APLICANDO REDES NEURAIS PARA REDUZIR ESPAÇO UTILIZADO NO ARMAZENAMENTO DE CONJUNTOS DE DADOS

Rafael Martins Trindade [1]
Ricardo Petri Silva [2]

RESUMO

Ao trabalhar com Aprendizagem de Máquina, existem casos em que manipular o conjunto de dados pode se tornar trabalhoso ou demorado devido à grande quantidade de informação. Apesar de existirem técnicas de redução de dimensionalidade como a análise de componentes principais ou a análise discriminante linear que podem ser utilizadas para reduzir a quantidade de atributos, elas não são capazes de reduzir o conjunto de dados sem perda de informação. Esse trabalho busca investigar a possibilidade de empregar técnicas que transformam dados de formas que seriam extremamente difíceis de reverter ao combiná-las com o uso de redes neurais para essa restauração dos dados, sem que haja perda de informação. Para isso, foi criada uma prova de conceito utilizando a base de dados clássica de flores Iris. Apesar de não terem sido capazes de restaurar o conjunto de dados original, os modelos testados foram capazes de obter valores muito próximos dos originais. Por se tratar de um estudo inicial, houveram diversas limitações nos testes realizados. Ainda que o objetivo proposto não tenha sido atingido, ficou evidente que ainda existe potencial inexplorado para este conceito.

Palavras-chave: aprendizagem de máquina; rede neural; redução de dimensionalidade; conjunto de dados; armazenamento de dados.

ABSTRACT

Working with Machine Learning, there are cases when manipulating datasets can become cumbersome or slow due to the great amount of data. Despite there being dimensionality reduction techniques such as principal component analysis or linear discriminant analysis that can be used to reduce the amount of attributes, these techniques are not able to reduce the dataset without information loss. This article aims to investigate the possibility of employing techniques which transform data in ways that would be extremely hard to revert by combining them with the use of neural networks for the data restoration, with no information loss. For this purpose, a proof of concept was made using the iris dataset. Although they weren't able to restore the original datasets, the tested models were able to produce values very close to the originals. Being an initial study, there were many limitations on the tests done. So even though

[1] Centro Universitário Filadélfia de Londrina - UniFil
[2] Centro Universitário Filadélfia de Londrina - UniFil

the proposed goal was not reached, it became clear that there's still much unexplored potential for this concept.

Keywords: machine learning; neural network; dimensionality reduction; dataset; data storage.

1 INTRODUÇÃO

A Inteligência Artificial (IA) e a Aprendizagem de Máquina (AM), que é uma de suas vertentes, têm sido adotadas cada vez mais em diversas áreas nos últimos anos, seja no desenvolvimento de veículos autônomos (Khayyam *et al*., 2019; Ma *et al*., 2020), no desenvolvimento de jogos (Skinner; Walmsley, 2019), ou na criação de assistentes virtuais inteligentes como Google Home, Alexa, Siri, dentre outras (Kepuska; Bohouta, 2018).

Na Aprendizagem de Máquina, o modelo de IA utiliza conjuntos de dados para aprender quais os resultados esperados para determinadas entradas. Esse processo é chamado de treinamento. Quanto mais dados adequados, mais ele pode aprender. Um fator crítico para esse treinamento é a quantidade de dados de qualidade disponíveis, que varia em cada caso.

Em alguns casos, a quantidade de dados disponíveis é limitada, como na área da saúde ao trabalhar com doenças raras, onde existem poucos registros de pacientes para uma determinada doença (Abedi *et al*., 2022; Dahmen; Cook, 2019). Em outros casos, essa quantidade é grande, como nos trabalhos relacionados à classificação de imagens, a quantidade de imagens pode chegar a milhões (DENG *et al*., 2009). Nos casos em que a quantidade de dados é grande, existem certos desafios relacionados ao armazenamento e ao processamento desses dados (Chen; Lin, 2014; Khan *et al*., 2014; Pal *et al*., 2020).

Existem técnicas como a compressão e a redução de dimensionalidade que podem mitigar o uso de espaço para armazenar grandes conjuntos de dados, mas são poucas as opções e estas têm suas limitações. Esse artigo busca investigar a possibilidade de trocar o custo de armazenamento de um conjunto de dados por um custo de processamento.

Ao levar em conta que modelos de aprendizagem de máquina como redes

neurais são capazes de estabelecer relações complexas entre as amostras e seus rótulos, nota-se a possibilidade de realizar transformações complexas em conjuntos de dados com o intuito de reduzir o espaço de armazenamento necessário, de forma que estes dados sejam posteriormente recuperados por um modelo de AM.

Tendo um conjunto, obtido pela transformação ou abstração das amostras de um conjunto maior, assim como o modelo que consegue utilizá-lo para produzir as amostras originais, não seria necessário manter o conjunto original armazenado, uma vez que este pode ser obtido ao processar o conjunto abstrato.

Além do espaço de armazenamento, abstrações complexas das amostras também criariam um aumento na segurança destes dados, uma vez que potencialmente dificultariam sua interpretação por humanos que não possuam o modelo capaz de processá-los. Esses modelos poderiam, ainda, ser utilizados em conjunto com técnicas de cache para reduzir o tráfego de informações na rede.

Utilizando o conjunto de flores Iris (Fisher, 1936), esse trabalho realizou experimentos para explorar essa proposta. Foram analisadas duas formas de obter um conjunto abstrato unidimensional a partir das amostras de flores e seus atributos, além de diversas configurações de redes neurais com o objetivo de produzir a amostra original a partir desse conjunto abstrato.

O conjunto Íris foi modificado para incluir as espécies de cada amostra como um novo atributo, de forma que toda essa informação fosse abstraída. Para realizar a abstração do conjunto, foram utilizadas uma técnica baseada na entropia e uma técnica baseada na covariância dos dados.

