

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования «Нижегородский государственный
университет им. Н.И. Лобачевского»**

П.Е. Овчинников

ПРИМЕНЕНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОБРАБОТКИ СИГНАЛОВ

Учебно-методическое пособие

Рекомендовано методической комиссией физического факультета для
студентов ННГУ, обучающихся по направлению 230400 «Информационные
системы»

Нижегород
2012

УДК 519.234
ББК 22.1

ПРИМЕНЕНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОБРАБОТКИ СИГНАЛОВ. Автор: Овчинников П.Е.. Учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2012. – 32 с.

Рецензент: к.ф.-м.н., с.н.с. НИФТИ ННГУ Н.С. Будников

Учебно-методическое пособие (УМП) соответствует тематике учебно-научного инновационного комплекса УНИК-2 – «Разработка методов получения, обработки, хранения и передачи информации, включая диагностику природных сред, искусственных материалов и живых систем». Комплекс УНИК-2 развивается в рамках приоритетного направления развития ННГУ как национального исследовательского университета «Информационно-телекоммуникационные системы: физические и химические основы, перспективные материалы и технологии, математическое обеспечение и применение», представляющего интерес для развития системы образования и повышения качества подготовки специалистов в ННГУ.

Пособие состоит из трех разделов, в которых приведены основные понятия искусственных нейронных сетей, дано описание групп классификаторов и рассмотрены вопросы применения нейросетей в задачах обработки сигналов.

УМП предназначено для знакомства студентов с нейросетевыми методами обработки данных и применением их в задачах обработки сигналов, для приобретения начальных навыков в реализации нейросетевых алгоритмов.

Настоящее УМП может быть успешно использовано при подготовке и проведении практических занятий для студентов старших курсов, бакалавров, магистрантов и аспирантов физического факультета ННГУ. УМП дополнит знания, даваемые студентам в рамках обучения бакалавров и магистров по направлению 230400 «Информационные системы».

Введение

Начиная с ранних работ Винера по адаптивным фильтрам [1] для определения параметров адаптивных систем обработки сигналов, включая искусственные нейронные сети (ИНС), в качестве критерия оптимальности почти исключительно использовалась среднеквадратическая ошибка. Основаниями для этого были, прежде всего, аналитическая простота и предположение о соответствии большинства случайных процессов в природе распределению Гаусса. Гауссово распределение вероятностей полностью описывается статистиками первого и второго порядка. Следовательно, в предположении гауссовости среднеквадратичной ошибки достаточно для извлечения из набора данных, полностью описываемых средним и дисперсией, всей возможной информации. В то же время не для всех процессов описание может быть ограничено статистиками второго порядка, поэтому велись и ведутся разработки алгоритмов, не ограничивающихся гауссовой моделью. Одним из направлений исследований в данной области является применение ИНС для обработки сигналов. Понятие нейронной сети было формализовано Маккалоком и Питтсом [2]. Хебб предложил модель человеческого обучения, ставшую основой методов обучения ИНС (правила Хебба) [3]. Затем Розенблатом была предложена однослойная сеть – персептрон, которая легла в основу большинства нейросетевых методов [4]. Хопфилд предложил ИНС для вычислительно эффективной минимизации квадратичной нормы при распознавании образов [5], Кохонен разработал сеть для кластеризации многомерных данных. Однако наибольшее распространение ИНС получили как универсальные аппроксиматоры функций многих переменных. Теоретическим основанием применения ИНС для аппроксимации функций многих переменных является работа А.Н. Колмогорова [6]. На практике ИНС стали широко применяться в этой области после создания группой авторов алгоритма обратного распространения ошибки для обучения многослойных сетей [7]. Среди отечественных учёных, развивающих теорию ИНС, А.И. Галушкин [8], А.Н. Горбань, В.Л. Дунин-Барковский [9]. Применение ИНС в задачах обработки сигналов позволяет отказаться от предположений линейности, стационарности и гауссовости. В частности это достигается благодаря использованию при обучении информационных критериев, основанных на понятии энтропии. Понятие энтропии распределения вероятностей введено Шенноном [10] и представляет собой скалярную величину, отражающую количество информации, содержащееся в распределении. Энтропия по определению относится ко всему распределению, а не к выделенным статистикам.

ИНС не содержат качественных ограничений на модель процесса, присущих традиционным методам обработки сигналов. ИНС – это параллельная и распределённая система обработки информации, состоящая из простых вычислительных элементов: искусственных нейронов, соединённых связями друг с другом и с входами и выходами сети. При

распространении сигналов по сети значение сигнала, передаваемого от нейрона к нейрону, умножается на весовой коэффициент, отдельный для каждой связи. Значения весовых коэффициентов определяются из набора экспериментальных данных, эту процедуру называют обучением. Модель процесса, соответствующая экспериментальным данным, формируется при обучении ИНС.

1. ИНС, основные понятия

1.1.1. Узлы искусственных нейронных сетей

В основе искусственных нейронных сетей лежит структура из простых нелинейных вычислительных элементов. Элемент (также называемый узлом) такого типа изображён на рис. 1. У узла есть N входов, обозначенных x_1, x_2, \dots, x_N , которые суммируются с весами w_1, w_2, \dots, w_N , сумма смещается и нелинейно преобразуется, давая выход y :

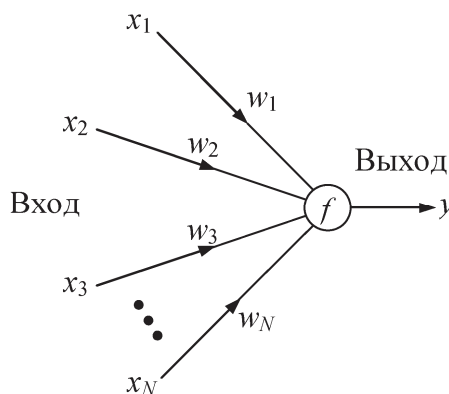


Рис. 1. Узел ИНС

$$y = f\left(\sum_{i=1}^N w_i x_i - \varphi\right) \quad (1)$$

где φ – внутренний порог или смещение, f – нелинейность одного из следующих типов:

Жёсткий ограничитель:

$$f(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (2)$$

Сигмовидные (sigmoid) функции:

$$f(x) = \tanh(\beta x), \quad \beta > 0 \quad (3)$$

или

$$f(x) = \frac{1}{1 + e^{-\beta x}}, \quad \beta > 0 \quad (4)$$

Сигмовидные нелинейности используются чаще, поскольку они непрерывно дифференцируемы. Более сложные узлы могут включать интегрирование по времени или другие типы зависимостей от времени и более сложные математические операции, чем суммирование [11].

В радиально-базисных сетях (RBF) и сетях Кохонена вместо взвешенной суммы вычисляется евклидово расстояние между входным и весовым векторами:

$$S^2 = \sum_{i=1}^N (x_i - w_i)^2 \quad (5)$$

Для RBF сетей, а в ряде случаев и для сетей Кохонена, в качестве функции активации используется функция Гаусса:

$$f(S) = \exp\left(\frac{-S^2}{2\sigma^2}\right) \quad (6)$$

здесь σ – среднеквадратичное отклонение, характеризующее ширину купола радиально-базисной функции.

1.1.2. Топологии нейронных сетей

По топологии ИНС можно разделить на четыре группы:

1. Однослойные и многослойные сети прямого распространения: персептроны.
2. Рекуррентные сети: Хопфилда (Hopfield), Хэмминга (Hamming), сети адаптивного резонанса (ART) [12].
3. Самоорганизующиеся сети Кохонена (Kohonen) [13].
4. Гибридные сети: радиально-базисные сети, иерархические классификаторы [14].

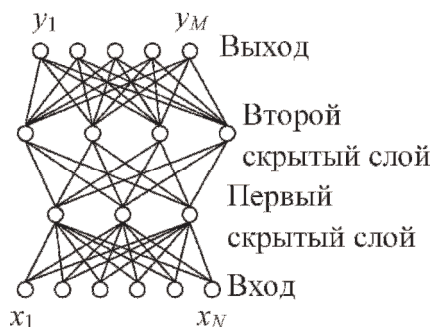


Рис. 2. Многослойный персептрон

Работа однослойного персептрона может быть описана как умножение входного вектора на матрицу весовых коэффициентов. В многослойных персептронах (МСП) выход узлов одного слоя образует вход для узлов следующего слоя. Пример МСП представлен на рис. 2. Особенность многослойных персептронов в нелинейности на каждом уровне. Эти нелинейности образуют карту соответствий выходов входам, что в частности даёт возможность использовать персептрон как классификатор. Более подробно работа МСП в качестве классификатора рассмотрена далее в соответствующем разделе.

В рекуррентных сетях выходы узлов соединяются с входами. В частности, в сети Хопфилда вход каждого из узлов соединён с выходами всех узлов. Входы и выходы зависят от времени: $x_i(t)$, $y_i(t)$. Вес, с которым соединены i -ый и j -ый узлы, обозначен w_{ij} . Тогда выход i -того узла в момент времени t можно записать так:

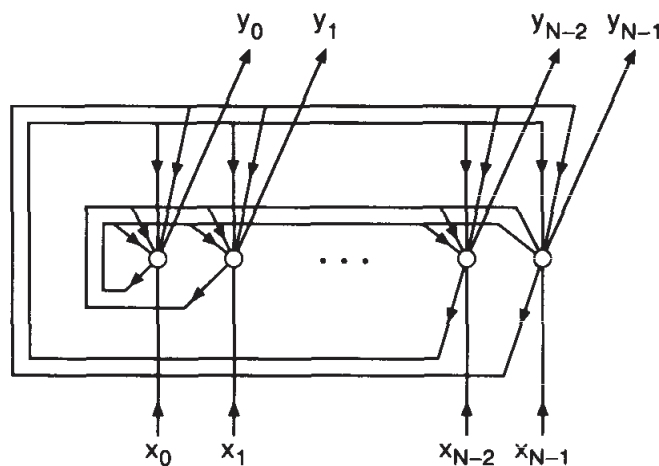


Рис. 3. Рекуррентная искусственная нейронная сеть

$$y_i(t) = f[x_i(t) + \sum_j w_{ij} y_j(t-1) - \varphi] \quad (7)$$

Рекуррентная сеть с N входами и N выходами изображена на рис. 3. Важным свойством сети Хопфилда является то, что если $w_{ij}=w_{ji}$ и вычисления происходят асинхронно для некоторого постоянного входа, то сеть может перейти в равновесное состояние, когда $y_i(t)=y_i(t-1)$ для всех i . Эти равновесные состояния сети могут быть использованы как ассоциативная память в приложениях, которые имеют фиксированный набор шаблонов для сравнения (например, печатные буквы) [5].

