RELU-ELU: An Enhance Approach For Improving The Performance Of Deep Neural Networks

Rashmi Awasthi[1] , Dr. B.K.Sharma[2]

[1]Scholar Department of Computer Science and Application Mandsaur University, Mandsaur.
[2]Professor Department of Computer Science and Application Mandsaur University, Mandsaur.

During pandemic the need for AI-based diagnostic tools has increased, especially in medical imaging. Activation functions are central to the performance of neural networks, influencing how well models learn and generalize complex data patterns. This study explores the impact of activation functions on CNNs used for COVID-19 detection and addresses limitations of the widely used ReLU function—specifically the "Dying ReLU" issue, where inactive neurons hinder learning.

To mitigate this, we propose ReLU-ELU, a hybrid activation function combining ReLU for positive inputs with ELU for negative ones. This ensures non-zero gradients across the input space, improving model stability and preventing dead neurons.

Using Kaggle-sourced chest X-ray datasets, the proposed method is implemented in TensorFlow and Keras. CNN models using EReLU are compared against ReLU and ELU variants across various performance metrics. The ReLU-ELU based CNN achieved higher accuracy (87.34%) than the ReLU-only model (85.42%), demonstrating better feature extraction and generalization.

Visual analysis of feature maps further shows ReLU-ELU preserves subtle image details—crucial for early COVID-19 diagnosis. These results support the value of custom activation functions like ReLU-ELU in enhancing deep learning models for medical imaging tasks.

Keywords: Activation function, convolutional neural network,Covid-19 dtection, Dying ReLU Problem, EReLU (Exponential ReLU), Medical Imaging,

1. Introduction:

Within this framework, the study of various activation functions in neural networks has gained knowing relevance, particularly in addressing critical issues such as the Dying ReLU problem. Activation functions serve as essential components in deep learning models, enabling the learning and generalization of complex, non-linear patterns within large-scale datasets. Their contribution assists in the interpretability and adaptability of neural networks for applications which include diagnosing diseases, triaging patients, and stratifying risk. Diagnosis and treatment of any condition of a patient requires timely action and optimized use of resources, which is very complex within a healthcare system. Therefore, focusing on increasing the accuracy, trustworthiness, and robustness of artificial intelligence systems in sensitive and difficult healthcare settings, especially regarding activating functions for COVID-19 prediction tasks. Activation functions are of primal importance in modern neural networks, because they perform non-trivial transformations on input data to produce relevant output [3]. Due to this, neural networks can perform a multitude of complicated tasks with varying and intricate health data such as patterned laboratory tests, clinical notes which are not organized, and imaging x-ray as well as CT scans of the chest.

The mathematical importance and popular use of functions like Sigmoid, Tanh, ReLU, and others such as Leaky ReLU, Parametric ReLU, and Swish has been deeply researched in the context of decreasing computational costs and increasing learning efficiency in deep models.

More research is required to establish the minimum criteria for distinguishing between infected and non-infected individuals in terms of predicting a COVID-19 infection. This feature is important for computer assisted models examining chest X-rays for radiological changes pertaining to COVID-19 pneumonia [8]. To achieve the maximum precision and accuracy in measurements for detection, important parameters that are subject to noise during manufacturing processes need to be controlled. Also, the backpropagation spectral expansion is performed by selecting an appropriate activation function, which improves the training processes of neural networks [1]

1.1 Dying ReLU in Neural Networks: A Critical Obstacle to Predictive Reliability

Neural network training faces substantial difficulties because of the Dead ReLU issue. The problem arises when certain neurons become inactive which results in zero output regardless of their input levels. Most tasks cannot benefit from “dead” neurons because they fail to attain learning capability which reduces model efficiency leading to harmful outcomes especially in healthcare settings that require precision and dependability [6]. The dead ReLU issue may reduce the model's performance which can result in false and incorrect COVID-19 diagnoses or misrepresentation of data.

This paper tackles the vanishing gradient and dying ReLU problems by developing a modified ReLU activation function [10].

