Sparse Low-rank Adaptation of Pre-trained Models

Команда:

Семикрас Александр
Пшеницын Артем
Словеснов Максим

NLA team, 2023

Превью

Для предварительно обученных моделей часто используется популярный метод адаптации низкого ранга LoRa. Но данный метод реализован с фиксированным и неизменяемым внутренним рангом, который не всегда может быть идеальным выбором.

- Задача:
  проверить эффективность **SoRa** слоев для fine tuning’a предварительно обученных моделей
- Мера качества:
  вычислительная сложность (=занимаемая память) + скорость работы + ориентированная на задачу метрика

LoRa

В LoRa предварительно подготовленные веса заморожены, а обучаемые модули LoRa представляют собой матрицы разложения низкого ранга:

$$\mathbf{W}_u \in \mathbb{R}^{p \times r} \quad \mathbf{W}_d \in \mathbb{R}^{r \times q} \quad \mathbf{\Delta} = \mathbf{W}_u \mathbf{W}_d \in \mathbb{R}^{p \times q}$$

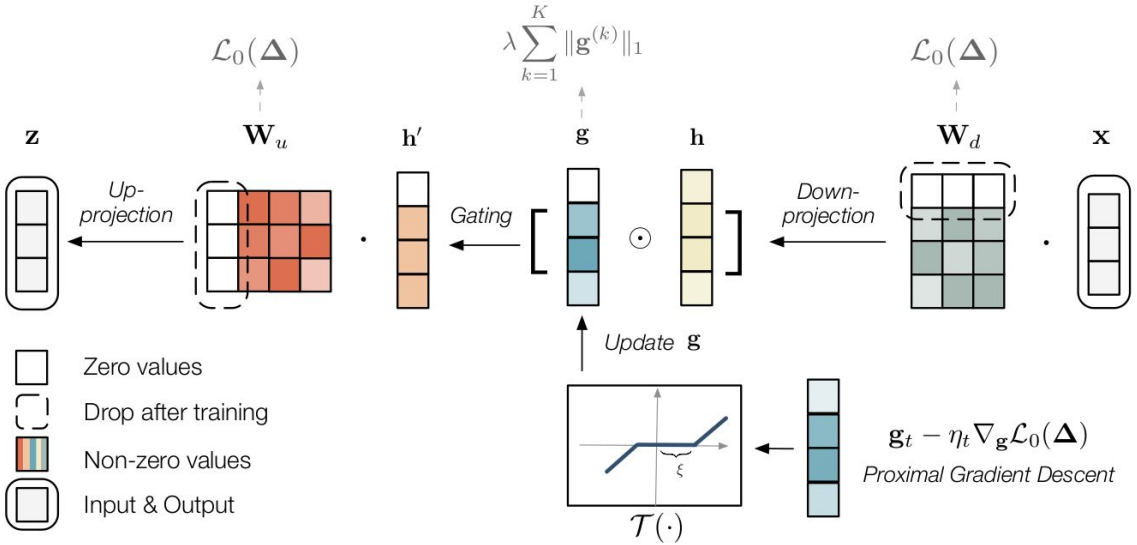
Таким образом, выходные данные текущего слоя могут быть представлены в виде:

$$\mathbf{y} \leftarrow \mathbf{W}_0 \mathbf{x} + \mathbf{W}_u \mathbf{W}_d \mathbf{x},$$

*Изначально матрица **Wu** инициализируется нулями, а **Wd** нормальным распределением, что соответствует нулевой начальной поправке к пре-тренированной модели

SoRa

Ключевая идея SoRA заключается в динамической настройке внутреннего ранга в процессе обучения с помощью разреженного стробирующего модуля.



