UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

SEBASTIÃO COSTA MAIA NETO

DESEMPENHO DA DEMODULAÇÃO DIGITAL EM UM CANAL AWGN UTILIZANDO REDES NEURAIS ARTIFICIAIS

PAU DOS FERROS
2022

SEBASTIÃO COSTA MAIA NETO

DESEMPENHO DA DEMODULAÇÃO DIGITAL EM UM CANAL AWGN UTILIZANDO REDES NEURAIS ARTIFICIAIS

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Francisco Carlos Gurgel da Silva Segundo

PAU DOS FERROS

2022

M469d MAIA NETO, SEBASTIÃO COSTA .
DESEMPENHO DA DEMODULAÇÃO DIGITAL EM UM CANAL AWGN UTILIZANDO REDES NEURAIS ARTIFICIAIS / SEBASTIÃO COSTA MAIA NETO. - 2022.
49 f. : il.

Orientador: Francisco Carlos Gurgel da Silva Segundo.
Monografia (graduação) - Universidade Federal Rural do Semi-árido, Curso de Engenharia de Computação, 2022.

1. Demodulação Digital. 2. Redes Neurais. 3. Perceptron de Múltiplas Camadas. I. Silva Segundo, Francisco Carlos Gurgel da , orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automáto em conformidade com AACR2 e os dados fornecidos pelo) autor(a).
Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 1115

SEBASTIÃO COSTA MAIA NETO

DESEMPENHO DA DEMODULAÇÃO DIGITAL EM UM CANAL AWGN UTILIZANDO REDES NEURAIS ARTIFICIAIS

Trabalho de Conclusão de Curso apresentado à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Engenharia de Computação.

Defendida em: 24 /11 / 2022.

BANCA EXAMINADORA

_____
Francisco Carlos Gurgel da Silva Segundo, Prof. Dr. (UFERSA)
Presidente

_____
Marco Diego Aurelio Mesquita, Prof. Me. (UFERSA)
Membro Examinador

_____
Josenildo Ferreira Galdino, Prof. Dr. (UFERSA)
Membro Examinador

Dedico este trabalho aos meus pais, Edileuda e Antônio, ao meu irmão Samuel. Também dedico a todos os meus amigos que contribuem em minha jornada.

AGRADECIMENTOS

Quero agradecer primeiramente a minha família, em particular, minha mãe Edileuda Costa por ser um exemplo de paciência e dedicação, ao meu pai Antônio Gilmar, que me incentiva continuamente, a meu irmão Samuel Alves que tirou minha paciência por grande parte da minha vida e que hoje é um grande companheiro.

A todos os meus familiares que me apoiaram, e que sempre ficavam perguntando se precisava de algo; a família maravilhosa da qual faço parte: Tomaz, Soares e Oiôr.

Agradeço aos meus amigos, que além de me acompanharem nessa jornada, se tornaram presentes em minha vida, a citar: Iorrane Nobre, Edvar Junior, Pedro David, Josimara Silva, Luan Guilherme, Igor Moíses, Fabíola Maia, Alex Potter, Amanda Cristina e em especial aos meus melhores amigos que a UFERSA me deu Brígido Conrado, Fabia Maia e Cayo Ravel, com quem divido minhas alegrias, dúvidas e crises existenciais.

Meu muito obrigado agora é para meu orientador Francisco Carlos Gurgel da Silva Segundo que me permitiu finalizar esse trabalho, oferecendo toda a sua inteligência e dedicação a mim, se tornando uma grande fonte de inspiração.

Dedico a todos esses citados, mas também dedico aqueles que contribuíram para minha jornada acadêmica.

“Convencido eu mesmo,
não procuro convencer os demais.”
Edgar Allan Poe

RESUMO

O processo de utilização das redes neurais artificiais vem sendo bastante empregada como ferramenta de soluções para problemas complexos, como a aplicação na etapa de demodulação de digital de sinais, conseguindo recuperar a informação enviada no canal de comunicação. Com o intuito de analisar esse processo, é comumente encontrado abordagens metodológicas nos quais é analisado o desempenho de redes neurais no processo de demodulação digital. Dada essa circunstância, neste trabalho é apresentado uma análise da aplicação da rede neural do tipo *perceptron* de múltiplas camadas (MLP) com distintos algoritmos de treinamento no processo de demodulação digital em arquivos textos. Dessa forma, foram definidas cinco estratégias de análises com variação no número de camadas escondidas, de neurônios e de épocas, classificando o desempenho de cada rede neural. Com o resultado dessas análises, o algoritmo de treinamento *Bayesian Regularization* obteve 99,5% na taxa de acertos de símbolos com duas camadas escondidas, 60 neurônios e 6000 épocas, alçando resultados satisfatórios e efetivando a estratégia utilizada neste trabalho.

Palavras-chave: Demodulação Digital. Redes Neurais. *Perceptron* de Múltiplas Camadas.

ABSTRACT

The process of using artificial neural networks has been widely used as a solution tool for complex problems, such as the application in the digital signal demodulation stage, to recuperate the information sent in the communication channel. In order to analyze this process, methodological approaches are commonly found in which the performance of neural networks in the digital demodulation process is analyzed. Given this circumstance, this work presents an analysis of the application of a multilayer perceptron neural network (MLP) with different training algorithms in the process of digital demodulation in text files. Thus, five analysis strategies were defined with variation in the number of hidden layers, neurons and epochs, classifying the performance of each neural network. With the result of these analyses, the Bayesian Regularization training algorithm obtained a 99.5% hit rate for symbols with two hidden layers, 60 neurons and 6000 epochs, achieving satisfactory results and implementing the strategy used in this work.

KEYWORDS: Digital Demodulation. Neural Networks. Multilayer Perceptron.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE GRÁFICOS

LISTA DE ABREVIATURAS E SIGLAS

AMC	Classificação Automática de Modulações
ASK	Modulação em Amplitude Por Chaveamento
AWGN	Ruído Aditivo Gaussiano Branco
BFGA	Algoritmo BFGS *quasi*-Newton
BRA	Algoritmo *Bayesian Regularization*
GDA	Algoritmo *Gradient Descent*
LMA	Algoritmo *Levenberg-Marquardt*
MLP	Rede *Perceptron* de Múltiplas Camadas
PSK	Modulação em Fase Por Chaveamento
QAM	Modulação de Amplitude em Quadratura
RBA	Algoritmo *Resilient Backpropagation*
RNA	Rede Neural Artificial
SCGA	Algoritmo *Scaled Conjugate Gradient*

SUMÁRIO

1 INTRODUÇÃO

Este capítulo apresenta uma visão geral deste trabalho e está organizado da seguinte forma: a Seção 1.1 apresenta uma contextualização; a Seção 1.2 apresenta a problematização; a Seção 1.3 explicita os argumentos que justificam o desenvolvimento deste trabalho; a Seção 1.4 apresenta os objetivos; e, por fim, a Seção 1.5 apresenta a organização deste trabalho.

1.1 CONTEXTUALIZAÇÃO

Vivencia-se uma era onde a troca de informações é recorrente no cotidiano, sendo que a todo momento são enviados fotos, áudios, vídeos e arquivos de texto, e às vezes se faz necessário compactar o arquivo para que o mesmo se adeque ao tamanho permitido pela ferramenta tecnológica utilizada e haja a otimização da transmissão no canal de comunicação. Assim, os algoritmos de compressão utilizam da redundância de informações, como as frequências de símbolos repetidos, definindo a localização e os padrões que o referem (VIRAKTAMATH; KOTI; BAMAGOD, 2017).

Ao passo que a troca de informações ocorre, os meios de comunicação digitais estão constantemente evoluindo a fim de melhorar aspectos relacionados à qualidade da informação que é transmitida. Nos últimos anos as redes neurais artificiais vêm sendo empregadas na demodulação de sinais digitais, conseguindo compensar as distorções existentes no modulador e no demodulador, sendo na fase de decisão no receptor, como também na transmissão de dados e no processo de decisão da conversão do sinal (ZHOU, 2003).

Diversas técnicas têm sido apresentadas na literatura para análise de desempenho, sendo a utilização de redes neurais clássicas como medidas estatísticas, possibilitando assim, extrair características das etapas de codificação, simulação de transferências via satélite e a etapa final de demodulação, incentivando inúmeros trabalhos nesta área (FONTES, 2012).

Nesse sentido, este estudo está pautado na intenção de conhecer e analisar o desempenho das redes neurais no auxílio da demodulação de um sinal transmitido em canal de ruído branco gaussiano.

Para analisar o desempenho das redes neurais, realizamos a transmissão de um sinal, contendo um arquivo texto codificado, utilizando a modulação QAM acoplado a um canal de ruído branco AWGN (do inglês, Additive White Gaussian Noise) e demodulamos o sinal com o auxílio de uma rede neural artificial. Analisando os aspectos relacionados ao desempenho da solução implementada, explorando o comportamento do sistema desenvolvido para simular a transmissão perante a variação de alguns parâmetros e a eficiência da rede neural em relação a demodulação 256-QAM.

1.2 PROBLEMÁTICA

O processo de codificação, transmissão e de demodulação é considerado por vezes difícil. A classificação automática de modulações (AMC, do inglês *automatic modulation classification*) que é responsável por identificar a modulação do sinal e entrega a informação enviada a seu destino sem distorção, é um processo crítico e de alto custo computacional (NASCIMENTO *et al*, 2018).

