

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

METODOLOGIA PARA COMPRESSÃO DE SINAIS DE ENERGIA ELÉTRICA A PARTIR
DE REGISTROS DE FORMA DE ONDA UTILIZANDO ALGORITMOS GENÉTICOS E
REDES NEURAIS ARTIFICIAIS

FABIOLA GRAZIELA NORONHA BARROS

TD 23/2016

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil
2016

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FABIOLA GRAZIELA NORONHA BARROS

METODOLOGIA PARA COMPRESSÃO DE SINAIS DE ENERGIA ELÉTRICA A PARTIR
DE REGISTROS DE FORMA DE ONDA UTILIZANDO ALGORITMOS GENÉTICOS E
REDES NEURAIS ARTIFICIAIS

TD 23/2016

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil
2016

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FABIOLA GRAZIELA NORONHA BARROS

METODOLOGIA PARA COMPRESSÃO DE SINAIS DE ENERGIA ELÉTRICA A PARTIR
DE REGISTROS DE FORMA DE ONDA UTILIZANDO ALGORITMOS GENÉTICOS E
REDES NEURAIS ARTIFICIAIS

Tese submetida à banca examinadora
do Programa de Pós-Graduação em
Engenharia Elétrica da Universidade
Federal do Pará para a obtenção do
grau de Doutora em Engenharia
Elétrica.

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil
2016

Barros, Fabiola Graziela Noronha, 1979.

Metodologia para compressão de sinais de energia elétrica a partir de registros de forma de onda utilizando algoritmos genéticos e redes neurais artificiais / Fabiola Graziela Noronha Barros – 2016.

Orientador: Ubiratan Holanda Bezerra

Tese (Doutorado) – Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2016.

1. Sistemas de energia elétrica – processamento de dados. 2. Algoritmos genéticos. 3. Redes neurais (computação). I. Título.

CDD 23. ed. 621.3191

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**"METODOLOGIA PARA COMPRESSÃO DE SINAIS DE ENERGIA ELÉTRICA A PARTIR DE REGISTROS DE
FORMA DE ONDA UTILIZANDO ALGORÍTIMOS GENÉTICOS E REDES NEURAS ARTIFICIAIS"**

AUTORA: FABIOLA GRAZIELA NORONHA BARROS

TESE DE DOUTORADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE DOUTORA EM ENGENHARIA ELÉTRICA NA ÁREA DE SISTEMAS DE ENERGIA ELÉTRICA.

APROVADA EM: 16/12/2016

BANCA EXAMINADORA:



Prof. Dr. Ubiratan Holanda Bezerra

(Orientador – PPGEE/UFPA)



Prof. Dr. Marcus Vinícius Alves Nunes

(Co-Orientador – PPGEE/UFPA)



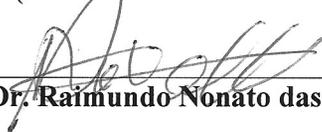
Prof. Dr. Evaldo Gonçalves Pelaes

(Avaliador Interno – PPGEE/UFPA)



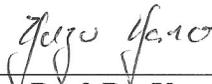
Prof. Dr. Roberto Célio Limão de Oliveira

(Avaliador Interno – PPGEE/UFPA)



Prof. Dr. Raimundo Nonato das Mercês Machado

(Avaliador Externo – IFPA)



Prof. Dr. Yuzo Iano

(Avaliador Externo – UNICAMP)

VISTO:

Prof. Dr. Evaldo Gonçalves Pelaes

(Coordenador do PPGEE/ITEC/UFPA)

AGRADECIMENTOS

Agradeço ao Prof. Dr. Ubiratan Holanda Bezerra pela coordenação dos diferentes aspectos tratados neste trabalho e pelo apoio e compreensão tão fundamentais quanto às orientações que me repassou ao longo de minhas atividades.

Agradeço ao Prof. Dr. Marcus Vinícius Alves Nunes pelo apoio que me foi dado desde o início de minhas atividades no doutorado. A autonomia e liberdade contínua que me conferiu foram importantes para o êxito deste trabalho.

Agradeço ao Professores Doutores Roberto Célio Limão de Oliveira (UFPA), Evaldo Gonçalves Pelaes (UFPA), Raimundo Nonato das Mercês Machado (IFPA), e Yuzo Iano (UNICAMP) por participarem das correções, contribuições e direcionamento deste trabalho. Isto conferiu qualidade e densidade no que se propôs alcançar.

Agradeço ao Prof. Dr. Wellington Alex dos Santos Fonseca (IFPA) pela orientação dada sempre que solicitado e pelas experiências repassadas sobre inteligência computacional aplicada à sistemas de energia elétrica. Suas contribuições sempre construtivas foram determinantes na consolidação deste trabalho.

Ao longo de minhas atividades de pesquisa sempre contei com o apoio do grupo CEAMAZON (Centro de Excelência e Eficiência Energética da Amazônia), portanto, agradeço aos colegas e professores pelas colaborações e incentivos.

Por fim, agradeço a FAPESPA - Fundação de Suporte a Pesquisa e Desenvolvimento Econômico do Estado do Pará, pelo apoio financeiro que tem dado aos pesquisadores do Estado do Pará em suas atividades e a todos os colegas que contribuíram direta ou indiretamente.

Fabiola Graziela Noronha Barros

EPÍGRAFE

“A mente que se abre a uma nova ideia jamais
voltará ao seu tamanho original”

Albert Einstein

SUMÁRIO

AGRADECIMENTOS	VI
EPÍGRAFE	VII
LISTA DE FIGURAS	XI
RESUMO	XIII
ABSTRACT	XIV
CAPÍTULO 1 – INTRODUÇÃO	1
1.1 INTRODUÇÃO	1
1.2 CONTRIBUIÇÕES DO TRABALHO	1
1.3 REVISÃO BIBLIOGRÁFICA	1
1.4 ESTRUTURA DO TRABALHO	8
CAPÍTULO 2 – OSCILOGRAFIA DIGITAL	9
2.1 INTRODUÇÃO	9
2.2 BREVE HISTÓRICO	9
2.3 OSCILÓGRAFOS	11
2.4 OSCILOGRAFIAS: QUANDO SÃO GERADAS	11
2.5 DURAÇÃO, CAPACIDADE E RESOLUÇÃO DA OSCILOGRAFIA	12
2.6 CONTEÚDO DA OSCILOGRAFIA	13
2.7 FORMATO DA OSCILOGRAFIA	13
2.7.1 <i>O padrão IEEE C37.111.1991</i>	14
2.7.2 <i>O padrão IEEE C37.111.1999</i>	15
2.8 CONSIDERAÇÕES FINAIS	16
CAPÍTULO 3 – ALGORITMOS GENÉTICOS	17
3.1 INTRODUÇÃO	17
3.2 BREVE HISTÓRICO	17
3.3 FUNDAMENTOS DOS ALGORITMOS GENÉTICOS	19
3.3.1 <i>Principais características dos algoritmos genéticos</i>	20
3.3.2 <i>Algoritmo de funcionamento de um AG</i>	20
3.3.3 <i>Representação cromossomial</i>	21
3.3.4 <i>População</i>	22

3.3.5	<i>Principais métodos de seleção de pais</i>	23
3.3.6	<i>Função de avaliação</i>	25
3.3.7	<i>Operador de crossover</i>	27
3.3.8	<i>Operador de mutação</i>	29
3.4	CONSIDERAÇÕES FINAIS	31
CAPÍTULO 4 – REDES NEURAS MULTICAMADAS		32
4.1	INTRODUÇÃO	32
4.2	BREVE HISTÓRICO	32
4.3	FUNDAMENTOS DAS REDES NEURAS ARTIFICIAIS	34
4.3.1	<i>Benefícios das redes neurais artificiais</i>	35
4.3.2	<i>Inteligência artificial e redes neurais</i>	36
4.3.3	<i>Processo de aprendizagem</i>	37
4.4	PERCEPTRON MULTICAMADAS - MLP	38
4.4.1	<i>Considerações preliminares</i>	40
4.4.2	<i>Modelo de um neurônio da rede neural perceptron multicamadas</i>	41
4.4.3	<i>Tipos de função de ativação para perceptron multicamadas</i>	43
4.4.4	<i>Algoritmo de retropropagação</i>	46
4.4.5	<i>Resumo do algoritmo de treinamento da rede neural Perceptron multicamadas (MLP)</i>	50
4.4.6	<i>Algoritmo de treinamento Levenberg-Maquardt</i>	51
4.4.7	<i>Resumo do algoritmo de treinamento Levenberg-Maquardt</i>	54
4.5	CONSIDERAÇÕES FINAIS	55
CAPÍTULO 5 – COMPRESSÃO DE DADOS		56
5.1	INTRODUÇÃO	56
5.2	BREVE HISTÓRICO	56
5.3	MÉTODOS DE COMPRESSÃO DE DADOS	57
5.3.1	<i>Compressão sem perdas e com perdas</i>	58
5.3.2	<i>Compressão simétrica e assimétrica</i>	61
5.3.3	<i>Compressão não-adaptativa e adaptativa</i>	61
5.4	CONSIDERAÇÕES FINAIS	62
CAPÍTULO 6 – COMPRESSÃO DE SINAIS DE ENERGIA ELÉTRICA UTILIZANDO ALGORITMOS GENÉTICOS E REDES NEURAS ARTIFICIAIS		63
6.1	INTRODUÇÃO	63
6.2	METODOLOGIA PROPOSTA	63

6.2.1	<i>Compressão de dados utilizando algoritmos genéticos e redes neurais artificiais</i>	65
6.2.2	<i>Representação cromossomial</i>	67
6.2.3	<i>Função objetivo</i>	68
6.2.4	<i>Compressão e descompressão do complemento de S por meio da rede neural artificial</i>	72
6.3	RESULTADOS	73
6.4	CONSIDERAÇÕES FINAIS	83
CAPÍTULO 7 – CONCLUSÃO		84
7.1	CONCLUSÕES	84
7.2	PROPOSTAS PARA TRABALHOS FUTUROS	84
REFERÊNCIAS BIBLIOGRÁFICAS		86
ANEXO 1		94
ANEXO 2		96

LISTA DE FIGURAS

FIGURA 2.1 – DESCRIÇÃO SIMPLIFICADA DE UMA REDE DE OSCILOGRAFIA (ADAPTADA DE SILVA ET. AL, 2013)....	12
FIGURA 3.1 – ALGORITMO DE FUNCIONAMENTO DE UM AG (ADAPTADA DE LINDEN, 2012).	21
FIGURA 3.2 – EXEMPLO DE ROLETA.....	24
FIGURA 3.3 – EXEMPLO DE APLICAÇÃO DO MÉTODO DO TORNEIO (ADAPTADA DE LINDEN, 2012).....	25
FIGURA 3.4 – EXEMPLOS DE PONTO DE CORTE.	28
FIGURA 3.5 – RESUMO ILUSTRATIVO DA OPERAÇÃO DE <i>CROSSOVER</i> DE UM PONTO.	28
FIGURA 3.6 – RESUMO ILUSTRATIVO DA OPERAÇÃO DE <i>CROSSOVER</i> DE DOIS PONTOS.	29
FIGURA 3.7 – FUNCIONAMENTO DO <i>CROSSOVER</i> UNIFORME.	29
FIGURA 3.8 – RESUMO DA OPERAÇÃO DO OPERADOR DE <i>CROSSOVER</i> DE UM PONTO E MUTAÇÃO (ADAPTADA DE LINDEN, 2012).	30
FIGURA 4.1 – GRAFO ARQUITETURAL DE UM <i>PERCEPTRON</i> MULTICAMADAS COM DUAS CAMADAS INTERMEDIÁRIAS (ADAPTADA DE HAYKIN, 2001).....	40
FIGURA 4.2 – MODELO DE UM NEURÔNIO ARTIFICIAL (ADAPTADA DE HAYKIN, 2011).	41
FIGURA 4.3 – FUNÇÃO DE LIMIAR.	44
FIGURA 4.4 – FUNÇÃO LINEAR POR PARTES.	45
FIGURA 4.5 – FUNÇÃO SIGMÓIDE PARA PARÂMETRO DE INCLINAÇÃO α VARIÁVEL.	46
FIGURA 6.1 – DIAGRAMA DE BLOCOS DA METODOLOGIA PARA COMPRESSÃO DE SINAIS DE ENERGIA ELÉTRICA A PARTIR DE ARQUIVOS COMTRADE.....	65
FIGURA 6.2 – DIAGRAMA DE BLOCOS DA METODOLOGIA PARA DESCOMPRESSÃO DE SINAIS DE ENERGIA ELÉTRICA A PARTIR DE ARQUIVOS COMTRADE.....	66
FIGURA 6.3 – REPRESENTAÇÃO GENÉTICA DO CROMOSSOMO.	67
FIGURA 6.4 – REPRESENTAÇÃO DA SELEÇÃO DE PONTOS UTILIZANDO ALGORITMOS GENÉTICOS.....	69
FIGURA 6.5 – REPRESENTAÇÃO LINEAR MAIS PRÓXIMA DO SINAL ORIGINAL UTILIZANDO ALGORITMOS GENÉTICOS.	69
FIGURA 6.6 – REPRESENTAÇÃO DAS ENTRADAS E SAÍDA DA REDE NEURAL ARTIFICIAL PARA A METODOLOGIA DE COMPRESSÃO.	72
FIGURA 6.7 – CURVA DE APTIDÃO DO MELHOR CROMOSSOMO PARA OS SINAIS DE TENSÃO E CORRENTE PARA AS FASES A, B E C, CONSIDERANDO $V_p = 2,5\%$	75
FIGURA 6.8 – CURVA DE APTIDÃO DO MELHOR CROMOSSOMO PARA OS SINAIS DE TENSÃO E CORRENTE PARA AS FASES A, B E C, CONSIDERANDO $V_p = 5\%$	75
FIGURA 6.9 – CURVA DE APTIDÃO DO MELHOR CROMOSSOMO PARA OS SINAIS DE TENSÃO E CORRENTE PARA AS FASES A, B E C, CONSIDERANDO $V_p = 10\%$	76
FIGURA 6.10 – SINAIS DE TENSÃO ORIGINAIS PARA AS FASES A (A), B (B) E C (C) RESPECTIVAMENTE.	77
FIGURA 6.11 – OS PONTOS DO CONJUNTO \mathcal{S} SELECIONADOS PELO AG PARA CADA SINAL DE TENSÃO NAS FASES A (A), B (B) E C (C) RESPECTIVAMENTE.....	78
FIGURA 6.12 – SINAIS DE TENSÃO RECONSTRUÍDOS PARA AS FASES A (A), B (B) E C (C) RESPECTIVAMENTE.	78
FIGURA 6.13 – SINAIS DE CORRENTE ORIGINAIS PARA AS FASES (A), B (B) E C (C) RESPECTIVAMENTE.....	79

FIGURA 6.14 – OS PONTOS DO CONJUNTO \mathcal{S} SELECIONADOS PELO $\mathcal{A}\mathcal{G}$ PARA CADA SINAL DE CORRENTE NAS FASES A (A), B (B) E C (C) RESPECTIVAMENTE.	79
FIGURA 6.15 – SINAIS DE CORRENTE RECONSTRUÍDOS PARA AS FASES A (A), B (B) E C (C) RESPECTIVAMENTE.	80

RESUMO

O presente trabalho propõe uma metodologia para compressão de sinais de energia elétrica a partir de registros de forma de onda em sistemas de energia, utilizando algoritmos genéticos (AG) e redes neurais artificiais (RNA). O algoritmo genético é utilizado para selecionar e preservar os pontos que melhor caracterizam os contornos da forma de onda; e a rede neural artificial é utilizada na compressão dos demais pontos bem como no processo de reconstrução do sinal. Assim, os dados resultantes são formados por uma parte do sinal original e pela parte complementar comprimida sob a forma de pesos sinápticos. A metodologia proposta seleciona e preserva um percentual de amostras do sinal original, que são aspectos não explorados na literatura. A metodologia foi testada usando dados reais obtidos a partir de um oscilógrafo instalado em um sistema de energia elétrica de 230 kV. Os resultados apresentam taxas de compressão que variam de 88,36% a 95,86% para taxas de preservação de pontos do sinal original que variam de 2,5% a 10% respectivamente.

PALAVRAS-CHAVES: Compressão de sinais de energia elétrica, algoritmos genéticos, redes neurais artificiais, inteligência computacional.

ABSTRACT

This thesis proposes a methodology for compression of electrical power signals from waveform records in electric systems, using genetic algorithm (GA) and artificial neural network (ANN). The genetic algorithm is used to select and preserve the points that better characterize the waveform contours; and the artificial neural network is used in the compression of other points as well as on the signal reconstruction process. Thus, the data resulting are formed by a part of the original signal and by a compressed complementary part in the form of synaptic weights. The proposed methodology selects and preserves a percentage of the original signal samples, which are aspects not explored in the literature. The method was tested using field data obtained from an oscillographic recorder installed in a 230kV electrical power system. The results presented compression rates ranging from 88.36% to 95.86% for preservation rates ranging from 2.5% to 10%, respectively.

KEYWORDS: *Compression of electrical power signals, genetic algorithms, artificial neural network, computational intelligence.*

CAPÍTULO 1 – INTRODUÇÃO

1.1 INTRODUÇÃO

A oscilografia é caracterizada pela medição de grandezas de tensão e corrente com alta resolução no domínio do tempo e cujo registro, feito por registradores digitais de perturbação (RDPs) ou unidades de proteção digital (UPDs), ocorre em consequência da detecção de uma perturbação (Tcheou, et al., 2006). É uma das formas mais utilizadas para documentar um evento, e desempenha um papel importante dentro dos centros de pós-operação, auxiliando no monitoramento dos sinais de tensão e corrente, sob análise em um sistema elétrico de potência (Nascimento, et al., 2007). O crescente aumento do número de registros oscilográficos gerados, tem motivado trabalhos de pesquisas na área de compressão de sinais de energia elétrica, com a finalidade de reduzir o tamanho desses arquivos e, por consequência, reduzir o tempo de transmissão dessas informações.

1.2 CONTRIBUIÇÕES DO TRABALHO

A metodologia desenvolvida utiliza algoritmos genéticos e redes neurais artificiais para compressão e descompressão de sinais de energia elétrica provenientes de registros de forma de onda em sistemas de potência. O objetivo na compressão desses sinais é reduzir o espaço necessário para armazenamento, bem como diminuir o tempo de transmissão através de canais de comunicação, sem que hajam perdas significativas de informações.

A metodologia apresenta uma estratégia inovadora ao utilizar os algoritmos genéticos e as redes neurais artificiais para compressão e descompressão de sinais de energia elétrica. Os algoritmos genéticos surgem como técnica de seleção dos pontos mais relevantes do sinal, preservando os principais contornos da forma de onda, e a rede neural, realiza a compressão do sinal original e atua como um interpolador para o processo de reconstrução do sinal comprimido. Neste sentido, destaca-se que a revisão bibliográfica, apresentada na próxima seção, não identificou trabalhos que fizessem uso de algoritmo genético e redes neurais artificiais para a compressão e descompressão de sinais de energia elétrica.

1.3 REVISÃO BIBLIOGRÁFICA

A compressão de dados é o ato de reduzir o espaço ocupado por dados em um determinado dispositivo, sendo realizada por meio de diversos algoritmos de compressão com o objetivo de reduzir a quantidade de *bytes* para representar um dado (Salomon, 2000).

Existem diversas formas de classificar os métodos de compressão de dados. Os mais conhecidos são pela ocorrência ou não de perdas de informação durante o processo de compressão.

Na compressão de dados com perdas, o algoritmo de compressão remove informações redundantes do sinal para atingir uma taxa de compressão mais elevada, enquanto que na compressão sem perdas, o algoritmo mantém a informação do sinal intacta (Gasparec, 2013). Desta forma, um método de compressão é um compromisso entre o volume de dados resultante da compressão e as informações que permaneceram (Nunez, et al., 2008) (Lorio, et al., 2004) (Wang, et al., 2005).

Nos últimos anos, os resultados apresentados na literatura científica mostram que os métodos de compressão mais utilizados para sinais de energia elétrica baseiam-se em transformadas *wavelets*.

Santoso et al. (1997) apresentaram uma técnica de compressão que utilizava transformada *wavelet* para os dados de distúrbios da qualidade de energia. A técnica de compressão era realizada por meio da decomposição do sinal, limiar de coeficientes de transformada *wavelet* e reconstrução do sinal. Foram determinados valores limites por meio da ponderação do valor máximo absoluto em cada escala. Coeficientes de transformada *wavelet* cujos valores estavam abaixo de um limiar, determinado pelos autores, foram descartados, enquanto que os que estavam acima deste limiar foram mantidos junto com as suas localizações temporais. Os autores mostraram a eficácia da técnica de compressão em dados reais. A taxa de compressão obtida nesta metodologia variou entre 66,66% (3,0:1,0) (arquivo comprimido três vezes menor que o arquivo original) e 82,76% (5,8:1,0) (arquivo comprimido cinco vírgula oito vezes menor que o arquivo original). Os autores concluíram que o custo relacionado ao armazenamento e transmissão de dados foram significativamente reduzido.

Hamid et al. (2002) apresentaram uma técnica de compressão de dados de distúrbios de qualidade de energia via transformada *wavelet* discreta (DWT) e transformada *wavelet packet* (WPT). A compressão de dados levou a uma aplicação em potencial para proteção remota e monitoramento da qualidade de energia. A técnica de compressão foi realizada por meio da decomposição do sinal até certo nível de detalhe, limiar de coeficientes *wavelet*, e reconstrução do sinal. A escolha de qual *wavelet* usar para a compressão foi de importância crítica, visto que a *wavelet* afeta a qualidade do sinal reconstruído e a concepção do sistema como um todo. Os autores propuseram utilizar a descrição do princípio do comprimento mínimo (MDL), como critério para a seleção de um filtro *wavelet* apropriado. Este critério permitiu a seleção não só do filtro *wavelet* adequado, mas também do melhor número de coeficientes *wavelet* para a reconstrução do sinal. O

trabalho experimental foi realizado para um único evento de falta monofásica, e os resultados obtidos de taxas de compressão foram de aproximadamente 88,88% (9,0:1,0).

Panda et al. (2002) apresentaram uma abordagem para a compressão de dados de qualidade de energia utilizando a transformada *wavelet* discreta (DWT) de *Slantlet* (STL). O desempenho do método foi avaliado em termos de taxa de compressão, porcentagem de energia retida e erro médio quadrático presente nos sinais reconstruídos. Diversos eventos de qualidade de energia, que incluem *sag*, *swell*, interrupção momentânea, harmônicos, transitórios e *flicker* foram utilizados para testar a metodologia. Os resultados das simulações indicaram que a transformada *wavelet* discreta de *Slantlet* ofereceu desempenho de compressão superior se comparada com as abordagens tradicionais utilizando a transformada discreta de cosseno e as abordagens baseadas em *wavelets* de tempo discreto. Os autores não informaram a taxa de compressão obtida para os diversos eventos, entretanto, eles afirmaram que os sinais (*sag*, *swell*, interrupção momentânea, harmônicos, transitórios e *flicker*) foram reconstruídos, respectivamente, com 93%; 95,44%; 94,11%; 93,14%; 95,04% e 95,18% de energia do sinal.

Lovisoló et al. (2003) utilizaram a técnica conhecida como *matching pursuits* (MP), oriunda da área de processamento de sinais, para análise de sinais provenientes de oscilografia, de forma a identificar os componentes transitórios que surgem durante as faltas em linhas de transmissão. Essa identificação dos componentes transitórios permitiu, aos autores, separar os fenômenos de interesse para análise, viabilizando a compactação dos arquivos de oscilografias. Um sinal pode ser representado por uma soma de componentes pré-definidas (chamadas estruturas), sendo que o conjunto de todas as possíveis estruturas é chamado de dicionário de dados; e que a priori, as funções que compõem o dicionário de dados são todas conhecidas (Lovisoló, et al., 2003). Os autores utilizaram um dicionário de dados para decompor o sinal; e dependendo da escolha do conjunto de estruturas do dicionário de dados, foi possível identificar os diferentes fenômenos que compõem o sinal, gerando uma representação compacta e adaptada ao sinal que está sendo analisado. No trabalho desenvolvido, os autores utilizaram a técnica de decomposição baseada em um dicionário de dados associada a um critério de parada para comprimir sinais. As taxas de compressão obtidas variaram em torno de 96,96% (33,0:1,0).

Lira (2004) desenvolveu um algoritmo de compressão e um sistema de classificação de sinais de tensão com distúrbio. A compressão do sinal foi realizada eliminando-se os coeficientes *wavelets* cujos módulos estão abaixo de um determinado limite de corte. Os resultados para sinais reais obtidos a partir de registradores digitais de perturbação forneceram altas taxas de compressão, geralmente em torno de 81%. Na classificação, o sinal de tensão foi pré-processado via *wavelets* e em seguida submetido a uma redução dimensional por meio da ferramenta estatística, Análises de Componentes Principais (ACP),

e então, submetido à rede neural MLP (*Multilayer Perceptron*), que indicou o tipo de distúrbio presente no sinal. Cada rede implementada foi treinada com uma base de conhecimento, cujos atributos foram constituídos a partir dos coeficientes *wavelets* de aproximação, ou de níveis de detalhes, ou de ambos. Na combinação das redes neurais, em cada um dos seis nós de saída, aplicou-se a média entre as três saídas das redes individuais. A decisão final do classificador corresponde à saída combinada de maior valor. A técnica de combinação de modelos diferentes na classificação mostra bons resultados ao corrigir os casos mal classificados pelas redes individualmente. O percentual na classificação do tipo de evento, e de acerto na combinação da rede treinada com os coeficientes de detalhes com àquela treinada com os coeficientes de aproximação para um conjunto de teste foi de 99,3%, enquanto que na rede individual treinada com ambos coeficientes, esse índice foi de 96,4%.

Ribeiro et al. (2004) apresentaram uma técnica de codificação de forma de onda, utilizando transformada *wavelet*, para fins de monitoramento de qualidade de energia. O método de compressão de dados proposto apresentou uma abordagem de processamento de sinal adaptativo para estimar a componente fundamental senoidal e separá-la dos transientes do sinal monitorado. Quando esses componentes não estacionários foram submetidos à técnica de compressão, a propriedade de representação esparsa da transformada *wavelet* forneceu uma melhora na taxa de compressão. Além disso, a degradação inserida pelas perdas durante o processo de compressão foi minimizada. Os resultados das simulações confirmaram a eficácia do método proposto, com taxas de compressão em torno de 93,42%.

Lorio et al. (2004) estudaram diferentes métodos de compressão de dados e aplicaram alguns desses métodos na compressão de sinais de qualidade de energia elétrica. Os autores fizeram um apanhado geral dos métodos estudados: métodos de compressão sem perdas (método de Huffman, codificação aritmética e algoritmo LZW), compressão via transformadas *wavelets* e, algoritmo EZW (*embedded zero-tree wavelet*). Os autores utilizaram os métodos de compressão citados para compressão de sinais de cintilação, harmônicos, *sag*, *swell* e transitórios impulsivos, e compararam os métodos em termos de taxa de compressão e erro médio quadrático. Os resultados mostraram que a taxa de compressão e o erro médio quadrático do sinal reconstruído variaram com o tipo de sinal. Dentre os métodos sem perdas, a codificação aritmética mostrou-se o método mais adequado, com taxa de compressão de aproximadamente 50%. Utilizando a transformada *wavelet* em associação aos métodos sem perdas a taxa de compressão aumentou, variando de 66,67% a 75%, sendo que essa variação foi devida ao número de escalas utilizadas para a reconstrução do sinal. O método EZW apresentou os melhores resultados, com taxa de compressão de 90%.

Wang et al. (2005) propuseram um método de compressão e reconstrução de sinais de distúrbios de qualidade de energia baseado em transformadas *wavelets*. O sinal foi decomposto por meio da análise multiresolução, guardando a informação (magnitude e posição) dos coeficientes escala para posterior utilização. Os coeficientes provenientes da análise multiresolução são codificados utilizando codificação aritmética. A decodificação e a reconstrução do sinal utilizaram a transformada *wavelet* inversa de acordo com a escala e os coeficientes *wavelets*. Os autores simularam quatro sinais típicos de qualidade de energia no *Alternative Transient Program* (ATP): *sag*, *swell*, interrupção e transitórios impulsivos. As taxas de compressão obtidas para os quatro tipos de eventos foram respectivamente: 1,07:1,00; 1,07:1,00; 1,07:1,00; e 1,07:1,00, ou seja 6%. Pelo método proposto, não houveram variações significativas quanto a taxa de compressão para os diferentes tipos de sinais.

Tcheou et al. (2006) propuseram, em seu trabalho, avaliar um método de compressão de oscilografias, por meio de duas abordagens: com base no conhecimento de especialistas em análise de perturbações elétricas; e aplicando os arquivos comprimidos a uma ferramenta de localização de faltas. O método de compressão analisado foi o método desenvolvido por Lovisolo et al. (2003), que utilizam a técnica *matching pursuits*. Os autores consideraram que avaliar o método de compressão era importante por três fatores: redução do espaço necessário para armazenamento das oscilografias, facilitação da transmissão desses arquivos, e para análise do desempenho da proteção. Os autores analisaram somente eventos de curto-circuito, sendo que 60% dos arquivos avaliados (originais e comprimidos) eram provenientes de oscilografias (faltas reais), e o restante dos arquivos analisados (40%) eram provenientes de faltas simuladas no ATP. Foram analisadas 21 oscilografias, entre arquivos pré-compressão e pós-compressão, tanto por especialistas quanto pela ferramenta de localização de faltas. Na avaliação feita por especialistas, eles deveriam identificar o tipo de falta (fase-terra, bifásico-terra ou trifásico), verificar as condições de pré-falta (cálculo dos fasores), identificar os tempos de início e fim da falta. Para a análise com taxa de compressão de 91% e de 96%, os especialistas classificaram corretamente o tipo de falta, entretanto, na condição pré-falta, os arquivos com taxa de compressão maiores foram classificados corretamente em 85,7% dos casos. Na análise por ferramenta de localização de faltas, o erro de localização da falta ficou em torno de 1% para a menor taxa de compressão, e em torno de 6% para a maior taxa de compressão.

