УДК 519.71

О.Г. Руденко, Р.В. Бобнев, А.А. Бессонов

Использование нормализующей компоненты при нейросетевом сжатии изображений

Предложены модификации нейронных сетей, используемых для сжатия изображений. Приведены результаты моделирования, свидетельствующие об эффективности описанной модификации.

A modification of neural networks for image compression is suggested in the article. Simulation results that demonstrate the effectiveness of the proposed modification are presented.

Запропоновано модифікації нейронних мереж, які використовуються для стискання зображень. Подано результати моделювання, які свідчать про ефективність описаної модифікації.

Введение. Непрерывное возрастание объемов перерабатываемой, в частности, зрительной, информации обусловливает необходимость ее компактного представления. Применение с этой целью традиционных методов, использующих префиксное или арифметическое кодирование либо обеспечивающих сжатие без потерь, требует значительных вычислительных ресурсов. В то же время наличие в мультимедийной информации операции дискретизации изображения и звука позволяет ставить и решать задачу эффективного сжатия информации с потерями.

В этих условиях перспективным представляется развитие подхода, в основе которого лежат искусственные нейронные сети (ИНС). При этом ИНС могут использоваться как при сжатии без потерь, так и при реализации сжатия с потерями, как, например, в стандарте *JPEG* 2000, основанном на вейвлет-преобразовании.

Наиболее значительные результаты в этом направлении получены на основе самоорганизующихся карт Кохонена [1] и сетей встречного распространения, также содержащих слой нейронов Кохонена [2, 3]. Самоорганизация в этих сетях представляет собой процесс кластеризации образов, осуществляемой аналогично методу главных компонент.

Модификация сети Кохонена

Если сеть Кохонена используют для разбиения всех пикселей изображения на классы, то сеть Гроссберга применяется, как правило, для «конвертации» номера кластера (нейрона победителя) в соответствующий цвет пикселя. Следует, однако, отметить, что поскольку на входной слой сети подаются соответствующие цветовые компоненты всех пикселей, то веса каждого нейрона–победителя будут представлять собой некий опорный образец для каждого класстера. При обучении сети Гроссберга по сути подбирается наиболее подходящий по цвету пиксель к номеру кластера. Отбросить же сеть Гроссберга и непосредственно использовать веса сети Кохонена не представляется возможным, так как сеть Кохонена практически неприменима, если входные данные не нормализованы. Эта нормализация гарантирует, что процесс обучения приведет к связному разделению пространства данных. Однако такой предобработке данных присущ недостаток, заключающийся в том, что после обучения сети веса каждого из нейронов нельзя использовать как образы–представители соответствующего класса (так как веса тоже нормализованы и процедура получения опорного образца невозможна).

В работе [4] предложена модификация сети Кохонена, позволяющая напрямую использовать значения весов сети в качестве представителей кластеров.

Данная модификация использует уравнивание длин всех входных векторов к одной величине, но не к единице, как в классической сети Кохонена, а к некоторому другому значению. Это осуществляется путем использования дополнительной компоненты входного сигнала X_{mod}, вычисляемой так:

$$X_{\mathrm{mod}} = \sqrt{A_{\max} - \sum_{i=0}^{n} X_i^2}\,, \tag{1}$$

где $A_{\max}$ – максимально представимое число либо максимально допустимая разрядность числа вычислительного устройства или АЦП.

Очевидно, что в данном случае для вычисления нейрона–победителя можно также использовать выходы сети, а не вычислять евклидово расстояние. При этом операция по вычислению дополнительного входа будет занимать меньше времени, чем операция их нормализации, так как на каждом шаге обучения вычисления проводятся только для одного входа, а не для каждого, как в обычной нормализации. Хотя добавление дополнительного входа увеличивает время вычисления выхода, это все же быстрее, чем вычисление аналогичного евклидова расстояния.

Следует отметить, что такая модификация не влечет за собой никаких изменений в способе обучения сети, так как для сети, по сути, лишь добавился еще один вход, поэтому обучение модифицированной сети Кохонена осуществляется по стандартному алгоритму:

Шаг 1. Инициализация весов сети.

Шаг 2. Вычисление взвешенных сумм для каждого выходного нейрона на основе его входных значений.

