UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FERNANDO FERREIRA DE LIMA

GERAÇÃO DE PENTES DE FREQUÊNCIA ÓPTICOS PROGRAMÁVEIS VIA DESIGN INVERSO E DEEP LEARNING

CAMPINA GRANDE, PB
2024

FERNANDO FERREIRA DE LIMA

GERAÇÃO DE PENTES DE FREQUÊNCIA ÓPTICOS PROGRAMÁVEIS VIA DESIGN INVERSO E DEEP LEARNING

Monografia de Trabalho de Conclusão de Curso de Graduação de Bacharelado submetida à Coordenadoria de Graduação em Engenharia Elétrica da Universidade Federal De Campina Grande - UFCG como parte dos requisitos necessários para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr Edson Porto da Silva

Fernando Ferreira de Lima
Orientando

Prof. Dr Edson Porto da Silva
Orientador

CAMPINA GRANDE, PB
2024

AGRADECIMENTOS

Primeiramente, agradeço aos meus pais, Severino e Dione, por seu amor incondicional, apoio constante e pelos valores que me transmitiram ao longo da vida. Sem a base sólida que eles proporcionaram, esta conquista não seria possível.

Aos meus irmãos, Filipe e Felix, agradeço por estarem sempre presentes, por compartilharem comigo risadas, conselhos e por serem fontes constantes de motivação e inspiração.

À minha companheira, Nathália, agradeço por seu apoio inabalável, compreensão e por ser minha fonte de força e inspiração nos momentos mais difíceis.

Aos demais familiares, amigos e colegas, expresso minha gratidão pela paciência, incentivo e compreensão ao longo desses anos.

Aos amigos que fiz durante a graduação, obrigado por compartilharem risadas, experiências e por tornarem essa jornada mais leve e memorável.

Aos professores que me guiaram e inspiraram, especialmente ao Professor Adolfo Herbster, meu orientador de projeto durante quase toda a graduação, e ao Professor Edson Porto, meu orientador de TCC, agradeço por seu conhecimento, orientação e pela confiança em meu potencial.

Por fim, agradeço a todos os que de alguma forma contribuíram para o desenvolvimento deste trabalho e para o meu crescimento pessoal e acadêmico.

A coragem não é a ausência do medo, mas a capacidade de vencê-lo.
(Nelson Mandela).

RESUMO

LIMA, Fernando. Geração de Pentes de Frequência Ópticos Programáveis via *Design* Inverso e *Deep Learning*. 2024. 40 f. Monografia de Trabalho de Conclusão de Curso de Graduação – Curso de Bacharelado em Engenharia Elétrica, Universidade Federal De Campina Grande. Campina Grande, PB, 2024.

Os pentes de frequência óptica (OFC) são conjuntos discretos de portadoras de onda contínua periodicamente espaçadas em frequência que têm importantes aplicações em medição, comunicação, espectroscopia e outros campos. Geralmente, os OFCs precisam ser projetados de acordo com diferentes aplicações. No entanto, os métodos existentes para projetar os parâmetros operacionais dos geradores de OFC, como aqueles baseados em algoritmos de otimização, são demorados, ineficientes e difíceis de alcançar resultados ótimos. Neste trabalho, é experimentado o método de design inverso (*Inverse Design*) de OFC utilizando redes neurais de aprendizado profundo (*Deep Learning Neural Networks*), que produz resultados em tempo hábil comparado a métodos convencionais e ainda pode melhorar o desempenho do OFC gerado. Nesse método, de acordo com o OFC alvo requerido, a rede neural treinada pode ser usada para projetar inversamente os parâmetros correspondentes. O método é aplicado a dois casos de geradores de OFC baseados em cascatas de moduladores eletro-ópticos (*Eletro-Optical Combs* - EOC) com um único sinal de controle. As redes neurais treinadas são testadas sob as mesmas condições e comparadas entre si. O método implementado pode ser estendido para geradores de OFC mais complexos, sendo uma ferramenta promissora para o design eficiente de OFCs.

Palavras-chave: Pentes de Frequência Óptica (OFCs). *Design* Inverso. Redes Neurais de Aprendizado Profundo. Pentes Eletro-Ópticos (EOCs). *Design* de parâmetros.

ABSTRACT

LIMA, Fernando. Programmable Optical Frequency Comb Generation via Inverse Design and Deep Learning. 2024. 40 f. Monografia de Trabalho de Conclusão de Curso de Graduação – Curso de Bacharelado em Engenharia Elétrica, Universidade Federal De Campina Grande. Campina Grande, PB, 2024.

Optical frequency combs (OFCs) are discrete sets of continuous wave carriers periodically spaced in frequency that have important applications in measurement, communication, spectroscopy and other fields. Generally, OFCs need to be designed according to different applications. However, existing methods for designing the operating parameters of OFC generators, such as those based on optimization algorithms, are time-consuming, inefficient and difficult to achieve optimal results. In this work, the Inverse Design method of OFC using Deep Learning neural networks is experimented with, which produces results in a timely manner compared to conventional methods and can also improve the performance of the generated OFC. In this method, according to the target OFC required, the trained neural network can be used to inversely design the corresponding parameters. The method is applied to two cases of OFC generators based on cascades of electro-optical modulators (EOC) with a single control signal. The trained neural networks are tested under the same conditions and compared with each other. The implemented method can be extended to more complex OFC generators, making it a promising tool for the efficient design of OFCs.

Keywords: Optical Frequency Combs (OFCs). Inverse Design. Deep Learning. Electro-Optical Combs (EOC). Parameter design.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Adam	*Adaptive Moment Estimation*
CW	*Continuous Wave*
DDMZM	*Dual-Drive Mach-Zehnder Modulator*
EOC	*Electro-Optical Comb*
EOM	*Electro-Optical Modulator*
FAIR	*Facebook AI Research*
FNN	*Feedforward Neural Network*
GD	*Gradient Descent*
GPU	*Graphics Processing Unit*
IM	*Intensity Modulator*
MAE	*Mean Absolute Error*
MLL	*Mode-Locked Laser*
MSE	*Mean Squared Error*
MZM	*Mach–Zehnder Modulator*
OFC	*Optical Frequency Comb*
PM	*Phase Modulator*
SGD	*Stochastic Gradient Descent*
WDM	*Wavelength-Division Multiplexing*

LISTA DE SÍMBOLOS

K	Índice de modulação
π	Constante irracional *pi*
V_π	Tensão de meia onda do modulador
V_b	Tensão de *bias* do modulador
ω_c	Frequência da portadora óptica
ω_m	Frequência de modulação
f_m	Taxa de repetição
t	Tempo
ω	Frequência
P	Potência do *laser* CW
φ_0	Fase do sinal elétrico de controle
V_0	Amplitude máxima do sinal elétrico de controle
A_0	Amplitude máxima do campo elétrico de saída
$V(t)$	Amplitude do sinal de controle
$A(t)$	Amplitude do campo elétrico de saída
$\tilde{A}(\omega)$	Amplitude do espectro óptico
$\Delta\phi$	Desvio de fase
$J_n(x)$	Função de Bessel de primeiro tipo e ordem n
$\delta(\omega)$	Impulso unitário no dominío da frequência
x_i	i-ésima entrada do neurônio
w_{ki}	i-ésimo peso do neurônio k
b_k	*Bias* do neurônio k
$\varphi(v_k)$	Função de ativação
v_k	Saída do neurônio k antes da função de ativação

y_k	Saída do neurônio k
X_i	i-ésima amostra do conjundo de dados
X_{max}	Maior valor do conjunto de dados
X_i'	i-ésima amostra do conjundo de dados normalizada
y_i	Valor real da i-ésima amostra do conjundo de dados
$\hat{y}_i$	Valor previsto pelo modelo para a i-ésima amostra do conjundo de dados

LISTA DE ALGORITMOS

SUMÁRIO

1 INTRODUÇÃO

Nas últimas décadas, as comunicações ópticas desempenham um papel crucial ao fornecer largura de banda e velocidade de transmissão crescentes para atender à crescente demanda por conectividade de alta velocidade e baixa latência. Como resultado, surge a necessidade constante de desenvolver tecnologias que permitam aumentar a capacidade e a eficiência dos sistemas de comunicação óptica.

Uma abordagem promissora para aprimorar o desempenho dos sistemas de comunicação óptica é o uso de Pentes de Frequência Óptica (*Optical Frequency Combs* - OFCs). Esses pentes são conjuntos discretos de portadoras de onda contínua periodicamente espaçadas na frequência, desempenhando papéis importantes em diversas aplicações, como medição óptica de precisão, comunicação e sensoriamento óptico.

No contexto das comunicações ópticas, os *OFCs* oferecem a capacidade de modular informações em múltiplas frequências, aumentando assim a capacidade de transmissão de dados em sistemas de fibra óptica. No entanto, para aproveitar todo o potencial dos OFCs em aplicações de comunicação, é essencial projetar geradores que produzam pentes de frequência com características desejadas, como formato espectral específico.

A geração de OFC utiliza várias técnicas, sendo o uso de moduladores eletro-ópticos uma escolha proeminente devido à sua simplicidade e flexibilidade de espaçamento entre as linhas do pente. No entanto, mesmo em geradores aparentemente simples, desafios surgem devido à natureza não linear do sistema e à distribuição desigual de potência ao longo do pente. Métodos tradicionais enfrentam limitações na determinação de parâmetros, dificultando a obtenção de resultados ótimos.

Dentro desse cenário, este trabalho apresenta uma abordagem promissora com o projeto inverso (*Inverse Design*) baseado em redes de aprendizado profundo para otimizar a criação de *OFCs*. Por meio do uso de técnicas de *Machine Learning* e *Deep Learning*, o objetivo é desenvolver uma rede neural capaz de projetar e controlar precisamente as características dos *OFCs*, garantindo que atendam às especificações desejadas de desempenho e funcionalidade. Ao alcançar este objetivo, espera-se contribuir significativamente para o avanço contínuo das comunicações ópticas e o desenvolvimento de sistemas mais eficientes e poderosos para aplicações futuras.

1.1 **Objetivos**

1.1.1 Objetivo geral

O presente trabalho tem por objetivo implementar e treinar por meio de simulação computacional uma rede neural de *Inverse Design* baseada em *Deep Learning* para que seu desempenho possa ser avaliado sob diferentes condições sistêmicas no contexto de aplicações

em comunicações ópticas.

1.1.2 Objetivos específicos

- Implementar computacionalmente um gerador de OFC baseado em cascata de moduladores eletro-ópticos;
- Implementar e treinar computacionalmente a rede neural baseada em *Deep Learning* e obter o *Inverse Design*;
- Verificar o desempenho da rede neural sob diferentes condições impostas (formato do OFC, número de picos);
- Comparar os resultados obtidos com os da literatura do assunto relacionado.

1.2 **Estrutura do Trabalho**

Este trabalho está dividido em cinco capítulos.

O Capítulo 1 oferece uma introdução ao tema, destacando a importância das comunicações ópticas e o papel dos pentes de frequência nesse contexto.

O Capítulo 2 é dedicado à fundamentação teórica, explorando os conceitos-chave relacionados aos pentes eletro-ópticos e às redes neurais de aprendizado profundo.

Na metodologia, apresentada no Capítulo 3, é descrito detalhadamente o processo de geração de pentes eletro-ópticos simulados, a criação e preparação do conjunto de dados e a definição dos hiperparâmetros para o treinamento das redes neurais.

Os resultados obtidos são discutidos no Capítulo 4, onde são apresentadas as análises das redes direta e inversa treinadas.

Finalmente, no Capítulo 5, são apresentadas as conclusões do trabalho, destacando os resultados alcançados, as contribuições para o campo e sugestões para trabalhos futuros.

2 REVISÃO DE LITERATURA

Neste capítulo, são abordados dois campos fundamentais para este trabalho: os Pentes Eletro-Ópticos e o Aprendizado de Máquina. Na seção dos pentes eletro-ópticos, são apresentados os moduladores e configurações de fonte mais comuns na geração de OFCs, e em seguida são mostradas algumas de suas vantagens e aplicações. Na seção de aprendizado de máquina, é dada uma introdução sobre os métodos de treinamento mais usuais e em seguida é apresentada uma explicação sobre as redes neurais de aprendizado profundo, com as informações necessárias para entender a composição dessa poderosa ferramenta.

2.1 **Pentes Eletro-Ópticos**

Os pentes eletro-ópticos, também conhecidos como EOCs (*Eletro-Optic Combs*), são um tipo especial de OFC gerado por meio de moduladores eletro-ópticos (EOMs). A geração de OFC utiliza uma diversidade de técnicas, destacando-se os moduladores eletro-ópticos devido à sua simplicidade e à capacidade de ajustar o espaçamento entre as linhas do pente de forma flexível, tornando-os uma escolha evidente (ZHUANG et al., 2023). Sua capacidade de gerar sinais ópticos em uma ampla gama de frequências e sua versatilidade os tornam importantes em diversas aplicações, tais como medição óptica de precisão, comunicação óptica, sensoriamento óptico, aplicações militares e diversas outras aplicações em espectroscopia (PARRIAUX; HAMMANI; MILLOT, 2021).

2.1.1 Moduladores Eletro-Ópticos

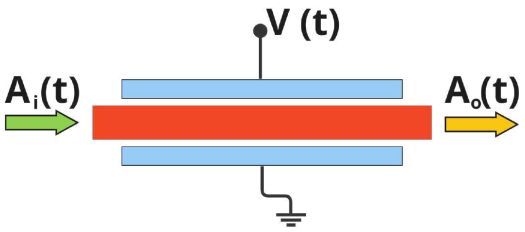
Os moduladores eletro-ópticos são baseados nos efeitos eletro-ópticos, sendo fenômenos que modificam o índice de refração de um material pela aplicação de um campo elétrico estático. Diversos fenômenos podem ser definidos como efeito eletro-óptico; no entanto, o mais evidente e forte entre eles é o efeito de Pokels, também chamado de efeito eletro-óptico linear. Tal efeito demonstra que o índice de refração de um material pode mudar linearmente com o campo elétrico aplicado, sendo também o efeito mais utilizado dentre os EOMs existentes.

Os EOMs mais comuns são os moduladores de fase (*Phase Modulator* — PM) e os moduladores de Mach–Zehnder (*Mach-Zehnder Modulator* — MZM).

2.1.1.1 Modulador de Fase

Um modulador de fase eletro-óptico é um dispositivo que altera a fase de um feixe de luz ao aplicar um campo elétrico a um material, modificando seu índice de refração. Essa alteração no índice de refração muda a fase da luz que atravessa o material. A ilustração de um PM é mostrada na Figura 1.

Figura 1 – Ilustração do modulador de fase eletro-óptico



Elaborado pelo autor.

Considerando um PM com índice de modulação $K = \frac{\pi}{V_\pi}$, alimentado por um *laser* CW (*Continuous Wave*) de campo óptico dado por $A_i(t) = A_0 e^{i\omega_c t}$, sendo ω_c a frequência portadora e A_0 a amplitude, e controlado por um gerador de forma de onda elétrica senoidal com frequência de modulação ω_m, o campo elétrico proporcional à tensão elétrica definida por $V(t) = V_0 sin(\omega_m t)$, onde V_0 é a tensão de pico, tem-se que na saída do modulador, o campo óptico inicial adquire uma fase $\Delta\phi(t) = KV(t)$, resultando na expressão do campo óptico de saída $A(t)$:

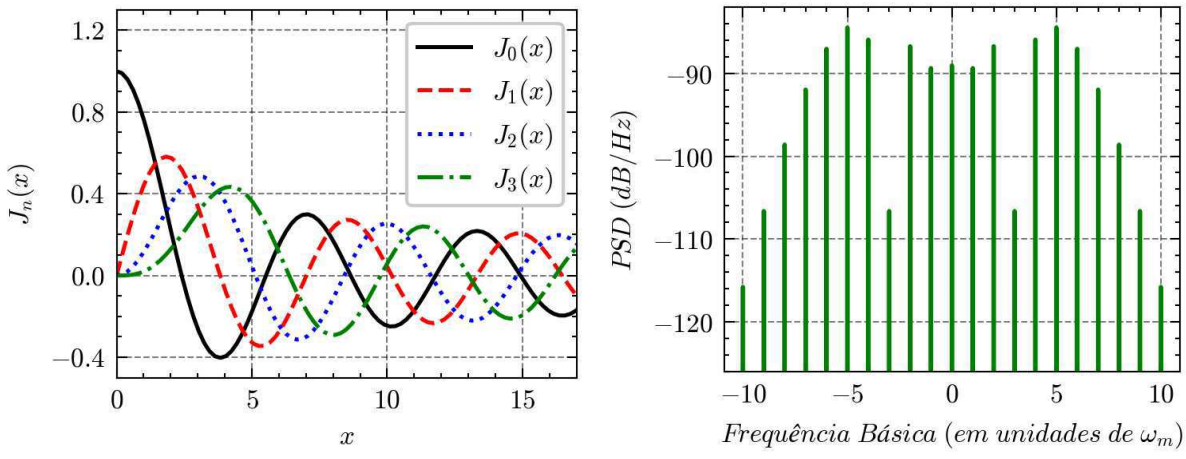
$$A(t) = A_i(t) \times e^{i\Delta\phi(t)} = A_0 e^{i\omega_c t} \times e^{iKV_0 sin(\omega_m t)}. \tag{1}$$

No domínio da frequência, a amplitude do espectro é obtida com a transformada de Fourier e outras manipulações (PARRIAUX; HAMMANI; MILLOT, 2020):

$$\tilde{A}(\omega) = A_0 \sum_{n=-\infty}^{\infty} J_n(KV_0)\delta(\omega - n\omega_m - \omega_c), \tag{2}$$

onde J_n é a função de Bessel de primeiro tipo e ordem n. Essa última equação mostra que, de fato, a modulação resulta em um OFC, centrado em ω_c com os impulsos $\delta(\omega)$ igualmente separados por ω_m.

