MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

AWP: Activation-Aware Weight Pruning and Quantization with Projected Gradient Descent

Liu, Jing; Koike-Akino, Toshiaki; Wang, Ye; Mansour, Hassan; Brand, Matthew

TR2025-111 July 31, 2025

Abstract

To address the enormous size of Large Language Models (LLMs), model compression methods, such as quantization and pruning, are often deployed, especially on edge devices. In this work, we focus on layer-wise post-training quantization and pruning. Drawing connections between activation-aware weight pruning and sparse ap- proximation problems, and motivated by the success of Iterative Hard Thresholding (IHT), we propose a unified method for Activation-aware Weight pruning and quantization via Projected gradient descent (AWP). Our experiments demonstrate that AWP outperforms state-of-the-art LLM pruning and quantization methods. Theoretical convergence guarantees of the proposed method for pruning are also provided.

International Conference on Machine Learning (ICML) workshop 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

AWP: Activation-Aware Weight Pruning and Quantization with Projected Gradient Descent

Jing Liu [1] **Toshiaki Koike-Akino** [1] **Ye Wang** [1] **Hassan Mansour** [1] **Mathew Brand** [1]

Abstract

To address the enormous size of Large Language Models (LLMs), model compression methods, such as quantization and pruning, are often deployed, especially on edge devices. In this work, we focus on layer-wise post-training quantization and pruning. Drawing connections between activation-aware weight pruning and sparse approximation problems, and motivated by the success of Iterative Hard Thresholding (IHT), we propose a unified method for **A**ctivation-aware **W**eight pruning and quantization via **P**rojected gradient descent (AWP). Our experiments demonstrate that AWP outperforms state-of-the-art LLM pruning and quantization methods. Theoretical convergence guarantees of the proposed method for pruning are also provided.

1. Introduction

Large transformer-based models have demonstrated superior performance in many tasks, including natural language processing and computer vision. However, their enormous size poses significant challenges for efficient deployment especially on the edge devices. Quantization and pruning are promising approaches for model compression. For example, it was found[1] that the 4-bit quantized Llama2-13B model achieves better perplexity, smaller size, and faster inference speed than the unquantized Llama2-7B model.

We consider Post-Training Quantization (PTQ) and pruning/sparsification. This is in contrast to Quantization-Aware Training (QAT) and sparse network training of large models which are still resource intensive, and many applications only consider leveraging pre-trained foundation models instead of training a model from scratch. In the post-training compression setup (Nagel et al., 2020; Li et al., 2021; Hubara et al., 2021; Liang et al., 2021), a trained but uncompressed model is given, together with a small amount of calibration data, with the aim to produce an accurate compressed model without retraining. As the foundation models are very large, which may exceed the hardware memory limit during compression, recent works, such as Wang et al. (2020); Nagel et al. (2020); Hubara et al. (2021), further break the compression task into layer-wise sub-tasks, identifying a compressed weight approximation for each layer, given a sub-sample of the layer's inputs and outputs based on calibration data. This approach is the focus of our work.

We introduce a new compression framework based on projected gradient descent (PGD), inspired by compressive sensing and sparse approximation methods. The contributions of our paper are listed below:

- We pose the LLM compression problem as a sparse approximation problem.
- We show that PGD is a viable solution to unify pruning and quantization problems without requiring computationally-intensive operations such as SVD.
- The proposed method outperforms state-of-the-art LLM compression methods on several benchmarks.
- We provide theoretical convergence guarantees for the proposed method (details are found in Appendix).

2. Related Work and Background

For the post-training quantization and pruning, activation-aware methods, e.g., Lybrand & Saab (2021); Lin et al. (2024); Frantar et al. (2022a); Wang et al. (2024a); Frantar & Alistarh (2023); Sun et al. (2023); Frantar et al. (2022b); Zhang & Saab (2025), demonstrate superior performance as they consider the input activation statistics to guide the quantization and pruning. The traditional magnitude based weight pruning solves a sparse approximation problem, without considering input activations:

$$\min_{\mathbf{W}_{\text{sparse}} \in \mathcal{C}_{\text{sparse}}} \left[\mathcal{L}(\mathbf{W}_{\text{sparse}}) := \|\mathbf{W} - \mathbf{W}_{\text{sparse}}\|_{\text{F}}^2 \right], \quad (1)$$

where $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is the original uncompressed model weight, $\mathbf{W}_{\text{sparse}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is the pruned/sparsified weight,

[1]Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 01239, USA.. Correspondence to: Jing Liu <jiliu@merl.com>.

Accepted to the 42^{nd} International Conference on Machine Learning Workshop, Vancouver, Canada. PMLR 267, 2025.

[1]https://github.com/ggml-org/llama.cpp/pull/1684

and $\mathcal{C}_{\text{sparse}}$ is a constraint set, which could be the ratio of zero elements in $\mathbf{W}_{\text{sparse}}$ and/or its sparsity patterns.

For transformer-based models, the linear layer takes in input activations $\mathbf{X} \in \mathbb{R}^{d_{\text{in}} \times n}$, where n is a total token length given calibration data. In contrast, activation-aware pruning changes the minimization objective to

$$\mathcal{L}'(\mathbf{W}_{\text{sparse}}) := \|\mathbf{W}\mathbf{X} - \mathbf{W}_{\text{sparse}}\mathbf{X}\|_{\text{F}}^2 \quad (2)$$

$$= \|\mathbf{W}\mathbf{C}^{\frac{1}{2}} - \mathbf{W}_{\text{sparse}}\mathbf{C}^{\frac{1}{2}}\|_{\text{F}}^2, \quad (3)$$

where $\mathbf{C} = \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ is the auto-correlation of the input activation, and $\mathbf{C}^{\frac{1}{2}}$ is the matrix square root of $\mathbf{C}$.

