Міністерство освіти і науки України
Донбаська державна машинобудівна академія

Кафедра «Технології машинобудування»

КОНСПЕКТ ЛЕКЦІЙ

по курсу «Основи сучасних теорій моделювання процесів»
для студентів спеціальності 131 - Прикладна механіка
очної та заочної форми навчання

Краматорськ 2018

Міністерство освіти і науки України
Донбаська державна машинобудівна академія
Кафедра «Технології машинобудування»

Укладач: д. т. н., проф. Ковалевський С. В.

КОНСПЕКТ ЛЕКЦІЙ

по курсу «Основи сучасних теорій моделювання процесів»

Затверджено на засіданні кафедри
«Технології машинобудування».
Протокол № 19 від 20 лютого 2018 р

Краматорськ 2018

УДК 621.05(075): 681.51

Конспект лекцій з курсу «Основи сучасних теорій моделювання процесів» (для студентів спеціальності 131 - Прикладна механіка, спе-соціалізації Технологія машинобудування очної та заочної форми навчання). У конспекті розглядаються загальні принципи побудови нейронних мереж, класифікація та алгоритми їх навчання.

Укладач: д. т. н., проф. Ковалевський С. В.

Відп. за випуск: зав. каф. ТМ, д. т. н., проф. Ковалевський С. В.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Общими свойствами интеллектуальной деятельности как человека, так и машины являются сбор и организация информации. При этом человеческое мышление накапливает приобретенные за весь свой жизненный опыт знания таким образом, чтобы затем использовать их в непредсказуемых ситуациях в будущем. Разработка информационных компьютерных систем с подобными качествами стала важным шагом в развитии интеллектуальных технологий. Математической моделью, в основе которой лежат современные представления о строении мозга человека, происходящих в нем процессах обработки информации, являются *искусственные нейронные сети* (ИНС).

Различия между процессами обработки информации человеком и ЭВМ состоят в скорости вычисления и подходе к решению задач. Мозг человека иногда представляют в виде огромного количества параллельно работающих «процессоров» (*нейронов*) с многочисленными и разнообразными связями между ними. Каждый из «процессоров» вполне автономен и влияет лишь на непосредственно связанные с ним. Параллельной обработке информации мозгом соответствует последовательное выполнение задачи в ЭВМ по определенному алгоритму.

Выполнение любых вычислений практически невозможно без памяти. Память человека носит *ассоциативный распределенный* характер. Преимущества такой памяти можно показать на следующем примере: услышав совершенно незнакомое слово, вы немедленно даёте ответ, что вы его не знаете. Компьютер для поиска ответа на подобный запрос потратит много времени. Человеческий мозг надежен, способен к накоплению новой информации без потери старой и к ее обобщению. Даже при разрушении некоторой части мозга, функции его сохраняются.

За последнее время сложилось представление о необходимости существования систем переработки информации, которые не используют принципы сотворенных человеком артефактов. При этом они должны характеризоваться высокой поведенческой сложностью, параллелизмом обработки информации, динамическими механизмами переработки, основанными на сложных нелинейных процессах, высокой эффективностью преобразования информации, значительной информационной сложностью исходных операций, способностью к изменчивости и эволюционному самообучению. Экспертные системы как вид информационных систем тоже могут использовать математическую модель ИНС. Их разработка направлена на имитацию мышления человека-эксперта, поэтому алгоритмы обработки информации с помощью определенных правил было бы удобно заменить параллельным механизмом обработки данных. Человеческий мозг достигает высокой скорости сложных процессов обработки благодаря параллельной организации нервной системы. При этом большое число нейронов, работающих со сравнительно низкой скоростью, одновременно выполняет относительно простые операции над сигналами, поступающими от

других нейронов. Этот же принцип параллельной работы положен в основу интенсивно разрабатываемых в течение последних лет нейросетей. Ряд алгоритмов может быть использован для решения экспертных задач, задач диагностики и прогнозирования.

Теория ИНС вытекает из множества дисциплин, включая физиологию, математику, нейробионику, физику, технику, философию, биологию и лингвистику, то есть ИНС-технология – результат работы многих наук по одному направлению – созданию интеллектуальных систем, имеющих практическое приложение с показателями критических технологий.

1 ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ НЕЙРОННЫХ СЕТЕЙ

Для описания алгоритмов и устройств в нейроинформатике выработана специальная «схемотехника», в которой элементарные устройства – *сумматоры*, *синапсы*, *нейроны* и т. п. объединяются в сети, предназначенные для решения задач.

Самый заслуженный и, вероятно, наиболее важный элемент нейросистем – *адаптивный сумматор*. Адаптивный сумматор вычисляет скалярное произведение вектора входного сигнала x на вектор параметров α. На схемах будем обозначать его так, как показано на рисунке 1.1.

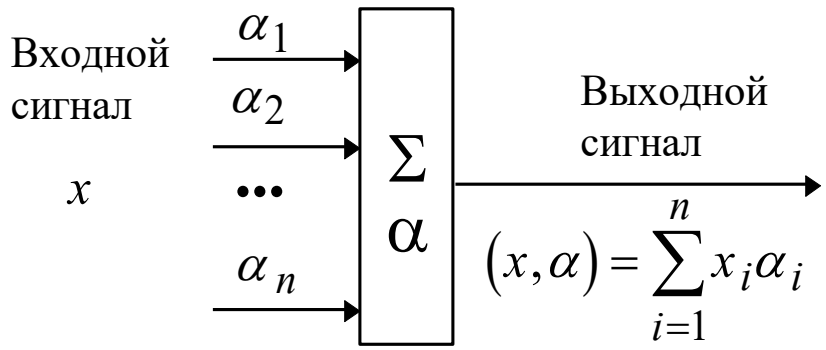


Рисунок 1.1 – Адаптивный сумматор

Адаптивным его называют из-за наличия вектора настраиваемых параметров α.. Для многих задач полезно иметь линейную неоднородную функцию выходных сигналов. Ее вычисление также можно представить с помощью адаптивного сумматора, имеющего n+1 вход и получающего на 0-й вход постоянный единичный сигнал (рисунок 1.2).

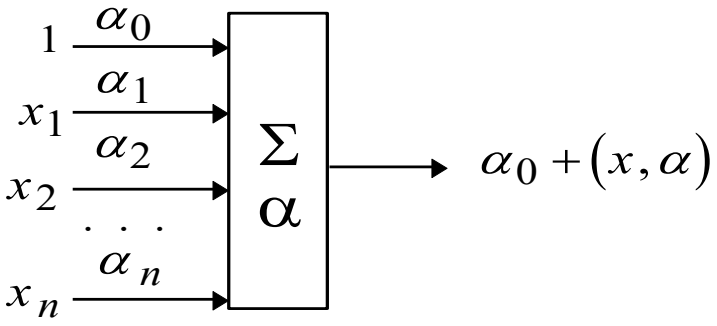


Рисунок 1.2 – Неоднородный адаптивный сумматор

Нелинейный преобразователь сигнала изображен на рисунке 1.3. Он получает скалярный входной сигнал *x* и переводит его в *φ(x)*.

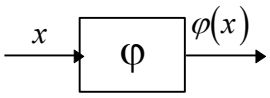


Рисунок 1.3 – Нелинейный преобразователь сигнала.

Точка ветвления служит для рассылки одного сигнала по нескольким адресам (рисунок 1.4). Она получает скалярный входной сигнал *x* и передает его всем своим выходам.

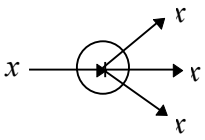


Рисунок 1.4 – Точка ветвления

Стандартный *формальный нейрон* составлен из входного сумматора, нелинейного преобразователя и точки ветвления на выходе (рисунок 1.5).

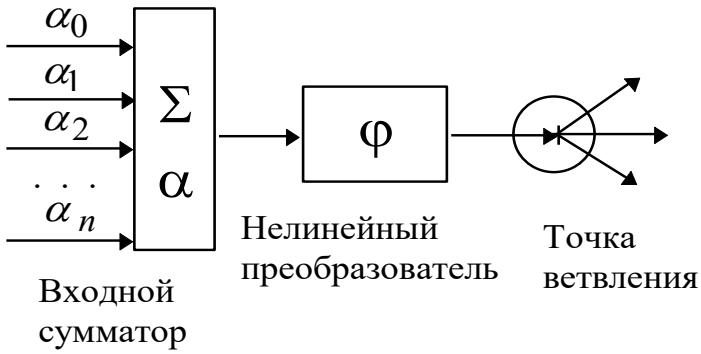


Рисунок 1.5 – Формальный нейрон

Линейная связь – *синапс* – отдельно от сумматоров не встречается, однако для некоторых рассуждений бывает удобно выделить этот элемент (рисунок 1.6). Он умножает входной сигнал *x* на «вес синапса» *a*.

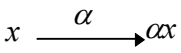


Рисунок 1.6 – Линейная связь (синапс)

Веса синапсов сети образуют набор адаптивных параметров, настраивая которые, нейронная сеть обучается решению задачи. Обычно на диапазон изменения весов синапсов накладываются некоторые ограничения, например, принадлежности веса синапса диапазону [-1,1].

