

Sistemas Híbridos Aplicados à Filtragem de Sinais de Alarme de Proteção de uma Subestação Telecomandada

Milde Maria da Silva Lira, Manoel Afonso de Carvalho Jr.
Departamento de Engenharia Elétrica e Sistemas de Potência – UFPE
Av. Acadêmico Hélio Ramos, s/n – Cidade Universitária
50740-530 Recife PE
E-mail: milde@npd.ufpe.br

Abstract

In this work an Artificial Intelligent Hybrid System is applied to the filtering process of an alarm signal set, generated at a substation.

This combined system consists in an Artificial Neural Network, using MLP backpropagation algorithm and a Fuzzy Logic Inference System of the Mamdani type.

The results show that the Hybrid System is able to interpret correctly the alarm information, resulting in a better performance than the Neural System alone.

1. Introdução

A crescente demanda de energia elétrica pela sociedade moderna ocasiona o aumento dos Sistemas Elétricos, tornando-se estes de grande porte.

Uma conseqüência imediata disso é a necessidade de implementar sistemas de proteção baseados em um grande número de informações. Quando ocorre um distúrbio, os eventos isolados apresentam-se em uma quantidade tal que dificultam ao operador o reconhecimento da causa destes distúrbios (diagnose) e o estabelecimento de ações corretivas a serem executadas.

Para tentar solucionar esses problemas, vários trabalhos [1, 2, 3, 4] vêm sendo desenvolvidos. Uma boa parte utiliza técnicas recentes como Redes Neurais e ainda, outra parte utiliza técnicas mais tradicionais, como sistemas baseados em Lógicas/Regras ou sistemas que utilizam duas ou mais destas Técnicas de Inteligência Artificial– Sistemas Híbridos – visando alcançar as vantagens de cada uma delas e reduzir suas desvantagens.

Neste trabalho, um sistema híbrido é aplicado ao estudo da filtragem de sinais de alarme de proteção da subestação da Mirueira, pertencente ao sistema da Companhia Hidroelétrica do São Francisco (CHESF). O Sistema de Supervisão Local daquela subestação está sendo automatizado e seus sinais de alarmes serão enviados para o Centro Regional de Operação Leste (CROL), com a finalidade de ser supervisionada e controlada remotamente.

Com o intuito de reduzir essa quantidade de alarmes a ser transmitida ao CROL/CHESF, o qual deverá abrigar não apenas os alarmes daquela subestação, mas futuramente os alarmes de todas suas outras a serem contempladas pelo projeto de teleassistência, será realizada uma filtragem dos alarmes da SE Mirueira. Quando essa filtragem é realizada através de “hardware”, o sistema apresenta-se pouco flexível e quaisquer modificações no sistema de proteção resultam alterações físicas neste “hardware”. Nesse caso, espera-se que o custo operacional seja elevado.

A utilização do Sistema Híbrido para realizar esta função de filtragem é bem mais vantajosa em virtude da necessidade de alterar apenas o programa controlador.

O objetivo deste trabalho é demonstrar a viabilidade e o bom desempenho desses sistemas para executar função de filtragem.

2. Metodologia

O Sistema Híbrido é composto de uma Rede Neural e um Sistema de Inferência Fuzzy. Estas duas técnicas apresentam funções bem específicas que serão detalhadas a seguir.

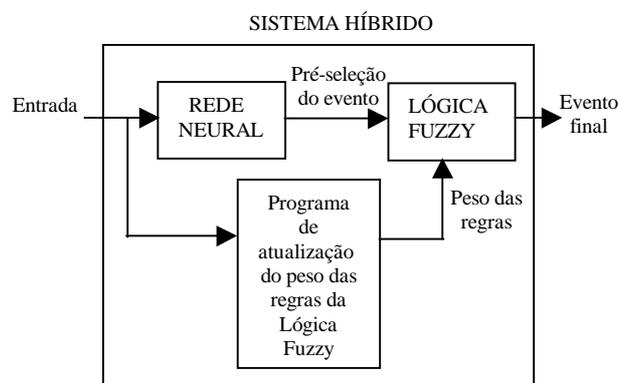


Figura 1: Diagrama esquemático do sistema Híbrido Inteligente

2.1. Redes Neurais

Uma das principais vantagens da Rede Neural é a sua capacidade de generalização, além de outras destacadas em [1]. Isso explica a razão do grande número de publicações utilizando esta técnica de Inteligência Artificial.

