

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

EVANDRO MONTEIRO JORGE JÚNIOR

Análise de eventos de fusão em traços de OTDR utilizando redes neurais

Uberlândia

2018

EVANDRO MONTEIRO JORGE JÚNIOR

Análise de eventos de fusão em traços de OTDR utilizando redes neurais

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciências

Área de concentração: Processamento digital de Sinais.

Orientador: Prof. Dr. Antônio Cláudio Paschoarelli Veiga

Uberlândia
2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

J82a Jorge Júnior, Evandro Monteiro, 1989-
2018 Análise de eventos de fusão em traços de OTDR utilizando redes
neurais [recurso eletrônico] / Evandro Monteiro Jorge Júnior. - 2018.

Orientador: Antônio Cláudio Paschoarelli Veiga.
Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Elétrica.
Modo de acesso: Internet.
Disponível em: <http://dx.doi.org/10.14393/ufu.di.2018.1189>
Inclui bibliografia.
Inclui ilustrações.

1. Engenharia elétrica. 2. Back propagation (Inteligência artificial).
3. Fibras óticas. 4. Redes neurais (Computação). I. Veiga, Antônio
Cláudio Paschoarelli, 1963- (Orient.) II. Universidade Federal de
Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III.
Título.

CDU: 621.3

Maria Salete de Freitas Pinheiro - CRB6/1262

Dissertação de Evandro Monteiro Jorge Júnior, sob o título "**Análise de eventos de fusão em traços de OTDR utilizando redes neurais**", apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, na linha de pesquisa: Processamento Digital de Sinais, como parte dos requisitos para a obtenção do título de Mestre em Ciências aprovada em 02 de agosto de 2018 pela comissão julgadora constituída pelos doutores:

Prof. Dr. Antônio Claudio Paschoarelli Veiga
Instituição: Universidade Federal de Uberlândia

Prof. Dr. Daniel Moraes Santos
Instituição: Universidade Federal dos Vales do Jequitinhonha e Mucuri

Prof. Dr. Lorenço Santos Vasconcelos
Instituição: Universidade Federal de Uberlândia

Agradecimentos

Primeiramente eu agradeço a Deus, por me iluminar e permitir que eu desenvolva esta pesquisa.

Ao Professor Antônio Claudio Paschoarelli Veiga por me orientar durante o desenvolvimento do mestrado

Ao Professor Alexander Bento Melo, que me ajudou no desenvolvimento da minha pesquisa.

Ao corpo docente da Faculdade de Engenharia Elétrica em especial para os professores, Keiji Yamanaka, Gilberto Carrijo, Edna Flores, além do orientador, devido ao fato de ter feito as disciplinas com eles, o que me ajudou neste trabalho.

Aos funcionários da Faculdade de Engenharia Elétrica pois sem eles não seria possível o funcionamento da unidade acadêmica.

Aos meu pai Evandro, minha mãe Marta, e minha irmã Verônica que sempre me apoiaram.

Ao meu amigo Tiago Nunes, que me apoiou no ingresso à pós-graduação.

Ao CAPES que me proveu a bolsa de estudos durante todo o período do mestrado.

E a todos os amigos e pessoas que me ajudaram ou apoiaram no desenvolvimento deste trabalho.

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

(Albert Einstein)

Resumo

JORGE JÚNIOR, Evandro Monteiro. **Análise de eventos de fusão em traços de OTDR utilizando redes neurais.** 2018. 91 p. Dissertação (Mestrado em Ciências) – Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia, 2018.

Quando um enlace óptico é instalado ou quando irá fazer uma atualização de tecnologia em um enlace já existente é necessário fazer testes para saber como é o estado desse enlace, e se nele é possível trafegar os dados demandados. O equipamento mais importante nesses testes é o OTDR, porém os *softwares* contidos nos equipamentos utilizam algoritmos bem simples que nem sempre conseguem detectar eventos que ocorrem nessa fibra. Esta dissertação tem o propósito de apresentar um algoritmo baseado em Redes Neurais Artificiais, através do método de treinamento *backpropagation* com intuito de criar uma nova forma de identificar eventos de fusão em traços de OTDR. Foi desenvolvido um *software* em Matlab, utilizando o algoritmo de *backpropagation* para realizar a detecção dos eventos de fusão em um traço de OTDR. Em seguida foi utilizado o *software* para detecção de fibras em casos reais. Por fim foi feita uma análise sobre os casos em que foi utilizada a Rede Neural, e chegou-se a conclusão de que é possível utilizar a técnica de Redes Neurais para resolver problemas deste tipo, além disso foram deixadas algumas propostas para trabalhos futuros envolvendo o tema.

Palavras-chave: Backpropagation, Caracterização, Fibra Óptica, Redes Neurais Artificiais, OTDR.

Abstract

JORGE JÚNIOR, Evandro Monteiro. **Análise de eventos de fusão em traços de OTDR utilizando redes neurais.** 2018. 91 p. Dissertation (Master of Science) – Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia, 2018.

When an optical link is installed or when it will perform a technology upgrade on an existing link, it is necessary to perform tests to determine the state of that link, and if it is possible to traffic the requested data. The most important equipment in these tests is the OTDR, but the software contained in the equipment uses very simple algorithms that can not always detect events that occur in this fiber. This dissertation aims to present an algorithm based on Artificial Neural Networks, through the backpropagation training method for a new way of identifying fusion events in OTDR traces. We developed software in Matlab, using the backpropagation algorithm to perform the detection of fusion events in an OTDR trace. Then the fiber detection software was used in real cases. Finally, an analysis was made on the cases in which the Neural Network was used, and it was concluded that it is possible to use the Neural Networks technique to solve problems of this type, in addition some proposals were left for future work involving the theme.

Keywords: Artificial Neural Networks, Backpropagation, Characterization, Optical Fibers, OTDR.

Lista de figuras

Figura 1 - Lápis no copo.....	16
Figura 2 - A luz que faz curva na água.....	17
Figura 3 - As regiões no espectro de frequências utilizados nas comunicações via rádio e fibra óptica	19
Figura 4 - Analise por Wavelet de eventos de fusão	26
Figura 5 - Mark I Perceptron	30
Figura 6 - Esquema básico do neurônio biológico	33
Figura 7 - Modelo de Neurônio não linear.	34
Figura 8 - Transformação afim causada pelo bias.....	35
Figura 9 - Rede <i>feedforward</i> com uma única camada.....	36
Figura 10 - Rede feedforward totalmente conectada com uma camada escondida e uma camada de saída.	37
Figura 11 - RNA multicamadas com uma camada escondida.....	39
Figura 12 - Função Sigmoide	42
Figura 13 - Função Sigmoide Bipolar	42
Figura 14 – Importância de não treinar a rede até que ela chegue a um mínimo.....	45
Figura 15 - OTDR Modelo Yokogawa AQ7270	48
Figura 16 - Espalhamento de Rayleigh.....	49
Figura 17 - Reflexão de Fresnel.	50
Figura 18 - Diagrama de blocos do OTDR.....	51
Figura 19 - Curva típica de um OTDR	54
Figura 20 - Janelas de transmissão em fibra óptica	56
Figura 21 - Configurações no OTDR.	57
Figura 22 - Display mostrando o resultado de um teste.	58
Figura 23 - Evento de fusão.....	58
Figura 24 - Evento reflexão no fim da fibra.....	59
Figura 25 - Tela do <i>software</i> Fiberizer	62
Figura 26 - Traço de OTDR	63
Figura 27 - Sinal original e o sinal filtrado com filtro de média	63
Figura 28 - Padrões de eventos a serem detectados no traço de OTDR.....	64
Figura 29 - Arquitetura da RNA utilizada nesta dissertação	65

Figura 30 - Gráfico entre a comparação do número de épocas, tempo de treinamento e número de camadas escondidas	75
Figura 31 - Resultado dos testes do Caso 1 utilizando a RNA treinada.....	77
Figura 32 - Resultado dos testes do Caso 2 utilizando a RNA treinada.....	78
Figura 33 - Resultado dos testes do Caso 3 utilizando a RNA treinada.....	79
Figura 34 - Resultado dos testes do Caso 4 utilizando a RNA treinada.....	80
Figura 35 - Resultado dos testes do Caso 5 utilizando a RNA treinada.....	81
Figura 36 - Resultado dos testes do Caso 6 utilizando a RNA treinada.....	82
Figura 37 - Parte em que ocorreram os eventos no Caso 1.	86

Lista de tabelas

Tabela 1 - Comparação entre o cérebro humano e o computador à primeira vista.....	31
Tabela 2 - Índices de refração.....	55
Tabela 3 - Relação entre o número de neurônios na camada escondida, número de épocas e tempo de treinamento.....	75
Tabela 4 - Tabela de eventos encontrados a partir do teste do Caso 1.....	77
Tabela 5 - Tabela de eventos encontrados a partir do teste do Caso 2.....	78
Tabela 6 - Tabela de eventos encontrados a partir do teste do Caso 3.....	79
Tabela 7 - Tabela de eventos encontrados a partir do teste do Caso 4.....	80
Tabela 8 - Tabela de eventos encontrados a partir do teste do Caso 5.....	81
Tabela 9 - Tabela de eventos encontrados a partir do teste do Caso 6.....	82

Lista de abreviaturas e siglas

ADALINE	Neurônio Linear Adaptativo
CRT	Tubos de Raios Catódicos
EMD	<i>Empirical Mode Decomposition</i>
FBG	Redes de Bragg
IA	Inteligência Artificial
IOR	Índice de Refração Óptico
LCD	<i>Display</i> de Cristal Líquido
MIT	Instituto de Tecnologia de Massachusetts
OTDR	Reflectometro Óptico no Domínio do Tempo
RNA	Redes Neurais Artificiais
SOR	Standart OTDR Record

Lista de símbolos

dB	decibel
Hz	Hertz
km	quilomêtro
ms	milisegundo
nm	nanometro
ns	nanosegundo
s	segundo
μm	micrometro

Sumário

1	Introdução	15
1.1	<i>Introdução sobre fibras ópticas</i>	15
1.1.1	Vantagens da comunicação via fibra óptica	20
1.1.2	As desvantagens da comunicação via fibra óptica	20
1.2	<i>Caracterização de fibras ópticas</i>	21
1.3	<i>Motivação</i>	22
1.4	<i>Objetivos</i>	22
1.5	<i>Estrutura do trabalho</i>	23
2	Estado da Arte	25
2.1	<i>Introdução.....</i>	25
2.2	<i>Artigos relacionados ao assunto.....</i>	25
2.3	<i>Dissertações relacionadas ao assunto.....</i>	27
2.4	<i>Considerações finais.....</i>	27
3	Redes Neurais Artificiais	29
3.1	<i>Introdução sobre Redes Neurais Artificiais.....</i>	29
3.1.1	Cérebro humano <i>versus</i> Computador	31
3.2	<i>A REDE NEURAL ARTIFICIAL.....</i>	32
3.2.1	O neurônio biológico.....	33
3.2.2	O neurônio de Mc Culloch-Pitts.....	33
3.2.3	Rede <i>feedforward</i> de camada única	35
3.2.4	Rede <i>Feedforward</i> multicamadas	36
3.3	<i>Backpropagation.....</i>	38
3.3.1	Arquitetura	39
3.3.2	Algoritmo	39
3.3.3	Nomenclatura utilizada no treinamento	40
3.3.4	Função de ativação	41
3.3.5	Algoritmo de treinamento	43
3.3.6	Utilizando a RNA treinada.....	46
3.4	<i>Vantagens e desvantagens da rede neural.....</i>	46
3.4.1	Vantagens:	46

3.4.2	Desvantagens das Redes Neurais Artificiais	47
4	Reflectometria Óptica no Domínio do Tempo (OTDR)	47
4.1	<i>Introdução.....</i>	47
4.1.1	Espalhamento de Rayleigh.....	49
4.1.2	Reflexões de Fresnel	50
4.2	<i>Princípio De Funcionamento do OTDR</i>	51
4.2.1	Fonte de Luz.....	51
4.2.2	Acoplador/Divisor.....	51
4.2.3	Sensor Óptico	52
4.2.4	Controlador.....	52
4.2.5	Display	54
4.3	<i>Teste com OTDR.....</i>	55
4.3.1	Etapas do teste realizado com OTDR	55
4.3.2	Traço de OTDR.....	57
4.3.3	Arquivos Standard OTDR Record (SOR).....	59
4.4	<i>Aplicações do OTDR</i>	59
4.5	<i>Considerações Finais</i>	60
5	Algoritmo Proposto	61
5.1	<i>Introdução.....</i>	61
5.2	<i>Matlab.....</i>	61
5.3	<i>Fiberizer.....</i>	62
5.4	<i>Descrição do Algoritmo.....</i>	62
5.4.1	Implementação no código no Matlab.....	66
5.5	<i>Considerações finais.....</i>	73
6	Resultados	74
6.1	<i>Introdução.....</i>	74
6.2	<i>Impacto da mudança no número de neurônios na camada escondida.....</i>	74
6.3	<i>Utilizando da rede neural treinada em traços de OTDR.....</i>	76
6.4	<i>Considerações finais.....</i>	83
7	Conclusões e Contribuições	84
7.1	<i>Conclusões</i>	84

7.2	<i>Contribuições</i>	87
7.3	<i>Sugestões para trabalhos futuros</i>	87
Bibliografia		Erro! Indicador não definido.

1 Introdução

1.1 Introdução sobre fibras ópticas

Na década de 1980 ocorreu uma revolução no mundo das telecomunicações, que afetou todas o modo de vida das pessoas, aumentando a capacidade para nos comunicarmos a grandes distâncias com eficiência. Essa revolução se deu por meio da substituição do cabeamento metálico por fibras compostas por fios de vidro que transportam luz (CHOMYCZ, 2000).

Desde os primórdios da história, a luz é utilizada pelos seres humanos para comunicação de longas distâncias, porém as técnicas utilizadas tornavam a comunicação lenta e não muito eficaz. As condições atmosféricas como nevoeiros e chuvas eram fatores limitantes, bem como a necessidade de uma linha de visão. Tanto os gregos quanto os fenícios utilizavam a luz do sol que era refletida por espelhos para comunicação entre torres ou faróis. Essa mesma técnica é utilizada na era moderna com algumas variações. A luz solar foi substituída por luz artificial e o chaveamento da luz se tornou mais estruturado se tornando parecido com o código Morse (CHOMYCZ, 2000).

Em 1621, o Holandês Willebrord Snell escreve a fórmula para o princípio da refração, ou o desvio da luz, quando um raio luminoso passa de um meio para outro. Este fenômeno é facilmente observado quando colocamos um lápis dentro da água, como ilustrado na Figura 1. Quando é visto de cima, o lápis parece se distorcer, porque a luz viaja mais devagar através da água, que é mais densa opticamente que o ar. A formula de Snell foi publicada 70 anos após sua morte, indicou que toda substância transparente tem um índice de refração diferente e o ângulo que a luz se desviará é baseado nos índices de refração relativos dos dois materiais que a luz atravessara. No experimento do lápis, no copo com água, os dois meios são a água e ar, que possuem índice de refração igual a 1 e 1,33 respectivamente (WOODWARD e HUSSON, 2005).

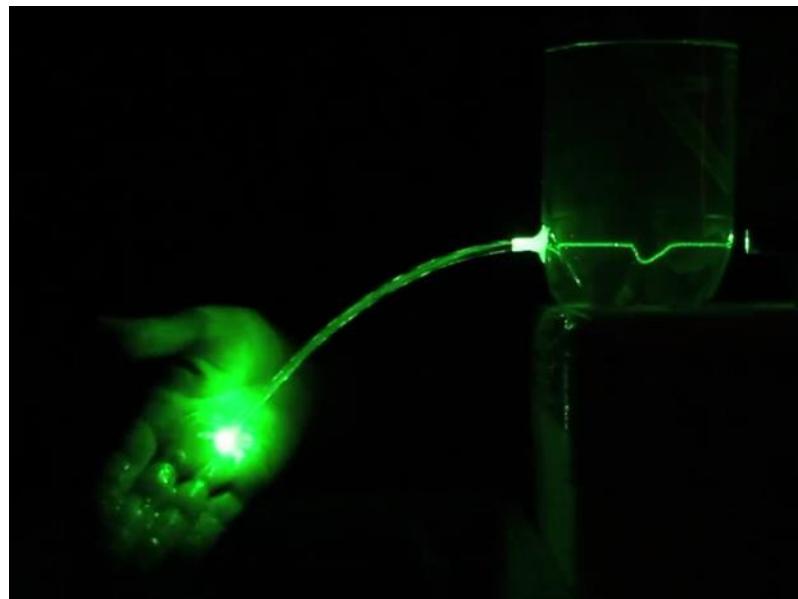
Figura 1 - Lápis no copo.



Fonte: <http://kasvi.com.br/refratometro-aplicacao/lapis-no-copo/> acessado em 10/07/2018

O próximo avanço veio através do físico suíço Daniel Colladon e o físico francês Jacques Babinet. Em 1840, Colladon e Babinet demonstraram que o brilho da luz poderia ser guiado por jatos de água através do princípio da reflexão interna total. Na demonstração, um arco de luz elétrica brilhou através de um recipiente de água. Próximo ao fundo do recipiente, na lateral, havia um buraco que a água podia sair através dele. Quando a água jorrou pelo buraco, o brilho da luz dentro do recipiente percorreu por dentro do jato de água através do arco, como é ilustrado na Figura 2. O uso desta descoberta, entretanto, era limitado a iluminações decorativas de fontes e efeitos especiais em operas. Isto levou o físico e filósofo da natureza John Tyndall a trazer o fenômeno para uma grande atenção. Em 1854, Tyndall realizou a demonstração ante a Sociedade Real Britânica e fez isto parte do seu trabalho de publicação em 1871, lançando uma sombra sobre a contribuição de Collaton e Babinet. Foi creditado a Tyndall a descoberta da reflexão interna total, porém Colladon e Babinet haviam demonstrado isto há 14 anos (WOODWARD e HUSSON, 2005).

Figura 2 - A luz que faz curva na água.



Fonte (MANUAL DO MUNDO, 2012)

A reflexão interna total acontece quando a luz passa por dentro de um meio que possui um alto índice de refração, atinge a borda entre os meios, com grande índice e baixo índice. Quando isso ocorre, a borda se transforma refletiva e a luz permanece dentro do meio.

Pouco depois que Tyndall, Colladon e Babinet lançaram as bases para passagem da luz através de um meio curvado, um outro experimento mostrou como a luz poderia ser usada para carregar um grande volume de dados (WOODWARD e HUSSON, 2005).

Em 1908 , Alexander Graham Bell trabalhou em um projeto para um "Photophone", que enviaria voz através de um feixe de luz. A luz do sol era refletida em um espelho que vibrava como as ondas sonoras de voz. O receptor era uma fotocélula conectada a uma corrente elétrica que passava para um alto falante. A ideia era boa, mas a tecnologia ainda não estava disponível para ser usada na prática. Depois que o laser foi inventado em 1958, outros estudos foram realizados com comunicação pelo uso da luz utilizando ar como meio de comunicação. Lasers forneceram uma banda estreita de radiação luminosa que poderia ser curvada com espelhos. A comunicação pela luz ainda não era prática porque exigia uma linha de visão clara; neblina ou chuva ainda apresentavam um problema de obstrução ao link. As experiências continuaram com a propagação da luz utilizando vidro como meio de propagação. Conclusões mostraram que o vidro seria melhor que o ar por duas razões, sua natureza constante e suas características mantêm inalteradas mesmo com as mudanças climáticas (CHOMYCZ, 2000).

Em 1970, a primeira fibra óptica de baixa perda foi desenvolvida. A fibra óptica, feita com vidro de sílica de aproximadamente 250 μm de diâmetro, da espessura de um cabelo

humano, foi usada na propagação de luz em um ambiente de laboratório. Este foi o início das fibras ópticas (CHOMYCZ, 2000).

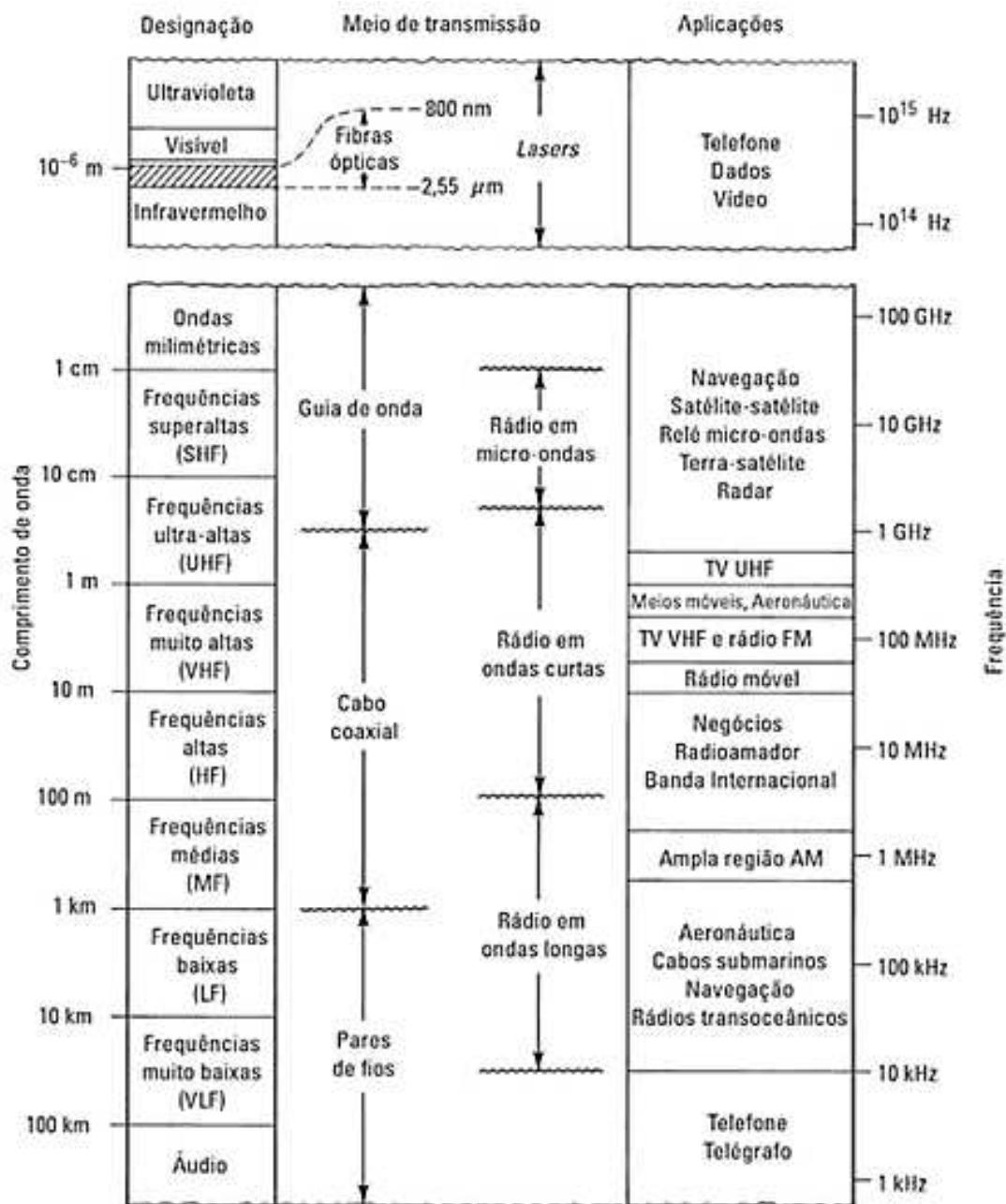
Logo depois, o processo de fabricação de fios de fibra de vidro foi consolidado e durante a década de 1970, a empresa Corning Inc já fabricava cabos de fibra óptica acessíveis comercialmente. Isto lançou a revolução da fibra óptica. Subsequentemente, sistemas de curtas distâncias foram testados por algumas companhias telefônicas. Com refinamento continuo da tecnologia, as distâncias das comunicações aumentaram e mais produtos se tornaram acessíveis (CHOMYCZ, 2000).

