International Journals of Power Electronics Controllers and Converters
Vol. 1: Issue 1
www.journalspub.com

Neural Network Approach for Processing Substation Alarms

Kuldeep S*, Dr.Anitha G S[1]
Assistant Professor, Electrical & Electronics Department, Jain University, Karnataka, India
[1]Associate Professor, Electrical & Electronics Department, RV College of Engineering, Karnataka, India*

Abstract
This paper studies the method of processing of alarms in an electric substation using artificial neural networks (ANN) as a tool. Whenever there is a fault in the power system, or there is a significant change in the network, alarms are issued to the substation operators, with each alarm being associated with a message. This is the driving force for research and experiments regarding innovations of more sophisticated methods of alarm processing. The method used in this study is artificial neural network. The network is a pattern recognition unit. Before using the network for alarm processing, the network is trained using back propagation algorithm to identify all possible combination of alarms received in a power system.

Keywords: *ANN, neural network, alarms, back propagation, expert systems*

****Author for Correspondence** E-mail: kuldeeps10@yahoo.com, 1anithags@rvce.edu.in*

INTRODUCTION

In any power network, reliability of power is a major criterion. Alarms can be defined as a device or signal that alerts the operator of any significant changes in the power system parameters. In a power system, during fault, the system parameters are never the same and there is a continuous change. These changes are made known to the station engineers or operators in the form of audio and visual signals known as alarms [1]. The operators are expected to go through the message/s, analyse the system condition, and take appropriate actions. The diagnosis of the fault is easy, and remedy simple if the number of alarms received is less. But unfortunately, issue of a single or less number of alarms is a rare phenomenon. Usually, multiple alarms are received in the substation at a time. Diagnosis and identification of the cause of the alarms depends on the experience and efficiency of the station engineers. Any delay in diagnosing the fault is not tolerated. Preventing wrong diagnosis of the fault is a reason for use of auto-alarm processors. This method of using auto-alarm systems is alarm processing, and tools such as artificial neural networks (ANNs), Fuzzy systems, Expert Systems (ES) are gaining more attention [2]

PROBLEM IDENTIFICATION

Alarms are issued for a number of reasons, some of them being:

a. Circuit breaker changes
b. Current limit exceeded
c. Voltage, frequency deviation
d. Large deviation in auxiliary parameters like oil temperature, pressure etc.

Whenever multiple alarms are received, the following problems are usually seen:

a. The alarms may not be specific enough
b. There may be too many alarms during a large disturbance
c. Multiplicity of alarms for the same event
d. Alarms may change too fast to be read on the display
e. Missing and false alarms

So when the number of alarms is more, there may be indecision on the part of the operators preventing them from taking the right action at the right time. Handling of a critical activity depends on the operator's ability to solve them under stress conditions, his/her past experience, training and efficiency. Since these vary from person to person, so does the fault diagnosis. This paper studies the possibility of using ANN for the alarm pattern recognition and expert systems (ES) as a knowledge base to decode the single output delivered by the ANN processor.

Data is collected from a 66/11 kV substation. Neuro-expert system, that is, a combination of ANN and ES is used to produce a single output for any set of alarms received. The neural network acts as a pattern recognition unit. It is trained to identify any set of alarms that appear at the input points.

What are alarms?

Alarm is a piece of information about any undesirable, unexpected, unauthorized event in a power system. It gives an idea of the health of the system to the station operators. The need for an alarm processor that clearly identifies the actual cause of the fault and presents to the station operators is essential to prevent the operators relying solely on their experience and knowledge. This is particularly required now as modern electrical systems are highly complex and interconnected.

Audio-visual alarm annunciators similar to the one in Fig 1 were used earlier, and are still being used in some of the smaller substations.

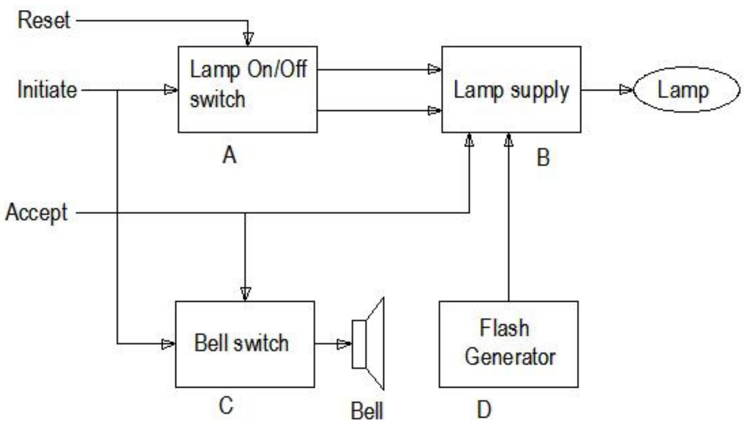

***Fig -1**: Alarm annunciator [3]*

The alarm annunciator provides visual and audible indication of a fault. It consists of four units-A and B are bistable circuits that control the lamp. C is the amplifier for the bell. D is an astable multivibrator that causes the lamp to flash. The alarm is initiated through contacts on the protective relay or on a moving member of the equipment under surveillance, causing the lamp to flash and the bell to ring.

