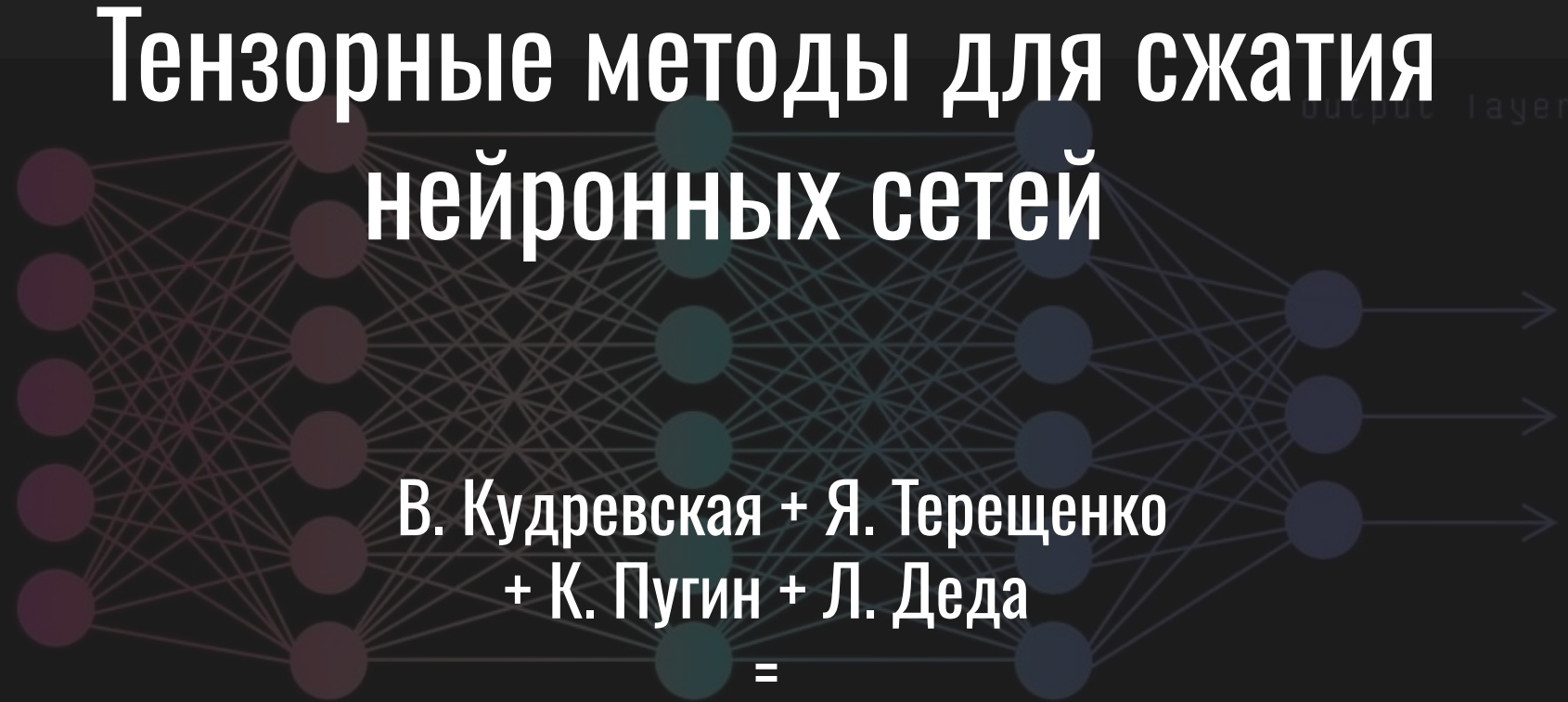


Тензорные методы для сжатия нейронных сетей

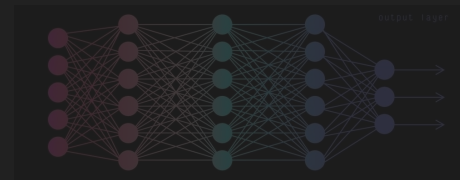


В. Кудревская + Я. Терещенко
+ К. Пугин + Л. Деда

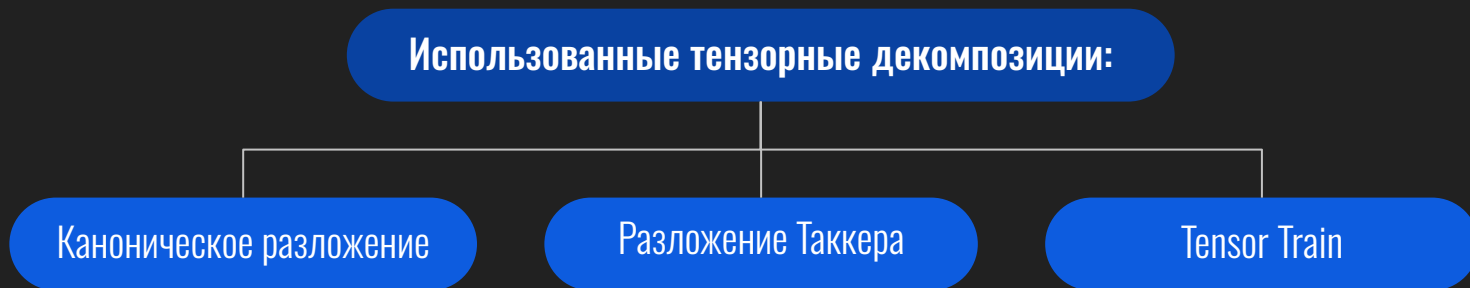
=

InterPoLazio

Что? Где? Когда?