Em 1980, Bell anuncia a instalação de 611 milhas de fibra óptica no seu corredor do Noroeste dos Estados Unidos. Da mesma forma, Saskatchewan Telephone anunciou a instalação de 3600 km de fibra óptica no Canada. Nas Olimpíadas de inverno, que foram realizadas em Lake Placid, a fibra óptica foi utilizada pela primeira vez para transmitir um sinal de televisão. Nos anos seguintes, a fibra óptica gradualmente ganhou popularidade no mundo das telecomunicações. Hoje, ela é amplamente aceita e uma tecnologia comprovada. A substituição das velhas linhas de comunicação metálica por novos cabos de fibras ópticas é a norma para várias aplicações (CHOMYCZ, 2000).

O princípio básico do chaveamento usado na comunicação por luz no passado é similar ao princípio usado na fibra óptica, hoje. O sinal da informação a ser transmitido controla a fonte de luz, ligando-a e desligando-a na sequência particular codificada, ou pela variação de intensidade. A luz é acoplada em uma fibra óptica que guia ela pela distância da comunicação. Na recepção, um detector decodifica a luz e reproduz o sinal da informação (CHOMYCZ, 2000).

A Figura 3 ilustra as bandas de espectro eletromagnético utilizados em telecomunicações, dentre eles o utilizado em comunicações por fibra óptica.

Figura 3 - As regiões no espectro de frequências utilizados nas comunicações via rádio e fibra óptica.



Fonte: (KEISER, 2014).

Na Figura 3 é mostrado a designação das faixas de frequências, o meio de transmissão e aplicação para cada faixa de frequência.

1.1.1 Vantagens da comunicação via fibra óptica

As fibras ópticas possuem uma grande largura de banda. As ondas de luz ocupam uma largura de frequência entre 2.10^{12} Hz e $3,7.10^{12}$ Hz. Então a capacidade de carregar informação de uma fibra óptica é muito alta (BAGAD, 2009).

As fibras ópticas também oferecem uma baixa atenuação sobre longa distâncias, normalmente menos de 1 dB/km, o que permite grandes distâncias entre os repetidores (BAGAD, 2009).

Elas também são imunes ao *cross-talk* e a interferências. Os cabos ópticos possuem alta imunidade aos campos elétricos e magnéticos, uma vez que as fibras ópticas não são condutoras de eletricidade, portanto, elas não produzem campos magnéticos. Devido a este motivo elas também são imunes a interferências elétricas causadas por motores elétricos, faíscas e lâmpadas fluorescentes (BAGAD, 2009).

Uma fibra óptica é fabricada a partir de vidro de sílica ou plástico que são materiais muito mais leves que o alumínio e o cobre. Portanto o custo de transporte se torna mais baixo. O diâmetro de uma fibra é muito menor também comparado a outros cabos, o que faz com que ela ocupe menos espaço de armazenamento. Além disto eles são fortes e robustos e consequentemente podem suportar mais peso (BAGAD, 2009).

As fibras ópticas possuem mais segurança que outros tipos de cabos. É quase impossível interceptar um sinal que trafega por uma fibra óptica (BAGAD, 2009).

A mão de obra que instala os cabos ópticos correm menos riscos do que os que instalaram sistemas que utilizam cabos metálicos, pois a fibra não é condutora de eletricidade e consequentemente não há perigo de choque elétrico, pois não há altas correntes e tensões (BAGAD, 2009).

1.1.2 As desvantagens da comunicação via fibra óptica

Embora os custos de transporte sejam mais baratos, o custo inicial de instalação e configuração é muito alto comparado a outros sistemas. A manutenção e reparo dos sistemas de fibra óptico não apresentam dificuldades, porém o custo é alto devido ao fato das dimensões de uma fibra óptica serem muito pequenas. O processo de emenda requer mão de obra qualificada (BAGAD, 2009).

Fibras ópticas são mais susceptíveis a curvatura e tensão por estresse do que cabos de cobre. Apesar do custo da fibra óptica não ser caro, ainda não é rentável substitui-los quando a conexão possui poucas distâncias, pois o preço dos transdutores opto elétricos ainda são bastante elevados (BAGAD, 2009).

O comprimento dos cabos ópticos até o receptor depende de várias perdas como espalhamentos, dispersões, atenuações e reflexões, que são fatores de perdas de potência óptica em um enlace óptico (BAGAD, 2009). (RAMOS, 2002)

Embora a fibra óptica não conduza altas correntes ou tensões elétricas, nelas podem trafegar altas potências luminosas que se o operador não tomar as preocupações pode ocorrer danos na visão.

1.2 Caracterização de fibras ópticas

O uso de cabos ópticos com a tecnologia da fibra óptica não seria possível se não fosse a caracterização dos mesmos. Após a instalação de uma nova rede óptica ou a atualização de uma nova tecnologia em uma rede óptica existente, é necessário fazer uma caracterização desta rede (PINHEIRO, 2016).

A caracterização nada mais é do que verificar qual estado aquela fibra está, e se é possível passar por ela o volume de tráfego demandado, e caso não seja possível, a caracterização irá apontar em que lugar as falhas devem ser corrigidas, ou seja, em sumo a caracterização serve para certificar a rede antes do seu uso (PINHEIRO, 2016).

É imprescindível que este procedimento seja respeitado, pois, no caso de o sistema ser ativado antes dos testes e, eventualmente surgir uma falha, a localização da causa desta, bem como seu diagnóstico, pode levar muito tempo, causando transtornos como desativação e paralisação da rede. Defeitos com *software* ou *hardware* costumam ser confundidos com defeitos de cabeamento. Dessa forma, é de importância fundamental que a rede seja devidamente testada antes de ser ativada (PINHEIRO, 2016).

Em um processo de caracterização são realizados vários testes na fibra óptica, dentre eles, a medição de atenuação, dispersão cromática e dispersão do modo de polarização. Um dos equipamentos utilizado na caracterização e ou certificação das redes ópticas é o Reflectômetro Óptico no Domínio do Tempo (OTDR), que popularmente é chamado pela sua sigla OTDR. O instrumento faz uso do fenômeno do espalhamento de Rayleigh para medir o comprimento da

fibra, atenuação das emendas, atenuação nos conectores, localizar defeitos e outros. Nesta dissertação, o Capítulo 4 é dedicado a explicar o funcionamento deste equipamento, bem como o espalhamento de Rayleigh.

1.3 Motivação

O autor dessa dissertação trabalhou profissionalmente entre os anos de 2012 e 2016 com caracterização de fibras ópticas e identificou que o resultado fornecido pelos OTDRs nem sempre refletem o estado da fibra com perfeição. Segundo (W. G. HU, S. P. WAN, *et al.*, 2012), são utilizadas duas técnicas neste tipo de equipamento, a primeira técnica é chamada de método de mínimos quadrados e a segunda técnica é conhecida como método dos dois pontos, essas duas técnicas combinadas são utilizadas para detecção de eventos em traços de OTDR. Esse algoritmo é relativamente simples e fácil de implementar, mas, ao mesmo tempo, é facilmente afetado pelo ruído e sua precisão não é alta.

Várias empresas multinacionais estão utilizando inteligência artificial (IA) para poderem melhorar os seus serviços, como por exemplo o Amazon, Google, IBM e Microsoft (DAVID, 2007). As Redes Neurais Articiais (RNA) é um dos ramos da inteligência artificial e embora ela exista há 50 anos, como será apresentado no Capítulo 3, o aumento da capacidade de processamento colocou a IA na vanguarda de decisões estratégicas.

O fato das técnicas utilizadas nos OTDRs serem bastante simples e possuírem algumas limitações com o fato das RNA estarem em alta e poderem resolver problemas ligados a reconhecimento de padrões motivaram o desenvolvimento deste trabalho.

1.4 Objetivos

Como foi citado anteriormente, as técnicas utilizadas nos *softwares* dos OTDRs são muitos simples e consequentemente os resultados encontrados pelo uso dessas técnicas não atende o necessário em alguns testes. Embora a caracterização de fibras ópticas já seja um procedimento comum no ramo de comunicações ópticas há algum tempo, poucas técnicas estão sendo utilizadas para melhorar a precisão e contornar os problemas causados por ruídos, uma delas é a utilização de transformadas wavelets, que serão abordadas no próximo capítulo (W. G. HU, S. P. WAN, *et al.*, 2012).

O objetivo deste trabalho é desenvolver um algoritmo utilizando a técnica de IA, RNA, para que, assim como no uso das transformadas wavelets, contornar os problemas causados pelo ruído, de forma mais precisa que os *softwares* que são utilizados atualmente pelos equipamentos.

Ao final, espera-se detectar a localização de eventos de emenda térmica, também conhecidos popularmente como fusão em um traço de OTDR utilizando inteligência artificial.

1.5 Estrutura do trabalho

A estrutura desta dissertação foi dividida em seis capítulos, com os respectivos títulos:

1. Introdução (este capítulo);
2. Estado da Arte;
3. Redes Neurais Artificiais;
4. Equipamento de OTDR;
5. Algoritmo Proposto;
6. Resultados;
7. Conclusões e Contribuições.

O Capítulo 1, Introdução, faz um breve histórico sobre a tecnologia das fibras ópticas, as principais vantagens e desvantagens de se utilizar essa tecnologia, as motivações que levaram a desenvolver este trabalho e os objetivos do mesmo. Também há neste capítulo, a estruturação do trabalho que tem o objetivo de facilitar o leitor do que se trata os próximos capítulos.

No Capítulo 2, Estado da Arte, são mostrados alguns artigos que foram publicados na área, o que ajuda a justificar a relevância deste trabalho.

No Capítulo 3, Redes Neurais Artificiais, em um primeiro momento é feito uma comparação entre o cérebro humano e um computador, a seguir é feito um histórico sobre o desenvolvimento das RNAs no decorrer das décadas. É explicado como funciona o modelo de neurônio artificial, redes de camadas únicas, redes multicamadas. No capítulo existe uma seção dedicada ao algoritmo de treinamento *backpropagation* que foi o que deu as redes neurais o poder de resolver problemas mais complexos e o que foi utilizado neste trabalho.

O Capítulo 4, Equipamento de OTDR, descreve o funcionamento do equipamento de OTDR. Nele é explicado o que são reflexões de Fresnel e espalhamento de Rayleigh. O capítulo também descreve o funcionamento do equipamento através das partes que o compõe. Também

é tratado o formato de arquivo que é utilizado nos equipamentos de OTDR e as aplicações do equipamento.

No Capítulo 5, Algoritmo Proposto, é feita uma introdução de como o que foi descrito nos capítulos 3 e 4 serão utilizados no algoritmo. É apresentada a ferramenta computacional utilizada para desenvolver o algoritmo e a outra ferramenta que é utilizada para fazer a leitura dos traços de OTDR e conversão dos arquivos. Também é explicado no capítulo o passo-a-passo do programa desenvolvido.

No Capítulo 6, Resultados, no primeiro momento é apresentado o impacto do número de neurônios na camada escondida que é um dos parâmetros da de uma RNA, explicado no Capítulo 3, no tempo de treinamento e no número de épocas que também é outro parâmetro de uma RNA. É apresentado também a utilização do programa desenvolvido em cinco casos reais diferentes.

Por fim no Capítulo 7, são apresentadas as conclusões sobre os resultados do Capítulo anterior, as contribuições deste trabalho e sugestões para trabalhos futuros.

2 Estado da Arte

2.1 Introdução

Neste capítulo serão apresentados alguns artigos e dissertações publicadas que utilizam outras propostas para a detecção de eventos em traços de OTDR. Estes artigos ajudam a justificar a relevância dessa dissertação. Não existem muitos trabalhos realizados com o propósito de melhorar as análises de traços de OTDR e os trabalhos encontrados foram publicados recentemente na última década.

2.2 Artigos relacionados ao assunto.

No artigo (HAN QIANG, ZHANG ZHIHUI, *et al.*, 2015), os autores ressaltam que as técnicas que até o presente momento, para a análise do traço de OTDR são relativamente simples, fácil de implementar, mas ao mesmo tempo, são facilmente afetado pelo ruído e a acurácia não é muito alta. As técnicas convencionais utilizam dois métodos combinados, o método de 2 pontos e os métodos dos mínimos quadrados para detecção de eventos e traços de OTDR.

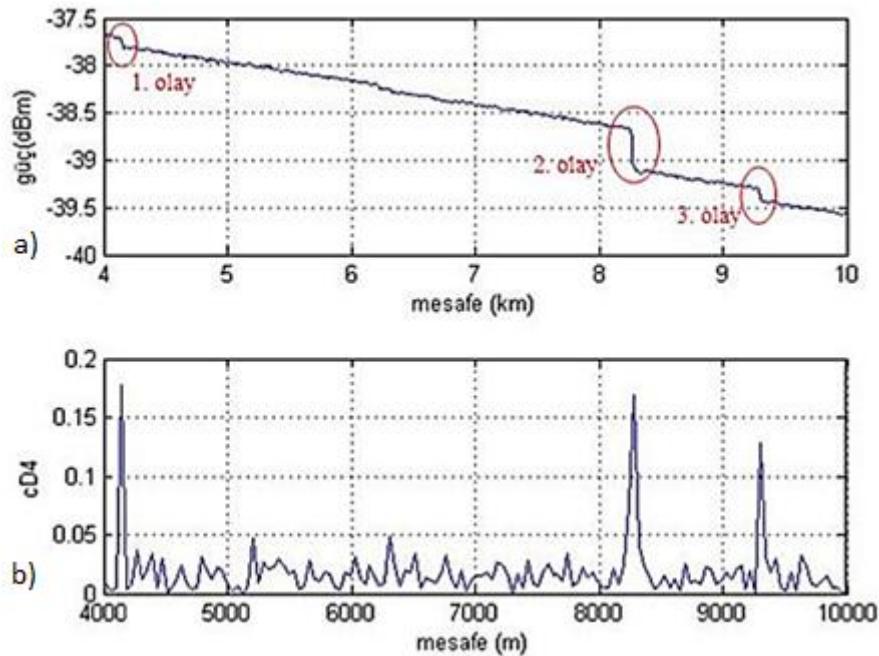
Para reduzir a influência do ruído e melhorar as características do sinal, os autores aplicaram uma técnica chamada Empirical Mode Decomposition (EMD). EMD é um método de análise de sinal adaptativo eficaz, que é apropriado para a análise no domínio do tempo e da frequência, é considerado como o grande avanço da análise de sinal baseada em Fourier.

Para a detecção dos eventos no OTDR, os autores deste artigo utilizaram a transformada wavelet que é uma ferramenta eficiente para análise de sinal e vem sido amplamente utilizada em processamentos de sinais. Em regiões de baixa frequência, grandes janelas de tempo são fornecidas por transformada wavelet, enquanto que em regiões de alta frequência, janelas de tempo estreitas são fornecidas. Essa característica é uma vantagem importante da transformada wavelet. Os pontos de variações agudas, que muitas vezes expressam as características mais importantes dos sinais, são chamados de singularidades. A transformada wavelet é extraordinariamente sensível para as singularidades do sinal, a partir dos coeficientes wavelets empíricos, é possível obter as posições e amplitudes das singularidades. O módulo máximo da transformada wavelet tem uma relação direta com a regularidade de Lipschitz, que é usada para

detectar as singularidades de uma maneira matemática. Os eventos no sinal OTDR podem ser posicionados com precisão, pesquisando-se o módulo local máximo.

Nos artigos (CHAOJU e JUN, 2010) e (ACAR, 2016), foi utilizado somente a transformada wavelet para encontrar os eventos no traço de OTDR, ou seja, não foi feito nenhum tipo de pré-processamento para reduzir o ruído. No artigo (CHAOJU e JUN, 2010), justifica-se o uso da transformada wavelet devido a capacidade deste tipo de transformada de não perder informações espaciais ao contrário da Transformada de Fourier. O traço de OTDR de (ACAR, 2016) é apresentado na Figura 4a e os resultados encontrados são ilustrados na Figura 4b e será usada no Capítulo 7, Conclusões e Contribuições, para comparação com o método utilizado nesta dissertação.

Figura 4 - Analise por Wavelet de eventos de fusão .



Fonte: (ACAR, 2016)

No que tange a utilização de Redes Neurais Artificiais para resolver problemas relacionados a fibra óptica, no artigo (NAYAK, KAMATH, *et al.*, 2006), foi utilizado a fibra óptica como sensor para detectar a qualidade da água. Uma fibra óptica colocada em pipetas com água e diferentes porcentagens de etanol. Foram realizados testes com o equipamento de OTDR, a partir dos testes, foi treinado uma rede neural artificial para identificar a quantidade de etanol na água. Com a rede neural treinada, ou seja, os pesos desta rede ajustados, ela foi

testada para a verificar a precisão, ou seja o porcentagem de acertos que a rede neural obteve nas pipetas.

2.3 Dissertações relacionadas ao assunto

Embora não haja muitos trabalhos no ramo das comunicações ópticas, alguns trabalhos foram realizados na área de engenharia elétrica. Nas dissertações (RAMOS, 2002) e (OLIVEIRA, 2005) foi utilizada a RNA para localização de falhas em linhas de transmissão, porém em (RAMOS, 2002) utilizou-se RNA para a obtenção de uma nova forma de identificar o tipo do curto-circuito e determinar a sua localização, utilizando dados obtidos em somente um terminal.

Enquanto que em (RAMOS, 2002), utilizou-se RNA diretamente alimentadas, por meio da análise de tensões e correntes trifásicas na frequência fundamental, também somente em um dos terminais da linha. Em ambos os casos, verificou-se que a utilização de RNAs para a identificação e a localização de falta mostrou-se bastante eficiente para os casos analisados, comprovando a aplicabilidade das redes neurais nesse problema.

Na dissertação, (RAMOS, BRAGA e BARBOSA, 2013) a metodologia empregada teve como base a análise dos padrões de deformação superficial, obtidos com o uso de grades de sensores à fibra óptica baseadas em redes de Bragg (FBG). A detecção é realizada comparando a energia das deformações superficiais dos corpos de prova nos casos com e sem defeito, enquanto a caracterização é obtida através da utilização de RNA, por meio de rotinas de reconhecimento de padrões.

2.4 Considerações finais.

Este capítulo apresentou alguns trabalhos envolvendo o assunto abordado nesta dissertação. Os principais artigos publicados envolvendo a detecção de eventos em traços de OTDR utilizam a transformada wavelet como metodologia, enquanto que os casos que se utilizam RNA em fibra óptica, que é o caso de (NAYAK, KAMATH, *et al.*, 2006) e (RAMOS, BRAGA e BARBOSA, 2013) a fibra óptica foi utilizada como sensor e não para transmissão de dados.

Também é importante destacar os outros trabalhos, (OLIVEIRA, 2005) e (RAMOS, 2002) que embora não tratem de comunicações via fibra óptica, são relacionados a área de engenharia elétrica e utilizam RNA para a detecção de falhas.

3 Redes Neurais Artificiais

3.1 Introdução sobre Redes Neurais Artificiais

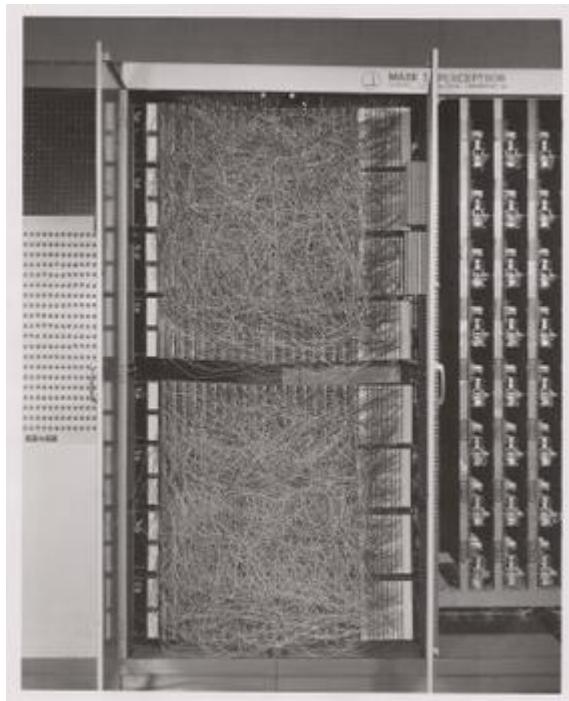
A história das RNA começa na década de 1940 quase simultaneamente a história da computação eletrônica programada (DAVID, 2007). Em 1945, Warren McCulloch e Walter Pitts introduziram os modelos de redes neurológicas, recriaram parâmetros de limiar baseados em neurônios e mostraram que mesmo redes simples desse tipo são capazes de calcular praticamente qualquer lógica ou função aritmética (DAVID, 2007). Em 1947, McCulloch e Pitts indicaram um campo prático de aplicação, o reconhecimento de padrões espaciais por RNA (DAVID, 2007).

Em 1949, Donald O. Hebb formulou a regra clássica Hebbiana, que representa na sua forma mais generalizada a base de quase todos os procedimentos de aprendizagem neural. A regra sugere que a ligação entre dois neurônios é reforçada quando ambos os neurônios estão ativos ao mesmo tempo. Essa mudança na força é proporcional ao produto das duas atividades. Hebb poderia postular essa regra, mas, por causa da ausência de pesquisas neurológicas, ele não foi capaz de verificar isso (DAVID, 2007).

Em 1951, em sua dissertação Marvin Minsky desenvolveu o neurocomputador Snark, que era capaz de ajustar os pesos automaticamente, porém nunca foi implementado na prática, uma vez que ele calculava, mas ninguém sabia o que realmente ele calculava (DAVID, 2007).

Em 1957-1958, no Instituto de Tecnologia de Massachussets (MIT), Frank Rosenblatt, Charles Wightman e seus colaboradores desenvolveram o primeiro neurocomputador de sucesso, o perceptron Mark I, que era capaz de reconhecer números simples por meio de um sensor de imagem de 20×20 pixels e trabalhava eletromecanicamente com 512 motores (DAVID, 2007) . Potenciômetros acionados - cada potenciômetro representando um peso que é ilustrado na Figura 5 a seguir

Figura 5 - Mark I Perceptron



Fonte: <https://en.wikipedia.org/wiki/Perceptron>, acessado em 16/07/2018

Em 1959, Frank Rosenblatt descreveu diferentes versões do perceptron, formulou e verificou seu teorema de convergência de perceptrons. Ele descreveu camadas de neurônios que imitam a retina, chaves de limiar e uma regra de aprendizado que ajusta os pesos de conexão (DAVID, 2007).

Em 1960, Bernard Widrow e Marcian E. Hoff apresentaram o Neurônio Linear Adaptativo (ADALINE), um sistema de aprendizado adaptativo rápido e preciso, sendo a primeira RNA amplamente usada comercialmente: ela pode ser encontrada em quase todos os telefones analógicos na filtragem adaptativa de eco em tempo real através da regra de Widrow-Hoff ou regra delta. Naquela época, Hoff, co-fundador da Intel Corporation, foi aluno de PhD de Widrow, que é conhecido como o inventor de microprocessadores modernos. Uma vantagem da regra delta sobre o algoritmo de aprendizado do perceptron original era sua adaptabilidade (DAVID, 2007).

