Convolutional Neural Network (CNN) with Randomized Pooling

Hafiz Imran
University of Sargodha
Khalid Aamir (khalid.aamir@uos.edu.pk)
University of Sargodha https://orcid.org/0000-0002-6852-7031
Muhammad Ilyas
University of Sargodha

Article

Keywords:

Posted Date: June 2nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1697292/v1

Convolutional Neural Network (CNN) with Randomized Pooling

Hafiz Muhammad Imran, Khalid Mahmood Aamir, Muhammad Ilyas

Department of Computer Science and IT, University of Sargodha, Sargodha, Pakistan

Email: imran.sargodha@gmail.com, khalid.aamir@uos.edu.pk, muhammad.ilyas@uos.edu.pk

***Abstract*-Convolutional Neural Network (CNN) is a deep learning approach to solve complex problems, and it has been widely used in image processing for image classification, object identification, semantic segmentation etc. It has overcome the constraint of traditional machine learning approaches. There has been a lot of effort done to improve the accuracy of CNN in many application areas, but there has been a lesser amount of work done to reduce the computational complexity of this model. There is a need to improve CNN's complexity problem. Here we have introduced randomized pooling (RANDpool) to CNN. Randomized pooling has reduced the computational complexity cost of CNN. We used MNIST dataset to demonstrate CNN with randomized pooling. This paper used randomization technique to reduce the dimensions of image instead of max, min or average pooling that have very extensive computations.**

I. Introduction

Deep Learning evolved for the solution of complex problems in the field of image classification, object identification, semantic segmentation etc. The demand for learnable algorithms increased, and researchers took an interest in deep learning to develop many learnable techniques and algorithms. In response to how machines acquire knowledge? Researchers observe the human activity of learning and incorporate the learning process in the machine, which evolved the term Machine Learning. The self-learning process from past experiences is the key feature of machine learning that reduces the efforts. The conventional machine learning techniques need extraction of features as a precondition, and there must be a domain expert. This is a challenging task to select proper features for under-considered problems. There is no need to feature extraction in Deep Learning because it automatically extracts the significant feature from raw input for a given problem. The model of Deep Learning has many processing layers that can extract various features from data by using different abstraction levels. The basic model used in Deep learning is artificial neural network with multiple layers.

The basic model of Artificial Neural Network (ANN) consists of one input layer and one output layer with multiple hidden layers.

The x_j is input neuron takes the input and $h_i$output produced by output neuron. The connection between each node have a number that is call *weight*. The value of the weight matrix is adjusted in the training process of the network. The representation of output is given in equation (1) and general ANN shown in figure 1 [1].

$$h_i = \sigma\left(\sum_{j=1}^{N} V_{ij}\, x_j + T_i^{hid}\right) \qquad (1)$$

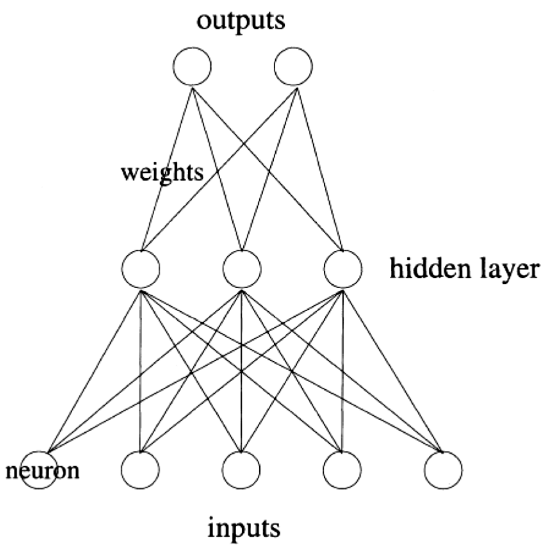


Figure 1: Basic architecture of ANN [1]

Where σ is activation function, v_{ij} represents the weights between nodes and T_i^{hid} is threshold value. There are many variant of ANN used to address the problems of different areas of Deep learning.one of them is Convolutional Neural Network that is under consideration of this paper.

Convolutional Neural Network (CNN) is one of the deep learning techniques widely used in various applications. CNN has achieved high-level results in speech analysis, image recognition, topic classification, sentiment analysis, natural language processing, language translation, and signal processing. CNN has the ability to classify based on contextual information. The typical model of CNN is given in figure 2.