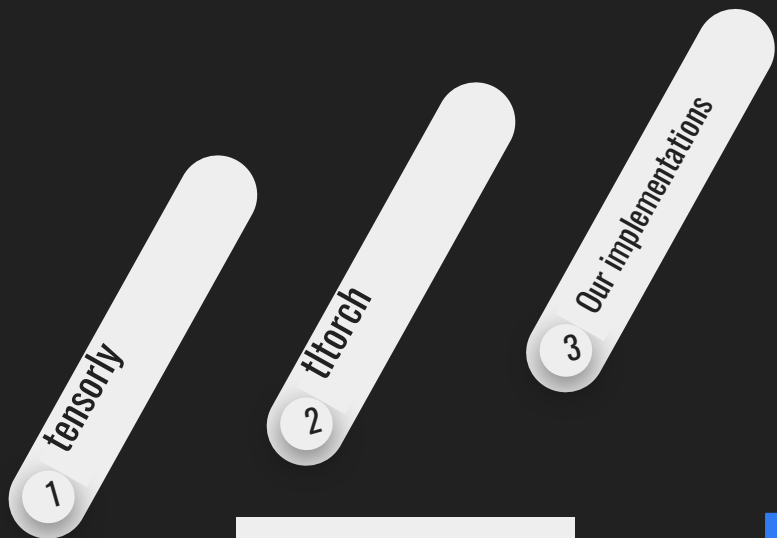
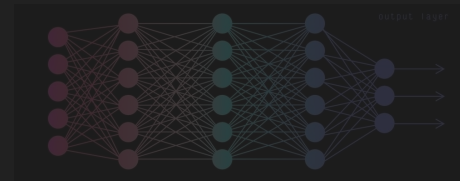


- Применение тензорных декомпозиций для сжатия нейронных сетей без потерь в качестве



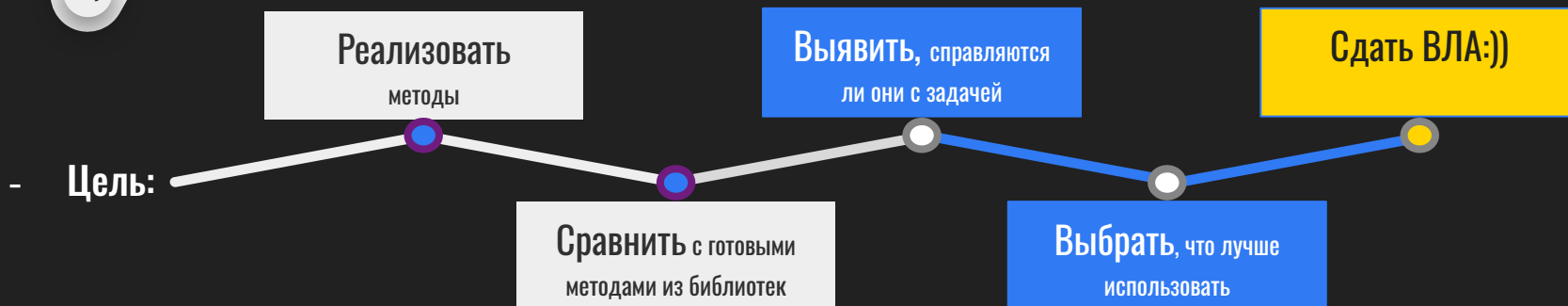
- Разложения применялись к полносвязным слоям и сверточным

Что? Где? Когда?

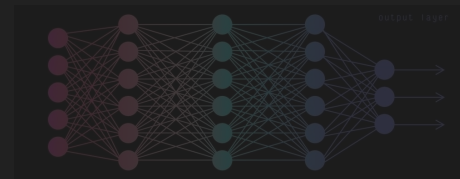


- **Датасет:** классификация медуз
(6 классов)

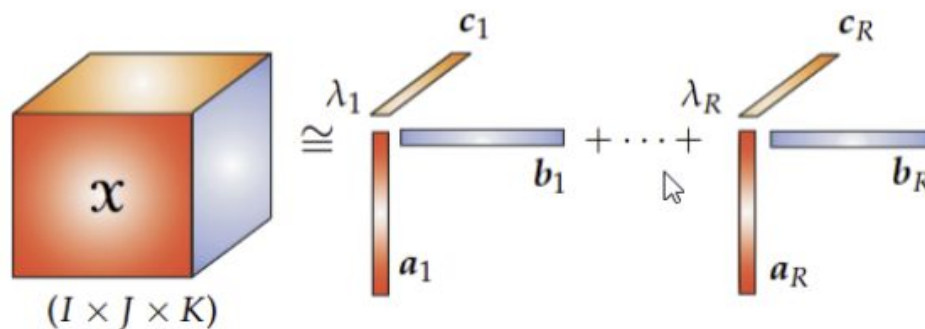
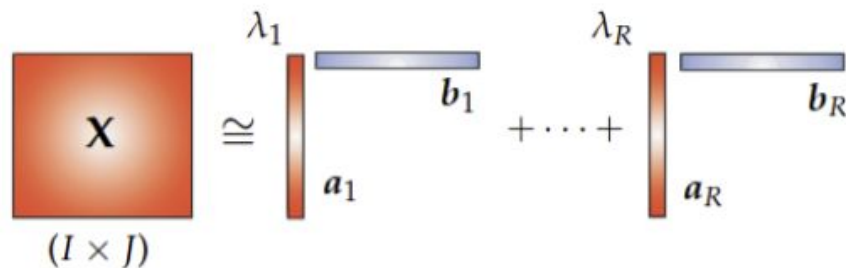
- **Метрика:** accuracy



