Escaping the Gradient Vanishing: Periodic Alternatives of Softmax in Attention Mechanism

Shulun Wang\textsuperscript{1} \hspace{1cm} Bin Liu\textsuperscript{2} \hspace{1cm} Feng Liu\textsuperscript{1}

slwang9353@163.com \hspace{1cm} xhyous@foxmail.com \hspace{1cm} fliu@bjtu.edu.cn

\textsuperscript{1}Department of Computer Science, Beijing Jiaotong University, Beijing, China
\textsuperscript{2}Key Laboratory of Deep Oil and Gas, China University of Petroleum (East China), Qingdao, China

\texttt{slwang.github.io/Period alternatives/}

Abstract

Softmax is widely used in neural networks for multiclass classification, gate structure and attention mechanisms. The statistical assumption that the input is normal distributed supports the gradient stability of Softmax. However, when used in attention mechanisms such as transformers, since the correlation scores between embeddings are often not normally distributed, the gradient vanishing problem appears, and we prove this point through experimental confirmation. In this work, we suggest that replacing the exponential function by periodic functions, and we delve into some potential periodic alternatives of Softmax from the view of value and gradient. Through experiments on a simply designed demo referenced to LeViT, our method is proved to be able to alleviate the gradient problem and yield substantial improvements compared to Softmax and its variants. Further, we analyze the impact of pre-normalization for Softmax and our methods through mathematics and experiments. Lastly, we increase the depth of the demo and prove the applicability of our method in deep structures.

![Figure 1: Visualization of the attention extracted from transformer blocks. Using Softmax, attention on the boundary is easier to be formed than that on the object, since the score of the boundary is moderate and is less likely to fall into the saturation area, especially in the early stages of training. With our unsaturated periodic alternatives, the attention on the object can be formed smoothly too. Compared to Softmax, our method only needs a short training to form satisfactory attention on the object.](image-url)

Preprint. Under review.
1 Introduction

Gradient Vanishing in Attention block: Since transformer is a pixel-wise process, the distribution of the input is more important than that for CNN. For CNN, the deviation caused by the value exceeding the expected range will be diluted in patches. The statistical assumption that the input is normal distributed supports the gradient stability of Softmax. However, there are always some values that exceed the expected value range, causing the gradient vanishing. This situation becomes worse in the attention mechanisms. The input of Softmax represents to the relationships of embedding (patches), and the distribution of the input should be variant for different images. That means in the attention mechanisms, there are always a part of the input stuck in the saturation area as shown in Figure 2, leading to the gradient vanishing and a long training.

Figure 2: Distribution of $Q \cdot K / d_{model}$ extracted from the transformer blocks. As we speculate, the value of $Q \cdot K / d_{model}$ is indeed not strictly normally distributed, and it might fall into the saturation area of Softmax.

Motivations: We are interested in the observation from the transformer-based models that the formation of attention corresponding to the objects often seems to lag behind that corresponding to the boundary, as shown in the first row of Figure 1. Attention can be formed on the boundary in the early stages of training, but slowly appear on objects only in the mid-late stages. However, object should be a more preferred position to put attention on, since there are no inductive bias such as translation equivariance and locality, and objects should be more important than boundary for transformer. By investigating the value of $Q \cdot K$, we find that the value corresponding to the object is larger than boundary, and is more likely to fall into the saturation area of Softmax. Therefore, we speculate that, object is indeed more important, but it is difficult to form attention on object, since the value of $Q \cdot K$ is too large and falls into the saturation area. In contrast, the $Q \cdot K$ value of boundary is moderate, so attention can be formed smoothly. We believe that this situation is one of the reasons why transformer needs a long training.

In this work, we suggest that replacing the exponential function by periodic functions, and we delve into some potential periodic alternatives of Softmax from the view of value and gradient. Through experiments on a simply designed demo referenced to LeViT, our method is proved to be able to alleviate the gradient problem and perform better compared to Softmax and its variants.

To summarize, the main contributions of this paper are:

- Explore the gradient performance of Softmax in the transformer block, and prove that the input of Softmax is not normal distributed and the gradient vanishing does happens;
- Introducing a series of periodic alternatives for Softmax in attention mechanisms, which compress the input into the unsaturation area by periodic functions to escape the gradient vanishing;
• Explore the impact of pre-normalization for Softmax and our methods and make an observation that pre-normalization is just a conditional solution but not always a good choice;

![Diagram](image)

Figure 3: The original designed attention block are proposed in [2]. Our work is trying to replace the Softmax with a series of periodic alternatives.

## 2 Related works

There are few studies on alternatives of Softmax, since Softmax is mostly used to output classification results, and we can avoid the gradient vanishing by using a joint gradient analytical solution of Softmax and Cross-Entropy. But in the attention mechanism, Softmax is used alone, so the gradient vanishing problem appears. There are other works devoted to enhancing the input feature of Softmax by normalization [3-7]. However, they are all focused on the representation of features, but not addressing the gradient vanishing problem.

