и номер столбца на пересечении которых расположен элемент: $\text{name}(\text{индекс1}, \text{индекс2})$. На рисунке 2.3 приведен пример задания матрицы и обращения к ее элементам.

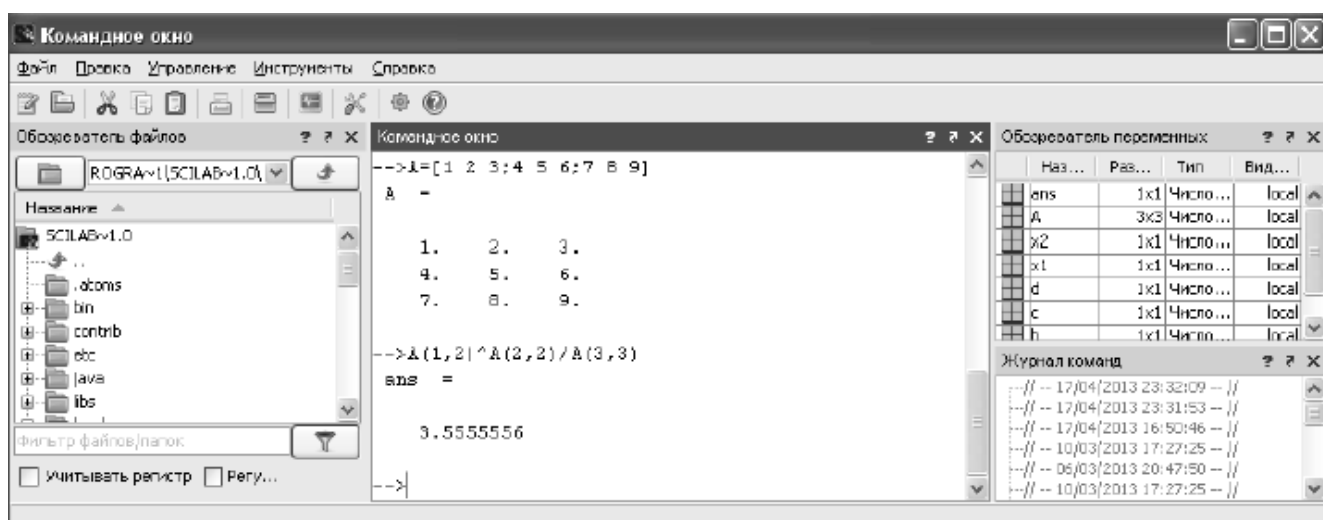


Рисунок 2.3 – Пример обращения к элементам матрицы

2.2.5 Действия над матрицами

Для работы с массивами и матрицами в Scilab предусмотрены следующие операции:

- «+» - сложение;
- «-» - вычитание;
- «'» - транспонирование;
- «*» - умножение на число или матричное умножение;
- «^» - возведение в степень;
- «.*» - поэлементное умножение матриц;
- «.^» - поэлементное возведение в степень.

На рисунке 2.4 показан пример действий над матрицами.

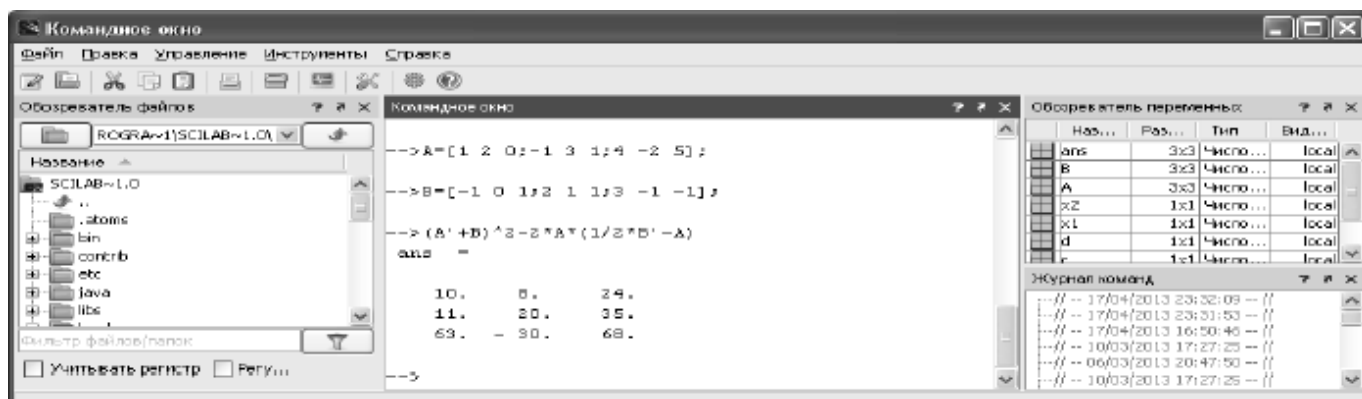


Рисунок 2.4 – Пример матричных операций

2.2.6 Специальные матричные функции

Для работы с массивами и матрицами в Scilab существуют специальные функции. Рассмотрим функции определения матриц:

- ones(m,n) - создает матрицу единиц из m строк и n столбцов;
- zeros(m,n) - создает нулевую матрицу из m строк и n столбцов;
- eye(m,n) - формирует единичную матрицу из m строк и n столбцов;

- $\text{sort}(X)$ - выполняет упорядочивание массива X , если X - матрица, сортировка выполняется по столбцам;

- $\text{max}(M[,f1])$ - вычисляет наибольший элемент в массиве M , имеет необязательный параметр $f1$. Если параметр $f1$ отсутствует, то функция $\text{max}(M)$ возвращает максимальный элемент массива M ; если $f1='r'$, то функция вернет строку максимальных элементов столбцов матрицы M ; если $f1='c'$, то результатом работы функции будет вектор-столбец, каждый элемент которого равен максимальному элементу соответствующих строк матрицы M . Функция $[x, \text{nom}] = \text{max}(M[,f1])$ вернет значение максимального элемента x и его номер в массиве nom ;

- $\text{min}(M[,f1])$ - вычисляет наименьший элемент в массиве M , работает аналогично функции max ;

- $\text{det}(M)$ - вычисляет определитель квадратной матрицы M ;

- $\text{linsolve}(A,b)$ - решает систему линейных алгебраических уравнений вида $A*x-b=0$.

