Использование ACM при сжатии линейных слоёв NN

Команда “Название временное”
Якуб Еналиев, Михаил Краснов

ACM - autocorrelation matrix

Постановка задачи

Использовать умножение на матрицу автокорреляции для уменьшения ошибки сжатия линейных весов моделей NN при малоранговой аппроксимации с помощью SVD.

Гипотеза: умножение на ACM будет минимизировать ошибку результата matvec вида y = xW

Ответственный: Якуб

Постановка задачи

Пусть мы хотим минимизировать ошибку сжатия под действием линейного слоя на одномерный вектор. Будем уменьшать математическое ожидание квадрата нормы $E_y||y - y_{compr}||_2^2$

Пусть $W - C_k = P$, где P – матрица ошибок сжатия, C_k – сжатая матрица, x – вектор, тогда:

$$E_y||y - y_{compr}||_2^2 = E_x||x(W - C_{compr})||_2^2 = E_x||xP||_2^2 = E_x(\sum^m \sum^m x_i x_j p_i p_j^T), \quad (1)$$

Сделаем замену $Q = R_{XX}^{\frac{1}{2}} W$, $Q_C = R_{XX}^{\frac{1}{2}} C_k$. Получаем:

$$arg\min||Q - Q_C||_F^2 \quad (2)$$

По т. Эккарта – Янга наилучшая малоранговая аппроксимация получается с помощью SVD разложения, тогда с учетом свойства нормы запишем:

$$C_k = (R_{XX}^{\frac{1}{2}})^{-1} Q_C = (R_{XX}^{\frac{1}{2}})^{-1} U_{:,:k} \Sigma_{:k,:k} V^*_{:k,:}, \text{ где } U\Sigma V^* = SVD(Q) \quad (3)$$

Ответственный: Якуб

Вычисление матрицы автокорреляции

Особенности:

(1) Нужно получить собственно матрицу автокорреляции; $O(n \log n)$

(2) Далее нужно найти из (1) корень; $O(n^2)$

(3) И найти обратную матрицу к матрице результата (2) $O(n^2)$

Ответственный: Якуб

Оптимальные вычисления матрицы автокорреляции

Вычислялась с использованием конволюции

```
ln = 2 * x.shape[0] - 1
Rxx = np.convolve(x,np.conj(x)[::-1])[ln//2 : ]
Rx = toeplitz(Rxx,np.hstack((Rxx[0],Rxx[1:])))
```

✓ 0.2s Python

Так же пробовал напрямую обратное преобразование Фурье, как в статье на Википедии, но из-за ошибок округления матрица выходила не положительно определённой (были очень малые комплексные слагаемые, из-за которых и возникали ошибки);

Ответственный: Якуб

Вычисление корня и обращение матрицы автокорреляции

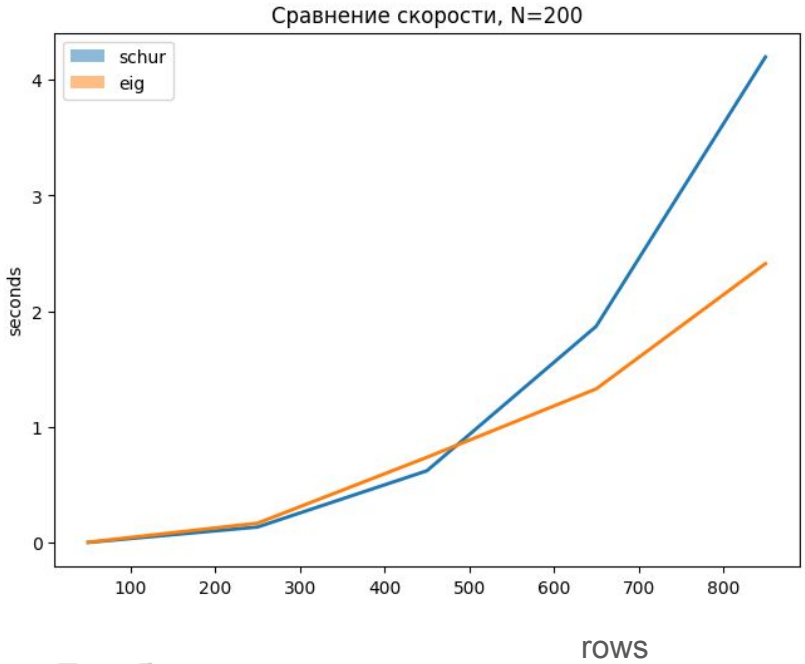
С использованием разложения Шура:

```
scipy.linalg.sqrtm(Rx)
```