Com isso, Lima (2018) constata que o equalizador baseado em rede neural extrai o sinal desejado do sinal distorcido com a aplicação de um algoritmo adaptativo, que minimiza o erro entre a saída do equalizador e o sinal de teste.

A partir da complexidade da aplicação da AMC e as técnicas de redes neurais, é comumente encontrado abordagens metodológicas nos quais é analisado o desempenho de redes neurais no processo de demodulação digital, no entanto, Wu *et al*. (2005) constatam que novas predições, ou até mesmo revisões, dos dados com base em redes neurais, devem ser constantementes aplicados/verificados para que possamos torná-los cada vez mais adequadas para aplicação designada.

1.3 JUSTIFICATIVA

A aplicação de novas abordagens de redes neurais na área de telecomunicações possibilitaram resultados mais promissores, possibilitando diversas aplicações, servindo para resoluções de problemas complexos de análise e estatísticos (VIEIRA; ROISENBERG, 2006).

Um cenário de bastante aplicação é a compensação de distorção utilizando rede neural, assim, Owaki e Nakamura (2017) consta que processamento digital de sinais baseado em redes neurais tem o mérito de poder compensar de forma adaptativa a distorção usando algoritmos de aprendizado supervisionado, promovendo diversos estudos que buscam análise e quantificar os modelos aplicados e fomentando novas hipóteses de implementação.

Nessa perspectiva, Coutinho (2013) propôs a questão: "As redes neurais do tipo *Self Organizing Map* (SOM) possuem um desempenho satisfatório no processo de demodulação a ponto de substituir um hardware demodulador dedicado?", possibilitando uma análise da qualidade e desempenho do cenário proposto.

Sendo assim, a aplicabilidade do uso de redes neurais para operação do processo de modulação/demodulação, pode-se propor uma análise de estudo do desempenho bastante ampla, possibilitando como base, por exemplo, para o desenvolvimento de técnicas e abordagem mais específicas e voltada para um conteúdo específico, conseguindo uma potencialização dos resultados.

1.4 OBJETIVOS

A Subseção 1.4.1 define o objetivo geral deste trabalho e a Subseção 1.4.2 apresenta os objetivos específicos:

1.4.1 Objetivo Geral

A finalidade deste trabalho consiste em analisar o desempenho da aplicabilidade das redes neurais no auxílio da demodulação de um sinal transmitido em um canal de ruído branco por meio de simulação no *software* MATLAB.

1.4.2 Objetivos Específicos

No intuito de alcançar o objetivo geral deste trabalho foram definidos os seguintes objetivos específicos.

- Realizar um levantamento bibliográfico para conhecer os trabalhos relacionados ao

tema;

- Aplicar a codificação Shannon-Fano em arquivos textos para esquematização de caracteres em símbolos binários;
- Aplicar a rede neural *perceptron* de múltiplas camadas como solução para demodulação digital;
- Submeter a rede neural *perceptron* de múltiplas camadas a distintos algoritmos de treinamentos em diversos cenários;
- Comparar o desempenho do processo de demodulação baseadas em redes neurais.

1.5 ORGANIZAÇÃO DO TRABALHO

O presente trabalho está organizado da seguinte forma: no Capítulo 2 é apresentada a fundamentação teórica utilizada para o desenvolvimento deste trabalho; no Capítulo 3 é descrita a metodologia; no Capítulo 4 são apresentados os resultados e discussões; e, por fim, no Capítulo 5 as considerações finais e a sugestão de trabalhos futuros são apresentadas.

2 REFERENCIAL TEÓRICO

Neste capítulo é descrito o referencial teórico utilizado na composição deste trabalho, sendo organizado da seguinte maneira: na Seção 2.1 são apresentados os conceitos sobre Modulação Digital; Na Seção 2.2 é exposto uma contextualização sobre Redes Neurais Artificiais; e na Seção 2.3 os trabalhos relacionados ao tema abordado são apresentados.

2.1 MODULAÇÃO DIGITAL

A finalidade de um sistema de comunicação é transportar informações a partir de sinais classificados como portadoras, permitindo transmitir informações entre o transmissor e o receptor (HAYKIN, 2007). No entanto, para a transmissão ocorrer, necessita-se de um procedimento de modulação do sinal, na qual a portadora é variada na amplitude, frequência e fase, assumindo um número de estados discretos em decorrência da informação a ser transmitida (FONTES, 2012).

O sinal modulado é uma sequência binária, ou seja, de 0 e 1, transmitindo a informação em forma de símbolo, que durante a transmissão, as formas de onda da portadora modulada são alteradas pelo ruído do canal, na recepção só importa decidir qual o símbolo corresponde à onda original (SANTOS; QUINTANILHA, 2015).

Dessa forma, dependendo de quais parâmetros são alterados na portadora, pode-se determinar tipos de modulação distintas, como Modulação por chaveamento de amplitude (ASK, do inglês *Amplitude Shift Keying*), Modulação por deslocamento de fase (PSK, do inglês *Phase Shift Keying*), Modulação por chaveamento de frequência (FSK, do inglês *Frequency Shift Keying*), Modulação de amplitude em quadratura (QAM, do inglês do inglês *Quadrature Amplitude Modulation*), o tipo de modulação empregada neste trabalho, e entre outras.

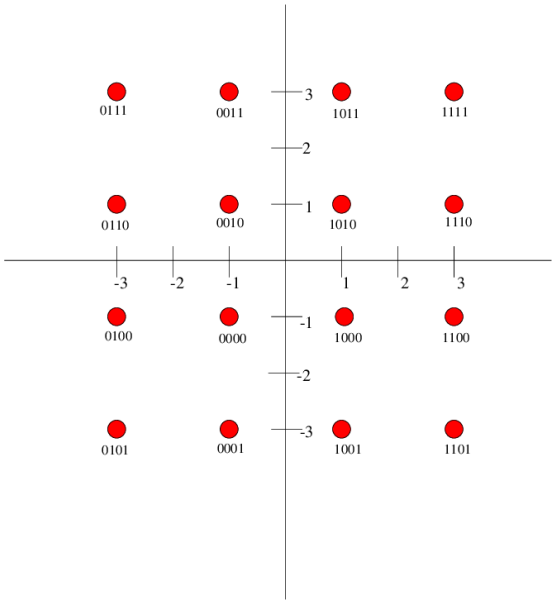
2.1.1 MODULAÇÃO DE AMPLITUDE EM QUADRATURA (QAM)

A modulação de amplitude quadratura é gerada a partir da modificação da amplitude e fase, combinando alguns das características da ASK (amplitude) com a PSK (fase), possibilitando apresentar maiores ordens de modulação, possibilitando alta taxa de

transferência de informação. Nesse sentido, os símbolos são mapeados em um diagrama de fase e quadratura, sendo que cada símbolo apresenta uma distância específica da origem do diagrama que representa a sua amplitude, isto significa que as informações são inseridas nos parâmetros de amplitude e quadratura da onda portadora (LATHI, 2012).

Na Figura 1, tem-se a representação de um mapa de constelação para uma modulação QAM de ordem 16, ou seja, representação da informação de 16 símbolos com informação de quatro bits. Essa representação é possível, por que a modulação 16-QAM apresenta doze possíveis fases e três possíveis amplitudes pela angulação em relação à origem, para cada símbolo transmitido representa a informação de quatro bits (FONTES, 2012). Similarmente ocorre para as demais ordens de simulação.

Figura 1. Mapa de Constelação 16-QAM



Fonte: LATHI, 2012

Por decorrência da sua alta taxa de transmissão de informação, a modulação QAM é amplamente aplicada, alguns dos principais exemplos da sua utilização são os enlaces de rádio digital e de microondas, televisão digital de alta definição, *modens* a cabo e ADSL.

2.1.2 CANAL AWGN (*Additive White Gaussian Noise*)

O meio físico entre o transmissor e o receptor, pode ser aplicado diferentes tipos de canais de comunicações, dependendo fundamentalmente das características do sistema que se

deseja aplicar. Neste trabalho, aplica-se um único sinal aleatório gaussiano branco que se adiciona ao sinal modulado, denominado canal AWGN.

O ruído branco tem se ao fato da sua distribuição uniforme em todo o espectro da sua faixa frequência, possuindo média nula e correlação nula em instantes de tempos distintos entre suas amplitudes, isto é, o ruído da amplitude em um determinado ponto é independente do valor de outro instante de tempo observado (FONTES, 2012).

2.2 REDES NEURAIS

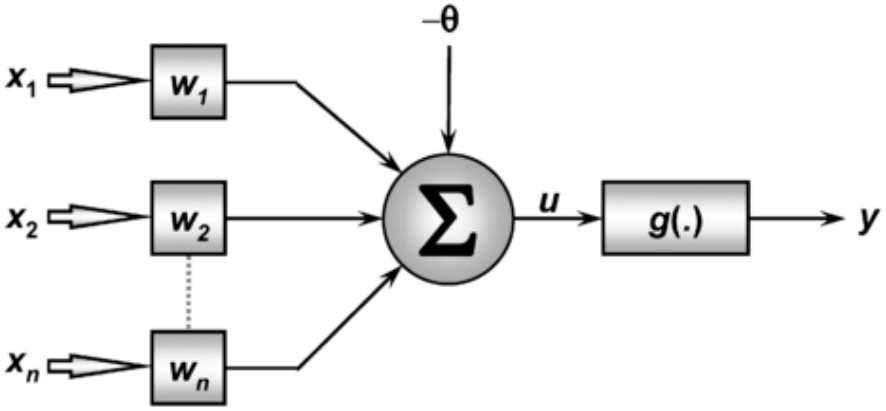
As redes neurais artificiais são modelos computacionais fundamentados no sistema nervoso de seres vivos. Dispõe da capacidade de obtenção e manutenção do conhecimento (baseado em informações) e são determinadas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, que são interligados por um grande número de interconexões (sinapses artificiais), sendo representadas, majoritariamente, por vetores/matrizes de pesos sinápticos (SILVA; SPATTI; FLAUZINO, 2016).

