Отчёт по финальному проекту курса NLP

Май 2024

Содержание

1. Введение *[аналог abstract, рассказать почему работа актуальна, чем уникальна]*

   1.1 Команда

2. Цели *[чего хотим достичь в рамках проекта]*
3. Сопутствующая работа *[нужно рассмотреть существующие к достижению поставленной цели, дать ссылки на источники (где сказано про эти подходы)]*
4. Исходные данные *[рассказ про саму модель для дальнейшей работы, а также про данные, используемые в процессе (откуда, что из себя представляют)]*
5. Метод *[подробно расписываем каким путём хотим получить результат (цель), сюда должна войти теория]*
6. Эксперимент

   6.1 Критерии *[критерии оценки качества результата (как валидации, так и теста)]*

   6.2 Инструкции *[инструкции для воспроизведения эксперимента]*

   6.3 Базовые подходы *[описание некоторого минимально сложного подхода, который давал бы минимально приемлемый по требованиям результат]*
7. Результаты *[что получилось в итоге, графики и т.д.]*
8. Интерпретация *[интерпретация результатов, т.е. как трактовать графики и др.]*
9. Заключение *[краткое summary проделанной работы]*

- Ссылки *[ссылки на использованную литературу]*

1 Введение

В результате значительного развития нейросетевых моделей, всё большей проблемой становится локальный запуск и, тем более, обучение. Модель облачных вычислений достаточно универсальна, но не лишена недостатков - самым простым из серьёзных недостатков облачной модели является необходимость наличия в достаточной степени широкого и надёжного канала связи. Не во всех местах использование облачных технологий возможно просто по той причине, что не везде есть достаточно дешёвая связь, а могут быть и случаи с полным отсутствием связи. Помимо этого, при использовании облачных технологий могут появляться заметные человеку задержки обработки. Решением подобного класса проблем является использование локальных вычислений. Однако, в случае локального запуска возникают проблемы в потенциальном недостатке вычислительной мощности. Таким образом, актуальными становятся рассмотренные в рамках данной статьи техники уменьшения размеров моделей, которые так же дадут значительное ускорение инференса. Таким образом, в результате получается нейросеть, которая может запускаться на значительно более слабых в вычислительном плане устройствах, чем исходная.

Данная работа интересна самим способом достижения результата. Чаще всего, уменьшение размеров модели достигается путём дистилляции и квантизации. В рамках данного проекта применён прунинг. В рамках поставленного эксперимента прунинг воспринимается как правильная инициация сети-ученика для дальнейшей дистилляции. Интересным фактом

является то, что полученная таким образом модель (меньшего размера), но сопоставимого по качеству с оригинальной сетью. В рамках статьи эксперимент поставлен над моделью архитектуры BERT.

1.1 Команда

Список участников проекта представлен в следующей таблице:

Участник	E-mail	Роль
Петров Александр Васильевич	petrov1c@yandex.ru	*{роль}*
Сентюрев Михаил Алексеевич	mixail_sen@outlook.com	*{роль}*

2 Цели

В рамках проекта поставлена цель показать на практике, что возможно достичь уменьшения размера модели архитектуры BERT без заметного ухудшения качества выдаваемого результата, то есть, получить в результате более быструю и компактную сеть без заметной потери в качестве.

3 Сопутствующая работа

Методов уменьшения размера моделей, помимо предложенного в данной работе, существует несколько. Наиболее популярные из них: дистилляция и квантизация. Эти подходы так же могут применяться вместе.

По технике дистилляции существует, к примеру, статья, в рамках которой была проделана работа по уменьшению размера оригинальной нейросети BERT. В результате была получена модель, названная DistilBERT. Авторы добились уменьшения числа параметров на 40%, сохранив при этом 97% исходного качества, как показывает бенчмарк GLUE. Более подробно про данный метод и результаты можно прочитать в статье авторов: https://arxiv.org/pdf/1910.01108.

