УДК 621.396

Сжатие изображений с помощью тензорной аппроксимации

М.К. Чобану, Д.В. Макаров

Национальный исследовательский университет «МЭИ», MakarovDmV@gmail.com

***Аннотация* — В статье рассматривается применение метода тензорной аппроксимации многомерных массивов для обработки изображений. Показана его эффективность и перспективы применения для сжатия изображений.**

***Ключевые слова* — тензор, тензорная аппроксимация, сжатие с потерями, аппроксимации цепочкой тензоров, сигнально-зависимые фильтры.**

I. Введение

Тензорный анализ и теория тензорных аппроксимаций играют все более важную роль в области вычислительной математики и численного анализа.

Эффективное представление d-мерного тензора (массива с d индексами) небольшим числом параметров может дать возможность работать с данными размерности d, равной 10, 100 или даже 1000 (такие проблемы возникают в квантовой молекулярной динамике, финансовом моделировании, при решении стохастических уравнений в частных производных).

Предлагается альтернативный метод сжатия изображений на основе тензорной аппроксимации. В условиях экспоненциального роста объемов передачи, хранения и обработки визуальной информации [1] применение данной подхода является оправданным.

Применённые методы тензорной аппроксимации показали хорошие результаты при работе с изображениями в оттенках серого, кроме того существует большой потенциал для их дальнейшего усовершенствования.

II. Тензорная аппроксимация

Идея аппроксимации тензоров заключается в нахождении закономерностей среди элементов тензора и приближении исходного тензора декомпозицией тензоров меньшей размерности.

Описание методов аппроксимации и современных алгоритмов их реализации дано в [2].

Известными методами аппроксимации являются:

1) Каноническая аппроксимация (наиболее удачный алгоритм получения такой декомпозиции это CANDECOMP/PARAFAC [3]).

2) Декомпозиция Такера (реализации N-mode PCA и N-mode SVD [4]).

3) Аппроксимация цепочкой тензоров (Tensor-Train Decomposition [5], TT).

Каноническая аппроксимация наиболее эффективна и позволяет представить тензор $\mathbf{T}$ любой размерности в виде набора двумерных тензоров $(\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_d)$, с помощью которых можно вычислить элементы $\mathbf{T}$ как сумму:

$$\mathbf{T}(i_1, i_2, \ldots, i_d) = \sum_{\alpha=1}^{r} \mathbf{U}_1(i_1, \alpha) \cdot \mathbf{U}_2(i_2, \alpha) \cdot \ldots \cdot \mathbf{U}_d(i_d, \alpha),$$

где $\mathbf{T}(i_1, i_2, \ldots, i_d)$ – исходный тензор размера $n_1 \times n_2 \times \ldots \times n_d$, $\mathbf{U}_1$ – двумерный тензор размера $n_k \times r$, r – канонический ранг.

Таким образом, возможна аппроксимация тензора $\mathbf{T}$ (при $n_1 \times n_2 \times \ldots \times n_d = n$) числом элементов, оцениваемым как $O(d \cdot n \cdot r)$. Преимуществом метода является то, что если ранг r невелик, то тензор можно представить очень компактно.

Однако алгоритмы получения канонической декомпозиции не являются стабильными, и даже если известно, что существует декомпозиция малого ранга r_{min}, то нет гарантии, что алгоритму удастся получить аппроксимацию с таким рангом.

Этого недостатка лишен метод аппроксимации цепочкой тензоров TT (Tensor-Train Decomposition). По числу элементов в аппроксимации метод приближается к канонической аппроксимации, при этом существует стабильный алгоритм для ее получения.

Идея метода заключается в представлении тензора большой размерности цепочкой тензоров размерности 3:

$$\mathbf{T}(i_1, i_2, \ldots, i_d) = \sum_{\alpha_0 \ldots \alpha_d} \mathbf{G}_1(\alpha_0, i_1, \alpha_1) \cdot \ldots \cdot \mathbf{G}_d(\alpha_{d-1}, i_d, \alpha_d), \quad (1)$$

где $\mathbf{G}_1$ – тензор размера $r_{k-1} \times n_k \times r_k$, $\alpha_0 = \alpha_1 = 1$.

