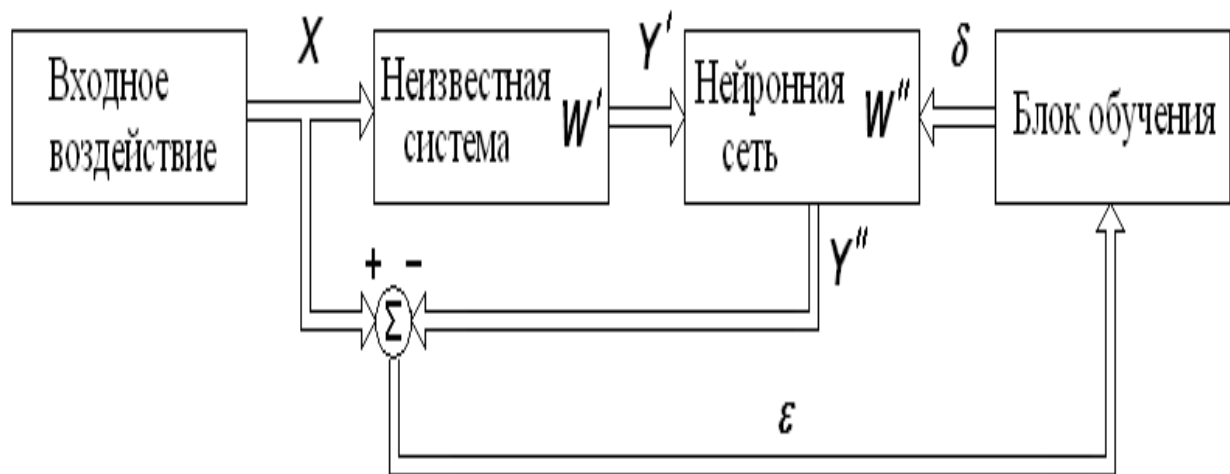


ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ВОЛЖСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

В. И. Капля

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Электронное учебное пособие



2018

УДК 004.8(07)
ББК 32.813я73
К 203

Рецензенты:

Зав. кафедрой «Автоматизация технологических процессов
и производств» филиала ФГБОУ ВО «Национальный исследовательский
университет «МЭИ» в г. Волжском,
канд. тех. наук, доцент
И. А. Болдырев,
ООО «СИНЕРГИЯ», инженер первой категории,
канд. тех. наук
С. А. Браганец

Печатается по решению редакционно-издательского совета
Волгоградского государственного технического университета

Капля, В. И.

Интеллектуальные системы [Электронный ресурс] : учеб. пособие /
В. И. Капля ; ВПИ (филиал) ВолГТУ. – Электрон. текстовые дан. (1 файл:
3,2 МБ). – Волжский, 2018. – Режим доступа: <http://lib.volpi.ru>. – Загл. с
титул. экрана.

ISBN 978-5-9948-2774-1

Содержит сведения о принципах построения и структуре интеллектуальных
систем, о средствах и методах моделирования и аппаратной реализации.

Рекомендуется для использования в учебном процессе по техническим
специальностям, в том числе по направлению магистратуры 15.04.04 «Автоматизация
технологических процессов и производств» и бакалавриата 09.03.04 «Программная
инженерия», при изучении дисциплин "Интеллектуальные системы", "Системы
искусственного интеллекта" и "Математические основы искусственного интеллекта"
для студентов дневной, вечерней и заочной форм обучения.

Ил. 61, табл. 1 , библиограф.: 24 назв.

ISBN 978-5-9948-2774-1

© Волгоградский государственный
технический университет, 2018
© Волжский политехнический
институт, 2018

Содержание

Содержание	3
Введение	4
1. Основные направления исследования ИС	5
2. Классификация ИС	6
2.1. Системы без обучения	7
2.2. Обучающиеся системы	8
2.3. Самообучающиеся системы	10
3. Средства и уровни понимания СИИ	11
4. Классификация типов поведения	15
5. Модель поступка. Мотивы и реализации	16
6. Машинное творчество, процедуры мышления	17
7. Модель формирования условных рефлексов	19
8. Корреляционная и пеленгационная характеристики изображений	21
9. Распознавание точечных изображений	26
10. Распознавание контурных изображений	28
11. Модель живого нейрона	32
12. Математическая модель нейрона и нейронных сетей	34
13. Алгоритм обучения персептрона	37
14. Топологические свойства однослойных, двухслойных и трехслойных НС. Теорема Минского	39
15. Свойства НС Хопфилда	44
16. Свойства НС Хемминга	47
17. Синхронные параллельные и последовательные НС	49
18. Метод обратного распространения ошибки	50
19. Матричная форма МОРО	53
20. Самообучение НС методом Хебба	54
21. Генетический алгоритм обучения НС	58
22. Сходимость алгоритмов обучения	60
23. Самоорганизующаяся коммуникация в системе с нейросетевым управлением	61
24. Структура микросхем, реализующих НС элементы	63
25. Распознавание образов на основе критерия Байеса	65
26. Распознавание образов с применением минимаксного критерия	68
27. Распознавание образов с применением критерия Неймана - Пирсона	69
28. Процедура последовательных решений в распознавании образов	70
29. Пример определения пороговых значений для вероятностных алгоритмов распознавания	71
30. Основные определения исчисления предикатов	73
31. Высказывания. Кванторы и их свойства	74
32. Правила вывода. Унификация	75
33. Выполнимость и удовлетворимость ППФ	76
34. Алгоритм преобразования предикатов в предложения	77
35. Резолюция и резольвента	78
36. Система опровержения на основе резолюций	79
37. Стратегии управления процессом поиска доказательства методом резолюций	81
38. Извлечение ответа из опровержения, основанного на резолюции. Этапы процесса извлечения ответа	83
39. Элементы логики нечетких множеств	85
40. Операции на нечетких множествах	86
41. Нечеткие правила вывода	88
42. Нечеткие нейронные сети	89
43. Идентификация систем обучаемыми интеллектуальными системами	89
44. Нейросетевая обработка динамических процессов	91
Список литературы	94

Введение

Интеллектуальные системы (ИС) – это перспективное и активно исследуемое направление развития систем управления производственными и сервисными процессами, которое основано на интенсивном взаимодействии с контролируемыми объектами путем интеллектуального анализа входной (измерительной) информации, учета предыдущих состояний системы и прогнозирования результатов управления [1-3]. Согласно фундаментальным изданиям и периодической печати существует ряд практически эквивалентных терминов: “интеллектуальные системы”, “системы искусственного интеллекта”, “машинное мышление”.

Интеллектуальность системы – это сложное и многогранное понятие, которое включает в качестве основных элементов понятия оптимальности, адаптивности и обучаемости, кроме того оно охватывает логические, семантические и нечеткие методологии. Реализация интеллектуальных систем осуществляется посредством разработки аппаратно-программных комплексов, имеющих развитую периферию для ввода измерительной и мультимедийной информации, а так же для реализации управляющих команд.

Известны работы направленные на обоснование целесообразности применения ИС для определения параметров качества деталей в механообработке, оптимизации процессов формообразования и автоматизированного управления [4,5]. Большинство широко используемых офисных и научных программ имеют в своей структуре интеллектуальные опции, например, проверка правописания, распознавание текста (изображений, звуков), запоминание часто повторяемых операций. Программы, специализированные в определенных областях деятельности человека, принято называть “экспертными системами”. Автономное воплощение ИС находят в специальных интеллектуальных системах, в частности в роботах, нейрокомпьютерах.

С точки зрения общих проблем познания предпринимаются попытки создания ИС для решения двух основных интеллектуальных задач [12]:

- интенсификации процесса диалога человек-ЭВМ с целью создания новых творческих произведений;
- модельной проверки гипотез о структуре и особенностях творческих процессов.

Многообразие интеллектуальных задач требует привлечения широкого круга математических объектов и методов, в первую очередь математический и функциональный анализ, методы оптимизации, алгебру множеств (классическую и нечеткую), теорию вероятности.

Основой данного учебного пособия является ранее изданное пособие [19], которое было расширено новой информацией и откорректировано по результатам анализа новой литературы и периодических изданий, а так же практических занятий со студентами.

1. Основные направления исследования ИС

Назначение ИС состоит в освобождении человека от трудоемких процессов обработки информации, накопления и применения знаний при решении сложных научно-технических задач.

Основные направления исследования ИС [12]:

1. Представление знаний (формализация).
2. Манипулирование знаниями: пополнение, классификация, обобщение, вывод новых знаний.
3. Общение на естественных языках и носителях с ЭВМ.
4. Восприятие зрительных сцен, речевых сообщений и текстов.
5. Обучение решению новых задач, повышению качества решений.
6. Поведение (учет состояния окружения).

Приведенная классификация исследовательских направлений не единственная. Все указанные направления не имеют четких границ, и в некоторой степени в каждом из них присутствуют все другие направления.

Разработка ИС предполагает: определение структуры исходной информации; разработку алгоритмов обучения и работы; создание баз данных и знаний.

Распознавание образов – основной вид задач, решаемых ИС. Термин “распознавание” является эквивалентом понятия “классификация” и предназначен для указания сложной структуры исходной информации. Объектами распознавания могут служить изображения, звуки, характеристики статистических данных. Выходным сигналом ИС является вектор решений, отражающий принадлежность входного образа определенному множеству.

2. Классификация ИС

Множественность параметров и свойств ИС предполагает несколько способов классификации этих систем. Рассмотрим классификацию по уровню обученности системы [6,7]. Такая классификация отображает уровень универсальности системы и значимость априорной информации для ее работы. Возможность обучения технической системы является эффективным способом настройки, который можно противопоставить процессу предварительного определения всех параметров системы до начала ее работы. Решение задачи обучения требует знания лишь структуры системы, а параметры ее настроек определяются в процессе обучения, что гораздо проще и реалистичнее, чем полное и предварительное решение всех возможных ситуационных задач.

Типичным видом задач, решаемых ИС, является задача классификации элементов некоторого множества на два или более классов. В пространстве признаков, используемых для решения этой задачи, необходимо определить

разделяющую функцию $\gamma(\bar{X}, A)$, которая принимает положительные значения для элементов подмножества Ω_1 и отрицательные для подмножества Ω_2 . На рис.2.1. показан пример решения такой задачи для случая $\gamma(\bar{X}, A) = \gamma(x_1, x_2, A)$.

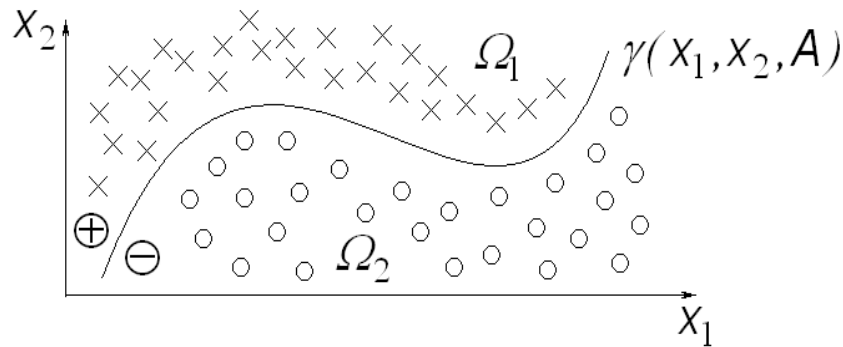


Рис.2.1. Разделение множеств в пространстве признаков.

Разделяющая функция является параметрической, то есть зависит от набора параметров, образующих матрицу A . Качество работы ИС во многом определяется свойствами разделяемых подмножеств, которые могут не иметь четкой границы. В случае решения подобной задачи для 3-х и большего числа подмножеств используется принцип дихотомии, который состоит в группировании разделяемых подмножеств и построении множества разделяющих функций.

2.1. Системы без обучения

Система устанавливает принадлежность объектов множества ω классам $\{\Omega_i\}_{i=1, \dots, m}$. Свойства объекта определяются техническими средствами измерения (ТСИ), которые формируют вектор признаков объекта $\bar{X}$. Анализ свойств производится с помощью блока алгоритмов распознавания (БАР) с использованием априорной информации (АИ). Схема ИС без обучения приведена на рис.2.2.

признаков $x_1, \dots, x_N$. Цель процедуры обучения – определение разделяющих функций $\gamma(x_1, x_2, A)$, $i=1, \dots, m$. Обучение проводится путем многократного предъявления системе распознавания различных объектов с указанием классов, к которым эти объекты принадлежат.

Обучающиеся системы распознавания "работают с учителем", как показано на рис.2.3.

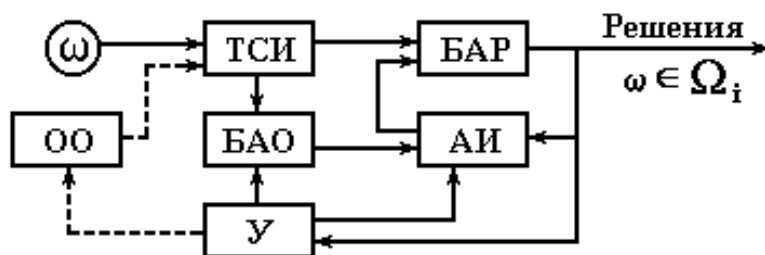


Рис.2.3. Схема обучающейся системы

--- режим обучения; — режим экзамена

Процесс обучения заключается в том, что "учитель" (У) многократно предъявляет системе обучающие объекты (ОО) всех выделенных классов и указывает, к каким классам они принадлежат. Затем "учитель" начинает экзаменовать систему распознавания, корректируя ее ответы до тех пор, пока количество ошибок в среднем не достигнет допустимого уровня.

В состав обучающейся системы входят: ТСИ – технические средства измерения; БАО – блок алгоритмов обучения; АИ – априорная информация; БАР – блок алгоритмов распознавания. Обучаемая система распознавания реализует разделяющую функцию $\gamma(\tilde{X}, A)$, однако, в начальной стадии обучения матрица параметров A является неизвестной и алгоритмы обучения направлены на поиск значений элементов этой матрицы.

Обучаемые системы обладают гибкостью в том смысле, что перенастройка системы для решения нового вида задач реализуется путем дополнительного цикла обучения, без изменения ее физической конфигурации. Обучение является формой адаптации системы при смене условий функционирования.

2.3. Самообучающиеся системы

В случае если первоначальной априорной информации достаточно лишь для определения словаря признаков $X_1, X_2, \dots, X_N$, но недостаточно для проведения классификации объектов, используются самообучающиеся системы. На стадии формирования самообучающейся системе предъявляют исходную совокупность объектов $\omega_1, \dots, \omega_z$, заданных значениями своих признаков для ω_1 - $(x_1^1, \dots, x_N^1)$; ... ; для ω_z - $(x_1^z, \dots, x_N^z)$. Однако из-за ограниченного объема первоначальной информации система при этом не получает указаний о том, к какому классу объекты исходной совокупности принадлежат. Эти указания заменяются набором правил, в соответствии с которыми на стадии самообучения система распознавания сама вырабатывает классификацию и в дальнейшем ее придерживается. Выбранная системой классификация может отличаться от естественной. Обобщенная структура самообучающейся ИС приведена на рис.2.4.

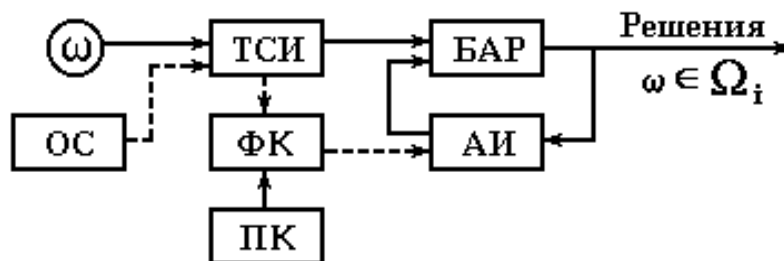


Рис.2.4. Схема самообучающейся системы

--- режим самообучения; — распознавание неизвестных объектов

В состав самообучающейся системы входят: ОС – объекты для самообучения; ТСИ – технические средства измерения; АИ – априорная информация; БАР – блок алгоритмов распознавания; ПК – правила классификации; ФК – формирователь классов.

Результаты решения задачи распознавания неизвестных объектов следует использовать для уточнения априорного описания классов, поэтому при построении систем распознавания целесообразно использовать принцип обратной связи. Блок априорной информации должен быть снабжен специальными алгоритмами корректировки априорных описаний классов. Разделяющие функции $\gamma(\bar{X}, A)$ в случае самообучающейся ИС определены лишь структурно, размерность выходного сигнала определяется в процессе самообучения.

Обучающиеся и самообучающиеся системы получают недостающую априорную информацию в процессе обучения или самообучения. Цель обучения и самообучения – выработать такое количество информации, которое необходимо для функционирования системы распознавания. Создание обучающихся и самообучающихся систем распознавания обусловлено тем, что на практике достаточно часто встречаются ситуации, когда отсутствует полная первоначальная априорная информация.

3. Средства и уровни понимания СИИ

Понимание – постижение смысла текста (или другого вида информации) и значения содержащейся в нем информации [12].

Средства понимания текстов:

- Герменевтика – использование только текста.
- Экзогетика – использование текста и всех сведений о контексте, в рамках которого этот текст создавался.
- Гомелетика – использование общих сведений о морально-этических нормах времени создания текста, мотивы и цели автора, его личностные характеристики.

Элементы средств понимания и их обозначения:

Т – текст на естественном языке, предъявленный СИИ;

Е – расширенный текст, включающий условия его порождения в некоторой среде (условия применения системы);

Р – расширенный текст, включающий его невербальные (неформальные) параметры, относящиеся к субъекту, создавшему текст (устройство системы);

TR – правила пополнения текста, опирающиеся на его структуру;

ER – правила пополнения текста, опирающиеся на знания о среде;

PR – правила пополнения текста, опирающиеся на знания о психологии личности;

А – ответ, формируемый системой;

К – знания в базе знаний;

Ф – факты, хранящиеся в базе фактов;

FRK – правила порождения знаний из фактов;

КМ – метафорические знания (обобщение, ассоциации, интуиция);

KRM – правила порождения знаний метафорического уровня.

СИИ понимает текст, если она дает ответы, правильные с точки зрения человека (эксперта).

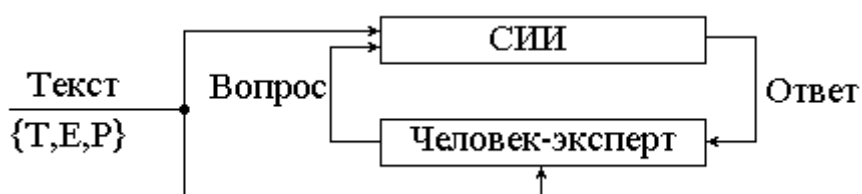


Рис.3.1. Общая структура системы понимания

Первый уровень понимания характеризуется схемой “ $T \Rightarrow A$ ”. СИИ этого уровня формирует ответ только на основе прямого содержания введенного текста Т. Лингвистический процессор проводит морфологический, синтаксический и семантический анализ текста и вопросов.

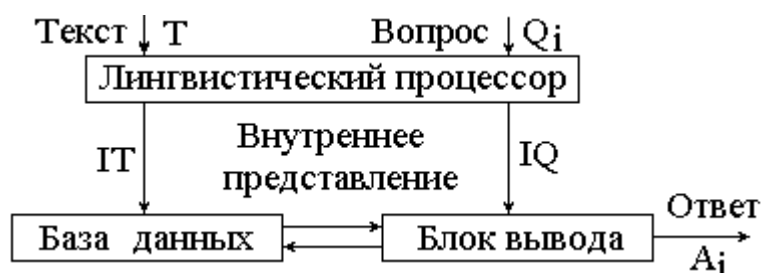


Рис.3.2. Первый уровень понимания (справочная система)

Второй уровень понимания характеризуется схемой “ $\{T, TR\} \Rightarrow A$ ”. СИИ использует средства логического вывода, то есть порождается информация, которая в явном виде в первоначальном тексте отсутствует. Например, при решении задачи поиска неисправности система способна локализовать положение неисправного элемента.



Рис.3.3. Второй уровень понимания (диагностическая система)

Третий уровень понимания характеризуется схемой “ $\{T, TR, ER\} \Rightarrow A$ ”. СИИ использует дедуктивный вывод и вывод по сценариям. Например, при решении задачи поиска неисправности система способна сообщить методы устранения выявленной неисправности.



Рис.3.4. Третий уровень понимания (советующая система)

Четвертый уровень понимания характеризуется схемой “ $\{E, TR, ER\} \Rightarrow A$ ”. СИИ выделяет условия возникновения текста и использует

их для решения задачи. Например, при решении задачи поиска неисправности система способна сообщить причины возникновения неисправности.

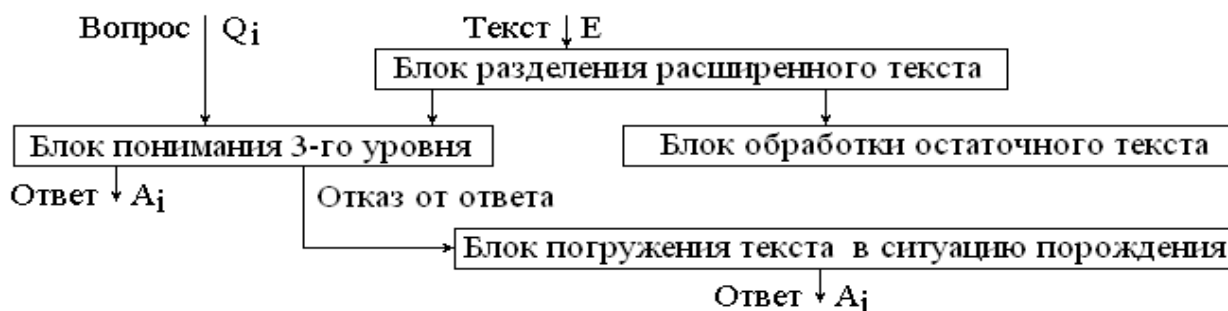


Рис.3.5. Четвертый уровень понимания (анализ причин неисправностей)

Пятый уровень понимания характеризуется схемой “{P,TR,PR} ⇒ A”. СИИ использует информацию о конкретном субъекте для коррекции своих ответов.



Рис.3.6. Пятый уровень понимания (анализ состояния системы с учетом персональных данных)

Первый метауровень понимания характеризуется схемой “{F,FRK} ⇒ K”. СИИ должна обучаться и пополнять свою базу знаний.

Второй метауровень понимания характеризуется схемой “{K,KRM} ⇒ KM”. СИИ использует методы аналогий и ассоциации.

4. Классификация типов поведения

Поведение – взаимодействие субъекта с внешней средой, основанное на анализе состояния элементов окружающей среды и путей достижения поставленной цели. Проблемы формирования поведения возникают при проектировании автономных интеллектуальных систем, оснащенных развитым функционалом для решения интеллектуальных задач.

Элементы поведения: наличие цели, наличие наблюдателя, наличие оценки. Цель может быть неосознанной субъектом в явной форме. Осознаваемый наблюдатель дает возможность субъекту взглянуть со стороны на свою деятельность. Роль наблюдателя может играть сам субъект или “мыслимый наблюдатель”. Наблюдатель должен формировать оценку деятельности субъекта. Наблюдатель является фоном, на котором планируется и строится поведение. Процесс достижения глобальной цели может включать достижение промежуточных целей, которые могут быть прямыми или косвенными.