Nascimento et al. (2007) propuseram uma técnica híbrida para compressão de sinais para oscilografia digital, utilizando um algoritmo que se baseia na codificação por transformadas (transformada discreta de *Hartley* (DHT); transformada discreta de cosseno (DCT); e transformada ortogonal por superposição (LOT)), aplicado à forma de onda resultante do processo de aquisição digital. O algoritmo selecionava os coeficientes mais

significativos, dentre os resultantes da aplicação da transformada, impondo uma relação sinal-ruído mínima para cada segmento do sinal em análise. Os autores justificaram o uso das transformadas que usam bases senoidais ortogonais, para avaliar o desempenho do algoritmo de compressão, pelo fato de que os sinais analisados e as transformadas terem características oscilatórias similares. De modo geral, considerando a taxa de compressão, os melhores valores foram obtidos quando se utilizou a transformada discreta de *Hartley*, com taxa de compressão aproximadamente igual a 90,9%.

Gasparese (2010) desenvolveu uma técnica de compressão de dados com base em decomposição *wavelet* e interpolação *spline*, para distúrbios de qualidade de energia. A técnica consistia na decomposição do sinal, limiar de coeficientes da transformada *wavelet*, decimação do último coeficiente e reconstrução do sinal usando a interpolação *spline* para o último coeficiente. O autor apresentou a ordem da *wavelets* de Daubechies que melhor se aplicava a sua técnica, a fim de atingir a melhor taxa de compressão, ficando em torno de 62,96%, e os resultados comparativos de compressão, utilizando diferentes tipos de distúrbios de qualidade de energia. O autor indicou a técnica principalmente para dados adquiridos com altas taxas de amostragem.

Zhang et al. (2011) apresentaram um método de compressão para sinais de qualidade de energia utilizando transformada de Fourier e transformada *wavelet*. O método proposto fez uso do algoritmo de interpolação da transformada de Fourier discreta para estimação de parâmetros (amplitude, frequência e fase) da componente fundamental e harmônicos e para separar os transientes do sinal de qualidade de energia original. A compressão foi feita por meio da quantização dos parâmetros da componente fundamental e harmônicos utilizando transformada de Fourier. Apenas os componentes transitórios do sinal foram comprimidos por meio de técnicas de codificação via transformada *wavelet*. Os autores afirmaram que o método proposto é adequado para vários tipos de sinais de qualidade de energia (harmônicos, transitório oscilatório e *sag*), e que a taxa de compressão para esses sinais variou em torno de 98,31%; 97,69% e 97,59% respectivamente.

Angarita (2013), em seu trabalho, abordou o estado da arte na compressão de sinais, com o objetivo de propor novas técnicas que viabilizassem a compressão das formas de onda de distúrbios. O autor apresentou três técnicas híbridas de compressão de dados que combinam algoritmos de compressão de dados com perdas e sem perdas, focando na estratégia de alocação dinâmica de *bits* no domínio da transformada. A primeira técnica proposta denominada codificação por transformadas com alocação dinâmica e inteligente de *bits* no espaço transformado (CTIA) executou a alocação de *bits* segundo um conjunto de padrões vetoriais característico de distribuição de energia. A segunda técnica proposta denominada codificação por transformada com curvas de perfil espectral para alocação dinâmica de *bits* no espaço transformado (CTCA) utilizou curvas de padrões de estimação

da distribuição de energia. A terceira técnica proposta denominada codificação por transformadas com abordagem dissociativa entre regime permanente e eventos transitórios (CTAD) identificou e desagregou as características comportamentais do regime permanente e do estado transitório para a alocação de *bits*. O autor concluiu que os resultados de simulações com banco de sinais reais mostram um bom desempenho dos algoritmos propostos no que tange a taxa de compressão de dados e a fidelidade da forma de onda reconstruída.

Tcheou et al. (2014) publicaram um artigo com um levantamento sobre as principais técnicas de compressão de sinais de energia elétrica desenvolvidas nos últimos anos. Neste trabalho, os autores destacaram a importância das técnicas de compressão quando aplicadas a formas de ondas de sinais de energia elétrica em *smart grids*. Vale ressaltar que nenhuma das técnicas de compressão apresentadas neste artigo utilizam inteligência computacional para compressão e descompressão dos sinais de energia elétrica.

A Tabela 1 resume as técnicas de compressão apresentadas nesta seção.

Tabela 1 – Resumo das técnicas de compressão apresentadas na revisão bibliográfica.

Técnica	Autor	Ano	Evento
	Santoso	1997	VTCDs
	Hamid	2002	falta monofásica
	Panda	2002	flicker, harmônicos, interrupção momentânea, <i>sag</i> , <i>swell</i> , e transitórios
Transformada wavelet: <i>discreta</i> , <i>packet</i> , <i>Slantlet</i> , e <i>Daubechies</i>	Lira*	2004	afundamentos e elevação de tensão, harmônicos, interrupção, e transitórios oscilatórios
	Ribeiro	2004	transitórios impulsivo e oscilatório
	Wang	2005	interrupção, <i>sag</i> , <i>swell</i> , e transitórios impulsivos
	Gasparesc	2010	transitórios impulsivo e oscilatórios, e <i>sag</i>
Dicionário de dados	Lovisolo	2003	falta monofásica, bifásica e trifásica
	Tcheou	2006	falta monofásica, bifásica e trifásica
<i>Estudo de diversas técnicas:</i>			
Método de Huffman, codificação aritmética, e algoritmo LZW	Lorio	2004	centilação, harmônicos, <i>sag</i> , <i>swell</i> e transitórios impulsivos
Transformada <i>wavelet</i> , e Algoritmo EZW			
<i>Técnica híbrida:</i>			
	Nascimento	2007	faltas trifásicas
T. discreta <i>Hartley</i> ,			

T. discreta de cosseno, e			
T. ortogonal por superposição			
Transformada de Fourier: <i>discreta</i>	Zhang	2011	harmônicos, sag, e transitórios oscilatórios
	Angarita	2013	harmônicos, e transitórios
Levantamento do estado da arte das principais para compressão de sinais de energia elétrica	Tcheou	2014	aplicações em <i>smart grids</i>

1.4 ESTRUTURA DO TRABALHO

O **Capítulo 2** apresenta um breve levantamento histórico da oscilografia junto a análise de faltas no Sistema Elétrico Brasileiro (SEB), bem como traz uma visão geral sobre a oscilografia digital, abordando suas principais características, e o oscilógrafo.

No **Capítulo 3** é apresentada uma breve revisão sobre os algoritmos genéticos, o que inclui um breve histórico sobre o seu surgimento, os seus conceitos básicos e características, tais como: representação cromossomial, população inicial, métodos de seleção, operador de *crossover* e mutação, função de avaliação, processamento paralelo e esquemas.

No **Capítulo 4** é apresentada uma breve revisão sobre as redes neurais artificiais, incluindo um histórico sobre as redes, bem como seus fundamentos, o que inclui seus benefícios, e o processo de aprendizagem; também apresenta as principais características da rede neural *Perceptron* multicamadas, além dos algoritmos de treinamento de retropropagação do erro e algoritmo Levenberg-Maquardt.

O **Capítulo 5** aborda os principais conceitos sobre compressão de dados, apresentando os principais tipos de compressão: com perdas e sem perdas; simétrica e assimétrica; e adaptativa e não-adaptativa.

O **Capítulo 6** descreve a metodologia desenvolvida, utilizando inteligência computacional, para compressão e descompressão de sinais de energia elétrica a partir de registros de forma de onda. Neste capítulo também são apresentados os resultados da metodologia.

O **Capítulo 7** apresenta a conclusão deste trabalho e as propostas para trabalhos futuros.

CAPÍTULO 2 – OSCILOGRAFIA DIGITAL

2.1 INTRODUÇÃO

A oscilografia é um registro das grandezas elétricas analógicas e digitais do sistema elétrico de potência, armazenado de forma automática ou manual nas unidades de proteção digital ou nos registradores digitais de perturbação. No registro manual, o processo de gravação é iniciado por comando do usuário e normalmente serve para analisar as condições operativas do sistema elétrico em um instante de interesse. As gravações automáticas ocorrem por programações efetuadas pelo usuário. Neste caso, o processo é iniciado por meio de mudanças nos valores analógicos e nos estados digitais em consequência de disparos de proteção ou variações de grandezas elétricas (Cardoso, 2011).

Durante as perturbações no sistema elétrico, os RDPs ou as UPDs gravam os registros das variações ocorridas em memórias internas não voláteis. Os registros armazenados em arquivos possibilitam analisar a operação do relé, identificar o tipo de defeito, as fases que foram envolvidas, definir os tempos da ocorrência de mudanças de estado ou valores analógicos, localizar o ponto de ocorrência da falta, analisar a coordenação da proteção, entre outros aspectos (Cardoso, 2011).

2.2 BREVE HISTÓRICO

Nos anos de 1950 iniciou-se uma coleta de registro de falhas em sistemas elétricos de potência por meio de aparelhos eletromecânicos a tinta. Nas décadas seguintes de 1960 e 1970, surgiu o sistema interligado brasileiro, e nesse mesmo período, surgiram os oscilógrafos analógicos. Esses oscilógrafos eram equipamentos eletromecânicos que desenhavam as oscilações induzidas pelos fenômenos eletromagnéticos, por meio de uma agulha móvel acoplada a um rolo de papel fotossensível em movimento, os quais, eram utilizados geralmente em sistemas elétricos prioritários (alta e extra alta tensão) por serem bastante onerosos. A partir deste momento, os registros gráficos passaram a ser considerados documentos e material de estudo para os analistas das operadoras, transmissoras e concessionárias do setor elétrico (Cormane, 2013). Na época, as oscilografias eram coletadas localmente e enviadas por meio de transporte convencional ou por *fax* para os centros de análise. As limitações tecnológicas da época e os altos custos associados à aquisição, operação e manutenção dos oscilógrafos analógicos, faziam do sistema de oscilografia um tópico importante no orçamento das empresas, isso sem considerar os custos com a manutenção dos registros gráficos, que se degradavam e conseqüentemente perdiam-se os dados (Tcheou, 2005).

Os sistemas isolados que atendiam as cidades maiores ou regiões metropolitanas foram aos poucos sendo interligados. À medida que estes sistemas cresciam e a tecnologia evoluía, surgiu naturalmente a conveniência e a necessidade da coordenação entre as diversas empresas. Esta coordenação não foi imposta a partir de um órgão regulador central, mas sim combinada pela conversa e entendimentos entre os seus dirigentes. A entidade informal criada para esse fim foi o Comitê Coordenador da Operação Interligada (CCOI) (Cardoso, 2012).

No começo, haviam comitês separados para as regiões sul e sudeste, até que o fortalecimento das interligações justificou a unificação. Paralelamente, um pouco mais tarde, foi criado o Comitê Coordenador da Operação do Nordeste (CCON). Nesses comitês eram discutidas e estabelecidas as regras básicas da operação interligada. Na área de manutenção era intensa a troca de experiências e estabelecimento de algumas normas de interesse comum (Cardoso, 2012).

Em 1973 foi publicada a chamada Lei de Itaipu, que formalizou a criação de Grupos de Coordenação para Operação Interligada (GCOI). O GCOI iniciou a análise de faltas no Sistema Elétrico Brasileiro (SEB). Alguns anos depois, foi criado o Grupo Coordenador do Planejamento dos Sistemas Elétricos (GCPS) (Cardoso, 2012).

Na metade da década de 1980, apareceram os oscilógrafos digitais, chamados de registradores digitais de perturbação, como ferramenta auxiliar na análise de faltas no sistema elétrico brasileiro; que se consolidaram, efetivamente, durante os anos de 1990.

Em meados da década de 1990 foi iniciada uma profunda transformação no modelo do setor elétrico brasileiro. O GCOI e o GCPS foram extintos. Foi criada a Agência Nacional de Energia Elétrica (ANEEL), e também foi criado o Operador Nacional do Sistema (ONS), que absorveu muitas das atribuições que eram do GCOI (Cardoso, 2012).

A partir da década de 1980, com a evolução tecnológica, o barateamento do *hardware* e ampliação do mercado, os recursos de oscilografias foram incorporados em grande parte das unidades de proteção digital utilizados em sistemas de proteção (Tcheou, 2005).

Com as transformações no cenário do setor elétrico mundial, que contemplam padrões de funcionamento com maiores níveis de exigência, tornou-se importante o monitoramento e a documentação das ocorrências como uma forma de melhorar a operação do sistema, e garantir a competitividade entre as empresas do setor elétrico (Li, et al., 2010). Neste contexto, as informações obtidas pelos equipamentos de medição passaram a ocupar um papel de destaque.

2.3 OSCILÓGRAFOS

Os oscilógrafos são utilizados para analisar fenômenos do sistema elétrico. Nas subestações automatizadas eles registram a atuação de relés de proteção, a abertura e o fechamento de disjuntores e chaves seccionadoras e outras indicações de estado de interesse, possibilitando o encadeamento histórico das ocorrências. Devido à elevada precisão, a aquisição desses dados é feita normalmente por equipamentos autônomos, os quais se comunicam com o centro de controle e demandam um dispositivo de sincronização de tempo (GPS – Sistema de Posicionamento Global) (Mello, 2006).

Os registradores digitais de perturbação podem ser utilizados como registradores tanto de eventos de curta duração do tipo curto-circuito e energização, como também para fenômenos de longa duração como oscilações e afundamentos de tensão, atendendo, desta maneira, tanto o setor de transmissão e distribuição quanto o setor de geração (Reason, 2006).

São armazenados três tipos diferentes de registros: os registros de forma de onda, os registros fasoriais e os registros de medição contínua. Os registros de forma de onda são os mais conhecidos, onde é possível analisar a própria forma de onda, e também realizar a análise harmônica do sinal. Os registros fasoriais são os menos conhecidos, entretanto, são muito úteis em usinas e para análise de fenômenos mais lentos, da ordem de minutos. Já os registros de medição contínua permitem acompanhar os níveis de corrente e tensão, assim como grandezas derivadas durante o dia (Reason, 2006).

2.4 OSCILOGRAFIAS: QUANDO SÃO GERADAS

As oscilografias são armazenadas pelo RDP ou pela UPD nas seguintes situações (Cardoso, 2011):

- Por disparo da proteção (*trip*);
- Por partida de registro programado; e
- Por partida de registro manual.

O **disparo da proteção** se refere à atuação do relé ou *trip*. Neste caso, a oscilografia é armazenada automaticamente sempre que for sensibilizada uma das funções de proteção, que esteja previamente programada para realizar a partida do registro oscilográfico. Este tipo de disparo está vinculado à lógica de *trip*, e as funções ali parametrizadas serão responsáveis por disparar a gravação do arquivo.

A **partida de registro programado** é um parâmetro ajustável que permite ao usuário escolher condições diversas para iniciar o armazenamento do registro de uma oscilografia. Pode-se programar a partida de uma função de sobrecorrente (*pickup*), o alarme de uma

função de desbalanço, o alarme por temperatura elevada, a energização de um transformador, entre outros eventos.

A **partida de registro manual** é um comando executado pelo operador que fará o RDP ou a UPD armazenar a oscilografia das condições atuais do sistema. Esse disparo é realizado, por exemplo, quando se deseja observar a diferença angular dos fasores de corrente e tensão, analisar inversões na conexão dos transformadores de corrente e investigar componentes harmônicas.

A Figura 2.1, mostra a descrição simplificada de uma rede de oscilografia. A rede de oscilografia possui um sistema de verificação e coleta automática das informações registradas pelos RDPs. Essas informações são organizadas em registros padronizados para envio ao servidor central da rede, onde são armazenadas e posteriormente compartilhadas pelos profissionais da empresa envolvidos com a atividade de análise (Silva, et al., 2007).

Figura 2.1 – Descrição simplificada de uma rede de oscilografia (adaptada de Silva et. al, 2013).



As informações decorrentes dos sistemas de oscilografia constituem o histórico das operações do sistema elétrico de potência, e são úteis na análise de equipamentos e do desempenho do sistema de proteção, na localização de faltas, na avaliação das ocorrências e dos distúrbios de qualidade de energia, além de auxiliar nos processos que visem a inserção de responsabilidade da empresa por eventuais ocorrências (Moreto, et al., 2010).

2.5 DURAÇÃO, CAPACIDADE E RESOLUÇÃO DA OSCILOGRAFIA

A duração da oscilografia corresponde ao tamanho do registro, em uma janela de tempo, armazenado no arquivo pelo RDP ou pela UPD. O tamanho é medido em número de ciclos da forma de onda que serão gravados em cada evento. É um parâmetro ajustável, com diferenças conforme o modelo do RDP ou da UPD. Uma parcela desse tempo é reservada para o tempo de registro de pré-falta que é também configurável (Cardoso, 2011).

A capacidade de armazenamento total, ou seja, a memória para gravar as oscilografias é fixa para cada equipamento. Quanto maior for o ajuste da duração, menor

será o número de eventos gravados. Quando a capacidade máxima for atingida o RDP ou a UPD irá apagar o evento mais antigo para gravar o evento mais novo (Cardoso, 2011).

A resolução da oscilografia corresponde ao número de amostras por ciclo. Este parâmetro pode ser fixo ou ajustável em alguns RDPs e UPDs.

2.6 CONTEÚDO DA OSCILOGRAFIA

Os dados incorporados em uma oscilografia são (Cardoso, 2011):

- As grandezas analógicas de tensão e corrente;
- Os estados digitais dos elementos de proteção e controle e entradas/saídas digitais;
- O sumário do evento (data, hora, tipo de falta, entre outros); e
- Os parâmetros de ajuste do RDP ou da UPD.

2.7 FORMATO DA OSCILOGRAFIA

Os dados das oscilografias estão no formato comprimido com extensão *.CEV, que é um tipo de arquivo de codificação de caractere, ou no formato **Common Format for Transient Data Exchange** (COMTRADE) com as extensões *.HDR, *.CFG, *.DAT e *.INF, sendo este último (*.INF) adicionado a partir da revisão de 1999. As extensões significam:

- HDR: para cabeçalho (*header*);
- CFG: para configuração (*configuration*);
- DAT: para dados (*data*); e
- INF: para informação complementar (*information*).

O formato COMTRADE poderá ser útil se o usuário desejar reproduzir novamente as formas de onda no equipamento. Isto é possível com a utilização de modernas caixas de injeção de tensão e corrente disponíveis no mercado, as quais reconhecem os sinais da oscilografia COMTRADE e reproduzem os mesmos sinais analógicos em seus terminais (Cardoso, 2011).

O COMTRADE define um formato comum para arquivos de dados digitais e mídias, necessários para troca de vários tipos de dados de perturbações, ensaios e simulação (IEEE Std C37.111.1991).

Os arquivos definidos pelo COMTRADE devem estar, a princípio, em formato ASCII. Os arquivos estão no formato "xxxxxxx.yyy" onde "xxxxxxx" é usado para identificar o evento e ".yyy" é usado para identificar o tipo do arquivo (IEEE Std C37.111.1991).

O arquivo de cabeçalho (*.HDR) é um arquivo de texto, opcional, no formato ASCII criado pelo originador dos dados no formato COMTRADE; geralmente por meio do uso de um programa destinado a processar palavras. Este arquivo deve ser lido ou impresso pelo

usuário. O criador do arquivo de cabeçalho pode incluir qualquer informação em qualquer ordem desejada (IEEE Std C37.111.1991).

Os arquivos de configuração (*.CFG) devem ser lidos por um programa de computador e seu formato, definido na norma, deve ser obedecido à risca. Possui informações que permitirão interpretar corretamente os dados. Estas informações incluem itens como: taxas de amostragem, número de canais, frequência da linha, informação dos canais, entre outros (IEEE Std C37.111.1991).

Os arquivos de dados (*.DAT) contém o valor da amostra de cada canal medido. O número guardado no arquivo será em formato inteiro, normalmente correspondente ao formato do equipamento digitalizador. No arquivo de configuração são especificados meios de traduzir estes valores adequadamente para grandezas reais, tais como *volts* e *amperes*. Dados digitais são representados pelos algarismos "0" e "1". O arquivo de dados pode estar em formato binário desde que se forneça juntamente um programa que o traduza para formato ASCII (IEEE Std C37.111.1991).

Um novo e opcional arquivo de informação (*.INF) foi adicionado para prover a transmissão de informação pública e privada em formas legíveis por computador. Este arquivo opcional contém informações extras que os criadores de arquivos podem disponibilizar para os usuários, em adição às informações mínimas do conjunto de dados requeridas para a aplicação. O formato provê para informação pública que qualquer usuário possa ler e usar, e para a informação privada que pode ser acessível somente para usuários de uma classe particular ou fabricante.

2.7.1 O padrão IEEE C37.111.1991

A rápida evolução e implementação de dispositivos para testes e gravação de dados de transitórios e faltas nas indústrias de energia elétrica gerou a necessidade de um formato padrão para a troca de dados, feita por usuários, com vários dispositivos de forma a melhorar e automatizar as análises, testes, avaliações, e simulações do sistema de energia e das condições de faltas e de distúrbios relacionados aos esquemas de proteção (IEEE Std C37.111.1999). Foi a partir desta necessidade que se elaborou os padrões IEEE C37.111.1991 e IEEE C37.111.1999, com o objetivo de definir um formato comum para arquivos de dados.

O formato do arquivo de configurações (*.CFG) definido pelo padrão IEEE C37.111.1991 pode ser visualizado no ANEXO 1.

2.7.2 O padrão IEEE C37.111.1999

A revisão de 1999 no padrão IEEE C37.111.1991 acrescentou novas informações ao formato COMTRADE. Estas novas informações caracterizam algumas diferenças que estão resumidas a seguir:

O arquivo de cabeçalho (*.HDR) é explicitamente definido como opcional.

O arquivo de configuração (*.CFG) foi modificado. Um campo contendo o ano de revisão do padrão COMTRADE foi adicionado para distinguir arquivos feitos sobre esta ou futuras revisões dos padrões. Se este campo não está presente, o formato do arquivo é definido pelo padrão C37.111.1991.

Um campo para um fator de multiplicação de amostra de tempo foi adicionado para atender à necessidade para arquivos com longas durações. Para ajudar na conversão dos dados, três novos campos de escala (primário, secundário e primário/secundário) foram adicionados para definir a razão de transformação do equipamento e se os dados registrados estão em escala para refletir valores primários ou secundários.

Campos de configuração para informações do canal de estados (canais digitais) foram expandidos para cinco campos para permitir o mesmo nível de definição dos canais analógicos.

A frequência da linha é definida como um ponto flutuante.

Suporte para dados do evento de disparo foram adicionados para informação da taxa de amostragem quando esta é variável.

O formato das amostras tempo/dados foi modificado com o dia do mês precedendo o campo de entrada do mês, e o campo de ano agora tem quatro dígitos do ano.

Foi adicionada uma exigência de que pelo menos um espaço será permitido nos campos de dados, incluindo aqueles campos para os quais nenhum dado está disponível (previamente especificado com delimitadores de vírgula (,) com nenhum espaço entre os delimitadores).

O formato do arquivo de configurações (*.CFG), definido pelo padrão IEEE C37.111.1999, pode ser visualizado no ANEXO 2.

Um novo formato para um arquivo de dados binário foi especificado e a exigência de um programa de conversão fornecido ao usuário foi eliminada.

Todas as descrições de campo são explicitamente definidas com respeito ao: estado crítico, formato, tipo, comprimento máximo/mínimo e valor máximo/mínimo.

2.8 CONSIDERAÇÕES FINAIS

Este capítulo mostrou que os registros de oscilografias, tanto dos registradores digitais de perturbação quanto das unidades de proteção digital, oferecem a oportunidade de estudar e compreender as ocorrências no sistema elétrico de potência devido à natureza das informações que neles se encontram armazenadas. Foram apresentadas a definição de oscilografia e um apanhado geral sobre os oscilógrafos e suas características, tais como o tipo de formato de armazenamento dos dados oscilográficos, que pode ser COMTRADE, utilizando o padrão IEEE C37.111 dos anos de 1991 e 1999.

CAPÍTULO 3 – ALGORITMOS GENÉTICOS

3.1 INTRODUÇÃO

Na metodologia desenvolvida, os algoritmos genéticos são usados para selecionar os pontos que melhor caracterizam os contornos dos sinais de tensão e corrente, registrados durante a ocorrência de um evento nos sistemas elétricos de potência. Desta forma, para o melhor entendimento da metodologia, o Capítulo 3 é dedicado a uma breve revisão sobre os algoritmos genéticos, o que inclui um breve histórico sobre o seu surgimento, os seus conceitos básicos e características, tais como: representação cromossomial, população inicial, métodos de seleção, função de avaliação, operador de *crossover* e mutação.

3.2 BREVE HISTÓRICO

A história dos algoritmos genéticos se inicia na década de 1940, quando cientistas começaram a se inspirar na natureza para criar o ramo da inteligência artificial. Neste período, as pesquisas se desenvolveram mais nos ramos da pesquisa cognitiva e na compreensão dos processos de raciocínio e aprendizado. A partir do final da década de 1950 os estudos passaram a buscar modelos de sistemas genéricos que pudessem gerar soluções candidatas para problemas que eram complexos demais para serem resolvidos computacionalmente (Linden, 2012) (Engelbrecht, 2007) (Ahn, 2006) (Mitchell, 1998).

Uma das primeiras tentativas de se associar a evolução natural aos problemas de otimização foi feita em 1957, quando Box (Box, 1957) apresentou seu esquema de operações evolucionárias; análoga ao que se conhece hoje por seleção e mutação (Goldberg, et al., 1988). Logo depois, no começo da década de 1960, Bledsoe (Bledsoe, 1961) e Bremermann (Bremermann, 1962) começaram a trabalhar com genes, usando tanto a representação binária quanto a representação inteira e a representação real, desenvolvendo os precursores dos operadores de recombinação (*crossover*) (Linden, 2012).

Uma tentativa de usar processos evolutivos para resolver problemas foi desenvolvida por Rechenberg, na primeira metade da década de 1960, quando este desenvolveu as estratégias evolucionárias (Rechenberg, 1973). Estas estratégias mantinham uma população de dois indivíduos com cromossomos compostos de números reais em cada instante, sendo que um dos dois cromossomos era filho do outro e era gerado por meio da aplicação exclusiva do operador de mutação. O processo descrito por Rechenberg tinha ampla fundamentação teórica, a mutação era aplicada a partir de uma distribuição gaussiana dos parâmetros, sendo usado com sucesso em vários problemas práticos.

Mesmo não incluindo conceitos amplamente aceitos atualmente, tais como maior número de cromossomos e menor taxa de mutação, o trabalho de Rechenberg pode ser considerado pioneiro, por ter introduzido a computação evolucionária em aplicações práticas (Linden, 2012).

Em trabalhos posteriores as estratégias evolucionárias supriram estas falhas, sendo modificadas para incluir conceitos de população e operador de *crossover*. A maneira como estas estratégias aplicaram este operador é interessante, pois incluíram a ideia de utilizar a média como operador, além de poderem envolver muitos pais.

Apesar de não ser o primeiro investigador da área, aquele que seria designado o pai dos algoritmos genéticos mostrou-se finalmente no final da década de 1960, quando John Holland “inventa” os algoritmos genéticos, embora concentrado eminentemente na codificação discreta (Holland, 1975).

Holland estudou formalmente a evolução das espécies e propôs um modelo heurístico computacional que quando implementado poderia oferecer boas soluções para problemas extremamente difíceis, que eram insolúveis de resolver computacionalmente até aquela época. Em 1975, Holland (Holland, 1975) publicou seu livro “*Adaptation in natural and artificial systems*”, no qual faz um estudo dos processos evolutivos. O trabalho de Holland apresentou os algoritmos genéticos como uma metáfora dos processos evolutivos, de forma que ele pudesse estudar a adaptação e a evolução no mundo real, simulando-a dentro de computadores. Um fato interessante quanto ao trabalho de Holland e sua influência na área de AG é que ele usou originalmente cromossomos binários, cujos genes eram apenas 0s e 1s (Linden, 2012).

Desde então, os algoritmos genéticos começaram a se expandir por toda a comunidade científica, gerando uma série de aplicações que puderam ajudar a resolver problemas extremamente importantes que talvez não fossem abordados de outra maneira. Além desse progresso científico, também houve o desenvolvimento comercial; nos anos de 1980 surgiram pacotes comerciais utilizando os algoritmos genéticos (Linden, 2012).

Na década de 1980, o progresso dos algoritmos evolucionários e sua popularização no meio científico fizeram com que surgissem as primeiras conferências dedicadas exclusivamente a estes tópicos. Hoje em dia, os AGs têm se beneficiado muito da interdisciplinaridade. Cada vez mais cientistas buscam inspiração em outras áreas de pesquisa de forma a absorver suas ideias e fazer com que os AGs sejam mais eficientes e inteligentes na resolução de problemas (Linden, 2012).