The alarm is acknowledged by pressing the Accept button, silencing the bell, but the lamp continues to flash. On clearance of the fault, the alarm is reset using the Reset button. Now the lamp is extinguished [3].

Types of alarms

Alarms can be broadly classified as Trip alarms and Caution alarms. Trip alarms are the result of opening of circuit breakers, while caution alarms are usually not associated with opening of breakers. They are just meant to warn about something wrong in the system, like deviation in voltage, frequency etc.

Other than these, there are certain types of alarms which need to be taken care so as to be certain whether there is any issue in the power system. They are:

a. Redundant alarms: There are certain events which give rise to multiple alarms. Each such event is called Compound event. Each compound event is characterized by a set of alarms. In these sets of alarms, there may be some alarms which are not necessary to define the said event. These redundant alarms may be useful when they are single, but in most cases, they are unnecessary. Therefore, it is required to identify them and suppress them.
b. Missing alarms: In a power system operation, it is not always that the alarms received will be exactly in accordance with the occurred event. Sometimes, due to non-operation of some of the relays, contactors or other control equipment, alarms which are supposed to be issued will not be issued. These are Missing alarms. In such cases, the operator is faced with lack of information about the event occurred, and the diagnosis may be erroneous, if it is a major event. It is the task of the alarm processor to alert the operator about the alarms which are absent.

c. False alarms: Similarly, there may be cases when alarms may be issued when they are not needed, due to the mal-operation of the relays and contactors. Receipt of false alarms may cause confusion leading to wrong conclusions. It becomes difficult to distinguish between real and false alarms.

The alarms are initiated only if the fault persists for more than 40 milliseconds to prevent initiation on spurious voltage surges. Each time an alarm is received, the control engineer must accept it, which is a waste of time. When the received alarms are multiple, many blinking lights at the same time may confuse the operators, make it difficult to identify the exact event. Therefore, "one event, one alarm" is the solution provided by the alarm processor. The processor used here is Artificial Neural Network (ANN)

ARTIFICIAL NEURAL NETWORK

ANN is a computational model that is developed based on the biological neural networks. An ANN is made up of artificial neurons that are connected with each other. The processing ability of the network is stored in the inter-node connection strengths, which are called weights. These weights are obtained by learning or adapting from a set of training patterns. Typically, an ANN adapts its structure based on the information coming to it. A set of systematic steps called learning rules needs to be followed when developing an ANN. Further, the learning process requires learning data to discover the best operating point of the ANN. They can be used to learn an approximation function for some observed data [4].

The neurons have inputs, similar to biological synapses, which are multiplied by the strength of the signals (called weights). The activation of the neuron is then calculated by a mathematical function. This activation is based on a given threshold. ANN is an interdependent network of such single neurons to process the input information coming to it [5].

If the weight has a negative value, it means that the signal is inhibited. The output of each neuron is different depending on the weights connected to it. For specific inputs, the weights can be adjusted to get the required output using algorithms. Also, the weight can be zero if there is no connection between two neurons. So these weights are what determine the output of the neural network [6].

In short, the weights connecting the neurons can be considered as memory of the neural network. This process of adjusting the weights is called learning or training. Weighted sum of inputs are given to the neuron, and its output is a function of this input [7].

