Comparative Study of Convolution Neural Network's Relu and Leaky-Relu Activation Functions

Arun Kumar Dubey and Vanita Jain

Abstract Convolutional neural networks refer to a collection of feed-forward artificial neural networks. These networks have been implemented successfully on visual imagery. It uses a variety of perceptrons. These perceptrons are multilayered, that need very little preprocessing. Shift invariant or space invariant NN are alias for CNN, because of their architecture which is based on shared weights. It is also established on translation invariance features. In this paper, we have used rectified linear unit (Relu) and Leaky-Relu activation for inner CNN layer and softmax activation function for output layer to analyze its effect on MNIST dataset.

Keywords Activation function · Relu · Leaky-Relu · CNN

1 Introduction

Deep learning is a machine learning method that has made extraordinary progress in fields like picture acknowledgment and discourse acknowledgment. There are four kinds of deep learning models [1]. Different models are as follows:

- Stacked autoencoder [2]: This model is generally built by stacking a few autoencoders. It consists of two phases, i.e., encoding and decoding stages. An autoencoder figures out how to pack information from the info layer into a short code and after that uncompress that code into something that intently coordinates the first information.
- Deep belief network (DBN) [3]: DBN is a generative graphical model, or on the other hand, a class of profound neural systems, made out of numerous layers of

A. K. Dubey
University School of Information, Communication and Technology, GGSIPU New Delhi, Delhi 110078, India
e-mail: arudubey@gmail.com

V. Jain (✉)
Bharati Vidyapeeth's College of Engineering New Delhi, Delhi, India
e-mail: Vanita.jain@bharatividyapeeth.edu

S. Mishra et al. (eds.), *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, Lecture Notes in Electrical Engineering 553,
https://doi.org/10.1007/978-981-13-6772-4_76

inert factors, with associations between the layers and not between units inside each layer. At the point when prepared on an arrangement of cases without supervision, a DBN can figure out how to probabilistically remake its sources of information. After this learning step, a DBN can be additionally prepared with supervision to perform arrangement. It is stacked with numerous confined Boltzman machines that use Gibbs inspection to prepare the illustrations.
- Convolutional neural network [4]: This is the most commonly utilized deep learning method for large-scale picture characterization. This model consists of an input layer and an output layer along with many hidden layers. The hidden layers of CNN model mainly consist of convolutional layer, pooling layer, and fully connected layer. **Convolutional layer** applies a convolution operation to the input and then passes the result to the next layer. **Pooling layer** combines the output of neuron cluster at one layer into another single neuron in the subsequent layer. **Fully connected layer** is the interface between each neuron in one layer and each neuron in another layer.
- Recurrent neural network [5]: It is another type of deep learning model where associations between units shape a coordinated diagram along an arrangement. It is learned to highlight the arrangement of information by memory of past data sources that are put away in the inner condition of neural systems.

Deep learning has been used in all the aspects of research work and significantly used in computer vision [6].

CNN [7] was initially inspired by biological processes as compared to the patterns of links of neurons. This pattern resembles the origin of visual cortex of animals. This is built on the response of stimuli and their restricted regions, called receptive field, which partially overlap to gain the entire coverage area.

2 Role of Activation Function in Analysis

Activation functions have very important role in analysis. Both linear unit, not good for complex dataset, and nonlinear unit, mostly used for multiple features, have different effects on different situations.

These functions can capture complex nonlinear relationship. Apart from learning from continuous data, it has a capability to learn from categorical data. Functions are important for biases and weights in the artificial neural network to perform in nonlinear function. With the activation function, the back propagation has become possible since the gradients can update the weights and biases on the basis of error values they have. These functions are monotonic; therefore, error surface associated with the model is guaranteed to be convex.

We cannot say that all activation functions can handle all situations; for example: Sigmoid has slow convergence rate and also kills gradients; Tanh has a problem called vanishing gradient; Relu: Some gradients may be soft/fragile while training the data and may die. In other words, it can result in dead neurons. Binary step: This

