Adaptive Activation Functions For Artificial Neural Networks

Marakhimov A.R., Khudaybergenov K.K., Ohundadaev U.R.

Abstract: Activation functions are considered as main component in artificial neural networks. The current paper considers learning activation functions with combination of activation functions. We propose two approaches to use activation functions and construction of adaptive activation parameters to input data. Namely, to show effectiveness, we investigate linear form and non-linear form to combine activation functions, then introduce adaptive activation function. Numerical experiments show the proposed activation techniques overcome by performances and accuracy than standard rectified unit family functions.

Index Terms: artificial neural networks, classification, activation function, adaptive activation, convolution.

1. INTRODUCTION

In the last two decades a lot of effort can be seen in development and implementation of convolutional neural networks (CNN) in solving practical problems, such as pattern recognition [1-3,17], face detection [4], face recognition [5,6], object detection [7] and tracking [8,9]. In order to get best accuracy in neural networks, choosing most suitable activation functions are very important. Among them, non-saturated activation functions are more accurate rather than saturated forms in deep learning models, which have shown themselves a successful of deep models. In recent researches, saturated forms of these functions are used more compared to non-saturated variants. Meanwhile, one problem exists with some of these functions, which gradients can to vanish when using saturated forms. However, these type of activation functions can be very helpful to accelerate training the deep learning models. Generally, in artificial neural networks, there are lot of different activation functions [34]. Among them, application of non-saturated activation functions in CNN models, such as image classification and face recognition, have grown considerably. Hence, a saturated activation function, which is called rectified linear unit (ReLU) [10] is defined as follows:

$$f_{relu}(x) = Max(0, x), \quad x \in R. \qquad (1)$$

This is a basic form in rectified unit family functions, which is equal to zero its negative side and its positive side is a simple identity function. Researchers propose a leaky ReLU activation function [12], a similar activation function on the base (1) with introducing a predefined parameter:

$$f_{prelu}(x) = Max(0, \alpha x), \quad \alpha \in (0,1], \; x \in R. \qquad (2)$$

Compared with a ReLU activation function, with a leaky ReLU (LReLU), we can give a relatively minor and predefined angular coefficient to its negative side.

With a generalization of activation functions of these type, the parametric ReLU [13] (PReLU) activation function can provide the angular coefficient to be trained from dataset and enhances the representation capability. Unlike with the property of linearness in aforementioned functions, ELU [14] exponentially decreases the steepness from a predetermined parameter to zero, and this property is useful to accelerate model training. Also, we introduce a parametric form for ELU activation function, which is called a parametric ELU (PELU) activation function.

By learning these basic activation functions, we can do approximation convex and non-convex functions with a piecewise linear activation function with an adaptive parameter [23]. In the current note, we consider ReLU activation function with various configurations. In most successful applications on CNN models, this type of the rectified unit family functions has been extensively used [24-27]. Meanwhile, in these models, most researchers used only basic forms of these activation functions, which is described above. Consequently, according to their simple forms of these activation functions, they have a very restricted representation ability of learning nonlinear transformations. Introducing some combinations with these activation functions, we propose new activation functions to increase the capability of learning non-linear representation and also be adaptive to all input signals. We rely on the new constructed activations techniques on the base activation functions have more flexible forms which can be defined learning from a dataset. Firstly, we introduce two approaches to join these activation functions into one which is given in (1). The first one is depicted as joined function, in which the activation operation is learned by linearly joining basic activation functions. In other word to construct activations functions be adaptable to the specific inputs, another adaptive activation function is proposed. The main idea is that the activation factor is obtained by non-linear joining the base activation functions of a rectified unit type. Secondly, to check the efficiency of the proposed approaches, we perform the several tests on CNN models on well-known benchmark datasets which is available on repository. Our contribution of the work as follows: we propose a new types of activation function on the base of rectified unit family, a joined activation and an adaptive activation to linear and non-linear joining of basic activation functions. The proposed activations functions increase the capability of learning non-linear mappings, and have ability of adaptable to the specific inputs. We constructed and investigated several deep learning CNN models,

- *Marakhimov Avazjon Rakhimovich, professor, Doctor of Technical Sciences, National University of Uzbekistan, Tashkent, Uzbekistan. E-mail: avaz.marakhimov@yandex.ru*
- *Khudaybergenov Kabul Kadibergenovich is currently pursuing PhD degree program in Artificial Intelligence in National University of Uzbekistan, Tashkent, Uzbekistan. E-mail: kabul85@mail.ru*
- *Ohundadaev Ulugbek is currently pursuing PhD degree program in Artificial Intelligence in National University of Uzbekistan, Tashkent, Uzbekistan.*

substituting classical activation functions with the proposed activation functions, which gives a high performance and accuracy. We have tested on the proposed activations on models with well-known datasets like MNIST [28,31], CIFAR [29,32], and ImageNet [30,33].

