МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННАЯ КОРПОРАЦИЯ ПО АТОМНОЙ ЭНЕРГИИ «РОСАТОМ»

РОССИЙСКАЯ АКАДЕМИЯ НАУК

РОССИЙСКАЯ АССОЦИАЦИЯ НЕЙРОИНФОРМАТИКИ

МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
СИСТЕМНЫХ ИССЛЕДОВАНИЙ РАН

НАУЧНАЯ СЕССИЯ МИФИ–2009

НЕЙРОИНФОРМАТИКА–2009

**XI ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ**

ЛЕКЦИИ
ПО НЕЙРОИНФОРМАТИКЕ

По материалам Школы-семинара
«Современные проблемы нейроинформатики»

Москва 2009

УДК 001(06)+004.032.26 (06) Нейронные сети
ББК 72я5+32.818я5
М82

НАУЧНАЯ СЕССИЯ МИФИ–2009. XI ВСЕРОССИЙСКАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ «НЕЙРОИНФОРМАТИКА–2009»: ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ. – М.: МИФИ, 2009. – 280 с.

В книге публикуются тексты лекций, прочитанных на Школе-семинаре «Современные проблемы нейроинформатики», проходившей 27–30 января 2009 года в МИФИ в рамках XI Всероссийской конференции «Нейроинформатика–2009».

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 978–5–7262–1053–7

Содержание

З. М. ШИБЗУХОВ
НИИ прикладной математики и автоматизации,
Кабардино-Балкарский научный центр РАН, г. Нальчик
E-mail: szport@gmail.com

ΣΠ–НЕЙРОННЫЕ СЕТИ: ВВЕДЕНИЕ

Аннотация

Рассмотрена модель ΣΠ–нейрона. Показано, что в модели ΣΠ–нейрона легко строятся прямые комбинаторно-алгебраические процедуры конструктивного обучения ΣΠ–нейронов и простейших ΣΠ–нейронных сетей по обучающим примерам. ΣΠ–нейрон, корректно функционирующий на обучающей последовательности строится за один проход. Применяя несложную процедуру оптимизации, можно строить множества таких ΣΠ–нейронов. Поэтому становится очевидно, что в модели ΣΠ–нейронных сетей становится возможным воплощение конструктивного комбинаторно-алгебраического подхода к обучению нейронных сетей, развитие которого открывает пути для построения прямых теоретически обоснованных алгебраических методов обучения семейств искусственных нейронных сетей, корректно (или квазикорректно) функционирующих на обучающем материале, а также комбинаторного поиска в этих семействах оптимальных искусственных нейронных сетей.

Z. M. SHIBZUKHOV
Institute for Applied Mathematics and Automation,
Kabardino-Balkarian Research Center of the RAS, Nalchik
E-mail: szport@gmail.com

ΣΠ–NEURAL NETWORKS: AN INTRODUCTION

Abstract

A model of ΣΠ–neuron is considered. We show that in this model we are able to construct direct combinatorial-algebraic procedure for constructive learning of ΣΠ–neurons and the simple ΣΠ–neural networks by examples. ΣΠ–neuron, properly functioning on training sequence is constructed in one pass. Using simple optimazation procedure we can build sets of ΣΠ–neurons. We can see that the model of ΣΠ-neural networks becomes possible to adopt meaningful combinatorial-algebraic approach to learning neural networks, development of which opens the way for the direct construction of theoretically based algebraic methods of teaching families for artificial neural networks properly operated on educational material as well as combinatorial search of the best artificial neural networks in these families.

УДК 001(06)+004.032.26 (06) Нейронные сети

Введение

Обучение искусственных нейронных сетей главным образом связано с двумя основными проблемами — проблемой *выбора архитектуры* и проблемой *настройки весов*. Поэтому особую актуальность приобретают такие процедуры обучения, которые решают обе указанные проблемы.

Выбор математической модели нейрона влияет на характер процедуры настройки весов нейрона и на характеристики архитектуры нейронной сети. Так, слабая способность к аппроксимации нейронов с линейной функцией суммарного сигнала вида

$$y = \mathrm{out}\big(\theta + \sum w_i x_i\big)$$

приводит к необходимости многослойной архитектуры.

Применение более сложной математической модели нейрона, которая полнее отражает основные свойства процессов обработки информации в нейронах нервной системы и обладает лучшими способностями по представлению зависимостей, позволяет упростить общую архитектуру сети, а в ряде случаев и ускорить процедуру настройки весов.

ΣΠ–нейрон как раз и представляет собой *алгебраическую модель* нейрона, отражающую процессы обработки информации в аксо-дендритной системе нейрона, которая обладает лучшими способностями по аппроксимации зависимостей.

Один ΣΠ–нейрон в своем классическом варианте имеет полилинейную функцию суммарного сигнала

$$y = \mathrm{out}\Big(\theta + \sum w_k \prod_{i \in \mathbf{i}_k} x_i\Big), \quad \mathbf{i}_k \subseteq \{1, \ldots, n\}$$

и способен представлять произвольные булевы функции.

Один ΣΠ–нейрон от пороговых функций

$$y = \mathrm{out}\Big(\theta + \sum w_k \prod_{i \in \mathbf{i}_k} (x_i - a_{ik})_+\Big)$$

способен представлять произвольные функции p–значной логики (p — простое число).

Один ΣΠ–нейрон от пороговых функций

$$y = \mathrm{out}\Big(\theta + \sum w_k \prod_{i \in \mathbf{i}_k} h(x_i - a_{ik})\Big)$$

способен представлять произвольные дискретные функции, определенные на конечных дискретных подмножествах $\mathbb{Z}^n$.

Один ΣΠ–нейрон от усеченных линейных функций

$$y = \mathrm{out}\Big(\theta + \sum w_k \prod_{i \in \mathbf{i}_k} (x_i - a_{ik})_+\Big)$$

способен аппроксимировать произвольные непрерывные функции, определенные на $[A_1, B_1] \times \ldots \times [A_n, B_n] \subset \mathbb{R}^n$.

ΣΠ–нейроны и ΣΠ–нейронные сети, построенные их основе, активно исследуются и применяются в прикладных задачах, начиная с 90-х годов. Для обучения используют методы обучения, основанные на *градиентных процедурах* минимизации функционала ошибки или на *генетических алгоритмах*. Главный недостаток таких процедур состоит в том, что они имеют невысокую скорость сходимости процесса обучения. Во многих случаях отсутствует доказательство сходимости процедуры обучения: на практике она обеспечивается за счет *эвристического* и/или *экспериментального* подбора параметров обучения и подходящего выбора начального состояния нейронной сети.

Прямые алгебраические процедуры обучения таких нейронных сетей практически *не развиты*.

