# minBERT Optimization with the SMART Learning Framework

Stanford CS224N Default Project

**Zixuan Xu**
Department of Civil and Environmental Engineering
Stanford University
zixuanxu@stanford.edu

**Yihan Shi**
Department of Civil and Environmental Engineering
Stanford University
yihanshi@stanford.edu

**Zeyu Sun**
Department of Civil and Environmental Engineering
Stanford University
sunzeyu@stanford.edu

## Abstract

In this project, we aim to implement the minBert model and improve the generalization ability of the model and increase prediction accuracy by integrating two important optimization methods: 1) the Smoothness- inducing Adversarial Regularization technique, which effectively manages the complexity of the model; 2) Bregman proximal point optimization, which is an instance of trust-region methods and can prevent aggressive updating. Our experiments show that both the Smoothness-Inducing Adversarial Regularization and Bregman Proximal Point Optimization improve the model performance with proper hyperparameters. The trade-off between performance improvements and training efficiency require further exploration.

## 1 Key Information to include

- Mentor: Sauren Khosla
- External Collaborators: N/A
- Sharing project: N/A

## 2 Introduction

In many NLP applications, the efficacy of machine learning techniques is contingent upon access to large quantities of labeled data. However, obtaining such data can be prohibitively expensive or time-consuming (Pan & Yang, 2009) [1]. To mitigate this issue, researchers have turned to transfer learning.

Transfer learning is a technique employed in the field of natural language processing (NLP) to address the challenge of limited labeled data availability. This technique involves leveraging knowledge from high-resource domains, characterized by an abundance of data, and applying it to low-resource target domains where labeled data is scarce. This process typically involves two stages: pre-training and

fine-tuning. In pre-training, a high-capacity model is trained on out-of-domain data from relevant tasks. In fine-tuning, this model is adapted to the low-resource task in the target domain (Pan & Yang, 2009)[1].

During pre-training, transformer models, such as ELMo (Peters et al., 2018)[2], GPT (Radford et al., 2019)[3], and BERT (Devlin et al., 2019)[4], can capture general semantic and syntactic information for use in downstream NLP tasks. These models are trained in an unsupervised manner using large amounts of unlabeled data. During fine-tuning, the pre-trained model is adapted to the target task/domain by replacing the top layer with a task/domain-specific sub-network and training on limited target data. This approach has achieved state-of-the-art performance in many NLP benchmarks (Liu et al., 2019)[5].

Due to the data limitation from the target task/domain and the extremely high complexity of the pre-trained model, aggressive fine-tuning frequently results in the adapted model overfitting the training data of the target task/domain and does not generalize well to unseen data. To solve this problem, many methods are implemented relying on hyper-parameter tuning heuristics, and require significant tuning efforts.

To mitigate the reliance on hyper-parameter tuning heuristics, Jiang et al.(2019)[6] proposed a fine-tuning framework through regularization optimization techniques. This framework includes two parts:

- Use the Smoothness Inducing Adversarial Regularization technique to improve the smoothness of the model prediction. By adding small and random perturbations to the inputs, Jiang et al.(2019)[6] adjust the hyperparameter to ensure the model outputs a probability distribution as consistent as possible within a certain perturbation range. This ingredient can effectively control the extremely high complexity of the model.
- Apply the Bregman Proximal Point Optimization methods to the fine-tuning of model parameters in order to prevent aggressive updating. By adjusting another hyperparameter and imposing a strong penalty at each iteration, the learning rate of the model parameter maintains steady.

In this project, we aim to implement minBert based on the base Bert model. To mitigate the overfitting problem, we apply the Smoothness Inducing Adversarial Regularization technique. Additionally, we utilize Bregman Proximal Point Optimization to prevent aggressive updating. Our goal is to confirm the improvement described in Jiang et al. (2019)[6] by combining these two techniques. We will analyze the effectiveness of each component of our proposed method and examine how performance varies with the removal of either component.

