International Journal of Internet, Broadcasting and Communication Vol.14 No.1 136-141 (2022)
http://dx.doi.org/10.7236/IJIBC.2022.14.1.136

IJIBC 22-1-17

Beta and Alpha Regularizers of Mish Activation Functions for Machine Learning Applications in Deep Neural Networks

Peter Beatus Mathayo[1] and Dae-Ki Kang[2,*]

[1]Master Student, Department of Computer Engineering, Graduate School, Dongseo University, Korea
[2]Professor, Department of Computer Engineering, Dongseo University, Korea
E-mail: [1]be120.pt@gmail.com, [2]dkkang@dongseo.ac.kr

Abstract

A very complex task in deep learning such as image classification must be solved with the help of neural networks and activation functions. The backpropagation algorithm advances backward from the output layer towards the input layer, the gradients often get smaller and smaller and approach zero which eventually leaves the weights of the initial or lower layers nearly unchanged, as a result, the gradient descent never converges to the optimum. We propose a two-factor non-saturating activation functions known as Bea-Mish for machine learning applications in deep neural networks. Our method uses two factors, beta (β) and alpha (α), to normalize the area below the boundary in the Mish activation function and we regard these elements as Bea. Bea-Mish provide a clear understanding of the behaviors and conditions governing this regularization term can lead to a more principled approach for constructing better performing activation functions. We evaluate Bea-Mish results against Mish and Swish activation functions in various models and data sets. Empirical results show that our approach (Bea-Mish) outperforms native Mish using SqueezeNet backbone with an average precision (AP50val) of 2.51% in CIFAR-10 and top-1accuracy in ResNet-50 on ImageNet-1k. shows an improvement of 1.20%.

Keywords: *Neural Network, Machine Learning Applications, α and β Regularizers, Activation Function.*

1. Introduction

In general, neural network and activation functions are the most important components of deep learning, mainly used to design large-scale neural networks or to determine the accuracy and performance efficiency of deep learning models that can be partitioned. An activation function defines an input or output of a set of inputs, that, the node of the node output given to the input. They basically decide to disable or activate neurons to get the desired output. It also performs non-linear transformations on the input to get better results in complex neural networks. The choice of activation function is essential to understanding the performance of deep neural networks in machine learning applications.

The activation such as sigmoid was extensively used in early literature [1]. The sigmoid function can accept any value, but always computes a value between 0 and 1. Tanh activation function modified the range of

Manuscript Received: December. 11, 2021 / Revised: December. 15, 2021 / Accepted: December. 20, 2021
*Corresponding Author: dkkang@dongseo.ac.kr
Tel: +82-51-320-1724, Fax: +82-51-327-8955
Professor, Department of Computer Engineering, Dongseo University, Korea

computation values of sigmoid to -1 up to +1, however at the end it did not work perfect for deep neural networks to resolve vanishing gradients. Rectified Linear Unit (ReLU) showed better generalization and improved convergence speed compared to Sigmoid and tanh [2,3]. Various alternatives to ReLU have been proposed, these include LReLU, ReLU, ELU and SELU [4-7]. However, none of these variants has managed to replace ReLU due to inconsistent gains.

Figure 1 shows the process of applying an activation function in a layer of a neural network.

