

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
МИНИСТЕРСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ПО АТОМНОЙ ЭНЕРГИИ
МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ, НАУКИ И ТЕХНОЛОГИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ
РОССИЙСКАЯ АССОЦИАЦИЯ НЕЙРОИНФОРМАТИКИ
МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

НАУЧНАЯ СЕССИЯ МИФИ–2004

НЕЙРОИНФОРМАТИКА–2004

**VI ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ**

**ЛЕКЦИИ
ПО НЕЙРОИНФОРМАТИКЕ**

Часть 1

По материалам Школы-семинара
«Современные проблемы нейроинформатики»

Москва 2004

УДК 004.032.26 (06)

ББК 32.818я5

М82

НАУЧНАЯ СЕССИЯ МИФИ-2004. VI ВСЕРОССИЙСКАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ «НЕЙРОИНФОРМАТИКА-2004»: ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ. Часть 1. – М.: МИФИ, 2004. – 199 с.

В книге публикуются тексты лекций, прочитанных на Школе-семинаре «Современные проблемы нейроинформатики», проходившей 28–30 января 2004 года в МИФИ в рамках VI Всероссийской научно-технической конференции «Нейроинформатика-2004».

Материалы лекций связаны с рядом проблем, актуальных для современного этапа развития нейроинформатики, включая ее взаимодействие с другими научно-техническими областями.

Ответственный редактор

Ю. В. Тюменцев, кандидат технических наук

ISBN 5-7262-0526-X

© *Московский инженерно-физический институт
(государственный университет), 2004*

Содержание

А. Ю. Дорогов. Быстрые нейронные сети: Проектирование, настройка, приложения	69
Введение	70
Структурный анализ алгоритмов БПФ	72
Структурный синтез быстрых нейронных сетей	76
Формальный язык регулярных сетей	76
Грамматика предложений и морфология сети	78
Семантическая интерпретация предложений	80
Топологическое проектирование быстрых нейронных сетей	81
Топологии нейронных слоев	82
Графическая интерпретация топологий	83
Граничные условия и топологические матрицы	86
Алгоритм обработки данных в БНС	89
Слабосвязанные сети	91
Системные характеристики БНС	94
Вычислительная эффективность	94
Пластичность	94
Способность к обучению	95
Перестраиваемые спектральные преобразования	98
Настройка перестраиваемых преобразований	99
Настройка на базис Уолша	100
Настройка на базис Фурье	103
Быстрое вейвлет-преобразование	104
Нейросетевая аппроксимация регулярных фракталов	106
Аналитическая форма регулярного фрактала	107
Дискретная аппроксимация фракталов	108
Фрактальная фильтрация сигналов	111
Фильтрация дискретных сигналов	112
Типы фрактальных фильтров	113
Фильтрация непрерывных сигналов	115

Приспособленные быстрые преобразования	116
Алгоритм приспособления	116
Приспособленные преобразования в нечетком пространстве	120
Спектральные приспособленные преобразования	122
Быстрые нейронные сети в квантовых вычислениях	126
Многомерные БНС	131
Литература	133

А. Ю. ДОРОГОВ

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ»

E-Mail: dorv@lens.spb.ru

**БЫСТРЫЕ НЕЙРОННЫЕ СЕТИ: ПРОЕКТИРОВАНИЕ,
НАСТРОЙКА, ПРИЛОЖЕНИЯ¹**

Аннотация

Обсуждается парадигма быстрых нейронных сетей (БНС). Представлены лингвистические методы проектирования структуры и топологии БНС. На уровне морфологии установлены общие принципы построения быстрых преобразований. Найдены количественные оценки быстродействия и способности нейронной сети к обучению. Рассмотрены методы настройки БНС при реализации спектральных преобразований, регулярных фракталов, оптимальных фильтров. Исследованы вопросы применения БНС в квантовых вычислениях. Рассмотрены методы построения многомерных БНС.

A. Yu. DOROGOV

Saint-Petersburg state electrotechnical university "LETI"

E-Mail: dorv@lens.spb.ru

**FAST NEURAL NETWORKS: DESIGN, TUNING AND
APPLICATIONS**

Abstract

A paradigm of fast neural networks (FNN) is discussed. Formal linguistic methods are presented to design structure and topology for FNN. A general principle of fast transformations have been founded on morphological level. Some quantitative estimations are obtained for processing speed and learning ability of FNN. Tuning techniques are described to implement FNN-based spectral transformations, regular fractals and optimum filters. Capabilities of FNN for quantum computations are investigated. A design approach is suggested to build multi-dimensional FNN.

¹Работа выполнена при поддержке гранта Минобразования РФ.

Введение

Хорошо известно, какую огромную роль в обработке сигналов сыграло открытие алгоритма *быстрого преобразования Фурье* (БПФ). Использование БПФ позволило кардинальным образом уменьшить количество вычислительных операций при выполнении спектральных преобразований, что значительно расширило сферу использования спектрального анализа для обработки данных. Реализация БПФ в технологии больших интегральных схем приводит к существенному уменьшению площади кристаллов спектральных анализаторов и, соответственно, к радикальному уменьшению их энергопотребления.

Для нейронных сетей задача сокращения числа вычислительных операций не менее актуальна. При больших размерностях обрабатываемых данных нейронные сети с многослойной структурой требуют значительного объема вычислительных операций как при обработке данных, так и при обучении сетей. Это ограничивает применение больших нейронных сетей данного класса в системах реального времени и существенно усложняет аппаратную реализацию.

Алгоритмы БПФ имеют выраженную многослойную структуру, подобную структуре многослойных перцептронов, поэтому напрашивается естественное решение — использовать идеологию БПФ для построения нейронных сетей [1]. Для этого достаточно в операциях «бабочка» БПФ заменить коэффициенты преобразования перестраиваемыми синаптическими весами и добавить нелинейные функции активации. Следует отметить, что первый шаг в этом направлении был сделан достаточно давно. Еще в исторической работе Гуда [2] (1958 г.) было приведено аналитическое описание быстрых алгоритмов для обобщенных спектральных преобразований. Обобщенное спектральное преобразование с позиций сегодняшнего дня можно рассматривать как нейронную сеть с линейными функциями активации. В последующие годы тема обобщенных спектральных преобразований развивалась в работах Г. Эндрюса, А. И. Солодовникова, В. Г. Лабунца и других авторов [3–6].

Алгоритм БПФ обладает двумя системными ограничениями: во-первых, размерность вектора обрабатываемых данных для всех слоев должна быть одинакова и, во-вторых, значение этой размерности должно быть составным числом, т. е. разлагаться в произведение целых множителей. Первое ограничение обусловлено ортогональностью матрицы БПФ, а второе — регулярностью алгоритма что, по-видимому, является неизбежной платой

за быстрое действие. Парадигма быстрых нейронных сетей наследует регулярность алгоритма БПФ, но расширяет ее возможности за счет отказа от ортогональности и жесткой схемы топологической реализации. Для структурно-регулярной сети может быть построена как регулярная, так и нерегулярная топология. На выбор типа топологии могут влиять два обстоятельства: структура данных и простота технической реализации. Для больших и сверхбольших нейронных сетей (потенциальная область применения БНС) простота технической реализации является определяющим условием. Поэтому регулярность топологии далее будет рассматриваться как составная компонента парадигмы БНС.

БНС представляют собой вариант многослойных сетей, поэтому для их обучения могут быть использованы градиентные методы типа Error Backpropagation, однако структурные особенности БНС позволяют упростить и процедуру обучения. Частным случаем БНС являются быстрые перестраиваемые линейные преобразования. От нейронных сетей они отличаются линейными функциями активации и отсутствием смещений. Ортогональные перестраиваемые преобразования с различной топологией традиционно используются для построения спектральных преобразований. В терминах перестраиваемых преобразований процедура обучения обычно называется настройкой. Типичной задачей для перестраиваемого преобразования является настройка на заданную систему базисных функций, например базис Фурье, Адамара–Уолша и др. Классические системы базисных функций, как правило, имеют аналитическую форму, поэтому настройка преобразования также выполняется в аналитическом виде. Поскольку структура БНС обладает фрактальными свойствами, то регулярные фрактальные последовательности могут быть реализованы как многослойные БНС. Фрактальность порождает и особые методы настройки БНС для реализации адаптивных фильтров. Идентичность структуры БНС со структурой тензорных произведений векторных пространств позволяет использовать БНС для построения алгоритмов квантовых вычислений. Парадигма БНС допускает многомерное обобщение, что наряду с высоким быстрым действием способствует использованию БНС для построения классификаторов зрительных сцен. Перечисленные выше вопросы составляют основное содержание настоящей лекции. Цель лекции состоит в том, чтобы дать представление о методах построения БНС, оценить их характеристики и показать потенциальные области применения.

Структурный анализ алгоритмов БПФ

Набор алгоритмов, называемых алгоритмами быстрого преобразования Фурье вошел в практику спектрального анализа в 60-х годах, начиная с работ Кули-Тьюки [7]. Существует множество различных схем построения алгоритмов БПФ, как правило, они основаны на способе разбиения входного вектора на подвекторы. Следуя [8], рассмотрим в качестве примера схему «с прореживанием по частоте». Для вектора $X = \{x(i)\}$, $0 \leq i \leq N - 1$ прямое дискретное преобразование Фурье (ДПФ) определяется выражением:

$$y(m) = \sum_{i=0}^{N-1} x(i)e^{-j\frac{2\pi}{N}im},$$

где $m = 0, 1, 2, \dots, N - 1$ — номер гармоники. Величину $\omega = e^{-j\frac{2\pi}{N}}$ обычно называют поворачивающим множителем. Число комплексных операций умножения при выполнении ДПФ в общем случае равно N^2 .

Для алгоритмов БПФ, как правило, выбирают значение $N = 2^r$, где r — целое число. В схеме «с прореживанием по частоте» вектор $X = \{x(i)\}$ разбивается на два подвектора по правилу:

$$\begin{aligned} x_1(i) &= x(i), & x_2(i) &= x(i + N/2), \\ i &= 0, 1, 2, \dots, (N/2) - 1. \end{aligned}$$

Этот прием позволяет свести вычисление N -точечного преобразования к вычислению двух $N/2$ -точечных преобразований, при этом число комплексных операций умножения будет значительно меньше N^2 . Математические детали этой идеи подробно представлены в главе 6 работы [8]. На рис. 1 в графической форме показан пример перехода от восьмиточечного ДПФ к двум четырехточечным ДПФ при «прореживании по частоте».

Графический образ в виде пары перекрещивающихся линий с кружком посередине (напоминающий «бабочку») обозначает базовую операцию над парой координат входного вектора. Для алгоритма БПФ «с прореживанием по частоте» базовая операция представляется унитарной матрицей вида:

$$\begin{pmatrix} 1 & \omega^i \\ 1 & -\omega^i \end{pmatrix}.$$

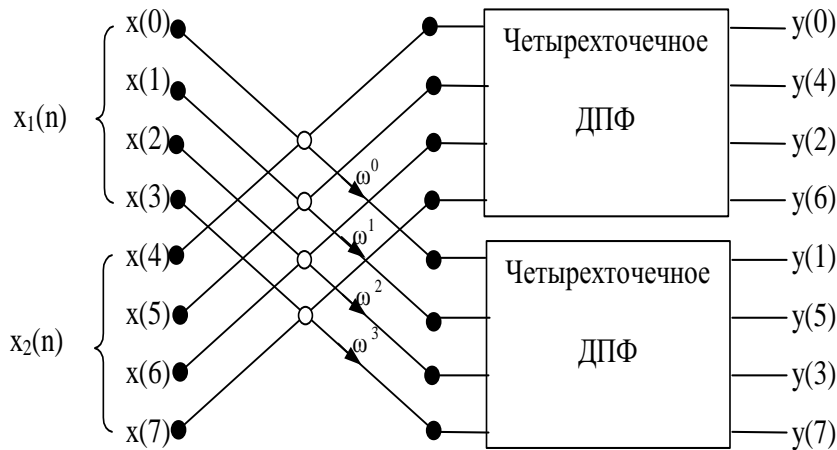


Рис. 1. Переход от восьмиточечного ДПФ к двум четырехточечным

В общем случае базовая операция определяется матрицей W размерности $p \times p$:

$$W = \begin{pmatrix} w_{00} & w_{01} & \cdots & w_{0,p-1} \\ w_{10} & w_{11} & \cdots & w_{1,p-1} \\ \cdots & \cdots & \cdots & \cdots \\ w_{p-1,0} & w_{p-1,1} & \cdots & w_{p-1,p-1} \end{pmatrix}$$

(здесь и далее принято, что нумерация строк и столбцов в матрицах начинается с нулевого индекса). Для базовой операции «бабочка» значение p равно двум, а для четырехточечного преобразования ДПФ будет $p = 4$.

Если сопоставить каждой базовой операции вершину графа, а дугам — операторы связи между базовыми операциями, то получим структурную модель алгоритма БПФ. На рис. 2 показана структурная модель для алгоритма, изображенного на рис. 1. На графе структурной модели приведены также размеры матриц базовых операций и ранги операторов связи (для алгоритмов БПФ ранги связей всегда равны единице). По структурной модели нетрудно подсчитать вычислительные затраты связанные с выполнением БПФ, для этого достаточно просуммировать число операций умножения по всем базовым операциям, например для структурной модели, показанной на рис. 2 получим:

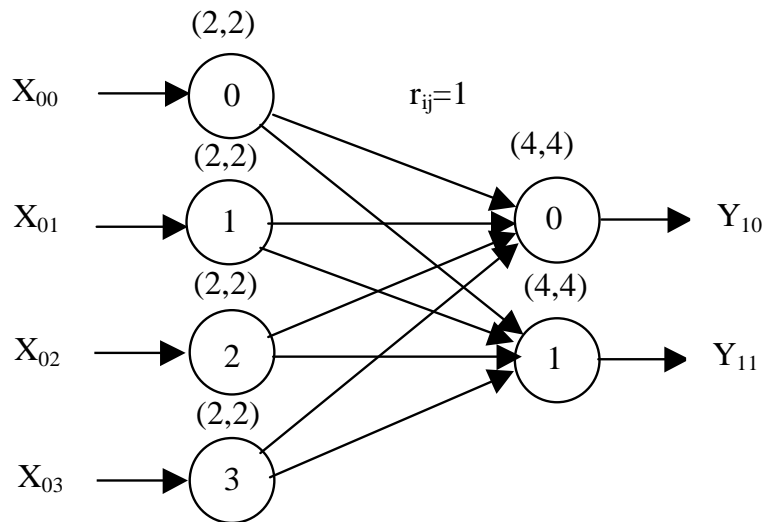


Рис. 2. Структурная модель БПФ первого шага

$$Z = 2 \times 2 + 2 \times 2 + 2 \times 2 + 2 \times 2 + 4 \times 4 + 4 \times 4 = 48.$$

Заметим для сравнения, что прямое восьмиточечное ДПФ имеет $8 \times 8 = 64$ операции умножения. Каждое дискретное преобразование размерности $N/2$ в свою очередь может быть сведено к двум $N/4$ точечным преобразованиям. На рис. 3 показан полный граф восьмиточечного БПФ с «прореживанием по частоте».

Для данного алгоритма спектральные коэффициенты на выходе преобразования оказываются переставленными в двоично-инверсном порядке по отношению к частоте гармоники. На рис. 4 приведена структурная модель для полного восьмиточечного БПФ. В общем случае размерность преобразования может быть составным числом: $N = n_0 n_1 \dots n_{\varkappa-1}$, где n_λ — целые числа. В этом случае структурная модель состоит из \varkappa слоев с размерностью базовых операций (n_λ, n_λ) в слое λ и числом вершин в слое равным $k_\lambda = N/n_\lambda$.

Обозначим через i номер вершины во входном слое, а через j — номер вершины в выходном слое и представим эти числа в позиционной

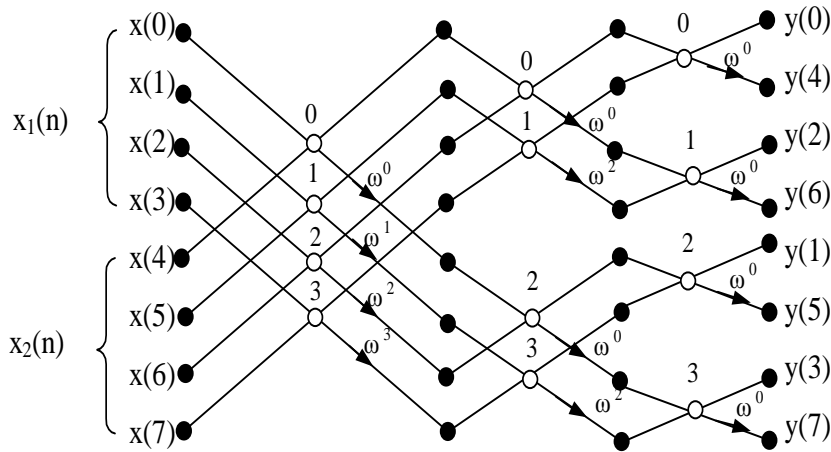


Рис. 3. Полный граф восьмиточечного БПФ с «прореживанием по частоте»

многоосновной системе счисления [9] следующим образом:

$$i = \langle i_{\lambda-2} i_{\lambda-3} \dots i_0 \rangle = i_{\lambda-2} n_{\lambda-3} n_{\lambda-4} \dots n_0 + \dots + i_1 n_0 + i_0,$$

$$j = \langle j_1 j_2 \dots j_{\lambda-1} \rangle = j_1 n_2 n_3 \dots n_{\lambda-1} + \dots + j_{\lambda-1} n_{\lambda-2} + j_{\lambda-1},$$

где $i_\lambda, j_\lambda \in [0, 1, \dots, (n_\lambda - 1)]$ – разрядные числа. Непосредственной проверкой можно убедиться, что нумерация вершин в графе структурной модели будет определяться выражением:

$$i^\lambda = \langle j_1 j_2 \dots j_\lambda i_{\lambda-2} i_{\lambda-3} \dots i_0 \rangle.$$

Из приведенного выше анализа можно сделать следующие выводы:

- На структурном уровне быстрый алгоритм представляется многослойной модульной сетью, где каждому модулю отвечает базовая операция.
- Размерность быстрого преобразования всегда выражается составным числом, где числовые множителями определяют размерности базовых операций.
- Базовые операции выражаются квадратными матрицами, которые в пределах слоя имеют одинаковый порядок.

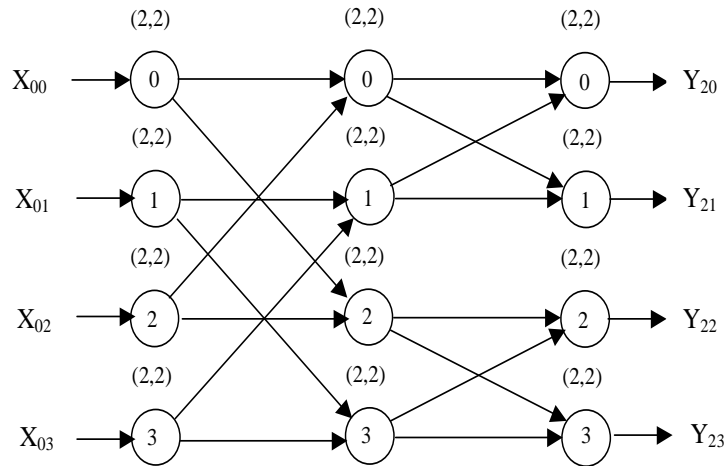


Рис. 4. Структурная модель восьмиточечного БПФ

- Ранги связей между базовыми операциями всегда равны единице.
- Алгоритм БПФ имеет регулярную структуру и регулярную топологию.

