



ФПМИ

МАТЕМАТИКА БОЛЬШИХ ДАННЫХ

---

# Тензорные сети

---

*Автор:*  
Семененко Александр

1.12.2023

---

### 3 Численные эксперименты

**Изображения** В этом разделе мы приведем результаты работы Greedy-TN и SVD для задач сжатия изображений. В качестве примера рассмотрим три разных изображения 'cruise', 'pepper' и 'fields' в градациях серого (см Рис 1).



Рис. 1: Рассматриваемые изображения.

Для простоты мы берём обрезанные изображения размера  $256 \times 256$  (соответствующую матрицу будем обозначать за  $X$ ). Их можно получить посредством класса `TargetImage(file_name=img_name, height=256, width=256)`.

**Greedy-TN** Для запуска алгоритма Greedy-TN после создания экземпляра `target_image` необходимо инициализировать модель:

```
model = GreedyTensorNetwork(num_cores=8, core_base=4),
```

где `num_cores` обозначает количество  $n$  факторов  $\{G^{(i)}\}_{i=1}^n$  тензорного разложения (??) (суть число узлов в тензорной сети), элементы которых просэмплированы из стандартного нормального распределения. В свою очередь, `core_base` обозначает размерность  $d$  каждого измерения (моды) (оно же вес связи в тензорной сети), то есть для данного изображения мы полагаем  $d_1 = d_2 = \dots = d_n = d$ .

Затем создается экземпляр `trainer` класса `Trainer(max_steps=30, num_gpus=1)`. Здесь `max_steps` это число обновлений тензорной сети, то есть количество шагов в рамках каждого из которых производится поиск оптимальной связи с точки зрения наименьшего значения функции потерь. Параметр `num_gpus` задаёт число используемых GPU, если `num_gpus = 0`, то задействуется CPU.

Наконец, для обучения факторов тензорной сети, задаваемых моделью `model`, вызывается метод класса `Trainer`:

```
trainer.fit(model, target_image),
```

который по умолчанию использует попеременную оптимизацию (ALS).

**SVD** Не лишним будет предварительно проанализировать скорость убывания сингулярных чисел каждой из матриц, на основании чего можно сделать оценку

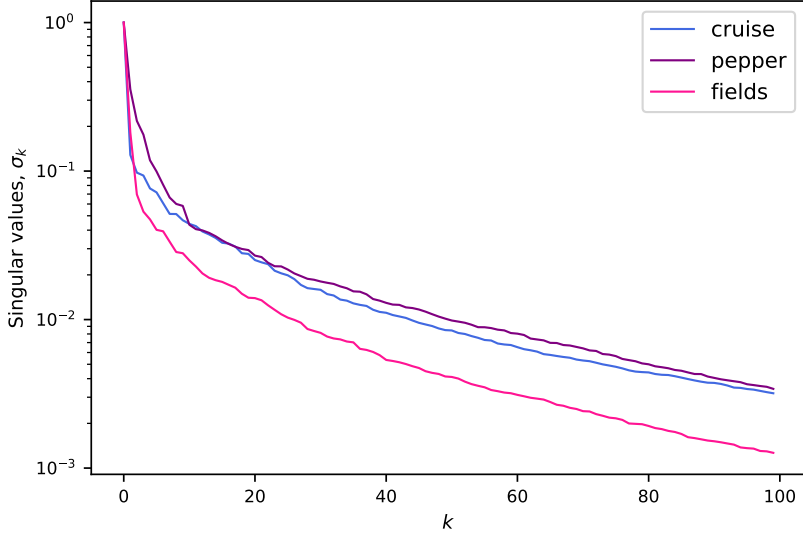


Рис. 2: Убывание значений сингулярных чисел для рассматриваемых изображений.