Os modelos testados foram capazes de produzir valores muito próximos aos dos atributos originais, ainda que não tenham sido capazes de restaurá-los de fato. Ainda assim, ficou claro que existe potencial inexplorado em relação a essa proposta.

Este trabalho foi dividido de forma que, na seção "Trabalhos Relacionados", serão mencionados trabalhos que abordam temas relevantes para essa investigação. Já na seção "Fundamentação Teórica", serão apresentados brevemente os conceitos necessários para compreender este trabalho. Então, na seção "Metodologia" serão descritos os métodos que foram utilizados ao realizar os experimentos apresentados na seção "Configurações dos Experimentos". Finalmente, na seção "Resultados" serão discutidos os resultados desses experimentos, onde o parecer final será dado

na seção "Conclusão".

2 TRABALHOS RELACIONADOS

É amplamente reconhecido que, ao trabalhar com grandes conjuntos de dados, o armazenamento desses dados é um desafio, como apontam Bhadani e Jothimani (2016), Chen e Lin (2014), Khan *et al*. (2014) e Pal *et al*. (2020). Entretanto, o foco de muitas das soluções atuais é no armazenamento ao processar esses dados, não havendo muitas opções para a redução do espaço de armazenamento em si. Dentre as técnicas para mitigar o uso de espaço no armazenamento dos conjuntos de dados, as mais frequentes são a compactação e a redução de dimensionalidade.

Para a compactação, muitas vezes são utilizados algoritmos especializados para conjuntos de dados, como *Context-Tree Weighting* (CTW) ou *Lempel-Ziv* (GAO; Parameswaran, 2016). Já a redução de dimensionalidade consiste em técnicas como a Análise de Componentes Principais (PCA) ou *Laplacian Eigenmap* (LE) (Anowar; Sadaoui; Selim, 2021), que visam extrair as informações (ou relações) que são mais relevantes para um dado problema. Uma limitação da redução de dimensionalidade, no entanto, é que ela resulta em perda de informação.

Na Ciência da Computação, a entropia é frequentemente associada a compressão de dados, em especial ao quanto os dados podem ser comprimidos (Ornstein; Weiss, 1993; Hansel; Perrin; Simon, 1992; Balakrishnan; Touba, 2007). É importante destacar que a entropia é uma medida obtida sobre os dados utilizados, de forma que os valores medidos são relacionados aos dados presentes.

Como esta pesquisa faz uso de redes neurais, fica evidente a necessidade das técnicas para o ajuste de hiperparâmetros (Yang; Shami, 2020; Schratz *et al*., 2019). Essas técnicas são essenciais para a criação de redes neurais capazes de aprender relações entre os dados de entrada e seus rótulos no treinamento. Também tornam possível aprimorar seu desempenho conforme são observadas falhas no treinamento.

Assim, foi percebida uma escassez de técnicas para a redução do espaço de armazenamento. Por tratar da medida da desordem entre dados, também foi notada a possibilidade de utilizar a entropia como uma forma de obter valores que possuem

correlação com um conjunto de dados.

3 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, será apresentada uma revisão das teorias, conceitos e estudos relacionados ao tema em análise. Essa fundamentação é essencial para a compreensão das escolhas e métodos deste trabalho, além da interpretação dos resultados. Vale destacar que os conceitos serão apresentados com foco na utilidade que possuem para essa pesquisa em particular.

3.1 ÁLGEBRA LINEAR

A álgebra linear é a área da Matemática que trabalha com vetores e suas operações, fica claro, portanto, sua importância ao trabalharmos com a manipulação de conjuntos de dados, que são, em sua essência, vetores.

Transformação linear é um conceito da álgebra linear que descreve uma função matemática entre dois espaços vetoriais. Essa função mapeia elementos de um espaço vetorial de origem em elementos de um espaço vetorial de destino, preservando certas propriedades fundamentais (Phillips, 1940). Na Ciência da Computação, é aplicada especialmente na computação gráfica (Wang *et al*., 2017; Dorst; Fontijne; Mann, 2010), na criptografia (Hill, 1931; Firdous; Rehman; Missen, 2019) e na aprendizagem de máquina (Raiko; Valpola; Lecun, 2012).

Outro tipo de operação de vetores, bastante utilizado em Estatística, é a matriz de covariância. Matriz de covariância é uma matriz quadrática e simétrica que descreve a covariância entre os pares de elementos de um vetor (Feller *et al*., 1971). Nesse trabalho, esse conceito será utilizado em conjunto com transformações lineares para a abstração do conjunto íris.

Assim como a matriz de covariância, a entropia é outro conceito utilizado ao trabalhar com as relações entre elementos de um vetor. Entropia pode ser entendida como a medida da desordem em um sistema (Shannon, 1948).

Na Ciência da Computação, a entropia está relacionada à teoria da informação e à compressão de dados. Uma de suas vantagens é a capacidade de identificar

padrões e redundâncias nos dados, permitindo a criação de representações mais compactas dos dados originais.

3.2 INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial é um ramo da Ciência da Computação que surgiu com o objetivo de criar algoritmos capazes de resolver problemas que necessitam da inteligência humana para tomar decisões (Alan, 1950; Došilović; Brčić; Hlupić, 2018). Uma das maneiras utilizadas para obter esses algoritmos complexos é pela Aprendizagem de Máquina.

Aprendizagem de Máquina é um ramo de IA onde modelos são programados para aprender suas próprias regras para tomadas de decisão por meio de algoritmos baseados em matemática e estatística, obtendo conclusões genéricas a partir de conjunto de dados contendo amostras que servem como exemplo.

Além disso, existem diversas classificações para esses modelos, como quanto ao aprendizado, objetivo ou ao algoritmo em si (Escovedo; Koshiyama, 2020; IBM, 2023a). Para casos onde há uma complexidade maior, existe também o aprendizado profundo, caracterizado pela utilização de redes neurais com várias camadas.