Uma rede neural artificial é composta por várias unidades de processamento, cujo funcionamento é bastante simples, essas unidades são conectadas por canais de comunicação que estão associadas a determinado peso, assim, fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões (SILVA SEGUNDO, 2018). O comportamento inteligente de uma rede neural artificial vem das interações entre as unidades de processamento da rede. Em seu trabalho, McCulloch e Pitts (1943) descrevem um cálculo lógico que unifica a lógica matemática com uma rede neural biológica, isto é, paralelismo e alta conectividade, sendo ainda o modelo mais empregado nas diversas partituras de redes neurais artificiais.

O neurônio é dividido em duas partes definidas como função de rede e função de ativação. A função de rede é responsável pela determinação das entradas da rede X_1, X_2, … , X_n as quais são executadas por meio de multiplicações com os pesos sinápticos W_1, W_2, … , W_n , denota assim, por uma saída u, sendo a soma ponderada das entradas.

Nessa representação, o neurônio pode ser implementado conforme mostrado na Figura 2.

Figura 2. Representação de um Neurônio



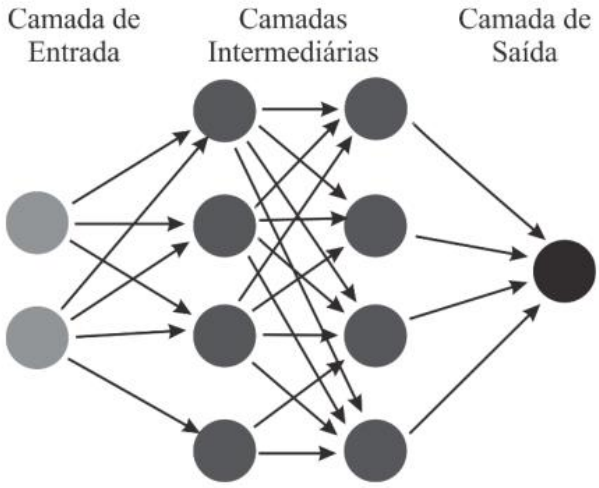
Fonte: Silva, Spatti e Flauzino, 2016.

A função de ativação está relacionada com a variável θ, chamada de bias. Este elemento do neurônio é o elemento para que o resultado alcançado pela combinação linear possa produzir um valor de ativação em direção à saída do neurônio. Dentre estas funções de ativação, pode-se destacar sigmóide, gaussiana, hard-limiter e rampa. Assim, com a função de rede e a função de ativação obtém-se a saída representada pela Equação 1.

$$u = \sum_{i=1}^{N} X_j \cdot W_j + \theta \quad (1)$$

As arquiteturas neurais são divididas em três partes, definidas cada uma como camada, as quais são nomeadas como camada de entrada, intermediária ou escondida e de saída, como representado na Figura 3.

Figura 3. Organização em Camadas



Fonte: Silva, Spatti e Flauzino, 2016.

A camada de entrada, é responsável por receber os dados, onde geralmente ocorre a normalização desses dados. A camada intermediária é composta por neurônios, realizando a maior parte do processamento, extraindo as características associadas às entradas. A camada de saída é composta por um neurônio, sendo responsável em apresentar os resultados obtidos (SILVA; SPATTI; FLAUZINO, 2016).

A seguir, será apresentada a rede *Perceptron* de múltiplas camadas, bem como os algoritmos de treinamento utilizados neste trabalho.

2.2.1 REDE *PERCEPTRON* DE MÚLTIPLAS CAMADAS

A rede *perceptron* de múltiplas camadas, do inglês *multilayer perceptrons* – MLP, é formado por um modelo de arquitetura *feedforward*, sem realimentação, formada por *n* camadas escondidas. Essa rede é constituída por uma camada de entrada, as quais não possuem pesos sinápticos; camadas ocultas, pode ser um ou mais camadas; e uma camada de saída. O treinamento dessa rede é baseado no padrão supervisionado e o algoritmo mais empregado é o *backpropagation* (SILVA; SPATTI; FLAUZINO, 2016).

2.2.2 ALGORITMOS DE TREINAMENTO *BACKPROPAGATION*

O algoritmo de treinamento supervisionado *backpropagation* funciona com base na aprendizagem de padrões, extraindo características dos dados de entradas e conseguindo corrigir seus pesos em todas as camadas, permitindo a rede a formar sua própria representação mais eficiente (BISHOP, 1995).

A partir deste método, vários outros foram desenvolvidos, tais como o *Levenberg-Marquardt*, *Bayesian Regularization*, *Scaled Conjugate Gradient*, *Resilient*, BFGS quasi-Newton, *Gradient Descent*, sendo os algoritmos de treinamento empregado neste trabalho.

2.2.2.1 *Levenberg-Marquardt*

O algoritmo de treinamento *Levenberg-Marquardt* (LMA) consiste em corrigir os pesos da rede a partir do comportamento não-linear baseado nos mínimos quadrados do método de Newton, conseguindo uma aproximação para a matriz Hessiana (J), mostrada da Equação 2.

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{2}$$

O valor de μ é diminuído após cada passo bem sucedido (redução na função de desempenho) e é aumentado apenas quando um passo experimental aumenta a função de desempenho. Desta forma, a função de desempenho é sempre reduzida a cada iteração do algoritmo (MARQUARDT, 1963).

2.2.2.2 *Bayesian Regularization*

O algoritmo *Bayesian Regularization* (BRA) atualiza seus valores de pesos e o bias θ a partir do algoritmo de *Levenberg-Marquardt*, utilizando a minimização dos erros quadrados e pesos, para determinar a combinação mais eficiente produzida com resultados de boa qualidade de generalização.

O BRA introduz pesos de rede na função objetivo de treinamento que é denotada pela Equação 3.

$$F\ (\omega)\ = \alpha E_\omega + \beta E_\omega \tag{3}$$

Os parâmetros E_ω é a soma dos pesos da rede ao quadrado e E_ω é a soma dos erros da rede. Ambos α e β são os parâmetros da função objetivo, definidos usando o teorema de *Bayes*, ou seja, com base em suas probabilidades anteriores (ou marginais) e posteriores (ou condicionais). Os pesos da rede são vistos como variáveis aleatórias, e então a distribuição dos pesos da rede e do conjunto de treinamento são considerados como distribuição gaussiana (BAGHIRLI, 2015).

2.2.2.3 *Scaled Conjugate Gradient*

O algoritmo *Scaled Conjugate Gradient* (SCGA) aplica a retropropagação para calcular as derivadas do desempenho em relação às variáveis de peso e de bias θ, sendo que ajusta os pesos na direção da descida mais íngrime, ou seja, a mais negativa do gradiente. Essa é a direção em que a função de desempenho está diminuindo mais rapidamente.

No algoritmo *Scaled Conjugate Gradient* uma busca é realizada ao longo de tal direção que produz convergência geralmente mais rápida do que a direção de descida mais íngreme, preservando a minimização de erros alcançada em todas as etapas anteriores (MOLLER, 1993).

O procedimento geral para determinar a nova direção de busca é combinar a nova direção de descida mais íngreme com a direção de busca anterior, sendo definida pela Equação 4.

$$p_k = -g_k + \beta_k p_{k-1} \tag{4}$$

Sendo p_k o valor atual, g_k é o gradiente, p_{k-1} o valor da iteração anterior, β_k é o gradiente conjugado, mostrado na Equação 5, que é a razão entre a norma ao quadrado do gradiente atual e a norma ao quadrado do gradiente anterior.

$$\beta_k = \frac{{g_k}^T g_k}{{g_{k-1}}^T g_{k-1}} \tag{5}$$

2.2.2.4 *Resilient Backpropagation*

O algoritmo *Resilient Backpropagation* (RBA) como o *Scaled Conjugate Gradient* é utilizado para modificar o desempenho a partir das variáveis de peso e de bias θ, sendo que o peso ajustado é o mais a esquerda no gradiente (RIEDMILLER, 1993). O algoritmo RBA se baseia na Equação 6.

$$dX = \Delta_x \cdot sign(gX) \tag{6}$$

Os elementos de Δ_x são todos inicializados para Δ_0, ou seja, para as interações anteriores, e gX é o gradiente.

Na iteração do algoritmo uma busca é realizada ao longo de tal direção que produz convergência geralmente mais rápida do que a direção de descida mais íngreme, preservando a minimização de erros alcançada em todas as etapas anteriores, sendo chamada de direção conjugada. O algoritmo RBA é utilizado com busca de linha. Isso significa que o tamanho do passo é aproximado com uma técnica de busca em linha, evitando o cálculo da Matriz Hessiana para determinar a distância ideal para se mover ao longo da pesquisa atual (BAGHIRLI, 2015).

