

В статті запропоновано новий оптимальний за швидкістю алгоритм навчання вейвлет-нейрона, що забезпечує налаштування в реальному часі не тільки синаптичних коефіцієнтів, але й параметрів розтягання та зсувів дочірніх вейвлетів. Алгоритм має як слідкуючі, так і згладжуючі властивості, що дозволяє ефективно використовувати вейвлет-нейрони як самостійно, так і в складі штучних нейронних мереж для рішення задач прогнозування нелінійних часових рядів довільної природи.

ючі властивості, що дозволяє ефективно використовувати вейвлет-нейрони як самостійно, так і в складі штучних нейронних мереж для рішення задач прогнозування нелінійних часових рядів довільної природи.

УДК 681.3

А.И. Липчанский, У.И. Лесовик, Зидат Хабис

СИНТЕЗ ЗАДАНОЙ НЕЙРОННОЙ СЕТИ В ПРОГРАММИРУЕМУЮ ЛОГИКУ

Описывается методология проектирования нейронных сетей (НС) для прогнозирования. С помощью пакета Neuro-Pro V 0.25 проведено обучение НС. Алгоритм функционирования НС описан на языке VHDL с учетом оптимальных параметров НС. Разработана модель НС, проверена правильность функционирования на тестовых примерах. При помощи пакета SynplifyPro 7.0 от Synplicity произведен синтез устройства с ориентацией на микросхему семейства Virtex-II XC2V6000. Приведены структуры НС, аппаратные затраты сведены в таблицу.*

уюмую логику, авторами допускаются следующие упрощения: структура НС заранее задана, оптимизирована и обучена по алгоритму обратного распространения известных количества нейронных элементов с весовыми коэффициентами каждого из нейронов, необходимых для правильной работы НС.

АКТУАЛЬНОСТЬ ПРОБЛЕМЫ И ПОСТАНОВКА ЗАДАЧИ

Аппаратная реализация нейронных сетей (НС) обеспечивает максимально возможное быстродействие при решении специфических нейросетевых задач, таких как диагностика, распознавание образов, управление, прогнозирование и т.д. Такие НС позволяют реализовать преимущества свойственного им параллелизма и работают на несколько порядков быстрее по сравнению с их программной симуляцией. Особенно это актуально для задач реального времени, связанных, например, с ориентацией объектов в пространстве (манипулятор робототехнического комплекса, беспилотный транспорт и т.п.).

В качестве примера для реализации рассмотрим НС, приведенную в [2], где описана целевая задача медицинской диагностики, моделирование и оптимизация структуры заданной НС. Несмотря на то, что статистическая медицинская диагностика не относится к задачам реального времени, методика синтеза НС, описанная ниже, применима для любой произвольно заданной структуры НС.

Использование современных перепрограммируемых пользователем базовых матричных кристаллов (FPGA) для реализации НС можно считать альтернативой по отношению к только программным или только аппаратным вариантам имплементации. Современные FPGA имеют значительный объем ресурсов - до 10 млн. системных вентилях на кристалл, высокую производительность с рабочими системными частотами до 420 МГц и возможностью реконфигурации кристалла непосредственно на рабочем месте [1].

Статья содержит пример реализации НС и описание этапов ее разработки от функционального описания модели до оценки временных и аппаратных характеристик. Приведена оценка оптимальности использования модели НС, полученной после описания используемого нейроалгоритма, в коде VHDL.

АППАРАТНЫЙ СИНТЕЗ ЗАДАНОЙ НС В ВИДЕ ПЛМ

Применение современной широкой номенклатуры FPGA, а также средств их автоматизированного проектирования и отладки, позволят объединить достоинства программного и аппаратного способов реализации НС. При этом могут быть решены вопросы, связанные с проблемой уменьшения стоимости и сроков разработки, а также снижения аппаратных затрат при схемной реализации НС и дальнейшей ее модификации.