**Taylor softmax:** Vincent et al. [8] used second order Taylor series approximation for $e^x$ as $1 + x + 0.5 \cdot x^2$ and derive the Taylor softmax as follows:

$$S_j = \frac{1 + x_j + 0.5 \cdot x_j^2}{\sum_{i=1}^{d} 1 + x_i + 0.5 \cdot x_i^2}, x_j \subseteq (x_1, x_2, \ldots, x_j, \ldots x_d)$$

where $d$ is the dimension of the input of Taylor softmax. Since the quadratic function changes smoothly, Taylor softmax can generate more soft classification results to alleviate overfitting. However, when used without Cross-Entropy, Taylor softmax will cause gradient vanishing too, because of the saturation area.

**Soft-margin softmax:** Liang et al. [9] introduced a distance margin into Softmax to strengthen the inter-class compactness and inter-class separability between learned features. The Soft-margin (SM) softmax can be described as follows:

$$S_j = \frac{e^{x_j - 1}}{e^{x_j - 1} + \sum_{i \neq j}^{d} e_i}, x_j \subseteq (x_1, x_2, \ldots, x_j, \ldots x_d)$$

where $m$ is manually set and when $m$ is set to zero, the SM-Softmax becomes identical to the original Softmax. SM-Softmax can be considered as a shifted version of Softmax. Similar to Taylor softmax, SM-Softmax and its variant, Ensemble Soft-Margin Softmax [10], are proposed to encourage the discriminability of features, and the gradient vanishing problem is still not addressed.

**SM-Taylor softmax:** Kunal et al. [11] explored higher order Taylor series approximation of $e^x$ to come up with an $n^{th}$ order Taylor softmax where:

$$f^n(x) = \sum_{i=0}^{n} \frac{x^i}{i!}$$

They proved that $f^n(x)$ is always positive definite if $n$ is even. Additionally, they combined the strengths of Taylor softmax and Soft-margin softmax, and proposed the SM-Taylor softmax as
follows:

\[ S_j = \frac{f^n(x_j)}{\sum_{i} f^n(x_i)}, x_j \subseteq (x_1, x_2 \ldots, x_j, \ldots x_d) \]

However, it is still a method to enhance features, but not a solution to the gradient problem.

3 Formulation

For convenience, we denote input, inter-value and scores by \( x_j, f(x) \) and \( S_j \). In this section, we try to build some periodic functions as the alternatives of Softmax. In addition, there are five aspects to determine whether a function is a favorable alternative: (1) value stability; (2) gradient stability; (3) saturation area; (4) zero-region gradient; (5) information submerged. Furthermore, when judging the gradient-related properties of the functions, we only consider aspects related to \( f(x) \) instead of the other elements contained by the input. Since the scores are mapped by \( f(x_j)/\sum_{i} f(x_j) \), the correlation between the gradient and the other elements is unavoidable for the periodic functions. The plots of \( \text{extre}(\partial S_j/\partial x_j) - \sum_{i} f(x_j) \) and more discussion on how the other elements in the input influence the gradient of \( x_j \) are provided in A.1.

![Figure 4: Gradient of Softmax and periodic alternatives proposed.](image)

According to the research in the Taylor Softmax function proposed by Vincent et al. [8], and the higher order Taylor Softmax proposed by Kunal et al. [11], it is reasonable to map the input with a monotonic function, since the input can adapt to a suitable value range as parameters update. Therefore, we suggest to use a periodic function \( f(x) \) to compress the input, so as to avoid approximating the small input to a fixed value (to keep them positive), and also avoid the output being too large to have an appropriate gradient.
3.1 Softmax

What Softmax does is mapping the input \( x \) to an inter-value \( f(x) = e^x \), and mapping the inter-value to scores \( S_j \). The exponential function keeps the negative input positive, but also makes the positive input extremely high. For a large input \( x_j \), \( e^{x_j} \) is too large and dominates \( S_j \), which means \( S_j \approx 1, \partial S_j / \partial x_j = 0 \); and for a small input \( x_j \), \( \partial e^{x_j} / \partial x_j = 0 \), and since \( \partial S_j / \partial x_j = (\partial S_j / \partial e^{x_j}) \cdot (\partial e^{x_j} / \partial x_j), \partial S_j / \partial x_j = 0 \). Therefore, the major cause of gradient vanishing in Softmax is that we need to compress the value \( f(x) \) with unknown upper and lower into \((0, +\infty)\). To do this, there has to be a saturation area for large and small value.

