

# ИНФОРМАТИКА

УДК 519.68:007.5; 512.5

## АЛГЕБРАИЧЕСКИЕ СВОЙСТВА АБСТРАКТНЫХ НЕЙРОННЫХ СЕТЕЙ

И. И. Слеповичев

Слеповичев Иван Иванович, старший преподаватель кафедры теоретических основ компьютерной безопасности и криптографии, Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского, gurgutan@yandex.ru

Современный уровень развития нейроинформатики позволяет использовать искусственные нейронные сети для решения различных прикладных задач. Однако многие применяемые на практике нейросетевые методы не имеют строгого формального математического обоснования, являясь эвристическими алгоритмами. Это накладывает определенные ограничения на развитие нейросетевых методов решения задач. В то же время существует широкий класс математических моделей, хорошо изученных в рамках таких дисциплин, как теория абстрактных алгебр, теория графов, теория конечных автоматов. Возможность использовать результаты, полученные в рамках этих дисциплин, применительно к нейросетевым моделям может быть хорошим подспорьем в изучении искусственных нейронных сетей, их свойств и возможностей. В данной работе даны формулировки и определения нейросетевых моделей с точки зрения универсальной алгебры и теории графов. Приведены основные теоремы универсальной алгебры в нейросетевой трактовке. В статье также предлагается способ формального описания нейросети граф-схемой, которая позволяет использовать результаты теории графов для анализа нейросетевых структур.

*Ключевые слова:* нейронные сети, гомоморфизм, конгруэнция, граф-схема нейросети, вычисления на графе.

DOI: 10.18500/1816-9791-2016-16-1-96-103

НАУЧНЫЙ  
ОТДЕЛ

### 1. ОПРЕДЕЛЕНИЕ АБСТРАКТНОЙ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ

В [1–5] приведены различные определения искусственной нейронной сети, в которых нейронная сеть рассматривается как некая вычислительная система, получающая на вход значения и преобразующая их в процессе своего функционирования в выходные значения. Такой подход эффективен при рассмотрении прикладных нейросетевых задач, однако он неудобен для изучения математических свойств нейронных сетей, так как не универсален и часто не основан на строгих формализованных понятиях математики. Эти недостатки можно преодолеть, определив дополнительно понятие *абстрактной нейронной сети* как алгебраической системы, которая при уточнении определения становится неотличимой от искусственной нейронной сети, определённой традиционным способом.

*Абстрактной нейронной сетью (абстрактной нейросетью)* будем называть алгебраическую систему  $(A, F)$ , в которой носитель алгебры

$$A = S \times X \times Y, \quad (1)$$

— непустое множество, определяемое декартовым произведением



множества векторов состояний, входных векторов и выходных векторов,  $F$  — набор операций на этом множестве, который включает операцию  $g$ , вычисляющую выходное значение, и операцию  $h$ , вычисляющую новое состояние:

$$g : S \times X \rightarrow Y, \quad h : S \times X \times Y \rightarrow S. \quad (2)$$

Можно говорить, что абстрактная нейронная сеть является трёхосновной алгеброй [6, с. 212]

$$N = (S, X, Y, g, h), \quad (3)$$

где множества  $S, X, Y$  — основные множества, а  $g, h$  — операции из  $F$ , которые мы в дальнейшем будем называть функциями. Множество  $F$  может содержать и другие операции. Например, нам может понадобиться функция, осуществляющая некую «оценку» результата работы нейросети:  $o_g(y)$ ,  $y \in Y$ , или функция, осуществляющая «оценку» текущего состояния нейросети:  $o_h(s)$ ,  $s \in S$ , или нам могут понадобиться нульарные операции, фиксирующие некоторое входное значение  $x^0$  и начальное значение состояния  $s^0$ . Такие операции могут быть определены при необходимости для уточнения понятия абстрактной нейросети.

Определение абстрактной нейросети как трёхосновной алгебры позволяет ввести аналоги конструкций универсальной алгебры. Например, непосредственным переносом мы можем определить для абстрактной нейросети понятие *устойчивости системы подмножеств, подсети, пересечения подсетей*.

Пусть  $N = (S, X, Y, g, h)$  — абстрактная нейросеть с непустыми множествами допустимых состояний, входных и выходных векторов нейросети. Система подмножеств  $S^* \subseteq S$ ,  $X^* \subseteq X$ ,  $Y^* \subseteq Y$  называется *устойчивой в нейросети*, если для функций  $g$  и  $h$  и  $\forall s^* \in S^*$ ,  $x^* \in X^*$ ,  $y^* \in Y^*$  элементы  $g(s^*, x^*) \in Y^*$ ,  $h(s^*, x^*, y^*) \in S^*$ . Ограничивая функции  $g, h$  для нейросети  $N$  на подмножествах  $S^*, X^*, Y^*$  получаем *подсеть*  $N^* = (S^*, X^*, Y^*, g, h)$ .

