

ПРЕДСТАВЛЕНИЕ КОНЕЧНЫХ ГРУПП НЕЙРОННЫМИ СЕТЯМИ

П. А. Головинский, О. С. Довжикова

Воронежский государственный архитектурно-строительный университет

Рассмотрено представление дискретных групп и полугрупп нейронными сетями. Построен алгоритм выделения из временного ряда структуры, представляемой конечным автоматом, и продемонстрирована реализация конечного автомата нейронными сетями. Установлена связь дискретных симметрий с симметриями перестановок входов нейронов в сети. Показана возможность применения нейронных сетей, использующих инварианты групп, для оценки точности симметрии данных.

1. ВВЕДЕНИЕ

Представляя информацию в нейронной сети, после предварительных процедур ее обобщения и сжатия, необходимо иметь возможность различить состояния внешних объектов. Состояния объектов можно кодировать с помощью различных внутренних состояний нейронной сети. Если после преобразований в сети образ объекта сужается до конечного набора параметров, принимающих дискретные значения, то состояние нейросети можно считать словом, то есть набором упорядоченных символов (a, b, c, \dots, d) . В качестве символов можно, в частности, использовать числа [1]. Последовательность переходов между словами часто подчинена сложным правилам, зависящим к тому же от изменения параметров внешней среды.

Взаимно однозначное соответствие состояний разных нейронных сетей и слов определяет их изоморфизм. С точки зрения представления и обработки информации естественно считать, что изоморфные нейронные сети эквивалентны: их физическое строение теряет значение, и важны лишь отношения между представляемыми объектами и состояниями сетей. Изменения в нейронной сети можно рассматривать как переход от одного слова к другому путем перестановки элементов. Такой переход легко записать с помощью матричного произведения, располагая символы в вектор-столбец $\mathbf{x} = {}^t(a, b, \dots, z)$ и действуя на него матрицей перестановок $A\mathbf{x} = \mathbf{y}$. Матрицы A таковы, что в каждой строке и в каждом столбце содержится только один ненулевой элемент, равный единице. Действие матриц обеспечивает любую перестановку элементов слова. Формально для невырожденных квадратных матриц осуществимо преобразова-

ние и в обратном направлении $A^{-1}\mathbf{y} = \mathbf{x}$, то есть соответствующие информационные процессы обратимы. Для реальных объектов такие переходы не всегда возможны, поскольку весьма распространены необратимые процессы.

В важном частном случае, когда внутри конечного множества возможна цепочка переходов из любого состояния в любое иное состояние, множество операций переходов замкнуто и образует группу преобразований. В свою очередь группа из конечного числа элементов изоморфна группе перестановок. По теореме Кэли любая конечная группа изоморфна некоторой симметрической группе, и, следовательно, основные свойства конечных групп и их представления можно исследовать с помощью группы перестановок.

Другая задача, которая приводит к использованию групп в нейронных сетях, — это распознавание образов. Здесь важно, чтобы в отображениях проявлялась универсальность, то есть, чтобы полученное впечатление или правило можно было извлечь и применить в других схожих ситуациях. В физике этот принцип инвариантности является основным при формулировке физических законов, которые должны оставаться справедливыми при определенных преобразованиях системы таких, как перенос и поворот системы в пространстве, или изменение начала отсчета времени [2].

В моделировании нейрофизиологических процессов роль симметрии была осознана еще в работах Мак Каллока и Питса, Винера и Розенблата [3—5], однако ее не представить развить в форме, адекватной отображению симметрии в искусственных нейронных сетях. Принципы симметрии по-прежнему представляются плодотворными и требуют дальнейшего развития, а данная работа посвящена пред-

ставлению дискретных симметрий в нейронных сетях.

2. ДИСКРЕТНЫЕ ГРУППЫ

Для представления объектов и их распознавания важны группы преобразований множества объектов в себя [6]. Групповой операцией здесь является последовательность отображений. Если преобразуемое множество наделено некоторыми физическими, геометрическими или другими сохраняемыми свойствами, то появляется группа симметрии.

