Opportunities for Post-Training Dynamic Layer Sparsity in Large Vision and Language Models

Jordan Dotzel Carly Jiang Mohamed Abdelfattah Zhiru Zhang
Cornell University
{dotzel, cjj43, mohamed, zhiruz}@cornell.edu

Abstract

Large language and vision models have recently achieved state-of-the-art performance across various tasks, yet due to their large computational requirements, they struggle with strict memory, latency, and power demands. To meet these demands, various forms of dynamic sparsity have been proposed that reduce compute on an input-by-input basis. These methods improve over static methods by exploiting the variance across individual inputs, which has steadily grown with the exponential increase in training data. This dynamic sparsity has been explored within the hidden dimension and attention heads. Yet, the increasing depth within modern models, currently with hundreds of layers, has opened opportunities for dynamic layer sparsity, which skips the computation for entire layers. In this work, we explore the practicality of layer sparsity within pre-trained models by profiling residual connections and establish the relationship between model depth and layer sparsity. For example, the residual blocks in the OPT-66B model have a median contribution of 5% to its output, and ViT-Huge has approximately a 7% contribution. We also find these contributions decrease linearly with model size, implying that state-of-the-art models have near a 1% median contribution on each layer, which creates significant opportunities for dynamic layer sparsity. We then insert oracles at each layer and threshold on these residual contributions to find that these models can support significant dynamic sparsity, with median dynamic depth close to 75% of their original depth.

1. Introduction

Large language and vision models have recently achieved state-of-the-art performance across various tasks, yet due to their large computational requirements, they struggle with strict memory, latency, and power demands. As these transformers grow larger, they create opportunities for dynamic layer sparsity, which can skip individual layers on an input-by-input basis, as shown in Figure 1. For instance, our residual block profiling in Section 4 suggests that modern state-of-the-art transformers likely have a median contribution around 1% to the output at each block, and that these contributions are *dynamic*, varying token by token. This type of sparsity was impractical at smaller scales and with previous neural architectures. At smaller scales, every layer contributes significantly to the computation for each input, and with previous architectures, e.g., convolutional neural networks (CNNs), models change their intermediate dimensions throughout their depth, making layer skipping impractical.

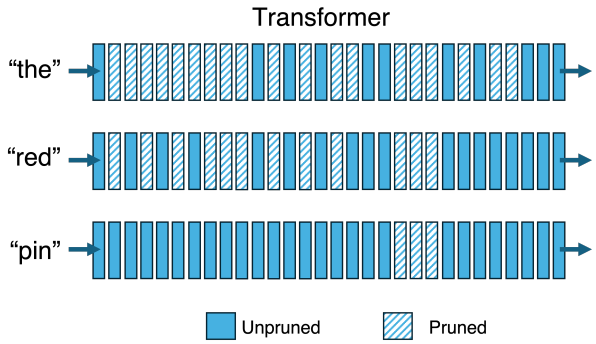

Figure 1. **Dynamic Layer Sparsity –** As transformers grow larger, each layer contributes less to the output and shows significant variation on a token by token basis. Dynamically pruning these layers allows for models to grow significantly without corresponding increases in model latency. Early profiling results suggest that individual layers contribute around 1% within modern state-of-the-art language models.

This work shows that the layer contributions vary among models and tasks, and often the earlier layers of the network contribute more than the later layers. This indicates that early-exit methods, which dynamically prune the later layers in the network, often focus on the wrong set of layers. This dynamic contribution can be exploited at the token-level if it can be predicted accurately and efficiently at runtime. This work explores the opportunities for dynamic sparsity within modern transformers by focusing on

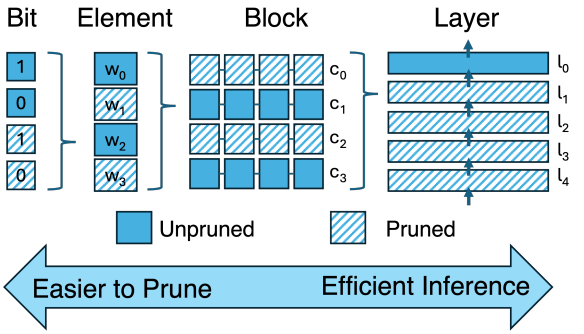

Figure 2. **Sparsity Granularity –** Bits form the basis for elements (weight or activations), which create blocks (rows, columns, heads), which then form individual layers. This leads to a sparsity spectrum where the smaller units are easier to prune without accuracy loss yet more difficult to accelerate. Layer sparsity has the highest potential for inference speedup and largest support within current hardware.

