Name: Zenish Yopinbhai Patel

Student number: 7898869

Research Project : COMP 3190

Training multilayer networks using backpropagation algorithm

Abstract: It is indeed true that backpropagation algorithm is leading the race of being the most efficient and fast algorithm to train the Multilayered neural networks on which the modern technology like image and speech recognition are based on. Backpropagation aims to reduce the error to minimum and provide us with the best possible output regardless of the quality of input provided and to achieve that goal it works on the most basic block which make the whole neural network. However, there are some limitations for a normal backpropagation algorithm associated with the data and efficiency that I will be discussing in the paper.

Index:

1. Introduction

2. What are Multilayer Neural networks?

a) How do Multilayer Neural networks work

3. Gradient Descent

4. Backpropagation agorithm

a) How it works

b) Pros and Cons of the backpropagation algorithm

c) Improvement that could be made on it

5. Conclusion

1) Introduction:-

Artificial Intelligence is one of the most popular field in modern computer science. And it is constantly evolving as the time passes and it is expanding at a great pace. When we say Artificial intelligence we mean that the ability for a non living thing to think, which is correct upto some extent but it is much more than that. For me, it is more like an ability to solve complex problems and learn from the failures encountered while solving that problem and avoid it when faced again. And the thing on which my definition of artificial intelligence is based on is Neural networks and what makes the back bone of Neural networks are the training algorithms like backpropagation algorithm, adaptive step alogrithm, second step algorithms, etc. Here in this paper I will explain multilayer neural networks by explaining the neurons and their functioning. Also, how linear regression and gradient descent train these neurons which comnined to backpropagation improves the multilayer neural network drastically. And at last what are the major benefits and problem encountered by improving the network using this algorithm and how we could tackle these problems.

2) MultiLayer Neural Networks :-

Neural networks are like a giant artificial web. As the name suggests neural networks resembles very much to the biological neural networks of human brain. Biological neural networks are made of small units known as neurons similarly, artificial NN too have a similar unit. There are different models which define it, one is McCulloch-Pitts model that refers the neuron as Threshold Logic unit(TLU) which takes binary values as input. And the another one is Perceptron model by Frank Rosenblatt where the base is a Perceptron also known as Linear Threshold Unit (LTU). This web is made up of such neurons which are connected via links. These links are called edges and these edges have some weight which are used to make calculations while carrying out some operations. We can think of a neural network as a wall with different layer each layer as a set of neurons connected with edges. The layer on one side as an input layer from where the network is feed input and the layer on other side as an output layer which provides output with the middle layers as hidden layers which is like a black box. This kind of model is know an a Feedforward neural network. For each hidden layer the previous layer works as the input layer and the next layer as output layer.

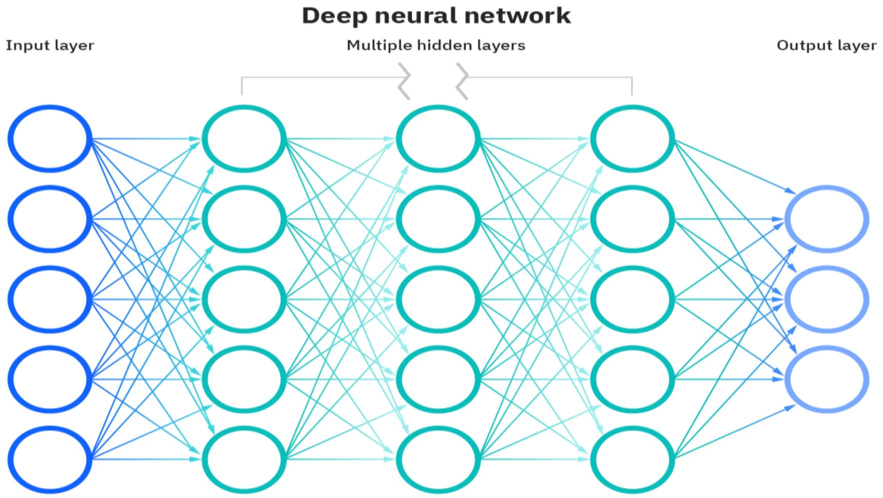


Figure 1Multilayred Neural Network (Education, https://www.ibm.com/cloud/learn/neural-networks)

a) How do Multilayer Neural networks work:

Every neuron has a threshold (capacity). And a neuron is feed with some data from the edge with some weight using which uses it to predict a value of some unknown variable using linear regression. Linear regression:- “This system takes a vector x ϵ R^n as input and predicts the value of a scaler y ϵ R as it’s output ”

We can define it as : $\hat{y} = \omega^T$x.(Page 107 Deep Learning)

Where ω ϵ R^n is a vector of Parameters. We can also assume itas a set of weights.“Think of each individual node as its own linear regression model, composed of input data, weights, a bias (or threshold), and an output.”

