ISSN 2229-5518

Training and Analysis of a Neural Network Model Algorithm

Prof Gouri Patil

Abstract—An algorithm is a set of instruction pattern given in an analytical process of any program/function-ale to achieve desired results. It is a model-programmed action leading to a desired reaction. A neural network is a self-learning mining model algorithm, which aligns/ learns relative to the logic applied in initiation of primary codes of network. Neural network models are the most suitable models in any management system be it business forecast or weather forecast. The paper emphasizes not only on designing, functioning of neural network models but also on the prediction errors in the network associated at every step in the design and function-ale process.

Index Terms— Input , Neural Network, Training, Weights

1 Neural Network – A Mining model

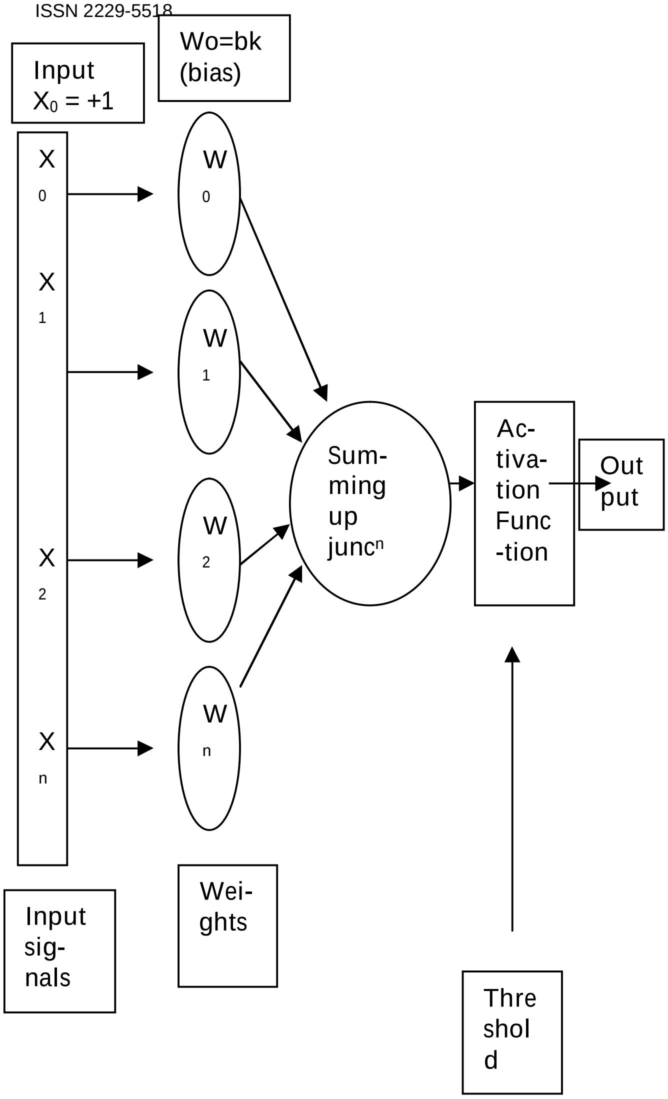
A Neural Network algorithm could contain multiple networks, depending on the number of columns used for both input and prediction, or that are used only for prediction. The number of networks a single mining model contains depends on the number of parameters connected by the input columns and predictable columns the mining model uses. Neural network function-ale is a mimic of human neural interconnections and memory. Human brain or memory comprises of an average of about ten billion neurons functioning in a network synchronization -- and every single neuron is, on average connected to several other neurons (may be around a thousand) indirectly or directly to the central neural mass, the brain. By way of these connections and interconnections, nerves send and receive messages as packets of energy called as impulses. One very important feature of human neurons is that they don't react immediately to the reception of energy called impulse. Instead, they sum their received energies, and then, they send their own quantities of energy to the other associated neurons only when this sum reaches a certain critical threshold the neurons trigger and respond back as signals or packets of energy called impulse. Brain learns by adjusting the number and strength of these connections and gives a desired response or output in terms of polarization or re-polarization of neurons and difference in the energy levels. Neural networks also work on the same principle as our brains work, they respond to a threshold level of input signal often calculated as weights in the network system.