их рангов (Рис. 2). Отсюда естественным образом возникает идея аппроксимации исходной матрицы  $X$ . Для этого достаточно взять усеченное сингулярное разложение. Итак, взяв в качестве  $r_e$  оценку ранга  $X$ , можно рассмотреть следующую аппроксимацию, которая в то же время служит сжатием:

$$X = \sum_{\alpha=1}^r \sigma_{\alpha} u_{\alpha} v_{\alpha}^T \approx \sum_{\alpha=1}^{r_e} \sigma_{\alpha} u_{\alpha} v_{\alpha}^T,$$

где  $u_{\alpha}$  и  $v_{\alpha}$  образуют соответствующие ортонормированные системы векторов. Более того, можно показать, что такая аппроксимация является наилучшей относительно произвольной унитарно инвариантной нормы (Eckart и Young 1936, Mirsky 1960).

**Сжатие** Прежде чем привести результат работы упомянутых двух алгоритмов, напомним, что мы рассматриваем тензорную сеть

$$\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \dots, \mathcal{G}^{(8)}$$

с восьмью узлами, где  $\mathcal{G}^{(i)} \in \mathbb{R}^{R_{1,i} \times \dots \times R_{i-1,i} \times d_i \times R_{i,i+1} \times \dots \times R_{i,8}}$  для  $i \in \{1, \dots, 8\}$ , причём  $d_i \equiv d = 4$ . Веса связей при инициализации полагаются равными единице, то есть  $R_{i,j} = 1$ , что означает отсутствие связей. Мы считаем, что  $R_{i,j} = R_{j,i}$  для  $i > j$ .

В качестве ошибки мы берем

$$\text{RMSE} = \frac{\|\mathcal{X} - \mathcal{T}\|_F}{\|\mathcal{T}\|_F},$$

где за  $\mathcal{X} = \llbracket \mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \dots, \mathcal{G}^{(8)} \rrbracket$  обозначен тензор, образованный из факторов тензорной сети, а  $\mathcal{T}$  это целевой тензор, полученный из исходного изображения путём взятия операции **Reshape**.

Коэффициент сжатия мы полагаем равным

$$C = \frac{\sum_{i=1}^8 \#\{\mathcal{G}^{(i)}\}}{\#\{\mathcal{T}\}} \times 100\%,$$

где  $\#\{\cdot\}$  обозначает общее число элементов тензора.

Так, на Рис. 3 видно, что ошибка приближения падает по мере увеличения числа параметров тензорной сети, то есть роста коэффициента  $C$ . Последнее является результатом увеличения весов связей (рангов или размерностей)  $R_{i,j}$ .

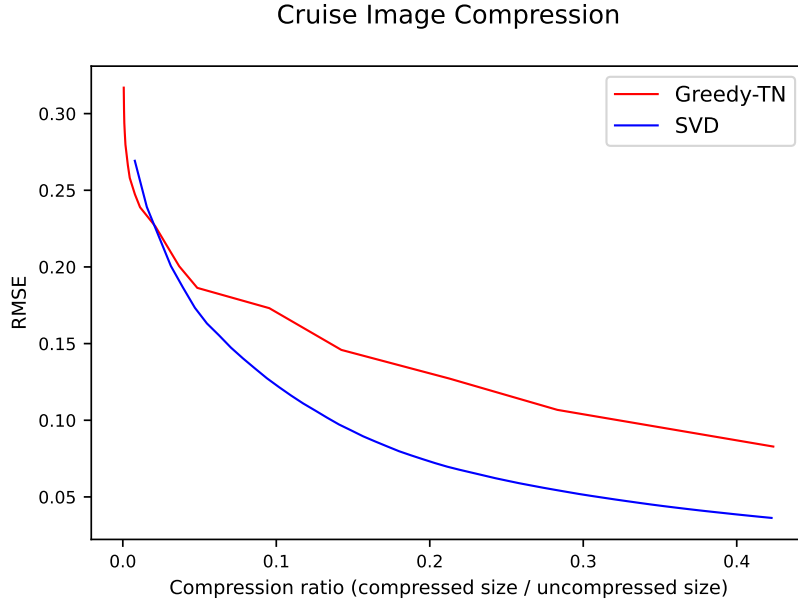


Рис. 3: Зависимость ошибки RMSE от коэффициента сжатия для изображения 'cruise'.

Визуальное сравнение двух подходов представлено на Рис. 4.

На этом примере видно, что сжатие с помощью SVD является более эффективным по сравнению с Greedy-TN, однако любопытным представляется рассмотреть область «экстремальных» сжатий, когда коэффициент  $C$  около нуля.