Типы поведения [12]:

- Нормативное поведение – определяется обществом (средой), к которому принадлежит субъект.
- Ситуационное поведение – определяется реальным, конкретным состоянием среды. Оно может вынудить нарушить правила нормативного поведения.
- Аналитическое поведение – характеризуется осознанностью целей и плана их достижения.
- Развлекательное поведение – направляется целями с низким уровнем осознания при отсутствии вреда другим индивидам.
- Игровое поведение – направляется целями нанесения ущерба противнику.

5. Модель поступка. Мотивы и реализации

Поступок имеет два основных элемента: мотив (замысел) и реализацию. Эти элементы можно представить в виде пары графов: $\langle G_1, G_2 \rangle$. Графы имеют одинаковый набор помеченных вершин [12]:

I – субъект;

H – объект, на который направлен поступок;

T – оценивающий объект (наблюдатель);

N – окружающая среда;

U – морально-этические нормы общества;

S – свод юридических законов.

Вершины соединяются дугами, которым приписывают определенные коэффициенты, которые могут быть положительными или отрицательными (пунктирными). На рис.5.1 приведены графы мотива и реализации примера недальновидного поступка.

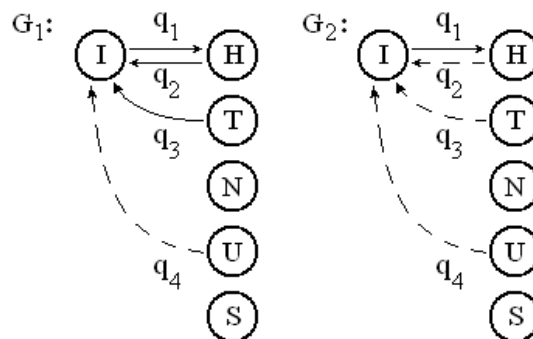


Рис.5.1. Графы примера недальновидного поступка

Субъект рассчитывает за счет положительного действия q_1 получить ответное действие с положительным весом q_2 , превышающим q_1 , то есть получить выгоду. Кроме того, он предполагает, что его персонифицированное окружение “Т” положительно воспримет его деятельность с весом q_3 , не смотря на отрицательную этическую оценку q_4 , то есть $q_2 + q_3 > q_1 + q_4$.

Если мотив и реализация в основном совпадут, то поступок следует характеризовать как “коростный”. Если вместо ожидаемой реализации, получится то, что соответствует графу G_2 , приведенному на рис.5.1, то поступок можно классифицировать как недальновидный.

Простейшие примеры графов других поступков приведены на рис.5.2. Для всех приведенных примеров считается, что мотив совпадает с реализацией.

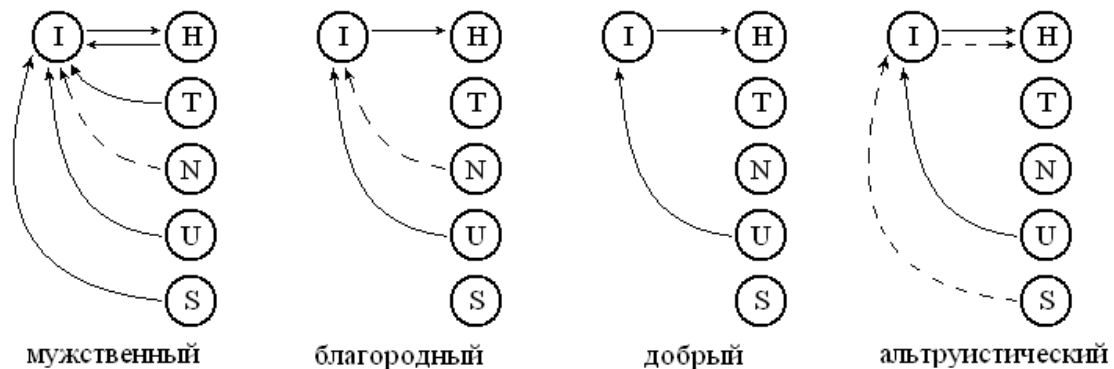


Рис.5.2. Примеры простейших графов различных поступков

Интерпретация понятия «поступок» для систем автоматизации соответствует адекватности формируемых управляющих сигналов состоянию операционной зоны системы, реакции на нештатные ситуации. Увеличение числа вершин графов целей и фактической реализации позволяет увеличить надежность и универсальность ИС управления.

6. Машинное творчество, процедуры мышления

Основная задача машинного творчества – моделирование творческого процесса с целью использования известных и поиска новых закономерностей человеческого творческого мышления [12].

Человек не фиксирует своего внимания на большинстве процедур мышления. После успешного завершения поиска алгоритма решения задачи,

считавшейся творческой, ее решение перестает считаться творческим процессом.

Основу творческого мышления человека составляют глобальные процедуры мышления:

1. Альтернативный выбор из множества вариантов.
2. Установление связей между начальной и целевой ситуациями.
3. Вычленение классифицирующей структуры.
4. Обобщение описаний.
5. Прогнозирование.
6. Заимствование ситуации из некоторого множества.
7. Варьирование ситуаций.

Художественные, научные и технологические произведения имеют определенную общую структуру:

- Основа – определенное множество исходных данных.
- Правила и ограничения, в том числе фантастические.
- Отклонения от правил, которые могут быть случайными, единичными и необязательными.

Литературное творчество строится на основе принципов литературных жанров, которым присущи определенные правила построения произведений:

- Задание исторических и предметных сцен.
- Разбиение персонажей на группы.
- Синтез сюжетной линии.

Стихотворные тексты не содержат полной информации и подразумевают неоднозначность восприятия, то есть читатель домысливает текст, ориентируясь на звуковую и эмоциональную направленность.

Методы машинной генерации литературных произведений используются в программах автоматического перевода и в системах диалога человек-ЭВМ.

Наделение технических систем управления функциями творческого решения необходимо в тех случаях, когда системе самостоятельно составить план действий: выбрать сценарий, определить необходимые средства для выполнения стоящей задачи. Например, перемещение предметов многозвенным манипулятором является примером задачи с большой степенью неоднозначности возможных решений даже для абсолютного пустого операционного пространства, так как число траекторий перемещения имеет мощность континуума.

7. Модель формирования условных рефлексов

Простейшим процессом обучения распознаванию можно считать формирование условных рефлексов. Процесс осуществляется на основе накопления положительного опыта, при этом способ определения результата опыта детально не рассматривается. Например, положительным результатом может являться факт превышения взаимной корреляционной функции двух изображений определенного порога.

Аналитическое соотношение, описывающее формирование условных рефлексов, имеет следующий вид:

$$A_n = a_n \vee \bigvee_{i=1}^s b_n^i \cdot P(N_n^i \geq N_0^i), \quad (7.1)$$

где A_n - реакция обучаемой системы на s раздражителей после обучающего шага n , a - безусловный раздражитель, b - нейтральный раздражитель, из которого формируется условный раздражитель, $P(.)$ - предикат, принимающий значение, равное 1, если условие в скобках выполнено, и 0 - в противном случае, N_n^i - параметр накопления воздействия нейтрального раздражителя, N_0^i - порог срабатывания i -го раздражителя.

Параметр накопления воздействия нейтрального раздражителя определяется по следующим формулам:

$$N_n^i = \sum_{k=1}^{n-1} \Delta N_k^i + \xi_n^i, \quad N_0 = 0, \quad \Delta N_n^i = \begin{cases} +\varphi_n^i, & \text{при } a_n \cdot b_n^i = 1 \\ -\psi_n^i, & \text{при } a_n \cdot \overline{b_n^i} = 1 \\ -\zeta_n^i, & \text{при } \overline{a_n} \cdot b_n^i = 1 \end{cases}, \quad (7.2)$$

где ξ_n^i - случайный фактор (настроение), φ_n^i - функция запоминания, ψ_n^i , ζ_n^i - функции забывания, обмана (угасания условного рефлекса).

В процессе формирования условных рефлексов принято выделять моменты появления условного рефлекса и его угасания.

Пример возможного задания последовательности появления раздражителей и функций запоминания и угасания приведен на рис.7.1.

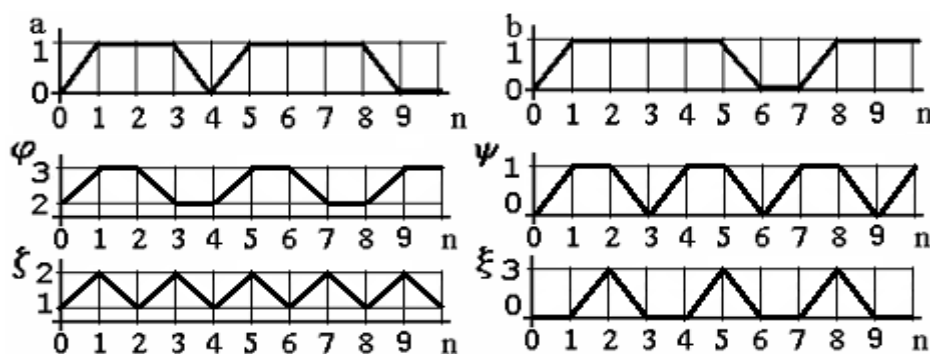


Рис.7.1. Диаграммы изменения функций, определяющих процесс формирования условных рефлексов

Кроме того, необходимо задать значение порога, например, $N_0=10$. Результаты расчетов по формулам (1.1) и (1.2) удобно заносить в следующую таблицу.

Таблица 7.1.

Результаты формирования условного рефлекса

n	ΔN	ΣN	N	P	A
1	3	3	3	0	1
2	3	6	9	0	1

3	2	8	8	0	1
4	-1	7	7	0	0
5	3	10	13	1	1
6	-0	10	10	1	1
7	-1	9	9	0	1

Моменты пересечения порога срабатывания N_0 представлены на рис.7.2.

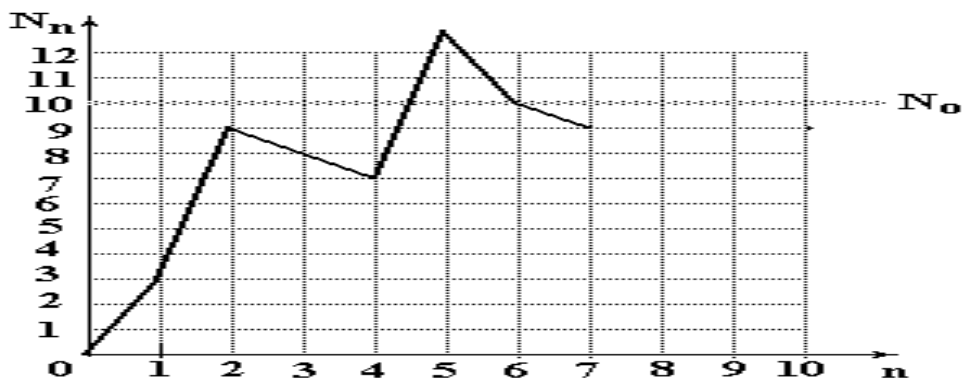


Рис.7.2. Процесс накопления положительного опыта

Множество вариантов можно получить, варьируя уровни и диаграммы факторов, времена прерывания и прекращения положительных и отрицательных воздействий.

8. Корреляционная и пеленгационная характеристики изображений

Распознавание изображений, основанное на анализе их корреляционных свойств, считается помехоустойчивым, так как шумовые составляющие обычно являются некоррелированными с изображениями процессами с единичным радиусом корреляции, то есть их вклад близок к нулю [6,8]. Значения автокорреляционной функции (АКФ) и взаимной корреляционной функции (ВКФ) двух изображений лежат в интервале $[-1, +1]$. Формула для вычисления коэффициента взаимной корреляции матриц A и B имеет вид:

$$\rho(A, B) = \text{corr}(A, B) = \frac{\sum_i \sum_j (A_{i,j} - m_A) \cdot (B_{i,j} - m_B)}{\sqrt{\sum_i \sum_j (A_{i,j} - m_A)^2 \cdot \sum_i \sum_j (B_{i,j} - m_B)^2}},$$

где m_A, m_B – средние значения элементов матриц A и B . Иногда с целью ускорения вычислений используются ненормированные и нецентрированные значения ВКФ, то есть в приведенной выше формуле отсутствует знаменатель и средние значения, при этом форма АКФ сохраняется. Понятие ВКФ возникает в том случае, когда одно или оба изображения зависят от некоторого параметра s , в качестве которого может выступать вектор пространственного смещения или время:

$$\rho(A, B(s)) = \text{corr}(A, B(s)).$$

В ЭВМ изображения хранятся в виде двумерных массивов – матриц. Если изображение черно-белое, то элемент такой матрицы соответствует яркости точки изображения, значение которой квантовано, и может принимать целые значения из диапазона 0-255. Цветные изображения кодируются в виде совокупности трех аналогичных матриц, каждая из которых содержит информацию об одной из цветовых компонент.

Изображение является исходной информацией для формирования набора фрагментов изображения, соответствующих определенным смещениям центра корреляционного отсчета. Следует отметить, что корреляционная функция определяется для бесконечных пределов суммирования (интегрирования), однако, для любого конкретного изображения это требование реализовать невозможно. Для решения практических задач можно вычислять приближенное значение АКФ или ВКФ, заранее ограничив пределы суммирования.

Критерием для определения размера фрагмента, ограничивающего процесс суммирования, служит радиус корреляции (РК) изображения. Существует несколько вариантов определения величины РК:

- величина относительного смещения изображений, при котором значение АКФ снижается до заданного уровня (например, 0.5 или 0.1 или 0);
- величина половины основания прямоугольника, площадь которого равна площади под графиком нормированной АКФ.

Определение радиуса корреляции, таким образом, требует использования АКФ, поэтому вычисление РК позволяет проверить правильность выбранного размера фрагмента. В первом приближении можно использовать следующее соотношение:

$$R_F \geq 5\rho_k, \quad (8.1)$$

где R_F - размер фрагмента, ρ_k - радиус корреляции.

Изображение имеет двумерную АКФ. При решении некоторых задач достаточно ограничиться одним сечением, например, горизонтальным, проходящим через центр. Использование одного сечения позволяет упростить аналитические соотношения.

Примем следующую систему обозначений: F_0 - центральный фрагмент изображения, который участвует в вычислении АКФ и считается неподвижным, F_i - фрагмент, смещенный на i дискретов относительно центрального фрагмента. Пример разбиения изображения на фрагменты показан на рис.8.1.

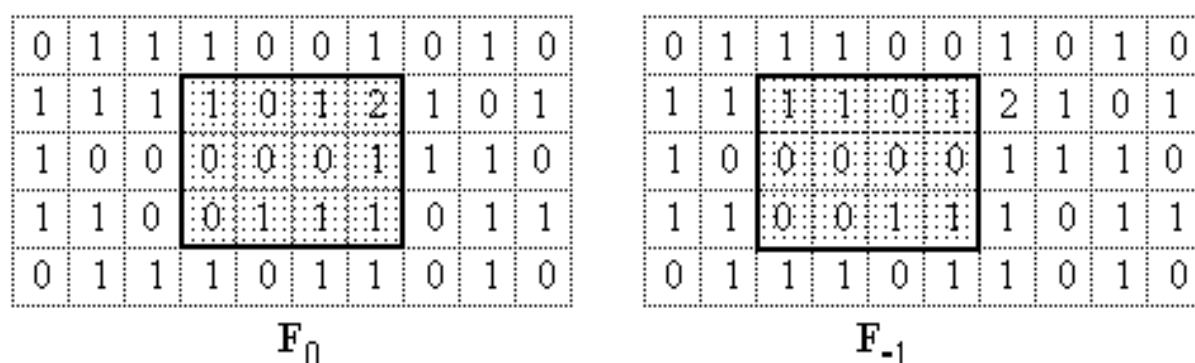


Рис.8.1. Центральный и смещенный влево фрагменты изображения.

Алгоритм вычисления АКФ имеет следующий вид:

$$R_A(s) = F_0 * F_S, \quad (8.2)$$

где s - величина смещения $\{-3, -2, -1, 0, 1, 2, 3\}$, $*$ - символ скалярного умножения матриц. Операция скалярного умножения матриц – это скалярное произведение векторов, образованных из элементов умножаемых матриц путем последовательного выстраивания их столбиков в вектора.

Ограниченность пределов вычисления АКФ приводит к потере некоторых свойств этой функции для стационарных процессов, например, симметричности. Для изображения, приведенного на рис.8.1, результат вычисления АКФ представлен в виде графика на рис.8.2.

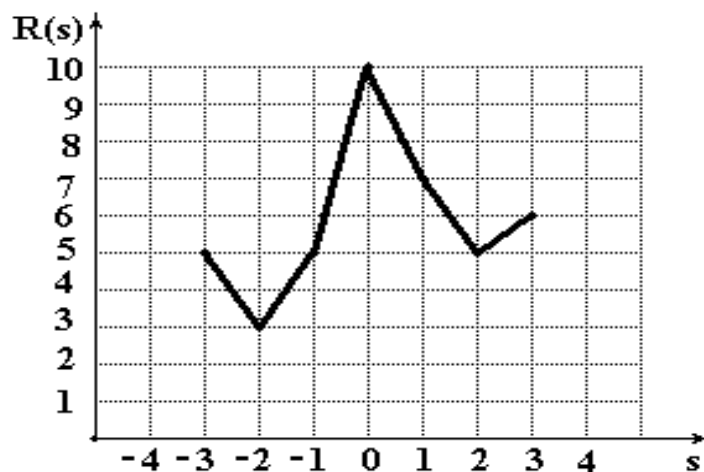


Рис.8.2. Сечение АКФ фрагмента изображения

Оценить радиус изображения можно по следующей приближенной

формуле:

$$\rho(s) = \frac{|s|}{1 - \frac{R(s)}{2 \cdot R(0)}} \quad (8.3.)$$

Наиболее точное значение РК получается при минимальном смещении, то есть при $s=1$. Если изображение считать изотропным, то целесообразно использовать усредненное значение по четырем возможным единичным смещениям фрагментов. Для ранее использованного изображения средний радиус корреляции вычисляется следующим образом:

$$\rho_{\text{ср}} = \frac{1}{4} \left(\frac{1}{1 - \frac{6}{20}} + \frac{1}{1 - \frac{5}{20}} + \frac{1}{1 - \frac{5}{20}} + \frac{1}{1 - \frac{7}{20}} \right) = 1.41$$

В системах технического зрения находит применение метод пеленгационных характеристик. Этот метод позволяет оценивать величину рассогласования в положении двух идентичных изображений. В качестве примера можно рассматривать задачу поиска точки сверления отверстия в заготовке сложной формы. Алгоритм формирования пеленгационной характеристики можно рассматривать в качестве примера одной из форм процесса обучения. Изображение является исходной информацией для формирования набора фрагментов изображения, соответствующих определенным смещениям поля зрения системы. В качестве примера множества фрагментов поля зрения используем рис.8.1.

Задача определения направления смещения в горизонтальном направлении может быть решена с помощью формирования обобщенной матрицы смещений по следующей формуле:

$$F_p = \sum_{i>0} F_i - \sum_{i<0} F_i. \quad (8.4)$$

Для примера, приведенного на рис.8.1, процесс вычисления обобщенной матрицы смещений имеет вид:

$$F_p = \begin{bmatrix} 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} - \\ - \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 2 & 3 & 2 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Обобщенная матрица смещений по размерам совпадает с размером фрагмента. Пеленгационная функция в качестве аргумента имеет величину смещения и задается следующим аналитическим соотношением:

$$P(s) = F_p * (F_s - F_0). \quad (8.5)$$

Результаты расчета пеленгационной функции для рассматриваемого примера по формулам (8.4) и (8.5) представлены на рис.8.3.

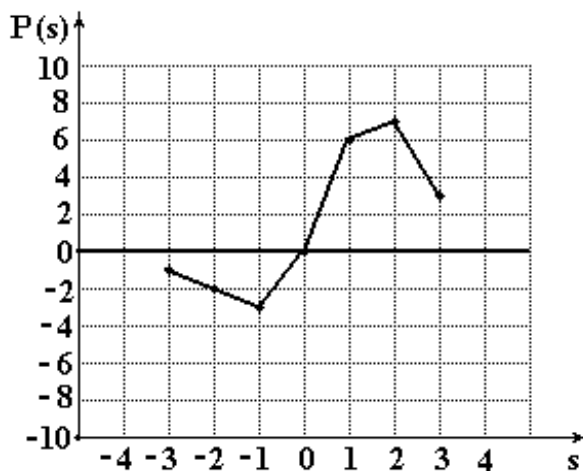


Рис.8.3. Пеленгационная функция

Полученный результат наглядно иллюстрирует свойства пеленгационной функции:

- при отклонении фрагмента вправо от центра функция положительна, при отклонении влево – отрицательна;
- имеется участок функции (рабочая зона), где ее производная положительна;
- размер рабочей зоны пропорционален радиусу корреляции АКФ.

Результатом формирования ПХ является обобщенное изображение, которое запоминается системой и используется для поиска требуемого объекта. Обобщенное изображение имеет размер одиночного фрагмента, что позволяет ускорить процесс вычислений и сократить объем необходимой памяти.

9. Распознавание точечных изображений

Изображение некоторых объектов на снимках соответствует одиночным пикселям. Примерами таких систем являются системы навигации по звездам или оптические измерительные системы, использующие точечные маркеры [6,8]. Распознавание точечных объектов на изображениях имеет существенную

особенность: радиус корреляции таких изображений составляет один пиксель. Применение обычных корреляционных методов для распознавания таких изображений невозможно, так как оптические и шумовые искажения разрушают узкие корреляционные пики.

Для идентификации точечных объектов применяют метод трасс и квазикорреляционный метод. Метод трасс предназначен для обработки последовательности кадров, которые соответствуют фазам перемещения объектов или носителя фотокамеры. Пример такой последовательности приведен на рис.9.1.

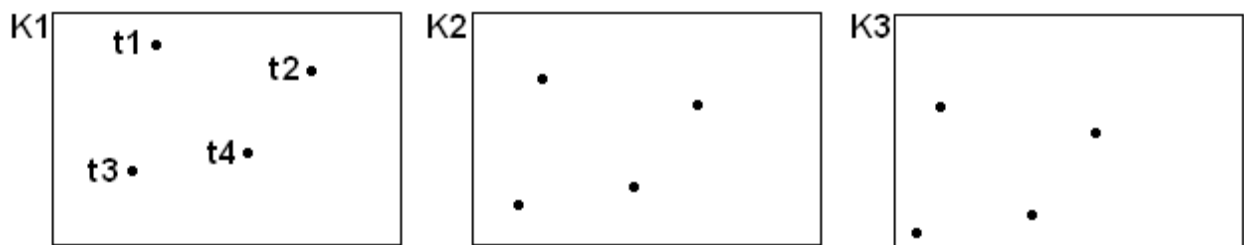


Рис.9.1. Последовательность кадров с точечными объектами

Трассы точечных объектов формируются путем совмещения краев кадров и последующим соединением ряда положений для каждого объекта, как показано на рис.9.2.

