

Московский государственный технический университет
им.Н.Э.Баумана
Калужский филиал
Приборостроительный факультет
Кафедра П2-КФ

Отчет-реферат по НИРС

на тему:

*“ Искусственные нейронные сети, их
разновидности и методы обучения”*

Студент : Комиссаров А.В.

Руководитель : Максимов А.В.

Калуга, 1998

Содержание

1. Формализация нейронных сетей
2. Постановка задачи и способы обучения формальных нейросетей
3. Формирование отображений при обучении в нейронных сетях
 - 3.1. Теорема Колмогорова-Арнольда.
 - 3.2. Работа Хехт-Нильсена.
 - 3.3. Следствия из теоремы Колмогорова-Арнольда-Хехт-Нильсена
 - 3.4. Использование других моделей для представления отображений
4. Настройка числа нейронов в скрытых слоях многослойных сетей в процессе обучения
 - 4.1. Алгоритмы сокращения
 - 4.2. Конструктивные алгоритмы
5. Алгоритм с настройкой передаточных функций для обучения нейронных сетей
 - 5.1. Алгоритм с настройкой только синаптических весов и смещений.
 - 5.2. Настройка передаточных функций
6. Парадигма «Back Propagation» («обратного распространения ошибки»)
 - 6.1. Функционирование многослойной сети.
 - 6.2. Обучение сети.
 - 6.3. Модификация алгоритма.
 - 6.4. Гетероассоциативная память.
 - 6.5. Прогнозирование.
 - 6.6. Автоассоциативная память.
7. Сети Кохонена (Kohonen's neural network)

7.1. Авторы и история создания.

7.2. Модель.

7.2.1. Алгоритм Кохонена формирования карт признаков

7.3. Области применения.

7.4. Недостатки

7.5. Преимущества

7.6. Модификации

Список литературы

1. Формализация нейронных сетей.

Нейрон (рис. 1) - это составная часть нейронной сети. Он состоит из элементов трех типов. Элементы нейрона - умножители (синапсы), сумматор и нелинейный преобразователь. Синапсы осуществляют связь между нейронами, умножают входной сигнал на число, характеризующее силу связи, - вес синапса. Сумматор выполняет сложение сигналов, поступающих по синаптическим связям от других нейронов, и внешних входных сигналов. Нелинейный преобразователь реализует нелинейную функцию одного аргумента - выхода сумматора. Эта функция называется "функция активации" или "передаточная функция" нейрона. Нейрон в целом реализует скалярную функцию векторного аргумента.

Математическая модель нейрона:

$$S = \sum_{i=1}^N w_i x_i + b \quad (1),$$

$$y = f(s) \quad (2),$$

где w_i - вес синапса (weight), ($i=1,2,\dots,N$); b - значение смещения (bias); s - результат суммирования (sum); x_i - компонента входного вектора (входной сигнал), ($i=1,2,\dots,N$); y - выходной сигнал нейрона; N - число входов нейрона; f - нелинейное преобразование (функция активации). В общем случае входной сигнал, весовые коэффициенты и значения смещения могут принимать действительные значения. Выход (y) определяется видом функции активации и может быть как действительным, так и целым. Во многих практических задачах входы, веса и смещения могут принимать лишь некоторые фиксированные значения. Синаптические связи с положительными весами называют возбуждающими, с отрицательными весами - тормозящими. Описанный вычислительный элемент можно считать упрощенной математической моделью биологических нейронов - клеток, из которых состоит нервная система человека и животных. Чтобы подчеркнуть различие нейронов биологических и математических, вторые иногда называют нейроноподобными элементами, или формальными нейронами. На входной сигнал (s) нелинейный преобразователь отвечает выходным сигналом $f(s,p)$, который представляет из себя выход нейрона (y). Здесь p - параметр или набор параметров, от которых зависит функционирование преобразователя. Пример передаточной функции представлен на рис. 2.

Нейронной сетью будем называть структуру, состоящую из связанных между собой нейронов. Нейронные сети могут иметь различные архитектуры. Можно выделить три основных типа нейронных сетей: полносвязные сети (рис. 3-а), многослойные сети (рис. 3-б), слабосвязные сети (нейронные сети с локальными связями) (рис. 3-в). В слабосвязных сетях нейроны располагаются в узлах прямоугольной решетки. Каждый нейрон связан с четырьмя или восемью своими ближайшими соседями. В полносвязной сети каждый нейрон связан со всеми остальными (на входы каждого нейрона подаются выходные сигналы остальных нейронов). В многослойных сетях нейроны объединяются в слои.

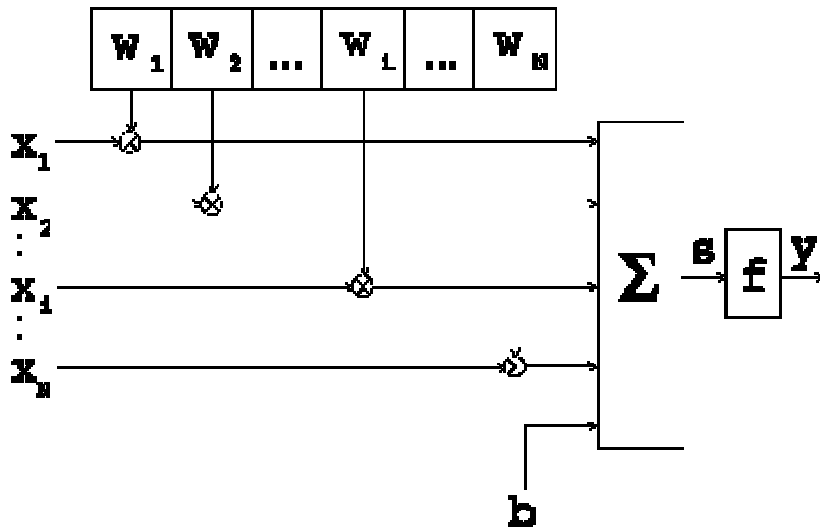


Рис. 1. Схема формального нейрона

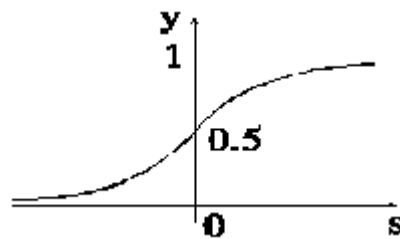


Рис. 2. Сигмоидальная передаточная функция

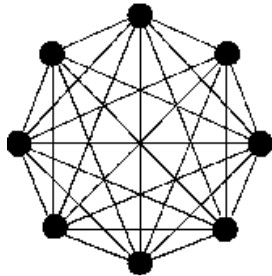


Рис. 3-а.