2. ACTIVATION FUNCTIONS OF THE CNN

In the rectified unit type functions, there exists a zero-type function that could be considered as a particular type of both ReLU and ELU activation functions. In this paper, we consider these two types in order to construct new activation functions.
In general, linear and exponential activation function are defined as follows:

$$f_{lrelu}(\mathrm{x}) = \begin{cases} x, & if \;\; x>0 \\ \alpha x, & if \;\; x \le 0, \end{cases} \quad f_{elu}(\mathrm{x}) = \begin{cases} x, & if \;\; x>0 \\ \alpha\left(e^{x}-1\right), & if \;\; x \le 0, \end{cases} \tag{3}$$

where x is input value for linear ReLU $f_{lrelu}(\cdot)$ and exponential ReLU $f_{elu}(\cdot)$ activation functions. α is a predefined hyper-parameter for controlling the angular coefficient. When parameter value of is $\alpha = 0$, then positive part of activation functions has the same value, and both $f_{lrelu}(\cdot)$ and $f_{elu}(\cdot)$ become $f_{relu}(\cdot)$ function. If α become trainable hyper-parameter, then $f_{lrelu}(\cdot)$ and $f_{elu}(\cdot)$ functions accordingly turns to $f_{prelu}(\cdot)$ and $f_{pelu}(\cdot)$. In order to make simple in use, we use as $f_{prelu}(\cdot)$ and $f_{pelu}(\cdot)$. For these activation functions, the backpropagation error to the previous layer and the gradients with respect to the hyper-parameter α are computed as follows:

$$\frac{\partial f_{prelu}(x)}{x} = \begin{cases} 1, & if \;\; x>0 \\ \alpha, & if \;\; x \le 0, \end{cases} \quad \frac{\partial f_{prelu}(x)}{\alpha} = \begin{cases} 0, & if \;\; x>0 \\ x, & if \;\; x \le 0, \end{cases} \tag{4}$$

$$\frac{\partial f_{pelu}(x)}{x} = \begin{cases} x & if \;\; x>0 \\ \alpha e^{x} & if \;\; x \le 0, \end{cases} \quad \frac{\partial f_{pelu}(x)}{\alpha} = \begin{cases} 0 & if \;\; x>0 \\ e^{x}-1 & if \;\; x \le 0, \end{cases} \tag{5}$$

3. PROPOSED ADAPTIVE ACTIVATION FUNCTIONS

3.1. Activation function

Convolutional Neural Networks (CNN) are part of artificial neural networks, known as Deep Training. These networks are able to perform classification with real-time video streaming, images, big data, etc. Convolutional neural networks are commonly structured whose components consists of convolution, activation and pooling layers. In this paper, we give training and adaptation into the activation operation in CNN.

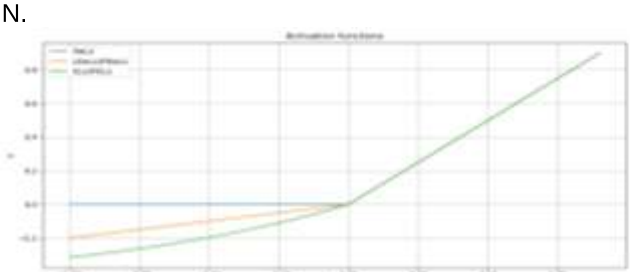

***Fig. 1**. Visual representation of basic activation functions. In all functions, the positive side is the same.*

First, we introduce an activation function which is constructed with basic LReLU and ELU functions, whose parameters is used as only constants, and give two approaches to combine some activation functions into one. The first approach comes with combination parameter, so this combined activation function is not adaptable to the particular input signals.
Training procedure in this approach results to a fixed combination of $f_{lrelu}(\cdot)$ and $f_{elu}(\cdot)$ activation functions. We name this approach as joined activation function. The next approach is adaptable to the particular network input signals. Training procedure in this approach results to a trained pattern (template) which defines an adaptively combining of function to show adaptability to the particular entry data. To mention the significance of the gating pattern, this approach is referred to as an adaptive function. In general, these two approaches employ joining of basic activation functions with a predetermined hyper-parameters.