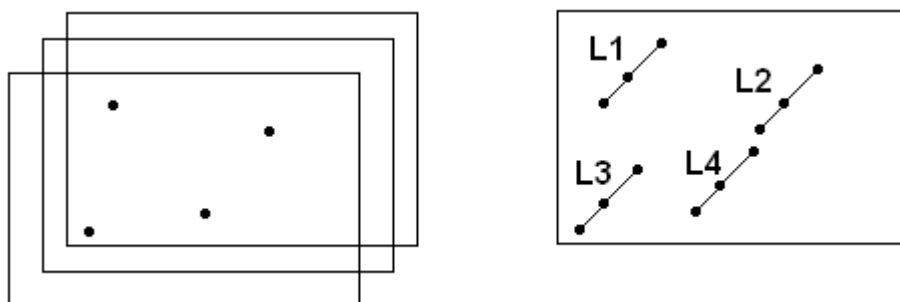


Рис.9.2. Формирование трасс точечных объектов

Полученные трассы позволяют анализировать положение и характер относительного движения точечных объектов. Квазикорреляционный метод

основан на замене точечных элементов эталона (звездной карты) кругами, центры которых совпадают с точечными прообразами. Интенсивность внутри кругов убывает от центра к краям, как показано на рис.9.3 для точек T1-T4.

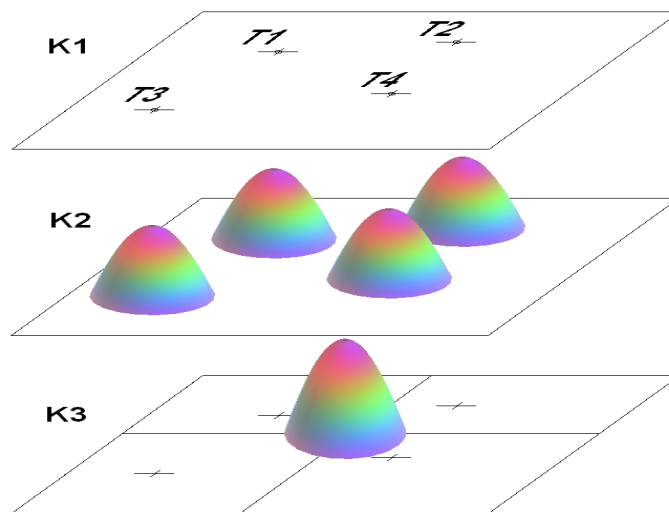


Рис.9.3. Совмещение точечных изображений с помощью квазикорреляционного метода

Текущее изображение K1 остается точечным. Вычисляется величина корреляции между преобразованным эталоном K2 и текущим изображением K1. Квазикорреляционная функция K3 имеет определенный радиус корреляции, что делает процесс сравнения изображений устойчивым и помехозащищенным.

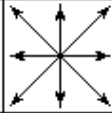
10. Распознавание контурных изображений

Контур является важной частью распознаваемого образа [22]. Человеческое восприятие изображения во многих случаях основано на выделении и анализа именно контура объекта. *Контур* является областями с высокой концентрацией информации, практически не зависящей от цвета и яркости контекста. Распознавание контурных изображений строится на

основе определения параметров формы и ориентации контуров соответствующих объектов.

Кодирование контуров базируется на дискретности цифровых изображений, то есть разбиении раstra на равновеликие квадраты (ячейки) сетью горизонтальных и вертикальных линий. Формат контурных изображений двоичный (черно-белый). Контур Γ , состоящий из k ячеек, может быть определен следующим образом: $\Gamma = \{\gamma(n)\}_{0,k-1}$, где $\gamma(n)$ - элементарный вектор (ЭВ), соединяющий центры соседних контурных ячеек и направленный в сторону обхода контура. Существует 8 возможных направлений элементарных векторов и три способа кодирования этих направлений (рис.10.1): кодирование одним числом (метод Фримена), кодирование с использованием декартовых координат, кодирование с помощью комплексных чисел.

5	6	7
4		0
3	2	1

-1,1	0,1	1,1
-1,0		1,0
-1,-1	0,-1	1,-1

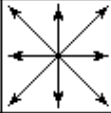
-1+i	i	1+i
-1		1
-1-i	-i	1-i

Рис.10.1. Кодирование элементарных векторов по Фримену, координатное и комплексное кодирование

Линии контуров могут иметь некоторую толщину, что является фактором избыточности и неоднозначности обхода контура. Приведение толщины контура к минимальному единичному значению называется процедурой *скелетирования контура*.

Контурные могут иметь несколько видов особых точек, идентификация которых важна с точки зрения распознавания контурного объекта. Виды

особых точек: тупиковая точка (начало-конец), одинарное ветвление, множественное ветвление, утолщение, самопересечение, внешнее пересечение.

Анализ контурных изображений начинается с обхода контура, который позволяет упорядочить точки контура. Если контур имеет минимальную толщину и у него нет точек ветвления и пересечения, то процедура обхода осуществляется однозначно, если она начата из тупиковой точки для разомкнутого контура. Метод «окрашивания» пройденных точек, позволяет обходить контуры с особыми точками, при этом цвет окрашивания зависит от кратности прохождения точек контура. Контур представляет собой множество пар $\{x_i, y_i\}_{i=0..M-1}$ упорядоченных координат точек. Каждой точке контура можно сопоставить комплексное число [14]:

$$z_i = x_i + j \cdot y_i.$$

Эффективным средством анализа свойств и цифровой фильтрации контуров служит дискретное преобразование Фурье:

$$F_k = \sum_{i=0}^{M-1} z_i \cdot \exp\left(-j \cdot \frac{2\pi}{M} \cdot k \cdot i\right).$$

В результате преобразования получается вектор компонент преобразования Фурье:

$$F = [F_0, F_1, \dots, F_{M-1}].$$

Нулевая составляющая этого вектора представляет собой координаты геометрического центра контура.

Пара компонентов (F_1, F_{M-1}) в обратном преобразовании Фурье соответствует первичному эллипсу контура:

$$\begin{aligned} z_{1k} &= \frac{1}{M} \cdot \left(F_1 \cdot \exp\left(-j \cdot \frac{2\pi k}{M}\right) + F_{M-1} \cdot \exp\left(-j \cdot \frac{2\pi k(M-1)}{M}\right) \right) = \\ &= \frac{1}{M} \cdot \left(F_1 \cdot \exp\left(-j \cdot \frac{2\pi k}{M}\right) + F_{M-1} \cdot \exp\left(j \cdot \frac{2\pi k}{M}\right) \right). \end{aligned}$$

Этот эллипс соответствует максимально сглаженному исходному контуру, и угол наклона его главной оси отображает ориентацию контура на плоскости:

$$\varphi_1 = -\frac{1}{2} \cdot (\arg(F_1) + \arg(F_{M-1})).$$

Определение размеров осей и угла наклона первичного эллипса позволяет нормализовать исходное изображение к параметрам эталонных образцов и осуществить классификацию одним из стандартных методов. Пример вычисления первичного эллипса приведен на рис.10.2. На данном рисунке исходная фигура состоит из 620 точек и образована четырьмя линиями Безье (сплошные линии). Первичный эллипс (пунктирная линия) охватывает основную часть фигуры.

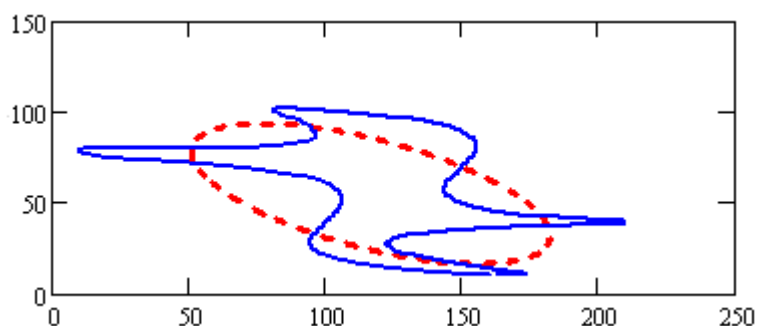


Рис.10.2. Исходный контур и первичный эллипс

Усеченный ряд Фурье обладает хорошими аппроксимирующими свойствами. На рис.10.3 приведен пример аппроксимации исходного контура с помощью первых 7-ми пар гармоник дискретного ряда Фурье (пунктирная линия). Исходный контур имеет несколько изломов и крутых изгибов, тем не менее, аппроксимирующая линия достаточно хорошо их повторяет. Использование для аппроксимации 20 гармоник делает различия между контурами почти неразличимыми.

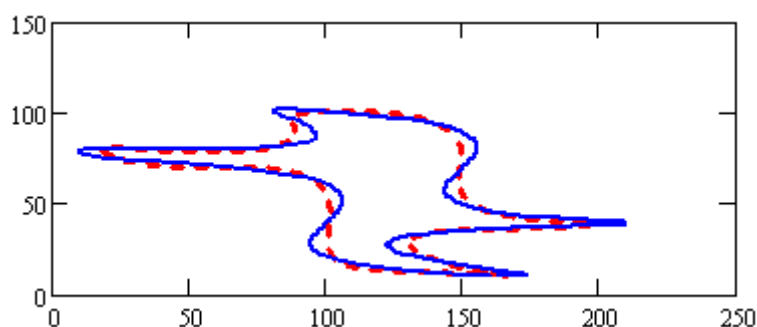


Рис.10.3. Исходный контур и контур, образованный первыми 7-ю парами гармоник дискретного ряда Фурье.

Спектральное представление контура является эффективным средством выделения компактного множества признаков, которые можно использовать на этапе формализации множества контуров для последующего их сравнения.

Контуров большинства реальных объектов формируются на основе растровых изображений. В этом случае линии контуров определяются как границы между областями изображения с однородной яркостью. Выделение контуров осуществляется путем применения специальных растровых фильтров.

Реальные изображения могут содержать большое количество контурных линий, обозначающих границы и структуру различных объектов, поэтому существует проблема идентификации контурных границ объектов, для решения которой используется априорная информация о геометрических свойствах объектов.

11. Модель живого нейрона

Кора головного мозга человека состоит из $\sim 3 \cdot 10^{10}$ нейронов с высокой степенью межнейронных связей. Каждый нейрон связан примерно с 10^4 других нейронов. Внутренность нейрона отделена от внешней среды плазматической мембраной. Различная концентрация положительных и отрицательных ионов внутри и вне нейрона создает разность потенциалов около 87 мВ. Нервы и мышцы способны генерировать электрические импульсы

одинаковой амплитуды с длительностью 1 мс. Энергетические затраты мозга [18] составляют 10^{-16} Дж. на операцию в секунду, в то же время ЭВМ затрачивает 10^{-6} Дж. Информация кодируется в виде количества и частоты этих импульсов.

Нейрон состоит из следующих частей: 1) тела (сомы); 2) синапсов – тонких отростков для связи с другими нейронами (3-100 мкм); 3) дендритов – коротких отростков, с которыми образуют контакт синапсы; 4) аксонов – длинных коммуникационных отростков, передающих импульсы от нейрона к мышцам и от органов чувств к нейрону; 5) глиальных (швановских) клеток – спирально закрученных клеток, покрывающих большую часть поверхности нейрона; 6) перехватов Ранвье – узкие (1 мкм) области контакта мембраны нейрона с окружающей средой; 7) разветвляющихся окончаний в мышечных волокнах. Схематичное изображение нейрона приведено на рис.11.1.

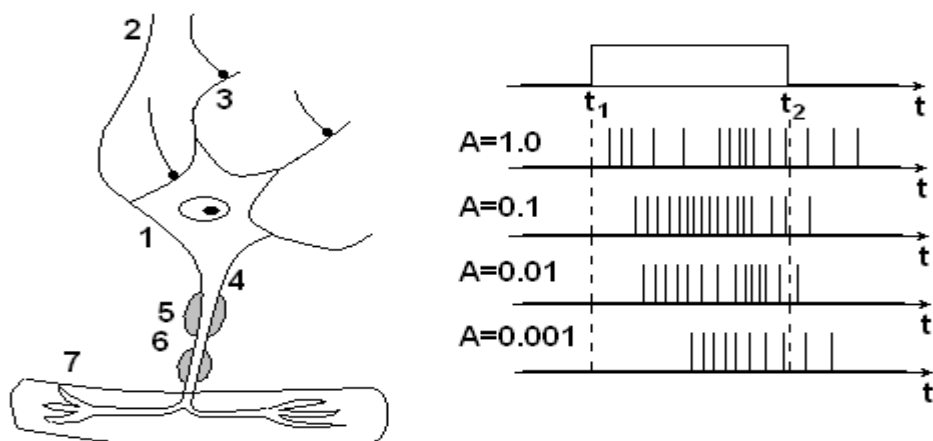


Рис.11.1. Структура живого нейрона и диаграмма импульсных последовательностей, генерируемых нейроном

Хорошей иллюстрацией работы нейрона служат диаграммы импульсных последовательностей, генерируемых нейроном. Диаграммы, демонстрирующие характер изменения импульсных последовательностей, формируемых зрительным нервом, приведены на рис.11.1. В качестве изменяемого параметра

для этих диаграмм использована интенсивность световой вспышки A , последовательно уменьшаемая в 10 раз.

12. Математическая модель нейрона и нейронных сетей

Математическая модель нейрона была создана на основе представления нейрона в виде элемента, суммирующего сигналы, поступающие в него через синаптические связи [2,3,4,16,18]. Модель предложена американскими учеными МакКаллоком (McCulloch) и Питцем (Pitt) в 1943 году. Каждому синапсу приписывается определенный коэффициент передачи (вес) электрических сигналов нервной системы. Нелинейные свойства нейрона моделируются с помощью специальной функции, которую часто называют “активационной”. Простейшая модель нейрона приведена на рис.12.1.

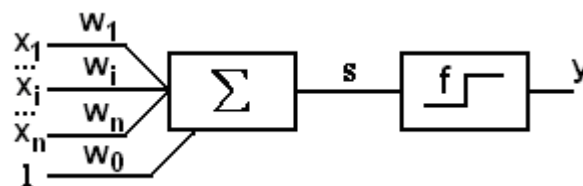


Рис.12.1. Модель нейрона

Вектор входного сигнала $X = (x_1, x_2, \dots, x_n)^T$ скалярно умножается на вектор весовых (синаптических) коэффициентов $(w_1, w_2, \dots, w_n)^T$ и к результату прибавляется смещение w_0 . Сигнал синаптической суммы s поступает на вход блока активационной функции $f(s)$. Активационная функция исполняет роль ограничителя выходного сигнала, что присуще биологическим системам. Она определяет вид нелинейности характеристик нейрона, расширяет его функциональные возможности. Математические зависимости модели нейрона имеют следующий вид:

$$s = \sum_{i=1}^n w_i \cdot x_i + w_0, \quad y = f(s).$$

В качестве активационной функции используются либо ступенчатые функции, либо непрерывные функции. Первый тип функций проще реализуется, а второй необходим для аналитических исследований свойств нейронов. Одна из возможных форм задания непрерывной активационной функции (sigmoid function) и ее первой производной:

$$f(s) = \frac{2}{1 + \exp(-s)} - 1, \quad f'(s) = \frac{2 \cdot \exp(-s)}{(1 + \exp(-s))^2}.$$

График активационной функции представлен на рис.12.2.

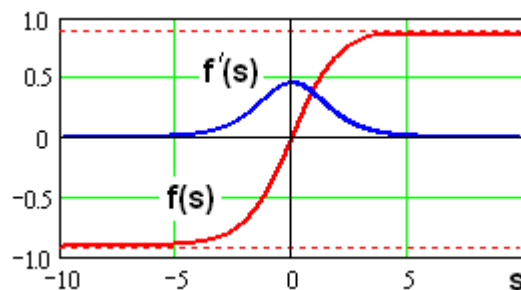


Рис.12.2. Непрерывная активационная функция

Известна [18] стохастическая модель нейрона, которая определяется с помощью ступенчатой активационной функции путем введения определенной вероятности активации нейрона (probability of firing) $P(s)$ и имеет следующий вид:

$$f(s) = \begin{cases} +1, & \text{с_вероятностью : } P(s); \\ -1, & \text{с_вероятностью : } 1 - P(s). \end{cases}$$

Множество связанных нейронов образуют нейронные сети (НС). НС в зависимости от вида основных элементов можно классифицировать следующим образом [10]:

1. По структуре сети: сети полносвязанные (fully connected), неполносвязанные (partially connected), сети слоистые, сети специальной структуры.
2. По типу взаимодействия нейронов: сети с парным взаимодействием, сети с множественным взаимодействием, сети с взаимодействием нейрона с самим собой, сети с симметричным и несимметричным взаимодействием.
3. Сети с дискретными и непрерывными состояниями нейронов.
4. Сети с дискретным и непрерывным временем.
5. Сети с детерминированной или стохастической динамикой, сети синхронного, параллельного или последовательного действия.
6. Сети с различными функциями возбуждения нейронов.
7. Сети с различными алгоритмами обучения и самообучения.

Нейронные сети могут обучаться, то есть изменение весовых коэффициентов по определенному правилу позволяет получить на выходе НС требуемые сигналы для определенных входных сигналов. Обучение представляет собой рекуррентную процедуру, в процессе которой образы обучающей выборки несколько раз подаются на вход НС. Порядок подачи может быть циклическим или случайным. Начальные значения должны быть ненулевыми, рекомендуется присваивать им случайные значения. Пример многослойной НС приведен на рис.12.3. НС состоит из k -слоев, имеет m -входных линий и n -выходных элементов.

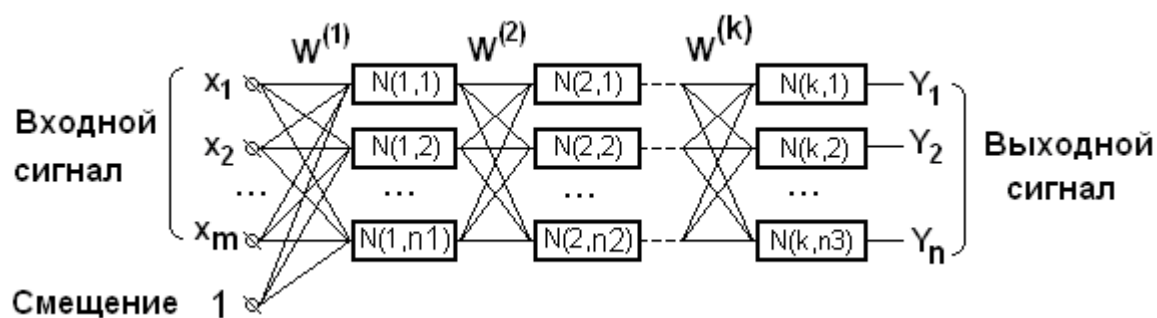


Рис.12.3. Многослойная полносвязанная НС.

Входные образы НС принято представлять в виде векторов. На вход НС можно подавать образы, соответствующие изображениям, звукам и наборам измерительной информации. Дискретные изображения структурно соответствуют матрицам, поэтому перед подачей на вход НС изображения векторизуют, то есть преобразуют в вектор путем последовательного перемещения столбиков один под другим в один общий столбик вектора. Единичный сигнал смещения можно считать последним элементом входного вектора X , а соответствующие весовые коэффициенты станут нижней строкой матрицы весовых коэффициентов $W^{(1)}$.

Математическая модель НС, приведенной на рис.12.3, основана на матрицах весовых коэффициентов и имеет следующий вид:

$$Y = \tilde{f}(W^{(k)} \cdot \tilde{f}(\dots \cdot \tilde{f}(W^{(2)} \cdot \tilde{f}(W^{(1)} \cdot X))\dots)),$$

где $\tilde{f}(v)$ - оператор поэлементного вычисления активационной функции $f(v_j)$ для элементов векторного аргумента v . Матрицы весовых коэффициентов объединяют нейроны слоя: каждой строке матрицы соответствует набор весовых коэффициентов соответствующего нейрона. Если НС является полносвязанной, то перестановка строк в матрицах внутренних слоев не изменит выходной вектор.

Нейронные сети являются универсальными аппроксиматорами [2], так как на них распространяется теорема Колмогорова-Арнольда: непрерывная функция нескольких переменных может быть представлена в виде суперпозиций и операций сложения функций одной переменной.

13. Алгоритм обучения персептрона

Однослойные НС иногда называют персептронами. Первые опыты с НС состояли в распознавании изображений цифр. В качестве входных сигналов

$\{x_i\}$ использовались сигналы матрицы фотоприемников. Весовые коэффициенты $\{w_{i,j}\}$ соответствовали регулируемым коэффициентам усиления фотоприемников. Каждому выходному каналу НС y_i сопоставлялась одна из распознаваемых цифр Ω_i , то есть выходной сигнал должен был равняться единице только в случае предъявления персептрону строго определенной цифры.

Настройка весовых коэффициентов осуществлялась по специальному алгоритму обучения, который состоял в следующем:

- Подать входной образ Ω_i и вычислить выходной сигнал y_i .
- Если выходной сигнал Y верен, то коррекция весовых коэффициентов W не проводится.
- Если выходной сигнал Y не верен, и $y_i = 0$, то к вектору весовых коэффициентов $\{w_{i,j}\}_{i=\text{const}}$ прибавить вектор входных сигналов X .
- Если выходной сигнал не верен, и $y_i = 1$, то из вектора весовых коэффициентов $\{w_{i,j}\}_{i=\text{const}}$ вычесть вектор входных сигналов X .

Структура обучающей системы персептрона показана на рис.13.1.

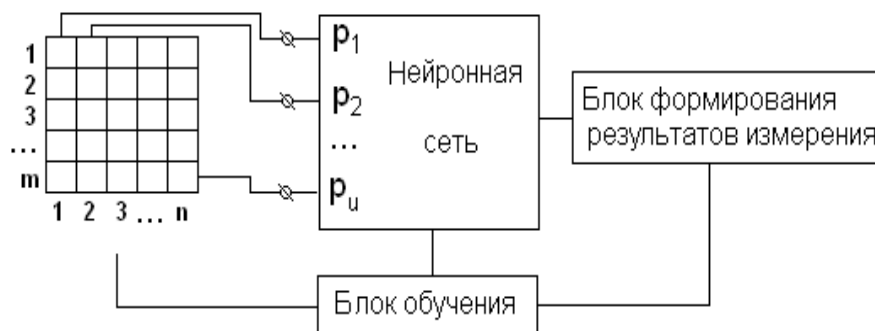


Рис.13.1. Структура персептрона, обрабатывающего изображения.

Аналитическая форма алгоритма:

$$\varepsilon = T - A,$$

$$W(n+1) = W(n) + \eta \cdot \varepsilon \cdot X,$$

где ε - вектор ошибки выходного сигнала, T - требуемый (правильный) выходной сигнал, A - фактический выходной сигнал.

Простота настройки персептрона позволила применять эти устройства на ранней стадии развития электроники (1930 г.) для распознавания почтовых индексов на почтовых конвертах.

14. Топологические свойства однослойных, двухслойных и трехслойных НС. Теорема Минского

Количество слоев НС определяет ее возможности по распознаванию изображений. Если рассматривать в качестве входного вектора НС координаты некоторой точки плоскости, то каждый нейрон однослойной сети может определить факт принадлежности точки определенной полуплоскости.

Будем считать, что (x, y) – координаты точки плоскости и одновременно входные сигналы нейрона. Соответствующая схема нейрона представлена на рис.14.1.