However, there is no closed-form solution to the above problem and several heuristic methods have been proposed. Wanda (Sun et al., 2023) computes the ℓ_2-norm of each row of $\mathbf{X}$, i.e., $\|\mathbf{X}[i,:]\|_2$, for $i = 1, \ldots, d_{\text{in}}$, to scale the corresponding column of $\mathbf{W}$, and then performs magnitude based pruning on this scaled matrix to obtain the pruning mask for the weight $\mathbf{W}$. It can be viewed as approximating $\mathbf{C}^{\frac{1}{2}}$ only by its diagonal in Equation (3). Therefore, the cross-correlation information of the input activations $\mathbf{X}$ between each dimension is completely discarded.

Similar to Wanda, the state-of-the-art Activation-aware Weight Quantization (AWQ) method (Lin et al., 2024) computes the scaled ℓ_1-norm of each row of $\mathbf{X}$, i.e., $\|\mathbf{X}[i,:]\|_1/n$, for $i = 1, \ldots, d_{\text{in}}$, to scale the corresponding column of $\mathbf{W}$, and performs quantization on the scaled matrix. We also note that Wanda empirically found that, given a target pruning ratio p, better performance is achieved with a semi-structured pruning, i.e., by specifically requiring uniform pruning ratio p for each row of $\mathbf{W}_{\text{sparse}}$, instead of restricting sparsity at only the whole matrix level.

The Optimal Brain Compression (OBC) method (Frantar et al., 2022b) simplifies the original Optimal Brain Surgeon (OBS) (LeCun et al., 1989; Hassibi et al., 1993) by breaking the compression task into layer-wise sub-problems. For each layer, they use a greedy solver that sequentially prunes (or quantizes) weights. To prune each row of $\mathbf{W}$, they iteratively zero out the entry that results in the smallest increase to approximation loss. However, their follow-up work (Frantar & Alistarh, 2023) mentions that this method is hard to scale to models with billions of parameters, and further propose several approximations (e.g., prune the weights from left to right and only recalculate approximation loss impact for the weights to the right) for speedup, leading to the SparseGPT (Frantar & Alistarh, 2023) for pruning, and the GPTQ (Frantar et al., 2022a) for quantization. Similarly, GPFQ (Lybrand & Saab, 2021; Zhang et al., 2023) uses a greedy path-following mechanism to quantize and/or prune weights from left to right, but has theoretical guarantees.

We refer interested readers to Zhu et al. (2024); Gholami et al. (2022); Nagel et al. (2021); Kim et al. (2023); Wan et al. (2023) for a comprehensive overview of weight quantization and pruning methods. Another line of work on structured pruning, e.g., Ma et al. (2023); Xia et al. (2024), removes entire structured components of a network, but usually involves post-training to recover the performance.

3. Motivation and Method

We further decompose (3) as

$$\mathcal{L}'(\mathbf{W}_{\text{sparse}}) = \sum_{i=1}^{d_{\text{out}}} \|\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}} - \mathbf{W}_{\text{sparse}}[i,:]\mathbf{C}^{\frac{1}{2}}\|_2^2. \quad (4)$$

Interestingly, when optimizing under the constraint

$$\mathcal{C}_{\text{row}} := \left\{\mathbf{\Theta} : \forall i \in \{1, \ldots, d_{\text{out}}\},\ \|\mathbf{\Theta}[i,:]\|_0 \leq k\right\}, \quad (5)$$

the problem for each term of (4) becomes exactly a well-studied sparse approximation problem[2] of the following general form:

$$\min_{\boldsymbol{\theta}} \left[f(\boldsymbol{\theta}) := \|\mathbf{y} - \mathbf{A}\boldsymbol{\theta}\|_2^2\right], \quad (6)$$
$$\text{s.t. } \|\boldsymbol{\theta}\|_0 \leq k := (1-p) \cdot d_{\text{in}},$$

where $\mathbf{y} = (\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}})^\top$, $\mathbf{A} = [\mathbf{C}^{\frac{1}{2}}]^\top = \mathbf{C}^{\frac{1}{2}}$, and $\boldsymbol{\theta}$ is the corresponding $\mathbf{W}_{\text{sparse}}[i,:]^\top$ which has $p \cdot d_{\text{in}}$ zeros that we want to find. $\|\boldsymbol{\theta}\|_0$ is the ℓ_0 pseudo norm of $\boldsymbol{\theta}$ that counts the total number of nonzero elements in $\boldsymbol{\theta}$.

Many existing sparse approximation and compressive sensing algorithms can be used to solve this problem, such as Matching Pursuit (Mallat & Zhang, 1993), Orthogonal Matching Pursuit (OMP) (Cai & Wang, 2011), Iterative Hard Thresholding (IHT) (Blumensath & Davies, 2009), CoSaMP (Tropp & Needell, 2008), Basis Pursuit (Chen et al., 2001), Sparse Bayesian Learning methods (Tipping, 2001; Wipf & Rao, 2004). In fact, the OBC can be viewed as the OMP in reverse order. However, it is well-known in Compressive Sensing that such methods are too greedy and often outperformed by IHT and ℓ_1 type methods.

We adopt the IHT method, since it is efficient to run and also has *recovery guarantees* (Blumensath & Davies, 2009). IHT is a special form of the projected gradient descent (PGD) method. Each iteration of PGD involves calculating the gradient[3] of $f(\boldsymbol{\theta})$, i.e., $\nabla f(\boldsymbol{\theta}^{(t)})$, and projecting the updated weights $(\boldsymbol{\theta}^{(t)} - \eta \nabla f(\boldsymbol{\theta}^{(t)}))$ onto the constraint set. For IHT, the projection onto the constraint $\|\boldsymbol{\theta}\|_0 \leq k$ is simply hard-thresholding, i.e., keeping the k largest-magnitude elements of $(\boldsymbol{\theta}^{(t)} - \eta \nabla f(\boldsymbol{\theta}^{(t)}))$, and setting the remaining elements to 0. Actually all the terms in (4) can be solved independently

[2] Since $\mathbf{C}^{\frac{1}{2}}$ is a square matrix, this is also a sparse linear regression problem.