Before the discussion of the alternatives of Softmax, it is necessary to clarify the advantages of Softmax. First of all, the output of Softmax is positive definite. And due to exponential function, the difference between inputs will be magnified, that means Softmax can show the difference between inputs well, which is a good characteristic for attention mechanisms. Besides, according to the definition:

\[
S_{\text{softmax}}(x_j) = e^{x_j} / \sum_{i=1}^{d} e^{x_i}, x_j \subseteq (x_1, x_2, \ldots, x_j, \ldots, x_d)
\]

\[
\partial S_j / \partial x_j = M \cdot e^{x_j} / (M + e^{x_j})^2, M = \sum_{i \neq j}^{d} e^{x_i}
\]

\[
\frac{\partial^2 S_j}{\partial x_j^2} = M \cdot e^{x_j} \cdot (e^{x_j} + M)^{-2} - 2 \cdot M \cdot e^{2x_j} \cdot (e^{x_j} + M)^{-3}
\]

Let \( \partial^2 S_j / \partial x_j^2 = 0 \), we have:

\[
M \cdot e^{x_j} \cdot (e^{x_j} + M)^{-2} - 2 \cdot M \cdot e^{2x_j} \cdot (e^{x_j} + M)^{-3} = 0
\]

\[
M \cdot e^{x_j} = M \cdot M / (M + M)^2 = 1/4
\]

which means the max gradient of Softmax is 0.25, so it will not cause gradient explosions. Additionally, in spite of the saturation problem, no matter how large the other elements of input are, a sufficiently large value can always get an appropriate gradient.

3.2 Sin-max-constant / Cos-max:

When it comes to periodic functions, there is a good reason to use sine function since it is widely used and derivable. Therefore, we set the \( f(x) = 1 + \sin(x) \) to keep the function positive definite following the suggestion in [12], and Sin-max-constant is defined as follows:

\[
S_{\text{sin max}}(x_j) = \frac{1 + \sin(x_j)}{d + M + \sin(x_j)}, M = \sum_{i \neq j}^{d} \sin(x_i)
\]

where \( d \) represents to the dimension of input. Sin-max compress \( x \) into \((0, 2)\) and for \( \sin(x) \), there is no saturation area.

However, let \( x \sim N(0, 1) \), then \( E(M) \approx - \sin(x_j) \), we have:

\[
E(S_{\text{sin max}}(x_j)) = \frac{1 + \sin(x_j)}{d} = \frac{1}{d} + \frac{\sin(x_j)}{d}, d \gg \sin(x_j)
\]

\[
E(S_{\text{sin max}}(x_j)) \approx \frac{1}{d}
\]

which means as the dimension of \( x \) increases, the influence of \( x \) will be weakened by the constant term, causing \( x \) to be overwhelmed and cannot be mapped to \( S \) correctly. Besides, consider from the view of gradient:

\[
\frac{\partial S_j}{\partial x_j} = \frac{(M + d - 1) \cdot \cos(x_j)}{(M + d + \sin(x_j))^2}
\]
\[ E \left( \frac{\partial S_j}{\partial x_j} \right) = \frac{(- \sin (x_j) + d - 1) \cdot \cos (x_j)}{d^2} \]

Similar to the \( E(S \sin \max_j) \), as the dimension of \( x \) increases, the gradient of \( \sin \max \) will drop to zero, and gradient vanish will occur in the entire value range. The main reason for these defects is the constant term 1 in \( f(x) \).

We try to remove the constant term in \( f(x) \), and the expression becomes to:

\[ S_{\sin max_j} = \frac{\sin (x_j)}{M + \sin (x_j)}, M = \sum_{i \neq j} \sin (x_i) \]

Now let \( x \sim N(0, 1) \), then \( E(M + \sin (x_i)) = 0 \), we have:

\[ E(S_{\sin max_j}) = \frac{\sin (x_j)}{0} \rightarrow \pm \infty \]

which means the value of \( S_{\sin max_j} \) is unstable, causing the network to have a great risk of breaking down. And for gradient, since \( E(M + \sin (x_j)) = 0 \) and \( E(M) = -\sin(x_j) \), we have:

\[ \frac{\partial S_j}{\partial x_j} = \frac{M \cdot \cos (x_j)}{(M + \sin (x_j))^2} \]

\[ E \left( \frac{\partial S_j}{\partial x_j} \right) = \frac{-0.5 \cdot \sin (2 \cdot x_j)}{0} \rightarrow \pm \infty \]

which is also unstable because \( f(x) \) is not positive definite.

