Modeling Error Propagation Quantization In Neural Networks

Jasur Kiyamov[1,2(✉)], Aleksandr Dik[1], and Egor Savkov[1]

[1] St. Petersburg University, Saint Petersburg, Russia
z.kiyamov@spbu.ru, a.dik@spbu.ru
[2] Samarkand Branch of Tashkent University of Information Technologies, Samarkand, Uzbekistan

Abstract. This paper tackles the issue of quantization error propagation analysis in computational graphs of neural networks. To estimate the contribution of different operators to the resulting error, we suggest an analytical model based on Jacobians. Error accumulation along the operator chain is examined and an experimental comparison between the analytical and empirical errors is carried out. It is demonstrated that the analytical model accurately predicts the direction and magnitude of the error; however, substantial error accumulation is seen when the model passes through nonlinear and MLP blocks. We are able to localize important model components and optimize quantization schemes thanks to the obtained results.

Keywords: Error Propagation · Linear Approximation · Taylor Expansion · Quantization Error · Post-Training Quantization

1 Introduction

In computer vision, natural language processing, and other domains, modern neural networks are extensively employed. However, computational resources restrict their use in real-world systems, which makes quantization techniques important [1,2]. Error results from quantization's reduction of parameter and activation representation precision. The mechanisms by which quantization error spreads throughout the network are still not well understood, despite its extensive use [3]. Nonlinearities distort the structure, linear layers amplify the error as it spreads through the product of Jacobians, and normalization partially stabilizes but does not stop the error from building up.

2 Statement of the Problem

A neural network is considered as a composition of sequentially applied operators [4–6]:

$$X_{k+1} = F_k(X_k), \quad k = 0, \ldots, n-1$$

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16759, pp. 133–143, 2027.
https://doi.org/10.1007/978-3-032-30518-3_9

where $X_k \in \mathbb{R}^{d_k}$ is the input at the k-th step, F_k is the computational graph operator (linear transformation, normalization, nonlinearity, etc.). Quantization is represented as adding a perturbation to the inputs and parameters:

$$X_k^q = X_k + \Delta X_k, \quad W_k^q = W_k + \Delta W_k$$

where ΔX_k and ΔW_k are the quantization errors that arise due to discretization. The output of the quantized model is defined as:

$$X_{k+1}^q = F_k(X_k^q, W_k^q)$$

The error occurring at the (k)-th step is defined as:

$$\Delta X_{k+1} = X_{k+1} - X_{k+1}^q = F_k(X_k, W_k) - F_k(X_k^q, W_k^q)$$

Thus, the error depends on both the input disturbance and the parameter errors. The error propagates along the chain of operators, forming a sequence:

$$\Delta X_0 \to \Delta X_1 \to \cdots \to \Delta X_n$$

Of primary interest is the analysis of dynamics:

- accumulation of errors
- changes in its structure
- amplification or attenuation at different stages

The following tasks are solved within the framework of this work [7–9]:

1. Error estimation at each stage of calculations

$$\Delta X_k \quad \text{for all } k$$

2. Comparison of analytical and empirical error models
   - analytical: through Jacobians and linear approximation
   - empirical: direct calculation of the difference in outputs
3. Analysis of error propagation along a chain of operators
   - nature of accumulation
   - presence of critical areas
4. Identifying the most sensitive operators
   - linear transformations (MatMul)
   - nonlinearities (ReLU, SiLU, Softmax)
   - normalization (LayerNorm, RMSNorm)
5. Assessing the influence of model structure on quantization resistance It is required to build a model that:
   - describes the error propagation

$$\Delta X_{k+1} = \Phi_k(\Delta X_k, \Delta W_k)$$

– provides an exact approximation:

$$\Delta X_k^{analytical} \approx \Delta X_k^{empirical}$$

– allows you to quantify the contribution of each operator to the final error:

$$|\Delta X_n|$$

As a result of solving the problem, it is necessary to obtain an analytical model of error propagation

3 Linear Approximation (1st Order)

The first-order *linear* approximation is a fundamental tool for analyzing how small perturbations in the input propagate through a function. Given a differentiable function F(x), its behavior around a point x can be approximated using the first-order Taylor expansion:

$$F(x + \Delta x) \approx F(x) + J_F(x), \Delta x$$

where $J_F(x)$ is the Jacobian matrix of F at point x. This approximation assumes that higher-order terms are negligible, which is valid when the perturbation Δx is sufficiently small [10,11]. In the context of neural networks, the linear approximation provides an efficient way to estimate the effect of input noise, quantization errors, or parameter perturbations on the output. It allows modeling error propagation layer by layer, where each transformation contributes according to its local Jacobian. However, this approach ignores nonlinear interactions between perturbations, which may lead to significant inaccuracies in highly nonlinear operations such as normalization layers and activation functions.

