

УДК 004.032 26



АВТОАССОЦИАТИВНАЯ МНОГОСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ И АЛГОРИТМ ЕЕ ОБУЧЕНИЯ ПРИ СЖАТИИ ИЗОБРАЖЕНИЙ

В.П. Машталир¹, А.Е. Путятин², Е.С. Сакало³

¹ХНУРЭ, г. Харьков, Украина, mashtalir@kture.kharkov.ua

^{2,3}ХНУРЭ, г. Харьков, Украина

Рассмотрена задача понижения размерности вектора признаков, часто встречающаяся при обработке изображений, с помощью гетерогенной автоассоциативной четырехслойной нейронной сети с прямой передачей информации. Предложен алгоритм ее обучения, обладающий улучшенными характеристиками по сравнению с традиционными градиентными процедурами обратного распространения ошибок.

НЕЙРОННАЯ СЕТЬ, СЖАТИЕ ИЗОБРАЖЕНИЙ, АНАЛИЗ ГЛАВНЫХ КОМПОНЕНТ

Введение

При решении широкого класса задач, связанных с обработкой сигналов высокой размерности и, прежде всего, обработкой изображений, достаточно часто возникает необходимость понижения размерности исходных векторов-образов с минимальной потерей информации. Наиболее часто для этого используется аппарат анализа главных компонент и главных подпространств, основанный на линейном преобразовании Карунена-Лоева. Вместе с тем, линейная техника не всегда способна выявить более сложные соотношения, присущие реальным изображениям. В ряде случаев более эффективным представляется нелинейный подход, основанный на нейросетевых технологиях [1].

Так, в [2] была рассмотрена задача сжатия информации с помощью двухслойной нейронной сети с прямой передачей информации и показано, что независимо от вида используемых активационных функций, такая архитектура реализует стандартный анализ главных компонент. В [1] предложена трехслойная автоассоциативная нейронная сеть обратного распространения и описан эксперимент, связанный с обработкой изображений, однако, сравнительный анализ с другими технологиями не проводился. В [3-5] показано, что оптимальное сжатие может быть обеспечено с помощью четырехслойной автоассоциативной нейронной сети с чередующимися линейными и нелинейными слоями и приведены примеры обработки изображений. Данная сеть реализует нелинейный анализ главных компонент (NLPCA) и обучается с помощью стандартного градиентного алгоритма распространения ошибок. Этот алгоритм, однако, обладает низкой скоростью сходимости, не учитывает неоднородности архитектуры сети и весьма чувствителен к воздействию различного рода возмущений.

В связи с этим представляется целесообразным оптимизировать процесс обучения четырехслойной гетерогенной автоассоциативной нейронной сети с прямой передачей информации, предназначенной для понижения размерности обрабатываемых векторов-образов, с учетом влияния помех и возможным нарушением выпуклости целевых функций скрытых слоев.

1. Архитектура нейронной сети

Архитектура рассматриваемой автоассоциативной нейронной сети, предназначенной для понижения размерности пространства исходных образов, приведена на рис. 1 и содержит четыре последовательно соединенных слоя нейронов.

На вход сети (рецепторный нулевой слой) подается последовательность векторов образов

$$x(1), x(2), \dots, x(k), \dots, x(N),$$

$$x(k) = (x_1(k), \dots, x_i(k), \dots, x_n(k))^T \in R^N,$$

предварительно пронормированных на гиперкубу так, что $x_i(k) \in [-1, 1], i = 1, 2, \dots, n$.