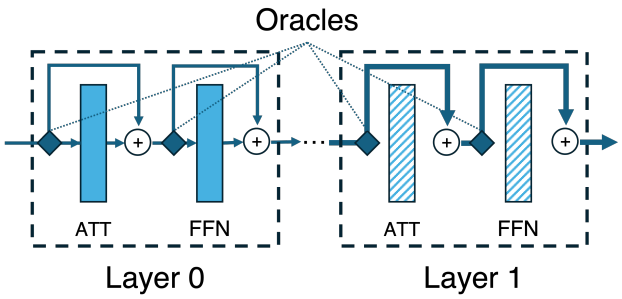

Figure 3. **Residual Blocks –** There are two types of residual blocks within transformers, attention (ATT) and feed-forward network (FFN). These blocks offer natural points to profile layer strength since block inputs and outputs are combined at a single point. To establish an upper bound on the effectiveness of dynamic layer sparsity, oracles are inserted before each block that know the layer contribution beforehand.

the OPT family of models [9] for language and ViT models [3] for vision. It profiles the residual blocks to quantify the importance of each intermediate layer to its output and then highlights trends across model size and block types. Then, it inserts *oracles* at every layer to calculate various accuracy proxies and simulate greedy decisions on which layers to dynamically skip per token.

2. Related Work

Sparsity research with deep neural networks has a long history, and broadly can be categorized in terms of granularity, structure, and mode (static vs. dynamic) [4]. Figure 2 shows sparsity granularity, beginning with bits that construct parameter elements, elements that build blocks, and blocks that form layers. As the unit becomes larger, it becomes more difficult to arbitrarily prune without accuracy loss yet easier to accelerate with modern hardware. For instance, unstructured element sparsity in weights leads to high compression levels while maintaining model accuracy, yet it requires specializing sparse accelerators to translate compression into end-to-end speedup.

In addition, the sparsity mode can either be static or dynamic. Static sparsity leads to more regular patterns that can be optimized by compilers and simpler architectures that do not need additional sparsity predictors. In contrast, dynamic sparsity can take advantage of input-dependent characteristics to increase model accuracy at higher levels of compression. This work focuses on dynamic layer sparsity, which can take advantage of the recent explosion in model depth within language and vision models.

2.1. Dynamic Sparsity

Multiple prior works have proposed dynamic sparsity to accelerate DNNs across granularities. For example, Channel Gating introduced a method for dynamic channel sparsity that reduced the compute of CNN workloads by up to $8\times$ without significant accuracy loss [5]. Precision Gating continued this line of research by applying dynamic sparsity at the bit level to reduce the required compute by up to $3\times$ [10]. Later, DejaVu applied a similar approach within LLMs to induce dynamic sparsity on the channels within the FFN layer and across the heads of the attention layer [6].

2.2. Early Exit

In addition to dynamic sparsity along the network width, multiple prior works have explored sparsity in the depth dimension. For instance, early-exit DNNs use dynamic sparsity along the depth dimension by allowing the computation to exit prematurely at fixed points within the network [1, 7]. This process must be trained end-to-end using a joint loss function that weights the contributions from each early-exit layer. However, this work shows that in many models, the earlier layers in the model often contribute more, and therefore early-exits are significantly more difficult to apply post-training.

3. Layer Sparsity

Transformer layers contain two residual blocks: attention (ATT) and feed-forward network (FFN) [8]. These blocks each contain the main residual branch $R(x)$, which comprises multiple individual layers, and the identity branch x, which bypasses the residual branch and simply returns its input. They combine these branch outputs together to compute $R(x) + x$, so that during training the main branch only has to learn the function residual $R(x) - x$.

These blocks offer natural breakpoints within the model to profile and induce layer sparsity since they already provide skip-connections that have been trained along with the model. Figure 3 shows a lower-level view of these blocks

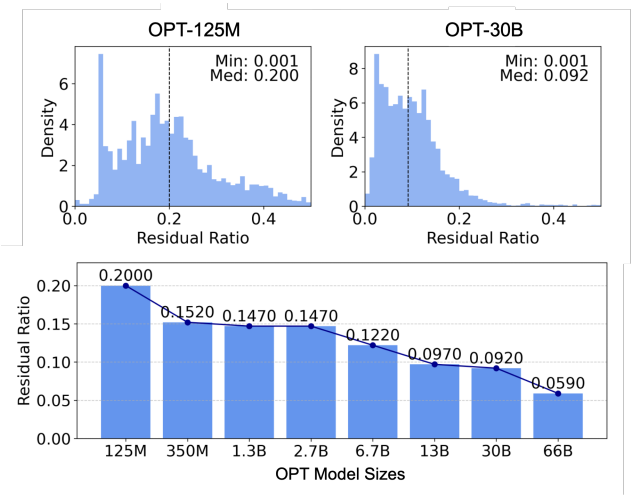

Figure 4. **Residual Ratio –** As models grow larger, the residual ratio decreases for each layer, and therefore more layers contribute less to the overall output. Each data point represents a residual ratio for a single token for both attention and feed-forward blocks.

within two transformer layers. It shows that the main branch and skip connections are combined at an addition node before they are passed to the next block. This structure enables easy profiling of the blocks by measuring the relative magnitudes into these additions.