[3] Using the fact that $\mathbf{y} = (\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}})^\top$ can simplify the gradient computation without doing SVD of $\mathbf{C}$, as explained later.

Algorithm 1 Activation-Aware Projected Gradient Descent

Input: original weight $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, input activation covariance $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top$, constraints $\mathcal{C}$, step size η
Initialize: $\mathbf{\Theta}^{(0)} \in \mathcal{C}$
Repeat
$\mathbf{Z}^{(t)} = \mathbf{\Theta}^{(t)} + \eta(\mathbf{W} - \mathbf{\Theta}^{(t)})\mathbf{C}$;
$\mathbf{\Theta}^{(t+1)} = \mathsf{Proj}_{\mathcal{C}}(\mathbf{Z}^{(t)})$;
until a stopping criterion is met
Output compressed weight $\mathbf{\Theta}$

and in parallel. This can also be viewed as minimizing the following objective, with the constraint of (5) that each row of $\hat{\mathbf{W}}$ is k-sparse:

$$\min_{\hat{\mathbf{W}} \in \mathcal{C}_{\text{row}}} \left[f_1(\hat{\mathbf{W}}) := \|\mathbf{W}\mathbf{C}^{\frac{1}{2}} - \hat{\mathbf{W}}\mathbf{C}^{\frac{1}{2}}\|_{\text{F}}^2 \right]. \tag{7}$$

For this general objective f_1 in (7), its gradient w.r.t. $\hat{\mathbf{W}}$ is:

$$\nabla f_1(\hat{\mathbf{W}}) = -2(\mathbf{W}\mathbf{C}^{\frac{1}{2}} - \hat{\mathbf{W}}\mathbf{C}^{\frac{1}{2}})(\mathbf{C}^{\frac{1}{2}})^\top \tag{8}$$

$$= -2(\mathbf{W} - \hat{\mathbf{W}})\mathbf{C}. \tag{9}$$

Fortunately, even though the objective in Equation (7) has $\mathbf{C}^{\frac{1}{2}}$, the actual gradient calculation in (9) only has $\mathbf{C}$, i.e., we can avoid calculating $\mathbf{C}^{\frac{1}{2}}$ and its expensive SVD computation.

The overall activation-aware PGD method for compressing the weight is described in Algorithm 1, which we refer to as **A**ctivation-aware **W**eight pruning and quantization via **P**GD (AWP). For semi-structured pruning, i.e., where each row is k-sparse, with $\mathcal{C}_{\text{row}}$ as given by (5), $\mathsf{Proj}_{\mathcal{C}_{\text{row}}}(\mathbf{Z})$ simply keeps the k largest-magnitude elements in each row of $\mathbf{Z}$ and sets the remaining elements to 0. For quantization, e.g., in INT4, $\mathsf{Proj}_{\mathcal{C}_{\text{INT4}}}(\mathbf{Z})$ would quantize $\mathbf{Z}$ into INT4.

Furthermore, Algorithm 1 allows joint pruning and quantization, where the constraint set $\mathcal{C}$ becomes the intersection of $\mathcal{C}_{\text{sparse}}$ and $\mathcal{C}_{\text{INT4}}$. One can either prune the quantized version of $\mathbf{Z}$, or first prune $\mathbf{Z}$ and obtain the corresponding sparsity mask, then quantize the pruned version and finally apply the sparsity mask.

Note that the main computational cost of Algorithm 1 is the gradient descent, which involves multiplication between $(\mathbf{W} - \mathbf{\Theta}^{(t)})$ and $\mathbf{C}$, incurring $O(d_{\text{out}} \times d_{\text{in}}^2)$. This is computationally more efficient than inverting $\mathbf{X}\mathbf{X}^\top$ required in OBC, SparseGPT, GPTQ, etc., and can be run in parallel on the GPU.

4. Experiments

We compare the proposed AWP method with state-of-the-art methods on pruning, quantization, as well as joint pruning and quantization. As in AWQ and Wanda, we evaluate the perplexity of the compressed model on the held-out WikiText-2 (Merity et al., 2016) validation set.

4.1. Pruning

To compare with Wanda, SparseGPT, and Magnitude-based pruning, we follow the exact setup of the experiments in Wanda[4]. More specifically, we tested on the Llama-2 7B and 13B models. The calibration data $\mathbf{X}$ is 128 sequences (each has 2048 tokens) sampled from the C4 training set (Raffel et al., 2020). We test pruning ratios from $\{50\%, 60\%, 70\%, 80\%, 90\%\}$. Note that larger pruning ratio corresponds to higher compression rate. For AWP, the iteration stops when the Frobenius norm of the gradient normalized by the Frobenius norm of the original weight is less than 0.0001, or 200 iterations is reached. The step size η is set as $2/\|\mathbf{C}\|_{\text{F}}$. As problem (3) is nonconvex, we initialize $\mathbf{\Theta}^{(0)}$ as the solution of Wanda, since a good initial point helps nonconvex optimization.

Table 1 shows the perplexity of the pruned Llama-2-7B model for different methods and pruning ratios, and Table 2 shows the corresponding results for the Llama-2-13B model. The results of SparseGPT and Magnitude-based pruning are from Sun et al. (2023). As a reference, the perplexity of the original Llama-2-7B dense model (i.e., pruning ratio = 0%) is 5.12 and the perplexity of the original Llama-2-13B dense model is 4.57.

Table 1. Perplexity on WikiText2 of pruned Llama-2-7B model by different methods under different pruning ratios.

	50%	60%	70%	80%	90%
MAGNITUDE	14.89	4e3	-	NAN	-
SPARSEGPT	6.51	9.58	-	1e2	-
WANDA	6.48	10.09	70.04	4e3	1e4
AWP	**6.42**	**9.44**	**22.10**	**83.28**	**8e2**

Table 2. Perplexity on WikiText2 of pruned Llama-2-13B model by different methods under different pruning ratios.