“We get the formula: ∑wixi + bias = w1x1 + w2x2 + w3x3 + bias

output = f(x) = 1 if ∑w1x1 + b>= 0; 0 if ∑w1x1 + b < 0”. (Education, https://www.ibm.com/cloud/learn/neural-networks) In simple words every input is multiplied to the each assigned weight and added passed through an activation function after which we receive some output which if exceeds the value of treshold then the data is transferred to the next node (Rojas 151-155). It is very similar to the photo electric effect just that here the data is released from a node inplace of the electron. This is helpful if the network is feedforward where the flow of data is one directional. However, for modern applications it is important to train the networks to be able to trace in back direction bringing the backpropgation in the scene. Backpropagation minimizes the error in the weight space giving us accurate output and to minimize the error it uses the method of gradient descent. (Rojas)

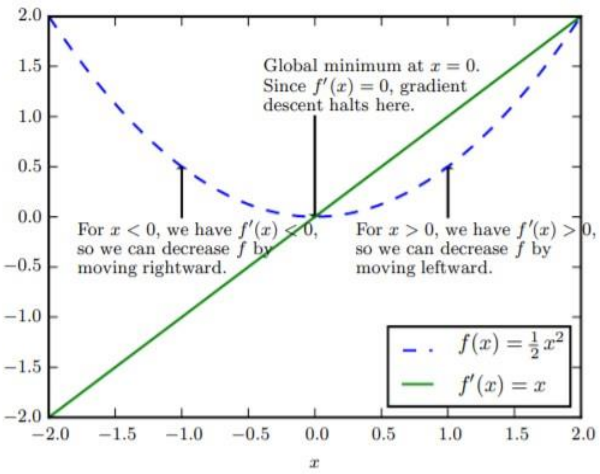
3) Gradient Descent:

Our motive is to improve the efficiency of the neural network and for that we use gradient descent. The main function in the gradient descent which is actually behind minimizing the error is known as the cost function/loss function/error function. It trys to figure out (i.e learn) the best possible parameters so that the error gets as smalll as possible (Ian Goodfellow 82). It is similar to the linear regression but as linear regression works for the linear functions, gradient descent is for the convex function. (Education) We can compare it with the hill climb as it looks opposite to it (Ian Goodfellow).Here we have to find the best local minima where the error is minimum compared to the error for other points for the function. Explaining it mathamatically. (Ian Goodfellow 82-93)

Let $y = f(x)$ be a function where x and y are both Real numbers

$(dy/dx) = f'(x)$ it gives the slope of x at $f(x)$

Thus it shows how to measure a small change in the input in order to get the change in the output i.e $f(x + \epsilon) \approx f(x) +\epsilon f'(x)$ (Ian Goodfellow 83). As we approach to $f'(x)$ to 0 the change in error decreases. And it becomes minimum at f ´(x) = 0 which we call as critical point or a local minima . Here the error at x is less compare to the error at points more and less than ‘x (Ian Goodfellow)’. This method of optimization of error is called gradient descent. The points where gradient descent seems to fail to get us the best possible minimum error with good efficiency is when the problems are complex and non convex and also at some saddle points. (Ian Goodfellow)



Gradient descent 1 (Ian Goodfellow 83)

4) Backpropagation algorithm: As the size of layers increases it becomes difficult for the machine to compute and find the correct combination of weights as the number of nodes and parameters increases. And thus to counter that we use backpropagation algorithm. It uses the method of gradient descent to find the minimum of the error function. Here to note that the error function must be continuious and differentiable. (Education, https://www.ibm.com/cloud/learn/gradient-descent)

“Sigmoid function is a very popular function for backpropagation networks a real function sc : IR → (0, 1) defined by the expression.” (Rojas 151-154)

$$S_c(x)=\frac{1}{1+e^{-cx}}$$

C is a constant , 1/C is temperature parameter. As C approaches to infinity the shape of sigmoid function gets closer to step function.

$$\frac{d}{dx}s(x) = \frac{e^{-x}}{(1+e^{-x})^2} = s(x)(1-s(x))$$

Its output range is from 0 to 1. (Rojas 151-155)

If the sigmoidal function is given some weights (W1,…,Wn) and a bias -theta, a sigmoidal unit computes for the input X1,……,Xn can be given by the equation

$$\frac{1}{1+\exp\,(sum\ n..i=1\ WiXi-theta)}$$

Another approach that could be used is by connecting the graph labeling problem with the gradient calculations. (Rojas)

a) How backpropagation algorithm works:

For any given N inputs and M outputs consisting of many hidden layers given a set to train that network { (x_1,t_1) , ... , (x_p,t_p)}, the weights of the edges are selected randomly. And for ever input x_i we get an output o_i for every target t_i for i = 1,2,3,4,….p. (Rojas 155)

The error function of the network is $E = \frac{1}{2}\sum_{i=1}^{p}\left|\left|o_i - t_i\right|\right|^2$.The network is initialized with randomly chosen weights and are corrected using the gradient of error function. (Rojas 156)

We can say E= $E_1+E_2+E_3+……….+E_i$.

