

# Лабораторная работа 207

Охотников Никита Владимирович

МФТИ

2024

# Введение

## Цели

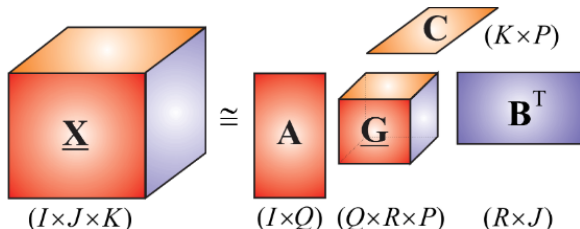
- ▶ Рассмотреть обобщение SVD на тензоры размерности 4 – алгоритм HOSVD
- ▶ Использовать truncated SVD для полученных разложений по аналогии с двумерным PCA для уменьшения размера хранимого представления
- ▶ Оценить качество восстановления 4-хмерных видео при снижении одной из размерностей в разложении до единицы

# Теоретическая часть

## Tucker decomposition

На примере трехмерного тензора, для понятной визуализации разложение Таккера выглядит следующим образом:

$$\underline{X} \in \mathbb{R}^{I \times J \times K}, \quad A \in \mathbb{R}^{I \times Q}, \quad B \in \mathbb{R}^{J \times R}, \quad C \in \mathbb{R}^{K \times P}$$



$$\underline{X} \cong \sum_{q=1}^Q \sum_{r=1}^R \sum_{p=1}^P g_{qrp} a_q \circ b_r \circ c_p = \underline{G} \times_1 A \times_2 B \times_3 C = [\underline{G}; A, B, C]$$

# Теоретическая часть

## Higher Order Singular Value Decomposition (HOSVD)

HOSVD – специальный случай разложения Таккера, в котором все фактор-матрицы ортогональны. Фактор матрицы находятся как матрицы из truncated-SVD для разверток  $X$  по соответствующей моде. Для трехмерного случая выше:

$$X_{(1)} = U_1 \Sigma_1 V_1^T \rightarrow A = U_1[1 : R_1]$$

$$X_{(2)} = U_2 \Sigma_2 V_2^T \rightarrow B = U_2[1 : R_2]$$

$$X_{(3)} = U_3 \Sigma_3 V_3^T \rightarrow C = U_3[1 : R_3]$$

Core-тензор после этого вычисляется как

$$\underline{G} = \underline{X} \times_1 A^T \times_2 B^T \times_3 C^T$$

# Практическая часть часть

## Данные

В качестве синтетического датасета рассмотрим собственноручно сгенерированный ColoredMovingMnist – пары семплов из MNIST <sup>1</sup> случайного цвета,двигающиеся на черном фоне в произвольных направлениях в течение 4 секунд.

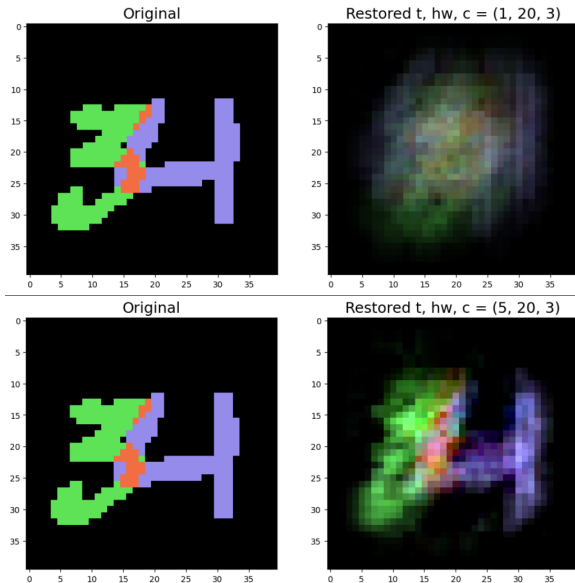
Исходный размер  $20 \times 40 \times 40 \times 3$ . Рассматривается сжатие по времени до  $[1, 5, 10]$ , по высоте и ширине до  $[5, 10, 20]$  и по каналам до  $[1, 2]$



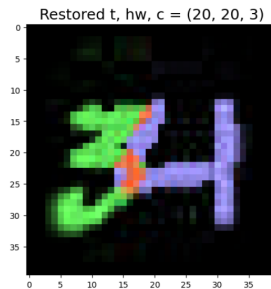
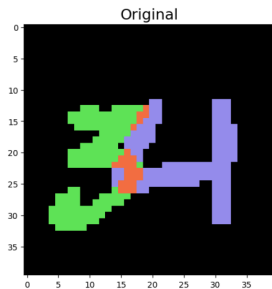
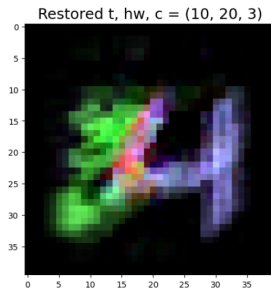
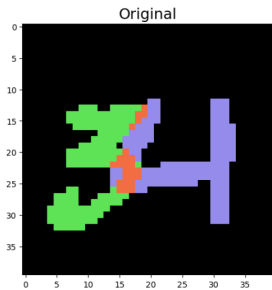
---