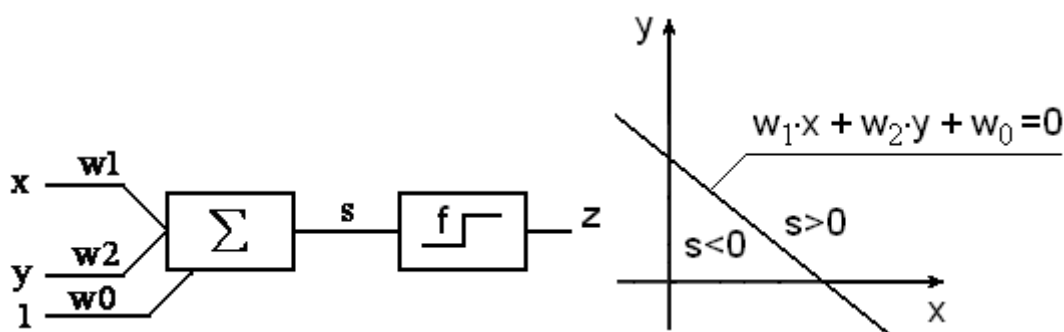


Рис.14.1. Нейрон и соответствующая разделительная линия двух полуплоскостей

Сигмоидальная сумма определяется по следующей формуле:

$$s = w_1 \cdot x + w_2 \cdot y + w_0.$$

Соответствующее каноническое уравнение прямой, являющейся границей двух полуплоскостей, имеет вид:

$$w_1 \cdot x + w_2 \cdot y + w_0 = 0.$$

В одной полуплоскости сигмоидальная сумма $s > 0$, а в другой $s < 0$, что отображено на рис.14.1.

Известно несколько теорем об ограниченности функциональных возможностей НС. Одной из таких теорем является теорема Минского.

Теорема Минского 1: Однослойный персептрон не может воспроизвести функцию «исключающее или».

Рассмотрим доказательство теоремы, построенное на геометрической интерпретации входных сигналов персептрона и логической функции. Персептрон должен иметь два входа, что соответствует двум аргументам функции «исключающее ИЛИ». Таблица истинности этой функции показана на рис.14.2.

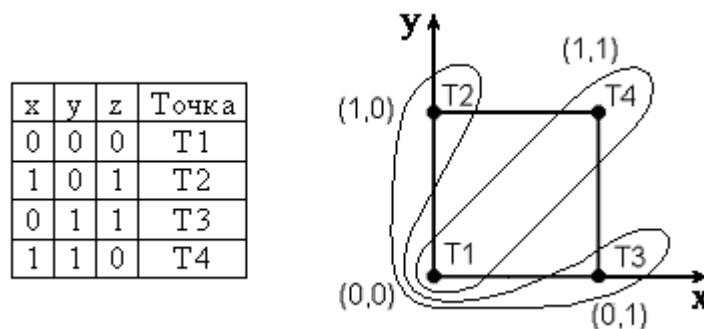


Рис.14.2. Множества, которые соответствуют функции “исключающее ИЛИ”

Будем считать, что x, y – координаты точек плоскости, имена которых указаны в правом столбике таблицы истинности. Точки T1 и T4 должны принадлежать множеству, которое соответствует значениям функции $z=1$, а точки T2 и T3 – множеству $z=0$. Один из вариантов конфигурации таких множеств приведен на рис.14.2.

На вход персептрона подаются координаты (x, y) , которые умножаются на весовые коэффициенты w_1 и w_2 , а затем складываются со смещением w_0 :

$$s(x, y) = w_1 \cdot x + w_2 \cdot y + w_0.$$

Полученная синаптическая сумма преобразуется активационной функцией в выходной сигнал $z \in \{0,1\}$. Выше было показано, что персептрон осуществляет линейное разделение точек плоскости, однако, как видно из рисунка, пары точек $\{T1, T4\}$ и $\{T2, T3\}$ невозможно разделить прямой. Таким образом, персептрон не способен воспроизвести функцию «исключающее или».

Двухслойная НС может решить более сложные задачи, комбинируя во втором слое нейронов выходные сигналы нейронов первого слоя. Для задачи идентификации положения точки этот тезис можно интерпретировать следующим образом: выходной сигнал нейрона второго слоя определяет факт принадлежности точки выпуклого многоугольника, граничные линейные участки которого определяются нейронами первого слоя. Например, для разделения множеств Ω_1 и Ω_2 , показанных на рис.14.3, необходима двухслойная НС, первый слой которой должен состоять из трех нейронов, а второй – из одного.

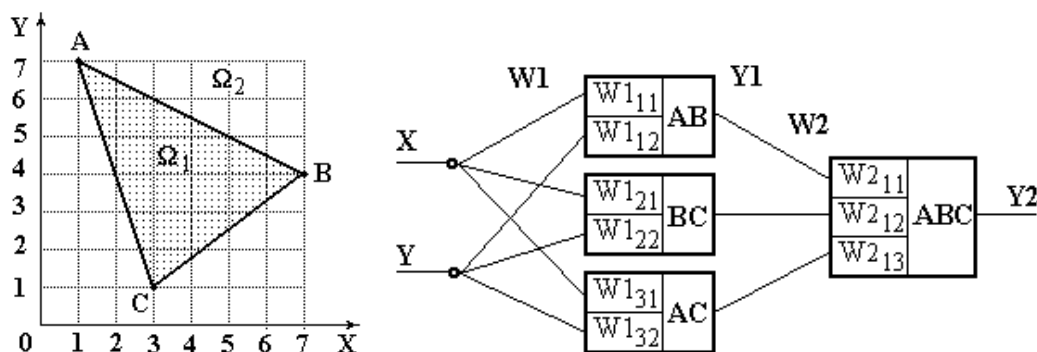


Рис.14.3. Области классификации точек плоскости и соответствующая двухслойная нейронная сеть, идентифицирующая точки плоскости

Весовые коэффициенты первого слоя рассчитываются на основе геометрических представлений и следующих аналитических соотношений:

$$\left\{ \begin{array}{l} \frac{X - X_A}{X_B - X_A} = \frac{Y - Y_A}{Y_B - Y_A} \\ \frac{X - X_A}{X_C - X_A} = \frac{Y - Y_A}{Y_C - Y_A} \\ \frac{X - X_B}{X_C - X_B} = \frac{Y - Y_B}{Y_C - Y_B} \end{array} \right. , \Rightarrow Q(X, Y) = W1 \cdot \begin{bmatrix} X \\ Y \end{bmatrix} + V1 \geq 0 .$$

Из приведенных соотношений видно, что матрица весовых коэффициентов $W1$ и вектор смещений $V1$ могут быть получены в результате трансформации системы уравнений для прямых, проходящих через заданные точки плоскости. Важным моментом является переход от системы уравнений к системе неравенств, который осуществляется для каждого равенства так, чтобы точки, принадлежащие выпуклому многоугольнику, обеспечивали компонентам функции $Q(X, Y)$ одинаковые знаки, для определенности – положительные. Из аналитической геометрии известно, что если прямая задается уравнением: $aX + bY + c = 0$, то вектор (a, b) направлен в сторону положительной полуплоскости, образуемой этой прямой. Таким образом, если вектор (a, b) направлен во внешнюю сторону многоугольника, то для получения требуемого неравенства необходимо изменить знаки коэффициентов: $-aX - bY - c > 0$, в противном случае изменения знаков не требуется: $aX + bY + c > 0$.

Для примера, приведенного на рис.14.3, весовые коэффициенты нейронов первого слоя определяются следующим образом:

$$\begin{array}{l} \text{AB: } \begin{cases} x + 2y - 15 = 0 \\ 3x + y - 10 = 0 \end{cases} \Rightarrow \begin{cases} -x - 2y + 15 > 0 \\ 3x + y - 10 > 0 \end{cases} \\ \text{AC: } \begin{cases} 3x + y - 10 = 0 \\ 3x - 4y - 5 = 0 \end{cases} \Rightarrow \begin{cases} 3x + y - 10 > 0 \\ -3x + 4y + 5 > 0 \end{cases} \end{array} , \Rightarrow W1 = \begin{bmatrix} -1 & -2 \\ 3 & 1 \\ -3 & 4 \end{bmatrix}, V1 = \begin{bmatrix} 15 \\ -10 \\ 5 \end{bmatrix}.$$

Результирующий сигнал $S1_i = W1_{i1} \cdot X + W1_{i2} \cdot Y + V1_i$ преобразуется с помощью активационной функции, приведенной на рис. 14.4. Если классифицируемая точка попадает в положительную полуплоскость, то

выходной сигнал $f(S1_i)=1$. Точку можно считать принадлежащей заданной области, только в том случае, если все выходные сигналы равны 1. Весовые коэффициенты нейрона второго слоя должны быть подобраны так, чтобы выполнялось неравенство: $W2_{11} \cdot 1 + W2_{12} \cdot 1 + W2_{13} \cdot 1 + V2_1 \geq 0$. Например, $W2 = [1 \ 1 \ 1]$, $V2 = [-2.5]$.

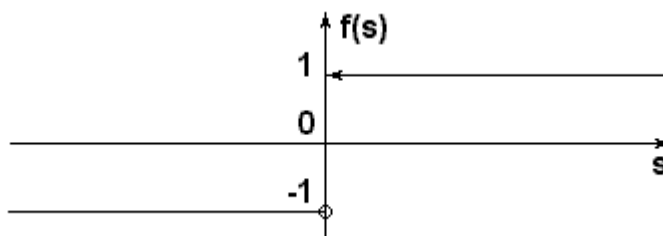


Рис. 14.4. Ступенчатая активационная функция

Трехслойные НС позволяют распознавать области плоскости, форма которых не ограничена требованием выпуклости, так как она образуется логической комбинацией выпуклых многоугольников. На рис.14.5 представлены примеры таких фигур.

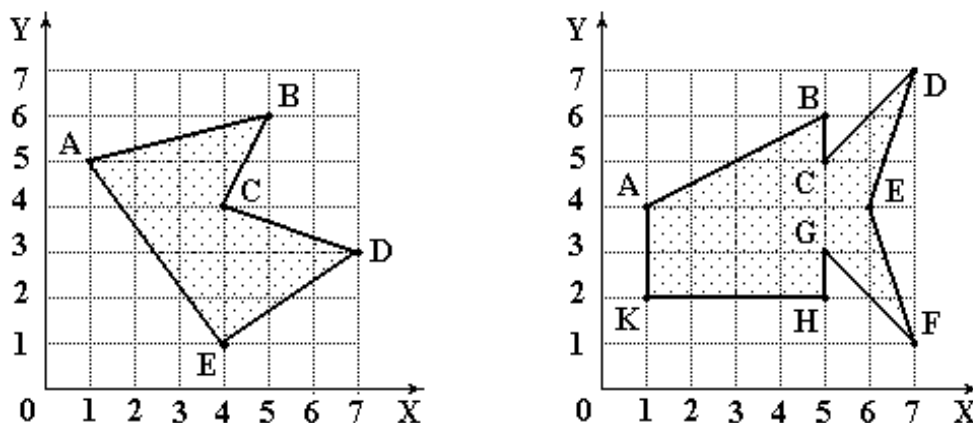


Рис.14.5. Распознаваемые области

Точки A, B и D лежат на одной прямой, и это позволяет ограничить число нейронов в первом слое до пяти. Если бы продолжение прямой CD проходило выше точки A, то НС пришлось бы дополнить шестым нейроном.

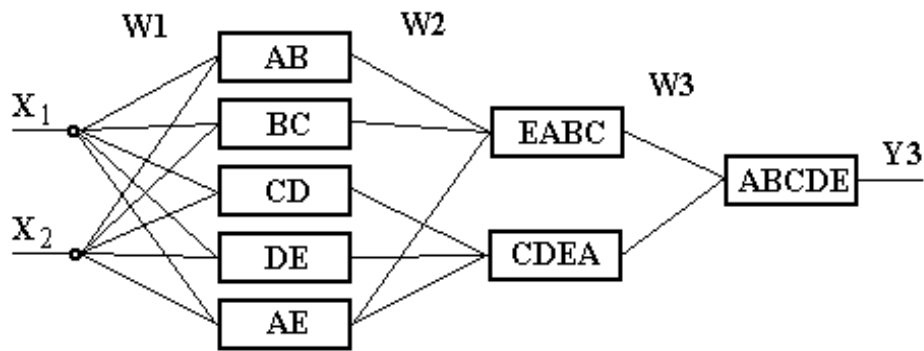


Рис.14.6. Пример трехслойной нейронной сети

Первый слой НС определяет множество полуплоскостей, второй слой нейронов формирует выпуклые многоугольники, а третий слой комбинирует многоугольники для образования требуемых фигур, не обладающих свойством выпуклости.

15. Свойства НС Хопфилда

НС Хопфилда является однослойной НС, предназначенной для распознавания бинарных образов [2,18]. Сеть охвачена обратной связью (feedback loop), как показано на рис.15.1. Каждая линия обратной связи содержит элемент задержки на 1 такт времени z^{-1} (unit delay element). Обратная связь элементов типа «связь самого с собой» в сети не используется.

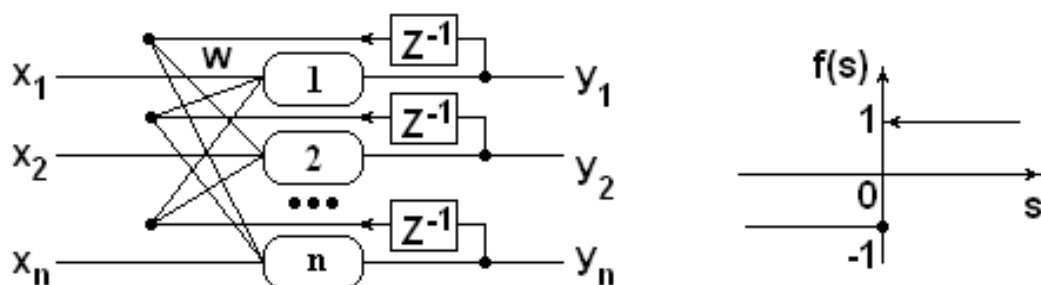


Рис.15.1. НС Хопфилда

Сеть используется в качестве ассоциативной памяти. Сеть должна уметь на основе двоичного неидеального сигнала на входе сформировать сигнал

идентификации или отказа на выходе. Весовые коэффициенты сети вычисляются на основе эталонной информации, которая имеет формат входной информации. $X^{(k)} = (x_1^k, x_2^k, \dots, x_n^k)^T$ - где $k = 0..m-1$, m - количество эталонов. Если $Y = X^{(k)}$, то сигнал распознан.

$$w_{i,j} = \begin{cases} \sum_{k=0}^{m-1} x_i^k \cdot x_j^k & \text{при } i \neq j \\ 0 & \text{при } i = j \end{cases}. \quad (15.1)$$

Число запоминаемых образов должно удовлетворять ограничению: $m \leq n/7$. Кроме того, эталоны попарно должны иметь существенные корреляционные отличия. Появление на входе системы нового сигнала вызывает переходный процесс, который описывается следующими векторными соотношениями:

$$Y(0) = X, \quad s(p+1) = W \cdot Y(p), \quad Y(p+1) = f(s(p+1)).$$

Возможными результатами переходного процесса могут являться:

- Одно из эталонных изображений $Y = X^{(k)}$.
- Негатив одного из эталонных изображений $Y = -X^{(k)}$.
- Зацикливания длины 2 некоторых битов выходного сигнала – отказ от распознавания.

Рассмотрим пример распознавания бинарных изображений с помощью НС Хопфилда. Даны два образца изображений:

$$Z1 = \begin{pmatrix} -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{pmatrix}, \quad Z2 = \begin{pmatrix} -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

и входное, распознаваемое изображение

$$Z3 = \begin{pmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & -1 \end{pmatrix}.$$

В этом случае $m=2$, $n=16$, т.е. выполняется неравенство $2 \leq 16/7 \approx 2.29$, возможно использование НС Хопфилда.

Данные матрицы преобразуем по столбцам в вектора:

$$Z1 = [-1 \ 1 \ 1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 1 \ 1 \ -1]^T,$$

$$Z2 = [-1 \ 1 \ -1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 1]^T,$$

$$Z3 = [1 \ 1 \ 1 \ 1 \ -1 \ 1 \ 1 \ -1 \ 1 \ 1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1]^T.$$

Матрица весовых коэффициентов сети:

$$W_{25}^2 = \begin{array}{c|cccccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ \hline 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 & -2 & 2 & 2 & -2 & -2 & 2 & 2 & -2 & -2 & 0 & 0 & 2 & -2 \\ 3 & 0 & 0 & -2 & 0 & -2 & -2 & 2 & 2 & -2 & -2 & 2 & 2 & 0 & 0 & -2 & 2 \\ 4 & 0 & 0 & 2 & -2 & 0 & 2 & -2 & -2 & 2 & 2 & -2 & -2 & 0 & 0 & 2 & -2 \\ 5 & 0 & 0 & 2 & -2 & 2 & 0 & -2 & -2 & 2 & 2 & -2 & -2 & 0 & 0 & 2 & -2 \\ 6 & 0 & 0 & -2 & 2 & -2 & -2 & 0 & 2 & -2 & -2 & 2 & 2 & 0 & 0 & -2 & 2 \\ 7 & 0 & 0 & -2 & 2 & -2 & -2 & 2 & 0 & -2 & -2 & 2 & 2 & 0 & 0 & -2 & 2 \\ 8 & 0 & 0 & 2 & -2 & 2 & 2 & -2 & -2 & 0 & 2 & -2 & -2 & 0 & 0 & 2 & -2 \\ 9 & 0 & 0 & 2 & -2 & 2 & 2 & -2 & -2 & 2 & 0 & -2 & -2 & 0 & 0 & 2 & -2 \\ 10 & 0 & 0 & -2 & 2 & -2 & -2 & 2 & 2 & -2 & -2 & 0 & 2 & 0 & 0 & -2 & 2 \\ 11 & 0 & 0 & -2 & 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 & 0 & 0 & 0 & -2 & 2 \\ 12 & 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\ 13 & -2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 14 & 0 & 0 & 2 & -2 & 2 & 2 & -2 & -2 & 2 & 2 & -2 & -2 & 0 & 0 & 0 & -2 \\ 15 & 0 & 0 & -2 & 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 & 2 & 0 & 0 & -2 & 0 \end{array} \quad \dots$$

Формирование i -го элемента выходного сигнала происходит следующим образом: i -я строка матрицы скалярно умножается на вектор входного сигнала. Полученное значение подставляется в активационную функцию. Например, первый элемент выходного вектора равен:

$$Y1_1 = f(0 \cdot 1 + (-2) \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot (-1) + 0 \cdot 1 + \dots) = f(-2) = -1.$$

Выходной вектор НС имеет вид:

$$Y1 = [-1 \ -1 \ 1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ 1 \ -1]^T.$$

Второй и третий выходные сигналы рассчитываются аналогично, но в качестве входного сигнала используется предыдущий выходной сигнал.

$$Y2 = [1 \ 1 \ 1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1 \ -1 \ -1 \ 1 \ -1]^T,$$

$$Y3 = [-1 \ -1 \ 1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ 1 \ -1]^T.$$

Третий выходной сигнал в точности повторяет первый, это говорит о заклипании нейронной сети.

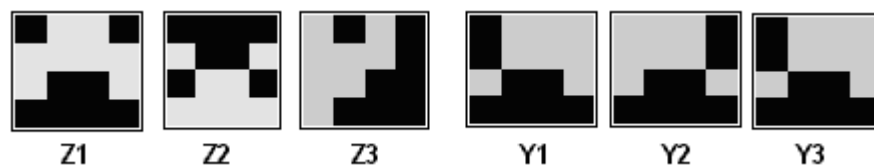


Рис. 15.2. Два образцовых изображения Z1, Z2, входной вектор Z3, выходные вектора Y1, Y2, Y3.

Приведенный пример расчета показывает, что переходной процесс формирования выходного сигнала был коротким, что свойственно для НС Хопфилда.

16. Свойства НС Хемминга

НС Хэмминга формирует на выходе номер распознанного эталона [2,18]. Сеть состоит из двух слоев, с обратной связью во втором слое, как показано на рис.16.1. Сеть выбирает только один эталон с минимальным расстоянием Хэмминга до входного сигнала и активизирует только один выход сети, соответствующий этому эталону. Первый слой НС формирует на своем выходе вектор, состоящий из коэффициентов корреляции входного образа со множеством запомненных образов. Второй слой определяет наибольший коэффициент корреляции и устанавливает в «1» соответствующую выходную

линию. Второй слой реализует стратегию WTA (winner take all) – «победитель получает все». Входной и выходной слою содержит столько пар нейронов, сколько образов должна запомнить НС Хэмминга.

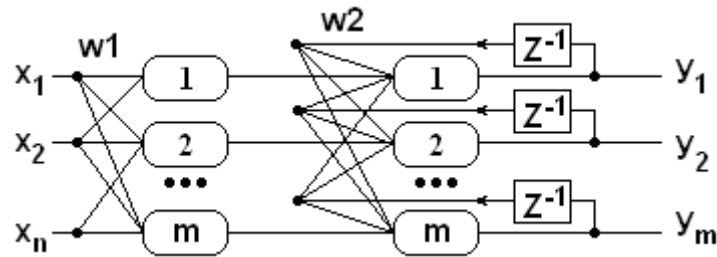


Рис.16.1. НС Хэмминга

Весовые коэффициенты 1-го слоя иницируются следующими значениями: $w_{i,k}^{(1)} = x_i^k / 2$, $B_k = n/2$, где $i=1...m$ - номер эталона, $k=1...n$ - номер входного сигнала. Весовые коэффициенты второго слоя назначаются следующим образом: $0 < w_{i,k}^{(2)} < 1/m$, если $i \neq k$, иначе $w_{i,i}^{(2)} = 1$. Число запоминаемых образов должно удовлетворять ограничению: $m \leq n/7$.

Выходные сигналы первого и второго слоев имеют вид:

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=1}^n W_{j,i}^{(1)} \cdot x_i + T_j, \quad y_j^{(2)}(0) = y_j^{(1)},$$

$$s_j^{(2)}(p+1) = y_j^{(2)}(p) - \varepsilon \cdot \sum_{i=1}^m y_i^{(2)}(p), \quad y_j^{(2)}(p) = f(s_j^{(2)}(p)).$$

Активационная функция первого слоя – линейная функция, второго слоя – положительная линейная ограниченная функция. Активационная функция и матрица второго слоя НС Хэмминга показаны на рис.16.2.

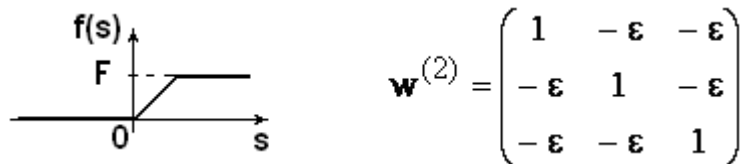


Рис.16.2. Активационная функция и матрица весовых коэффициентов второго слоя

Матрица $w^{(2)}$ имеет отрицательные и положительные элементы. Такая матрица сохраняет наибольший сигнал и ослабляет до нуля меньшие сигналы. Из каждого сигнала вычитается $1/m$ часть других. Максимальный сигнал вносит больший вклад в уменьшение других сигналов, поэтому другие сигналы через несколько шагов обращаются в ноль (отрицательные значения исключены активационной функцией).

