CS510 Project Proposal: Exploring Efficiency in IR Model Pruning

Daniel Campos
dcampos3@illinois.edu

Abstract

Neural Networks have proven to be effective methods for retrieving relevant document however these methods can prove difficult to use in production. Neural models tend to be large and require specialized hardware for inference(GPU, TPU, FPGA, etc) which makes them difficult to deploy for all use cases. Researchers in fields like computer vision have leveraged methods in model compression such as structured and unstructured pruning to decrease model size while preserving or exceeding original model performance. In our work we will explore how model compression behaves on information retrieval tasks.

Keywords: information retrieval, network pruning

ACM Reference Format:
Daniel Campos. 2021. CS510 Project Proposal: Exploring Efficiency in IR Model Pruning. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Neural Networks have become popular choices for complex computation tasks like image recognition [12], image generation [10], speech processing [29], and question answering [24]. These networks have grown to hundred billions of parameters [3] which require specialized hardware like clusters of GPUs and FPGAs in order to infer on unseen data. Recent work has shown that larger networks can learn quicker [17], are more accurate and are more sample efficient [14]. Seeking to allow the improvements that large models have brought to be used on smaller devices and in a more energy efficient way, researchers have explored methods like: quantization, distillation and pruning, which produce smaller networks that approximate, match, or exceed the original network performance.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Model distillation, quantization, and pruning have emerged as successful methods to produce smaller networks from the original over-parametized network. In model distillation [1] the original network is used to train a smaller network to mimic the behavior of the large network. In model quantization [11] models are made smaller by reducing the numbers of bits that represent each weight. Model Pruning [16] has focused on finding sub-networks in the original network by structured and unstructured pruning. In structured pruning, successful sub-networks are found by removing neurons [28] or larger network specific structures like attention heads [27]. In unstructured pruning the successful sub-networks are found by setting individual weights to zero [15].
Using network pruning, The Lottery Ticket Hypothesis [8] proved the concept that in large neural networks there can exist a sub network which can match the accuracy of the full network despite its smaller size. Building on the notion that there are many sub networks in an overparamertized network, network pruning can be formulated as an optimization problem. Given an initial structure S and a target network size ϵ the goal is to find the sub network s_m of size ϵ which maximizes the model performance.
We believe that neural models will only continue to grow and we seek to understand how model sparsity effects performance for Information Retrieval (IR) tasks. To research the topic we will apply standard unstructured pruning methods to a variety of neural information retrieval methods and validate on model performance.

2 Problem Description and Project Plan

In this project we seek to answer the following questions:

1. Can we achieve a high sparsity(80%+) on neural IR methods and retain performance?
2. How do results of IR pruning compare to computer vision and other NLP tasks.

To answer these questions we will use the MSMARCO passage ranking dataset (MPR) [4] and and explore how performance on well established neural IR method varies with sparsity.
The MPR is a dataset which originates from a question answering dataset. The original dataset consists of 8.8 million passages and 1 million user queries issued to a commercial search engine. For each query, a judge read the top 10 passages extracted by Bing and wrote an answer to the query and attributed the answer to one or more passages. The MPR

is an adaptation of the question answering dataset to produce a passage ranking tasks. The document collection is a combination of all unique passages from the question answering task and the relevance judgements are binary labels representing which passage was used to generate the human generated answer. As the tasks is recall focused, the evaluation metric is Mean Reciprocal Rank (MRR) @1000. The large data regime of this dataset has made it one of the most popular evaluation frameworks for neural IR with over 100 submissions and was used in the 2019, 2020, and 2021 TREC Deep Learning Track.

Time is our main constraint but we seek to implement a variety of neural IR models: a neural language model based on BERT [7], a non language model transformer based model such as Conformer-Kernel [20], and a convolutional neural model such as Conv-KNRM [6].

For each of these models we will prune using gradual magnitude pruning with target sparsity's of 50-95% in 5% increments. This will effectively give us 30 different models which we will evaluate on the validation portion of the MSMARCO passage ranking and the TREC Deep Learning dataset.

We believe the results of our work could motivate our future research and that of the broader IR community.

2.1 Work Plan

Our work plan is broad but is laid out below. We may decrease the amount of pruned models and models implemented based on compute availability and coding efficiency.

1. Initial dataset exploration - March 30th
2. Baseline model running and evaluated - April 15th
3. Models pruned and evaluated - April 30th
4. Presentation of results and paper - May 5th

3 Related Work

3.1 Neural Information Retrieval

Like many computer science fields, Information Retrieval has seen neural network based systems out performing previous systems. Systems like DUET [19], DSSM [13], C-KRNM [6], and C-DSSM [25] built performant systems which rivaled performance of traditional non neural methods. With the introduction of neural language models countless BERT-based models produced new state of the art results similar to many other NLP fields. Building on the success of the transformer and seeking to build document wide dependency modeling models leveraging the transformer such as Conformer-Kernel[20] have shown efficiency in quality and scale.

3.2 Model Compression

Neural network compression is an area that has attracted the attention of researchers for the last few decades. Methods for producing smaller networks that approximate original network performance include: distillation, quantization, structured and unstructured pruning. While each of these methods can compress models substantially on their own many researchers have found that some combination of these methods can produce the smallest models with the highest performance [21], [23].

Model distillation [1] addresses compressing models by first training a large network and calling it a teacher. Then using this teacher model a smaller student model learns to approximate what the teacher model would do. This framework is quite popular because it can leverage existing large models easily and the student model can be designed to fit the application requirements in terms of speed and model size. Distillation has been one of the most common methods of deployment of large scale language models where student models like DistilBERT [22] can approximate full model performance at a fraction of the size.