Более подробно сети прямого распространения, рекуррентные сети, самоорганизующиеся сети, а также другие сети, применяемые в задачах классификации, распознавания и детектирования, описаны в следующем разделе.

2. Классификация образов при помощи ИНС

Цель классификации – отнести входной образ к одному из M известных классов. Далее будем считать, что входной образ представляет собой вектор \vec{x} , состоящий из N действительных значений:

$$\vec{x} = \{x_1, x_2, \dots, x_N\} \quad (8)$$

Элементы вектора представляют собой характеристики объекта, выбранные в качестве признаков для классификации. Входной образ можно понимать как точку во входном пространстве размерности N . Назначение классификатора – разделить это многомерное пространство на области, относящиеся к классам, и определять, к какой области относится тот или иной входной образ. В традиционных Байесовых классификаторах для описания классов используют функции плотности распределения вероятностей входных значений и Байесову теорию принятия решений для формирования границ областей классов по этим плотностям вероятностей [15]. При построении адаптивных непараметрических классификаторов функции плотностей вероятностей в явном виде не оцениваются, вместо этого для формирования областей принятия решений используются классифицирующие функции. Применение классификатора образов прежде всего требует выбора характеристик объекта. Характеристики должны содержать достаточно информации для различения между классами и при этом не зависеть от характеристик, не относящихся к данной задаче классификации. Кроме того характеристик должно быть как можно меньше, чтобы не снижать производительность и не увеличивать требуемый объем обучающих данных. Измерение характеристик должно производиться автоматически, без обработки вручную, чтобы исключить возможность внесения ложных предположений. После выбора характеристик необходимо сформировать отдельные наборы образцов для обучения и тестирования. При обучении ограниченное количество данных (обучающая выборка) и априорные знания о предметной области используются для определения структуры классификатора и настройки его параметров. После обучения полученный классификатор тестируется на данных из второго набора. Новые для классификатора образцы (из выборки для тестирования)

Группа	Область решений	Вычислительный элемент	Классификаторы
Вероятностные		В зависимости от распределения	Сумма гауссиан
Гиперплоскостные		Сигмовидная функция	МСП, машина Больцмана
Ядерные		Ядро	Метод потенциальных функций, СМАС
По образцу		Эвклидова метрика	k-ближайших, LVQ
Иерархические	Зависит от структуры	Зависит от структуры	DBNN, МОЕ, experts-on-class, classes-in-expert

Рис. 4. Группы классификаторов

классифицируются, и по доле ошибок классификации оценивается способность полученного классификатора к обобщению.

Для настройки параметров классификаторов и кластеризаторов используются методы обучения с учителем, без учителя, а также смешанные методы. Обучение с учителем подразумевает включение в обучающую выборку дополнительной информации о принадлежности образцов к классам (метка или требуемый выход классификатора для каждого образца из обучающей выборки). Алгоритмы кластеризации (векторного квантования) используют процедуры обучения без учителя, группируя непомеченные данные в кластеры. Смешанное обучение классификаторов, как правило, состоит из двух этапов: сначала проводится кластеризация непомеченных данных, затем найденные кластеры помечаются. В качестве пометки найденные кластеры могут быть заменены небольшим количеством помеченных данных. Сбор и ручная пометка образцов для обучения весьма трудоёмки, поэтому смешанное обучение применяется с целью уменьшения количества требуемых для обучения помеченных данных.

Нейросетевые классификаторы отличаются друг от друга по способу формирования областей принятия решений. По этому способу классификаторы могут быть разделены на пять больших групп, изображенные на рис. 4. В левой колонке приведены названия групп классификаторов; во второй колонке изображены примеры формирования регионов принятия решений; в третьей колонке представлены вычислительные элементы (узлы ИНС), реализующие вычисления на самом низком уровне; в правой колонке приведены названия характерных классификаторов для каждой из групп. Четыре первые группы образованы классификаторами, в составе которых есть только одна ИНС. Верхняя группа содержит традиционные вероятностные (Байесовы) классификаторы, три следующие группы содержат адаптивные классификаторы. Эти классификаторы позволяют формировать произвольные регионы принятия

решений, могут быть реализованы с высокой степенью параллелизма вычислений и используют сравнительно несложные вычисления для последовательной адаптации. Пятая группа образована классификаторами, состоящими более чем из одной ИНС и имеющими иерархические структуры: сети, основанные на решении (decision-based neural network – DBNN); смесь экспертов (mixture of experts – MOE network); эксперты в классе (experts-in-class network); классы в эксперте (classes-in-expert network) [14].

2.1. Вероятностные классификаторы

Использование вероятностных классификаторов предполагает априорное знание распределений вероятностей для входных характеристик. Чаще всего используется гауссово распределение или сумма гауссовых распределений. Параметры распределений, как правило, определяются в процессе обучения с учителем, при этом все данные для обучения должны быть доступны одновременно. Эти классификаторы обеспечивают оптимальное качество классификации, если используемые распределения являются точной моделью тестовых данных и доступно достаточное для точного определения параметров количество обучающих данных. Эти два условия часто не выполняются для нестационарных систем и данных реального мира (результатов измерений).

2.2. Гиперплоскостные классификаторы (Hyperplane Classifiers)

Гиперплоскостные классификаторы формируют сложные области принятия решений, используя узлы, которые формируют границы принятия решений в виде гиперплоскостей в пространстве входов. Как правило, узлы вычисляют нелинейную функцию от взвешенной суммы входных значений. Чаще всего используется сигмовидная нелинейность, но также используются другие нелинейности, включая полиномы высокого порядка. Эти классификаторы имеют малую вычислительную сложность и требуют мало памяти в процессе классификации, но могут требовать много времени для обучения и/или сложные алгоритмы обучения. Сюда входят многослойные перцептроны, машины Больцмана [16], классифицирующие бинарные деревья [17], сети высокого порядка [18], сети, формируемые методом Group Method of Data Handling (GMDH) [19].

2.2.1. Сеть прямого распространения (многослойный перцептрон)

Сеть прямого распространения, или многослойный перцептрон (МСП), представляет собой структуру из простых вычислительных элементов (узлов/нейронов), сгруппированных в соединённые последовательно слои. В пределах слоя узлы работают параллельно. Работа перцептрона описывается следующими выражениями:

$$S_{jl} = \sum_i w_{ijl} \cdot x_{ji} - \phi_{jl} \quad (9)$$

$$y_{jl} = f(S_{jl}) \quad (10)$$

$$x_{ij(l+1)} = y_{il}, \quad (11)$$

здесь i – номер входа, j – номер нейрона в слое, l – номер слоя, x_{ijl} – входы узлов, y_{jl} – выходы узлов, w_{ijl} – весовые коэффициенты, ϕ_{jl} – пороговые уровни узлов, f – нелинейная функция активации.

Классификаторы на основе многослойных персептронов с сигмовидными функциями активации формируют области принятия решений сложной формы [8]. МСП обучается с учителем. Наиболее распространены способы обучения на основе градиентного спуска, при этом отыскивается минимум функции ошибки. Чаще всего в качестве функции ошибки используется или суммарное квадратичное отклонение выходов персептрона от требуемых значений из выборки для обучения, или взаимная энтропия. В задачах классификации требуемые выходные значения задаются малыми (обычно, нулевыми) для всех узлов выходного слоя, кроме одного узла, соответствующего классу, к которому относится текущий образ (входной вектор). Для соответствующего текущему классу узла требуемое выходное значение задаётся высоким, как правило, единицей. Каждый узел выходного слоя вычисляет классифицирующую функцию, которая разделяет все входные на два группы: относящиеся к соответствующему узлу классу и относящиеся к другим классам. Эту схему обучения называют «1 из N-классификатор». Было показано [20], что при таком выборе требуемых значений при обучении выходы многослойного персептрона аппроксимируют апостериорные вероятности отнесения к классам. Точность аппроксимации возрастает при увеличении обучающей выборки.

На ранней стадии интерес к МСП был вызван предположением о его применимости к биологическим нейронным сетям. И хотя это предположение не подтвердилось, классификаторы с МСП успешно применяются в различных областях. МСП, обучаемые алгоритмом обратного распространения ошибки (ОРО), были использованы для классификации фонем, в локационных, медицинских и других задачах обработки сигналов [21].

На рис. 5 показаны примеры использования МСП с ОРО для аппроксимации функций. В примерах использован персептрон с одним входом, двадцатью узлами с сигмовидными функциями активации в первом скрытом слое, пятью – во втором и одним выходным узлом с линейной функцией активации. Входное значение на каждом шаге обучения выбиралось случайным образом из аппроксимируемого интервала. Весовые коэффициенты изменялись на каждом шаге. Функции «ступенька» и гауссов пик аппроксимировались всё более точно по мере возрастания числа шагов обучения. На рисунке сплошными линиями показаны аппроксимируемые кривые, а точками и штрихами – результаты аппроксимации.

