Министерство образования Российской Федерации

Тамбовский государственный технический университет

ВОПРОСЫ ПРИБЛИЖЕНИЯ ФУНКЦИЙ

Методические указания
для студентов 2 курса специальности 2203
дневной формы обучения

Тамбов
Издательство ТГТУ
2001

УДК 519.6(075)
ББК Вя73
В74

Утверждено Редакционно-издательским советом университета

Рецензент
Доцент, кандидат технических наук
М. Н. Краснянский

Авторы-составители:
Ю. В. Литовка, А. В. Романенко, И. Л. Коробова

В74 Вопросы приближения функций: Метод. указ. / Авт.-сост.: Ю. В. Ли-товка, А. В. Романенко, И. Л. Коробова. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2001. 20 с.

Методические указания содержат описание современных методов интерполяции и аппроксимации, в том числе методики использования теории нейронных сетей как универсального аппроксиматора.

Предназначены для студентов специальности 2203 дневной формы обучения при изучении основ численных методов решения задач приближения функций в рамках дисциплины "Вычислительная математика".

УДК 519.6(075)
ББК Вя73

ВВЕДЕНИЕ

Возникновение вычислительных машин ознаменовало большой скачок в развитии точных и технических наук. В результате появления ЭВМ с программным управлением осуществилась возможность увеличить скорость расчетов от 0,1 операции в секунду при ручном счете до 10^9 операций в секунду на современных ЭВМ. Появление качественно новых возможностей в связи с созданием вычислительной техники иногда сравнивается с промышленной революцией. На свет появилось новое понятие - математический эксперимент, который с помощью расчетов позволяет обнаружить и предсказать даже ранее не наблюдавшиеся явления. Краеугольным камнем в этом процессе является процесс математизации различных наук: от химии, экономики, биологии, экологии до медицины и психологии, суть которой состоит в построении математических моделей реальных процессов и явлений. Для получения возможности исследовать математические модели на ЭВМ необходимы численные методы решения поставленных задач, позволяющие получать результат с заданной точностью и в приемлемые сроки.

В практических расчетах нередко используются функции, непосредственное вычисление которых по различным причинам затруднено. Наиболее типичны следующие ситуации:

1) функция $f(x)$ задана таблицей своих значений $y_i = f(x_i)$, $i = 1, 2, \ldots, n$, а вычисление необходимо произвести в точке x, не совпадающей с табличными;

2) вычисление значения функции $f(x)$ связано с проведением сложных расчетов и требует значительных затрат машинного времени (ситуация может быть усугублена необходимостью многократного вычисления функции f);

3) при заданном значении x значение $f(x)$ может быть найдено только по результатам эксперимента (такой метод "вычислений" для вычислительных алгоритмов часто бывает неприемлем).

Возникшая проблема может быть решена следующим образом: функция $f(x)$ приближенно заменяется новой функцией $g(x)$, мало отличающейся от значений $f(x)$ на множестве точек x_i ($i = 1, 2, \ldots, n$). После этой замены значения функции $g(x)$ с определенной погрешностью принимают за значения исходной функции $f(x)$ (конечно, значения $g(x)$ должны вычисляться быстро и надежно).

Данные лабораторные работы предназначены для закрепления у студентов навыков в применении численных методов приближения функций.

Лабораторная работа № 1

МЕТОДЫ ИНТЕРПОЛЯЦИИ

Цель: приобретение навыков применения алгоритмов интерполяции табличной функции.

Задание: произвести численное решение на ЭВМ задачи интерполяции табличной функции с целью нахождения ее значения для заданного значения аргумента.

Методика решения

Пусть в точках $x_0, x_1, \ldots, x_n$, принадлежащих отрезку $[a, b]$, задана таблица значений функции $f(x)$. Задача интерполяции состоит в определении значения функции f для некоторого значения x такого, что $x_0 < x < x_n$ и $x \neq x_i$, $i = 1, 2, \ldots, n - 1$. Точки x_i называются узлами интерполяции.

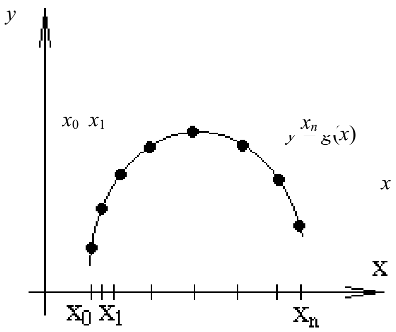

Рис. 1

Для решения задачи интерпо-ляции строится новая функция $g(x)$, график которой проходит через заданные точки, т.е. $g(x_i) = y_i$, $i = 0, 1, 2, \ldots, n$ (рис. 1).

Исходя из условий задачи, можно построить бесконечное мно-жество интерполирующих функций. На практике функцию g выбирают из достаточно узкого класса функций, гарантирующего единственность выбора.

Наиболее простым, полно ис-следованным и широко распрост-раненным на практике является случай интерполирования алгебраи-ческими многочленами вида $P_n(x) = \sum_{k=0}^{n} a_k \varphi_k(x)$, удовлетворяющими условию $P_n(x_i) = y_i$ ($i = 0, 1, \ldots, n$).

Одной из форм записи интерполяционного многочлена является *многочлен Лагранжа*

$$L_n(x)=\sum_{j=0}^{n} y_j l_{nj}(x), \quad (1)$$

где $$L_{nj}(x)=\prod_{\substack{k=1\\k\neq j}}^{n}\frac{x-x_k}{x_j-x_k}. \quad (2)$$

Из записи выражения (2) видно, что многочлен $l_{nj}(x)$ удовлетворяет условию

$$l_{nj}(x)=\begin{cases}1 \text{ при } i=j;\\ 0 \text{ при } i\neq j.\end{cases}$$

Таким образом, становится очевидным, что многочлен Лагранжа является интерполяционным.