Рассмотрим реализацию представления произвольной конечной группы нейронной сетью. Она может быть основана на сопоставлении элементу группы нейрона, а бинарное отображение группы в нейронной сети можно построить по таблице группового умножения. При этом пара нейронов, возбужденных в определенном порядке, будет соответствовать некоторый третий так называемый матричный нейрон, являющийся индикатором результата умножения элементов в группе. Поскольку в произвольной группе нет коммутативности, то результирующий нейрон произведения в общем случае зависит от порядка возбуждения пары нейронов-сомножителей.

На первом шаге умножения в сети возбуждается входной нейрон столбца таблицы. Далее возбуждение передается нейрону строки, задающему первый сомножитель, а нейрон столбца деовозбуждается. Эта операция есть результат умножения на единицу. На следующем шаге умножения возбуждается нейрон столбца, который становится вторым сомножителем. Далее на пересечении строки и столбца нейронной сети возбуждается матричный нейрон, а нейроны строк и столбцов деовозбуждаются. На последнем шаге матричный нейрон возбуждает выходной нейрон столбца, соответствующий результату произведения. Полученный элемент можно умножить затем на следующий элемент группы слева, начиная действия со второго шага.

Описанная процедура устанавливает систему переходов между элементами группы, либо подгруппы. Все переходы в сети, кроме последнего, производятся универсально, то есть не зависят от процедуры обучения на группе. Для их осуществления достаточно произвести классификацию элементов группы, например с помощью самоорганизующихся карт Кохонена, а

затем обеспечить распознавание разных элементов группы, например, сетью Хопфилда. Обучение сети производится по таблице, на основе которой формируется матричный нейрон с весом $w_{ij}^k = \delta_{kl}$ для заданных значений нейронов строки и столбца с номерами i и j . Такой выбор весов обеспечивает выполнение табличных правил умножения в группе, задающих по номерам i, j сомножителей g_i, g_j нейрон произведения g_l в столбце с номером $k = l$.

В задаче представления конечной группы нейронной сетью первым шагом является выявление в потоке данных структуры группы. Входные данные часто имеют вид многомерного временного ряда, то есть последовательности «кадров». Тогда для выявления группы необходимо обнаружить последовательности пар преобразований, обеспечивающих получение кадров-объектов, принадлежащих некоторому конечному множеству.

Выявить группу можно, основываясь на частоте повторения событий во временном ряде, если имеется последовательность событий в дискретные моменты времени t : $z_1(t) \xrightarrow{g_1} z_2(t+1) \xrightarrow{g_2} z_3(t+2) \dots$. Преобразования g_1, g_2, \dots обеспечивают бинарную операцию умножения в том случае, если временной ряд содержит также прямые переходы вида $z_1(t) \xrightarrow{g_3=g_2g_1} z_3(t+1)$. Объединим последовательные события ряда в упорядоченные пары: $v_{ij}(t) = \{z_i(t), z_j(t+1)\}$, тогда временной ряд $v_{ij}(t)$ отображает множество переходов. Построим множество $V(t)$, накапливающее переходы каждого вида к моменту времени t :

$$V(t) = \sum_{ij} n_{ij}(t)v_{ij}. \quad (1)$$

Здесь $n_{ij}(t)$ — число повторений переходов вида v_{ij} за время t , и со временем разные виды переходов накапливаются в разной степени. Если имеется конечное подмножество часто повторяющихся переходов, то его можно выделить из множества (1), рассматривая предел среднего значения $v_{ij}(t)$ на больших временах t :

$$\bar{v} = \lim_{t \rightarrow \infty} \frac{V(t)}{t}. \quad (2)$$

Если предел не является пустым множеством, то совокупность переходов v_{ij} с весами $\lim_{t \rightarrow \infty} (n_{ij}(t)/t) \neq 0$ образует подгруппу преобразований. Если кроме того переходы присутствуют симметричными парами v_{ij} и v_{ji} , то совокупность переходов является также группой. В

реальных ситуациях вместо предельного перехода естественно провести оценку величины $n_{ij}(t)/t$ при больших значениях t с заданной точностью.