SoRa

Мы оптимизируем матрицы нисходящей и восходящей проекции с помощью методов стохастического градиента, как в LoRa, в то время как каждый элемент g обновляется способом, способствующим разреженности:

$$\mathbf{g}_{t+1} \leftarrow \mathcal{T}_{\eta_t \cdot \lambda}(\mathbf{g}_t - \eta_t \nabla_{\mathbf{g}} \mathcal{L}_0(\mathbf{\Delta}_t)),$$

$$\mathcal{T}_\xi(x) := \begin{cases} x - \xi, & x > \xi \\ 0, & -\xi < x \leq \xi \\ x + \xi, & x \leq -\xi \end{cases}$$

Реализация SoRa-модуля в PyTorch

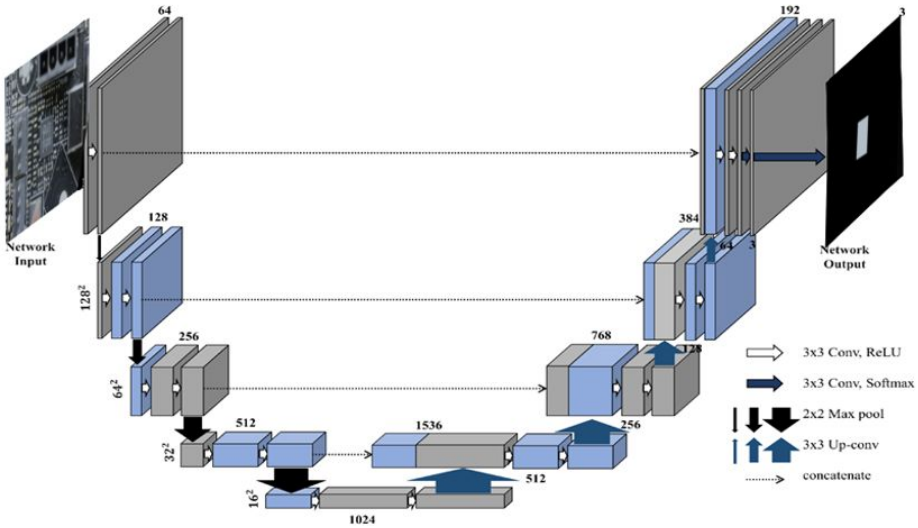
В нашей реализации, SoRa-модуль - это обертка над полносвязным слоем

В обычном режиме он функционирует соответствующим образом, но при активации переходит в SoRa-режим - создает соответствующие матрицы, и зануляет вычисление градиента полносвязного слоя

При деактивации, произведение новых матриц прибавляется к полносвязному слою, а сами они удаляются

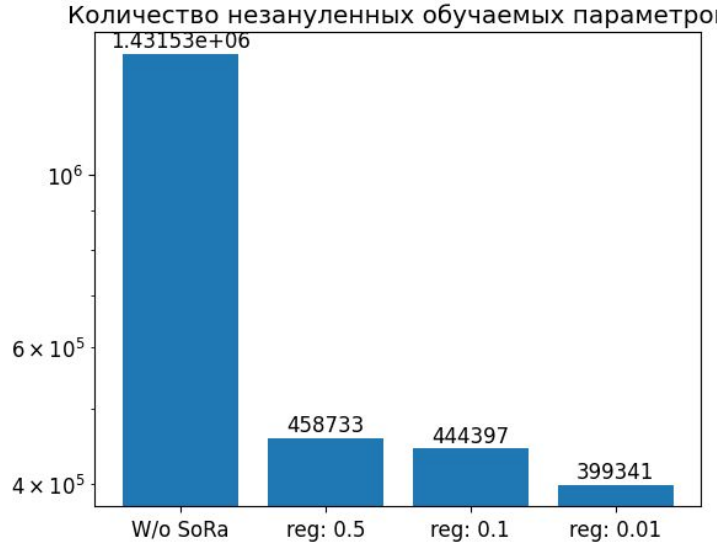
Использование SoRa модуля в CV

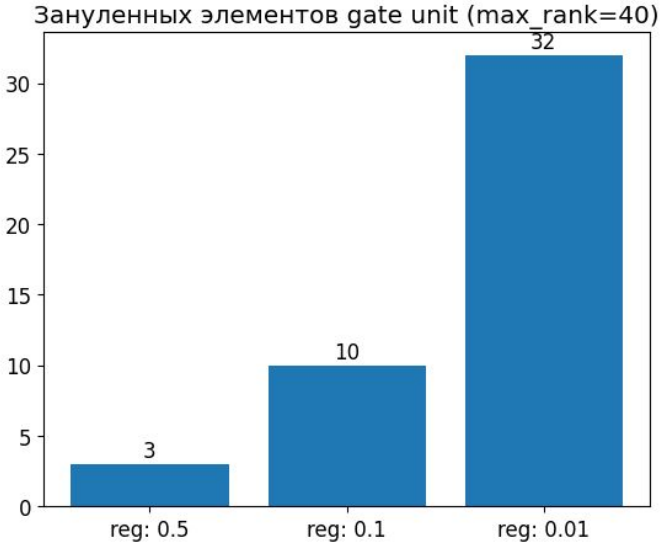
В качестве тестового примера использования модуля была выбрана нейросеть U-net, с полносвязным “бутылочным горлышком”, размера 1024×1024.