Для реализации НС в аппаратном виде полученная структурная модель НС была представлена в кодах языка VHDL. Полученная на языке VHDL модель содержит функциональную модель НС и составленный "тест-бенч" с необходимыми тестами для проверки функционирования до синтеза, после синтеза и после размещения логики на чипе. Следующим этапом модель была синтезирована в цифровую логику для последующей имплементации на таких цифровых устройствах как FPGA.

Поскольку в данной статье основное внимание уделяется методическим вопросам синтеза НС в программи-

VHDL (VHSIC Hardware Description Language, IEEE Standard 1076-1993)[3] является языком описания аппаратных средств общего назначения, используется для описания и моделирования функционирования широкого ряда цифровых систем и позволяет, с принятием подходящего из стилей кодирования, синтез кода в цифровую логику [4]. Язык позволяет проектировщику использовать одинаковое описание модели при создании и симуляции предварительной концепции проекта и, если

она была завершена успешно, использовать эту модель для ее синтеза в цифровую логику. При использовании методологии проектирования "сверху-вниз" описание проектируемого устройства первоначально формируется на системном/алгоритмическом уровне, и после этого модель анализируется на функциональную правильность. Затем генерируются более детальные модели с увеличением точности описания и учетом всех аспектов аппаратурной реализации.

Для реализации данной НС был выбран чип Xilinx Virtex II FPGA. Он имеет встроенные блоки умножителей 18x18, специальную логику ускоренного переноса для высокоскоростных арифметических операций, частичное реконфигурирование, гибкие логические ресурсы: до 122 880 триггеров, до 122 880 16-ти разрядных сдвиговых регистров на базе LUT, поддержка многоходовых умножителей и логических функций. Сделан по технологии 0.15-мкм с 8-слойной металлизацией и 0.12-мкм быстродействующими транзисторами. Частичное реконфигурирование позволяет производить очень быстрые нарастающие изменения свойственные алгоритму обучения. Архитектура, базирующаяся на мультиплексорах, гарантирует отсутствие конфликтных ситуаций при прохождении сигналов, и таким образом, что произвольные конфигурационные данные не приведут чип к самоуничтожению.

В процессе синтеза основная задача - это описание модели НС синтезируемым стилем. Если код VHDL компилируется и правильно моделируется, то не обязательно, что он будет правильно синтезироваться. И даже если код VHDL правильно синтезируется, результирующая реализация может быть неэффективной. Обычно, инструментальные средства синтеза в качестве входа будут принимать только подмножество VHDL. Другие изменения должны быть сделаны в коде VHDL, поэтому инструмент синтеза "понимает" намерения проектировщика. Чтобы произвести эффективную реализацию, могут потребоваться дальнейшие изменения в коде VHDL.

Например, когда используются целочисленные сигналы, важно определить целочисленный диапазон. Если никакой диапазон не определен, VHDL синтезатор может интерпретировать целочисленный сигнал для представления 32-разрядного регистра, так как максимальный размер целого числа VHDL - 32 бита. Когда целочисленный диапазон определен, большинство синтезаторов реализует целочисленное сложение и вычитание, используя двоичные сумматоры с соответствующим числом битов. Вообще, когда сигналу VHDL присваивается значение, он будет сохранять это значение до тех пор, пока ему не присвоят новое значение. Из-за этого свойства, некоторые VHDL синтезаторы могут подразумевать задвижку, хотя она не была задумана проектировщиком.