Let F: $\mathbb{R}^n \to \mathbb{R}^m$ Taylor expansion:

$$F(X + \Delta X) = F(X) + J_F(X)\Delta X + O(\|\Delta X\|^2)$$

where

$$J_F(X) = \frac{\partial F}{\partial X}$$

Error:

$$\Delta Y = F(X + \Delta X) - F(X)$$

Linear model:

$$\Delta Y \approx J_F(X)\Delta X$$

Norm assessment:

$$\|\Delta Y\| \leq \|J_F(X)\| \cdot \|\Delta X\|$$

The Jacobian determines the sensitivity of the operator

3.1 Second-Order Approximation

The second-order approximation extends the linear model by incorporating curvature information of the function. For a twice-differentiable function $F(x)$, the second-order Taylor expansion around point x is given by [12,13]:

$$F(x + \Delta x) \approx F(x) + J_F(x), \Delta x + \frac{1}{2}, \Delta x^T H_F(x), \Delta x$$

where $J_F(x)$ is the Jacobian and $H_F(x)$ is the Hessian matrix of second derivatives. The additional quadratic term captures interactions between perturbations and reflects the local curvature of the function. In neural networks, this approximation provides a more accurate model of error propagation, especially for nonlinear components such as normalization layers (e.g., LayerNorm, RMSNorm) and activation functions (e.g., SiLU, Softmax). Unlike the first-order approximation, which assumes purely linear behavior, the second-order model accounts for how errors interact and amplify due to nonlinearity [14–16].

To improve accuracy:

$$F(X + \Delta X) = F(X) + J_F(X)\Delta X + \frac{1}{2}\Delta X^T H_F(X)\Delta X$$

where

$$H_F(X) = \frac{\partial^2 F}{\partial X^2}$$

Then:

$$\Delta Y \approx J_F(X)\Delta X + \frac{1}{2}\Delta X^T H_F(X)\Delta X$$

important for:

- softmax
- SiLU
- normalization

Propagation of error along the chain For the model:

$$X_{k+1} = F_k(X_k)$$

the error is spreading:

$$\Delta X_{k+1} \approx J_{F_k}(X_k)\Delta X_k$$

Iteratively:

$$\Delta X_n \approx \left(\prod_{k=0}^{n-1} J_{F_k}(X_k)\right) \Delta$$

Growth Rating:

$$\|\Delta X_n\| \leq \prod_{k=0}^{n-1} \|J_{F_k}\| \cdot \|\Delta X_0\|$$

possible:

– attenuation (stable)
– exponential growth (unstable)

Taking into account weights (parametric sensitivity) for the operator F(X,W):

$$\Delta Y \approx J_X \Delta X + J_W \Delta W$$

where:

$$J_X = \frac{\partial F}{\partial X}, \quad J_W = \frac{\partial F}{\partial W}$$

For the linear layer:

$$\Delta Y \approx W \Delta X + \Delta W X$$

Quantization model

$$X_q = X + \delta X, \quad W_q = W + \delta W$$

where:

$$\delta X \sim \mathcal{Q}(X) - X$$

It is often assumed:

$$\mathbb{E}[\delta X] = 0, \quad Var(\delta X) = \sigma^2$$

or uniform noise:

$$\delta X \sim \mathcal{U}(-\epsilon, \epsilon)$$

Error covariance model If errors are independent:

$$\mathrm{Cov}(\Delta Y) = J_X \Sigma_X J_X^T + J_W \Sigma_W J_W^T$$

For the linear layer:

$$\mathrm{Cov}(\Delta Y) \approx W \Sigma_{\Delta X} W^T + \Sigma_{\Delta W} R_X$$

where:

$$R_X = \mathbb{E}[X X^T]$$

the error depends on:

– scale structures
– login statistics

4 Description of the Experiment

This experiment aims to analyze how quantization errors propagate through neural network operators and to evaluate the accuracy of first-order and second-order approximations in modeling these errors [17,18]. The study considers perturbations introduced into both inputs and model parameters and compares the estimated error with the true observed deviation (Table 1).