Figura 2 – Funções de Bessel de primeiro tipo. Exemplo de OFC gerado por modulação PM



(a) (b)

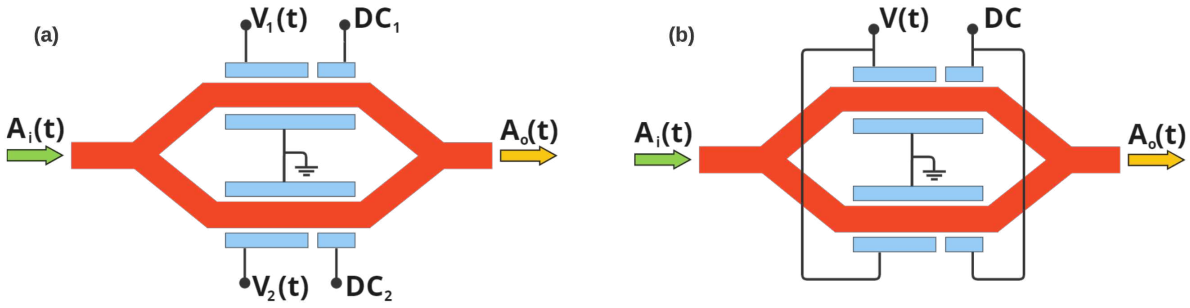
Elaborado pelo autor.

A Equação 2 também demonstra que as amplitudes das linhas do EOC seguem as funções de Bessel do primeiro tipo J_n, como apresentado na Figura 2(a) para as quatro primeiras ordens. Essa característica é particularmente importante, pois se observa que o índice de modulação K e a tensão de pico V_0 são os únicos parâmetros que podem determinar a amplitude, a planicidade e o número de linhas do pente. Na Figura 2(b), é mostrado um exemplo de EOC simulado gerado com um PM, onde é possível observar dois máximos em torno das extremidades do espectro, sendo essa característica natural desse tipo de OFC.

2.1.1.2 Modulador de Mach-Zehnder

Um modulador de Mach-Zehnder eletro-óptico é um dispositivo que controla a fase e a amplitude de um feixe de luz utilizando interferência. É composto por dois braços interferométricos, onde a luz é dividida e depois recombinada, permitindo a modulação precisa da luz. Trata-se basicamente da estrutura do Interferômetro de Mach-Zenhder (*Mach-Zehnder Interferometer* — MZI) auxiliado por dois moduladores de fase, um em cada braço. A Figura 3 mostra a ilustração de duas diferentes configurações de MZMs.

Figura 3 – Ilustração de duas configurações comuns de moduladores de baseados em MZI. (a) Modulador de Mach-Zenhder Controle Duplo (DDMZM). (b) Modulador de Mach-Zenhder de Controle Único (SDMZM)



Elaborado pelo autor.

O caso mais geral de um modulador de Mach-Zehnder, (visto na Figura 3a) é chamado de MZM de duplo controle (*Dual-Drive Mach-Zehnder Modulator* — DDMZM), que possui dois braços, cada um com seu respectivo índice de modulação K_1 e K_2. Para a geração de OFCs, cada braço do modulador recebe um sinal elétrico senoidal: $V_1(t) = V_{01}sin(\omega_m t)$ é aplicado em um braço, e $V_2(t) = V_{02}sin(\omega_m t)$ é aplicado no outro. A saída desse modulador no domínio do tempo é dada por $A(t)$, conforme a seguinte equação:

$$A(t) = \frac{A_0}{2} e^{i\omega_c t} \times \left(e^{iK_1(V_1(t)+V_{b1})} + e^{iK_2(V_2(t)+V_{b2})} \right), \tag{3}$$

onde, V_{b1} e V_{b2} são, respectivamente, os vieses de tensão (*bias voltage*) aplicados nos braços 1 e 2. O espectro óptico é dado por

$$\tilde{A}(\omega) = \frac{A_0}{2} \sum_{n=-\infty}^{\infty} \left(J_n(K_1 V_{01}) e^{iK_1 V_{b1}} + J_n(K_2 V_{02}) e^{iK_2 V_{b2}} \right) \delta(\omega - n\omega_m - \omega_c). \tag{4}$$

A Equação 4 mostra que a amplitude das linhas do EOC são controladas pelos parâmetros K, V_0 e V_b dos dois braços do modulador, o que garante mais graus de liberdade úteis para planificação do espectro.

O DDMZM pode ser configurado para funcionar com apenas uma entrada de controle (Figura 3b), o que é chamado de SDMZM (*Single-Drive Mach-Zehnder Modulator*) ou simplesmente MZM. Fazendo $K_1 = K_2 = K$, $V_{01} = V_{02} = V_0$ e $V_{b1} = V_{b2} = V_b$, tem-se o chamado MZM modo *push-push*, que se comporta como um PM. No domínio do tempo tem-se a amplitude do campo óptico:

$$A(t) = A_0 e^{i\omega_c t} \times e^{iK(V(t)+V_b)}, \tag{5}$$

onde $V(t) = V_0 sin(\omega_m t)$. No domínio da frequência tem-se a amplitude do espectro óptico:

$$\tilde{A}(\omega) = A_0 e^{iKV_b} \sum_{n=-\infty}^{\infty} J_n(KV_0)\delta(\omega - n\omega_m - \omega_c). \tag{6}$$

Pode-se observar a semelhança que a Equação 6 tem com a Equação 2, a diferença é apenas a influência do V_b nas amplitudes das linhas do EOC.

Existe também outra configuração SDMZM, chamada de *push-pull*, obtida ao fazer $K_1 = K_2 = K$, $V_{01} = -V_{02} = V_0$ e $V_{b1} = -V_{b2} = V_b$. Tal configuração faz o MZM funcionar como um puro modulador de intensidade (*Intensity Modulator* — IM), com amplitude do campo óptico dada por:

$$A(t) = \frac{A_0}{2} e^{i\omega_c t} \times \left(e^{iK(V(t)+V_b)} + e^{-iK(V(t)+V_b)}\right) = A_0 e^{i\omega_c t} \times cos(K(V(t) + V_b)), \tag{7}$$

onde foi usada uma propriedade decorrente da fórmula de Euler: $2cos(\theta) = e^{i\theta} + e^{-i\theta}$. No domínio da frequência, tem-se a amplitude do espectro óptico:

$$\begin{aligned}\tilde{A}(\omega) &= \frac{A_0}{2} \sum_{n=-\infty}^{\infty} \left(J_n(KV_0)e^{iKV_b} + J_n(-KV_0)e^{-iKV_b}\right) \delta(\omega - n\omega_m - \omega_c) \\ &= \frac{A_0}{2} \sum_{n=-\infty}^{\infty} J_n(KV_0) \left(e^{iKV_b} + (-1)^n e^{-iKV_b}\right) \delta(\omega - n\omega_m - \omega_c),\end{aligned} \tag{8}$$

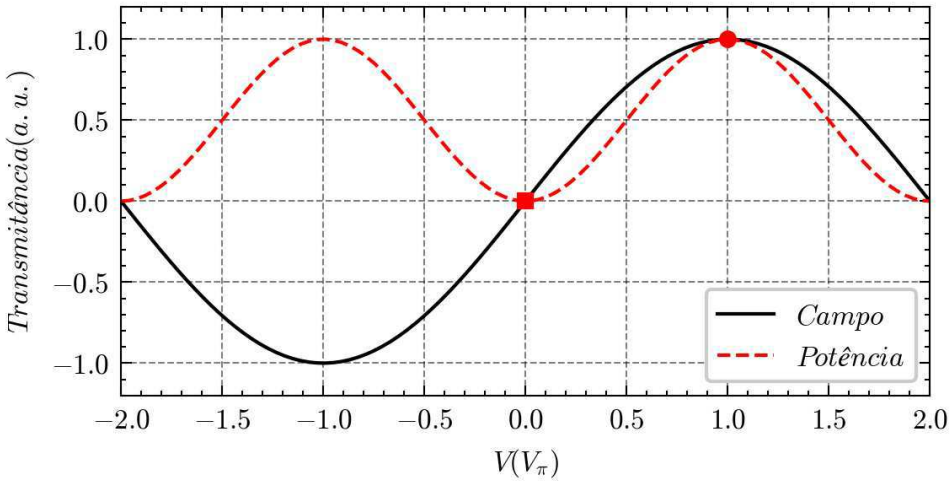
onde se utilizou que as Funções de Bessel do primeiro tipo são simétricas para n par e antissimétricas para n ímpar.

A partir da Equação 7 é possível obter a função de transferência do *single-drive* MZM no modo *push-pull*:

$$T = \frac{|A_{out}|^2}{|A_{in}|^2} = cos^2(KV(t) + KV_b). \tag{9}$$

A Figura 4 mostra, com base na Equação 9, um exemplo de curva de operação do MZM no modo *push-pull* (puro IM). Os pontos de máxima e nula transmissão estão destacados na figura.

Figura 4 – Exemplo de curva de operação do *single-drive* MZM no modo *push-pull*



Elaborado pelo autor.

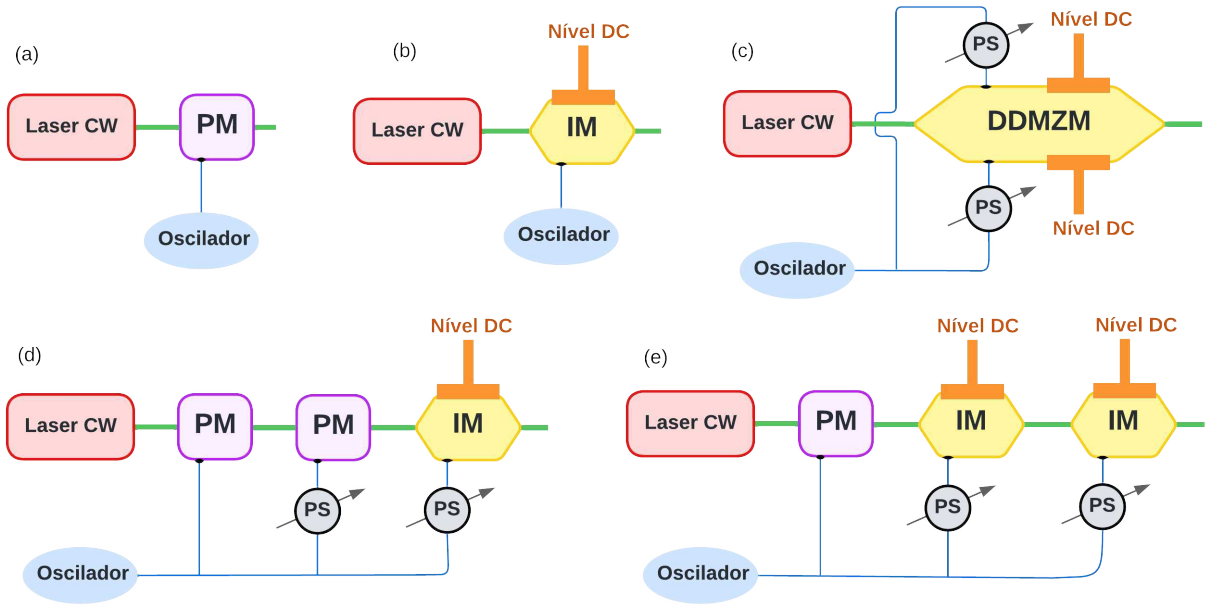
2.1.2 Configurações de Fonte Comuns

Exitem diferentes tipos de configurações que podem ser utilizadas para a geração de EOCs. A seguir são introduzidas as configurações mais comuns: moduladores individuais, cascata de modulares e cavidades eletro-ópticas. Algumas dessas configurações são ilustradas na Figura 5.

- **Moduladores Individuais:**
  A geração de EOC pode ser simplificada quando se utiliza apenas um EOM. Três configurações principais são possíveis, conforme apresentado nas Figuras 5(a)–5(c). Do ponto de vista experimental, essas configurações são limitadas pela largura de banda eletro-óptica, que determina o espaçamento máximo das linhas. Atualmente, os EOMs comerciais têm uma largura de banda comum de cerca de 40 GHz, com alguns alcançando até 70 GHz. Quanto aos DDMZMs, configurações foram propostas devido à alta flexibilidade desses EOMs, e essas configurações são geralmente projetadas para produzir um pente plano.
- **Cascata de Moduladores:**
  Como um EOM pode ser danificado se uma tensão RF muito alta for aplicada, com esse limite geralmente sendo abaixo de 40 dBm para os EOMs comerciais mais eficientes, o número máximo de linhas geradas por um EOM é limitado. Esse problema pode ser resolvido ao cascatear vários EOMs em série, aumentando o número de linhas geradas por um fator igual ao número de EOMs. Uma configuração em cascata comum é feita com uma série de PM seguidos de um IM utilizado para melhorar a planicidade do espectro. As Figuras 5(d)-5(e) mostram possíveis configurações de cascata com três moduladores.
- **Cavidades eletro-ópticas:**
  Outra categoria de configurações comuns usadas para a geração de EOC é definida por aquelas que exploram o efeito eletro-óptico em uma cavidade. Um arranjo natural que

pode ser utilizado para aumentar o potencial de um PM é colocar o EOM numa cavidade, como a cavidade Fabry–Perot ou uma cavidade de fibra. Tais configurações requerem condições particulares, uma vez que o tempo de trânsito da cavidade deve corresponder à taxa de repetição do sinal RF. Portanto, essas configurações são geralmente controladas para contrabalançar desvios que poderiam deslocar o sistema para fora da ressonância. Apesar das dificuldades de implementação, os recursos de filtragem de uma cavidade são mais uma característica interessante que pode ser vantajosa em algumas aplicações específicas (KIM; RICHARDSON; SLAVíK, 2017).

Figura 5 – Exemplos de configurações de fontes de EOC. (a) PM individual. (b) IM individual. (c) DDMZM individual. (d) Cascata PM-PM-IM. (e) Cascata PM-IM-IM.



Elaborado pelo autor.

2.1.3 Vantagens dos EOCs em Comparação com Fontes Habituais

Além dos EOMs, existem outras configurações que podem gerar um OFC. As mais conhecidas são as que utilizam os chamados lasers pulsados (*mode-locked lasers* — MLL) (KELLER, 2021) e os micro-ressonadores (KIPPENBERG et al., 2011). A seguir são apresentadas algumas das vantagens dos EOMs em comparação a outras fontes de OFC.

- **Capacidade de Sintonização:**
  Uma das primeiras vantagens dos EOCs é que eles são originados de um laser CW que frequentemente é escolhido sintonizável em frequência. Dessa forma, o pente produzido por alguma configuração eletro-óptica também é sintonizável em frequência, uma vez que os EOMs geralmente têm uma larga largura de banda de trabalho em frequência. Além disso, a taxa de repetição $f_m = \frac{\omega_m}{2\pi}$ dos EOCs também é facilmente sintonizável, por ser fornecida por um gerador de formas de onda elétricas. Portanto, a taxa de repetição pode ser facilmente ajustada agindo diretamente nos dispositivos puramente eletrônicos.