17. Синхронные параллельные и последовательные НС

Рассмотрим НС с парным взаимодействием [5]. Действие нейрона самого на себя будем считать отсутствующим. Фрагмент такой НС показан на рис. 17.1.

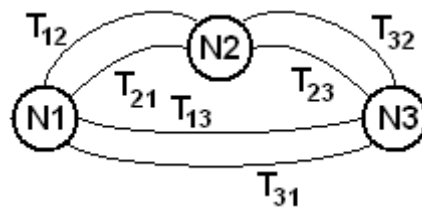


Рис.17.1. Фрагмент НС с парным взаимодействием

В начальный момент сеть характеризуется вектором начальных состояний: $X(0) = (X_i(0))$. Процесс дальнейшего изменения состояний нейронов определяется следующим уравнением:

$$X_i(t+1) = \begin{cases} g\left(\sum_j T_{i,j} \cdot X_j(t) - U_i\right), & \text{если } i \in S(t), \\ X_i(t), & \text{если } i \notin S(t) \end{cases}$$

где $g(\cdot)$ - активационная функция, U - вектор потенциала активации.

Динамические свойства НС определяют одним из следующих правил:

- Синхронная динамика – каждый момент времени активны все нейроны: $i = \overline{1, N}$.

- Параллельная динамика – множество активных нейронов зависит от времени и определяется специальной функцией: $i \in S(t)$.
- Последовательная динамика – нейроны работают по одному; номер активного нейрона $i = K$ определяется по некоторому правилу, например, случайно или в зависимости от величины выходных сигналов в текущий момент времени.

Состояния НС дискретны или непрерывны, если дискретна или непрерывна активационная функция. НС с симметричными связями $T_{i,j} = T_{j,i}$, дискретными состояниями и синхронной динамикой может иметь группы зацикленных нейронов; длина зацикливания ограничена числом состояний нейрона.

Последовательная динамика исключает зацикливания состояний нейронов. Параллельная динамика в общем случае уменьшает количество зацикленных нейронов, но полностью их исключить не способна.

НС с непрерывными состояниями в зависимости от вида динамики и крутизны активационной функции может иметь следующие виды предельных ($t \rightarrow \infty$) состояний: стационарное состояние, периодический режим, квазипериодический режим и хаотический режим. Модель НС с непрерывным временем задают с помощью следующих уравнений:

$$\frac{dH_i}{dt} = \sum_j T_{i,j} \cdot X_j + I_i - \frac{H_i}{\tau_i}, \quad X_i = g(H_i),$$

где H_i - линейный выход нейрона, X_i - нелинейный выход, I_i - внешний сигнал, τ_i - инерционность нейрона. Предельные состояния такой НС всегда стационарны.

18. Метод обратного распространения ошибки

Обучение нейронной сети методом градиентного поиска, с учетом слоистой структуры, называется *методом обратного распространения ошибки* (МОРО – back propagation algorithm). Данный метод хорошо изучен (Вербос (Werbos) 1974 год) и достаточно устойчив [2,14,18]. Сигналы ошибки распространяются от выходов НС к ее входам, в направлении обратном прямому распространению сигналов в обычном режиме работы. Согласно методу наименьших квадратов минимизируемой целевой функцией ошибки НС является сигнал ошибки обучения:

$$E(w) = \frac{1}{2} \cdot \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2, \quad (18.1)$$

где $y_{j,p}^{(N)}$ - реальное выходное состояние нейрона j выходного слоя N нейронной сети при подаче на ее входы p -го образа; $d_{j,p}$ - идеальное (желаемое) выходное состояние этого нейрона. Суммирование ведется по всем нейронам выходного слоя и по всем обрабатываемым сетью образам. Минимизация ведется методом градиентного спуска, что означает подстройку весовых коэффициентов следующим образом:

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

где $w_{ij}^{(n)}$ - весовой коэффициент синаптической связи; i - номер нейрона; j - номер входа нейрона; n - номер слоя; η - коэффициент скорости обучения, $0 < \eta < 1$. Производная сложной функции имеет вид:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial s_j} \cdot \frac{\partial s_j}{\partial w_{ij}},$$

где y_j - выход нейрона j ; s_j - взвешенная сумма входных сигналов, то есть аргумент активационной функции. Для производной сигнала ошибки по выходному сигналу имеем:

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

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

Введя новую переменную

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

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

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

Для выходного слоя:

$$\delta_i^{(N)} = (y_i^{(N)} - d_i) \cdot \frac{dy_i}{ds_i}.$$

Теперь можно записать основное выражение для весовых коэффициентов в обобщенном виде:

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

Рассмотренный метод обучения не всегда приводит к правильному решению, что обусловлено рядом причин. Во-первых, в процессе обучения может возникнуть ситуация, когда большие положительные или отрицательные значения весовых коэффициентов сместят рабочую точку на сигмоидах многих нейронов в область насыщения. Малые величины производной от активационной функции приведут к остановке обучения, что парализует НС. Во-вторых, применение метода градиентного спуска не гарантирует, что будет найден глобальный, а не локальный минимум целевой функции. Эта проблема связана еще с одной, а именно – с выбором величины скорости обучения. Занижение скорости обучения приводит к увеличению длительности обучения. С другой стороны, слишком большие коррекции весов могут привести к потере

устойчивости процесса обучения. Несмотря на указанные недостатки, рассмотренный метод считается одним из наиболее эффективных методов обучения нейронных сетей.

19. Матричная форма МОРО

Наиболее просто метод обратного распространения ошибки представляется в матричной форме. Прямое распространение сигнала по НС, состоящей из N слоев, описывается следующей системой уравнений:

$$\begin{aligned} \mathbf{s}^{(1)} &= \mathbf{W}^{(1)} \cdot \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}, & \mathbf{Y}^{(1)} &= f(\mathbf{s}^{(1)}), \dots, \\ \mathbf{s}^{(i)} &= \mathbf{W}^{(i)} \cdot \mathbf{Y}^{(i-1)}, & \mathbf{Y}^{(i)} &= f(\mathbf{s}^{(i)}), \dots, \\ \mathbf{s}^{(N)} &= \mathbf{W}^{(N)} \cdot \mathbf{Y}^{(N-1)}, & \mathbf{Y}^{(N)} &= f(\mathbf{s}^{(N)}). \end{aligned}$$

Вектор входного сигнала наращивается единичной компонентой, так как необходимо учитывать величину смещения в каждом слое нейронов. Обратное распространение сигнала ошибки по НС позволяет вычислить корректирующие коэффициенты с помощью следующих уравнений:

$$\begin{aligned} \delta^{(N)} &= (\mathbf{Y}^{(N)} - \mathbf{U}) \frac{\partial f(\mathbf{s}^{(N)})}{\partial \mathbf{s}^{(N)}}, \dots, \\ \delta^{(i)} &= (\mathbf{W}^{(i+1)})^T \cdot \delta^{(i+1)} \cdot \frac{\partial f(\mathbf{s}^{(i)})}{\partial \mathbf{s}^{(i)}}, \dots, \\ \delta^{(1)} &= (\mathbf{W}^{(2)})^T \cdot \delta^{(2)} \cdot \frac{\partial f(\mathbf{s}^{(1)})}{\partial \mathbf{s}^{(1)}}. \end{aligned}$$

Коррекция весовых коэффициентов НС проводится с помощью следующих уравнений:

$$\begin{aligned} \Delta \mathbf{W}^{(1)} &= -\eta \cdot \delta^{(1)} \cdot [\mathbf{X}^T, 1], \dots, & \mathbf{W}^{(1)} &= \mathbf{W}^{(1)} + \Delta \mathbf{W}^{(1)}, \\ \Delta \mathbf{W}^{(i)} &= -\eta \cdot \delta^{(i)} \cdot \mathbf{Y}^{(i-1)T}, \dots, & \mathbf{W}^{(i)} &= \mathbf{W}^{(i)} + \Delta \mathbf{W}^{(i)}, \end{aligned}$$

$$\Delta W^{(N)} = -\eta \cdot \delta^{(N)} \cdot Y^{(N)T}, \quad W^{(N)} = W^{(N)} + \Delta W^{(N)}.$$

где η – коэффициент скорости обучения.

Приведенные выше уравнения являются компактной формой описания метода обратного распространения ошибки, используемого для обучения нейронных сетей. Применение данных уравнений позволяет строить модели НС с произвольным числом слоев.

Пример зависимости невязки от величины весовых коэффициентов приведен на рис.19.1. Алгоритм построения графиков: трехслойная НС была обучена распознаванию заданных образов, произвольным образом во втором и третьем слоях выбраны два весовых коэффициента, в квадратной окрестности, окружающей оптимальные значения коэффициентов, задана сеть значений коэффициентов, для которых вычислена невязка.

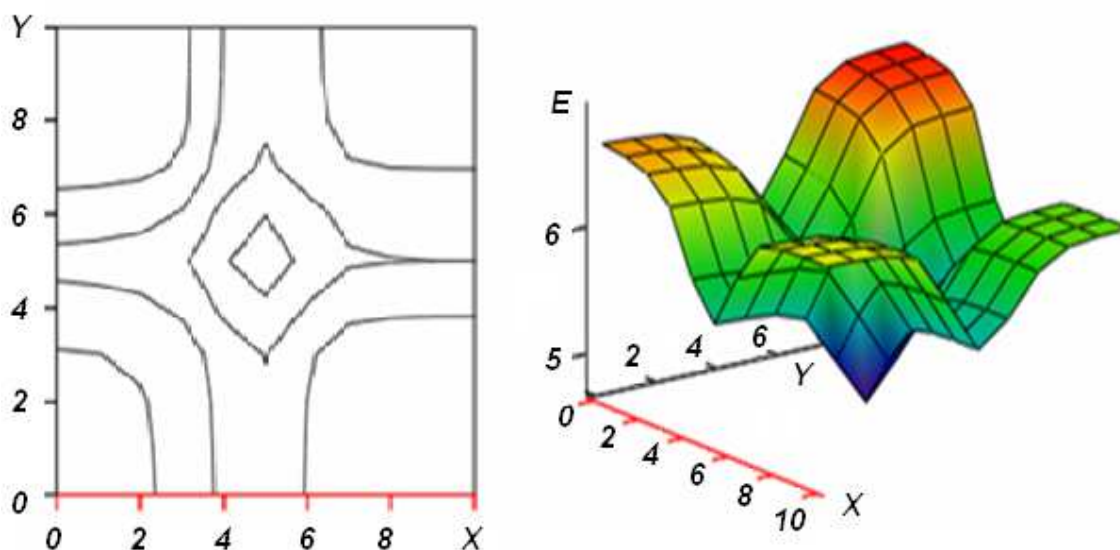


Рис.19.1. Графики зависимости невязки на выходе НС от значений двух весовых коэффициентов

Приведенные графики наглядно показывают наличие минимума невязки и демонстрируют сложность исследуемой поверхности: к точке минимума сходится 4 взаимно ортогональных канала.

20. Самообучение НС методом Хебба

Алгоритм Хебба позволяет обучать НС без контроля правильности ответов. Алгоритм основан на принципе усиления связи между возбужденными нейронами. Сигнальный метод обучения Хебба корректирует весовые коэффициенты НС в соответствии со следующими формулами:

$$\delta_{i,j}^{(n)} = y_i^{(n-1)} \cdot y_j^{(n)}, \quad W_{i,j}^{(n)}(t) = W_{i,j}^{(n)}(t-1) - \eta \cdot \delta_{i,j}^{(n)},$$

где $y_j^{(n)}$ - выходной сигнал обучаемого нейрона, $y_i^{(n-1)}$ - выходной сигнал нейрона предыдущего слоя, который является одним из входных сигналов обучаемого нейрона, t - номер шага обучения. Из формулы видно, что наибольшие изменения претерпит та синаптическая связь, которая соединяет большие входные и выходные сигналы. В матричной форме данный алгоритм можно записать в следующем виде:

$$\delta_P^{(1)} = X_P \cdot Y_P^{(1)T}, \quad \Delta W^{(1)} = -\eta \cdot \sum_P \delta_P^{(1)},$$

$$W^{(1)}(t) = W^{(1)}(t-1) + \Delta W^{(1)}, \dots,$$

$$\delta_P^{(n)} = Y_P^{(n-1)} \cdot Y_P^{(n)T}, \quad \Delta W_P^{(n)} = -\eta \cdot \sum_P \delta_P^{(n-1)},$$

$$W^{(n)}(t) = W^{(n)}(t-1) + \Delta W^{(n)}.$$

Нейронная сеть Хебба самостоятельно определяет количество классов, на которые она разбивает поток обучающей выборки. Количество выходных линий НС должно браться с запасом, поскольку заранее количество классов неизвестно, при этом следует учитывать, что классы кодируются на выходе позиционным двоичным кодом.

Пример. Создана имитационная модель самообучаемой НС со следующими параметрами: количество слоев – 3, количество входов – 60, количество нейронов по слоям – 16, 12, 12, число эталонов – 9, количество циклов обучения – 300, скорость обучения – 0.001. В качестве обучающей выборки были взяты 9 изображений трех цифр, приведенные на рис.20.1.

Разрешение изображений: 10х6. Изображения каждой цифры отличаются за счет наложенных искажений. Классификация данных изображений для человека не вызывает трудностей.

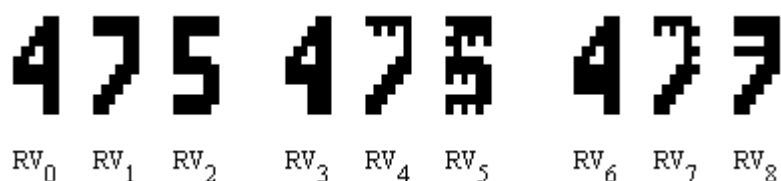


Рис.20.1. Обучающая выборка.

В результате самообучения на выходе НС сформировалось множество выходных сигналов, представленное на рис.20.2. Каждый столбик приведенной диаграммы представляет собой число, соответствующее 12-битовому числу, получаемому на выходе НС, если считать, что каждый выход соответствует определенному разряду двоичного кода числа.

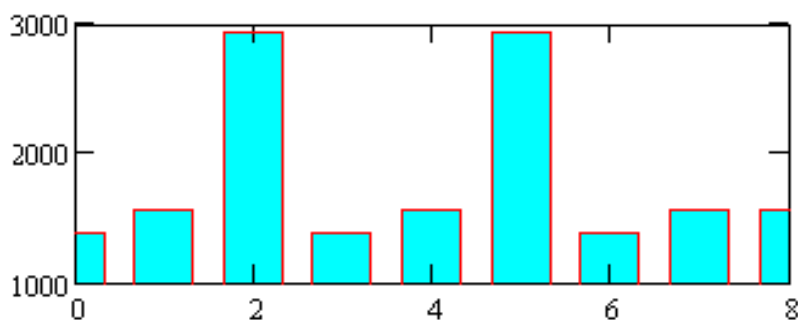


Рис.20.2. Множество выходных сигналов.

Сопоставление рисунков 20.1 и 20.2 позволяет сделать заключение, что нейронная сеть абсолютно правильно осуществляет классификацию изображений цифр. Столбики 0,3,6 соответствуют различным вариантам изображения цифры «4», столбики 2,4,7,8 – «7», столбики 2,5 – «5».

Карты выходных сигналов НС приведены на рис.20.3. На горизонтальной оси карты указаны номера выходов нейронной сети от 0 до 11, а по вертикали – значения соответствующих выходов. Каждый из трех графиков соответствует одной из трех цифр. Если НС относит два изображения к одному классу, то на

карте линии этих изображений сливаются для всех номеров выходов. Поскольку все значения близки по абсолютной величине к 1, то можно заключить, что 300 циклов самообучения было достаточно.

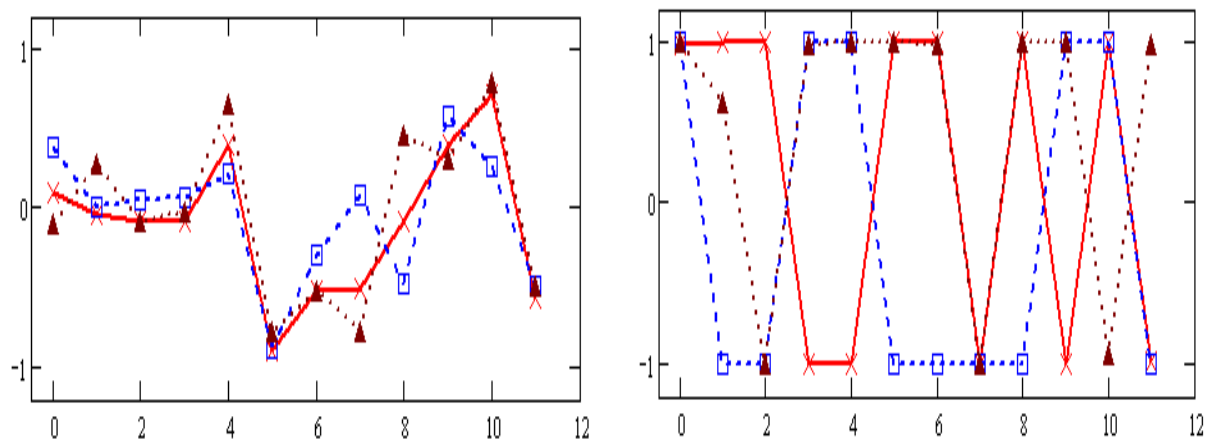


Рис.20.3. Карты выходных сигналов до и после обучения.

Карта выходных сигналов до обучения иллюстрирует несовпадение реакции НС на различные изображения одного и того же символа. После обучения эти графики сливаются в один.

Успешность самообучения зависит от начальных значений матриц весовых коэффициентов. В описываемом примере при некоторых генерациях число распознанных классов уменьшалось до двух.

21. Генетический алгоритм обучения НС

Генетический алгоритм обучения состоит в совмещении процедур случайного и направленного поиска оптимального решения задачи обучения. Рассматривается сразу несколько вариантов решения (*популяция*), которые случайным образом заполняют область допустимых решений. Из текущей популяции строится новое поколение путем применения специальных процедур:

- *Селекция* – наилучшие представители популяции напрямую попадают в следующее поколение.
- *Скращивание* – из представителей популяции получается два новых, посредством разрыва этих представителей по одному и тому же сечению и обмена полученными частями.
- *Мутация* – представитель популяции изменяется в результате замены части его элементов на противоположные.

Работа этих процедур носит случайный характер в соответствии с приписываемыми им вероятностями:

$$P_{\text{селекции}} + P_{\text{скращивания}} + P_{\text{мутации}} = 1.$$

Численность популяции на каждом шаге ограничивается на основе *схемы селекции* и критерия оптимизации. В качестве такого критерия в первую очередь используется критерий минимума невязки (18.1). Каждая новая популяция может в процессе своего существования *циклически* обучаться путем применения одного из известных методов перед последующей генерацией нового поколения.

Применение генетического алгоритма к обучению нейронных сетей сводится к преобразованию матриц весовых коэффициентов по заданным и, в принципе, неоднозначно определяемым правилам. Хорошие результаты

обучения дают следующие правила скрещивания и мутации матриц весовых коэффициентов:

$$W_{CROSS(A,B)}^{(p+1)} = W_A^{(p)} \cdot (1 - k_{CROSS}) + W_B^{(p)} \cdot k_{CROSS}$$

$$W_{MUT(A)}^{(p+1)} = -W_A^{(p)} + k_{MUT} \cdot (RND(1) - 0.5)$$

где p - номер популяции, $W_A^{(p)}, W_B^{(p)}$ - скрещиваемые матрицы (предки), k_{CROSS} - коэффициент скрещивания, $W_{CROSS(A,B)}^{(p)}$ - результирующая матрица (потомок), $W_{MUT(A)}^{(p+1)}$ - результат мутации, k_{MUT} - коэффициент мутации. Функция $RND(1)$ является генератором случайных чисел, подчиненных равномерному закону распределения и принадлежащих интервалу $[0,1]$. Коэффициент скрещивания k_{CROSS} рекомендуется назначать из интервала $[0,1]$. Коэффициент k_{MUT} является масштабирующим и если весовые коэффициенты генерируются в интервале $[-1,+1]$, то его можно приравнять 1.

Пример применения генетического алгоритма для обучения трехслойной НС распознаванию изображений приведен на рис.20.1. Об успешности процесса обучения целесообразно судить по величине невязки выходного сигнала.

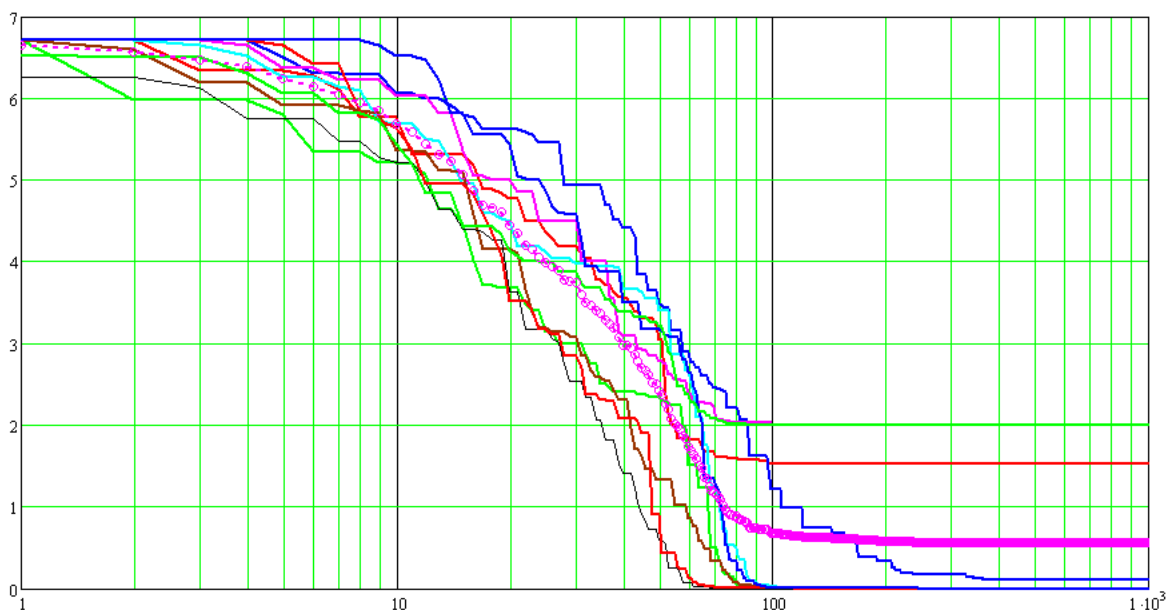


Рис.21.1. График изменения невязки в процессе обучения.

На рисунке приведены зависимости невязки от номера шага обучения для 10-ти особей (наборов весовых матриц). Графики демонстрируют две особенности алгоритма:

- сочетание плавного и скачкообразного изменения невязки,
- некоторые особи прекращают обучаться.

Плавное изменение невязки соответствует процедуре скрещивания, а скачки – появлению удачных особей-мутантов.