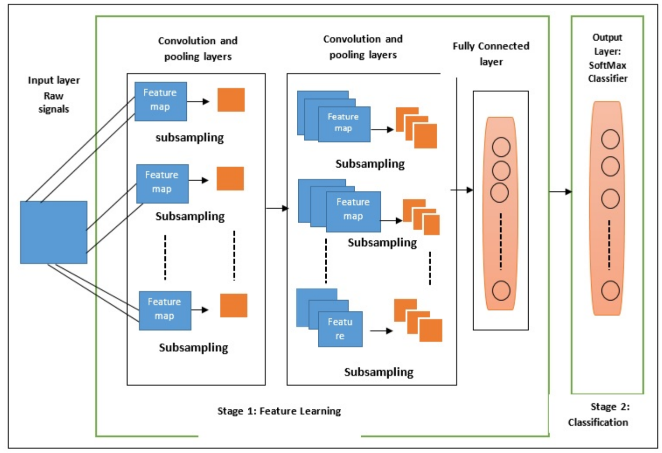


Figure 2: Basic architecture of a CNN [2]

The CNN has a convolution layer, pooling layer, activation function and fully connected layer. The convolution layer takes the image as input and produces a feature map of input by applying the convolution operation. The pooling layer reduces the data by taking average or selecting maximum or minimum values from the feature map. The activation function normalizes the feature map's values and passes high weight values to the next layer. A fully connected layer gives a classified result to the output layer [2]

The convolution layer is the central part of the neural network and the purpose of this layer is to characterize the input. This layer includes several feature maps that extract local features from different places. There are different activation function are used but common activation function is Rectified Linear Unit (ReLU) [2].

The effect of the pooling layer is the extraction of secondary features that reduce the dimensions and increase the strength of feature mining. The pooling layer mostly works between two convolutional layers. The dimension of the feature map depends upon the size of kernels and stride. In the pooling layer, average, min or max operations are applied. The convolution layer and pooling layer extract the high-level features of inputs

A convolutional neural network's classifier is usually contains one or more completely interconnected layers. All neurons from the preceding layer are extracted and joined to each neuron in next layer. The completely connected layers does not store spatial information. The output layer comes after the last completely linked layer. In order to classify, Softmax regression is commonly utilized to produces high-quality results [2].

Extensive literature is available on using sigma-shaped activation functions in existing machine learning algorithms, rectified linear unit (ReLU) has proven to be superior to the former. To begin with, calculating the partial derivative of ReLU is straightforward. Second, taking into account training time as one of the factors, the saturation ratio sigmoid-like linearity f (x) = (1 + e-x-1) is slower than the unsaturated one. Nonlinearity like ReLU given in equation 2 [3].

$$f(x) = \begin{cases} 0\,, if\ x < 0 \\ x, if\ x \geq 0 \end{cases} \qquad (2)$$

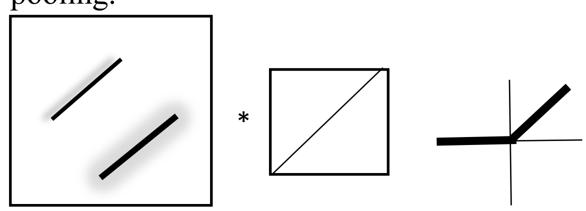
This section introduced the basic architecture of CNN and function of each layer. There are several researches have done to improve the accuracy of this model but lesser amount to reduce time complexity. Related work done in different research paper are given in next section. The basic objective of this paper to reduce the number of computations and reduce the computation time that improver the overall performance of CNN. The methodology used in this paper given in methodology section after the related work and experimental results are given in result section.

II. Literature Review

Matthew uses the idea of a stochastic process in pooling [4]. He computes the probability of each pooling region and passes the selected value for activation. According to his claim, max pooling selects only maximal values, while stochastic pooling also utilized non-maximal values. Stochastic pooling process shown in figure 4 and stochastic operation given in equation 3.

$$P_i = \frac{a_i}{\sum_{K \in R_i} a_k} \qquad (3)$$

The stochastic pooling process has huge computation compared to average pooling, and the computational complexity of average pooling is greater than proposed randomized pooling.



a) Image *b) Filter* *c) Rectified Linear*

1.6	0	0
0	0	0
0	0	2.4

d) Activation, a1

0.4	0	0
0	0	0
0	0	0.6

e) Probabilities, p1

Sample location 1.6

From *p (): e.g., l=1*

f) Sampled activation, s

Figure 3 stochastic process in pooling [4]

Biologically inspired pooling is LP-Pooling which is modelled on complex cells. The summary of operations is given in equation (4), where I and O are the input and output feature maps, respectively, and G is the Gaussian kernel. LP-pooling increases the weights of prominent features and decreases weaker features. The behavior of LP-pooling becomes average pooling when P=1 and gives results like max-pooling when the value of p=∞. [5]

$$O = \sum\sum I(i,j)^p \times G(i,j))^{1/p} \qquad (4)$$

In [6] spatial pooling is incorporated in the algorithm of visual reorganization where nearby feature detectors are combined that preserve task-related information and remove irrelevant information. To achieve more compact representations, better robustness to noise, invariance to image transformations, clutter pooling is used.