- **Planicidade Espectral:**
  Comparados a outras fontes de pentes de frequência, os EOMs possuem vários graus de liberdade para moldar o pente gerado, os quais podem ser utilizados para obter um pente com baixas variações de intensidade entre as linhas, sendo a forma ideal um pente puramente plano. Técnicas como a cascata de moduladores, o uso de moduladores Mach-Zehnder e a otimização do sinal de controle são algumas das possibilidades para planificar o espectro dos pentes.

2.1.4 Aplicações dos EOCs

Os OFCs primeiro revelaram seu potencial em metrologia de frequência (UDEM et al., 1999), mas muitas outras aplicações foram desenvolvidas posteriormente. Diversas aplicações em espectroscopia de alta resolução e calibração, medição de distâncias, imageamento óptico (IDEGUCHI et al., 2013).

Os EOCs podem ser bastante eficazes na geração de formas de onda arbitrárias, devida a sua grande versatilidade. O formato espectral do pente pode ser controlado com o ajuste de parâmetros (tensão dos sinais de controle, tensões de polarização, etc.) ou utilização de outras configurações de gerador (e.g cascata de moduladores), e essas técnicas podem ser usadas para geração de formas de ondas particulares no domínio do tempo como pulsos gaussianos ou triangulares (DAI et al., 2013). A possibilidade do controle do formato espectral do pente permite diversas formas de filtragem e amostragem úteis no processamento de sinais (TORRES-COMPANY; WEINER, 2014).

Em telecomunicações, os EOMs são excelentes fontes para sistemas de multiplexação por divisão de comprimento de onda (*Wavelength-Division Multiplexing* — WDM), uma vez que é possível gerar EOCs com muitas linhas de pentes.

Essas linhas de pentes podem ser vistas como componentes de frequência coerentes, caracterizados por serem igualmente espaçados ou de fase trancada (*phase-locked*), evitando desvio de canal e permitindo um denso empacotamento de portadoras além do que pode ser alcançado usando lasers independentes (LUNDBERG et al., 2018), tornando-os ideais como frequências portadoras do WDM. Devido à pequena faixa de guarda necessária entre os canais modulados com pentes (MILLAR et al., 2016), a eficiência espectral da modulação pode aumentar consideravelmente, permitindo sistemas de taxas mais altas.

2.1.5 Controle de Parâmetros de EOMs

Os EOMs destacam-se na geração de OFCs devido à sua simplicidade e à capacidade de ajustar o espaçamento entre as linhas do pente de forma flexível. No entanto, mesmo em geradores aparentemente simples, a natureza não linear do sistema e a distribuição desigual de potência ao longo do pente representam desafios no controle de parâmetros para obtenção de OFCs com o espectro desejado.

Com o aumento da complexidade da tecnologia de geração de OFC, especialmente quando impulsionada por múltiplos sinais de RF, métodos tradicionais como o controle variável, que ajusta um parâmetro por vez e exige experiência extensiva em pesquisa para determinar os parâmetros-chave, ou métodos mais recentes que utilizam algoritmos de otimização, como o algoritmo de evolução diferencial (PENDIUK; NEVES; POHL, 2018) por exemplo, enfrentam dificuldades para encontrar soluções ótimas. A complexidade escalonada com o aumento dos parâmetros, a suscetibilidade a soluções ótimas locais e a demanda por muitas iterações consomem muito tempo e recursos computacionais, tornando inviável o fornecimento de parâmetros em tempo hábil.

Uma técnica que se mostrou muito eficiente no controle de parâmetros de OFCs é o chamado *Design* Inverso (*Inverse Design*) (MA et al., 2022) baseado em redes de aprendizado profundo (*Deep Learning*). Ao aproveitar a capacidade única de aprendizado de características, a poderosa capacidade de modelagem e a habilidade de generalização do aprendizado profundo, essa abordagem visa revolucionar o *design* de parâmetros dos geradores de OFC. O método demanda um esforço singular e temporário, que está concentrado na etapa de treinamento, e embora seja exigente inicialmente, permite uma vantagem significativa e duradoura em termos de tempo para o fornecimento de parâmetros, comparado aos métodos tradicionais.

2.2 **Aprendizado de Máquina - Machine Learning**

O aprendizado de máquina é uma área da inteligência artificial que está transformando a forma os computadores processam dados. Em vez de serem programados com regras específicas, os sistemas de aprendizado de máquina são treinados para aprender com exemplos passados, permitindo que identifiquem padrões e façam previsões ou tomem decisões com base nesses padrões. Esse campo tem aplicações em diversas áreas, desde reconhecimento de padrões e análise de dados até processamento de linguagem natural e robótica, impulsionando avanços significativos em campos como medicina, finanças e tecnologia.

2.2.1 Redes Neurais

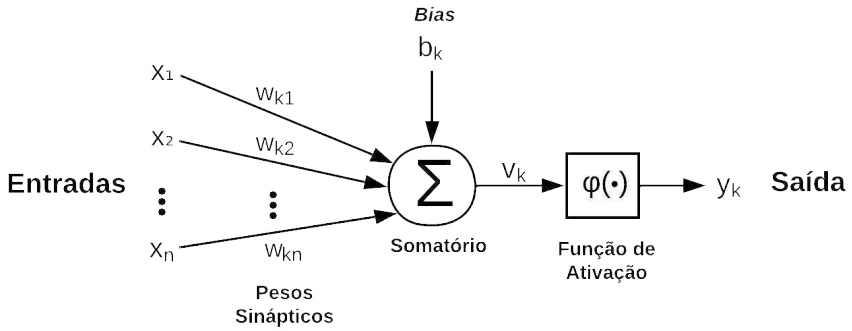
As redes neurais artificiais ganham esse nome devido à inspiração no funcionamento do cérebro humano. Elas consistem em uma coleção de neurônios interconectados, organizados em camadas.

Um neurônio (*perceptron*), ilustrado na Figura 6, pode ser representado matematicamente por uma função de n entradas x multiplicadas por pesos w_k, somadas e adicionadas a um viés (*bias*) b_k resultando em v_k.

$$v_k = b_k + \sum_{i=1}^{n} x_i w_{ki} \tag{10}$$

O resultado desse somatório (v_k) é transformado pela chamada função de ativação $\varphi(\bullet)$ que por fim resulta na saída y_k. O objetivo da função de ativação é introduzir não-linearidade

Figura 6 – Ilustração do modelo matemático genérico de um neurônio artificial



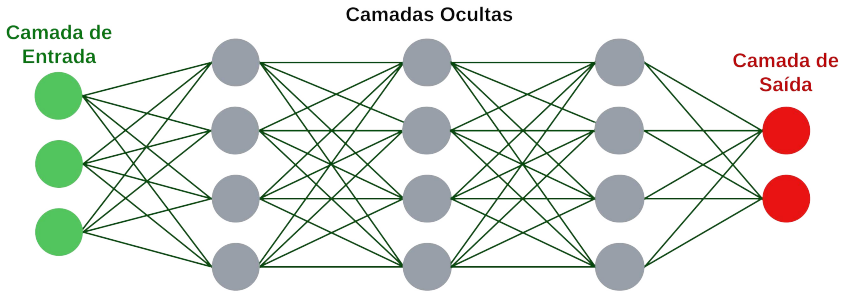
Elaborado pelo autor.

ao modelo, pois isso é crucial para permitir que a rede neural aprenda e represente relações complexas entre os dados de entrada e a saída desejada.

$$y_k = \varphi(v_k) \tag{11}$$

A combinação de neurônios é chamada de perceptron de múltiplas camadas (*Multi-Layer Perceptron* — MLP) ou simplesmente rede neural (*Neural Network* — NN). A Figura 7 mostra a ilustração de uma NN onde se pode observar a presença da camada de entrada, da camada de saída, e das camadas ocultas (*hidden layers*).

Figura 7 – Ilustração de uma rede neural



Elaborado pelo autor.

2.2.2 Redes Neurais de Aprendizado Profundo

As redes neurais de aprendizado profundo, também conhecidas como redes neurais profundas, são uma forma avançada de redes neurais que consistem em muitas camadas ocultas.

Dentre as várias estruturas existentes na área de aprendizado profundo, existem as chamadas redes neurais de alimentação direta, ou *Feedforward Neural Networks* (FNN), configuram o tipo mais básico de rede neural profunda e seguem a estrutura de uma rede neural (NN) apresentada na Figura 7.

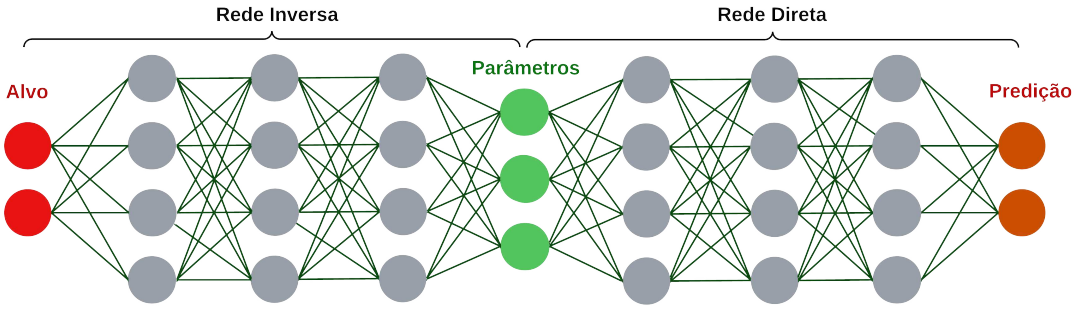
Segundo a disponibilidade de dados e o método de treinamento utilizado, as técnicas de aprendizado de máquina são separadas em diferentes categorias (GOODFELLOW et al., 2016), sendo o aprendizado supervisionado a mais comum. Tal categoria envolve treinar algoritmos

com dados rotulados, onde cada amostra de treinamento consiste em uma entrada e uma saída desejada. É usado em problemas de classificação para previsão de valores discretos e em problemas de regressão para previsão de valores numéricos contínuos.

No contexto das FNN, existe uma rede neural específica chamada de autocodificador, formada por duas FNN em série: codificador e decodificador. A primeira rede visa comprimir os dados de entrada, e a segunda rede visa reconstruir os dados comprimidos. Geralmente, essa rede encontra aplicações em reconhecimento de padrões e remoção de ruído em imagens (HALES et al., 2020). O treinamento do codificador e do decodificador é feito simultaneamente, e por isso o funcionamento do decodificador depende do codificador, uma vez que os dados de entrada do decodificador não têm um sentido palpável.

A técnica de design inverso (MA et al., 2022) é uma adaptação de uma rede de autocodificador onde o codificador e o decodificador são treinados separadamente. Primeiramente, é feito o treinamento supervisionado da rede direta, ou seja, as entradas e saídas do sistema que se deseja modelar são conhecidas. Em seguida, é feito o treinamento da rede inversa, onde suas entradas serão as saídas da rede direta já treinada. A rede de principal interesse nesse caso é a rede inversa, que poderá conseguir realizar o design inverso do sistema, ou seja, a partir da saída desejada para o sistema, a rede inversa poderá devolver a entrada necessária.

Figura 8 – Ilustração cascata de redes neurais para o treinamento com a técnica de *design* inverso



Elaborado pelo autor.

O desempenho da rede é constantemente observado durante o seu treinamento para evitar que o modelo permaneça na zona de subajuste ou entre na zona de sobreajuste. A zona de subajuste, também chamada de *underfitting*, ocorre quando o modelo está mal ajustado e precisa geralmente de mais épocas de treinamento. Já a zona de sobreajuste, também chamada de *overfitting*, é quando o modelo tem baixa capacidade de generalização por estar muito ajustado aos dados de treinamento. Normalmente, é desejado que o modelo seja treinado até atingir uma zona entre o *underfitting* e o *overfitting*, onde ele pode generalizar bem para dados não vistos.

Para o treinamento supervisionado de uma FNN, antes é feita a construção do conjunto de dados e a definição do modelo de rede. O conjunto de dados é construído a partir informações de entrada e saída do sistema pré-processadas para facilitar o treinamento. A definição do modelo de rede é feito com a escolha dos chamados hiperparâmetros, os quais são parâmetros

definidos antes do processo de treinamento que controlam o comportamento do algoritmo de aprendizado.

Diferentemente dos parâmetros do modelo (como os pesos das conexões na rede neural, e os *bias* de cada neurônio), sendo ajustados automaticamente durante o treinamento, os hiperparâmetros são definidos pelo projetista. Alguns dos hiperparâmetros mais comuns incluem:

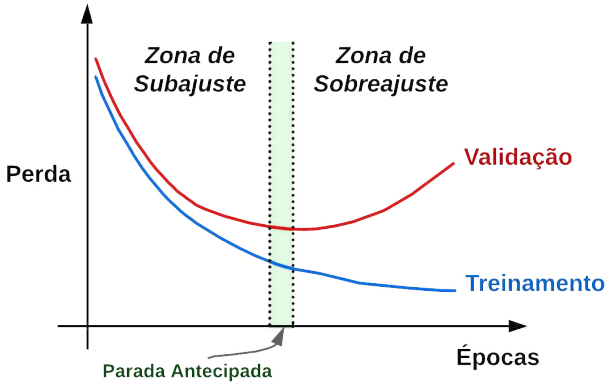
- **Taxa de Aprendizado ($Learning\ Rate$):**
  Controla o tamanho dos passos que o algoritmo dá ao ajustar os pesos com base no gradiente do erro. Uma taxa de aprendizado muito alta pode fazer o treinamento divergir, enquanto uma taxa muito baixa pode tornar o treinamento muito lento.
- **Número de Épocas de Treinamento($Epochs$):**
  Refere-se ao número de vezes que o algoritmo de aprendizado passa por todo o conjunto de dados de treinamento. Mais épocas podem melhorar o aprendizado, ou seja, tirar o modelo da zona de *underfitting*, mas também podem levá-lo *overfitting*(Figura 9).
- **Tamanho do Lote ($Batch\ Size$):**
  Define o número de amostras de treinamento usadas para calcular o gradiente antes de atualizar os pesos. Um tamanho de lote pequeno pode fornecer atualizações mais frequentes e detalhadas, enquanto tamanhos de lote maiores podem ser mais eficientes em termos de computação.
- **Número de Camadas e Neurônios por Camada:**
  Compõem a arquitetura da rede, definindo quantas camadas ocultas a rede possui e quantos neurônios há em cada camada. Isso determina a capacidade da rede de modelar funções complexas.
- **Função de Ativação:**
  É a função não-linear que determina a saída de um neurônio dado um conjunto de entradas. Exemplos incluem *ReLU*, *sigmoid* e *tanh*. A escolha da função de ativação pode afetar a convergência e a capacidade da rede de aprender.
- **Função Custo:**
  Mede o quão bem a rede neural está se saindo na tarefa de aprendizado. Exemplos incluem o erro quadrático médio (*Mean Squared Error* — MSE) ou erro absoluto médio (*Mean Absolute Error* — MAE) para problemas de regressão, e entropia cruzada (*cross-entropy*) para problemas de classificação.
- **Algoritmo de Otimização:** Refere-se ao método usado para ajustar os pesos da rede durante o treinamento. Exemplos incluem o SGD (*Stochastic Gradient Descent*), Adam, RMSprop, Adagrad. Algoritmos mais avançados como Adam podem ajustar a taxa de aprendizado de forma automática e geralmente convergem mais rapidamente.
- **Regularização:**
  Trata-se de métodos usados para prevenir o *overfitting*, penalizando grandes pesos e/ou desligando neurônios aleatoriamente durante o treinamento.

- **Momentum:**
  É uma técnica que ajuda a acelerar o treinamento e a convergência, acumulando uma fração do gradiente anterior nas atualizações de peso atuais.
- **Decaimento da Taxa de Aprendizado (*Learning Rate Decay*):**
  Trata-se de uma técnica que diminui a taxa de aprendizado ao longo do tempo para permitir passos de ajuste menores conforme o treinamento progride, auxiliando a rede a se estabilizar em um mínimo local.

A escolha e a configuração adequada dos hiperparâmetros são cruciais para o desempenho da rede neural. Frequentemente, a otimização dos hiperparâmetros envolve experimentação e técnicas de busca, como busca em grade (*Grid Search*), busca aleatória (*Random Search*), ou até mesmo buscas mais inteligentes, como a baseada em Otimização Bayesiana (*Bayesian Optimization*) (SNOEK et al., 2012).

