



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Tecnologia

Vinícius Abrantes Pereira

**Autoencoder para geração de alfabeto de feixes
Laguerre-Gauss com Momento Angular Orbital**

Limeira
2020

Vinícius Abrantes Pereira

**Autoencoder para geração de alfabeto de feixes Laguerre-Gauss com
Momento Angular Orbital**

Dissertação apresentada à Faculdade de Tecnologia da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação, na área de Sistemas de Informação e Comunicação.

Orientador: Prof. Dr. Ulisses Martins Dias

Este exemplar corresponde à versão final da Dissertação defendida por Vinícius Abrantes Pereira e orientada pelo Prof. Dr. Ulisses Martins Dias.

Limeira
2020

FOLHA DE APROVAÇÃO

Abaixo se apresentam os membros da comissão julgadora da sessão pública de defesa de dissertação para o Título de Bacharel em Sistemas de Informação na área de concentração Sistemas de Informação e Comunicação, a que se submeteu o aluno Vinícius Abrantes Pereira, em 14 de janeiro de 2020 na Faculdade de Tecnologia – FT/UNICAMP, em Limeira/SP.

Prof. Dr. Ulisses Martins Dias
Presidente da Comissão Julgadora

Prof. Dr. Cristhof Johann Roosen Runge
FT/UNICAMP

Prof. Dr. Luis Augusto Angelotti Meira
FT/UNICAMP

Ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós Graduação da FT.

Agradecimentos

Ao meu orientador, Prof. Dr. Ulisses Martins Dias, por toda atenção e suporte durante o desenvolvimento do presente trabalho.

À minha namorada, por estar sempre ao meu lado, suportando meu jeito de ser, me ajudando e incentivando a alcançar minhas conquistas.

Aos meus pais, por terem me proporcionado as oportunidades que nunca tiveram.

Aos ótimos amigos que fiz na faculdade, os quais muito me ajudaram nessa jornada.

Aos diversos professores que passaram pela minha vida desde o ensino fundamental, sem vocês eu não teria chegado até aqui, muito menos onde almejo chegar.

Resumo

Com a pandemia causada pelo COVID-19, tarefas cotidianas realizadas predominantemente de forma presencial (e.g. estudo e trabalho) passaram a ser realizadas parcialmente ou integralmente online. Essas mudanças contribuíram para o aumento da demanda de largura de banda, recurso limitado que crescia em ritmo acelerado devido ao surgimento de serviços como streaming de vídeos, internet das coisas e as redes 5G. Neste cenário, estudos que buscam melhorar a transferência de dados e eficiência dos sistemas de comunicação mostram-se relevantes. Dentre as propriedades físicas dos feixes de luz, uma delas os possibilita carregar momento angular orbital, cujo valor é denominado carga topológica e sua ortogonalidade inerente possibilita que feixes com diferentes cargas topológicas sejam superpostos e individualmente identificados mesmo após serem transmitidos através da atmosfera, tornando atraente sua utilização para a transmissão de informações. Existem diferentes modelagens para feixes com momento angular orbital, sendo os feixes com modo Laguerre-Gauss um dos mais conhecidos. Este trabalho busca propor e desenvolver um autoencoder que simule a transmissão de feixes com momento angular orbital superpostos de modo Laguerre-Gauss multiplexados, visando obter um alfabeto de transmissão otimizado levando em conta a degradação do feixe transmitido causada pela turbulência atmosférica.

Abstract

Since the pandemic caused by COVID-19, everyday tasks performed predominantly in person (e. g. study and work) started to be done partially or entirely online. These changes contributed to increase the demand for bandwidth, a limited resource that was growing at a fast pace due to the emergence of services such as video streaming, internet of things and 5G networks. In this scenario, studies that seeks to improve data transfer and efficiency of communication systems are relevant. Among the physical properties of light beams, there is one that allows them to carry orbital angular momentum (OAM), which is defined as the topological charge, and its inherent orthogonality permit beams with different topological charges to be superposed and recognized after being transmitted through the atmosphere, making it attractive to be used for information transmission. There are different modes for OAM beams, the Laguerre-Gaussian (LG) beam is one of the most well-known. This work propose and develop an autoencoder that simulates the transmission and receiving of multiplexed Laguerre-Gaussian beams carrying superposed modes of orbital angular momentum, aiming to obtain an optimized transmission alphabet based on the degradation of the transmitted beam caused by atmospheric turbulence.

Lista de Figuras

| | | |
|-----|---|----|
| 2.1 | Propagação do feixe com polarização linear | 14 |
| 2.2 | Frentes de fase helicoidal | 14 |
| 2.3 | Geração de feixe LG com a utilização de holograma | 15 |
| 2.4 | Estrutura de um LTU | 17 |
| 2.5 | Estrutura de um perceptron | 18 |
| 2.6 | Resultado da convolução de dois filtros distintos | 19 |
| 2.7 | Camada de subamostragem utilizando <i>max-pooling</i> | 19 |
| 2.8 | Arquitetura de um autoencoder | 20 |
| 2.9 | Representação das arquiteturas espelhadas do encoder e decoder | 20 |
| 3.1 | Representação de um sistema de comunicação simples, composto por um transmissor, canal e receptor | 22 |
| 4.1 | Arquitetura proposta para o autoencoder | 25 |
| 4.2 | Aplicação dos diferentes níveis de turbulência | 26 |
| 4.3 | Arquitetura do autoencoder desenvolvido | 27 |
| 4.4 | Imagens representando o perfil transversal dos resultados obtidos | 29 |

Lista de Tabelas

| | | |
|-----|--|----|
| 4.1 | Conjunto base para o treinamento do modelo | 24 |
| 4.2 | Valores preditos pelo encoder para cada símbolo de entrada | 28 |

Sumário

| | | |
|----------|--|-----------|
| 1 | Introdução | 10 |
| 1.1 | Objetivo | 11 |
| 1.2 | Organização do trabalho | 11 |
| 2 | Referencial teórico | 13 |
| 2.1 | Momento angular orbital | 13 |
| 2.2 | Turbulência atmosférica | 15 |
| 2.3 | Aprendizado de máquina | 16 |
| 2.3.1 | Redes neurais convolucionais | 18 |
| 2.3.2 | Autoencoder | 20 |
| 3 | Levantamento bibliográfico | 21 |
| 3.1 | Sistemas de comunicação OAM | 21 |
| 3.2 | Autoencoder como otimizador de sistemas de comunicação | 22 |
| 4 | Desenvolvimento | 23 |
| 4.1 | Ambiente de criação do modelo | 23 |
| 4.2 | Conjunto de dados | 24 |
| 4.3 | Arquitetura proposta para o modelo | 25 |
| 4.4 | Geração de feixes OAM-LG | 25 |
| 4.5 | Simulação da turbulência atmosférica | 26 |
| 4.6 | Modelo desenvolvido | 26 |
| 4.7 | Análise dos resultados | 28 |
| 5 | Conclusões | 30 |
| | Referências bibliográficas | 31 |

Capítulo 1

Introdução

As diversas tecnologias e serviços que emergiram nos últimos anos (e. g. internet das coisas e serviços de compartilhamento de vídeos em tempo real) resultaram em um aumento significativo no volume de dados transmitidos e, por se tratar de um recurso escasso, a largura de banda torna-se um limitante dos sistemas de comunicação (ROUTRAY et al., 2019). Com o constante e acelerado crescimento de dados trafegados, estudos que buscam novas técnicas de multiplexação visando o aumento da capacidade de transmissão e eficiência espectral dos sistemas de comunicação tornam-se relevantes (GUO et al., 2018).