Основываясь на модели *алгебраического* ΣΠ–*нейрона*, были разработаны прямые комбинаторно-алгебраические процедуры построения ΣΠ–нейронов по обучающей последовательности примеров. За один проход предварительно упорядоченной обучающей последовательности примеров строится ΣΠ–нейрон, корректно функционирующий на ней. Применяя несложную процедуру минимизации рангов мультипликативных слагаемых в процессе обучения, можно построить множества таких ΣΠ–нейронов.

Применяя различные дополнительные ограничения (например, степень корректности функционирования на контрольном множестве примеров), можно отбирать наилучший (или наилучшие) нейрон, применять коллективные решающие правила или строить линейные (квазилинейные) корректирующие операции для повышения способности к обобщению.

В целом развитие комбинаторно-алгебраического подхода к обучению искусственных нейронных сетей открывает пути для построения *прямых* теоретически обоснованных алгебраических методов и алгоритмов обучения *семейств* искусственных нейронных сетей, корректно (или квазикорректно) функционирующих на обучающем материале, а также комбинаторного поиска в этих семействах оптимальных искусственных нейронных

сетей. В рамках этого подхода можно отказаться от трудоемких оптимизационных процедур для обеспечения корректности обученной нейронной сети: корректность достигается в результате прямого построения.

В модели ΣΠ–нейронных сетей становится возможным воплощение конструктивного комбинаторно-алгебраического подхода к обучению для решения общей проблемы обучения нейронных сетей.

Модель ΣΠ–нейрона

Первая модель *искусственного нейрона*, предложенная Мак-Каллоком и Питсом [7], имеет следующий вид:

$$\mathrm{sn}(\boldsymbol{x}) = \mathrm{out}\Big(\theta + \sum w_i x_i\Big),$$

где $\boldsymbol{x} = (x_1, \ldots, x_n)$ — входы, $\mathrm{out}(s)$ — функция выхода, θ — смещение, $w_1, \ldots, w_n$ — веса, $\mathrm{sn}(\boldsymbol{x})$ — функция преобразования нейрона. Это модель Σ–*нейрона*.

Наиболее известные среди них следующие:

- *пороговые нейроны* с функцией выхода

$$\mathrm{out}(s) = \begin{cases} 1, & \text{если } s \geqslant 0, \\ -1, & \text{если } s < 0, \end{cases} \quad \text{или} \quad \mathrm{out}(s) = \begin{cases} 1, & \text{если } s \geqslant 0, \\ 0, & \text{если } s < 0; \end{cases}$$

- *сигмоидальные нейроны* с функцией выхода

$$\mathrm{out}(s) = \frac{1 - \mathrm{e}^{-as}}{1 + \mathrm{e}^{-as}} \quad \text{или} \quad \mathrm{out}(s) = \frac{1}{1 + \mathrm{e}^{-as}}.$$

Модель Σ–нейрона является наиболее простой моделью реального нейрона и отражает достаточно упрощенные представления о структуре нейрона и его связях с другими нейронами. Модель Σ–нейрона основана на ряде допущений относительно структуры и функционирования нейрона:

- все синапсы простые, т. е. в образовании синапса участвует один вход (см. рис. 1);
- вклад синапса пропорционален величине его входа ($w_i x_i$);
- величина суммарного входа нейрона линейно складывается из вкладов синапсов.

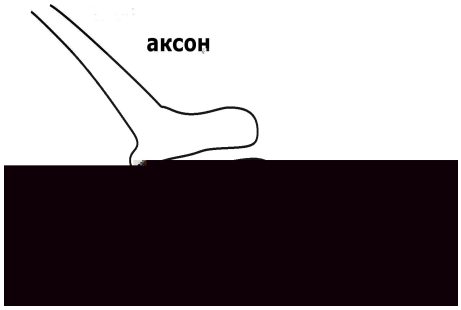


Рис. 1. Аксо-дендритный синапс

Аксо-дендритная система реального нейрона (рис. 2) образует *разветвленную пространственную структуру* с различными типами связей: аксо-дендритными, аксо-соматическими, аксо-аксональными, аксо-синаптическими.

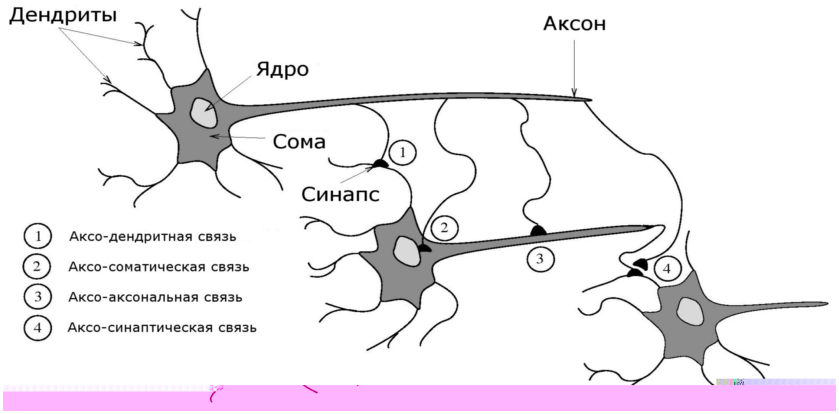


Рис. 2. Природные нейроны и связи между ними

В общем случае в ней могут встречаться *синаптические кластеры*, в образовании которых могут участвовать несколько входов. Вклад каждого входа в кластере может зависеть от величин других входов.

Для того, чтобы отразить эти особенности реальных нейронов в модели

искусственного нейрона необходимо ослабить допущения, положенные в основу модели Σ–нейрона, и расширить ее.

Простейший способ расширения модели Σ–нейрона с математической точки зрения состоит в ослаблении требования линейности вкладов, независимости входов и простоты синапсов и перехода к полилинейным функциям суммарного входного сигнала. Они, будучи линейными по каждому отдельно взятому аргументу (когда значения всех остальных зафиксированы), в то же время являются нелинейными функциями многих переменных.

Принятие полилинейной функции суммирования сигнала отражает в себе следующие простые допущения:

- синапсы могут быть как простыми, так и сложными (синаптические кластеры);
- величина суммарного входа нейрона складывается из величин, пропорциональных вкладам синаптических кластеров;
- величина вклада синаптического кластера представляет полилинейную форму от входов, участвующих в его образовании.

Например, в пресинаптической аксо-дендритной связи (пресинаптическое торможение, рис. 3) вклад может представляться членом вида wx_1x_2.

