Target detection based on a new triple activation function

Guanyu Chen\textsuperscript{a}, Quanyu Wang\textsuperscript{a}, Xiang Li\textsuperscript{b} and Yanyun Zhang\textsuperscript{c}

\textsuperscript{a}Informatization Office, China University of Geosciences, Wuhan, People’s Republic of China; \textsuperscript{b}School of Computer Science, China University of Geosciences, Wuhan, People’s Republic of China; \textsuperscript{c}School of Computer Science and Information Engineering, Hubei University, Wuhan, People’s Republic of China

\textbf{ABSTRACT}
As one of the important parts of Neural Network, activation function plays a very important role in model training in Neural Network. In this paper, the status quo, advantages and disadvantages of the existing common activation functions are analysed, and a new activation function is proposed and applied to target detection. To test the performance of the new activation function, this paper compares it with the ReLU activation functions on a variety of Neural Networks and data sets, and not only analyses the performance of the activation function itself but also verifies the effectiveness of the activation function in target detection.

\textbf{ARTICLE HISTORY}
Received 3 January 2022
Accepted 14 June 2022

\textbf{KEYWORDS}
Target detection; neural network; activation function

1. Introduction

The purpose of generic target detection (Jiao et al., 2019; Wu et al., 2020; Zou et al., 2019) is to determine the location of target instances in natural images based on a large number of predefined categories. This is one of the most basic and challenging problems in the computer vision field. The deep learning that has emerged in recent years is a powerful method that can learn features directly from data and has made significant breakthroughs in the field of general target detection. There are many Neural Network models in deep learning, including Perceptron Neural Network (PNN) (Du et al., 2007), Back Propagation (BP) Neural Network (Huang, 1999), Radical Basis Function (RBF) Neural Network (Du et al., 2006; Huang \& Du, 2008), Feedforward Neuron Network (FNN) (Han \& Huang, 2008), Extreme Learning Machine (ELM) (Han \& Huang, 2006) and so on, which have been widely used in the field of computer vision.

Target detection is a long-standing fundamental problem in the field of computer vision, and it has been an active research field for decades. Given an image, target detection can determine whether there are target instances of a given category (such as people, cars, bicycles, dogs and cats); if so, return the spatial location and coverage of each target instance (such as Return an anchor box Ren et al., 2015). As the cornerstone of image understanding and computer vision, target detection is the foundation for solving more complex and higher-level vision tasks such as instance segmentation, scene understanding, target tracking, image description, event detection and activity recognition (Han et al., 2010). Target detection has a wide range of applications in many fields of artificial intelligence and information technology, including robotic vision, consumer electronics, security, autonomous driving (Han et al., 2008), human–computer interaction, content-based image retrieval, intelligent video surveillance and augmented reality.

An ideal target detector should have both a high precision and high efficiency. Among them, the high precision mainly includes the accuracy of locating and recognition of the target, and the main problem is that different instances of the same type of target often have different colours, shapes, postures, etc., and the detection effect is largely affected by the background. For example, in pedestrian detection, a pedestrian target on a sunny day is often easier to detect than a pedestrian target in heavy fog. In addition, image noise is also an important factor affecting detection accuracy. The high efficiency mainly includes time efficiency, memory efficiency and storage efficiency in the detection process. In the target detection process, especially in the field of real-time target detection, the requirements for time efficiency are very high. For example, in the field of autonomous driving, real-time detection of various targets in front of the vehicle is required. If the frame number of the front camera is 30 FPS, the detection frame number of the target detector is required to reach the corresponding level, otherwise...
it will be difficult to be applied in the real world. Since a large number of targets will be encountered during the detection process, and each target needs to be identified and located, the memory and storage level of the detector should also be highly efficient.