O momento angular orbital (OAM), revelado inicialmente por Allen et al. (1992), trata-se de uma propriedade física inerente aos feixes de luz, onde os fótons possuem polarização circular que resultam em uma frente de onda helicoidal. Tal propriedade pode ser quantificada em função de sua carga topológica, indicada por ℓ . Existem diferentes modelagens de feixes com momento angular orbital, sendo os feixes do tipo Laguerre-Gauss (LG) um dos mais conhecidos. Em princípio, feixes OAM com diferentes cargas topológicas são mutuamente ortogonais e podem ser individualmente identificados mesmo após a propagação, possibilitando a multiplexação de diferentes modos OAM na transmissão de dados em sistemas de comunicação óptica no espaço livre (FSO) (GUO et al., 2018).

O conjunto de inteiros contendo todos os diferentes valores de ℓ a serem utilizados para o envio de informação determina o alfabeto do sistema. O volume de informação que pode ser trafegada nesse tipo de sistema é limitado pela quantidade de canais que, no caso de sistemas OAM, refere-se à quantidade de feixes multiplexados. A princípio, a ortogonalidade inerente a feixes OAM estabelece que o limite de feixes multiplexados tenda ao infinito, mas a distorção

causada por problemas característicos desses sistemas (e. g. turbulência atmosférica) age como fator limitante.

1.1 Objetivo

Em estudo recente, O'Shea e Hoydis (2017) propõem a inovadora ideia de interpretar um sistema de comunicação ponta a ponta com a perspectiva de um autoencoder. Nesse estudo, os autores afirmam que, de forma simplificada, um sistema de comunicação é composto por um transmissor, canal e receptor, onde o transmissor utiliza o canal para enviar sinais que devem ser recebidos pelo receptor e que, geralmente, o hardware do transmissor impõe restrições ao sinal que deve ser enviado (e. g. restrição de energia ou amplitude).

Os autores alegam ainda que, em sua maioria, algoritmos de processamento de sinais têm fundamentos estatísticos e são, muitas vezes, comprovados teoricamente através de modelos matemáticos que buscam descrever a realidade mas que, na prática, os sistemas contêm imperfeições e não linearidades que somente podem ser aprendidas através da utilização de redes neurais. Dessa forma, os autores apontam que o modelo de sistema de comunicação acima descrito pode ser visto como um autoencoder.

O presente trabalho apropria-se da proposta do trabalho citado para a concepção e desenvolvimento de um autoencoder que simule um sistema óptico de comunicação via feixes de luz contendo Momento Angular Orbital de modo Laguerre-Gauss (OAM-LG) multiplexados, com o intuito de gerar um alfabeto de feixes OAM-LG otimizado a partir do treinamento da rede, levantando a hipótese de que tal alfabeto seria mais eficiente do que um alfabeto composto por valores definidos de forma arbitrária.

1.2 Organização do trabalho

O trabalho é composto por um total de cinco capítulos, com o objetivo final de apresentar o alfabeto de feixes OAM-LG otimizado gerado a partir do desenvolvimento de um autoencoder para a simulação do envio de mensagem de um sistema de comunicação FSO.

No Capítulo 2 encontra-se o referencial teórico que embasa o trabalho desenvolvido, contemplando os seguintes tópicos: feixes OAM, sistemas de comunicação FSO e suas limitações, aprendizado de máquina e autoencoder.

O Capítulo 3 abrange os trabalhos precursores das tecnologias e técnicas utilizadas no desenvolvimento do trabalho.

O Capítulo 4 abrange os materiais e métodos utilizados para a concepção do projeto, descrevendo o ambiente de desenvolvimento utilizado, o conjunto de dados de treinamento e a proposta inicial do modelo. Em seguida, o capítulo apresenta as atividades realizadas no desenvolvimento do modelo proposto, suas limitações e desafios enfrentados e, posteriormente, apresenta a análise dos resultados obtidos.

Por fim, o Capítulo 5 sintetiza as contribuições que podem ser alcançadas com o desenvolvimento do modelo proposto, os próximos passos para dar continuidade ao projeto e hipóteses futuras a serem verificadas.

Capítulo 2

Referencial teórico

Sistema de comunicação óptica no espaço livre (FSO) é uma tecnologia de comunicação na qual dados são transmitidos através da propagação da luz no espaço livre (MALIK; SINGH, 2015). Nesse tipo de sistema, a comunicação ocorre com base na quantificação de alguma propriedade inerente aos feixes de luz como, por exemplo, diferentes frequências, amplitudes ou cargas topológicas que os mesmos podem conter.

Um feixe de luz pode ser descrito como um fenômeno de onda eletromagnética, no qual tem-se a variação de seus fótons através de seu eixo de propagação Z devido à oscilação transversal entre o campo elétrico e magnético gerado por seus átomos. Na sua forma mais simples, um feixe se encontra linearmente polarizado, indicado pelo seu campo elétrico, que oscila em direção linear perpendicular ao eixo de propagação, enquanto seu campo magnético oscila perpendicularmente ao campo elétrico, como ilustra a Figura 2.1.

Em contrapartida, o momento angular orbital, descoberto inicialmente por Allen et al. (1992), se dá por um perfil de polarização de onda distinto, denominado polarização circular, no qual a frente de onda apresenta comportamento helicoidal ao redor do eixo de propagação.

2.1 Momento angular orbital

Estudado inicialmente por Allen et al. (1992), o momento angular orbital trata-se de uma propriedade física associada a feixes de luz com polarização circular. Em seu estudo, feixes contendo momento angular orbital foram obtidos a partir da difração causada por um prato de onda utilizado para alterar o padrão do feixe, gerando uma oscilação com formato helicoidal ao redor do eixo de propagação, denominado vórtice óptico.

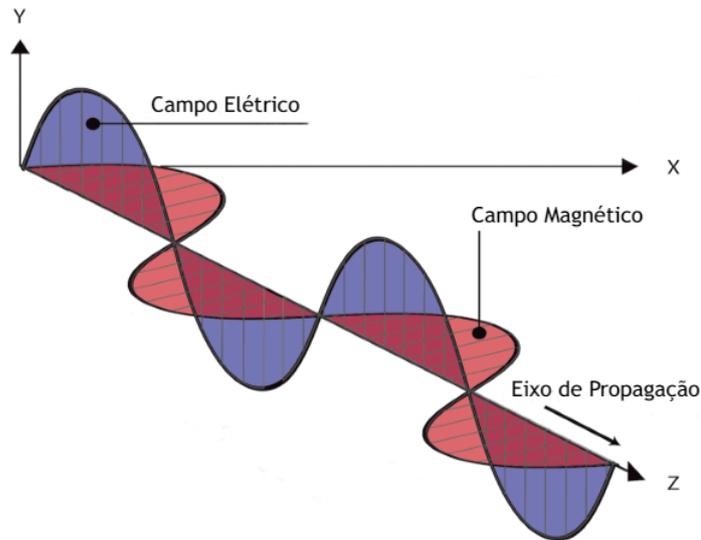


Figura 2.1: Propagação do feixe com polarização linear.

Fonte: Adaptado de Flickenger (2013).

De acordo com o estudo, feixes que carregam momento angular orbital possuem componente de fase azimutal determinado por $\exp(-i\ell\phi)$, onde o ângulo ϕ é a coordenada azimutal da seção transversal do feixe, ℓ é um número inteiro que indica a carga topológica do feixe e $|\ell|$ representa a quantidade de entrelaçamentos helicoidais que ocorrem ao redor do eixo de propagação. Para $\ell > 0$, tem-se uma rotação no sentido horário, $\ell < 0$ no sentido anti-horário e $\ell = 0$ indica a ausência de momento angular orbital, conforme ilustra a Figura 2.2.