Em 1969, Marvin Minsky e Seymour Papert publicaram uma análise matemática precisa do perceptron para mostrar que o modelo do perceptron não era capaz de representar problemas não lineares, como por exemplo, a porta lógica Ou Exclusivo, o que colocou fim à sua popularidade, superestimação e fundos de pesquisa, o que resultou em um congelamento das pesquisas durante 15 anos (DAVID, 2007).

Em 1985, John Hopfield publicou um artigo descrevendo uma maneira de encontrar soluções aceitáveis para o problema do caixeiro viajante usando redes Hopfield (DAVID, 2007).

Em 1986, O procedimento de aprendizado *Backpropagation of error*, como uma generalização da regra delta foi separadamente desenvolvido e amplamente publicado pelo Parallel Distributed Processing Group. Problemas não linearmente separáveis puderam ser resolvidos por perceptrons multicamadas, e as avaliações negativas de Marvin Minsky foram refutada. A partir deste momento, o desenvolvimento do campo de pesquisa foi quase explosivo (DAVID, 2007).

3.1.1 Cérebro humano *versus* Computador

O cérebro humano e os computadores possuem várias diferenças, porém a principal delas, é a capacidade de aprendizado. Enquanto os cérebros dos humanos podem aprender, os computadores possuem unidades de processamento que os permite realizar cálculos matemáticos complexos em um curto período de tempo, porém elas não são adaptativas. Se for comparado um computador com um cérebro, é possível notar que teoricamente o computador deveria ser mais poderoso que cérebros humanos. Por exemplo, um computador pode ter 1 bilhão de transistor e o tempo de resposta destes é de 1 ns, enquanto os cérebros contem 100 bilhões de neurônios que tem o tempo de resposta de aproximadamente 1 ms (DAVID, 2007).

Tabela 1 - Comparaçāo entre o cérebro humano e o computador à primeira vista.

	Cérebro	Computador
Unidades de processamento	$\approx 10^{11}$	$\approx 10^9$
Tipo de unidade de processamento	Neurônio	Transistor
Tipo de calculo	Massivamente paralelo	Baseado em endereçamento
Tempo de resposta	$\approx 10^{-3}$ s	$\approx 10^{-9}$ s
Mudanças de operações possíveis	$\approx 10^{13}$ 1/s	$\approx 10^{18}$ 1/s
Mudanças de operações reais	$\approx 10^{12}$ 1/s	$\approx 10^{10}$ 1/s

Fonte: (DAVID, 2007)

A maior parte do cérebro trabalha continuamente, enquanto a maior parte de um computador é apenas armazenamento de dados passivo. Assim, o cérebro tem um funcionamento em paralelo e, portanto, está próximo de seu máximo teórico. Por outro lado, o computador, está a uma certa ordem de distância de seu máximo, conforme é mostrado na

Tabela 1. Além disso, um computador é estático, enquanto que o cérebro é uma rede neural biológica que pode se reorganizar durante sua "vida útil" e, portanto, é capaz de aprender, compensar erros e assim por diante (DAVID, 2007).

3.2 A REDE NEURAL ARTIFICIAL

Uma RNA é um sistema de processamento de informação que possui certas características de desempenho em comum com as redes neurais biológicas (HAYKIN, 2009). RNA foram desenvolvidas como generalizações de modelos matemáticos da cognição humana ou da biologia neural, com base nas suposições de que

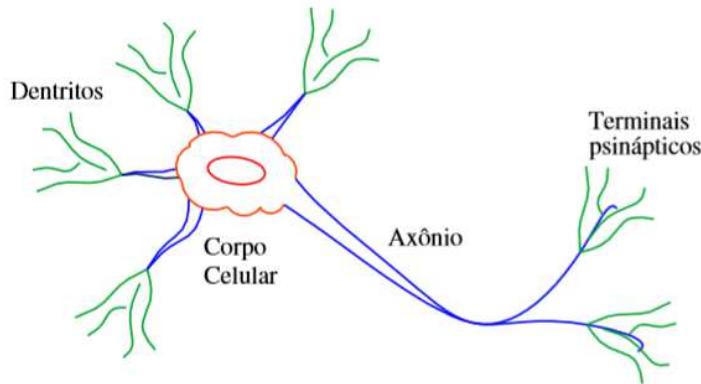
1. O processamento da informação ocorre em alguns elementos simples chamados neurônios.
2. Sinais são passados entre os neurônios através de *links* de conexão.
3. Cada *link* de conexão tem um peso associado, que, em uma RNA típica, multiplica o sinal transmitido.
4. Cada neurônio aplica uma função de ativação (geralmente não-linear) para sua entrada da rede (soma dos sinais de entrada ponderados) para determinar seu sinal de saída.

Uma RNA é caracterizada pela sua arquitetura, que é o padrão de conexão entre os neurônios, o treinamento, aprendizagem, algoritmo, que é o método para determinar os pesos das conexões e a sua função de ativação. E consiste em um grande número de elementos de processamento simples chamados neurônios, unidades, células ou nós. Cada neurônio é conectado a outros neurônios por meio de *links* de comunicação direcionados, cada um com um peso associado. Os pesos representam informações sendo usadas pela rede para resolver um problema.

As RNA podem ser aplicadas a uma ampla variedade de problemas, como armazenar e recuperar dados ou padrões, classificar padrões, executar mapeamentos gerais de padrões de entrada para padrões de saída, agrupar padrões semelhantes ou encontrar soluções para problemas de otimização restritos (HAYKIN, 2009).

3.2.1 O neurônio biológico

Figura 6 - Esquema básico do neurônio biológico



Fonte: (MEJÍA SÁNCHEZ e MEGGIOLARO, 2009)

A Figura 6 ilustra o neurônio biológico. No cérebro humano o neurônio recebe sinais de outros neurônios através dos dendritos, cada um destes sinais é multiplicado pelo próprio peso da conexão. Estes sinais são adicionados no corpo celular ou função somatória e ,quando este sinal composto alcança um valor umbral, um sinal potencial é enviado pelo axônio, o qual é a saída do neurônio (HAYKIN, 2001).

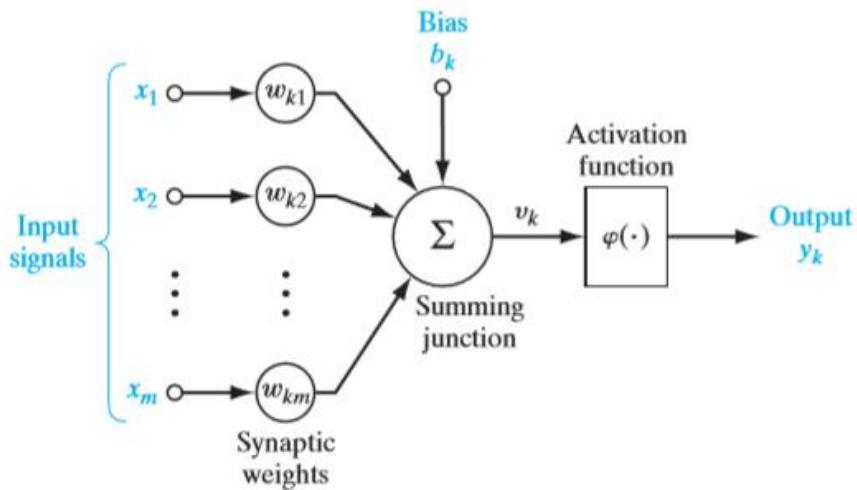
3.2.2 O neurônio de Mc Culloch-Pitts

Um neurônio é uma unidade de processamento de informações que é fundamental para a operação de uma RNA. O diagrama de blocos da Figura 7 mostra o modelo de um neurônio, que forma a base para projetar uma grande família de RNAs (HAYKIN, 2009). Os três elementos básicos do modelo neural são:

- 1- Um conjunto de sinapses, ou elos de ligação, cada um dos quais é caracterizado por um peso ou força própria. Especificamente, um sinal x_j na entrada da sinapse ligada ao neurônio k é multiplicado pelo peso sináptico w_{kj} . É importante observar a maneira que os pesos sinápticos w_{kj} são escritos . O primeiro subscrito em $w_{kj}, 'k'$ refere-se ao neurônio em questão, e o segundo subscrito,'j', refere à extremidade de entrada da sinapse à qual o peso refere-se.
- 2- Um somador para os sinais de entrada, ponderados pelas respectivas forças sinápticas do neurônio, as operações descritas constituem um combinador linear.

- 3- Uma função de ativação para limitar a amplitude da saída do neurônio. Também podemos nos referir a função de ativação como uma função logística (ou uma curva logística é um formato de “S” comum), na medida em que atinge (limita) a faixa de amplitude permitida do sinal de saída para algum valor finito.

Figura 7 - Modelo de Neurônio não linear.



Fonte: (HAYKIN, 2009)

O modelo neural da Figura 7 também inclui um *bias* aplicado externamente, chamado de b_k . O *bias* b_k tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se é positiva ou negativa, respectivamente. Em termos matemáticos, podemos descrever o neurônio k descrito na Figura 7, escrevendo o par de equações, Equação (1) e Equação (2):

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (1)$$

e

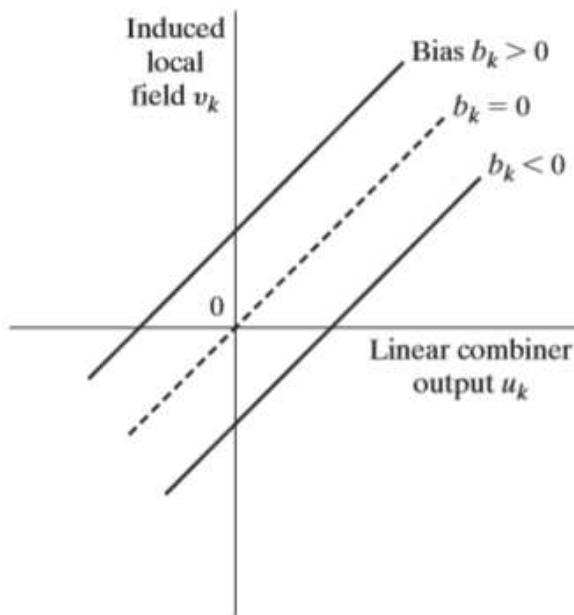
$$y_k = \varphi(u_k + b_k) \quad (2)$$

Em que x_1, x_2, \dots, x_m são sinais de entrada, $w_{k1}, w_{k2}, \dots, w_{km}$ são respectivamente os pesos sinápticos do neurônio k, u_k é a saída do combinador linear devida aos sinais de entrada; b_k é o *bias*; $\varphi(\cdot)$ é a função de ativação; e y_k é o sinal de saída do neurônio. O uso de bias b_k tem o efeito de aplicar uma transformação afim à saída u_k do combinador linear no modelo da Figura 7, como mostrado pela Equação (3):

$$v_k = u_k + b_k \quad (3)$$

Em particular, dependendo se o *bias* ‘ b_k ’ é positivo ou negativo, a relação entre o campo local induzido, ou potencial de ativação, v_k do neurônio k e a saída do combinador linear u_k é modificada da maneira ilustrada na Figura 8. Observe que, como resultado dessa transformação afim, o gráfico de v_k versus u_k não passa mais pela origem.

Figura 8 - Transformação afim causada pelo bias.



Fonte: (HAYKIN, 2009)

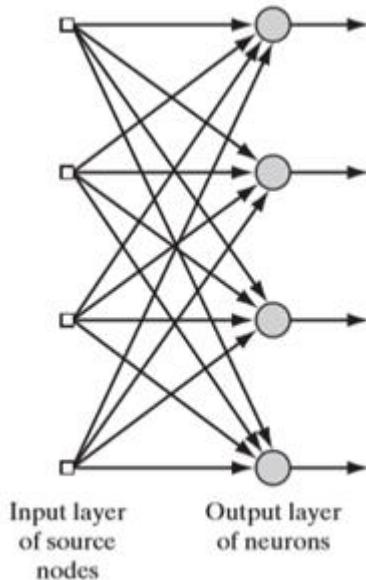
Cada neurônio tem um estado interno, chamado de ativação ou nível de atividade, que é uma função das entradas recebidas. Normalmente, um neurônio envia sua ativação como um sinal a vários outros neurônios. É importante dizer que um neurônio pode enviar apenas um sinal de cada vez, embora esse sinal seja transmitido para vários outros neurônios.

3.2.3 Rede *feedforward* de camada única

A rede de camada única é a forma mais simples que temos de uma rede em camadas. Neste tipo de rede, temos uma camada de entrada de nós de origem que se projeta diretamente em uma camada de saída de neurônios (nós de cálculos), mas não vice-versa. Em outras palavras, essa rede é estritamente de um tipo de *feedforward*. A Figura 9 ilustra o caso de quatro nós nas camadas de entrada e saída. Essa rede é um exemplo de rede de camada única, a

designação de “camada única” se refere à camada de saída dos nós de cálculos (neurônios). Não é considerada a camada de entrada dos nós de origem, porque nenhum cálculo é executado lá.

Figura 9 - Rede *feedforward* com uma única camada.



Fonte: (HAYKIN, 2009)

3.2.4 Rede *Feedforward* multicamadas

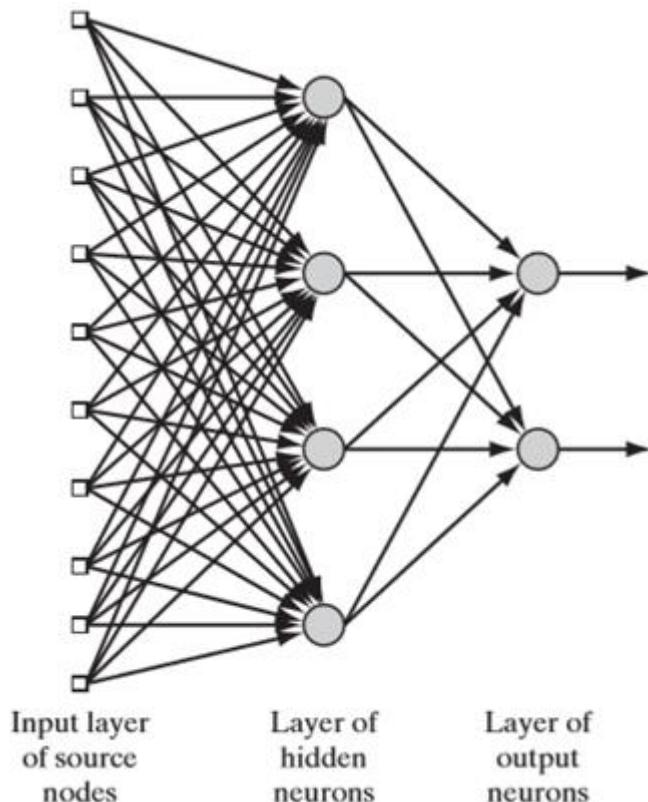
A segunda classe de rede neural *feedforward* se distingue da primeira pela presença de neurônios ocultos, na chamada camada oculta. O termo “oculto” se dá devido ao fato dessa parte da rede neural não ser vista diretamente na entrada ou na saída da rede. Ao adicionar uma ou mais camadas ocultas, a rede consegue extraír estatísticas de ordem superior à de sua entrada, ou seja, a rede adquire uma perspectiva global, apesar de sua conectividade ser local, devido ao conjunto extra de conexões sinápticas e dimensão extra de interações neurais.

Os nós de origem na camada de entrada da rede fornecem os respectivos elementos do padrão de ativação (vetor de entrada), que constituem os sinais de entrada aplicados aos neurônios (nós de cálculos) na segunda camada (ou seja, a primeira camada oculta), os sinais da segunda camada são usados como entradas para a terceira camada, e assim por diante, para o resto da rede.

Normalmente, os neurônios em cada camada da rede têm como entradas apenas os sinais de saída da camada precedente. Os sinais de saída dos neurônios na camada de saída (final) da rede constituem a resposta geral da rede ao padrão de ativação fornecido pelos nós de origem na camada de entrada.

A Figura 10 ilustra o *layout* de uma RNA *feedforward* com várias camadas, para o caso de uma única camada oculta. A Figura 10 ilustra uma rede 10–4–2 porque possui 10 nós de origem, 4 neurônios ocultos e 2 neurônios de saída (HAYKIN, 2009).

Figura 10 - Rede feedfoward totalmente conectada com uma camada escondida e uma camada de saída.



Fonte: (HAYKIN, 2009)

A RNA na Figura 10 é chamada de totalmente conectada devido ao fato de que cada nó em cada camada da rede está conectado a todos os outros nós na camada direta adjacente. Se, no entanto, alguns dos links de comunicação (conexões sinápticas) estiverem faltando na rede, dizemos que a rede está parcialmente conectada.

Uma rede multicamadas permite a ela resolver problemas que uma rede com apenas uma camada não resolveria, por outro lado, é mais difícil de treinar redes multicamadas (HAYKIN, 2009). Não há uma regra que determine os parâmetros das camadas intermediárias, confiavelmente, os pesos são ajustados de acordo com o desempenho da rede e depende do profissional que está implementando uma RNA (FAUSETT, 1992).

3.3 Backpropagation

Como citado na introdução deste capítulo, as RNA com apenas uma camada possuem algumas limitações, o que fez com que ocorresse um declínio no interesse pela área na década de 1970. Com o surgimento do método de treinamento *backpropagation* foi possível contornar essas limitações. Pois a natureza muito geral do método de treinamento *backpropagation* significa que uma rede de retropropagação (uma rede de múltiplas camadas, alimentada por retropropagação) pode ser usada para resolver problemas em muitas áreas (FAUSETT, 1992).

O *backpropagation* é, indiscutivelmente, o algoritmo mais importante na história das RNA – sem *backpropagation* (eficiente), seria impossível treinar redes de aprendizagem profunda da forma que é vista hoje. O *backpropagation* pode ser considerado a pedra angular das redes neurais modernas e aprendizagem profunda (DATA SCIENCE ACADEMY, 2018).

O algoritmo é bastante simples e pode ser implementado em poucas linhas de código, e o algoritmo, até hoje, sem dúvida foi implementado milhares de vezes (DATA SCIENCE ACADEMY, 2018). Aplicações que usam essas redes podem ser encontrados em praticamente todos os campos que usam redes neurais para problemas que envolvem o mapeamento de um determinado conjunto de entradas para um conjunto especificado de saídas de destino (isto é, redes que usam treinamento supervisionado) (FAUSETT, 1992).

Como é o caso da maioria das redes neurais, o objetivo é treinar a rede para alcançar um equilíbrio entre a capacidade de responder corretamente aos padrões de entrada usados para treinamento (memorização) e a capacidade de fornecer respostas razoáveis (boas) à entrada, isso é similar, mas não idêntico, àquele usado no treinamento (generalização) (FAUSETT, 1992).

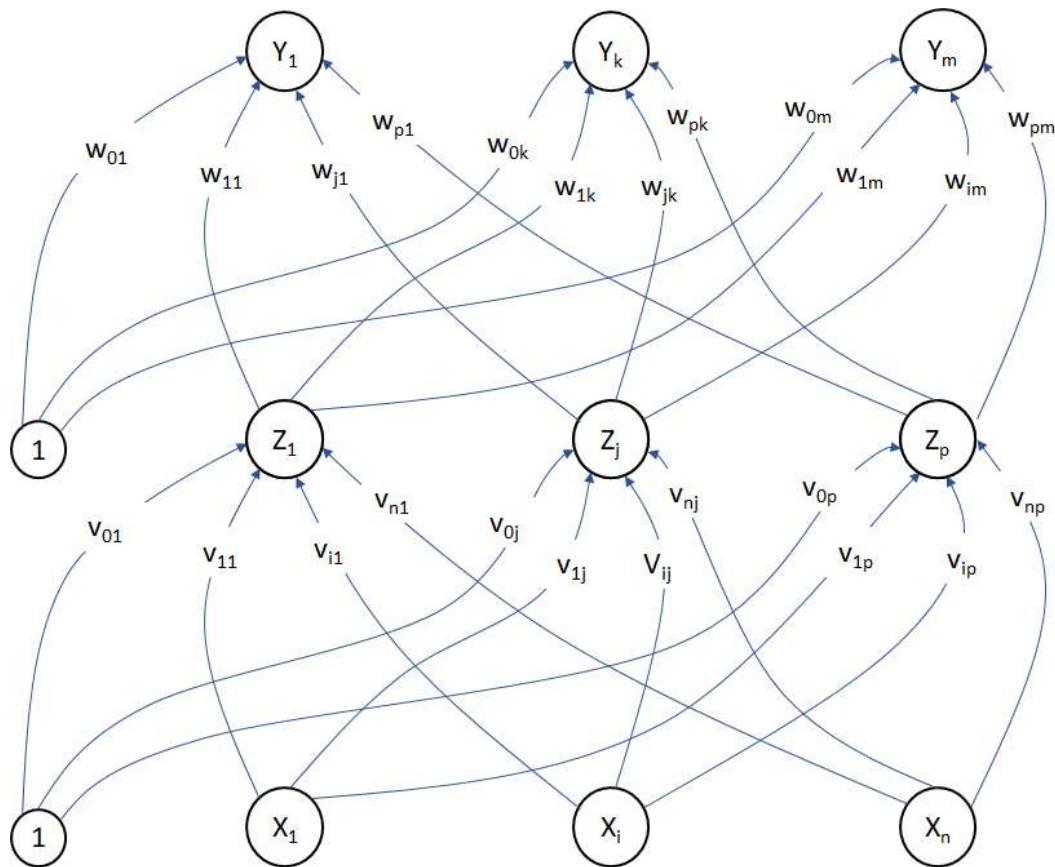
O processo de aprendizado do algoritmo *backpropagation* tem como objetivo a atualização dos pesos através da minimização do erro quadrático médio pelo método do Gradiente Descendente.

O treinamento de uma rede *backpropagation* envolve três estágios: o *feedforward* do padrão de treinamento de entrada, o cálculo e retropropagação do erro associado e o ajuste dos pesos. Após o treinamento, a aplicação da rede envolve apenas os cálculos da fase de *feedforward*. Mesmo que o treinamento seja lento, uma rede treinada pode produzir sua saída muito rapidamente. Numerosas variações de retropropagação foram desenvolvidas para melhorar a velocidade do processo de treinamento (FAUSETT, 1992).

3.3.1 Arquitetura

Para pode ser treinada utilizando o aprendizado *backpropagation*, a rede necessariamente deve ter uma ou mais camadas ocultas como ilustrado na Figura 11 abaixo em que a camada escondida é representada pelos neurônios Z, os neurônios de saída são os neurônios Y e a camada de entrada pelos neurônios X.

Figura 11 - RNA multicamadas com uma camada escondida.



Fonte: Evandro Monteiro Jorge Júnior, 2018. baseado em uma ilustração em (FAUSETT, 1992).

3.3.2 Algoritmo

Como mencionado anteriormente, o treinamento de uma rede por *backpropagation*, envolve três estágios, o *feedforward* das entradas padrões de treinamento, a retropropagação do erro associado e o ajuste dos pesos.

Durante o *feedforward*, cada unidade de entrada (X_i) recebe um sinal de entrada e envia este sinal para cada uma das unidades ocultas Z_1, \dots, Z_p . Cada unidade oculta calcula sua ativação e envia seu sinal (z_j) para cada unidade de saída. Cada unidade de saída (Y_k) calcula sua ativação (y_k) para formar a resposta da rede para o dado padrão de entrada.

