Price of Efficiency: Interpreting the Effects of Quantization on LLMs

Anonymous ACL submission

Abstract

Quantization offers a practical solution to deploy LLMs in resource-constraint environments. But it's effect on internal representation is understudied, which can question it's reliability. In this study, using various interpretation techniques, we explore the effects of quantization on model and neuron's behavior. We investigate two LLMs Phi-2 and Llama-2-7b, employing 4-bit and 8-bit quantization. Our findings reveal several important insights. First, 4-bit quantized models exhibit slightly better calibration than 8-bit and 16-bit models. Second, our analysis of neuron activations indicates that the number of dead neurons, i.e., those with activation values close to 0 across the dataset, remains consistent regardless of quantization. Regarding contribution of neurons in model prediction, we observe that full-precision models have fewer salient neurons overall. The effect of quantization on neuron redundancy varies across models. Our findings suggest that quantization is a viable approach for the efficient and reliable deployment of LLMs in resource-constrained environments.

1 Introduction

The last decade has seen a tremendous amount of work done in language modeling, specifically in large language models (LLMs) (Devlin et al., 2019; Liu et al., 2023a; Touvron et al., 2023). There is a common trend to increase the number of parameters in LLMs to improve the performance of models. However, this approach exacerbates the challenge of resource requirements, including computational and energy costs (Patterson et al., 2021). Quantization is one of the model compression techniques that is widely used because of its effectiveness and simplicity (Bondarenko et al., 2024; Dettmers et al., 2022; Wu et al., 2023). Quantization reduces the model size by using lower precision weights and/or activations, which can improve its inference speed while using less storage space. The effect of quantization is generally measured by comparing a model's performance on downstream NLP tasks (Li et al., 2024; Kurtić et al., 2024).

While performance on downstream tasks is crucial to understand the end-to-end impact, the evaluation is limited to a set of downstream tasks used for evaluation. In other words, it does not provide complete insights into the effect of quantization on the knowledge learned by the model. In this work, we argue that the interpretation serves as an additional metric and evidence to analyze the effect of quantization on the model. For instance, it may reveal which types of knowledge or relationships preserved or degraded by quantization, giving a deeper understanding of whether essential patterns remain intact. This is especially important for safety-critical applications such as finance, law, and healthcare (Hassan et al., 2024) where reliability of a model is necessary . The insights from interpretation can further motivate creation of selective quantization strategies, where precision can be preserved in sensitive parts of the network while being reduced elsewhere, balancing efficiency and performance.

In this research, we study the effect of quantization, specifically LLMs quantized in 4-bit and 8-bit, to investigate its' effect on the model's behavior and internal representations.

Specifically, we target the research questions given below:

1. What is the effect of quantization on a model's confidence and calibration?
2. Does quantization influence the contribution of neurons to model predictions?
3. How does quantization affect the number of "dead neurons"?
4. Does quantization affect the redundancy of neurons? In other words, does it result in more neurons learning identical information?

We analyze two open-source models, Phi-2 (Javaheripi and Bubeck, 2023) and Llama-2-7b (Touvron et al., 2023) under two quantization settings: 4-bit (Dettmers et al., 2023) and 8-bit (Dettmers et al., 2022) and compare them with the full-precision float-16 weight model. We found that these LLMs under different quantizations remain similar in some aspects and are positively impacted in other aspects, such as model calibration. This provides an empirical evidence on reliability of quantized model.

We summarize our main findings as follows:

1. Model confidence remains consistent across quantization.
2. 4-bit quantized model exhibits less calibration error for both subject models.
3. Based on neuron activations, quantization does not have a major effect, i.e., the number of dead neurons remains largely unchanged.
4. In the attribution-based neuron contribution, we observe that the full-precision model has a lower number of salient neurons.
5. Neuron redundancy differs between the two subject models. In Phi-2, the full-precision model exhibits a higher number of correlated neuron pairs, indicating greater redundancy, whereas in Llama-2-7b, quantization causes only a minor difference in redundancy.

2 Methodology

We study the model confidence, output calibration, neuron activation and attribution with respect to quantization.

2.1 Confidence Analysis

Confidence analysis aims to find the average confidence of a model in its predictions over a dataset (Abdar et al., 2021). We calculate the average confidence of the model using the following equation:

$$\text{Average Confidence} = \frac{1}{N}\sum_{i=1}^{N} \max P(y_i)$$

Here, N is the total number of data points in the dataset, and $P(y_i)$ represents the softmax probability of the output label y_i with the highest probability for the i-th prediction. The term $\max\big(P(y_i)\big)$ indicates the confidence of the model in its selected prediction for each example.

2.2 Calibration Analysis

Calibration can be defined as the degree to which a model's predicted probabilities reflect the actual frequencies of those outcomes (Nixon et al., 2020). Despite high accuracy in classification tasks, modern deep neural networks often suffer from *miscalibration*—meaning that their confidence scores do not accurately represent their probability of correctness (Guo et al., 2017).

