

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**Нижегородский государственный университет  
им. Н.И. Лобачевского (ННГУ)**

Институт Информационных Технологий, Математики и Механики

**П.Ю. Белокрылов, П.Д. Басалин, В.В. Банкрутенко**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ  
ПО КУРСАМ «ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ  
СИСТЕМЫ», «СИСТЕМЫ ИСКУССТВЕННОГО  
ИНТЕЛЛЕКТА И ПРИНЯТИЯ РЕШЕНИЙ»**

**СИНТЕЗ СХЕМ ПРОИЗВОЛЬНОЙ КОМБИНАЦИОННОЙ  
ЛОГИКИ В НЕЙРОСЕТЕВОМ БАЗИСЕ**

Рекомендовано методической комиссией ИИТММ  
для студентов ННГУ, обучающихся по направлениям подготовки  
09.03.03 «Прикладная информатика»,  
01.03.02 «Прикладная математика и информатика»

Нижний Новгород

20015

**УДК 681.3.01, 681.5.01**

**Б-43**

**Б-43 Белокрылов П.Ю., Басалин П.Д., Банкрутенко В.В.** Учебно-методическое пособие по курсам «Интеллектуальные информационные системы» и «Системы искусственного интеллекта и принятия решений». Синтез схем произвольной комбинационной логики в нейросетевом базисе. – Нижний Новгород: Нижегородский госуниверситет, 2015. – 26 с.

Рецензент: д. ф.м.-н., профессор Чекмарев Д. Т.

В учебно-методическом пособии излагается нейросетевой подход к синтезу комбинационной части цифрового автомата. В рамках этого подхода задача синтеза сводится к формальной задаче обучения (оптимизации) многослойной нейронной сети прямого распространения. В качестве примера выступает задача синтеза одноразрядного комбинационного сумматора.

Материал учебно-методического пособия предназначен для студентов, обучающихся по направлениям подготовки «Прикладная информатика», «Прикладная математика и информатика»

**© Нижегородский государственный университет им. Н.И. Лобачевского, 2015**

**© Белокрылов П.Ю., Басалин П.Д., Банкрутенко В.В.**

## Введение

В рамках теории структурного синтеза автоматов, базовые положения которой изложены в работе [1], рассматриваются общие приемы, правила и формальные операции, обеспечивающие принципиальную возможность построения структурной схемы автомата путем композиции элементарных автоматов, принадлежащих к заранее определенному (структурно полному) множеству их типов. Однако отсутствие конструктивной методологии практического применения этих положений в общем случае сводит задачу, по сути, к полному перебору.

Исключение составляет так называемый канонический метод структурного синтеза. Он подразделяет все элементарные автоматы на два класса. К первому относятся автоматы с памятью (последовательностные схемы), имеющие более одного внутреннего состояния. Из этих элементов (запоминающих элементов) путем простого их объединения строится запоминающая часть (память) синтезируемого автомата (рис. 1), фиксирующая в себе последовательно все его текущие структурные состояния. Конкретное структурное состояние определяется кодом упорядоченной последовательности структурных состояний запоминающих элементов. Произведение чисел состояний запоминающих элементов должно быть не меньше общего количества состояний синтезируемого автомата. Установление соответствия между обозначениями абстрактных состояний и их структурными представлениями определяет процесс кодирования состояний.

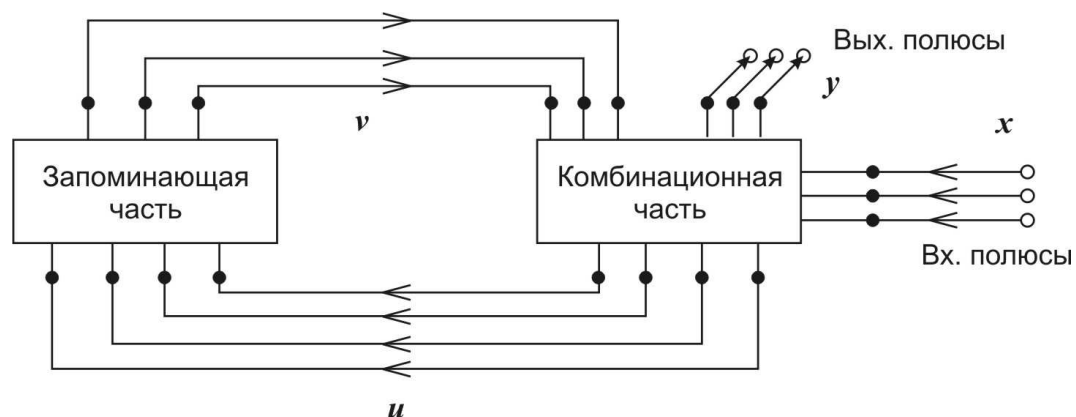


Рис 1. Каноническая схема структурного цифрового автомата

Второй класс объединяет автоматы без памяти (комбинационные схемы), имеющие только одно (исходное) внутреннее состояние. Наиболее элементарными из них являются логические элементы. Предполагается, что результатом композиции последних могут быть только комбинационные схемы. Они могут составить комбинационную часть синтезируемого автомата.

Таким образом, канонический метод структурного синтеза сводит проблему синтеза автомата к синтезу схем его комбинационной части, которая представляет собой многофункциональную (векторную) комбинационную схему.

Задача синтеза комбинационной схемы в общей постановке состоит в построении правильной комбинационной схемы, реализующей любую заданную выходную систему функций в любом функционально полном логическом базисе при условии воздействия на входные узлы лишь входных переменных. Классическая теория комбинационного синтеза разработана применительно к схемам, имеющим один выходной полнос, т.е. выполняющим одну функцию. Для построения схемы реализуемая ею функция должна быть представлена системой булевых выражений в выбранном логическом базисе. Эта система интерпретируется как система булевых функций непосредственных связей правильной комбинационной схемы и однозначно определяет ее структуру. Исходной информацией для получения системы непосредственных связей может служить таблица истинности. По этой таблице легко находится СДНФ (или СКНФ) булевой функции, которая затем минимизируется и приводится к необходимому для построения схемы формату с использованием известных тождеств алгебры выбранного логического базиса.

При решении общей задачи комбинационного синтеза рациональнее всего брать за основу булев логический базис, т.к. для него используется начальная структуризация логической функции в виде СДНФ (СКНФ) с дальнейшим применением известных методов минимизации булевых функций в классе ДНФ (КНФ). Поэтому в процессе синтеза к данному базису сводятся все другие функционально полные системы, взятые за основу. Этот подход используется как общий конструктивный прием при синтезе любых правильных комбинационных схем в двоичном структурном алфавите из произвольных логических элементов. При этом получается одно из возможных решений, которое, как правило, отличается излишней сложностью и требует последующей минимизации.

Задача минимизации булевых функций всегда разрешима в силу конечности функционально полных систем. Для ее решения существуют методы, приемы и алгоритмы, разработанные применительно к конкретным функционально полным системам и учитывающие специфику представления в них функций. Наиболее богатый запас таких приемов предлагает булева алгебра, оперируя с представлениями функций в ДНФ (КНФ) и преследуя цель — уменьшить количество букв в них (каноническая задача минимизации булевых функций). Решение канонической задачи минимизации булевых функций осуществляется в два этапа: на первом этапе по исходной СДНФ (СКНФ) находится сокращенная ДНФ (КНФ) булевой функции; на втором этапе по ней строятся приведенные (тупиковые) ДНФ (КНФ), среди которых выбираются минимальные. Для автоматизации процесса поиска решений более подходят аналитические методы минимизации, хотя их практическое применение во многом оказывается проблематичным в силу следующих причин. Во-первых, с увеличением размерности пространства аргументов функции алгоритмы становятся громоздкими и трудно контролируемыми. Во-вторых, эти методы не являются универсальными с точки зрения произвольного выбора логического базиса для синтеза схем, т.к. они оперируют

в классе ДНФ (КНФ) булевых функций. В-третьих, методы не предусматривают возможность одновременной (согласованной в соответствии с дополнительными критериями) минимизации системы функций. Минимизируя каждую функцию в отдельности, они способны устранить многие пересечения логических выражений, сохранение которых могло бы дать более ощутимый эффект в плане сокращения общего числа задействованных в многофункциональной схеме логических элементов.

Отмеченные недостатки классических методов являются стимулирующим фактором для поиска новых приемов и подходов к решению проблемы синтеза схем произвольной комбинационной логики.

