The Orthogonality of Weight Vectors: The Key Characteristics of Normalization and Residual Connections

Zhixing Lu, Yuanyuan Sun*, Zhihao Yang, Qin Zhou, Hongfei Lin
College of Computer Science and Technology, Dalian University of Technology, Dalian, 116081, Liaoning, China
{syuan,yangzh,hflin}@dlut.edu.cn, {lzx2021, 2428074877}@mail.dlut.edu.cn

Abstract

Normalization and residual connections find extensive application within the intricate architecture of deep neural networks, contributing significantly to their heightened performance. Nevertheless, the precise factors responsible for this elevated performance have remained elusive. Our theoretical investigations have unveiled a noteworthy revelation: the utilization of normalization and residual connections results in an enhancement of the orthogonality within the weight vectors of deep neural networks. This, in turn, induces the Gram matrix of neural network weights to exhibit a pronounced tendency towards strict diagonal dominance, thereby amplifying the neural network’s capacity for feature learning. Meanwhile, we have designed the parameters independence index (PII) to precisely characterize the orthogonality of parameter vectors. In tandem with our theoretical findings, we undertook empirical validations through experiments conducted on prevalent network models, including fully connected networks (FNNs), convolutional neural networks (CNNs), Transformers, pre-trained language models (PLMs) and large language models (LLMs) composed of Transformers. Finally, we have found that a fine-tuning technique (LoRA) preserves the orthogonality of parameter vectors, a revelation that carries importance within the framework of fine-tuning techniques for LLMs.

1 Introduction

Normalization techniques, such as batch normalization [Ioffe and Szegedy, 2015] and layer normalization [Ba et al., 2016], belong to a class of widely used techniques for enhancing the training capabilities of deep neural networks (DNNs). In general, normalization can be attributed to several advantageous properties for DNNs, such as reducing the network’s dependence on initial parameter values [De and Smith, 2020] [Shao et al., 2020], improving the convergence speed of the network [Karakida et al., 2019], auto-tuning of learning rates [Arora et al., 2018], and smoothing the loss landscape [Yong et al., 2020]. The residual connection [He et al., 2016] is a technique that involves adding skip connections in the middle of network layers, aiming to alleviate the gradient vanishing and exploding issues encountered during the training of DNNs. Furthermore, residual connections can significantly enhance training stability and generalization accuracy. Consequently, it has become an essential component in various domains, including biomedical imaging and generative models like U-Net [Ronneberger et al., 2015]. In natural language processing, it is exemplified by the Transformer [Vaswani et al., 2017], and in reinforcement learning, its effectiveness is demonstrated by AlphaGo Zero [Silver et al., 2017].

In recent years, the combination of normalization and residual connections has been widely applied in DNNs. Almost all state-of-the-art models in recent years have adopted the combination of normalization and residual connections [Touvron et al., 2023] [Chowdhery et al., 2023] [Achiam et al., 2023]. Particularly, the Transformer [Vaswani et al., 2017] has become the fundamental building block of the most powerful large language models (LLMs) currently available. So, what is the origin of this effect?

In addressing this question, scholars have conducted theoretical studies. De and Smith proposed that batch normalization tends to bias residual blocks towards the identity function in DNNs [De and Smith, 2020]. Balduzzi et al. [Balduzzi et al., 2017] and Yang et al. [Yang et al., 2019] argued that including identity skip connections and batch normalization layers on the residual branch in ResNets helps maintain correlations between various mini batches in deep networks. Liu et al. [Liu et al., 2020] found that normalization can effectively address the problem of spurious gradient exploding or vanishing correlated with the depth of models resulting from residual connections. However, the origin of this effect is still poorly understood.