Table 1. Jacobians of Common Operators

Operator	Function $F(x)$	Jacobian J_F
Matrix Multiplication/Linear Layer	$y = Wx + b$	$J = W$
Addition	$y = x + r$	$J = I$
Subtraction	$y = x - r$	$J = I$
Element-wise Multiplication	$y = x \odot m$	$J = \text{diag}(m)$
Element-wise Division	$y = x \odot \frac{1}{d}$	$J = \text{diag}\left(\frac{1}{d}\right)$
Transpose	$y = Px$	$J = P$
Reshape	$y = \text{reshape}(x)$	$J = P$
Softmax	$y = \text{softmax}(x)$	$J = \text{diag}(y) - yy^T$
ReLU	$y = \max(0, x)$	$J = \text{diag}(\mathbf{1}_{x>0})$
SiLU/Swish	$y = x\sigma(x)$	$J = \sigma(x) + x\sigma(x)(1 - \sigma(x))$

1. The input tensor is specified (Fig. 1):

$$X_0$$

2. An error is added to the input and/or parameters:

$$X_k^q = X_k + \Delta X_k, \quad W_k^q = W_k + \Delta W_k$$

where $\Delta X_k, \Delta W_k$ are quantization errors.

For each operator F_k the following are calculated:

$$X_{k+1} = F_k(X_k, W_k)$$
$$X_{k+1}^q = F_k(X_k^q, W_k^q)$$

Calculating the error at each step

$$\Delta X_{k+1} = X_{k+1} - X_{k+1}^q$$

Calculation of metrics

– absolute error:

$$|\Delta X_k|_2$$

– relative error:

$$\frac{|\Delta X_k|_2}{|X_k|_2}$$

– maximum error:

$$\max |\Delta X_k|$$

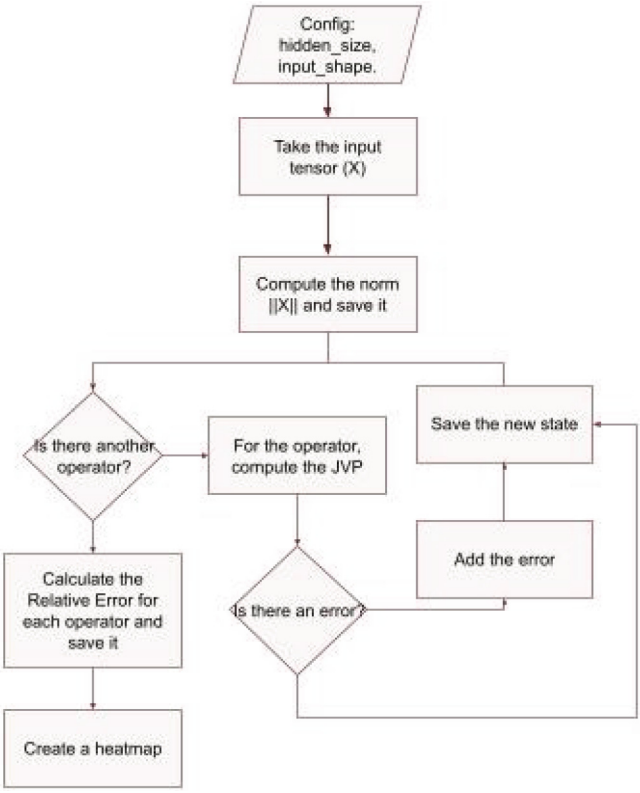

Fig. 1. Error Propagation Modeling.

5 Model Accuracy

In this section, we experimentally validate the suggested error propagation model on a series of operators, such as the Softmax function, a linear transformation (MatMul), nonlinearity (ReLU), and normalization (LayerNorm).

The findings shown in Fig. 2 illustrate how the relative output error varies with the degree of input disturbance and the operator's location along the computational chain. The error is found to increase as it moves through the model, but the type of operator greatly influences the type of increase. The linear transformation (MatMul) contributes most to the increase in error. The input error is scaled by the weight matrix, which explains this:

$$\Delta Y \approx W \Delta X$$

The primary source of error accumulation in the model is linear layers since the error increases proportionately to the weight norm. An extra error transformation related to zeroing out negative components is introduced by the ReLU operator. In contrast to the linear layer, this modifies the error's structure but

