Interval Type-2 Fuzzy Systems as Deep Neural Network Activation Functions

Aykut Beke and Tufan Kumbasar
Control and Automation Engineering Department
Istanbul Technical University
Istanbul, Turkey
{beke, kumbasart}@itu.edu.tr

Abstract

In this paper, we propose a novel activation function, namely, Interval Type-2 (IT2) Fuzzy Rectifying Unit (FRU), to improve the performance of the Deep Neural Networks (DNNs). The IT2-FRU can generate linear or sophisticated activation functions by simply tuning the size of the footprint of uncertainty of the IT2 Fuzzy Sets. The novel IT2-FRU also alleviates vanishing gradient problem and has a fast convergence rate since it pushes the mean activation to zero by allowing the negative outputs. In order to test the performance of the IT2-FRU, comparative experimental studies are performed on the CIFAR-10 dataset. IT2-FRU is compared with widely used conventional activation functions. Experimental results show that IT2-FRU significantly speeds up the learning and has a superior performance compared to other handled activation functions.

Keywords: Interval type-2 fuzzy system, Footprint of Uncertainty, Activation unit, Deep learning.

1 Introduction

During the past decade, deep learning with neural networks has made a breakthrough in various areas such as image classification, object detection, character recognition, etc. [1-3]. One of the most commonly employed Deep Neural Network (DNN) is the Convolutional Neural Networks (CNNs) [4-6]. Activation functions for DNNs is still an active research area as they have an important role during the learning process. After the discovery of Rectified Linear Unit (ReLU) [7], DNNs have made a great progress as the ReLUs have not only alleviated the vanishing gradient problem but also improve the training performance of DNNs [7], [8].

Activation functions for DNNs in literature can be grouped as the extensions of ReLU and their variants. For instance, Leaky ReLU (LReLU) allows the negative part by assigning a predefined slope value [9] while Parametric ReLU (PReLU) also does not saturate the negative part to zero by learning the slope parameter during training [10]. Moreover, Exponential Linear unit (ELU) and Parametric ELU (PELU), that are defined as identity for positive arguments and $a(e^x - 1)$ for negative ones, proposed to reduce bias shift problem and make learning faster [11], [12]. Even though the activation functions mentioned here have a good performance in DNNs, they all have a disadvantage that is their weak ability to express nonlinear input-output mappings. To define nonlinear input-output mappings, it has been shown that Fuzzy Logic Systems (FLSs), especially Interval T2 (IT2) FLSs, are very powerful tools [13–21]. This lies due to the fact that IT2-FLSs use and employ IT2 Fuzzy Sets (FSs) that have more extra degree of freedom provided by the Footprint of Uncertainty (FOU) [13-21]. Hence, many theoretical studies have been presented on how the size/shape of the FOU affects the IT2 Fuzzy Mappings (FMs) [20-23].

In this paper, we will propose a novel IT2 Fuzzy Rectifying Unit (FRU) that inherits the powerful properties of IT2-FMs to improve the learning performances of the DNNs. The novel activation function IT2-FRU is learned for each neuron independently during training and can express a linear or sophisticated activation function by simply tuning the FOU sizes of IT2-FSs. Thus, IT2-FRU gives the opportunity to the DNNs to have better learning capability. Moreover, the novel IT2-FRU also allows negative outputs and has the similar advantages with the other non-saturated activation functions; it alleviates vanishing gradient problem and has a fast convergence rate. In order to investigate the performance of the proposed IT2-FRU, we will present two different CNN structures; the first one is selected as a shallow network and the second one is a DNN structure composed of three stacked CNN layers. In both cases, we will compare the performance of IT2-FRU with ReLU, PReLU and ELU on the benchmark CIFAR-10 dataset. We will also test the performances of the rectifying units in presence of Batch Normalization (BN) layer which is widely done in benchmarking of DNN structures. The experimental results will show that the proposed IT2-FRU can have a significant performance improvement when compared to other activation units and have a very robust performance with or without BN layer(s).

This paper is organized as follows. Section 2 provides some preliminaries on activation functions. Section 3 presents the proposed novel IT2-FRU. The comparative experimental results of the proposed IT2-FRU are given in section 4. Finally, the conclusions and future works are given in section 5.