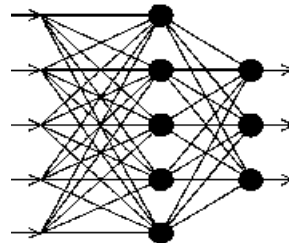


Рис.3-б.

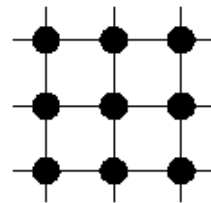


Рис.3-в.

Слой - это совокупность нейронов с единым входным сигналом. Внешние входные сигналы подаются на входы нейронов первого слоя, а выходами сети являются выходные сигналы последнего слоя. Кроме входного и выходного слоев, в многослойной нейронной сети есть один или несколько промежуточных (скрытых) слоев. Вход нейронной сети можно рассматривать как выход "нулевого слоя" вырожденных нейронов. Связи от выходов нейронов некоторого слоя m ко входам нейронов следующего слоя $(m+1)$ называются *последовательными*. Если нейроны каждого слоя сети имеют единую функцию активации, то такую нейронную сеть будем называть *однородной*.

2. Постановка задачи и способы обучения формальных нейросетей

В процессе функционирования нейронная сеть формирует выходной сигнал Y в соответствии со входным сигналом X , реализуя некоторую функцию g : $Y=g(X)$. Если архитектура сети задана, то вид функции g определяется значениями синаптических весов и смещений сети. Обозначим буквой G множество всех возможных функций g , соответствующих заданной архитектуре сети. Пусть решение некоторой задачи - функция r : $Y=r(X)$, заданная парами входных-выходных данных $(X_1, Y_1), (X_2, Y_2), \dots (X_M, Y_M)$, для которых $Y_m=r(X_m)$ ($m=1, 2, \dots, M$). D - функция ошибки, показывающая для каждой из функций g степень близости к r . Решить поставленную задачу с помощью нейронной сети заданной архитектуры - это значит построить (синтезировать) функцию, подобрав параметры нейронов (синаптические веса и смещения) таким образом, чтобы функционал качества обращался в оптимум для всех пар (X_m, Y_m) . Задача обучения определяется совокупностью пяти элементов: $\langle X, Y, r, G, D \rangle$, где X и Y - вход и выход соответственно; r - функция - определяет желаемый результат обучения; в задаче обучения по примерам функция r задается парами входных-выходных данных : $(X_1, Y_1), (X_2, Y_2), \dots (X_M, Y_M)$, для которых $Y_m=r(X_m)$ ($m=1, 2, \dots, M$); архитектура связей нейронной сети считается заданной до начала обучения, она определяет множество функций G ; D - функция ошибки, показывающая для каждой функции степень близости к r ; обучение состоит в поиске (синтезе) функции g , оптимальной по D .

Обучение - это итерационная процедура. На каждой итерации происходит уменьшение значения функции ошибки. Обучение требует длительных вычислений. Если выбраны множество обучающих примеров - пар (X_m, Y_m) ($m=1, 2, \dots, M$) - и способ вычисления функции ошибки D , обучение нейронной сети превращается в задачу многомерной оптимизации. Размерность задачи - от нескольких тысяч до 108. Функция D может иметь произвольный вид. Поэтому обучение в общем случае - многоэкстремальная невыпуклая задача оптимизации. Для решения этой задачи могут быть использованы следующие алгоритмы:

- 1. Алгоритмы локальной оптимизации с вычислением частных производных первого порядка.** К ним относятся : градиентный алгоритм (метод скорейшего спуска), методы с одномерной и двумерной оптимизацией целевой функции в направлении антиградиента, метод сопряженных градиентов; методы, учитывающие направление антиградиента на нескольких шагах алгоритма.
- 2. Алгоритмы локальной оптимизации с вычислением частных производных первого и второго порядка.** Ко второй группе относятся метод Ньютона, методы оптимизации с разреженными матрицами Гессе, квазиньютоновские методы, метод Гаусса-Ньютона, метод Левенберга-Маркардта.

3. Стохастические алгоритмы оптимизации. Стохастическими алгоритмами являются поиск в случайном направлении, имитация отжига, метод Монте-Карло (численный метод статистических испытаний).

4. Алгоритмы глобальной оптимизации. Задачи глобальной оптимизации решаются с помощью перебора значений переменных, от которых зависит целевая функция.

Для сравнения методов обучения нейронных сетей необходимо использовать два критерия:

1. Количество шагов алгоритма, которые необходимо выполнить для получения решения;
2. Количество дополнительных переменных, которые потребуются для организации вычислительного процесса.

Для иллюстрации термина "дополнительные переменные" приведем следующий пример. Пусть p_1 и p_2 - некоторые параметры нейронной сети с заданными значениями. В процессе обучения сети по некоторому алгоритму на каждом шаге по меньшей мере два раза потребуется выполнить умножение $p_1 * p_2$. Для того, чтобы выполнять указанное умножение только один раз и не тратить время на повторное умножение, используется дополнительная переменная, в которой сохраняется значение произведения после первого умножения.

Пусть нейронная сеть содержит P изменяемых параметров (синаптических весов и смещений). Существует лишь две группы алгоритмов обучения, которые требуют менее $2 * P$ дополнительных параметров и при этом дают возможность обучать нейронные сети за приемлемое число шагов. Это алгоритмы с вычислением частных производных первого порядка и (возможно) одномерной оптимизацией. Именно эти алгоритмы и будут рассмотрены в следующих подразделах. Хотя указанные алгоритмы дают возможность находить только локальные экстремумы, они могут быть использованы на практике для обучения нейронных сетей с многоэкстремальными целевыми функциями (функциями ошибки). Дело в том, что экстремумов у целевой функции, как правило не очень много. Достаточно лишь раз или два "выбить" сеть из локального минимума с большим значением целевой функции для того, чтобы в результате итераций в соответствии с алгоритмом локальной оптимизации сеть оказалась в локальном минимуме со значением целевой функции, близким к нулю. Если после нескольких попыток "выбить" сеть из локального минимума желаемого результата добиться не удастся, необходимо увеличить число нейронов во всех слоях с первого по предпоследний. (Для того, чтобы "выбить" сеть из локального минимума, синаптическим весам и смещениям случайным образом присваиваются значения из заданного диапазона.) Многочисленные эксперименты по обучению нейронных сетей показали, что совокупное использование алгоритма локальной оптимизации, процедуры

"выбивания" сети из локального минимума и процедуры увеличения числа нейронов приводит к успешному обучению нейронных сетей.