Тестовые датасеты:
egohand для обучения, и
gtea для fine tuning’a.

Максимальный ранг
малоранговой
аппроксимации выбран 40.


Количество незануленных обучаемых параметров
1.43153e+06
458733
444397
399341
W/o SoRa
reg: 0.5
reg: 0.1
reg: 0.01


Зануленных элементов gate unit (max_rank=40)
3
10
32
reg: 0.5
reg: 0.1
reg: 0.01


Dice loss на обучающей выборке
w/0 sora
reg: 0.5
reg: 0.1
reg: 0.01
Dice Loss
Количество итераций


Dice loss на тестовой выборке
w/0 sora
reg: 0.5
reg: 0.1
reg: 0.01
Dice Loss
Количество итераций

SoRA for Transformers

В качестве модельной задачи была выбрана задача саммаризации текстов на датасете medium-articles (190k):

…

The Sahel is one of Africa’s poorest and most fragile regions. Marked by chronic poverty, instability and high levels of gender inequality, it is very vulnerable to violence and conflict from extremist and criminal organisations.

…

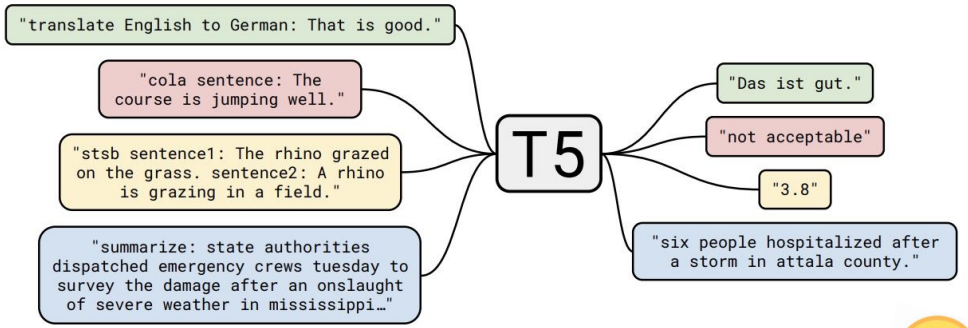
target:

The British Army has arrived in Mali: Here’s what you need to know about the deployment

T5-small (60M)