Каноническое разложение



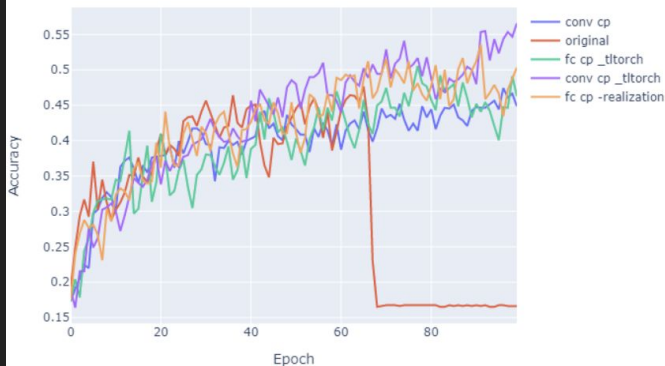
$$A(i_1, \dots, i_d) \approx \sum_{\alpha=1}^r U_1(i_1, \alpha) \dots U_d(i_d, \alpha)$$



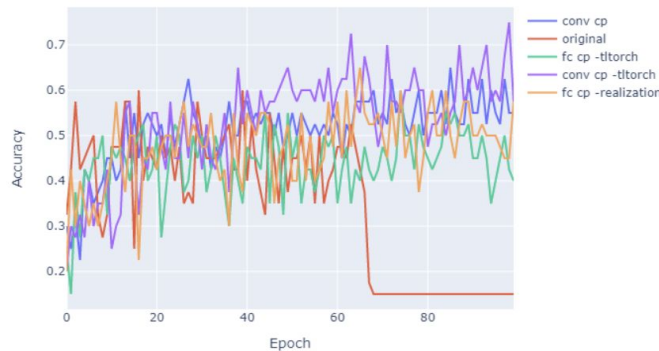
Метод 1: Каноническое разложение



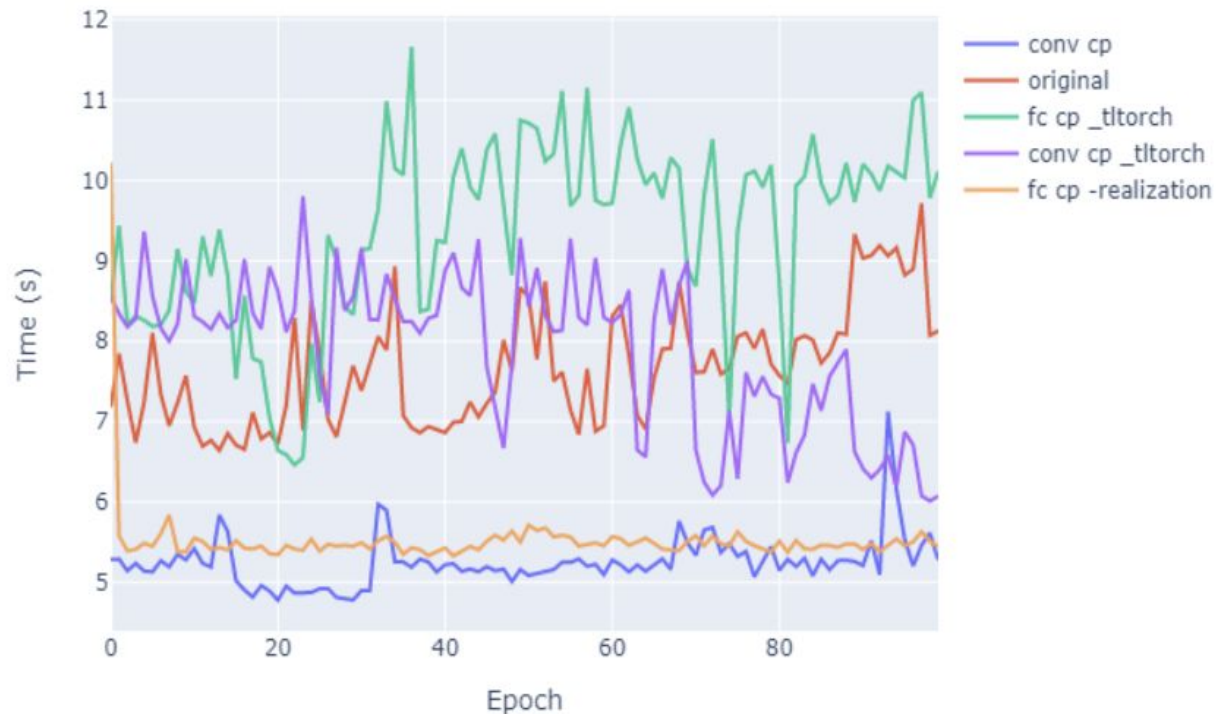
Training Accuracy



