

В. В. Сергеев, В. Н. Копенков, А. В. Чернов

*(Самара)***СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ НЕЙРОННЫХ СЕТЕЙ  
И ИЕРАРХИЧЕСКОЙ АППРОКСИМАЦИИ  
В ЗАДАЧАХ ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЙ\***

Рассмотрены два различных нелинейных метода аппроксимации многомерной регрессии: нейронные сети и иерархическая аппроксимация функции, которые применяются к задачам фильтрации изображений в условиях существования априорной информации в виде пары изображений, интерпретируемых как «идеальное» и «искаженное». Представлена методология построения иерархической функции аппроксимации, и проведено сравнение эффективности рассмотренных методов аппроксимации на различных системах признаков, полученных в результате обработки изображения в режиме «скользящего окна».

**Введение.** Задача восстановления зависимостей по эмпирическим данным в общей постановке выглядит следующим образом. Есть набор данных  $X$ , каждый элемент которого характеризуется вектором признаков (параметров)  $\xi = (x_1, x_2, \dots, x_K)$ , множество ответов  $Y$  и множество отображений (функций)  $U$  из  $X$  в  $Y$ . Существует отображение  $y^*: X \rightarrow Y$  (не обязательно принадлежащее  $U$ ), значения которого априорно известны только на конечном наборе элементов  $y(n) = y^*(\xi(n))$ ,  $n = 1, \dots, N$  (обучающей выборке). Требуется по априорной информации построить функцию  $\bar{y} = f^*(\xi)$ ,  $f^* \in U$ , наилучшим образом приближающую восстанавливаемую зависимость не только на элементах обучающей выборки, но и на всем множестве объектов  $X$ .

Рассмотрим задачу восстановления регрессии, без ограничения общности подразумевая  $X = R^K$ ,  $Y = R$  и используя среднеквадратичный критерий близости как функционал качества:

$$\varepsilon^2 = \sum_n (y(n) - f^*(\xi(n)))^2. \quad (1)$$

Конкретный вид класса  $U$  алгоритмов (функций регрессии) во многом определяется исследователем на основе специфики задачи.

\* Работа выполнена при поддержке Министерства образования и науки РФ, Администрации Самарской области и Американского фонда гражданских исследований и развития (CRDF) (проект SA-014-02) в рамках российско-американской программы «Фундаментальные исследования и высшее образование» (BRHE).

**Методы построения функции регрессии.** Существует множество различных способов построения функции регрессии. Мы остановимся на сравнении двух методов: нейронные сети [1, 2] и древовидная аппроксимация [3, 4]. Целью данной работы является их экспериментальное сравнение в применении к задачам фильтрации изображений.

*Нейронная сеть.* Используем нейронную сеть с  $K$  входами, одним скрытым слоем ( $M$  нейронов в скрытом слое) и одним выходом (рис. 1).

Для подстройки весов – коэффициентов синоптических связей  $w_{ij}^{(k)}$  ( $k$  – индекс номера слоя;  $i$  и  $j$  – индексы нейронов слоя  $k$  и  $(k + 1)$ ) – использовался известный алгоритм обратного распространения ошибки [1, 2], который в случае двухслойной нейронной сети выглядит следующим образом:

$$w_{ij}^{(k)}(n) = w_{ij}^{(k)}(n-1) - \eta \frac{\partial E^{(k)}}{\partial w_{ij}^{(k)}}(n), \quad k=1,2,$$

$$\frac{\partial E^{(2)}}{\partial w_{ij}^{(2)}}(n) = [\bar{Y}^{(2)}(n) - Y(n)] \bar{Y}_j^{(1)}(n),$$

$$\frac{\partial E^{(1)}}{\partial w_{ij}^{(1)}}(n) = [\bar{Y}^{(2)}(n) - Y(n)] w_{ij}^{(2)}(n) X_i(n) F'(\bar{Y}_j^{(1)}(n)),$$

где  $\eta$  – коэффициент скорости обучения,  $0 < \eta < 1$ .