We use the Adaptive Calibration Error (ACE) metric (Nixon et al., 2020), which adjusts its assessment based on the actual distribution of confidence values, enabling a more flexible and precise evaluation of calibration. ACE is calculated as follows:

$$\text{ACE} = \frac{1}{KR}\sum_{k=1}^{K}\sum_{r=1}^{R} |\text{acc}(r,k) - \text{conf}(r,k)|$$

Here, K is the number of classes., R is the number of adaptive calibration ranges, $\text{acc}(r,k)$ and $\text{conf}(r,k)$ are the accuracy and confidence values for the adaptive range r for class k, respectively. The calibration range r is determined by dividing the predictions into R equally populated intervals based on sorted confidence scores. This way, each range contains approximately $\lfloor N/R \rfloor$ predictions, where N is the total number of data points.

2.3 Neuron's Attribution

A neuron's attribution refers to its role and significance in a model's predictions for a given dataset, as determined by attribution methods such as integrated gradient (Sundararajan et al., 2017). To evaluate the impact of quantization on neuron attributions, we analyze the number of salient neurons that contribute significantly to the model's predictions. This analysis shows quantization affects on the model's ability to identify and rely on the important features.

Using Layer Integrated Gradients, we obtain attribution scores for each input token for a given layer as:

$$\text{IG}([x_1, x_2, \ldots, x_n]) = \{a_1, a_2, \ldots, a_n\}$$

Here, x_i represents each input token and a_i is the attribution score for the token x_i.

The attribution score a_i is calculated as the sum of the contributions from individual neurons in a given layer:

$$a_i = \sum_{j=1}^{N} n_j$$

where N is the total neurons in the given layer, and n_j is the attribution score of neuron j.

Selection of Top Contributing Neurons: The input to the model consists of a sequence of tokens. We propose two separate methods to select the salient neuron with respect to the prediction. Specifically, we select most salient neurons based on 1) the most salient input token and 2) the input sequence and combine them. Each technique highlights neurons with varying levels of granularity and context sensitivity.

Most attributed token-based: In this technique, we only consider the most attributed token's (i.e., input token with max attribution score) representation and select neurons that have a normalized attribution score > 0.8. This identifies neurons that are most important in determining the model's predictions for the specific context of the selected token. Given as:

$$x_{best} = \arg\max_i \{a_i\}$$

$$n_j^{\text{salient}} = \{n_j \mid \frac{n_j}{\max(n_j)} > 0.8\}, \forall j \in \text{Layer}$$

Here, a_i is the attribution score for token x_i and n_j is the attribution score of neuron j for x_{best}.

Input sequence-based: To identify neurons that are salient in the context of the input sequence, we calculate the total attribution over the entire input sequence by summing the attributions across all input tokens. We select the neurons that have an attribution score > 0.8 after normalization. This approach ensures that the selected neurons reflect their contributions to the overall meaning of the input, rather than being limited to the most attributed token only. Given as:

$$s_j = \sum_{i=1}^{n} a_{ij}$$

$$n_j^{\text{salient}} = \{n_j \mid \frac{s_j}{\max(s_j)} > 0.8\}, \forall j \in Layer$$

Here, a_{ij} is the attribution of neuron j for token x_i, and s_j is the total attribution score of neuron j summed over all tokens.

Token-agnostic: Here, we select the attribution score of a neuron based on its maximum attribution over all tokens in the input sequence. This selection emphasizes neurons important for any part of the input sequence, regardless of specific tokens. Given as:

$$m_j = \max_i \{a_{ij}\}$$

$$n_j^{\text{salient}} = \{n_j \mid \frac{m_j}{\max(m_j)} > 0.8\}, \forall j \in Layer$$

Here, a_{ij} is the attribution score of neuron j for token x_i, and m_j is the maximum attribution score for neuron j over all tokens.

Using all the strategies outlined above, we identify the most important neurons contributing to a single datapoint prediction and collate it over the dataset. Although the same neurons may be selected under different strategies, we consider only one occurrence of each selected neuron.

2.4 Neuron's Activations

Given the quantization reduced the precision of weights, it may increase the number of insignificant neurons in the network. To select insignificant neurons, we adopt a similar approach to Voita et al. (2023), identifying *dead neurons*—neurons whose activations remain consistently close to zero across the dataset.

2.4.1 Dead/Insignificant Neurons

Voita et al. (2023) observed that the number of dead neurons increases with the growth of a model's parameter count. Their analysis of the OPT language model family, which uses the ReLU activation function, shows that over 70% of neurons in some layers are dead. We hypothesize that quantization, by reducing the precision of weights, may contribute to an increase in the number of dead neurons in the network.