Недостаток генетического алгоритма обучения: реализация алгоритма требует больших вычислительных ресурсов, по сравнению с методом обратного распространения ошибки.

Основное преимущество генетического алгоритма обучения состоит в том, что он не зависит от внутренней структуры обучаемой системы. Алгоритмы МОРО и Хебба применимы только для обучения слоистых НС без обратных связей, генетический алгоритм может быть применен к любому виду НС или других видов систем трансформации информации.

22. Сходимость алгоритмов обучения

Сходимость алгоритмов обучения зависит от параметров разделимости множеств распознаваемых объектов. Идеальное обучение возможно лишь в том случае, если множества распознаваемых объектов разделимы на множестве функций, реализуемых обучаемой системой. Кроме того, решения задачи обучения могут не обладать свойством однозначности.

Процесс обучения может быть представлен в следующем виде:

$$W(t+1) = W(t) - \gamma(t) \cdot \delta(W(t)) \cdot y(W(t)).$$

В этом случае алгоритм сходится, если выполняются следующие

условия: $\gamma(t) > 0$, $\sum_{t=1}^{\infty} \gamma(t) = \infty$, $\sum_{t=1}^{\infty} \gamma(t)^2 < \infty$.

Коэффициент скорости обучения влияет на устойчивость процесса обучения. На рис.21.1 приведены три графика изменения невязки в процессе обучения для трех значений этого коэффициента.

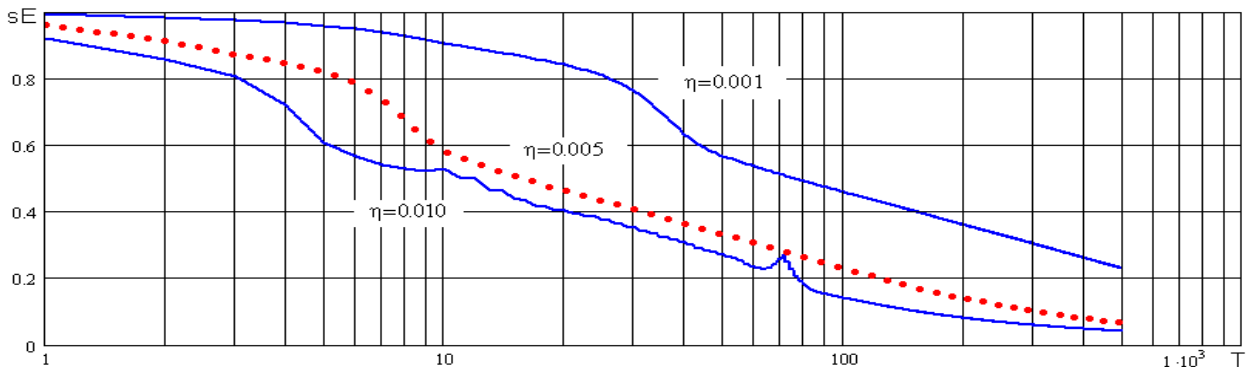


Рис.22.1. Пример влияния коэффициента скорости обучения на процесс обучения

Наиболее близким к оптимальному значению, судя по полученным графикам, следует считать $\eta = 0.005$, так как на графике для $\eta = 0.010$ видно несколько кратковременных срывов процесса обучения, а график для $\eta = 0.001$ соответствует замедленному уменьшению невязки.

23. Самоорганизующаяся коммуникация в системе с нейросетевым управлением

Примером эффективного применения НС служит решение задачи управления коллективом роботов, занимающихся поиском и сбором мин.

Каждый робот управляется НС прямого распространения с сигмоидальной активационной функцией. Обучение проводится генетическим алгоритмом. Структура нейросетевой системы управления одного робота приведена на рис.23.1.

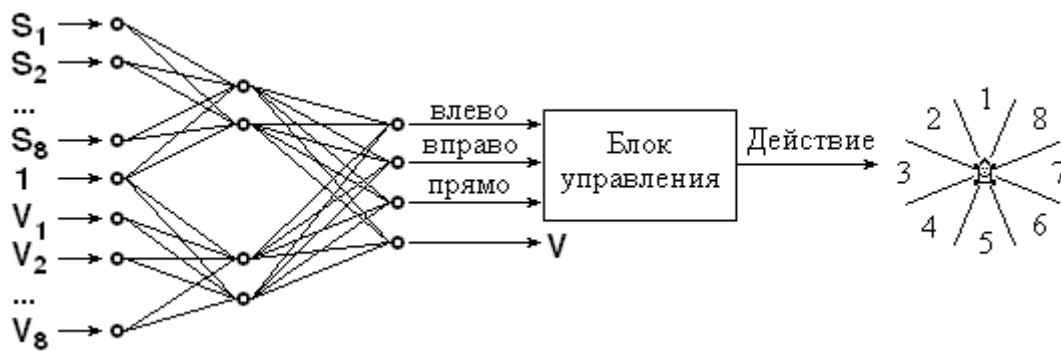


Рис.23.1. НС система управления роботом

Входными сигналами системы являются:

- $S_1 \div S_8$ - информация о не обезвреженных минах;
- $V_1 \div V_8$ - сигналы направленных приемников коммуникационной информации от других роботов;
- $R_{i,j}$ - расстояние до j -той мины или робота в i -том секторе;
- $\alpha \in [0;4]$ - параметр чувствительности робота;
- V – интенсивность излучаемого роботом коммуникационного сигнала.

Расчетные соотношения для входных сигналов:

$$S_i = \sum_j \frac{1}{R_{i,j}^\alpha}, \quad V_i = \sum_j \frac{1}{(v_{i,j} \cdot R_{i,j})^\alpha}.$$

Обучению подвергаются 52 весовых коэффициента и параметр α . Вероятность скрещивания – 0,8, мутации – 0,05. Критерий оптимизации – количество обезвреженных мин в течение 100 единиц времени. Через каждые 100 единиц времени генерируется новое поколение.

Результаты моделирования процесса сбора мин роботами представлены в виде графика (рис.23.2). Начальный участок длиной 1000 единиц времени был отведен для обучения системы, и поиск осуществлялся случайным образом. Разминирование выполнялось автоматически за 5 единиц времени. Карта имела размер 160x100 клеток/шагов и на ней находилось 800 мин. Дальность обнаружения мин составляла 3 дискрета.

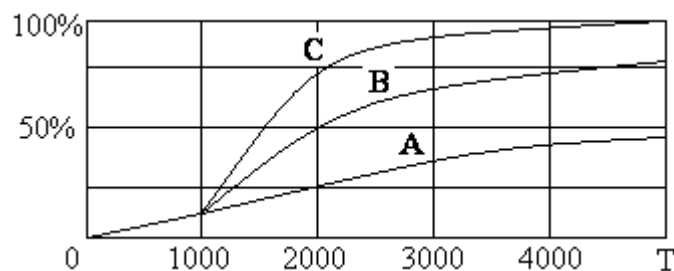


Рис.23.2. Результаты моделирования процесса сбора мин

Линия «А» на графике соответствует работе системы без обучения, то есть на основе случайного поиска. Линия «В» соответствует работе обучающейся системы, но без использования коммуникационных сигналов. Линия «С» соответствует работе обучающейся системы, использующей коммуникационные сигналы.

24. Структура микросхем, реализующих НС элементы

Специализированные цифровые нейронные кристаллы реализуют архитектуру, основанную на сочетании небольших процессорных блоков и матричных элементов памяти. Такая архитектура использует специфическую особенность НС: массовое, параллельное, многократно повторяемое умножение матрицы на вектор. В ОЗУ таких микросхем сохраняются матрицы весовых коэффициентов и входные вектора. Управление осуществляется с использованием специальных команд:

- загрузка весовых коэффициентов и данных;
- движение информации через память весов;
- коррекция весовых коэффициентов.

Технологические особенности микросхем ограничивают количество межнейронных связей; поэтому появилось несколько типов микросхем, отличающихся способами организации связи между нейронами. Наиболее

просто организованы клеточные НС, структура ячеек которых показана на рис.24.1. Такая структура образована непосредственными связями нейрона только с ближайшими нейронами и магистральной связью с другими ячейками нейронов.



Рис.24.1. Структура ячейки клеточной НС

Один из возможных вариантов организации многослойной НС на основе клеточных НС микросхем приведен на фрагменте А рис.24.2.

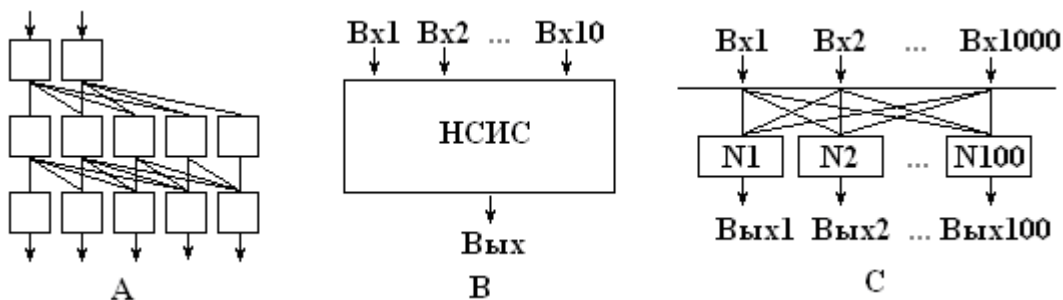


Рис.24.2. Многослойная клеточная НС и мультиплексная НС интегральная схема

Распознавание изображений и другие высоко информативные задачи требуют использования НС с большим количеством нейронов. При этом возникает проблема большого числа входов и выходов интегральных микросхем, реализующих НС. Мультиплексирование, то есть использование одних и тех же выводов микросхемы для передачи различных сигналов за счет их временного чередования, позволяет решать эту проблему. На фрагментах В и С рис.24.2 приведены соответственно электрическая и структурная схемы мультиплексной НС интегральной схемы. Схема организации вычислительных процессов в мультиплексной НС интегральной микросхеме приведена на

рис.24.3. Такая схема позволяет организовать вычисления синхронно с поступающей входной информацией и сократить число вычислительных элементов.

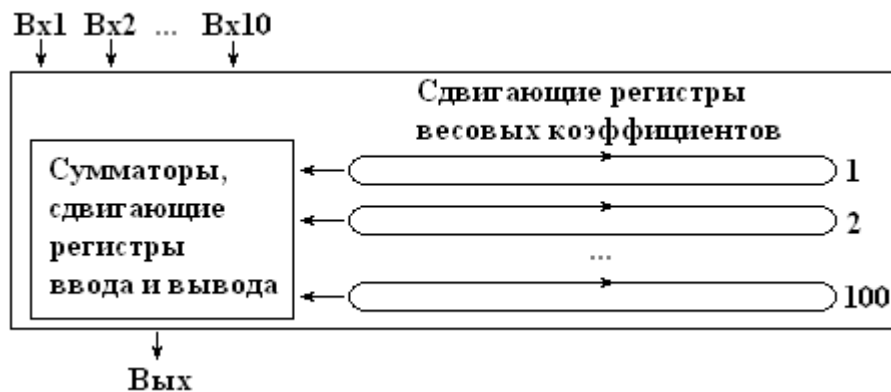


Рис.24.3. Схема организации вычислительного процесса

В настоящее время, микросхемы, реализующие НС элементы, используют для разработки нейросетевых компьютеров, представляющих собой результат интеграции обычных последовательных ЭВМ и нейросетевых плат.

25. Распознавание образов на основе критерия Байеса

Условием непосредственного использования критерия Байеса [2] для распознавания образов является условие полноты априорной информации. Если это условие не выполняется, то необходимо дополнительно использовать процедуры обучения или самообучения.

Считаются известными следующие величины:

- Ω_1, Ω_2 – возможные классы;
- $P(\Omega_1), P(\Omega_2)$ – априорные вероятности появления объектов;
- $f_1(x) = f(x | \Omega_1), f_2(x) = f(x | \Omega_2)$ – условные плотности вероятности признака;
- C – платежная матрица;
- x – признак, по которому осуществляется классификация.

Априорная информация используется для определения величины x_{Π} , которая является пороговым значением, относительно которого происходит классификация.

Правило принятия решений при классификации:

если $x > x_{\Pi}$, то $x \in \Omega_2$,

если $x \leq x_{\Pi}$, то $x \in \Omega_1$.

Вероятностный характер распознавания допускает возможность возникновения ошибочной классификации, которую принято характеризовать вероятностями ошибок первого и второго рода. Иногда, эти ошибки называют ложной тревогой и пропуском цели соответственно.

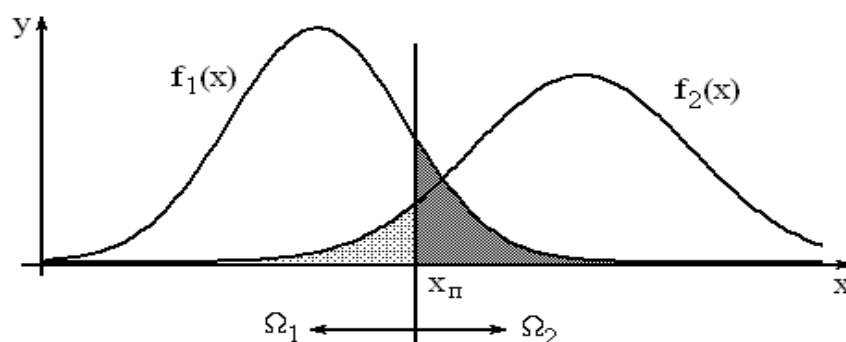


Рис.25.1. Порог критерия Байеса

Ошибка первого рода соответствует ситуации, когда распознаваемый объект принадлежит первому множеству, но классификатор относит его ко второму классу. Вероятность ошибки первого рода $Q_1(x)$ определяется по формуле:

$$Q_1(x_{\Pi}) = \int_{x_{\Pi}}^{\infty} f_1(x) dx.$$

Ошибка второго рода: $Q_2(x_{\Pi}) = \int_{-\infty}^{x_{\Pi}} f_2(x) dx.$

Условные вероятности правильных решений:

$$D_1(x_n) = 1 - Q_1(x_n), \quad D_2(x_n) = 1 - Q_2(x_n).$$

Выбор значения x_n проводят на основе учета потерь, сопряженных с правильными и ошибочными решениями. Величины потерь задаются в виде платежной матрицы:

$$C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}.$$

Средний риск при многократном распознавании неизвестных объектов равен сумме потерь, связанных с неправильными и правильными решениями с учетом вероятностей их появления и априорными вероятностями появления объектов классов:

$$\begin{aligned} \bar{R}(x) = & P(\Omega_1) \cdot c_{11} \cdot (1 - Q_1(x)) + P(\Omega_1) \cdot c_{12} \cdot Q_1(x) + \\ & + P(\Omega_2) \cdot c_{22} \cdot (1 - Q_2(x)) + P(\Omega_2) \cdot c_{21} \cdot Q_2(x). \end{aligned} \quad (24.1)$$

Пороговое значение признака для критерия Байеса выбирают так, чтобы средний риск был минимален, для чего необходимо решить уравнение:

$$\frac{d\bar{R}(x)}{dx} = 0.$$

Вычисление производных слагаемых среднего риска приводит к следующему уравнению:

$$P(\Omega_1) \cdot [c_{11} \cdot f_1(x) - c_{12} \cdot f_1(x)] + P(\Omega_2) \cdot [c_{21} \cdot f_2(x) - c_{22} \cdot f_2(x)] = 0.$$

Окончательно, форма уравнения для определения порога x_n имеет вид:

$$\lambda(x) = \lambda_0, \quad (24.2)$$

где $\lambda(x) = \frac{f_2(x)}{f_1(x)}$ - функция отношения правдоподобия,

$\lambda_0 = \frac{P(\Omega_1) \cdot (c_{12} - c_{11})}{P(\Omega_2) \cdot (c_{21} - c_{11})}$ - критическое значение коэффициента правдоподобия.

26. Распознавание образов с применением минимаксного критерия

Минимаксный критерий используется для построения систем распознавания, когда априорные вероятности появления объектов соответствующих классов $P(\Omega_1), P(\Omega_2)$ неизвестны. В этом случае невозможно минимизировать средний риск, и байесовская стратегия непосредственно неприменима.

Минимаксный критерий рассматривает наихудший вариант величин априорных вероятностей $P(\Omega_1), P(\Omega_2)$ по отношению к величине среднего риска $\bar{R}$. Рассматривая классификацию на два класса, можно считать, что справедливо равенство: $P(\Omega_2) = 1 - P(\Omega_1)$. Это равенство позволяет записать формулу для среднего риска (30.1) в виде функции от априорной вероятности. Типичный вид такой функции приведен на рис.25.1. Точки графика вычисляются путем решения уравнения байесовской стратегии (24.2) для каждого значения априорной вероятности $P(\Omega_1)$.

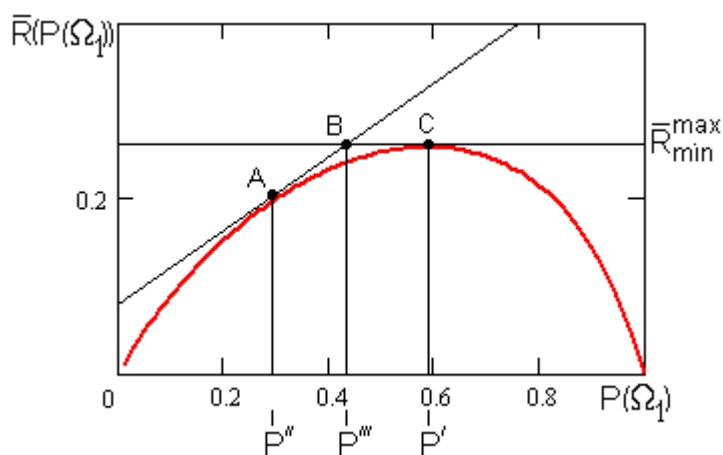


Рис.26.1. Зависимость среднего риска от априорной вероятности

Минимаксная стратегия соответствует точке C, выбор которой обеспечивает то, что при любом фактическом значении априорной вероятности

$P(\Omega_1)$ средние потери не будут превышать $\bar{R}(P(\Omega_1)) = \bar{R}_{min}^{max}$. Значение порога минимаксного критерия определяется из уравнения:

$$\frac{d\bar{R}(P(\Omega_1))}{dP(\Omega_1)} = 0.$$

Вычисление производных слагаемых среднего риска приводит к следующему уравнению:

$$(c_{12} - c_{11}) \cdot Q_1(x) + c_{11} = (c_{21} - c_{22}) \cdot Q_2(x) + c_{22}.$$

Иногда используют упрощающие соотношения и обозначения: $c_{11} = c_{22} = 0$, $c_{12} = c_1$, $c_{21} = c_2$. В таком случае уравнение для порога критерия минимакса имеет вид:

$$c_1 \cdot Q_1(x) = c_2 \cdot Q_2(x).$$

Выбор иной стратегии, например стратегии, соответствующей точке А, приводит к тому, что на интервале $[0, P''']$ средний риск меньше $\bar{R}(P(\Omega_1)) < \bar{R}_{min}^{max}$, но на интервале $[P''', 1]$ потери будут больше гарантированного значения $\bar{R}(P(\Omega_1)) > \bar{R}_{min}^{max}$.

Минимаксная стратегия есть байесовская стратегия для наихудших значений априорных вероятностей, дающая осторожное, но гарантированное значение среднего риска.

27. Распознавание образов с применением критерия Неймана - Пирсона

Критерий Неймана-Пирсона используется для построения систем распознавания в условиях, когда неизвестны априорные вероятности появления заданных классов $P(\Omega_1), P(\Omega_2)$ и платежная матрица С.

Критерий состоит в том, что задают допустимое значение условной вероятности ошибки первого рода и находят минимум условной вероятности ошибки второго рода. Из доступной информации задают условие:

$$Q_1(x) \leq A.$$

Из определения условных вероятностей ошибок первого и второго рода и из рис.25.1. видно, что уменьшение порогового значения приводит к росту ошибки первого рода и уменьшению ошибки второго рода. Следовательно, минимальное значение ошибки второго рода достигается при максимально допустимом значении ошибки первого. Уравнение для порога критерия Неймана-Пирсона имеет вид:

$$Q_1(x) = A.$$

Значение параметра $A \in (0,1)$ задается на основе опытной информации о процессе распознавания в конкретных условиях.

28. Процедура последовательных решений в распознавании образов

Процесс распознавания часто строится не на одном признаке, а на некотором множестве признаков (векторе признаков), физическая природа которых существенно различна. Увеличение числа признаков приводит к росту стоимости распознавания, поэтому стремятся к минимизации размерности вектора признаков. Процедура последовательных решений предназначена для решения задач распознавания при использовании минимального числа признаков и состоит в следующем:

- После измерения очередного признака задействуется алгоритм распознавания, использующий все имеющиеся к данному моменту времени признаки.
- В зависимости от результатов распознавания либо объект считается распознанным, либо принимается решение о необходимости продолжения измерений.

Процедура последовательных решений на каждом шаге формирует два пороговых значения для функции отношения правдоподобия. Для этого

задаются допустимыми значениями ошибок первого и второго рода. Значения порогов вычисляют по формулам:

$$A \leq (1 - Q_1) / Q_2, \quad B \geq Q_1 / (1 - Q_2).$$

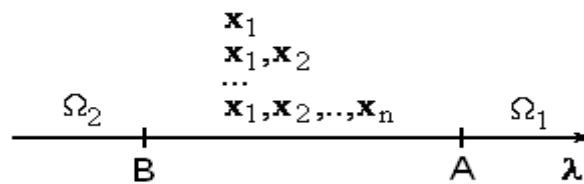


Рис. 28.1. Критические области процедуры последовательных решений

Значение функции отношения правдоподобия вычисляют по формуле:

$$\lambda(x_1, x_2, \dots, x_n) = \frac{f_2(x_1, x_2, \dots, x_n)}{f_1(x_1, x_2, \dots, x_n)}.$$

Алгоритм процедуры последовательных решений:

- Если $\lambda \geq A$, то распознаваемый объект $\omega \in \Omega_1$.
- Если $\lambda \leq B$, то распознаваемый объект $\omega \in \Omega_2$.
- Если $B < \lambda < A$, то принимается решение о получении очередного нового признака.

Процедура последовательных решений применяется, например, для диагностирования некоторых заболеваний, когда получение новых анализов о состоянии организма больного связано с риском для здоровья.

29. Пример определения пороговых значений для вероятностных алгоритмов распознавания

Заданы следующие функции условной плотности вероятности:

$$f_1(x) = \begin{cases} k_1 \cdot x/t_1, & x \leq t_1 \\ k_1 \cdot ((1-x)/(1-t_1))^2, & x > t_1 \end{cases}$$

$$f_2(x) = \begin{cases} k_2 \cdot (x/t_2)^2, & x \leq t_2 \\ k_2 \cdot ((1-x)/(1-t_2))^2, & x > t_2 \end{cases}.$$

Параметры и константы этих функций имеют следующие значения:

$t_1 = 0.1$, $t_2 = 0.6$, $c_1 = 1.1$, $c_2 = 1.3$, $k_1 = 2.857$, $k_2 = 3.0$, $A = 0.1$.

Графики функций $f_1(x)$ и $f_2(x)$ приведен на рис.29.1.