This figure also shows the insertion of oracles that can switch on and off the main branch using various accuracy proxies, such as the residual ratio as defined in Section 4.1. When they are switched on, the block operates normally by combining the skip and residual branches, and when switched off, only the skip connection is active. This work focuses on the opportunities for dynamic layer sparsity and simulates layer skipping by allowing these oracles to have access to future information.

4. Profiling

The primary proxy used by these oracles is the *residual ratio*, which captures the relative importance of the main and skip branches. This section uses this ratio to analyze the layer sparsity within OPT and ViT models with examples taken from WikiText-2 and COCO. The WikiText-2 examples are packed together to avoid the use of padding to simulate batch-size one inference. This batch-size one setting is very common in practice and avoids many complications with dynamic layer sparsity that arise when using batches of examples.

4.1. Residual Ratio

To profile these opportunities for dynamic sparsity, this section defines the residual ratio r as:

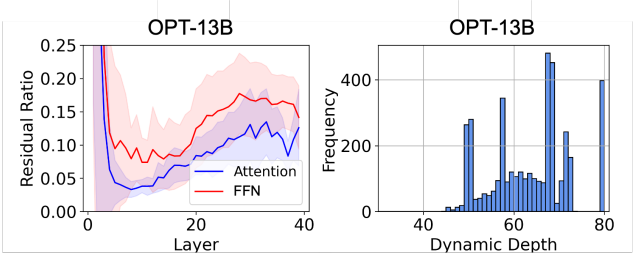

Figure 5. **Dynamic Depth –** The deeper layers in the network contribute more than the earlier layers, except for the very first layers. This relationship benefits from a routed architecture as opposed to early-exit, since early-exit skips the deeper layers. In addition, there is significant variance in the dynamic depth of the model, allowing for token-specific sparsity.

$$\mathrm{r} = \frac{\|R(x)\|_2}{\|x\|_2} \tag{1}$$

This simple quantity captures the contribution of the residual branch, and acts as an efficient post-training proxy for more expensive metrics, such as empirical layer sensitivities. For example, a block with a 2% residual ratio indicates the main branch provides a 2% average contribution at the output, although there can be large element-wise variance. Therefore, skipping blocks with ratios this small should have little overall effect on the output of the network.

4.2. Model Size

This ratio can be used to understand the relationship between model size and dynamic sparsity. Figure 4 explores this by plotting the residual ratio across OPT models for the residual attention or feed-forward block (more plots shown in Appendix A). Each data point represents a single token during the model generation phase. It shows that as the model size grows, the ratio distribution becomes more skewed to the left, indicating that opportunities for layer sparsity expand with model size. For instance, while the median residual ratio for OPT-125M is only 20%, it drops to 5.9% for OPT-66B.

In addition, the ratio seems to track the number of model parameters, not just the number of layers. For example, OPT-2.7B and OPT-6.7B have the same number of layers, differing only in their hidden dimensions, yet the ratio for OPT-6.7B continues the decreasing trend. This trend likely continues for even larger models, making dynamic layer sparsity more practical within modern state-of-the-art models with greater than one trillion parameters.

4.3. Dynamic Depth

Dynamic layer sparsity leads to dynamic depth networks that adjust their depth based on their model inputs. Figure 8 shows the residual ratio across the layers of an OPT-13B model (all models shown in Appendix A). All values

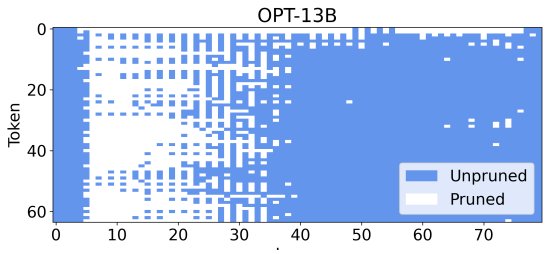

Figure 6. **OPT-13B Routing –** The majority of skipped layers are in the earlier part of the network, implying that traditional early-exit techniques may target the wrong set of layers. A layer skip is assumed when the residual ratio drops below 5%. ATT and FFN layers are interleaved in the diagram.