2 Preliminaries on Activation Functions

In this section, we will give some information about the most popular activation functions used in DNNs which are the ReLU, PReLU and ELU.

- The activation function of ReLU is defined as [7]

$$f(x) = \max(0, x) \tag{1}$$

where, x is the input.

- The activation functions of LReLU [9] and PReLU [10] are defined as

$$f(x) = \max(0, x) + a \min(0, x) \tag{2}$$

where $a > 0$. The only difference between the PReLU and the LReLU is that the leak parameter a is learned during training in order to find a proper positive slope for negative inputs in PReLU. This allows negative outputs and prevents the negative inputs saturated always to zero.

- Another proposed ReLU based activation unit is the ELU [11] that is defined with:

$$f(x) = \max(0, x) + a(\mathrm{e}^{\min(0,x)} - 1) \tag{3}$$

The ELU also allows the negative outputs and the hyperparameter a controls the values of the output in the negative quadrant and weights them exponentially.

3 Interval Type-2 Fuzzy Rectifying Unit

In this section, we will present a novel activation function based on IT2-FLSs. We will firstly present the properties SIT2-FLS which provides the basis of the IT2-FRU. Then, the proposed novel fuzzy activation unit will be introduced.

3.1 Internal Structure of the SIT2-FLS

In this study, we use and employ a SIT2-FLS composed of 3 rules which is defined as:

$$R_n\text{: If } \sigma \text{ is } \tilde{A}_n \text{ Then } \varphi_o \text{ is } B_n, n = 1,2,3 \tag{4}$$

Here, φ_o is the output, σ is the normalized input value that is defined as:

$$\sigma = K_s x \tag{5}$$

where x is the actual input and K_s is scaling factor that helps to normalize the inputs in between $\sigma \in [-1, 1]$. The consequent parts (B_n) in (4) are defined with crisp MFs as $B_1 = N$, $B_2 = Z$ and $B_3 = P$, while the antecedent part is defined with the IT2-FSs $(\tilde{A}_n)$ shown in Fig. 1. Here, the heights of the LMFs m_n is the only parameter that creates the FOU of the IT2-FS [20].

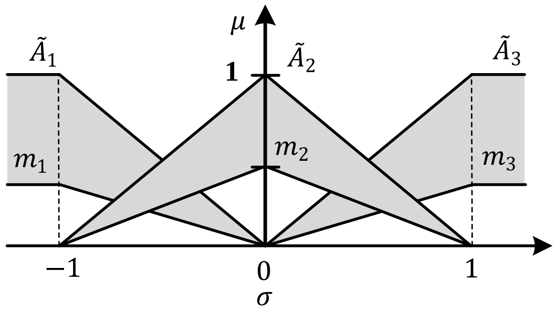

Figure 1: Illustration of the antecedent IT2-FSs.

In the rest of the paper, we will employ the following equalities; $Z = 0$ to guarantee that $\sigma = 0 \Rightarrow \varphi_o = 0$. Additionally, the height of the LMFs is employed as $m_2 = \alpha, m_1 = m_3 = 1 - \alpha$ as suggested in [20]. The resulting IT2-FM $(\varphi_o(\sigma))$ for $\sigma \in [0, 1]$ can be formulated as

$$\varphi_o(\sigma) = P\sigma k(\sigma) \tag{6}$$

where $k(\sigma)$ is defined as:

$$k(\sigma) = \frac{1}{2}\left(\frac{1}{\alpha + \sigma - \alpha\sigma} + \frac{-1 + \alpha}{-1 + \alpha\sigma}\right) \tag{7}$$

Similarly, for the input interval $\sigma \in [-1, 0]$ the IT2-FM can be derived as [20]:

$$\varphi_o(\sigma) = N\sigma k(-\sigma) \tag{8}$$

The following design guidelines have been presented in [20] by defining the criteria $\varepsilon_0(\sigma) = |\varphi_o(\sigma)| - |\sigma|$:

(i) Case-1: If $0 < \alpha \leq \alpha_A$, then $\varepsilon_0(\sigma) > 0$ is always satisfied for $\forall\, \sigma \in [-1, 1]$. Thus, an Aggressive FM (A-FM) is obtained compared to a Unit Mapping (UM).

