Pruning Attention in Transformers for Nonlinear Channel Compensation in Optical Systems

Behnam Behinaein Hamgini, Hossein Najafi*, Ali Bakhshali, and Zhuhong Zhang
Ottawa Research Centre, Huawei Technologies Canada, 303 Terry Fox Drive, K2K3J1, Kanata, ON, Canada
**hossein.najafi@huawei.com*

Abstract: We study pruning attention in Transformers for optical nonlinear channel compensation. We show the impact of statistical pruning on the performance and complexity of nonlinear equalization and compare it with a physics-informed pruning scheme.

1. Introduction

Various techniques from deep neural networks have been employed for nonlinear compensation (NLC) in optical transmission systems. Specifically, feed-forward perturbation-based models and recurrent neural networks (RNNs) have been proposed to learn and compensate fiber nonlinearity and the associated memory from accumulated dispersion in coherent systems [1–3]. However, the high computational complexity of generating perturbation triplets and large latency due to serial structure and folding memory in RNNs introduce significant challenges for hardware implementation in long-haul applications. Recently, Transformer-NLCs have been proposed for channel nonlinear compensation where parallelizable structure in both training and inference stages makes them a great candidate for applications in high throughput long-haul optical transceivers [4].

Attention mechanism is at core of a Transformer and its computational complexity grows proportional to the square of input sequence length. This could be prohibitive in high symbol rate applications that require a high degree of parallelization. One universal approach to reduce the complexity of a large neural network is to prune the model by systematically removing the least significant weights [5,6]. Specifically, several attempts have been made to reduce the complexity of self-attention in Transformers by increasing the sparsity of the attention matrix [4,7]. For Transformer-NLC, a physic-informed (PI) mask based on perturbation theory was introduced in [4] in order to reduce the complexity of the attention mechanism with minimal impact on the performance. In this paper, we explore the use of statistical pruning in Transformer-NLCs where the attention mechanism is pruned with different number of parameters. Also, the results are compared with the PI mask.

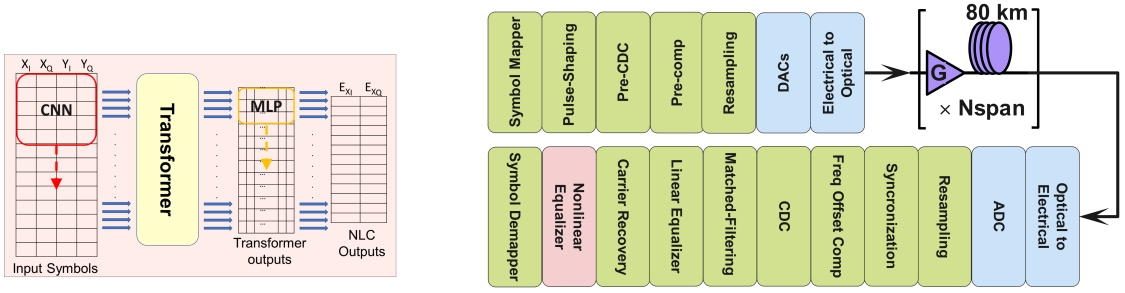
2. Pruning Attention

Here, Transformer encoder structure is used for nonlinear equalization. It consists of a positional encoder, several layers of multi-head self-attention, feed-forward networks, and add-normalization layers. Transformer-NLC structure is depicted in Fig. 1(a). The input convolutional neural network (CNN) acts as an embedding generator module and creates blocks of input symbols with more suitable representations for the Transformer. Due to the memory of nonlinear channel, we also consider t symbols before and after each input block, where t is denoted as tap size. Finally, a multi-layer perceptron (MLP) is added to generate the equalizer's outputs.

Self-attention mechanism is the core of a Transformer. At first, the input sequences are transformed by linear layers into queries (Q), keys (K), and values (V) where $Q,K,V \in R^{N\times d_K}$, N is the input sequence length, and d_K is the size of representations in Q, K, and V. Then, attention is given by

$$A = \text{softmax}(\frac{QK^T}{\sqrt{d_K}}), \qquad \text{Attention}(Q,K,V) = AV. \tag{1}$$

In general, pruning is a systematic way of reducing the number of parameters in a neural network, for which, several criteria have been proposed [6]. Here, we use the magnitudes of elements after softmax (in Eq. 1) to remove a certain portion of elements with the least magnitude scores. To implement the pruning process, at first, a mask (a matrix with the same size of the attention weight matrix A) is created. Next, the elements of this mask corresponding to the elements of A that must be pruned are filled with negative infinity and the elements that should be kept are filled with zeros. Finally, this mask is added to $\frac{QK^T}{\sqrt{d_K}}$ in Eq. 1. Note that since we have multiple layers and heads in the Transformer architecture, the pruning masks can be generated at different granularity. We explore three scenarios for pruning: Scenario I) the same mask for all layers and heads, Scenario II) a separate mask for each layer used by all heads within that layer, and Scenario III) a separate mask for each head in each