In DNNs that employ the combination of normalization and residual connections, such as the Transformer and pre-trained models (PLMs) composed of the Transformer, it has been observed that the parameter vectors exhibit good orthogonality, meaning the cosine similarity between any two row vectors (or column vectors) of the parameter matrix is close to zero. Based on feature learning theory [Radhakrishnan et
al., 2022][Beaglehole et al., 2023], the Gram matrix[Tanton, 2005] of network parameters is proportional to the average gradient outer product with respect to patches of the input to that layer. The Gram matrix and the average gradient outer product contain the same feature information[Radhakrishnan et al., 2022][Beaglehole et al., 2023]. In our perspective, when the column vectors of network parameters are orthogonal, the Gram matrix becomes a diagonal matrix with diagonal elements being all the eigenvalues. Consequently, at this point, the Gram matrix has the highest correlation with the average gradient outer product, indicating the best fitting capability of the trained network.

Moreover, some related studies have also demonstrated that the orthogonality of vectors can enhance the fitting capability of DNNs[Xiao et al., 2018][Wang et al., 2020][Li et al., 2019][Huang et al., 2018]. In generally, DNNs are typically initialized using random or approximately random methods[Glorot and Bengio, 2010][LeCun et al., 2002], ensuring the orthogonality of initial network parameter vectors. However, little attention is paid to incorporating optimization or regularization conditions to maintain the orthogonality of parameter matrices during the training process. The factors responsible for preserving the orthogonality of parameter vectors are not well understood. To address this issue, we conducted pertinent research, and the key contributions of our study are summarized as follows.

- We propose the parameters independence index (PII) to assess the orthogonality between network parameter vectors, where a lower PII indicates better overall orthogonality of parameter vectors.
- We deduce that the combination of normalization and residual connections can enhance the orthogonality of parameter vectors, improve the feature learning capability of DNNs, and consequently, enhance the performance of DNNs. Then, we validated the theoretical correctness on network models including FNNs, CNNs, Transformers, PLMs and LLMs composed of the Transformer.
- We discovered that the LLM fine-tuning technique based on LoRA[Hu et al., 2021] also maintains the orthogonality of parameter vectors during the adjustment of network parameters. This discovery holds significance in the context of LLMs fine-tuning techniques and also prove the universality of theory in this study. The supplementary materials and all implementation codes are available on the Github¹.

2 Related Works

Normalization and residual connections are widely applied in DNNs, and researchers have conducted relevant theoretical research to explore the mechanisms through which they enhance network performance.

2.1 Normalization Techniques

Normalization comes in various forms, and our research primarily focuses on the widely used batch normalization and layer normalization. Therefore, we will highlight relevant studies related to these techniques. For example, batch normalization divides the optimization task into optimizing the length and direction of parameters separately[Kohler et al., 2019]. Batch normalization orthogonizes representations in deep random networks[Daneshmand et al., 2021]. Batch normalization is proven to avoid rank collapse for randomly initialized deep networks[Daneshmand et al., 2020]. In contrast to batch normalization, layer normalization overcomes the dependency on batch size, and empirical evidence shows that it is more suitable for recurrent neural networks(RNNs) and natural language processing tasks[Ba et al., 2016].

2.2 Residual Connections

Theoretical research on residual connections is also a hot topic in the theory of the DNN. Katsman et al. extended residual connections to general Riemannian manifolds in a geometrically principled manner[Katsman et al., 2023]. Orhan et al. discovered that residual connections can eliminate singularities[Orhan and Pitkow, 2017]. By employing principles from linear algebra and random matrix theory, researchers explore the reasons behind the enhanced ease of optimization and improved generalization exhibited by DNNs with residual connections[Oyedotun et al., 2022].

2.3 The Combination of Normalization and Residual Connections

Regarding the impact of the combination of normalization and residual connections on DNNs, scholars have also conducted research. For instance, studies indicate that residual connections and batch normalization can enhance data separability[Furusho and Ikeda, 2019]. Batch normalization reduces the scale of hidden activations in the residual branch by approximately the square root of the network depth[De and Smith, 2020]. Furusho and Ikeda evaluated the generalization gap and the convergence rate to demonstrate why skip connections and batch normalization improve performance[Furusho and Ikeda, 2020].

Researchers have extensively studied normalization and residual connections, offering various explanations for how they can enhance the capabilities of DNNs. However, the internal mechanisms through which they profoundly impact DNNs are far from being fully understood.