Кратко опишем недостатки других алгоритмов. Стохастические алгоритмы требуют очень большого числа шагов обучения. Это делает невозможным их практическое использование для обучения нейронных сетей больших размерностей. Экспоненциальный рост сложности перебора с ростом размерности задачи в алгоритмах глобальной оптимизации при отсутствии априорной информации о характере целевой функции (функции ошибки) также делает невозможным их использование для обучения нейронных сетей больших размерностей. Метод сопряженных градиентов очень чувствителен к точности вычислений, особенно при решении задач оптимизации большой размерности. Методы, учитывающие направление антиградиента на нескольких шагах алгоритма, и методы, включающие в себя вычисление матрицы Гессе, требуют дополнительных переменных более, чем $2 \cdot P$. В зависимости от способа разрежения, вычисление матрицы Гессе требует от $2 \cdot P$ до P^2 дополнительных переменных.

3. Формирование отображений при обучении в нейронных сетях

Многие задачи: распознавания образов (зрительных, речевых и т.д.), выполнения функциональных преобразований при обработке сигналов, управления, прогнозирования, идентификации сложных систем и т.д. - сводятся к следующей математической постановке. Необходимо построить отображение такое, чтобы на каждый возможный входной сигнал X формировался правильный выходной сигнал Y . Отображение задается конечным набором пар (<вход>, <известный выход>). Число таких пар (обучающих примеров) существенно меньше общего числа возможных сочетаний значений входных и выходных сигналов. Совокупность всех обучающих примеров носит название *обучающей выборки*. В задачах распознавания образов X - некоторое представление образа (изображение, вектор чисел и т.д.), Y - номер класса, к которому принадлежит входной образ. В задачах управления X - набор контролируемых параметров управляемого объекта, Y - код, определяющий управляющее воздействие, соответствующее текущим значениям контролируемых параметров. В задачах прогнозирования в качестве входных сигналов используются временные ряды, представляющие значения контролируемых переменных на некотором интервале времени. Выходной сигнал - множество переменных, которое является подмножеством переменных входного сигнала. При идентификации X и Y представляют входные и выходные сигналы системы соответственно. Вообще говоря, большая часть прикладных задач может быть сведена к реализации сложного многомерного функционального преобразования. В результате построения отображения необходимо добиться того, чтобы:

1. обеспечивалось формирование правильных выходных сигналов в соответствии со всеми примерами обучающей выборки.
2. обеспечивалось формирование правильных выходных сигналов в соответствии со всеми возможными входными сигналами, которые не вошли в обучающую выборку.

Второе требование в значительной степени усложняет задачу формирования обучающей выборки. Эта задача в общем виде в настоящее время не решена. Однако во всех известных случаях было найдено частное решение. Дальнейшие рассуждения предполагают, что обучающая выборка уже сформирована.

3.1. Теорема Колмогорова-Арнольда

Отображение $g: X \rightarrow Y$ в общем случае является многомерным, то есть X и Y - вектора с размерностью больше единицы. Построить отображение - это значит представить его с помощью традиционных математических операций, которые, в частности, легко могут быть реализованы на персональном компьютере. Примеры таких операций: сложение, умножение, деление, SIN, COS и т.д. Эти операции имеют не более двух переменных (аргументов).

Проблема представления функций многих переменных в виде суперпозиции функций меньшего числа переменных восходит еще к так называемой 13-й проблеме Гильберта. Этот математик предполагал, что представить функцию многих переменных в виде суперпозиции функций меньшего числа переменных невозможно.

В результате многолетней научной полемики между А.Н. Колмогоровым и В.И. Арнольдом был получен целый ряд важных теоретических результатов, опровергающих тезис Гильберта:

1. Теорема о возможности представления непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных. (1956 г.) [1].
2. Теорема о представлении любой непрерывной функции трех переменных в виде суммы функций не более двух переменных (1957 г.) [5].
3. Теорема о представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения (1957 г.) [2,3].

3.2. Работа Хехт-Нильсена.

Последний, наиболее сильный результат в 1987 году был переложен Хехт-Нильсенем в термины теории нейронных сетей [4].

Теорема Хехт-Нильсена доказывает представимость функции многих переменных достаточно общего вида $R^k \rightarrow R^m$ с помощью *двухслойной* нейронной сети с прямыми полными связями (рис. 3-б) с N компонентами входного сигнала, 2N+1 компонентами первого ("скрытого" слоя) с заранее известными ограниченными функциями активации (например, сигмоидальными) и M компонентами второго слоя с *неизвестными* функциями активации. Теорема, таким образом, в неконструктивной форме доказывает решаемость задачи представления функции достаточно произвольного вида на НС и указывает для каждой задачи минимальные значения числа нейронов сети, необходимых для решения.

3.3. Следствия из теоремы Колмогорова-Арнольда-Хехт-Нильсена

Следствие 1. Из теоремы в формулировке Хехт-Нильсена следует представимость любой многомерной функции нескольких переменных с помощью нейронной сети фиксированной размерности. Неизвестными остаются следующие характеристики функций активации нейронов:

1. Ограничения области значений (т.е. координаты асимптот) сигмоидальных функций активации нейронов "скрытого" слоя.
2. Наклон сигмоидальных функций активации.
3. Вид функций активации нейронов второго слоя.

Про функции активации нейронов второго слоя из теоремы Хехт-Нильсена известно только то, что они - нелинейные функции общего вида. В одной из работ, продолжающих развитие теории, связанной с рассматриваемой теоремой, доказывается, что функции нейронов второго слоя должны быть монотонно возрастающими. Это утверждение в некоторой степени сужает класс функций, которые могут потребоваться при реализации отображения с помощью двухслойной нейронной сети.

На практике требования теоремы Хехт-Нильсена к функциям активации удовлетворяют следующим образом. В нейронных сетях как для первого, так и для второго слоя используют сигмоидальные передаточные функции с настраиваемыми параметрами. То есть, в процессе обучения индивидуально для каждого нейрона настраиваются следующие параметры функций активации:

1. максимальное и минимальное значение функции,
2. наклон сигмоидальной функции.

Следствие 2. В монографии [6] показано, что для любого множества пар (X^k, y^k) (где y^k - скаляр) существует двухслойная однородная (с одинаковыми функциями активации) нейронная сеть первого порядка с последовательными связями и с конечным числом нейронов, которая

выполняет отображение $X \rightarrow Y$, выдавая на каждый входной сигнал X^k правильный выходной сигнал y^k . Нейроны в такой двухслойной нейронной сети должны иметь сигмоидальные передаточные функции.