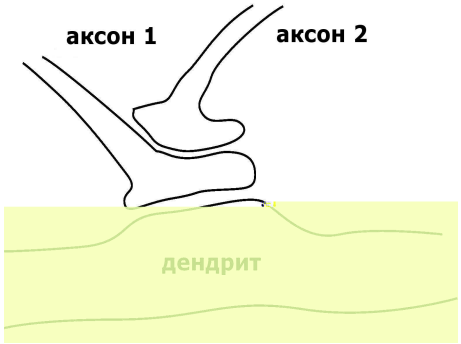


Рис. 3. Пресинаптическое торможение

В синаптическом кластере, показанном на рис. 4, входы не являются независимыми: первый и третий, второй и четвертый, первый и второй, третий и четвертый входы оказывают влияние друг на друга. Поэтому,

вклад синаптического кластера можно представить, например, в виде следующей полилинейной формы:

$$w_1x_1x_2 + w_2x_3x_4 + w_3x_1x_3 + w_4x_2x_4.$$

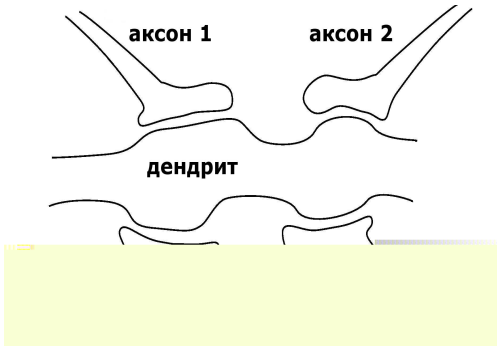


Рис. 4. Синаптический кластер

Модель искусственного нейрона с полилинейной функцией суммарного входа имеет вид:

$$\mathrm{spn}(\boldsymbol{x}) = \mathrm{out}\Big(\theta + \sum w_k \prod_{i\in\mathbf{i}_k} x_i\Big),$$

где $\mathbf{i}_k \subseteq \{1, 2, \ldots, n\}$ — мультииндексы ($\mathbf{i}_k \neq \varnothing$). Такие нейроны будем называть ΣΠ–*нейронами.* Они также известны, как ΣΠ–элементы или порогово-полиномиальные элементы [2, 6, 8, 9].

В такой модели произведения $\prod_{i\in\mathbf{i}_k} x_i$ используются для отражения вклада группы зависимых входов с номерами $i \in \mathbf{i}_k$, которые в совокупности оказывают влияние друг на друга, с одной стороны, а, с другой стороны, дают нулевой вклад, если хотя бы один из входов окажется равным нулю.

ΣΠ–нейрон (рис. 5) можно представлять, как двухслойную сеть с промежуточным слоем из Π–элементов и Σ–нейроном на выходном слое:

$$y = \mathrm{out}\Big(\theta + \sum w_k s_k\Big),$$
$$s_k = \prod_{i\in\mathbf{i}_k} x_i.$$

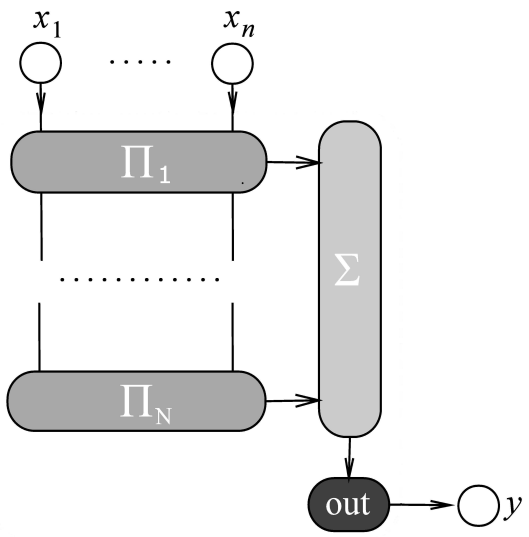


Рис. 5. ΣΠ–нейрон

ΣΠ–нейрон также можно представлять в виде арифметического ΣΠ–дерева, а именно функцию суммарного сигнала

$$\mathrm{sp}(\boldsymbol{x}) = \theta + \sum w_k \prod_{i \in \mathbf{i}_k} x_i$$

можно представить в виде дерева из *билинейных элементов* вида:

$$\mathrm{b}(\boldsymbol{x}, \boldsymbol{u}) = \sum_{i \in \mathbf{i}} u_i x_i + \sum_{j \in \mathbf{j}} w_j x_j + \sum_{t \in \mathbf{t}} c_t u_t,$$

где

$$\boldsymbol{u} = (u_0, u_1, \ldots, u_n), \quad \boldsymbol{x} = (x_1, \ldots, x_n), \quad \mathbf{i} \subseteq \{1, 2, \ldots, n\},$$
$$\mathbf{j} \subseteq \{1, 2, \ldots, n\}, \quad \mathbf{t} \subseteq \{1, 2, \ldots, n\}.$$

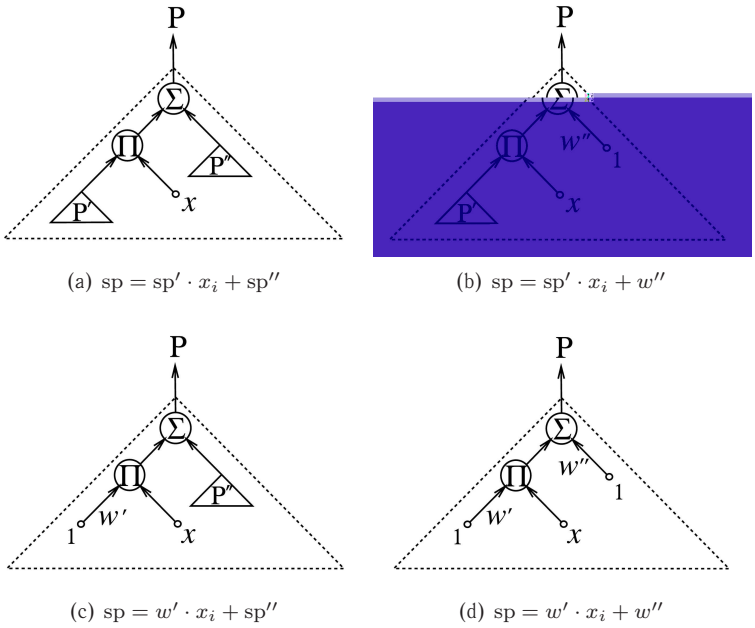
Частным случаем билинейного элемента является элемент, реализующий свертку:

$$\mathrm{b}(\boldsymbol{x}, \boldsymbol{u}) = \sum_{i=1}^{n} u_i x_i.$$

Билинейные элементы отличаются тем, что

- если набор $\boldsymbol{u}$ зафиксирован, то $\mathrm{b}(\boldsymbol{x}, \bullet)$ — линейная функция от $\boldsymbol{x}$;
- если набор $\boldsymbol{x}$ зафиксирован, то $\mathrm{b}(\bullet, \boldsymbol{u})$ — линейная функция от $\boldsymbol{u}$.

Например, такое (каноническое) представление можно получить на основе записи $\mathrm{sp}(\boldsymbol{x})$ в скобочной записи, используя разложения вида, показанного на рис. 6, где sp' и sp'' не зависят от x_i.