Шаг 3. Определение нейрона–победителя (нейрона с максимальным выходным значением).

Шаг 4. Коррекция весов нейрона–победителя по формуле

$$w_i(t+1) = w_i(t) + \alpha(t)\cdot(X_i(t) - w_i(t)),\ i = \overline{0,n}, \quad (2)$$

где w_i – вес i-го входа нейрона–победителя; $\alpha(t)$ – параметр, влияющий на скорость обучения сети; X_i – значение i-го входа.

Шаг 5. Формирование нового входного вектора и подача его на вход сети, после чего переход к п. 2, продолжение обучения до тех пор, пока не будет достигнуто необходимое число шагов обучения или обучающих образов.

Предложенная модификация может быть применена практически к любым существующим реализациям сети Кохонена.

Очевидно, что после обучения сети, веса каждого из нейронов (за исключением веса, связанного с дополнительным входом) можно использовать как вектор–представитель класса. Таким образом, после данной модификации, сеть Кохонена может быть применена для решения задач, которые решались только с применением сетей векторного квантования.

Модификация сети «Нейро-Газ»

Очевидно, что такая модификация может быть легко применима к известной сети «Нейро-Газ», имеющей такую же структуру, как и сеть Кохонена, и отличающаяся от последней способом обучения [5–7]. В сети «Нейро-Газ» после каждого предъявления образа все нейроны сети сортируются по значению близости весов текущего нейрона к предъявленному образу, характеризующейся, как правило, евклидовым расстоянием между векторами входного образа и весами нейрона. Алгоритм подстройки весов сети выглядит так:

$$w(t+1) = w(t) + \alpha(t) * e^{-k/\lambda(t)}(X - w(t)), \quad (3)$$

где $\alpha(t)$ – параметр, влияющий на скорость обучения сети, аналогичный параметру $\lambda(t)$ в (2); k – индекс нейрона в отсортированном массиве после предъявления входного образа; $\lambda(t)$ – параметр, определяющий степень влияния удаленности нейрона от входного образа для текущего момента времени, аналогичный параметру $\alpha(t)$.

Модификация радиально-базисной сети (РБС)

РБС осуществляют аппроксимацию функции $f(\boldsymbol{x})$ некоторой системой базисных функций (БФ) – нелинейных функций $\varphi(\boldsymbol{x}, \boldsymbol{t})$, зависящих от расстояния (радиального) $r = \| \boldsymbol{x} - \boldsymbol{t} \|$, где $\boldsymbol{t}$ – вектор центров БФ. Представление нелинейной функции радиально-базисной сетью имеет вид

$$\hat{f}(k) = a_0 + \sum_{i=1}^{N} w_i \Phi_i\left(\boldsymbol{x}(k), \mu, \sigma\right), \quad `(4)$$

где a_0 – смещение нейрона выходного слоя; w_i – вес связи i-го нейрона скрытого слоя с нейроном выходного слоя; N – число нейронов в скрытом слое; Φ_i – БФ i-го нейрона.

Выбор БФ важен при построении РБС, так как существенно влияет на сложность вычислений, а для упрощения обычно предполагается, что активационные функции нейронов одинаковы и выбраны экспертом. В РБС в качестве БФ могут быть использованы, например, следующие функции:

$$\Phi_i(x) = \exp\left\{-\frac{\|x-\mu_i\|^2}{\sigma_i^2}\right\}, \qquad (5)$$

$$\Phi_i(x) = \left(1-\frac{(x-\mu_i)^2}{\sigma_i^2}\right)e^{-\frac{(x-\mu_i)^2}{\sigma_i^2}}, \qquad (6)$$

$$\Phi_i(x) = \exp\left\{-\frac{|x-\mu_i|}{\sigma_i}\right\}, \qquad (7)$$

$$\Phi_i(x) = \frac{2(x-\mu_i)}{\sigma_i}\exp\left\{-\frac{(x-\mu_i)^2}{\sigma_i^2}\right\}, \qquad (8)$$

где μ_i, σ_i – центры и радиусы базисных функций соответственно; $\|\cdot\|$ – евклидова норма.