## 3 Related Work

The baseline minBERT model is from base Bert model by Devlin et al. (2018)[4], a multi-layer bidirectional Transformer encoder based on the original implementation in Vaswani et al. (2017)[7]. Devlin et al. (2018)[4] pre-trained the model on large amounts of text data from diverse sources, including Wikipedia and the BookCorpus. And the model learns to generate contextually sensitive representations of words in text by predicting masked words and next sentence prediction. The pre-trained model parameters will be first initialized and then fine-tuned using labeled data from the downstream tasks. These tasks have separate fine-tuned models. The Bert model has the ability to capture deep contextual relationships between words, which allows it to effectively handle a wide range of NLP tasks. It's reported that the Bert model outperforms existing state-of-the-art models on a variety of NLP tasks, including sentiment analysis, question answering, and natural language inference. The success of BERT has led to a plethora of subsequent research endeavors aimed at improving pre-training performance. These efforts included the introduction of novel unsupervised learning tasks by Yang et al.(2019)[8], and multi-tasking Liu et al. (2019)[9].

The pre-trained Bert model is then adapted in downstream tasks and fine-tuned. To prevent overfitting caused by limitation of data, Jiang et al. (2019)[6] proposed a new learning framework for robust and efficient fine-tuning for pre-trained models through regularized optimization techniques. This SMART framework consists of Smoothness-Inducing Adversarial Regularization and Bregman Proximal Point Optimization. The regularization method can effectively control the model complexity

and the optimization methods can impose a strong penalty at each iteration to prevent the model from aggressive update. After implementing SMART on BERT, the proposed model outperform the BERT baseline model demonstrating the effectiveness of two ingredients.

Our regularization implementation is also inspired by Miyato et al., (2018)[10]. Miyato et al. proposes a new regularization method for neural networks called Virtual Adversarial Training (VAT). This method improves the robustness of neural networks by adding small perturbations to the input data and encouraging the neural network to output similar predictions for the original and perturbed data. They introduce two versions of VAT, one for supervised learning and another for semi-supervised learning. And they demonstrate that VAT can effectively leverage unlabeled data in the semi-supervised setting to improve the accuracy of the neural network.

Our Bregman Optimization techniques is inspired by previous methods, including Bregman proximal methods, Accelerated Bregman proximal gradient (ABPG) by (Gutman-P 2018)[11], momentum Bregman proximal point (MBPP) by (Tarvainen and Valpola, 2017)[12]. MBPP method accelerates the Bregman proximal point method by introducing an additional momentum to the update. This method is called "Mean Teacher" method in Tarvainen and Valpola, (2017)[12].

# 4 Approach

We took the minBERT model as the baseline model, which is a multi-layer bidirectional Transformer encoder based on the original implementation. To improve the model generalization ability and solve the overfitting problem, we want to use the Smoothness Inducing Adversarial Regularization technique to improve the smoothness of the model prediction and apply the Bregman Proximal Point Optimization methods to the fine-tuning of model hyperparameters in order to prevent aggressive updating. We will go through the details of the approaches we utilized including the minBERT model, Smoothness-Inducing Adversarial Regularization, and Bregman Proximal Point Optimization methods.

## 4.1 Baseline Model: minBERT

We implemented a transformer-based model, minBERT model, as our baseline model. The model includes a Multihead Attention mechanism that allows it to attend to different parts of the input text. Specifically, it joints scaled-dot product attention from different heads. Scaled-dot product consists of queries, keys of dimension $d_k$, and values of dimension $d_v$. This product could be calculated as follows: after computing the dot products of the query with keys, the product is divided by $\sqrt{d_k}$, and applied with a softmax function to get the weights on values. We get:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Then we concat multi-heads to get the multihead attention:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

where:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $W^O \in \mathbb{R}^{hd_v \times d_{model}}$

The base BERT model has 12 identical Encoder Transformer layers and each layer consists of multi-head attention, followed by an additive and normalization layer, a feed-forward layer, and a final additive and normalization layer. The minBERT model is trained using the Adam optimizer, which is a stochastic gradient descent method that computes an adaptive learning rate for each weight parameter. This method updates exponentially decaying moving averages of past gradients and squared gradients at each step and uses hyperparameters to control the rate of exponential decay of the averages. With moving averages initialized at 0, the algorithm performs bias correction to get the two gradients.