3. ПРЕДСТАВЛЕНИЕ АВТОМАТОВ

Для получения модели объекта или явления, реагирующего определенным образом на входные воздействия, эффективно использование абстрактного автомата [7]. Конечные детерминированные автоматы не описывают такие свойства, как риск, случайность, вероятность и поэтому имеют ограниченную применимость при описании реальных объектов внешнего мира. Тем не менее, даже модель детерминированного автомата может быть применена к широкому кругу задач. Основой для построения конечного автомата является установление связи между поведением объекта (то есть реакцией выходов на различные входные сигналы) и изменением его внутренних состояний. Прием стандартное определение автомата [7], согласно которому автомат имеет конечное множество состояний $Q = \{q_1, q_2, \dots, q_k\}$, конечное значение входных воздействий, называемых входным алфавитом $X = \{x_1, x_2, \dots, x_m\}$, и конечное число выходных параметров автомата $Y = \{y_1, y_2, \dots, y_n\}$. Действие автомата определяется функцией переходов $\delta : Q \times X \rightarrow Q$ и функцией выходов $\lambda : Q \times X \rightarrow Y$.

Последовательность переходов автомата, соответствующую последовательности входных воздействий, можно алгебраически рассматривать как результат перемножения отображений для отдельных переходов. В результате получится отображение, соответствующее входной последовательности. Так как множество состояний автомата конечно, то множество отображений перехода замкнуто относительно умножения и представляет собой абстрактную полугруппу [7]. Существует два основных вида отображения переходов:

1. Перестановки, когда отображение взаимно однозначно на множестве состояний. Автомат с такими переходными отображениями обладает группой преобразований и является обратимым.

2. Отображения стягивания, когда два различных состояния переходят в одно и то же состояние. Такие переходы отражают наличие необратимых процессов, описываемых полугруппами [1].

Есть несколько эквивалентных способов задания автомата.

При задании автомата графом каждая вершина соответствует некоторому определенному состоянию q_i . Вершины соединены ориентированными ребрами, показывающими направления переходов между состояниями автомата. Каждому ребру приписана пара элементов, указывающая значение входного элемента x_j , вызывающего переход в новое состояние, и значение выходного элемента y_l , вырабатываемого автоматом в результате перехода. Пример подобного задания автомата [7] показан на рис. 1. Данный автомат имеет три возможных состояния q_1, q_2, q_3 , а входной алфавит содержит два элемента x_1 и x_2 , так же, как есть и два выходных элемента y_1, y_2 . Наличие замкнутых дуг показывает, что имеются переходы, при которых состояние автомата не меняется, но вырабатывается выходной сигнал.

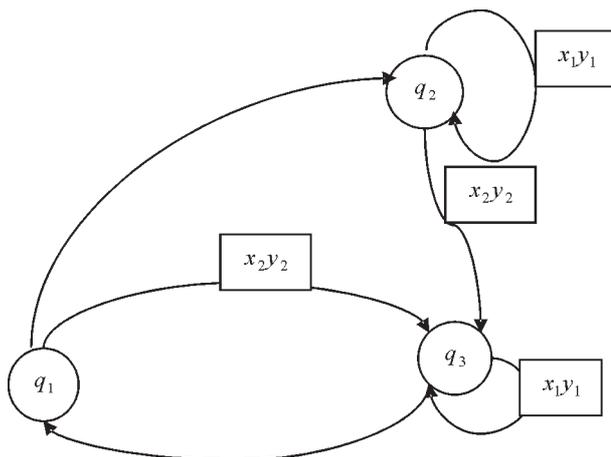


Рис. 1. Пример задания конечного автомата графом

2. Можно задать автомат с помощью таблицы переходов со строками состояний и столбцами входных элементов, где на пересечении соответствующей строки и столбца находятся значения конечного состояния и значения выходного элемента, как показано в табл. Таковую таблицу мы фактически использовали при построении нейронной сети, описывающей группу.

