Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

Trabalho de Conclusão de Curso

Uma Investigação Sobre o Uso da Amostragem Compressiva Como Técnica de Pré-Processamento de uma Rede Neural Artificial

Micael Espínola Fonseca Tomaz

Campina Grande - PB
Outubro de 2024

Micael Espínola Fonseca Tomaz

Uma Investigação Sobre o Uso da Amostragem Compressiva Como Técnica de Pré-Processamento de uma Rede Neural Artificial

Trabalho de Conclusão de Curso submetido à Coordenaria de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do Grau de Engenheira Eletricista.

Universidade Federal de Campina Grande - UFCG

Centro de Engenharia Elétrica e Informática - CEEI

Departamento de Engenharia Elétrica - DEE

Coordenação de Graduação em Engenharia Elétrica - CGEE

Edmar Candeira Gurjão, Dr.

(Orientador)

Campina Grande - PB

Outubro de 2024

Micael Espínola Fonseca Tomaz

Uma Investigação Sobre o Uso da Amostragem Compressiva Como Técnica de Pré-Processamento de uma Rede Neural Artificial

Trabalho de Conclusão de Curso submetido à Coordenaria de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do Grau de Engenheira Eletricista.

Aprovado em ____ /____ /____

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Edmar Candeira Gurjão
Universidade Federal de Campina Grande
Orientador

Campina Grande - PB
Outubro de 2024

"Dedico este trabalho às três gerações de mães com as quais fui agraciado por Deus: minha Bisa Maria, minha Vó Lu e minha Mãe. As senhoras são responsáveis por todo o amor que existe dentro de mim. Seus exemplos e conselhos estarão, para sempre, gravados como uma assinatura no meu jeito de ser."

Agradecimentos

Agradeço à minha família por todos os ensinamentos passados, por todo o amor e acolhimento que me foram dados. Em especial, à minha vó Lu e à minha mãe Nadja; seus sacrifícios serviram de alicerce para os meus estudos e me fizeram valorizar cada vez mais o que tenho e o que posso conquistar. Agradeço ao meu pai Sérgio, que, apesar da distância, nunca deixou de demonstrar a importância que tenho em sua vida. Agradeço à tia Niedja, à tia Natália, à minha irmã Kadja e aos meus primos queridos Cora, Isaac e Melina, que tornam minhas tardes de domingo mais felizes. Agradeço à minha bisa Maria por todas as noites de oração por mim, assim como a todos que também oram por mim. Todos vocês fazem parte de quem me tornei.

Agradeço ao mestre Isaías por ser meu primeiro incentivo acadêmico e a Valmir pelo exemplo diário de trabalho e perseverança. Vocês são meus exemplos de como um homem deve ser e agir.

Agradeço à minha namorada, Raissa Bucar, por todas as conversas, abraços e incentivos bondosos. Obrigado por ser você em todos os sentidos possíveis; sua presença me acalmou durante os momentos de nervosismo e insegurança, assim como serve de motivação na busca dos meus objetivos.

Agradeço a todos os bons amigos que fiz na UFCG: Andrey Lucyan, Lucas Lobo, Leonardo Pessoa, Yuri Siqueira, Tayrone Oliveira, Victor Hugo, Caio Rodrigues, Bruno Nascimento, Moisés de Araújo, Airon Cirilo, Igor Costa, Jan Paulo, Débora Costa, Dhara Pamplona, Sabrina Araújo, Marcos Paulo e Dimas Nathan. Os momentos compartilhados com vocês são inestimáveis, desde os grupos de estudo desesperados até as conversas descontraídas em humanas, que funcionavam quase como uma pausa para descansar dos perrengues da graduação.

Agradeço aos meus amigos de longa data: Higor Emmanuel, Eliab Pina, Felipe Nóbrega, José Crispim e David Wilson, pelas discussões profundas, risos autênticos e reuniões surpresas que temos desde a época da escola.

Agradeço também a todos que contribuíram para o meu desenvolvimento acadêmico, especialmente ao meu professor e orientador Edmar Candeia Gurjão. Guardo com carinho todos os conselhos e puxões de orelha que recebi, que certamente contribuíram para o meu crescimento pessoal na UFCG.

"Pois, se amardes os que vos amam, que galardão tereis? Não fazem os publicanos também o mesmo?",
(Mateus 5:46)

Resumo

As Redes Neurais Artificiais (RNAs) são um tema atual e recorrente em qualquer âmbito tecnológico e suas diversas aplicações confirmam tal fato. No treinamento de uma RNA é necessário usar um grande conjunto de dados para representar as possibilidades de entrada e saída do problema que está sendo modelado, o que leva a uma grande demanda de processamento computacional. Este trabalho apresenta uma investigação sobre a aplicação da Amostragem Compressiva como técnica de pré-processamento em uma RNA. O objetivo principal é avaliar o desempenho dessa técnica em diferentes conjuntos de dados, considerando sua eficiência em reduzir a dimensionalidade dos dados de entrada e, simultaneamente, manter a precisão dos resultados. A pesquisa aborda conceitos fundamentais sobre representação digital de imagens, esparsidade e a teoria da Amostragem Compressiva, além de detalhar a metodologia implementada, incluindo a arquitetura das redes neurais. Os resultados obtidos demonstram que, apesar de uma possível redução na precisão do modelo, a Amostragem Compressiva pode proporcionar ganhos significativos em termos de eficiência computacional. Este estudo contribui para o entendimento das potencialidades da Amostragem Compressiva em aplicações de aprendizado de máquina, destacando direções futuras para pesquisas nessa área.

Palavras-chave: Amostragem Compressiva, Redes Neurais Artificiais, Redução Dimensional, Aprendizado de Máquina.

Abstract

Artificial Neural Networks (ANNs) are a current and recurring theme in any technological field, and their various applications confirm this fact. Training an ANN requires using a large dataset to represent the input and output possibilities of the problem being modeled, which leads to a high demand for computational processing. This work presents an investigation into the application of Compressed Sampling as a preprocessing technique in an ANN. The main objective is to evaluate the performance of this technique across different datasets, considering its efficiency in reducing the dimensionality of input data while simultaneously maintaining the accuracy of the results. The research addresses fundamental concepts regarding digital image representation, sparsity, and the theory of Compressed Sampling, as well as detailing the implemented methodology, including the architecture of the neural networks. The obtained results demonstrate that, despite a possible reduction in model accuracy, Compressed Sampling can provide significant gains in computational efficiency. This study contributes to understanding the potential of Compressed Sampling in machine learning applications, highlighting future research directions in this area.

Keywords: Compressive Sampling, Artificial Neural Networks, Dimensionality Reduction, Machine Learning.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

RNA	Redes Neurais Artificiais
DNN	*Dense Neural Networks*
CNN	*Convolutional Neural Networks*
CS	*Compressed Sampling*
MNIST	*Modified National Institute of Standards and Technology*

Sumário

1 Introdução

O avanço tecnológico nas últimas décadas popularizaram o tema do aprendizado de máquina das mais diversas formas, e isso se reflete em suas aplicações. Por exemplo, em (FENG et al., 2021) vemos o uso de inteligência artificial no teste de direção de carros autônomos. Já em (SARACCO, 2017), sua utilização na medicina permite obter diagnósticos automáticos, rápidos e precisos. Contudo, esse assunto envolve inúmeras técnicas, desde algoritmos probabilísticos, como o Naïve Bayes (MUKHERJEE; SHARMA, 2012), até o uso de Redes Neurais Artificiais (RNAs), que serão o foco do nosso estudo.

As Redes Neurais Profundas (do inglês *Dense Neural Networks* ou DNNs) são uma categoria de RNAs que possuem uma maior complexidade em relação às demais. Elas são formadas por um modelo neural matemático que contém um padrão de treinamento. As informações entram pela primeira camada, são ponderadas matematicamente nas camadas ocultas e, por fim, revelam os resultados na camada de saída. Geralmente em arquiteturas supervisionadas, esse processo pode precisar de um grande número de ocorrências de entradas e saídas conhecidas para que a rede neural fique devidamente treinada. Dependendo da complexidade do problema (dados de um hospital, pesquisas, imagens, etc.), há necessidade de um grande processamento computacional. Uma forma de contornar esse problema é tratar os dados de entrada. Nas Redes Neurais Convolucionais (do inglês *Convolutional Neural Networks* ou CNNs), por exemplo, percebemos não só o uso de convolução para destacar ou atenuar particularidades desejadas, mas também processos para a redução dimensional dessas características (BHATT et al., 2021).

A compressão de dados tem sido bastante aplicada quando percebemos suas várias utilizações na representação de informações de interesse, um exemplo notável dessa técnica é o formato JPEG que pode representar uma imagem da ordem de megabytes em dados com tamanho de ordem de kilobytes (MEDEIROS et al., 2010). Esse tipo de processo não deve ser feito de qualquer maneira, uma vez que pode trazer perda de qualidade perceptual, logo é sempre necessária uma ponderação das informações relevantes que se deseja representar com a ordem da redução dos dados e sua aplicabilidade.

Paralelamente a tudo isso, um vasto campo de estudos chamado Amostragem Compressiva (do inglês *Compressed Sampling* ou CS) tem se tornado mais presente nas pesquisas acadêmicas. O conceito de CS é unir etapas de amostragem e compressão em uma única etapa, permitindo uma amostragem a uma taxa menor que a taxa de Nyquist, sem perda de informação (CANDÈS; WAKIN, 2008). A Amostragem Compressiva se baseou em ideias matemáticas pouco exploradas na época de sua concepção e, consequentemente, proporcionou novos horizontes de pesquisa. Considerando isso, o foco dessa pesquisa é

justamente a aplicação das ideias de amostragem e compressão no processamento de dados de uma Rede Neural Densa.

1.1 Objetivo

O principal objetivo deste trabalho é apresentar uma investigação sobre o uso da Amostragem Compressiva como técnica de redução dimensional no pré-processamento dos dados de uma Rede Neural Densa, avaliando seu desempenho em diferentes bancos de dados, de acordo com suas respectivas precisões. Espera-se que essa abordagem traga ganhos computacionais, embora com uma possível redução na precisão do modelo neural.

1.2 Organização do Trabalho

O presente trabalho está organizado em seis capítulos. No Capítulo 1, é feita uma introdução ao trabalho e apresentado seu objetivo. No Capítulo 2, é realizada uma revisão teórica de conceitos matemáticos necessários para um bom entendimento da pesquisa. No Capítulo 3, abordamos a teoria da Amostragem Compressiva, suas restrições e condições. No Capítulo 4, é dada uma visão geral sobre o funcionamento de uma RNA, assim como detalhada toda a metodologia adotada na pesquisa, suas motivações e características. No Capítulo 5, são apresentados, comparados e discutidos os resultados obtidos no uso da técnica de amostragem compressiva como pré-processamento dos dados de uma RNA. Por fim, no Capítulo 6, são discutidas as conclusões do trabalho realizado e propostas direções futuras para complementar e aprimorar as atividades desenvolvidas.

2 Conceitos e Definições

2.1 Introdução

O objetivo deste capítulo é revisar conceitos e definições importantes, não só para o bom entendimento de Amostragem Compressiva, mas também para um melhor direcionamento de sua aplicação elaborada nesta pesquisa. São abordados os seguintes temas:

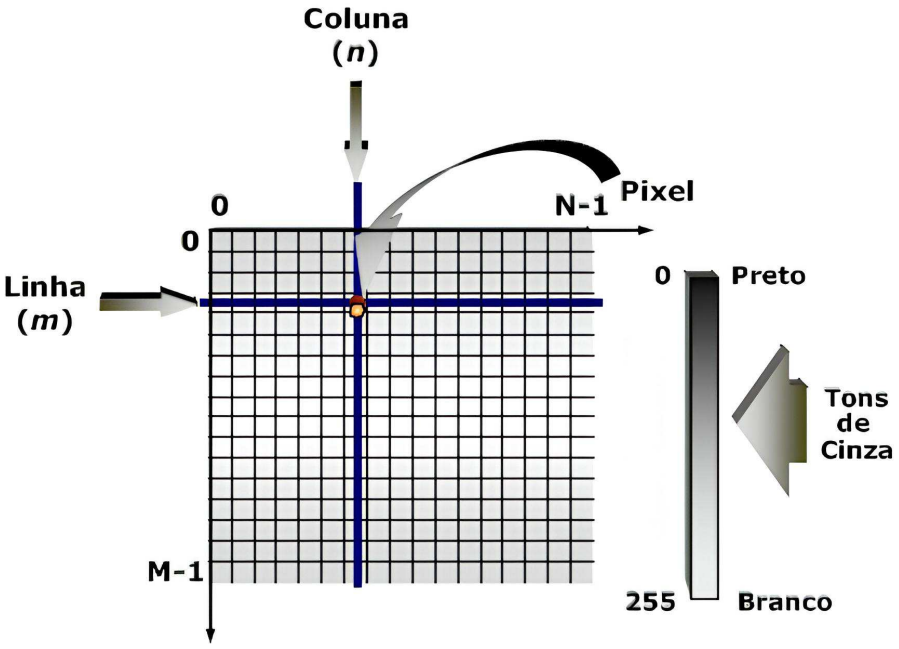
- Representação Digital de uma Imagem
- Base Vetorial
- Representação de um Vetor em uma Base
- Normas ℓ_0 e ℓ_1
- Ortogonalidade e Normalidade de uma Matriz
- Esparsidade de um Vetor
- Incoerência entre Bases
- Amostragem Discreta

2.2 Representação Digital de uma Imagem

As imagens podem ser capturadas e representadas de várias formas, como em pinturas, fotografias, filmagens e outros meios artísticos. Contudo, digitalmente, as imagens são representadas por uma matriz de pixels. Dessa maneira, o número de pixels se torna uma das características determinantes na resolução da imagem. A título de exemplo, quando optamos pela resolução padrão dos monitores atuais do mercado "*full HD*", na verdade estamos escolhendo uma opção que possui 1920 colunas por 1080 linhas de pixels em sua propiedade.