(ii) Case-2: If $\alpha_S \leq \alpha < 1$, then $\varepsilon_0(\sigma) < 0$ is always satisfied for $\forall\, \sigma \in [-1, 1]$. Therefore, a Smooth FM (S-FM) is obtained in comparison with a UM.

(iii) Case-3: If $\alpha_A < \alpha < \alpha_S$, then $\varepsilon_0(\sigma) \approx 0$ for $\forall\, \sigma \in [-1, 1]$. Therefore, nearly a U-FM is obtained.

where $\alpha_A = (3 - \sqrt{5})/2$ and $\alpha_S = (-1 + \sqrt{5})/2$.

In Fig. 2, the A-FM ($\alpha = 0.1$), S-FM ($\alpha = 0.9$) and U-FM ($\alpha = 0.5$) are sketched for all of the consequent parameters are selected as $P = N = 1$. Also, a UM is presented for the comparison. Here, it can be seen that the S-FM has relatively low input sensitivity when σ is close to "0" in comparison to the A-FM. Moreover, as seen in Fig. 2, nearly a UM can be obtained when $\alpha_A < \alpha < \alpha_S$ is satisfied. Thus, the α can be seen as learning hyperparameter to be set or learned to increase the performance of DNNs.

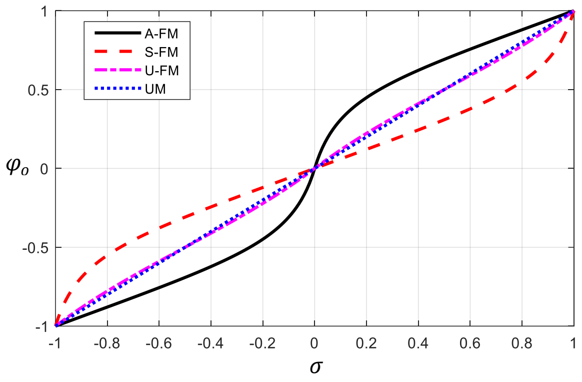

Figure 2: Illustration of the input-output FMs.

3.2 Learning with IT2-FRU

Here, we will introduce the novel IT2-FRU. The proposed activation unit can be formulated by arranging the (6) and (8) as following:

$$f(\sigma) = \begin{cases} P\sigma k(\sigma), & \text{if } \sigma > 0 \\ N\sigma k(-\sigma), & \text{if } \sigma \le 0 \end{cases} \tag{9}$$

The resulting output of the IT2-FRU could be a linear or nonlinear activation depending on selection of the parameters. IT2-FRU has three learnable parameters P, N and α. As shown in Fig. 3, the parameter P controls the slope of the function in positive quadrant, while the parameter N controls the slope of the function in negative quadrant. In the case of $\alpha_A < \alpha < \alpha_S$, the outputs converge to linear functions with slopes of P or N as seen in Fig. 3. Moreover, when $0 < \alpha \le \alpha_A$ or $\alpha_S \le \alpha < 1$ is satisfied, the activation function $f(\sigma)$ can imitate a sophisticated input-output mappings such as A-FM or S-FM. Thus, the total number of additional parameters that could be learned when using the IT2-FRUs in a layer is only $3C$, where C is the overall number of hidden units in the activation layer. This is a small number in comparison with the total number of weights in typical networks. In contrast to the ReLU, IT2-FRU can process negative input values that helps to push the mean activation of the layer around zero. It is known that layers with mean activations closer to zero tend to learn faster [11]. Unlike the other rectifying linear units (ReLU, ELU, LReLU, etc.), IT2-FRU can also learn each neurons contribution in the positive quadrant. Besides, proposed IT2-FRU can learn sophisticated input-output mappings for the negative quadrant by tuning the FOU parameter of the IT2-FS unlike its counterparts. This flexibility gives the network the opportunity to improve its learning performance. Moreover, IT2-FRU can alleviate the effect of small variations for certain hyperparameter settings if necessary. For instance, if the S-FM is employed, then the IT2-FRU has a relatively low input sensitive for $\sigma \approx 0$. Low input sensitivity indicates that variating information propagated to the next layer is decreased, thus the resulting network will be more robust against noisy inputs.