### 4.2 Smoothness-Inducing Adversarial Regularization

We add regularization into the loss function, thus we should solve the optimization for fine-tuning below:

$$min_\theta F(\theta) = L(\theta) + \lambda_s R_s(\theta)$$

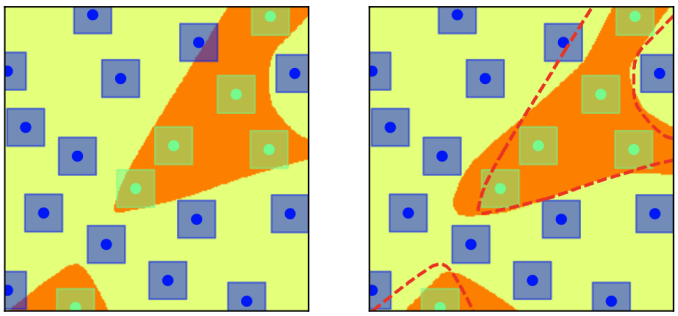
$L(\theta)$ is loss function:

$$L(\theta) = \frac{1}{n}\Sigma_{i=1}^{n} l(f(x_i;\theta), y_i)$$

$R_\theta$ is the smoothness-inducing adversarial regularizer, we should keep $\|\bar{x_i} - x_i\|_p \leq \epsilon$:

$$R_s(\theta) = \frac{1}{n}\Sigma_{i=1}^{n} max l_s(f(\bar{x_i};\theta), f(x_i;\theta))$$

$l_s$ is chosen as the symmetrized KL divergence.



(a) (b)

Figure 1: Decision boundaries learned without(a)and with(b)smoothness-inducing adversarial regularization, respectively. The red dotted line in (b) represents the decision boundary in (a). As can be seen, the output f in (b) does not change much within the neighborhood of training data points.[6]

### 4.3 Bregman

This model imposes a strong penalty at each iteration to prevent the model from aggressive updates. At the $(t+1)^t h$ iteration, the vanilla Bregman proximal point takes:

$$\theta_{t+1} = argmin_\theta F(\theta) + \mu D_{Breg}(\theta, \theta_t)$$

$\mu$ is a tuning parameter, and $D_{Breg}$ is Bregman divergence:

$$D_{Breg}(\theta, \theta_t) = \frac{1}{n}\sigma_{i=1}^{n} l_s(f(x_i;\theta), f(x_i;\theta_t))$$

## 5 Experiments

### 5.1 Data

We mainly utilized four datasets:

- Stanford Sentiment Treebank (SST) dataset. It contains of 11,855 single sentences from movie reviews.
- Quora Dataset. It consists of 400,000 question pairs with labels indicating whether particular instances are paraphrases of one another.
- SemEval STS Benchmark Dataset. It consists of 8,628 different sentence pairs of varying similarity on a scale from 0 (unrelated) to 5 (equivalent meaning).

### 5.2 Evaluation method

To determine if our modified model is effective in improving the model performance, we employ the EM score and the F1 score as our evaluation metrics. The EM score measures the percentage of cases where the model predicts the exact answer. And the F1 score is the harmonic mean of precision and recall, which measures the balance between the model's ability to correctly identify positive and negative cases. By comparing the EM score and the F1 score of train data and dev data, we could determine whether our model is overfitting and whether our modification has improved the generalization performance of the model. And we could determine if our modification improves the performance over the baseline model and assess the effectiveness of our modification By comparing the EM score and F1 score of each model.

### 5.3 Experimental details

We trained the minBERT model to simultaneously perform sentiment analysis, paraphrase detection, and semantic textual similarity tasks, which is implemented by the Round-Robin method. Experiments are performed with two techniques: the Smoothness Inducing Adversarial Regularization technique and the Bregman Proximal Point Optimization method. To improve the performance on all three downstream tasks, we adopted a batch training strategy with three datasets in each batch, and the batch sizes are set to 2 for the SST and STS dataset and 24 for the Quora dataset. The epoch is 10, the learning rate is 1e-5, hidden size is 768 and the hidden layer dropout probability is 0.1. All experiments utilize the Adam optimizer with learning rate = 1e-5, eps = 1e-6, $\beta_1$ = 0.9, $\beta_2$ = 0.999.