Apart from ReLU, other activation functions such as GELU (Hendrycks and Gimpel, 2016) and SiLU (Elfwing et al., 2017) may not produce activation values that are exactly zero. To generalize the concept of dead neurons for these activation functions, we define a threshold of -0.1 to 0.1, categorizing neurons as dead if their activation values consistently remain within this range across the dataset. For different activation functions, we define dead neurons as follows:

$$n_j^{dead}(ReLU) = \{n_j \mid \quad a_{j,d} = 0, \quad \forall d \in dataset\} \tag{1}$$

$$n_j^{dead}(OtherActivations) = \{n_j \mid -0.1 \leq a_{j,d} \leq 0.1, \forall d \in dataset\} \quad (2)$$

Here, $a_{j,d}$ represents the activation of neuron n_j for a given data point d in the dataset.

2.5 Correlation Analysis

We hypothesize that a low-precision quantization may cause more neurons to represent identical information, i.e., as precision is reduced, high precision neuron values may map to the same low precision value. Similar to Dalvi et al. (2020), we calculate the Pearson correlation of neurons at a layer to identify neurons representing similar information. The Pearson correlation is given by:

$$r = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}}$$

Here, x and y are activation arrays for the selected neuron pair. μ_x and μ_y are the means of x and y, respectively, and n is the number of elements in the arrays. $\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}$ and $\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}$ are standard deviation for x and y respectively. The value of r ranges between -1 and 1, where $r = 1$ indicates perfect positive correlation, $r = -1$ indicates perfect negative correlation, and $r = 0$ indicates no linear correlation.

In this study, we use the absolute values of correlation to focus solely on the strength of the relationship, disregarding its positive or negative direction. We consider a neuron pair to be redundant if their correlation score $r > 0.8$.

3 Experiment Setup

3.1 Datasets

We used two datasets for this study: BoolQ (Clark et al., 2019) and the Jigsaw Toxicity dataset (cjadams et al., 2017). BoolQ is a question-answering dataset, while the Jigsaw Toxicity dataset focuses on toxicity classification.

Dataset	Train	Validation	Test
BoolQ	9,427	3,270	-
Jigsaw Toxicity	159,570	63,977	89,185

Table 1: Datapoints count in different split for BoolQ and Jigsaw Toxicity dataset

Table 1 shows the number of datapoints in different splits for both datasets. For our experiment, we have used 10k datapoints for BoolQ after combining the train and validation sets, and for the Toxicity dataset, 9k randomly sampled datapoints from the train set. The label distributions for these datasets are as follows: BoolQ has 62% true labels and 38% false labels, while the Toxicity dataset is balanced with 50% true labels and 50% false labels.

To limit computational resource usage, we performed activation-based analysis exclusively on the BoolQ dataset, which contains 8,421 unique tokens for Phi-2 and 6,472 unique tokens for Llama-2-7b after tokenization. However, both datasets were included in other analyses.

Both datasets have binary outputs, either *true* or *false*. To align model output to be binary, we modify the prompt to instruct the primary models to generate output as either *true* or *false*. Appendix A shows sample prompts and gold outputs from the BoolQ and Jigsaw Toxicity datasets, respectively.

3.2 Models

The primary models analyzed in our study are Phi-2 (Javaheripi and Bubeck, 2023) and Llama-2-7B (Touvron et al., 2023). Both models feature a similar decoder-only architecture (Vaswani et al., 2017), each comprising 32 decoder blocks.

To examine the internal representations within these models, we focus on the output of the first feed-forward layer in the multi-layer perceptron (MLP) block, post-activation. We select this layer as our analysis on the effect of quantization on dead neurons expects output from the activation function. For computational efficiency, experiments are conducted on selected layers at decoder blocks 1, 15, and 32 (further in the study named as Layer 1, 15, and 32). Within each of these layers, Phi-2 and Llama-2-7B models contain 10,240 and 11,008 neurons, respectively. These models differ in their choice of activation functions: Phi-2 employs the Gaussian Error Linear Unit (GELU) activation function (Hendrycks and Gimpel, 2016), while Llama-2-7B uses Sigmoid Linear Unit (SiLU) (Elfwing et al., 2017).

To find the number of dead neurons and compare results with the ReLU activation function, we include the OPT-6.7B model from the OPT model family (Zhang et al., 2022). This model utilizes a similar decoder-only architecture with a layer structure containing 16,384 neurons.

During generation, the seed is set to 42,

and default arguments from the Huggingface `transformers` library are used.

3.3 Quantization Configurations

To perform comparative analysis across models under different quantization settings, we employed two widely-used quantization techniques: 4-bit (Dettmers et al., 2023) and 8-bit (Dettmers et al., 2022). Models are quantized using bitsandbytes config through Huggingface transformers integration. Table 2 shows the hyperparameters used during quantization.