(a) (b)

Fig. 1: (a) Nonlinear Equalizer: Transformer-NLC, (b) System model.

layer. To compute the magnitude scores in order to mark the elements of A for pruning, we run the models with several batches of training data and record the elements of A as a_{lbhij}, where l, b, h, i, and j represent the layer, batch, head, row, and column indices of the recorded values, respectively. The pruning scores are computed for each scenario as follows:

$$s^{I}_{ij} = \frac{1}{LBH}\sum_{l=1}^{L}\sum_{b=1}^{B}\sum_{h=1}^{H}|a_{lbhij}|, \qquad s^{II}_{l,ij} = \frac{1}{BH}\sum_{b=1}^{B}\sum_{h=1}^{H}|a_{lbhij}|, \qquad s^{III}_{lh,ij} = \frac{1}{B}\sum_{b=1}^{B}|a_{lbhij}|, \tag{2}$$

where s^{I}_{ij}, $s^{II}_{l,ij}$, and $s^{III}_{lh,ij}$ are scores for Scenario I to III, respectively, L is the number of layers, B is the batch size times the number of used batches, and H is the number of heads. For each scenario, first, a mask filled with zero elements is created, next a certain percentage of the smallest magnitude scores are selected, and finally the elements in the mask corresponding to the selected elements are set to negative infinity.

Pruning can be performed pre-train or post-train. In the post-train approach, pruned models are retrained (fine-tuned) to improve the performance [5]. In addition, pruning can be carried out in one or several steps. Here, we explore post-train pruning with fine-tuning and rewinding [8] in one step and also in several steps using the method introduced in [9]. Moreover, for PI masks, we explore pre-train and post-train with fine-tuning and rewinding.

3. Setup and results

The system setup for simulation results is given in Fig. 1(b). We use 16QAM modulation with 32Gbaud and the link consists of 40 spans of standard single-mode fiber (SSMF). Training and evaluation of models are all performed at 2dBm launch power. We explore four Transformer models with the hyper-parameters listed in Table 1, all with the block size of 128. Transformer parameters are defined similar to [4]. To investigate the statistical pruning, ratios of 50, 60, 70, 80, and 90 percent in 1, 3, 5, 7, and 9 steps are explored. For PI masks, models are run with a ρ of 2.6. It should be noted that the sparsity of PI masks only depends on ρ, tab size, and block size.

The Q-factor (dB) for the four selected models with different pruning scenarios are shown in Fig. 2 where the best results across all selected pruning steps are reported. Without any nonlinear compensation (before applying Transformer-NLC), Q value is 6.68 dB. As seen here, in Model1 (highest complexity), all scenarios perform reasonably well until 80% pruning, and Scenario III performs well even in 90% pruning. In Model2, Scenario III still performs well until 80% pruning while the performance of other scenarios degrades after 50% of pruning. In Model3, as we increase pruning, the performance decreases linearly. In Model4, it seems that the model has not enough capacity to use all the information from the attention, and pruning has no effects on the performance.

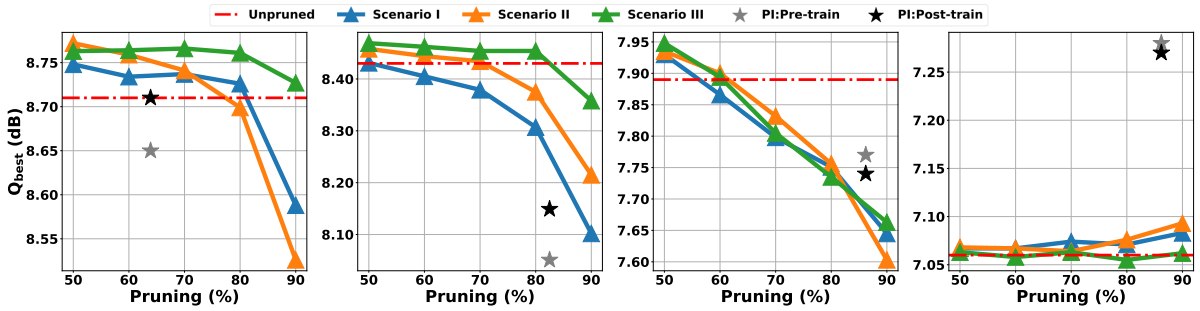
Table 2 summarizes the performances of PI pre-train/post-train pruning of the four models as well as the best results among all the selected pruning steps for Scenarios I to III with the same amount of pruning as PI masks. In Model1, PI performs close to Scenarios I-III. In Model2 at 82.5% pruning, the model with PI post-train method performs almost the same as Scenarios I-II, although it cannot beat Scenario III. Generally, as we decrease the model complexities, the PI schemes catch up with Scenarios I-III and even in Model4 at 86.2% pruning, PI schemes show a better performance compared to the unpruned model.

Table 1: Hyper-parameters for Transformer Models.