	50%	60%	70%	80%	90%
MAGNITUDE	6.37	11.23	-	5e4	-
SPARSEGPT	5.63	7.80	-	1e2	-
WANDA	5.59	7.97	43.06	1e3	2e4
AWP	**5.54**	**7.49**	**16.57**	**75.68**	**1e3**

We can see that for all methods, the perplexity gets worse when the pruning ratio increases. Nonetheless, the activation-aware methods, i.e., SparseGPT, Wanda, and proposed AWP perform significantly better than the magnitude based pruning, which is not activation-aware. The proposed AWP method has the best perplexity under all pruning ratios, especially when the pruning ratio is over 60%.

4.2. Quantization

We compare with state-of-the-art layer-wise weight quantization methods AWQ[5] and GPTQ[6]. We follow the experi-

[4]https://github.com/locuslab/wanda
[5]https://github.com/mit-han-lab/llm-awq
[6]https://github.com/AutoGPTQ/AutoGPTQ/tree/main

ment setups of AWQ, which focus on weight-only grouped quantization with a group size of 128. As in AWQ, we use a small calibration set from the Pile dataset (Gao et al., 2020) in order not to overfit to a specific downstream domain. Besides INT4 and INT3 quantization, we additionally experiment with INT2 quantization.

For the proposed AWP method, the step size η is set to $1.5/\|\mathbf{C}\|_{\mathrm{F}}$, and we only run 10 iterations. We simply initialize $\mathbf{\Theta}^{(0)}$ to be the straightforward (i.e., not activation-aware) Round-To-Nearest (RTN) quantized version of $\mathbf{W}$.

Table 3 shows the perplexity of the INT4/INT3/INT2 quantized Llama-3.1-8B model by GPTQ, AWQ, and our proposed AWP method. AWP quantized INT4 and INT3 models have better perplexity than corresponding INT4 and INT3 models quantized by GPTQ and AWQ. For INT2 quantization, the resulting perplexities of all methods are very poor, although GPTQ has the lowest perplexity.

Table 3. Perplexity on WikiText2 of quantized Llama-3.1-8B model by different methods.

	INT4	INT3	INT2
GPTQ	9.95	12.54	**2e3**
AWQ	6.64	8.14	$3e4$
AWP	**6.55**	**8.06**	$1e6$

4.3. Joint Pruning and Quantization

Note that, as shown in Table 3, INT4 quantization by AWP achieves good perplexity (much better than INT2). Further, Table 1 and Table 2 show that pruning the original model up to 70% has moderate perplexity degradation using the proposed AWP method. This motivates us to consider combining pruning with quantization to further compress the model, instead of solely pushing the quantization to extremely low-bits.

Note that the proposed AWP method naturally allows joint pruning and quantization. We further compare it with sequential quantization and pruning using state-of-the-art methods AWQ+Wanda, as well as sequential pruning and quantization using Wanda+AWQ. We use INT4 quantization for all methods and test pruning ratios of 25%, 50%, and 75%.

For AWP, we fix the step size $\eta = 1.5/\|\mathbf{C}\|_{\mathrm{F}}$ as in the quantization setting. Instead of directly compressing the model into the target bits and pruning ratio, we first gradually increase the pruning ratio without quantization for 50 iterations, then perform 50 joint pruning and INT4 quantization iterations. More specifically, in the first 25 iterations of purely pruning, we linearly increase the pruning ratio from 0% to the target pruning ratio, then keep this pruning ratio unchanged in the remaining 75 iterations. In each iteration of our joint pruning and INT4 quantization, the projection is $\mathsf{Proj}_{\mathcal{C}_{\text{INT4}}}(\mathsf{Proj}_{\mathcal{C}_{\text{row}}}(\mathbf{Z}))$. At the end of iterations, the corresponding sparsity mask is applied to ensure that the final weight is both sparsified and quantized.

Table 4. Perplexity on WikiText2 of pruned and INT4 quantized Llama-3.1-8B model by different methods.

PRUNING RATIO:	25%	50%	75%
AWQ+WANDA	6.93	9.71	$3e2$
WANDA+AWQ	**6.81**	9.46	$2e2$
AWP	**6.81**	**9.32**	**1e2**

Table 5. Perplexity on WikiText2 of pruned and INT4 quantized Llama-3.2-1B model by different methods.

PRUNING RATIO:	25%	50%	75%
AWQ+WANDA	11.63	23.95	$2e3$
WANDA+AWQ	11.30	21.90	$1e3$
AWP	**11.20**	**18.41**	**3e2**

We first test on the Llama-3.1-8B model. Table 4 shows the perplexity of pruned and INT4 quantized models by different methods under different pruning ratios. We can see that Wanda+AWQ (pruning first) is consistently better than AWQ+Wanda (quantization first) under different pruning ratios. The proposed AWP achieves best performance under all pruning ratios, especially when pruning ratio is high.

An interesting finding is that, compared with INT2 quantization in Table 3, INT4 quantization with 75% pruning ratio achieves significantly better perplexity. Note that 4 bits combined with 75% pruning rate is roughly equivalent to 2 bits, as we need 1 bit to store the pruning mask. As recent efforts try to push the quantization to extremely low bits, e.g., 1.58 bit (Ma et al., 2024; Wang et al., 2024b), our results show that combining quantization with pruning may achieve much better performance.

5. Conclusion and Discussion

We have proposed a layer-wise activation-aware post-training quantization and pruning method based on PGD. We provided a new insight from the compressive sensing framework to compress large foundation models. Empirical studies show that the proposed method outperforms the state-of-the-art LLM pruning and quantization methods. We hope our work will inspire more advanced LLM compression methods by leveraging cutting-edge compressive sensing techniques.

Our future directions include extending pruning to structured sparsity, e.g., NVIDIA's 2:4 sparsity (Mishra et al., 2021), though the current unstructured sparsity can already be accelerated by GPUs like Cerebras Wafer-Scale Engine[7]. Another direction is to provide theoretical guarantees of the proposed method for quantization.