The reason why \( f(x) = 1 + \cos(x) \) is not good is similar to \( \sin \max \)-constant where the difference within \( x \) will be submerged due to the constant term as the dimension growing. Let \( f(x) = \cos(x) \), and \( x \sim N(0, 1) \), we can define Cos-max as:

\[ S_{\cos max_j} = \frac{\cos (x_j)}{M + \cos (x_j)}, M = \sum_{i \neq j} \cos (x_i) \]

\[ \frac{\partial S_j}{\partial x_j} = \frac{-M \cdot \sin (x_j)}{(M + \cos (x_j))^2} \]

Assume that \( x \) strictly belongs to \([-1, 1]\), then Cos-max can be considered as the \( \sin \max \) shifted to a positive definite range. However, a gradient vanishing problem will appear when \( x \) clusters in the zero-region. Besides, from the view of gradient stability, we have:

\[ \frac{\partial^2 S_j}{\partial x_j^2} = -M \cdot \cos (x_j) \cdot (M + \cos (x_j))^{-2} + 2 \cdot M \cdot \sin (x_j) \cdot (M + \cos (x_j))^{-3} \]

Let \( \partial^2 S_j / \partial x_j^2 = 0 \), we have:

\[ \cos (x_j) - 2 \cdot \sin (x_j) \cdot (M + \cos (x_j))^{-1} = 0 \]

\[ -4 + (M^2 + 1) \cdot \cos^2 (x_j) + 2 \cdot M \cdot \cos^3 (x_j) + \cos^4 (x_j) = 0 \]

According to the solving provided in A.5, we can have:

\[ \text{Extre} \left( \frac{\partial S_j}{\partial x_j} \right) \in (-\infty, +\infty) \]

As \( M \) increases, \( \text{Extre}(\partial S_j / \partial x_j) \) approaches \( \pm \infty \), which means Cos-max is gradient unstable, causing the network to have a great risk of breaking down like \( \sin \max \).
3.3 Sin2-max-shifted

To ensure that $f(x)$ is positive definite, and no extra constant term is introduced, $\sin^2 x$ is a reasonable choice. So we can define Sin2-max as:

$$S_{\sin\text{-}2\text{max},j} = \frac{\sin^2 (x_j)}{M + \sin^2 (x_j)}, M = \sum_{i \neq j} \sin^2 (x_i)$$

$$\frac{\partial S_j}{\partial x_j} = \frac{M \cdot \sin (2 \cdot x_j)}{(M + \sin^2 (x_j))^2}$$

Note that although $\sin^2(x) = 1/2 - \cos(2 \cdot x)/2$, $f_{\sin\text{-}2\text{max}}(x) = 1/2 - f_{\cos\text{max}}(2 \cdot x)/2$. Sin2-max is not just a scaled double-frequency version of Cos-max owing to the constant terms. Therefore, the numerical and gradient characteristics of Sin2-max and Cos-max are different.

As shown in Figure 4, the possible problem of Sin2-max is that, assuming $x \sim N(0, 1)$, most of $x$ clusters in the region close to 0 so most of the gradients close to 0, which makes the parameters difficult to update. To solve this ‘conditional’ problem, we can shift $\sin^2(x)$ to the non-zero region by adding a phase $\varphi$ to $x$.

Let $\partial^2 S_j / \partial x_j^2 = 0$. We have:

$$\cos (2 \cdot x_j) = -\frac{1}{2} \left(2 \cdot M + 1 \pm \sqrt{8 + (2 \cdot M + 1)^2}\right)$$

$x_j$ for max $\left(\frac{\partial S_j}{\partial x_j}\right) = \frac{1}{2} \arccos \left(-\frac{1}{2} \left(2 \cdot M + 1 \pm \sqrt{8 + (2 \cdot M + 1)^2}\right)\right)$

Unfortunately, $x_j$ for max $\left(\frac{\partial S_j}{\partial x_j}\right)$ will change with $M$, so we have to find an approximate solution. Besides, with $M$ change, the gradient will oscillate in the range of $(0, +\infty)$, causing gradient explosion or vanishing in the entire value range. Since $\sin (2 \cdot x_j)$ and $\sin^2 (x_j)$ has a same cycle of period $\pi$, the period of $\partial S_j / \partial x_j$ is $\pi$. We set $\varphi = \pi/4$ to make the gradient stable. So we get Sin2-max-shifted as follows:

$$S_{\sin\text{-}2\text{max-shifted},j} = \frac{\sin^2 (x_j + 0.25 \cdot \pi)}{M + \sin^2 (x_j + 0.25 \cdot \pi)}, M = \sum_{i \neq j} \sin^2 (x_i + 0.25 \cdot \pi)$$

$$\frac{\partial S_j}{\partial x_j} = \frac{M \cdot \sin (2 \cdot x_j + 0.5 \cdot \pi)}{(M + \sin^2 (x_j + 0.25 \cdot \pi))^2}$$

3.4 Sin-Softmax

From another view, instead of replacing the exponential function, compressing the input into the unsaturation area is also a reasonable choice. To keep the gradient of input in zero region away from 0, we choose sine but not cosine. We define Sin-Softmax as follows:

$$S_{\sin\text{-}\text{Softmax},j} = \frac{e^{\sin (x_j)}}{M + e^{\sin (x_j)}}, M = \sum_{i \neq j} e^{\sin (x_i)}$$

$$\frac{\partial S_j}{\partial x_j} = \frac{M \cdot e^{\sin (x_j)} \cdot \cos (x_j)}{(M + e^{\sin (x_j)})^2}$$