A função da Rede Neural no Sistema Híbrido apresentado é realizar uma pré-seleção do evento ocorrido, baseado na configuração dos sinais de alarmes que lhes são apresentados. No esquema da figura 1, pode-se verificar que a entrada da rede neural é a entrada do Sistema Híbrido.

Para o desenvolvimento do Sistema Inteligente com rede neural é necessário, antes de qualquer procedimento, escolher um conjunto de treinamento que contenha todas as características das classes do problema. Em seguida, de posse dessa base de dados e através de experimentos, identificar a configuração da rede que apresenta melhor desempenho na classificação dos eventos, de acordo com os diferentes parâmetros, como por exemplo, a taxa de aprendizagem e o momento.

2.1.1. Geração dos exemplos de treinamento

Uma carga severa de trabalho na tarefa de treinamento de uma única rede neural representativa de todo o sistema elétrico é inevitável quando ela emprega padrões de treinamento derivados simplesmente do estado operacional dos disjuntores e relés de um Sistema de Potência complexo de grande porte [5].

Para atenuar este problema, o Sistema de Potência foi dividido em 4 módulos: um representativo do Módulo Linha de Transmissão em 230kV (antiga)- LT-04C4; um representativo do Módulo Linha de transmissão em 230kV (nova)- LT-04F3; um representativo do Módulo Transformadores- TR-04T1 e um representativo do Módulo Linha de Transmissão em 69kV- LT-02J1. Cada módulo tem sua Rede Neural correspondente, permitindo, desta forma, construir padrões de treinamento com mais facilidade para estas redes do que para uma única rede representativa de todo o sistema. Além disso, em caso de mudança no sistema parcial, um pequeno número de RNAs precisam ser treinadas novamente.

Através de estudos realizados por Engenheiros da CHESF foram estabelecidos os alarmes, em função da configuração mínima de alarmes necessária para prover os meios para a coordenação da operação e da manutenção do sistema elétrico, que deveriam chegar até o CROL/CHESF e foram montado diagramas de blocos para cada um dos módulos da SE.

Mediante o diagrama de blocos de cada um dos módulos e observando o grupo de alarmes que através da

porta OU causaria a atuação de um único alarme (evento), gerou-se a base de dados para cada uma das redes. Cada um dos relés no conjunto de entrada é representado pelo número um (1), significando relé ativado ou zero (0) desativado. E as saídas (eventos) são representadas por códigos com 2 ou 4 dígitos dependendo do número de eventos do módulo a ser tratado.

Para geração dos padrões ou exemplos de treinamento e teste, utilizou-se o programa Excel para gerar aleatoriamente números 1s e 0s indicadores do estado de cada um dos relés ligados pela lógica OU que fornecesse o evento correspondente estabelecido no diagrama de bloco. Para maiores esclarecimentos observar a tabela 1 referente aos exemplos criados para treinamento da rede do módulo LT-02J1. Pode-se verificar que os alarmes 1 e 2 estão ligados por uma porta lógica OU, e os alarmes 3 e 4, por uma outra porta lógica OU.