<sup>1</sup><http://yann.lecun.com/exdb/mnist/>

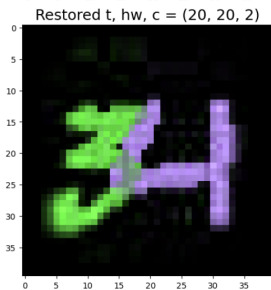
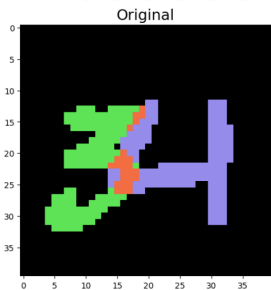
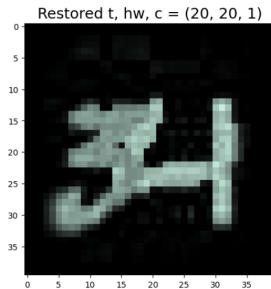
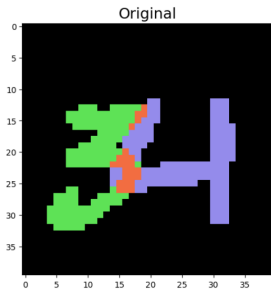
# Результаты эксперимента



# Результаты эксперимента



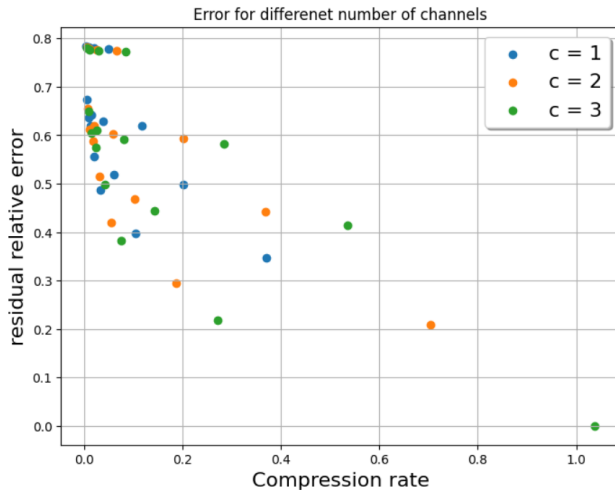
# Результаты эксперимента





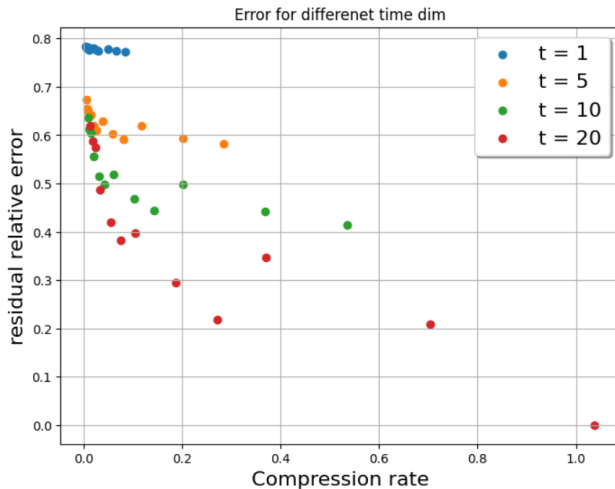
# Результаты эксперимента

## Зависимость ошибки от compression rate



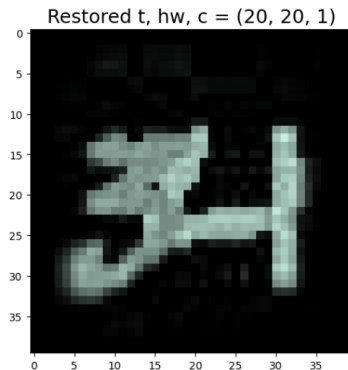
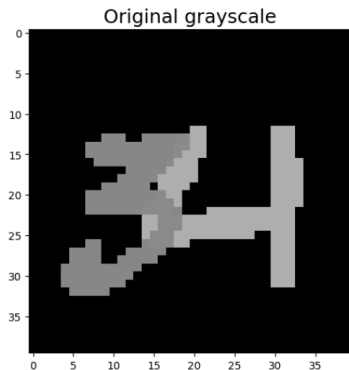
# Результаты эксперимента

## Зависимость ошибки от compression rate



# Результаты эксперимента

## Grayscale



# Заключение

## Выводы

- ▶ Метод работает
- ▶ Метод едва ли позволяет значительно сжать видео рассмотренного размера, поскольку вместо одного тензора приходится хранить четыре
- ▶ избавиться от целой размерности можно только по размерности каналов
- ▶ Сжатие по размерности каналов до одного сравнимо с ручным по яркости