Каждый синтезатор имеет собственный функциональный пакет для операций, обычно используемых в аппаратурных моделях. IEEE предоставляет стандартный синтезирующий пакет, включающий функции для арифметических операций над `bit_vectors` и `std_logic`

векторами. Пакет `numeric_bit` определяет арифметические операции `bit_vector` векторов. Пакет содержит два класса предопределенных массивов для подачи беззнаковых и знаковых битовых значений:

```
type unsigned is array (natural range <>) of bit;  
type signed is array (natural range <>) of bit;
```

Числа со знаками представляются в двоичном суммирующем коде. Пакет содержит версии арифметических операций, операций отношения, логических и операций сдвига, которые перезагружаются, так же как и функции преобразования. Пакет `numeric_std` определяет аналогичные операции для `std_logic` векторов. Типы без знака и со знаком определяются как массивы `std_logic` векторов вместо массивов битов. Тип `UNSIGNED` основан на IEEE-стандартной логике, мы также заменили всю `bit` логику на тип `STD_LOGIC`.

VHDL-код проекта содержит функции, которых нет в синтезирующих библиотеках. По этой причине мы заменили `std_logic` на тип `UNSIGNED` или `signed`, чтобы можно было использовать определенные для этих типов перезагружающиеся операторы "+" и "*".

В программе были использованы пакеты библиотеки IEEE:

```
IEEE.std_logic_1164.all;  
ieee.numeric_std.all;  
ieee.std_logic_unsigned.all;  
ieee.std_logic_signed.all.
```

Была также созданы собственные пакеты (например, пакет `global_matrixs` содержит описание матрицы весов первого нейрона первого слоя), содержащие описание типа сигналов весов синапсов нейрона. В дальнейшем можно пользоваться этими пакетами, введя следующую строчку разделе описания используемых библиотек:

```
use global_matrixs.all;
```

Основные проблемы, которые могут возникнуть при реализации НС в реконфигурируемой аппаратуре, связаны с представлением чисел с плавающей точкой и нелинейной активационной функции нейрона.

Работа с числами с плавающей точкой является проблемой, так как арифметические операции и схемы для чисел с плавающей точкой намного сложнее чем для целых чисел, медленнее работают и занимают большую площадь кристалла. Подходящим решением для усовершенствования производительности проекта и уменьшения аппаратурных затрат является конвентирование чисел с плавающей точкой в числа типа `integer`. Конечно, это накладывает некоторые ограничения на точность, но в данном частном случае хорошие результаты были достигнуты. Диапазон изменения весов (-1; 1) - число с плавающей точкой - описывается на языке VHDL как число в формате `real`. Числа формата `real` не синтезируемы, необходимо перейти к представлению весов в целочисленном формате. Для этого было сделано следующее. Описан сигнал, представляющий весовой коэф-

коэффициент синапса нейрона, числом типа unsigned (целое без знака), имея в виду, что таким образом создаем двоичное представление дробной части весового коэффициента. То есть, если весовой коэффициент 2-го синапса первого нейрона $W_{1,2} = 0.5$ в двоичной системе исчисления представляется как 0.1, то в данном проекте он будет представлен как 1000 0000.

В программе, реализующей НС, на входные сигналы X_i подаются признаки в уже нормализованном виде. То есть сигналы X_i тоже находятся в диапазоне (-1;1) и должны быть представлены типом unsigned. Результат умножения $X*W$ будет меньше единицы. В модели нейрона это синапс, который должен быть суммирован с другими синапсами нейрона. Для того, чтобы результат умножения был представим в удобном для нас виде и впоследствии мог быть использован в суммировании сдвигаем произведение $Qp1$ на 8 разрядов вправо, а затем берем младшие 8 битов. Таким образом, мы округляем полученное нами произведение до 8-ми бит:

```
Qout_frac1:= Qp3 Srl n;
Q_frac1:= Qout_frac1 (n-1 downto 0);
```

Другая проблема представления арифметических операций в цифровой аппаратуре заключается в реализации активационной функции нейрона. Некоторые активационные функции, например такие как сигмоидная (используемая в модели нейрона при создании модели с помощью пакета NeuroPro 0.5), нуждаются в некоторых модификациях для того чтобы упростить ее для проектирования в аппаратуре. В нашем случае, сигмоидная функция была замещена кусочно-линейной функцией (Рис. 1).