Sin-Softmax can be considered as a periodic-normalized version of Softmax, which is also similar to the periodic activation function proposed in SirenNet [13]. The best part of Sin-Softmax is that, the input is compressed into the well performed region of Softmax by periodic function, so the value and gradient are both stable, as show in Figure 4 and A.1. Additionally, $S_j$ is positive definite owing to the exponential function, and the gradient in the zero-near region is also well performed. However, the possible defect of Sin-Softmax is that, the largest score can only be $e^2$ times the smallest, since for $x_j \in R$:

$$\sin (x_j) \in (-1, +1), e^{\sin (x_j)} \in \left(\frac{1}{e}, e\right)$$

which might cause oblivion of the most contributing value covered in a large number of low contributing values, as the dimension of the score maps (or the number of embeddings) increasing.
3.5 Siren-max

Inspired by SirenNet [13] where sine is used as the activation function, we define a \( f(x) \) with benefic gradient properties by:

\[
f(x) = \frac{\sin(x)}{1 - \sin(x)}
\]

Since \( f(x) \in [-0.5, +\infty) \), to make \( S_j \) positive definite, we add 0.5 to \( f(x) \) and define Siren-max as follows:

\[
S_{\text{siren-max}} = 1 + \frac{\sin(x_j)}{2 - 2 \times \sin(x_j)} \left( \frac{1 + \sin(x_j)}{2 - 2 \times \sin(x_j)} \right), \quad M = \sum_{i \neq j} \frac{1 + \sin(x_i)}{2 - 2 \times \sin(x_i)}
\]

\[
\frac{\partial S_j}{\partial x_j} = \frac{4 \times M \times \cos(x_j)}{(2 \times M - 2 \times M \times \sin(x_j) + 1 + \sin(x_j))^2}
\]

Note that, the upper bound of \( f(x) \) is infinity, so adding a constant term to it will not cause the difference between \( x \) being submerged like Sin-max-constant. As shown in Figure 4, there is no saturation area in Siren-max and it is well performed in zero-near region. The possible defect is that, the gradient has periodic jump points which might make training unstable.

### Table 1: A summary of Softmax and the periodic alternatives.

| Value stability? | Gradient stability? | No saturation area? | Zero-region gradient good? | Info submerged? |
|-------------------|----------------------|----------------------|-----------------------------|-----------------|
| Softmax           | ✓                    | ✓                    | ✓                           | ✓               |
| Sin-max-constant  | x/✓                  | ✓                    | ✓                           | ✓               |
| Cos-max           | ✓/✓                  | ✓/✓                  | ✓                           | ✓               |
| Sin2-max          | ✓                    | ✓                    | ✓                           | ✓               |
| Sin2-max-shifted  | ✓                    | ✓                    | ✓                           | ✓               |
| Sin-Softmax       | ✓                    | ✓                    | ✓                           | ✓               |
| Siren-max         | ✓/✓                  | ✓/✓                  | ✓                           | ✓               |

3.6 Pre-normalization

Note that the pre-normalization discussed is not parameterized like Batch-norm [14], Layer-norm [15] or Group-norm [16]. We denote the pre-normalization by normalizing the elements of \( Q \cdot K \) in row. Considering the saturation problem, normalizing the input is also a reasonable operation. However, since the distribution of attention score maps is different for variant images, the normalization can hardly compress the maps into a specified value range precisely. Besides, the function \( f(x_j) = \text{normalized}(x_j) \) also might be saturated, and the saturation area shifts with \( E(x_j) \) and \( \text{Var}(x_j) \).

Therefore, the gradient situation of \( \text{normalized}(x_j) \) is similar to that of Softmax, which may cause gradient vanishing too. As a result, although pre-normalization can roughly gather the values in a specified range, but on the contrary, it may bring new gradient problems. More discussion and the gradient plots of normalization, pre-normalized version of Softmax and periodic alternatives are provide in A.2.

### 4 Experiments

To eliminate the unexpected effects of various tricks, the experiments are operated on a simply designed demo referenced to LeViT [1], as shown in Figure 5.
Figure 5: The structure of the demo trained in experiments. Except for the number of attention blocks and the number of neurons of the classification head, the other parts remain unchanged in the experiment.

In experiments, there is an observation that most of the gradient of Softmax are very small, and only a small part of updates can be successfully back-propagated, even in the early stages of training, as shown in Figure 6. This phenomenon proves our point that the $Q \cdot K$ value, which are used to generate the attention scores, are related to the input images content and are not strictly normal distributed. Therefore, even if divided by $\sqrt{d}$ following the original designed transformer block, the value still might fall into the saturation area of Softmax, making updates difficult. In our method, there is no saturation area in the functions, so the gradient is satisfactory at each training stage, which promotes the updating of parameters. More 3D graphs of gradients extracted from experiment are provide in A.3.