Durante o processo de treinamento, os modelos aprendem padrões entre as amostras e seus rótulos analisando os atributos de cada amostra e realizando ajustes de acordo com a taxa de aprendizado e outros fatores.

O aprendizado do modelo pode ser supervisionado – onde um especialista verifica as conclusões e as decisões tomadas pelo modelo para garantir que são o resultado esperado – ou não. O algoritmo pode ser, baseado na entrada e nas regras aprendidas, classificativo, onde, dada uma entrada, tenta determinar a qual das classes conhecidas esta pertence; ou preditivo, onde retorna uma previsão não necessariamente observada antes.

3.3 REDE NEURAL

Redes neurais são um tipo de modelo de aprendizagem de máquina que tem o nome e estrutura inspirados no cérebro humano, imitando as interações entre

neurônios biológicos. São compostas de uma ou mais camadas de neurônios. No aprendizado profundo, costumam ser compostas de uma camada de entrada, várias camadas ocultas, e uma camada de saída (IBM, 2023b). Um neurônio transmite ou não dados para a próxima camada baseado na sua configuração, que por sua vez é ajustada por meio de treinamento.

Além das configurações ajustadas durante o treinamento, existem outras configurações como os hiperparâmetros da rede neural que podem ser ajustados para aumentar a precisão (Passos; Mishra, 2022; Yang; Shami, 2020; Schratz *et al*., 2019). Os hiperparâmetros são valores como o número de camadas, o número de neurônios em cada camada, dentre outros, que determinam como a rede é estruturada e como o algoritmo de aprendizado é executado, afetando o desempenho e o comportamento do modelo de rede neural.

Outra técnica utilizada é a normalização dos dados, transformando os valores para escalas semelhantes, melhorando o desempenho e a estabilidade do treinamento do modelo (Ali *et al*., 2014; Singh; Singh, 2020).

Entretanto, por serem algoritmos complexos e com muitos dos ajustes sendo realizados pelo próprio algoritmo durante o treinamento, existe também o problema da caixa preta: é extremamente difícil determinar quais os motivos que levam às decisões de uma rede neural (Hussain, 2019).

4 METODOLOGIA

A fim de investigar a viabilidade de utilizar redes neurais na redução do espaço necessário para armazenar conjuntos de dados, foram realizados diversos experimentos. Esta seção descreve a abordagem utilizada ao realizar esses experimentos. Aqui será apresentado a base de dados, as fórmulas e funções utilizadas para normalização e abstração dos dados, as etapas dos experimentos e os métodos de análise de resultados utilizados.

Para esse trabalho, foi escolhido o conjunto de dados Íris para ser utilizado como rótulo. Contendo 150 amostras com 4 atributos, existem diversas redes neurais capazes de classificar suas amostras com grande precisão, o que demonstra que redes neurais têm boa capacidade de classificar as amostras deste conjunto. As

amostras utilizadas como entrada são compostas de quatro atributos numéricos:

1. comprimento das sépalas
2. largura das sépalas
3. comprimento das pétalas
4. largura das pétalas

Assim como a entrada, os rótulos também são codificados em valores numéricos, correspondentes a cada espécie de flor no conjunto, como apresentado na Tabela 1.

Tabela 1 – Espécies de flores no Conjunto Iris

espécie	valor numérico	índices das amostras
setosa	0	0 a 49
versicolor	1	50 a 99
virginica	2	100 a 149

Fonte: Autor (2023)

Por terem tipo numérico, os dados são naturalmente compatíveis com os modelos utilizados. Dessa forma, o pré-processamento dos dados pode focar apenas nos ajustes necessários para a investigação, descartando adaptações extras.

Primeiramente, é criado um conjunto modificado com base no conjunto de dados de flores Iris, onde cada amostra recebe um novo atributo contendo o valor referente à sua espécie. Em seguida, é realizada a normalização desse conjunto quando necessário.

Para a normalização dos dados é utilizada a função min-max, que distribui os valores dos atributos para um intervalo $[0,1]$ de acordo com os valores máximo e mínimo presentes nos atributos. Essa função utiliza a fórmula 1, onde x é o valor sendo normalizado, x_{min} e x_{max} são, respectivamente, os valores mínimo e máximo possíveis:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

A partir do conjunto modificado, é realizada a abstração dos dados de acordo

com as especificações do experimento. Se o modelo for capaz de restaurar o conjunto de dados a partir dessa abstração, esse será o único conjunto necessário para armazenar ao final do processo.

Por ser a medida de dispersão dos valores, a entropia do conjunto de dados pode ser aplicada para obter um novo vetor, onde cada amostra terá alguma relação com o conjunto original. Quando necessário, a entropia de cada amostra x, é obtida ao aplicar a fórmula 2, onde x_i são os valores dos atributos da amostra:

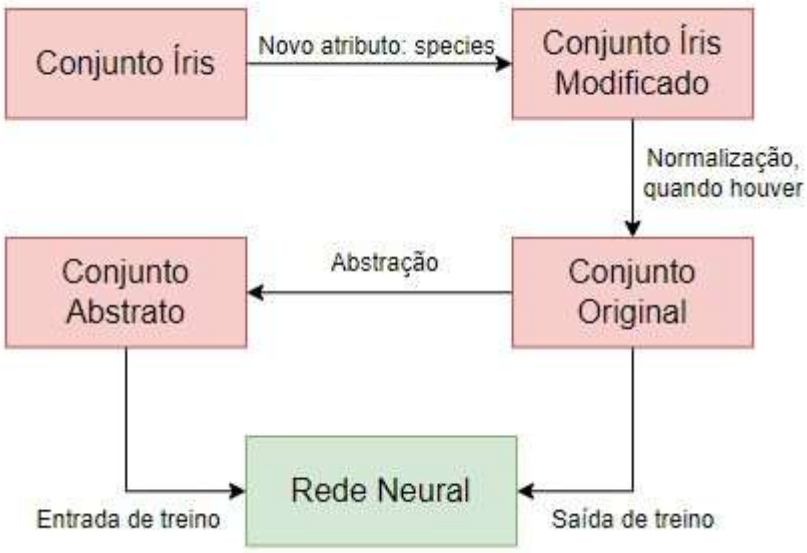
$$s = -\sum \quad x_i \, log_2 \, x_i \tag{2}$$