3 Background and Preliminaries

3.1 Network Description

For a FNN, the forward propagation process is as follows:

\[
z^l = W^l x^{l-1} + b^l, x^l = f\left(z^l\right)
\]

Where \(x^{l-1}\) represents the output of the preceding layer, serving as the input to the current layer. \(f\) denotes the activation function, \(z^l\) is the pre-activation value of the neuron, \(W^l\) and \(b^l\) are the weights and bias of the network, with our focus being on \(W^l\), temporarily disregarding \(b^l\). Our research primarily encompasses several widely used activation functions, including ReLU[Nair and Hinton, 2010], GELU[Hendrycks and Gimpel, 2016], Sigmoid[DeMaris, 1995], and Tanh[Fan, 1986].
Let $x' = BN (x'^{-1} + f (z'))$ \hfill (2)

Then $x'_i = \frac{x'_i}{\sigma} - \mu$.

### 3.2 Feature Learning

$G_{W^l} = (W^l)^T W^l$ is the Gram matrix of the weight $W^l$ in the l-th layer of the DNN. Research shows that the structure of $W^l$ can characterize how features are updated in a trained DNN\cite{Radhakrishnan et al., 2022}\cite{Baeogle et al., 2023}. For a trained DNN, studying the importance of a particular feature, a natural approach is to examine the amplitude of changes in that feature under perturbations. The amplitude can be calculated through the outer product of the gradients of the neural network with respect to the input, that is $(\nabla_{x'} f (x)) (\nabla_{x'} f (x))^T$, where $\nabla_{x'} f (x)$ represents the gradient of the DNN $f$ at point $x$. $(W^l)^T W^l$ has the following relationship with $(\nabla_{x'} f (x)) (\nabla_{x'} f (x))^T$.

**Theorem 1.** \cite{Radhakrishnan et al., 2022} Let $f$ be an l-hidden layer network with ReLU activation, suppose we sample weight $a_{k'}$, $W^l$ for $l > 1$ in an i.i.d manner so that $E \left[ a_{k'}^2 \right] = 1$, $E \left[ W_{l,k'}^2 \right] = 1$, $E \left[ a_{k'} \right] = 0$ and $E \left[ W_{l,k'} \right] = 0$. Suppose $W^1$ is fixed and arbitrary. Let $(x_i, y_i)_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$. If $x \sim N (0, I_d)$, then

$$
\frac{1}{k_1} (W^l)^T W^l = E_x \left[ \lim_{k_2, \ldots, k_l \to \infty} E_a \left[ \nabla_{x'} f (x) \nabla_{x'} f (x)^T \right] \right] \hfill (4)
$$

$N (0, I_d)$ is the standard normal distribution, $k_1$ is the number of neurons in the first layer, and $W^l$ is the network parameter of the first layer. The results of Theorem 1 can be naturally extended to other network layers. Furthermore, for DNNs with finite width

$$(W^l)^T W^l \propto \frac{1}{n} \sum_{p=1}^n \nabla f_i (h_i (x_p)) \nabla f_i (h_i (x_p))^T \hfill (5)$$

where $h_i (x)$ is the input into layer $i$, $\nabla f_i (h_i (x_p))$ denotes the gradient of $f_i$ with respect to $h_i (x_p)$. Therefore, the outer product of gradients of the DNN with respect to the input data $h_i (x_p)$ is proportional to the Gram matrix of the weights.

In our study, we found that if $G_{W^l} = (W^l)^T W^l$ is a diagonal matrix or diagonally dominant matrix, the average outer product of gradients is also a diagonal or diagonally dominant matrix. The proportional relationship between them is mainly concentrated on the diagonal elements. In this case, the network can more fully learn the feature information of the input data. Therefore, if the Gram matrix of the network tends to be a diagonally dominant matrix after training, it will enhance the DNN’s ability to fit the training data.