При этом аппроксимация выполняется с точностью ε:

$$\left\| \mathbf{T} - \mathbf{T}^* \right\|_F \le \varepsilon \left\| \mathbf{T} \right\|_F,$$

где $\mathbf{T}$ – исходный тензор, $\mathbf{T}^*$ – тензор, которым был аппроксимирован $\mathbf{T}$.

Тензоры $\mathbf{G}_k$ вычисляются с помощью алгоритма SVD, что гарантирует получение декомпозиции для любых данных.

Сумму (1) можно представить матричным произведением, т.к. каждому значению i_k в трехмерном массиве $\mathbf{G}_k$ соответствуют матрицы $\mathbf{H}_k$:

$$\mathbf{T}(i_1, i_2, ..., i_d) = \mathbf{H}_1(i_1) \cdot \mathbf{H}_2(i_2) \cdot ... \cdot \mathbf{H}_d(i_d), \qquad (2)$$

где $\mathbf{H}_k$ – матрица размером $r_{k-1} \times r_k$. При этом необходимо, чтобы $r_0 \times r_d = 1$(результатом матричного произведения должен быть единственный элемент).

Т.к. ранги r_k обычно невелики, то по числу элементов аппроксимация приближается к канонической, и ее размер оценивается как $O((d-2) \cdot n \cdot r^2 + 2 \cdot n \cdot r)$. Алгоритм метода и численные результаты изложены в [5].

III. Применение вейвлетной цепочки фильтров (Wavelet Tensor-Train)

Сжатие с помощью аппроксимации осуществляется за счет выигрыша по числу элементов. Но снижение числа элементов не всегда приводит к сжатию, т.к. можно получить меньше элементов, но они при этом будут иметь более сложную структуру, чем исходные, вследствие чего будут закодированы менее эффективно.

Такая ситуация может произойти в методе TT: можно получить меньшее число элементов в аппроксимации, но они будут более сложными (дробные числа со знаком, кодируемые 8-ю байтами), чем исходные (целые без знака, кодируются 1 байтом). Поэтому возможно, что в результате сжатие будет невелико.

Применение вейвлетной цепочки фильтров (Wavelet Tensor-Train, WTT [6]) является модификацией метода TT, позволяющей представить исходный сигнал в более разреженном виде. Идея заключается в использовании тензоров $\mathbf{H}_k$ в качестве фильтров для исходного сигнала (т.е. $\mathbf{H}_k$ используется как матрица без преобразования в 3-х мерный тензор).

Фильтры являются базисом для данного сигнала, и при проецировании сигнала на этот базис можно получить массив с большим количеством нулей (разреженное представление сигнала, [7]). Такой массив коэффициентов имеет малую энтропию и хорошо сжимается. В результате фильтры являются сигнально-зависимыми, поэтому необходимо хранить их вместе со сжатым изображением.

Чтобы фильтры не были очень большими, необходимо ограничить их ранг, иначе они будут в точности представлять сигнал и их будет неудобно хранить. Введем параметр r_{max}, задающий максимальный ранг фильтров. Алгоритм получения фильтров приведен в [6].

Применение фильтров к изображению сводится к последовательному перемножению фильтра и матрицы изображения. Сигнал можно восстановить, т.к. фильтры являются ортогональными (свойство SVD алгоритма), т.е. выполняется соотношение

$$\mathbf{H}_k \cdot \mathbf{H}_k^{\mathrm{T}} = \mathbf{H}_k^{\mathrm{T}} \cdot \mathbf{H}_k = \mathbf{E}_k,$$

где $\mathbf{H}_k$ – ортогональная матрица фильтра (размера $r_k \times r_k$), $\mathbf{H}_k^{\mathrm{T}}$ – транспонированная матрица $\mathbf{H}_k$, $\mathbf{E}_k$ – единичная матрица размера $r_k \times r_k$.

