A Review of "Efficient exact gradient update for training deep networks with very large sparse targets."

Jacob Gursky
St. Olaf College
Northfield, MN 55057
gursky1@stolaf.edu

Abstract

Extremely large output is a persistent problem facing neural networks in many language-related tasks, such as word embeddings and translation. The commonly used softmax activation has many useful properties, but is ultimately too complex to quickly solve large-scale problems. We address multiple approaches to mitigating this problem, specifically an efficient and exact gradient calculation for large outputs proposed by Vincent et al.[17]. We discuss their proposal for a family of spherical loss functions whose complexity are independent of output size, albeit with lower overall model performance. While this approach has some interesting experimental results and implications, we contend that the large output problem, which is very often tied to the rare-word problem, is better solved by a combination of other methods that do not sacrifice performance for speed and can even achieve a higher accuracy to train-time ratio.

1 Introduction

One key application of neural networks in recent years has been language modeling, where there has been great success in tasks such as word embeddings [13] and translation [15]. However, the very large output of these algorithms constrains the performance with a linear increase in compute time by output size. This can be especially problematic in language models with large vocabularies, where the output can easily reach into the hundreds of thousands if not restricted [6]. The root cause of this bottleneck is the softmax activation function, which is defined in the form below using the notation found in [3]:

$$softmax_i(o) = \frac{exp(o_i)}{\sum_{k=1}^{D} exp(o_k)}$$

The softmax activation is often used due to its scale invariance properties and the logarithm of which yields the log-likelihood loss of the output [10]. These convenient properties allow the softmax to accurately model large targets, but incurs a computational cost of $O(dD)$, where d is the number of nodes in the previous hidden layer and D is the vocabulary size, due to its non-sparse output normalization [17]. Note that for language models, D is often many times larger than d ($d \ll D$).

2 The Efficient Exact Gradient Method

There have been numerous efforts to address the large output problem, such as alternative loss functions that are either more efficient or yield sparse outputs [1,5,6,7,10,17], subsampling methods[11], or word segmentation algorithms [2]. The efficient exact gradient method proposed by

Preprint. Work in progress.

Vincent et al.[17] solves the problem of extremely large targets by making the complexity of several key computation steps independent of output size. Whereas other approaches to the large output problem involve approximations of the gradient using tree-based heuristics or sampling, they show it is possible to make the complexity of gradient calculations and weights updates depend solely on the dimension of the last hidden layer.

2.1 Efficient Exact Gradient Update Method

Vincent et al.[17] note that for the traditional softmax activation, a prohibitive $O(dD)$ computation is required at three separate steps of the training process:

- The forward propagation pass to calculate the normalized non-sparse output
- The backpropagation of the gradient with respect to the last hidden layer of the network
- The update to weight matrix W of the output layer

They then offer a novel approach this problem by enabling $O(d^2)$ computation of the exact gradient calculation and weight updates. One caveat however, is that this efficient exact gradient method is only possible for the spherical family of loss functions, to which the squared error and spherical softmax belong. In order to to achieve $O(d^2)$ time for all three bottlenecks listed above, their method requires two additional considerations in calculating the sparse output, corresponding loss, gradient, and weight updates:

- Maintaining an updated matrix $Q = W^{\mathrm{T}}W$, where W is the output weight matrix
- Implicitly representing matrix W with factors V and U

Vincent et al.[17] show that the former allows for the calculation of the squared error loss and the gradient of the last hidden layer in $O(d^2)$ time, rewriting the loss and gradient calculation to utilize Q and separating the single $O(dD)$ terms into several $O(d^2)$ and $O(Kd)$ parts, where K is the number of non-sparse elements in the output. The implicit representation of W in the second requirement enables $O(d^2)$ time implicit updates of the output layer weight matrix, as the updates to the U and V can be achieved in $O(d^2)$ time [17], whereas an explicit update to W incurs the full $O(dD)$ penalty.