## **Нейросетевой подход к проблеме синтеза схем комбинационной логики**

### ***Концепция искусственной нейронной сети***

*Искусственная нейронная сеть* (ИНС) — это существенно параллельно распределенный процессор, обладающий способностью к приобретению, сохранению и репрезентации опытного знания и сходный с головным мозгом в двух аспектах:

- знание приобретается сетью в процессе обучения;
- для сохранения знания используются силы межнейронных связей (*синаптические веса*).

Составными элементами ИНС являются спецпроцессоры двух видов:

- *искусственные нейроны* (или просто нейроны), суммирующие поступающие на их входы сигналы и преобразующие смещенную на величину порога сумму в соответствии с заданной активационной функцией нейрона;
- *связи между нейронами*, реализующие межнейронные взаимодействия в виде сигналов, умножаемых на синаптические веса связей.

Схематическое представление модели нейрона с входными связями изображено на рис. 2, где использованы следующие обозначения:

$x_1, x_2, \dots, x_n$  — сигналы, поступающие на входные связи нейрона;

$w_1, w_2, \dots, w_n$  — синаптические веса входных связей;

$w_0$  — порог срабатывания нейрона (порог активационной функции);

$$s = \sum_{i=1}^n w_i x_i - w_0;$$

$y = f(s)$  — активационная функция нейрона, реализующая тот или иной вид преобразования суммы взвешенных входных сигналов, смещенной на величину порога активационной функции  $w_0$ .

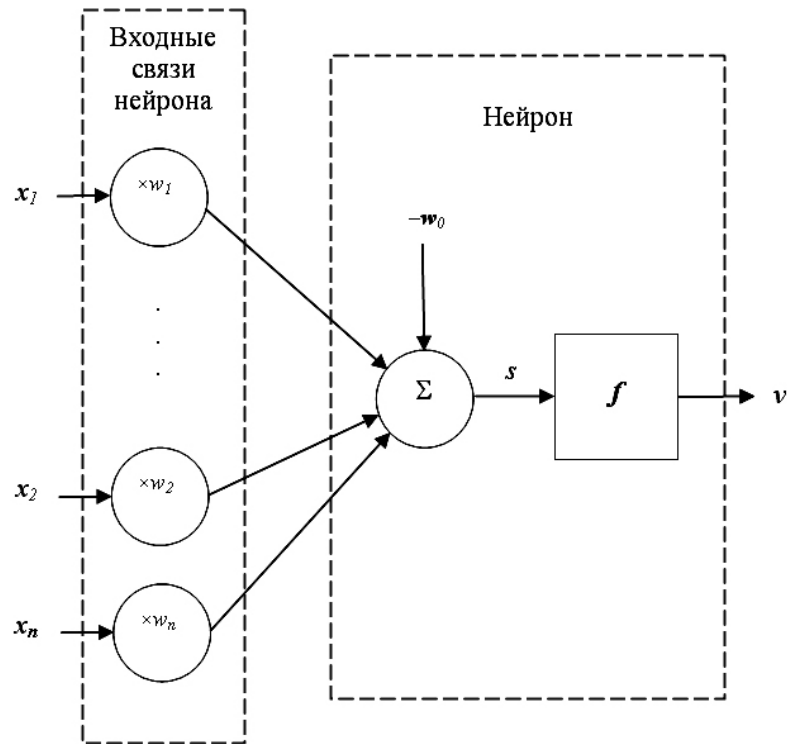


Рис 2. Схематическое представление нейрона с входными связями

Выбор той или иной функции активации определяется спецификой решаемой задачи, алгоритмом обучения, удобством реализации сети в программном и аппаратном исполнении.

В нейронной сети между нейронами  $i$  и  $j$  могут быть установлены направленные связи  $(i, j)$  и  $(j, i)$ , каждой из которых присваиваются, соответственно, веса  $w_{ij}$  и  $w_{ji}$  (синаптические веса) (рис. 3).

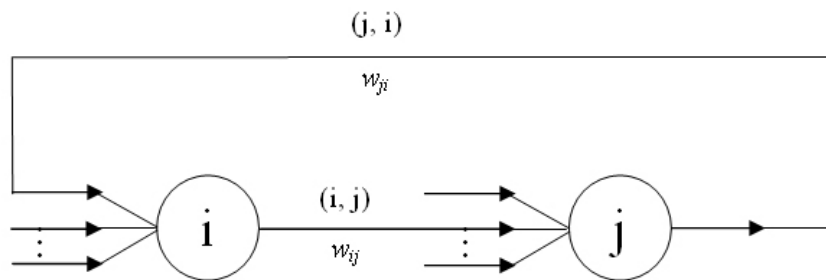


Рис 3. Межнейронные связи

Различные типы архитектур ИНС отличаются характером входных сигналов (*бинарные* или *аналоговые*), структурой сети, типом обучения (*с супервизором*, т.е. внешней процедурой по шаблонам, или через *самоорганизацию* сети). Обучить ИНС — это значит подобрать такие значения синаптических весов и, возможно, порогов активационных функций нейронов, при которых сеть способна выполнять желаемый вид преобразований.

По виду структуры и связанной с ней спецификой обучения и функционирования нейронные сети можно разделить на два больших класса.

К одному из них следует отнести *многослойные сети прямого распространения (персептроны)*, к другому — так называемые *рекуррентные нейронные сети* [2].

В общем случае многослойный персептрон стандартной топологии (рис.4) представляет собой сеть с одним входным (сенсорным) слоем, одним выходным (моторным) слоем и несколькими внутренними (скрытыми, связующими) слоями нейронов.

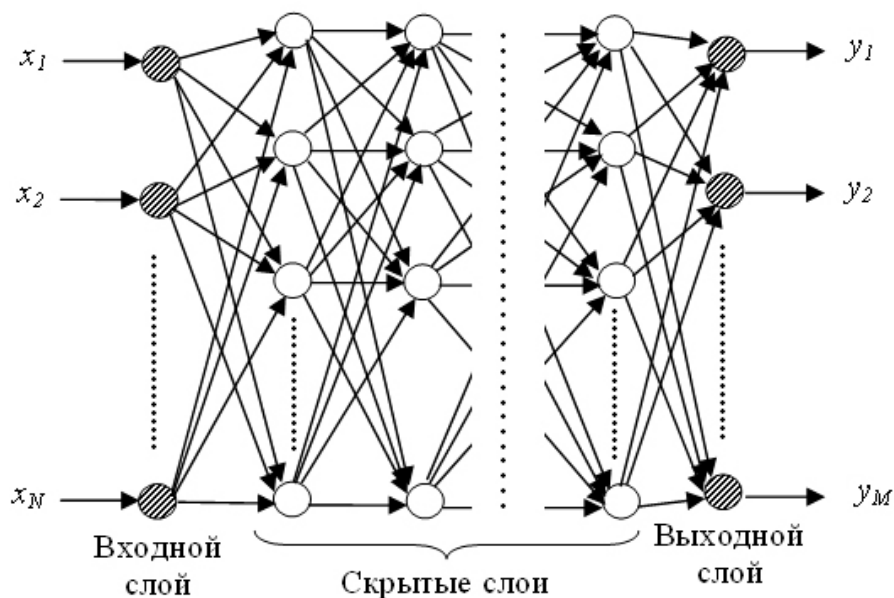


Рис 4. Стандартная топология многослойного персептрона

Характерной его чертой является *прямонаправленность*: информация, преобразуясь, передается от входного слоя через  $K$  скрытых слоев к выходному слою. При этом в стандартной топологии узел  $i$  в слое  $k$  ( $k = 1, 2, \dots, K + 1$ ) связан посредством синаптических весов  $w_{ji}^k$  со всеми узлами  $j$  предыдущего ( $k - 1$ )-го слоя ( $k = 0$  и  $k = K + 1$  обозначают, соответственно, входной и выходной слои).

Модифицированные версии многослойного персептрона могут иметь нерегулярные связи между слоями и прямые связи между несмежными слоями.

Нейроны входного слоя персептрона выполняют функции приема и ретрансляции входных сигналов  $x_1, x_2, \dots, x_N$  на нейроны первого скрытого слоя. Основное (в общем случае нелинейное) преобразование сигналов происходит в скрытых и выходном слоях.

Обучение персептрона — это итеративный целенаправленный процесс изменения значений весов синаптических связей (и, возможно, порогов активационных функций нейронов), реализуемый «внешней» процедурой (алгоритмом обучения) с использованием тренировочных шаблонов (обучение с супервизором) до тех пор, пока сеть не приобретет желаемые свойства.