The update rule of the learnable parameters $o \in \{P, N, \alpha\}$ is derived by the chain rule and defined as follows:

$$\frac{\partial L}{\partial o_c} = \sum_j \frac{\partial L}{\partial f_c(\sigma_{cj})} \frac{\partial f_c(\sigma_{cj})}{\partial o_c} \tag{10}$$

where $c = 1, \ldots, C$ indexes of the channels, L represents the objective function of the DNN, j indexes of the elements over height, width, and observations. The term $\partial L / \partial f_c(\sigma_c)$ is the gradient propagated from the deeper layers after the IT2-FRU. The gradients of the IT2-FRU for each parameter are given in (11) and

$$\frac{\partial f_c(\sigma_c)}{\partial P_c} = \begin{cases} \sigma_c k(\sigma_c), & \text{if } \sigma_c > 0 \\ 0, & \text{if } \sigma_c \le 0 \end{cases} \tag{12}$$

$$\frac{\partial f_c(\sigma_c)}{\partial N_c} = \begin{cases} 0, & \text{if } \sigma_c > 0 \\ \sigma_c k(-\sigma_c), & \text{if } \sigma_c \le 0 \end{cases} \tag{13}$$

The rule for updating the parameters o_c by momentum method is:

$$\Delta o = \mu \Delta o + \varepsilon \frac{\partial L}{\partial o} \tag{14}$$

Here μ represents the momentum terms and ε is the learning rate.

In the training of IT2-FRU, the slope parameters P and N are not restricted with an upper or lower bound and thus can be learned freely during the training. However, we constrain the parameter of α in the interval of $[0, 1]$ to end up with an IT2-FS.

4 Experiments with IT2-FRU

In this section, we will examine the learning performance of DNN structures with the IT2-FRUs. To evaluate the performance of novel activation function, we performed several experiments with CIFAR-10 data set [1] for different CNN structures. The CIFAR-10 dataset contains color images with size of 32×32. It contains 50000 training images and 10000 testing images with 10 different classes [1]. Besides, we did not perform any data augmentation. We performed the classification experiments in MATLAB and CUDA environments. All experiments are conducted on a PC that includes Intel Core i7 3.3GHz CPU, 32GB RAM. Moreover, we used a NVIDIA GTX 1080 TI GPU to reduce the training time.

$$\frac{\partial f_c(\sigma_c)}{\partial \alpha_c} = \begin{cases} \dfrac{P_c\sigma_c}{2}\left(\dfrac{1}{\alpha_c\sigma_c - 1} + \dfrac{\sigma_c - 1}{(\alpha_c + \sigma_c - \alpha_c\sigma_c)^2} + \dfrac{\sigma_c - \sigma_c\alpha_c}{(\alpha_c\sigma_c - 1)^2}\right), & \text{if } \sigma_c > 0 \\ -\dfrac{N_c\sigma_c}{2}\left(\dfrac{1}{\alpha_c\sigma_c + 1} + \dfrac{\sigma_c + 1}{(\alpha_c - \sigma_c + \alpha_c\sigma_c)^2} + \dfrac{\sigma_c - \sigma_c\alpha_c}{(\alpha_c\sigma_c + 1)^2}\right), & \text{if } \sigma_c \le 0 \end{cases} \tag{11}$$