Além disso, cada pixel pode ser traduzido em diferentes números de bits, afetando diretamente a percepção da imagem. Com apenas 1 bit, podemos variar entre os estados preto e branco, o que limita consideravelmente o nível de detalhes. Com 8 bits (1 Byte), temos $2^8 = 256$ intensidades em cada pixel, permitindo assim imagens em escala de cinza. Podemos perceber a representação de uma imagem digital no esquema da Figura 1.

Figura 1 – Esquema de representação de uma imagem digital



Fonte: (ESTUDO. . . ,)

2.3 Base Vetorial

Um conjunto de vetores $\Phi = \{\phi_1, \phi_2 ..., \phi_N\}$ é uma base do espaço vetorial V quando todos os N vetores de Φ são linearmente independentes e Φ gera V (WINTERLE; STEINBRUCH, 1987). Em outras palavras, todo vetor v $\in$ V pode ser obtido por uma combinação linear dos vetores de Φ.

Exemplo 1 Considerando os vetores ϕ_i:

$$\phi_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$$
$$\phi_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T$$
$$\phi_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$$
$$\phi_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$$

O conjunto Φ é uma base que gera o espaço vetorial $\mathbb{R}^4$. A base Φ também pode ser representada na forma matricial.

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.4 Normas ℓ_0 e ℓ_1

Norma é uma medida que pode ser aplicada a um vetor $\mathbf{v}$. A norma ℓ_0 quantifica o número de elementos não nulos de $\mathbf{v}$, enquanto a norma ℓ_1 corresponde ao somatório dos valores absolutos de seus elementos.

Dada a função $I[k] = \begin{cases} 0, & \text{se } v[k] = 0 \\ 1, & \text{se } v[k] \neq 0 \end{cases}$

define-se a norma ℓ_0 como $\|\mathbf{v}\|_0 = \sum_{i=0}^{N-1} I[i]$

define-se a norma ℓ_1 como $\|\mathbf{v}\|_1 = \sum_{i=0}^{N-1} |v[i]|$

generalizando, define-se a norma ℓ_p como $\|\mathbf{x}\|_p = \left(\sum_{i=0}^{N-1} |x[i]|^p\right)^{\frac{1}{p}}$, sendo $p \geq 1$

Exemplo 2 Dado o vetor $\mathbf{y} = \begin{bmatrix} 0 & 5 \end{bmatrix}$, vamos calcular as suas normas ℓ_0 e ℓ_1:

$\|\mathbf{x}\|_0 = 0 + 1 = 1$

$\|\mathbf{x}\|_1 = 0 + 5 = 5$

2.5 Ortogonalidade e Normalidade de uma Matriz

Dada matriz $\Phi = [\phi_1,\ \phi_2\ ...,\ \phi_N]$ será:

- Normal se $\|\phi_\mathbf{i}\|_2 = 1\ \forall\ i \in [1,\ 2,\ ...,\ N]$
- Ortogonal se $\phi_i \phi_j^T = 0\ \forall\ i,j \in [1,\ 2,\ ...,\ N]$, sendo $i \neq j$
- Ortonormal se for normal e ortogonal

2.6 Esparcidade de um Vetor

Definição 1 A esparcidade de um vetor $\mathbf{v}$ é dada por $K(\mathbf{v}) = \|\mathbf{v}\|_0$.

Um vetor é dito esparso quando uma grande fração de seus elementos for 0. Contudo, como um vetor pode ser representado em várias bases diferentes, sua esparsidade pode mudar de acordo com a base utilizada (GOLUB; LOAN, 2013).

Exemplo 3 O vetor $\mathbf{y} = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \end{bmatrix}^T$ pode ser representado por y_a na base de *Fourier* $\Psi = \{\psi_1,\ \psi_2,\ ...\ ,\ \psi_8\}$, em que $\psi[k] = \frac{1}{\sqrt{N}} e^{2\pi jk/N}$. Assim, na forma matricial, temos que $y_a = \Psi\mathbf{y}$.

$$\begin{bmatrix} y_a[0] \\ y_a[1] \\ y_a[2] \\ \vdots \\ y_a[7] \end{bmatrix} = \begin{bmatrix} \psi_1[0] & \psi_2[0] & \psi_3[0] & \cdots & \psi_8[0] \\ \psi_1[1] & \psi_2[1] & \psi_3[1] & \cdots & \psi_8[1] \\ \psi_1[2] & \psi_2[2] & \psi_3[2] & \cdots & \psi_8[2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \psi_1[7] & \psi_2[7] & \psi_3[7] & \cdots & \psi_8[7] \end{bmatrix} \begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[7] \end{bmatrix} \tag{2.1}$$

Donde pode-se obter $y_a = \begin{bmatrix} 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 \end{bmatrix}^T$. Com isso, podemos perceber que a representação de y na base de *Forrier* é mais esparsa que o vetor y em sua base original ($K(\mathbf{y}) > K(y_a)$). Acontece que se um vetor $\boldsymbol{y}$, pertecente a um espaço vetorial V de dimensão N, for representado em uma base onde ele seja suficientemente esparso, então os valores em V podem ser representados em um subespaço de dimensão M, onde M < N. Em outras palavras, caso $\boldsymbol{y}$ seja muito esparso, ele pode ser comprimido sem perdas de informações.

Definição 2 A base Ψ é dita base de esparsidade de um vetor $\mathbf{y}$ de dimensão N quando $K(\Psi \boldsymbol{y}) \ll N$.

2.7 Incoerência entre Bases

A coerência entre duas bases enuncia o máximo valor absoluto entre todos os produtos internos entre dois vetores, sendo um de cada base. Matematicamente, para duas bases ortonormais $\Psi, \Phi \in \mathbb{R}^N$, a coerência é definida como:

$$\mu(\Psi, \Phi) = \sqrt{N}\ max\{\psi\ \phi\} \tag{2.2}$$

Quando duas bases têm alta coerência, seus vetores são similares, resultando em uma grande sobreposição entre as bases. Portanto, a máxima coerência de uma base ocorre quando comparada com ela mesma. Em contraste, se duas bases forem distintas, seus vetores estarão bem separados, e a coerência entre elas será baixa (ou seja, serão altamente incoerentes). Por exemplo, na base gaussiana, os elementos de seus vetores são variáveis aleatórias independentes e identicamente distribuídos, com uma distribuição gaussiana. Essa característica resulta em baixa coerência com qualquer outra base (MEDEIROS et al., 2010).

2.8 Amostragem Discreta

O processo de amostragem é expresso por:

$$\mathbf{y_a} = \Phi \mathbf{y} \tag{2.3}$$

sendo Φ a matriz $M \times N$ ortonormal de amostragem e $\mathbf{y}$ um vetor que representa um sinal discreto $\mathbf{y} = [y[0]\ y[1]\ ...\ y[N-1]]^T$. Na amostragem tradicional de sinais, a matriz Φ é a matriz identidade, pois o sinal é amostrado de forma linear e uniforme no domínio do tempo, o que facilita sua reconstrução. No entanto, a matriz Φ pode variar, configurando diferentes tipos de amostragem e exigindo métodos distintos de reconstrução.

Exemplo 4 O vetor $\mathbf{y} = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \end{bmatrix}^T$ pode ser amostrado com a matriz Φ sendo a) Identidade, b) Gaussiana

a) $\mathbf{y_a} =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

b) $\mathbf{y_a} =$

$$\begin{bmatrix} -0,1611 & -0,2300 & 0,5382 & -0,3047 & 0,2537 & -0,3863 & 0,4766 & -0,3129 \\ -0,6204 & -0,3852 & 0,0196 & 0,2928 & 0,0786 & 0,4842 & -0,1137 & -0,3564 \\ 0,0467 & 0,1090 & -0,0823 & 0,4536 & 0,8500 & -0,1703 & -0,0470 & 0,1403 \\ 0,1072 & -0,2819 & -0,1937 & -0,5718 & 0,3713 & 0,5033 & 0,1890 & 0,3431 \\ -0,4270 & 0,0614 & -0,0381 & -0,4716 & 0,1464 & -0,3349 & -0,6710 & 0,0788 \\ 0,4436 & -0,7883 & -0,0542 & 0,1050 & -0,0391 & -0,2405 & -0,3219 & -0,0695 \\ 0,4429 & 0,2804 & 0,3199 & -0,1648 & 0,2008 & 0,3579 & -0,3576 & -0,5472 \\ -0,0140 & -0,0594 & 0,7476 & 0,1676 & -0,0762 & 0,1895 & -0,2008 & 0,5720 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

$$\mathbf{y_a} = \begin{bmatrix} -0,9222 & -0,4476 & 1,0260 & 0,4832 & 0,4284 & 0,7805 & 0,6814 & -0,6370 \end{bmatrix}^T$$

Podemos perceber que a matriz identidade preserva o vetor que representa o sinal em seu próprio domínio, mantendo inalterada sua esparsidade. Já a matriz de Fourier, como vimos no exemplo 2 deste capítulo, torna a representação do sinal mais esparsa, concentrando a informação em poucos componentes. Em contraste, a matriz Gaussiana tem a propriedade de tornar o sinal o menos esparso possível, espalhando a informação por todas as componentes.

3 Amostragem Compressiva

3.1 Introdução

A maneira como representamos a informação sempre foi um fator especial no progresso da humanidade. Questões como "Como comunicar efetivamente minhas ideias?" e "Como posso compartilhar meus sentimentos?" sempre foram recorrentes. Isso levou ao desenvolvimento de novos sistemas de medidas, tipos de escrita, músicas, entre outros métodos de comunicação. Mesmo na atualidade, essas questões ainda perduram no imaginário popular e continuam sendo relevantes.

No contexto da representação de sinais, os grandes avanços tecnológicos que surgiram com a modernidade trouxeram novos desafios. Um exemplo relevante é a complexidade na representação de informações em grande escala. Dependendo do tipo de informação, sua representação pode demandar uma grande quantidade de dados, o que exige estratégias eficientes para sua aplicação. Por exemplo, imagens que originalmente têm tamanho na ordem de megabytes podem ser representadas de forma mais eficiente no formato JPEG, ocupando apenas alguns kilobytes. Isso é possível porque, na maioria das aplicações, nem todos os dados amostrados são necessários para representar a informação de interesse.

Da mesma forma, a Amostragem Compressiva busca combinar as etapas de aquisição e compressão de dados, permitindo que a informação de interesse seja representada com um número reduzido de amostras (CANDES; ROMBERG, 2007). Para que essa técnica funcione, é necessário que os sinais sejam esparsos em alguma base conhecida. Isso não costuma ser um obstáculo, já que a maioria dos sinais pode ser representada de forma esparsa em bases adequadas.

3.2 Amostragem Aleatória

Em (CANDÈS; ROMBERG; TAO, 2006), foi proposta uma maneira de amostrar sinais abaixo da taxa de Nyquist sem necessariamente infringir o teorema associado. Para isso, eles exploraram o conhecimento prévio sobre a esparsidade dos sinais a serem amostrados.

Matematicamente, assumindo um sinal y esparso em uma base Ψ de dimensões N x N que será amostrado em uma base Φ de dimensões M x N, temos:

$$y_a = \Phi y \tag{3.1}$$

A amostragem aleatória proposta por Candès baseia-se na esparsidade da base Ψ (Seção 2.6) e na sua baixa coerência com a base Φ (Seção 2.7), de modo que, após a amostragem, a representação do sinal na base Φ não seja esparsa. O objetivo é que a informação, originalmente concentrada em poucas componentes no sinal y, seja distribuída de maneira uniforme por todos os coeficientes de y_a. É dessa forma que a redundância de um sinal esparso, aliado a um mecanismo de amostragem aleatória e identicamente distribuída entre os coeficientes do sinal, permitirá sua reconstrução mesmo quando amostrado com $M < N$ coeficientes.

Diante disso, observa-se que, na Amostragem Compressiva, além de garantir uma representação esparsa, é de fundamental importância que o sinal seja amostrado por uma base de baixa coerência com a base em que está sendo representado. Felizmente, bases aleatórias, como a Gaussiana (Seção 2.7), apresentam baixa coerência com qualquer outra base de esparsidade.

3.3 Reconstrução

Até o momento, discutimos as condições de compressão que seguem os princípios teóricos da amostragem compressiva. No entanto, um dos maiores desafios dessa técnica está na reconstrução do sinal a partir das amostras observadas. Isso ocorre porque, para reconstruir y a partir de y_a, há mais incógnitas do que equações. Matematicamente, podemos descrever as amostras como:

$$y_a(k) = \langle y, \phi_k \rangle, \quad k \in M \tag{3.2}$$

onde $M \subset \{1, \ldots, N\}$ é um subconjunto de cardinalidade $M < N$. Com base nesses conceitos, Candès propôs a recuperação do sinal original minimizando a norma ℓ_1. A reconstrução y^* pode ser expressa como $y^* = \Psi x^*$, onde x^* é a solução do seguinte problema de otimização (CANDÈS; WAKIN, 2008):

$$\min_{\tilde{x} \in \mathbb{R}^n} \|\tilde{x}\|_1 \quad \text{sujeito a} \quad y_a(k) = \langle \phi_k, \Psi \tilde{x} \rangle, \, \forall k \in M \tag{3.3}$$

Dessa fomra, entre todos os possíveis $\tilde{y} = \Psi \tilde{x}$ que são consistentes com os dados amostrados, escolhe-se aquele cuja sequência de coeficientes tem a menor norma ℓ_1 (CANDÈS; WAKIN, 2008). Este problema pode ser reformulado como um problema de programação linear, facilitando a resolução. Além disso, o algoritmo *Orthogonal Matching Pursuit* (OMP) é frequentemente empregado para buscar uma solução aproximadamente esparsa (TROPP; GILBERT; STRAUSS, 2006). Contudo, para que algoritmos de otimização, como o OMP ou métodos baseados em ℓ_1, funcionem adequadamente, é essencial que

a matriz Φ satisfaça o *Princípio da Isometria Restritiva* (do inglês *Restricted Isometry Principle* - RIP) (CANDÈS; ROMBERG; TAO, 2006).