Durante o treinamento, cada unidade de saída compara a sua saída y_k , já calculada a ativação, com um objetivo t_k para determinar o erro associado para esse padrão com esse neurônio. Baseado neste erro o fator δ_k ($k=1, \dots, m$) é calculado. δ_k é usado para distribuir o erro nos neurônios de saída Y_k , para todos os neurônios da camada escondida da camada anterior (os neurônios escondidos conectados a Y_k). δ_k também será usado, posteriormente, para atualizar os pesos entre as camadas de saída e a camada escondida. De maneira similar, o fator δ_j ($j=1, \dots, p$) é calculado para cada neurônio escondido Z_j . Não é necessário propagar o erro para a camada de entrada da rede, porém δ_k é utilizado para atualizar os pesos entre a camada escondida e a camada de entrada (FAUSETT, 1992).

Após todos os fatores δ serem determinados, os pesos de todas as camadas são ajustados simultaneamente. O ajuste do peso w_{jk} (do neurônio da camada escondida Z_j para a saída Y_k) é baseado no fator δ_k e da ativação de z_j dos neurônios escondidos Z_j . O ajuste do peso v_{ij} (do neurônio da camada de entrada X_j para o neurônio da camada escondida Z_j) é baseado no fator δ_j e da ativação de x_j da camada de entrada (FAUSETT, 1992).

3.3.3 Nomenclatura utilizada no treinamento

Nesta seção são apresentadas as nomenclaturas utilizadas no algoritmo de treinamento.

x – vetor de entrada do treinamento, $x=(x_1, \dots, x_i, \dots, x_n)$.

t – vetor objetivo da saída, $t=(t_1, \dots, t_k, \dots, t_m)$.

δ_k - parte do ajuste de peso da correção de erros para w_{jk} que é devido a um erro na unidade de saída Y_k , também, a informação sobre o erro na unidade Y_k que é propagada de volta para as unidades escondidas que alimentam a unidade Y_k .

δ_j - parte do ajuste de peso da correção de erros para v_{ij} que é devido à retropropagação de informações de erro da camada de saída para a unidade oculta Z_j .

α – Taxa de aprendizado, é um valor constante no intervalo entre 0 e 1 e está ligado à convergência do processo de aprendizado. Ela influênciaria a variação dos pesos nas iterações, ou seja, uma taxa de aprendizado pequena faz com que a atualização dos pesos seja pequena, fazendo com que o número de iterações aumente, porém, uma taxa de aprendizado mais alta,

faz com que aumente a velocidade de aprendizado diminuindo o número de iterações, no entanto, pode acontecer da rede neural não convergir nessa situação.

X_i – unidade de entrada i .

v_{oj} – Bias para a unidade escondida j .

Z_j – Unidade escondida j .

w_{ok} – Bias para a unidade de saída k .

Y_k – Unidade de saída.

3.3.4 Função de ativação

Uma função de ativação para uma rede *backpropagation* deve ter algumas importantes características: deve ser continua, diferenciável e estritamente crescente. Além disso, para eficiência computacional, é desejável que a derivada da função seja fácil de ser calculada. Para a maioria das funções de ativação utilizadas, o valor da derivada pode ser expressa em termos dos valores da função. Habitualmente, espera-se que a função sature, aproxime do máximo e do mínimo infinito de forma assintótica (FAUSETT, 1992).

Uma das mais comuns funções de ativação é a função binária sigmoide que tem o alcance de (0,1) e é definida pela Equação (4)

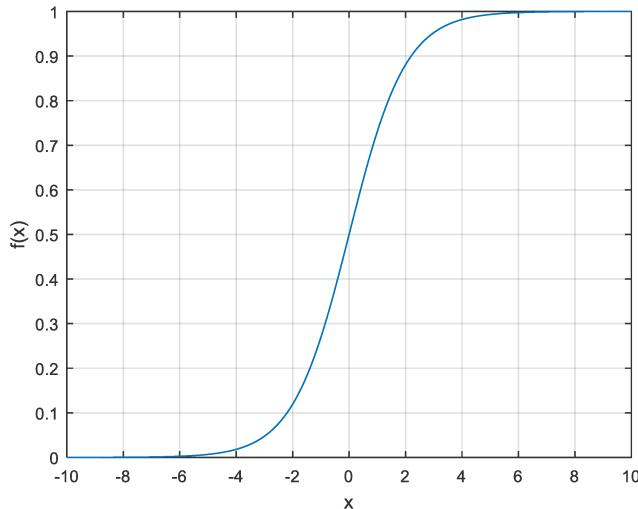
$$f(x) = \frac{1}{1 + \exp(-x)} \quad (4)$$

Com a derivada sendo a Equação (5)

$$f'(x) = f(x)(1 - f(x)) \quad (5)$$

A Figura 12 ilustra a função sigmoide

Figura 12 - Função Sísmoide



Fonte: Evandro Monteiro Jorge Júnior, 2018.

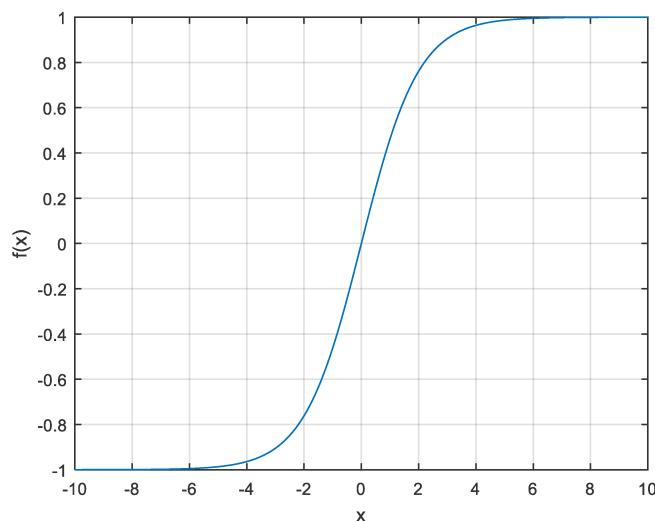
Uma outra função e ativação bastante comum é a bipolar sigmoide ilustrada na Figura 13, que tem alcance de (-1,1) e é definida pela Equação (6):

$$f_2(x) = \frac{2}{1 + \exp(-x)} - 1 \quad (6)$$

Com sua derivada sendo a Equação (7)

$$f'_2(x) = \frac{1}{2} [1 + f_2(x)][1 - f_2(x)] \quad (7)$$

Figura 13 - Função Sísmoide Bipolar



Fonte: Evandro Monteiro Jorge Júnior, 2018.

3.3.5 Algoritmo de treinamento

Nesta seção será apresentada as etapas do algoritmo de treinamento que será utilizado no Capítulo 5.

Passo 1: inicializar os pesos, geralmente valores pequenos, entre -0,5 a 0,5 e aleatórios.

Passo 2: Enquanto o critério de parada for falso, realizar do passo 3 ao 10.

Passo 3: Para cada par de treinamento realizar do passo 4 ao 9.

Feedforward

Passo 4: Cada unidade de entrada (X_i , $i=1, \dots, n$) recebe o sinal de entrada x_i , e transmite este sinal para todas as unidades da próxima camada.

Passo 5: Cada unidade oculta (Z_j , $j=1, \dots, p$) é somado ao sinal de entrada ponderado pelos pesos. Como mostrado na Equação (8)

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (8)$$

É aplicado a função de ativação para calcular o sinal de saída pela Equação (9)

$$z_j = f(z_in_j) \quad (9)$$

E envia este sinal para todas as unidades da próxima camada

Passo 6: Cada unidade se saída (Y_k , $k=1, \dots, m$) é somado ao sinal de entrada ponderado pelos pesos, conforme Equação (10).

$$y_in_k = w_{0k} + \sum_{j=1}^m z_j w_{jk} \quad (10)$$

É aplicado a função de ativação para calcular o sinal de saída como mostra a Equação (11).

$$y_j = f(y_in_k) \quad (11)$$

Retropropagação do erro (*Backpropagation*):

Passo 7: Cada unidade de saída (Y_k , $k=1, \dots, m$) recebe um padrão objetivo correspondente ao padrão de treinamento de entrada, calcula seu erro de informação, com a Equação (12).

$$\delta_k = (t_k - y_k) f'(y_in_k) \quad (12)$$

É calculado os pesos de correção (usados para atualizarem w_{jk} mais tarde), como apresentado na Equação (13)

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (13)$$

É calculado a correção do bias (usado para atualizar o w_{0k} mais tarde) mostrado na Equação (14).

$$\Delta w_{0k} = \alpha \delta_k \quad (14)$$

E envia o δ_k para todas as unidades da camada anterior.

Passo 8: Cada unidade escondida (Z_j , $j=1, \dots, p$) é somada a entrada delta(das unidades da próxima camada), apresentado na Equação (15)

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk} \quad (15)$$

Este valor é multiplicado pela derivada da função ativação e é calculado o erro de informação, conforme Equação (16).

$$\delta_j = \delta_in_j f'(z_in_j) \quad (16)$$

É calculado a correção dos pesos (usado para atualizar v_{ij} mais tarde), na Equação (17).

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (17)$$

E também é calculado a correção do bias, pela Equação (18)

$$\Delta v_{0j} = \alpha \delta_j \quad (18)$$

Atualizar os pesos e os *biases*;

Passo 9: Atualizar os pesos e os bias ($j=0, \dots, p$) para cada unidade de saída ($Y_k, k=1, \dots, m$), como mostrado na Equação (19):

$$w_{jk}(\text{novo}) = w_{jk}(\text{velho}) + \Delta w_{jk} \quad (19)$$

Também são atualizados os pesos e bias ($i=0, \dots, n$) para cada unidade escondida ($Z_j, j=1, \dots, p$), apresentado na Equação (20):

$$v_{ij}(\text{novo}) = v_{ij}(\text{velho}) + \Delta v_{ij} \quad (20)$$

Passo 10: Verificar os critérios de parada.

Na maioria das redes *backpropagation*, o critério de parada é quando o erro quadrático médio é menor ou igual a um valor escolhido, esse valor varia de acordo com o padrão treinado na rede, o erro quadrático médio é dado pela Equação (21)

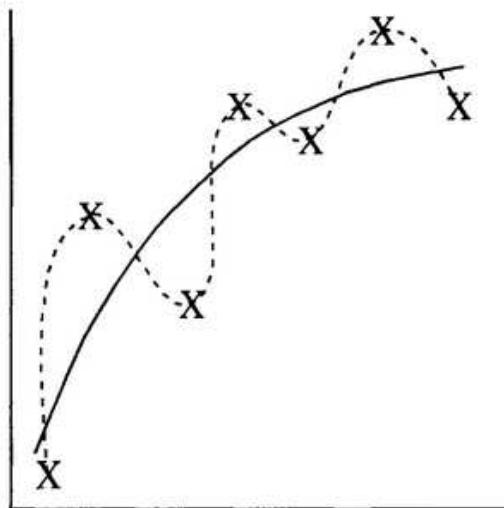
$$E = \frac{1}{2} \sum_p \sum_{i=1}^k (t_i^p - y_i^p)^2 \quad (21)$$

Em que, p é o número de padrões, k é o número de neurônios na saída e t_i^p e y_i^p são os valores desejados(objetivo) e a saída gerada pela rede para o neurônio i quando é apresentado o padrão p.

Dentro do treinamento, cada ciclo, ou seja, do passo 2 ao passo 10 é chamado de época, normalmente, são necessárias várias épocas para o treinamento de uma rede neural *backpropagation*, pois a cada época uma variação é aplicada aos pesos e de época em época essa atualização é acumulada.

Uma vez que a motivação usual para aplicar uma rede *backpropagation* é alcançar um equilíbrio entre respostas corretas aos padrões de treinamento e boas respostas a novos padrões de entrada (isto é, um equilíbrio entre memorização e generalização), não é necessariamente vantajoso continuar treinando até que o erro quadrático total realmente atinja um mínimo (FAUSETT, 1992). A Figura 14 ilustra uma rede neural que atende o equilíbrio entre respostas corretas aos padrões de treinamento e boas respostas a novos padrões de entrada.

Figura 14 – Importância de não treinar a rede até que ela chegue a um mínimo.



Fonte: (CHAUVIN e RUMELHART, 1995)

É sugerido utilizar para o treinamento, dois conjuntos de dados de forma separada, o primeiro conjunto de padrões de treinamento, e o segundo padrão para os testes do treinamento. Os ajustes de peso com base nos padrões de treinamentos, porém, em intervalos durante o

treinamento, utilizar os padrões de teste de treinamento para calcular os erros. Enquanto o erro dos padrões de teste de treinamento diminuir, o treinamento deve continuar, entretanto quando o erro começa a aumentar, a rede estará começando a memorizar os padrões de treinamento de forma específica, ou seja, perdendo a capacidade de generalizar. Neste ponto o treinamento é finalizado (CHAUVIN e RUMELHART, 1995).

3.3.6 Utilizando a RNA treinada

Após a rede treinada, é aplicada somente a fase do *feedforward* do algoritmo de treinamento. O procedimento para utilizar a rede treinada é o seguinte:

Passo 1: Inicializa os pesos (calculados nos algoritmos de treinamento)

Passo 2: Para cada vetor de entrada, realizar dos passos 3 ao 5.

Passo 3: Para $i = 1, \dots, n$, definir a entrada de ativação para cada unidade x_i ;

Passo 4: Para $j=1, \dots, p$, utilizando a Equação (8) e Equação (9)

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (8)$$

$$z_j = f(z_in_j) \quad (9)$$

Passo 5: Para $k=1, \dots, m$, utilizando a Equação (10) e a Equação (11):

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (10)$$

$$y_k = f(y_in_k) \quad (11)$$

3.4 Vantagens e desvantagens da rede neural

Nesta seção serão apresentadas vantagens e desvantagens da utilização das RNAs.

3.4.1 Vantagens:

-A principal vantagem da utilização das RNAs é a capacidade de lidar com ruído na aquisição de sinais e informações incompletas, pois, mesmo com esses fatores indesejáveis ela oferece saídas eficazes.

- As RNAs assim como o cérebro humano apresentam processamento paralelo.
- O aprendizado é feito a partir de informações apresentadas à RNA, devido a isto, ela não precisa ser alimentada com dados antes de iniciar o processo.
- Outra principal característica das RNAs é a capacidade de lidar com padrões e oferecer respostas satisfatórias, mesmo que estes não tenham sido utilizados no treinamento.

3.4.2 Desvantagens das Redes Neurais Artificiais

- As RNAs fazem parte de um campo ligado à computação, e por isto, os utiliza-se computadores para a realização dos cálculos, tornando-se difícil justificar o comportamento das mesmas em algumas situações, com isso é muito comum encontrar em livros que ela são tidas como “caixas pretas”.
- O treinamento efetuado no algoritmo *backpropagation* utiliza de iterações chamadas de épocas que são repetidas até se encontrar o resultado satisfatório e isto pode demandar um tempo muito grande em alguns casos.
- No primeiro item, é citado que as RNAs são tidas como “caixas pretas” e devido a isto torna-se difícil determinar a arquitetura ideal para que ela consiga resolver o problema de modo mais rápido possível.
- Para que a RNA aprenda é necessário apresentar padrões, fazendo com que seja necessário um banco de dados na fase de treinamento.

4 Reflectometria Óptica no Domínio do Tempo (OTDR)

4.1 Introdução

A reflectometria óptica no domínio do tempo, foi demonstrada pela primeira vez por Barnoski e Jensen em 1976 com o objetivo de medir a distribuição de atenuação ao longo de uma fibra óptica (BARNOSKI e JENSEN, 1975).

O equipamento de OTDR, que é ilustrado na Figura 15, nada mais é que um dispositivo que lança pulsos de luz de duração muito curta em uma das extremidades de uma fibra óptica e registra a quantidade de luz que é refletida até o equipamento (WOODWARD e HUSSON, 2005). Analogamente, o OTDR pode ser comparado a um sistema de radar, ambos emitem ondas eletromagnéticas, no caso do OTDR essa onda eletromagnética é em forma de luz, ao

meio e analisam a quantidade e a o tempo necessário para a onda voltar para onde ela foi emitida. A vantagem do OTDR sobre outros equipamentos e medições ópticas é que o OTDR permite que o usuário veja o que acontece com grande qualidade o que ocorre no decorrer do enlace óptico.

Com o OTDR, um par de fibras ou o cabo em teste, ou seja, os enlaces ópticos com várias fibras deixaram de ser uma “caixa preta” (WOODWARD e HUSSON, 2005). A luz refletida de volta em uma fibra óptica é o resultado da reflexão ou retrodifusão. Reflexões são quando a luz que viaja através da fibra óptica encontra mudanças no índice de refração óptico (IOR), essas reflexões são chamadas de reflexões de Fresnel. Retroespalhamento, ou espalhamento de Rayleigh, resulta de variações de composição e densidade distribuídas uniformemente na fibra óptica. Os fôtons são espalhados ao longo do comprimento da fibra óptica e os fôtons que viajam de volta ao OTDR são considerados retroespalhados. Alguns OTDRs contêm até localizador de falhas (WOODWARD e HUSSON, 2005).

Figura 15 - OTDR Modelo Yokogawa AQ7270



Fonte: <https://www.viavisolutions.com/en-us/products/hi-res-multimode-otdr-solution> , acessado em 20/06/2018

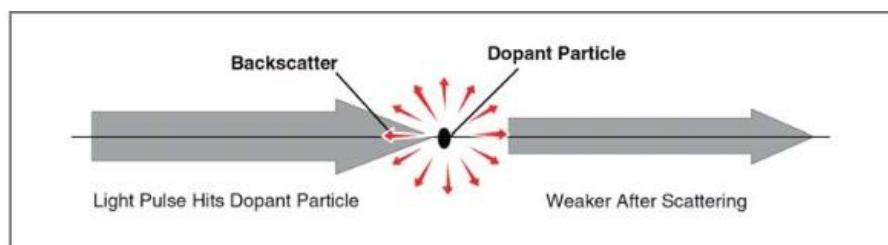
O OTDR é considerado a mais importante ferramenta de investigação para as fibras ópticas, pois é aplicável em medição da perda de fibras, perda de conectores e para a determinação do local exato e o valor das descontinuidades do cabo. Por meio de pulsos muito curtos, também é possível medir a dispersão modal de fibras multimodais. A principal atração da técnica é que ela permite que anomalias, defeitos e outros problemas sejam localizados não destrutivamente.

Pode também ser usado para detectar a distribuição espacial de campos externos que afetam as condições de propagação da luz, como; elétrico, magnético, estresse, tensão e temperatura (ZÓLOMY, 1997). Outra grande vantagem do equipamento de OTDR é a capacidade de fazer a análise somente através de uma extremidade da fibra óptica.

4.1.1 Espalhamento de Rayleigh

Quando um pulso de luz é enviado para dentro uma fibra, parte do pulso se colide com partículas microscópicas (chamadas *dopants*) no vidro e se espalha em todas as direções. Isto é chamado de Espalhamento de Rayleigh. Alguma luz, aproximadamente 0,0001% é espalhada na direção oposta à direção do pulso e é chamado de retroespalhado. Uma vez que os *dopants* são distribuídos uniformemente na fibra óptica devido ao processo de fabricação, o efeito de espalhamento ocorre durante todo seu comprimento (ANRITSU, 2011). A Figura 16 ilustra o espalhamento de Rayleigh.

Figura 16 - Espalhamento de Rayleigh.



Fonte: (ANRITSU, 2011)

O espalhamento de Rayleigh é o principal fator de perdas em fibras. Comprimentos de onda mais maiores apresentam menos dispersão do que comprimentos de onda menores. Por exemplo, a luz a 1550 nm apresenta uma perda de 0,2 a 0,3 dB por quilômetro (dB/ Km) de comprimento de fibra devido ao espalhamento de Rayleigh, enquanto uma luz com comprimento de onda de 850 nm tem uma perda de 4,0 a 6,0 dB/Km de espalhamento (ANRITSU, 2011).

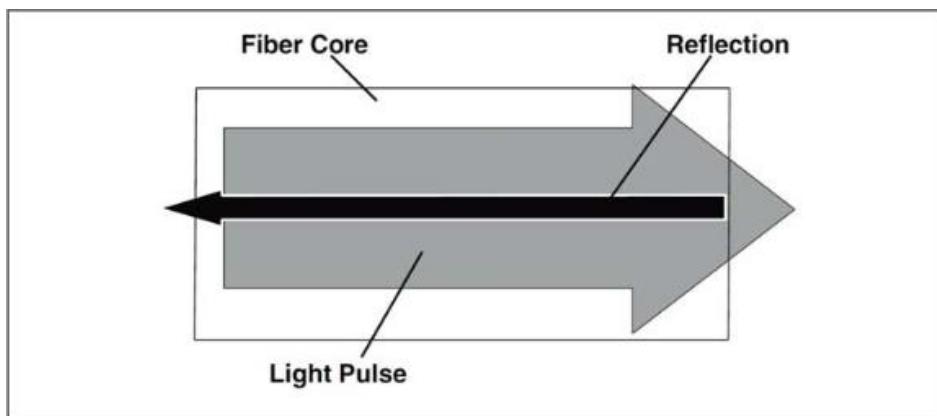
O efeito de espalhamento de Rayleigh é como o brilho de um farol em um nevoeiro à noite: o feixe de luz é difundido - ou espalhado - pelas partículas de umidade. Um nevoeiro espesso dispersará mais da luz porque há mais partículas para obstruí-lo. Você vê o nevoeiro porque as partículas de umidade dispersam pequenas quantidades da luz de volta para você. O feixe de luz pode percorrer um longo caminho se o nevoeiro não for muito espesso, mas, em um denso nevoeiro, a luz é atenuada rapidamente devido a esse efeito de dispersão. As

partículas dopantes na fibra agem como as partículas de umidade do nevoeiro, retornando pequenas quantidades de luz para a fonte à medida que a luz as atinge (ANRITSU, 2011).

4.1.2 Reflexões de Fresnel

Sempre que a luz viaja em um material (como uma fibra óptica) e encontra um material de densidade diferente (como o ar), parte da luz - até 4% - é refletida de volta para a origem da luz enquanto que a outra parte da luz prossegue para o outro meio. Essas mudanças súbitas na densidade ocorrem nas extremidades das fibras, nas rupturas das fibras e, às vezes, nos pontos de emendas. A quantidade da reflexão depende da magnitude da mudança na densidade do material (descrita pelo IOR - maiores IORs significam maiores densidades) e o ângulo em que a luz atinge a interface entre os dois materiais. Este tipo de luz devolvida é chamado de Reflexão de Fresnel. É usado pelo OTDR para determinar com precisão a localização das terminações das fibras. A Figura 17 ilustra uma Reflexões de Fresnel.

Figura 17 - Reflexão de Fresnel.



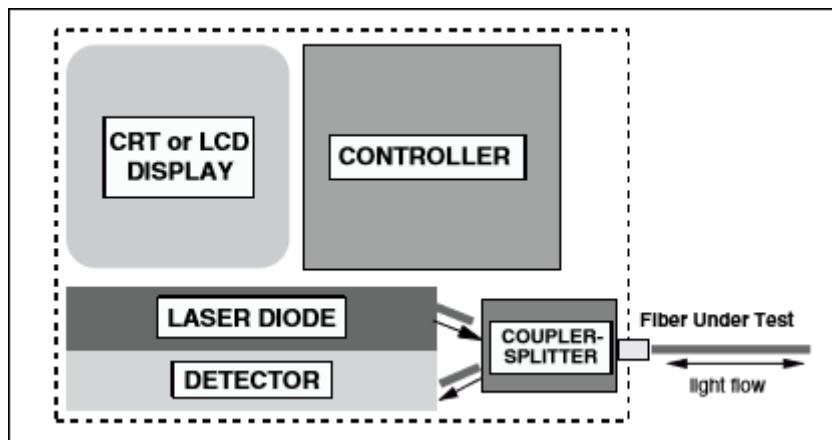
Fonte: (ANRITSU, 2011)

Uma reflexão Fresnel é como brilho de uma lanterna em uma janela. A maior parte da luz passa pela janela, mas parte dela reflete em você. O ângulo em que o feixe de luz atinge a janela determina se o reflexo irá ou não retornar à lanterna, aos seus olhos ou ao teto (ANRITSU, 2011).

