

■ Лекция 1: Введение в теорию нейронных сетей.....	2
■ Лекция 2: Основы нейробиологии.....	13
■ Лекция 3: Модели нейронов.....	34
■ Лекция 4: Методы обучения отдельного нейрона.....	49
■ Лекция 5: Вычислительные возможности отдельного нейрона.....	56
■ Лекция 6: Классификация нейронных сетей.....	90
■ Лекция 7: Аналитический метод обучения нейронных сетей.....	108
■ Лекция 8: Метод обратного распространения ошибки.....	112
■ Лекция 9: Аналитический метод обратного распространения ошибки.....	115
■ Лекция 10: Вычислительные возможности нейронных сетей.....	126
■ Лекция 11: Семантические сети.....	139
■ Лекция 12: Сети автоматов.....	141
■ Лекция 13: Полносвязные сети Хопфилда.....	144
■ Лекция 14: Двухнаправленная ассоциативная память.....	150
■ Лекция 15: Сети Хемминга.....	152
■ Лекция 16: Самоорганизующиеся сети Кохонена.....	164
■ Лекция 17: Генетические алгоритмы.....	166

# Лекция 1. Введение в теорию нейронных сетей.

Теория нейронных сетей представляет собой не только математический аппарат, предназначенный для решения задач, но и набор методов, не имеющих строгого математического доказательства. Эти методы получены путем формализации наблюдений нейрофизиологов за живыми нервными системами. Обоснование работоспособности этих методов заключается в следующем принципе.

*Если это работает у естественных нейронных сетей, то можно попытаться применить это для искусственных нейронных сетей.*

Применяя методы нейронных сетей и другие интеллектуальные алгоритмы, следует исходить из следующего принципа естественной целесообразности.

*Если нам известен хорошо работающий математический метод для решения этой задачи, то следует применить именно его, если же мы не знаем эффективного математического метода для решения этой задачи, то можно применить эвристический метод, основанный на нейробиологических знаниях.*

В качестве иллюстрации того, что как следует пользоваться принципом естественной целесообразности рассмотрим несколько примеров конкретных задач.

**Задача 1.** Известны длина  $x$  и ширина  $y$  прямоугольника.  
Найти площадь прямоугольника.

Еще в начальной школе нас учат *нейросетевому алгоритму быстрого решения этой задачи, основанному на таблице умножения...* по крайней мере, до 9 на 9.

С другой стороны, нас учат и чисто математическим методам, например, *умножению столбиком*.

Большинство людей постепенно расширяют область применимости *таблицы умножения*.

Например, следует ли умножать столбиком 3 на 11?

Кроме того, многие люди постепенно расширяют спектр математических методов для решения этой задачи.

Например, метод, основанный на китайской теореме об остатках.

В задаче 1 мы сталкиваемся с двумя нейросетевыми алгоритмами.

Первый из них – таблица умножения, второй – решатель, определяющий выбор метода нахождения площади в каждом конкретном случае.



## **Задача для вычислительно эксперимента 1.**

Ограничиваясь таблицей умножения, умножением столбиком и китайской теоремой об остатках, решить задачу нахождения площади.

**Задача 2.** Известны длина  $x$  и ширина  $y$  прямоугольника.  
Верно ли, что его площадь меньше числа  $z$ ?

Очевидно, что для задачи 2 можно использовать тот же метод, что и для задачи 1, и последующее сравнение.

Но представляется более эффективным использование решателя, который бы отбраковывал очевидно плохие числа  $z$ .

**Вопрос 1.** Какие именно критерии использовать для решателя в задаче 2?

**Задача 3.** Известны длина  $x$  и ширина  $y$  прямоугольника.  
Верно ли, что его площадь равна числу  $z$ ?

В задаче 3 решатель может быть усовершенствован.

**Вопрос 2.** Какие именно критерии использовать для усовершенствования решателя в задаче 3?

# Лекция 2.

## Основы нейробиологии.

# Естественный нейрон

Элементом клеточной структуры мозга является нервная клетка – нейрон.

Нейрон в своем строении имеет много общих черт с другими клетками:

- ❑ тело нейрона окружено плазматической мембраной, внутри которой находится цитоплазма,
- ❑ ядро и другие составляющие клетки.

Однако нервная клетка существенно отличается от других клеток по своему функциональному назначению.

Нейрон по своему функциональному назначению можно рассматривать как устройство для обработки и передачи информации.

Информация переносится в виде импульсов нервной активности, имеющих электрохимическую природу.



Нейроны крайне разнообразны по форме, которая зависит от их местонахождения в нервной системе и особенностей функционирования.

На сегодняшний день нейрофизиологами выделено около 50 различных типов нейронов, которые функционируют разными способами и не являются взаимозаменяемыми.

Мы рассмотрим лишь весьма упрощенную схему функционирования нейронов.

Они выполняют три основные функции:

- ☐ прием;
- ☐ преобразование;
- ☐ передачу.

Таким образом, каждая нервная клетка является полноценным сетевым устройством.

Рассмотрим в общих чертах устройство и принципы работы нервной клетки.

В нейроне можно выделить четыре основные части:

- ☐ дендриты;

- ☐ сома;

- ☐ аксон;

- ☐ синапсы.

Сома – это тело клетки размером от 3 до 100 микрон, снабженное множеством коротких разветвлённых отростков и одним длинным отростком.

Короткие отростки – дендриты, длинный – аксон. На конце аксона располагается еще одно семейство коротких разветвлённых отростков, называемых синапсами.

Длина аксонов обычно заметно превосходит размеры дендритов, в отдельных случаях достигая десятков сантиметров и даже метров.

Гигантский аксон кальмара имеет толщину около миллиметра.

Именно наблюдение за аксоном кальмара послужило выяснению механизма передачи нервных импульсов между нейронами.

Каждая нервная клетка может находиться в двух основных состояниях:

- ☐ обычном;
- ☐ возбуждённом.

В возбуждённом состоянии клетка генерирует электрический импульс величиной около 100 мВ и длительностью 1 мс, который проходит по аксону до синапсов. Синапс при приходе импульса выделяет вещество, способствующее проникновению положительных зарядов внутрь соседней клетки.

Синапсы имеют разную способность концентрировать вещество, способствующее проникновению положительных зарядов внутрь соседней клетки. При этом некоторые даже препятствуют его выделению – они называются тормозящими.

Если суммарный заряд, попавший в клетку, превосходит некоторый порог, клетка возбуждается и генерирует импульс, который распространяется по аксону и доходит до синапсов, что способствует возбуждению следующих клеток.

После возбуждения клетки наступает перерыв – период релаксации. Некоторое время клетка не способна генерировать новые импульсы.

Благодаря этому клетки работают по тактам. Поэтому их можно рассматривать как дискретно функционирующие автоматы.

Сеть таких автоматов передаёт направленную волну импульсов.

# Естественные нейронные сети

Взаимодействующие между собой посредством передачи через отростки возбуждений нейроны формируют нейронные сети.



Скорость распространения нервного импульса составляет приблизительно 100 м/с, что в миллион раз меньше скорости распространения электрического сигнала в медной проволоке.

Тем не менее, сложные задачи распознавания человек решает за десятые доли секунды.

Это означает, что нейровычисления требуют порядка 100 последовательных тактов и выполняются с большой степенью параллелизма.

Кора головного мозга человека содержит порядка 100 000 000 000 нейронов, и каждый нейрон связан с 1000 – 10 000 других нейронов. Это обеспечивает высокую взаимозаменяемость нервных клеток и надежность нервной системы в целом. Отказ даже существенной доли нейронов не нарушает нормального хода распространения нервного импульса.

Установлено, что в головном мозге совокупность нейронов в объеме масштаба 1 куб. мм формирует относительно независимую локальную сеть, несущую определенную функциональную нагрузку.

Обычно выделяют три основные типа нейронных сетей, отличающихся структурой и назначением.

- ❑ Иерархические сети.
- ❑ Локальные сети.
- ❑ Дивергентные сети с одним входом.

Иерархические нейронные сети часто встречаются в сенсорных и двигательных путях. Информация в таких сетях передается в процессе последовательного перехода от одного уровня иерархии к другому.

Нейроны образуют два характерных типа соединений – конвергентные, когда большое число нейронов одного уровня контактирует с меньшим числом нейронов следующего уровня, и дивергентные, в которых контакты устанавливаются со все большим числом клеток последующих слоев иерархии.

Сочетание конвергентных и дивергентных соединений обеспечивает многократное дублирование информационных путей, что является решающим фактором надежности нейронной сети.

При гибели части клеток, сохранившиеся нейроны оказываются в состоянии поддерживать функционирование сети.

Локальные нейронные сети формируются нейронами с ограниченными сферами влияния.

Нейроны локальных сетей производят переработку информации в пределах одного уровня иерархии.

При этом функционально локальная сеть представляет собой относительно изолированную тормозящую или возбуждающую структуру.

Дивергентные сети с одним входом являются частным случаем иерархических. На первом уровне иерархии располагается один нейрон, называемый командным. Командный нейрон может оказывать влияние сразу на множество нейронов. Поэтому сети с одним входом выступают согласующим элементом в сложном сочетании нейросетевых систем всех типов.

# Обучение естественных нейронных сетей

Структура основных типов нейронных сетей генетически предопределена.

Исследования в области сравнительной нейроанатомии говорят о том, что по фундаментальному плану строения мозг очень мало изменился в процессе эволюции.

Генетическая предопределенность имеет место также и в отношении свойств отдельных нейронов.

Изменчивость на клеточном уровне проявляется в пластичности синаптических контактов.

Характер метаболической активности нейрона и свойства проницаемости синаптической мембраны могут меняться в ответ на длительную активизацию или торможение нейрона.

Синаптический контакт «тренируется» в ответ на условия функционирования.

Изменчивость на уровне сети связана со спецификой нейронов.

Нервная ткань практически лишена характерной для других типов тканей способности к регенерации путем деления клеток. Однако нейроны демонстрируют способность к формированию новых отростков и новых синаптических контактов.



Ряд экспериментов с преднамеренным повреждением нервных путей указывает, что развитие нейронных ответвлений сопровождается конкуренцией за обладание синаптическими участками.

Это свойство в целом обеспечивает устойчивость функционирования нейронных сетей при относительной ненадежности их отдельных частей – нейронов.

Специфическая изменчивость нейронных сетей и свойств отдельных нейронов лежит в основе их способности к обучению – адаптации к условиям функционирования – при неизменности в целом их морфологической структуры. Следует заметить, однако, что рассмотрение изменчивости и обучаемости малых групп нейронов не позволяет в целом ответить на вопросы об обучаемости на уровне высших форм психической деятельности, связанных с интеллектом, абстрактным мышлением, речью.



1/15

# Модели нейронов

Модель нейрона МакКаллока — Питса

Функции активации нейрона МакКаллока — Питса

Модель нейрона Паде



## Модель нейрона МакКаллока — Питса

Модель нейрона МакКаллока — Питса была предложена в 1943г. (см. [1], [2]). В рамках этой модели нейрон считается бинарным элементом.

Входной сигнал, поступающий с  $j$ -ого узла (в роли узла может выступать выход другого нейрона или начальный вход) обозначается  $x_j$ . Всего у нейрона предполагается наличие  $n$  входных сигналов, образующих множество

$$\{x_j \mid j \in \{1, 2, \dots, n\}\}.$$

Кроме того, на вход нейрона поступает сигнал с так называемого поляризатора. В отличие от других входных сигналов, являющихся переменными величинами, сигнал, поступающий с поляризатора всегда равен 1. Поляризатор считается узлом с номером 0.



Все входные сигналы умножаются на числа  $w_{ij}$ , где  $j$  — номер входного узла, а  $i$  — номер самого нейрона.

Таким образом, на  $t$ -ом шаге работы сумматор образует сигнал

$$u_i = \sum_{j=0}^n w_{ij} x_j(t),$$

где  $x_0(t)$  всегда равняется 1.

В качестве активационной функции в нейроне МакКаллока — Питса используется пороговая функция  $f$ , заданная следующим образом:

$$f(u) = \begin{cases} 1, & u > 0 \\ 0, & u \leq 0 \end{cases}$$



# Функции активации нейрона МакКаллока — Питса

Модель МакКаллока — Питса может быть обобщена на случай произвольных вещественных входов и выходов, и произвольных функций активации. Рассмотрим наиболее часто используемые функции активации.



4/15



Функция Хэвисайда.

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



5/15



*Сигмоидная функция.*

$$f(x) = \frac{1}{1 + \frac{1}{e^x}}.$$



6/15





*Гиперболический тангенс.*

$$f(x) = \frac{2}{1 + \frac{1}{e^{2x}}} - 1.$$



7/15



*Рациональная сигмоидная функция.*

$$f(x) = \frac{x}{1 + |x|}.$$



8/15



*Логарифмическая функция.*

$$f(x) = \ln(x + \sqrt{x^2 + 1}).$$



9/15



Гауссовская функция.

$$f(x) = e^{-\frac{x^2}{2}}.$$



10/15



*Линейная функция.*

$$f(x) = x.$$



11/15



*Кусочно-линейная функция.*

$$f(x) = \begin{cases} 1, & x \geq \frac{1}{2} \\ |x|, & \frac{1}{2} > x > -\frac{1}{2} \\ 0, & x \leq -\frac{1}{2} \end{cases}$$



12/15



*Биполярная функция.*

$$f(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$



13/15



## Модель нейрона Паде

Нейрон Паде всегда имеет только два входа. У него нет сумматора. Соответственно в качестве нелинейного преобразователя используется функция двух переменных:

$$f(x_1, x_2) = \frac{x_1}{x_2}.$$



14/15







## Список литературы

- [1] McCulloch W. S., Pitts W. H. A logical calculus of ideas immanent in nervous activity // Bull. Math. Biophysics. 1943. Vol. 5. P. 115 – 119.
- [2] Rosenblatt F. Principle of neurodynamics. N.Y.: Spartan, 1992.



# Лекция 4. Методы обучения отдельного нейрона.

# Персептрон Розенблатта

В 1957 году Розенблатт предложил эвристический алгоритм обучения нейрона, основанный на принципах, рассмотренных в нейрофизиологии.

Экспериментально было обнаружено, что при синхронном возбуждении двух связанных нервных клеток синаптическая связь между ними усиливается. Чем чаще синапс угадывает правильный ответ, тем сильнее становится связь.

Своеобразная тренировка связи приводит к постепенному запоминанию информации. Если же синапс начинает часто ошибаться или вообще перестаёт использоваться, связь ослабевает, информация начинается забываться. Таким образом, память реализуется в синапсах. В математической модели нейрона роль памяти играет вектор синаптических весов.

Правило Розенблатта нетрудно формализовать. Как признаки, так и ответы будем полагать бинарными.

Перед началом обучения вектор весов некоторым способом инициализируется, например, заполняется нулевыми или случайными значениями. Затем обучающие объекты по очереди подаются на вход модели МакКаллока–Питтса. Выданные ответы сравниваются с правильными. Если полученный ответ совпадает с предполагаемым, то вектор весов не изменяется. Если полученный ответ равен 0 и предполагаемый ответ равен 1, то вектор весов увеличивается. Если полученный ответ равен 1 и предполагаемый ответ равен 0, то вектор весов уменьшается.

Увеличивать имеет смысл только те веса, которые соответствуют ненулевым компонентам, так как изменение других компонент не повлияет на результат. Соответственно ненулевые веса должны изменяться пропорционально их значению, а значит, и вкладу в получаемый ответ.

# Правило Хэбба

Часто удобнее полагать, что классы помечены числами  $-1$  и  $1$ , а нейрон выдаёт знак скалярного произведения входного вектора и вектора весов.

В этом случае несовпадение знаков скалярного произведения и предполагаемого значения означает, что нейрон ошибается на соответствующем входном векторе. При этом применяется следующее правило модификации весов:

$$\langle x, w \rangle y < 0 \Rightarrow w := w + \eta xy,$$

где  $\langle w, x \rangle$  — скалярное произведение входного вектора и вектора весов,  $y$  — предполагаемое значение,  $\eta$  — шаг обучения. Это правило называется правилом Хэбба.



# 1 Вычислительные возможности нейронов

Рассмотрим вычислительные возможности одного нейрона в самом общем случае. Пусть  $\mathfrak{F}(m, k)$  - множество кусочно-постоянных функций, которые имеют не более  $m$  разрывов, а мощность области значений не более  $k$ , причем  $m, k \in \mathbb{N} \cup \aleph_0$ .  $\omega$ -пороговым нейроном назовем нейрон с функцией активации  $f \in \mathfrak{F}(m, k)$ . Покажем, что вычислительные возможности таких нейронов весьма высоки – в частности, они реализуют любую функцию, заданную на не более, чем счетном множестве. Позже мы покажем, как свести  $\omega$ -пороговый нейрон к нейросети, состоящей из персептронов.

**Теорема 1.1.** Пусть  $\mathbb{L} \subset \mathbb{R}^n$ ,  $\mathbb{L}$  – не более, чем счетно. Пусть на  $\mathbb{L}$  задана функция  $g$ , а  $\mathbb{A} = g(\mathbb{L})$ . Тогда существует  $\omega$ -пороговый нейрон с функцией активации из  $\mathfrak{F}(|\mathbb{L}|, |\mathbb{A}|)$ , реализующий эту функцию.

*Доказательство.* Рассмотрим вектор  $A \in \mathbb{R}^n$ , такой что для любых  $X, Y \in \mathbb{L}$  выполняется  $A \cdot X \neq A \cdot Y$ . Такой вектор  $A$  наводит линейный порядок на множестве  $\mathbb{L}$  – каждому вектору  $X \in \mathbb{L}$  будет поставлено в соответствие уникальное число  $A \cdot X$ , которое определит линейный порядок, и мы будем называть его линейно-порожденным. Векторы, наводящие линейно-порожденный порядок, будем называть разрезающими.

Покажем, что в условиях теоремы разрезающий вектор существует. Определим  $\mathcal{D}_{r,q} = \{A : A \cdot X^r = A \cdot X^q\}$ .  $\mathcal{D}_{r,q}$  для любых  $r, q$  является гиперплоскостью в  $\mathbb{R}^n$ , а значит, имеет меру равную нулю в  $\mathbb{R}^n$ .

Пусть

$$\mathcal{D} = \bigcup_{r \neq q} \mathcal{D}_{r,q}$$

$\mathcal{D}$  является объединением счетного числа множеств меры нуль, а значит, само имеет меру нуль. Отсюда следует, что разность  $\mathbb{R}^n \setminus \mathcal{D}$  не пуста.

Рассмотрим  $A \in \mathbb{R}^n \setminus \mathcal{D}$ . Поскольку

$$\mathbb{R}^n \setminus \mathcal{D} = \bigcap_{r \neq q} (\mathbb{R}^n \setminus \mathcal{D}_{r,q})$$

то для любых  $X^q, X^r$ ,  $A \cdot X^q \neq A \cdot X^r$ .

Теперь определим функцию  $f(x)$  как кусочно-постоянную функцию, удовлетворяющую уравнениям:

$$f(A \cdot X) = g(X)$$

Очевидно, существует функция из  $f \in \mathfrak{F}(|\mathbb{L}|, |\mathbb{A}|)$ , удовлетворяющая этому уравнению, что и доказывает утверждение теоремы.  $\square$

Следует отметить, что условие теоремы является существенным. Так, функция на шаре в пространстве  $\mathbb{R}^2$  в общем случае не может быть реализована нейроном с действительными весами, однако может быть распознана нейроном с весами из  $\mathbb{C}$  и функцией активации, заданной на множестве

комплексных чисел. Развивая это направление, можно сделать и более общее утверждение: функция на  $\mathbb{R}^n$  может быть распознана нейроном с весами из алгебры Клиффорда порядка  $n$ .

**Теорема 1.2.** Пусть  $\mathbb{L}$  – конечное подмножество  $\mathbb{R}^n$ , и  $g : \mathbb{L} \rightarrow \mathbb{A}$  реализуется нейроном  $N$  с функцией активации  $f$  из  $\mathfrak{F}(m, k)$ ,  $m, k \in \mathbb{N}$ . Тогда существует сеть из  $m + k$  персептронов и одного нейрона-сумматора, которая реализует функцию  $g$ .

*Доказательство.* Построим трехслойную нейросеть. Нейроны первого слоя обозначим как  $N_i^1$  ( $i = 1 \dots m$ ), второго –  $N_i^2$  ( $i = 1 \dots k$ ), а в третьем слое будет единственный нейрон-сумматор.

Пусть  $f$  имеет следующий вид:

$$f(x) = \begin{cases} b_1 & c_0 \leq x < c_1 \\ b_2 & c_1 \leq x < c_2 \\ \dots & \\ b_m & c_{m-1} \leq x < c_m \end{cases}$$

Определим распознающие наборы нейронов первого слоя как  $(c_i; A)$ , где  $A$  – вектор весов нейрона  $N$ . В этом случае при подаче любого вектора из  $\mathbb{L}$  возбуждаются первые  $t$  нейронов первого слоя – важно, что они возбуждаются подряд.

Добавим веса между первым и вторым слоем по правилу: связь между  $i$ -ым нейроном первого слоя и  $j$ -ым нейроном второго слоя существует, если:

- 1) При  $c_{i-1} \leq x < c_i$   $f(x) = b_j$
- 2) При  $c_{i-2} \leq y < c_{i-1} \leq x < c_i$   $f(x) \neq b_j$ , а  $f(y) = b_j$

Такие связи являются «переключателями» – связи первого правила переносят возбуждение, если при данном  $x$  функция нейрона  $N$  выходит на значение  $b_j$ , а связи второго – если она уходит с  $b_j$ .

Поскольку нейроны первого слоя возбуждаются подряд, то нейроны второго слоя должны лишь посчитать четность входных сигналов – в данном случае это можно сделать с помощью вектора  $(-\frac{1}{2}, +1, -1, +1, -1, \dots)$ . При возбуждении нейронов первого слоя подряд такой набор обеспечит срабатывание в случае четности сигналов.

Далее, на последнем слое стоит нейрон-сумматор, соединенный весами со всеми нейронами второго слоя, причем вес  $j$  связи равен  $b_j$ .  $\square$

**Следствие 1.1.** Любая функция, заданная на конечном подмножестве  $\mathbb{R}^n$ , может быть реализована сетью из персептронов и сумматоров

**Следствие 1.2.** Любая функция распознавания конечного подмножества  $\mathbb{R}^n$  (т.е. функция, принимающая значения  $-1$  и  $1$ ) может быть реализована персептронной сетью.

Итак, мы показали, что вычислительные возможности одного нейрона весьма высоки. В частности, любая функция на счетном множестве может быть представлена в виде одного нейрона. С другой стороны, в общем

случае сложность функции активации для такого представления весьма велика. Далее мы будем рассматривать нейроны с простыми функциями активации, и прежде всего персептроны.

Однако, количество разрывов не всегда делает функцию активации вычислительно трудной. Например, функция  $\text{sign}(\sin e^x)$  имеет бесконечное число разрывов на своей области определения, не является периодической, и тем не менее ее вычисление осуществляется быстро. В связи с этим, интерес могут представлять не только функции, реализуемые одним персептроном, но и функции, реализуемые одним нейроном с другими быстро вычисляемыми функциями активации.

## 2 Линейно-порожденные порядки

### 2.1 Определение линейно-порожденного порядка.

Пусть  $\mathbb{L} \subset \mathbb{R}^n$  - конечное. Линейно-порожденным порядком (ЛПП), порожденным вектором  $A$ , будем называть линейный порядок на векторах из  $\mathbb{L}$ , такой что  $X < Y \Leftrightarrow A \cdot X < A \cdot Y$ . Вектор, наводящий на  $\mathbb{L}$  линейный порядок, будем называть разрезающим. Также разрезающим будем называть множество, все векторы которых наводят одинаковый линейный порядок.

Более формальное определение строится следующим образом. Пусть векторы  $\mathbb{L}$  пронумерованы, и  $\mathbb{L} = \{X_1, \dots, X_m\}$ . Тогда перестановка  $\pi \in S(m)$  является ЛПП, если существует вектор  $A$  такой что  $A \cdot X_{\pi(i)} < A \cdot X_{\pi(i+1)}$ . Из теоремы 1.1 непосредственно вытекает, что на любом не более чем счетном множестве может быть наведен линейно-порожденный порядок.

Также сформулируем теорему о связи между линейно-порожденными порядками и функциями, которые могут быть реализованы одним нейроном.

**Теорема 2.1.** Пусть  $\mathbb{L} \subset \mathbb{R}^n$  - конечно,  $g : L \rightarrow A \subset \mathbb{R}$  - произвольная функция. Пусть существует ЛПП  $X_1, \dots, X_m$ , порожденный вектором  $A$ , такой что существуют числа  $i_1, \dots, i_k$ , и для любых  $i_p \leq i \leq j < i_{p+1}$  справедливо  $g(X_i) = g(X_j)$ . Тогда существует кусочно-постоянная функция активации  $f$ , имеющая  $k+1$  разрыв, такая, что  $f(A \cdot X) \equiv g(X)$ .

*Доказательство.* Поскольку для любых  $i_p \leq i \leq j < i_{p+1}$  справедливо  $g(X_i) = g(X_j)$ , можно определить  $f(x)$  уравнениями следующего вида:  $f(x) = g(X_{i_p})$  при всех  $A \cdot x_{i_p} < x < A \cdot x_{i_{p+1}}$ .  $\square$

Данная теорема обладает наглядной геометрической интерпретацией. Семейство гиперплоскостей, перпендикулярных вектору  $A$  из условий теоремы, и проходящих через точки  $X_{i_p}$  делит  $\mathbb{L}$  на слои, причем в каждом слое функция  $g(X)$  принимает только одно значение. Соответственно, функция  $f$  определена уже на всем множестве  $\mathbb{R}^n$ , точно так же делит его на слои, и в каждом слое является константой.

**Следствие 2.1.** Пусть  $f : \mathbb{L} \rightarrow \{-1, 1\}$ . Тогда  $f$  может быть реализована перцептроном тогда и только тогда, когда существует ЛПП, для которого определено  $m$  такое, что для всех  $i \geq m$   $f(X_i) = 1$ ,  $i < m$   $f(X_i) = -1$ .

Геометрическая интерпретация этого факта состоит в том, что одна гиперплоскость делит  $\mathbb{L}$  на две части, и в первой  $g$  тождественно равна  $-1$ , а во второй  $+1$ .

## 2.2 Структура разрезающих множеств

Изучим структуру разрезающих множеств, т.е. множеств, все векторы которых наводят одинаковый линейный порядок на конечном носителе  $\mathbb{L} \subset \mathbb{R}^n$ .

В данном параграфе мы будем пользоваться некоторыми фактами из конического анализа. Коническим множеством называется множество векторов  $\mathbb{L}$  из  $\mathbb{R}^n$ , такое что для любых векторов  $A, B \in \mathbb{L}$  и любых неотрицательных чисел  $\lambda, \mu$  вектор  $\lambda A + \mu B$  также лежит в  $\mathbb{L}$ . Выражение  $\lambda A + \mu B$  называется конической комбинацией векторов  $A$  и  $B$ . Коническое множество является неограниченным, то есть для любого числа  $M$  найдется вектор  $A$  такой, что  $\|A\| > M$ .

Коническое множество однозначно задается своей вершиной  $V_0$  и множеством образующих  $R_1, \dots, R_k$ . Любой элемент конического множества может быть представлен в виде  $X = V_0 + \lambda_1 R_1 + \dots + \lambda_m R_m$ , где  $\lambda_1, \dots, \lambda_m$  – неотрицательные числа.

Точка множества  $M$  называется внутренней если существует  $\varepsilon > 0$  такой, что для любого  $Y$  из того, что  $\|X - Y\| < \varepsilon$  следует, что  $Y$  принадлежит  $M$ . Иными словами,  $X$  входит в  $M$  вместе со своей окрестностью радиуса  $\varepsilon$ . Множество  $M$  называется открытым, если все его точки являются внутренними. Замыканием множества  $M$  называется наименьшее множество, содержащее  $M$  и все его предельные точки.

**Предложение 2.1.** Пусть  $\mathcal{T}$  – множество всех векторов, наводящих одинаковый ЛПП. Тогда  $\mathcal{T}$  является открытым коническим множеством. Замыкание  $\mathcal{T}$  является коническим множеством с конечным числом образующих и вершиной в точке  $0, \dots, 0$ .

*Доказательство.* Пусть нам дан какой-то ЛПП, т.е. векторы из  $\mathbb{L}$  занумерованы как  $X_1, \dots, X_m$ . Запишем систему неравенств, ассоциированную с ним:

$$A \cdot X_i \leq A \cdot X_{i+1}, i = 0, \dots, m-1$$

Или, что эквивалентно:

$$A \cdot (X_i - X_{i+1}) \leq 0, i = 0, \dots, m-1$$

Полученная система неравенств является системой однородных линейных неравенств. Известно, что решение такой системы является коническим множеством. Решение соответствующей системы строгих неравенств (т.е.

системы, полученной заменой знаков  $\leq$  на  $<$ ) является множество внутренних точек этого конического множества.

Нулевое решение (т.е. вектор  $(0, \dots, 0)$ ) обращает все неравенства в равенства, поскольку система является однородной. Из выпуклого анализа известно, что такое решение задает крайнюю точку множества решений, т.е. его вершину. Конечность числа образующих также известна из выпуклого анализа. □

**Предложение 2.2.** *Конусы  $T_1$  и  $T_2$  различных ЛПП не пересекаются.*

*Доказательство.* Пусть имеется пересечение. Тогда точка этого пересечения обязана удовлетворять двум системам строгих линейных неравенств, ассоциированных с различными ЛПП. Понятно, что для любых двух различных ЛПП системы неравенств являются противоречивыми, следовательно, такой точки быть не может. □

**Предложение 2.3.** *Если некоторая точка  $B$  не принадлежит ни одному конусу ЛПП, то  $B$  является граничной точкой для некоторого конуса ЛПП.*

*Доказательство.* Пусть  $A$  – некоторый разрезающий вектор. Поскольку конусов ЛПП для конечного  $\mathbb{L}$  конечное число (поскольку конечно число перестановок  $\mathbb{L}$ ), то существует лишь конечное число  $\varepsilon_1, \dots, \varepsilon_n$ , для которых  $(\varepsilon A + B)$  не является вектором ЛПП. Следовательно, можно выбрать минимальный из них, пусть это первый, и тогда при любом  $0 < \varepsilon < \varepsilon_1$   $(\varepsilon A + B)$  является вектором ЛПП и, кроме того, лежит в одном и том же конусе  $T$ . Тогда очевидно, что  $B$  является граничной точкой для конуса  $T$ . □

Приведенные предложения позволяют сформулировать следующую теорему.

**Теорема 2.2.** *Для любого конечного носителя  $\mathbb{L} \subset \mathbb{R}^n$ , множество векторов  $\mathbb{R}^n$  разбивается на множество непересекающихся конусов ЛПП на  $\mathbb{L}$  и их грани.*

На основании доказательства Предложения 2.2 может быть разработан простейший алгоритм проверки того, является ли  $X_1, \dots, X_m$  линейно-порожденным порядком. Для этой проверки необходимо записать систему линейных неравенств, соответствующую последовательности  $X_i$ , и решить ее. Если у нее существует нетривиальное решение (т.е. решение со внутренней точкой, обращающее все неравенства в строгие неравенства), то линейно-порожденный порядок существует.

Алгоритмы решения систем линейных неравенств хорошо известны – это алгоритм Моцкина-Бургера и созданная на его основе схема Черникова. Эти алгоритмы позволяют найти все образующие множества решений. К сожалению, в худшем случае эти алгоритмы работают экспоненциальное время.

## 2.3 Количество линейно-порожденных порядков

Естественным является вопрос о количестве линейно-порожденных порядков на данном множестве  $\mathbb{L}$ . Ответ на этот вопрос дается одним из фундаментальных результатов в этой области – теоремы Ковера.

Введем некоторые предварительные определения. Будем говорить, что множество точек из  $\mathbb{R}^n$  находится в общем положении, если в любой гиперплоскости размерности  $k$ ,  $k < n$ , находится не более  $k + 1$  точки. Например, нет трех точек на одной прямой, четырех точек в одной плоскости и так далее.

**Теорема 2.3.** (Cover, 1967) Пусть  $\mathbb{L} \subset \mathbb{R}^n$ ,  $|\mathbb{L}| = m$  и  $\mathbb{L}$  находится в общем положении. Пусть множество различных линейно-порожденных порядков определяется функцией  $Q(m, n)$ . Тогда  $Q(m, n)$  удовлетворяет следующим рекуррентными соотношениям:

$$Q(m + 1, n) = Q(m, n) + mQ(m, n - 1)$$

$$Q(m, 1) = 2$$

$$Q(2, n) = 2$$

Удивительным следствием теоремы Ковера является тот факт, а именно, что количество ЛПП не зависит от конфигурации  $\mathbb{L}$ , если  $\mathbb{L}$  находится в общем положении. Между тем, структура ЛПП может быть принципиально различной.