2.2.2.5 BFGS quasi-Newton Backpropagation

No algoritmo BFGS quasi-Newton (BFQN), a retropropagação é usada para calcular as derivadas do desempenho em relação às variáveis de peso e bias θ, cada variável é obtida a partir da Fórmula 7.

$$X = X + a \cdot dX \tag{7}$$

O parâmetro dX é a direção da busca e o parâmetro a é selecionado para minimizar o desempenho ao longo da direção de busca. A primeira direção é a mais a esquerda do gradiente de desempenho, e para as sucessivas interações é aplicada a Fórmula 8,

$$dX = -H/gX \tag{8}$$

gX é o gradiente e H é uma matriz Hessiana aproximada (DAS; PRATIHAR, 2019).

2.2.2.5 Gradient Descent Backpropagation

O algoritmo *Gradient Descent* (GD), a retropropagação é usada para calcular as derivadas do desempenho de desempenho em relação às variáveis de peso e bias θ. O comportamento se dá pelo o ajuste do peso pelo o gradiente negativo entre os instantes de iterações, ou seja, se a perda diminui com o aumento do peso, o gradiente será negativo. A Fórmula 9 mostra o cálculo do peso,

$$w_i = w_{i-1} - \alpha \cdot |dE/dw| \tag{9}$$

onde w_{i-1} é o peso anterior, α é a taxa de aprendizado e $|dE/dw|$ é a derivada do erro pelo o peso (SONG; WU; SOH, 2008).

2.3 TRABALHOS RELACIONADOS

Shi et al. (2020) elaboraram uma forma de reduzir a complexidade da demodulação do esquema de modulação APSK (*Amplitude-phase shift keying)* por meio do emprego de redes neurais. O APSK é um tipo de modulação multinível usada nas comunicações via satélite que visa principalmente otimizar a utilização da banda de frequência e reduzir os efeitos provenientes do uso do amplificador de potência não-linear. A proposta dos autores é reduzir a complexidade da demodulação a partir de uma rede neural Hopfield, aumentando a estabilidade da demodulação. Os testes mostraram que a demodulação com rede neural se mostrou mais estável e rápida que os métodos tradicionais.

Ainda sobre comunicações via satélite, Ibnkahla (2000) traz em seu estudo uma proposta de pré-distorção baseada em redes neurais. As comunicações via satélite que fazem uso de amplificadores não-lineares tendem a serem sensíveis a distorções não-lineares quando o número de símbolos cresce. Nesse sentido o autor emprega o uso das redes neurais para modelar a Transformada de Fourier do amplificador não linear.

Já Zhang et al. (2020) um receptor baseado em redes neurais para comunicações via satélite de órbita terrestre baixa. O receptor em questão visa minimizar os problemas provenientes dos amplificadores não-lineares e melhorar a qualidade da transmissão, nesse sentido a rede neural é treinada para aprender sobre os efeitos do canal, não-linearidades dos amplificadores, esquemas de modulação digital na recepção do sinal para demodulação e compensação de efeitos não desejados. Os resultados apontam compatibilidade com diferentes tipos de modulações e um desempenho satisfatório de taxa de erro de bit.

Catanese et al. (2020) propõe a utilização de redes neurais na transmissão de dados via fibra óptica a fim de minimizar os efeitos não-lineares resultantes do meio de transmissão. Os autores apontam que os meios para mitigar os efeitos não-lineares na fibra são trabalhosos e dependem de recursos computacionais pesados, nesse sentido é feita uma proposta para reduzir tais efeitos a partir de uma função de transferência inversa do canal de transmissão não-linear que faz uso das redes neurais artificiais.

Zhou (2003) utiliza a rede neural artificial de Kohonen para compensar distorções lineares em sistemas de comunicação digital. O autor simula a aplicação da rede neural em um

sinal QPSK e mostra que o sistema proposto minimiza problemas de distorções estáticas e mutáveis, comprovando a eficácia do esquema.

Diante do exposto, pode-se dizer que as redes neurais vêm sendo utilizadas para melhorar o desempenho no processo de demodulação digital.

3 METODOLOGIA

Neste capítulo é apresentado o procedimento de execução deste trabalho e está organizado da seguinte forma: a Seção 3.1 apresenta a condução da pesquisa; a Seção 3.2 expõe a estratégia para a implementação do codificador e decodificador; e por fim, a Seção 3.3 apresenta a implementação da modulação e demodulação, como também a rotina utilizada nas análises da rede neural aplicada.

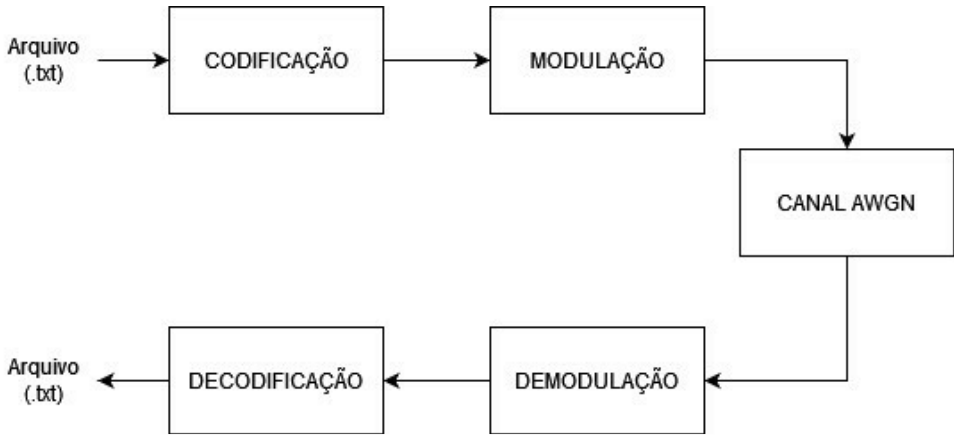
3.1 CONDUÇÃO DA PESQUISA

A pesquisa consistiu em uma simulação computacional baseada no software *Matlab*, com as rotinas definidas para análises da aplicação da rede neural MLP. Com base na seleção dos dados de amostra, se deu início ao processo de coleta, e posteriormente, foram processados os sinais.

Desta forma, este trabalho pretende obter uma constatação baseada em uma simulação laboratorial, sendo as condições de coleta de dados controlada, a fim de poder gerar resultados decorrentes do experimento que possam ser devidamente interpretados e avaliados a fim de se gerar as conclusões do estudo.

O desenvolvimento desse trabalho se dará na seguinte rotina, representada na Figura 4.

Figura 4. Rotina da Execução do Trabalho



Fonte: Autoria própria, 2022.

Inicialmente, utilizando um arquivo texto do formato ‘*.txt*’, têm-se o processo de

codificação Shannon-Fano na compressão para um arquivo binário. Seguindo para modulação do sinal, onde-se o sinal de entrada é modulado na ordem de modulação M especificada, conseguindo o gráfico de constelação do sinal de saída. Continuando, adiciona-se um ruído branco ao passar o sinal de saída pelo um canal AWGN, com uma potência a ser determinada em dBW. Posteriormente, passou-se esse sinal pelo processo de demodulação, aplicando o sinal com ruído na rede neural, conseguindo demodular o sinal. Por fim, pode-se utilizar a decodificação Shannon-Fano, conseguindo obter o arquivo original na saída.

3.2 IMPLEMENTAÇÃO DO CODIFICADOR E DECODIFICADOR

A implementação do algoritmo seguiu o método apresentado por Shannon (1948), seguindo a diretriz de uma árvore ponderada com frequência de ocorrência de cada símbolo no texto, ocorrendo a divisão da árvore e representação dos símbolos por sequências binárias com base na sua recorrência no texto.

No processo de decodificação, utilizou-se o alfabeto obtido na etapa de codificação, conseguindo traduzir o texto binário (codificado) no texto original padrão.

3.3 IMPLEMENTAÇÃO DA MODULAÇÃO E DEMODULAÇÃO

Na etapa de implementação da modulação e demodulação, o principal objetivo é modular o sinal obtido na etapa de codificação, aplicar o canal de ruído branco (AWGN) e aplicar uma análise ao processo de demodulação com utilização de redes neurais *backpropagation*.

3.3.1 Implementação da Modulação e Aplicação do Canal Ruído

A modulação implementada é QAM (do inglês, *Quadrature Amplitude Modulation*), pois é amplamente utilizada para sistemas que necessitam de alta taxa de transferência de informação. Consiste em duas portadoras que são utilizadas em quadratura, sendo que cada símbolo obtido no processo de codificação é representado por um valor de amplitude e fase. Assim, pode-se aplicar a amostra a canal de ruído branco, sendo utilizando um ruído de 10dB,

obtendo assim, diagrama de constelação para as diferentes ordens de modulação, sendo que neste trabalho foi empregado a modulação 256-QAM.

3.3.2 Análise e Implementação da Demodulação

Na finalidade deste trabalho em analisar o desempenho das redes neurais no auxílio da demodulação, definiu-se algumas estratégias para a obtenção de dados. Como a rede utilizada é *perceptron* de múltiplas camadas com *Backpropagation*, definiu-se inicialmente quais tipos de algoritmo seriam analisados, sendo eles: *Levenberg-Marquardt Backpropagation*, *Bayesian Regularization Backpropagation*, *Scaled Conjugate Gradient Backpropagation*, *Resilient Backpropagation*, *BFGS quasi-Newton Backpropagation*, *Gradient Descent Backpropagation*.