- **Baseline model.** The pretrained minBERT model. The training loss is calculated by the cross-entropy between the predictions on the input and the true labels of the input.
- **Smoothness-inducing adversarial regularized model.** Finetuned baseline model with adversarial training. We add additional perturbation to the input and introduce this adversarial training loss into the total loss. Tuning parameters in the experiments are: the learning rate in the adversarial training = 1e-3, the iteration variable iter_var $\in (1, 3)$, the noise scale of the perturbation $\epsilon$ = 1e-5, and $\lambda_s \in (1, 0.1)$.
- **Bregman proximal point optimized model.** Finetuned baseline model with Bregman proximal point optimization. During training, the model parameters are updated depends on all its predecessors, which are recorded by the Bregman momentum. The tuning hyperparameters are $\beta \in (0.1, 0.2)$ and $\mu \in (0.999, 0.9)$.

### 5.4 Results

Table 1: Summary of Prediction Accuracy on Dev Examples.

| **Model** | **Sentiment** | **Paraphrase** | **Similarity** |
|---|---|---|---|
| Baseline | 0.515 | 0.849 | 0.874 |
| Baseline + Adv($\epsilon$ = 1e-5, iter = 1, $\lambda$ = 1) | 0.510 | 0.867 | 0.872 |
| Baseline + Adv($\epsilon$ = 1e-5, iter = 3, $\lambda$ = 1) | 0.520 | 0.865 | 0.869 |
| Baseline + Adv($\epsilon$ = 1e-5, iter = 3, $\lambda$ = 0.1) | 0.491 | 0.869 | 0.879 |
| Baseline + Bregman($\mu$ = 0.2,$\beta$ = 0.9) | 0.508 | 0.864 | 0.869 |
| **Baseline + Bregman**($\mu$ = 0.1,$\beta$ = 0.999) | **0.518** | **0.868** | **0.871** |
| Baseline + Adv ($\epsilon$ = 1e-5, iter = 1, $\lambda$ = 1) +Bregman($\mu$ = 0.2,$\beta$ = 0.9) | 0.498 | 0.870 | 0.873 |

$*$ F1 score is only evaluated on the classification tasks, i.e., sentiment analysis and paraphrase detection, and not applied on the regression task, i.e., semantic textual similarity.
$*$ Due to the limit of three predictions test set, only two of the finetuned models with overall better performance are evaluated.

Table 2: EM and F1 Scores on Dev/Test Examples.

| **Model** | **dev EM** | **dev F1**$*$ | **test EM**$*$ |
|---|---|---|---|
| Baseline | 0.746 | 0.862 | N/A |
| Baseline + Adv($\epsilon$ = 1e-5, iter = 1, $\lambda$ = 1) | 0.750 | 0.857 | |
| Baseline + Adv($\epsilon$ = 1e-5, iter = 3, $\lambda$ = 1) | 0.751 | 0.857 | N/A |
| Baseline + Adv($\epsilon$ = 1e-5, iter = 3, $\lambda$ = 0.1) | 0.746 | 0.859 | |
| Baseline + Bregman($\mu$ = 0.2,$\beta$ = 0.9) | 0.747 | 0.856 | N/A |
| **Baseline + Bregman**($\mu$ = 0.1,$\beta$ = 0.999) | **0.752** | **0.860** | **0.759** |
| Baseline + Adv ($\epsilon$ = 1e-5, iter = 1, $\lambda$ = 1) +Bregman($\mu$ = 0.2,$\beta$ = 0.9) | 0.747 | 0.861 | 0.750 |