Estabelecendo $\Phi \in \mathbb{R}^{M \times N}$, com $M < N$, dizemos que Φ satisfaz o RIP de ordem $2S$ com uma constante de isometria δ_{2S} se, para todo vetor $\mathbf{y} \in \mathbb{R}^N$ que seja S-esparso (ou seja, que tenha no máximo S entradas não nulas), a seguinte desigualdade é satisfeita:

$$(1 - \delta_{2S})\|\mathbf{y}\|_2^2 \leq \|\mathbf{\Phi y}\|_2^2 \leq (1 + \delta_{2S})\|\mathbf{y}\|_2^2 \tag{3.4}$$

Aqui, δ_{2S} deve ser suficientemente pequeno, sendo uma condição típica que $\delta_{2S} < \sqrt{2} - 1$, a fim de garantir que a reconstrução seja precisa e que o algoritmo encontre uma solução esparsa com alta probabilidade. Em outras palavras, o critério RIP garante que a matriz de compressão Φ mantenha a distância entre os vetores esparsos aproximadamente igual. Felizmente, as bases Gaussianas e de Bernoulli também obedecem ao RIP na grande maioria dos casos.

3.4 Conclusão

A Amostragem Compressiva surgiu como uma nova forma de lidar com as representações de sinais. Amostrar sinais abaixo da taxa de Nyquist, até então, poderia parecer impensável. Contudo, (CANDÈS; ROMBERG; TAO, 2006) mostrou não só que é possível, como também que não infringe a teoria de Nyquist. Isso se deve ao fato de que, na Amostragem Compressiva, estamos lidando com um sinal esparso (ou compressível), ou seja, explorar a esparsidade do sinal é a chave para o sucesso dessa técnica.

Além disso, vimos que a reconstrução do sinal amostrado se traduz em um problema de minimização de norma ℓ_1, e que esse problema pode ser resolvido por meio de programação linear e algoritmos como o OMP. Contudo, ainda hoje, a aplicação desses métodos de reconstrução em tempo real continua sendo um desafio, devido ao alto grau de complexidade computacional dessas soluções.

Neste Trabalho, nos concetramos no uso dos conceitos de Amostragem Compressiva no pré-processamento de uma rede neural classificatória de imagens. Portanto, os desafios computacionais relacionados à fase de reconstrução não foram um problema, já que o objetivo não era reconstruir o sinal amostrado, mas trabalhar com ele diretamente no domínio comprimido. Essa questão será discutida em maior profundidade no próximo capítulo. Assim, apesar da importância do conhecimento técnico e matemático quando tratamos da reconstrução dos sinais, essa etapa da amostragem compressiva não será essencial para a compreensão dos resultados deste estudo.

4 Pré-processamento via amostragem compressiva em uma Rede Neural

Este capítulo tem como objetivo apresentar o processo de utilização da Amostragem Compressiva como técnica de pré-processamento de uma rede neural artificial. Primeiramente, será feita uma revisão geral sobre o funcionamento de um modelo neural e, em seguida, detalharemos toda a metodologia utilizada no estudo, onde será possível compreender não apenas o passo a passo prático realizado, mas também as motivações teóricas que tornaram possível o uso dessa estratégia.

4.1 Visão Geral de uma Rede Neural Artificial

No último século, foi possível observar a significativa participação tecnológica das indústrias. A busca por automação e aumento de produtividade trouxe um nível adicional de complexidade para esse âmbito. Em resposta a essa necessidade, surgiram pesquisas voltadas à modelagem matemática aliada à simulação de estados futuros, tendo em vista o grande potencial de suas aplicações.

Nesse contexto, o uso de Redes Neurais Artificiais torna-se uma opção atrativa na busca por sistemas de controle compatíveis com as complexidades enfrentadas na maioria dos processos industriais, considerando o alto poder computacional atual. Prova desse nível de capacidade computacional é que o uso das RNAs transcende os processos industriais e permeia o dia a dia de grande parte da população. Por exemplo, em (VARGAS; PAES; VASCONCELOS, 2016), vemos um estudo sobre o uso de RNAs na detecção de pedestres.

As RNAs possuem, entre suas características, a capacidade de aprender por meio de exemplos e de generalizar a informação, gerando um modelo não linear. Isso as torna uma solução eficaz para problemas nos quais o comportamento das variáveis não é rigorosamente conhecido (FLECK et al., 2016). Em termos de topologia, uma rede neural é constituída por:

- **Camada de entrada:** Onde identificamos o número de variáveis que serão utilizadas para alimentar o modelo neural, sendo normalmente as variáveis de maior importância para o problema estudado.
- **Camada Oculta:** Responsável pelo processamento e transformação dos dados de entrada. Nessa camada, a rede neural aprende as representações internas por meio de ajustes nos pesos sinápticos, buscando identificar padrões complexos que não são

visíveis diretamente. Dependendo da arquitetura, pode haver uma ou mais camadas ocultas, o que aumenta a capacidade de generalização do modelo.

- **Camada de Saída:** Onde os resultados finais da rede são gerados. O número de neurônios na camada de saída corresponde ao número de variáveis de resposta ou classes de saída esperadas. Nesta fase, a rede utiliza as representações aprendidas nas camadas ocultas para produzir a resposta ou classificação para o problema estudado.

A seguir, apresentaremos uma breve visão sobre conceitos importantes das redes neurais, que abrangem desde sua arquitetura até o seu aprendizado.

4.1.1 Arquitetura

A arquitetura de uma rede neural é crucial, pois pode influenciar diretamente a solução de determinados problemas. Geralmente, três tipos de arquiteturas são as mais comuns: redes neurais recorrentes, redes com múltiplas camadas e redes com apenas uma camada. No entanto, todas elas compartilham uma estrutura básica composta por entradas conectadas a unidades de processamento, denominadas neurônios, que estão interligados por meio de pesos sinápticos (MACHADO; JUNIOR, 2013).

As entradas são modificadas pelos pesos sinápticos e pela função de ativação (AF) do modelo específico. Dada uma camada de entrada com n neurônios (y_i), o neurônio k calcula sua saída através de:

$$y_k = AF\left(\sum_{i=1}^{n} y_i w_{ki} + b_k\right) \tag{4.1}$$

Onde:

- y_i representa a saída calculada pelo neurônio.
- w_{ki} é o peso sináptico entre o neurônio i e o neurônio k.
- b_k é o termo de bias associado ao neurônio k, que permite ajustar a função de ativação ao deslocar sua curva.

Caso k seja um neurônio da primeira camada, y_i será diretamente o valor da entrada correspondente ao neurônio i. A função de ativação (AF), por sua vez, introduz a não linearidade no modelo, transformando a soma ponderada das entradas y_i pelos pesos w_{ki}, somados ao *bias* b_k, em uma saída y_k. Durante o treinamento da rede neural, os parâmetros (pesos e *bias*) são ajustados iterativamente até alcançar a convergência (MACHADO; JUNIOR, 2013). Na iteração j, o peso w_{ki} é atualizado da seguinte forma:

$$w_{ki}(j+1) = w_{ki}(j) + \Delta w_{ki}(j) \tag{4.2}$$

onde $\Delta w_{ki}(j)$ é o termo de correção aplicado ao parâmetro w_{ki} na iteração j.

4.1.2 Funções de ativação

Cada unidade de um modelo neural pode adquirir características distintas com base no padrão de transformação dos dados que a atravessam. Esse processo é regido pela função de ativação, que define como a entrada interna e o estado atual de ativação influenciam a determinação do próximo estado de ativação da unidade. Existe uma variada gama de funções de ativação, sendo as mais utilizadas:

- **Função Linear:** A função linear implica que a saída é proporcional à própria entrada, como mostrado na Figura 2a. No entanto, essa função também pode ser limitada, conforme descrito na equação:

$$f_a(u) = \begin{cases} 1 & \text{se } u \geq +\frac{1}{2} \\ u & \text{se } -\frac{1}{2} < u > +\frac{1}{2} \\ 0 & \text{se } u \leq -\frac{1}{2} \end{cases} \tag{4.3}$$

- **Função Limiar:** A função possui uma saída binária $[0, 1]$, geralmente retornando 0 quando a entrada é negativa e 1 quando a entrada é positiva. Graficamente, podemos observá-la na Figura 2b, enquanto, matematicamente, temos:

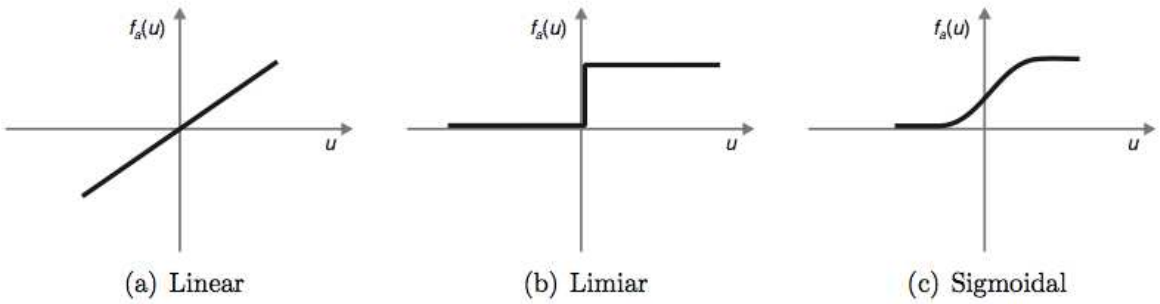
$$f_a(u) = \begin{cases} 1 & \text{se } u \geq 0 \\ 0 & \text{se } u < 0 \end{cases} \tag{4.4}$$

- **Função sigmoidal:** Trata-se da função mais popular nas arquiteturas de redes neurais, apresentando um bom equilíbrio entre linearidade e não linearidade, com valores que variam entre 0 e 1 (Figura 2c). Essa função pode ser descrita da seguinte forma:

$$f_a(u) = \frac{1}{1 + \exp(-\alpha u)} \tag{4.5}$$

Sendo α a componente que determina a inclinação da função sigmóide.

Figura 2 – a) Função Linear; b) Função Limiar; c) Função Sigmoidal



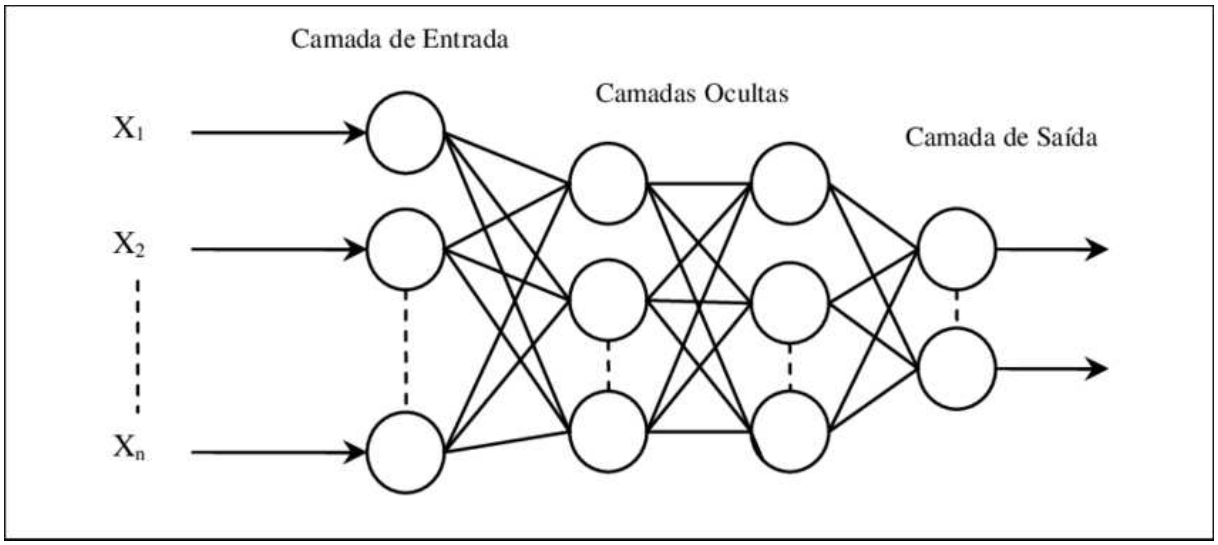
Fonte: (GUIMARãES, 2024)

4.1.3 Perceptrons de Multiplas Camadas

O perceptron é uma forma simples de RNA, com apenas uma camada, e está limitado a resolver problemas de classificação linearmente separáveis. Portanto, quando se trata da maioria dos problemas abordados por modelos neurais, a arquitetura escolhida é a do perceptron de múltiplas camadas (do inglês, *Multi-Layer Perceptron* - MLP) (SANTOS et al., 2005).

Esse tipo de arquitetura é composto por camadas de neurônios organizadas em sequência. A camada de entrada transmite as informações para as camadas intermediárias (ou camadas ocultas), e a solução é alcançada na camada de saída. Os neurônios de cada camada estão conectados apenas aos neurônios da camada imediatamente posterior, sem realimentações. Outro ponto importante é que as camadas são totalmente conectadas, como pode ser visualizado na Figura 3.