*Пересечением подсетей*  $N_1 = (S_1, X_1, Y_1, g, h)$ ,  $N_2 = (S_2, X_2, Y_2, g, h)$  абстрактной нейросети  $N = (S, X, Y, g, h)$  будем называть абстрактную нейросеть  $N_1 \cap N_2 = (S_1 \cap S_2, X_1 \cap X_2, Y_1 \cap Y_2, g, h)$ , функциями которой являются функции из  $N$ , ограниченные на соответствующих множествах.

Легко доказывается, что пересечение любого семейства подсетей снова будет абстрактной нейросетью, и, следовательно, совокупность  $Sub(N)$  всех подсетей нейросети  $N$  будет полной решёткой<sup>1</sup>, при условии, что её наименьшим элементом будет нейросеть с пустыми множествами и пустыми операциями. Действительно, если  $N_i = (S_i, X_i, Y_i, g, h)$  — подсеть  $N = (S, X, Y, g, h)$ , то

$$g(s, x) \in Y_i \& h(s, x, y) \in S_i \Rightarrow g(s, x) \in \cap Y_i \& h(s, x, y) \in \cap S_i. \quad (4)$$

*Замыканием* множеств  $S^* \in S$ ,  $X^* \in X$ ,  $Y^* \in Y$  относительно функций  $g, h$  называется множество всех элементов (включая элементы из  $S^*, X^*, Y^*$ ), которые можно получить из этих множеств, применяя функции  $g, h$ .

Пусть  $N_1 = (S_1, X_1, Y_1, g_1, h_1)$ ,  $N_2 = (S_2, X_2, Y_2, g_2, h_2)$  — две нейросети. Семейство отображений

$$\Gamma = (\Gamma_S, \Gamma_X, \Gamma_Y), \quad \Gamma_S : S_1 \rightarrow S_2, \quad \Gamma_X : X_1 \rightarrow X_2, \quad \Gamma_Y : Y_1 \rightarrow Y_2 \quad (5)$$

будем называть *гомоморфизмом* абстрактной нейросети  $N_1$  в абстрактную нейросеть  $N_2$ , если оно согласовано с операциями  $g_i$  и  $h_i$ ,  $i = 1, 2$ , в том смысле, что

$$\Gamma_S(h_1(s, x, y)) = h_2(\Gamma_S(s), \Gamma_X(x), \Gamma_Y(y)), \quad (6)$$

$$\Gamma_Y(g_1(s, x)) = g_2(\Gamma_S(s), \Gamma_X(x)), \quad \forall s \in S_1, \quad \forall x \in X_1, \quad \forall y \in Y_1. \quad (7)$$

Нейросеть  $N_2$ , являющаяся гомоморфным образом абстрактной нейросети  $N_1$ , можно представить как некоторую огрублённую модель абстрактной нейросети  $N_1$ . Сохраняя принципы функционирования своего прообраза, нейросеть  $N_2$  теряет часть информации о входных векторах, векторах состояний и выходных векторах.

<sup>1</sup>Полная решётка — частично упорядоченное множество, в котором всякое непустое подмножество имеет точную верхнюю и точную нижнюю грань [6, с. 157]



Так как  $g$  отображает  $S \times X$  исключительно в  $Y$ , а  $h$  отображает  $S \times X \times Y$  в  $S$ , то пару отображений  $(g, h)$  мы можем рассматривать как отображение  $F : S \times X \times Y \rightarrow S \times Y$ :

$$F(s, x, y) = (h(s, x, y), g(s, x)), \quad \forall s \in S, \quad x \in X, \quad y \in Y. \quad (8)$$

Такие обозначения позволяют нам использовать более короткую запись определения гомоморфизма нейросети  $N_1$  в  $N_2$ :

$$\Gamma(F_1(s, x, y)) = F_2(\Gamma(s, x, y)), \quad \forall (s, x, y) \in S \times X \times Y, \quad (9)$$

или в префиксной операторной форме записи  $(\Gamma \circ F_1)(s, x, y) = (F_2 \circ \Gamma)(s, x, y)$ . Из такой записи видно, что для поиска гомоморфизмов нейросети  $N_1$  в  $N_2$  нужно решить операторное уравнение  $\Gamma \circ F_1 = F_2 \circ \Gamma$  относительно параметров отображения  $\Gamma$ .

Гомоморфизм  $\Gamma$  называют инъективным, сюръективным, биективным, если соответствующим свойством обладает каждая его компонента  $\Gamma_S, \Gamma_X, \Gamma_Y$ . Биективный гомоморфизм абстрактной нейросети будем называть изоморфизмом. Выходные значения и значения состояний, вычисленные изоморфными нейросетями, совпадают при одинаковых входных значениях и начальных состояниях. При этом внутренняя структура и способ вычисления значений этими нейросетями могут различаться.

Если  $\Gamma : N_1 \rightarrow N_2$  и  $E : N_2 \rightarrow N_3$  — гомоморфизмы нейросетей, то можно определить композицию (произведение) этих гомоморфизмов, полагая

$$\Gamma \circ E := (\Gamma_1 \circ E_1, \Gamma_2 \circ E_2, \Gamma_3 \circ E_3). \quad (10)$$

Без труда можно перенести все алгебраические свойства гомоморфизмов многоосновных алгебр, связанные с произведением, на случай абстрактных нейросетей. Докажем теорему о связи между подсетями и гомоморфизмами.