4.2 Princípio De Funcionamento do OTDR

O equipamento de OTDR consiste em uma fonte de luz, um sensor óptico, acoplador/divisor, uma *display* e um controlador como ilustra a Figura 18 a seguir

Figura 18 - Diagrama de blocos do OTDR.



Fonte: (ANRITSU, 2011)

4.2.1 Fonte de Luz

O diodo laser do OTDR é um conversor elétrico-óptico (E/O) acionado por um gerador de pulsos. O pulso de luz é acoplado numa fibra em teste via um acoplador direcional óptico. Alguns equipamentos de OTDRs podem possuir mais que um diodo laser permitindo que as fibras possam ser testadas com diferentes comprimentos de ondas, porém somente um laser pode ser usado por vez (ANRITSU, 2011).

4.2.2 Acoplador/Divisor

No acoplador/divisor existem várias portas, uma para cada fonte de luz, uma para a fibra em teste e uma para o sensor. É um dispositivo que permite que a luz viaje para uma direção específica. Da fonte de luz para a fibra e da luz refletida na fibra pelo retroespalhamento e das reflexões de Fresnel para o sensor óptico (ANRITSU, 2011).

4.2.3 Sensor Óptico

O sensor é um fotodetector que mede o nível de luz que reflete da fibra sob teste. Ele converte potência óptica da luz para um nível elétrico correspondente, ou seja, uma potência óptica alta corresponde a um nível elétrico alto. Os sensores no OTDR são projetados especialmente para medir níveis extremamente baixos de luz retroespelhadas. No sensor óptico está incluído um amplificador elétrico para aumentar mais ainda o nível do sinal elétrico.

A potência das reflexões de Fresnel pode ser até 40 000 vezes maior que do retroespelhamento e pode ser maior que o sensor consegue medir, sobrecarregando-o e levando saturação. A saída do nível elétrico é subsequentemente ceifado no nível máximo de saída. Portanto, sempre que um pulso de teste encontra um fim de uma fibra, seja em uma emenda mecânica ou no final da fibra, faz com que o sensor fique “cego” enquanto o pulso ocorrer. Este período cego é conhecido como a "zona morta" (ANRITSU, 2011).

4.2.4 Controlador

O controlador pode ser considerado o cérebro do OTDR. Ele diz ao diodo laser quando enviar o pulso de luz, obtém os níveis de potência do sensor, calcula a distância até a dispersão e reflexão dos pontos na fibra, armazena os pontos de dados individuais e envia as informações para a seção de exibição (ANRITSU, 2011).

Um componente importante da seção do controlador é um circuito relógio muito preciso que é usado para medir com precisão a diferença de tempo entre quando os pulsos de luz são enviados e quando o sensor detecta a luz que retorna ao equipamento. Multiplicando esse tempo de percurso de pulso de ida e volta pela velocidade que a luz se propaga na fibra (que é a velocidade da luz no espaço livre corrigido pelo IOR), é calculada a distância de ida e volta. A distância do OTDR até o ponto (distância unidirecional) é simplesmente a metade da distância de ida e volta (ANRITSU, 2011).

Uma vez que o retroespelhamento ocorre ao longo de toda fibra, há um fluxo contínuo de luz que volta para o OTDR. O controlador faz a amostragem do nível medido pelo sensor em cada intervalo de tempo. Cada nível medido pode ser descrito em uma sequência temporal (que se relaciona com a distância). Como o pulso original se torna mais fraco à medida que se desloca na fibra (devido à perda induzida pelo espalhamento de Rayleigh), o nível de retroespelhamento correspondente tornar-se mais fraco. Portanto, os níveis medidos são

decrescentes considerando o sentido, início-fim de fibra. Porém quando ocorre uma reflexão de Fresnel, o nível de potência naquele ponto, subitamente sobe ao nível máximo, muito acima do nível do retroespalhamento imediatamente anterior a ele (ANRITSU, 2011).

Quando o controlador reune todos os níveis na sequência temporal, ele plota as informações na tela do equipamento. O primeiro nível é mostrado na borda esquerda do gráfico como o ponto inicial da fibra. Sua posição vertical é baseada no nível de potência do sinal retornado: uma potência maior é plotada mais acima no gráfico. Os níveis subsequentes são colocados à direita, um nível a cada intervalo de resolução. O traço resultante é uma linha inclinada que vai do canto superior esquerdo para o canto inferior direito. A inclinação da linha indica seu valor de perda de unidade/distância (dB / km). Variações bruscas significam valores maiores de dB / km. Os níveis de retroespalhamento formam a linha. As reflexões de Fresnel são picos que surgem no decorrer do nível de retroespalhamento. Um deslocamento súbito do nível de retroespalhamento indica um “ponto de perda” que indica a localização de uma emenda térmica (ANRITSU, 2011).

A distância d onde ocorre o evento pode ser calculada pela Equação (22):

$$d = \frac{V_g t}{2} = \frac{ct}{2n_g} \quad (22)$$

Em que:

c =velocidade da luz no vácuo;

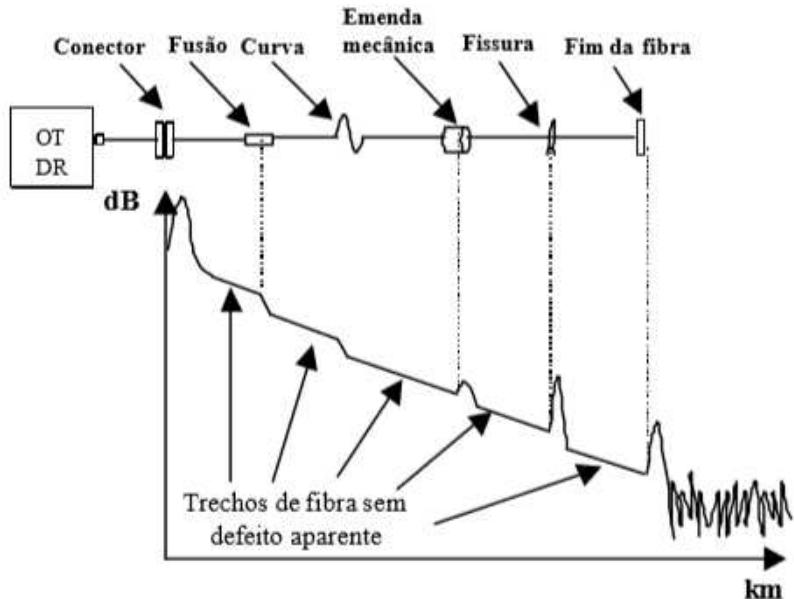
v_g =velocidade de grupo do sinal óptico na fibra;

n_g = IOR de grupo da fibra;

t = intervalo de tempo entre o sinal transmitido e o sinal recebido pelo OTDR (ABBADE, A. L. R. e CAMPOS, M.R., 2002).

A Figura 19 ilustra o processo que ocorre no controlador. O eixo das ordenadas é o nível de potência refletida e o eixo das abscissas, a distância ao ponto de interesse.

Figura 19 - Curva típica de um OTDR



Fonte: (HEWLETT-PACKARD, 1999).

A curva obtida na Figura 19 ilustra o nível de luz retroespalhada e picos de reflexão devido às reflexões de Fresnel em um conector, em uma emenda mecânica e em uma fissura. Aparecem também descontinuidades devido a emenda por fusão ou curvatura acentuada da fibra. É também mensurável a atenuação na fibra ao observar a inclinação da curva. É importante observar que, caso não haja um conector na outra extremidade da fibra em teste, pode não existir o pulso no final da fibra. Nesta situação haveria uma brusca descontinuidade do sinal, evidenciando-se o aparecimento do ruído. Com base neste ruído, pode-se determinar o alcance dinâmico do OTDR.

4.2.5 Display

O *display* é uma tela Tubo de Raios Catódicos (CRT) (em equipamentos mais antigos) ou Display de Cristal Líquido (LCD) que mostra os pontos de dados que compõem o traço da fibra e exibe as condições e medições de configuração do OTDR. A maioria dos monitores OTDR conecta os pontos de dados a uma linha para fornecer uma visão mais clara do traço como um todo. Os cursores podem ser manipulados na tela para selecionar qualquer ponto no

traço da fibra. A distância até o cursor é exibida na tela. Um OTDR com dois cursores exibirá as distâncias de cada cursor e a diferença entre os níveis de retroespalhamento. Pode ser escolhido o tipo de medição que está sendo feita com os cursores, como Perda de 2 pontos, dB / Km, Perda de emenda e Refletância (ANRITSU, 2011).

4.3 Teste com OTDR

Nesta seção será mostrado as etapas do processo da medição de um enlace óptico através do OTDR.

4.3.1 Etapas do teste realizado com OTDR

O primeiro passo em um teste com OTDR é no equipamento preencher alguns campos referentes a fibra a ser testada, como por exemplo, origem, destino, nome do cabo óptico, número da fibra óptica.

O segundo passo é a indicação dos parâmetros ópticos no OTDR, os campos que são comuns entre os OTDRs da maioria das marcas são:

- IOR, que é dado pelo fabricante da fibra óptica, a Tabela 2 abaixo mostra os índices de refração de várias fibras de vários fabricantes.

Tabela 2 - Índices de refração.

Manufacturer	Singlemode Fiber Type	1300nm Singlemode	1550nm Singlemode	Multimode Core Size	850nm Multimode	1300nm Multimode
Alcatel	Normal	1.4660	1.4660	—	N/A	N/A
Lucent	Normal	1.4659	1.4666	50/125	1.4840	1.4790
	D.S.	1.4743	1.4750	62.5/125	1.4960	1.4910
	TrueWave™	n/a	1.4700			
Corning	SMF-21	1.4640	1.4640	50/125	1.4897	1.4856
	SMF-28	1.4675	1.4681	62.5/125	1.5014	1.4966
	D.S.	1.4718	1.4711	100/140	1.4805	1.4748
	LS™	1.4710	1.4700			
	LEAF™	n/a	1.4700			
Sumitomo	Normal	1.4670	1.4670	50/125	1.4840	1.4790
				62.5/125	1.4960	1.4910
<i>Note: D.S. stands for "Dispersion Shifted"</i>						

Fonte: (GN NETTEST, 2000)

-Largura do Pulso a ser disparado pelo equipamento, quanto menor for a largura do pulso mais detalhado será o teste, porém a potência do pulso será menor e dependendo do

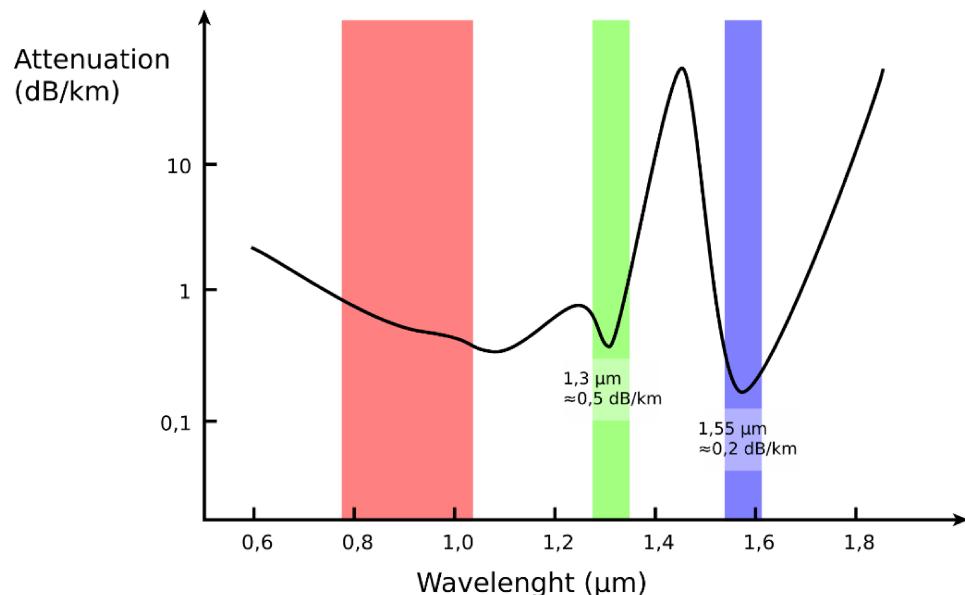
comprimento da fibra óptica informações sobre o final da fibra óptica ficarão comprometidos, já com uma largura de pulso, maior, o pulso possui maior potência, com isso será possível ter informações sobre todo o comprimento da fibra, porém alguns detalhes podem ser perdidos.

- Comprimento da fibra, este valor deve ser ajustado para o valor mais próximo conhecido do comprimento da fibra a ser testada.

- Tempo de medida, esse será o tempo em que o equipamento irá realizar as medidas, ou seja, se for ajustado um tempo de 10 segundos, o equipamento repetirá o teste por esses 10 segundos, e o resultado final será a média dos testes realizados nestes 10 segundos, portanto quanto maior o tempo de medida, mais detalhado será o teste.

- Comprimento de onda, geralmente os testes de fibra óptica são realizados no comprimento de onda de 1550 nm, pois este comprimento é o que a fibra apresenta a menor atenuação que é em torno de 0,2 db/km, como já citado anteriormente. Outros comprimentos de onda também podem ser utilizados, isto dependerá do fim que a fibra será utilizada. A Figura 20 ilustra as janelas de transmissão de uma fibra óptica, em rosa está a primeira janela de transmissão que centra em torno de 850 nm, em verde a segunda janela de transmissão que centra em 1310 nm e a terceira janela, em azul, de transmissão que centra em 1550 nm.

Figura 20 - Janelas de transmissão em fibra óptica



Fonte: https://it.wikipedia.org/wiki/Fibra_ottica, acessado em 30/06/2018

A Figura 21 ilustra o *display* de um equipamento da marca Overtek, nela é possível ver que nele é possível fazer as configurações citadas anteriormente. Neste modelo de equipamento ainda existem mais algumas configurações.

Figura 21 - Configurações no OTDR.



Fonte: (OVERTEK)

O próximo passo será iniciar o teste, que durará aproximadamente o tempo que foi configurado na medida, importante lembrar que além do tempo de medida, existe o tempo para ligar o equipamento ligar o diodo laser, e o tempo para salvar o arquivo após o teste. Após a realização do teste, aparecerá na tela do equipamento o traço de OTDR que será discutido na próxima subseção. Ao final é uma boa prática verificar se os arquivos foram salvos de forma correta.

4.3.2 Traço de OTDR.

Nesta subseção será discutido sobre o traço de OTDR.

Os resultados da medição são mostrados no visor do OTDR. A Figura 22 a seguir ilustra um visor de um equipamento da marca VIAVI, essa figura é semelhante à Figura 19 ,porém a Figura 19 é apenas uma ilustração enquanto a Figura 22 é o próprio *display* do equipamento.

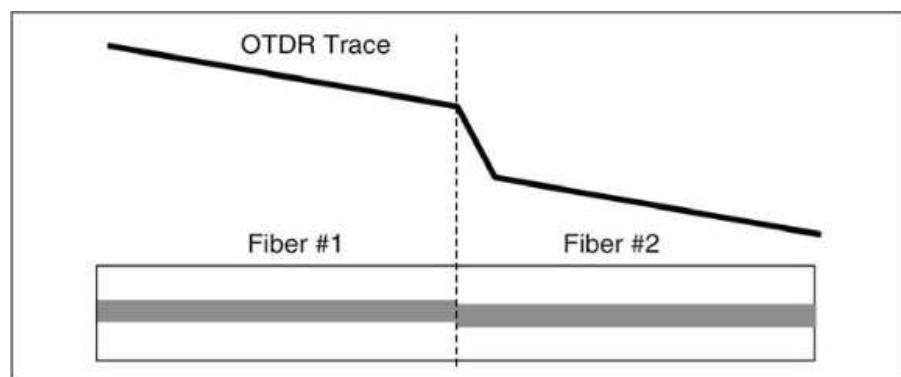
Figura 22 - Display mostrando o resultado de um teste.



Fonte: (VIAVI, 2015)

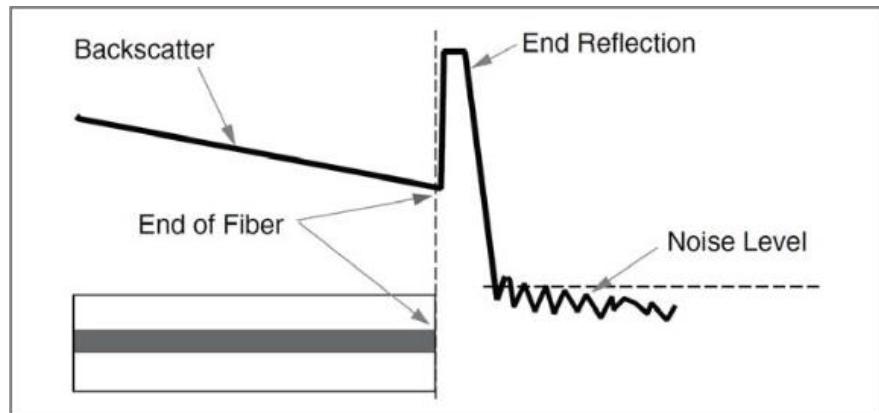
Neste traço podemos observar que a fibra possui 5,5 km e é possível verificar na tela do equipamento que o teste foi realizado com o comprimento de onda de 1310 nm e a largura de pulso de 100 ns. Além disso foram identificados sete eventos. Dentre esses eventos, pelo padrão visto na tela do equipamento e ilustrado na Figura 23, seis são de emenda térmica, e um é um evento de reflexão que indica o fim da fibra e é ilustrado na Figura 24.

Figura 23 - Evento de fusão.



Fonte: (ANRITSU, 2011)

Figura 24 - Evento reflexão no fim da fibra.



Fonte: (ANRITSU, 2011)

4.3.3 Arquivos Standard OTDR Record (SOR)

O formato de dados SOR é usado para armazenar os dados de fibra do reflectômetro óptico no domínio do tempo do OTDR. O formato é definido pelo padrão Telcordia SR-4731, número 2 . Embora seja um padrão, infelizmente não é aberto, pois as especificidades do formato de dados não estão disponíveis abertamente.

Porém existem inúmeros programas em código aberto, em que é possível transformar o arquivo no formato SOR, em um formato CSV, possibilitando abrir no *software* de planilhas como por exemplo, o Excel. Ao abrir este arquivo é possível extrair várias informações dos testes realizados, data e hora da realização, comprimento de onda do laser, comprimento da fibra, largura de pulso, tempo de execução, IOR, número de pontos, distância da resolução e o mais importante, a distância e o nível óptico medido na fibra, esses valores que serão utilizados no decorrer deste trabalho.

4.4 Aplicações do OTDR

Segundo (ANRITSU, 2011) os OTDRs são amplamente utilizados em todas as fases da vida útil de um sistema de fibra, desde a construção até a manutenção, até a localização e restauração de falhas. Um OTDR é usado para:

- Medir a perda total (fim-a-fim) para aceitação e comissionamento do sistema; e para inspeção de entrada e verificação de especificações em bobinas de fibra;
- Medir a perda de emenda térmica e emenda mecânica, durante as operações de instalação,

construção e restauração;

- Medir a refletância ou Perda de Retorno Ótico (ORL) de conectores e emendas mecânicas para CATV, SONET e outros sistemas digitais, analógicos ou de alta velocidade em que as reflexões devem ser controladas;
- Localização de quebras e defeitos na fibra, Indicação do melhor alinhamento óptico das fibras nas operações de emenda;
- Detecção de degradação gradual ou repentina da fibra, fazendo comparações com testes de fibra documentados anteriormente.

4.5 Considerações Finais

Este capítulo teve o propósito de explicar o funcionamento do equipamento de OTDR, nele foi feita uma introdução sobre o mesmo bem como o motivo no qual ele é considerado o equipamento mais importante para a investigação de defeitos em fibras ópticas. Na introdução é descrito o que é o espalhamento de Rayleigh e reflexões de Fresnel que são bases para entender o que acontece em um traço de OTDR.

No capítulo também há uma seção dedicada em explicar o funcionamento do equipamento de OTDR, são explicados o que cada parte contida dentro do equipamento faz, dando mais importância para o controlador que é o “cérebro do equipamento” e onde são gerados os traços que serão utilizados neste trabalho. Logo a seguir foi explicado as etapas da realização de um teste com o OTDR, quais são os parâmetros que são ajustados pré-teste e o resultado do teste.

Além disso é feita uma explicação sobre o formato de arquivo SOR que é o formato de arquivo padrão nos equipamentos de OTDR.

Por fim é citado as aplicações do equipamento.

5 Algoritmo Proposto

5.1 Introdução

No Capítulo 4, foi descrito o funcionamento do equipamento de OTDR e porque ele é um dos equipamentos mais importantes nos testes de fibra óptica, também é mostrado que após o teste é gerado um arquivo com os dados do teste. No Capítulo 3 é descrito como funciona uma RNA, e que ela é muito utilizada em reconhecimento de padrões, também foi descrito modo de treinamento *backpropagation*, que através dele é possível treinar redes multicamadas e resolver problemas de vários tipos incluindo os que não são separáveis linearmente. Também foi visto que embora às vezes uma RNA demande um tempo para ser treinada, após treinada, a utilização dela não é feito de forma bem mais rápida.

Neste capítulo será descrito como foi concebido o algoritmo para detectar eventos em um traço de OTDR utilizando uma RNA, a partir dos padrões que são conhecidos de eventos em um traço de OTDR. O Algoritmo foi desenvolvido no *software* Matlab e envolve a escolha dos padrões a serem treinados, treinamento da rede Neural e o salvar os pesos das redes para serem utilizados em outros traços, que serão descritos no próximo capítulo.

5.2 Matlab

O Matlab é um *software* desenvolvido pela empresa MathWorks Inc. cujo seu nome vem das palavras *MATRIX LABORATORY* ou seja, Laboratório de Matrizes em uma tradução livre. Ele é um *software* interativo de alta performance voltado para o cálculo numérico. Na época que ele foi desenvolvido no fim da década de 1970 as linguagens mais populares eram Fortran, Basic e C, e ao contrário delas, o Matlab permite a resolução de problema numéricos utilizando menos caracteres do que é gasto para escrever nas linguagens citadas, outra grande vantagem é que a solução é apresentada da forma bem semelhante do que é escrito matematicamente.

Nos dias de hoje o Matlab é utilizado em vários ramos da engenharia como civil, telecomunicações, química e controle, pois nele está embutido mais de uma centena de *toolbox* que são plataformas prontos para aplicações.