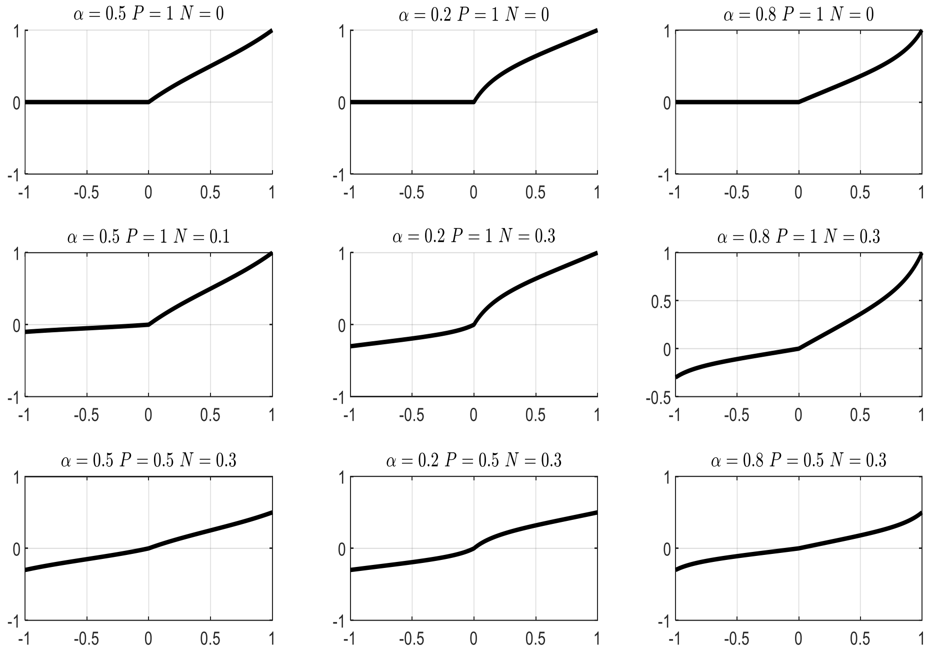

Figure 3: Resulting activation functions for different values of P, N and α.

The experiments are performed for two different DNN structures. In the first structure, we selected a shallow network with single CNN layer and for the second network we composed a deep layer by stacking three CNN layers. Since we mainly focus on testing the effects of IT2-FRU on the performance of deep networks, we only change the activation units in the employed network structures and keep other elements of the network unchanged. We compared our proposed IT2-FRU to ReLU, PReLU and ELU activation functions on two networks with and without BN layer. Also, cross-entropy is selected for loss function as we handle a classification problem.

$$L(y_n, \hat{y}_n) = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log \hat{y}_n + (1 - y_n)\log(1 - \hat{y}_n)] \qquad (15)$$

Here, N is the number of classes.

In learning process, we applied the stochastic gradient descent algorithm with a minibatch size of 64 samples and momentum of 0.9 for 5 epochs. The L2-weight decay regularization term was set to 0.004. To make a fair comparison, we used the same learning settings for all employed activation units in the handled networks.

4.1 Classification Performance with the Shallow CNN Network

In this subsection, we test the learning behavior and performance of the proposed IT2-FRU in a simple CNN structure rather than a complex neural network to show the superiority of the IT2-FRU. The shallow network structure has an image input layer with a ‘zero-center’ normalization. The convolution layer of the network has 32 filters. Each kernel size has been selected as 5×5 and has been padded by 2 pixels on each side. After convolution layer, in one structure a BN layer is added and it is named as Shallow CNN-BN, while the other structure is constructed without BN layer that is named as Shallow CNN. Then, the convolutional layer or BN layer is followed by an activation unit with the selections of ReLU, PReLU, ELU or IT2-FRU. Finally, one fully connected layer with size of 10 and a softmax classification layer have been added to the network.

In learning process, the learning rate starts at 10^{-2} for Shallow CNN-BN and 10^{-4} for Shallow CNN, after 3 epochs its value divided by 10 for both network. These initial learning rates are experimentally determined as the optimal initial learning rates for these networks.

In Fig. 4, the medians of training loss over five experiments on CIFAR-10 dataset are presented for both structure Shallow CNN and Shallow CNN-BN. Results show that proposed IT2-FRU has achieved a lower training loss value compared to ReLU, PReLU and ELU with both structure; with or without BN. Moreover, as it can be seen from Fig. 4a, IT2-FRU has a much better and faster convergence behavior with Shallow CNN structure compared to its counterparts. In Fig. 4b, we observed that BN is generally improved training performance. Also, we observe that IT2-FRU has the best training performance in Shallow CNN-BN as illustrated in Fig. 4b.

