IIT Bombay

22M2119

Home Logout IITB Website

Roll no	22M2119	Name	Badri Vishal Kasuba
Department	Computer Science and Engineering	Program	Master of Science By Research

Payment | Performance Summary | New Entrants | Graduation Requirements | Personal Information | Forms/Requests

Academic Performance Summary

Year	Sem	SPI	CPI	Sem Credits Used for SPI	Completed Semester Credits	Cumulative Credits Used for CPI	Completed Cumulative Credits
2023	Spring	8.0	9.03	12.0	12.0	58.0	58.0
2023	Autumn	0.0	9.3	0.0	0.0	46.0	46.0
2022	Spring	9.25	9.3	24.0	24.0	46.0	46.0
2022	Autumn	9.36	9.36	22.0	22.0	22.0	22.0

Semester-wise Details

*This registration is subject to approval(s) from faculty advisor/Course Instructor/Academic office.

Report Problem

Year/Semester: 2023-24/Spring

Course Code	Course Name	Credits	Tag	Grade	Credit/Audit
CS 753	Automatic Speech Recognition	6.0	Department elective	BB	C
CS 769	Optimization in Machine Learning	6.0	Department elective	BB	C

Year/Semester: 2023-24/Autumn

Course Code	Course Name	Credits	Tag	Grade	Credit/Audit
CS 663	Fundamentals of Digital Image Processing	6.0	Additional Learning	BC	C

Year/Semester: 2022-23/Spring

Course Code	Course Name	Credits	Tag	Grade	Credit/Audit
CS 694	Seminar	4.0	Core course	AA	C
CS 763	Computer Vision	6.0	Department elective	BB	C
CS 772	Deep Learning for Natural Language Processing	6.0	Department elective	AB	C
CS 778	M.S. R&D 2	8.0	Core course	AA	C
CS 899	Communication Skills	6.0	Core course	PP	N
TA 101	Teaching Assistant Skill Enhancement & Training (TASET)	0.0	Core course	PP	N

Year/Semester: 2022-23/Autumn

EFFECT OF ENFORCING OVERSMOOTHING IN TRANSFORMERS FOR TEXT CLASSIFICATION

Dhruv Kudale, Badri Vishal Kasuba & Kishan Maharaj
Computer Science and Engineering Department
Indian Institute of Technology Bombay
India
{22m2116, 22m2119, 22m2122}@iitb.ac.in

ABSTRACT

A recent line of work argues that transformers are inherently low-pass filters, leading to oversmoothing as the number of layers increases. This oversmoothing phenomenon eventually causes the representation of all features to converge to the same vector. In this project, we aim to explore the effect of oversmoothing induced by transformers on the text classification task. We report our scores on the GLUE (General Language Understanding Evaluation) benchmark with text classification tasks and examine how they vary based on the extent of observed oversmoothing. Our investigation sheds light on the relationship between oversmoothing in transformer architecture and the performance of text classification models, contributing to a deeper understanding of these aspects of language understanding. In other words, we aim to optimize the effect of oversmoothing for better accuracy of text classification tasks.

1 INTRODUCTION

The oversmoothing phenomenon in transformers has garnered significant attention in recent research due to its potential impact on model performance, particularly in sequence prediction or classification tasks. Transformers, while being powerful models for Natural Language Processing (NLP) tasks, are inherently susceptible to oversmoothing, especially as the number of layers increases (Geshkovski et al. (2024)). Oversmoothing occurs when the transformer model tends to blur or flatten out the representations of different input features, ultimately leading to a loss of discriminative information. This phenomenon is similar to the application of a low-pass filter, where high-frequency details are suppressed (regions with high variance), resulting in a more generalized, smoothed-out representation. Based on the task at hand, oversmoothing can limit the generalization capabilities of transformer-based models. However, it is shown that smoothing can enhance generalization for image classification tasks but can also lead to performance degradation for text generation tasks(Dovonon et al. (2024)). We attempt to study the effect of oversmoothing in transformers for the text classification task to determine whether oversmoothing will improve performance. In the context of our problem, we attempt to control the extent of oversmoothing brought about by transformers to achieve a balance between model complexity and generalization. The oversmoothing effect should be high enough to cluster the common patterns of the data also and adequately low enough to capture diverse patterns in the data.

1.1 OUR CONTRIBUTION

We aim to propose and implement a transformers-based network and introduce another hyperparameter to control the degree or extent of oversmoothing through the number of layers and other parameters. Eventually, we will see how classification performance is affected by this smoothing extent. We also report the performance of our model on different benchmark datasets based on different extents of smoothing and make concrete conclusions.

1.2 Problem Setting

Text classification involves assigning a class from a set of predefined categories to open-ended text. This problem includes automatically categorizing textual data based on its content. This task is fundamental in NLP and has numerous real-world applications across various domains. Here's a more detailed overview of the problem:

- **Input**: The input data for text classification typically consists of textual content from articles, emails, reviews, tweets, or any other form of text. Each text sample is associated with one or more predefined categories or labels, which serve as the target variables that the model aims to predict.
- **Output**: The output of a text classification model is a prediction of the category or label for each input text sequence. The categories can be binary (e.g., spam/not spam) or multi-class (e.g., sentiment analysis with positive, negative, and neutral classes).