Тренировочный шаблон представляет собой пару векторов  $\langle \vec{X}, \vec{Y} \rangle$ , один из которых  $\vec{X} = (X_1, X_2, \dots, X_N)$  — вектор известных входных сигналов, другой  $\vec{Y} = (Y_1, Y_2, \dots, Y_M)$  — вектор желаемых выходных сигналов. В процессе обучения

вектора  $\bar{X}^q$  из тренировочного набора  $P = \left\{ \langle \bar{X}^q, \bar{Y}^q \rangle, q = 1, 2, \dots, Q \right\}$  последовательно подаются на вход персептрона, и для каждого из них оценивается ошибка между фактическим  $\bar{y}^q$  и желаемым  $\bar{Y}^q$  откликом (выходом) сети  $e_q = \frac{1}{2} \|\bar{y}^q - \bar{Y}^q\|^2, q = 1, 2, \dots, Q$ . Затем определяется общая ошибка  $e = \sum_{q=1}^Q e_q$ , на

основании которой (а, возможно, и с использованием данных о предыдущих итерациях) алгоритм обучения осуществляет модификацию значений настроечных параметров сети, направленную на уменьшение ошибки. Как вариант, модификация значений варьируемых параметров сети может осуществляться после оценки действия каждого очередного шаблона, т.е. по «локальной» ошибке  $e_q$ . Процесс обучения повторяется до тех пор, пока сеть не приобретет способность выполнять желаемый тип преобразования, заданный тренировочным набором шаблонов  $P$ .

В результате обучения сеть достигает способности правильно реагировать не только на шаблоны множества  $P$ , но и на другие допустимые возмущения, с которыми она никогда ранее не имела дела. В этом состоит так называемое *свойство нейронной сети обобщать*, которое трудно переоценить, особенно при решении трудно формализуемых задач. Ошибка в обобщении, как правило, имеющая место, может определяться тремя причинами:

- ограниченностью размеров сети, ведущей к недостаточному качеству аппроксимации;
- недостаточностью объема обучающей выборки;
- неэффективностью алгоритма обучения.

Схема, иллюстрирующая процесс обучения многослойного персептрона, представлена на рис. 5.

В качестве алгоритма обучения в ней может быть использован предложенный в 1974 году Вербосом алгоритм обратного распространения ошибки (BPE — Back-Propagation Error). Привлекательность этого алгоритма никак не связана с исходно заложенной в нем стратегии градиентного спуска, а скорее определяется встроенным в него рекурсивным механизмом вычисления точных (аналитических) значений производных от функции ошибки по варьируемым параметрам нейронной сети. Поэтому для получения большего эффекта целесообразно сочетание этого механизма с квазиньютоновскими методами оптимизации.



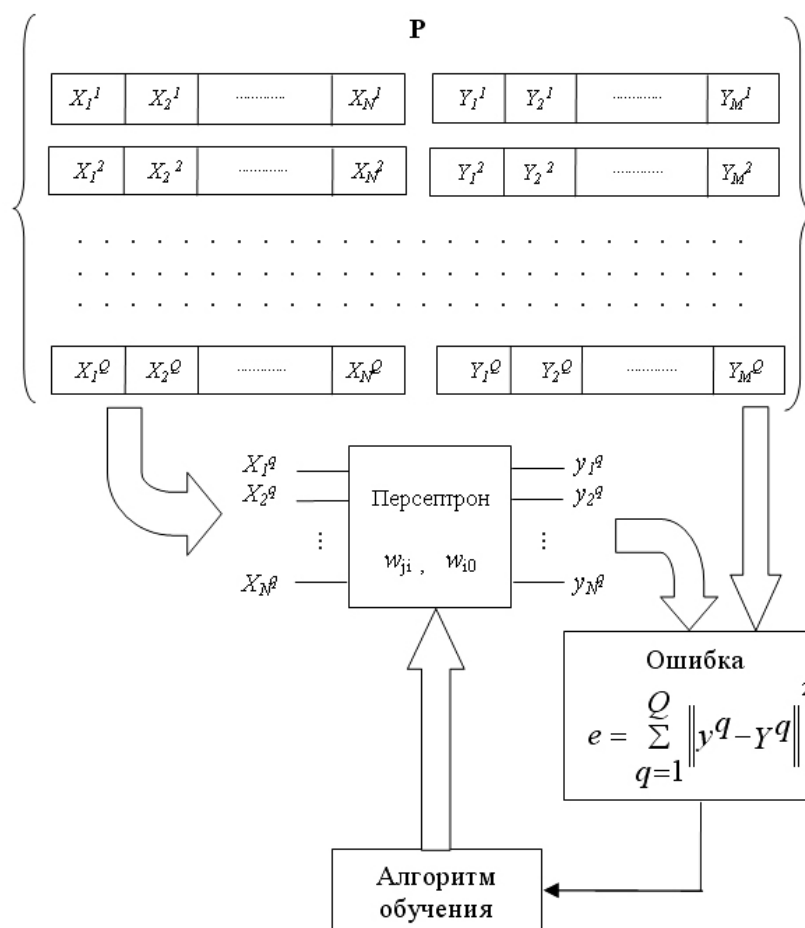


Рис 5. Схематическое представление процесса обучения многослойного персептрона

В принципе, для обучения многослойного персептрона может быть использован любой метод поисковой оптимизации, обладающий способностью работать на существенно нелинейных, овражных и многоэкстремальных функциях с большим числом независимых переменных. В определенной мере таким требованиям отвечают генетические алгоритмы [3].

### **Представление задач в нейросетевом базисе**

Базовые принципы организации современных ВС не могут в полной мере удовлетворить потребности всех прикладных областей, в особенности тех, в которых преобладают задачи неформального плана. Непреодолимыми также остаются проблемы получения точных решений *NP*-полных задач формального характера в пространстве переменных большой размерности. Обозначенные проблемы в определенной степени являются следствием универсальности цифровых ВС и главенствующего при работе с ними принципа подстройки решаемой задачи под ВС. Кардинально изменить этот принцип на обратный («ВС под задачу») позволяет представление задач в нейросетевом базисе.

Обученная конкретному приложению нейронная сеть является спецпроцессором этого приложения (эмулированным в среде универсальной цифровой ВС или реализованным аппаратно в аналоговом, цифровом или гибридном исполнении), учитывающим в своей структуре естественный параллелизм решаемых задач. С позиций коннекционизма (раздела

искусственного интеллекта, связанного с построением, исследованием и развитием моделей функционирования человеческого мозга) нейроны представляются простыми автоматами, и вся функциональная сложность нейронной сети (а значит сложность решаемой ею задачи) определяется ее структурой, т.е. тем, какое количество нейронов каким образом связано между собой. При этом вклад каждой связи в общую структуру определяется не только фактом и характером воздействия одного нейрона на другой, но и силой этого воздействия, т.е. синаптическим весом связи.

Бесконечное множество существующих и потенциально возможных приложений, индивидуальность и неоднозначность их представления в нейросетевом базисе обуславливают неформальный характер процесса построения нейронной сети, приемлемой для решения рассматриваемого класса задач. Побуждающим мотивом для представления конкретных классов задач в нейросетевом базисе является, как правило, стремление получить качественное решение за приемлемое время.

Так, для *формализуемых* задач, имеющих четкие алгоритмы их решения, манипулирующие с переменными, носящими определенный физический или математический смысл, представление в нейросетевом базисе позволяет качественно распараллелить вычислительный процесс с целью повышения его эффективности. Как правило, такой перевод базируется на аналитических преобразованиях, приводящих вычислительный процесс к суперпозиции элементарных процедур, однозначно определяющей структуру (эквивалентной) нейронной сети, виды и пороги активационных функций нейронов, а также синаптические веса межнейронных связей. При этом прослеживается связь синаптических весов и порогов активационных функций нейронов с параметрами исходного алгоритма, что позволяет рассматривать дальнейший процесс его адаптации к конкретным условиям применения как формальную задачу обучения нейронной сети с использованием отработанных методов и алгоритмов.

К категории *трудноформализуемых* относятся задачи, имеющие формальные алгоритмы решения, но не гарантирующие получение качественного результата за приемлемое для пользователя время, особенно в случае высокой их размерности (NP-трудные проблемы). К этой категории следует отнести задачу структурного синтеза цифровых автоматов, конкретнее — задачу синтеза схем произвольной комбинационной логики.

