Exploring Alternatives to Softmax Function

Kunal Banerjee[1,*][a], Vishak Prasad C.[2] and Rishi Raj Gupta[2,†], Kartik Vyas[2,†], Anushree H.[2,†] and Biswajit Mishra[2]

[1]*Walmart Global Tech, Bangalore, India*

[2]*Intel Corporation, Bangalore, India*

Keywords: Softmax, Spherical Loss, Function Approximation, Classification.

Abstract: Softmax function is widely used in artificial neural networks for multiclass classification, multilabel classification, attention mechanisms, etc. However, its efficacy is often questioned in literature. The log-softmax loss has been shown to belong to a more generic class of loss functions, called *spherical family*, and its member log-Taylor softmax loss is arguably the best alternative in this class. In another approach which tries to enhance the discriminative nature of the softmax function, *soft-margin softmax (SM-softmax)* has been proposed to be the most suitable alternative. In this work, we investigate Taylor softmax, SM-softmax and our proposed SM-Taylor softmax, an amalgamation of the earlier two functions, as alternatives to softmax function. Furthermore, we explore the effect of expanding Taylor softmax up to ten terms (original work proposed expanding only to two terms) along with the ramifications of considering Taylor softmax to be a finite or infinite series during backpropagation. Our experiments for the image classification task on different datasets reveal that there is always a configuration of the SM-Taylor softmax function that outperforms the normal softmax function and its other alternatives.

1 INTRODUCTION

Softmax function is a popular choice in deep learning classification tasks, where it typically appears as the last layer. Recently, this function has found application in other operations as well, such as the attention mechanisms (Vaswani et al., 2017). However, the softmax function has often been scrutinized in search of finding a better alternative (Vincent et al., 2015; de Brébisson and Vincent, 2016; Liu et al., 2016; Liang et al., 2017; Lee et al., 2018).

Specifically, Vincent et al. explore the spherical loss family in (Vincent et al., 2015) that has log-softmax loss as one of its members. Brebisson and Vincent further work on this family of loss functions and propose log-Taylor softmax as a superior alternative than others, including original log-softmax loss, in (de Brébisson and Vincent, 2016).

Liu et al. take a different approach to enhance the softmax function by exploring alternatives which may improve the discriminative property of the final layer as reported in (Liu et al., 2016). The authors propose *large-margin softmax (LM-softmax)* that tries to increase inter-class separation and decrease intra-class separation. LM-softmax is shown to outperform softmax in image classification task across various datasets. This approach is further investigated by Liang et al. in (Liang et al., 2017), where they propose *soft-margin softmax (SM-softmax)* that provides a finer control over the inter-class separation compared to LM-softmax. Consequently, SM-softmax is shown to be a better alternative than its predecessor LM-softmax (Liang et al., 2017).

In this work, we explore the various alternatives proposed for softmax function in the existing literature. Specifically, we focus on two contrasting approaches based on spherical loss and discriminative property and choose the best alternative that each has to offer: log-Taylor softmax loss and SM-softmax, respectively. Moreover, we enhance these functions to investigate whether further improvements can be achieved. The contributions of this paper are as follows:

- We propose SM-Taylor softmax – an amalgamation of Taylor softmax and SM-softmax.
- We explore the effect of expanding Taylor softmax up to ten terms (original work (de Brébisson

[a] https://orcid.org/0000-0002-0605-630X

*Work done when the author worked at Intel Corporation

†Work done during internship at Intel Corporation

Banerjee, K., C., V., Gupta, R., Vyas, K., H., A. and Mishra, B.
Exploring Alternatives to Softmax Function.
DOI: 10.5220/0010502000810086
In *Proceedings of the 2nd International Conference on Deep Learning Theory and Applications (DeLTA 2021)*, pages 81-86
ISBN: 978-989-758-526-5

and Vincent, 2016) proposed expanding only to two terms) and we prove higher order even terms in Taylor's series are positive definite, as needed in Taylor softmax.

- We explore ramifications of considering Taylor softmax to be a finite or infinite series during backpropagation.
- We compare the above mentioned variants with Taylor softmax, SM-softmax and softmax for image classification task.

It may be pertinent to note that we do not explore other alternatives such as, dropmax (Lee et al., 2018), because it requires the true labels to be available; however, such labels may not exist in other tasks where softmax function is used, for example, attention mechanism (Vaswani et al., 2017). Consequently, dropmax cannot be considered as a drop-in replacement for softmax universally and hence we discard it.