Приведем пример различных структур ЛПП. Рассмотрим следующие множества точек.  $\mathbb{L}_1 = \{(0, 0), (0, 1), (1, 0), (1.5, 1.5)\}$ ,  $\mathbb{L}_2 = \{(0, 0), (2, 0), (0, 2), (0.5, 0.5)\}$ . Множество  $\mathbb{L}_1$  представляет собой прямоугольный треугольник и одну точку вне его, а множество  $\mathbb{L}_2$  – прямоугольный треугольник и одну точку внутри него. Схематично эти множества изображены на следующем рисунке.



Путем перебора можно построить множества всех ЛПП для обоих множеств. Для  $\mathbb{L}_1$  это 4231, 4321, 4312, 3412, 3142, 3124, 1324, 1234, 2134, 2143, 2413, 4213. Для  $\mathbb{L}_2$  это 2431, 2341, 3241, 3421, 3412, 3142, 1342, 1432, 1423, 1243, 2143, 2413. Можно показать, что ни каком переименовании элементов множества  $\mathbb{L}_1$ , линейно-порожденные порядки  $\mathbb{L}_1$  не совпадут с порядками на  $\mathbb{L}_2$ . Таким образом мы видим, что количество ЛПП одинаково на обоих множествах, но структура этих ЛПП различна.

Теорема Ковера также позволяет применять недетерминированные алгоритмы для поиска всех ЛПП на данном множестве. Очевидным способом такого поиска является составление систем линейных неравенств для

всевозможных ЛПП и проверка их на совместность. Понятно, что требуется проверить  $n!$  систем неравенств, причем проверка одной системы неравенств экспоненциальна в худшем случае. Поэтому такой алгоритм является неэффективным.

С другой стороны, по теореме 2.2, лишь граничные точки конусов не наводят ЛПП. Поэтому из всех точек  $\mathbb{R}^n$ , лишь множество точек меры нуль не является разрезающими векторами. Поэтому выбранный наугад вектор с вероятностью 1 наводит ЛПП. Соответственно, мы можем бросать векторы наугад в  $\mathbb{R}^n$ , вычислять ЛПП, им соответствующий, и закончить процедуру тогда, когда мы соберем нужное число ЛПП. Это число вычисляется по теореме Ковера.

## 2.4 Линейно-порожденные порядки на симплексах

Исследование линейно-порожденных порядков имеет важное практическое применение. Пусть нам известен способ быстрой проверки того, является ли последовательность  $X_1, \dots, X_m$  линейно-порожденным порядком. В данном контексте «быстрый» означает не более полиномиальный по переменной  $m$  и размерности векторов  $n$ . Это означает, что экспоненциальный в худшем случае алгоритм Моцкина-Бургера не подходит. Имея такой алгоритм и используя теорему 2.1, можно быстро представить любую заданную функцию  $g$  одним нейроном с заданным числом разрывов функции активации либо доказать, что такого представления не существует. Также интерес представляет полиномиальный алгоритм поиска образующих разрезающего множества, соответствующего линейно-порожденному порядку.

В общем случае, для произвольных множеств проверка линейной порожденности последовательности эквивалентна проверке совместности системы неравенств и не может быть решена полиномиально. Соответственно, поиск образующих разрезающего множества эквивалентен решению системы линейных неравенств. Однако, интерес может представлять та же проверка на специальных сериях множеств. Например, множество  $\{-1, 1\}^n$  является носителем  $n$ -арных булевых функций. Исследование линейно-порожденных порядков на этом множестве может позволить решать проблемы представления персептронами булевых функций. Эта проблема имеет самостоятельную ценность.

В данном параграфе мы покажем, каким образом поиск образующих разрезающего множества может быть реализован без использования алгоритма Моцкина-Бургера на примере простых множеств. Мы будем рассматривать симплексные множества. Симплексом в  $\mathbb{R}^n$  называется  $n + 1$ -элементное множество в общем положении. Изучим некоторые свойства симплексов, связанные с линейно-порожденными порядками.

**Теорема 2.4.** *Любой порядок на симплексе является линейно-порожденным.*

*Доказательство.* Докажем эту теорему по индукции. База индукции (при  $n = 2$ ) очевидна.

Рассмотрим теперь некоторый линейно-порожденный порядок на симплексе  $\mathbb{S}$ . Пусть он начинается с некоторой точки  $X$ , и записывается как  $X, X_1, \dots, X_n$ . Пусть  $\mathbb{S}' = \mathbb{S} \setminus \{X\}$ . Очевидно,  $\mathbb{S}'$  тоже симплекс, но его размерность на единицу меньше. Перейдем в систему координат такую, что  $X = (0, 0, \dots, 0, 1)$ , а любой вектор из  $\mathbb{S}'$  имеет последнюю координату, равную нулю: очевидно, в такую систему координат перейти возможно.

Поскольку  $\mathbb{S}'$  является симплексом, то  $X_1, \dots, X_n$  является линейно-порожденным порядком по предположению индукции. Пусть  $A'$  – его вектор. Пусть  $u = \min\{A' \cdot X_i\} - 1$ . Рассмотрим вектор  $A = (A'; u)$ . Очевидно, что  $A \cdot X \leq A \cdot X_i$ , по построению, и, кроме того, понятно, что  $A$  является вектором ЛПП для  $X_1, \dots, X_n$ . Тогда  $A$  – вектор ЛПП для  $X, X_1, \dots, X_n$ , что доказывает утверждение теоремы.  $\square$

Зафиксируем на симплексе некоторый порядок  $X_0, \dots, X_n$ . Аффинным преобразованием всегда можно привести его к базису, в котором  $X_0 = 0$ , а  $X_i = E_i$ ,  $E_i$  – единичный вектор. Такой базис будем называть естественным. Далее будем рассматривать только линейно-порожденные порядки, начинающиеся с элемента  $X_0$ , и обозначать эти порядки как ЛПП-0. Легко видно, что векторы таких порядков имеют неотрицательные компоненты.

**Лемма 2.1.** *Множество  $\{0, 1\}^n \setminus 0$  в точности является множеством всех ребер всех конусов ЛПП.*

*Доказательство.* Рассмотрим ребро для ЛПП. Поскольку ребро  $A$  соответствует  $n - 1$ -грани, то в системе неравенств для ЛПП симплекса  $n - 1$  неравенство должно превратиться в равенство. Тогда  $\mathbb{S}$  разбивается на  $\mathbb{S}_1, \mathbb{S}_2$ , такие что

- 1)  $0 \in \mathbb{S}_1$
- 2)  $\forall X, Y \in \mathbb{S}_i \ A \cdot X = A \cdot Y$ .

Без ограничения общности можно считать, что в  $\mathbb{S}_1$  помимо нуля попали ребра  $E_1, \dots, E_{k-1}$ , а в  $\mathbb{S}_2$  – ребра  $E_k, \dots, E_n$ . Тогда, для  $\mathbb{S}_1$   $A \cdot E_i = A \cdot 0$ , но  $A \cdot E_i = a_i$ , а  $A \cdot 0 = 0$ , следовательно  $a_i = 0$  для всех  $1 < i < k$ .

Для  $\mathbb{S}_2$  аналогично устанавливаем, что  $a_i = a_j$  при  $i, j \geq k$ , и  $a_i \neq 0$  для  $i \geq k$ . Это доказывает, что любое ребро имеет требуемый вид. Доказательство обратного утверждения полностью аналогично.  $\square$

**Теорема 2.5.** *Пусть  $g$  – перестановка множества  $\{1, \dots, n\}$ . Пусть  $Y_i = X_{g(i)}$ ,  $Y_0 = X_0$ . Пусть  $D$  – верхнетреугольная матрица, ненулевые компоненты которой равны единице.  $D'$  – это матрица  $D$ , к столбцам которой применена перестановка  $g$ .  $\mathcal{V}$  – множество векторов, координаты которых записаны в строках матрицы  $D'$ . Тогда  $\mathcal{V}$  является множеством образующих конуса для ЛПП  $Y_0, \dots, Y_n$ . Соответственно, множества  $\mathcal{V}$ , определенные таким образом, исчерпывают все ЛПП.*

*Доказательство.* Докажем, что коническая оболочка  $\mathcal{T} = \text{Cone}(\mathcal{V})$  является конусом какого-либо ЛПП. В самом деле, образующими  $\mathcal{T}$  являются векторы, которые являются ребрами для линейно-порожденных порядков по лемме 2.1. Кроме того, ни один вектор из  $\mathcal{V}$  не выражается через другие.



Далее,  $\mathcal{T}$  очевидно имеет внутреннюю точку в  $\mathbb{R}^n$ . Из всего этого следует, что  $\mathcal{T}$  является конусом ЛПП.

Докажем теперь, что  $\mathcal{T}$  соответствует нужному ЛПП. Докажем по индукции. Рассмотрим матрицу  $D$  и покажем, что она соответствует порядку  $X_0, \dots, X_n$ . В самом деле, выберем внутреннюю точку конуса, соответствующего  $D$ , а именно  $A = \sum_{i=1}^n D_i$ ,  $D_i$  – строки матрицы. Очевидно, что  $A \cdot E_i = i$ , что доказывает базу индукции. Шаг индукции сводится к следующему: матрица, соответствие которой нужному ЛПП установлено, подвергается транспозиции двух столбцов и доказываемся, что эта транспозиция перемещает и соответствующие точки в ЛПП.

Действительно, пусть при транспозиции двух столбцов матрицы  $i$  и  $j$  получается матрица  $D'$ . Тогда что  $D' \cdot Y_i = D \cdot Y_j$  и  $D' \cdot Y_j = D \cdot Y_i$ . Но в этом случае очевидно, что элементы  $Y_i$  и  $Y_j$  меняются местами в ЛПП. Поскольку любая перестановка сводится к композиции транспозиций, утверждение теоремы доказано.  $\square$

### 3 Линейно-разделимые булевы функции

Булевой функцией называется функция из множества  $\{0, 1\}^n$  в  $\{0, 1\}$ . Булева функция является линейно-разделимой, если она может быть реализована одним персептроном.

Прежде всего необходимо выбрать кодирование, которое бы сопоставляло множеству  $\{0, 1\}^n$  в смысле элементов булевой алгебры подмножество  $\mathbb{L} \subset \mathbb{R}^n$ . Имеет смысл рассматривать кодирование вида

$$x_i \rightarrow \begin{cases} p(i) & , \quad x = 0 \\ t(i) & , \quad x = 1 \end{cases} ,$$

где  $p(i)$ ,  $t(i)$  – некоторые функции от натурального числа,  $p(i) \neq t(i)$ . Все такие кодирования являются эквивалентными с точки зрения реализации персептроном. В самом деле, путем аффинного преобразования пространства  $\mathbb{R}^n$  можно перевести одну схему кодирования в другую. Аффинное преобразование сохраняет гиперплоскости, и если какое-то множество точек было отделимо гиперплоскостью до преобразования, то и после преобразования оно отделимо. Мы выберем схему кодирования с  $p(i) = -1$ ,  $t(i) = 1$  по причинам удобства для доказательств. В зависимости от контекста (говорим ли мы о булевой алгебре или о ее векторном представлении), значению «ложь» будет сопоставляться и символ 0, и символ  $-1$ .

Рассмотрим некоторые известные факты о булевых функциях. Известно, что не все булевы функции являются таковыми. В частности, классическим результатом, сыгравшим значительную роль в развитии теории нейронных сетей, является теорема Минского.

**Теорема 3.1.** (*Minsky, 1969*). *Функция XOR не может быть реализована одним персептроном.*

*Доказательство.* Запишем систему линейных неравенств, определяющую веса персептрона для реализации функции XOR.

$$\begin{cases} A_0 - A_1 - A_2 > 0 \\ A_0 + A_1 + A_2 > 0 \\ A_0 - A_1 + A_2 < 0 \\ A_0 + A_1 - A_2 < 0 \end{cases}$$

Складывая первое неравенство со вторым, а третье – с четвертым, получаем такую систему линейных неравенств:

$$\begin{cases} 2A_0 > 0 \\ 2A_0 < 0 \end{cases}$$

Очевидно, система является несовместной, а значит, ни при каких весах персептрон не способен реализовать функцию XOR.  $\square$

Пример функции XOR не является редким или экзотическим. Количество линейно-разделимых функций было оценено Ковером в 1965 году, и составляет  $2^{\Theta(n^2)}$ , где  $n$  – аргумент функции. Количество всех булевых функций от  $n$  переменных составляет  $2^{2^n}$ , и, следовательно, линейно-разделимые булевы функции составляют исчезающе малую часть всех булевых функций.

Кроме того, Hegedus и Megiddo в 1996 году получили следующий результат. Если функция  $f$  дана в виде дизъюнктивной нормальной формы, то задача определения того, является ли  $f$  линейно-разделимой, co-NP-полна.

### 3.1 Множества, ассоциированные с булевыми функциями

Пусть  $f$  – некоторая булева функция. Мы будем говорить, что функция  $f$  почти реализуется персептроном с весами  $A$ , если  $f(X) = \text{sign}(a_0 + A \cdot X)$  на тех  $X$ , для которых  $a_0 + A \cdot X \neq 0$ . Прочие  $X$  не накладывают ограничений. В случае персептрона мы также можем говорить о весах, которые реализуют (соответственно, почти реализуют)  $f$ , поскольку функция активации зафиксирована.

Через  $\mathcal{G}(f)$  обозначим все векторы  $(a_0; A)$ , которые почти реализуют  $f$ . Через  $\mathcal{H}(f)$  – все векторы  $A$  такие, что существует  $a_0$  при котором  $(a_0; A) \in \mathcal{G}(f)$ . И, наконец, как  $\mathcal{I}(f, A)$  обозначим множество весов активации, которые допустимы для данного  $A$ , т.е.  $\mathcal{I}(f, A) = \{a_0 : (a_0; A) \in \mathcal{G}(f)\}$

**Лемма 3.1.** Для любой функции  $f$  множества  $\mathcal{G}(f)$  и  $\mathcal{H}(f)$  являются коническими.

*Доказательство.* Пусть  $X$  из  $\mathbb{B}^n$ . Очевидно,  $\mathcal{G}(f)$  является множеством решений следующей системы неравенств относительно  $A$ :

$$\begin{cases} a_0 + A \cdot X \leq 0 & , & X \in f^{-1}(-1) \\ a_0 + A \cdot X \geq 0 & , & X \in f^{-1}(1) \end{cases} ,$$

где через  $f^{-1}$  обозначена операция взятия полного прообраза. Неравенства однородны, следовательно, множество решений системы является конусом [?].

Множество  $\mathcal{H}(f)$  – это проекция  $\mathcal{G}(f)$ , следовательно, оно также является коническим.  $\square$

**Лемма 3.2.** Для любой функции  $f$  и вектора  $A \in \mathcal{H}(f)$ :

- 1)  $\mathcal{I}(f, A)$  является интервалом вида  $[a, b]$ .
- 2)  $b = \infty$  если и только если  $f$  тождественно равна 1.
- 3)  $a = -\infty$  если и только если  $f$  тождественно равна  $-1$ .
- 4) Если  $b \neq \infty$ , то существует  $X_b$ , такой, что  $b + A \cdot X_b = 0$ , и  $f(X_b) = -1$ .
- 5) Если  $a \neq -\infty$ , то существует  $X_a$ , такой, что  $a + A \cdot X_a = 0$  и  $f(X_a) = 1$ .

*Доказательство.* Рассмотрим следующие системы неравенств относительно  $a_0$ :

$$\{a_0 + A \cdot X \leq 0, \quad X \in f^{-1}(-1)\} \quad (1)$$

$$\{a_0 + A \cdot X \geq 0, \quad X \in f^{-1}(1)\} \quad (2)$$

Пусть  $f(X)$  не является тождественной. Тогда обе системы имеют хотя бы одно неравенство, и множество решений системы (1):

$$\left(-\infty, \min_{X \in f^{-1}(-1)} -A \cdot X\right]$$

Аналогично, множество решений системы (2):

$$\left[\max_{X \in f^{-1}(1)} A \cdot X, \infty\right)$$

Ясно, что  $M_1$  имеет на правой границе значение, удовлетворяющее выводу 4 леммы. Аналогично,  $M_2$  имеет на левой границе значение, удовлетворяющее выводу 5 леммы. Общее решение обеих систем является пересечением  $M_1$  и  $M_2$ , и, следовательно, удовлетворяет выводам 1,4,5.

Если  $f$  – тождественная единица, то  $M_1 = (-\infty, \infty)$  и решение совпадает с  $M_2$ , откуда следует вывод 2. Во всех других случаях первая система неравенств пуста. Аналогично доказывается вывод 3.  $\square$

На интервалах естественным образом можно рассматривать операции сложения  $([a, b] + [c, d] = [a + c, b + d])$  и умножения на положительный скаляр  $(\lambda[a, b] = [\lambda a, \lambda b])$ .

**Лемма 3.3.** Пусть  $A, B \in \mathcal{H}(f)$ ,  $\lambda, \mu$  – положительные числа,  $C = \lambda A + \mu B$ . Через  $\Omega$  обозначим  $\lambda \mathcal{I}(f, A) + \mu \mathcal{I}(f, B)$ . Тогда  $\mathcal{I}(f, C) = \Omega$ .

*Доказательство.* Докажем, что  $\mathcal{I}(f, C) \supset \Omega$ . В самом деле, пусть  $c \in \Omega$ , тогда можно представить  $c$  в виде  $c = \lambda a + \mu b$ ,  $a \in I(f, A)$ ,  $b \in I(f, B)$ . Тогда:

$$c + C \cdot X = \lambda a + \mu b + (\lambda A + \mu B) \cdot X = \lambda(a + A \cdot X) + \mu(b + B \cdot X).$$

Таким образом,  $(c; C) \in \mathcal{G}(f)$ . Следовательно,  $c \in \mathcal{I}(f, C)$ , следовательно  $\mathcal{I}(f, C) \supset \Omega$ .

Докажем теперь, что  $\mathcal{I}(f, C) \subset \Omega$ . От противного – пусть  $\Omega = [a, b]$ , и пусть  $c < a$ . Покажем, что  $(c; C) \notin \mathcal{I}(f, C)$ . В самом деле, по утверждению леммы 3.2, либо  $a = -\infty$  (и тогда указанных  $c$  не существует), либо существует  $X_a$  такой, что  $a + C \cdot X_a = 0$  и  $f(X_a) = 1$ . Отсюда следует, что  $c + C \cdot X_a < 0$ . Из этого вытекает, что  $\text{sign}(c + C \cdot X_a) = -1$ , а значит,  $(c; C) \notin \mathcal{I}(f, C)$ . Полученное противоречие доказывает положение леммы, для второго края  $\Omega$  доказательство полностью аналогично.  $\square$

**Лемма 3.4.** *Если  $A \in \mathcal{H}(f)$  является внутренней точкой, то  $\mathcal{I}(f, A)$  имеет внутренние точки. Если  $A$  – внутренняя точка  $\mathcal{H}(f)$ ,  $a_0$  – внутренняя точка  $\mathcal{I}(f, A)$ , то  $(a_0; A)$  – внутренняя точка  $\mathcal{G}(f)$ . Если  $\mathcal{G}(f)$  содержит внутренние точки, то  $f$  строго реализуется персептроном.*

*Доказательство.* Пойдем от противного. Единственный интервал, который не содержит внутренних точек – это интервал  $[a, a]$ . Тогда по утверждению леммы 3.2 существует  $X_1$ , такой, что  $f(X_1) = 1$  и  $X_2$  такой, что  $f(X_2) = -1$ . Тогда  $A \cdot X_1 = A \cdot X_2 = 0$ , а значит,  $A$  является граничной точкой  $\mathcal{H}(f)$ .

Второе утверждение очевидно – если  $\mathcal{A} = O_\varepsilon(A)$  – полностью лежит в  $\mathcal{H}(f)$ , то для каждой точки внутри  $\mathcal{A}$  интервал также имеет внутреннюю точку. Таким образом строится окрестность в  $\mathcal{G}(f)$ .

Третье утверждение вытекает из того, что внутренняя точка конуса решений системы однородных линейных неравенств обращает неравенства в строгие [6].  $\square$

## 3.2 Критерий линейной разделимости

Определим операцию конкатенации булевых функций  $*$  следующим образом: если  $f_1 = f_1(x_1, \dots, x_{n-1})$  и  $f_2 = f_2(x_1, \dots, x_{n-1})$ , а  $f = f_1 * f_2$ , то  $f = f(x_1, \dots, x_{n-1}, x_n)$  и

$$f(x_1, \dots, x_{n-1}, x_n) = (\neg x_n \wedge f_1(x_1, \dots, x_{n-1})) \vee (x_n \wedge f_2(x_1, \dots, x_{n-1})).$$

Наглядно конкатенация булевых функций означает, что мы ставим таблицу истинности  $f_1$  на таблицу истинности  $f_2$ .

**Теорема 3.2.**  *$f_1 * f_2$  является функцией, реализуемой одним персептроном, в том и только том случае, когда  $\mathcal{H}(f_1)$  и  $\mathcal{H}(f_2)$  имеют пересечение со внутренней точкой.*

*Доказательство.* Необходимость. Пусть  $X = (Y; x_n)$ ,  $A = (a_0; B; a_n)$ ,  $A$  – вектор, реализующий функцию  $f$ . При  $x_n = -1$  выполняется следующее равенство:

$$a_0 + A \cdot X = a_0 + B \cdot Y - a_n = (a_0 - a_n) + B \cdot Y.$$

Кроме того, если  $x_n$  равно  $-1$ , то  $f(X) = f_1(Y)$ . Поэтому  $B \in \mathcal{H}(f_1)$ , и является внутренней точкой. Доказательство для  $f_2$  аналогично.

Достаточность. Возьмем  $A \in \mathcal{H}(f_1) \cap \mathcal{H}(f_2)$ , являющийся внутренней точкой. Пусть  $a_0^1 \in I(f_1, A)$ , а  $a_0^2 \in I(f_2, A)$ , являющиеся внутренними точками (они существуют по лемме 3.4). Пусть  $u - t = a_0^1$ , а  $u + t = a_0^2$ . Определим вектор  $C = (c_0; C)$ , где  $C = (A; t)$ ,  $X = (Y, x_n)$ .

Пусть  $x_n = -1$ . Тогда

$$c_0 + C \cdot X = u + A \cdot Y - t.$$

Поскольку  $a_0^1$  и  $A$  являются внутренними точками  $\mathcal{I}(f_1, A)$  и  $\mathcal{H}(f)$  соответственно, то по лемме 3.4 они строго реализуют функцию  $f_1$  на  $x_n = -1$ , а значит,  $(c_0; C)$  строго реализует эту функцию на  $x_n = -1$ . Аналогично этот же вектор строго реализуют  $f_2$  на  $x_n = 1$ . Тогда  $(c_0; C)$  строго реализуют  $f$ , что и требовалось доказать.  $\square$

### 3.3 Обобщения критерия линейной разделимости

Можно рассмотреть возможные обобщения приведенных утверждений на случай произвольных функций распознавания. В частности, леммы 3.1–3.4 справедливы для любой функции распознавания. Доказательство текстуально повторяет приведенные доказательства для булевых функций.

Рассмотрим теперь для операции  $*$  следующее обобщение. Пусть  $f_1, f_2, \dots, f_m$  – произвольные функции распознавания на носителе  $\mathbb{L}$ . Тогда мы также можем определить их конкатенацию  $f = *_m(f_1, \dots, f_m)$  следующим образом:

$$f(X; x_n) = \begin{cases} f_1(X) & , \quad x_n = a_1 \\ f_2(X) & , \quad x_n = a_2 \\ \dots & \\ f_n(X) & , \quad x_n = a_n \end{cases}$$

В этом случае  $f$  будет функцией распознавания на  $\mathbb{L} \times \mathbb{A}$ ,  $\mathbb{A} = \{a_1, \dots, a_m\}$ .

Для такой обобщенной операции  $*_m$  можно доказать два факта. Во-первых, если  $m = 2$ , то теорема 3.2 остается истинной, и ее доказательство текстуально повторяет доказательство, приведенное для булевых функций.

Однако, для  $m > 2$  теорема 3.2 перестает являться достаточным условием. В частности, рассмотрим  $f_+$  и  $f_-$  – функции на одноэлементном носителе  $\mathbb{L} = \{0\}$ ,  $f_+$  обозначает тождественную истину,  $f_-$  – тождественную ложь. Ясно, что  $\mathcal{H}(f_+) = \mathcal{H}(f_-) = \mathbb{R}$ . Однако  $*_3(f_+, f_-, f_+)$  не может быть реализована одним перспетроном.

### 3.4 Свойства операции $*$

Операция конкатенации булевых функций может представлять самостоятельный интерес. В этом параграфе мы докажем некоторые свойства этой операции.

**Предложение 3.1.** Если  $f = f_1 * f_2$  – монотонна, то  $f_1, f_2$  – монотонны.

*Доказательство.* Непосредственно следует из того факта, что  $f_1 = f(X; -1)$ ,  $f_2 = f(X; 1)$ .  $\square$

**Предложение 3.2.** Если  $f = f_1 * f_2$ ,  $g = f_2 * f_1$ , то  $g(X; X_n) = f(X; -X_n)$ .

*Доказательство.* Непосредственно следует из определения операции  $*$ .  $\square$

**Предложение 3.3.** Если  $f = f_1 * f_2$  и  $f_1, f_2$  – монотонны, то:

- 1) Если  $f_1 = f_2$ , то  $f$  не зависит существенно от  $X_n$ .
- 2) Если  $f_1 \leq f_2$ , тогда и только тогда  $f$  – монотонна.
- 3) Если  $f_1 \not\leq f_2$  и  $f_2 \not\leq f_1$ , то  $f_1 * f_2, f_2 * f_1$  – немонотонны.

*Доказательство.* Первое утверждение очевидно:  $f = f_1 * f_2 = (\neg X_n \wedge f_1) \vee (X_n \wedge f_1) = f_1$ .

Второе утверждение докажем от противного – пусть  $f_1 \not\leq f_2$ . Тогда существует  $Y$  такой, что  $f_1(Y) = 1$  и  $f_2(Y) = -1$ . Тогда  $f(Y; -1) > f(Y; 1)$ , но поскольку  $(Y; -1) \prec (Y; +1)$ , то  $f_1 * f_2$  не монотонна. Доказательство достаточности полностью аналогично.

Третье утверждение следует из доказательства второго.  $\square$

**Предложение 3.4.** Если  $f_1 * f_2 \in \mathfrak{M}$ , то  $f_1 * f_2 = f_1 \vee (X_n \wedge f_2)$ .

*Доказательство.* При значении  $X_n = -1$   $f_1 \vee (X_n \wedge f_2) = f_1 * f_2$ . При  $X_n = +1$  это выражение равно  $f_1 \vee f_2$ , но поскольку  $f_1 \leq f_2$ , то  $f_1 \vee f_2 = f_2$ .  $\square$

**Предложение 3.5.** Обозначим через  $\tilde{f}$  функцию, двойственную к  $f$ . Пусть  $f = f_1 * f_2 \in \mathfrak{M}$ , тогда  $\tilde{f} = \tilde{f}_2 * \tilde{f}_1$ .

*Доказательство.*

$$\widetilde{f_1 * f_2} = (\neg X_n \wedge f_1) \vee (X_n \wedge f_2) = (\neg \widetilde{X_n \wedge f_1}) \wedge (\widetilde{X_n \wedge f_2}) = (\neg X_n \vee \tilde{f}_1) \wedge (X_n \vee \tilde{f}_2)$$

Раскрывая скобки, получим:

$$(\neg X_n \wedge X_n) \vee (\neg X_n \wedge \tilde{f}_2) \vee (X_n \wedge \tilde{f}_1) \vee (\tilde{f}_1 \wedge \tilde{f}_2)$$

$\neg X_n \wedge X_n$  является тождественной ложью. Далее  $f_1 \leq f_2$ , следовательно  $\tilde{f}_1 \geq \tilde{f}_2$ , и тогда  $\tilde{f}_1 \wedge \tilde{f}_2 = \tilde{f}_2$ , а такая добавка ни на что не влияет, как было показано ранее. Следовательно, остается:

$$(\neg X_n \wedge \tilde{f}_2) \vee (X_n \wedge \tilde{f}_1)$$

что в точности равно  $\tilde{f}_2 * \tilde{f}_1$   $\square$

**Предложение 3.6.** Пусть  $f = f_1 * f_2$  – самодвойственна и монотонна тогда и только тогда, когда  $f_1 = f_2$ .

*Доказательство.* Из предыдущего предложения мы знаем, что  $(f_1 * f_2) = \tilde{f}_2 * \tilde{f}_1$ , откуда непосредственно следует, что  $f_1 = \tilde{f}_2$ . Обратное же условие очевидно.  $\square$

**Предложение 3.7.** Пусть  $f_1 = f_1(X; Y; Z)$ ,  $f_2 = f_2(X; Y; Z)$  – монотонные булевы функции, причем  $f_1$  не зависит существенно от  $Z$ , а  $f_2$  – от  $X$ . Тогда  $f_1 * f_2$  монотонна только в том случае, если для любого  $Y'$  либо  $f_1(X; Y'; Z)$  – тождественная ложь, либо  $f_2(X; Y'; Z)$  – тождественная истина (понимается тождественность при фиксированных переменных  $Y$ ).

*Доказательство.* Зафиксируем некоторое  $Y'$ . Пусть существует  $X'$  такое, что  $f_1(X'; Y'; Z) = 1$  для всех  $Z$ , ведь  $f_1$  существенно от  $Z$  не зависит. Поскольку  $f_2 \geq f_1$  (иначе  $f_1 * f_2$  не монотонна), то  $f_2(X'; Y'; Z) = 1$  для любого  $Z$ . Но  $f_2$  существенно от  $Z$  не зависит, а значит,  $f_2(X, Y', Z) = 1$ .

Аналогично доказывается, что если  $f_2(X, Y', Z)$  не тождественная истина, то  $f_1(X, Y', Z)$  – тождественная ложь.  $\square$

**Предложение 3.8.** Любая булева функция может быть записана с помощью тождественной лжи, тождественной истины и операции  $*$ .

*Доказательство.* При определении конкатенации мы отметили, что таблица истинности  $f_1 * f_2$  представляет собой таблицу истинности  $f_2$ , сверху на которую «поставлена» таблица истинности  $f_1$ . Любая таблица истинности может быть «собрана» из нулей и единиц, что и доказывает утверждение предложения.  $\square$

**Теорема 3.3.** Пусть  $f = f(X_1, \dots, X_n) = f_1 * f_2$ ,  $g = g(Y_1, \dots, Y_m)$ , а  $h = f(X_1, X_2, \dots, X_{n-1}, g(Y_1, \dots, Y_m))$ . Тогда, если взять запись  $g$  в базисе 0, 1, ast и заменить каждый 0 на  $f_1$ , а каждую единицу – на  $f_2$ , результат будет равен  $h$ . Иными словами, операция композиции булевых функций легко выражается через  $*$ .

*Доказательство.* Доказательство этой теоремы удобно провести в виде таблиц истинности. Рассмотрим таблицу истинности  $h$ . Очевидно, что если  $g(Y) = 1$ , то  $h(X, Y) = f_2(X)$ , а если  $g(Y) = 0$ , то  $h(X, Y) = f_1$ . Тогда, если в таблицу истинности  $g$  добавить переменные  $X_1, \dots, X_n$ , а вместо единиц и нулей подставить  $f_2$  и  $f_1$  соответственно, то получится в точности таблица истинности  $h$ , что и доказывает утверждение теоремы.  $\square$

Также можно сформулировать предложения для линейно-разделимых булевых функций.

**Предложение 3.9.** Если  $f$  – линейно-разделима, то  $0 * f$  и  $f * 1$  также являются линейно-разделимыми. Если  $f$  – линейно-разделима и монотонна, то  $w * f$  и  $f * v$  являются монотонными, где  $w$  – многомерная конъюнкция  $(x_1 \wedge x_2 \wedge \dots x_n)$ , а  $v$  – многомерная дизъюнкция  $(x_1 \vee x_2 \vee \dots x_n)$

*Доказательство.*  $n$ -мерная тождественная единица имеет реализующий конус, равный  $\mathbb{R}^n$ , поэтому, по теореме, она может быть конкатенирована с любой функцией. Аналогично с тождественным нулем. Кроме того, стоит отметить, что  $0 * f = f \wedge X_n$ , а  $f * 1 = f \vee X_n$ . Это тривиально следует, например, из факта представления  $*$  в виде поставленных друг на друга таблиц истинности.

