[Paper Review] Universal Language Model Fine-Tuning for Text Classification

Paul Jason Mello
Department of Computer Science and Engineering
University of Nevada, Reno
pmello@unr.edu

Abstract

"Inductive transfer learning has greatly impacted computer vision, but existing approaches in NLP still require task-specific modifications and training from scratch. We propose Universal Language Model Fine-tuning (ULMFiT), an effective transfer learning method that can be applied to any task in NLP, and introduce techniques that are key for fine-tuning a language model. Our method significantly outperforms the state-of-the-art on six text classification tasks, reducing the error by 18-24% on the majority of datasets. Furthermore, with only 100 labeled examples, it matches the performance of training from scratch on 100× more data. We open source our pretrained models and code." [1]

1 Summary

Released in 2018, this paper develops a method to apply transfer learning to the medium of text. In order to achieve this, the authors introduce a few novel ideas aimed at improving the models sample efficiency. Fundamentally, their methods focus purely on improving the training and fine-tuning process. This results in an error reduction rate of approximately $21\tilde{\%}$ and decreases required training data by $100\times$ when compared to similar performing model sizes.

2 Motivation

In 2018, NLP had not yet had their "computer vision moment". Computer vision had seen significant advancements, particularly in the development of foundational models which, through fine-tuning, could be applied to task specific domains. However, there are challenges in NLP including overfitting or catastrophic forgetting particularly regarding the fine-tuning of these models for domain specific tasks, which make applying transfer learning from CV to NLP difficult. The researchers of this work successfully introduce methods to allow for transfer-learning and significant progress on fine-tuning in the NLP setting. Moreover, they open source their models and code.

3 Main Contributions

In this work, the researchers introduce a few novel contributions to NLP, demonstrating the feasibility and performance improvements that can be gained through smarter training methodologies. Particularly, they propose Universal Language Model Fine-Tuning (ULMFiT). ULMFiT is capable of working in any language modeling task given their approach is defined by improved training and fine-tuning with applications for robust inductive transfer. Their contributions in the development of ULMFiT consist of the following:

3.1 Key Contributions

- They achieve transfer learning in the field of NLP. This contribution is highlighted as it is the sum of all other contributions.
- The first contribution is "slanted triangular learning rates". The authors improve learning rates by starting with a low learning rate, then rapidly increase the learning rate over a short number of epochs, $\tilde{200}$ in the paper. At a specific epoch, the learning rate abruptly stops growing and begins gradually decreasing back towards the initial learning rate over the course of the remaining training, $1\tilde{3}00$ in the paper. The rapid increase in learning rate allows the model to find a general minima, then through gradual decreases in the learning rate, the model is tuned to find strong local minima.
- Discriminative fine-tuning is another method they introduce. This is composed of tuning each layer with a unique learning rate. Since each layer learns a unique feature representation, they provide each layer with its own granularity of learning. Empirically, they find taking the learning rate from the last layer η^L and fine-tuning only the last layer from this learning rate, then updating the other layers through $\eta^{l-1} = \frac{\eta^l}{2.6}$ provides the best dynamics.
- Gradual layer-wise unfreezing is another key contribution of this paper consisting of gradually unfreezing each layer over time. Starting from the last layer, which contains the least general knowledge, all other layers are frozen. Over the training epochs, the researchers unfreeze each layer compounding as they backpropagate. In other words, the final layer is unfrozen, fine-tuned, then the last and second to last layer is unfrozen and fine-tuned with this process repeating until all layers are unfrozen at the same time and converge.
- The final contribution they introduce is backpropogation for text classification (BPT3C). This consists of dividing a document into fixed-length batch sizes. They keep track of the hidden states with the mean and max pooling as we will describe later, then specifically update only the weights which contributed to the final prediction. This is essentially a dynamic backpropogation focusing on activated neurons.

Notably, a few lesser components are introduced I would wish to highlight here. Since documents will have sparse meaning and context strung throughout its contents, they use concat pooling which consists of concatenating the hidden state at the last timestep h_T and pool the max and mean representations of all other hidden states while maximizing GPU memory utilization under $h_{\text{concat pooling}} = [h_T, \text{maxpool}(H), \text{meanpool}(H)]$. They also utilize bidirectional language models which define a model that is pretrained both in a forward and backward language model.

3.2 Universal Language Model Fine-Tuning

ULMFiT consists of three stages that we outline in order below.