The paper is organized as follows. Section 2 elaborates on the softmax function and its several alternatives explored here. Experimental results are provided in Section 3. Section 4 concludes the paper and shares our plan for future work.

2 ALTERNATIVES TO SOFTMAX

In this section, we provide a brief overview of the softmax function and its alternatives explored in this work.

2.1 Softmax

The softmax function $sm : \mathbb{R}^K \to \mathbb{R}^K$ is defined by the formula:

$$sm(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1,\dots,K \text{ and } \mathbf{z} = (z_1,\dots,z_K) \in \mathbb{R}^K \tag{1}$$

To clarify, the exponential function is applied to each element z_i of the input vector $\mathbf{z}$ and the resulting values are normalized by dividing by the sum of all the exponentials. The normalization guarantees that the elements of the output vector $sm(\mathbf{z})$ sum up to 1.

2.2 Taylor Softmax

The Taylor softmax function as proposed by Vincent et al. (Vincent et al., 2015) uses second order Taylor series approximation for e^z as $1+z+0.5z^2$. They then derive the Taylor softmax as follows:

$$Tsm(\mathbf{z})_i = \frac{1+z_i+0.5z_i^2}{\sum_{j=1}^{K} 1+z_j+0.5z_j^2} \quad \text{for } i = 1,\dots,K \text{ and } \mathbf{z} = (z_1,\dots,z_K) \in \mathbb{R}^K \tag{2}$$

Moreover, the second order approximation of e^z as $1+z+0.5z^2$ is positive definite, and hence it is suitable to represent a probability distribution of classes (de Brébisson and Vincent, 2016). Again, it has a minimum value of 0.5, so the numerator of the equation 2 never becomes zero, that enhances numerical stability.

We explore higher order Taylor series approximation of e^z (as $f^n(z)$) to come up with an n^{th} order Taylor softmax.

$$f^n(z) = \sum_{i=0}^{n} \frac{z^i}{i!} \tag{3}$$

Thus, the Taylor softmax for order n is

$$Tsm^n(\mathbf{z})_i = \frac{f^n(z_i)}{\sum_{j=1}^{K} f^n(z_j)} \quad \text{for } i = 1,\dots,K \text{ and } \mathbf{z} = (z_1,\dots,z_K) \in \mathbb{R}^K \tag{4}$$

It is important to note that $f^n(z)$ is always positive definite if n is even. We will prove it by the method of induction.

Base Case: We have already shown that $f^n(z)$ is positive definite for $n = 2$ in Section 2.

Induction Hypothesis: $f^n(z)$ is positive definite for $n = 2k$.

Induction Step: We will prove that it holds for $n = 2(k+1) = 2k+2$, where k is an integer starting from 1.

We denote $f^{2k+2}(z) = S(k+1)$, so

$$S(k+1) = \sum_{i=0}^{2k+2} \frac{z^i}{i!}$$

$$S(k+1) = \sum_{i=0}^{2k} \frac{z^i}{i!} + \frac{z^{2k+1}}{(2k+1)!} + \frac{z^{2k+2}}{(2k+2)!}$$

Let us consider this series with $p \in \mathbb{R}$ and $p > 1$

$$S(k+1,p) = \sum_{i=0}^{2k} \frac{z^i}{i!} + \frac{z^{2k+1}}{(2k+1)!} + \frac{z^{2k+2}}{(2k+2)!p}$$

Clearly, $S(k+1) > S(k+1,p)$ and

$$S(k+1,p) = \sum_{i=0}^{2k-1} \frac{z^i}{i!} + \frac{z^{2k}}{(2k)!}\left(\frac{(4-p)k+2-p}{2(2k+1)} + \frac{(z+k+1)^2}{(2k+1)(2k+2)}\right) > \sum_{i=0}^{2k-1} \frac{z^i}{i!} + \frac{z^{2k}}{(2k)!}\left(\frac{(4-p)k+2-p}{2(2k+1)}\right)$$