Как будет показано далее, задачу синтеза схемы произвольной комбинационной логики можно сформулировать в виде задачи дискретной параметрической оптимизации на модели логической схемы исходно избыточной структуры, представленной в нейросетевом базисе. При этом задача синтеза сводится к формальной задаче обучения эквивалентной бинарной (или биполярной) нейронной сети (нейроэмулятора). Такой подход позволяет значительно сократить время однократного анализа оптимизируемых и контролируемых характеристик, установить и сохранить физический смысл варьируемых параметров сети в процессе ее обучения, использовать

отработанные алгоритмы для поиска решений, обеспечивающих необходимый компромисс по критериям быстродействия и аппаратной сложности синтезируемых схем.

Существенный эффект на этапе обучения нейронной сети можно получить также от параллельной организации вычислений, реализующих оптимизационную стратегию поиска настроечных параметров сети. В значительной степени этому может способствовать применение эволюционно-генетических алгоритмов оптимизации.

К категории *неформализуемых* принято относить задачи, присущие областям, суть явлений в которых еще не изучена и не может быть представлена адекватными математическими моделями с явными функциями (характеристиками) и параметрами. К таким задачам применяется синтетический подход, при котором моделируется внешнее поведение рассматриваемой системы посредством построения ее модели с помощью параметрических базисных функций или уравнений. Используемая в качестве базиса моделирования нейронная сеть способна запомнить и обобщить конкретные эмпирические зависимости и в дальнейшем воспроизводить динамическое поведение моделируемого объекта. При этом абсолютно не акцентируется внимание на том, с чем связана активность того или иного нейрона, и какой физический смысл приобретают синаптические веса. Объект моделируется сразу целиком в реальных, практических режимах без выделения идеальных элементарных явлений, свойственного аналитическому подходу к моделированию.

Способность нейронных сетей к аппроксимации математически обоснована в ряде работ Колмогорова А.Н., Арнольда В.И. и Горбаня А.Н. В частности, последним было математически обосновано следующее утверждение: для дискретного случая класс многомерных функций (отображений), вычислимых с помощью нейронных сетей, замкнут относительно линейных операций и унарной операции, осуществляемой в соответствии с нелинейной активационной функцией нейрона. Тем самым была показана принципиальная возможность нейросетевой аппроксимации произвольного дискретного отображения. *Но остается открытой проблема поиска топологии сети, способной осуществить эту аппроксимацию.* В каждом конкретном приложении эта проблема должна решаться своими специфическими методами, базирующимися на известных теориях предметных областей и вырабатываемых в них эвристиках. При этом обязательно выполнение следующих этапов:

- определение класса нейронной сети. Для аппроксимации заданного отображения достаточно использовать многослойные сети прямого распространения;
- выбор архитектуры сети. Данный этап связан с нейросетевой интерпретацией моделируемой системы. Он не имеет четкого математического представления и для каждого конкретного случая

решается по-своему. Например, можно осуществить декомпозицию наблюдаемой системы в виде суперпозиции некоторых функциональных зависимостей и линейных операций, определенных на их множестве. Далее, это представление нужно корректно интерпретировать в терминах нейронной сети: видах активационных функций нейронов и преобразований сигналов межнейронными связями;

- определение обучающего множества шаблонов. Данные для составления этого множества получаются в результате наблюдения поведения моделируемой системы в формате «вход – выход».
- выбор исходной топологии (структуры) сети. Данный этап призван определить конкретные параметры структуры сети (число слоев, количество нейронов в них, структуру межнейронных соединений), достаточные для решения поставленной задачи.

Последний из отмеченных этапов является исходным для дальнейшего обучения нейронной сети, в процессе которого с помощью методов оптимизации находится сеть минимальной топологии (например, сеть с минимальным количеством нейронов).

Можно определить две эвристические стратегии поиска минимальной топологии.

Первая из них имеет *регрессивный* характер и заключается в следующем. В качестве исходной выбирается сеть, имеющая заведомо избыточное количество компонентов или структура, являющаяся сетевой интерпретацией уже известного решения. Полученная в ходе обучения топология сети, обеспечивающая нулевое значение функции ошибки отклика и содержащая меньшее количество компонентов, принимается в качестве исходной, для дальнейшего продолжения процесса поиска. Процесс повторяется до тех пор, пока не будет получена сеть, содержащая не уменьшаемое количество своих элементов (например, нейронов). Эта нейросетевая структура может рассматриваться как приемлемое решение.

Вторая стратегия носит *развивающий* характер и заключается в следующем. В качестве начальной топологии выбирается такая, которая имеет заведомо недостаточную вычислительную мощность, т.е. значение функции ошибки для нее гарантированно ненулевое. В структуру сети постепенно включают дополнительные вычислительные элементы — нейроны и связи, до тех пор, пока в процессе обучения не будет получено первое нулевое значение функции ошибки. Эта нейросетевая структура может рассматриваться как приемлемое решение.

Как вариант, исходная для обучения (оптимизации) топология нейронной сети может генерироваться случайным образом на основе структуры с фиксированным количеством нейросетевых элементов. Такая структура может быть определена из соображений конструктивно-технологического исполнения нейронной сети, мощности эмулирующей ИНС вычислительной системы или

создания «универсального» для некоторых классов задач нейросетевого вычислителя. Случайный выбор различных исходных топологий с последующей их оптимизацией позволяет генерировать потенциально весь спектр решений поставленной задачи, среди которых можно выбрать оптимально-компромиссные с точки зрения других критериев (а не только критерия простоты нейромодели).

В качестве примера представления трудноформализуемой задачи в нейросетевом базисе может быть рассмотрена задача синтеза комбинационных схем в нейросетевом базисе.

Синтез комбинационной схемы в малом логическом базисе (SLC – Simple Logic Cells) предоставляет широкие возможности для поиска наиболее рациональных решений. Однако, как было уже отмечено, традиционные методы минимизации булевых функций не являются универсальными с точки зрения произвольного выбора малого логического базиса для синтеза схем, т.к. они оперируют в классе ДНФ (КНФ) булевых функций. Эти методы не предусматривают возможность одновременной (согласованной в соответствии с дополнительными критериями) минимизации системы функций. С увеличением размерности пространства аргументов функции становятся громоздкими и трудно контролируемыми, процесс носит эвристический характер и порождает обширное множество вариантов схем, среди которых не всегда оказываются лучшие.

Подход, преодолевающий указанные ограничения, может состоять в следующем.

Синтезируемая комбинационная схема представляется в виде искусственной нейронной сети (ИНС) из класса многослойных персептронов, имеющая всевозможные межнейронные связи, не нарушающие его свойство прямонаправленности. Число входов персептрона равно числу аргументов  $x = (x_1, x_2, \dots, x_N)$  реализуемых логических функций, а число выходов — числу реализуемых функций  $y = (y_1, y_2, \dots, y_M)$ .

В соответствии со спецификой рассматриваемой задачи, составными элементами нейронной сети являются спецпроцессоры двух видов:

— *искусственные нейроны*, преобразующие поступающие на их входы сигналы в единственный выходной сигнал в соответствии с заданной активационной функцией нейрона;

— *связи между нейронами*, реализующие межнейронные взаимодействия в виде бинарных (принимающих значения 0 или 1) или биполярных (принимающих значения 1 или -1) сигналов, умножаемых на синаптические веса связей.

Виды активационных функций и значения синаптических весов определяется выбором SLC. Так, в случае {И, ИЛИ, НЕ} базиса, активационные функции нейрона представляются следующими выражениями:

$$y_i = f(s) = \begin{cases} 1, & \text{если } s > 0 \\ -1, & \text{если } s < 0 \end{cases}, \quad (1)$$

где  $y_i$  — сигнал на выходе  $i$ -го нейрона;

$$s = \sum_j w_{ji} x_j - w_{i0}, \quad x_j \text{ — сигнал на выходе } j\text{-го нейрона, воздействующего}$$

на  $i$ -тый нейрон ( $x_j = y_j$ ),  $j \in \{1, \dots, N_i\} \subset N$ ;

$N_i$  — общее количество нейронов, непосредственно воздействующих на  $i$ -й нейрон;

$w_{i0} = N_i - 1$  — порог активационной функции  $i$ -го нейрона, реализующего логическую функцию *И*;

$w_{i0} = 1 - N_i$  — порог активационной функции  $i$ -го нейрона, реализующего логическую функцию *ИЛИ*.

