SOFTSWISH NEURAL NETWORK APPROXIMATION WITH ZUI-CUI MODULUS

http://dx.doi.org/10.47832/MinarCongress3-7

Hawraa Abbas ALMURIEB[1]
Zainab Abdulmunim SHARBA[2]

Abstract:

Until today, many formulas of neural networks are defined to be used for function approximation, they vary with respect to the weights, activation functions and other standards. Moreover, researchers have studied the approximation of different spaces of functions. In this paper, we approximate functions from multivariate L_p spaces with a neural network with a new defined form of Swish function, named SoftSwish. Also, multivariate Zou-Cui modulus is introduced to express the degree of approximation by our Swish neural network that we call "SoftSwish Neural Network"..

Key words: Approximation, Neural Network, Swish, Modulus of Smoothness.

[1] University of Babylon, Iraq, pure.hawraa.abbas@uobabylon.edu.iq, https://orcid.org/0000-0001-6888-0903
[2] University of Babylon, Iraq, zainab.abd@uobabylon.edu.iq, https://orcid.org/0000-0002-5990-8406

Introduction and Preliminaries:

Artificial Neural Network (ANN) is an artificial intelligent method, that is created to understand the abilities of learning realizing, which is one of the characteristics of human brain to discover new information automatically and generate new information. In recent decades, neural networks have enjoyed a big evolution in the field of functions approximation.

In neural networks, the activation function plays a crucial role. Sigmoid has long been one of the most popular activation functions in neural network behaviour, but its small derivative causes disappearing gradients, making it less ideal for learning, see (X. He & Xu, 2007; Klambauer et al., 2017).

In(Glorot, 2011), the authors introduced another activation function called ReLU. In the positive region, it has an identity derivative and therefore it gives less susceptible to vanishing gradients.Therefore, the function has become the most common recently.

Due to the success of ReLU, numerous adaptations have been proposed in(Bhaya & Sharba, 2020; Dittmer et al., 2019; K. and Z. X. and R. S. and S. J. He, 2015), but did not reach the same popularity of Sigmoids, this is due to the simplicityof the mathematical formulas of each. Moreover, none hasgotten the same tractionbecause gains appear inconsistent or negligible across the data set and models.

There are certain properties that are characterized by the activation functions which are considered important for successful learning, such as derivative and monotonicity, and whether their range finite or not.

Ramachandran et al(Ramachandran, n.d.) introduced the Swish activation function, which is related to Sigmoidal function simply by the formula

$$\sigma_j(x) = x_j.sigmoid(x_j)$$

In their paper, the researchers showed the powerful characters that explains why their Swish should be preferred among other activations including ReLU and the general Sigmoid.

Many moduli of convexity developed and characterized to match the Banach spaces (Zuo & Cui, 2009) and quasi-Banach spaces(Kwun et al., 2018). In the multivariate spaces, the activation function should be defined accurately to generate a multivariate neural network so it could be the approximation of a multivariate function. In (Anastassiou, 2011) defined the the sigmoid activation function in the space of $\prod_{i=1}^{n}[a_i, b_i]$ and approximate the functions from $C(\prod_{i=1}^{n}[a_i, b_i])$ by his SoftMax neural network. The degree of Anastassiou's approximation is in terms of the first modulus of smoothness $\omega_1\left(f, \frac{1}{n}\right)$. Later, the authors in (Almurieb & Bhaya, 2020), defined Soft Max neural network, then they estimate the degree of approximation with the kth order modulus of smoothness $\omega_k\left(f, \frac{1}{n}\right)$.

2. Construction of SoftSwish Neural Networks

The general form of any neural network is given by

$$N_d(x) = \sum_{j=0}^{d} c_j \sigma_j(w_j \cdot x + \mathrm{b}_j) \qquad 2.1$$

b_j, c_j are constants in $\mathbb{R}$, $w_j$are the weights, x_j are the inputs, , σ_j is the activation function.