**Теорема 1.** Пусть  $\Gamma : N_1 \rightarrow N_2$  — гомоморфизм нейросети  $N_1$  в нейросеть  $N_2$ . Если  $N'_1 = (S'_1, X'_1, Y'_1, g_1, h_1)$  — подсеть в  $N_1$ , то  $\Gamma(N'_1) := (\Gamma_S(S'_1), \Gamma_X(X'_1), \Gamma_Y(Y'_1), g_2, h_2)$  — подсеть в  $N_2$ . Если  $N'_2 = (S'_2, X'_2, Y'_2, g_2, h_2)$  — подсеть в  $N_2$ , то подсеть в  $N_1$  будет  $\Gamma^{-1}(N'_2) := (\Gamma_S^{-1}(S'_2), \Gamma_X^{-1}(X'_2), \Gamma_Y^{-1}(Y'_2), g_1, h_1)$ .

**Доказательство.** Пусть  $N'_1$  — подсеть  $N_1$  и  $s_2 \in \Gamma_S(S'_1), x_2 \in \Gamma_X(X'_1), y_2 \in \Gamma_Y(Y'_1)$  — произвольные элементы. Тогда найдутся  $s_1 \in S'_1$  и  $x_1 \in X'_1$  такие, что  $s_2 = \Gamma_S(s_1), x_2 = \Gamma_X(x_1)$ . Так как  $g_1(s_1, x_1) \in Y'_1$  и  $h_1(s_1, x_1, y_1) \in S'_1$ , то поскольку  $\Gamma$  — гомоморфизм, имеем:  $g_2(s_2, x_2) = g_2(\Gamma_S(s_1), \Gamma_X(x_1)) = \Gamma_Y(g_1(s_1, x_1)) \in \Gamma_Y(Y'_1)$ ,  $h_2(s_2, x_2, y_2) = h_2(\Gamma_S(s_1), \Gamma_X(x_1), \Gamma_Y(y_1)) = \Gamma_S(h_1(s_1, x_1, y_1)) \in \Gamma_S(S'_1)$ , и, значит,  $\Gamma(N'_1)$  — подсеть  $N_2$ .

Пусть теперь  $N'_2$  — подсеть  $N_2$ . Если хотя бы одно из множеств  $\Gamma_S^{-1}(S'_2), \Gamma_X^{-1}(X'_2), \Gamma_Y^{-1}(Y'_2)$  пусто, то  $\Gamma^{-1}(N'_2)$  — подсеть  $N_1$ . Допустим, что  $\Gamma_S^{-1}(S'_2) \neq \emptyset, \Gamma_X^{-1}(X'_2) \neq \emptyset, \Gamma_Y^{-1}(Y'_2) \neq \emptyset$ . Используя то, что  $N'_2$  подсеть  $N_2$ , а  $\Gamma$  — гомоморфизм из  $N_1$  в  $N_2$ , получаем:

$$\begin{aligned} \Gamma_Y(g_1(s_1, x_1)) &= g_2(\Gamma_S(s_1), \Gamma_X(x_1)) \in Y'_2, \\ \Gamma_S(h_1(s_1, x_1, y_1)) &= h_2(\Gamma_S(s_1), \Gamma_X(x_1), \Gamma_Y(y_1)) \in S'_2, \end{aligned}$$

откуда  $g_1(s_1, x_1) \in \Gamma_Y^{-1}(Y'_2)$  и  $h_1(s_1, x_1, y_1) \in \Gamma_S^{-1}(S'_2)$  и, значит,  $\Gamma^{-1}(N'_2)$  — подсеть  $N_1$ .  $\square$

Конгруэнцией абстрактной нейросети  $N = (S, X, Y, g, h)$  будем называть семейство  $\Theta = \Theta_S \subseteq S \times S, \Theta_X \subseteq X \times X, \Theta_Y \subseteq Y \times Y$  эквивалентностей на множествах  $S, X, Y$ , обладающее свойством устойчивости относительно операций  $g$  и  $h$ :

$$\begin{aligned} h : S \times X \times Y \rightarrow S : (s_1, s_2) \in \Theta_S \&(x_1, x_2) \in \Theta_X \&(y_1, y_2) \in \Theta_Y \Rightarrow (h(s_1, x_1, y_1), h(s_2, x_2, y_2)) \in \Theta_S, \\ g : S \times X \rightarrow Y : (s_1, s_2) \in \Theta_S \&(x_1, x_2) \in \Theta_X \Rightarrow (g(s_1, x_1), g(s_2, x_2)) \in \Theta_Y. \end{aligned} \quad (11)$$

Факторсеть  $N/\Theta$  мы определим как абстрактную нейросеть  $(S/\Theta_S, X/\Theta_X, Y/\Theta_Y, g, h)$ , операции которой выражаются через одноимённые операции нейросети  $N$  по формуле

$$h(\Theta_S(s), \Theta_X(x), \Theta_Y(y)) := \Theta_S(h(s, x, y)), \quad g(\Theta_S(s), \Theta_X(x)) := \Theta_Y(g(s, x)).$$

Непосредственным переносом из теории многоосновных алгебр мы можем получить следующие важные теоремы.