Структурный синтез быстрых нейронных сетей

В данном разделе рассматриваются лингвистические модели регулярных БНС на уровне структуры и морфологии. Лингвистическая модель основана на концепции формального языка. В контексте языка структура сети интерпретируется как графическая семантика допустимых предложений.

Формальный язык регулярных сетей

Любой формальный язык является средством для выражения инвариантных свойств предметной области знаний. Для построения языка вводится алфавит, грамматика и устанавливается семантика слов и предложений. Практическая ценность формального языка определяется его семантической интерпретацией, поэтому построение языка регулярных сетей целесообразно начать с его семантики.

Семантика слов. Пусть $\langle z_{n-1}z_{n-2} \dots z_0 \rangle$ — некоторое допустимое слово формального языка. Поставим в соответствие каждой букве z_i целое положительное число p_i , и будем рассматривать букву как символическое обозначение переменной принимающей целочисленные значения из интервала $Z_i = [0, 1, \dots, p_i - 1]$. Слово языка будем интерпретировать как целочисленную функцию, заданную правилом:

$$\langle z_{n-1}z_{n-2} \dots z_0 \rangle = z_{n-1}p_{n-2}p_{n-3} \dots p_0 + z_{n-2}p_{n-3}p_{n-4} \dots p_0 + \dots + z_1p_0 + z_0 \quad (1)$$

Порядок следования аргументов для данной функции имеет существенное значение, поэтому каждое слово языка представляет собой кортеж, т. е. множество, на котором зафиксирован линейный порядок. Для сокращенного обозначения кортежа все элементы которого имеют общее родовое имя (в данном случае — z), будем использовать следующую форму: $\langle z \rangle_J$, где J — отношение линейного порядка. Для определенности будем полагать, что собственные позиции кортежа всегда упорядочены по возрастанию слева направо, начиная с нулевой. В дальнейшем интерпретирующую функцию будем называть *конвергенцией*. Нетрудно видеть, что функция конвергенции представляет собой правило перехода от позиционного представления числа в многоосновной системе счисления к его количественному значению. В этом контексте переменную z_i уместно назвать разрядом, а константу p_i основанием разряда.

Алфавит языка и грамматика слов. Алфавитом языка будем считать кортеж неповторяющихся символов $A = \langle z_{i_0}z_{i_1} \dots z_{i_x} \rangle$. Отношение линейного порядка на алфавитном множестве назовем *канон* языка и обозначим через K . В сокращенной форме кортеж алфавита, состоящий из букв с общим родовым именем, записывается в виде: $A = \langle z \rangle_K$. Канон языка может быть задан таблицей:

$$K = \begin{pmatrix} z_{i_0} & z_{i_1} & \dots & z_{i_x} \\ 0 & 1 & \dots & \varkappa \end{pmatrix}.$$

Нижняя строка таблицы определяет порядок символов в алфавите. Далее будем использовать алфавит с двумя различными типами букв. Каждый тип буквы определяется его родовым именем (например, u, v) в этом случае для сокращенной записи алфавита будем использовать обозначения вида:

$$A = \langle \langle u \rangle \oplus \langle v \rangle \rangle_K.$$

Пусть алфавит A состоит из \varkappa букв. Допустимым словом языка будем считать любой кортеж длиной $n \leq \varkappa$, состоящий из неповторяющихся букв алфавита. Если последовательность букв в слове соответствует канону, то слово будем называть *каноническим*.

Операция перестановки. Пусть Q есть множество всевозможных перестановок букв, определенных на кортеже слова $\langle z \rangle_K$, и q — некоторая перестановка принадлежащая этому множеству. Операцию перестановки для слова $\langle z \rangle_K$ будем записывать в виде: $\langle z \rangle_K * q$, например,

$$\langle z_1 z_3 z_0 z_6 \rangle * \begin{pmatrix} 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 \end{pmatrix} = \langle z_6 z_0 z_3 z_1 \rangle.$$

На множестве слов языка определим отношение эквивалентности π следующим образом. Два слова $\langle z \rangle_{J_1}$ и $\langle z \rangle_{J_2}$ одинаковой длины будем считать эквивалентными, если существует перестановка q такая, что

$$\langle z \rangle_{J_1} * q = \langle z \rangle_{J_2}.$$

Отношение эквивалентности разбивает множество слов языка на классы. Понятно, что если $\langle z \rangle_{J_1}$ и $\langle z \rangle_{J_2}$ два различных канонических слова, то они всегда принадлежат разным классам, т. е. канонические слова являются представителями эквивалентных классов. Каждый класс может быть образован всевозможными перестановками букв канонического слова.

Смежные классы. Пусть задан алфавит длиной \varkappa . Рассмотрим множество эквивалентных классов с длиной слов, равной n . Если $n = \varkappa$, то множество состоит из одного класса. Для слов длиной $n = \varkappa - 1$ число возможных классов равно \varkappa . Два класса на множестве слов длиной n назовем *смежными*, если они отличаются только одной буквой. Смежные классы образуют упорядоченные цепочки по подмножествам входящих букв. Длина каждой цепочки не превышает $n + 1$. Цепочки смежных классов длиной $n + 1$ будем называть *максимальными*.

Грамматика предложений и морфология сети

Правила построения предложений (иначе эти правила называются порождающей грамматикой) целиком подчинены задачам предметной области. Порождающая грамматика составляет основу метода синтеза регулярных многослойных графов.

В линейных языках каждое предложение имеет одно начало и один конец. Возможны различные способы кодирования границ предложения. Будем использовать для этой цели вариант родовых имен. Определим алфавит языка как линейно упорядоченный набор неповторяющихся букв с двумя видами родовых имен:

$$A = \langle \langle i \rangle_I \oplus \langle j \rangle_J \rangle_K.$$

Число букв каждого вида будем считать одинаковыми и равными $\varkappa - 1$. Таким образом, длина алфавита равна $2(\varkappa - 1)$. Рассмотрим множество всевозможных слов длиной $(\varkappa - 1)$, из которых будем строить предложения. Как и прежде будем полагать, что на множестве слов операцией перестановки определено отношение эквивалентности π , разбивающее это множество на непересекающиеся классы. Допустимым предложением будем считать упорядоченную последовательность слов, каждое из которых является представителем максимальной цепочки смежных классов. В допустимом предложении представлены все смежные классы цепочки, и каждый класс представлен только одним словом. Длина предложения всегда равна \varkappa . Последовательность слов в предложении определяется правилом вывода. Это правило устанавливает начальное слово предложения и описывает связи между смежными словами. Зафиксируем следующее правило вывода:

- слова в предложении линейно упорядочены по числу букв с родовым именем j ;
- первое слово предложения выбирается из класса $\langle i \rangle_I$ и имеет, поэтому нулевое число вхождений букв с родовым именем j .

Каждая цепочка смежных классов порождает множество однотипных предложений. Среди этого множества существует единственное предложение, составленное из канонических слов, такое предложение назовем *каноническим*. Все остальные предложения для данной цепочки могут быть получены произвольной перестановкой букв в словах канонического предложения. Поэтому достаточно рассматривать порождающую грамматику только для канонических предложений. Пусть числа λ и $\lambda + 1$ определяют порядковые номера двух смежных классов в цепочке. Обозначим через I_λ и J_λ канонически упорядоченные подмножества букв каждого типа для смежного класса λ , а через $\langle \langle i \rangle_{I_\lambda} \oplus \langle j \rangle_{J_\lambda} \rangle$ его каноническое слово. Тогда правило вывода для канонических предложений можно сформулировать

следующим образом:

$$I_\lambda \supset I_{\lambda+1}, \quad J_\lambda \subset J_{\lambda+1}, \\ I_0 = I, \quad I_{\kappa-1} = \emptyset, \quad J_{\kappa-1} = J, \quad J_0 = \emptyset.$$

Построенное правило не связано ни с размерностью сети, ни с ее топологией, ни со структурными характеристиками вершин — это правило морфологического уровня, которое в аксиоматической форме раскрывает внутреннюю сущность быстрых преобразований.

Семантическая интерпретация предложений

Поставим в соответствие каноническому предложению n -слойный сетевой граф. Каждое слово с порядковым номером λ будем интерпретировать как функциональное правило вычисления номера вершины в слое λ , в соответствии с семантикой (1). Вершины двух смежных слоев будем считать связанными дугой, если в поразрядном представлении номеров вершин

$$i^\lambda = \langle \langle i \rangle_{I_\lambda} \oplus \langle j \rangle_{J_\lambda} \rangle, \quad i^{\lambda+1} = \langle \langle i \rangle_{I_{\lambda+1}} \oplus \langle j \rangle_{J_{\lambda+1}} \rangle$$

одноименные разрядные переменные имеют совпадающие значения. Рассмотрим пример построения сетевого графа. Пусть алфавит языка определен кортежем

$$A = \langle i_1 i_2 i_3 j_2 j_1 j_0 \rangle.$$

Нетрудно убедиться, что предложение

$$[\langle i_1 i_2 i_3 \rangle \langle i_1 i_2 j_0 \rangle \langle i_1 j_1 j_0 \rangle \langle j_2 j_1 j_0 \rangle] \quad (2)$$

является каноническим. Будем полагать, что основания разрядных переменных заданы таблицей

$$\begin{pmatrix} A \\ p \end{pmatrix} = \begin{pmatrix} i_1 & i_2 & i_3 & j_2 & j_1 & j_0 \\ 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix}.$$

В соответствии с семантикой (1) слова предложения интерпретируются как следующие алгебраические выражения:

$$\begin{aligned} i^0 &= \langle i_1 i_2 i_3 \rangle = 2^2 i_1 + 2 i_2 + i_3, \\ i^1 &= \langle i_1 i_2 j_0 \rangle = 2^2 i_1 + 2 i_2 + j_0, \\ i^2 &= \langle i_1 j_1 j_0 \rangle = 2^2 i_1 + 2 j_1 + j_0, \\ i^3 &= \langle j_2 j_1 j_0 \rangle = 2^2 j_2 + 2 j_1 + j_0. \end{aligned}$$

Эти формулы используются для вычисления номеров вершин сетевого графа. На рис. 5 приведены поразрядные представления номеров вершин для каждого слоя графа и частично показаны принципы образования межслойных связей. На рис. 6 показан полный граф, соответствующий каноническому предложению (2).

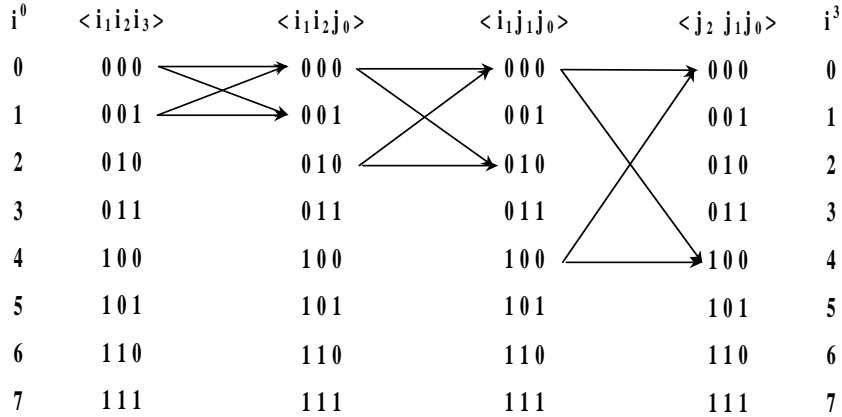


Рис. 5. Правило построения графической интерпретации предложения

Топологическое проектирование быстрых нейронных сетей

Рассмотрим регулярную сеть и будем полагать, что каждая вершина слоя λ представляет собой однослойный персептрон с размерностью рецепторного поля, равной p_λ и числом нейронов — g_λ . Персептрон, соответствующий вершине графа, будем называть *нейронным ядром*. Обозначим через $u_\lambda = [0, 1, \dots, p_\lambda - 1]$ номер рецептора, а через $v_\lambda = [0, 1, \dots, g_\lambda - 1]$ номер нейрона в нейронном ядре. Будем полагать, что любой нейрон имеет только один выходной аксон. Каждой дуге графа поставим в соответствие точную однозначную связь аксон-рецептор. Алфавит языка расширим буквами с родовыми именами u и v .

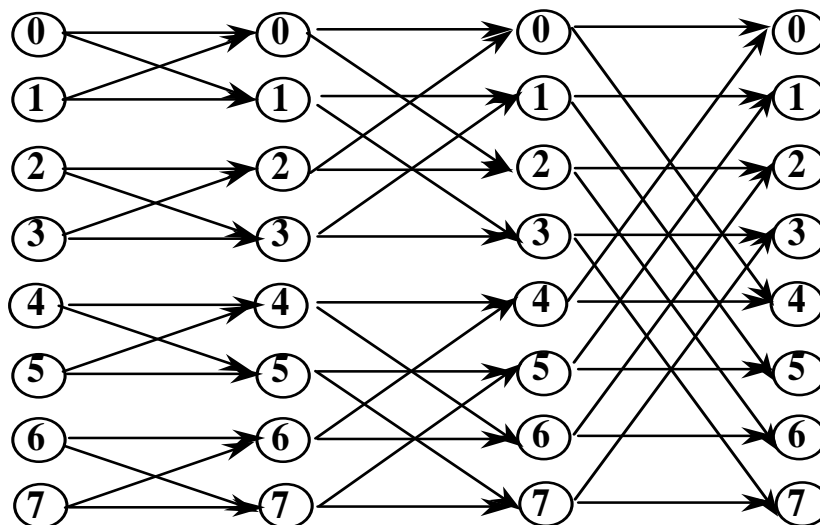


Рис. 6. Графическая интерпретация канонического предложения

Топологии нейронных слоев

Позиционный номер рецепторов и аксонов в пределах слоя может быть задан конвергенциями

$$U^\lambda = \langle \langle i^\lambda \rangle \oplus u_\lambda \rangle, \quad V^\lambda = \langle \langle i^\lambda \rangle \oplus v_\lambda \rangle. \quad (3)$$

Данные выражения включают в себя поразрядное представление номера вершины и дополнительный разряд, соответствующий локальному номеру рецептора/аксона в пределах нейронного ядра. Дополнительный разряд может занимать любую позицию в кортеже слова, что позволяет в широком диапазоне варьировать размещение терминальных полей нейронного ядра в нейронном слое. Слова языка, соответствующие конвергенциям (3), назовем *топологиями* нейронного слоя. Для смежного слоя с порядковым номером $\lambda + 1$ топологии определяются конвергенциями:

$$U^{\lambda+1} = \langle \langle i^{\lambda+1} \rangle \oplus u_{\lambda+1} \rangle, \quad V^{\lambda+1} = \langle \langle i^{\lambda+1} \rangle \oplus v_{\lambda+1} \rangle.$$

Потребуем, чтобы соответствие между аксонами и рецепторами двух смежных слоев удовлетворяло условиям регулярности. В контексте фор-

мального языка регулярное соответствие может быть выражено операцией перестановки

$$\langle\langle i^\lambda \rangle \oplus v_\lambda \rangle * q^\lambda = \langle\langle i^{\lambda+1} \rangle \oplus u_{\lambda+1} \rangle.$$

Это выражение справедливо для всех значений разрядных переменных, т. е. является тождеством. Топологию будем называть компактной, если все межслойные перестановки тождественны единичной. Сеть с компактной топологией наиболее проста в технической реализации. Учитывая, что $I_\lambda \supset I_{\lambda+1}$, $J_\lambda \subset J_{\lambda+1}$, можно сделать вывод о существовании взаимно однозначных соответствий

$$I_\lambda \setminus I_{\lambda+1} \leftrightarrow u_{\lambda+1}, \quad J_{\lambda+1} \setminus J_\lambda \leftrightarrow v_\lambda, \quad (4)$$

(символ “\” означает разность множеств). Соответствия (4) позволяют во всех словах допустимого предложения заменить буквы вида i, j на буквы с родовыми именами вида u, v , в результате получим:

$$V^\lambda = \langle\langle u \rangle_{S_\lambda} \oplus \langle v \rangle_{C_\lambda} \rangle, \quad U^{\lambda+1} = \langle\langle u \rangle_{S_\lambda} \oplus \langle v \rangle_{C_\lambda} \rangle * q^\lambda,$$

где $S_\lambda \supset S_{\lambda+1}$ и $C_\lambda \subset C_{\lambda+1}$ — подмножества смежных классов в алфавите u, v .

Терминальные поля нейронной сети занимают особое положение. Обозначим через U позиционный номер рецептора входного слоя, а через V позиционный номер аксона выходного слоя, тогда топологии терминальных полей можно записать в виде конвергенций:

$$U = \langle\langle i^0 \rangle \oplus u_0 \rangle, \quad V = \langle\langle i^{\varkappa-1} \rangle \oplus v_{\varkappa-1} \rangle.$$

В этих выражениях u_0 и $v_{\varkappa-1}$ — свободные переменные, которые не связаны каким либо соответствием вида (4) с переменными $\langle i \rangle, \langle j \rangle$. Число букв с родовыми именами u, v на единицу больше (по каждому виду), чем число букв с родовыми именами i, j .