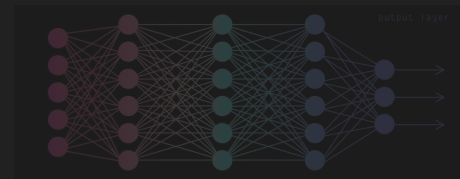
Testing Accuracy



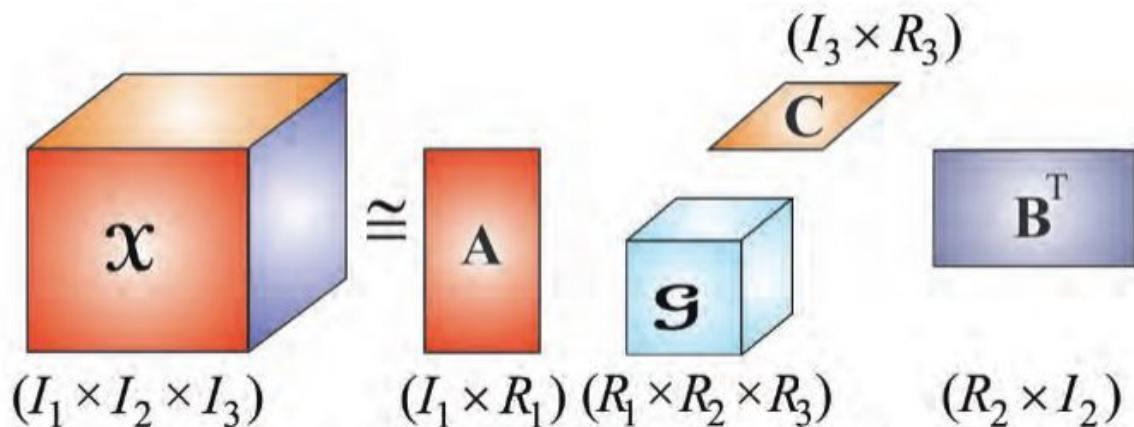
Time per Epoch



Разложение Таккера (Tucker)



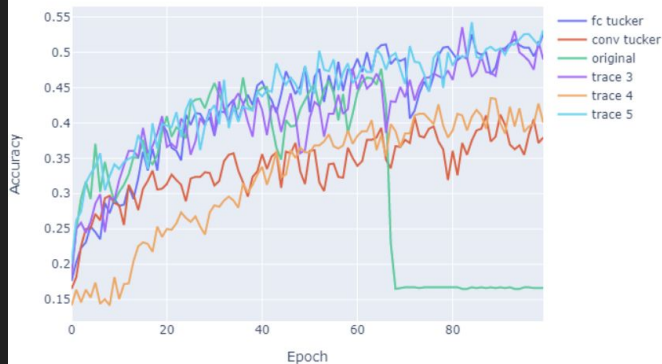
$$A(i, j, k) \approx \sum_{\alpha, \beta, \gamma} G(\alpha, \beta, \gamma) U_1(i_1, \alpha) U_2(i_2, \beta) U_3(i_3, \gamma)$$