Hyperparameter	Value
8-bit Quantization	
load_in_8bit	True
bnb_8bit_compute_dtype	torch.float16
bnb_8bit_use_double_quant	True
4-bit Quantization	
load_in_4bit	True
bnb_4bit_quant_type	nf4
bnb_4bit_use_double_quant	True
bnb_4bit_compute_dtype	torch.float16

Table 2: Quantization Hyperparameters

3.4 Attribution Technique

To find salient neurons in a neural network, we use Integrated Gradients (Sundararajan et al., 2017) using Captum (Kokhlikyan et al., 2020; Miglani et al., 2023).

4 Findings

4.1 Accuracy

We calculate the accuracy of subject models for selected datasets, i.e., BoolQ (Clark et al., 2019) and Jigsaw Toxicity dataset (cjadams et al., 2017) to ensure that the models under observation have comparable performance to results reported in the literature. Since both datasets require the output token to be either *true* or *false*, we constrain model generation to a single token. As language models can start generation with arbitrary tokens (such as " ", "\n", "Answer: " etc.) even after giving instruction prompt, we only inspect the Softmax probability of *true* or *false*, and whichever has the highest probability is selected as the final model generation. This final token is then used to calculate the accuracy for both models in all quantizations.

Figure 1 presents a line chart depicting the accuracy of both models across various levels of quantization. The x-axis represents different quantization levels, while accuracy is displayed on the y-axis.

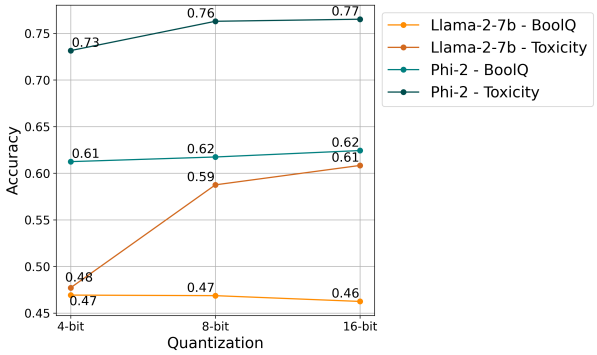

Figure 1: Accuracy of Phi-2 and Llama-2-7b on BoolQ and Toxicity datasets within different quantizations.

Model	Quant.	L1	L15	L32	Total
Phi-2	4-bit	65	1004	35	1104
	8-bit	61	1048	45	1154
	16-bit	57	868	41	966
Llama-2-7b	4-bit	39	1334	20	1393
	8-bit	52	1209	18	1279
	16-bit	66	1198	16	1280

Table 3: Number of salient neurons for Phi-2 (total neurons - 10, 240) and Llama-2-7b (total neurons - 11, 008) across quantizations (Quant.) within different layers (L∗).

Quantization effect on accuracy is dataset-dependent; for the BoolQ dataset, both the subject models, irrespective of quantization, exhibit similar performance. For Toxicity dataset, 4-bit quantized model has worse accuracy, notably in Llama-2-7b it dropped 10% and in Phi-2 it decreased by 3%. However, we observe comparable accuracy for both datasets for 8-bit and 16-bit models.

4.2 Effect of Quantization on Confidence and Calibration

In this analysis, we observe the effect of quantization on the model's confidence and calibration.

4.2.1 Confidence Analysis

Figure 2 shows the average confidence of subject models across quantization separately for both datasets. Overall, we notice a very minor effect with only 0.02 due to quantization on average confidence, with the exception of 4-bit quantized Llama-2-7b on the toxicity dataset where average confidence dropped ∼0.12. For BoolQ, quantization has stable accuracy for both subject models.

Interestingly, comparing average confidence with our previous analysis on accuracy, we do not

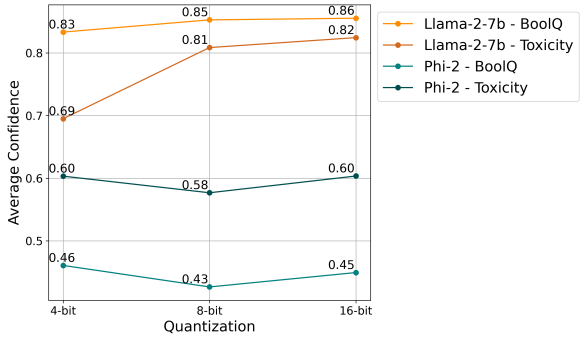

Figure 2: Average confidence of Phi-2 and Llama-2-7b under different quantizations.

observe similar values for accuracy and average confidence of the model. In particular, for a given model, whether quantized or full-precision, we notice more than 10% difference in accuracy and average confidence. This observation motivated our subsequent analysis on model calibration.