Перейдем к следующему изображению и проведем для него те же эксперименты, но уже в области малых коэффициентов сжатия (см. Рис. 5, 6). На этом примере уже заметно преимущество жадной стратегии построения тензорного разложения, нежели аппроксимации сингулярным разложением.

Третий пример показывает немного ярче, чем первый то, что метод Greedy-TN может быть более эффективен по сравнению с SVD при достаточно малом числе параметров (см. Рис. 7, 8).

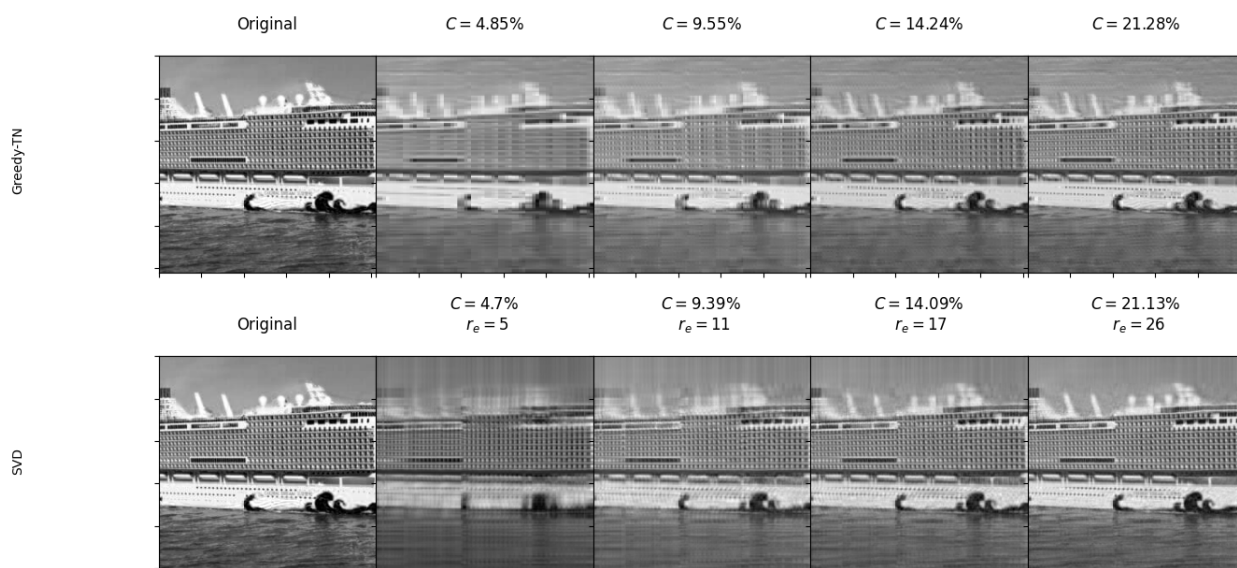


Рис. 4: Результат сжатия изображения 'cruise' с помощью Greedy-TN (первая строка) и SVD (вторая строка). Здесь  $C$  коэффициент сжатия, а  $r_e$  является оценкой ранга изображения в SVD.

## Список литературы

- Eckart, Carl и G. Marion Young (1936). «The approximation of one matrix by another of lower rank». В: *Psychometrika* 1, с. 211–218. URL: <https://api.semanticscholar.org/CorpusID:10163399>.
- Mirsky, L. (1960). «SYMMETRIC GAUGE FUNCTIONS AND UNITARILY INVARIANT NORMS». В: *Quarterly Journal of Mathematics* 11, с. 50–59. URL: <https://api.semanticscholar.org/CorpusID:120585992>.