2.2 Implications

What is remarkable about the work of Vincent et al.[17] is that the gradient calculations of a network become independent of the vocabulary size D, only depending on the number of nodes d in the last hidden layer, where D may be much greater than d. Theoretically, this allows for gradient calculations hundreds to thousands of times faster than conventional methods. Vincent et al.[17] note that their efficient algorithm is able to converge at nearly the same speed as hierarchical softmax training word embeddings on the One Billion Word dataset [4], although at lower final scores. Furthermore, as opposed to other methods such as NCE and hierarchical softmax, which only approximate the gradient update, they show it is feasible to calculate the exact gradient update independent of vocabulary size and possibly speed up convergence.

2.3 Continuations on the Paper

2.3.1 "An exploration of softmax alternatives belonging to the spherical loss family."

de Brébisson et al.[5] investigate the properties and performance of the efficient exact gradient method proposed by Vincent et al.[17] using two variants of spherical-softmax on a variety of tasks. They first compare the effectiveness of the spherical family of loss functions against the softmax activation on the MNIST, CIFAR10, and CIFAR100 image recognition datasets, finding that for the first two their proposed log-Taylor softmax loss outperformed the traditional softmax. de Brébisson et al.[5] later find however, that the softmax loss function performed best on CIFAR100 and Penntree [9], where the output is of a much higher dimensionality, and that there was no difference in convergence speed between competing models.

2.3.2 "The Z-loss: a shift and scale invariant classification loss belonging to the spherical family."

de Brébisson et al.[6] expand on the previous study by introducing a new loss function called Z-loss, belonging to the spherical family with scale and shift invariance properties and subject to the efficient exact gradient method proposed by Vincent et al.[17] They show that on language tasks such as the Penntree and One Billion Words datasets, their proposed Z-loss is able to beat log-softmax loss in Top-k measures in Penntree and is competitive with hierarchical softmax on One Billion Words (vocab size $\tilde{8}$00k) with a 4 times faster convergence rate. It is also shown that Z-normalization can be applied before traditional loss function to improve performance on Top-k tasks.[6]

2.4 Shortcomings of Spherical Loss Functions

While de Brébisson et al.[6] show that the spherical family of loss functions, specifically the proposed Z-loss, can perform comparatively to hierarchical softmax, the accuracy of resulting models is lower, sometimes even when the architecture of the Z-loss model is more complex. Also, while de Brébisson et al.[5] note that their proposed spherical softmax variant performs better than traditional softmax on low-dimensional output tasks such as MNIST (where $D = 10$), this is undercut by the fact that spherical softmax functions have a higher theoretical complexity, $O(d^2)$, in situations where the dimension of the output is higher than that of the last hidden layer, whereas traditional softmax only incurs an $O(dD)$ penalty. The implication of this is that the efficient exact gradient method only improves performance when it is less efficient than traditional methods.

This leaves improved convergence rates with lower performance on specific tasks over hierarchical softmax as the sole advantage of the efficient exact gradient method. However, there exists a wide body of literature on improving the convergence rates of traditional softmax and hierarchical softmax models that were not investigated by Vincent et al.[17]. Furthermore, it is unclear if improved convergence of spherical losses carries over to more complex tasks, such as translation.

3 Alternative Approaches to Large Output

3.1 Literature Review

While Vincent et al.[17] do offer a novel solution to extremely large targets, it is important to note the trade-off of speed and performance of their method. There is a wide body of literature on alternative methods for modeling large outputs, and of which we contend can offer a better solution without the performance loss associated with spherical loss functions:

3.1.1 Hierarchical Softmax Loss

Morin et al.[12] originally proposed a tree-based heuristic as an alternative to the traditional softmax activation. Hierarchical softmax models each word as a leaf node, while terminal nodes represent classes which are usually determined using a clustering algorithm, and the probability of a word is the product of all probabilities on the path to the leaf. Chen et al.[3] showed that K-means clusters of word embeddings yields higher performance than randomly assigned clusters. Hierarchical softmax also scales much more efficiently with vocab size than traditional softmax, as in the case of a perfectly balanced tree hierarchical softmax achieves $O(logD)$ time.[3]

