УДК 004.852

SimplexLoRA : Expand important adapters

И. В. Шалыгин, Г. Р. Давыденко, А. С. Веприков
Московский физико-технический институт (национальный исследовательский университет)

В современном мире активно используются языковые модели и не менее важно их правильное дообучение. При файн-тюнинге мы решаем следующую задачу оптимизации:

$$\min_{\Delta\mathcal{W}}\{\mathcal{L}(W^1 + \Delta W^1, ..., W^n + \Delta W^n)\}\text{ , где }\Delta\mathcal{W} = \{\Delta W_i\}_{i=1}^n$$

Одной из наиболее распространенных техник дообучения является LoRA (low rank adaptation). В данном фреймворке адаптер состоит из малоранговых матриц A и B и задача оптимизации сводится к эквивалентной задаче поиска матриц малорангового приближения:

$$\min_{\mathcal{A},\mathcal{B}}\{\mathcal{L}(W^1 + A_1 \cdot B_1, ..., W^n + A_n \cdot B_n)\}\text{ , где }\mathcal{A} = \{A_i\}_{i=1}^n\text{, и }\mathcal{B} = \{B_i\}_{i=1}^n$$

Однако фреймворк LoRA требует много памяти для достижения точных результатов, потому что на все слои добавляются адаптеры одинаковых рангов, и интуиции, на какие слои ее необходимо добавлять. Поэтому в нашем исследовании мы представляем фреймворк SimplexLoRA, который обходит обе эти проблемы с помощью адаптивного изменения рангов адаптеров.

В предлагаемом методе мы вводим новые параметры веса адаптеров $\omega = \{\omega_i\}_{i=1}^n$. Вектор данных весов лежит на $(n-1)$-симплексе размера n, что означает следующие ограничения:

$$\omega \in \left\{ (\omega_1, ..., \omega_n) \mid (\omega_i \geq 0 \ \ \forall i) \wedge \left(\sum_{i=1}^{n} \omega_i = n \right) \right\}$$

Таким образом, модель получает дополнительный параметр веса модели, на основе значений которого мы в процессе обучения делаем изменение рангов соответствующих адаптеров. Выход i-ого слоя в фреймворке SimplexLoRA выглядит следующим образом и похож на стандартный выход слоя в LoRA:

$$h_i = W_i x + \omega_i A_i B_i x$$

Расширение и сжатие рангов адаптеров производилось с помощью выведенных формул с использованием QR и SVD разложений таким образом, чтобы минимизировать норму Фронебиуса разности между произведением изначальных матриц адаптера и произведением новых:

$$\|A_{\text{old}}B_{\text{old}} - A_{\text{new}}B_{\text{new}}\|_F \longrightarrow \min_{A_{\text{new}},B_{\text{new}}}$$

Для проверки нашего алгоритма на практике был реализованы методы работы с рангами матриц, переписан код Adam оптимизатора и код класса LoRA библиотеки PEFT. Реализованный алгоритм файн-тюнинга выглядит так:

0. Все ранги адаптеров равны гиперпараметру r_0, веса — $\omega_i^{(0)} = 1$
1. После шага в оптимизаторе делаем проекцию весов $\omega^{(k)}$ на $(n-1)$-симплекс с суммой координат n
2. Через фиксированное количество шагов T в соответствии с полученными весами обновляем ранги адаптеров по правилу $r_i^{(k+1)} = \left\lfloor \omega_i^{(k)} \cdot r_0 \right\rfloor$
3. Пункты 1 и 2 повторяем K раз
4. Запускаем стандартное обучение LoRA с подобранными рангами

Эксперименты были проведены с моделями deberta и llama на бенчмарке GLUE. Таблицы с результатами и метриками SimplexLoRA и LoRA для сравнения будут представлены на защите нашей работы.