Tabela 1: Exemplos Entrada / Saída para treinamento do módulo LT-02J1

ENTRADA						
Exemplos	Alarm1	Alarm2	Alarm3	Alarm4	SAÍDA	
1	1	0	0	0	1	0
2	0	1	0	0	1	0
3	1	1	0	0	1	0
4	0	0	1	0	0	1
5	0	0	0	1	0	1
6	0	0	1	1	0	1

Os números de exemplos de treinamento e teste de cada módulo encontram-se indicados na tabela 2. A variação da quantidade de exemplos entre os módulos resulta do número total de alarmes e do número de alarmes que estão ligados pela porta lógica OU em cada um desses módulos.

Tabela 2: Números de exemplos de treinamento e teste

Módulo	Número de exemplos de treinamento	Número de exemplos de teste	total de exemplos
LT-04C4	550	66	616
LT-04F3	1008	120	1128
TR-04T1	180	33	213
LT-02J1	6	0	6

Tabela 3: Alarmes de entrada e eventos de cada um dos módulos

Módulo	Nº de alarmes de entrada	Nº de eventos	Nº de dígitos do evento
LT-04C4	19	8	4
LT-04F3	32	11	4
TR-04T1	28	15	4
LT-02J1	4	2	2

A tabela 3 mostra o número de alarmes de entrada, de eventos e de dígitos associados a estes eventos. Esses parâmetros são de grande importância para definir a arquitetura da rede de cada um dos módulos. Por exemplo, o número de alarmes de entrada fixa o número de neurônios na camada de entrada da rede, e o número de dígitos do código do evento estabelece o número de neurônios na camada de saída.

Deve-se notar que o número de neurônios das camadas de entrada e saída dependem da aplicação específica do sistema a ser analisado e que a determinação do número de neurônios das camadas intermediárias é empírica. Este número é obtido através da análise do desempenho da rede neural.

2.1.2. Arquitetura da rede e o simulador SiReNe

O estabelecimento do tamanho de uma rede neural é feito por tentativa. O número de neurônios na camada intermediária está associado com a habilidade de mapeamento da rede. Quanto maior a rede mais poderosa ela é. Entretanto, crescendo continuamente o tamanho da rede, haverá um ponto a partir do qual a generalização começará a piorar. Esta piora na generalização deve-se ao fato de estar sendo utilizado um número de parâmetros de ajuste (pesos) muito grande para o conjunto de treinamento. Como consequência a rede memoriza os padrões que lhes foram apresentados no conjunto de treinamento e responde de forma imprevisível aos padrões nunca visto antes (conjunto de teste). O problema é encontrar o menor número de graus de liberdade que alcance o desempenho requerido no conjunto de teste.

Na tabela 4 pode-se verificar a arquitetura das redes utilizadas após várias simulações realizadas através do simulador de redes neurais- SiReNe[6, 7], com alteração de uma série de parâmetros considerados relevantes para seu treinamento, como a taxa de aprendizagem, o momento e outros.

O SiReNe é um simulador de redes neurais, que é utilizado em ambiente Windows e oferece recursos e facilidades para criar, configurar, testar e salvar redes neurais para posterior utilização por aplicações convencionais.

Tabela 4: Arquitetura das Redes Neurais

REDE NEURAL	NÚMERO DE NEURÔNIOS		
	Camada de Entrada	Camada Intermediária	Camada de Saída
LT-04C4	19	10	4
LT-04F3	32	4	4
TR-04T1	28	4	4
LT-02J1	4	1	2

O modelo de Rede Neural escolhido para compor o Sistema Híbrido foi a MLP de 3 camadas com treinamento backpropagation. Foram adotados a taxa de aprendizagem adaptativa e o momento com o objetivo de acelerar o treinamento e reduzir o risco de oscilação.

2.1.3. Resultado dos Testes

Com o fim do treinamento, foi realizado um teste no conjunto de exemplos de teste, e o resultado para cada uma das redes é mostrado na tabela 5.