В качестве активационной функции были исследованы:

- 1) функция Ферми  $y = f(s) = \frac{1}{1 + e^{-s}}$ ;
- 2) гиперболический тангенс  $y = f(s) = \text{th}(s)$ ;
- 3) рациональная функция  $y = f(s) = \frac{1}{1 + |s|}$ ;
- 4) линейная функция  $y = f(s) = s$ .

Процесс обучения нейронной сети продолжается до тех пор, пока ошибка аппроксимации не достигнет минимума либо будет меньше какого-то на-

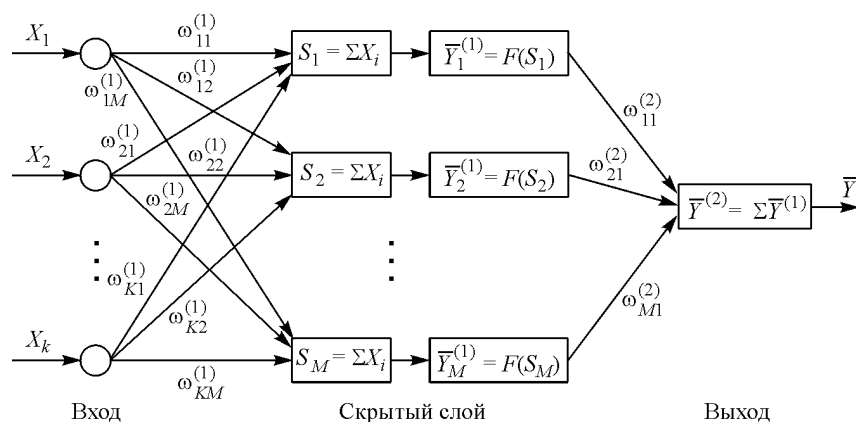


Рис. 1. Схема нейронной сети.

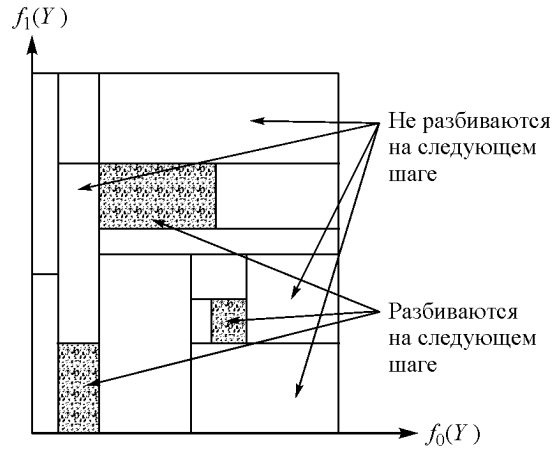


Рис. 2. Схема разбиения признакового пространства

перед заданного порога. На исследованных изображениях лучше всего себя показали функция Ферми и гиперболический тангенс.

*Иерархическая аппроксимация.* В процессе построения функции аппроксимации область определения, представляющая собой  $K$ -мерный гиперкуб, последовательно разбивается по осям и порождает в памяти ЭВМ древовидную структуру [4] (рис. 2). Каждый последовательный шаг разбиения включает следующие этапы:

- 1) оценку перспективности деления вершин на основе ошибки аппроксимации в них;
- 2) нахождение оптимального порога разбиения признака;
- 3) определение признака, деление по которому дает максимальное уменьшение ошибки;
- 4) деление (образование новых терминальных вершин);
- 5) проход по обучающей выборке и накопление статистик для новых вершин по элементам выборки, попавшим в них;
- 6) построение функции аппроксимации и вычисление ошибки в новых вершинах.

Для того чтобы реализовать шаги 2 и 3, необходимо выполнить шаги 1, 4–6 для всех возможных сочетаний признаков и порогов разбиения.