Графическая интерпретация топологий

Пусть алфавит языка определен кортежем

$$A = \langle i_1 i_2 i_3 j_2 j_1 j_0 \rangle.$$

Рассмотрим каноническое предложение

$$[\langle i_1 i_2 i_3 \rangle \langle i_1 i_2 j_0 \rangle \langle i_1 j_1 j_0 \rangle \langle j_2 j_1 j_0 \rangle], \quad (5)$$

определяющее структурную модель регулярной сети. Используя соответствия (4), сделаем замены переменных

$$\begin{aligned} I_0 \setminus I_1 = i_3 \leftrightarrow u_1, & \quad I_1 \setminus I_2 = i_2 \leftrightarrow u_2, & \quad I_2 \setminus I_3 = i_1 \leftrightarrow u_3, \\ J_3 \setminus J_2 = j_2 \leftrightarrow v_2, & \quad J_2 \setminus J_1 = j_1 \leftrightarrow v_1, & \quad J_1 \setminus J_0 = j_0 \leftrightarrow v_0, \end{aligned} \quad (6)$$

В новых переменных каноническое предложение примет вид:

$$[\langle u_3 u_2 u_1 \rangle \langle u_3 u_2 v_0 \rangle \langle u_3 v_1 v_0 \rangle \langle v_2 v_1 v_0 \rangle].$$

Для построения топологий рецепторных полей необходимо к каждому слову с порядковым номером λ добавить переменную u_λ . Переменные могут быть добавлены в любые позиции, например, добавим их в конец слов, тогда множество рецепторных топологий будет описываться предложением

$$[\langle u_3 u_2 u_1 u_0 \rangle \langle u_3 u_2 v_0 u_1 \rangle \langle u_3 v_1 v_0 u_2 \rangle \langle v_2 v_1 v_0 u_3 \rangle]. \quad (7)$$

Аналогичным образом для построения топологий аксоновых полей необходимо к каждому слову канонического предложения с порядковым номером λ добавить переменную v_λ . Например, это можно сделать следующим образом:

$$[\langle u_3 u_2 v_0 u_1 \rangle \langle u_3 v_1 v_0 u_2 \rangle \langle v_2 v_1 v_0 u_3 \rangle \langle v_3 v_2 v_1 v_0 \rangle]. \quad (8)$$

Все другие топологии могут быть получены всевозможными перестановками букв в словах полученных предложений. Предложения (7) и (8) будем называть *топологическими*, а их совокупность — *топологической траекторией*. Топологическую траекторию можно интерпретировать как сетевой многослойный граф, каждой вершине которого соответствует один нейрон, а дуги определяют связи между нейронами смежных слоев. Каждое слово в предложении (8) определяет правило для вычисления номера нейрона в пределах слоя. Слова предложения (7) интерпретируются как номер рецептора в пределах слоя. Рецептор и нейрон смежных слоев связываются между собой дугой графа, если значения одноименных разрядов в соответствующих конвергенциях совпадают. Правило построения графа показано на рис. 7. В данном примере основания всех разрядных переменных приняты равными двум. На рис. 8 приведен полный топологический граф быстрой нейронной сети. Для построения компактной топологической траектории можно воспользоваться правилом скользящего интервала. Выберем канон алфавита в виде

$$A = \langle u_0 u_1 \dots u_{\varkappa-1} v_0 v_1 \dots v_{\varkappa-2} v_{\varkappa-1} \rangle.$$

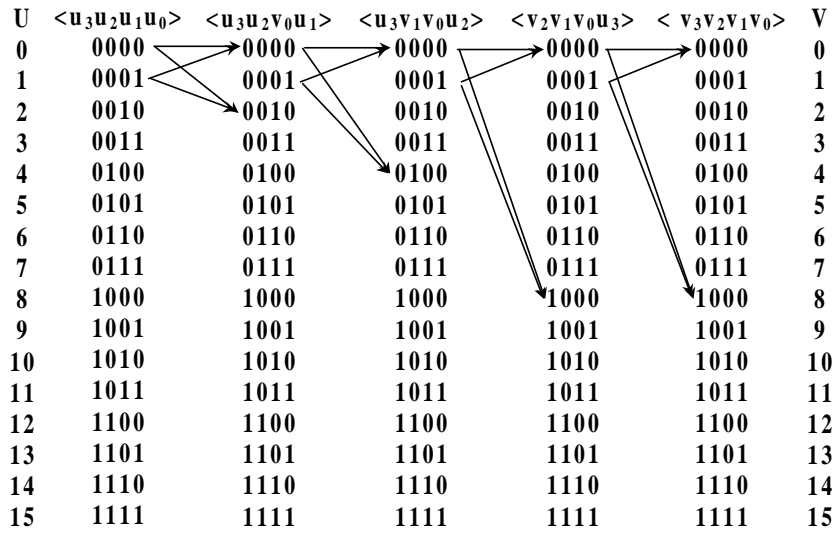


Рис. 7. Правило построения топологического интерпретирующего графа

Будем перемещать пустой интервал длиной \varkappa , ограниченный угловыми скобками вдоль алфавитного кортежа, начиная с крайней левой позиции. На каждом шаге подвижный интервал вычленяет из кортежа каноническое слово. В результате последовательность первых $\varkappa - 1$ порождаемых слов будет определять топологию рецепторных полей нейронной сети, а последнее \varkappa -е слово — топологию терминального аксонового поля сети. Обратное движение интервала с крайне правой позиции к началу канона порождает топологии аксоновых полей сети.

Правило скользящего интервала можно представить в аналитической форме. Обозначим через t^λ слово, выделяемое подвижным интервалом на шаге λ . Нетрудно заметить, что

$$t^\lambda = \langle u_\lambda u_{\lambda+1} \dots u_{\varkappa-1} v_0 v_1 \dots v_{\lambda-1} \rangle. \quad (9)$$

Можно проверить, что данное правило соответствует схеме алгоритма БПФ в форме Гуда (см. рис. 9). Особенность порождающей схемы Гуда состоит в том, что все слои графа имеют одинаковую топологию, что упрощает алгоритм реализации. Однако выходные операнды на каждой базовой операции

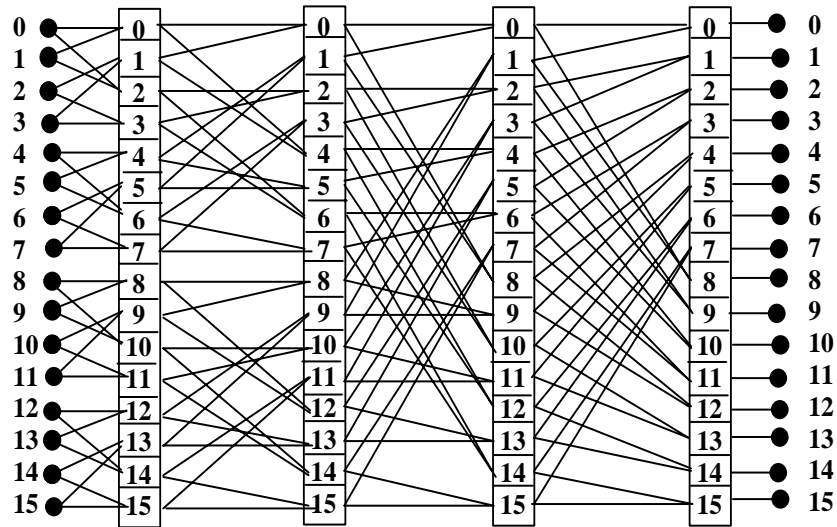


Рис. 8. Графическая интерпретация топологической траектории

не могут замещать входные отсчеты, поэтому требуется дополнительная память для хранения выходной последовательности. Далее по тексту будут представлены другие варианты порождающих схем (см. также [10]).

Граничные условия и топологические матрицы

В общем случае топологии терминальных полей сети могут быть записаны в следующем виде:

$$U = \langle u \rangle_S * q_b, \quad V = \langle v \rangle_C * q_a, \quad (10)$$

где q_a, q_b — регулярные перестановки. Векторы на входе и выходе сети имеют свою собственную нумерацию координат, для которой также будем использовать поразрядную форму, положив

$$U = \langle U \rangle_X \quad \text{и} \quad V = \langle V \rangle_Y,$$

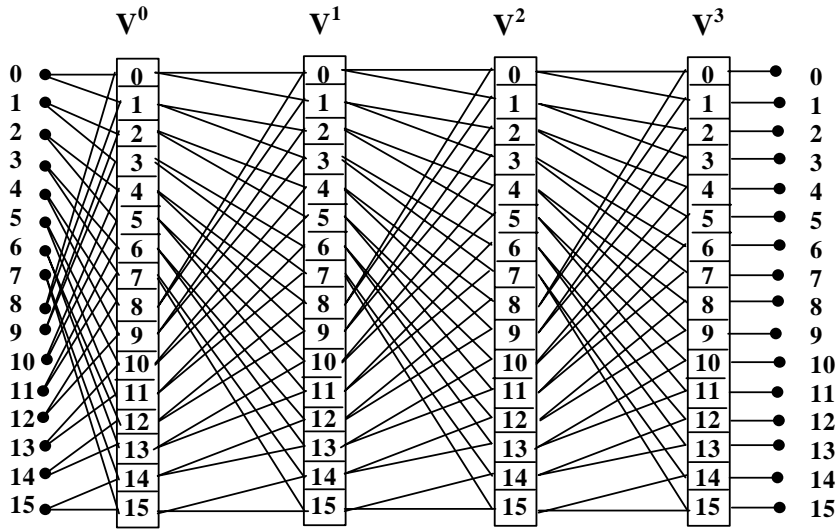


Рис. 9. Топологическая реализация схемы Гуда

где X, Y — линейно упорядоченные множества разрядных переменных. Например, можно выбрать

$$U = \langle U_{\varkappa-1} U_{\varkappa-2} \dots U_0 \rangle, \quad V = \langle V_{\varkappa-1} V_{\varkappa-2} \dots V_0 \rangle.$$

Разрядные числа U_i, V_i назовем *глобальными* разрядными числами в отличие от u_i, v_i для которых будем в дальнейшем использовать определение «*локальные*». В данном случае к определению «глобальные» следует добавить и определение «терминальные», поскольку они описывают граничные слои сети. Выражения (10) устанавливают соответствия между глобальными и локальными разрядными числами. Эти соответствия можно рассматривать как граничные условия для топологической траектории нейронной сети. Подобным образом можно ввести глобальные разрядные переменные для всех скрытых слоев, положив:

$$U^\lambda = \langle U_{\varkappa-1}^\lambda U_{\varkappa-2}^\lambda \dots U_0^\lambda \rangle, \quad V = \langle V_{\varkappa-1}^\lambda V_{\varkappa-2}^\lambda \dots V_0^\lambda \rangle.$$

Для каждого слоя справедливы следующие соответствия между локальными и глобальными переменными:

$$\langle U_{\varkappa-1}^\lambda U_{\varkappa-2}^\lambda \dots U_0^\lambda \rangle = t^\lambda * q_b^{\lambda-1}, \quad \langle V_{\varkappa-1}^\lambda V_{\varkappa-2}^\lambda \dots V_0^\lambda \rangle = t^\lambda * q_a^\lambda \quad (11)$$

(разложение $q^\lambda = (q_a^\lambda)^{-1} q_b^\lambda$ использовано из соображений симметрии). Последовательность слов вида $\langle \langle U \rangle_{X_\lambda} \oplus \langle V \rangle_{Y_\lambda} \rangle$ в алфавите терминальных переменных, соответствующая последовательности шагов топологической траектории, назовем *внешней траекторией топологий*. Пусть t^λ слово топологического предложения, определяемое некоторой регулярной порождающей схемой. Ограничимся компактными топологиями и представим соответствия (11) в табличном виде. Например, для топологии Гуда:

$$\begin{aligned} U^\lambda = t^\lambda &= \langle u_\lambda u_{\lambda+1} \dots u_{\varkappa-2} u_{\varkappa-1} v_0 v_1 \dots v_{\lambda-1} \rangle, \\ V^\lambda = t^{\lambda+1} &= \langle u_{\lambda+1} \dots u_{\varkappa-2} u_{\varkappa-1} v_0 v_1 \dots v_{\lambda-1} v_\lambda \rangle. \end{aligned} \quad (12)$$

Данное соответствие представляется таблицей 1. Построим по данной таблице матрицу, T_λ состоящую из нулей и единиц, используя при этом следующее правило: элемент матрицы $T_\lambda(U^\lambda, V^\lambda)$ равен единице, когда одноименные разрядные числа в конвенциях U^λ, V^λ совпадают по значениям. Следуя данному правилу и используя первую и третью строки таблицы, можно получить следующую аналитическую форму для определения элементов матрицы

ТАБЛИЦА 1. Соответствие локальных и глобальных переменных

$U^\lambda =$	u_λ	$u_{\lambda+1}$	\dots	$u_{\varkappa-2}$	$u_{\varkappa-1}$	v_0	\dots	$v_{\lambda-3}$	$v_{\lambda-2}$	$v_{\lambda-1}$
$U^\lambda =$	$U_{\varkappa-1}^\lambda$	$U_{\varkappa-2}^\lambda$	\dots	$U_{\varkappa-\lambda+1}^\lambda$	$U_{\varkappa-\lambda}^\lambda$	$U_{\varkappa-\lambda-1}^\lambda$	\dots	U_2^λ	U_1^λ	U_0^λ
$V^\lambda =$	$u_{\lambda+1}$	$u_{\lambda+2}$	\dots	$u_{\varkappa-1}$	v_0	v_1	\dots	$v_{\lambda-2}$	$v_{\lambda-1}$	v_λ
$V^\lambda =$	$V_{\varkappa-1}^\lambda$	$V_{\varkappa-2}^\lambda$	\dots	$V_{\varkappa-\lambda+1}^\lambda$	$V_{\varkappa-\lambda}^\lambda$	$V_{\varkappa-\lambda-1}^\lambda$	\dots	V_2^λ	V_1^λ	V_0^λ

$$T_\lambda(U^\lambda, V^\lambda) = \delta(U_{\varkappa-2}^\lambda, V_{\varkappa-1}^\lambda) \delta(U_{\varkappa-3}^\lambda, V_{\varkappa-2}^\lambda) \dots \delta(U_0^\lambda, V_1^\lambda),$$

где $\delta(\cdot)$ — функция Кронекера. Например, для четырехслойной сети с основанием 2 (см. рис. 9) топологическая матрица слоя λ будет определяться выражением

$$T_\lambda(U^\lambda, V^\lambda) = \delta(U_2^\lambda, V_3^\lambda) \delta(U_1^\lambda, V_2^\lambda) \delta(U_0^\lambda, V_1^\lambda).$$

На рис. 10 показан вид топологической матрицы, отвечающей данному выражению; там же для удобства показана поразрядная нумерация строк и столбцов. На поле матрицы прямоугольникам выделены элементы, относящиеся к одному нейронному ядру, незаполненные позиции матрицы соответствуют нулевым значениям.

По существу данная матрица представляет собой шаблон синаптической карты нейронного слоя. При реализации нейронной сети ненулевые позиции топологической матрицы занимают синаптическим весами нейронных ядер. Обозначим матрицу синаптической карты через H_λ . Используя аналитическую форму для топологической матрицы и приведенную выше таблицу, можно получить следующее аналитическое представление для элементов матрицы H_λ :

$$h_\lambda(U^\lambda, V^\lambda) = w_{i_\lambda}^\lambda(U_{\varkappa-1}^\lambda, V_0^\lambda) \delta(U_{\varkappa-2}^\lambda, V_{\varkappa-1}^\lambda) \delta(U_{\varkappa-3}^\lambda, V_{\varkappa-2}^\lambda) \dots \dots \delta(U_0^\lambda, V_1^\lambda),$$

здесь через $w_{i_\lambda}^\lambda(\cdot)$ обозначен элемент синаптической карты нейронного ядра i^λ . Номер ядра, выраженный в глобальных переменных, может быть представлен одним из выражений:

$$i^\lambda = \langle U_{\varkappa-2}^\lambda U_{\varkappa-3}^\lambda \dots U_0^\lambda \rangle = \langle V_{\varkappa-1}^\lambda V_{\varkappa-2}^\lambda \dots V_1^\lambda \rangle.$$

Алгоритм обработки данных в БНС

Обработка данных в нейронном ядре i^λ слоя λ определяется выражениями:

$$\begin{aligned} s_{i_\lambda}^\lambda(v_\lambda) &= \sum_{u_\lambda} x_{i_\lambda}^\lambda(u_\lambda) w_{i_\lambda}^\lambda(u_\lambda, v_\lambda), \\ y_{i_\lambda}^\lambda(v_\lambda) &= f_{v_\lambda}^\lambda(s_{i_\lambda}^\lambda(v_\lambda)), \end{aligned} \quad (13)$$

V_3	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
V_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
V_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
V_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$U_3U_2U_1U_0$																
0000	1	1														
0001			1	1												
0010					1	1										
0011							1	1								
0100									1	1						
0101											1	1				
0110													1	1		
0111															1	1
1000	1	1														
1001			1	1												
1010					1	1										
1011							1	1								
1100									1	1						
1101											1	1				
1110													1	1		
1111															1	1

Рис. 10. Топологическая матрица для порождающей схемы Гуда

где $f_{v_\lambda}^\lambda(\cdot)$ — функция активации, $x_{i_{\lambda+1}}^{\lambda+1}(v_\lambda)$ и $y_{i_\lambda}^\lambda(v_\lambda)$ — входные и выходные переменные. Для построения алгоритма обработки данных необходимо выполнить переход от локальных переменных нейронного ядра к глобальным переменным нейронного слоя. Рассмотрим в качестве примера нейронную сеть с топологией Гуда. Кортеж соответствующий номеру ядра можно получить, удалив из топологического слова t^λ (см. выражение (9)) букву u_λ . В результате получим:

$$i^\lambda = \langle u_{\lambda+1} \dots u_{\lambda-1} v_0 v_1 \dots v_{\lambda-1} \rangle. \tag{14}$$

Следуя семантике слов, глобальный номер рецептора и глобальный номер аксона в слое λ можно вычислить, используя алгебраические выражения:

$$\begin{aligned} U^\lambda &= u_\lambda p_{\lambda+1} p_{\lambda+2} \dots p_{\lambda-1} g_0 g_1 \dots g_{\lambda-1} + i^\lambda, \\ V^\lambda &= i^\lambda g_\lambda + v. \end{aligned} \tag{15}$$

Формулы (13), (14), (15) полностью определяют обработку данных в нейронном слое. В целом для нейронной сети алгоритм заключается в последовательной обработке данных по нейронным слоям.

Слабосвязанные сети

Условие слабой связанности [11–13] выражает собой отношение между окрестностями вершин и их проекциями на терминальные слои. В терминах нейронных сетей входной терминальный слой называется *афферентом*, а выходной — *эфферентом*. Под афферентной проекцией некоторой вершины B (далее обозначается $\text{Afr}(B)$) понимается подмножество вершин афферентного слоя, связанных дугами с этой вершиной. Аналогично под эфферентной проекцией вершины B (далее обозначается $\text{Efr}(B)$) понимается подмножество вершин эфферентного слоя связанных дугами с этой вершиной. Определим также рецепторную окрестность $\Gamma^{-1}(B)$ как множество вершин непосредственно предшествующего слоя, связанных дугами с вершиной B и аксоновую окрестность $\Gamma(B)$, как множество вершин последующего слоя, связанных дугами с вершиной B . На рис. 11 показан пример структурной модели трехслойной ядерной нейронной сети. Для вершины B_1 данного примера терминальные проекции будут иметь следующий состав:

$$\text{Afr}(B_1) = \{A_3, A_4, A_5\}, \quad \text{Efr}(B_1) = \{C_0, C_1\}.$$

Принцип слабой связанности может быть выражен парой симметричных условий:

$$\begin{aligned} \text{Afr}(B) &= \sum_{A \in \Gamma^{-1}(B)} \text{Afr}(A), \\ \text{Efr}(B) &= \sum_{C \in \Gamma(B)} \text{Efr}(C), \end{aligned}$$

где символ \sum обозначает прямую сумму соответствующих множеств. Первое выражение устанавливает, что в слабосвязанной сети для любой вершины B афферентные проекции вершин ее рецепторной окрестности не пересекаются. Второе — устанавливает аналогичное условие для эфферентных проекций. Фактически оба выражения двойственны друг другу и если выполняется одно из них, то обязательно будет выполнено и другое. Используя данное определение несложно показать, что в слабосвязанных сетях отсутствуют параллельные пути между вершинами, т. е. сети являются структурно безызбыточными.