Несмотря на то, что, основываясь на теореме Колмогорова [6], можно утверждать, что многослойный персептрон может быть использован для аппроксимации сколь угодно сложных функций нескольких переменных, анализу аппроксимационных возможностей МСП посвящен ряд теоретических работ. Независимо были разработаны схожие доказательства, например [11], того, что двух слоёв с функциями активации в виде ступеньки достаточно для формирования многослойным персептроном произвольных областей принятия решений. Позже было доказано [22], что для формирования произвольных областей принятия решений достаточно персептрона с одним скрытым слоем с сигмовидными функциями активации, при этом необходимое число узлов в скрытом слое должно быть определено для каждой конкретной задачи. Часто необходимый для решения задачи размер сети определяется экспериментальным образом. Так, например, в работе [23] предлагается метод определения способности к обобщению нейронной сети не на основе традиционного вероятностного подхода, а с использованием концепции *information-gap theory*. МСП обучается стандартным образом. После обучения определяется возможность сети к обобщению, при этом для оценки используется мера неопределённости во входных данных. В работе [24] предложен метод на основе сингулярного разложения для оценки обобщающей способности обученной сети с одним скрытым слоем. Такая оценка позволяет определить оптимальное число нейронов в скрытом слое, обеспечивающее максимальную обобщающую способность классификатора. Также существуют методы обучения, включающие в себя подбор оптимального числа узлов в скрытом слое [25].

Нужно отметить, что классификаторы с ОРО обучаются сравнительно долго. Время обучения возрастает с усложнением геометрии областей принятия решений и увеличением числа слоёв. Как и для других классификаторов, скорости обучения и классификации для МСП увеличиваются, если размер сети подобран оптимально для данной задачи: достаточный для точной аппроксимации и не слишком большой, чтобы ограничить число определяемых параметров (весовых коэффициентов). Были предложены различные способы ускорения обучения; чаще всего

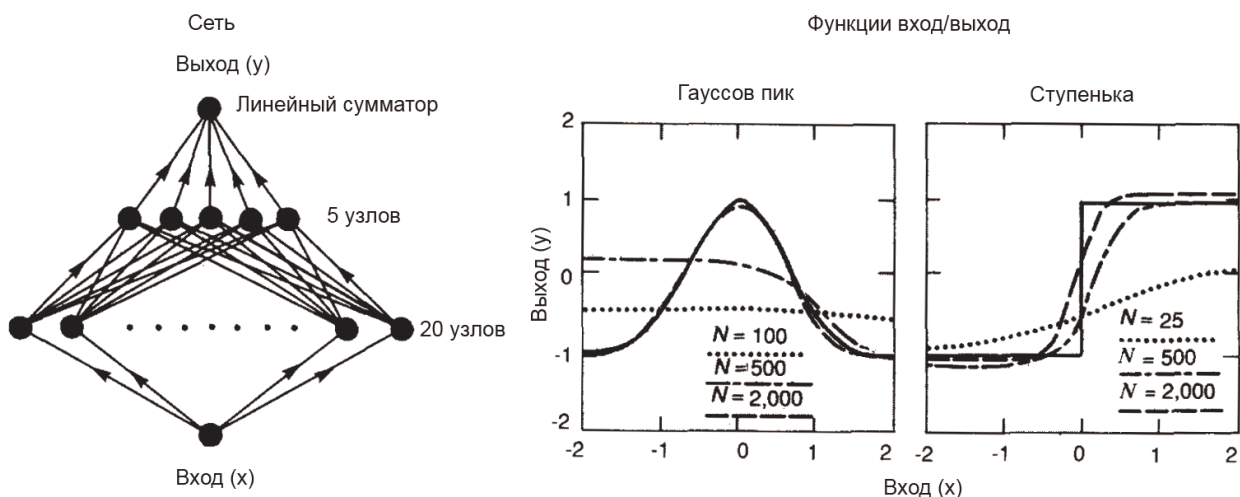


Рис. 5. Примеры аппроксимации

используются представление данных для обучения в случайном порядке и нормировка входных образов. Кроме того, разработаны более быстрые модификации алгоритма ОРО (см. далее).

Важной характеристикой ОРО является то, что полученные с помощью него классификаторы может быть трудно интерпретировать, то есть описать сформированные решающие правила. Ряд работ посвящен извлечению решающих правил из обученных классификаторов, например [26], [27]. В [28] показано, что обученная сеть прямого распространения может быть инвертирована для получения набора данных, соответствующего требуемому выходу сети (в частности, может быть получен набор данных, относящихся к одному классу).

2.2.1.1. Обучение

Построение алгоритма обучения начинается с определения функции ошибки, которая должна быть минимизирована путём настройки весовых коэффициентов сети. Различают два принципиальных подхода: метод поиска и методы, основанные на градиентном спуске. Поисковые методы выборочно исследуют функцию ошибки путём случайных изменений весовых коэффициентов сети. К поисковым методам относятся простой метод Монте-Карло, имитация отжига, различные варианты эволюционных алгоритмов формирования и настройки ИНС [29], [30]. Несмотря на то, что поисковые методы наиболее подходят для параллельных вычислений, они тратят большую часть времени на обсчёт неудачных попыток. Для непрерывно дифференцируемых функций ошибки возможно применение методов градиентного спуска, которые сходятся значительно быстрее поисковых методов, особенно если число параметров велико (>100). Многослойные сети прямого распространения с сигмовидными функциями активации попадают в эту категорию.

В большинстве случаев градиентные методы обучения минимизируют суммарную квадратичную ошибку. Для обучения сетей прямого распространения наиболее часто применяется алгоритм обратного распространения ошибки и его модификации (в том числе эвристические). Модификации разрабатываются для уменьшения времени обучения и улучшения качеств обученной сети. Например, предложен адаптивный метод обратного распространения ошибки, основанный на теории устойчивости Ляпунова. При обучении весовые коэффициенты нейронной сети изменяются так, чтобы уменьшать значение функции Ляпунова [31]. Наиболее успешные вариации алгоритма обратного распространения ошибки – QUICKPROP и RPROP [32], [33]. И хотя эти эвристические алгоритмы не до конца теоретически обоснованы, они, как правило, превосходят сложные алгоритмы нелинейной оптимизации. Помимо модифицированных методов минимизации функции ошибки для уменьшения времени обучения также разработан ряд способов начального выбора значений весовых коэффициентов сети [34], [35].

Большинство методов обучения – итеративные, основанные на градиентном спуске. Для сетей с одним скрытым слоем существует неитеративный алгоритм обучения ELM (extreme learning machine). Весовые коэффициенты определяются по обучающей выборке путём вычисления псевдообратной матрицы. На основе алгоритма ELM, требующего всю обучающую выборку сразу, разработан метод обучения для случая последовательного поступления обучающих данных [36].

Кроме поисковых и градиентных методов существуют также их комбинации. В [37] предлагается метод оптимизации многослойного персептрона, производящий одновременно и оптимизацию структуры сети и весовых коэффициентов. Метод представляет собой комбинацию имитации отжига, метода штрафов (tabu search) и обратного распространения ошибки. Целью оптимизации сети является повышение эффективности классификации при понижении числа узлов.

Метод обратного распространения ошибки

Пусть задан многослойный персептрон с непрерывно дифференцируемой функцией активации. Сеть задается вектором параметров – набором весовых коэффициентов и пороговых уровней:

$$\vec{P} = \begin{pmatrix} \vec{W} \\ \vec{\Phi} \end{pmatrix} \quad (12)$$

где \vec{W} – вектор, компоненты которого – все весовые коэффициенты сети; $\vec{\Phi}$ – вектор пороговых уровней сети. Таким образом, если считать обучающее множество заданным, то ошибка сети зависит только от вектора параметров: $E = E(\vec{P})$.

При обучении на каждой итерации параметры корректируются в направлении антиградиента E :

$$\Delta \vec{P} = -\varepsilon \nabla E(\vec{P}) \quad (13)$$

В теории оптимизации доказано, что такая процедура обеспечивает сходимость к одному из локальных минимумов функции ошибки, при условии правильного выбора ε на каждой итерации. Такой метод оптимизации называется методом наискорейшего спуска. Коррекции необходимо рассчитывать на каждой итерации. Поэтому каждая итерация требует расчета градиента ошибки и выбора оптимального шага ε . Слишком малый шаг замедляет обучение, а слишком большой приводит к нестабильности: значение функции ошибки сильно колеблется от итерации к итерации. Выбору шага посвящен ряд работ, в частности в [38] предложен способ адаптивного выбора шага градиентного спуска, основанный на функции Ляпунова. Другой способ стабилизации и ускорения обучения – это добавление инерционного слагаемого (momentum) в выражение для коррекции параметров. Сходимость градиентного метода с моментом для двухслойной сети рассмотрена в [39].

Прямое вычисление градиента связано с большим объёмом вычислений, так как требует многократного (пропорционально числу параметров сети) вычисления выходных значений нейронов на каждой итерации. Алгоритм обратного распространения ошибки учитывает структуру сети и позволяет отказаться от многократного вычисления откликов нейронов на каждом шаге градиентного метода оптимизации.

Идея метода состоит в том, чтобы представить E в виде сложной функции и последовательно рассчитать частные производные как для сложной функции.

Запишем (13) для весовых коэффициентов:

$$\Delta w_{ijl} = -\varepsilon \left(\frac{\partial E(W, \Phi)}{\partial w_{ijl}} \right) \Big|_{W, \Phi} \quad (14)$$

Составим выражение для производной сложной функции:

$$\frac{\partial E}{\partial w_{ijl}} = \frac{\partial E}{\partial y_{jl}} \cdot \frac{dy_{jl}}{dS_{jl}} \cdot \frac{\partial S_{jl}}{\partial w_{ijl}} \quad (15)$$

Такое выражение требует вычисления производной от функции активации. Поэтому для работы алгоритма необходимы нелинейности с непрерывной первой производной.

Теперь вычислим множители правой части (15).

$$\frac{dy_{jl}}{dS_{jl}} = \frac{df(S_{jl})}{dS_{jl}} \quad (16)$$

Для сигмоиды и гиперболического тангенса производные легко вычисляются через значения выходов нейронов.