Пример 1 Пусть задана таблица значений некоторой функции. Необходимо найти значение этой функции для узла интерполяции x = 2 с помощью интерполяционного многочлена Лагранжа.

x	1	3	4	7
y	0	5	2	4

Произведем расчет по формуле (1):

F(2) = 0 + 5(2 - 1)(2 - 4)(2 - 7)/((3 – 1)(3 - 4)(3 - 7)) +
+ 2(2 - 1)(2 - 3)(2 - 7)/((4 - 1)(4 - 3)(4 - 7)) +
+4(2 - 1)(2 - 3)(2 - 4)/((7 - 1)(7 - 3)(7 - 4)) = 5,25.

Другой способ построения интерполяционного многочлена основывается на том факте, что интерполяционный многочлен можно рассматривать как обобщение отрезка ряда Тейлора:

$$f(x)=\sum_{i=0}^{n}\frac{f^{(i)}(x_0)}{i!}(x-x_0)^i+R_n(x),$$

где $R_n(x)$ - остаточный член формулы Тейлора.

Обобщением понятия производной является понятие разделенной разности. Разделенные разности нулевого порядка совпадают со значениями функции $f(x_i)$. Разности первого порядка определяются равенством

$$f(x_i,\ x_j)=\frac{f(x_j)-f(x_i)}{x_j-x_i}.$$

Разности второго порядка - равенством

$$f(x_i,\ x_j,\ x_k)=\frac{f(x_j,\ x_k)-f(x_i,\ x_j)}{x_k-x_i} \text{ и т.д.}$$

В общем случае разности k-го порядка можно определить из соотношения

$$f(x_1,\ldots,x_k)=\frac{f(x_2,\ldots,\ x_{k+1})-f(x_1,\ldots,\ x_k)}{x_{k+1}-x_1}$$

В результате можно составить таблицу распределенных разностей следующего вида

$\mathbf{X_0}$	$F(x_0)$			
		$F(x_0, x_1)$		
X_1	$F(x_1)$			
		$F(x_1, x_2)$		
X_2	$F(x_2)$	…	…	$F(x_0, x_1, \ldots, x_n)$
…	…			
		$F(x_{n-1}, x_n)$		
X_n	$F(x_n)$			

Любая разность в таблице может быть определена из равенства

$$f(x_1, \ldots, x_k) = \sum_{j=1}^{k} \frac{f(x_j)}{\prod\limits_{i \neq j} (x_j - x_i)}.$$

Разделенные разности обладают следующими свойствами:

1) разделенная разность $f(x_i, x_{i+1}, \ldots, x_{i+k})$ является симметричной относительно своих аргументов (т.е. ее значение неизменно при любом их сочетании);

2) если функция $f(x)$ имеет на отрезке $[a, b]$, содержащем точки $x_i, x_{i+1}, \ldots, x_{i+k}$, производную k-го порядка, то справедливо равенство

$$f(x_i, x_{i+1}, \ldots, x_{i+k}) = \frac{f^{(k)}(\xi)}{k!},$$

где ξ - некоторая точка, расположенная на интервале $[a, b]$.

Используя разделенные разности, можно записать *интерполяционный многочлен Ньютона с разделенными разностями*:

$$P_n(x) = f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) + \ldots +$$
$$+ f(x_0, x_1, \ldots, x_n)(x - x_0)(x - x_1)\ldots(x - x_n).$$

Рассмотрим, как изменится интерполяционная формула Ньютона для случая равноотстоящих узлов интерполяции (т.е. $x_i = x_0 + ih$, $i = 0, 1, 2, \ldots, n$). Для этого случая используем формулу связи разделенных разностей с конечными:

$$f(x_i, x_{i+1}, \ldots x_{i+k}) = \frac{\Delta^k y_i}{h^k k!} = \frac{1}{h^k k!} \sum_{i=0}^{k} (-1)^{k-l} \frac{k! y_{i+l}}{l!(k-l)!},$$

где $\Delta^k y_i$ - конечная разность.

Обозначив через $t = (x - x_0)/h$, запишем *интерполяционный многочлен Ньютона с конечными разностями*. Его можно выразить двумя способами:

1) *для интерполяции вперед*

$$P_n(x) = P_n(x_0 + ht) = y_0 + \frac{\Delta y_0}{1!} t + \frac{\Delta^2 y_0}{2!} t(t-1) + \ldots + \frac{\Delta^n y_0}{n!} t(t-1)\ldots(t-n+1);$$

2) *для интерполяции назад* (используем вместо t величину $q = (x - x_n)/h$)

$$P_n(x) = P_n(x_0 + hq) = y_n + \frac{\Delta y_{n-1}}{1!} q + \frac{\Delta^2 y_{n-2}}{2!} q(q-1) + \ldots +$$
$$+ \frac{\Delta^n y_0}{n!} q(q+1)\ldots(q+n-1).$$

Интерполяционные многочлены Лагранжа или Ньютона на всем отрезке $[a, b]$ с использованием большого числа узлов интерполяции часто не дают хорошего приближения из-за накопления погрешности в процессе вычислений. Помимо этого, вследствие расхождения процесса интерполирования, увеличение числа узлов не всегда приводит к повышению точности. Выходом может стать использование для интерполяции табличной функции полиномов невысокой степени, которые используются на отдельных участках отрезка $[a, b]$. Потребность в наличии функций, сочетающих в себе локальную простоту многочлена невысокой степени и глобальную на всем отрезке $[a, b]$ гладкость, привела к появлению сплайнов.