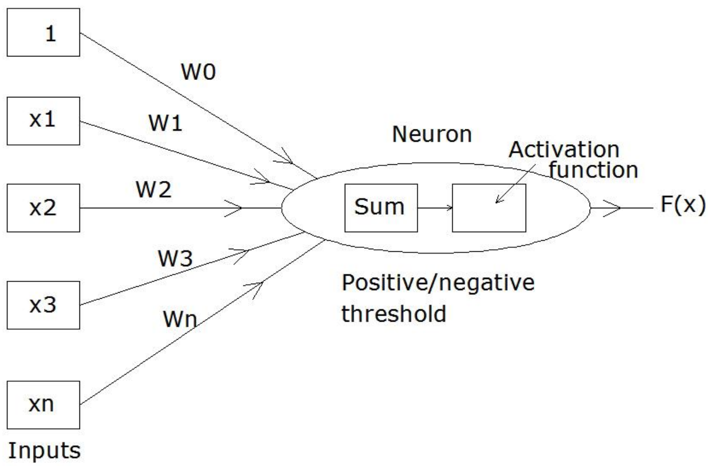

***Fig -2**: Model of an artificial neuron*

Output function, F(x) is a sigmoid function given by

$$F(x) = \frac{1}{1 + e^{-sum}}$$

where

$$sum = \sum_{i=0}^{n} x_i W_i$$

The sum is the weighted sum of the inputs multiplied by the weights between one layer and the next. The activation function used is a sigmoid function, which is a continuous and differentiable approximation of a step function. [6] The architecture of the ANN comprises of-

-input layer: Contains neurons equal to the number of inputs

-hidden layer(s): The number of hidden layers and the number of neurons in each layer depends on the complexity

-output layer: Though multiple outputs can be present, usually it has one neuron, and its output ranges from 0 to 1, that is, greater than 0 and less than 1 [8]

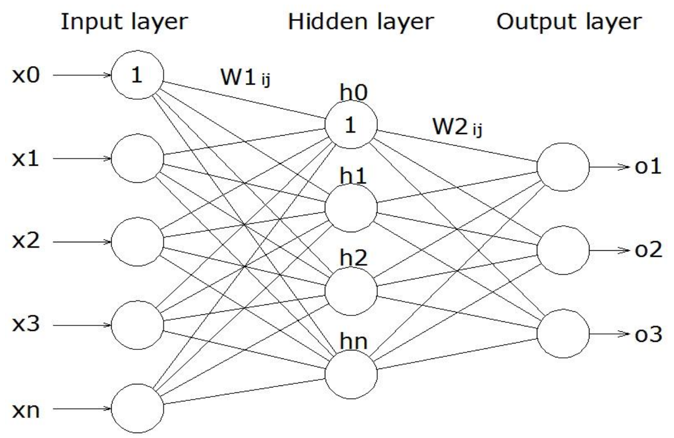

***Fig -3**: Neural network architecture*

BACK PROPAGATION ALGORITHM

Back propagation algorithm is proposed for training the neural network in this paper. Initially a random set of weights and thresholds are assigned. The network adjusts the weights each time it sees an input-output pair. Each pair requires a forward pass stage and a backward pass stage [9]

During the forward pass, a sample input is presented to the network and activations flow until they reach the output layer. During the backward pass, the network's actual output is compared with the target output and error estimates are computed for the output units. The errors are reduced by adjusting the weights to the output layer. This gives the errors for the hidden layers. This repeats till the errors propagate back to the input layer, hence the name, back propagation algorithm [9].

After the back propagation algorithm has seen all input-output pairs, and adjusted the weights that many times, we say that one epoch has been completed. Training a back propagation network usually requires many epochs [10].

TRAINING STEPS [10]

Step 1: 'M' is the number of units in the input layer, 'L' is number of units in output layer, 'N' is number of units in hidden layer. The input and hidden layers have one extra unit for thresholding. So the number of units in these layers will be indexed by the ranges 0 to M and 0 to N.

x_j, h_j, o_j are activation levels of the units in the input, hidden, output layers respectively.

Weights connecting input to hidden layers denoted by $W1_{ij}$, and weights connecting hidden to output layer denoted by $W2_{ij}$, where 'i' represents from-neuron of previous layer and 'j' represents to-neuron of the next layer

Step 2: Initialize the weights in the network, randomly selected between -0.1 and 0.1

$W1_{ij}$=random (-0.1, 0.1) for all i=0 to M, j=1 to N

$W2_{ij}$=random (-0.1, 0.1) for all i=0 to N, j=1 to L

Step 3: Initialize activations of the thresholding units. These are fixed throughout and do not change $x_0=1.0$, $h_0=1.0$

Step 4: Choose an input-output pair, such as x_i for input vector and y_i for output vector. Assign activation levels to the input units

Step 5: Propagate the activations from input layer units to hidden layer units using the activation function-

$$h_j = \frac{1}{1+e^{-\sum_{i=0}^{M} W1_{ij}x_i}} \quad \text{for all j=1…N}$$

where $W1_{0j}$ is the thresholding weight of hidden unit 'j'