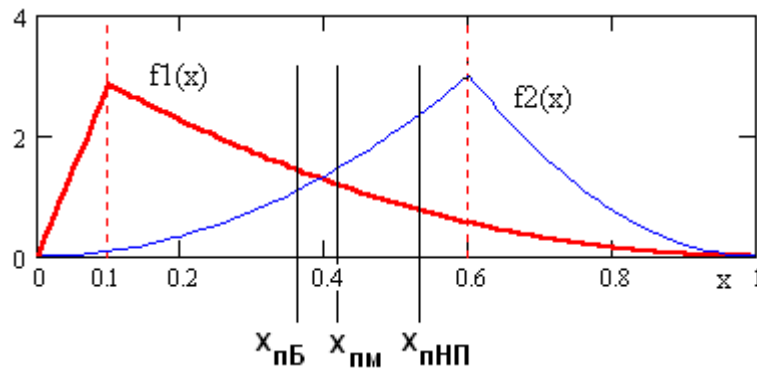


Рис.29.1. Графики функций условной плотности вероятности

С целью упрощения процесса поиска пороговых значений, параметры распределений подобраны так, что для всех критериев значения порогов находятся в интервале $[x_1, x_2]$, поэтому в уравнениях используются правая ветвь функции $f_1(x)$ и левая - $f_2(x)$. Для критерия Байеса уравнение имеет вид:

$$\frac{f_2(x)}{f_1(x)} = \frac{c_1 \cdot P(\Omega_1)}{c_2 \cdot P(\Omega_2)}.$$

Подстановка численных значений и последовательные преобразования позволяют получить квадратное уравнение:

$$f_2(x) = \frac{1.1}{1.3} \cdot f_1(x), \quad k_2 \cdot (x/0.6)^2 = \frac{11}{13} \cdot k_1 \cdot ((1-x)/(1-0.1))^2,$$

$$\frac{3.0}{0.36} \cdot x^2 = \frac{1.1 \cdot 2.857}{1.3 \cdot 0.81} \cdot (1-x)^2, \quad 5.348 \cdot x^2 + 5.970 \cdot x - 2.985 = 0.$$

Решение квадратного уравнения дает следующее значение порога критерия Байеса: $x_{пБ} = 0.374$.

Уравнение для порога критерия минимакса имеет вид:

$$c_1 \cdot Q_1(x) = c_2 \cdot Q_2(x).$$

Подставив в это уравнение заданные функции получим:

$$1.1 \cdot \int_{x_{\Pi}}^1 \frac{2.857}{0.81} \cdot (1-x)^2 dx = 1.3 \cdot \int_0^{x_{\Pi}} \frac{3.0}{0.36} \cdot x^2 dx,$$

$$\beta \cdot \left(x - x^2 + \frac{x^3}{3} \right) \Big|_{x_{\Pi}}^1 = \frac{x^3}{3} \Big|_0^{x_{\Pi}}, \quad \beta = \frac{1.1 \cdot 2.857 \cdot 0.36}{1.3 \cdot 0.81 \cdot 3.0} = 0.358,$$

$$3.796 \cdot x^3 - 3 \cdot x^2 + 3 \cdot x - 1 = 0.$$

Решение кубического уравнения дает следующее значение порога критерия минимакса: $x_{\Pi\text{М}} = 0.415$.

Уравнение для порога критерия Неймана-Пирсона имеет вид:

$$Q_1(x) = A.$$

Подставив в это уравнение заданные функции, получим:

$$\int_{x_{\Pi}}^1 \frac{2.857}{0.81} \cdot (1-x)^2 dx = 0.1,$$

$$\gamma \cdot \left(x - x^2 + \frac{x^3}{3} \right) \Big|_{x_{\Pi}}^1 = 0.1, \quad \gamma = \frac{2.857}{0.81} = 3.522,$$

$$x^3 - 3 \cdot x^2 + 3 \cdot x - 0.915 = 0.$$

Решение кубического уравнения дает следующее значение порога критерия Неймана-Пирсона: $x_{\Pi\text{НП}} = 0.56$.

30. Основные определения исчисления предикатов

Предикат – это высказывательная функция, определенная на некотором множестве M , то есть такая n -местная функция P , которая каждому упорядоченному набору $(a_1, a_2, \dots, a_n)$ элементов множества M сопоставляет некоторое высказывание, обозначаемое $P(a_1, a_2, \dots, a_n)$ [20].

Предикат – фундаментальное понятие математической логики, являющееся условием, сформулированным в терминах некоторого точного логико-математического языка. Предикат содержит обозначения для произвольных объектов – переменные. При замещении переменных именами объектов предикат задает точно определенное *высказывание*.

Предикат, имеющий одну переменную, называется свойством, а предикат с двумя переменными – отношением. *Исчисление предикатов первого порядка* – формальный язык, выражающий разнообразные утверждения.

Правильно построенные формулы (ППФ) – допустимые выражения исчисления предикатов. Элементы формул: предикатные символы, символы переменных, функциональные символы, символы констант, символы логических операций. *Система продукций* – обобщение вычислительного формализма, использующего предикаты и оперирующее понятиями: структура, класс, объект, свойства, правила.

Пример: $P(x > 2)$ - предикат, который имеет значение «истина», если $x = 5$. Если $x = -3$, то предикат возвратит значение «ложь».

31. Высказывания. Кванторы и их свойства

Исчисление предикатов использует специальные операторы – кванторы, которые могут быть двух видов:

$\forall x$ - квантор общности,

$\exists x$ - квантор существования.

Связанная переменная в ППФ – квантифицированная переменная. Свободная переменная – не квантифицирована.

Высказывание – ППФ, у которой все переменные связаны.

Первый порядок исчисления предикатов не допускает квантификации предикатных и функциональных символов.

Пример высказывания и соответствующей ППФ [11]:

$$(\forall x)(\text{Слон}(x) \rightarrow \text{Цвет}(x, \text{Серый})) = \text{«Все слоны серые»}.$$

В данном примере свойство «Слон(х)» определяет возможность отношения «Цвет(х, Серый)».

Свойства кванторов:

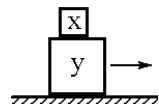
1. $(\exists x)\overline{P(x)} = (\forall x)\overline{P(x)}$.
2. $(\forall x)\overline{P(x)} = (\exists x)\overline{P(x)}$.
3. $(\forall x)(P(x) \wedge Q(x)) = (\forall x)P(x) \wedge (\forall x)Q(x)$.
4. $(\exists x)(P(x) \vee Q(x)) = (\exists x)P(x) \vee (\exists x)Q(x)$.
5. $(\forall x)P(x) = (\forall y)P(y)$.
6. $(\exists x)P(x) = (\exists y)P(y)$.

Пример интерпретации первого свойства:

Не существует ни одного слона, боящегося мышей. = Все слоны не боятся мышей.

Пример представления высказывания в исчислении предикатов:

Все кубики, находящиеся на кубиках, которые были сдвинуты или были скреплены с кубиками, которые сдвигались, также были сдвинуты:



$$(\forall x)(\forall y)\{[K(x) \wedge K(y) \wedge \{\text{Наверху}(x, y) \vee \text{Скр}(x, y)\}] \wedge \text{Сдв}(y)\} \rightarrow \text{Сдв}(x)\}.$$

32. Правила вывода. Унификация

Назначение правил вывода: на основе множества ППФ получить новые корректные ППФ. Известно большое количество правил вывода, среди которых наиболее часто применяемыми являются следующие правила:

Правило вывода 1: $\frac{A, A \Rightarrow B}{B}$. Модус поненс - операция, создающая ППФ W2

из ППФ вида W1 и $W1 \Rightarrow W2$. Это правило интерпретируется так: если истинно утверждение A , и из A следует B , то утверждение B тоже истинно.

Правило вывода 2: $\frac{A \Rightarrow B, B \Rightarrow C}{A \Rightarrow C}$. Транзитивность оператора импликации.

Правило вывода 3: $\frac{A \cup B, A \Rightarrow B}{B}$. Слияние утверждений A и B .

Правило вывода 4: $\frac{A \cup B, \neg A \cup C}{B \cup C}$. Резолюция двух ППФ.

Правило вывода 5: Специализация - создает ППФ $W(A)$ из ППФ $\forall(x)W(x)$, где A - любой символ константы.

Унификация – процесс поиска подстановочных термов на место переменных. Обозначение подстановок: $s = \{t_1/v_1, t_2/v_2, \dots, t_n/v_n\}$, где t_i - новая переменная, v_i - заменяемая переменная. $\{E_i\}s$ - подстановка s применяется к каждому элементу множества ППФ $\{E_i\}_{i \in 1..n}$. Назначение унификации – приведение ППФ к заданному виду.

Множество ППФ $\{E_i\}_{i \in 1..n}$ называется *унифицируемым*, если существует подстановка s такая, что $E_1s = E_2s = \dots = E_ns$. Такая подстановка называется *унификатором*. Применение унификатора сворачивает унифицируемое множество в один элемент.

33. Выполнимость и удовлетворимость ППФ

Если ППФ имеет значение Т ("истина") для всех возможных интерпретаций, то она называется *выполненной* (тавтологией) [11]. Пример: $P(A) \Rightarrow [P(A) \vee P(B)]$.

Метод таблицы истинности может всегда определить выполнимость любой ППФ, не содержащей переменных.

Теорема о неразрешимости исчисления предикатов. Невозможно найти универсальный метод установления факта выполненности квантифицированных ППФ. Если ППФ выполнена, то существует процедура, позволяющая установить выполнимость этой ППФ (полуразрешимость). Интерпретация удовлетворяет множеству ППФ, если каждая ППФ этого множества имеет значение Т ("истина") при этой интерпретации.

ППФ X логически следует из множества S ППФ, если каждая интерпретация, удовлетворяющая системе S , удовлетворяет также и X (но взаимной однозначности нет).

34. Алгоритм преобразования предикатов в предложения

Предложение – правильно построенная формула (ППФ), состоящая из дизъюнкции литералов (предикатов) [11].

Теорема: Любая ППФ исчисления предикатов может быть преобразована во множество предложений.

Пример ППФ: $(\forall x)\{P(x) \Rightarrow \{(\forall y)[P(y) \Rightarrow P(f(x, y))]\} \wedge (\forall y)[Q(x, y) \Rightarrow P(y)]\}$.

Этапы преобразования:

1. Исключение символов импликации подстановкой $\overline{X1} \vee X2$ вместо

$$X1 \Rightarrow X2: (\forall x)\{\overline{P(x)} \vee \{(\forall y)[\overline{P(y)} \vee P(f(x, y))]\} \wedge (\forall y)[Q(x, y) \vee \overline{P(y)}]\}.$$

2. Ограничение области действия символа отрицания одной атомарной формулой (закон Моргана):

$$(\forall x)\{\overline{P(x)} \vee \{(\forall y)[\overline{P(y)} \vee P(f(x, y))]\} \wedge (\exists y)[Q(x, y) \wedge \overline{P(y)}]\}.$$

3. Разделение квантифицированных переменных:

$$(\forall x)\{\overline{P(x)} \vee \{(\forall y)[\overline{P(y)} \vee P(f(x, y))]\} \wedge (\exists \omega)[Q(x, \omega) \wedge \overline{P(\omega)}]\}.$$

Переменные переименовываются так, чтобы каждый квантор имел свою переменную.

4. Исключение кванторов существования путем введения сколемовской функции. Эта функция в качестве аргументов должна содержать переменные, связанные кванторами общности, в область действия которых попадает исключаемый квантор существования. Если $\exists$ не входит в область действия никакого $\forall$, то вводится просто константа: $(\exists x)P(x) \rightarrow P(A)$.
 $(\forall x)\{P(x) \vee \{(\forall y)[P(y) \vee P(f(x, y))]\} \wedge [Q(x, g(x)) \wedge P(g(x))]\}$,
 где $g(x)$ – сколемовская функция.
5. Преобразование в предварительную форму: все $\forall$ перемещаются в начало ППФ: $(\forall x)(\forall y)\{P(x) \vee \{[P(y) \vee P(f(x, y))]\} \wedge [Q(x, g(x)) \wedge P(g(x))]\}$.
6. Приведение матрицы ППФ к конъюнктивной нормальной форме.
 Используется замена $X1 \vee (X2 \wedge X3) = (X1 \vee X2) \wedge (X1 \vee X3)$:
 $(\forall x)(\forall y)\{[P(x) \vee P(y) \vee P(f(x, y))]\} \wedge [P(x) \vee Q(x, g(x))]\} \wedge [P(x) \vee P(g(x))]\}$.
7. Исключение кванторов общности. Предполагается, что все переменные относятся к $\forall$.
8. Исключение символов $\wedge$ (конъюнкции). Переход к множеству ППФ, каждая из которых является дизъюнкцией литералов, т.е. предложений.
9. Переименовывание переменных. Символы переменных должны быть изменены так, чтобы каждый присутствовал только в одном предложении:

$$\left[\begin{array}{l} P(x) \vee P(y) \vee P(f(x, y)) \\ \overline{P(x)} \vee Q(x, g(x)) \\ \overline{P(x)} \vee \overline{P(g(x))} \end{array} \right] \rightarrow \left[\begin{array}{l} P(x1) \vee P(y) \vee P(f(x1, y)) \\ \overline{P(x2)} \vee Q(x2, g(x2)) \\ \overline{P(x3)} \vee \overline{P(g(x3))} \end{array} \right].$$

35. Резолюция и резольвента

Резолюция – правило вывода, применяемое к ППФ в виде предложений [9,11]. Метод резолюций предложен Д. Робинсоном в 1965 году и служит основой для создания алгоритмов автоматического доказательства теорем и формирования ответов на вопросы. Структура метода приведена далее.

$$\text{Дано: } \begin{cases} P_1 \vee P_2 \vee \dots \vee P_N \\ \overline{P_1} \vee Q_2 \vee \dots \vee Q_M \end{cases}$$

Результат: $Q_2 \vee \dots \vee Q_M \vee P_2 \vee \dots \vee P_N$ (резольвента).

Резольвента *логически следует* из своих порождающих предложений.

Если предложения содержат переменные и задаются множеством литералов $\{L_i\}$ и $\{M_i\}$, соединенных знаком дизъюнкции $\vee$, и если можно определить соответствующие подмножества $\{l_i\}$ и $\{m_i\}$, такие, что для объединения $\{l_i\}$ и $\{\overline{m_i}\}$ существует наиболее общий унификатор, то резольвента имеет вид:

$$\{\{L_i\} - \{l_i\}\}s \vee \{\{M_i\} - \{m_i\}\}s.$$

Два предложения могут иметь более одной резольвенты. Количество резольвент равно количеству способов выбора подмножеств $\{l_i\}$ и $\{m_i\}$.

36. Система опровержения на основе резолюций

Правило построения резолюций позволяет формализовать процесс доказательства теорем [11]. Система доказательств использует метод доказательства от “противного”.

Дано: S - множество ППФ.

Доказать: W - целевая ППФ.

Алгоритм:

- 1) К множеству S добавляется отрицание ППФ W.
- 2) Расширенное множество преобразуется в множество предложений.
- 3) Ищется резолюция, представленная пустым предложением, т.е. соответствующая противоречию.
- 4) Если пустое предложение найдено, то теорема считается доказанной.
- 5) Если не удастся получить резолюцию, соответствующую противоречию, то теорема считается не доказанной.

Пример 1: Имеются следующие утверждения:

1) Кто может читать, тот грамотный: $(\forall x)(\text{Читает}(x) \rightarrow \text{Грамотный}(x))$.

2) Дельфины не грамотны: $(\forall x)(\text{Дельфин}(x) \rightarrow \neg \text{Грамотный}(x))$.

3) Некоторые дельфины обладают интеллектом:

$(\exists x)(\text{Дельфин}(x) \wedge \text{Интеллект}(x))$.

Требуется доказать: некоторые из тех, кто обладает интеллектом, не могут читать: $(\exists x)(\text{Интеллект}(x) \wedge \neg \text{Читает}(x))$.

Множество утверждений соответствует следующим предложениям:

1) $\neg \text{Читает}(x) \vee \text{Грамотный}(x)$

2) $\neg \text{Дельфин}(y) \vee \neg \text{Грамотный}(y)$

3) $\text{Дельфин}(A), \text{Интеллект}(B)$, где A, B – сколемовские константы.

4) $\neg \text{Интеллект}(z) \vee \text{Читает}(z)$ – отрицание доказываемой теоремы.

Процесс поиска противоречивой резолюции представляют в виде диаграммы, называемой деревом резолюций. Для рассматриваемого примера диаграмма представлена на рис.36.1. Базисное множество резолюций расположено в верхней строке диаграммы, а вычисляемые резолюции располагаются на более низких уровнях.

Рассматриваемое утверждение следует считать доказанным, так найдено пустое предложение. Верхняя строка диаграммы состоит из предложений *базового множества*. Вторая и все следующие строки принадлежат к различным *уровням резолюций*, номер уровня вершины дерева доказательств является одной из характеристик скорости получения доказательства.

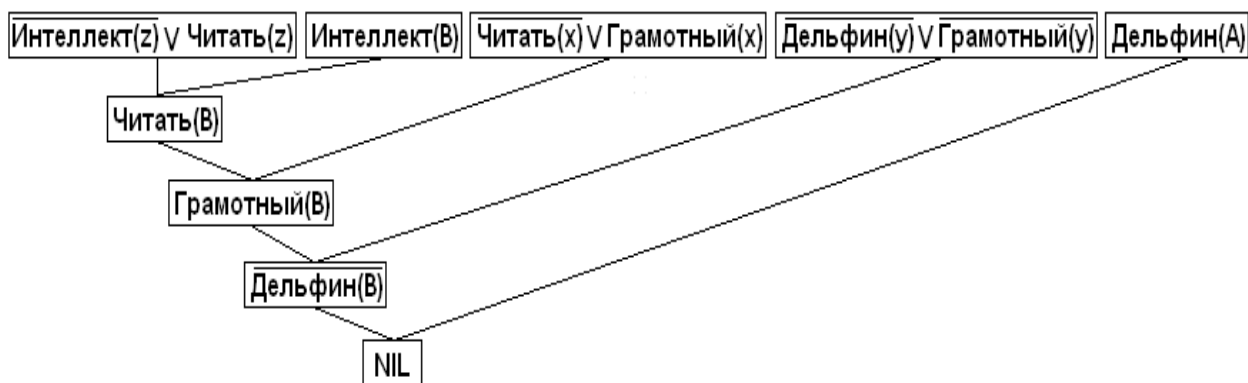


Рис. 36.1. Диаграмма вычисления резолюции

37. Стратегии управления процессом поиска доказательства методом резолюций

Процесс поиска доказательства методом резолюций в общем случае требует большого объема вычислений и ресурсов памяти, так как количество порождаемых предложений растет, как число возможных сочетаний предложений базисного множества и предложений более низких уровней резолюций. Необходимость предотвращения или ослабления комбинаторного взрыва обусловила применение специальных стратегий.

Виды стратегий:

- стратегия поиска в ширину,
- стратегия опорного множества,
- стратегия предпочтения одночленам,
- стратегия линейная по входу,

Стратегия поиска в ширину предусматривает полный перебор возможных резольвент на каждом уровне резолюций до обнаружения пустого предложения. Эта стратегия является полной, то есть она всегда приводит к решению, если оно существует.

Стратегия опорного множества состоит в том, что в каждой резольвенте один из родителей порожден потомком отрицания целевой ППФ. Все такие

резольвенты несут идею обратного рассуждения. Стратегия позволяет сократить число резольвент на нижних уровнях.

Стратегия предпочтения одночленам является модификацией стратегии опорного множества, предусматривающей преимущественное использование однолитеральных предложений для вычисления резольвент нижних уровней. Такая стратегия концентрирует поиск в направлении пустого предложения.

Стратегия линейная по входу требует обязательной принадлежности одного из родительских предложений базовому множеству. Пример предыдущего параграфа соответствует этой стратегии.

Ограничения, вводимые стратегиями, приводят к увеличению глубины диаграммы опровержения и отсутствию полноты стратегии. На рис.37.1 приведена диаграмма процесса поиска доказательства в ширину, которая показывает, что доказательство получено на третьем уровне резолюций, в отличие от линейной стратегии (рис.36.1), позволившей завершить поиск лишь на четвертом уровне.

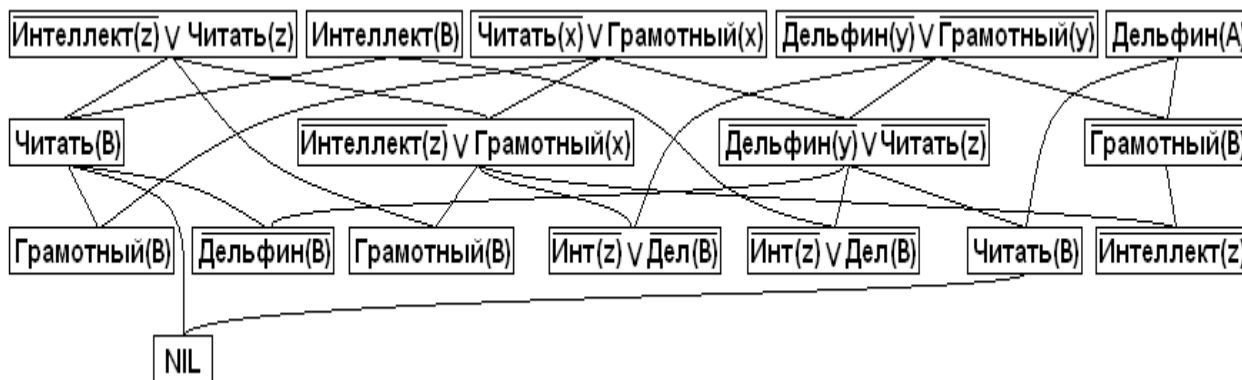


Рис.37.1. Диаграмма процесса поиска доказательства в ширину

Наличие в базовом множестве предложений, состоящих из трех и более литералов, вызывает появление на более низких уровнях резолюций все более длинных предложений, которые могут содержать более четырех литералов.

Снижение скорости роста новых предложений при вычислении резолюций обеспечивают, применяя *стратегии упрощения*:

- Любое предложение, содержащее литерал и его отрицание, может быть отброшено. Например: $P(x) \vee B(y) \vee \bar{B}(y)$.
- Если литерал, входящий в предложение, для любых допустимых интерпретаций имеет значение “ложь”, то его можно исключить. Например: $P(x) \vee Q(y) \vee R(7 = 3)$ эквивалентно $P(x) \vee Q(y)$.
- Предложение в неудовлетворимом множестве, которое подсуммируется с другим предложением, можно отбросить. Если $\{L_i\}_s$ является подмножеством $\{M_i\}_s$, то $\{L_i\}_s$ подсуммирует $\{M_i\}_s$. Например: $P(x)$ подсуммирует $P(y) \vee Q(z)$.

38. Извлечение ответа из опровержения, основанного на резолюции. Этапы процесса извлечения ответа

Извлечение ответа – это преобразование дерева опровержения в дерево доказательства с некоторым утверждением в корневой вершине. Это утверждение может быть использовано в качестве ответа. *Дерево доказательств* – это доказательство методом резолюции, которое основано на аксиомах и тавтологиях [11]. Алгоритм извлечения ответа основан на преобразовании в тавтологию каждого предложения, возникающего в результате отрицания целевой функции.

