Using the Transformer

DistilBERT (Sanh et al., 2019)

Learning goals

- Understand model distillation in general
- Training regime of DistilBERT

MODEL DISTILLATION ▸ HINTON ET AL. (2015)

Model compression scheme:

- Motivation comes from having computationally expensive, cumbersome ensemble models. ▸ Bucila et al. (2006)
- Compressing the knowlegde of the ensemble into a single model has the benefit of easier deployment and better generalization
- Reasoning:
  - Cumbersome model generalizes well, because it is the average of an ensemble.
  - Small model trained to generalize in the same way typically better than small model trained "the normal way".

MODEL DISTILLATION ▸ HINTON ET AL. (2015)

Distillation:

- Knowledge transfer via *soft targets* from original model
  - *Hard targets:* $[0, 0, 0, 1, 0]$
  - *Soft targets:* $[0.01, 0.05, 0.03, 0.87, 0.04]$

  $\rightarrow$ Train the student to *mimick* the teacher's behaviour
- *No need to know the true labels of the training instances*

$\rightarrow$ When true labels are known: Weighted average of two different objective functions possible (on soft and hard targets)

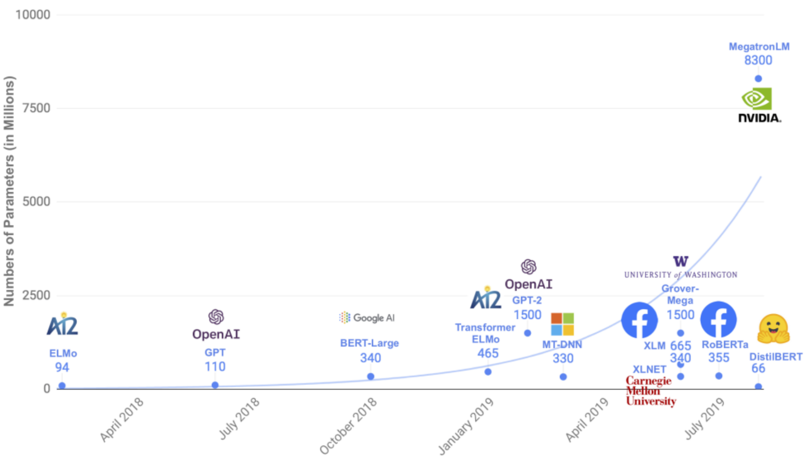
- Additional controlling parameter:

$\rightarrow$ Temperature T in the softmax:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

DISTILBERT ▸ SANH ET AL. (2019)

Motivation:



Source: Sanh et al. (2019)

STUDENT ARCHITECTURE

Characteristics of *DistilBERT*:

- Half the number of layers compared to BERT*
- Half of the size of BERT, but retains 95% of the performance
- Initialize from BERT (taking one out of two hidden layers)
- Same pre-training data as BERT (Wiki + BooksCorpus)

*Rationale for "only" reducing the number of layers:

Larger influence on the computation efficiency compared to e.g. hidden size dimension

TRAINING AND PERFORMANCE

Combination of different loss functions:

- Distillation loss $L_{ce} = \sum_i t_i \cdot \log(s_i)$
- Masked language modeling loss L_{mlm} (cf. BERT)
- Cosine-Embedding-Loss L_{cos}
  (*Rationale:* Keep DistilBERT's embeddings close to the ones from BERT)

Use improvement from RoBERTa:

- Stop using the NSP loss
- Dynamic masking for MLM
- Train with large batches

BENCHMARK PERFORMANCE

Performance differences to BERT:

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

Source: Sanh et al. (2019)

EXAMINING THE IMPORTANCE OF THE LOSSES

Ablation study regarding the loss:

Table 4: **Ablation study.** Variations are relative to the model trained with triple loss and teacher weights initialization.

Ablation	**Variation on GLUE macro-score**
$\emptyset$ - L_{cos} - L_{mlm}	-2.96
L_{ce} - $\emptyset$ - L_{mlm}	-1.46
L_{ce} - L_{cos} - $\emptyset$	-0.31
Triple loss + random weights initialization	-3.69

Source: Sanh et al. (2019)

SIZE AND SPEED

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

Source: Sanh et al. (2019)