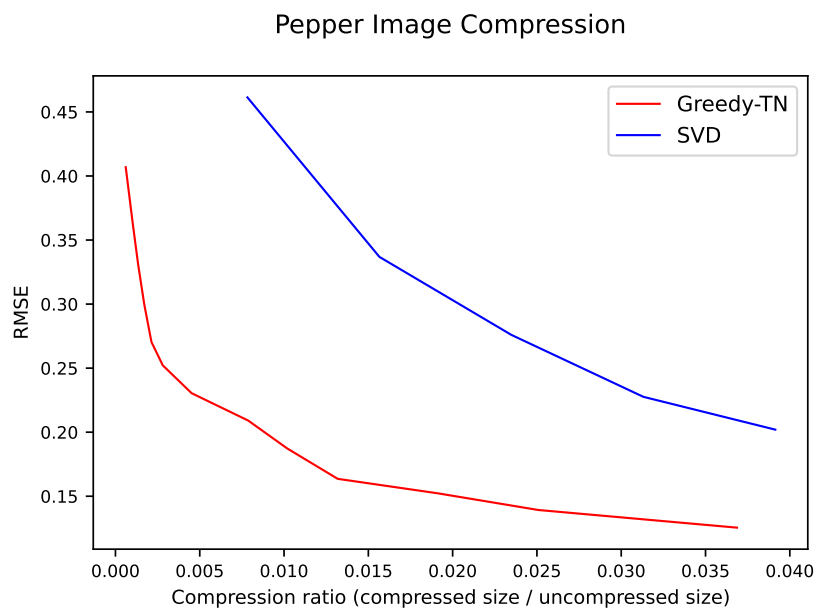


Рис. 5: Зависимость ошибки RMSE от коэффициента сжатия для изображения 'pepper'.

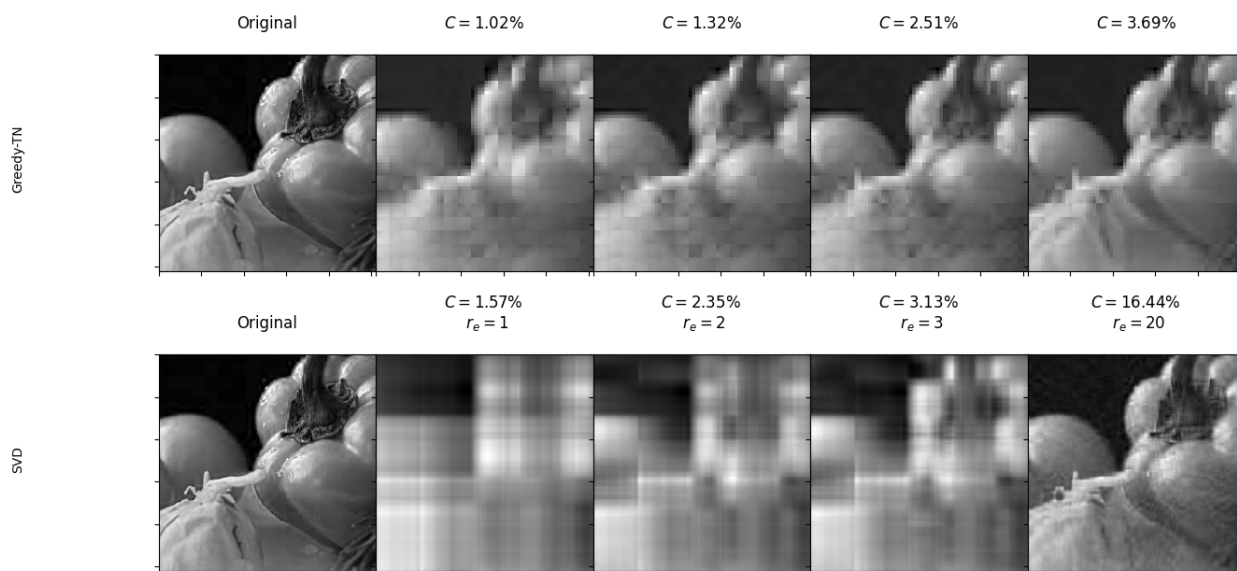


Рис. 6: Результат сжатия изображения 'pepper' с помощью Greedy-TN (первая строка) и SVD (вторая строка). Здесь  $C$  коэффициент сжатия, а  $r_e$  является оценкой ранга изображения в SVD.