### 3.3 Gram Matrix and Parameters Independence Index

For the weight $W^l = \{w^l_1, w^l_2, \ldots, w^l_n\} \in \mathbb{R}^{m \times n}$, $w^l_i = \{w^l_{i1}, w^l_{i2}, \ldots, w^l_{im}\}^T$, $i \in \{1, 2, \ldots, n\}$, the Gram matrix\cite{Tanton, 2005} of $W^l$ is $G_{W^l} = (W^l)^T (W^l)$ with elements $g_{ij} = \langle w^l_i, w^l_j \rangle$, where $\langle \cdot \rangle$ is the inner product. The diagonal elements of $G_{W^l}$ are $\langle w^l_i, w^l_i \rangle$, $i \in 1, 2, \ldots, n$, and the off-diagonal elements are $\langle w^l_i, w^l_j \rangle$, $i \neq j$. Therefore, when $G_{W^l}$ is a diagonal matrix, for any $i$ and $j$, it is necessary to satisfy $\langle w^l_i, w^l_j \rangle = 0$ for $i \neq j$. In practical applications, cosine similarity is commonly used to characterize the angle between vectors. When the cosine similarity is 0, the vectors are orthogonal to each other. When the cosine similarity is 1 or -1, the angle between vectors is 0 or 180 degrees, respectively. We focus on the orthogonality between vectors, where a cosine similarity closer to 0 indicates a closer approach to orthogonality. Therefore, we assess the orthogonality between vectors by computing the absolute value of the cosine similarity. In most cases, the parameter size of the DNN is large. To evaluate the overall orthogonality between parameter vectors, we introduce the PII.

**Definition 1.** For the weight matrix $W^l = \{w^l_1, w^l_2, \ldots, w^l_n\} \in \mathbb{R}^{m \times n}$,

$$PII = \text{average} \left( \text{abs} \left( \frac{w^l_i \cdot w^l_j}{\|w^l_i\| \|w^l_j\|} \right) \right), \hfill (6)$$

for all $i, j \in n, i \neq j$, where $\text{abs}(\cdot)$ denotes the absolute value, and $\text{average}(\cdot)$ calculates the average of all data. Specifically, the PII is the average absolute value of cosine similarities between any two row vectors in the parameter matrix. The PII ranges from $[0, 1]$, where PII=0 indicates that any two row vectors in the parameter matrix are independent, and evidently, $G_{W^l}$ is a diagonal matrix in this case. PII=1 signifies that the vectors are linearly dependent.

We conducted a study on the orthogonality of vectors by examining vector angles. Negative cosine similarity values lack practical significance. Therefore, we used the absolute value of cosine similarity to represent vector orthogonality. Concurrently, we calculated the cosine similarity between all column vector pairs in a parameter matrix. For an $M \times N$ matrix, there are $N^2$ total results, which is a substantial quantity. We need an indicator that can holistically reflect the orthogonality of the parameter vectors. We mean, or expected value, can represent the global orthogonality of the matrix vectors. Moreover, using the absolute value function avoids canceling out between positive and negative cosine similarities. This ensures the mean accurately reflects the overall orthogonality of the parameter vectors.

### 4 Main Results

#### 4.1 Conformal Capability of The Optimizer

Theorem 1 stipulates that the network parameters must form a matrix with zero mean and unit variance. Our investigation reveals that even when the variance approximates 1, it does not significantly alter the proportional relationship between
and the average gradient outer product. Currently prevalent network initialization methods, such as orthogonal initialization, Xavier initialization [Glorot and Bengio, 2010], and LeCun initialization [LeCun et al., 2002], all adhere to this condition. The incorporation of normalization techniques in network layers, such as batch normalization and layer normalization, can satisfy the condition for each layer’s input \( x \sim N(0, I_d) \). Consequently, it is advantageous for the network’s training method to possess a certain conformal capability, ensuring that the angles between parameter vectors do not undergo significant changes. Subsequent research suggests that the SGD optimizer exhibits superior conformal capabilities when compared to the Adam optimizer.

**Theorem 2.** The SGD optimizer outperforms the Adam optimizer in terms of conformal capability among parameter vectors.