Text classification models are typically evaluated using various performance metrics, including accuracy, precision, recall, F1-score, area under the ROC curve (AUC), and confusion matrices. The choice of evaluation metrics depends on the specific requirements and characteristics of the classification task. We follow the evaluation metrics used by the benchmark datasets for text classification like CoLA, and SST-2 from GLUE (Wang et al. (2019)).

1.3 Objective Function

Since this is a classification problem, we aim to minimize the categorical cross-entropy loss. This loss function is suitable for multi-class classification tasks. It compares the predicted class probabilities to the true class labels across all classes.

Categorical Cross-Entropy Loss = $-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \cdot \log(p_{ij})$

Where,

- y_{ij} as the indicator function where $y_{ij} = 1$ if sample i belongs to class j, and $y_{ij} = 0$ otherwise.
- p_{ij} as the predicted probability that sample i belongs to class j.
- N is the total number of samples
- C is the total number of classes.

This loss function penalizes the model more severely when it assigns low probabilities to the true classes. It encourages the model to predict higher probabilities for the correct classes. Since this loss function is often used in conjunction with softmax activation, we also will be applying the softmax activation at the output layer of our transformer-based network. The softmax function ensures that the predicted probabilities sum up to 1 so that they can be interpreted as class probabilities. By minimizing this loss, we aim to make our model classify text in a better manner. To analyze the effect of oversmoothing on this text classification task, we propose to do the following experiments.

2 Proposed Experiments

We are interested in studying and interpreting the properties of transformers in the context of oversmoothing with textual data. We want to investigate the evolution of Query(Q), Key(K), and Values(V) matrices along the layers to understand if the oversmoothing phenomenon (or clustering from the perspective of particle dynamics Geshkovski et al. (2024)) is helping the text classification task or hindering the performance of the architecture. For this, we plan to analyze the gradients and other mathematical properties (for example, spectral properties like eigenvalues) of the Query, Key, and Value matrices. We will then make changes in the weight-update rule of the transformer to control the over-smoothing behavior so that it can benefit the text classification task. We will also study the inter-layer gradient flows in this regard. This will be followed by insights into the depth (number of layers) and width (hidden dimension) of the transformer-based architecture in the context of oversmoothing. We will end our discussion with an empirical analysis of how oversmoothing can be adjusted to yield better text classification performance.

2.1 DATASETS

We will use the General Language Understanding Evaluation (GLUE) benchmark dataset (Wang et al. (2019)), which is a collection of diverse natural language understanding tasks. GLUE is designed to evaluate the performance of models across a range of linguistic phenomena. GLUE consists of several individual datasets, each representing a distinct NLP task. These tasks cover various aspects of language understanding, including sentence-level semantics, textual entailment, sentiment analysis, and more. We will particularly focus on the tasks that boil down to text classification to report our results. The following list gives a detailed overview of different types of tasks and datasets in GLUE that we will be explicitly focusing on.

1. **Single-Sentence Tasks**:
    - CoLA (Corpus of Linguistic Acceptability): This involves determining whether a given sentence is grammatically acceptable or not.
    - SST-2 (Stanford Sentiment Treebank): This focuses on binary sentiment classification, where the goal is to predict whether a sentence expresses a positive or negative sentiment.
2. **Similarity and Paraphrase Tasks**:
    - MRPC (Microsoft Research Paraphrase Corpus): This task involves determining whether pairs of sentences are semantically equivalent or not.
    - QQP (Quora Question Pairs): It focuses on determining whether pairs of questions asked on Quora are semantically equivalent or not.

We will be redefining each of the above-mentioned tasks as a text classification problem. Our experimentation includes determining the classification accuracy and scores for the respective tasks (as reported by GLUE benchmarks) based on the degree of oversmoothing effect.

2.2 METHODOLOGY

We will be exploring all of the following methodologies to bring about and control the extent of smoothing in our transformer-based model.

2.2.1 ENFORCING WEIGHT UPDATE CONDITION

We use the proposed way to reparameterize the transformer update weights to control the update's filtering behavior. (Dovonon et al. (2024)) This enforces that the transformer-based model is being made to act as a low pass filter (Wang et al. (2022))

2.2.2 MAKING ARCHITECTURAL MODIFICATION

To enhance the smoothing effect in a transformer model, one common approach is to increase the number of layers (Geshkovski et al. (2024)). This strategy allows the model to capture more complex patterns and dependencies in the data, leading to better generalization and smoother predictions. Additionally, adjusting various hyperparameters like batch size, learning rate, regularization, etc. can further optimize the model's performance and fine-tune its behavior according to the specific classification task at hand.

3 ANTICIPATED CONCLUSIONS AND FUTURE WORKS

Through this proposed study, we wish to present the outcomes and in-depth analysis of the effects of oversmoothing on the performance of text classification tasks within the domain of language understanding. We also plan to investigate this phenomenon within the broader context of oversmoothing brought about by transformers. As suggested by some of the recent works smoothing can occasionally enhance outcomes, as observed in the case of image classification. Therefore, we will be exploring, assessing, and presenting the effects of oversmoothing on the text classification tasks.

REFERENCES

Gbètondji J-S Dovonon, Michael M. Bronstein, and Matt J. Kusner. Setting the record straight on transformer oversmoothing, 2024.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. A mathematical perspective on transformers, 2024.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.

Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice, 2022.