2.2.7 Принятие решений в смешанных стратегиях

Для принятия решений в смешанных стратегиях используем методы оптимизации. Для решения задач оптимизации в Scilab предназначена функция linpro следующей структуры:

$[x,kl,f] = \text{linpro}(c,A,b[,ci,cs][,k][,x0])$.

Здесь c - массив (вектор-столбец) коэффициентов при неизвестных функции цели, длина вектора n совпадает с количеством неизвестных x .

A - матрица при неизвестных из левой части системы ограничений, количество строк матрицы равно количеству ограничений m , а количество столбцов совпадает с количеством неизвестных n .

b - массив (вектор-столбец), содержит свободные члены системы ограничений, длина вектора m .

c_i - массив (вектор-столбец) размерности n содержит нижнюю границу переменных ($c_{ij} \leq x_j$); если такой нет, указывают [].

cs - массив (вектор-столбец) размерности n содержит нижнюю границу переменных ($cs_j \leq x_j$); если такой нет, указывают [].

k - целочисленная переменная, используется, если в систему ограничений кроме неравенств входят и равенства, в матрице они будут находиться в k первых строках, оставшиеся l строк займут неравенства, т.е. $m=k+l$.

x_0 - вектор-столбец начальных приближений длиной n .

Функция `linpro` возвращает массив неизвестных x , минимальное значение функции f и массив множителей Лагранжа kl .

2.2.8 Методика выполнения контрольной работы

- 1 Запустить Scilab.
- 2 Задать матрицу A .
- 3 Найти седловую точку.
- 4 Определить цену игры.
- 5 Найти оптимальные стратегии двух игроков.

2.2.9 Задание

Задана матрица A . Необходимо найти оптимальные стратегии двух игроков.

Вариант 1. $A = \begin{pmatrix} 0 & 1 & -1 \\ 0 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix}$.

Вариант 2. $A = \begin{pmatrix} 1 & -2 & 1 \\ 2 & 1 & 2 \\ 1 & 3 & -1 \end{pmatrix}$.

Вариант 3. $A = \begin{pmatrix} 1 & -1 & -2 \\ -2 & 1 & 1 \\ 0 & 2 & 1 \end{pmatrix}$.

Вариант 4. $A = \begin{pmatrix} 3 & -2 & 1 \\ -2 & 1 & -1 \\ 1 & 1 & -1 \end{pmatrix}$.

Вариант 5. $A = \begin{pmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 1 & 1 & -1 \end{pmatrix}$.

Вариант 6. $A = \begin{pmatrix} -1 & 1 & -1 \\ 0 & 1 & 1 \\ 1 & 0 & -1 \end{pmatrix}$.

Вариант 7. $A = \begin{pmatrix} 3 & 1 & 1 \\ 0 & -1 & -3 \\ 1 & 0 & 2 \end{pmatrix}$.

Вариант 8. $A = \begin{pmatrix} 0 & 1 & 1 \\ 2 & -1 & -2 \\ 1 & 0 & -1 \end{pmatrix}$.

Вариант 9. $A = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 1 & 2 \\ 1 & -1 & -1 \end{pmatrix}$.

Вариант 10. $A = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 1 & -2 \\ 1 & 0 & -1 \end{pmatrix}$.

2.2.10 Контрольные вопросы

- 1 Что такое платежная матрица?
- 2 Что такое цена игры?
- 3 Что такое седловая точка?
- 4 В чем заключается смешанная стратегия?
- 5 Назовите основные функции по работе с матрицами в Scilab.
- 6 Как задаются матрицы в Scilab?

3 Применение искусственных нейронных сетей в интеллектуальных системах высокой точности и надежности

Искусственные нейронные сети (ИНС) являются эффективным средством решения сложных, плохо формализуемых задач. К этому классу традиционно относятся задачи классификации, кластеризации, аппроксимации многомерных отображений, прогнозирования временных рядов, нелинейной фильтрации, идентификации, а также управления сложными технологическими объектами. В настоящее время методы нейротехнологии активно применяются для обработки аэрокосмических изображений и гидроакустических сигналов, идентификации и управления нелинейными динамическими объектами в реальном времени.

Наибольшее распространение в технических приложениях получили многослойные нейронные сети прямого действия, что обусловлено простотой их алгоритмической и программно-аппаратной реализации, наличием развитых методов обучения, возможностью параллельного выполнения вычислений.

Теория управления линейными системами, которые описываются линейными дифференциальными уравнениями, была достаточно хорошо разработана в 60-е годы с применением аппарата преобразований Лапласа и частотно-фазовых характеристик. Тогда же предпринимались первые попытки применения цифровых вычислительных машин в качестве корректирующих фильтров в контуре управления динамическими системами.

Первые примеры разработки методов управления нелинейными системами относятся также к началу 60-х годов и связаны в основном с методами решения нелинейных дифференциальных уравнений.

Последовавшее затем развитие вычислительных машин с массовым параллелизмом, к которым относятся и нейрокомпьютеры, привело к созданию принципиально новых алгоритмов и методов управления нелинейными динамическими системами. Они связаны с нейросетевыми алгоритмами решения обыкновенных нелинейных дифференциальных уравнений и, как следствие, с

включением нейрокомпьютера в контур управления нелинейной динамической системой.

В последнее время интеллектуальное управление становится широко распространенным средством для многих технических и промышленных приложений. Такие системы управления обладают способностью адаптации к возмущениям, изменениям внешней среды и условиям работы.

В настоящее время исследования в области экспертных систем, традиционно считавшиеся основным инструментом интеллектуальных систем, сокращаются, а применение нейросетевых технологий стабильно нарастает.

Искусственные нейронные сети, благодаря своим способностям к самоорганизации и обучению, рассматриваются как перспективные средства для разработки интеллектуальных систем высокой точности и надежности.

4 Практическая работа №2. Построение и обучение нейросети

Цель работы:

Построить, обучить и протестировать нейронную сеть.