Para a função de ativação, foi utilizada a sigmóide, e como seus resultados se encontram na faixa de [0, 1], utilizou-se um artifício de aproximação numérica para conseguir na saída apenas representação binárias, ou seja, 0 e 1, no intuito de ser condizente com codificação empregada.

Com isso, definiu-se cinco análises distintas com mudanças no número de camadas escondidas, sendo aplicada uma ou duas câmadas, mudando o número de neurônios 20 ou 40, e também, na números de épocas (interações), 1000 ou 2000. Assim, pode-se dividir em cinco análises.

- Primeira análise: 20 neurônios e 1000 épocas para uma e duas camadas escondidas.
- Segunda análise: 40 neurônios e 1000 épocas para uma e duas camadas escondidas.
- Terceira análise: 20 neurônios e 2000 épocas para uma e duas camadas escondidas.
- Quarta análise: 40 neurônios e 4000 épocas para uma e duas camadas escondidas.
- Quinta análise: Identificar o melhor algoritmo de treinamento, a partir das quatro análises anteriores, e variando o número de neurônios e iterações.

Por fim, utilizou-se os seguintes dados coletados para analisar quais redes tendem a ter o melhor desempenho, e como ela se comporta em cada cenário, definindo a melhor estratégia para o processo de demodulação.

- Números de acertos e erros: A partir da matriz de confusão definir a porcentagem de símbolos correspondente da saída com a entrada.
- Performance: define o desempenho da rede calculado de acordo com os valores de treinamento, entrada e saída.
- Regressão linear (R): quantifica a dispersão entre a entrada e saída.
- Tempo de execução: Definir o tempo que cada rede demorou para executar o seu treinamento.

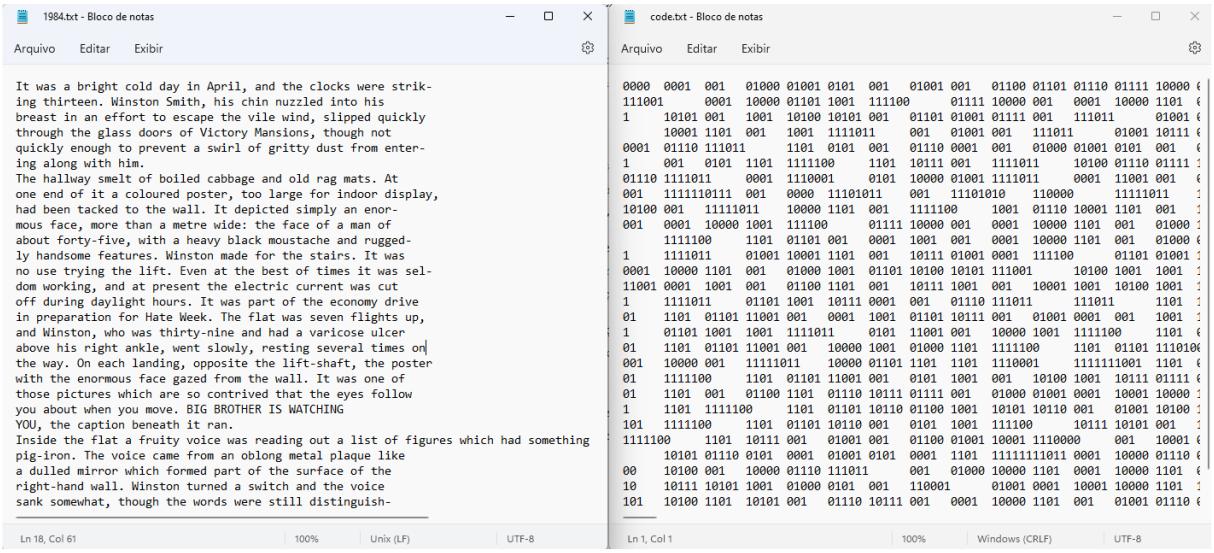
4 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados, as estatísticas dos desempenhos e a análise da eficiência e comportamento redes neurais aplicadas, comparando os algoritmos de aprendizado utilizados, mostrando assim, um levantamento quantitativo e qualitativo.

4.1 IMPLEMENTAÇÃO DO ALGORITMO DE CODIFICAÇÃO E DECODIFICAÇÃO SHANNON-FANO

A implementação do algoritmo seguiu o método apresentado por Shannon (1948), seguindo a diretriz de uma árvore ponderada com frequência de ocorrência de cada símbolo no texto, ocorrendo a divisão da árvore e representação dos símbolos por sequências binárias com base na sua recorrência no texto. A partir disso, faz-se a codificação do arquivo modelo para o arquivo codificado, como representado na Figura 5.

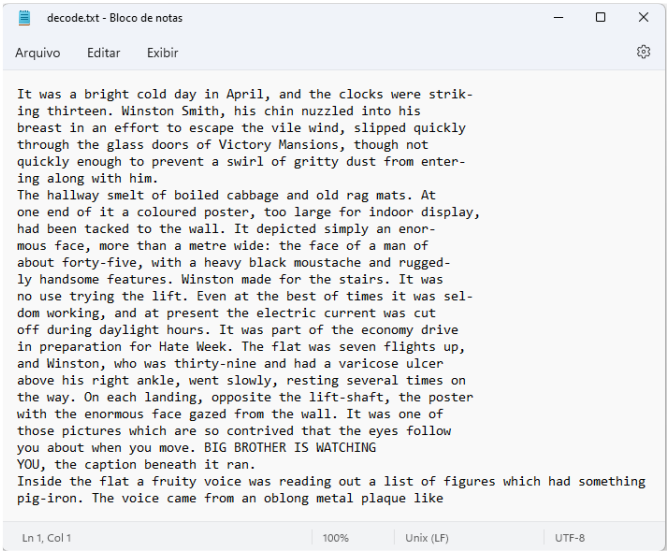
Figura 5. Texto Original e Texto Codificado.



Fonte: Autoria própria, 2022.

No processo de decodificação, utilizou-se o alfabeto obtido na etapa de codificação, recuperou-se a informação do arquivo texto original, conseguindo alcançar um processo satisfatório na decodificação do texto, uma vez que se recuperou sem erro o texto codificado, como representado na Figura 6.

Figura 6. Texto Decodificado.



Fonte: Autoria própria, 2022.

4.2 ESQUEMATIZAÇÃO DA MODULAÇÃO QAM E DO CANAL RUÍDO

A modulação QAM foi-se utilizada pela sua maior taxa de transferência, sendo que retrata cada símbolo por amplitude e fase, ocorrendo a necessidade de duas portadoras com deslocamento de fase.

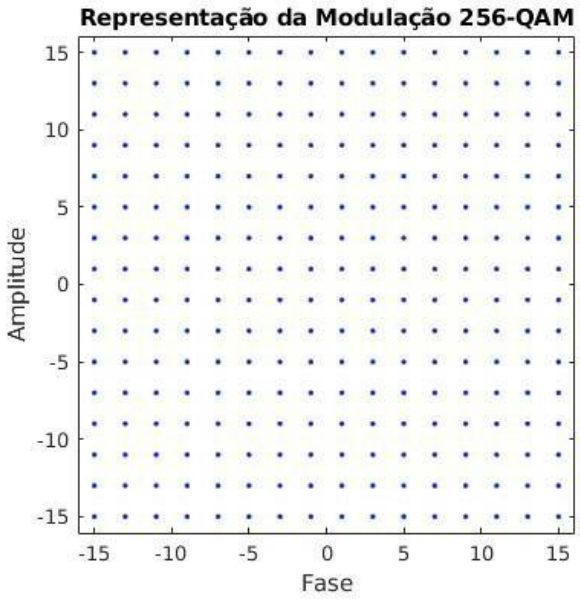
Para a representação simulada de um sistema de telecomunicação, aplicou-se através do *software Matlab*, um ruído branco gaussiano (AWGN), na finalidade de uma melhor representação de um canal real de transmissão. Assim, determinando um ruído de 10dB adicionado ao mapa de constelação QAM.

Com isso, e definindo a ordem de modulação aplicada neste trabalho, 256-QAM, pode-se alterar o parâmetro da ordem de modulação, sendo representado pela letra *M*, assim, conseguindo obter os mapas de constelações, como pode-se observar na Figura 7 e na Figura 8.

Na Figura 7, tem-se a representação do mapa de constelação da modulação 256-QAM, onde cada ponto representa um símbolo, variando sua amplitude e frequência, assim como mencionado na Subseção 2.1.1.

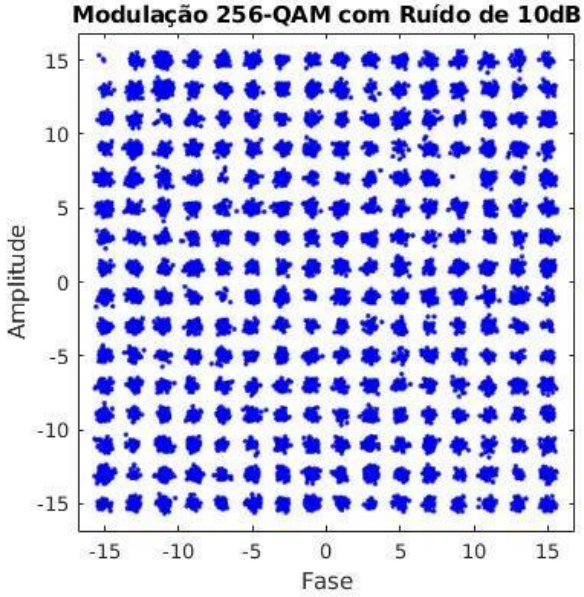
Observa-se que na Figura 8, a representação da informação adquirida na codificação, sendo que mostra 256 símbolos com informação de oitos bits cada.