Figura 3 – Arquitetura MLP de uma Rede Neural Artificial.



Fonte: (UMA..., 2024)

No aprendizado de uma rede MLP, um padrão de ativação é propagado da camada de entrada até a camada de saída. Ao final, a camada de saída gera um conjunto de respostas

que refletem o comportamento atual da rede. Em seguida, os pesos sinápticos são ajustados de acordo com uma regra de correção de erro. Dessa forma, o erro é retropropagado, ajustando os pesos sinápticos de trás para frente, com o objetivo de aproximar a resposta real da rede à resposta desejada (NIED, 2007).

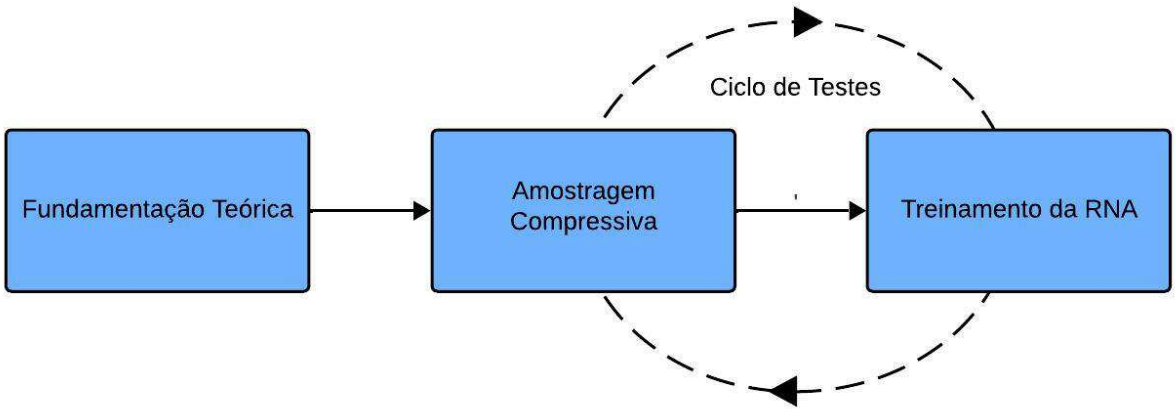
4.2 Metodologia

Para um melhor entendimento e organização geral de trabalho efetuado, todo o processo de realização da pesquisa pode ser dividido em três etapas bem definidas, a saber:

- **1ª Etapa:** Fundamentação Teórica.
- **2ª Etapa:** Implementação da Amostragem Compressiva nos Bancos de Dados.
- **3ª Etapa:** Projeto, treinamento e teste das Redes Neurais Densas.

Contudo, apesar da enumeração crescente das etapas e da ideia de sucessividade que ela sugere, a execução dessas etapas não ocorreu de maneira estritamente consecutiva. Em diversas ocasiões, foi necessário realizar esses processos de forma paralela ou, principalmente, de maneira cíclica. Por exemplo, durante o cilco de treinamento e teste da RNA foi preciso aplicar a amostragem compressiva repetidas vezes nos dados trabalhados. Portanto, essa divisão deve ser vista como um guia geral, não nos restringindo a seguir cada passo de forma isolada, mas focando no processo como um todo. O esquema da metodologia pode ser visualizado na Figura 4, e a descrição detalhada de cada etapa será apresentada ao longo desta seção.

Figura 4 – Esquema visual da metodologia.



Fonte: Autoria Própia

4.2.1 Fundamentação Teórica

Essa etapa envolveu o estudo não apenas da Amostragem Compressiva, mas também dos conceitos fundamentais necessários para seu pleno entendimento, os quais foram detalhados no Capítulo 2, como esparsidade, incoerência entre bases e amostragem discreta. Além disso, foi necessário investigar o funcionamento das redes neurais, avaliando as melhores opções de arquitetura e bancos de dados. Nessa fase, as orientações e propostas do professor Edmar Gurjão, juntamente com a tese de mestrado de Rubem José Vasconcelos (MEDEIROS et al., 2010), forneceram as motivações adequadas para a utilização dessa técnica.

Reforçando o que foi discutido no Capítulo 1, dependendo da complexidade dos dados de entrada, uma rede neural pode enfrentar dificuldades relacionadas ao alto custo computacional de seu treinamento. Esses obstáculos podem ser contornados por meio de estratégias aplicadas aos dados de entrada. Por exemplo, na classificação de imagens por CNNs, é comum utilizar técnicas de pré-processamento das imagens, de modo que características de interesse sejam realçadas ou suprimidas por convolução, e que os dados de entrada sejam reduzidos dimensionalmente através do processo de *pooling* (BHATT et al., 2021).

Com base nisso, o objetivo deste trabalho foi utilizar a Amostragem Compressiva como técnica de pré-processamento de um modelo neural. Pode-se questionar a razão do uso da Amostragem Compressiva, visto que, conforme discutido no capítulo anterior, sua implementação exige um alto poder computacional, o que seria aparentemente contrário ao objetivo inicial de um pré-processamento eficiente. Contudo, vale destacar que o elevado custo computacional está associado à reconstrução do sinal, e não à sua amostragem. Dessa aparente contradição, surge a hipótese: "e se utilizássemos os dados diretamente no domínio comprimido, eliminando a necessidade de reconstrução?".

Essa hipótese se torna promissora quando consideramos que certos vetores que representam sinais físicos, como imagens, embora não sejam naturalmente esparsos, possuem a maioria de suas componentes próximas de zero. Definimos tais sinais como quase esparsos ou compressíveis. Assim, considerando a incoerência entre bases gaussianas e qualquer outra base, juntamente com a esparsidade ou quase-esparsidade desses sinais, as condições para uma amostragem aleatória (discutidas na Seção 3.2) são atendidas.

A ideia, portanto, é selecionar bancos de dados que sejam naturalmente esparsos ou quase esparsos, aplicar a amostragem aleatória como técnica de redução dimensional nesses dados, utilizá-los como entrada em uma rede neural de classificação e comparar os resultados com aqueles obtidos com os dados originais. Espera-se que essa abordagem traga ganhos computacionais, dado o menor volume de dados de entrada, embora, simultaneamente, seja prevista uma perda de precisão na classificação da rede.

4.2.2 Implementação da Amostragem Compressiva

Para a realização desta etapa, a utilização do software *MATLAB* foi de extrema importância. Essa ferramenta oferece um ambiente de programação amplamente reconhecido por suas capacidades em análise numérica, modelagem e simulação. Além disso, o *MATLAB* disponibiliza a biblioteca "ℓ_1 - *MAGIC*", que contém rotinas para resolver problemas de otimização convexa relacionados à amostragem compressiva, incluindo algoritmos eficientes para lidar com grandes conjuntos de dados (CANDÈS, 2007).

É importante notar que a utilidade do *MATLAB* nesta etapa se restringiu à criação e ao armazenamento da matriz de amostragem $\mathbf{A}$. Para isso, definimos as dimensões $M \times N$ $(M < N)$, geramos seus coeficientes por meio de uma distribuição Gaussiana e ortogonalizamos a matriz. Em seguida, armazenamos essa matriz em um arquivo separado por vírgulas (do inglês, *Comma-Separated Values* - CSV). Na Figura 5, apresentamos a parte principal do código responsável pela geração da matriz de amostragem.

Figura 5 – Código para criação da matriz de amostragem.

```
 3
 4 -    path(path, './Optimization');
 5 -    path(path, './Data');
 6
 7 -    tamnho = 784;
 8 -    ordem = 200;
 9
10 -    disp('Creating measurment matrix...');
11 -    A = randn(ordem,tamanho);
12 -    A = orth(A')';
13 -    disp('Done.');
14
15 -    writematrix(A', 'amostragem_compressiva.csv');
```

Fonte: Autoria Própia

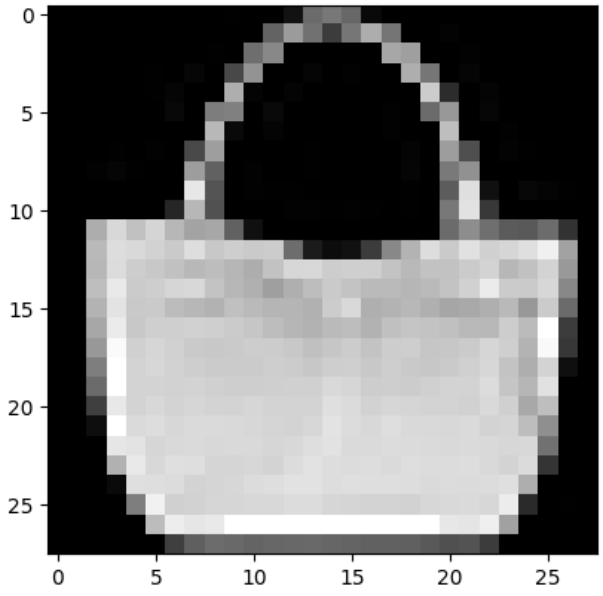
A matriz de amostragem, salva em formato CSV, foi então exportada para um ambiente de Jupyter Notebook no VScode. A partir disso, o processo de redução dimensional consistiu em tomar um vetor de informação $\mathbf{X}$ de tamanho $N \times 1$, explorar sua esparsidade em conjunto com a matriz de amostragem $\mathbf{A}$, de dimensões $M \times N$ ($M < N$), e obter um vetor $\mathbf{y} = \mathbf{A} \cdot \mathbf{X}$, de tamanho $M \times 1$, que é uma versão comprimida de $\mathbf{X}$.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}, \ sendo\ m < n \tag{4.6}$$

4.2.3 Projeto da Rede Neural

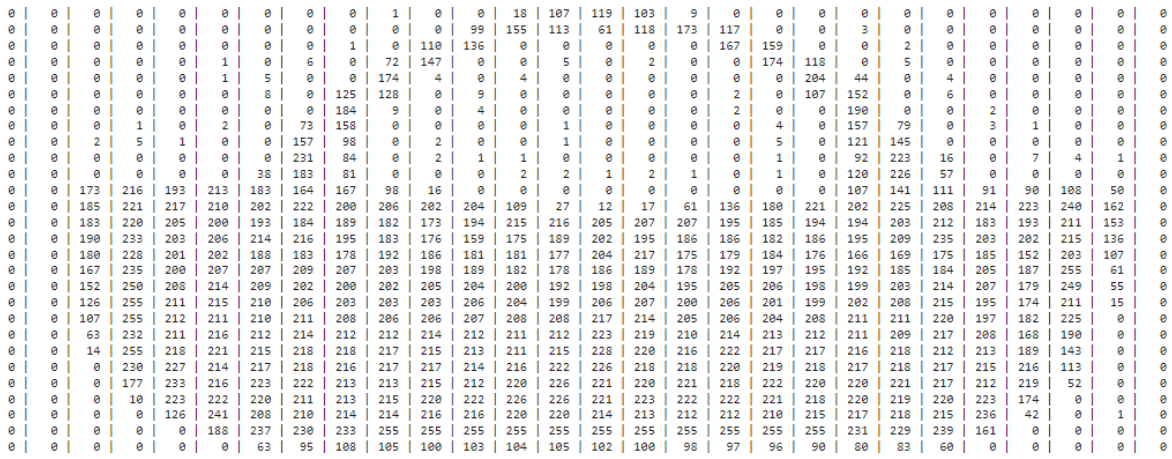
Nessa fase, nos concentramos no projeto, treinamento e teste da rede neural. O objetivo foi avaliar as mudanças ocasionadas pelo uso dos dados no domínio comprimido, tais como tempo de treinamento, número de parâmetros analisados, tempo de execução e precisão do modelo. Como mencionado anteriormente, esperava-se que o ganho computacional proporcionado pela redução dos dados resultasse em uma perda de precisão. Para comprovar essa hipótese, aplicamos a amostragem aleatória para a redução dimensional dos dados de interesse em diversos tamanhos diferentes, de modo a obter uma visão clara do *trade-off* entre ganho computacional e precisão do modelo neural.

Figura 6 – a) Bolsa em escala de cinza b) Representação Matricial



Fonte: Autoria Própia

b)



Fonte: Autoria Própia

A arquitetura escolhida para a rede neural foi uma MLP voltada para a classificação de imagens, o que influenciou diretamente a escolha dos bancos de dados. Como estávamos trabalhando com imagens, aquelas em escala de cinza foram consideradas ideais, não apenas por serem representadas de forma natural em matrizes bidimensionais, mas também pela característica de esparsidade, que poderia ser observada pela predominância de valores iguais ou próximos de zero, correspondendo ao preto na imagem (constatamos isso na Figura 6). A MLP utilizada possui três camadas ocultas, enquanto o número de neurônios na camada de entrada variou de acordo com o grau de redução dos dados.

Os hiperparâmetros do modelo neural foram definidos com base nas especificações do artigo de referência de outro estudo no qual participamos (ALSAEDI et al., 2020). Nesse estudo, foi utilizado um tipo específico de rede neural recorrente para classificar ataques de negação de serviço, alcançando bons resultados, o que nos levou a adotar uma abordagem similar nesta nova pesquisa. Esses hiperparâmetros estão detalhados na Tabela 1.