Для сигмоиды: $\frac{df(S_{jl})}{dS_{jl}} = y_{jl}(1 - y_{jl})$

Для гиперболического тангенса: $\frac{df(S_{jl})}{dS_{jl}} = 1 - y_{jl}^2$

$$\frac{\partial S_{jl}}{\partial w_{ijl}} = x_{ijl} \quad (17)$$

Первый множитель в (15) вычисляется по-разному в зависимости от слоя: для последнего слоя он получается непосредственным дифференцированием функции ошибки.

Для остальных (внутренних) слоёв:

$$\frac{\partial E}{\partial y_{jl}} = \sum_k \frac{\partial E}{\partial x_{jk(l+1)}} = \sum_k \frac{\partial E}{\partial y_{k(l+1)}} \cdot \frac{dy_{k(l+1)}}{dS_{k(l+1)}} \cdot \frac{\partial S_{k(l+1)}}{\partial x_{jk(l+1)}} = \sum_k \frac{\partial E}{\partial y_{k(l+1)}} \cdot \frac{dy_{k(l+1)}}{dS_{k(l+1)}} \cdot w_{jk(l+1)}, \quad (18)$$

здесь суммирование по k – это суммирование по узлам следующего (верхнего) слоя, с которыми соединён выход данного узла; соответственно $w_{jk(l+1)}$ – вес соединения j -узла данного слоя с k -узлом следующего слоя.

Подставим (15)–(18) в (13) и, введя обозначения:

для выходного слоя:

$$\delta_j = -\frac{df(S_j)}{dS_j} \frac{\partial E}{\partial y_j}, \quad (19)$$

для внутренних слоёв:

$$\delta_{jl} = \frac{df(S_{jl})}{dS_{jl}} \sum_k \delta_{k(l+1)} w_{jk(l+1)}, \quad (20)$$

получим простое выражение для рекурсивной коррекции весов:

$$\Delta w_{ijl} = \varepsilon \delta_{jl} x_{ijl} \quad (21)$$

Для вычисления коррекций пороговых уровней их можно считать весовыми коэффициентами связей с постоянным входом $x=1$, тогда:

$$\Delta \phi_{jl} = \varepsilon \delta_{jl} \quad (22)$$

Применение суммарной квадратичной ошибки в качестве функции ошибки

В качестве целевой функции возьмём суммарную квадратичную ошибку:

$$E = \frac{1}{2} \sum_j \sum_s (y_j^s - d_j^s)^2 \quad (23)$$

$$\frac{\partial E}{\partial y_{jl}} = \sum_s (y_j^s - d_j^s), \quad (24)$$

здесь индекс s нумерует образцы, предъявляемые сети, и соответствующие требуемые ответы d , j – номер выхода персептрона, y – значение на выходе персептрона (результат прямого прохода для данного образца).

Из выражений (19) и (24) выражаются дельта-множители для выходного слоя:

$$\delta_j = \frac{df(S_j)}{dS_j} (d_j - y_j) \quad (25)$$

Для остальных слоёв дельта-множители вычисляются рекурсивно, согласно выражению (20).

Применение взаимной энтропии в качестве функции ошибки

Рассмотрение процесса обучения со статистической точки зрения показывает, что применение взаимной энтропии в качестве функции ошибки обладает рядом преимуществ по сравнению с традиционным использованием суммарной квадратичной ошибки.

Главная цель обучения нейронной сети – это построение статистической модели процесса формирования данных. Обученная нейросеть должна оценивать совместную вероятность, $p(\vec{x}, \vec{d})$ где $\vec{x} = (x_1, x_2, \dots, x_m)$ – входной вектор, $\vec{d} = (d_1, d_2, \dots, d_c)$ – требуемый выходной вектор. Совместная вероятность может быть расписана как:

$$p(\vec{x}, \vec{d}) = p(\vec{d} | \vec{x})p(\vec{x}). \quad (26)$$

Можно показать, что дельта-множители для выходного слоя вычисляются так (индекс n нумерует образцы):

$$\delta_z = \sum_{n=1}^N \{d_{zn} - y_{zn}\} \quad (27)$$

Если в качестве функции активации используется гиперболический тангенс, то аналогично можно получить:

$$\delta_z = \sum_{n=1}^N \left\{ d_{zn} - y_{zn} + \frac{d_{zn}}{y_{zn}} - 1 \right\} \quad (28)$$

Для остальных слоёв дельта-множители вычисляются рекурсивно, согласно выражению (20).

В большинстве случаев применение такой функции ошибки обеспечивает повышение качества классификации и скорости обучения по сравнению с минимизацией суммарной квадратичной ошибки [40].

Регуляризация

В описанных выше методах обучения функция ошибки в явном виде зависит только от обучающей выборки и не содержит ограничений на весовые коэффициенты сети. Было показано, что способность сети к обобщению зависит от величин весовых коэффициентов сети: чем больше сумма значений весов, тем больше ошибка классификации. Это связано с тем, что при уменьшении значений весовых коэффициентов ИНС формирует более гладкое отображение «вход-выход». Чтобы учесть это при обучении, в функцию ошибки добавляется регуляризирующее слагаемое:

$$E = E_T + \beta E_{\vec{w}} \quad (29)$$

Здесь E_T – функция, отражающая соответствие поведения сети обучающей выборке (например, суммарное квадратичное отклонение), $E_{\vec{w}}$ – регуляризирующая функция. В простейшем случае применяется сумма квадратов весовых коэффициентов, такая регуляризация снижает вероятность переобучения сети.

Примерами регуляризации обучения ИНС являются: двойное ОРО (double backpropagation – DBP) [41] и возмущение входов (input perturbation – IP) [42]. Оба метода направлены на приведение ИНС к состоянию, в котором малому изменению входных сигналов соответствует малое изменение отклика сети. Это делается путём добавления в минимизируемый функционал ограничений на производные аппроксимируемой сетью функции (якобиан) [43]. Также сглаживание отображения применено в [44] для определения структуры сети, названной обобщённой регуляризационной сетью (generalized regularization network – GRN). В работе [45] проведена попытка минимизации величин вторых производных.

2.2.2. Классифицирующие деревья

Классифицирующие деревья (decision tree classifiers) относятся к гиперплоскостным классификаторам и требуют сравнительно малых вычислений и объемов памяти для классификации. Эти классификаторы могут быть реализованы с высокой степенью параллельности, формируют регионы принятия решений, используя легко интерпретируемые операции над входными величинами, могут использовать непрерывные и дискретные входные величины. Размер таких классификаторов может быть легко подобран для приведения в соответствие сложности классификатора и имеющегося набора данных для обучения. Эффективные процедуры обучения классифицирующих деревьев сложны и требуют одновременного доступа ко всем образцам для обучения. Эти классификаторы нашли применение в ряде задач [46], [47]. В простейшем случае каждый узел классификатора оперирует с одним неравенством и одной входной величиной. Тогда регионы принятия решений разделяются гиперплоскостями, параллельными осям входного пространства. Такой классификатор эквивалентен прореженной нейронной сети. Более сложные регионы принятия решений, с границами, непараллельными осям входного пространства могут быть аппроксимированы бинарными деревьями с большим числом узлов или более общими деревьями, которые допускают линейные комбинации входных переменных в неравенствах каждого узла [48].

Процедуры обучения для построения бинарных деревьев не минимизируют общую функцию стоимости в явном виде, вместо этого дерево формируется постепенно, с минимизацией локальной функции стоимости на каждой стадии обучения. Локальная функция отражает успех разделения обучающей выборки на классы текущим узлом. Бинарные деревья разработаны для быстрого обучения. Данные для обучения сортируются отдельно вдоль каждой переменной входного пространства и функция стоимости вычисляется для всех возможных разделений данных для обучения. Сравнение бинарного дерева и классификатора с ОРО обнаружило небольшую разницу в вероятности ошибки, но бинарные деревья обучаются значительно быстрее [49].

В [47] обсуждается сходство классификации при помощи многослойных сетей и классифицирующих деревьев. Описывается метод информационно-оптимального синтеза дерева классификации и последующего переноса синтезированного алгоритма классификации в многослойную сеть. Такой подход позволяет получить классификатор на базе нейронной сети с оптимальными параметрами.

2.2.3. Сети высокого порядка

Классификаторы на основе МСП и деревья принятия решений содержат элементы (узлы), которые вычисляют взвешенную сумму входных переменных и затем эту сумму пропускают через нелинейность. Сети высокого порядка наряду с линейными слагаемыми оперируют с

произведениями и степенями входных переменных. На рис. 6 изображен пример сети высокого порядка с двумя входами (x_1 и x_2), которая вычисляет выход y , содержащий произведение входов ($w_3 x_1 x_2$). Однослойный персептрон высокого порядка может быть обучен при помощи алгоритма минимальной среднеквадратичной ошибки (LMS) [11] для формирования полиномиальных разделяющих функций, при этом слагаемые высокого порядка рассматриваются как дополнительные входы. Это может устранить необходимость использования нескольких слоёв и тем самым увеличить скорость обучения, когда априорно известно, что разделяющая функция имеет полиномиальный вид. Однако использование большого числа нелинейных слагаемых ведёт к чрезмерному увеличению числа параметров и ухудшению обобщающей способности сети.

GMDH сети [19], которые также называют сетями адаптивного обучения (adaptive learning networks), представляют собой развитый метод решения проблемы согласования сложности многослойной сети высокого порядка с объёмом имеющейся выборки для обучения. Простая GMDH сеть может быть построена из полиномиальных подсетей, подобных изображённой на рис. 6. Обучение включает в себя послойное построение иерархической сети с использованием подходов подобных тем, что применяются при построении бинарных деревьев. При обучении первого слоя его веса изменяются так, чтобы минимизировать квадратичную ошибку выходов первого слоя, и выходы, которые по существу не вносят вклада, удаляются. Дополнительные слои добавляются до тех пор, пока увеличение сложности оправдывается уменьшением вероятности ошибки, оцениваемой на выборке для обучения или на тестовой выборке. Эти сети предоставляют большинство возможностей классификаторов с ОРО, но, как и деревья принятия решений, используют сложные, но быстрые процедуры обучения, требующие одновременного доступа ко всем образцам для обучения. Экспериментальное сравнение модифицированной GMDH сети и сети с ОРО показало, что для задачи нелинейного отображения случайных временных рядов сеть GMDH обеспечивает меньшую среднеквадратическую ошибку и требует меньшее число внутренних соединений [50].