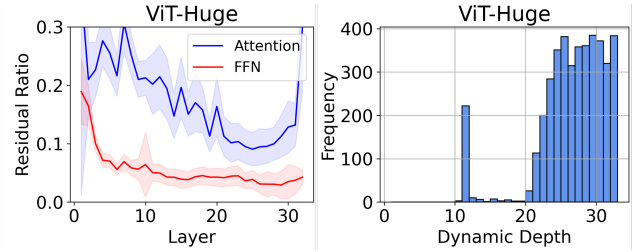

Figure 7. **Vision Profiling –** Current vision transformers are substantially smaller than language models yet demonstrate similar trends with residual ratio and dynamic depth. ViT-Huge (632M) shows median ratios comparable to OPT-350M, and the dynamic depth assumes skipping layers with lower than 10% residual ratio.

shown are mean residual ratios taken across tokens from Wikitext-2 data using a sequence length of 256. The ratio variance is highlighted in lighter colors centered around the mean. It demonstrates that the earlier residual blocks contribute more compared to the later layers, except for the first few layers. In addition, there is significant variance across tokens suggesting the opportunity to apply dynamic layer sparsity to only the tokens with lower ratios.

This figure additionally shows the dynamic depth induced by this layer sparsity. It assumes oracles that threshold the residual ratio at each block and skip the residual branch if it falls below this threshold. Since computing the ratio requires running the residual branch, this is only used for profiling and simulation purposes. Each data point represents an inference of a single token using a ratio threshold of 5%. The figure confirms a spread within the network depth, where most tokens only need between 40 and 70 blocks, instead of the full network at 80 blocks.

4.4. Routing Traces

For more detailed analysis, Figure 6 shows the routing for the OPT-13B model across a batch of WikiText-2 examples. It reveals how the lower residual ratios in Figure 8 lead to a significant number of skipped layers in the beginning of the model. This again motivates the use of dynamic layer sparsity over early-exit models, since early exit can only skip later layers, which contribute the most to the network.

4.5. Vision

This analysis so far has focused on large language models, since they are currently 10 to 100× larger than large vision models. Yet, recent vision transformers have been proposed with tens of billions of parameters [2]. These weights are not yet released, yet the trends between the smaller language and vision models can still be aligned at smaller scales to suggest the behavior of large vision models with billions or trillions of parameters.

Figure 7 shows a comparison for the largest released ViT model, which contains 632M parameters across 24 layers. It shows that vision transformers at this size have comparable residual ratios to the similarly sized OPT-350M. In addition, Appendix A lists smaller ViT versions and shows a similar trend between model size and residual ratio, suggesting that as vision transformers increase in size they will benefit from the same layer sparsity opportunities as the OPT models.

5. Conclusion

In the past, dynamic layer sparsity has not been practical due to small model sizes and incompatible neural architectures, which caused large contributions from each layer and varying internal dimensions. For these reasons, dynamic layer sparsity has only been possible with techniques like early-exit, which require expensive, specialized training. Yet, as language and vision transformers grow, each layer contributes less to output, creating opportunities for post-training dynamic layer sparsity. Following the trends in Figure 4, modern language models with over one trillion parameters likely have median residual ratios less than 1%. And in the future, as vision and multi-modal models catch up to language models, their residual ratios should follow similar scaling trends.

This work establishes the opportunities for dynamic layer sparsity, yet future work will need to measure the effects on model accuracy, the correlation of the residual ratio with end-to-end accuracy, and the ratio prediction accuracy during inference. First, this will involve establishing a strong upper bound on model accuracy using more advanced oracle methods, such as token-level layer sensitivities. Then, additional accuracy proxies should be compared with residual ratio to confirm that these track the oracle methods. Finally, small routing networks will need to be trained to replace the oracles and provide accurate and efficient sparsity predictions during inference.

References

[1] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. *ICML*, 2017. 2

[2] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, and Robert Geirhos. Scaling vision transformers to 22 billion parameters. *ICML*, 2023. 4

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2

[4] Cong Hao, Jordan Dotzel, Jinjun Xiong, Luca Benini, Zhiru Zhang, and Deming Chen. Enabling design methodologies and future trends for edge ai: Specialization and codesign. *IEEE Design & Test*, 2021. 2

[5] Weizhe Hua, Yuan Zhou, Christopher De Sa, Zhiru Zhang, and G. Edward Suh. Channel gating neural networks. *NeurIPS*, 2019. 2

[6] Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. Deja vu: Contextual sparsity for efficient llms at inference time. *ICML*, 2023. 2

[7] Surat Teerapittayanon, Bradley McDanel, and H.T. Kung. Branchynet: Fast inference via early exiting from deep neural networks. *Intl. Conf. on Pattern Recog.*, 2016. 2

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 2

[9] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. 2

[10] Yichi Zhang, Ritchie Zhao, Weizhe Hua, Nayun Xu, G. Edward Suh, and Zhiru Zhang. Precision Gating: Improving Neural Network Efficiency with Dynamic Dual-Precision Activations. *ICLR*, 2020. 2