2 Input values in a neural network model

As all inputs and outputs of neural networks are numeric, primary task in designing a neural network is to define a transfer function wide enough to accept input in any numeric range and specific enough to give output in the desired range. The range and limit of inputs should be predefined. Saturation / threshold limit for inputs could then be pre-defined, so that inputs in a way are programmed to give the desired outputs or those inputs or functions incompetent of giving the desired outputs are straight away ruled out in a programmed function-ale network model. Networks are built in a manner that they induce a stepwise logical action called the learning rule. Numeric functional values for the inputs are thus predefined and scaled but for some non-numeric nominal values as male or female, yes or no the network functions are divided as different links in the network and graded on an another chain of Numeric functional values as inputs like (0 and 1) and thus branching the network in two simulating paths working on the same predefined threshold value to decide the final output of the process. The input data for training is the fundamental factor of a neural network, as it gives required information to "discover" the optimal threshold operating point and the result or output of the network. If the output of the network is known and weighed inputs are adjusted to reach the desired output then the network training is called supervised neural network. When the output is not defined or known, as in the case of sale or stock prediction the network is an unsupervised network, which is generally programmed to trigger for all hit value of the input. In case of unsupervised networks for hidden neurons, input values of the network may adjust and interpret differentially in different situations and the network ripens or learns based on the adjustments made in the input values of hidden neural layer.

Prof Gouri. A.Patil is currently working as a faculty for Bioinformatics with Welingkar Institute of Management, Development and Research, Matunga Mumbai -Central, India, contact-+91 9833025213

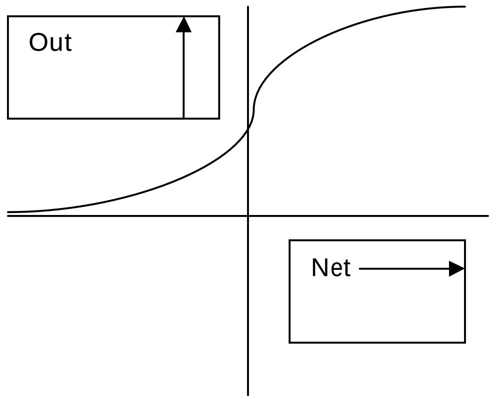
ISSN 2229-5518



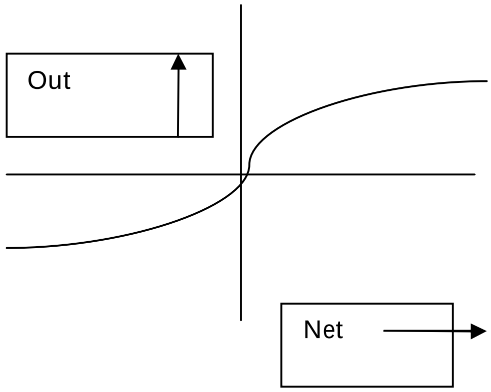
3 FUNCTION-ALE OF A NEURAL NETWORK MODEL

The adjustments made in the input values of the hidden neural layer of an unsurpervised network is nothing but calculations in neural network in terms of adjustments of their weights at the summing up junction and the decisions made at the activation point by defining an activation function value specific for every condition and value. Decisions of the factors include calculations with loops and functions. Initially some time lapse as a function is used to process the data and "winner" input in the time lapse of the model takes all the credit in the first step, where the neuron with the highest value from the calculation fires and takes a value 1, and all other neurons take the value 0.

It is advisable that the sigmoid curve is used as a transfer function because it has the effect of "squashing" the inputs into the range [0,1]. The sigmoid function has the additional benefit of having an extremely simple derivative function for back propagating errors through the neural network.



Out = $1/1+e^{-Net}$
Sigmoid function



Out = tanh (Net/2)
tanh function

Typically the weights in a neural network are initially set to small random values; this represents that the network knows nothing to begin with. As the training process proceeds, these weights would converge to values allowing them to perform a useful computation. Thus it can be said that the neural network begins with knowing nothing and moves on to gain admirable real world application.