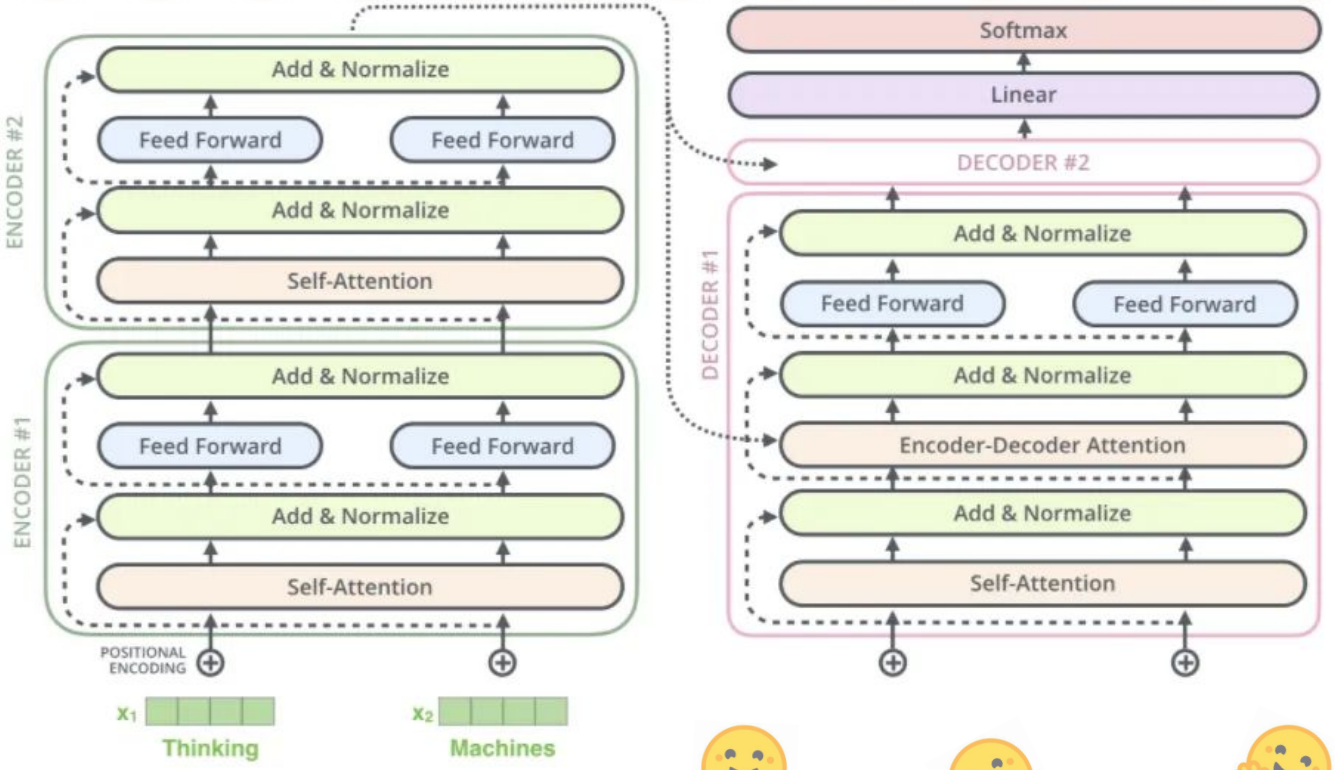
Заменялись все слои в Attention блоках и в энкодере, и в декодере (по 4 FC слоя в каждом), а также в блоках DenseReLU