4.1 Теоретическое введение

На практике нейросети используются в двух видах – как программные продукты, выполняемые на обычных компьютерах, и как специализированные аппаратно-программные комплексы. Основная задача нейрокомпьютеров – обработка образов, основанная на обучении – та же, что и у биологических нейросистем. Подобно биологическим, искусственные нейросети нацелены на параллельную обработку широкополосных образов. В новой схмотехнике, как и в мозгу, отсутствуют общие шины, нет разделения на активный процессор и пассивную память. Вычисления, как и обучение, распределены по всем активным элементам – нейронам, каждый из которых есть элементарный процессор образов,

так как производит хотя и простейшую операцию, но сразу над большим количеством входов. Как вычисления, так и обучение полностью параллельны

Несмотря на значительное число специализированных аппаратных разработок нейросетей, основное применение в настоящее время получили программные реализации различных нейросетевых парадигм, называемые нейропакетами (нейроэмуляторами). Иначе говоря, нейропакет – это программная оболочка, эмулирующая для пользователя нейросреду на обычном персональном компьютере.

В случае использования нейросетей в виде программных продуктов встроенный параллелизм нейросетевых алгоритмов не используется. Для многих задач при анализе и обобщении баз данных, особенного быстрогодействия и не требуется: для них вполне хватает производительности современных универсальных процессоров. В этих приложениях используется исключительно способность нейросетей к обучению, к извлечению скрытых в больших массивах информации закономерностей.

Преимущества нейроэмуляторов для относительно небольших задач очевидны:

- не нужно приобретать новую специализированную аппаратуру (спецпроцессоры);

- пользователь не должен осваивать особенности программирования на спецпроцессорах и способы их сопряжения с базовым компьютером;

- универсальные ЭВМ не накладывают никаких ограничений на структуру сетей и способы их обучения, тогда как спецпроцессоры зачастую имеют ограниченный набор встроенных в них функций активации и достигают пиковой производительности лишь на определенном круге задач.

В настоящее время существует несколько десятков фирм-производителей нейропакетов, а предлагаемый ими ассортимент составляет несколько сот наименований.

4.2 Общие сведения об нейроэмуляторе NeuroPro

Программа NeuroPro 0.25 является свободно распространяемой бета-версией разрабатываемого программного продукта для работы с нейронными сетями и производства знаний из данных с помощью обучаемых искусственных нейронных сетей.

Основные возможности программы:

- 1 создание нейропроекта;
- 2 подключение к нейропроекту файла (базы) данных;
- 3 редактирование файла данных;
- 4 добавление в нейропроект нейронной сети слоистой архитектуры с числом слоев нейронов от 1 до 10, числом нейронов в слое – до 100 (число нейронов для каждого слоя сети может задаваться отдельно);
- 5 выбор алгоритма обучения, назначение требуемой точности прогноза, настройка параметров нейронной сети;
- 6 обучение нейронной сети решению задачи предсказания или классификации;
- 7 тестирование нейронной сети на файле данных;
- 8 вычисление показателей значимости каждого из входных сигналов для решения задачи, сохранение вычисленных показателей значимости в файле на диске;
- 9 упрощение нейронной сети;
- 10 генерация и визуализация вербального описания нейронной сети. Редактирование и сохранение вербального описания нейронной сети в файле на диске;
- 11 сохранение нейропроекта на диске.

4.3 Описание меню

Меню программы содержит следующие пункты, относящиеся к нейронным сетям и работе с ними:

Файл – базовые операции с файлами:

Создать – создает новый файл проекта.

Открыть – открывает существующий на диске файл проекта.

Сохранить – сохраняет файл. Возможно сохранение файлов проекта, файлов данных, показателей значимости входных сигналов сети, вербального описания сети.

Сохранить как – сохраняет файл под другим именем или в другом формате. Возможно сохранение файлов проекта, файлов данных, результатов тестирования, показателей значимости входных сигналов сети, вербального описания сети.

Выход – завершение работы программы.

Нейросеть – операции с нейронными сетями. Операция выполняется над активной в данный момент в нейропроекте нейросетью.

Обучение – обучение нейронной сети.

Тестирование – тестирование нейронной сети.

Анализ обучающего множества – вычисление константы Липшица для обучающей выборки, формирование набора конфликтных примеров в задачнике (набора примеров, для которых одинаковым значениям входных сигналов соответствуют разные значения выходных сигналов).

Сокращение числа входных сигналов – удаление наименее значимых входных сигналов.

Сокращение числа нейронов – удаление наименее значимых нейронов сети.

Сокращение числа синапсов – удаление наименее значимых синапсов сети.

Сокращение числа неоднородных входов – удаление наименее значимых неоднородных входов нейронов сети.

Равномерное упрощение сети – сокращение максимального числа приходящих на нейрон сети сигналов до задаваемого пользователем.

Бинаризация синапсов сети – приведение значений весов синапсов и неоднородных входов нейронов к выделенным значениям.

Вербализация – генерация вербального описания нейронной сети.

Значимость входов – подсчет и отображение значимости входных сигналов нейронной сети.

Возмущение весов синапсов – добавление случайных поправок к весам синапсов сети.

Настройка – операции по настройке. Настройки действуют в пределах нейропроекта, сохраняются в файле нейропроекта и восстанавливаются при его чтении программой.

Метод оптимизации – выбор метода оптимизации для обучения сети. Из реализованных в настоящее время в программе методов (градиентный спуск, модифицированный ParTan метод, метод сопряженных градиентов и квазиньютоновский BFGS-метод) при создании нейропроекта автоматически предлагается ParTan.

Норма накопления значимости – выбор нормы накопления градиента при подсчете показателей значимости. При создании нейропроекта автоматически выбирается норма в виде суммы модулей.

Веса бинаризованных синапсов – выбор набора выделенных значений, к которым приводятся веса синапсов. После приведения веса синапса к выделенному значению синапс исключается из обучения; при переходе от более подробного набора выделенных значений к менее подробному те из бинаризованных ранее синапсов, веса которых не попадают в выделенные значения, возвращаются в обучение.

Расширять задачник при контрастировании входов – при сокращении числа входных сигналов после каждого исключения сигнала будет делаться перестройка обучающего множества. Если ранее в обучающее множество не попадали те примеры, где имелись пропущенные значения у входных признаков, то после перестройки обучающего множества в него могут войти новые примеры – те примеры, где пропуски находятся на месте исключенного входного признака.