Существует много методов определения структуры РБС (количества и вида БФ). Например, изменение структуры сети может осуществляться, как в [8–10]. На первом шаге структура сети содержит один нейрон с выбранной базисной (активационной) функцией. Затем подается обучающая последовательность, и, в зависимости от реакции сети, характеризуемой ошибкой ***e*****,** происходит либо ее обучение, либо добавление нового нейрона.

При этом изменение структуры путем добавления очередного нейрона происходит всякий раз, когда появление нового входного сигнала приводит к возникновению ошибки e, превышающей допустимую. Если в l-й момент времени сеть содержала N нейронов, а появление сигнала $x(l)$ привело к появлению ошибки $e(l) > e_{\text{доп}} = \varepsilon$, то в сеть вводится новый, (N+1)-й, нейрон, центр БФ μ_{N+1} которого принимается равным $\mu_{N+1} = x(l)$, вес – $w_{N+1} = e(l)$, и $\sigma_{N+1} = \| x(l) - \mu_m(l) \|$, где $\mu_m(l)$ – центр БФ для l-го входа сигнала. Таким образом, условием введения нового нейрона есть выполнение неравенств

$$e(l) > \varepsilon\,, \qquad (9)$$

$$\|x(l)-\mu_m(l)\| > \rho, \qquad (10)$$

где ε и ρ – априорно устанавливаемые предельно допустимые значения ошибки реакции сети и отклонения обобщенного сигнала x_l от ближайшего к данному входу центра.

Однако в задаче сжатия информации структура может быть выбрана фиксированной в зависимости от требуемого коэффициента сжатия.

Достаточно широко распространен рекуррентный алгоритм метода наименьших квадратов (РМНК) с экспоненциальным взвешиванием информации вида

$$w(k) = w(k-1) + \frac{P(k-1)\nabla f(k)}{\lambda + \nabla f^T(k)P(k-1)\nabla f(k)}e(k), \qquad (11)$$

$$P(k) = \frac{1}{\lambda}\left(P(k-1) - \frac{P(k-1)\nabla f(k)\nabla f^T(k)P(k-1)}{\lambda + \nabla f^T(k)P(k-1)\nabla f(k)}\right), \qquad (12)$$

где

$$\nabla f(k) = \Big[1,\ \Phi_1(x(k)),\ 2\Phi_1(x(k))c_1\sigma_1^{-2}(x(k)-\mu_1)^T,$$
$$2\Phi_1(x(k))c_1\sigma_1^{-3}\|x(k)-\mu_1\|^2,$$
$$\ldots,\ \Phi_N(x(k)),\ 2\Phi_N(x(k))c_N\sigma_N^{-2}(x(k)-\mu_N)^T,$$
$$2\Phi_N(x(k))c_N\sigma_N^{-3}\|x(k)-\mu_N\|^2\Big]^T; \quad \lambda\in(0,\ 1].$$

Как видно из (11), (12), свойства алгоритма существенно зависят от значения коэффициента взвешивания λ, общих рекомендаций по выбору оптимального значения которого в настоящее время нет. Предлагаемые же различными авторами процедуры выбора $\lambda^{\text{опт}}$ требуют достаточно много, обычно недоступной в реальных условиях, информации. Поэтому на практике чаще всего используют значения $\lambda = 0{,}995 \div 0{,}999$.

Хотя РБС работает с нормализованными данными лучше, в задаче сжатия изображений нормализация входных данных не представляется возможной, так как сеть восстановит нормализованное изображение, что приведет к потере качества сжатия. Однако РБС также может быть использована для сжатия изображений, но уже не путем кластеризации пикселей, а методом, известным как «бутылочное горлышко». Суть его состоит в том, что нейронная сеть имеет скрытый слой меньшей размерности, чем размерности входного и выходного слоев, которые, как правило, одинаковы.

Собственно сжатие осуществляется благодаря тому, что вместо данных изображения для каждого блока изображения сохраняются значения выходов скрытого слоя. Таким образом, сеть восстановит весь входной вектор, включая нормализующую компоненту. На выходе же для получения реального исходного значения входного вектора эта компонента должна быть отброшена.

Итак, по аналогии с рассматриваемыми сетями, РБС следует модифицировать путем добавления кроме нормализующей компоненты входного вектора, нормализующую компоненту выходного сигнала. Тогда сеть будет пытаться восстановить входной вектор, который на одну размерность больше, чем реальный, но при этом нормализован.