Первый скрытый слой содержит $h \geq n$ нейронов – элементарных персептронов Розенблатта с сигмоидальными активационными функциями. В случае отсутствия сигнала смещения первый скрытый слой содержит hn настраиваемых синаптических весов и описываемых соотношениями вида:

$$O_j^{[1]}(k) = \psi_j^{[1]}(u_j^{[1]}(k)) = \psi_j^{[1]}(\sum_{i=1}^n w_{ji}^{[1]} x_i(k)), j = 1, 2, \dots, h,$$

$$O^{[1]}(k) = \Psi^{[1]}(W^{[1]} x(k)),$$

где $O_j^{[1]}(k)$ – выходной сигнал j -го нейрона первого скрытого слоя $x(k)$; $\psi_j^{[1]}(\bullet)$ – активационная функция j -го нейрона первого скрытого слоя; $u_j^{[1]}(k)$ – сигнал внутренней активации j -го нейрона первого скрытого слоя; $w_{ji}^{[1]}$ – синаптический вес i -го входа j -го нейрона первого скрытого слоя; $O^{[1]}(k) = (O_1^{[1]}(k), \dots, O_j^{[1]}(k), \dots, O_h^{[1]}(k))^T$, $\Psi^{[1]} = \text{diag}(\Psi_1^{[1]}, \dots, \Psi_j^{[1]}, \dots, \Psi_h^{[1]})$ – $(h \times h)$ – диагональная матрица активационных функций, $W^{[1]} = \{w_{ji}^{[1]}\}$ – $(h \times n)$ – матрица синаптических весов.

Если нейроны первого скрытого слоя включают сигнал смещения, то приведенные выше соотношения принимают вид:

$$O_j^{[1]}(k) = \psi_j^{[1]}(u_j^{[1]}(k)) = \psi_j^{[1]}(\sum_{i=1}^n w_{ji}^{[1]} x_i(k) + \Theta_j^{[1]}) =$$

$$= \psi_j^{[1]}(\sum_{i=0}^n w_{ji}^{[1]} x_i(k)),$$

$$O^{[1]}(k) = \Psi^{[1]}(W^{[1]} x(k)),$$

где $\Theta_j^{[1]}$ – сигнал смещения j -го нейрона первого скрытого слоя; $w_{j0}^{[1]} = \Theta_j^{[1]}$; $x_0(k) = 1, W^{[1]} - (h \times n + 1)$ – матрица синаптических весов.

$x(k) = (1, x_1(k), \dots, x_n(k))^T$. В этом случае слой содержит $h(n+1)$ настраиваемых параметров.

В качестве активационных функций используются либо логистическая функция

$$\psi_j^{[1]}(u_j^{[1]}) = \frac{1}{1 - e^{-\gamma_j u_j^{[1]}},$$

либо гиперболический тангенс

$$\psi_j^{[1]}(u_j^{[1]}) = \tanh(\gamma_j u_j^{[1]}) = \frac{1 - e^{-2\gamma_j u_j^{[1]}}}{1 + e^{-2\gamma_j u_j^{[1]}}},$$

где γ_j – положительный параметр (возможно настраиваемый), задающий «крутизну» функции. Полезно также ввести в рассмотрение производные логистической функции

$$\frac{\partial \psi_j^{[1]}(u_j^{[1]})}{\partial u_j^{[1]}} = \gamma_j \psi_j^{[1]}(u_j^{[1]}) (1 - \psi_j^{[1]}(u_j^{[1]}))$$

и гиперболического тангенса

$$\frac{\partial \psi_j^{[1]}(u_j^{[1]})}{\partial u_j^{[1]}} = \gamma_j (1 - (\psi_j^{[1]}(u_j^{[1]}))^2)$$

соответственно.

Второй скрытый слой содержит $m < n$ нейронов – адаптивных линейных ассоциаторов, а его выходной сигнал $y = (y_1, \dots, y_m)^T$ является выхо-

дом нейронной сети в целом и представляет собой «сжатый» входной образ x .

В отсутствие смещения второй скрытый слой описывается соотношениями:

$$y_j(k) = \sum_{i=1}^h w_{ji}^{[2]} o_i^{[1]}(k), j = 1, 2, \dots, m,$$

$$y(k) = W^{[2]} O^{[1]}(k),$$

где $W^{[2]} = \{w_{ji}^{[2]}\} - (m \times h)$ – матрица синаптических весов.