Значения синаптических весов  $w_{ji} \in \{-1, 0, 1\}$ . Значение  $w_{ji} = -1$  реализует логическую функцию *НЕ*,  $w_{ji} = 0$  определяет отсутствие данной связи между нейронами,  $w_{ji} = 1$  определяет обычную ретрансляционную связь.

Сигналы, распространяющиеся в данной сети, принимают значения из  $\{-1, 1\}$ .

В случае использования  $\{И-НЕ\}$ ,  $\{ИЛИ-НЕ\}$  базисов активационные функции нейрона представляются следующими выражениями:

$$y_i = f(s) = \begin{cases} 0, & \text{если } s > 0 \\ 1, & \text{если } s \leq 0 \end{cases}, \quad (2)$$

$$y_i = f(s) = \begin{cases} 0, & \text{если } s \geq 0 \\ 1, & \text{если } s < 0 \end{cases}, \quad (3)$$

где  $y_i$  — сигнал на выходе  $i$ -го нейрона;

$$s = \sum_j w_{ji} x_j - w_{i0}, \quad x_j \text{ — сигнал на выходе } j\text{-го нейрона, воздействующего}$$

на  $i$ -тый нейрон ( $x_j = y_j$ ),  $j \in \{1, \dots, N_i\} \subset N$ ;

$N_i$  — общее количество нейронов, непосредственно воздействующих на  $i$ -й нейрон;

$w_{i0} = N_i - 1$  — порог активационной функции  $i$ -го нейрона, реализующего логическую функцию *И-НЕ*;

$w_{i0} = 1$  — порог активационной функции  $i$ -го нейрона, реализующего логическую функцию *ИЛИ-НЕ*.

Значения синаптических весов  $w_{ji} \in \{0, 1\}$ ,  $w_{ji} = 0$  определяет отсутствие данной связи между нейронами,  $w_{ji} = 1$  определяет обычную ретрансляционную связь.

Сигналы, распространяющиеся в данной сети, принимают значения из  $\{0,1\}$ .

Определяется множество обучающих шаблонов  $P = \{ \langle \bar{X}^q, \bar{Y}^q \rangle, q = \overline{1, Q} \}$  роль которых играет исходная таблица истинности, т. е. функциональная таблица автомата ( $\bar{X}^q = (X_1^q, X_2^q, \dots, X_N^q)$  —  $q$ -й вектор известных значений аргументов реализуемых функций,  $\bar{Y}^q = (Y_1^q, Y_2^q, \dots, Y_M^q)$  — вектор соответствующих значений функций,  $Q$  — мощность обучающего множества, т.е. число строк таблицы истинности равно  $2^N$  для полностью определенных функций).

В процессе обучения, вектора  $\bar{X}^q, q = \overline{1, Q}$ , из множества  $P$  последовательно подаются на вход персептрона, и для каждого из них оценивается разность между фактическим  $\bar{y}^q$  и желаемым  $\bar{Y}^q$  откликом (выходом) сети —  $e_q = \frac{1}{2} \|\bar{y}^q - \bar{Y}^q\|^2, q = \overline{1, Q}$ . Затем, определяется общая ошибка

$e = \sum_{q=1}^Q e_q$  — целевая функция процесса оптимизации топологии сети, на

основании которой алгоритм обучения осуществляет модификацию значений настроечных параметров сети, направленную на уменьшение ошибки. Обучение продолжается до тех пор, пока сеть не приобретет способность выполнять требуемый тип преобразования, заданный набором шаблонов  $P$ , то есть  $e = 0$ . Т.о. решением данной задачи будет набор значений варьируемых параметров (синаптических весов межнейронных связей и типов активационных функций нейронов), обеспечивающий минимальное (нулевое) значение функции  $e$ .

Для выбора начальной топологии используется следующий подход. После предварительного исследования поставленной задачи из соображений мощности эмулирующей ИНС вычислительной системы выбирается топология, содержащая достаточное количество нейросетевых элементов для решения поставленной задачи. На ней случайным образом генерируется некоторая начальная структура из множества допустимых, и запускается процесс ее оптимизации. В результате многократного повторения такой процедуры получается набор нейросетевых структур, обеспечивающих нулевое значение функции ошибки отклика сети. Среди них могут встречаться структуры, содержащие меньшее количество нейросетевых элементов по сравнению с другими, но не обязательно минимальное. Данный подход потенциально позволяет получить весь спектр решений поставленной задачи синтеза, возможный для заданной начальной топологии (в том числе и минимальное). Из него могут быть выбраны те, которые обеспечивают приемлемый компромисс в пространстве критериев быстродействия, аппаратной сложности, технологичности исполнения схемы и других.

## **Эволюционно-генетические алгоритмы обучения нейронной сети**

Итак, проблема синтеза схем произвольной комбинационной логики сведена к формальной оптимизационной задаче обучения с супервизором (*supervised learning*) многослойного персептрона:

найти значения параметров  $\vec{Z} = (z_1, z_2, \dots, z_n)$ , обеспечивающие

$$\min_{\vec{Z} \in D^n} e(\vec{Z}), \quad (4)$$

где  $\vec{Z}$  — вектор варьируемых переменных, компонентами  $z_i, i = \overline{1, n}$ , которого являются синаптические веса, определяющие типы межнейронных связей сети, и переменные, задающие виды активационных функций ее нейронов;

$D^n$  — дискретное  $n$ -мерное пространство варьируемых переменных;

$e(\vec{Z})$  — функция ошибки отклика на выходе нейронной сети.

Задача (4) относится к классу задач дискретной оптимизации. Варьируемые переменные в ней носят структурный характер представления объекта оптимизации.

Из природы исходной задачи синтеза схем произвольной комбинационной логики следует, что задача (4) имеет не единственное решение  $\vec{Z}^*$ , обращающее в 0 значение функции критерия ( $e(\vec{Z}^*)=0$ ). При этом зависимость минимизируемой функции от переменных носит существенно нелинейный характер.

Учитывая сказанное, а также ставшее уже традиционным и накопившее определенный опыт применение генетических алгоритмов для обучения нейронных сетей, в основу механизмов обучения многослойного персептрона положен эволюционно-генетический подход.

В пользу эволюционно-генетического подхода говорят следующие его свойства:

- он способен уверенно функционировать в условиях неопределенного представления характера поведения целевой функции и отслеживать ее многоэкстремальность;
- его алгоритмы работают с кодовыми последовательностями, в которых может быть выражен набор параметров, определяющих аргументы целевой функции;
- алгоритмы исследуют сразу множество точек поискового пространства одновременно, что обеспечивает возможность распараллеливания процесса поиска решения задачи;
- алгоритмы используют как вероятностные, так и детерминированные правила для нахождения новых точек пространства поиска;



- алгоритмы могут осуществлять поиск оптимального решения по единой стратегии как для унимодальных, так и для многоэкстремальных функций.

Применительно к рассматриваемой задаче генетический алгоритм обучения нейронной сети выглядит следующим образом.

Для описания эволюционно-генетической модели нейронной сети использована схема непосредственного кодирования, специфицирующая генными кодами каждую синаптическую связь исходной избыточной сети и вид активационной функции каждого нейрона. При этом генетический код особи эволюционирующей популяции представлен в виде двух составляющих. Первая составляющая (слева направо) ставит в соответствие каждому слою нейронов участки хромосомы, определяющие его связи со всеми последующими слоями. Вторая фиксирует типы активационных функций для каждого нейрона скрытых слоев и выходного слоя в порядке возрастания их номеров в сети сверху вниз и слева направо, начиная с нейронов первого скрытого слоя.

Предложенная структура хромосомного набора предполагает возможность применения оригинальных схем генетических операторов репродукции и отбора решений, учитывающих специфику рассматриваемой предметной области. В частности, используются многоточечные операторы кроссовера и мутации, что позволяет порождать новые нейронные сети путем комбинации их фрагментов на основе перестановки связей между нейронами, путем изменения активационных функций нейронов при сохранении существующих связей или путем перестановки нейрона со всеми его связями. Причем выбор того или иного способа порождения нового решения может определяться значением целевой функции, усредненной по популяции. Как правило, на начальных этапах работы алгоритма используются перестановки нейрона со всеми его связями, на завершающих — перестановки связей между нейронами.

После выполнения очередных операций кроссовера и мутации в структуре сети могут появляться нейроны с разорванными связями по всем входам или выходам (значения синаптических весов всех входных или выходных связей равны 0). В таких случаях фиксируется факт отмирания (выключения) нейронов, и происходит удаление их из сети вместе со всеми связями. Такие структуры являются предпочтительными, так как способствуют минимизации числа элементов нейронной сети, представляющей некоторое решение поставленной задачи.