Пусть отрезок $[a, b]$ разбит на n участков $[x_{i-1}, x_i]$ точками $a = x_0 < < x_1 < \ldots < x_n = b$. Сплайном степени m называется функция $S_m(x)$, обладающая следующими свойствами:

1) функция $S_m(x)$ непрерывна на отрезке $[a, b]$ со всеми своими производными до некоторого порядка p;
2) на каждом частичном отрезке $[x_{i-1}, x_i]$ функция $S_m(x)$ совпадает с некоторым алгебраическим многочленом $P_{m,i}(x)$ степени m.

В вычислительной практике широко распространен случай использования сплайнов, определяемых полиномами третьей степени (кубические сплайны). Такие сплайны на каждом из частных отрезков $[x_{i-1}, x_i]$ совпадают с кубическими многочленами:

$$S_3(x) = P_{3,i}(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3$$

и имеют на отрезке $[a, b]$ как минимум одну непрерывную производную $S'_3(x)$.

Зададим некоторую функцию $y = f(x)$ таблицей ее значений $y_i = f(x_i)$, $i = 0, 1, \ldots, n$. Сплайн $S_m(x)$ считается интерполяционным, если $S_m(x_i) = y_i$. Значение $s_i = S'_m(x_i)$ называется наклоном сплайна в точке x_i. Интерполяционный кубический сплайн на отрезке $[x_{i-1}, x_i]$ однозначно определяется значениями y_{i-1}, y_i, s_{i-1}, s_i, а формула для вычислений приобретает вид

$$S_3(x) = P_{3,i}(x) = \frac{(x - x_i)^2\left(2(x - x_{i-1}) + h_i\right)}{h_i^3} y_{i-1} + \frac{(x - x_{i-1})^2\left(2(x_i - x) + h_i\right)}{h_i^3} y_i +$$
$$+ \frac{(x - x_i)^2 (x - x_{i-1})}{h_i^2} S_{i-1} + \frac{(x - x_{i-1})^2 (x - x_i)}{h_i^2} S_i,$$

где $h_i = x_i - x_{i-1}$.

Открытым остается вопрос о вычислении наклонов сплайна s_{i-1}, s_i. Эту проблему можно решить следующим образом. Потребуем непрерывность второй производной сплайна $S_3''(x)$. Для этого наклоны s_i необходимо выбирать так, чтобы в точках "стыка" x_i совпадали значения вторых производных многочленов $P_{3,i}$ и $P_{3,i+1}$: $P''_{3,i}(x_i) = P''_{3,i+1}(x_i)$.

Приравнивая выражения вторых производных, получим систему уравнений относительно коэффициентов s_i:

$$h_i^{-1}s_{i-1} + 2(h_i^{-1} + h_{i+1}^{-1})s_i + h_{i+1}^{-1}s_{i+1} = 3[h_i^{-2}(y_i - y_{i-1}) + h_{i+1}^{-2}(y_{i+1} - y_i)],$$
$$i = 1, 2, \ldots, n - 1.$$

Число уравнений (равное n - 1) меньше числа неизвестных (равное n + 1). Для того, чтобы выбрать два оставшихся уравнения, можно потребовать равенство нулю вторых производных функции f на краях отрезка $[a, b]$.

Порядок выполнения

1 Составить алгоритм решения задачи интерполяции табличной функции для двух заданных методов.

2 Составить программу для ЭВМ, реализующую составленные алгоритмы.

3 С помощью составленной программы интерполяции табличной функции найти значение функции f в заданной точке x (варианты заданий приведены в табл. 1).

4 Сравнить результаты решения интерполяционной задачи разными методами.

Содержание отчета

1 Постановка задачи лабораторной работы.

2 Описание составленных алгоритмов и распечатка разработанных программ.

3 Результаты решения задачи для разных значений аргументов (3 - 4 значения).
Литература: [1 - 4].

Таблица 1

№	$F(x_1)$	$F(x_2)$	$F(x_3)$	$F(x_4)$	$F(x_5)$
1	2,5	5,0	8,5	13,0	18,5
2	0,5	1,0	2,5	5,0	8,5
3	4,0	9,0	16,0	25,0	36,0
4	2,0	7,0	14,0	23,0	34,0
5	-1,5	-1,0	0,5	3,0	6,5
6	0,5	3,0	6,5	11,0	16,5
7	1,5	5,0	9,5	15,0	21,5
8	3,5	7,0	11,5	17,0	23,5
9	-0,5	-1,0	-0,5	1,0	3,5
10	-5,9	-6,6	-7,1	-7,4	-7,5
11	1,2	-1,2	-3,2	-4,8	-6,0
12	0.0	1,0	4,0	9,0	16,0
13	-6,8	-9,2	-11,2	-12,8	-14,0
14	-0,8	2,8	6,8	11,2	16,0
15	8,6	15,4	23,4	32,6	43,0
16	2,6	9,4	17,4	26,6	57,0
17	-1,4	-4,6	-6,6	-7,0	-7,4
18	-7,4	-10,6	-12,6	-13,4	-13,0
19	11,5	18,0	27,5	40,0	55,5
20	-4,5	2,0	11,5	24,0	59,5
21	-8,5	-6,0	-0,5	8,0	19,5
22	4,2	11,8	21,8	34,2	49,0
23	-1,8	-2,2	-0,2	4,2	11,0
24	7,2	10,8	14,8	19,2	24,0
25	6,2	13,8	23,8	36,2	51,0

Лабораторная работа № 2

АППРОКСИМАЦИЯ ТАБЛИЧНЫХ ФУНКЦИЙ

Цель: приобретение навыков решения задач аппроксимации табличных функций.