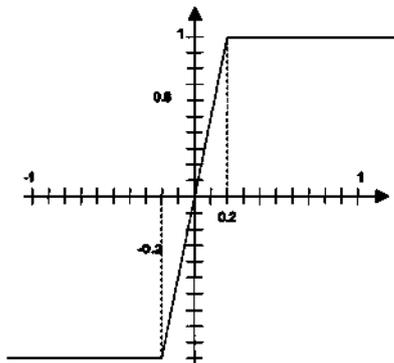


Рисунок 1

Таким образом, активационная функция может быть описана на языке VHDL следующим образом:

```
if ADD_frac2 <= COMP then
    Q_B:="11111111";
else Q1:= ADDITION*ADD_f;
    Q1:=Q1 sll 3;
    Q_B:= Q1(10 downto 3);
end if;
```

То есть, если значение на выходе сумматора лежит в диапазоне от -0,2 до 0,2, то Y будет вычисляться по функции $y=5x$, в другом случае функция будет равна -1 или 1 (1 - если выход сумматора положителен).

АРХИТЕКТУРА СИНТЕЗИРУЕМОЙ НС

Полученная НС состоит из трех слоев. Входной слой имеет с четыре нейрона, на которые параллельно поступает информация с тестовой выборки. Каждый нейрон насчитывает различное количество синапсов - входные сигналы (признаки) поступают на каждый из нейронов входного слоя. Первый нейрон имеет три синапса, второй - 14, третий - 12, четвертый - 13.

Второй скрытый слой состоит из трех нейронов, которые обрабатывают данные, поступившие от нейронов первого слоя. Каждый из нейронов второго слоя имеет 4 входные шины, представляющие синапсы нейрона и сигнал bias - сигнал смещения.

Третий слой - слой суммирования состоит из двух суммирующих элементов, суммирующих выходы от 3-х нейронов предыдущего слоя.

Матрица весов описывается как одномерный массив, элементами которого являются весовые коэффициенты каждого из синапсов нейрона. Матрица весов отдельна для каждого из нейронов НС.

При описании модели НС на языке VHDL были созданы следующие компоненты: четыре нейрона первого слоя, содержащие описание в четырех отдельных файлах - sample3.vhdl, neuron1l_2.vhdl, neuron1l_3.vhdl, neuron1l_4.vhdl; один компонент описывающий нейроны второго слоя (они одинаковы по архитектуре) в файле neuron2l_comp.vhdl и суммирующий элемент суммирующего слоя в файле sum_component.vhdl. Второй слой состоит из трех компонентов нейрона второго слоя, суммирующий слой состоит из двух суммирующих элементов и схемы сравнения, так как на выходной триггер поступает результат с того суммирующего элемента, чей выход больше.

Каждый из компонентов, представляющих нейрон, содержит в себе определенное количество синапсов, один сумматор и пороговую функцию. Шина, отвечающая за подачу входных сигналов, а также другие входные шины имеют размерность 8 бит (7 downto 0), так как большая точность не требуется.

Синапс в модели нейрона реализован операцией умножения $X_{i,j}*W_{i,j}$, где

$X_{i,j}$ - входной признак поступающий на j-тый синапс i-го нейрона,

$W_{i,j}$ - весовой коэффициент j-ого синапса i-го нейрона.

Суммирование оперирует с числами в формате signed. Для этого, перед тем как суммировать, необходимо преобразовать результат умножения из формата unsigned в signed с учетом знака произведения.

В ходе работы над проектом было выяснено, что синтезированная схема полностью зависит от способа описания функционирования устройства. Описание НС и контроллера было сделано на структурном уровне с помощью языка VHDL, и в результате, после синтеза получили такие элементы логики как регистры, сумматоры, триггера и другие логические блоки, которые были сопоставлены к соответствующим объектам структуры.

Для функционального моделирования был использован пакет фирмы Aldec Active-HDL 5.0.