É comum dividir o conjunto de dados em três partes: conjunto de treinamento (usado para treinar o modelo), conjunto de validação (usado para ajustar hiperparâmetros e evitar sobreajuste), e conjunto de teste (usado para avaliar a desempenho final do modelo). Uma divisão comum consiste em utilizar 80% do conjunto de dados para treinamento, 10% para validação e 10% para teste.

Figura 9 – Ilustração das curvas de aprendizado de uma rede neural profunda



Elaborado pelo autor.

Os valores da função de custo utilizada no treinamento e validação, são observados durante o treinamento da rede para monitorar a evolução do modelo e realizar ajustes nos hiperparâmetros. A Figura 9 mostra a ilustração das curvas de perda de treinamento e validação, onde geralmente é desejado que o modelo seja treinado até uma época em que não haja mais uma diminuição considerável da perda de validação, alcançando um estado entre o *underfitting* e o *overfitting*.

Para decidir quando parar o treinamento, é geralmente utilizada alguma técnica de parada antecipada (*Early Stopping*) (PRECHELT, 2000). A mais comum consiste em identificar a desaceleração do decréscimo da perda de validação, realizando a contagem de épocas até que ocorra um novo mínimo. Se a contagem ultrapassar o limite estabelecido, o treinamento é interrompido, indicando que o modelo alcançou uma faixa de melhor desempenho.

3 METODOLOGIA

Os EOMs destacam-se na geração de OFCs devido à sua simplicidade e à capacidade de ajustar o espaçamento entre as linhas do pente. No entanto, mesmo em geradores aparentemente simples, a natureza não linear do sistema e a distribuição desigual de potência apresentam desafios no controle de parâmetros para obter o espectro desejado. Métodos tradicionais, como o método da variável de controle ou mesmo os métodos mais recentes que utilizam algoritmos de otimização, enfrentam dificuldades com a complexidade crescente e a necessidade de muitas iterações, tornando-os lentos e custosos. Nesse contexto, o *Design* Inverso baseado em *Deep Learning* mostra-se uma solução eficiente, oferecendo uma modelagem poderosa e capacidade de generalização, proporcionando vantagens significativas em termos de tempo e eficiência em comparação com métodos tradicionais, apesar do alto esforço inicial para o treinamento.

Neste capítulo, é abordada a metodologia adotada para o treinamento da rede inversa utilizando técnicas de aprendizado profundo. A primeira seção trata do gerador de EOC escolhido, destacando os dados de entrada e saída da rede. Em seguida, apresenta-se a construção do conjunto de dados a ser utilizado para o treinamento, validação e teste da rede. Após a construção do conjunto de dados, é necessária a definição dos hiperparâmetros dos modelos de rede direta e rede inversa. Com o conjunto de dados e os hiperparâmetros da rede definidos, segue-se a etapa de treinamento, onde a rede direta é treinada para, por fim, auxiliar no treinamento da rede inversa.

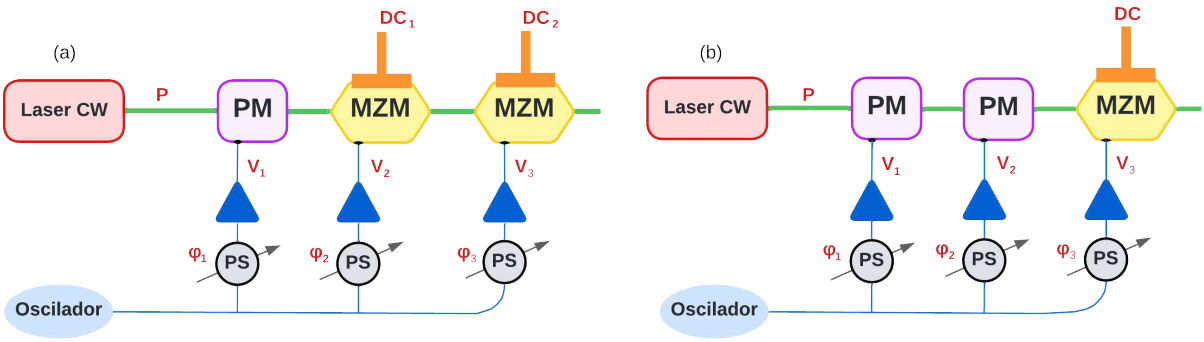
Todos os arquivos produzidos para a aplicação da metologia estão disponibilizados no repositório público deste trabalho (LIMA, 2024).

3.1 Gerador de Pentes Eletro-Ópticos

Foram escolhidos dois casos de gerador de EOCs para o treinamento de redes inversas com a técnica de *Inverse Design*. O primeiro caso é uma cascata de um modulador de fase (PM) seguido de dois moduladores de intensidade (IM) (Figura 10a). O segundo caso é uma cascata de dois moduladores de fase (PM) seguidos de um modulador de intensidade (IM) (Figura 10b). Os moduladores de intensidade utilizados são moduladores de Mach-Zehnder *single-drive* MZM em modo *push-pull*.

As configurações da Figura 10 exibem alguns dos parâmetros que podem ser utilizados no controle do espectro óptico dos EOCs. No primeiro caso (Figura 10a), têm-se os parâmetros de: amplitude (V_1, V_2 e V_3) e fase (φ_1, φ_2 e φ_3) do sinal elétrico do oscilador, tensões de polarização (DC_1 e DC_2) e potência do *laser* CW (P). O segundo caso (Figura 10b) possui parâmetros similares, sendo que há apenas uma tensão de polarização (DC).

Figura 10 – Ilustração dos casos escolhidos para treinamento de redes inversas com a técnica de *Inverse Design*. (a) Cascata PM-MZM-MZM. (b) Cascata PM-PM-MZM



Elaborado pelo autor.

3.2 **Conjunto de Dados**

Foi construído um conjunto de dados (*dataset*) para cada caso da Figura 10. Dentre os parâmetros dessas configurações, o parâmetro de potência do *laser* CW (P) foi desconsiderado, uma vez que o mesmo não altera o formato dos EOCs, mas apenas controla a potência média do espectro (algo que pode ser feito manualmente). Assim são 8 parâmetros a serem usados no primeiro caso, e 7 parâmetros no segundo caso.

3.2.1 Simulação de Pentes Eletro-Ópticos

As duas configurações foram simuladas numericamente com base nas equações de amplitude do campo óptico no domínio do tempo do PM e do IM, as quais são respectivamente as Equações 12–13, baseadas nas Equações (1) e (7):

$$A(t) = E_i \times e^{iKV(t))}, \tag{12}$$

$$A(t) = E_i \times cos(K(V(t) + V_b)), \tag{13}$$

onde $V(t) = V_0 sin(2\pi f_m t + \varphi_0)$ é o sinal de controle do modulador, E_i é o sinal modulante, $K = \frac{\pi}{V\pi}$ é o índice de modulação, e V_b é a tensão de polarização do MZM.

Foi decidido analisar 11 linhas dos pentes gerados aleatoriamente com distribuição uniforme nos limites:

- $[0, 3]$ V para os parâmetros de amplitude de tensão (V_1, V_2 e V_3) – O sistema se mostrou indiferente a polaridade da tensão de controle, logo apenas valores maiores que 0 V foram considerados. O limite superior foi definido com base na quantidade de linhas de pentes a serem analisadas. A amplitude máxima de 3 V foi suficiente para gerar onze linhas de potências semelhantes.
- [0, 2π] rad para os parâmetros de fase (φ_1, φ_2 e φ_3) – O sistema se mostrou periódico em relação à fase do sinal elétrico de controle com período 2π.
- [0, $4V_\pi$] V para os parâmetros de polarização (DC_1, DC_2 e DC) – O sistema se mostrou periódico em relação à tensão de polarização dos MZM com período $4V_\pi$.

Por simplicidade, os parâmetros V_π dos moduladores foram ambos definidos como $2\ V$, e a taxa de repetição $f_m = \frac{\omega_m}{2\pi}$ do sinal do oscilador foi definida como $10\ GHz$, a qual é uma largura de banda usual para um canal óptico. Outras configurações de parâmetros mais específicos do simulador podem ser conferidas diretamente nos códigos e arquivos disponibilizados no repositório deste trabalho (LIMA, 2024).

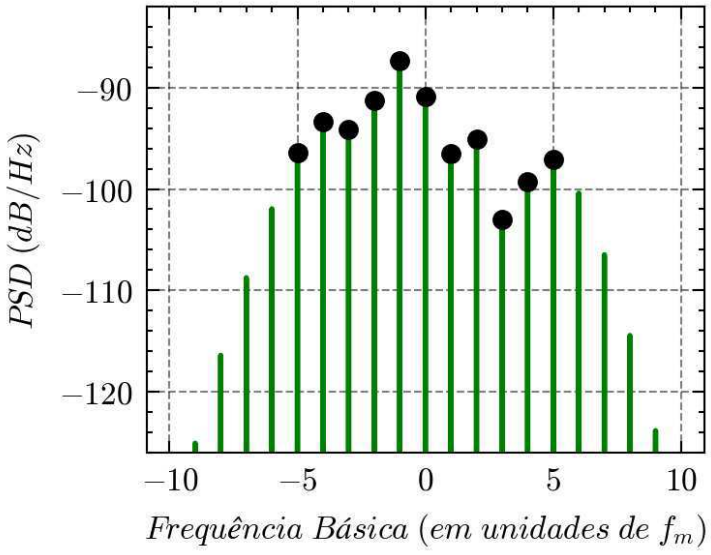
3.2.2 Criação do Conjunto de Dados

O conjunto de dados foi criado em um formato compatível para uso com a biblioteca PyTorch. O tamanho de tal conjunto de dados foi definido após testes de generalização com alguns modelos iniciais. E esse tamanho foi aumentado até que não houvesse mais melhoria na capacidade de generalização do modelo. Por fim, o *dataset* foi definido com 375.000 amostras, sendo ainda somado a mais um conjunto de dados de EOCs planos, e por fim dividido em três partes: 80% para treinamento, 10% para validação e 10% para teste.

3.2.2.1 Coleta dos Picos dos Pentes

Foram geradas 375.000 amostras variando os parâmetros do gerador de EOCs nos intervalos já citados. Essas amostras são os chamados *inputs* ou dados de entrada. Para a montagem do conjunto de dados, apenas os valores de pico das linhas dos pentes serão considerados os dados de saída do gerador. Assim, de cada um dos pentes do *dataset*, foram extraídos os valores de pico das onze linhas centrais, tendo em mente que tais linhas estão separadas entre si por f_m. A Figura 11 mostra um caso de OFC onde os picos centrais a serem coletados estão marcados.

Figura 11 – Ilustração de um EOC com os picos de algumas linhas centrais destacados



Elaborado pelo autor.

3.2.2.2 Pré-Processamento dos Dados

Em seguida a etapa de geração dos dados, foi realizado o pré-processamento dos dados de saída. Essa etapa consistiu em remover a média dos picos de cada pente e normalizá-los para o intervalo [0, 1].

A média dos picos de cada pente foi removida, pois a potência do *laser* CW pode controlar essa média sem afetar o formato espectral dos pentes, tornando a informação da média desnecessária para o caso onde apenas o formato do EOC é importante.

Após a remoção da média, os dados passaram pela etapa de normalização, a qual é uma técnica de pré-processamento que pode melhorar o desempenho de treinamento dos modelos de aprendizado de máquina. Em geral, é comum normalizar os dados para estarem nas faixas de [-1, 1] ou [0, 1]. O método de normalização varia a depender da natureza dos dados e do algoritmo utilizado. Foi escolhido o método de normalização pelo valor máximo absoluto do conjunto de dados:

$$X_i' = \frac{X_i}{X_{max}}, \tag{14}$$

onde X_i é uma amostra de saída, X_{max} é o maior valor encontrado em todo o conjunto de dados. Por fim, ainda foi feita a operação $X_i' = (X_i' + 1)/2$ para colocar os dados na faixa de [0, 1].

3.2.3 Adição de Pentes Eletro-Ópticos Planos

Foi constatado que, dentro das 375.000 amostras geradas, poucas delas resultaram em pentes de espectro plano (*flat combs*), os quais são EOCs importantes para o aprendizado das redes devido à sua vasta gama de aplicações. Para garantir a maior representatividade das configurações pelo conjunto de dados, foi necessário produzir EOCs planos.

Para geração de EOCs planos, foi utilizado um algoritmo genérico de minimização, com a função alvo sendo a variância dos picos, uma vez que um OFC plano perfeito têm variância nula. No entanto, qualquer OFC que apresente uma diferença entre o maior e menor pico menor que 1 dB, já é considerado um OFC plano.

Foram gerados 40.000 amostras de EOCs planos para a primeira configuração (Figura 10a), e apenas cerca de 700 amostras de EOCs planos para a segunda configuração (Figura 10b). Como a configuração do primeiro caso tem um modulador de intensidade a mais, tal configuração se torna mais versátil para a criação de pentes planos, e por isso foi possível gerar muito mais amostras que no caso da segunda configuração.

3.3 **Definição de Hiperparâmetros**

A escolha e configuração adequada dos hiperparâmetros são cruciais para o desempenho da rede neural. Alguns desses hiperparâmetros podem ser guiados pela natureza dos dados e pelo poder computacional disponível para o treinamento, como é o caso da função custo,

da função de ativação e do algoritmo de otimização. Por outro lado, há hiperparâmetros cuja escolha não é tão imediata que precisam ser experimentados para verificar quais configurações proporcionam o melhor desempenho da rede. Esse é o caso de hiperparâmetros como a taxa de aprendizado do algoritmo de otimização, o número de camadas e de parâmetros da rede, e o tamanho do lote de dados do treinamento.

3.3.1 Função Custo

Sabendo que o treinamento será feito pelo método de aprendizado supervisionado, e que se trata de um problema de regressão e não de classificação, ou seja, os dados são valores reais, as possibilidades de funções de custo se limitam às que trabalham nesse domínio.

As funções de custo mais comuns e simples para problemas de regressão são a do erro quadrático médio (*Mean Squared Error* — MSE) e a do erro absoluto médio (*Mean Absolute Error* — MAE). O MSE foi escolhido para treinamento por apresentar uma penalização de erros não linear, que faz com que grandes erros sejam mais fortemente penalizados que pequenos erros, incentivando o modelo a ser mais preciso.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{15}$$

A Equação 15 mostra a fórmula da função custo MSE, onde n é o número de amostras, y_i é o valor real da i-ésima amostra e $\hat{y}_i$ é o valor previsto pelo modelo para a i-ésima amostra.

A função custo MAE, que é similar à Equação 15 (exceto por usar o valor absoluto em vez do quadrado da diferença), se mostra mais vantajosa em casos onde os dados podem conter os chamados *outliers*, ou pontos fora da curva, que podem ocorrer mais facilmente quando o *dataset* é criado com medições experimentais, o que não é o caso deste trabalho.

Outra função que pode ser interessante em casos onde o *dataset* é criado com medições experimentais é a função de erro Huber, que combina as vantagens do MSE e do MAE, utilizando o MSE para pequenos erros e o MAE para grandes erros.

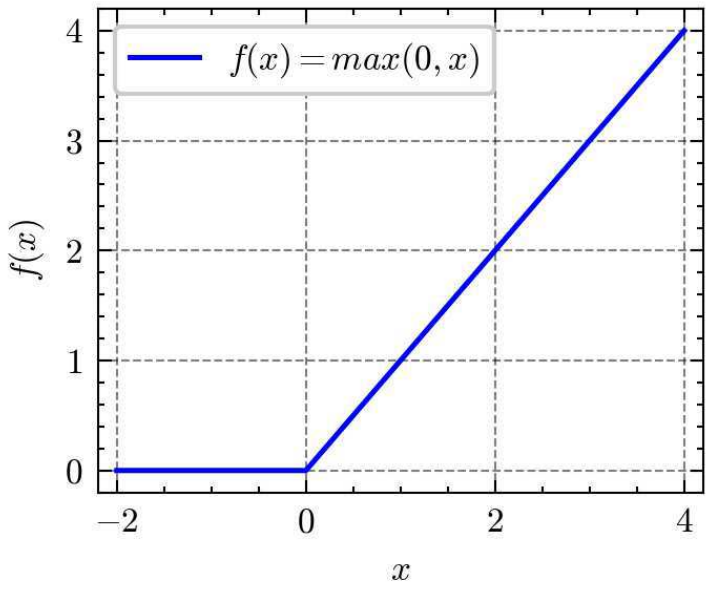
3.3.2 Função de Ativação

Como os dados de saída da rede foram normalizados para valores no intervalo de [0, 1], a função de ativação mais simples e eficaz nesse caso é a *ReLU* (*Rectified Linear Unit*), que pode ser implementada como $f(x) = max(0, x)$, que retorna o valor 0 para entradas menores que 0, ou o próprio valor do neurônio caso contrário. A Equação 16 descreve a função ReLU, que também está ilustrada na Figura 12.

$$f(x) = \begin{cases} 0, & \text{se } x < 0 \\ x, & \text{caso contrário} \end{cases}, \tag{16}$$

onde x é o resultado do neurônio. A *ReLU* é a função de ativação mais comum em redes neurais profundas e tem sido amplamente adotada devido à sua capacidade de lidar bem com o problema de dissipação de gradientes (HOCHREITER, 1998) e eficiência computacional.