Figure 6: 3D graph of the gradient extracted from attention blocks under Softmax and Sin-Softmax. The input of Softmax represents the correlation of embedding (patches), and the distribution of correlation should be variant for different images. As a result, the distribution of $K, Q$ will gradually deviate from the normal distribution, as will the input of Softmax. Since a part of the input might fall into the saturation area, gradient vanishing occurs. And our methods successfully address this problem by introducing the period functions to generate attention scores. More 3D gradient graphs of our methods are provided in A.3.
As shown in Figure 1 and Figure 7, due to the gradient vanishing problem, Softmax might cause difficulty in the formation of attention, especially in the early stages of training. We observe that attention is formed more smoothly on the boundary, and on the contrary, the attention corresponding to the objects can only be formed in the later stages of training. A possible reason is that the scores of the boundary between object and content are moderate, so the gradient flows smoothly. While, the scores of the objects are larger and might fall into the saturation area of Softmax, causing the gradient vanishing and the formation of attention being locked. While under the periodic alternatives, the attention is updated unrestrained on the image, which strengthens our arguments.

Figure 7: Visualization of the attention extracted from transformer blocks. In softmax, attention is more likely to be trapped on the boundary, since the $QK$ value of the boundary is moderate and not easy to fall into the saturation area. By contrast, our method eliminates the saturation area, and attention can be updated smoothly on the objects too.

Table 2: Comparison with Softmax under same model with increasing depth. We also report pre-normalization version result of each functions. Note that, since the trainings under Sin-max-constant, Cos-max, Sin2-max breaks down in the early stages, we omit their results here. In addition, the depth represents to the number of attention blocks stacked, and the rest structures of demo remain unchanged.

| Structure = Demo | Depth = 1 | Depth = 2 | Depth = 4 | Depth = 8 |
|------------------|-----------|-----------|-----------|-----------|
| Dataset = Cifar-100 | norm | norm | norm | norm |
| Softmax | 52.67 | 53.80 | 58.21 | 59.57 | 67.01 | 68.75 | 81.12 | 82.31 |
| Sin2-max-shifted | 53.90 | 53.16 | 60.37 | 59.84 | 71.48 | 72.38 | 84.81 | 83.20 |
| Sin-Softmax | **54.64** | **54.27** | **60.40** | **60.18** | 72.39 | 72.84 | **85.14** | 84.63 |
| Siren-max | \ | \ | \ | **73.25** | \ | \ | 84.70 |

*\ means training breaks down in the early stages

The gradient performance in the zero-region is crucial for training, and the early breaking down of training under Cos-max and Sin2-max can be ascribed to this. Besides, the stability of gradient is also very important. Since there are jump points within the range of input, the training under Siren-max breaks down too. In addition, since the input is submerged in the constant terms, the training under Sin-max-constant diverges.

Encouragingly, Sin2-max-shifted, Sin-Softmax exceed Softmax in the result just as we speculate, and norm-Siren-max is also surprisingly well performed. The result are shown in Table 2. The
major drawbacks of Cos-max and Sin-max-constant are gradient performance in zero-region and information submergence respectively, which cannot be optimized by pre-normalization. As for Siren-max, pre-normalization optimizes the distribution of input and helps Siren-max avoid the gradient jump points, resulting in a satisfactory performance. Softmax can also be improved since pre-normalization helps the input escape from the saturation area to some degree. However, Sin2-max-shifted and Sin-Softmax are not subject to input distribution, so they cannot get benefits from pre-normalization. On the contrary, since pre-normalization will bring unexpected gradient problems, the performance of norm-Sin2-max-shifted and norm-Sin-Softmax decrease slightly. The plots and complete results of the experiment are provided in A.4.

5 Conclusion

Through the visualization of attention and gradient extracted from transformer blocks, we prove that in the attention mechanism, Softmax does lead to the gradient vanishing problem and makes training difficult. To address the problem, we propose a series of periodic alternatives of Softmax, and the experimental results prove that Sin-Softmax, Sin2-max-shifted, and norm-Siren-max are better performed than Softmax in attention mechanism. Additionally, we make an observation that pre-normalization is just a conditional solution but not always a good choice.