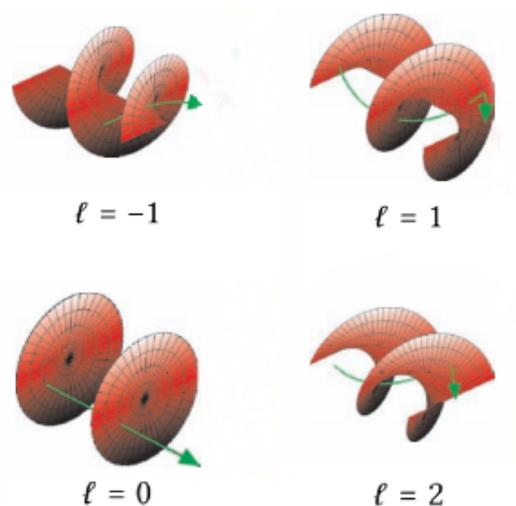


Figura 2.2: Frentes de fase helicoidal para diferentes valores de ℓ .

Fonte: Adaptado de Padgett, Courtial e Allen (2004).

Existem diferentes modelagens de feixes com momento angular orbital, sendo a modelagem de feixes do tipo Laguerre-Gauss a mais comum. Feixes LG podem ser obtidos ao incidir um feixe de onda plana em hologramas calculados especificamente para gerar, através da difração, a interferência necessária para obter o feixe com a carga topológica desejada (PADGETT; COURTIAL; ALLEN, 2004). A Figura 2.3 demonstra a obtenção de um feixe LG com $\ell = \pm 3$.

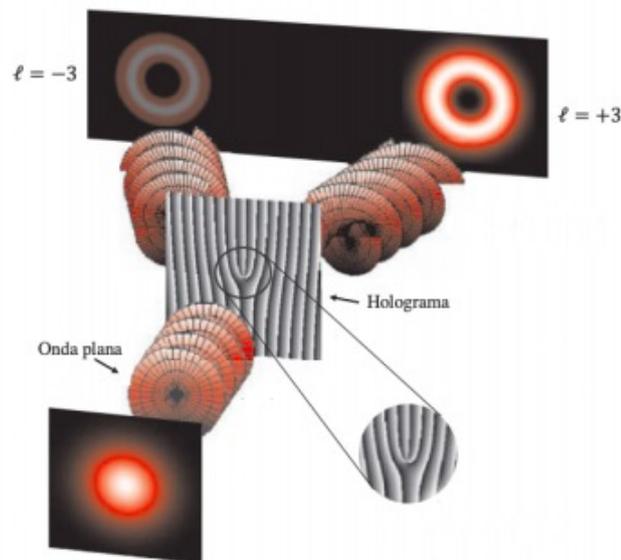


Figura 2.3: Geração de feixe LG com a utilização de holograma.

Fonte: Adaptado de Padgett, Courtial e Allen (2004).

2.2 Turbulência atmosférica

A turbulência atmosférica é um dos problemas enfrentados por sistemas de comunicação FSO, onde a flutuação aleatória do índice de refração atmosférica resulta na difração e degradação do sinal transmitido (LI et al., 2018). A teoria mais aceita para descrever os efeitos da turbulência foi proposta inicialmente por A. N. Kolmogorov.

Li et al. (2018) afirmam que a escala de Kolmogorov pode relacionar a flutuação de temperatura com a flutuação do índice de refração, que pode ser expresso pela Equação 2.1, que descreve uma função de distribuição de probabilidade, onde n indica o índice de refração, \mathbf{r}_1 e \mathbf{r}_2 representam dois pontos no espaço e $\langle \cdot \rangle$ representa o conjunto médio.

$$D_n(\mathbf{r}_1, \mathbf{r}_2) = \langle |n(\mathbf{r}_1) - n(\mathbf{r}_2)|^2 \rangle \quad (2.1)$$

De acordo com a teoria de Kolmogorov, D_n pode ser definido como $D_n(\Delta r) = C_n^2 \Delta r^{2/3}$, onde $\Delta r = |\mathbf{r}_1 - \mathbf{r}_2|$ e C_n^2 é conhecido como constante de estrutura atmosférica, que caracteriza a flutuação de intensidade.

Lane, Glindemann e Dainty (1992) propõem a Equação 2.2 como função de aproximação da densidade espectral da potência de Kolmogorov, onde r_0 é o parâmetro de Fried e D representa o comprimento de cada dimensão de abertura bidimensional quadrada do feixe recebido.

$$\Phi(k_x, k_y) = 0.023(2D/r_0)^{5/3}(k_x^2 + k_y^2)^{-11/3} \quad (2.2)$$

2.3 Aprendizado de máquina

De acordo com Géron (2019), aprendizado de máquina é a ciência e arte de programar computadores para que eles possam aprender através de dados. Em outras palavras, o aprendizado de máquina permite a criação de programas (i. e. modelos) capazes de aprender e realizar ações sem a necessidade de explicitar todas as possíveis regras, o que torna o aprendizado de máquina especialmente útil para problemas com soluções complexas ou desconhecidas. Existem diversos problemas que podem ser resolvidos utilizando técnicas de aprendizado de máquina, como classificação de imagens, previsão de valores futuros, reconhecimento de voz ou escrita, criação de sistemas capazes de atender pessoas ou oferecer sugestões com base na preferência de clientes com perfil semelhante.

Os sistemas de aprendizado de máquina podem ser classificados em quatro categorias, de acordo com a quantidade e tipo de supervisão requerida durante o treinamento, que são: aprendizado supervisionado, aprendizado não supervisionado, aprendizado semi supervisionado e aprendizado por reforço. Para o presente trabalho, é relevante abordar o aprendizado supervisionado.

Classificadores são o tipo de problema padrão resolvidos através do aprendizado supervisionado. Neles, os dados utilizados no treinamento necessitam de um rótulo, que indica a solução desejada. Um clássico exemplo seria classificar números escritos à mão, onde o conjunto de dados deve conter, para cada imagem, o rótulo informando o número correto que o sistema deve ser capaz de classificar.

Uma das principais etapas na criação de um modelo é a fase de treinamento, cujo objetivo é estimar os parâmetros que o tornem capaz de prever os valores corretos a fim de obter

uma boa performance no uso real (BOROVICKA et al., 2012). Para alcançar uma predição de erro futuro confiável, é essencial que a base de dados gerada na fase de pré processamento dos dados seja balanceada, livre de ruídos e de informações irrelevantes e que seus valores sejam normalizados.

Uma das técnicas para a criação de classificadores é através da utilização de redes neurais artificiais (ANNs). Essas redes são compostas por neurônios, denominados *perceptrons*, que representam um modelo matemático inspirado na biologia humana.

O perceptron, uma das arquiteturas mais simples de ANN foi inventado em 1957 por Frank Rosenblatt, baseado em um modelo de neurônio mais simples, denominado *linear threshold unit (LTU)* ((GÉRON, 2019). A Figura 2.4 demonstra a arquitetura de um LTU, onde cada valor do vetor de entrada $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ é conectado ao seu respectivo peso do vetor $\mathbf{W} = \{w_1, w_2, \dots, w_n\}$. O LTU computa a somatória $z = \sum_{i=1}^n x_i w_i$ e então aplica a função de ativação $y = \varphi(z)$, que representa a saída do neurônio.