Figura 12 – Ilustração da função de ativação retificadora *ReLU*



Elaborado pelo autor.

3.3.3 Algoritmo de Otimização

O algoritmo de otimização diz respeito ao método usado para ajustar os pesos da rede durante o treinamento. Os mais utilizados em problemas de regressão, devido à sua eficácia comprovada e à sua capacidade de lidar com uma variedade de cenários de treinamento, são: o Gradiente Descendente (*Gradient Descent* — GD), o Gradiente Descendente Estocástico (*Stochastic Gradient Descent* — SGD) e o Adam (*Adaptive Moment Estimation*). No entanto, para grandes conjuntos de dados como o deste trabalho, os algoritmos mais eficientes são o SGD e o Adam.

Comparado ao SGD, o algoritmo Adam oferece vantagens adicionais que podem resultar em convergência mais rápida e eficiente, especialmente em problemas mais complexos de aprendizado de máquina que possuam muitos parâmetros.

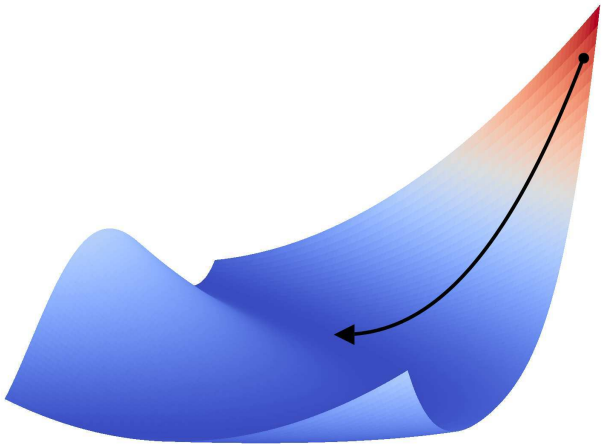
Ambos os algoritmos foram testados neste trabalho e, de fato, o Adam se mostrou mais vantajoso por ter uma convergência mais rápida sem maior custo computacional, e por isso foi o algoritmo de otimização escolhido para o treinamento das redes. A Figura 13 mostra a ilustração de um algoritmo de otimização em busca do valor mínimo do espaço de soluções.

3.3.4 Sintonização de Hiperparâmetros por Otimização Bayesiana

Alguns hiperparâmetros têm um número limitado de opções e podem ter sua escolha guiada pela natureza do problema em questão, como é o caso da função custo, da função de ativação e do algoritmo de otimização. No entanto, existem hiperparâmetros numéricos, para os quais há infinitas possibilidades de escolha.

No caso da definição de hiperparâmetros numéricos, uma abordagem comum é realizar diversos testes em limites pré-estabelecidos, nos quais se imagina que os valores ideais possam

Figura 13 – Ilustração de um algoritmo de otimização em busca do valor mínimo no espaço de soluções



Elaborado pelo autor.

estar. A busca pelos hiperparâmetros ideais dentro desses limites pode ser feita sequencialmente, testando todas as possibilidades do conjunto (*Grid Search*), de forma aleatória, testando algumas possibilidades aleatórias do conjunto (*Random Search*), ou utilizando alguma forma de otimização que busque as melhores combinações, como o método de otimização bayesiana (*Bayesian Optimization*).

O método *Grid Search* garante que todas as possibilidades do conjunto estabelecido sejam testadas, exigindo um grande custo de tempo para obtenção de resultados promissores. Já o método de *Random Search*, por mais que não seja tão sistemático quanto outras abordagens, pode ser eficaz para explorar o espaço de hiperparâmetros de forma mais ampla. A combinação da busca aleatória, aliada a algum método de otimização, como o *Bayesian Optimization*, pode ser a escolha ideal para a obtenção de ótimos resultados com baixo custo de tempo.

O método de busca baseado em Otimização Bayesiana (HEATON, 2023) foi utilizado para a sintonização de hiperparâmetros como a taxa de aprendizado (*learning rate*), número de camadas e de parâmetros da rede, e tamanho do lote de dados (*batch size*) de treinamento.

3.3.4.1 Taxa de Aprendizado

Os algoritmos de otimizadores, em geral, têm seus próprios parâmetros que são hiperparâmetros do modelo de rede neural. Alguns desses hiperparâmetros de otimizador já estão pré-definidos com valores ideais para os casos de aprendizado de máquina, como é também o caso do otimizador Adam (KINGMA; BA, 2014).

Em geral, o hiperparâmetro mais trabalhado em um algoritmo de otimização é a taxa de aprendizado, que pode ser um valor constante ao longo do treinamento, ou pode possuir um fator de decaimento. O algoritmo do SGD, por exemplo, utiliza uma taxa de aprendizado fixa, enquanto o Adam, ajusta automaticamente a taxa de aprendizado de cada um dos parâmetros da rede (pesos e vieses) com base num valor inicial.

A taxa de aprendizado determina a rapidez com que um modelo se ajusta aos dados.

Um valor muito alto pode fazer o algoritmo oscilar ou divergir, enquanto uma taxa muito baixa pode resultar em um treinamento lento, ficando preso em mínimos locais sem alcançar o mínimo global da função custo.

É comum explorar taxas de aprendizado pequenas, e um valor inicial padrão pode ser $\alpha = 0.001$. Para a busca utilizando Otimização Bayesiana foi utilizado o intervalo $[10^{-6}, 10^{-3}]$.

3.3.4.2 Estrutura das Redes

Uma estrutura de rede neural pode ser definida a partir da quantidade de camadas ocultas, da quantidade de neurônios, e como esses neurônios estão distribuídos entre as várias camadas da rede.

Não existe uma regra para a definição da distribuição dos neurônios entre as camadas ocultas, geralmente isso é determinado de forma empírica por meio de experimentações. Nesse trabalho foram experimentadas distribuições não uniformes e distribuições uniformes dos neurônios dentre as camadas ocultas, e a distribuição uniforme se mostrou mais vantajosa por ser mais simples e apresentar praticamente os mesmos resultados dos melhores casos de distribuições não uniformes.

O número de parâmetros de uma rede neural totalmente conectada (*Fully Connected Network*), é a quantidade de pesos somada quantidade de *bias* (KHANNA, 2020), e pode ser definido pela Equação 17.

$$N = (I+1)H_1 + (H_n+1)O + \sum_{i=1}^{n-1}(H_i+1)H_{i+1},\ \ n > 0, \tag{17}$$

onde I é o número de neurônios de entrada da rede, O é o número de neurônios de saída, H_i é o número de neurônios da i-ésima camada oculta, e n é o número de camadas ocultas. No caso da distribuição uniforme de neurônios dentre as camadas ocultas têm-se $H_i = H_{i+1} = H_1 = H_n = H$, e substituindo na Equação 17, obtêm-se:

$$N = (n-1)H^2 + (I+O+n)H + O, \tag{18}$$

onde H é o número de neurônios por camada. Pode-se ainda encontrar a expressão de H em função de N, resolvendo a Equação quadrática da Equação 18:

$$H = -\frac{I+O+n}{2(n-1)} + \sqrt{\frac{(I+O+n)^2}{4(n-1)^2} - \frac{O-N}{n-1}}. \tag{19}$$

- **Número de Camadas Ocultas:** Para a busca utilizando Otimização Bayesiana foram experimentadas redes com número de camadas ocultas n no intervalo de [3, 9].
- **Número de Parâmetros da Rede:** Dado um número de camadas ocultas n, pode-se experimentar a busca pela estrutura de rede ideal utilizando Otimização Bayesiana para variar o número de camadas H, ou o número de parâmetros da rede N, seguindo a Equação 19. Escolheu-se variar diretamente o H, limitado no intervalo de [128, 512].

3.3.4.3 Tamanho do Lote de Dados

A escolha do tamanho do lote de dados (*batch size*) é geralmente feita por experimentação. O *batch size* afeta diretamente a eficiência computacional, a memória utilizada e a qualidade do treinamento.

Lotes de dados pequenos têm a vantagem de introduzir mais variações no treinamento, o que pode ajudar a evitar mínimos locais. No entanto, um *batch size* pequeno significa que poucos dados estão sendo usados ao mesmo tempo, resultando em menor uso de memória durante o treinamento, mas fazendo com que cada época dure mais tempo.

Lotes de dados grandes têm a vantagem de proporcionar maior eficiência computacional, especialmente em hardware otimizado para processamento em paralelo, como os que possuem GPU (*Graphics Processing Unit*) dedicada, fazendo com que o treinamento seja mais rápido. No entanto, são feitas menos atualizações de gradiente por época, aumentando as chances do modelo ficar preso em mínimos locais.

Assim, é comum utilizar os chamados *mini-batches*, os quais são lotes de tamanhos intermediários, variando tipicamente entre 32 e 512 amostras e, por isso, na busca desse hiperparâmetro por Otimização Bayesiana, foi utilizado o intervalo de tamanho de lote de [32, 512].

3.4 **Treinamento**

Dadas as configurações da Figura 10, o objetivo final do treinamento é conseguir uma rede neural que possa receber, como entrada, os picos de um EOC e devolver, na saída, os parâmetros dos moduladores necessários para produzir o formato de EOC desejado. Para auxiliar no treinamento dessa rede, chamada rede inversa, é necessário primeiro treinar a rede direta. A rede direta irá imitar os geradores de EOC com base nos conjuntos de dados de treinamento, devolvendo, na saída, os picos de pente que correspondem a um EOC gerado com os parâmetros dos moduladores recebidos na entrada.

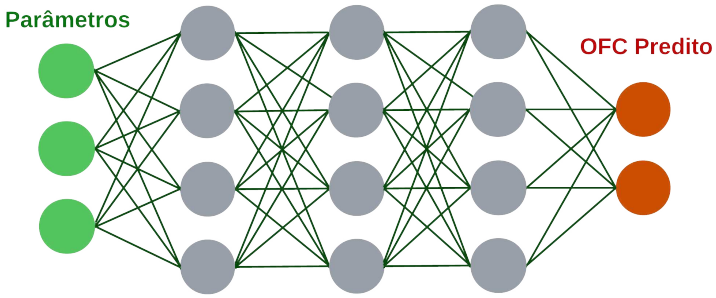
A linguagem de programação *Python* foi escolhida para implementação dos algoritmos de treinamento e simulação dos dados devido à sua facilidade de uso (FOUNDATION, 2024). Para o tratamento de redes neurais foi escolhida a biblioteca *PyTorch*, a qual é uma biblioteca de aprendizado de máquina de código aberto (PASZKE et al., 2019) desenvolvida pelo grupo de pesquisa em inteligência artificial do *Facebook*, atual *Meta*, conhecido como *Facebook AI Research* (FAIR). O *PyTorch* é amplamente utilizado por pesquisadores e engenheiros de aprendizado de máquina devido à sua facilidade de uso, flexibilidade e suporte para computação acelerada por GPU.

O material elaborado e utilizado no desenvolvimento deste trabalho está disponibilizado no repositório online (LIMA, 2024). Nesse repositório encontram-se os módulos, *scripts* e *datasets* criados e utilizados para o treinamento das redes neurais direta e inversa.

3.4.1 Rede Direta

Antes do treinamento da rede, é necessário definir seus hiperparâmetros. Primeiramente, uma arquitetura de rede neural totalmente conectada é instanciada com pesos e *bias* iniciais, com base no número de camadas ocultas, número de neurônios por camada e função de ativação já decididos. Em seguida, são definidas a função custo e o otimizador, juntamente com a taxa de aprendizado. O número de épocas inicial de treinamento também é pré-definido. A figura 14 mostra uma ilustração da rede direta, que tem os parâmetros dos moduladores como entrada, e a previsão dos picos do EOC na saída.

Figura 14 – Ilustração da rede direta



Elaborado pelo autor.

Com os hiperparâmetros definidos, o treinamento é realizado iterativamente, com o número de épocas final determinado pela técnica de parada antecipada. Em cada época, todo o conjunto de dados é analisado em lotes de tamanho definido. Para cada lote de dados, os seguintes procedimentos são realizados: carregamento dos dados (entrada e saída-alvo); cálculo da saída da rede atual usando os dados de entrada; cálculo do erro entre a saída da rede e a saída-alvo; propagação dos erros de volta na rede para calcular os gradientes; e atualização dos pesos da rede usando o otimizador. No fim de cada época, o erro médio de treinamento e o de validação é calculado para monitorar o progresso de aprendizagem. O Algoritmo 1 (linhas 1-12) mostra um pseudocódigo com as etapas citadas.

O Algoritmo 2 mostra o pseudocódigo da iteração para o cálculo da perda média de validação. Tal iteração é bem similar a de treinamento, linhas 4-11 Alg. 1, exceto pela atualização da rede feito no treinamento (linhas 8-10 Alg.1) que não deve ser feita durante a validação. Visualizando as perdas de treinamento e validação é possível tirar conclusões a respeito do desempenho da rede e determinar se mais épocas serão necessárias, por exemplo, como mostrado na figura 9 da seção 2.2.

3.4.2 Rede Inversa

Tendo a rede direta já treinada, a mesma será usada para auxiliar o treinamento da rede inversa, que tem etapas bem similares ao treinamento rede direta.

Na etapa inicial, é instanciada uma arquitetura de rede neural com o número de neurônios e camadas ocultas (que não são necessariamente os mesmos da rede direta) e a

Algoritmo 1: Treinamento de Rede Neural

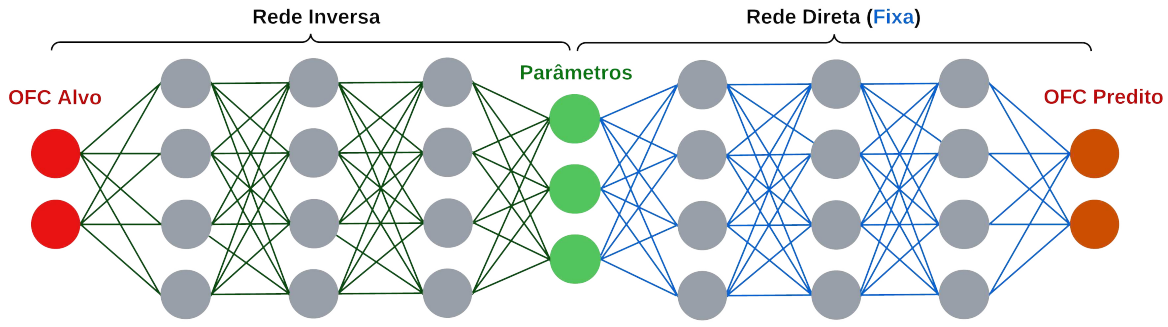
1. Instanciar rede: número de camadas e neurônios, e função de ativação (ex: ReLU)
2. Definir função de perda (ex: MSE)
3. Definir otimizador (ex: Adam) com taxa de aprendizado
4. **para** cada época **no** número total de épocas **faça**:
5. **para** cada lote de dados (entradas e saídas-alvo) **no** conjunto de treinamento **faça**:
6. Calcular saídas da rede usando dados de entrada
7. Calcular perda comparando saídas com alvos
8. Zerar gradientes acumulados no otimizador
9. Propagar erros para trás e calcular gradientes
10. Atualizar pesos da rede usando otimizador
11. **fim para**
12. Monitorar perda média de treinamento da época
13. Monitorar perda média de validação da época (Algoritmo 2)
14. **se** critério de parada for atingido **faça:**
15. Encerrar treinamento
16. **fim se**
17. **fim para**

Algoritmo 2: Validação de Rede Neural

1. **para** cada lote de dados (entradas e saída-alvos) **no** conjunto de validação **faça**:
2. Calcular saídas da rede usando dados de entrada
3. Calcular perda comparando saídas com alvos
4. **fim para**
5. Monitorar perda média de validação da época

função de ativação já decididos. Em seguida, são definidas a função custo e o otimizador, juntamente com a taxa de aprendizado. O número de épocas de treinamento também é pré-definido. Por fim, antes das iterações de treinamento, é carregada a rede direta com todos os seus parâmetros congelados, uma vez que apenas a rede inversa irá ser treinada. A figura 15 mostra a ilustração do esquema de cascata da rede inversa com a rede direta congelada, segundo o método do *Inverse Design*, para o treinamento da rede inversa.