Аналогично, реализующими конусами  $w$  и  $v$  являются  $\mathbb{R}_+^n$ , где  $\mathbb{R}_+$  – множество неотрицательных чисел. Поскольку любая линейно-разделимая монотонная функция распознается персептроном с неотрицательными весами, второе утверждение также справедливо.  $\square$

Суммируя вышеперечисленные предложения, можно сделать несколько замечаний. Во-первых, на множестве  $\mathfrak{M}$  операцию  $*$  можно сделать коммутативной – поскольку разве лишь один результат из  $f_1 * f_2$  и  $f_2 * f_1$  будет лежать в  $\mathfrak{M}$ . Затем, мы получили также возможность строить ДНФ для функции, полученной с помощью  $*$  из двух функций, ДНФ которых нам известна. Кроме того, мы получили вполне хорошее средство для порождения всех монотонных функций из монотонных функций предыдущих арностей, а поскольку применение  $*$  допускает алгоритм, который бы проверял не только то, что мы остаемся в  $\mathfrak{M}$ , но и то, что мы остаемся в  $\mathfrak{B}^+$ , то на его основании можно построить  $\mathfrak{B}^+$ -функции по крайней мере на малых арностях.

В заключение приведем пример использования конкатенации булевых функций для решения хорошо известной задачи – алгоритма построения полинома Жегалкина для функции. Пусть  $f$  представима полиномом Жегалкина в виде

$$f(x_1, \dots, x_n) = \bigoplus_{y_1, \dots, y_n \in \{0,1\}} x_1^{y_1} x_2^{y_2} \dots x_n^{y_n} f^p(y_1, \dots, y_n)$$

где  $f^p$  – «свертка» функции  $f$ , кодирующая коэффициенты при полиноме Жегалкина, а  $x^y$  – символическое обозначение следующей булевой функции:

$x$	$y$	$x^y$
0	0	1
0	1	0
1	0	1
1	1	1

Иначе говоря,  $x^y$  – альтернативная запись  $y \rightarrow x$ .

Определим правила построения таблицы истинности  $f^p$  по таблице истинности  $f$ . Если  $f$  – нульмерная функция, т.е. 0 или 1, то  $f^p$ , очевидно, равно  $f$ . Также очевидно, что  $(f_1 \oplus f_2)^p = f_1^p \oplus f_2^p$ , и если  $f$  не зависит от  $x$ , то  $(xf)^p = xf^p$ .

Пусть  $f = f_1 * f_2$ . Выразим конкатенацию в базисе Жегалкина:

$$f_1 * f_2 = (\neg x_n \wedge f_1) \vee (x_n \wedge f_2) = (\neg x_n \wedge f_1) \oplus (x_n \wedge f_2) \oplus (\neg x_n \wedge f_1 \wedge x_n \wedge f_2)$$



Последнее слагаемое равно 0, так как  $\neg x_n \wedge x_n$  тождественно ложно. Продолжая преобразования, получим, что  $f_1 * f_2 = f_1 \oplus x_n(f_1 \oplus f_2)$ . Далее проведем преобразования для  $(f_1 * f_2)^p$ .

$$(f_1 * f_2)^p = (f_1 \oplus x_n(f_1 \oplus f_2))^p = f_1^p \oplus (x_n(f_1 \oplus f_2))^p = f_1^p \oplus x_n(f_1^p \oplus f_2^p)$$

Это преобразование позволяет нам свести вычисление  $f^p$  к вычислению  $f_1^p$  и  $f_2^p$ , аргументы которых меньше.

Соответственно, алгоритм заключается в следующем преобразовании таблицы истинности. Таблица истинности выписывается в строку. После этого выполняется  $n$  итераций, где  $n$  – аргумент функции. В течение итерации  $i$  таблица разбивается на блоки размера  $2^i$ , после чего к каждому четному блоку прибавляется предыдущий нечетный. После последней итерации, получается таблица истинности  $f^p$ , по которой выписывается полином Жегалкина в соответствии с определением.

Например, рассмотрим функцию  $f(x, y, z) = (x \rightarrow y) \oplus z$ . Выпишем таблицу истинности и проведем итерации алгоритма.

1	0	1	0	0	1	1	0
1	1	1	1	0	1	1	1
1	1	0	0	0	1	1	0
1	1	0	0	1	0	1	0

Выписываем полином Жегалкина:

$$x^0 y^0 z^0 \oplus x^0 y^0 z^1 \oplus x^1 y^0 z^0 \oplus x^1 y^1 z^0,$$

или, упрощая

$$1 \oplus z \oplus x \oplus xy.$$

Нетрудно видеть, что этот полином действительно является полиномом Жегалкина для функции  $(x \rightarrow y) \oplus z$ .

## 4 Алгоритмы проверки линейной делимости булевых функций

На основе теоремы 3.2 можно разработать Алгоритм 1, который по образующим конусам  $\mathcal{H}(f_1)$  и  $\mathcal{H}(f_2)$  будет строить образующие конусы для  $\mathcal{G}(f_1 * f_2)$ .

Пусть  $\mathcal{H} = \mathcal{H}(f_1) \cap \mathcal{H}(f_2)$ , а  $f = f_1 * f_2$ . Пусть  $A \in \mathcal{H}$ .

Рассмотрим интервалы  $\mathcal{I}(f_1, A) = [a_1, b_1]$  и  $\mathcal{I}(f_2, A) = [a_2, b_2]$ . Как видно из доказательства теоремы 3.2, для любых  $(u, t)$  таких, что  $u + t \in [a_2, b_2]$ ,  $u - t \in [a_1, b_1]$  веса  $(u; A; t)$  будут реализовывать  $f$ , но, возможно, нестрого. Множество всех таких пар  $(u, t)$  является выпуклым, и его вершины определяются так:

$$\begin{aligned}
I^1(A) &= \left( \frac{b_2+b_1}{2}, \frac{b_2-b_1}{2} \right) \\
I^2(A) &= \left( \frac{b_2+a_1}{2}, \frac{b_2-a_1}{2} \right) \\
I^3(A) &= \left( \frac{a_2+b_1}{2}, \frac{a_2-b_1}{2} \right) \\
I^4(A) &= \left( \frac{a_2+a_1}{2}, \frac{a_2-a_1}{2} \right)
\end{aligned}$$

Это выпуклое множество обозначим как  $\mathcal{R}(A)$ .

Через  $\lambda\mathcal{R}(A) + \mu\mathcal{R}(B)$  обозначим четырехугольник, вершины которого получаются как конические комбинации соответствующих вершин. Из леммы 3.3 непосредственно следует, что  $\mathcal{R}(\lambda A + \mu B) = \lambda\mathcal{R}(A) + \mu\mathcal{R}(B)$ . Это следует из леммы 3.3.

Теперь вспомним, что  $\mathcal{H}$  является пересечением конусов, и, следовательно, сам является конусом с конечным числом образующих  $H_1, \dots, H_m$ . По теореме 3.2, только векторы из этого конуса могут выступить как реализаторы функции  $f$ . Любой из этих векторов можно записать как коническую  $H_i$ , и в этом случае дополнительные координаты  $(u, t)$  будут внутри соответствующей конической комбинации  $\mathcal{R}(H_i)$ . Следовательно, можно записать систему образующих для  $\mathcal{G}(f)$  в виде  $G_{i,j} = (C_{i,j}; H_j; D_{i,j})$ , где  $i = 1 \dots 4$ ,  $j = 1 \dots m$ ,  $C_{i,j}$  – первая координата вектора  $I^i(H_j)$ ,  $D_{i,j}$  – вторая координата того же вектора.

Обозначим множество всех линейно-разделимых  $n$ -арных функций через  $\mathfrak{N}(n)$ , а всех линейно-разделимых функций – через  $\mathfrak{N}$ . На основании изложенных выкладок сформулируем алгоритм  $\mathfrak{A}$ , порождающий список функций из  $\mathfrak{N}(n)$  по имеющемуся списку функций из  $\mathfrak{N}(n-1)$ . А именно, этот алгоритм пробегает все пары функций из  $\mathfrak{N}(n-1)$  и строит для каждой пары функций их конкатенацию. Если конкатенация является линейно-разделимой, то функция вносится в выходной список. В последующих параграфах мы изучим некоторые дополнительные свойства линейно-разделимых функций, которые позволят нам существенно сократить этот алгоритм.

#### 4.1 Сокращение алгоритма за счет отрицаний аргументов

Рассмотрим метод сокращения перебора за счет отбрасывания немонотонных функций. Покажем, что все функции из  $\mathfrak{N}$  являются монотонными с точностью до отрицаний аргументов. Для этого нам понадобится следующее утверждение.

**Лемма 4.1.** Пусть  $\mathbb{T}_1(n)$  – множество векторов с положительными весами. Тогда функция  $f \in \mathfrak{N}(n)$  является монотонной тогда и только тогда, когда существует вектор  $A$  из  $\mathbb{T}_1(n)$ , реализующий  $f$ .

*Доказательство.* Необходимость. Пусть  $f$  реализуется вектором  $A$ ,  $a_i > 0$ . Рассмотрим векторы  $X, Y$  такие, что  $X \prec Y$  и  $X$  непосредственно предшествует  $Y$ . Тогда для некоторой координаты  $i$  верно, что  $y_i = 1$  и  $x_i = -1$ , а для любой координаты  $j$ ,  $j \neq i$  выполняется  $y_j = x_j$ . Тогда  $A \cdot Y = A \cdot X + 2a_i$ .

Следовательно,  $A \cdot Y > A \cdot X$ , поскольку  $a_i > 0$ . Отсюда вытекает, что  $f(Y) \geq f(X)$ . Поэтому  $f$  – монотонна.

Достаточность. Аналогично доказывается, что если  $a_i < 0$ , то  $f$  антимонотонна по  $i$ -ому аргументу. Если же существуют векторы  $A, B$ , реализующие  $f$ , причем  $a_i < 0 < b_i$ , то функция, очевидно, не зависит существенно от  $i$ -ого аргумента.  $\square$

**Теорема 4.1.** Пусть  $f \in \mathfrak{N}(n)$ . Тогда существует монотонная функция  $g \in \mathfrak{N}(n)$  и  $\alpha_1, \dots, \alpha_n \in \{\neg, \varepsilon\}$  такие, что  $f = g(\alpha_1 x_1, \dots, \alpha_n x_n)$ .

*Доказательство.* Пусть  $A$  – реализующий вектор для  $f$ ,  $a_0$  – вес активации. Тогда через  $g$  обозначим функцию, которую реализует вектор  $A' = (|a_1|, \dots, |a_n|)$  с тем же весом активации  $a_0$  – по лемме 1,  $g$  является монотонной. Пусть  $\alpha_i = \varepsilon$ , если  $a_i \geq 0$ , и  $\alpha_i = \neg$  в противном случае. Через  $X'$  обозначим вектор  $(\alpha_i x_i, \dots, \alpha_n x_n)$ . Рассмотрим цепочку тождеств:

$$f(X) \equiv \text{sign}(a_0 + A \cdot X) \equiv \text{sign}(a_0 + A' \cdot X') \equiv g(\alpha_1 x_i, \dots, \alpha_n x_n).$$

Первое и последнее тождество верны, поскольку  $f$  и  $g$  реализуются векторами  $A$  и  $A'$ , а промежуточное – поскольку  $a_i x_i = |a_i| \text{sign}(a_i) x_i = a'_i x'_i$ .  $\square$

Подмножества  $\mathfrak{M}$  и  $\mathfrak{N}(n)$ , состоящие из монотонных функций, обозначим через  $\mathfrak{M}_1$  и  $\mathfrak{M}_1(n)$  соответственно. Из теоремы 4.1 следует, что мы можем ограничить  $\mathfrak{A}$  на класс  $\mathfrak{M}_1$ , а прочие функции можно получить, при необходимости, применением отрицаний.

**Теорема 4.2.** Если  $f_1 * f_2 \in \mathfrak{M}_1$ , то  $f_1, f_2 \in \mathfrak{M}_1$ .

*Доказательство.* Поскольку  $f_1 * f_2 \in \mathfrak{M}$ , по лемме 4.1 существует вектор  $A$  с положительными компонентными, реализующий  $f_1 * f_2$ . Представим  $A$  в виде  $A = (B; a_n)$ . По теореме о конкатенации  $B \in \mathcal{H}(f_1) \cap \mathcal{H}(f_2)$ . Следовательно,  $B \in \mathcal{H}(f_1)$ . Согласно лемме 4.1,  $f_1$  – монотонна, аналогично доказывается монотонность  $f_2$ .  $\square$

Из теоремы 4.2 следует, что для получения  $\mathfrak{M}_1(n)$  достаточно перебирать конкатенации пар функций из  $\mathfrak{M}_1(n-1)$ . Соответственно, формулируется алгоритм  $\mathfrak{A}_1$ , который перебирает пары функций из входного списка  $\mathfrak{M}_1(n-1)$ , а также не включает в выходной список функции не из  $\mathfrak{M}_1(n)$ .

## 4.2 Сокращение алгоритма за счет порядка аргументов

Покажем теперь, каким образом можно отказаться от порождения функций, отличающихся лишь перестановкой аргументов. Введем понятие ранга  $i$ -ого аргумента:

$$r_i = |\{(x_1, \dots, x_n) : x_i = 1 \wedge f(x_1, \dots, x_n) = 1\}|.$$

Зафиксируем две выбранные координаты  $i$  и  $j$ ,  $i < j$ . Запись  $[a, b, Z]$ , где  $a, b$  – числа, а  $Z$  –  $(n-2)$ -мерный вектор будет обозначать вектор  $(x_1, \dots, x_n)$ ,

где  $x_i = a$ ,  $x_j = b$ , а все прочие координаты записаны в векторе  $Z$  в порядке возрастания. Запись  $f[a, b, Z]$  будет обозначать запись  $f(x_1, \dots, x_n)$  для координат соответствующего  $X$ .

**Лемма 4.2.** Пусть  $f \in \mathfrak{N}_1$ .

1. Если существует  $Z$  такой, что  $f[0, 1, Z] < f[1, 0, Z]$ , то: а) для любого  $Z'$  справедливо  $f[0, 1, Z'] \leq f[1, 0, Z']$ ; б) для любого  $A \in \mathcal{H}(f)$  выполняется  $a_i > a_j$ ; в)  $r_i > r_j$ .

2. Если существует  $Z$  такой, что  $f[0, 1, Z] > f[1, 0, Z]$ , то справедливы выводы, обратные выводам первого пункта.

3. Если  $Z$  ни для первого, ни для второго пункта не существует, то первые два аргумента перестановочны.

*Доказательство.* Докажем первый пункт, применяя рассуждения, близкие к использованным в доказательстве теоремы 4.1. Пусть необходимый  $Z$  существует. Пусть  $A$  – реализующий вектор  $f$ , и  $A = [a_i, a_j, B]$ . Поскольку  $f[0, 1, Z] < f[1, 0, Z]$ , то справедливо, что

$$-a_i + a_j + B \cdot Z < a_i - a_j + B \cdot Z.$$

Следовательно,  $a_i > a_j$ , что доказывает вывод 1.б. Поэтому для любого  $Z'$  справедливо

$$-a_i + a_j + B \cdot Z' < a_i - a_j + B \cdot Z'.$$

Из этого вытекает, что  $f[0, 1, Z'] \leq f[1, 0, Z']$ , что доказывает вывод 1.а.

Рассмотрим теперь ранги аргументов  $i$  и  $j$ . Из 1.а. следует, что если  $f[0, 1, Z'] = 1$ , то и  $f[1, 0, Z'] = 1$ . Следовательно,  $r_i \geq r_j$ . Но из условия существует  $Z$  такой, что  $f[0, 1, Z'] = 0$  при  $f[1, 0, Z'] = 1$ , следовательно,  $r_i > r_j$ .

Второй пункт доказывается аналогично. Третий пункт очевиден.  $\square$

**Теорема 4.3.** Для любой функции  $f$  можно построить функцию  $f'$  путем сортировки аргументов в порядке убывания ранга. Все такие сортировки дают одинаковые функции  $f'$ .

*Доказательство.* Различные варианты сортировки по убыванию ранга отличаются лишь положением аргументов с одинаковыми рангами. Но такие переменные перестановочны по лемме 4.2. Из этого следует, что различные способы сортировки дают одинаковые функции. Следовательно, определение  $f'$  корректно.  $\square$

Теорема 4.3 дает нам возможность рассматривать для функций из  $\mathfrak{N}_1$  каноническую форму, соответствующую перестановке аргументов в порядке убывания ранга. Через  $\mathfrak{N}_2$  обозначим множество всех канонических функций, т.е. функций, совпадающих со своей канонической формой. Поскольку сортировка переменных обратима, то любая функция из  $\mathfrak{N}_1$  получается из своей канонической формы перестановкой аргументов. Поэтому имеет смысл модифицировать  $\mathfrak{A}_1$  для порождения множества  $\mathfrak{N}_2$ .

Обозначим через  $\mathbb{T}_2$  множество векторов из  $\mathbb{R}^n$  с неотрицательными компонентами, упорядоченными в порядке возрастания. Из леммы 4.2 непосредственно вытекает, что  $f \in \mathfrak{N}_2$  тогда и только тогда, когда существует вектор  $A \in \mathbb{T}_2$ , реализующий функцию  $f$ .

**Теорема 4.4.** *Если  $f_1 * f_2 \in \mathfrak{N}_2$ , то и  $f_1, f_2 \in \mathfrak{N}_2$ .*

*Доказательство.* Используя утверждение, что  $f_1 * f_2$  реализуется вектором из  $\mathbb{T}_2$ , доказательство данного утверждения проводится по схеме доказательства теоремы 4.2.  $\square$

По аналогии с теоремами 4.1 и 4.2, теоремы 4.3 и 4.4 дают возможность определить алгоритм  $\mathfrak{A}_3$ , отличающийся от  $\mathfrak{A}_2$  тем, что он перебирает лишь пары функций из  $\mathfrak{N}_2$ , и включает в результат тоже лишь функции из  $\mathfrak{N}_2$ .

### 4.3 Оценка вычислительной сложности алгоритмов

Рассмотрим теперь вычислительную сложность приведенных алгоритмов. Пусть  $T(n)$  – время проверки принадлежности функции классу  $\mathfrak{N}(n)$ , а  $m_1(n)$ ,  $m_2(n)$ ,  $m_3(n)$  – мощности соответственно  $\mathfrak{N}(n)$ ,  $\mathfrak{N}_1(n)$ ,  $\mathfrak{N}_2(n)$ . Ясно, что сложность изложенных алгоритмов составляет  $O(m_i(n) \cdot T(n))$ .

Для  $T(n)$  известна верхняя дважды экспоненциальная оценка. Однако, на практике оказывается, что проверка осуществляется намного быстрее. Также известно, что  $m_1(n) = 2^{\Theta(n^2)}$  [?]. Из теорем 4.1 и 4.3 непосредственно вытекает, что  $m_2 \approx \frac{m_1(n)}{2^n}$ , а  $m_3(n) \approx \frac{m_2(n)}{n!}$ . Приведенные оптимизации позволяют существенно сократить время вычислений, в частности, вычисления для арности 7 занимают примерно 1,5 часа времени работы персонального компьютера [?].

## 5 Классификация линейно-разделимых функций

В этом разделе мы обобщим приведенные свойства линейно-разделимых булевых функций. Как мы уже показали, любая линейно-разделимая функция может быть сведена к монотонной, и любая монотонная – к канонической. Мы продолжим эти сведения и укажем метод классификации линейно-разделимых булевых функций на основании этого подхода.

### 5.1 Основные определения

Рассмотрим  $n$ -мерный вектор  $X = (x_1, \dots, x_n)$  и индексное множество  $M = \{m_1, \dots, m_k\}$ ,  $1 \leq m_i \leq n$ . Проекцией  $X$  на  $M$  назовем вектор  $X|_M = (x_{m_1}, \dots, x_{m_k})$ . Дополнительным индексным множеством  $\bar{M}$  называется такое множество, для которого  $M \uplus \bar{M} = \{1, \dots, n\}$ . Наконец, если  $|M| = k$ ,  $Y \in \mathbb{R}^k$ ,  $Z \in \mathbb{R}^{(n-k)}$ ,  $V \in \mathbb{R}^n$ ,  $V|_M = Y$ ,  $V|_{\bar{M}} = Z$ , то  $V$  называется смещением  $Y$  и  $Z$  по  $M$  и обозначается как  $X \circ_M Y$ .

Пусть  $E_k$  —  $k$ -мерный вектор вида  $(1, -1, \dots, -1)$ , а  $O_k$  —  $k$ -мерный вектор вида  $(-1, 1, \dots, 1)$ . Назовем булеву функцию нерегулярной на  $M$ , если существуют  $Y_1, Y_2$  такие, что  $f(E_k \circ_M Y_1) > f(O_k \circ_M Y_1)$  и  $f(E_k \circ_M Y_2) < f(O_k \circ_M Y_2)$ . В противном случае  $f$  регулярна на  $M$ . Функция  $f$  является регулярной, если она регулярна на любом индексном множестве. Регулярная функция  $f$  является инвариантной на  $M$ , если для всех  $X$  выполняется  $f(E_k \circ_M X) = f(O_k \circ_M X)$ . Наконец, регулярная и неинвариантная функция является верхней на  $M$ , если для некоторого  $X$  выполняется  $f(E_k \circ_M X) > f(O_k \circ_M X)$ , и нижней в противном случае.

Обозначим через  $\mathfrak{N}_k(n)$  множество всех линейно-разделимых функций, которые являются верхними либо инвариантными относительно всех  $M$  таких, что  $|M| \leq k$ . Очевидно, что это определение согласуется с ранее данными определениями  $\mathfrak{N}_1(n)$  и  $\mathfrak{N}_2(n)$ .

Построим теперь множества  $\mathbb{T}_3(n), \mathbb{T}_4(n), \dots$  как продолжение серии  $\mathbb{T}_1(n), \mathbb{T}_2(n)$ . А именно,  $\mathbb{T}_k(n)$  есть подмножество  $\mathbb{T}_{k-1}(n)$  такое, что для любого  $A \in \mathbb{T}_2(n)$  выполняется:

$$a_i \geq \sum_{j=1}^k a_{i+j}.$$

## 5.2 Свойства множеств $\mathfrak{N}_k(n)$

Рассмотрим некоторые важные для построения классификации свойства множеств  $\mathfrak{N}_k(n)$ .

### Теорема 5.1.

1. *Всякая линейно-разделимая функция регулярна на любом индексном множестве  $M$ .*
2. *Для того, чтобы функция принадлежала к  $\mathfrak{N}_k(n)$  необходимо и достаточно, чтобы она реализовывалась вектором из  $\mathbb{T}_k(n)$ .*
3. *Если  $f_1 * f_2 \in \mathfrak{N}_k(n)$ , то  $f_1, f_2 \in \mathfrak{N}_k(n-1)$ .*

*Доказательство.*

1. Рассмотрим линейно-разделимую функцию  $f$ , которая реализуется произвольным вектором  $A$ , и произвольное индексное множество  $M$ . Пусть  $A = B \circ_M C$ . Предположим, что существуют  $Y_1, Y_2$  такие, что  $f(E_k \circ_M Y_1) > f(O_k \circ_M Y_1)$  и  $f(E_k \circ_M Y_2) < f(O_k \circ_M Y_2)$ .

Рассмотрим произведение  $A \cdot (E_k \circ_M Y_1)$ . Представим  $A$  в виде  $A = B \circ_M C$ , в этом случае

$$A \cdot (E_k \circ_M Y_1) = (B \circ_M C) \cdot (E_k \circ_M Y_1) = B \cdot E_k + C \cdot Y_1.$$

Аналогично,  $A \cdot (O_k \circ_M Y_1) = B \cdot O_k + C \cdot Y_1$ . Поскольку  $f(E_k \circ_M Y_1) > f(O_k \circ_M Y_1)$ , то  $A \cdot (E_k \circ_M Y_1) > A \cdot (O_k \circ_M Y_1)$ , а значит,  $B \cdot E_k > B \cdot O_k$ . С другой стороны, проводя аналогичные рассуждения для  $Y_2$ , приходим к

выводу, что  $B \cdot E_k < B \cdot O_k$ , что является противоречием. Следовательно, любая линейно-разделимая функция регулярна.

2. Для того, чтобы функция была верхней либо инвариантной на  $M$ , по определению требуется, чтобы  $f(E_k \circ_M X) \geq f(O_k \circ_M X)$  для всех  $X$ . Это происходит тогда и только тогда, когда некоторый реализующий вектор  $A = B \circ_M C$  обладает свойством  $A \cdot (E_k \circ_M X) \geq A \cdot (O_k \circ_M X)$ , что выполняется тогда и только тогда, когда  $B \cdot E_k \geq B \cdot O_k$ .

Поскольку  $B \cdot E_k = b_1 - b_2 - \dots - b_k$ , а  $B \cdot O_k = -b_1 + b_2 + \dots + b_k$ , получаем, что

$$b_1 > b_2 + \dots + b_k.$$

Поскольку это должно быть справедливо для любого  $M$ , выберем серию  $M_1, \dots, M_r$  таких, что  $M_i$  содержит  $k$  последовательно возрастающих натуральных чисел, начиная с  $i$ . В этом случае, получаем ограничения:

$$a_i \geq \sum_{j=1}^k a_{i+j},$$

которые являются определением множества  $\mathbb{T}_k(n)$ . Понятно, что для любых других индексных множеств приведенных ограничений достаточно для выполнения требования к функции быть инвариантной либо верней.

3. Доказательство аналогично приведенным ранее для монотонных и канонических функций. Поскольку  $\mathcal{H}(f_1 * f_2) \cap \mathbb{T}_k(n)$  содержит внутренние точки, то и  $\mathcal{H}(f_1) \cap \mathbb{T}_k(n-1)$  содержит внутренние точки, а значит,  $f_1 \in \mathfrak{N}_k(n-1)$ .  $\square$

Уже было установлено, что  $\mathfrak{N}_1(n)$  состоит из монотонных линейно-разделимых функций,  $\mathfrak{N}_2(n)$  – из канонических. Изучим множества  $\mathfrak{N}_{n-1}(n)$  и  $\mathfrak{N}_n(n)$ . Пусть  $t_{i,n}$  – булева функция от  $n$  аргументов, в таблице истинности которой в первых  $i$  строках находятся значения «ложь», а в остальных – «истина». Пусть  $q_n$  –  $n$ -арная булева функция, полученная конкатенацией  $(n-1)$ -арной конъюнкции и  $(n-1)$ -арной дизъюнкции.

**Теорема 5.2.** *Множество  $\mathfrak{N}_n(n)$  равно множеству функций  $t_{i,n}$  для всех  $i$  от 0 до  $2^n$ .*

*Доказательство.* Поскольку множество  $\mathfrak{N}_n(n)$  верхнее по всем индексным множествам, это означает, что:

$$a_i \geq \sum_{j=i+1}^n a_j.$$

Минимальной последовательностью, обладающей таким свойством, очевидно является последовательность вида  $B = (16, 8, 4, 2, 1)$ . Увеличение любой компоненты такой последовательности ведет к увеличению всех предыдущих. Понятно, что при умножении подобного вектора  $B$  на вектор аргументов булевой функции  $X$  получается двоичное число, соответствующее

этому набору аргументов в смысле числа в двоичном виде  $-b(X)$ . Отсюда немедленно вытекает, что  $f(Y) \geq f(X)$ , если  $b(X) < b(Y)$ . А значит, все эти функции имеют вид  $t_{i,n}$ . □

### 5.3 Построение классификации

Для завершения продолжения идей, примененных для анализа монотонных и канонических функций, необходим способ приведения произвольной функции к функции из  $\mathfrak{N}_k(n)$ . Рассмотрим следующие функционалы:

$$[\xi, M]f(X \circ_M Y) = \begin{cases} f(X \circ_M Y) & , \quad X \neq E_k, X \neq O_k \\ f(E_k \circ_M Y) & , \quad X = O_k \\ f(O_k \circ_M Y) & , \quad X = E_k \end{cases},$$

$$[\zeta, M]f(X \circ_M Y) = \begin{cases} [\xi, M]f, & \text{если } f \text{ нижняя на } M \\ f, & \text{иначе} \end{cases}.$$

В частности,  $\xi[f, \{i\}]$  эквивалентно отрицанию  $i$ -ого аргумента, а  $\xi[f, \{i, j\}]$  – перестановке  $i$ -ого и  $j$ -ого аргументов.

Назовем функцию  $(\zeta, k)$ -замкнутой, если для любого  $M$ ,  $|M| \leq k$ ,  $f' = [\zeta, M]f$  и  $f'$  либо равна  $f$ , либо не является линейно-разделимой. По определению, все функции из  $\mathfrak{N}_k(n)$  являются  $(\zeta, k)$ -замкнутыми. Наконец, функция является  $\zeta$ -замкнутой, если она  $(\zeta, k)$ -замкнута для  $k = n$ .

Рассмотрим произвольную не  $\zeta$ -замкнутую функцию. По определению, существует  $M$  такое, что  $f' = [\zeta, M]f$  является линейно-разделимой функцией, причем  $f' \neq f$ . Если  $f'$  не  $\zeta$ -замкнута, то она может быть приведена к следующей функции, и так далее. Докажем, что в подобной цепочке преобразований невозможны циклы.

**Теорема 5.3.** Пусть  $f_1, \dots, f_m$  – произвольные не  $\zeta$ -замкнутые функции, причем  $f_i = [\zeta, M]f_{i-1}$  для некоторого  $M$ . Тогда для любых  $i \neq j$ ,  $f_i \neq f_j$ .

*Доказательство.* Рассмотрим произвольную функцию  $f$ , нижнюю на  $M$ . В ходе применения  $[\zeta, M]$  преобразования, функция меняет значения  $f(E_k \circ_M X)$  на  $f(O_k \circ_M X)$  для всех  $X$ . Поскольку по условию  $f$  нижняя на  $M$ , то  $f(E_k \circ_M X) \leq f(O_k \circ_M X)$ , причем для некоторого  $X$  неравенство строгое.

Рассмотрим числа  $b(E_k \circ_M X)$  и  $b(O_k \circ_M X)$ , соответствующие представлению вектора в виде двоичного числа. Поскольку  $b(E_k) > b(O_k)$  по определению  $E_k$  и  $O_k$ , то и  $b(E_k \circ_M X) > b(O_k \circ_M X)$ . Таким образом, в таблице истинности значение «истина» на  $i$ -ой строке поменялось местами со значением «ложь» на  $j$  строке, причем  $i < j$ . При этом, никакое значение «истина» не переместилось на строку с меньшим номером.

Рассмотрим теперь число  $d(f)$ , которое равно числу, двоичная запись которого располагается в таблице истинности  $f$ , считая верхнюю строчку младшим разрядом. В силу вышеприведенных фактов, очевидно что для



всех функций  $f_i$  из условия выполняется  $d(f_{i+1}) > d(f_i)$ . Следовательно, циклы в цепочке преобразований невозможны.  $\square$

Таким образом, любая функция может быть приведена к некоторой  $\zeta$ -замкнутой функции. В связи с этим мы можем рассматривать область функции  $f$  –  $\text{Area}(f)$ , которое определено как множество всех функций, которые могут быть приведены к  $f$  путем применения  $\zeta$  функционалов.

Примеры  $\zeta$ -замкнутых функций уже были рассмотрены: это функции  $t_{i,n}$ , составляющие множество  $\mathfrak{N}_n(n)$ .

#### Теорема 5.4.

1. Если  $f \in \text{Area}(t_{i,n})$ , то  $f$  имеет  $i$  ровно ложных значений в таблице истинности.
2. Для различных  $i, j$ ,  $\text{Area}(t_{i,n})$  и  $\text{Area}(t_{j,n})$  не пересекаются.

*Доказательство.*

1. На основе доказательства теоремы 5.3 очевидно, что при применении  $\zeta$  преобразования количество ложных значений в таблице истинности не изменяется. Поскольку  $t_{i,n}$  имеет  $i$  ложных значений, то и любая  $f \in \text{Area}(t_{i,n})$  также должна иметь  $i$  ложных значений.

2. Непересечение областей очевидно следует из предыдущего пункта.  $\square$

Обозначим через  $\mathfrak{R}(n)$  объединение  $\bigcup \text{Area}(t_{i,n})$  для всех  $i$ , через  $\mathfrak{R}$  – объединение всех  $\mathfrak{R}(n)$ . В этом случае, для  $\mathfrak{R}(n)$  справедлива следующая теорема.