В ходе эволюции персептрона избыточной структуры последовательно получают все допустимые в ее рамках эквивалентные варианты сетей, реализующие исходную таблицу истинности (для них значение целевой функции  $e$  должно быть нулевым). Из них для дальнейшего рассмотрения оставляется только одна с учетом критериев отбора, обусловленных дополнительными требованиями к топологии сети. Например, наряду с критерием обеспечения требуемой функциональности устройства, к нему могут

также предъявляться требования по критериям аппаратной сложности и быстродействию. С точки зрения быстродействия предпочтение отдается схемам с минимальным числом ступеней преобразования входных сигналов (схемам минимальной ярусности, представляемым нейронными сетями с минимальным количеством слоев). Аппаратно более простыми, как правило, являются многоярусные схемы.

Среди эквивалентных вариантов сетей возможно присутствие идентичных структур, построенных на различных наборах нейроэлементов (число таких структур напрямую зависит от степени избыточности исходного персептрона). Из этих структур для дальнейшего рассмотрения остается одна (любая).

### Пример синтеза схемы одноразрядного комбинационного сумматора

В качестве одного из примеров, иллюстрирующего возможности нейросетевого подхода к синтезу комбинационных схем, рассмотрим синтез схемы одноразрядного комбинационного сумматора.

Функционирование данной схемы определяется таблицей истинности, представленной в табл. 1.

$x_1$	$x_2$	$x_3$	$v_1$	$v_2$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Таблица 1. Таблица истинности одноразрядного комбинационного сумматора

В табл. 1 приняты следующие обозначения:  $x_1, x_2$  — двоичные сигналы складываемых чисел;  $x_3$  — сигнал переноса из соседнего младшего разряда;  $y_1$  — значение сигнала переноса в соседний старший разряд;  $y_2$  — значение разрядной суммы.

Одна из структурных схем сумматора, используемых на практике, реализована в базисе {И, ИЛИ, НЕ} и имеет вид, представленный на рис. 6. Схема является трехуровневой и содержит 13 цифровых элементов.

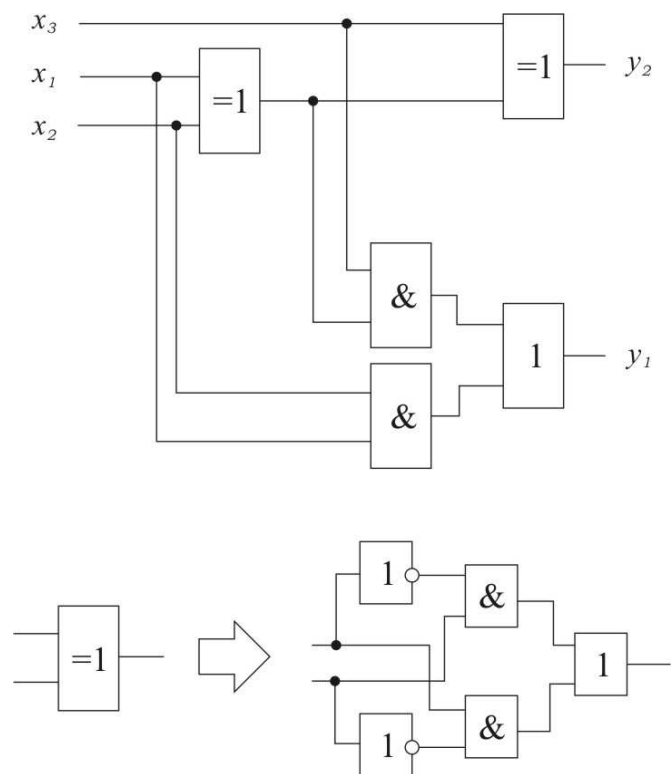


Рис. 6. Структурная схема сумматора, реализованная в базисе {И, ИЛИ, НЕ}.

В качестве исходной нейронной сети для синтеза одноразрядного комбинационного сумматора с использованием предложенного подхода был выбран четырехслойный персептрон структуры  $NN-3-4-3-3-2$ . Сеть имеет три входа для аргументов реализуемых логических функций:  $x_1, x_2$  — входы складываемых чисел;  $x_3$  — вход переноса из соседнего младшего разряда и два выхода:  $y_1$  — выход переноса в соседний старший разряд;  $y_2$  — выход разрядной суммы. Общая структура сети представлена на рис. 7.

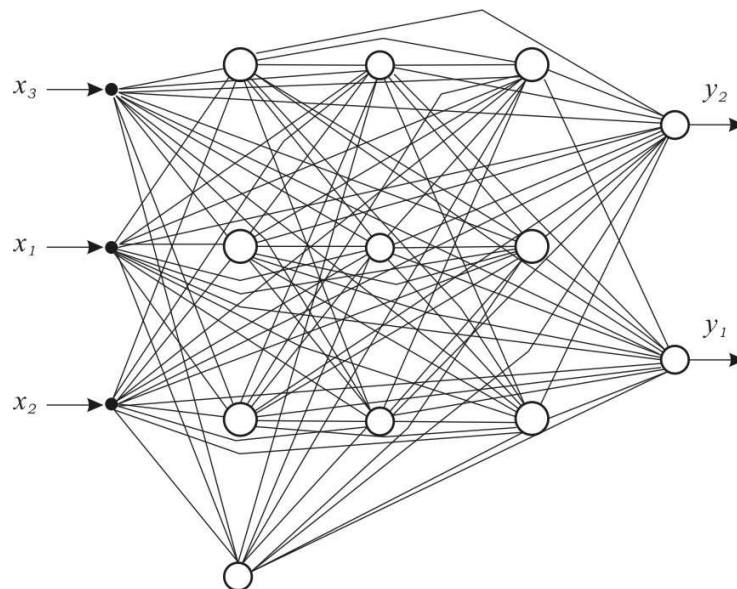


Рис. 7. Исходная структура сети для синтеза схемы одноразрядного сумматора.

Вид активационных функций нейронов, множество значений синаптических весов связей и сигналов, распространяющихся в сети, определяется выбором логического базиса  $\{И, ИЛИ, НЕ\}$ .

В качестве метода обучения сети был использован эволюционно-генетический алгоритм [4].

Для описания эволюционно-генетической модели нейронной сети применена схема непосредственного кодирования, специфицирующая генными кодами каждую синаптическую связь исходно избыточной нейронной сети (0 означает разрыв связи, 1 — наличие непосредственной связи, 2 — связь через инвертор) и вид активационной функции каждого нейрона (2 отмечает нейрон И, 1 — нейрон ИЛИ, 0 — отсутствие нейрона). Пример генетического кода особи эволюционирующей популяции представлен на рис. 8 в виде хромосомы с двумя составляющими L и N, где L — составляющая, кодирующая межнейронные связи, N — составляющая, кодирующая типы активационных функций нейронов.

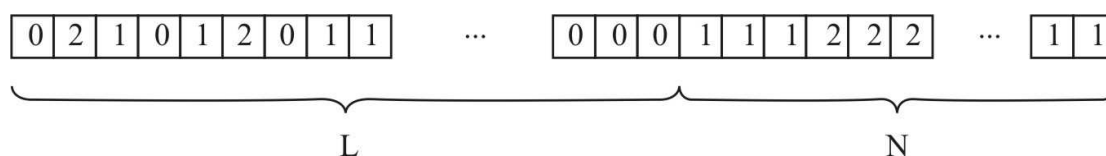


Рис. 8. Структура хромосомы, кодирующей нейронную сеть.

Таким образом, первые L-составляющая ставит в соответствие каждому нейрону участки хромосомы, определяющие его связи со всеми нейронами в других слоях. N-составляющая фиксирует типы активационных функций для каждого нейрона скрытых слоев и выходного слоя в порядке возрастания их номеров в сети сверху вниз и слева направо. Учитывая все возможные связи, в том числе и несмежные (через один или несколько слоев), L-составляющая хромосомы содержит 89 генов. N-составляющая включает 12 генов. Общее число генов в хромосоме  $D = 101$ .

Значение функции приспособленности особи определяется величиной обратной хеммингову расстоянию между фактическим значения  $\bar{y}^q$  на выходе сети с заданным таблицей истинности значением  $\bar{Y}^q$ . Для синтезируемой схемы минимальное значение функции приспособленности соответствует хеммингову расстоянию, равному 8.