Также важен при построении такой функции вид функции аппроксимации в терминальных вершинах дерева. В данной работе используется кусочно-линейная аппроксимация (КЛА)

$$\bar{y} = \sum_{i=0}^M a_i x_i + a_0. \quad (2)$$

Оптимальная система коэффициентов  $\mathbf{A} = \{a_k\}_{k=0}^K$ , минимизирующая критерий (1), находится по методу наименьших квадратов за один проход по обучающей выборке:

$$\mathbf{BA} = \mathbf{C}, \quad (3)$$

где

$$\mathbf{B} = \left\{ \sum_{n=0}^{N-1} f_k(\xi(n)) f_l(\xi(n)) \right\}_{k,l=0}^{M-1}; \quad \mathbf{B}_{l,M} = \mathbf{B}_{M,l} = \left\{ \sum_{n=0}^{N-1} f_l(\xi(n)) \right\}_{l=0}^{M-1};$$

$$\mathbf{B}_{M,M} = N; \quad \mathbf{C} = \left\{ \sum_{n=0}^{N-1} f_k(\xi(n)) y(n) \right\}_{k=0}^{M-1}; \quad \mathbf{C}_M = \sum_{n=0}^{N-1} y(n).$$

Таким образом, можно выписать аналитические выражения для  $\{a_0, \dots, a_K\}$  и ошибки аппроксимации  $\varepsilon^2$ :

$$\mathbf{A} = \{a_k\}_{k=0}^K = \mathbf{B}^{-1} \mathbf{C}, \quad \varepsilon^2 = \frac{1}{N} (\mathbf{Y} - \mathbf{C}^T \mathbf{B}^{-1} \mathbf{C}), \quad \mathbf{Y} = \sum_{n=0}^{N-1} y^2(n).$$

Воспользовавшись выражением (3) в каждой из вершин полученного дерева, найдем локальные коэффициенты  $a_i$ .

**Применение методов восстановления регрессии к задачам фильтрации изображений.** При фильтрации изображений часто можно указать желаемый результат обработки, например, в виде согласованной пары изображений, интерпретируемых как «входное»  $X$  и «выходное»  $Y$ . Таким образом, задача фильтрации сводится к задаче аппроксимации функции решения. Общая схема такой обработки изображений представлена на рис. 3. Рассмотрим использование аппроксимации функции решения в такой задаче. Требуется восстановить изображение  $Y$  по  $X$ , однако на практике в результате обработки удается получить результат  $\bar{Y}$  – аппроксимацию  $Y$ . По паре изображений  $Y$  и  $X$ , используемых в качестве обучающей выборки, строится восстанавливающая процедура, состоящая из формирования по  $X$  набора изображений – признаков  $\xi = \{X_1, X_2, \dots, X_K\}$  и построения функции аппроксимации  $\bar{Y} = f(X_1, X_2, \dots, X_K)$ .

Тогда задача фильтрации (получение оператора преобразования  $f^*$ ) сводится к задаче восстановления регрессии, если рассматривать в качестве обучающей выборки отсчеты заданной пары изображений. При этом сначала по искаженному изображению формируется  $K$ -мерный вектор признаков-

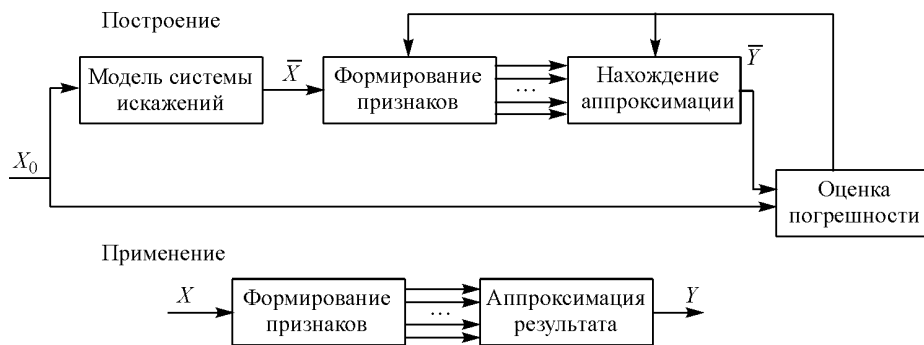


Рис. 3. Общая схема преобразования данных

изображений  $\xi(n_1, n_2) = \{x_k(n_1, n_2)\}_{k=1}^K$ , а затем на его основе вычисляется оценка отсчетов результирующего изображения (см. рис. 3). При построении  $f^*$  по обучающей выборке минимизируется сумма квадратов разностей отсчетов идеального и полученного изображений.