A outra maneira utilizada para obter um conjunto derivado do original é obter a matriz de covariância das amostras e calcular seus autovalores e autovetores. Então multiplica-se o conjunto de dados original pelo autovetor correspondente ao maior autovalor obtido. Ao final desse processo é gerado um novo vetor contendo 150 valores que possuem relação com o conjunto de dados original, que pode ser utilizado como entrada para a rede neural.

É então realizado o treinamento do modelo, fornecendo uma abstração como entrada e o conjunto modificado como a saída esperada, como mostrado na Figura 1. O treinamento é realizado de forma supervisionada, por se tratar de um problema de classificação (Delua, 2021), onde os parâmetros da rede neural são ajustados arbitrariamente, com o objetivo de maximizar sua precisão.

Comparado a um treinamento tradicional, existem algumas características irregulares sobre o treinamento dessa rede neural. Primeiramente, cada entrada deve essencialmente produzir uma saída única, de forma que não é necessária a divisão do conjunto em amostras para treinamento e teste. Da mesma forma, o *overfitting* não é uma grande preocupação, visto que todas as amostras são conhecidas.

Figura 1 – Visualização do processo realizado para cada experimento



Fonte: Autor (2023)

5 CONFIGURAÇÕES DOS EXPERIMENTOS

A partir das amostras do conjunto íris, foram obtidas abstrações que sejam um novo conjunto unidimensional. Essas abstrações foram obtidas pela transformação linear da matriz de covariância das amostras, ou calculando a entropia das amostras em relação ao conjunto.

A configuração de cada modelo foi ajustada de forma empírica, com valores arbitrários obtidos ao avaliar seu desempenho. Para tentar reduzir o problema de caixa preta, o uso de ferramentas para ajuste de hiperparâmetros nos experimentos foi evitado.

5.1 EXPERIMENTOS

Nos experimentos, foram testadas diversas combinações de derivações do conjunto íris e métodos de abstração. Essas derivações foram criadas de acordo com os seguintes raciocínios:

A. Com o objetivo de abstrair o conjunto de dados por completo, os rótulos e

amostras do conjunto íris foram combinados em um novo conjunto, de forma que o rótulo tornou-se mais um atributo de cada amostra.

B. As amostras que pertencem à mesma espécie possuem maior semelhança, por isso, foram separadas e testadas as amostras de uma única espécie de cada vez, a fim de determinar se essa relação mais próxima afetariam o desempenho da rede neural. Nesses casos, é descartada a espécie como atributo, visto que todas as amostras pertencem à mesma espécie.

Também foi testado se normalizar o conjunto de dados poderia facilitar o treinamento da rede neural, uma vez que os valores de cada atributo teriam menos variação entre eles, uniformizando o ajuste dos pesos. A Tabela 2 contém as configurações dos experimentos iniciais realizados, além dos códigos que serão utilizados ao se referir a cada experimento.

Tabela 2 – Configurações iniciais dos experimentos

código	conjunto inicial	espécie	normalizado	abstração
E1	A	-	não	entropia
E2	A	-	sim	entropia
E3	A	-	não	transformação linear
E4	A	-	sim	transformação linear
E5	B	setosa	não	entropia
E6	B	setosa	sim	entropia
E7	B	setosa	não	transformação linear
E8	B	setosa	sim	transformação linear
E9	B	versicolor	não	entropia
E10	B	versicolor	sim	entropia
E11	B	versicolor	não	transformação linear
E12	B	versicolor	sim	transformação linear
E13	B	virginica	não	entropia
E14	B	virginica	sim	entropia
E15	B	virginica	não	transformação linear
E16	B	virginica	sim	transformação linear

Fonte: Autor (2023)