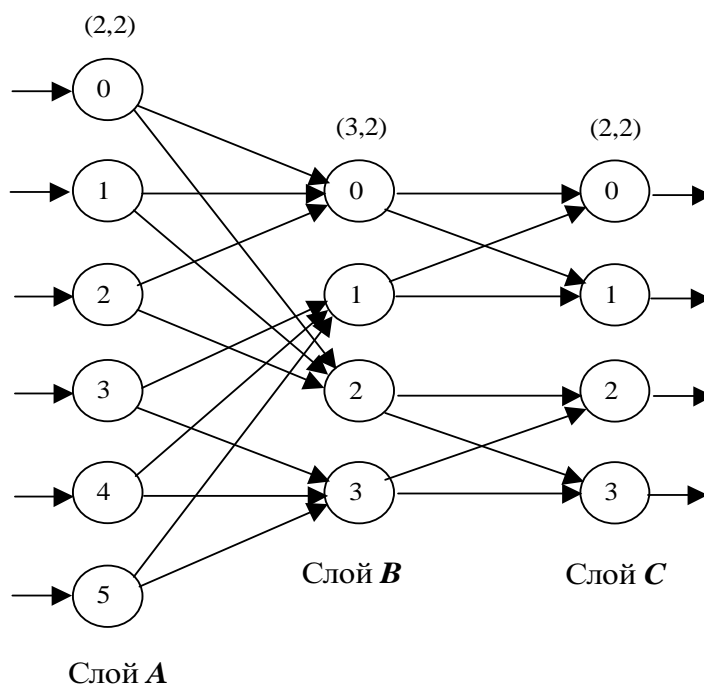


Рис. 11. Переход от восьмиточечного ДПФ к двум четырехточечным

Докажем теперь, что грамматика формального языка порождает слабо-связанные структуры. В интерпретирующем графе регулярной сети номер вершины в слое λ определяется выражением

$$i^\lambda = \langle \langle i \rangle_{I_\lambda} \oplus \langle j \rangle_{J_\lambda} \rangle, \quad (16)$$

а номера вершин начального и конечного слоя можно представить в виде:

$$i = \langle \langle i \rangle_{I_\lambda} \oplus \langle i \rangle_{I \setminus I_\lambda} \rangle, \quad j = \langle \langle j \rangle_{J_\lambda} \oplus \langle j \rangle_{J \setminus J_\lambda} \rangle, \quad (17)$$

где $I \setminus I_\lambda$ и $J \setminus J_\lambda$ обозначают разность множеств. По принципу построения интерпретирующего графа вершины двух смежных слоев связаны дугой, если одноименные переменные в смежных словах предложения имеют совпадающие значения. Выполняя индукцию по номеру слоя, нетрудно пока-

зять, что вершина i^λ будет связана с теми вершинами начального и конечного слоя, для которых одноименные разрядные переменные в выражениях (17) будут иметь те же значения, что и в выражении (16). Следовательно, терминальные проекции вершины i^λ можно выразить следующими параметрическими формами:

$$\text{Afr}(i^\lambda) = \left\langle \langle i \rangle_{I_\lambda} \oplus \left(\langle i \rangle_{I \setminus I_\lambda} \right) \right\rangle, \quad \text{Efr}(i^\lambda) = \left\langle \langle j \rangle_{J_\lambda} \oplus \left(\langle j \rangle_{J \setminus J_\lambda} \right) \right\rangle.$$

Здесь круглыми скобками выделены варьируемые разряды терминальных слов. Тогда для афферента вершины $i^{\lambda-1}$ слоя $\lambda - 1$ можно записать

$$\text{Afr}(i^{\lambda-1}) = \left\langle \langle i \rangle_{I_{\lambda-1}} \oplus \left(\langle i \rangle_{I \setminus I_{\lambda-1}} \right) \right\rangle. \quad (18)$$

Будем полагать, что вершины i^λ и $i^{\lambda-1}$ связаны между собой. По правилам порождающей грамматики $I_\lambda \subset I_{\lambda-1}$, причем разность множеств $I_{\lambda-1} \setminus I_\lambda$ состоит точно из одной буквы (разряда). Из свойств интерпретирующей функции (1) следует

$$\text{Afr}(i^\lambda) = \bigoplus_{I_{\lambda-1} \setminus I_\lambda} \text{Afr}(i^{\lambda-1}), \quad (19)$$

где прямая сумма подмножеств берется по всем значениям разрядного числа $I_{\lambda-1} \setminus I_\lambda$. Аналогично для эфферентов вершин смежных слоев λ и $\lambda + 1$ можно получить:

$$\text{Efr}(i^\lambda) = \bigoplus_{J_{\lambda+1} \setminus J_\lambda} \text{Efr}(i^{\lambda+1}). \quad (20)$$

Выражения, (19), (20) показывают, что интерпретирующий граф представляет собой слабосвязанную сеть.

Можно указать ряд причин, стимулирующих интерес к слабосвязанным нейронным сетям:

- К классу слабосвязанных сетей принадлежат как БНС, так и обычные многослойные полностью связанные сети.
- При реализации нейронных сетей на суперкомпьютерах структура нейронной сети подчиняется структуре вычислительной системы. Такие известные структуры суперкомпьютеров как Вануан, «Омега», и др. [14] являются слабосвязанными сетями или строятся на основе слабосвязанных сетей.

- В слабосвязанных сетях реализуется принцип генетического подоби́я структуры [15], по заключениям нейрофизиологов [16] подобный принцип явился основным порождающим механизмом в эволюции новой коры головного мозга.
- Слабосвязанные сети обладают фрактальными свойствами [17].

Системные характеристики БНС

Вычислительная эффективность

Быстродействие вычислительных алгоритмов, как правило, оценивается числом операций умножения, необходимых для их реализации. Синаптическая карта нейронного ядра с размерностью рецепторного поля p и числом нейронов g представляет собой матрицу размерности $p \times g$. Обработка входного вектора в нейронном ядре соответствует умножению вектора на матрицу с последующим нелинейным преобразованием, определяемым функциями активации. Для выполнения матричного умножения требуется $p \cdot g$ скалярных операций умножения. Операции умножения, связанные с вычислением функций активации обычно не учитываются.

Рассмотрим \varkappa -слойную БНС с характеристиками нейронных ядер:

$$(p_0, g_0), (p_1, g_1), \dots, (p_{\varkappa-1}, g_{\varkappa-1}).$$

Сеть имеет размерность по входу $N = p_0 p_1 \dots p_{\varkappa-1}$, а по выходу $M = g_0 g_1 \dots g_{\varkappa-1}$. Из (14) следует, что число ядер в слое m будет равно произведению

$$k_m = p_{m+1} p_{m+2} \dots p_{\varkappa-1} g_0 g_1 \dots g_{m-1}. \quad (21)$$

При этом в каждом нейронном ядре слоя m выполняется $p_m \cdot g_m$ операций умножения. Суммируя по всем нейронным слоям, получим:

$$Z = \sum_{m=0}^{\varkappa-1} p_m p_{m+1} \dots p_{\varkappa-1} g_0 g_1 \dots g_{m-1} g_m.$$

Пластичность

Высокое быстродействие БНС достигается за счет ограничения числа межнейронных связей. Расплатой за вычислительную эффективность является

снижение «уровня интеллекта» нейронной сети. Качественной характеристикой интеллекта является известное из биологии понятие пластичности, т. е. способности нейронной сети изменять свои параметры при обучении. К количественной оценке степени пластичности можно подойти следующим образом. Нейронная сеть, заданная на векторных пространствах E, D реализует отображение $E \rightarrow D$. Класс операторов нейронной сети, порождаемый изменением ее синаптических весов, образует многомерную поверхность в пространстве операторов, размерность которой определяется максимальной размерностью ее касательного пространства. Это числовая характеристика является наиболее подходящей оценкой степени параметрической пластичности нейронной сети. Нейронную сеть можно представить как материальную точку, дрейфующую в операторном пространстве при вариации синаптических весов. В этой интерпретации степень параметрической пластичности является прямым эквивалентом известного из механики понятия «число степеней свободы» материальной точки.

В касательном пространстве все операторы нейронной сети можно рассматривать как линейные, поэтому исходная нелинейная задача сводится к исследованию пластичности перестраиваемых линейных операторов. Для БНС найдено аналитическое выражение для расчета числа степеней свободы по структурным характеристикам сети [18]. Расчетная формула имеет вид:

$$S = \sum_{m=0}^{n-1} p_m g_m k_m - \sum_{m=0}^{n-2} D_m.$$

где (p_m, g_m) — размерность ядра слоя m ; k_m — число ядер в слое m ; $D_m = g_m k_m$ — число связей в переходе между слоями m и $m + 1$.

На рис. 12 приведены относительные зависимости пластичности и вычислительных затрат от размерности БНС. Графики построены для сети с одинаковой размерностью по входу и выходу. Базой для сравнения является однослойный перцептрон для которого оценки равны, очевидно, $Z_0 = S_0 = N^2$.

Способность к обучению

Покажем, как степень параметрической пластичности связана с количеством линейно независимых образов, которое может распознавать нейронная сеть. Пусть H — линеаризованный оператор нейронной сети. Обозначим через X и Y , соответственно, входной и выходной векторы сети.

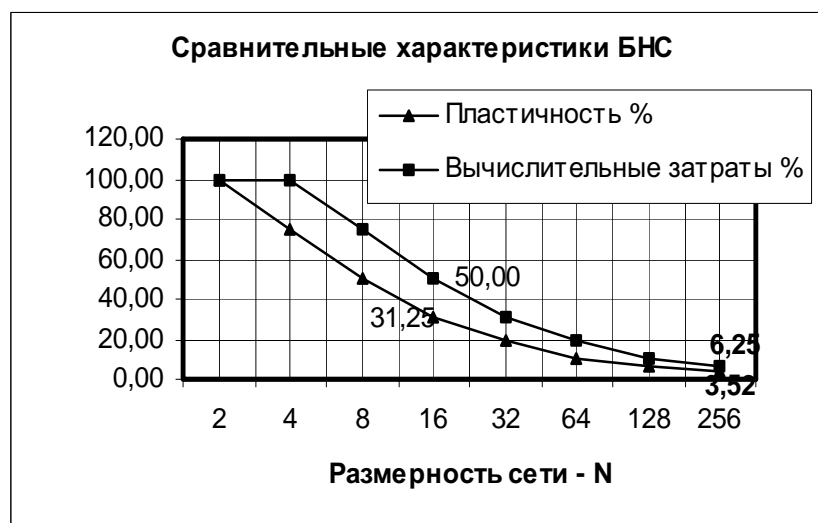


Рис. 12. Зависимость вычислительных затрат и быстродействия БНС от размерности сети

Множество обучающих примеров можно задать набором, состоящим из k пар векторов: $(X_1 Y_1), (X_2 Y_2), \dots, (X_k Y_k)$. Обозначим через $L(X)$ линейную оболочку входных векторов, а через $L(Y)$ — линейную оболочку выходных векторов обучающего множества. Линейные оболочки являются минимальными векторными пространствами, которые содержат все вектора обучающего множества, но, кроме того, они содержат несчетное множество векторов, которые могут быть получены как линейные комбинации образующих векторов.

Способность нейронной сети к обобщению позволяет расширить обучающее множество до непрерывного многообразия примеров W_{XY} , которое, очевидно, изоморфно тензорному произведению линейных оболочек: $W_{XY} \cong L(X) \otimes L(Y)$. Обозначим размерности линейных оболочек $L(X)$ и $L(Y)$, соответственно, через r_X и r_Y . Тогда размерность многообразия примеров будет равна $\dim W_{XY} = r_X r_Y$.

Необходимым условием непротиворечивости данных является выполнение неравенства: $r_Y \leq r_X$. Вариации синаптических весов нейронной сети порождают многообразие операторов W_H . Для того, чтобы нейронная

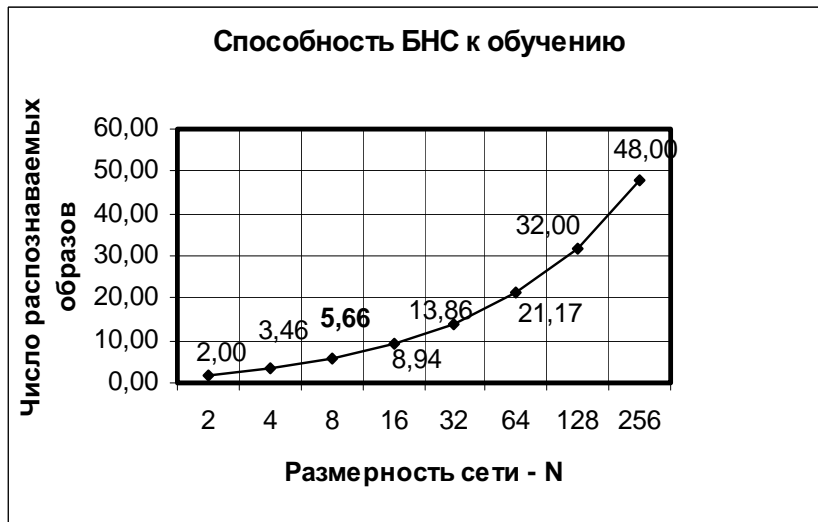


Рис. 13. Зависимость числа распознаваемых образов от размерности БНС

сеть была способна обучиться к многообразию примеров, необходимо выполнить условие $W_H \geq W_{XY}$, следствием которого является неравенство:

$$S \geq r_X r_Y, \quad (22)$$

где S — число степеней свободы нейронной сети. Если образы в обучающем множестве линейно независимы, то набор векторов Y_i образует базис линейной оболочки $L(Y)$, это означает, что $r_Y = k$. При отсутствии противоречий размерность многообразия $L(X)$ удовлетворяет условию: $r_X \geq k$. В этом случае из (22) непосредственно следует $k \leq \sqrt{S}$. На рис. 13 приведена расчетная зависимость числа распознаваемых образов от размерности сети. Например, для БНС размерности $N = 8$, показанной на рис. 4, расчетное количество независимо распознаваемых образов равно 5,66.

На рис. 14 приведены результаты эксперимента по распознаванию образов для данной нейронной сети. В эксперименте использовалась нейронная сеть с сигмоидными функциями активации в первом и во втором слое и линейными функциями активации в последнем, третьем слое. В качестве обучающего множества использовались ортогональные функции Уолша по

входу и унитарный единичный код по выходу. Сеть обучалась с помощью алгоритма Error BackPropagation.



Рис. 14. Зависимость ошибки обучения БНС (N=8) от размера обучающей выборки

Перестраиваемые спектральные преобразования

После разработки метода БПФ интенсивность исследований в области цифровой обработки резко возросла. Развитие исследований, во-первых, привело к расширению класса используемых спектральных преобразований, а во-вторых, к разработке эффективных методов их реализации. В практике цифрового спектрального анализа вошли преобразования Уолша–Адамара, Виленкина–Крестенсона, Хаара, косинусное, наклонное, вейвлет и другие. Каждый вид преобразования обладает теми или иными свойствами, определяющими область его применения. Довольно быстро выяснилось, что большинство используемых спектральных преобразований имеет один и тот же принцип формирования быстрого алгоритма. Этот принцип получил название «факторизация матриц» и заключался в разложении матрицы

дискретного спектрального преобразования H в произведение слабозаполненных матриц:

$$H = H_0 H_1, \dots, H_{x-1}.$$

Структура слабозаполненных матриц устанавливалась подходящей теоремой факторизации. Было доказано множество теорем факторизации (это «творчество» продолжается и в наши дни), но постепенно стало понятным, что по одной и той же схеме факторизации можно реализовать быстрые алгоритмы для различных спектральных преобразований, путем изменения только коэффициентов в факторизованном представлении. В итоге, начиная с работ Г. Эндрюса (1970 г.), получило развитие направление обобщенного спектрального анализа, предметом исследования которого стали перестраиваемые спектральные преобразования. При реализации перестраиваемых преобразований выбирается одна из известных регулярных топологий (Гуда или Кули–Тьюки) и под данную топологию определяются значения коэффициентов слабозаполненных матриц. Ограничением развития этого направления вскоре стала сама топологическая схема, поскольку закрепленная топология, ограничивала возможности синтеза спектральных преобразований. На этом этапе перестраиваемые преобразования пересеклись с нейронными сетями, в тех и других присутствует массив настраиваемых коэффициентов и те и другие имеют послойную структурную организацию, но в отличие от нейронных сетей спектральные перестраиваемые преобразования являются линейными и ортогональными. Аналогом нейронного ядра в перестраиваемых преобразованиях является спектральное ядро, которое определяется квадратной ортогональной матрицей небольшой размерности. Функции активации нейронов ядра линейны, а смещения отсутствуют. В предыдущих параграфах были рассмотрены структурные модели и топологии БНС, полученные результаты в полной мере переносятся на перестраиваемые спектральные преобразования. Развитые методы топологического проектирования БНС позволяют существенно расширить класс перестраиваемых преобразований.

Настройка перестраиваемых преобразований

Перестраиваемое преобразование можно рассматривать как быструю нейронную сеть, где в каждом нейронном ядре i^λ выполнится линейное пре-

образование:

$$y_{i\lambda}^\lambda(v_\lambda) = \sum_{u_\lambda} x_{i\lambda}^\lambda(u_\lambda) w_{i\lambda}^\lambda(u_\lambda, v_\lambda). \quad (23)$$

Метод настройки основан на аналитическом представлении сигнальных передач рецептор-аксон между терминальными слоями сети. Функция передач определяется частной производной

$$h(U, V) = \frac{\partial y_{i_{\varkappa-1}}^{\varkappa-1}(v_{\varkappa-1})}{\partial x_{i_0}^0(u_0)},$$

где U, V — глобальные номера рецепторов и аксонов терминальных слоев. В терминологии спектральных преобразований U — это индекс координаты вектора входного сигнала, а V — номер спектрального коэффициента. Элементы $h(U, V)$ образуют квадратную матрицу H спектрального преобразования $Y = XH$. Поскольку БНС — слабосвязанная сеть, то для каждой пары нейронных ядер, одно из которых принадлежит входному слою, а второе — выходному, существует единственный связывающий их путь. Последовательно дифференцируя вдоль выбранного пути цепочку уравнений (23), получим:

$$h(U, V) = w_{i_0}^0(u_0, v_0) w_{i_1}^1(u_1, v_1) \dots w_{i_{\varkappa-1}}^{\varkappa-1}(u_{\varkappa-1}, v_{\varkappa-1}). \quad (24)$$

Цель настройки перестраиваемого преобразования состоит в том, чтобы по известной матрице H найти элементы ядер и определить топологию слабозаполненных матриц H_λ . Топология данных матриц непосредственно связана с внешним представлением топологической траектории. Поэтому задача топологического синтеза интерпретируется как реализация заданной траектории. Процедура синтеза топологии, как правило, допускает множество возможных решений. Дополнительными требованиями могут быть компактность и регулярность топологической траектории, что упрощает реализацию алгоритма быстрого преобразования.

Настройка на базис Уолша

Функции базиса Уолша в упорядочении Пэли [9] задаются на интервале длиной $N = 2^\varkappa$ следующим выражением :

$$\text{pal}(U, V) = \prod_{\lambda=0}^{\varkappa-1} (-1)^{U_\lambda V_\lambda}, \quad (25)$$

где

$$U = \langle U_{\varkappa-1} U_{\varkappa-2} \dots U_0 \rangle \quad \text{и} \quad V = \langle V_{\varkappa-1} V_{\varkappa-2} \dots V_0 \rangle. \quad (26)$$

Все разрядные переменные в данных формулах принимают значения $\{0, 1\}$.

Построим матрицу спектрального преобразования так, чтобы функции базиса располагались вдоль столбцов. В процессе настройки необходимо определить, топологическую траекторию быстрого преобразования и параметры нейронных ядер. Дополнительно потребуем, чтобы топологическая траектория была компактной и регулярной.