Задание: аппроксимировать табличную функцию с заданной погрешностью.

Методика аппроксимации

Очень часто в виде таблиц, ставящих в соответствие некоторому значению аргумента x значение функции $y = f(x)$, оформляются результаты экспериментальных исследований и подобные им данные, в составе которых неизбежно присутствует некоторая погрешность. В этом случае результат решения задачи интерполяции также будет отличаться от истинного значения функции f, а это нежелательно. Решением проблемы может стать некоторая функция $\varphi(x)$, которая не проходит через значения $y_i = f(x_i)$ ($i = 0, 1, \ldots, n$), но ее график расположен как можно ближе к точкам (x_i, y_i). На практике функцию $\varphi(x)$ часто задают в виде полинома $\varphi(x) = P_k(x) = = a_0 + a_1x + a_2x^2 + \ldots + a_kx^k$. С увеличением степени полинома его график будет проходить все ближе от точек (x_i, y_i) и по достижении им степени n он точно пройдет через заданные точки, превратившись в интерполяционный многочлен.

Задача аппроксимации состоит в определении коэффициентов $a_0, a_1, \ldots, a_n$ для полинома степени k, отклонение которого от точек (x_i, y_i) ($i = 0, 1, \ldots, n$) не превышает заданной погрешности.

Классическим методом решения задачи аппроксимации является *метод наименьших квадратов*. Для нахождения коэффициентов аппроксимирующего полинома P_k составляется сумма квадратов величин отклонения его от данных таблицы:

$$S = \sum_{i=0}^{n}\left[P_k(x_i) - y_i\right]^2 = \sum_{i=0}^{n}\left[a_0 + a_1x_i + \ldots + a_kx_i^k - y_i\right]^2.$$

Фактически мы получили функцию неизвестных переменных $a_0, a_1, \ldots, a_k$. Минимизируем функцию $S(a_0, a_1, \ldots, a_k)$, для чего приравняем нулю ее частные производные по входящим в нее переменным:

$$\frac{\partial S}{\partial a_0} = 2\sum_{i=0}^{n}\left(a_0 + a_1x_i + \ldots + a_kx_i^k - y_i\right) = 0;$$

$$\frac{\partial S}{\partial a_1} = 2\sum_{i=0}^{n}x_i\left(a_0 + a_1x_i + \ldots + a_kx_i^k - y_i\right) = 0;$$

$$\frac{\partial S}{\partial a_0} = 2\sum_{i=0}^{n}\left(a_0 + a_1x_i + \ldots + a_kx_i^k - y_i\right) = 0.$$

Преобразуем полученные суммы следующим образом:

$$na_0 + a_1\sum_{i=0}^{n}x_i + a_2\sum_{i=0}^{n}x_i^2 + \ldots + a_k\sum_{i=0}^{n}x_i^k - \sum_{i=0}^{n}y_i = 0;$$

$$a_0\sum_{i=0}^{n}x_i + a_1\sum_{i=0}^{n}x_i^2 + a_2\sum_{i=0}^{n}x_i^3 + \ldots + a_k\sum_{i=0}^{n}x_i^{k+1} - \sum_{i=0}^{n}x_iy_i = 0;$$

$$a_0\sum_{i=0}^{n}x_i^k + a_1\sum_{i=0}^{n}x_i^{k+1} + a_2\sum_{i=0}^{n}x_i^{k+2} + \ldots + a_k\sum_{i=0}^{n}x_i^{2k} - \sum_{i=0}^{n}x_iy_i = 0.$$

Мы получили систему линейных алгебраических уравнений, из решения которой можно найти коэффициенты аппроксимирующего полинома.

Использовать метод наименьших квадратов не всегда бывает удобно, так как аппроксимировать заданную табличную функцию с требуемой погрешностью может полином сложного вида, дальнейшее использование которого будет затруднено. Повысить эффективность аппроксимации можно, например, методом наращивания. Зададим некоторую функцию начального приближения F_0 (ее вид может быть подсказан данными таблицы). При отсутствии заданной функции можно задать константу. Для функции "нулевого" уровня ищутся входящие в нее параметры, минимизирующие отклонение Δ_0 от табличной функции:

$$\Delta_0 = \frac{1}{n}\sqrt{\sum_{i=0}^{n}\left(F_0\left(x_i\right) - F_{\text{т}}\right)^2},$$

где n - количество точек; $F_{\text{т}}$ - значение табличной функции.

Если Δ_0 удовлетворяет заданной точности аппроксимации, то поиск завершается. В противном случае производится переход на "первый" уровень, на котором производится построение набора функций наращиванием функции "нулевого" уровня путем ее алгебраического сложения, умножения и деления с набором заранее заданных элементарных одномерных функций: ax, $a\sin x$, $a\exp x$ и т.д., где a - искомый коэффициент. Для всех функций "первого" уровня рассчитываются значения параметров, минимизирующих отклонение Δ_1^i. Если для какой-нибудь функции этого уровня погрешность аппроксимации меньше заданной точности, поиск останавливается. В противном случае относительно функции "первого" уровня с наименьшей погрешностью осуществляется "наращивание" и получается набор функций "второго" уровня. После этого описанные действия повторяются.