As shown in Table 1, in the testing experiments, IT2-FRU obtained a higher median accuracy rate on CIFAR-10 with or without BN layer when compared to ReLU, PReLU and ELU. The deployment of the IT2-FRU layer has increased the median accuracy rate by 2.04% compared to ReLU counterpart and achieved the maximum median accuracy rate of 50.90% on CIFAR-10 without BN network. On the other hand, we see that BN is generally increased the median testing accuracy. Besides, proposed IT2-FRU has improved the median testing accuracy of the second best rectifying unit, ELU, in amount of 3.94% as given in Table 1.

It can be concluded that both the IT2-FRU's training and testing performances are significantly better when compared to the other activation units for a shallow CNN structure. BN is generally improved the testing performances of rectifying units. It also can be seen that IT2-FRU network clearly outperforms its counterparts in Shallow CNN-BN.

4.2 Classification Performance with Stacked CNN Network

In this subsection, we examine the learning behavior and generalization capabilities of the proposed IT2-FRU against the other activation functions with a relatively more sophisticated CNN structure. Our relatively complex network structure has an image input layer with a 'zero-center' normalization. We designed our complex network by stacking 3 convolutional layers one after another. These 3 convolutional layers have been constructed with 32, 32 and 64 filters, respectively. Each one of them has a kernel size of 5×5. Each CNN layer is padded with 2 pixels on each side.

After each convolutional layer, in one structure a BN layer is added and it is named as Stacked CNN-BN, while the other structure is constructed without BN layer and named as Stacked CNN. Then, each convolutional layer or each BN layer is followed by an activation unit with the selections of ReLU, PReLU, ELU or IT2-FRU. After the activation, the layers are followed by the max-pooling layers. Pooling layers have a kernel size of 3×3 with a stride of 2. Then, two fully connected layers with sizes of 64 and 10, respectively, with an activation unit between them have been added to the network. Finally, a softmax classification layer is added.

In the learning process, the learning rate starts with a value of 10^{-2} for both stacked CNNs (Stacked CNN-BN and Stacked CNN). Learning rate is divided by 10 after 3 epochs during training. This learning rate is determined experimentally for these DNNs.

In Fig. 5, the medians of training loss over five experiments on CIFAR-10 dataset are presented for both structure Stacked CNN-BN and Stacked CNN. It should be noted that we do not present the results for PReLU in Fig. 5, since it was not able to learn the data set with the selected learning rate value. The results show that the proposed IT2-FRU has achieved a lower training loss value compared to ReLU, PReLU and ELU with both stacked structure with or without BN layer. The training performance of the IT2-FRU significantly outperforms other activation units, in particular, within Stacked CNN structure as seen in Fig. 5a. The deployment of a BN layer has significantly improved the training performance of ReLU, PReLU and ELU. BN, also enhanced the training performance of IT2-FRU as seen in Fig. 5b.

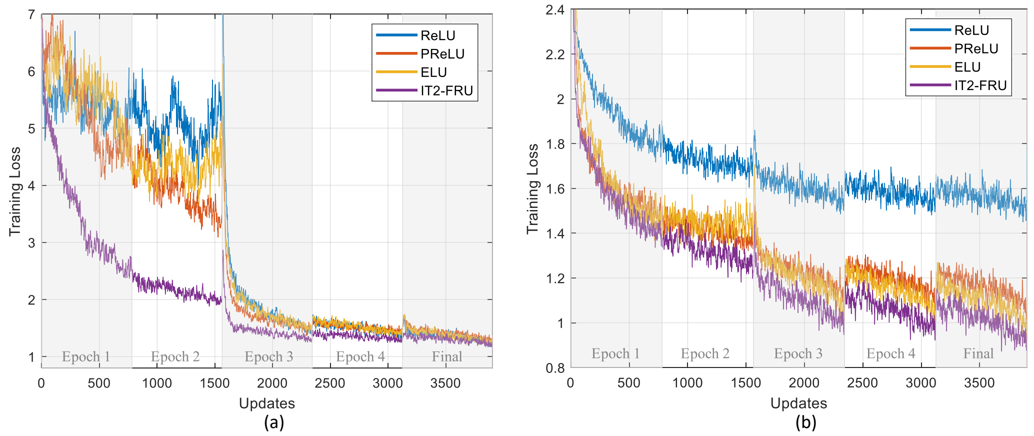

Figure 4: Medians of training loss values over five trials: Performances of the activation units (a) Shallow CNN (b) Shallow CNN-BN.