If we select $p < \frac{4k+2}{k+1}$ then

$$\frac{(4-p)k+2-p}{2(2k+1)} > 0$$

If we set

$$q = \frac{2(2k+1)}{(4-p)k+2-p}$$

then the expression becomes

$$S(k+1,p) > \sum_{i=0}^{2k-1} \frac{z^i}{i!} + \frac{z^{2k}}{(2k)!q} = S(k,q)$$

We go further to prove $S(k,q) < S(k-1,r)$, it requires

$$q < \frac{4k-2}{k} \Leftrightarrow \frac{2(2k+1)}{(4-p)k+2-p} < \frac{4k-2}{k}$$

$$\text{which is true if } p < \frac{4k+2}{k+1}$$

$$\text{Hence, } S(k+1) > S(k,p) > S(k-1,q) ... > S(1,t)$$

$$\text{and } S(1,t) > 0 \text{ for } t = \frac{5}{3}$$

$$\text{so, } S(k+1) > 0$$

□

The actual back propagation equation for Taylor softmax cross entropy loss function (L) is

$$\frac{\partial L}{\partial z_i} = \frac{f^{n-1}(z_i)}{\sum_{j=1}^{k} f^n(z_j)} - y_i \frac{f^{n-1}(z_i)}{f^n(z_i)} \quad (5)$$

Instead of using equation 5, we used softmax like equation 6 for backpropagation. For very large n (i.e., as n tends to infinity), equations 5 and 6 are equivalent, we denote this variation as Taylor_inf. This equation 5 is corresponding to negative log likelihood loss function of the Taylor softmax probabilities with a regularizer R(z) defined by equation 7; it is because of the regularization effect this method performs better.

$$\frac{\partial L}{\partial z_i} = Tsm^n(\mathbf{z})_i - y_i \quad (6)$$

$$R(\mathbf{z}) = \log \frac{Tsm(\mathbf{z})}{sm(\mathbf{z})} \quad (7)$$

2.3 Soft-margin Softmax

Soft-margin (SM) softmax (Liang et al., 2017) reduces intra-class distances but enhances inter-class discrimination, by introducing a distance margin into the logits. The probability distribution for this, as described in (Liang et al., 2017), is as follows:

$$SMsm(\mathbf{z})_i = \frac{e^{z_i - m}}{\sum_{j \neq i}^{K} e^{z_j} + e^{z_i - m}} \quad (8)$$

$$\text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$$

2.4 SM-Taylor Softmax

SM-Taylor softmax uses the same formula as given in equation 8 while using equation 3, for some given order n, to approximate e^z.

3 EXPERIMENTAL RESULTS

In this section, we share our results for image classification task on MNIST, CIFAR10 and CIFAR100 datasets, where we experiment on the softmax function and its various alternatives. Note that our goal was not to reach the state of the art accuracy for each dataset but to compare the influence of each alternative. Therefore, we restricted ourselves to reasonably sized standard neural network architectures with no ensembling and no data augmentation. Our code is available at https://github.com/kunalbanerjee/softmax_alternatives.

The topology that we have used for each dataset is given in Table 1. The topology for MNIST is taken from (Liang et al., 2017); we experimented with the topologies for CIFAR10 and CIFAR100 given in (Liang et al., 2017) as well to make comparison with the earlier work easier – however, we could not reproduce the accuracies mentioned in (Liang et al., 2017) with the prescribed neural networks. Consequently, we adopted the topology for CIFAR10 mentioned in (Brownlee) and for CIFAR100, we borrow the topology given in (Clevert et al., 2016); in both cases, no data augmentation was applied. The abbreviations used in Table 1 are explained below: (i) Conv[MxN,K] – convolution layer with kernel size MxN and K output channels, we always use stride of 1 and padding as "same" for convolutions; (ii) MaxPool[MxN,S] – maxpool layer with kernel size MxN and stride of S; (iii) FC[K] – fullyconnected layer with K output channels, an appropriate *flatten* operation is invoked before the fullyconnected layer that we have omitted for brevity; (iv) BN – batchnorm layer with default initialization values; (v) Dropout[R] – dropout layer with dropout rate R; (vi) DO – dropout layer with rate 0.5, note that CIFAR100 topology uses a uniform rate for all its dropouts; (vii) {layer1[,layer2]}xN – this combination of layer(s) is repeated N times. In all these topologies, we replace the final softmax function by each of its alternatives in our experiments.

Table 2 shows the effect of varying soft margin m on accuracy for the three datasets. We vary m from 0 to 0.9 with a step size of 0.1, as prescribed in the original work (Liang et al., 2017). We note that m set to 0.6 provided the best accuracy for all the datasets consid-

Table 1: Topologies for different datasets.