In the periodic alternatives, embedding requiring more attention does not necessarily require a larger $Q \cdot K$ value, which makes the generation of $Q, K$ more free, and it is hard to say whether this will lead to unexpected problems. This change might affect the representation of the model, and we will explore how this change happens in further works.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In Proc. ICLR, 2021.
[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. arXiv preprint arXiv: 1706.03762v5, 2017.
[3] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In ICML, volume 2, page 7, 2016.
[4] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: $L_2$ hypersphere embedding for face verification. arXiv preprint arXiv: 1704.06369v4, 2017.
[5] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. $L_2$-constrained softmax loss for discriminative face verification. arXiv preprint arXiv:1703.09507v3, 2017.
[6] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. arXiv preprint arXiv:1704.08063v4, 2018.
[7] Yutong Zheng, Dipan K Pal, and Marios Savvides. Ring loss: Convex feature normalization for face recognition. 1803.00130v1, 2018.
[8] Pascal Vincent, Alexandre De Brébisson, and Xavier Bouthillier. Efficient exact gradient update for training deep networks with very large sparse targets. arXiv preprint arXiv:1412.7091v1, 2014.
[9] Xuezhi Liang, Xiaobo Wang, Zhen Lei, Shengcui Liao, and Stan Z Li. Soft-margin softmax for deep classification. In Proc. ICONIP, pages 413–421. Springer, 2017.
[10] Xiaobo Wang, Shifeng Zhang, Zhen Lei, Si Liu, Xiaojie Guo, and Stan Z Li. Ensemble soft-margin softmax loss for image classification. arXiv preprint arXiv:1805.03922v1, 2018.
[11] Kunal Banerjee, Rishi Raj Gupta, Karthik Vyas, Biswajit Mishra, et al. Exploring alternatives to softmax function. arXiv preprint arXiv:2011.11538v1, 2020.
[12] Alexandre de Brébisson and Pascal Vincent. An exploration of softmax alternatives belonging to the spherical loss family. arXiv preprint arXiv:1511.05042v3, 2016.
[13] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. arXiv preprint arXiv: 2006.09661v1, 2020.
[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv: 1502.03167v3, 2015.
[15] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450v1, 2016.
[16] Yuxin Wu and Kaiming He. Group normalization. arXiv preprint arXiv: 1803.08494v3, 2018.
Escaping the Gradient Vanishing: Periodic Alternatives of Softmax in Attention Mechanism

Appendix

In this supplemental material, we provide more plots and solutions to support our argument. In addition, we also provide more 3D graphs and experiment results omitted from the paper.

A.1 How the other elements in the input influence the gradient.

![Graphs showing the influence of other elements in the input on the gradient.](image)

Figure 8: The plot of the sum of $f(x)$ without $f(x_j)$ and the extreme gradient of $x_j$, where $d$ represents to the dimension of input. Since norm-Sin-max_constant contains a constant term, there is an extra variable $d$ changing from $[2, 256]$ with step size $= 2$.

Softmax and the periodic alternatives can be described as follows:

$$S_j = \frac{f(x_j)}{f(x_j) + M} \quad M = \sum_{i \neq j} f(x_i)$$

where $d$ represents to the dimension of input $x$.

$$\frac{\partial S_j}{\partial x_j} = g(M) \cdot \frac{\partial f(x_j)}{\partial x_j}, g(M) = \frac{M}{(M + f(x_j))^2}$$

Since $f(x_j)$ is expected to be positive definite, we have $M \in (0, +\infty)$. The $g(M)$ can be considered as a band-pass filter with a maximum pass rate of $-\infty$, and the center frequency is $f(x_j)$. As the $M$ moves away from the center frequency, the pass rate gradually decays to 0. Therefore, the closer $M$ is to $f(x_j)$, the more smoothly the gradient of $x_j$ can pass through the filter. In another words, the greater the difference between $f(x_j)$, the less smoothly the gradient flows. Accordingly, from the perspective of gradient, $f(x)$ should not be a function with sharp attenuation. Less attenuation is one of the reasons why periodic alternatives are superior to Softmax.
A.2 pre-Normalization and gradient

Normalization can be described as follows:

$$\text{norm} (x_j) = \frac{x_j - \text{mean} (x_j)}{\sqrt{\text{var} (x_j)}}$$

$$\frac{\partial \text{norm} (x_j)}{\partial x_j} = \frac{d - 1}{d \cdot \sqrt{\text{var} (x_j)}} \cdot \frac{x_j - \text{mean} (x_j)}{2 \cdot \sqrt{\text{var} (x_j)}} \cdot \frac{\partial \text{var} (x_j)}{\partial x_j}$$

Denote the dimension of $x$ by $d$, then we have:

$$\frac{\partial \text{var} (x_j)}{\partial x_j} = \frac{d - 1}{d^2} \cdot (x_j - \text{mean} (x_j)) - \frac{1}{d^2} \sum_{i \neq j}^d (x_i - \text{mean} (x_j))$$

$$\text{mean} (x_j) = x_j + \sum_{i \neq j}^d x_i$$

Then we can plot the gradient of normalization as Figure 9. The gradient of normalization is related to the distribution of input, which might lead to extra gradient problems, such as gradient vanishing or explosion. Therefore, although pre-normalization can roughly gather the input into the best performed area of functions as shown in Figure 10, it is not always a good choice, but just a conditional solution depending on the distribution of input.