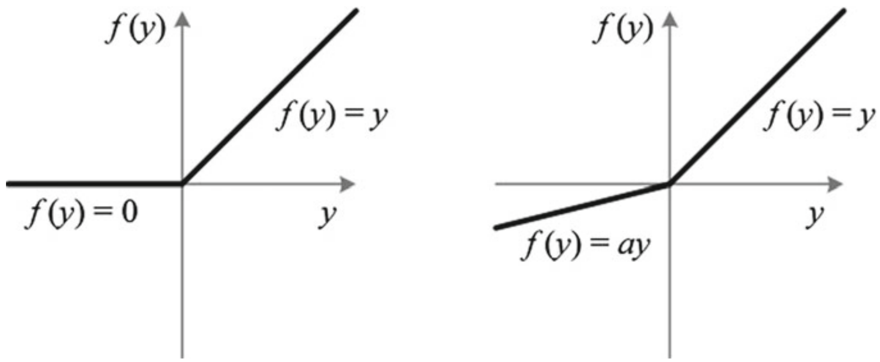

Fig. 1 Comparison between Relu and Leaky-Relu [9]

Table 1 Mathematical expression of Relu and Leaky-Relu

	Relu	Leaky-Relu
Function	$F(z) = \begin{cases} 0, & z<0 \\ z, & z \geq 0 \end{cases}$	$F(z) = \begin{cases} 0.01z, & z<0 \\ z, & z \geq 0 \end{cases}$
Derivative	$F'(z) = \begin{cases} 0, & z<0 \\ 1, & z \geq 0 \end{cases}$	$F'(z) = \begin{cases} 0.01, & z<0 \\ 1, & z \geq 0 \end{cases}$

function is not continuously differentiable (at 0), therefore not suitable for gradient-based optimization. Leaky-Relu: It suffers from exploding gradient problem during front propagation problem if rate of learning is set too high.

Rectified linear unit (Relu) [8] and Leaky-Relu are very popular activations in current era, but there is some difference between these two. It is shown in Fig. 1. Both are monotonic and their differentiation is also shown in Table 1.

Softmax is a type of regression technique, specifically logistic regression technique, where we consider multiple classes at the same time instead of binary classes, so the output label y can take on K different values, rather than only two. Thus, we have our training set given as below:

$$\{(x(1), y(1)),\ \ldots,\ (x(m), y(m))\}\{(x(1), y(1)),\ \ldots,\ (x(m), y(m))\},$$

For this particular training set, we have our label having values as below:

$$y(i)\{1, 2,\ \ldots,\ K\}y(i) \in \{1, 2,\ \ldots,\ K\}$$

One example of softmax regression is digit recognition through MNIST in which images of digits along with their corresponding labels are stored.

3 Experimental Results

We have used simple CNN on hand-written MNIST dataset which is part of NIST and used Relu and Leaky-Relu activation functions and softmax activation function for output layer. Figure 2 shows this dataset.

Fig. 2 MNIST dataset

Problem Type: Multi-class classification (Supervised learning)	
Image size:	28 × 28
Samples:	60,000
Input size:	60,000 × 1 × 28 × 28
Convolutional network configuration	
Number of RGB channels:	1
Layers of CNN:	1 (simple CNN)
Number of filters:	32
Size of filters:	8 × 8
Polling method used:	Max-Polling
Polling size:	2 × 2
Layers of MLP:	2
Number of nodes in input MLP layer:	128
Number of nodes in output MLP layer:	10
Dropout percent:	40% (Purpose: To avoid overfitting)
Number of epochs:	5
Batch size:	32