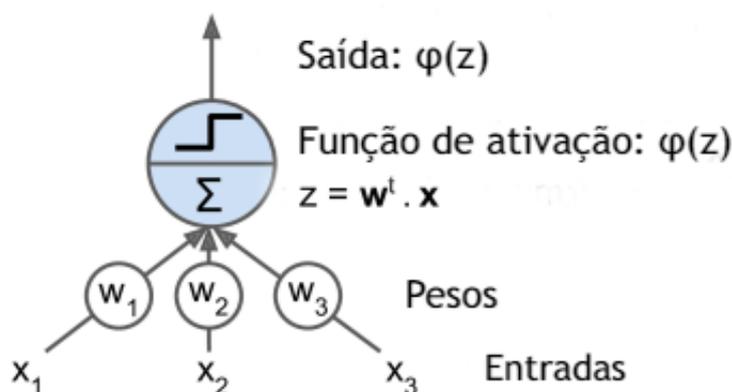


Figura 2.4: Estrutura de um LTU.

Fonte: Adaptado de Géron (2019).

Um perceptron, por sua vez, é composto por uma única camada de LTUs, com todos os neurônios conectados a todas as entradas. Além disso, um perceptron contém um parâmetro adicional denominado viés b . A Figura 2.5 ilustra a estrutura de um perceptron composto por duas entradas e três saídas. A partir dessas estruturas simples, é possível criar ANNs muito mais complexas, com diversas camadas cujas entradas \mathbf{X} são compostas pelas saídas da camada anterior.

Um clássico problema enfrentado por ANNs é o denominado *sobreajuste* dos parâmetros (do inglês, *overfit*), onde a rede ajusta os pesos de forma equivocada para memorizar as instâncias

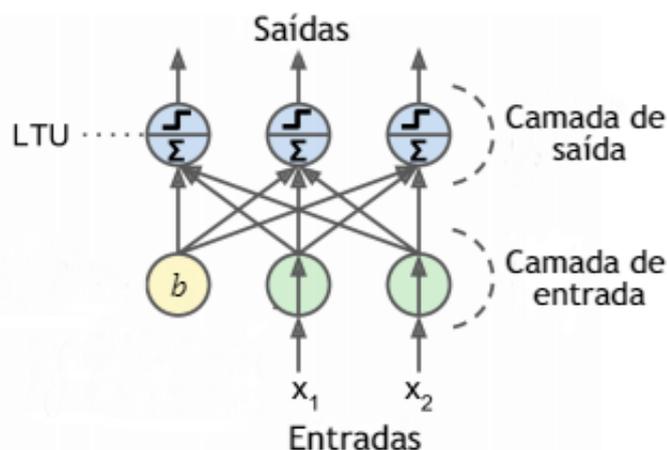


Figura 2.5: Estrutura de um perceptron.

Fonte: Adaptado de Géron (2019).

de treinamento, alcançando uma ótima acurácia para estes dados mas incapaz de generalizar e classificar corretamente novas instâncias.

2.3.1 Redes neurais convolucionais

De acordo com Géron (2019), as redes neurais convolucionais (CNNs) emergiram de estudos relacionados ao funcionamento do córtex visual do cérebro, os quais serviram de inspiração para o modelo denominado *neocognitron* proposto por Fukushima (1980), que, por sua vez, evoluiu para se tornar o modelo de CNN, um tipo de ANN especialmente eficiente para tarefas que requerem reconhecimento de imagens.

A arquitetura de uma CNN contém dois tipos de camadas especiais, que são as camadas de convoluções e as camadas de subamostragem. Uma camada convolucional serve para reconhecer padrões e extrair características de sua entrada através da aplicação dos denominados *filtros*, enquanto a camada de subamostragem serve para, como o próprio nome sugere, realizar uma subamostragem dos dados de entrada, reduzindo sua dimensionalidade.

Uma camada convolucional pode conter diversos filtros, que são representados por matrizes de pesos que se convolucionam com a entrada. No caso de reconhecimento de imagens, filtros podem simbolizar a detecção de características que vão desde bordas e círculos até a detecção de formatos complexos, como diferentes objetos ou texturas.

A Figura 2.6 demonstra o resultado da convolução de dois filtros distintos com dimensões 7×7 a uma mesma imagem, onde o primeiro filtro é composto por valores 1 na coluna central e 0 nas colunas laterais e o segundo contém valores 1 na linha central e 0 nas demais linhas.

No resultado da convolução pode-se notar que o primeiro filtro realça as partes com linhas verticais, enquanto o segundo captura um padrão horizontal.

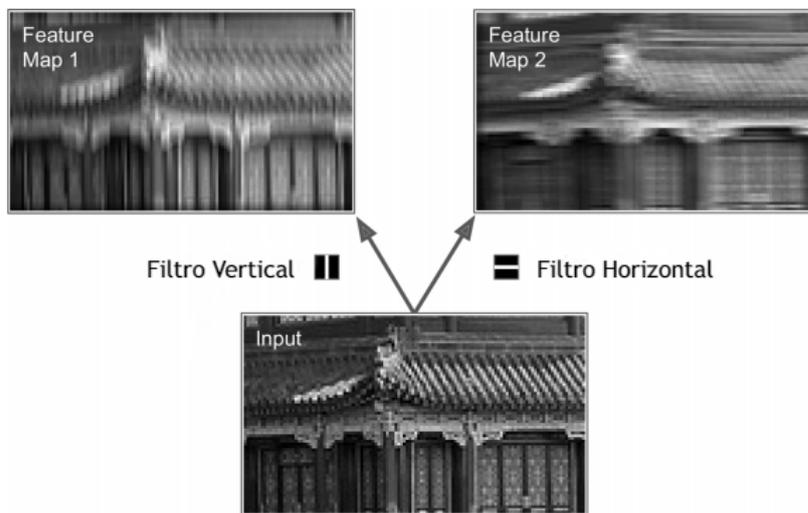


Figura 2.6: Resultado da convolução de dois filtros distintos.

Fonte: Adaptado de Géron (2019).

As camadas de subamostragem contribuem para a redução da demanda computacional, número de parâmetros a serem treinados e o risco de *overfitting*. A subamostragem é o resultado da agregação dos valores obtidos a partir do deslocamento de um *kernel* nos dados de entrada. Exemplos de função de agregação são a utilização de valor máximo (*max-pooling*) e utilização do valor médio (*average-pooling*). A Figura 2.7 demonstra o resultado da aplicação de subamostragem com *max-pooling*. Nela, é definido um kernel no formato 2x2, que percorre a matriz de entrada e utiliza o maior valor para gerar a imagem resultante.

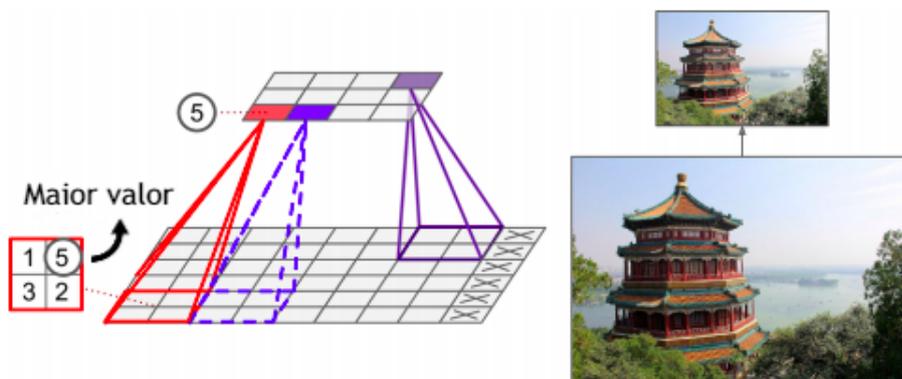


Figura 2.7: Camada de subamostragem utilizando *max-pooling*.

Fonte: Adaptado de Géron (2019).

