BERT: A Master of All Trades or Jack of None?

Marie Huynh [1] Josselin Somerville Roberts [2] Tom Pritsky [3]

[1]Department of Biomedical Data Science [2]Department of Computer Science [3]Department of Biomedical Data Science

Introduction

In most machine learning tasks today, we focus on a specific task and train and optimize our model to perform best on this one task. However, trying to learn different tasks at the same time may increase the accuracy by sharing knowledge or simply saving space and computational time on smaller devices.

In this paper, we will review state-of-the-art methods to train a multitask model with a BERT backbone, PALs or 'projected attention layers'[2], a novel method for scheduling training (also from the PAL paper), gradient surgery and gradient vaccine.

We will also explore different classifier heads: Fully connected, Recurrent Neural Network (RNN), Long Short Term Memory (LSTM). In addition we will study the influence of the sizes of these networks as well as the features fed to BERT.

Finally, we introduce Gradient Compromise, a combination of PAL scheduling [2] and Gradient vaccine [3], that massively increases the training speed during the first epochs of finetuning.

Data and Evaluation

We are using the default project datsets, which have the following splits (Table 1.). During pre-processing we **filter out the top 2% longest** training inputs to enable use of large batch sizes and **Easy Data Augmentation** on the Sentiment and Paraphrase Datasets, which helps address class imbalance (seen in Figure 1).

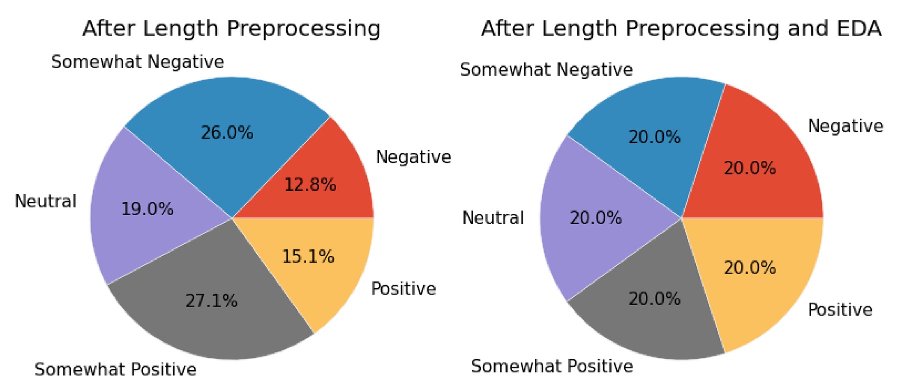

Figure 1. Distribution of classes in SST-5

Datasets	Labels	Size:	Task	Metric	Loss
Quora Dataset	Question pairs with paraphrase labels	Train: 141,506 Dev: 20,215 Test: 40,431	Paraphrase detection *Binary Classification*	Accuracy	Binary Cross Entropy
SemEval STS Benchmark	Sentence pairs labeled 0 (unrelated) to 5 (equivalent)	Train: 6,041 Dev: 864 Test: 1,726	Sentence similarity *Regression*	Pearson Correlation	Mean Squared Error
Stanford Sentiment Treebank (SST-5)	Movie reviews with 5 categorical labels from neg to pos	Train: 8,544 Dev: 1,101 Test: 2,210}	Sentiment analysis *Multiclass Classification*	Accuracy	Cross Entropy

Table 1. Datasets (Default project)

Classifier Heads

Fully connected

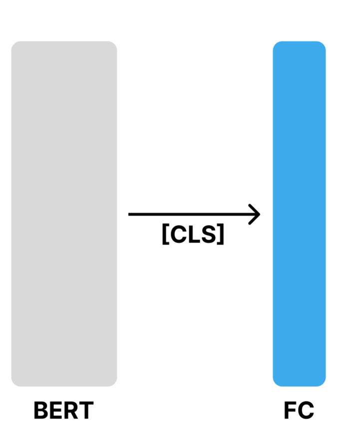

Figure 2. Fully connected classifier head

RNN/LSTM

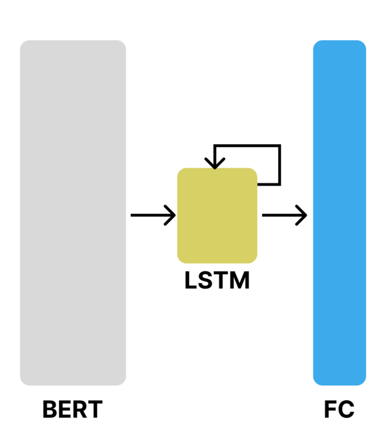

Figure 3. RNN classifier head

LSTM + [CLS] embed.