К сожалению, эта теорема не конструктивна. В ней не заложена методика определения числа нейронов в сети для некоторой конкретной обучающей выборки.

Для многих задач не достаточно единичной размерности выходного сигнала. Необходимо иметь возможность строить с помощью нейронных сетей функции, где Y имеет произвольную размерность. Следующее утверждение является теоретической основой для построения таких функций на базе однородных нейронных сетей.

Утверждение. Для любого множества пар входных-выходных векторов произвольной размерности $\{(X^k, Y^k), k=1...K\}$ существует двухслойная однородная нейронная сеть с последовательными связями, с сигмоидальными передаточными функциями и с конечным числом нейронов, которая для каждого входного вектора X^k формирует соответствующий ему выходной вектор Y^k .

Доказательство.

Пусть выходные векторы Y^k имеют размерность N , то есть $Y^k=(y^k_1, y^k_2, \dots, y^k_i, \dots, y^k_N)$. В соответствии со следствием из теоремы Колмогорова-Арнольда для любого множества пар $\{(X^k, y^k_i), k=1...K\}$ существует двухслойная однородная нейронная сеть с последовательными связями и с сигмоидальными передаточными функциями, которая для каждого входного вектора X^k формирует соответствующий ему выходной скаляр y^k_i . Такая нейронная сеть существует для каждого номера элемента выходного вектора $i=1..N$. Во втором слое эти сети содержат по одному нейрону. Пусть в первом слое сети, соответствующей элементу выходного сигнала с номером i содержится M_i нейронов.

Построим двухслойную однородную нейронную сеть с последовательными связями и с сигмоидальными передаточными функциями, содержащую в первом слое $(M_1+M_2+\dots+M_i+\dots+M_N)$ нейронов. Значения весов и смещений нейронов первого слоя равны соответствующим значениям нейронов сетей, построенных для пар (X^k, y^k_i) . Связи в этой сети между нейронами первого и второго слоев определим следующим образом. Значения синаптических весов первых M_1 связей и значение смещения первого нейрона второго слоя совпадают с соответствующими значениями сети, построенной для пар (X^k, y^k_1) . Остальные синаптические веса первого нейрона - нулевые. Значения синаптических весов связей с номерами $(M_1+1) \dots M_2$ и значение смещения второго нейрона второго слоя совпадают с соответствующими значениями сети, построенной для пар (X^k, y^k_2) . Значения остальных синаптических весов второго нейрона равны нулю. Аналогично строятся синаптические веса и смещения других нейронов второго слоя. В результате такого построения получаем двухслойную

однородную нейронную сеть с последовательными связями, с сигмоидальными передаточными функциями и с конечным числом нейронов, которая для каждого входного вектора X^k формирует соответствующий ему выходной вектор Y^k . Это доказывает утверждение. •

Таким образом, для представления многомерных функций многих переменных может быть использована двухслойная однородная нейронная сеть с сигмоидальными передаточными функциями. Для оценки числа нейронов с скрытых слоев однородных нейронных сетей можно воспользоваться формулой для оценки необходимого числа синаптических весов N_w в многослойной сети с сигмоидальными передаточными функциями [7]:

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

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 скрытых 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.

3.4. Использование других моделей нейронных сетей для представления отображений

Иногда целесообразно использовать сети с большим числом слоев (см. рис. 4). Такие многослойные нейронные сети могут иметь меньшие размерности матриц синаптических весов нейронов одного слоя, чем двухслойные сети, реализующие то же самое отображение. К сожалению, строгая методика построения таких сетей пока отсутствует. Аналогичная ситуация и с многослойными сетями, в которых помимо последовательных связей используются и прямые (то есть связи от слоя с номером m к слою с номером $m+s$, где $s > 1$). Нет строгой теории, которая показывала бы возможность и целесообразность построения таких сетей. Наибольшие проблемы возникают при использовании так называемых сетей циклического функционирования. К этой группе нейросетевых моделей относятся многослойные сети с обратными связями (от слоя с номером m к слою с номером $m+s$, где $s < 0$), а также полносвязные сети. Для успешного функционирования этих моделей необходимо соблюдение условий динамической устойчивости. В противном случае при функционировании сеть может не сойтись к правильному решению, либо достигнув на некоторой итерации правильного значения выходного сигнала, после нескольких итераций уйти от этого значения. Проблема динамической устойчивости подробно исследована, пожалуй, лишь для одной модели из рассматриваемой группы - нейронной сети Хопфилда. Отсутствие строгой теории, связанной с перечисленными моделями нейронных сетей, не

препятствует исследованию возможностей их применения для решения практических задач.