A. Сравнение эффективности TT и WTT

Для сравнения эффективности методов WTT и TT оценим число бит на пиксель (*bit per pixel, bpp*), выраженное через энтропию результата:

$$bpp = \frac{entropy(result) \cdot size\_of(result)}{n \cdot m},$$

где *result* – массив, содержащий выходной результат работы TT/WTT, *entropy* – функция, вычисляющая энтропию элементов массива, *size_of* – определяет количество элементов в массиве, *n, m* – размеры матрицы изображения. Результатом работы WTT является разреженный массив коэффициентов и фильтры.

Максимальный ранг фильтров (r_{max}) в тесте равен 3. Число бит на пиксель будем сравнивать на соответствующих значениях качества восстановленных изображений, полученных с помощью показателя *PSNR.*

Результат сравнения представлен на рис. 1. В качестве входных данных использовалось стандартное тестовое изображение «Lena» в оттенках серого. Для оценки энтропии результирующие значения были приведены к 16-ти битным целым.

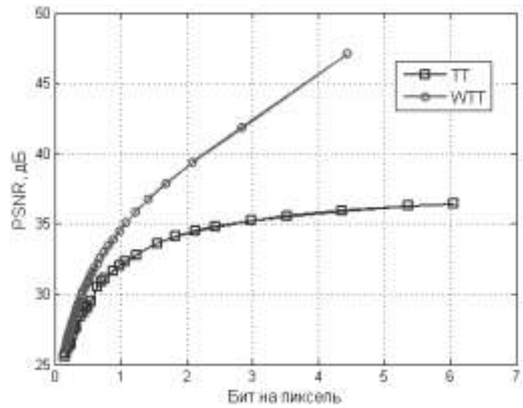

Рис. 1. Сравнение числа бит на пиксель, оцененного с помощью энтропии, для методов WTT и TT

Из графика видно, что WTT позволит эффективнее сжать изображение, чем TT.

B. Выбор ранга фильтров

В методе WTT можно ограничивать размер фильтров максимальным рангом r_{max}. Чем меньше ранг, тем меньше по размеру получаем фильтры, но при этом фильтры получаются менее эффективными для данного сигнала.

На рис. 2 приведена зависимость энтропии (bpp на основе энтропии) и качества восстановленного изображения (PSNR) от максимального ранга фильтров

(r_{max}). Энтропия вычислена для разреженного сигнала и фильтров.

Характерно, что до максимального ранга, равного 2, качество восстановленного изображения возрастает до *34 дБ* и при последующем увеличении максимального ранга стабилизируется и меняется незначительно.

Также видно, что минимум количества бит на пиксел достигается при максимальном ранге, равном *3* (для разных изображений). Поэтому наиболее подходящим будет выбор максимального ранга близким к *3*.

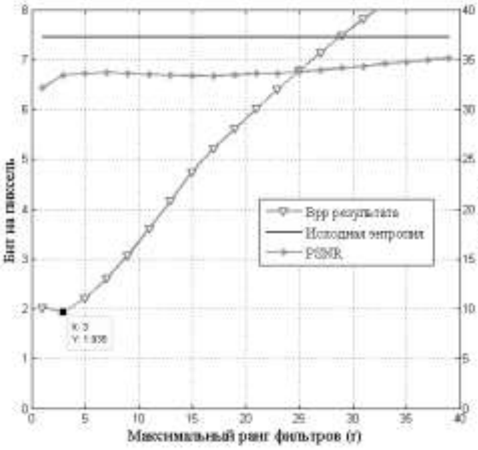

Рис. 2. Зависимость количества бит на пиксель и PSNR от r_{max} для изображения «Lena»

C. Использования предвычисленных фильтров

Проведен эксперимент по применению заранее вычисленных фильтров. Он заключался в следующем: было выбрано несколько изображений, и на их основе получены наборы фильтров, затем эти фильтры были применены ко всем исходным изображениям.

Для экспериментов были выбраны 5 стандартных тестовых изображений, включая «Lena», «pirate», «livingroom», «cameraman» и «бабуин».

Первый эксперимент проводился с фильтрами малого ранга (r_{max}). Результат опыта отражен в таблице 1 и таблице 2. Второй эксперимент – с фильтрами большого ранга (таблицы 3-4).

