Вопросы по докладу

Зиманов Алихан

20 февраля 2023 г.

Вопрос. Что такое fine-tuning? Что такое parameter-efficient fine-tuning и зачем он нужен?

Ответ. Fine-tuning — использование уже обученной, но на другую задачу модели как инициализацию весов для новой модели, решающей текущую задачу. Parameter-efficient fine-tuning — fine-tuning, который обучает лишь маленькую часть весов модели, тем самым позволяя делить большую часть исходной модели между разными задачами и меняя между ними только эту малую часть обучаемых весов.

Вопрос. Что такое continual learning и почему его не стоит использовать для решения нескольких задач в случайном порядке?

Ответ. Continual learning это подход обучения одной модели для решения нескольких задач. В нём одной и той же модели последовательно даются задачи и мы дообучаем её решать текущую задачу. Этот плохо подходит для решения задач в случайном порядке потому что когда мы обучаем нашу модель на новую задачу, её способность решать предыдущую утрачивается, что приводит к заметному ухудшению качества модели.

Вопрос. В чём основная идея adapter tuning и как его применить к трансформеру?

Ответ. Adapter tuning заключается в том, что мы добавляем adapter слои между слоями исходной модели и далее обучаем только веса этих добавленных слоев, а веса исходной модели не меняем. Adapter слой состоит из трех преобразований: проекция входного вектора в меньшее пространство, нелинейность, проекция полученного вектора обратно в пространство исходной размерности. Для применения adapter tuning в трансформере надо вставить adapter слои после каждого feed-forward слоя.

Вопрос. В чём заключается идея prefix-tuning? Как стоит обучать префиксы-контексты?

Ответ. В моделях NLP часто используются векторы контекста, которые призваны подсказывать модели следующие токены. Prefix-tuning утилизирует эту идею и с помощью специального префикса входных данных подсказывает модели какую именно задачу она сейчас должна решать. Если обучать веса префиксов P_θ явно в лоб, то получится слабенькое качество, поэтому можно заменить их скрытыми префиксами P'_θ и небольшой нейронной сетью MLP_θ так, что $P_\theta[i,:] = MLP_\theta(P'_\theta[i,:])$ и теперь обучать P'_θ и веса у MLP_θ.

Вопрос. Выпишете функцию, которую мы хотим максимизировать во время fine-tuning и объясните все обозначения (по-хорошему, формулировку из метода LoRA).

Ответ. Функция выглядит как:

$$\max_{\Delta\Phi} \sum_{(x,y)\in Z} \sum_{t=1}^{|y|} \log\left(P_{\Phi_0+\Delta\Phi}(y_t|x, y_{<t})\right).$$

- Φ_0 – веса исходной модели;
- $\Delta\Phi$ – изменение весов исходной модели во время fine-tuning;
- Z – датасет новой задачи и $(x, y) \in Z$ – пара входных и выходных данных задачи;
- $P_{\Phi_0+\Delta\Phi}$ – модель после обучения на новую задачу и $P_{\Phi_0+\Delta\Phi}(y_t|x, y_{<t})$ – соответствующая вероятность получить токен y_t при условии токенов из x и $y_{<t}$.

Вопрос. В чём заключается метод LoRA для одной матрицы? К каким матрицам трансформера стоит применять этот метод?

Ответ. Допустим у нас есть матрица W_0 из исходной модели. Добавим в модель рядом с W_0 две низкоранговые матрицы A и B так, чтобы при входе x данная часть модели выдавала $W_0x + BAx$ и будем обучать только эти добавленные матрицы A и B. Матрицы A и B играют роль низкорангового приближения матрицы ΔW, которая является изменением матрицы W_0 во время честного fine-tuning. В трансформере эти низкоранговые приближения стоит применять к матрицам слоев self-attention.