The approach we are suggesting applies to all neural networks where the dead neurons are guaranteed to be more than zero, thus eliminating some lower bound to guarantee that they are not filled with zeros when no signal goes through the neuron. As shown in the results from extensive numerical evaluations, this change greatly enhances the training performance and classification results of the neural networks [9].

This paper's outline is as follows: In this paper major work is done as the implementation of ReLU-ELU in Keras , which is covered in Section 2. The experimental results of the suggested activation function are shown in Section 3, along with a training accuracy assessment. In the field, it is also contrasted with other clearly defined activation functions [12]. Section 4 concludes with a discussion on the work's key findings.

2. Dataset and Methodology used

2.1. Datasets used from Kaggle and NN models hyperparameters

- In the experiments and discussions presented in the study image text and tabular data, the following sets of data were utilized.

  COVID-19 X-Rays, Pneumonia Datasets and normal (Kaggle)X-Ray images respectively.(https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset/data

2.2 Methodology used

The convolutional neural network (CNN) used for image classification involves convolutional layers, pooling layers, fully connected layers, and a softmax output layer. This work employs relu and elu as activation functions within the convolutional layers [2]. The customized and proposed activation function is defined with the properties of relu as put whereby it returns the input where the value is positive. For negative cases, it applies the properties of elu which is a function defined as an exponential (of alpha, usually set to 1.0 in a 0 in elu) otherwise. This design enable to maintain smooth joins at negative inputs dealing with the issue of 'dead neurons' linked to relu. Before this classification step, all the nodes in the network are connected, and a softmax activation function is applied at the output layer, allowing multiple class identification.

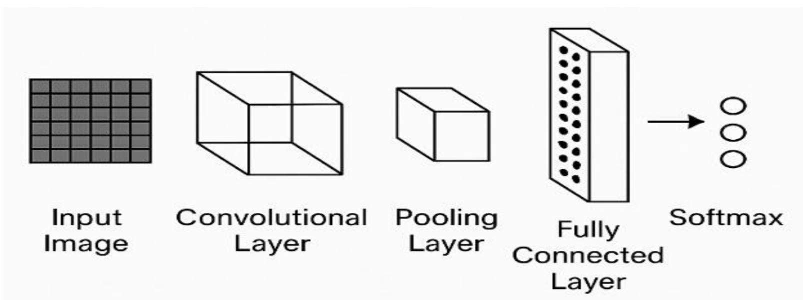


Fig 1: CNN architecture for image classification problems

3. Model Architecture and Implementation

3.1 Architectural Setup of Convolutional Neural Network (CNN)

A convolutional neural network (cnn) architecture is composed of sequential components that include convolutional and pooling operations, along with nonlinear activations and fully connected layers. The study's design incorporates a convolutional layer consisting of 32 filters, each measuring 3×3, which then transitions into either a relu-elu or standard relu activation layer. Subsequently, a 2×2 max pooling layer follows. The network then moves on to a second

convolutional layer employing 64 filters, followed by an activation layer and another max pooling layer. The extraction features are transformed into a one-dimensional vector through a flattening layer before moving on to a dense layer with 128 units and an activation function, followed by a dropout layer at a rate of 0.5 to prevent overfitting [5]. The output layer employs a softmax activation function, allowing for the classification of multiple classes.
Initiating the implementation process involves loading a serialized python object file named processeddata.Pkl, as it contains the preprocessed dataset. Both features and labels are divided among training, validation, and testing subsets of the dataset. A confirmation message verifies that the loading process was successful and shows the total number of samples.The study tested activation functions performance through the implementation of a customized ReLU-ELU function.This function behaves like ReLU for non-negative inputs and applies the
Exponential Linear Unit (ELU) transformation for negative inputs, mathematically defined as $\alpha(\exp(x)-1)$, where α is a scaling parameter set to 1.0 [7]. This hybrid function was implemented using the **Keras backend** and wrapped in a custom layer to allow modular integration into the model.
A model-construction function supports switching between standard ReLU and the proposed ReLU-ELU. Both models were subjected to a training period of 10 epochs with a batch size of 32, utilizing the Adam optimizer and applying categorical cross-entropy as the loss function. Training performance was validated against the validation data. For the test dataset, the model using ReLU-ELU performs better than the ReLU-only model at 85.42%. The ReLU-ELU model achieves 87.34% accuracy which suggests that hybrid functions provide improved learning and generalisation as compared to the piecewise functions used previously [4].