Если же нейроны второго скрытого слоя включают сигнал смещения, то приведенные выше соотношения принимают вид:

$$y_j(k) = \sum_{i=1}^h w_{ji}^{[2]} o_i^{[1]}(k) + \Theta_j^{[2]} = \sum_{i=0}^h w_{ji}^{[2]} o_i^{[1]}(k)$$

$$y(k) = W^{[2]} O^{[1]}(k),$$

где $W^{[2]} = \{w_{ji}^{[2]}\} - (m \times (h + 1))$ – матрица синаптических весов: $O^{[1]}(k) = (1, O_1^{[1]}(k), \dots, O_h^{[1]}(k))^T$.

Третий скрытый слой подобен первому и также содержит h элементарных перцептронов Розенблатта. В отсутствие смещения этот слой описывается соотношениями:

$$O_j^{[3]}(k) = \psi_j^{[3]}(u_j^{[3]}(k)) = \psi_j^{[3]}(\sum_{i=1}^m w_{ji}^{[3]} y_i(k)), j = 1, 2, \dots, h,$$

$$O^{[3]}(k) = \Psi^{[3]}(W^{[3]} y(k)),$$

где $W^{[3]} - (h \times m)$ – матрица синаптических весов.

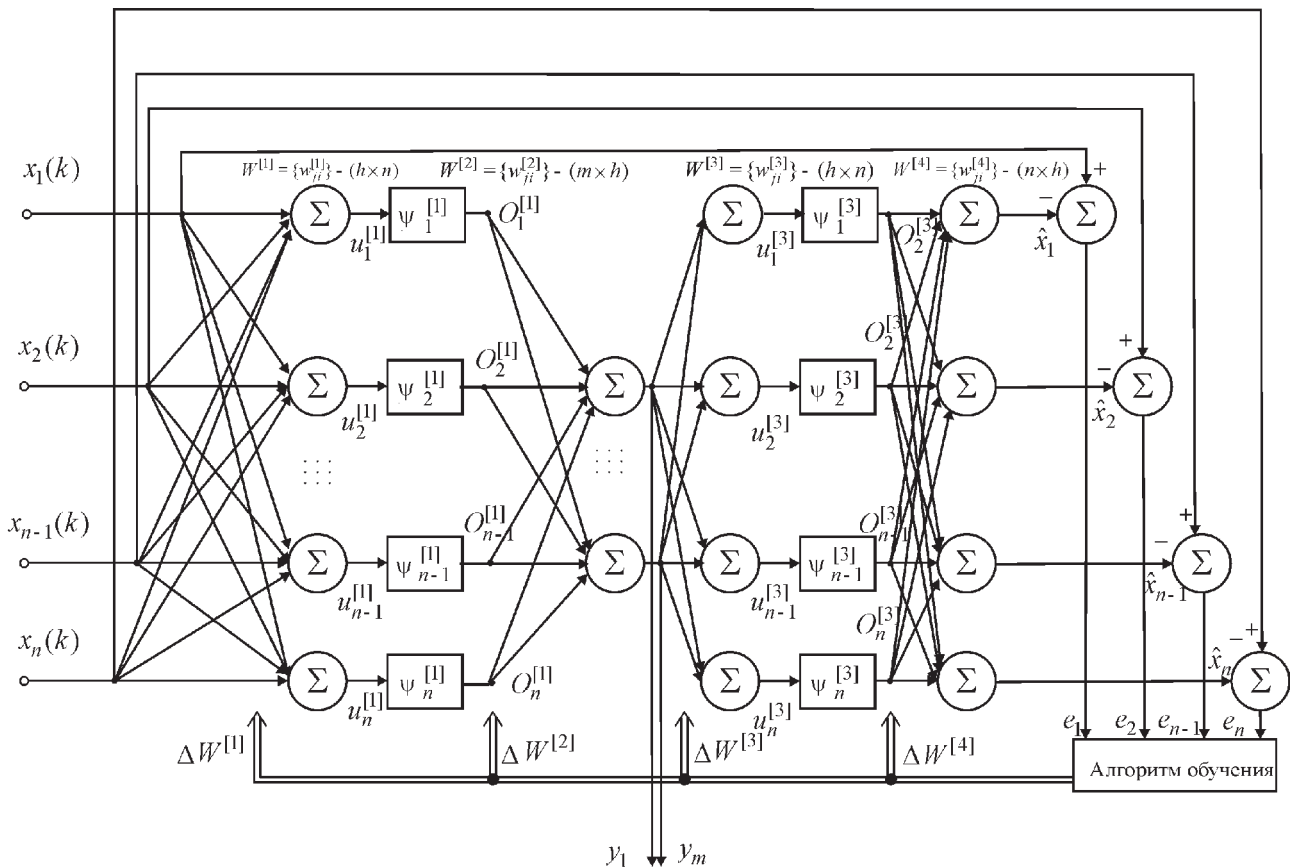


Рис. 1. Автоассоциативная многослойная нейронная сеть

Если же нейроны третьего скрытого слоя включают сигнал смещения, то приведенные выше соотношения имеют вид:

$$\begin{aligned} O_j^{[3]}(k) &= \Psi_j^{[3]}(u_j^{[3]}(k)) = \\ &= \Psi_j^{[3]}(\sum_{i=1}^m w_{ji}^{[3]} y_i(k) + \Theta_j^{[3]}), j = 1, 2, \dots, h, \\ O^{[3]}(k) &= \Psi^{[3]}(W^{[3]}y(k)), \end{aligned}$$

где $W^{[3]}$ – $(h \times (m+1))$ – матрица синаптических весов.

Четвертый выходной слой подобен второму скрытому и содержит n нейронов – адаптивных линейных ассоциаторов. В отсутствие смещения четвертый слой описывается соотношениями:

$$\begin{aligned} \hat{x}_j(k) &= \sum_{i=1}^h w_{ji}^{[4]} o_i^{[3]}(k), j = 1, 2, \dots, n, \\ \hat{x}(k) &= W^{[4]} O^{[3]}(k), \end{aligned}$$

где $\hat{x}(k)$ – $(n \times 1)$ – вектор, являющийся оценкой входного сигнала $x(k)$, восстановленного после сжатия; $W^{[4]}$ – $(n \times h)$ – матрица синаптических весов.

Если адаптивные линейные ассоциаторы четвертого слоя включают сигнал смещения, то приведенные выше соотношения имеют вид:

$$\begin{aligned} \hat{x}_j(k) &= \sum_{i=1}^h w_{ji}^{[4]} o_i^{[3]}(k) + \Theta_j^{[4]} = \sum_{i=1}^h w_{ji}^{[4]} o_i^{[3]}(k), o_0^{[3]}(k) = 1, \\ \hat{x}(k) &= W^{[4]} O^{[3]}(k), \end{aligned}$$

где $W^{[4]}$ – $(n \times (h+1))$ – матрица синаптических весов.

Таким образом, отображение, реализуемое четырехслойной автоассоциативной нейронной сетью, приведенной на рис. 1, имеет вид:

$$\hat{x}(k) = W^{[4]}(\Psi^{[3]}(W^{[3]}W^{[2]}\Psi^{[1]}(W^{[1]}x(k)))).$$

2. Обучение нейронной сети

В качестве критерия обучения будем использовать стандартную квадратичную функцию

$$\begin{aligned} E(k) &= \sum_{j=1}^n E_j(k) = \frac{1}{2} \sum_{j=1}^n (x_j(k) - \hat{x}_j(k))^2 = \\ &= \frac{1}{2} \sum_{j=1}^n e_j^2(k) = \frac{1}{2} \|e(k)\|^2 = \frac{1}{2} \|x(k) - \hat{x}(k)\|^2, \end{aligned}$$

где $e(k) = (e_1(k), \dots, e_n(k))^T$ – вектор ошибок восстановления (обучения) входного сигнала по всем координатам.