Model summary

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 32, 21, 21)        2080
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 32, 21, 21)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 32, 10, 10)        0
_________________________________________________________________
dropout_3 (Dropout)          (None, 32, 10, 10)        0
_________________________________________________________________
flatten_3 (Flatten)          (None, 3200)              0
_________________________________________________________________
dense_5 (Dense)              (None, 128)               409728
_________________________________________________________________
dense_6 (Dense)              (None, 10)                1290
=================================================================
Total params: 413,098
Trainable params: 413,098
Non-trainable params: 0
_________________________________________________________________
```

(a) **Learning Curve**:

Training and testing curves are approaching each other; therefore, CNN model used is not affected by the problem of overfitting and underfitting. As per following results of learning curve, digit model accuracy and model loss on Relu are better than Leaky-Relu (Fig. 3).

(b) **Top *K* categorical accuracy**:

It is a metric function to calculate accuracy on dataset which shows accurate prediction of model. After using both activation functions, it is found that epoch Relu constantly performed better than Leaky-Relu. Keras function has the following format:

top_k_categorical_accuracy(Q_true,Q_pred,k) here Q_true represents True label and Q_pred shows prediction. best result we get on $k = 5$ value.

Figure 4 depicts the accuracy curve for top three most accurately recognizable classes. It shows that there exist the classes which are more easily recognizable than other classes with an accuracy of nearly 100%. Therefore, it means that the test score might be misleading sometimes, and there is a need for other metrics such as F-score and confusion metrics. Misleading accuracy due to existing simple examples can be avoided by using stratified *k*-fold cross-validation.

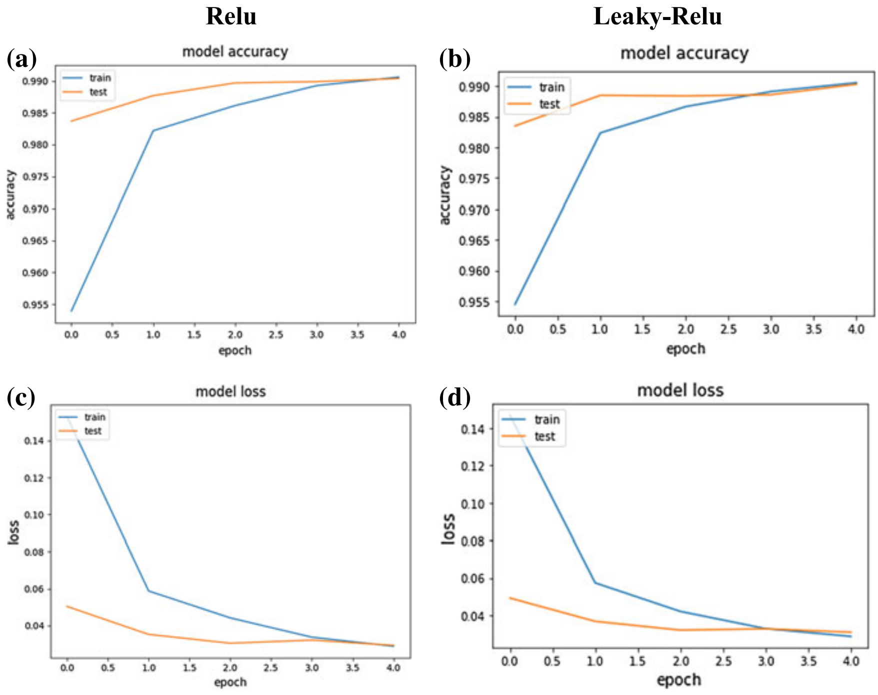

Fig. 3 Model accuracy and model loss

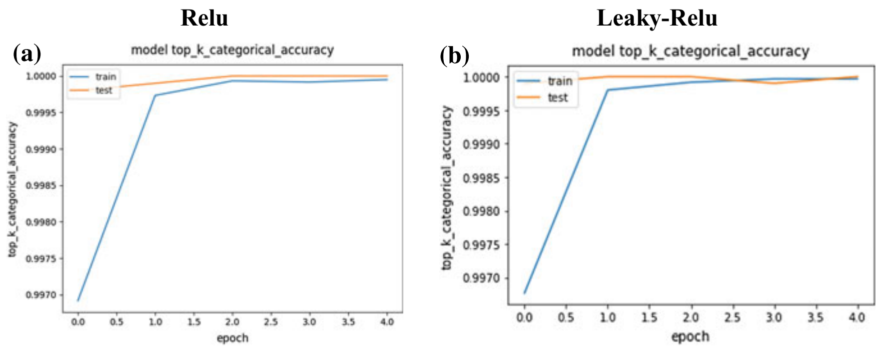

Fig. 4 Metric function for accuracy

(c) **Recall**:

It shows that fraction of positive predictions out of positive samples is increasing, i.e., false negative is decreasing (Fig. 5).

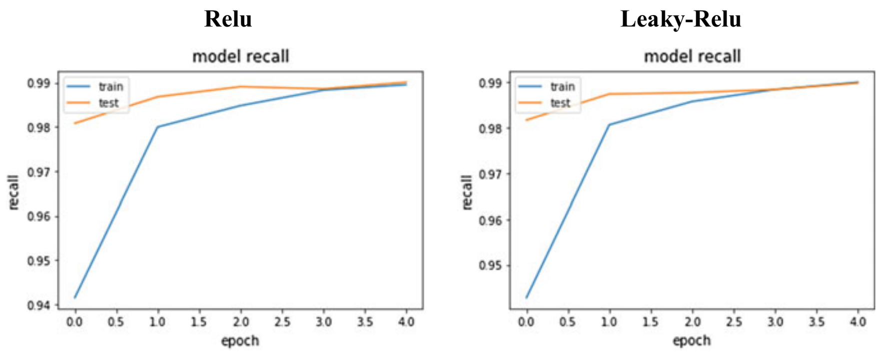

Fig. 5 Recall of Relu and Leaky-Relu

$$\text{Recall} = \frac{\text{digits correctly identified}}{\text{digits correctly identified} + \text{individuals incorrectly identifed as not digit}}$$

(d) **FMeasure**:

Curve shows the progress in F-score with increasing number of epochs. Increasing F-score shows that the model is better than models making most frequent prediction and making random prediction; it is weighted harmonic mean (HM) of precision and recall. Figure 6 shows that increasing epoch value leads to equivalent F-score.