Наиболее часто используются следующие функции активации (таблица 1.1)

а) линейная $f(x) = x$, эквивалентная отсутствию порогового элемента;

б) кусочно-линейная $f(x) = \gamma,\ x \geq \gamma$

$x,\ |x| < \gamma;$

$-\gamma,\ x \leq -\gamma$

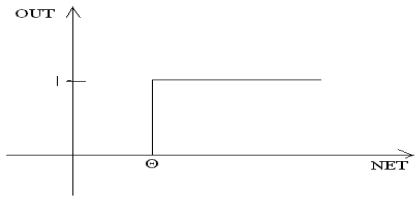
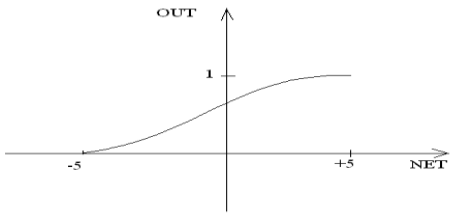
в) ступенчатая пороговая $f(x) = \gamma,\ x > 0;$

$-\gamma,\ x \leq 0$

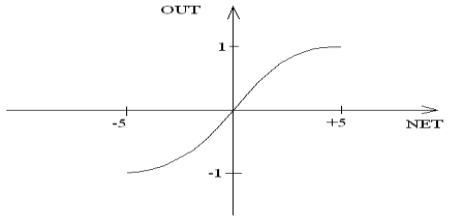
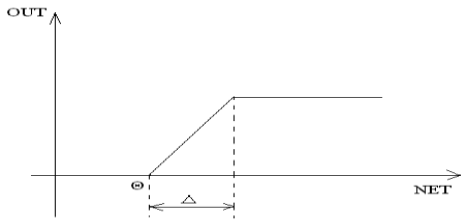
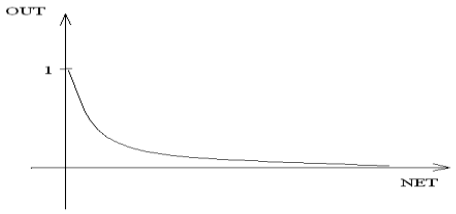
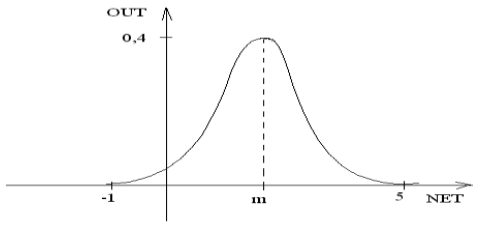
г) сигмоидная $S(x) = 1/(1+ 1/exp(x))$

д) гиперболический тангенс $S(x) = tanh(x)$.

Таблица 1.1 – Виды активационных (пороговых) функций

Наименование активационной функции	**Вид активационной функции**
жёсткая ступенька	
сигмоида (функция Ферми, логистическая функция)	

Продолжение таблицы 1.1

Наименование активационной функции	**Вид активационной функции**
гиперболический тангенс	
пологая ступенька	
экспонента	
кривая Гаусса	

Одной из первых попыток созданий ИНС был *персептрон*, предложенный Ф. Розенблаттом. Существует несколько топологий, относящихся к персептронам: *удвоенный персептрон*, *перекрестно-удвоенный*, *обратно удвоенный*. На рисунке 1.7 изображен элементарный персептрон (он и будет описан).

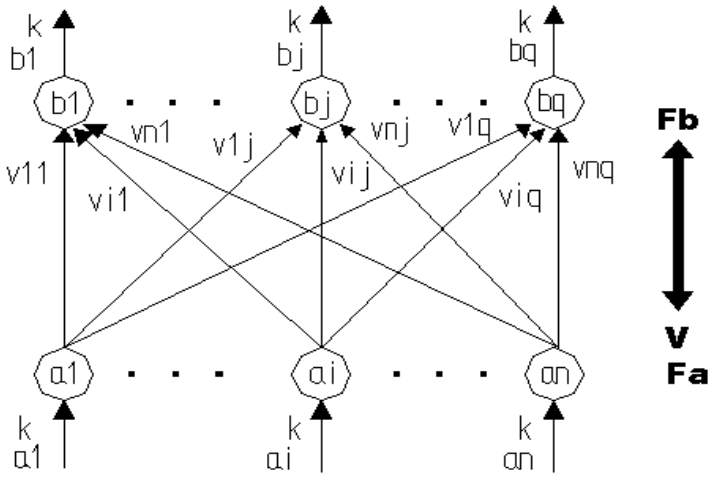


Рисунок 1.7 – Топология однонаправленного персептрона

Рисунок 1.8 иллюстрирует *удвоенный персептрон* – многослойный персептрон со статическими случайно установленными весами связей между элементами F_a и F_b и подстраивающимися весами связей между элементами слоя F_b и слоя F_c.

Элементарный персептрон – это двухслойный связный преобразователь наборов, который запоминает примерные пары (A_kB_k), $k = 1…m.$, используя процедуру корректировки ошибок. F_a обеспечивает прием входных образов $A=(a_{1k}, a_{2k}, \ldots a_{nk})$, где $k = 1, 2, \ldots m$, а выходной слой F_b состоит из нейронов со ступенчатой функцией активизации и обеспечивает формирование выходных бинарных образов $B = (b_1, b_2, \ldots, b_p)$, принимающих значения 0 и 1.

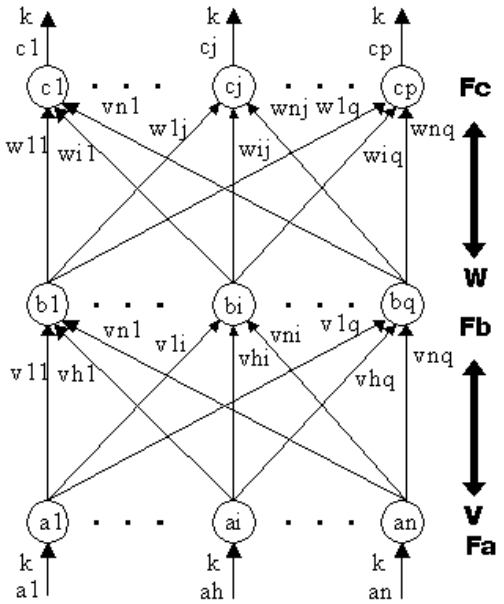


Рисунок 1.8 – Удвоенный персептрон

Исследователями доказано два положения. Первое, утверждавшее, что персептрон способен обучиться всему, что он может представлять, стимулировало его дальнейшие исследования (Ф. Розенбалатт). Второе, доказанное М.Минским, о способности представлять лишь ограниченный класс *линейно-разделимых образов*, развеяло иллюзии насчёт возможностей персептрона, и работы в данном направлении были прекращены. Проиллюстрируем это утверждение на примере задачи разделения образов на два класса в случае двух признаков. Выходной слой персептрона состоит из одного нейрона вида

$$b=f(w_1a_1 + w_2a_2 - \theta),$$

где b – выход нейрона;

f – ступенчатая функция активации;

θ – значение порога.

Пусть и входные признаки принимают бинарные значения 0 или 1. Комбинацию признаков можно представить на плоскости OXY (рисунок 1.9). Значения w_1, w_2 и порога θ определяют уравнение $w_1a_1 + w_2a_2 = \theta$, делящее плоскость на две части, два класса:

$$F(w_1,w_2,\theta)=1,\ w_1a_1 + w_2a_2 - \theta > 0$$
$$-1,\ w_1a_1 + w_2a_2 - \theta \le 0$$

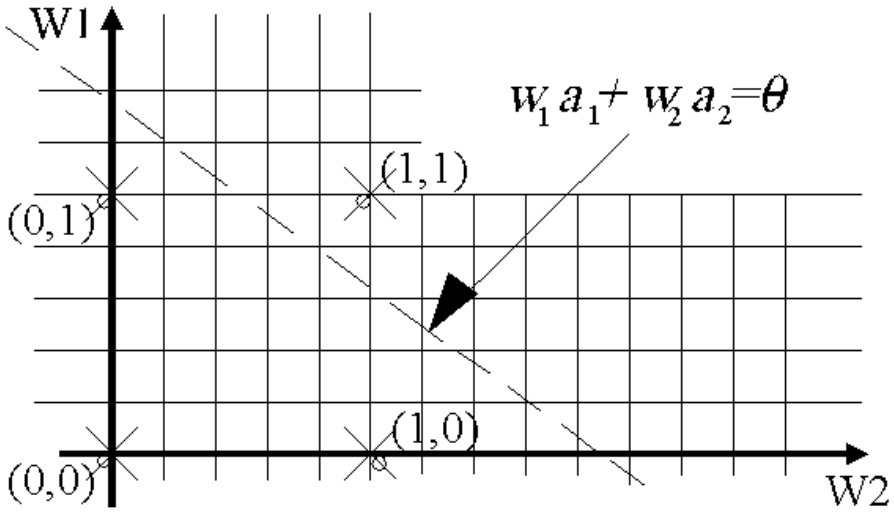


Рисунок 1.9 – Деление плоскости на классы

Этот пример иллюстрирует, что возможности персептрона ограничены классом линейноразделимых образов. Подобная топология ИНС не позволяет реализовать логическую функцию «*исключающее ИЛИ*»:

$$f(x_1,x_2)=x_1x_2+x_1x_2=x_1+x_2,$$

так как невозможно подобрать прямую так, чтобы точки (1,1) и (0,0) находились по одну сторону от нее, а точки (1,0) и (0,1) по другую. Такие рассуждения справедливы и для произвольного числа признаков и выходных классов. В этом случае разделяющая функция будет представлять собой гиперплоскость в *n*-мерном пространстве признаков.