Сам же процесс обучения будем проводить на основе градиентной процедуры вида:

$$w_{ji}^{[s]}(k+1) = w_{ji}^{[s]}(k) - \eta^{[s]}(k) \frac{\partial E(k)}{\partial w_{ji}^{[s]}}, s = 1, 2, 3, 4,$$

или

$$\Delta w_{ji}^{[s]} = w_{ji}^{[s]}(k+1) - w_{ji}^{[s]}(k) = -\eta^{[s]}(k) \frac{\partial E(k)}{\partial w_{ji}^{[s]}},$$

где $\eta^{[s]}(k)$ – параметр шага обучения s -го слоя, определяющий скорость процесса настройки.

Следуя концепции обратного распространения ошибок, обучение сети начинается с четвертого (выходного) слоя. При этом:

$$\begin{aligned} \Delta w_{ji}^{[4]} &= -\eta^{[4]}(k) \frac{\partial E(k)}{\partial w_{ji}^{[4]}} = -\eta^{[4]}(k) \frac{\partial E_j(k)}{\partial \hat{x}_j} \cdot \frac{\partial \hat{x}_j}{\partial w_{ji}^{[4]}} = \\ &= \eta^{[4]}(k) e_j(k) O_i^{[3]}(k). \end{aligned}$$

Поскольку выходной сигнал слоя $\hat{x}(k)$ линейно зависит от настраиваемых параметров, для обучения можно использовать либо оптимальный по быстродействию алгоритм Качмажа-Уидроу-Коффа [6-8]:

$$\Delta w_{ji}^{[4]} = \frac{e_j(k) O_i^{[3]}(k)}{\|O^{[3]}(k)\|^2},$$

в векторной форме, имеющий вид:

$$\Delta w_j^{[4]} = \frac{e_j(k) O^{[3]}(k)}{\|O^{[3]}(k)\|^2}, j = 1, 2, \dots, n,$$

либо, если требуется дополнительная фильтрация возмущений и сглаживание, процедуру вида [9]:

$$\begin{cases} \Delta w_j^{[4]} = (p^{[4]}(k))^{-1} e_j(k) O^{[3]}(k), \\ p^{[4]}(k) = \alpha p^{[4]}(k-1) + \|O^{[3]}(k)\|^2, \end{cases} \quad (1)$$

где $0 \leq \alpha \leq 1$ – параметр сглаживания.

Обучение параметров третьего скрытого слоя производится с помощью алгоритма

$$\begin{aligned} \Delta w_{ji}^{[3]} &= -\eta^{[3]}(k) \frac{\partial E(k)}{\partial w_{ji}^{[3]}} = -\eta^{[3]}(k) \frac{\partial E(k)}{\partial u_j^{[3]}} \cdot \frac{\partial u_j^{[3]}}{\partial w_{ji}^{[3]}} = \\ &= \eta^{[3]}(k) \delta_j^{[3]}(k) y_i(k), \end{aligned} \quad (2)$$

где $\delta_j^{[3]}(k) = \frac{\partial E(k)}{\partial u_j^{[3]}}$, $j = 1, 2, \dots, h$ так называемая начальная ошибка (δ -ошибка) третьего скрытого слоя, которая также может быть представлена в форме

$$\delta_j^{[3]}(k) = -\frac{\partial E(k)}{\partial O_j^{[3]}(k)} \cdot \frac{\partial O_j^{[3]}(k)}{\partial u_j^{[3]}}$$

с учетом того, что

$$\partial O_j^{[3]}(k) = \Psi_j^{[3]}(u_j^{[3]}(k))$$

логично записать

$$\delta_j^{[3]}(k) = -\frac{\partial E(k)}{\partial O_j^{[3]}(k)} \cdot \frac{\Psi_j^{[3]}(u_j^{[3]}(k))}{\partial u_j^{[3]}},$$

