

Автокодировщик цветных изображений на основе кватернионной полносвязной нейронной сети

Н.А. Петровский М.И. Вашкевич
{nick.petrovsky, vashkevich}@bsuir.by

Белорусский государственный университет
информатики и радиоэлектроники
Кафедра электронных вычислительных средств
Минск, Беларусь

26-я конференция DSPA'2024
Цифровая обработка сигналов и её применение
Москва, Россия

28 марта, 2024



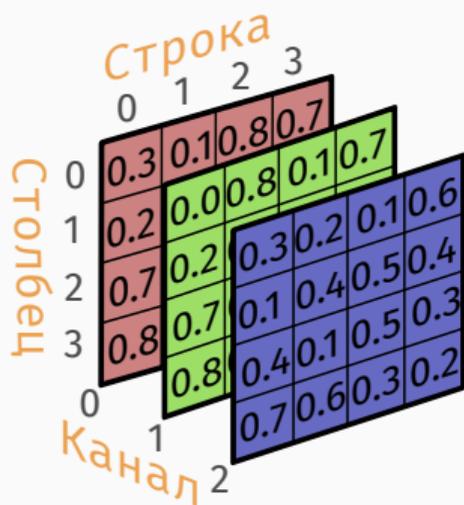
Содержание

1. Введение
2. Использование автокодировщиков для сжатия изображений
3. Алгебра кватернионов
4. Полносвязный слой кватернионной нейронной сети
5. Автокодировщик на основе вещественнозначной нейронной сети (RAE – *real autoencoder*)
6. Автокодировщик на основе кватернионной нейронной сети (QAE – *quaternion autoencoder*)
7. Описание эксперимента и результаты

Введение

Введение

- В настоящее время наблюдается интерес к использованию **кватернионов** при построении **нейронных сетей** для обработки **многомерных данных**^a
- Цветные изображения являются важным примером многомерных данных
- Обычно RGB-изображения обрабатываются при помощи сверточных нейронных сетей. Входное изображение интерпретируется, как 3-х мерный тензор



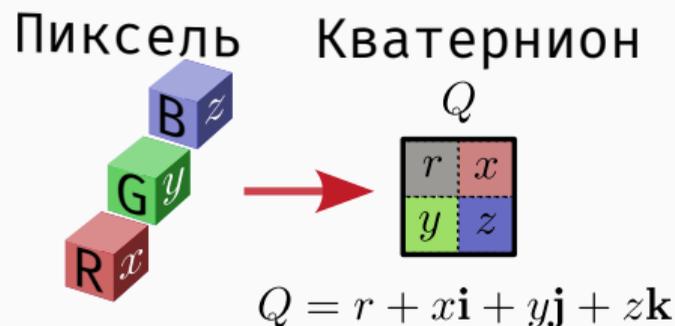
^aKusakabe, T., Kouda, N., Isokawa, T., Matsui, N. : A Study of Neural Network Based on Quaternion. Proceeding of SICE Annual Conference (2002) 776–779

Кватернион как контейнер для RGB-пикселя

Требования к модели представления изображения

- Модель должна учитывать локальные связи между каналами одного пикселя (R, G, B);
- Модель должна учитывать структурные признаки, описывающие межпиксельные связи.
- Предпочтительно обрабатывать RGB-изображение, не как три отдельных, независимых значения R, G и B, а как единым многомерный объект.

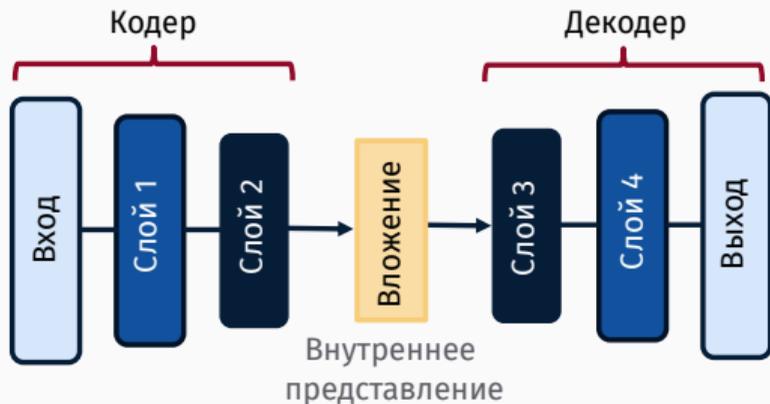
Алгебра кватернионов позволяют получить эффективный способ представления RGB-изображения для последующей обработки.



Сжатие цветных изображений с использованием автокодировщиков

Автокодировщик – нейронная сеть, состоящая из двух частей – кодера и декодера.