(a) $\mathrm{sp} = \mathrm{sp}' \cdot x_i + \mathrm{sp}''$ (b) $\mathrm{sp} = \mathrm{sp}' \cdot x_i + w''$

(c) $\mathrm{sp} = w' \cdot x_i + \mathrm{sp}''$ (d) $\mathrm{sp} = w' \cdot x_i + w''$

Рис. 6. Представление ΣΠ–нейрона в виде дерева

Логико-арифметические ΣΠ–нейроны

Преимущества модели ΣΠ–нейрона по сравнению с моделью Σ–нейрона проявляется в том, что произвольную частичную логическую функцию

$$f : \mathbf{X} \to \{0, 1\}, \quad \mathbf{X} \subseteq \{0, 1\}^n$$

можно представить при помощи единственного логико-арифметического ΣΠ–нейрона, который представляет собой ΣΠ–нейрон $\mathrm{out}(s)$, принимающей значения в $\{0,1\}$. Например, пороговая функция

$$\mathsf{h}(s) = \begin{cases} 1, & \text{если } s \geqslant 0, \\ 0, & \text{если } s < 0. \end{cases}$$

Пример 1. Логическая функция XOR («исключающее или»):

$x_1 \hat{} x_2$	x_1	x_2
0	0	0
1	0	1
1	1	0
0	1	1

представляется в виде:

$$x_1 \hat{} x_2 = \mathsf{h}(-1 + x_1 + x_2 - 2x_1x_2)$$

или в виде двухслойной сети:

$$x_1 \hat{} x_2 = \mathsf{h}(-1 + s_1 + s_2 - 2s_3),$$
$$s_1 = x_1, \quad s_2 = x_2, \quad s_3 = x_1x_2,$$

или в виде композиции билинейных элементов:

$$x_1 \hat{} x_2 = \mathsf{h}(u_1 + u_2x_1),$$
$$u_1 = -1 + x_1, \quad u_2 = 1 - 2x_1.$$

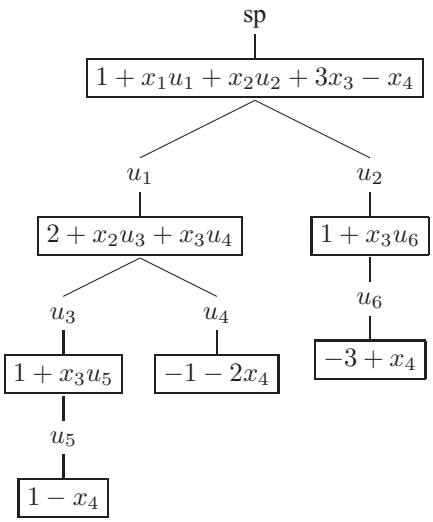
Пример 2. Пусть ΣΠ-форма нейрона имеет вид:

$$\begin{aligned} \mathrm{sp}(x_1, \ldots, x_4) &= 1 + 2x_1 + x_2 + 3x_3 - x_4 + \\ &+ x_1x_2 - x_2x_3 - 3x_2x_3 - x_1x_3 + \\ &+ x_1x_2x_3 - 2x_1x_3x_4 + x_2x_3x_4 - x_1x_2x_3x_4. \end{aligned}$$

Представление в скобочной записи имеет вид:

$$\begin{aligned}\mathrm{sp}(x_1,\ldots,x_4) &= 1 + \underbrace{x_1(2 + \underbrace{x_2(1 + \underbrace{x_3(\underbrace{1 - x_4}_{u_5})}_{u_3}) + x_3(\underbrace{-1 - 2x_4}_{u_4})}_{u_1}) +}\\ &+ \underbrace{x_2(1 + x_3(\underbrace{-3 + x_4}_{u_6}))}_{u_2} + 3x_3 - x_4.\end{aligned}$$

Представление в форме композиции билинейных элементов имеет вид:



$$\begin{aligned}\mathrm{sp}(x_1,\ldots,x_4) &= 1 + x_1u_1 + x_2u_2 + 3x_3 - x_4,\\ u_1 &= 2 + x_2u_3 + x_3u_4,\\ u_2 &= 1 + x_3u_6,\\ u_3 &= 1 + x_3u_5,\\ u_4 &= -1 - 2x_4,\\ u_5 &= 1 - x_4,\\ u_6 &= -3 + x_4.\end{aligned}$$

Обучение

Обучить логико-арифметический ΣΠ–нейрон представлять логическую функцию очень просто. Пусть задана непротиворечивая таблица истинности $\mathbf{T} = \{(y_k, \mathbf{x}_k)\}$, $k = 1, 2, \ldots, N$:

y	x_1	$\ldots$	x_n
y_1	x_{11}	$\ldots$	x_{1n}
$\ldots$	$\ldots$	$\ldots$	$\ldots$
y_N	x_{N1}	$\ldots$	x_{Nn}

где $\mathrm{y}_k \in \{0, 1\}$, $\mathbf{x}_k \in \{0, 1\}^n$. Перед обучением ее строки упорядочиваются в порядке возрастания значения суммы $|\mathbf{x}| = x_1 + \cdots + x_n$ (количество «единиц» в $\mathbf{x}$).

В процессе обучения строится последовательность логико-арифметических ΣΠ–нейронов $\{\mathrm{spn}_k(\boldsymbol{x})\}$. Процесс построения рекуррентный:

1) В начале процесса $\mathrm{sp}_0(\boldsymbol{x}) = \theta$,
2) На k–ом шаге $\mathrm{sp}_k(\boldsymbol{x}) = \mathrm{sp}_{k-1}(\boldsymbol{x}) + w_k \prod\limits_{i \in \mathbf{i}_k^0} x_i$, где $\mathbf{i}_k^0 = \{i : x_{ki} = 1\}$.

Вес w_k находится следующим образом:

$$w_k = \begin{cases} s_k - \mathrm{sp}_{k-1}(\mathbf{x}_k), & \text{если } \mathrm{spn}_{k-1}(\mathbf{x}_k) \neq \mathrm{y}_k, \\ 0, & \text{если } \mathrm{spn}_{k-1}(\mathbf{x}_k) = y_k, \end{cases}$$

где s_k — произвольное значение, такое что $\mathrm{out}(s_k) = \mathrm{y}_k$. Например, если out — пороговая функция, то можно положить $s_k = \mathrm{y}_k - 1$ или $s_k = 2\mathrm{y}_k - 1$.

После N шагов искомый логико-арифметический ΣΠ–нейрон построен. Этот очень простой алгоритм обучения, в случае, когда out — пороговая функция, был первоначально предложен Тимофеевым [3].

Пример построения ΣΠ–нейрона. Рассмотрим простой пример построения ΣΠ–нейрона.

Обучающая таблица

k	y	x_1	x_2	x_3	x_4	x_5
1	1	1	1	1	0	0
2	0	0	1	1	1	0
3	1	0	0	1	1	1
4	0	1	0	0	1	1
5	1	1	1	0	0	1

В начале: $\mathrm{sp}_0(\boldsymbol{x}) = 0$.