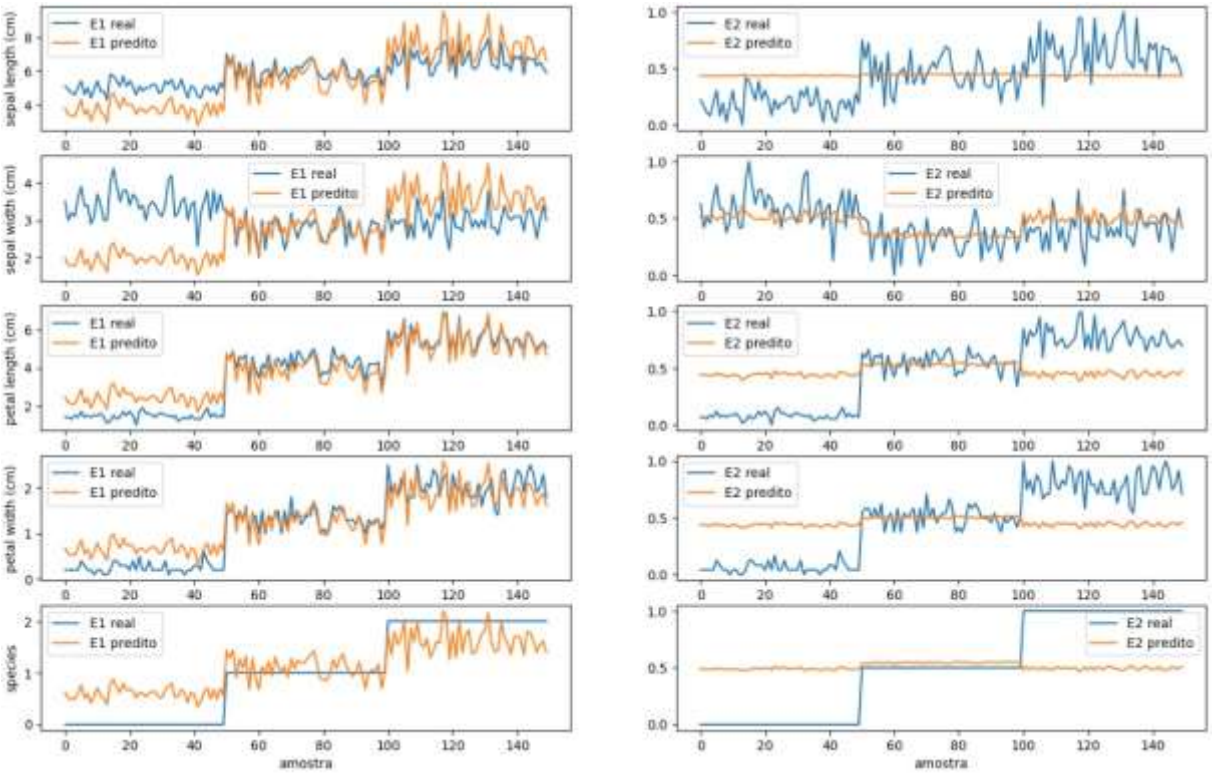
Inicialmente, o modelo foi configurado arbitrariamente com 3 camadas de 64 neurônios cada, mais uma camada de saída, onde a camada de entrada e a camada de saída utilizaram a função de ativação *linear*, e as camada intermediária utilizaram a função *relu*. Os treinamentos iniciais utilizaram 100 épocas. Visando recuperar o máximo possível de informação diretamente da rede neural, foi evitado o pós-processamento dos dados.

6 RESULTADOS

Após treinar os modelos referentes aos experimentos na Tabela 2, foram fornecidos os conjuntos de entrada para a realização das predições, e em seguida analisada a proximidade entre os valores obtidos e os valores esperados.

Ao comparar os modelos que não foram normalizados com os que foram, constatou-se que, ao contrário do esperado, os experimentos onde foi realizada a normalização dos valores tiveram maior discrepância entre os valores reais e os valores preditos. A figura 2 mostra a diferença entre os resultados iniciais de E1 e E2.

Figura 2 – Resultados iniciais de E1 e E2

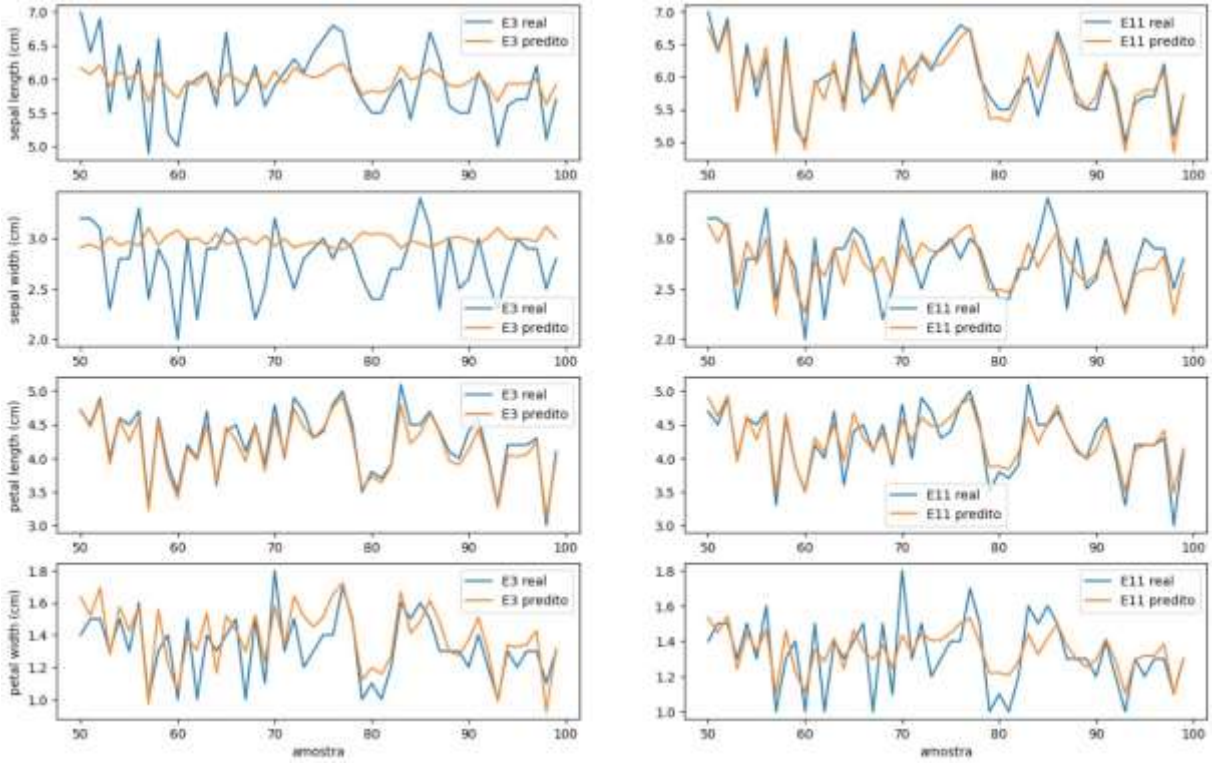


Fonte: Autor (2023)

É possível que essa discrepância seja pela necessidade de uma configuração diferente que seja específica para os modelos que operam dados normalizados, no entanto, dada a proximidade dos valores sem normalização com os valores esperados, foi entendido que a normalização poderia ser descartada.