A comparative study of different pooling techniques is provided in [7]. This study provides the accuracy of RunPool pooling, Max pooling and average pooling. The classification performance of the model is calculated by incorporating different pooling functions. By fixing Max pooling hyper parameters, tuning, and environmental performance, Runpool pooling and average pooling are measured. In the 1st experiment, the pooling layer is replaced with the convolution layer, which provides a poor classification result. It means that CNN without pooling layer provides poor results. The experimental comparison shows that non-deterministic pooling (Runpool pooling) is better than deterministic pooling (max pooling, average pooling etc.). This study also shows that accuracy will decrease with the large number of epochs.

To aggregate information within a region, handcraft pooling operations are used, but this does not minimize training error. This drawback leads to a new learned pooling operation called LEAP (LEAring Pooling). In this technique, the learning process is executed for each feature map, and one shared linear combination of the neuron technique is used. This technique gives results identical to average pooling when all weights are equal. In this study, CNN's Convolution layer is replaced with simplified convolution, and the combination of this layer and LEAP provides state of the art classification performance on three object reorganization benchmarks name as (1) CIFAR10 dataset (2) CIFAR100 dataset (3) ImageNet dataset.

There are many pooling methods implemented in different papers to improve the accuracy of the CNN but there is no work done on time complexity. This paper address the problem of time complexity of the CNN to improve the efficiency. Randomized technique is implemented in pooling layer of CNN that minimize the computation without decreasing accuracy. In the next section implemented method is presented and explanation of dataset used.

III. Methodology

In this paper the MINIST dataset is used that has 28x28 handwritten images of grayscale. The first convolution operation applied on 28x28 images with kernel size 9x9 and having stride 1 that gives the output of feature maps 20x20 images. The second convolution iteration is applied to the output of the first iteration images with a kernel size of 9x9 and stride 2 which gives 6x6 images. The formula of image size is given equation (5).

$$M = \frac{i_s - k_s + \mathbf{1}}{n_s} \quad (5)$$

Where i_s is the size of input image and k_s is the size of the kernel applied on input image, and n_s is the number of strides. Pooling is applied after each convolution operation that reduces the dimension of image. There are different methodologies used in different research papers for pooling operations. The methodology used in this paper is given below.

A. Definition of pooling window:

Pooling window, *w* is a square matrix of size $k \times k$ on which pooling is applied. *w* is defined as

$$w = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ \vdots & \vdots & \cdots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kk} \end{bmatrix}$$

Converting to linear indices, define as,
$p = k(i-1) + j$ for i, j=1…, k
Therefore x_{ij} maps to y_p
It is obvious that p =1, …, k^2.
In average pooling $x_{avg\_pooling = \frac{1}{k^2} \sum_{p=1}^{k^2} y_p}$
Expected pooled value is:

$$E[x_{avg\_pooling}] = E\left[\frac{1}{k^2}\sum_{p=1}^{k^2} y_p\right]$$

Assumed that all $y_p(p = 1 \cdots, k^2)$ are independent and identically distributed (IID). Therefore

$=\frac{1}{k^2} \sum_{p=1}^{k^2} E[y_p]$

$$E[x_{avg\_pooled}] = \frac{1}{k^2} \cdot k^2\, E[y_p] = E[y_p]$$

From this, we defined randomized pooling as choosing any of y'_{ps} .

B. Randomized pooling:

Pick any element of w randomly, say y_p (linearly indexed).
Due to the assumption of IID,
$E[x_{avg\_pooled}] = E[y_p]$.
Hence, average pooling and randomized pooling work are the same way. The distribution of the original image, average image and random image is given in figure 4.