[7]https://www.cerebras.net/blog/harnessing-the-power-of-sparsity-for-large-gpt-ai-models

References

Blumensath, T. and Davies, M. E. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

Cai, T. T. and Wang, L. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, 57(7):4680–4688, 2011. doi: 10.1109/TIT.2011.2146090.

Chen, S. S., Donoho, D. L., and Saunders, M. A. Atomic decomposition by basis pursuit. *SIAM review*, 43(1): 129–159, 2001.

Frantar, E. and Alistarh, D. SparseGPT: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pp. 10323–10337. PMLR, 2023.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022a.

Frantar, E., Singh, S. P., and Alistarh, D. Optimal brain compression: a framework for accurate post-training quantization and pruning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022b. Curran Associates Inc. ISBN 9781713871088.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gholami, A., Kim, S., Zhen, D., Yao, Z., Mahoney, M., and Keutzer, K. *A Survey of Quantization Methods for Efficient Neural Network Inference*, pp. 291–326. 01 2022. ISBN 9781003162810. doi: 10.1201/9781003162810-13.

Hassibi, B., Stork, D. G., and Wolff, G. J. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pp. 293–299. IEEE, 1993.

Hubara, I., Nahshan, Y., Hanani, Y., Banner, R., and Soudry, D. Accurate post training quantization with small calibration sets. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4466–4475. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/hubara21a.html.

Jain, P., Tewari, A., and Kar, P. On iterative hard thresholding methods for high-dimensional m-estimation. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pp. 685–693, Cambridge, MA, USA, 2014. MIT Press.

Kim, S., Hooper, C., Wattanawong, T., Kang, M., Yan, R., Genc, H., Dinh, G., Huang, Q., Keutzer, K., Mahoney, M. W., et al. Full stack optimization of transformer inference: a survey. *arXiv preprint arXiv:2302.14017*, 2023.

LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Li, Y., Gong, R., Tan, X., Yang, Y., Hu, P., Zhang, Q., Yu, F., Wang, W., and Gu, S. BRECQ: pushing the limit of post-training quantization by block reconstruction. *CoRR*, abs/2102.05426, 2021. URL https://arxiv.org/abs/2102.05426.

Liang, T., Glossner, J., Wang, L., Shi, S., and Zhang, X. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.07.045. URL https://www.sciencedirect.com/science/article/pii/S0925231221010894.

Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024.

Liu, H. and Foygel Barber, R. Between hard and soft thresholding: optimal iterative thresholding algorithms. *Information and Inference: A Journal of the IMA*, 9(4): 899–933, 12 2019. ISSN 2049-8772. doi: 10.1093/imaiai/iaz027. URL https://doi.org/10.1093/imaiai/iaz027.

Lybrand, E. and Saab, R. A greedy algorithm for quantizing neural networks. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.

Ma, S., Wang, H., Ma, L., Wang, L., Wang, W., Huang, S., Dong, L., Wang, R., Xue, J., and Wei, F. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 1, 2024.

Ma, X., Fang, G., and Wang, X. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

Mallat, S. G. and Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993. doi: 10.1109/78.258082.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Mishra, A., Latorre, J. A., Pool, J., Stosic, D., Stosic, D., Venkatesh, G., Yu, C., and Micikevicius, P. Accelerating sparse deep neural networks. *arXiv preprint arXiv:2104.08378*, 2021.

Nagel, M., Amjad, R. A., Van Baalen, M., Louizos, C., and Blankevoort, T. Up or down? adaptive rounding for post-training quantization. In *International conference on machine learning*, pp. 7197–7206. PMLR, 2020.

Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., van Baalen, M., and Blankevoort, T. A white paper on neural network quantization. *ArXiv*, abs/2106.08295, 2021. URL https://api.semanticscholar.org/CorpusID:235435934.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Sun, M., Liu, Z., Bair, A., and Kolter, J. Z. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Tipping, M. E. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1 (Jun):211–244, 2001.

Tropp, J. and Needell, D. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. Anal*, 26(3):301–321, 2008.

Wan, Z., Wang, X., Liu, C., Alam, S., Zheng, Y., Liu, J., Qu, Z., Yan, S., Zhu, Y., Zhang, Q., et al. Efficient large language models: A survey. *arXiv preprint arXiv:2312.03863*, 2023.

Wang, C., Wang, Z., Xu, X., Tang, Y., Zhou, J., and Lu, J. Q-VLM: Post-training quantization for large vision-language models. *arXiv preprint arXiv:2410.08119*, 2024a.

Wang, J., Zhou, H., Song, T., Mao, S., Ma, S., Wang, H., Xia, Y., and Wei, F. 1-bit ai infra: Part 1.1, fast and lossless bitnet b1. 58 inference on cpus. *arXiv preprint arXiv:2410.16144*, 2024b.

Wang, P., Chen, Q., He, X., and Cheng, J. Towards accurate post-training network quantization via bit-split and stitching. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9847–9856. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/wang20c.html.

Wang, T., Berthet, Q., and Plan, Y. Average-case hardness of rip certification. *Advances in Neural Information Processing Systems*, 29, 2016.

Wipf, D. P. and Rao, B. D. Sparse bayesian learning for basis selection. *IEEE Transactions on Signal processing*, 52(8):2153–2164, 2004.

Xia, M., Gao, T., Zeng, Z., and Chen, D. Sheared llama: Accelerating language model pre-training via structured pruning. 2024. Publisher Copyright: © 2024 12th International Conference on Learning Representations, ICLR 2024. All rights reserved.; 12th International Conference on Learning Representations, ICLR 2024 ; Conference date: 07-05-2024 Through 11-05-2024.

Zhang, H. and Saab, R. Unified stochastic framework for neural network quantization and pruning. *Applied and Computational Harmonic Analysis*, 79:101778, 2025. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2025.101778. URL https://www.sciencedirect.com/science/article/pii/S1063520325000326.

Zhang, J., Zhou, Y., and Saab, R. Post-training quantization for neural networks with provable guarantees. *SIAM Journal on Mathematics of Data Science*, 5(2):373–399, 2023. doi: 10.1137/22M1511709. URL https://doi.org/10.1137/22M1511709.