2 АЛГОРИТМЫ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

Описать процедуру построения персептрона можно следующим образом:

1. Инициализировать весовые коэффициенты W_{ij} и пороговые значения θ_i всех нейронов случайными числами, принимающими значения в интервале [+1,-1]:

$$W_{ij}(0) = r,\ \theta_i = r,\ i = 1\ldots n,\ \ j = 1\ldots p.$$

2. Для каждого набора (A_k, B_k) обучающей выборки $(k=1\ldots m)$ выполнить:

2.1 активацию входного слоя F_a вектором A_k , то есть $a_i = a_i,\ i = 1\ldots n;$

2.2 вычисление сигналов на выходах нейронов слоя F_b по формуле:

$$b_i = f\ (\Sigma W_{ij} a_i - \theta_j),\ \ j = 1\ldots p;\ f(x) = 1,\ x \exists 0;$$
$$-1,\ x[0$$

2.3 вычисление ошибки между выходными величинами b_j и компонентами желаемого образа B_k примерного набора:

$$e_i = b_i - b_j,\ \ j = 1\ldots p;$$

2.4. корректировку весовых коэффициентов согласно соотношению:

$$Wij(t+1) = Wij(t) + \acute{\alpha} * aiej,\ \ i = 1\ldots n,\ \ j = 1\ldots p,$$

где $\acute{\alpha}$ - скорость обучения.

3. Повторить шаг 2 пока ошибки e_j не станут достаточно малыми величинами для всех обучающих примеров.

Ф. Розенблатт предложил экспериментальный алгоритм для настройки связей между слоями для удвоенного персептрона (рисунок 1.6).

Процедура, о которой идет речь, называется *процедурой корректировки ошибок на основе обратного распространения*, поскольку она основана на получении ошибок в выходном слое, распределении корреляции в обратном

направлении к входам сети в случае неудовлетворительного значения, полученного на выходе. Процедура настройки связей к заданному слою, будь он скрытым или выходным, совершенно идентична процедуре настройки для элементарного персептрона.

Персептрон имел огромный успех как фундамент последующих более мощных моделей, таких как *Adaline* и *Backpropogation*.

Итак, персептрон ограничен условием линейной разделимости входных образов, требует долгого обучения с учителем. Сильная сторона персептрона – в его хорошо понятном поведении, возможности хранить данные и немедленной выдаче результата. Персептрон используют для решения задач распознавания наборов, последовательностей, для обработки изображений, принятия управленческих решений; он идеально подходит для проблем, допускающих два возможных ответа. Трудности, связанные с использованием персептрона на практике, состоят в том, что затруднительно заранее проверить условие линейной разделимости симптомов и предсказать время обучения сети.

2.1 Алгоритм обратного распространения ошибки

Недостаток персептрона, связанный с ограничением представляемых функций классом линейноразделимых, может быть легко преодолен введением в сеть промежуточных слоев. В самом деле, каждый нейрон скрытого слоя трехслойной ИНС характеризуется некоторой гиперповерхностью в пространстве входных параметров. В выходном слое мы получаем некоторые выпуклые области между гиперплоскостями. Возможности нейронной сети можно увеличить введением новых промежуточных слоев. Так, возможности сети с двумя промежуточными слоями же не ограничены только выпуклыми областями. На их выходе можно получать классификацию по более сложным невыпуклым областям. Таким образом, выбором большого количества нейронов можно аппроксимировать многомерные области произвольной формы. Возможности многослойных ИНС известны давно, однако их развитие сдерживалось долгое время отсутствием теоретически обоснованного алгоритма обучения. Важную роль в возрождении интереса к НС сыграло появление алгоритма обратного распространения ошибки (*Backpropogation* – ВР). Этот алгоритм предназначен для обучения нейронных сетей с многослойной структурой, причем количество слоев может быть произвольным. Требования накладываются лишь на функцию активации, которая должна быть дифференцируема. Обычно использует сигмоидную функцию:

$$S(x)=1/(1+1/exp(x))$$

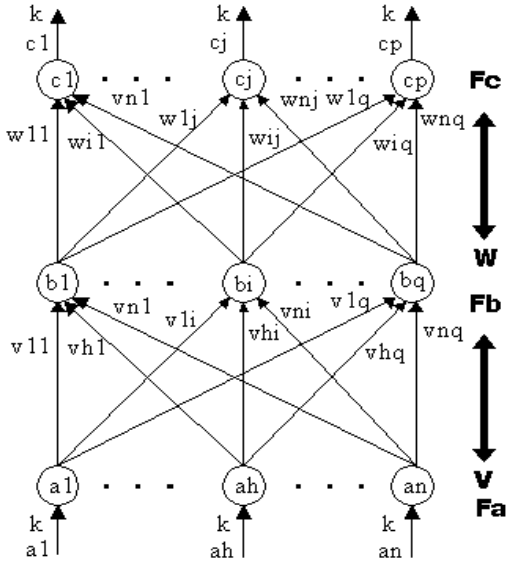


Рисунок 2.1 – Топология элементарной ВР-нейронной сети

Элементарная топология НС обратного распространения ошибки изображена на рисунке 2.1. Это трехслойная ИНС с прямыми межслоевыми связями. Сеть ВР сохраняет пары (A_k, C_k) обучающего набора, используя алгоритм межслоевого градиентного спуска корректировки ошибок. Существует множество различных вариантов этого алгоритма, поскольку он был открыт независимо разными учеными и до сих пор тщательно изучается и анализируется. Суть его заключается в получении корректного отображения входных образов в выходные путем минимизации весовой функции, либо квадрата ошибки результата, либо линейной ошибки, либо степени r абсолютной погрешности на выходе.

2.2 Алгоритм настройки сети по квадратичной погрешности

Необходимо выполнить следующее:

1. Назначить случайные значения в диапазоне [-1, +1] всем связям между слоями F_a и F_b, V_{hi}, всем связям между слоями F_b и F_c, W_{ij}, порогам элементов слоя F_b, θ_i, порогам элементов слоя F_c, Γ_j.

2. Для каждой обучающей пары (A_k, C_k) выполнить следующее:

а) Передать входные сигналы A_k на элементы слоя F_a, пропустив их через матрицу V и вычислить новые значения элементов слоя F_b, используя формулу:

$$b_i=f(\sum a_h V_{hi}+\theta_i),\ \ i=1\ldots p,$$

где b_i – значение, активизированное на выходе i-того элемента слоя;

θ_i – порог i-того элемента слоя F_b;

f – логическая пороговая сигмоидная функция $S(x)=1/(1+1/\exp(x))$

б) Пропустить полученные значения выходных сигналов в слое F_b через матрицу W:

$$c_j=f(\sum b_i w_{ij}+\Gamma_j),\ j=1\ldots q,$$

где c_j – значение, активизированное на выходе j-того элемента слоя F_c;

Γ_j – порог j-того элемента слоя F_c.

в) Вычислить ошибки в слое F_b, связанные с погрешностями:

$$e_i=b_i(1-b_i)\sum w_{ij}d_j,\ \ i=1\ldots p,$$

где e_i – i-ая вычисленная ошибка слоя.

д) Настроить связи от F_b к F_c:

$$\Delta w_{ij}=\alpha b_i d_j,\ i=1\ldots p,\ j=1\ldots q,$$

где Δw_{ij} – величина, на которую следует изменить вес связи от i-го элемента слоя F_b к j-му элементу слоя F_c;

α – положительный коэффициент скорости обучения.

е) Отрегулировать пороги всех элементов слоя F_c:

$$\Delta\Gamma_j=\alpha d_i,\ j=1\ldots q,$$

где $\Delta\Gamma_j$ – величина, на которую следует изменить пороговое значение элемента F_b;

ж) Настроить связи от F_a к F_b:

$$\Delta Vh_i=\beta a h e_i,\ i=1\ldots p,\ h=1\ldots n,$$

где ΔV_{hi} – величина, на которую следует изменить вес связи от h-го элемента слоя F_a к i-тому элементу слоя F_b;

β – положительный коэффициент скорости обучения;

з) Отрегулировать пороги всех элементов слоя F_b:

$$\Delta\theta_i=\beta e_i,\ i=1\ldots p,$$

где $\Delta\theta_i$ – величина, на которую следует изменить пороговое значение элемента.

Повторить шаг 2 до тех пор, пока значение ошибки d_i не станет равным нулю или достаточно малым для всех обучающих пар. Алгоритм ВР широко используется в различном программном обеспечении, однако у него есть существенные недостатки, например, длительное время обучения. В случае сложных задач он вообще не позволяет найти корректное решение, что может быть обусловлено двумя причинами: параличом сети, когда все нейроны функциониру-

ют на грани насыщения с большими значениями выхода, и попаданием в ловушку, локальный минимум целевой функции. Паралича сети избегают снижением скорости обучения (коэффициенты α, β), увеличивая тем самым время обучения сети. Попадание в локальный минимум объясняется тем, что изрезанная целевая функция может иметь нескольких локальных минимумов. В каждом из них вектор-градиент близок к 0, поэтому градиентный алгоритм поиска минимума прекращает свою работу, сеть недоучивается, а синаптическая карта (весовая матрица) не является оптимальной.

2.3 Алгоритм «машины» Больцмана

Для предотвращения «зависания» сети используют стохастические методы обучения НС, обладающие способностью находить глобальный минимумом целевой функции. Суть такого подхода заключается в случайном преобразовании весовых коэффициентов сети и сохранении тех, которые ведут к уменьшению заданной функции. Основной вопрос – в выборе величины случайных приращений, позволяющих «выбираться» из локальных минимумов и достигать глобального. Ученые провели аналогию между процессом оптимизации синаптической карты нейронной сети и физическими процессами, происходящими при отжиге металла. В расплавленном металле атомы находятся в беспорядочном движении, при понижении температуры они стремятся перейти в состояние энергетического минимума (кристаллизации), то есть к глобальному минимуму.