Этапы извлечения ответа:

- Поиск дерева опровержения методом резолюций. Отмечаются подмножества унификации, то есть литералы с изменяемыми аргументами.
- Сколемовские функции в предложениях, являющихся результатом отрицания целевой функции, заменяются на новые переменные.
- Предложения, являющиеся результатом отрицания целевой функции, преобразуются в тавтологии.

Д. Модификация дерева опровержений в дерево доказательств при неизменном множестве унификаторов.

Е. Получение ответа в корневой вершине.

В качестве примера рассмотрим следующую логическую задачу:

“Если Джим ходит туда же, куда ходит Джон, и если Джон находится в школе, то где Джим?”

Задача содержит два утверждения, которые в предикатной форме имеют следующий вид:

$$(\forall x)[V(\text{Джон}, x) \rightarrow V(\text{Джим}, x)],$$
$$V(\text{Джон}, \text{школа}),$$

где x – названия мест, в которых могут находиться Джон и Джим.

Целевая функция, соответствующая вопросу, имеет вид:

$$(\exists x)V(\text{Джим}, x),$$

что буквально можно трактовать следующим образом:

«существует место X , такое, что в нем находится Джим».

Дерево опровержения строится обычным образом и показано на рис.38.1.

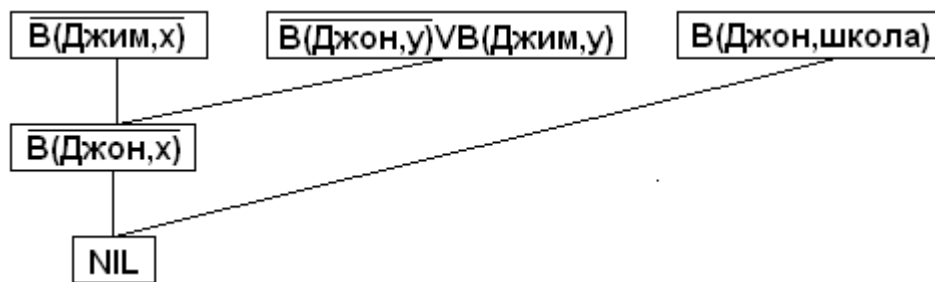


Рис.38.1. Дерево опровержения

Дерево доказательств показано на рис.37.2 и строится следующим образом:

- К каждому предложению, порожденному отрицанием целевой функции, добавляется его отрицание, то есть ППФ преобразуется в тавтологию.
- Вычисляются резолюции, аналогично дереву опровержений.

- В корневой вершине формируется ответ.

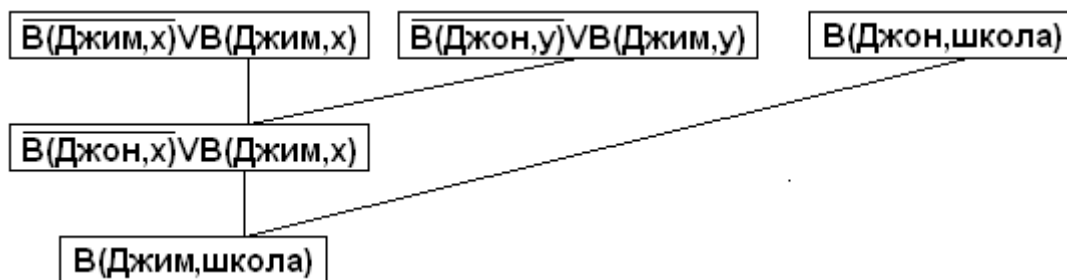


Рис.38.2. Дерево доказательств

39. Элементы логики нечетких множеств

Расширение классической теории множеств привело к понятию «частичной принадлежности» элемента к множеству [14]. Такой подход позволяет принадлежать одному элементу сразу нескольким множествам. Мерой принадлежности элемента x нечеткому множеству A является функция принадлежности:

$$\mu_A(x) \in [0,1].$$

Случай четкой принадлежности соответствует следующим значениям функции принадлежности:

$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

Функция принадлежности может определяться в виде непрерывной функции или в дискретной (лингвистической) форме.

Носитель нечеткого множества:

$$Supp(A) = \{x, \mu_A(x) > 0\}.$$

Кардинальное число нечеткого множества:

$$M(A) = \sum \mu_A(x).$$

Нечеткое множество называется нормальным, если хотя бы один элемент имеет значение функции принадлежности равное 1.

Сечением A_α нечеткого множества A называется подмножество для элементов которого выполняется условие: $\mu_{A_\alpha}(x) > \alpha$.

Пример задания дискретных функций принадлежности:

$$\mu_A(x) = \left\{ \frac{1.0}{x_1} \quad \frac{0.7}{x_2} \quad \frac{0.5}{x_3} \quad \frac{0.1}{x_4} \right\}, \quad \mu_B(x) = \left\{ \frac{0.2}{x_1} \quad \frac{0.5}{x_2} \quad \frac{0.6}{x_3} \quad \frac{0.7}{x_4} \right\}.$$

40. Операции на нечетких множествах

Введение меры нечеткости множеств привело к изменению определений операций с множествами [2,14]. В данном разделе приведены определения 11 операций на нечетких множествах с примерами. Значения функций принадлежности исходных множеств взяты из предыдущего раздела.

1) Логическая сумма множеств $A \cup B$:

$$\mu_{A \cup B}(x) = \mu_A(x) \cup \mu_B(x) = \max(\mu_A(x), \mu_B(x)).$$

Пример: $\mu_{A \cup B}(x) = \left\{ \frac{1.0}{x_1} \quad \frac{0.7}{x_2} \quad \frac{0.6}{x_3} \quad \frac{0.7}{x_4} \right\}.$

2) Логическое произведение множеств $A \cap B$:

$$\mu_{A \cap B}(x) = \mu_A(x) \cap \mu_B(x) = \min(\mu_A(x), \mu_B(x)).$$

Пример: $\mu_{A \cap B}(x) = \left\{ \frac{0.2}{x_1} \quad \frac{0.5}{x_2} \quad \frac{0.5}{x_3} \quad \frac{0.1}{x_4} \right\}.$

3) Отрицание множества $\bar{A}$:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

Пример: $\mu_{\bar{A}}(x) = \left\{ \frac{0}{x_1} \quad \frac{0.3}{x_2} \quad \frac{0.5}{x_3} \quad \frac{0.9}{x_4} \right\}.$

4) Равенство множеств A и B :

$$\mu_A(x) = \mu_B(x).$$

5) Операция концентрации $CON(A)$:

$$\mu_{CON(A)}(x) = \mu_A(x)^2.$$

6) Операция растягивания $DIL(A)$:

$$\mu_{DIL(A)}(x) = \mu_A(x)^{0.5}.$$

7) Алгебраическое произведение множеств $A * B$:

$$\mu_{A*B}(x) = \mu_A(x) * \mu_B(x).$$

8) Ограниченная сумма двух нечетких множеств $A|+|B$:

$$\mu_{A|+|B}(x) = \min(1, \mu_A(x) + \mu_B(x)).$$

9) Ограниченная разность двух нечетких множеств $A|-|B$:

$$\mu_{A|-|B}(x) = \max(0, \mu_A(x) - \mu_B(x)).$$

10) Ограниченное произведение двух нечетких множеств $A|\cdot|B$:

$$\mu_{A|\cdot|B}(x) = \min(0, \mu_A(x) + \mu_B(x) - 1).$$

11) Нормализация нечеткого множества $NORM(A)$:

$$\mu_{NORM(A)}(x) = \frac{\mu_A(x)}{\max(\mu_A(x))}.$$

Множество A называется подмножеством множества B , то есть $A \subset B$, если для всех его элементов выполняется неравенство:

$$\mu_A(x) \leq \mu_B(x).$$

Мера нечеткости множеств основана на сравнении множества с собственным отрицанием. Известно несколько различных мер нечеткости, которые отличаются способом вычисления. Мера нечеткости Р. Егера определяется следующей формулой:

$$FUZ_p(A) = 1 - \frac{D_p(A, \bar{A})}{n^{1/p}},$$

где $D_p(A, \bar{A})$ - метрика расстояния между множествами A и $\bar{A}$, p - размерность метрики. В метрике Хэмминга используется зависимость:

$$D_1(A, \bar{A}) = \sum_{k=1}^n |2 \cdot \mu_A(x_k) - 1|,$$

в метрике Евклида:

$$D_2(A, \bar{A}) = \sqrt{\sum_{k=1}^n (2 \cdot \mu_A(x_k) - 1)^2}.$$

Мера нечеткости Б. Коско основана на вычислении кардинальных чисел для пересечения и объединения множества и его отрицания:

$$FUZ(A) = \frac{M(A \cap \bar{A})}{M(A \cup \bar{A})}.$$

Четкие множества имеют нулевую меру нечеткости.

41. Нечеткие правила вывода

Получение новых нечетких высказываний на основе базисного множества высказываний осуществляется путем применения нечетких правил вывода [2,14]. Нечеткое правило вывода состоит из предпосылки и следствие, которые можно представить в следующем виде:

R: если x это A, то y это B.

В качестве конкретного правила вывода может применяться логическое произведение множеств:

$$\mu_{A \Rightarrow B}(x) = \min(\mu_A(x), \mu_B(x)),$$

или алгебраическое произведение множеств:

$$\mu_{A \Rightarrow B}(x) = \mu_A(x) * \mu_B(x).$$

Система нечеткого вывода образована теорией нечетких множеств, правилами импликации и нечетких рассуждений. Входные четкие величины преобразуются в нечеткие с помощью специальных элементов – фузификаторов. Нечеткая форма переменных позволяет применить к ним

нечеткую импликацию. Формирование четкой выходной величины (агрегирование) осуществляется дефузификатором.

42. Нечеткие нейронные сети

Нечеткие нейронные сети позволяют параллельно вычислять набор нечетких правил вывода для вектора входного образа. Известна структура нечеткой нейронной сети типа TSK (Тагаки, Сугено, Канга) [14], которая имеет слои фуззификации, параметризации, импликации и агрегирования. Выходной сигнал нечеткой нейронной сети определяется следующей формулой:

$$y = \frac{\sum_{i=1}^M c_i \cdot \prod_j^N \mu_{A_i}(x_j)}{\sum_{i=1}^M \prod_j^N \mu_{A_i}(x_j)},$$

где M - количество нечетких правил вывода, N - размер входного вектора, $\{c_i\}$ - множество параметрических весовых коэффициентов.

Обучение нейронной сети сводится к определению значений множества параметрических весовых коэффициентов.

Наиболее проблемным моментом применения нечетких нейронных сетей (и теории нечетких множеств в целом) является определение функций принадлежности, так как до сих пор нет четких методик для задания или вычисления функций принадлежности.

43. Идентификация систем обучаемыми интеллектуальными системами

Задача идентификации системы состоит в том, чтобы по набору соответствующих пар векторов входных и выходных сигналов $\{ (X_j, Y_j) \}_{j=0..N}$ оценить структурные параметры системы W' .

Схема решения этой задачи [18] с помощью нейронной сети W'' приведена на рис.43.1. Параллельно к входам X идентифицируемой, неизвестной системы подключена нейронная сеть; блок обучения получает вектор невязки между выходными сигналами $\varepsilon = Y' - Y''$ и формирует корректирующую матрицу весовых коэффициентов δ .

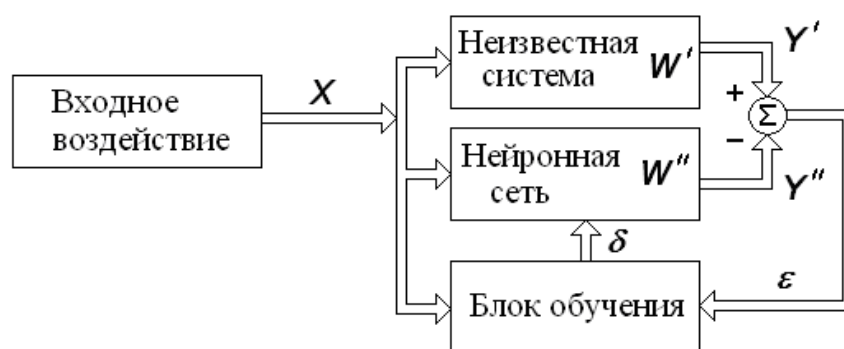


Рис.43.1. Идентификация системы с помощью НС.

В результате обучения нейронная сеть станет функциональной копией идентифицируемой системы, не смотря на внутренние фактические различия. В данном случае последовательность входных воздействий является обучающей выборкой, а выходной вектор реакции идентифицируемой системы – обучающим эталоном ответа. Данный метод будет успешным, если существует нейронная сеть, которая с требуемой точностью аппроксимирует передаточные характеристики исследуемой системы.

По аналогичной методике осуществляется инверсная идентификация, то есть определяются параметры W'' системы с передаточной функцией, обратной по отношению к неизвестной системе W' . Схема инверсной идентификации приведена на рис.43.2. Нейронная сеть включена встречно, по отношению к идентифицируемой системе: входным сигналом для НС является

выходной вектор Y' системы, а невязка определяется как разница между входным сигналом и выходным вектором НС: $\varepsilon = X - Y''$. В качестве эталонов обучающей выборки используется входной сигнал X .

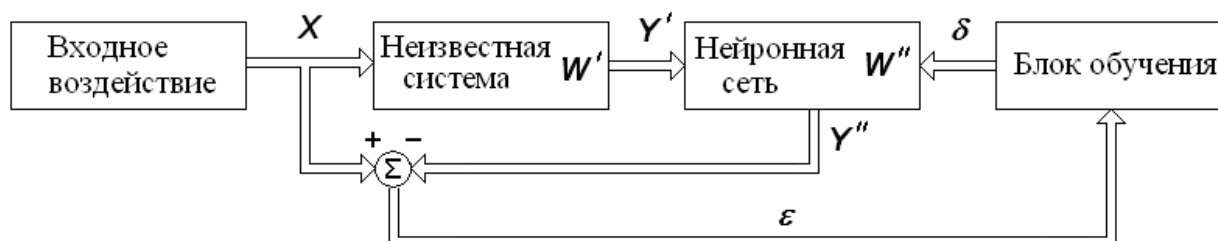


Рис.43.2. Инверсная идентификация системы с помощью НС.

Задачи инверсной идентификации свойственны измерительным системам и элементам противодействия возмущениям и компенсации ошибок. В данных примерах знания о связи входного и выходного сигнала обобщаются в виде параметров НС.

44. Нейросетевая обработка динамических процессов

В качестве входных образов для НС могут выступать не только статические объекты, например изображения, но и временные выборки значений динамических процессов. Входным образом для интеллектуальной системы может служить ряд отсчетов значений процесса, взятых через равные промежутки времени: $\{x(n)\}_{n=0..∞}$. Применение последовательной цепочки единичных элементов задержки позволяет сформировать вектор входных сигналов для многослойной НС, как показано на рис.44.1. Если выходные сигналы НС так же должны образовывать временную последовательность, то эталонные образы естественно формировать аналогичной цепочкой элементов задержки, причем ее длина может отличаться от длины входного вектора. С позиций теории цифровой обработки сигналов такая нейронная сеть реализует

функции цифровой фильтрации, однако, способ настройки – обучение, и расширенные возможности трансформации позволяют отнести такую систему к интеллектуальным устройствам.

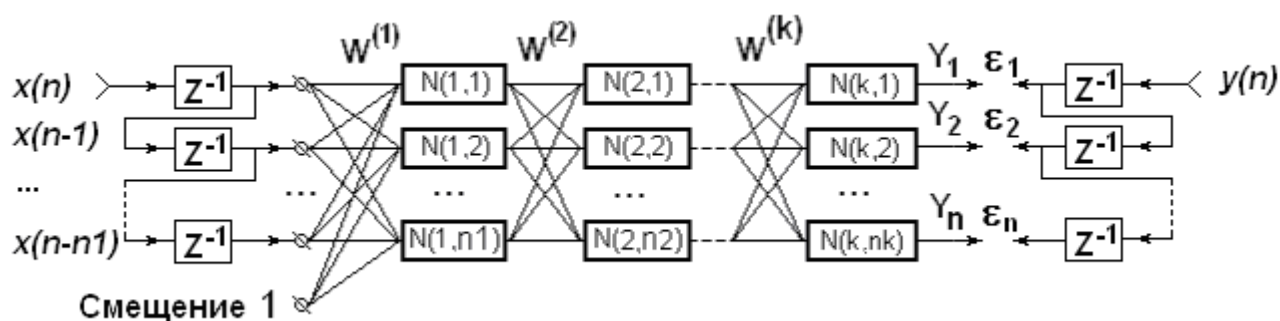


Рис.44.1. Обработка динамического процесса в НС

Величина вектора ошибки вычисляется как разность фактического выходного сигнала и эталона: $\varepsilon = Y^{(k)} - Y(n)$. Способ минимизации этой ошибки определяется методом обучения нейронной сети. Аналитическое описание нейронной и метода обучения сети не отличается от модели НС, приведенной в разделе 19. Дискретная линия задержки на входе системы выполняет функции краткосрочной памяти, содержащей динамически изменяющийся входной процесс.

При перемещении выборок входного процесса по линии задержки каждый такт процесса последовательно попадет на все входы отдельно взятого нейрона, при условии, что НС является полносвязанной сетью.

Вычислительную мощность нейронной сети можно увеличить, если дополнить ее системой рекуррентных (обратных) связей. Пример возможной конфигурации такой НС приведен на рис.44.2. Показана обратная связь только одного выхода НС с входными нейронами, связи других выходов полностью идентичны показанной связи.

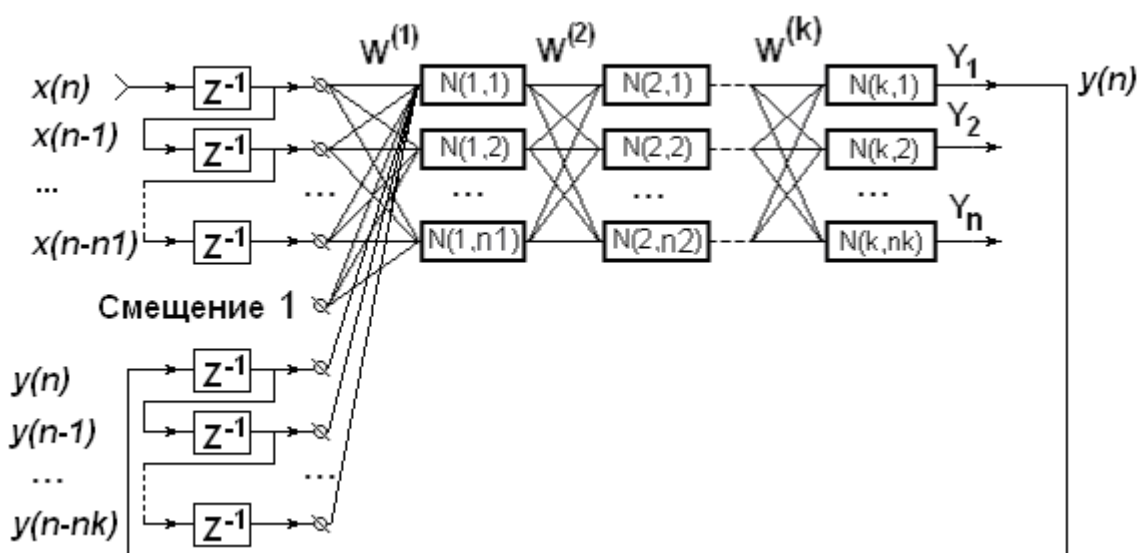


Рис.44.2. Обработка динамического процесса в НС с обратной связью

Рекуррентная связь обладает *памятью конечной длины* для выходных сигналов, что существенно повышает ее вычислительные возможности. Нейронные сети этого типа способны имитировать работу вычислительных машин и могут быть отнесены к множеству конечных автоматов.

Список литературы

1. Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах. М.: ФИЗМАТЛИТ, 2004. 704 с.
2. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. М.: Горячая линия-Телеком. 2002. 382с.
3. Рассел С., Норвиг П. Искусственный интеллект. Современный подход. М.: Издательский дом "Вильямс", 2006. 1408 с.
4. Бровкова М.Б. Системы искусственного интеллекта в машиностроении. Саратов. Саратовский ГТУ. 2004. 119 с.
5. Пономарев О.С. Нечеткие множества в задачах автоматизированного управления и принятия решения. Харьков. 2005. 232 с.
6. Анисимов Б. В. Распознавание и цифровая обработка изображений. - М: Высшая школа, 1983. - 295с.
7. Горелик А. Л., Скрипкин В. А. Методы распознавания. - М: Высшая школа, 1984. - 208с.
8. Гинзбург В. М. Формирование и обработка изображений в реальном времени. - М.: Радио и связь, 1986. - 232с.
9. Фомин В. Н. Математическая теория обучаемых опознающих систем. - Л.: Ленинградский ун-т, 1976. - 236с.
10. Меламед И. И. Нейронные сети и комбинаторная оптимизация. // Автоматика и телемеханика. - 1994. - №10. - С. 3-40.
11. Нильсон Н. Принципы искусственного интеллекта. -М.: Мир, 1985. – 376с.
12. Искусственный интеллект. Справочник в 3-х книгах. // Под ред. Поспелова Д. А. - М.: Наука, 1990.
13. Дуда Р., Харт П. Распознавание образов и анализ сцен. – М.: Мир, 1976. – 464с.
14. Осовский С. Вычислительные нейронные сети. – М.: Финансы и статистика, 2002. – 344с.
15. Люгер Дж.Ф. Искусственный интеллект: Стратегии и методы решения сложных проблем.: - М: Вильямс, 2003 г. 864 с.
16. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Радио и связь, 2004 - 383с.
17. Джонс М. Т. Программирование искусственного интеллекта в приложениях. – М.: ДМК, 2004 - 311 с.
18. Хайкин С. Нейронные сети: полный курс. М.: «Вильямс». 2006. 1104 с.
19. Капля В.И. Системы искусственного интеллекта: учебное пособие. Волгоград.: ИУНЛ ВолгГТУ. 2011. – 97 с.
20. Математическая энциклопедия. Том 4. Гл. ред. И.М. Виноградова.- М.: «Советская энциклопедия». 1984. 1216 с.
21. Джексон П. Введение в экспертные системы. М.:«Вильямс».2001. 624с.
22. Гурский Д.А., Турбина Е.С. Вычисления в MathCad 12. СПб.: «Питер», 2006. 544 с.
23. Братко И. Алгоритмы искусственного интеллекта на языке PROLOG. М.: «Вильямс». 2004. 640 с.
24. Малпас Дж. Реляционный язык Пролог и его применение. М.: Наука. 1990. 464 с.

Электронное учебное издание

Виктор Иванович **Капля**

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Учебное пособие

Электронное издание сетевого распространения

Редактор Н.И. Матвеева

Темплан 2018 г. Поз. № 20.

Подписано к использованию 22.02.2018 г. Формат 60×84 1/16.

Гарнитура Times. Усл. печ. л. 5,88.

Волгоградский государственный технический университет
400005, г. Волгоград, пр. Ленина, 28, корп. 1.

ВПИ (филиал) ВолгГТУ.
404121, г. Волжский, ул. Энгельса, 42а.