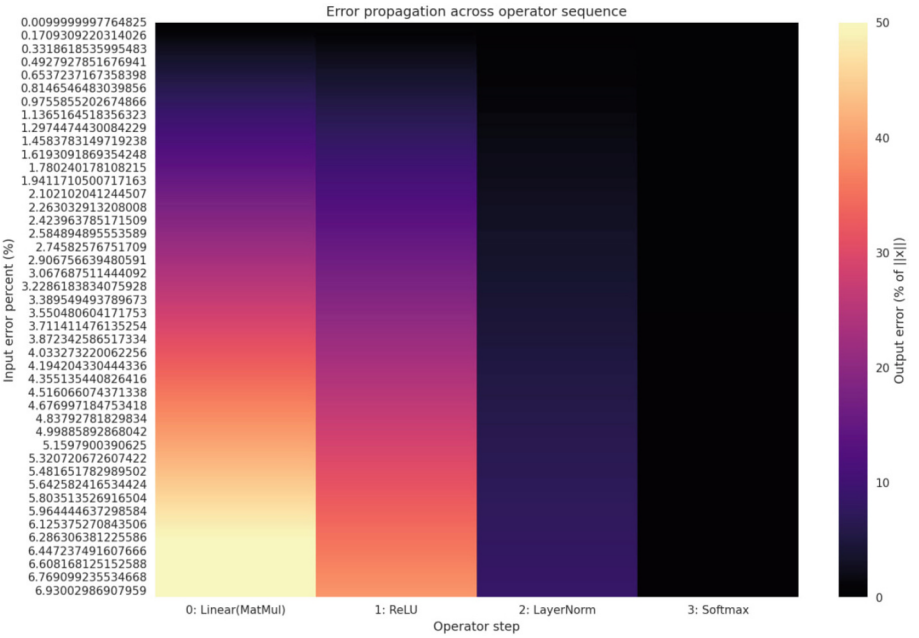

Fig. 2. Accuracy verification for the chain.

does not greatly increase its magnitude. Therefore, nonlinearity primarily affects the error's distribution rather than its scale.

The error is stabilised by LayerNorm, which limits further growth and lessens its spread. The normalisation of input activations is the cause of this:

$$y = \frac{x - \mu}{\sigma}$$

Normalisation reduces sensitivity to input disturbances and equalises the scale of errors. The Softmax function shows the least amount of error growth contribution. Despite being nonlinear, it simply redistributes the error among the output vector's components rather than amplifying it. This demonstrates how resilient Softmax is to minor input perturbations. The following conclusions can be made based on the experimental results:

- The linear transformation (MatMul) is the primary cause of the error's growth;
- Nonlinear functions alter the error's structure but do not significantly amplify it;
- Normalisation prevents errors from spreading;
- Error accumulation does not occur from Softmax.

Normalisation and nonlinear functions serve as stabilisers, but linear transformations play a major role in determining error propagation in the operator

chain. This enables us to pinpoint important model elements and utilise the outcomes to improve quantisation strategies. The obtained values are narrow and concentrated, according to an analysis of the error distribution. The quantisation error exhibits predictable behaviour since the majority of observations are concentrated within a narrow range. Rare but critically large deviations do not occur during the computational process, as indicated by the lack of significant outliers. This indicates that even with quantisation, the model does not display unstable operating modes. The fact that the error distribution is stable and well-controlled attests to the accuracy of the analytical model employed and its suitability for evaluating the effects of quantisation. The scale of the output signal and the quantisation error's magnitude did not significantly correlate, according to the analysis. The relative error does not significantly change when the output vector $|\alpha Y|$'s norm is changed. This suggests that the operator's structural characteristics and the input data's statistics, rather than the signals' absolute magnitude, are what ultimately determine the error. Specifically, the following are important:

- roperties of weight matrices (norm, spectrum)
- type of nonlinearity
- distribution of input activations

The quantization error is structurally determined and is not a simple consequence of the scaling of the output values.

6 Influence of Second-Order Terms

The analysis shows that the accuracy of the error approximation depends significantly on the magnitude of the perturbation. For small quantization errors, the first-order linear model:

$$\Delta Y \approx J_F(X)\Delta X$$

provides high accuracy and adequately describes the error propagation. However, as the magnitude of the disturbance increases, the contribution of the second-order terms becomes significant (Table 2):

$$\Delta Y \approx J_F(X)\Delta X + \frac{1}{2}\Delta X^T H_F(X)\Delta X$$

Ignoring the second order in this case leads to a noticeable discrepancy between the analytical and empirical error. The greatest second-order influence is observed for operators with pronounced nonlinearity and high curvature, such as:

- Softmax, due to exponential dependence
- SiLU (Swish), due to smooth nonlinearity
- LayerNorm, due to dependence on input statistics

Table 2. Modeling Second-Order Errors Across Different Operators and Noise Levels