The proof is provided in the appendix. Here, we offer an intuitive understanding. DNNs typically employ mini-batch sample data for gradient computation and parameter updates. Let the training set be \( \mathcal{D} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N} \), where \( N \) is the batch size. Each sample \( x^{(n)} \) is the input of the network, resulting in the network output \( \hat{y}^{(n)} \). The loss on the dataset \( \mathcal{D} \) is given by:

\[
R(W, b) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(y^{(n)}, (\hat{y}^{(n)}))
\]  

When using the SGD to update parameters,

\[
w_{l,t+1} = w_{l,t} - \eta_k \nabla w_{l,t}
\]

\( \eta_k \) is the learning rate, \( w_{l,t} \) is the parameter vector and \( \nabla w_{l,t} \) the gradient with respect to \( w_{l,t} \). When updating parameters using the Adam,

\[
w_{l,t+1} = w_{l,t} - \alpha_t \frac{m_{l,t+1}}{\|m_{l,t+1}\| + \epsilon} b_{l+1}
\]

\[
m_{l,t+1} = \beta_1 m_{l,t} + (1 - \beta_1) \nabla w_{l,t}
\]

\[
v_{l,t+1} = \beta_2 v_{l,t} + (1 - \beta_2) \left( \nabla w_{l,t} \right)^2
\]

\[
b_{l+1} = \sqrt{1 - \beta_2^{l+1}} - \sqrt{1 - \beta_1^{l+1}}
\]

\( m_{l,t+1} \) and \( v_{l,t+1} \) are momentums obtained by the \( \nabla w_{l,t} \). Therefore, when the gradients of the network are the same, the SGD optimizer is closer to a translation transformation of the parameter vector \( w_{l,t} \). Translation of a vector does not change the angle between vectors before and after the transformation, making it an conformal mapping. Hence, SGD possesses better conformal capabilities than the Adam optimizer.

To validate Theorem 2, we conducted the following experiments about the FNN and CNN. The datasets are MNIST, CIFAR-10 and CIFAR-100. Results of CIFAR-100 are presented in the appendix. We computed the PII of the parameter matrix under both SGD and Adam optimizer. A PII closer to the initial value indicates better conformal capability of the optimizer.

<table>
<thead>
<tr>
<th>Optimizers</th>
<th>PII Layer_1↑</th>
<th>PII Layer_2↓</th>
<th>ACC %</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGD</td>
<td>1.0/0.996</td>
<td>0.035/0.049</td>
<td>93.59</td>
</tr>
<tr>
<td>Adam</td>
<td>1.0/0.432</td>
<td>0.035/0.153</td>
<td>98.22</td>
</tr>
<tr>
<td>AdamW</td>
<td>1.0/0.411</td>
<td>0.035/0.114</td>
<td>97.92</td>
</tr>
<tr>
<td>RMSprop</td>
<td>1.0/0.766</td>
<td>0.035/0.234</td>
<td>97.94</td>
</tr>
</tbody>
</table>

Table 1: Conformal capabilities of optimizers in FNNs.

<table>
<thead>
<tr>
<th>Optimizers</th>
<th>PII Layer_1↑</th>
<th>PII Layer_2↓</th>
<th>PII Layer_3↓</th>
<th>ACC %</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGD</td>
<td>1.0/0.795</td>
<td>0.070/0.195</td>
<td>0.050/0.149</td>
<td>84.04</td>
</tr>
<tr>
<td>Adam</td>
<td>1.0/0.328</td>
<td>0.070/0.217</td>
<td>0.050/0.242</td>
<td>85.94</td>
</tr>
<tr>
<td>AdamW</td>
<td>1.0/0.337</td>
<td>0.070/0.208</td>
<td>0.050/0.237</td>
<td>85.96</td>
</tr>
<tr>
<td>RMSprop</td>
<td>1.0/0.37</td>
<td>0.070/0.230</td>
<td>0.050/0.207</td>
<td>84.80</td>
</tr>
</tbody>
</table>

Table 2: Conformal capabilities of optimizers in CNNs.
4.2 The Conformal Capability of The Optimizer with Normalization and Residual Connections

The earlier results indicate that, under the same conditions, the conformal capability of the SGD is the best. However, the SGD optimizer tends to trap DNNs in local optimum, preventing the network from achieving a better testing capabilities. Optimizers like Adam and AdamW overcome this issue by introducing momentum mechanisms. However, previous research suggests that these optimizers have poor conformal capability. The subsequent research suggests that adding normalization and residual connections in the middle of network layers can enhance the conformal capability of all the optimizers.