Tabela 5: Resultado das redes neurais após treinamento

Rede Neural	Erro Médio Quadrático		Nº de exemplos de teste	Acerto (%)
	Conjunto de treinamento	Conjunto de teste		
LT-04C4	$2,70 \times 10^{-5}$	$2,61 \times 10^{-2}$	66	98
LT-04F3	$1,10 \times 10^{-2}$	$5,86 \times 10^{-2}$	120	94
TR-04T1	$5,43 \times 10^{-5}$	$1,14 \times 10^{-3}$	33	100
LT-02J1	$8,65 \times 10^{-5}$	-	6	100

Nesta tabela pode-se observar que, o percentual de acerto das redes variou de 94% a 100%. Isso mostra que as redes conseguiram entender e resolver o problema, obtendo um excelente resultado na generalização dos casos para os quais não foram treinadas.

2.2. Lógica Fuzzy

A Lógica Fuzzy é uma técnica de inteligência artificial que tem como principal característica, a manipulação de dados com um determinado grau de incerteza. Através da utilização da Teoria de Conjuntos Fuzzy [8], as incertezas inerentes aos dados de um sistema podem ser representadas através de expressões lingüísticas comumente utilizadas.

A função do Sistema de Inferência Fuzzy neste trabalho é corrigir todos os resultados classificados erroneamente pela Rede Neural, tornando o Sistema Híbrido mais confiável.

2.2.1. Concepção do sistema de inferência fuzzy

A ferramenta utilizada na concepção do Sistema de Inferência Fuzzy [9], provê os recursos para se criar e editar o sistema dentro de uma plataforma do MatLab. A Caixa de Ferramentas da Lógica Fuzzy fornece um número de ferramentas iterativas que permite o acesso a muitas funções através da interface gráfica com o usuário. Juntas, estas interface dispõem de um ambiente

para a montagem da arquitetura, implementação e análise do Sistema de Inferência Fuzzy.

Após vários experimentos através da substituição da forma da função membro, modificação de seus parâmetros e alteração dos pesos das regras, a fim de que o sistema apresentasse uma saída satisfatória (classificar corretamente os eventos mal classificados pela rede neural), chegou-se ao sistema mostrado na figura 2 para o módulo LT-02J1. O mesmo procedimento foi realizado com os demais módulos.

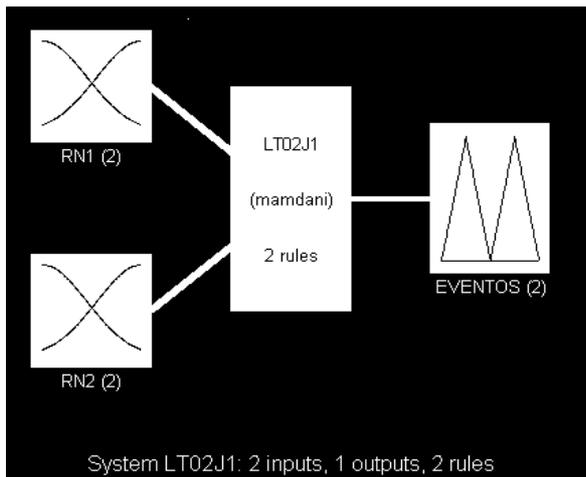


Figura 2: Sistema de Inferência Fuzzy do módulo LT-02J1

Para montagem do Sistema de Inferência tipo Mamdani de cada um dos módulos utilizou-se os seguintes processos: da lógica E = min ;da lógica OU = max; de implicação = min; de agregação = max e finalmente de defuzificação = centróide.

2.3. Programa de alteração dos pesos das regras

Analisando-se o Sistema de Inferência Fuzzy montado, concluiu-se que o mesmo só seria capaz de classificar corretamente os casos mal classificados pela rede neural, se os pesos das regras fossem atualizados em função do conjunto de estado dos alarmes, o qual é um dado de entrada da rede neural (ver fig.1). Com esta finalidade foi montado no MatLab um programa para cada um dos módulos, que lê o arquivo de entrada da rede neural, processa esses dados e fornece a atualização dos pesos ao Sistema de Inferência Fuzzy.