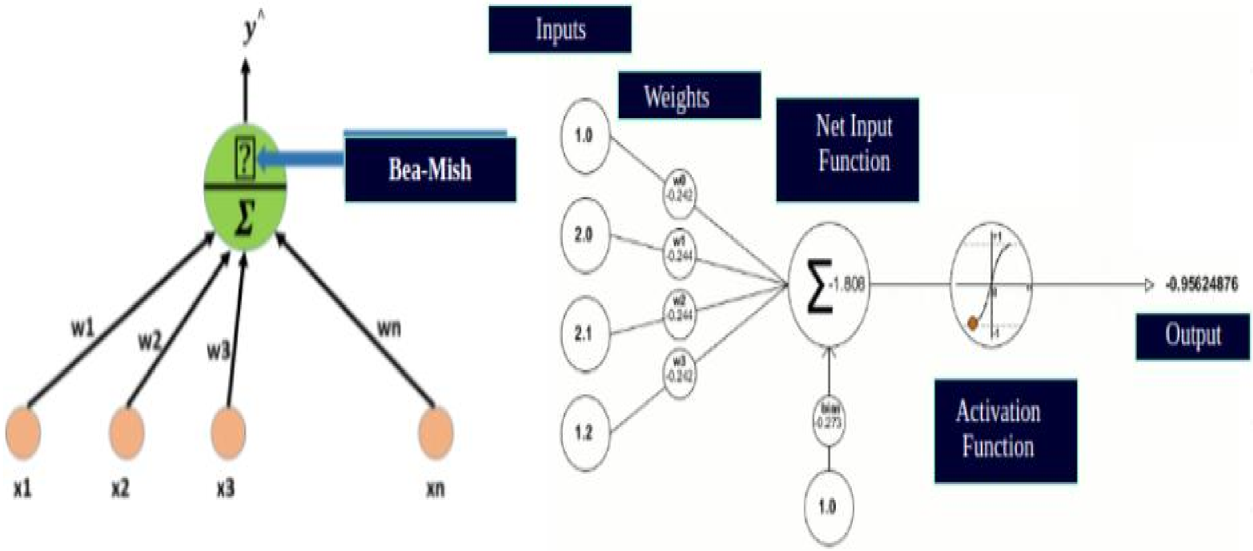

Figure 1. The process of applying an activation function in a layer of a neural network can be mathematically realized as $y = a(z) = a(\sum_i^n wixi + b)$ where y is the output of the activation function a(z)

Recently, a set of non-monotonic activation functions, such as the Gaussian Error Linear Unit (GELU)[8] use Φ(x), known as the standard Gaussian cumulative distribution function. The Google brain team proposed an activation function called Swish that was discovered in a nonlinear function space by Neural Architecture Search (NAS) which use a controlled search agent [9,10]. HardSwish is a kind of activation function based on Swish, but it replaces the computationally expensive sigmoid with a piecewise linear analog [11]. Rectified Exponential Unit (REU) and Mish have been proposed with similar mathematical formulas and show robust and improved results in various models and data sets [12,13]. The main difference between monotonic and non-monotonic activation functions is their non-monotonic properties in the negative quadrant. ReLU is described by equation (1), which is not constrained on the positive axis of the input and sparse on the negative part [2].

$$f(x) = \begin{Bmatrix} x \; if \; x > 0 \\ 0 \; if \; x \leq 0 \end{Bmatrix} \tag{1}$$

From equation (1) x is the input and $f(x)$ is the output (the activation value), ReLU uses identification mapping in the positive quadrant and output 0 in the negative part, identification mapping can alleviate the problem of disappearing gradients. GELU nonlinearity weights inputs by their value, rather than gates inputs by their sign as in ReLUs ($x > 0$) [8].

Mish, a novel self-regularized non-monotonic activation function inspired by the self-gating property of Swish. Mish is mathematically defined as:$f(x) = x \cdot tanh(softplus(x))$ [13]. However, self-regulation did not solve gradient convergence as an optimal problem in deep neural networks.

2. Theory

In this work, we propose to Bea-Mish a two-factor regularized non-monotonic activation function for machine learning applications in deep neural networks to solve gradient descent converges to the optimum challenge. Our method uses two factors, beta (β) and alpha (α), to regularize the region below the boundary in the Mish activation function. In this method, together these factors are called Bea regularizer. Bea-Mish uses a common Mish mathematical expression, but we introduce additional definitions:

$$softplus(r) = \ln(1+e^{r})\, where\; r = e^{\frac{\alpha x}{\sqrt{\beta+x^2}}} \;,\; f(x) = x \cdot tanh\left(\ln\left(1+e^{\frac{\alpha x}{\sqrt{\beta+x^2}}}\right)\right) \qquad (2)$$

First derivative of the equation (3) is described by the equation (6) below.

$$f'(x) = \frac{\frac{\alpha\beta x\sqrt{\beta+x^2}\,e^{\frac{\alpha x}{\sqrt{\beta+x^2}}}}{cosh^2(softplus(x))} + (x^2+\beta)^2(1+e^{\frac{\alpha x}{\sqrt{\beta+x^2}}})\tanh(softplus(x))}{(x^2+\beta)^2(1+e^{\frac{\alpha x}{\sqrt{\beta+x^2}}})} \qquad (3)$$

The value of α is determined by the value of β using the ratio α:β=1:5, and the value of β ranges from 1 to 200. Bea-Mish avoids saturation, which typically slows down training quickly due to near-zero gradients [14].

3. Experiment Settings

We compared Bea-Mish results against Mish and Swish activation functions in various models and data sets. Our results show especially in computer vision tasks such as image classification, Bea-Mish consistently exceeded the highest performing networks. Multiple runs were used to observe the statistical significance of the results. In this section we analyze the statistical results and the ablation study of β and α.

4. Results and Discussion

4.1 CIFAR-10 Dataset on Various Baseline Activation Functions

Bea-Mish activation function compared to baseline activation functions, we compared the mean test accuracy, mean test loss, and standard deviation of test accuracy for CIFAR-10 image classification task using SqueezeNet following the parameter settings in [13, 15, 16]. Table 1 shows that Bea-Mish outperforms the other activation functions with the highest mean accuracy (μ acc), the second lowest mean loss(μ loss), and the third lowest accuracy standard deviation (σ acc).