а представив $-\frac{\partial E(k)}{\partial O_j^{[3]}(k)}$ в виде

$$-\frac{\partial E(k)}{\partial O_j^{[3]}(k)} = -\sum_{i=1}^n \frac{\partial E(k)}{\partial \hat{x}_i(k)} \cdot \frac{\partial \hat{x}_i(k)}{\partial O_j^{[3]}} =$$

$$= \sum_{i=1}^n \left(-\frac{\partial E(k)}{\partial \hat{x}_i(k)}\right) \cdot \frac{\partial}{\partial O_j^{[3]}} \left(\sum_{p=1}^h w_{ip}^{[4]}(k) O_p^{[3]}(k)\right) =$$

$$= \sum_{i=1}^n e_i(k) \cdot \frac{\partial}{\partial O_j^{[3]}} \left(\sum_{p=1}^h w_{ip}^{[4]}(k) O_p^{[3]}(k)\right) = \sum_{i=1}^n e_i(k) w_{ij}^{[4]}(k),$$

получить окончательное выражение:

$$\delta_j^{[3]}(k) = \frac{\partial \Psi_j^{[3]}(u_j^{[3]}(k))}{\partial u_j^{[3]}} \sum_{i=1}^n e_i(k) w_{ij}^{[4]}(k).$$

Тогда процедура (2) приобретает форму:

$$\begin{aligned} \Delta w_{ji}^{[3]} &= \eta^{[3]}(k) \delta_j^{[3]}(k) y_i(k) = \\ &= \eta^{[3]}(k) y_i(k) \frac{\Psi_j^{[3]}(u_j^{[3]}(k))}{\partial u_j^{[3]}} \sum_{i=1}^n e_i(k) w_{ij}^{[4]}(k), \end{aligned}$$

или в векторном виде:

$$\Delta w_j^{[3]} = \eta^{[3]}(k) \delta_j^{[3]}(k) y(k). \quad (3)$$

Свойства общей процедуры (3) определяются выбором параметра $\eta^{[3]}(k)$, от обоснованного выбора которого зависит время обучения в целом.

Введем в рассмотрение модификацию регуляризованного алгоритма обучения [10, 11] вида:

$$\Delta w_j^{[3]} = \frac{\eta^{[3]}(k) \delta_j^{[3]}(k) y(k) + \rho^{[3]} \Delta w_j^{[3]}(k-1)}{\|y(k)\|^2}, \quad (4)$$

где $\eta^{[3]} = const > 0, 0 \leq \rho < 1$ – параметр регуляризации.

Несложно заметить, что при $\rho^{[3]} = 0$ процедура приобретает форму нелинейной модификации мультипликативного алгоритма Качмажа, характеризующегося высоким быстродействием. Если же требуется дополнительная фильтрация, можно от процедуры (4) перейти к процедуре общего вида:

$$\begin{cases} \Delta w_j^{[3]} = (p^{[3]}(k))^{-1} (\eta^{[3]}(k) \delta_j^{[3]}(k) y(k) + \Omega_3), \\ p^{[3]}(k) = \alpha p^{[3]}(k-1) + \|y(k)\|^2, \end{cases} \quad (5)$$

где $\Omega_3 = \rho^{[3]} \Delta w_j^{[3]}(k-1)$, объединяющей в себе быстродействие алгоритма Качмажа-Уидроу-Коффа, сглаживающие свойства алгоритма, введенного в [12], и возможность проходить «плоские» участки целевой функции присущие регуляризованным алгоритмам с $\rho^{[3]} > 0$.