É importante salientar que as regras da Lógica Fuzzy são as mesmas para cada um dos módulos, qualquer que seja a configuração do conjunto dos alarmes. O que se altera, são os pesos das regras.

Ainda dentro desse programa foi realizada a leitura do arquivo de saída da rede neural fornecido pelo SiReNe, e efetuada sua conexão ao Sistema de Inferência fuzzy. Além disso, para melhor compreensão dos

resultados da defuzificação do Sistema de Inferência foi implementada uma saída de forma lingüística.

3. Resultados

O resultado obtido utilizando apenas a rede neural, na tabela 5, mostra que a rede apresenta um bom desempenho na tarefa de classificação, apresentando um percentual de acerto bastante elevado (94% a 100%) no conjunto de teste, porém a necessidade de sistemas que apresentem maior confiabilidade é imprescindível em Sistema de Potência e, por isto, a preocupação em melhorar esses resultados deu origem ao Sistema Híbrido utilizado neste trabalho.

Utilizando o Sistema Híbrido, os resultados apresentados confirmam o objetivo do trabalho, cuja principal finalidade é classificar corretamente o evento, dado o estado do conjunto dos alarmes.

Tabela 6: Saída desejada e obtida pela Rede Neural do módulo LT-04F3.

		E X E M P L O				
		1	2	3	4	5
S A Í D A	Desejada 1	1	0	0	1	1
	Obtida 1	0,9900	0,0021	0,0262	0,9891	0,9983
	Desejada 2	0	1	0	0	1
	Obtida 2	0,0162	0,9994	0,0562	0,0151	0,8080
	Desejada 3	0	1	1	1	0
	Obtida 3	0,0000	0,0000	0,0000	0,0000	0,0000
	Desejada 4	1	0	1	0	0
	Obtida 4	0,0000	0,0000	0,0000	0,0000	0,0000
E V E N T O	Real (código)	1 0 0 1	0 1 1 0	0 0 1 1	1 0 1 0	1 1 0 0
	Classificado (código)	1 0 0 0	0 1 0 0	0 0 0 0	1 0 0 0	1 1 0 0

A tabela 6 mostra 5 exemplos, onde os exemplos de 1 a 4 foram mal classificados pela Rede Neural representativa do módulo LT-04F3. As linhas em cinza referem-se aos resultados desejados e as linhas em branco aos obtidos, os quais foram extraídos do arquivo de saída do SiReNe.

Para obter o código do evento classificado, indicado na tabela 6, adotou-se como critério: **1** (um) aos valores de saída processados pela rede maiores ou iguais a 0,5; e **0** (zero) aos valores menores que 0,5.

Naquela tabela observa-se que a Rede Neural, além de classificar os exemplos 1, 2 e 4 em classes erradas, ela classificou o exemplo 3 em uma classe (0000) que não existe neste módulo, como pode-se verificar na tabela 7.

Tabela 7: Código e descrição dos eventos do módulo LT-04F3

Código do evento	Descrição do evento	Intervalo de defuzificação
1 0 0 0	ANORMAL_PROTEÇÃO	0 - 1
0 1 0 0	ATUAÇÃO_PROTEÇÃO	1 - 2
0 0 1 0	DISJ_ANORM ChLocManut	2 - 3
1 1 0 0	ANORMALIDADE_VÃO_LT-04F3	3 - 4
1 0 0 1	DISJ_ABERTURA_BAIXA_PRESSÃO	4 - 5
0 1 1 0	DISJUNTOR_APTO_RELIGAR	5 - 6
0 0 1 1	DISJUNTOR_BLOQUEIO_FUNÇÃO	6 - 7
1 0 1 0	DISJUNTOR_NÃO_DESLIGOU	7 - 8
0 1 0 1	PROT_FALHA_DISJ_DESLIG	8 - 9
1 1 1 0	SECCIONADORA_FALHA_OPERAÇÃO	9 - 10
1 1 0 1	TELEP_TRANSM_PERMIS_TRIP	10 - 11

As saídas da rede neural de cada um dos exemplos foram submetidas ao Sistema de Inferência Fuzzy, resultando, do processo de defuzificação, os valores mostrados na última coluna da tabela 8.