Архитектура автокодировщика



Цель исследования

- Получить базовую модель автокодировщика на основе кватернионов
- Выяснить дает ли преимущество использование кватернионов в задаче сжатия изображений при помощи автокодировщика
- Оценить преимущества кватернионной нейронной сети над вещественнозначной в задаче сжатия изображений (MSE, PSNR, SIMM)

Алгебра кватернионов

Алгебра кватернионов \mathbb{H}

Алгебра кватернионов \mathbb{H} — ассоциативная некоммутативная четырёхмерная алгебра:

$$\mathbb{H} = \{Q = q_1 + q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k} \mid q_1, q_2, q_3, q_4 \in \mathbb{R}\},$$

где соотношения ортогональных мнимых частей описывают операцию правила умножения

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1, \mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \mathbf{ki} = -\mathbf{ik} = \mathbf{j}.$$

К примеру, произведение кватернионов P и Q описывается как **Hamilton Product**¹

$$\begin{aligned} PQ = & p_1 q_1 - p_2 q_2 - p_3 q_3 - p_4 q_4 + \\ & (p_1 q_2 + p_2 q_1 + p_3 q_4 - p_4 q_3)\mathbf{i} + \\ & (p_1 q_3 + p_3 q_1 + p_4 q_2 - p_2 q_4)\mathbf{j} + \\ & (p_1 q_4 + p_4 q_1 + p_2 q_3 - p_3 q_2)\mathbf{k}, \end{aligned}$$

¹Parcollet T., Morchid M., Linarès G. Quaternion convolutional neural networks for heterogeneous image processing // *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. – 2019. – С. 8514-8518.

Алгебра кватернионов III: операция умножения

Однако, представив мнимые части i, j, k в виде блочно-диагональных матриц 4×4 , возможно описать произведение кватернионов в матричном виде:

$$PQ = \mathbf{M}^+(P) Q = \begin{bmatrix} p_1 & -p_2 & -p_3 & -p_4 \\ p_2 & p_1 & -p_4 & p_3 \\ p_3 & p_4 & p_1 & -p_2 \\ p_4 & p_3 & p_2 & p_1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

- Вычислительная сложность произведения кватернионов не зависит от типа умножения



Полносвязный слой кватернионной нейронной сети (НС)

Полносвязный слой НС

$$\mathbf{y} = h(\mathbf{W}\mathbf{x}),$$

где \mathbf{x} , \mathbf{y} – вектор входных и выходных значений, $h(\cdot)$ – функция активации, \mathbf{W} – матрица весов слоя НС:

$$\mathbf{W} = \begin{bmatrix} w_{00} & w_{01} & \dots & w_{0,N-1} \\ w_{10} & w_{11} & \dots & w_{1,N-1} \\ \vdots & \ddots & \dots & \vdots \\ w_{M-1,0} & w_{M-1,1} & \dots & w_{M-1,N-1} \end{bmatrix}.$$

Полносвязный слой кватернионной НС

$$Q_j = \sum_{i=0}^{N-1} \mathbf{M}^+(W_{ji}) X_i,$$

$$Y_j = g(Q_j)$$

где X_i – данные на i -м входе НС, представленные в виде кватернионов, W_{ji} – веса-кватернионы НС, $g(\cdot)$ – расщепленная функция активации.

Расщепленная функция активации

Общий вид расщепленной функции активации¹²

$$g(Q) = f(r) + f(x)\mathbf{i} + f(y)\mathbf{j} + f(z)\mathbf{k}$$

где $f(\cdot)$ – любая стандартная функция активации sigmoid, tanh, ReLU и другие.

