# Тензорные разложения как инструмент сжатия нейронных сетей

Singular Matrixmen

Кафанов Степан
Дворянков Роман
Иудин Егор
Мусаева Асият

# Цель проекта

Сравнение работы CNN с использованием тензорного разложения и без него

- обучение классической resnet18
- модификация и обучение resnet18
- сравнение результатов (вычислительная сложность, память, accuracy)

# Введение про тензорное разложение

Для матриц известно множество разложений.

Самые популярные для приложений:

$$A = LU$$

$$A = QR$$

$$A = U\Sigma V^*$$

рассматривали в прошлом проекте

# Введение про тензорное разложение

Тензорные разложения представляют собой удобную модель структуры многомерных массивов, основанную на идее разделения переменных.

Тензоры являются обобщением матриц, структура которых также имеет повсеместное применение.

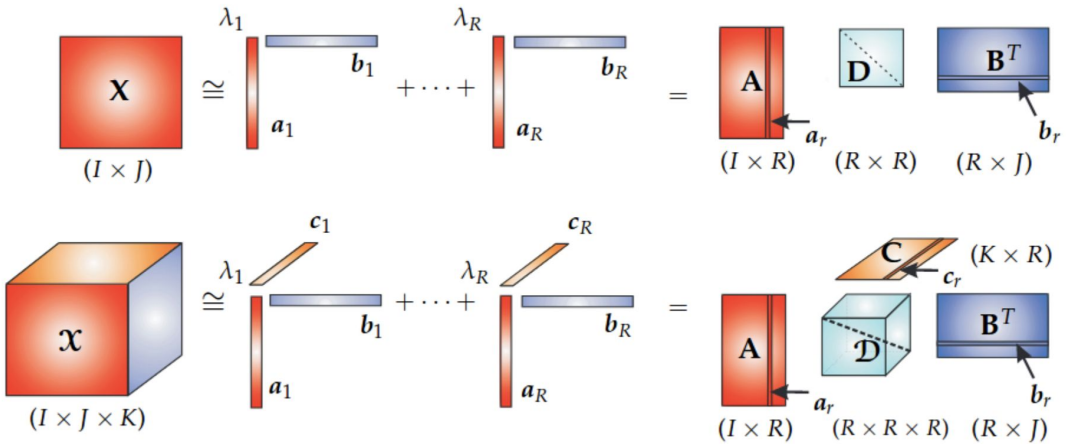
*Вопросы:* Можно ли также найти какие-то “хорошие” разложения, как и для матриц? Обобщается ли SVD, в частности?

*Спойлер:* единого обобщения нет

# CPD (Canonical Polyadic decomposition)

Простейшим обобщение SVD-разложения.

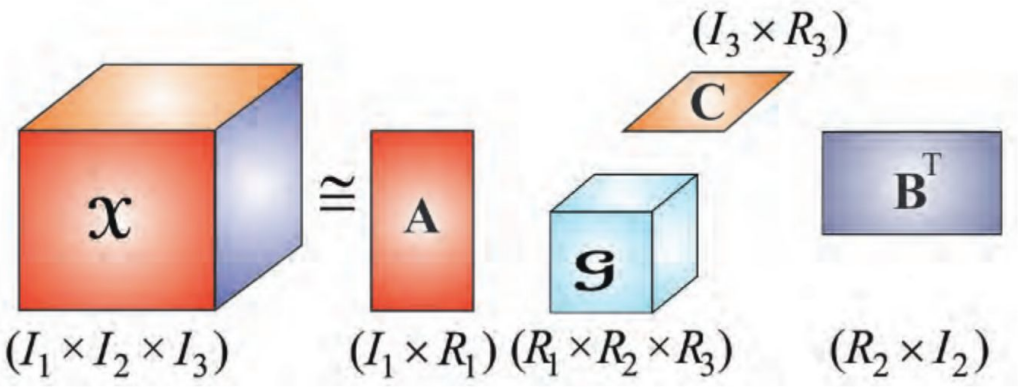
$$T(i_1, ..., i_d) = \sum_{r=1}^{R} U_1(i_1, r) U_2(i_2, r) ... U_d(i_d, r)$$



# TKD (Tucker decomposition)

Является более устойчивым вариантом разложения. Его можно получить путём SVD-разложения вдоль каждой размерности и дальнейшего объединения результатов.