Моделирование

При моделировании процесса сжатия для оценки качества сжатия использовались показатели *PSNR* (*peak signal–to–noise ratio*,) – отношение максимально возможного уровня сигнала к уровню искажающего его шума (поскольку большинство сигналов имеют очень широкий динамический диапазон, *PSNR* обычно представляют в логарифмическом масштабе), и *MSE* (*mean squared error*) – среднеквадратичная ошибка.

Для двух монохромных изображений I и K размерностью $m \times n$ (где одно из изображений зашумлено представлением второго) *MSE* вычисляется так:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left\| I(i,j) - K(i,j) \right\|^2 .$$

Для цветных изображений с тремя *RGB*-компонентами *MSE* определяется как сумма всех квадратичных разностей, деленная на размер изображения и на три.

Показатель *PSNR* определяется так:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right),$$

где MAX_I представляет собой максимальное значение пикселя в изображении. В случае, если пиксели представлены восьмибитовыми значениями, $MAX_I = 255$. В общем случае при использовании для представления B бит максимально возможное значение для MAX_I равно $2^B - 1$.

Обычно для алгоритмов сжатия *PSNR* находится в диапазоне 30–40 *dB*.

В эксперименте использовано эталонное изображение «*Lenna*» (рис. 1,*а*) размером 512×512 пикселей, размеры кодируемой области выбирались 1×1 и 4×4 пикселей.

При моделировании модифицированной сети Кохонена использовано 128 нейронов. В алгоритме обучения (шаг 2) выбиралось $\alpha(t) = \alpha_0 \left(\alpha_f / \alpha_0\right)^{t/t_{\max}}$ (в данном эксперименте α_0 – начальное значение (1,0); α_f – конечное значение (0,005)). При исследовании сети «Нейро-Газ» также использовалось 128 нейронов, в алгоритме обучения (шаг 3) $\alpha(t)$ применялось как в сети Кохонена, а правило изменения $\lambda(t)$ было аналогично правилу для $\alpha(t)$, за исключением лишь начальных параметров (для данного эксперимента $\lambda_0 = 10$; $\lambda_f = 0{,}01$ выбраны экспериментально).

Рис. 1. *а* – эталонное изображение «*Lenna*» и результаты сжатия изображения сетью: *б* – встречного распространения; *в* – сетью РБФ; *г* – сетью «Нейро-Газ» с нормализующим входом

При исследовании работы модифицированной РБС входной и выходной слои содержали по 48 нейронов, а количество нейронов скрытого слоя – 15 выбрано экспериментально на основании правила, что число нейронов скрытого слоя должно быть гораздо меньше. Количество циклов обучения составляло 100. В качестве базисной функции для РБС использовалась функция (5), а обучение осуществлялось с помощью алгоритма (11) – (12).

Таблица 1. Сравнение значений коэффициент *PSNR* для красной (*R*), зеленой (*G*) и синей (*B*) составляющих изображения соответственно, а также среднее значение по компонентам.

Название компоненты	Значение коэффициента *PSNR*					
	СВР	МСК	РБС	МРБС	НГ	МНГ
R	27,16	34,98	21,46	25,72	35,26	34,36
G	30,09	35,50	24,40	26,25	35,48	34,41
B	31,87	34,88	24,14	25,75	34,93	33,68
Среднее по компонентам	29,27	35,12	23,33	25,91	35,23	34,15

Таблица 2. Сравнение времени, затраченного на процесс компрессии и декомпрессии.

Название сети	Время, затраченное на процесс компрессии и декомпрессии	Количество настраиваемых весов	Количество циклов обучения сети
Для области 1x1 пиксель			
Сеть встречного распространения	00:06,06	768	2
Модифицированная сеть Кохонена	00:05,01	512	2
Сеть РБФ	02:04,30	180	100
Модифицированная сеть РБФ	02:32,71	240	100
Для области 4x4 пикселя			
Сеть встречного распространения	00:05,01	12288	2
Модифицированная сеть Кохонена	00:03,39	8192	2
Сеть РБФ	01:36,80	2880	100
Модифицированная сеть РБФ	01:39,82	3840	100
Сеть «Нейро-Газ»	00:37,77	384	2
Модифицированная сеть «Нейро-Газ»	00:42,29	512	2