По технике квантования существует, к примеру, эта статья, в рамках которой было выполнено обучение бинарных нейросетей на основе MNIST, CIFAR-10 и SVHN с получением уменьшения потребления вплоть до 32 раз (float переведён в bool) и ускорения вычислений вплоть до 7 раз (за счёт оптимизации вычислений на бинарных параметрах). При этом, была достигнута точность, достаточно близкая к исходной. Более подробно про данный метод и результаты можно прочитать в статье авторов: https://arxiv.org/pdf/1602.02830.

4 Исходные данные

К исходным данным для этой работы относится датасет (англо-русский параллельный корпус) от Yandex. Корпус содержит 1 миллион пар параллельных предложений на русском и английском языках, случайным образом выбранных из экспериментальных корпусов, собранных в 2011-2013 годах из параллельных документов, найденных в Интернете в автоматическом режиме. Размер архива с датасетом составляет 122 МБ, содержимое: два текстовых файла с предложениями на русском и английском языках, выровненными по номерам строк в кодировке UTF-8. Доступ к датасету можно запросить непосредственно у компании Yandex по этой ссылке: https://translate.yandex.ru/corpus.

В качестве исходный данных также был взят двуязычный вариант модели LaBSE, которая была разработана компанией Google. LaBSE - это BERT-подобная модель, разработанная для целей перевода между более чем 100 языками. Ключевым свойством архитектуры данной модели является то, что корпуса с разных языков отображаются в одно скрытое пространство таким образом, что близкие по смыслу высказывания отображаются в похожие по косинусной близости вектора. Благодаря такой архитектуре, стало возможным без значительных усилий создать вариант этой модели с меньшим числом поддерживаемых языком (и соответственно, меньшим размером модели) путём просто исключения из модели токенов, не относящихся к желаемым языкам. Именно таким путём была создана модель LaBSE-en-ru. Автор этой

модели оставил в ёё токенайзере только русские и английские токены, в результате чего модель уменьшилась в 4 раза, так как уменьшение количества токенов естественным образом слоя эмбеддингов и выходного слоя пуллинга. Нельзя не упомянуть, что автор LaBSE-en-ru также получил из неё дистиллированную версию, которая в русскоязычном сообществе хорошо известна под названием rubert-tiny2. Более подробно про эту модель автор рассказал в своей статье.

5 Метод

За основу в данной работе взята модель LaBSE-en-ru.

Если описывать процесс кратко, то од делится на 3 этапа: 1. уменьшение за счёт исключения из исходной модели части голов внимания (attention heads); 2. обучение полученной модели путём дистилляции знаний от оригинальной модели; 3. оценить качество полученной модели.

Теперь более подробно по каждому пункту. Для сокращения нотации, будет называть оригинальную модель (LaBSE-en-ru) - моделью-учителем, а получаемую в рамках данной работы модель - моделью-учеником.

Оригинальная модель имеет 12 слоев внимания по 12 голов в каждом. На первом этапе применяется разложение в ряд Тейлора, чтобы понять, какие головы оказывают меньше всего влияния на выход энкодера. Суть подхода в том, что нейронная сеть и функция потерь представляются обычной функцией (веса модели модели представляются переменными), к котрой далее применяется разложение в ряд Тейлора $L(w) = \sum_{p=0}^{p} \frac{L^{(p)}(w_0)}{p!}(w - w_0) + R_p(w)$ до первого порядка:

$$L(w = 0) = L(w_0) - w_0 \frac{\partial L}{\partial w}(w_0) + R_p(w)$$

$$\min |L(w = 0) - L(w_0)| = \min \left| w_0 \frac{\partial L}{\partial w}(w_0) \right|$$

Далее принимается решение о важности конкретных весов согласно критерию Солиенси:

$$\text{mins}_w = \min \left| w \frac{\partial L}{\partial w} \right|$$

Данный способ был выбран по нескольким причинам: 1. он имеет теоретическое обоснование; 2. он достаточно эффективен в реализации, так как коэффициенты ряда можно получить через накопление градиентов (далее, по величине накопленных градиентов определять важность голов).