Нейронная сеть, основанная на принципах стохастического обучения, называется машиной Больцмана и функционирует согласно следующему алгоритму (рисунок 1.8):

1. Назначить случайные значения в диапазоне [+1,-1] всем связям между слоями F_a и F_b, V_{hi} всем связям между слоями F_b и F_c, W_{ij}, выбрать начальное значение T (1) искусственной температуры достаточно большим.

2. Начиная с момента времени $t = 1$ для каждой пары (A_k, C_k)обучающего примера проделать следующее:

а) Пустить на входы НС входной образ A_k, а на выходы сети – выходной образ C_k.

б) Случайным образом выбрать элемент b_i слоя F_b и изменить его состояние по формуле:

$$b_i=1,\ b_i=0$$
$$0,\ b_i\neq 0.$$

в) Вычислить изменение общей энергии по отношению к измененному элементу:

$$\Delta E_i=\sum Vh_i b_i+\sum w_{ij} b_i$$

г) Если $\Delta E_i<0$, сохранить изменение, если $\Delta E_i>0$, вычислить распределение Больцмана $P_i=exp(-\Delta E_i/T(t))$; сгенерировать случайное число r, принимающее значения на интервале [0, 1],

если $P_i>r$, сохранить изменение,

если $P_i<r$, возвратиться к первоначальному состоянию (шаг б).

д) Выбрать новый элемент F_b и повторить шаги (б, в, г).

е) Увеличить время t на 1 и вычислить новое значение для температуры $T(t)$:

$$T(t)=T(1)/(1+log(t))$$

ж) Повторить шаги (б-е) до тех пор, пока не будут получены равенства $\Delta E_i=0$, что соответствует равновесному состоянию сети и получению глобального минимума.

з) Сохранить активизированные значения всех элементов слоя F_b в виде вектора D_k (для дальнейшего статистического анализа равновесных состояний):

$$d_i=b_i,\ i=1\ldots p,$$

где d_i – i-тая составляющая вектора D_k.

3. Используя вычисленные состояния равновесия D_k, вычислить вероятность нахождения h-го элемента слоя F_a и i-го элемента F_b в одинаковом состоянии:

$$Qh_i=1/m(\sum\Phi(ah, d_i)),\ h=1\ldots n,\ i=1\ldots p,$$

где функция корреляции $\Phi(x,y)=1,\ x=y$

$0,\ x\neq y,$

где x и y принимают двоичные значения.

Таким же образом вычислить вероятность нахождения j-го элемента слоя F_c и i-го элемента F_b в одинаковом состоянии:

$$R_{ij}=1/m(\sum\Phi(c_j,d_i)),\ j=1\ldots q,\ i=1\ldots p.$$

4. Снова начать с момента времени $t = 1$ и для каждой пары обучающей выборки $A_k,\ k = 1\ldots m$ проделать следующее:

а) Подать значения A_k на входные элементы нейросети F_a.

б) Случайным образом выбрать элемент b_i слоя F_b и изменить его состояние по формуле:

$b_i=1,\ b_i=0$

$0,\ b_i\neq 0$

в) Вычислить изменение общей энергии по отношению к измененному элементу:

$$\Delta E_i=\sum V_{hi}b_i.$$

г) Если $\Delta E_i<0$, сохранить изменение,

если $\Delta E_i>0$, вычислить распределение Больцмана для данной ΔE_i $P_i=exp(-\Delta E_i/T(t))$. Сгенерировать случайное число r, принимающее значения на интервале [0,1],

если $P_i>r$, сохранить изменение,

если $P_i<r$, возвратиться к первоначальному состоянию (шаг б).

д) Повторить еще раз шаги (б, в, г).

е) Увеличить время t на 1 и вычислить новое значение для температуры $T(t)$:

$$T(t)=T(1)/(1+log(t)).$$

ж) Повторить шаги (б-е) до тех пор, пока не будут получены равенства $\Delta E_i=0$, что соответствует равновесному состоянию сети и получению глобального минимума энергии.

з) Сохранять активизированные значения всех элементов слоя F_b в виде вектора:

$$D_k:d_ik = bi,\ i=1\ldots p,$$

где d_i – i-ая составляющая вектора D_k.

5. Используя вычисленные состояния равновесия D_k, вычислить вероятность нахождения i-того элемента слоя F_a и i-го элемента F_b в одинаковом состоянии:

$$Q'h_i=1/m(\sum\Phi(ah, d_i)),\ h=1\ldots n,\ i=1\ldots p.$$

Таким же образом вычислить вероятность нахождения j-го элемента слоя F_c и i-го элемента F_b в одинаковом состоянии:

$$R'_{ij}=1/m(\sum\Phi(c_j,d_i)),\ j=1\ldots q,\ i=1\ldots p.$$

6. Настроить весовые матрицы W и V.

а) Отрегулировать связи между слоями F_a и F_b:

$$\Delta V_{hi}=\alpha(Q_{hi}-Q'_{hi}),$$

где ΔV_{hi} – приращение веса связи от h-го элемента F_a к i-му элементу F_b;

α – коэффициент скорости обучения.

б) Отрегулировать связи между слоями F_b и F_c:

$$\Delta w_{ij}=\alpha(R_{ij}-R'_{ij}),$$

где Δw_{ij} – приращение веса связи от i-го элемента F_b к j-му элементу F_c.

7. Повторить шаги 2-6 до тех пор, пока ΔV_{hi} и Δw_{ij} не станут достаточно малыми.

Приведенный стохастический алгоритм позволяет достичь точки глобального минимума, однако время обучения сети при этом значительно увеличивается и возможность «зацикливания» в локальном минимуме хотя и маловероятна, но все же существует.

2.4 Алгоритм встречного распространения ошибки

Алгоритм встречного распространения ошибки (*counterpropogation* – CPN) позволяет сократить время обучения сети, но не является таким же общим, как и backpropogation. Это полезно для случаев ограничения по времени. Структура сети CPN представлена на рисунке 2.2. Трехслойная нейронная сеть осуществляет отображение посредством первичной классификации входного образа элементами слоя F_b, затем передачей сигналов от выигравшего элемента к элементам слоя F_c. Во время преобразования сигнала в слое F_b происходит незаметное соревнование между элементами для определения класса представленного образа. Соревнование осуществляется через горизонтальные связи, которые соединяют каждый элемент слоя F_b с каждым другим элементом слоя F_b, а также с самим собой.

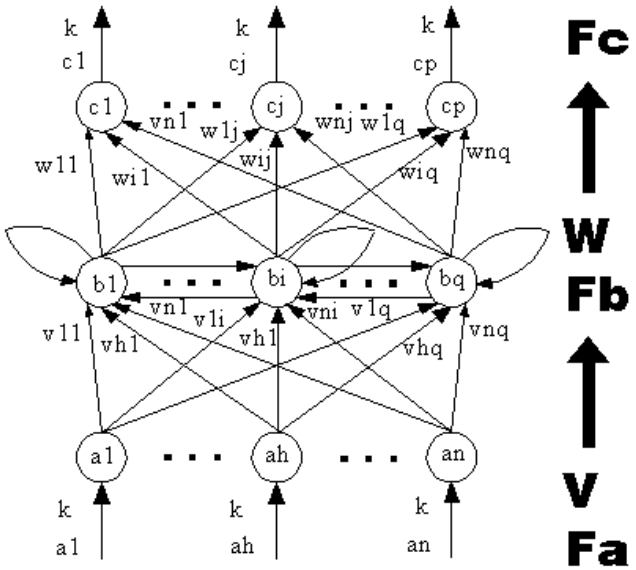


Рисунок 2.2 – Топология CPN-нейросети с обучением без учителя

Алгоритм является объединением двух известных нейросетевых парадигм: *самоорганизующиеся карты Кохонена* и *сети Гроссберга*. Слой F_b – слой Кохонена – обучается без учителя по принципу: «победитель забирает все». Для заданного входного образа активизируется лишь один элемент слоя

Кохонена – тот, у которого наибольший сигнал на выходе. Таким образом, осуществляется классификация входных сигналов на группу схожих. Выигравший элемент вместе со всеми элементами слоя F_c образует топологию Гроссберга – минимальную нейросетевую отображающую структуру. Этот участок сети обучается уже «с учителем». В результате обучения слоя Кохонена происходит классификация входных образов, которая далее используется в слое Гроссберга для получения на выходе сети заданных образов.

Базовый алгоритм встречного распространения ошибки можно представить состоящим из следующих основных шагов:

1. Нормализовать:

$$A_k: a_{hk}=a_{hk}/|A_k|,\ h=1\ldots n,\ k=1\ldots m.$$

2. Присвоить всем связям от слоя F_a к слою F_b случайные значения в диапазоне [0, 1].