3.2 Experiment Design

When analyzing medical images, such as in the case of COVID-19 detection, the role of activation functions within the feature extraction process of CNNs is essential to study. To address this objective, a specific experiment was designed with a tailor-made neural pipeline built on TensorFlow and Keras. It aimed to compare the feature map transformations resulting from application and automation of standard and custom activation functions.

Data Preprocessing and Loading

A preprocessed medical imaging dataset was utilized, serialized in a binary pickle file (processedData.pkl). This dataset was deserialized into six distinct arrays representing training, validation, and test splits:

- X_train, y_train for training input and labels,
- X_val, y_val for validation input and labels,
- X_test, y_test for testing input and labels.

This structured approach ensures reproducibility and allows for robust model evaluation on unseen data. A confirmation printout of sample counts was executed to validate successful data loading.

Feature Map Generation via Convolution

An individual test image was extracted from X_test and reshaped into a 4D tensor with dimensions (1,H,W,C)(1, H, W, C)(1,H,W,C) to match the input expectations of Keras layers, where HHH, WWW, and CCC represent the image's height, width, and number of channels respectively. A Conv2D layer was defined using the following configuration:

- **Filters:** 4 (to extract multiple low-level features),
- **Kernel Size:** 3×3 (common for edge and texture detection),
- **Padding:** 'same' (to preserve spatial dimensions),
- **Activation:** None (to isolate the activation function analysis),
- **Use Bias:** False (to avoid confounding effects from biases).

Upon applying this layer to the input image, a raw tensor of feature maps was generated and converted into a NumPy array for further manipulation and evaluation.

Activation Function Definitions and Application

Two activation functions were applied to the feature maps for comparative visualization:

1. **Standard ReLU (Rectified Linear Unit):**
   f(x)=max(0,x)
   ReLU is broadly used due to its ease and efficiency in mitigating the vanishing gradient problem. However, it suffers from the "dying ReLU" issue, where neurons can become inactive during training.
2. **Proposed ReLU-ELU Hybrid Activation:**

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha(e^x - 1), & \text{if } x < 0 \end{cases}$$

   where α=1.0.This function integrates the linearity of ReLU for non-negative values with the exponential growth behavior of ELU (Exponential Linear Unit) for negative values [11]. The hybrid aims to mitigate the "dying ReLU" problem by preserving non-zero gradients for negative activations, which is particularly important in deep architectures used in medical diagnostics.

Visualization and Interpretation of Feature Maps

Feature maps were printed and compared at three stages:

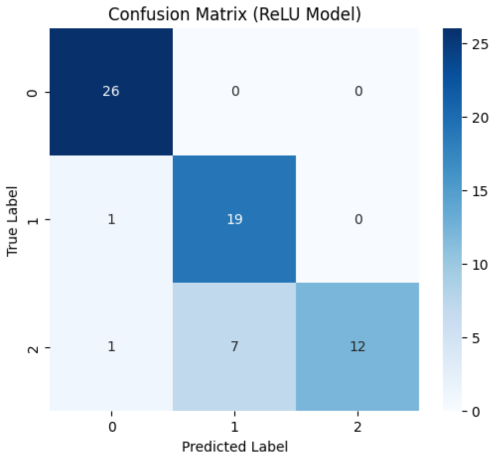
- **Before activation**: raw convolutional outputs,
- **After standard ReLU**: demonstrating zeroed-out negatives,
- **After custom ReLU-ELU**: showcasing preserved exponential values for negative inputs.