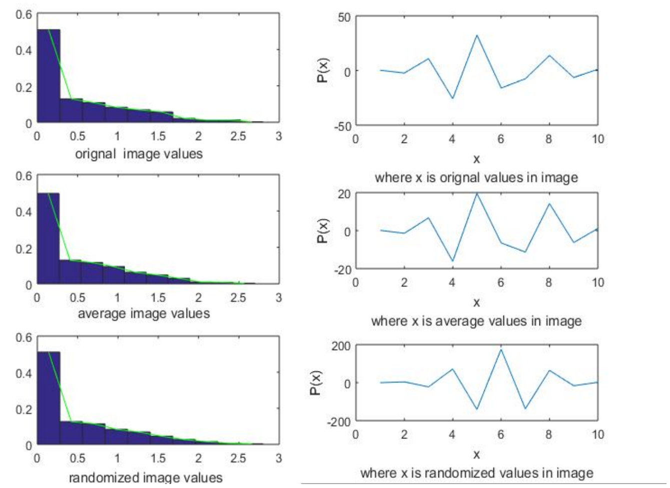


Figure 4 Regression coefficient for nine degree polynomial

IV. THE EXPERIMENTAL RESULTS

In typical pooling layer, max pooling and average pooling operations take massive calculations, while the proposed RANDpool pick a value from the fixed position without any calculation that reduces the pooling layer's time complexity. The number of operations in average pooling calculated as given below

$$2 \times (n^2 \times 256 \times no\ of\ images\,)$$

Where n^2 is the size of images, and 256 is the number of filtered copies. From the MNIST dataset, we use 8000 images of size 28×28 for training and 2000 images for testing. The number of filters is 256 each have 9x9 size and two pooling layers. The number of key operations is given below that is a very large number.

$$2 \times 28 \times 28 \times 256 \times 8000.$$

The proposed technique reduces the time complexity and improves the accuracy given in figure 5.

The convolutional neural network algorithm runs on the MNIST dataset to recognize images using average pooling and random pooling (RANDpool) on the same machine; the results are given in Table 1.

Let A be an image of size $n \times n$ size of window for pooling in $p \times p$ such that $p \ll n$

No of Computation $= \left(\frac{n}{P}\right)^2 . P^2 = n^2$

In randomized pooling, we pick numbers (from each window) at a fixed position. Therefore, there is only indexing and no computation.

Table 1: Training time, testing time, and accuracy of Avg-Pooling and RANDpool

#Epoc	Pooling method	Training Time	Testing Time	Accuracy in %
1	Avg Pooling	73.02	7.90	87.3
	Rand Pooling	65.30	6.25	87.25
2	Avg Pooling	230.45	10.97	92.35
	Rand Pooling	131.31	5.84	92.50
3	Avg Pooling	313.87	10.70	94.10
	Rand Pooling	197.22	6.03	94.85
4	Avg Pooling	433.50	11.13	95.05
	Rand Pooling	265.71	5.84	95.40
5	Avg Pooling	481.82	11.54	96.00
	Rand Pooling	330.95	5.81	96.50
6	Avg Pooling	488.16	8.40	96.35
	Rand Pooling	393.93	5.87	96.35
7	Avg Pooling	511.00	8.50	96.30
	Rand Pooling	461.90	5.99	96.40
8	Avg Pooling	585.83	7.97	96.30
	Rand Pooling	532.65	5.88	96.35
9	Avg Pooling	673.22	7.97	96.60
	Rand Pooling	603.49	5.68	96.85
10	Avg Pooling	743.55	7.95	96.85
	Rand Pooling	666.44	5.87	96.95

.