3. Связи от слоя F_a к i-му элементу слоя F_b образуют весовой вектор $V_i=(v_{1i}, \ldots v_{hi}, \ldots v_{ni})$. Эти веса нормализуются:

$$v_{hi}=v_{hi}/|V_i|,\ h=1\ldots n,\ i=1\ldots m.$$

4. Для каждой пары (A_k, D_k) проделать следующее:

а) Найти V_i, ближайший к A_k:

$$|A_k-V_g| = MIN\ |A_k-V_i|.$$

б) Приблизить V_g к A_k:

$$\Delta V_{hg}=\alpha(t)(a_h-V_{hg}),\ h=1\ldots n,$$

где ΔV_{hg} – приращение веса связи от h-того элемента F_a к g-тому элементу F_b;

$\alpha(t)$ – коэффициент скорости обучения, определяемый одной из формул:

$$\alpha(t)=1/t \text{ или } \alpha(t)=0{,}2(1-1/10000).$$

в) Вновь нормализовать V_g:

$$v_{hg}=v_{hg}/|V_g|,\ h=1\ldots n.$$

г) Вычислить выходные величины нейронов слоя F_b Кохонена:

$$b_i=\begin{cases}1, & i=g\\ 0, & i\neq g.\end{cases}$$

д) Вычислить выходные величины слоя F_c Гроссберга согласно выражению:

$$c_j=\sum b_i w_{ij},\ j=1\ldots q.$$

е) Определение ошибки между вычисленными выходными величинами и компонентами желаемого выходного вектора D_k обучающей выборки:

$$e_j=d_j-c_j, j=1\ldots q.$$

ж) Корректировка синаптических весов w_{gj} нейронов слоя F_b Гроссберга по формуле:

$$w_{gj}(t+1)=w_{gj}(t)+\beta(t)b_g e_j,\ j=1\ldots q.$$

5. Повторение шага 3 алгоритма до тех пор, пока ошибки e_j не станут достаточно малыми величинами для всех обучающих примеров (A_k, D_k).

Как видно из алгоритма, значительно сократить время обучения сети позволяет тот факт, что настраиваются не все связи слоя Гроссберга, а лишь связанные с «выигравшим» нейроном слоя Кохонена. Рассмотренный метод функционирования нейросети с активизацией лишь одного нейрона во внутреннем слое, называется *методом аккредитации*. Используют также более общий метод интерполяции, при котором в слое Кохонена активизируется целая группа нейронов с наибольшими значениями выхода. В этом случае вектор выходов нейронов группы нормализуется и используется далее для вычисления входов нейронов Гроссберга. При этом реализуется более сложная функциональная зависимость: метод интерполяции позволяет реализовывать более сложные ассоциации образов.

При реализации алгоритмов обучения на персональном компьютере необходимо учитывать две характеристики:

1. Объем оперативной памяти, который требуется для размещения вспомогательных переменных алгоритма.
2. Время обучения, то есть число итераций обучения, умноженное на время выполнения одной итерации.

Далее представлено описание и проведен сравнительный анализ алгоритмов обучения, которые при реализации на персональном компьютере требуют строго меньше $2*N$ вспомогательных переменных, где N – число параметров сети, настраиваемых в процессе обучения (синаптических весов и смещений). Это следующие алгоритмы:

1. Обобщенный градиентный алгоритм обучения,
2. Градиентный алгоритм обучения с автоматическим определением длины шага (автономный градиентный алгоритм обучения),
3. Алгоритм поиска в случайном направлении,
4. Градиентный алгоритм обучения с одномерной оптимизацией,
5. Градиентный алгоритм обучения с одномерной оптимизацией и с автоматическим определением длины шага.

В обобщенном градиентном алгоритме на каждой итерации выполняется вычисление градиента функции ошибки (определяются значения частных производных по синаптическим весам и смещениям) и делается шаг в направлении антиградиента. Величина шага задается пользователем.

Обобщенный градиентный алгоритм обучения в отличие от традиционного метода обратного распространения ошибки дает возможность обучать многослойные сети с произвольным числом слоев. Использование двойственных переменных ускоряет процесс обучения.

Градиентный алгоритм с автоматическим определением длины шага определяется следующим набором параметров:

- начальное значение шага,
- количество итераций, через которое происходит запоминание данных сети (синаптических весов и смещений),
- величина (в процентах) увеличения шага после запоминания данных сети, и величина уменьшения шага в случае увеличения функции ошибки.

В начале обучения с помощью автономного градиентного алгоритма записываются на диск значения весов и смещений сети. Затем происходит заданное число итераций обучения с заданным шагом. Если после завершения этих итераций значение функции ошибки не возросло, то шаг обучения увеличивается на заданную величину, а текущие значения весов и смещений записываются на диск. Если на некоторой итерации произошло увеличение функции ошибки, то с диска считываются последние запомненные значения весов и смещений, а шаг обучения уменьшается на заданную величину.

При использовании автономного градиентного алгоритма происходит автоматический подбор длины шага обучения в соответствии с характеристиками адаптивного рельефа. Замечено, что в начале обучения шаг должен быть порядка 0,1, а в конце обучения – от 10^5 до 10^6. Автономный алгоритм по сравнению с обобщенным градиентным алгоритмом имеет преимущество в том, что шаг обучения автоматически увеличивается, подстраиваясь под адаптивный рельеф. Тем самым существенно сокращается количество шагов, которое требуется для обучения сети.

В алгоритмах с одномерной оптимизацией на каждом шаге обучения выполняется два пробных шага: один в направлении антиградиента, другой - в противоположном направлении. По трем точкам (включая исходную точку итерации) строится парабола. Для параболы с лучами, направленными вверх, шаг делается в вершину параболы. Если лучи параболы направлены вниз, то выполняется шаг в сторону антиградиента на заданную величину. Таким образом, а этих алгоритмах адаптивный рельеф вдоль антиградиента функции ошибки аппроксимируется параболой.

В программной реализации алгоритма с одномерной оптимизацией исключено увеличение значения функции ошибки. Значения этой функции запоминаются для четырех точек (исходная точка, два пробных шага и минимум параболы). Шаг делается в точку с минимальным значением функции ошибки.

В алгоритме с одномерной оптимизацией и автоматическим определением длины шага выполняются действия, аналогичные автономному градиентному алгоритму. Величина пробных шагов подбирается под адаптивный рельеф. Данный способ исключения шагов с увеличением функции ошибки позволяет уменьшить количество вычислений на каждом шаге алгоритма.

В алгоритме поиска в случайном направлении на каждой итерации делается шаг, направление которого задается случайным образом. Если данный шаг приводит к увеличению функции ошибки, то происходит возврат к исходным значениям синаптических весов и смещений и выполняется шаг в другом случайном направлении. Компьютерные эксперименты по обучению нейронных сетей показали, что наибольшей скоростью сходимости среди всех описанных алгоритмов обладает автономный градиентный алгоритм.

Таблица 2.1 – Сравнение алгоритмов обучения

Название алгоритма обучения	Один шаг, сек	Количество шагов решения контрольной задачи	Объем вычислений
Обобщенный градиентный алгоритм	2.02	715	2.08
Градиентный алгоритм с автоматическим определением длины шага (автономный градиентный алгоритм)	2.03	342	1.00
Алгоритм поиска в случайном направлении	1.64	52700	124.48
Градиентный алгоритм с одномерной оптимизацией	6.85	206	2.03
Градиентный алгоритм с одномерной оптимизацией и автоматическим определением длины шага	5.34	290	2.23

3 КЛАССИФИКАЦИЯ ТИПОВ НЕЙРОННЫХ СЕТЕЙ

3.1 Однослойный персептрон

Описание: Однослойный персептрон способен распознавать простейшие образы. Отдельный нейрон вычисляет взвешенную сумму элементов входного сигнала, вычитает значение смещения и пропускает результат через жесткую пороговую функцию, выход которой равен +1 или -1. В зависимости от значе-

ния выходного сигнала принимается решение: +1 – входной сигнал принадлежит классу *A*, -1 – входной сигнал принадлежит классу *B*. Персептрон, состоящий из одного нейрона, формирует две решающие области, разделенные гиперплоскостью.

Области применения: Распознавание образов, классификация.

Преимущества: Программные или аппаратные реализации модели очень просты. Простой и быстрый алгоритм обучения.

Недостатки: Примитивные разделяющие поверхности (гиперплоскости) дают возможность решать лишь самые простые задачи распознавания.

3.2 Нейронная сеть с обучением по методу обратного распространения ошибки

Описание: Алгоритм обратного распространения – это итеративный градиентный алгоритм, который используется с целью минимизации среднеквадратичного отклонения текущего выхода многослойного персептрона и желаемого выхода.

Области применения: Распознавание образов, классификация, прогнозирование.

Преимущества: Эффективный алгоритм обучения многослойных нейронных сетей.

Недостатки: Многокритериальная задача оптимизации рассматривается как набор однокритериальных – на каждой итерации происходят изменения значений параметров сети, улучшающие работу лишь с одним примером обучающей выборки. Такой подход существенно уменьшает скорость обучения.

3.3 Нейронная сеть с «генетическим» алгоритмом обучения

Описание: Инициализируется популяция и все хромосомы (*хромосома* – это вектор, состоящий из набора синаптических весов и смещений) сравниваются в соответствии с выбранной функцией оценки. Далее (возможно многократно) выполняется процедура репродукции популяции хромосом. Родители выбираются случайным образом в соответствии со значениями оценки (вероятность того, что данная хромосома станет родителем, пропорциональна полученной оценке). Репродукция происходит индивидуально для одного родителя путем мутации хромосомы либо для двух родителей путем кроссенговера генов. Получившиеся дети оцениваются в соответствии с заданной функцией и помещаются в популяцию. В результате использования описанных операций на каждой стадии эволюции получаются популяции с все более совершенными индивидуумами.

Области применения: Распознавание образов, классификация, прогнозирование

Преимущества: Генетические алгоритмы эффективны в поиске глобальных минимумов адаптивных рельефов, так как в них исследуются большие области допустимых значений параметров нейронных сетей. Генетические алгоритмы дают возможность оперировать дискретными значениями параметров нейронных сетей. Это упрощает разработку цифровых аппаратных реализаций нейронных сетей. При обучении на компьютере нейронных сетей, не ориентированных на аппаратную реализацию, возможность использования дискретных значений параметров в некоторых случаях может приводить к сокращению общего времени обучения.