Figura 7. Representação do Mapa de Constelação para a Modulação 256-QAM



Fonte: Autoria própria, 2022.

Figura 8. Representação da Modulação 256-QAM com Ruído AWGN de 10dB



Fonte: Autoria própria, 2022.

4.3 DEMODULAÇÃO COM REDE NEURAL

Com a finalidade de analisar o desempenho das redes neurais do tipo *perceptron* de múltiplas camadas no processo de demodulação de um sinal, buscou-se aplicar alguns diferentes algoritmos de treinamento, buscando a melhor estratégia de reverter o processo da modulação sem perdas.

Dessa forma, buscou-se na literatura alguns algoritmos de treinamento, sendo escolhidos os seguintes:

- *Levenberg-Marquardt Backpropagation* (LMA)
- *Bayesian Regularization Backpropagation* (BRA)
- *Scaled Conjugate Gradient Backpropagation* (SCGA)
- *Resilient Backpropagation* (RBA)
- *BFGS quasi-Newton Backpropagation* (BFGA)
- *Gradient Descent Backpropagation* (GDA)

Levando em consideração a topologia do problema, que é mapeamento de uma classificação binária, uma vez que os resultados se encontram na faixa de [0, 1], e a determinar de uma saída a partir da amplitude e fase do sinal QAM, sendo assim, denota-se a utilização da função de ativação sigmóide. Com isso, treinou-se uma rede neural para modulação 256-QAM, por ser o modelo mais complexo, assim assegurando sua eficiência para outras ordens de modulação como 2-QAM, 8-QAM, 16-QAM, 64-QAM, para cada tipo de algoritmo de treinamento com a função de ativação em questão, e sumarizando os resultados obtidos.

4.3.1 Primeira Análise do Desempenho da Rede Neural

No primeiro momento, foi aplicado na rede neural *perceptron* de múltiplas camadas com uma camada oculta e a com a função de ativação sigmóide, definido 20 neurônios com 1000 épocas (número de interações). Assim, pode-se obter os números de acertos e erros dos símbolos a partir da matriz de confusão, a performance e a regressão linear (R) dos alvos em relação às saídas, mostrado na Tabela 1.

Tabela 1. Dados de Treinamento em MLP de Camada Única com 20 Neurônios e 1000 épocas

Algoritmo de treinamento	**Número de neurônios**	**Número de interações**	**Acertos/Erros**	**Performance**	**R**	**Tempo**
LMA	20	1000	76,9%/23,1%	0,0942	0,81256	0:03:14
BRA	20	1000	68,4%/31,6%	0,0988	0,80610	0:03:14
SCGA	20	1000	52,8%/47,2%	0,3067	0,31620	0:00:11
RBA	20	1000	51,6%/48,4%	0,2815	0,34406	0:00:06
BFGA	20	1000	46,6%/53,4%	0,3087	0,38387	0:00:38
GDA	20	1000	13,7%/86,3%	0,4497	0,13923	0:00:08

Fonte: Autoria própria, 2022.

Com base na Tabela 1, percebe-se um melhor resultado nos algoritmos de treinamento *Levenberg-Marquardt* (LMA) com 76,9% de acertos e com 23,1% de erros, como também um melhor resultado na performance da rede com 0,0942 e com um valor R de 0,81256; e também no treinamento utilizando *Bayesian Regularization* (BRA) com 68,4% de acertos e 31,6% de erros, tendo uma performance de 0,0988 e regressão linear (R) de 0,80610. No entanto, os algoritmos *BFGS quasi-Newton* (BFGA) e *Gradient Descent* (GDA) tiveram um total de erros maior que o total de acertos, invalidando sua aplicação para o cenário proposto por esse trabalho.

Seguindo a mesma conduta, aplicou-se os algoritmos de treinamentos para a rede neural *perceptron* de múltiplas camadas com duas camadas ocultas e a com a função de ativação sigmóide, definido 20 neurônios com 1000 épocas (número de interações), mostrado na Tabela 2.

Tabela 2. Dados de Treinamento em MLP de Camada Dupla com 20 Neurônios e 1000 épocas

Algoritmo de treinamento	**Número de neurônios**	**Número de interações**	**Acertos/Erros**	**Performance**	**R**	**Tempo**
LMA	20	1000	83,4%/16,6%	0.0852	0,82942	0:03:17
BRA	20	1000	67,2%/32,6%	0.0997	0,80037	0:03:31
SCGA	20	1000	54,2%/45,8%	0.2610	0,47738	0:00:07

RBA	20	1000	51,3%/48,7%	0.2919	0,41534	0:00:30
BFGA	20	1000	48,8%/51,2%	0.3163	0,36649	0:00:41
GDA	20	1000	22,3%/77,7%	0.4462	0,10608	0:00:06

Fonte: Autoria própria, 2022.

De acordo com a Tabela 2 e comparando com os resultados representado na Tabela 1, percebe-se que tanto o algoritmo *Levenberg-Marquardt* e *Bayesian Regularization* mantém o melhor resultado, sendo que a incorporar uma outra camada, *Levenberg-Marquardt* teve um aumento de 76,9% para 83,4% de acertos, entretanto, *Bayesian Regularization* permaneceu na mesma margem de erros e acertos, com 68,4% e 67,2%, respectivamente.

Em relação aos demais algoritmos, *Scaled Conjugate Gradient Backpropagation*, *Resilient Backpropagation* e *BFGS quasi-Newton Backpropagation* não tiveram variação considerável na aplicação dessa segunda camada na rede. E para *Gradient Descent Backpropagation*, a taxa de erro diminuiu de 86,3% para 77,7%, no entanto, ainda pode-se ser considerado inválido para essa aplicação, por manter a taxa de erro elevada.

4.3.2 Segunda Análise do Desempenho da Rede Neural

No processo de segunda análise dobrou-se o número de neurônios, e seguindo o mesmo procedimento da primeira análise, Sub-Secção 4.3.1.1, foi aplicado na rede neural *perceptron* de múltiplas camadas com uma camada oculta e a com a função de ativação sigmóide, definido 40 neurônios com 1000 épocas (número de interações), conseguindo assim, obter os dados apresentados na Tabela 3.

Tabela 3. Dados de Treinamento em MLP de Camada Única com 40 Neurônios e 1000 épocas

Algoritmo de treinamento	**Número de neurônios**	**Número de interações**	**Acertos/Erros**	**Performance**	**R**	**Tempo**
LMA	40	1000	73,1%/26,9%	0,0751	0,85060	0:09:02
BRA	40	1000	89,3%/10,7%	0,0382	0,92391	0:09:03
SCGA	40	1000	50,1%/49,9%	0,2557	0,42337	0:00:08
RBA	40	1000	48,7%/51,3%	0,2785	0,44278	0:00:13

BFGA	40	1000	52,5%/47,5%	0,2764	0,44653	0:00:57
GDA	40	1000	30,1%/69,9%	0,4740	0,23919	0:00:08

Fonte: Autoria própria, 2022.

Com retratado na Tabela 3, percebe-se que os algoritmos de treinamento *Levenberg-Marquardt* com 73,1% de acertos e com 26,9% de erros e no treinamento utilizando *Bayesian Regularization* com 89,3% de acertos e 10,7% de erro, continuam com o melhor desempenho comparado a primeira análise, subsecção 4.3.1.2. Não ocorreu uma alteração considerável nos números de acertos para *Levenberg-Marquardt* com aumento de neurônios, a regressão linear teve um melhor desempenho em relação à primeira análise. Em relação a *Bayesian Regularization* teve um desempenho elevado tanto na porcentagem de acertos, de 68,4%/31,6% para 89,3%/10,7%, na performance que era de 0,0988 para 0,0382, e consequentemente a regressão linear de 0,80610 para 0,92391.

Em relação aos demais, não ocorreu alterações significativas nos *Scaled Conjugate Gradient* e *Resilient Backpropagation* se manteram aproximadamente com os mesmos valores. Para os *BFGS quasi-Newton* e *Gradient Descent* tiveram um melhor desempenho em relação à primeira análise, entretanto, seus dados resultados ainda são inferiores em relação a *Levenberg-Marquardt* e *Bayesian Regularization*.

Prosseguindo com o mesmo procedimento, aplicou-se os algoritmos de treinamentos para a rede neural *perceptron* de múltiplas camadas com duas camadas ocultas e a com a função de ativação sigmóide, sendo agora com 40 neurônios com 1000 épocas (número de interações), mostrado na Tabela 4.

Tabela 4. Dados de Treinamento em MLP de Camada Dupla com 40 Neurônios e 1000 épocas

Algoritmo de treinamento	**Número de neurônios**	**Número de interações**	**Acertos/Erros**	**Performance**	**R**	**Tempo**
LMA	40	1000	77,6%/22,4%	0,0698	0,86079	0:09:43
BRA	40	1000	84,8%/15,2%	0,0921	0,82327	0:11:27
SCGA	40	1000	47,4%/52,6%	0,2785	0,44564	0:00:16
RBA	40	1000	56,3%/43,7%	0,2444	0,51061	0:00:37
BFGA	40	1000	51,1%/48,9%	0,2789	0,44189	0:00:55

GDA	40	1000	41,9%/58,1%	0,5730	0.4473	0:00:09

Fonte: Autoria própria, 2022.