3.3 FUNDAMENTOS DOS ALGORITMOS GENÉTICOS

Algoritmos evolucionários usam modelos computacionais dos processos naturais de evolução como uma ferramenta para resolver problemas. Apesar de haver uma grande variedade de modelos computacionais propostos, todos eles têm em comum o conceito de simulação da evolução das espécies por meio de seleção, recombinação e mutação. Os algoritmos evolucionários funcionam mantendo uma população de estruturas, denominadas indivíduos ou cromossomos, operando sobre estas estruturas de forma semelhante à evolução das espécies. Aos cromossomos são aplicados os chamados operadores genéticos, como recombinação e mutação. Cada indivíduo recebe uma avaliação que é uma quantificação da sua qualidade como solução do problema em questão. Com base nesta avaliação são aplicados os operadores genéticos de forma a simular a sobrevivência do indivíduo mais apto (Linden, 2012).

Os algoritmos genéticos são um ramo dos algoritmos evolucionários, e como tal podem ser definidos como uma técnica de busca (recebe um problema como entrada e retorna uma solução sob a forma de uma sequência de ações que ele recomenda que sejam tomadas para que se atinja o objetivo desejado (Russel, et al., 2004)) baseada em uma metáfora do processo biológico de evolução natural. Nos AGs, populações de indivíduos são criadas e submetidas aos operadores genéticos: seleção, recombinação (*crossover*) e mutação.

Assim como na natureza, a informação (que é um ponto crucial dentro do AG; representada pela função de avaliação do problema) deve ser codificada nos cromossomos e a reprodução se encarregará de fazer com que a população evolua. A mutação cria diversidade, mudando aleatoriamente genes dentro de indivíduos e, assim como na natureza, é aplicada de forma menos frequente que a recombinação, a qual é resultado da reprodução. A recombinação e a mutação são aplicadas em indivíduos selecionados dentro da população. A seleção deve ser feita de tal forma que os indivíduos mais aptos sejam selecionados mais frequentemente do que aqueles menos aptos, de forma que as boas características dos indivíduos mais aptos passem a predominar dentro da nova população de soluções. De forma alguma os indivíduos menos aptos têm que ser descartados da população reprodutora. Isso causaria uma rápida convergência genética de todas as soluções para um mesmo conjunto de características e evitaria uma busca mais ampla pelo espaço de soluções (Linden, 2012).

De modo geral, os algoritmos genéticos não são algoritmos de busca da solução ótima de um problema, mas sim uma heurística que encontra boas soluções a cada execução, mas não necessariamente a mesma, todas às vezes (Linden, 2012).

3.3.1 Principais características dos algoritmos genéticos

Pode-se dizer que os algoritmos genéticos são uma técnica de busca com as seguintes características positivas (Linden, 2012):

- Paralela: mantém uma população de soluções que são avaliadas simultaneamente.
- Global: AGs não usam apenas informação local, logo, não necessariamente ficam presos em máximos locais como alguns métodos de busca.
- Não totalmente aleatória: AGs têm componentes aleatórios, mas como utilizam a informação da população corrente para determinar o próximo estado de busca, não podem ser considerados totalmente aleatórios.
- Não afetado por descontinuidades na função ou em suas derivadas: os AGs não usam informações de derivadas na sua solução nem necessitam de informação sobre o seu entorno para poder efetuar sua busca.
- Capaz de lidar com funções discretas e contínuas: os AGs são capazes de lidar com funções reais, discretas, booleanas e até mesmo categóricas (não-numéricas), sendo possível inclusive misturar as representações sem prejuízo para a habilidade dos AGs de resolverem problemas.

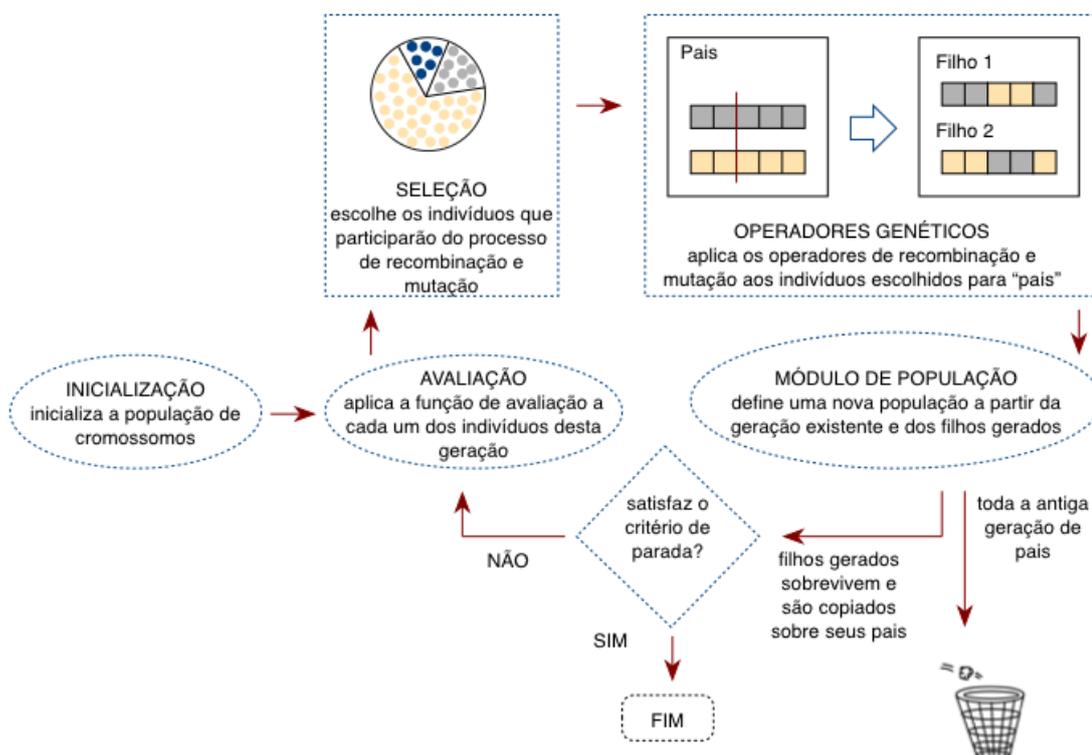
Além disto, tendo em vista que são buscas direcionadas e inteligentes, o AG é uma boa técnica para atacar problemas de busca com espaços de busca intratavelmente grandes, que não podem ser resolvidos por técnicas tradicionais (Linden, 2012).

3.3.2 Algoritmo de funcionamento de um AG

O algoritmo de funcionamento de um AG pode ser visto na Figura 3.1 e também resumidamente nos passos abaixo:

1. Inicializa-se a população de cromossomos;
2. Avalia-se cada cromossomo na população;
3. Seleciona-se os pais para gerar novos cromossomos;
4. Aplica-se os operadores de recombinação e mutação a estes pais de forma a gerar os indivíduos da nova geração;
5. Apaga-se os velhos membros da população;
6. Avalia-se todos os novos cromossomos e inseri-os na população;
7. Se o tempo acabou, ou o melhor cromossomo satisfaz os requisitos e desempenho, retorná-lo, caso contrário, voltar para o passo 3.

Figura 3.1 – Algoritmo de funcionamento de um AG (adaptada de Linden, 2012).



3.3.3 Representação cromossomial

A representação cromossomial é fundamental para um algoritmo genético. Essa representação consiste em uma maneira de traduzir a informação do problema em uma maneira viável de ser tratada pelo computador. Na representação cromossomial, cada pedaço indivisível desta representação é chamado de **gene**. A representação cromossomial é completamente arbitrária, sendo que a mais simples e mais utilizada pelos praticantes da área de algoritmos genéticos é a binária, ou seja, um cromossomo é uma sequência de *bits* e cada gene é somente um *bit* (Linden, 2012). Essa representação foi adotada inicialmente por Holland, e pelo fato de ser muito simples e também por questões históricas é amplamente adotada por pesquisadores da área de AG.

Para representar números reais como números binários é importante levar em consideração dois fatores (Linden, 2012):

- A faixa de operação de cada uma das variáveis; e
- A precisão desejada.

Estes dois parâmetros definem, em conjunto, quantos *bits* por variável deve-se usar. A precisão máxima de uma variável é dada pela equação 3.1 (Linden, 2012):

$$\text{Precisão máxima} = \frac{\text{sup}_i - \text{inf}_i}{2^k - 1} \quad 3.1$$

Sendo:

sup_i : o limite de operação superior;

inf_i : o limite de operação inferior; e

k : o número de *bits*.

Para converter o número binário dentro do cromossomo corrente para o número real correspondente dentro da faixa, o que se deve fazer é obter o número inteiro r_i correspondente ao número binário e depois utilizar a equação 3.2 (Linden, 2012).

$$Real = inf_i + \frac{sup_i - inf_i}{2^k - 1} \cdot r_i \quad 3.2$$

3.3.4 População

A inicialização da população, na maioria dos trabalhos desenvolvidos na área, é realizada da forma mais simples possível, fazendo uma escolha aleatória independente para cada indivíduo da população inicial.

O desempenho do algoritmo genético é extremamente sensível ao tamanho da população, logo este parâmetro deve ser definido com muito cuidado. Caso este número seja pequeno demais, não haverá espaço para uma variedade genética suficientemente grande dentro da população, o que fará com que o algoritmo seja incapaz de achar boas soluções e caso este número seja grande demais, o algoritmo demorará demais a achar boas soluções (Linden, 2012).

Existem algumas técnicas que permitem explorar melhor as qualidades da população da geração atual para a melhoria da população da próxima geração, entre elas destacam-se o **elitismo**, e o **steady state** (estado estacionário) (Linden, 2012).

- Elitismo

O elitismo é uma pequena modificação no módulo de população que quase não altera o tempo de processamento, mas que garante que o desempenho do AG sempre cresça com o decorrer das gerações. A ideia básica por trás do elitismo é: os k melhores indivíduos de cada geração não devem “morrer” junto com sua geração, mas sim passar para a próxima visando garantir que seus genomas sejam preservados.

A manutenção do melhor indivíduo da geração t na população da geração $t + 1$ garante que o melhor indivíduo da geração $t + 1$ seja, no mínimo, igual ao melhor indivíduo da geração t . Com isso, o gráfico da avaliação do melhor indivíduo como função do número de gerações decorridas é uma função monotonicamente crescente.

- Steady State

No “mundo real”, os indivíduos nascem aos poucos, os mais velhos morrem de forma lenta e há interação entre as gerações. Isto quer dizer que indivíduos da geração $t + 1$ podem reproduzir com indivíduos da geração t . A ideia por trás da técnica *steady state* é reproduzir este tipo de característica natural das populações biológicas. Em vez de se criar uma população completa de uma só vez, vai criando-se “filhos”, de acordo com a conveniência do operador de recombinação, e substituindo os piores “pais” por estes novos indivíduos (Linden, 2012).

Usando o *steady state*, o conceito de geração dentro do AG fica difuso, e pode haver cruzamento entre indivíduos recém-criados e outros da geração anterior. Isto permite uma maior dominação dos melhores esquemas, e normalmente faz com que a população convirja mais rapidamente, especialmente se forem eliminados sempre os piores elementos a cada operação realizada.

Segundo Linden (2012), um dos grandes problemas do *steady state* é que, de forma geral, há uma convergência muito rápida da população, com a conseqüente diminuição da variedade genética, que ocorre principalmente quando não é controlada a idade e a permanência dos indivíduos. O motivo principal é que os filhos tendem a ter avaliações razoavelmente semelhantes às dos seus pais, pois herdam os principais esquemas destes. Conseqüentemente, a população tende a se reunir em grupos de indivíduos, sendo que, os grupos com maiores avaliações médias tendem a predominar.

Para evitar que essa convergência seja rápida demais, pode-se usar a técnica de *steady state* sem duplicatas, que é caracterizada pelo fato de descartar o indivíduo gerado caso ele seja idêntico a algum já presente na população, ou seja, para cada operação de recombinação ou de mutação realizada, é necessário verificar se os filhos resultantes já estão presentes na população (Linden, 2012).

3.3.5 Principais métodos de seleção de pais

O método de seleção de pais deve simular o mecanismo de seleção natural que atua sobre as espécies biológicas, em que os pais mais capazes geram mais filhos, ao mesmo tempo em que permite que os pais menos aptos também possam gerar descendentes. O conceito fundamental é privilegiar os indivíduos com função de avaliação alta, sem desprezar completamente aqueles indivíduos com função de avaliação extremamente baixa, pois se apenas os melhores indivíduos se reproduzirem, a população tenderá a ser composta de indivíduos cada vez mais semelhantes, e dessa forma, faltará diversidade à população para que a evolução possa prosseguir de forma satisfatória. Este efeito denomina-se **convergência genética** (Blickle, et al., 1997).

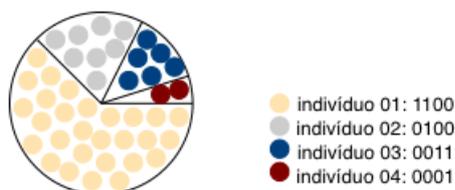
A **pressão seletiva**, ou intensidade de seleção, é a força que o método de seleção faz para impelir os esquemas contidos nas melhores soluções para a próxima geração, favorecendo o aspecto de aproveitamento do AG. É comum medir a intensidade da pressão seletiva por meio da melhoria obtida na avaliação média dos indivíduos da população atual, normalizada pelo desvio padrão. Ao final de uma execução do AG, este valor tende a diminuir bastante, devido à convergência genética (Linden, 2012).

Existem consolidados na literatura alguns métodos de seleção de pais, dentre os quais pode-se destacar a **roleta** e o **torneio**.

- Roleta

Neste método, cria-se uma roleta (virtual) na qual cada cromossomo recebe um pedaço proporcional à sua avaliação (número entre 0 e 100; ou entre 0 e 360, que representa uma posição do círculo), sendo que a soma dos pedaços não pode ser maior que 100%; depois roda-se a roleta (aleatoriamente) e o selecionado será o indivíduo sobre o qual ela parar. Quando se faz este sorteio um número suficiente de vezes, cada indivíduo é selecionado um número de vezes igual à sua fração na roleta. A Figura 3.2 mostra um exemplo de roleta, onde cada fração da roleta representa um número binário proporcional ao seu valor numérico (Linden, 2012).

Figura 3.2 – Exemplo de roleta.



- Torneio

O método do torneio consiste em selecionar uma série de indivíduos da população e fazer com que eles entrem em competição direta, usando como arma a sua avaliação. Neste método, existe um parâmetro denominado **tamanho do torneio** (k) que define quantos indivíduos são selecionados aleatoriamente dentro da população para competir. Uma vez definidos os competidores, aquele dentre eles que possui a melhor avaliação é selecionado para a aplicação do operador genético. O valor mínimo de k é igual a 2, pois do contrário não haverá competição, mas não há nenhum limite teórico para o valor máximo deste parâmetro. Se for escolhido o valor igual ao tamanho da população (n) o vencedor será sempre o melhor de todos os indivíduos, e se forem escolhidos valores próximos do tamanho da população, os $(n - k)$ indivíduos tenderão a predominar, uma vez que sempre um deles será o vencedor do torneio (Linden, 2012).

Os indivíduos são selecionados para participar do torneio de forma completamente aleatória. Não existe nenhum favorecimento para os melhores indivíduos. A única vantagem que os melhores indivíduos da população têm é que, se selecionados, eles vencerão o torneio (Linden, 2012). O método do torneio está resumido no exemplo da Figura 3.3, onde à esquerda tem-se a população com a avaliação de cada indivíduo; à direita, os elementos sorteados para cada torneio e o vencedor do mesmo (em destaque), selecionado para o operador a ser aplicado.

Figura 3.3 – Exemplo de aplicação do método do torneio (adaptada de Linden, 2012).

INDIVÍDUO	AVALIAÇÃO
X1	200
X2	100
X3	9500
X4	100
X5	100
X6	10000
X7	1
X8	40



TORNEIOS		
X1	X7	X8
X2	X3	X5
X6	X4	X4
X2	X7	X1
X5	X5	X5
X3	X4	X2
X4	X2	X6
X4	X6	X5

Não há nenhum impedimento para o fato de um elemento ser selecionado mais de uma vez dentro do mesmo torneio. Como o sorteio é completamente aleatório, isto pode acontecer, sendo apenas pouco provável.

3.3.6 Função de avaliação

A função de avaliação, também chamada de função de custo, é uma maneira utilizada pelo AG para determinar a qualidade de um indivíduo como solução do problema em questão, ou seja, calcula um valor numérico que reflete quão bem os parâmetros representados no cromossomo resolvem o problema. Dada a generalidade dos AGs, a função de avaliação, em muitos casos, é a única ligação verdadeira do programa com o problema real. Isto decorre do fato de que a função de avaliação só julga a qualidade da solução que está sendo apresentada por aquele indivíduo, sem armazenar qualquer tipo de informação sobre as técnicas de resolução do problema. Isto leva à conclusão de que o mesmo AG pode ser usado para descobrir o máximo de toda e qualquer função de n variáveis sem nenhuma alteração das estruturas de dados e procedimentos adotados, alterando-se, apenas, a função de avaliação (Linden, 2012).

De modo geral, a função de avaliação deve refletir os objetivos a serem alcançados na resolução de um problema, de forma numérica, e é derivada diretamente das condições impostas pelo problema em questão (Koza, et al., 2003).

A função de avaliação costuma ser usada como medida de qualidade do indivíduo, mas em alguns casos, como a questão do superindivíduo e da pequena diferença entre as avaliações, esse procedimento pode fazer com que o desempenho do AG degenere. A questão do superindivíduo ocorre quando há um ou mais indivíduos cuja avaliação é muito superior à dos outros membros da população. Neste caso, este indivíduo ou este grupo será quase sempre escolhido pelo módulo de seleção, causando uma perda imediata da diversidade genética nas gerações imediatamente subsequentes. No caso em que há uma pequena diferença entre as avaliações, todos os indivíduos têm funções de avaliação que diferem muito pouco percentualmente. Na maioria dos casos em que isso ocorre, uma pequena diferença entre as funções de avaliação significa uma grande diferença na qualidade da solução (Linden, 2012).

Para resolver estes problemas são utilizadas técnicas como a **normalização linear** e o **janelamento**.

- Normalização linear

Na técnica de normalização linear, os cromossomos são ordenados em ordem decrescente de valor, criando novas funções de avaliação para cada um dos indivíduos, de forma que o melhor de todos receba um valor fixo (k) e os outros recebam valores iguais ao do indivíduo imediatamente anterior na lista ordenada menos um valor de decremento constante (t). A técnica pode ser resumida de acordo com as equações 3.3 e 3.4 (Linden, 2012):

$$aval_0 = k \quad 3.3$$

$$aval_i = aval_{i-1} - t, \quad \forall \quad i = 1, 2, \dots, n - 1 \quad 3.4$$

Sendo:

$aval_i$: a função de avaliação para o indivíduo i ;

k : o valor fixo do indivíduo com melhor função de avaliação, geralmente na literatura, é escolhido o valor 1;

i : a posição do indivíduo na população; e

t : o valor de decremento constante.

- Janelamento

Existem situações em que as diferenças entre os valores absolutos das avaliações dos indivíduos são muito pequenas. Isto pode ocorrer por dois motivos: convergência genética ou característica inerente da função de avaliação utilizada.

1. A convergência genética é o fenômeno que ocorre conforme as gerações vão passando, quando os indivíduos vão se tornando cada vez mais parecidos, o que faz com que, no caso do uso do método de seleção baseada em roleta, a pressão seletiva seja diminuída (Linden, 2012).
2. Para evitar problemas relacionados as características inerentes da função de avaliação, pode-se usar a técnica denominada janelamento, que pode ser resumida da seguinte maneira (Linden, 2012):
 - ✓ Achar o valor mínimo dentre as funções de avaliação da população e diminuir esse valor mínimo de um valor pequeno arbitrário.
 - ✓ Designar para cada um dos cromossomos uma avaliação que seja igual à quantidade que excede este valor mínimo.

A subtração de um pequeno valor é feita de forma que o indivíduo de menor avaliação não passe a ter um *fitness* igual a zero, o que faria com que ele nunca fosse selecionado.

3.3.7 Operador de *crossover*

O operador de *crossover* é aplicado após o processo de seleção, sendo comumente executado por técnicas como o ***crossover* de um ponto**, de **dois pontos** ou **uniforme** (Davis, 1991). O *crossover* consiste na formação de um novo indivíduo por meio da junção de parte dos genes dos indivíduos resultantes do processo de seleção.

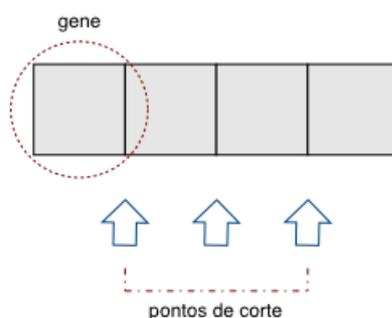
Portanto, um algoritmo genético usa a seleção para determinar as áreas mais promissoras de pesquisa e o *crossover* para combiná-las de forma a tentar gerar soluções de maior qualidade para o problema em questão (Linden, 2012).

Historicamente, a taxa de *crossover* tem recebido um valor percentual muito alto, variando de 60% a 95%. Isto tem sido associado a taxas de mutação bastante baixas, reforçando a importância do operador de *crossover* na geração de soluções (Linden, 2012).

- Operador de *crossover* de um ponto

O operador de *crossover* de um ponto é o mais simples de todos os operadores. Depois de selecionados dois pais, pelo módulo de seleção, um ponto de corte (posição entre dois genes de um cromossomo) é selecionado (Figura 3.4) (Linden, 2012).

Figura 3.4 – Exemplos de ponto de corte.



Depois de sorteado o ponto de corte, os pais são separados em duas partes (uma à esquerda do ponto de corte e outra à direita), sendo que o tamanho das partes não necessariamente é igual (Linden, 2012).

O primeiro filho é composto por meio da concatenação da parte do primeiro pai à esquerda do ponto de corte com a parte do segundo pai à direita do ponto de corte. O segundo filho é composto por meio da concatenação das partes restantes (a metade do segundo pai à esquerda do ponto de corte com a metade do primeiro pai à direita do ponto de corte) (Figura 3.5) (Linden, 2012).

Figura 3.5 – Resumo ilustrativo da operação de *crossover* de um ponto.



É comum associar uma probabilidade (P_C) maior ou igual a 80% para o uso do operador de *crossover*.

- Operador de *crossover* de dois pontos

O funcionamento do *crossover* de dois pontos é similar ao funcionamento do *crossover* de um ponto, com a diferença no ponto de corte, que passa a ser dois em vez de apenas um. E assim, como no *crossover* de um ponto, no *crossover* de dois pontos os pontos de corte também são sorteados. O primeiro filho será então formado pela parte do primeiro pai fora dos pontos de corte e pela parte do segundo pai entre os pontos de corte e o segundo filho será formado pelas partes restantes (Figura 3.6) (Linden, 2012).

Figura 3.6 – Resumo ilustrativo da operação de *crossover* de dois pontos.



- *Crossover* uniforme

O *crossover* uniforme é capaz de combinar todo e qualquer esquema existente. Para cada gene é sorteado um número 0 ou 1. Se o valor sorteado for igual a 1, o filho número 1 recebe o gene da posição corrente do primeiro pai e o segundo filho o gene da posição corrente do segundo pai. Por outro lado, se o valor sorteado for 0, as atribuições são invertidas (Linden, 2012) como mostra a Figura 3.7.

Figura 3.7 – Funcionamento do *crossover* uniforme.



O *crossover* uniforme tende a obter resultados superiores se comparado com o *crossover* de um ponto ou de dois pontos, isso graças a sua capacidade de combinar esquemas.

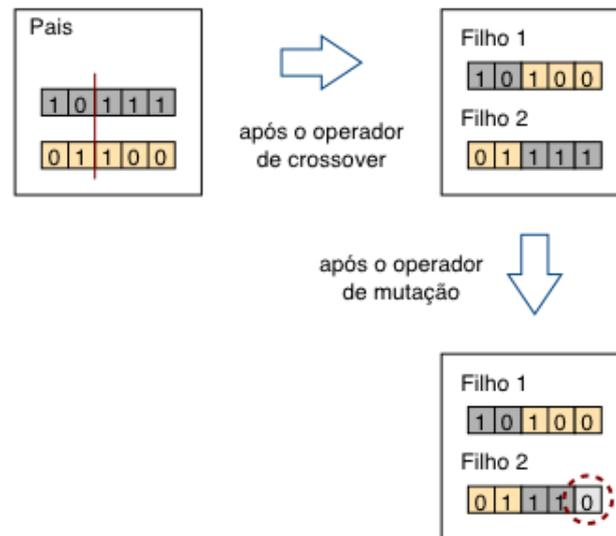
3.3.8 Operador de mutação

Depois de compostos os filhos, entra em ação o operador de mutação, que tem como função preservar a diversidade genética da população de soluções (Linden, 2012).

O operador de mutação funciona da seguinte forma: ele tem associada uma probabilidade extremamente baixa (da ordem de 0,5%); e é sorteado um número entre 0 e 1.

Se o número sorteado for menor que a probabilidade predeterminada, então o operador atua sobre o gene em questão, alterando-lhe o valor aleatoriamente. Repete-se então o processo para todos os genes componentes dos dois filhos (Linden, 2012). A Figura 3.8 apresenta um resumo da operação de *crossover* de um ponto e mutação.

Figura 3.8 – Resumo da operação do operador de *crossover* de um ponto e mutação (adaptada de Linden, 2012).



O conceito fundamental quanto ao valor da probabilidade é que esse deve ser baixo, geralmente associado ao operador de *crossover* ($P_m = 100\% - P_c$). O algoritmo genético se comportará de forma estranha se a taxa de mutação for igual a 100%, pois os *bits* do cromossomo serão invertidos e a qualidade da população degenerará rapidamente e dificilmente o AG convergirá para uma solução (Linden, 2012).

Pode-se dizer que o operador de mutação é uma heurística exploratória, injetando novos cromossomos na população e permitindo que o AG busque soluções fora dos limites definidos pela população inicial (Vose, 2004). Logo, se o valor da probabilidade atribuída ao operador de mutação for baixo demais, a população não terá diversidade depois de um certo número de gerações, estagnando bem rápido devido à convergência genética. Por outro lado, se o operador de mutação receber uma probabilidade alta demais, então o AG passará a ter um comportamento mais parecido com um algoritmo aleatório e perderá suas características interessantes (Lira, 2004).

3.4 CONSIDERAÇÕES FINAIS

De modo geral, as técnicas de inteligência computacional têm como objetivo a busca de soluções, que podem ser soluções numéricas, soluções do significado de uma expressão linguística ou de qualquer outro tipo que tenha significado em um determinado problema. Os algoritmos genéticos são uma técnica de busca extremamente eficiente no seu objetivo de varrer o espaço de soluções e encontrar as soluções próximas da solução ótima, quase sem necessitar de interferência humana. Neste sentido, este capítulo apresentou um breve histórico sobre os algoritmos genéticos, bem como seus fundamentos e as suas principais características.

CAPÍTULO 4 – REDES NEURAI MULTICAMADAS

4.1 INTRODUÇÃO

As redes neurais artificiais representam uma tecnologia que tem raízes em muitas disciplinas como: neurociência, matemática, estatística, física, ciência da computação e engenharia. As redes neurais encontram aplicações em campos tão diversos em virtude de uma importante propriedade que é a habilidade de aprender a partir de dados de entrada de forma supervisionada e não supervisionada.

O presente capítulo aborda os princípios, definições e principais características das redes neurais multicamadas (MLP). Os fundamentos das redes neurais são um tópico importante, pois a metodologia utiliza as redes neurais MLP para compressão e descompressão dos sinais de energia elétrica.

4.2 BREVE HISTÓRICO

O trabalho pioneiro utilizando redes neurais artificiais foi desenvolvido por McCulloch e Pitts (McCulloch, et al., 1943) dentro de uma comunidade de modelagem neural que tinha estado em atividade, na *University of Chicago*, por pelo menos cinco anos antes da publicação de 1943 intitulada “*A logical calculus of the ideas immanent in nervous activity*”, sob a liderança de *Rashevsky*. No seu clássico artigo, McCulloch e Pitts descrevem um cálculo lógico das redes neurais que unificava os estudos de neurofisiologia e da lógica matemática (Ferreira, 2002).

O próximo desenvolvimento significativo das redes neurais veio em 1949, com a publicação do livro de Hebb (Hebb, 2002), “*The Organization of Behavior*”, no qual foi apresentada pela primeira vez uma formulação explícita de uma regra de aprendizagem fisiológica para a modificação sináptica. O livro de Hebb foi imensamente influente entre os psicólogos, mas lamentavelmente teve pouco ou nenhum impacto sobre a comunidade de engenharia.

Em 1952 foi publicado o livro de Ashby (Ashby, 1960), “*Design for a brain: the origin of adaptive behavior*”, que trata da noção básica de que o comportamento adaptativo não é inato, mas sim aprendido, e que por meio da aprendizagem o comportamento de um sistema normalmente muda para melhor. Em 1954, Minsky (Minsky, 1954) apresentou sua tese de doutorado em redes neurais, na *University of Princeton*, intitulada “*Theory of neural-analog reinforcement systems and its application to the brain-model problem*”.

Em 1956, Rochester (Rochester, et al., 1956) e um grupo de pesquisadores tentaram, talvez pela primeira vez, usar a simulação computacional para testar uma teoria neural bem-formulada com base no postulado de aprendizagem de Hebb; os resultados de simulação

relatados naquele artigo mostraram claramente que devia-se adicionar inibição para que a teoria realmente funcionasse. Neste mesmo ano, Uttley (Uttley, 1959) demonstrou que uma rede neural com sinapses modificáveis poderia aprender a classificar conjuntos simples de padrões binários em classes correspondentes (Ferreira, 2002).