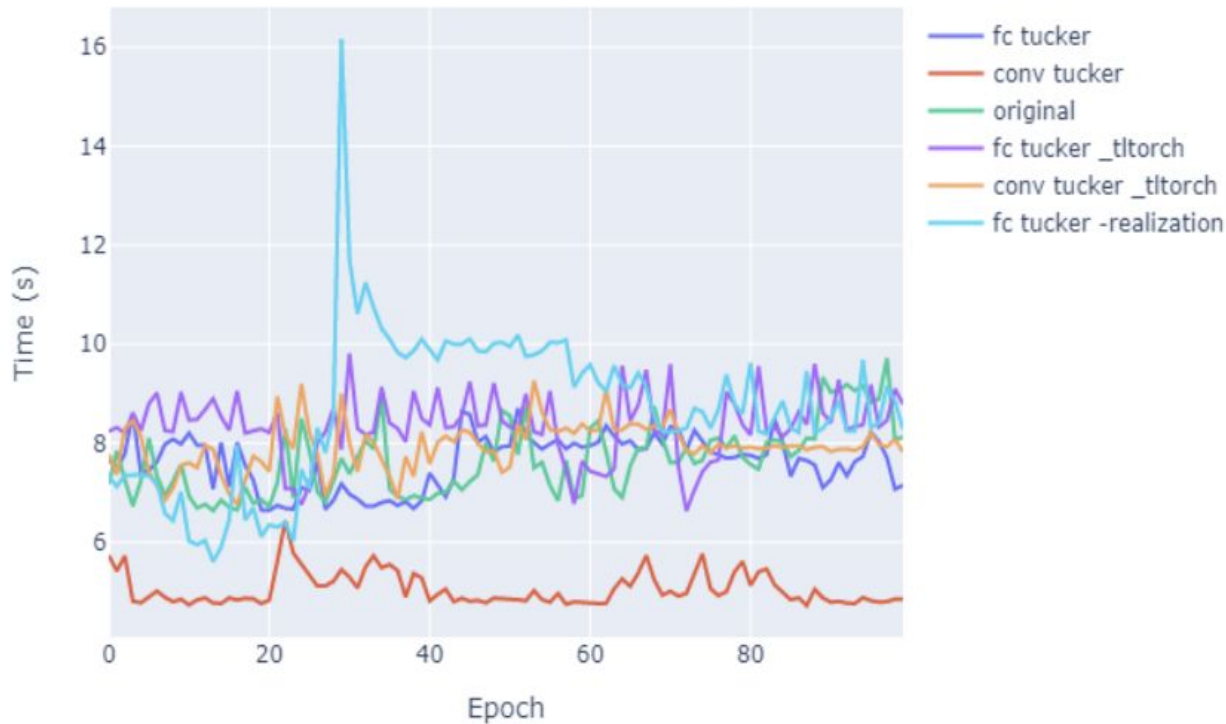
Метод 2: разложение Таккера



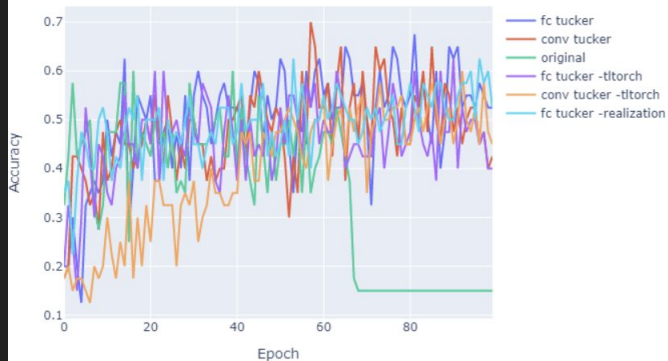
Training Accuracy



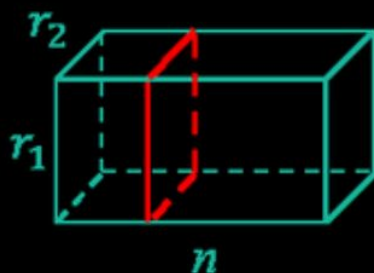
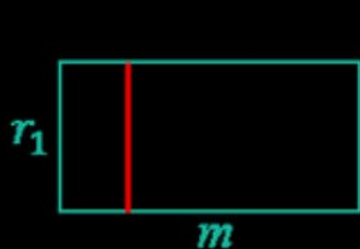
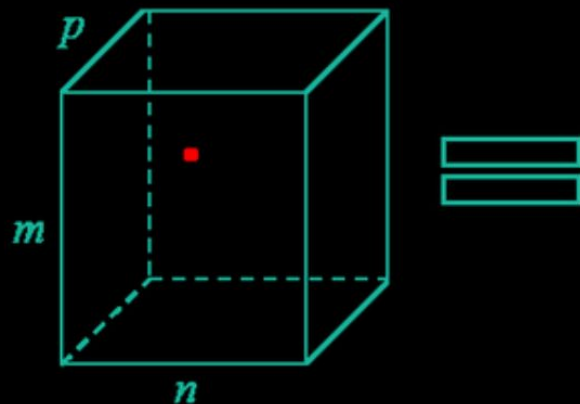
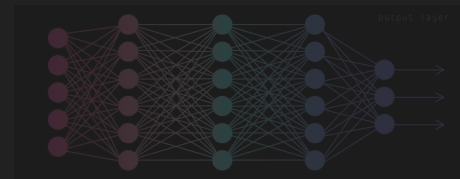
Time per Epoch



Testing Accuracy

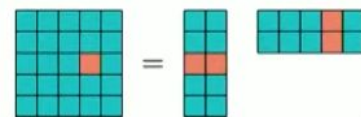


Tensor train decomposition



$$A(i_1, \dots, i_d) \approx G_1(i_1) \dots G_d(i_d)$$

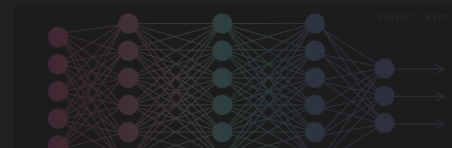
$G_k(i_k)$ имеет размер $r_{k-1} \times r_k$, $r_0 = r_d = 1$.



