Школа анализа данных
Машинное обучение, часть 2
Теоретическое задание №1, часть 2

Кошман Дмитрий

Задача 4

LambdaNets

1. Какую задачу решают авторы?

При применении слоев self-attention к длинным последовательностям авторы сталкивались с большими затратами оперативной памяти, что ограничивало эффективное обучение моделей. Авторы предлагают модификацию архитектуры self-attention под названием lambda layers, которая позволит обучать большие батчи длинных последовательностей.

2. Какова основная идея предлагаемого авторами решения поставленной задачи?

Архитектура лямбда слоев строится на применении к существующему слою self-attention простой идеи: поскольку матричное умножение ассоциативно:

$$(AB)C = A(BC),\ A \in \mathbb{R}^{w \times x}\ B \in \mathbb{R}^{x \times y}\ C \in \mathbb{R}^{y \times z},$$

то у нас есть возможность выбирать более эффективный порядок вычислений, причем в первом случае возникает промежуточная матрица размера $w \times y$, а во втором - $x \times z$. Именно в случае с слоем self-attention вычисления выглядят так:

$$Y = \sigma(QK)V$$

$$Q \in \mathbb{R}^{b \times n \times k},\ K \in \mathbb{R}^{m \times k},\ V \in \mathbb{R}^{m \times v}$$

Получается, что в таком случае возникает промежуточная матрица QK размера $b \times n \times m$, и при работе с длинными последовательностями(n) и большим контекстом(m), мы ограничены в размерах батча(b). Если же как-то преобразовать вычисления так, что сначала перемножаются K и V, то затраты памяти будут не так драматично зависеть от размера батча.

Дальше возникает идея факторизации - похожая на факторизацию свертки в depthwise и pointwise слои. Здесь факторизуется матрица QK, которая в каждой ячейке одновременно содержит информацию о релевантности контекста к подаваемому запросу и по содержанию, и по относительным позициям. Хочется разделить ее на матрицу, в которой хранится релевантность позиций в последовательности - $\lambda^p \in \mathbb{R}^{n \times k \times v}$, она будет одинаковая для каждой последовательности в батче, и на матрицу релевантности содержания - $\lambda^c \in \mathbb{R}^{k \times v}$, она не будет зависеть от размера контекста.

Посмотрим поближе на разницу в вычислениях self-attention и lambda слоев. Они оба получают на вход батч последовательностей $X \in \mathbb{R}^{b \times n \times d}$, контекст $C \in \mathbb{R}^{m \times d}$ и имеют параметры $W_Q \in \mathbb{R}^{d \times k}, W_K \in \mathbb{R}^{d \times k}, W_V \in \mathbb{R}^{d \times v}$. Помимо этого lambda слой требует предпосчитанные необучаемые

позиционные эмбединги $E \in \mathbb{R}^{n \times m \times k}$. Это добавляет сложности модели, но позволяет внести в модель априорные знания о структуре последовательностей. Все вычисления выглядят так:

$$Q = XW_Q, K = CW_k, V = CW_V$$

$$Y_{attention} = \sigma(QK)V$$

$$Y_{lambda} = \lambda Q = (\lambda_p + \lambda_c)Q = (EV + V\sigma(K))Q$$

3. Каковы результаты, полученные авторами?

1. Сложность вычислений слоя self-attention: первое умножение - $\Theta(bnmk)$, второе - $\Theta(bnmv)$, всего $\Theta(bnm(k+v))$, затраты памяти - $\Theta(bnm)$. Сложность lambda слоя: вычисление EVQ - $\Theta(nmkv + bnkv) = \Theta(nkv(m+b))$, вычисление $V\sigma(K)Q$ – $\Theta(mvk + bnkv)$, всего $\Theta(nkv(m+b))$. Затраты памяти - $\Theta(knm)$. Соотношение старых затрат к новым: $\frac{bm(k+v)}{kv(m+b)} \approx \frac{b(k+v)}{kv} \approx \frac{b}{k}$ по вычислениям, $\frac{b}{k}$ по памяти. То есть в случаях, когда размер батча превышает размерность внутреннего представления элементов последовательности, получаем значимое ускорение и уменьшение памяти.
2. Поскольку затраты памяти не зависят мультипликативно от размера батча, можем использовать большие батчи.
3. Предложена аналогичная multi-head attention вариация multi-query lambda, которая конкатенирует выход нескольких уменьшенных слоев, что позволяет еще больше уменьшить затраты.
4. На датасете ImageNet - в десятки раз меньшее потребление памяти, чем attention, в полтора раза ускорена поэкземплярное обучение, на 1% увеличена top-1 accuracy.