Zhu, X., Li, J., Liu, Y., Ma, C., and Wang, W. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577, 2024. doi: 10.1162/tacl_a_00704. URL https://aclanthology.org/2024.tacl-1.85/.

A. Theoretical Justifications of AWP for Pruning

A.1. Convergence under Restricted Isometry Property (RIP)

We first recall the theoretical guarantees of IHT algorithm (Blumensath & Davies, 2009), which uses the iteration:

$$\boldsymbol{\theta}^{(t+1)} = H_k(\boldsymbol{\theta}^{(t)} + \mathbf{A}^\top(\mathbf{y} - \mathbf{A}\boldsymbol{\theta}^{(t)}))$$

where $H_k(\cdot)$ is the hard-thresholding operator that keeps k largest-magnitude elements of the input vector and sets the remaining elements to 0.

Theorem A.1. *[Blumensath & Davies (2009)] Given a noisy observation $\mathbf{y} = \mathbf{A}\boldsymbol{\theta}_k + \mathbf{e}$, where $\boldsymbol{\theta}_k$ is k-sparse. If $\mathbf{A}$ has the restricted isometry property with $\beta_{3k} < 1/8$, then, at iteration t, IHT will recover an approximation $\boldsymbol{\theta}^{(t)}$ satisfying*

$$\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_k\|_2 \leq \|\boldsymbol{\theta}_k\|_2/2^t + 4\|\mathbf{e}\|_2.$$

Furthermore, after at most $t' = \lceil \log_2(\|\boldsymbol{\theta}_k\|_2/\|\mathbf{e}\|_2) \rceil$ iterations,

$$\|\boldsymbol{\theta}^{(t')} - \boldsymbol{\theta}_k\|_2 \leq 5\|\mathbf{e}\|_2.$$

The definition of β_{3k} in RIP can be found in Equation (6) of Blumensath & Davies (2009).

Note that running AWP Algorithm 1 for pruning with $\mathcal{C}_{\text{row}} = \{\boldsymbol{\Theta} : \|\boldsymbol{\Theta}[i,:]\|_0 \leq k, i = 1, ..., d_{\text{out}}\}$ and step size $\eta = 1$ is equivalent to running IHT for each row of the weight matrix in parallel. Recall that in our semi-structured pruning (i.e., each row is targeted to be k-sparse), $\mathcal{C}_{\text{row}} = \{\boldsymbol{\Theta} : \|\boldsymbol{\Theta}[i,:]\|_0 \leq k, i = 1, ..., d_{\text{out}}\}$ and $\mathsf{Proj}_{\mathcal{C}_{\text{row}}}(\mathbf{Z})$ simply keeps k largest-magnitude elements in each row of $\mathbf{Z}$ and set remaining elements to 0.

More specifically, recall that each term in (4) is:

$$\|\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}} - \mathbf{W}_{\text{sparse}}[i,:]\mathbf{C}^{\frac{1}{2}}\|_2^2 = \|\underbrace{(\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}})^\top}_{\mathbf{y}} - \underbrace{(\mathbf{C}^{\frac{1}{2}})^\top}_{\mathbf{A}}\underbrace{\mathbf{W}_{\text{sparse}}[i,:]^\top}_{\boldsymbol{\theta}}\|_2^2.$$

We can view $\mathbf{y} = (\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}})^\top$, $\mathbf{A} = (\mathbf{C}^{\frac{1}{2}})^\top$, and $\mathbf{W}_{\text{sparse}}[i,:]^\top$ corresponds to $\boldsymbol{\theta}$. Let us denote the **global optimal** k-sparse solution $\mathbf{W}_{\text{sparse}}[i,:]$ to the above problem as $\mathbf{W}^*_{\text{sparse}}[i,:]$, which corresponds to $\boldsymbol{\theta}_k$, then we have the corresponding optimal k-sparse approximation error $\mathbf{e} = (\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}} - \mathbf{W}^*_{\text{sparse}}[i,:]\mathbf{C}^{\frac{1}{2}})^\top$, which is treated as a noise term.

Therefore, we have following guarantee of AWP pruning for each row:

Theorem A.2. *If $\mathbf{C}^{\frac{1}{2}}$ has restricted isometry property with $\beta_{3k} < 1/8$, Algorithm 1 with $\mathcal{C}_{row} = \{\boldsymbol{\Theta} : \|\boldsymbol{\Theta}[i,:]\|_0 \leq k, i = 1, ..., d_{out}\}$ and $\eta = 1$ will recover an approximation $\boldsymbol{\Theta}[i,:]^{(t)}$ satisfying*

$$\|\boldsymbol{\Theta}[i,:]^{(t)} - \mathbf{W}^*_{sparse}[i,:]\|_2 \leq \|\mathbf{W}^*_{sparse}[i,:]\|_2/2^t + 4\|\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}} - \mathbf{W}^*_{sparse}[i,:]\mathbf{C}^{\frac{1}{2}}\|_2.$$

Furthermore, after at most $t' = \max_i \lceil \log_2(\|\mathbf{W}^*_{sparse}[i,:]\|_2/\|(\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}} - \mathbf{W}^*_{sparse}[i,:]\mathbf{C}^{\frac{1}{2}})^\top\|_2) \rceil$ iterations,

$$\|\boldsymbol{\Theta}[i,:]^{(t')} - \mathbf{W}^*_{sparse}[i,:]\|_2 \leq 5\|\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}} - \mathbf{W}^*_{sparse}[i,:]\mathbf{C}^{\frac{1}{2}}\|_2.$$

Note that one can always scale the matrix $\mathbf{C}^{\frac{1}{2}}$ in the activation-aware objective function (3).