Additionally, our research has found that only normalization or residual connections cannot guarantee a definite improvement in the conformal capability of optimizers. Due to space constraints, relevant studies are presented in the appendix.

**Theorem 3.** The combination of normalization and residual connections can enhance the conformal capability of the optimizer.

The proof is provided in the appendix. Here, we also offer an intuitive understanding. We take batch normalization as an example, the result are the same for layer normalization. When DNNs do not incorporate batch normalization and residual connections, the gradient of the parameter $W^l$ is

$$
\nabla W^l = f'(z') \odot \left( W^{l+1} \sigma (z^{l+1}) \right) (x^{l-1})^T = \delta^l (x^{l-1})^T
$$

where $\odot$ represents element-wise multiplication, and $\delta^l = f'(z') \odot \left( W^{l+1} \sigma (z^{l+1}) \right)$ has the same form across different network layers, we refer to $\delta^l$ as the error term of the neuron. After incorporating batch normalization and residual connections, the gradient of the parameter $W^l$ is

$$
\nabla W^l = \frac{1}{\sigma^l} f'(z') \odot P_{z^l} P_{x^{l+1}}
$$

$$
\left[ \frac{1}{\sigma^l} \left( 1 + f'(z^{l+1}) \right) \right] \odot W^{l+1} \odot P_{z^l} P_{x^{l+1+1}} (x^{l+1})
$$

$$
\cdot (x^{l-1})^T = \delta^l_{NR} (x^{l-1})^T
$$

$\delta^l_{NR}$ represents the error term of the $l$-th layer neuron after incorporating normalization and residual connections,

$$
(x^{l+1})^T = \frac{\partial R}{\partial x^l}. P_{z^l} (\cdot) \text{ and } P_{x^{l+1}} (\cdot) \text{ indicate the projection onto the directions of vectors } 1 \text{ and } x^l \text{ respectively. } \sigma^l \text{ and } \sigma^{l+1} \text{ represent the variances of neurons in the } l \text{-th and } l + 1 \text{-th layers, respectively. It can be observed that normalization and residual connections modify the error term of neurons. Compared to Equation (13), the addition of the vector 1 with the activation function prevents the problem of gradient vanishing in } \nabla W^l. \text{ In the computation of the error term } \delta^l_{NR}, \text{ division by the variance of each layer’s neurons is required. Therefore, for neurons with distributions having variances greater than 1 and sparser gradients, this approach reduces the sparsity of the gradients. In summary, the parameter gradient } \delta^l_{NR} (x^{l-1})^T \text{ in Equation (14) is a product of a unit-norm vector and a more uniformly distributed vector. During optimizer updates, this tends to map towards translations along the gradient direction, displaying better isotropy characteristics.}

4.3 Experiments

To validate the correctness of the theory, we conducted extensive experiments. The neural network comprise FNNs, CNNs, Transformers, PLMs, and LLMs. The datasets consist of MNIST, CIFAR-10, CIFAR-100, WikiText-2. For PLMs and LLMs, calculating the PII of their parameters is sufficient, and there is no need to retrain the models. Moreover, to confirm whether the fine-tuning process of LLMs preserves the orthogonality of parameter vectors, we conducted LoRA [Hu et al., 2021] fine-tuning on the Qwen model [Bai et al., 2023] and assessed changes of the PII before and after the fine-tuning.

Figure 2 illustrates the impact of normalization and residual connections on the orthogonality of neural network parameter vectors. The left side presents results without normalization and residual connections, while the right side showcases results after their inclusion. Both result sets indicate a significant enhancement in the orthogonality of network parameter row and column vectors through the incorporation of normalization and residual connections.