- The first stage is to train a full model on a general-domain corpus. This provides a model with general understanding regarding the field or task it is required to fulfill. This approach focuses on learning high-level feature representations and utilizing a lot of data to provide a model with ample understanding.
- The second stage begins fine-tuning the trained model on a smaller corpus of high quality, task specific data. To improve the sampling efficiency they utilize discriminative fine-tuning with slanted triangular learning rates.
- The final stage, and what they consider to be the most important, consists of using a classifier which is fine-tuned to the specific task through gradual unfreezing, discriminative fine-tuning, and slanted triangular learning rates. This results in the preservation of low-level feature representations found in the task-specific corpus, while adapting the high level feature representations learned from the general-domain corpus.

To see how this all comes together, consider the following. Data can be interpreted as a probability distribution, where some data, patterns, or features are more likely to occur than others. General domain data will not be the same probability distribution as task-specific data. The goal of stage one is to capture that initial general domain distribution. The more general the data, the more

general the distribution. In the second stage we focus on capturing task specific data. In this way we effectively train the model to find a balance between general domain knowledge and our task specific knowledge, thus moving the distribution into a balance between the two. In the final stage, the researchers double down on stage two by updating the weights of the model in a more precise manner towards the task specific data distribution. This effectively tweaks the general knowledge to provide better overall coverage of the task specific data. Essentially, sacrificing general domain knowledge and shifting it towards task-specific knowledge. Through this process, the model co-adapts the general domain knowledge to be better suited to task specific output distributions.

4 Strengths and Weaknesses

This approach has many strengths, particularly in its universality as the only real components being effected in training or fine tuning is the addition of a classification output layer, layer-wise learning rates, focused dynamic backpropogation, and compounded layer-wise unfreezing. In each instance, we are simply improving the model training and fine-tuning with smarter approaches. These surmise into a strong language model by significantly improving the learning process and sample efficiency. One weakness of the paper, is a lack of applications of this method to other mediums or domains such as images or reinforcement learning. This universal approach should easily allow for the transfer of these methods to other models.

4.1 Strengths

The universality of this approach is its core strength. The authors focus on improving the model training dynamics by focusing on "squeezing" more from training through structured methodologies on data timing and learning. Prior works likely did not incorporate so many different methods which lead to lost performance. Open sourcing the model to provide the NLP domain with a foundational model and publishing the code are also strengths of their research.

4.2 Weaknesses

The most defining weakness of this paper is a lack of "all-out" training. In the paper they list the model as AWDLSTM [2] and the data as Wikitext. Notably, AWDLSTM does not utilize a self-attention mechanism. The authors of this paper utilize the same model to provide an exact comparison between approaches. While this is the correct approach to demonstrate improvements over prior approaches, seeing the full capabilities of their approach with an attention infused AWDLSTM and increasing the data corpus would be a thoughtful experiment. I note this especially because they open source their model to the NLP community as a foundational model.

4.3 Areas of Improvements

While this paper does a very strong job at making its case for improved fine-tuning methodologies, there are still many points which could be improved. Specifically, more dynamic approaches to per-layer learning rates could be a simple, but very strong improvement. Rather than relying on the same learning rate scaled differently, each layer may benefit for more dynamic learning rates, or simply different learning rates. Covering additional tasks was a weakness I previously mentioned, but also a potential area of improvement. This could help demonstrate the capabilities in other domains and datasets and especially the universality of this approach. Additionally, while they focus on efficient sampling in this paper, further additions to reducing the computational cost through other techniques could offer better scalability and efficiency. Consider distillation or pruning nodes as potential points for further improvements.

5 Discussion

Published in 2018, this paper defined many of the advancements of fine-tuning and model design. It is the complete package of ULMFiT that makes this approach so successful at inductive transfer learning to NLP and potentially other tasks. The field has since adopted or co-opted many of these approaches and design decisions with varying degrees of success and implementation. Today layer-wise dynamic learning rates, fine tuning on task specific data, and model freezing are considered

fundamentals to the AI domain. With this in mind, and a lack of personal familiarity with the SOTA of NLP in 2018, this approach defines a well optimized model. It exhibits strong understanding of the internal mechanisms of deep neural networks and how to leverage those mechanisms.

6 Conclusion

ULMFiT was designed in a way to be task-agnostic by focusing on the layer-wise improvements and fine-tuning. Through this research they are able to train models to similar performance with significantly less data. The subsequently achieve a reduction in error rate to the tune of $21\tilde{\%}$. Through methods like discriminative fine-tuning, slanted triangular learning rates, and gradual unfreezing they are able to handle common challenges in the field of fine-tuning NLPs such as overfitting and catastrophic forgetting. By open sourcing the models and code, the researchers contributed to the democratization of AI, and likely sparked many additional researchers to seek sample efficient methods in their own work.

References

[1] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.

[2] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models, 2017.