**Теорема 5.5.** Для любой функции из  $f \in \mathfrak{R}(n) \cap \mathfrak{N}_k(n)$  существует функция  $g \in \mathfrak{R}(n) \cap \mathfrak{N}_{k+1}(n)$  и цепочка  $\zeta$ -преобразований такая, что она приводит  $f$  к  $g$ .

*Доказательство.* Поскольку любая функция из  $\mathfrak{R}(n)$  может быть приведена к  $t_{i,n}$ , а все  $t_{i,n}$  принадлежат  $\mathfrak{N}_k(n)$  при всех  $k$ , теорема очевидно справедлива.  $\square$

Таким образом, на основании этой теоремы мы можем схематично изобразить структуру множества  $\mathfrak{R}$ :

$$\begin{array}{ccccccc}
 \mathfrak{R}_1(1) & \xrightarrow{*} & \mathfrak{R}_1(2) & \xrightarrow{*} & \mathfrak{R}_1(3) & \xrightarrow{*} & \mathfrak{R}_1(4) \xrightarrow{*} \dots \\
 & & \downarrow \zeta & & \downarrow \zeta & & \downarrow \zeta \\
 & & \mathfrak{R}_2(2) & \xrightarrow{*} & \mathfrak{R}_2(3) & \xrightarrow{*} & \mathfrak{R}_2(4) \xrightarrow{*} \dots \\
 & & & & \downarrow \zeta & & \downarrow \zeta \\
 & & & & \mathfrak{R}_3(3) & \xrightarrow{*} & \mathfrak{R}_3(4) \xrightarrow{*} \dots \\
 & & & & & & \downarrow \zeta \\
 & & & & & & \dots
 \end{array}$$

Стрелки, помеченные \*, указывают на то, что один класс может быть получен из другого путем применения конкатенации, а стрелки, помеченные  $\zeta$  – что функции первого класса сводятся к другому  $\zeta$ -преобразованиями.

В настоящий момент неизвестно, верно ли что  $\mathfrak{R} = \mathfrak{N}$ . В ходе вычислительного эксперимента, основанного на идеях второй главы, были получены списки всех линейно-разделимых функций аргументности до семи включительно, и было установлено, что  $\mathfrak{R}(n) = \mathfrak{N}(n)$  для всех  $n < 8$ . В то же время, доказательство равенства для всех  $n$  не получено.

Однако, даже в случае, если  $\mathfrak{N} \neq \mathfrak{R}$ , это означает лишь, что множество  $\zeta$ -замкнутых функций расширяется некоторыми новыми функциями. Области этих функций, совместно с  $\mathfrak{R}$ , позволяют породить все  $\mathfrak{N}$ . Поэтому примененный нами метод остается возможным и в случае, если  $\mathfrak{R} \neq \mathfrak{N}$ , и в данном случае исследование и классификация линейно-разделимых функций сводится к исследованию нетривиальных  $\zeta$ -замкнутых функций.

## 6 Линейно-порожденные частичные

В этом разделе мы рассмотрим обобщение линейно-порожденных порядков. До сих пор все линейно-порожденные порядки, которые мы рассматривали, являлись линейными, т.е. элементы носителя  $\mathbb{L}$  выстраивались в ряд. Однако, возможно также изучение частичных порядков, в которые выстраиваются элементы  $\mathbb{L}$ .

Под частичным порядком будем понимать транзитивное и рефлексивное отношение на  $\mathbb{L}$ . Отметим, что мы не требуем от нашего отношения антисимметричности. Порядки будем обозначать строчными латинскими буквами. Если  $X$  предшествует  $Y$  в отношении  $a$ , мы будем писать  $X \prec_a Y$ .

Частичный порядок  $a$  на  $\mathbb{L}$ , наведенный некоторым множеством векторов  $\mathcal{A} \subset \mathbb{R}^n$ , определяется следующим образом:  $X \prec_a Y$  тогда и только тогда, когда для всех  $A \in \mathcal{A}$  выполняется  $A \cdot X \leq A \cdot Y$ . Отметим, что в этом определении мы используем нестрогие неравенства. Частичный порядок является линейно-порожденным частичным порядком (ЛПЧП), если существует множество векторов, порождающее его. ЛПЧП, порожденный набором векторов  $\mathcal{A}$ , обозначается через  $\text{Lipo}(\mathcal{A})$ .  $\text{Cone}(a)$  обозначает наибольшее по включению множество векторов, порождающее данный порядок. Нас будут интересовать только полные множества, наводящие ЛПЧП, т.е. такие множества, для которых  $\text{Cone}(\text{Lipo}(\mathcal{A})) = \mathcal{A}$ . О таких полных множествах можно сформулировать следующую лемму.

**Лемма 6.1.** *Для того, чтобы множество векторов  $\mathcal{A}$  являлось полным множеством, наводящим ЛПЧП необходимо и достаточно, чтобы оно являлось решением системы неравенств, состоящей из неравенств вида  $A \cdot X \leq A \cdot Y$ , где  $A$  – неизвестный вектор,  $X, Y \in \mathbb{L}$ .*

*Доказательство.* Проверим необходимость. Пусть  $\mathcal{A}$  является полным множеством, наводящим некоторый ЛПЧП  $a$ . Покажем, что оно является решением системы неравенств заданного вида. Построим такую систему. Если

$X \prec_a Y$ , то в систему внесем неравенство  $A \cdot X \leq A \cdot Y$ . Пусть множество решений этой системы  $\mathcal{B}$ . Докажем, что  $\mathcal{A} = \mathcal{B}$ .

Проверим, что  $\mathcal{A} \subset \mathcal{B}$ . Пусть  $A \in \mathcal{A}$ . Это означает, что для любых  $X$  и  $Y$ , если  $X \prec_a Y$ , то  $A \cdot X \leq A \cdot Y$ . Раз так, то  $A$  является решением системы неравенств, и следовательно,  $A \in \mathcal{B}$ . Аналогично доказывается, что  $\mathcal{B} = \mathcal{A}$ .

Проверим достаточность. Пусть  $\mathcal{A}$  является решением системы неравенств вида  $A \cdot X \leq B \cdot Y$ . Покажем, что  $\mathcal{A}$  наводит некоторый ЛПЧП  $a$ . Определим отношение  $\prec_a$  следующим образом:  $X \prec_a Y$ , если для всех  $A \in \mathcal{A}$  выполняется что  $A \cdot X \leq A \cdot Y$ . Очевидно, это отношение рефлексивно и транзитивно, а значит,  $\prec$  является частичным порядком.

Поймем теперь, что  $\mathcal{A}$  является полным множеством, которое порождает этот ЛПЧП. Рассмотрим отношение  $\rho$ , такое что  $X \rho Y$  тогда и только тогда, когда неравенство  $A \cdot X \leq A \cdot Y$  входит в систему. Очевидно, что  $\prec_c$  является рефлексивно-транзитивным замыканием  $\rho$ . Добавление в систему неравенства  $A \cdot X \leq A \cdot X$  не меняет решения системы. Добавление в систему неравенства  $A \cdot X \leq A \cdot Z$ , если неравенства  $A \cdot X \leq A \cdot Y$  и  $A \cdot Y \leq A \cdot Z$  также не меняет решения. Поэтому система, соответствующая отношению  $\prec_c$ , эквивалентна по множеству решений системе, соответствующей  $\rho$ . Таким образом, если  $\mathcal{A}$  является решением исходной системы, то он и является решением системы для ЛПЧП, а значит, является полным.

□

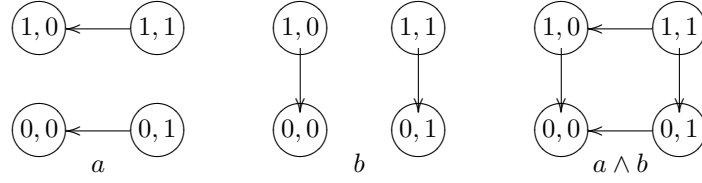
**Следствие 6.1.** *Полное множество, порождающее ЛПЧП, является замкнутым конусом с конечным числом образующих.*

Изучим теперь структуру всех ЛПЧП на некотором множестве  $\mathbb{L}$ . Через  $\mathfrak{L}_1$  обозначим множество ЛПЧП, соответствующих линейно-порожденным линейным порядкам. Более формально, пусть  $\mathcal{A}$  является открытым конечным множеством, порождающим линейный порядок. Через  $\langle \mathcal{A} \rangle$  обозначим замыкание  $\mathcal{A}$  по предельным точкам. Для любого  $\mathcal{A}$ , наводящего линейный порядок, порядок  $\text{Liro}(\langle \mathcal{A} \rangle)$  будет являться элементом  $\mathfrak{L}_1$ . Корректность этого определения докажем в следующем предложении.

**Предложение 6.1.** *Пусть  $\mathcal{A}$  является конусом, порождающим некоторый ЛПП. Тогда  $\mathcal{A}$  является полным множеством, порождающим ЛПЧП.*

*Доказательство.* ЛПП описывается некоторой системой линейных неравенств, и решение этой системы, очевидно,  $\langle \mathcal{A} \rangle$ . Напоминаем, что в решение включаются граничные точки, поскольку неравенства нестрогие. Тогда по лемме 6.1,  $\langle \mathcal{A} \rangle$  является полным множеством. □

Введем оператор инфинума на множестве ЛПЧП. Будем говорить, что  $c$  является нижней границей  $a$  и  $b$ , т.е.  $c = a \wedge b$ , тогда и только тогда, когда  $\text{Cone}(c) = \text{Cone}(a) \cap \text{Cone}(b)$ . Пример такой операции изображен на рисунке. Для наглядности, пример приведен не для ЛПП, а для меньших частичных порядков. Здесь и далее на мы соединяем стрелками только элементы, связанные отношением покрытия, опуская транзитивные следствия.



Пример инфинума для ЛПЧП.

Докажем корректность этого определения следующим предложением.

**Предложение 6.2.** *Для любых двух ЛПЧП  $a$  и  $b$ ,  $a \wedge b$  также является ЛПЧП.*

*Доказательство.* По лемме 6.1, множества  $\text{Cone}(a)$  и  $\text{Cone}(b)$  являются решениями систем линейных неравенств  $M_a$  и  $M_b$ , состоящей из неравенств вида  $A \cdot X \leq A \cdot Y$ ,  $X, Y \in \mathbb{L}$ . Обозначим через  $\mathcal{C}$  пересечение  $\text{Cone}(a)$  и  $\text{Cone}(b)$ . Очевидно,  $\mathcal{C}$  является решением системы линейных неравенств, которая составлена из всех неравенств  $M_a$  и  $M_b$ . Тогда по лемме 6.1,  $\mathcal{C}$  является множеством, наводящим ЛПЧП.  $\square$

Через  $\mathfrak{L}_2$  будем обозначать наименьшее множество, содержащее  $\mathfrak{L}_1$  и замкнутое относительно операции инфинума.

Предположим, что мы ищем инфинум в двух порядков  $a$  и  $b$ , и для некоторых  $X, Y$  верно, что  $X \prec_a Y$  и  $Y \prec_b X$ . В этом случае для всех  $A \in \text{Cone}(a)$  справедливо, что  $A \cdot X \leq A \cdot Y$ , а для всех  $B \in \text{Cone}(b)$  – что  $B \cdot X \geq B \cdot Y$ . Отсюда следует, что для любого  $C \in \text{Cone}(a) \cap \text{Cone}(b)$  выполняется  $C \cdot X = C \cdot Y$ . В данном случае мы говорим, что порядок  $a \wedge b$  склеивает точки  $X$  и  $Y$ . Очевидно, что множества склеенных точек образуют разбиение множества  $\mathbb{L}$ .

Пусть  $(K_1, \dots, K_m)$  – некоторое разбиение  $\mathbb{L}$ , т.е.  $K_i \in 2^{\mathbb{L}}$ ,  $\bigcup_{i=1}^m K_i = \mathbb{L}$  и для различных  $i, j$  множества  $K_i$  и  $K_j$  не пересекаются. Будем называть такое разбиение структурой склеивания, если существует вектор  $A$  такой что:

1. для любого  $i$ , для любых  $X, Y$  из  $K_i$ ,  $A \cdot X = A \cdot Y$
2. для различных  $i, j$ , для любых  $X \in K_i$  и  $Y \in K_j$  выполняется что  $A \cdot X \neq A \cdot Y$ .

Иначе говоря, структура склеивания – это такое разбиение точек, что существует некоторый вектор, склеивающий точки из одного класса структуры, и не склеивающий точки из различных классов. Будем называть структуру склеивания нетривиальной, если хотя бы один из ее классов не пуст. Будем говорить, что ЛПЧП  $a$  соответствует структуре склеивания, если для всех пар  $X, Y$  таких что  $X, Y \in K_i$  для некоторого  $i$  верно, что  $X \prec_a Y$  и  $Y \prec_a X$ .

**Теорема 6.1.** Пусть  $\mathfrak{A}$  – множество всех ЛПЧП над  $\mathbb{L}$ , которые удовлетворяют структуре склеивания  $(K_1, \dots, K_m)$ . Тогда существует  $m$ -элементное множество  $\mathbb{L}'$  и линейное отображение  $\varphi$ , удовлетворяющие следующему свойству. Если  $\mathfrak{A}'$  – множество всех ЛПЧП над  $\mathbb{L}'$ , то для любого  $a \in \mathfrak{A}$  существует  $a' \in \mathfrak{A}'$ , что  $\varphi(\text{Cone}(a)) = \text{Cone}(a')$ .

*Доказательство.* Пусть  $\mathcal{U}$  – гиперплоскость, определяемая системой уравнений

$$\{ A \cdot X = A \cdot Y,$$

где  $X, Y$  пробегают всевозможные пары векторов внутри классов разбиения, т.е.  $X, Y \in K_i$ , для  $i = 1, \dots, m$ . Пусть  $\mathcal{V}$  – ортогональное дополнение для  $\mathcal{U}$ . Тогда  $\mathbb{R}^n$  можно разложить в прямую сумму ортогональных подпространств. Рассмотрим вектор  $X \in \mathbb{L}$ . Поскольку  $\mathbb{L} \subset \mathbb{R}^n$ , то  $X$  может быть представлен в виде  $X = X_u + X_v$ , где  $X_u \in \mathcal{U}$ ,  $X_v \in \mathcal{V}$ .

Пусть  $a$  – ЛПЧП из  $\mathfrak{A}$ . Очевидно, что  $\text{Cone}(a) \subset \mathcal{U}$ . Рассмотрим  $A \in \text{Cone}(a)$ . При скалярном умножении на  $X \in \mathbb{L}$  получаем:

$$A \cdot X = A \cdot (X_u + X_v) = A \cdot X_u + A \cdot X_v = A \cdot X_u,$$

поскольку  $X_v \in \mathcal{V}$  и следовательно  $X_v \perp A$ . Таким образом, при рассмотрении ЛПЧП, соответствующий структуре склеивания,  $X$  может быть заменен на  $X_u$ . Очевидно, что если  $X, Y \in K_i$ , то  $X_u = Y_u$ . Определим теперь  $\mathbb{L}' = \{X_u : X \in \mathbb{L}\}$ . Понятно, что поскольку для  $X, Y$  из одного класса разбиения  $X_u = Y_u$ , то  $|\mathbb{L}'| = m$ .

Пусть  $(U_1, \dots, U_k)$  – базис  $\mathcal{U}$ , а  $(E_1, \dots, E_k)$  – естественный базис  $\mathbb{R}^k$ . Пусть  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^k$ , определенная только на  $\mathcal{U}$ . Тогда определим  $\varphi$  следующим образом:

$$\varphi(X) = \varphi(x_1 U_1 + x_2 U_2 + \dots + x_k U_k) = x_1 E_1 + x_2 E_2 + \dots + x_k E_k.$$

Понятно, что таким образом определенная является линейным биективным отображением  $\mathcal{U}$  на  $\mathbb{R}^k$ .  $\square$

Из доказательства теоремы также непосредственно вытекает следующее предложение.

**Предложение 6.3.** Если ЛПЧП  $a$  соответствует нетривиальной структуре разбиения, то  $\text{Cone}(a)$  не содержит внутренних точек в  $\mathbb{R}^n$ .

Теорема 6.1 позволяет свести «вырожденные» ЛПЧП, т.е. ЛПЧП, конусы которых не содержат внутренних точек, к невырожденным ЛПЧП на другом множестве.

Введем теперь на множестве ЛПЧП операцию супремума. А именно,  $a \vee b$  является ЛПЧП таким, что множество, его образующее является минимальным замкнутым коническим множеством, содержащее конусы  $a$  и  $b$  и являющееся объединением конусов из  $\mathfrak{L}_2$ .

**Теорема 6.2.** Пусть  $a$  и  $b$  – ЛПЧП. Тогда  $a \vee b$  также является ЛПЧП.

*Доказательство.* Проведем сначала доказательство для невырожденных ЛПЧП, т.е. ЛПЧП, конусы которых имеют внутреннюю точку.

Сначала покажем, что  $a \vee b$  всегда определено. Действительно,  $\mathbb{R}^n$  является замкнутым коническим множеством, которое содержит конусы для любых  $a$  и  $b$ . По теореме 2.2,  $\mathbb{R}^n$  разбивается на конусы ЛПП и их граничные точки. Следовательно, по предложению 6.1,  $\mathbb{R}^n$  разбивается на множество конусов ЛПЧП, которые соответствуют ЛПП. Все ЛПЧП, соответствующие ЛПП, по определению принадлежат  $\mathcal{L}_1$ , следовательно, принадлежат и  $\mathcal{L}_2$ .

Теперь покажем, что если  $\mathcal{A}$  является замкнутым коническим множеством и является объединением конусов из  $\mathcal{L}_2$ , то  $\mathcal{A}$  является полным множеством, наводящим ЛПЧП. По лемме 6.1, множество является полным множеством, если оно является решением системы уравнений вида  $A \cdot X \leq A \cdot Y$ ,  $X, Y \in \mathbb{L}$ . Покажем, что  $\mathcal{A}$  является таким решением.

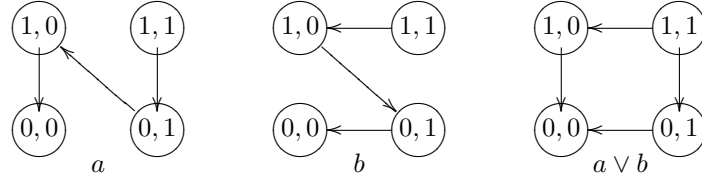
В самом деле, поскольку  $\mathcal{L}_2$  конечно для любого конечного  $\mathbb{L}$ , то  $\mathcal{A}$  является объединением конечного числа множеств. Это означает, что любая граница  $\mathcal{A}$  является границей какого-либо конуса из  $\mathcal{L}_2$ . Но конусы из  $\mathcal{L}_2$  ограничиваются только гиперплоскостями, соответствующими уравнениям вида  $A \cdot X = A \cdot Y$ . Поскольку  $\mathcal{A}$  также является коническим множеством, то его границы тоже являются такими гиперплоскостями. Тогда  $\mathcal{A}$  является решением системы уравнений требуемого вида.

Наконец, докажем единственность минимального замкнутого конического множества, которое является объединением конусов из  $\mathcal{L}_2$  и содержит два данных конуса. В самом деле, пусть  $\mathcal{A}$  и  $\mathcal{B}$  – два различных множества, удовлетворяющих этому требованию. Тогда  $\mathcal{A} \cap \mathcal{B}$  также удовлетворяет этому требованию, что означает, что  $\mathcal{A}$  и  $\mathcal{B}$  не минимальны, т.к.  $\mathcal{A} \cap \mathcal{B} \subset \mathcal{A}$  и  $\mathcal{A} \cap \mathcal{B} \subset \mathcal{B}$ .

Рассмотрим теперь доказательство для вырожденных ЛПЧП. Пусть  $a$ ,  $b$  – вырожденные ЛПЧП, и  $\mathcal{A}$  – минимальная гиперплоскость в  $\mathbb{R}^n$ , содержащая  $\text{Cone}(a)$  и  $\text{Cone}(b)$ . Если  $\mathcal{A} = \mathbb{R}^n$ , то доказательство проводится по схеме доказательства для невырожденных ЛПЧП. Если же  $\mathcal{A}$  обладает меньшим рангом, чем  $n$ , то требуется другая схема, потому что на  $\mathcal{A}$  отсутствуют конусы ЛПП.

Однако, по теореме 6.1, существует линейное отображение из  $\mathcal{A}$  в  $\mathbb{R}^k$  и множество  $\mathbb{L}'$ , причем  $|\mathbb{L}'| < |\mathbb{L}|$ , такие, что конус любого ЛПЧП в  $\mathcal{A}$  переводится с помощью  $\varphi$  в конус ЛПЧП на  $\mathbb{L}'$ . Пусть  $a'$  и  $b'$  – ЛПЧП над  $\mathbb{L}'$ , соответствующие  $a$  и  $b$ . Возможно два варианта. В первом случае,  $a' \wedge b'$  является невырожденным ЛПЧП над  $\mathbb{L}'$ , и тогда можно провести доказательство для невырожденных ЛПЧП  $a'$  и  $b'$  уже существующим способом, и затем отобразить положив  $a \vee b = \varphi^{-1}(a' \vee b')$ . Если же  $a' \vee b'$  является вырожденным, то существует  $\mathbb{L}''$  меньшей размерности, чем  $\mathbb{L}'$  такое, что  $a', b'$  сводятся к ЛПЧП  $a'', b''$  над  $\mathbb{L}''$ . В данном случае мы проводим индукцию по мощности множеств  $\mathbb{L}, \mathbb{L}', \mathbb{L}''$ , поскольку  $0 < \dots < |\mathbb{L}''| < |\mathbb{L}'| < |\mathbb{L}|$ .  $\square$

Пример операции  $\vee$  изображен на следующем рисунке.



Пусть  $\mathfrak{L}_3$  минимальное множество, содержащее  $\mathfrak{L}_2$  и замкнутое относительно операции супремума и инфимума.

**Теорема 6.3.**  $\mathfrak{L}_3$  состоит из всех ЛПЧП, и только из них.

*Доказательство.* Сначала убедимся, что  $\mathfrak{L}_3$  содержит только ЛПЧП. В самом деле, любой элемент  $\mathfrak{L}_3$  может быть получен из конусов ЛПП (из которых состоит  $\mathfrak{L}_1$ ) с помощью операций  $\vee$  и  $\wedge$ . Если  $a, b$  – ЛПЧП, то  $a \vee b$  и  $a \wedge b$  – ЛПЧП. Следовательно, в  $\mathfrak{L}_3$  содержатся лишь ЛПЧП.

Покажем теперь, что любой невырожденный ЛПЧП может быть получен из ЛПП с помощью операции  $\vee$ . Докажем следующий факт: пусть  $\mathcal{A}$  – конус, наводящий некоторый ЛПП без граничных точек,  $b$  – некоторый невырожденный ЛПЧП,  $\mathcal{B} = \text{Cone}(b)$ . Тогда либо  $\mathcal{A} \subset \mathcal{B}$ , либо  $\mathcal{A} \cap \mathcal{B} = \emptyset$ .

В самом деле, пусть  $\mathcal{A} \cap \mathcal{B} = \mathcal{C}$ ,  $\mathcal{C} \neq \emptyset$  и в то же время  $\mathcal{C} \neq \mathcal{A}$ . Рассмотрим вектор  $A \in \mathcal{C}$  и вектор  $B \in (\mathcal{A} \setminus \mathcal{C})$ . Поскольку  $A$  и  $B$  наводят одинаковый линейный порядок (ведь  $A$  и  $B$  принадлежат  $\mathcal{A}$ ), то  $A \cdot X \leq A \cdot Y \Leftrightarrow B \cdot X \leq B \cdot Y$ . Но тогда  $B \in \mathcal{B}$ , поскольку  $\mathcal{B}$  – полное множество, наводящее ЛПЧП.

Таким образом, конус любого ЛПЧП либо целиком содержит конус ЛПП, либо не пересекается с ним. Пусть для некоего невырожденного ЛПЧП  $a, b_1, \dots, b_k$  – все ЛПЧП из  $\mathfrak{L}_1$ , конусы которых целиком содержатся в  $\text{Cone}(a)$ . Тогда  $a = b_1 \vee \dots \vee b_k$ . В самом деле,  $\text{Cone}(a)$  является коническим множеством по следствию 6.1. Он содержит конусы  $\text{Cone}(b_1), \dots, \text{Cone}(b_k)$ , и не содержит других внутренних точек. Понятно, что  $\text{Cone}(a)$  является наименьшим множеством с подобными свойствами. Тогда по определению операции  $\vee$ ,  $a = b_1 \vee \dots \vee b_k$ .

Доказательство для вырожденных ЛПЧП проводится по индукции, схема которой приведена в предыдущей теореме.  $\square$

**Теорема 6.4.**  $\mathfrak{L}_3$  с операциями  $\vee$  и  $\wedge$  является решеткой.

*Доказательство.* Проверим для операций  $\vee$  и  $\wedge$  свойства операций решетки. Коммутативность и идемпотентность обеих операций. Доказательство ассоциативности  $\wedge$  сводится к ассоциативности пересечения множеств, поскольку

$$\begin{aligned} a \wedge (b \wedge c) &= \text{Lipo}(\text{Cone}(a) \cap (\text{Cone}(b) \cap \text{Cone}(c))) = \\ &= \text{Lipo}((\text{Cone}(a) \cap \text{Cone}(b)) \cap \text{Cone}(c)) = (a \wedge b) \wedge c \end{aligned}$$

Проверим законы поглощения. Рассмотрим  $a \vee (a \wedge b)$ . Поскольку  $\text{Cone}(a \wedge b) = \text{Cone}(a) \cap \text{Cone}(b)$ , то  $\text{Cone}(a \vee (a \wedge b)) \subset \text{Cone}(a)$ . Тогда очевидно, что  $\text{Cone}(a \vee (a \wedge b)) = \text{Cone}(a)$ .

является минимальным коническим множеством, содержащим  $\text{Cone}(a)$ , и следовательно,  $a \vee (a \wedge b)$ .

Рассмотрим теперь выражение  $a \wedge (a \vee b)$ . Поскольку  $\text{Cone}(a \vee b)$  по определению содежит  $\text{Cone}(a)$ , то  $\text{Cone}(a) \cap \text{Cone}(a \vee b) = \text{Cone}(a)$ , и следовательно,  $a \wedge (a \vee b) = a$ .

Покажем ассоциативность  $\vee$ . Докажем, что  $a \vee (b \vee c)$  является минимальным коническим объединением конусов из  $\mathfrak{L}_2$  и содержит  $\text{Cone}(a)$ ,  $\text{Cone}(b)$  и  $\text{Cone}(c)$ . Доказательство этого факта, очевидно, докажет ассоциативность  $\vee$ .

Итак, пусть  $N = a \vee (b \vee c)$ , а  $M$  – это минимальное коническое объединение конусов из  $\mathfrak{L}_2$  такое, что оно содержит  $\text{Cone}(a)$ ,  $\text{Cone}(b)$ ,  $\text{Cone}(c)$ . Понятно, что  $N$  содержит  $\text{Cone}(a)$ ,  $\text{Cone}(b)$ ,  $\text{Cone}(c)$ , а значит,  $N \subset M$ , поскольку  $M$  – минимальное множество, обладающее этим свойством.

Покажем теперь, что  $M \subset N$ . Докажем сначала, что  $\text{Cone}(a \vee b) \subset M$ . Пойдем от противного и предположим, что  $M$  не содержит  $\text{Cone}(a \vee b)$ . Возьмем  $K = M \cap \text{Cone}(a \vee b)$ . Поскольку  $M$  и  $\text{Cone}(a \vee b)$  являются коническими множествами, составленными из конусов  $\mathfrak{L}_2$ , то и  $K$  является коническим множеством, составленным из  $\mathfrak{L}_2$ .  $M$  и  $\text{Cone}(a \vee b)$  содержат  $\text{Cone}(a)$  и  $\text{Cone}(b)$ , значит, и  $K$  тоже содержит  $\text{Cone}(a)$  и  $\text{Cone}(b)$ . Но поскольку  $M$  не содержит  $\text{Cone}(a \vee b)$ , то  $K$  является строгим подмножеством  $\text{Cone}(a \vee b)$ , а это невозможно, поскольку  $\text{Cone}(a \vee b)$  – по определению минимальное коническое множество, составленное из  $\mathfrak{L}_2$  и содержащее  $\text{Cone}(a)$  и  $\text{Cone}(b)$ . В результате мы пришли к противоречию.

Итак,  $\text{Cone}(a \vee b) \subset M$ . Также по определению множества  $M$ ,  $\text{Cone}(c) \subset M$ . Тогда  $M \subset N$ , поскольку  $N$ , по определению, минимальное множество, содержащее  $\text{Cone}(a \vee b)$  и  $\text{Cone}(c)$ .

Покажем, что решетка  $(\mathfrak{L}_3, \vee, \wedge)$  является недистрибутивной. (!!!)  $\square$

Итак, множество всех ЛПЧП представляет собой конечную решетку с операциями  $\vee$  и  $\wedge$ . Приведем общее описание этой решетки в нотации глобуса. «Северным полюсом», т.е. наибольшим элементом, является ЛПЧП, наводимый множеством  $\mathbb{R}^n$ . В этом ЛПЧП, очевидно, любые два элемента являются несравнимыми. В самом деле, если  $X \prec Y$  тогда и только тогда, когда для всех  $A$  из  $\mathbb{R}^n$  выполняется что  $A \cdot X \leq A \cdot Y$ , то любые два различных элемента несравнимы. «Южным полюсом», т.е. наименьшим элементом, является ЛПЧП, наводимый множеством  $\{0\}$ . В этом ЛПЧП для любых двух  $X, Y$  верно, что  $X \prec Y$  и  $Y \prec X$ .

Роль экватора выполняет множество  $\mathfrak{L}_1$ , т.е. ЛПЧП, которые соответствуют ЛПП. Ниже экватора располагаются вырожденные ЛПЧП, и, следовательно, область решетки ниже экватора содержит подрешетки, изоморфные решеткам ЛПЧП на других множества. Выше экватора лежат все невырожденные ЛПЧП, в которых присутствуют несравнимые элементы. В частности, там лежат всевозможные ЛПЧП на подмножествах  $\mathbb{L}$ , например, всевозможные ЛПП на всех симплексных подмножествах  $\mathbb{L}$ .

Нам представляется, что построение общего описания этой решетки для произвольных множеств  $\mathbb{L}$  весьма затруднительную. В то же время, для



дальнейших исследований представляет интерес точное строение этой решетки на типовых множествах, в частности, на  $\mathbb{B}^n$ .

## 6.1 ЛПЧП и функции

В этом разделе мы изучим взаимосвязь ЛПЧП и функций, которые реализуются одним нейроном. Введем понятие  $k$ -дольного ЛПЧП. ЛПЧП  $a$  на носителе  $\mathbb{L}$  является  $k$ -дольным, если  $\mathbb{L}$  можно разбить на  $L_1, \dots, L_k$  непересекающихся подмножеств, и для любых  $i < j$ ,  $X \in K_i$ ,  $Y \in K_j$  выполняется  $X \prec_a Y$  и не выполняется  $Y \prec_a X$ .

Рассмотрим функцию активации  $q_k(C_1, \dots, C_k, x)$ , определенную следующим образом:

$$q_k(C_1, \dots, C_{k-1}, x) = \begin{cases} 1, & x < C_1 \\ 2, & C_1 \leq x < C_2 \\ \dots \\ k-1, & C_{k-2} \leq x < C_{k-1} \\ k, & C_{k-1} \leq x \end{cases}$$

Нейрон с весами  $(C; A)$ , где  $C = (C_1, \dots, C_{k-1})$ ,  $A = (A_1, \dots, A_n)$ , работающий по правилу

$$\text{neu}(X) = q_k(C_1, \dots, C_{k-1}, A \cdot X)$$

будем называть обобщенным персептроном.

Понятно, что функция  $q_k$  может быть определена с другими выходными значениями. В общем случае, она выдает не числа  $1, 2, 3, \dots$ , а числа  $b_1, b_2, b_3, \dots$ . Если числа  $b_i$  попарно различны, то вычислительные возможности обобщенного персептрона, очевидно, сохраняются с точностью до замены выходного значения. В этом случае, персептрон является частным случаем обобщенного персептрона, где  $k = 2$  и функция  $q_k$  выдает значения  $-1, 1$ . Мы упростили определение функции  $q_k$  для облегчения изложения.

