International Journal of Computer Science Trends and Technology (IJCST) – Volume 6 Issue 4, Jul - Aug 2018

RESEARCH ARTICLE OPEN ACCESS

Error Rectification Using Backpropagation

P. T. V.Lakshmi
Assistant Professor
NadarSaraswathi College of Arts and Science
Theni – Tamil Nadu
India

ABSTRACT

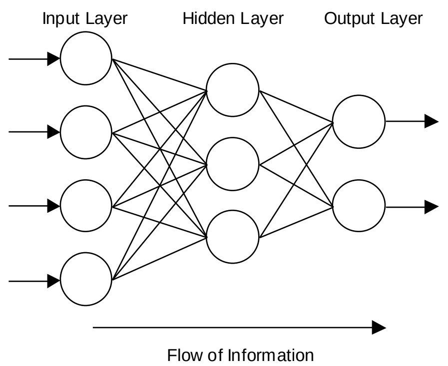
Artificial neural networks are a simulation abstract of nervous system, which contains collection of neurons. With the emergence of neural networks design, modern methods of controlling nonlinear system have been more accurate and convenient. On the path I am interested in giving backpropagation learning for the starters with an example of actual numbers. This post is my attempt to explain how it works with a concrete example that folks can compare their own calculations in order to ensure they understand backpropagation correctly.The goal of backpropagation is to optimize the weights so that the neural network can learn how to correctly map arbitrary inputs to outputs. Error will be rectified according to the rule through the activation function, forward method and backward method.

Keywords:- Artificial Neural Networks, Nonlinear System, Backpropagation, Activation function**.**

I. INTRODUCTION

Back propagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Standard back propagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. The term back propagation refers to the manner in which the gradient is computed for nonlinear multilayer network.

Neural networks are typically arranged in layers. Each layer in a layered network is an array of processing elements or neurons. Information flows through each element in an input-output manner. In other words, each element receives an input signal, manipulates it and forwards an output signal to the other connected elements in the adjacent layer. A common example of such a network is *the Multilayer Perceptron* (MLP). MLP networks normally have three layers of processing elements with only one hidden layer, but there is no restriction on the number of hidden layers. The only task of the input layer is to receive the external stimuli and to propagate it to the next layer. The hidden layer receives the weighted sum of incoming signals sent by the input units (Eq. 1), and processes it by means of an activation function. The activation functions most commonly used are the saturation, sigmoid (Eq. 4) and hyperbolic tangent(Eq.5) functions. The hidden units in turn send an output signal towards the neurons in the next layer. This adjacent layer could be either another hidden layer of arranged processing elements or the output layer. The units in the output layer receive the weighted sum of incoming signals and process it using an activation function. Information is propagated *forwards* until the network produces an output.



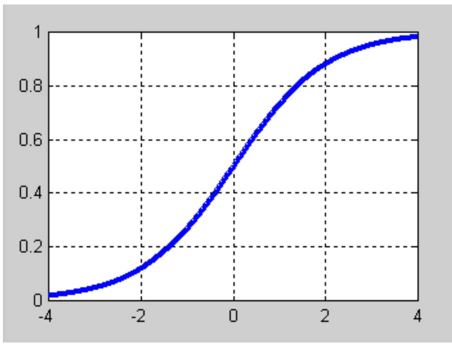
$$SUM = \sum_{i=1}^{n} x_i w_i \qquad [1]$$

$$y = f\left(\sum_{i=0}^{n} x_i w_i\right) \qquad [2]$$

$$f(x) = \begin{cases} 1\ if \sum_{i=1}^{n} x_i w_i > 0 \\ 0\ if \sum_{i=1}^{n} x_i w_i \leq 0 \end{cases} \qquad [3]$$

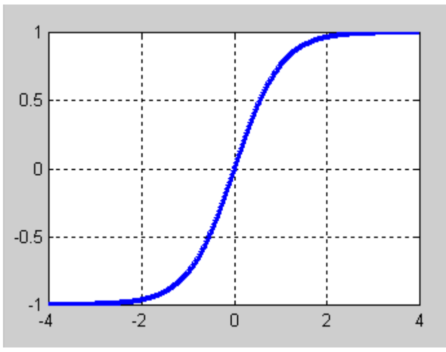
II. SIGMOID FUNCTION

$$f(x) = \frac{1}{1+e^{-x}} \qquad [4]$$



Sigmoid function

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

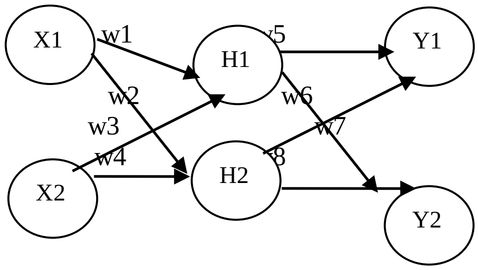


Hyperbolic tangent function

III. ERROR RECTIFICATION USING BACK PROPAGATION

Consider a problem with the given weights.As compared with the target value it isNot satisfied so it is going to be back propagated to attain the target value.

Let us consider this problem as,



H1=x1*w1+x2*w2+b1

Activation function sigmoid = $1/1+e^{-x}$

[5] Let us apply the values for input us

x1=0.05 x2=0.10

Weights as

w1=0.15w5=0.40

w2=0.20w6=0.45

w3=0.25 w7=0.50

w4=0.30 w8=0.55

Bias value as

b1=0.35,b2=0.60

Target value t1=0.01,t2=0.99

Forward pass:

H1=x1*w1+x2*w2+b1

H1=0.05*0.15+0.10*0.20+0.35

H1=0.3775

Activation function:

$H1=1/1+e^{-h1}$

$=1/e^{-0.3775}$

=0.593269992

Similarly calculate the value of H2,now the value is,

H2=0.596884378

After getting the value of h1,h2 now we have to calculate y1 and y2 values,

Y1=outH1*w5+outH2*w6+b2

=0.593269992*0.59+0.59688437*0.46+0.6

=1.1059

$Y1=1/1+e^{-y1}$

$=1/1+e^{-1.1059}$

Y1=0.75136507

Similarly calculate value of Y2 ,

Y2=0.77292

But the target values are,

T1=0.01 T2=0.99

It does not match the target value,so we have to calculate the total error.