3.1.2 Noise Contrastive Estimation and Negative Sampling

Gutmann et al.[7] introduced Noise Contrastive Estimation (NCE) as a method for address the scaling problem of language models with large outputs. The principle of NCE is that a well-trained model can differentiate true data from noise with logistic regression effectively. Mikolov et al.[11] state that it can be shown this practice approximates the log-likelihood property of the traditional softmax function, at much faster speeds. They introduce a simpler variation of this method called Negative sampling which draws from a distribution of random noise k number of times, but lacks the property of approximating log-likelihood. They also show that for analogical reasoning tasks their Negative sampling approach can be competitive with hierarchical softmax.

3.1.3 Subsampling of Frequent Words

Mikolov et al.[11] also introduce a novel approach to the rare-word problem by subsampling of infrequent words in the corpus. They posit that rare words carry more meaning than extremely common ones, and that subsampling infrequent words increases training speed. For example, proper nouns are more important in a sentence than the word "the", but are much less frequent. The probability that a frequent word will be discarded via subsampling is shown in [11], where t is the sampling threshold:

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}, f(w_i) = freq_i$$

The study also found that the use of subsampling infrequent words leads to higher accuracy in representations of rare words after training on a phrase analogy task, especially for large vocabulary tasks. We propose this could be further extended to translation tasks by considering the average or minimum frequency value of a given sentence.

3.1.4 Infrequent Normalization

Andreas and Klein [1] proposed a technique called infrequent normalization to bypass the complexity constraints of the softmax function. In order to reduce algorithm complexity, they modify the output normalization process, leaving the outputs unnormalized, but introducing a modified loss function that includes a penalty that encourages normalized predictions. The objective of this loss function is to generate outputs that are self normalizing, so as to remove the costly $O(dD)$ operation associated with normalizing with a softmax activation. However, in their experiments on various language modeling task, Chen et al.[3] showed that self normalizing approaches did not perform as well as alternative approaches such as hierarchical softmax and differentiated softmax.

3.2 A Different Approach to the Large Output Problem: LMVR

3.2.1 The Rare-Word Problem

It is important to note the relationship between the large-output problem in general and the rare-word problem experienced in many language modeling tasks. The relationship between the frequency of individual words, in English specifically, and their rank in usage (the most used word having a rank of one) is an inverse power law [8], where r is the rank of a word:

$$P(r) = \frac{C}{r^\alpha}, C \approx 1, \alpha \approx 0.1$$

The implications of this law are obvious, a few hundred words can account for most of the words present in a given corpus. This gives rise to the rare-word problem, which describes how the majority of distinct words in a corpus lead to difficulty training due to few examples of usage and a linear increase in output size.

3.2.2 Linguistically-Motivated Vocabulary Reduction

All of the methods discussed previously have dealt with the large output problem in neural networks by attempting to scale current algorithms and their complexities accordingly. However, a novel approach to this problem is presented by Ataman et al.[2], who propose a method called linguistically-motivated vocabulary reduction. Rather than trying to evaluate large outputs more quickly, they acknowledge that many of the words used in a corpus are rare and often different morphological forms of a root word. LMVR is a preprocessing step for the commonly-used NMT architecture [15], where words are segmented into component morphemes and thereby reduce the number of distinct tokens. They show that for translation tasks from English to Turkish (a highly inflected language with many rare words), it is possible to drastically reduce the output size of a model by 4.25 times (170k to 40k) and even increase BLEU scores significantly. For translation tasks, this is a novel approach to addressing the large output problem by simply reducing the dimensionality of the target itself.