Результаты симуляции схемы до синтеза и после него совпадают, что означает правильность синтеза схемы.

ОПИСАНИЕ СИНТЕЗИРОВАННОЙ МОДЕЛИ

Синтез окончательных VHDL файлов был осуществлен с использованием пакета SynplifyPro 7.0 от Synplicity® и ориентирован на Virtex-II семейство ПЛИС с архитектурой FPGA. Была использована XC2V6000 микросхема семейства Virtex-II. Она содержит 6млн системных вентилях 144 блока умножителей и блочную память емкостью 2592 Кбит.

Synplify Pro - пакет предназначенный для синтеза логических элементов, ориентированный на FPGA (Field Programmable Gate Arrays) и CPLD, разработанный фирмой Synplicity®. Работа с Synplify Pro начинается с высокоуровневых проектов, написанных на языках Verilog и VHDL (языки описания аппаратуры). Используя собственную Behavior Extracting Synthesis Technology (B.E.S.T.) технологию, пакет конвертирует HDL в маленькие, высокопроизводительные списки соединений проекта (design netlists), оптимизированные для общераспространенных поставщиков технологий. При необходимости, Synplify Pro может производить VHDL and

Verilog списки соединений после синтеза (post-synthesis netlists) которые можно использовать для моделирования после синтеза, чтобы проверить правильность синтеза.

Рис.2 содержит окончательную схему устройства. Устройство состоит из контроллера и самой НС. Блок контроля работает на системной частоте 20 MHz и предоставляет требуемые сигналы контроля. Контроллер управляет работой НС. Он синхронизирует работу каждого слоя и сообщает, когда можно подавать на входы следующий тестовый вектор.

На следующем рисунке изображена схема НС, синтезированная с помощью SynplifyPro 7.0. На рисунке видны четыре нейрона во входном слое и два компонента скрытого слоя и слоя суммирования.

Каждый из нейронов содержит определенное количество логических вентилях, реализующих функцию нейронного элемента.

На рис.4 изображена схема первого нейрона входного слоя. Он состоит из 5-ти регистров (для инвертирования шины), 2-х сумматоров, 4-х DE-триггеров, 10-ти мультиплексоров 2-в-1, 4-х умножителей, 1 триггер-латч, 2-х входных элементов "или", "и" и "исключающее или".

