Gradient Descent in Multi-Task Learning

Stanford CS224N Default Project
Mentor: Arvind Venkat Mahankali

David Saykin
Department of Computer Science
Stanford University
saykind@stanford.edu

Kfir Dolev
Department of Computer Science
Stanford University
dolev@stanford.edu

Abstract

Following Yu et al. (2020) we investigate a method for mitigating gradient interference in multi-task learning by projecting conflicting task gradients onto the normal plane of each other, thus avoiding detrimental gradient conflicts. We are applying projected conflicting gradients method to test if it will improve BERT's performance on sentiment analysis, paraphrase detection (quora dataset), and semantic textual similarity tasks (SemEval STS benchmark dataset). We find that PCgrad multitask learning performs worse than single task learning, but note that there are a significant number of unexplored variations of our model which likely perform better.

1 Introduction

We attempt to improve upon a pre-trained BERT Devlin et al. (2019) model so that it performs well when simultaneously used on three separate linguistic tasks: Sentiment Analysis, Paraphrase detection, and Semantic textual analysis. We attempt to do this by implementing the multi-tasking learning method described Yu et al. (2020). The method used there is to consider a modification of gradient descent for a loss function consisting of a sum of loss functions for multiple individual tasks. The modification involves computing the individual task gradients, and projecting them onto each others normal plans so that they do not interfere with each other. We hypothesise that this method will improve the performance of BERT fine tuned individually to each task.

The baseline we compare our work to is a pre-trained BERT that is individually fine tuned to accomplish each of these tasks separately. The BERT model takes in a sequence of tokens, and returns a contextual representation of each. The first token is always set to "CLS", and its representation "pools" together the information in the entire input sentence into one vector. This output vector is then used as input for a final task dependent classifier layer.

The algorithm we use to attempt to improve BERT, PCGrad Yu et al. (2020) utilizes a straightforward method to resolve gradient conflicts during optimization. When two tasks' gradients are opposed, indicated by a negative cosine similarity, PCGrad projects each task's gradient onto the orthogonal plane of the other's gradient. This process effectively eliminates the component of the gradient causing the conflict, thereby minimizing harmful interference between the tasks' gradients. Fig. 2. in the document illustrates this concept visually.

2 Related Work

The paper by Yu et al., "Gradient Surgery for Multi-Task Learning" Yu et al. (2020) addresses the challenge of gradient interference in multi-task learning environments. The motivation stems from the observation that when training models on multiple tasks simultaneously, conflicting gradients can hamper learning, leading to suboptimal performance on one or more tasks. The authors propose a novel solution, PCGrad (Projected Conflicting Gradients), which minimizes negative interactions

between task gradients by projecting conflicting gradients onto each other's normal planes. This technique allows for more harmonious learning across tasks, potentially unlocking greater efficiencies and performance enhancements in both supervised and reinforcement learning settings. Cartoon explanation of PCGrad method is provided on Figure 1 (copied from Yu et al. (2020)).

Previous solutions (e.g. Devin et al. (2016); Fernando et al. (2017); Rosenbaum et al. (2019)) include architectural adjustments and problem decomposition to simplify learning, but these often require complex integration techniques like network distillation. For example, there are prior works which combat optimization challenges by only rescaling task gradients Chen et al. (2018); Sener and Koltun (2018). In contrast, this paper introduces a straightforward, effective algorithm, PCGrad, which resolves gradient conflicts by altering both the magnitude and direction of task gradients, based on their cosine similarity, to improve multi-task learning performance. While idea of using cosine similarity of gradients is not novel Du et al. (2018), paper Yu et al. (2020) was first to employ it for identification of conflicts between task gradients.

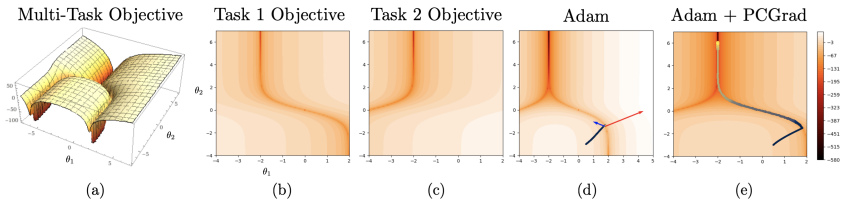

Figure 1: Visualization of PCGrad on a 2D multi-task optimization problem Yu et al. (2020). (a) A multi-task objective landscape. (b) & (c) Contour plots of the individual task objectives that comprise (a). (d) Trajectory of gradient updates on the multi-task objective using the Adam optimizer. The gradient vectors of the two tasks at the end of the trajectory are indicated by blue and red arrows, where the relative lengths are on a log scale. (e) Trajectory of gradient updates on the multi-task objective using Adam with PCGrad.