Для решения задачи использованы следующие генетические операторы: двухточечная мутация, двухточечный или четырехточечный кроссовер. Выбор вида кроссовера осуществляется на этапе формирования настроечных параметров алгоритма. В ходе работы алгоритма вид кроссовера может меняться, например, в случае возникновения стагнации лучшей функции приспособленности в популяции.

Схема скрещивания представляет собой панмиксию генотипов. Система селекции — ранговая с детерминированным выбором. Репродукционное множество составляют мутанты и потомки.

Условий остановки работы алгоритма несколько. Например, совпадение номера текущего поколения с числом заданных итераций. В этом случае глобальный минимум целевой функции ошибки  $e$  может не достигаться. Однако при этом имеется возможность исследования промежуточных топологий сети и дальнейшей ее модификации. Другим условием остановки является минимальное (нулевое) значение функции приспособленности.

Размер начальной популяции  $|P_0|$  выбирался согласно двум критериям: обеспечения покрытия области экстремума целевой функции и допустимого времени работы алгоритма. Экспериментально установленный для данного примера размер популяции составляет величину не менее 250 особей.

Структура алгоритма обучения сети выглядит следующим образом:

1. Инициализируется начальная популяция  $P_0$ . Для этого выполняется следующая последовательность действий: устанавливается номер текущего поколения  $t = 0$ ; генерируется случайным образом хромосомный набор из  $|P_0|$  строковых кодировок фиксированной длины  $D = 101$ , в котором хэммингово расстояние между любой парой кодировок не равно нулю.
2. Воспроизводство потомков с наследственными генетическими свойствами родителей производится следующим образом:
  - 2.1. Согласно схеме скрещивания, из текущей популяции  $P_t$  выбираются кодировки двух родителей, образующих «брачную пару».
  - 2.2. Оператором кроссовера для выбранной «брачной пары» генерируются две кодировки потомков. При двухточечном кроссовере случайным образом выбираются две точки разрыва  $r_1, r_2 \in \{1, 2, \dots, D-1\}$ ,  $r_1 < r_2$ . Для четырехточечного кроссовера случайным образом выбираются только две точки разрыва  $r_3, r_4$ , принадлежащие N-составляющей хромосомы, которые однозначно определяют (согласно схеме кодирования) две другие точки  $r_1, r_2$ , принадлежащие L-составляющей.
  - 2.3. С помощью функции приспособленности оценивается каждая кодировка потомков.
  - 2.4. Все операции, начиная с пункта 2.1, повторяются до тех пор, пока не будет рассмотрено число брачных пар, равное  $\frac{1}{2}|P_0|$ .
3. Создание мутантов производится следующим образом:
  - 3.1. Из числа родителей случайным образом выбирается кодировка.
  - 3.2. Для выбранной кодировки при помощи оператора мутации генерируется мутант. Например, для родительской кодировки  $\chi = (\chi_1, \dots, \chi_D)$  потомок-мутант получается в результате замены

значений аллелей в двух случайно выбранных генах хромосомы на одно из альтернативных значений. Альтернативные значение  $i$ -того аллеля мутанта  $\chi_i$  выбирается с равной вероятностью из множества  $\{\{0,1,2\} \setminus \chi_r\}$ .

3.3. С помощью функции приспособленности оценивается кодировка мутанта.

3.4. Все операции, начиная с пункта 3.1, повторяются до тех пор, пока не будет получено число мутантов равно  $|P_0|$ .

4. Производится смена текущей популяции  $P_t$  новой популяцией  $P_{t+1}$ .

4.1. Формируется репродукционное множество кодировок  $R^t$ , содержащее потомков и мутантов.

4.2. Из репродукционного множества с помощью оператора селекции выбираются  $|P_0|$  строковых кодировок.

*1 этап выбора.* Для популяции  $P_{t+1}$  определяется ожидаемое число копий  $n_i^t$  каждой  $i$ -ой кодировки  $\chi_i \in R^t$ , где  $i = 1, \dots, \nu$ ,  $\nu = |R^t|$

$$n_i^t = n^- + (n^+ - n^-) \cdot \frac{r_i^t - 1}{\nu - 1},$$

здесь  $r_i^t = i$  – ранг  $i$ -ой кодировки, определяемый после сортировки множества  $R^t$  в порядке монотонного неубывания значений функции приспособленности;  $n^+$  выбирается случайным образом из интервала  $(1, 2]$  и представляет собой верхнюю границу ожидаемого числа копий для «наилучшей» кодировки, имеющей ранг  $r_i^t = \nu$ ;  $n^- = 2 - n^+$  — нижняя граница ожидаемого числа копий для «наихудшей» особи, имеющей ранг  $r_i^t = 1$ .

*2 этап выбора.* Генерируется новая популяция  $P_{t+1}$  путем указания конкретных кодировок  $\chi_i \in R^t$ , копируемых в  $P_{t+1}$  на основании соответствующего им ожидаемого числа копий  $n_i^t$ . Для этого используется детерминированный выбор, состоящий из двух фаз.

*Целочисленная фаза.*

1) В популяцию  $P_{t+1}$  репродуцируется  $[n_i^t]$  копий каждой особи из репродукционного множества  $R^t$ .

2) Определяется число недостающих копий  $\bar{v} = |P_t| - \sum_{i=1}^v [n_i^t]$ ,  
необходимых для полного заполнения популяции  $P_{t+1}$

3) Если  $\bar{v} \leq 0$ , то работа оператора селекции завершается ( $\bar{v} < 0$  лишние особи из  $P_{t+1}$  удаляются). В противном случае ( $\bar{v} > 0$ ) выполняется дробная фаза.

*Дробная фаза.*

1) Решения репродукционного множества  $R^t$  упорядочиваются в порядке уменьшения дробной части  $n_i^t$ .

2) В популяцию  $P_{t+1}$  отбираются первые  $\bar{v}$  копий упорядоченных на 1-ом шаге.

5. Сменяется номер текущего поколения  $t = t+1$ , проверяется условие остановки алгоритма. Если условие выполнено, то производится выбор хромосомы с наименьшим значением функции приспособленности в качестве решения поставленной задачи.

Для предотвращения сходимости к локально-оптимальному решению, алгоритм использует оператор «генетического взрыва», суть которого состоит в следующем. Если в текущей популяции образуются группы индивидуумов с тождественным хромосомным набором и со значением функции приспособленности отличным от требуемого в течении некоторого количества итераций алгоритма, то все индивидуумы этой группы, за исключением лучшей особи, уничтожаются и вместо них случайно генерируются новые особи. Это рассматривается как генерация популяции следующего поколения.

На рис. 9 приведена одна из полученных в процессе машинного эксперимента структур нейронных сетей, повторяющая классическое схемное решение для сумматора, представленное на рис. 6.

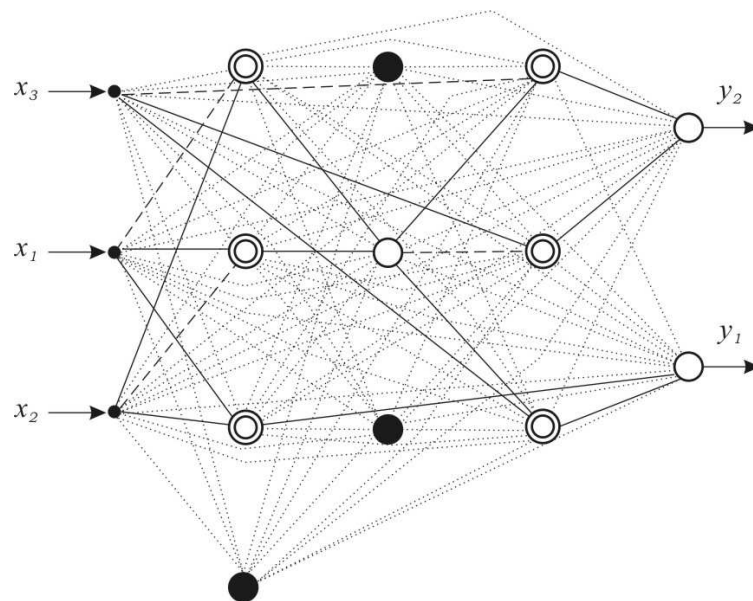


Рис. 9. Нейросетевое представление классического одноразрядного сумматора.