At present, algorithms in the field of target detection are mainly divided into two categories, namely algorithms based on the One-stage framework and algorithms based on the Two-stage framework. Considering that not all regions contain image targets to be detected, the Two-stage framework performs target detection in two steps. First, this type of algorithms first selects some Region Proposals (RP) (Girshick et al., 2014). These RP have a greater probability of containing targets, and it is more meaningful to run a convolutional network classifier on these regions. Second, these algorithms only choose to run the convolutional network classifier on a few windows, instead of running the detection algorithm for each sliding window. Obviously, one of the shortcomings of this type of algorithm is its slow detection speed, and for this problem, there are currently a series of improved algorithms for this type of algorithm, mainly represented by Region Convolutional Neural Network (R-CNN) (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2015), Feature Pyramid Network (FPN) (Lin, Dollár, et al., 2017), etc. Another type of target detection algorithm is based on the One-stage framework. There is no concept of RP in this type of algorithm, and the detection algorithm only needs to be run once with the convolutional network to get the result. The advantage of this type of algorithm is that the detection speed is faster, but the accuracy is lower than that of the Two-stage detection algorithm. Its main representatives are You Only Look Once (YOLO) (Redmon et al., 2016), Single Shot MultiBox Detector (SSD) (Liu et al., 2016), RetinaNet (Lin, Goyal, et al., 2017) and so on. Gai et al. (2021) proposed an improved Tiny YOLOv3 algorithm with both lightweight and high accuracy of object detection. Li et al. (2021) proposed a combined detection method based on deep learning to further improve the accuracy of carter detection.

The activation function is also called nonlinear mapping and serves as a decision function. It is introduced to increase the expressive ability of the entire network (i.e. nonlinearity) and helps in learning of intricate patterns. The stack of several linear operation layers can still only play the role of linear mapping and cannot form complex functions. Commonly used activation functions are Sigmoid, Tanh, ReLU and so on (Gu et al., 2018; Nwankpa et al., 2018). In the work of Ertuğrul (2018), a simpler and a more effective approach to determine optimal activation function was proposed, in which an activated function was trained for each particular neuron by linear regression. Lau and Lim (2018, December) reviewed three categories of activation functions in DNN i.e. saturated, unsaturated and adaptive activation functions. Misra (2019) proposed a self-regularized non-monotonic neural activation function, with which the results of a Neural Network task were improved compared to Swish and ReLU. Hayou et al. (2019, May) gave a comprehensive theoretical analysis of the Edge of Chaos, and showed that the training could be accelerated and the performance could be improved by tuning the initialization parameters and the activation function. Maguolo et al. (2021) proposed an ensemble of Convolutional Neural Networks which were trained with different activation functions, and improved the performance in small/medium sized biomedical datasets. To overcome vanishing gradient and negative region problems, Kilianslan and Celik (2021) proposed an ensemble of Convolutional Neural Networks and Section 4 is the summary.

This paper aims to design a new activation function by studying the existing activation functions, which can improve the detection efficiency under the condition of ensuring the accuracy of target detection. The proposed activation function has been applied on both the One-stage and Two-stage algorithms, and experiments are carried out on the algorithms to validate its performance. The content of this paper is organized as following: Section 2 is the introduction of the activation function design, the experiments are shown in Section 3 and Section 4 is the summary.

2. The triple activation function

In this section, three classic activation functions (namely Sigmoid, Tanh and ReLU) are briefly introduced first, and then the design of the proposed activation function, namely the Triple activation function, is described.
2.1. Common activation functions

2.1.1. Sigmoid

The formula of Sigmoid function is as following, and the function graph is shown in Figure 1.

\[ f(x) = \frac{1}{1 + e^{-x}} \]  

(1)

The Sigmoid function maps the number between minus infinity and infinity to the interval (0,1). According to Figure 1, it can be found that when the input less than \(-5\), the gradient of the function is gradually close to 0, on the contrary, when the input is greater than 5, similar situations can be observed. The phenomenon is also called the vanishing gradient, which leads to a very slow learning process and the deep Neural Network is basically unavailable. In addition, the output value of the Sigmoid function is always greater than 0, which will lead to slower convergence speed of model training.

2.1.2. Tanh

The formula of the Tanh function is shown in Equation (2), and Figure 2 is the corresponding graph.

\[ f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \]  

(2)

From Figure 2, it can be observed that when the input is smaller than \(-2\) or greater than 2, the vanishing gradient is also exist as shown in Sigmoid function. And the difference from Sigmoid is that Tanh is zero-centred.

2.1.3. ReLU

The ReLU function, also known as the Rectified Linear Unit, and it is a piecewise linear function. The formula and graph of the ReLU function are as follows (Figure 3):

\[ f(x) = \max(0, x) \]  

(3)

Since it only needs to judge if the input is greater than 0 and only has a linear relationship, whether it is forward propagation or backward propagation, the calculation speed is much faster than Sigmoid and Tanh (Sigmoid and Tanh need to calculate exponents). And the convergence speed is also faster than Sigmoid and Tanh.