Значение bpp было вычислено на основе энтропии разреженных данных изображений (без фильтров). На диагоналях в таблицах приведен результат, соответствующий сжатию изображений «собственными» фильтрами.

Из этих данных видно, что при «малом» максимальном ранге фильтров качество изображений и степень сжатия практически не зависят от того, какими фильтрами сжато изображение.

Из этого следует вывод, что фильтры малого ранга можно вычислить заранее и не сохранять с изображением. Таким образом, исключив фильтры из массива сжимаемых данных, можно повысить степень сжатия и скорость работы алгоритма.

Таблица 1

Бит на пиксель (r_{max} = 3)

Изображение / Фильтр №	1	2	3	4	5
1	0,809	1,215	1,342	0,791	2,645
2	0,821	1,193	1,297	0,795	2,647
3	0,845	1,227	1,265	0,813	2,615
4	0,845	1,225	1,320	0,788	2,673
5	0,877	1,247	1,345	0,836	2,611

Таблица 2

PSNR (r_{max} = 3)

Изображение / Фильтр №	1	2	3	4	5
1	33,51	32,47	32,25	34,63	32,52
2	33,45	32,47	32,34	34,8	32,56
3	33,49	32,59	32,62	34,91	32,61
4	33,34	32,46	32,31	34,52	32,61
5	33,42	32,37	32,33	34,88	32,59

Таблица 3

Бит на пиксель (r_{max} = 50)

Изображение / Фильтр №	1	2	3	4	5
1	0,556	2,708	2,869	2,264	3,961
2	2,272	1,140	2,811	2,270	3,931
3	2,458	2,823	1,214	2,364	3,948
4	2,290	2,748	2,829	0,528	3,939
5	2,613	3,006	3,073	2,603	2,882

Таблица 4

PSNR (r_{max} = 50)

Изображение / Фильтр №	1	2	3	4	5
1	37,72	32,42	31,05	33,64	33,26
2	33,21	34,15	32,19	33,55	33,28
3	32,02	32,36	34,26	33,68	33,31
4	33,51	32,59	32,27	38,89	33,27
5	32,3	31,93	31,73	33,67	35,45

IV. КВАНТОВАНИЕ И ЭНТРОПИЙНОЕ КОДИРОВАНИЕ

На рис. 3 приведено типичное распределение коэффициентов преобразованного с помощью WTT изображения. Введем усредненный показатель качества на один коэффициент:

$$\Delta Q = \frac{\Delta PSNR}{\Delta count},$$

где $\Delta count$ – количество коэффициентов, исключенных из массива (коэффициенты заменены на 0), $\Delta PSNR$ – насколько ухудшилось качество из-за исключения коэффициентов.

Из рис. 3 видно, что влияние коэффициента на качество восстановленного изображения с ростом величины коэффициента (по модулю) возрастает.

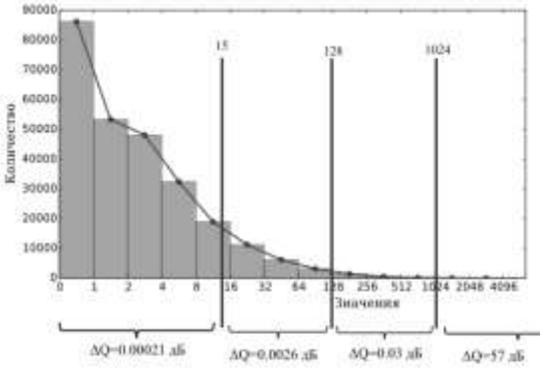

Рис. 3. Разбиение области значений на интервалы в зависимости от показателя ΔQ

Весь диапазон значений был разделен на участки, которые квантовались в зависимости от их влияния на качество (рис. 3): (0; 15] – квантование двумя битами, (15; 128] – квантование пятью битами, (128; 1024] – квантование девятью битами, (1024; Max] – квантование пятнадцатью битами.

Т.к. последний интервал (1024, Max] является наиболее значимым, то его надо квантовать наиболее точно. Коэффициенты из первого интервала, напротив, можно квантовать с большой погрешностью, значительно повысив тем самым степень сжатия.