Шаг 1. Имеем: $\mathrm{y}_1 = 1$, $\mathbf{x}_1 = (1,1,1,0,0)$, $\mathrm{sp}_0(\boldsymbol{x}) = 0$.
$\mathrm{sp}_0(\mathbf{x}_1) = 0$, $\mathrm{spn}_0(\mathbf{x}_1) = 1 \Rightarrow \mathrm{sp}_1(\boldsymbol{x}) = \mathrm{sp}_0(\boldsymbol{x}) = 0$.
Итак, $\mathrm{sp}_1(\boldsymbol{x}) = 0$.

Шаг 2. Имеем: $\mathrm{y}_2 = 0$, $\mathbf{x}_2 = (0,1,1,1,0)$, $\mathrm{sp}_1(\boldsymbol{x}) = 0$.
$\mathrm{sp}_1(\mathbf{x}_2) = 0$, $\mathrm{spn}_1(\mathbf{x}_2) = 1 \Rightarrow \mathrm{sp}_2(\boldsymbol{x}) = \mathrm{sp}_1(\boldsymbol{x}) + w_2 x_2 x_3 x_4$,
$w_2 = -1 - \mathrm{sp}_1(\mathbf{x}_2) = -1$.
Итак, $\mathrm{sp}_2(\boldsymbol{x}) = -x_2 x_3 x_4$.

Шаг 3. Имеем: $\mathrm{y}_3 = 1$, $\mathbf{x}_3 = (0,0,1,1,1)$, $\mathrm{sp}_2(\boldsymbol{x}) = -x_2 x_3 x_4$.
$\mathrm{spn}_2(\mathbf{x}_3) = 1 \Rightarrow \mathrm{sp}_3(\boldsymbol{x}) = \mathrm{sp}_2(\boldsymbol{x})$.
Итак, $\mathrm{sp}_3(\boldsymbol{x}) = -x_2 x_3 x_4$.

Шаг 4. Имеем: $\mathrm{y}_4 = 0$, $\mathbf{x}_4 = (1,0,0,1,1)$, $\mathrm{sp}_3(\boldsymbol{x}) = -x_2 x_3 x_4$.
$\mathrm{sp}_3(\mathbf{x}_4) = 0$, $\mathrm{spn}_3(\mathbf{x}_4) = 1 \Rightarrow \mathrm{sp}_4(\boldsymbol{x}) = \mathrm{sp}_3(\boldsymbol{x}) + w_4 x_1 x_4 x_5$,
$w_4 = -1 - \mathrm{sp}_3(\mathbf{x}_4) = -1$.
Итак, $\mathrm{sp}_4(\boldsymbol{x}) = -x_2 x_3 x_4 - x_1 x_4 x_5$.

Шаг 5. Имеем: $\mathrm{y}_5 = 1$, $\mathbf{x}_5 = (1,1,0,0,1)$, $\mathrm{sp}_4(\boldsymbol{x}) = -x_2 x_3 x_4 - x_1 x_4 x_5$.
$\mathrm{sp}_4(\mathbf{x}_5) = 0$, $\mathrm{spn}_4(\mathbf{x}_5) = 1 \Rightarrow \mathrm{sp}_5(\boldsymbol{x}) = \mathrm{sp}_4(\boldsymbol{x})$.
Итак, $\mathrm{sp}_5(\boldsymbol{x}) = -x_2 x_3 x_4 - x_1 x_4 x_5$.

Результат: $\mathrm{spn}(\boldsymbol{x}) = \mathrm{h}\big(-x_2 x_3 x_4 - x_1 x_4 x_5\big)$.

Секрет простоты обучения. Простота обучения — следствие *треугольности матрицы*

$$\begin{pmatrix} p_{11} & \cdots & p_{1k} & \cdots & p_{1N} \\ 0 & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & p_{kk} & \cdots & p_{kN} \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & p_{NN} \end{pmatrix},$$

где

$$p_{kj} = \mathrm{pn}(\mathbf{x}_j, \mathbf{i}_k) = \prod_{i \in \mathbf{i}_k} \mathrm{x}_{kj} \,.$$

Это означает, что $\{\mathrm{pn}(\boldsymbol{x}, \mathbf{i}_k)\}$ — треугольно упорядоченная на $\{\mathbf{x}_k\}$.

Определение. Последовательность $\{\mathrm{pn}(\boldsymbol{x}, \mathbf{i}_k)\}$ — *треугольно упорядоченная* на $\{\mathbf{x}_k\}$, если

- $\mathrm{pn}(\mathbf{x}_j, \mathbf{i}_k) = 0$ для всех $j < k$;
- $\mathrm{pn}(\mathbf{x}_k, \mathbf{i}_k) \neq 0$ для всех k.

Упорядоченности последовательности $\{\mathbf{x}_k\}$ по возрастанию

$$|\boldsymbol{x}| = x_1 + \ldots + x_n$$

достаточно для треугольной упорядоченности $\left\{\mathrm{pn}(\boldsymbol{x}, \mathbf{i}_k^0)\right\}$ на ней, где

$$\mathbf{i}_k^0 = \{i \mid x_{ki} \neq 0\}.$$

Чувствительность и сложность

Чувствительность ΣΠ–нейрона к малым изменениям значений входов зависит от величины его структурной сложности

$$\langle \mathrm{spn} \rangle = \sum |\mathbf{i}_k|.$$

Так, если

$$|\Delta x_i| \leqslant \delta,$$

то

$$|\Delta \mathrm{sp}(\boldsymbol{x})| = O\big(\langle \mathrm{spn} \rangle\big)\delta.$$

Минимизация сложности

Оказывается, что во многих случаях можно построить логико-арифметические ΣΠ–нейроны, имеющие

- существенно меньшую сложность ($|\mathbf{i}_k| < |\mathbf{i}_k^0|$ для большинства k);
- правильно функционирующий на таблице $\mathbf{T}$.

Процедура обучения остается неизменной, если вместо $\mathbf{i}_k^0$, взять мультииндексы $\mathbf{i}_k \subset \mathbf{i}_k^0$, такие что для всех $j = 1, 2, ..., k-1$

$$\prod_{i \in \mathbf{i}_k} x_{ji} = 0.$$

Для поиска таких мультииндексов можно для каждого k применить процедуру исключения из $\mathbf{i}_k^0$ «несущественных» индексов [1, 4].

Таким образом алгоритм обучения, дополненный процедурой исключения «несущественных» входов, участвующих в произведениях $\prod_{i\in\mathbf{i}_k} x_i$, позволяет строить *множества* логико-арифметических ΣΠ–нейронов, правильно функционирующих на обучающей таблице $\mathbf{T}$ и имеющих потенциально минимальную структурную сложность.