2.3. Ядерные классификаторы (Kernel Classifiers)

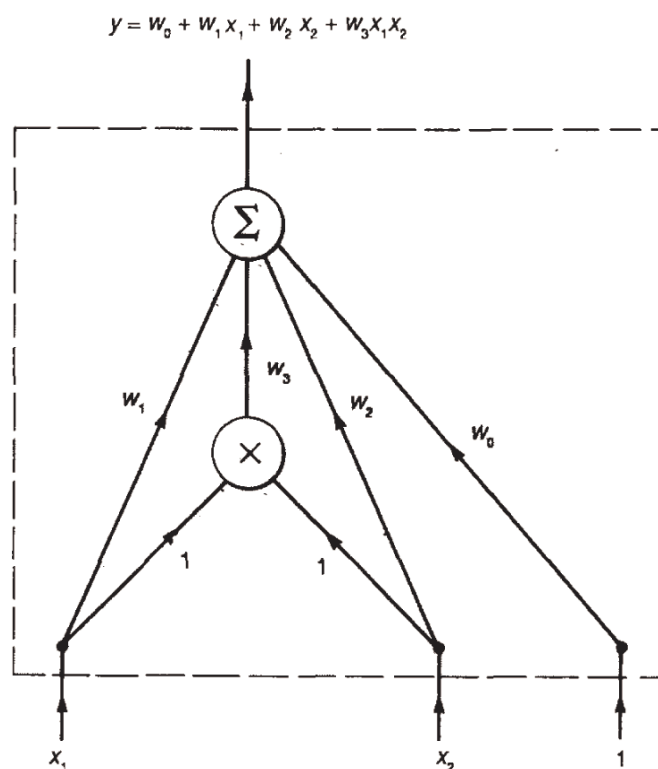


Рис. 6. Сеть высокого порядка

Ядерные классификаторы (классификаторы, основанные на рецептивных областях) формируют сложные области принятия решений из набора перекрывающихся рецептивных областей, ограничиваемых kernфункциями узлов. Узлы используют kernфункции, как показано на рис. 4, которые обеспечивают максимальный отклик, когда вход близок к центру узла. Выход kernфункции узла j определяется как:

$$y_j = f_k(\|\vec{x} - \vec{m}_j\|/h) \quad (30)$$

Здесь f_k – kernфункция, норма – эвклидова, m_j – центр узла j , h – параметр, определяющий ширину kernфункции. Из выражения (30) видно, что отклик kernфункции максимален, когда входной вектор совпадает с центром узла, и монотонно спадает при возрастании расстояния между входом и центром узла. Коэффициент h спада с дистанцией определяет область действия узла. Принятие решения (классификация) производится узлами на верхнем уровне, которые вычисляют функцию от взвешенной суммы выходов kernфункций узлов нижнего уровня.

Ядерные классификаторы обучаются относительно быстро, могут использовать смешанное (с учителем/без учителя) обучение и имеют средние требования по вычислительному времени и памяти. Центры kernфункций могут быть помещены на помеченные образцы из выборки для обучения (на некоторые или все сразу), или определены из кластеризации, или помещены на случайно выбранные непомеченные обучающие образцы. К ядерным классификаторам относятся традиционные классификаторы, которые оценивают распределения вероятностей, используя подход Парзена, и классификаторы, использующие kernфункции [51]. Нейросетевые ядерные классификаторы включают подходы, основанные на картах, использующих массивы узлов с kernфункциями [48]; классификаторы, основанные на Cerebellar Model Articulation Controller (СМАС) [52]; и классификаторы, использующие метод потенциальных функций (классификаторы с радиальными базисными функциями) [53], машины опорных векторов (support vector machine – SVM) [54], [55].

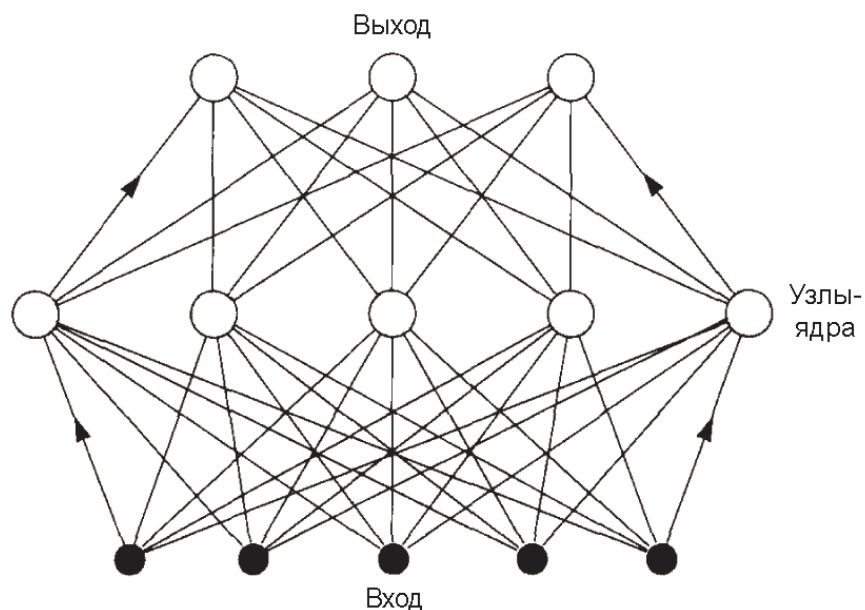


Рис. 7. Классификатор с радиальными базисными функциями

2.3.1. Классификаторы с радиальными базисными функциями

Классификаторы с радиальными базисными функциями (radial basis function, RBF) относятся к ядерным классификаторам. Схема RBF-классификатора представлена на рис. 7. Структура и работа данного классификатора схожи со структурой и работой классификатора на основе карты признаков, за исключением того, что в RBF-классификаторах не производится отбор узлов с максимальными выходами. Вместо этого все узлы с высокими выходными уровнями оказывают влияние на результат классификации. Узлы-ядра вычисляют радиально симметричные функции, которые максимальны, когда входной образ близок к центру узла. Весовые коэффициенты узлов выходного слоя определяются при помощи алгоритма минимальной среднеквадратичной ошибки или средствами матричной алгебры, которые требуют одновременного доступа ко всем образцам для обучения [56]. Кernels функции, как правило, имеют форму Гауссианы. Ширины Гауссовых пиков являются параметрами, которые наряду с числом узлов определяются экспериментально на тестовых данных для обеспечения высокой обобщающей способности сети. Центры kernel функций обычно определяются случайным выбором образцов из набора для обучения. Вероятность ошибки для такого классификатора обычно сходна с вероятностью ошибки классификатора с ОРО. При этом RBF-сети обучаются значительно быстрее МСП, но требуют в несколько раз большего числа весовых коэффициентов.

2.4. Классификаторы по образцу

Классификаторы по образцу осуществляют классификацию, опираясь на сходство входного образа и ближайших соседей из образцов для обучения (экземпляров). Ближайшие соседи могут быть определены с использованием

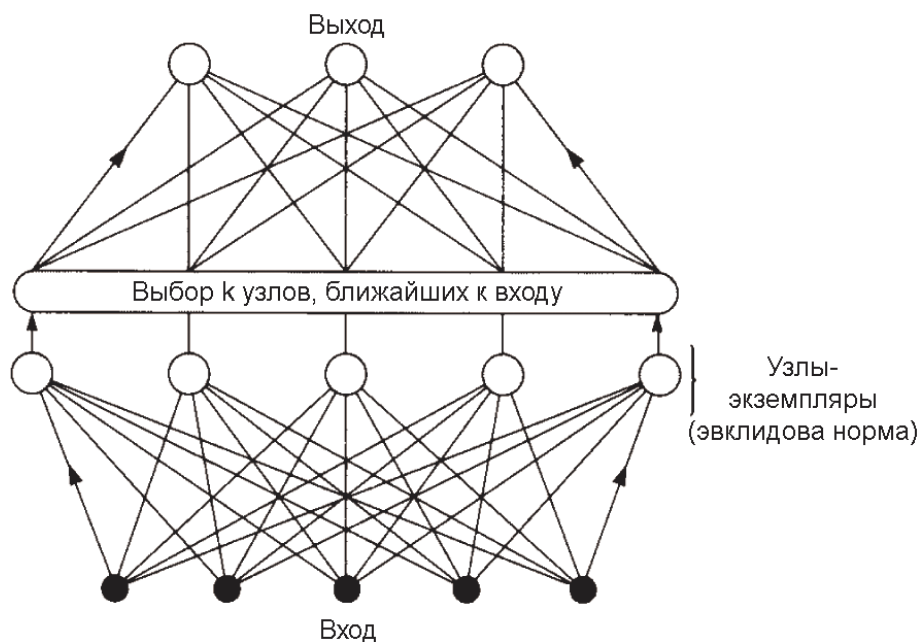


Рис. 8. Классификатор – карта признаков

узлов-экземпляров, сходных с узлами кернфункций. Узлы-экземпляры вычисляют взвешенное эвклидово расстояние между входом и центрами узлов. Центры узлов соответствуют помеченным образцам для обучения или центрам кластеров, сформированных при смешанном обучении. Классификаторы по образцу обучаются быстро, но могут требовать большое количество памяти и вычислений в процессе классификации. Усовершенствованные структуры данных, такие как K-D-деревья (K-D trees) [57], могут уменьшать вычислительные требования на последовательных компьютерах ценой усложнения обучения и адаптации. К классификаторам по образцу относятся «k-ближайших соседей» классификаторы (k-nearest neighbor classifiers) [58], карты признаков (feature-map classifier) [59], обучающиеся векторные квантователи (Learning Vector Quantizer (LVQ)) [13], Restricted Coulomb Energy (RCE) classifiers [60], классификаторы адаптивного резонанса (Adaptive Resonance Theory (ART) classifiers) [12], классификаторы с памятью (“memory-based reasoning”) [61], классификаторы, использующие локально-линейную интерполяцию [62].