3.2. Joining basic activation functions

Firstly, we consider a new approach, which is called a combined type of activation function. Then, we introduce adaptive approach to activation functions. In this approach, basic activation functions of neural network are linearly joined into one activation function and combination coefficients are trained during training process.

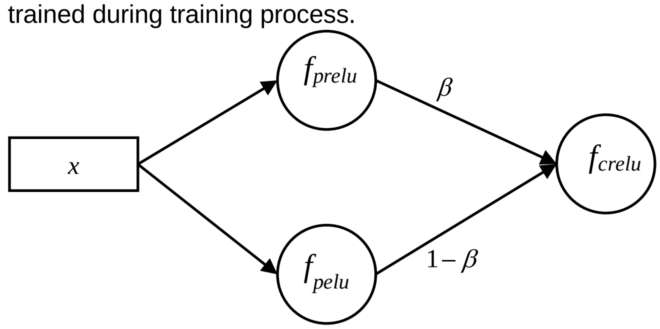

***Fig.** 2. Joined $f_{prelu}(\cdot)$ and $f_{pelu}(\cdot)$ activation activations.*

If we considering the negative sides, linear-type activation functions are not saturated when the input value tends to small values. However, exponential type activation functions are considered as saturated to a negative value when the input value tends very small values. It is considered that, activation function changes signals with different forms in the forward propagation and backward propagation. Hyper-parameters α and β can be viewed as representation a degree of signal change to certain extension.

3.3. Joined activation function

Joined activation function is depicted in Fig. 3, and its analytic form for this approach is given below:

$$f_{crelu} = \beta f_{prelu}(\mathrm{x}) + (1-\beta) f_{pelu}(\mathrm{x}) \tag{6}$$

where $\beta \in [0,1]$ is a joining coefficient indicating the certain join of $f_{prelu}(\cdot)$ and $f_{pelu}(\cdot)$. The joining coefficient β is obtained from during training process.
If we are given output loss function E, we can train the joining coefficient β. The back-propagation error can be calculated

as follows

$$\frac{\partial E}{\partial \beta} = \delta\left(f_{prelu}(\mathrm{x}) - f_{pelu}(\mathrm{x})\right) \tag{7}$$

where $\delta = \dfrac{\partial E}{\partial f_{crelu}}$ is the back-propagated error from the following layer. The error also needs to be propagated back to the previous layer, which can be computed as follows:

$$\frac{\partial E}{\partial x} = \delta\left(\beta \frac{f_{prelu}(\mathrm{x})}{\partial x} + (1-\beta)\frac{f_{pelu}(\mathrm{x})}{\partial x}\right) \tag{8}$$

3.4. Adaptive activation function

However, once each joining coefficient is trained in combined activation approach, then the joined activation is used as constant value. This means that, it is not adaptable to the certain input. Now we consider an adaptive approach which is depicted in Fig. 3. In particular, instead of training a joining coefficient, we instead train a gating pattern. The result of the gating pattern and input signals is presented to a SoftPlus function to form the joining coefficient with $f_{lrelu}(\cdot)$ and $f_{elu}(\cdot)$. Consequently, the actual joining coefficient rely on both the gating pattern and the input signals. Let us consider that the input signal and the learnable parameters are denoted by x and λ, respectively. Then the joining coefficient can be determined as follows:

$$f_{\mathrm{softplus}}(x) = \ln(1+\mathrm{e}^{\lambda x})$$

where $f_{softplus}(\cdot)$ is a SoftPlus function [15], $\lambda \in (0,1]$ is constant value in the given interval.

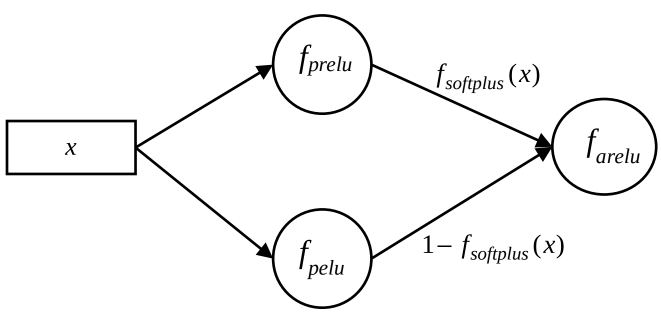

***Fig. 3**. Adaptive with basic activation functions.*

In the adaptive approach, the combination coefficient is replaced with SoftPlus function, and combined with a basic $f_{prelu}(\cdot)$ and $f_{pelu}(\cdot)$. According to adaptive approach aforementioned earlier, the joining of basic $f_{prelu}(\cdot)$ and $f_{pelu}(\cdot)$ is considered adaptable to the certain input signals. Therefore, we can define an adaptive activation function as follows:

$$f_{arelu}(x) = f_{softplus}(x) f_{prelu}(x) + (1 - f_{softplus}(x)) f_{pelu}(x) \tag{9}$$

To calculate gradients with respect to λ and x are as follows

$$\frac{\partial E}{\partial \lambda} = \delta \frac{x}{1+e^{\lambda x}}\left(f_{prelu}(x) - f_{pelu}(x)\right) \tag{10}$$

$$\frac{\partial E}{\partial x} = \delta\left(\frac{\lambda}{1+e^{\lambda x}}\left(f_{prelu}(x) - f_{pelu}(x)\right) + f_{softplus}(x)\left(\frac{\partial f_{prelu}(x)}{\partial x} - \frac{\partial f_{pelu}(x)}{\partial x}\right) + \frac{\partial f_{pelu}(x)}{\partial x}\right) \tag{11}$$

$$\frac{\partial E}{\partial \alpha} = \delta\, f_{softplus}(x) \frac{\partial f_{prelu}}{\partial \alpha}, \quad \frac{\partial L}{\partial \beta} = \delta(1 - f_{softplus}(x)) \frac{\partial f_{pelu}}{\partial \beta} \tag{12}$$

The combined approach and the adaptive approach can adopt both linear and nonlinear transformations. If we compare the two approaches, the significance is that the joined approach is not adaptive to the specific input signals. Therefore, using flexible joining parameter $f_{\mathrm{softplus}}(\cdot)$ is more preferable. Particularly, in adaptive approach trains λ, which determines the degree of signal change for the input value. After training λ, $f_{\mathrm{softplus}}(\cdot)$ defines the certain proportion of signal variability for the certain x. Additionally, when x changes to a small value in $f_{\mathrm{softplus}}(\cdot)$, the degree of signal change tends to be defined only by one of the activation function. Overall, the adaptive approach is desired to be adaptive to signal change in online training.

In the joined and adaptive approaches, we only use basic $f_{prelu}(\cdot)$ and $f_{pelu}(\cdot)$ activation functions in order to test the productiveness. In general, the adaptive approach can increase the ability of training neural models which its datasets have complex structure.

4. EXPERIMENTS

We provide numerical experiments of the proposed activation approaches with several deep CNN models on MNIST, CIFAR10, CIFAR100 datasets. Firstly, we do comparison between common activation function (ReLU), combined activation, and then adaptive activation. First, we perform numerical experiments on the CIFAR dataset. Then, we mainly focus on evaluating the effects of adaptive activation function, and perform the validation experiments with deep CNNs on all datasets. In numerical experiments we use our proposed activation functions replacing $f_{relu}(\cdot)$ activation function in CNN architecture, which is given in Table 1. After creating models with the configuration in Table 1, we train all models with the training dataset. We use Caffe model as the deep learning framework to perform experiments, and employ NVIDIA GTX GPU to test the model training.

***TABLE 2**. CLASSIFICATION RESULTS OF COMPARISON BETWEEN ACTIVATION FUNCTIONS. THERE ARE GIVEN RUNS OF FIVE SEPARATE TESTS MEAN VALUE OF CLASSIFICATION RESULT.*

Configuration of CNN model	Classification Rates		
	MNIST dataset	CIFAR10 dataset	CIFAR100 dataset
$f_{relu}(\cdot)$	0,9568	0,9645	0,9580
$f_{prelu}(\cdot)$	0,9588	0,9523	0,9468
$f_{elu}(\cdot)$	0,9750	0,9636	0,9467
$f_{crelu}(\cdot)$ with $f_{prelu}(\cdot) / f_{pelu}(\cdot)$	0,9902	0,9872	0,9884
$f_{arelu}(\cdot)$ with $f_{prelu}(\cdot) / f_{pelu}(\cdot)$	0,9915	0,9912	0,9910