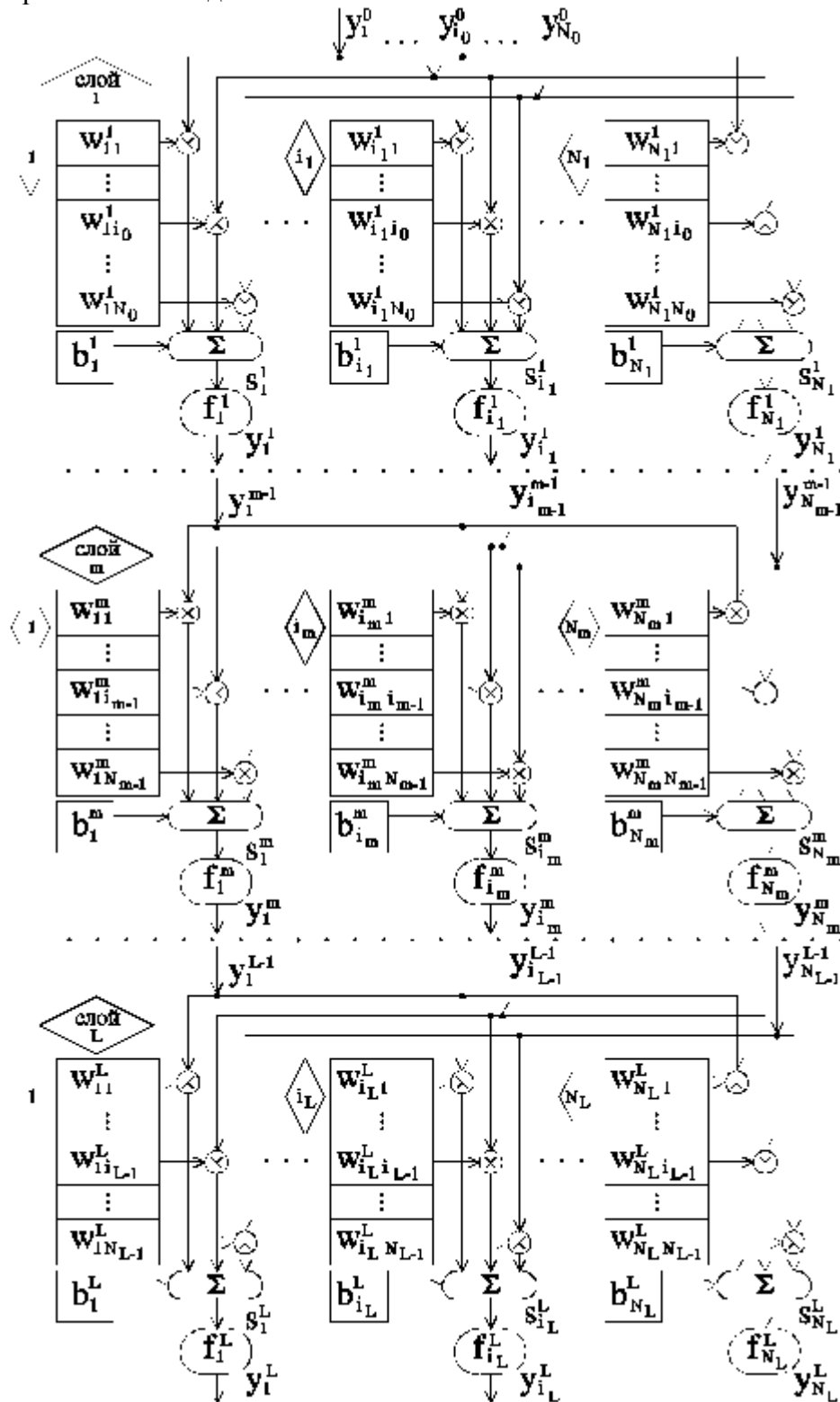


Рис. 4. Многослойная сеть с последовательными связями

4. Настройка числа нейронов в скрытых слоях многослойных сетей в процессе обучения

В настоящее время 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 [8]. Существуют два подхода к реализации алгоритмов сокращения:

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

$$C = \sum_{i=1}^N \sum_{j=1}^M w_{ij}^2 \quad (1)$$

где w - синаптический вес, i - номер нейрона, j - номер входа, N - число нейронов, M - размерность входного сигнала нейронов;

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

$$w_{ij} = \begin{cases} 0, & -\epsilon \leq w_{ij} \leq \epsilon \\ w_{ij}, & w_{ij} < -\epsilon, w_{ij} > \epsilon, \end{cases} \quad (2),$$

где ϵ - некоторая константа.

Алгоритмы сокращения имеют, по крайней мере, два недостатка:

- нет методики определения числа нейронов скрытых слоев, которое является избыточным, поэтому перед началом работы алгоритма нужно угадать это число;
- 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.

Суть алгоритма расщепления проиллюстрирована на рис. 1.

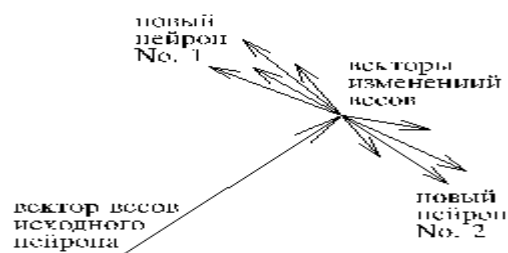


Рис. 1. Вектор весов нейрона скрытого слоя и изменения, соответствующие отдельным обучающим примерам

На рисунке 1 показан вектор весов нейрона скрытого слоя на некотором шаге обучения и векторы изменения весов, соответствующие отдельным обучающим примерам. 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 из себя 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.

Полное описание алгоритма расщепления представлено в [9]. Он включает в себя построение ковариационной матрицы векторов изменений синаптических весов, 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 автором отчета предлагается упрощенный алгоритм расщепления, который не требует значительных вычислений. Общее число нейронов в сети, построенной с помощью предлагаемого алгоритма по заданной обучающей выборке, может быть несколько больше, чем у сети, построенной с помощью исходного алгоритма.

В упрощенном алгоритме для расщепления выбирается нейрон с наибольшим значением функционала

$$F_i = \frac{\sum_{e=1}^P |\delta w_i^e|}{\left| \sum_{e=1}^P \delta w_i^e \right|} \quad (3),$$

где δw - вектор изменений синаптических весов нейрона, i - номер нейрона, $i=1,2,\dots,N$, N - число нейронов, e - номер обучающего примера, P - число примеров в обучающей выборке, $|\bullet|$ - длина вектора.

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 [8,9], 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.

5. Алгоритм с настройкой передаточных функций для обучения нейронных сетей

5.1. АЛГОРИТМ С НАСТРОЙКОЙ ТОЛЬКО СИНАПТИЧЕСКИХ ВЕСОВ И СМЕЩЕНИЙ

Нейронной сети в ходе обучения предъявляются примеры и их решение оценивается. По результатам оценки делается шаг, меняются параметры. Обучающий пример - пара <вход; известный выход> - имеет вид:

$$\langle \sigma_1^e, \sigma_2^e, \dots, \sigma_{N_0}^e; \zeta_1^e, \zeta_2^e, \dots, \zeta_{N_L}^e \rangle$$

где $\sigma_{i_0}^e$ и $\zeta_{i_L}^e$ - элемент входного и выходного сигнала, $i_0=1,2,\dots,N_0$, $i_L=1,2,\dots,N_L$, L - число слоев, e - номер обучающего примера в выборке, $e=1,2,\dots,P$, P - число обучающих примеров в выборке. На каждый входной сигнал сеть должна выдать соответствующий выходной сигнал.

Если в необученную многослойную нейронную сеть с последовательными связями ввести входной сигнал одного из примеров обучающей выборки:

$$y_{i_0}^e = \sigma_{i_0}^e, i_0 = 1, 2, \dots, N_0; \quad (1)$$

$$d_{i_m}^e = \sum_{i_{m-1}=1}^{N_{m-1}} w_{i_m i_{m-1}} y_{i_{m-1}}^e - b_{i_m}, i_m = 1, 2, \dots, N_m, m = 1, 2, \dots, L; \quad (2)$$

$$y_{i_m}^e = f(s_{i_m}^e), i_m = 1, 2, \dots, N_m, m = 1, 2, \dots, L, \quad (3),$$

то выходные сигналы будут отличаться от требуемых, которые определены в обучающем примере.