Пусть  $f$  – некоторая функция из  $\mathbb{L}$  в  $\{1, \dots, k\}$ . Будем говорить, что веса  $(C; A)$  реализуют функцию  $f$ , если  $f(X)$  реализуется обобщенным персептроном с весами  $(C; A)$ . Для функции  $f$  можно ввести множества  $\mathcal{G}(f)$ ,  $\mathcal{H}(f)$ , и  $\mathcal{I}(f, A)$  по аналогии с обычными персептронами следующим образом. Множество  $\mathcal{G}(f)$  – это множество всех  $(C; A)$  таких, что  $(C; A)$  реализуют функцию  $f$ . Множество  $\mathcal{H}(f)$  – множество всех  $A$  таких, что существует  $C$  и  $(A; C) \in \mathcal{G}(f)$ . Наконец, множество  $\mathcal{I}(f, A)$  – это множество всех  $C$  таких, что  $(C; A) \in \mathcal{G}(f)$ . Для этих множеств можно сформулировать обобщения лемм 3.1 и 3.2 следующим образом.

**Теорема 6.5.** *Для любой функции  $f$ ,  $\mathcal{G}(f)$  и  $\mathcal{H}(f)$  являются конечными множествами с конечным числом образующих. Множество  $\mathcal{I}(f)$  является множеством вида  $[a_1, b_1] \times [a_2, b_2] \times \dots \times [a_{k-1}, b_{k-1}]$ .*

*Доказательство.* Пусть  $K_i = f^{-1}(i)$ , где  $f^{-1}$  – полный прообраз  $f$ . Построим системы линейных неравенств для множеств  $\mathcal{G}(f)$  и  $\mathcal{H}(f)$ . Множество

$\mathcal{H}(f)$ , очевидно, определяется системой линейных неравенств (1) вида:

$$\{A \cdot X \leq A \cdot Y ,$$

где  $X \in K_i$ ,  $Y \in K_j$  и  $i < j$ .  $\mathcal{H}(f)$  является решением этой системы, и следовательно  $\mathcal{H}(f)$  – коническое множество с конечным числом образующих.

Множество  $\mathcal{I}(f, A)$ , где  $A \in \mathcal{H}(f)$ , определяется системой линейных неравенств (2):

$$\left\{ \begin{array}{lll} C_1 & \geq & A \cdot X_1 \quad , \quad X_1 \in K_1 \\ C_1 & \leq & A \cdot X_2 \quad , \quad X_2 \in K_2 \\ C_2 & \geq & A \cdot X'_2 \quad , \quad X'_2 \in K_2 \\ C_2 & \leq & A \cdot X_3 \quad , \quad X_3 \in K_3 \\ \dots & & \\ C_{k-1} & \leq & A \cdot X'_{k-1} \quad , \quad X'_{k-1} \in K_{k-1} \\ C_{k-1} & \leq & A \cdot X_k \quad , \quad X_k \in K_k \end{array} \right.$$

Пусть  $r_i = \min_{X \in K_i} A \cdot X$ ,  $s_i = \max_{X \in K_i} A \cdot X$ . Тогда понятно, что систему можно записать в виде (3):

$$\left\{ \begin{array}{lll} C_1 & \geq & s_1 \\ C_1 & \leq & r_2 \\ C_2 & \geq & s_2 \\ C_2 & \leq & r_3 \\ \dots & & \\ C_{k-1} & \geq & s_{k-1} \\ C_{k-1} & \leq & r_k \end{array} \right.$$

Понятно, что решение для системы (3) соответствует приведенному в условии теоремы, т.е.  $\mathcal{I}(f, A) = [s_1, r_1] \times [s_2, r_2] \times \dots \times [s_{k-1}, r_{k-1}]$ .

Системой для  $\mathcal{G}(f, A)$  является объединение систем (1) и (2). Из этого следует, что  $\mathcal{G}(f)$  – коническое множество с конечным числом образующих.  $\square$

ЛПЧП и функции, реализуемые персептроном, связывает следующая теорема.

**Теорема 6.6.** *Если функция  $f$  на носителе  $\mathbb{L}$  реализуется персептроном, то существует  $k$ -долных ЛПЧП такой, что его доли определяются как  $K_i = f^{-1}(i)$ , где  $f^{-1}$  – полный прообраз  $f$ .*

*Доказательство.* Доказательство непосредственно следует из доказательства предыдущей теоремы.  $\square$

# Лекция 6.

## Классификация нейронных сетей.

Нейронные сети обычно различают по трем признакам:

- ❑ структуре сети (типу связей между нейронами),
- ❑ особенностям моделей нейронов,
- ❑ особенностям обучения сети.

По структуре нейронные сети можно разделить на

- ☐ неполносвязные,
- ☐ полносвязные,
- ☐ со случайными связями,
- ☐ с регулярными связями,
- ☐ с симметричными связями,
- ☐ с несимметричными связями.

Неполносвязные нейронные сети обычно описываются неполносвязным ориентированным графом. Их подразделяют на однослойные (простейшие перцептроны) и многослойные.

Многослойные нейронные сети разделяют на сети с прямыми, перекрестными и обратными связями. В нейронных сетях с прямыми связями нейроны какого-либо слоя по входам могут соединяться только с нейронами предыдущих слоев. В нейронных сетях с перекрестными связями допускаются связи внутри одного слоя. В нейронных сетях с обратными связями ограничений на связи нет.

По используемым на входах и выходах сигналам нейронные сети можно разделить на:

- ☐ аналоговые,

- ☐ бинарные.

По моделированию времени нейронные сети подразделяются на

- ☐ сети с непрерывным временем,
- ☐ сети с дискретным временем.

Для программной реализации применяется как правило дискретное время.



По способу подачи информации на входы нейронной сети различают:

- ☐ подачу сигналов на синапсы входных нейронов;
- ☐ подачу сигналов на выходы входных нейронов;
- ☐ подачу сигналов в виде весов синапсов входных нейронов;
- ☐ аддитивную подачу на синапсы входных нейронов.

По способу съема информации с выходов нейронной сети различают:

- ☐ съем с выходов выходных нейронов;
- ☐ съем с синапсов выходных нейронов;
- ☐ съем в виде значений весов синапсов выходных нейронов;
- ☐ аддитивный съем с синапсов выходных нейронов.

По организации обучения разделяют обучение нейронных сетей с учителем и без учителя.

При обучении с учителем предполагается, что есть внешняя среда, которая предоставляет обучающие примеры (значения входов и соответствующие им значения выходов) на этапе обучения или оценивает правильность функционирования нейронной сети и в соответствии со своими критериями меняет состояние нейронной сети или поощряет (наказывает) нейронную сеть, запуская тем самым механизм изменения ее состояния.

Под состоянием нейронной сети, которое может изменяться, обычно понимается:

- ❑ веса синапсов нейронов;
- ❑ веса синапсов и пороги нейронов (обычно в этом случае порог является более легко изменяемым параметром, чем веса синапсов);
- ❑ установление новых связей между нейронами (свойство биологических нейронов устанавливать новые связи и ликвидировать старые называется пластичностью).

По способу обучения разделяют обучение по входам и по выходам.

При обучении по входам обучающий пример представляет собой только вектор входных сигналов, а при обучении по выходам в него входит и вектор выходных сигналов, соответствующий входному вектору.

По способу предъявления примеров различают предъявление одиночных примеров и множеств примеров.

В первом случае изменение состояния нейронной сети (обучение) происходит после предъявления каждого примера. Во втором – после предъявления множества примеров на основе анализа сразу их всех.

# Многослойная нейронная сеть

Из связанных определенным образом нейронов строится нейронная сеть с определенным количеством входов и выходов. Обычно в нейронной сети различают три типа нейронов:

- ☐ входной слой нейронов,
- ☐ выходной слой,
- ☐ скрытые слои нейронов.

Функционирование нейронной сети состоит из двух этапов:

- ❑ обучения сети адекватному реагированию на входную информацию,
- ❑ использования обученной сети.

Обучение многослойной сети может осуществляться разными способами (в том числе способами не учитывающими многослойности структуры).

В обученной сети предполагается, что сигнал передается от слоя к слою, начиная с входного слоя.



Теоретически число слоев и число нейронов в каждом слое может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированной микросхемы, на которых обычно реализуется нейронная сеть.

Чем сложнее нейронная сеть, тем масштабнее задачи, подвластные ей. Выбор структуры нейронной сети осуществляется в соответствии с особенностями и сложностью задачи. Для решения некоторых отдельных типов задач уже существуют оптимальные, на сегодняшний день, конфигурации.

Если задача не может быть сведена ни к одному из известных типов, разработчику приходится решать сложную проблему синтеза новой конфигурации. При этом он руководствуется несколькими основополагающими принципами:

- ❑ возможности сети возрастают с увеличением числа нейронов сети,
- ❑ возможности сети возрастают с увеличением плотности связей между нейронами,
- ❑ возможности сети возрастают с увеличением числа выделенных слоев,
- ❑ введение обратных связей наряду с увеличением возможностей сети поднимает вопрос о динамической устойчивости сети,
- ❑ сложность алгоритмов функционирования сети (в том числе, например, введение нескольких типов синапсов — возбуждающих, тормозящих и др.) также способствует усилению мощности сети.

Вопрос о необходимых и достаточных свойствах сети для решения того или иного рода задач представляет собой целое направление нейрокомпьютерной науки. Так как проблема синтеза нейронных сетей сильно зависит от решаемой задачи, дать общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант получается на основе интуитивного подбора.

Очевидно, что процесс функционирования нейронных сетей, то есть сущность действий, которые она способна выполнять, зависит от величин синаптических связей, поэтому, задавшись определенной структурой нейронных сетей, отвечающей какой-либо задаче, разработчик сети должен найти оптимальные значения всех переменных весовых коэффициентов (некоторые синаптические связи могут быть постоянными).

# Характеристики моделей нейронных сетей

Разные модели нейронных сетей с целью сравнения можно охарактеризовать на основе следующих свойств:

- ☐ структура связей между нейронами;
- ☐ тип входных сигналов;
- ☐ тип выходных сигналов;
- ☐ тип передаточной функции;
- ☐ особенности функционирования при распознавании;
- ☐ особенности алгоритма обучения;
- ☐ емкость сети — количество образов (классов), которое может запомнить и распознавать сеть.

Лекция 7.

Аналитический метод обучения  
сигмоидных сетей.

Учитывая то, что сигмоидная функция активации является непрерывной и дифференцируемой, сигмоидную нейронную сеть можно рассматривать как непрерывную и дифференцируемую функцию нескольких переменных, полагая переменными веса.

Это позволяет сигмоидную нейронную сеть обучать, настраивая веса и(или) параметр сигмоиды путем решения экстремальной задачи для множества векторов обучающей выборки. При этом параметр сигмоиды можно настраивать одинаково для всех нейронов или независимо для каждого.

Совершенно естественно, что точного решения может и не существовать. Поэтому изначально желательно минимизировать какую-либо ошибку, например, среднеквадратичную.

**Задача 1.** Рассмотреть сигмоидную нейронную сеть с двумя скрытыми слоями по три нейрона каждый. Аналитически обучить, настраивая только веса, эту сеть распознаванию функции XOR.

**Задача 2.** Рассмотреть сигмоидную нейронную сеть с двумя скрытыми слоями по два нейрона каждый. Аналитически обучить, настраивая веса и единый параметр сигмоиды, эту сеть распознаванию функции  $xy + 2x$  по точкам (1,1), (2,3) и (5,1). Найти среднеквадратичную ошибку по точкам (1,-1) и (4,-3).



# Лекция 8.

## Метод обратного распространения ошибки.

При обучении методом обратного распространения ошибки необходимо

- ❑ по входному обучающему вектору найти результат, сопоставляя результат с ожидаемым значением, вычислить ошибку сети,
- ❑ разделить ошибку сети пропорционально весам между нейронами последнего скрытого слоя,
- ❑ модифицировать веса пропорционально ошибке и коэффициенту обучения,
- ❑ рассмотреть нейроны последнего скрытого слоя в качестве выходов и повторить процедуру.

**Задача 1.** Рассмотреть нейронную сеть с двумя скрытыми слоями по три нейрона каждый, предполагая, что нейроны имеют стандартную пороговую функцию активации. Методом обратного распространения ошибки обучить эту сеть распознаванию функции XOR.

# Лекция 9.

## Аналитический метод обратного распространения ошибки.

Возможно совмещение метода обратного распространения ошибки и аналитического метода.

Согласно методу наименьших квадратов, минимизируемой целевой функцией ошибки нейронной сети в этом случае является величина:

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2 \quad (1)$$

где  $y_{j,p}^{(N)}$  – реальное выходное состояние нейрона  $j$  выходного слоя  $N$  нейронной сети при подаче на ее входы  $p$ -го образа;  $d_{jp}$  – идеальное (желаемое) выходное состояние этого нейрона.

Суммирование ведется по всем нейронам выходного слоя по всем обрабатываемым сетью образам. Минимизация методом градиентного спуска, что означает подстройку коэффициентов следующим образом:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}}$$

Здесь  $w_{ij}$  — весовой коэффициент синаптической соединяющей  $i$ -ый нейрон слоя  $n-1$  с  $j$ -ым нейроном слоя  $n$ ,  $\eta$  — коэффициент скорости обучения,  $0 < \eta < 1$ .

Имеет место равенство,

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \cdot \frac{\partial s_j}{\partial w_{ij}} \quad (3)$$

Здесь под  $y_j$ , как и раньше, подразумевается выход нейрона  $j$ , а под  $s_j$  — взвешенная сумма его входных сигналов, то есть аргумент активационной функции.

Так как множитель  $dy_j/ds_j$  является производной этой функции по ее аргументу, из этого следует, что производная активационной функция должна быть определена на всей оси абсцисс. В связи с этим функция единичного скачка и прочие активационные функции с неоднородностями не подходят для рассматриваемых нейронных сетей. В них применяются такие гладкие функции, как гиперболический тангенс или классический сигмоид с экспонентой.



В случае гиперболического тангенса

$$\frac{dy}{ds} = 1 - s^2 \quad (4)$$

Третий множитель  $\partial s_j / \partial w_{ij}$ , очевидно, равен выходу нейрона предыдущего слоя  $y_i^{(n-1)}$ .

Что касается первого множителя в соотношении (3), он легко раскладывается следующим образом:

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot w_{jk}^{(n+1)} \quad (5)$$

Здесь суммирование по  $k$  выполняется среди нейронов слоя  $n+1$ .

Введя новую переменную

$$\delta_j^{(n)} = \frac{\partial \mathcal{E}}{\partial y_j} \cdot \frac{dy_j}{ds_j} \quad (6)$$

мы получим рекурсивную формулу для расчетов величин  $\delta_j^{(n)}$  слоя n из величин  $\delta_k^{(n+1)}$  более старшего слоя n+1.

$$\delta_j^{(n)} = \left[ \sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{ds_j} \quad (7)$$

Для выходного же слоя

$$\delta_l^{(N)} = (y_l^{(N)} - d_l) \cdot \frac{dy_l}{ds_l} \quad (8)$$

Теперь мы можем записать (2) в раскрытом виде:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)} \quad (9)$$

Иногда для придания процессу коррекции весов некоторой инерционности, сглаживающей резкие скачки при перемещении по поверхности целевой функции, (9) дополняется значением изменения веса на предыдущей итерации

$$\Delta w_{ij}^{(n)}(t) = -\eta \cdot (\mu \cdot \Delta w_{ij}^{(n)}(t-1) + (1-\mu) \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}) \quad (10)$$

где  $\mu$  – коэффициент инерционности,  $t$  – номер текущей итерации.

Таким образом, полный алгоритм обучения нейронной сети с помощью процедуры обратного распространения строится так:

1. Подать на входы сети один из возможных образов и в режиме обычного функционирования нейронной сети, когда сигналы распространяются от входов к выходам, рассчитать значения последних.

2. Рассчитать  $\delta^{(N)}$  для выходного слоя по формуле (8).

Рассчитать по формуле (9) или (10) изменения весов  $\Delta w^{(N)}$  слоя N.

3. Рассчитать по формулам (7) и (9) (или (7) и (10)) соответственно  $\delta^{(n)}$  и  $\Delta w^{(n)}$  для всех остальных слоев,  $n=N-1, \dots, 1$ .

4. Скорректировать все веса в сети

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t) \quad (11)$$

5. Если ошибка сети существенна, перейти на шаг 1.

В противном случае – конец.

# Нейронные сети и аппроксимация

М. Л. Морнев

29.11.2005

Данный текст посвящен доказательству следующей теоремы:

**Теорема.** Любая непрерывная действительнозначная функция, заданная на  $d$ -мерном единичном кубе  $I^d$  может быть сколь угодно точно равномерно аппроксимирована функциями вида

$$g(x_1, \dots, x_d) = \alpha + \sum_{n=1}^N \beta_n \eta \left[ \gamma_n + \sum_{m=1}^{M_n} \delta_{nm} \eta(\lambda_{nm} + \kappa_{nm} x_m) \right], \quad (1)$$

причем в случае  $d = 1$  можно положить

$$g(x) = \alpha + \sum_{n=1}^N \beta_n \eta(\gamma_n + \delta_n x).$$

Здесь  $\alpha, \beta_n, \gamma_n, \delta_{nm}, \kappa_{nm}, M_n, N$  зависят только от приближаемой функции и точности приближения, а функция  $\eta$  задается так:

$$\eta(x) = \begin{cases} 0, & x < 0, \\ x, & 0 \leq x \leq 1, \\ 1, & x > 1. \end{cases}$$

Перед началом доказательства введем для удобства несколько обозначений. Если  $\vec{a}$  — вектор (координатная строка), то  $a_r$  соответствует его  $r$ -й координате. Покоординатное произведение векторов обозначается  $\vec{a} \cdot \vec{b}$ , а евклидова норма  $|\vec{a}|$ . Множество всех  $d$ -мерных векторов с координатами 0 и 1 обозначается  $B_d$ . Значки частной производной и дифференциала перед вектором обозначают произведения частных производных (дифференциалов) координат:

$$d\vec{a} = da_1 da_2 \cdots da_n, \quad \frac{\partial f(\vec{a})}{\partial \vec{a}} = \frac{\partial^n f(a_1, a_2, \dots, a_n)}{\partial a_1 \partial a_2 \cdots \partial a_n}.$$

При этом порядок частного дифференцирования определяется по размерности вектора аргументов.

Нам пригодится следующая лемма из элементарного анализа:

**Лемма 1.** Если  $f$  — бесконечно дифференцируемая функция на  $I^d$ , и область  $\Omega(\vec{x}) = [0, x_1] \times \cdots \times [0, x_d]$ , то

$$\int_{\Omega(\vec{x})} \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} d\vec{\xi} = \sum_{\vec{b} \in B_d} (-1)^{d+|\vec{b}|^2} f(\vec{b} \cdot \vec{x}).$$

**Доказательство.** Раскрыв обозначения, можно без труда узнать в этой формуле многомерное обобщение формулы Ньютона-Лейбница. Действительно, если  $d = 1$ , то

$$\int_0^x \frac{\partial f(\xi)}{\partial \xi} d\xi = \sum_{b \in \{0,1\}} (-1)^{1+b} f(bx) = f(x) - f(0).$$

Это будет база индукции. Шаг индукции: считаем лемму доказанной для всех размерностей до  $d - 1$ . Тогда

$$\begin{aligned} \int_{\Omega(\vec{x})} \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} d\vec{\xi} &= \int_0^{x_d} \left[ \sum_{\vec{b} \in B_{d-1}} (-1)^{d-1+|\vec{b}|^2} \frac{\partial f(b_1x_1, \dots, b_{d-1}x_{d-1}, \xi_d)}{\partial \xi_d} \right] d\xi_d = \\ &= \sum_{\vec{b} \in B_{d-1}} (-1)^{d-1+|\vec{b}|^2} \left[ f(b_1x_1, \dots, b_{d-1}x_{d-1}, x_d) - f(b_1x_1, \dots, b_{d-1}x_{d-1}, 0) \right] = \\ &= \sum_{\substack{\vec{b} \in B_d \\ b_d=1}} (-1)^{d+|\vec{b}|^2} f(\vec{b} \cdot \vec{x}) + \sum_{\substack{\vec{b} \in B_d \\ b_d=0}} (-1)^{d+|\vec{b}|^2} f(\vec{b} \cdot \vec{x}) = \sum_{\vec{b} \in B_d} (-1)^{d+|\vec{b}|^2} f(\vec{b} \cdot \vec{x}). \end{aligned}$$

Объясним, как были изменены знаки членов сумм в предпоследнем равенстве. Вначале, отметим, что для вектора  $\vec{b}$  из  $B_n$  имеет место  $|\vec{b}|^2 = b_1 + \cdots + b_n$ . Отсюда получим, что

$$(-1)^{d-1+\vec{b}} = (-1)^{d+b_1+\cdots+b_{d-1}-1}(-1)^2 = (-1)^{d+b_1+\cdots+b_d},$$

где  $b_d = 1$ . Аналогично преобразовывался знак при членах второй суммы. Таким образом, лемма доказана. ■

При доказательстве теоремы можно рассматривать только бесконечно дифференцируемые функции вместо непрерывных, поскольку всякую



непрерывную функцию можно сколь угодно точно приблизить бесконечно дифференцируемыми. Само доказательство проведем в два приема.

### §1. Первый прием

Здесь я покажу, что любую бесконечно дифференцируемую действительную функцию  $f$ , заданную на  $I^d$ , можно сколь угодно точно равномерно приблизить функциями вида

$$g(\vec{x}) = \alpha + \sum_{n=1}^N \beta_n \prod_{m=1}^d \eta(\gamma_{nm} + \delta_{nm} x_m). \quad (2)$$

Числа  $\alpha, \beta_n, \gamma_{nm}, \delta_{nm}, N$  зависят только от приближаемой функции и точности приближения. Доказательство проводится индукцией по  $d$ . Поскольку база и шаг здесь практически не отличаются, я приведу их одновременно.

Лемма 1 утверждает, что можно записать

$$f(\vec{x}) = \int_{\Omega(\vec{x})} \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} d\vec{\xi} - \sum'_{\vec{b} \in B_d} (-1)^{d+|\vec{b}|^2} f(\vec{b} \cdot \vec{x}),$$

где штрих при знаке суммирования означает пропуск случая  $b_1 = \dots = b_d = 1$ . Членами этой суммы являются функции от не более чем  $d-1$  переменных, которые мы (согласно индукции) уже умеем приближать. Тогда осталось показать, что мы можем сколь угодно точно аппроксимировать интеграл. Если для любого  $\varepsilon > 0$  мы найдем функцию  $\varphi(\vec{\xi})$ , такую, что

$$\sup_{\vec{\xi} \in I^d} \left| \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} - \varphi(\vec{\xi}) \right| < \frac{\varepsilon}{2},$$

то

$$\sup_{\vec{x} \in I^d} \left| \int_{\Omega(\vec{x})} \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} d\vec{\xi} - \int_{\Omega(\vec{x})} \varphi(\vec{\xi}) d\vec{\xi} \right| \leq \sup_{\vec{x} \in I^d} \int_{\Omega(\vec{x})} \left| \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} - \varphi(\vec{\xi}) \right| d\vec{\xi} \leq \frac{\varepsilon}{2} < \varepsilon.$$

Функцию  $\varphi(\vec{\xi})$  мы будем строить *разрывной*. Начнем с разбиения единичного куба  $I^d$  на мелкие кубики  $K_{\vec{n}}$  со стороной  $1/P$ , где  $P$  — натуральное число:

$$K_{\vec{n}} = \prod_{r=1}^d \left[ \frac{n_r}{P}; \frac{n_r + 1}{P} \right].$$

Координаты  $n_r$   $d$ -мерного вектора  $\vec{n}$  здесь и во всех следующих выражениях пробегает целые значения от 1 до  $P - 1$ . Само число  $P$  выберем так, чтобы колебание функции  $\partial f / \partial \vec{\xi}$  на каждом кубике не превосходило  $\varepsilon/4$  (по лемме о равномерной непрерывности, так всегда можно сделать). Определим функцию  $\varphi(\vec{\xi})$ :

$$\varphi(\vec{\xi}) = R(\vec{\xi}) + \sum_{\vec{n}} y_{\vec{n}}(\vec{\xi}).$$

Здесь

$$y_{\vec{n}}(\vec{\xi}) = \begin{cases} W_{\vec{n}}, & \vec{\xi} \in \ker K_{\vec{n}}, \\ 0, & \text{иначе,} \end{cases} \quad \text{и } W_{\vec{n}} = \sup_{\vec{\xi} \in K_{\vec{n}}} \left| \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} \right|.$$

Функция  $R(\vec{\xi})$  определяется так:

$$R(\vec{\xi}) = \begin{cases} \partial f(\vec{\xi}) / \partial \vec{\xi}, & \text{если для любого } \vec{n} \text{ справедливо } \vec{\xi} \notin \ker K_{\vec{n}}, \\ 0, & \text{иначе.} \end{cases}$$

Пусть  $\vec{\xi} \in \ker K_{\vec{n}}$  для какого то конкретного  $\vec{n}$ . Тогда

$$\left| \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} - \varphi(\vec{\xi}) \right| = \left| \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} - W_{\vec{n}} \right| < \frac{\varepsilon}{4},$$

поскольку колебание функции  $\partial f / \partial \vec{\xi}$  на каждом кубике не превосходит  $\varepsilon/4$ . Если же для любого  $\vec{n}$  точка  $\vec{\xi}$  не входит в  $\ker K_{\vec{n}}$ , то

$$\left| \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} - \varphi(\vec{\xi}) \right| = \left| \frac{\partial f(\vec{\xi})}{\partial \vec{\xi}} - R(\vec{\xi}) \right| = 0.$$

Отсюда уже можем заключить, что  $\|\partial f / \partial \vec{\xi} - \varphi\| \leq \varepsilon/4 < \varepsilon/2$ , что и требовалось показать.

Мы установили, что интеграл функции  $\varphi(\vec{\xi})$ , который обозначим  $\Phi(\vec{x})$ , действительно аппроксимирует  $f$ . Выразим теперь  $\Phi(\vec{x})$  в явной форме. Имеем

$$\Phi(\vec{x}) = \int_{\Omega(\vec{x})} \varphi(\vec{\xi}) d\vec{\xi} = \int_{\Omega(\vec{x})} \left[ R(\vec{\xi}) + \sum_{\vec{n}} y_{\vec{n}}(\vec{\xi}) \right] d\vec{\xi} = \sum_{\vec{n}} \int_{\Omega(\vec{x})} y_{\vec{n}}(\vec{\xi}) d\vec{\xi},$$

поскольку функция  $R(\vec{\xi})$  отлична от нуля лишь на множестве меры нуль.

Если обозначить

$$j(\xi) = \begin{cases} 1, & \xi \in (0, 1) \\ 0, & \text{иначе,} \end{cases}$$

то, с одной стороны,

$$\int_0^x j(\xi) d\xi = \eta(x).$$

С другой стороны, нетрудно видеть, что

$$y_{\vec{n}}(\vec{\xi}) = W_{\vec{n}} \prod_{r=1}^d j\left(\frac{n_r}{P} + \frac{1}{P}\xi_r\right).$$

Тогда,

$$\int_{\Omega(\vec{x})} y_{\vec{n}}(\vec{\xi}) d\vec{\xi} = W_{\vec{n}} \prod_{r=1}^d \int_0^{x_r} j\left(\frac{n_r}{P} + \frac{1}{P}\xi_r\right) d\xi_r = W_{\vec{n}} P^d \prod_{r=1}^d \eta\left(\frac{n_r}{P} + \frac{1}{P}x_r\right).$$

Это выражение имеет вид (2), поэтому, и функция  $\Phi(\vec{x})$  и сумма этой функции и аппроксимаций функций меньшего числа переменных будут иметь вид (2), что и требовалось показать.

Таким образом, осталось показать, что функция  $F(\vec{x}) = \eta(x_1) \cdots \eta(x_d)$  может быть сколь угодно точно приближена выражениями вида (1). Действительно, поскольку при линейной замене координат выражение вида (1) остается в той же форме, то это будет означать справедливость приближения для всех функций формы (2).

## §2. Второй прием

Функцию  $F(\vec{x})$  удобно приближать на всем евклидовом пространстве  $\mathbb{R}^d$ .

Для заданной ошибки  $\varepsilon$  мы будем строить приближение в следующем виде:

$$A(\vec{x}) = \frac{1}{M} \sum_{m=0}^{M-1} h_m(\vec{x}).$$

Функции  $h_m$  удовлетворяют следующим условиям:

$$\begin{aligned} F(\vec{x}) < \frac{m}{M} &\Rightarrow h_m(\vec{x}) = 0, \\ F(\vec{x}) > \frac{m+1}{M} &\Rightarrow h_m(\vec{x}) = 1, \\ \frac{m}{M} \leq F(\vec{x}) \leq \frac{m+1}{M} &\Rightarrow 0 \leq h_m(\vec{x}) \leq 1. \end{aligned}$$

Предположим, что такая система функций построена. Пусть точка  $\vec{x}$  фиксирована. Тогда имеем

$$A(\vec{x}) = \frac{1}{M} \sum_{m=1}^M h_m(\vec{x}) = \frac{1}{M} \left( r + h_{r+1}(\vec{x}) \right),$$

где  $r$  — наименьшее число, такое, что

$$\frac{r}{M} \leq F(\vec{x}) \leq \frac{r+1}{M}.$$

Поскольку  $0 \leq h_{r+1}(\vec{x}) \leq 1$ , то

$$\frac{r}{M} \leq A(\vec{x}) \leq \frac{r+1}{M}.$$

Но раз и  $A(\vec{x})$  и  $F(\vec{x})$  попадают в один и тот же отрезок, то расстояние между ними не может превышать длины отрезка, т. е.  $1/M$ . Но тогда

$$\sup_{\vec{x} \in \mathbb{R}^d} |F(\vec{x}) - A(\vec{x})| \leq \frac{1}{M}.$$

Если теперь выбрать число  $M$  так, чтобы  $1/M \leq \varepsilon/2$ , то непосредственно  $\|F - A\| < \varepsilon$ , что и требовалось показать.

Осталось доказать существование для всякого  $M$  функций  $h_m$ . Для этого нам потребуется ввести несколько вспомогательных понятий.

**Определение 1.** Множество  $A$  из  $\mathbb{R}^d$  называется  $\vartheta$ -множеством, если для любой точки  $\vec{x}$  из  $A$  все точки  $\vec{y}$ , удовлетворяющие условиям

$$y_1 \geq x_1, y_2 \geq x_2, \dots, y_d \geq x_d,$$

также входят в  $\Phi$ .

Как обычно, определяем  $\vartheta$ -пополнение множества  $A$ :

$$\vartheta A = \bigcup_{\vec{x} \in A} \{\vec{y} \in \mathbb{R}^d \mid y_1 \geq x_1, \dots, y_d \geq x_d\}.$$

**Предложение 1.** *Имеют место следующие тривиальные свойства  $\vartheta$ -пополнения ( $A, B$  — подмножества  $\mathbb{R}^d$ ):*

$$A \subseteq B \Rightarrow \vartheta A \subseteq \vartheta B, \quad \vartheta \vartheta A = \vartheta A, \quad \vartheta(A \cup B) = \vartheta A \cup \vartheta B.$$

**Доказательство.** Пусть  $A \subseteq B$  и  $\vec{x}$  входит в  $\vartheta A$ . Тогда найдется такая точка  $\vec{a}$  из  $A$ , что  $\vec{x}$  входит в  $\vartheta \vec{a}$ . Но раз  $A$  включается в  $B$ , то  $\vec{a}$  тоже входит в  $B$ , поэтому,  $\vec{x}$  входит в  $\vartheta B$ .

Перейдем ко второму свойству. Поскольку  $A \subseteq \vartheta A$ , то, по первому свойству,  $\vartheta A \subseteq \vartheta \vartheta A$ . Пусть  $\vec{x}$  входит в  $\vartheta \vartheta A$ . Тогда найдется  $\vec{y}$  из  $\vartheta A$ , такое, что  $\vec{x}$  содержится в  $\vartheta \vec{y}$ . Но раз  $\vec{y}$  входит в  $\vartheta A$ , то найдется  $\vec{z}$  из  $A$ , такое, что  $\vec{y}$  входит в  $\vartheta \vec{z}$ . Но тогда и  $\vec{x}$  входит в  $\vartheta \vec{z}$ , а значит, и в  $\vartheta A$ . Отсюда  $\vartheta \vartheta A \subseteq \vartheta A$  и, как следствие,  $\vartheta A = \vartheta \vartheta A$ .