На рис. 1,*б* показаны некоторые результаты сжатия изображения «*Lenna*» предложенными модифицированными сетями. Поскольку полученные результаты визуально мало чем отличаются, для большей наглядности они сведены в табл. 1 и 2. Здесь использованы следующие сокращения: СВР – сеть встречного распространения, МСК – модифицированная сеть Кохонена, РБС – радиально-базисная сеть, МСБР – модифицированная РБС, НГ – сеть «Нейро-Газ», МНГ – модифицированная НГ. Как следует из результатов моделирования, введение дополнительного нормализующего входа в ряде случаев позволяет существенно увеличить качество сжатия информации. Для сети «Нейро-Газ» результаты сжатия с нормализующим входом по качеству несколько хуже, скорее всего это связано с тем, что веса всех (включая добавленный) нейронов участвуют в вычислении расстояния между вектором весов и входным образом.

Заключение. Суть предложенной модификации нейронных сетей, используемой для сжатия изображений состоит в использовании в сетях дополнительного «нормализующего» входа. В РБС такая модификация обеспечивает повышение качества изображения после декомпрессии. Хотя при этом и возрастает число входов (на единицу), а следовательно, увеличивается количество настраиваемых параметров (на три), это несущественно в вычислительном процессе. Описанная модификация сети встречного распространения позволяет не только убрать слой Гроссберга (по сути оставив лишь одну сеть Кохонена), но и перенести вычисления по нормализации входных данных на формирование значения для дополнительного входа. Несмотря на то, что данная модификация сети «Нейро-Газ» не обеспечила качественное сжатие, она позволяет изменить способ обучения этой сети путем замены вычисления расстояния между вектором весов и исходным образом более простым вычислением выходного значения. Использование выходов сети «Нейро-Газ» в качестве близости вектора весов к исходному образу не представляется возможным, так как входные данные не были нормализованными, что привело бы к постоянному доминированию выхода с наибольшими значениями весов.

1. *Kohonen T.* Self-Organising Maps. – Berlin: Springer Verlag, 1995. – 362 p.

Окончание на стр. 37

2. *Хайкин С.* Нейронные сети: Полный курс. – М.: Вильямс, 2006. – 1104 с.
3. *Бодянский Е.В.*, *Руденко О.Г.* Искусственные нейронные сети: архитектура, обучение, применения. – Харьков: Телетех, 2004. – 372 с.
4. *Руденко О.Г.*, *Бобнев Р.В.* Об одной модификации сети Кохонена // Вестн. Херсон. нац. техн. ун-та. – 2000. – № 2(**38**). – С. 117–119.
5. *Martinetz T.M.*, *Schulten K.J.* A "neural-gas" network learns topologies // Artificial Neural Networks. – North-Holland: Elsevier Science Publ., 1991. – P. 397–402.
6. *Martinetz T.M.*, *Berkovich S.G.*, *Schulten K.J.* "Neural-gas" network for vector quantization and its application to time series prediction // IEEE Trans. on Neural Networks. – 1993. – N 4(**4**). – P. 558–569.
7. *Fritzke B.* A growing neural gas network learns topologies // Advances in Neural Inform. Processing Syst. – Cambridge MA, MIT Press, 1995. – **7**. – P. 625–632.
8. *Li Y.*, *Sundararajan N.*, *Saratchandran P.* Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems // IEE Proc. – Control Theory Appl. – 2000. – **147**, № 4. – P. 476–484.
9. *Руденко О.Г.*, *Бессонов А.А.* Адаптивное управление многомерными нелинейными объектами на основе радиально-базисных сетей // Кибернетика и системный анализ. – 2005. – № 2. – С. 168–176.
10. *Аппроксимация* гауссовских базисных функций в задаче адаптивного управления нелинейными объектами / О.Г Руденко, А.А. Бессонов, А.С. Ляшенко и др. // Там же. – 2011. – № 1. – С. 3–13.

Поступила 12.12.2012

Тел. для справок: +38 057 705-0862, 702-1354,
+38 095 475-2171, 651-5503, +38 067 971-2782 (Харьков)
E-mail: *o.bezsonov@adbglobal.com*, *fatumr@rambler.ru*

●