Combining the error bounds from all rows, we have following corollary:

Corollary A.3. *If $\mathbf{C}^{\frac{1}{2}}$ has restricted isometry property with $\beta_{3k} < 1/8$, running Algorithm 1 with $\mathcal{C}_{row} = \{\boldsymbol{\Theta} : \|\boldsymbol{\Theta}[i,:]\|_0 \leq k, i = 1, ..., d_{out}\}$ and $\eta = 1$ after at most $t' = \max_i \lceil \log_2(\|\mathbf{W}_{sparse}[i,:]^*\|_2/\|(\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}} - \mathbf{W}^*_{sparse}[i,:]\mathbf{C}^{\frac{1}{2}})^\top\|_2) \rceil$ iterations, we have*

$$\|\boldsymbol{\Theta}^{(t')} - \mathbf{W}^*_{sparse}\|_{\text{F}} \leq 5\|\mathbf{W}\mathbf{C}^{\frac{1}{2}} - \mathbf{W}^*_{sparse}\mathbf{C}^{\frac{1}{2}}\|_{\text{F}}. \tag{10}$$

Proof. From Theorem A.2, we have

$$\|\boldsymbol{\Theta}[i,:]^{(t')} - \mathbf{W}^*_{\text{sparse}}[i,:]\|_2 \leq 5\|\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}} - \mathbf{W}^*_{\text{sparse}}[i,:]\mathbf{C}^{\frac{1}{2}}\|_2.$$

Squaring both sides, we have

$$\|\mathbf{\Theta}[i,:]^{(t')} - \mathbf{W}^*_{\text{sparse}}[i,:]\|_2^2 \leq 25\|\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}} - \mathbf{W}^*_{\text{sparse}}[i,:]\mathbf{C}^{\frac{1}{2}}\|_2^2.$$

Summing over all rows, we have

$$\sum_i^{d_{\text{out}}} \|\mathbf{\Theta}[i,:]^{(t')} - \mathbf{W}^*_{\text{sparse}}[i,:]\|_2^2 \leq 25 \sum_i^{d_{\text{out}}} \|\mathbf{W}[i,:]\mathbf{C}^{\frac{1}{2}} - \mathbf{W}^*_{\text{sparse}}[i,:]\mathbf{C}^{\frac{1}{2}}\|_2^2.$$

So we have

$$\|\mathbf{\Theta}^{(t')} - \mathbf{W}^*_{\text{sparse}}\|_{\text{F}}^2 \leq 25\|\mathbf{W}\mathbf{C}^{\frac{1}{2}} - \mathbf{W}^*_{\text{sparse}}\mathbf{C}^{\frac{1}{2}}\|_{\text{F}}^2.$$

Finally, taking square root on both sides, we obtain Equation (10). □

A.2. Convergence under Restricted Strong Convexity (RSC) and Restricted Smoothness (RSM) Property

On the other hand, instead of requiring the restricted isometry property (RIP) of $\mathbf{C}^{\frac{1}{2}}$, which is not easy to verify (Wang et al., 2016), Jain et al. (2014) provided the recovery guarantee of IHT method based on Restricted Strong Convexity (RSC) and Restricted Smoothness (RSM) properties defined below (Jain et al., 2014; Liu & Foygel Barber, 2019):

Definition A.4. (RSC Property) A differential function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies restricted strong convexity with parameter α at sparsity level k, abbreviated as (α, k)-RSC, if the following holds for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ $s.t.$ $\|\boldsymbol{\theta}_1\|_0 \leq k$ and $\|\boldsymbol{\theta}_2\|_0 \leq k$:

$$f(\boldsymbol{\theta}_1) \geq f(\boldsymbol{\theta}_2) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 \rangle + \frac{\alpha}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.$$

Definition A.5. (RSM Property) A differential function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies restricted strong smoothness with parameter β at sparsity level k, abbreviated as (β, k)-RSM, if the following holds for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ $s.t.$ $\|\boldsymbol{\theta}_1\|_0 \leq k$ and $\|\boldsymbol{\theta}_2\|_0 \leq k$:

$$f(\boldsymbol{\theta}_1) \leq f(\boldsymbol{\theta}_2) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 \rangle + \frac{\beta}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.$$

Based on the above two properties, Liu & Foygel Barber (2019) states that for an objective function f satisfying (α, k)-RSC and (β, k)-RSM, IHT with step size $\eta \propto 1/\beta$, i.e., $\boldsymbol{\theta}^{(t+1)} = H_k(\boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^{(t)}))$, satisfies

$$f(\boldsymbol{\theta}^{(t)}) \leq \min_{\|\boldsymbol{\theta}\|_0 \leq k/(32\kappa^2)} \big[f(\boldsymbol{\theta}) + \big(1 - \frac{1}{12\kappa}\big)^t \cdot \big(f(\boldsymbol{\theta}^{(0)}) - f(\boldsymbol{\theta})\big)\big],$$

where $\kappa = \beta/\alpha$, known as the condition number of Hessian of f. In other words, it shows linear convergence to the bound

$$\lim_{t\to\infty} f(\boldsymbol{\theta}^{(t)}) \leq \min_{\|\boldsymbol{\theta}\|_0 \leq k/(32\kappa^2)} f(\boldsymbol{\theta}).$$

For our quadratic objective $f(\boldsymbol{\theta}) := \|\mathbf{y} - \mathbf{A}\boldsymbol{\theta}\|_2^2$ in Equation (6), we have exactly

$$f(\boldsymbol{\theta}_1) = f(\boldsymbol{\theta}_2) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 \rangle + (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)^\top \frac{H_f(\boldsymbol{\theta}_2)}{2}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \tag{11}$$

$$= f(\boldsymbol{\theta}_2) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 \rangle + (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)^\top \frac{2\mathbf{A}^\top\mathbf{A}}{2}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \tag{12}$$

$$= f(\boldsymbol{\theta}_2) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 \rangle + (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)^\top \frac{2\mathbf{C}}{2}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2). \tag{13}$$

Therefore, one can simply use the smallest singular value of $2\mathbf{C}$, i.e., $2\lambda_{\min}(\mathbf{C})$ as α, and largest singular value of $2\mathbf{C}$, i.e., $2\lambda_{\max}(\mathbf{C})$ as β. In other words, our objective (6) satisfies $(2\lambda_{\min}(\mathbf{C}), k)$-RSC and $(2\lambda_{\max}(\mathbf{C}), k)$-RSM, and AWP pruning inherits above convergence and recovery guarantees with $\kappa = \lambda_{\max}(\mathbf{C})/\lambda_{\min}(\mathbf{C})$, i.e., the condition number of $\mathbf{C}$.