We begin by defining the new activation function in $\mathbb{R}$
Let

$$\emptyset_j(x_j) = x_j\sigma_j(x_j) + \sigma_j(x_j)\left(1 - x_j\sigma_j(x_j)\right) \qquad 2.2$$

For purposes of function approximation, it is better to use (2.2) than the Sigmoid and Swish functions for several reasons appeared in the properties below,

For any $j = 1, \dots, d, \emptyset_j$ satisfies

(i) $$\sum_{j=1}^{d} \emptyset_j(x_j) \leq 1 \qquad 2.3$$

Proof

$$\sum_{j=1}^{d} \emptyset_j(x_j) = \sum_{j=1}^{d} x_j\sigma_j(x_j) + \sigma_j(x_j)\left(1 - x_j\sigma_j(x_j)\right)$$
$$\leq d\sum_{j=1}^{d} \sigma_j(x_j) + \sum_{j=1}^{d} \sigma_j(x_j)\left(1 - d\sum_{j=1}^{d} \sigma_j(x_j)\right)$$
$$= 1,$$

since that

$$\sum_{j=1}^{d} \sigma_j(x_j) = 1$$

Also, it is easy to prove that

(ii) $$\int_{-1}^{1} \emptyset_j(x_j)d\,x_j = 1 \qquad 2.4$$

The multivariate neural network with inputs $\boldsymbol{x} = (x_1, \cdots, x_d)$from $[-1,1]^d = [-1,1] \times \cdots \times [-1,1]d$-times. The multivariate Swish activation function is given by

$$\emptyset(\boldsymbol{x}) = \emptyset(x_1, \cdots, x_d) = \prod_{j=1}^{d} \emptyset_j(x_j) \qquad 2.5$$

We call $\emptyset(\boldsymbol{x})$ in 2.5, SoftSwish like SoftMax in the sigmoid case.

More properties that are similar to those of univariate case of $\emptyset_j$ are hold for multivariate case of $\emptyset$as follow

$$\text{(i)} \sum_{|\boldsymbol{x}-\boldsymbol{k}|\leq 1} \emptyset(\boldsymbol{x} - \boldsymbol{k}) = \sum_{|x_j-k_j|\leq 1} \prod_{j=1}^{d} \emptyset_j(x_j)$$

$$\leq \prod_{j=1}^{d} \sum_{|x_j-k_j|\leq 1} \emptyset_j(x_j-k_j) \leq 1$$

$$\text{(ii)} \int_{-1}^{1} \emptyset(\boldsymbol{x})d\,\boldsymbol{x} = \int_{-1}^{1} \cdots \int_{-1}^{1} \prod_{j=1}^{d} \emptyset_j(x_j)\,d\,x_j$$

$$\leq \prod_{j=1}^{d} \int_{-1}^{1} \cdots \int_{-1}^{1} \emptyset_j(x_j)d\,x_j = 1$$

We reach to define the multivariate neural network with respect to the function $\boldsymbol{f} \in L_p[-1,1]^d$, where $\boldsymbol{f} = (f_1, \cdots, f_d)$

$$N_d(\boldsymbol{x},\boldsymbol{f}) = \sum_{\boldsymbol{k}} c\boldsymbol{f}(\boldsymbol{k})\ \emptyset(\boldsymbol{x}-\boldsymbol{k}) \qquad 2.6$$

$$= \sum_{k_1} \cdots \sum_{k_d} c \prod_{j=1}^{d} f_j(k_j)\emptyset_j(x_j-k_j)$$

where $c = \frac{\|\boldsymbol{f}(\boldsymbol{x})+t\boldsymbol{f}(\boldsymbol{y})\|_p^p+\|\boldsymbol{f}(\boldsymbol{x})-t\boldsymbol{f}(\boldsymbol{y})\|_p^p}{2nC^p}$

3. Modulus of Smoothness

Moduli of smoothness of functions measures how smooth the function is. Many mathematicians defined several types of moduli. The most important is that Ditzian and Totik. In 1980 Ivanov defined moduli which he characterized as the best algebraic approximation in *Lp* space see (Klambauer et al., 2017).