4.2.2 Calibration Analysis

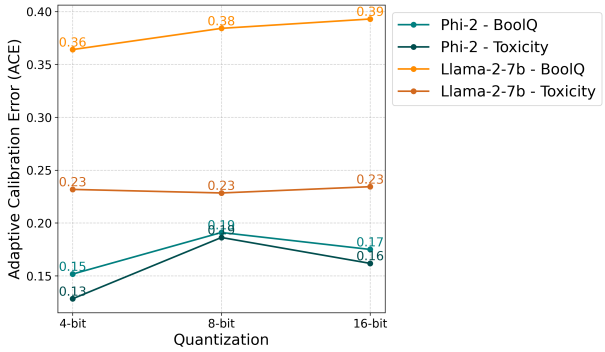

Figure 3: Adaptive Calibration Error (ACE) scores for Phi-2 and Llama-2-7b on BoolQ and Toxicity dataset within different quantizations (lower is better).

To evaluate the calibration of a model, we need to assess how well the model's predicted probabilities align with the true likelihood of outcomes. Figure 3 presents the Adaptive Calibration Error (ACE) scores, illustrating the impact of quantization on model calibration. As results indicate, using 4-bit quantization slightly decreases the error in calibration which makes it better calibrated, making it a better option to deploy models using this quantization where calibration is of utmost importance. This drop in calibration error is the result of quantized model dropped accuracy and equivalently drop in confidence for those predictions making it better calibrated.

Phi-2 model exhibits lower error even in full-precision. While it slightly increases in the 8-bit quantized model, it further decreases again with 4-bit quantization. For the Llama-2-7b model, the results vary by dataset. On the toxicity dataset, the calibration error remains consistent across different quantization levels. However, for the BoolQ dataset, the overall calibration error is higher. Nonetheless, there is a declining trend with respect to quantization levels, with lower quantization (e.g., 4-bit) reducing the error by approximately 0.3 when compared with full-precision.

Model	Quant.	L1 (%)	L15 (%)	L32 (%)
OPT-6.7B	4-bit	23.43	0.35	**0.12**
	8-bit	23.45	0.26	0.15
	16-bit	**23.35**	**0.24**	0.14
Phi-2	4-bit	**21.46**	0.00	0.01
	8-bit	21.52	0.00	0.01
	16-bit	21.51	0.00	0.01
Llama-2-7B	4-bit	0.05	0.00	0.00
	8-bit	0.05	0.00	0.00
	16-bit	0.05	0.00	0.00

Table 4: Percentage of dead neurons across models and quantizations(Quant.) within different layers (L∗).

4.3 Quantization's Effect on the Contribution of Neurons to Model Predictions

Table 3 shows the count of salient neurons for both the Phi-2 and Llama-2-7b within different quantization, divided by layers. Phi-2 in full precision has the least number of salient neurons, i.e., 966, compared to 4-bit and 8-bit which have 1104 and 1154 salient neurons respectively. This highlights that in full-precision model there are fewer neurons contributing to the final prediction for the BoolQ dataset. A higher number of salient neurons in a quantized model indicates that quantization makes the model sensitive to certain features as more neurons need to contribute to the final prediction. As the attribution technique quantizes the contribution of each input token in final prediction and each input token attribution is sum over individual neurons contribution for that token, a higher number of neurons indicates that the model becomes more sensitive to certain features, and more neurons need to collectively contribute to prediction.

Llama-2-7b has a similar number of salient neurons for 8-bit and 16-bit as 1279 and 1280 respectively. The 4-bit quantized model contains a higher number of salient neurons given as 1393, which

indicates that in 4-bit quantization, there are relatively more neurons contributing to the prediction of the model.

4.4 Quantization Affect on the number of "dead neurons"?

To measure the effect of quantization on neuron activation we report the number of dead neurons across models and quantizations.

As shown in Table 4, quantization causes only a minor change in the count of dead neurons. The trend across layers seems to be consistent, as the initial layer has sparse neurons, while the intermediate and the final layer contain few to none, with the exception of Llama-2-7b, in which there are only 0.5% dead neurons in the initial layer and no dead neurons in middle and last layer.

This pattern likely reflects the role of initial layers in learning sparse, low-level features, while later layers capture higher-level contextual features (Dalvi et al., 2022; Voita et al., 2023). In Llama-2-7b, we hypothesize that the consistently low count of dead neurons is due to the use of the SiLU activation function.

4.5 Quantization's Affect on the Redundancy of Neurons

As identified in the works of Dalvi et al. (2020) language models can maintain 97% of performance while using only 10% of the original neurons. This finding is valuable for model pruning efforts. We investigate whether quantization leads to higher redundancy. We perform correlation analysis on neurons of a model where a high correlation reflects higher redundancy.

4.5.1 Correlation Analysis

Figures 4 and 5 show neuron pairs count corresponding to correlation scores for 4-bit, 8-bit and full-precision configurations of Phi-2 and Llama-2-7b respectively. The X-axis highlights the different correlation score bins ranging from 0.0-0.1 to 0.9-1.0. This binning process helps to clearly observe the redundant neuron pairs count across all the layers. The Y-axis shows the count of neuron pairs that fall in that bin. Notice that the count is given for neuron pairs across all the layers, as our main focus for this analysis is to observe the effect on redundancy of neurons within different quantizations.