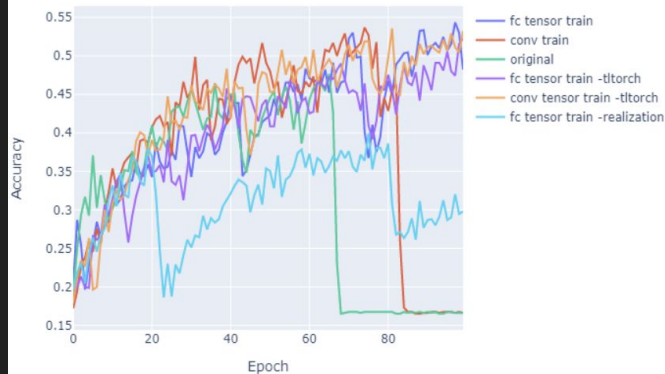
$$A_{3,4} = u_{3,:}^\top v_{4,:}$$

$$B_{2,3,1} = u_{2,:}^\top v_{3,:} w_{1,:}$$

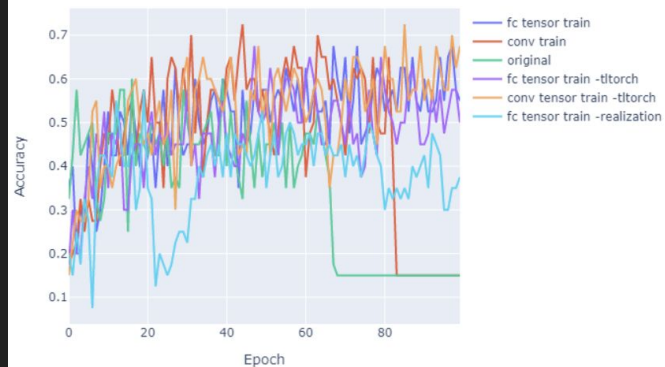
Метод 3: tensor train decomposition



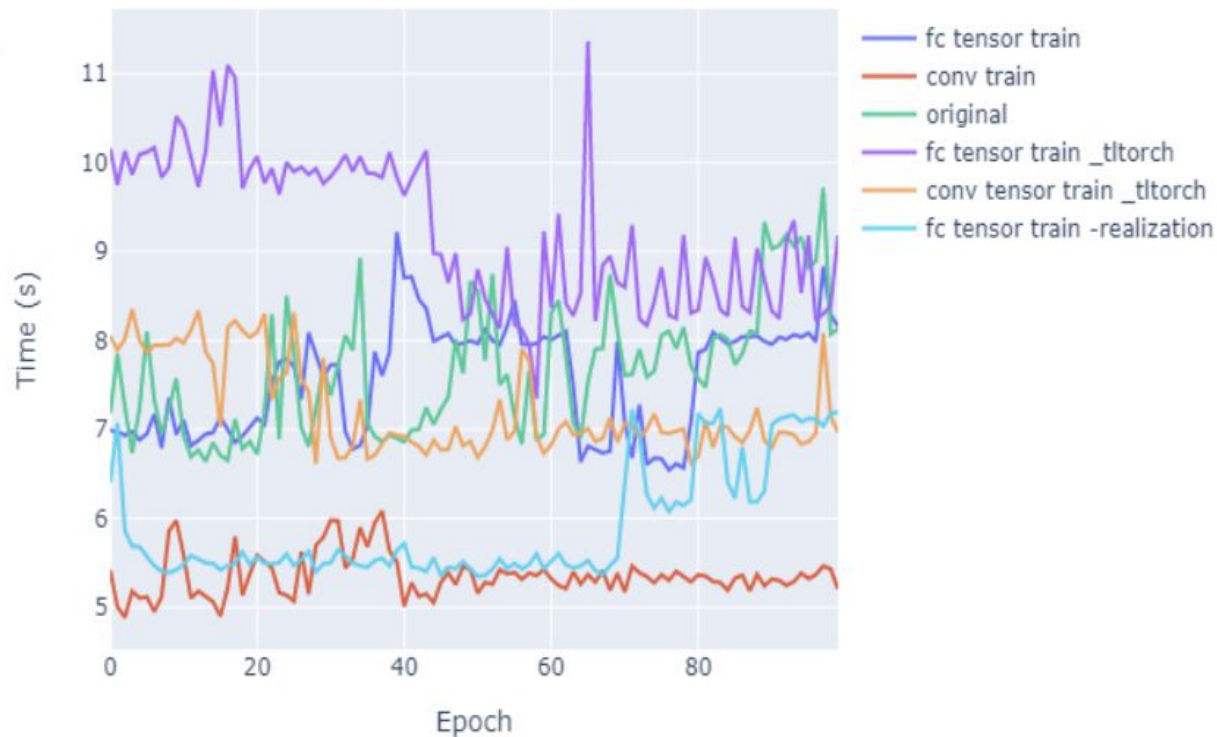
Training Accuracy



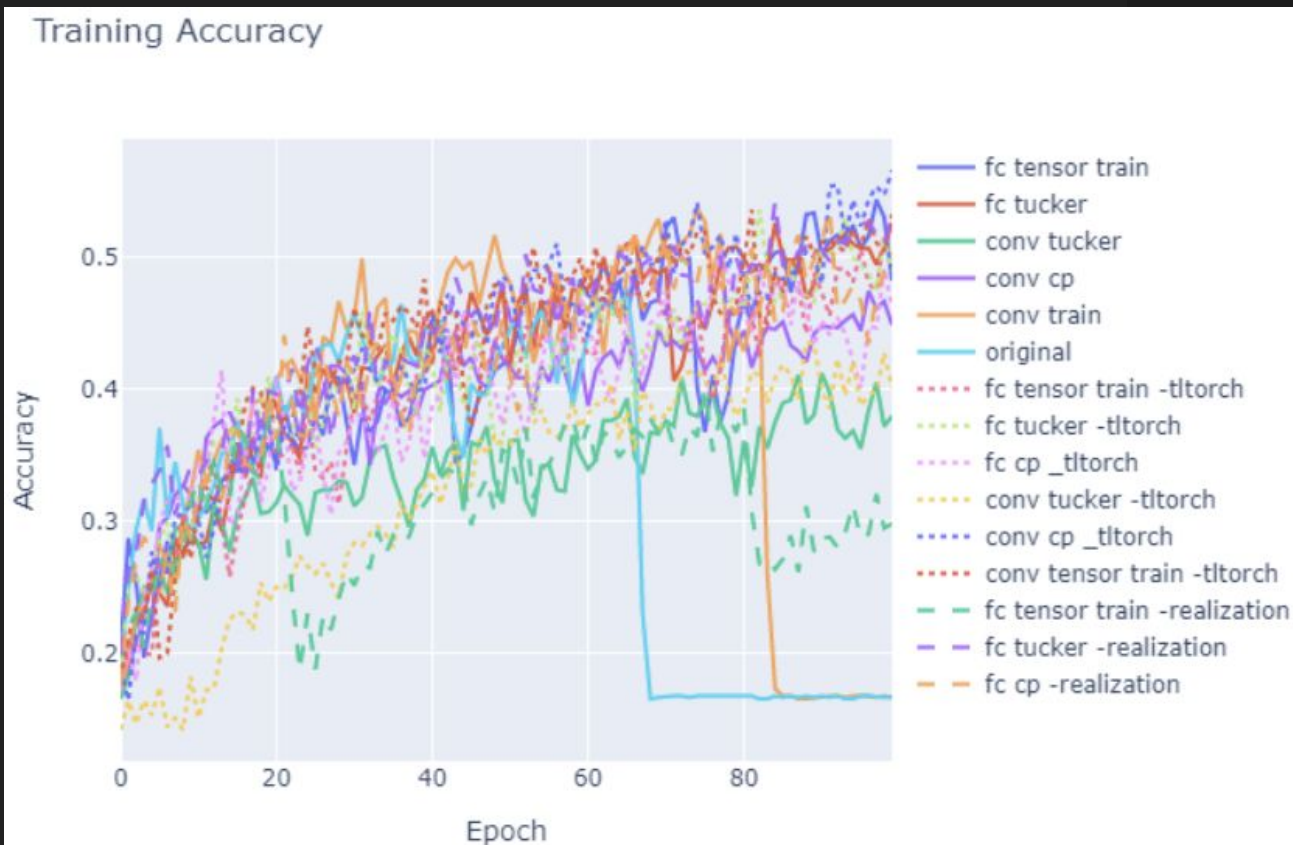
Testing Accuracy



Time per Epoch



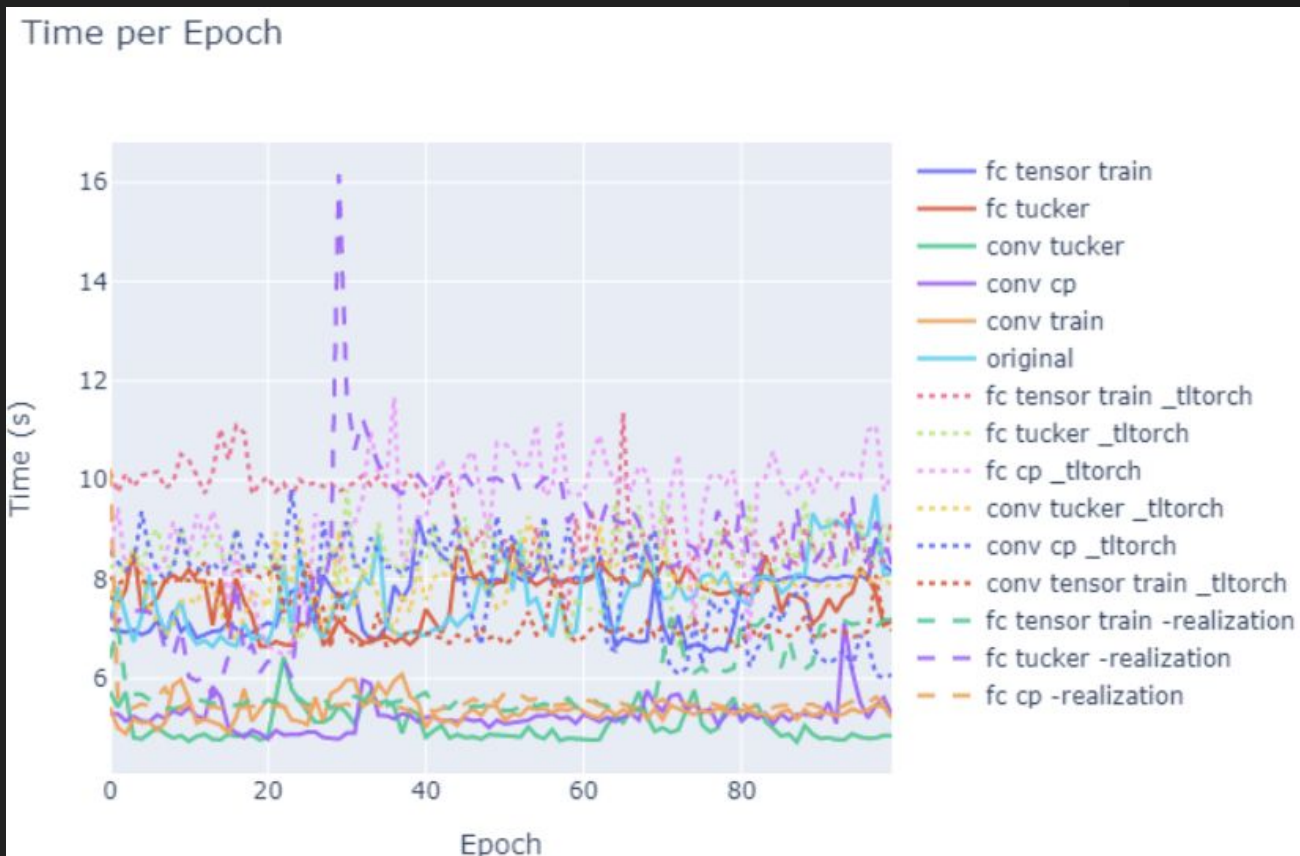
Общие графики



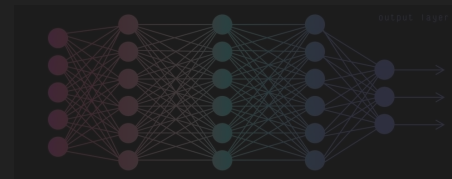
Общие графики



Общие графики



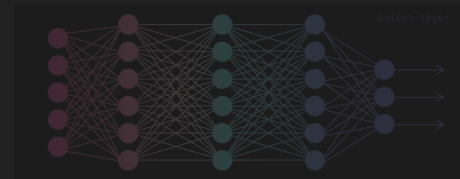
Еще один график



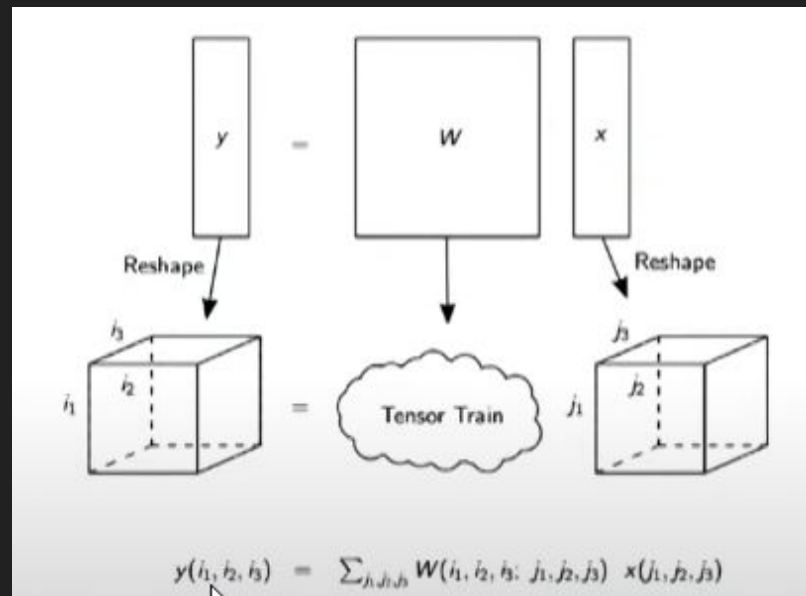
Выводы