Como esperado, ao comparar os modelos treinados com o conjunto completo – como E1 e E3 – aos modelos especializados em uma única espécie, os modelos especializados obtiveram previsões mais consistentes e próximas dos valores reais. Um exemplo disso pode ser observado na Figura 3, que compara os resultados iniciais de E3 e E11.

Figura 3 – Resultados iniciais de E3 (apenas amostras de versicolor) e E11



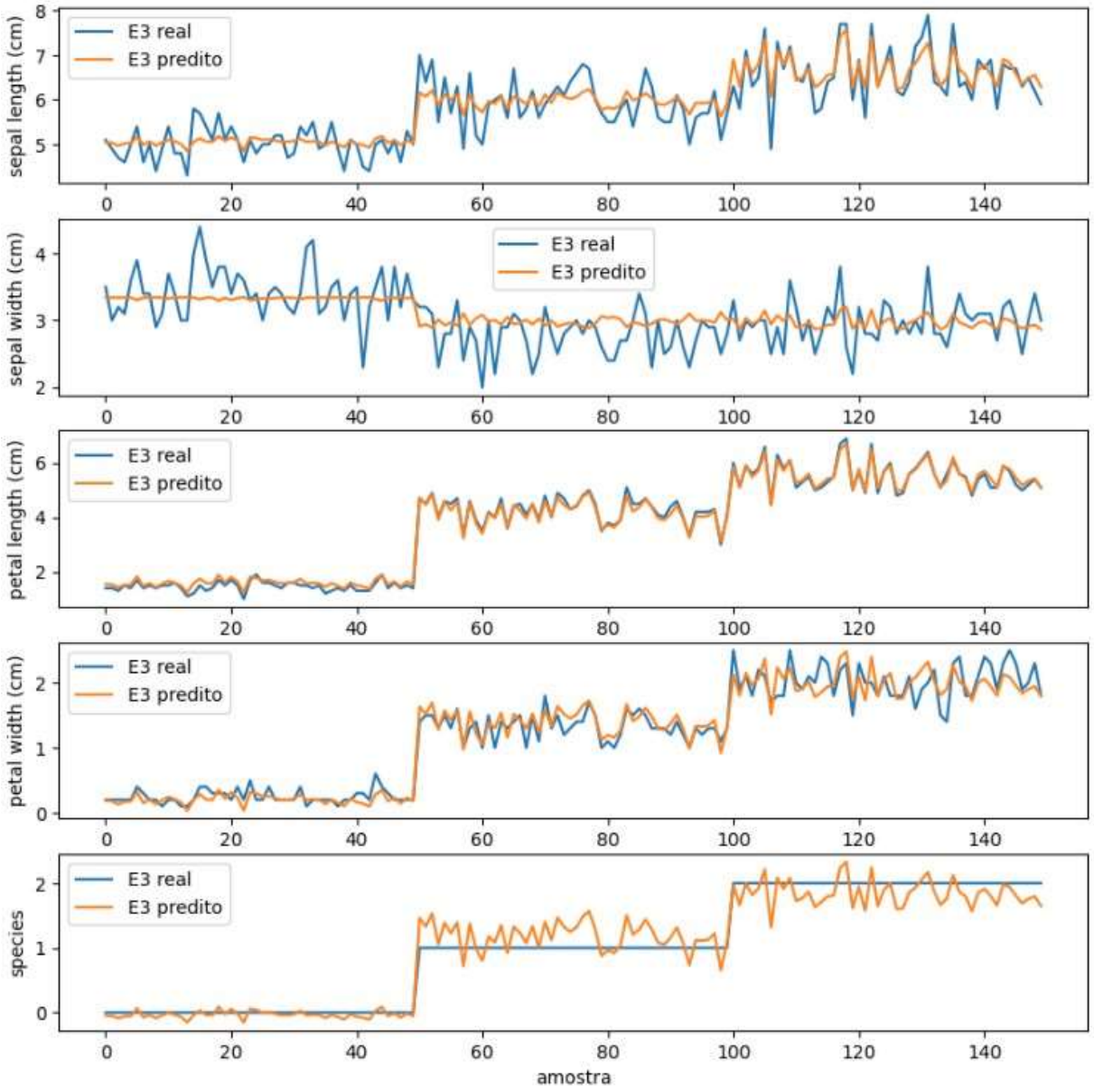
Fonte: Autor (2023)

Dado o objetivo de abstrair o conjunto completo, é evidente o interesse em especial sobre os experimentos E1, E2, E3 e E4. Ao descartarmos os experimentos E2 e E4, que ambos fazem uso de normalização, é natural focar a análise sobre os modelos E1 e E3.

O experimento E3 obteve valores especialmente próximos para o

comprimento das pétalas de cada espécie, além do valor da espécie em si, conforme pode ser observado na Figura 4. No entanto, os valores para a largura das sépalas foram essencialmente perdidos.

Figura 4 – Resultados iniciais de E3 para todas as espécies



Fonte: Autor (2023)

O experimento E1 obteve valores mais próximos que E3 para a largura das sépalas das espécies versicolor e virginica, ainda que os valores preditos para as amostras de setosas em particular também tenham tido grande discrepância dos

valores reais.

Ambos os experimentos E1 e E3 apresentaram maior discrepância nos valores obtidos para a espécie setosa, especialmente no valor obtido para a largura das sépalas.

Por fim, ao realizar diversos ajustes de hiperparâmetros no modelo E3, chegou-se ao modelo mostrado na Figura 5, com o qual foi possível obter valores mais significativos para a largura das sépalas, ainda que sejam muito diferentes dos valores reais. Também foi possível aproximar ainda mais os valores obtidos para o comprimento e largura das pétalas, e para o valor das espécies, com destaque para o valor da espécie setosa. Os resultados desse modelo podem ser observados na figura 6.