Model quantization [9] [11] addresses compressing models by reducing the number of bits that are require to represent parameters in a model. In simple implementations this means changing representation of weights from Float32 to float16(effectively cutting model size in half). Complex implementations tune networks to find the smallest amount of bits that can be used for weights, biases, and gradient updates using values as low as 1 bit [5]. Quantization is particularly effective because it both leverages that networks are defaulted to a level of precision which is too high and by decreasing the size of representations networks are forced to share weights making networks more robust.

Model pruning [16] addresses model compression by decreasing the connection in a network. The goal of network pruning is to produce a sub network of the original network which optimizes some network property(accuracy, speed, robustness) while preserving the original network function. Network pruning has been show to produce a similar effect to random noise injection [2] and this noise can be used to make the network more efficient. Bartoldson et al., showed that network pruning is not just used for decreasing size but can be used to increase the generalization of the network. As mentioned earlier, there is structured pruning and unstructured pruning. In structured pruning, the structure of the network is altered by removal of entire neurons, layers, or portions neural network. This method has proven especially successful in language model compression where despite having dozens of attention heads [26] few heads do most of the work and the rest can be removed [18]. In unstructured pruning the network structure is altered by removal of individual weights. Unstructured pruning when paired with optimization engines can produce networks that are smaller, more accurate, and run faster than the original network.

References

[1] Jimmy Ba and R. Caruana. 2014. Do Deep Nets Really Need to be Deep? *ArXiv* abs/1312.6184 (2014).

[2] Brian Bartoldson, Ari S. Morcos, Adrian Barbu, and G. Erlebacher. 2019. The Generalization-Stability Tradeoff in Neural Network Pruning. *ArXiv* abs/1906.03728 (2019).

[3] T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, T. Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *ArXiv* abs/2005.14165 (2020).

[4] Daniel Fernando Campos, T. Nguyen, M. Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, L. Deng, and Bhaskar Mitra. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *ArXiv* abs/1611.09268 (2016).

[5] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1. *arXiv: Learning* (2016).

[6] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-Hoc Search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 126–134. https://doi.org/10.1145/3159652.3159659

[7] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[8] Jonathan Frankle and Michael Carbin. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. *arXiv: Learning* (2019).

[9] Yunchao Gong, L. Liu, Ming Yang, and Lubomir D. Bourdev. 2014. Compressing Deep Convolutional Networks using Vector Quantization. *ArXiv* abs/1412.6115 (2014).

[10] Ian J. Goodfellow, Jean Pouget-Abadie, M. Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NIPS*.

[11] Song Han, Huizi Mao, and W. Dally. 2016. Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. *arXiv: Computer Vision and Pattern Recognition* (2016).

[12] A. Howard, Menglong Zhu, Bo Chen, D. Kalenichenko, W. Wang, Tobias Weyand, M. Andreetto, and H. Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv* abs/1704.04861 (2017).

[13] Po-Sen Huang, X. He, Jianfeng Gao, L. Deng, A. Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. *Proceedings of the 22nd ACM international conference on Information Knowledge Management* (2013).

[14] J. Kaplan, Sam McCandlish, T. Henighan, T. Brown, Benjamin Chess, R. Child, Scott Gray, A. Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *ArXiv* abs/2001.08361 (2020).

[15] S. Kwon, D. Lee, Byeongwook Kim, Parichay Kapoor, Baeseong Park, and Gu-Yeon Wei. 2019. Structured Compression by Unstructured Pruning for Sparse Quantized Neural Networks. *ArXiv* abs/1905.10138 (2019).

[16] Y. LeCun, J. Denker, and S. Solla. 1989. Optimal Brain Damage. In *NIPS*.

[17] Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, K. Keutzer, D. Klein, and J. Gonzalez. 2020. Train Large, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers. *ArXiv* abs/2002.11794 (2020).

[18] Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One? *ArXiv* abs/1905.10650 (2019).

[19] Bhaskar Mitra, F. Diaz, and Nick Craswell. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. *Proceedings of the 26th International Conference on World Wide Web* (2017).

[20] Bhaskar Mitra, Sebastian Hofstätter, Hamed Zamani, and Nick Craswell. 2020. Conformer-Kernel with Query Term Independence for Document Retrieval. *ArXiv* abs/2007.10434 (2020).

[21] A. Polino, Razvan Pascanu, and Dan Alistarh. 2018. Model compression via distillation and quantization. *ArXiv* abs/1802.05668 (2018).

[22] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv* abs/1910.01108 (2019).

[23] Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement Pruning: Adaptive Sparsity by Fine-Tuning. *ArXiv* abs/2005.07683 (2020).

[24] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional Attention Flow for Machine Comprehension. *ArXiv* abs/1611.01603 (2017).

[25] Y. Shen, X. He, Jianfeng Gao, Li Deng, and G. Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. *Proceedings of the 23rd International Conference on World Wide Web* (2014).

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.

[27] Elena Voita, David Talbot, F. Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *ACL*.

[28] H. Wang, Qiming Zhang, Yuehai Wang, and H. Hu. 2019. Structured Pruning for Efficient ConvNets via Incremental Regularization. *2019 International Joint Conference on Neural Networks (IJCNN)* (2019), 1–8.

[29] Y. Zhao, Xingyu Jin, and Xiaolin Hu. 2017. Recurrent convolutional neural network for speech processing. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), 5300–5304.