Недостатки: Генетические алгоритмы обучения сложны для понимания и программной реализации

3.4 Сеть Хопфилда

Описание: Исходными данными для расчета значений синаптических весов сети являются векторы – образцы классов. Сеть функционирует циклически. Выход каждого из нейронов подается на входы всех остальных нейронов. Нейроны сети имеют жесткие пороговые функции.

Области применения: Ассоциативная память, адресуемая по содержанию, задачи распознавания образов, задачи оптимизации

Преимущества: Простота

Недостатки: Сеть обладает небольшой емкостью. Наряду с запомненными образами в сети хранятся и их негативы. Размерность и тип входных сигналов в точности совпадают с размерностью и типом выходных сигналов. Это существенно ограничивает применение сети в задаче распознавания образов. При использовании коррелированных векторов-образцов возможно зацикливание сети в процессе функционирования. При увеличении размерности входного сигнала квадратично растёт число синапсов.

3.5 Сеть Кохонена

Описание: Алгоритм Кохонена дает возможность строить нейронную сеть для разделения векторов входных сигналов на подгруппы. Сеть состоит из нейронов, образующих прямоугольную решетку на плоскости. Элементы входных сигналов подаются на входы всех нейронов сети. В процессе работы алгоритма настраиваются синаптические веса нейронов. Входные сигналы последовательно предъявляются сети. Желаемые выходные сигналы не определяются. После того, как было предъявлено достаточное число входных векторов, синаптические веса сети определяют кластеры. Кроме того, веса организуются так, что топологически близкие узлы чувствительны к похожим внешним воздействиям.

Области применения: Кластерный анализ, распознавание образов, классификация

Преимущества: Способна функционировать в условиях помех, так как число классов фиксировано, веса модифицируются медленно, настройка весов заканчивается после обучения

Недостатки: Сеть может быть использована для кластерного анализа только в том случае, если заранее известно число кластеров.

3.6 Сеть поиска максимума

Описание: Сеть циклического функционирования. На каждой итерации большие сигналы на выходах нейронов подавляют слабые сигналы на выходах других нейронов. Если в начале функционирования сети сигнал на выходе одного из нейронов имел максимальное значение, то в конце функционирования все выходы нейронов, кроме максимального, имеют значения, близкие к нулю. Таким образом, нейрон с наибольшим выходным сигналом побеждает. Это достигается за счет использования латеральных связей.

Области применения: Используется в составе нейросетевых систем распознавания образов

Преимущества: Простота

Недостатки: Число итераций функционирования сети заранее не определено. Сеть определяет, какой из входных сигналов имеет максимальное значение, но в процессе функционирования теряет само значение максимального сигнала. При увеличении размерности входного сигнала квадратично растёт число синапсов

3.7 Нейронные сети, обучаемые по методу имитации отжига

Описание: Разновидности:

1. Градиентный алгоритм с изменением величины шага по правилу отжига. На каждой итерации вычисляется направление антиградиента адаптивного рельефа и делается шаг заданной величины. В процессе обучения величина шага уменьшается с увеличением номера итерации.
2. Стохастический алгоритм. В процессе обучения совершаются шаги по адаптивному рельефу в случайных направлениях.

Области применения: Обучение как многослойных, так и полносвязных сетей. Можно строить любые отображения XY, где X и Y – векторы некоторой размерности. К этому сводятся многие задачи распознавания образов, адаптивного управления, многопараметрической идентификации, прогнозирования и диагностики.

Преимущества: «Тепловые флуктуации», заложенные в алгоритм дают возможность не задерживаться в локальных минимумах. Показано, что алгоритм отжига может быть использован для поиска глобального оптимума адаптивного рельефа нейронной сети.

Недостатки: Низкая скорость сходимости при обучении нейронных сетей большой размерности

3.8 Сеть теории адаптивного резонанса

Описание: Сеть обучается без учителя и реализует алгоритм кластеризации. В соответствии с этим алгоритмом первый входной сигнал считается образцом первого кластера. Следующий входной сигнал сравнивается с образцом первого кластера. Говорят, что входной сигнал «следует за лидером» и принадлежит первому кластеру, если расстояние до образца первого кластера меньше порога. В противном случае второй входной сигнал – образец второго кластера. Этот процесс повторяется для всех следующих входных сигналов. Таким образом, число кластеров растет с течением времени и зависит как от значения порога, так и от метрики расстояния, использующейся для сравнения входных сигналов и образцов классов.

Области применения: Распознавание образов, кластерный анализ

Преимущества: Обучение без учителя

Недостатки: Неограниченное увеличение числа нейронов в процессе функционирования сети.

В присутствии шума возникают значительные проблемы, связанные с неконтролируемым ростом числа образцов.

3.9 Двунаправленная ассоциативная память

Описание: ДАП относится к гетероассоциативной памяти. Входной вектор поступает на один набор нейронов, а соответствующий выходной вектор вырабатывается на другом наборе нейронов. Входные образы ассоциируются с выходными. ДАП способна к обобщению, вырабатывая правильные выходные сигналы, несмотря на искаженные входы.

Области применения: Ассоциативная память, распознавание образов

Преимущества: Дает возможность строить ассоциации между векторами *A* и *B*, которые в общем случае имеют разные размерности. ДАП – простая сеть, которая может быть реализована в виде отдельной СБИС или оптоэлектронным способом. Процесс формирования синаптических весов простой и быстрый. Сеть быстро сходится в процессе функционирования.

Недостатки: Емкость ДАП жестко ограничена. ДАП обладает некоторой непредсказуемостью в процессе функционирования, возможны ложные ответы.

3.10 Машина Больцмана

Описание этапов больцмановского обучения:

1. Определить переменную *T*, представляющую искусственную температуру.
2. Предъявить сети множество входов и вычислить выходы и целевую функцию.
3. Дать случайное изменение весу и пересчитать выход сети и изменение целевой функции в соответствии со сделанным изменением веса.
4. Если целевая функция улучшилась (уменьшилась), то сохранить изменение веса.

Области применения: Распознавание образов, классификация

Преимущества: Алгоритм дает возможность сети выбираться из локальных минимумов адаптивного рельефа

Недостатки: Медленный алгоритм обучения

3.11 Сеть встречного распространения

Описание: В сети встречного распространения объединены два алгоритма: самоорганизующаяся карта Кохонена и звезда Гроссберга. Сеть встречного распространения имеет два слоя с последовательными связями. Первый слой - слой Кохонена, второй – слой Гроссберга. Каждый элемент входного сигнала подается на все нейроны слоя Кохонена. Каждый нейрон слоя Кохонена соединен со всеми нейронами слоя Гроссберга. В процессе обучения сети встречного распространения входные векторы ассоциируются с соответствующими выходными векторами. Эти векторы могут быть двоичными или непрерывными. После обучения сеть формирует выходные сигналы, соответствующие входным сигналам. Обобщающая способность сети дает возможность получать правильный выход, когда входной вектор неполон или искажен.

Области применения: Распознавание образов, восстановление образов (ассоциативная память), сжатие данных (с потерями).

Преимущества: Сеть встречного распространения проста. Она дает возможность извлекать статистические свойства из множеств входных сигналов. Сеть быстро обучается. Сеть полезна для приложений, в которых требуется быстрая начальная аппроксимация. Сеть дает возможность строить функцию и обратную к ней, что находит применение при решении практических задач.

Недостатки: Сеть не дает возможности строить точные аппроксимации (точные отображения).

3.12 Delta-Bar-Delta сеть

Описание: Алгоритм использует предыдущие значения градиента функции. Зная эту информацию, он совершает изменения в пространстве весов с помощью ряда эвристических правил. Используется эвристический подход.

Эвристики:

1. Если последовательные изменения веса имеют противоположные знаки, то значит данный вес осциллирует, и, следовательно, скорость обучения должна быть уменьшена.
2. Если серия последовательных изменений веса имеет одинаковые знаки, то скорость обучения должна быть увеличена.

Области применения: Распознавание образов, классификация, прогнозирование.

Преимущества: Ускоряется процесс конвергенции алгоритма обратного распространения за счет использования дополнительной информации об изменении параметров и весов во время обучения.

Недостатки: Даже небольшое линейное увеличение коэффициента обучения может привести к значительному росту скорости обучения, что вызовет скачки в пространстве весов. Геометрическое уменьшение коэффициента иногда оказывается не достаточно быстрым.

3.13 Расширенная Delta-Bar-Delta сеть

Описание: Введён параметр момент связи (*momentum*), представляющий собой некоторое число, пропорциональное предыдущему изменению веса. Значения момента используются для ускорения обучения с помощью ряда эвристических правил.

Области применения: Распознавание образов, классификация, прогнозирование.

Преимущества: Обучение происходит быстрее, чем в DBD сети.

Недостатки: То же, что и в DBD сети.

3.14 Сеть поиска максимума с прямыми связями

Описание: Многослойная сеть с прямыми связями. Входные сигналы попарно сравниваются друг с другом. Наибольший сигнал в каждой паре передается на следующий слой для дальнейших сравнений. На выходе сети лишь один сигнал имеет ненулевое значение. Он соответствует максимальному сигналу на входе сети. В основе лежит *нейросетевой компаратор*, который выполняет

сравнение двух аналоговых сигналов. На выходе – значение максимального сигнала. Выходы показывают, какой именно входной сигнал имеет максимальное значение.

Области применения: Применяется совместно с сетью Хемминга, в составе нейросетевых систем распознавания образов.

