6TH INTERNATIONAL CONFERENCE ON ELECTROMECHANICAL AND POWER SYSTEMS
October 4-6, 2007 - Chişinău, Rep.Moldova

A RADIAL BASIS FUNCTION NEURAL NETWORK WITH PRUNING STRATEGY FOR DISTANCE RELAYING SYSTEMS

Ovidiu IVANOV, Mihai GAVRILAŞ, Marcel ISTRATE

"Gh. Asachi" Technical University of Iasi, Romania
ovidiuivanov@ee.tuiasi.ro; mgavril@ee.tuiasi.ro; mistrate@ee.tuiasi.ro;

Abstract **– Distance relays are often used for the protection of transmission lines. Presently, the traditional methods are replaced by relays using modern numerical algorithms, which offer improved protection sensitivity and selectivity. This paper presents a hybrid ANN-fuzzy fault classification technique based on a Gaussian RBF neural network with a fixed number of hidden units. A pruning strategy is used to remove the insignificant units. A case study with results for several type faults proves the method's efficiency.**

Keywords**:** *distance protection, RBF neural networks, pruning strategy, fault classification .*

1. INTRODUCTION

The sensitivity and selectivity of protection systems are fundamental requirements for a reliable protection of the power system against complex, unexpected events, such as cascading failures which can produce heavy technical damages and economic losses.

Distance relays for the protection of transmission lines are used on a large scale for the protection of transmission lines. Along with other protection and automation devices applied in power systems, distance relays have been developed during the last decades using at first traditional methods, then modern, digital units based on dedicated computational algorithms.

Most algorithms used for fault identification and location estimate the impedance between the distance relay and the fault location, as the impedance of the line is proportional to its length. This is usually done by evaluating the system's state after a fault has occurred, using not measured impedance data, but voltage and current waveforms measured in the post-fault state.

Traditional methods use well-known algorithms such as the Fourier transform, the Kalman filter, symmetrical components, orthogonal components or the model of long lines differential equations.

Basically, the numerical processing of the voltage and current waveforms that define the post-fault state it is a pattern recognition problem, which is suitable for use with today's modern computational intelligence algorithms.

The current approaches found in the literature use methods such as artificial neural networks (ANNs) and fuzzy systems (FSs) [1,2].

This paper presents the second stage of a numeric simulation of a digital distance protection developed by the authors.

The first stage aimed to estimate the fundamental frequency of the voltage and current waveforms by their amplitude and phase [3]. A well known neural network architecture was used, the Multilayer Perceptron (MLP), trained with the error backpropagation and RProp algorithms. Subsequently, to determine the accurate amplitude and phase values, a least mean square (LMS) error algorithm was applied fort the values estimated by the MLP network. Results were presented for different system conditions, fault inception angles and fault locations.

The following stage aims to identify accurately the fault type, and it is presented in this paper.

The third and last stage will determine the fault location on the line.

For the fault type identification, the authors chose a hybrid ANN-FS algorithm that uses a special type of ANN architecture, the radial basis function (RBF) network. The method was tested on a transmission line model simulated in the ATP-EMTP program and the results prove its efficiency.

2. THE RADIAL BASIS FUNCTION NEURAL NETWORK

As stated before, the numerical processing of the voltage and current waveforms that define the post-fault state it is a pattern recognition problem, easily applicable to ANNs.

For the fault type identification problem, the authors used the radial basis function network (Fig. 1). Compared to other ANN types, like the MLP, the RBF has a more compact structure and requires less computation time. The RBF network used in this approach has three layers: one input layer, a hidden layer and an output layer

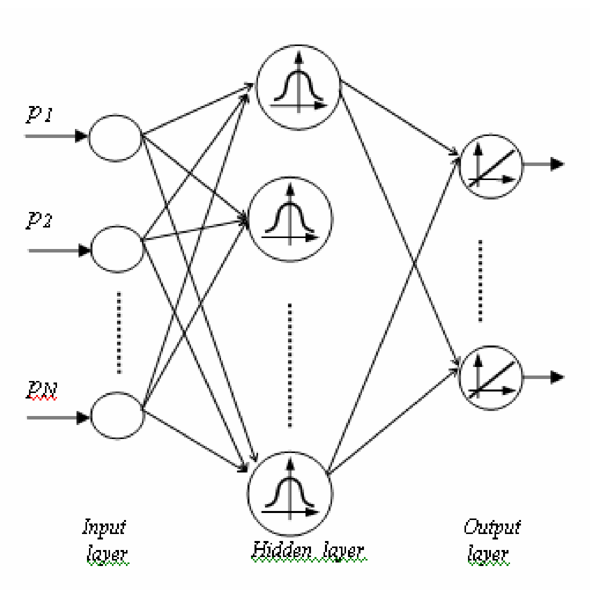

Figure 1: The RBF network architecture

Each hidden neuron has as net input the vector distance between its weight vector *w* and input *p*, multiplied with the bias vector *b*.