PCGrad utilizes a straightforward method to resolve gradient conflicts during optimization. When two tasks' gradients are opposed, indicated by a negative cosine similarity, PCGrad projects each task's gradient onto the orthogonal plane of the other's gradient. This process effectively eliminates the component of the gradient causing the conflict, thereby minimizing harmful interference between the tasks' gradients. Fig. 2. in the document illustrates this concept visually.

Algorithm 1 PCGrad Update Rule

Require: parameters θ, task minibatch $\mathcal{B} = \{\mathcal{T}_k\}$

$\mathbf{g}_k \leftarrow \nabla_\theta \mathcal{L}_k(\theta) \;\; \forall k$

$\mathbf{g}_k^{\text{PC}} \leftarrow \mathbf{g}_k \;\; \forall k$

for $\mathcal{T}_i \in \mathcal{B}$ **do**

  for $\mathcal{T}_j \overset{\text{uniformly}}{\sim} \mathcal{B} \setminus \mathcal{T}_i$ in random order **do**

    if $\mathbf{g}_i^{\text{PC}} \cdot \mathbf{g}_j < 0$ **then**

      #Subtract the projection of $\mathbf{g}_i^{\text{PC}}$ *onto* $\mathbf{g}_j$

      Set $\mathbf{g}_i^{\text{PC}} = \mathbf{g}_i^{\text{PC}} - \frac{\mathbf{g}_i^{\text{PC}} \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2} \mathbf{g}_j$

return update $\Delta\theta = \mathbf{g}^{\text{PC}} = \sum_i \mathbf{g}_i^{\text{PC}}$

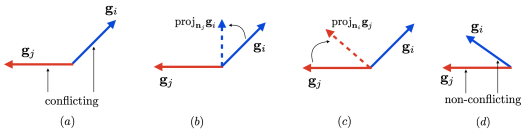

Figure 2: Conflicting gradients and PCGrad. In (a), tasks i and j have conflicting gradient directions, which can lead to destructive interference. In (b) and (c), we illustrate the PCGrad algorithm in the case where gradients are conflicting. Non-conflicting task gradients (d) are not altered under PCGrad.

The paper contributes to the field by providing a model-agnostic approach that can be easily integrated into existing multi-task frameworks, demonstrating its effectiveness through extensive experiments across various domains. While the paper presents a compelling method for addressing gradient interference, a deeper analysis of the types of tasks and models where PCGrad is most effective could strengthen its findings.

Despite these potential areas for further exploration, the paper's findings are convincing, offering a practical solution to a pervasive problem in multi-task learning. By improving the interaction between task gradients, PCGrad represents a significant step forward in the quest for more efficient and effective multi-task learning models, contributing to the broader goal of developing versatile and powerful artificial intelligence systems.

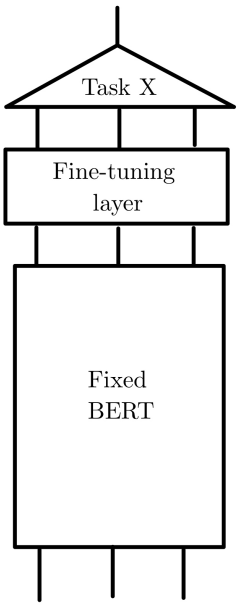

Figure 3: Proposed architecture. Only the parameters in the fine-tuning layer and the task specific final layer will be updated

This method contrasts with sequential learning strategies in continual learning by focusing on positive transfer across multiple tasks simultaneously, without the need for complex programming solutions. The proposed approach demonstrates superior performance compared to previous methods, highlighting its potential for more efficient multi-task learning.

Authors of "Gradient Surgery for Multi-Task Learning" Yu et al. (2020) pinpoint three primary hurdles in multi-task optimization: conflicting gradients, high positive curvature, and significant gradient differences. They introduce PCGrad, a "gradient surgery" technique, to address these issues, enhancing optimization significantly. The simplicity and model-independent nature of PCGrad suggest its potential applicability in broader contexts like meta-learning and natural language processing, offering a promising direction for future research into overcoming optimization challenges in deep learning.

3 Approach

We show our architecture in 3. The output of BERT consists of a single vector encoding the CLS token appended to the beginning of the sentence. The fine-tuning layer consists of a single linear layer followed by a non-linearity. The final layer is a linear layer returning logits for the task at hand, or a single real number for tasks requiring continuous outputs.

In single task training, this entire architecture is separately fine tuned for each task X before being evaluated.

The in multi-task PCgrad training, all tasks are trained at once. A batch is taken from each of the three datasets, and for each the loss function is computed. Gradients are then computed with respect to all parameters, and gradients of loss functions in one task are simply zero with respect to parameters of a final layer for a different task.

4 Experiments

4.1 Data

We describe the datasets in table 1.