Operator	Error (%)	True	1st Order (%)	2nd Order (%)	Improvement (%)
LayerNorm					
LayerNorm	0.01	0.000789	0.124596	0.046161	0.078435
LayerNorm	0.10	0.007893	0.061336	0.004509	0.056827
LayerNorm	1.00	0.078921	0.508784	0.004902	0.503882
LayerNorm	2.00	0.157804	1.016194	0.019512	0.996682
LayerNorm	5.00	0.394055	2.540733	0.121827	2.418906
RMSNorm					
RMSNorm	0.01	0.000794	0.057812	0.040447	0.017365
RMSNorm	0.10	0.007941	0.050564	0.003842	0.046722
RMSNorm	1.00	0.079401	0.510653	0.004956	0.505696
RMSNorm	2.00	0.158762	1.021473	0.019730	1.001744
RMSNorm	5.00	0.396443	2.555264	0.123285	2.431979
SiLU					
SiLU	0.01	0.000522	0.069817	0.043226	0.026591
SiLU	0.10	0.005219	0.051649	0.004097	0.047551
SiLU	1.00	0.052193	0.511570	0.002566	0.509004
SiLU	2.00	0.104397	1.022947	0.010072	1.012874
SiLU	5.00	0.261079	2.555468	0.062831	2.492637
Softmax					
Softmax	0.01	1.8×10^{-5}	0.097104	0.057421	0.039683
Softmax	0.10	0.00018	0.066108	0.005894	0.060214
Softmax	1.00	0.001801	0.647290	0.003427	0.643863
Softmax	2.00	0.003603	1.294104	0.013413	1.280691
Softmax	5.00	0.009017	3.231060	0.083720	3.147340

Taking the second order into account is necessary for correct modeling of errors under conditions of significant disturbances.

The study demonstrates that an analytical model based on linear approximation provides highly accurate quantization error prediction. Empirical results confirm that both the magnitude and direction of the error are in good agreement with the analytical estimates. Analysis of the propagation of the error along the chain of operators demonstrates its cumulative nature : as it passes through the model, the error is successively transformed and increases, and the growth occurs unevenly. It was found that the main contribution to the degradation of accuracy is made by MLP blocks, where linear transformations with large weights lead to a significant increase in error. Nonlinear operators (such as ReLU, SiLU, and Softmax) not only change the structure of the error, but in some cases can amplify it due to their high sensitivity to input disturbances.

It is shown that despite the change in scale, the direction of the error remains the same, which is confirmed by the high value of cosine similarity between the analytical and empirical errors.

7 Conclusion

This paper proposes an effective analytical model for studying the propagation of quantization error in neural networks at the level of individual operators and their composition. The model is based on linear approximation using Jacobians and is supplemented with statistical error analysis. It is shown that first-order linear approximation provides high accuracy for small perturbations and agrees well with empirical results. Furthermore, it is established that the quantization error is structured and not random noise, but is determined by the properties of the operators and the distribution of the input data. An analysis of error propagation along the operator chain revealed key nodes where the greatest error amplification occurs. These nodes include attention blocks and MLPs, which play a decisive role in the degradation of model accuracy after quantization.

References

1. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
3. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
4. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: ICCV (2015)
5. Hendrycks, D., Gimpel, K.: Gaussian Error Linear Units (GELUs). arXiv:1606.08415 (2016)
6. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv:1607.06450 (2016)
7. Zhang, B., Sennrich, R.: Root mean square layer normalization. In: NeurIPS (2019)
8. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
9. Higham, N.J.: Accuracy and stability of numerical algorithms. In: SIAM (2002)
10. Banner, R., Nahshan, Y., Soudry, D.: Post training 4-bit quantization of convolutional networks for rapid-deployment. In: NeurIPS (2018)
11. Jacob, B., et al.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: CVPR (2018)
12. Nagel, M., et al.: Data-free quantization through weight equalization and bias correction. In: ICCV (2019)
13. Frantar, E., et al.: GPTQ: accurate post-training quantization for generative pre-trained transformers. arXiv:2210.17323 (2022)
14. Xiao, G., et al.: SmoothQuant: accurate and efficient post-training quantization for large language models. In: ICML (2023)
15. Dettmers, T., et al.: LLM.int8(): 8-bit matrix multiplication for transformers at scale.' arXiv:2208.07339 (2022)
16. Krishnamoorthi, R.: Quantizing deep convolutional networks for efficient inference: a whitepaper. arXiv:1806.08342 (2018)
17. Martens, J.: New insights and perspectives on the natural gradient method. In: JMLR (2020)
18. Papyan, V.: The prevalence of neural collapse during the terminal phase of deep learning training. In: PNAS (2020)