Процедуры формирования признаков и аппроксимации функции решения, настроенные в итоге обучения на неформально заданное преобразование данных, далее используются при обработке других изображений того же класса.

Для ускорения процедуры получения признаков изображения применяется обработка в скользящем окне. Суть обработки изображения в скользящем окне состоит в том, что некоторая ограниченная двумерная область – «окно обработки» – последовательно (например, в порядке построчной развертки) занимает все возможные положения в плоскости изображения, и для каждого положения окна по значениям лежащих в нем входных отсчетов вычисляется значение только одного отсчета выходного изображения.

Пространственно-инвариантная (не зависящая от координат) обработка такого вида описывается общим соотношением

$$y(n_1, n_2) = G[\{x(n_1 - m_1, n_2 - m_2)\}, (m_1, m_2) \in D],$$

где  $x(n_1, n_2), y(n_1, n_2)$  – двумерные последовательности отсчетов входного и выходного изображений соответственно;  $m_1, m_2$  – координаты отсчетов изображения внутри области  $D$ ;  $G$  – оператор преобразования;  $D$  – конечное множество отсчетов, заданное относительно начала координат и определяющее форму и размеры окна обработки. В рамках предлагаемого подхода данное преобразование для каждого положения окна реализуется через формирование  $K$ -мерного вектора признаков  $\{\mu_k(n_1, n_2)\}_{k=0}^{K-1}$  и его использование в процедуре аппроксимации результата для каждого выходного отсчета.

В этом случае рассматриваются только локальные признаки изображения, вычисляемые по окну сравнительно небольшого размера, поэтому их число ограничено. Решение принимается индивидуально для каждого отсчета выходного изображения. Объем обучающей выборки достаточно велик, он равен числу положений скользящего окна на изображении (т. е. самих отсчетов изображения), по которому производится обучение. Таким образом, получаем достаточно большое количество элементов обучающей выборки. При необходимости выборку еще большего (теоретически неограниченного) объема можно получить с помощью компьютерного моделирования тестовых изображений, их «эталонных» преобразований и искажений в системе видеозмерений.

**Экспериментальные исследования.** В качестве исходного материала использовались две последовательности согласованных пар идеальных и зашумленных изображений с разными реализациями шума (отношение сигнал/шум  $D_x/D_y = 5$ ). Тесты проводились на двух различных изображениях размером  $512 \times 512$  отсчетов, представляющих собой портретное изображение и космический фотоснимок.

В первом эксперименте сравним представленные выше способы аппроксимации на небольшом количестве признаков (два признака). В качестве признаков используем зашумленное изображение и зашумленное изображение, сглаженное гауссовским фильтром. Во втором эксперименте увеличим количество признаков до пяти. Добавим производную гауссовского размы-

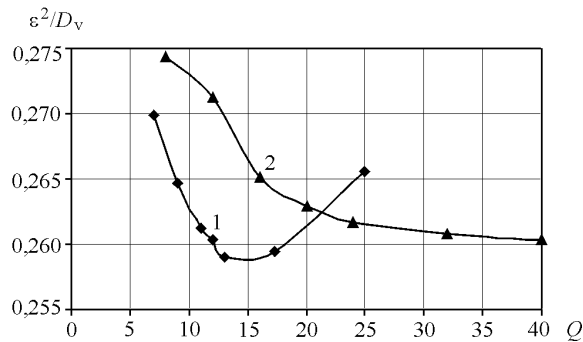


Рис. 4. Сравнение различных типов аппроксимации на двух признаках (кривая 1 – КЛА, 2 – нейросеть)

тия, медианное сглаживание и зашумленное изображение с подчеркнутыми границами. А в третьем эксперименте используем главные моментные инварианты (7 инвариантов).

Для сравнения были взяты методы с одинаковой сложностью решающего правила: взят критерий количества операций на отсчет.