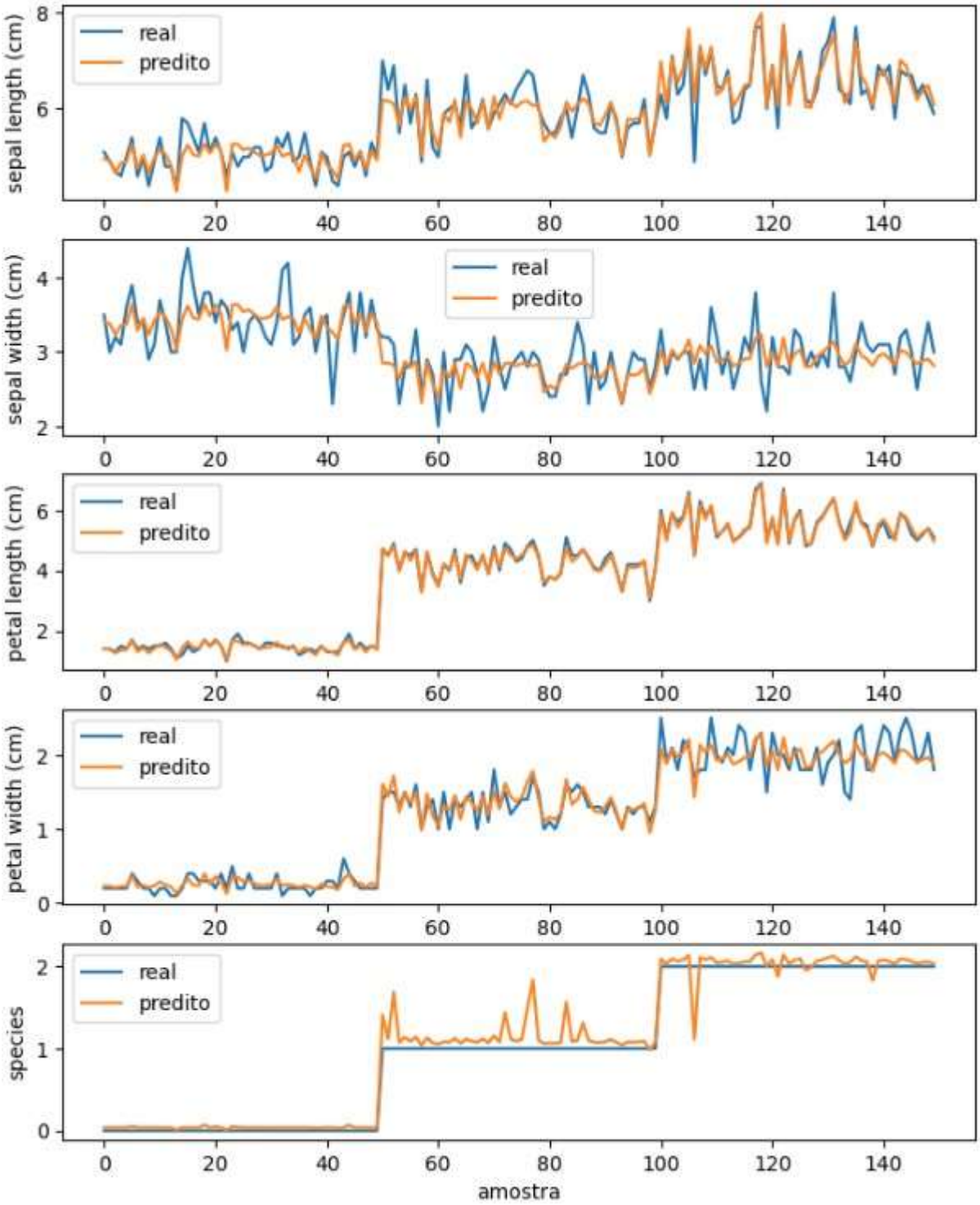
Figura 5 – Visão geral do modelo E3 final

Layer (type)	Output Shape	Param #
dense_68 (Dense)	(None, 256)	512
dense_69 (Dense)	(None, 512)	131584
dense_70 (Dense)	(None, 64)	32832
dense_71 (Dense)	(None, 5)	325

Total params: 165253 (645.52 KB)
Trainable params: 165253 (645.52 KB)
Non-trainable params: 0 (0.00 Byte)

Fonte: Autor (2023)

Figura 6 – Resultados finais de E3



Fonte: Autor (2023)

7 CONCLUSÃO

Fica evidente, portanto, que utilizando os modelos descritos neste trabalho, com os parâmetros apresentados, não foi possível restaurar os valores originais, ainda que os valores obtidos tenham sido muito próximos dos reais em alguns casos.

Dadas as limitações como não utilizar ferramentas especializadas para o ajuste de hiperparâmetros, não realizar pós-processamento nos valores preditos para aproximá-los dos reais, além dos métodos simples utilizados para a criação das abstrações do conjunto real, esse trabalho deve ser considerado um estudo inicial.

Assim, os resultados obtidos neste trabalho não devem ser interpretados como conclusivos sobre o conceito em si, que ainda possui muitas possibilidades inexploradas, como outras abordagens ou o aprimoramento dos métodos utilizados. Conforme mencionado anteriormente, as vantagens em termos de armazenamento e segurança, além da proximidade atingida, são grandes motivações para maiores explorações da proposta deste artigo.

Dentre os aprimoramentos possíveis de serem explorados, podem ser destacados a experimentação de outras funções para a abstração e para a normalização, as ferramentas para ajuste de hiper-parâmetro, a utilização de pós-processamento, além da aplicação em conjuntos de dados mais extensos ou mais complexos.