Таблица

Переходы в автомате

Входные воздействия	Состояния		
	q_1	q_2	q_3
x_1	q_2y_1	q_2y_1	q_3y_1
x_2	q_3y_2	q_3y_2	q_1y_2

3. Удобным с вычислительной точки зрения является задание автомата совокупностью матриц, каждая из которых ставит в соответствие начальному состоянию некоторое конечное состояние при определенном значении входного элемента. Для рассматриваемого примера входному элементу x_1 соответствует матрица

$$\begin{matrix} & q_1 & q_2 & q_3 \\ q_1 & \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix} \quad (3)$$

Входному элементу x_2 соответствует матрица переходов

$$\begin{matrix} & q_1 & q_2 & q_3 \\ q_1 & \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{matrix} \quad (4)$$

Начальные состояния нумеруют строки, а конечные состояния нумеруют столбцы.

Если представить начальное состояние вектором-строкой, содержащим только одну единицу, а остальные нули, то результат перехода описывается с умножением этого вектора на матрицу перехода, соответствующую входному значению x_i . В результате перемножения вновь получится вектор-строка, где одна координата — единица, а остальные нули. Номер единичной координаты дает номер конечного состояния.

Поскольку, каждому автомату Мили можно сопоставить автомат Мура, имитирующий его поведение [11; 12], мы ограничимся представлением автоматов Мура, для которых последовательность переходов автомата описывается перемножением матриц

$$T = T_1 T_2 T_3 \dots T_k \quad (5)$$

размером $N \times N$. Для представления автомата нейронной сетью нужно осуществить перемножение пары матриц

$$(T_1 T_2)_{ik} = \sum_j T_{ij}^{(1)} T_{jk}^{(2)}. \quad (6)$$

Суммирование идет по столбцам первой матрицы и по строкам второй. Операция перемножения реализуется набором из N^2 линейных персептронов, где отдельный нейрон осуществляет операцию скалярного умножения.

Для представления автомата нейронной сетью можно расположить параллельно N персептронов с входными весами в виде столбцов

матрицы T_2 и последовательно подавать на N входов каждого из них значения из N строк матрицы T_1 . На выходах персептронов последовательно появятся сначала все элементы первой строки, затем элементы второй строки и так далее. На шаге с номером N появятся элементы последней строки матрицы произведения. Входной нейрон в сети определяет, на какую линейку персептронов будут поданы последовательности входных строк. Если начальное состояние определено, то каждая операция перехода представляет собой умножение вектора строки на матрицу, что обеспечивается на одном шаге N линейными персептронами с весами в виде столбцов соответствующей матрицы T_m . Структура матриц перехода T_m дает требуемые значения скалярных произведений в виде нулей или единиц.

Коммутацию подачи состояния автомата на нужную линейку нейронов в зависимости от входного элемента можно обеспечить управляющим нейроном со ступенчатой функцией отклика, а в качестве весов нейрона задать столбец значений с одной единицей, отвечающей выбранному переходу, и остальными нулевыми значениями. При входной строке с единицей в той же позиции, что и у столбца весов, отклик нейрона равен единице, а в противном случае — нулю. Умножение входного сигнала на значение выхода такого нейрона обеспечит либо передачу, либо отсутствие сигнала, то есть требуемую его коммутацию. Совокупность коммутации и матричного умножения реализует представление автомата нейронной сетью персептронов Мак-Каллока-Питса.

Возможность такого представления дискретного автомата с помощью нейронных сетей ничего не говорит о его эффективности, поэтому желательно рассмотреть и другие представления. Произвольный автомат можно построить также с помощью аппарата гетероассоциативных сетей. В гетероассоциативных сетях определенному вектору, подаваемому на вход, ставится в соответствие некоторый вектор на выходе, входящий в ассоциативную связь с входным вектором. Соответствующие рекуррентные сети подробно описаны С. Осовским [8]. Для представления автомата входной вектор составим из двух частей, первая из которых отобразит состояние автомата, а вторая часть кодирует входной символ. Выходной вектор также составим из двух частей, одна из которых отобразит

новое состояние, в которое переходит автомат, а вторая — кодирует выходной сигнал. Обучение может быть проведено, например, методом проекций. После обучения на полном множестве переходов полугруппы сеть будет изоморфна автомату, который ее обучал.