Cerca de quinze anos após a publicação do clássico artigo de McCulloch e Pitts, uma nova abordagem para o problema de reconhecimento de padrões foi introduzida por Rosenblatt (Rosenblatt, 1958), em seu trabalho sobre o *perceptron*, um método inovador de aprendizagem supervisionada. Durante o período clássico dos *perceptrons* nos anos 1960, parecia que as redes neurais poderiam realizar qualquer coisa. Mas, então, em 1969, foi publicado o livro de Minsky (Minsky, et al., 1969), que utilizava a matemática para demonstrar que existiam limites fundamentais para aquilo que os *perceptrons* de camada única poderiam calcular. Um problema importante encontrado no projeto de um *perceptron* multicamadas, relacionados aos neurônios intermediários da rede, contribuiu de um modo ou de outro para o esmorecimento do interesse continuado em redes neurais nos anos de 1970.

Uma atividade importante que emergiu nos anos de 1970 foram os mapas auto-organizáveis utilizando aprendizagem competitiva.

Nos anos de 1980, foram desenvolvidas importantes contribuições em várias frentes para teoria e o projeto de redes neurais. Em 1982 foi publicado o artigo de Kohonen (Kohonen, 1982) referentes aos mapas auto-organizáveis utilizando uma estrutura de rede unidimensional ou bidimensional. Em 1983 Kirkpatrick (Kirkpatrick, et al., 1983) descreveram um novo procedimento denominado “recozimento simulado” para resolver problemas de otimização combinatória. Em 1985, a ideia do recozimento simulado foi utilizada por Ackley (Ackley, et al., 1985) no desenvolvimento de uma máquina estocástica conhecida como “a máquina de Boltzmann”, que foi a primeira realização bem-sucedida de uma rede neural de múltiplas camadas (Ferreira, 2002).

Em 1986 foi relatado por Rumelhart (Rumelhart, et al., 1986) o desenvolvimento do algoritmo de retropropagação. A aprendizagem por retropropagação emergiu como o algoritmo de aprendizagem mais popular para o treinamento de *perceptron* multicamadas.

Em 1988, Linsker (Linsker, 1988) descreveu um novo princípio para a auto-organização em uma rede perceptiva. Neste mesmo ano, Broomhead (Broomhead, et al., 1988) descreveram um procedimento para o projeto de redes alimentadas adiante, em camadas utilizando funções de base radial, as quais fornecem uma alternativa aos *perceptron* multicamadas (Ferreira, 2002).

No início dos anos de 1990, Vapnik (Vapnik, et al., 1995) desenvolveram uma classe de redes de aprendizagem supervisionada poderosa do ponto de vista computacional, chamada de máquinas de vetor de suporte (Ferreira, 2002).

As redes neurais certamente trilharam um longo caminho desde os dias iniciais de McCulloch e Pitts. De fato, elas se estabeleceram como um tema interdisciplinar com raízes profundas em neurociências, psicologia, matemática, ciências físicas e engenharia.

4.3 FUNDAMENTOS DAS REDES NEURAIS ARTIFICIAIS

O trabalho em redes neurais artificiais, usualmente denominadas “redes neurais”, tem sido motivado desde o seu início pelo reconhecimento de que o cérebro humano processa informações de uma forma inteiramente diferente do computador digital convencional. O cérebro é um “computador” (sistema de processamento de informação) altamente complexo, não-linear e paralelo. Ele tem a capacidade de organizar seus constituintes estruturais, conhecidos por neurônios, de forma a realizar certos processamentos, como o reconhecimento de padrões, por exemplo, muito mais rapidamente que o mais rápido computador digital hoje existente (Haykin, 2001).

No momento do nascimento, o cérebro tem uma grande estrutura e a habilidade de desenvolver suas próprias regras por meio do que usualmente denomina-se “experiência”. Essa experiência vai sendo acumulada com o tempo, sendo que o mais dramático desenvolvimento, por ligações físicas, do cérebro humano, acontece durante os dois primeiros anos de vida; mas o desenvolvimento continua para muito além desse estágio (Haykin, 2001).

Um neurônio em “desenvolvimento” é sinônimo de um cérebro plástico. A plasticidade é a habilidade que o sistema nervoso em desenvolvimento tem de se adaptar ao seu meio ambiente (Haykin, 2001). Assim como a plasticidade parece ser essencial para o funcionamento dos neurônios como unidades de processamento de informação do cérebro humano, ela também é essencial no que se refere às redes neurais construídas com neurônios artificiais.

Na sua forma mais geral, uma rede neural é uma máquina projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse; a rede é normalmente implementada utilizando-se componentes eletrônicos ou é simulada por programação em um computador (Haykin, 2001).

Para alcançar um bom desempenho, as redes neurais empregam uma interligação maciça de células computacionais simples denominadas neurônios ou unidades de processamento. Sendo assim, pode-se definir uma rede neural, vista como uma máquina adaptativa, como um processador maciçamente e paralelamente distribuído constituído de unidades de processamento simples, que tem a propensão natural de armazenar conhecimento experimental e torná-lo disponível para uso (Haykin, 2001).

Uma rede neural artificial se assemelha ao cérebro humano em dois aspectos (Aleksander, et al., 1990): o conhecimento é adquirido pela rede a partir de seu ambiente por meio de um processador de aprendizagem; e forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

O procedimento utilizado para realizar o processo de aprendizagem é chamado de algoritmo de aprendizagem, cuja função é modificar os pesos sinápticos da rede neural de uma forma ordenada para alcançar um objetivo de projeto desejado (Haykin, 2001). A modificação dos pesos sinápticos é o método tradicional para o projeto de redes neurais artificiais.

4.3.1 Benefícios das redes neurais artificiais

Uma rede neural extrai seu poder computacional por meio, primeiramente, de sua estrutura maciçamente paralelamente distribuída e segundo, de sua habilidade de aprender e, portanto, de generalizar. A generalização se refere ao fato da rede neural produzir saídas adequadas para entradas que não estavam presentes durante o treinamento, ou seja, durante a aprendizagem (Haykin, 2001). Estas duas capacidades de processamento de informação tornam possível para as redes neurais resolverem problemas complexos, de grande escala. Na prática, as redes neurais não podem fornecer uma solução trabalhando individualmente. Elas precisam ser interligadas em uma abordagem consistente de engenharia. Então, um problema complexo de interesse é decomposto em um número de tarefas relativamente simples, e atribui-se às redes neurais um subconjunto de tarefas que “coincidem” com suas capacidades inerentes.

O uso das redes neurais oferece as seguintes propriedades úteis e capacidades (Haykin, 2001):

- Não-linearidade: um neurônio artificial pode ser linear ou não-linear. Uma rede neural, constituída por conexões de neurônios não-lineares é não-linear, sendo que sua não-linearidade é distribuída por toda a rede.
- Mapeamento de entrada-saída: um paradigma popular de aprendizagem chamado aprendizagem supervisionada envolve a modificação dos pesos sinápticos de uma rede neural pela aplicação de um conjunto de amostras de treinamento rotuladas. Cada exemplo consiste de um sinal de entrada único e de uma resposta desejada correspondente. Apresenta-se para a rede um exemplo do conjunto escolhido ao acaso, e os pesos sinápticos da rede são modificados para minimizar a diferença entre a resposta desejada e a resposta real da rede, produzida pelo sinal de entrada, de acordo com um critério estatístico apropriado. O treinamento da rede é repetido

para muitos exemplos do conjunto até que a rede alcance um estado estável onde não hajam mais modificações significativas nos pesos sinápticos.

- Adaptabilidade: as redes neurais têm a capacidade de adaptar seus pesos sinápticos às modificações do meio ambiente. Em particular, uma rede neural treinada para operar em um ambiente específico pode ser facilmente re-treinada para lidar com pequenas modificações nas condições operativas do ambiente.
- Resposta a evidências: no contexto de classificação de padrões, uma rede neural pode ser projetada para fornecer informação não somente sobre qual padrão particular selecionar, mas também sobre a confiança na decisão tomada.
- Informação contextual: o conhecimento é representado pela própria estrutura e estado de ativação de uma rede neural. Cada neurônio da rede é potencialmente afetado pela atividade de todos os outros neurônios da rede. Conseqüentemente, a informação contextual é tratada naturalmente pela rede neural.
- Tolerância a falhas: se um neurônio e suas conexões são danificados, a recuperação de um padrão armazenado é prejudicada em qualidade. Contudo, devido à natureza distribuída da informação armazenada na rede, o dano tem que ser extenso para que a resposta global da rede seja degradada seriamente. Uma rede danificada pode ainda ser capaz de funcionar satisfatoriamente, e a aprendizagem não precisa ser perfeita (Kerlirzin, et al., 1993).
- Uniformidade de análise e projeto: as redes neurais desfrutam de universalidade como processadores de informação. Isso pode ser dito no sentido de que a mesma notação é utilizada em todos os domínios envolvendo a aplicação de redes neurais. Esta característica se manifesta de diferentes modos: os neurônios, de uma forma ou de outra, representam um ingrediente comum a todas as redes neurais.
- Analogia neurobiológica: o projeto de uma rede neural é motivado pela analogia com o cérebro, que é uma prova viva de que o processamento paralelo tolerante a falhas é não somente possível fisicamente, mas também rápido e poderoso. O paralelismo não é somente um conceito essencial ao processamento de informação em redes neurais, mas é também a fonte de sua flexibilidade. Além disso, o paralelismo pode ser maciço (centenas de milhares de neurônios), o que dá as redes neurais uma forma notável de robustez.

4.3.2 Inteligência artificial e redes neurais

O objetivo da inteligência artificial (IA) é o desenvolvimento de paradigmas ou algoritmos que requeiram máquinas para realizar tarefas cognitivas, para as quais os humanos são atualmente melhores. Um sistema de IA deve ser capaz de realizar três

tarefas: armazenar o conhecimento, aplicar o conhecimento armazenado para resolver problemas, e adquirir novo conhecimento por meio da experiência (Haykin, 2001).

Um sistema de inteligência artificial tem três componentes fundamentais: **representação, raciocínio e aprendizagem** (Sage, 1990).

- Representação

Provavelmente, a característica mais distintiva da IA seja o uso difundido de uma linguagem de estruturas simbólicas para representar tanto o conhecimento genérico sobre um domínio do problema de interesse como o conhecimento específico sobre a solução do problema.

- Raciocínio

Na sua forma mais básica, raciocínio é a habilidade de resolver problemas. Para um sistema ser qualificado como um sistema de raciocínio, ele deve satisfazer as seguintes condições (Fischler, et al., 1987):

- ✓ O sistema deve ser capaz de expressar e resolver uma vasta gama de problemas e tipos de problemas;
- ✓ O sistema deve ser capaz de tornar conhecidas para ele tanto a informação explícita como a informação implícita; e
- ✓ O sistema deve ter um mecanismo de controle que determine quais operações devem ser aplicadas para um problema particular, quando uma solução para este problema foi obtida, ou quando deve ser encerrado o tratamento deste problema.

- Aprendizagem

O elemento de aprendizagem utiliza a informação fornecida pelo ambiente para aperfeiçoar a base de conhecimento. O elemento de desempenho utiliza a base de conhecimento para executar a sua tarefa.

4.3.3 Processo de aprendizagem

A propriedade de importância primordial para uma rede neural é a sua habilidade de aprender a partir de seu ambiente e de melhorar o seu desempenho por meio da aprendizagem. A melhoria do desempenho ocorre com o tempo de acordo com alguma medida pré-estabelecida. Uma rede neural aprende sobre o seu ambiente por meio de um processo iterativo de ajustes, aplicados a seus pesos sinápticos e níveis de *bias*. Idealmente, a rede se torna mais instruída sobre seu ambiente após cada iteração do processo de aprendizagem (Haykin, 2001).

Há diversas atividades associadas à noção de “aprendizagem” para justificar a sua definição de forma precisa. Além disso, o processo de aprendizagem depende do ponto de vista, o que causa dificuldades em se obter um consenso sobre uma definição precisa do termo. Contudo, pode-se definir aprendizagem, no contexto de redes neurais artificiais, como um processo pelo qual os parâmetros livres de uma rede neural são adaptados por meio de um processo de estimulação pelo ambiente no qual a rede está inserida (Mendel, et al., 1970). O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre.

Esta definição do processo de aprendizagem implica a seguinte sequência de eventos:

1. A rede neural é estimulada por um ambiente;
2. A rede neural sofre modificações nos seus parâmetros livres como resultado desta estimulação; e
3. A rede neural responde de uma maneira nova ao ambiente, devido às modificações ocorridas na sua estrutura interna.

Um conjunto pré-estabelecido de regras bem definidas para a solução de um problema de aprendizagem é denominado de algoritmo de aprendizagem. Os algoritmos de aprendizagem podem ser: **aprendizagem por correção de erro**, **aprendizagem baseada em memória**, **aprendizagem Hebbiana**, **aprendizagem competitiva** e **aprendizagem de Boltzmann**. Como se pode esperar, não há um algoritmo de aprendizagem único para o projeto de redes neurais e sim, um “conjunto de ferramentas” representado por uma variedade de algoritmos de aprendizagem, cada qual oferecendo vantagens específicas (Haykin, 2001).

Basicamente, os algoritmos de aprendizagem diferem entre si pela forma como é formulado o ajuste de um peso sináptico de um neurônio. Outro fator a ser considerado é a maneira pela qual uma rede neural, constituída de um conjunto de neurônios interligados, se relaciona com o seu ambiente. Este último caso refere-se ao paradigma de aprendizagem de um modelo do ambiente no qual a rede neural opera, podendo ser supervisionado (com um professor) e não-supervisionado (sem um professor) (Haykin, 2001).

4.4 PERCEPTRON MULTICAMADAS - MLP

As redes de múltiplas camadas alimentadas adiante são uma importante classe de redes neurais. Tipicamente, a rede consiste de um conjunto de unidades sensoriais (nós de fonte) que constituem a camada de entrada, uma ou mais camadas intermediárias de nós computacionais e uma camada de saída. O sinal de entrada se propaga para frente através

da rede, camada por camada. Estas redes neurais são normalmente chamadas de *perceptrons* multicamadas (MLP, *multilayer perceptron*) (Haykin, 2001).

Os *perceptrons* multicamadas têm sido aplicados com sucesso para resolver diversos problemas difíceis, por meio do seu treinamento de forma supervisionada com um algoritmo muito popular conhecido como algoritmo de retropropagação de erro (*error back-propagation*). Este algoritmo é baseado na regra de aprendizagem por correção de erro (Haykin, 2001).

Basicamente, a aprendizagem por retropropagação de erro consiste de dois passos através das diferentes camadas da rede: um passo para frente, a propagação, e um passo para trás, a retropropagação. No passo para frente, um padrão de atividade (vetor de entrada) é aplicado aos nós sensoriais da rede e seu efeito se propaga através da rede, camada por camada. Finalmente, um conjunto de saídas é produzido como a resposta real da rede. Durante o passo de propagação, os pesos sinápticos da rede são todos fixos (Haykin, 2001).

Durante o passo para trás, por outro lado, os pesos sinápticos são todos ajustados de acordo com uma regra de correção de erro. Especificamente, a resposta real da rede é subtraída de uma resposta desejada (alvo) para produzir um sinal de erro. Este sinal de erro é então propagado para trás através da rede, contra a direção das conexões sinápticas. Os pesos sinápticos são ajustados para fazer com que a resposta real da rede se mova para mais perto da resposta desejada.

O algoritmo de retropropagação de erro é também referido na literatura como algoritmo de retropropagação (*back-propagation*). O processo de aprendizagem realizado com o algoritmo é chamado de aprendizagem por retropropagação.

Um *perceptron* multicamadas tem três características distintas (Haykin, 2001):

1. O modelo de cada neurônio da rede inclui uma função de ativação não-linear. O ponto importante a se enfatizar aqui é que a não-linearidade é suave, isto é, diferenciável em qualquer ponto.
2. A rede contém uma ou mais camadas de neurônios intermediários, que não são parte da entrada ou da saída da rede. Estes neurônios intermediários capacitam a rede a aprender tarefas complexas extraindo progressivamente as características mais significativas dos padrões (vetores) de entrada.
3. A rede exibe um alto grau de conectividade, determinado pelas sinapses da rede. Uma modificação na conectividade da rede requer uma mudança na população das conexões sinápticas ou de seus pesos.

É por meio da combinação destas características, juntamente com a habilidade de aprender a partir da experiência por meio de treinamento, que o *perceptron* multicamadas deriva seu poder computacional (Haykin, 2001).

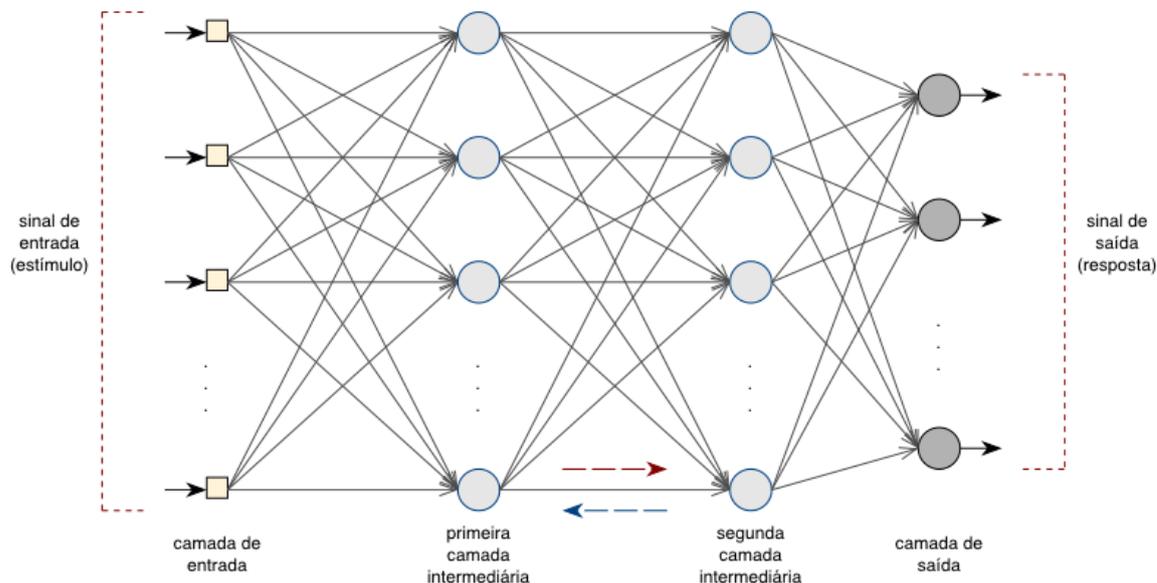
O desenvolvimento do algoritmo de retropropagação representa um marco no estudo das redes neurais artificiais, pois fornece um método computacional eficiente para treinamento de *perceptron* multicamadas.

4.4.1 Considerações preliminares

Para ajudar na descrição do *perceptron* multicamadas na sua forma geral, a Figura 4.1 mostra o grafo arquitetural de uma rede *perceptron* multicamadas com duas camadas intermediárias e uma camada de saída.

A rede neural utilizada neste exemplo é totalmente conectada. Isto significa que um neurônio em qualquer camada da rede está conectado a todos os nós/neurônios da camada anterior. O fluxo de sinal através da rede se propaga para frente, da esquerda para direita e de camada em camada.

Figura 4.1 – Grafo arquitetural de um *perceptron* multicamadas com duas camadas intermediárias (adaptada de Haykin, 2001).



Ainda na Figura 4.1 pode-se identificar dois tipos de sinais: os sinais funcionais (representados pela seta vermelha), que são sinais de entrada, ou seja, estímulos que incidem no terminal de entrada da rede e propagam-se para frente (neurônio por neurônio) através da rede e emergem no terminal de saída da rede como um sinal de saída; e os sinais de erro (representados pela seta azul), que se originam em um neurônio de saída da rede e se propagam para trás (camada por camada) através da rede.

Os neurônios de saída constituem a camada de saída da rede e os neurônios restantes constituem as camadas intermediárias da rede. Assim, as unidades intermediárias não são parte da saída ou da entrada da rede. A primeira camada intermediária é alimentada pela camada de entrada, constituída de unidades sensoriais (nós de fonte); as saídas resultantes da primeira camada intermediária são por sua vez aplicadas à próxima camada intermediária; e assim por diante para o resto da rede.

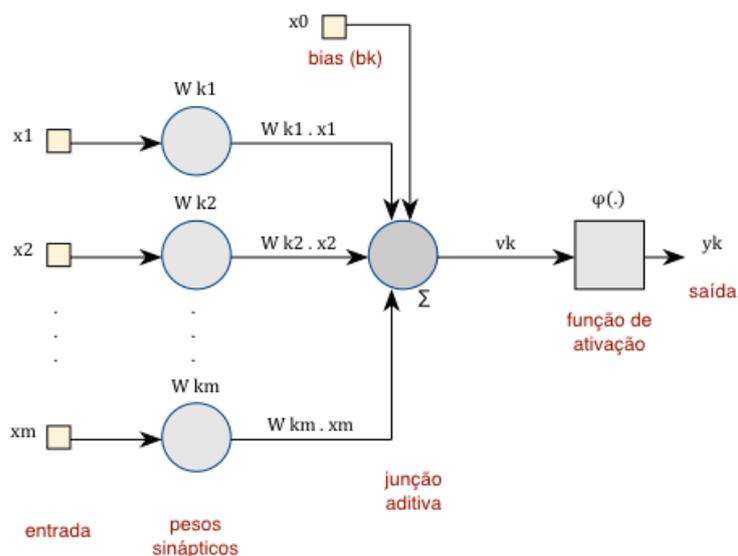
Cada neurônio intermediário ou de saída de uma rede *perceptron* multicamadas é projetado para realizar dois cálculos (Haykin, 2001):

1. O cálculo do sinal funcional que aparece na saída de um neurônio, que é expresso como uma função não-linear do sinal de entrada e dos pesos sinápticos associados com aquele neurônio; e
2. O cálculo de uma estimativa do vetor gradiente, isto é, os gradientes da superfície de erro em relação aos pesos conectados às entradas de um neurônio, que é necessário para a retropropagação através da rede.

4.4.2 Modelo de um neurônio da rede neural *perceptron* multicamadas

Um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural. A Figura 4.2 apresenta o modelo de um neurônio artificial, que forma a base para o projeto de redes neurais artificiais. Nesta figura, podem ser identificados três elementos básicos do modelo neuronal: os **pesos sinápticos**, a **junção aditiva**, e a **função de ativação**.

Figura 4.2 – Modelo de um neurônio artificial (adaptada de Haykin, 2011).



Sendo:

- x_m : o sinal de entrada (estímulo) da sinapse m ;
- w_{km} : os pesos sinápticos do neurônio k ;
- b_k : o bias;
- v_k : a saída do combinador linear;
- $\varphi(\cdot)$: o campo local induzido ou função de ativação; e
- y_k : a saída do neurônio.

- Pesos sinápticos

As sinapses são unidades estruturais e funcionais elementares que medeiam as interações entre os neurônios (Shepherd, et al., 1990).

Cada sinapse, em um conjunto de sinapses ou elos de conexão, é caracterizada por um peso ou força própria. De acordo com a Figura 4.2, um sinal x_m na saída da sinapse m conectada ao neurônio k é multiplicado pelo peso sináptico w_{km} . O primeiro índice do peso sináptico w_{km} se refere ao neurônio em questão (neurônio k) e o segundo se refere ao terminal de entrada da sinapse à qual o peso se refere. O peso sináptico de um neurônio artificial pode estar em um intervalo que inclui valores negativos bem como positivos (Shepherd, et al., 1990).

- Junção aditiva

A junção aditiva é um somador para os sinais de entrada, ponderados pelas respectivas sinapses do neurônio. Essa operação de somar os sinais de entrada x_m que foram previamente multiplicados pelos respectivos pesos sinápticos w_{km} constitui um combinador linear (Haykin, 2001).

- Função de ativação

A função de ativação restringe a amplitude da saída de um neurônio. É também conhecida como função restritiva já que restringe (limita) o intervalo permissível de amplitude do sinal de saída a um valor finito (Haykin, 2001).

Normalmente, o intervalo normalizado da amplitude da saída de um neurônio é escrito como um intervalo unitário fechado $[0, 1]$ ou alternativamente $[-1, 1]$.

O modelo neuronal mostrado na Figura 4.2 inclui também um *bias* aplicado externamente, representado por b_k . O *bias* é um parâmetro externo do neurônio artificial k e tem o efeito de aumentar ou diminuir a entrada da função de ativação (Haykin, 2001).

Em termos matemáticos, pode-se descrever um neurônio k de acordo com as equações 4.1 e 4.2:

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j \quad 4.1$$

e

$$y_k = \varphi(u_k + b_k) \quad 4.2$$

Sendo:

x_1, x_2, \dots, x_m : os sinais de entrada;

$w_{k1}, w_{k2}, \dots, w_{km}$: os pesos sinápticos do neurônio k ;

u_k : a saída do combinador linear devido aos sinais de entrada;

b_k : o *bias*;

$\varphi(\cdot)$: a função de ativação; e

y_k : o sinal de saída do neurônio.

O uso do *bias* b_k tem o efeito de aplicar uma transformação afim à saída u_k do combinador linear no modelo da Figura 4.2, como mostrado pela equação 4.3 a seguir:

$$v_k = u_k + b_k \quad 4.3$$

Sendo:

v_k : o campo local induzido ou potencial de ativação.

4.4.3 Tipos de função de ativação para *perceptron* multicamadas

A função de ativação, representada por $\varphi(v)$, define a saída de um neurônio em termos do campo local induzido v . Pode-se identificar três tipos básicos de funções de ativação (Haykin, 2001): **função de limiar**, **função linear por partes** e **função sigmóide**.

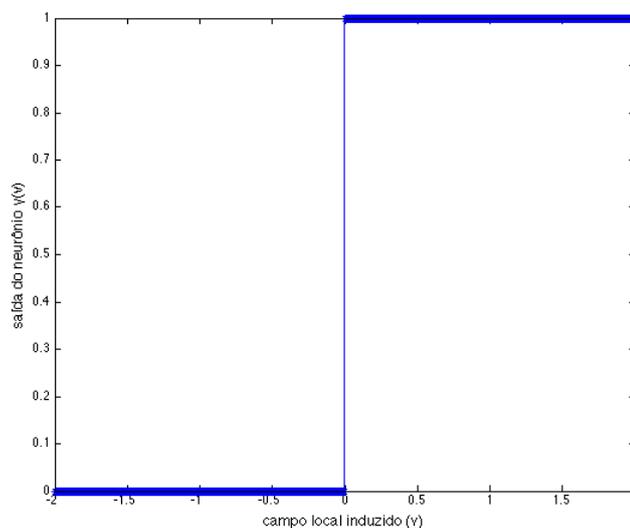
- Função de limiar

Para este tipo de função de ativação tem-se a descrição matemática vista na equação 4.4:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases} \quad 4.4$$

A função de ativação de limiar é ilustrada na Figura 4.3.

Figura 4.3 – Função de limiar.



Correspondentemente, a saída do neurônio k que emprega esta função de limiar é expressa pela equação 4.5:

$$y_k = \begin{cases} 1, & \text{se } v_k \geq 0 \\ 0, & \text{se } v_k < 0 \end{cases} \quad 4.5$$

Sendo:

v_k : o campo local induzido do neurônio.

Isto pode ser observado pela equação 4.6.

$$v_k = \sum_{j=1}^m w_{kj} \cdot x_j + b_k \quad 4.6$$

Tal neurônio é referido na literatura como o modelo de *McCulloch-Pitts*.

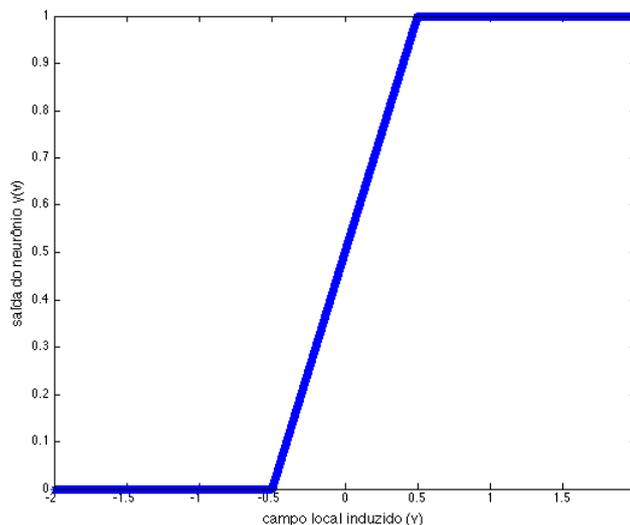
- Função linear por partes

A função linear por partes pode ser descrita pela equação 4.7.

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq +\frac{1}{2} \\ v + \frac{1}{2}, & \text{se } +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & \text{se } v \leq -\frac{1}{2} \end{cases} \quad 4.7$$

A Figura 4.5 mostra a função linear por partes, onde se assume que o fator de amplificação dentro da região linear de operação é a unidade.

Figura 4.4 – Função linear por partes.



- Função sigmóide

A função sigmóide, cujo gráfico tem a forma de S, é a forma mais comum de função de ativação utilizada na construção de redes neurais artificiais. Ela é definida como uma função estritamente crescente que exibe um balanceamento adequado entre comportamento linear e não-linear (Mennon, et al., 1996). Um exemplo de função sigmóide é a função logística, definida pela equação 4.8:

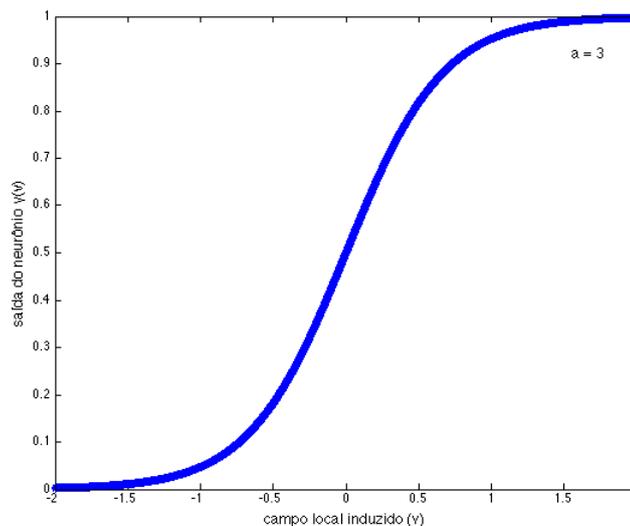
$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad 4.8$$

Sendo:

a : o parâmetro de inclinação da função sigmóide.