Como os dados extraídos do equipamento de OTDR são em forma de matrizes em formato CSV como citado no capítulo 2, e a facilidade de gerar gráficos para interpretação dos

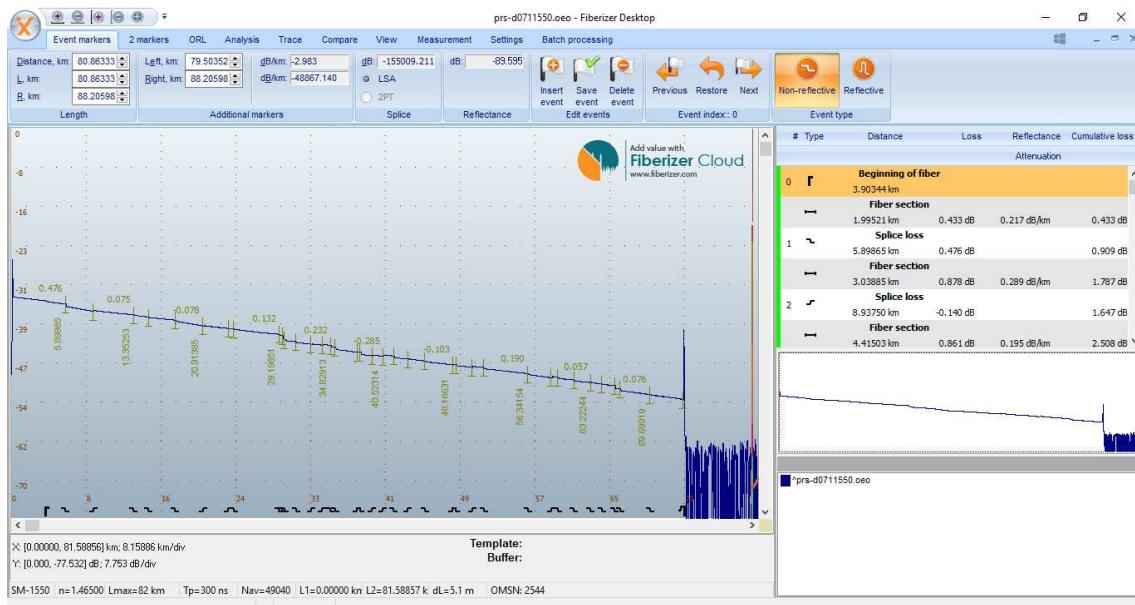
resultados e a facilidade para desenvolver operações com essas matrizes foi escolhido este *software* deste trabalho.

5.3 Fiberizer

O Fiberizer é um *software* para pós-processamento de diferentes tipos de resultados de testes de fibra ótica. Ele permite abrir os arquivos do tipo SOR que são gerados nos OTDRs e visualizar o traço de OTDR ou arquivo de medição de perda de potência óptica na fibra.

Este *software* foi utilizado somente para a leitura dos arquivos de OTDR e conversão em arquivos CSV, que é um formato que pode ser lido por programas como o Excel e o próprio Matlab. A Figura 25 ilustra a tela do *software* Fiberizer

Figura 25 - Tela do *software* Fiberizer



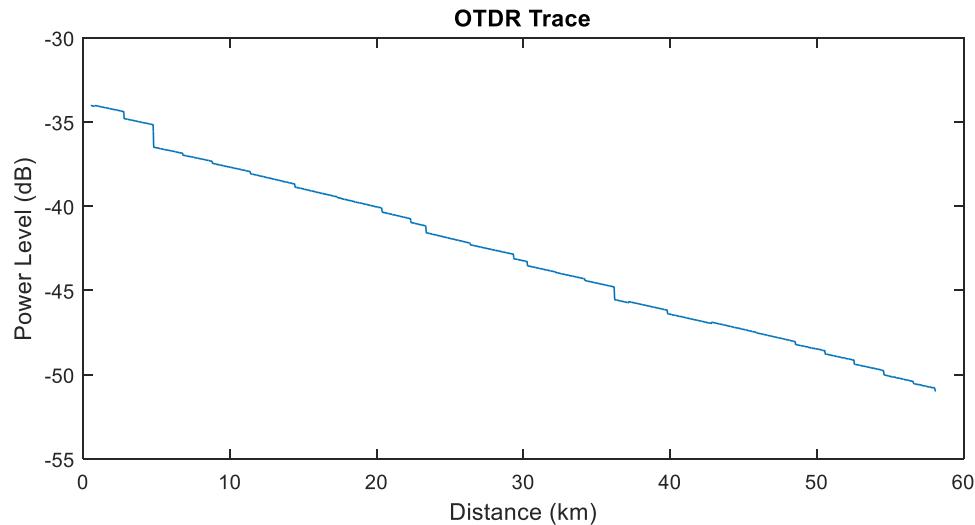
Fonte: Evandro Monteiro Jorge Júnior, 2018.

5.4 Descrição do Algoritmo

Nesta seção será descrito o funcionamento do algoritmo, ou seja, será explicado passo a passo como foi feito o treinamento da rede neural para que ela possa identificar os eventos em um traço de OTDR. Na próxima seção será explicado como foi feito o código no programa Matlab. Os traços de OTDR utilizados nos treinamentos todos foram feitos com o comprimento de onda de 1550 nm, que é mais utilizado na prática devido a atenuação da fibra nessa faixa.

A primeira parte do algoritmo é a importação do traço de OTDR a partir do arquivo CSV, esta é uma tarefa bastante simples pois o *software* Matlab possui funções que importam esse tipo de arquivo e coloca os valores em uma matriz. A Figura 26 ilustra o traço de OTDR importado no *software* Matlab.

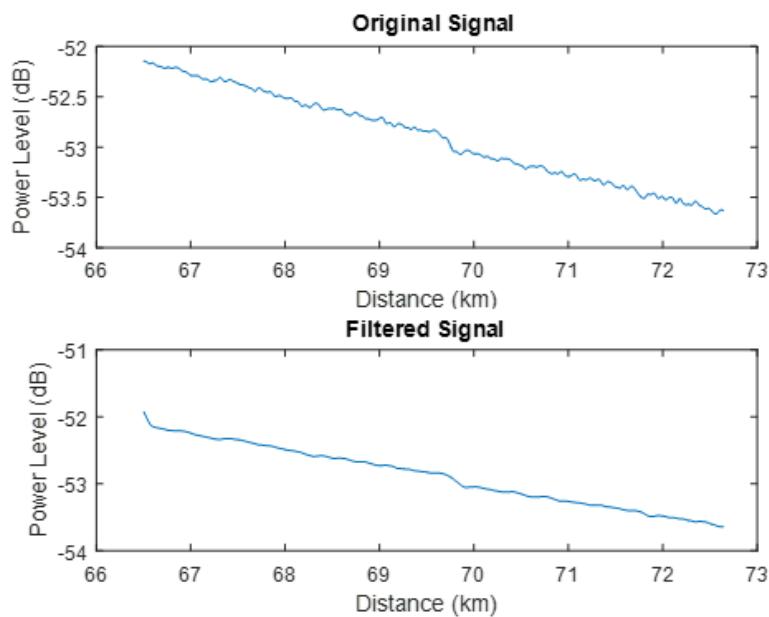
Figura 26 - Traço de OTDR



Fonte: Evandro Monteiro Jorge Júnior, 2018.

Como o sinal obtido pelo OTDR é muito ruidoso, optou-se por aplicar um filtro de média de 20 pontos, que é um filtro do tipo passa-baixo, para eliminar altas frequências. A Figura 27 ilustra o sinal original obtido a partir da medida do OTDR e o sinal filtrado com um filtro de média nela é possível ver claramente a diminuição do nível de ruído no traço de OTDR.

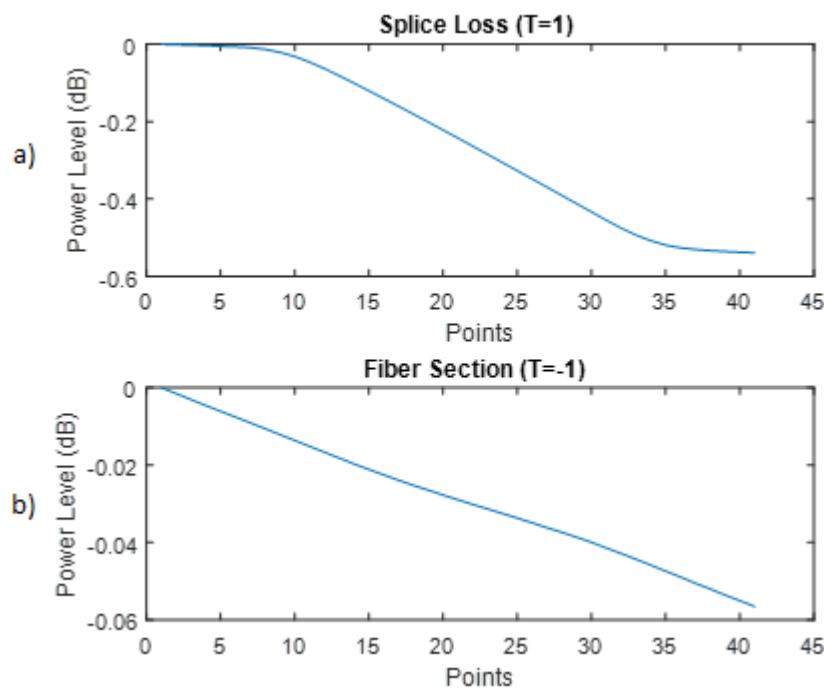
Figura 27 - Sinal original e o sinal filtrado com filtro de média



Fonte: Evandro Monteiro Jorge Júnior, 2018.

A partir dessa curva de OTDR foram escolhidos alguns padrões para o reconhecimento. Como a maioria das redes ópticas possuem apenas eventos do tipo emenda térmica, também chamada de fusão, pois a atenuação neste tipo de emenda óptica é menor do que quando são utilizados conectores, este padrão foi escolhido. Na Figura 28 (a) é ilustrado o vetor que possui 41 valores que é o padrão para uma emenda térmica, também conhecido como fusão, e na Figura 28 (b) é ilustrado o vetor também com 41 valores que é padrão para um seção de fibra.

Figura 28 - Padrões de eventos a serem detectados no traço de OTDR



Fonte: Evandro Monteiro Jorge Júnior, 2018.

Tendo a definição dos padrões de eventos que serão treinados na RNA como base, foi atribuído um valor de objetivo, para os eventos, para o padrão de fusão, foi definido que a saída da rede neural deve ser igual a 1, e para quando é encontrado um padrão seção de fibra, o valor de -1, devido a isto a função de ativação a ser utilizada na RNA é a sigmoide bipolar com amplitude indo de -1 até 1.

Portanto, a partir daí temos os elementos necessários para treinar a rede neural, os vetores de entrada, e os objetivos para cada tipo de evento.

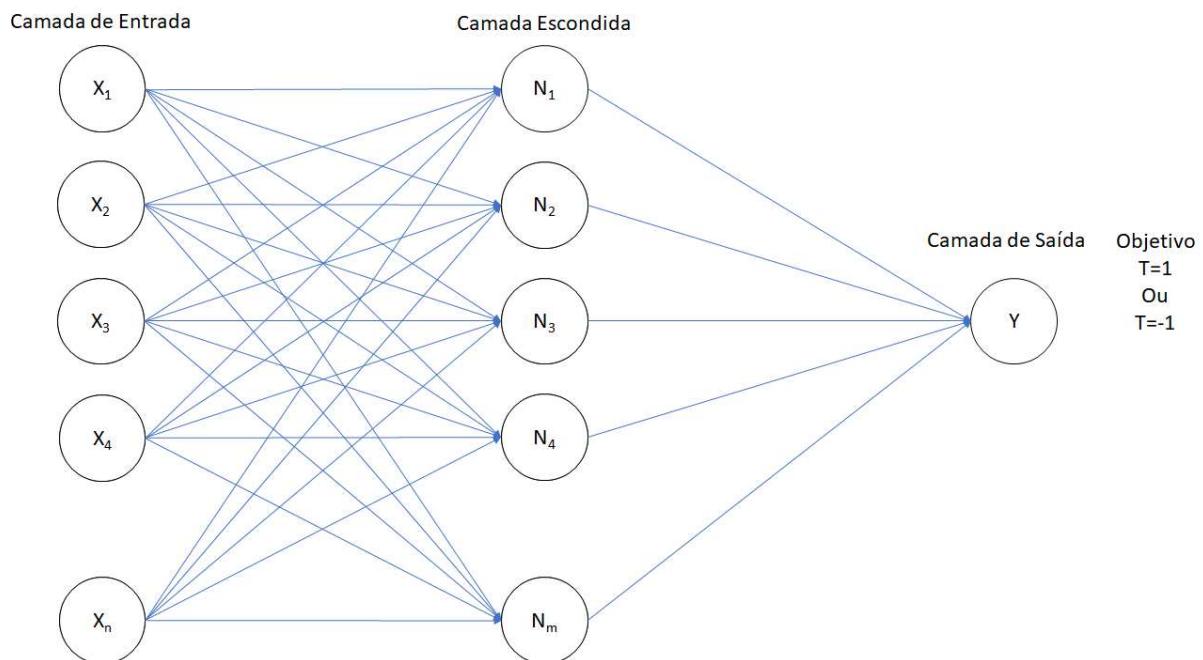
O próximo passo no algoritmo, é definir a RNA. Foi escolhida uma rede multicamadas, pois este tipo de rede consegue resolver problemas no que tange padrões, como citado

anteriormente, que não são linearmente separáveis. O número de camadas escondidas definido foi de apenas uma, pois como é citado no Capítulo 3, esse tipo de rede consegue resolver a maioria dos problemas que envolve RNA e, caso a rede com uma camada escondida não resolvesse, seria adicionada mais camadas escondidas.

No código foi deixado em aberto o número de neurônios na camada escondida, para que possa ver a influência deste valor na velocidade de treinamento.

A Figura 29 ilustra a rede neural utilizada neste trabalho.

Figura 29 - Arquitetura da RNA utilizada nesta dissertação



Fonte: Evandro Monteiro Jorge Júnior, 2018.

O próximo critério a se definir é o critério de parada. Para este trabalho, foi definido o princípio que seria adotado o critério Erro Quadrático Médio entre o valor da saída e o valor do objetivo de cada neurônio na camada de saída. Para este trabalho foi adotado o valor de 0,0001, e caso este valor não fosse adequado, ele seria ajustado até que se encontre um valor melhor. Outro parâmetro a ser definido é a taxa de aprendizagem. Foi escolhido inicialmente o valor de 0,02, porém é foi deixado em aberto para futuras modificações. Pois como cita o Capítulo 3 uma das desvantagens das redes neurais é a dificuldade para ajustar o valor para resolver o problema.

Com a definição de todos os parâmetros, a RNA é treinada, com o mesmo algoritmo de treinamento *backpropagation* descrito no Capítulo 3, ou seja, é feito o *feedforward*, calcula-se o erro, a partir de então entra-se na parte do *backpropagation*, os pesos são ajustados, este ciclo,

chamado de época é realizado até que o critério de parada seja alcançado. Após isto, os valores são salvos para ser utilizado na detecção dos eventos nos traços de OTDR.

Para a utilização da RNA Neural foi criada um novo código fonte. Neste código está um filtro de média para o sinal a ser analisado e a parte da rede neural, porém esta parte contém somente a *feedforward* da rede, pois os pesos já são conhecidos, já que os mesmos foram calculados no outro código. Para saber a localização dos eventos no traço de OTDR foi realizado uma varredura de ponto em ponto com a RNA no traço de OTDR, ou seja, cada parte do traço de OTDR é passado pela rede neural deslocando o vetor a cada ciclo, para que a rede distinguir se é um segmento de fibra óptica ou se há um evento de fusão naquele comprimento da fibra óptica. Após a rede identificar a localização dos eventos de emenda térmica, foi desenvolvido no algoritmo, uma separação desses eventos para que fosse feito a medida da atenuação de cada evento.

5.4.1 Implementação no código no Matlab.

Nesta seção será mostrado o código desenvolvido no Matlab e será explicado o que cada parte do código faz.

O código começa com a leitura do arquivo que contém os dados do traço de OTDR.

```
entrada=xlsread('traco.csv');
pos=entrada(:,1);
aten=entrada(:,2);
```

A variável ‘entrada’ recebeu a matriz através da função `xlsread()` que contém a potência e a posição de cada ponto obtido pelo equipamento de OTDR, a seguir foi utilizada mais duas variáveis, ‘pos’ e ‘aten’, em que cada coluna da matriz foi transformada em um vetor com a posição em que foi obtido o ponto, e o valor da potência naquele ponto.

O próximo passo é preparar o vetor com os valores da potência para realizar a filtragem de média.

```
aten1=zeros([1 tfiltro/2]);
[alntrada,lntrada]=size(entrada);
tfiltro=20;
for i=tfiltro/2+1:alntrada+tfiltro/2
    atenf(i-tfiltro/2)=mean(aten1(i-tfiltro/2:i+tfiltro/2));
end
```

A variável `aten1` recebe o vetor preparado para realizar o filtro de média, a variável ‘`tfiltro`’ corresponde qual o número de pontos que serão realizados a média e a iteração ‘`for`’

realiza o calculo da média dos pontos em ‘tfiltro’ em todo o vetor e atribui a saída à variável ‘atenf’. A função ‘size’ calcula o tamanho dos vetores para ser utilizado na iteração ‘for’.

O próximo passo, foi extrair alguns padrões de fusão e de segmento de fibra, atribuir a uma variável para que eles possam ser utilizados no treinamento.

```
x(1,:)=atenf(1135:1175);
x(1,:)=x(1,:)-min(x(1,:));
x(1,:)=x(1,:)-max(x(1,:))/2;
x(2,:)=atenf(1005:1045);
x(2,:)=x(2,:)-min(x(2,:));
x(2,:)=x(2,:)-max(x(2,:))/2;
```

Foi atribuída na linha 1 da variável ‘x’ um padrão de fusão que existe entre as posições no vetor ‘atenf’ 1135 à 1175. Da mesma maneira foi atribuída na linha 2 da variável ‘x’ um padrão de segmento de fibra entre as posições 1005 à 1045 no vetor ‘atenf’. As duas linhas abaixo de cada variável x, serve para deslocar os valores dos vetores em torno do eixo x, ou seja, é eliminado o *offset* de cada vetor caso ele possua.

```
ne=10; %numero de neuronios na camada oculta
t=[1 -1]; % objetivo da rede neural
[xl,xa]=size(x);
alfa=0.02;
%inicialização dos pesos
b1=rand([1 ni])-0.5;
v=rand([ni xa])-0.5;
w=rand([1 ni])-0.5;
b2=rand()-0.5;
erro=100;
k=1;
epocas=0;
```

Nesta parte do código é definido o número de neurônios na camada oculta através da variável ‘ne’, os objetivos da rede neural através da variável ‘t’, a taxa de aprendizado cujo valor está na variável ‘alfa’ e a inicialização dos pesos e *bias* através da função rand, que atribui um número aleatório entre 0 e 1 que é subtraído de 0,5, para os pesos iniciais serem aleatórios entre 0 e 1. As outras variáveis são apenas a inicialização delas para serem utilizadas posteriormente.

```
while erro>0.0001
    epochas=epochas+1;
    for j=1:xl
        xi=x(j,:);
        ti=t(j);

        %feedforward
        for i=1:ne
            zin(i)= b1(i)+sum(xi.*v(i,:));
        end
        z=fativ(zin);

        yin=b2+sum(z.*w);
        y=fativ(yin);
        e(j)=0.5*(ti-y)^2;
```

Com a definição das variáveis é colocado uma condição, para o código se repetir até que o critério de parada seja atingido. É também colocado um contador de épocas através da variável ‘época’. Essa parte do código apresenta a primeira parte do algoritmo que é o *feedforward* que foi explicado no capítulo 3. Nesta parte do código existe uma função chamada de ‘fativ()’, esta função criada caso seja necessário substituir a função de ativação, que é a função sigmoide bipolar, também foi criada a função ‘dfativ()’ com a derivada da função de ativação pelo mesmo motivo que a função ‘fativ()’.

```
%backpropagation of error
deltak=(ti-y)*dfativ(yin);
deltaw=alfa*deltak*z;
deltab2=alfa*deltak;

deltain=sum(deltak*w);
deltaj=deltain*dfativ(zin);
for i=1:xa
deltav(:,i)=alfa*deltaj.*xi(i);
end

deltab1=alfa*deltaj;

w=w+deltaw;
v=v+deltav;
b1=b1+deltab1;
b2=b2+deltab2;
end
erro=sum(e)
Et(k)=sum(e);
k=k+1;
end
```

Esta parte do código é o treinamento *backpropagation*, que também foi utilizado conforme explicado no capítulo 3. Importante destacar que nesta parte do código os pesos são atualizados para serem testados na próxima iteração.

```
save('w.dat','w');
save('v.dat','v');
save('b1.dat','b1');
save('b2.dat','b2');
save('ni.dat','ne');
```

Esta parte do código serve para guardar os valores dos pesos que foram ajustados no treinamento *backpropagation* para que seja utilizado depois para detectar os eventos em curvas de OTDR.

Um outro código em um outro arquivo foi criado para a utilização da rede neural e de seus pesos ajustados.

```
entrada=xlsread('traco2.csv');
pos=entrada(:,1);
aten=entrada(:,2);
[aentrada,lentrada]=size(entrada);
tfiltro=20;
aten1=zeros([1 tfiltro/2]) aten' zeros([1 tfiltro/2]);
for i=tfiltro/2+1:aentrada+tfiltro/2
    atenf(i-tfiltro/2)=mean(aten1(i-tfiltro/2:i+tfiltro/2));
end
```

O código se inicia da mesma forma que o código acima com o filtro de média na curva a ser analisada.

```
for i=1:aentrada-tfiltro
    desvio(i)=std(atenf(i:i+tfiltro));
end
for i=50:aentrada

    if mean(desvio(i:i+10))>1
        top=i;
        break;
    end
end
dados=atenf(1:top+tfiltro);
```

Foi implementado também no algoritmo uma função baseada no desvio padrão da curva para eliminar a parte da zona de ruído no fim do traço de OTDR. Após a eliminação da parte da zona de ruído, foi atribuída a variável ‘dados’ que são os dados que serão analisados pela rede neural.

```
load('w.dat','-mat');
load('v.dat','-mat');
load('b1.dat','-mat');
load('b2.dat','-mat');
load('ni.dat','-mat');
```

Nesta parte do código são carregadas as funções salvas no código anterior através da função load().

```
for j=1:top+20

xi=dados(j:j+40);
```

```
%xi=xi';
xi=xi-min(xi);
xi=xi-max(xi)/2;

%feedforward
for i=1:ne
    zin(i)= b1(i)+sum(xi.*v(i,:));
end
z=fativ(zin);

yin=b2+sum(z.*w);
saida(j)=fativ(yin);

end
```

Com os pesos carregados é só realizar o *feedforward* da rede neural, pois os pesos, *biases* e números de neurônios na camada escondida foram carregados no Matlab, nesta parte do código também, o primeiro ‘for’ é responsável por deslocar o traço de OTDR que está na variável dados, cada iteração do for, é deslocado um valor do vetor e é verificado através da rede neural se aquela parte é um segmento de fibra ou uma fusão.

```
figure
subplot(3,1,1)
grid on
plot(pos(1:top+20),aten(1:top+20))
title('Sinal Original')
xlabel('Distance(km)')
ylabel('Power Level(dB)')
subplot(3,1,2)
plot(pos(1:top+20),dados(1:top+20));
title('Sinal Filtrado')
xlabel('Distance(km)')
ylabel('Power Level(dB)')
subplot(3,1,3)
plot(pos(1:top+20),saida);
title('Eventos')
xlabel('Distance(km)')
ylabel('Eventos')
```

Esta parte do código server para plotar no mesmo gráfico, o sinal origina, o sinal filtrado e a localização dos eventos pela RNA.

```

for i=1:lsaida
    if saida(i)>-.98
        result(i+tfiltro)=dados(i+tfiltro);
    else
        result(i+tfiltro)=0;
    end
end
for i=2:lsaida
    if result(i)~=0 && result(i-1)==0;
        evento(a,1)=i;
    end
    if result(i)==0 && result(i-1)~=0
        evento(a,2)=i;
        a=a+1;
    end
end
evento(a,2)=lsaida;
[aevento,levento]=size(evento);
for i=1:aevento
    if evento(i,1)~=0
        valor(i,2)=mean(entrada(evento(i,1)-2:evento(i,1),2))-mean(entrada(evento(i,2):evento(i,2),2));
    else
        valor(i,2)=0;
    end
    valor(i,1)=pos(floor(mean(evento(i,:))));

end

f =figure;
table=uitable(f);
table.Data=valor;
table.ColumnName = {'Distância (km)', 'Atenuação (dB)'};

```

Nesta parte do código foi utilizada para separar a parte em que a RNA detectou o segmento de fibra óptica da parte em que a rede neural detectou um evento de fusão. As fusões

foram colocadas em uma variável chamada ‘evento’ e foi feito o cálculo de seu valor, no término é apresentada uma tabela com os valores das atenuações detectadas no traço de OTDR.