4.4 Нейросетевая парадигма

На настоящий момент имеется возможность работы только со слоистыми и слоистыми монотонными нейронными сетями с числом слоев нейронов от 1 до 10, числом нейронов в слое – до 100. Число нейронов в слое не зависит от числа входных сигналов и числа выходных сигналов. После последнего слоя нейронов сеть имеет слой адаптивных сумматоров с числом сумматоров, равных числу выходных сигналов, с которых и снимаются выходные сигналы сети.

Число входных и выходных сигналов сети ограничено максимальным числом полей в файле данных и не может в сумме превышать 255 или 511 (в зависимости от файла данных) сигналов. Каждому входному и выходному сигналу соответствует поле в файле данных. Однако, при обработке качественных признаков, каждому входному или выходному полю могут соответствовать несколько входных или выходных сигналов сети.

Веса синапсов при обучении могут изменяться в диапазоне $[-1,1]$, при создании сети иницируются случайными числами.

Нелинейная функция нейрона имеет вид $f(A)=A/(c+|A|)$, где c – параметр крутизны переходного участка сигмоидной функции, который задается при создании сети в диапазоне $[0.0001,1]$ и не изменяется при обучении. Параметр крутизны можно задавать отдельно для каждого слоя сети.

4.5 Подача и снятие сигналов

Нейронная сеть может обрабатывать только поля в файле данных, содержащие числовые значения. Из всего набора полей в файле данных можно выбрать отдельные поля для обработки нейросетью.

Обрабатываемые поля могут быть непрерывными (количественными) и дискретнозначными (качественными).

Каждое количественное поле в файле данных будет соответствовать одному входному или выходному сигналу сети. Перед подачей количественных входных

полей нейронной сети происходит их нормировка в диапазон значений $[-1,1]$ по каждому полю. Выходные сигналы сети нормируются в диапазон истинных значений.

Каждому качественному полю может соответствовать от 2 до 20 входных или выходных сигналов сети (зависит от числа дискретных состояний, которых может принимать данное поле). Дискретные состояния могут задаваться не только отдельными значениями, но и интервалами значений. Входные сигналы сети при обработке качественных признаков принимают значения -1 либо 1 . Для каждого выходного качественного признака строится правило принадлежности его к тому или иному дискретному состоянию, основываясь на значениях выходных сигналов сети для каждого его дискретного состояния. Правило подачи и интерпретации качественных признаков может меняться в зависимости от того, обладают ли состояния качественного признака отношением упорядоченности и предшествования (например, состояния качественного признака, соответствующие тому или иному направлению ветра («Север», «Запад», «Юг», «Восток») отношением предшествования не обладают, а состояния при кодировании качественным признаком степени тяжести заболевания – обладают отношением упорядоченности и предшествования (более тяжелой стадии всегда предшествовала более легкая).

4.6 Точность решения задачи

Нейросеть может решать одновременно несколько задач как прогнозирования (предсказания значений нескольких количественных признаков), так и задачи классификации (предсказания состояний нескольких качественных признаков), так и задачи прогнозирования и классификации одновременно. Для каждой из задач могут быть установлены свои требования к точности.

Для прогнозируемого качественного признака точность означает максимально допустимое отклонение прогноза сети от истинного значения признака. Желательно задавать как можно менее жесткие требования к точности. Это ускорит как процесс

обучения, так и процесс упрощения сети. Также задачу можно будет решить на основе нейронной сети с меньшим числом слоев и нейронов, и, обычно, на основании меньшего числа входных сигналов. Требуемая точность ни в коем случае не должна превосходить погрешностей получения сигнала (погрешностей измерительных приборов, погрешностей округления значений при вводе их в компьютер). Так, если значение признака изменяется в диапазоне $[0,10]$ и измерительный прибор имеет собственную точность ± 0.1 , то нельзя требовать от сети предсказания с точностью ± 0.01 .

Для качественного признака точность (надежность) означает уверенность в принадлежности качественного признака тому или иному дискретному состоянию. Чем больше уровень требуемой уверенности, тем более надежно должна сеть диагностировать отличия каждого дискретного состояния от других.

4.7 Обучение нейронной сети

На этапе обучения происходит вычисление синаптических коэффициентов в процессе решения нейронной сетью задач, в которых нужный ответ определяется не по правилам, а с помощью примеров, сгруппированных в обучающие множества. Так что нейросеть на этапе обучения сама выполняет роль эксперта в процессе подготовки данных для построения экспертной системы. Предполагается, что правила находятся в структуре обучающих данных. Для обучения нейронной сети требуются обучающие данные. Они должны отвечать свойствам представительности и случайности или последовательности. Такие данные представляют собой ряды примеров с указанием для каждого из них значением выходного параметра, которое было бы желательно получить. Действия, которые при этом происходят, можно назвать контролируемым обучением: «учитель» подаем на вход сети вектор исходных данных, а на выходной узел сообщает желаемое значение результата вычислений. Контролируемое обучение нейросети можно рассматривать как решение оптимизационной задачи. Ее целью является минимизация функции ошибок на данном множестве примеров путем выбора

значений весов. Достижение минимума называется сходимостью процесса обучения. Поскольку ошибка зависит от весов нелинейно, получить решение в аналитической форме невозможно, и поиск глобального минимума осуществляется посредством итерационного процесса – так называемого обучающего алгоритма.

Обучение нейронной сети на некотором задатнике производится градиентными методами оптимизации, градиент вычисляется по принципу двойственности. В программе реализованы четыре алгоритма оптимизации:

- Градиентный спуск.
- Модифицированный ParTan.
- Метод сопряженных градиентов.
- Квазиньютоновский BFGS-метод.

При создании нейропроекта в качестве алгоритма по умолчанию принимается ParTan.

Примеру задатника соответствует запись (строка) файла данных. Для включения записи файла данных в задатник в записи должны присутствовать данные для всех полей, используемых нейронной сетью в качестве входных и выходных. Обучение прекращается при достижении заданной точности решения задачи либо при невозможности дальнейшей оптимизации.

4.8 Упрощение нейронной сети

Упрощение нейронной сети выполняется на основе вычисленных показателей значимости. Показатели значимости вычисляются в пяти точках в пространстве адаптивных параметров сети и усредняются в одной из норм.

Процедура упрощения строится как последовательное исключение из сети наименее значимого входного сигнала или элемента сети, дальнейшее дообучение сети и исключение другого сигнала или элемента, если удалось дообучить сеть до заданной пользователем точности. В противном случае процедура упрощения останавливается.