Figures 2 (a)-(d) depict the distribution of cosine similarity matrices, where a closer proximity of non-diagonal elements to yellow signifies improved vector orthogonality. The results suggest that, upon integrating normalization and residual connections, the values of non-diagonal elements in the matrix notably decrease, indicating a pronounced tendency toward orthogonality for parameter vectors. Figures 2 (e)-(h) portray the distribution of Gram matrices, with larger values in the diagonal elements indicating superior vector orthogonality. The results indicate that, following the introduction of normalization and residual connections, the Gram matrices of parameters distinctly trend towards being diagonally dominant. Particularly noteworthy are the outcomes in Figure (e) and Figure (f), where, despite minimal changes in the color of non-diagonal elements, the values of diagonal elements significantly increase. This suggests that, with the addition of normalization and residual connections, more data features converge in the diagonal elements, thereby enhancing the neural network’s capability for feature learning.

Moreover, Figure 2 demonstrates that the combination of
normalization and residual connections leads to a significant decrease in PII. PII accurately captures the orthogonality of parameter vectors. The combination of normalization and residual connections can simultaneously enhance the orthogonality of row vectors and column vectors, while the Gram matrix focuses primarily on column vectors. Therefore, in the subsequent experiments, we only present the results for column vectors.

In Table 3, the term ‘Original’ denotes the scenario where normalization and residual connections are excluded, while ‘Norm_Res’ signifies the inclusion of normalization and residual connections. The initial four rows of data in the Table present PIIIs for different activation functions under the Adam optimizer. The curves of ReLU and GELU functions display similar distributions, and their PIIIs before and after incorporating normalization and residual connections are also comparable. The last two rows of data showcase results for various optimizers under the same activation function. The SGD inherently exhibits a good conformal capability, evident in the already relatively low PIIIs even without normalization and residual connections. Nevertheless, the combination of normalization and residual connections still succeeds in further reducing its PII.

Table 3 illustrates that, under various activation functions and optimizer conditions, the integration of normalization and residual connections enhances the orthogonality of parameter vectors. Furthermore, as PII decreases, there is a general improvement in the overall test accuracy of the network. This implies that, for FNNs, improving the orthogonality of parameter vectors indeed enhances the network’s fitting capability.

Table 4 delineates the influence of normalization and residual connections on the PII and test accuracy of CNNs on CIFAR-10, results of CIFAR-100 are presented in the appendix. The network adheres to a ResNet, incorporating normalization and residual connections between each convolutional layer. The ‘Original’ outcomes represent the scenario where normalization and residual connections are omitted, consequently transforming the network into a fully connected convolutional network. We have extracted experimental outcomes from a single convolutional layer, with other network layers exhibiting comparable characteristics, a comprehensive set of experimental results is available in the appendix. The findings in Table 4 showcase that normalization and residual connections have the potential to improve both the orthogonality of parameter vectors and the test accuracy of CNNs.

The Transformer[Vaswani et al., 2017] stands out as one of the most prevalent neural network architectures in natural lan-

**Table 3:** The influence of the combination of normalization and residual connections on the orthogonality of parameter vectors and testing accuracy in FNNs.

<table>
<thead>
<tr>
<th>Optimizers</th>
<th>PII</th>
<th>PII ↓</th>
<th>ACC %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sigmoid/Adam</td>
<td>0.278</td>
<td>0.139</td>
<td>97.88/98.13</td>
</tr>
<tr>
<td>Tanh/Adam</td>
<td>0.136</td>
<td>0.118</td>
<td>98.01/98.14</td>
</tr>
<tr>
<td>ReLU/Adam</td>
<td>0.138</td>
<td>0.09</td>
<td>98.23/98.45</td>
</tr>
<tr>
<td>GELU/Adam</td>
<td>0.141</td>
<td>0.081</td>
<td>98.16/98.24</td>
</tr>
<tr>
<td>GELU/SGD</td>
<td>0.053</td>
<td>0.036</td>
<td>97.54/98.13</td>
</tr>
</tbody>
</table>

**Table 4:** The influence of the combination of normalization and residual connections on the orthogonality of parameter vectors and testing accuracy in CNNs.