Tabela 1 – Hiperparâmetros do modelo neural

Parâmetros	**Valor/Funcão**
Camadas ocultas	3
Neurônios das camadas ocultas	1(128); 2(100); 3(64)
Épocas de treinamento	40
Função de ativação das camadas ocultas	relu
Função de ativação da camada de saída	softmax(multi-class)
Função de perda	sparse categorical crossentropy
Dropout	0.2
Otimizador	Adam

Além disso, quatro métricas diferentes foram empregadas para avaliar o modelo neural. Especificamente, são elas: *Accuracy*, *Recall*, *Precision* e *F1-Score*. A *Accuracy* avalia o modelo considerando a proporção de imagens classificadas corretamente em relação ao total de imagens. O *Recall* destaca a proporção de itens de uma classe específica que foram identificados corretamente pelo modelo. A *Precision* mostra a porcentagem de itens classificados corretamente em relação ao total de previsões feitas para essa classe. Por outro lado, o *F1-Score* calcula a média harmônica entre *Recall* e *Precision*. Essas medidas são calculadas da seguinte forma:

$$Accuracy = \frac{(VP + VN)}{VP + VN + FP + FN} \tag{4.7}$$

$$Recall = \frac{VP}{VP + FN} \tag{4.8}$$

$$Precision = \frac{VP}{VP + FP} \tag{4.9}$$

$$F - score = 2 \cdot \frac{(Recall \cdot Precision)}{Recall + Precision} \tag{4.10}$$

Sendo os verdadeiros positivos (VP) o número de imagens classificadas corretamente, verdadeiros negativos (VN) o número de detecções de outras classes feitas corretamente, falsos negativos (FN) o número de itens de uma classe classificados erroneamente como outra, e falsos positivos (FP) o número de itens de outras classes incorretamente classificados como pertencentes à classe em questão. Vale salientar que essas métricas foram utilizadas de maneira global, por exemplo, em vez de apenas olhar para o recall de uma única classe, calculamos o recall para cada classe e, em seguida, tiramos uma média.

4.3 Conclusão

Neste capítulo, foram apresentados os conceitos fundamentais de uma rede neural, incluindo sua arquitetura, funções de ativação e o processo de atualização dos pesos. Com isso, foi possível fornecer uma visão geral da arquitetura utilizada na pesquisa: o perceptron de múltiplas camadas. Ademais, a partir da revisão dos conceitos do capítulo 2 e do aprofundamento teórico sobre amostragem compressiva discutido no capítulo 3, não apenas se evidenciaram as motivações da pesquisa, mas também foi detalhado o processo completo de aplicação dessa técnica como etapa de pré-processamento de um modelo MLP voltado para a classificação de imagens. Com isso, finalmente podemos nos concentrar na exposição e discussão dos resultados obtidos.

5 Resultados e Discussões

Nesse capítulo, apresentaremos os resultados obtidos durante a pesquisa, divididos em duas seções correspondentes aos dois bancos de dados selecionados. A escolha desses bancos de dados não foi aleatória, sua seleção foi baseada nos seguintes critérios:

- Imagens na escala de cinza
- Naturalmente esparsos ou quase esparsos
- Facilidade de implementação
- Banco de dados consolidado

5.1 Banco de dados MNIST

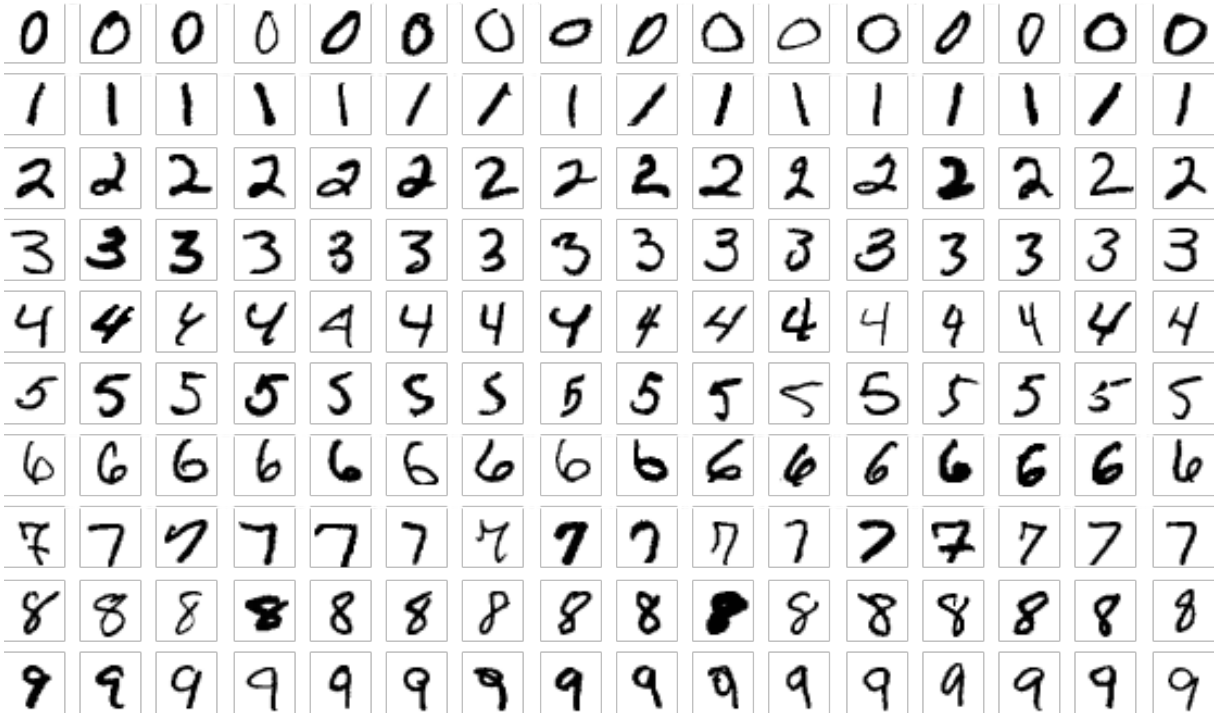
O banco de dados MNIST (do inglês *Modified National Institute of Standards and Technology*) (LECUN; BOTTOU; HAFFNER, 1998) é um conjunto de dados muito conhecido no treinamento e validação de técnicas de aprendizado de máquina desde 1998, ano de sua criação. Trata-se de uma coleção de imagens em escala de cinza de dígitos manuscritos, formatadas em matrizes de 28x28 pixels e associadas a um rótulo de 10 classes distintas. O conjunto possui 60.000 imagens destinadas ao treinamento dos modelos e 10.000 para teste. A Figura 7 apresenta uma visualização do banco de dados MNIST.

Para o pré-processamento da rede neural, os dados organizados em matrizes de dimensão 28×28 foram transformados em vetores coluna de tamanho 784×1, normalizados e utilizados como entrada para o modelo. Em uma imagem digital, cada pixel é representado por um valor de 0 a 255, de modo que a normalização foi realizada simplesmente pela divisão de todos os valores da matriz por 255. Esse processo consiste em um escalonamento pelo valor máximo:

$$x_{\text{norm}} = \frac{x}{x_{\text{max}}} \tag{5.1}$$

A rotina de treinamento e teste do modelo baseou-se na redução dimensional por amostragem aleatória do vetor de entrada, isto é, o vetor coluna que representa a imagem. Nesse contexto, a rede neural foi treinada, validada e testada utilizando as métricas de *Accuracy*, *Precision*, *F1-score* e *Recall*, considerando um conjunto variado de vetores de entrada e, consequentemente, de neurônios na camada de entrada.

Figura 7 – Amostra de imagens do Banco de Dados MNIST



Fonte: (SCIENCE, 2024).

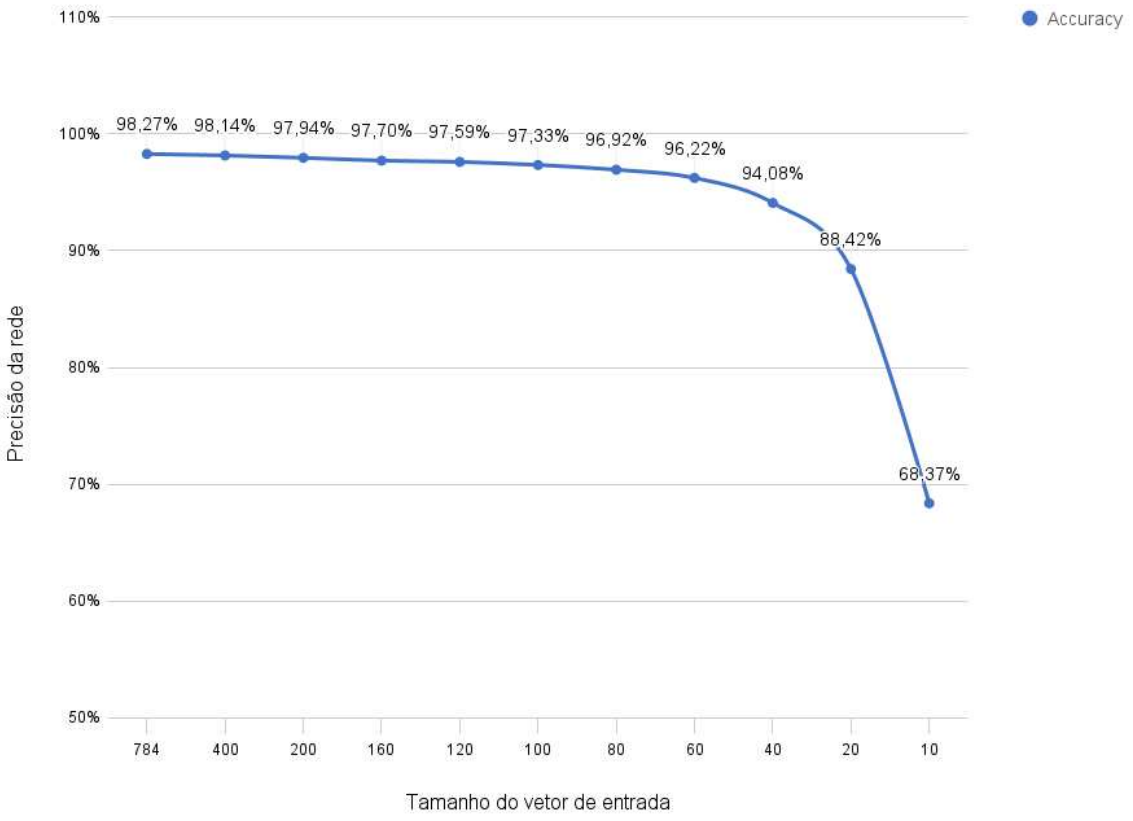
Além disso, a rede neural foi treinada durante 40 épocas, com o tempo de execução registrado em todos os casos analisados. Na Tabela 2, é possível visualizar a relação entre o vetor de entrada, o resultado das métricas utilizadas e o tempo de treinamento da RNA. É importante destacar que o vetor de entrada de dimensão 784×1 representa a informação original, enquanto os demais tamanhos correspondem às versões no domínio comprimido.

Tabela 2 – Resultados do modelo neural para os diferentes tamanhos do vetor de entrada.

Mnist					
Vetor de Entrada ($N \times 1$)	**Accuracy** (%)	**Precision** (%)	**F-score** (%)	**Recall** (%)	**Tempo de Treino**
784	98,27	98,27	98,27	98,27	157,7 s
400	98,14	98,14	98,14	98,14	134,2 s
200	97,94	97,94	97,94	97,94	108,6 s
160	97,70	97,71	97,70	97,70	106,8 s
120	97,59	97,59	97,59	97,59	105,2 s
100	97,33	97,34	97,33	97,33	102,7 s
80	96,92	96,92	96,92	96,92	102,4 s
60	96,22	96,23	96,22	96,22	100,4 s
40	94,08	94,08	94,07	94,08	100 s
20	88,42	88,45	88,41	88,42	98,5 s
10	68,37	68,29	68,13	68,37	99,5 s

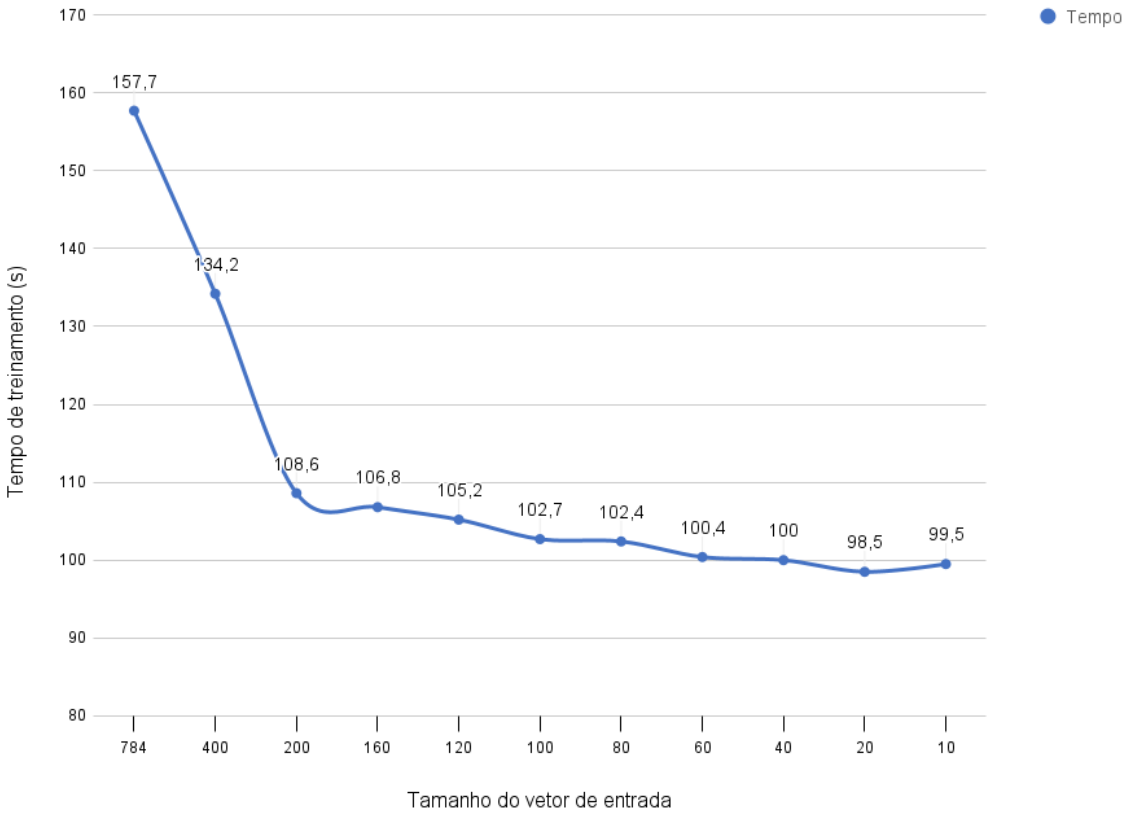
Inicialmente, ao analisarmos os resultados da Tabela 2, constatamos um comportamento decrescente tanto no tempo de treinamento quanto na precisão do modelo neural construído. No entanto, em um segundo momento, observamos que, a partir do vetor comprimido de dimensão 100×1, o tempo de treinamento se estabiliza em torno de 100 segundos, enquanto a precisão do modelo continua a diminuir. Essa característica torna-se mais evidente nas Figuras 8 e 9.