4 Proposal for Future Study

In order to address the real-world performance of the efficient exact gradient method with spherical losses proposed by Vincent et al.[17], it is important to apply it on more complex tasks and compare it to optimized alternatives. We propose a study on the performance of different large output methods a modern language processing task: translation. Specifically, this study would investigate softmax variants on English to Russian, where the target language is highly inflected similar to the study of Ataman et al.[2]. Such a setting would allow for effective use of linguistically-motivated vocabulary reduction. On a high level, this study would involve training several modern NMT architectures with a variety of the methods addressing large output discussed thus far, with the key metrics of interest being minibatch processing speeds, convergence rates in epochs and time, overall accuracy, BLEU scores, and inference speeds of trained models, which can differ between activation functions as discussed by Martins et al.[10]

4.1 Methods

In order to demonstrate that conventional softmax variants can match the convergence properties of efficient exact gradient method, we provide a detailed approach based on existing literature. We also outline the computational and architectural settings in which our study would take place.

4.1.1 Improving the Convergence Rates of Softmax Variants

There is a considerable body of evidence indicating that it is possible to increase the convergence rates of models using softmax or hierarchical softmax activations without the loss in performance associated with spherical losses. Mikolov et al.[11] showed that the use of subsampling infrequent words as discussed above reduced the training time of models using hierarchical softmax activations by 50% and even improved accuracy by 8% on analogical reasoning tasks.

Furthermore, Chen et al.[3] note the importance of proper initialization of the terminal nodes in a hierarchical softmax tree, finding that using weighted K-means rather than random initialization yielded significantly higher performance (13%) in accuracy on the Billion Words dataset, and later imply an increase in convergence. Vincent et al.[17] do not discuss the type of initialization used in the hierarchical softmax they compared against their spherical loss methods, so it is difficult to say if their study is an adequate comparison between the two methods.

It is also common practice to use pretrained word embeddings on certain tasks such as translation and document classification [14], as the model does not need to learn them from scratch, which is one of the most expensive parts of model training. Chen et al.[3] demonstrated that the initialization of the input and output layers with pretrained embeddings yielded a significant increase in convergence rates and performance. They also show that the freezing of input and output layer weights has a similar effect. Vincent et al.[17] demonstrate that there are three $O(dD)$ steps in the output layer for softmax loss, but freezing the output weight matrix W eliminates two of those costly steps (the gradient calculation and weight update), leading to a theoretical 67% increase in minibatch processing speed.

It is important to note that all of these methods can be applied simultaneously to improve traditional and hierarchical softmax performance. Chen et al.[3] find that for language tasks with large vocabularies, hierarchical softmax typically performs the best, and when coupled with subsampling, proper structure and initialization of hierarchical softmax trees, the usage of pretrained embeddings, and freezing input and output embedding weights, we argue one can achieve drastically higher accuracy and convergence rates. We propose the combined usage of these large-output oriented approaches be called *optimized hierarchical softmax*.

4.1.2 Proposed NMT Architecture

To examine the optimized hierarchical softmax and efficient gradient method in a relevant setting, we propose two well-documented NMT architectures: the traditional seq2seq model[15], and the recent Transformer architecture proposed by Vaswani et al.[17]. The latter replaces recurrent layers with a multi-head attention mechanism and is therefore significantly faster to train. By training on radically different architectures we can also observe the behaviour of the proposed loss functions in a variety of settings.

4.1.3 Computational Environment

The proposed study would be conducted in Python, using the Tensorflow library to implement the NMT models. Furthermore, we acknowledge two relevant settings: state-of-the-art performance and practical usage. To analyze both settings, we would first conduct the study on a high-performance GPU cluster to show that our implementation of optimized hierarchical softmax can achieve state-of-the-art BLEU scores on English-to-Russian translation. Later, we would train our models on a single GPU to show that our optimized hierarchical softmax is a viable alternative to efficient exact gradient method on a typical desktop setup in terms of train-time and accuracy.