Step 6: Propagate the activations from hidden layer units to output layer units using the activation function

$$o_j = \frac{1}{1+e^{-\sum_{i=0}^{M} W2_{ij}h_i}} \quad \text{for all j=1…L}$$

where $W2_{0j}$ is the thresholding weight of output unit 'j'

Step 7: Compute the errors of the units in the output layer, denoted by $\delta 2_j$. The errors are based on the network's actual output o_j, and target output, y_i.

$$\delta 2_j = o_j(1-o_j)(y_j - o_j) \quad \text{for all j=1…L}$$

Step 8: Compute the errors of the units in the hidden layer, denoted by $\delta 1_{ij}$.

$$\delta 2_{ij} = Ih_j(1-h_j)\sum_1^L \delta 2_i W2_{ij} \quad \text{for all j=1…N. I}$$

is input at the node

Step 9: Adjust the weights between hidden and output layer. The learning rate, denoted by η, is taken as 0.35 as a reasonable value as it can be between 0.2 and 0.5

$\Delta W2_{ij} = \eta\delta2_j h_i$ for all i=0…N, j=1…L

Step 10: Adjust the weights between input and hidden layer.

$\Delta W1_{ij} = \eta\delta1_j x_i$ for all i=0…M, j=1…N

Step 11: Go to step 4 and repeat the same for all input-output pairs. When all the input-output pairs have been presented to the network, one epoch has been completed. Repeat steps 4 to 10 for as many epochs as desired.

The speed of learning can be increased by modifying the weight modification steps 9 and 10 to include a momentum term, α
The weight update formulae become-

$$\Delta W2_{ij}(t+1) = \eta\delta2_j h_i + \alpha W2_{ij}(t)$$
$$\Delta W1_{ij}(t+1) = \eta\delta1_j x_i + \alpha W1_{ij}(t)$$

where h_i, x_i, $\delta2_j$ are measured at time 't+1'. $\Delta W_{ij}(t)$ is the change the weight experienced during the previous forward-backward pass [10].

DESIGN CONSIDERATIONS

Selection of number of layers and number of units in each layer depends on the complexity of the problem

a. Criterion for choosing number of layers: Sometimes it is necessary to have extra hidden layers. Learning is sometimes faster with multiple hidden layers as the input is highly nonlinear, i.e., hard to separate with a series of straight lines [10].

b. Criterion for selection of number of neurons in a layer: In the input layer, number of neurons depends on number of input variables. In the hidden layer, the choice is flexible. In the output layer, usually one output neuron will serve the purpose [11]. However, any number of output neurons can be selected, depending on the suitability and nature of the output.

In this study, the number of input neurons is taken as 10, hidden as 20, and output as 1. The output value ranges from 0.1 to 0.9.

c. Selection of learning rate, η: It is a scale factor which tells how far to move in the direction of the gradient. A small value will lead to slower speed, while a larger value will cause overshooting the solution vector [10]

d. Selection of momentum gain, α: Best results have been obtained by letting α to be zero for the first few iterations, then increasing it to 0.9 for the rest of the training. This process gives the algorithm some time to find the right direction, and then move in that direction with enhanced speed

e. Weight initialization: Initial weights randomly chosen as 0.1. However it would be advantageous to replace the initialized set of weights by the trained set of weights after some iterations, to ensure there is no undue delay in training when training has to be done repeatedly.

f. Knowledge base: A good amount of knowledge base is required that include messages about alarms for simple events, compound events, set of weights between layers, input pattern for each event, and target output for each event [9].

Input pattern refers to any combination of alarms in a group that is expected to be received in the substation for a fault. These patterns for all events are stored in the knowledge base. Similarly, for all patterns of inputs, corresponding output value is stored in the knowledge base, which is need for the training of ANN.

IMPLEMENTATION METHODOLOGY

Any fault which causes the issue of alarms is defined as an event. The alarms associated with it are called elements of this event. Alarms corresponding to a particular event are called characteristic set of alarms. Thus each event is associated with a characteristic set of alarms. These characteristic sets are treated as patterns for training the ANN. One network accepts these characteristic set of alarms as inputs and displays a specific message about the event that has occurred as the output.

An example of an event including four alarms is-
E = {a1, a2, a3, a4}; where E represents the event and 'a' represent the characteristic set of alarms.