2.3.2 Autoencoder

Autoencoders são ANNs capazes de aprender uma representação eficiente dos dados de entrada, denominada *codificação*, de forma não supervisionada (i. e. utilizam base de dados não rotulada) através da utilização da mesma base de dados como entrada e saída esperada da rede (GÉRON, 2019). Geralmente, a codificação dos dados consiste em uma representação intermediária com dimensão reduzida, exigindo que o autoencoder seja capaz de extrair as características mais relevantes para permitir a reconstrução dos dados codificados.

A Figura 2.8 representa a arquitetura geral de um autoencoder. O objetivo da rede é, dado o vetor de entrada $X = \{x_1, x_2, \dots, x_n\}$, conseguir codificá-lo em uma menor dimensão e ser capaz de reconstruir a entrada na saída da rede, representado pelo vetor estimado $\hat{X} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$.

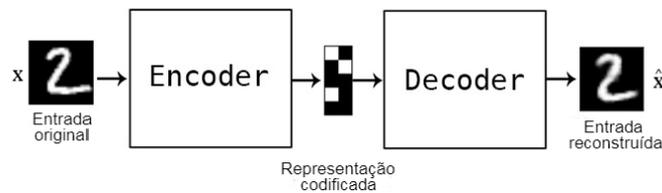


Figura 2.8: Arquitetura de um autoencoder.

Fonte: Adaptado de Francois (2020).

O conjunto de camadas que vão desde a entrada inicial até a camada que representa a codificação dos dados define uma rede neural interna denominada *encoder*, e o conjunto de camadas composto pelas demais camadas a partir da camada intermediária determina o *decoder*. Por vezes, o decoder é composto pela arquitetura inversa do encoder, conforme ilustra a Figura 2.9.

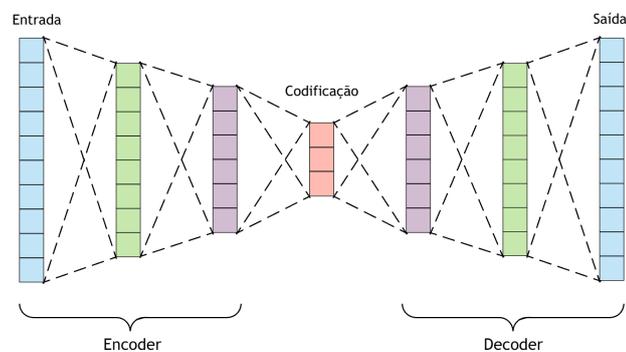


Figura 2.9: Representação das arquiteturas espelhadas do encoder e decoder.

Fonte: Adaptado de Dertat (2020).

Capítulo 3

Levantamento bibliográfico

Desde a descoberta de que feixes de luz podem conter momento angular orbital (ALLEN et al., 1992), surgiram diversos estudos que visam encontrar formas eficientes de utilizá-los para comunicação óptica em espaço livre.

Neste capítulo são apresentados alguns trabalhos relacionados ao tema que foram importantes para o desenvolvimento do projeto. A Seção 3.1 contém dois trabalhos relacionados a sistemas de comunicação utilizando feixes com momento angular orbital para a transmissão de informação, e a Seção 3.2 descreve o estudo que apresentou o conceito de modelo no qual o presente trabalho é embasado.

3.1 Sistemas de comunicação OAM

O conjunto que define os modos OAM a serem utilizados constituem o *alfabeto de modulação* do sistema. Runge et al. (2019) e Freitas et al. (2020) realizaram a simulação da transmissão de feixes OAM do tipo Laguerre Gauss pelo espaço livre recebidos por uma câmera CCD considerando o efeito da turbulência atmosférica, onde as imagens simuladas do perfil de intensidade do feixe transmitido foram utilizadas no treinamento de uma CNN. Os estudos utilizaram níveis de turbulência fraca, moderada e forte para quatro cenários distintos. Entre os diferentes cenários, haviam alfabetos formados somente pela superposições simples de feixes OAM $\pm\ell$ e alfabetos formados pela multiplexação de dois feixes OAM ($\pm\ell_1$ e $\pm\ell_2$). Os resultados obtidos demonstram que a utilização de alfabetos compostos pela multiplexação de dois modos OAM apresentam desempenho similar a alfabetos compostos pela superposição simples de modos $\pm\ell$ nas simulações de turbulência fraca e moderada, o que

demonstra uma vantagem na utilização de alfabetos formados pela multiplexação de feixes, uma vez que esta técnica possibilita obter alfabetos com maior cardinalidade sem ter o desempenho comprometido.

3.2 Autoencoder como otimizador de sistemas de comunicação

Diversos estudos sobre sistemas de comunicação buscam encontrar formas mais eficientes de se transmitir dados. Grande parte desses estudos procuram encontrar algoritmos que possam ser aplicados no processamento de sinais buscando descrever o comportamento do mundo real de forma precisa. O problema é que, frequentemente, os sistemas sofrem de aleatoriedades que não podem ser descritas de forma matemática, o que gera a necessidade de se criar compensações de hardware, aumentando a complexidade do sistema.

O'Shea e Hoydis (2017) introduzem a ideia de representar um sistema de comunicação através de um autoencoder, visando otimizar a comunicação entre o transmissor e o receptor e demonstram que esse tipo de abordagem pode alcançar desempenho equivalente ou superior aos sistemas baseados nos modelos tradicionais.

A Figura 3.1 demonstra a estrutura geral da rede proposta, onde o transmissor deseja enviar o sinal s , que passa por diversas transformações e restrições físicas do modelo, representado por x . O sinal x é então passado pelo canal, onde é somado um ruído referente à distorção causada pela transmissão no meio físico, gerando o sinal y recebido pelo receptor. Por fim, a tarefa do receptor é reconstruir e estimar o sinal s inicialmente enviado.

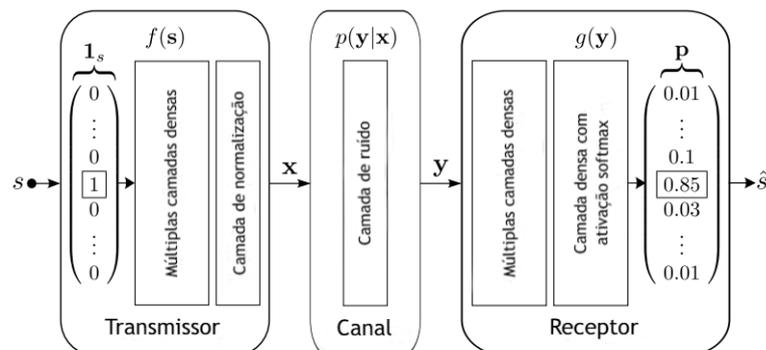


Figura 3.1: Representação de um sistema de comunicação simples, composto por um transmissor, canal e receptor.

Fonte: O'Shea e Hoydis (2017).

Capítulo 4

Desenvolvimento

O presente trabalho se baseia na ideia apresentada por O’Shea e Hoydis (2017) para a criação de um autoencoder que simule um sistema de comunicação de feixes com momento angular orbital (OAM) de modos Laguerre Gauss (LG) multiplexados, com base na hipótese de que o alfabeto utilizado pelo autoencoder, otimizado para alcançar uma performance satisfatória no ambiente simulado, seria mais eficiente que um alfabeto composto por símbolos definidos de maneira arbitrária.

Na Seção 4.1 encontram-se as tecnologias utilizadas na construção do projeto, englobando a linguagem e o ambiente de desenvolvimento. A Seção 4.2 define o conjunto de dados para treinamento do modelo, a Seção 4.3 descreve uma visão geral de sua arquitetura, a Seção 4.4 descreve como os feixes com modos OAM serão simulados e a Seção 4.5 contém a equação que simula a distorção atmosférica causada ao transmitir feixes OAM em espaço livre. Por fim, a Seção 4.6 apresenta o modelo desenvolvido e suas limitações, enquanto a Seção 4.7 apresenta os resultados obtidos com o autoencoder desenvolvido.