4.2 Objectives of the Study

The proposed English-to-Russian translation task in different architecture and computational settings would aim to show that an optimized hierarchical softmax approach using the methods listed previously can achieve state-of-the-art results, improving both the convergence and overall performance of NMT models over the efficient exact gradient method. Furthermore, we would aim to show that the large output and rare-word problems that currently plague language modeling can be mitigated in general by using a combination of subsampling infrequent words [11] and vocabulary reduction with LMVR [2]. Lastly, we wish to examine the behaviour of different NMT architectures, optimization algorithms, and training methods for translation tasks with optimized hierarchical softmax.

4.3 Dissemination of Results

Given the objective of achieving both state-of-the-art performance and a practical implementation, our proposed study would be presented in the format of a research paper to be submitted to an appropriate conference such as NeurIPS. We would also provide an in-depth explanation and tutorial of our method on a public platform such as GitHub to allow easy adoption of the method.

5 Conclusions

While Vincent et al.[17] introduce an original and novel approach to efficiently calculating the exact gradient update of a model for large output spaces, the evidence for the effectiveness of spherical loss functions on modern NLP tasks is mixed. There are a variety of methods that have been shown to increase the performance of tradition loss functions, such as the hierarchical softmax, in ways that do not suffer the speed-performance trade-off incurred by methods such as Z-loss. It is further shown that alternative approaches to the large output problem itself can be solved via vocabulary reduction methods such as LMVR. In light of this, we proposed a study on the effectiveness of different methods addressing the large output and rare-word problems on modern NLP tasks such as translation, with a contention that an optimized hierarchical softmax architecture can outperform the efficient exact gradient approach.

References

[1] Andreas, Jacob, and Dan Klein. "When and why are log-linear models self-normalizing?." *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* 2015.

[2] Ataman, Duygu, et al. "Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English." *The Prague Bulletin of Mathematical Linguistics* 108.1 (2017): 331-342.

[3] Chen, Welin, David Grangier, and Michael Auli. "Strategies for training large vocabulary neural language models." *arXiv preprint arXiv*:1512.04906 (2015).

[4] Chelba, Ciprian, et al. "One billion word benchmark for measuring progress in statistical language modeling." *arXiv preprint arXiv*:1312.3005 (2013).

[5] de Brébisson, Alexandre, and Pascal Vincent. "An exploration of softmax alternatives belonging to the spherical loss family." *arXiv preprint arXiv*:1511.05042 (2015).

[6] de Brébisson, Alexandre, and Pascal Vincent. "The Z-loss: a shift and scale invariant classification loss belonging to the spherical family." *arXiv preprint arXiv*:1604.08859 (2016).

[7] Gutmann, Michael, and Aapo Hyvärinen. "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models." *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010.

[8] Li, Wentian. "Random texts exhibit Zipf's-law-like word frequency distribution." *IEEE Transactions on information theory* 38.6 (1992): 1842-1845.

[9] Marcus, Mitchell, Beatrice Santorini, and Mary Marcinkiewicz. "Building a large annotated corpus of English: the Penn Tree bank" in the distributed Penn Tree Bank Project CD-ROM." *Linguistic Data Consortium, University of Pennsylvania* (2006).

[10] Martins, Andre, and Ramon Astudillo. "From softmax to sparsemax: A sparse model of attention and multi-label classification." *International Conference on Machine Learning*. 2016.

[11] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.

[12] Morin, Frederic, and Yoshua Bengio. "Hierarchical probabilistic neural network language model." *Aistats*. Vol. 5. 2005.

[13] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

[14] Peters, Matthew E., et al. "Deep contextualized word representations." *arXiv preprint arXiv*:1802.05365 (2018).

[15] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems*. 2014.

[16] Vaswani, Ashish, et al. "Attention is all you need." *Advances in Neural Information Processing Systems*. 2017.

[17] Vincent, Pascal, Alexandre De Brébisson, and Xavier Bouthillier. "Efficient exact gradient update for training deep networks with very large sparse targets." *Advances in Neural Information Processing Systems*. 2015.