![Figure 9: The gradient plot of normalization. Note that there are three extra variables: dimension of input $d$, statistical variance of input $\text{var}$ and the sum of input without $x_j$, and we only annotate two sets of variables here.](image)
Figure 10: The gradient plot of pre-normalized version of Softmax and periodic alternatives, where \( M = \sum_{i \neq j} f(x_i) \), and \( d \) represents to the dimension of input. For example, for norm-Sin-max_constant, \( M = \sum_{i \neq j} 1 + \sin(x_i) \).

Figure 11: The pre-normalized version of the plots on the sum of input without \( x_j \) and the extreme gradient of \( x_j \). Since norm-Sin-max_constant contains a constant term, there is an extra variable \( d \) changing from \([2, 256]\) with step_size = 2, representing to the dimension of input.
A.3 3D graphs of gradients extracted from experiment

As shown in Figure 12, just as we speculate, the gradient of Cos-max falls to zero due to the poor gradient performance in zero-region, and the gradient Sin2-max-shifted and Sin-Softmax is well performed at each stage of training as shown in Figure 13 and Figure 14.

Figure 12: 3D gradient graph of Cos-max in experiment

Figure 13: 3D gradient graph of Sin-Softmax in experiment
Figure 14: 3D gradient graph of Sin2-max-shifted in experiment

Figure 15: 3D gradient graph of Softmax in experiment
A.4 Experiment results

Figure 16: Experiment results of Softmax, periodic alternatives and their pre-normalized version. The depth represents to the number of attention blocks stacked, and the rest structures of demo remain unchanged.

Table 3: Softmax and periodic alternatives are tested under increasing depth. The depth represents to the number of attention blocks stacked, and the rest structures of demo remain unchanged.

| Structure = Demo | Dataset = Cifar-100 | Depth = 1 | Depth = 2 | Depth = 4 | Depth = 8 |
|------------------|---------------------|-----------|-----------|-----------|-----------|
|                  | norm                | norm      | norm      | norm      | norm      |
| Softmax          | 52.67               | 53.80     | 58.21     | 59.57     | 67.01     | 68.75     | 81.12     | 82.31     |
| Sin-max-constant | \                  | \         | \         | \         | \         | \         | \         | \         |
| Cos-max          | 0.496               | 0.480     | 0.493     | \         | \         | \         | \         | \         |
| Sin2-max         | \                  | \         | \         | \         | \         | \         | \         | \         |
| Sin2-max-shifted | 53.90               | 53.16     | 60.37     | 59.84     | 71.48     | 72.38     | 84.81     | 83.20     |
| Sin-Softmax      | \                  | \         | \         | 60.40     | 60.18     | 72.39     | 71.84     | 85.14     | 84.63     |
| Siren-max        | \                  | \         | \         | \         | \         | \         | \         | 73.25     | 84.70     |

*\ means training breaks down in the early stages

As shown in Table 3 and Figure 16, due to the poor zero-region gradient performance, training under Sin2-max breaks down in the early stages, as same as Cos-max under a greater depth. The input of Sin-max-constant is submerged by the constant term, and the breaking down also occurs. In addition, Siren-max need normalization to make gradient avoid jump point, otherwise it perform poorly.
A.5 Solutions

Solution for the range of extremum gradient of Cos-max:

\[
\frac{\partial S_j}{\partial x_j} = \frac{-M \cdot \sin(x_j)}{(M + \cos(x_j))^2}
\]

\[
\frac{\partial^2 S_j}{\partial x_j^2} = -M \cdot \cos(x_j) \cdot (M + \cos(x_j))^{-2} + 2 \cdot M \cdot \sin(x_j) \cdot (M + \cos(x_j))^{-3}
\]

Let \(\frac{\partial^2 S_j}{\partial x_j^2} = 0\), we have:

\[
\cos(x_j) - 2 \cdot \sin(x_j) \cdot (M + \cos(x_j))^{-1} = 0
\]

\[
\sin(x_j) \cdot (M + \cos(x_j))^{-2} = 0.5 \cdot \cos(x_j) \cdot (M + \cos(x_j))^{-1}
\]

\[
\therefore \text{Extremum } \left(\frac{\partial S_j}{\partial x_j}\right) = \frac{-M \cdot \cos(x_j)}{M + \cos(x_j)} = \frac{M^2}{2M + 2 \cos(x_j)} - \frac{M}{2}
\]

\[
\therefore \text{Extremum } \left(\frac{\partial S_j}{\partial x_j}\right) \in \left[\frac{M^2}{2M + 2} - \frac{M}{2}, \frac{M^2}{2M - 2} - \frac{M}{2}\right]
\]

Since Cos-max is not positive definite, \(M \in [-\infty, +\infty]\). So we have:

\[
\text{Extremum } \left(\frac{\partial S_i}{\partial x_j}\right) \in (-\infty, +\infty)
\]