Figura 8 – Gráfico que relaciona a *Accuracy* do modelo neural com o tamanho do vetor de entrada para o banco de dados MNIST.



Fonte: Autoria Própia

Figura 9 – Gráfico que relaciona o tempo de treinamento do modelo neural com o tamanho do vetor de entrada para o banco de dados MNIST.

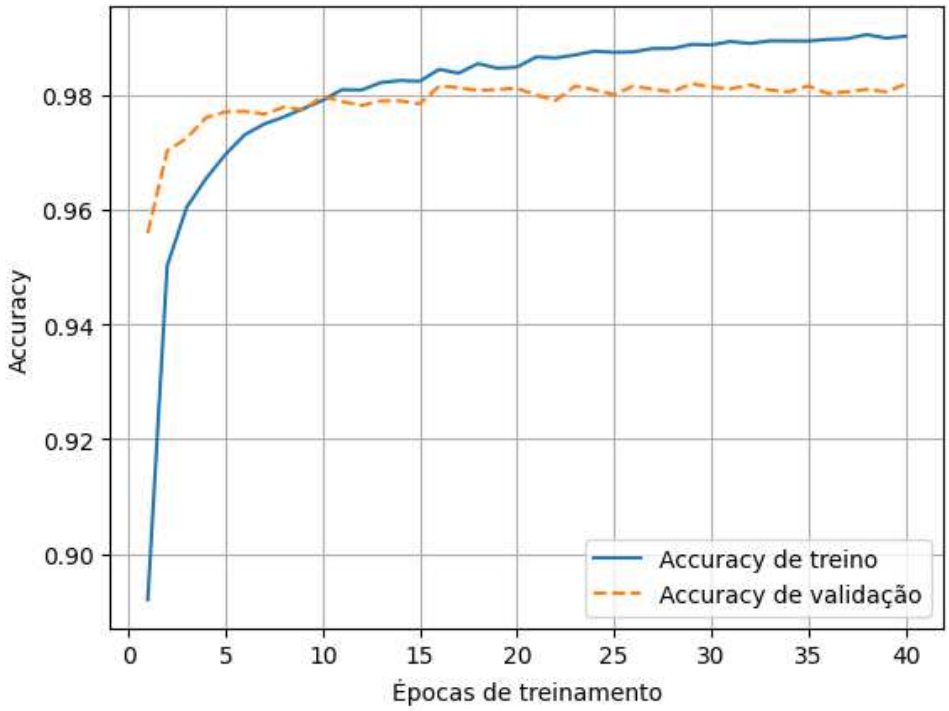


Fonte: Autoria Própia

Com base nos resultados apresentados, fica evidente que algumas reduções são mais vantajosas do que outras, tornando necessário um equilíbrio cuidadoso entre o ganho computacional e a perda de precisão do modelo neural. Um bom exemplo é o vetor de entrada com dimensão 100×1, no qual foi possível obter uma redução de 34,9% do tempo de treinamento do modelo (equivalente a 55 segundos), resultando em uma queda de aproximadamente 0,95% na precisão da RNA, quando comparado ao vetor de entrada original.

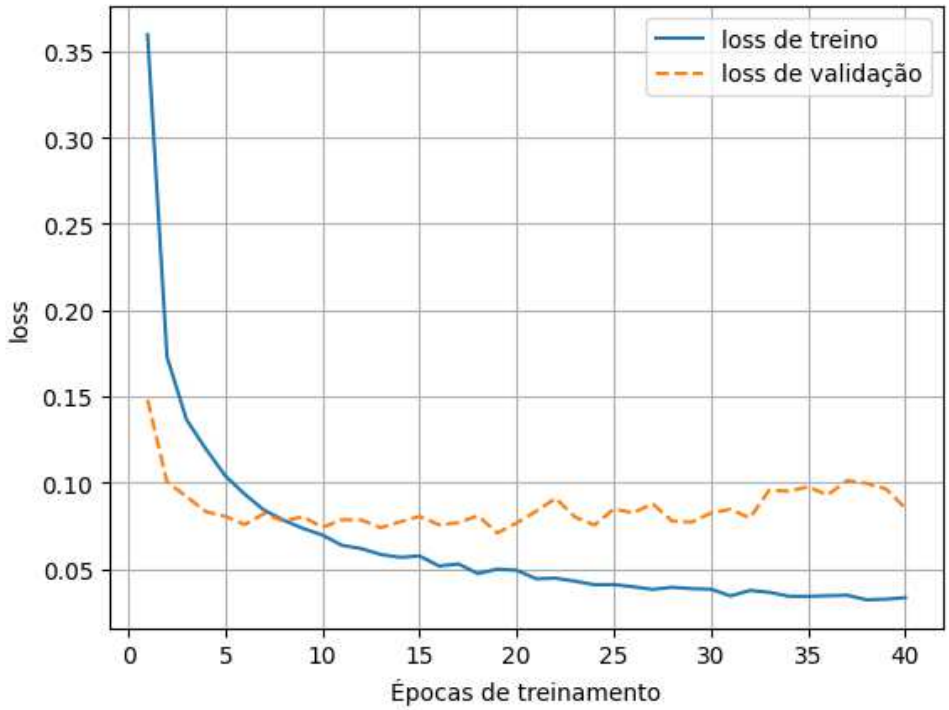
As altas precisões obtidas pela rede neural poderiam inicialmente sugerir um possível sobreajuste (em inglês, *overfitting*). No entanto, essa hipótese é descartada ao analisarmos o comportamento das precisões e erros de treinamento e validação ao longo das 40 épocas, pois o gráfico de validação acompanha consistentemente o de treinamento. Nas Figuras 10 a 13, observa-se essa análise tanto para os dados originais quanto para os dados comprimidos de dimensão 100×1.

Figura 10 – Gráfico da *Accuracy* de treino e validação ao longo das épocas de treinamento para os dados originais (784×1) do banco de dados MNIIST.



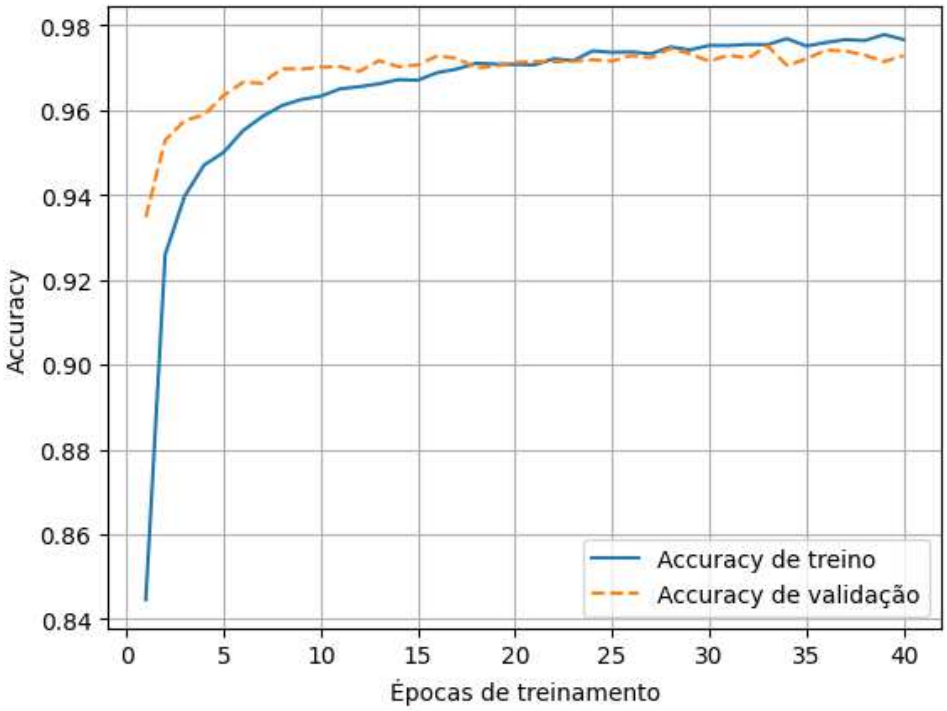
Fonte: Autoria Própia

Figura 11 – Gráfico do *loss* de treino e validação ao longo das épocas de treinamento para os dados originais (784×1) do banco de dados MNIIST.



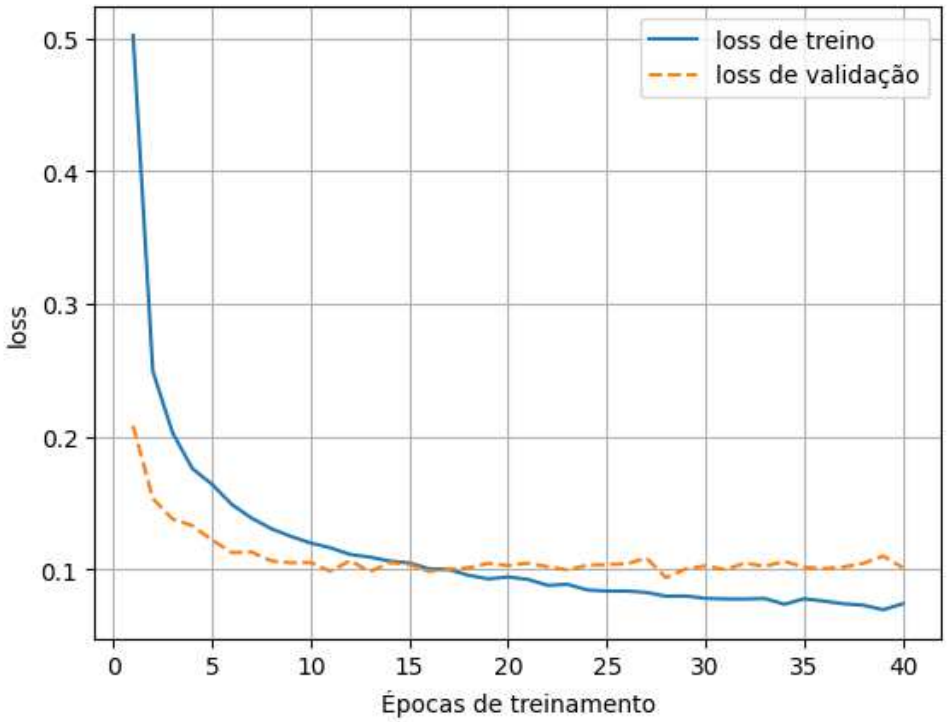
Fonte: Autoria Própia

Figura 12 – Gráfico da *Accuracy* de treino e validação ao longo das épocas de treinamento para os dados comprimidos (100×1) do banco de dados MNIIST.



Fonte: Autoria Própia

Figura 13 – Gráfico do *loss* de treino e validação ao longo das épocas de treinamento para os dados comprimidos (100×1) do banco de dados MNIIST.



Fonte: Autoria Própia

5.2 Banco de dados Fashion MNIST

O banco de dados Fashion MNIST (XIAO; RASUL; VOLLGRAF, 2017) foi criado em 2017 com o objetivo de suceder o MNIST, já que compartilha o mesmo formato de imagens, bem como a estrutura de treinamento e teste, mas com um nível adicional de complexidade. Ele consiste em uma coleção de imagens em escala de cinza de artigos de moda, divididos em 10 categorias diferentes, formatadas em matrizes de 28×28 pixels e associadas a um rótulo correspondente a uma das 10 classes. O conjunto possui 70.000 imagens para treinamento dos modelos e 10.000 para teste. A Figura 14 apresenta uma visualização do banco de dados Fashion MNIST.

Figura 14 – Amostra de imagens do Banco de Dados Fashion MNIST.

Fonte: (PYLYPENKO, 2024)

O pré-processamento da rede neural foi similar ao utilizado no banco de dados MNIST, em que os dados organizados em matrizes de dimensão 28×28 foram transformados em vetores coluna de dimensão 784×1 e normalizados conforme a Equação 5.1. Além disso, foi empregada a mesma rotina de treinamento e teste, baseada nos diferentes níveis de compressão dos dados originais e nas métricas: *Accuracy*, *Precision*, *F1-score* e *Recall*.