Remark A.6. Note that, in Definition A.4 and Definition A.5, $\|\boldsymbol{\theta}_1\|_0 \leq k$ and $\|\boldsymbol{\theta}_2\|_0 \leq k$, therefore $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_0 \leq \min(2k, d_{\text{in}})$, which is still a sparse vector when $k < d_{\text{in}}/2$. So $2\lambda_{\min}(\mathbf{C})$ and $2\lambda_{\max}(\mathbf{C})$ are usually loose lower and upper bounds for α and β, respectively, and $\lambda_{\max}(\mathbf{C})/\lambda_{\min}(\mathbf{C})$ is usually a loose upper bound for κ. One can see that, the smaller the κ, the better the convergence guarantee and recovery guarantee.

B. Activation-Aware Loss Derivation

The expression for the activation-aware loss in Equation (3) is derived as follows:

$$\begin{aligned}\|\mathbf{W}\mathbf{X} - \mathbf{W}_{\text{sparse}}\mathbf{X}\|_{\text{F}}^2 &= \text{tr}\big[(\mathbf{W} - \mathbf{W}_{\text{sparse}})(\mathbf{X}\mathbf{X}^\top)(\mathbf{W} - \mathbf{W}_{\text{sparse}})^\top\big] \\ &= \text{tr}\big[(\mathbf{W} - \mathbf{W}_{\text{sparse}})(\mathbf{X}\mathbf{X}^\top)^{1/2}(\mathbf{X}\mathbf{X}^\top)^{1/2}(\mathbf{W} - \mathbf{W}_{\text{sparse}})^\top\big] \\ &= \|(\mathbf{W} - \mathbf{W}_{\text{sparse}})(\mathbf{X}\mathbf{X}^\top)^{1/2}\|_{\text{F}}^2 \\ &= \|\mathbf{W}(\mathbf{X}\mathbf{X}^\top)^{1/2} - \mathbf{W}_{\text{sparse}}(\mathbf{X}\mathbf{X}^\top)^{1/2}\|_{\text{F}}^2.\end{aligned}$$

C. Activation-aware Loss w.r.t. Iterations

Figure 1 shows an example of normalized activation-aware loss (7), i.e., $\|\mathbf{W}\mathbf{C}^{\frac{1}{2}} - \mathbf{\Theta}^{(t)}\mathbf{C}^{\frac{1}{2}}\|_{\text{F}}/\|\mathbf{W}\|_{\text{F}}$, w.r.t. iteration t during AWP pruning of a layer in the Llama-2 7B model. We can see that such activation-aware approximation loss is effectively minimized by the proposed activation-aware projected gradient descent method described in Algorithm 1.

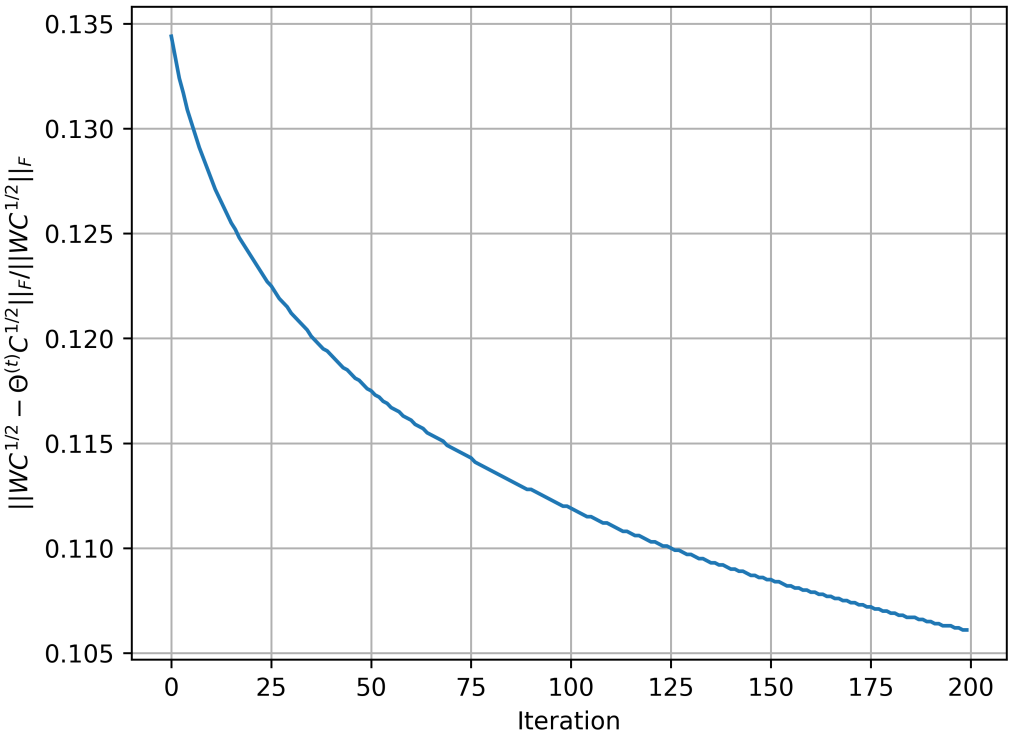

Figure 1. $\|\mathbf{W}\mathbf{C}^{\frac{1}{2}} - \mathbf{\Theta}^{(t)}\mathbf{C}^{\frac{1}{2}}\|_{\text{F}}/\|\mathbf{W}\|_{\text{F}}$, w.r.t. iteration t during AWP pruning of a layer in the Llama-2 7B model.