Table 1. Statistical results of various activation functions for image classification of the CIFAR10 data set using SqueezeNet for 23 rounds and 50 epochs for β= 80 and α= 16

Activation Function	μ acc	μ loss	σ acc
Bea-Mish (Ours)	**90.97%**	**4.05%**	**0.33**
Mish[13]	87.48%	4.13%	0.40
Swish[9]	82.32%	4.22%	0.41

4.2 CIFAR-10 Dataset on Various Standard Neural Network Architectures

We compare the performance of various criterion activation functions for image classification tasks in the CIFAR-10 dataset using different standard neural network architectures [17]. We alternated the activation function and keep all other network parameters and training parameter constant. We evaluate the performance of Bea-Mish compared to Mish and Swish using empirical results such as in native Mish for various standard network architectures [9,13]. Table 2 shows that Bea-Mish activation consistently outperforms Mish, and Swish activation in all standard architectures used in the experiments, providing performance gains up to 1% compared to native Mish network architectures.

Table 2. Comparison between Bea-Mish, Mish and Swish activation functions based on test accuracy on image classification of CIFAR-10 across various network architectures

Architecture	**Bea-Mish(ours)**	Mish	Swish
ResNet-20	**92.69%**	92.02%	91.61%
WRN-10-2	**86.97%**	86.83%	86.83%
SimpleNet	**92.31%**	91.70%	91.44%
Xception Net	**88.94%**	88.73%	88.56%

4.3 ImageNet-1k Dataset on Various Standard Neural Network Architectures

We evaluate the performance of Bea-Mish against native Mish and Swish for ImageNet-2012 dataset classification task. ImageNet is considered one of the most challenging and important classification tasks in the domain of computer vision [18]. ImageNet consists of 1.28 million training images spread across 1,000 classes. To evaluate the performance of the trained network, we use a validation set of 50,000 images. We set the batch size, mini-batch size, initial learning rate, momentum, and weight decay to 128, 32, 0.01, 0.9, and 5e-4 respectively with total number of 6 million training steps. Table 3 results are more consistent with Bea-Mish and generally guaranteed performance improvement in all neural network architecture for ImageNet classification.

Table 3. Comparison of Bea-Mish, Mish and Swish activation functions for image classification on ImageNet-1k datasets on various standard architectures

Model	Augmentation	**Bea-Mish(ours)**		Mish		Swish	
		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ResNet-18[15]	No	**72.40%**	**90.47%**	71.20%	89.90%	71.20%	90.10%
ResNet-50[15]	No	**76.84%**	**93.01%**	76.10%	92.80%	75.90%	92.80%
SpineNet-49[19]	Yes	**73.62%**	**91.02%**	71.40%	90.40%	71.50%	90.70%
CSP-ResNet-50[20]	Yes	**78.45%**	**95.08%**	78.10%	94.20%	-	-

4.4 Ablation Study of α and β on CIFAR-10

We observe the effect of increasing the size of α and β by considering the ratio α: β=1:5 on CIFAR-10 dataset using ResNet framework. The network was optimized using SGD with a batch size of 128 [21]. For a fair comparison, the same learning rate was maintained for the different size of α and β. Table 4 show that the value of β, from 1 to 200, provides high test accuracy with increasing of values α and β in ResNet framework.

Table 4. Effect of increasing α and β values on CIFAR-10 for β (1 to 200)

Model	α, β	Accuracy	Loss	Runtime(s)	Bounded	Non-Linear?
ResNet-20	$\alpha = -1, \beta = -5$	-	-	-	NaN	-
	$\alpha = 1, \beta = 5$	89.80%	0.316	2548	-0.4697 to ∞	Yes
	$\alpha = 7, \beta = 35$	92.47%	0.279	2526	-0.2646 to ∞	Yes
	$\alpha = 16, \beta = 80$	92.69%	0.265	2502	-0.1549 to ∞	Yes
	$\alpha = 40, \beta = 200$	92.81%	0.262	2484	-0.0578 to ∞	Yes
ResNet-20	$\alpha = -1, \beta = -5$	-	-	-	NaN	-
	$\alpha = 1, \beta = 5$	90.46%	0.299	3954	-0.4697 to ∞	Yes
	$\alpha = 7, \beta = 35$	92.97%	0.278	4046	-0.2646 to ∞	Yes
	$\alpha = 16, \beta = 80$	93.08%	0.267	3991	-0.1549 to ∞	Yes
	$\alpha = 40, \beta = 200$	93.56%	0.2564	3940	-0.0578 to ∞	Yes

5. Conclusion

In this work, we propose to Bea-Mish a two-factor regularized non-monotonic activation function for machine learning applications in deep neural networks to solve gradient descent converges to the optimum challenge. We propose the Bea-Mish activation function with two-element regularizer of the Mish activation function. Bea-Mish demonstrates a clear understanding of the behaviors and conditions governing this regularization term may motivate a more principled approach to constructing better performing activation functions. Bea-Mish shares many properties with Mish, such as infinite positive domain, limited negative domain, non-monotonic form, and smooth out-growth. Empirical results show that Bea-Mish outperforms native Mish in all aspects of different datasets and neural architecture models. The future study includes evaluating the performance of the Bea-Mish activation function another state-of-the-art model on various tasks in the domain of computer vision such as object detection [22].