5.5 Considerações finais

Neste Capítulo foi apresentado o programa desenvolvido para treinar uma RNA com os padrões de segmento de fibra e de atenuação, e também o programa para efetuar a utilização da rede neural treinada. Em um primeiro momento é descrito detalhadamente. Após isto é apresentado o código fonte que foi escrito no Matlab, e também foi comentado as operações que cada parte do código realiza.

Com a apresentação do algoritmo proposto o próximo mostrará testes realizados a partir da RNA treinada.

6 Resultados

6.1 Introdução

No capítulo anterior foi desenvolvido o algoritmo para detecção de eventos de fusão em um traço de OTDR, também foi mostrada como a rede é treinada e como faz-se a utilização da rede para a análise dos traços.

Neste capítulo será apresentado a relação entre o número de neurônios na camada escondida, o número de épocas e o tempo de processamento. Além disso serão apresentados cinco casos em que o algoritmo foi utilizado para análise de traços de OTDR.

6.2 Impacto da mudança no número de neurônios na camada escondida.

O tempo de processamento na computação é algo muito importante, e pode até tornar inviável um algoritmo caso o mesmo seja elevado. Nesta seção será apresentado a relação entre essas três variáveis: neurônios na camada escondida, número de épocas e tempo de processamento. Como cita o capítulo 3, o algoritmo de *backpropagation*, tem a desvantagem de às vezes demorar para encontrar o resultado ideal.

O número de neurônios na camada escondida, foi variado e anotado os dados sobre o número de épocas e o tempo de treinamento. Foram realizados cinco treinamentos com 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 neurônios na camada escondida e foram registrados na Tabela 3 a relação entre o número de neurônios na camada oculta com o número de épocas e o tempo gasto para o treinamento.

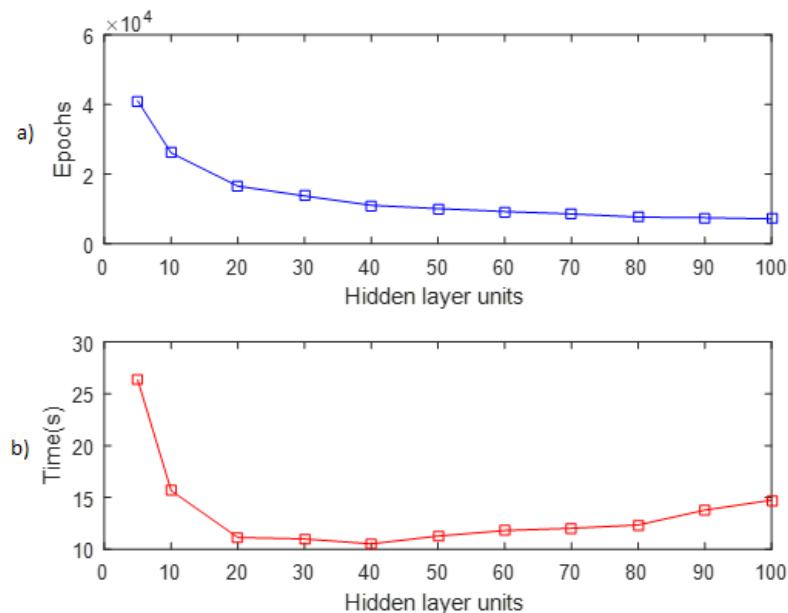
Tabela 3 - Relação entre o número de neurônios na camada escondida, número de épocas e tempo de treinamento.

Nº de neuronios camada escondida	Número de épocas/ Tempo de Treinamento											
	Treinamento 1		Treinamento 2		Treinamento 3		Treinamento 4		Treinamento 5		Média	
	Épocas	Tempo(s)	Épocas	Tempo(s)	Épocas	Tempo(s)	Épocas	Tempo(s)	Épocas	Tempo(s)	Épocas	Tempo(s)
5	40855	26,817	40405	30,672	41590	21,474	41044	26,791	41217	25,955	41022,2	26,342
10	25819	13,901	26069	19,205	26312	16,018	25306	15,416	26818	13,727	26064,8	15,653
20	16876	10,909	16722	10,657	16618	10,481	16067	10,862	16348	12,760	16526,2	11,134
30	15120	11,648	13125	10,941	13577	10,808	13231	11,080	13487	10,506	13708,0	10,997
40	10950	10,469	11539	10,825	10918	10,350	10855	10,662	10842	10,300	11020,8	10,521
50	9996	11,029	10341	11,105	9363	10,720	9461	10,834	11131	12,660	10058,4	11,270
60	9414	12,266	9151	11,409	9930	11,838	8723	12,308	8974	11,186	9238,4	11,801
70	8545	12,009	8917	12,383	7653	11,142	8199	11,955	9412	12,657	8545,2	12,029
80	7213	12,029	7494	12,063	7492	12,539	8216	12,606	7783	12,449	7639,6	12,337
90	6532	11,965	6887	12,355	6936	12,337	6799	12,218	7035	12,638	6837,8	12,303
100	7087	14,630	7752	15,709	6772	14,341	7071	14,747	6829	14,239	7102,2	14,733

Fonte: Evandro Monteiro Jorge Júnior, 2018.

Para uma melhor interpretação da Tabela 3 foi colocada uma escala de cores na parte da tabela que são mostrados os valores médios de épocas e o tempo de treinamento, também desenvolvido o gráfico em que o Figura 30 ‘a’ apresenta a relação entre o número de neurônios na camada escondida e o número de épocas no treinamento, enquanto que na Figura 30 ‘b’ é apresentado o número de neurônios na camada escondida em relação ao tempo de treinamento.

Figura 30 - Gráfico entre a comparação do número de épocas, tempo de treinamento e número de camadas escondidas.



Fonte: Evandro Monteiro Jorge Júnior, 2018.

Podemos perceber através da Tabela 3 e das Figura 30a e Figura 30b que ao aumentar o número de neurônios na camada escondida, o número de épocas para atingir o Erro Quadrático

Médio Total igual a 0,0001, diminui consideravelmente, porém quando coloca-se mais neurônios na camada escondida, aumenta o número de operações, ou seja, exige mais tempo de processamento do computador no programa, aumentando o tempo de treinamento.

6.3 Utilizando da rede neural treinada em traços de OTDR

Nesta seção serão mostrados os resultados da rede neural treinada. Utilizou-se um banco de dados com testes realizados em campo e em laboratório. O resultado apresenta três gráficos, o primeiro gráfico é o sinal original, sem nenhum tipo de pré-processamento, o segundo gráfico é o sinal filtrado pelo filtro de média e o terceiro gráfico é o resultado da rede neural, quando o traço de OTDR é uma seção da fibra, ele apresenta o valor igual -1 conforme treinamento apresentado no capítulo anterior e quando ele identifica um evento de emenda térmica o eixo das ordenadas assume o valor igual a 1.

Em suma, quando o traço de OTDR está em uma seção de fibra o eixo das ordenadas ficará em -1, quando a RNA encontrar um evento de emenda térmica, a ordenada irá para o valor de 1 durante o instante que estive ‘varrendo’ o evento, e depois ele volta para o valor -1 na próxima seção de fibra, porém nem sempre os valores dos eventos atingiram o valor igual a 1, esse valor será discutido no próximo Capítulo, Conclusões e Contribuições. Também foi feita uma tabela com a localização dos eventos de emenda térmica bem com o valor da atenuação. Logo abaixo dos gráficos será apresentada a tabela que mostra a distância dos eventos de emenda térmica localizado bem como o valor de atenuação daquela emenda.

A rede neural foi treinada com alguns padrões de treinamento conforme Figura 28, ou seja, foram ajustados os pesos da rede neural para esses padrões de treinamento. Esses pesos ajustados foram utilizados em alguns casos reais de traço de OTDR, ou seja, só foi realizado um treinamento.

-Caso 1

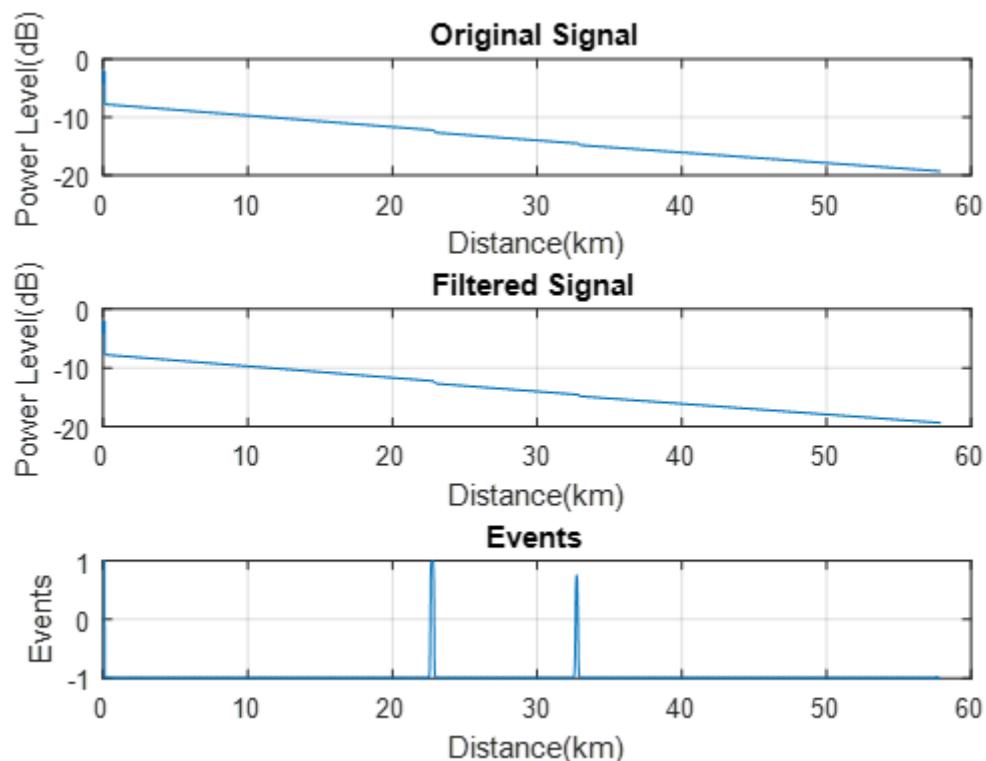
Equipamento: MTP-6000

Fabricante: Optronik

Local do teste: Laboratório

A Figura 31 mostra os resultados obtidos no teste, a partir da utilização da RNA treinada no Caso 1. A Tabela 4 apresenta os eventos encontrados a partir do teste do Caso 1.

Figura 31 - Resultado dos testes do Caso 1 utilizando a RNA treinada.



Fonte: Evandro Monteiro Jorge Júnior, 2018.

Tabela 4 - Tabela de eventos encontrados a partir do teste do Caso 1

No. do evento	Distância(km)	Atenuação (dB)
1	22.820	0.452
2	32.812	0.325

- Caso 2

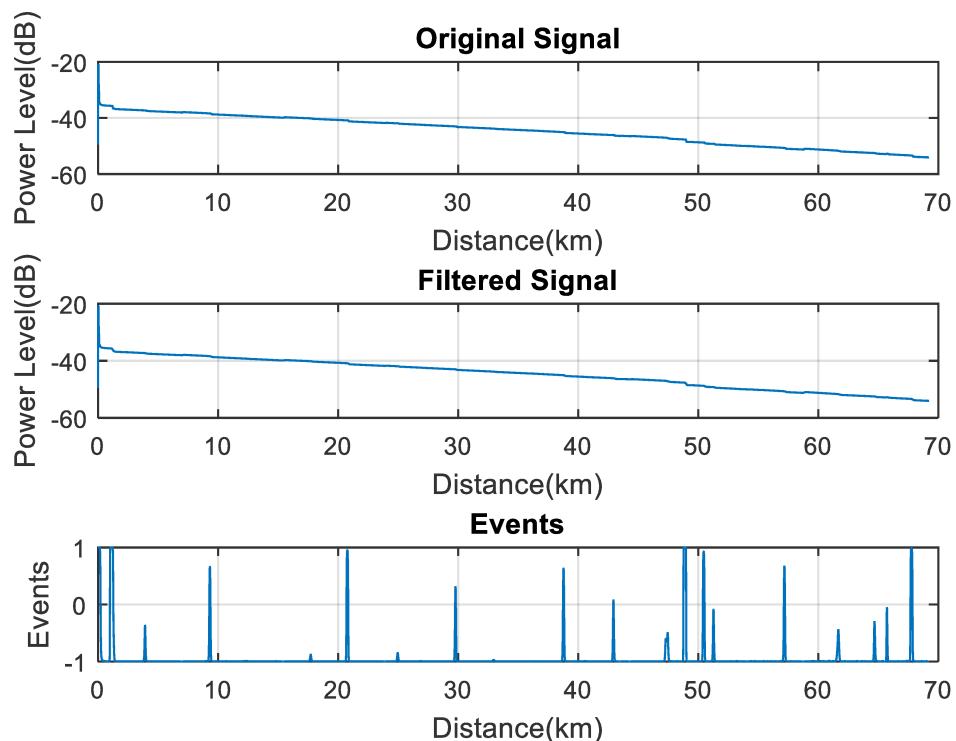
Equipamento: MTS 8000

Fabricante: JDSU/VIAVI

Local do teste, início: Ibiá/MG, fim: Campos Altos/MG

A Figura 32 mostra os resultados obtidos no teste, a partir da utilização da RNA treinada no Caso 2. A Tabela 5 apresenta os eventos encontrados a partir do teste do Caso 2.

Figura 32 - Resultado dos testes do Caso 2 utilizando a RNA treinada.



Fonte: Evandro Monteiro Jorge Júnior, 2018.

Tabela 5 - Tabela de eventos encontrados a partir do teste do Caso 2

No. do evento	Distância(km)	Atenuação (dB)	No. do evento	Distância(km)	Atenuação (dB)
1	0.343	0.664	11	43.091	0.256
2	1.330	1.178	12	47.537	0.400
3	4.067	0.214	13	49.031	0.868
4	9.470	0.289	14	50.612	0.371
5	17.870	0.151	15	51.430	0.242
6	20.924	0.379	16	57.334	0.306
7	25.119	0.155	17	61.805	0.334
8	29.928	0.265	18	64.854	0.226
9	33.105	0.077	19	65.893	0.184
10	38.937	0.301	20	67.929	0.441

- Caso 3

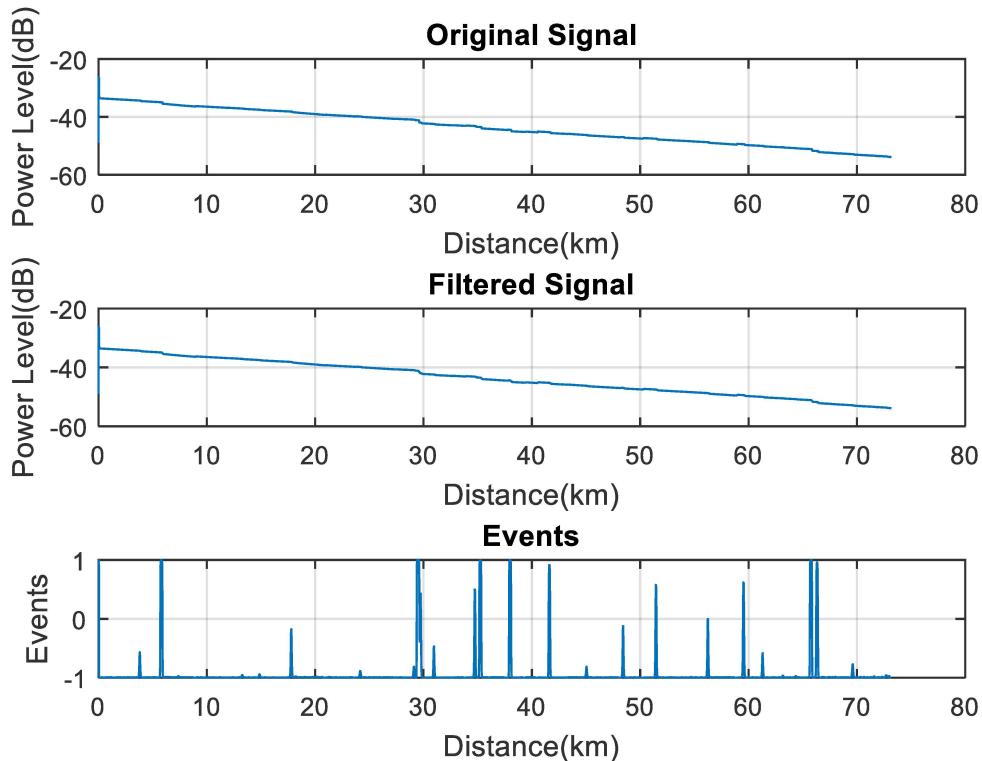
Equipamento: MTS 8000

Fabricante: JDSU/VIAVI

Local do teste, início: Pará de Minas/MG, fim: Divinópolis/MG

A Figura 33 mostra os resultados obtidos no teste, a partir da utilização da RNA treinada no Caso 3. A Tabela 5 apresenta os eventos encontrados a partir do teste do Caso 3.

Figura 33 - Resultado dos testes do Caso 3 utilizando a RNA treinada.



Fonte: Evandro Monteiro Jorge Júnior, 2018.

Tabela 6 - Tabela de eventos encontrados a partir do teste do Caso 3

No. do evento	Distância(km)	Atenuação (dB)	No. do evento	Distância(km)	Atenuação (dB)
1	0.179	0.004	16	41.771	0.358
2	3.965	0.211	17	45.199	0.162
3	5.965	0.563	18	48.576	0.241
4	7.531	0.066	19	51.609	0.292
5	13.424	0.096	20	56.393	0.248
6	15.020	0.115	21	59.693	0.305
7	17.952	0.253	22	61.452	0.219
8	18.356	0.017	23	63.325	0.019
9	24.321	0.161	24	64.512	0.059
10	29.273	0.170	25	65.908	0.600
11	29.744	1.094	26	66.481	0.414
12	31.115	0.208	27	69.761	0.181
13	34.896	0.294	28	71.7558	0.0517
14	35.397	0.521	29	72.866	0.101
15	38.149	0.550	30	73.117	0.069

- Caso 4

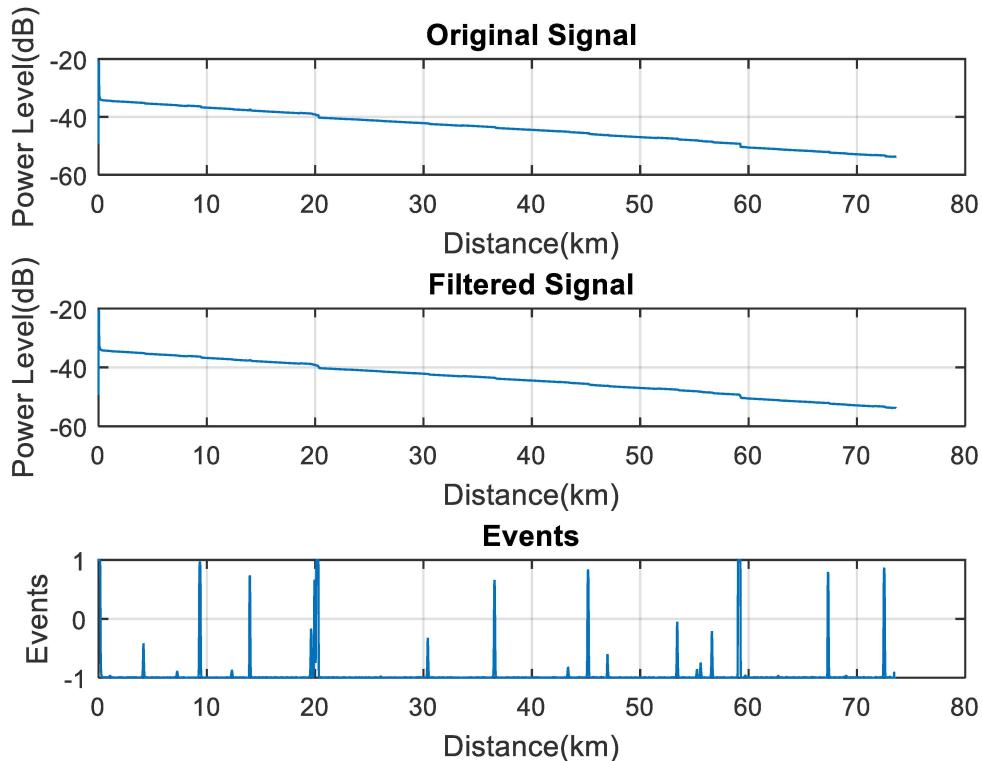
Equipamento: MTS 8000

Fabricante: JDSU/VIAVI

Local do teste, início: Santo Antônio do Monte/MG, fim: Iguatama/MG

A Figura 34 mostra os resultados obtidos no teste, a partir da utilização da RNA treinada no Caso 4. A Tabela 7 apresenta os eventos encontrados a partir do teste do Caso 4.

Figura 34 - Resultado dos testes do Caso 4 utilizando a RNA treinada.



Fonte: Evandro Monteiro Jorge Júnior, 2018.

