APPLICATION OF DEEP NEURAL NETWORK IN EDGE DETECTION

Srikanta Patnaik[1], Dr.K.Parvathi[2], Dr.S.S. Nayak[3]

[1]PhD Scholar, CUTM, Parlakhemudi, Odisha, India
[2]Professor, KIIT University, Bhubaneswar, Odisha, India
[3]Professor, CUTM, Parlakhemudi, Odisha, India.

Abstract: Edge detection has a significant role in Image processing and computer vision. Conventional techniques are not able to get the clarity required for computer vision type of applications. Deep Neural networks based on convolution offer substantial clarity over the known detectors.

Key words: *DNN, ANN, NN, Edge detection, Robust, Sobel, Canny.*

I. Introduction

Neural networks were named after biological neural networks [1] as there have been developed to mimic various neural feature of human brain. Early degrees of NN or ANN noticed the models of threshold logic unit of Warren Mc Culloch and Walter Pitts in 1943 and the understanding via Frame Rosenblatt in 1957 [2]. The human brain has 10^{11} neurons which operate parallelly. Artificial neuron is a mathematical function which is implemented on computers in serial manner. Research and design aspects of NN are focused more on developments in mathematics and engineering instead of biology [3].

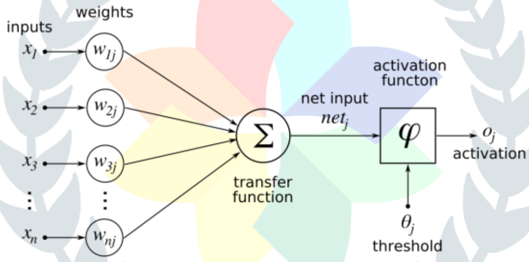

Figure 1 an Artificial Neuron.

A McCulloch-Pitts-based artificial neuron model has been illustrated as figure1 [4] .The neuron receives m input parameters having m weight parameters (like synaptic weights in BNN). The weight parameters are associated with a bias factor Which has the corresponding dummy input with a specified value. These weights and the inputs are normally merged in a linear fashion and then summed. This sum is given to an activation function which gives an output [5].

$$y_k = \varphi(s_k) = \varphi\left(\sum_{j=0}^{m} w_{kj}\, x_j\right) \tag{1}$$

With suitable value of the weight, neurons can be properly trained for required output with every concern input. In a feed-forward fully connected multilayer neural networks Write down anything you want. To paraphrase it, click the Quill It button on the right.

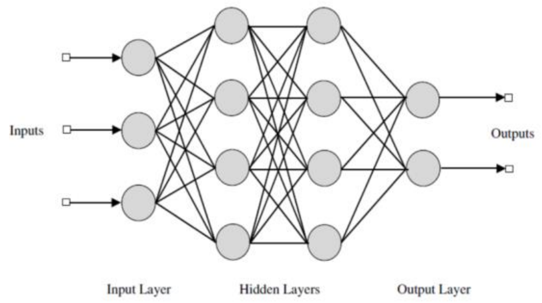

Figure 2. A feed-forward fully connected multilayer neural network

Typically there are 3 layers: input, hidden and output. The input stage process data without modification to the hidden layer which has most of the computation. The work of output layer is to convert the hidden layer activation to an output like classification. A feed-forward multi layer network with one or more hidden layer that work as Universal approximation suitable for computation of any function [7]. A convolutional network has limited connection and use parameter sharing compared to a fully connected networks. Recurrent networks can have special advantages of speech and text recognition [8].

II .Back Propagation Algorithm

The choice of loads of neurons is required for causing the system to figure out how to estimated target yields from known information sources. Systematic techniques for fathoming the neuron loads are unwieldy. The back engendering calculation gives a basic of successful answer for unraveling the loads iteratively [9]. The early form utilized inclination plunge as improvement strategy. This can move toward becoming tedious and the worldwide least isn't ensured. With legitimate setup (like hyper parameters) this strategy can give great outcomes in the principal stage, an information vector is proliferated forward through the neural system. Preceding the loads of system neurons are introduced to certain qualities (little arbitrary qualities) [10]. The got yield of the system is contrasted with the ideal yield known through preparing models by utilizing a misfortune work. The slope of this misfortune capacity is processed to give blunder esteem. While using the mean squared blunder and misfortune work. The yield layer blunder is essentially the contrast among present and wanted yield [11].

The error values at the output are back propagated through network to compute the hidden layer parameters [12] . The chain rule derivatives can be utilised to solve the hidden layer activities associated to loss function gradients. Lastly, these neuron weights are again updated by calculating the gradient of weights and then subtracting a proportion of previous weights[13]. The learning rate denotes the ratio which can be fixed or dynamic. To achive the convergence the phases in the algorithm are executed continuously with different inputs[14].