Существует некоторая сложность, заключающаяся в том, что на каждом слое головы представлены общими линейными слоями. Так сделано для целей оптимизации вычислений, так что нужно удалять головы, вырезая каналы из этих общих слоев.

На втором этапе для только что урезанной модели выполняется стягивание к выходам модели LaBSE-en-ru с использованием комбинированной функции потерь, состоящей из функции потерь, оценивающей похожесть русских и английских предложений, и функции потерь по выходам сети-учителя и сети-ученика.

Последний этап - замер качества - производится на бейнчмарке русскоязычных энкодеров encodechka, на основе которого будет производиться сравнение с LaBSE, LaBSE-en-ru, rubert-tiny2, так как целью является получение модели не сильно хуже по качеству, чем LaBSE и LaBSE-en-ru, и помимо этого, полученная модель не должна проиграть по качеству модели rubert-tiny2.

6 Эксперимент

6.1 Критерии

Качество модели было решено оценивать на бенчмарке encodechka. Бенчмарк оценивает качество эмбеддингов по следующим типам задач:

- `Semantic text similarity (STS)` на основе переведённого датасета STS-B;
- `Paraphrase identification (PI)` на основе датасета paraphraser.ru;
- `Natural language inference (NLI)` на датасете XNLI;
- `Sentiment analysis (SA)` на данных SentiRuEval2016;
- `Toxicity identification (TI)` на датасете токсичных комментариев из OKMLCup;
- `Inappropriateness identification (II)` на датасете Сколтеха;
- `Intent classification (IC)` и её кросс-язычная версия ICX на датасете NLU-evaluation-data, который был переведён на русский язык машинным переводом: в IC классификатор обучается на русских данных, а в ICX – на английских, в обоих случаях тестирование проводится на русском языке;
- `Named Entity Recognition (NER)` - распознавание именованных сущностей на датасетах factRuEval-2016 (NE1) и RuDReC (NE2): эти две задачи требуют получать эмбеддинги отдельных токенов, а не целых предложений, поэтому по этому критерию возможно оценить не все модели.

6.2 Инструкции

[инструкции для воспроизведения эксперимента]

Инструкции по подготовке окружения, получении данных, подключении системы отслеживания экспериментов находятся в репозитории в файле README.md

Так же в нем указаны скрипты для запуска эксперимента и просмотра результатов. Так для запуска эксперимента в подготовленной среде достаточно ввести команду make train

ВАЖНО. Для работы необходимо не менее 24 ГБТ видеопамяти

6.3 Базовые подходы

[описание некоторого минимально сложного подхода, который давал бы минимально приемлемый по требованиям результат]

7 Результаты

[что получилось в итоге, графики и т.д.]

Здесь (ссылка) можете ознакомиться с полным циклом эксперимента и полученными результатами В результате мы получили сеть, которая в 1.5 раза меньше по количеству параметров работает на 40% быстрее имеет в два раза меньшую вычислительную сложность по среднему проценту качества на всех задачах отличается от оригинальной сети всего на 0.5% на задаче NLI показывает качество выше оригинальной сети

8 Интерпретация

[интерпретация результатов, т.е. как трактовать графики и др.]

9 Заключение

[краткое summary проделанной работы] Данная работа показывает, что можно эффективно уменьшать и ускорять сети. Текущий результат был получен в рамках ограниченности ресурсов как временных, так и аппаратных. Однако он вполне сопоставим с результатами известных работ, описанных в п.3. Более того, ряд техник не были применены из-за ограниченности по времени. Авторы данной работы будут продолжать этот эксперимент и нацелены уменьшить исходную модель в 10 раз, что более чем согласуется с теорией лотерейных билетов (ссылка)

Ссылки

[ссылки на использованную литературу]