**Теорема 2.** Пусть  $\Gamma = (\Gamma_S, \Gamma_X, \Gamma_Y)$ ,  $\Gamma_S : S_1 \rightarrow S_2$ ,  $\Gamma_X : X_1 \rightarrow X_2$ ,  $\Gamma_Y : Y_1 \rightarrow Y_2$  — гомоморфизм нейросети  $N_1 = (S_1, X_1, Y_1, g, h)$  в  $N_2 = (S_2, X_2, Y_2, g, h)$ . Тогда его ядро<sup>2</sup>  $\text{Ker } \Gamma := \{\text{Ker } \Gamma_S, \text{Ker } \Gamma_X, \text{Ker } \Gamma_Y\}$  является конгруэнцией нейросети  $N_1$ . С другой стороны, если  $\Theta = \{\Theta_S, \Theta_X, \Theta_Y\}$  — конгруэнция на  $N_1$ , то семейство отображений  $\{\text{nat } \Theta_S, \text{nat } \Theta_X, \text{nat } \Theta_Y\}$  будет гомоморфизмом нейросети  $N_1$  на факторсет  $N_1/\Theta$ .

**Теорема 3 (теорема о гомоморфизмах нейросети).** Если  $\Gamma : N_1 \rightarrow N_2$  — сюръективный гомоморфизм нейросети, то факторсет  $N_1/\text{Ker } \Gamma$  изоморфна нейросети  $N_2$ .

Доказательства этих теорем получаются непосредственным переносом соответствующих рассуждений из [6, § 2.1].

Пусть  $N_1 = (S_1, X_1, Y_1, g_1, h_1)$ ,  $N_2 = (S_2, X_2, Y_2, g_2, h_2)$  — две абстрактные нейросети. Их прямым произведением называется абстрактная нейросеть  $N_1 \times N_2 := (S_1 \times S_2, X_1 \times X_2, Y_1 \times Y_2, g, h)$ , где

$$g((s^1, s^2), (x^1, x^2)) := (g_1(s^1, x^1), g_2(s^2, x^2)),$$

$$h((s^1, s^2), (x^1, x^2), (y^1, y^2)) := (h_1(s^1, x^1, y^1), h_2(s^2, x^2, y^2)),$$

для любых  $(s^1, x^1, y^1) \in S_1 \times X_1 \times Y_1$ ,  $(s^2, x^2, y^2) \in S_2 \times X_2 \times Y_2$ .

## 2. ОПИСАНИЕ НЕЙРОСЕТИ С ПОМОЩЬЮ ГРАФ-СХЕМ

Определение абстрактной нейросети, приведённое выше, даёт нам наиболее общий класс математических моделей нейросетей. Это позволяет без существенных усилий перенести аналогичные алгебраические понятия в теорию нейросетей. Однако для решения прикладных задач нейросеть используется как устройство, вычисляющее выходное значение по входным значениям и текущему состоянию. При таком подходе понятие нейросети необходимо уточнить, определив свойства множеств  $S, X, Y$  и способ задания функций  $g, h$  из (2).

Основные множества нейросети могут быть дискретными или непрерывными. Если множества  $S, X, Y$  — конечные непустые множества состояний, входных и выходных символов соответственно, а  $h$  и  $g$  задают функции переходов и выходов, то нейросеть  $N = (S, X, Y, g, h)$ , по сути, является автоматом Мили, свойства которого хорошо изучены и могут быть применены к такой нейронной сети. В случае непрерывности множеств  $S, X, Y$ , некоторые вопросы требуют особого рассмотрения с переосмыслением ряда важных свойств нейросети.

Для решения прикладных задач наиболее востребованы нейросети с множествами  $S \subseteq R^n$ ,  $X \subseteq R^m$ ,  $Y \subseteq R^p$ , где  $R^n, R^m, R^p$  — линейные векторные пространства размерности  $n, m$  и  $p$  соответственно над полем действительных чисел. Одна из таких задач — распознавание образов. В этой задаче необходимо по набору внешних признаков некоторого объекта определить, к какому классу он относится. В более общем смысле задача распознавания образов заключается в поиске системы правил, которая позволила бы в полностью автоматическом режиме (т. е. без участия эксперта) распознавать входные образы. Нейросетевое решение этой задачи предполагает построение нейросети, которая по входному вектору признаков объекта вычисляет вектор значений, идентифицирующий распознаваемый объект, или определяет его принадлежность некоторому классу. Для такой нейросети  $S$  — это множество допустимых внутренних параметров,  $X$  — множество допустимых векторов признаков распознаваемых объектов, а  $Y$  — множество возможных ответов (классов). Функция  $g$  из определения выступает в роли решающего правила, а функция  $h$  используется для настройки параметров нейросети под конкретные условия задачи. В силу непрерывности к множествам  $S, X, Y$  неприменимы многие из методов дискретной математики (например, булевозначные матрицы для описания переходов между состояниями).