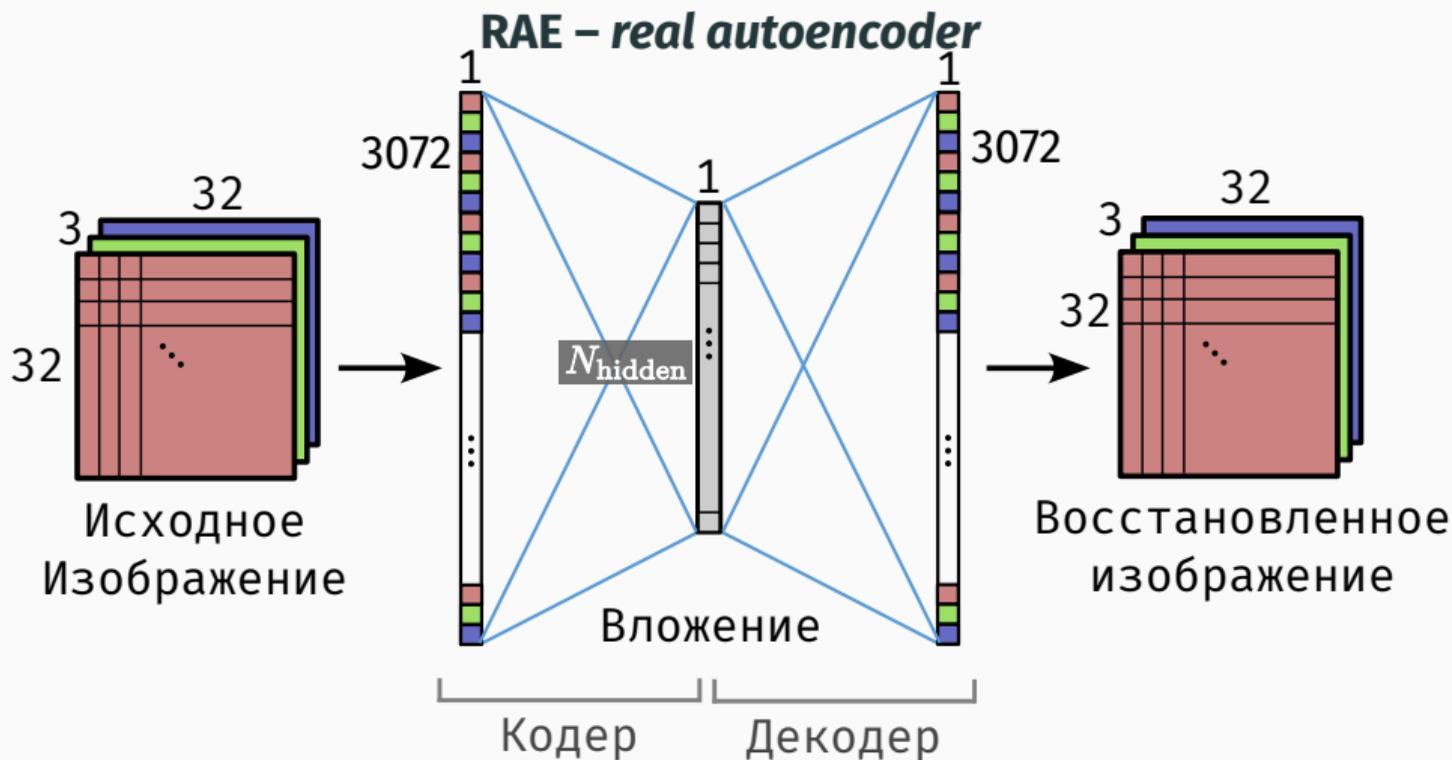
Расщеплённая функция активации позволяет использовать типовой подход обучения НС, без специфики гиперкомплексной алгебры.

¹Arena P, Fortuna L, Occhipinti L, Xibilia MG (1994) Neural networks for quaternion-valued function approximation. In: 1994 IEEE International symposium on circuits and systems, ISCAS'94, vol 6. IEEE, pp 307–310