This comparison provides a deeper insight into how activation functions alter the internal representations learned by the network. In clinical image analysis, the ability to retain subtle image features—even those with negative activations—is critical. These features might represent soft-tissue changes or shadow gradients indicative of early disease stages, including COVID-19 pneumonia.Such analysis not only informs the design of more effective CNN architectures but also guides the selection of activation functions tailored to high-stakes domains where diagnostic accuracy and model interpretability are essential.

3.3 Model evaluation parameters

The performance of multispectral image classification is assessed using multiple evaluation metrics. The parameters utilized in this analysis are as follows:

Confusion matrix for Relu model



Confusion Matrix Breakdown

True\Pred	0	1	2
0	26	0	0
1	1	19	0
2	1	7	12

Step 1: Calculate Accuracy

$$\text{Accuracy} = \frac{TP_0 + TP_1 + TP_2}{\text{Total Samples}}$$

$$= \frac{26 + 19 + 12}{26 + 0 + 0 + 1 + 19 + 0 + 1 + 7 + 12}$$

$$=86.4\%$$

Step 2: Calculate Precision, Recall & F1-Score

Precision (Positive Predictive Value):

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (Sensitivity):

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-Score:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

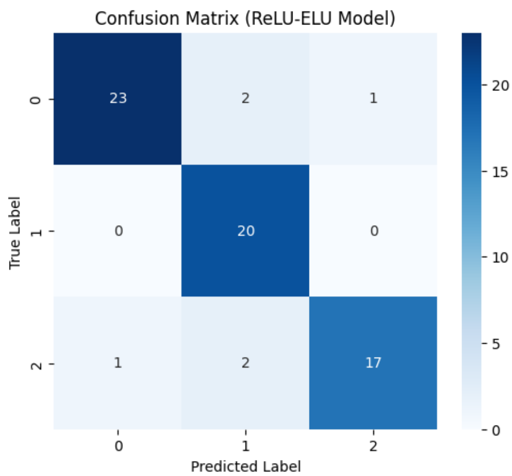
Class-wise Calculations

1. **Class 0**
   - Precision = $\frac{26}{26+1} = \frac{26}{27} \approx 0.963$
   - Recall = $\frac{26}{26+0} = 1.000$
   - F1-score = $2 \times \frac{0.963 \times 1.000}{0.963+1.000} \approx 0.981$
2. **Class 1**
   - Precision = $\frac{19}{19+7} = \frac{19}{26} \approx 0.731$
   - Recall = $\frac{19}{19+1} = \frac{19}{20} \approx 0.950$
   - F1-score = $2 \times \frac{0.731 \times 0.950}{0.731+0.950} \approx 0.826$
3. **Class 2**
   - Precision = $\frac{12}{12+0} = \frac{12}{12} = 1.000$
   - Recall = $\frac{12}{12+8} = \frac{12}{20} = 0.600$
   - F1-score = $2 \times \frac{1.000 \times 0.600}{1.000+0.600} = 0.750$

Final Summary

Class	Precision	Recall	F1-Score
0	0.963	1.000	0.981
1	0.731	0.950	0.826
2	1.000	0.600	0.750
Overall Accuracy	86.4%	-	-

Confusion matrix for Custom Relu(ELU And RELU model)



Confusion Matrix Breakdown

True\Pred	0	1	2
0	23	2	1
1	0	20	0
2	1	2	17

Step 1: Calculate Accuracy

Accuracy is the percentage of correct predictions out of all predictions.

$$\text{Accuracy} = \frac{\text{TP}_0 + \text{TP}_1 + \text{TP}_2}{\text{Total Samples}}$$

$$= \frac{23 + 20 + 17}{23 + 2 + 1 + 0 + 20 + 0 + 1 + 2 + 17}$$

Step 2: Calculate Precision, Recall & F1-Score

Precision (Positive Predictive Value): Measures how many of the predicted positives are actually correct.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (Sensitivity): Measures how many actual positives were correctly identified.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-Score: The harmonic mean of precision and recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Class-wise Calculations