Variando-se o parâmetro a , obtêm-se funções sigmóides com diferentes inclinações, como se pode observar na Figura 4.5.

Figura 4.5 – Função sigmóide para parâmetro de inclinação a variável.



A função logística deriva seu nome de uma “lei de crescimento logístico” transcendental que resultou em uma imensa literatura. Se medidos em unidades apropriadas, todos os processos de crescimento são supostamente representados pela função distribuição logística, que é definida pela equação 4.9 (Haykin, 2001).

$$F(t) = \frac{1}{1 + e^{(at-\beta)}} \quad 4.9$$

Sendo:

t : representa o tempo; e

α e β : são constantes.

No limite, quando o parâmetro de inclinação se aproxima do infinito, a função sigmóide se torna simplesmente uma função de limiar. A função de limiar assume o valor de 0 ou 1, e a função sigmóide assume um intervalo contínuo de valores entre 0 e 1.

4.4.4 Algoritmo de retropropagação

O sinal de erro na saída do neurônio j , na iteração n é definido pela equação 4.10; o neurônio j corresponde a um nó de saída (Haykin, 2001):

$$e_j(n) = d_j(n) - y_j(n) \quad 4.10$$

Sendo:

$e_j(n)$: o sinal de erro na saída do neurônio j , para a iteração n ;

$d_j(n)$: a resposta desejada para o neurônio j ; e

$y_j(n)$: o sinal funcional que aparece na saída do neurônio j .

Define-se o valor instantâneo da energia do erro para o neurônio j como sendo (Haykin, 2001): $\frac{1}{2} \cdot e_j^2(n)$.

Correspondentemente, o valor instantâneo $\mathcal{E}(n)$ da energia total do erro é obtido somando-se os termos ($\frac{1}{2} \cdot e_j^2(n)$) de todos os neurônios da camada de saída; sendo assim, pode-se escrever o valor instantâneo da energia total do erro de acordo com a equação 4.11 (Haykin, 2001):

$$\mathcal{E}(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad 4.11$$

Sendo:

C : o conjunto que inclui todos os neurônios da camada de saída da rede.

Considerando que N represente o número total de padrões (exemplos) contidos no conjunto de treinamento; a energia média do erro quadrático é obtida somando-se os $\mathcal{E}(n)$ para todos os n e posteriormente normalizando em relação ao tamanho do conjunto N , como mostrado na equação 4.12 (Haykin, 2001):

$$\mathcal{E}_{med} = \frac{1}{N} \sum_{n=1}^N \mathcal{E}(n) \quad 4.12$$

A energia instantânea do erro $\mathcal{E}(n)$, e conseqüentemente a energia média do erro \mathcal{E}_{med} , é uma função de todos os parâmetros livres, isto é, pesos sinápticos e níveis de *bias*, da rede. Para um dado conjunto de treinamento, \mathcal{E}_{med} representa a função de custo como uma medida do desempenho de aprendizagem. Sob esse ponto de vista, o objetivo do processo de aprendizagem em redes neurais é ajustar os parâmetros livres da rede para minimizar \mathcal{E}_{med} (Haykin, 2001).

Especificamente, considera-se um método simples de treinamento no qual os pesos são atualizados de padrão em padrão até formar uma época, isto é, uma apresentação do conjunto de treinamento que está sendo processado. Os ajustes dos pesos são realizados de acordo com os respectivos erros calculados para cada padrão apresentado à rede. A média aritmética destas alterações individuais de peso sobre o conjunto de treinamento é, portanto, uma estimativa da alteração real que resultaria da modificação dos pesos baseada na minimização da função de custo \mathcal{E}_{med} sobre o conjunto de treinamento (Haykin, 2001).

O campo local induzido $v_j(n)$ produzido na entrada da função de ativação associada ao neurônio j é representado pela equação 4.13 (Haykin, 2001):

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) \cdot y_i(n) \quad 4.13$$

Sendo:

m : o número total de entradas (excluindo o *bias*) aplicadas ao neurônio j .

O peso sináptico w_{j0} (correspondendo à entrada fixa $x_0 = +1$) é igual ao *bias* b_j aplicado ao neurônio j . Assim, o sinal funcional $y_j(n)$ que aparece na saída do neurônio j na iteração n é dado pela equação 4.14 (Haykin, 2001):

$$y_j(n) = \varphi_j(v_j(n)) \quad 4.14$$

O algoritmo de retropropagação aplica uma correção $\Delta w_{ji}(n)$ ao peso sináptico $w_{ji}(n)$, que é proporcional à derivada parcial $\partial \mathcal{E}(n) / \partial w_{ji}(n)$. De acordo com a regra da cadeia, pode-se expressar este gradiente de acordo com a equação 4.15 (Haykin, 2001):

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad 4.15$$

Sendo:

$\mathcal{E}(n)$: o valor instantâneo da energia total do erro na iteração n ;

$e_j(n)$: o sinal de erro na saída do neurônio j , para a iteração n ;

$y_j(n)$: o sinal funcional que aparece na saída do neurônio j , na iteração n ;

$v_j(n)$: o campo local induzido produzido na entrada da função de ativação associada ao neurônio j ; e

$w_{ji}(n)$: o peso sináptico conectado a saída do neurônio i à entrada do neurônio j , na iteração n .

A derivada parcial $\partial \mathcal{E}(n) / \partial w_{ji}(n)$ representa um fator de sensibilidade, determinando a direção de busca no espaço de pesos, para o peso sináptico w_{ji} (Haykin, 2001).

Diferenciando-se ambos os lados da equação 4.11 em relação a $e_j(n)$, obtêm-se a equação 4.16 (Haykin, 2001):

$$\frac{\partial \mathcal{E}(n)}{\partial e_j(n)} = e_j(n) \quad 4.16$$

Diferenciando-se ambos os lados da equação 4.10 em relação a $y_j(n)$, obtêm-se a equação 4.17 (Haykin, 2001):

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \quad 4.17$$

Diferenciando-se a equação 4.14 em relação a $v_j(n)$, obtêm-se a equação 4.18 (Haykin, 2001):

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi'_j(v_j(n)) \quad 4.18$$

Finalmente, diferenciando-se a equação 4.13 em relação a $w_{ji}(n)$, obtêm-se a equação 4.19 (Haykin, 2001):

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \quad 4.19$$

Usando as equações 4.16 e 4.19 na equação 4.15 obtém a equação 4.20 (Haykin, 2001):

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -e_j(n) \cdot \varphi'_j(v_j(n)) \cdot y_i(n) \quad 4.20$$

A correção $\Delta w_{ji}(n)$ aplicada a $w_{ji}(n)$ é definida pela regra delta generalizada mostrada na equação 4.21 (Haykin, 2001):

$$\Delta w_{ji}(n) = -\eta \cdot \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} \quad 4.21$$

Sendo:

η : o parâmetro da taxa de aprendizagem do algoritmo de retropropagação.

O uso do sinal negativo indica a descida do gradiente no espaço de pesos.

Usando a equação 4.20 na equação 4.21, obtêm-se a equação 4.22 para correção do peso sináptico (Haykin, 2001):

$$\Delta w_{ji}(n) = -\eta \cdot \delta_j(n) \cdot y_i(n) \quad 4.22$$

Sendo:

δ : o gradiente local.

O gradiente local $\delta_j(n)$ é definido na equação 4.23 que segue (Haykin, 2001):

$$\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} \quad 4.23$$

Utilizando novamente a regra da cadeia obtém a equação 4.24 para o gradiente local (Haykin, 2001):

$$\delta_j(n) = e_j(n) \cdot \varphi'_j(v_j(n)) \quad 4.24$$

O gradiente local aponta para as modificações necessárias nos pesos sinápticos (Haykin, 2001). De acordo com a equação 4.24, o gradiente local $\delta_j(n)$ para o neurônio de saída j é igual ao produto do sinal de erro $e_j(n)$, correspondentemente para aquele neurônio, pela derivada $\varphi'_j(v_j(n))$ da função de ativação associada.

Das equações 4.22 e 4.24 pode-se notar que um fator-chave envolvido no cálculo do ajuste de peso $\Delta w_{ji}(n)$ é o sinal de erro $e_j(n)$ na saída do neurônio j . Neste contexto, podem-se identificar dois casos distintos, dependendo de onde, na rede, o neurônio j está localizado. No primeiro caso, o neurônio j é um nó de saída. Este caso é simples de se tratar, pois cada nó de saída da rede é suprido com uma resposta desejada particular, fazendo com que o cálculo do sinal de erro associado seja direto. No segundo caso, o neurônio j é um nó intermediário. Apesar dos neurônios intermediários não serem acessíveis diretamente, eles compartilham a responsabilidade por qualquer erro cometido na saída da rede.

4.4.5 Resumo do algoritmo de treinamento da rede neural *Perceptron* multicamadas (MLP)

As variáveis e parâmetros utilizados são:

- ✓ O vetor de entrada ($X(n)$);
- ✓ O vetor de peso ($w(n)$);
- ✓ *Bias* ($b(n)$);
- ✓ Resposta real ($y(n)$);
- ✓ Resposta desejada ($d(n)$); e
- ✓ O parâmetro da taxa de aprendizagem, uma constante positiva menor que a unidade (η).

O algoritmo funciona em cinco etapas, que correspondem a: inicialização, ativação, cálculo da resposta real, adaptação do vetor de peso e continuação, como descrito a seguir:

1. Inicializa-se o vetor de peso $w(0) = 0$. Executa-se, então, os seguintes cálculos para os passos de tempo $n = 1, 2, \dots$
2. No passo de tempo n , ativa-se o *perceptron* aplicando o vetor de entrada de valores contínuos $x(n)$ e a resposta desejada $d(n)$.
3. Calcula-se a resposta real do *perceptron* ($y(n) = \text{senal}[w^T(n) \cdot x(n)]$). Sendo $\text{senal}[\cdot]$ a função sinal.
4. Atualiza-se o vetor de peso do *perceptron* ($w(n+1) = w(n) + \eta[d(n) - y(n)] \cdot x(n)$).
5. Incrementa-se o passo de tempo n em uma unidade e volta para o passo 2 (ativação).

4.4.6 Algoritmo de treinamento Levenberg-Maquardt

O algoritmo de Levenberg-Maquardt foi desenvolvido para resolver iterativamente problemas de minimização de funções não-lineares pelo método de mínimos quadráticos (Winandy, et al., 2007). Este é o algoritmo de otimização mais utilizado hoje, superando outros algoritmos (Campani, 2005) em uma grande variedade de problemas.

Este algoritmo é bastante eficiente quando se trata de redes neurais que não possuem mais do que algumas centenas de conexões a serem ajustadas (Hagan, et al., 1994). Isto se deve, principalmente, ao fato de que algoritmos necessitam armazenar uma matriz quadrada cuja dimensão é da ordem do número de conexões da rede neural (Hagan, et al., 1994).

O funcional de erro apresentado na equação 4.25 representa o erro médio quadrático (MSE) (Hagan, et al., 1994):

$$J(\theta) = \frac{1}{N} \sum_{l=1}^N (g(X) - \hat{g}(X, \theta))^2 \quad 4.25$$

Sendo:

$J(\cdot)$: a superfície do erro de aproximação;

θ : o vetor de parâmetros;

N : o número de amostras;

$g(X)$: a função real; e

$\hat{g}(X, \theta)$: a função fornecida pela rede neural.

Considerando como funcional a soma dos erros médios quadráticos, obtêm-se a equação 4.26 para o funcional de erro (Hagan, et al., 1994):

$$J(\theta) = \sum_{i=1}^N \sum_{j=1}^m (g(X) - \hat{g}(X, \theta))^2 = \sum_{k=1}^q r_k^2 \quad 4.26$$

Sendo:

l : o número de neurônios nas camadas intermediárias;

r : o erro residual;

m : o número de saídas da rede neural; e

q : o produto $N \cdot m$.

Seja J o Jacobiano (matriz das derivadas primeiras) do funcional J dado pela equação 4.26. Esta matriz pode ser escrita como mostra a equação 4.27 (Hagan, et al., 1994):

$$J = \begin{bmatrix} \nabla r_1^T \\ \vdots \\ \nabla r_q^T \end{bmatrix} \quad 4.27$$

Diferenciando-se a equação 4.26 obtêm-se a equação 4.28 (Hagan, et al., 1994):

$$\nabla J = 2J^T r = 2 \sum_{k=1}^q r_k \nabla r_k \quad 4.28$$

A matriz de derivadas segundas do funcional de erro é chamada de matriz hessiana¹. Quando os erros residuais são suficientemente pequenos, a matriz hessiana pode ser aproximada pelo primeiro termo da equação 4.29 (Hagan, et al., 1994), resultando na equação 4.30:

$$\nabla^2 J = 2 \sum_{k=1}^q r_k \nabla r_k \quad 4.29$$

$$\nabla^2 J \approx 2J^T J \quad 4.30$$

Esta aproximação geralmente é válida em um mínimo de J para a maioria dos propósitos, e é a base para o método de Gauss-Newton (Hagan, et al., 1994). A regra de atualização é dada pela equação 4.31:

$$\Delta \theta = [J^T J]^{-1} J^T r \quad 4.31$$

¹ A matriz Hessiana foi desenvolvida no século XIX pelo alemão Ludwing Otto Hesse, razão porque mais tarde James Joseph Sylvester lhe deu este nome. Em matemática, a matriz Hessiana de uma função f de n variáveis é a matriz quadrada com “ n ” colunas e “ n ” linhas ($n \times n$) das derivadas parciais de segunda ordem da função. Por isto, esta matriz descreve a curvatura local da função “ f ” (Winandy, et al., 2007).

A modificação de Levenberg-Maquardt para o método de Gauss-Newton é dada pela equação 4.32 (Hagan, et al., 1994):

$$\Delta\theta = [J^T J + \mu I]^{-1} J^T r \quad 4.32$$

Sendo:

μ : o fator de ajuste.

O efeito da matriz adicional μI é adicionar μ a cada autovalor de $J^T J$. Uma vez que a matriz $J^T J$ é semi-definida positiva e, portanto, o autovalor mínimo possível é zero, qualquer valor positivo, pequeno, mas numericamente significativo, de μ será suficiente para restaurar a matriz aumentada e produzir uma direção descendente de busca (Hagan, et al., 1994).

Os valores de μ podem ser escolhidos de várias maneiras; a mais simples é escolher zero, a menos que a matriz hessiana encontrada na iteração i seja singular. Quando isso ocorrer, um valor pequeno pode ser usado (Hagan, et al., 1994).

É importante observar que, quanto maior for o valor de μ , menor é a influência da informação de segunda ordem e mais este algoritmo se aproxima de um método de primeira ordem (Hagan, et al., 1994).

Pode-se observar que a regra de atualização leva em consideração tanto a inclinação da superfície de erro (método do gradiente descendente) quanto à curvatura desta superfície (método de Gauss-Newton).

No método do gradiente descendente o vetor de pesos sinápticos é modificado sempre na direção contrária à direção de maior aumento do erro médio quadrático, caminhando assim para minimizar o erro médio quadrático e fazer com que a rede neural funcione como se deseja (Winandy, et al., 2007). Por sua vez, o método de Gauss-Newton pode ser visto como uma modificação do método de Newton para achar o mínimo de uma função (Nocedal, et al., 1999).

O fator de ajuste indica qual dos dois métodos será predominante: para fatores de ajustes grandes, o método do gradiente descendente predomina e a atualização dos pesos ocorre fortemente na direção de inclinação da superfície do erro; caso contrário, o método de Gauss-Newton predomina e a atualização ocorre mais no sentido da curvatura da função (Winandy, et al., 2007).

O algoritmo controla o valor do fator de ajuste da seguinte forma (Winandy, et al., 2007):

- ✓ Inicia-se com um valor arbitrário;
- ✓ Calcula-se o erro na situação atual e aplica-se a regra de atualização de pesos;
- ✓ Calcula-se então o novo erro;

- ✓ Caso o erro tenha aumentado, desfaz-se a atualização e aumenta-se o fator de ajuste (geralmente multiplicando o fator de ajuste por 10);
- ✓ Em seguida, deve-se recomeçar a iteração;
- ✓ Caso o erro tenha diminuído, aceita-se a iteração e diminui-se o fator de ajuste (geralmente dividindo o fator de ajuste por 10).

Desta maneira, a atualização dos pesos caminhará mais na direção do gradiente quanto mais distante se estiver do ponto mínimo. Ao se aproximar do mínimo global, o algoritmo de Gauss-Newton será predominante, o que faz com que o algoritmo de Levenberg-Marquardt funcione utilizando o que cada um dos algoritmos anteriores tem de melhor (Winandy, et al., 2007).

Resumidamente, o método de Levenberg-Marquardt é usado para encontrar um ponto de mínimo para uma função de ativação dada pela soma dos quadrados de funções não-lineares. O método consiste em um aperfeiçoamento do método de Gauss-Newton que, por sua vez, é uma variante do método de Newton. Assim como os antecessores, Levenberg-Marquardt é um método iterativo. Isto significa que, dado um ponto inicial, o método produz uma série de vetores que se espera que vá convergir para um mínimo local para a função de entrada a ser ajustada (Luo, et al., 2007).

4.4.7 Resumo do algoritmo de treinamento Levenberg-Marquardt

O algoritmo de treinamento Levenberg-Marquardt está resumido abaixo:

1. Atribui-se um valor inicial $\theta_0 \in \mathfrak{R}^P$ para o vetor de parâmetros e um valor arbitrariamente pequeno para uma constante $\varepsilon > 0$.
2. Enquanto a condição de parada não for satisfeita:
 - a. Constrói-se a matriz J utilizando as equações 4.25 e 4.26.
 - b. Calcula-se o ajuste utilizando a equação 4.32.
 - c. Utiliza-se um procedimento de busca unidimensional para encontrar um α_i que seja solução ótima do problema $\min_{\alpha_i \in (0,1]} J(\theta_i + \alpha_i d_i)$.
 - d. Lei de ajuste: $\theta_{i+1} = \theta_i - \alpha_i \Delta(\theta_i)$.

4.5 CONSIDERAÇÕES FINAIS

As redes neurais *perceptrons* multicamadas mostram que é possível construir máquinas inteligentes, capazes de aprender, de errar, e principalmente de se adaptarem. Neste capítulo foi apresentado um breve histórico sobre as redes neurais, bem como seus fundamentos, os benefícios de seu uso, a relação das redes com a inteligência artificial, e o processo de aprendizagem. Foi apresentada também a rede *perceptron* multicamadas; o modelo de neurônio para este tipo de rede; os tipos de função de ativação; o algoritmo de retropropagação; um resumo do algoritmo de treinamento da rede *perceptron* multicamada; e o algoritmo de Levenberg-Maquardt.

CAPÍTULO 5 – COMPRESSÃO DE DADOS

5.1 INTRODUÇÃO

Nos últimos anos, com o aumento de cargas eletrônicas sensíveis, as empresas do setor elétrico têm dado mais atenção à qualidade da energia elétrica, tornando-se necessário um sistema de monitoramento, que grave grande volume de dados durante a perturbação do sistema, a fim de analisar e avaliar a energia fornecida aos consumidores de grande, médio e pequeno porte. Como resultado de um processo contínuo de armazenamento dos diversos sinais de energia elétrica, é gerado um grande volume de dados armazenados, o que torna desejável o desenvolvimento de técnicas capazes de reduzir o tamanho desses arquivos, e conseqüentemente reduzir os custos para esta finalidade.

5.2 BREVE HISTÓRICO

O código Morse, desenvolvido em 1835 por Samuel Morse, é um sistema binário de representação de números e sinais de pontuação por meio de um sinal codificado enviado intermitentemente. Morse aproveitou o fato de que existem letras do alfabeto que são usadas com maior frequência (no inglês) para atribuir códigos de comprimento variável (Cormane, 2013). Segundo Wolfram (2002), o código Morse foi um dos primeiros exemplos de compressão de dados baseado em pequenos dicionários de palavras.

Apesar do código Morse ter sido desenvolvido a cerca de 175 anos, somente na década de 1940 foi que a compressão de dados começou a ser desenvolvida como uma área distinta, em consequência do desenvolvimento da teoria da informação (Sayood, 2006).

Em 1948, Claude Shannon definiu matematicamente a quantidade de informação contida em um experimento aleatório por meio de um parâmetro chamado de autoinformação, e com esse resultado foi gerada uma base teórica para a compressão de dados (Sayood, 2006).

Em 1952, David Huffman desenvolveu um método de compressão, que utilizava as probabilidades de ocorrência dos símbolos no conjunto de dados a ser comprimido, para determinar códigos de tamanho variável para cada símbolo (Schmidt, 2007).

Na metade dos anos de 1970, surgiu a ideia de atualizar dinamicamente os dicionários de palavras do método de Huffman (Schmidt, 2007). No final desta mesma década, começou a se tornar comum o armazenamento *on-line* de textos, conseqüentemente, diversos programas de compressão de dados começaram a ser desenvolvidos, e em sua maioria, baseados no código adaptativo de Huffman (Schmidt, 2007).

Na metade dos anos de 1980, Terry Welch desenvolveu e patenteou o algoritmo de compressão de dados *Lempel-Ziv-Welch* (LZW), que rapidamente se tornou o método de compressão mais escolhido para o uso geral em sistemas de compressão. O LZW baseia-se na localização e no registro das padronagens de uma estrutura. É geralmente utilizado em imagens, nas quais não se pode perder a definição original. Atualmente os formatos que utilizam esse padrão são: o TIFF, por opção; e o GIF, por padrão (Schmidt, 2007).

No final dos anos de 1980, imagens digitais começaram a se tornar mais comuns, e os padrões para comprimi-las surgiram (Schmidt, 2007).

No começo dos anos de 1990, métodos de compressão com perda de informação também começaram a ser amplamente utilizados (Schmidt, 2007).

5.3 MÉTODOS DE COMPRESSÃO DE DADOS

A compressão de dados é um ramo da teoria da informação que tem como principal objetivo reduzir o tamanho da representação da informação, tanto para armazenamento, como para envio por um canal de comunicação, reduzindo assim os custos com transmissão ao economizar a banda do canal (Campani, 2005).

A compressão de dados é o ato de reduzir o espaço ocupado por dados num determinado dispositivo (Lelewer, et al., 1987). Essa operação é realizada por meio de diversos algoritmos de compressão, reduzindo a quantidade de *bytes* para representar um dado, podendo esse dado ser uma imagem, um texto, ou um arquivo qualquer (Salomon, 2000). Um algoritmo de compressão é considerado eficiente quando consegue subtrair a informação relevante e estatisticamente não redundante do total de dados, diminuindo a quantidade de *bits* da representação (Cormane, 2013).

Comprimir dados destina-se também a retirar a redundância, pois muitos dados contêm informações redundantes que podem ou precisam ser eliminadas (Salomon, 2000).

A quantidade ou qualidade de compressão produzida por uma codificação pode ser medida pela taxa de compressão (Lelewer, et al., 1987), dada em termos percentuais pela Equação 5.1.

$$CR(\%) = \frac{(Sinal_{original}) - (Sinal_{comprimido})}{(Sinal_{original})} \cdot 100 \quad 5.1$$

Sendo:

$CR(\%)$: a taxa de compressão do sinal em valor percentual.

Embora possam parecer sinônimos, compressão e compactação de dados são processos distintos. A compressão tem como objetivo reduzir a quantidade de *bits* para

representar algum dado, enquanto a compactação tem a função de unir dados que não estejam unidos (Salomon, 2000).

Existem diversas formas de classificar os métodos de compressão de dados. O mais conhecido é pela ocorrência ou não de perdas de dados durante o processo de compressão. Entretanto, diversas outras formas de classificações são úteis para avaliar e comparar os métodos de compressão de dados, e sua aplicação em problemas específicos (Salomon, 2000).

5.3.1 Compressão sem perdas e com perdas

Esta é a forma mais conhecida de se classificar os métodos de compressão de dados.

- Compressão sem perdas

Diz-se que um método de compressão é sem perdas (*loss/less*) se os dados obtidos após a descompressão são idênticos aos dados originais. A compressão sem perdas envolve algoritmos que permitem a reconstrução exata dos dados originais a partir dos dados comprimidos, são métodos de compressão considerados reversíveis. Esses métodos são úteis para dados que são obtidos diretamente por meios digitais, como textos, programas de computador, planilhas eletrônicas, entre outros, em que uma pequena perda de dados acarreta o não funcionamento ou torna os dados incompreensíveis. Como exemplo: texto com letras trocadas, planilha com valores faltantes ou inexatos, ou programa de computador com comandos inválidos (Salomon, 2000).

Os métodos de compressão sem perdas podem ser classificados em três categorias, segundo sua operação (Cormane, 2013): os **métodos por substituição de sequência de símbolos repetidos**, os **métodos estatísticos**, e os **métodos que utilizam dicionário de dados**.

1. Métodos por substituição de sequência de símbolos repetidos

Os métodos que pertencem a esta categoria empregam códigos de comprimento fixo na representação dos símbolos, além de utilizar um caractere especial para indicar quando o símbolo repetido aparece.

2. Métodos estatísticos ou de codificação de entropia

Visa alocar uma quantidade variável de *bits* durante a codificação, aproximando o comprimento médio por símbolo ou grupo de símbolos ao valor da entropia². Para isso, uma tabela de códigos predefinida é ordenada de forma decrescente de acordo com o comprimento de cada código, onde os símbolos mais prováveis usam códigos de menor comprimento (menos *bits*) que os menos prováveis (mais *bits*). Os **métodos de Huffman** e a **codificação aritmética** são os métodos estatísticos mais utilizados.

3. Métodos de compressão que utilizam dicionário de dados

Utilizam a seleção de sequências de símbolos que são substituídas por índices associados a sequências similares contidas no dicionário. Na sequência, os índices são codificados.

Os dicionários de dados são classificados em duas categorias segundo a forma como são atualizados (Cormane, 2013): **estático**, quando seus elementos são permanentes ou somente permite a adição de novas sequências (sem exclusões); e **dinâmico** ou **adaptativo**, quando seus elementos são atualizados de acordo com as sequências encontradas no fluxo de dados a ser codificado, permitindo a adição ou exclusão de sequências.

- Compressão com perdas

Algumas situações permitem que perdas de dados poucos significativos ocorram. Em geral, quando se digitaliza informações que normalmente existem de forma analógica, como fotografias, sons e filmes, podem-se considerar algumas perdas que não seriam percebidas pelo olho ou ouvido humano. Sons de frequências muito altas ou muito baixas que os humanos não ouvem, detalhes muito sutis como a diferença de cor entre duas folhas de uma árvore, movimentos muito rápidos que não se consegue acompanhar num filme, todos estes detalhes podem ser omitidos sem que as pessoas percebam que eles não estão lá. Nesses casos, podem-se comprimir os dados simplesmente por omitir tais detalhes. Assim, os dados obtidos após a compressão não são idênticos aos originais, pois não possuem mais as informações irrelevantes. O método descrito é o método de compressão com

² O termo entropia corresponde a mínima taxa de codificação com que os símbolos da fonte podem ser codificados (Cormane, 2013), ou seja, a entropia é a menor taxa de codificação possível, alcançada pelo menor esquema de compressão sem perdas (Sayood, 2006).

perdas (*lossy*) (Salomon, 2000). Esses métodos objetivam encontrar a melhor relação taxa-distorção para amenizar o impacto da perda de informação (Sayood, 2006).

A compressão com perdas se faz útil quando a reconstrução exata dos dados não é indispensável (Wiegand, et al., 2011). Os métodos de compressão com esse enfoque são ajustados para diferentes níveis de qualidade por meio da regulação da distorção e da taxa de codificação (Cormane, 2013).

Existem três tipos gerais de técnicas de compressão com perdas (Cormane, 2013): **codificação por transformadas, codificação diferencial e quantização diferencial.**

1. Codificação por transformadas

Considera-se a aplicação de transformadas lineares uma técnica de compressão natural, isto porque a energia se concentra em poucos coeficientes durante o mapeamento entre domínios (Cormane, 2013). No domínio da transformada, os coeficientes de interesse são isolados, quantizados e codificados usando técnicas convencionais de compressão sem perdas, enquanto que os restantes são descartados (Cormane, 2013). A fidelidade da reconstrução dos dados depende dos níveis de quantização e da quantidade de coeficientes descartados.

2. Codificação diferencial

A codificação diferencial baseia-se na predição do valor de cada amostra com base no valor da amostra anterior. Neste tipo de codificação, apenas são codificadas as diferenças entre um dado e seu vizinho imediatamente anterior (ou um valor de referência). A técnica é aplicada a sequências que não apresentam muita variação. Desta forma, a diferença entre os dois valores consecutivos é pequena, o que permite utilizar menos *bits* na representação (Cormane, 2013).

3. Quantização diferencial

De modo geral, a quantização é o processo de representação de cada número produzido pelo amostrador no nível mais próximo escolhido dentre um número finito de níveis de amplitude discreta (Haykin, et al., 2002). Por ser um processo de aproximação, existe um erro associado ao processo que deve ser minimizado para garantir a boa qualidade da nova representação. Desta forma, considera-se uma boa quantização quando o erro de quantização tende a zero (Cormane, 2013).