По разложению по собственным числам, полученным с помощью преобразований Гивенса :

```
np.linalg.eig(Rx)
```

было запущено в Коллабе;



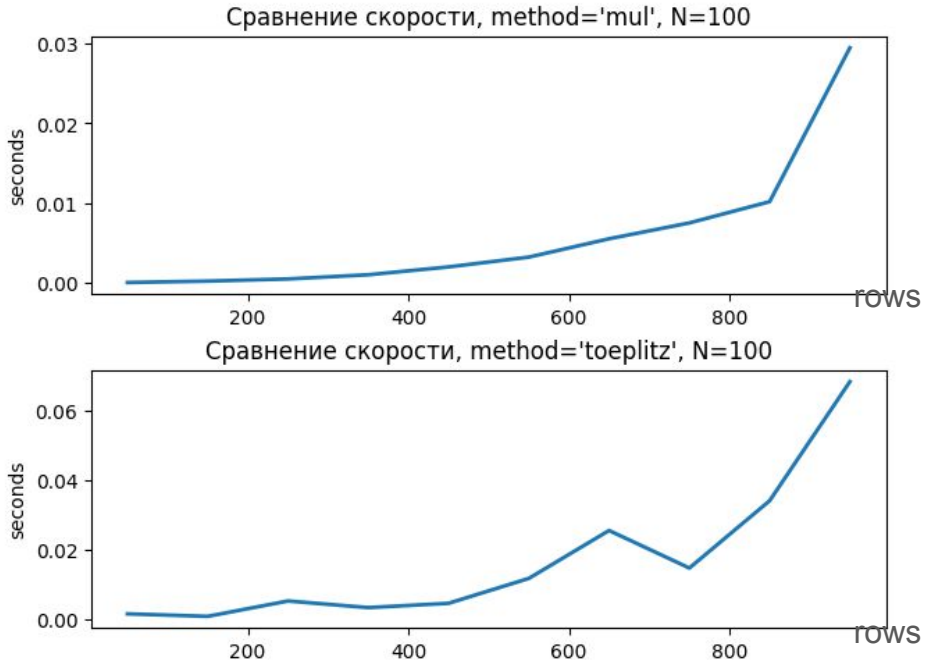
Ответственный: Якуб

Умножение на матрицу автокорреляции

scipy.linalg.matmul_toeplitz

Обычный matmul (mul)

было запущено в Коллабе;