Considering the highly correlated neurons, i.e., bins having correlation score >=0.8, Phi-2 in full precision shows the highest redundancy, with 907,352 correlated neuron pairs, compared to 781,583 in the 4-bit and 748,867 in the 8-bit configurations. This points to Phi-2 in full-precision having higher redundancy compared to quantized models.

In Llama-2-7b, the 8-bit configuration has the highest redundancy with 24,124 correlated neuron pairs, which is slightly better in 4-bit with 23,315 pairs. unlike Phi-2, the full-precision Llama-2-7b has the fewest correlated pairs (21,644), indicating lower redundancy compared to its quantized versions. However, the difference between neuron pairs in quantized versions is not as substantial as Phi-2.

5 Related Work

This section reviews the relevant literature and recent advancements in quantization techniques and their analysis.

5.1 Quantization Techniques and Analysis

Quantization (Gray and Neuhoff, 1998) is a technique used to reduce the memory requirement by reducing the size of weight and/or activation and increasing the inference time of a model (Jacob et al., 2017; Gholami et al., 2021).

Quantization can be applied by re-training the model, i.e., Quantization aware training or after the training, i.e., post-training quantization.

Quantization-aware training (QAT) is costly and uses re-training of a model on a dataset to maintain accuracy (Liu et al., 2023b; Du et al., 2024; Dettmers et al., 2023; Kim et al., 2023).

Post-training quantization quantizes models without any additional finetuning of the model with a limited dataset, but also suffers from performance issues (Banner et al., 2019; Cai et al., 2020). In case of LLM's Post Training Quantization can be of 3 types: i) Weight-Only Quantization (Park et al., 2024; Frantar et al., 2023; Chee et al., 2024; Lin et al., 2024), ii) Weight-Activation Quantization (Yao et al., 2022; Yuan et al., 2023; Guo et al., 2023; Wei et al., 2023), and iii) KV Cache Quantization (Hooper et al., 2024; Yue et al., 2024).

Xia et al. (2021) explores confidence and calibration relation between quantized and full-precision model by using symmetric quantization. Proskurina et al. (2024) shows quantization improves calibration in LLMs.

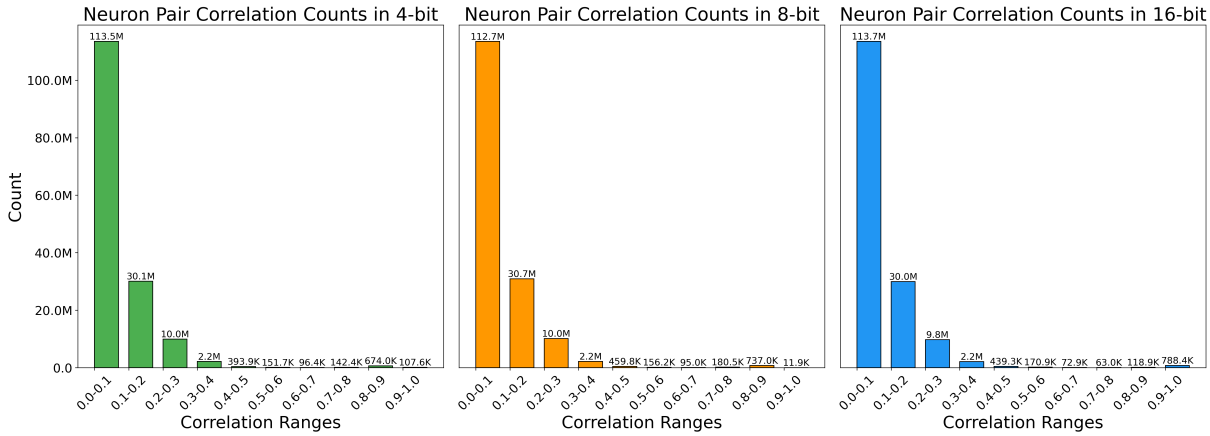

Figure 4: Neurons pair count based on correlation for Phi-2.