$$net_k = \prod_{i=1}^{N} (\|w_{k,i} - p_{k,i}\| \cdot b_h) \qquad (1)$$

where N is the size of the input model vector *p*.
The hidden neuron transfer function is the Gaussian function:

$$f(net_k) = e^{-net_k^2} \qquad (2)$$

When a input model is presented, each hidden unit will have an output which describes how close is that input to the unit's weights. This means that the hidden unit will produce outputs near 0 if the input and weight vectors are quite different and outputs close to 1 when they are almost identical.
The use of Gaussian activation functions gives the RBF network characteristics similar to fuzzy systems, as the outputs of the hidden units combined on the inputs of the network's third layer describe fuzzy rules, one rule for each output unit.
The network architecture is determined through an optimization procedure. For the hidden layer, two optimization procedures can be applied. The first starts with 0 hidden neurons and uses the principle of network growing, which adds hidden units whenever the performance does not improve [1].
The second approach, used in this paper, involves a reciprocal method. The algorithm starts with a fixed number of hidden neurons, equal with the number of training models, and applies a pruning strategy by removing insignificant input and hidden units. This is done because often a number of inputs and hidden neurons have a insignificant contribution to the global output and their removal provides a more compact topology and reduces computation time.

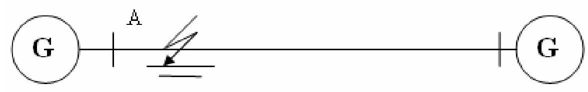

Figure 2: The transmission line model

Parameter	Value
Length	300 [km]
Voltage	400 [kV]
Type	aerial
Phase type	2*450 mm^2

Table 1: Transmission line parameters

3. CASE STUDY

3.1. Description

The RBF neural network with pruning strategy described above was used as a fault classifier. To obtain the training data, the model of a transmission line was simulated using the ATP/EMTP software package [4]. The line model, presented in Fig. 2 and Table 1, uses nominal π multipoles with lumped parameters, where all magnetic mutual influences are taken into account.
The RBF network was trained with data for different fault types and its performances were tested.
Several fault types were simulated: *ag, bc, bcg, abc*, (where *a, b, c* are the tree phases and *g* is the ground) at 25%, 50% and 75% distance from point A, fault resistances of 0, 4, 7 and 10 Ω and 0 degrees inception angle.
One input model has 147 samples, (Fig. 3), representing 7 complete periods for the current and/or voltage fundamental waves on all three phases and the zero sequence current, sampled at a rate of 1 ms and taken from the post-fault stabilized operating condition, obtained with the technique described in [3]. There are a total of 39 input models.
The network performances were tested for all fault types and resistances mentioned above, with data measured for faults occurring at 33% distance from point A.
As for the neural network architecture, the maximum number of hidden neurons was set equal to the inputs

model number. Because the outputs are linear, this allows network training with zero error.
For the fault identification problem, there are four output units, the *a,b,c*, phases and *g* (ground). The desired outputs are 1 for faulted phase or 0 for unaffected phase. In the generalization stage, the following convention applies: faulted phase for values greater than 0.5 and unaffected phase for outputs greater than 0.5.

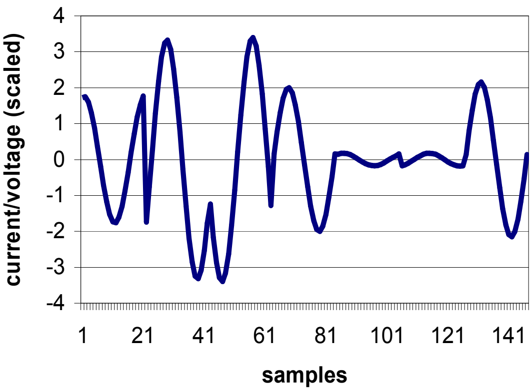

Figure 3: Input data for the *ag* fault type

3.2. Results

The fist goal was to establish the standard deviation value for which the RBF network identifies fault types with maximum precision, i.e. values closer to 1 and 0. As performance parameter, the average difference between the desired and the obtained outputs, for the entire test data set, was used.
As the Fig. 4 plot shows, a standard deviation value of 1 was too narrow for a feasible approximation, but for the range from 2 to 30, the results in the generalization stage are much better and comparable, with a minimum of 0.0246 (standard deviation 5) and not exceeding 0.08 (standard deviation 18). The results presented in the following stage were obtained with a RBF network trained with a standard deviation value of 5.
The results obtained for the test data set are presented in Tables 2-5 for different fault types and resistances.
Subsequently, a pruning strategy was applied to reduce the network topology. At first, a reduction of the hidden neurons number was attempted based on their global output value. As seen in Table 6, a number of maximum two neurons could be pruned, with outputs no larger than 1. The average error of the pruned network was 0.0386.