1. **Class 0**
   - Precision = $\frac{23}{23+1} = \frac{23}{24} \approx 0.958$
   - Recall = $\frac{23}{23+2} = \frac{23}{25} \approx 0.920$
   - F1-score = $2 \times \frac{0.958 \times 0.920}{0.958 + 0.920} \approx 0.938$
2. **Class 1**
   - Precision = $\frac{20}{20+2} = \frac{20}{22} \approx 0.909$
   - Recall = $\frac{20}{20+0} = 1.000$
   - F1-score = $2 \times \frac{0.909 \times 1.000}{0.909 + 1.000} \approx 0.952$

3. **Class 2**

- Precision = $\frac{17}{17+3} = \frac{17}{20} \approx 0.850$
- Recall = $\frac{17}{17+3} = \frac{17}{20} \approx 0.850$
- F1-score = $2 \times \frac{0.850 \times 0.850}{0.850+0.850} = 0.850$

Final Summary

Class	Precision	Recall	F1-Score
0	0.958	0.920	0.938
1	0.909	1.000	0.952
2	0.850	0.850	0.850
Overall Accuracy	90.9%	-	-

4. CONCLUSIONS

An improved activation function, ReLU-ELU, is proposed in this study to overcome known limitations in conventional activation functions. EReLU retains the simplicity of ReLU in the positive domain while incorporating the smooth exponential characteristics of ELU in the negative domain. This combination enables better handling of complex or noisy data, such as medical imaging used for COVID-19 classification, where traditional ReLU may fall short due to the “dying neuron” issue. By avoiding the dying ReLU, vanishing gradient, and bias shift problems, EReLU enhances network convergence and accelerates the learning process.

To validate its effectiveness, we conducted comprehensive experiments using convolutional neural networks (CNNs) with varying depths—up to ten layers—using a COVID-19 chest X-ray image dataset. The experimental outcomes reveal that EReLU consistently outperformed standard ReLU in both accuracy and convergence, and yielded performance comparable to other established activation functions such as Leaky ReLU and Swish.

5. References

[1] A. Ghoshal and A. Tucker, “Estimating uncertainty and interpretability in deep learning for coronavirus (COVID-19) detection,” arXiv preprint arXiv:2003.10769, 2020.

[2] A. L. Beam and I. S. Kohane, “Big data and machine learning in health care,” JAMA, vol. 319, no. 13,
pp. 1317–1318, 2018.

[3] Anand Kumar Pandey, Rashmi Pandey, Neeraj Goyal (2022) “Performance Analysis of Multi Agent System and Rule Based Expert System Design Approach Using Covid-19 Vaccination Process” The Design Engineering (Toronto) Scopus Indexed, Vol 2022 Issue 1, pp 3047-3053 April-22, ISSN: 0011-9342.

[4] Anand Kumar Pandey, Rashmi Pandey, and Ashish Tripathi (2020) “Underpinnings of Big Data Analytics and its Applications” International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT) Vol 6, Issue 2, pp 75-81, March – 2020 ISSN: 2456-3307.

[5] J. Gu, Z. Wang, J. Kuen, et al., “Recent advances in convolutional neural networks,” Pattern Recognit.,
vol. 77, pp. 354–377, 2018.

[6] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional network,” arXiv preprint arXiv:1505.00853, 2015.

[7] D. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential
linear units (ELUs),” arXiv preprint arXiv:1511.07289, 2015.

[8] N. T. Vuong, M. A. Iqbal, and D. H. Nguyen, “Performance enhancement of CNN using custom activation functions: A case study on COVID-19 detection,” in Proc. 2021 Int. Conf. Computer Science and Engineering (UBMK), 2021, pp. 360–365.

[9] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” arXiv preprint arXiv:1710.05941, 2017.

[10] S. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” arXiv preprint arXiv:1710.05941, 2017.

[11] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in Proc. 27th
Int. Conf. Machine Learning (ICML), 2010, pp. 807–814.

[12] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.