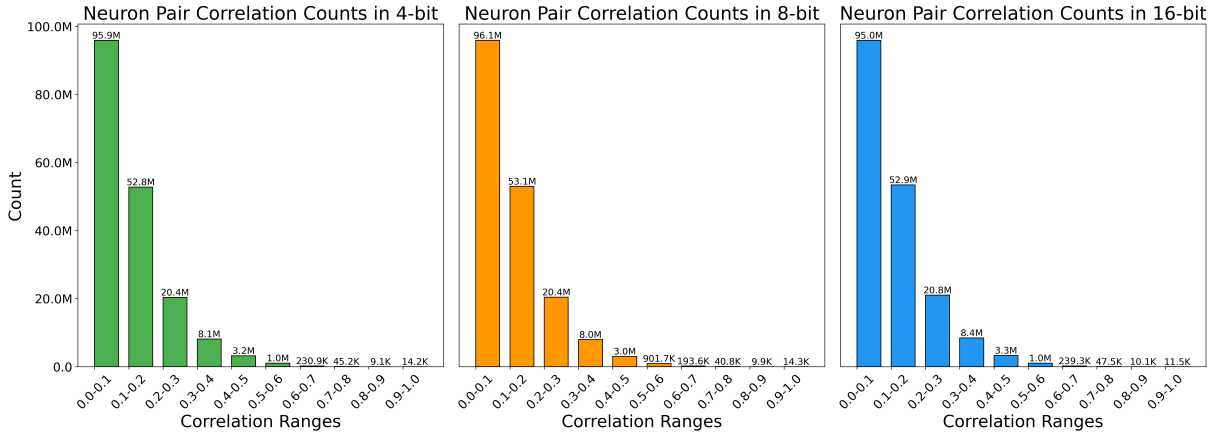

Figure 5: Neurons pair count based on correlation for Llama-2-7b.

6 Conclusion

In this study, we have investigated the impact of quantization on internal representations of LLMs. Our experimental settings focused on two main LLMs: Microsoft's Phi-2 and Meta's Llama-2-7b, employing two widely adopted quantization techniques - 4-bit and 8-bit precision. To evaluate the effects of quantization, we utilized two datasets: BoolQ for boolean question answering capabilities and the Jigsaw Toxicity dataset for content moderation assessment. This systematic investigation provides crucial insights into the trade-offs between model compression and knowledge preservation.

We have found that representation within model neurons is either preserved or improved in some cases of quantization. Confidence and Calibration analysis reveal that 4-bit quantization slightly improves the calibration of the model. Neuron's contribution highlights number of salient neurons remains low for full-precision model. In terms of activations, there is no major change in number of dead neurons. In terms of redundancy, Phi-2 and Llama-2-7b exhibit different patterns. As in the case of Phi-2 in full-precision had a higher number of neurons learning similar information, while in Llama-2-7b, there was a minor difference between highly correlation neuron pairs.

Overall, these findings contribute to our understanding of the quantization effect on LLM's internal representation and knowledge preservation. The results suggest that the effect of quantization could be dependent on the model's architecture and task. However, in our analysis, we don't see any major effect that could discourage the use of quantization as a practical approach for model deployment.

7 Limitations

Like all research, this study has certain limitations that should be considered when interpreting the results. Due to computational constraints, our experiments were limited to specific quantization configurations, model sizes, and datasets, which may not

fully capture the impact of quantization across all LLMs or in varied deployment settings. Expanding the study to include a broader set of models with similar architecture could help confirm some of the hypotheses of architecture effect within quantized and full-precision model.

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion*, 76(C):243–297.

Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. 2019. Post-training 4-bit quantization of convolution networks for rapid-deployment. *Preprint*, arXiv:1810.05723.

Yelysei Bondarenko, Riccardo Del Chiaro, and Markus Nagel. 2024. Low-rank quantization-aware training for llms. *Preprint*, arXiv:2406.06385.

Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. Zeroq: A novel zero shot quantization framework. *CoRR*, abs/2001.00281.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. 2024. Quip: 2-bit quantization of large language models with guarantees. *Preprint*, arXiv:2307.13304.

cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. 2017. Toxic comment classification challenge.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *Preprint*, arXiv:1905.10044.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in bert. *Preprint*, arXiv:2205.07237.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Exploiting redundancy in pre-trained language models for efficient transfer learning. *CoRR*, abs/2004.04010.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *Preprint*, arXiv:2208.07339.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. 2024. Bitdistiller: Unleashing the potential of sub-4-bit llms via self-distillation. *Preprint*, arXiv:2402.10631.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2017. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *CoRR*, abs/1702.03118.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers. *Preprint*, arXiv:2210.17323.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference. *CoRR*, abs/2103.13630.

R.M. Gray and D.L. Neuhoff. 1998. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. *Preprint*, arXiv:1706.04599.

Cong Guo, Jiaming Tang, Weiming Hu, Jingwen Leng, Chen Zhang, Fan Yang, Yunxin Liu, Minyi Guo, and Yuhao Zhu. 2023. Olive: Accelerating large language models via hardware-friendly outlier-victim pair quantization. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ISCA '23, New York, NY, USA. Association for Computing Machinery.

Sabit Hassan, Anthony Sicilia, and Malihe Alikhani. 2024. Active learning for robust and representative llm generation in safety-critical scenarios. *Preprint*, arXiv:2410.11114.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. 2024. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *Preprint*, arXiv:2401.18079.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2017. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Preprint*, arXiv:1712.05877.

Mojan Javaheripi and Sébastien Bubeck. 2023. Phi-2: The surprising power of small language models. https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/.