3.1. Comparison results of combined and adaptive activation functions

First, we evaluate a comparison results among combined activation and adaptive activation. We use basic activation functions, such as $f_{relu}(\cdot)$, $f_{lrelu}(\cdot)$ and $f_{pelu}(\cdot)$, which is used mostly deep learning models. The numerical results in Table 2, our proposed activation functions show superiority over basic activation functions. The combined activation function and the adaptive activation functions shows better performance results than basic activation functions both on MNIST and CIFAR datasets. Comparison of test on MNIST dataset, the adaptive activation function performs the best and achieves an average improvement compared with other activation functions. We also determine which of the activation approach to use, compared with training combination coefficients on the network layers, the performance of learning joining coefficients per network layers boosts on accuracy ($f_{crelu}(\cdot)$ *with* $f_{prelu}(\cdot)/f_{pelu}(\cdot) > f_{crelu}(\cdot)$ *with* $f_{lrelu}(\cdot)/f_{elu}(\cdot)$ and $f_{arelu}(\cdot)$ *with* $f_{prelu}(\cdot)/f_{pelu}(\cdot) > f_{arelu}(\cdot)$ *with* $f_{lrelu}(\cdot)/f_{elu}(\cdot)$) with enhancing the quality of learned hyper-parameters. Additionally, the performance achieved by the adaptive approach is almost consistently better than that achieved by the non-adaptive approach with learning combination coefficients per network layer on the same dataset. Overall, the trend of performance achieved by learning activation functions is almost "$f_{arelu}(\cdot)$ *with* $f_{prelu}(\cdot)/f_{pelu}(\cdot) > f_{arelu}(\cdot)$ *with* $f_{lrelu}(\cdot)/f_{elu}(\cdot) > f_{crelu}(\cdot)$ *with* $f_{prelu}(\cdot)/f_{pelu}(\cdot) > f_{crelu}(\cdot)$ *with* $f_{lrelu}(\cdot)/f_{elu}(\cdot)$". Then we perform experiments to analyze on the classification performance, and compare adaptive activation with combined activation function.

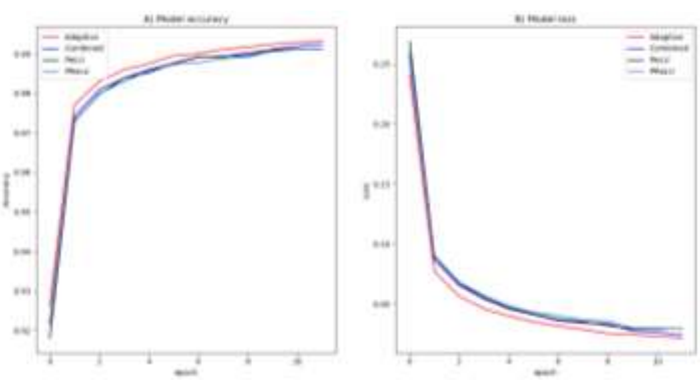

***Fig. 4**. The training process of deep CNN model with various activation functions on MNIST dataset.*

Compared with the combined approach, the adaptive approach structure is much more complex to train model. To verify the effectiveness of the adaptive structure, we perform experiments that use $f_{prelu}(\cdot)/f_{pelu}(\cdot)$ for the combined approach and adaptive approach. From Table 3, we can see that the combined approach and the adaptive approach with $f_{prelu}(\cdot)/f_{pelu}(\cdot)$ do not show obvious performance difference, when compared with these two approaches with $f_{lrelu}(\cdot)/f_{elu}(\cdot)$. The adaptive approach with $f_{prelu}(\cdot)/f_{pelu}(\cdot)$ can consistently perform better, and achieve moderate improvements.

***TABLE 3**. CLASSIFICATION ERROR WITH COMPARISON AMONG THE MIXED STRATEGY, THE GATED STRATEGY AND HIERARCHICAL STRATEGY. WE RUN FIVE SEPARATE TRIALS AND REPORT MEANS CLASSIFICATION RATE.*

Configuration of CNN model	Classification Rates		
	MNIST dataset	CIFAR10 dataset	CIFAR100 dataset
$f_{crelu}(\cdot)$ *with* $f_{lrelu}(\cdot)/f_{elu}(\cdot)$	0,9902	0,9872	0,9884
$f_{crelu}(\cdot)$ *with* $f_{prelu}(\cdot)/f_{pelu}(\cdot)$	0,9908	0,9973	0,9887
$f_{arelu}(\cdot)$ *with* $f_{lrelu}(\cdot)/f_{elu}(\cdot)$	0,9915	0,9912	0,9910
$f_{arelu}(\cdot)$ *with* $f_{prelu}(\cdot)/f_{pelu}(\cdot)$	0,9922	0,9919	0,9912