Предоставляются следующие возможности по упрощению сети:

- Сокращение числа входных сигналов.
- Сокращение числа нейронов сети.
- Сокращение числа синапсов сети.
- Сокращение числа неоднородных (пороговых) входов нейронов сети.
- Равномерное упрощение сети, чтобы на каждый нейрон сети приходило не более n сигналов.
- Бинаризация весов синапсов и неоднородных входов сети. Бинаризованные синапсы и неоднородные входы в дальнейшем не обучаются.

Сокращение входных сигналов и нейронов может достигаться и при выполнении других операций по упрощению сети, а не только при целенаправленном сокращении именно входных сигналов и нейронов.

Нейрон сети считается удаленным, если у него удалены все выходы или его сигнал не используется нейронами следующего слоя.

Входной сигнал считается удаленным, если удалены все синапсы, по которым этот сигнал поступал на нейроны первого слоя сети.

Удаленные при упрощении элементы физически остаются в нейронной сети, но при генерации вербального описания сети не вносятся в вербальное описание.

4.9 Вербализация нейронной сети

При генерации вербального описания в тексте перечисляются используемые поля файла данных, правила их предобработки для подачи сети, описание нелинейных функций нейронов, функционирование нейронной сети послойно и понейронно, правила нормировки выходных сигналов сети в диапазон истинных значений. Сигналам, генерируемым нейронами сети, присваиваются некоторые имена и в дальнейшем пользователь при анализе сети может именовать эти сигналы в терминах проблемной области.

Получив вербализованное описание нейронной сети, пользователь может попытаться восстановить правила, сформированные сетью для решения задачи –

записать на естественном языке алгоритм решения неформализованной задачи предсказания или классификации.

4.10 Этапы решения задач с использованием нейронных сетей

Решение задачи с применением нейронной сети может состоять из следующих этапов:

1 **Определение проблемы.** Решение о том, какую входную информацию использовать и что сеть должна делать. Формулируется постановка задачи и выделяется набор ключевых параметров, характеризующих предметную область. Обычно для этого привлекаются эксперты по соответствующей проблеме, которые перечисляют набор факторов и данных, учитываемых ими при принятии решения.

Постановка задачи для нейронной сети имеет определенную специфику. Прежде всего, необходимо решить, относится ли решаемая задача к одному из стандартных типов нейросетевых постановок: задачи классификации (категоризации), задачи построения функциональной модели (идентификации систем), задачи прогноза, задачи оптимизации и нейроматематики, задачи управления и, наконец, задачи распознавания образов и обработки сигналов.

Нестандартная постановка задачи для нейроЭВМ обычно требует проведения специальных исследований и большого опыта решения других задач. На этом этапе обязательно нужно ответить на вопрос: а нужна ли вообще для решения данной задачи нейронная сеть? Вполне возможно (и часто бывает так), что решение может быть получено алгоритмическим способом. В этом случае применение нейроимитатора обычно оказывается не эффективным.

2 **Решение о том, как представлять информацию и сбор ее в виде примеров.** Выбирается парадигма нейронной сети, наиболее подходящая для решения данного класса задач.

Тип используемой нейросети во много диктуется поставленной задачей. Так, для задачи классификации удобными могут оказаться многослойный персептрон и сеть Липпмана-Хемминга. Персептрон также применим и для задач идентификации

систем и прогноза. При решении задач категоризации потребуются карта Кохонена, архитектура встречного распространения или сеть с адаптивным резонансом. Задачи нейроматематики обычно решаются с использованием различных модификаций модели Хопфилда.

3 Определение архитектуры нейронной сети. Выбор входов сети и спецификация выходов, а также количества нейронов скрытого слоя (в случае многослойной сети). Подготавливается возможно более широкий набор обучающих примеров, организованных в виде векторов входных параметров, ассоциированных с известными выходными значениями.

4 Обучение сети. Входные вектора по очереди предъявляются нейронной сети, а полученное выходное значение сравнивается с эталоном. Далее прослеживается путь от выходов сети до ее входного слоя, и все нейронные межсоединения, способствовавшие принятию правильного решения, поощряются – их веса увеличиваются. Те же пути, которые привели к ошибкам в компонентах выходного вектора, наказываются уменьшением веса (как правило, в зависимости от величины ошибки).

Идеальной является ситуация, когда можно получить произвольно много различных данных для задачи. В этом случае следует позаботиться об отсутствии систематических ошибок и уклонений в данных. Целесообразно включение в обучающую выборку прежде всего тех данных, которые описывают условия, близкие к условиям дальнейшего использования нейросистемы.

Для практических целей следует часть обучающей выборки не использовать при обучении, а применить для последующего тестирования работы нейросети. Полезно понимать, что очень большая выборка обучающих данных сильно замедлит процесс обучения без существенного улучшения результата.

Если в вашем распоряжении имеется весьма ограниченный объем данных, то потребуется анализ его достаточности для решения вашей задачи. Обычно это оказывается весьма непростым вопросом. В любом случае, обучающих данных должно быть больше, чем обучаемых параметров нейросети.

5 Тестирование обученной сети. Процесс повторяется до тех пор, пока суммарная ошибка в реакции сети на всех предъявленных примерах не станет меньше наперед заданной величины, то есть сеть не придет в стационарное состояние.

6 Настроенную и обученную сеть можно использовать, предъявляя ей реальные ситуации.

4.11 Практическая часть

Одним из преимуществ нейронных сетей является возможность решения неформализованных задач классификации и прогноза тех задач, явный алгоритм решения которых не известен.

Эмулятор нейронных сетей NeuroPro можно использовать как систему для социального прогноза, предсказывающую исход выборов. Возможности эмулятора рассматриваются на примере выборов президента США

Какая партия победит на очередных выборах в США – правящая или оппозиционная? На первый взгляд кажется, что это зависит от личностей кандидатов и от их программ.

Оказывается, что если предвыборные кампании всех кандидатов отработаны добросовестно и все участники сделали все возможное, то выбор практически предопределяется объективными признаками ситуации и не зависит ни от программ, ни от личностей, ни от названий партий, а только от того, к какой партии принадлежит правящий президент. В одних ситуациях побеждает правящая партия, в других – оппозиция.