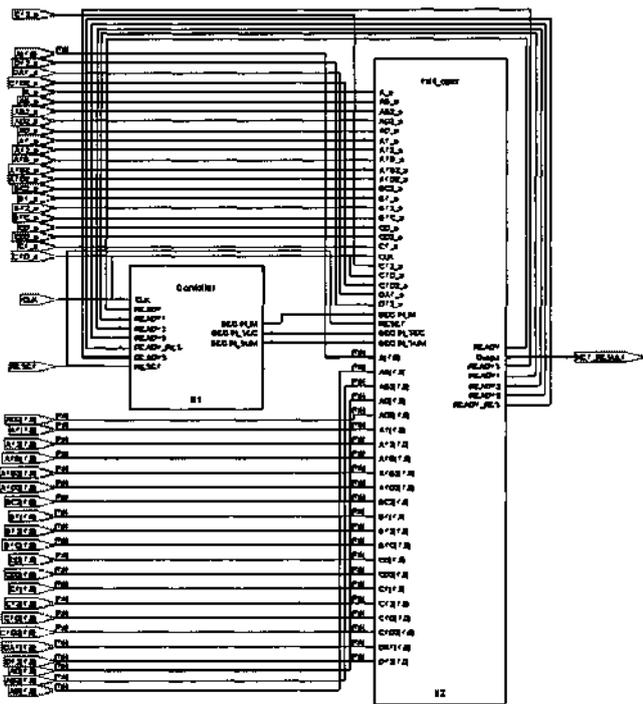


Рисунок 2

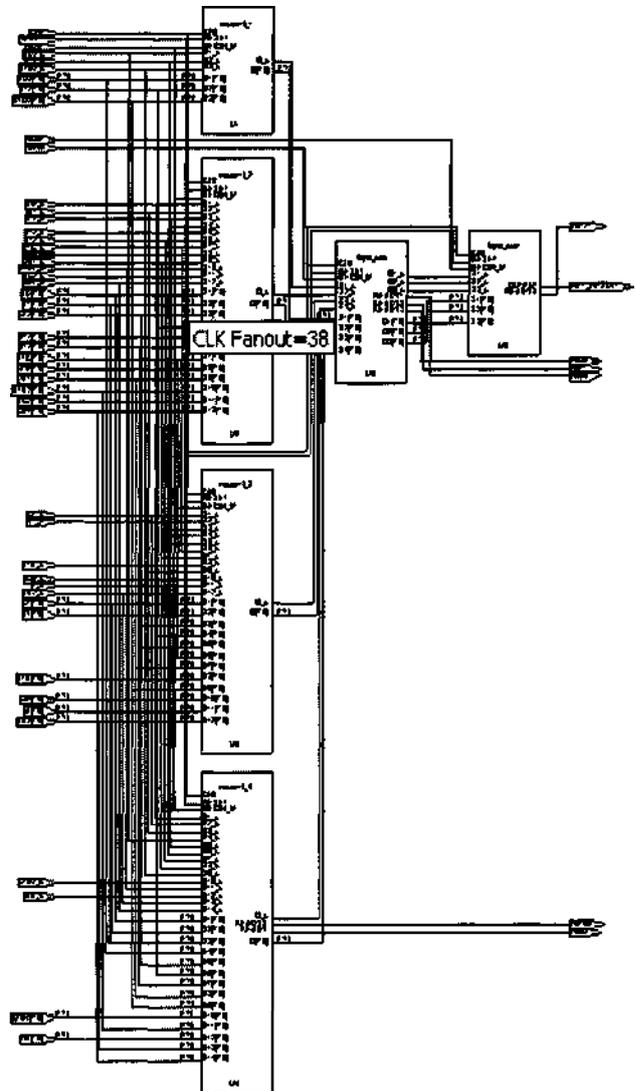


Рисунок 3

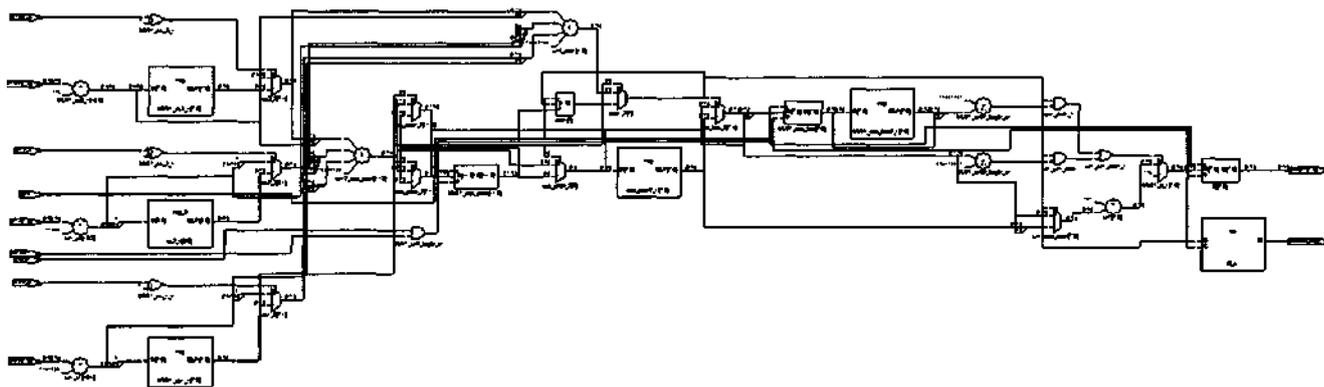


Рисунок 4

Необходимо проверить правильность функционирования устройства после синтеза. Это моделирование покажет, что синтез прошел успешно. Симмуляция после этапа place and route необходима для уверенности, что проект будет работать с требуемой скоростью, беря во внимание время загрузки логических вентилей и задержку распространения сигнала в линиях.