На рис. 9 двойными окружностями помечены нейроны, моделирующие логические элементы И, одинарными — элементы ИЛИ, черными кругами — «мертвые» нейроны, у которых отсутствуют все входные или выходные связи. Сплошными линиями отмечены непосредственные связи между нейронами, пунктирными — инверсные связи, точечными линиями — отсутствующие связи.

Генетический код для нейронной сети, показанной на рис. 9, имеет вид, представленный на рис. 10.

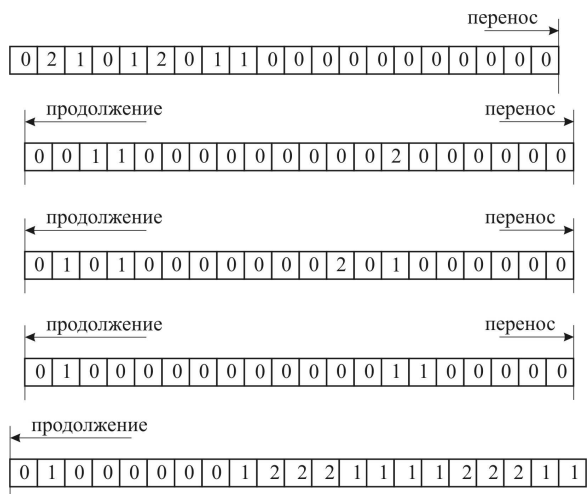


Рис. 10. Генетический код нейросетевой модели классического одноразрядного сумматора.

На рис. 11 приведена еще одна структура нейронной сети, полученная в результате машинного эксперимента.

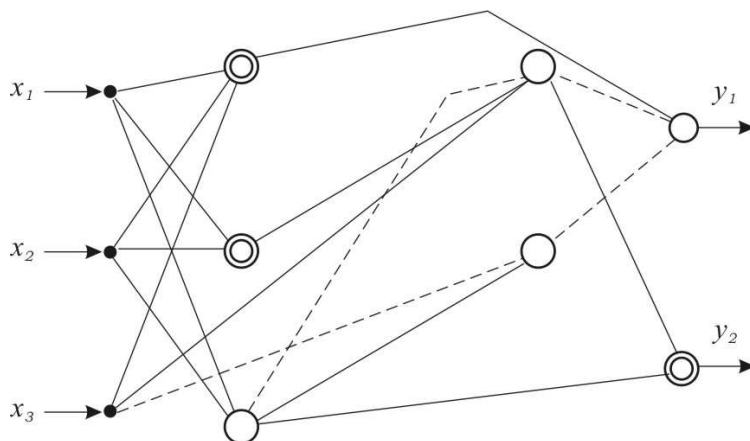


Рис. 11. Нейросетевое представление другого варианта одноразрядного сумматора.

Это решение представляется более экономичным по числу логических элементов и может оказаться предпочтительным для реализации.

На рис. 12 представлен схемный эквивалент нейронной сети, показанной на рис. 11.



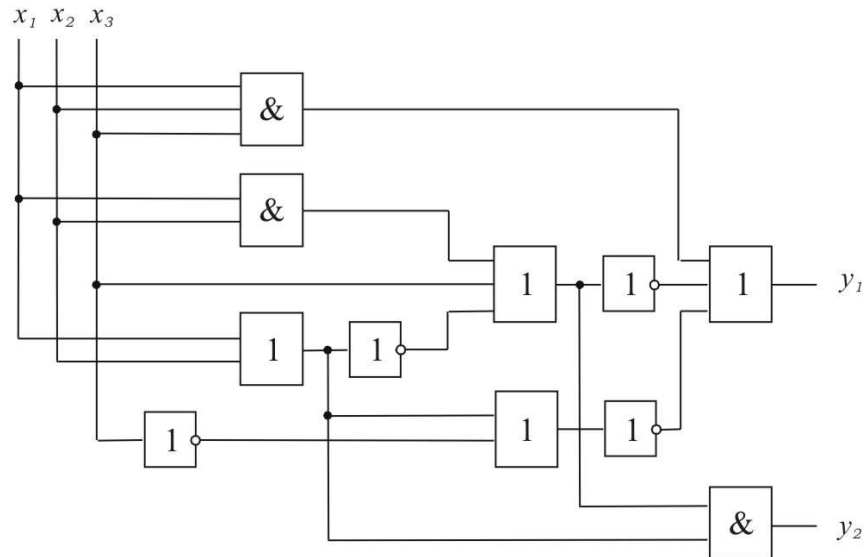


Рис. 12. Структурная схема одноразрядного сумматора, соответствующая нейронной структуре рис.11.

Нейросетевое решение задачи синтеза схемы одноразрядного сумматора в базисе {И-НЕ} представлено на рис. 13. На рис. 14 представлена хромосома, соответствующая данному решению.

Для описания эволюционно-генетической модели нейронной сети использована схема непосредственного кодирования, специфицирующая генными кодами каждую синаптическую связь исходной избыточной нейронной сети (0 означает разрыв связи, 1 — наличие непосредственной связи) и вид активационной функции каждого нейрона (1 — нейрон И-НЕ, 0 — отсутствие нейрона).

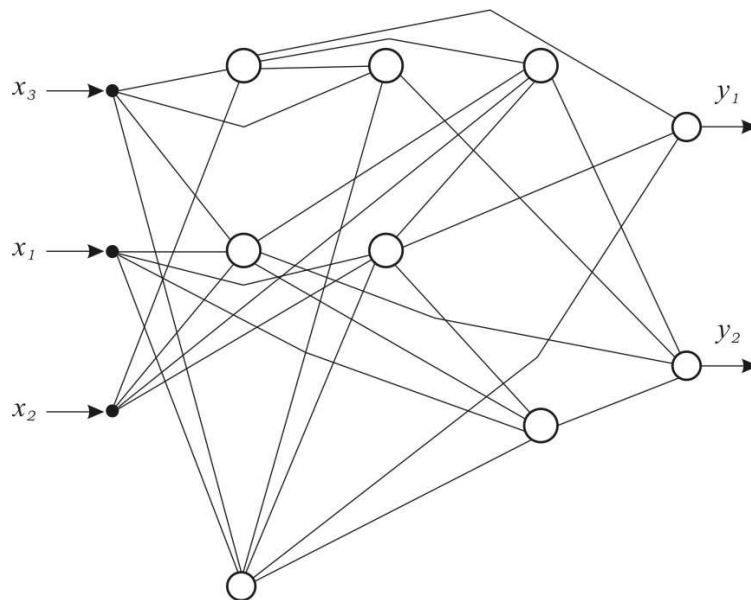


Рис. 13. Нейросетевое решение задачи синтеза схемы одноразрядного сумматора в базисе И-НЕ.

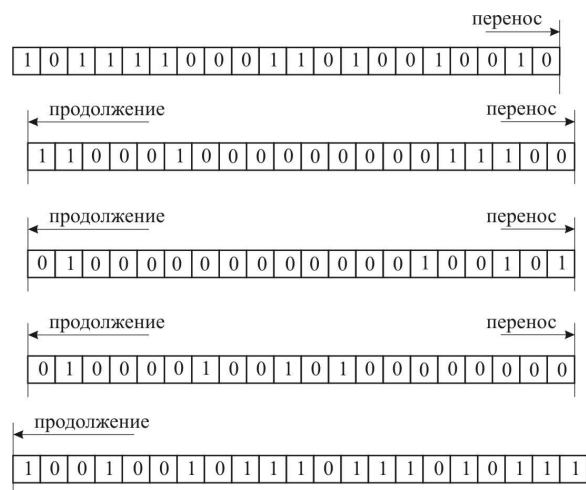


Рис. 14. Структура хромосомы нейронной сети представленной на рис. 13.

Данное решение отличается экономичностью с точки зрения количества цифровых компонентов по сравнению со структурами, представленной на рис. 9 и 11. В первую очередь это является следствием выбора базиса  $\{И-НЕ\}$ , содержащего минимальное количество базисных функций.

## Литература

- Глушков В.М. Синтез цифровых автоматов. – М.: Физматгиз, 1962. – 476 с.
1. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. 2-е изд. – М.: Горячая линия – Телеком, 2002. – 382 с.
  2. Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев А.С. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. – Х.:ОСНОВА, 1997. – 112 с.
  3. Батищев Д.И., Костюков В.Е., Старостин Н.В., Смирнов А.И. Популяционно-генетический подход к решению задач покрытия множества: Учебное пособие. – Нижний Новгород: Изд-во ННГУ им.Н.И.Лобачевского, 2004. – 152 с.