Преимущества: Заранее известен объем вычислений, который требуется для получения решения

Недостатки: Число слоев сети растет с увеличением размерности входного сигнала.

3.15 Сеть Хемминга

Описание: Реализует параллельное вычисление расстояний Хемминга от входного вектора до нескольких векторов. Расстояние Хемминга между двумя бинарными векторами одинаковой длины – это число несовпадающих бит в этих векторах.

Области применения: Распознавание образов, классификация, ассоциативная память, надежная передача сигналов в условиях помех.

Преимущества: Сеть работает предельно просто и быстро. Выходной сигнал (решение задачи) формируется в результате прохода через всего лишь один слой нейронов. В модели использован один из самых простых алгоритмов формирования синаптических весов и смещений сети. Ёмкость сети Хемминга не зависит от размерности входного сигнала, она в точности равна количеству нейронов.

Недостатки: Сеть способна правильно распознавать (классифицировать) только слабо зашумленные входные сигналы. Возможность использования только бинарных входных сигналов существенно ограничивает область применения.

3.16 Нейросетевой гауссов классификатор

Описание: В основе построения Гауссова классификатора лежат предположения о распределениях входных сигналов. Считается, что эти распределения известны и соответствует закону Гаусса.

Области применения: Распознавание образов, классификация

Преимущества: Программные или аппаратные реализации модели очень просты. Простой и быстрый алгоритм формирования синаптических весов и смещений.

Недостатки: Примитивные разделяющие поверхности (гиперплоскости) дают возможность решать лишь самые простые задачи распознавания. Счита-

ются априорно известными распределения входных сигналов, соответствующие разным классам.

3.17 Входная звезда

Описание: Входная звезда состоит из нейрона, на который подается группа входов через синаптические веса. Входная звезда обучается реагировать на определенный входной вектор и ни на какой другой. Это обучение реализуется путем настройки весов таким образом, чтобы они соответствовали входному вектору. Выход звезды определяется как взвешенная сумма ее входов. С другой точки зрения, выход можно рассматривать как свертку входного вектора с весовым вектором. Следовательно, нейрон должен реагировать наиболее сильно на входной образ, которому был обучен. После завершения обучения предъявление входного вектора будет активизировать обученный входной нейрон. Хорошо обученная входная звезда будет реагировать не только на определенный вектор, но и на незначительные изменения этого вектора. Таким образом, звезда будет проявлять способность к обобщению. Это достигается постепенной настройкой нейронных весов при предъявлении в процессе обучения векторов, представляющих вариации исходного входного вектора. Веса настраиваются таким образом, чтобы усреднить величины обучающих векторов, и нейроны получают способность реагировать на любой вектор этого класса.

Области применения: Распознавание образов

Преимущества: Входная звезда хорошо моделирует некоторые функции компонент живых (биологических) нейронных сетей. Сеть, включающая в себя входные звезды, может быть достаточно хорошей моделью отдельных участков мозга. При решении практических задач входные звезды могут быть использованы для построения простых быстро обучаемых сетей.

Недостатки: Каждая звезда в отдельности реализует слишком простую функцию. Из таких звезд невозможно построить нейронную сеть, которая реализовала бы любое заданное отображение. Это ограничивает практическое применение входных звезд.

3.18 Выходная звезда

Описание: Выходная звезда состоит из нейрона, управляющего группой. Она вырабатывает требуемый возбуждающий сигнал для других нейронов всякий раз, когда возбуждается. В процессе обучения нейрона выходной звезды, его веса настраиваются в соответствии с требуемым целевым вектором. Веса выходной звезды постепенно настраиваются на множество векторов, представляющие собой вариации идеального (исходного) вектора. В этом случае выходной сигнал нейронов представляет собой статистическую характеристику обучающего набора.

Области применения: Распознавание образов

Преимущества: Входная звезда хорошо моделирует некоторые функции компонент живых (биологических) нейронных сетей. Сеть, включающая в себя входные звезды, может быть достаточно хорошей моделью отдельных участков мозга. При решении практических задач входные звезды могут быть использованы для построения простых быстро обучаемых сетей.

Недостатки: Каждая звезда в отдельности реализует слишком простую функцию. Из таких звезд невозможно построить нейронную сеть, которая реализовала бы любое заданное отображение. Это ограничивает практическое применение входных звезд.

4 ТЕОРИЯ ПРОЕКТИРОВАНИЯ НЕЙРОСЕТЕЙ

Для того чтобы многослойная нейронная сеть реализовывала заданное обучающей выборкой отображение, она должна иметь достаточное число нейронов в скрытых слоях. В настоящее время нет формул для точного определения необходимого числа нейронов в сети по заданной обучающей выборке. Однако предложены способы настройки числа нейронов в процессе обучения, которые обеспечивают построение нейронной сети для решения задачи и дают возможность избежать избыточности. Эти способы настройки можно разделить на две группы: *конструктивные алгоритмы* (constructive algorithms) и *алгоритмы сокращения* (pruning algorithms).

4.1 Алгоритмы сокращения

В основе алгоритмов сокращения лежит принцип постепенного удаления из нейронной сети синапсов и нейронов. В начале работы алгоритма обучения с сокращением число нейронов в скрытых слоях сети заведомо избыточно.

Алгоритмы сокращения можно рассматривать как частный случай алгоритмов *контрастирования*. Существуют два подхода к реализации алгоритмов сокращения:

а) *метод штрафных функций*: в целевую функцию алгоритма обучения вводится штрафы за то, что значения синаптических весов отличны от нуля; пример штрафа;

б) *метод проекций*: синаптический вес обнуляется, если его значение попало в заданный диапазон:

Алгоритмы сокращения имеют, по крайней мере, два недостатка:

1. Нет методики определения числа нейронов скрытых слоев, которое является избыточным, поэтому перед началом работы алгоритма нужно угадать это число,
2. В процессе работы алгоритма сеть содержит избыточное число нейронов, поэтому обучение идет медленно.

4.2 Конструктивные алгоритмы

Предшественником конструктивных алгоритмов можно считать методику обучения многослойных сетей, включающую в себя следующие шаги:

1. Выбор начального числа нейронов в скрытых слоях,
2. Инициализация сети, то есть присваивание синаптическим весам и смещениям сети случайных значений из заданного диапазона,
3. Обучение сети по заданной выборке,
4. Завершение в случае успешного обучения; если сеть обучить не удалось, то число нейронов увеличивается, и повторяются шаги со второго по четвертый.

В конструктивных алгоритмах число нейронов в скрытых слоях также изначально мало и постепенно увеличивается. В отличие от описанной методики, в конструктивных алгоритмах сохраняются навыки, приобретенные сетью до увеличения числа нейронов.

Конструктивные алгоритмы различаются правилами задания значений параметров в новых – добавленных в сеть – нейронах:

1. Значения параметров – случайные числа из заданного диапазона,
2. Значения синаптических весов нового нейрона определяются путем *расщепления* (splitting) одного из старых нейронов.

Первое правило не требует значительных вычислений, однако его использование приводит к некоторому увеличению значения функции ошибки после каждого добавления нового нейрона. В результате случайного задания значений параметров новых нейронов может появиться избыточность в числе нейронов скрытого слоя. Расщепление нейронов лишено двух указанных недостатков.

Векторы изменений имеют два преимущественных направления и образуют в пространстве область, существенно отличающуюся от сферической. Суть алгоритма заключается в выявлении и расщеплении таких нейронов. В результате расщепления вместо одного исходного в сети оказывается два нейрона. Первый из этих нейронов имеет вектор весов, представляющий собой сумму вектора весов исходного нейрона и векторов изменений весов одного из преимущественных направлений. В результате суммирования векторов изменений весов другого преимущественного направления и вектора весов исходного нейрона получают синаптические веса второго нового нейрона.

Убирать из сети – расщеплять – нейроны, векторы изменений которых имеют два преимущественных направления, необходимо потому, что наличие таких нейронов приводит к осцилляциям при обучении классическим методом обратного распространения. При обучении методом с интегральной функцией ошибки наличие таких нейронов приводит к попаданию сети в локальный минимум с большим значением ошибки.

Алгоритм расщепления включает в себя:

1. Построение ковариационной матрицы векторов изменений синаптических весов,
2. Вычисление собственных векторов и собственных значений полученной матрицы с помощью итерационного алгоритма *Ойа (Oja)*, в соответствии с которым выполняется стохастический градиентный подъем и ортогонализация Грамма-Шмидта.

Самым большим недостатком алгоритма является экспоненциальный рост времени вычислений при увеличении размерности сети.

Для преодоления указанного недостатка предложен упрощенный алгоритм расщепления, который не требует значительных вычислений. Общее число нейронов в сети может быть несколько больше, чем у сети, построенной с помощью исходного алгоритма.

В упрощенном алгоритме для расщепления выбирается нейрон с наибольшим значением функционала.

Таким образом, в качестве критерия выбора нейрона для расщепления используется отношение суммы длин векторов изменений синаптических весов нейрона, соответствующих различным обучающим примерам, к длине суммы этих векторов.

В результате расщепления вместо исходного нейрона в сеть вводятся два новых нейрона. Значение каждого синаптического веса нового нейрона есть значение соответствующего веса старого нейрона плюс некоторый очень небольшой шум. Величины весов связей выходов новых нейронов и нейронов следующего слоя равны половине весов связей исходного нейрона с соответствующими весами следующего слоя. Упрощенный алгоритм, как и исходный, гарантирует, что функция ошибки после расщепления увеличиваться не будет.

