Multi-Task Learning for Language Model Fine-Tuning

Stanford CS224N Default Project

Anonto Zaman
Department of Aeronautics and Astronautics
Stanford University
anontoz@stanford.edu

1 Key Information to include

- External collaborators (if you have any): N/A
- Mentor (custom project only): N/A
- Sharing project: N/A

2 Research paper summary

Title	Gradient Surgery for Multi-Task Learning
Venue	Conference on Neural Information Processing Systems (NeurIps)
Year	2020
URL	https://arxiv.org/abs/2001.06782

Table 1: Bibliographical information for chosen research paper [1].

Background. Deep learning methods have exhibited excellent capabilities across a variety of complex tasks. When applying a particular method to multiple objectives, however, performance often suffers, owing in-part to complexities with individual training and stringent data requirements. Oftentimes, it may be desirable to optimize a network over multiple tasks simultaneously, but doing so is difficult and may lead to worse performance.

Multiple explanations have been proposed for why multi-objective optimization is challenging, including varied learning speeds for different tasks and plateaus in the optimization landscape. The authors of this work propose that the gradients from different tasks may conflict with each other, thus inhibiting learning progress. The authors further hypothesize that these conflicting gradients are particularly detrimental when they have high positive curvature and a large difference in magnitude. The authors develop a method to address gradient conflicts in the multi-objective learning task.

Summary of contributions. The authors proposed method is called projecting conflicting gradients (PCGrad). When gradients from different tasks conflict, each is projected onto the plane of the other and then passed to the optimizer. By directly altering the gradients, the algorithm prevents interfering components of the gradient from being applied. The authors provide a theoretical proof of the optimization conditions when PCGrad outperforms traditional multi-task optimization. In their empirical analysis, they assessed PCGrad's performance on a variety of multi-task problems, such as scene understanding and goal-conditioned RL. The results showed that their method yielded improvements in optimization speed, data efficiency, and performance. Additionally, their method can be implemented alongside prior state-of-the-art methods.

Limitations and discussion. In general, machine learning models may struggle with safety, bias in training data, and large data requirements for training. Though the PCGrad algorithm can be used to improve performance on multi-task problems, it is still susceptible to the aforementioned risks.

In their experimental results, the authors evaluated their method's performance on supervised learning and reinforcement learning tasks. In the supervised learning case, the authors compared PCGrad's performance to routing networks, cross-stitch, and task specific training. Though PCGrad outperforms cross-stitch and task specific networks, it achieves only marginal improvements over the routing networks. Combining the routing network and PCGrad led to only a 2.8% improvement in test accuracy on the CIFAR-100 multi-task dataset. Similarly, the authors evaluated their method's performance against Sener and Koultun on the CelebA classification dataset. Though their method achieves superior performance, it only does so by a margin of 0.26%. In summary, though their method does improve learning, the performance benefits are dependent on the dataset and underlying network architecture. The results in this section may have also been enhanced by including some natural language results, as they primarily evaluated their method on image classification tasks.

In the reinforcement learning task, the authors evaluated their algorithm on the Meta-World benchmark. They show that their method yielded significant improvements in skill acquisition and data data efficiency. The reinforcement learning results more convincingly demonstrate the utility of the PCGrad algorithm, as it shows clear advantages over preexisting methods. In comparison, the supervised learning improvements appear more limited.

Why this paper? I have previously taken classes in optimization theory and have learned about different methods for multi-objective optimization, including weighted sums of different cost functions, constraining multiple objectives, and goal programming. I have previously trained multiple-tail networks for multi-objective optimization and found that the approach is incredibly data-intensive. When I was reading through the suggested improvements, I was surprised by the simplicity of this method for multi-objective optimization.

Overall, I feel that this method gave me a better understanding of the challenges with multi-task learning. Though PCGrad is not a perfect solution, I feel that its an interesting avenue to explore for my final project.

Wider research context. The authors highlight how PCGrad is widely applicable across a variety of applications, including image classification and multi-task reinforcement learning. Though the authors don't specifically test the algorithm on NLP datasets, it appears to have broad applicability for multi-task language models.

In general, multi-task optimization is incredibly useful across a variety of domains. By training a single model for multiple tasks, engineers do not have to create bespoke solutions for individual applications. Additionally, multi-task models may yield improvements in performance and data efficiency over single-task methods.

3 Project description (1-2 pages)

Goal. The final project involves training a single language model to address sentiment analysis, paraphrase detection, and semantic textual similarity tasks. My goal is to benchmark the performance of a network trained using PCGrad against standard multi-objective gradient descent and individual task-specific training.

I chose this goal because I'm curious to see whether PCGrad yields substantial improvements in performance or if we achieve similar results using simpler frameworks. Time permitting, I would also like to see how the model's performance is impacted by different weighting terms on the task-specific objective functions.

Task. I will integrate PCGrad into the model training architecture, optimizing over the shared parameters in the minBERT model. The PCGrad algorithm will take the individually calculated gradients for each objective and project them onto a similar direction, thus eliminating non-conflicting components. Each task will have their own output layer with individually trained weights and biases for classification. I will train an additional model with the multi-task objective using standard gradient descent.

In addition to the PCGrad and multi-task training, I will train the network on each task individually. These performance of three networks (PCGrad, multi-objective, and individual training) will then be compared.

Data. I will train the model on the provided project datasets:

Stanford Sentiment Treebank (SST) Dataset

- `train` (8,544 examples)
- `dev` (1,101 examples)
- `test` (2,210 examples)

CFIMDB Dataset

- `train` (1,701 examples)
- `dev` (245 examples)
- `test` (488 examples)

Quora Dataset

- `train` (283,010 examples)
- `dev` (40,429 examples)
- `test` (80,859 examples)

SemEval STS Benchmark Dataset

- `train` (6,040 examples)
- `dev` (863 examples)
- `test` (1,725 examples)

I am not planning on completing any additional preprocessing.

Methods. I plan on following the general method described in the PCGrad paper, using the non-conflicting gradient techniques to update the shared weights in the minBERT model, while updating the final task-specific layers of the model with associated gradients. I plan on implementing the gradient update myself in PyTorch, though I may refer to the author's GitHub for specific details on their implementation method.

Baselines. I will use the task-specific trained model as my baseline. I chose this as my baseline because it seems to be the most "intuitive" choice for training (in past classes I have used task-specific training for multi-task models). I will implement this model myself.

Evaluation. For evaluating I will utilize the standard accuracy metrics used in the final project handout. The loss function is computed using the cross-entropy between the predicted distribution and true label.

Ethical Challenges. As with all machine learning tasks, there are ethical considerations with regard to bias in the provided dataset. It is possible that any model trained on the datasets will perpetuate these biases and present them in any output. To address this limitation, we could try and leverage a variety of datasets from different sources, especially those with known diversity in terms of authorship and viewpoint (i.e. datasets written by non-native English speakers). Another ethical concern is that any training paradigm requires significant computational resources, which is not accessible to all NLP developers. To address this concern, we could publish the final models online and open-source, making them accessible for other researchers to test and further develop.

References

[1] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning, 2020.