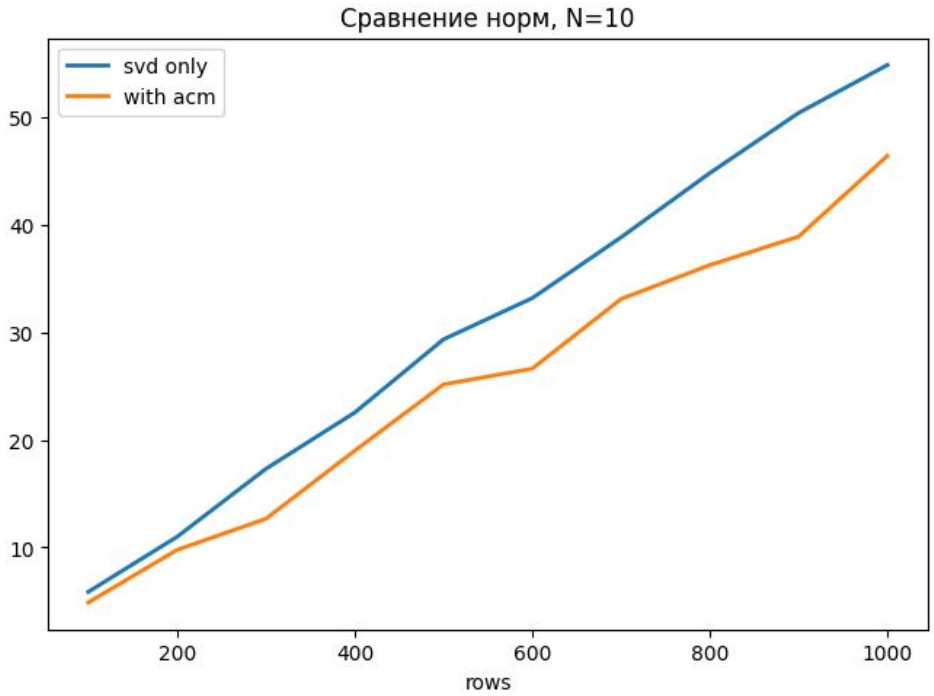
Ответственный: Якуб

Сравнение с обычным SVD

Сравнивалось для ранга аппроксимации x.rows // 5 ;

По оси абсцисс указано количество строк в случайной матрице, подвергающейся компрессии;

Чем больше строк, тем лучше результат



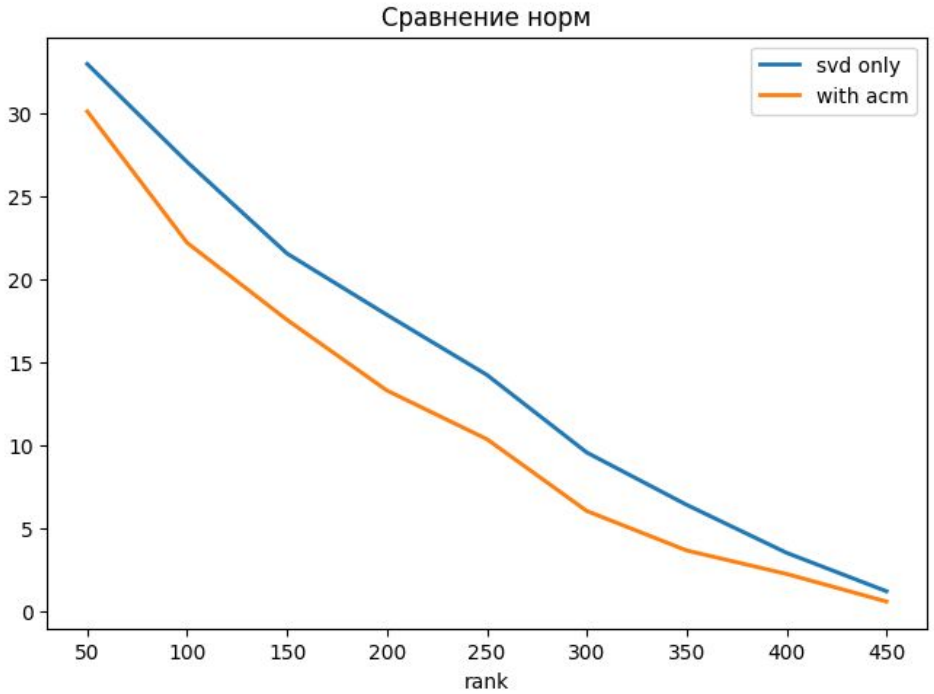
Ответственный: Якуб

Сравнение с обычным SVD

Сравнивалось для случайной матрицы из 500 строк;

По оси абсцисс отложен ранг матриц аппроксимации;

Чем больше сжатие, тем заметнее улучшение



Ответственный: Якуб

Проверка на реальных данных

В качестве модели была выбрана T5 переделанная для предсказания временных рядов

https://huggingface.co/amazon/chronos-bolt-base

Она была протестирована на двух датасетах:

https://github.com/mbohlkeschneider/gluon-ts/tree/mv_release/datasets - данные такси, 2000 рядов, длинной 1400