2.4.1. Карты признаков

Карта признаков (feature-map classifier) является классификатором, который использует смешанное, без учителя и с учителем, обучение. Схема такого классификатора изображена на рис. 8. Узлы промежуточного слоя данной сети вычисляют эвклидово расстояние между входным образом и центрами узлов-экземпляров, положения которых задаются весовыми коэффициентами. Выходы узлов сравниваются для определения ближайшего к входному образу узла-экземпляра. Соединения с выходными узлами ассоциируют узлы-экземпляры с метками классов. Решение классификации принимается по метке узла-экземпляра, который ближе всего к входному образу. Эта сеть может реализовывать метод «k-ближайших соседей», если

выбирать k узлов-экземпляров, которые ближе всего к входному образу, и в промежуточном слое столько же узлов, сколько образцов для обучения. Такая реализация требует большого объёма памяти, но этот объём может быть уменьшен следующим образом. Сначала нижняя подсеть обучается без учителя, чтобы сформировать векторный квантователь, используя алгоритм k -средних или метод самоорганизующихся карт Кохонена [13]. Затем выходной слой обучается с учителем модифицированным алгоритмом минимальной среднеквадратичной ошибки [59].

2.4.2. Обучающиеся векторные квантователи

Классификатор, основанный на обучающемся векторном квантователе (Learning Vector Quantizer, LVQ) использует такую же нейросетевую структуру, как классификатор, основанный на карте признаков (см. рис. 8). Этот классификатор требует дополнительного обучения с учителем после обучения карты признаков. На дополнительной стадии обучения при появлении ошибок корректируются веса узлов-экземпляров, центры узлов-экземпляров смещаются так, чтобы увеличить вероятность правильной классификации. При этом немного смещаются границы принятия решений при постоянном числе узлов-экземпляров. Такие классификаторы как правило меньше вероятности ошибки, чем классификаторы, основанные на карте признаков, особенно при малом количестве узлов-экземпляров.

2.4.3. Гиперсферные классификаторы

Гиперсферные классификаторы (Hypersphere Classifiers) относятся к классификаторам по образцу. Эти классификаторы создают регионы принятия решений из гиперсфер разного размера с центрами в узлах-экземплярах. На выходе узла-экземпляра формируется высокий уровень, только если текущий входной образ находится от центра узла на расстоянии, не превышающем радиус сферы данного узла, иначе выходной уровень низкий. Решение принимается по метке, относящейся к большинству узлов с высокими выходными уровнями. Если входной образ не попал ни в одну гиперсферу, то считается, что он не относится ни к одному из классов. Одним из вариантов гиперсферных классификаторов является RCE-классификатор (Restricted Coulomb Energy classifier) [60]. Этот классификатор сходен с методом « k -ближайших соседей», в том числе и по скорости обучения, но требует значительно меньше узлов-экземпляров.

2.5. Выбор классификатора

В ряде случаев классификаторы из четырёх названных групп обеспечивают одинаково низкий уровень ошибочной классификации, но при этом существенно различаются по другим характеристикам. На рис. 9 показаны требования различных классификаторов по вычислительному времени и памяти при реализации на последовательных компьютерах. Положения точек относительные и могут меняться в зависимости от решаемой задачи и модификаций алгоритмов. Гиперплоскостные

классификаторы, такие как многослойный персептрон, требуют длительного обучения, но не требуют большого объема памяти и быстро проводят классификацию.

Классификаторы, использующие методы ближайших соседей, крайне быстро обучаются, но требуют больших объемов памяти и вычислительных ресурсов. Ядерные классификаторы и множество классификаторов по образцу обладают промежуточными

характеристиками. Наибольшее распространение получили классификаторы на основе многослойных сетей, но другие методы классификации тоже находят применение благодаря возможностям использования непомеченных данных для обучения, высокой степени параллельности алгоритмов, простых правил адаптации.

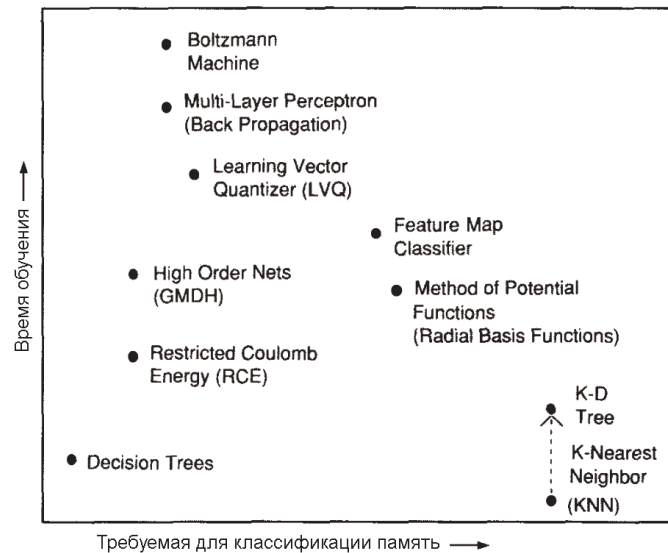


Рис. 9. Сравнение классификаторов [49]

3. Приложения ИНС в задачах обработки сигналов

Классические методы обработки сигналов основываются на фиксированной теоретической модели известного процесса с неизвестными параметрами. Настройка метода под конкретную задачу состоит в определении параметров модели для сигналов, искаженных шумами с известными статистическими характеристиками [63]. Для обработки сигналов с шумами с неизвестной статистикой могут быть использованы нейронные сети. ИНС обладают рядом свойств, которые обеспечивают преимущества при обработке сигналов.

Во-первых, ИНС предоставляет возможность подстройки свободных параметров под изменения статистических свойств среды. При этом для успешного применения свойства адаптивности ИНС необходимо найти баланс между пластичностью и стабильностью. Это означает, что постоянная времени системы должна быть достаточно большой, чтобы игнорировать паразитные возмущения, и достаточно малой, чтобы реагировать на существенные изменения в среде. Традиционные адаптивные фильтры также обладают возможностью автоматического изменения параметров в соответствии со статистическими вариациями в среде [65], но их возможности адаптивной обработки сигналов ограничены их структурой. В работе [66] описан метод применения многослойного персептрона для построения адаптивной системы с долговременной и кратковременной памятью. Такая архитектура позволяет определять модель процесса не априорно, а по реализациям. Построенная модель благодаря наличию кратковременной памяти подстраивается под изменения в среде.

Во-вторых, ИНС предоставляет возможность непараметрического подхода к нелинейному анализу данных. Сети прямого распространения (МСП, РБФ) подвергаются обучению, в процессе которого их параметры (весовые коэффициенты и пороговые уровни) настраиваются так, чтобы минимизировать функцию стоимости (ошибки). При обучении сеть по примерам строит отображение «вход-выход» для решаемой задачи непараметрическим образом. Термин «непараметрический» используется здесь в статистическом смысле, это означает, что знание лежащих в основе распределений вероятностей не требуется.

Традиционный подход математической статистики предполагает использование математически хорошо изученных моделей, предполагающих идеализированные условия линейности, стационарности в широком смысле, гауссовости. При этом параметры модели (например, среднее, стандартное отклонение) предполагают асимптотическую оценку при стремлении числа примеров к бесконечности, но определяются по небольшому набору обучающих примеров. Методы, основанные на нейронных сетях, привлекательны для практических приложений благодаря возможности ИНС работать с нелинейностями, нестационарностями и в отсутствие предположения о гауссовости.

Во многих случаях нейросетевые методы работают лучше, чем сравнимые статистические методы. Превосходство нейронных сетей объясняется различиями в соответствующих процедурах оптимизации. В статистических методах, как правило, оптимизация параметров базисных функций ведётся последовательно, ошибки при принятии решений на ранних стадиях оптимизации не всегда могут быть впоследствии исправлены. В нейронных сетях, напротив, весь набор базисных функций, представляемый выходами нейронов скрытых слоёв, оптимизируется одновременно итеративным образом. Такая процедура более устойчива.

В-третьих, нейронные сети, обучаемые с учителем, являются универсальными аппроксиматорами непрерывных отображений «вход-выход». Это позволяет использовать сети для определения отношения правдоподобия в задачах детектирования и классификации. В таких случаях ИНС прямого распространения с временными задержками обучается на нескольких реализациях принятого сигнала. В [67] показано, что такие сети являются универсальными аппроксиматорами нелинейных динамических систем, но их применение ограничено стационарными процессами. Применение ИНС для нестационарных процессов требует учёта временной природы сигналов в структуре сети, специального обучения рекуррентной ИНС, или применения методов предобработки, учитывающих нестационарность.

Благодаря перечисленным свойствам нейронные сети могут использоваться в условиях частичной определённости и находят применение в различных задачах обработки сигналов.

3.1. Фильтрация

В отличие от традиционного аналитического подхода к получению выражений для фильтра, нейросетевые фильтры синтезируются по реализациям процесса, которые могут быть смоделированы или получены из эксперимента. При этом не требуется традиционных предположений о линейности, нормальном распределении, аддитивности шума, марковости и т.п. Нейросетевой фильтр с подходящей архитектурой, обученный должным образом, извлекает наиболее информативные статистики и аппроксимирует оптимальный метод для любой заданной точности. Благодаря параллельной архитектуре нейросетевой фильтр может быть реализован для работы в реальном времени.