Рассмотрим третье свойство. Пусть  $\vec{x}$  входит в  $\vartheta(A \cup B)$ . Тогда найдется  $\vec{y}$ , входящее или в  $A$ , или в  $B$ , такое, что  $\vec{x}$  входит в  $\vartheta \vec{y}$ . Тавтологически,  $\vec{x}$  входит или в  $\vartheta A$ , или в  $\vartheta B$ , т. е.  $\vartheta A \cup \vartheta B$ . Значит,  $\vartheta(A \cup B) \subseteq \vartheta A \cup \vartheta B$ . Пусть  $\vec{x}$  входит в  $\vartheta A$  или в  $\vartheta B$ . Тогда найдется  $\vec{y}$  из  $A$ , или из  $B$ , такая, что  $\vec{x}$  входит в  $\vartheta \vec{y}$ , откуда  $\vec{x}$  входит в  $\vartheta(A \cup B)$ . Значит,  $\vartheta A \cup \vartheta B \subseteq \vartheta(A \cup B)$ , что вместе с предыдущим доказывает свойство. ■

**Определение 2.** *Множество  $\Phi$  называется  $\vartheta$ -конусом, если оно является  $\vartheta$ -множеством, и найдется такая точка  $\vec{x}$  из  $\mathbb{R}^d$ , что  $\Phi \subseteq \vartheta\{\vec{x}\}$ .*

Нам пригодится следующая лемма:

**Лемма 2.** *Пусть  $K, L$  являются  $\vartheta$ -конусами, причем конус  $K$  является  $\delta$ -вписанным в  $L$ , т. е.*

$$\forall \vec{x} \in \mathbb{R}^d \quad \text{dist}(\vec{x}, K) < \delta \Rightarrow \vec{x} \in L.$$

*Тогда между ними можно вписать конечно порожденный  $\vartheta$ -конус. Иными словами, найдется такая система точек  $\vec{y}_1, \dots, \vec{y}_p$ , что*

$$K \subseteq \vartheta\{\vec{y}_1, \dots, \vec{y}_p\} \subseteq L.$$

**Доказательство.** Определим множества  $U_m$  следующим образом:

$$U_m = \{\vec{x} \in 2^{-m}\mathbb{Z}^d \mid \text{dist}(\vec{x}, K) \leq \delta/2\}.$$

По условию получается, что при любом натуральном  $m$  имеет место  $U_m \subseteq L$ , откуда  $\vartheta U_m \subseteq L$ . Выберем  $m$  так, чтобы  $2^{-m}\sqrt{d} \leq \delta/2$ . Тогда для всякой точки  $\vec{x}$  из  $K$  расстояние до любой вершины того кубика решетки  $2^{-m}\mathbb{Z}^d$ , в который она попадает, не будет превышать  $\delta/2$ . Но, среди этих вершин найдется и такая вершина  $\vec{y}$ , что  $\vec{x}$  входит в  $\vartheta\vec{y}$  — это будет вершина с наименьшими координатами. Значит,  $\vec{x}$  входит в  $\vartheta U_m$ .

Осталось показать, что всякий конус  $\vartheta U_m$  является конечно порожденным. Я сделаю это индукцией по  $d$ . В случае  $d = 1$  всякий  $\vartheta$ -конус на решетке  $2^{-m}\mathbb{Z}$  порождается своей точкой с наименьшей координатой, чем база доказана. Пусть мы доказали конечную порожденность конусов вплоть до  $d - 1$  измерений. Сделаем это для  $d$ .

Выберем в множестве  $U_m$  произвольную точку  $\vec{w}$ . Обозначим  $H_j$  множество таких точек  $\vec{y}$  из  $U_m$ , что  $y_j < w_j$ . Тогда

$$\vartheta[\{\vec{w}\} \cup H_1 \cup H_2 \cup \dots \cup H_d] = \vartheta U_m.$$

Действительно, все точки из порождающего множества в левой части равенства входят в  $U_m$ , поэтому включение слева на право имеет место. С другой стороны, если какая-то точка  $\vec{x}$  входит в  $\vartheta U_m$ , то найдется  $\vec{y}$  из  $U_m$ , такая, что  $\vec{x} \in \vartheta\vec{y}$ . Если  $\vec{y} \in \vartheta\vec{w}$ , то  $\vec{x} \in \vartheta\vec{w}$ . Если же  $\vec{y} \notin \vartheta\vec{w}$ , то хотя бы для одного  $j$  имеет место  $y_j < w_j$ , откуда  $\vec{y} \in H_j$ . Поэтому, включение справа на лево также справедливо.

Покажем, что любое множество  $\vartheta H_j$  на самом деле конечно порождено. Для любой точки  $\vec{z}$  из  $H_j$  имеет место  $z_j < w_j$ . С другой стороны, поскольку  $U_m$  является  $\vartheta$ -конусом, то найдется  $u_j$ , такая, что  $u_j \leq z_j$ . Поэтому, у всех точек  $\vec{z}$  имеется лишь конечное число различных координат  $z_j$ . Обозначим  $H_j(v)$  множество точек из  $H_j$ , имеющих  $v$  в качестве  $j$ -й координаты. Рассмотрим множество  $\overline{H}_j(v)$ :

$$\overline{H}_j(v) = \{(z_1, \dots, z_{j-1}, z_{j+1}, \dots, z_d) \mid \vec{z} \in H_j(v)\}.$$

Тогда  $\vartheta\overline{H}_j(v)$  является  $\vartheta$ -конусом размерности  $d-1$ , конечно порожденным по предположению индукции. Обозначим порождающее множество  $\overline{H}'_j(v)$ , а также определим

$$H'_j(v) = \{(z_1, \dots, z_{j-1}, v, z_{j+1}, \dots, z_d) \mid (z_1, \dots, z_{j-1}, z_{j+1}, \dots, z_d) \in \overline{H}'_j(v)\}.$$

Очевидно  $\vartheta H'_j(v) \subseteq \vartheta H_j(v)$ . Пусть  $\vec{x}$  входит в  $\vartheta H_j(v)$ . Тогда найдется  $\vec{h}$  из  $H_j(v)$ , такой, что  $\vec{x}$  содержится в  $\vartheta \vec{h}$ . Для вектора  $\vec{k}$ , определенного как

$$\vec{k} = (h_1, \dots, h_{j-1}, h_{j+1}, \dots, h_d),$$

найдется вектор  $\vec{k}'$  из  $\overline{H}'_j(v)$ , такой, что  $\vec{k}$  входит в  $\vartheta \vec{k}'$ . Рассмотрим вектор

$$\vec{h}' = (k'_1, \dots, k'_{j-1}, v, k'_{j+1}, \dots, k'_d).$$

Имеем  $k'_1 \leq h_1, \dots, k'_d \leq h_d$  и  $h'_j = h_j = v$ . Но тогда  $\vec{h}$  входит в  $\vartheta \vec{h}'$ , а значит, туда входит и  $\vec{x}$ . Поэтому, действительно  $\vartheta H'_j(v) = \vartheta H_j(v)$ .

Примем  $H'_j = \bigcup_v H'_j(v)$ , где  $v$  пробегает множество значений из решетки  $2^{-m}\mathbb{Z}$  в пределах от  $u_j$  до  $w_j$ . Как было выше замечено, множество этих значений конечно, поэтому  $H'_j$  конечно. Но также

$$\vartheta H'_j = \bigcup_v \vartheta H'_j(v) = \bigcup_v \vartheta H_j(v) = \vartheta H_j.$$

Если определить  $U'_m = \{\vec{w}\} \cup H'_1 \cup \dots \cup H'_d$ , то получается, что  $\vartheta U'_m = \vartheta U_m$ . Но  $U'_m$  конечно, поэтому  $\vartheta U_m$  конечно порождено, ч. т. д. ■

Введем операцию сдвига множества  $A$  на вектор  $\vec{x}$ :

$$A + \vec{x} = \{\vec{y} + \vec{x} \mid \vec{y} \in A\}.$$

Сдвиг на скаляр определяется как сдвиг на вектор, все координаты которого равны этому скаляру.

**Предложение 2.** Если  $A$  —  $\vartheta$ -множество, то  $A + \vec{x}$  также является  $\vartheta$ -множеством. Если  $A$  —  $\vartheta$ -конус, то  $A + \vec{x}$  также  $\vartheta$ -конус.

**Доказательство.** Пусть  $A$  —  $\vartheta$ -множество. Если  $\vec{y}$  входит в  $A + \vec{x}$ , то  $\vec{y} - \vec{x}$  входит в  $A$ . Если  $\vec{z} \geq \vec{y}$ , то  $\vec{z} - \vec{x} \geq \vec{y} - \vec{x}$ , поэтому  $\vec{z} - \vec{x}$  входит в  $A$ . Но тогда  $\vec{z}$  входит в  $A + \vec{x}$ , откуда  $A + \vec{x}$  является  $\vartheta$ -множеством.

Пусть  $A$  —  $\vartheta$ -конус. Тогда найдется такая точка  $\vec{y}$ , что для всех точек  $\vec{z}$  из  $A$  имеет место  $\vec{y} \leq \vec{z}$ . Мы уже знаем, что для любого  $\vec{x}$  множество  $A + \vec{x}$  является  $\vartheta$ -множеством. Для любой точки  $\vec{z}$  этого множества, точка  $\vec{z} - \vec{x}$  входит в  $A$ . Для нее  $\vec{y} \leq \vec{z} - \vec{x}$ , откуда  $\vec{y} + \vec{x} \leq \vec{z}$  для всех точек множества  $A + \vec{x}$ . Значит, оно также является  $\vartheta$ -конусом. ■

**Предложение 3.**  $\vartheta(A + \vec{x}) = \vartheta A + \vec{x}$ .

**Доказательство.** Пусть  $\vec{y}$  входит в  $\vartheta(A + \vec{x})$ . Тогда найдется точка  $\vec{z}$  из  $A + \vec{x}$ , такая, что  $\vec{y}$  входит в  $\vartheta \vec{z}$ . Точка  $\vec{z} - \vec{x}$  входит в  $A$ , и  $\vec{y} - \vec{x}$

принадлежит  $\vartheta(\vec{z} - \vec{x})$ . Но тогда  $\vec{y}$ , по определению сдвига, принадлежит  $\vartheta(\vec{z} - \vec{x}) + \vec{x}$ , которое входит в  $\vartheta A + \vec{x}$ .

Пусть  $\vec{y}$  входит в  $\vartheta A + \vec{x}$ . Тогда  $\vec{y} - \vec{x}$  входит в  $\vartheta A$ , а значит, найдется такое  $\vec{z}$  из  $A$ , что  $\vec{z} \leq \vec{y} - \vec{x}$ . Но тогда  $\vec{z} + \vec{x} \leq \vec{y}$ , а поскольку  $\vec{z} + \vec{x}$  входит в  $A + \vec{x}$ , то  $\vec{y}$  входит в  $\vartheta(A + \vec{x})$ . Двустороннее включение доказывает равенство. ■

**Предложение 4.** Если  $\vartheta$ -множество  $A$   $\delta$ -вписано в  $\vartheta$ -множество  $B$ , то  $A \subseteq B + \delta/2$ .

**Доказательство.** Пусть точка  $\vec{x}$  входит в  $A$ . Любая точка  $\vec{y}$ , такая, что  $|\vec{x} - \vec{y}| < \delta$ , входит в  $B$ . Для точки  $\vec{y} = \vec{x} - \delta/2$  имеем  $|\vec{x} - \vec{x} + \delta/2| = |\delta/2| < \delta$ , поэтому,  $\vec{x} - \delta/2$  входит в  $B$ . Но это значит, что  $\vec{x}$  входит в  $B + \delta/2$ , что и требовалось доказать. ■

Ключевое значение для всего доказательства имеет следующая лемма:

**Лемма 3.** Пусть  $K, L$  —  $\vartheta$ -конусы, причем  $K$  является  $\delta$ -вписанным в  $L$ , а  $L$  конечно порожден. Тогда существует функция  $f(\vec{x})$  вида (1), такая, что

$$\begin{aligned}\vec{x} \notin L &\Rightarrow f(\vec{x}) = 0, \\ \vec{x} \in L \setminus K &\Rightarrow 0 \leq f(\vec{x}) \leq 1, \\ \vec{x} \in K &\Rightarrow f(\vec{x}) = 1.\end{aligned}$$

**Доказательство.** Пока доказательство есть только для частного случая  $d = 2$ . Пусть  $Y$  — оптимальный базис конуса  $L$ , т. е. такой, что из него нельзя выкинуть ни одной точки без потери куска конуса. Мы будем искать функцию  $f(x_1, x_2)$  в виде  $g_\alpha(x_1, x_2)$  при  $\alpha > 0$ , где

$$g_\alpha(x_1, x_2) = \eta \left[ \sum_{\vec{y} \in Y} (\eta(\alpha x_1 - \alpha y_1) + \eta(\alpha x_2 - \alpha y_2)) - |Y| \right], \quad (3)$$

Эта функция имеет вид (1), и для всякого  $\vec{x}$  справедливо  $0 \leq g_\alpha(x_1, x_2) \leq 1$ .

Для любых двух различных точек  $\vec{y}, \vec{z}$  базиса  $Y$  имеет место следующая альтернатива: либо  $y_1 < z_1$  и  $y_2 > z_2$ , либо  $y_1 > z_1$  и  $y_2 < z_2$ . Доказывая это, отметим, что любые две различные точки базиса  $Y$  отличаются сразу по обоим координатам, поскольку, если бы  $y_1 = z_1$ , то при  $y_2 \leq z_2$  точка  $\vec{z}$  входит в  $\vartheta\vec{y}$  и ее можно выкинуть из базиса, а при  $y_2 \geq z_2$  точка  $\vec{y}$  входит в  $\vartheta\vec{z}$  и ее можно удалить, и аналогично в случае  $y_2 = z_2$ . Если



$y_1 < z_1$ , то при  $y_2 < z_2$  точка  $\vec{z}$  входила бы в  $\vartheta\vec{y}$  и базис  $Y$  был бы неоптимальным, поэтому  $y_2 > z_2$ . Аналогично, в случае  $y_1 > z_1$ .

Покажем, что если  $\vec{x}$  не входит в  $L$ , то  $g_\alpha(x_1, x_2) = 0$ . Поскольку  $\vec{x}$  не входит в  $L$ , для всякой точки  $\vec{y}$  из  $Y$  либо  $x_1 < y_1$ , либо  $x_2 < y_2$ . В первом случае  $\eta(\alpha x_1 - \alpha y_1) = 0$ , а во втором случае  $\eta(\alpha x_2 - \alpha y_2) = 0$ . Поэтому, имеем

$$\sum_{\vec{y} \in Y} (\eta(\alpha x_1 - \alpha y_1) + \eta(\alpha x_2 - \alpha y_2)) \leq |Y|.$$

Поэтому, аргумент внешней функции  $\eta$  в (3) отрицателен и  $g_\alpha(x_1, x_2) = 0$ .

Покажем, что если  $\vec{x}$  входит в  $L + 1/\alpha$ , то  $g_\alpha(x_1, x_2) = 1$ . Поскольку  $\vec{x}$  входит в  $L + 1/\alpha$ , то найдется точка  $\vec{y}$  из  $Y$ , такая, что  $x_1 \geq y_1 + 1/\alpha$  и  $x_2 \geq y_2 + 1/\alpha$  (см. предложение 3). Тогда

$$\eta(\alpha x_1 - \alpha y_1) + \eta(\alpha x_2 - \alpha y_2) = 2.$$

Если  $\vec{z}$  — другая точка базиса, то при  $z_1 < y_1$  имеет место  $\eta(\alpha x_1 - \alpha z_1) = 1$ , а при  $z_2 < y_2$  имеет место  $\eta(\alpha x_2 - \alpha z_2) = 1$  и вообще

$$\eta(\alpha x_1 - \alpha z_1) + \eta(\alpha x_2 - \alpha z_2) \geq 1.$$

Поэтому,

$$\sum_{\vec{y} \in Y} (\eta(\alpha x_1 - \alpha y_1) + \eta(\alpha x_2 - \alpha y_2)) \geq 2 + |Y| - 1 = |Y| + 1,$$

откуда аргумент внешней функции  $\eta$  в (3) больше либо равен 1. Следовательно,  $g_\alpha(x_1, x_2) = 1$ .

По доказанному выше, найдется такое  $\gamma > 0$ , что  $K \subseteq L + \gamma$ . Выбирая  $\alpha$  столь большим, что  $1/\alpha < \gamma$ , мы получаем функцию  $f(x_1, x_2) = g_\alpha(x_1, x_2)$  со всеми требуемыми в условии леммы свойствами. ■

Изучая функцию  $F$ , введем множества  $W(z)$ , которые при  $0 < z \leq 1$  определяются так:

$$W(z) = \{\vec{x} \mid F(\vec{x}) \geq z\}.$$

В дополнение к этому, полагаем  $W(0) = \vartheta\{\vec{0}\}$ , т. е. интуитивно

$$W(0) = \lim_{z \rightarrow 0} W(z).$$

**Утверждение 1.** *Все  $W(z)$  являются  $\vartheta$ -конусами.*

**Доказательство.** Для  $W(0)$  это следует из определения. Теперь покажем, что если  $\vec{x} \leq \vec{y}$ , то  $F(\vec{x}) \leq F(\vec{y})$ . Воспользуемся индукцией по  $d$ . Поскольку  $\eta$  является монотонной, то база индукции справедлива. Далее, из  $\vec{x} \leq \vec{y}$  мы имеем, что

$$\eta(x_1)\eta(x_2)\cdots\eta(x_d) \leq \eta(y_1)\eta(y_2)\cdots\eta(y_d).$$

Кроме того, раз  $x_{d+1} \leq y_{d+1}$ , то  $\eta(x_{d+1}) \leq \eta(y_{d+1})$  и

$$\eta(x_{d+1}) \prod_{r=1}^d \eta(x_r) \leq \eta(x_{d+1}) \prod_{r=1}^d \eta(y_r) \leq \eta(y_{d+1}) \prod_{r=1}^d \eta(y_r),$$

что и требовалось показать.

Пусть  $\vec{x} \in W(z)$  и  $\vec{y} \geq \vec{x}$ . Тогда  $F(\vec{y}) \geq F(\vec{x}) \geq z$ , откуда по определению  $\vec{y}$  входит в  $W(z)$ . Значит,  $W(z)$  является  $\vartheta$ -множеством. Кроме того, если хотя бы одна координата  $x_r$  точки  $\vec{x}$  неположительна, то  $F(\vec{x}) = 0$ , поэтому,  $W(z) \subseteq \vartheta\{\vec{0}\}$  и является  $\vartheta$ -конусом. ■

**Утверждение 2.** Если  $z_1 < z_2$ , то найдется такое  $\delta > 0$ , что  $W(z_2)$  является  $\delta$ -вписанным в  $W(z_1)$ .

**Доказательство.** Отметим следующее неравенство

$$\left| \prod_{r=1}^d \eta(x_r) - \prod_{r=1}^d \eta(y_r) \right| \leq \sum_{r=1}^d |x_r - y_r|, \quad (4)$$

справедливое для любых  $\vec{x}$  и  $\vec{y}$ . Выберем любую точку  $\vec{x}$  из  $W(z_2)$  и рассмотрим кубик со стороной  $(z_2 - z_1)/2d$  и центром в этой точке. Если  $\vec{y}$  — любая точка из этого кубика, то  $|x_r - y_r| \leq (z_2 - z_1)/2d$  и, следовательно

$$\sum_{r=1}^d |x_r - y_r| \leq (z_2 - z_1)/2.$$

Из этой оценки и неравенства (4) получаем  $|F(\vec{x}) - F(\vec{y})| \leq (z_2 - z_1)/2$ . Пусть  $\vec{y}$  не входит в  $W(z_1)$ . Тогда  $F(\vec{y}) < z_1$ . Но раз  $\vec{x}$  входит в  $W(z_2)$ , то  $F(\vec{x}) \geq z_2$ , откуда  $|F(\vec{x}) - F(\vec{y})| \geq z_2 - z_1$ , что невозможно. Поэтому, всякая точка  $\vec{y}$  кубика входит в  $W(z_1)$ . Но в этот кубик полностью входит окрестность точки  $\vec{x}$  радиусом  $(z_2 - z_1)/4$ . Поэтому, конус  $W(z_2)$  вписан в  $W(z_1)$  с  $\delta = (z_2 - z_1)/4$ , что и требовалось показать.

Нам осталось только обосновать неравенство (4), что сделаем индукцией по размерности  $d$ . В одномерном случае  $|\eta(x) - \eta(y)| \leq |x - y|$ . Пусть

неравенство доказано для всех размерностей по  $d-1$  включительно. Шаг индукции выполним, воспользовавшись тривиальным неравенством: если  $0 \leq x, y \leq 1$ , то  $xy \leq x + y$ . Тогда для  $0 \leq a, b, c, d \leq 1$  имеет место

$$|ab - cd| \leq |a + b - c - d| \leq |a - c| + |b - d|.$$

Значит,

$$\begin{aligned} \left| \eta(x_d) \prod_{r=1}^{d-1} \eta(x_r) - \eta(y_d) \prod_{r=1}^{d-1} \eta(y_r) \right| \leq \\ \left| \prod_{r=1}^{d-1} \eta(x_r) - \prod_{r=1}^{d-1} \eta(y_r) \right| + |\eta(x_d) - \eta(y_d)| \leq \sum_{r=1}^d |x_r - y_r|, \end{aligned}$$

что и требовалось доказать. ■

Приступим к построению функций  $h_m$ . Для заданного  $m$  рассмотрим множества  $W[2m/2M]$  и  $W[(2m+1)/2M]$  и, применив лемму 2, получим конечно порожденный  $\vartheta$ -конус  $Y$ , заключенный между ними. По утверждению 2,  $W[(2m+2)/2M]$  будет  $\delta$ -вписаным в  $W[(2m+1)/2M]$ , а значит, и подаловно в  $Y$ . Применим к  $Y$  и  $W[(2m+2)/2M]$  лемму 3, а полученную функцию обозначим  $h_m(\vec{x})$ . Покажем, что она действительно удовлетворяет всем требуемым условиям.

В первую очередь, по построению, для всех  $\vec{x}$  из  $\mathbb{R}^d$  имеет место  $0 \leq h_m(\vec{x}) \leq 1$ . Если  $F(\vec{x}) < m/M$ , то  $\vec{x} \notin W(m/M)$ , и подаловно,  $\vec{x} \notin Y$ , откуда  $h_m(\vec{x}) = 0$ . Если же  $F(\vec{x}) > (m+1)/M$ , то  $\vec{x} \in W[(m+1)/M]$ , откуда  $h_m(\vec{x}) = 1$ . Все условия выполнены.

Мы видим, что система функций  $h_m$  существует, а значит, теорема доказана.

# Семантические сети

Способ представления знаний с помощью сетевых моделей наиболее близок к тому, как они представлены в текстах на естественном языке. В его основе лежит идея о том, что вся необходимая информация может быть описана как совокупность троек  $(arb)$ , где  $a$  и  $b$  — объекты или понятия, а  $r$  — бинарное отношение между ними. Формально сетевые модели представления знаний могут быть заданы в виде  $H = \langle I, C_1, \dots, C_n, \Gamma \rangle$ , где  $I$  — множество информационных единиц,  $C_1, \dots, C_n$  — множество типов связей между элементами  $I$ , отображение  $\Gamma$  задает между информационными единицами, входящими в  $I$ , связи из заданного набора типов связей  $\{C_i\}$ .

В зависимости от типов связей  $\{C_i\}$  различают:

1. *Классифицирующие сети* — в них используются отношения структуризации, они позволяют вводить в базы знаний различные иерархические отношения между элементами множества  $I$ .
2. *Функциональные сети* — вычислительные модели, характеризующиеся наличием функциональных отношений, они позволяют описывать процедуры вычислений одних информационных единиц через другие.
3. *Сценарии* — в них используются каузальные отношения (причинно-следственные или устанавливающие влияние одних явлений или фактов на другие), а также отношения типов «средство — результат», «орудие — действие» и т. д.

Если в сетевой модели допускаются связи различного типа, то ее называют семантической сетью.

Термин «семантическая» означает «смысловая», а сама *семантика* — это наука, устанавливающая отношения между символами и объектами, которые они обозначают, то есть наука, определяющая смысл знаков.

*Семантическая сеть* — это модель, основой для которой является формализация знаний в виде ориентированного графа с размеченными вершинами и дугами. Вершинам соответствуют объекты, понятия или ситуации,

а дугам — отношения между ними. Это наиболее общая модель представления знаний, так как в ней имеются средства реализации всех характерных для знаний свойств: внутренней интерпретации, структурированности, семантической метрики и активности.

Достоинства сетевых моделей:

1. большие выразительные возможности;
2. наглядность системы знаний, представленной графически;
3. близость структуры сети, представляющей систему знаний, семантической структуре фраз на естественном языке;
4. соответствие современным представлениям об организации долговременной памяти человека.

Недостатки сетевых моделей:

1. сетевая модель не дает (точнее, не содержит) ясного представления о структуре предметной области, которая ей соответствует, поэтому формирование и модификация такой модели затруднительны;
2. сетевые модели представляют собой пассивные структуры, для обработки которых необходим специальный аппарат формального вывода и планирования.

Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, соответствующего некоторой подсети поставленной задачи. Это, в свою очередь, обуславливает еще один недостаток модели — сложность поиска вывода на семантических сетях.

Еще раз подчеркнем, что сетевые модели являются очень наглядным и достаточно универсальным средством представления знаний. Однако их формализация в конкретных моделях представления, использования и модификации знаний оказывается достаточно трудоемкой, особенно при наличии множественных отношений между ее элементами.

## 1 Сети автоматов

Будем называть  $(L_1, L_2)$ -трансдюсером устройство, которое принимает слова из языка  $L_1$  и производит слова из языка  $L_2$ . Таким устройством может быть конечный трансдюсер, арочный трансдюсер и другие автоматы. Пусть  $A_1$  является  $(L_1, L_2)$ -трансдюсером, а  $A_2$  –  $(L_2, L_3)$ -трансдюсером. Сеть из автоматов  $A_1$  и  $A_2$  обозначается как  $A_1 \rightarrow A_2$ . Это устройство, работающее следующим образом. Входное слово из языка  $L_1$  посимвольно подается автомату  $A_1$ . Автомат  $A_1$  производит слово из языка  $L_2$  посимвольно. Как только автомат  $A_1$  выдает символ, этот символ подается на вход автомату  $A_2$ . В свою очередь,  $A_2$  производит посимвольно слово из  $L_3$ , которое и является выходом всей сети. Легко видеть, что такая сеть является  $(L_1, L_3)$ -трансдюсером. Соответственно, если теперь  $A_3$  является  $(L_3, L_4)$ -трансдюсером, то может быть построена сеть  $(A_1 \rightarrow A_2) \rightarrow A_3$ .

Типичным примером сети автоматов является front-end компилятор. Традиционно он состоит из двух частей. Первая часть – лексический анализатор, который обычно представлен конечным трансдюсером. Вторая часть – синтаксический анализатор, обычно являющийся детерминированным автоматом с магазинной памятью. Лексический анализатор принимает код программы и производит поток токенов. Соответственно, лексический анализатор является  $(L_1, L_2)$ -трансдюсером, где  $L_1$  – язык программирования, а  $L_2$  – промежуточный язык токенов. Далее, синтаксический анализатор выполняет анализ слова на  $L_2$  и производит слово на промежуточном языке компилятора, который затем передается back-end компилятору.

Конечным трансдюсером будем называть детерминированный конечный автомат-трансдюсер. В случае сети из конечного трансдюсера и детерминированного автомата с магазинной памятью, сеть из этих автоматов не расширяет вычислительные возможности ДАМПа. Это отражено в следующем утверждении.

**Предложение 1.1.** *Если некоторый язык  $L$  распознается сетью из конечного трансдюсера и ДАМПа, то существует ДАМП, распознающий язык  $L$ .*

Также может быть сформулировано аналогичное утверждение для конечных трансдюсеров.

**Предложение 1.2.** *Если некоторый язык  $L$  распознается сетью из конечного трансдюсера и детерминированного конечного автомата, то существует детерминированный конечный автомат, распознающий тот же язык.*

Однако в общем случае сеть из автоматов может обладать большими вычислительными возможностями, чем ее составляющие части. В частности, это имеет место быть для модели арочных автоматов.

Модель арочных автоматов (англ. nested automata) была предложена в работах Раджива Алюра. Арочный автомат является автоматом с магазинной памятью, на который наложены дополнительные ограничения. Так,

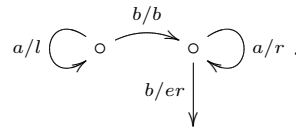
входной алфавит  $\Sigma$  разбит на три непересекающиеся части – множество левых символов  $\Sigma_l$ , множество правых символов  $\Sigma_r$  и множество внутренних символов  $\Sigma_i$ . Работа арочного автомата со стеком полностью определяется типом входного символа. Так, если автомат в состоянии  $q$  принимает левый символ  $a \in \Sigma_l$ , он обязан положить в стек один символ и изменить свое состояние в зависимости от  $q$  и  $a$ . Если автомат в состоянии  $q$  принимает правый символ  $b \in \Sigma_r$ , он обязан извлечь из стека один символ  $\gamma$  и изменить свое состояние в зависимости от  $q$ ,  $b$  и  $\gamma$ . Наконец, если автомат в состоянии  $q$  получает внутренний символ  $c \in \Sigma_i$ , он может лишь изменить свое состояние в зависимости от  $q$  и  $c$ . Иначе говоря, арочный автомат действует как конечный автомат при обработке символа из  $\Sigma_i$ .

Арочные автоматы распознают все регулярные языки и дополнительно распознают некоторые контекстно-свободные языки. Арочные автоматы наследуют многие свойства конечных автоматов. В частности, любой арочный автомат можно детерминировать, а языки арочных автоматов замкнуты относительно объединения, пересечения, итерации, дополнения и конкатенации.

**Предложение 1.3.** *Существует язык такой, что он не распознается никаким арочным автоматом, однако распознается сетью из конечного трансдюсера и арочного автомата.*

*Доказательство.* Рассмотрим язык  $a^n b a^n$ . Очевидно, такой язык не может быть распознан арочным автоматом. Действительно,  $a$  может быть либо левым, либо правым, либо внутренним символом. Если  $a$  является внутренним символом, то арочный автомат действует как конечный автомат, и следовательно, не может распознать такой язык. Если  $a$  является правым символом, то он постоянно извлекает из пустого стека, а значит, тоже действует как конечный автомат. Аналогично, если  $a$  – левый символ, то автомат постоянно складывает в стек символы  $a$  и никогда их не извлекает.

Для того, чтобы арочный автомат распознал язык  $a^n b a^n$  необходимо сложить в стек символы  $a$ , которые предшествуют  $b$ , затем вытащить их из стека и убедиться, что количество символов в стеке совпадает с количеством символов  $a$  после  $b$ . Для этого необходимо преобразовать входную строку. Рассмотрим следующий конечный трансдюсер.



Этот трансдюсер переводит слова из языка  $a^n b a^m$  в язык  $l^n b r^m$ . Теперь легко построить арочный автомат, который бы распознавал подмножество языка  $l^n b r^m$ , равное  $l^n b r^n$ . Для этого  $l$  следует объявить левым символом,  $r$  – правым символом и затем выполнить операции складывания в стек и подсчета символов в стеке. Таким образом, мы построили сеть из конечного

трансдюсера и арочного автомата, которая обладает большими вычислительными способностями, чем один арочный автомат.  $\square$

Однако, расширение арочного автомата подобным «конечным препроцессором» оказывается менее мощно, чем детерминированный автомат с магазинной памятью. Для доказательства этого утверждения нам понадобится операция смешения языков. Смешением языков  $L_1$  и  $L_2$  называется язык  $L_1 \circ L_2$ , определяемый следующим образом:  $\{u_1 v_1 u_2 v_2 \dots v_n u_{n+1}\}$ , где  $u_1 u_2 \dots u_{n+1} \in L_1$  и  $v_1 v_2 \dots v_n \in L_2$ .

**Предложение 1.4.** *Существует язык  $L$ , распознаваемый детерминированным автоматом с магазинной памятью, не распознаваемый сетью из конечного трансдюсера и арочного автомата.*

*Доказательство.* Построим такой язык. Пусть  $L_1$  – язык, порожденный следующей грамматикой.

$$S \rightarrow aSc \mid bSd \mid SS \mid \lambda$$

Пусть  $L$  является смешением языков  $L_1$  и  $\{c, d\}^*$ . Докажем, что такой  $L$  соответствует условиям предложения.

Язык  $L$  состоит из слов, в которых символы  $a, b$  «закрыты» символами  $c, d$  в правильных скобочных последовательностях, и, возможно, присутствуют другие символы  $c, d$ . Детерминированный автомат с магазинной памятью работает следующим образом. Все символы  $a, b$  он складывает в стек. Далее, если на вход поступает символ  $c$  и наверху стека лежит символ  $a$ , он снимает символ  $a$  со стека. Аналогичные действия выполняются для пары символов  $b, d$ . Если же на вход поступает символ  $c$  и наверху стека лежит не символ  $a$ , автомат переходит к следующему символу. Аналогично он действует при символе  $d$  на входе и любом символе, кроме  $b$  наверху стека. Очевидно, такой ДАМП распознает язык  $L$ .