В формулах: y - выходной сигнал нейрона (внешний входной сигнал считается выходом вырожденных нейронов нулевого слоя), w - синаптический вес, s - выход сумматора, b - смещение, f - передаточная функция, i - номер нейрона, m - номер слоя.

Функция ошибки определяет степень близости выходных сигналов к требуемым при решении всей совокупности примеров обучающей выборки:

$$D = \frac{1}{2} \sum_{e=1}^P \sum_{i_L=1}^{N_L} (\zeta_{i_L}^e - y_{i_L}^e)^2. \quad (4)$$

На каждой итерации вычисляется градиент функции ошибки в адаптивном пространстве сети и делается шаг по адаптивному рельефу в направлении, противоположном градиенту. Размерность адаптивного пространства сети на единицу больше числа параметров, настраиваемых в процессе обучения:

$$\dim = \sum_{m=1}^L N_m N_{m-1} + \sum_{m=1}^L N_m + 1. \quad (5)$$

По осям координат в адаптивном пространстве откладываются значения функции ошибки, синаптических весов и смещений сети.

Адаптивный рельеф - это поверхность в адаптивном пространстве, каждая точка которой имеет вид :

$$(\|w_{i_1 i_0}\|, \|b_{i_1}\|, \|w_{i_2 i_1}\|, \|b_{i_2}\|, \dots, \|w_{i_m i_{m-1}}\|, \|b_{i_m}\|, \dots, \|w_{i_L i_{L-1}}\|, \|b_{i_L}\|, D),$$

где $\|w_{i_m i_{m-1}}\|$ и $\|b_{i_m}\|$ - матрица весовых коэффициентов и столбец m -го слоя.

Задача минимизации функции ошибки сводится к поиску минимума на адаптивном рельефе. Итерации алгоритма могут быть описаны следующими формулами:

$$w_{i_m i_{m-1}}^m(t) = w_{i_m i_{m-1}}^m(t-1) + \varepsilon \sum_{e=1}^P \Delta_{i_m}^{m,e}(t-1) y_{i_{m-1}}^{m-1,e}(t-1) \\ i_m = 1, 2, \dots, N_m, i_{m-1} = 1, 2, \dots, N_{m-1}, m = L, L-1, \dots, 2; \quad (6)$$

$$i_1 = 1, 2, \dots, N_1, i_0 = 1, 2, \dots, N_0; \quad (7)$$

$$\Delta_{i_L}^{L,e} = [\zeta_{i_L}^e - f_{i_L}^L(s_{i_L}^{L,e})] (f_{i_L}^L(s_{i_L}^{L,e}))', i_L = 1, 2, \dots, N_L; \quad (8)$$

$$\Delta_{i_m}^{m,\varepsilon} = \sum_{i_{m-1}=1}^{N_{m-1}} \Delta_{i_{m-1}}^{m+1,\varepsilon} W_{i_{m-1},i_m}^{m+1} \left(f_{i_m}^{m,\varepsilon}(s_{i_m}^{m,\varepsilon}) \right)'$$

$$i_m = 1, 2, \dots, N_m, m = L-1, L-2, \dots, 1; \quad (9)$$

$$b_{i_m}^{m,\varepsilon}(t) = b_{i_m}^m(t-1) - \varepsilon \sum_{\tau=1}^p \Delta_{i_m}^{m,\varepsilon}(t-1)$$

$$i_m = 1, 2, \dots, N_m, m = L, L-1, \dots, 1. \quad (10)$$

В этих формулах t - номер итерации алгоритма обучения. В формулах (8) и (9) определены двойственные переменные.

5.2. НАСТРОЙКА ПЕРЕДАТОЧНЫХ ФУНКЦИЙ

Рассмотрим настройку однопараметрической передаточной функции типа рациональная сигмоида:

$$f_{i_m}^{m,\varepsilon}(s_{i_m}^{m,\varepsilon}) = \frac{s_{i_m}^{m,\varepsilon}}{|s_{i_m}^{m,\varepsilon}| + \alpha_{i_m}^{m,\varepsilon}} \quad (11)$$

Для настройки параметров передаточных функций на каждой итерации обучения выполняются следующие вычисления:

$$\alpha_{i_m}^{m,\varepsilon}(t) = \alpha_{i_m}^{m,\varepsilon}(t-1) + \varepsilon \sum_{\tau=1}^p \bar{\Delta}_{i_m}^{m,\varepsilon}(t-1)$$

$$i_m = 1, 2, \dots, N_m, m = L, L-1, \dots, 1 \quad (12)$$

$$\bar{\Delta}_{i_L}^{L,\varepsilon} = \left[\zeta_{i_L}^\varepsilon - f_{i_L}^L(s_{i_L}^{L,\varepsilon}) \right] \left(f_{i_L}^L(s_{i_L}^{L,\varepsilon}) \right)'_{\alpha_{i_L}^{L,\varepsilon}}, i_L = 1, 2, \dots, N_L; \quad (13)$$

$$\bar{\Delta}_{i_m}^{m,\varepsilon} = \sum_{i_{m-1}=1}^{N_{m-1}} \bar{\Delta}_{i_{m-1}}^{m+1,\varepsilon} W_{i_{m-1},i_m}^{m+1} \left(f_{i_m}^{m,\varepsilon}(s_{i_m}^{m,\varepsilon}) \right)'_{\alpha_{i_m}^{m,\varepsilon}};$$

$$i_m = 1, 2, \dots, N_m, m = L-1, L-2, \dots, 1; \quad (14)$$

где

$$\left(f_{i_m}^{m,\varepsilon}(s_{i_m}^{m,\varepsilon}) \right)'_{\alpha_{i_m}^{m,\varepsilon}} = - \frac{s_{i_m}^{m,\varepsilon}}{\left(|s_{i_m}^{m,\varepsilon}| + \alpha_{i_m}^{m,\varepsilon} \right)^2},$$

$$i_m = 1, 2, \dots, N_m, m = 1, 2, \dots, L; \quad (15)$$

В процессе обучения значение параметра α не должно выходить за пределы допустимого диапазона: $0.1 \leq \alpha \leq 2$.

6. Парадигма «Back Propagation» («обратного распространения ошибки»).

История многослойных нейронных сетей началась в 1960-х годах и связана с работами Розенблатта, Минского, Пейперта и др. Лишь в середине 1980-х несколькими исследователями независимо друг от друга был предложен эффективный алгоритм обучения многослойных персептронов, основанный на вычислении градиента функции ошибки. Алгоритм был назван "обратным распространением ошибки".