- Декомпозиция сверточных слоев почти всегда показывает хорошие результаты и по качеству и по времени, в отличие от полносвязных
- Библиотека `tf.nn.conv2d` является самой интуитивно понятной, выдает достаточно высокое качество, но уступает по времени
- Нейронки с декомпозицией более устойчивы, нежели без при большом количестве эпох обучения
- Многое зависит от архитектуры модели и от подбора ранга в каждом из методов

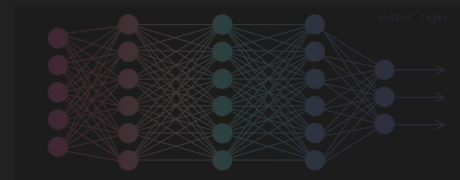
Нереализовано



- Изменение размеров векторов ответов и признаков
- Не до конца изучены зависимости
- Как влияет одновременное применение к полносвязным и сверточным одного метода (а микс разных методов?)
- Способы подбирать ранг более “умным” способом



Полезные ссылки



Ссылка на рабочую версию программы: <https://github.com/dmsy4/aim-nla-tensor-decompositions>

Другие полезные ссылки:

- V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, V. Lempitsky, *Speeding-up convolutional neural networks using fine-tuned cp-decomposition*, URL: <https://arxiv.org/pdf/1412.6553.pdf>, 2015.
- Yu Pan, Maolin Wang, Zenglin Xu, *TedNet: A Pytorch Toolkit for Tensor Decomposition Networks*, URL: <https://arxiv.org/pdf/2104.05018.pdf>, 2021.
- <https://jacobgil.github.io/deeplearning/tensor-decompositions-deep-learning>
- <https://tensorly.org/stable/index.html>

Спасибо за внимание!