Tabela 7 - Tabela de eventos encontrados a partir do teste do Caso 4

No. do evento	Distância(km)	Atenuação (dB)	No. do evento	Distância(km)	Atenuação (dB)
1	0.312	0.541	14	47.148	0.186
2	1.223	0.080	15	53.569	0.247
3	4.303	0.207	16	55.385	0.158
4	7.418	0.139	17	55.743	0.181
5	9.516	0.395	18	56.771	0.241
6	12.468	0.150	19	59.293	1.164
7	14.140	0.320	20	59.861	0.094
8	19.773	0.244	21	62.895	0.075
9	20.244	1.163	22	67.510	0.328
10	30.547	0.223	23	69.147	0.090
11	36.707	0.308	24	72.677	0.300
12	43.490	0.160	25	73.582	0.019
13	45.342	0.325			

- Caso 5

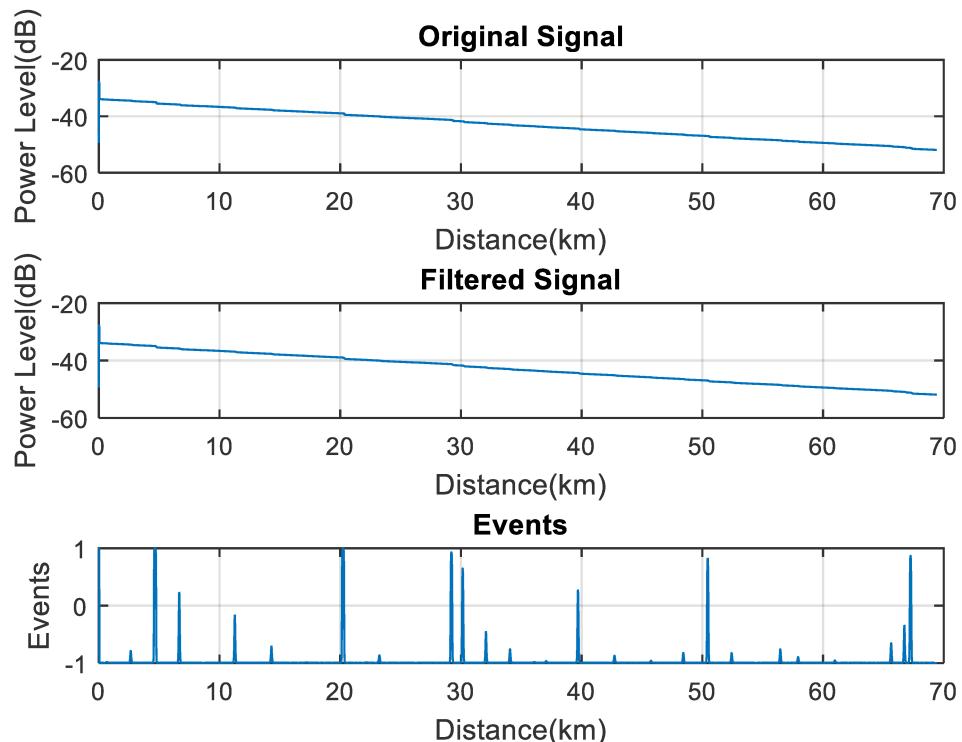
Equipamento: MTS 8000

Fabricante: JDSU/VIAVI

Local do teste, início: Trindade do Sul/RS, fim: Seberi/RS

A Figura 35 mostra os resultados obtidos no teste, a partir da utilização da RNA treinada no Caso 5. A Tabela 8 apresenta os eventos encontrados a partir do teste do Caso 5.

Figura 35 - Resultado dos testes do Caso 5 utilizando a RNA treinada



Fonte: Evandro Monteiro Jorge Júnior, 2018.

Tabela 8 - Tabela de eventos encontrados a partir do teste do Caso 5

No. do evento	Distância(km)	Atenuação (dB)	No. do evento	Distância(km)	Atenuação (dB)
1	0.179	0.012	14	39.848	0.262
2	2.798	0.167	15	42.877	0.150
3	4.809	0.534	16	45.895	0.096
4	6.814	0.266	17	48.581	0.165
5	11.419	0.235	18	50.596	0.318
6	14.453	0.179	19	52.581	0.159
7	20.387	0.511	20	56.613	0.172
8	23.395	0.155	21	58.086	0.137
9	29.365	0.369	22	61.130	0.105
10	30.302	0.298	23	65.796	0.210
11	32.225	0.223	24	66.885	0.220
12	34.220	0.177	25	67.402	0.349
13	37.213	0.074			

- Caso 6

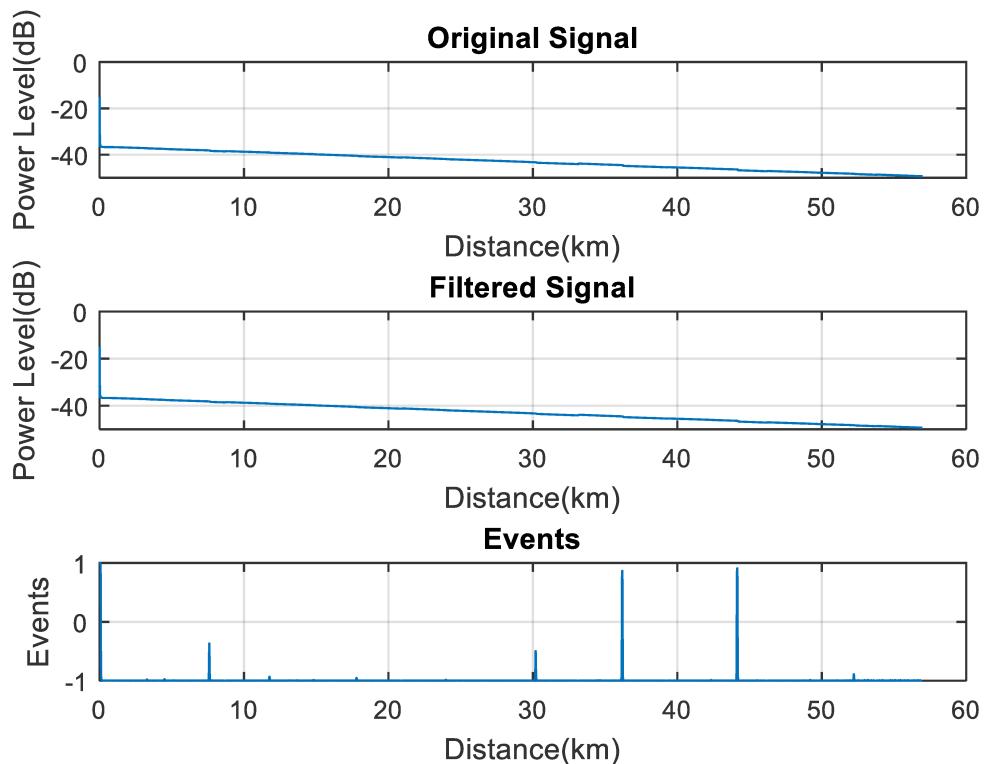
Equipamento: MTS 8000

Fabricante: JDSU/VIAVI

Local do teste, início: Bambuí/MG, fim: Iguatama/MG

A Figura 32 mostra os resultados obtidos no teste, a partir da utilização da RNA treinada no Caso 6. A Tabela 5 apresenta os eventos encontrados a partir do teste do Caso 6.

Figura 36 - Resultado dos testes do Caso 6 utilizando a RNA treinada



Fonte: Evandro Monteiro Jorge Júnior, 2018.

Tabela 9 - Tabela de eventos encontrados a partir do teste do Caso 6

No. do evento	Distância(km)	Atenuação (dB)
1	0.146	0.554
2	3.366	0.023
3	4.579	0.043
4	7.679	0.223
5	11.843	0.112
6	17.875	0.109
7	30.271	0.193
8	36.267	0.338
9	44.222	0.367
10	52.297	0.126

6.4 Considerações finais

Este Capítulo teve o propósito de apresentar os resultados obtidos a partir de traços de OTDR, que foram obtidos a partir do *software* que contém o algoritmo do Capítulo anterior. Parte dos traços fazem parte de um banco de dados de testes realizados em campo com todas as adversidades.

Na seção 6.2 foi realizado uma série de testes para podermos estabelecer uma relação entre o número de neurônios na camada escondida, número de épocas de treinamento e tempo de treinamento, para melhor compreensão desta relação foi apresentada a Tabela 3.

Na seção 6.3 são apresentados os resultados dos testes, em cada figura há três gráficos que permite visualizar em que lugar foram encontrados os eventos de fusão. Abaixo dos gráficos é apresentada uma tabela com os eventos encontrados no traço, a distância que foram encontrados e a atenuação de cada um deles.

No próximo Capítulo, será apresentada as conclusões referentes aos resultados obtidos neste Capítulo.

7 Conclusões e Contribuições

7.1 Conclusões

Neste trabalho foi desenvolvido um algoritmo para detecção de evento do tipo emenda térmica (fusão) em traços de OTDR. A técnica utilizada neste algoritmo foi a Rede Neural Artificial com o tipo de aprendizado chamado de *backpropagation*. O Capítulo 2 explica o funcionamento do equipamento de OTDR e explica como ele é um dos mais importantes em testes envolvendo fibra óptica, pois como é explicado neste capítulo ele permite saber várias informações sobre a fibra através de apenas uma extremidade.

No Capítulo 3 desta dissertação foi feita uma introdução como surgiu a rede neural artificial a partir do neurônio de McCulloch Pitts, as suas limitações nos primeiros anos e como foi resolvida essas limitações, e seu uso em reconhecimento de padrões. Foi explicado o funcionamento de uma Rede Neural Artificial, como foi proposto o modelo do neurônio artificial, a partir do que é conhecido do neurônio biológico. No capítulo também é explicado com funciona as redes de camada única e as redes multicamadas. Foi dedicada uma seção ao método de treinamento *backpropagation* que permitiu o treinamento de redes neurais com múltiplas camadas, e o mais importante, que a partir do desenvolvimento deste tipo de treinamento foi possível resolver problemas que não são linearmente separáveis, possibilitando que uma rede neural, seja possível de resolver problemas complexos. Na seção foi descrito passo a passo como funciona este método de treinamento

A partir daí, no Capítulo 4, através do software Matlab foi desenvolvido o algoritmo seguindo o passo-a-passo mostrado no Capítulo 3 para detecção de eventos de emenda térmica em traços de OTDR. Foram desenvolvidos dois programas, no primeiro é utilizado para os ajustes de pesos através do método de treinamento *backpropagation*, após ajustados os pesos são salvos para serem utilizados posteriormente no segundo programa. O segundo programa utiliza os pesos já ajustados no primeiro programa em uma rede neural, e a partir daí os traços de OTDRs são “varridos” pela rede neural para a identificação dos eventos. Também foi desenvolvido no algoritmo a separação das partes em que foram detectados.

No Capítulo 5 foram demonstrados os resultados, na primeira parte do capítulo é feita uma relação entre o número de camadas escondidas, número de épocas de treinamento e tempo de treinamento e foi concluído que:

-Quanto maior o número de neurônios na camada escondida, menor o número de épocas o programa atende ao critério de parada, porém com o número maior de neurônios na camada escondida, aumenta-se o número de operações matemáticas com isso aumenta o tempo de processamento como mostrados na Figura 30 e Tabela 3.

No resto do capítulo foram mostrados cinco casos, em que se utilizou a detecção de fusões em traços de OTDR.

Foi concluído que:

-Após uma inspeção visual nos traços de OTDR, foi constatado que o algoritmo conseguiu detectar os eventos com precisão, ou seja, é possível utilizar uma rede neural para a detecção deste tipo de evento.

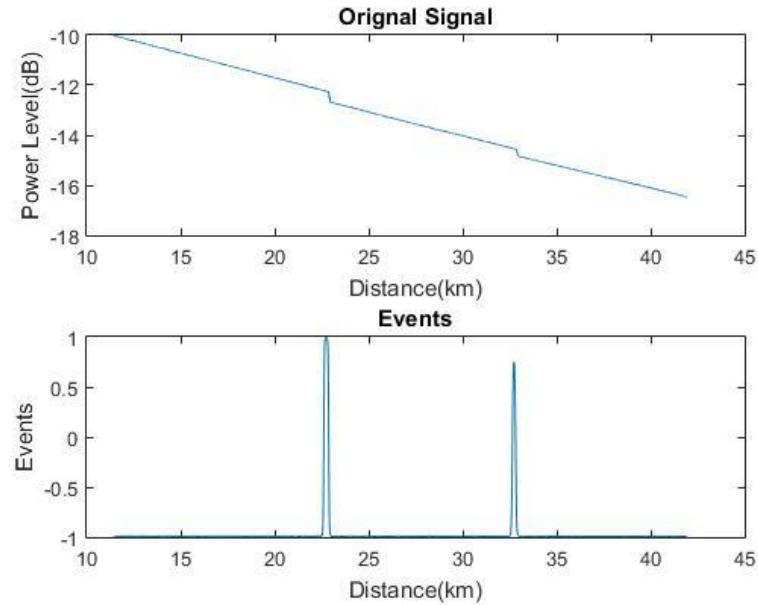
- O algoritmo de treinamento *backpropagation* apresenta uma vantagem de poder ser desenvolvido em poucas linhas de código, e funciona eficientemente.

- O gráfico em que são demonstrados a detecção dos eventos o gráfico apresenta -picos onde existe uma fusão, e a amplitude deste pico é proporcional a fusão com a menor atenuação utilizada no treinamento, e caso a fusão detectada tenha o valor maior do que a que foi utilizada no treinamento, o pico se satura no valor 1. Ou seja, caso a menor atenuação utilizada no treinamento tenha valor igual a 1dB e, no traço de OTDR testado tenha uma fusão com o valor 0,5 dB, o pico para esta fusão será em torno do valor 0, uma vez que a rede foi treinada com uma função sigmoide bipolar que varia de -1 a 1.

-Após a RNA treinada, para sua utilização o tempo gasto é praticamente instantâneo.

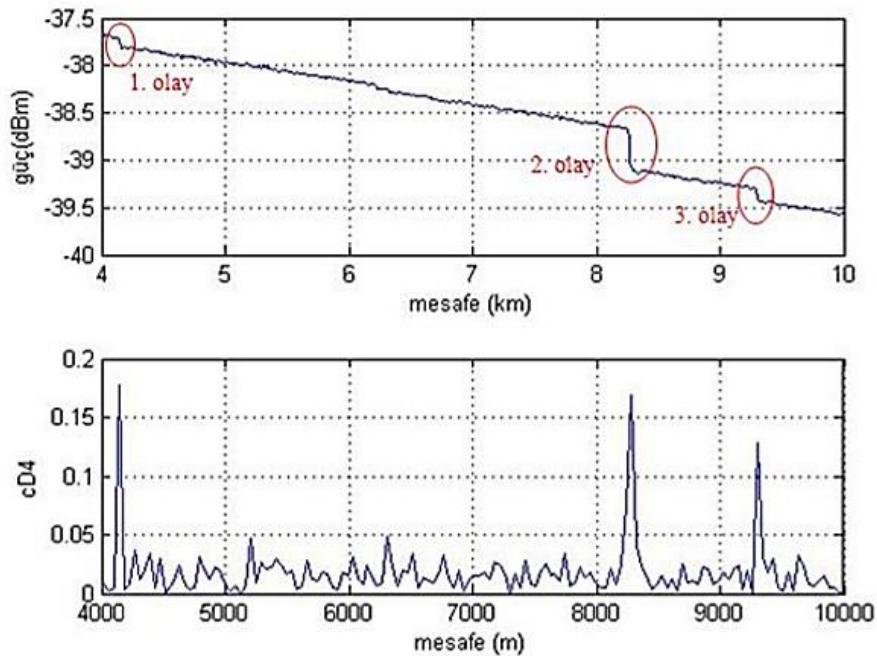
Na Figura 37, foi separada somente a parte em que ocorrem os eventos de fusão no Caso 1, o motivo dessa separação, foi para a comparação com a Figura 4 em que foi utilizada a transformada wavelet para a detecção dos eventos de fusão, não foi utilizado o mesmo traço de OTDR, porém foi utilizado um traço com característica parecida . A título de melhor observação a Figura 4 foi repetida.

Figura 37 - Parte em que ocorreram os eventos no Caso 1.



Fonte: Evandro Monteiro Jorge Júnior, 2018.

Figura 4 - Analise por wavelet de eventos de Fusão.



Fonte: (ACAR, 2016)

Podemos perceber que a RNA apresenta para traços de OTDR com características parecidas, resultados bem semelhantes ao que se utilizou transformada wavelet. Verificou-se

também que para este caso a RNA foi mais imune aos ruídos do traço de OTDR do que a transformada wavelet.

7.2 Contribuições

O estudo apresentado nesta dissertação traz algumas contribuições importantes, visto que estudos utilizando Inteligência Artificial, estão em alta, com várias publicações nesta área terem sido realizadas ultimamente.

A principal contribuição deste trabalho foi desenvolver um novo algoritmo que pode ser utilizado em traços de OTDR. Foi importante desenvolver um estudo envolvendo esta área junto à área de comunicações ópticas. Uma vez que o Brasil deve alcançar em 2018, 9 milhões de quilômetros de cabos ópticos por ano (BUCCO, 2015), e todos esses cabos após implantados necessitam de testes, e melhorando a qualidade desses testes ajudaria no desenvolvido das telecomunicações no Brasil.

7.3 Sugestões para trabalhos futuros

A partir deste trabalho é possível desenvolver outros trabalhos, como por exemplo:

- Converter o *software* da linguagem Matlab para uma linguagem de programação que possa ser utilizada comercialmente como por exemplo Python, Java ou C#.
- Com a linguagem de programação comercial, desenvolver uma interface gráfica para que fique mais fácil de usuários com menor nível de conhecimento de programação possa utilizar o *software*.
- O trabalho a princípio foi desenvolvido somente para detecção de eventos do tipo emenda térmica, o treinamento pode ser aumentado para detecção de eventos do tipo picos de reflexão, inclusive picos de reflexão na zona de ruído do traço, para o algoritmo ficar mais completo.
- Embarcar o programa desenvolvido nesta dissertação em equipamentos de OTDR.
- O programa foi desenvolvido baseado no equipamento de OTDR, ou seja, equipamento utilizado em comunicação óptica. Poderia ser desenvolvido programa semelhante para equipamento de TDR que é utilizado para realizar testes em cabos metálicos e que também possuem padrões que podem ser detectados através de RNA.

Referências

- ABBADE, A. L. R.; CAMPOS, M.R. Aplicação do OTDR na Análise de Problemas de Atenuação em Fibras Ópticas: Estudo de Casos. **Revista Científica Periódica – Telecomunicações**, v.5,n.2, INATEL, Dezembro 2002.
- ACAR, H. Analysis of OTDR Measurement Data With Wavelet Transform. **Advanced Research in Electrical & Eletronics Systems**,v.1,n.1, Diyarbarkir, 10 Novembro 2016. 1-5.
- ANRITSU. **MT9083 Access Master for OTDR**. [S.1.]: [s.n.], 2006.
- ANRITSU. **Understanding OTDRs**, 2011. Disponível em: <<https://rossfibersolutions.com/fiber-optic-pdfs/Anritsu-understanding-otdrs.pdf>>. Acesso em: 3 dez. 2017.
- BAGAD, V. S. **Optical Fiber Communications**. Pune: Technical Publications, 2009.
- BARNOSKI, M. K.; JENSEN, S. M. Fiber waveguides: a novel technique for investigating attenuation characteristics. **Applied Optics**, v. 15, p. 2, 1975.
- BUCCO, R. Telesintese. **Brasil vai consumir 9 milhões de KM de cabo óptico em 2018, prevê MiniCom**, 2015. Disponível em: <<http://www.telesintese.com.br/brasil-vai-consumir-9-milhoes-km-de-cabo-optico-em-2018-preve-minicom/>>. Acesso em: 31 maio 2018.
- CHAOJU, H.; JUN, L. **The Application of Wavelet Transform in Analysis of OTDR Curve**. 2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics. Nanjing, Jiangsu: [s.n.]. 2010. p. 216-219.
- CHAUVIN, Y.; RUMELHART, D. E. **Backpropagation: Theory, Architectures, and Applications**. Hillsdale Hove: Lawrence Erlbaum Associates, 1995.
- CHOMYCZ, B. **Fiber Optic Installer's Field Manual**. New York: McGraw Hill, 2000.
- DATA SCIENCE ACADEMY. **Deep Learning Book**, 2018. Disponível em: <<http://deeplearningbook.com.br>>. Acesso em: 5 Julho 2018.
- DAVID, K. **A Brief Introduction to Neural Networks**, 2007. Disponível em: <<http://www.dkriesel.com>>. Acesso em: 5 mar. 2018.
- DAVID, M. Como Amazon, Google, Microsoft e IBM Oferecem IA como um Serviço. **Ciencia e Dados**, 2018. Disponível em: <<http://www.cienciaedados.com/como-amazon-google-microsoft-e-ibm-oferecem-ia-como-um-servico/>>. Acesso em: 7 Julho 2018.

- FAUSETT, L. V. **Fundamentals of neural networks: architectures, algorithms, and applications.** 1^a. ed. [S.l.]: Prentice Hall, 1992.
- GN NETTEST. **Understanding OTDRs.** [S.l.]: [s.n.], 2000.
- HAN QIANG et al. **n OTDR event analysis algorithm based on EMD-based denoising and wavelet transform.** 12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI). Qingdao: [s.n.]. 2015. p. 1034-1038.
- HAYKIN, S. **Neural Networks - A comprehensive Foundation.** Canada: Prentice Hall/Pearson, 2001.
- HAYKIN, S. **Neural Networks and Learning Machines.** Nova Iorque: Prentice Hall/Pearson, 2009.
- HECHT-NIELSEN, R. **Neurocomputing.** Reading: Addison-Wesley, 1990.
- HEWLETT-PACKARD. **Usando um Refletômetro Óptico (OTDR) para diagnosticar falhas na rede óptica.** Brasil: Hewlett-Packard, 1999.
- KEISER, G. **Comunicações por fibras ópticas.** 4. ed. Porto Alegre: Amgh, 2014.
- LI, H.-Y. S. Git Hub. **The OTDR (Optical Time Domain Reflectometer) Data Format,** 2016. Disponivel em: <[https://github.com/sid5432/pubOTDR/wiki/The-OTDR-\(Optical-Time-Domain-Reflectometer\)-Data-Format](https://github.com/sid5432/pubOTDR/wiki/The-OTDR-(Optical-Time-Domain-Reflectometer)-Data-Format)>. Acesso em: 17 maio 2018.
- MANUAL DO MUNDO. A luz que faz curva na água (experiência de física). **Youtube,** 2012. Disponivel em: <<https://www.youtube.com/watch?v=F69tWoZa4ic>>. Acesso em: 16 jul. 2018.
- MEJÍA SÁNCHEZ, E. C.; MEGGIOLARO, M. A. **Controle por Aprendizado Acelerado e Neuro-Fuzzy de Sistemas Servo-Hidráulicos de Alta Frequencia.** PUC. Rio de Janeiro, p. 130. 2009.
- NAYAK, G. S. et al. Principal component analysis and artificial neural network analysis of oral tissue fluorescence spectra: Classification of normal premalignant and malignant pathological conditions. **Biopolymers**, Manipal, v. 82, n. 2, p. 152-166, 5 Junho 2006. <https://doi.org/10.1002/bip.20473>
- OLIVEIRA, A. R. D. **Redes Neurais Artificiais Aplicadas Na Detecção, Classificação E Localização De Defeitos Em Linhas De Transmissão.** UFJF. Juiz de Fora. 2005.
- OVERTEK. **Manual de Instruções Palm OTDR.** [S.l.]: [s.n.].
- PINHEIRO, J. M. S. Ispblog. **Testes Para Certificação de Redes Ópticas,** 2016. Disponivel em: <<https://www.ispblog.com.br/2016/04/07/testes-para-certificacao-de-redes-opticas/>>. Acesso em: 12 Abril 2018.

RAMOS, A. F. **Sistema de identificação e localização de faltas em linhas de transmissão baseado em redes neurais.** PUC. Rio de Janeiro. 2002.

RAMOS, D. L.; BRAGA, A. M. B.; BARBOSA, C. R. H. **Detecção e caracterização de danos estruturais através de sensores a rede de Bragg e Redes Neurais Artificiais.** PUC. Rio de Janeiro, p. 173. 2013.

VIAVI. Important Factors for choosing an Optical Time Domain Reflectometer (OTDR). **viavisolutions**, 2015. Disponível em: <<https://www.viavisolutions.com/ru-ru/literature/important-factors-choosing-optical-time-domain-reflectometer-otdr-white-paper-en.pdf>>. Acesso em: 2018 maio 20.

W. G. HU et al. **Study on the Detection Signal of OTDR Based on Wavelet Denoising and Approximate Entropy.** 2012 Symposium on Photonics and Optoelectronics. Shanghai: [s.n.]. 2012. p. 4.

WOODWARD, B.; HUSSON, E. B. **Fiber Optics Installer and Technician Guide.** [S.l.]: Sybex, 2005.

ZÓLOMY, A. **Measurement Guide: Optical Time Domain Reflectometer.** Budapest University Of Technology And Economics. Budapest, p. 1-7. 1997.