Для построения нейронной сети изоморфной данному автомату занумеруем пары элементов на входе автомата $\{x_l, q_m\} \in X \times Q$ индексом $i \in I$, а пары на выходе автомата — $\{y_k, q_n\} \in Y \times Q$ индексом $j \in J$. Связывание ассоциацией папы вход-выход означает бинарное отображение $I \rightarrow J$ с определенным номером k . Если закодировать входы и выходы сети ортогональными векторами \mathbf{b}_k и \mathbf{c}_k , то построение ассоциативного образа \mathbf{c} для входного вектора \mathbf{b} реализуется методом ортогональных проекций:

$$\mathbf{c} = \sum_k \mathbf{c}_k(\mathbf{b}_k, \mathbf{b}). \quad (7)$$

Вектор \mathbf{c} может отличаться от точного образа с определенным номером k из-за зашумленности входных данных. Результат можно уточнить, используя карты Кохонена или автоассоциативные сети Хопфилда. В последнем случае мы по сути получаем эквивалент рекуррентной гетероассоциативной сети Коско, в которой используется бинарное кодирование векторов, а их ортогональность не предполагается.

Рассмотрим в качестве примера построение нейронной сети для автомата, задаваемого графом (рис. 1), при дополнительном наличии шума. Переходы, являющиеся шумовой добавкой, показаны на рис. 2 пунктиром. Пусть наблюдается серия переходов v_{ij} , описываемая суммой графов рис. 1 и 2, с числом повторений $v_{12}(30)$, $v_{13}(20)$, $v_{22}(50)$, $v_{23}(25)$, $v_{24}(2)$, $v_{43}(2)$, $v_{33}(32)$, $v_{32}(48)$ и общим числом переходов $n = 212$. Доля переходов v_{24} и v_{43} составляет менее 1 % и с этой точностью ими можно пренебречь в формуле (1). В результате выделяется автомат, показанный на рис. 1 и полугруппа его переходов v_{ij} .

Представим полученный автомат с помощью гетероассоциативной сети Коско. Три разных состояния автомата закодируем двумя бинарными состояниями b_2, b_3 на входе сети и состояниями c_2, c_3 — на выходе, а два разных входных и два выходных сигнала двумя бинарными состояниями b_1 и c_1 соответственно. Работа сети как автомата, задающего переход, состоит в нахождении сетью Коско ассоциации $\mathbf{b} \rightarrow \mathbf{c}$ с переопределением значений $b_2 = c_2$,

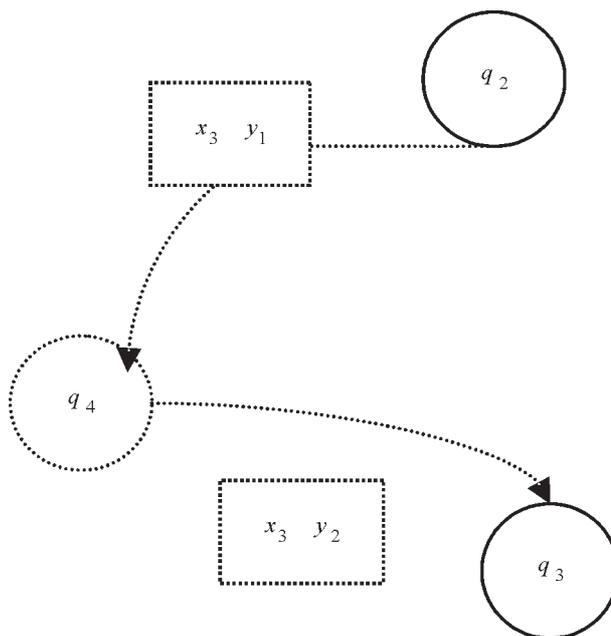


Рис. 2. Переходы, создающие шум в автомате

$b_3 = c_3$, после чего автомат находится в новом состоянии.