Результаты выборов можно предсказать на основании ответов на 12 вопросов:

- 1 Правящая партия была у власти более одного срока?
- 2 Правящая партия получила больше 50% на прошлых выборах?
- 3 В год выборов была активна третья партия?
- 4 Была серьезная конкуренция при выдвижении от правящей партии?
- 5 Кандидат от правящей партии был президентом в год выборов?

6 Год выборов был временем спада или депрессии?

7 Рост среднего национального валового продукта на душу населения больше 2,1%?

8 Правящий президент произвел существенные изменения в политике?

9 Во время правления были существенные социальные волнения?

10 Администрация правящей партии виновна в серьезной ошибке или скандале?

11 Кандидат правящей партии – национальный герой?

12 Кандидат оппозиционной партии – национальный герой?

Эмулятору NeuroPro для обучения предоставляется информация об итогах выборов в США за 100 лет. Для всех выборов известны ответы на 12 вопросов и какая партия, когда победила. После обучения нейронная сеть дает свой вариант для дополнительного набора ответов, т.е. предсказывает результаты выборов.

Данные для эмулятора представлены в виде таблицы Excel, где в качестве полей представлены 12 вопросов, описанные выше, а в качестве записей – ответы на все эти вопросы по всем годам проведения выборов, начиная с 1860 и заканчивая 1980. Ниже представлено описание аббревиатур всех полей:

- MORE1 – правящая партия была у власти более одного срока;
- MORE5 – правящая партия получила больше 50% голосов избирателей на прошлых выборах;
- THIRD – в год выборов была активна третья партия;
- CONC – была серьезная конкуренция при выдвижении от правящей партии;
- PREZ – кандидат от правящей партии был президентом в год выборов;
- DEPR – год выборов был временем спада или депрессии;
- VAL2_1 – рост среднего национального валового продукта на душу населения больше 2,1%;
- CHANG – правящий президент произвел существенные изменения в политике;
- WAVE – во время правления были существенные социальные волнения;

– MIST – администрация правящей партии виновна в серьезной ошибке или скандале;

– R_HERO – кандидат правящей партии – национальный герой;

– O_HERO – кандидат оппозиционной партии – национальный герой.

Ответы на вопросы представлены в виде 0 и 1. Единица означает утвердительный ответ на вопрос, ноль – отрицательный. Выходное поле обозначается как I___, где результатом является значение переменной, соответствующее коду выигравшей партии (1 – правящая партия, 2 – оппозиционная партия).

Работа с нейронными сетями возможна только в рамках некоторого нейропроекта. Для того чтобы создать нейропроект, необходимо выбрать пункт меню «Файл/Создать» или нажать кнопку «Создать» в панели кнопок, при этом появляется окно следующего вида (рисунок 4.1).

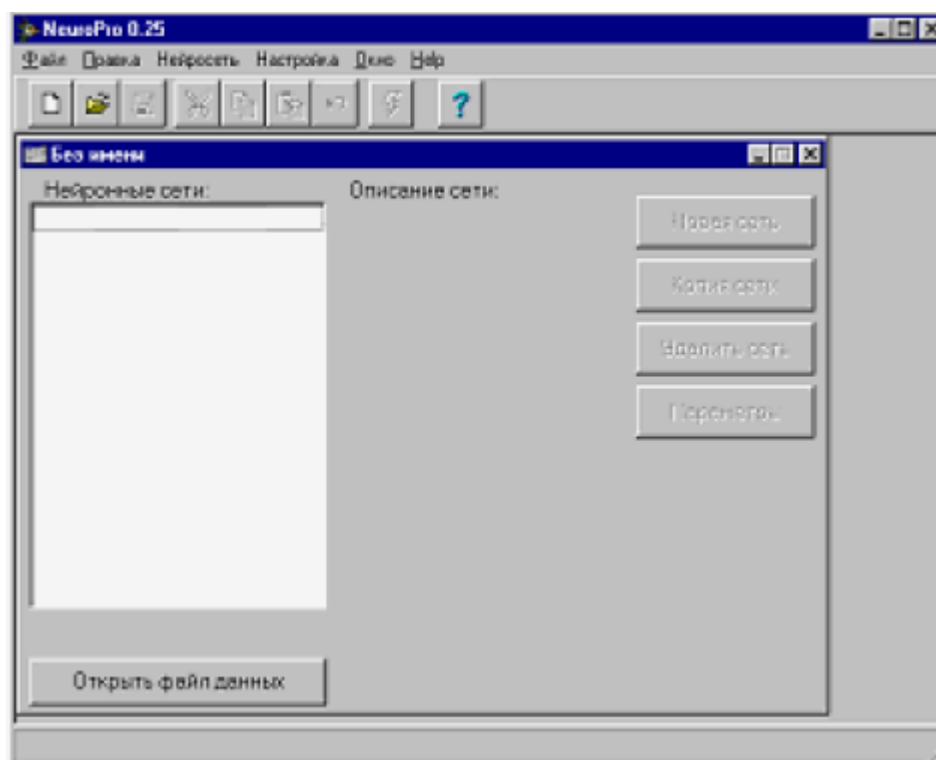


Рисунок 4.1 – Вид окна после выбора «Файл/Создать»

После создания нейропроекта в него можно вставлять нейронные сети с помощью кнопки «Открыть файл данных» и работать с ними. Созданный нейропроект может также быть сохранен при помощи команды меню «Файл/Сохранить», «Файл/Сохранить как» или нажатием на кнопку «Сохранить» на панели инструментов.

В дальнейшем возможна работа с сохраненными файлами нейропроекта. Для этого необходимо выбрать пункт меню «Файл-Открыть» или нажать кнопку «Открыть» и выбрать в открывшемся диалоговом окне имя желаемого проекта.

Большинство операций с нейронными сетями требуют присутствия подключенного к нейропроекту файла данных.