Figura 15 – Ilustração do esquema de cascata da rede inversa com rede direta para o *Inverse Design* de OFCs



Elaborado pelo autor.

A iteração de treinamento da rede inversa também é bem similar à da rede direta, exceto pelo detalhe de que informação que entra na rede direta que será a saída da rede inversa, ou seja, as duas redes são conectadas em série para o treinamento da rede inversa. O Algoritmo 3 mostra o pseudocódigo usado para o treinamento da rede inversa.

O Algoritmo 4 mostra o pseudocódigo da iteração para o cálculo da perda média de validação feito após cada época de treinamento da rede inversa. Tal iteração é bem similar a de treinamento, linhas 5-13 Alg. 3, exceto pela atualização da rede feito no treinamento (linhas 10–12 Alg.3) que não deve ser feita durante a validação.

Algoritmo 3: Treinamento da Rede Neural Inversa

1. Instanciar rede inversa: número de camadas e neurônios, e função de ativação (ex: ReLU)
2. Definir função de perda (ex: MSE)
3. Definir otimizador (ex: Adam) com taxa de aprendizado
4. Carregar rede direta com parâmetros congelados
5. **para** cada época **no** número total de épocas **faça**:
6. **para** cada lote de dados (entradas e saídas-alvo) **no** conjunto de treinamento **faça**:
7. Calcular saídas da rede inversa usando as saídas-alvo
8. Calcular saídas da rede direta usando as saídas da rede inversa
9. Calcular perda comparando saídas da rede direta com as saídas-alvo
10. Zerar gradientes acumulados no otimizador
11. Propagar erros para trás e calcular gradientes
12. Atualizar pesos da rede usando otimizador
13. **fim para**
14. Monitorar perda média de treinamento da época
15. Monitorar perda média de validação da época (Algoritmo 4)
16. **se** critério de parada for atingido **faça:**
17. Encerrar treinamento
18. **fim se**
19. **fim para**

Algoritmo 4: Validação da Rede Neural Inversa

1. **para** cada lote de dados (entradas e saídas-alvo) **no** conjunto de validação **faça**:
2. Calcular saídas da rede inversa usando as saídas-alvo
3. Calcular saídas da rede direta usando as saídas da rede inversa
4. Calcular perda comparando saídas da rede direta com as saídas-alvo
5. **fim para**
6. Monitorar perda média de validação da época

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Para ambas configurações de EOM baseadas em cascata de moduladores mostradas na seção 3.1, e com base na metodologia apresentada no capítulo anterior, foi feito o treinamento das redes direta e inversa.

Foram utilizadas a função de ativação ReLU, a função de perda MSE e o otimizador Adam, seguindo as motivações da seção 3.3. Os demais hiperparâmetros, taxa de aprendizado, número de neurônios e camadas ocultas, e tamanho do lote de dados, foram definidos com base na busca utilizando Otimização Bayesiana.

4.1 Cascata PM-MZM-MZM

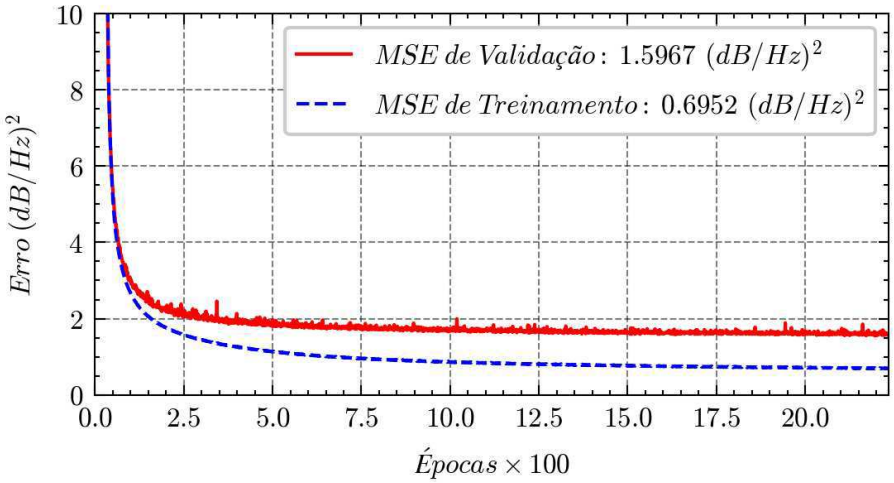
A busca por hiperparâmetros utilizando Otimização Bayesiana sintonizou a taxa de aprendizado para $\alpha = 8{,}5 \times 10^{-5}$ em ambos os casos de treinamento de rede, direta e inversa. O valor do tamanho do lote de dados dos melhores resultados variou entre 70 e 100 amostras.

4.1.1 Rede Direta

A respeito da estrutura da rede direta, testes pontuais indicaram que, para uma mesma quantidade de parâmetros de rede, o aumento do número de camadas ocultas resultou em melhor desempenho no treinamento. A busca utilizando Otimização Bayesiana também convergiu para valores elevados de número de camadas.

A busca por hiperparâmetros foi realizada com os seguintes intervalos: [3, 9] camadas ocultas; [128, 512] neurônios por camada; [64, 512] de tamanho do lote de dados; e [10^{-6}, 10^{-3}] de taxa de aprendizado.

Figura 16 – Curvas de perda de treinamento e de validação durante o aprendizado da rede direta da configuração PM-MZM-MZM.



Elaborado pelo autor.

O melhor resultado da otimização de hiperparâmetros resultou em uma rede com 8 camadas ocultas, 176 neurônios por camada, 80 amostras de tamanho de lote de dados e uma taxa de aprendizado de $8{,}5 \times 10^{-5}$ (conforme mencionado anteriormente).

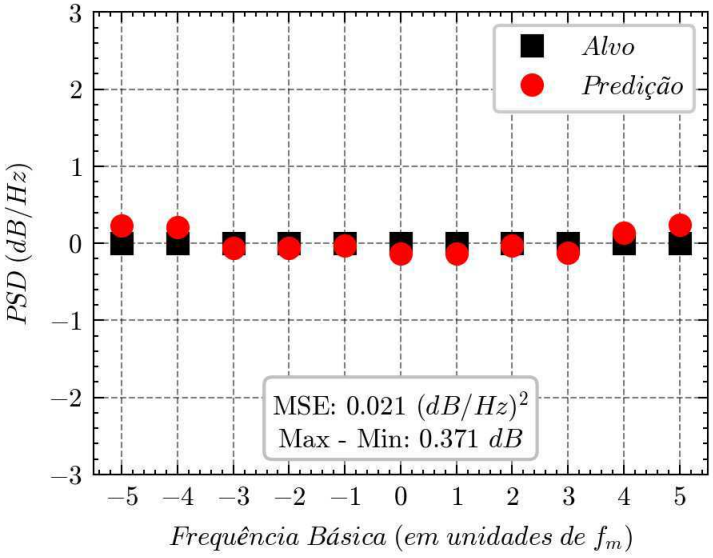
A Figura 16 mostra o acompanhamento das perdas de treinamento e validação durante o treinamento da rede direta. Esse processo levou aproximadamente 2250 épocas para ser concluído, utilizando a técnica de parada antecipada para determinar o seu término. O treinamento teve uma duração de cerca de 13 horas e, ao final, a rede apresentou um erro (perda) de treinamento de $0{,}6952\ (dB/Hz)^2$ e um erro de validação de $1{,}5967\ (dB/Hz)^2$.

Após a etapa de treinamento, realiza-se a etapa de teste, na qual é possível avaliar o desempenho final do modelo em poucos instantes. O processo de teste é praticamente o mesmo da validação, mudando apenas o *dataset* e a sua execução, a qual é completamente separada da etapa de treinamento. Assim, espera-se observar um erro similar ao último erro de validação verificado e, de fato, o erro encontrado nessa etapa foi de $1{,}5975\ (dB/Hz)^2$. O resumo do treinamento é mostrado na Tabela 1.

Tabela 1 – Resumo do treinamento da rede direta da configuração PM-MZM-MZM.

Número de Épocas	2250
Tempo de Treinamento	13h
MSE de Treinamento	0,6952 $(dB/Hz)^2$
MSE de Validação	1,5967 $(dB/Hz)^2$
MSE de Teste	1,5975 $(dB/Hz)^2$

Figura 17 – Desempenho da rede direta na predição de um EOC de espectro plano da configuração PM-MZM-MZM.



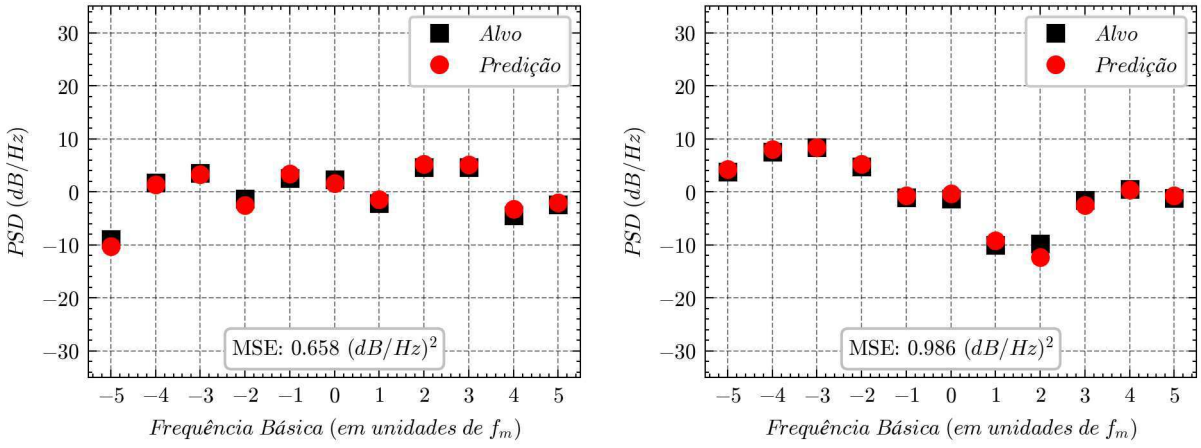
Elaborado pelo autor.

Foi separada uma amostra de EOC plano para verificar o desempenho da rede com esse tipo de formato de espectro. A Figura 17 mostra que o EOC predito tem um erro de

apenas $0{,}021\ (dB/Hz)^2$ em comparação ao EOC real. Além disso, a diferença entre o máximo e o mínimo, também chamada de planicidade, é de $0{,}371\ dB$, que é menor que $1\ dB$, indicando que o EOC predito também é considerado como sendo de espectro plano.

A Figura 18 mostra o desempenho da rede direta na predição de algumas amostras aleatórias de EOC do *dataset* de teste que têm uma boa precisão, como indica o erro médio da etapa de teste.

Figura 18 – Desempenho da rede direta na predição de algumas amostras aleatórias do *dataset* de teste.



(a) (b)

Elaborado pelo autor.

4.1.2 Rede Inversa

A princípio, por intuição, experimentou-se uma estrutura de rede igual à utilizada no treinamento da rede inversa. No entanto, os resultados obtidos não foram satisfatórios, levando à decisão de deixar a sintonização dos hiperparâmetros a cargo da busca com Otimização Bayesiana.

A busca utilizando otimização seguiu os mesmos limites de busca aplicados no caso da rede direta. O melhor resultado encontrado foi uma rede com 6 camadas ocultas, 345 neurônios por camada e aproximadamente 80 amostras para o tamanho do lote de dados. Como mencionado anteriormente, a taxa de aprendizado nesse caso também convergiu para $8{,}5 \times 10^{-5}$.

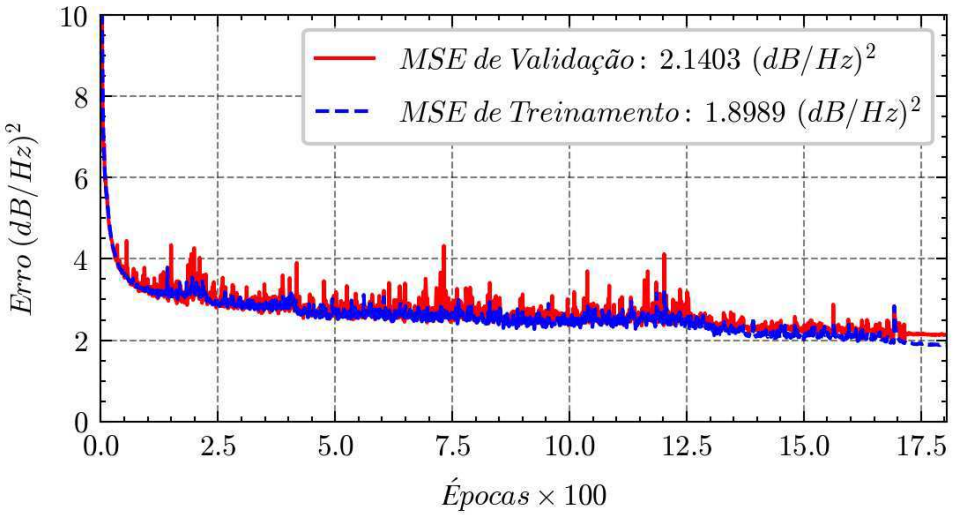
Utilizando a Equação 18, é possível calcular o número de parâmetros de cada rede, direta e inversa. A rede direta possui cerca de 220.000 parâmetros, enquanto a rede inversa possui cerca de 600.000. Devido ao maior número de parâmetros, é esperado que a rede inversa seja mais lenta tanto durante o treinamento quanto na sua aplicação após o treinamento.

Outras tentativas de busca por hiperparâmetros foram realizadas utilizando a Equação 19 para limitar a quantidade de parâmetros da rede inversa a valores menores. Foi utilizado o

limite de [200.000, 500.000] parâmetros de rede na busca utilizando otimização bayesiana, no entanto, não foi possível encontrar outra rede que apresentasse um desempenho superior.

A Figura 19 mostra o acompanhamento das perdas de treinamento e validação durante o treinamento da rede inversa, destacando uma instabilidade significativa no início do treinamento. Para mitigar essa instabilidade, é comum utilizar técnicas de decaimento da taxa de aprendizado ou aumento progressivo do tamanho do lote de dados. Como o otimizador Adam já ajusta a taxa de aprendizado individualmente para cada parâmetro da rede, optou-se por explorar a técnica de aumento progressivo do *batch size*.

Figura 19 – Curvas de perda de treinamento e de validação durante o aprendizado da rede inversa da configuração PM-MZM-MZM.



Elaborado pelo autor.

O treinamento começou com um *batch size* de 80 amostras, conforme indicado pela sintonização de hiperparâmetros. Após algumas épocas, o *batch size* foi dobrado repetidamente até que não houvesse mais melhorias consideráveis. Iniciar com um *batch size* pequeno permite uma melhor exploração do espaço de soluções, evitando que o treinamento fique preso em mínimos locais. À medida que o modelo se aproxima da convergência, utilizar *batch sizes* maiores permite estimativas mais precisas do gradiente, resultando em uma convergência mais estável e em direção a mínimos mais profundos.

Esse processo de treinamento levou aproximadamente 1800 épocas para ser concluído, com uma duração total de cerca de 21 horas. Graças à técnica de aumento progressivo do *batch size*, o treinamento foi mais rápido do que se esperaria para uma rede com quase o triplo da quantidade de parâmetros da rede direta.