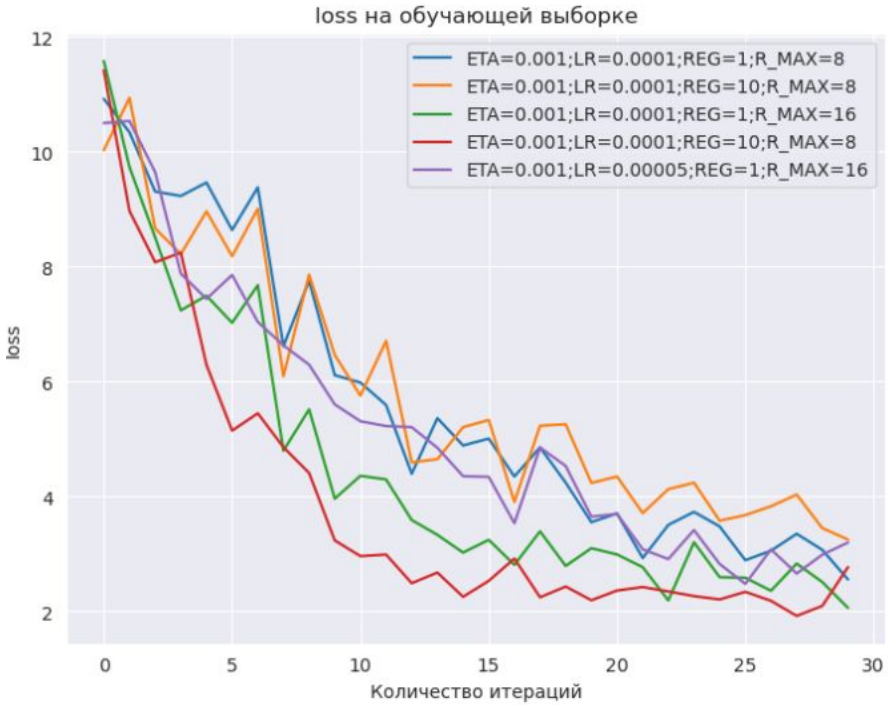
Максимальный ранг был выбран 8 и 16

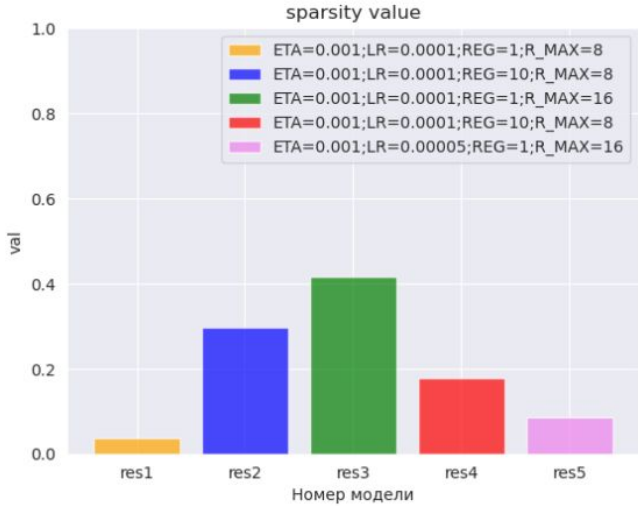


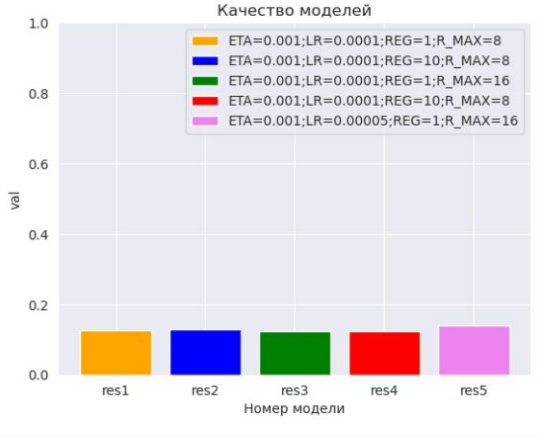
T5

По архитектуре - довольно ванильный трансформер, без особых изощрений, (кроме кормления данных и в целом дизайна претрейна)




loss на обучающей выборке
ETA=0.001;LR=0.0001;REG=1;R_MAX=8
ETA=0.001;LR=0.0001;REG=10;R_MAX=8
ETA=0.001;LR=0.0001;REG=1;R_MAX=16
ETA=0.001;LR=0.0001;REG=10;R_MAX=8
ETA=0.001;LR=0.00005;REG=1;R_MAX=16
loss
Количество итераций


sparsity value
ETA=0.001;LR=0.0001;REG=1;R_MAX=8
ETA=0.001;LR=0.0001;REG=10;R_MAX=8
ETA=0.001;LR=0.0001;REG=1;R_MAX=16
ETA=0.001;LR=0.0001;REG=10;R_MAX=8
ETA=0.001;LR=0.00005;REG=1;R_MAX=16
val
res1
res2
res3
res4
res5
Номер модели


Качество моделей
ETA=0.001;LR=0.0001;REG=1;R_MAX=8
ETA=0.001;LR=0.0001;REG=10;R_MAX=8
ETA=0.001;LR=0.0001;REG=1;R_MAX=16
ETA=0.001;LR=0.0001;REG=10;R_MAX=8
ETA=0.001;LR=0.00005;REG=1;R_MAX=16
val
res1
res2
res3
res4
res5
Номер модели

Что еще можно было бы реализовать:

1. Реализовать архитектуру refined U-net, и попробовать добиться приемлемого качества сегментации на полноценной версии датасетов, перед использованием fine tuning’a
2. Упростить интерфейс для взаимодействия с SoRa - модулями, и их обновления
3. Каким-то образом написать свой трейнер на базе класса с HF…

Полезные ссылки:

1. https://arxiv.org/abs/2311.11696 - основная статья про SoRa
2. https://www.sciencedirect.com/science/article/pii/S092523122004696 - опорная статья для CV-модуля, с используемыми датасетами, и описанием архитектуры refined U-net
3. https://arxiv.org/abs/1910.10683v4 - T5
4. https://github.com/ShaeNaZar/AIM-numerical-linalg-project-2.git