Na quantização vetorial, os quantizadores são armazenados na forma de um vetor, onde o número de posições depende da quantidade de blocos gerados na sequência, de tal forma, que as sequências de dados estatisticamente semelhantes usam o

mesmo vetor durante a quantização, preservando as características comportamentais dos dados pela adaptabilidade do processo (Cormane, 2013).

A principal vantagem dos métodos de compressão com perda em relação aos métodos de compressão sem perdas é que no primeiro método consegue-se maior ganho de compressão, mantendo, no entanto, uma qualidade mínima em relação aos dados originais (Cormane, 2013).

5.3.2 Compressão simétrica e assimétrica

Quando fala-se em métodos simétricos ou assimétricos de compressão de dados, fala-se nas diferenças de complexidade entre a compressão e a descompressão. Quando a compressão e a descompressão são feitas executando-se métodos ou algoritmos idênticos ou bem semelhantes, diz-se que o método de compressão é simétrico (Salomon, 2000). Bons exemplos são os algoritmos de codificação aritmética, ou o método LZW, baseado em dicionários. Neles os algoritmos de compressão e de descompressão são praticamente idênticos, e apresentam a mesma complexidade.

Quando o método de compressão é mais complexo que o de descompressão (ou em casos raros, o de descompressão é mais complexo que o de compressão), diz-se que o método de compressão é assimétrico. Este tipo de método é útil quando deseja-se comprimir apenas uma vez, mas descomprimir várias vezes, como por exemplo, as músicas em formato MP3 (Salomon, 2000).

5.3.3 Compressão não-adaptativa e adaptativa

A compressão de dados pode ser baseada em métodos rígidos, cujas regras não variam de acordo com os dados, nem à medida que os dados são lidos. São os métodos não adaptativos. Por outro lado, diversos métodos conseguem ir se adaptando aos dados à medida que estes são processados. Nesse caso, o método é adaptativo. Métodos baseados em dicionário de dados são naturalmente adaptativos, já que é inviável que os programas de compressão carreguem dicionários de dados padronizados, ou que os dicionários sejam enviados junto com os arquivos (Salomon, 2000).

5.4 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado um breve histórico sobre a compressão de dados, bem como os principais métodos de compressão existentes e suas classificações: compressão sem perdas e com perdas; compressão simétrica e assimétrica; e compressão não-adaptativa e adaptativa. Com base nas classificações apresentadas na literatura, a metodologia desenvolvida neste trabalho, para compressão de sinais de energia elétrica, pode ser classificada como: compressão com perdas, assimétrica e adaptativa.

CAPÍTULO 6 – COMPRESSÃO DE SINAIS DE ENERGIA ELÉTRICA UTILIZANDO ALGORITMOS GENÉTICOS E REDES NEURAIS ARTIFICIAIS

6.1 INTRODUÇÃO

No presente capítulo segue uma descrição da metodologia para compressão de sinais de energia elétrica, provenientes de oscilografias digitais, utilizando algoritmos genéticos e redes neurais artificiais. O cerne desta metodologia está no uso de algoritmos genéticos para selecionar os pontos mais relevantes do sinal, preservando seus principais contornos e possibilitando a inserção de uma rede neural MLP, como interpolador, no processo de compressão e reconstrução do sinal comprimido. Neste sentido, o sinal comprimido é formado pelos pontos selecionados pelo AG e pelos pesos sinápticos da rede neural, que contém informações para a interpolação dos pontos referentes a todos os sinais de tensão e corrente presentes no arquivo COMTRADE.

Levantamentos bibliográficos realizados sobre o tema “compressão de sinais de energia elétrica”, mostram que as pesquisas realizadas e os trabalhos desenvolvidos na área têm utilizado outras técnicas de compressão, como transformada *wavelet* e dicionário de dados, não utilizando algoritmos genéticos e redes neurais artificiais conjuntamente para compressão e descompressão desses sinais, o que, por sua vez, caracteriza-se como uma estratégia inovadora na área de compressão de sinais de energia elétrica.

6.2 METODOLOGIA PROPOSTA

Registradores digitais de perturbação e unidades de proteção digital são amplamente utilizados em sistemas de energia elétrica para registrar sinais de tensão e corrente e sinais digitais (de funções de proteção, disjuntores, entre outros) durante a ocorrência de perturbações em sistemas elétricos de potência. As amostras são adquiridas com uma boa resolução no domínio do tempo para a maioria das aplicações de detecção de distúrbios e análise (Grigsby, 2012). Em muitas aplicações, as informações provenientes de dados oscilográficos podem ser usadas para detectar ocorrência de falhas no sistema elétrico, fornecendo informações sobre a sua duração, severidade e tipo (monofásico, bifásico, trifásico), a sua localização, e mesmo o desempenho de funcionamento de relés e disjuntores (Grigsby, 2012). Essas aplicações tornam os registros de dados oscilográficos muito importante para o funcionamento dos sistemas de energia, especialmente na análise de pós-operação.

Desde 1980, o número de registros de forma de onda tem aumentado devido à expansão dos sistemas de energia elétrica e aos investimentos na instalação de RDPs e UPDs. Neste cenário, a utilização de técnicas de compressão de dados é muito oportuna.

Tcheou (Tcheou, et al., 2014) apresentaram uma revisão das principais técnicas de compressão desenvolvidas para aplicações em formas de onda de sinais de energia elétrica que fornece uma visão geral dos avanços obtidos nas últimas décadas. As metodologias para compressão de sinais de energia elétrica discutidas por Tcheou são baseadas em: **técnica LZW** (Lempel-Ziv-Welch) (Gerek, et al., 2008) (Lorio, et al., 2004); **transformada discreta de cosseno** (Qing, et al., 2011) (Nascimento, 1997); **codificação por transformada** (Tcheou, et al., 2012) (Ribeiro, et al., 2001) (Ribeiro, et al., 2002) (Ramos, et al., 2002) (Ribeiro, et al., 2004) (Ribeiro, et al., 2007) (Zhang, et al., 2011); **abordagens da codificação Huffman** (Lorio, et al., 2004) (Zhang, et al., 2009) (Hsieh, et al., 2003); **EZW** (Chung, et al., 1999) (Lorio, et al., 2004); **algoritmo *matching pursuits*** (Lovisollo, et al., 2005) (Mallat, et al., 1993); **transformada *wavelet packet*** (Ning, et al., 2011) (Bingham, et al., 1995) (Santoso, et al., 1997) (Hsieh, et al., 1998) (Littler, et al., 1999) (Hamid, et al., 2002) (Wu, et al., 2003) (Shang, et al., 2003) (Gerek, et al., 2004) (Huang, et al., 2004) (Yuan, et al., 2006); **transformada *wavelet* discreta (DTW)** (Panda, et al., 2002); **transformada *wavelet*, interpolação *spline* e redes neurais artificiais** (Gasparese, 2010) (Dash, et al., 2003) (Meher, et al., 2004).

As metodologias de compressão com perdas citadas no parágrafo anterior não apresentam a habilidade de selecionar e preservar os pontos que melhor caracterizam os contornos do sinal original. Preservar estes pontos significa preservar a informação que caracteriza o contorno do sinal original, ou seja, para esses pontos não há perda de informação. Considerando esse aspecto, este trabalho propõe uma estratégia inovadora que emprega algoritmos genéticos e redes neurais artificiais para compressão e descompressão de sinais de energia elétrica. Esta metodologia utiliza o AG para selecionar um percentual de amostras (pontos) que melhor caracterizam os contornos do sinal original, e uma rede neural para comprimir e descomprimir os demais pontos. Selecionar amostras (pontos) que melhor caracterizam os contornos do sinal original e preservar um percentual do sinal original são aspectos não explorados na literatura vigente. Adicionalmente, como pode ser visto nas referências bibliográficas citadas previamente, metodologias de compressão de sinais de energia elétrica não usam AG, ou AG em conjunto com RNA como ferramenta de compressão e descompressão.

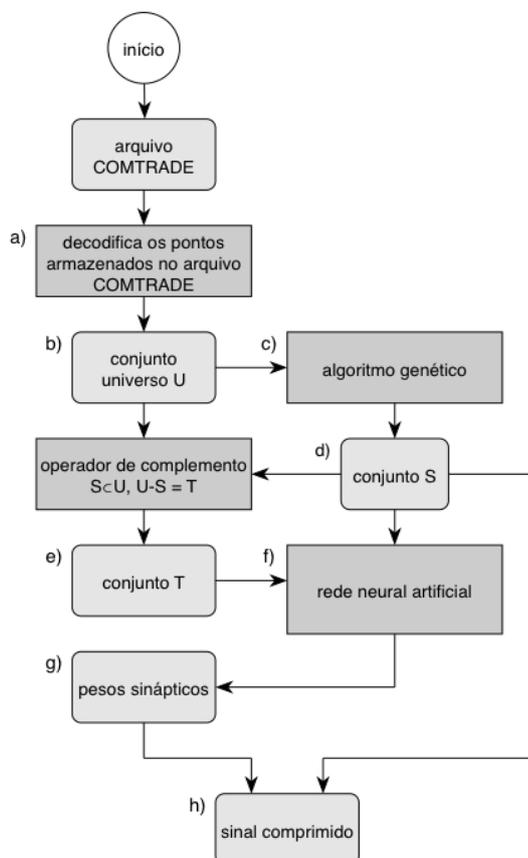
6.2.1 Compressão de dados utilizando algoritmos genéticos e redes neurais artificiais

A motivação para a utilização de algoritmos genéticos e redes neurais artificiais na compressão e descompressão de sinais de energia elétrica se deve ao fato de que os algoritmos genéticos são uma técnica de busca eficiente em explorar o espaço de soluções para obtenção de soluções próximas da ótima em problemas de otimização (Linden, 2012), além da capacidade das redes neurais artificiais em aprender e generalizar (Haykin, 2001).

Na metodologia desenvolvida, o AG é utilizado para selecionar quais são os pontos que melhor caracterizam os contornos da forma de onda, e a RNA atua como um interpolador para os pontos selecionados pelo AG.

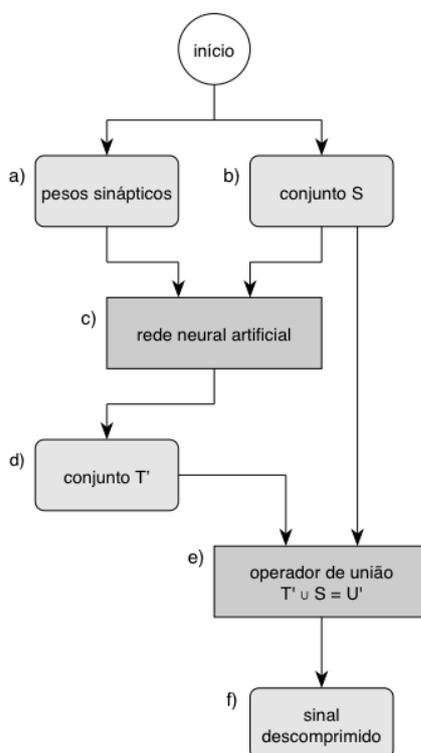
A metodologia é dividida em dois módulos: o primeiro, mostrado na Figura 6.1, é responsável pela compressão de dados, e o segundo, mostrado na Figura 6.2, é responsável pela descompressão.

Figura 6.1 – Diagrama de blocos da metodologia para compressão de sinais de energia elétrica a partir de arquivos COMTRADE.



O bloco (a) (Figura 6.1) decodifica os pontos armazenados no arquivo COMTRADE que contém as formas de onda de tensão ou corrente registradas durante o evento. Estes pontos, representados na Figura 6.1 – b, formam o espaço de busca, ou conjunto universo U , do AG. O AG (Figura 6.1 – c) visa selecionar um conjunto S de pontos que melhor caracterizam os contornos dos sinais de tensão ou corrente. Portanto, cada cromossomo deve representar um conjunto de pontos, como um possível candidato para a solução. Uma RNA MLP é treinada para executar a interpolação entre os pontos selecionados pelo AG. O processo de treinamento utiliza os pontos do conjunto S (Figura 6.1 – d) e os pontos do conjunto T (Figura 6.1 – e), complemento do conjunto S em U . O treinamento da rede neural (Figura 6.1 – f) produz valores de pesos sinápticos (Figura 6.1 – g) que contém informações sobre os pontos não escolhidos pelo AG, de modo a permitir a reconstrução dos sinais de tensão ou corrente através de interpolação. Finalmente, os pontos selecionados pelo AG e os pesos sinápticos da rede neural formam o sinal comprimido (Figura 6.1 – h).

Figura 6.2 – Diagrama de blocos da metodologia para descompressão de sinais de energia elétrica a partir de arquivos COMTRADE.



O segundo módulo descomprime os sinais de tensão ou corrente. Para reconstruir os sinais, este módulo utiliza os pesos sinápticos (Figura 6.2 – a) ajustados no primeiro módulo, e os pontos no conjunto S (Figura 6.2 – b), que foram selecionados pelo AG. A rede neural usa os pesos sinápticos ajustados em sua estrutura e os pontos do conjunto S como sinais de entrada. A RNA (Figura 6.2 – c) determina, por interpolação, uma aproximação do

conjunto complementar (conjunto T') de S , chamado T' (Figura 6.2 – d). Ao final do processo, é realizada uma operação de união (Figura 6.2 – e) entre os conjuntos T' e S (Figura 6.2 – b), resultando no conjunto U' cujos pontos formam o sinal descomprimido (Figura 6.2 – f).

6.2.2 Representação cromossomial

A representação cromossomial, neste trabalho, consiste em uma forma de representar os pontos a serem selecionados pelo AG. As ordenadas dos pontos são os valores dos sinais de tensão ou corrente. As abscissas são valores que representam o instante em que cada ordenada foi amostrada. Como os sinais de tensão e corrente são amostrados sequencialmente ao longo do tempo e com uma frequência de amostragem fixa (f), os valores das abscissas formam uma progressão aritmética com uma relação igual a $\Delta t = 1/f$. Portanto, o valor de cada ponto é uma função da posição da amostra, definida pela sequência de amostragem, como representado pela equação 6.1.

$$t(n) = t(1) + (n - 1) \cdot \Delta t \quad 6.1$$

Sendo:

n : a posição da abscissa na sequência de amostragem;

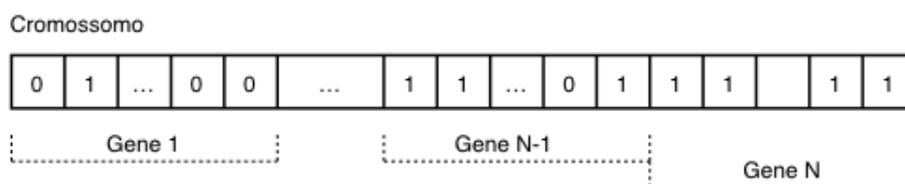
$t(n)$: a abscissa correspondente à posição n na sequência de amostragem;

$t(1)$: a abscissa correspondente à posição $n = 1$ na sequência de amostragem; e

Δt : o período de amostragem.

O sinal amostrado atende aos conceitos de função dado em (Guidorizzi, 2008), e existe uma relação biunívoca, dada pela equação 6.1, entre as abscissas e suas posições na sequência de amostragem. Assim, os pontos para o AG podem ser representados no cromossomo pelas suas posições n na sequência de amostragem. Então, o cromossomo é dividido em genes (Figura 6.3), em que cada gene codifica a posição da abscissa (n) por meio de 0s e 1s.

Figura 6.3 – Representação genética do cromossomo.



O número de genes no cromossomo é definido pela equação 6.2. Assim, o cromossomo terá $(Q_{bits} \cdot N)$ bits.

$$N = V_p \cdot \left(\frac{Q_s}{100} \right) \quad 6.2$$

Sendo:

N : o número de genes;

V_p : o valor em percentual, definido pela quantidade de pontos do sinal originalmente amostrado;

Q_s : a quantidade de pontos do sinal originalmente amostrado.

O número de bits em cada gene é definido pela equação 6.3 (Fonseca, 2012) como segue:

$$Q_{bits} = \log_2[(V_{sup} - V_{inf}) \cdot 10^{ps} + 1] \quad 6.3$$

Sendo:

Q_{bits} : o número de bits do gene;

V_{inf} : o menor valor do intervalo $[V_{inf}, V_{sup}]$, e é igual a 1 por se tratar da primeira amostra;

V_{sup} : o maior valor do intervalo $[V_{inf}, V_{sup}]$, e é igual ao número de amostras; e

ps : a precisão, dada pela quantidade de casas decimais; e é igual a 0 por se tratar da representação de números inteiros.

Finalmente, o conjunto S é formalmente definido, neste trabalho, como o conjunto dos pontos G_k representados no cromossomo, isto é: $S = \{G_k | 1 \leq k \leq N\}$.

k é a posição dos pontos, no conjunto S , ordenados de acordo com a sequência de amostragem, então, para $G_k(t(p), a_p)$ e $G_{k+1}(t(q), a_q)$ não existe um ponto $Z(t(z), a_z)$ tal que $p < z < q$.

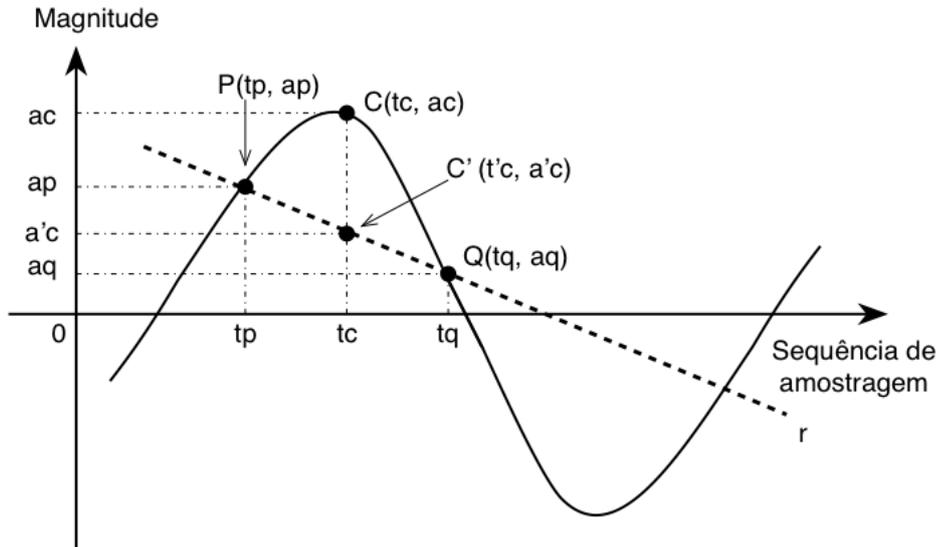
6.2.3 Função objetivo

A função objetivo, proposta para o AG, visa selecionar os pontos que melhor caracterizam os contornos dos sinais de tensão e corrente. A Figura 6.4 ilustra como o AG seleciona esses pontos.

$P(t_p, a_p)$ e $Q(t_q, a_q)$ são pontos que pertencem ao conjunto S , e dentro deste conjunto, são consecutivos com relação a sequência de amostragem, então, $P = G_k$ e $Q = G_{k+1}$. Os valores de $t(p)$, $t(c)$ e $t(q)$, também conhecidos como t_p , t_c e t_q respectivamente,

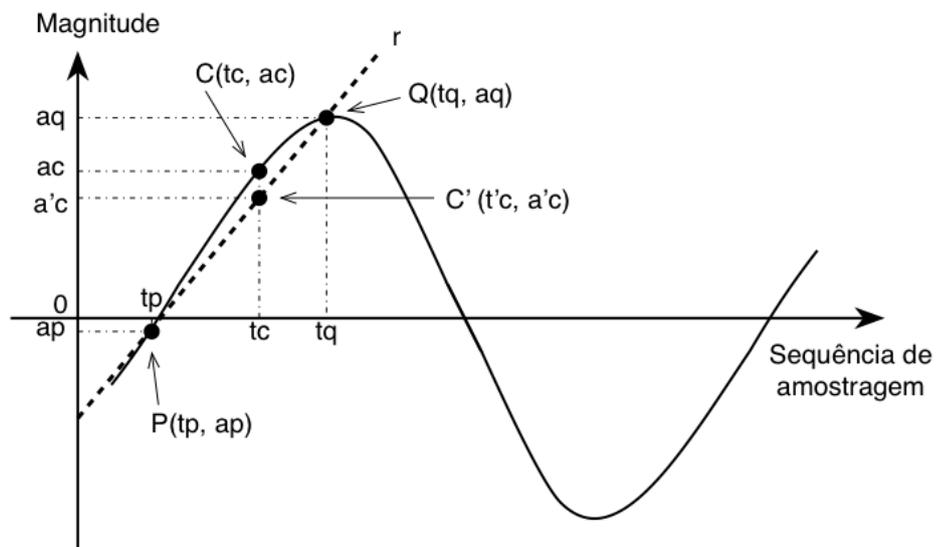
são definidos pela equação 6.1. Sendo $C(t_c, a_c)$ um ponto que pertence ao conjunto T , complementar de S em U , tal que $p < c < q$, como na Figura 6.4.

Figura 6.4 – Representação da seleção de pontos utilizando algoritmos genéticos.



O AG seleciona os pontos P e Q de modo que o segmento de reta \overline{PQ} seja a representação linear mais próxima do sinal original (Figura 6.5), entre as abscissas t_p e t_q . Quanto mais próximo o segmento de reta \overline{PQ} estiver do sinal original, melhor será a sua representação. Neste sentido, o objetivo é reduzir a distância entre os pontos representados por C , pertencente ao sinal original, e os pontos representados por C' , sobre o segmento de reta \overline{PQ} .

Figura 6.5 – Representação linear mais próxima do sinal original utilizando algoritmos genéticos.



A distância Euclidiana entre os pontos C e C' é definida pela equação 6.4 (Iezzi, 2005), que resulta na equação 6.5, após manipulações algébricas.

$$d_{C,C'} = \sqrt{(t_c - t'_c)^2 + (a_c - a'_c)^2} \quad 6.4$$

Sendo:

$$t_c = t'_c$$

$$d_{C,C'} = |a_c - a'_c| \quad 6.5$$

r (Figura 6.4) é uma reta definida pelos pontos P e Q . Como P , C' e Q são colineares (Iezzi, 2005), tem-se:

$$\begin{vmatrix} t_c & a'_c & 1 \\ t_p & a_p & 1 \\ t_q & a_q & 1 \end{vmatrix} = 0$$

Desenvolvendo o determinante pelo teorema de Laplace aplicado à primeira linha do determinante (Iezzi, 2005), obtém-se:

$$\begin{vmatrix} a_p & 1 \\ a_q & 1 \end{vmatrix} \cdot t_c - \begin{vmatrix} t_p & 1 \\ t_q & 1 \end{vmatrix} \cdot a'_c + \begin{vmatrix} t_p & a_p \\ t_q & a_q \end{vmatrix} \cdot 1 = 0$$

Desta forma, a equação 6.6 (Iezzi, 2005) define a reta r (Figura 6.4), obtida a partir dos pontos $P(t_p, a_p)$ e $Q(t_q, a_q)$.

$$(a_p - a_q) \cdot t_c + (t_q - t_p) \cdot a'_c + (t_p \cdot a_q - t_q \cdot a_p) = 0 \quad 6.6$$

Como o ponto C' pertence à reta r , seu valor de ordenada a'_c (equação 6.7) é determinado a partir da equação 6.6, ou seja:

$$(a_p - a_q) \cdot t_c + (t_q - t_p) \cdot a'_c + (t_p \cdot a_q - t_q \cdot a_p) = 0$$

$$(t_q - t_p) \cdot a'_c = a_q \cdot t_c - a_p \cdot t_c + t_q \cdot a_p - t_p \cdot a_q$$

$$a'_c = \frac{a_q \cdot (t_c - t_p) + a_p \cdot (t_q - t_c)}{t_q - t_p} \quad 6.7$$

Substituindo a equação 6.7 na equação 6.5, tem-se (equação 6.8):

$$d_{C,C'} = \left| a_c - \frac{a_q \cdot (t_c - t_p) + a_p \cdot (t_q - t_c)}{t_q - t_p} \right| \quad 6.8$$

A soma da distância Euclidiana entre os pontos representados por C , pertencentes ao sinal original, e os pontos representados por C' , sobre o segmento de reta \overline{PQ} , é definida pela equação 6.9.

$$d_{C,C'} = \sum_{c=p}^q \left| a_c - \frac{a_q \cdot (t_c - t_p) + a_p \cdot (t_q - t_c)}{t_q - t_p} \right| \quad 6.9$$

A função objetivo, dada pela equação 6.10, a seguir, é definida como a soma dos valores obtidos pela aplicação da equação 6.9 a todos os segmentos de reta formados por dois pontos consecutivos, G_k e G_{k+1} , pertencentes ao conjunto S . Quanto menor for o valor da função objetivo, melhor a aptidão do cromossomo, isto é, melhor é a representação linear do sinal.

$$V_{fitness} = \sum_{k=1}^{k=N-1} \left[\sum_{c=p}^q \left| a_c - \frac{a_q \cdot (t_c - t_p) + a_p \cdot (t_q - t_c)}{t_q - t_p} \right| \right] \quad 6.10$$

$P = G_k;$
 $Q = G_{k+1}$

Sendo:

$V_{fitness}$: o valor de aptidão do cromossomo.

É importante mencionar que os pontos pertencentes ao conjunto S , selecionados pelo AG, tendo a equação 6.10 como função objetivo, correspondem a um percentual V_p do total de pontos que pertencem ao sinal original. Estes pontos representam uma aproximação linear otimizada, isto é, que melhor caracterizam os contornos da forma de onda para a quantidade de pontos definida pelo usuário através de V_p .

Finalmente, qualquer que seja o sinal (tensão, corrente, fase A, fase B ou fase C), a seleção de pontos utiliza o mesmo princípio, isto é, o algoritmo genético utilizando a representação cromossomial descrita na seção 6.2.2 e a função objetivo dada pela equação 6.10.

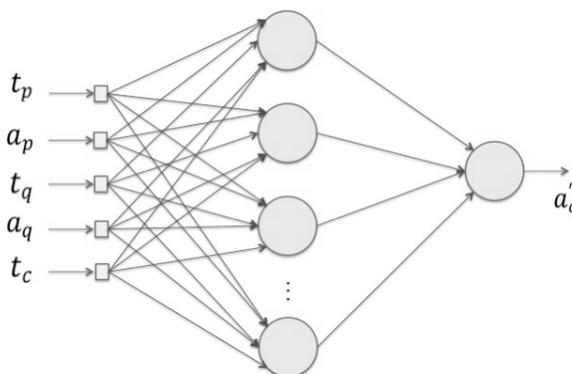
6.2.4 Compressão e descompressão do complemento de S por meio da rede neural artificial

Os pontos P e Q , pertencentes a S , são selecionados pelo AG para gerar um segmento de reta \overline{PQ} tão próximo quanto possível do sinal original (aproximação linear). No entanto, esta aproximação por segmentos de reta pode causar uma perda significativa de informação, dependendo das características não-lineares do sinal e do número de pontos do conjunto S , definido por V_p . Neste contexto, este trabalho propõe o uso das redes neurais artificiais para gerar uma aproximação não-linear entre os pontos P e Q . As redes neurais artificiais apresentam características de não-linearidade quando utilizam neurônios com função de ativação não-linear. A estratégia consiste em substituir o segmento de reta, que já está próximo do sinal original, por uma aproximação não-linear, ainda mais próxima do sinal original.

Para gerar uma aproximação não-linear do sinal original, é necessário determinar um ponto $C'(t'_c, a'_c)$, que é uma representação aproximada do ponto $C(t_c, a_c)$. A Figura 6.4 mostra que $t_c = t'_c$. Para determinar a'_c em sua saída, a rede neural toma como entrada as coordenadas dos pontos P e Q , bem como as abscissas do ponto C . Para ajustar os pesos sinápticos, o algoritmo de treinamento usa a diferença entre a'_c e a_c como sinal de erro. Assim, o conjunto dos dados de treinamento são todos os pontos do conjunto S (todos os pontos P e Q) e T (todos os pontos C). Como todos os pontos do conjunto universo U são conhecidos, não há a necessidade de conjuntos de teste e validação para preservar ou verificar a capacidade de generalização da rede neural.

Nesta metodologia, a estrutura da rede neural tem cinco entradas (nomeadas de t_p, a_p, t_q, a_q, t_c), uma camada intermediária de neurônios e uma camada de saída (para os valores de a'_c) como mostrada na Figura 6.6.

Figura 6.6 – Representação das entradas e saída da rede neural artificial para a metodologia de compressão.



A rede neural utilizada é uma *perceptron* multicamadas, com função de ativação sigmóide na camada intermediária, e com função de ativação linear na camada de saída; e o algoritmo de treinamento é o algoritmo Levenberg-Marquardt. Em geral, para redes que contém algumas centenas de pesos sinápticos, o algoritmo de Levenberg-Marquardt convergirá mais rapidamente. Esta vantagem é especialmente notável se for necessário um treinamento muito preciso (Agachi, et al., 2006).

Os valores de amplitude do sinal são convertidos em valores por unidade (p.u.), tomando como base o maior valor absoluto de amplitude do sinal.

O treinamento da rede neural finaliza o processo de compressão do complemento de S , durante o qual, as características do sinal, entre os sucessivos pontos do conjunto S , são armazenados nos pesos sinápticos da rede neural. Ao final do processo de compressão, o sinal comprimido é formado pelos pontos do conjunto S e pelos pesos sinápticos da rede neural. Em outras palavras, a informação que representa o sinal original é armazenada nos pesos sinápticos e nos pontos selecionados pelo AG.