REFERÊNCIAS

ABEDI, M. *et al*. Gan-based approaches for generating structured data in the medical domain. **Applied sciences (Basel, Switzerland)**, v. 12, p. 7075, 2022.

ALAN, M. Turing. **Computing machinery and intelligence. Mind**, v. 59, n. 236, p. 433–460, 1950.

ALI, P. J. M. *et al*. Data normalization and standardization: a technical report. **Mach Learn Tech Rep**, v. 1, n. 1, p. 1–6, 2014.

ANOWAR, F.; SADAOUI, S.; SELIM, B. Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). **Computer Science Review**, Elsevier, v. 40, p. 100378, 2021.

BALAKRISHNAN, K. J.; TOUBA, N. A. Relationship between entropy and test data

compression. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 26, n. 2, p. 386–395, 2007.

BHADANI, A. K.; JOTHIMANI, D. **Big data: Challenges, opportunities, and realities.** 2016.

CHEN, X.-W.; LIN, X. Big data deep learning: Challenges and perspectives. **IEEE Access**, v. 2, p. 514–525, 2014.

DAHMEN, J.; COOK, D. Synsys: A synthetic data generation system for healthcare applications. **Sensors (Basel, Switzerland)**, v. 19, 2019. ISSN 14248220.

DELUA, J. **Supervised vs. Unsupervised learning: What's the difference?** 2021. Disponível em: https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning. Acesso em: 19 mar. 2023.

DENG, J. *et al*. ImageNet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition.** [S.l.]: IEEE, 2009.

DORST, L.; FONTIJNE, D.; MANN, S. **Geometric algebra for computer science: an object-oriented approach to geometry.** [S.l.]: Elsevier, 2010.

DOŠILOVIĆ, F. K.; BRČIĆ, M.; HLUPIĆ, N. Explainable artificial intelligence: A survey. In: IEEE. **2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO).** [S.l.], 2018. p. 0210–0215.

ESCOVEDO, T.; KOSHIYAMA, A. **Introdução a Data Science: Algoritmos de Machine Learning e métodos de análise.** [S.l.]: Casa do Código, 2020.

FELLER, W. *et al*. An introduction to probability theory and its applications. New York: John Wiley, 1971.

FIRDOUS, A.; REHMAN, A. ur; MISSEN, M. M. S. A highly efficient color image encryption based on linear transformation using chaos theory and sha-2. **Multimedia Tools and Applications**, Springer, v. 78, p. 24809–24835, 2019.

FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of Eugenics**, v. 7, 1936.

GAO, Y.; PARAMESWARAN, A. Squish: Near-optimal compression for archival of relational datasets. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.** [S.l.: s.n.], 2016. p. 1575–1584.

HANSEL, G.; PERRIN, D.; SIMON, I. Compression and entropy. In: SPRINGER. **Annual Symposium on Theoretical Aspects of Computer Science.** [S.l.], 1992. p. 513–528.

HILL, L. S. Concerning certain linear transformation apparatus of cryptography. **The American Mathematical Monthly**, Taylor & Francis, v. 38, n. 3, p. 135–154, 1931.

HUSSAIN, J. **Deep Learning Black Box Problem**. 2019.

IBM. **What is machine learning**. 2023. Disponível em: https://www.ibm.com/topics/machine-learning. Acesso em: 23 out. 2023.

IBM. **What is neural network**. 2023. Disponível em: https://www.ibm.com/topics/neural-networks. Acesso em: 23 out. 2023.

KEPUSKA, V.; BOHOUTA, G. Next-generation of virtual personal assistants (Microsoft cortana, apple siri, amazon alexa and google home). [S.l.: s.n.], 2018. v. 2018-January.

KHAN, N. *et al*. Big data: survey, technologies, opportunities, and challenges. **The scientific world journal**, Hindawi, v. 2014, 2014.

KHAYYAM, H. *et al*. **Artificial Intelligence and Internet of Things for Autonomous Vehicles**. 2019.

MA, Y. *et al*. Artificial intelligence applications in the development of autonomous vehicles: A survey. **IEEE/CAA Journal of Automatica Sinica**, v. 7, 2020.

ORNSTEIN, D.; WEISS, B. Entropy and data compression schemes. **IEEE Transactions on Information Theory**, v. 39, n. 1, p. 78–83, 1993.

PAL, S. *et al*. Big data in biology: The hope and present-day challenges in it. **Gene Reports**, Elsevier, v. 21, p. 100869, 2020.

PASSOS, D.; MISHRA, P. A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks. **Chemometrics and Intelligent Laboratory Systems**, Elsevier, v. 223, p. 104520, 2022.

PHILLIPS, R. S. On linear transformations. **Transactions of the American Mathematical Society**, v. 48, n. 3, p. 516–541, 1940.

RAIKO, T.; VALPOLA, H.; LECUN, Y. Deep learning made easier by linear transformations in perceptrons. In: PMLR. **Artificial intelligence and statistics**. [S.l.], 2012. p. 924–932.

SCHRATZ, P. *et al*. Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. **Ecological Modelling**, Elsevier, v. 406, p. 109–120, 2019.

SHANNON, C. E. A mathematical theory of communication. **The Bell system technical journal**, Nokia Bell Labs, v. 27, n. 3, p. 379–423, 1948.

SINGH, D.; SINGH, B. Investigating the impact of data normalization on classification performance. **Applied Soft Computing**, Elsevier, v. 97, p. 105524, 2020.

SKINNER, G.; WALMSLEY, T. Artificial intelligence and deep learning in video games a brief review. [S.l.: s.n.], 2019.

WANG, W. *et al*. Fast image dehazing method based on linear transformation. **IEEE Transactions on Multimedia**, IEEE, v. 19, n. 6, p. 1142–1155, 2017.

YANG, L.; SHAMI, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. **Neurocomputing**, Elsevier, v. 415, p. 295–316, 2020.