Ещё один класс нейросетей — нейросети, моделирующие нечёткий вывод [3]. Основная задача таких нейросетей — вычисления нечёткого вывода для некоторых фиксированных значениях нечётких переменных. В таких нейросетях множества носителя обычно следующие:  $S \subseteq R^n$ ,  $X \subseteq R^m$ ,  $Y \subseteq [0, 1]^p$ , где  $[0, 1]^p$  — векторы с действительными значениями на единичном кубе размерности  $p$ .

<sup>2</sup>Ядро отображения  $f : A \rightarrow B$  — это отношение  $\text{Ker } f := \{(a_1, a_2) \in A \times A | f(a_1) = f(a_2)\}$  [6, с. 60].



Есть много других задач, где используется модель нейросети как основной метод построения вычислений. Общим для большинства таких задач является применение методов условной или безусловной оптимизации, а также численных методов решения уравнений для поиска оптимальных параметров нейросети. Это накладывает на множества  $S, X, Y$  условие непрерывности, а на функцию  $g(s, x)$  — условие непрерывности и дифференцируемости по всем аргументам.

Рассмотрим теперь способы представления функций нейросети. Функция  $g$  задаёт выход нейросети и может быть задана различными способами. Наиболее популярно определение  $g$  в виде формулы векторно-матричных операторов или граф-схемы функциональных элементов. Первый вариант удобен при моделировании работы нейросети на компьютере при заранее известной функции  $g$ . Второй способ удобен на стадии решения задачи синтеза нейросети из заданных «простых элементов».

Граф-схему функциональных элементов для функции выхода определим в два этапа: на первом этапе приведём структурную часть этого понятия, на втором этапе — функциональную. Для начала определим понятие набора базовых операций — элементарных «блоков», из которых будет строиться вычислительная модель нейросети.

Предположим, что у нас имеется набор операций  $B = \{\beta_1, \beta_2, \dots, \beta_q\}$ . Это могут быть операции, вычисляющие сумму, произведение, возведение в квадрат значений или любые другие операции от любого числа переменных, не превосходящего  $r := n + m + p^3$ . Обычно набор операций определяется рядом конкретных требований, однако он должен быть достаточным для реализации любой из возможных функций нейросети. Например, в данном наборе должны быть нульарные операция присвоения значения переменным. Для обозначения операции присвоения мы можем использовать символ переменной, которой присваивается значение. Кроме того, полезной может оказаться тождественная функция скалярного аргумента:  $\beta(x) := x$ . Базовая операция может быть вектор-функцией, т. е. функцией, значениями которой являются векторы размерности более 1. Например, это может быть унарная операция, которая формирует вектор, используя некоторое скалярное входное значение (рис. 1):  $\beta(x) := \underbrace{(x, x, \dots, x)}_{n \text{ раз}}$ .

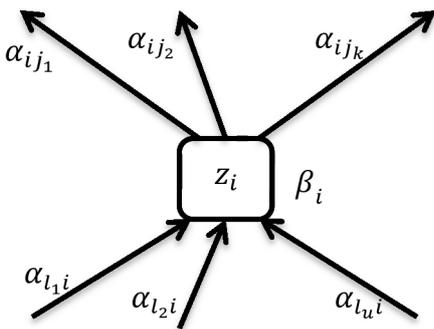


Рис. 1. Вершина графа  $v_i$ , помеченная символом операции  $\beta_i(\alpha_{l_1 i}, \alpha_{l_2 i}, \dots, \alpha_{l_u i})$  и символом переменной  $z_i$

Доказана теорема [7], согласно которой для реализации любой непрерывной действительной функции многих переменных достаточно бинарной операции умножения, сложения, определённых стандартным образом над полем действительных чисел, а также некоторой нелинейной унарной операции. В общем случае подбор минимального набора базовых операций является сложной задачей и выходит за рамки данной статьи. Перейдём к определению структуры нейросети.

Доказана теорема [7], согласно которой для реализации любой непрерывной действительной функции многих переменных достаточно бинарной операции умножения, сложения, определённых стандартным образом над полем действительных чисел, а также некоторой нелинейной унарной операции. В общем случае подбор минимального набора базовых операций является сложной задачей и выходит за рамки данной статьи. Перейдём к определению структуры нейросети.

**I этап. Определение граф-схемы из функциональных элементов с точки зрения её структуры.** Структуру граф-схемы можно задать, используя понятие ориентированного графа [6, с. 227].

Граф-схемой функции выхода нейросети (схемой функционирования нейросети)<sup>4</sup> над базисом  $B = \{\beta_1, \beta_2, \dots, \beta_q\}$  будем называть ориентированный граф<sup>5</sup>  $\Sigma_g = (V_g, E_g)$ , между вершинами которого и множеством переменных  $\{s_1, s_2, \dots, s_n, x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_p\}$  существует взаимно однозначное соответствие. При этом вершины и дуги графа помечаются следующим образом:

1. Каждая вершина помечена символом операции  $\beta_i \in B$ . Количество и порядок дуг, входящих в вершину  $v_i$ , должен соответствовать количеству и порядку аргументов функции  $\beta_i$ ,  $i = 1, 2, \dots, n + m + p$ . При этом источники графа не могут быть помечены символами операций присвоения выходных переменных  $y_k$ .