Tabela 8: Resultado da defuzificação

Exemplo	Saída da Rede Neural				Lógica Fuzzy (defuzificação)
	RN1	RN2	RN3	RN4	
1	0,9900	0,0162	0,0000	0,0000	4,5000
2	0,0021	0,9994	0,0000	0,0000	5,4923
3	0,0262	0,0562	0,0000	0,0000	6,4922
4	0,9891	0,0151	0,0000	0,0000	7,4927
5	0,9983	0,8080	0,0000	0,0000	3,5002

Estas saídas numéricas correspondem, de acordo com os intervalos de defuzificação referentes ao módulo LT-04F3 (última coluna da Tab.7), a forma lingüística abaixo:

4,5000 [DISJ_ABERTURA_BAIXA_PRESSÃO]
 5,4923 [DISJUNTOR_APTO_RELIGAR]
 6,4922 [DISJUNTOR_BLOQUEIO_FUNÇÃO]
 7,4927 [DISJUNTOR_NÃO_DESLIGOU]
 3,5002 [ANORMALIDADE_VÃO_LT-04F3]

cujos códigos dos eventos são respectivamente: 1001, 0110, 0011, 1010, 1100.

Conclui-se que o Sistema de Inferência classificou corretamente os exemplos de 1 a 4 classificados erroneamente pela Rede Neural e o exemplo 5 classificado corretamente, manteve na mesma classificação.

4. Conclusões

A Rede Neural implementada neste trabalho foi capaz de fornecer bons resultados na classificação dos eventos gerados em uma subestação de energia elétrica.

A combinação desta Rede Neural com o Sistema de Inferência Fuzzy resultou em um Sistema Híbrido Inteligente com habilidades para interpretar corretamente os eventos ocorridos, corrigindo algumas falhas da Rede Neural sozinha, tornando o sistema bem mais confiável.

Referências

- [1] V. Navarro A. Silva e R. Salem Zebulum, Sistema Híbrido para diagnose em sistemas de potência utilizando Redes Neurais e Lógica Nebulosa. II Congresso Brasileiro de Redes Neurais, 1995.
- [2] L. Biondi Neto, M. A. Pacheco, M. M. Vellasco, E. Lopes Passos e L. Chinganer. Sistema Híbrido de apoio à decisão para detecção e diagnóstico de falhas em redes elétricas. III Simpósio Brasileiro de Redes Neurais, 1996.
- [3] M. Kezunovic, I. Rikalo. Detect and Classify Faults Using Neural Nets. IEEE Computer Application in Power October 1996.
- [4] E. Chan. Using Network to Interpret Multiple Alarms. IEEE Computer Application in Power, April 1990.
- [5] K. Kim, J. Park. Application of hierarchical neural networks to fault diagnosis of power systems. Electrical Power & Energy Systems, Vol. 15, N° 2, April 1993.
- [6] A. Moura Viera. Simulador de Redes Neurais. DEL-UFRJ, 1998
- [7] L. Videira Assaf. NeuroLab - Ambiente de Desenvolvimento e Aplicação de Redes Neurais (Projeto Final de Curso). DCC/IM - UFRJ, 1999
- [8] Yuan-Yih Hsu, Chung-Ching Su. A Rule Based Expert System For Steady-State Stability Analysis. IEEE Transaction on Power Systems, Vol. 6, N° 2, May 1991.
- [9] J. S. Roger Jang and Ned Gulley. Fuzzy Logic Toolbox, for use with MATLAB.