Mid-Submission Report: Evaluating NLP Models Across Multiple Benchmarks

Kushagra Dhingra
Abhyudit Singh
Samagra Bharti

1 Introduction

This report summarizes our progress on evaluating various NLP models on a range of benchmarks, including both language understanding and mathematical reasoning tasks. Our study covers baseline experiments as well as novel modifications aimed at reducing computational costs via low-rank adaptation (LoRA). We focus on experiments in three main domains: language understanding (SuperGLUE, TREC, CoNLL), translation, and mathematical problem solving (SVAMP).

Additionally, we reference relevant literature that underpins our approach, including recent studies on transformer models' mathematical capabilities and works focusing on efficient fine-tuning strategies. Code execution details and further experimentation notebooks (available on Kaggle) are referenced for reproducibility and transparency.

2 Baseline Models and Benchmarks

2.1 Baseline Evaluation

Benchmarks Used:

- **SuperGLUE**: Evaluates language understanding, inference, and reasoning with a variety of challenging tasks.
- **TREC**: Focuses on text retrieval and classification tasks.
- **CoNLL**: A standard benchmark for Named Entity Recognition (NER) and sequence labeling tasks.

Approach: We have run a set of baseline models on these benchmarks to obtain initial performance scores. These metrics will serve as a comparison for the improvements achieved using our modifications. Detailed score results are documented in our experimental logs and notebooks.

3 Modifications to the Original Outline

3.1 Expanded Task Coverage

Math Benchmark – SVAMP: To evaluate the models' arithmetic reasoning abilities, we have added the SVAMP benchmark, which challenges the model with diverse mathematical problems.

3.2 New Implementation Approach: "LoRA From Scratch"

Objective: The key idea behind this modification is to incorporate low-rank adaptation (LoRA) directly into our model training. Rather than relying solely on traditional full fine-tuning methods, we implement LoRA from the ground up to enable more parameter-efficient updates.

Detailed Implementation Components:

- **Matrix Decomposition:** Decompose large weight matrices into smaller, low-rank matrices (denoted as A and B) which reduces the number of trainable parameters. Introduce scaling factors (e.g., α) to modulate the strength of the low-rank updates.
- **Layer Integration:** Replace standard layers (such as Linear layers) in Transformer architectures with LoRA-enhanced variants. Ensure these new layers can be seamlessly integrated into attention blocks and feed-forward modules in popular models like RoBERTa, XLNet, and Llama 3.0.
- **Training Logic Adjustments:** Freeze original pre-trained weights, limiting optimization to the newly introduced low-rank components. Fine-tune key hyperparameters such as rank, the scaling factor (α), and dropout to balance model efficiency with performance gains.
- **Evaluation Strategy:** Measure improvements in computational efficiency (memory consumption and training time) compared to full fine-tuning. Benchmark the performance of LoRA-enhanced models against baseline results across tasks like NER, binary, and multi-class classification.

3.3 Planned Experiments with LoRA

Models Under Study:

- RoBERTa for tasks such as NER and binary classification.
- XLNet for similar language tasks, with a particular focus on multi-class classification.
- Llama 3.0 exploring performance in both language understanding and translation tasks.

Experimental Variables:

- Parameter Ranks: Evaluate the impact of varying low-rank dimensions (e.g., 4, 8, 16).
- Efficiency Metrics: Compare against baseline fine-tuning in terms of computational efficiency (memory footprint and training time).

3.4 Incorporation of Advanced Reasoning Techniques

Chain-of-Thought (CoT) for Mathematical Tasks: To enhance the models' ability to handle complex reasoning steps, we will experiment with chain-of-thought prompting techniques.

Expanded Math Dataset Coverage: Alongside the SVAMP benchmark, we are now including the GSM8K dataset.

GSM8K Integration: The GSM8K dataset is known for its 8,000 grade-school math problems that require multi-step reasoning. By applying chain-of-thought prompting on both SVAMP and GSM8K, we aim to systematically evaluate and compare the impact of intermediate reasoning on overall task performance.

Expected Benefits: Incorporating GSM8K allows us to study the generalization of chain-of-thought strategies across different types of math problems, offering insights into how our low-rank adaptation (LoRA from scratch) techniques interact with advanced reasoning methodologies.

4 Architecture Study and Literature Review

4.1 Model Architecture Comparison

Benchmark Performance Analysis: Analyze the differences in performance across benchmarks (SuperGLUE, TREC, CoNLL, SVAMP) to understand how model architecture influences task-specific outcomes.

Architecture Adaptations: Focus on modifications in attention mechanisms and feed-forward network designs in Transformer models, including integration of low-rank adaptations.

4.2 Literature Review

Efficient Fine-Tuning: Reviews on approaches such as parameter-efficient fine-tuning (PEFT) have motivated our "LoRA from scratch" strategy.