O conjunto de dados e a arquitetura apresentada são baseados na criação de um autoencoder para a geração de um alfabeto com cardinalidade 16, cujos símbolos representam a multiplexação de dois feixes LG com momento angular orbital $\pm\ell$ distintos superpostos, com $1 \leq |\ell| \leq 16$.

4.1 Ambiente de criação do modelo

O modelo foi desenvolvido utilizando o Google Colaboratory (conhecido popularmente como *Colab*), um ambiente de desenvolvimento da Google que possibilita a escrita e execução de

código através do navegador utilizando o sistema Google Cloud. Nele, é possível escrever código em python e fazer uso de bibliotecas de aprendizado de máquina e tecnologias de processamento como graphics processing unit (GPU) e tensor processing unit (TPU) sem a necessidade de configurações complexas. O código da aplicação foi desenvolvido em python com a utilização do Keras, uma interface de programação de aplicação de alto nível para desenvolvimento de modelos de aprendizado de máquina.

4.2 Conjunto de dados

O conjunto de dados que alimenta a rede depende da cardinalidade k do alfabeto que o modelo deve simular, formado por vetores contendo a representação binária do antecessor de cada um dos números naturais contidos no intervalo $[1, k]$. Dado que o presente trabalho busca encontrar um alfabeto de cardinalidade $k = 16$, o conjunto de dados é formado pelos vetores da Tabela 4.1, a qual lista, para cada natural no intervalo $[1, 16]$, o vetor composto pela representação binária de seu antecessor.

Tabela 4.1: Conjunto base para o treinamento do modelo.

| Símbolo | Representação binária |
|----------------|------------------------------|
| 1 | [0, 0, 0, 0] |
| 2 | [0, 0, 0, 1] |
| 3 | [0, 0, 1, 0] |
| 4 | [0, 0, 1, 1] |
| 5 | [0, 1, 0, 0] |
| 6 | [0, 1, 0, 1] |
| 7 | [0, 1, 1, 0] |
| 8 | [0, 1, 1, 1] |
| 9 | [1, 0, 0, 0] |
| 10 | [1, 0, 0, 1] |
| 11 | [1, 0, 1, 0] |
| 12 | [1, 0, 1, 1] |
| 13 | [1, 1, 0, 0] |
| 14 | [1, 1, 0, 1] |
| 15 | [1, 1, 1, 0] |
| 16 | [1, 1, 1, 1] |

4.3 Arquitetura proposta para o modelo

A Figura 4.1 apresenta a arquitetura proposta para o autoencoder, que pode ser dividida em três principais partes: (I) encoder, que representa o transmissor, (II) a camada onde é calculada a imagem da seção transversal do feixe transmitido, que representa o canal aplicando turbulência atmosférica em forma de ruído na imagem, e (III) o decoder, que representa o receptor através de uma rede neural convolucional que tenta reconstruir o vetor de entrada.

O encoder é composto por três camadas, que recebem como entrada a representação binária do símbolo e gera como saída dois números, denominados d_1 e d_2 , que serão utilizados para gerar a imagem do feixe transmitido.

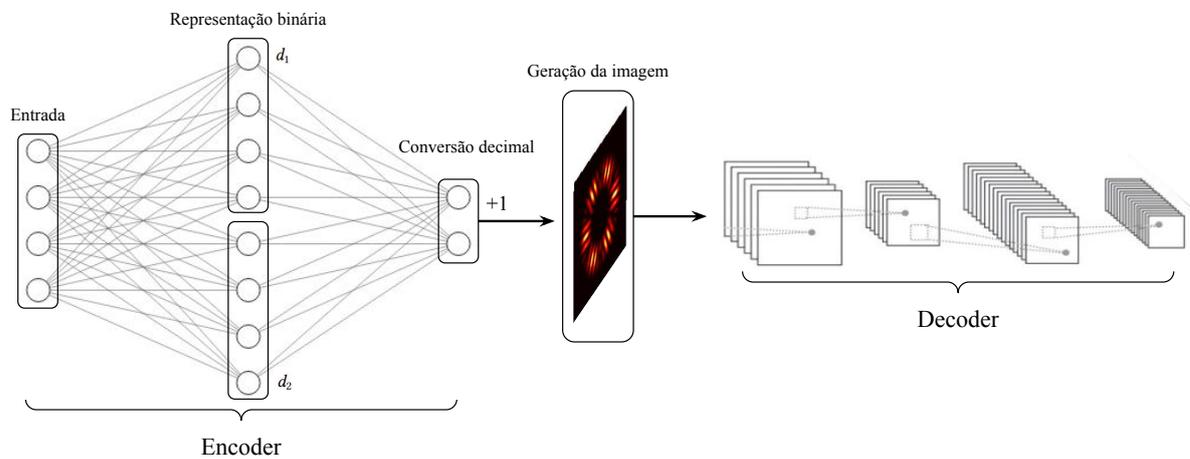


Figura 4.1: Arquitetura proposta para o autoencoder.

Fonte: Autoria própria.

4.4 Geração de feixes OAM-LG

Os feixes transmitidos são gerados através da Equação 4.1, utilizada por Runge et al. (2019) para a obtenção de feixes OAM-LG de modos com superposição simples $\pm\ell$, onde ψ representa a solução da aproximação paraxial para modos LG e L representa o conjunto de todos os possíveis valores de ℓ disponíveis para a formação do alfabeto.

$$S(x, y, z) = \sum_{\ell \in L} (\psi_0^{+\ell} + \psi_1^{-\ell}) \quad (4.1)$$

4.5 Simulação da turbulência atmosférica

A distorção do feixe causada pela turbulência atmosférica é simulada através da aplicação de uma matriz de turbulência gerada pela Equação 4.2, conforme Runge et al. (2019), onde $\Phi(k_x, k_y)$ representa a função de aproximação da densidade espectral da potência de Kolmogorov (Equação 2.2), $IFFT$ é a transformada inversa bidimensional de Fourier e C representa uma matriz de ruído formada por variáveis aleatórias gaussianas complexas contendo a mesma dimensão da quantidade de pixels que formam a imagem do feixe gerado.

$$\phi = IFFT(C\sqrt{\Phi(k_x, k_y)}) \quad (4.2)$$

A Figura 4.2 demonstra as imagens resultantes de um feixe OAM-LG com carga topológica $\ell = \pm 4$ exposto a três níveis de intensidade de turbulência, de maneira similar às intensidades utilizadas em Runge et al. (2019). O nível de turbulência é dado por $D/r_0 = 0.0005$ para turbulência fraca, $D/r_0 = 0.003$ para turbulência moderada e $D/r_0 = 0.009$ para turbulência forte.

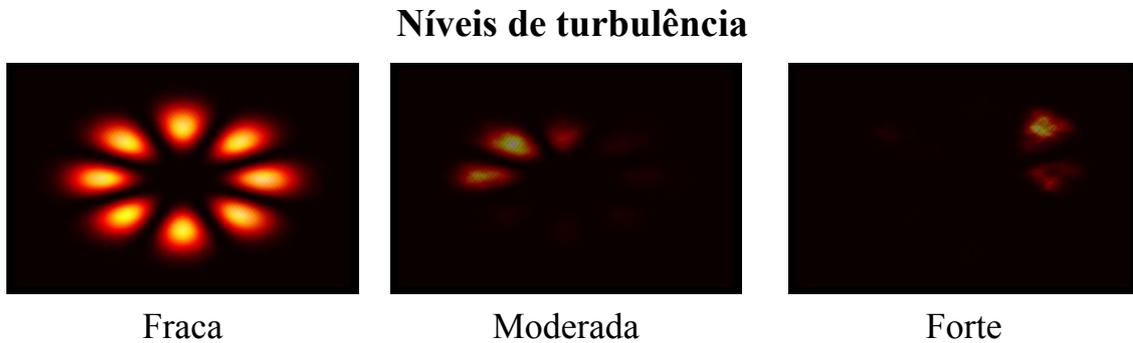


Figura 4.2: Aplicação dos diferentes níveis de turbulência.