Кроме описанных способов выбора нейронов для расщепления, может быть использован анализ чувствительности, в процессе которого строятся *матрицы Гессе* – матрицы вторых производных функции ошибки по параметрам сети. По величине модуля второй производной судят о важности значения данного параметра для решения задачи. Параметры с малыми значениями вторых производных обнуляют. Анализ чувствительности имеет большую вычислительную сложность и требует много дополнительной памяти.

4.3 Обучение нейронных сетей как задача оптимизации

С математической точки зрения обучение нейронных сетей – это многопараметрическая задача нелинейной оптимизации. В классическом методе обратного распространения ошибки (*single*-режим) обучение НС рассматривается как набор однокритериальных задач оптимизации. Критерий для каждой задачи – качество решения одного примера из обучающей выборки. На каждой итерации алгоритма обратного распространения параметры НС (синаптические веса и смещения) модифицируются так, чтобы улучшить решение одного примера.

Таким образом, в процессе обучения циклически решаются однокритериальные задачи оптимизации.

Опыт показывает, что при одиночном предъявлении примеров обучение происходит медленно и неустойчиво. Оценки за решаемые примеры то падают, то вновь растут, возможны длительные колебания, в некоторых случаях алгоритм не сходится вообще.

Если примеры предъявляются по одному, то шаг изменения параметров для улучшения решения одного примера должен быть достаточно малым. Иначе навык работы с другими примерами необратимо разрушится. Поэтому в этом случае нет смысла применять быстросходящиеся методы, дающие значительное улучшение.

Из теории оптимизации следует, что при решении многокритериальных задач модификации параметров следует производить, используя сразу несколько критериев (примеров), в идеале – все. Тем более нельзя ограничиваться одним примером при оценке производимых изменений значений параметров.

Для учета нескольких критериев при модификации параметров используют агрегированные или интегральные критерии, которые могут быть, например, суммой, взвешенной суммой или квадратным корнем от суммы квадратов оценок решения отдельных примеров.

Из априорных соображений при модификациях параметров лучше использовать сразу все примеры. Однако здесь есть ограничения:

1. При обучении на компьютере все примеры могут не умещаться в оперативной памяти.
2. Совсем не обученная сеть часто не может освоить сразу все примеры - полезно предварительное обучение на меньшем количестве наиболее простых примеров.
3. Всех примеров заранее может и не быть – «задачник» может расширяться.

Поэтому приходим к принципу *постраничного обучения*. *Страница* – серия примеров, предъявляемая сети для обучения. Модификации параметров сети осуществляется, исходя из всех примеров страницы.

4.4 Рекомендации по формированию обучающей выборки

Содержимое первой страницы. Первую страницу рекомендуется формировать из *реперных* (опорных) примеров, характеризующих особенности функции, которую должна будет реализовывать обученная нейронная сеть. Например, если сеть обучают классифицировать образы, то в первую страницу рекомендуется включить наиболее ярких представителей каждого класса.

Изменение объема страниц в процессе обучения. В ходе обучения объем страницы и разнообразие примеров на ней можно увеличить: совсем необученная сеть слишком медленно учится на больших страницах, а после «начального

образования» появляются возможности для быстрого освоения все больших страниц.

Распределение примеров по страницам. Важно, чтобы каждая страница была достаточно разнообразной и в ней присутствовали представители разных классов.

Эксперименты показывают, что постраничное обучение в задачах распознавания визуальных образов и классификации дает выигрыш не менее чем в 10-100 раз при прочих равных условиях.

4.5 Выбор числа нейронов в скрытых слоях нейронных сетей

Размерность входного сигнала и число нейронов последнего слоя в многослойных нейронных сетях определяются заданной обучающей выборкой. Определение числа нейронов в скрытых слоях представляет собой нетривиальную задачу.

Для приблизительной оценки этого числа можно воспользоваться теоремой Колмогорова-Арнольда и следствием из нее. В соответствии с этими теоретическими результатами в негомогенной двухслойной нейронной сети для реализации произвольного отображения потребуется $2*N$ нейронов в скрытом слое, где N – размерность выходного сигнала или число нейронов последнего слоя. Сеть негомогенная, нет никаких ограничений на передаточные функции нейронов. Для гомогенных сетей, в которых передаточные функции нейронов фиксированы, данная оценка является заниженной. Число $2*N$ можно рассматривать как нижнюю границу необходимого числа нейронов скрытого слоя.

Для более точной оценки числа нейронов в скрытых слоях можно воспользоваться формулой для оценки необходимого числа синаптических весов N_w в многослойной сети с сигмоидальными передаточными функциями:

$$\frac{N_y N_p}{1+\log_2(N_p)} \le N_w \le N_y\left(\frac{N_p}{N_x}+1\right)(N_x+N_y+1)+N_y$$

где N_y – размерность выходного сигнала,

N_p – число элементов обучающей выборки,

N_x – размерность входного сигнала.

Оценив необходимое число весов, можно рассчитать число нейронов в срытых слоях. Например, число нейронов в двухслойной сети составит:

$$N = \frac{N_w}{N_x + N_y}$$

Аналогично можно рассчитать число нейронов в сетях с большим числом слоев.

4.6 Масштабирование входных и выходных данных

В нейропарадигме «back propagation» на типы входных и выходных данных не накладывается никаких ограничений. Входные и выходные сигналы сети могут принадлежать как к одному, так и к разным типам данных, они могут быть двоичными, целыми или действительные (вещественными). Главное – чтобы все элементы сигналов принадлежали к одному типу. Кроме того, для успешного обучения и функционирования нейронной сети желательно, чтобы диапазоны изменений элементов входных сигналов незначительно отличались друг от друга.

Однако при решении практических задач эти требования часто не соблюдаются. Например, в задачах финансовых прогнозов входной сигнал может состоять из различных элементов: курс валюты, день недели, месяц (или квартал), индекс Доу-Джонса и т. д.

Первый элемент имеет величину порядка нескольких тысяч и в общем случае является вещественным числом, второй элемент – целое число от 1 до 7 и т. д.

Для того чтобы представить все элементы входного сигнала числами одного типа из одного диапазона, используется операция масштабирования.

Выбирается диапазон изменения элементов входного сигнала («общий диапазон»). Для каждого элемента входного сигнала определяется его диапазон и выполняется линейное преобразование данного элемента таким образом, чтобы в результате его значения принадлежали общему диапазону.

Еще одна проблема обучения и функционирования нейронных сетей состоит в следующем. В некоторых задачах на значение выходных сигналов сети существенно влияют не абсолютные значения входных сигналов, а небольшие изменения во входных сигналах. Например, небольшие колебания курса валюты могут существенно повлиять на финансовый прогноз. Классический вариант нейропарадигмы «back propagation» позволяет учитывать главным образом абсолютные значения входных сигналов, а не небольшие колебания. Это затрудняет решение некоторых практических задач, в частности, прогнозирования. Для разрешения этой проблемы также можно использовать масштабирование.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Барцев С. И., Гилев С. Е., Охонин В. А. Принцип двойственности в организации адаптивных сетей обработки информации // Динамика химических и биологических систем. Новосибирск: Наука, 1989, С. 6 – 55.
2. Барцев С. И., Охонин В. А. Адаптивные сети обработки информации. Красноярск: Ин-т физики СО АН СССР, 1986. Препринт № 59Б. – 20 с.
3. Барцев С. И. Некоторые свойства адаптивных сетей (Программная реализация). Красноярск: Ин-т физики СО АН СССР, 1987. Препринт № 71Б. – 17 с.
4. Биотехника – новое направление компьютеризации/ Ю. К. Ахапкин, С.И.Барцев, Н. Н. Всеволодов и др. – М.: Наука, 1990. – 144 с.
5. Д.-Э. Бэстенс, В. М. Ван Ден Берг, Д. Вуд «Нейронные сети и финансовые рынки» М.: ТВП «Научное издательство», 1997.– 236 с.
6. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. – М.: Мир, 1985. – 509 с.
7. Горбань А. Н. Обучение нейронных сетей. М.: СП ПараГраф. 1991.
8. Евтихиев Н. Н., Оныкий Б. Н., Перепелица В.В., Щербаков И.Б. Математические модели и оптические реализации многослойных и полиномиальных нейронных сетей. М.: Препринт/МИФИ, 004-94, 1994. - 32 с.
9. Евтихиев Н.Н., Оныкий Б.Н., Перепелица В.В., Щербаков И.Б. Многослойная нейронная сеть и ее реализация на основе оптического вектор-матричного перемножителя // Нейрокомпьютер, No. 1-2, 1994.
10. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. М.: Мир. 1992.
11. М. Холодниок и др. «Методы анализа нелинейных динамических моделей» М.: Высшая школа, 1994.-256 с.
12. Шеметов Д.В., Осипов Ю.М. Определение показателя "значимость технического решения" аппаратом нейронных сетей./ Нейроинформатика и ее приложения: Тезисы доклада VI Всероссийского семинара, 2-5 октября 1998г. Крсноярск, 1998. – С. 190-191.
13. Г. Шустер «Детерминированный хаос. Введение» - М.: Мир, 1988.- 240 с.
14. В.Л. Яковлев, Г.Л. Яковлева, Л.А. Лисицкий «Применение нейросетевых алгоритмов к анализу финансовых рынков» // «Информационные технологии» № 6, 1999. С.37-48.
15. В. Л. Яковлев, Г. Л. Яковлева, Д. А. Малиевский Нейросетевая экспертная система управления портфелем банка // V Всероссийская конференция «Нейрокомпьютеры и их применение». Сборник докладов - 1999, С.291-294.