When the input is positive, it compensates for the vanishing gradient of the Sigmoid function and the Tanh function. While if the input is negative, the vanishing gradient also exists. What’s more, the ReLU function is also not zero-centred.

2.2. The triple activation function

Currently, the Sigmoid and Tanh activation functions are commonly used in deep convolutional Neural Networks, both of which are prone to the problem of vanishing gradient. The mean output of the Sigmoid function is 0.5, is 0.5, and the non-zero mean output will make the input of the neurons of the next layer distributed on both sides with 0.5 as the centre. For the Sigmoid function, it is better to give an input on both sides with 0 as the centre, since the problem of vanishing gradient will not occur at this time. At the same time, the input should not be completely distributed in the linear region of the image, which can be achieved by adjusting the parameters of Batch Norm (BN). ReLU and its improved method solve the problem of vanishing gradient, but they are all non-zero mean. Therefore, this paper designs a new Triple Activation Function, which does not have the problem of vanishing gradient, and the output is zero mean. The specific form of the Triple activation function is shown.
in Equation (4) and the corresponding graph is shown in Figure 4.

\[ f(x) = \alpha x^3 \]  

(4)

where \( \alpha \) is an extra parameter. Compared with Sigmoid, Tanh and ReLU series, this activation function not only solves the problem of vanishing gradient but also has zero mean value.

3. Experiments

All experiments in the paper are performed under the CentOS 7 operating system environment. The hardware includes dual-core CPU and 64G memory. It is also equipped with NVIDIA TESLA K80 GPU graphics card, with a memory capacity of 24G. The deep learning framework used is Tensorflow and Keras. To test the performance of the Triple activation function, the experiments selected the commonly used data sets in image classification, namely MNIST and CIFAR10, for testing, and used two classic convolutional Neural Networks, AlexNet (Krizhevsky et al., 2017) and VGG16 (Simonyan & Zisserman, 2014), for comparison. In the experiments, the four values of parameter \( \alpha \) in 0.1, 0.5, 1, 2 were compared and analysed. The following are the specific experimental combinations and results. Since the effect of ReLU is generally better than Sigmoid and Tanh, and the application of ReLU activation function is more extensive, therefore, only ReLU is selected for comparison in the experiment.

3.1. Experiment on MNIST with AlexNet

Since the images in the data set MNIST are 28 \( \times \) 28 grayscale images, it is relatively easy to train. Therefore, in the experiment, only 20 epochs are trained on the MNIST data set. The experimental results are shown in Figure 5.

Figure 5 reflects the change of Loss with the number of epochs during the training of MNIST. According to the experimental results, it can be found that when \( \alpha = 1 \) is selected, the Loss value of the Triple activation function decreases the fastest, that is, the network converges the fastest, but the Loss value of ReLU and the proposed activation function will eventually drop to a very close value. This shows that the proposed activation function has a great influence on the convergence speed of the network, but hardly affects the accuracy of the experiment.

3.2. Experiment on MNIST with VGG16

In this experiment, the classic convolutional Neural Network VGG16 is used instead of AlexNet, the number of epochs is still set to 20, and the changing curves of the trained Loss with Epoch are shown in Figure 6.

Compared with the results shown in Figure 5, the Loss value obtained by training with VGG16 is relatively small on the whole, and from the experimental results, it can also be found that the Loss of the Triple activation function drops the fastest when \( \alpha = 1 \) is selected.

According to the above two experimental results, the network converges fastest when the activation function
3.3. Experiment on CIFAR10 with AlexNet

The CIFAR10 data set contains color images, and the size is 32 \times 32, which is more complicated than the images in MNIST. In the experiment, the number of epochs is set to 50, and the result is shown in Figure 7.

From the experimental results in Figure 7, it can be seen that the activation function that makes the fastest convergence rate is not the Triple activation function designed in this paper, but the commonly used ReLU. And the Loss during the training process of the data set CIFAR10 is larger than the Loss on MNIST. According to the downward trend of the curve in the figure, it is expected that the Loss value still has room for decline.

3.4. Experiment on CIFAR10 with VGG16

According to the experimental results in Figure 8, when the VGG16 network is used to train the CIFAR10 data set, the fastest convergent is still achieved by the Triple activation function with \( \alpha = 1 \), and the overall Loss is smaller than that when using AlexNet, but it also has further drop space.