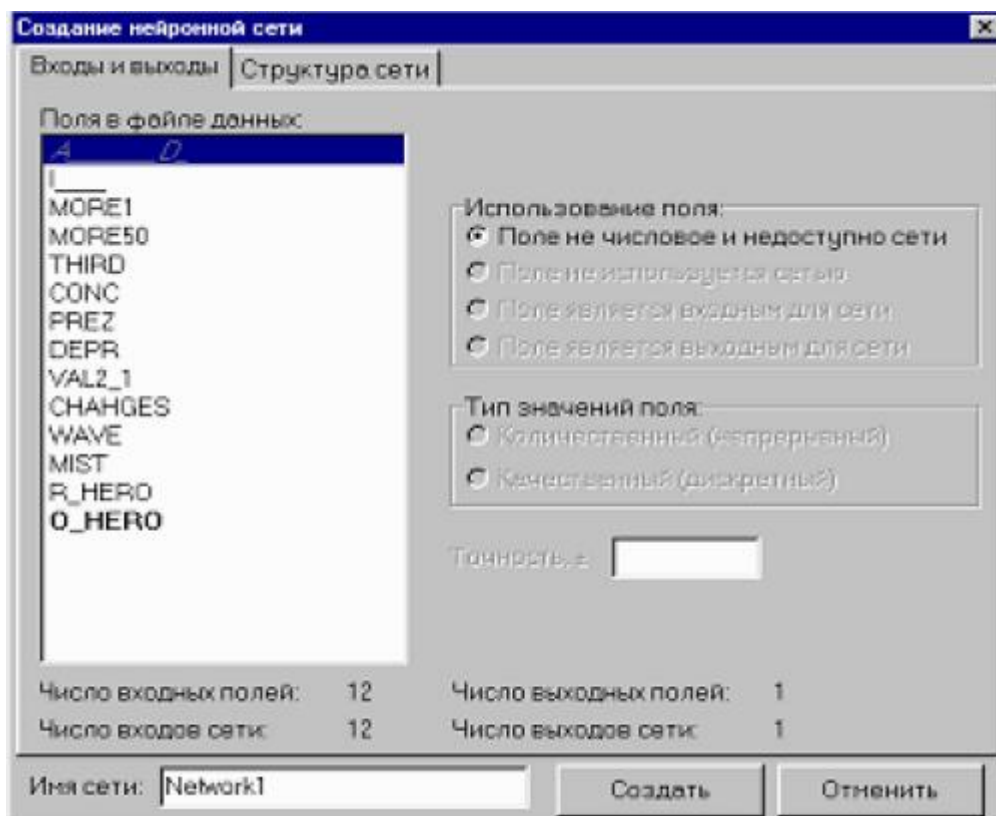


Рисунок 4.2 - Окно для создания нейронной сети

Для подключения файла данных или его замены необходимо нажать кнопку «Открыть файл данных» в окне нейропроекта и далее выбрать имя необходимого файла данных. Открытый файл отображается в собственном окне, где

предоставляется возможность его редактирования. При подключенном файле данных можно проводить операции создания новых сетей, их обучения, тестирования и упрощения.

Для создания новой нейронной сети необходимо нажать кнопку «Новая сеть» в окне нейропроекта и заполнить окно для создания нейронной сети (рисунок 4.2).

Для нашего примера поле I___ необходимо пометить как выходное, все остальные поля будут входными. После нажатия кнопки «Создать» создается нейронная сеть со следующими параметрами:

- число входных полей: 12;
- число входов сети: 12;
- число выходных полей: 1;
- число выходов сети: 1;
- слой 1: 10 нейронов;
- слой 2: 10 нейронов;
- слой 3: 10 нейронов.

Созданную нейронную сеть можно далее обучать, тестировать, упрощать и сохранять на диске вместе с нейропроектом.

Для обучения активной в данный момент в нейропроекте нейронной сети необходимо выбрать пункт меню «Нейросеть-Обучение». Если в файле данных имеются все необходимые поля, и он не пустой, то запускается процесс обучения сети. При этом на экран выводится «Окно обучения», где пользователь имеет возможность наблюдать процесс обучения и при необходимости самостоятельно завершить обучение нажатием кнопки «Завершить» (рисунок 4.3).

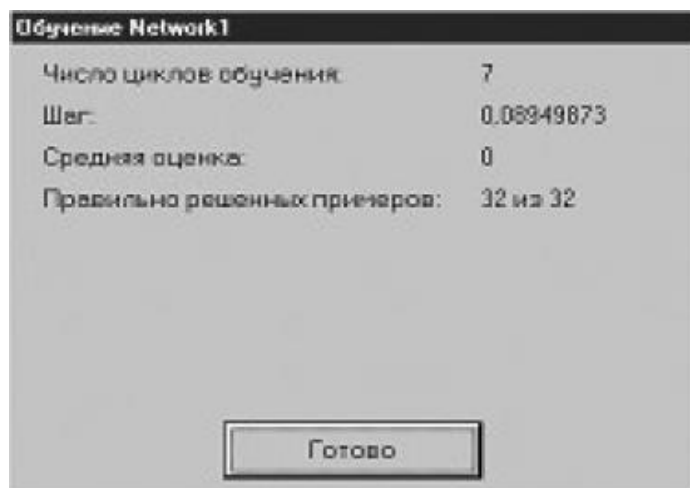


Рисунок 4.3 - Отображение процесса обучения сети

Обучение прекращается при достижении нулевого значения средней оценки на задачнике, в случае невозможности дальнейшего улучшения оценки либо при аварийных ситуациях (нулевой или бесконечный шаг в направлении оптимизации).

Имея нейронную сеть, можно посмотреть, насколько точно она прогнозирует значения выходных полей в файле данных. Для тестирования нейронной сети выбираем пункт меню «Нейросеть/Тестирование». Результат тестирования сети выводится в «Окно тестирования сети» (рисунок 4.4).

В окне представлен результат прогноза сети по всем годам, средняя и максимальная ошибки при прогнозировании.

№	_____	Прогноз сети	Ошибка
29	1	0.9671643	0.03283572
30	2	1.995806	0.004194498
31	2	2.017038	-0.01703811
32	2	2.027888	-0.02788758
		Правильно:	32 (100%)
		Неправильно:	0 (0%)
		Всего:	32
		Ср.ошибка:	0.0250394
		Макс.ошибка:	0.06044531

Рисунок 4.4 - Результат тестирования сети

Возможно тестирование сети на другом файле данных. Для этого необходимо сначала подключить к проекту другой файл данных, а затем протестировать сеть. Результат тестирования можно сохранить в текстовом файле на диске. Далее этот файл можно обрабатывать в другой программе.