Аналогичным образом можно записать алгоритм обучения второго скрытого слоя в виде:

$$\Delta w_{ji}^{[2]} = \eta^{[2]}(k) \delta_j^{[2]}(k) O_i^{[1]}(k),$$

где

$$\delta_j^{[2]}(k) = \sum_{i=1}^h \delta_i^{[3]}(k) w_{ij}^{[3]}(k).$$

Окончательно же:

$$\begin{cases} \Delta w_j^{[2]} = (p^{[2]}(k))^{-1} (\eta^{[2]}(k) \delta_j^{[2]}(k) O^{[1]}(k) + \Omega_2), \\ p^{[2]}(k) = \alpha p^{[2]}(k-1) + \|O^{[1]}(k)\|^2, \end{cases} \quad (6)$$

где $\Omega_2 = \rho^{[2]} \Delta w_j^{[2]}(k-1)$.

И, наконец, для первого скрытого слоя справедливы соотношения:

$$\Delta w_{ji}^{[1]} = \eta^{[1]}(k) \delta_j^{[1]}(k) x_i(k),$$

$$\delta_j^{[1]}(k) = \frac{\partial \Psi_j^{[1]}(u_j^{[1]}(k))}{\partial u_j^{[1]}} \sum_{i=1}^h \delta_i^{[2]}(k) w_{ij}^{[2]}(k).$$

$$\begin{cases} \Delta w_j^{[1]} = (p^{[1]}(k))^{-1} (\eta^{[1]}(k) \delta_j^{[1]}(k) x(k) + \Omega_1), \\ p^{[1]}(k) = \alpha p^{[1]}(k-1) + \|x(k)\|^2, \end{cases} \quad (7)$$

где $\Omega_1 = \rho^{[1]} \Delta w_j^{[1]}(k-1)$.

Таким образом, процесс обучения нейронной сети, показанной на рис. 1, сводится к последовательному использованию процедур (1), (5), (6) и (7) для каждого предъявляемого вектора-образа $x(k)$. Для достаточно длинной обучающей выборки процесс настройки может быть реализован в реальном времени. Если же объем исходных данных ограничен, обучение может быть организовано по эпохам путем многократного «прогона» данных через нейронную сеть.

3. Результаты эксперимента

Для анализа качества компрессии изображений, получаемых при помощи построенной нейронной сети, был взят критерий, основанный на среднеквадратичной ошибке значений пикселей (mean square error – MSE):

$$d(x, y) = \sqrt{\sum_{i,j=1}^n ((x_{i,j} - y_{i,j})^2 / n^2)}. \quad (8)$$

Он подразумевает сравнение исходного изображения и восстановленного после сжатия. Так как предложенный метод сжатия следует отнести к сжатию с потерями, то естественно желание выяснить степень потерь. Однако, следует отметить, что данный критерий дает правильные результаты при отсутствии шума на изображении. При наличии же шума получим серьезные ошибки для данного критерия. В такой ситуации лучше использовать критерий отношения пикового значения сигнала к шуму (PSNR – peak signal-to-noise ratio), который является более устойчивым к подобным воздействиям.

В результате сравнения исходных изображений и после применения предлагаемого метода сжатия, использующего представленную четырехслойную нейронную сеть на основе критерия (8) получены результаты, представленные на рис. 2.

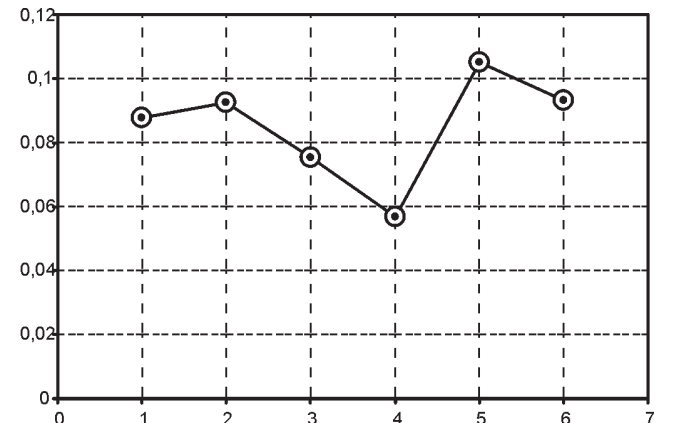


Рис. 2. Среднеквадратическое отклонение пикселей изображений, полученное при сжатии

Таким образом, полученные результаты свидетельствуют о том, что приведенный метод сжатия приводит к потерям информации в районе 8-9%, что для методов сжатия с потерями является приемлемым результатом.