We choose Zuo-Cui modulus in the quasi-Banach space from(Zuo & Cui, 2009). Actually, we need to define the multivariate Zuo-Cui modulus for functions f from L_p space as follow

$$\zeta^{(k)}(\boldsymbol{f},t) = \sup_{x,y\in[-1,1]^d} \left\{ \frac{\|\boldsymbol{f}(\boldsymbol{x}+t\boldsymbol{y})\|_p^p + \|\boldsymbol{f}(\boldsymbol{x}-t\boldsymbol{y})\|_p^p}{2C^p} \right\} \qquad 3.1$$

where $0 < t < 1$.

4. Application to Function Approximation by SoftSwish Neural Network

Neural networks have been widely used in the field of approximation of functions. The following theorem is the existence theorem of such a neural network that approximate $L_p$functions. The degree of neural approximation is estimated here in terms of modulus of smoothness.

Theorem

For anyconvex function $\boldsymbol{f} \in L_p[-1,1]^d$, there is a neural network of the form (2.1) s.t.

$\|\boldsymbol{f}-\boldsymbol{N}\|_p^p \leq C\zeta^{(p)}(t)$

Proof

Since f is convex, then

$f(x+ty) \leq f(x) + tf(y)$

So by 2.1-2.5 and 3.1, we have

$$\|f-N\|_p^p = \left\| \sum_k cf(k)\,\emptyset(x-k) - f(x) \right\|_p^p$$

$$\leq \sum_k \int_{-1}^{1} |cf(k)\,\emptyset(x-k) - f(x)|^p\,dx$$

$$\leq \sum_k \frac{\|f(x)+tf(y)\|_p^p + \|f(x)-tf(y)\|_p^p}{2nC^p} \int_{-1}^{1} |f(k)-f(x)|^p |\emptyset(x-k)|^p\,dx$$

$$\leq \sum_{k_1} \cdots \sum_{k_d} \frac{\|f(x)+tf(y)\|_p^p + \|f(x)-tf(y)\|_p^p}{2nC^p} \prod_{j=1}^{d} \int_{-1}^{1} |f_j(k_j) - f_j(x_j)|^p |\emptyset(x_j-k_j)|^p\,dx_j$$

$\leq C(p,|k|)\zeta^{(p)}(f,t)$ ■

Conclusions

We realize that we can use neural networks with an appropriate activation function for approximating continuous functions. In our work, we introduce function approximation on Lp space by using a new multivariate formula of Swish activation function. New version of direct inequality using neural networks with generalized Swish activation function in terms of Zuo-Cui modulus of smoothness can be estimated. It open doors wide for more theorems and applications.

References

X. He and S. Xu, Process Neural Networks. Springer, 2007.

G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, “Self-Normalizing Neural Networks,” 2017.

X. A. B. and Y. B. Glorot, “Deep sparse rectifier neural networks,” Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, vol. 15, pp. 315–323, 2011.

S. Dittmer, E. J., and P. Maass, “Singular Values for ReLU Layers,” IEEE Transactions on Neural Networks and Learning Systems, no. 2, pp. 1–12, 2019, doi: 10.1109/tnnls.2019.2945113.

K. and Z. X. and R. S. and S. J. He, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

E. S. Bhaya and Z. A. Sharba, “L_p_ approximation by ReLU neural networks,” Karbala International Journal of Modern Science, vol. 6, no. 4, pp. 414–419, 2020, doi: 10.33640/2405-609X.2362.

P. and Z. B. and L. Q. v Ramachandran, “Searching for activation functions,” arXiv preprint arXiv:1710.05941.

Z. Zuo and Y. Cui, “A coefficient related to some geometric properties of a banach space,” Journal of Inequalities and Applications, vol. 2009, 2009, doi: 10.1155/2009/934321.

Y. C. Kwun, H. M. U. Ahmad Qadri, W. Nazeer, A. U. Haq, and S. M. Kang, “On Generalized Moduli of Quasi-Banach Space,” Journal of Function Spaces, vol. 2018, 2018, doi: 10.1155/2018/7324783.

G. A. : Anastassiou, “Multivariate Sigmoidal Neural Network Approximation,” Neural Networks, vol. 24, pp. 378–386, 2011.

H. A. Almurieb and E. S. Bhaya, “SoftMax Neural Best Approximation,” 2020, doi: 10.1088/1757-899X/871/1/012040.