Following steps used for alarm processing
Step 1: Alarm set received is acknowledged and accepted
Step 2: Trip and caution alarms are separated. This is done by checking its group
Step 3: Within the group, the number of alarms received is determined
Step 4: If received set is single, it is identified and alarm message displayed from the knowledge base
Step 5: For multiple alarms, the set is transferred to the neural network for processing
Step 6: Output of hidden layer calculated by getting the trained set of weights between input and hidden layer
Step 7: Output of ANN calculated by getting the trained set of weights between hidden layer and output layer. This output displays a message about the fault corresponding to the received set of alarms
Step 8: Identification of missing and false alarms: The characteristic set for this event is compared with the received set. Any alarm in the defined set, but not in the received set is a missing alarm. Any alarm in the received set, but not in the defined set is a false alarm.

SYSTEM STUDIED

The system under study is a 66/11 kV substation consisting of two transformer banks of 66/11 kV with a 66 kV circuit breaker. The LV side has six feeders. For convenience sake, faults are categorized into three groups. The first group incorporates faults of the 66 kV circuit breaker alone. The next two groups correspond to faults of the feeders.

For the group 1, list of alarms is as-

Alarm	Alarm message
a1	66 kV breaker tripped on differential protection
a2	66 kV breaker tripped HV restricted EF
a3	66 kV breaker tripped LV backup protection
a4	66 kV breaker tripped HV backup protection
a5	66 kV breaker tripped Tr1 backup protection
a6	66 kV breaker tripped Tr2 Bucholtz relay
a7	Incoming supply failure

Here, a7 is a redundant alarm as it is obvious and comes in with every event.

Compound events of group 1-

Event	Description	Characteristic set of alarms
1	Tr 1 HV earth fault	[a1, a2, a3, a5, a7]
2	Tr 1 HV phase fault	[a1, a3, a5, a7]
3	Tr 2 HV phase fault	[a1, a3, a6, a7]
4	Tr 3 HV earth fault	[a1, a2, a3, a6, a7]
5	Tr 1 LV fault	[a1, a4, a5, a7]
6	Tr 2 LV fault	[a1, a4, a6, a7]

The input to the ANN for event number 1 in this group 1 is entered as [1 1 1 0 1 0 1]. For events in group 2, input may be entered as [2 0 2 2 2 0 2], for group 3, it may be as [0 3 3 0 0 3] and so on.

A combination of ANN and Expert Systems is used to give out the result. For example, for group 1, if the output of the ANN is in the range of 0.05 to 0.15, based on the knowledge base in the ES, the processor displays "Transformer 1 HV earth fault"

CONCLUSIONS AND OBSERVATIONS

1. It is generally observed that increase in number of input patterns increase the number of iterations.
2. ANN faces inability in recognizing patterns which are closely similar as two distinct patterns. It would be necessary to enhance the number of hidden neurons to overcome this.
3. After training, substituting initial set of weights has yielded in saving time for further training
4. When the patterns are distinct, the number of iterations has been observed to be around a few hundreds. It may be in thousands if the patterns are closely similar.
5. The network is flexible and can be enlarged to include more input variables, and more hidden layers with little effort.

REFERENCES

1. Eric Zabre, V Jiménez, "Diagnosis of Alarm Systems and Impact in the Maximization of Operator's Effectiveness at Power Plants", Proceedings of the World Congress on Engineering and Computer Science 2014 Vol I WCECS 2014, 22-24 October, 2014, San Francisco, USA
2. Andreas Bauer et al, "Alarm Processing with Model-Based Diagnosis of Discrete Event Systems", 22nd International Workshop on Principles of Diagnosis, NICTA, Australia
3. T S Madhava Rao, "Power System Protection: Static Relays", TMH
4. Selvakumar Kamalanathan, Sendhilkumar Selvaraju, "Collaborative Approaches for Personalized Web Search Using Fuzzy Neural Networks", Global Trends in Information Systems and Software Applications: 4th International Conference, ObCom 2011 Proceedings
5. Gopal Datt, "An Evolutionary Approach: Analysis of Artificial Neural Networks", International Journal of Emerging Technology and Advanced Engineering (IJETAE), Volume 2, Issue 1, January 2012
6. Carlos Gershenson, "Artificial Neural Networks for Beginners"
7. Jeff Heaton, "Solving Problems with Neural Networks", http://www.heatonresearch.com/
8. M. Abdelrahman "Artificial neural networks based steady state security analysis of power systems", Thirty-Sixth Southeastern Symposium on System Theory 2004 Proceedings, 2004
9. Kevin Knight, "Connectionist ideas and algorithms (neural networks)", Communications of the ACM, Nov 1990
10. Elaine Rich, Kevin Knight, Shivashankar B Nair, "Artificial Intelligence", 3rd edition, TMH
11. Lee, C.J.. "Sensitivity analysis on a multilayer perceptron model for recognizing liquefaction cases", Computers and Geotechnics, 200909