Выводы

Рассмотрена задача сжатия данных высоких размерностей с помощью автоассоциативной гетерогенной четырехслойной нейронной сети типа многослойного персептрона. Предложена процедура ее обучения на основе обратного распространения ошибки, характеризующаяся высоким быстродействием, сглаживающими и следящими свойствами, введением дополнительной регуляризации и численного критерия. По сравнению с процедурами сжатия на основе преобразования Карунена-Лоэва предложенный нейросетевой подход позволяет выявить более «тонкие» эффекты, присутствующие в реальном изображении.

Список литературы: 1. *Dony R.D., Haykin S.* Neural network approaches to image compression // Proc. IEEE. – 1995. – 83. – P. 288-303. 2. *Bishop C.M.* Neural network for pattern recognition. – Oxford: Clarendon Press, 1995. – 482p. 3. *Kramer M.L.* Nonlinear principal component analysis using autoassociative neural networks / AICHE J. – 1991. – 32. – No 2. – P.233-243. 4. *Tan S., Mavrouniotis M.* Reducing data dimensionality through optimizing neural-network inputs // AICHE J. – 1995. – 41. – No 6. – P. 1471-1480. 5. *Malthouse E.C.* Limitations of nonlinear PCA as performed with generic neural networks // IEEE Trans. on Neural Networks. – 1998. – 9. – P. 165-173. 6. *Kaczmarz S.* Approximate solution of system of linear equations // Int. J. Control. – 1993. – 57. – No 6. – P. 1269-1271. 7. *Widrow B., Hoff Jr. M.E.* Adaptive switching circuits // 1960 IRE Western Electric Show and Con-

nection Record. – 1960. – Part 4. – P. 96-105. 8. *Widrow B., Lee M.* 30 years of adaptive neural networks perceptron, adaptive and backpropagation // Proc. IEEE. – 1990. – 78. – P. 1415-1442. 9. *Бодянский Е.В., Плисс И.П., Соловьева Т.В.* Многошаговые оптимальные упредители многомерных нестационарных стохастических процессов // Докл. АН УССР. – 1986. – Сер. А. – № 12. – С. 47-49. 10. *Cichocki A., Unbehauen R.* Neural networks for optimization and signal processing. – Stuttgart: Feubner, 1993. – 526p. 11. *Haykin S.* Neural networks. A comprehensive foundation. – Upper saddle river, N.J.: Prentice Hall, Inc., 1999. – 842p. 12. *Otto P., Bodyanskiy Ye., Kolodyazhniy V.* A new learning algorithm for a forecasting neuro-fuzzy network // Integrated Computer-Aided Engineering. – 2003. – 10. – No 4. – P399-405.

Поступила в редколлегию 3.09.2008

УДК 004.032 26

Автоасоціативна багатосарова нейронна мережа та алгоритм її навчання при стисненні зображень / В.П. Машталір, О.Є. Пуяткіна, Є.С. Сакало // Біоніка інтелекту: наук.-техн. журнал – 2008. – № 2 (69). – С. 40-44.

Запропоновано та розглянуто побудову багатосарової нейронної мережі при розв'язанні задачі стиснення зображень. Також запропоновано алгоритм її навчання та проведено аналіз отриманих результатів.

Л. 2. Бібліогр.: 12 найм.

UDC 004.032 26

Autoassociative multi-layer neural network and educational algorithm for image compression / V.P. Mashtalir, A.E. Putiatina, E.S. Sakalo // Bionics of Intelligence: Sci. Mag. – 2008. – № 2 (69). – P. 40-44.

Multi-layer neural network for image compression are proposed. Educational algorithm are offered and results analysis are conducted.

Fig. 2. Ref.: 12 items.