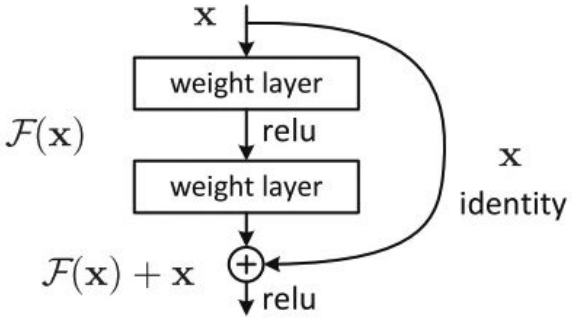
$$T(i_1, ..., i_d) = \sum_{r_1,...,r_d} G(r_1, ..., r_d) U_1(i_1, r_1) U_2(i_2, r_2) ... U_d(i_d, r_d)$$



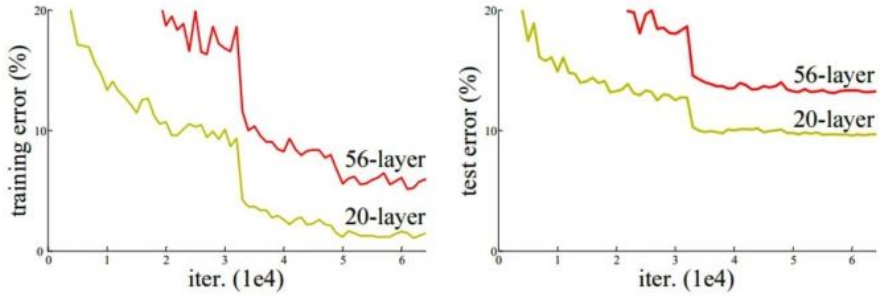
# ResNet18

Проблема: деградация качества при добавлении новых слоев по причине затухания градиента

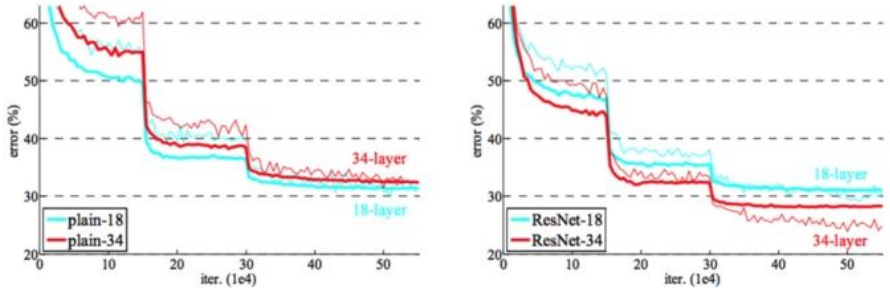
Решение: введение residual blocks, в котором есть skip connection




