Introduction

Large language models like BERT have achieved state-of-the-art results across a wide range of NLP tasks but suffer from significant computational costs, slow inference speed, and high memory demands. These limitations restrict their deployment on resource-constrained devices. The DistilBERT paper, titled "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter" by Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf (Sanh et al., 2019), directly addresses these challenges.

DistilBERT uses a distillation process where a smaller "student" model is trained to replicate the behavior of a larger "teacher" model (BERT-base). The method combines a soft-label loss (KL-divergence between teacher and student outputs), a masked language modeling (MLM) loss, and a cosine embedding loss to preserve both language understanding and hidden state representations.

Their contribution is a model that is 40% smaller, 60% faster, and retains 97% of BERT's performance, making it highly efficient for practical use cases. Our project re-implements and evaluates the distillation process to create "TinyDistilBERT," using a smaller training corpus and limited computational resources.

Chosen Result

Rather than targeting a single figure, we aimed to replicate the overall distillation process of DistilBERT and validate it through three critical tables from the original paper:

- Table 1: DistilBERT retains 97% of BERT performance on GLUE benchmark tasks.
- Table 2: DistilBERT yields comparable performance on downstream tasks (IMDb and SQuAD 1.1).
- Table 3: DistilBERT is significantly smaller while achieving faster inference times.

We selected these results because together they capture the essence of model distillation which is maintaining high performance while improving size and speed, which aligns directly with our project goals.

Methodology

- **Pretraining:**
  - Teacher model: BERT-base.
  - Student model: TinyDistilBERT.
  - Dataset: WikiText-2 (1/10 of the full original corpus due to GPU limitations).
  - Loss functions: Combined KL-divergence (soft targets), masked language modeling (MLM) loss, and cosine embedding loss.
- **Fine-tuning:**
  - Separate fine-tuning on GLUE tasks (e.g., MNLI, SST-2), IMDb sentiment classification, and SQuAD v1.1 QA tasks.
  - Each task used 4-5 epochs to balance training time and performance, adjusted due to dataset size and compute limitations.
- **Inference Measurement:**
  - Measured end-to-end inference speed on CPU (single core) and GPU for batch size = 1.
- **Key Design Choices and Modifications:**
  - Used 6 separate Python notebooks to modularize pretraining, fine-tuning, and evaluation.
  - Limited training epochs and reduced dataset size to fit GPU constraints.

Results & Analysis

Findings from our results:

- **Table 1 (Performance Retention):** TinyDistilBERT retained a large portion of BERT's performance across GLUE tasks (70%), validating the effectiveness of distillation even under smaller pretraining data. However it struggled a lot with CoLA and STS-B.
- **Table 2 (Downstream Task Performance):** TinyDistilBERT achieved comparable results on IMDb but underperformed a lot on SQuAD, consistent with original observations about distillation challenges for QA tasks.
- **Table 3 (Size and Speed Gains):** We confirmed significant reductions in model size and consistent improvements in inference speed on both CPU and GPU. Inference speed testing on DistilBERT and TinyDistilBERT yielded faster results than BERT.

Challenges Encountered:

- Some GLUE tasks initially returned near-zero scores due to inadequate pretraining; mitigated by tuning learning rates and increasing fine-tuning epochs.
- Training from scratch on WikiText-2 introduced gaps in generalization, particularly on more complex tasks such as CoLA, STS-B, and SQuAD.

Analysis:

- Despite resource constraints, our reimplementation validates the main claims of DistilBERT:
    - Distillation is highly effective in trading off size, speed, and accuracy.
- Discrepancies are expected given the reduced dataset size and limited training time which we saw in a lower benchmark score across the board.

Reflections

Lessons Learned: Pretraining data scale matters significantly for downstream generalization and modular code design (separate notebooks for each task) streamlined experimentation and error analysis.

Key Takeaways: Model distillation offers a promising avenue for deploying language models in resource-constrained environments and smaller pretraining corpora impact syntax-heavy tasks (e.g., QA) more than simple classification tasks.

Future Directions: Experiment with richer or augmented pretraining datasets to increase syntax-heavy tasks scores and explore more aggressive compression strategies like pruning or quantization alongside distillation.

References

1. Sanh, Victor, et al. “DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter.” *ArXiv.org*, 2019, arxiv.org/abs/1910.01108.
2. WikiText-2 Dataset: https://paperswithcode.com/dataset/wikitext-2
3. GLUE Benchmark: https://gluebenchmark.com/
4. IMDb Dataset: https://www.kaggle.com/datasets/hijest/genre-classification-dataset-imdb
5. SQuAD Dataset: https://rajpurkar.github.io/SQuAD-explorer/