Параметры нейронных ядер. Сравнивая (24) с определением функций Уолша, получим

$$w_{i\lambda}^\lambda(u_\lambda, v_\lambda) = (-1)^{u_\lambda v_\lambda}, \quad (27)$$

$$u_\lambda = U_\lambda, \quad v_\lambda = V_\lambda. \quad (28)$$

Выражению (27) соответствует матрица ядра:

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

Очевидно, что нейронные ядра одинаковы во всех слоях сети.

Траектория топологий. Граничные условия для компактной траектории устанавливаются выражениями (26), (28) и имеют вид:

$$\begin{aligned} U &= \langle U_{\varkappa-1} U_{\varkappa-2} \dots U_0 \rangle = \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_0 \rangle, \\ V &= \langle V_{\varkappa-1} V_{\varkappa-2} \dots V_0 \rangle = \langle v_{\varkappa-1} v_{\varkappa-2} \dots v_0 \rangle. \end{aligned}$$

На промежуточные шаги траектории, каких либо требований не накладывается, поэтому траекторию топологий можно выбрать с большим произволом. Воспользуемся регулярной порождающей схемой Кули-Тьюки «с прореживанием по времени»:

$$\begin{aligned} U^\lambda &= \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_{\lambda+1} u_\lambda v_{\lambda-1} v_{\lambda-2} \dots v_0 \rangle, \\ V^\lambda &= \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_{\lambda+1} v_\lambda v_{\lambda-1} v_{\lambda-2} \dots v_0 \rangle. \end{aligned}$$

Для данной схемы номер ядра определяется выражением

$$i^\lambda = \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_{\lambda+1} v_{\lambda-1} v_{\lambda-2} \dots v_0 \rangle.$$

Структуру топологических матриц можно определить из табличного соответствия (см. таблицу 2). На основании данной таблицы получим, что аналитическая форма слабозаполненных матриц будет иметь вид

ТАБЛИЦА 2. Соответствие локальных и глобальных переменных в топологии Кули-Тьюки «с прореживанием по времени»

$U^\lambda =$	$u_{\kappa-1}$	$u_{\kappa-2}$	\dots	$u_{\lambda+1}$	u_λ	$v_{\lambda-1}$	\dots	v_2	v_1	v_0
$U^\lambda =$	$U_{\kappa-1}^\lambda$	$U_{\kappa-2}^\lambda$	\dots	$U_{\lambda+1}^\lambda$	U_λ^λ	$U_{\lambda-1}^\lambda$	\dots	U_2^λ	U_1^λ	U_0^λ
$V^\lambda =$	$u_{\kappa-1}$	$u_{\kappa-2}$	\dots	$u_{\lambda+1}$	v_λ	$v_{\lambda-1}$	\dots	v_2	v_1	v_0
$V^\lambda =$	$V_{\kappa-1}^\lambda$	$V_{\kappa-2}^\lambda$	\dots	$V_{\lambda+1}^\lambda$	V_λ^λ	$V_{\lambda-1}^\lambda$	\dots	V_2^λ	V_1^λ	V_0^λ

$$h_\lambda(U^\lambda, V^\lambda) = w_{i^\lambda}^\lambda(U_\lambda^\lambda, V_\lambda^\lambda) \delta(U_{\kappa-1}^\lambda, V_{\kappa-1}^\lambda) \dots \dots \delta(U_{\lambda+1}^\lambda, V_{\lambda+1}^\lambda) \delta(U_{\lambda-1}^\lambda, V_{\lambda-1}^\lambda) \dots \delta(U_0^\lambda, V_0^\lambda), \quad (29)$$

где $i^\lambda = \langle V_{\kappa-1}^\lambda V_{\kappa-2}^\lambda \dots V_{\lambda+1}^\lambda V_{\lambda-1}^\lambda \dots V_0^\lambda \rangle$. На рис. 15 показано матричное представление алгоритма быстрого преобразования Уолша для размерности $N = 2^3$. Матрица преобразования факторизуется в произведение трех матриц:

$$H = H_0 H_1 H_2.$$

Поразрядное представление строк и столбцов матрицы H можно записать в виде $U = \langle U_2 U_1 U_0 \rangle$, $V = \langle V_2 V_1 V_0 \rangle$. На основании (29) имеем, что слабозаполненные матрицы по слоям определяются выражениями:

- Слой 0:

$$h_0(U^0, V^0) = w_{i^0}^0(U_0^0, V_0^0) \delta(U_2^0, V_2^0) \delta(U_1^0, V_1^0), \quad i^0 = \langle V_2^0 V_1^0 \rangle.$$

- Слой 1:

$$h_1(U^1, V^1) = w_{i^1}^1(U_1^1, V_1^1) \delta(U_2^1, V_2^1) \delta(U_0^1, V_0^1), \quad i^1 = \langle V_2^1 V_0^1 \rangle.$$

- Слой 2:

$$h_2(U^2, V^2) = w_{i^2}^2(U_2^2, V_2^2) \delta(U_1^2, V_1^2) \delta(U_0^2, V_0^2), \quad i^2 = \langle V_1^2 V_0^2 \rangle$$

V_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
V_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
V_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
U_2	U_1	U_0																						
0	0	0	1	1					1	1							1			1				
0	0	1	1	-1					1	1							1			1				
0	1	0		1	1				1	-1								1				1		
0	1	1		1	-1				1	-1								1				1		
1	0	0			1	1					1	1					1			-1				
1	0	1			1	-1					1	1					1			-1				
1	1	0				1	1				1	-1					1			-1				
1	1	1				1	-1				1	-1					1			1				-1

Рис. 15. Матричное представление быстрого алгоритма для преобразования Уолша-Пэли

Перемножив матрицы факторизованного представления, получим следующую матрицу базисных функций:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

Эта же матрица может быть получена из определения (25).

Настройка на базис Фурье

По определению функции базиса Фурье задаются выражением:

$$F_N(U, V) = \frac{1}{\sqrt{N}} \exp\left(-j \frac{2\pi}{N} UV\right),$$

где U — временной отсчет, V — частота (или номер) базисной функции, $j = \sqrt{-1}$, $N = p_0 p_1 \dots p_{\kappa-1}$ — размерность преобразования (для быстрых преобразований размерность всегда является составным числом). Использование методов топологической настройки БНС позволило реализовать быстрое преобразования Фурье с естественным упорядочением по частотам следования (факторизованное представление для $N = 2^3$ представлено на рис.16, множители $1/\sqrt{2}$ с целью упрощения не показаны, поворачивающий множитель определяется выражением $\omega = \exp(-j\frac{2\pi}{8})$.) Детали реализации БПФ приведены в [10]).

V_0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	
V_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
V_2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
U_2	U_1	U_0																						
0	0	0	1	1					1	1							1			1				
0	0	1		1	1				1	1							1			1				
0	1	0			1	1					1	1					1			1				
0	1	1				1	1					1	1				1			1				
1	0	0	1	-1					1	-1							1			-1				
1	0	1		1	-1				ω^2	$-\omega^2$						ω			$-\omega$					
1	1	0			1	-1					1	-1				ω^2			$-\omega^2$					
1	1	1				1	-1					ω^2	$-\omega^2$			ω^3			$-\omega^3$					

Рис. 16. Факторизованное представление алгоритма БПФ с естественным упорядочением по частотам следования

Быстрое вейвлет-преобразование

Пакетный вейвлет-базис на интервале длиной $N = p_0 p_1 \dots p_{\kappa-1}$ может быть задан выражением:

$$h(U, V) = \varphi_m(U_{\kappa-m}, V_{\kappa-m}) \delta(\tau_m, \langle U_{\kappa-1} U_{\kappa-2} \dots U_{\kappa-m+1} \rangle), \quad (30)$$

где $U = \langle U_{\kappa-1} U_{\kappa-2} \dots U_0 \rangle$ — номер временного отсчета, $V = \langle V_0 V_1 \dots V_{\kappa-1} \rangle$ — номер вейвлет-функции (предполагается, что вейвлет-функции располо-

жены вдоль столбцов базисной матрицы), $\varphi_m(U_{\varkappa-m}, V_{\varkappa-m})$ — набор ортогональных образующих импульсов в частотной локализации номера m , $\delta(\cdot)$ — функция Кронекера, τ_m — порядковый номер временной локализации вейвлет-функции в частотной локализации m . Множество функций вейвлет-базиса, принадлежащих одной частотной локализации в дальнейшем будем называть *поликадой*. Функции в пределах поликады обладают одними и теми же частотными свойствами, но отличаются позициями образующих импульсов. Упорядочим функции в каждой поликаде по временным локализациям следующим правилом:

$$\tau_m = \langle V_{\varkappa-1} V_{\varkappa-2} \dots V_{\varkappa-m+1} \rangle. \quad (31)$$

При таком упорядочении функцию Кронекера в выражении (30) можно разложить в произведение функций Кронекера, в результате получим:

$$h(U, V) = \varphi_m(U_{\varkappa-m}, V_{\varkappa-m}) \delta(U_{\varkappa-m+1}, V_{\varkappa-m+1}) \dots \\ \dots \delta(U_{\varkappa-2}, V_{\varkappa-2}) \delta(U_{\varkappa-1}, V_{\varkappa-1}).$$

Сопоставив данное выражение с формулой вычисления элементов матрицы перестраиваемого преобразования (24), нетрудно найти правило определения элементов ядер (детали реализации алгоритма можно найти в работе [10]).

Одним из простейших вейвлет-преобразований является базис Хаара. Базис строится на интервале длиной $N = 2^z$ и порождается двуполярным импульсом с временной базой, равной двум. В случае $N = 2^3$, базисные функции по частотным локализациям разбиваются на три октавы, Матрица образующих импульсов каждой октавы имеет вид:

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

а матрица преобразования факторизуется в произведение трех слабозаполненных матриц (см. рис. 17).

V_0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
V_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
V_2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
U_2	U_1	U_0																						
0	0	0	1	1				1	1						1	1								
0	0	1	1	-1					1	0						1	0							
0	1	0		1	1			1	-1									1	0					
0	1	1		1	-1				0	1										1	0			
1	0	0				1	1				1	1			1	-1								
1	0	1				1	-1					1	0			0	1							
1	1	0					1	1			1	-1					0	1						
1	1	1					1	-1				0	1					0	1					

Рис. 17. Факторизованное представление быстрого преобразования Хаара

Свернув факторизованное произведение, получим матрицу базиса в виде:

$$H = \begin{bmatrix} 1 & 1 & 1 & & 1 & & & \\ 1 & 1 & 1 & & -1 & & & \\ 1 & 1 & -1 & & & & 1 & \\ 1 & 1 & -1 & & & & -1 & \\ 1 & -1 & & 1 & & 1 & & \\ 1 & -1 & & 1 & & -1 & & \\ 1 & -1 & & -1 & & & & 1 \\ 1 & -1 & & -1 & & & & -1 \end{bmatrix}$$

Нейросетевая аппроксимация регулярных фракталов

Термин «фрактал» был введен *Бенуа Мандельбротом* в 1975 году для обозначения особого класса многомерных функций, обладающих свойством самоподобия и дробной размерности. Первоначально к фракталам относились как к математической экзотике, но за прошедшие десятилетия ситуация кардинальным образом изменилась. Оказалось, что окружающий нас

мир наполнен фракталами гораздо в большей степени, чем «привычными» нам объектами. Показательным примером являются, казалось бы, хорошо изученные динамические системы. Современные исследования показали, что в пространстве фазовых координат для нелинейных динамических систем характерно существование фрактальных притягивающих множеств (аттракторов) [19]. Фрактальные аттракторы связаны с так называемыми сценариями динамического хаоса [20], определяющего границы возможного прогнозирования поведения динамических систем.

В ряде работ [21, 22] было показано, что многослойные нейронные сети могут служить генераторами фрактальных структур и использоваться в этом качестве как инструмент моделирования и анализа нелинейных явлений. Вид фрактала накладывает определенные ограничения на структуру, топологию и параметры нейронной сети. В данном разделе будет показано использование БНС для генерации фрактальных структур.

Аналитическая форма регулярного фрактала

На рис. 18 показаны три итерации фрактала Кантора. На каждой итерации из непрерывной области удаляется средняя треть, что эквивалентно масштабной деформации исходной функции в отношении 1:3.

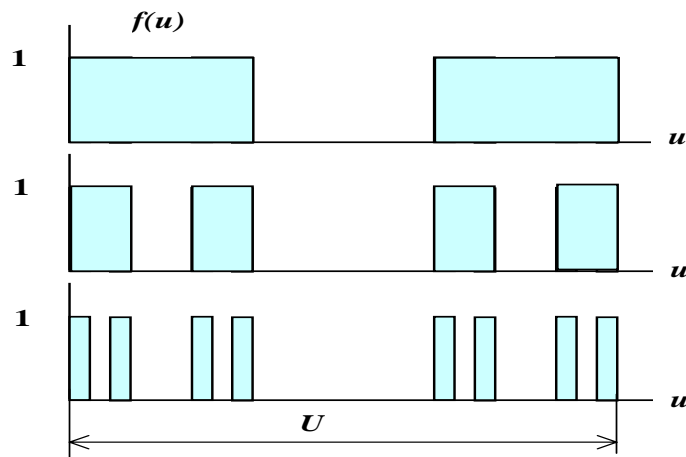


Рис. 18. Одномерный фрактал Кантора

Построим аналитическую форму вначале для фрактала Кантора, а затем обобщим ее на более широкий класс регулярных фракталов. Будем полагать, что фрактал Кантора задан на непрерывном интервале $U = [0, 1)$. Любую точку данного интервала $u \in U$ можно записать в троичной системе счисления в виде дроби:

$$u = 0, u_1 u_2 \dots u_n \dots, \quad (32)$$

где $u_i \in \{0, 1, 2\}$ — целые разрядные числа. Введем также непрерывные переменные $\tilde{u}_i \in [0, 3)$. Очевидно, что последовательность разрядных чисел (32) можно в любом месте оборвать, завершив ее непрерывной переменной, в результате точка интервала будет представлена в виде:

$$u = 0, u_1 u_2 \dots u_{n-1} \tilde{u}_n.$$

Разрядные числа u_i будем рассматривать как результат действия оператора выделяющего целую часть из непрерывной переменной \tilde{u}_i . Операцию масштабирования можно рассматривать как изменение правила представления точек интервала. Например, для базовой функции Кантора аргумент определяется правилом $u = \langle 0, \tilde{u}_1 \rangle = 3^{-1} \tilde{u}_1$. Для функции второй итерации: $u = \langle 0, u_1 \tilde{u}_2 \rangle = 3^{-1} u_1 + 3^{-2} \tilde{u}_2$. Для функции третьей итерации: $u = \langle 0, u_1 u_2 \tilde{u}_3 \rangle = 3^{-1} u_1 + 3^{-2} u_2 + 3^{-3} \tilde{u}_3$, и т. д. На непрерывном интервале $[0, 3)$ определим функцию:

$$\varphi(\tilde{u}) = \begin{cases} 1 & \text{для } \tilde{u} \in [0, 1), \\ 0 & \text{для } \tilde{u} \in [1, 2), \\ 1 & \text{для } \tilde{u} \in [2, 3) \end{cases}$$

Нетрудно видеть, что фрактал Кантора в аналитическом виде можно записать в виде бесконечного произведения:

$$f(u) = \varphi(\tilde{u}_1) \varphi(\tilde{u}_2) \varphi(\tilde{u}_3) \dots \varphi(\tilde{u}_n) \dots$$

Дискретная аппроксимация фракталов

В дискретном варианте фрактал строится на последовательности расширяющихся дискретных интервалах:

$$U_i = \{0, 1, 2, \dots, N_i - 1\}, \quad i = 0, 1, 2, \dots, \varkappa - 1 \dots$$

Точки интервалов определяются правилом $u = \langle u_{\kappa-1}u_{\kappa-2} \dots u_0 \rangle$. В общем случае разрядные переменные имеет различные локальные области определения: $u_i \in \{0, 1, \dots, p_i - 1\}$. На локальных интервалах определены кусочно-постоянные дискретные функции $\varphi_i(u_i)$. Аппроксимация фрактала записывается в виде конечного произведения:

$$f(u) \approx \varphi(u_0) \varphi(u_1) \dots \varphi(u_{\kappa-1}).$$

Например, для фрактала Кантора все дискретные образующие функции одинаковы и имеют вид:

$$\varphi(u) = \begin{cases} 1, & \text{для } u = 0, \\ 0, & \text{для } u = 1, \\ 1, & \text{для } u = 2. \end{cases}$$

Ковер Кантора. Двумерным вариантом фрактала Кантора являются плоская фигура, заданная на непрерывных интервалах $U = [0, 1)$, $V = [0, 1)$. На рис. 19 показана одна из итераций фрактала. Координаты точки в плоскости фигуры определяются бесконечными дробями:

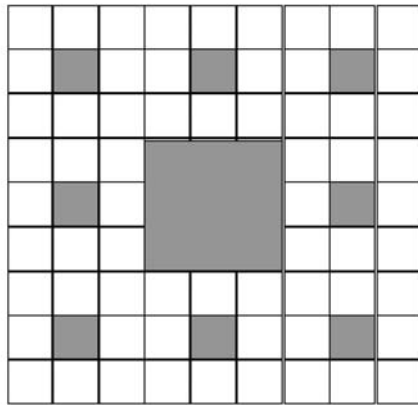


Рис. 19. Двумерный фрактал Кантора

$$u = 0, u_1 u_2 \dots u_n \dots; u_i \in \{0, 1, 2\},$$

$$v = 0, v_1 v_2 \dots v_n \dots; v_i \in \{0, 1, 2\}.$$

Обозначим через $\tilde{u}_i \in [0, 3)$ и $\tilde{v}_i \in [0, 3)$ соответствующие непрерывные переменные, тогда ковер Кантора в аналитической форме можно записать в виде бесконечного произведения:

$$f(u, v) = \varphi(\tilde{u}_1, \tilde{v}_1) \varphi(\tilde{u}_2, \tilde{v}_2) \varphi(\tilde{u}_3, \tilde{v}_3) \dots \varphi(\tilde{u}_n, \tilde{v}_n) \dots$$

Дискретная аппроксимация для ковра Кантора представляет собой конечное произведение:

$$f(u, v) \approx \varphi(u_0, v_0) \varphi(u_1, v_1) \varphi(u_2, v_2) \dots \varphi(u_{x-1}, v_{x-1}), \quad (33)$$

где все дискретные образующие функции имеют вид:

$$\varphi(u_i, v_i) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Нетрудно заметить, что полученное выражение по форме совпадает с представлением (24) для элементов матрицы H нейронной сети. Сравнивая (24) и (33), получим следующие правила определения параметров нейронных ядер:

$$w_{i\lambda}^\lambda(u_\lambda, v_\lambda) = \varphi(u_\lambda, v_\lambda).$$

На рис. 20 показана нейросетевая аппроксимация для ковра Кантора на интервале длиной $N = 3^2$. Аппроксимирующая нейронная сеть имеет два слоя, каждой матрице сомножителю отвечает один нейронный слой. Все непоказанные элементы матриц равны нулю. Из рисунка видно, что результирующая матрица подобна ковра Кантора.