Acknowledgement

This work was supported by Dongseo University, “Dongseo Cluster Project” Research Fund of 2021 (DSU-20210001).

References

[1] J. Kilian and H. Siegelmann, “On the Power of Sigmoid Neural Networks,” in *Proc. 6th Annual Conference on Computational Learning Theory*, pp. 137-143, August 1993.
DOI: https://doi.org/10.1145/168304.168321

[2] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proc. 27th International Conference on Machine Learning*, pp. 807–814, June 2010.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Communications of the ACM*, Vol. 60, No. 6, pp. 84-90, June 2017.
DOI: https://doi.org/10.1145/3065386

[4] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier Nonlinearities Improve Neural Network Acoustic Models,” in *Proc. 30th ICML Workshop on Deep Learning for Audio, Speech, and Language Processing (WDLASL 2013)*, June 16-21, 2013.

[5] K. He, X. Zhang, S. Ren, and J Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” in *Proc. International Conference on Computer Vision (ICCV 2015)*, pp. 1026-1034, Dec

7-13, 2015.
DOI: https://doi.org/10.1109/ICCV.2015.123

[6] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *Proc. 4th International Conference on Learning Representations (Poster)*, May 2016.

[7] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-Normalizing Neural Networks," in *Proc. Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Dec 4-9, 2017.

[8] D. Hendrycks and K. Gimpel, "Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units," *CoRR*, abs/1606.08415, 2016.

[9] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions," *CoRR*, abs/1710.05941, 2017.

[10] B. Zoph, and Q. V. Le, "Neural Architecture Search with Reinforcement Learning," in *Proc. 5th International Conference on Learning Representations*, April 24-26, 2017.

[11] A. Howard et al., "Searching for MobileNetV3," in *Proc. the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314-1324, Oct. 27-Nov. 2, 2019.
DOI: https://doi.org/10.1109/ICCV.2019.00140

[12] Y. Ying et al., "Rectified Exponential Units for Convolutional Neural Networks," *IEEE Access*, Vol. 7, pp. 101633-101640, 2019.
DOI: https://doi.org/10.1109/ACCESS.2019.2928442

[13] D. Misra, "Mish: A Self Regularized Non-Monotonic Activation Function," in *Proc. 31st British Machine Vision Virtual Conference (BMVC)*, Sep. 7-10, 2020.

[14] X. Glorot, and Y. Bengio, "Understanding the Difficulty of Training Deep Feed Forward Neural Networks," in *Proc. 13th International Conference on Artificial Intelligence and Statistics (AISTAT)*, May 13-15, 2010.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, June 27-30, 2016.
DOI: https://doi.org/10.1109/CVPR.2016.90

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognitio (CVPR)n*, pp. 7132-7141, June 18-22, 2018.
DOI: https://doi.org/10.1109/CVPR.2018.00745

[17] A. Krizhevsky, *Learning Multiple Layers of Features from Tiny Images,* Master Thesis. University of Toronto, Canada, 2009.

[18] J. Deng et al., "ImageNet: a Large-Scale Hierarchical Image Database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248-255, June 20-25, 2009.
DOI: https://doi.org/10.1109/CVPR.2009.5206848

[19] X. Du et al., "SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11589-11598, June 13-19, 2020.
DOI: https://doi.org/10.1109/CVPR42600.2020.01161

[20] C.-Y. Wang et al., "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1571-1580, June 14-19, 2020.
DOI: https://doi.org/10.1109/CVPRW50498.2020.00203

[21] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," in *Proc. 19th International Conference on Computational Statistics (COMPSTAT 2010)*, pp. 177-186, Aug. 22-27, 2010.
DOI: https://doi.org/10.1007/978-3-7908-2604-3_16

[22] S. E. Budiman and S. Lee, "Object Tracking with Histogram weighted Centroid augmented Siamese Region Proposal Network," *International Journal of Internet, Broadcasting and Communication*, Vol.13 No.2, pp. 156-165, May 2021.
DOI: http://dx.doi.org/10.7236/IJIBC.2021.13.2.156