III. Methodology

Types of learning

The above description pretends to learning by lines. After each input this calculates the weight updates. This has demerit of “Zig-Zagging” in which the estimate of a single data point gradiant continuously changes direction and it does not reach minimum point directly. In full batch learning the weight updates are computed for the complete data set; the computation cost is heavy. A compromise version is mini batch learning which only take a small portion of the training set for each update [15].

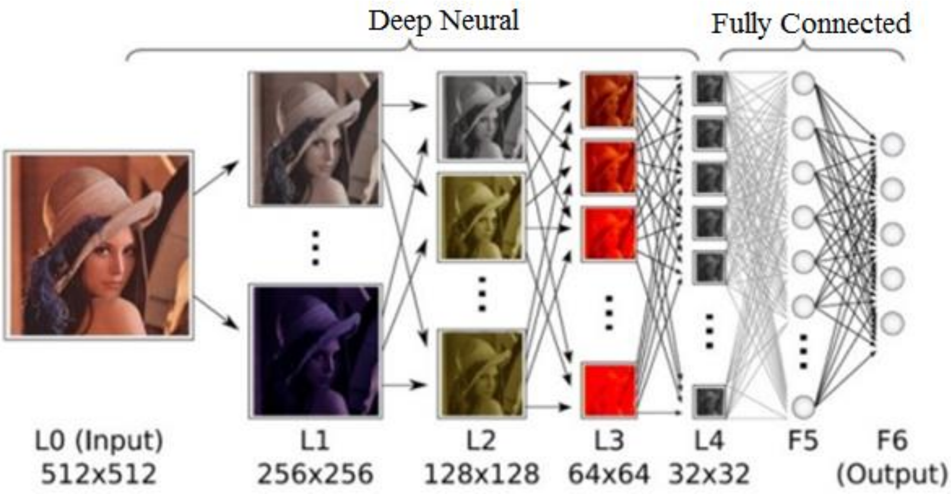

Figure 3. Deep Neural Networks structure.

Activation function

There are several disadvantages of perceptrons and other available linear systems. These could not solve problems which were not linearly separable like XOR problem. Often, linear systems can be used to solve such problems using handcrafted feature detectors. Just adding layers does not improve the characteristics as linear neurons of the network still remain linear in spite of multiple layers in figure 4.

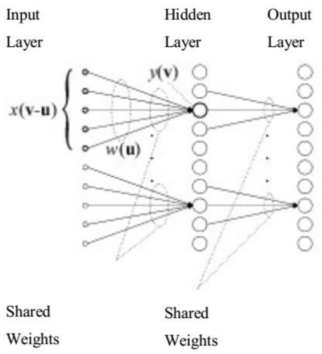

Figure 4 Shared Weight Neural Network Edit image

In figure 3 deep neural networks have multiple hidden layers structure. However a non linear network of light and effective weight can be created by rectified linear units. This rectified linear function generates output using a ramp function such as

$$\varphi(s) = \max(0, s) \tag{2}$$

Such functions can be easily computed and differentiated (suitable for back propagation). A demerit of such function is that it is not differentiable at zero. In spite of this drawback, ReLu have replaced sigmoid activation function which have smooth derivatives but have the problems of gradient saturation and sluggish computations.

The soft max activation can be used for solving the multi class classification problem of the output layer of a network.

$$\varphi(s) = \frac{\exp s_k}{\sum_{k=1}^{K} \exp s_k} \qquad (3)$$

The soft max function has a vector of arbitrary large value and outputs a vector which ranges between 0 to 1 and sums up to 1. The value of soft max can be utilised as class probabilities. It may be noted that main problem of deep learning is the curse of dimensionality. As the Due number of variables increases it also increases the number of configurations exponentially.

IV. Edge Detection

As expressed in figure [5], the brightness of image can be changed properly by the points of digital image in the process of edge detection . The basic requirement is to find discontinues in depth and surface orientation as well as changes in material properties and illumination of scene. In remote sensing images corruption due to noise is an important problem mathematical morphological based techniques of edge detection analyse and process geometrical structures using set theory [16].