Como mostrado na Tabela 4 e refletido com os dados da Tabela 3, como esperado os algoritmos *Levenberg-Marquardt* e *Bayesian Regularization* dispõem dos melhores resultados, obedecendo ao mesmo comportamento representado na primeira análise.

4.3.3 Terceira Análise do Desempenho da Rede Neural

Na última análise, aumentou em duas vezes o número de interações em relação às análises anteriores, sendo assim, 200 épocas com 20 neurônios e com a função de ativação sigmóide, alcançado os dados mostrados na Tabela 5.

Tabela 5. Dados de Treinamento em MLP de Camada Única com 20 Neurônios e 2000 épocas

Algoritmo de treinamento	**Número de neurônios**	**Número de interações**	**Acertos/Erros**	**Performance**	**R**	**Tempo**
LMA	20	2000	65,2%/34,8%	0.1013	0,79832	0:06:11
BRA	20	2000	71,4%/28,6%	0.0954	0,81052	0:04:04
SCGA	20	2000	45,6%/54,4%	0.2965	0,40612	0:00:21
RBA	20	2000	51,5%/48,5%	0.3016	0,39606	0:00:11
BFGA	20	2000	51,6%/48,4%	0.2254	0,55193	0:00:53
GDA	20	2000	31,1%/68,9%	0.4218	0,19354	0:00:11

Fonte: Autoria própria, 2022.

Observando a Tabela 5, percebe-se que não ocorreu alteração nos parâmetros de dados em comparação às análises anteriores, dessa forma, identifica-se que ao aumentar a quantidade de interações para esse uso de neurônios não ocorreu a melhoria desejada.

Tabela 6. Dados de Treinamento em MLP de Dupla Camada com 20 Neurônios e 2000 épocas

Algoritmo de treinamento	**Número de neurônios**	**Número de interações**	**Acertos/Erros**	**Performance**	**R**	**Tempo**

LMA	20	2000	59,2%/40,8%	0,1125	0,77792	0:06:30
BRA	20	2000	78,6%/21,4%	0,0984	0,81049	0:06:20
SCGA	20	2000	51,1%/48,9%	0,2771	0,45251	0:00:25
RBA	20	2000	56,4%/43,6%	0,3009	0,39718	0:00:32
BFGA	20	2000	54,8%/45,2%	0,1909	0,62184	0:00:55
GDA	20	2000	24,2%/75,8%	0,5296	0,06098	0:00:12

Fonte: Autoria própria, 2022.

Na Tabela 6, vê-se que o aumento de uma nova camada não mudou positivamente os resultados obtidos. Desse modo, levanta-se a hipótese que para ter uma melhoria nos dados de análises, precisa ter um aumento proporcional entre os números de neurônios como para o número de épocas.

4.3.4 Quarta Análise do Desempenho da Rede Neural

Como destacado na terceira análise na sub-secção 4.3.1.3, precisa ter um aumento tanto no número de interações como para o número de neurônios aplicados nas redes neurais. Assim, aplicou-se 40 neurônios para uma rede de 2000 épocas com uma camada, obtendo os dados representados na Tabela 7, e na Tabela 8 com duas camadas escondidas.

Tabela 7. Dados de Treinamento em MLP de Camada Única com 40 Neurônios e 2000 épocas

Algoritmo de treinamento	**Número de neurônios**	**Número de interações**	**Acertos/Erros**	**Performance**	**R**	**Tempo**
LMA	40	2000	80,7%/19,3%	0,0625	0,87609	0:17:48
BRA	40	2000	82,1%/17,9%	0,0426	0,9537	0:17:49
SCGA	40	2000	55,8%/44,2%	0,2835	0,43304	0:00:43:
RBA	40	2000	50,0%/50,0%	0,2975	0,40478	0:00:24
BFGA	40	2000	54,3%/45,7%	0,2607	0,48000	0:02:37
GDA	40	2000	23,9%/76,1%	0,4338	0,19325	0:00:24

Fonte: Autoria própria, 2022.

Dessa forma, observando a terceira análise e comparando os dados com a Tabela 7 e a Tabela 8, que além de aumentar os números de neurônios, também têm que elevar os números de interações para melhorar a saída e o desempenho da rede empregada. Assim, denota-se o resultado mais satisfatório para a rede com treinamento *Bayesian Regularization*, com 95,6% de acertos e 0,96890 com valor R de regressão linear, comprovando-se como o método mais eficiente no processo de demodulação digital para esse tipo de sinal.

Tabela 8. Dados de Treinamento em MLP de Dupla Camada com 40 Neurônios e 2000 épocas

Algoritmo de treinamento	**Número de neurônios**	**Número de interações**	**Acertos/Erros**	**Performance**	**R**	**Tempo**
LMA	40	2000	78,4%/21,6%	0,0685	0,86305	0:27:04
BRA	40	2000	95,6%/4,4%	0,0156	0,96890	0:27:16
SCGA	40	2000	50,2%/49,8%	0,2735	0,45269	0:00:34
RBA	40	2000	55,6%/44,4%	0,2572	0,48609	0:00:19
BFGA	40	2000	50,6%/49,4%	0,2197	0,56040	0:02:00
GDA	40	2000	37,1%/62,9%	0,4179	0,16690	0:00:18

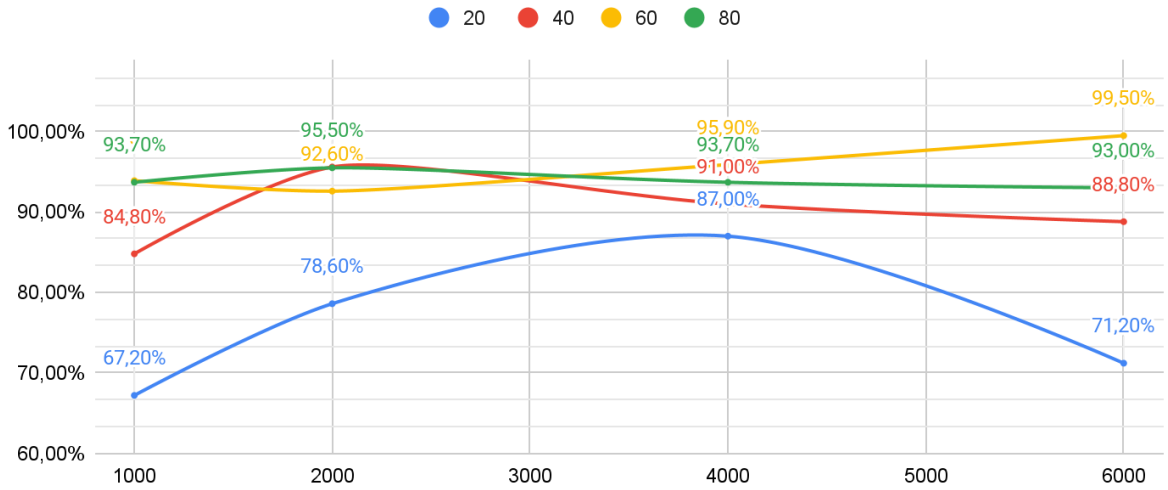
Fonte: Autoria própria, 2022.

4.3.5 Quinta Análise do Desempenho da Rede Neural

A partir das quatro análises anteriores, percebe-se que o *Bayesian Regularization* é que teve o melhor desempenho, justificando sua aplicação na rede MLP proposta. No entanto, com os resultados obtidos entende-se em elevar tanto o número de neurónios como os números de épocas, conseguindo assim, melhorar a taxa de acertos dos símbolos na demodulação do sinal.

De acordo com o Gráfico 1, observa-se o comportamento como analisado na subseção anterior, o número de acertos aumentando de acordo o número de neurônios e de épocas, podendo notar que ao variar muito o número de época para uma quantidade de neurônio, ocorre uma diminuição no número de acertos, isso fica mais visível para 20 neurônios (linha azul), que tem um incremento no número de acertos até 6000 época, onde diminui consideravelmente a taxa de acertos.

Gráfico 1. Taxa de Acertos por números de neurônios e iterações.



Fonte: Autoria própria, 2022.

Prosseguindo, teve-se o melhor resultado de taxa de acertos para a combinação de 60 neurônios e 6000 iterações, conseguindo uma taxa de 99,5%, entretanto, a melhor regressão linear e performance foi alcançada com 4000 iterações, mas com uma variação muito pequena, como mostrado na Tabela 9.

Tabela 9. Desempenho por números de neurônios e iterações.

neurônio /épocas	**1000**	**2000**	**4000**	**6000**
20	67,2%/32,6% R: 0,80037 P: 0.0997 T: 0:03:31	78,6%/21,4% R: 0,81049 P: 0,0984 T: 0:06:20	87%,0/13,0% R: 0,91341 P:0,0433 T:0:13:28	71,2%/28,7% R: 0,86866 P: 0,0662 T: 0:20:34
40	84,8%/15,2% R: 0,82327 P: 0,0921 T: 0:11:27	95,6%/4,4% R: 0,96890 P: 0,0156 T: 0:27:16	91,0%/9,0% R: 0,96863 P: 0,0158 T: 0:49,56	88,8%/11,2% R: 0,93780 P: 0,0314 T: 1:07:02
60	93,9%/6,1% R: 0,96223 P: 0,0190 T: 0:26:23	92,6%/7,4% R: 0,91892 P:0,0405 T:0:45:18	95,9%/4,1% R:0,97171 P: 0,0142 T: 1:48:43	99,5%/0,5% R: 0,96602 P: 0,0173 T: 2:16:15
80	93,7%/6,3% R: 0,96843	95,5%/4,5% R: 0,90692	93,7%/6,3% R: 0,95190	93,0%/7,0% R: 0,87515

neurônio /épocas	**1000**	**2000**	**4000**	**6000**
	P: 0,0160 T: 0:37:58	P: 0,0466 T: 1:18:53	P: 0,0241 T: 2:36:44	P: 0,0662 T: 4:01:03

Fonte: Autoria própria, 2022.