Фрактал Серпинского. На рис. 21 показана одна из итераций двумерного фрактала, называемого «салфеткой Серпинского». Фрактал задан на непрерывных интервалах $U = [0, 1)$, $V = [0, 1)$. Дискретная аппроксимация фрактала Серпинского определяется на интервале длиной $N = 2^x$ выражением (33), где образующая функция имеет вид

$$\varphi(u_i, v_i) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

На рис. 22 показана нейросетевая аппроксимация для салфетки Серпинского на интервалах длиной $N = 2^3$. Аппроксимирующая нейронная сеть имеет три слоя, каждой матрице сомножителю отвечает один нейронный слой. Из рисунка видно, что результирующая матрица подобна непрерывной форме данного фрактала.

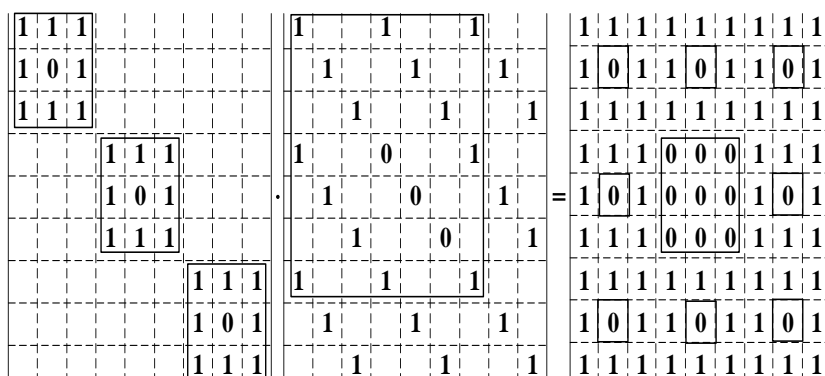


Рис. 20. Нейросетевая аппроксимация двумерного фрактала Кантора

Фрактальная фильтрация сигналов

Хорошо известны трудности связанные с выделением полезной информации из сигналов, обладающих частотно-локальными свойствами. Характерным примером являются сигналы, отражающие динамику турбулентных потоков, порожденных нелинейно взаимодействующими процессами в широких диапазонах пространственных частот и временных локализаций. Для анализа подобных сигналов в настоящее время широко используется вейвлет-преобразование, поскольку его элементы хорошо локализованы и обладают подвижным частотно-временным окном [23]. В основе вейвлет-анализа лежит представление сигналов в виде суперпозиции масштабных преобразований и сдвигов образующих импульсов. Самоподобие вейвлет-функций при масштабных преобразованиях является отличительным свойством фрактальных последовательностей, поэтому вейвлет-преобразование можно считать одним из представителей фрактальных методов обработки данных.

Принцип самоподобия можно также использовать при построении фрактальных фильтров. В обобщенном понимании фильтрация это процедура, реализующая проекцию сигнала, заданного в функциональном пространстве, в пространство меньшей размерности. В отличие от вейвлет-преобразования фрактальная фильтрация необратима и связана с потерей информации. Цель фрактальной фильтрации заключается в том, чтобы

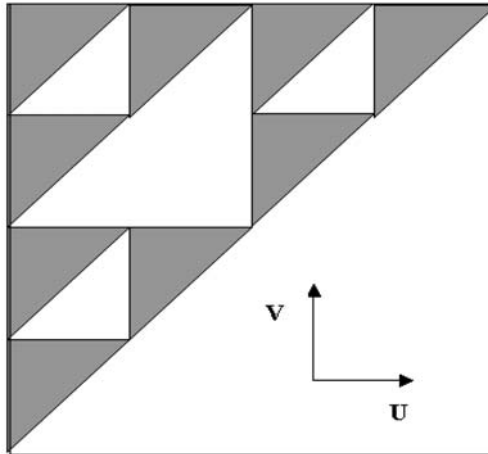


Рис. 21. Двумерный фрактал Серпинского

выявить пространственно распределенные свойства изучаемого объекта в кратных временных масштабах.

Фильтрация дискретных сигналов

Рассмотрим сигнал, определенный функцией $f(u)$, заданной на дискретном интервале длиной $N = p_0 p_1 \dots p_{\kappa-1}$. Представим аргумент этой функции в позиционной многоосновной системе счисления с основаниями $p_0, p_1, \dots, p_{\kappa-1}$:

$$u = \langle u_{\kappa-1} u_{\kappa-2} \dots u_0 \rangle = u_{\kappa-1} p_{\kappa-2} p_{\kappa-3} \dots p_0 + \\ + u_{\kappa-2} p_{\kappa-3} p_{\kappa-4} \dots p_0 + \dots + u_1 p_0 + u_0,$$

где $u_i \in [0, 1, \dots, p_i - 1]$ — разрядные переменные. В результате данного преобразования сигнал представляется как многомерная функция вида $f \langle u_{\kappa-1} u_{\kappa-2} \dots u_0 \rangle$, где каждый аргумент функции определяет некоторый масштабный срез сигнала. Зафиксируем все аргументы функции кроме u_m . Варьируя свободный аргумент u_m , получим выборку S_m (с числом элементов p_m). *Фрактальным фильтром* частотной локализации m будем называть произвольный функционал $F(S_m)$, определенный на выборке S_m .

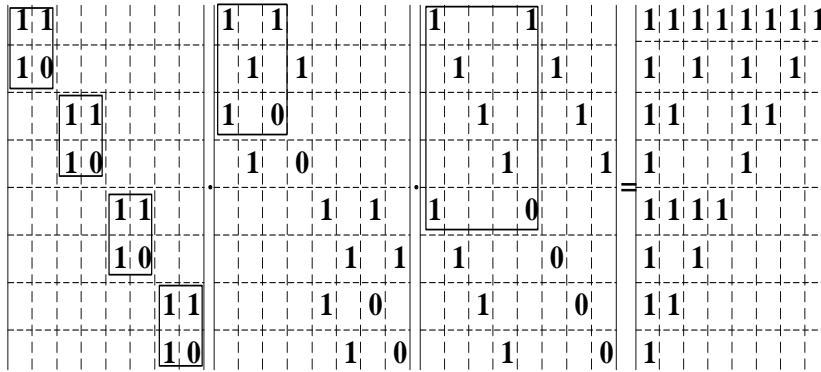


Рис. 22. Нейросетевая аппроксимация двумерного фрактала Серпинского

Операцию фрактальной фильтрации можно записать в виде в виде:

$$f_{out} \langle u_{\kappa-1} u_{\kappa-2} \dots u_{m+1} u_{m-1} \dots u_0 \rangle = F_{u_m} (f_{inp} \langle u_{\kappa-1} u_{\kappa-2} \dots u_0 \rangle).$$

Если $m = 0$, то фильтр будем называть *фильтром нижних фрактальностей*, такой фильтр генерализует сигнал, сглаживая мелкие детали. Если $m = \kappa - 1$, то фильтр назовем *фильтром верхних фрактальностей*, такой фильтр сохраняет детали и нивелирует трендовые изменения. Фильтр с $0 < m < \kappa - 1$ можно назвать *препарирующим*. Фрактальный фильтр выполняет проекцию функционального пространства размерности N в подпространство размерности N/p_m , однако в ряде случаев результат фильтрации удобно представлять на исходном интервале $U = [0, 1, \dots, N - 1]$. Это можно сделать, положив равными значения функции в точках интервала, различающихся только по значению аргумента u_m . Полученная функция может быть вновь подвержена фрактальной фильтрации в некоторой частотной локальности $n \neq m$. Последовательная фрактальная фильтрация, реализуется принцип иерархической многомасштабной обработки данных.

Типы фрактальных фильтров

Линейные фильтры — могут быть реализованы произвольными линейными функционалами. Примерами могут служить следующие типы:

- *Фильтр средних:*

$$\begin{aligned} f_{out} \langle u_{x-1} u_{x-2} \dots u_{m+1} u_{m-1} \dots u_0 \rangle &= \\ &= \frac{1}{p_m} \sum_{u_m} f_{inp} \langle u_{x-1} u_{x-2} \dots u_0 \rangle; \end{aligned}$$

- *Унарный линейный фильтр:*

$$\begin{aligned} f_{out} \langle u_{x-1} u_{x-2} \dots u_{m+1} u_{m-1} \dots u_0 \rangle &= \\ &= f_{inp} \langle u_{x-1} u_{x-2} \dots u_{m+1} t_m u_{m-1} \dots u_0 \rangle, \quad t_m = \text{const}; \end{aligned}$$

- *Аффинный фильтр:*

$$\begin{aligned} f_{out} \langle u_{x-1} u_{x-2} \dots u_{m+1} u_{m-1} \dots u_0 \rangle &= \\ &= \sum_{u_m} f_{inp} \langle u_{x-1} u_{x-2} \dots u_0 \rangle w(u_m) + w_0. \end{aligned}$$

Аффинный фильтр принято условно относить к классу линейных.

Нелинейные фильтры. Нелинейный функционал можно задать в алгоритмическом или аналитическом виде. Примерами алгоритмически заданных функционалов являются ранговые фильтры. При ранговой фильтрации выборка ранжируется. Поскольку длина выборки равна p_m то число возможных рангов не превышает p_m . Значением функционала является выборочное значение, соответствующее рангу $r \leq p_m$. Конкретными примерами могут служить следующие типы фильтров:

$$\begin{aligned} f_{out} \langle u_{x-1} u_{x-2} \dots u_{m+1} u_{m-1} \dots u_0 \rangle &= \min_{u_m} f_{inp} \langle u_{x-1} u_{x-2} \dots u_0 \rangle, \\ f_{out} \langle u_{x-1} u_{x-2} \dots u_{m+1} u_{m-1} \dots u_0 \rangle &= \max_{u_m} f_{inp} \langle u_{x-1} u_{x-2} \dots u_0 \rangle. \end{aligned}$$

Первый из них соответствует значению ранга $r = 0$, а второй — значению $r = p_m$. Примером нелинейных фильтров с аналитически заданными функционалами являются нормирующие фильтры, для которых функционал определяется той или иной векторной нормой, конкретными примерами могут служить следующие типы:

- *Фильтр Евклида:*

$$f_{out} \langle u_{\kappa-1} u_{\kappa-2} \dots u_{m+1} u_{m-1} \dots u_0 \rangle = \sqrt{\sum_{u_m} f_{inp} \langle u_{\kappa-1} u_{\kappa-2} \dots u_0 \rangle^2};$$

- *Фильтр Хэмминга:*

$$f_{out} \langle u_{\kappa-1} u_{\kappa-2} \dots u_{m+1} u_{m-1} \dots u_0 \rangle = \sum_{u_m} |f_{inp} \langle u_{\kappa-1} u_{\kappa-2} \dots u_0 \rangle|.$$

Многомерные фрактальные фильтры. Многомерные фильтры реализуются функционалами, заданными на многомерной выборке. Например, двумерный фрактальный фильтр в общем случае может быть определен в виде:

$$f_{out} \langle u_{\kappa-1} u_{\kappa-2} \dots u_0 \rangle = F_{u_m, u_n} (f_{inp} \langle u_{\kappa-1} u_{\kappa-2} \dots u_0 \rangle).$$

Такой фильтр выполняет проекцию сигнала из пространства размерности N в пространство размерности $N/(p_m p_n)$.

Фильтрация непрерывных сигналов

Пусть сигнал $f(u)$ определен на непрерывном интервале $U = [0, 1)$. Зададим позиционную систему счисления множеством оснований p_1, p_2, p_3, \dots , и представим аргумент функции в виде бесконечной дроби:

$$u = \langle 0, u_1 u_2 u_3 \dots \rangle = u_1 p_1^{-1} + u_2 p_2^{-1} p_1^{-1} + u_3 p_3^{-1} p_2^{-1} p_1^{-1} + \dots$$

Непрерывный фрактальный фильтр частотной локализации m определяется функционалом:

$$f_{out} \langle 0, u_1 u_2 \dots u_{m-1} u_{m+1} \dots \rangle = F_{u_m} (f_{inp} \langle 0, u_1 u_2 \dots u_m \dots \rangle).$$

В частности, фильтр верхних фрактальностей будет иметь вид:

$$f_{out} \langle 0, u_2 u_3 u_4 \dots \rangle = F_{u_1} (f_{inp} \langle 0, u_1 u_2 u_3 \dots \rangle),$$

а многомерный фильтр нижних частот можно задать функционалом:

$$f_{out} \langle 0, u_1 u_2 u_3 u_4 \dots u_m \rangle = \underset{u_{m+1}, u_{m+2}, \dots}{F} (f_{inp} \langle 0, u_1 u_2 u_3 \dots \rangle).$$

Для непрерывных фильтров выходной сигнал автоматически масштабируется в исходный интервал $U = [0, 1)$.

Приспособленные быстрые преобразования

В данном разделе будет рассмотрено использование фрактальной фильтрации для настройки множественных адаптивных фильтров реализованных на базе БНС. Наличие быстрого алгоритма позволяет использовать данные реализации при обработке сигналов высокой размерности в системах реального времени.

Методы цифровой адаптивной фильтрации (такие как согласованная фильтрация, инверсная фильтрация, фильтр Винера и т. п. [24]) используют априорную информацию о сигнале или помехе для настройки коэффициентов фильтра. Подобные фильтры позволяют получить максимальное отношение «сигнал/шум» при фильтрации опорных эталонных сигналов. Поэтому адаптивные фильтры находят широкое применение в технических приложениях.

Большой класс адаптивных фильтров представляют собой линейные преобразования среди которых можно выделить подкласс ортогональных преобразований. Перестраиваемые ортогональные преобразования, адаптируемые к одиночной функции получили название приспособленных преобразований [5]. Это название в дальнейшем будет использовано по отношению к любым линейным перестраиваемым преобразованиям, настраиваемых по эталонным функциям.

Алгоритм приспособления

Линейное преобразование будем называть приспособленным к функции, если один из столбцов матрицы преобразования совпадает с заданной функцией. Подобным образом можно определить приспособленность к нескольким функциям (в том случае, когда используется правое умножение матрицы на вектор, приспособление более удобно выполнять к строкам

матрицы). Идея метода настройки БНС к одной или нескольким функциям основана на представлении каждой функции заданного набора в виде фрактального произведения, отвечающем структуре БНС. Для одной функции задача фрактальной аппроксимации формулируется следующим образом.

Пусть на дискретном интервале длиной $N = p_0 p_1 \dots p_{\varkappa-1}$ задана вещественная функция $f(u)$. Ставится задача найти представление функции в виде произведения:

$$f(u) = \varphi_{i^0}(u_0) \varphi_{i^1}(u_1) \dots \varphi_{i^{\varkappa-1}}(u_{\varkappa-1}). \quad (34)$$

Правило выбора образующих функций для каждого шага фрактальной итерации подчинено условию $i^m = \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_{m+1} \rangle$. Поскольку элементы матрицы БНС представляются через элементы ядер:

$$h(U, V) = w_{i^0}^0(u_0, v_0) w_{i^1}^1(u_1, v_1) \dots w_{i^{\varkappa-1}}^{\varkappa-1}(u_{\varkappa-1}, v_{\varkappa-1}), \quad (35)$$

где $i^m = \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_{m+1} v_{m-1} v_{m-2} \dots v_0 \rangle$, то столбец с номером $V = \langle v_{\varkappa-1} v_{\varkappa-2} \dots v_0 \rangle$ матрицы H будет совпадать с заданной функцией, если элементы ядер совпадают с множителями фрактального разложения (34). Так как для нулевого слоя номер ядра $i^0 = \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_1 \rangle$ не зависит от переменных v , то, возможно, одновременно приспособить БНС к такому количеству функций, сколько значений может принимать переменная v_0 . Для нахождения фрактального разложения (34) введем последовательность вспомогательных функций $f_0, f_1, \dots, f_{\varkappa-1}$, определив их рекуррентным правилом:

$$\begin{aligned} f(u) &= f_0 \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_0 \rangle, \\ f_0 \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_0 \rangle &= f_1 \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_1 \rangle \varphi_{i^0}(u_0), \\ f_1 \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_1 \rangle &= f_2 \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_2 \rangle \varphi_{i^1}(u_1), \\ &\vdots \\ f_{\varkappa-2} \langle u_{\varkappa-1} u_{\varkappa-2} \rangle &= f_{\varkappa-1} \langle u_{\varkappa-1} \rangle \varphi_{i^{\varkappa-2}}(u_{\varkappa-2}), \\ f_{\varkappa-1} \langle u_{\varkappa-1} \rangle &= \varphi_{i^{\varkappa-1}}(u_{\varkappa-1}). \end{aligned} \quad (36)$$

Из этих соотношений непосредственно следует:

$$\varphi_{i^m}(u_m) = \frac{f_m \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_m \rangle}{f_{m+1} \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_{m+1} \rangle}, \quad m = 0, 1, \dots, \varkappa - 2. \quad (37)$$

Анализируя (36) нетрудно заметить, что последовательность вспомогательных функций представляют собой выходы цепочки фильтров нижних

фрактальностей (см. рис. 23) Для построения вспомогательных функций можно использовать любой метод фрактальной фильтрации. Это означает, что задача построения приспособленного преобразования имеет множество возможных решений.

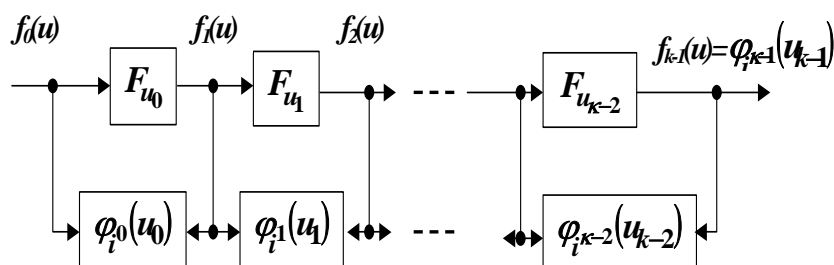


Рис. 23. Схема расчета аппроксимирующих компонент

Положительным аспектом множественности является простое разрешение проблемы цензурирования нулей для знаменателя выражения (37). Нулевое значение знаменателя всегда можно заменить на любое удобное ненулевое (например, единичное). Это ведет к локальному изменению фрактального фильтра, но не влияет на результат аппроксимации.

Множественное приспособление. Если область возможных значений разрядной переменной v_0 соответствует интервалу $[0, g_0 - 1]$, то с помощью выше изложенного алгоритма можно настроить преобразование точно на g_0 функций. При этом остаются элементы ядер, которые могут быть выбраны произвольно.

На рис. 24 показано факторизованное представление быстрого преобразования приспособленного к двум функциям, размещенным в двух первых столбцах матрицы. Символом «*» выделены элементы ядер, которые выбираются произвольно. Не задействованные степени свободы можно использовать для расширения опорной базы приспособления. Это можно реализовать за счет использования фрактально фильтрованных образов добавочных функций. Рассмотрим алгоритм подобного приспособления.