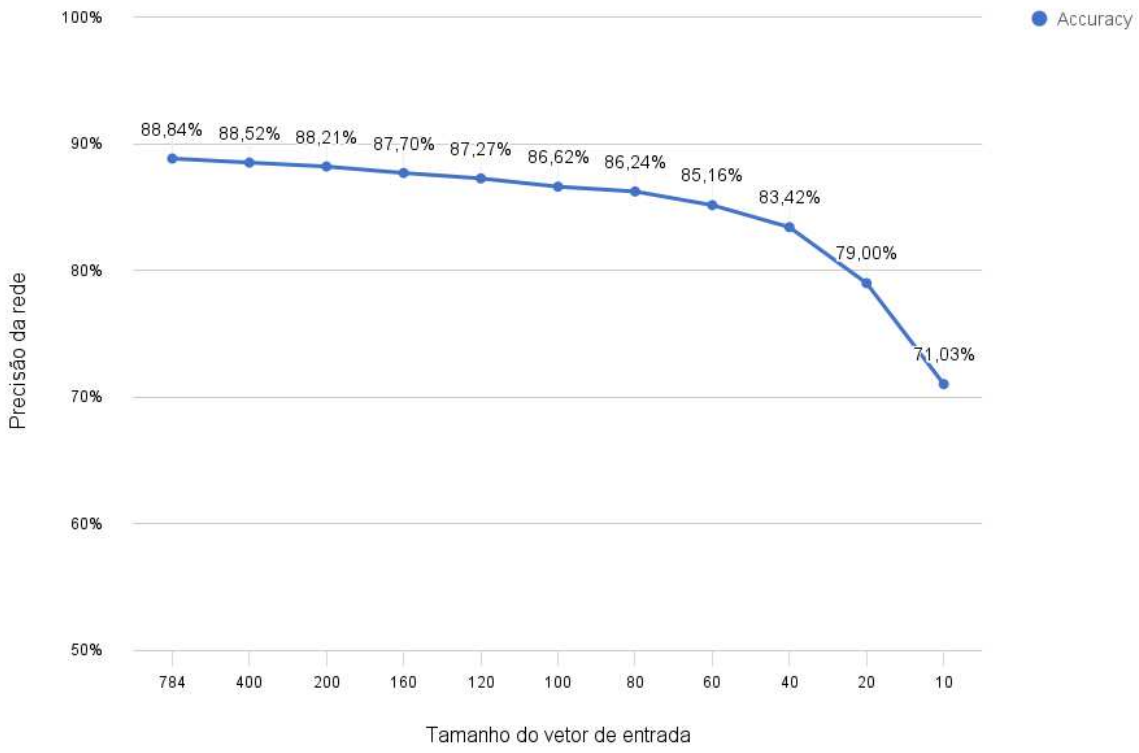
Assim como descrito na seção anterior, a rede neural foi treinada por 40 épocas, com o tempo de execução registrado em todos os casos analisados. A Tabela 3 apresenta a relação entre o vetor de entrada, os resultados das métricas utilizadas e o tempo de treinamento da RNA. Observa-se que esses resultados seguem um comportamento decrescente semelhante

ao identificado na Tabela 2. No entanto, desta vez, é a partir do vetor de dimensão 120×1 que o tempo de treinamento se estabiliza em torno de 100 segundos, enquanto a precisão do modelo continua a diminuir. Podemos visualizar essa característica novamente nas Figuras 15 e 17.

Tabela 3 – Resultados do modelo neural para os diferentes tamanhos do vetor de entrada.

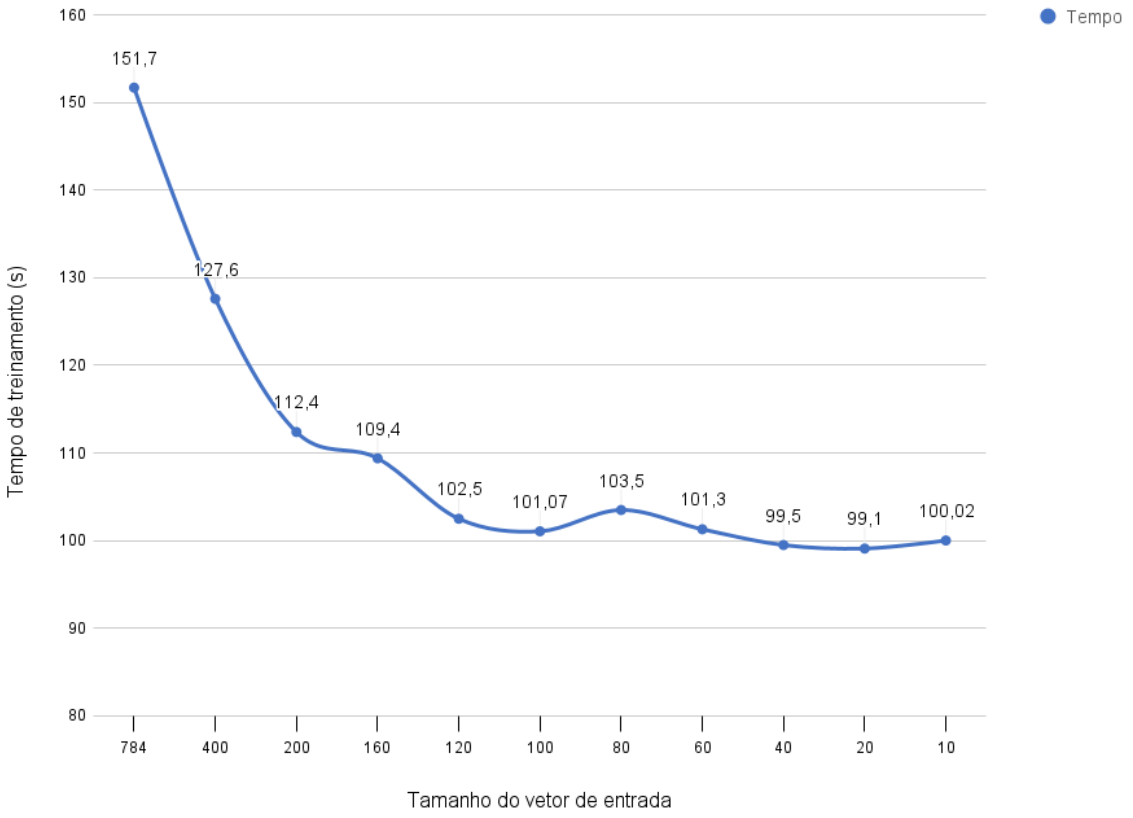
Fashion_Mnist					
Vetor de Entrada ($N \times 1$)	**Accuracy** (%)	**Precision** (%)	**F-score** (%)	**Recall** (%)	**Tempo de Treino**
784	88,84	88,88	88,77	88,84	151,7 s
400	88,32	88,28	88,21	88,32	127,60 s
200	88,21	88,20	87,99	88,21	112,4 s
160	87,70	87,73	87,71	87,70	109,4 s
120	87,27	87,29	87,22	87,27	102,5 s
100	86,62	86,58	86,56	86,62	101,07 s
80	86,24	86,18	86,17	86,24	103,5 s
60	85,16	85,21	85,11	85,16	101,3 s
40	83,42	83,40	83,28	83,42	99,5 s
20	79	78,92	78,73	79	99,1 s
10	71,03	70,23	70,34	71,03	100,02 s

Figura 15 – Gráfico que relaciona a *Accuracy* do modelo neural com o tamanho do vetor de entrada para o banco de dados Fashion MNIST.



Fonte: Autoria Própia

Figura 16 – Gráfico que relaciona o tempo de treinamento do modelo neural com o tamanho do vetor de entrada para o banco de dados Fashion MNIST.

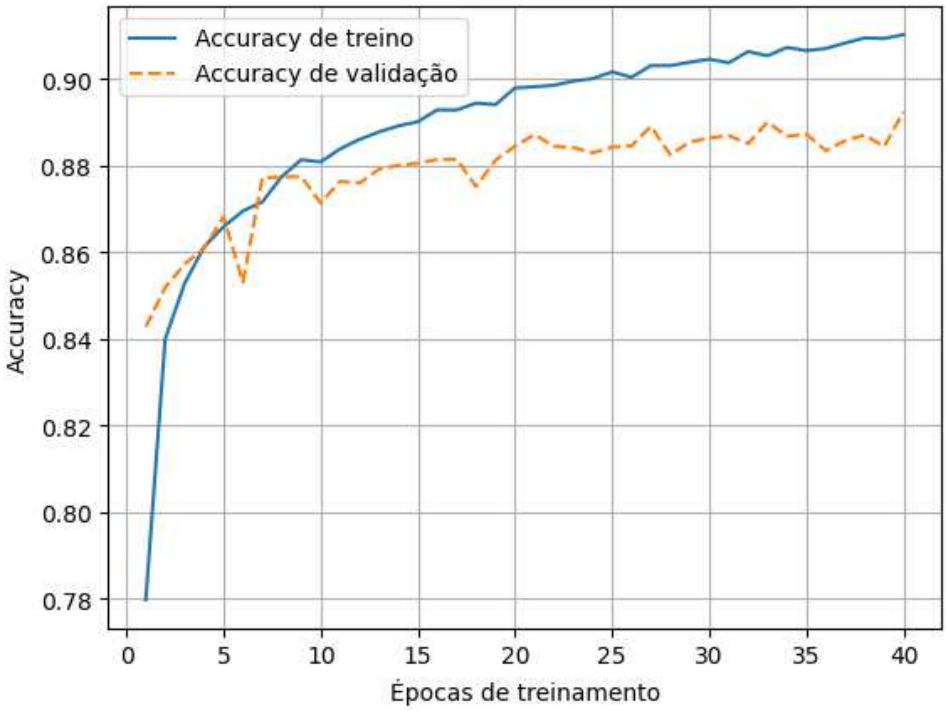


Fonte: Autoria Própia

Analisando os gráficos das Figuras 15 e 17, podemos equilibrar o ganho computacional e a perda de precisão do modelo neural causados pela redução dimensional do vetor de entrada. Nesse caso, o melhor exemplo a ser citado é o vetor de entrada com dimensão 120×1, no qual foi possível obter uma redução de 32,4% no tempo de treinamento do modelo (equivalente a 49,2 segundos), resultando em uma queda de aproximadamente 1,57% na precisão da RNA, quando comparado ao vetor de entrada original.

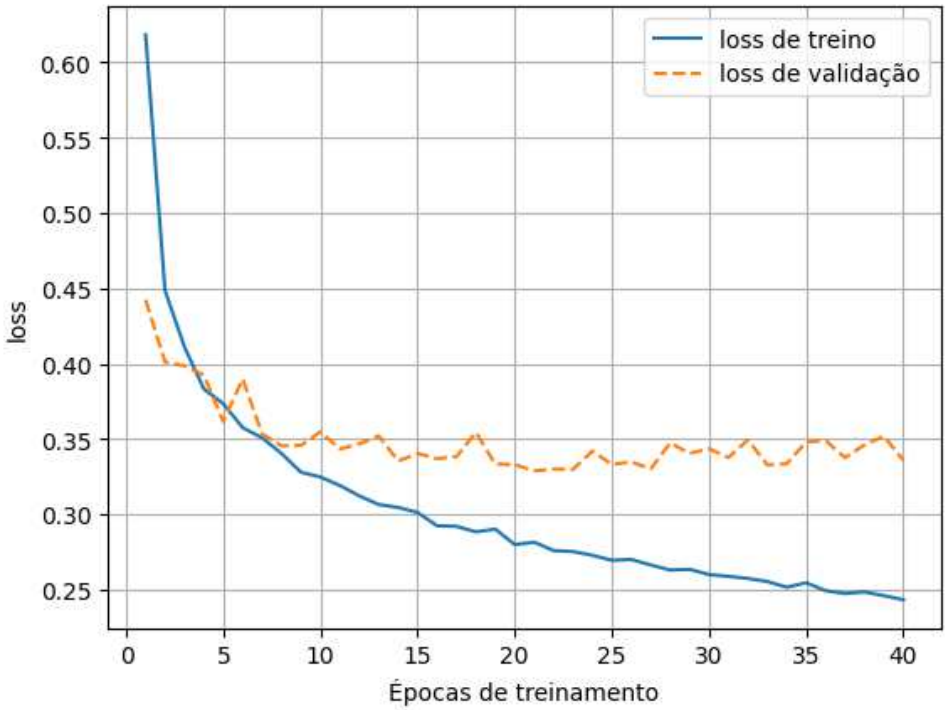
Assim como no caso do banco de dados MNIST, avaliamos a presença de *overfitting* no treinamento do conjunto de dados. Para isso, analisamos o comportamento das precisões e dos erros de treinamento e validação ao longo das 40 épocas. Nas Figuras 17 à 20, observamos que os gráficos de validação acompanham consistentemente o de treinamento, tanto para os dados originais quanto para os comprimidos, o que é um forte indicativo da ausência de sobreajuste na RNA.

Figura 17 – Gráfico da *Accuracy* de treino e validação ao longo das épocas de treinamento para os dados comprimidos (784×1) do banco de dados Fashion MNIST.