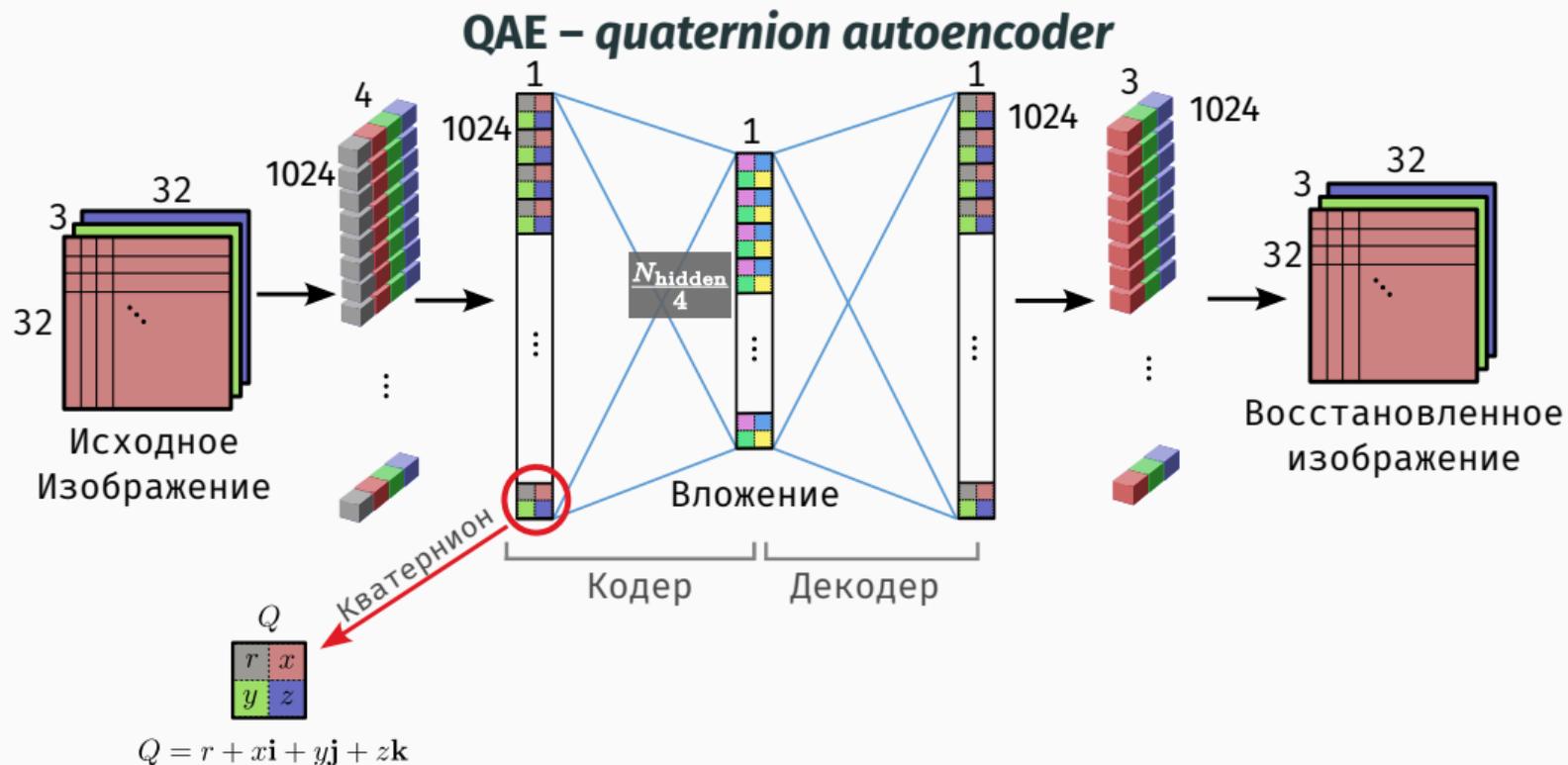
²Arena P, Fortuna L, Muscato G, Xibilia MG (1997) Multilayer perceptrons to approximate quaternion valued functions. Neural Netw 10(2):335–342

