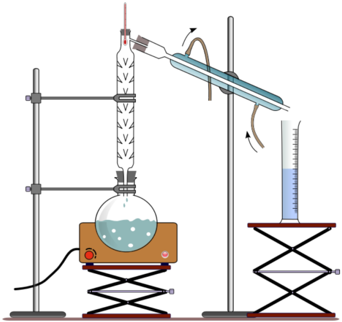
Post-BERT Era

Model distillation



Learning goals

- soft vs. hard targets
- understand how distillation works
- DistilBERT
- other approaches towards compression

MODEL COMPRESSION

Motivation: ▸ Bucila et al., 2006

- Existence of computationally expensive, cumbersome ensemble models
- Accuracy/Performance not everything the should be taken into account
- Time and space requirements also of importance
- *Model Compression* to “to obtain fast, compact yet highly accurate models“
- The fast and compact model should approximate the function learned by the slower and larger model

MODEL COMPRESSION

Advantages:

- Compressing the knowlegde of the ensemble into a single model has the benefits of
  - easier deployment
  - better generalization
  - (potential) interpretability
- *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 COMPRESSION

- *Challenge*:
  - Bucila et al. assumed only knowledge about the weights of the large (ensemble) model
  - No access to its training data
    $\rightarrow$ Their work is mainly on generating suitable pseudo data for model compression
- *NLP:*
  - Abundant amounts of data; no need for creating pseudo data
  - Self-supervised objectives can be used for model compression

MODEL DISTILLATION

Types of targets

- "*Hard*" Targets

$$[0, 0, 0, 1, 0]$$

  - Present when learning from labeled data
  - Ordinary way of training models
- Knowledge transfer possible via "*soft*" targets from original model
- Possible "*soft*" targets
  - Logits + L2 loss ▸ Bucila et al., 2006
  - Softmax scores
  - Temperature T in the softmax ▸ Hinton et al., 2015

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

MODEL DISTILLATION

Temperature:

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

- z_i/z_j are the raw logits
- Parameter T controls to smoothness of the distribution
  - Higher T: “softer“ distribution
  - Lower T: more pronounced distribution
- *Distillation phase*: Apply same temperature to teacher and student
- *Inference*: Set $T = 1$ to recover standard softmax

MODEL DISTILLATION

Intuition

- Train the student to *mimick* the teacher's behaviour
- **Question:** Advantage of the soft targets:

  $\rightarrow$ No need to know the true labels of the training instances

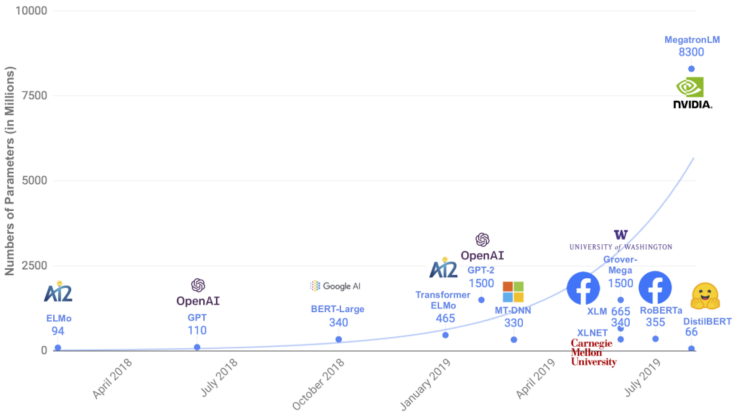
  $\rightarrow$ More fine-grained information about the teacher's behaviour
- When true labels are known:
  Weighted average of two different objective functions possible

  $\rightarrow$ On soft targets (*Be close to the teacher*) and

  $\rightarrow$ on hard targets (*be close to the "ground truth"*)

DISTILBERT

Motivation:



Source: Sanh et al., 2019

DISTILBERT

Characteristics of the student architecture:

- Half the number of layers compared to BERT[1] (6 instead of 12)
- Remove segment embeddings[2] (required for NSP objective)
- Half of the size of BERT, but retains up to 97% of the performance
- Initialize from BERT (taking one out of two hidden layers)
- Same pre-training data as BERT (Wiki + BooksCorpus)

[1] Rationale for "only" reducing the number of layers:
Larger influence on the computation efficiency compared to e.g. hidden size dimension

[2] Sanh et al. call them “token-type embeddings“ in their paper

DISTILBERT

Combination of three different loss functions:

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

Adopt improvement from RoBERTa:

- Stop using the NSP loss (hence removed segment embeddings)
- Dynamic masking for MLM
- Train with large batches

DISTILBERT

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

Takeaways:

- Largest performance on textual entailment task (RTE)
- Even better reported performance on Winograd schemas (WNLI)
- *Overall:* Very consistent performance a little worse than BERT

DISTILBERT

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

Takeaways:

- Start training from BERT's weights is crucial
- Removing MLM loss has little impact

  $\rightarrow$ Signal in the soft targets sufficiently strong
- Notable contribution of the cosine embedding loss

DISTILBERT

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

RELATED APPROACHES (1)

Quantization:

- *In General:* Mathematical procedure of converting a number from a continuous scale to a discrete scale
- *In Machine Learning (usually):* Convert a number from high precision (float16/float32) to a lower precision (float8/int8)
- For models with billions of parameters, this can save enormous amount of memory
- Given that model sizes (at the moment) grow faster than GPU memory, this can be crucial for even making *inference* of large models feasible
- *Trade-Off:* Memory savings vs. model quality

RELATED APPROACHES (2)

Pruning:

- Remove ("*prune*") certain parts of the model according to some score measuring the importance of that part for performance
- Common importance score (in NLP):
  Sensitivity of the loss wrt the values of the neurons ▸ Yang et al., 2022

$$\mathrm{IS}(\boldsymbol{\Theta}) = \mathbb{E}_{x \sim X} \left| \frac{\partial \mathcal{L}(x)}{\partial \boldsymbol{\Theta}} \boldsymbol{\Theta} \right|$$

- Common choices for $\boldsymbol{\Theta}$ (in Transformers):
  - Output of an attention head
  - An intermediate neuron in the FFN layer

RELATED APPROACHES (2)

Self-supervised Pruning ▸ Yang et al., 2022

- Substitute $\mathcal{L}$ by

$$\mathcal{L}_{\mathrm{KL}}(x) = \mathrm{KL}(\mathrm{stopgrad}(q(x))||p(x)),$$

- where
  - $q(x)$ is the original model prediction distribution,
  - $p(x)$ is the to-be-pruned model prediction distribution,
  - and `stopgrad` is used to stop back-propagating gradients