Ao final do treinamento, a rede apresentou um erro (perda) de treinamento de $1{,}9625\ (dB/Hz)^2$ e um erro de validação de $2{,}1854\ (dB/Hz)^2$. Após o treinamento, foi realizada a etapa de teste para avaliar o desempenho final do modelo. Como esperado, o erro encontrado nessa etapa foi de $2{,}1808\ (dB/Hz)^2$, similar ao último erro de validação verificado, confirmando a capacidade de generalização da rede. O resumo do treinamento da rede inversa

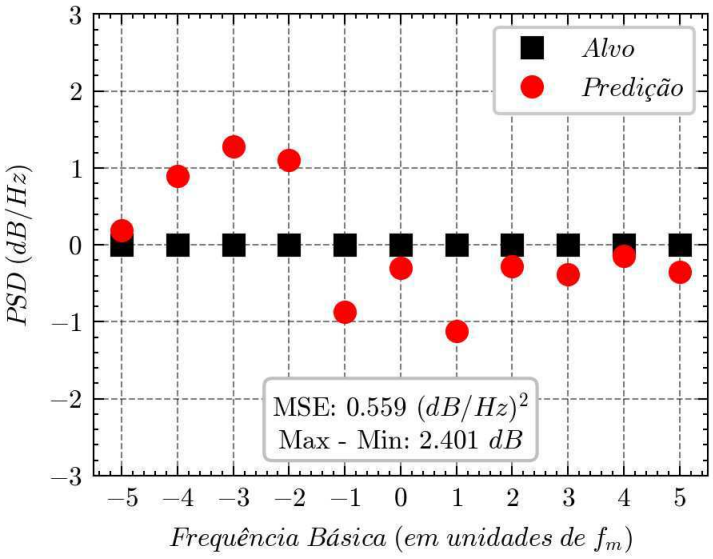
em comparação à rede direta é mostrado na Tabela 2.

A Figura 20 mostra o desempenho da rede na predição dos parâmetros de um EOC plano ideal. Para esta avaliação, foram inseridos na rede inversa 11 pontos com variância nula em relação à sua média (planicidade ideal). No entanto, por mais que o erro médio da predição seja consideravelmente pequeno, o espectro gerado com os parâmetros preditos apresentou uma planicidade de $2{,}401$ dB, excedendo o limite de categorização de pentes planos.

Tabela 2 – Resumo do treinamento das redes direta e inversa da configuração PM-MZM-MZM.

	Rede Direta	Rede Inversa
Número de Épocas	2250	1800
Tempo de Treinamento	13h	21h
MSE de Treinamento	0,6952 $(dB/Hz)^2$	1,9625 $(dB/Hz)^2$
MSE de Validação	1,5967 $(dB/Hz)^2$	2,1854 $(dB/Hz)^2$
MSE de Teste	1,5975 $(dB/Hz)^2$	2,1808 $(dB/Hz)^2$

Figura 20 – Desempenho da rede inversa na predição dos parâmetros um EOC de espectro plano ideal na configuração PM-MZM-MZM.

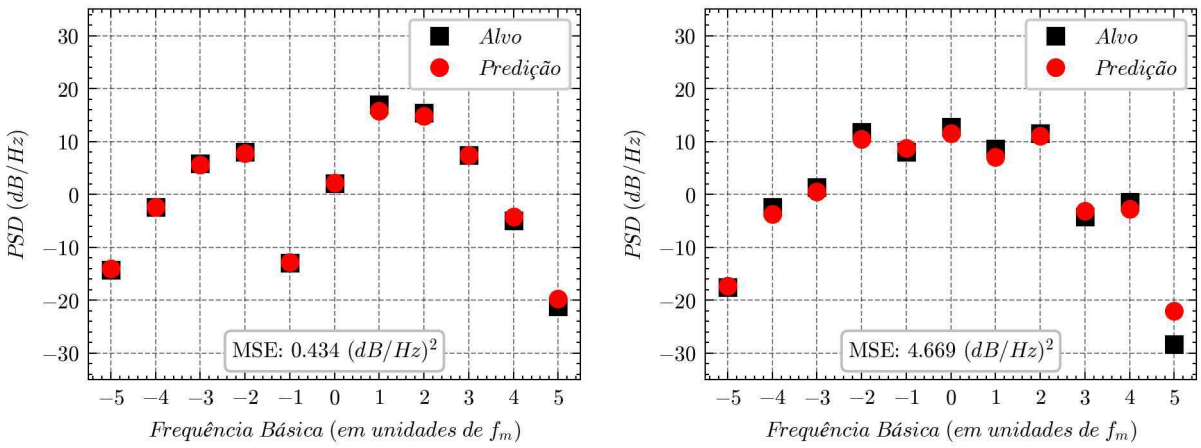


Elaborado pelo autor.

A Figura 21 mostra o desempenho da rede inversa na predição de algumas amostras aleatórias do *dataset* de teste. Em média, o erro de predição da rede inversa é consideravelmente baixo, o que indica que boa capacidade de generalização. Seu melhor desempenho se deu no controle de geração de EOCs com formato espectral mais natural, os quais são pentes que geralmente apresentam uma concavidade voltada para baixo.

A Figura 22 mostra tentativa de predição de EOCs com espectral menos natural que pode ser útil na criação de formas de onda arbitrárias e na equalização não linear de sinais, por exemplo. O desempenho da rede nesses casos não é tão alto, provavelmente devida a falta de representatividade do *dataset*. Tal questão que poderia ser superada ao manipular a criação do

Figura 21 – Desempenho da rede inversa na predição dos parâmetros de algumas amostras aleatórias do *dataset* de teste da configuração PM-MZM-MZM

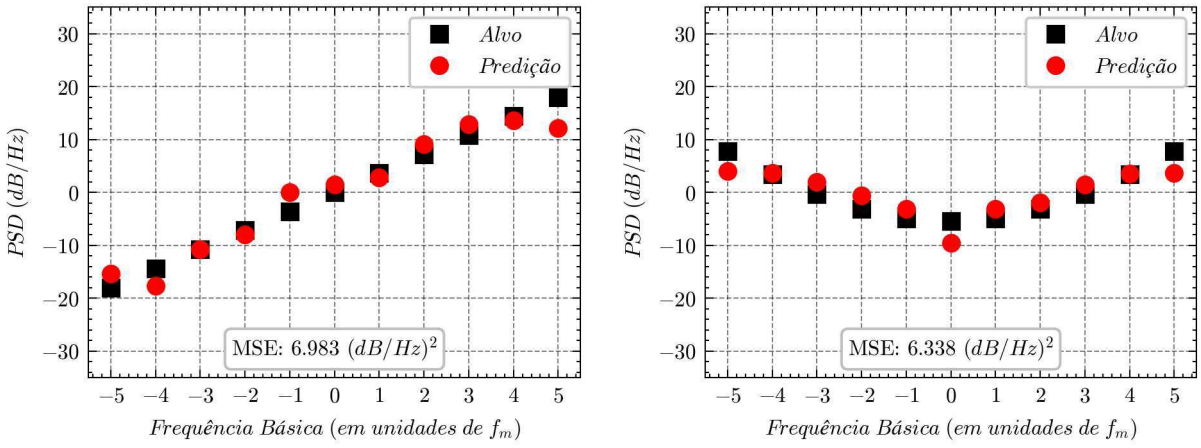


(a) (b)

Elaborado pelo autor.

conjunto de dados, incluindo amostras com esses de tipos formatos espectrais, de modo que a rede pudesse aprendê-los e melhorar sua predição.

Figura 22 – Desempenho da rede inversa na predição dos parâmetros de alguns formatos diversos de EOCs



(a) (b)

Elaborado pelo autor.

4.2 **Cascata PM-PM-MZM**

Para o treinamento das redes nesta configuração de gerador de EOCs, foram utilizados os mesmos hiperparâmetros da seção anterior, visto que eles também apresentaram um bom desempenho de treinamento neste caso.

4.2.1 Rede Direta

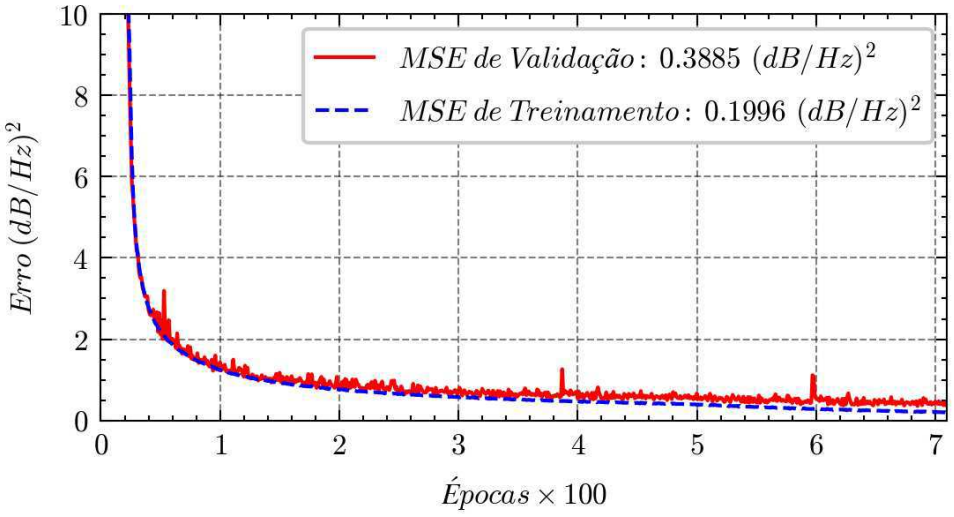
A Figura 23 mostra o acompanhamento das perdas de treinamento e validação durante o treinamento da rede direta. Esse processo levou 700 épocas para ser concluído, utilizando a técnica de parada antecipada para determinar o seu término. O treinamento teve uma duração de cerca de 5 horas e, ao final, a rede apresentou um erro de treinamento de $0{,}1996$ $(\text{dB/Hz})^2$ e um erro de validação de $0{,}3885$ $(\text{dB/Hz})^2$.

Após o treinamento, foi realizada a etapa de teste para avaliar o desempenho final do modelo. Como esperado, o erro encontrado nessa etapa foi de $0{,}3902$ $(\text{dB/Hz})^2$, similar ao último erro de validação verificado, confirmando a capacidade de generalização da rede. O resumo do treinamento é mostrado na Tabela 3.

Tabela 3 – Resumo do treinamento da rede direta da configuração PM-PM-MZM.

Número de Épocas	700
Tempo de Treinamento	5h
MSE de Treinamento	0,1996 $(dB/Hz)^2$
MSE de Validação	0,3885 $(dB/Hz)^2$
MSE de Teste	0,3902 $(dB/Hz)^2$

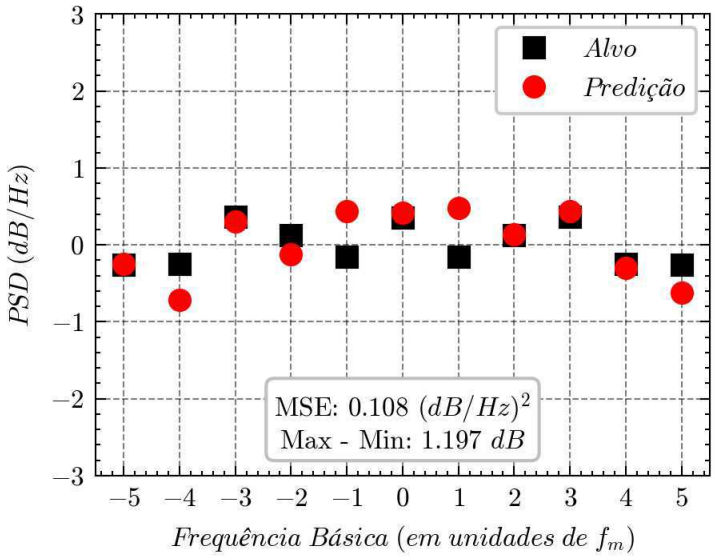
Figura 23 – Curvas de perda de treinamento e de validação durante o aprendizado da rede direta da configuração PM-PM-MZM.



Elaborado pelo autor.

Foi separada uma amostra de EOC plano para verificar o desempenho da rede com esse tipo de formato de espectro. A Figura 24 mostra que o EOC predito tem um erro de apenas $0{,}108$ $(\text{dB/Hz})^2$ em comparação ao EOC real. Contudo, embora o erro médio dessa predição seja pequeno, sua planicidade, $1{,}197$ dB, ultrapassa o limite para categorização de EOCs planos, sendo de no máximo 1 dB.

Figura 24 – Desempenho da rede direta na predição de um EOC de espectro plano da configuração PM-PM-MZM.

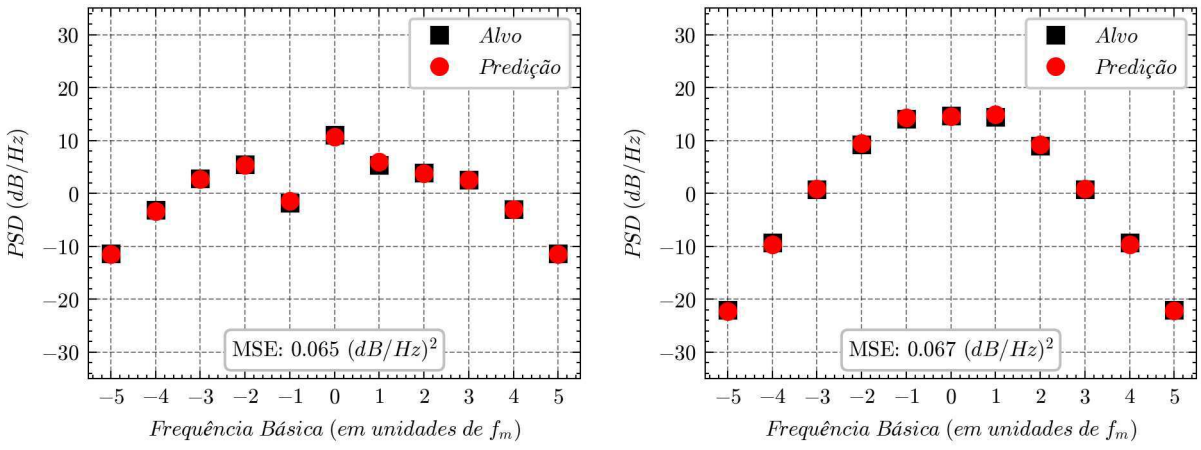


Elaborado pelo autor.

Essa limitação na predição de EOCs planos pode ser devida à considerável falta de representatividade desse tipo de EOC no *dataset*. Como mencionado na seção 3.2.2, essa configuração de gerador de EOC apresentou grande dificuldade na obtenção de pentes planos durante a construção do conjunto de dados. Investir mais tempo na geração de pentes planos para a composição do *dataset* poderia melhorar o desempenho da rede na predição desse tipo de EOC.

Apesar das limitações, a rede direta apresentou uma baixa perda média no treinamento, indicando que outros formatos de EOCs podem ter sido aprendidos com alta precisão. A Figura 25 mostra o desempenho da rede direta na predição de algumas amostras aleatórias.

Figura 25 – Desempenho da rede direta na predição de algumas amostras aleatórias do *dataset* de teste.



(a) (b)

Elaborado pelo autor.

4.2.2 Rede Inversa

A Figura 26 mostra o acompanhamento das perdas de treinamento e validação durante o treinamento da rede direta. Similar ao treinamento da rede inversa da configuração PM-MZM-MZM, nesse caso também foi observada uma instabilidade significativa no início do treinamento. E, da mesma forma, foi utilizada a técnica de aumento progressivo do tamanho do lote de dados para acelerar o treinamento e garantir a estabilidade.

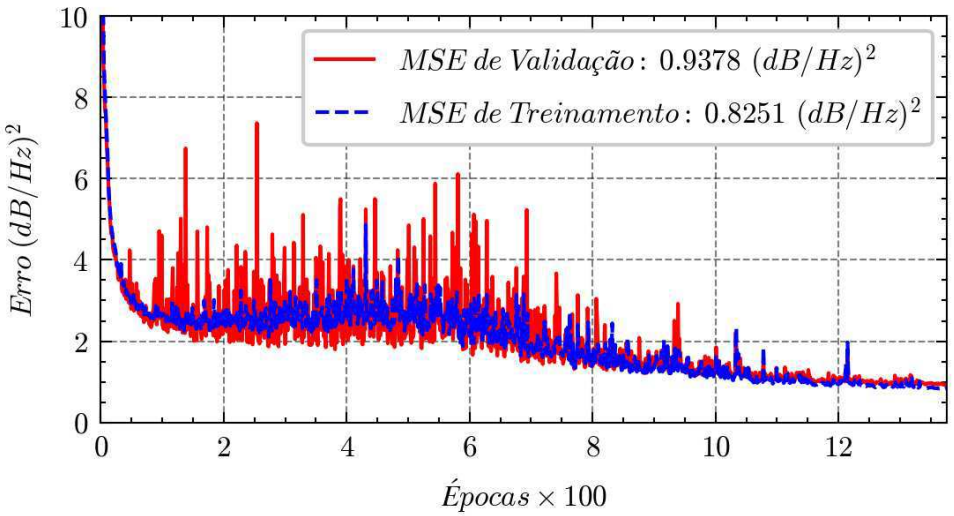
O treinamento levou cerca de 1400 épocas para ser concluído, utilizando a técnica de parada antecipada para determinar o seu término. O treinamento teve uma duração de cerca de 16 horas e, ao final, a rede apresentou um erro de treinamento de $0{,}8251$ $(\text{dB/Hz})^2$ e um erro de validação de $0{,}9378$ $(\text{dB/Hz})^2$.