2. Каждая дуга  $(v_i, v_j) \in E_g$  помечена символом переменной  $\alpha_{ij}$ .

<sup>3</sup> $r := n + m + p$  — это размерность множества  $S \times X \times Y$ .

<sup>4</sup>В литературе, посвящённой нейросетям, часто используют другой термин — «структура» или «архитектура» нейросети. Термин «схема» порождён аналогиями нейроинформатики с дисциплинами, изучающими схемы функциональных элементов [8, 9], а термин «структура» обычно используется в системном анализе [10].

<sup>5</sup>В общем случае граф-схема нейросети может содержать циклы, но в данной статье такие нейросети с обратными связями рассматриваться не будут, поэтому далее будем рассматривать только ациклические графы



Кортеж всех переменных граф-схемы будем обозначать

$$z = (s_1, s_2, \dots, s_n, x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_p) := (z_1, z_2, \dots, z_{(n+m+p)}). \quad (12)$$

Отдельная вершина графа  $\Sigma_g$  изображена на рис. 1 прямоугольником.

Другими словами, граф-схема  $\Sigma_g = (V_g, E_g)$  — ориентированный граф, который состоит из вершин  $v_i := (z_i, \beta_i) \in V_g$  и дуг  $e_{ij} = (v_i, v_j, \alpha_{ij}) \in E_g$ ,  $i, j = 1, 2, \dots, n+m+p$ .

**II этап. Определение вычислений на граф-схеме  $\Sigma_g$ .** Граф-схеме  $\Sigma_g$  сопоставим в соответствие систему функций

$$\left\{ \begin{array}{l} \alpha_{ijk} = \beta_{ik}(\alpha_{l_1i}, \alpha_{l_2i}, \dots, \alpha_{l_u i}), \\ i = 1, 2, \dots, n+m+p, \quad k = 1, 2, \dots, d^+(v_i), \quad u = d^-(v_i) \end{array} \right\}, \quad (13)$$

где  $n, m, p$  — размерности  $S, X, Y$  соответственно,  $(v_i, v_{jk}, \alpha_{ijk}), (v_{l_u}, v_i, \alpha_{l_u i}) \in E_g$ ,  $\beta_{ik}$  — значение  $k$ -й компоненты результата функции  $\beta_i(\alpha_{l_1i}, \alpha_{l_2i}, \dots, \alpha_{l_u i})$ . Вычисления на этом графе определим по индукции.

*Базис индукции.* Сначала вычисляются значения для вершин  $v_i$ , помеченных нульарными операциями присвоения значений переменных  $\beta_i \in \{s_1, s_2, \dots, s_n, x_1, x_2, \dots, x_m\} \in B$ . По определению это будут источники графа, так как количество входящих в вершину дуг равно количеству аргументов операции, приписанной этой вершине. Вычисленные по формулам (13) значения присваиваем переменным  $\alpha_{ij_1}, \alpha_{ij_2}, \dots, \alpha_{ij_r}$ , которыми помечены исходящие из вершин  $v_i$  дуги:

$$\alpha_{ijk} = \beta_{ik}, \quad k = 1, 2, \dots, d^+(v_i), \quad \forall i \text{ таких, что } d^-(v_i) = 0. \quad (14)$$

*Индуктивный переход.* Если для некоторой вершины  $v_i$  определены все значения переменных  $\alpha_{l_1i}, \alpha_{l_2i}, \dots, \alpha_{l_u i}$  для дуг, заходящих в эту вершину, то вычисляем значение соответствующей операции в этой вершине. Полученное значение присваиваем переменным  $\alpha_{ij_1}, \alpha_{ij_2}, \dots, \alpha_{ij_k}$ , которыми помечены исходящие из  $v_i$  дуги:

$$\alpha_{ijk} = \beta_{ik}(\alpha_{l_1i}, \alpha_{l_2i}, \dots, \alpha_{l_u i}), \quad k = 1, 2, \dots, d^+(v_i), \quad \forall i \text{ таких, что } d^-(v_i) > 0. \quad (15)$$

Данное определение задаёт естественный порядок вычислений на графе: вычисления начинаются для вершин источников с присвоения значений переменных  $x$  и  $s$ . Далее вычисляются значения для тех вершин, все заходящие дуги которых являются окончанием маршрутов от источников единичной длины. Потом вычисляются значения в вершинах, все заходящие дуги которых являются окончанием маршрутов от источников длины 2 и т.д. Очевидно, что если в графе нет циклов, то вычисления заканчиваются в вершинах с максимальной длиной маршрута от источников (рис. 2). В случае наличия циклов для остановки вычислений на графе необходим дополнительный критерий, который будет проверяться на каждой итерации вычислений.

Аналогичным образом задаётся граф-схема функции переходов  $h$ , с тем лишь отличием, что начальными значениями в вычислениях могут быть также и значения выходных переменных  $y$ .