MNIST	CIFAR10	CIFAR100
{Conv[3x3,64]}x4 MaxPool[2x2,2] {Conv[3x3,64]}x3 MaxPool[2x2,2] {Conv[3x3,64]}x3 MaxPool[2x2,2] FC[256] FC[10]	{Conv[3x3,32],BN}x2 MaxPool[2x2,1] Dropout[0.2] {Conv[3x3,64],BN}x2 MaxPool[2x2,1] Dropout[0.3] {Conv[3x3,128],BN}x2 MaxPool[2x2,1] Dropout[0.4] FC[128],BN Dropout[0.5] FC[10]	Conv[3x3,384] MaxPool[2x2,1],DO Conv[1x1,384] Conv[2x2,384] {Conv[2x2,640]}x2 MaxPool[2x2,1],DO Conv[3x3,640] {Conv[2x2,768]}x3 Conv[1x1,768] {Conv[2x2,896]}x2 MaxPool[2x2,1],DO Conv[3x3,896] {Conv[2x2,1024]}x2 MaxPool[2x2,1],DO Conv[1x1,1024] Conv[2x2,1152] MaxPool[2x2,1],DO Conv[1x1,1152] MaxPool[2x2,1],DO FC[100]

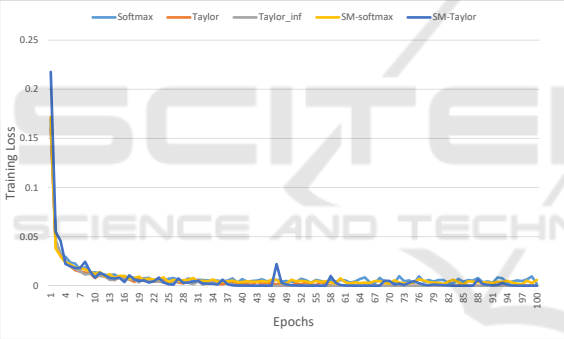

Figure 1: Plot of training loss vs epochs for MNIST dataset.

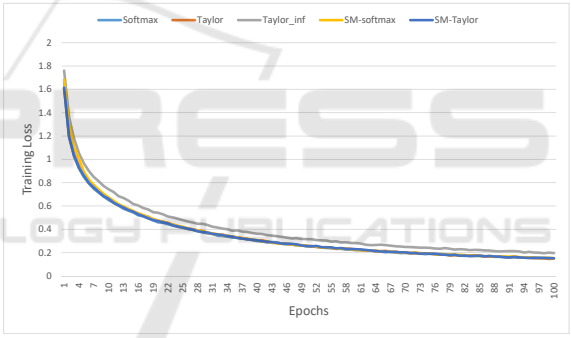

Figure 2: Plot of training loss vs epochs for CIFAR10 dataset.

ered, although there are other values which provide the same best accuracy for MNIST and CIFAR10. Hence, for simplicity, we fix m to 0.6 for all further experiments.

Table 3 compares the softmax function and its various alternatives with respect to image classification task for the different datasets. For Taylor softmax (Taylor) and its variant where we consider equation 6 from Section 2 while doing gradient calculation (Taylor_inf), we look into Taylor series expansion of orders 2 to 10 with a step size of 2. For SM-Taylor softmax, we follow the same expansion orders while keeping soft margin fixed at 0.6. For these three variants, we choose the order which gives the best accuracy and mention it again in the column labeled "Accuracy". As can be seen from Table 3, there is always a configuration for SM-Taylor softmax (namely, $m = 0.6$, $order = 2$ for MNIST and CIFAR10, and $m = 0.6$, $order = 4$ for CIFAR100) that outperforms other alternatives.

The plots for training loss vs epochs for MNIST, CIFAR10 and CIFAR100 are given in Figure 1, Figure 2 and Figure 3, respectively, It may be pertinent to note that in Figure 1, we see fluctuation in the training loss for the softmax function, whereas the plot is comparatively smoother for all its alternatives.

4 CONCLUSION AND FUTURE WORK

Softmax function can be found in almost all modern artificial neural network models whose applications range from image classification, object detection, language translation to many more, However,

Table 2: SM-softmax accuracies for different datasets.