According to the results, the overall model performance can be improved by the two techniques, which aligns with our expectations. The best result we have achieved is to improve $0.9\%$ accuracy in the sentiment analysis with applying adversarial training, and $2.5\%$ accuracy in the paraphrase detection with both adversarial training and Bregman optimization, and $1.7\%$ correlation in the semantic textual similarity task with adversarial training. The average EM score on the three downstream tasks is best improved by $0.8\%$ on the dev set, and achieved $0.759$ on the test set, with SST test accuracy $0.540$, paraphrase test accuracy $0.865$ and STS test correlation $0.872$. As for the F1 score, it suggests that, the model made close but not exact predictions, so it has lower EM score but still has higher F1 score. We noticed that both of the two methods show stronger improvements on the paraphrase detecting task, which also aligns with our expectations. For the smoothness inducing adversarial regularization method, the larger dataset may have more variability and noise in the data, making it more challenging for the model to learn a good representation. Adding perturbations to the input during training can help the model learn to be more robust to this variability, which can improve the model's generalization performance on the larger dataset. And for the Bregman proximal point optimization method, since it updates the model parameters depends on all the predecessors recorded by the Bregman momentum, it has a strong ability to capture more information in the larger dataset and allows the model to incorporate more information from the data during training and to make more informed decisions about how to update the model parameters. When training on larger datasets, the amount of available information is significantly higher than in smaller datasets, and the Bregman momentum can capture and leverage this information to improve the convergence rate and the quality of the final solution.
However, the performance of the model on each downstream task also shows some unexpected variations when using the smoothness inducing adversarial training, which leads the model to different directions at different tasks. In the experiments, this method leads the model in different directions for different tasks, resulting in mixed performance outcomes. It can improve the EM accuracy on paraphrase detection, while it leads to unexpected effects for sentiment analysis and semantic textual similarity, which depends on the specific hyperparameters chosen for the tasks. This behavior is not aligned with our initial expectations, given that the SST dataset ($\sim$8,500) for sentiment analysis and the STS dataset ($\sim$6,000) for semantic textual similarity are smaller than the Quora dataset ($\sim$140,000) for paraphrase detection. We expected that this regularization technique would demonstrate a stronger influence on tasks that suffer from overfitting drawbacks due to limited data, which is typical in NLP applications.

## 6 Analysis

Our system applied the SMART learning framework through regularized optimization techniques to achieve more robust and efficient fine-tuning and better generalization performance of our model. We use the smoothness inducing adversarial regularization to improve the smoothness of the model prediction and apply the Bregman proximal point optimization to the fine-tuning of model parameters in order to prevent aggressive updating.

- The smoothness inducing adversarial regularization adds additional perturbation to the input, and uses gradient ascent to determine a training direction with the maximum adversarial training loss. As this part of loss is added into the original training loss, the regularization

method turns the learning direction of the model to improve its generalization ability and robustness. Our experiments show that this method is more effective on the paraphrase detection task with a larger dataset, and may fail on the sentiment analysis and the semantic textual similarity tasks with smaller datasets due to the chosen hyperparameters.

- The Bregman proximal point optimization works through introducing all the predecessors to update the model parameters, which are recorded by the Bregman momentum. During the training process, the Bregman momentum is used to determine the divergence between the current model parameters and all their previous values. By adding this part of training loss to the original loss, this method imposes a strong penalty at each iteration to maintain a steady learning rate of the model parameters. According to our results, this method can success on the sentiment analysis and the paraphrase detection tasks with proper hyperparamter configurations.
- The SMART framework with both of these two methods gains success on the paraphrase detection task in our experiments, but may not perform well on the sentiment analysis and the semantic textual similarity tasks.

Overall, the SMART system can be useful for preventing overfitting and aggressive updating and improving generalization performance, but its effectiveness may depends on not only the dataset size and the type of the downstream task, but also the hyperparameters and the quality of the dataset.

## 7 Conclusion

In this project, we present the SMART framework for three individual tasks, sentiment analysis, paraphrase detection, and semantic textual similarity. Based on the default minBERT in the starter code, we implement a multitask classifier on the top-level layer to simultaneously perform tasks. To finetune the minBERT model, we implement the smoothness inducing adversarial regularization to improve the smoothness of the model prediction and apply the Bregman proximal point optimization to the fine-tuning of model parameters in order to prevent aggressive updating. With proper hyperparameters chosen, our model shows a good performance on both dev and test datasets, especially for the paraphrase detection task. In our experiments, we also notice that the Bregman proximal point optimization requires more time-consuming computation, so we need to trade off between the performance improvements and the training efficiency, which can be further studied in the future explorations.

## References


[1] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[2] Matthew E Peters, Sebastian Ruder, and Noah A Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*, 2019.

[3] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[6] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*, 2019.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[8] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[9] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

[10] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[11] David H Gutman and Javier F Pena. A unified framework for bregman proximal methods: subgradient, gradient, and accelerated gradient schemes. *arXiv preprint arXiv:1812.10198*, 2018.

[12] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.