	Shallow CNN-BN	Shallow CNN	Stacked CNN-BN	Stacked CNN
ReLU	43.84%	48.86%	77.60%	56.24%
PReLU	55.94%	47.65%	77.65%	-
ELU	57.29%	48.70%	77.64%	59.77%
IT2-FRU	**61.23%**	**50.90%**	**77.49%**	**74.35%**

Table 1: Median values of test accuracy on CIFAR-10 over five experiments.

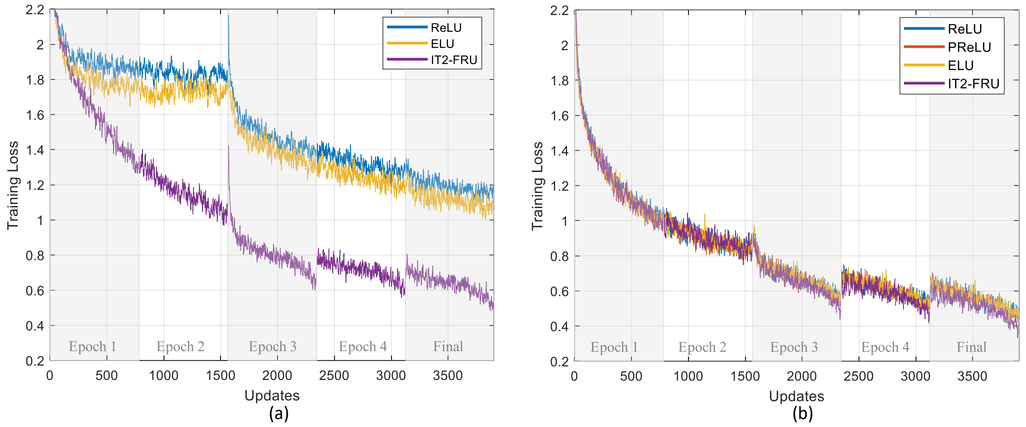

Figure 5: Medians of training loss values over five trials: Performances of the activation units (a) Stacked CNN (b) Stacked CNN-BN.

In Table 1, the median values of test accuracy on CIFAR-10 are presented. We observed that IT2-FRU has a significantly higher median accuracy rate on CIFAR-10 with Stacked CNN. IT2-FRU has increased median accuracy rate by 18.11% compared to ReLU counterpart and achieved the maximum accuracy rate of 74.35% on CIFAR-10 without BN layer as seen in Table 1. With Stacked CNN-BN network, all employed rectifying units have nearly same median accuracy rate values. These results clearly reveal that BN has a crucial influence on the ReLU, PReLU and ELU, while the IT2-FRU can achieve satisfactory performance with or without BN networks.

Remark: It should be underlined that more sophisticated DNN structures have been employed in the literature to reach a satisfactory classification performance on CIFAR-10 dataset. As the aim of this section is to show that the IT2-FRU is a powerful tool and can enhance the performance, thus we preferred relatively simpler DNN structures.

To sum up, the deployment of IT2-FRU layer resulted with a robust and high performance with the Stacked CNN structure when compared to other activation units. It is also worth to underline that the IT2-FRU can achieve high classification performances with or without BN layer networks unlike its counterparts.

5 Conclusions and Future Works

In this paper, we proposed a novel IT2-FRU that is based on IT2-FMs to enhance the learning performance of the DNNs both in accuracy and convergence speed. The advantage of the proposed IT2-FRU is that it has the capability to learn sophisticated activation functions, alleviates vanishing gradient problem and has a fast convergence rate as it does not map negative input values to zero output values. The presented classification results obtained from different DNN structures demonstrated that IT2-FRU has effectively improved the performance of DNNs, especially without BN layer.