Fonte: Autoria própria.

4.6 Modelo desenvolvido

O modelo foi desenvolvido visando a geração de um alfabeto com cardinalidade 16 (i. e. composto por 16 símbolos), onde cada símbolo representa a multiplexação de dois feixes LG com momento angular orbital $\pm\ell$ distintos superpostos, no qual $1 \leq |\ell| \leq 16$.

A Figura 4.3 representa a arquitetura do encoder desenvolvido. A primeira camada contém 4 neurônios, que receberão o vetor de bits representando o número do símbolo. Na segunda

camada, o autoencoder utiliza uma função sigmoide para gerar uma saída de 8 números com valores aproximadamente binários. A terceira camada utiliza os valores binários aproximados para convertê-los em dois números inteiros d_1 e d_2 (contidos no intervalo $[0, 15]$), que são utilizados na próxima camada para gerar a seção transversal do feixe transmitido, onde $\ell_1 = d_1 + 1$ e $\ell_2 = d_2 + 1$, garantindo que $1 \leq |\ell| \geq 16$.

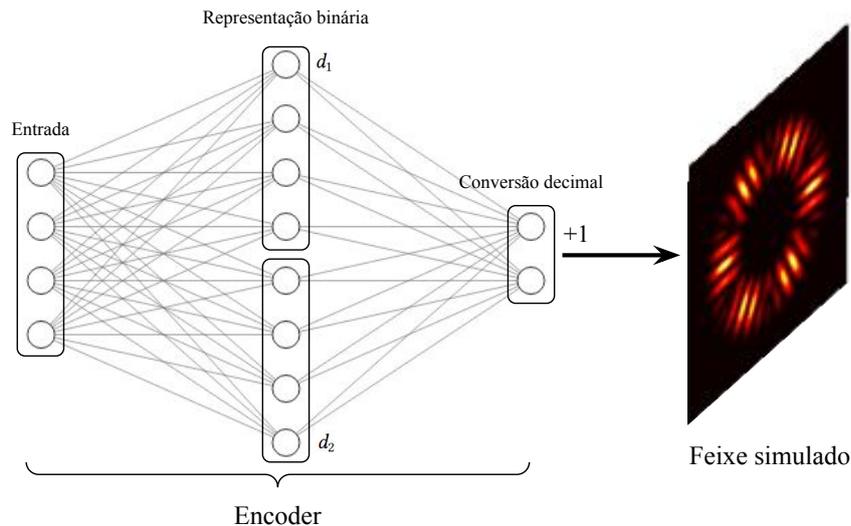


Figura 4.3: Arquitetura do autoencoder desenvolvido.

Fonte: Autoria própria.

O desempenho do autoencoder desenvolvido não obteve os resultados propostos inicialmente. A implementação do autoencoder ocorreria em duas etapas: inicialmente, o modelo seria construído sem a simulação da turbulência atmosférica, e uma vez que este atingisse um desempenho satisfatório, seria incrementada a turbulência, proporcionando a aprendizagem em um ambiente mais próximo do real.

A implementação da etapa inicial não alcançou desempenho satisfatório e não convergiu em uma boa acurácia, estagnando em valores próximos de 50%, sem obtenção de sucesso nas tentativas de solucionar o problema. Duas hipóteses foram levantadas acerca da causa da limitação do modelo, sendo a primeira que as restrições aplicadas ao modelo, tais como a utilização de uma sigmoide ingreme para aproximar os valores aprendidos à bits, limitam de forma excessiva o gradiente de aprendizado no qual o modelo tenta aprender. A segunda seria que, por realizar operações com números muito baixos, fazem com que o modelo se perca e não consiga prosseguir com o aprendizado. Dessa forma, os resultados apresentados são referentes ao modelo mais simples, sem a incrementação da turbulência atmosférica.

4.7 Análise dos resultados

A Tabela 4.2 apresenta os resultados obtidos na saída do encoder utilizando os dados apresentados na Seção 4.2 e os pares $(\pm\ell_1, \pm\ell_2)$ que compõem o alfabeto obtido. Nota-se que os valores contém algumas anomalias: a predição dos símbolos 1 e 11 são idênticas, o que demonstra que o modelo deve implementar alguma forma de restringir o uso da mesma representação para duas entradas diferentes. Ainda em relação ao símbolo 1, tem-se $\ell_1 = \ell_2$, o que não representa a superposição de dois modos distintos.

Tabela 4.2: Valores preditos pelo encoder para cada símbolo de entrada.

| Símbolo | Decimais preditos pelo encoder | Valor dos modos $\pm\ell$ |
|----------------|---------------------------------------|---|
| 1 | (7, 7) | (8, 8) |
| 2 | (4, 2) | (5, 3) |
| 3 | (2, 11) | (3, 12) |
| 4 | (6, 0) | (7, 1) |
| 5 | (0, 13) | (1, 14) |
| 6 | (6, 1) | (7, 2) |
| 7 | (2, 14) | (3, 15) |
| 8 | (1, 14) | (2, 15) |
| 9 | (2, 7) | (3, 8) |
| 10 | (5, 2) | (6, 3) |
| 11 | (7, 7) | (8, 8) |
| 12 | (6, 0) | (7, 1) |
| 13 | (0, 7) | (1, 8) |
| 14 | (8, 3) | (9, 4) |
| 15 | (6, 9) | (7, 10) |
| 16 | (11, 1) | (12, 2) |

A Figura 4.4 ilustra a imagem da representação do perfil transversal de cada símbolo obtido, listados na Tabela 4.2. As imagens foram calculadas a partir das mesmas equações utilizadas em Runge et al. (2019).