4. ПРИБЛИЖЕННАЯ СИММЕТРИЯ

Обычная математическая симметрия носит точный характер, в то время как в реальных условиях она может быть лишь приближенной (Fuzzy symmetry [10]), поэтому важно понять принципы отображения нечеткой симметрии в нейронных сетях. Идея отображения симметрий в нейронных сетях основана на том, что на вход нейрона подается совокупность сигналов \mathbf{x} , по которой вычисляется аргумент передаточной функции

$$u = \sum_i w_i x_i. \quad (8)$$

Симметрия означает неизменность отклика нейрона или группы нейронов на разные комбинации входных сигналов. Поскольку дискретная группа эквивалентна группе перестановок, симметрию можно связать с инвариантным откликом на перестановки входов при разных сигналах. Ясно, что инвариантность при перестановке сигналов x_i и x_j на входе одного нейрона обеспечивается равенством весов $w_i = w_j$, а инвариантность относительно перестановки некоторого подмножества из N_1 сигналов обеспечивается равенством совокупности весов, когда

$$u = w \sum_{i=1}^{N_1} x_i + \sum_{i=N_1+1}^N w_i x_i. \quad (9)$$

Это соотношение справедливо в идеальном случае точной симметрии, в то время как приближенная симметрия подразумевает выделение подмножества входов с близкими по величине весами, которые могут выделить, например, самоорганизующиеся карты Кохонена. Соответствующие инвариантные входы будут отражать симметрию данных. Наоборот, при проектировании сети, предназначенной для работы с некоторыми инвариантными данными, стоит заранее до обучения сети предусмотреть равенство весов у соответствующих симметричных входов.

Рассмотрим, в частности, как выявить в совокупности точек на плоскости симметрию равностороннего треугольника. Как известно, все инварианты группы вращений можно построить из скалярных произведений векторов [9]. Если перейти в систему центра масс трех точек вершин треугольника, то их положение задается векторами $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ равной длины, проведенными из этого центра. Построим нейронную сеть, проверяющую равносторонность треугольника. Образует матрицу всех возможных скалярных произведений

$$W = \begin{pmatrix} (\mathbf{x}_1, \mathbf{x}_1) & (\mathbf{x}_2, \mathbf{x}_2) & (\mathbf{x}_3, \mathbf{x}_3) \\ (\mathbf{x}_1, \mathbf{x}_2) & (\mathbf{x}_2, \mathbf{x}_1) & (\mathbf{x}_3, \mathbf{x}_1) \\ (\mathbf{x}_1, \mathbf{x}_3) & (\mathbf{x}_2, \mathbf{x}_3) & (\mathbf{x}_3, \mathbf{x}_2) \end{pmatrix}, \quad (10)$$

и вычислим нормированные расстояния между парами столбцов матрицы:

$$d_{jk} = \sqrt{\sum_{i=1}^3 (W_{ji} - W_{ki})^2} / \sqrt{\sum_{i=1}^3 W_{ji}^2}. \quad (11)$$

Если все величины $d_{jk} < \varepsilon$, то треугольник является равносторонним, и он инвариантен относительно вращений с точностью $\varepsilon > 0$. В формулах (10) удобная для последующего сравнения расстановка компонент матрицы произведена при выборе последовательности вычисления скалярных произведений.

Таким же образом легко определить наличие симметрии точек вершин квадрата или другого правильного многоугольника. Приведение к единой стандартной перестановке можно обеспечить упорядочиванием компонент векторов по величине. В качестве конкретного примера оценим точность симметрии звезды. На рис. 3 приведены две звезды, одна из которых обладает точной поворотной симметрией, а вторая лишь приближенно является правильной. Точность совпадения звезд на основании сравнения