Нейросетевые фильтры применяются для компенсации различных нелинейных искажений. МСП применяется для адаптивной коррекции сигнала прошедшего через канал с дисперсией и аддитивным шумом, для фильтрации сигналов с кодовым разделением доступа при наличии импульсного шума в негауссовом канале. Возможно применение ИНС для слепого разделения сигналов.

3.2. Фазовая автоподстройка частоты

Нейросетевые методы самоорганизующихся карт применяются для систем фазовой автоподстройки. В [68] описан метод определения несущей частоты для систем связи с гауссовской минимальной манипуляцией (GMSK). Авторы используют обучающийся векторный квантователь (LVQ) для восстановления передаваемой фазы. По разности восстановленной и ожидаемой фазы определяется сдвиг несущей частоты. Рассмотрены случаи одно- и многолучевого распространения. В [69] исследуется схема на основе предсказывающей самоорганизующиеся карты (predictive self-organizing map – P-SOM) для слежения за фазой зашумлённого сигнала в случае наличия эффекта Доплера.

3.3. Определение типа модуляции

Ряд работ посвящён разработке нейросетевых методов определения типа манипуляции в принятом сигнале. Для решения этой задачи часто используются РБФ-сети. Сеть как правило используется для выбора одного из набора предусмотренных видов модуляции.

3.4. Детектирование

Для решения задач детектирования применяются различные архитектуры нейронных сетей. Наиболее востребованы в задачах детектирования сети прямого распространения (МСП и РБФ-сети), это связано с их возможностью аппроксимирования произвольных отображений. Кроме работы непосредственно с принятым сигналом или вычисленными характеристиками, в ряде работ рассматривается следующий подход к детектированию сигналов, прошедших через нестационарную среду. Сначала производится смена представления: вычисляется частотно-временное

изображение принятого сигнала. Такое представление позволяет учесть нестационарность сигнала в явном виде и перейти от задачи детектирования к задаче адаптивного распознавания образов. Переход к распознаванию образов позволяет применить для детектирования уже разработанный аппарат обучаемых ИНС [70]. Этот подход отличается от классических методов заменой компактного, часто слишком упрощённого, описания принятого сигнала, сильно избыточным описанием.

Кроме детектирования передаваемых символов ИНС также применяется для детектирования сигналов, то есть обнаружения (расознавания). Применяются ИНС и в задачах детектирования радиолокационных сигналов.

3.5. Расознавание речи

Расознавание речи является многоуровневой задачей распознавания образов. На различных уровнях акустические сигналы разбиваются на фонемы (или группы фонем), слова, словосочетания и предложения. В естественной речи каждый уровень вносит дополнительные ограничения, связанные с сочетаемостью единиц речи, например произношения (транскрипции) слов, характерный для языка порядок слов. Ограничения на более высоких уровнях используются для исправления ошибок и разрешения неопределённостей на более низких уровнях. Для использования всей иерархии ограничений на низких уровнях решениям придают вероятностный характер, а дискретные решения принимаются только на верхнем уровне.

Применение ИНС на различных уровнях системы распознавания позволяет по полученной последовательности производить рекурсивную сегментацию для уточнения модели при обучении [71].

Поскольку ИНС обладают свойствами обобщения и устойчивости к шуму, то в системах распознавания речи наиболее часто они используются для акустического моделирования. Кроме акустического моделирования ИНС применяются в задачах распознавания речи для обработки гипотез на верхних уровнях системы распознавания.

4. Моделирование

Аппаратная реализация нейросетевых алгоритмов (на базе нейрочипов, ПЛИС) позволяет полностью использовать преимущества параллельной структуры ИНС. Но наибольшее распространение получили компьютерные модели нейронных сетей. Это прежде всего связано с возможностями построения моделей всех существующих сетей и внесения разнообразных модификаций в алгоритмы обучения и функционирования. Кроме того темп роста быстродействия универсальных вычислительных машин (компьютеров) существенно превосходит темпы развития специализированных микросхем (нейрочипов).

Для моделирования нейросетевых алгоритмов разработано большое число программ. Библиотеки нейросетевых алгоритмов существуют для большинства распространённых языков программирования, в том числе для С, С++ и Matlab. Наибольший интерес представляют библиотеки на языке

C++ в связи с возможностью вычислительно эффективной структурированной реализации алгоритмов. Одной из развитых библиотек с открытым кодом является библиотека «Fast Artificial Neural Network Library». Для освоения основ искусственных нейронных сетей и реализации сложных алгоритмов обучения (с нестандартной функцией ошибки, регуляризацией, переменным обучающим множеством) можно написать библиотеку следующих классов: нейрон, слой, сеть. Приведённый набор классов позволяет описывать сети со слоистой структурой: персептроны, в том числе многослойные, рекуррентные сети, сети Кохонена и некоторые другие. При необходимости набор может быть дополнен.

Рассмотрим функционирование многослойного персептрона. Каждый нейрон (экземпляр соответствующего класса) хранит набор весовых коэффициентов и вычисленное нейроном значение. Здесь целесообразно напомнить, что для удобства вычислений пороговый уровень нейрона можно считать весовым коэффициентом дополнительного входа с постоянным уровнем. Слой содержит массив нейронов. Сеть содержит массив слоёв. При прямом проходе (вычислении отклика персептрона) нейроны послойно перемножают входные значения с весовыми коэффициентами, вычисляют и сохраняют значение функции активации. При обратном проходе (обучении методом обратного распространения ошибки или его модификациями) в нейронах послойно вычисляются и сохраняются дельта-множители. После вычисления дельта-множителей во всей сети производится коррекция весовых коэффициентов. Экспериментально обнаружено, что вычисление и применение коррекций весов целесообразно делать отдельно для каждого образца из обучающей выборки. Пакетное обучение (с вычислением полного градиента по всей выборке) как правило происходит медленнее. Рекомендуется нормировать входные и выходные значения сети на единицу, тогда начальные значения весовых коэффициентов могут быть выбраны случайным образом из диапазона шириной порядка одной десятой вокруг нуля. Шаг градиентного метода в выражении (13) выбирается из диапазона (0;1).

Заключение

Кратко изложены основы ИНС: описаны узлы (искусственные нейроны) и ряд типичных структур сетей. Приведено описание алгоритмов обучения ИНС.

Приведена систематизация нейросетевых классификаторов, даны рекомендации по выбору классификатора соответственно решаемой задаче.

Обсуждены преимущества и недостатки, а также условия применения существующих методов обработки сигналов при помощи ИНС. Проведён обзор приложений нейросетевых методов решения актуальных задач обработки радиосигналов. Рассмотрены задачи фильтрации, фазовой автоподстройке частоты, определения типа модуляции и детектирования. Кратко описано применение ИНС в системе распознавания речи.

Рассмотрено компьютерное моделирование работы ИНС. Приведена структура программы для реализации простых нейронных сетей. Даны рекомендации относительно выбора параметров сетей и алгоритмов.

Литература

1. Wiener N., "Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications" Cambridge, MA: MIT Press, 1949, 163 p.
2. McCulloch W., Pitts W., "A logical calculus of ideas imminent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.
3. Hebb D.O., "The Organization of Behavior: A neuropsychological theory." New York: Wiley, 1949.
4. Розенблат Ф. «Принципы нейродинамики: Перцептрон и теория механизмов мозга», пер. с англ. – М.: Мир, 1965, 175 с.
5. Hopfield J.J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, April 1982, pp. 2554-2558.
6. Колмогоров А.Н., «О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения» Докл. АН СССР, т. 114, н. 5, 1957, с. 953-956.
7. Rumelhart D.E., Hinton G.E., and Williams R.J., "Learning internal representations by error propagation," In *Parallel distributed processing*, vol. 1, pp. 318–362, Cambridge, MA: MIT Press, 1986.
8. Галушкин А.И., «Синтез многослойных систем распознавания образов». М.: Энергия, 1974, 368 с.
9. Горбань А.Н., Дунин-Барковский В.Л., Миркес Е.М. и др., «Нейроинформатика». Новосибирск: Наука. Сиб. предприятие РАН, 1998, 296 с.
10. Shannon C.E., "A mathematical theory of communications," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623-656, 1948.
11. Lippmann R. P., "An Introduction to Computing with Neural Nets," *IEEE ASSP Mag.*, vol. 4 (2). pp. 4-22, Apr. 1987.
12. Головкин В.А., «Нейронные сети: обучение, организация и применение.» – М.: ИПРЖР, 2001, 256 с.
13. Kohonen T., "Self-organizing maps," 3. ed. – Berlin; Heidelberg; New York; Barcelona; Hong Kong; London; Milan; Paris; Singapore; Tokyo: Springer, 2001, 501 p.
14. Kung S.-Y., Taur J., Lin S.-H., "Synergistic Modeling and Applications of Hierarchical Fuzzy Neural Networks," *Proceedings of the IEEE*, vol. 87, no. 9, September 1999, pp. 1551-1574.
15. Моррис У., «Наука об управлении. Байесовский подход.» М.: Мир, 1971.
16. Ackley D.H., Hinton G.E., Sejnowski T.J., "A Learning Algorithm for Boltzmann Machines," *Cognitive Science*, vol. 9, pp. 147-160, 1985.
17. Breiman L., Friedman J.H., Olshen R.A., Stone C.J., "Classification and Regression Trees," Wadsworth International Group, Belmont, CA, 1984.