Jeonghoon Kim, Jung Hyun Lee, Sungdong Kim, Joonsuk Park, Kang Min Yoo, Se Jung Kwon, and Dongsoo Lee. 2023. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. *Preprint*, arXiv:2305.14152.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch. *Preprint*, arXiv:2009.07896.

Eldar Kurtić, Alexandre Marques, Mark Kurtz, and Dan Alistarh. 2024. We Ran Over Half a Million Evaluations on Quantized LLMs: Here's What We Found. https://neuralmagic.com/blog/we-ran-over-half-a-million-evaluations-on-quantized-llms-heres-what-we-found/.

Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. 2024. Evaluating quantized large language models. *Preprint*, arXiv:2402.18158.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for llm compression and acceleration. *Preprint*, arXiv:2306.00978.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. 2023a. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023b. Llm-qat: Data-free quantization aware training for large language models. *Preprint*, arXiv:2305.17888.

Vivek Miglani, Aobo Yang, Aram H. Markosyan, Diego Garcia-Olano, and Narine Kokhlikyan. 2023. Using captum to explain generative language models. *Preprint*, arXiv:2312.05491.

Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. 2020. Measuring calibration in deep learning. *Preprint*, arXiv:1904.01685.

Gunho Park, Baeseong Park, Minsub Kim, Sungjae Lee, Jeonghoon Kim, Beomseok Kwon, Se Jung Kwon, Byeongwook Kim, Youngjoo Lee, and Dongsoo Lee. 2024. Lut-gemm: Quantized matrix multiplication based on luts for efficient inference in large-scale generative language models. *Preprint*, arXiv:2206.09557.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *Preprint*, arXiv:2104.10350.

Irina Proskurina, Luc Brun, Guillaume Metzler, and Julien Velcin. 2024. When quantization affects confidence of large language models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1918–1928, Mexico City, Mexico. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *Preprint*, arXiv:1703.01365.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2023. Neurons in large language models: Dead, n-gram, positional. *Preprint*, arXiv:2309.04827.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. 2023. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1648–1665, Singapore. Association for Computational Linguistics.

Xiaoxia Wu, Cheng Li, Reza Yazdani Aminabadi, Zhewei Yao, and Yuxiong He. 2023. Understanding int4 quantization for transformer models: Latency speedup, composability, and failure cases. *Preprint*, arXiv:2301.12017.

Guoxuan Xia, Sangwon Ha, Tiago Azevedo, and Partha Maji. 2021. An underexplored dilemma between confidence and calibration in quantized neural networks. *CoRR*, abs/2111.08163.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 27168–27183. Curran Associates, Inc.

Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiaxiang Wu, and Bingzhe Wu. 2023. Rptq: Reorder-based post-training quantization for large language models. *Preprint*, arXiv:2304.01089.

Yuxuan Yue, Zhihang Yuan, Haojie Duanmu, Sifan Zhou, Jianlong Wu, and Liqiang Nie. 2024. Wkvquant: Quantizing weight and key/value cache for large language models gains more. *Preprint*, arXiv:2402.12065.

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. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

A Dataset Samples

Tables 5 and 6 show samples from BoolQ and Jigsaw Toxicity respectively.

Prompt	Gold
Respond with True or False only. do iran and afghanistan speak the same language Answer:	True
Respond with True or False only. do good samaritan laws protect those who help at an accident Answer:	True
Respond with True or False only. is windows movie maker part of windows essentials Answer:	True
Respond with True or False only. is confectionary sugar the same as powdered sugar Answer:	True
Respond with True or False only. is elder scrolls online the same as skyrim Answer:	False

Table 5: Samples from BoolQ dataset

Prompt	Gold
Given a sentence, respond with only 'True' if the sentence is toxic or else 'False'. Sentence: 'Red herrings to continue to excuse Breivik. Go away, nasty person.' Answer:	True
Given a sentence, respond with only 'True' if the sentence is toxic or else 'False'. Sentence: 'That is just plain BS, forgive the wording....if you look at that rule, it is meant for "empty" sites that redirect to a site with content. The original URL is to a site with content. User:kalimantan' Answer:	False
Given a sentence, respond with only 'True' if the sentence is toxic or else 'False'. Sentence: 'Iori Furukawa There's a name warrior at Iori Furukawa, thought you should check it out since you moved it. Thanks,' Answer:	False
Given a sentence, respond with only 'True' if the sentence is toxic or else 'False'. Sentence: 'YOU'RE BLOCKED FOR 700 DAYS. SHUSHING COW' Answer:	True
Given a sentence, respond with only 'True' if the sentence is toxic or else 'False'. Sentence: 'Bautista's Bat Flip Do you think it should be added in this article or in 2015 Toronto Blue Jays season article? If it should be added, then an image of the bat flip should be added as well. What do you think?' Answer:	False

Table 6: Samples from Jigsaw Toxicity dataset