Пример 3. Получение различных ΣΠ–нейронов за счет варьирования возможных вариантов выбора произведения $\mathrm{pn}(\mathbf{x};\mathbf{i}_k)$.

k	y	x_1	x_2	x_3	x_4	x_5	
1	1	1	1	1	0	0	$x_1\vert x_2\vert x_3$
2	0	0	1	1	1	0	x_4
3	1	0	0	1	1	1	x_5
4	0	1	0	0	1	1	$x_1x_4\vert x_1x_5$
5	1	1	1	0	0	1	$x_1x_5\vert x_2x_5$

В последнем столбце данной таблицы указаны возможные варианты выбора произведения $\mathrm{pn}(\mathbf{x};\mathbf{i}_k)$. Комбинируя разные варианты можно в общей сложности построить 12 ΣΠ–нейронов:

$$\mathrm{sp}(\boldsymbol{x}) = w_1 \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} + w_2x_4 + w_3x_5 + w_4 \begin{matrix} x_1x_4 \\ x_1x_5 \end{matrix} + w_5 \begin{matrix} x_1x_5 \\ x_2x_5 \end{matrix}$$

Модель логико-арифметического ΣΠ–нейромодуля

Классификация с непересекающимися классами

Один логико-арифметический ΣΠ–нейрон позволяет эффективно разделять конечные множества $\mathbf{X} \subseteq \{0,1\}^n$ на два класса. Для разделения его на q непересекающихся классов ($q > 2$) «в эпоху однослойного перцептрона» строили сеть из q нейронов так, чтобы j–й ($1 \leqslant j \leqslant q$) Σ–нейрон

$$y_j = \mathrm{sn}_j(\boldsymbol{x})$$

реализовывал бы характеристическую функцию j–го класса $\mathbf{C}_j \subset \mathbf{X}$. Однако это возможно далеко не всегда.

На помощь приходит модель однослойной сети с конкурирующим функционированием. Наиболее известная из них — модель однослойной сети с

функционированием по правилу WTA (Winner Takes All — «победитель забирает всё»):

$$y_j = \mathsf{h}\big(s_j - \max\{s_1, \ldots, s_q\}\big),$$
$$s_j = \mathrm{s}_j(x_1, \ldots, x_n) = \theta_j + \sum w_{ji} x_i.$$

Здесь $y_j = 1$ для нейрона, у которого значение суммарного сигнала — максимальное. В такой модели удается решать некоторые задачи разделения на непересекающиеся классы, которые непосильны для однослойного перцептрона.

Модель ΣΠ–нейромодуля с конкурирующим функционированием по правилу WTA

$$y_j = \mathsf{h}\big(s_j - \max\{s_1, \ldots, s_q\}\big),$$
$$s_j = \mathrm{sp}_j(x_1, \ldots, x_n)$$

позволяет разделять любое подмножество $\mathbf{X}$ на q непересекающихся классов. ΣΠ–нейромодуль с конкурирующим функционированием по правилу WTA обучается так же легко, как и единственный логико-арифметический ΣΠ–нейрон.

Общая задача классификации

Произвольная задача q–классовой классификации множеств $\mathbf{C}_j \subset \mathbf{X} \subseteq \{0, 1\}$ (множества $\mathbf{C}_j$ могут пересекаться между собой), где $j = 1, 2, \ldots, q$, под силу однослойной сети из независимо функционирующих логико-арифметических ΣΠ–нейронов

$$\mathbf{spn}(\boldsymbol{x}) = \big(\mathrm{spn}_1(\boldsymbol{x}), \ldots, \mathrm{spn}_q(\boldsymbol{x})\big),$$

где $\mathrm{spn}_j(\boldsymbol{x})$ представляет собой характеристическую функцию j–го класса. Каждый нейрон обучается независимо от других нейронов.

Однослойную сеть логико-арифметических ΣΠ–нейронов можно представить как двухслойную сеть со скрытым слоем из Π–элементов:

$$y_j = \mathrm{sn}_j(s_1, \ldots, s_N), \quad j = 1, 2, \ldots, q,$$
$$s_k = \prod_{i \in \mathbf{i}_k} x_i, \quad k = 1, 2, \ldots, N.$$

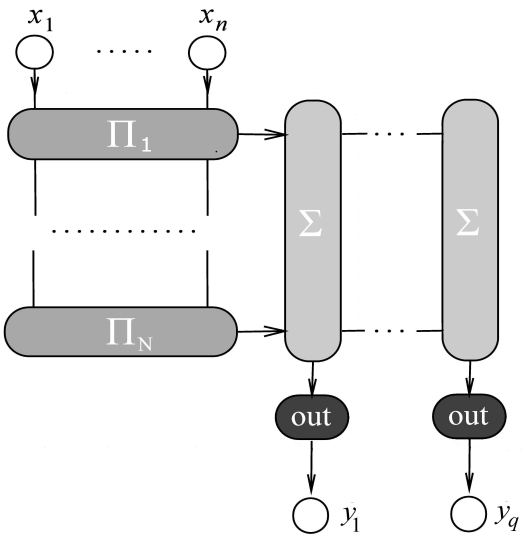


Рис. 7. Слой ΣΠ–нейронов

Модель слоя логико-арифметических ΣΠ–нейронов (рис. 7) с независимым и конкурирующим функционированием по правилу WTA обобщается до модели логико-арифметического ΣΠ–нейромодуля:

$$\boldsymbol{y} = \mathbf{out}\big(\mathbf{sp}(\boldsymbol{x})\big),$$

где

$$\boldsymbol{y} = (y_1, \ldots, y_m), \quad \boldsymbol{x} = (x_1, \ldots, x_n), \quad \mathbf{sp}(\boldsymbol{x}) = \big(\mathrm{sp}_1(\boldsymbol{x}), \ldots, \mathrm{sp}_m(\boldsymbol{x})\big).$$

Преобразование

$$\mathbf{out} = \big(\mathrm{out}_1(\boldsymbol{s}), \ldots, \mathrm{out}_m(\boldsymbol{s})\big),$$

где

$$\mathrm{out}_j(\boldsymbol{s}), \quad \boldsymbol{s} = (s_1, \ldots, s_m),$$

принимает значения в $\{0, 1\}$ и представляет правило конкурирующего функционирования m логико-арифметических ΣΠ–нейронов.

Можно привести примеры правил функционирования:

(1) $\mathrm{out}_j(\boldsymbol{s}) = \mathrm{out}_j(s_j)$ («независимое функционирование»);

(2) $\mathrm{out}_j(\boldsymbol{s}) = \mathsf{h}\big(s_j - (1-\delta)\max\{s_1,\dots,s_m\}\big)$ («только лучшие»);

(3) $\mathrm{out}_j(\boldsymbol{s}) = \mathsf{h}\big(s_j - (1+\delta)\mathrm{med}\{s_1,\dots,s_m\}\big)$ («все, кто лучше среднего»);

Здесь

$$0 < \delta \ll 1, \quad \mathrm{med}\{s_1,\dots,s_m\} = \sum \alpha_j s_j, \quad \sum \alpha_j = 1.$$

Общий метод обучения логико-арифметического ΣΠ–нейромодуля так же прост, как и метод обучения логико-арифметического ΣΠ–нейрона.

