Лекция 8 Ускорение и сжатие модели

1) Какие задачи решаются в NLP, как они решаются?

2) Как устроен пайплайн кл-и?

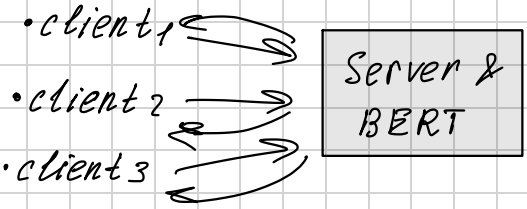
Чтобы сделать модель эффективнее, что можно делать?

- model size
- количество сэмплов в секунду (throughput)
- сколько времени обрабатывается пример (latency)

Когда полезен throughput / samples?

(1) Как можно применять модель?

1) Inference server



Плюсы:

- удобно деплоить
- модель не украдут
- клиенты не обременены вычисл.

Какие факторы важны?

throughput, latency, size

↑ задержки сети делают бесполезными улучшения

↑ отдельный сервер, можете позвонить

Минусы:

- вы платите за инференс
- клиенты не могут работать с моделью
- сеть

Как с этим работать?

1) Лучше обрабатывать батч запросов, как это устроить?

Ждать достаточный размер / ввести задержку.

2) Пусть в батче есть короткие посл-ти и одна длинная, тогда вы будете терять время на паддингах.

Будем группировать посл-ти по длине статически.

Лучше изменять границы групп динамически.

Что делать когда много серверов?

1) можно рас-ть равномерно

2) можно учитывать размер очереди

2) Local Inference

очень важна скорость ответа.

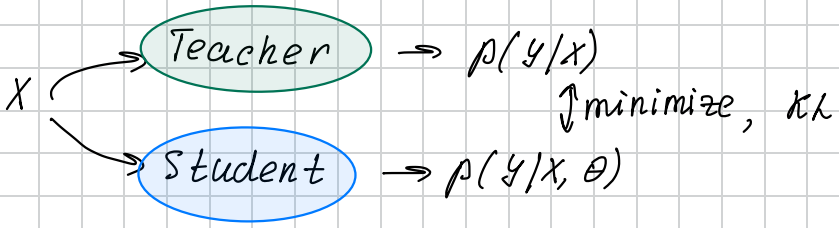
Где используется: 1) беспилотники
2) отчасти колонки, умные камеры

Факторы: latency, size, throughput

3) Smartphone app

Факторы: size, latency, throughput

② Distillation



Цель: обучить маленькую модель приближать выходы большой

Плюсы:

1) можно обучать студента без разметки.

2) Ученик решает более простую задачу, поэтому обучается лучше (махи руками)

3) Можно применять учителя только на некоторых примерах

DistilBERT

Применяют soft-distillation, а также минимизируют кос. рас-е между эмбеддингами

Как можно дис-ть на даунстрим задачу?

distil → finetune +

finetune → distil −

Table 1: **DistilBERT retains 97% of BERT performance.** Comparison on the dev sets of the GLUE benchmark. ELMo results as reported by the authors. BERT and DistilBERT results are the medians of 5 runs with different seeds.

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Table 2: **DistilBERT yields to comparable performance on downstream tasks.** Comparison on downstream tasks: IMDb (test accuracy) and SQuAD 1.1 (EM/F1 on dev set). D: with a second step of distillation during fine-tuning.

Model	IMDb (acc.)	SQuAD (EM/F1)
BERT-base	93.46	81.2/88.5
DistilBERT	92.82	77.7/85.8
DistilBERT (D)	-	79.1/86.9

Table 3: **DistilBERT is significantly smaller while being constantly faster.** Inference time of a full pass of GLUE task STS-B (sentiment analysis) on CPU with a batch size of 1.

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

Уменьшение параметров можно делать по-разному:

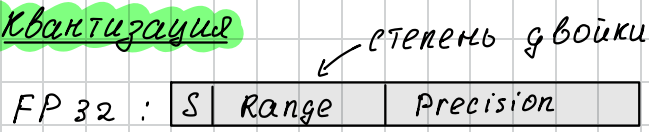
1) Низкоранговое приближение

2) Sparce матрицы

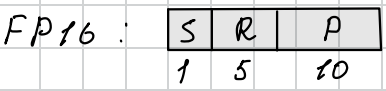
3) Обрубание слоев

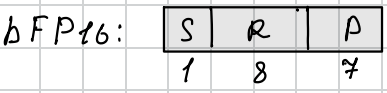
4) Обучение ансамбля → дистилляция в одну модель

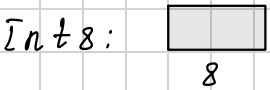
③ Квантизация



Модели для инференса обычно не нужна такая высокая точность.



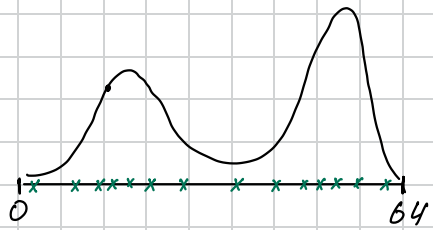




– тяжко работать без дробей, поэтому нужно подбирать диапазон

1) Можно умножать на скейл

2) Как учесть рас-е, чтобы брать большую точность там, где больше чисел:



⇒ будем хранить каждый вес, как ближайший перцентиль

т.е. на перцентиль будем тратить 32 бита, а на его индекс 4-8 бит, обычно это позволяет сильнее сжимать, но делается это дольше.

High-dimensional case

1) Делаем K-means на векторах

2) Заменяем вектор его центроидом и будем хранить номер центроида

Quantization Aware Training

Модель плавно привыкает во время обучения к квантизованным слоям.