Graphic from the original paper

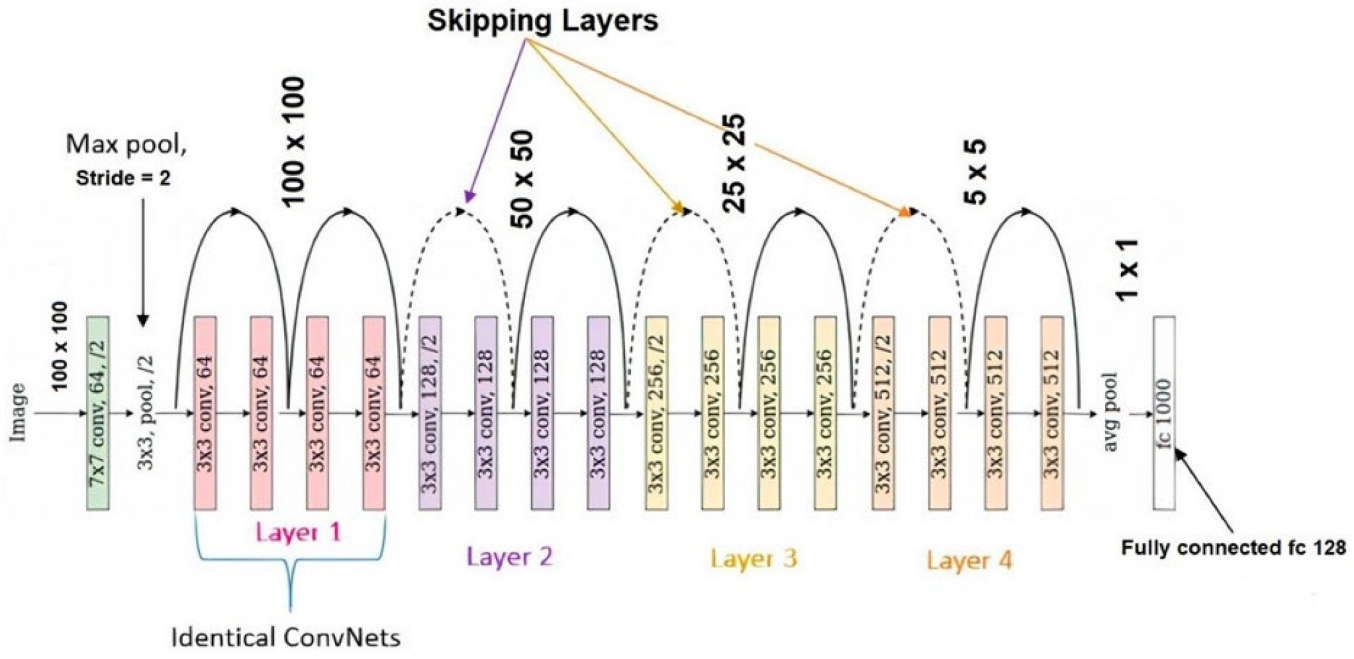



Classification error with Cifar-10 data. Graphic taken from the orignal paper.




Graphic from the original paper

# ResNet18




ResNet-18 Architecture

# Пример разложения сверточного слоя

* Before the decomposition:

Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))

* After the decomposition:

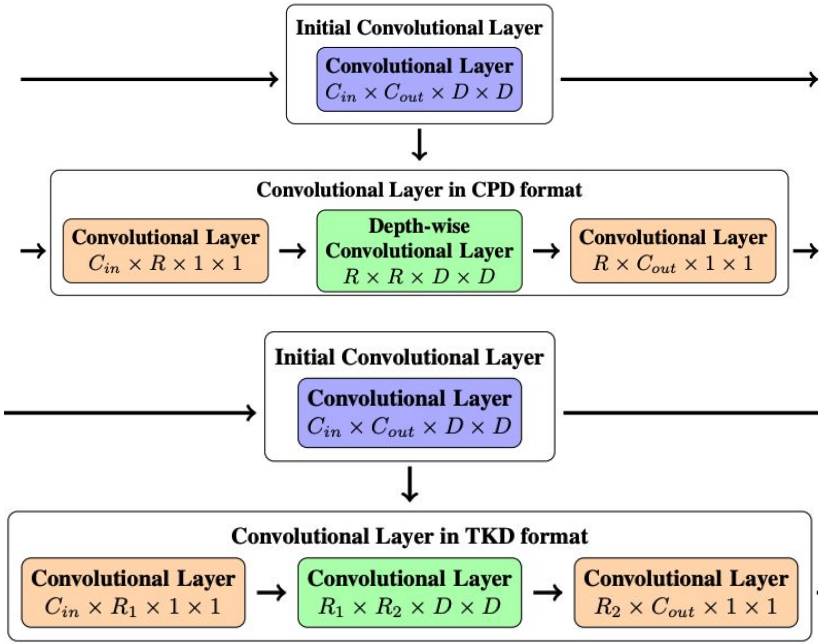
Sequential(

  (0): Conv2d(64, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)

  (1): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

  (2): Conv2d(3, 3, kernel_size=(1, 1), stride=(1, 1))

)



# Эксперименты

1. Resnet 18 -> Fine-tune (CFAR 10) -> результаты
2. Resnet 18 -> Fine-tune (CFAR 10) + CPD -> Fine-tune (CFAR 10) -> результаты
3. Resnet 18 -> Fine-tune (CFAR 10) + TKD -> Fine-tune (CFAR 10) -> результаты

# Результаты экспериментов (GPU)

CPD

- Вычислительная сложность: -45%
- Кол-во параметров: -50%

TKD

- Вычислительная сложность: -41%
- Кол-во параметров: -41%

| | Pure | CPD | TKD |
|---|---|---|---|
| Computational complexity | 37.25 MMac | 20.55 Mmac (-45%) | 21.81 Mmac (-41%) |
| Number of parameters | 11.18 M | 5.62 M (-50%) | 6.4 M (-42%) |
| Accuracy | 71.070% | 17.840% | 22.130% |
| Fine-tune | 84% (20 эпох) | 68.130% (5 эпох) | 65.790% (5 эпох) |

# Выводы

Тензорное разложение сверточных слоев позволяется почти **в два раза** сократить количество параметров модели, число операций и при этом сохранить приемлемое качество с возможностью его повышения, если увеличить число эпох дообучения.

# Планы

- На более мощном железе дообучить на большем количестве эпох, чтобы получить лучшее качество.
- Так как сжатие сети приводит к неустойчивости. В дальнейшем можно применить к факторизованным слоям нейронной сети алгоритм минимизации функции чувствительности тензорных разложений.

# Ссылка на гитхаб

https://github.com/Mr-Grag-Universe/NLA_project_2.git