Пусть F — это система из $M = g_0 g_1 \dots g_{k-1}$ функций заданных на интервале длиной $N = p_0 p_1 \dots p_{k-1}$. Для аппроксимации функций системы F будем использовать БНС, реализующую перестраиваемое линейное преобразование размерности $N \times M$.

v_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
v_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
v_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
u_2	u_1	u_0																						
0	0	0	1	1					1	*							1			*				
0	0	1	1	1					1	*							1			*				
0	1	0		1	1				1	*								*			*			
0	1	1		1	1				1	*								*				*		
1	0	0			1	1					1	*					1			*				
1	0	1			1	1						1	*				1			*				
1	1	0				1	1					1	*					*			*			
1	1	1				1	1						1	*					*			*		

Рис. 24. Факторизованное преобразование приспособленное к одной функции

Пусть $x = \langle x_{\mathcal{N}-1} x_{\mathcal{N}-2} \dots x_0 \rangle$ — некоторый базовый столбец матрицы преобразования, который должен быть точно приспособлен к соответствующей функции системы. Тогда все столбцы $x = \langle x_{\mathcal{N}-1} x_{\mathcal{N}-2} \dots x_1 v_0 \rangle$, где v_0 варьируется в интервале $[0, g_0 - 1]$, также будут точно приспособлены к соответствующим функциям опорной системы. В алгоритме множественного приспособления настройка БНС выполняется путем последовательной фрактальной фильтрации всего массива функций одновременно, при этом на каждом шаге строится опорное множество функций, фрактальные множители которых используются для заполнения нейронных ядер БНС. Опорные множества функций вложены друг в друга и последовательно окутывают базовый столбец. Правило формирования окутывающей последовательности в зависимости от номера итерации (номера слоя) показано в табл. 3.

Сравнивая правило формирования опорных множеств с правилом нумерации нейронных ядер (последний столбец таблицы 3), можно сделать вывод, что в результате последовательного выполнения алгоритма все ядра преобразования будут заполнены полностью. Перед выполнением алго-

Таблица 3. Правило формирования опорных множеств аппроксимируемых функций

Номер слоя m	Опорное множество функций	Размер опорного множества	Нумерация нейронных ядер i^m
0	$\langle x_{\varkappa-1} x_{\varkappa-2} \dots x_1 v_0 \rangle$	g_0	$\langle u_{\varkappa-1} u_{\varkappa-2} \dots u_1 \rangle$
1	$\langle x_{\varkappa-1} x_{\varkappa-2} \dots x_2 v_1 v_0 \rangle$	$g_1 g_0$	$\langle u_{\varkappa-1} u_{\varkappa-2} \dots u_2 v_0 \rangle$
2	$\langle x_{\varkappa-1} x_{\varkappa-2} \dots x_3 v_2 v_1 v_0 \rangle$	$g_2 g_1 g_0$	$\langle u_{\varkappa-1} u_{\varkappa-2} \dots u_3 v_1 v_0 \rangle$
...
$\varkappa - 2$	$\langle x_{\varkappa-1} v_{\varkappa-2} \dots v_1 v_0 \rangle$	$g_{\varkappa-2} \dots g_1 g_0$	$\langle u_{\varkappa-1} v_{\varkappa-3} \dots v_1 v_0 \rangle$
$\varkappa - 1$	$\langle v_{\varkappa-1} v_{\varkappa-2} \dots v_1 v_0 \rangle$	$g_{\varkappa-1} \dots g_1 g_0$	$\langle v_{\varkappa-2} \dots v_1 v_0 \rangle$

ритма, функции опорной системы упорядочиваются по требуемой глубине приспособления. Функции с номером базового столбца и ближайшие к ней, составляющие опорное множество для $m = 0$ будут точно приспособлены, функции опорного множества для $m = 1$ будут приспособлены по одному фильтрованному образам, функции опорного множества для $m = 2$ — будут приспособлены по дважды фильтрованным образам и т. д.

Приспособленные преобразования в нечетком пространстве

Реализация приспособленных преобразований в нечетком пространстве имеет по крайней мере два преимущества: во-первых, обеспечивается высокая вычислительная эффективность — поскольку логические операции выполняются существенно быстрее арифметических и, во-вторых, отсутствует необходимость цензурирования нулей на выходе фрактальных фильтров — поскольку при вычислении фрактальных компонент функции не используется операция деления.

Геометрия нечеткого пространства определяется операциями \min и \max , действующих на ограниченном отрезке $[0, 1]$ множества вещественных чисел. Пусть $f(u)$ — произвольная вещественная функция, заданная на интервале длиной $N = p_0 p_1 \dots p_{\varkappa-1}$. В нечетком пространстве область возможных значений функции принадлежит диапазону $[0, 1]$. С целью упрощения записи для операций \min и \max далее будем использовать обозначения « \circ » и « \oplus », соответственно, рассматривая их как логическое умножение и логическое сложение.

Ставится задача построить быстрое преобразование в нечетком пространстве, приспособленное к заданной функции. Будем использовать для этой цели БНС, в которой арифметические операции сложения и умножения вещественных чисел заменены соответствующими логическими операциями. Задача сводится к нахождению для функции $f(u)$ фрактального логического разложения:

$$f(u) = f \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_0 \rangle = \varphi_{i^0}(u_0) \circ \varphi_{i^1}(u_1) \circ \dots \circ \varphi_{i^{\varkappa-1}}(u_{\varkappa-1}),$$

где $i^m = \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_{m+1} \rangle$ — позиционный номер функции-компоненты масштабного уровня m . Функции сомножители можно определить, рекуррентно раскладывая исходную функцию в последовательность логических произведений:

$$\begin{aligned} f(u) &= f_0 \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_0 \rangle, \\ f_0 \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_0 \rangle &= f_1 \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_1 \rangle \circ \varphi_{i^0}(u_0), \\ f_1 \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_1 \rangle &= f_2 \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_2 \rangle \circ \varphi_{i^1}(u_1), \\ &\vdots \\ f_{\varkappa-2} \langle u_{\varkappa-1} u_{\varkappa-2} \rangle &= f_{\varkappa-1} \langle u_{\varkappa-1} \rangle \circ \varphi_{i^{\varkappa-2}}(u_{\varkappa-2}), \\ f_{\varkappa-1} \langle u_{\varkappa-1} \rangle &= \varphi_{i^{\varkappa-1}}(u_{\varkappa-1}). \end{aligned} \quad (38)$$

Существует определенная свобода в выборе вспомогательных функций $f_1, \dots, f_{\varkappa-1}$. Оставаясь в рамках нечеткого пространства, будем использовать для построения вспомогательных функций ранговый фрактальный фильтр следующего вида:

$$f_{m+1} \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_{m+1} \rangle = \max_{u_m} (f_m \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_m \rangle).$$

В этом случае из (38) следует, что функции-компоненты будут определяться выражением:

$$\varphi_{i^m}(u_m) = f_{m+1} \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_{m+1} \rangle \circ f_m \langle u_{\varkappa-1} u_{\varkappa-2} \dots u_m \rangle.$$

Дальнейшая процедура настройки БНС не отличается от аналогичной процедуры арифметического пространства. Преобразование, приспособленное к одной функции, имеет свободные элементы нейронных ядер, которые могут быть заполнены при добавлении опорных функций, подобно тому, как это было сделано для преобразований в арифметическом пространстве.

При обработке данных с помощью нейронной сети выполняются операции « \circ » и « \oplus ». Если входной образ совпадает с опорной функцией приспособленного преобразования, то на соответствующем выходе сети будет получено максимальное значение. На этом может быть основано классификационное правило распознавания сигналов.

В нечетком пространстве операции « \circ » и « \oplus » полностью симметричны, поэтому существует двойственная форма фрактального представления функций, которая может быть задана выражениями:

$$\begin{aligned} f(u) &= f\langle u_{x-1}u_{x-2} \dots u_0 \rangle = \varphi_{i^0}(u_0) \oplus \varphi_{i^1}(u_1) \oplus \dots \oplus \varphi_{i^{x-1}}(u_{x-1}) \\ f_{m+1}\langle u_{x-1}u_{x-2} \dots u_{m+1} \rangle &= \min_{u_m} (f_m\langle u_{x-1}u_{x-2} \dots u_m \rangle), \\ \varphi_{i^m}(u_m) &= f_{m+1}\langle u_{x-1}u_{x-2} \dots u_{m+1} \rangle \oplus f_m\langle u_{x-1}u_{x-2} \dots u_m \rangle. \end{aligned}$$

Спектральные приспособленные преобразования

Для задач классификации и распознавания сигналов существенное значение имеют процедуры предварительной обработки, ориентированные на устранение избыточности и выделении информативных признаков. Использование спектральных преобразований для этих целей позволяет представить информацию, содержащуюся в исходном сигнале в виде взаимно-независимых ортогональных составляющих. Поскольку энергия сигнала определяется суммой квадратов спектральных коэффициентов, то по величине спектрального коэффициента можно непосредственно судить о значимости информативного признака. Известно, что максимальное сокращение избыточности данных обеспечивается ортогональным преобразованием Карунена-Лоэва, которое образуется собственными векторами ковариационной матрицы сигнальной группы. Однако использование данного преобразования сопряжено со значительными вычислительными затратами. По этой причине метод Карунена-Лоэва не применяется для обработки данных высокой размерности. Если при построении преобразования Карунена-Лоэва ограничиться только одним собственным вектором, имеющим максимальный ранг (главной компонентой), то объем вычислений резко сокращается. Достаточно часто главную компоненту можно выделить из сигнала непосредственно, как опорный образ, относительно которого наблюдаются все сигнальные вариации. Ортогональное преобразование, которое настроено на одну главную компоненту относится к классу приспособленных. В силу ортогональности результатом воздействия приспособленного преобразования на опорный образ является выходной вектор,

у которого одна из координат равна единице, в то время как все остальные координаты равны нулю. Таким образом, приспособленное ортогональное преобразование идеально подходит для селективного распознавания эталонного сигнала. Построение быстрых приспособленных ортогональных преобразований для сигналов определенных на интервале с длиной кратной степени двойки рассматривалось в работе [5]. Ниже будет показано использование БНС для синтеза приспособленного ортогонального преобразования любой составной размерности.

Достаточные условия приспособленности. Для спектральных преобразований удобно использовать конструктивное определение приспособленности, которое можно сформулировать следующим образом. Ортогональное преобразование, заданное матрицей $h(u, v)$, приспособлено к функции $f(u)$ по столбцу, x если выполнены условия:

$$\begin{cases} \sum_u h(u, v) f(u) = 1, & \text{при } v = x, \\ \sum_u h(u, v) f(u) = 0, & \text{при } v \neq x. \end{cases} \quad (39)$$

Здесь предполагается, что опорная функция нормирована условием

$$\sum_u f^2(u) = 1.$$

Подставив в левую часть (39) выражения (34) и (35), после преобразований получим:

$$\begin{aligned} \sum_u h(u, v) f(u) &= \sum_{u_0} w_{i^0}(u_0, v_0) \varphi_{i^0}(u_0) \sum_{u_1} w_{i^1}(u_1, v_1) \varphi_{i^1}(u_1) \times \dots \\ &\dots \times \sum_{u_{\varkappa-1}} w_{i^{\varkappa-1}}(u_{\varkappa-1}, v_{\varkappa-1}) \varphi_{i^{\varkappa-1}}(u_{\varkappa-1}). \end{aligned}$$

Из данной формулы следует, что приспособленность спектрального преобразования обеспечивается, когда нейронные ядра любого слоя m приспособлены к соответствующим компонентам фрактального разложения опорной функции. Иначе это достаточное условие можно записать в виде системы уравнений:

$$\begin{cases} \sum_{u_m} w_{i^m}(u_m, v_m) \varphi_{i^m}(u_m) = 1, & \text{при } v_m = x_m, \\ \sum_{u_m} w_{i^m}(u_m, v_m) \varphi_{i^m}(u_m) = 0, & \text{при } v_m \neq x_m. \end{cases}$$

Таким образом, задача настройки БНС сводится к построению приспособленных ортогональных ядер. Рассмотрим возможный алгоритм такого построения.

Алгоритм построения приспособленных ортогональных ядер. Пусть компонента фрактального разложения представлена вектором вида $\varphi = (\varphi_0 \varphi_1 \dots \varphi_{p-1})$, размерности p . Разместим этот вектор в первом столбце ортогональной матрицы. Понятно, что существует множество способов достроить остальные столбцы матрицы. Алгоритм построения во многом определяется выбранной топологией матрицы. Поскольку никаких других требований на ортогональную матрицу не накладывается, то при выборе ее топологии целесообразно руководствоваться принципом минимальной длины описания (иначе именуемым «бритья Оккама»).

Вектор-образ может содержать нулевые значения, не теряя общности, можно полагать, что все нулевые значения размещены в конце вектора начиная с позиции $k + 1$. Предлагаемая топология матрицы показана на рис. 25. В данной матрице y_1, y_2, \dots, y_k неизвестные параметры, определяемые из условий ортогональности столбцов:

$$\begin{pmatrix} \varphi_0 & \varphi_0 & \varphi_0 & \dots & \varphi_0 & \varphi_0 & 0 & 0 & 0 & 0 \\ \varphi_1 & \varphi_1 & \varphi_1 & \dots & \varphi_1 & y_1 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & y_2 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \varphi_{k-2} & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \varphi_{k-1} & y_{k-1} & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ \varphi_k & y_k & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Рис. 25. Топология матрицы приспособленного ядра

$$y_\alpha \varphi_\alpha + \sum_{u=0}^{\alpha-1} \varphi_u^2 = 0.$$

Для построения ортогонального ядра W достаточно в построенной матрице произвести нормировку столбцов, так чтобы выполнялись условия:

$$\sum_{u=0}^{p-1} w^2(u, v) = 1.$$

Множественное приспособление. Ортогональное приспособленное преобразование не полностью использует степени свободы БНС. Если точка приспособления совпадает с первым столбцом результирующей матрицы, то будут настроены только ядра

$$i^m = \langle u_{\neq-1} u_{\neq-2} \dots u_{m+1} 0_{m-1} 0_{m-2} \dots 0_0 \rangle.$$

На рис. 26 приведено факторизованное представление ортогонального преобразования приспособленного к одной функции размещенной в первом столбце. Символом «*» отмечены элементы незаполненных ядер.

Для того чтобы преобразование было ортогональным, достаточно заполнить свободные ядра элементами любых ортогональных матриц соответствующей размерности. В практических алгоритмах, можно либо заполнить свободные ядра единичными матрицами вида:

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

либо повторить ядра, уже найденные для текущего слоя. В первом случае будут получены вейвлет-подобные ортогональные преобразования с явно выраженными свойствами пространственной локализации, во втором случае базисные функции будут фрактально близки друг к другу (т. е. их фильтрованные образы могут совпадать на некоторых масштабных уровнях). Использовать более глубокое приспособление, например, к фильтрованным образам опорных функций (как это было сделано в предыдущем параграфе) возможно только тогда, когда опорные функции фрактально ортогональны между собой (т. е. ортогональны на всех масштабных уровнях). В общем случае условие фрактальной ортогональности не выполняется.

v_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1		
v_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1		
v_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1		
u_2	u_1	u_0																								
0	0	0	1	1					1	1											1					
0	0	1	1	1					*	*										*						
0	1	0		1	1				1	1									*					*		
0	1	1		1	1				*	*									*				*			
1	0	0			1	1					1	1							1			1				
1	0	1			1	1						*	*					*			*					
1	1	0				1	1				1	1					*			*						
1	1	1				1	1					*	*				*			*						

Рис. 26. Топология ортогонального преобразования, приспособленного к одной функции

Быстрые нейронные сети в квантовых вычислениях

Квантовые вычисления (вычисления, основанные на методах квантовой механики) были инициированы в начале 80-х годов работами *Ричарда Фейнмана* [25, 26]. В 1994 году *Питер Шор* показал, что NP-сложная задача — разложение целых чисел на множители — может быть решена на квантовом компьютере за полиномиальное время [27]. Это результат дал толчок к бурному развитию квантовых вычислений. В августе 2000 года агентством Reuters было объявлено о первом практическом успехе: «IBM says it develops most advanced quantum computer («фирма IBM разработала первый квантовый компьютер»)). В данной разработке впервые был реализован квантовый регистр на 5 атомах. Практически была доказана принципиальная возможность реализации квантовых компьютеров.

В классическом компьютере регистр длиной n имеет конечное множество состояний равное 2^n , поскольку каждый бит имеет только два состояния. В квантовом компьютере функциональным аналогом бита является q -бит, который ассоциируется с двумерным комплексным пространством

и поэтому имеет континуум возможных состояний. В пространстве q -бита выделены два базисных вектора $|0\rangle$ и $|1\rangle$. Кроме базисных состояний для q -бита возможны также состояния, представляющие собой любые линейные комбинации базисных векторов: $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, где $\alpha^2 + \beta^2 = 1$. Квантовый регистр представляет собой конечный набор q -битов. Базисными состояниями квантового регистра являются тензорные произведения базисных состояний входящих в него q -бит. Для обозначения тензорных произведений в квантовой механике используются сокращения:

$$|x_{n-1}\rangle \otimes |x_{n-2}\rangle \otimes \dots \otimes |x_0\rangle = |x_{n-1}x_{n-2}\dots x_0\rangle.$$

Согласно общим принципам квантовой механики, возможным состоянием квантового регистра длиной n может быть любой вектор единичной длины в комплексном пространстве размерности 2^n . Квантовое состояние регистра записывается в виде:

$$|\varphi\rangle = \sum_{|x\rangle \in |x_{n-1}x_{n-2}\dots x_0\rangle} a_x |x\rangle,$$

где суммирование производится по всем возможным базисным состояниям. Для вектора единичной длины сумма квадратов коэффициентов a_x всегда равна единице. Квантовые состояния в комплексном пространстве задаются с точностью до фазового множителя $e^{j\phi}$. Квадрат значения каждого коэффициента a_x интерпретируется как вероятность нахождения квантового регистра в данном базовом состоянии. Состояние квантового регистра называется запутанным, если его не возможно представить в виде произведения квантовых состояний входящих в него q -бит.

Эволюция состояний квантового регистра во времени описывается уравнением Шредингера, решением которого являются унитарные преобразования размерности 2^n . При квантовых вычислениях в q -регистре готовится начальное состояние в виде базовых состояний q -бит. Затем используется некоторое унитарное преобразование над отдельными q -битами, зависящее от решаемой задачи, в итоге возникает новое состояние квантового регистра. Результатом вычислительной операции считается состояние квантового регистра, которое имеет вероятность достаточно близкую к единице. Эффективность квантовых вычислений достигается за счет того, что одновременно с унитарным преобразованием над состояниями q -бита «бесплатно» выполняется тензорное произведение, экспоненциально расширяющее область действия унитарного преобразования. Наибольший

эффект достигается для факторизуемых квантовых состояниях, однако и для широкого класса запутанных состояний, возможна реализация полиномиально эффективных алгоритмов.