Activation function is an important function, which decides the maturation and output of a neural network. Activation functions of the hidden values in the neural network could introduce non linearity and desired matura-

ISSN 2229-5518

tion of a neural network otherwise the network would have been just a plain mathematical algorithm without logical application. For feedback or feed forward learning of network, the activation function should be differentiable as it helps in most of the learning curves for training a neural network like the bounded sigmoid function , the logistic tanh function with positive and negative values or the Gaussian function. Almost all of these nonlinear activation function conditions assure better numerical conditioning and induced learning.
Networks with threshold limits with out activation functions are difficult to train as they are programmed for step wise constant rise or fall of weight. Where as a sigmoid activation functions with threshold limits makes a small change in the input weight produces change in the output and also makes it possible to predict whether the change in the input weight is good or bad.

In the activation function training, Numerical condition is one of the most fundamental and important concepts of the algorithm, it is very important that the activation function of a network algorithm is a predefined numeric condition. Numerical condition affects the speed and accuracy of most numerical algorithms. Numerical condition is especially important in the study of neural networks because ill-conditioning is a common cause of slow and inaccurate results from many network algorithms.

Numerical condition is mostly decided based on condition number of the input value, which for a neural network is the ratio of the largest and smallest eigenvalues. of the Hessian matrix. The eigenvalues of inputs are the squares of the singular values of the primary input and Hessain matrix is the matrix of second-order partial derivatives of the error function with respect to the weights and biases.

4 MODEL NEURAL NETWORK FOR A SAMPLE SIZE OF 20 INDIVIDUALS TO TEST RISK TO DIABETIS.

Type 2 diabetes, is a non-neonatal kind of diabetes that develops in the later stages of life due varied reasons, is far more severe and chronic than type 1 or neonatal diabetis in its destructive metabolic effects on the body. An associated gland called pancreas in our body synthesis hormone called insulin required to metabolize, breakdown and capture glucose for every cell of the body to synthesize energy in the form of energy rich molecule called ATP (Adenosine tri- phosphate). Metabolism of glucose is an indispensable process for the body as this metabolism generates energy for the cell at micro and body at the macro level. There are varied factors response for the onset of this metabolic disorder called diabetes, control of some of these factors could induce synthesis of insulin in correct amount and at required times so that body learns to metabolize efficiently with inefficient or week pancreas. Ageing is one of the prime factor for all metabolic disorders and so it goes for diabetes, our pancreas ages right along with us, and doesn't pump adequate levels of insulin as efficiently as it did when we were younger. Also, as our cells age, they become more resistant to insulin (the carrier of glucose for the cells) as well. Modern sedentary lifestyle is damaging our healths and is a prime responsive factor for growing obesity problems." being obese or overweight is one of the prime factors for increasing level of glucose in the blood causing diabetes". Obesity increases fat cells in the body and fat cells lack insulin or glucose receptors compared to muscle cells, thus increasing fat in the cells is an indirect call to diabetes, exercising and reducing fat in the cells can act as an alert to stay away from this disorder. Eating less of fat and enough fibre and complex carbohydrates compared to simple carbohydrates could contribute to reduce the risk of diabetes. A survey research on 20 individuals for risk to diabetes was conducted using only 4 non-clinical parameters. A supervised network was built and trained to give an output equivalent to associated risk to diabetis with a predefined threshold limit to reach safe non diabetic level with maximum iterations possible at the activation function to reach the out put, which of course was not fixed but was predefined to reach to zero risk to diabetes.