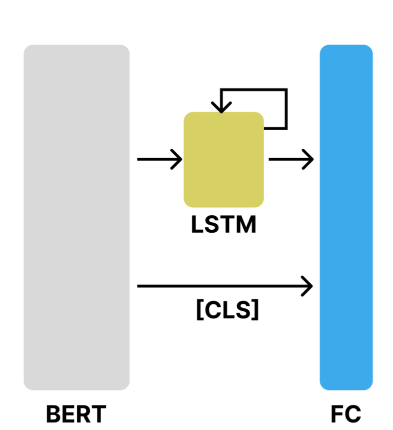

Figure 4. CLS classifier head

Methods to Train a Multitask Model

Baseline

Our baseline consists of a shared BERT model with task specific linear and dropout layers.

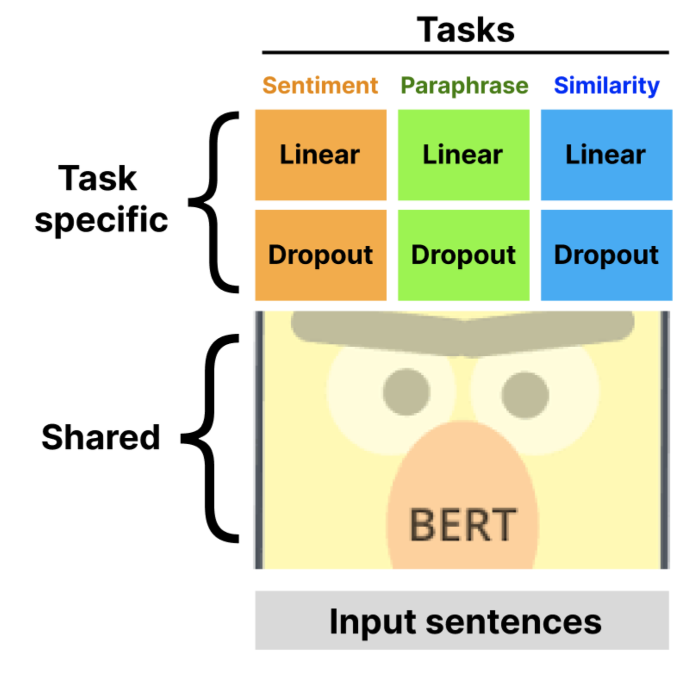

Figure 5. Scheme of our Baseline Model.

Pal Scheduling

How to handle imbalanced datasets? The Pal scheduling [2] strategically samples tasks to prevent overfitting. The probability of choosing task i at epoch e amongst E epochs is:

$$\mathbb{P}(task_i) = \mathbb{N}_i^{\alpha}$$

$$\alpha = 1 - 0.8 * \frac{e-1}{E-1}$$

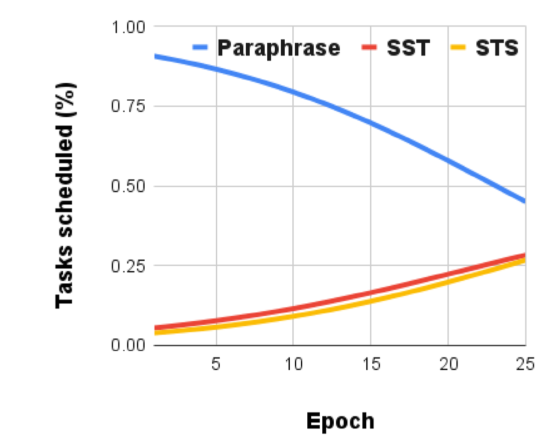

Figure 6. Task Ratios with Scheduler

Gradient Surgery

Gradient Surgery [3] projects competing gradients from one task onto the normal plane of another, preventing the competing gradient components from being applied to the network and impairing optimization.