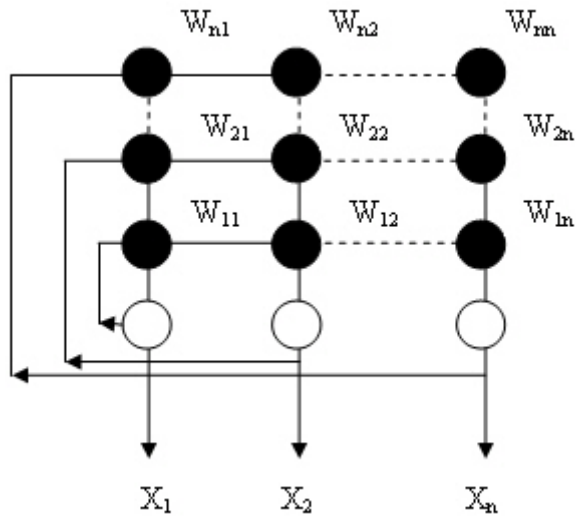
Покажем, что арочный автомат неспособен распознать язык  $L$ . В самом деле, поскольку  $L_1$  является подмножеством  $L$ , арочный автомат должен уметь распознавать скобочные последовательности. Это значит, что символы  $a$  и  $b$  должны быть левыми символами, а символы  $c$  и  $d$  – правыми. Но это означает, что арочный автомат должен снимать символ со стека при появлении  $c$  и  $d$  вне зависимости от того, что находится наверху стека. В результате, информация о том символе, который должен быть закрыт, теряется. Так, для строк  $adc$  и  $add$  невозможно определить, является ли третий символ парным для первого.

$\square$



# Полносвязные сети Хопфилда

Сеть *Хопфилда* — однослойная сеть. Все нейроны связаны друг с другом связями  $w_{ij}$ , причем сигнал с выхода нейрона может подаваться на его же вход и необязательно  $w_{ij} = w_{ji}$ .



Поскольку сигнал с выхода каждого нейрона подается на входы всех остальных, входной вектор начинает циркулировать, преобразуясь по сети до тех пор, пока сеть не придет в устойчивое состояние (то есть когда все нейроны на каждом последующем цикле будут вырабатывать тот же сигнал, что и на предыдущем). Очевидно, возможны случаи бесконечной циркуляции входного вектора без достижения устойчивого состояния.

Выберем функцию элементов в виде:

$$x_j(t+1) = \begin{cases} 1, & \sum_{i=1}^n w_{ij} x_i(t) > \theta_j, \\ 0(-1), & \sum_{i=1}^n w_{ij} x_i(t) < \theta_j, \\ x_j(t), & \sum_{i=1}^n w_{ij} x_i(t) = \theta_j, \end{cases}$$

*Состояние сети* — множество текущих значений сигналов  $x$  от всех нейронов. Функционирование сети геометрически может быть представлено как движение

вектора  $x$ , характеризующего состояние сети, на кубе  $[0,1]^n$ . Когда подается новый входной вектор, сеть переходит из вершины в вершину, пока не стабилизируется. Устойчивая вершина определяется сетевыми весами, текущими входами и величиной порога. Если входной вектор частично неправилен или неполон, то сеть стабилизируется в вершине, ближайшей к желаемой.

В общем случае все возможные состояния сети образуют некое подобие холмистой поверхности, а текущие состояния сети аналогичны положениям тяжелого шарика, пущенного на эту поверхность, — он движется вниз по склону в ближайший локальный минимум. Каждая точка поверхности соответствует некоторому сочетанию активностей нейронов в сети, а высота подъема поверхности в данной точке характеризует «энергию» этого состояния. Энергия данного сочетания активностей определяется как сумма весов связей между парами активных нейронов, взятая со знаком минус (при  $0 = 0$ ).

Таким образом, если связь между двумя какими-то нейронами имеет большой положительный вес, то сочетания, в которых эти нейроны активны, характеризуются низким уровнем энергии — именно к таким сочетаниям и будет стремиться вся сеть. И, наоборот, нейроны с отрицательной связью при активации добавляют к энергии сети большую величину, так что сеть стремится избегать подобных состояний.

Динамику сети Хопфилда удобно описывать так называемой функцией энергии, которая в достаточно общем виде может быть определена как

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j + \sum_{j=1}^n \theta_j x_j.$$

Функция энергии определяет устойчивость сети, другими словами — это функция Ляпунова сети Хопфилда, то есть функция, которая всегда убывает при изменении состояния сети. В конце концов эта функция должна достичь минимума и прекратить изменение, гарантируя тем самым устойчивость сети.

Изменение состояния какого-либо элемента сети всегда вызывает уменьшение энергии сети. Действительно, пусть изменил свое состояние элемент  $k$  ( $k = 1 : n$ ), то есть его состояние изменилось с  $+1$  на  $0$  (или  $-1$ ) или наоборот, тогда

**Ошибка!**

$$\begin{aligned}
\Delta E &= -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i(t+1) x_j(t+1) + \sum_{j=1}^n \theta_j x_j(t+1) + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i(t) x_j(t) - \sum_{j=1}^n \theta_j x_j(t) \\
\Rightarrow \Delta E_k &= -x_k(t+1) \sum_{i=1}^n w_{ik} x_i(t+1) + \theta_k x_k(t+1) + x_k(t) \sum_{i=1}^n w_{ik} x_i(t) - \theta_k x_k(t) \Rightarrow \\
\Rightarrow \Delta E_k &= (x_k(t) - x_k(t+1)) \left( \sum_{i=1}^n w_{ik} x_i(t+1) - \theta_k \right) \quad \{x_k(t+1) \equiv x_i(t)\}
\end{aligned}$$

Видно, что в результате изменения  $k$ -го элемента  $\Delta E_k \leq 0$ ,  $\Delta E_k = 0$ , когда в сети не происходит никаких изменений. Благодаря такому непрерывному стремлению к уменьшению энергии, в конце концов, должна достигнуть минимума и прекратить изменение. По определению такая сеть является устойчивой. Сети Хопфилда называются также сетями, минимизирующими свою энергию. Сети Хопфилда имеют многочисленные применения. Некоторые из них связаны со способностью этих сетей запоминать, а затем восстанавливать даже по неполной входной информации различные образы. Другие применения связаны с возможностью использования сетей Хопфилда для решения оптимизационных задач.

# Двунаправленная ассоциативная память

Основной причиной неудач исследователей в области искусственного интеллекта, потративших свыше 20 лет на безуспешные попытки моделирования интеллектуальной деятельности на базе обычных цифровых ЭВМ, является, по-видимому, тот факт, что в современных ЭВМ существует прямая зависимость времени поиска от количества хранимых образцов. Компьютер запоминает отдельные объекты в отдельных ячейках, как бы заучивает сведения наизусть, и при изучении наук примеры для него, в отличие от человека, отнюдь не полезнее правил. НейроЭВМ, построенная на базе нейросетей, обладает ассоциативной памятью и классифицирует поступившие образы со скоростью, которая не зависит от количества уже поступивших образцов, — она немедленно связывает новый образ с ближайшим имеющимся.

Память человека является ассоциативной — один предмет напоминает нам о другом, а тот, в свою очередь о третьем и т. д. Наши мысли как бы перемещаются от предмета к предмету по цепочке умственных ассоциаций. Способность к ассоциациям может быть использована для восстановления забытых образов («мы с вами где-то встречались»). Двунаправленная ассоциативная память является гетероассоциативной; входной вектор поступает на один набор нейронов, а соответствующий выходной вектор вырабатывается на другом наборе нейронов. Как и сеть Хопфилда, двунаправленная ассоциативная память способна к обобщению, вырабатывая правильные реакции, несмотря на возможные искажения входа.

Очевидно, состояние нейронов можно рассматривать, как кратковременную память, так как она может быстро изменяться при появлении другого входного вектора. В то же время значения коэффициентов весовой матрицы образуют долговременную память (ассоциации) и могут изменяться на более длительном отрезке времени, используя соответствующий метод обучения. Обучение производится с использованием обучающего набора из пар векторов  $x$  и  $y$ . Предположим, что все запомненные образцы представляют собой двоичные векторы.

Решение задачи с помощью двунаправленной ассоциативной памяти можно разбить на два этапа: режим обучения и непосредственно решение (распознавание). Рассмотрим оба эти этапа на примере.

Каждый нейрон  $a_i$  в первом слое  $A$  имеет синапсы, соединяющие его с нейронами  $b_j$  во втором слое  $B$ . Пусть нейроны имеют следующий «смысл»:  $a_1$  — валюта,  $a_2$  — доллары,  $a_3$  — марки,  $a_4$  — рубли,  $b_1$  — США,  $b_2$  — Россия,  $b_3$  — Канада,  $b_4$  — Германия.

## Режим обучения бинарными образами

Подадим на нейросеть три бинарных связи  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ . Пусть

$$\begin{aligned} x_1 &= (1, 1, 0, 0) \rightarrow y_1 = (1, 1, 1, 0); \\ x_2 &= (1, 0, 1, 0) \rightarrow y_2 = (0, 1, 0, 1); \\ x_3 &= (1, 0, 0, 1) \rightarrow y_3 = (0, 1, 0, 0); \end{aligned}$$

Смысл обучающих связей очевиден: если возбуждены нейроны  $a_1$  и  $a_2$  (в нашем распоряжении есть доллары), то по соответствующим синапсам возбудятся нейроны  $b_1$ ,  $b_2$ ,  $b_3$  (мы можем ими воспользоваться в США, России и Канаде), и т. д. От бинарных связей перейдем к биполярным (это сделано исключительно для простоты, чтобы не нужно было вводить ненулевой порог срабатывания нейронов):

$$\begin{aligned} x_1 &= (1, 1, -1, -1) \rightarrow y_1 = (1, 1, -1, -1); \\ x_2 &= (1, -1, 1, -1) \rightarrow y_2 = (1, -1, 1, -1); \\ x_3 &= (1, -1, -1, 1) \rightarrow y_3 = (1, -1, -1, 1); \end{aligned}$$

Составим матрицу весов:

$$W = \sum_{i=1}^3 x_i^T y_i = \begin{pmatrix} -1 & 3 & -1 & -1 \\ 3 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \\ -1 & -1 & -1 & -1 \end{pmatrix}.$$

## Режим распознавания

Оценим эффективность запоминания обучающих связей. Убедимся, что матрица  $W$  хранит связи  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ . Подадим на вход  $x_1$  тогда  $x_1 = (2, 2, 2, -2)$  — это означает, что в слое  $B$  возбудятся первые три нейрона (порог срабатывания принят равным нулю). Тогда в бинарной форме  $y = (1, 1, 1, 0)$ , что является требуемой ассоциацией. Это означает, что подача на вход  $x_1$ , приводит к  $y_1$ , то есть ЭВМ действительно «запомнила» связь  $(x_1, y_1)$ . Аналогично проверяется запоминание остальных связей.

Сеть является двунаправленной:  $y_1 W^T = (1, 5, -3, -3) \rightarrow (1, 1, 0, 0) \rightarrow x_1$ , и т. д.

Определим энергию связей в памяти:

$$E(x_1, y_1) = -x_1 W y_1^T = -6,$$

аналогично  $E(x_2, y_2) = -4$  и  $E(x_3, y_3) = -2$ . Следует ожидать, что при ошибке в исходной информации связь  $(x_1, y_1)$  будет притягивать к себе больше образов, так как это точка устойчивого равновесия с минимальным энергетическим уровнем. Действительно, подадим на вход образ  $x' = (1, 1, 0, 1)$  — искаженный на один бит  $x_1$  и  $x_3$  тогда  $x' W = (1, 1, 1, -3) \rightarrow (1, 1, 1, 0) \rightarrow y_1$ . Аналогично, если взять  $x'' = (1, 0, 1, 1)$  — вектор, расположенный «между»  $x_2$  и  $x_3$ , то получим  $(-3, 1, -3, 1) \rightarrow (0, 1, 0, 1) \rightarrow y_2$  — связь  $(x_2, y_2)$ , притягивает к себе, так как ее энергия меньше энергии  $(x_3, y_3)$ .

Работа с неопределенными данными. Рассмотрим случай, когда тип валюты неопределен  $x' = (1, 0, 0, 0)$ , тогда  $x' W = (-1, 3, -1, -1) \rightarrow (0, 1, 0, 0) \rightarrow y_3$ . Это означает, что она может быть использована только в той стране, где в ходу любая валюта.

Если валюта может быть любой, например, доллары и марки, то она может использоваться везде:

$$x' = (1, 1, 1, 0) \rightarrow x' M = (1, 1, 1, 1) \rightarrow y'.$$

Проведенное исследование показывает, что построенная нейросеть способна запомнить необходимую информацию на этапе обучения, а в рабочем режиме позволяет решать задачи распознавания, то есть реализует функции ассоциативной памяти. Вся полученная при обучении информация сосредоточена в матрице  $W$ . За счет параллельной структуры сеть решает задачу «мгновенно» — за одно действие — умножение входного вектора на

матрицу памяти. Так как информация как бы интегрирована в матрицу  $W$ , сеть способна достаточно эффективно решать задачу и при частичных искажениях в исходных данных.



## Сети Хемминга

Попов В.Ю.

## 1 Введение

Сеть Хемминга (в дальнейшем стандартная сеть Хемминга) была предложена Р.Липпманом в работе [1]. Она строится по фиксированному набору векторов и предназначена для нахождения по подаваемому на вход вектору ближайшего по расстоянию Хемминга вектора из исходного набора.

Цель данной заметки — изучение классов нейронных сетей, позволяющих находить по подаваемому на вход вектору ближайший по расстоянию Хемминга вектор из исходного набора более эффективно, чем стандартная сеть Хемминга.

## 2 Основные определения

В соответствии с принципами функционирования биологических нейронов созданы различные математические модели, которыми в большей или меньшей степени реализуются свойства природной нервной клетки. Обобщенная схема, составляющая основу большинства таких моделей, восходит к модели МакКаллока – Питса [2]. Согласно этой модели (см. рис. 1) нейрон состоит из входов  $x_0, x_1, \dots, x_n$ , выходов  $y_1, y_2, \dots, y_m$  и двух соединенных последовательно преобразователей: сумматора  $+$  и нелинейного преобразователя  $F$ . Вход  $x_0$  называется *поляризатором* и предполагается постоянным. Его значение всегда равно 1. Входы  $x_1, \dots, x_n$  являются переменными. При переходе от входа  $x_i$ ,  $0 \leq i \leq n$ , к сумматору сигнал умножается на число  $w_i$ , называемое *весом*. Сумматор вычисляет сумму полученных сигналов  $\sum_{i=0}^n w_i x_i$ , значение которого передается на нелинейный преобразователь. В качестве значения нелинейного преобразователя можно рассматривать произвольную функцию  $F(x_1, \dots, x_n)$ . Все выходы полагаются одинаковыми. Их значения равны  $F(x_1, \dots, x_n)$ .

Выходы одного нейрона могут быть входами других нейронов, образуя пару выход – вход. Это позволяет создавать из нейронов вычислительные структуры, называемые *нейронными сетями*. В зависимости от способа объединения нейронов нейронные сети могут быть *однонаправленными* или *рекуррентными*. В однонаправленной сети нейроны объединены в слои

$S_1, S_2, \dots, S_k$ . При этом сигнал от переменных входов передается только в нейроны слоя  $S_1$ , к выходам сигнал поступает только от нейронов слоя  $S_k$ . Все выходы слоя  $S_i$  являются входами слоя  $S_{i+1}$ , а все переменные входы слоя  $S_{i+1}$  являются выходами слоя  $S_i$ , т.е. все выходы слоя  $S_i$  и все переменные входы слоя  $S_{i+1}$  замкнуты в пару выход – вход. Нейронные сети, полученные из однонаправленных добавлением обратных связей между различными слоями нейронов или связей внутри одного слоя, называются рекуррентными.

Сети, у которых множество входов совпадает с множеством выходов, называются *автоассоциативными*. Сети, у которых множество входов не пересекается с множеством выходов, называются *гетероассоциативными*.

Нейрон называется *линейным*, если

$$F(x_1, \dots, x_n) = \sum_{i=0}^n w_i x_i.$$

Нейрон называется *полулинейным*, если для некоторого действительного числа  $a$  функция  $F(x_1, \dots, x_n)$  имеет вид

$$F(x_1, \dots, x_n) = \begin{cases} \sum_{i=0}^n w_i x_i, & \sum_{i=0}^n w_i x_i > a \\ 0, & \sum_{i=0}^n w_i x_i \leq a \end{cases}$$

Расстоянием Хемминга  $D_H$  между векторами

$$(x_1, x_2, \dots, x_n), (y_1, y_2, \dots, y_n) \in \mathbb{Z}_2^n$$

называется величина

$$\sum_{i=1}^n (x_i \oplus y_i),$$

где  $\sum$  — обычное суммирование, а  $\oplus$  — сложение по модулю 2.

Вектор

$$(y_1, y_2, \dots, y_n)$$

называется *биполярным представлением* вектора

$$(x_1, x_2, \dots, x_n),$$

если  $y_i = 1$  при  $x_i = 1$  и  $y_i = -1$  при  $x_i = 0$ . Биполярное представление вектора

$$(x_1, x_2, \dots, x_n)$$

будем обозначать

$$(x_1^*, x_2^*, \dots, x_n^*).$$

### 3 Стандартная сеть Хемминга

Стандартная сеть Хемминга — это трехслойная рекуррентная структура (см. рис. 2). Она позиционируется как специализированное гетероассоциативное запоминающее устройство.

Предполагается, что в памяти сети хранится набор векторов

$$(z_{1,1}, z_{1,2}, \dots, z_{1,n}), \dots, (z_{p,1}, z_{p,2}, \dots, z_{p,n}) \in \mathbb{Z}_2^n.$$

На вход подается биполярное представление вектора

$$(x_1, x_2, \dots, x_n) \in \mathbb{Z}_2^n.$$

Вектор на выходе

$$(y_1, y_2, \dots, y_n)$$

является биполярным представлением вектора

$$(z_{j,1}, z_{j,2}, \dots, z_{j,n})$$

такого, что

$$D_H((x_1, x_2, \dots, x_n), (z_{j,1}, z_{j,2}, \dots, z_{j,n})) = \min_{1 \leq i \leq p} D_H((x_1, x_2, \dots, x_n), (z_{i,1}, z_{i,2}, \dots, z_{i,n})).$$

Веса всех нейронов в стандартной сети Хемминга постоянны. Первый слой состоит из нейронов  $H_{1,1}, \dots, H_{1,p}$ , второй — из  $H_{2,1}, \dots, H_{2,p}$ , а третий — из  $H_{3,1}, \dots, H_{3,n}$ . Первый слой со вторым и третий слой с выходами соединены по принципу “свой со своим”, а входы с первым слоем и второй слой с третьим соединены по принципу “каждый с каждым”. Кроме того, по принципу “каждый с каждым” обратными связями соединены между собой все нейроны второго слоя. Нейроны первого и третьего слоев являются линейными, а второго — полулинейными с нулевыми поляризаторами и  $a = 0$ .

Первый слой конструируется таким образом, что на выходе нейрона  $H_{1,i}$  появляется величина

$$1 - \frac{1}{p} D_H((x_1, x_2, \dots, x_n), (z_{i,1}, z_{i,2}, \dots, z_{i,n})).$$

Получив входной вектор, первый слой срабатывает один раз, иницируя второй слой, и удаляется. После этого начинает работу второй слой. Второй слой срабатывает по рекуррентному принципу WTA, при котором определяется нейрон-победитель (единственный нейрон, значение которого отлично от нуля) и только значение нейрона-победителя передается в третий слой. До выявления победителя сигнал передается только в обратные связи. Вес обратной связи нейрона с самим собой считается равным 1, а веса обратных

связей нейронов с другими нейронами попарно различны и определяются по формуле

$$-\frac{1}{p-1} + \xi,$$

где  $\xi$  — случайная величина с достаточно малой амплитудой.

## 4 Сеть Хемминга $NH_1$

Сеть Хемминга  $NH_1$  (см. рис. 3) является двухслойной рекуррентной структурой. Первый слой состоит из полулинейных нейронов  $M_{1,1}, \dots, M_{1,p}$  при  $a = 0$ , а второй — из линейных  $M_{2,1}, \dots, M_{2,n}$ . Вес поляризатора нейрона  $M_{i,j}$  обозначим через  $w_{0,i,j}$ . Вес нейронной связи входа  $x_l$  и нейрона  $M_{1,j}$  обозначим через  $w_{l,1,j}$ , а вес нейронной связи нейрона  $M_{1,i}$  и нейрона  $M_{2,j}$  обозначим через  $w_{i,2,j}$ . Вес обратной связи нейрона  $M_{1,j}$  мы будем обозначать  $v_j$ .

Предполагается, что в памяти сети хранится набор векторов

$$(z_{1,1}, z_{1,2}, \dots, z_{1,n}), \dots, (z_{p,1}, z_{p,2}, \dots, z_{p,n}) \in \mathbb{Z}_2^n.$$

На вход подается биполярное представление вектора

$$(x_1, x_2, \dots, x_n) \in \mathbb{Z}_2^n.$$

Для любого  $i$  будем полагать, что

$$w_{l,1,i} = \frac{p+1}{2} z_{i,l}^*$$

при  $n+1 > l > 0$ ,

$$w_{n+1,1,i} = \frac{n(p+1)}{2} + p + 2 - i.$$

и  $w_{0,1,i} = -1$ . Все веса  $w_{0,2,i}$  будем считать равными нулю, а  $w_{i,2,j} = z_{i,j}$ . Первый слой срабатывает по рекуррентному принципу WTA. Веса  $v_j$  полагаются равными  $-1$ .

**Теорема 1.** По вектору

$$(x_1^*, x_2^*, \dots, x_n^*, 1),$$

где

$$(x_1, x_2, \dots, x_n) \in \mathbb{Z}_2^n,$$

нейронная сеть  $NH_1$  выдает вектор

$$(z_{j,1}, z_{j,2}, \dots, z_{j,n})$$

такой, что

$$D_H((x_1, x_2, \dots, x_n), (z_{j,1}, z_{j,2}, \dots, z_{j,n})) = \min_{1 \leq i \leq p} D_H((x_1, x_2, \dots, x_n), (z_{i,1}, z_{i,2}, \dots, z_{i,n})).$$

**Доказательство теоремы 1.** Рассмотрим вектор

$$(x_1, x_2, \dots, x_n) \in \mathbb{Z}_2^n.$$

В этом случае на вход  $NH_1$  подается вектор

$$(x_1^*, x_2^*, \dots, x_n^*, 1).$$

Поэтому результатом суммирования для  $i$ -го нейрона будет величина

$$\frac{n(p+1)}{2} + p + 1 - i + \sum_{l=1}^n \frac{p+1}{2} z_{i,l}^* x_l^*.$$

Обозначим через  $s_i$  количество совпадающих координат у векторов

$$(x_1^*, x_2^*, \dots, x_n^*)$$

и

$$(z_{i,1}^*, z_{i,2}^*, \dots, z_{i,n}^*),$$

а через  $t_i$  — количество несовпадающих. Очевидно, что если  $z_{i,l}^* = x_l^*$ , то  $z_{i,l}^* x_l^* = 1$ , а если  $z_{i,l}^* \neq x_l^*$ , то  $z_{i,l}^* x_l^* = -1$ . Следовательно,

$$\begin{aligned} \frac{n(p+1)}{2} + p + 1 - i + \sum_{l=1}^n \frac{p+1}{2} z_{i,l}^* x_l^* &= \\ \frac{n(p+1)}{2} + p + 1 - i + \frac{p+1}{2} \sum_{l=1}^n z_{i,l}^* x_l^* &= \\ \frac{n(p+1)}{2} + p + 1 - i + \frac{p+1}{2} (s_i - t_i). \end{aligned}$$

Так как  $s_i + t_i = n$ , имеем

$$\begin{aligned} \frac{n(p+1)}{2} + p + 1 - i + \frac{p+1}{2} (s_i - t_i) &= \\ \frac{(s_i + t_i)(p+1)}{2} + p + 1 - i + \frac{p+1}{2} (s_i - t_i) &= \\ \frac{p+1}{2} (s_i + t_i + s_i - t_i) + p + 1 - i &= (p+1)s_i + p + 1 - i. \end{aligned}$$

Поскольку число  $(p+1)s_i$  является неотрицательным и  $p+1 > i$  для любого допустимого значения  $i$ , число  $(p+1)s_i + p+1 - i$  является натуральным. Кроме того, из соотношения

$$(p+1)s_i + p+1 - i = p+1 - i \bmod(p+1)$$

получаем, что для различных значений  $i$  числа  $(p+1)s_i + p+1 - i$  различны. Так как для любого  $i$  вес  $v_i$  равен 1, а вес поляризатора равен  $-1$ , через  $r$  тактов рекуррентного функционирования первого слоя на выходе  $i$ -го нейрона будет число  $(p+1)s_i + p+1 - i - r > 0$  или 0, если  $(p+1)s_i + p+1 - i - r \leq 0$ . Заметим, что

$$((p+1)s_i + p+1 - i - r) - ((p+1)s_j + p+1 - j - r) = (p+1)(s_i - s_j) - i + j.$$

Поэтому

$$(p+1)s_i + p+1 - i - r > (p+1)s_j + p+1 - j - r \Leftrightarrow s_i > s_j.$$

Отсюда вытекает, что по завершении рекуррентного процесса мы получим вектор, все координаты которого равны нулю, кроме  $i$ -ой координаты, равной единице. При этом число  $i$  будет номером вектора, имеющего с исходным максимальное количество совпадающих координат, а значит, находящегося на минимальном расстоянии Хемминга. Отсюда в силу соотношения  $w_{i,2,j} = z_{i,j}$  непосредственно вытекает утверждение теоремы. **Теорема 1 доказана.**

Итак, мы установили, что нейронная сеть  $NH_1$  может решать ту же задачу, что и стандартная сеть Хемминга. При этом непосредственно из определения сети  $NH_1$  и стандартной сети Хемминга следует, что стандартная сеть Хемминга содержит  $2p + n$  нейронов и  $2np + p + p^2$  взвешенных нейронных связей, т.е.  $2np + 3p + n + p^2$  элементарных преобразователей, а для сети Хемминга  $NH_1$  имеет место следующее утверждение.

**Теорема 2.** *Сеть Хемминга  $NH_1$  содержит  $p + n$  нейронов и  $2np + 2p$  взвешенных нейронных связей, т.е.  $2np + 3p + n$  элементарных преобразователей.*

Заметим, что при решении практических задач число  $p$ , как правило, значительно больше числа  $n$ . Поэтому уменьшение количества преобразователей с  $O(p^2)$  до  $O(np)$  делает сеть  $NH_1$  существенно более эффективной по сравнению с стандартной сетью Хемминга.

Из доказательства теоремы 1 очевидным образом вытекает

**Теорема 3.** *Сеть Хемминга  $NH_1$  работает за время  $O(np)$ .*

## 5 Сеть Хемминга $NH_2$

Существенным недостатком сети  $NH_1$  является то, что она работает за время  $O(np)$ , т.е. за то же время, что и обычный однопроцессорный алгоритм.

Это можно устранить, модифицировав первый слой сети  $NH_1$  и добавив к нему рекуррентный анализатор (см. рис. 4). К сети  $NH_1$  мы добавляем  $p+1$  нейрон  $N_1, \dots, N_p, K$ . Предполагается, что новые нейроны функционируют в рамках процесса WTA. Нейроны  $N_1, \dots, N_p$  образуют отдельный слой и нейрон  $K$  образует слой из одного нейрона. Нейроны первого слоя сети  $NH_1$  соединены по принципу “свой со своим” с нейронами  $N_1, \dots, N_p$ , каждый из которых соединен с нейроном  $K$ . Нейрон  $K$  соединен с каждым нейроном первого слоя сети  $NH_1$ . Сигнал от нейрона  $M_{1,i}$  к нейрону  $N_i$  передается с весом 1. Поляризаторы всех нейронов  $N_i$  имеют нулевой вес. Нелинейный преобразователь каждого из нейронов  $N_i$  работает на основе функции

$$F(x_1, \dots, x_n) = \begin{cases} 1, & \sum_{i=0}^n w_i x_i > p \\ 0, & \sum_{i=0}^n w_i x_i \leq p \end{cases}$$

Поляризатор нейрона  $K$  имеет вес  $p$ . Для любого  $i$  сигнал от нейрона  $N_i$  к нейрону  $K$  передается с весом  $-1$ . Нелинейный преобразователь нейрона  $K$  работает на основе функции

$$F(x_1, \dots, x_n) = \begin{cases} 1, & \sum_{i=0}^n w_i x_i < p \\ 0, & \sum_{i=0}^n w_i x_i \geq p \end{cases}$$

Сигнал от нейрона  $K$  к каждому из нейронов  $M_{1,i}$  передается с весом  $p$ .

Сеть, полученную из сети  $NH_1$  добавлением нейронов  $N_1, \dots, N_p, K$  указанным выше способом, будем называть  $NH_2$ . Опираясь на конструкцию сети  $NH_2$ , несложно установить следующие утверждения.

**Теорема 4.** По вектору

$$(x_1^*, x_2^*, \dots, x_n^*, 1),$$

где

$$(x_1, x_2, \dots, x_n) \in \mathbb{Z}_2^n,$$

нейронная сеть  $NH_2$  выдает вектор

$$(z_{j,1}, z_{j,2}, \dots, z_{j,n})$$

такой, что

$$D_H((x_1, x_2, \dots, x_n), (z_{j,1}, z_{j,2}, \dots, z_{j,n})) = \min_{1 \leq i \leq p} D_H((x_1, x_2, \dots, x_n), (z_{i,1}, z_{i,2}, \dots, z_{i,n})).$$

**Теорема 5.** Сеть Хемминга  $NH_2$  содержит  $2p+n+1$  нейронов и  $2np+5p$  взвешенных нейронных связей, т.е.  $2np+7p+n+1$  элементарных преобразователей.

**Теорема 6.** Сеть Хемминга  $NH_2$  работает за время  $O(p+n)$ .

## **Литература**

- [1] Lippman R. An introduction to computing with neural nets // IEEE ASSP Magazine. 1987. April. P. 4 – 22.
- [2] McCulloch W. S., Pitts W. H. A logical calculus of ideas immanent in nervous activity // Bull. Math. Biophysics. 1943. Vol. 5. P. 115 – 119.



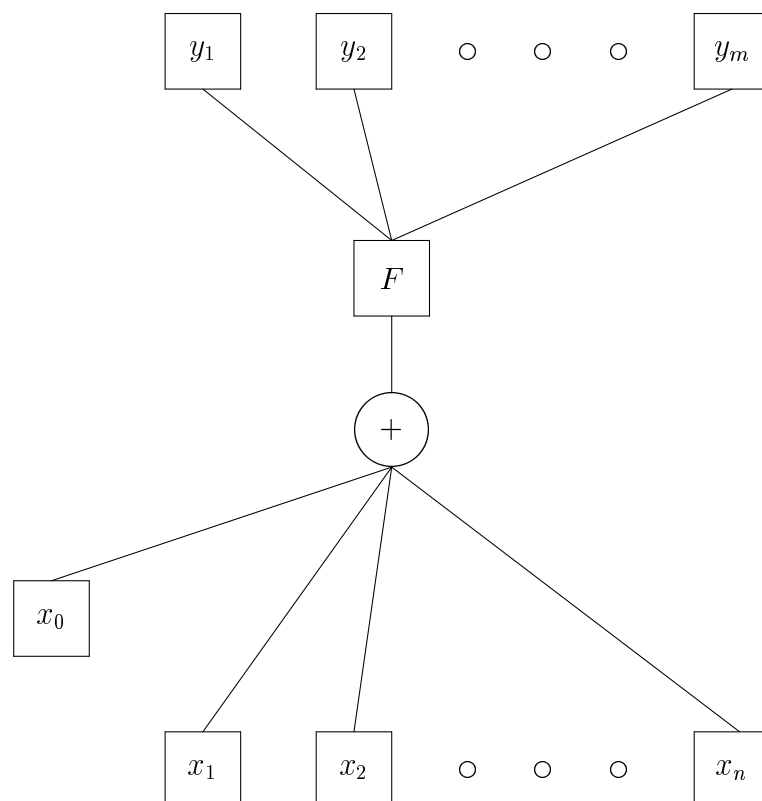


Рис. 1. Модель нервной клетки по МакКаллоку – Питсу. (Здесь и далее предполагается, что если направление распространения сигнала явно не указано, то сигнал от узла к узлу передается по линиям от нижнего конца к верхнему).