*Граф-схемой функции переходов (схемой обучения нейросети)* над базисом  $B = \{\beta_1, \beta_2, \dots, \beta_q\}$  будем называть ориентированный упорядоченный граф  $\Sigma_h = (V_h, E_h)$ , между вершинами которого и множеством переменных  $\{s_1, s_2, \dots, s_n, x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_p\}$  существует взаимно однозначное соответствие. При этом вершины и дуги графа помечаются следующим образом.

1. Каждая вершина помечена символом операции  $\beta_i \in B$ . Заходящие в вершину  $v_i$  дуги упорядочены в соответствии с порядком аргументов  $\beta_i$ . Количество и порядок аргументов  $\beta_i$  соответствует количеству и порядку заходящих в вершину  $v_i$  дуг.

2. Каждая дуга  $(v_i, v_j) \in E_h$  помечена символом переменной  $\alpha_{ij}$ .

Вершины-источники граф-схемы функции перехода могут быть помечены любыми операциями множества  $B$ . Граф-схему, не меняющую вектор выхода или вектор состояния, будем обозначать единицей. Такая граф-схема соответствует тождественным функциям выхода и перехода (рис. 3):

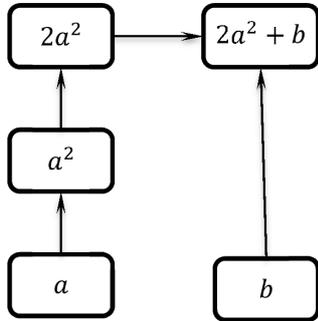
$$1_g : g(s, x) := x;$$



$$1_h : h(s, x) := s.$$

Как видно из определения, граф-схемы нейросети характеризуются структурой — графом определённого вида и функционированием — законом преобразования входных векторов и векторов состояния в выходные. Это позволяет нам для определения нейросети использовать следующие обозначения:

$$N = (S, X, Y, \Sigma_g, \Sigma_h, B), \tag{16}$$



где  $S, X, Y$  — множества состояний, входных векторов и выходных векторов соответственно,  $\Sigma_g$  и  $\Sigma_h$  — граф-схемы функционирования и обучения соответственно, а  $B = \{\beta_1, \beta_2, \dots, \beta_q\}$  — базисный набор операций. При этом вычисления на графах  $\Sigma_g, \Sigma_h$  производятся по правилам (13)–(15).

**Пример.** Пусть у нас есть нейросеть, вычисляющая квадрат аргумента  $N = (\emptyset, R, R, g(a, b) = 2a^2 + b, 1_h)$ . Переменные состояния у этой нейросети отсутствуют, а граф-схему функции выхода можно определить, как показано на рис. 2.

Рис. 2. Граф-схема функции выхода  $g(a, b) = 2a^2 + b$ . В вершинах указана формула для получения результата операции над значениями переменных  $a, b$

Задание функций нейросети граф-схемами более конструктивно, чем через формулы в том смысле, что даёт удобное представление порядка вычислений в нейросети. Однако такое описание нейросети не равнозначно определению (1)–(3), так как одна и та же функция может быть задана несколькими различными способами (граф-схемами).

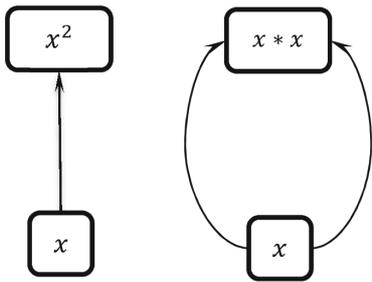


Рис. 3. Две различных граф-схемы, вычисляющие одну и ту же функцию выхода  $g(x) = x^2$

Например, функцию выхода, заданную формулой  $g(x) = x^2$ , можно определить двумя разными графами вычислений (рис. 3). Первый способ использует базис  $B = \{x^2, x\}$  и граф с одной дугой и двумя вершинами, а во втором —  $B = \{*, x\}$  и граф с двумя дугами и двумя вершинами.

Ещё один важный вопрос — любое ли отображение в определении абстрактной нейросети представимо с помощью граф-схем? Ответ на этот вопрос утвердительный. Действительно, если у нас есть нейросеть  $N = (S, X, Y, g, h)$ , то мы можем выбрать базисный набор операций  $B = \{g, h, s_1, s_2, \dots, s_n, x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_p\}$  и определить граф-схемы, как показано на рис. 4.