In order to further verify whether the adaptive structure makes learning harder, we compare the adaptive approach with the combined approach and the adaptive approach, and all of these approaches use $f_{prelu}(\cdot)/f_{pelu}(\cdot)$. We visualize the training losses on MNIST and CIFAR10 in Fig. 5. Compared with the combined approach, the training loss of the adaptive approach does not show obvious difference.

***TABLE 4**. PARAMETERS OF MNIST AND CIFAR10 MODEL.*

Convolutions Layer #	MNIST			CIFAR10		
	α	β	λ	α	β	λ
Conv.Layer-1	0.18	0.22	0.41	0.01	0.87	-0.52
Conv.Layer-2	0.05	0.23	-0.30	0.31	0.45	-1.21
Conv.Layer-3	0.27	1.02	-0.08	0.002	-0.12	0.54

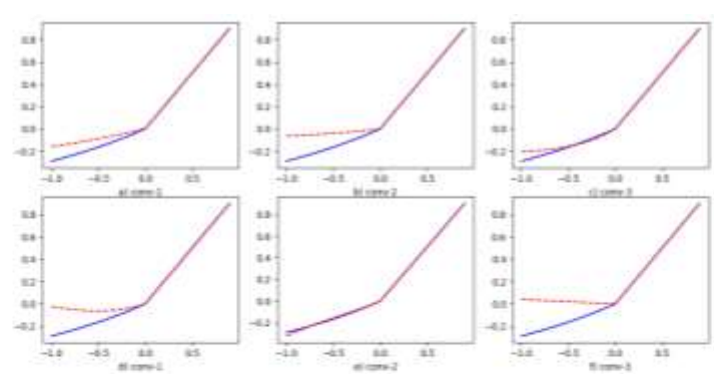

***Fig. 5**. Visualization of trained activation on MNIST (figures: a, b, and c) and CIFAR10 (figures: d, e, and f) datasets. Red dashed lines indicate activation functions after training process; blue lines indicate initial state of activation functions; The positive parts are shared as same by activation functions.*

5. CONCLUSION

In the current note, we explored the activation functions and its various forms. To accelerate training process and to achieve high accuracy, we proposed two approaches. The first approach focuses on training a combined activation function, and the second approach aims to training an adaptive activation function. These approaches improve the ability of activation functions in training non-linear transformations and are adaptive to the network input. Namely, to show effectiveness, we investigate linear form and non-linear form to combine activation functions. Numerical experiments show that the proposed activation functions overcome by performance than rectified unit family functions.

REFERENCES

[1] Guo Y., Liu Y., Oerlemans A., Lao S., Wu S., Lew M.S. Deep learning for visual understanding: a review, Neurocomputing, Vol. 187, 27–48 (2016).

[2] Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks, Advances In Neural Information Processing Systems, Vol. 25, 1106–1114 (2012).

[3] Li X., Cai C., Zhang R., Ju L., He J. Deep cascaded convolutional models for cattle pose estimation, Computers and Electronics in Agriculture, Vol. 164, 45-67 (2019).

[4] Gu G., Liu J., Li Z., Huo W., Zhao Y. Joint learning based deep supervised hashing for large-scale image retrieval, Neurocomputing, Vol. 385, 348-357 (2020).

[5] Yang J., Zhang D., Frangi A., Yang J. Two-dimensional PCA a new approach to appearance-based face representation and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, Issue 1, 131-137 (2004).

[6] Taigman Y., Yang M., Ranzato M., Wolf L. Deepface: Closing the gap to human-level performance in face verification, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, U.S.A, 1701-1708 (2014).

[7] Li C., Chen Z., Wu Q. M., Liu C. Deep saliency detection via channel-wise hierarchical feature responses, Neurocomputing, Vol. 322, 80-92 (2018).

[8] Tuo Q., Zhao H., Hu Q. Hierarchical feature selection with subtree based graph regularization, Knowledge-Based Systems, Vol. 163, 996-1008 (2019).

[9] Wu G., Lu W., Gao G., Zhao C., Liu J. Regional deep learning model for visual tracking, Neurocomputing, Vol. 175, 310–323 (2016).