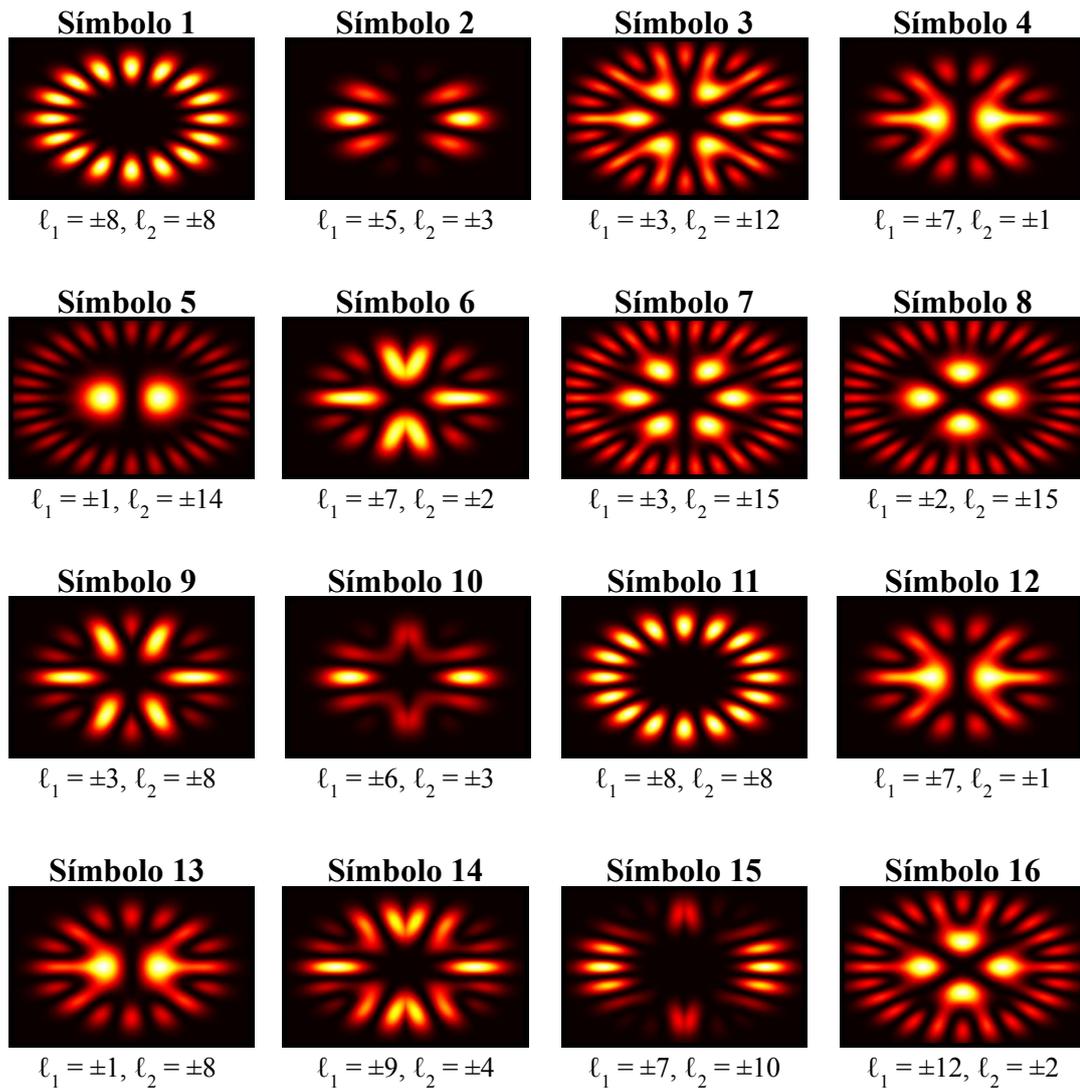


Figura 4.4: Imagens representando o perfil transversal dos resultados obtidos.

Fonte: Autoria própria.

Capítulo 5

Conclusões

A técnica proposta por O'Shea e Hoydis (2017) mostra-se inovadora, mas sua implementação não é uma tarefa trivial, sendo a parte da simulação das restrições físicas do sistema uma tarefa especialmente complexa de ser realizada.

O modelo desenvolvido corresponde apenas a uma versão inicial, com pendências a serem solucionadas. Seu desenvolvimento será continuado ao decorrer de um trabalho de mestrado, onde uma vez finalizado, o alfabeto obtido será comparado com os estudos realizados por Runge et al. (2019) e Freitas et al. (2020), a fim de verificar se este apresenta uma acurácia maior do que os alfabetos que compõem os cenários de teste estudados. Se for comprovado que o alfabeto obtido pelo modelo desenvolvido é realmente otimizado em relação aos demais alfabetos, o trabalho contribuirá para o avanço de sistemas de comunicação por feixes com momento angular orbital.

Referências bibliográficas

ALLEN, L.; BEIJERSBERGEN, M. W.; SPREEUW, R. J. C.; WOERDMAN, J. P. Orbital angular momentum of light and the transformation of Laguerre-Gaussian laser modes. **Phys. Rev. A**, American Physical Society, v. 45, p. 8185–8189, 11 jun. 1992. DOI: 10.1103/PhysRevA.45.8185. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevA.45.8185>>.

BOROVICKA, T.; JR., M. J.; KORDIK, P.; JIRINA, M. Selecting Representative Data Sets. In: KARAHOCA, A. (Ed.). **Advances in Data Mining Knowledge Discovery and Applications**. Rijeka: IntechOpen, 2012. cap. 2. DOI: 10.5772/50787. Disponível em: <<https://doi.org/10.5772/50787>>.

DERTAT, A. (Ed.). **Applied Deep Learning - Part 3: Autoencoders**. 2020. Disponível em: <<https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>>. Acesso em: 26 dez. 2020.

FLICKENGER, R. **Wireless Networking in the Developing World**. [S.l.]: CreateSpace Independent Publishing Platform, 2013. ISBN 9781484039359. Disponível em: <<http://wndw.net/>>.

FRANCOIS, C. (Ed.). **Building Autoencoders in Keras**. 2020. Disponível em: <<https://blog.keras.io/building-autoencoders-in-keras.html>>. Acesso em: 26 dez. 2020.

FREITAS, B. S. et al. Optimized OAM Laguerre-Gauss Alphabets for Demodulation using Machine Learning. In: 2020 IEEE 8th International Conference on Photonics (ICP). [S.l.: s.n.], 2020. p. 24–25. DOI: 10.1109/ICP46580.2020.9206470.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**, v. 36, n. 4, p. 193–202, abr. 1980. ISSN 1432-0770. DOI: 10.1007/BF00344251. Disponível em: <<https://doi.org/10.1007/BF00344251>>.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. [S.l.]: O'Reilly Media, 2019. ISBN 9781492032595. Disponível em: <<https://books.google.com.br/books?id=HnetDwAAQBAJ>>.

GUO, Z.; WANG, Z.; DEDO, M. I.; GUO, K. The Orbital Angular Momentum Encoding System With Radial Indices of Laguerre-Gaussian Beam. **IEEE Photonics Journal**, v. 10, n. 5, p. 1–11, 2018. DOI: 10.1109/JPHOT.2018.2859807.

LANE, R. G.; GLINDEMANN, A.; DAINTY, J. C. Simulation of a Kolmogorov phase screen. **Waves in Random Media**, Taylor & Francis, v. 2, n. 3, p. 209–224, 1992. DOI: 10.1088/

0959-7174/2/3/003. eprint: <https://doi.org/10.1088/0959-7174/2/3/003>. Disponível em: <<https://doi.org/10.1088/0959-7174/2/3/003>>.

LI, S. et al. Atmospheric turbulence compensation in orbital angular momentum communications: Advances and perspectives. **Optics Communications**, v. 408, p. 68–81, 2018. Optical Communications Exploiting the Space Domain. ISSN 0030-4018. DOI: <https://doi.org/10.1016/j.optcom.2017.09.034>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S003040181730809X>>.

MALIK, A.; SINGH, P. Free Space Optics: Current Applications and Future Challenges. **International Journal of Optics**, v. 2015, p. 1–7, nov. 2015. DOI: 10.1155/2015/945483.

O'SHEA, T.; HOYDIS, J. An Introduction to Deep Learning for the Physical Layer. **IEEE Transactions on Cognitive Communications and Networking**, v. 3, n. 4, p. 563–575, 2017. DOI: 10.1109/TCCN.2017.2758370.

PADGETT, M.; COURTIAL, J.; ALLEN, L. Light's Orbital Angular Momentum. **Physics Today**, v. 57, n. 5, p. 35–40, 2004. DOI: 10.1063/1.1768672. eprint: <https://doi.org/10.1063/1.1768672>. Disponível em: <<https://doi.org/10.1063/1.1768672>>.

ROUTRAY, S. K. et al. **Communication Bandwidth for Emerging Networks: Trends and Prospects**. [S.l.: s.n.], 2019. arXiv: 1903.04811 [eess.SP].

RUNGE, C. et al. Utilização de Aprendizado de Máquina na Demodulação de Sistemas OAM utilizando Multiplexação de Modos Orbitais. In: DOI: 10.14209/sbrt.2019.1570558862.