Порядок выполнения работы

1 Составить алгоритм решения задачи аппроксимации табличной функции заданным методом (наименьших квадратов или наращивания).
2 Разработать программу для ЭВМ, реализующую составленный алгоритм.
3 Найти аппроксимирующую функцию для своего варианта (см. табл. 1).
4 Привести графическую иллюстрацию решения.

Содержание отчета

1 **Исходная табличная функция и функция, аппроксимирующая ее с заданной точностью.**
2 Графическая иллюстрация полученного решения.
Литература: [1, 5].

Лабораторная работа № 3

НЕЙРОННЫЕ СЕТИ КАК УНИВЕРСАЛЬНЫЙ АППРОКСИМАТОР

Цель: изучение основ нейронного моделирования.

Задание: создать нейронную сеть для аппроксимации таблично заданной функции. Определить оптимальные параметры сети при заданном количестве нейронов в скрытом слое.

Варианты заданий

Входные параметры			Желаемый выход (индекс соответствует номеру варианта)									
*X*1	*X*2	*X*3	*D*1	*D*2	*D*3	*D*4	*D*5	*D*6	*D*7	*D*8	*D*9	*D*10
0	0	0	0	1	0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1	0	1	0	0	1
0	1	0	0	1	1	0	0	1	0	1	1	0
0	1	1	1	0	1	0	0	1	1	0	0	1
1	0	0	0	1	0	1	1	0	1	0	0	0
1	0	1	1	0	0	1	1	0	0	1	1	1

1	1	0	0	1	1	0	0	1	1	0	0	1
1	1	1	1	0	1	0	1	0	0	1	1	0
Число нейронов в скрытом слое			2; 3	2; 4	3; 4	2; 3	2; 4	3; 4	2; 3	2; 4	3; 4	2; 3

Напоминание При аппроксимации функций нейронная сеть должна содержать один дополнительный входной параметр, всегда равный 1 (или -1).

Методические указания

В основе теории нейронных сетей лежит желание воспроизвести функции мозга при решении конкретной задачи. Однако, создающиеся системы не полностью воспроизводят функции мозга, а, скорее, представляют математическую модель, воспроизводящую отдельные возможности человеческого мозга, по аналогии с которым искусственные нейронные сети характеризуются следующими свойствами:

- обучение (т.е. изменение поведения в зависимости от окружающей среды);
- обобщение (реакция сети после обучения будет, до известной степени, нечувствительна к малым изменениям входящих сигналов);
- абстрагирование (способность выявления различий во входных сигналах).

Описание биологического нейрона

Из нейробиологии известно, что человеческий мозг состоит из 10^{10}-10^{11} нейронов. На рис. 2 схематично представлен один биологический нейрон. Он содержит клеточное тело и отростки (аксон и дендриты).

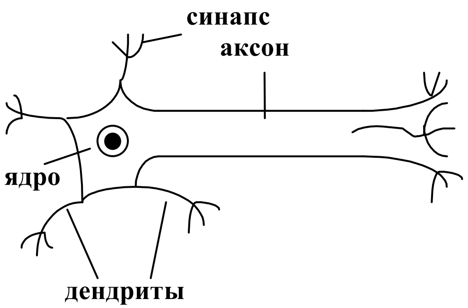

Рис. 2

Клеточное тело состоит из ядра и окружающей его цитоплазмы. На внешней поверхности содержится мембрана, включающая три слоя. Она отделяет клеточное тело от окружающих его крайних окончаний аксона.

Аксон (выход) - отросток нейрона, который служит для передачи нервных импульсов к другим нейронам или эффекторным органам (мускульным волокнам, клеткам желез).

Дендриты (входы) - отростки, которые связывают нейрон с другими нейронами. Связь осуществляется через специальные контакты, называемыми *синапсами*.

В упрощенном виде работу нейрона можно представить так. Клеточное тело принимает входной сигнал от других нейронов через синаптические связи дендритов, преобразует его и передает выходной сигнал через аксон другим нейронам. Скорость передачи зависит и от значений входных сигналов, и от силы синаптических связей. Несмотря на то, что функция нейрона - нелинейная, нейробиологи считают, что большинство нейронов производят линейную аппроксимацию, т.е. выходной сигнал нейрона пропорционален, в некоторой степени, линейной комбинации значений входных сигналов.

Искусственный нейрон

Отдельный обрабатываемый элемент искусственной нейронной сети называется искусственным нейроном. Каждый нейрон производит относительно простую работу. На его вход поступает набор сигналов $X = [x_1, x_2, ..., x_n]$, каждый из которых может быть выходом от другого нейрона или другого источника. Каждый вход умножается на соответствующий угловой коэффициент $W = [w_1, w_2, ..., w_n]$, который соответствует силе синапса биологического нейрона, и поступает на вход суммирующего блока, где все произведения $w_i x_i$ суммируются. По этой величине определяется общий вход нейрона:

$$h = \sum_i (w_i x_i + \theta_i)\ , \tag{1}$$

где θ_i - пороговая величина i-го нейрона.