Алгоритм обратного распространения - это итеративный градиентный алгоритм обучения, который используется с целью минимизации среднеквадратичного отклонения текущего выхода и желаемого выхода многослойных нейронных сетей.

В нейропарадигме "back propagation" чаще всего используются сигмоидальные передаточные функции, например

$$f(s) = \frac{s}{|s| + \alpha}$$

Сигмоидальные функции являются монотонно возрастающими и имеют отличные от нуля производные на всей области определения. Эти характеристики обеспечивают правильное функционирование и обучение сети.

6.1. Функционирование многослойной сети выполняется в соответствии с формулами:

$$s_{i_m} = \sum_{i_{m-1}=1}^{N_{m-1}} w_{i_m i_{m-1}} y_{i_{m-1}} - b_{i_m}, i_m = 1, 2, \dots, N_m, m = 1, 2, \dots, L$$
$$y_{i_m} = f(s_{i_m}), i_m = 1, 2, \dots, N_m, m = 1, 2, \dots, L$$

где s - выход сумматора, w - вес связи, y - выход нейрона, b - смещение, i - номер нейрона, N - число нейронов в слое, m - номер слоя, L - число слоев, f - функция активации.

6.2. Обучение сети разбивается на следующие этапы:

- 1) Инициализация сети: Весовым коэффициентам и смещениям сети присваиваются малые случайные значения из диапазонов (w_{\min}, w_{\max}) и (w_{\min}, w_{\max}) соответственно.
- 2) Определение элемента обучающей выборки: (<текущий вход>, <желаемый выход>). Текущие входы ($x_0, x_1 \dots x_{N-1}$), должны различаться для всех элементов обучающей выборки. При использовании многослойного персептрона в качестве классификатора

желаемый выходной сигнал ($d_0, d_1 \dots d_{N-1}$) состоит из нулей за исключением одного единичного элемента, соответствующего классу, к которому принадлежит текущий входной сигнал.

3) Вычисление текущего выходного сигнала: Текущий выходной сигнал определяется в соответствии с традиционной схемой функционирования многослойной нейронной сети.

4) Настройка синаптических весов:

Для настройки весовых коэффициентов используется рекурсивный алгоритм, который сначала применяется к выходным нейронам сети, а затем проходит сеть в обратном направлении до первого слоя. Синаптические веса настраиваются в соответствии с формулой:

$$w_{ij}(t+1) = w_{ij}(t) + \eta g_j x_i'$$

где w_{ij} - вес от нейрона i или от элемента входного сигнала i к нейрону j в момент времени t , x_i' - выход нейрона i или i -ый элемент входного сигнала, η - шаг обучения, g_j - значение ошибки для нейрона j . Если нейрон с номером j принадлежит последнему слою, то

$$g_j = y_j(1 - y_j)(d_j - y_j)$$

где d_j - желаемый выход нейрона j , y_j - текущий выход нейрона j . Если нейрон с номером j принадлежит одному из слоев с первого по предпоследний, то

$$g_j = x_j'(1 - x_j') \sum_k g_k w_{jk}$$

где k пробегает все нейроны слоя с номером на единицу больше, чем у того, которому принадлежит нейрон j . Внешние смещения нейронов b настраиваются аналогичным образом.

Рассмотренная модель может быть использована для распознавания образов, классификации, прогнозирования. Были попытки построения экспертных систем на основе многослойных персептронов с обучением по методу обратного распространения.

У этой модели нейронной сети есть и недостатки. Многокритериальная задача оптимизации в методе обратного распространения 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.

Классический метод обратного распространения относится к алгоритмам с линейной сходимостью. Для увеличения скорости сходимости необходимо использовать матрицы вторых производных функции ошибки.

Несмотря на указанные недостатки обратное распространение - первый эффективный алгоритм обучения многослойных нейронных сетей. Один из самых популярных алгоритмов обучения, с его помощью были решены и решаются многочисленные практические задачи.

6.3. Модификации алгоритма.

Были предложены многочисленные модификации алгоритма обратного распространения, которые связаны с использованием различных функций ошибки, различных процедур определения направления и величины шага:

1) функции ошибки:

- интегральные функций ошибки по всей совокупности обучающих примеров,
- функции ошибки целых и дробных степеней.

2) процедуры определения величины шага на каждой итерации:

- дихотомия,
- инерционные соотношения, например

$$w_{ij}(t+1) = w_{ij}(t) + \eta g_j x_i' + \alpha (w_{ij}(t) - w_{ij}(t-1))$$

(α - некоторое положительное число, меньше единицы) ;

- отжиг.

3) процедуры определения направления шага:

- с использованием матрицы производных второго порядка (метод Ньютона и др.),
- с использованием направлений на нескольких шагах (пар-тан метод и др.).

6.4. Гетероассоциативная память

Использование нейропарадигмы "back propagation" для построения моделей гетероассоциативной памяти является традиционным. Нейронная сеть в процессе обучения приобретает способность строить ассоциации

между входным сигналом X и выходным сигналов Y . Обучающая выборка состоит из пар

(<вход X >, <известный выход Y >).

В общем случае X и Y - вектора. В моделях гетероассоциативной памяти размерности входных и выходных векторов различаются ($\dim X \neq \dim Y$). В большинстве задач распознавания и прогнозирования $\dim X > \dim Y$. Дан некоторый входной вектор, требуется определить, к какому классу он относится. В задачах восстановления образа по номеру класса (по некоторому коду) $\dim X < \dim Y$.

6.5. Прогнозирование

С математической точки зрения задача прогнозирования является частным случаем задачи построения гетероассоциативной памяти. В качестве входных сигналов сети используются временные ряды, представляющие значения контролируемых переменных на некотором интервале времени. Выходной сигнал - множество переменных, которое является подмножеством переменных входного сигнала.

Характерные для прогнозирования проблемы:

1. элементы входных сигналов принадлежат разным типам данных,
2. интервалы значений разных элементов входных сигналов существенно различаются,
3. на значения выходных сигналов сети влияют не столько абсолютные значения входных сигналов, сколько их небольшие изменения.

Эти проблемы могут быть решены с помощью масштабирования элементов входных сигналов (см. ниже).

6.6. Автоассоциативная память

В моделях автоассоциативной памяти размерности входных и выходных сигналов совпадают (в результате функционирования сети необходимо получить выходной сигнал, который имеет тот же тип и ту же размерность, что и входной сигнал).

7. Сети Кохонена. (Kohonen's Neural Network)

7.1. АВТОРЫ И ИСТОРИЯ СОЗДАНИЯ

Предложены Кохоненом в 1984 году. К настоящему времени существует множество модификаций исходной модели с богатой математической теорией вокруг них.