Sample data collected

No	Age	Stress level/ Kind of work	BMI	Obesity
1	30	Stressful labour	30	Over Weight
2	24	Sedentary work	29	Over weight
3	23	Sedentary work	32	Obese
4	35	Sedentary work	40	Highly obese
5	40	Minimum work	33	Obese
6	45	Minimum work	34	Obese
7	43	Minimum work	22	fit
8	50	Minimum work	26	Slightly over weight
9	55	Maximum work	20	fit
10	58	Maximum work	18	fit
11	37	Maximum work	19	fit
12	38	Minimum work	30	Obese
13	27	Maximum work	25	fit
14	29	Stressful labour	17	fit
15	30	Stressful labour	18	fit
16	34	Sedentary work	20	fit
17	38	Sedentary work	22	fit
18	32	Sedentary work	21	fit
19	42	Maximum work	20	fit
20	43	Maximum work	18	Fit

ISSN 2229-5518

Predefined Weights for data

No	Input Parameter	Range	Predefined Weight
1	Age	(0-20)	1
2	Age	(20-30)	2
3	Age	(30-40)	3
4	Age	(40-60)	4
5	Stressful labor	Stressful labor	0
6	Maximum work	Maximum work	1
7	Minimum work	Minimum work	2
8	Sedentary work	Sedentary work	3
9	No work	No work	4
10	BMI	(1 -25)	1
11	BMI	(25-30)	2
12	BMI	(30-35)	3
13	BMI	(35-40)	4
14	Obesity test	Fit	1
15	Obesity test	Slightly over weight	2
16	Obesity test	Over Weight	3
17	Obesity test	Obese	4
18	Obesity test	Highly obese	5

Out put = Threshold limit and risk to diabetes

Condi-tion No	Maximum Threshold limit	Risk to Diabetis
1	5	No risk
2	6	Minimum rirk
3	7 or more	Acute risk

Network training and summing up Junction

No	We-ight Input age n^1	Wei-ght of input stress level n^2	We-ight of input BMI n^3	Weig-ht of input obesity n^4	Summ-ing up Junction $\sum n^n$
1	3	0	2	3	8
2	2	3	2	3	9
3	2	3	3	4	10
4	3	3	4	5	15
5	4	2	3	3	12
6	4	2	3	3	12
7	4	3	1	1	9
8	4	3	2	2	11
9	4	2	1	1	8
10	4	2	1	1	8
11	3	1	1	1	6
12	3	2	3	4	12
13	2	1	1	1	5
14	2	0	1	1	4
15	3	0	1	1	5
16	3	3	1	1	8
17	3	3	1	1	8
18	3	3	1	1	8
19	4	2	1	1	8
20	4	1	1	1	7

Actiavtion function and Iterations to reach the final output from Summing up Junction

No	Summ-ing up Junction $\sum n^n$	Activation function $\sum n^n = n^1 + n^2 + n^3 + n^4 - n^1/n^2/n^3/n^4 - 1/2/3$ such as $\sum n^n \approx 5$	Itera-tions/tr-aining discard	Output-Predict how/when; $\sum n^n \approx 5$
1	8	∑nn = n1+ n2+ n3+ n4 - n1	1	regular exer-cise required
2	9	∑nn = n1+ n2+ n3+ n4 - n1- 3	2	Acute risk to diabetes
3	10	∑nn = n1+ n2+ n3+ n4- n1- 5	4	Diabetic
4	15	∑nn = n1+ n2+ n3+ n4- n1- 7	4	Diabetic
5	12	∑nn = n1+ n2+ n3+ n4- n1-3	4	Acute risk to diabetes
6	12	∑nn = n1+ n2+ n3+ n4- n1-3	4	Acute risk to diabetes
7	9	∑nn = n1+ n2+ n3+ n4- n1	1	No risk
8	11	∑nn = n1+ n2+ n3+ n4- n1-2	3	Acute risk to diabetes
9	8	∑nn = n1+ n2+ n3+ n4- n1+1	1	No risk
10	8	∑nn = n1+ n2+ n3+ n4- n1+1	1	No risk
11	6	∑nn = n1+ n2+ n3+ n4- n1+2	1	No risk
12	12	∑nn = n1+ n2+ n3+ n4- n1-4	4	Diabetic
13	5	∑nn = n1+ n2+ n3+ n4	0	No risk
14	4	∑nn = n1+ n2+ n3+ n4+1	-1 ≈ 0	No risk
15	5	∑nn = n1+ n2+ n3+ n4	0	No risk
16	8	∑nn = n1+ n2+ n3+ n4- n1	1	No risk
17	8	∑nn = n1+ n2+ n3+ n4- n1	1	No risk
18	8	∑nn = n1+ n2+ n3+ n4- n1	1	No risk
19	8	∑nn = n1+ n2+ n3+ n4- n1+1	1	No risk
20	7	∑nn = n1+ n2+ n3+ n4- n1+2	1	No risk