From all of the above experimental results, it can be seen that although the results in the experiment of Section 3.3 are abnormal, the overall effect of the Triple activation function with \( \alpha = 1 \) for classification network training is better than ReLU. The setting of the parameter \( \alpha \) of the proposed activation function will also affect the experimental results. The above four experiments prove that if the wrong parameter value is selected, it will seriously affect the convergence process of the network. The experimental results at 0.1, 0.5, and 2 are not even as good as ReLU. The adjustment of parameter in deep learning is often difficult, and there may be better values, but more experimental verification is needed in the future.

3.5. Analysis between triple and ReLU

In the experiment, the Two-Stage detection framework Faster RCNN and One-Stage detection framework RetinaNet were used as the basic detection frameworks, and two underlying networks, ResNet-50 and ResNet-101, were selected. The data set used for training and testing is the MS COCO, and the number of iterations of the experiment is 10,000. The underlying network uses the Triple activation function and ReLU activation function for experimental comparison. The comparison results are all shown in form of detection accuracy, and are presented as a percentage. Some terms involved in the experiment are explained in Table 1.

The experimental results are shown in Table 2. It can be seen from the experimental results that for different underlying networks and detection frameworks, the results of using the ReLU activation function and the Triple activation function on AP are not much different, and the difference between the two activation functions on various AP values is basically maintained within 1%.
Therefore, it can be proved that using the Triple activation function for the underlying network in the target detection framework hardly affects the detection accuracy. However, through the previous experimental results, it can be seen that the Triple activation function mainly affects the convergence speed of the algorithm during the training process. Although the Triple activation function has little effect on improving the detection accuracy, the Triple activation function can be used to rapidly network convergence to accelerate the process of the experiment and reduce the number of iterations in the training process.

4. Conclusion

In this paper, the commonly used activation functions in Neural Networks were discussed at first, and then a new Triple activation function is proposed to get rid of the vanishing gradient and non-zero-centred. To verify the performance of the Triple activation function, it is applied to a variety of Neural Networks on different types of data sets. It has been proven that the Triple activation function with $\alpha = 1$ has a faster convergence speed than ReLU overall, for classification network training on data set MNIST and CIFAR10 with AlexNet and Vgg16. And for the targets detection, it can be observed that the proposed Triple activation function can almost match ReLU. However, even though the extra introduced parameter $\alpha$ has been simply discussed, a further investigation should be done to improve the performance of the Triple in the future.

Disclosure statement

No potential conflict of interest was reported by the authors.

Data availability statement

Data sharing is not applicable to this article as no new data were created or analysed in this study.

References

Apicella, A., Donnarumma, F., Isgrò, F., & Prevete, R. (2021). A survey on modern trainable activation functions. *Neural Networks*, 138(2021), 14–32. https://doi.org/10.1016/j.neunet.2021.01.026.

Du, J. X., Huang, D. S., Wang, X. F., & Gu, X. (2007). Shape recognition based on neural networks trained by differential evolution algorithm. *Neurocomputing*, 70(4–6), 896–903. https://doi.org/10.1016/j.neucom.2006.10.026

Du, J. X., Huang, D. S., Zhang, G. J., & Wang, Z. F. (2006). A novel full structure optimization algorithm for radial basis probabilistic neural networks. *Neurocomputing*, 70(1–3), 592–596. https://doi.org/10.1016/j.neucom.2006.05.003

Ertuğrul, Ö. F. (2018). A novel type of activation function in artificial neural networks: Trained activation function. *Neural Networks*, 99(2018), 148–157. https://doi.org/10.1016/j.neunet.2018.01.007.

Gai, W., Liu, Y., Zhang, J., & Jing, G. (2021). An improved tiny YOLOv3 for real-time object detection. *Systems Science & Control Engineering*, 9(1), 314–321. https://doi.org/10.1080/21642583.2021.1901156

Girshick, R. (2015). Fast R-CNN. In 2015 IEEE international conference on computer vision (ICCV) (pp. 1440–1448). IEEE. https://doi.org/10.1109/ICCV.2015.169.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In 2014 IEEE conference on computer vision and pattern recognition (pp. 580–587). IEEE. https://doi.org/10.1109/CVPR.2014.81.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1), 142–158. https://doi.org/10.1109/TPAMI.2015.2437384

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77(2018), 354–377. https://doi.org/10.1016/j.patcog.2017.10.013.