В табл.1 показаны сведения о различных ресурсах, использованных в FPGA.

Таблица 1

Число CLB's	3471/600000 (3%)
Число bonded IOBs	201/ (%)
Число глобальных буферов	1/8 (12 %)
Сумма использованных вентилей	3948
Минимальный период	25.405 ns
Максимальная частота	39.362 MHz
Максимальная задержка на линии	1.148 ns
Средняя задержка связи	3.494 ns
Average Connection Delay on critical nets	0.000 ns
Средняя расфазировка синхронизирующих импульсов	0.248 ns
Максимальная задержка контакта	1.148 ns
Средняя задержка связи на худших связях	9.184 ns

ЗАКЛЮЧЕНИЕ

Даже центральный процессор относительно высокого быстродействия не может осуществить обучение НС с большим количеством нейронов и синапсов, а также формировать ответ сети в реальном масштабе времени. Реализованная аппаратно НС может производить арифметические операции намного быстрее по сравнению с ее программной реализацией. FPGA обрабатывают дан-

ные, поступающие параллельно, с высокой скоростью, что сокращает общее время вычислений. Кроме этого, архитектура FPGA достаточно простая и может быть без существенных затрат расширена до 128 входов.

К недостаткам реализованной аппаратно НС следует отнести такие факторы как сложная схема синхронизации при сложных имплементациях. Кроме этого, количество входных/выходных линий ограничено ресурсами чипа. Аппаратурная реализация НС не имеет гибкости присущей программным средствам и для нее трудно реализовать процесс обучение сети.

В статье было продемонстрирована методология проектирования предварительно обученной НС для решения задачи прогнозирования. Были оценены аппаратные и временные затраты.

НС имеет цифровую реализацию базирующуюся на языке VHDL, которая обеспечивает эффективное использование площади кристалла и быстрый автоматизированный технологический процесс проектирования.

ПЕРЕЧЕНЬ ССЫЛОК

1. Кнышев Д.А., Кузелин М.О. ПЛИС фирмы "Xilinx": описание структуры основных семейств. - М.: Издательский дом "Додэка-XXI", 2001.
2. Сулима Т.Н., Лесовик У.И. Использование метода нейронных сетей для прогнозирования состояния здоровья подростков по результатам медико-генетического обследования // Проблемы бионики. Харьков, № 57, 2002. С. 61-66
3. "IEEE Standard VHDL Language Reference Manual", ANSI/IEEE Std 1076-1993
4. Sjöholm S. and Lindh L. (1997). "VHDL for Designers", Prentice-Hall, 1997, ISBN 0-13-473414-9

Надійшло 26.03.04

The implementation of the neural network for the prediction on the programmable logic / A.I. Lipchansky, U. I. Lesovik //The example of № № realization is considered. Also description of all its design stages from № № function model description to its timing and hardware characteristics estimation is considered. № № structural model is presented in VHDL code. Through SynplifyPro 7.0 package from Synplicity the system synthesis with the orientation on Virtex-II XC2V6000 family is made out. The estimation of the optimality of the synthesized № № model utilization is accomplished. № № structures are shown; hardware costs are taken to the table.*

Розглядається методологія проектування нейронних мереж для прогнозування. За допомогою пакета Neuro-Pro V 0.25 проведено навчання нейронної мережі(НМ). Подано приклад реалізації нейромережі та опис етапів її розробки з функційного опису моделі НМ до оцінки її часових і апаратних характеристик. Структурна модель НМ при-

ведена в кодах мови VHDL. За допомогою пакета SynplifyPro 7.0 синтезовано пристрій на мікросхемах сімейства Virtex-II XC2V6000. Здійснена оцінка оптимальності виконання синтезованої моделі НМ. Приведені структури НМ, апаратні витрати зведені в таблицю.