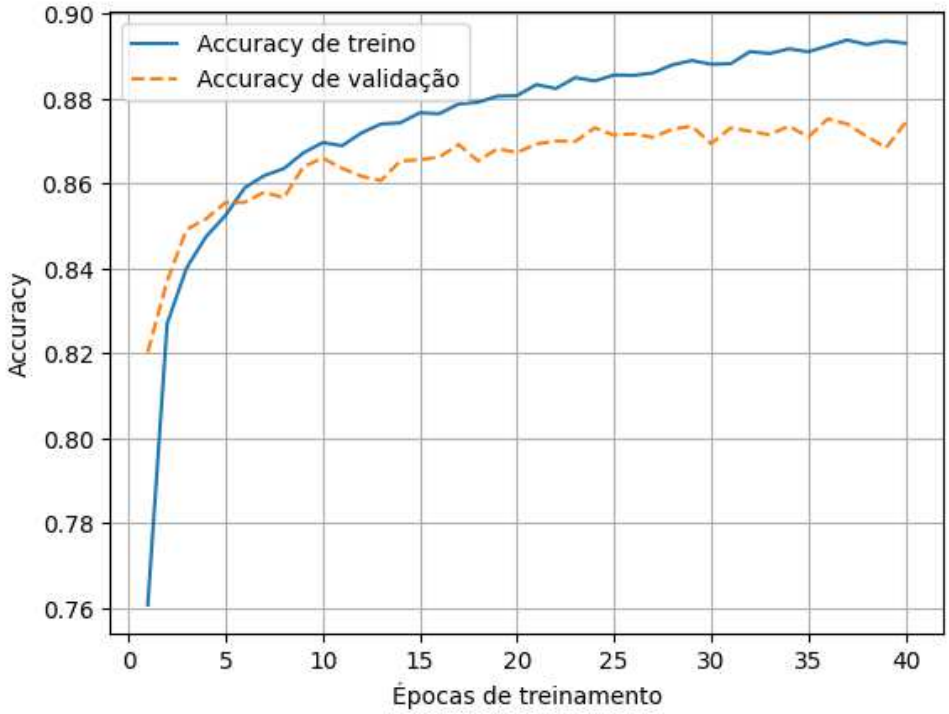
Fonte: Autoria Própia

Figura 18 – Gráfico do *loss* de treino e validação ao longo das épocas de treinamento para os dados originais (784×1) do banco de dados Fashion MNIST.



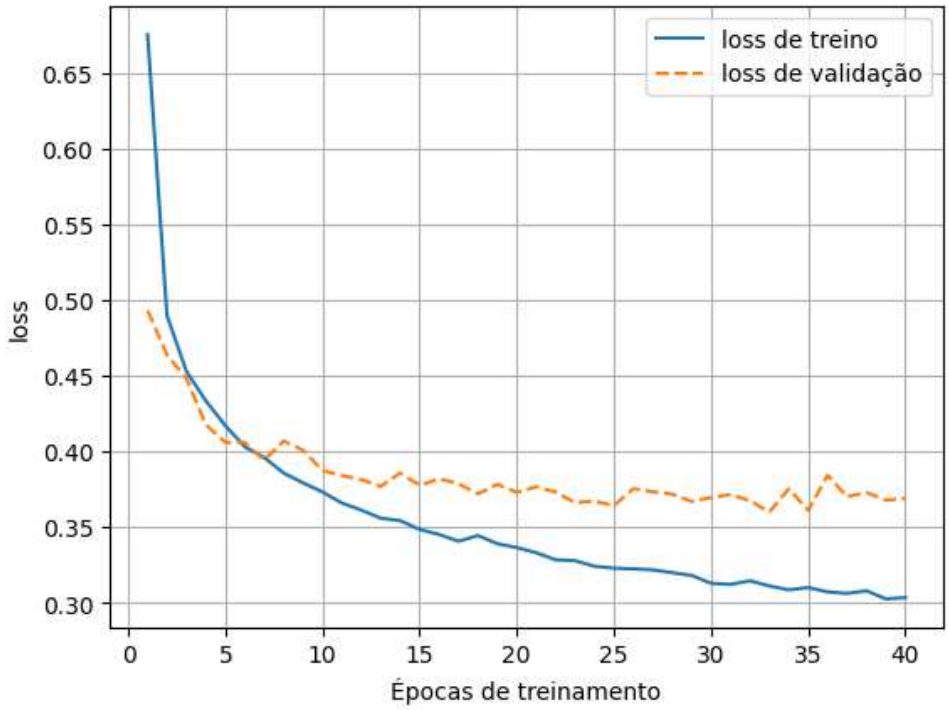
Fonte: Autoria Própia

Figura 19 – Gráfico da *Accuracy* de treino e validação ao longo das épocas de treinamento para os dados comprimidos (120×1) do banco de dados Fashion MNIST.



Fonte: Autoria Própia

Figura 20 – Gráfico do *loss* de treino e validação ao longo das épocas de treinamento para os dados comprimidos (120×1) do banco de dados Fashion MNIST.



Fonte: Autoria Própia

6 Conclusão

Inicialmente, discutimos o uso crescente das RNAs nos mais diversos contextos, desde aplicações cotidianas, como sugestões de busca no Google e geração de música, até cenários mais específicos, como previsões de comportamento na bolsa de valores e classificação de imagens. Além disso, abordamos os consequentes desafios dessa ampla gama de aplicações, com destaque para a elevada demanda computacional exigida para processar grandes volumes de dados, e como essas dificuldades podem ser contornadas por meio de estratégias de pré-processamento.

Foi tratado também sobre o funcionamento da técnica de amostragem compressiva, que busca amostrar e reconstruir sinais abaixo da taxa de Nyquist, aproveitando a redundância presente em sinais representados em bases esparsas, em conjunto com a baixa coerência de bases gaussianas no processo de amostragem. Detalhamos as condições necessárias para a aplicação desta técnica, assim como o funcionamento de uma Rede Neural Artificial.

Nesse contexto, o objetivo deste trabalho foi avaliar a viabilidade do uso da amostragem compressiva como técnica de redução dimensional no pré-processamento de redes neurais artificiais. Ao reduzir a quantidade de dados necessários para o treinamento dos modelos, a amostragem compressiva mostrou-se uma alternativa promissora, especialmente em cenários onde o custo computacional é um fator crítico. Através de experimentos com os bancos de dados MNIST e Fashion MNIST, observou-se uma redução significativa no tempo de treinamento, com impacto mínimo na precisão dos modelos. No caso do MNIST, por exemplo, houve uma redução de 34,9% no tempo de execução, com uma queda de apenas 1% na precisão da rede neural. Os resultados sugerem que a técnica pode ser mais eficaz em dados de menor complexidade, embora seja necessário testar mais bancos de dados para confirmar essa hipótese.

Embora a amostragem compressiva seja tradicionalmente associada à reconstrução de sinais, este estudo concentrou-se no processamento direto dos dados no domínio comprimido, eliminando a etapa de reconstrução e, assim, reduzindo o custo computacional sem comprometer significativamente o desempenho do modelo. Os resultados indicam que é possível atingir um equilíbrio entre eficiência computacional e precisão, especialmente em cenários onde pequenas perdas de precisão são aceitáveis em troca de uma maior velocidade de processamento.

Por fim, este trabalho contribui para o campo do aprendizado de máquina ao apresentar uma nova abordagem que combina técnicas de compressão de dados com redes neurais densas. A estratégia proposta não só afirma a viabilidade dessa técnica, como

também abre novas possibilidades para investigações futuras, com o potencial de explorar diferentes tipos de dados e arquiteturas neurais, ampliando o escopo de aplicação da amostragem compressiva em diversos domínios.

6.1 Perspectivas para o Futuro

Considerando o caráter emergente da amostragem compressiva como técnica de pré-processamento em redes neurais artificiais, fica evidente o grande potencial a ser explorado nessa tema. A seguir, estão alguns tópicos que podem ser investigados e aprofundados futuramente:

- **Arquitetura da RNA**: Utilizar outros tipos de arquiteturas de RNA na aplicação da técnica.
- **Uso de algoritmos de aprendizado de máquina**: Testar o comportamento dessa técnica com métodos de aprendizado de máquina diferentes de uma RNA.
- **Aumentar a complexidade dos bancos de dados**: Observar o comportamento da técnica no pré-processamento de dados mais complexos e com menos redundância.
- **Explorar bases esparsas**: Procurar bases esparsas correspondentes aos dados trabalhados e aplicar a técnica.

Referências Bibliográficas

ALSAEDI, A. et al. Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems. *IEEE Access*, IEEE, v. 8, p. 165130–165150, 2020. Citado na página 19.

BHATT, D. et al. Cnn variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, MDPI, v. 10, n. 20, p. 2470, 2021. Citado 2 vezes nas páginas 1 e 16.

CANDES, E.; ROMBERG, J. Sparsity and incoherence in compressive sampling. *Inverse problems*, IOP Publishing, v. 23, n. 3, p. 969, 2007. Citado na página 8.

CANDÈS, E. J. *L1-MAGIC: Recovery of Sparse Signals via L1 Minimization*. 2007. Accessado: 8 de outubro de 2024. Disponível em: <https://candes.su.domains/software/l1magic/>. Citado na página 17.

CANDÈS, E. J.; ROMBERG, J.; TAO, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, IEEE, v. 52, n. 2, p. 489–509, 2006. Citado 2 vezes nas páginas 8 e 10.

CANDÈS, E. J.; WAKIN, M. B. An introduction to compressive sampling. *IEEE signal processing magazine*, IEEE, v. 25, n. 2, p. 21–30, 2008. Citado 2 vezes nas páginas 1 e 9.

ESTUDO de Técnicas de Realce de Imagens Digitais e Suas Aplicações. Scientific Figure on ResearchGate. Accessed: 10 Oct 2024. Disponível em: <https://www.researchgate.net/figure/Figura-2-Representacao-de-uma-imagem-digital-3-PRE-PROCESSAMENTO-O-pre-proce\ssamento-de_fig1_267712029>. Citado na página 4.

FENG, S. et al. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, Nature Publishing Group UK London, v. 12, n. 1, p. 748, 2021. Citado na página 1.

FLECK, L. et al. Redes neurais artificiais: Princípios básicos. *Revista Eletrônica Científica Inovação e Tecnologia*, v. 1, n. 13, p. 47–57, 2016. Citado na página 11.

GOLUB, G. H.; LOAN, C. F. V. *Matrix Computations*. 4. ed. [S.l.]: Johns Hopkins University Press, 2013. Citado na página 5.

GUIMARãES, A. *Tutorial 05: Redes Neurais com Scikit-Learn*. 2024. Accessed: 10 Oct 2024. Disponível em: <https://notebook.community/adolfoguimaraes/machinelearning/SupervisedLearning/Tutorial05_RedesNeuraisComScikitLearn>. Citado na página 14.

LECUN, Y.; BOTTOU, L.; HAFFNER, P. *MNIST handwritten digit database*. 1998. Accessado: 8 de outubro de 2024. Disponível em: <http://yann.lecun.com/exdb/mnist/>. Citado na página 21.

MACHADO, W. C.; JUNIOR, E. S. d. F. Redes neurais artificiais aplicadas na previsão do vtec no brasil. *Boletim de Ciências Geodésicas*, SciELO Brasil, v. 19, p. 227–246, 2013. Citado na página 12.

MEDEIROS, R. J. V. d. et al. Investigação sobre aplicação de amostragem compassiva a sinais de áudio. 2010. Dissertação de Mestrado, Unidade Acadêmica de Engenharia Elétrica, Universidade Federal de Campina Grande. Citado 3 vezes nas páginas 1, 6 e 16.

MUKHERJEE, S.; SHARMA, N. Intrusion detection using naive bayes classifier with feature reduction. *Procedia Technology*, Elsevier, v. 4, p. 119–128, 2012. Citado na página 1.

NIED, A. Treinamento de redes neurais artificiais baseado em sistemas de estrutura variável com taxa de aprendizado adaptativa. Universidade Federal de Minas Gerais, 2007. Citado na página 15.

PYLYPENKO, I. *Exploring Neural Networks with Fashion MNIST*. 2024. Accessado: 8 de outubro de 2024. Disponível em: <https://medium.com/@ipylypenko/exploring-neural-networks-with-fashion-mnist-b0a8214b7b7b>. Citado na página 27.

SANTOS, A. M. d. et al. Usando redes neurais artificiais e regressão logística na predição da hepatite a. *Revista Brasileira de Epidemiologia*, SciELO Public Health, v. 8, p. 117–126, 2005. Citado na página 14.

SARACCO, R. Congrats xiaoyi. you are now a medical doctor. *IEEE Future Directions*, 2017. Acesso em: 7 fev. 2021. Disponível em: <https://cmte.ieee.org/futuredirections/2017/12/02/congrats-xiaoyi-you-are-now-a-medical-doctor/>. Citado na página 1.

SCIENCE, T. D. *Part 5: Training the Network to Read Handwritten Digits*. 2024. Accessado: 8 de outubro de 2024. Disponível em: <https://towardsdatascience.com/part-5-training-the-network-to-read-handwritten-digits-c2288f1a2de3>. Citado na página 22.

TROPP, J. A.; GILBERT, A. C.; STRAUSS, M. J. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal processing*, Elsevier, v. 86, n. 3, p. 572–588, 2006. Citado na página 9.

UMA ESTIMAÇÃO DO VALOR DA COMMODITY DE AÇÚCAR UTILIZANDO REDES NEURAIS ARTIFICIAIS - Scientific Figure on ResearchGate. 2024. Acessado em 10 de outubro de 2024]. Disponível em: <https://www.researchgate.net/figure/Figura-2-Perceptron-Multicamadas-Fonte-Adaptado-de-LNCC-2008_fig1_228432919>. Citado na página 14.

VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: SN. *Proceedings of the xxix conference on graphics, patterns and images*. [S.l.], 2016. v. 1, n. 4. Citado na página 11.

WINTERLE, P.; STEINBRUCH, A. *Álgebra linear*. [S.l.]: 1987, 1987. Citado na página 4.

XIAO, H.; RASUL, K.; VOLLGRAF, R. *Fashion MNIST: A Novel Image Dataset for the Fashion Community*. 2017. Accessado: 8 de outubro de 2024. Disponível em: <https://github.com/zalandoresearch/fashion-mnist>. Citado na página 27.