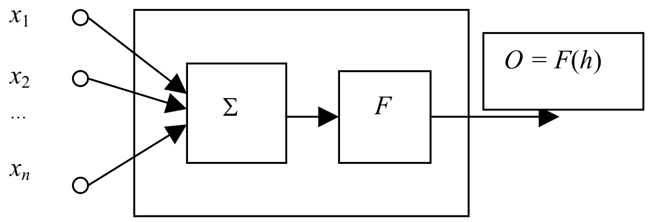

Рис. 3

Для определения выхода нейрона O (рис. 3) используется функция активации:

$$O = F(h) = F\left[\sum_i (w_i x_i + \theta_i)\right]. \tag{2}$$

Наиболее типичными функциями активации являются:

- экспоненциальная

$$F(h) = \frac{2}{1+\exp(-\lambda h)} - 1, \quad \lambda > 0; \tag{3}$$

- функция знака

$$F(h) = \mathrm{sgn}(h) = \begin{cases} +1, & h > 0; \\ -1, & h < 0. \end{cases} \tag{4}$$

Функции, записанные в таком виде, называются биполярными. Возможно использование униполярных функций:

$$F(h) = \frac{1}{1+\exp(-\lambda h)}, \quad \lambda > 0; \tag{5}$$

$$F(h) = \begin{cases} 1, & h > 0; \\ 0, & h < 0. \end{cases} \tag{6}$$

Следует отметить, что при $\lambda \to \infty$ экспоненциальная функция приближается к функции знака.

Многослойные нейронные сети

Для решения практических задач часто используются многослойные нейронные сети. Обычно в таких сетях все нейроны в слое связаны со всеми нейронами в предыдущем слое через однонаправленную связь. При решении задач аппроксимации чаще используется нейронная сеть с одним скрытым слоем (рис. 4).

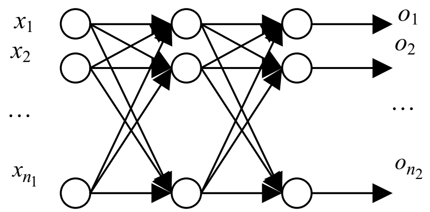

Рис. 4

Многослойная нейронная сеть имеет разное количество нейронов в слоях и разные весовые коэффициенты нейронов. Каждый нейрон характеризуется множеством входов и одним выходом. Связь вход-выход для сети, изображенной на рис. 3, можно представить в матричной форме:

$$O = F(X) = f\,[W^2 f\,[W^1 X]], \qquad (7)$$

где X - вектор входных параметров; O - вектор выходных параметров; W^1, W^2- матрицы весовых коэффициентов для скрытого и выходного слоя, соответственно; f - функция активации. При решении задачи аппроксимации обычно используется экспоненциальная функция (3, 5).

Обучение нейронной сети

Очевидно, чтобы система хорошо работала и решала практические задачи, необходимо ее обучить. Если говорить в общем, то обучение - относительно постоянный процесс изменения поведения при поступлении жизненного опыта. Если говорить о человеке, то результат его обучения оценивается по действиям и поступкам. Обучение же нейронных сетей - более прямой процесс.

Обучение нейронных сетей рассматривается как процесс аппроксимации непрерывной функции $y(X)$ другой функцией $Y(W, X)$, где $X = [x_1, x_2, ..., x_n]^t$ - входной вектор, а $W = [w_1, w_2, ..., w_n]^t$ - вектор весовых коэффициентов.

Задача обучения состоит в выборе вектора W такого, что достигается лучшая аппроксимация:

$$\rho(Y(W^*, X), y(X)) \le \rho(Y(W^*, X), y(X)), \qquad (8)$$

где $\rho(Y(W, X), y(X))$ - функция расстояния, которая определяет значение качества аппроксимации между $Y(W, X)$ и $y(X)$).

Все алгоритмы обучения делятся на две большие группы: с учителем и без учителя.

Для нашего случая используется алгоритм обучения с учителем. Предполагается, что в каждый момент времени вместе с входами формируется желаемое значение выхода d, которое поступает от учителя. Это иллюстрируется на рис. 5.

По значениям реального выхода и желаемого выхода формируется ошибка, которая используется для корректировки параметров нейронной сети. Множество

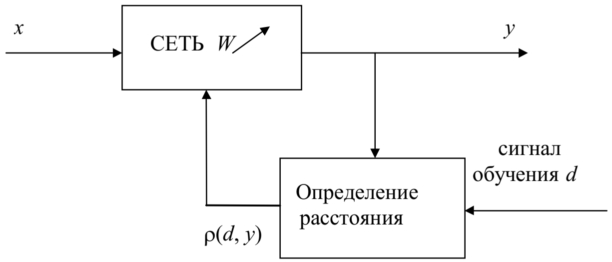

Рис. 5

входных и выходных образцов, называемых обучающим множеством, является необходимым для такого способа обучения. "Учитель" оценивает негативное направление градиента ошибки и соответственно сеть уменьшает ошибку. Во многих ситуациях, входы, выходы и вычисляемые градиенты являются детерминистическими, однако, минимизация осуществляется по случайному закону. И, как результат, большинство алгоритмов обучения с учителем используют стохастическую минимизацию ошибки в многомерном пространстве весов.

Для нашего случая рассмотрим стандартный алгоритм обучения многослойных нейронных сетей на основе обратного распространения ошибки (back-propagation).

Цель обучения состоит в определении всех весовых коэффициентов, при которых ошибка вычислений будет минимальной. Обучение сети осуществляется на основе множества пар "вход-выход". Каждый пример обучения состоит из вектора $X = [x_1, x_2, ..., x_{n1}]$ входных сигналов и вектора $D = [d_1, d_2, ..., d_{n3}]$ желаемых результатов. Обучение состоит в определении всех весовых коэффициентов таких, что значение ошибки между желаемыми и действительными выходными сигналами будет минимальной (близкой к 0).