Calculate total error,

$E_{total}=\Sigma\, 1/2(T-O)^2$

$=\underbrace{½(T1-outY1)^2}_{E1}+\underbrace{½(T1-outY2)^2}_{E2}$

$=1/2(0.01-0.75136507)^2+1/2(0.99-0.772)^2$

=0.274811083+0.023560026

E_{total}=0.298371109

As we have the total error,we have to backpropagate to update the values,here we are going to apply partial differentiation to update the value of w5,

So Error at w5 is

$$W5 = \frac{\partial E_{total}}{\partial w5}$$

$$\frac{\partial E_{total}}{\partial w5} = \frac{\partial E_{total}}{\partial outy1} \text{ x } \frac{\partial outy1}{\partial y1} \text{ x } \frac{\partial y1}{\partial w5}$$

E_{total} = ½(T1-outY1)2+½(T1-outY2)2

$$\frac{\partial E_{total}}{\partial outy1} = 2* ½(T1-outY1)^{2-1*}-1+0$$

=-(T1-outY1)

=(-(0,01-0.75136507)

=0.74136507

$outY1=1/1+e^{-y1}$

$$\frac{\partial outy1}{\partial y1} = outY1(1-outY1)=0.75136507(1-0.75136507)$$

=0.186815602

$$\frac{\partial y1}{\partial w5} =1*outH1*w5^{(1-1)}+0+0$$

=outH1=0.593269992

Finally substituting values for,

$$\frac{\partial E_{total}}{\partial w5}= \frac{\partial E_{total}}{\partial outy1} \text{ x } \frac{\partial outy1}{\partial y1} \text{ x } \frac{\partial y1}{\partial w5}$$

=0.74136507 * 0.186815602 * 0.593269992

=0.082167041 ⟶ Change in w5

Updating w5,

$$W5 =w5 –\eta* \frac{\partial Etotal}{\partial w5}$$

η is eta which is called as a Learning rate.It may vary from 0-1.

Now we are going to substitute the value of η as 0.5

=0.4 – 0.5 * 0.08216

W5 = 0.3589

In the same way,

W6 = 0.40866618

W7 = 0.511301270

W8 = 0.061370121

Then substitute the values of updated weight in the particular appropriate places.Similarly in the same way update the values of w1,w2,w3,w4.This process of updating and doing backward process is called Backpropagation.

IV. CONCLUSION

Thus the Artificial neural network involves two passes. In the forward pass the input signals propagate from the network input to output.In the reverse pass,the calculated error signals propagate backwards through the network where they are used to adjust the weights.The calculation of output is carried out layer by layer in the forward direction.The output of one layer in weighted manner will be the input of next layer. In the reverse pass,the weights of the output neuron layer are adjusted first since the target value of each output neuron is availableto guide the adjustment of associated weights.

REFERENCE

[1] Andrian, Y. & Putra, H.P. 2014. Analysis of Addition of Momentum to Prediction of Rainfall in Medan Using Backpropagation Neural Network.Seminar National Informatika 2014: 165 – 172.

[2] Brian, T. 2016.Analysis Learning Rates On Backpropagation Algorithm For Classification of Diabetes.JurnalIlmiahEdutic3(1):21 – 27.

[3] Dhanewara, G. &Moertini, V. 2004. Artificial Neural Networks Back Propagation For Integral Data Classification9(3):117 – 131.

[4] Hamid N.A. &Nawi, N.M(2011).The Effect of Adative Gain And Adative Momentum in Improving Training Time Of Gradient Descent BackpropagationAlgorith on Classification Problems. Proceedings of the Internasional Conference on Advanced Sciense, Engineering And Information Technology 2011.ISASEIT:pp. 178 – 184

[5] Hamid N.A, Nawi, N.M, Ghazali, R. &Saleh, M N.M. (2011) Accelerating Learning Performance of Backpropagation Algorithm by Using Adaptive Gain Together with Adaptive 0 Momentum and Adaptive Learning Rate on Classification Problems. International Journal of Software Engineering and Application.8(4) : 31 – 43.

[6] Huang, D., Wu, Z. 2017. Forecasting Out Patient Visits Using Empirical Mode Decomposition Coupled With Backpropagation Artificial Neural Networks Optimized By Particle Swarm Optimization. Journal Plos One 12(2): 1-18.

[7] Shanmuganathan, S. &Samarasinghe, S. (Editor). 2016. Studies in Computational Intelligence: Artificial Neural Network Modelling. Springer: Switzerland

[8] Sitanggang, I.S., Hermadi,I., Edward. 2007. Implementation of Neural Networks in Predicting the Understanding Level of Students Subject. JurnalIlmiahIlmukomputer. 5(2): 124-143

[9] Sumijan, Windarto, P.A., Muhammad, A. &Budiharjo. 2016. Implementation of Neural Networks in Predicting the Understanding Level of Students Subject. International Journal of Software Engineering and Its Applications10(10): 189-204.

[10] Sumijan, Windarto, P.A., Muhammad, A. &Budiharjo. 2016. Implementation of Neural Networks in Predicting the Understanding Level of Students Subject. International Journal of Software Engineering and Its Applications10(10): 189-204.