This is a very simple prototype of a network, where every layer of inputs leads to an output. There is no secondary training and learning or firing rule developed in this network, however every network begin with as simple learn-

ISSN 2229-5518

ing single neuron like this and mature to most neural and complex forms. As neural links in the network increases complexity in training and analyzing the network also increases, at the latest with improper links, training and analysis most networks often fails. The next section discusses the common causes for failure of maturing networks and ways to combat these issues with defining open/flexible activation function along with threshold limits to fire an output.

5 SOME COMMON CAUSES AND REMEDIES FOR ILL CONDITIONING IN A NETWORK

1. Low coefficient of variation (standard deviation divided by the mean) of input variables causes great difference in the numeric functions. Such an issue probably could be minimized by subtracting the mean of each input variable from the variable.

2. If the variances among input variables increase measurably, problem can be cured by dividing each input variable by its standard deviation.

3. High correlations among input variables. This problem could be cured by ortho-normalizing the input variables using Gram-Schmidt, SVD, principal components. The orthogonal components in this case should be standardized before taking as input value. Single orthogonal components which fail standardization should be removed from the network as they cannot trained or accepted in the network. Training them in the long process mail cause network failure.

4. If some hidden training units in the network becomes unavoidable and in the long run, the network may become saturated with many hidden units. This issue could be handled by using weight decay or similar regularization methods on the input-to-hidden weights, at the risk of less accuracy in learning discontinuities or steep areas of the target function.

5. Low coefficient of variation for activation values of a hidden unit.-this problem can often be ameliorated by using a hidden unit activation function also with an output range of (-1,1), such as tanh, instead of (0,1) as in the logistic function.

6 REFERENCES

[1] Neural Network design, Martin T Hagan, Howard B Demuth, Mark Beale, 1996 Cengage Learning India Pvt Ltd, ISBN-13 978-81-315-0395-9W.-K. Chen, Linear Networks and Systems.

[2] Ill-Conditioning in Neural Networks, Warren S. Sarle, SAS Institute Inc., Cary, NC, USA Sep 5, 1999Copyright 1999 by Warren S. Sarle, Cary, NC, USA

[3] The Machine Learning Dictionary for COMP9414

[4] Analytics Consulting, Prasanna Parthasarathy, Management Journal Bharathidasan Institute of Management, Trichy , 324-327

[5] Comparison of Artificial Neural Network and Logistic Regression Models for Prediction of Mortality in Head Trauma Based on Initial Clinical Data, Bio-Med Central Feb 2005

[6] R. Roy.. - Furuhashi And Homann, Soft Computing and Industry Recent Applications, ,CiteSeer.IST Seintific Digital Literature Library

[7] Genetic Programming in Data Mining for Drug Discovery - Langdon, Barrett (2004) CiteSeer.IST Seintific Digital Literature Library

[8] Making Indefinite Kernel Learning Practical - Mierswa (2006) (Correct)

[9] Evolutionary Learning with Kernels: A Generic Solution for Large.. - Mierswa (2006) CiteSeer.IST Seintific Digital Literature Library

[10] Combining Decision Trees and Neural Networks for Drug.. - Langdon, Barrett, etal. (2002) CiteSeer.IST Seintific Digital Literature Library

[11] Diversity in Neural Network Ensembles - Gavin Brown To (2003)

[12] Linear Equality Constraints and Homomorphous Mappings in PSO - Christopher Monson CiteSeer.IST Seintific Digital Literature Library