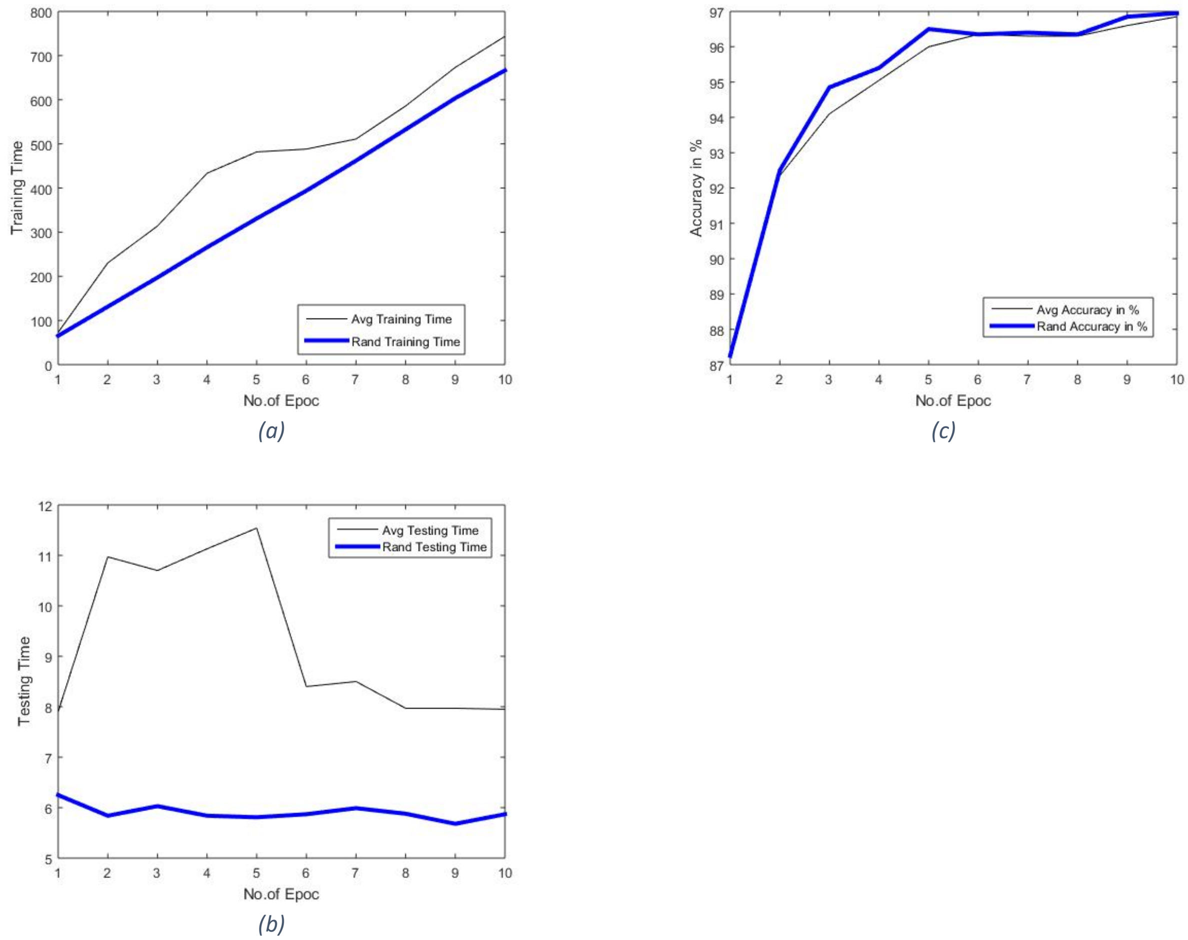


Figure: 5 Comparative results (Training Time (a), Testing Time (b), and Accuracy (c)) of Average and Randomized pooling

V. Conclusion

The time complexity have key role in any algorithm of computer science's application. This manuscript introduce randomization technique to reduce the time complexity of Convolutional neural network that improve the efficiency of CNN without changing in original data and improve accuracy. The use of this technique will improve time complexity of other Neural Networks that implemented pooling layers.

References

[1] S.-C. Wang, "The Springer International Series in Engineering and Computer Science book series," in *Interdisciplinary Computing in Java Programming*, Springer, p. 81–100.

[2] X. X. ,. W. C. T. Lei Si, "A Deep Convolutional Neural Network Model for Intelligent Discrimination between Coal and Rocks in Coal Mining Face," *Hindawi Mathematical Problems in Engineering,* vol. 2020, p. 12, 202.

[3] S. I. a. C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," *arxiv,* vol. 1502, 2015.

[4] R. F. Matthew D. Zeiler, *Stochastic Pooling for Regularization of Deep Convolutional Neural Networks,* New York, 2013.

[5] P. W. Huang Jin Jie, "RunPool: A Dynamic Pooling Layer for Convolution Neural Network," *International Journal of Computational Intelligence Systems,* vol. 13, no. 1, p. 66–76, 2020.

[6] L. D. O. A.-H. and D. Y. , "A DEEP CONVOLUTIONAL NEURAL NETWORK USING HETEROGENEOUS POOLING FOR TRADING ACOUSTIC INVARIANCE WITH PHONETIC CONFUSION".

[7] M. and Z. , "Learning Pooling for Convolutional Neural Network," *Neurocomputing,* vol. 224, pp. 96-104, 2017.

[8] K. F. a. S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," *in Competition and cooperation in neural nets,* pp. 267-285, 1982.

[9] e. a. Y. LeCun, "Handwritten digit recognition with a back-propagation network," *in Advances in neural information processing systems,* pp. 396-404, 1990.

[10] e. a. Y. LeCun, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, pp. 2278-2324, 1998.

[11] R. Hecht-Nielsen, "Theory of the backpropagation neural network," *Neural networks for perception,* no. Elsevier, pp. 65-93, 1992.

[12] S. C. ,. Y. L. Pierre Sermanet, *Convolutional Neural Networks Applied to House Numbers Digit Classification,* New York, 2012.

[13] J. P. L. Y-Lan Boureau, *A Theoretical Analysis of Feature Pooling in Visual Recognition,* New York.