Task	Description	Dataset Name	Dataset Description
Sentiment Analysis	Sentiment of a sentence is to be assigned an integer from 0 (most negative) to 4 (most positive).	Stanford Sentiment Treebank	Single sentences from movie reviews annotated by human judges. Train/dev/test split of 8,544/1,101/2,210 examples.
Paraphrase detection	Two sentences are taken as input and a binary variable is output indicating specifying if they are paraphrases of one another.	Quora dataset	Question pairs with binary labels specifying if they are paraphrases of one another. Train/dev/test split of 141,506 /20,215/40,431 examples.
Semantic textual analysis	Two sentences are taken as input and given a continuous similarity score from 0 (least similar) to 5 (most similar).	SemEval STS Benchmark Dataset	Sentance pairs labeled by degree of similarity. Train/dev/test split of 6,041 /864/1,726 examples.

Table 1: Table describing tasks and datasets used to train our model to accomplish these tasks.

4.2 Evaluation method

We evaluate our model on the three tasks described in table1. To evaluate Sentiment Analysis and Paraphrase detection, we use a cross entropy loss function since the task is discrete classification. For the Semantic textual analysis we use the Pearson correlation coefficient. We use this rather than least-square means because it is only necessary to preserve the ground-truth outputs up to a positive linear transformation to maintain the information about semantic similarly.

For Sentiment Analysis and Semantic textual analysis, we also compute accuracy scores, i.e. the percentage of the dev set data for which the model correctly predicts the data.

4.3 Experimental details

We use a learning rate of 10^{-5}. Models are trained for 10 epochs and a batch size of 8.

4.4 Results

Task	Metric	Single-Train	PCgrad
Sentiment Analysis	Percent accuracy	0.520	0.436
Paraphrase detection	Percent accuracy	0.696	0.373
Semantic textual analysis	Pearson Coefficient	0.286	0.279

Table 2: Performance of PCgrad on the various tasks compared to training specifically on one task.

Our PCgrad multi-task implementation performs significantly worse than expected, give that it is meant to be an improvement over single task training. It performs worse on every task. We note that our implementation performs similarly poorly with only multi-task learning even without PC grad. Therefor, it is likely we did not overcome certain basic obstacles when transitioning from single to multi-task learning.

5 Analysis

Consider figure 4 which shows the performance of the model for the various tasks after each epoch of training. After about two epochs the model performs significantly worse, before going back to around the original performance. After this more or less does not improve at all as training continues. We were not able precisely to pinpoint the root of this behavior. It is possible that longer training times or more layers after the BERT model would overcome this obstacle.

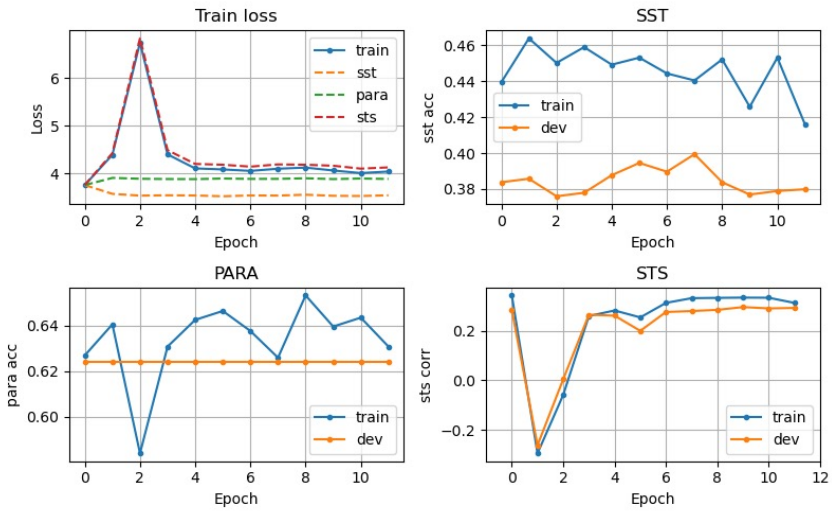

Figure 4: Learning history of PCgrad multi-task model.

6 Conclusion

In this work we have contrasted the performance of PCgrad multitask learning on a BERT model with task specific heads to a similar architecture with single task training on the same tasks. We found that the multitask training performed significantly worse than the single task training. However, our work is severely limited as it left unexplored a number of avenues for potentially overcoming this decrease in performance, as we have mentioned. In future work we hope to explore these avenues and show that PCgrad with multitask learning can perform better.

References

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*.

Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. 2016. Learning modular neural network policies for multi-task and multi-robot transfer. *CoRR*, abs/1609.07088.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Razvan Pascanu, and Balaji Lakshminarayanan. 2018. Adapting auxiliary losses using gradient similarity. *CoRR*, abs/1812.02224.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. 2017. Pathnet: Evolution channels gradient descent in super neural networks. *CoRR*, abs/1701.08734.

Clemens Rosenbaum, Ignacio Cases, Matthew Riemer, and Tim Klinger. 2019. Routing networks and the challenges of modular and compositional computation. *arXiv:1904.12774*.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836.