Using it we can calciulate the total error of network. To minimize the error we have to minimize the error of individual node. Which could be done only by altering the weights. (Rojas 157)

Therefore :- $\nabla E = \left(\frac{\partial E}{\partial w1}, \frac{\partial E}{\partial w2}, \ldots \ldots \ldots \ldots \ldots \ldots, \frac{\partial E}{wl}\right)$. (Rojas)

$\Delta wi = -\gamma \frac{\partial E}{\partial wi}$ for i = 1, . . ., l. (Rojas)

Here, γ is called the learning constant.

With this we can adjust the weights of the network iteratively and reduce the error to 0.To reach there we first compute the vectors provided to the feedforward network which are sorted and stored at each unit (Node). Then we propagate in the backward direction from the output layer to some units of hidden layers . (Rojas)

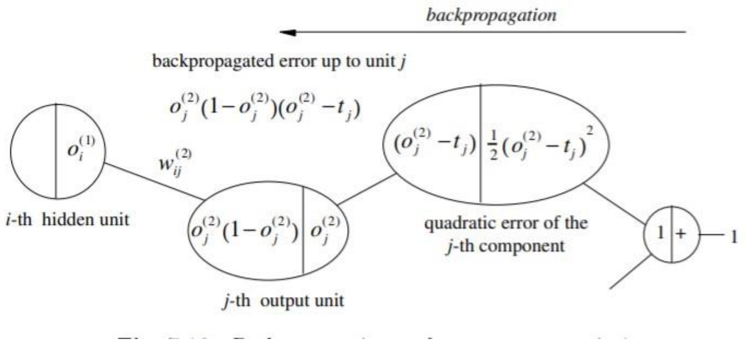

backpropagation
backpropagated error up to unit j
$o_j^{(2)}(1-o_j^{(2)})(o_j^{(2)}-t_j)$
$o_i^{(1)}$
$w_{ij}^{(2)}$
i-th hidden unit
$o_j^{(2)}(1-o_j^{(2)})$
$o_j^{(2)}$
j-th output unit
$(o_j^{(2)}-t_j)$
$\frac{1}{2}(o_j^{(2)}-t_j)^2$
quadratic error of the j-th component
1
+
1

Figure 2BackProp for different layers (Rojas 167)

Therefore, for this path the backpropagated error is

$\delta_j^{(2)} = o_j^{(2)}(1 - o_j^{(2)})(o_j^{(2)} - t_j)$ (Rojas)

and the partial derivative will be

$$\frac{\partial E}{\partial w_{ij}^{(2)}} = [o_j^{(2)}(1 - o_j^{(2)})(o_j^{(2)} - t_j)]o_i^{(1)} = \delta_j^{(2)} o_i^{(1)}$$

Figure 3equation (Rojas 168)

After this the same thing goes on iterarartively for the whole hidden layer and this is done with the knowlegde of every imaginable backward pattern i.e:- for different values of j (from the image)n or it can be said that for different sets of layers in the network. (Rojas 166-173)And at the end after the whole cycle of backpropagation is completed the weights are updated in the negative gradient descent. It must not be done in the middle of the cycle as it will make no sense for any calculated steps as the weights get changed for the next step and the algorithm might end up loosing the track and fails very badly. (Rojas 166-173)

b) **Pros and Cons of the backpropagation algorithm:**

Modern technology such image and speech recognition relays heavily on backpropagation algorithm it. And the perfect example of it is the image quality improvement even if the quality of image captured is very poor it can be improved heavily using it in no time which shows that it is very fast. Also this algorithm has no limitations on the length of the network or the number of inputs. However, sometimes it performs poorly given the data is in very very bad condition. (Rojas)

c) **Improvement that could be made on it:**

To enhance the efficiency and speed of the network I suggest to perform some sorting and providing the sorted dataas input to the algorithm. Doing this the time taken by the algorithm gets reduced. However, in some cases the depending on the size of the sum of time taken by the algorithm and time taken to sort the data could become much more than the time taken by algorithm to process unsorted input data.

5) Conclusion:

There are many algorithms that could be used to train neural networks but the most widely used is backpropagation algorithm and this is because of how it functions , from the very bottom or we can say the core of the neural network. What this algorithm does is checks and alters the weights of the neurons which are the basic building blocks of the neural network inorder to minimize the error using gradient descent and linear regression. This algorithm looks perfect but still it has some limitations which are yet to be solved.

Bibliography

Education, IBM Cloud. *https://www.ibm.com/cloud/learn/gradient-descent*. 27 October 2020.

—. *https://www.ibm.com/cloud/learn/neural-networks*. 17 August 2020.

Ian Goodfellow, Yoshua Bengio,Aaron Courville. *Deep Learning*. MIT Press, 2016.

Rojas, R. *Neural Networks: A Systematic Introduction*. Berlin: Springer, 1955.