https://zenodo.org/records/4656042 - данные по банкам, всего 70 рядов

Ответственный: Михаил

Проверка на реальных данных и модели

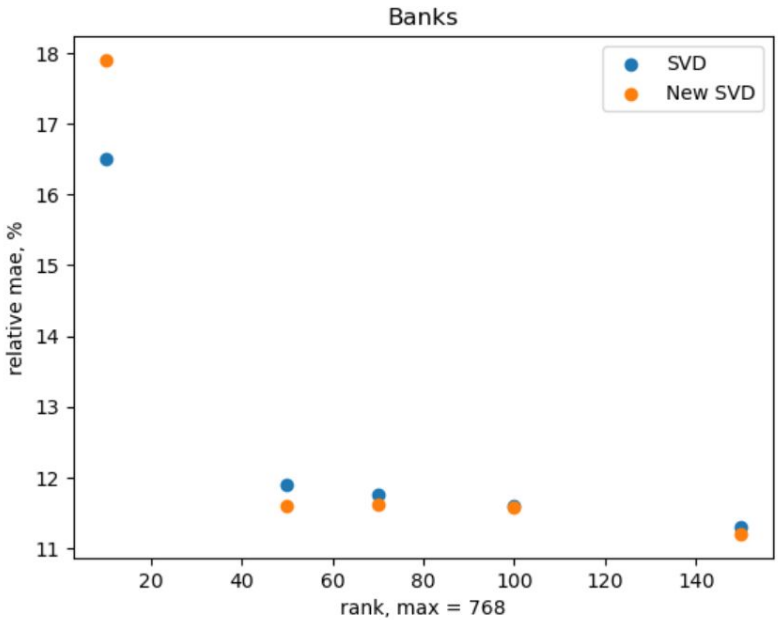
Методика проверки: проходимся окном размером 512 (если позволяет длина ряда) и предсказываем следующие 10 значений.

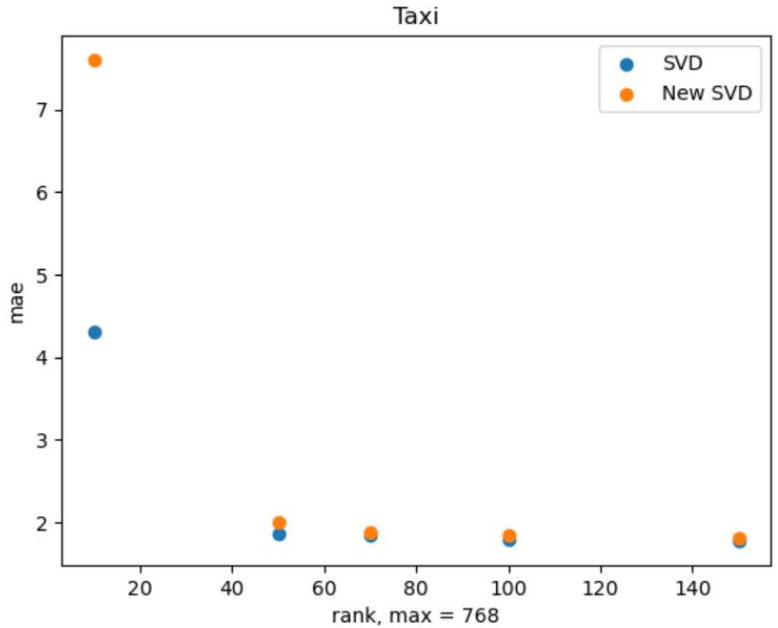
Упрощенный алгортим преобразования модели:

1) Проходимся по данным, итеративно строя матрицу автокорреляции для инупотов линеныйх слоев трансформеров
2) Делаем разложение и превращаем один линейый слой в два слоя. Здесь была встречена проблема больших чисел обусловленности матрицы автокорреляции - добавлял возмущение на диагонали чтобы повысить стабильность
3) Измеряем качество преобразованной моделью

Ответственный: Михаил

Проверка на реальных данных и модели





Вывод: для конкретной модели и данных датасетов, подход не показывает значительного улучшения по сравнению с классическим SVD
https://github.com/mishakrasnov/ai_masters_nla_project

Выводы

Умножение на АСМ уменьшает норму ошибки аппроксимации

Попробовать найти более оптимальные методы обращения матриц Тёплица

Ссылка на репозитории

https://github.com/Watashicuvu/svd-acm/tree/main

https://github.com/mishakrasnov/ai_masters_nla_project

Литература

1. QERA: an Analytical Framework for Quantization Error Reconstruction
2. Robert M. Gray,”Toeplitz and circulant matrices – an overview”,Department of Electrical Engineering,Stanford University,Stanford 94305,USA.↗
3. Viswanathan M. Digital modulations using Python. – Mathuranathan Viswanathan, 2019. – С. 8-3.