$$P: \ \text{Precision}$$

$$P = \frac{\text{digits correctly identified}}{\text{digits correctly identified} + \text{individuals incorrectly identifed as digit}}$$

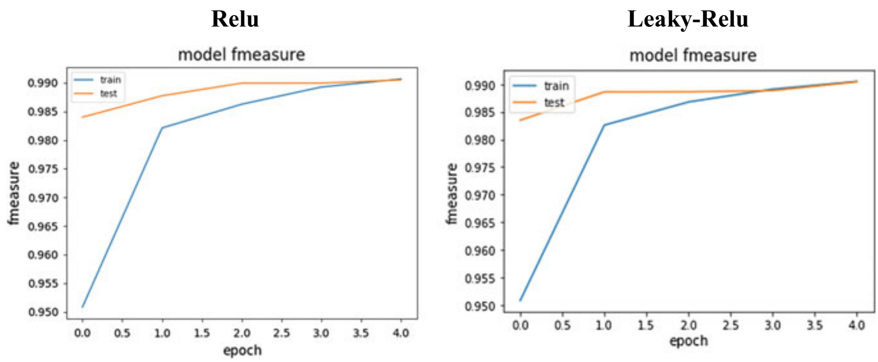

Fig. 6 F-Score of Relu- and Leaky-Relu-based activation function modes

$$F - \text{score} = 2\frac{\text{Precision} * \text{Recal}}{\text{Precision} + \text{Recal}}$$

4 Conclusion

Convolution neural network for deep learning is highly and effectively used in this world, and rectified linear unit and Leaky-Relu are famous activation functions. Digit recognition in MNIST, hand-written digit dataset, using these functions has delivered good result. Model accuracy and model loss on Relu are better than Leaky-Relu, but on comparison of Recall and F-score, Leaky-Relu works effectively and gives better accuracy. Top *k* category metric function is used to calculate accuracy on dataset which shows accurate prediction of model. This metric performed better on k value at five. This paper has shown the comparative results on activation functions, while CNN layer had not been changed for MNIST dataset.

References

1. Gheisari M, Wang G, Bhuiyan MZA (2017) A survey on deep learning in big data. In: 2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC), Guangzhou, pp 173–180
2. Du B, Xiong W, Wu J, Zhang L, Zhang L, Tao D (2017) Stacked convolutional denoising auto-encoders for feature representation. IEEE Trans. Cybern 47(4):1017–1027
3. Kamada S, Ichimura T (2016) An adaptive learning method of Deep Belief Network by layer generation algorithm. In: 2016 IEEE region 10 conference (TENCON), Singapore, pp 2967–2970
4. Manana M, Tu C, Owolawi PA (2017) A survey on vehicle detection based on convolution neural networks. In: 2017 3rd IEEE international conference on computer and communications (ICCC), Chengdu, pp 1751–1755
5. Shukla P, Biswas KK, Kalra PK (2017) Recurrent neural network based action recognition from 3D skeleton data. In: 2017 13th international conference on signal-image technology & internet-based systems (SITIS), Jaipur, pp 339–345
6. Koohzadi M, Charkari NM (2017) Survey on deep learning methods in human action recognition. IET Comput. Vis 11(8):623–632
7. Takahashi N, Gygli M, Pfister B, Van Gool L (2016) Deep convolutional neural networks and data augmentation for acoustic event recognition. In: Proceedings of interspeech, Sept 2016
8. Hang ST, Aono M (2016) Open world plant image identification based on convolutional neural network. In: 2016 Asia-Pacific signal and information processing association annual summit and conference (APSIPA), Jeju, pp 1–4
9. Xu B, Wang N, Chen T, Li M (2015) Empirical evaluation of rectified activations in convolutional network. ICML deep learning workshop, Lille, France, 06–11 July 2015