Observando ainda a Tabela 7, a partir que vai aumentando a complexidade da rede neural, ampliando o tempo gasto para de treinamento, isto ocorre porque *Bayesian Regularization* exige mais memória do que outros algoritmos de retropropagação (SARAJLIĆ et al., 2017).

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Neste trabalho foi realizado uma análise de desempenho de redes neurais do tipo *perceptron* de múltiplas camadas, propondo um estudo quantitativo no intuito de apresentar uma solução possível no processo de demodulação digital. Os resultados destacados podem auxiliar em trabalhos que deseje buscar por métodos que demodulação, como também possibilita a eles identificarem novas estratégias a serem exploradas.

Analisando os resultados apresentados na Subseção 4.3.1.5, pode-se considerar que a aplicação da rede MLP com algoritmo de treinamento *Bayesian Regularization*, conseguiu um resultado satisfatório de 99,5% acertos em um processo de demodulação 256-QAM, efetivando a estratégia utilizada neste trabalho. Ainda mostra que mesmo *Levenberg-Marquardt* não apresentar melhor desempenho entre os algoritmos de treinamento, conseguiu bons resultados, alcançando uma taxa de acertos de 83,4% com duas camadas escondidas com a proporção de 20 neurônios e 1000 épocas, possibilitando estudos futuros com novas estratégias.

É importante salientar que a MLP aqui apresentada não contém todas as questões e implicações desse tipo de rede na aplicação da demodulação digital, mas serve como suporte à melhoria desse tipo de trabalho.

Para trabalhos futuros, considera-se realizar novas estratégias para análise de desempenho desse tipo de rede, buscar variar outros parâmetros além dos números de neurônios e de épocas, como também aumentar o número de camadas escondidas da rede. Além disso, buscar aplicar estudos com essa mesma temática de análise para arquivos de outras categorias, como de áudio, imagem e vídeos, aprofundando mais o estudo da aplicabilidade das redes neurais no auxílio da demodulação de um sinal transmitido em um canal de ruído branco.

REFERÊNCIAS

BAGHIRLI, Orkhan. **Comparison of Lavenberg-Marquardt, Scaled Conjugate Gradient and Bayesian Regularization Backpropagation Algorithms for Multistep Ahead Wind Speed Forecasting Using Multilayer Perceptron Feedforward Neural Network**, Uppsala University, 2015, Dissertation of Master em Science, 2015.

BISHOP, C. M. **Neural Networks for Pattern Recognition**, Clarendon Press, Oxford, 1995.

CATANESE C.; AYASSI, R.; PINCEMIN, E.; JAOUËN, Y., **A Fully Connected Neural Network Approach to Mitigate Fiber Nonlinear Effects in 200G DP-16-QAM Transmission System**, 2020 22nd International Conference on Transparent Optical Networks (ICTON), pp. 1-4, 2020.

COUTINHO, A. J. C. R. **Identificação de Modulação Utilizando Redes Neurais do Tipo Auto-Organizáveis**. Universidade Federal Fluminense, 2013, 126f. Dissertação de Mestrado de Engenharia de Telecomunicações.

DAS, A. K.; PRATIHAR, D. K. **A Directional Crossover (DX) Operator For Real Parameter Optimization Using Genetic Algorithm**. Appl Intell, v. 49, p. 1841–1865, 2019.

FONTES, Aluisio Igor Rêgo. **Classificação Automática de Modulação Digital com uso de Correntropia para Ambientes de Rádio Cognitivo**. Universidade Federal do Rio Grande do Norte, 2012, Dissertação de Mestrado em Engenharia Elétrica e de Computação.

IBNKAHLA, M. **Neural Network Predistortion Technique for Digital Satellite Communications**. Computer Engineering, n. section 3, p. 3506–3509, 2000.

LATHI, B. P. **Sistemas de Comunicações Analógicos e Digitais Modernos**. 4ª edição. Editora LTC, 2012.

LIMA, Letícia Alves. **Análise de Desempenho de Arquiteturas de Redes Neurais no Processo de Equalização de Canais de Comunicação Dispersivos e Não Lineares**. 2018.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) - Universidade de Brasília, Brasília, 2018.

MACKAY, David J. C. **Bayesian Interpolation**, Neural Computation. v. 4, n. 3, p. 415-447, 1992.

MARQUARDT, D. W. **An Algorithm for Least-Squares Estimation of Nonlinear Parameters**, SIAM Journal on Applied Mathematics, v. 11, n. 2, p. 431-441, 1963.

MCCULLOCH, W.; PITTS, W. **A Logical Calculus of the Ideas Immanent in Nervous Activity**. Bulletin of Mathematical Biophysics, Vol. 5, No. 4, 115-133, December, 1943.

MOLLER, Martin Fodslette. **A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning**. *Neural Networks*, v. 6, p. 525–533, 1993.

NASCIMENTO, Ingrid; MENDES, Flavio; DIAS Marcus; SILVA, Andrey; KLAUTAU, Aldebaro. **Deep Learning in RAT and Modulation Classification with a New Radio Signals Dataset**, XXXVI SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS, 2018.

OWAKI, S.; NAKAMURA, M. **Simultaneous Compensation of Waveform Distortion Caused by Chromatic Dispersion and SPM Using a Three-Layer Neural-Network**, 2017 Opto-Electronics and Communications Conference (OECC) and Photonics Global Conference (PGC), p. 1-3, 2017.

RIEDMILLER, M.; BRAUN, H. **A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm**, Proceedings of the IEEE International Conference on Neural Networks, v. 1, p. 586–591, 1993.

ROSENBLATT, F. **The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain**. Psychological Review, v. 65, p. 386-408, 1958.

SANTOS, Guilherme Franco. QUINTANILHA, Samuel Gomes. **Otimização de Sistemas de Comunicação: Análise de Modulações Digitais**. Universidade de Brasília, 2015. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica), Brasília, 2015.

SARAJLIĆ, Dženana; ABDEL-ILAH, Layla; FOJNICA, Adnan; OSMANOVIĆ, Ahmed. **Prediction of the Size of Nanoparticles and Microspore Surface Area Using Artificial Neural Network**, Genetics & Applications, v. 1, n. 1. p. 65-70, 2017.

SILVA SEGUNDO, F. C. G. da. **Aplicação de Inteligência Computacional em Projetos de Superfície Seletiva em Frequência Multibanda e/ou Banda Larga para Aplicações Comerciais**. Universidade Federal do Rio Grande do Norte, 2018. Tese de Doutorado (Pós-Graduação em Engenharia Elétrica e de Computação), Rio Grande do Norte, 2018.

SHANNON, C. E. **A Mathematical Theory of Communication**. *Bell Syst. Tech. J.*, vol. 27, no. 4, pp. 623-656, Oct. 1948.

SHI, Y.; LIU, P.; YAN, D.; CHEN, Y.; LI, C.; LU, Z. **A Hopfield Neural Network Based Demodulator For APSK Signals**, 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), p. 2005-2009, 2020.

SILVA, I. N. da; SPATTI, D.; FLAUZINO, R. **Redes Neurais Artificiais para Engenharia e Ciências Aplicadas: Fundamentos Teóricos e Aspectos Práticos**, Artliber Editora Ltda, São Paulo, 2016.

SONG, Qing; WU, Yilei; SOH, Yeng Chai. **Robust Adaptive Gradient-Descent Training Algorithm for Recurrent Neural Networks in Discrete Time Domain**. IEEE Transactions on Neural networks, v. 19, n. 11, p. 1841-1853, 2008.

VIRAKTAMATH, S. V.; KOTI, M. V.; BAMAGOD, M. M., **Performance Analysis of Source Coding Techniques**, 2017 International Conference on Computing Methodologies and Communication (ICCMC), p. 689-692, 2017.

WU, Yong; WANG, Youqu;, LIU, Xiangwei; ZHONG, Jianmin. **Revision of the Observed Data Based on Neural Networks**. Proceedings of ICSSSM '05. 2005 International Conference on Services Systems and Services Management, v. 2, p. 1110-1112, 2005.

ZHANG, Yufeng; WANG, Zhugang; HUANG, Yonghui; REN, Jian; YIN, Yingzeng; LIU, Ying; PEDERSEN, Gert Frølund; SHEN, Min. **Deep Neural Network-Based Receiver for Next-Generation LEO Satellite Communications**, IEEE Access, v. 8, p. 222109-222116, 2020.

ZHOU, H. **Neural Network for Linear Distortion Compensation in Digital Communications**. Proceedings - 7th International Symposium on Signal Processing and Its Applications, ISSPA 2003, v. 1, p. 297–300, 2003.