Рассматриваемый метод использует пошаговый градиентный подход для минимизации функции квадрата ошибки. Тогда локальная функция ошибки для p-го примера обучения формулируется как:

$$E_p = \frac{1}{2}\sum_{j=1}^{n_2}(d_{jp} - y_{jp})^2 = \frac{1}{2}\sum_{j=1}^{n_2} e_{jp}^2 . \tag{9}$$

Тогда общая функция ошибки имеет вид

$$E = \sum_p E_p = \frac{1}{2}\sum_p \sum_j (d_{jp} - y_{jp})^2, \tag{10}$$

где d_{jp} и y_{jp} - желаемый и действительный выходные сигналы j-го выходного нейрона для p-го образца, соответственно.

Подход, который используется нами, предполагает, что для каждого примера обучения синаптические веса w_{ji}^s (s - число уровней сети) изменяются на величину Δw_{ji}^s пропорционально отрицательному градиенту локальной функции E_p:

$$\Delta w_{ji}^s = \eta \frac{\partial E_p}{\partial w_{ji}^s}; \quad \eta > 0, \tag{11}$$

где η - параметр обучения (малое число). Или в непрерывной форме:

$$\frac{dw_{ji}^s}{dt} = -\mu \frac{\partial E_p}{\partial w_{ji}^s}; \quad \mu > 0. \qquad (12)$$

По этой процедуре минимизируется общая функция ошибки $E = \sum_p E_p$. Решение этой задачи возможно с использованием многих итерационных алгоритмов. Рассмотрим нахождение решения на примере двухслойной нейронной сети.

Определим синаптические веса w_{ji}^s ($s = 2$) выходного уровня. Имеем:

$$\Delta w_{ji}^2 = -\eta \frac{\partial E_p}{\partial w_{ji}^2} = -\eta \frac{\partial E_p}{\partial o'_j} \frac{\partial o'_j}{\partial w_{ji}^2}. \qquad (13)$$

При этом

$$o'_j = \sum_{i=1}^{n_2} \left(w_{ji}^2 h_i \right). \qquad (14)$$

Локальная ошибка, называемая "дельта", определяется как

$$\delta_j^2 = \frac{\partial E_p}{\partial o'_j} = -\frac{\partial E_p}{\partial e_{jp}} \frac{\partial e_{jp}}{\partial o'_j} = e_{jp} \frac{\partial f^2}{\partial o'_{jp}} = (d_p - y_p) \frac{\partial f^2}{\partial o'_{jp}}. \qquad (15)$$

Теперь можно получить общую формулу для определения весов в выходном слое:

$$\Delta w_{ji}^2 = \eta \delta_j^2 h_i. \qquad (16)$$

Затем определим синаптические веса в скрытом слое. Можем записать:

$$\Delta w_{ji}^1 = -\eta \frac{\partial E_p}{\partial w_{ji}^1} = -\eta \frac{\partial E_p}{\partial h'_j} \frac{\partial h'_j}{\partial w_{ji}^1} = \eta \delta'_j x_i, \qquad (17)$$

где локальная ошибка для скрытого слоя определяется как

$$\delta'_j = -\frac{\partial E_p}{\partial h'_j}, \quad j = \overline{1, n_2}. \qquad (18)$$

Как заметим, в данном случае локальная ошибка определяется не напрямую через известные значения, как для выходного слоя. Поэтому, продолжим рассуждения для определения локальной ошибки в скрытом слое. Запишем:

$$\delta'_j = -\frac{\partial E_p}{\partial h'_j} = -\frac{\partial E_p}{\partial h_j} \frac{\partial h_j}{\partial h'_j}. \qquad (19)$$

Принимая $h_j = f^1(h_j^1)$, имеем:

$$\delta_j^1 = -\frac{\partial E_p}{\partial h_j} \frac{\partial f^1}{\partial h'_j}. \qquad (20)$$

Фактор $\partial E_p / \partial h_j$ будем определять как

$$-\frac{\partial E_p}{\partial h_j} = -\sum_{i=1}^{n_2} \frac{\partial E_p}{\partial o_i'} \frac{\partial o_i'}{\partial h_j} = \sum_{i=1}^{n_2} \left(-\frac{\partial E_p}{\partial o_i'} \right) \frac{\partial}{\partial h_j} \left(\sum_{k=1}^{n_2} w_{ik}^2 h_k \right) = \sum_{i=1}^{n_2} \left(\delta_i^2 w_{ij}^2 \right). \qquad (21)$$

Тогда локальную ошибку в скрытом слое можно определить, используя формулу:

$$\delta_j^1 = \frac{\partial f^1}{\partial h_j'} \sum_{i=1}^{n_2} \delta_i^2 w_{ij}^2. \qquad (22)$$

Как видно, локальная ошибка внутреннего скрытого слоя определяется на базе локальных ошибок следующего слоя. Стартуя с высшего выходного слоя, мы вычисляем δ^2. Эта ошибка возвращается на нижний уровень. На рис. 6 представлена функциональная схема алгоритма обучения с обратным распространением ошибки. Главным при данном способе обучения является определение локальной ошибки δ_j^s (s = 1, 2). В выходном слое ошибка определяется как функция от желаемого и действительного результатов и сигмоидальной функции активации. Для скрытого слоя локальная ошибка определяется на базе ранее определенных локальных ошибок выходного слоя.

Тогда алгоритм реализации обучения с обратным распространением ошибки включает следующую последовательность шагов.

1 Инициализируются все синаптические веса w_{ij}^s как малое случайное число.