Обучение осуществляется по непротиворечивой таблице истинности $\mathbf{T} = \{\langle \mathbf{y}_k, \mathbf{x}_k\rangle\}$, $\mathbf{y}_k = (\mathrm{y}_{k1},\dots,\mathrm{y}_{km})$, $\mathbf{x}_k = (\mathrm{x}_{k1},\dots,\mathrm{x}_{kN})$, $k = 1,2,\dots,N$:

y_1	$\dots$	y_ℓ	x_1	$\dots$	x_n
y_{11}	$\dots$	y_{1m}	x_{11}	$\dots$	x_{1n}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
y_{N1}	$\dots$	y_{Nm}	x_{N1}	$\dots$	x_{Nn}

где $\mathbf{y}_k \in \{0,1\}^m$, $\mathbf{x}_k \in \{0,1\}^n$. Перед обучением ее строки также упорядочиваются в порядке возрастания значения суммы $|\mathbf{x}| = x_1 + \dots + x_n$.

В процессе обучения строится последовательность ΣΠ–нейромодулей

$$\big\{\mathbf{spn}_k(\boldsymbol{x}) = \mathbf{out}\big(\mathbf{sp}_k(\boldsymbol{x})\big)\big\}, \quad k = 0,1,2,\dots$$

Процесс построения рекуррентный.

Вначале

$$\mathbf{sp}_0(\boldsymbol{x}) = (\theta_1, ..., \theta_q).$$

На k–ом шаге вычисляется значение $\mathbf{spn}_{k-1}(\mathbf{x}_k)$. Если

$$\mathbf{spn}_{k-1}(\mathbf{x}_k) = \mathbf{y}_k,$$

то

$$\mathbf{sp}_k(\boldsymbol{x}) = \mathbf{sp}_{k-1}(\boldsymbol{x}).$$

Если

$$\mathbf{spn}_{k-1}(\mathbf{x}_k) \neq \mathbf{y}_k,$$

то некоторым образом выбирается набор

$$\mathbf{s}_k = (\mathrm{s}_{k1}, ..., \mathrm{s}_{kq}),$$

такой что $\mathbf{out}(\mathbf{s}_k) = \mathbf{y}_k$.[1] Затем полагается

$$\mathbf{sp}_k(\boldsymbol{x}) = \mathbf{sp}_{k-1}(\boldsymbol{x}) + \mathbf{w}_k \prod_{i \in \mathbf{i}_k} x_i,$$

где $\mathbf{w}_k = \mathbf{s}_k - \mathbf{sp}_{k-1}(\mathbf{x}_k)$.

После N шагов искомый ΣΠ–нейромодуль построен. Этот простой алгоритм обучения был предложен в [4, 5].

Рекуррентный Σ–нейрон

Рекуррентный Σ–нейрон реализует преобразование в дискретном времени $t = 0, 1, 2, \ldots$ и, в отличие от обычного Σ–нейрона, имеет обратные связи:

$$y(t) = \mathrm{sn}\big(x_1(t), \ldots, x_n(t), y(t-1), \ldots, y(t-\ell)\big),$$

где $x_1(t), ..., x_n(t)$ — входы в момент времени t, $y(t), y(t-1), \ldots, y(t-\ell)$ — значения на выходе в моменты времени $t, t-1, \ldots, t-\ell$, соответственно; ℓ — глубина связей по времени. Вначале, при $t = 0$, заданы $y(-1), \ldots, y(-\ell)$.

Рекуррентный логико-арифметический ΣΠ–нейрон

Рекуррентный логико-арифметический ΣΠ–нейрон (рис. 8) является рекуррентным аналогом обычного логико-арифметического ΣΠ–нейрона:

$$y(t) = \mathrm{spn}\big(x_1(t), \ldots, x_n(t), y(t-1), \ldots, y(t-\ell)\big).$$

Обучение рекуррентного логико-арифметического ΣΠ–нейрона можно свести к обучению обычного логико-арифметического ΣΠ–нейрона

$$\mathrm{spn}\big(x_1, \ldots, x_n, u_1, \ldots, u_\ell\big),$$

где

$$u_1(t) = y(t-1), \ldots, u_\ell(t) = y(t-\ell)$$

[1] Если при некоторых j получается $\mathrm{spn}_{kj}(\mathbf{x}_k) = \mathrm{y}_{kj}$, то, как правило, $\mathbf{s}_k$ стараются выбирать так, чтобы $\mathrm{out}_j(\mathbf{s}_k) = \mathrm{y}_{kj}$.

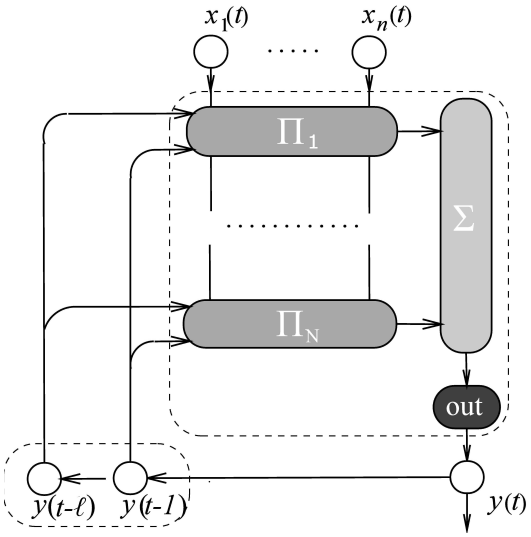


Рис. 8. Рекуррентный ΣΠ–нейрон

по таблице

$$\mathbf{T} = \big\{\big\langle \mathrm{y}(t), \big(\mathrm{x}_1(t), \ldots, \mathrm{x}_n(t), \mathrm{y}(t-1), \ldots, \mathrm{y}(t-\ell)\big)\big\rangle\big\}$$

$y(t)$	$x_1(t)$	$\ldots$	$x_n(t)$	$y(t-1)$	$\ldots$	$y(t-\ell)$
$\mathrm{y}(1)$	$\mathrm{x}_1(1)$	$\ldots$	$\mathrm{x}_n(1)$	$\mathrm{y}(1)$	$\ldots$	$\mathrm{y}(1-\ell)$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$\mathrm{y}(T)$	$\mathrm{x}_1(T)$	$\ldots$	$\mathrm{x}_n(T)$	$\mathrm{y}(T-1)$	$\ldots$	$\mathrm{y}(T-\ell)$

в случае непротиворечивости $\mathbf{T}$.