In our future work, we will examine the proposed IT2-FRU on different datasets with more complex DNNs.

Acknowledgement

This research is supported by the project (118E807) of Scientific and Technological Research Council of Turkey (TUBITAK). All of these supports are appreciated.

References

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, Imagenet classification with deep convolutional neural networks, In advances in neural information processing systems (2012) 1097–1105.

[2] R. Girshick, J. Donahue, T. Darrell and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[3] D. C. Ciresan, U. Meier, L. M. Gambardella and J. Schmidhuber, Convolutional neural network committees for handwritten character classification, in International Conference on Document Analysis and Recognition, 2011, pp. 1135–1139.

[4] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (2015) 436–444.

[5] A. Krizhevsky and G. Hinton, Learning multiple layers of features from tiny images, Technical report, University of Toronto, Toronto, USA (2009).

[6] S. Ren, K. He, R. Girshick and J. Sun, Faster r-cnn: Towards realtime object detection with

region proposal networks, in Advances in Neural Information Processing Systems, (2015) 91–99.

[7] V. Nair and G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in Proceedings of the 27th International Conference on Machine Learning, 2010, pp.807–814.

[8] G. E. Dahl, T. N. Sainath and G. E. Hinton, Improving deep neural networks for LVCSR using rectified linear units and dropout, IEEE international conference on acoustics, speech and signal processing, 2013, pp. 8609–8613.

[9] A. L. Maas, A. Y. Hannun and A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in Proc. Icml, (2013) 3.

[10] K. He, X. Zhang, S. Ren and J. Sun, 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.

[11] D.-A. Clevert, T. Unterthiner and S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus), 2015, arXiv preprint arXiv: 1511.07289.

[12] L. Trottier, P. Gigu and B. Chaib-draa, Parametric exponential linear unit for deep convolutional neural networks, in 16th IEEE International Conference on Machine Learning and Applications , 2017, pp. 207–214.

[13] J. M. Mendel, Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions, Springer, 2017.

[14] O. Castillo and P. Melin, Type-2 Fuzzy Logic: Theory and Applications, Spinger, 2008.

[15] J. M. Mendel, H. Hagras, W.W Tan, W.W. Melek and H. Ying, Introduction to Type-2 Fuzzy Logic Control, John Wiley and IEEE Press, Hoboken, NJ, 2014.

[16] D. Wu, On the fundamental differences between interval type-2 and type-1 fuzzy logic controllers, IEEE Transaction on Fuzzy Systems, (2012) 832–848.

[17] Y. Li, H. K. Lam, and L. Zhang, Control design for interval type-2 polynomial fuzzy-model-based systems with time-varying delay, IET Control Theory & Applications, (2017) 2270–2278.

[18] H. Li, J. Wang, H. K. Lam, Q. Zhou and H. Du, Adaptive sliding mode control for interval type-2 fuzzy systems, IEEE Transactions on Systems, Man, and Cybernetics: Systems, (2016) 1654–1663.

[19] D. Wu, Y. Yuan and Y. Tan, Optimize TSK Fuzzy Systems for Big Data Regression Problems: Mini-Batch Gradient Descent with Regularization, DropRule and AdaBound (MBGD-RDA), 2019, arXiv preprint arXiv: 1903.10951.

[20] T. Kumbasar, Robust stability analysis and systematic design of single-input interval type-2 fuzzy logic controllers, IEEE Transaction on Fuzzy Systems, (2016) 675–694.

[21] A. Beke and T. Kumbasar, Type-2 Fuzzy Logic-Based Linguistic Pursuing Strategy Design and Its Deployment to a Real-World Pursuit Evasion Game, IEEE Transactions on Cybernetics, (2018).

[22] X. Du and H. Ying, Derivation and analysis of the analytical structures of the interval type-2 fuzzy-PI and PD controllers, IEEE Transaction on Fuzzy Systems, (2010) 802–814.

[23] M. Nie and W. W. Tan, Analytical structure and characteristics of symmetric Karnik–Mendel type-reduced interval type-2 fuzzy PI and PD controllers, IEEE Transaction on Fuzzy Systems, (2012) 416–430.