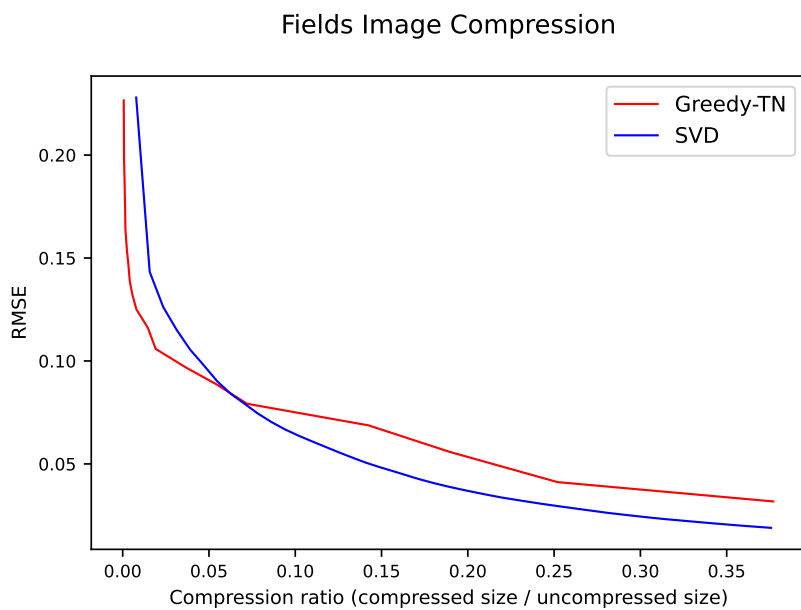


Рис. 7: Зависимость ошибки RMSE от коэффициента сжатия для изображения 'fields'.

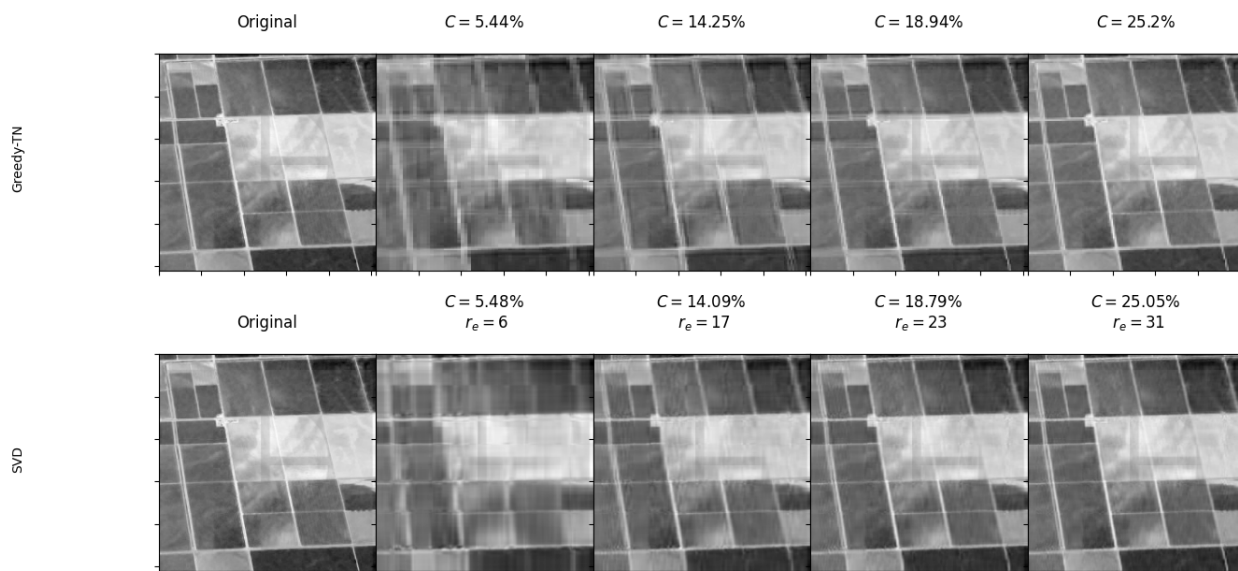


Рис. 8: Результат сжатия изображения 'fields' с помощью Greedy-TN (первая строка) и SVD (вторая строка). Здесь  $C$  коэффициент сжатия, а  $r_e$  является оценкой ранга изображения в SVD.