[10] An S., Boussaid F., Bennamoun M., Sohel F. Exploiting layerwise convexity of rectifier networks with sign constrained weights, Neural Networks, Vol. 105, 419-430 (2018).

[11] Apicella A., Isgro F., Prevete R. A simple and efficient architecture for trainable activation functions, Neurocomputing, Vol. 370, 1-15 (2019).

[12] He K., Zhang X., Ren S., Sun J. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, The IEEE International Conference on Computer Vision, 1026-1034 (2015).

[13] Li Y., Fan C., Li Y., Wu Q., Ming Y. Improving deep neural network with Multiple Parametric Exponential Linear Units, Neurocomputing, Vol. 301, 11-24 (2018).

[14] Glorot X., Bordes A., Bengio Y. Deep sparse rectifier neural networks. Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA. Vol. 15, 315-323 (2011).

[15] Marakhimov A. R., Khudaybergenov K. K. A fuzzy MLP approach for identification of nonlinear systems, Contemporary problems in mathematics and physics, CMFD, Vol. 65, no. 1, Peoples' Friendship University of Russia, M., 44–53 (2019).

[16] Marakhimov A.R., Khudaybergenov K.K. Convergence analysis of feedforward neural networks with backpropagation, Bulletin of National University of Uzbekistan: Mathematics and Natural Sciences: Vol. 2, Issue 2, 77-93 (2019), Available at: https://uzjournals.edu.uz/mns_nuu/vol2/iss2/1

[17] Yusupbekov N. R., Marakhimov A. R., Igamberdiev H. Z., Umarov Sh. X. An Adaptive Fuzzy-Logic Traffic Control System in Conditions of Saturated Transport Stream, The Scientific World Journal Vol. 2016, 23-36 (2016).

[18] Yusupbekov N.R., Marakhimov A.R., Igamberdiev H.Z., Umarov Sh.X. Application of soft-computing technologies to the traffic control system design problems. 12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016, 29-30 August, Vienna, Austria (2016).

[19] Marakhimov A.R., Siddikov I.H., Nasridinov A., Byun J.Y. Structural Synthesis of Information Computer Networks of Automated Control Systems Based on Genetic Algorithms, Computer Science and its Applications, Vol. 330, 1055-1063 (2015).

[20] Nasridinov A., Marakhimov A. Park Y.H. A design of wireless sensor networks based on fuzzy modeling for comfortable human life, Asia Live Sciences, The Asian International Journal of Life Sciences, July 2015, Philippines, 265-277.

[21] Yusupbekov N.R., Marakhimov A.R. Synthesis of the intelligent traffic control systems in conditions saturated transport stream, International Journal of International Journal of Chemical Technology, Control and Management Jointly with The Journal of Korea Multimedia Society. Special Issue, South Korea, Seoul, 12-18 (2015).

[22] Jagtap D.A., Kawaguchi K., Karniadakis G.E. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, Journal of Computational Physics, Vol. 404, 45-67 (2020).

[23] Konstantinidis D., Argyriou V., Stathaki T., Grammalidis N. A modular CNN-based building detector for remote sensing images, Computer Networks, Vol. 168, 93-121 (2020).

[24] Jiang W., Wu L., Liu S., Liu M. CNN-based two-stage cell segmentation improves plant cell tracking, Pattern Recognition Letters, Vol. 128, 311-317 (2019).

[25] Xu Z., Zhao J., Yu Y., Zeng H. Improved 1D-CNNs for behavior recognition using wearable sensor network, Computer Communications, Vol. 15, Issue 11, 165-171 (2020).

[26] Amin S. U., Alsulaiman M., Muhammad G., Mekhtiche M. A., Hossain M. S. Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion, Future Generation Computer Systems, Vol. 101, 542-554 (2019).

[27] LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition, IEEE 86, Vol. 11, 2278–2324 (1998).

[28] Krizhevsky A., Learning multiple layers of features from tiny images. Technical report, University of Toronto, (2009).

[29] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: a large-scale hierarchical image database, Conference: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami, Florida, USA, 20-25 June 2009.

[30] MNST Dataset, available at: http://yann.lecun.com/exdb/mnist/

[31] CIFAR Dataset, available at: https://www.cs.toronto.edu/~kriz/cifar.html

[32] ImageNet Dataset, available at: http://image-net.org/download

[33] https://en.wikipedia.org/wiki/Activation_function