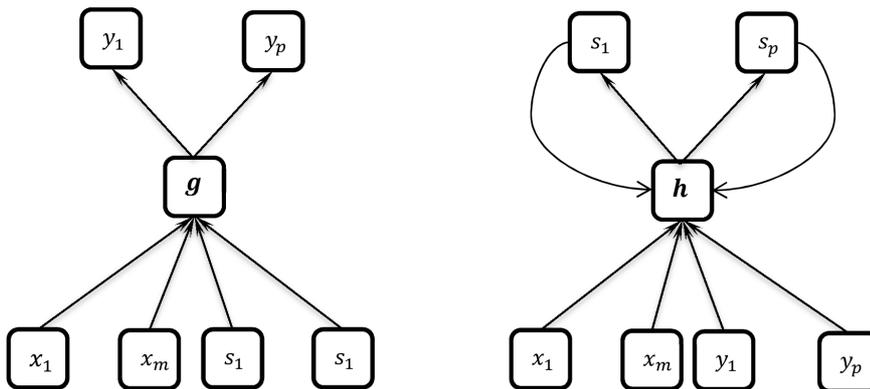


Рис. 4. Тривиальные граф-схемы для реализации функций  $g$  и  $h$

Таким образом, мы доказали

**Утверждение.** Для любой нейросети  $N = (S, X, Y, g, h)$  функции  $g, h$  можно задать с помощью граф-схем  $\Sigma_g, \Sigma_h$  соответственно, однако такое представление может быть не единственным.



## Библиографический список

1. Головкин В. А. Нейронные сети : обучение, организация и применение. Кн. 4 : учеб. пособие для вузов / общ. ред. А. И. Галушкина. М. : ИПРЖР, 2001.
2. Горбань А. Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сиб. журн. вычисл. матем. 1998. Т. 1, № 1. С. 11–24.
3. Круглов В. В., Дли М. И., Голунов Р. Ю. Нечёткая логика и искусственные нейронные сети. М. : Физматлит, 2001.
4. Тархов Д. А. Нейронные сети. Модели и алгоритмы. М. : Радиотехника, 2005.
5. Хайкин С. Нейронные сети : полный курс. М. : Издат. дом «Вильямс», 2006.
6. Богомолов А. М., Салий В. Н. Алгебраические основы теории дискретных систем. М. : Наука, 1997.
7. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одной переменной и сложения // Докл. АН СССР. 1957. Т. 114, № 5. С. 953–956.
8. Горбань А. Н. Обучение нейронных сетей. М. : СП «Параграф», 1990.
9. Алексеев В. Б., Ложкин С. А. Элементы теории графов, схем и автоматов. М. : Издат. отдел ф-та ВМиК МГУ, 2000.
10. Митрофанов Ю. И. Системный анализ. Саратов : Науч. кн., 2000.

## Algebraic Properties of Abstract Neural Network

I. I. Slepovichev

Slepovichev Ivan Ivanovich, Saratov State University, 83, Astrakhanskaya st., Saratov, Russia, 410012, gurgutan@yandex.ru

The modern level of neuroinformatics allows to use artificial neural networks for the solution of various applied problems. However many neural network methods put into practice have no strict formal mathematical substantiation, being heuristic algorithms. It imposes certain restrictions on development of neural network methods of the solution of problems. At the same time there is a wide class of mathematical models which are well studied within such disciplines as theory of abstract algebras, graph theory, automata theory. Opportunity to use results received within these disciplines in relation to neural network models can be a good help in studying of artificial neural networks, their properties and functionality. In this work formulations and definitions of neural network models from the point of view of universal algebra and the theory of graphs are given. The main theorems of universal algebra are provided in neural network treatment. In article is also offered the way of the formal description of a neuronet by graph-schemes which allows to use results of graph theory for the analysis of neural network structures.

*Key words:* neural net, homomorphism, congruence, graph-scheme of a neural network, computation on a graph.

## References

1. Golovko V. A. *Nejronnye seti : obuchenie, organizacija i primenenie* [Neural networks: training, and organizing your application]. Book 4 : Manual for High Schools / ed. A. I. Galushkina, Moscow, IPRZhR, 2001 (in Russian).
2. Gorban A. N. Generalized approximation theorem and computational capabilities of neural networks. *Sib. Zh. Vychisl. Mat.*, 1998, vol. 1, no. 1, pp. 11–24 (in Russian).
3. Kruglov V. V., Dli M. I., Golunov R. U. *Nechjotkaja logika i iskusstvennye nejronnye seti* [Fuzzy Logic and Artificial Neural Network]. Moscow, Fizmatlit, 2001 (in Russian).
4. Tarhov D. A. *Nejronnye seti. Modeli i algoritmy* [Neural network. Models and Algorithms]. Moscow, Radiotekhnika, 2005 (in Russian).
5. Haykin S. *Neural Networks. A Comprehensive Foundation*. Hamilton, Ontario, Canada, 1999.
6. Bogomolov A. M., Salii V. N. *Algebraicheskie osnovy teorii diskretnykh sistem* [Algebraic foundations of the theory of discrete systems]. Moscow, Nauka, 1997 (in Russian).
7. Kolmogorov A. N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Dokl. Akad. Nauk SSSR*, 1957, vol. 114, no. 5, pp. 953–956 (in Russian).
8. Gorban A. N. Training neural networks. Moscow, USSR-USA JV "Paragraph", 1990 (in Russian).
9. Alekseev V. B., Lozhkin S. A. *Elementy teorii grafov, skhem i avtomatov* [Elements of the theory of graphs, charts and machines]. Moscow, Izdat. otel fakulteta VMiK MGU, 2000 (in Russian).
10. Mitrofanov Yu. I. System analysis. Saratov, Nauchnaia kniga, 2000 (in Russian).