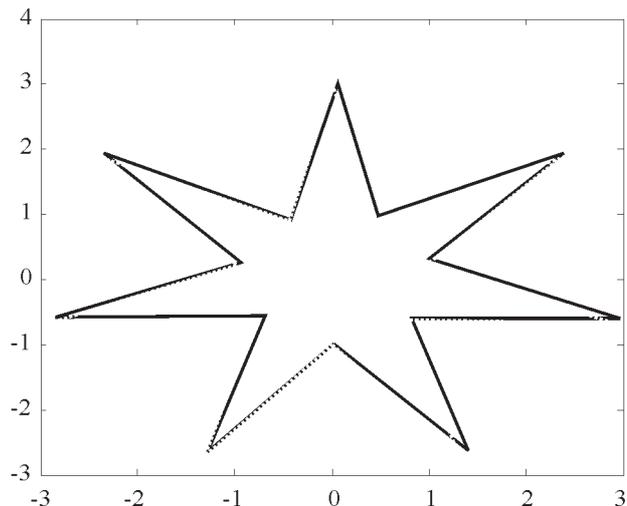


Рис. 3. Симметричная семиугольная звезда — пунктир, и приближенно симметричная звезда — сплошная линия

их инвариантов составила 0,04. Произвольный объект такого вида содержит 14 точек вершин углов, и при его описании возникает симметричная матрица скалярных произведений $W = W^T$ размером 14×14 . В силу этого имеется 105, вообще говоря, различающихся между собой чисел, которые полностью описывают взаимное расположение этих точек. Для симметричной фигуры число различных между собой элементов матрицы W значительно сокращается. Для симметричной звезды рис. 3 таких различных по величине скалярных произведений оказывается всего 12, что нетрудно проверить путем рассмотрения пар вершин. Любая другая фигура достаточно близкая к такой симметричной звезде должна характеризоваться тем же набором 12 отличных между собой чисел. Это количество элементов достаточно для матрицы сравнения, что почти в десять раз сокращает объем хранения информации, необходимой для распознавания образа. Рассмотренный пример относился к случаю, когда графы объектов совпадают, а отличаются метрические характеристики. Приближенное совпадение графов рассматривается, в частности, в [11], но эта задача выходит за рамки данной работы.

ЗАКЛЮЧЕНИЕ

Проделанный анализ некоторых принципов представления конечных групп искусственными нейронными сетями демонстрирует полезность принципов симметрии как в задачах

распознавания образов, так и в задачах представления конечных автоматов с помощью таких сетей. Учет внутренней симметрии данных позволяет существенно сокращать количество информации, необходимое для их кодирования и распознавания. Возможности, которые открываются на пути использования представления групп и полугрупп нейронными сетями, связаны в том числе с автоматическим программированием сложных последовательностей событий и операций.

СПИСОК ЛИТЕРАТУРЫ

1. *Эшби У.Р.* Введение в кибернетику / У. Р. Эшби. — М. : Ком. книга, 2005. — 432 с.
2. *Вигнер Е.* Инвариантность и законы сохранения. Этюды о симметрии / Е. Вигнер. — М. : Едиториал УРСС, 2002. — 320 с.
3. *Pitts W.H.* How We Know Universals: The Perception of Auditory and Visual Forms / W. H. Pitts, W. S. McCulloch // Bull. Math. Biophys. — 1947. — Vol. 9, P. 127—147.
4. *Винер Н.* Кибернетика / Н. Винер. — М. : Наука, 1983. — 340 с.
5. *Розенблат Ф.* Принципы нейродинамики: Перцептроны и теория механизмов мозга / Ф. Розенблат. — М. : Мир, 1965.
6. *Хамермеш М.* Теория групп и ее применения к физическим проблемам / М. Хамермеш. — М. : Едиториал УРСС, 2002. — 588 с.
7. Алгебраическая теория автоматов, языков и полугрупп / под ред. М. А. Арбиба. — М. : Статистика, 1975. — 335 с.
8. *Осовский С.* Нейронные сети для обработки информации / С. Осовский. — М. : Финансы и статистика, 2002. — 344 с.
9. *Вейль Г.* Классические группы. Инварианты и представления / Г. Вейль. — М. : Едиториал УРСС, 2004. — 400 с.
10. *Köhler A.T.* Fuzzy symmetries of point ensembles. Computers and Mathematics with Applications / A. T. Köhler. — 1999. — Vol. 37. — P. 89—113.
11. *Buchheim C.* Detecting symmetries by branch and cut. Mathematical Programming, Series B / C. Buchheim, M. Jünger. — 2003. — Vol. 98. — P. 369—384.