Автокодировщик для сжатия изображений

Автокодировщик на основе вещественнозначной НС



Автокодировщик на основе кватернионной НС



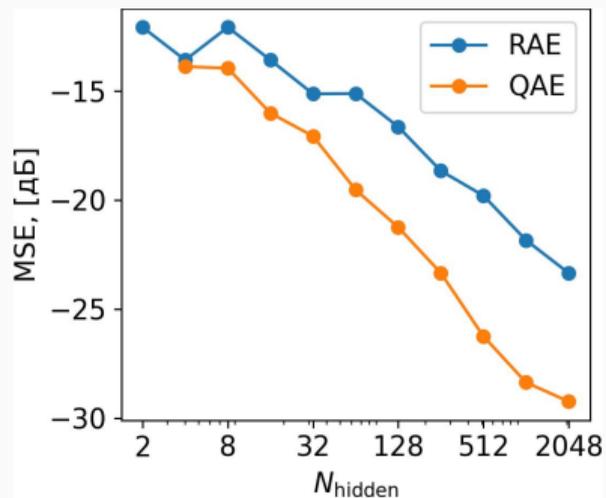
Эксперимент и результаты

Описание эксперимента

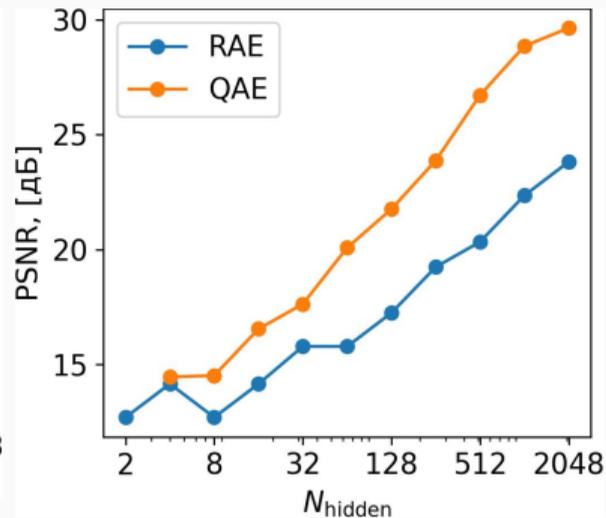
- Набор данных CIFAR-10 (60 тыс. RGB-изображения 32×32)
- В кодере использовалась функция активации ReLU, в декодере – логистический сигмоид
- Обучались автокодировщики с различным размером внутреннего слоя $N_{hidden} = \{2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$
- Инициализация весов выполнялась методом Хе
- Целевая функция – MSE
- Оптимизация производилась с использованием алгоритма Adam (скорость обучения $\eta = 10^{-3}$, число эпох – 50, размер батча – 256)
- Для оценки качества декодирования изображений использовались метрики MSE , $PSNR$ и $SIMM$

Результаты

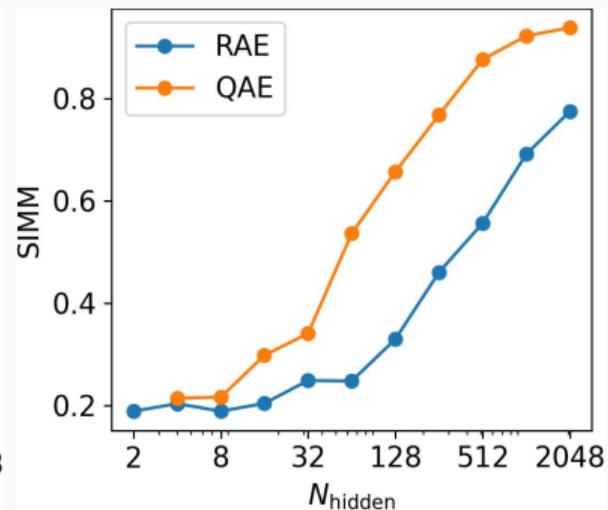
MSE



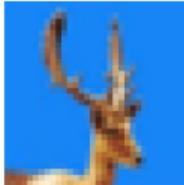
PSNR



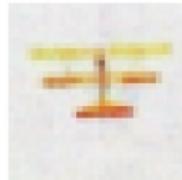
SIMM



Результат работы автокодировщика RAE-2048

#8100	#1270	#802	Original #4753	#3198	#7063	#9637
						
MSE = 0.0247	MSE = 0.0230	MSE = 0.0228	RAE-2048 MSE = 0.0205	MSE = 0.0202	MSE = 0.0190	MSE = 0.0188
						
MSE = 0.0019	MSE = 0.0040	MSE = 0.0035	QAE-2048 MSE = 0.0055	MSE = 0.0032	MSE = 0.0023	MSE = 0.0026
						

Результат работы автокодировщика QAE-2048

#3444	#8684	#1078	Original #7803	#4753	#9921	#8903
						
MSE = 0.0064	MSE = 0.0113	MSE = 0.0071	RAE-2048 MSE = 0.0064	MSE = 0.0205	MSE = 0.0080	MSE = 0.0081
						
MSE = 0.0066	MSE = 0.0062	MSE = 0.0057	QAE-2048 MSE = 0.0056	MSE = 0.0055	MSE = 0.0055	MSE = 0.0054
						

Сравнение полученных результатов

- Полученные автокодировщики на основе кватернионной НС сравнивались с другими вещественнозначным автокодировщиком, имеющим архитектуру «бутылочное горлышко»².
- Автокодировщик¹, использует НС с двумя слоями: первый сверточный, а второй полносвязный и обеспечивает степень сжатия 2:1, т.е. внутренний слой содержал в два раза меньше элементов, чем входной и выходной.
- У автокодировщика¹ индекс структурной схожести (SIMM) равен **0,905**.
- Полученный в данной работе автокодировщик QAE-1024 обеспечивает сжатие 3:1 и имеет SIMM равный **0,922**.

²Yijing Watkins et al. "Image compression: Sparse coding vs. bottleneck autoencoders". In: 2018 *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. IEEE. 2018, pp. 17–20.

Выводы

- Рассмотренный вычислительный эксперимент на основе полносвязного автокодировщика показывает, что представление скоррелированных данных, таких как цветные RGB-изображения, в алгебре кватернионов позволяет лучше учитывать характер исходных данных.
- Предложенные автокодировщики, имеющие различную степень сжатия, позволяют получить более высокие значения объективных характеристик декодирования цветных изображений по сравнению с аналогичными вещественнозначными автокодировщиками (PSNR в среднем выше на 3,85 дБ, SSIM на 0,18).