Разбиение было произведено лишь на 4 интервала, т.к. большое количество интервалов уменьшит эффективность энтропийного кодирования.

Каждый интервал кодируется отдельно в пределах собственного контекста, образуя поток. Но при разделении на потоки теряется информация о расположении коэффициента в исходном массиве, поэтому требуется сохранить «карту» расположения коэффициентов.

V. Сравнение с современными алгоритмами сжатия изображений

Для оценки эффективности алгоритма сжатия изображений WTT было проведено сравнение с современными алгоритмами сжатия (JPEG2000 и JPEG).

На рис. 4 и 5 приведены графики зависимости показателя PSNR от количества бит на пиксель для WTT, JPEG и JPEG2000.

VI. Заключение

Было установлено, что по показателю PSNR, алгоритм сжатия изображений на основе WTT сравним с JPEG, но уступает JPEG2000.

Тензорные аппроксимации ориентированы на сжатие многомерной информации большого объема и, возможно, в дальнейшем их применение позволит решить проблему эффективного представления и сжатия новых типов контента: трехмерного телевидения, мультивидового видео и т.д. [1].

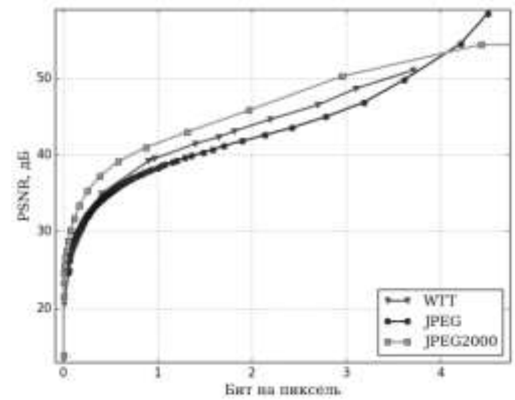

Рис. 4. Сравнение WTT с JPEG2000 и JPEG для изображения «Lena»

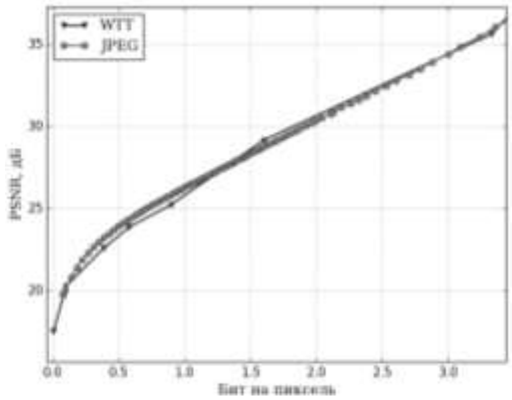

Рис. 5. Сравнение WTT с JPEG для изображения «Бабуин»

Поддержка

Работа выполнена при поддержке гранта РФФИ № 12-07-00388-а.

Литература

[1] Дворкович В., Чобану М. Проблемы и перспективы развития систем кодирования динамических изображений // MediaVision. 2011. № 2. С. 55-64.

[2] Kolda T.G., Bader B.W. Tensor decompositions and applications // SIAMRev. 2009. № 51. P. 455–500.

[3] Kiers H.A.L. A three-step algorithm for CANDECOMP/PARAFAC analysis of large data sets with multicollinearity // Journal of Chemometrics. 1998. № 12. P. 171-255.

[4] Vasilescu M.A.O., Terzopoulos D., Multilinear analysis of image ensembles: Tensor-Faces, in ECCV 2002 // Proceedings of the 7th European Conference on Computer Vision. 2002. V. 2350 of Lecture Notes in Computer Science. P. 447-460.

[5] Oseledets I.V. Tensor-train decomposition // SIAM J. Sci. Comput. 2011. V. 33. № 5. P. 2295–2317.

[6] Oseledets I.V. Algebraic wavelet transform via quantics tensor train decomposition // INM RAS. 2010. Preprint 2010-03.

[7] Чобану М.К. Многомерные многоскоростные системы обработки сигналов. М.: Техносфера, 2009. 480 с.