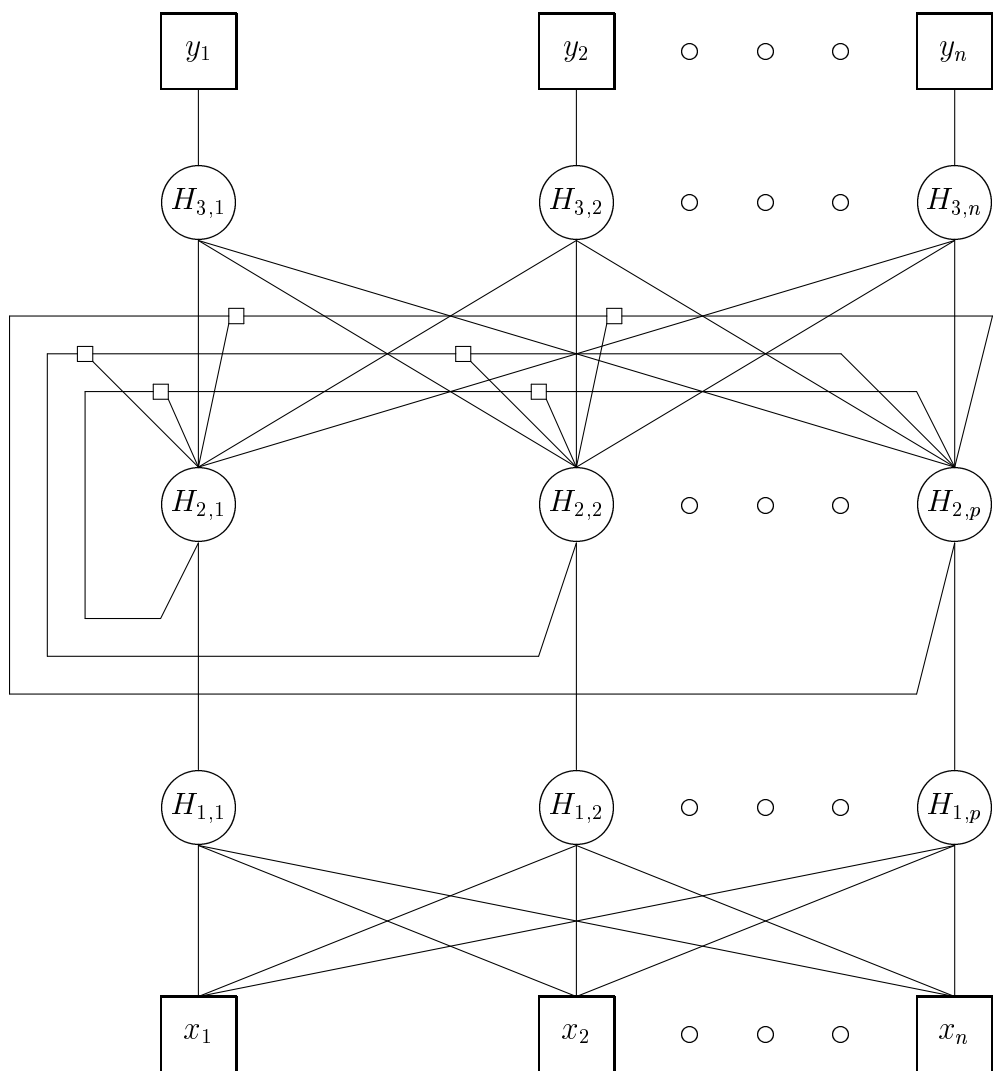


Рис. 2. Структура стандартной сети Хемминга.

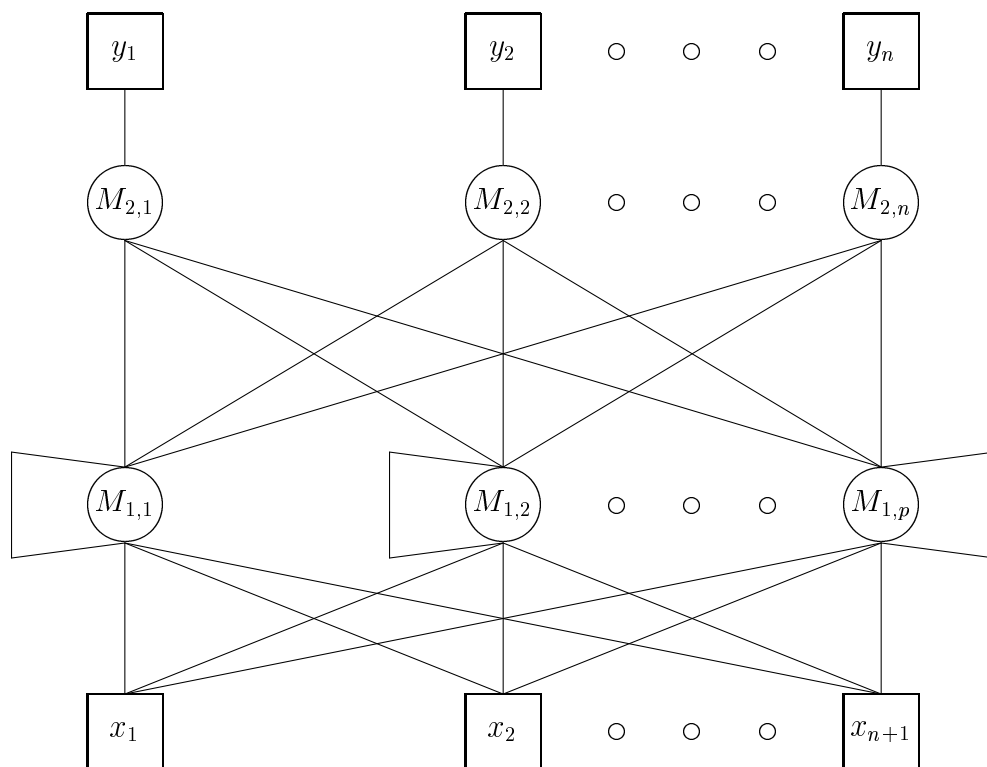


Рис. 3. Структура сети Хемминга  $NH_1$ .

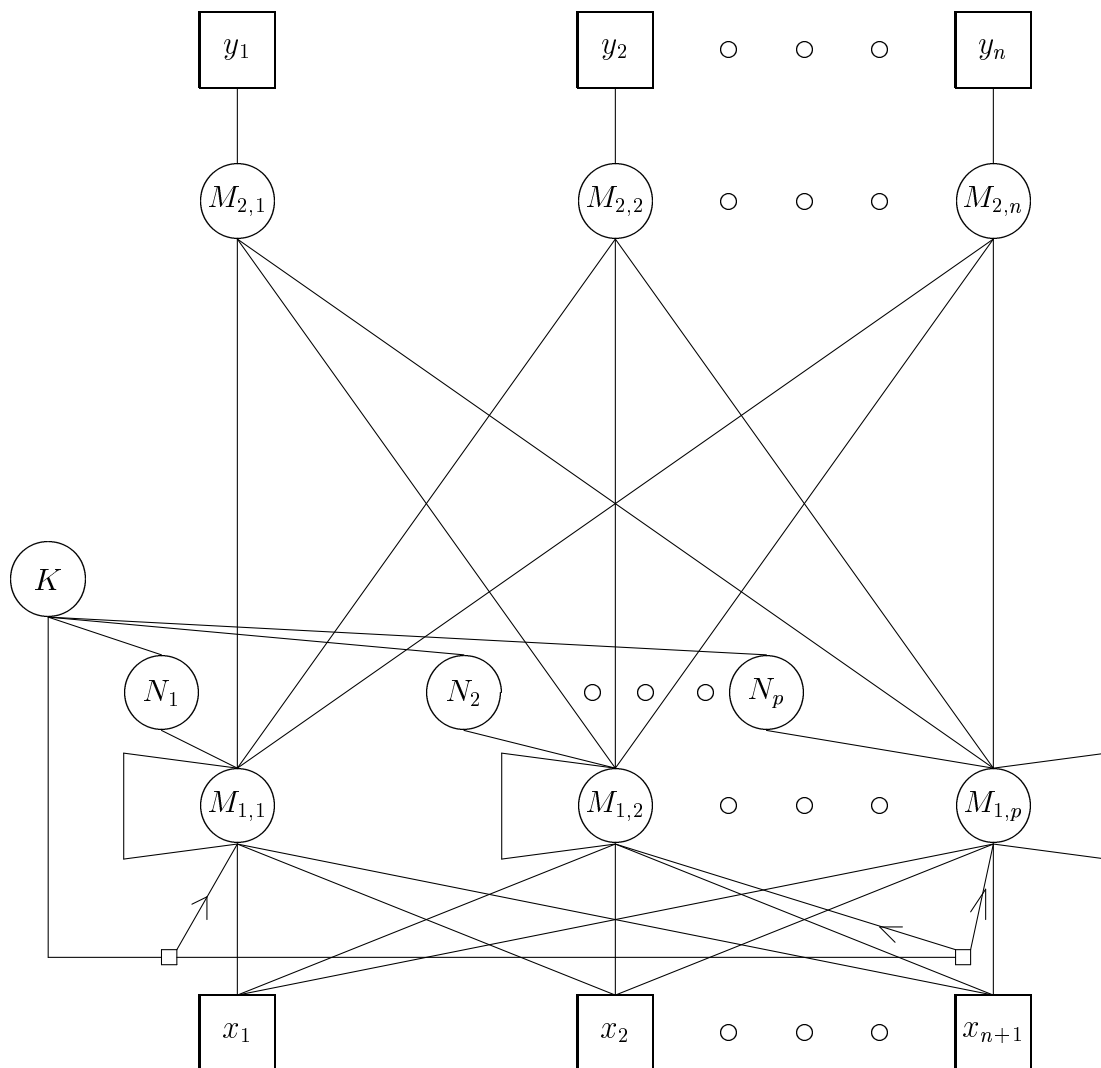


Рис. 4. Структура сети Хемминга  $NH_2$ .

# Самоорганизующиеся сети Кохонена

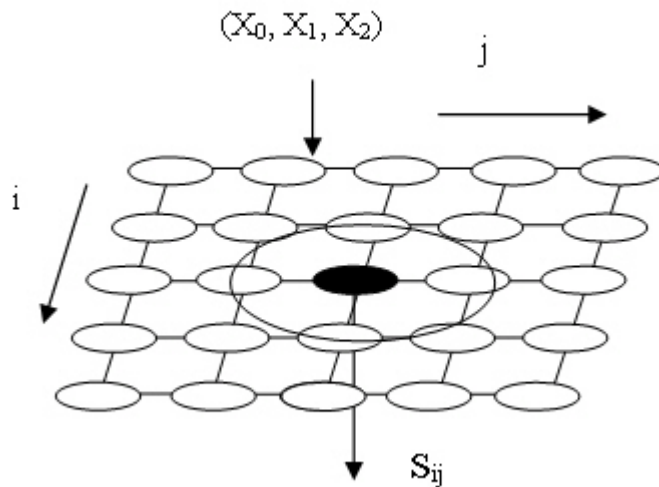
Характерной особенностью мозга является то, что его структура, по-видимому, отражает организацию внешних раздражителей, которые в него поступают. На пример, известно соответствие относительного положения рецепторов на поверхности кожи и нейронов в мозге, которые воспринимают и обрабатывают сигналы от этих рецепторов, а участки кожи, которые плотно «населены» рецепторами (лицо, руки), ассоциированы с пропорционально большим числом нейронов. Это соответствие образует то, что называется соматотропической картой, которая отражает поверхность кожи в часть мозга — соматосенсорную кору которая и воспринимает ощущение касания.

Системы, построенные на базе, например, многослойных сетей с обратным распределением ошибок имеют очевидный недостаток, заключающийся в том, что мы (как учитель) должны заранее заготовить входы и запастись соответствующими правильными ответами для обучения сети. Принципы функционирования природной соматотропической карты легли в основу создания самоорганизующихся сетей (карт, решеток) Кохонена, для которых не требуется предварительное обучение на примерах. Сеть Кохонена воспринимает только вход и способна вырабатывать свое собственное восприятие внешних стимулов.

Самоорганизующиеся сети Кохонена — это карты или многомерные решетки,  $d$  каждым узлом которой ассоциирован входной весовой вектор, то есть набор из  $k$  входных весов нейрона трактуется как вектор в пространстве. Входной весовой вектор имеет ту же размерность, что и вход в сеть. Обучение происходит в результате конкуренции, возникающей между узлами сети за право отклика на полученный входной сигнал. Элемент сети, который выигрывает в этой конкуренции (победитель), и его ближайшее окружение (свита) модифицируют веса своих входных связей. Перед обучением каждая компонента входного весового вектора инициализируется случайным образом. Обычно каждый вектор нормализуется в вектор с единичной длиной в пространстве весов. Это делается делением соответствующего веса на корень из суммы квадратов компонент этого весового вектора. Входные вектора нормализуются аналогично. Обучение сети состоит из следующих этапов:

1. Вектор  $x = (x_1, x_2, \dots, x_k)$  подается на вход сети.
2. Определяется расстояние  $d_{ij}$  (в  $k$ -мерном пространстве) между  $x$  и весовыми векторами  $ij$

каждого нейрона, например:  $d_{ij} = \sqrt{\sum_k (x_k - w_{ijk})^2}$ .



3. Нейрон, который имеет весовой вектор, самый близкий к  $x$ , объявляется «победителем». Этот весовой вектор  $w_{i,j}$  становится основным в группе входных весовых векторов, которые лежат в пределах расстояния  $D$  от  $w_{i,j}$ . Таким образом определяется "свита" победителя.

4. Группа входных весовых векторов модифицируется (поощряется) в соответствии со следующим выражением  $w_{ij}(t+1) = w_{ij}(t) + \eta(x - w_{ij}(t))$  для всех весовых векторов в пределах расстояния  $D$  от  $w_{i,j}$ .

5. Шаги 1 — 4 повторяются для каждого входного вектора. В процессе обучения значения  $D$  и постепенно уменьшаются:  $\eta : 1 \rightarrow 0$ ,  $D$  в начале обучения может равняться максимальному расстоянию между весовыми векторами, а к концу обучения доходить до величины, при которой будет обучаться только один нейрон.

Из формулы адаптации входного весового вектора следует, что он (для победителя и его «свиты») сдвигается по направлению к входному вектору. Таким образом, по мере поступления новых входных векторов весовые векторы сети разделяются на группы, формирующиеся в виде облаков (сгустков, кластеров) вокруг входных векторов. По мере обучения плотность весовых векторов будет выше в тех позициях пространства, где входные векторы появляются чаще, и наоборот. В результате сеть Кохонена адаптирует себя так, что плотность весовых векторов будет соответствовать плотности входных векторов. Так, если, например, на вход сети подать поток равномерно распределенных случайных величин, то сетевые веса будут самоприводиться в порядок в регулярную решетку.

# Генетические алгоритмы

Генетические алгоритмы - это аналитические технологии, созданные и выверенные самой природой за миллионы лет ее существования. Они позволяют решать задачи прогнозирования, классификации, поиска оптимальных вариантов, и совершенно незаменимы в тех случаях, когда в обычных условиях решение задачи основано на интуиции или опыте, а не на строгом (в математическом смысле) ее описании.

## Естественный отбор в природе

Эволюционная теория утверждает, что каждый биологический вид целенаправленно развивается и изменяется для того, чтобы наилучшим образом приспособиться к окружающей среде. В процессе эволюции многие виды насекомых и рыб приобрели защитную окраску, еж стал неуязвимым благодаря иглам, человек стал обладателем сложнейшей нервной системы. Можно сказать, что эволюция - это процесс оптимизации всех живых организмов. Рассмотрим, какими же средствами природа решает эту задачу оптимизации.

Основной механизм эволюции - это естественный отбор. Его суть состоит в том, что более приспособленные особи имеют больше возможностей для выживания и размножения и, следовательно, приносят больше потомства, чем плохо приспособленные особи. При этом благодаря передаче генетической информации (генетическому наследованию) потомки наследуют от родителей основные их качества. Таким образом, потомки сильных индивидуумов также будут относительно хорошо приспособленными, а их доля в общей массе особей будет возрастать. После смены нескольких десятков или сотен поколений средняя приспособленность особей данного вида заметно возрастает.

Чтобы сделать понятными принципы работы генетических алгоритмов, поясним также, как устроены механизмы генетического наследования в природе. В каждой клетке любого животного содержится вся генетическая информация этой особи. Эта информация записана в виде набора очень длинных молекул ДНК (Дезоксирибонуклеиновая Кислота). Каждая молекула ДНК - это цепочка, состоящая из молекул нуклеотидов четырех типов, обозначаемых А, Т, С и G. Собственно, информацию несет порядок следования нуклеотидов в ДНК. Таким образом, генетический код индивидуума - это просто очень длинная строка символов, где используются всего 4 буквы. В животной клетке каждая молекула ДНК окружена оболочкой - такое образование называется хромосомой.

Каждое врожденное качество особи (цвет глаз, наследственные болезни, тип волос и т.д.) кодируется определенной частью хромосомы, которая

называется геном этого свойства. Например, ген цвета глаз содержит информацию, кодирующую определенный цвет глаз. Различные значения гена называются его аллелями.

При размножении животных происходит слияние двух родительских половых клеток и их ДНК взаимодействуют, образуя ДНК потомка. Основным способом взаимодействия - кроссовер (cross-over, скрещивание). При кроссовере ДНК предков делятся на две части, а затем обмениваются своими половинками.

При наследовании возможны мутации из-за радиоактивности или других влияний, в результате которых могут измениться некоторые гены в половых клетках одного из родителей. Измененные гены передаются потомку и придают ему новые свойства. Если эти новые свойства полезны, они, скорее всего, сохранятся в данном виде - при этом произойдет скачкообразное повышение приспособленности вида.

## Что такое генетический алгоритм

Пусть дана некоторая сложная функция (целевая функция), зависящая от нескольких переменных, и требуется найти такие значения переменных, при которых значение функции максимально. Задачи такого рода называются задачами оптимизации и встречаются на практике очень часто.

Один из наиболее наглядных примеров - задача распределения инвестиций. В этой задаче переменными являются объемы инвестиций в каждый проект, а функцией, которую нужно максимизировать - суммарный доход инвестора. Также даны значения минимального и максимального объема вложения в каждый из проектов, которые задают область изменения каждой из переменных.

Попытаемся решить эту задачу, применяя известные нам природные способы оптимизации. Будем рассматривать каждый вариант инвестирования (набор значений переменных) как индивидуума, а доходность этого варианта - как приспособленность этого индивидуума. Тогда в процессе эволюции (если мы сумеем его организовать) приспособленность индивидуумов будет возрастать, а значит, будут появляться все более и более доходные варианты инвестирования. Остановив эволюцию в некоторый момент и выбрав самого лучшего индивидуума, мы получим достаточно хорошее решение задачи.

Генетический алгоритм - это простая модель эволюции в природе, реализованная в виде компьютерной программы. В нем используются как аналог механизма генетического наследования, так и аналог естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде.



## Моделирование генетического наследования:

Хромосома	Вектор (последовательность) из нулей и единиц. Каждая позиция (бит) называется геном.
Индивидуум = генетический код	Набор хромосом = вариант решения задачи.
Кроссовер	Операция, при которой две хромосомы обмениваются своими частями.
Мутация	Случайное изменение одной или нескольких позиций в хромосоме.

Чтобы смоделировать эволюционный процесс, сгенерируем вначале случайную популяцию - несколько индивидуумов со случайным набором хромосом (числовых векторов). Генетический алгоритм имитирует эволюцию этой популяции как циклический процесс скрещивания индивидуумов и смены поколений.

Жизненный цикл популяции - это несколько случайных скрещиваний (посредством кроссовера) и мутаций, в результате которых к популяции добавляется какое-то количество новых индивидуумов.

Отбор в генетическом алгоритме - это процесс формирования новой популяции из старой, после чего старая популяция погибает. После отбора к новой популяции опять применяются операции кроссовера и мутации, затем опять происходит отбор, и так далее.

Отбор в генетическом алгоритме тесно связан с принципами естественного отбора в природе следующим образом:

Приспособленность индивидуума	Значение целевой функции на этом индивидууме.
Выживание наиболее приспособленных	Популяция следующего поколения формируется в соответствии с целевой функцией. Чем приспособленнее индивидуум, тем больше вероятность его участия в кроссовере, т.е. размножении.

Таким образом, модель отбора определяет, каким образом следует строить популяцию следующего поколения. Как правило, вероятность участия индивидуума в скрещивании берется пропорциональной его приспособленности. Часто используется так называемая стратегия элитизма, при которой несколько лучших индивидуумов переходят в следующее поколение без изменений, не участвуя в кроссовере и отборе. В любом случае каждое следующее поколение будет в среднем лучше предыдущего. Когда приспособленность индивидуумов перестает заметно увеличиваться, процесс останавливают и в качестве решения задачи оптимизации берут наилучшего из найденных индивидуумов.

Возвращаясь к задаче оптимального распределения инвестиций, поясним особенности реализации генетического алгоритма в этом случае.

Индивидуум = вариант решения задачи = набор из 10 хромосом  $X_j$

Хромосома  $X_j$  = объем вложения в проект  $j$  = 16-разрядная запись этого числа

Так как объемы вложений ограничены, не все значения хромосом являются допустимыми. Это учитывается при генерации популяций.

Так как суммарный объем инвестиций фиксирован, то реально варьируются только 9 хромосом, а значение 10-ой определяется по ним однозначно.

## Генетический алгоритм

1. Создание структуры решения искомой задачи в виде массива  $a[i]$ ,  $i = 1, \dots, n$ , где  $n$  - максимальное число компонент структуры. Пример: поиск функции  $y=f(x)$  наилучшего в классе полиномов приближения экспериментальных точек  $\{x_i, y_i\}$ ,  $j=1, \dots, m$ .

Структура определяется битовым массивом, где каждому элементу массива сопоставлен простейший многочлен типа  $x^i$ ,  $i=1, \dots, n$ , где  $n$  - максимальная степень полинома.

2. Создание показателя эффективности структуры, заполненной конкретными значениями. Пример: Показателем эффективности для нашего примера будет невязка определенная методом наименьших квадратов  $J_a = I_1 + I_2 + \dots + I_m$ , где  $I_j = (y_j - f_a(x_j))^2$ ,

где  $f_a(x)$  есть сумма всех элементов вида  $a_i x^i$ , где  $a_i = 0$  или 1

3. Задание некоторого массива различных структур  $S_k$ ,  $k=1, \dots, N$ , размерностью  $N$ , большей, чем число компонент  $n$  в структуре

Данный массив можно сгенерировать случайно, задав нули и единицы в каждой структуре.

4. Расчет показателей эффективности  $J_k$  для каждой структуры  $S_k$ . По формуле заданной в пункте 2.

5. Естественный отбор структур по некоторому правилу выбора наилучших структур среди заданного массива структур. Пример: можно по правилу вида  $J_0 = M(J_k)$  - среднее значение  $J_k$ , если  $J_k < J_0$ , то структура остается, иначе умирает. Если разница между предыдущим  $J_0$  и новым  $J_0$  меньше какого-то малого числа, то конец расчета.

6. Замена выбывших структур на новые, рожденные от наиболее приспособленных структур с помощью генетических операторов

а.) мутация - замена в структуре одного из значений случайно выбранной компоненты

Пример: из (1, 1, 0, 1, 0, 0, 1, 0) получится (1, 1, 0, 1, 1, 0, 1, 0).

б.) инверсия - перестановка в структуре некоторой ее части наоборот

Пример: из (1, 1, 0, 1, 0, 0, 1, 0) получится (1, 1, 0, 0, 1, 0, 1, 0).

в.) кроссинговер - создание структуры, основанной на двух структурах - заменой одной части первой структуры на ту же область во второй.

Пример: из (A, B, C, D, E) и (a, b, c, d, e) получится (A, B, c, d, E).

7. Переход к этапу 4.

Влияние параметров генетического алгоритма на эффективность поиска

Операторы кроссовера и мутации

Наиболее традиционным подходом является отход от традиционной схемы "размножения", используемой в большинстве реализованных ГА-мах и повторяющих классическую схему. Классическая схема предполагает ограничение численности потомков путем использования так называемой вероятности кроссовера. Такая модель придает величине, соответствующей численности потомков, вообще говоря, недетерминированный характер. Есть метод предлагающий отойти от вероятности кроссовера и использовать фиксированное число брачных пар на каждом поколении, при этом каждая брачная пара "дает" двух потомков. Такой подход хорош тем, что делает процесс поиска более управляемым и предсказуемым в смысле вычислительных затрат.

В качестве генетических операторов получения новых генотипов "потомков", используя генетическую информацию хромосомных наборов родителей мы применяются два типа кроссоверов - одно- и двухточечный. Вычислительные эксперименты показали, что даже для простых функций нельзя говорить о преимуществе того или иного оператора. Более того было показано, что использование механизма случайного выбора одно- или двух точечного кроссовера для каждой конкретной брачной пары подчас оказывается более эффективным, чем детерминированный подход к выбору кроссоверов, поскольку достаточно трудно определить который из двух операторов более подходит для каждого конкретного ландшафта приспособленности. Использование же случайного выбора преследовало целью прежде всего сгладить различия этих двух подходов и улучшить показатели среднего ожидаемого результата. Для всех представленных тестовых функций так и произошло, - случайного выбор оказался эффективнее худшего.

Повышение эффективности поиска при использовании случайного выбора операторов кроссовера повлияло на то, чтобы применить аналогичный подход при реализации процесса мутагенеза новых особей, однако в этом случае преимущество перед детерминированным подходом не так очевидно в силу традиционно малой вероятности мутации. В основном вероятность мутации составляет 0.001 - 0.01.

## Выбор родительской пары

Первый подход самый простой - это случайный выбор родительской пары ("*панмиксия*"), когда обе особи, которые составят родительскую пару, случайным образом выбираются из

всей популяции, причем любая особь может стать членом нескольких пар. Несмотря на простоту, такой подход универсален для решения различных классов задач. Однако он достаточно критичен к численности популяции, поскольку эффективность алгоритма, реализующего такой подход, снижается с ростом численности популяции.

Второй способ выбора особей в родительскую пару - так называемый *селективный*. Его суть состоит в том, что "родителями" могут стать только те особи, значение приспособленности которых не меньше среднего значения приспособленности по популяции, при равной вероятности таких кандидатов составить брачную пару. Такой подход обеспечивает более быструю сходимость алгоритма. Однако из-за быстрой сходимости селективный выбор родительской пары не подходит тогда, когда ставится задача определения нескольких экстремумов, поскольку для таких задач алгоритм, как правило, быстро сходится к одному из решений. Кроме того, для некоторого класса задач со сложным ландшафтом приспособленности быстрая сходимость может превратиться в преждевременную сходимость к

квазиоптимальному решению. Этот недостаток может быть отчасти компенсирован использованием подходящего механизма отбора (о чем будет сказано ниже), который бы "тормозил" слишком быструю сходимость алгоритма.

Другие два способа формирования родительской пары, на которые хотелось бы обратить внимание, это *инбридинг* и *аутбридинг*. Оба эти метода построены на формировании пары на основе близкого и дальнего "родства" соответственно. Под "родством" здесь понимается расстояние между членами популяции как в смысле геометрического расстояния особей в пространстве параметров. В связи с этим будем различать генотипный и фенотипный (или географический) инбридинг и аутбридинг. Под инбридингом понимается такой метод, когда первый член пары выбирается случайно, а вторым с большей вероятностью будет максимально близкая к нему особь. Аутбридинг же, наоборот, формирует брачные пары из максимально далеких особей. Использование генетических инбридинга и аутбридинга оказалось более эффективным по сравнению с географическим для всех тестовых функций при различных параметрах алгоритма. Наиболее полезно применение обоих представленных методов для многоэкстремальных задач. Однако два этих способа по-разному влияют на поведение генетического алгоритма. Так инбридинг можно охарактеризовать свойством концентрации поиска в локальных узлах, что фактически приводит к разбиению популяции на отдельные локальные группы вокруг подозрительных на экстремум участков ландшафта, напротив аутбридинг как раз направлен на предупреждение сходимости алгоритма к уже найденным решениям, заставляя алгоритм просматривать новые, неисследованные области.

## Механизм отбора

Обсуждение вопроса о влиянии метода создания родительских пар на поведение генетического алгоритма невозможно вести в отрыве от реализуемого механизма отбора при формировании нового поколения. Наиболее эффективные два механизма отбора элитный и отбор с вытеснением.

Идея элитного отбора, в общем, не нова, этот метод основан на построении новой популяции только из лучших особей репродукционной группы, объединяющей в себе родителей, их потомков и мутантов. В основном это объясняют потенциальной опасностью преждевременной сходимости, отдавая предпочтение пропорциональному отбору. Быстрая сходимость, обеспечиваемая элитным отбором, может быть, когда это необходимо, с успехом компенсирована подходящим методом выбора родительских пар, например аутбридингом. Именно такая комбинация "аутбридинг - элитный отбор" является одной из наиболее эффективных.

Второй метод, на котором хотелось бы остановиться, это отбор вытеснением. Будет ли особь из репродукционной группы заноситься в популяцию нового поколения, определяется не только величиной ее приспособленности, но и тем, есть ли уже в формируемой популяции следующего поколения особь с аналогичным хромосомным набором. Из всех особей с одинаковыми генотипами предпочтение сначала, конечно же, отдается тем, чья приспособленность выше. Таким образом, достигаются две цели: во-первых, не теряются лучшие найденные решения, обладающие различными хромосомными наборами, а во-вторых, в популяции постоянно поддерживается достаточное генетическое разнообразие. Вытеснение в данном случае формирует новую популяцию скорее из далеко расположенных особей, вместо особей, группирующихся около текущего найденного решения. Этот метод особенно хорошо себя показал при решении многоэкстремальных задач, при этом помимо определения глобальных экстремумов появляется возможность выделить и те локальные максимумы, значения которых близки к глобальным.

## Особенности генетических алгоритмов

Генетический алгоритм - новейший, но не единственно возможный способ решения задач оптимизации. С давних пор известны два основных пути решения таких задач - переборный и локально-градиентный. У этих методов свои достоинства и недостатки, и в каждом конкретном случае следует подумать, какой из них выбрать.

Рассмотрим достоинства и недостатки стандартных и генетических методов на примере классической задачи коммивояжера. Суть задачи состоит в том, чтобы найти кратчайший замкнутый путь обхода нескольких городов, заданных своими координатами. Оказывается, что уже для 30 городов поиск оптимального пути представляет собой сложную задачу, побудившую развитие различных новых методов (в том числе нейросетей и генетических алгоритмов).

Каждый вариант решения (для 30 городов) - это числовая строка, где на  $j$ -ом месте стоит номер  $j$ -ого по порядку обхода города. Таким образом, в этой задаче 30 параметров, причем не все комбинации значений допустимы. Естественно, первой идеей является полный перебор всех вариантов обхода.

Переборный метод наиболее прост по своей сути и тривиален в программировании. Для поиска оптимального решения (точки максимума целевой функции) требуется последовательно вычислить значения целевой функции во всех возможных точках, запоминая максимальное из них. Недостатком этого метода является большая вычислительная стоимость. В частности, в задаче коммивояжера потребуется просчитать длины более  $10^{30}$  вариантов путей, что совершенно нереально.

Второй способ основан на методе градиентного спуска. При этом вначале выбираются некоторые случайные значения параметров, а затем эти значения постепенно изменяют, добиваясь наибольшей скорости роста целевой функции. Достигнув локального максимума, такой алгоритм останавливается, поэтому для поиска глобального оптимума потребуются дополнительные усилия.

Градиентные методы работают очень быстро, но не гарантируют оптимальности найденного решения. Они идеальны для применения в так называемых *унимодалных* задачах, где целевая функция имеет единственный локальный максимум (он же - глобальный). Легко видеть, что задача коммивояжера унимодалной не является.

Типичная практическая задача, как правило, мультимодальна и многомерна, то есть содержит много параметров. Для таких задач не существует ни одного универсального метода, который позволял бы достаточно быстро найти абсолютно точное решение.

Однако, комбинируя переборный и градиентный методы, можно надеяться получить хотя бы приближенное решение, точность которого будет возрастать при увеличении времени расчета.

Генетический алгоритм представляет собой именно такой комбинированный метод. Механизмы скрещивания и мутации в каком-то смысле реализуют переборную часть метода, а отбор лучших решений - градиентный спуск.