Не все входные сигналы сети и синапсы нейронов необходимы для правильного решения задачи. Часто можно достаточно сильно упростить сеть без ухудшения точности решения задачи.

При проведении процесса упрощения сети сокращается число входных сигналов сети. В тех случаях, когда можно правильно решить задачу на основе меньшего набора входных данных, это может в дальнейшем сократить временные и материальные затраты на сбор информации.

После упрощения нейронная сеть может приобрести логически прозрачную структуру и ее возможно будет более просто реализовать на аппаратной платформе.

Известно, что почти невозможно понять, как обученная нейронная сеть решает задачу. После упрощения нейронная сеть становится достаточно обзримой и можно попытаться построить алгоритм решения задачи сетью на основе графического представления или словесного описания структуры сети.

Для упрощения нейронной сети имеются следующие операции в меню «Нейросеть»:

Сокращение числа входных сигналов – удаление наименее значимых входных сигналов (рисунок 4.5)

Упрощение Network1	
Число циклов обучения:	53
Шаг:	0.00015
Средняя оценка:	0.04
Правильно решенных примеров:	30 из 32
Удалено входов:	6 (6 из 12)
Удалено нейронов:	0 (0 из 30)
Удалено синапсов:	60 (60 из 361)
Бинаризовано синапсов:	0 (0 из 301)

Готово

Рисунок 4.5 – Изменение параметров сети при ее упрощении

При уменьшении количества входов с 12 до 6 и уменьшении количества промежуточных связей (синапсов) уменьшается количество правильно решенных примеров, соответствующих заданному уровню ошибки, 30 вместо 32.

Упрощение нейронной сети можно производить по различным параметрам.

Сокращение числа нейронов – удаление наименее значимых нейронов сети.

Сокращение числа синапсов – удаление наименее значимых синапсов сети.

Сокращение числа неоднородных входов – удаление наименее значимых неоднородных входов нейронов сети.

Равномерное упрощение сети – сокращение числа приходящих на нейроны сети сигналов до количества, задаваемого пользователем.

Бинаризация синапсов сети – приведение значений весов синапсов и неоднородных входов нейронов к выделенным значениям.

После упрощения сети обученная сеть минимизируется по числу входных параметров и связей. При использовании эмулятора NeuroPro как системы для социального прогноза, предсказывающей исход выборов в США оказалось, что для надежного предсказания исхода выборов в США достаточно знать ответы всего на пять вопросов, приведенных ниже в порядке значимости:

- 1 Была серьезная конкуренция при выдвижении от правящей партии?
- 2 Во время правления были существенные социальные волнения?

- 3 Год выборов был временем спада или депрессии?
- 4 Правящий президент произвел значительные изменения в политике?
- 5 В год выборов была активна третья партия?

Остальные признаки слабо связаны с итогами выборов, т.е. они мало влияют на результат, и их можно не использовать при моделировании. Система позволяет построить многофакторную модель и убрать из нее факторы, мало влияющие на результат, т.е. существенно упростить исходную модель для данной задачи.

На примере представленной практической работы можно посмотреть, как решается задача прогнозирования с помощью нейронных сетей. При решении какой-либо задачи прогнозирования можно также упростить сеть без существенного ухудшения точности решения задачи.

4.12 Варианты заданий к контрольной работе

Данная практическая работа выполняется в несколько этапов.

1 Создайте новый проект в NeuroPro. Составьте аббревиатуры для каждого вопроса, и внесите правильные ответы в проект (для того чтобы, проекту было с чем сравнивать ответы тестируемых студентов).

- 1) Традиционный признак системы обработки данных – это неотделимость операционного и фактуального знаний? (1)
- 2) Характерный признак системы баз данных – это неотделимость операционного и фактуального знаний? (0)
- 3) Характерный признак системы, основанной на знаниях – это выделение операционного знания в базу знаний? (1)
- 4) Информация – это факты, характеризующие объекты, процессы и явления предметной области, а также их свойства? (0)
- 5) Информация – это данные, рассматриваемые в каком-либо контексте, из которого пользователь может составить собственное мнение? (1)

- 6) Знания – это закономерности проблемной области, полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области? (1)
- 7) Данные соответствуют прагматическому аспекту отражения действительности? (0)
- 8) Информация соответствует семантическому аспекту отражения действительности? (0)
- 9) Знаниями являются осмысленные факты? (1)
- 10) В качестве единиц знаний используются правила и факты? (1)
- 11) Традиционный признак системы обработки данных – это выделение метазнания, описывающего структуру знаний и отражающего модель предметной области? (0)
- 12) Отличие ИИС от обычных ИС заключается в наличии БД? (0)

2 Проведите тестирование у двух студентов по отдельности, они должны ответить на 12 вопросов на тему «Интеллектуальные системы управления». Ответами на вопросы должны быть 0 и 1. Единица означает утвердительный ответ на вопрос, ноль – отрицательный.

Выходное поле обозначается как $I_{___}$, где результатом является значение переменной, соответствующее коду выигравшего студента (1 –Первый студент, 2 – Второй студент).

3 На основе анкетирования подготовить файлы в соответствии с требованиями пакета NeuroPro.

4 Провести пробное обучение. В случае, если нейронная сеть не может обучиться решению задачи проанализировать задачник на предмет непротиворечивости. Если противоречий нет, обратиться к преподавателю.

5 Составить отчет о проделанной работе и полученных результатах.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Пупков, К.А. Интеллектуальные системы (Исследование и создание): учебное пособие / К.А. Пупков, В.Г. Коньков. – изд. 1-е. – Москва, 2001. – 194 с.
- 2 Методы искусственного интеллекта / Осипов Г.В. – М.: Физматлит, 2011. – 296 с.
- 3 Жукова, М. Н. Управление информационной безопасностью. Ч. 2. Управление инцидентами информационной безопасности [Электронный ресурс]: учеб. пособие / М. Н. Жукова, В. Г. Жуков, В. В. Золотарев. – Красноярск: Сиб. гос. аэрокосмич. ун-т, 2012. – 100 с.
- 4 Информационные аналитические системы [Электронный ресурс]: учебник / Т. В. Алексеева, Ю. В. Амириди, В. В. Дик [и др.]; под ред. В. В. Дика. – М.: МФПУ Синергия, 2013. – 384 с. – (Университетская серия).