Элементарным квантовой операция на регистре определяется двумерным унитарным преобразованием, действующим в пространстве одного q -бита. Данное преобразование (унарный вентиль) задается комплекснозначной матрицей размером 2×2 :

$$W = \begin{array}{c|cc} & |0\rangle & |1\rangle \\ \hline |0\rangle & w_{00} & w_{01} \\ |1\rangle & w_{10} & w_{11} \end{array}$$

В канонической форме унитарная матрица второго порядка может быть записана в виде:

$$W = \begin{pmatrix} e^{j\gamma_{00}} \cos \theta & e^{j\gamma_{01}} \sin \theta \\ e^{j\gamma_{10}} \sin \theta & -e^{j\gamma_{11}} \cos \theta \end{pmatrix}. \quad (40)$$

Условие унитарности в этом случае сводится к соотношению между углами комплексных экспонент:

$$\gamma_{00} - \gamma_{01} = \gamma_{10} - \gamma_{11}.$$

Бинарные вентили задаются унитарными матрицами размера 4×4 . Примером может служить элемент «управляемый фазовый сдвиг» [28], задаваемый матрицей:

$$W = \begin{array}{c|cccc} & |00\rangle & |01\rangle & |10\rangle & |11\rangle \\ \hline |00\rangle & 1 & 0 & 0 & 0 \\ |01\rangle & 0 & 1 & 0 & 0 \\ |10\rangle & 0 & 0 & 1 & 0 \\ |11\rangle & 0 & 0 & 0 & e^{j\varphi} \end{array}$$

Можно показать, что результирующее состояние регистра после бинарного вентиля в общем случае является запутанным. Последовательная комбинация элементарных преобразований в принципе позволяет выполнить любое квантовое вычисление. Существенным вопросом является эффективность вычислений. В работе [27] для построения эффективного квантового вычисления Шор использовал модификацию быстрого алгоритма Фурье с основанием 2. Как было показано в предыдущих разделах,

быстрое преобразование Фурье может быть реализовано в классе быстрых нейронных сетей, а в общем случае БНС могут быть использованы для формирования широкого набора унитарных факторизуемых преобразований.

Применительно к квантовым вычислениям размерность нейронной сети должна быть равна степени двойки. Это ведет к тому, что все нейронные ядра сети структурно-подобны и представляют собой унитарные матрицы размера 2×2 . Если все ядра в пределах слоя параметрически одинаковы, то нейронный слой однозначно сопоставляется с одним q -битом квантового регистра. В этом случае эквивалентный унитарный оператор нейронной сети, описывающий эволюцию состояния q -битного регистра размерности n , выражается кронекеровским произведением матриц нейронных ядер: $H = W_0 \times W_1 \times \dots \times W_{n-1}$. На рис. 27 показана квантовая схема подобного вычисления. Для данной схемы число квантовых операций равно размерности квантового регистра и, следовательно, данное преобразование является максимально эффективным. Результирующее состояние для данной квантовой схемы всегда факторизуемо.

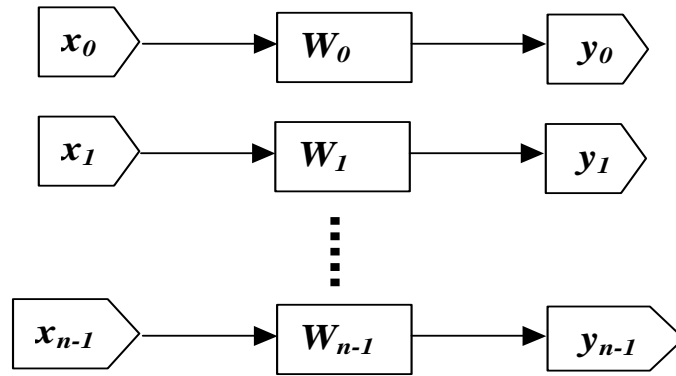


Рис. 27. Квантовая схема БНС с однородными ядрами в слое

Следует отметить, что моделирование квантового регистра на обычном компьютере приводит к NP -сложной задаче. На рис. 28 показана квантовая схема быстрого преобразования Фурье для размерности 2^3 . В отличие от предыдущего случая квантовая схема содержит бинарные вентили управляемого фазового сдвига. Значение фазового сдвига и число бинар-

ных вентилях зависит от номера нейронного слоя. Все унарные вентили определяются матрицей преобразования Адамара:

$$W = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Детали построения квантовой схемы изложены в работе [29]. Результирующее состояние квантового регистра является запутанным, тем не менее, квантовая схема является полиномиально эффективной.

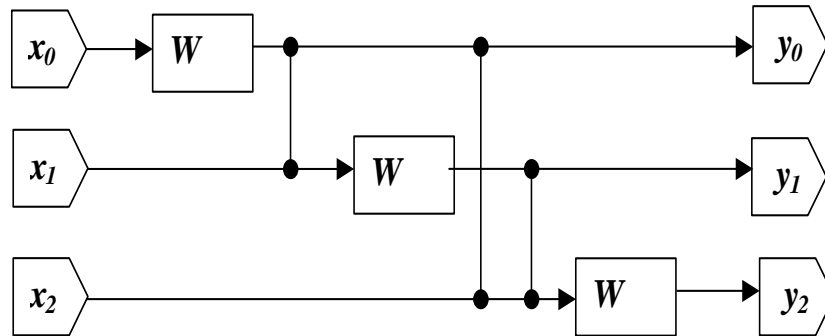


Рис. 28. Квантовая схема быстрого преобразования Фурье

Нейронная технология обычно имеет дело с образами. В квантовом варианте образом является квантовое состояние регистра. Задача распознавания образа сводится к построению такого унитарного преобразования, которое формирует на выходе одно из базисных состояний регистра с вероятностью близкой или равной единице. По существу подобная задача была рассмотрена при построении приспособленных спектральных преобразований. Было показано, что настройка сводится к фрактальной декомпозиции функции и выбору параметров нейронных ядер. Методы декомпозиции основаны на фрактальной фильтрации, и отличаются только комплексным характером фрактальных фильтров.

Для нейронных ядер размерности два можно предложить более общий способ выбора параметров ядер, чем было изложено ранее. Обозначим через $\dot{w}_i(u, v)$ коэффициенты нейронного ядра в слое i , а через $\dot{\varphi}_i(u)$ (здесь и далее точка над символом означает комплексную величину, а «черточка» — комплексно сопряженную) — соответствующую компоненту фрактального

разложения функции-образа $\dot{f}(u)$. Значениями функции $\dot{f}(u)$, заданной на интервале длиной 2^n , являются коэффициенты базисных векторов состояния квантового регистра. Как было показано ранее, спектральный оператор будет приспособлен к функции, если нейронные ядра будут приспособлены к соответствующим компонентам фрактального разложения функции. Для унитарных ядер условие приспособленности имеет вид:

$$\begin{cases} \sum_{u_m} \dot{\psi}_{i^m}(u_m, v_m) \bar{\varphi}_{i^m}(u_m) = 1, & \text{при } v_m = x_m, \\ \sum_{u_m} \dot{\psi}_{i^m}(u_m, v_m) \bar{\varphi}_{i^m}(u_m) = 0, & \text{при } v_m \neq x_m. \end{cases}$$

Например, для компоненты $\dot{\varphi}_{i^m}(u_m) = \varphi_{i^m}(u_m) e^{j\alpha_{u_m}}$ и ядра вида (40) приспособленного по первому столбцу данные условия приводятся к виду:

$$\begin{aligned} \varphi_{i^m}(0) e^{j(\gamma_{00}-\alpha_0)} \cos \theta + \varphi_{i^m}(1) e^{j(\gamma_{10}-\alpha_1)} \sin \theta &= 1, \\ \varphi_{i^m}(0) e^{j(\gamma_{01}-\alpha_0)} \sin \theta - \varphi_{i^m}(1) e^{j(\gamma_{11}-\alpha_1)} \cos \theta &= 0. \end{aligned}$$

Откуда следуют расчетные формулы для определения параметров ядер:

$$\operatorname{tg} \theta = \varphi_{i^m}(1)/\varphi_{i^m}(0), \quad \gamma_{00} = \alpha_0, \quad \gamma_{10} = \alpha_1, \quad \gamma_{01} - \alpha_0 = \gamma_{11} - \alpha_1.$$

Данными формулами углы γ_{01}, γ_{11} определяются неоднозначно, для определенности можно положить $\gamma_{01} = \alpha_0, \gamma_{11} = \alpha_1$. Если исходное состояние квантового регистра совпадает с функцией образа, то квантовое состояние после воздействия оператора нейронной сети будет в точности совпадать с одним из базовых состояний. В общем случае БНС выполняет элементарные преобразования над состояниями квантового регистра, а не отдельных q -бит. Поэтому в случае сильно запутанных состояний вычислительная эффективность квантовых БНС падает до NP -класса.

Многомерные БНС

Парадигма многослойных БНС достаточно просто распространяется на многомерное пространство. Грамматика формального языка полностью сохраняется, основные отличия заключаются в многомерной интерпретации букв алфавита. Для d -мерной сети любая буква алфавита заменяется символьным вектором, в котором каждая координата соответствует одной из размерностей сети.

Рассмотрим, например БНС размерности $d = 2$. Любая буква алфавита в этом случае представляется двумерным вектором. Будем кодировать координаты каждого вектора двухсимвольным набором, где первый символ выбирается из множества i, j, u, v и означает родовое имя буквы, а второй символ (x либо y) означает пространственное направление. В такой интерпретации номер ядра в слое λ обозначается символьным вектором $i^\lambda = (ix^\lambda \ iy^\lambda)$. Ядра слоя располагаются в плоскости, координаты ядер задаются значениями компонент вектора i^λ . Синаптическая карта каждого нейронного ядра представляет собой четырехмерную матрицу с элементами $w_{i^\lambda}^\lambda(u_\lambda, v_\lambda)$, где значения $u_\lambda = (ux_\lambda \ uy_\lambda)$ определяют локальные пространственные координаты рецепторов, а $v_\lambda = (vx_\lambda \ vy_\lambda)$ — локальные координаты аксонов ядра (см. рис. 29).

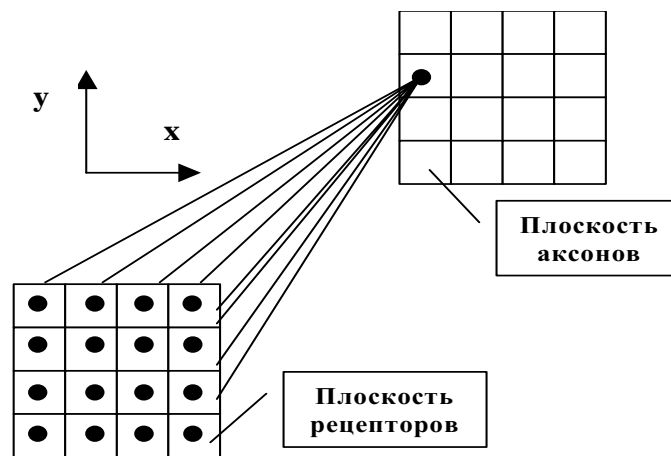


Рис. 29. Нейронное ядро двумерной БНС

Структурная модель κ -слойной БНС как и прежде задается каноническим предложением состоящим из κ слов, упорядоченных по числу вхождений двумерных букв с родовым именем i . При построении графической интерпретации, дугами соединяются нейронные ядра смежных слоев, которые имеют совпадающие значения для одноименных двумерных букв. На рис. 30 показано правило построения структурной модели для канонического предложения:

$$[\langle i_1 i_2 \rangle \langle i_2 j_1 \rangle \langle j_1 j_0 \rangle] .$$

Все разрядные переменные имеют основания равные двум. Пространственные координаты ядер определяются словами $\langle ix_1ix_2 \rangle, \langle iy_1iy_2 \rangle$ — для слоя 0, $\langle ix_2jx_1 \rangle, \langle iy_2jy_1 \rangle$ — для слоя 1, и $\langle jx_1jx_0 \rangle, \langle jy_1iy_0 \rangle$ — для слоя 2.

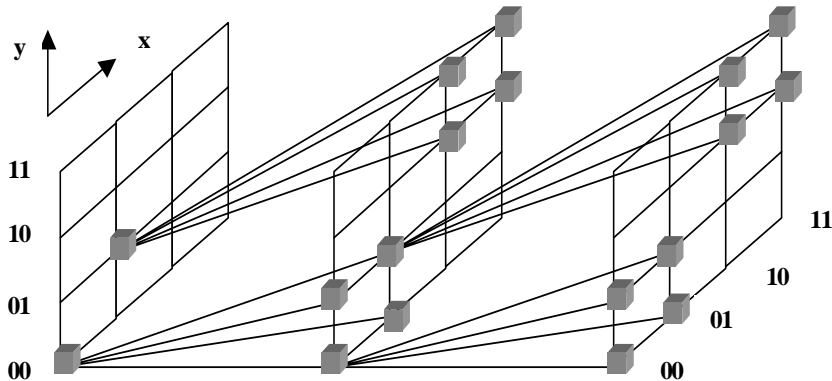


Рис. 30. Правило построения двумерной БНС

Литература

1. Дорогов А. Ю., Алексеев А. А., Буторин Д. А. Нейронные сети со структурой быстрого алгоритма // *Нейроинформатика и ее приложения: 6 Всерос. семинар, 20–25 октября 1998г., Красноярск. Тезисы докл.* – Красноярск, КГТУ, 1998. – С. 53.
2. Good I. J. The Interaction Algorithm and Practical Fourier Analysis // *Journal of Royal Statistical Society.* – Ser. B. – 1958. – Vol. 20. – No. 2. – P. 361–372.
3. Andrews H. C., Caspari K. L. A General Techniques for Spectral Analysis // *IEEE Trans. Computers.* – 1970. – Vol. C-19. – Jan., No. 1. – P. 16–25.
4. Эндрюс Г. Применение вычислительных машин для обработки изображений: Перевод с англ. под ред. Б.Ф. Курьянова. – М., 1977. – 160 с.
5. Солодовников А. И., Спиваковский А. М. Основы теории и методы спектральной обработки информации. – Л., 1986. – 272 с.
6. Лабунец В. Г. Единый подход к алгоритмам быстрых преобразований // *Применение ортогональных методов при обработке сигналов и анализа систем: Межвуз. Сб.* – Свердловск: Уральск. Политехн. Ин-т. – 1980. – С. 4–14.

7. *Cooley J. W., Tukey J. W.* An Algorithm for the Machine Computation of Complex Fourier Series // *Math. Comp.* – 1965. – No. 19. – P. 297–301.
8. *Рабинер Л., Гоулд.* Теория и применение цифровой обработки сигналов: Пер. с англ. – М.: Мир, 1978. – 848 с.
9. *Трахтман А. М., Трахтман В. А.* Основы теории дискретных сигналов на конечных интервалах. – М.: Сов. Радио, 1975. – 208 с.
10. *Дорогов А. Ю.* Быстрые нейронные сети. – СПб.: Изд-во С. Петерб. ун-та, 2002. – 80 с.
11. *Дорогов А. Ю.* Структурный анализ слабосвязанных нейронных сетей // *Управление в социальных, экономических и технических системах*. Кн. 3. Управление в технических системах: Труды Межреспубликанской научной конференции, г. Кисловодск 28 июня–2 июля 1998 г. – С. 75–80.
12. *Дорогов А. Ю.* Структурный синтез модульных слабосвязанных нейронных сетей. Часть 1. Методология структурного синтеза модульных нейронных сетей // *Кибернетика и системный анализ*. – 2001. – № 2. – С. 34–42.
13. *Дорогов А. Ю.* Структурный синтез модульных слабосвязанных нейронных сетей. Часть 2. Ядерные нейронные сети // *Кибернетика и системный анализ*. – 2001. – № 4. – С. 13–20.
14. *Uhr L.* Algorithm-Structured Computer Arrays and Networks: Parallel Architectures for Perception and Modelling. – Academic Press, New York, 1984.
15. *Дорогов А. Ю.* Генезис слабосвязанных нейронных сетей // Всероссийская научно-техническая конференция «Нейроинформатика-99», Москва, 20–22 января 1999 г. – Сб. науч. тр. Часть 1. – М.: 1999. – С. 64–70.
16. *Эдельман Дж., Маункасл В.* Разумный мозг: Кортикальная организация и селекция групп в теории высших функций головного мозга / Пер. с англ. *Н. Ю. Алексеенко*, под ред. *Е. К. Соколова*. – М.: Мир. – 1981. – 133 с.
17. *Дорогов А. Ю.* Фракталы и нейронные сети // *Проблемы нейрокибернетики* (материалы Юбилейной международной конференции по нейрокибернетике посвященной 90-летию со дня рождения проф. А. Б. Когана, 23–29 октября 2002 г., Ростов-на-Дону). Том 2. – Ростов-на-Дону. – 2002. – С. 9–14.
18. *Дорогов А. Ю., Алексеев А. А.* Пластичность многослойных слабосвязанных нейронных сетей // *Нейрокомпьютеры: разработка и применение* № 11, 2001, с. 22–40.
19. *Малинецкий Г. Г., Потапов А. Б.* Современные проблемы нелинейной динамики. – М.: Эдиториал УРСС, 2000. – 336 с.
20. *Кроновер Р. М.* Фракталы и хаос в динамических системах. – М.: Постмаркет, 2000. – 350 с.

21. Макаренко Н. Г. Фракталы, аттракторы, нейронные сети и все такое // «Нейроинформатика 2002»: 4-й Всеросс. науч. техн. конф. 23–25 января 2002 г., Москва – Лекции по нейроинформатике. – Часть 2. – с. 136–169.
22. Stark J. Iterated function systems as neural networks // *Neural Networks*. – 1991. – V. 4. – pp. 679–690.
23. Астафьева Н. М. Вейвлет-анализ: основы теории и примеры применения // *Успехи физических наук*. – Т. 166. – 1996. – № 11. – С. 1145–1170.
24. Каппелини В. и др. Цифровые фильтры и их применение: Пер. с англ. – М.: Энергоатомиздат, 1983. – 360 с.
25. Feynman R. Simulating physics with computers // *International Journal of Theoretical Physics*, **21**, 6&7, 467–488, 1982.
26. Feynman R. Quantum mechanical computers // *Optics News* **11**, 1985. Also in *Foundations of Physics*, **16**: 507–531, 1986.
27. Shor P. W. Algorithms for quantum computation: Discrete log and factoring // In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (Nov. 1994). pp. 124–134, 1994. Institute of Electrical and Electronic Engineers Computer Society Press.
URL: <ftp://netlib.att.com/netlib/att/math/shor/quantum.algorithms.ps.Z>.
28. Риффель Э., Полак В. Основы квантовых вычислений // *Квантовый компьютер и квантовые вычисления*. – Т. 1, № 1, 2000. – с. 4–57.
29. Ekert A., Hayden P., Inamori H. Basic concepts in quantum computation. – Centre for Quantum Computation University of Oxford OX1 3PU, United Kingdom.

Александр Юрьевич ДОРОГОВ, кандидат технических наук, доцент кафедры автоматизации и процессов управления Санкт-Петербургского государственного электротехнического университета («ЛЭТИ»). Области научных интересов: нейронные сети, быстрые перестраиваемые преобразования, фракталы, системный анализ. Автор более 100 печатных работ и одной монографии.

НАУЧНАЯ СЕССИЯ МИФИ–2004

НЕЙРОИНФОРМАТИКА–2004

VI ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ

ЛЕКЦИИ
ПО НЕЙРОИНФОРМАТИКЕ
Часть 1

Оригинал-макет подготовлен Ю. В. Тюменцевым
с использованием издательского пакета \LaTeX 2_ε
и набора PostScript–шрифтов PSCyr

Подписано в печать 25.11.2003 г. Формат 60 × 84 1/16
Печ. л. 12, 5. Тираж 200 экз. Заказ №

*Московский инженерно-физический институт
(государственный университет)
Типография МИФИ
115409, Москва, Каширское шоссе, 31*