O sinal descomprimido no conjunto U' (Figura 6.2 – f) é formado pela união (Figura 6.2 – e) dos conjuntos T' e S (Figura 6.2 – b e c). O conjunto T' consiste em todos os pontos $C'(t'_c, a'_c)$, onde t'_c é obtido pela equação 6.1 e a'_c é obtido na saída da rede neural, utilizando como entrada as coordenadas dos pontos P e Q , bem como t'_c .

6.3 RESULTADOS

Os sinais de tensão e corrente, utilizados para testar e validar a metodologia desenvolvida, foram obtidos a partir de um evento de curto-circuito registrado por um oscilógrafo instalado em um sistema real de energia elétrica de 230 kV/60 Hz. As Figuras 6.10 e 6.13 mostram os sinais de tensão e de corrente nas fases A, B e C registrados pelo oscilógrafo.

A arquitetura da rede neural, a taxa de aprendizagem e os parâmetros do AG foram definidos a partir da aplicação da metodologia em sinais de energia elétrica, após algumas simulações e análises dos resultados obtidos. A estrutura da rede neural e os seus parâmetros são mostrados na Tabela 2; e os parâmetros do AG são mostrados na Tabela 3.

Tabela 2 – Estrutura e parâmetros da rede neural artificial.

Estrutura e parâmetros	
Nós de fonte na camada de entrada	5
Neurônios na camada intermediária	15
Neurônios na camada de saída	1
Paradigma de aprendizagem	Aprendizado supervisionado
Algoritmo de aprendizagem	Levenberg-Marquardt
Função de ativação	Tangente hiperbólica (camada intermediária), e linear (camada de saída)
Épocas	400
Erro	0,000001
Taxa de aprendizagem	0,1

Tabela 3 – Parâmetros do AG.

Parâmetros	
População	100
Geração	1000
Técnica de seleção	Torneio
Técnica de cruzamento	100 - ou seja, um ponto por gene
Taxa de mutação	0,0001
Técnica de geração	Troca de geração

As Figuras 6.7, 6.8, 6.9, 6.11 e 6.14 mostram os resultados do AG, considerando os sinais de energia elétrica mostrados nas Figuras 6.10 e 6.13. As Figuras 6.12 e 6.15 apresentam os resultados da rede neural, considerando os resultados apresentados nas Figuras 6.11 e 6.14. Neste experimento, V_p é igual a 10%.

As Figuras 6.7, 6.8 e 6.9 mostra os valores de aptidão do melhor cromossomo para cada geração, considerando $V_p = 2,5\%$, 5% e 10% respectivamente. Observa-se que o valor de aptidão diminui ao longo das gerações. Como explicado na Seção 6.2.3, quanto menor o valor de aptidão, melhor a aptidão do cromossomo, a qual corresponde, a melhor representação linear do sinal.

Figura 6.7 – Curva de aptidão do melhor cromossomo para os sinais de tensão e corrente para as fases A, B e C, considerando $V_p = 2,5\%$.

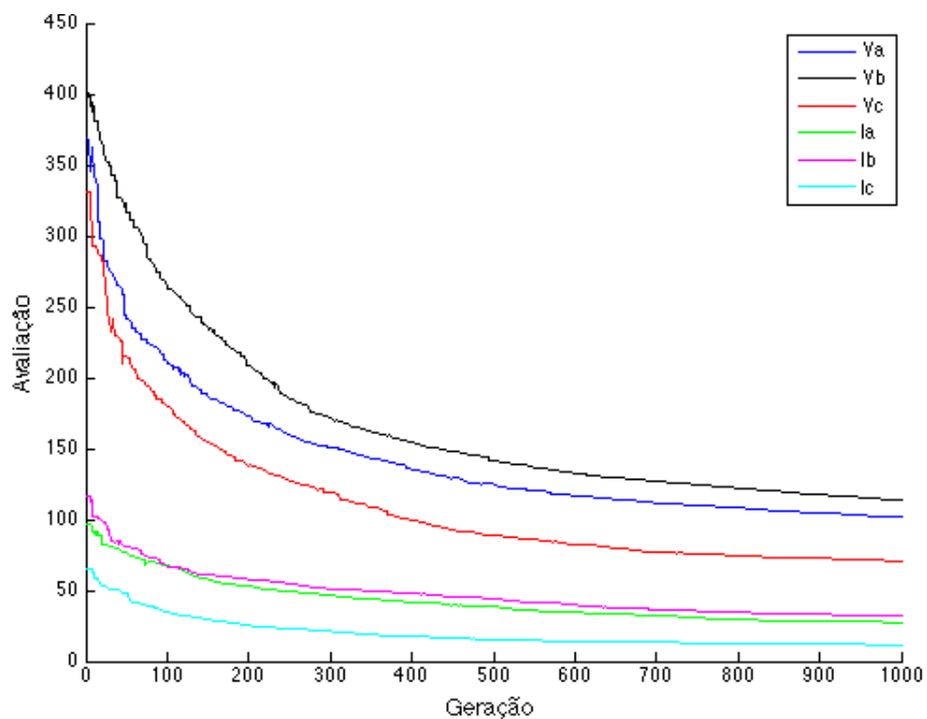


Figura 6.8 – Curva de aptidão do melhor cromossomo para os sinais de tensão e corrente para as fases A, B e C, considerando $V_p = 5\%$.

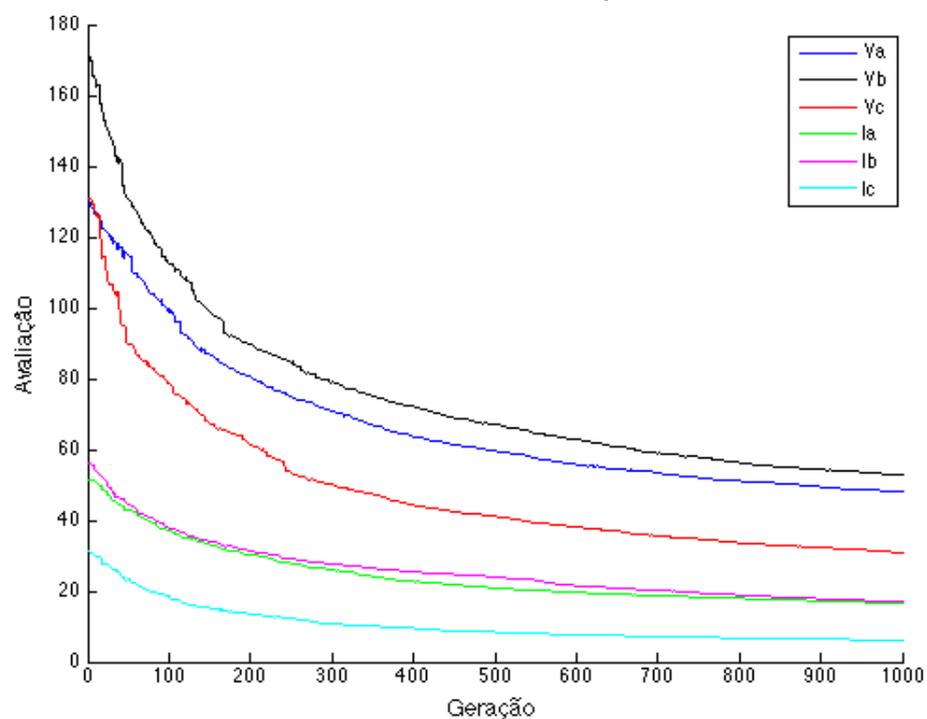
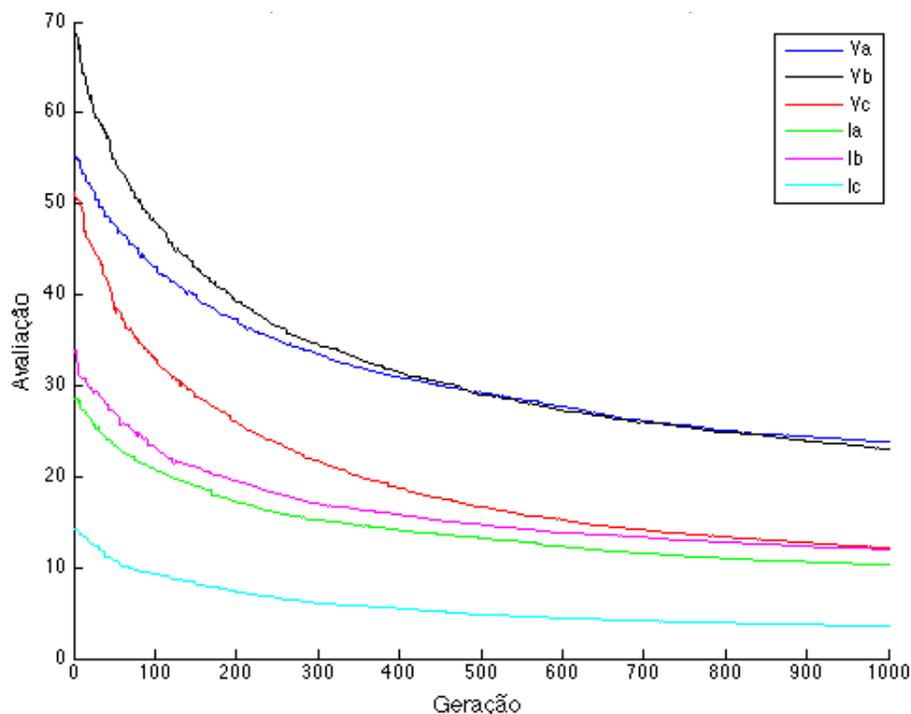


Figura 6.9 – Curva de aptidão do melhor cromossomo para os sinais de tensão e corrente para as fases A, B e C, considerando $V_p = 10\%$.



Com relação as Figuras 6.10 à 6.15, para uma melhor visualização gráfica dos resultados, apenas 5.000 do total de 19.456 amostras do sinal original foram representadas graficamente.

As Figuras 6.10 e 6.13 mostram, respectivamente, os sinais originais de tensão e corrente nas fases A, B e C, em p.u., amostrados com frequência de 15.360 Hz.

As Figuras 6.11 e 6.14 mostram os pontos do conjunto S determinados pelo AG, respectivamente para cada sinal de tensão e corrente. Nessas figuras, os pontos estão conectados por segmentos de reta.

Como mostrado nas Figuras 6.11 e 6.14, o AG seleciona os pontos que melhor caracterizam os contornos do sinal. As Figuras 6.12 e 6.15 mostram, respectivamente, os sinais de tensão e corrente reconstruídos para as fases A, B e C.

Cada sinal nas Figuras 6.12 e 6.15 (a, b e c) é a união entre os conjuntos S e T' , e como pode-se observar visualmente nesses resultados, as formas de onda reconstruídas representam satisfatoriamente os sinais de tensão e corrente originais respectivamente.

Os processos que mais demandam tempo correspondem ao algoritmo genético e a etapa de treinamento da rede neural, porque, em geral, são rotinas demoradas. Para cada forma de onda de tensão e corrente a ser comprimida é utilizado um algoritmo genético e uma rede neural artificial. O tempo de processamento do AG é da ordem de 0,2s por geração e o tempo de treinamento da rede neural está em torno de 0,05s por época,

considerando $V_p = 10\%$. No entanto, o processo de compressão pode ser realizado imediatamente após o evento de falha ter sido registrado e, portanto, essas grandezas de tempo de compressão não são valores críticos para análises em sistemas de energia elétrica, especialmente em análise de pós-operação. Por outro lado, quando o sinal original está comprimido e pronto para uso, o tempo de descompressão fica em torno de 0,0351s, o que é muito desejável para aplicações em tempo real. Esses valores foram medidos considerando um computador pessoal típico.

Figura 6.10 – Sinais de tensão originais para as fases A (a), B (b) e C (c) respectivamente.

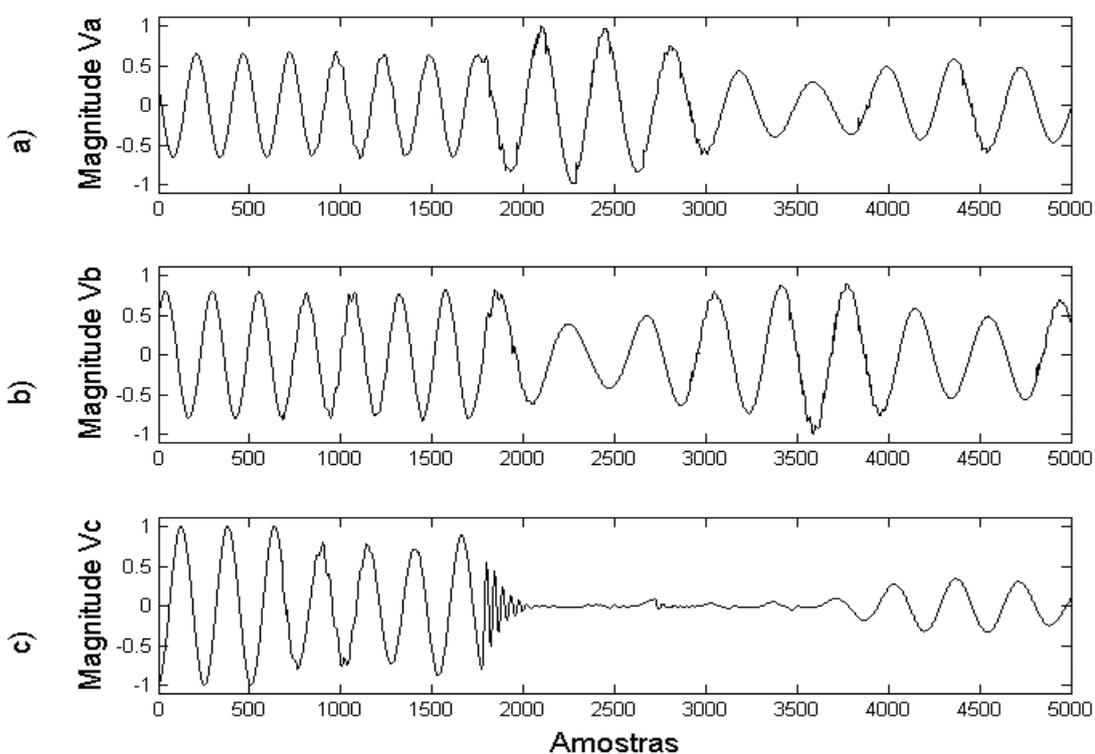


Figura 6.11 – Os pontos do conjunto S selecionados pelo AG para cada sinal de tensão nas fases A (a), B (b) e C (c) respectivamente.

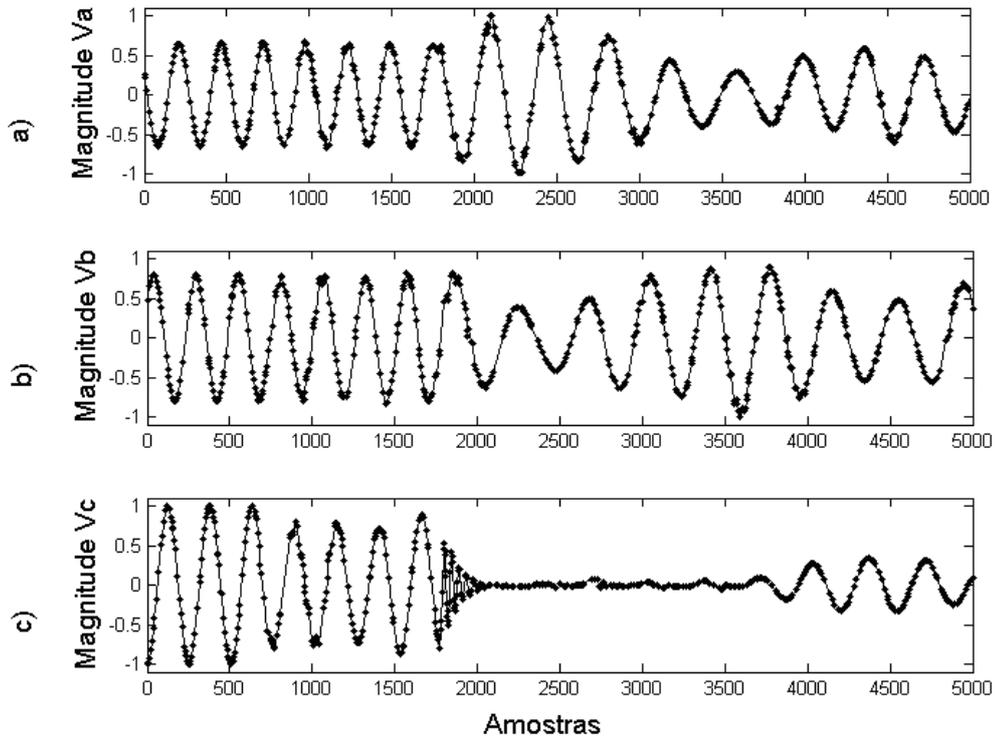


Figura 6.12 – Sinais de tensão reconstruídos para as fases A (a), B (b) e C (c) respectivamente.

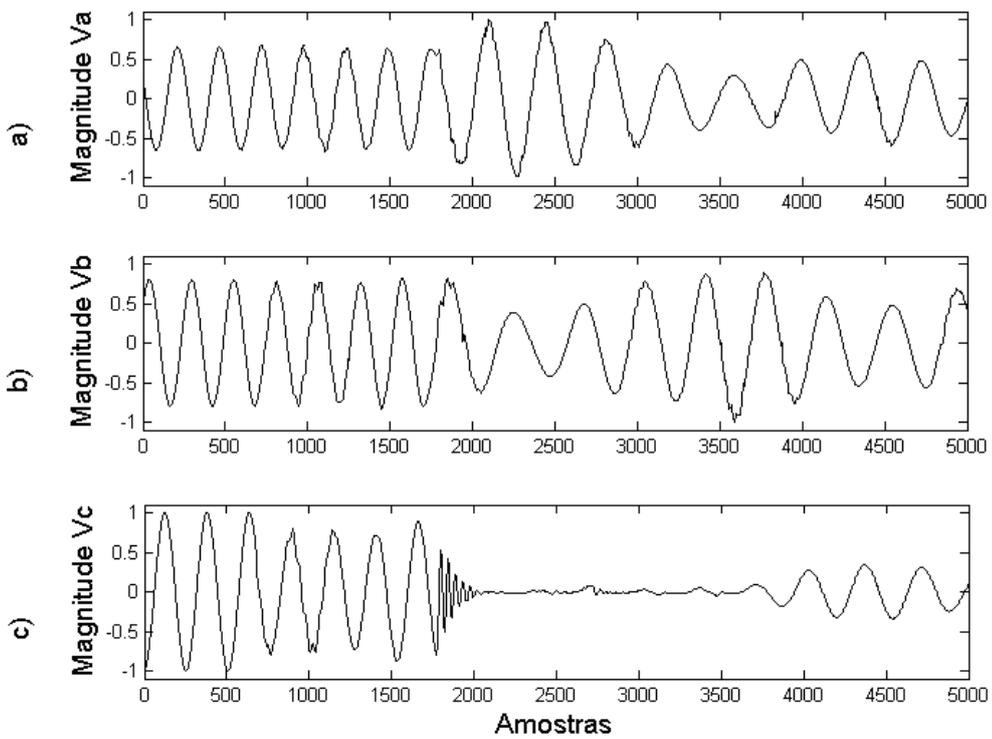


Figura 6.13 – Sinais de corrente originais para as fases (a), B (b) e C (c) respectivamente.

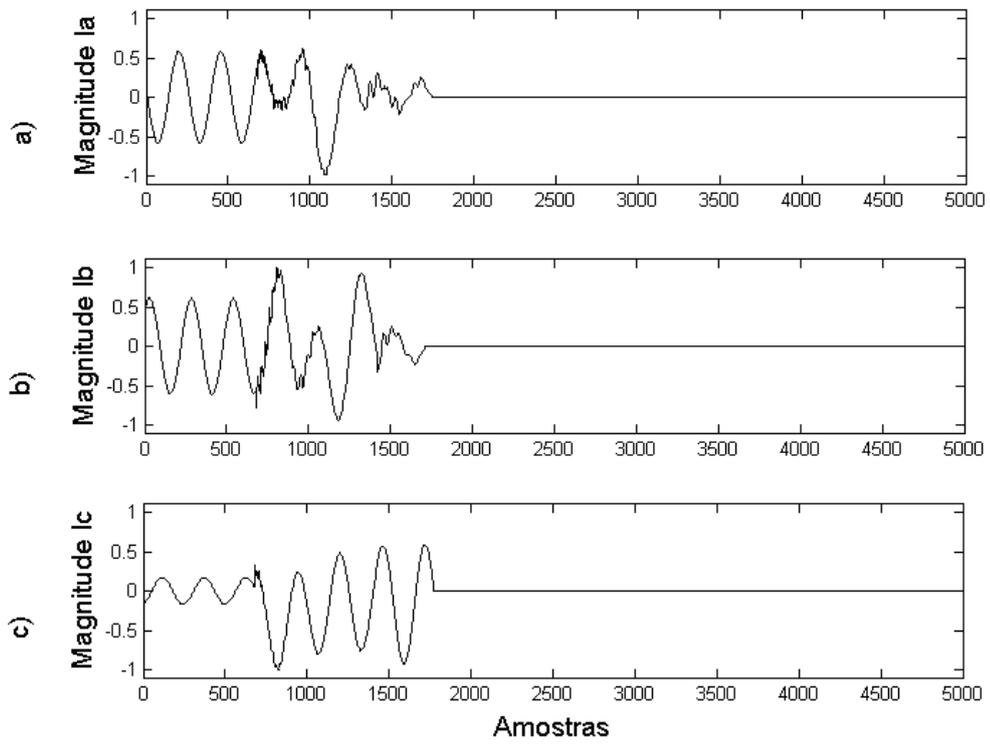


Figura 6.14 – Os pontos do conjunto S selecionados pelo AG para cada sinal de corrente nas fases A (a), B (b) e C (c) respectivamente.

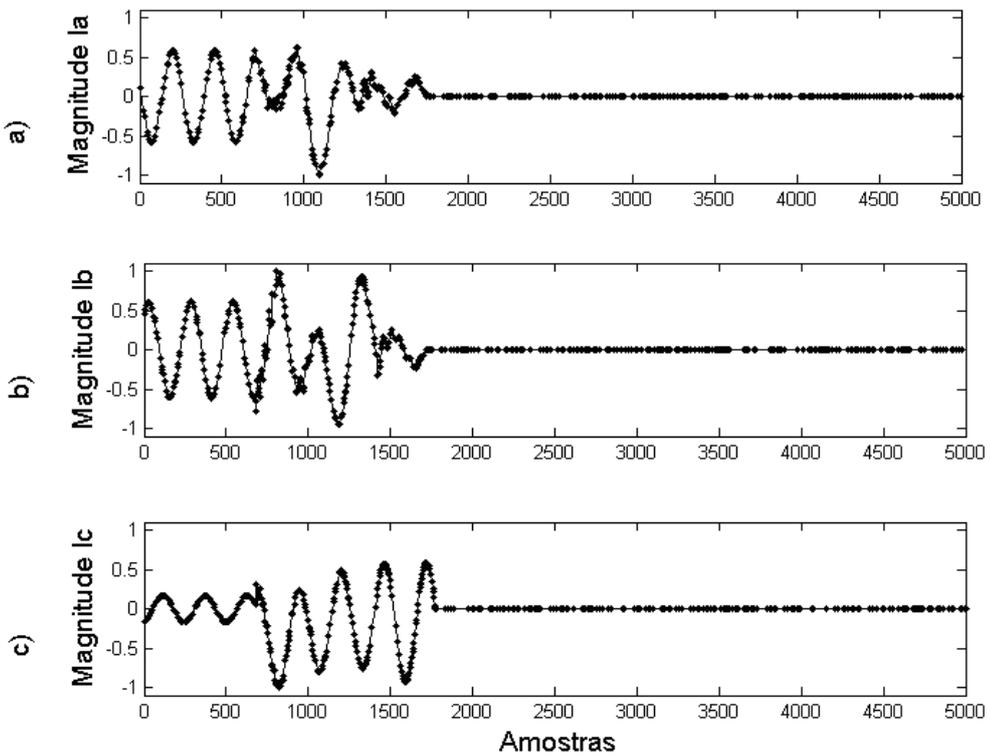
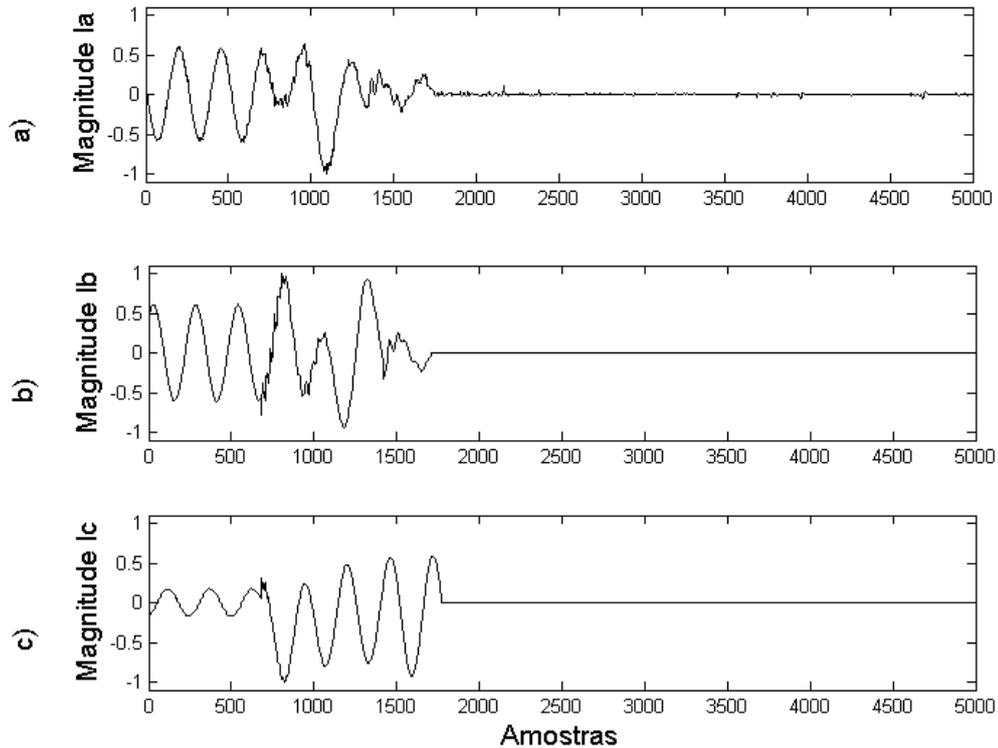


Figura 6.15 – Sinais de corrente reconstruídos para as fases A (a), B (b) e C (c) respectivamente.



Neste trabalho, a taxa de compressão (*compression rate* – CR(%)), a razão de compressão (*compression ratio* – CR), o erro médio quadrático (*mean square error* - MSE) e a relação sinal-ruído (*signal to noise ratio* - SNR) avaliam o desempenho da metodologia desenvolvida. Estes parâmetros são amplamente utilizados na literatura como critérios de avaliação para as metodologias de compressão (Zhang, et al., 2011) (Chang, et al., 2009) (Wang, et al., 2005). A razão de compressão é definida pela equação 6.11 (Santoso, et al., 1997):

$$CR = \frac{Sinal_{original}}{Sinal_{comprimido}} \quad 6.11$$

O erro médio quadrático é definido pela equação 6.12 (Lorio, et al., 2004). Como pode ser observado, quanto menor o valor de MSE, menor será a distância entre os pontos do sinal reconstruído e os pontos do sinal original.

$$MSE = \sum_{i=1}^a \frac{(Sinal_{original}(i) - Sinal_{reconstruido}(i))^2}{a} \quad 6.12$$

Sendo:

a : o número de amostras do sinal.

A relação sinal-ruído é definida pela equação 6.13 (Lorio, et al., 2004):

$$SNR(dB) = 10 \log_{10} \left(\frac{\sum_{i=1}^a |Sinal_{original}(i)|^2}{\sum_{i=1}^a |Sinal_{original}(i) - Sinal_{reconstruido}(i)|^2} \right) \quad 6.13$$

As Tabelas 4, 5 e 6 mostram a taxa de compressão, a razão de compressão, o erro médio quadrático e a relação sinal-ruído, considerando V_p igual a 2,5%, 5% e 10%, respectivamente.

Tabela 4 – Parâmetros de compressão, considerando $V_p = 2,5\%$.

V_p (%)	Sinal	CR (%)	CR	SNR (dB)	MSE
2,5	VA	95,86	24,16: 1,00	25,7648	$4,7 \cdot 10^{-4}$
	VB			25,5758	$6,65 \cdot 10^{-4}$
	VC			21,7693	$7,9 \cdot 10^{-4}$
	IA			25,3146	$1,08 \cdot 10^{-4}$
	IB			20,6267	$4,66 \cdot 10^{-4}$
	IC			32,2094	$2,7 \cdot 10^{-5}$

Tabela 5 – Parâmetros de compressão, considerando $V_p = 5\%$.

V_p (%)	Sinal	CR (%)	CR	SNR (dB)	MSE
5	VA	93,37	15,08: 1,00	29,5252	$2 \cdot 10^{-4}$
	VB			30,4215	$2,2 \cdot 10^{-4}$
	VC			31,7996	$7,9 \cdot 10^{-5}$
	IA			27,2816	$6,9 \cdot 10^{-5}$
	IB			29,2941	$6,3 \cdot 10^{-5}$
	IC			37,9206	$7,3 \cdot 10^{-6}$

Tabela 6 – Parâmetros de compressão, considerando $V_p = 10\%$.