УДК 004.93:007.52

Г. Сетлак

ИСПОЛЬЗОВАНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ КЛАССИФИКАЦИИ В МЕНЕДЖМЕНТЕ

В работе представлены результаты исследования возможности использования различных видов искусственных нейронных сетей для решения задач классификации. Целью исследований были анализ и оценка используемых методов классификации.

1 ВВЕДЕНИЕ

Проблемы классификации являются одними из наиболее часто возникаемых и решаемых задач как в повседневной хозяйственной деятельности, так и в экономическом анализе этой деятельности организации. Решение задач классификации заключается в разработке, конструкции таких правил и закономерностей, которые позволили бы распознать определённые явления или объекты и определить их принадлежность к каким-то классам (группам, категориям). Задача значительно усложняется, если при определении классов необходимо брать во внимание очень большое количество характеристик исследуемых объектов и явлений (как например, при оценке конкурентоспособности предприятия), тем более, если эти характеристики слабо определены или трудно формализуемы.

Раньше для решения задач классификации использовались так называемые таксономические методы. При этом таксономия рассматривается как статистическая научная дисциплина, занимающаяся разработкой основ и процедур классификации. Главная трудность во всех стандартных статистических методах заключается в том, что большая часть знаний, используемых для классификации, представляет собой распределение случайных переменных. Особенно большие проблемы появляются при использовании непараметрических методов классификации для многомодальных распределений [1,2]. В последние годы для решения сложных задач классификации всё чаще используются искусственные нейронные сети. На основе многочисленных исследований подтверждено, что нейронные сети обеспечивают высокую эффективность распознавания [2,5]. При этом отличаются исключительной устойчивостью перед случайными помехами. Нейронные сети приспособлены для обнаружения сложных зависимостей при отсутствии априорных

знаний об исследуемых процессах или объектах. Кроме этого нейронные сети отлично работают со всеми наиболее сложными распределениями данных. Поэтому с успехом используются везде, где производилось оценивание при помощи статистических методов анализа, таких как регрессионный, кластерный, дискриминантный анализ или временные ряды, в том числе и для решения задач классификации.

Целью данной работы является анализ и оценка возможностей использования для решения задач классификации в стратегическом анализе различных видов нейронных сетей, в том числе многослойного перцептрона, радиальной базисной функции, нейронной карты Кохонена, вероятностной нейронной сети. Практическая реализация и анализ выполнены с использованием программного пакета Statistica Neural Networks 4.0E [1].

2 ПОСТАНОВКА ЗАДАЧИ. ОСНОВНЫЕ ПОЛОЖЕНИЯ

В работе решается задача стратегического менеджмента: выбор наиболее перспективных рынков сбыта для изготавливаемых изделий домашнего хозяйства. Это одна из многочисленных задач классификации, решаемых в процессе функционирования каждой производственной системы. Анализ возможностей использования для решения задач классификации различных видов нейронных сетей в работе выполняется на данных маркетингового анализа рынка изделий, а именно пылесосах. В качестве классифицируемых наблюдений используем информацию о пылесосах, которые нужно распределить по четырём классам, т.е. рынкам сбыта. Входные параметры, характеризующие каждый пылесос, приведены в табл.1. Первый столбец таблицы содержит описание каждой характеристики, тип которой указан во втором столбце. Третий столбец содержит сокращённое имя соответствующего параметра, под которым он фигурирует в программном пакете Statistica Neural Networks.

Выходной параметр классификации - один из четырёх классов рекомендуемого рынка сбыта (в пакете STNN ему соответствует сокращение CLASS). Описания рынков представлены в таблице 2.