Após o treinamento, foi realizada a etapa de teste para avaliar o desempenho final do modelo. Como esperado, o erro encontrado nessa etapa foi de $0{,}8952$ $(\text{dB/Hz})^2$, similar ao último erro de validação verificado, confirmando a capacidade de generalização da rede. O resumo do treinamento da rede inversa em comparação à rede direta é mostrado na Tabela 4.

Tabela 4 – Resumo do treinamento das redes direta e inversa da configuração PM-PM-MZM.

	Rede Direta	Rede Inversa
Número de Épocas	700	1400
Tempo de Treinamento	5h	16h
MSE de Treinamento	0,1996 $(dB/Hz)^2$	0,8251 $(dB/Hz)^2$
MSE de Validação	0,3885 $(dB/Hz)^2$	0,9378 $(dB/Hz)^2$
MSE de Teste	0,3902 $(dB/Hz)^2$	0,8952 $(dB/Hz)^2$

Figura 26 – Curvas de perda de treinamento e de validação durante o aprendizado da rede inversa da configuração PM-PM-MZM.

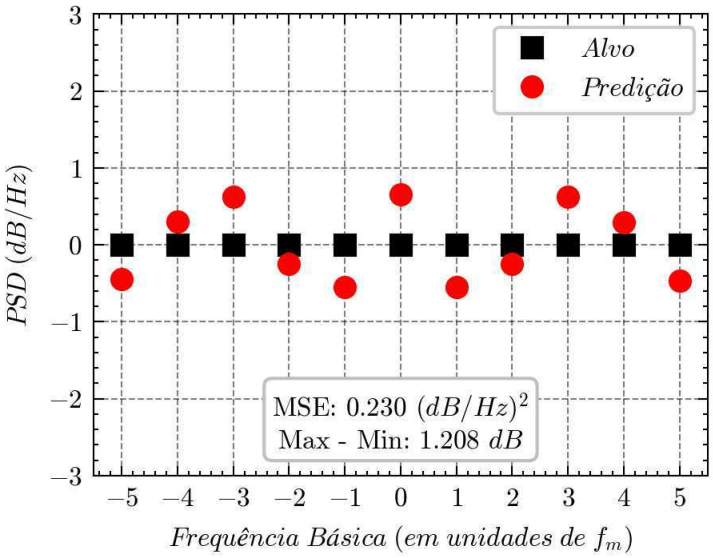


Elaborado pelo autor.

A Figura 27 mostra o desempenho da rede na predição dos parâmetros de um EOC

plano ideal. É possível perceber que a rede falhou em fornecer os parâmetros necessários para a geração de um EOC plano, uma vez que o espectro gerado com os parâmetros preditos apresentou uma planicidade de $1{,}208$ dB, excedendo o limite de categorização de pentes planos. No entanto, já era esperado que a rede inversa falhasse nessa predição, uma vez que a própria rede direta falhou. É importante lembrar que a rede direta é a responsável por ensinar a rede inversa, assim se a rede direta possuir alguma deficiência, tal problema poderá ser passado para rede inversa. Contudo, é interessante observar que o erro entre a predição da rede direta e da rede inversa é bem parecido, o que indica que, na verdade, a rede inversa aprendeu bem e a principal limitação está na rede direta.

Figura 27 – Desempenho da rede inversa na predição dos parâmetros um EOC de espectro plano ideal na configuração PM-PM-MZM.



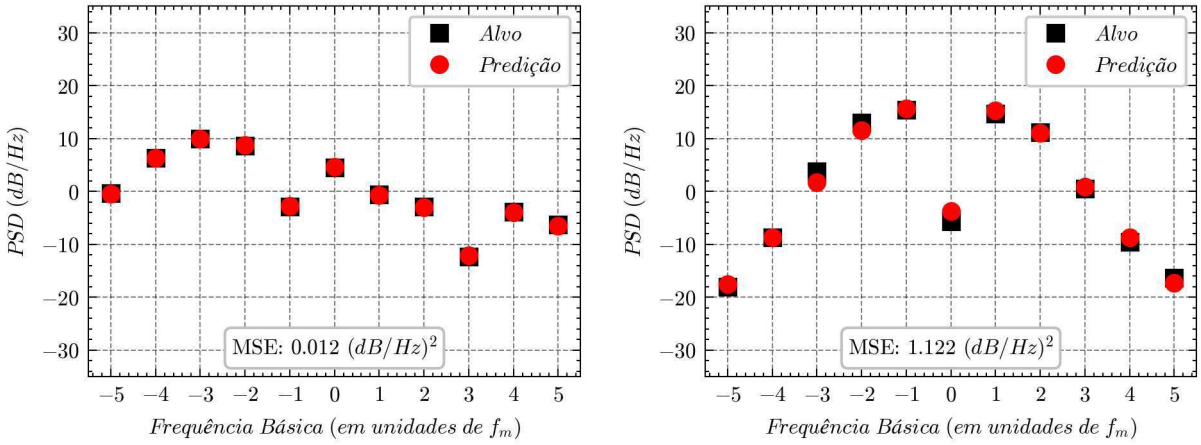
Elaborado pelo autor.

A Figura 28 mostra o desempenho da rede inversa na predição de algumas amostras aleatórias do *dataset* de teste. O fato de o erro médio de predição da rede ser bem baixo indica que ela apresenta boa generalização e um ótimo desempenho no controle de geração de EOCs com formato espectral mais natural (formatos que geralmente se assemelham a uma gaussiana).

Já no controle de geração de EOCs com formato espectral menos natural, como mostra a Figura 29, a rede apresenta resultados com erro bem maior do que a média. O primeiro caso, Figura 29a, mostra que a rede falhou na predição de um EOC com formato espectral linear, enquanto o segundo caso, Figura 29b, mostra uma predição mais aceitável para um formato de uma gaussiana invertida. O desempenho da rede nesses casos não é tão alto, provavelmente devida a falta de representatividade do *dataset*. Tal questão que poderia ser superada ao manipular a criação do conjunto de dados, incluindo amostras com esses de tipos formatos espectrais, de modo que a rede pudesse aprendê-los e melhorar sua predição.

Comparando a Figura 22a com a Figura 29a, observa-se que a rede da configuração

Figura 28 – Desempenho da rede inversa na predição dos parâmetros de algumas amostras aleatórias do *dataset* de teste da configuração PM-PM-MZM

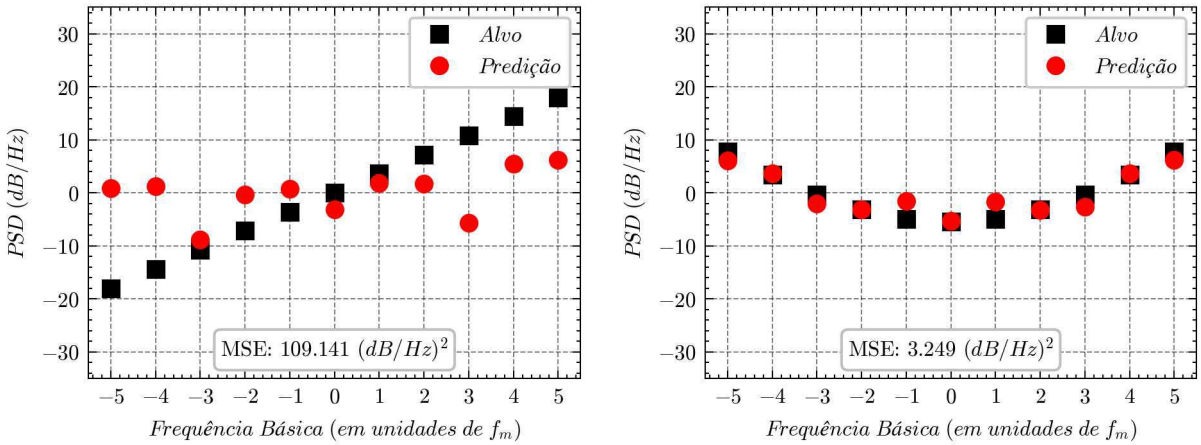


(a) (b)

Elaborado pelo autor.

1 (PM-MZM-MZM) se saiu melhor na predição desse formato de pente onde a potência das linhas são lateralizadas linearmente. Por outro lado, no segundo caso, a Figura 29b mostra que, comparada à Figura 22b, a rede da configuração PM-PM-MZM se sobressai na predição de um pente com formato quadrático. Isso mostra que a depender da aplicação, um gerador pode ser mais adequado que outro para emprego do método de *design* inverso com aprendizado profundo.

Figura 29 – Desempenho da rede inversa na predição dos parâmetros de alguns formatos diversos de EOCs



(a) (b)

Elaborado pelo autor.

5 CONCLUSÃO

Neste trabalho, aplicou-se o método de *design* inverso de OFC baseado em aprendizado profundo para o treinamento do chamado modelo inverso. Foram estudados dois casos de geradores de OFC baseados em cascatas de moduladores eletro-ópticos acionados por um único sinal de controle. Os geradores foram simulados numericamente para a criação dos conjuntos de dados de treinamento. Os resultados do trabalho mostram que, em diversos casos, os modelos inversos treinados podem obter os parâmetros do gerador correspondentes ao OFC alvo com grande precisão e em tempo hábil, comparado aos métodos de controle de parâmetros mais tradicionais, como o da variável de controle, ou mesmo métodos mais recentes, como a otimização diferencial evolutiva.

Um trabalho anterior (MA et al., 2022), que utilizou como gerador um único DDMZM, mostrou que o método de *design* inverso foi eficiente na geração de OFCs com até 9 linhas de pente de planicidade menor que 1 dB. O presente trabalho pode não ter alcançado melhores resultados na predição de pentes planos, provavelmente de devido ao aumento da complexidade do gerador, que poderia ser superado com o aumento da representatividade do conjunto de dados e o maior investimento de tempo na sintonização de hiperparâmetros.

Apesar das limitações, as etapas de teste mostraram que as redes inversas apresentam alta capacidade de generalização e isso mostra que o método de *design* inverso com aprendizado profundo tem um desempenho promissor na predição de OFCs.

5.1 Trabalhos Futuros

Para avançar na melhoria deste trabalho, algumas ideias poderiam ser seguidas para a execução de trabalhos futuros.

A experimentação do método de *design* inverso com outros geradores de mesma complexidade seria ideal para a comparação do desempenho dos modelos treinados.

No que diz respeito à metodologia de treinamento, a inclusão do parâmetro de potência do *laser* e/ou a remoção dos parâmetros de fase dos sinais de controle na criação do *dataset* são maneiras de alterar a complexidade dos dados e facilitar possivelmente o treinamento dos modelos.

Uma vez que os modelos estivessem bem treinados e com boa generalização, poderia-se estender o método para o caso de geradores mais complexos, que utilizassem mais moduladores e/ou diferentes sinais de controle.

Além disso, a utilização de dados experimentais na construção do conjunto de dados, para treinamento e teste dos modelos em um cenário real, seria de grande valia para a validação do método de *design* inverso baseado em aprendizado profundo.

Referências

DAI, B. et al. Generation of versatile waveforms from cw light using a dual-drive mach-zehnder modulator and employing chromatic dispersion. **Journal of Lightwave Technology**, v. 31, n. 1, p. 145–151, 2013. Citado na página 9.

FOUNDATION, P. S. **Python**. 2024. <https://www.python.org/>. Acesso em 24 de maio de 2024. Citado na página 23.

GOODFELLOW, I. et al. **Deep Learning**. [S.l.]: MIT Press, 2016. 104–108 p. Citado na página 11.

HALES, P. et al. Combined denoising and suppression of transient artifacts in arterial spin labeling mri using deep learning. **Journal of Magnetic Resonance Imaging**, v. 52, 06 2020. Citado na página 12.

HEATON, J. **Bayesian Hyperparameter Optimization for PyTorch**. 2023. <https://github.com/jeffheaton/app_deep_learning/blob/main/t81_558_class_08_4_bayesian_hyperparameter_opt.ipynb>. Acesso em 15 de abril de 2024. Citado na página 21.

HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. **International Journal of Uncertainty Fuzziness and Knowledge-Based Systems**, v. 7, 1998. Citado na página 19.

IDEGUCHI, T. et al. Coherent raman spectro-imaging with laser frequency combs. **Nature**, Nature Publishing Group, v. 502, n. 7471, p. 355–358, 2013. Citado na página 9.

KELLER, U. Optical frequency comb from modelocked lasers. Springer International Publishing, Cham, p. 639–702, 2021. Citado na página 8.

KHANNA, C. Number of parameters in a feed-forward neural network. **Towards Data Science**, September 2020. Disponível em: <https://towardsdatascience.com/number-of-parameters-in-a-feed-forward-neural-network-4e4e33a53655>. Citado na página 22.

KIM, J.; RICHARDSON, D. J.; SLAVíK, R. Cavity-induced phase noise suppression in a fabry–perot modulator-based optical frequency comb. **Opt. Lett.**, Optica Publishing Group, v. 42, n. 8, p. 1536–1539, Apr 2017. Disponível em: <https://opg.optica.org/ol/abstract.cfm?URI=ol-42-8-1536>. Citado na página 8.

KINGMA, D. P.; BA, J. L. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, v. 9, 2014. Citado na página 21.

KIPPENBERG, T. J. et al. Microresonator-based optical frequency combs. **Science**, v. 332, n. 6029, p. 555–559, 2011. Citado na página 8.

LIMA, F. F. de. **TCC_OFB**. 2024. <https://github.com/fernfl/TCC_OFB>. Acesso em 24 de maio de 2024. Citado 3 vezes nas páginas 15, 17 e 23.

LUNDBERG, L. et al. Frequency comb-based wdm transmission systems enabling joint signal processing. **Applied Sciences**, v. 8, n. 5, p. 8, 2018. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/8/5/718>. Citado na página 9.

MA, Y. et al. Inverse design of programmable optical frequency comb using deep learning. **Journal of Optics**, v. 24, p. 085702, 2022. Citado 3 vezes nas páginas 10, 12 e 38.

MILLAR, D. S. et al. Design of a 1 tb/s superchannel coherent receiver. **Journal of Lightwave Technology**, v. 34, n. 6, p. 1453–1463, 2016. Citado na página 9.

PARRIAUX, A.; HAMMANI, K.; MILLOT, G. Electro-optic frequency combs. **Advances in Optics and Photonics**, v. 12, n. 1, p. 223–266, 2020. Citado na página 4.

PARRIAUX, A.; HAMMANI, K.; MILLOT, G. Electro-optic frequency combs for spectroscopic applications. p. ITu2B.3, 01 2021. Citado na página 3.

PASZKE, A. et al. **PyTorch: An Imperative Style, High-Performance Deep Learning Library**. 2019. <https://pytorch.org/>. Acesso em 24 de maio de 2024. Citado na página 23.

PENDIUK, G. F. I.; NEVES, P. de T.; POHL, A. de A. P. Flatness optimization of optical frequency combs using an adapted differential evolution algorithm. In: **Frontiers in Optics / Laser Science**. Optica Publishing Group, 2018. p. JW3A.82. Disponível em: <https://opg.optica.org/abstract.cfm?URI=FiO-2018-JW3A.82>. Citado na página 10.

PRECHELT, L. Early stopping - but when? 03 2000. Citado na página 14.

SNOEK, J. et al. Practical bayesian optimization of machine learning algorithms. **Advances in neural information processing systems**, v. 25, 2012. Citado na página 14.

TORRES-COMPANY, V.; WEINER, A. M. Optical frequency comb technology for ultra-broadband radio-frequency photonics. **Laser & Photonics Reviews**, v. 8, n. 3, p. 368–393, 2014. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/lpor.201300126>. Citado na página 9.

UDEM, T. et al. Accurate measurement of large optical frequency differences with a mode-locked laser. **Optics Letters**, Optical Society of America, v. 24, n. 13, p. 881–883, 1999. Citado na página 9.

ZHUANG, R. et al. Electro-optic frequency combs: Theory, characteristics, and applications. **Laser & Photonics Reviews**, p. 1–27, 2023. Citado na página 3.