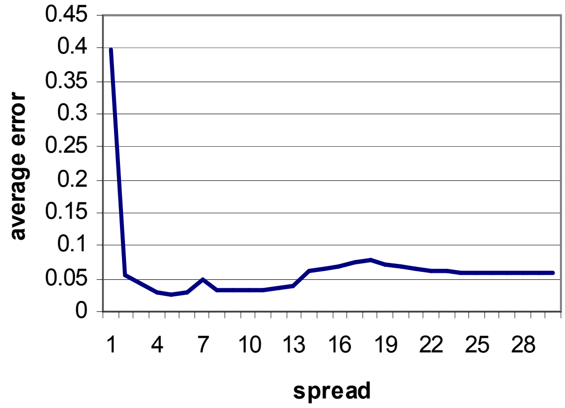

Figure 4: The optimal standard deviation value for the RBF network

	R_f=0Ω	R_f=4Ω	R_f=7Ω	R_f=10Ω
a	1.0041	1.0038	1.0035	1.0033
b	-0.0044	-0.0041	-0.0038	-0.0036
c	-0.0044	-0.0041	-0.0038	-0.0036
g	1.0044	1.0041	1.0038	1.0036

Table 2: Results for the ag fault

	R_f=0Ω	R_f=4Ω	R_f=7Ω	R_f=10Ω
a	-0.0003	-0.0019	-0.0033	-0.0048
b	0.9963	0.9980	0.9996	1.0013
c	0.9963	0.9980	0.9996	1.0013
g	1.0155	1.0134	1.0122	1.0113

Table 3: Results for the *bcg* fault

	R_f=0Ω	R_f=4Ω	R_f=7Ω	R_f=10Ω
a	-0.0090	-0.0070	-0.0444	-0.1360
b	1.0090	1.0070	1.0443	1.1357
c	1.0090	1.0070	1.0443	1.1357
g	-0.0097	-0.0070	-0.0444	-0.1360

Table 4: Results for the *bc* fault

	R_f=0Ω
a	1.0031
b	1.0080
c	1.0080
g	-0.0134

Table 5: Results for the *abc* fault

Pruning threshold	Pruned neurons	Average error
0	0	0.02456
0.1	2	0.03861
1	2	0.03861
1.1	3	0.27533

Table 6: The maximum number of pruned neurons

Using a similar procedure, insignificant input layer weights were pruned. A number of maximum 1237 input weights (22.7%) could be pruned without affecting the recognition process. This number corresponds to a threshold of 0.17. The results for the pruned network are presented in Tables 7-11.

Pruning threshold	No. of pruned input weights	Average error
0.01	232	0.038462
0.05	549	0.078508
0.1	786	0.082385
0.11	847	0.068133
0.12	902	0.124621
0.13	945	0.085623
0.14	1014	0.093077
0.15	1087	0.085987
0.17	1237	0.101636
0.19	1354	0.10995

Table 7: The maximum number of pruned input weights

	R_f=0Ω	R_f=4Ω	R_f=7Ω	R_f=10Ω
a	1.0045	0.9706	0.9476	0.9275
b	-0.0049	0.0287	0.0515	0.0714
c	-0.0049	0.0287	0.0515	0.0714
g	1.0055	0.9719	0.9491	0.9291

Table 8: Results for the *ag* fault, pruned network

	R_f=0Ω	R_f=4Ω	R_f=7Ω	R_f=10Ω
a	-0.0878	-0.0346	0.0025	0.0356
b	1.0866	1.0316	0.9932	0.9589
c	1.0866	1.0316	0.9932	0.9589
g	0.5406	0.8832	1.1175	1.3215

Table 9: Results for the *bcg* fault, pruned network

	R_f=0Ω	R_f=4Ω	R_f=7Ω	R_f=10Ω
a	-0.0136	0.0390	0.0805	0.2669
b	1.0100	0.9612	0.9198	0.7337
c	1.0100	0.9612	0.9198	0.7337
g	-0.0899	0.0393	0.0805	0.2669

Table 10: Results for the *bc* fault, pruned network

	R_f=0Ω
a	1.0175
b	0.9900
c	0.9900
g	0.1243

Table 11: Results for the *abc* fault, pruned network

Increasing the pruning threshold over 0.17 resulted in output values above 1.5 or below -0.5 and recognition errors, without increasing substantially the number of pruned weights.

4. CONCLUSIONS

A RBF neural network fault classifier is proposed in this paper, as the second stage in the development of a numeric simulation of a digital distance protection. The RBF network is trained using sampled fundamental current and voltage waveforms from the post-fault stabilized state and the zero sequence current. A pruning strategy is applied to remove insignificant hidden units and input weights. The fault classifier identifies properly all types of faults, for different fault distances and resistances

References

[1] P.K.Dash, A.K. Pradhan, G. Panda *A Novel Fuzzy Neural Network BasedDistance Relaying Scheme* , IEEE Trans. on Power Delivery, vol. 15, no. 3, 2000, pp. 902-907.

[2] A. Osman, T. Abdelazim, O.P. Malik, *Adaptive Distance Relaying Technique using On/line Trained Neural Network*, Proc. IEEE PES General Meeting, Toronto, Canada, 2003

[3] M. Gavrilaş, M. Istrate, O. Ivanov *Estimation of Fundamental Frequency Components of Voltage and Current for Digital Distance Relays*, Proc. International Conference on Fundamentals of Electrotechnics and Circuit Theory, IC-SPETO 2007, Gliwice, Poland, 2007

[4] ***, *Alternative Transients Program (ATP) Rule Book*, 1987-92, Canadian/American User Group.