V_p (%)	Sinal	CR (%)	CR	SNR (dB)	MSE
10	VA	88,36	8,59: 1,00	32,9065	$9,1 \cdot 10^{-5}$
	VB			34,9042	$7,8 \cdot 10^{-5}$
	VC			38,2935	$1,8 \cdot 10^{-5}$
	IA			31,7433	$2,5 \cdot 10^{-5}$
	IB			36,0167	$1,3 \cdot 10^{-5}$
	IC			39,3476	$5,3 \cdot 10^{-6}$

Como pode ser observado nas Tabelas 4, 5 e 6, quanto maior for o valor de V_p , menor será a taxa e a razão de compressão. Adicionalmente, quanto maior for o valor de V_p , melhores serão os valores da relação sinal-ruído e do erro médio quadrático.

Os resultados de taxa e razão de compressão, relação sinal-ruído e erro médio quadráticos nas Tabelas 4, 5 e 6 estão resumidos e destacados em negrito na primeira linha da Tabela 7.

Tabela 7 – Comparação de desempenho de taxa de compressão entre a metodologia proposta e outras metodologias pesquisadas na literatura.

Metodologias pesquisadas	Referências	SNR (dB)	MSE	CR (%)	CR
Metodologia proposta	-----	$(V_p = 10)$		88,36	8,59: 1,00
		32,9065 a	$5,3 \cdot 10^{-6} a$	93,37	15,08: 1,00
		39,3476	$9,1 \cdot 10^{-5}$	95,86	24,16: 1,00
Codificação por transformada	(Tcheou, et al., 2014)	31	---	93,75	16,00: 1,00
Métodos de Huffman e TW	(Hsieh, et al., 2003)	---	10^{-4}	70,85	3,43: 1,00
Transformada <i>wavelet</i> discreta	(Panda, et al., 2002)	---	$-19 (dB)$	90	10,00: 1,00
Transformada <i>wavelet</i> e RNA	(Meher, et al., 2004)	---	$-25 (dB)$	93,33	15,00: 1,00
Transformada discreta de cosseno	(Nascimento, 1997)	---	---	85,8 a	7,04: 1,00 a
				89,09	9,17: 1,00
<i>Embedded zero-tree wavelet</i>	(Chung, et al., 1999)	---	10^{-5}	90	10,00: 1,00
Transformada <i>wavelet</i>	(Gasparesc, 2010)	---	---	91,25	11,43: 1,00
<i>Lempel-Ziv-Welch</i>	(Lorio, et al., 2004)	---	10^{-5}	71,99 a	3,57: 1,00 a
				90,20	10,20: 1,00
Codificação aritmética adaptativa	(Huang, et al., 2004)	---	---	89,25	9,30: 1,00
	(Ribeiro, et al., 2004)	---	$-30 (dB)$		
	(Santoso, et al., 1997)	---	10^{-6}		
Transformada <i>wavelet packet</i>	(Littler, et al., 1999)	---	---	72,75 a	3,67: 1,00 a
				87,55	8,03: 1,00
	(Hamid, et al., 2002)	---	---		
	(Wu, et al., 2003)	---	---		

(Huang, et al., 2004)	---	10^{-5}
--------------------------	-----	-----------

Como pode ser observado na primeira linha da tabela anterior, o melhor resultado é 95,86% (24,16:1,00) e o pior resultado corresponde a 88,36% (8,59:1,00). Comparando estes resultados com os dos métodos pesquisados na literatura, e também apresentados na Tabela 7, verifica-se que algumas taxas de compressão obtidas utilizando a metodologia desenvolvida são maiores que a maioria das taxas obtidas pelos métodos pesquisados na literatura, e, portanto, melhores.

6.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou a metodologia desenvolvida para compressão e descompressão de sinais de energia elétrica a partir de registros de forma de onda. A estratégia desenvolvida para a utilização do algoritmo genético e da rede neural artificial mostrou-se eficaz como ferramenta de compressão para aplicações práticas em sistemas de energia elétrica, principalmente em análises de pós-operação.

A técnica de compressão de sinais de energia elétrica foi implementada utilizando o *software MatLab2011* em sua versão para o sistema operacional *Windows*. A leitura do arquivo no formato COMTRADE, com as formas de onda utilizadas neste trabalho, foi feita pela função `read_comtrade.m` existente na própria ferramenta *MatLab*. Os dados armazenados no arquivo em formato COMTRADE foram salvos no *MatLab workspace* (que permite a importação de variáveis, a partir de arquivos de dados ou outros programas, para o *MatLab*) e então carregados e utilizados pelo programa desenvolvido.

CAPÍTULO 7 – CONCLUSÃO

7.1 CONCLUSÕES

A metodologia proposta foi aplicada em seis sinais de energia elétrica reais com diferentes contornos, considerando diferentes valores para V_p (2,5%, 5% e 10%). Para estes sinais, os resultados gráficos, considerando V_p igual a 10%, mostram que a representação cromossomial, a função objetivo e os parâmetros da Tabela 3 dão ao algoritmo genético a habilidade de selecionar os pontos que melhor caracterizam os contornos dos sinais de energia elétrica. Os resultados gráficos também demonstram que a parametrização da rede neural e o algoritmo de treinamento Levenberg-Marquardt dão à rede neural artificial a capacidade de comprimir e descomprimir os sinais de energia elétrica. Como pode ser observado visualmente nas figuras, as formas de onda reconstruídas representam satisfatoriamente os sinais originais de tensão e corrente.

Os resultados obtidos nas tabelas mostram que o percentual de pontos selecionados pelo AG é relativamente pequeno e as taxas de compressão obtidas são relativamente altas e o que resulta em arquivos de dados com tamanhos relativamente pequenos. Estes resultados são comparados com os resultados obtidos em outros trabalhos publicados, demonstrando que algumas taxas de compressão obtidas, utilizando a metodologia proposta, são melhores do que as taxas de compressão obtidas pelas outras metodologias. Assim, a metodologia proposta apresenta um grande potencial para aplicações práticas na compressão de sinais de energia elétrica.

É importante destacar que a metodologia desenvolvida e os resultados alcançados foram enviados, em formato de artigo de pesquisa intitulado “*Compression of electrical power signals from waveform records using genetic algorithm and artificial neural network*”, para a revista *Electric Power System Research*, tendo sido publicado eletronicamente em novembro de 2016 e em revista impressa na edição de janeiro de 2017.

7.2 PROPOSTAS PARA TRABALHOS FUTUROS

A seguir, são citadas algumas propostas para trabalhos futuros provenientes da metodologia de compressão de sinais de energia elétrica desenvolvida:

1. Aplicação de uma técnica de compressão de dados (sem perdas) nos arquivos comprimidos resultantes da metodologia desenvolvida, com o objetivo de reduzir ainda mais o tamanho dos sinais comprimidos.

2. Aplicação da metodologia em sinais de natureza fisiológica (eletromiografia, eletrocardiografia e eletroencefalograma) para verificação dos resultados de taxas de compressão para sinais desta natureza.
3. Otimização da rede neural artificial, utilizada para compressão e descompressão dos sinais de energia elétrica, através dos ajustes de seus parâmetros utilizando os algoritmos genéticos.
4. Substituir, na metodologia desenvolvida, a rede neural MLP por redes neurais artificiais profundas.
5. Substituir, na metodologia desenvolvida, os algoritmos genéticos por outras técnicas de busca.

REFERÊNCIAS BIBLIOGRÁFICAS

- Ackley, D. H., Hinton, G. E. e Sejnowski, T. J. 1985.** A learning algorithm for boltzmann machines. *Cognitive Science*. 1985, Vol. 9, 1, pp. 147-169.
- Agachi, P. S., et al. 2006.** *Model Based Control: Case Studies in Process Engineering*. [ed.] Wiley-VCH. 2006. p. 41.
- Ahn, C. W. 2006.** *Advances in Evolutionary Algorithms: Theory, Design and Practice*. New York: Springer-Verlag New York, 2006.
- Aleksander, I. and Morton, H. 1990.** *An Introduction to Neural Computing*. London: Chapman and Hall, 1990.
- Angarita, J. A. C. 2013.** *Compressão de Dados para Sinais de Distúrbios no Ambiente das Redes Elétricas Inteligentes*. Brasília: Tese de Doutorado, 2013.
- Ashby, W. 1960.** *Design of a Brain: The Origin of Adaptative Behavior*. 2ª Ed. New York: Springer-Verlag, 1960.
- Bingham, R. P., Kreiss, D. e Santoso, S. 1995.** *Advances in Data Reduction Techniques for Power Quality Instrumentation*. 1995.
- Bledsoe, W. W. 1961.** *The use of biological concepts in the analytical study of systems*. Panoramic Research Report, 1961.
- Blickle, T. and Thiele, L. 1997.** *A Comparison of Selection Schemes used in Evolutionary Algorithms*. Zurich, Suíça, 1997.
- Box, G. E. P. 1957.** Evolutionary Operation: a method for increasing industrial productivity. *Applied Statistics*. 1957, Vol. 6, 2, pp. 81-101.
- Bremermann, H. J. 1962.** Optimization through evolution and recombination. 1962, pp. 96-106.
- Broomhead, D. S. e Lowe, D. 1988.** Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems*. 1988, Vol. 2, 3, pp. 321-355.
- Campani, C. A. P. 2005.** *Avaliação da Compressão de Dados e da Qualidade de Imagem em Modelos de Animação Gráfica para WEB: uma Nova Abordagem Baseada em Complexidade de Kolmogorv*. Porto Alegre: Tese de Doutorado, 2005.
- Cardoso, J. C. 2012.** ABARTE. Associação Brasileira das Grandes Empresas de Transmissão de Energia Elétrica. [Online] 2012. www.abrate.com.br/palavra.asp.

- Cardoso, R. 2011.** http://www.selinc.com.br/guia_aplic/Analise_de_Oscilografias_em_reles_digitais.pdf. *Selinc*. [Online] 2011.
- Chang, S. G., Yu, B. e Vitterli, M. 2009.** Adaptive Wavelet Thresholding for Image Denoising and Compression. *IEEE Transaction on Image Processing*. 2009, Vol. 9, pp. 1532-1546.
- Chung, J., Powers, E. J. e Grady, W. 1999.** Variable Rate Power Disturbance Signal Compression Using Embedded Zerotree Wavelet Transform Coding. *IEEE Power Engineering Society Winter Meeting*. 1999, Vol. 2, pp. 1305-1309.
- Cormane, J. A. A. 2013.** *Compressão de Dados para Sinais de Distúrbios no Ambiente das Redes Elétricas Inteligentes*. Brasília: Universidade de Brasília - Tese de Doutorado, 2013.
- Dash, P. K., Panigrahi, B. K. e Sahoo, D. K. 2003.** Power Quality Disturbance Data Compression. *IEEE Transaction on Power Delivery*. 2003, Vol. 18, pp. 595-600.
- Davis, L. 1991.** *Handbook of Genetic Algorithms*. Van Reinhold Nostrand, 1991.
- Engelbrecht, A. P. 2007.** *Computational intelligence - an introduction*. Andries P. Engelbrecht, 2007.
- Ferreira, W. M. 2002.** *Modelagem e Simulação do Processo de Extração Seletiva de Cobalto e Níquel por Solução Orgânica Através da Aplicação de Técnicas de Redes Neurais*. Trabalho Acadêmico de Conclusão de Curso, 2002.
- Fischler, M. A. and Firschein, O. 1987.** *Intelligence: The Eye, The Brain, and The Computer*. Addison-Wesley, 1987.
- Fonseca, W. A. dos S. 2012.** *Estimação da Seção em Falta e Detecção de Falhas na Proteção de Sistemas Elétricos a partir da Normalização dos Registros de Alarmes dos Dispositivos de Proteção*. Belém: Tese de Doutorado, 2012.
- Gasparesc, G. 2010.** Data Compression of Power Quality Disturbances using Wavelet Transform and Spline Interpolation Method. *International Conference on Environment and Electrical Engineering*. 2010.
- Gasparesc, 2013.** *Power Quality Data Compression*. Ahmed Zobaa, 2013.
- Gerek, O. N. e Ece, D. 2004.** 2-D Analysis and Compression of Power Quality Event Data. *IEEE Transaction on Power Delivery*. 2004, Vol. 19, pp. 791-798.
- Gerek, O. N. e Ece, D. G. 2008.** Compression of Power Quality Event Data Using 2D Representation. *Electric Power Systems Research*. 2008, Vol. 78, pp. 1047-1052.

- Goldberg, D. E. e Holland, J. H. 1988.** Genetic Algorithms and Machine Learning. *Machine Learning*. 1988, Vol. 3, 2, pp. 95-99.
- Grigsby, L. L. 2012.** *The Electric Power Engineering Handbook*. IEEE Press: CRC Press, 2012.
- Guidorizzi, H. L. 2008.** *Um Curso de Cálculo*. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora S.A., 2008. pp. 26-37. Vol. 1.
- Hagan, M. T. e Menhaj, M. B. 1994.** Training feedforward networks with the Marquardt algorithm. *IEEE Transaction Neural Networks*. 1994, Vol. 5, 6, pp. 989-993.
- Hamid, E. Y. and Kawasaki, Z. I. 2002.** Wavelet-based Compression of Power Disturbances using the Minimum Description Length Criterion. *IEEE Transactions on Power Delivery*. 2002, Vol. 17.
- Haykin, S. e Veen, B. V. 2002.** *Sinais e Sistemas*. Porto Alegre: Bookman (Reimpressão), 2002.
- Haykin, S. 2001.** *Redes Neurais: Princípios e Prática*. 2ª Edição. Porto Alegre: Bookman, 2001.
- Hebb, D. O. 2002.** *The Organization of Behavior*. Taylor Print on Dema, 2002.
- Holland, J. 1975.** *Adaptation in Natural and Artificial Systems*. 1975.
- Hsieh, C. -T. e Huang, S. -J. 2003.** Disturbance Data Compression of a Power System Using the Huffman Coding Approach with Wavelet Transform Enhancement. 2003, Vol. 150, pp. 7-14.
- Hsieh, C. -T., Huang, S. -J. e Huang, C. -L. 1998.** Data Reduction of Power Quality Disturbances: A Wavelet Transform Approach. *Electric Power Systems Research*. 1998, Vol. 47, pp. 79-86.
- Huang, S. -J. e Jou, M. -J. 2004.** Application of Arithmetic Coding for Electric Power Disturbance Data Compression with Wavelet Packet Enhancement. *IEEE Transaction on Power System*. 2004, Vol. 19, pp. 1334-1341.
- IEEE Std C37.111.1991.** IEEE Standard Common Format for Transient Data Exchange (CONTRADE) for Power Systems. ISBN 1-55937-156-0.
- IEEE Std C37.111.1999.** IEEE Standard Common Format for Transient Data Exchange (CONTRADE) for Power Systems. ISBN 0-7381-1667-X SS94743.
- Iezzi, G. 2005.** *Fundamentos de Matemática Elementar*. São Paulo: Editora Saraiva, 2005. pp. 6-7, 28-29, 90-93. Vol. 7.

- Kerlirzin, P. and Vallet, F. 1993.** *Neural Computation: Robustness in Multilayer Perceptrons*. 1993. pp. 473-482. Vol. 5.
- Kirkpatrick, S., Gelatt, D. C. Jr. e Vecchi, M. P. 1983.** Optimization by simulated annealing. *Science*. 1983, Vol. 13, pp. 671-680.
- Kohonen, T. 1982.** Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*. 1982, Vol. 43, 1, pp. 59-69.
- Koza, J., Keane, M. A. and Streeter, M. J. 2003.** *Genetic programming IV: Routine Human-Competitive Machine Intelligence*. Hingham: Kluwer Academic Publishers, 2003.
- Lelewer, D. e Hirschberg, D. 1987.** Data Compression. 1987, Vol. 19, pp. 261-296.
- Li, F., et al. 2010.** Smart Transmission Grid: Vision and Framework. Setembro de 2010, Vol. 1, 2, pp. 168-177.
- Linden, R. 2012.** *Algoritmos Genéticos*. Rio de Janeiro : Ciência Moderna, 2012.
- Linsker, R. 1988.** Self-Organization in a Perceptual Network. *Journal Computer*. 1988, Vol. 21, 3, pp. 105-117.
- Lira, M. M. da S. 2004.** *Wavelets na Compactação e Processamento de Sinais de Distúrbio em Sistemas de Potência para Classificação Via Redes Neurais Artificiais*. Pernambuco: Tese de Doutorado, 2004.
- Little, T. B. e Morrow, D. J. 1999.** Wavelets for the Analysis and Compression of Power Systems Disturbances. *IEEE Transaction on Power Delivery*. 1999, Vol. 14, pp. 358-364.
- Lorio, F. and Magnago, F. 2004.** Analysis of Data Compression Methods for Power Quality Events. *IEEE Proceedings of the Power Engineering Society General Meeting*. 2004, Vol. 1.
- Lovisol, L., Silva, E. A. B. e Rodrigues, M. A. M. 2005.** Efficient Coherent Representations of Power Systems Signals Using Damped Sinusoids. *IEEE Transaction on Signal Processing*. 2005, Vol. 53, pp. 3831-3846.
- Lovisol, Lisandro, et al. 2003.** Eliminação de Resposta Transitória em Oscilogramas Utilizando Matching Pursuits. *VII Seminário Técnico de Proteção e Controle*. 2003.
- Luo, X., Liao, L. and Tam, H. W. 2007.** Convergence Analysis of the Levenberg-Marquardt Method. *Optimization Methods and Software*. 2007, Vol. XXII, 4.
- Mallat, S. e Zhang, Z. 1993.** Matching Pursuits with Time-frequency Dictionaries. *IEEE Transaction on Signal Processing*. 1993, Vol. 4, pp. 3397-3415.
- McCulloch, W. e Pitts, W. 1943.** A logical calculus of the ideas immanent in nervous activity. 1943, Vol. 5, pp. 115-133.

- Meher, S. K., Pradhan, A. K. e Panda, G. 2004.** An Integrated Data Compression Scheme for Power Quality Events Using Spline Wavelet and Neural Network. *Electric Power Systems Research*. 2004, Vol. 69, pp. 213-220.
- Mello, N. F. B. 2006.** *Automação Digital de Subestações de Energia Elétrica*. Rio de Janeiro : Trabalho de Conclusão de Curso, 2006.
- Mendel, J. M. and McLaren, R. W. 1970.** *Adaptive, Learning, and Pattern Recognition Systems: Theory and Applications*. New York: Academic Press, 1970. pp. 287-318. Vol. LXVI.
- Mennon, A., et al. 1996.** *Neural Networks: Characterization of a Class of Sigmoid Functions with Applications to Neural Networks*. 1996. pp. 819-835. Vol. IX.
- Minsky, M. e Papert, S. 1969.** *Perceptrons. An introduction to computational geometry*. M.I.T. Press, 1969.
- Minsky, M. L. 1954.** *Theory of Neural-analog Reinforcement Systems and Its Application to the Brain Model Problem*. Princeton University, 1954.
- Mitchell, M. 1998.** *An introduction to genetic algorithms*. MIT press, 1998.
- Moreto, M. e Rolim, J. 2010.** Análise Automática de Oscilografias em Sistemas Elétricos de Potência. Junho de 2010, Vol. 21, 4, pp. 347-362.
- Nascimento, F. A. O. 1997.** Data Compression Algorithm for Transient Recording System. 1997, Vol. 3, pp. 1126-1130.
- Nascimento, S., et al. 2007.** Técnica de Compressão de Sinal para Oscilografia Digital de Sistemas de Potência. *VII Conferência Brasileira sobre Qualidade da Energia Elétrica*. 2007.
- Ning, J., Wang, J. e Gao, W. 2011.** A Wavelet-based Data Compression Technique for Smart Grid. *IEEE Transaction on Smart Grid*. 2011, Vol. 2, pp. 212-218.
- Nocedal, J. and Wright, S. 1999.** *Numerical Optimization*. New York: Springer, 1999. 0-387-98793-2.
- Nunez, V. B., Frigola, J. M. and Jaramillo, S. H. 2008.** A Survey on Voltage Sag Events in Power Systems. *Proceedings of the International Conference on Renewable Energies and Power Quality*. 2008, Vol. 1, 6.
- Panda, G., et al. 2002.** Data Compression of Power Quality Events using the Slantlet Transform. *IEEE Transactions on Power Delivery*. 2002, Vol. 17, 2.
- Qing, A., Hongtao, Z. e Zhikun, H. 2011.** *A Compression Approach of Power Quality Monitoring Data Based on two-dimension DCT*. 2011. pp. 20-24. Vol. 1.

- Ramos, F. R., Ribeiro, M. V. e Romano, J. M. T. 2002.** On Signal Processing Approach for Even Detection and Compression Applied to Power Quality Evaluation. 2002, Vol. 1, pp. 133-138.
- Reason. 2006.** <http://www.reason.com.br>. *REASON Tecnologia*. [Online] 2006.
- Rechenberg, I. 1973.** *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog Verlag, 1973.
- Ribeiro, M. V. e Duque, C. A. 2002.** The Word Length Influence on Waveform Coding Techniques Based on Wavelet Transform Applied to Disturbance Compression. 2002, Vol. 1, pp. 139-143.
- Ribeiro, M. V. e Park, S. H. 2007.** A Novel MDL-based Compression Method for Power Quality Applications. *IEEE Transaction on Power Delivery*. 2007, Vol. 22, pp. 27-36.
- Ribeiro, M. V., Romano, J. M. T. and Duque, C. A. 2004.** An Improved Method for Signal Processing and Compression in Power Quality Evaluation. *IEEE Transactions on Power Delivery*. 2004, Vol. 19, 2.
- Ribeiro, M. V., Romano, J. M. T. e Duque, C. 2001.** An Enhanced Data Compression Method for Applications in Power Quality Analysis. 2001, Vol. 1, pp. 676-681.
- Rochester, N., et al. 1956.** Tests on a cell assembly theory of the action of the brain, using a large digital computer. *IRE Transactions on Information Theory*. IEEE, 1956, Vol. 2, 3, pp. 80-93.
- Rosenblatt, F. 1958.** The Perceptron: A probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*. 1958, Vol. 65, 6, pp. 386-408.
- Rumelhart, D. E., Hinton, G. E. e Williams, R. J. 1986.** Learning Representations by Back-propagating Errors. *Nature*. 1986, Vol. 323, pp. 533-536.
- Russel, S. J. and Norving, P. 2004.** *Inteligência Artificial*. Rio de Janeiro: Elsevier, 2004.
- Sage, A. P. 1990.** *Concise Encyclopedia of Information Processing in Systems and Organizations*. New York: Pergamon, 1990.
- Salomon, D. 2000.** *Data Compression: The complete reference*. New York: Springer, 2000.
- Santoso, S., Powers, E. J. and Grady, W. M. 1997.** Power Quality Disturbance Data Compression Using Wavelet Transform Methods. *IEEE Transactions on Power Delivery*. 1997, Vol. 12, 3, pp. 1250-1257.
- Sayood, K. 2006.** *Introduction to Data Compression*. Morgan Kaufman, 2006.
- Schmidt, D. P. 2007.** *Ferramenta de Compressão de Arquivos Open Source*. 2007.

- Shang, L. e Krebs, J. J. R. 2003.** Efficiency Analysis of Data Compression of Power Systems Transient Using Wavelet Transform. 2003, Vol. 4.
- Shepherd, G. M. and Koch, C. 1990.** *The Synaptic Organization of the Brain: Introduction to Synaptic Circuits*. New York: G. M. Shepherd, 1990. pp. 3-31.
- Silva, K. M, et al. 2007.** Detecção e Classificação de Faltas a partir de Análise de Registros Oscilográficos via Redes Neurais Artificiais e Transformada Wavelet. *Sba Controle e Automação*. 2007, Vol. 18, 2.
- Tcheou, M. P. 2005.** Análise e Representação de Sinais de Oscilografia usando Decomposições Adaptativas Redundantes. *COPPE/UFRJ*. Rio de Janeiro, Brasil : Dissertação de Mestrado, 2005.
- Tcheou, M. P., et al. 2006.** Avaliação de um Método de Compressão para Oscilografia com Base em Procedimentos de Análise de Perturbações Elétricas. *X Simpósio de Especialistas em Planejamento da Operação e Expansão Elétrica*. 2006, 10.
- Tcheou, M. P., et al. 2012.** How far Can One Compress Digital Fault Records? Analysis of a Matching Pursuit-based Algorithm. *Digital Signal Processing*. 2012, Vol. 22, pp. 228-297.
- Tcheou, M. P., et al. 2014.** The Compression of Electric Signal Waveforms for Smart Grids: State of the Art and Future Trends. *IEEE Transaction on Smart Grid*. 2014, pp. 291-302.
- Uttley, A. M. 1959.** The design of conditional probability computers. *Information and Control*. Elsevier, 1959, Vol. 2, 1, pp. 1-24.
- Vapnik, V. e Cortes, C. 1995.** Support-vector networks. *Machine Learning*. 1995, Vol. 20, 3, pp. 273-297.
- Vose, M. D. 2004.** *The Simple Genetic Algorithm*. Nova Dheli: Prentice-Hall, 2004.
- Wang, J. and Wang, C. 2005.** Compression of Power Quality Disturbance Data Based on Energy Threshold and Adaptive Arithmetic Encoding. *IEEE Proceedings of the TENCON*. 2005.
- Wiegand, T. e Schwarz, H. 2011.** Source Coding: Part I of Fundamentals of Source and Video Coding. *Foundations and Trends in Sample*. 2011.
- Winandy, C., Borges F., E. and Bento, L. V. 2007.** Algoritmos para Aprendizagem Supervisionada. *Inteligência Artificial*. 2007.
- Wolfram, S. 2002.** *A New Kind of Science*. Wolfram Media, 2002. 1-57955-008-8.

Wu, C. -J., Fu, T. -H. e Huang, C. -P. 2003. Data Compression Technique in Recording Electric Arc Furnace Voltage and Current Waveforms for Tracking Power Quality. 2003, Vol. 1, pp. 383-388.

Yuan, Y., Yu, X. e Du, H. 2006. Power System Fault Data Compression Using the Wavelet Transform and Vector Quantification. 2006, pp. 1-6.

Zhang, D., Bi, Y. e Zhao, J. 2009. A New Data Compression Algorithm for Power Quality Online Monitoring. 2009, pp. 1-4.

Zhang, M., Li, K. and Hu, Y. 2011. A High Efficient Compression Method for Power Quality Applications. *IEEE Transactions on Instrumentation and Measurement*. 2011, Vol. 60, 6.

ANEXO 1

```

station_name, rec_dev_id
TT, ##A, ##D
An,ch_id, ph,ccbm, uu, a, b, skew, min, max
.....
Dn,ch_id, ph, ccbm, y
.....
lf
nrates
samp1, endsamp1
samp2, endsamp2
.....
sampn, endsampn
dd/mm/yyyy, hh:mm:ss.ssssss
dd/mm/yyyy, hh:mm:ss.ssssss
ft

```

Sendo:

station_name: o nome exclusivo do registrador;

rec_dev_id: o número exclusivo do registrador;

TT: o número total de canais;

##A: o número de canais analógicos seguido do identificador A;

##D: o número de canais digitais seguido do identificador D;

An: o número indicador do canal analógico;

Dn: o número indicador do canal digital;

ch_id: o identificador (nome) do canal;

ph: a identificação da fase do canal (ex.: A, B, C, N, entre outras);

ccbm: a identificação do componente/circuito sendo monitorado;

uu: a unidade do canal (ex.: kV, kA, entre outras);

a: veja *b*;

b: o fator de conversão do canal é $ax + b$ (ex.: um valor registrado de *x* corresponde a $ax + b$ na unidade *uu* especificada acima);

skew: provê informação sobre a diferença de tempo entre amostragem dos canais dentro do período de amostragem de um registro (em μ s);

min: o inteiro igual ao valor mínimo (limite inferior da faixa de amostragem) para amostras deste canal;

max: o inteiro igual ao valor máximo (limite superior da faixa de amostragem) para amostras deste canal;

y: (0 ou 1) o estado normal de operação do canal (aplicado somente a canais digitais);

lf: a frequência da linha em Hz;

nrates: o número total de taxas de amostragem seguida de uma lista com cada taxa de amostragem e o número da última amostra da taxa dada;

sampn: a enésima taxa de amostragem;

ft: identifica o tipo de arquivo de dados: ASCII;

dd/mm/yyyy: dia/mês/ano; e

hh:mm:ss.ssss: hora/minutos/segundos.

O primeiro campo data/hora é para registrar a data e a hora do início do registro de dados e o segundo é para registrar a data e a hora do ponto de disparo da oscilografia.

ANEXO 2

```

station_name, rec_dev_id, rev_year
TT, ##A, ##D
An, ch_id, ph, ccbm, uu, a, b, skew, min, max, primary, secondary, OS
.....
Dn, ch_id, ph, ccbm, y
.....
lf
nrates
samp1, endsamp1
samp2, endsamp2
.....
sampn, endsampn
dd/mm/yyyy, hh:mm:ss.ssssss
dd/mm/yyyy, hh:mm:ss.ssssss
ft
timemult

```

Sendo:

rev_year: o ano da revisão do padrão COMTRADE (ex. 1999) aplicado na formatação do arquivo;

primary: o fator primário da razão de transformação de corrente ou tensão;

secondary: o fator secundário da razão de transformação de corrente ou tensão;

PS: este campo especifica se o valor recebido a partir da equação de fator de conversão do canal $ax + b$ representa um valor primário ou secundário;

lf: a frequência da linha. O valor do campo é um ponto flutuante; e

ft: identifica o tipo de arquivo de dados: ASCII ou binário.