Рекуррентный слой Σ–нейронов

Рекуррентный Σ–нейронов, известный как RTRN (*Real Time Recurrent Network*),

$$y_j(t) = \mathrm{sn}_j\big(x_1(t), \ldots, x_n(t), y_1(t-1), \ldots, y_m(t-1)\big), \quad j = 1, 2, \ldots, m$$

получает на входе $x_1, \ldots, x_n$ и значения

$$y_1(t-1), \ldots, y_m(t-1),$$

вычисленные в предыдущий момент времени $t-1$; здесь

$$y_1(0), \ldots, y_m(0)$$

представляют собой заданные значения выходов слоя в начальный момент времени $t = 0$.

Рекуррентный слой логико-арифметических $\Sigma\Pi$–нейронов

Рекуррентный слой логико-арифметических $\Sigma\Pi$–нейронов

$$y_j(t) = \mathrm{spn}_j\big(x_1(t), \ldots, x_n(t), y_1(t-1), \ldots, y_m(t-1)\big), \quad j = 1, 2, \ldots, m$$

рекуррентным аналогом слоя логико-арифметических $\Sigma\Pi$–нейронов. Поскольку слой логико-арифметических $\Sigma\Pi$–нейронов можно рассматривать как двухслойную сеть, то рекуррентный слой логико-арифметических $\Sigma\Pi$–нейронов можно считать аналогом рекуррентной двухслойной сети из $\Sigma\Pi$–нейронов:

$$y_j(t) = \mathrm{sn}_j\big(u_1(t), \ldots, u_L(t)\big), \quad j = 1, 2, ..., m;$$
$$u_p(t) = \mathrm{sn}_p\big(x_1(t), \ldots, x_n(t), y_1(t-1), \ldots, y_m(t-1)\big), \quad p = 1, 2, \ldots, L.$$

Обучение рекуррентного логико-арифметического $\Sigma\Pi$–нейрона можно свести к обучению обычного слоя логико-арифметических $\Sigma\Pi$–нейронов $y_j = \mathrm{spn}_j\big(x_1, \ldots, x_n, u_1, \ldots, u_\ell\big)$, где $u_1(t) = y(t-1), \ldots, u_\ell(t) = y(t-\ell)$ по таблице

$$\mathbf{T} = \big\{\big\langle\big(\mathrm{y}_1(t), \ldots, \mathrm{y}_m(t)\big), \big(\mathrm{x}_1(t), \ldots, \mathrm{x}_n(t), \mathrm{y}(t-1), \ldots, \mathrm{y}(t-\ell)\big)\big\rangle\big\}$$

$y_1(t)$	$\ldots$	$y_m(t)$	$x_1(t)$	$\ldots$	$x_n(t)$	$y(t-1)$	$\ldots$	$y(t-\ell)$
$\mathrm{y}_1(1)$	$\ldots$	$\mathrm{y}_m(1)$	$\mathrm{x}_1(1)$	$\ldots$	$\mathrm{x}_n(1)$	$\mathrm{y}(1)$	$\ldots$	$\mathrm{y}(1-\ell)$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$\mathrm{y}_1(T)$	$\ldots$	$\mathrm{y}_m(T)$	$\mathrm{x}_1(T)$	$\ldots$	$\mathrm{x}_n(T)$	$\mathrm{y}(T-1)$	$\ldots$	$\mathrm{y}(T-\ell)$

в случае непротиворечивости $\mathbf{T}$.

Заключение

Итак, мы рассмотрели модель модель ΣΠ–*нейрона* и показали, что в модели ΣΠ–нейрона легко строятся прямые комбинаторно-алгебраические процедуры конструктивного обучения ΣΠ–нейронов и простейших ΣΠ–нейронных сетей по обучающим примерам. ΣΠ–нейрон, корректно функционирующий на обучающей последовательности строится за один проход. Применяя несложную процедуру оптимизации, можно строить множества таких ΣΠ–нейронов.

Таким образом в модели ΣΠ–нейронных сетей становится возможным воплощение конструктивного комбинаторно-алгебраического подхода к обучению нейронных сетей, развитие которого открывает пути для построения *прямых* теоретически обоснованных алгебраических методов обучения *семейств* искусственных нейронных сетей, корректно (или квазикорректно) функционирующих на обучающем материале, а также комбинаторного поиска в этих семействах оптимальных искусственных нейронных сетей.

Литература

1. *Авсаркисян Г. С.* Рекуррентные полиномиальные формы частичных булевых функций // *Известия АН СССР. Техническая кибернетика*. – 1987. – № 4. – с. 131–135.
2. *Тимофеев А. В.* Об одном классе полиномиальных решающих функций в задачах распознавания и диагностики // *Методы вычислений*. - Л.: Изд-во ЛГУ, 1971. - т. 7. – с. 106–121.
3. *Тимофеев А. В., Пшибихов А. В.* Алгоритмы обучения и минимизации сложности полиномиальных распознающих систем // *Изв. АН СССР. Техн. кибернетика*. – 1974. – № 5. – с. 214–217.
4. *Шибзухов З. М.* Рекуррентные методы для конструктивного обучения нейронных сетей из логико-арифметических сигма-пи нейронов // *Нейрокомпьютеры: Разработка и применение*. – 2002. – № 5–6. – с. 50–57.
5. *Шибзухов З. М.* Рекуррентный метод конструктивного обучения некоторых сетей алгебраических $\sum\prod$-нейронов и $\sum\prod$-нейромодулей // *Журнал вычислительной математики и математической физики*. – 2003. – т. 43, № 8. – с. 1298–1310.
6. *Gurney K. N.* Training nets of hardware realizable sigma-pi units // *Neural Networks*. – 1992. – Vol. 5. – pp. 289–303.

7. *McCulloch W., Pitts W.* Logical calculus of ideas immanent in nervous activity // *Bulletin of Mathematical Biophysics*. – 1943. – No. 7. – pp. 115–133.

8. *Mel B. W.* The sigma-pi column: A model of associative learning in cerebral neocortex. – California Institute of Technology. CNS Memo No. 6: Tech. Rep. – Pasadena, California 91125: 1990.

9. *Rumelhart D. E., Hinton G., Williams R.* Learning internal representation by error propagation // Parallel Distributed Processing. – Cambridge, MA: MIT Press, 1986. – Vol. 1. Foundations. – pp. 318–362.

Заур Мухадинович ШИБЗУХОВ, доктор физико-математических наук, ведущий научный сотрудник Отдела интеллектуализации информационных и управляющих систем НИИ прикладной математик и автоматизации КБНЦ РАН (г. Нальчик). Область научных интересов — математические основы и прикладные вопросы нейроинформатики, математическая теория распознавания и прогнозирования, интеллектуальные системы, основанные на знаниях. Автор более 60 публикаций и одной монографии.