2 Задаются все примеры обучения в виде пар "вход-выход"; вычисляются действительные значения выходов всех нейронов, используя заданные значения w_{ij}^s и значения входов.

3 Используя значения желаемого и действительного выходов, определяются локальные ошибки δ_j^s для всех уровней.

4 Пересчитываются синаптические веса по итерационной формуле

$$\Delta w_{ji}^s = \eta \delta_j^s x_i^s, \quad s = 1, 2. \qquad (23)$$

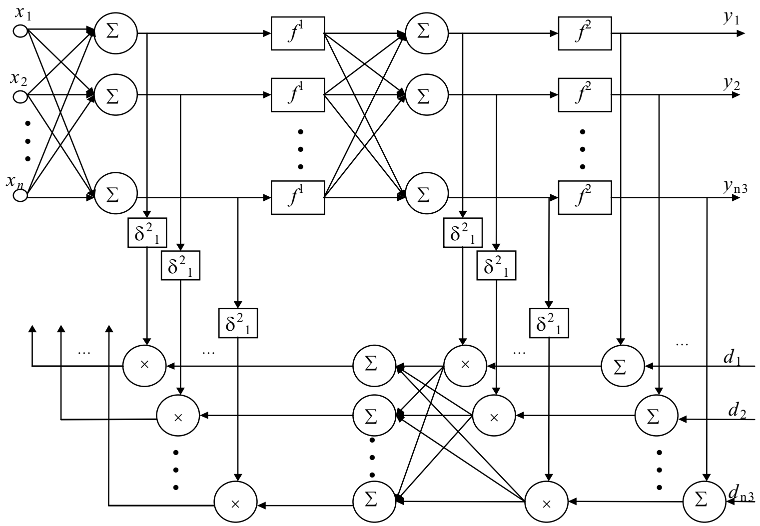

Рис. 6

5 Переходим ко второму примеру обучения и возвращаемся на пункт 2.

Заметим, что все тренировочные примеры обрабатываются циклически, пока ошибка обучения не станет малой. После обучения многослойный персептрон (нейронная сеть) обычно обладает свойствами объекта, для которого он обучался. Теперь можно вводить любые входные значения и получать выходные без дополнительного обучения.

Содержание отчета

1 Задание.
2 Схемы нейронной сети при заданном числе нейронов в скрытом слое.
3 Описание алгоритма решения.
4 Результат, включающий оптимальные значения весовых коэффициентов и число шагов обучения.

Контрольные вопросы

1 Какие свойства биологического нейрона используются при создании искусственных нейронных сетей?
2 Как используются многослойные нейронные сети?
3 Какие известны алгоритмы обучения?
4 Чем отличаются ошибка обучения и ошибка обобщения?
5 На каком принципе основан метод обучения с обратным распространением ошибки?

СПИСОК ЛИТЕРАТУРЫ

1 Амосов А. А., Дубинский Ю. А., Копченова Н. В. Вычислительные методы для инженеров. М.: Высшая школа, 1994. 544 с.
2 Бахвалов И. В., Жидков Н. П., Кобельков Г. М. Численные методы. М.: Физматлит, 2000. 624 с.
3 Демидович Б. П., Маарон И. А., Шувалова Э. З. Численные методы анализа. М.: Наука, 1967. 368 с.

4 Самарский А. А., Гулин А. В. Численные методы. М.: Наука, 1989. 432 с.

5 Литовка Ю. В., Романенко А. В., Коломин А. В. Алгоритм поиска аналитической зависимости табличной функции многих переменных // Тез. докл. 9 Международной конференции "Математические методы в химии и химической технологии". Тверь, 1995. С. 11 - 12.

6 Zurada J. M. Introduction to artificial neural systems. West Publishing Company, 1991. 680 p.

7 Wasserman P. Neurocomputing. Theoty and practice. Nostram Reinhold, 1990. (Рус. пер. Ф. Уоссерман. Нейрокомпьютерная техника. М.: Мир, 1992).

8 Горбань А. Н., Россиев Д. А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996.

9 Cichocki A., Unbehauen R. Neural Networks for Optimization and Signal Processing. New York, 1993.

10 Blum A. Neural Networks in C++. New York, 1992.

11 Горбань А. Н., Функции многих переменных и нейронные сети // Соровский образовательный журнал. 1998. № 12. С. 105 - 112.

12 Korobova I. L., Gatapova N. Z., Konovalov V. I., Kudra T. Opportunities for using fuzzy systems and neural networks to optimize quality of dried materials with complex rheology // Proc. IDS'2000, Elsevier. Amsterdam, 2000.

Учебное издание

ВОПРОСЫ ПРИБЛИЖЕНИЯ ФУНКЦИЙ

Методические указания

Авторы-составители: **ЛИТОВКА** Юрий Владимирович,
РОМАНЕНКО Александр Васильевич,
КОРОБОВА Ирина Львовна

Редактор Т. М. Глинкина
Инженер по компьютерному макетированию
Г. Ю. Корабельникова

ЛР № 020851 от 13.01.99 Плр № 020079 от 28.04.97

Подписано к печати 19.09.2001
Формат 60 × 84/16. Бумага офсетная. Печать офсетная
Объем: 1,16 усл. печ. л.; 1,2 уч.-изд. л.
Тираж 150 экз. С. 591

Издательско-полиграфический центр ТГТУ
392000, Тамбов, Советская, 106, к. 14

ВОПРОСЫ
ПРИБЛИЖЕНИЯ ФУНКЦИЙ

Издательство ТГТУ