Hyper-parameter	Model1	Model2	Model3	Model4
tap size	96	16	12	12
hidden size	96	64	24	12
key size	64	64	24	12
number of heads	4	4	4	4
number of encoder layers	3	3	2	1
FFN hidden size	64	32	32	32
window size	15	15	7	7

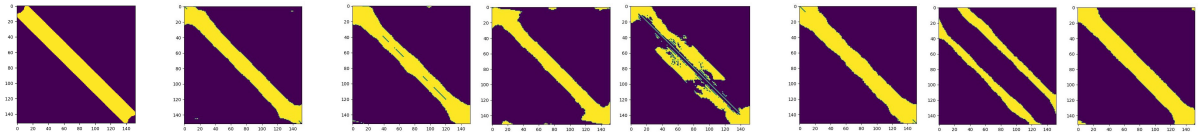
Table 2: Q(dB) for Different Pruning Approaches.

	Model1	Model2	Model3	Model4
Pruning%	63.9	82.5	86.2	86.2
No Pruning	8.71	8.43	7.89	7.06
Scenario I	8.74	8.26	7.73	7.07
Scenario II	8.76	8.34	7.71	7.07
Scenario III	8.77	8.43	7.71	7.05
PI Pre-train	8.65	8.05	7.77	7.28
PI Post-train	8.71	8.15	7.74	7.27



(a) (b) (c) (d)

Fig. 2: Performance of the four models for the three scenarios in different pruning ratios. Figures (a), (b), (c), and (d) shows the performance vs pruning ratios for Model 1, 2, 3, and 4, respectively.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 3: Pruning masks for Model2: (a) PI, (b) Scenario I, (c)-(e) Scenario II, masks for layers 1, 2, and 3 respectively, and (f)-(h) Scenario III for head number 4, masks for layers 1, 2, and 3, respectively. Yellow and purple areas show where mask elements are zero or negative infinity, respectively.

As observed so far, at higher model complexities, post-train PI has better performance compared with pre-train PI method. It is also interesting to note that PI with post-train can match or beat Scenario I which shows that PI pruning outperforms the statistical one with the same degree of freedom in our models while only needed to be retrained once. However, one restriction in PI approaches is that the pruning amount cannot be varied widely for a fixed tab size and block size. In summary, if we have enough model parameters, a fully flexible pruning scheme in scenario III can provide performance and complexity advantages. However, as we reduce the size of the initial model, the PI mask is a better choice compared to the selected scenarios of statistical pruning.

Finally, to picture the pruning masks, we use an example in Fig. 3 where the masks for Scenarios I-III and PI with 82.5% pruning are depicted. PI and Scenario I masks are almost similar (the same mask for all layers and heads). These results show the compatibility of statistical pruning with the physics of the problem where we expect the adjacent representations of the symbols to be more important for computing the nonlinear distortion of each symbol. However, as we relax the requirement of having the same mask for all the layers and heads, new patterns emerge, although the masks for the first layer remain diagonal. We hypothesize that as we go deeper in the Transformer layers, since pruning process has more freedom, it chooses some of the off-diagonal elements (especially for Scenario III), which enables the Transformer to use higher relationships among internal representations.

4. Conclusion

We presented three pruning approaches for Transformer-NLCs and compared them with a PI mask. We showed that in larger models, statistical pruning with degree of freedom maintains the model performance even when 90% of the attention matrix coefficients are removed. However, at lower complexities, PI masks have advantages.

References

1. S. Zhang, F. Yaman, K. Nakamura, T. Inoue, V. Kamalov, L. Jovanovski, V. Vusirikala, E. Mateo, Y. Inada, and T. Wang, "Field and lab experimental demonstration of nonlinear impairment compensation using neural networks," Nat. Commun. **10**, 1–8 (2019).
2. P. J. Freire, Y. Osadchuk, B. Spinnler, A. Napoli, W. Schairer, N. Costa, J. E. Prilepsky, and S. K. Turitsyn, "Performance versus complexity study of neural network equalizers in coherent optical systems," J. Light. Technol. **39**, 6085–6096 (2021).
3. A. Bakhshali, H. Najafi, B. B. Hamgini, and Z. Zhang, "Neural network architectures for optical channel nonlinear compensation in digital subcarrier multiplexing systems," Opt. Express **31**, 26418–26434 (2023).
4. B. B. Hamgini, H. Najafi, A. Bakhshali, and Z. Zhang, "Application of transformers for nonlinear channel compensation in optical systems," arXiv preprint arXiv:2304.13119 (2023).
5. D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" Proc. Mach. Learn. Syst. **2**, 129–146 (2020).
6. S. Vadera and S. Ameen, "Methods for pruning deep neural networks," IEEE Access **10**, 63280–63300 (2022).
7. T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of Transformers," AI Open (2022).
8. A. Renda, J. Frankle, and M. Carbin, "Comparing rewinding and fine-tuning in neural network pruning," arXiv (2020).
9. M. Zhu and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression," arXiv (2017).