18. Giles C.L., Maxwell T., "Learning, Invariance, and Generalization in High-Order Networks," *Applied Optics*, vol. 26, pp. 4972-4978, Dec. 1987.
19. Farlow S., "Self-Organizing Methods in Modeling," Marcel Dekker, 1984.
20. Hampshire J., Pearlmutter B., "Equivalence Proofs for Multi-Layer Perceptron Classifiers and the Bayesian Discriminant Function," *Proc. of the 1990 Connectionist Models Summer School*, San Mateo, CA: Morgan Kaufmann Publishers, 1990, p. 1.
21. Нейрокомпьютеры в системах обработки сигналов. / Под ред. Гуляева Ю.В. и Галушкина А.И.. М: Радиотехника, 2003.
22. Cybenko G., "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals and Systems*, v.2, 1989, pp. 303-314.
23. Pierce S.G., Ben-Haim Y., Worden K., Manson G., "Evaluation of Neural Network Robust Reliability Using Information-Gap Theory," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, November 2006, pp. 1349-1361.
24. Teoh E.J., Tan K.C., Xiang C., "Estimating the Number of Hidden Neurons in a Feedforward Network Using the Singular Value Decomposition," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, September 2006, pp. 1623-1629.
25. Fukumizu K., "Statistical Active Learning in Multilayer Perceptrons," *IEEE Transactions on Neural Networks*, vol. 11, no. 1, January 2000, pp. 17-26.
26. Царегородцев В.Г., «Извлечение знаний из обучаемых нейронных сетей». Нейроинформатика и ее приложения: Тез. докл. 6-го Всероссийского семинара. Красноярск, 1999, с. 150-151.
27. Lu H.; Setiono R., Liu H., "Effective data mining using neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no 6, December 1996, pp. 957-961.
28. Jensen C.A., Reed R.D., Marks R.J., II, El-Sharkawi M.A., Jung J.-B., Miyamoto R.T., Anderson G. M., Eggen C. J., "Inversion of Feedforward Neural Networks: Algorithms and Applications," *Proceedings of the IEEE*, vol. 87, no. 9, September 1999, 1536-1549.
29. Yao X., "Evolving Artificial Neural Networks," *Proceedings of the IEEE*, vol. 87, no. 9, September 1999, 1423-1447.
30. Хомич А.В., Жуков Л.А., «Метод эволюционной оптимизации и его приложение к задаче синтеза искусственных нейронных сетей», *Нейрокомпьютеры: разработка, применение – М.:Радиотехника*, № 12, 2004 г, с. 3-15.
31. Man Z., Wu H. R., Liu S., Yu X., "A New Adaptive Backpropagation Algorithm Based on Lyapunov Stability Theory for Neural Networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, November 2006, 1580-1591.
32. Riedmiller M., Braun H., "A direct adaptive method for faster backpropagation learning: The rprop algorithm," *Proc. of IEEE International Conference on Neural Networks*, 1993.

33. Осовский С., «Нейронные сети для обработки информации». Пер. с польского И.Д.Рудинского. – М.: Финансы и статистика, 2002, 344 с.
34. Yam J.Y.F., Chow T.W.S., “Feedforward Networks Training Speed Enhancement by Optimal Initialization of the Synaptic Coefficients,” IEEE Transactions on Neural Networks, vol. 12, no. 2, March 2001, pp. 430-434.
35. Галушкин А.И., «Формирование начальных условий для ускорения настройки коэффициентов нейронных сетей в задачах оптимизации». Научная сессия МИФИ – 2006. VIII Всероссийская научно-техническая конференция «Нейроинформатика-2006» Сборник научных трудов. В 3-х частях. Ч.2. М.: МИФИ, 2006. с. 87-94.
36. Liang N.-Y., Huang G.-B., Saratchandran P., Sundararajan N., “A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks,” IEEE Transactions on Neural Networks, vol. 17, no. 6, November 2006, pp. 1411-1423.
37. Luderer T.B., Yamazaki A., Zanchettin C., “An Optimization Methodology for Neural Network Weights and Architectures,” IEEE Transactions on Neural Networks, vol. 17, no. 6, November 2006, pp. 1452-1459.
38. Behera L., Kumar S., Patnaik A., “On Adaptive Learning Rate That Guarantees Convergence in Feedforward Networks,” IEEE Transactions on Neural Networks, vol. 17, no. 5, September 2006, pp. 1116-1125.
39. Zhang N., Wu W., Zheng G., “Convergence of Gradient Method with Momentum for Two-Layer Feedforward Neural Networks,” IEEE Transactions on Neural Networks, vol. 17, no. 2, September 2006, pp. 522-525.
40. Joost M., Schiffmann W., “Speeding up backpropagation algorithms by using cross-entropy combined with pattern normalization,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS), vol. 6, no. 2, pp. 117-126, 1998.
41. Drucker H., LeCun Y., “Improving generalization performance using double backpropagation,” IEEE Transactions on Neural Networks, 1992, vol. 3, pp. 991-997.
42. Bishop C., “Training with noise is equivalent to Tikhonov regularization,” Neural Comput., vol. 7, no. 1, pp. 108-116, 1995.
43. Aires F., Schmitt M., Chedin A., Scott N., “The “Weight Smoothing” Regularization of MLP for Jacobian Stabilization,” IEEE Transactions on Neural Networks, vol. 10, no. 6, November 1999, pp. 1502-1510.
44. Girossi F., Jones M., Poggio T., “Regularization theory and neural networks architectures,” Neural Comput., vol. 7, pp. 219-269, 1995.
45. Bishop C., “Curvature-driven smoothing: A learning algorithm for feedforward networks,” IEEE Transactions on Neural Networks, 1993, vol. 4, pp. 882-884.
46. Safavian S.R., Landgrebe D., “A survey of decision tree classifier methodology,” IEEE Trans. Syst., Man, Cybern., vol. 21, no. 3, pp. 660–674, 1991.

47. Sethi I. K., "Entropy Nets: From Decision Trees to Neural Networks," Proceedings of the IEEE, vol. 78, no. 10, October 1990, 1605-1613.
48. Rojer A., Schwartz E., "A Multiple-Map Model for Pattern Classification," Neural Computation, vol. 1(1), pp. 104-115, 1989.
49. Lippmann R.P., "Pattern Classification Using Neural Networks," IEEE Communications Magazine, November 1989, pp. 47-50, 59-64.
50. Tenorio M.F., Lee W.T., "Self Organizing Neural Networks for the Identification Problem," Advances in Neural Information Processing Systems I, D. S. Touretzky, ed., pp. 57-64, San Mateo, CA: Morgan Kauffman, 1989.
51. Hand D.J., ed., "Kernel Discriminant Analysis," John Wiley and Sons Ltd., New York, NY, 1982.
52. Miller W.T. III, Glanz F.H., Kraft L.G. III, "CMAC: An Associative Neural Network Alternative to Backpropagation," Proceedings of the IEEE, vol. 78, no. 10, October 1990 pp. 1561-1567.
53. Тархов Д.А., «Нейронные сети. Модели и алгоритмы.» Кн. 18 – М.: Радиотехника, 2005. – 256 с.
54. Müller K.-R., Mika S., Rätsch G., Tsuda K., Schölkopf B., "An Introduction to Kernel-Based Learning Algorithms," IEEE Transactions on Neural Networks, vol. 12, no. 2, March 2001, pp. 181-201.
55. Хайкин С., «Нейронные сети: полный курс», 2-е издание. Пер. с англ. – М.: Издательский дом «Вильямс», 2006, 1104 с.
56. Haykin S., "Neural Networks: A Comprehensive Foundation". New York: MacMillan, 1994.
57. Omohundro S.M., "Efficient Algorithms with Neural Network Behavior," Complex Systems, vol. 1, pp. 273-347, 1987.
58. Holmstrom L., Koistinen P., Laaksonen J., Oja E., "Neural and statistical classifiers-taxonomy and two case studies," IEEE Transactions on Neural Networks, vol. 8, no. 1, January 1997, pp 5-17.
59. Huang W.M., Lippmann R.P., "Neural Net and Traditional Classifiers," Neural Info. Processing Syst., Anderson D., ed., pp. 387-396, NY: American Institute of Physics, 1988.
60. Olmez T., "Classification of EEG waveforms by using RCE neural networks and genetic algorithms," Electron. Lett., vol. 33, no. 18, pp. 1561-1562, 1997.
61. Stanfill C, Waltz D., "Toward Memory-Based Reasoning," Commun. of the ACM, vol. 29(12), pp. 1213-1228, Dec. 1986.
62. Wolpert D., "Alternative Generalizers to Neural Nets," Neural Networks, vol. 1, supp. 1, p. 474, Abstracts of 1st Annual INNS Meeting, Boston, 1988.
63. Garth L.M., Poor H.V., "Detection of Non-Gaussian Signals: A Paradigm for Modern Statistical Signal Processing," Proceedings of the IEEE, vol. 82, no. 7, July 1994, pp. 1061-1095.
64. Окунев Ю.Б., «Цифровая передача информации фазомодулированными сигналами», М.: Радио и связь, 1991, 296 с.

65. Haykin S., "Adaptive Filter Theory," Third Edition, Prentice-Hall, 1996, 989 p.
66. Lo J.T., Bassu D., "Adaptive Multilayer Perceptrons With Long- and Short-Term Memories," IEEE Transactions On Neural Networks, vol. 13, no. 1, January 2002.
67. Sandberg I.W., Xu L., "Uniform approximation of multidimensional myopic maps," IEEE Transactions on Circuits Systems, vol. 44, no.6, June 1997, pp. 477-485.
68. Aiello A., Grimaldi D., "Frequency Error Measurement in GMSK Signals in a Multipath Propagation Environment", IEEE Transactions on Instrumentation and Measurement, vol. 52, no. 3, June 2003, pp. 938-945.
69. Hirose A., Nagashima T., "Predictive Self-Organizing Map for Vector Quantization of Migratory Signals and Its Application to Mobile Communications," IEEE Transactions on Neural Networks, vol. 14, no. 6, November 2003, pp. 1532-1539.
70. LeCun Y., Bengio Y., "Convolution networks for images, speech, and time series," in The Handbook of Brain Theory and Neural Networks, M. A. Arbib, Ed. Cambridge, MA: MIT, 1995.
71. Tebelskis J., "Speech Recognition using Neural Networks," School of Computer Science Carnegie Mellon University Pittsburgh, Pennsylvania, 1995, 180 p.
72. Becchetti C., Ricotti L.R., Speech Recognition Theory and C++ Implementation. John Wiley & Sons Ltd, 1999.