Dataset	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
MNIST	99.46	99.42	99.45	99.48	99.52	99.46	**99.54**	99.46	**99.54**	99.47
CIFAR10	87.09	87.10	87.29	87.15	**87.33**	87.30	**87.33**	87.22	87.12	87.25
CIFAR100	48.28	48.03	48.06	48.11	47.82	47.68	**48.95**	48.03	47.96	48.02

Table 3: Comparison among softmax and its alternatives.

Dataset	Variants	Accuracy	2	4	6	8	10
MNIST	softmax	99.41					
	Taylor	99.65	99.54	99.59	99.50	99.65	99.51
	Taylor_inf	99.62	99.54	99.60	99.59	99.62	99.47
	SM-softmax	99.54					
	SM-Taylor	**99.67**	**99.67**	99.59	99.63	99.47	99.45
CIFAR10	softmax	86.87					
	Taylor	87.29	86.86	87.06	87.17	87.29	87.29
	Taylor_inf	87.46	87.46	87.37	87.34	87.00	87.38
	SM-softmax	87.33					
	SM-Taylor	**87.47**	**87.47**	86.86	87.08	87.08	87.27
CIFAR100	softmax	48.57					
	Taylor	49.94	44.70	49.24	49.94	49.84	49.04
	Taylor_inf	49.81	44.62	47.31	49.81	46.69	45.97
	SM-softmax	48.95					
	SM-Taylor	**49.95**	44.77	**49.95**	49.56	49.69	48.11

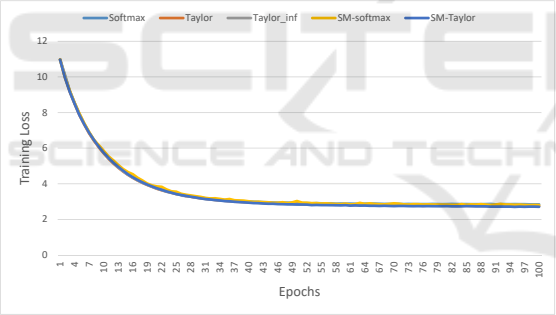

Figure 3: Plot of training loss vs epochs for CIFAR100 dataset.

there has been a lot of research dedicated to finding a better alternative to this popular softmax function. One approach explores the loss functions belonging to the spherical family, and proposes log-Taylor softmax loss as arguably the best loss function in this family (Vincent et al., 2015) Another approach that tries to amplify the discriminative nature of the softmax function, proposes soft-margin (SM) softmax as the most appropriate alternative. In this work, we investigate Taylor softmax, soft-margin softmax and our proposed SM-Taylor softmax as alternatives to softmax function. Moreover, we study the effect of expanding Taylor softmax up to ten terms, in contrast to the original work that expanded only to two terms, along with the ramifications of considering Taylor softmax to be a finite or infinite series during gradient computation. Through our experiments for the image classification task on different datasets, we establish that there is always a configuration of the SM-Taylor softmax function that outperforms the original softmax function and its other alternatives.

In future, we want to explore bigger models and datasets, especially, the ILSVRC2012 dataset (Russakovsky et al., 2015) and its various winning models over the years. Next we want to explore other tasks where softmax is used, for example, image caption generation (Xu et al., 2015) and language translation (Vaswani et al., 2017), and check how well do the softmax alternatives covered in this work perform for the varied tasks. Ideally, we would like to discover an alternative to softmax that can be considered as its drop-in replacement irrespective of the task at hand.

REFERENCES

Brownlee, J. How to develop a CNN from scratch for CIFAR-10 photo classification. https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/ Accessed: 2020-06-21.

Clevert, D., Unterthiner, T., and Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (elus). In *ICLR*.

de Brébisson, A. and Vincent, P. (2016). An exploration

of softmax alternatives belonging to the spherical loss family. In *ICLR*.

Lee, H. B., Lee, J., Kim, S., Yang, E., and Hwang, S. J. (2018). Dropmax: Adaptive variational softmax. In *NeurIPS*.

Liang, X., Wang, X., Lei, Z., Liao, S., and Li, S. Z. (2017). Soft-margin softmax for deep classification. In *ICONIP*, pages 413–421.

Liu, W., Wen, Y., Yu, Z., and Yang, M. (2016). Large-margin softmax loss for convolutional neural networks. In *ICML*, pages 507–516.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*, pages 5998–6008.

Vincent, P., de Brébisson, A., and Bouthillier, X. (2015). Efficient exact gradient update for training deep networks with very large sparse targets. In *NeurIPS*, pages 1108–1116.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057.