Для нейронной сети количество операций на отсчет  $Q = (M + 2)N + 1$ , где  $M$  – число признаков,  $N$  – количество нейронов в скрытом слое. Для иерархической аппроксимации количество операций на отсчет составляет

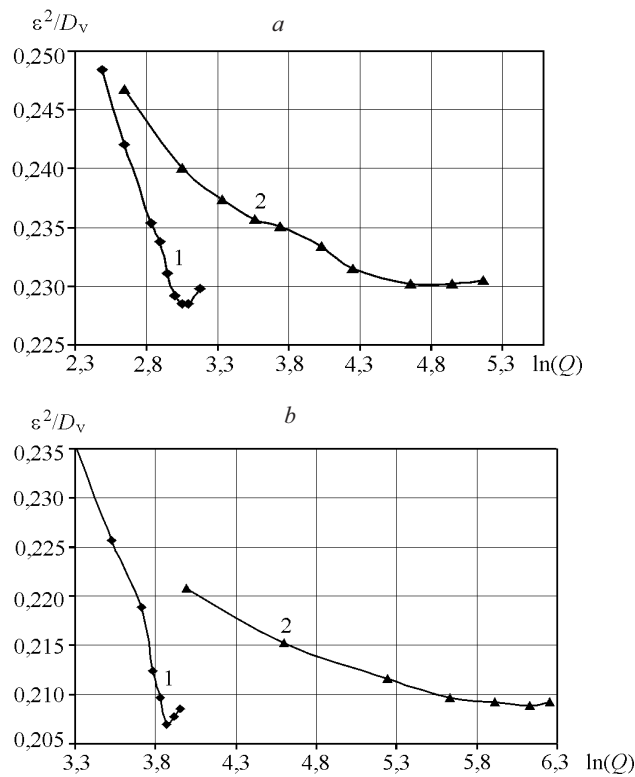


Рис. 5. Сравнение различных типов аппроксимации: на пяти признаках (а) и на семи признаках (б) (кривые 1 – КЛА, 2 – нейросеть)

$Q = H + L$ , где  $H$  – средняя глубина дерева,  $L$  – количество операций в терминальной вершине ( $L = 2M$  операций для КЛА).

Коэффициент понижения шума  $\varepsilon^2/D_v$  использовался как критерий качества фильтрации.

Результаты первого эксперимента на портретном изображении представлены на рис. 4. Результаты второго и третьего экспериментов на космическом фотоснимке приведены на рис. 5, *a, b*.

Таким образом, иерархическая аппроксимация показывает лучшие результаты в отличие от нейросети. Нейронная сеть стабилизируется на сложности около 40–50 опер./отсчет (15 нейронов) в случае двух признаков и на сложности 150–200 опер./отсчет (20–25 нейронов) в случае пяти признаков.

Кроме того, построение функции иерархической аппроксимации происходит значительно быстрее, чем обучение нейронной сети. В среднем построение дерева сложности 25–30 опер./отсчет соответствует 3–10 итерациям обучения подобной по сложности нейронной сети (для нормальной работы нейронной сети необходимо несколько тысяч итераций обучения).

**Заключение.** Из экспериментов следует, что метод иерархической аппроксимации не только обладает лучшим качеством аппроксимации по сравнению с нейронной сетью при равной сложности применения, но и демонстрирует более высокую скорость построения функции аппроксимации.

#### СПИСОК ЛИТЕРАТУРЫ

1. **Krose B., Van der Smagt P.** An introduction to Neural networks. 1996. P. 15.
2. **Короткий С.** Нейронные сети: алгоритм обратного распространения // [http://lil.newmail.ru/NN/KOROTKY/N2/kor\\_nn2.htm](http://lil.newmail.ru/NN/KOROTKY/N2/kor_nn2.htm)
3. **Гашников М. В., Глузов Н. И., Ильясова Н. Ю. и др.** Методы компьютерной обработки изображений /Под ред. В. А. Соифера. М.: Физматлит, 2001. С. 527.
4. **Sergeyev V. V., Chernov A. V.** Image reconstruction methods based on the principles of pattern recognition theory // Pattern Recogn. and Image Analys. 1997. 7, N 4. P. 474.

*Институт систем обработки изображений РАН,  
Самарский государственный  
аэрокосмический университет,  
E-mail: vserg@smr.ru  
vkor@smr.ru*

*Поступила в редакцию  
11 октября 2005 г.*