(Gradient Vaccine follows a similar approach with a more complex projection)

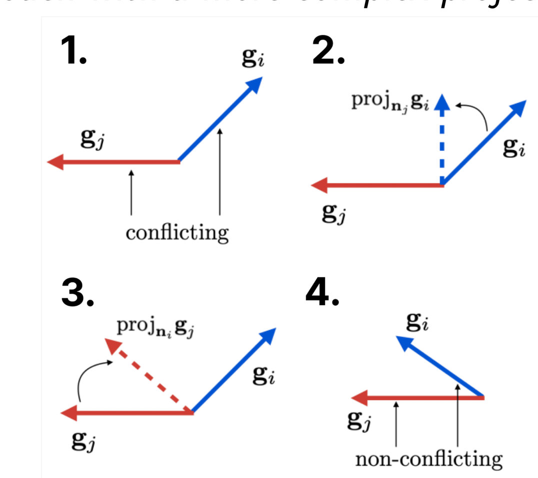

Figure 7. Conflicting Gradients and PCGrad

PALs

Some low-rank task-specific attention layers are added at each BERT Layer:

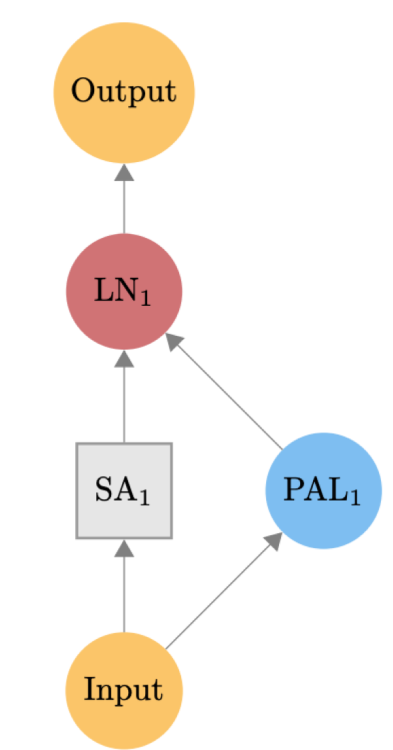

Figure 8. Pal Schematic

Gradient Compromise

Our method that compromises pal scheduling with gradient vaccine:

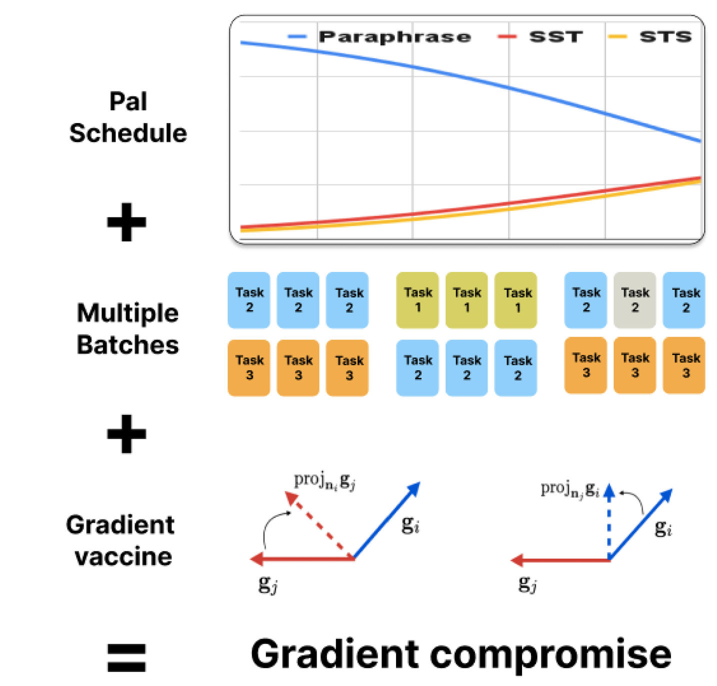

Figure 9. Gradient Compromise Components

Results

Method	Dev. SST	Dev. Quora	Dev. STS	Mean
BERT + concat embed.	0.378	0.693	0.174	0.415
BERT + concat embed. + Pal schedule	0.492	0.764	0.337	0.531
BaseModel: BERT + concat sentences	0.465	0.731	0.749	0.648
BaseModel + Pal schedule	0.499	0.869	0.856	0.741
BaseModel + Pal schedule + 1 hidden	0.504	0.876	0.862	0.747
BaseModel + Pal schedule + 1 hidden + data augment.	0.513	0.879	0.868	0.753
***AdvancedModel*: BaseModel + Pal schedule + 1 hidden + indiv. pretrain + data augment.**	0.520	0.882	0.872	0.758
BaseModel + 1 hidden + PCGrad	0.484	0.871	0.832	0.729
BaseModel + 1 hidden + Vaccine	0.514	0.853	0.845	0.737
BaseModel + 1 hidden + Vaccine + SMART	0.509	0.864	0.846	0.740
BaseModel + 1 hidden + Grad. compromise (ours)	0.516	0.834	0.848	0.733
BaseModel + 1 hidden + RNN 128	0.428	0.841	0.812	0.694
BaseModel + 1 hidden + LSTM 256 + [CLS] embed.	0.500	0.842	0.872	0.738
BaseModel + 1 hidden + LSTM 256 (STS only) + [CLS]	0.517	0.860	0.876	0.751
AdvancedModel + PAL	0.247	0.64	0.523	0.470
AdvancedModel + PAL (only during finetuning)	**0.524**	**0.882**	**0.876**	**0.761**

Discussion

- **Most important methods:** Pal Scheduling, Individual Pretaining, Sentence Concatenation.
- Gradient treatment methods (Surgery, Vaccine, Vaccine + SMART) do not work well with very imbalanced datasets.
- More complex classifier heads do not perform better (RNN, LSTM, LSTM + [CLS] embed.)
- PALs are very hard to train. Only work well when added after finetuning and trained individually.
- **Limitations:** variance of the accuracy depends on the seed, poor hyperparameters.

Gradient Compromise

Though our method failed to improve accuracy (hitting only 0.733 on the dev set), it significantly sped up training (as seen in figure 10). While gradient compromise impaired accuracy relative to the PAL scheduler, we think high accuracy can be preserved by first finetuning with gradient compromise and then completing training with a PAL scheduler.

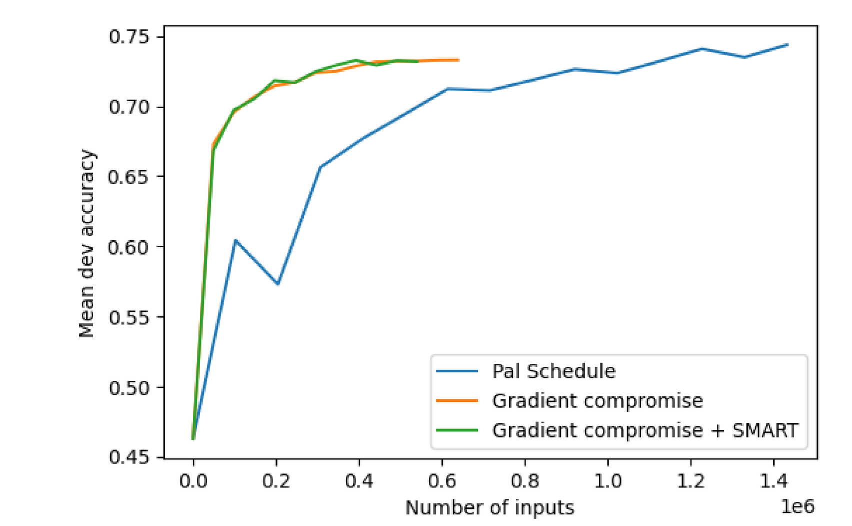

Figure 10. Accuracy on the Dev Set

Future Work

- Training larger PAL layers from scratch.
- Pre-training the LSTM and correctly initializing weights to leverage all sentence embeddings.
- Finetuning the hyperparameters.
- Implement specific approaches that are known to perform well on our datasets, such as Heisen routing for SST [1].
- Leverage other public datasets to potentially increase accuracy and our model generalization.

References

[1] Franz A. Heinsen.
An algorithm for routing vectors in sequences, 2022.

[2] Asa Cooper Stickland and Iain Murray.
Bert and pals: Projected attention layers for efficient adaptation in multi-task learning, 2019.

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

mahuynh@stanford.edu josselin@stanford.edu tom5@stanford.edu