Figure 5 Edge detection Patterns

Originally develop for binary images, these are extended to grey scale functions and images for suppressing noise. Images can be enhanced and edges can be detected in greater detail than Sobel or Prewitt or Canny or Laplacian type of edge detectors. Gradient for a function of f(x,y) the gradient of f at coordinates (x,y) is given by

$$\nabla f = \text{grad} f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix} \qquad (4)$$

The important property of this vector is that it points in direction of the greatest rate of change of f at location (x,y). The magnitude and length of vector ∇f is $M(x,y) = \sqrt{g_{x^2} + g_{y^2}}$

This gives the value of rate of change in the direction of gradient vector it is an image of same size as the original created when x and y are allowed to vary overall pixel locations in f. In equation 4 and 5. This image is called ‘Gradient image or gradient”. Approximately this is the sum of $|g_x|$ and $|g_y|$. Most popular masks used to approximate the gradient or isotropic or multiples of 90^0. Assuming a 3*3 having centre point z_5. So that $g_x = (z_8 - z_5)$ and $g_y = (z_6 - z_5)$. Using cross differences (Roberts) $g_x = (z_9 - z_5)$ and $g_y = (z_8 - z_6)$. Using the summation of approximation, M(x,y) = $|z_9 - z_5| + |z_8 - z_6|$ .The corresponding masks are called Robert’s cross gradient operators. Sobel operator’s use

$$g_x = \partial f/\partial x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \partial f/\partial y = (z_3 + 2z_6 - z_9) - (z_1 + 2z_4 + z_9)$$

$$M(x,y) = |g_x| + |g_y|$$

(5)

1	0
0	1

0	-1
1	0

Roberts cross gradient pattern

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

V. Simulation Results

Deep neural networks used in this paper are of convolutional type the improved clarity with respect to standard detectors like Sobel, Prewitt, Canny, Robust are evident from the figures. The base activation function used these of sigmoid type it can be further improved by using ReLu function or Softmax function which represented in figure 6 and 7.

Original Image

epochs =200,

epochs =300,

epochs =400

epochs =600

epochs =800

epochs =2000 epochs =10000 epochs =100000

Figure 6. Different network statues

Simulation results on "Lena" image Simulation results on "cameraman" image

Figure 7 Comparison of different techniques Vs proposed technique

VI. Conclusion

Deeper Networks with more hidden layers are likely to improve the clarity further. Use of activation functions such as Softmax can be used for further improvements.

References

1. Das, Dr RP, and L. Sreedhar. Neural Networks and Fuzzy Logic. SK Kataria and Sons, 2012.
2. McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics 5, no. 4 (1943): 115-133.
3. Cochocki, A., and Rolf Unbehauen. Neural networks for optimization and signal processing. John Wiley & Sons, Inc., 1993.
4. Zhang, Ling, and Bo Zhang. "A geometrical representation of McCulloch-Pitts neural model and its applications." IEEE Transactions on Neural Networks 10, no. 4 (1999): 925-929.
5. Takefuji, Y., and K. C. Lee. "An artificial hysteresis binary neuron: A model suppressing the oscillatory behaviors of neural dynamics." Biological Cybernetics 64, no. 5 (1991): 353-356.

6. A.Sampath Dakshina Murthy , T.Syamala , S.Koteswara Rao, Dr. Rudra Pratap Das."Real Time Differential Global Positioning System Using Fuzzy Logic Extended Kalman Filter." Journal of Advanced Research in Dynamical and Control Systems Vol. 9. Sp– 14 / 2017.
7. Srikant Patnaik, K.Parvathi and S.K.Mandal." Efficient and Simplified Adaptive Neuro Fuzzy Based Edge Detection for Digital Image Processing Using Anfis for Edge Strength" International Journal of Pure and Applied Mathematics ,Volume 117 No. 15 2017, 253-261.
8. Huang, Guang-Bin, Lei Chen, and Chee Kheong Siew. "Universal approximation using incremental constructive feedforward networks with random hidden nodes." IEEE Trans. Neural Networks 17, no. 4 (2006): 879-892.
9. Leshno, Moshe, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function." Neural networks 6, no. 6 (1993): 861-867.
10. Dr. Rudra Pratap Das ,G. Roja , A.Sampath Dakshina Murthy, Dr.S.Koteswarao Rao."Global Positioning System Object Tracking by Applying Fuzzy Logic Non Linear Techniques". Journal of Advanced Research in Dynamical and Control Systems Vol. 9. Sp– 14 / 2017.
11. Patnaik, Srikant, K. Parvathi, and S. K. Mandal. "Development Of Simple Edge Detection Technique Using FIS." (2017): 323-332.
12. Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How transferable are features in deep neural networks?." In Advances in neural information processing systems, pp. 3320-3328. 2014.
13. Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." arXiv preprint arXiv:1510.00149 (2015).
14. Seide, Frank, Gang Li, and Dong Yu. "Conversational speech transcription using context-dependent deep neural networks." In Twelfth annual conference of the international speech communication association. 2011.
15. Kim, Yoon. "Convolution neural networks for sentence classification." arXiv preprint arXiv: 1408.5882 (2014).
16. Toshev, Alexander, and Christian Szegedy. "Deeppose: Human pose estimation via deep neural networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1653-1660. 2014.