Mathematical Reasoning in Transformers:

- Hu, Y. et al. (2023). *Case-Based or Rule-Based: How Do Transformers Do the Math?*
- Li, C. et al. (2023). *Common 7B Language Models Already Possess Strong Math Capabilities*

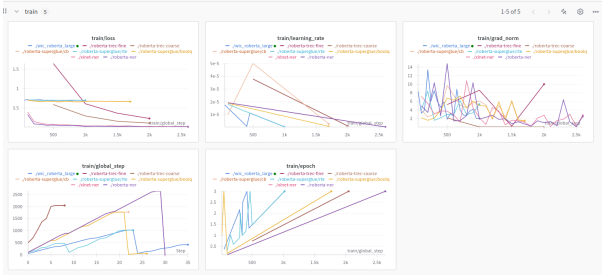
Model-Specific Literature:

- RoBERTa: Studies on robust pre-training and fine-tuning.
- XLNet: Research highlighting its autoregressive and permutation-based strategies.
- Llama 3.0: Literature on scaling laws and adaptation in large-scale language models.

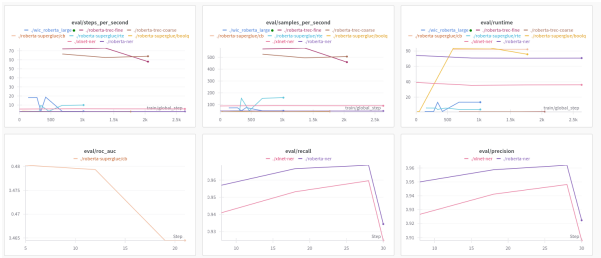
Benchmark-Specific References: We reference foundational works on SuperGLUE, TREC, CoNLL, and SVAMP to validate our dataset choices.

5 Links to Code Base and Models

- RoBERTa - TREC: Code added via zip folder.
- RoBERTa SuperGLUE: `https://colab.research.google.com/drive/1J29A1GrKwPxGFZaXb9p70ta8aBGcl`
  `usp=sharing`
- RoBERTa SuperGLUE (alt): `https://colab.research.google.com/drive/1HRx7dkj437AsABY26CpdkHA7`
  `usp=sharing`
- RoBERTa CoNLL: `https://colab.research.google.com/drive/1JRZs7h6c1Cwo-ld3wrsJDzrW6Mu93UDo?`
  `usp=sharing`
- XLNet - CoNLL: `https://colab.research.google.com/drive/1iMMI1n8vbyoGX02PIn0dC2tB8gohvHqX?`
  `usp=sharing`
- XLNet Kaggle: `https://www.kaggle.com/code/abhyuditsinghiitr/inlp-proj-xlnet`
- Llama Kaggle: `https://www.kaggle.com/code/bitmap04/inlp-proj-llama3`



(a) Training Statistics



(b) Evaluation Statistics

Figure 1: Model Training and Evaluation Metrics

6 Project Timeline for the Month

- **Week 1:**
  - Setup baseline performance using pre-trained models on math benchmarks.
  - Conduct literature review on fine-tuning for mathematical reasoning.
- **Week 2:**
  - Implement fine-tuning strategies for each model.
  - Experiment with XLNet span-prediction, RoBERTa logic, Llama 3 scaling, and GPT-2 CoT.
- **Week 3:**
  - Evaluate performance on selected benchmarks.
  - Perform error analysis focused on CoT.
- **Week 4:**
  - Compile results and write final report.
  - Explore future directions including ensembles and data augmentation.

7 Conclusion

Summary of Progress:

- **Baseline Performance:** Initial runs completed on SuperGLUE, TREC, CoNLL with performance logs collected.
- **Modifications:** LoRA from scratch implemented, translation tasks added, and math evaluation extended with CoT prompting.
- **Project Timeline:** Weekly milestones established for implementation and analysis.

This revised document outlines our current progress and sets the stage for further exploration and refinements over the next month. For reproducibility and detailed code execution steps, refer to the Kaggle and Colab links provided.

8 References

- Liu, Y. et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* arXiv:1907.11692
- Yang, Z. et al. (2019). *XLNet: Generalized Autoregressive Pretraining for Language Understanding.* arXiv:1906.08237
- Touvron, H. et al. (2024). *LLaMA 3: Open Foundation and Instruction Models.* Meta AI
- Hu, E. et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models.* arXiv:2106.09685
- Wei, J. et al. (2022). *Chain of Thought Prompting Elicits Reasoning in Large Language Models.* arXiv:2201.11903
- Hu, Y. et al. (2023). *Case-Based or Rule-Based: How Do Transformers Do the Math?*
- Li, C. et al. (2023). *Common 7B Language Models Already Possess Strong Math Capabilities*
- Wang, X. et al. (2021). *SVAMP: A New Benchmark for Math Word Problems*
- Wang, A. et al. (2019). *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*
- Hendrickx, I. et al. (2010). *Introduction to the CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text*