7.2. МОДЕЛЬ

В мозге нейроны располагаются в определенном порядке так, что некоторые внешние физические воздействия вызывают ответную реакцию нейронов из определенной области мозга. Например, в той части мозга, которая отвечает за восприятие звуковых сигналов, нейроны группируются в соответствии с частотами входного сигнала, на которых они резонируют. Хотя строение мозга в значительной степени предопределяется генетически, отдельные структуры мозга формируются в процессе самоорганизации. Алгоритм Кохонена в некоторой степени напоминает процессы, происходящие в мозге.

Алгоритм Кохонена дает возможность строить нейронную сеть для разделения векторов входных сигналов на подгруппы. Сеть состоит из M нейронов, образующих прямоугольную решетку на плоскости (рис. 5).

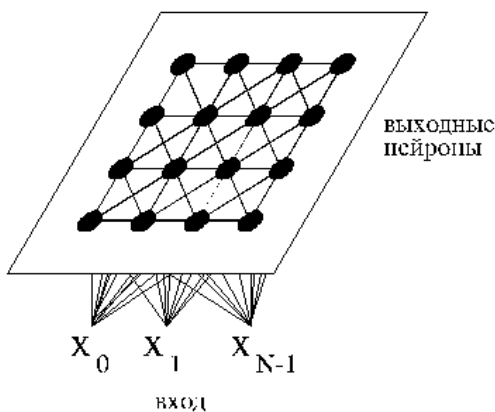


Рис.5. Сеть Кохонена.

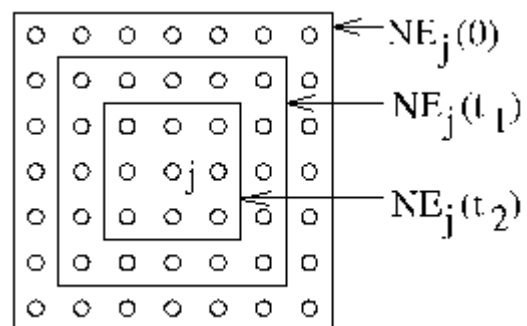


Рис.6. Зоны топологического соседства на карте признаков в различные моменты времени

Элементы входных сигналов подаются на входы всех нейронов сети. В процессе работы алгоритма настраиваются синаптические веса нейронов.

Входные сигналы - вектора действительных чисел - 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 (входным сигналам).

Для реализации алгоритма необходимо определить меру соседства нейронов (меру близости). На рис. 6 показаны зоны топологического соседства нейронов на карте признаков в различные моменты времени. $NE_j(t)$ - множество нейронов, которые считаются соседями нейрона j в момент времени t . Зоны соседства уменьшаются с течением времени.

7.2.1. Алгоритм Кохонена формирования карт признаков:

Шаг 1. Инициализация сети:

Весовым коэффициентам сети присваиваются малые случайные значения. Общее число синаптических весов - $M*N$ (см. рис. 5). Начальная зона соседства показана на рис. 6.

Шаг 2. Предъявление сети нового входного сигнала.

Шаг 3. Вычисление расстояния до всех нейронов сети:

Расстояния d_j от входного сигнала до каждого нейрона j определяются по формуле:

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2,$$

где

x_i - i -ый элемент входного сигнала в момент времени t ,

$w_{ij}(t)$ - вес связи от i -го элемента входного сигнала к нейрону j в момент времени t .

Шаг 4. Выбор нейрона с наименьшим расстоянием:

Выбирается нейрон j^* , для которого расстояние d_j наименьшее.

Шаг 5. Настройка весов нейрона j^* и его соседей:

Производится подстройка весов для нейрона j^* и всех нейронов из его зоны соседства NE . Новые значения весов:

$$w_{ij}(t+1) = w_{ij}(t) + r(t)(x_i(t) - w_{ij}(t)),$$

где $r(t)$ - шаг обучения, уменьшающийся с течением времени (положительное число, меньше единицы).

Шаг 6. Возвращение к шагу 2.

7.3. ОБЛАСТИ ПРИМЕНЕНИЯ:

кластерный анализ, распознавание образов, классификация.

7.4. НЕДОСТАТКИ.

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

7.5. ПРЕИМУЩЕСТВА.

В отличие от сети ART Гроссберга, сеть Кохонена способна функционировать в условиях помех, так как число классов фиксировано, веса модифицируются медленно, настройка весов заканчивается после обучения (в сети ART настройка продолжается непрерывно).

7.6. МОДИФИКАЦИИ.

Одна из модификаций состоит в том, что к сети Кохонена добавляется сеть MAXNET, которая определяет нейрон с наименьшим расстоянием до входного сигнала.

Список литературы

1. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // Докл. АН СССР, том 108, с. 2, 1956.
2. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения // Докл. АН СССР, том 114, с. 953-956, 1957.
- Повторная публикация: Нейрокомпьютер, N 1-2, с. 51-55, 1994.
3. Kolmogorov A.N. On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition, American Math. Soc. Transl., 28 (1963), pp. 55-63.
4. Hecht-Nielsen R. Kolmogorov's Mapping Neural Network Existence Theorem// IEEE First Annual Int. Conf. on Neural Networks, San Diego, 1987, Vol. 3, pp. 11-13.
5. Арнольд В.И. // Докл. АН СССР, том 114, N 4, 1957.
6. Muller B., Reinhart J. Neural Networks: an introduction, Springer-Verlag, Berlin Heidelberg, 1990.
7. Widrow B., Lehr M.A. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation // Proceedings of the IEEE, vol. 78, No. 9, September, 1990, p. 1415-1442.
8. Горбань А.Н. Обучение нейронных сетей. М.: СП ПараГраф. 1990.
9. Wynne-Jones M. Node splitting: A constructive algorithm for feed-forward neural networks // Neural Computing and Applications, v.1, No. 1, 1993, p.17-22.
10. Wynne-Jones M. Constructive algorithms and pruning: Improving the multi layer perceptron. In: Vichnevetsky R., Miller JJH, editors. Proceeding of the 13th IMACS World Congress on Computation and Applied Mathematics; 1991 July; Dublin: 747-750.
11. Ash T. Dynamic node creation in backpropagation networks. La Jolla (CA): Institute for cognitive Science, UCSD; 1989 Feb. Technical Report 8901.
12. Lippman R.P. An introduction to computing with neural nets // IEEE ASSP Magazine. Apr. 1987. P.4-22.
13. Muller B., Reinhardt J. Neural networks. Springer -Verlag. 1990. 267 p.

14. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск, «Наука», 1996.