Han, F., & Huang, D. S. (2008). A new constrained learning algorithm for function approximation by encoding a priori information into feedforward neural networks. *Neural Computing and Applications*, 7(5–6), 433–439. https://doi.org/10.1007/s00521-007-0135-5

Han, F., Ling, Q. H., & Huang, D. S. (2008). Modified constrained learning algorithms incorporating additional functional constraints into neural networks. *Information Sciences*, 178(3), 907–919. https://doi.org/10.1016/j.ins.2007.09.008

Han, F., Ling, Q. H., & Huang, D. S. (2010). An improved approximation approach incorporating particle swarm optimization and a priori information into neural networks. *Neural Computing & Applications*, 9(2), 255–261. https://doi.org/10.1007/s00521-009-0274-y

Han, F., & Huang, D. S. (2006). Improved extreme learning machine for function approximation by encoding a
priori information. Neurocomputing, 69(16–18), 2369–2373. https://doi.org/10.1016/j.neucom.2006.02.013

Hayou, S., Doucet, A., & Rousseau, J. (2019, May). On the impact of the activation function on deep neural networks training. In International conference on machine learning (pp. 2672–2680). PMLR. https://doi.org/10.1016/j.neunet.2022.05.030.

Hu, H., Liu, A., Guan, Q., Li, X., Chen, S., & Zhou, Q. (2021). Adaptively customizing activation functions for various layers. IEEE Transactions on Neural Networks and Learning Systems, 1–12. https://doi.org/10.1109/TNNLS.2021.3133263.

Huang, D. S. (1999). Radial basis probabilistic neural networks: Model and application. International Journal of Pattern Recognition and Artificial Intelligence, 13(7), 1083–1101. https://doi.org/10.1142/S0218001499000604

Huang, D. S., & J. X Du (2008). A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. IEEE Transactions on Neural Networks, 19(12), 2099–2115. https://doi.org/10.1109/TNN.2008.2004370

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. IEEE Access, 7(2019), 128837–128868. https://doi.org/10.1109/ACCESS.2019.2939201.

Kiliarslan, S., & Celik, M. (2021). Rsigelu: A nonlinear activation function for deep neural networks. Expert Systems with Applications, 174(2), Article 114805. https://doi.org/10.1016/j.eswa.2021.114805

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90. https://doi.org/10.1145/3065386

Lau, M. M., & Lim, K. H. (2018, December). Review of adaptive activation function in deep neural network. In 2018 IEEE-EMBS conference on biomedical engineering and sciences (IECBES) (pp. 686–690). IEEE. https://doi.org/10.1109/IECBES.2018.8626714.

Li, H., Jiang, B., Li, Y., & Cao, L. (2021). A combined method of crater detection and recognition based on deep learning. Systems Science & Control Engineering, 9(sup2), 132–140. https://doi.org/10.1080/21642583.2020.1852980

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In 2017 IEEE conference on computer vision and pattern recognition (CVPR) (pp. 936–944). IEEE. https://doi.org/10.1109/CVPR.2017.106.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2), 318–327. https://doi.org/10.1109/TPAMI.2018.2858826.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). SSD: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer vision – ECCV 2016. Lecture Notes in Computer Science. Vol. 9905. Springer. https://doi.org/10.1007/978-3-319-46448-0_2.

Maguolo, G., Nanni, L., & Ghidoni, S. (2021). Ensemble of convolutional neural networks trained with different activation functions. Expert Systems with Applications, 166(2021), Article 114048. https://doi.org/10.1016/j.eswa.2020.114048.

Misra, D. (2019). Mish: A self regularized non-monotonic neural activation function. arXiv preprint arXiv:1908.08681, 4, 2.

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. arXiv Prepr arXiv181103378.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In 2016 IEEE conference on computer vision and pattern recognition (CVPR) (pp. 779–788). IEEE. https://doi.org/10.1109/CVPR.2016.91.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Wu, X., Sahoo, D., & Hoi, S. C (2020). Recent advances in deep learning for object detection. Neurocomputing, 396(2020), 39–64. https://doi.org/10.1016/j.neucom.2020.01.085.

Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object detection in 20 years: A survey. arXiv Prepr arXiv:1905.05055v2.