<table>
<thead>
<tr>
<th>Optimizers</th>
<th>PII</th>
<th>PII ↓</th>
<th>ACC %</th>
</tr>
</thead>
<tbody>
<tr>
<td>GELU/Adam</td>
<td>0.196</td>
<td>0.176</td>
<td>95.34/98.66</td>
</tr>
<tr>
<td>GELU/SGD</td>
<td>0.206</td>
<td>0.155</td>
<td>85.48/90.22</td>
</tr>
</tbody>
</table>

Figure 2: The cosine similarity and Gram matrix with and without normalization and residual connections in FNNs.
language processing. Within the Transformer framework, both Pre-trained Language Models (PLMs) and Language Models (LLMs), which are composed of the Transformer, utilize normalization and residual connections to link attention and fully connected layers. In order to assess their influence on the orthogonality of parameter vectors in PLMs and LLMs, we conducted the following experiments.

The content of Table 5 consists of three sections. The PII for each section includes the attention’s queries, keys, and values matrices, the output layer parameter matrix of the attention, and the two parameter matrices of the fully connected layer. The first section presents the experimental results for the Transformer, which is composed of six Transformers and trained on the WikiText-2 dataset. The initial five PII are all close to zero, suggesting a high level of orthogonality among parameter vectors. However, the last PII is notably greater than zero. This discrepancy arises from the fact that, in our designed Transformer, the final fully connected layer in the last segment did not integrate the combination of normalization and residual connections. Consequently, this experiment confirms that it is the combination of normalization and residual connections that improves the orthogonality of parameter vectors in the Transformer.

The second section of Table 5 showcases the experimental outcomes for well-known PLMs. All of their PII are close to zero, signifying exceptional orthogonality among their parameter vectors. It’s noteworthy that the T5/Decoder omits a FFN layer, leading to empty data entries for Linear1.weight and Linear2.weight, denoted by ‘-’.

The third section of Table 5 showcases the experimental outcomes for LLMs. Their PII are smaller than those of PLMs, suggesting superior orthogonality in the parameter vectors of LLMs. Moreover, to investigate whether fine-tuning LLMs preserves vector orthogonality, we calculated the PII before and after LoRA fine-tuning. The fine-tuning experiment was conducted using the Qwen model, with the PII for Qwen in the table representing the result before LoRA fine-tuning, and the corresponding PII for LoRA indicating the result after fine-tuning. The findings indicate that LoRA adjusts network parameters while upholding the orthogonality of parameter vectors. Our study contributes to an enhanced understanding of LLM fine-tuning techniques.

5 Conclusion

The combination of normalization and residual connections is commonly employed in DNNs, and it often enhances the network’s stability and fitting capability. However, there has been a lack of theoretical explanations for this phenomenon. We have identified that the combination of normalization and residual connections maintains robust orthogonality between parameter vectors, leading to a Gram matrix that tends to be diagonal or diagonally dominant. The Gram matrix of parameters is proportionate to the outer product of the network’s gradient vector concerning input data. Diagonal or diagonally dominant matrices concentrate effective data features in the diagonal elements, thereby improving the neural network’s ability to learn data features. To validate the correctness of the theory, we conducted experiments on various network models suitable for the combination of normalization and residual connections. These models encompass FNNs, CNNs, Transformers, PLMs, LLMs, and networks fine-tuning based on LLMs. All the results confirm that the combination of normalization and residual connections significantly improves the orthogonality between parameter vectors after training.

Our research explains from the perspective of parameter vector orthogonality why normalization and residual connections can enhance the performance of DNNs, contributing to improve the interpretability of DNNs. Additionally, we have uncovered a novel insight that LoRA does not compromise the orthogonality of parameter vectors. This discovery holds significance in the context of LLMs fine-tuning techniques. Moving forward, our next phase of research will focus on investigating the influence of orthogonality on LLMs fine-tuning methods and exploring more efficient, resource-saving LLMs fine-tuning techniques.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (No. 2022YFC3301801), Natural Science Foundation of China (No. 62276043).

References

[Achiam et al., 2023] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni


