UNIVERSIDADE FEDERAL DE SERGIPE

PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Agregação de Variáveis nas Redes Neurais Artificiais Baseada em Medidas de Informação Mútua

Gabriel Francisco Alves Bastos

São Cristóvão – SE, Brasil

Agosto de 2026

Agregação de Variáveis nas Redes Neurais Artificiais Baseada em Medidas de Informação Mútua

Gabriel Francisco Alves Bastos

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica – PROEE, da Universidade Federal de Sergipe, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica

Orientador: Jugurta Rosa Montalvão Filho

São Cristóvão – SE, Brasil

Agosto de 2026

UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA-PROEE

TERMO DE APROVAÇÃO

"AGREGAÇÃO DE VARIÁVEIS NAS REDES NEURAIS ARTIFICIAIS BASEADA EM MEDIDAS DE INFORMAÇÃO MÚTUA"

Discente:

Gabriel Francisco Alves Bastos

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Sergipe, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Aprovada pela banca examinadora composta por:

Documento assinado digitalmente
gov.br EDUARDO OLIVEIRA FREIRE
Data: 26/08/2026 11:28:42-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Eduardo Oliveira Freire (PROEE/UFS)
Presidente

Documento assinado digitalmente
gov.br LEONARDO NOGUEIRA MATOS
Data: 26/08/2026 15:35:11-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Leonardo Nogueira Matos (PROCC/UFS)
Examinador Externo

Documento assinado digitalmente
gov.br JANIO COUTINHO CANUTO
Data: 26/08/2026 06:10:17-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Janio Coutinho Canuto (DCOMP/UFS)
Examinador Externo

Documento assinado digitalmente
gov.br ALEXANDRE LUIS MAGALHAES LEVADA
Data: 24/08/2026 09:29:48-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Alexandre Luís Magalhães Levada (PPGCC/UFSCar)
Examinador Externo

Documento assinado digitalmente
gov.br GABRIEL FRANCISCO ALVES BASTOS
Data: 27/08/2026 11:54:20-0300
Verifique em https://validar.iti.gov.br

Gabriel Francisco Alves Bastos
Discente

Cidade Universitária "Prof. José Aloísio de Campos", 18 de agosto de 2026.

FICHA CATALOGRÁFICA ELABORADA PELO SIBIUFS
UNIVERSIDADE FEDERAL DE SERGIPE

Bastos, Gabriel Francisco Alves
B327a Agregação de variáveis nas redes neurais artificiais baseada em medidas de informação mútua / Gabriel Francisco Alves Bastos ; orientador Jugurta Rosa Montalvão Filho. – São Cristóvão, SE, 2026.
115 f. ; il.

Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Sergipe, 2026.

1. Engenharia elétrica. 2. Redes neurais (Computação). 3. Inteligência artificial. 4. Análise de dados. 5. Estatística. I. Montalvão Filho, Jugurta Rosa, orient. II. Título.

CDU 621.3:007.52

Agradecimentos

Confesso que nunca soube direito como me expressar em uma seção de agradecimentos. Mas, seria injusto da minha parte não usar essa plataforma para expressar o tamanho da gratidão que eu sinto. Depois de sete anos na UFS, posso afirmar que tive a sorte de encontrar uma profissão que eu genuinamente amo, e me tornar mestre é mais um passo para exercê-la. Isso é motivo de muita gratidão. Mais gratidão ainda tenho pelas pessoas que me ajudaram a chegar aqui, e pelas que eu ganhei no meio do caminho.

Falando em gratidão, então, só posso começar pela minha mãe. Ela foi o melhor exemplo que eu poderia ter, para a vida. Eu escreveria por horas sobre o quão bela é a trajetória dela, e ainda assim seria insuficiente. Ela é a minha maior demonstração do quanto o esforço e a luta pela educação transforma vidas, como mãe e também como professora. Sem o esforço imensurável, o apoio incondicional, e a luta diária dela, nada disso seria possível. Falando em tudo isso, não posso deixar de mencionar os meus avós maternos, Luiz e Marizete. Palavras não são suficiente para explicar o tamanho do amor que eles derramaram em mim, desde que nasci. Palavras também falham em explicar o quão guerreiros eles foram, ao criar 9 filhos, em condições adversas, e ver todos eles se tornarem homens e mulheres admiráveis. Falando nisso, eu não posso deixar de expressar minha gratidão pelos meus tios e tias. Eles sempre foram para mim tudo que eu precisei, e muito mais. Esse núcleo familiar é o melhor que eu poderia ter, e eu sempre vou me referir a ele com profundo orgulho.

Academicamente, fui igualmente sortudo. A Jugurta Montalvão, orientador deste trabalho: muito obrigado. Quando eu entrei para o Biochaves, em 2022, eu achei que ganharia um orientador de IC, que faria alguns trabalhos sob a orientação dele, e que isso faria bem para a minha formação. Eu não imaginava o que viria depois. Falar sobre o brilhantismo dele como cientista, professor e orientador é redundância, afinal todo mundo já sabe. Mas tudo bem, no primeiro texto que o enviei, em 2023, ele me disse que apreciava redundâncias, e eu aprendi a apreciar também. Eu me sinto muito honrado em continuar minha formação para um dia ser um cientista, mas isso é pouco comparado ao orgulho que sinto por ter aprendido o que aprendi sobre isso com ele.

Agradeço também ao professor Lucas Molina e ao Professor Elyson. Foram os professores com quem mais fiz disciplinas na graduação e na pós graduação, e eu os admiro profundamente. Admiro o amor que eles sentem pela educação, a forma como eles são competentes e sérios para caramba, e ao mesmo tempo engraçados e divertidos para caramba, e admiro a capacidade que eles sempre tiveram de se adaptar ao que o departamento precisa. Agradeço também por terem ajudado a construir esse ambiente

incrível, como alunos e depois como professores.

Dentro desse ambiente incrível, também tive a oportunidade de conhecer um amigo muito especial, Pedro Henrique a.k.a. Pedin. Ele foi um anjo na minha vida desde 2023, e sempre teve paciência comigo para me explicar as nuances da vida, mesmo eu sendo um tabaréu. Algumas pessoas acham curioso a gente ser tão amigo, por parecermos muito diferentes. Sabem de nada...

Amigos especiais, aliás, não faltaram. Então, preciso fazer um grande agradecimento ao meus amigos do grupo de instrumentação e do GPR. A lista de nomes aqui seria muito extensa, mas eu só tenho a agradecer a todos eles. Conviver com essas pessoas na UFS foi incrível. As tardes, as madrugadas, as aulas, as carregadas no vôlei, os rolês depois do vôlei, as maluquices no ferreirinho. Tudo isso foi com eles, e eu não vou esquecer nunca.

Uma das coisas mais legais que eu tive a oportunidade de fazer na UFS foi desenvolver trabalhos acadêmicos com amigos de verdade. Então, agradeço muito a Kekinho, Cone, David e Ataíde por isso. Eles são amigos muito especiais, e trabalhar com eles nunca pareceu trabalho.

Agradeço também aos meus amigos do Biochaves, grupo do qual faço parte, e continuarei fazendo de alguma forma. Em particular, quero agradecer a Israel, que já estava fora do país quando eu entrei no grupo, e, mesmo assim, se tornou um grande amigo e sempre me ajudou como uma pessoa muito especial, sou muito grato ao que aprendi com ele. Agradeço também a Vitor Magno, por ter sido um amigo tão gentil desde que entrei no grupo. Preciso fazer um agradecimento especial a Thiago Rocha, que sempre mostrou muito interesse quando eu falava sobre qualquer trabalho que eu estava fazendo. Me dói que ele não esteja mais conosco, e que no dia que eu defender esse trabalho, ele não estará lá para assistir minha defesa. Isso nunca fará sentido para mim.

Durante o mestrado eu também tive a experiência de lecionar a disciplina de reconhecimento de padrões, por um período. Assim, não poderia deixar de agradecer aos alunos da turma, que fizeram tudo ser muito melhor do que eu estava esperando, e olha que eu já estava muito feliz mesmo antes de começar.

Dedico esse trabalho ao sistema de ensino público brasileiro, e a todas as pessoas, que por algum motivo, não tiveram a mesma oportunidade que eu tive, de usufruir de uma formação de qualidade.

Resumo da Dissertação apresentada ao PROEE/UFS como parte dos requisitos necessários para a obtenção do grau de Mestre (Me.)

Agregação de Variáveis nas Redes Neurais Artificiais Baseada em Medidas de Informação Mútua

Gabriel Francisco Alves Bastos

Agosto/2026

Orientador: Jugurta Rosa Montalvão Filho

As redes neurais artificiais constituem uma classe de modelos que estão no centro do aprendizado de máquina moderno. Uma característica notável desses modelos, em muitos dos casos, é a sua grande quantidade de parâmetros, que acompanha a necessidade por regularizações, implícitas ou explícitas, para viabilizar a obtenção de resultados expressivos. Uma forma de embutir regularizações implícitas nos modelos é restringindo quais variáveis podem interagir ao longo de neurônios artificiais, a exemplo do que acontece nas camadas de convolução e *pooling* das redes neurais convolucionais (*convolutional neural networks*, CNNs), em que cada variável só interage com os seus vizinhos espaciais. Dentro desse contexto, a hipótese que motiva este trabalho é a de que, em algumas situações, é benéfico que esse tipo de restrição seja imposta de modo que as computações aplicadas em subconjuntos das variáveis de entrada envolvam aquelas que compartilham alta dependência estatística. Nesse sentido, um resultado relevante dessa dissertação é uma prova de conceito que evidenciam as CNNs, conforme definidas convencionalmente, como um exemplo prático da aplicação desse pressuposto. Adicionalmente, são coletadas evidências experimentais de que a extrapolação desse princípio para dados tabulares, através do uso de camadas localmente conectadas, é uma abordagem promissora para o aprendizado de representações e autocodificação de padrões. Ao longo do trabalho, faz-se necessário, por muitas vezes, quantificar a dependência estatística entre pares de variáveis, através de medidas de informação mútua. Assim, este documento também inclui um estudo aprofundado sobre o problema de estimação de informação mútua, que consiste em um problema ativo de pesquisa. Os resultados desse estudo, portanto, incluem a proposição de um novo método de estimação para variáveis discretas, e a adoção de um estimador de informação mútua entre fontes contínuas para ser usado ao longo do trabalho.

Palavras-chaves: dependência estatística, informação mútua, redes neurais, localidade, *autoencoders*, redes convolutivas.

Abstract of Dissertation presented to PROEE/UFS as a partial fulfillment of the requirements for the degree of Master

Aggregation of Variables in Artificial Neural Networks Based on Mutual Information Measures

Gabriel Francisco Alves Bastos

Agosto/2026

Advisor: Jugurta Rosa Montalvão Filho

Artificial neural networks constitute a class of models that lies at the core of modern machine learning. A notable attribute of these models is, in many applications, their large number of parameters, which gives rise to the need for either implicit or explicit regularization in order to achieve strong predictive performance. One way to incorporate implicit regularization is by constraining which variables are allowed to interact within artificial neurons, as exemplified by the convolutional and the pooling layers of convolutional neural networks (CNNs), where each variable interacts only with its spatial neighbors. Within this context, the central hypothesis motivating this dissertation is that, in certain scenarios, it is beneficial to impose such constraints so that computations performed over subsets of the input variables involve those exhibiting high statistical dependence. In this regard, one of the main contributions of this dissertation is a proof of concept demonstrating that conventionally defined CNNs provide a practical example of the application of this principle. Furthermore, experimental evidence is presented showing that extending this principle to tabular data through the use of locally connected layers is a promising approach for representation learning and autoencoding. Throughout the dissertation, it is frequently necessary to quantify the statistical dependence between pairs of variables using mutual information measures. Accordingly, this document also presents an in-depth study of the mutual information estimation problem, which remains an active research hotspot. As a result, the outcomes of this study include the proposal of a novel estimation method for discrete variables and the adoption of a mutual information estimator for continuous sources, which is employed throughout the dissertation.

Keywords: statistical dependence, mutual information, neural networks, locality, autoencoders, convolutional networks.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

As redes neurais artificiais (RNAs) remetem a uma classe de modelos que se conhece há muitas décadas, e que passaram por diversos momentos marcados por pouco interesse prático e científico ao longo de sua história [1]. Em contrapartida, as últimas décadas foram marcadas por um notável crescimento de interesse por parte de pesquisadores e praticantes, à medida que esses modelos revolucionaram o aprendizado de máquina moderno e se tornaram a principal engrenagem das tecnologias que hoje são rotuladas como inteligência artificial.

Do ponto de vista acadêmico, diversas novas arquiteturas são publicadas em um ritmo vertiginoso e clamam desempenhos correspondes ao estado da arte em inúmeras aplicações. Em contrapartida, esse sucesso prático não é acompanhado de um entendimento pleno acerca dos mecanismos internos dessas arquiteturas e de explicações consolidadas para o sucesso delas, uma vez que o aprendizado profundo parece contradizer muitas das expectativas derivadas da teoria clássica do aprendizado [2, 3]. Como resultado, uma evidente limitação na capacidade de propor e melhorar modelos de maneira fundamentada decorre desse entendimento ainda opaco acerca do funcionamento interno das RNAs, fazendo com que muitos dos desenvolvimentos mencionados acima recorram a combinações pouco fundamentadas de componentes já conhecidos e processos empíricos de tentativa e erro.

Como é de esperar, reduzir essa lacuna se tornou um dos objetivos mais relevantes da comunidade científica ao longo dos últimos anos, e, embora o avanço empírico baseado em desempenhos ainda aparente ser muito mais volumoso, diversos grupos têm proposto visões diferentes e complementares acerca do problema [4, 5, 6, 7, 8].

Para esse trabalho de mestrado, entretanto, a principal motivação não está nas teorias mais abrangentes que buscam justificar o aprendizado profundo. Uma descrição mais fiel para o que motiva as investigações aqui apresentadas, e que deve se tornar evidente ao longo do texto, está no estudo de mecanismos internos das redes neurais artificiais, e em entender como estes podem ser manipulados para generalizar determinados princípios de maneira fundamentada.

Em particular, o mecanismo estudado ao longo desta dissertação é um dos mais básicos e destacáveis do aprendizado profundo, e em um certo sentido, pouco questionado: a etapa de agregação de variáveis. Para o propósito deste trabalho, define-se a agregação de variáveis como uma função que recebe como entrada um vetor d-dimensional real e o mapeia em um único escalar real. Em outras palavras, trata-se de uma função que agrega d variáveis e as resumem a uma única variável de saída.

Seu caráter destacável se dá pelo fato de que as RNAs podem ser interpretadas como modelos que são capazes de aproximar funções complexas através da composição de inúmeras unidades de agregação de variáveis. De fato, essas unidades são inspiradas no modelo do neurônio artificial [1], que executam agregações da forma

$$y = \varphi\left(\sum_{i=1}^{d} w_i x_i + b\right), \tag{1.1}$$

em que $x_1, \ldots, x_d$ são as d variáveis de entrada a serem agregadas, $w_1, \ldots, w_d$ e b são parâmetros ajustáveis, enquanto $\varphi(\cdot)$ representa a aplicação de uma não linearidade, conhecida como função de ativação. A ideia mais básica das RNAs, que permeia a grande maioria dos modelos bem sucedidos, é a de que unidades tão simples quanto essas, que combinam uma mera agregação linear de variáveis com uma não linearidade, podem se conectar ao longo de múltiplas camadas e aproximar funções arbitrariamente complexas.

A respeito da etapa de agregação de variáveis, alguns trabalhos recentes consideram sua importância, ao passo que questionam um pressuposto que por muito tempo se manteve intacto: o de que a agregação dentro de cada uma das unidades básicas deve seguir o formato da Equação 1.1. Nesses estudos, novas formas de agregação a serem acopladas nas RNAs foram propostas [9, 10, 11].

Um outro questionamento, mais condizente com o que é estudado ao longo desta dissertação, remete a quais variáveis devem ser agregadas dentro de cada uma das unidades básicas das RNAs. Por exemplo, no formato mais primitivo de aprendizado profundo temos as redes do tipo *multilayer perceptron* (MLP), nas quais todas as unidades promovem agregações entre todas as variáveis de saída da camada anterior [12]. Nas camadas convolucionais das CNNs (*convolutional neural networks*), por outro lado, a agregação de variáveis produzida em cada uma das unidades é restrita a um subconjunto fixo das variáveis de saída da camada anterior, através das noções de vizinhança e localidade inerente à sinais definidos em grade (grid), a exemplo de imagens, séries temporais e vídeos [13]. Indo além, outros tipos de restrições podem ser identificadas se analisarmos, sob essa perspectiva de agregação de variáveis, as unidades constituintes de outros tipos de arquiteturas, como as redes neurais recorrentes (RNNs) [14], as LSTMs (*Long short-term memory*) [15], e os *Transformers* [16].

Nesse contexto, o tema central do estudo proposto neste trabalho de mestrado é a imposição de restrições como essas nas agregações de variáveis que compõem os modelos de RNAs, através de medidas de dependência estatística entre variáveis. Especificamente, o objetivo central deste trabalho é investigar a hipótese enunciada abaixo.

(**H1**) A imposição de restrições nos modelos de RNAs de modo que suas unidades básicas, ao longo de múltiplas camadas, agreguem apenas variáveis que compartilhem

alta dependência estatística, sob determinadas condições, pode produzir ganhos de acurácia, robustez e eficiência nos modelos.

Algumas das motivações que motivaram a formulação dessa hipótese estão listadas abaixo (de maneira não exaustiva):

(**M1**) Restrições acerca de quais variáveis devem ser agregadas em cada uma das unidades naturalmente reduzem, potencialmente de maneira drástica, a quantidade de parâmetros dos modelos, o que por sua vez engendra uma maior eficiência estatística e uma maior capacidade de generalização, ao mesmo tempo que, se as restrições forem pertinentes — i.e., consistentes com o fenômeno sendo modelado —, elas preservam o poder expressivo necessário para obter bons resultados, como conhecidamente acontece nas CNNs.

(**M2**) Um pressuposto amplamente aceito pela comunidade do aprendizado de máquina é o de que padrões multivariados encontrados em grande parte das bases de dados de interesse prático ocupam, aproximadamente, estruturas geométricas, de dimensão mais baixa do que a dimensão nominal, que são comumente chamadas de *manifolds*. Isso sugere que as representações em altas dimensões são inerentemente redundantes, devido às fortes relações de dependências entre as variáveis que compõem esses padrões. Tendo em vista os desafios associados ao processamento de dados em altas dimensões, parece ser um consenso entre pesquisadores da área que projetar as observações multivariadas em espaços latentes de baixa dimensão com variáveis idealmente independentes (i.e., não redundantes) é um objetivo relevante [17, 18]. Dessa forma, impor a agregação entre variáveis dependentes parece ser uma forma de induzir os modelos a explorarem essas redundâncias e obterem tais representações latentes de maneira implícita.

(**M3**) A ideia de agregar subconjuntos de variáveis dependentes já é implicitamente explorada em outros algoritmos comuns ao aprendizado de máquina. Por exemplo, na análise de componentes principais (*principal component analysis*, PCA), grupos de variáveis correlacionadas passam por agregações lineares, sendo reduzidos a um número menor de coeficientes que preserva a maior parte da variância existente nas variáveis originais. Analogamente, a análise de componentes independentes (*independent component analysis*, ICA), por sua vez, remete a uma classe de algoritmos que dão um passo à frente, em relação ao PCA. Enquanto o PCA busca representar os dados em um sistema de coordenadas no qual as variáveis são descorrelacionadas, os algoritmos de ICA buscam fazer o mesmo em um sistema no qual as variáveis são estatisticamente independentes [19]. Ambos exploram agregações entre variáveis dependentes, com a diferença que o PCA só é sensível às dependências do tipo linear,

enquanto o ICA busca explorar quaisquer tipos de dependências. Assim, estudar as propriedades da agregação entre variáveis dependentes nas redes neurais artificiais surge como um objetivo relevante, dada a sua importância em ferramentas clássicas e consolidadas.

Para realizar essas investigações, uma etapa fundamental é a de quantificar a dependência estatística entre todos os pares de variáveis em um determinado conjunto de dados, a fim de mapear quais subconjuntos de variáveis devem interagir e serem agregados ao longo dos neurônios artificiais. Essa etapa se assemelha ao que acontece no PCA, em que o primeiro passo consiste na determinação da matriz de covariância amostral. Para este trabalho, no entanto, propõe-se trabalhar com dependências em um sentido mais amplo, em contraste às medidas de covariância, que — assim como sua versão normalizada, a correlação de Pearson — apenas medem relações do tipo linear, falhando em capturar interações mais complexas.

Para contornar essa aparente limitação, aqui, a dependência estatística entre pares de variáveis é quantificada através de medidas de informação mútua, oriundas da teoria da informação, que mede o quanto de informação uma variável aleatória contém a respeito de outra em termos de entropia de Shannon [20].

Essa escolha, no entanto, é acompanhada de um desafio inerente. Ao contrário das medidas de correlação de Pearson, que podem ser estimadas de maneira direta e confiável a partir de uma coleção de instâncias independentes, estimar a informação mútua entre um par de variáveis aleatórias pode ser um desafio considerável. Isso decorre da definição de informação mútua baseada em entropia, uma vez que a estimação de entropia (e consequentemente, de informação mútua) é um problema ativo de pesquisa há mais de 50 anos, sem que nenhuma solução se apresente como universal e confiável em todos os regimes [21].

Logo, antes de dar início aos estudos acerca da agregação de variáveis dependentes, este trabalho de mestrado se dedica, em um primeiro momento, a um estudo acerca do problema estimação de entropia e de informação mútua. O principal objetivo desse estudo é entender quais os maiores gargalos associados ao problema, quais as abordagens existentes na literatura e quais os regimes em que cada uma delas se aplica. Naturalmente, o estudo se dividiu em dois ramos que, devido a uma característica da própria literatura, são desenvolvidos em paralelo: a estimação da entropia para variáveis discretas e contínuas, cada uma com o seu conjunto próprio de abordagens e desafios associados.

Dentro do contexto do estudo sobre os estimadores de entropia para fontes discretas, os elementos essenciais que caracterizam os estimadores mais confiáveis foram identificados, a ponto de motivar a articulação desses elementos no desenvolvimento de um novo estimador, culminando na síntese de um artigo de conferência, publicado em

[22]. Ressalta-se, contudo, que os estudos subsequentes não dependem desse estimador em particular, podendo ser conduzidos a partir de diferentes abordagens existentes na literatura. A contribuição central da dissertação reside, portanto, no estudo acerca da imposição da agregação de variáveis dependentes nas arquiteturas de redes neurais, a partir da ótica da estimação de informação mútua, enquanto o método proposto para a estimação de entropia emerge como uma contribuição adicional e autônoma desse percurso investigativo.

De volta ao eixo principal da investigação, na motivação **M1**, enunciada anteriormente, é feita uma conexão entre o estudo proposto e as restrições baseadas em vizinhanças espaciais que são típicas das CNNs. Essa conexão é de interesse particular neste trabalho, devido a expectativa de que imagens naturais, a menos das bordas, sejam dominadas por componentes de baixa frequência, e, portanto, pares de pixels espacialmente adjacentes possuam valores de intensidade semelhantes [23]. Essa observação é antiga, explorada repetidamente por métodos locais de processamento de imagens, e se reflete em uma relação implícita em bases de imagens: pixels vizinhos nas imagens normalmente apresentam alta dependência estatística.

Essa simples intuição é retomada diversas vezes ao longo desta dissertação. Em particular, é verificado que medidas de informação entre variáveis aleatórias que representam pixels em imagens são suficientes à remontagem das relações de vizinhança 2D, mesmo após a aplicação de uma permutação fixa, assumida desconhecida, nas matrizes de pixels. Esse resultado é usado como uma prova de princípio, indicando que, em imagens, as CNNs podem ser interpretadas sob a perspectiva da agregação de variáveis dependentes a pesos compartilhados, e que, através da localização de dependências, os ganhos de desempenho delas (em relação as MLPs) podem ser obtidos mesmo quando o *a priori* resultante da ordenação natural dos pixels não está presente.

Outra investigação pertinente, que motiva a continuidade do trabalho, é acerca do quanto o princípio descrito acima é específico para as imagens, e de quais são as propriedades resultantes de sua aplicação em outros domínios. Especificamente, este trabalho se encerra com uma investigação acerca da imposição da agregação entre variáveis dependentes em *autoencoders* aplicados ao aprendizado de representações para dados tabulares, não equipados com a estrutura espacial inerente às imagens. Em outras palavras, investiga-se se o princípio da localização de dependências pode ser extrapolado para esse tipo de dado, e se essa abordagem é capaz de generalizar, ao menos parcialmente, os ganhos de desempenho característicos das CNNs.

Em resumo, as principais contribuições deste trabalho de mestrado podem ser listadas da seguinte forma:

(i) Apresentação de um estudo acerca do problema de estimação de entropia e infor-

mação mútua para variáveis aleatórias discretas e contínuas.

(ii) Proposição de um novo estimador de entropia para fontes discretas.

(iii) Proposição de um algoritmo para recuperar a topologia 2D de imagens usando medidas de informação mútua entre pares de pixels. Além disso, também é verificado que a informação de que o arranjo de pixels é bidimensional também pode ser recuperada através das medidas de informação mútua.

(iv) Experimentos demonstrando que os vieses indutivos [1] das CNNs podem ser recuperados, quase perfeitamente, mesmo quando a organização espacial dos pixels não é conhecida inicialmente.

(v) Proposição de um viés indutivo alicerçado em localidades artificiais, construídas a partir de medidas de informação mútua, para *autoencoders* aplicados a dados tabulares.

Essas contribuições estão organizadas ao longo da dissertação da seguinte maneira. No Capítulo 2 são introduzidos os conceitos de entropia, informação mútua e suas variantes, para fontes discretas e contínuas, bem como o problema de estimação e alguns dos principais estimadores. Para o caso discreto, um foco especial é dado ao estimador proposto. No Capítulo 3, são trazidos exemplos simples que ilustram as motivações para estudar a agregação entre variáveis dependentes. O Capítulo 4 traz um estudo sobre as CNNs sob a perspectiva da agregação entre variáveis dependentes, cujos *insights* obtidos motivam o estudo acerca da agregação entre variáveis dependentes no aprendizado de representações tabulares, apresentado no Capítulo 5. Finalmente, considerações finais são feitas no Capítulo 6.

[1] No aprendizado de máquina, o termo viés indutivo se refere a um conjunto de suposições que um algoritmo/modelo utiliza para restringir o espaço de hipóteses e facilitar a obtenção de uma regra de aprendizado generalizável.

2 Sobre medidas de informação mútua e o problema de estimação

Uma etapa fundamental deste trabalho é medir o grau de dependência estatística entre pares de variáveis, e, conforme dito anteriormente, as medidas de informação mútua foram adotadas como veículo para tal. Dessa forma, o objetivo deste capítulo é estabelecer os fundamentos necessários para a construção das estimativas utilizadas ao longo desta dissertação.

Para isso, neste capítulo, são revisados os conceitos de teoria da informação relevantes ao problema, discutidas as dificuldades envolvidas na estimação de entropia e informação mútua, para variáveis discretas e contínuas, e apresentados alguns dos estimadores considerados durante esta investigação, incluindo um estimador para fontes discretas que foi publicado como resultado deste trabalho. Ao final do capítulo, é justificada a adoção do estimador KSG como ferramenta de estimação para os experimentos desenvolvidos nos capítulos subsequentes.

Em um certo sentido, o estudo aqui apresentado ultrapassa a finalidade estritamente instrumental de justificar a escolha do estimador empregado no restante do trabalho. Durante seu desenvolvimento, questões relacionadas à estimação de entropia e informação mútua mostraram-se suficientemente relevantes para motivar uma investigação própria, incluindo o desenvolvimento de um novo estimador para fontes discretas. Embora essa contribuição não seja utilizada diretamente nos próximos capítulos, ela integra o percurso científico que conduziu à formulação da proposta principal e, por esse motivo, é preservada nesta dissertação.

2.1 Entropia de Shannon

Seja X uma variável aleatória (V.A.) discreta com suporte $\mathcal{X} = \{1, \ldots, S\}$, em que S é a cardinalidade do suporte, e com distribuição de probabilidade $P = (p_1, p_2, \ldots, p_S)$ — i.e. $\Pr(X = i) = p_i$ — , a entropia de Shannon dessa variável aleatória é dada por

$$H(X) = -\sum_{i=1}^{S} p_i \log(p_i), \tag{2.1}$$

em que, durante este trabalho, assume-se a convenção de que $-p\log(p) = 0$ quando $p = 0$. Alternativamente, é comum referir-se à entropia como um atributo da distribuição — e não da variável aleatória — e denotá-la por $H(P)$. A base logarítmica na definição de

entropia corresponde à escolha da unidade. Por exemplo, para a base 2 a entropia é dada em bits de informação, enquanto para a base natural e a entropia é dada em nats.

Esse índice, que quantifica o conteúdo de informação médio de uma fonte aleatória [1], foi originalmente proposto por Claude Shannon, em seu trabalho seminal em 1948 [20], no contexto do estudo de limites fundamentais para a compressão e a transmissão de dados em um canal de comunicação. De um ponto de vista mais geral, a medida de entropia quantifica o grau de incerteza envolvida em um experimento aleatório, e rapidamente encontrou várias aplicações em diversas áreas, para além da teoria da informação, como em linguística [24], aprendizado de máquina [25], bioinformática [26], dentre outros.

Duas propriedades relevantes da entropia de Shannon são listadas a seguir:

- Não negatividade: $H(X) \geq 0$, em que a igualdade ocorre se e somente se $p_i = 1$ para um elemento do suporte, e $p_i = 0$ para todos os outros elementos.
- A entropia é maximizada quando X é uniformemente distribuída: $H(X) \leq \log(S)$, com igualdade se e somente $p_i = 1/S$ para todo $i \in \{1, \ldots, S\}$.

Uma forma intuitiva de interpretar a medida de entropia é associando-a à noção de cardinalidade efetiva [27], definida como

$$C(X) = b^{H(X)}, \tag{2.2}$$

em que b é a base logarítmica usada para calcular a entropia.

Por definição, a quantidade $C(X)$ se iguala à cardinalidade do suporte de uma V.A. Y uniformemente distribuída com a mesma entropia que X. Para ver isso, perceba que, se esse for o caso, cada elemento em Y possui probabilidade $1/C(X)$, de modo que o conteúdo de informação médio (i.e. a entropia) é dado por $H(Y) = \log_b(C(X)) = \log_b(b^{H(X)}) = H(X)$. Note, no entanto, que a cardinalidade de uma verdadeira V.A. discreta é necessariamente um valor inteiro, o que não é o caso da cardinalidade efetiva. Portanto, trata-se de uma extrapolação do conceito de cardinalidade, que pode ajudar a interpretar o grau de incerteza envolvida em um experimento aleatório, como ilustrado no exemplo abaixo.

> **Exemplo:** Considere uma V.A. X cuja distribuição é $P = (0.75, 0.18, 0.07)$. Aplicando a definição na Equação 2.1, sua entropia $H(X)$ é de aproximadamente 1.025 bits, enquanto sua cardinalidade efetiva é dada por $C(X) \approxeq 2.035$.

[1] O conteúdo de informação de um símbolo x é dado por $\log\left(\frac{1}{\Pr(X=x)}\right)$ e mede o quão "incerto" ou "surpreendente" é o evento $X = x$. Ou seja, a partir da definição na Equação 2.1 é fácil intuir que a entropia mede a incerteza média acerca do resultado de um experimento aleaório.

Isso significa que, embora a V.A. X possa assumir três valores distintos, adivinhar o seu valor em uma dada realização é aproximadamente tão difícil quando adivinhar o resultado do lançamento de uma moeda honesta, que possui apenas duas faces.

2.1.1 Entropia conjunta

Sejam X e Y duas variáveis aleatórias discretas, com suportes $\mathcal{X} = \{1, \ldots, S_X\}$ e $\mathcal{Y} = \{1, \ldots, S_Y\}$, e distribuições P_X e P_Y, respectivamente. A entropia conjunta do par (X, Y) é dada por

$$H(X,Y) = -\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \Pr(X = x, Y = y) \cdot \log \Pr(X = x, Y = y). \tag{2.3}$$

Um caso de interesse particular é aquele em que X e Y são independentes, no qual a distribuição conjunta é dada por $\Pr(X = x, Y = y) = \Pr(X = x) \cdot \Pr(Y = y)$. Nesse caso,

$$\begin{aligned} H(X,Y) &= -\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \Pr(X = x) \cdot \Pr(Y = y)(\log \Pr(X = x) + \log \Pr(Y = y)) \\ &= -\sum_{y\in\mathcal{Y}} \Pr(Y = y) \cdot \sum_{x\in\mathcal{X}} \Pr(X = x) \cdot \log \Pr(X = x) \\ &\quad - \sum_{x\in\mathcal{X}} \Pr(X = x) \cdot \sum_{y\in\mathcal{Y}} \Pr(Y = y) \cdot \log \Pr(Y = y) \\ &= H(X) + H(Y). \end{aligned} \tag{2.4}$$

Ou seja, a entropia é aditiva para variáveis aleatórias independentes.

2.1.2 Entropia condicional

A entropia condicional de X dado o evento $Y = y$ é dada por

$$H(X|Y = y) = -\sum_{x\in\mathcal{X}} \Pr(X = x|Y = y) \cdot \log \Pr(X = x|Y = y). \tag{2.5}$$

A entropia condicional de X dado Y, por sua vez, é a esperança, em y, da entropia condicional de X dado o evento $Y = y$:

$$\begin{aligned} H(X|Y) &= -\sum_{y\in\mathcal{Y}} \Pr(Y = y) \sum_{x\in\mathcal{X}} \Pr(X = x|Y = y) \log \Pr(X = x|Y = y) \\ &= -\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \Pr(X = x, Y = y) \log \Pr(X = x|Y = y). \end{aligned} \tag{2.6}$$

A entropia condicional $H(X|Y)$ mede o grau de incerteza restante sobre o valor X, quando o valor de Y é conhecido. Abaixo, duas propriedades relevantes sobre $H(X|Y)$ são listadas:

- $H(X|Y) \leq H(X)$, com a igualdade ocorrendo apenas quando X e Y forem independentes.
- $H(X|Y) \geq 0$, com a igualdade ocorrendo apenas quando X for uma função determinística de Y.

2.1.3 Entropia relativa

A entropia relativa, ou divergência de Kullback-Leibler, entre duas distribuições de probabilidade $P = (p_1, \ldots, p_S)$ e $Q = (q_1, \ldots, q_S)$ definidas no mesmo suporte $\mathcal{X} = \{1, \ldots, S\}$ é dada por

$$D_{KL}(P||Q) = \sum_{i=1}^{S} p_i \log\left(\frac{p_i}{q_i}\right). \tag{2.7}$$

Uma propriedade relevante é a de que $D(P||Q) \geq 0$, com a igualdade ocorrendo apenas quando $P = Q$.

No contexto de teoria da informação, a divergência de Kullback-Leibler mede a quantidade adicional de bits por símbolo que seriam usados caso os símbolos de uma fonte discreta fossem codificados assumindo uma distribuição Q, quando P é a verdadeira distribuição da fonte. No entanto, em contextos mais gerais, incluindo o próprio aprendizado de máquina [28, 25], a divergência de Kullback-Leibler é usada como uma medida de dissimilaridade entre duas distribuições de probabilidade.

2.2 Entropia diferencial

A entropia de Shannon pode ser estendida para medir a incerteza/aleatoriedade em variáveis aleatórias contínuas usando o conceito de entropia diferencial. Seja X uma variável aleatória contínua cujo conjunto suporte é $\mathcal{X}$ e a função densidade de probabilidade é f_X, a entropia diferencial de X é dada por

$$h(X) = \mathbb{E}[\log(1/f_X(X))] = -\int_{\mathcal{X}} f_X(x) \log f_X(x) dx. \tag{2.8}$$

A definição acima abrange tanto os casos em que X é uma variável aleatória univariada (i.e. $\mathcal{X} \subseteq \mathbb{R}$) quanto os casos em que X é uma variável aleatória multivariada (i.e. $\mathcal{X} \subseteq \mathbb{R}^d$, em que $d > 1$).

Análogo ao caso da entropia para fontes discretas, uma noção de volume efetivo pode ser obtida aplicando a potência de base b, em que b é a base logarítmica usada, na entropia diferencial:

$$V(X) = b^{h(X)}. \tag{2.9}$$

Ou seja, $V(X)$ é a medida de Lebesgue do conjunto suporte de uma V.A. uniformemente distribuída [29], cuja entropia diferencial é $h(X)$[2].

Uma diferença notável em comparação ao caso discreto, em que a cardinalidade do suporte é sempre maior ou igual a um, é que a medida de Lebesgue do suporte de uma V.A. contínua pode assumir valores menores que 1, que implica valores negativos quando posta em escala logarítmica. Ou seja, a entropia diferencial pode assumir valores negativos, e tende para $-\infty$ quando o volume efetivo tende para zero — i.e. quando toda a massa de probabilidade se concentra em um único resultado possível.

Outra diferença em relação ao caso discreto é a de que se X é uma função invertível de Y, não é possível afirmar que $h(X) = h(Y)$, como demonstrado no exemplo a seguir.

> **Exemplo:** Seja $X \sim U(0,1)$ uma variável aleatória uniformemente distribuída entre 0 e 1 e $Y = 2X$. Claramente, $Y \sim U(0,2)$, e as entropias diferenciais de X e Y são dadas, respectivamente, por
>
> $$h(X) = -1 \cdot \log_2 1 \cdot (1-0) = 0 \text{ bits} \tag{2.10}$$
>
> e
>
> $$h(Y) = -0.5 \cdot \log_2 0.5 \cdot (2-0) = 0.6931 \text{ bits}. \tag{2.11}$$

Isso introduz um aspecto contraintuitivo na definição de entropia diferencial como medida de incerteza associada a uma variável aleatória contínua. A saber, embora exista uma relação de um para um entre X e Y, e o conhecimento do valor de uma das duas variáveis determina unicamente o valor da outra, as entropias diferenciais $h(X)$ e $h(Y)$ são diferentes. Isso decorre do fato que a entropia diferencial não depende somente da incerteza acerca do valor de uma variável aleatória, mas também de sua escala.

Usando a definição de entropia diferencial, os conceitos de entropia conjunta, entropia condicional e entropia relativa (divergência de Kullback-Leibler) podem ser estendidos para variáveis/distribuições contínuas, conforme apresentado nas equações 2.12, 2.13 e 2.14, respectivamente:

$$h(X,Y) = -\int_{\mathcal{X}\times\mathcal{Y}} f_{XY}(x,y) \log f_{XY}(x,y) dxdy, \tag{2.12}$$

[2] A medida de Lebesgue é uma generalização do conceito de comprimento em $\mathbb{R}$, área em $\mathbb{R}^2$ e volume em $\mathbb{R}^3$.

$$h(X|Y) = -\int_{\mathcal{X}\times\mathcal{Y}} f_{XY}(x,y)\log f_{XY}(x|Y=y)dxdy, \tag{2.13}$$

$$D_{KL}(f||q) = \int_{\mathcal{X}} f_X(x)\log\left(\frac{f_X(x)}{q_X(x)}\right)dx. \tag{2.14}$$

No caso específico da divergência de Kullback-Leibler, f_X e q_X representam, respectivamente, a verdadeira função densidade de probabilidade da variável X, e uma distribuição candidata definida no mesmo domínio. Ou seja, esse abuso de notação não deve ser interpretado como a existência de duas distribuições diferentes para uma mesma variável X.

2.3 Informação mútua entre variáveis aleatórias discretas

Sejam X e Y um par de variáveis aleatórias discretas com suporte em $\mathcal{X}\times\mathcal{Y}$, distribuição conjunta P_{XY} e distribuições marginais P_X e P_Y. A informação mútua entre X e Y é definida como

$$I(X;Y) = D_{KL}(P_{XY}||P_X\cdot P_Y) = \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \Pr(\text{X=x, Y=y})\cdot\log\frac{\Pr(\text{X=x, Y=y})}{\Pr(X=x)\cdot\Pr(Y=y)}. \tag{2.15}$$

A partir da definição acima, a informação mútua mede a dissimilaridade, através da divergência de Kullback-Leibler, entre a distribuição conjunta do par (X,Y) e o que seria a distribuição conjunta caso X e Y fossem independentes. Como mencionado na Seção 2.1.3, $D_{KL}(P_{XY}||P_X\cdot P_Y) = 0$ se e somente se $P_{XY} = P_X\cdot P_Y$. Isto é, $I(X;Y)\geq 0$, em que a igualdade ocorre apenas quando X e Y são independentes. Partindo dessa definição, outra propriedade imediata da informação mútua é a simetria, isto é: $I(X;Y) = I(Y;X)$.

Equivalentemente, a informação mútua entre X e Y também pode ser definida em termos de entropias e entropias condicionadas:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{2.16}$$

Sob a perspectiva dessa definição, a informação mútua pode ser interpretada como a redução na incerteza acerca do valor de X devido ao conhecimento do valor de Y (e vice-versa). Se existir uma relação bijetiva entre X e Y, isto é, se o conhecimento do valor de uma das variáveis determinar unicamente o valor da outra, então $H(X|Y) = H(Y|X) = 0$ e $I(X;Y) = H(X) = H(Y)$.

Também pode-se inferir que, para variáveis aleatórias discretas, a informação mútua satisfaz a desigualdade

$$0 \leq I(X;Y) \leq \min(H(X), H(Y)). \tag{2.17}$$

Finalmente, outra definição equivalente que é conveniente em muitos casos é dada por

$$I(X;Y) = H(X) + H(Y) - H(X,Y). \tag{2.18}$$

A informação mútua satisfaz propriedades que a fazem uma medida ideal de dependência entre variáveis aleatórias. Diferente do coeficiente de correlação de Pearson, que leva em conta somente relações lineares, e outros coeficientes de correlação de postos, que apenas levam em conta relações monotônicas, a informação mútua leva em conta todo tipo de dependência. Por outro lado, um aspecto que dificulta sua interpretabilidade, conforme expresso na desigualdade em 2.17, é o de que o valor máximo que $I(X;Y)$ assume, quando a dependência é máxima, depende das entropias marginais das variáveis envolvidas. Em muitas situações, no entanto, é desejável trabalhar com versões normalizadas dessa medida, que facilitam interpretações e comparações.

2.3.1 Versões normalizadas

Como mencionado anteriormente, a informação mútua $I(X;Y) = H(X) - H(X|Y)$ quantifica a redução da incerteza associada ao valor de X devido ao conhecimento do valor de Y. Logo, uma maneira natural de contornar sua sensibilidade ao valor de $H(X)$ e obter uma versão normalizada é pensando em termos de redução relativa da incerteza em X dado Y, como descrito na equação

$$U(X;Y) = \frac{H(X) - H(X|Y)}{H(X)} = \frac{I(X;Y)}{H(X)}. \tag{2.19}$$

O coeficiente $U(X;Y)$ é conhecido como coeficiente assimétrico de incerteza, definido em [30]. No mesmo trabalho, também foi definida uma versão simétrica desse coeficiente, dada por

$$S(X;Y) = \frac{2I(X;Y)}{H(X) + H(Y)}. \tag{2.20}$$

Essa medida, por sua vez, assume valores no intervalo $[0, 1]$, e satisfaz a propriedade de que $S(X;Y) = 1$ se e somente se existir uma relação de um para um entre X e Y.

Finalmente, em [31], uma outra versão normalizada da informação mútua, que também assume valores entre 0 e 1, foi definida como

$$I_d^*(X;Y) = \frac{I(X;Y)}{\min(H(X), H(Y))}. \tag{2.21}$$

2.4 Informação mútua entre variáveis aleatórias contínuas.

Análogo ao caso discreto, a informação mútua entre um par de variáveis aleatórias contínuas X e Y, com densidades conjuntas f_{XY} e densidades marginais f_X e f_Y, é dada pela divergência de Kullback-Leibler entre as distribuições f_{XY} e $f_X \cdot f_Y$:

$$I(X;Y) = \int_{\mathcal{X}\times\mathcal{Y}} f_{XY}(x,y) \log \frac{f_{XY}(x,y)}{f_X(x) \cdot f_Y(y)} dxdy, \tag{2.22}$$

em que $\mathcal{X}$ e $\mathcal{Y}$ são os suportes de X e Y, respectivamente. Da mesma maneira, a informação mútua entre variáveis contínuas pode ser definida em termos de entropias diferenciais a partir das equações

$$I(X;Y) = h(X) - h(X|Y) = h(Y) - h(Y|X), \tag{2.23}$$

e

$$I(X;Y) = h(X) + h(Y) - h(X,Y). \tag{2.24}$$

Pela definição na Equação 2.22, uma propriedade imediata é a de que $I(X;Y) \geq 0$, em que, assim como no caso discreto, a igualdade ocorre se e somente se X e Y forem independentes. No entanto, quando existe uma relação biunívoca entre X e Y, como no Exemplo 2.2, $h(X|Y)$ e $h(Y|X)$ tendem para $-\infty$, fazendo com que a informação mútua tenda para ∞.

Por esse motivo, as variantes normalizadas da informação mútua entre variáveis discretas, apresentadas nas Equações 2.19, 2.20 e 2.21, não podem ser generalizadas para o caso contínuo.

Em [31], uma versão normalizada da informação mútua entre variáveis aleatórias contínuas foi definida, partindo da observação de que, se X e Y são normalmente distribuídas, com coeficiente de correlação ρ, a informação mútua entre elas é dada por $I(X;Y) = -1/2 \cdot \log(1-\rho^2)$. A medida de informação mútua normalizada proposta em [31] é dada por

$$I_c^*(X;Y) = \sqrt{1 - \exp(-2I(X;Y))}. \tag{2.25}$$

Claramente, I_c^* coincide com o coeficiente de correlação quando X e Y são normalmente distribuídas. Além disso, quando X e Y são independentes, $I(X;Y) = 0$ implica $I_c^*(X;Y) = 0$, e quando X e Y são completamente dependentes, $I(X;Y) \to \infty$ implica $I_c^*(X;Y) \to 1$.

Para variáveis aleatórias contínuas, uma propriedade importante da informação mútua é que, diferente da entropia diferencial, ela é uma medida invariante a reparametrizações. Isto é, se $X' = F(X)$ e $Y' = G(Y)$ são funções invertíveis, então $I(X;Y) =$

$I(X';Y')$. Assim, apesar da entropia diferencial ser sensível a mudanças de escala, a informação mútua é preservada. Essa é uma propriedade bastante desejável, enquanto medida dependência entre variáveis, e a diferencia de outras medidas, como aquelas baseadas em generalizações da entropia de Shannon, como as entropias de Rényi [32].

2.5 Estimação de informação mútua entre variáveis discretas

O restante do capítulo se dedica ao problema de como estimar a informação mútua entre duas variáveis aleatórias X e Y, tanto para o caso discreto quanto para o caso contínuo, dado um conjunto de amostras do par (X, Y).

Em ambos os casos, o problema de estimar informação mútua se reduz ao problema de estimação de entropia (ou entropia diferencial, para o caso contínuo). No caso discreto, dado um estimador de entropia, a informação mútua pode ser estimada substituindo as estimativas das entropias marginais e conjuntas na Equação 2.18. O mesmo pode ser dito para o caso contínuo, em que um estimador de entropia diferencial pode ser usado para estimar informação mútua usando a definição apresentada na Equação 2.24. Além disso, todas as variantes normalizadas da informação mútua aqui apresentadas também podem ser obtidas a partir de estimativas de entropia.

Para o caso discreto, um estimador de entropia foi proposto durante a primeira parte do desenvolvimento deste trabalho, resultando em artigo que foi publicado na *International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA 2026)*, disponível em [22]. No restante desta seção, o problema de estimação de entropia de variáveis aleatórias discretas é introduzido, em termos gerais, seguido da apresentação do estimador proposto.

2.5.1 Estimação de entropia de Shannon

Dada uma variável aleatória discreta X com suporte $\mathcal{X} = \{1, \ldots, S\}$ e distribuição $P = (p_1, \ldots, p_S)$, a entropia de X pode ser calculada usando a Equação 2.1. No entanto, em situações práticas, é muito raro que o parâmetro P seja conhecido, e toda informação disponível é um conjunto de N realizações (instâncias independentes) da variável aleatória, denotado por $X^N = (x_1, x_2, \ldots, x_N)$. Dessa forma, o problema de estimar a entropia é formulado como a construção de uma função que recebe como entrada o conjunto de dados e retorna como saída uma estimativa para a entropia da distribuição que gerou os dados de entrada. Isto é, o objetivo é obter $\hat{H}$ tal que $\hat{H}(X^N) \approx H(P)$.

Por se tratar de um funcional de uma distribuição de probabilidade, a ideia mais natural para se estimar a entropia é obter estimativas empíricas para a distribuição de probabilidade $\hat{P} = (\hat{p}_1, \ldots, \hat{p}_S)$ e substituí-las na Equação 2.1 para obter $\hat{H}$, em que

as estimativas empíricas são dadas por $\hat{p}_i = \frac{n_i}{N}$, com n_i representando a quantidade de ocorrências do símbolo i na amostra X^N. De fato, esse estimador, denominado *plug in*, é o mais simples e o mais usado na estimação da entropia de fontes discretas. Porém, é amplamente conhecido na literatura que essa estimativa é altamente imprecisa quando a quantidade de instâncias N é pequena, e possivelmente menor que a cardinalidade do suporte S, o que é uma situação cada vez mais comum em aplicações modernas [24, 33]. Na próxima seção, essa deficiência do estimador *plug in* é ilustrada e discutida através de um exemplo.

2.5.2 Sobre o gargalo da estimação empírica

A principal limitação do estimador empírico se dá pelo fato de que, em amostras pequenas, é esperado que símbolos pouco prováveis não sejam observados e, consequentemente, probabilidades nulas sejam atribuídas a eles pelo estimador empírico. Ao mesmo tempo, a derivada da função $f(x) = -x\log(x)$ tende a infinito quando x tende a 0. Ou seja, trata-se de um "golpe duplo": se por um lado as estimativas de probabilidade dos símbolos mais raros tendem a ser subestimadas pelo estimador empírico, por outro lado, esses símbolos tendem a causar erros maiores nas estimativas de entropia quando pequenos erros são cometidos na estimativa de suas probabilidades. De fato, essa observação explica o fato do estimador *plug in* ser negativamente enviesado [34], podendo levar a resultados desastrosos em regimes de escassez de dados.

O exemplo a seguir ilustra o argumento apresentado acima.

Exemplo: Considere uma distribuição de Dirichlet definida no *simplex* de probabilidade com dimensão $S = 1000$, com parâmetros $\alpha_1 = \cdots = \alpha_{1000} = 0.02$. Dessa distribuição, foi instanciada uma função massa de probabilidade P cuja entropia é $H = 4.3772$ nats. Em seguida, foi obtida uma amostra com $N = 1000$ instâncias de variáveis aleatórias i.i.d. $X^{1000} = (X_1, \ldots, X_{1000})$. Um conjunto de estatísticas suficientes (ver seção 3.6 em [35] para uma definição formal) para estimar a entropia a partir de uma amostra é dado por

$$h_i = |\{j \text{ t.q. } n_j = i\}|, \tag{2.26}$$

em que $|\cdot|$ representa cardinalidade do conjunto e, novamente, n_j representa a quantidade de vezes que o símbolo j foi observado na amostra. Colocando em palavras, h_i representa a quantidade de símbolos que foram observados i vezes na amostra. Na literatura, é comum referir-se a $h_1, \ldots, h_N$ como o perfil da amostra. Para a amostra coletada neste exemplo, o perfil é mostrado na Figura 1.

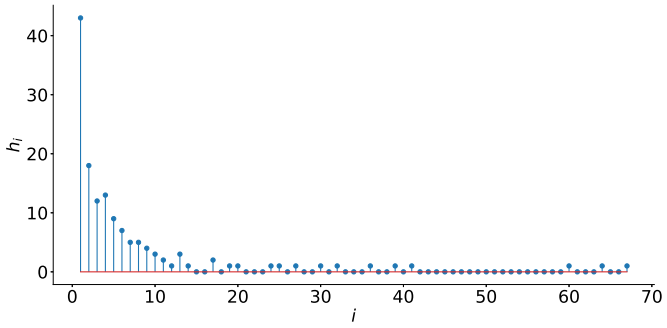


Figura 1 – Perfil da amostra X^{1000}.

Nessa amostra, apenas 141 símbolos diferentes foram observados, em um total de 1000 símbolos possíveis. No entanto, os 859 símbolos não observados são extremamente raros, e a massa de probabilidade total desse conjunto, dada por

$$m_0 = \sum_{i \ t.q. \ n_i=0} p_i, \tag{2.27}$$

é de apenas 0.0467.

No estimador *plug in*, cada símbolo contribui para a estimativa final com um erro $e = p\log(p) - \frac{n}{N}\log\left(\frac{n}{N}\right)$, em que p é a verdadeira probabilidade do símbolo em questão e n é a quantidade de ocorrências do mesmo. Assim, podemos associar ao conjunto de símbolos que foi observado n vezes um erro acumulado, definido por

$$e_n = \sum_{i \ t.q. \ n_i=n} p_i \log(p_i) - \frac{n}{N}\log\left(\frac{n}{N}\right). \tag{2.28}$$

Para a amostra coletada da distribuição P, o erro e_n, em função de n, é apresentado na Figura 2

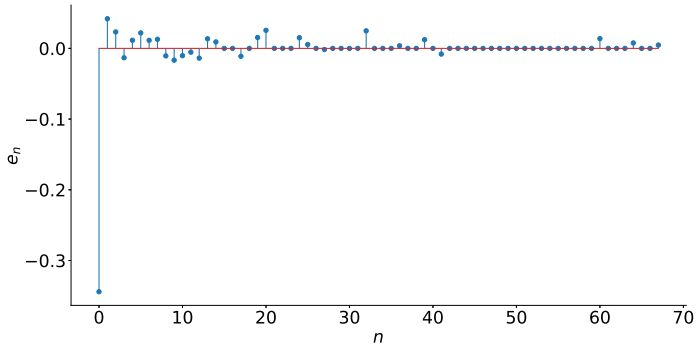


Figura 2 – Erro acumulado pelo estimador *plug in* em função da quantidade de ocorrências.

Ao analisar esse gráfico, é possível observar um fenômeno intrigante: um evento extremamente raro, que só ocorre cerca de 5% das vezes, é responsável por praticamente

todo o viés induzido na estimativa da entropia. Esse simples exemplo ilustra a relevância de se levar em conta a quantidade e a massa de probabilidade dos símbolos não observados na amostra para a estimação da entropia.

No estimador proposto neste trabalho, a contribuição dos símbolos não vistos é considerada através de ferramentas que permitem estimar essas duas propriedades do conjunto de símbolos não vistos em uma amostra. Os estimadores dessas propriedades são apresentados na sequência.

2.5.3 Estimando a massa total dos símbolos que foram observados k vezes

Seja X^N uma amostra com perfil $(h_1, \ldots, h_N)$, de uma V.A. com distribuição $P = (p_1, \ldots, p_S)$, a probabilidade total dos símbolos que foram observados k vezes em X^N é dada por

$$m_k = \sum_{i \ t.q. \ n_i = k} p_i. \tag{2.29}$$

Um estimador tradicional para m_k é o estimador de Good-Turing [36]. Ele foi desenvolvido sob um modelo de Poisson, em que o tamanho da amostra é considerado uma V.A. com distribuição de Poisson com média N, em vez de fixo. Essa hipótese facilita a análise, uma vez que, sob esse modelo, $n_1, \ldots, n_S$ são variáveis independentes com distribuições de Poisson e parâmetros $\lambda_i = \lambda p_i$, respectivamente, em vez de seguirem uma distribuição multinomial, que seria o caso com N fixo.

Em [37], os autores levaram em conta as dependências entre as variáveis $n_1, \ldots, n_S$ sob o modelo multinomial, e desenvolveram uma expressão para o viés do estimador de Good-Turing, que é incorretamente identificado como não enviesado sob o modelo de Poisson. Com a análise, um estimador com viés mínimo para m_k foi proposto, dado por

$$\hat{m}_k = -\binom{N}{k} \sum_{i=1}^{N-k} \frac{(-1)^i h_{k+1}}{\binom{N}{k+1}}. \tag{2.30}$$

Além disso, os autores também propuseram um estimador de erro quadrático médio mínimo, que pode ser obtido através de um problema de busca.

Um caso particular deste estimador é $\hat{m}_0$, que é uma estimativa da massa de probabilidade total do conjunto de símbolos que não foram observados na amostra X^N.

2.5.4 Estimando a quantidade de símbolos não vistos

Na sua formulação mais tradicional, o problema de estimar o número de símbolos não vistos é posto da seguinte forma: dada uma amostra X^N, qual a quantidade U de símbolos não vistos em X^N que seriam observados se uma nova amostra X^M fosse coletada? Por conveniência, escrevemos $M = aN$, em que a é um fator de amplificação da

amostra. O estimador mais tradicional para U foi proposto em [38], por Good e Toulmin. O estimador de Good-Toulmin é não enviesado para qualquer a, mas a variância cresce indefinidamente para $a > 1$. Em [39], ele foi modificado usando uma técnica de suavização, de modo a garantir erro quadrático médio ótimo para qualquer a na ordem de $\log(N)$ ou menor. Finalmente, em [40], o estimador foi estendido para levar em conta a multiplicidade dos símbolos não vistos. Ou seja, dada uma amostra X^N, qual a quantidade U_μ de símbolos não vistos em X^N que seriam observados pelo menos μ vezes em X^M. O estimador proposto em [40] é dado por

$$\hat{U}_\mu = \sum_{i=1}^{N} s_i h_i \tag{2.31}$$

em que

$$s_i = -\sum_{j=0}^{\min\{\mu-1,i\}} (-a)^i (-1)^j \binom{i}{j} Pr(\text{Poi}(r) \geq i + j), \tag{2.32}$$

em que $\text{Poi}(r)$ denota uma variável aleatória de Poisson com parâmetro r, que por sua vez é um hiperparâmetro que os autores fixam em

$$r = \frac{\log(n(a+1)^2/(a-1))}{2a}. \tag{2.33}$$

Uma implementação do estimador apresentado acima, bem como reproduções dos resultados apresentados em [40], pode ser encontrada no repositório disponível em `https://github.com/gabrielfab0022/Estimating-the-number-of-Unseen-Species-With-Multiplicity---Reproduction`.

2.5.5 Estimador proposto

Baseado no raciocínio desenvolvido na Seção 2.5.2, é possível fazer três observações: (i) Os símbolos não observados na amostra induzem grande parte do viés na estimativa da entropia (ii) símbolos que foram observados com baixa frequência também tendem a induzir um grande viés de estimação, pois correspondem a probabilidades pequenas, na região de alta inclinação no gráfico da função $f(x) = -x\log(x)$ (iii) a estimativa empírica dos símbolos que aparecem com maior frequência é mais confiável para fins de estimação de entropia.

Tendo isso em vista, o estimador proposto neste trabalho é baseado na conjectura que se o suporte da V.A. de interesse for dividido nesses três subconjuntos, é possível obter uma estimativa confiável da entropia se estratégias diferentes forem escolhidas para estimar a contribuição de cada um dos três subconjuntos separadamente. Para isso, dada uma amostra X^N, definem-se os três subconjuntos a seguir:

$$S_1 = \{i \text{ t.q. } n_i = 0\}, \tag{2.34}$$

$$S_2 = \{i \text{ t.q. } 0 < n_i \leq \lambda\}, \tag{2.35}$$

$$S_3 = \{i \text{ t.q. } n_i > \lambda\}. \tag{2.36}$$

Colocando em palavras, S_1 é o conjunto de símbolos não observados, S_2 é o conjunto de símbolos que foram observados raramente e S_3 é o conjunto de símbolos observados com frequência. Aqui, a noção de raridade está associada a um limiar λ. Dessa forma, define-se a distribuição $P_S = (P_{S_1}, P_{S_2}, P_{S_3})$ em que $P_{S_i} = Pr(X \in S_i)$. Com isso, é possível usar a propriedade da decomposibilidade — ver seção 2.5 de [41] —, que é uma propriedade recursiva que permite reescrever a entropia como

$$H(X) = H(P_S) + P_{S_1} H(X|X \in S_1) + P_{S_2} H(X|X \in S_2) + P_{S_3} H(X|X \in S_3). \tag{2.37}$$

A proposta consiste em estimar cada um dos quatro termos separadamente, usando a estratégia mais adequada para cada um deles. Para os dois primeiros termos, é necessário estimar propriedades do conjunto de símbolos não vistos. Para esse fim, são usadas as ferramentas apresentadas nas subseções 2.5.3 e 2.5.4. Abaixo, são apresentadas as estratégias para a estimação de cada um dos quatro termos na Equação 2.37.

(A) Estimação do primeiro termo

Para o primeiro termo $H(P_S)$, as probabilidades de cada subconjunto podem ser estimadas usando o estimador de massa total, como apresentado na Equação 2.30, da seguinte forma:

$$\hat{P}_{S_1} = \hat{m}_0, \tag{2.38}$$

$$\hat{P}_{S_2} = \sum_{i=1}^{\lambda} \hat{m}_i, \tag{2.39}$$

$$\hat{P}_{S_3} = 1 - \hat{P}_{S_1} - \hat{P}_{S_2}. \tag{2.40}$$

As estimativas $\hat{P}_{S_1}, \hat{P}_{S_2}$ e $\hat{P}_{S_3}$ podem ser substituídas diretamente na definição de entropia (Equação 2.1) para obter uma estimativa $\hat{H}(P_S)$ para o primeiro termo.

(B) Estimação do segundo termo

A entropia condicional $H(X|X \in S_1)$ é a parcela correspondente aos símbolos não observados na amostra, responsáveis pela maior parte do viés negativo na estimação da entropia. Para estimar $H(X|X \in S_1)$, será usada a estimativa da quantidade de símbolos não vistos, dada pela Equação 2.31 (com $\mu = 1$), e será assumida uma distribuição uniforme dentro desse conjunto de símbolos muito raros, uma vez que nenhuma outra

informação *a priori* sobre as verdadeiras probabilidades dos símbolos nesse conjunto está disponível. Com isso a estimativa será dada por

$$\hat{H}(X|X \in S_1) = \log(\hat{U}_1). \tag{2.41}$$

Combinando as Equações 2.38 e 2.41, pode-se obter uma estimativa $\hat{P}(S_1)\hat{H}(X|X \in S_1)$ para o segundo termo.

(C) Estimação do terceiro termo

A entropia do segundo subconjunto $H(X|X \in S_2)$ é a parcela correspondente aos símbolos observados raramente na amostra. Ela também é problemática por ser sensível a flutuações estatísticas, de modo que as estimativas empíricas de probabilidade não são confiáveis para o propósito de estimação de entropia. Para mitigar esse efeito, propõe-se que a distribuição também seja considerada uniforme entre os símbolos pouco frequentes. Assim, obtém-se a estimativa

$$\hat{H}(X|X \in S_2) = \log(|S_2|). \tag{2.42}$$

Combinando as Equações 2.39 e 2.42, pode-se obter uma estimativa $\hat{P}(S_2)\hat{H}(X|X \in S_2)$ para o terceiro termo.

(D) Estimação do quarto termo

A última entropia condicional $H(X|X \in S_3)$ é a parcela correspondente aos símbolos observados com maior frequência. Como se tratam de símbolos com probabilidade maior, a estimativa de suas contribuições para a entropia é menos problemática. Dessa forma, é usado o tradicional estimador de Miller-Madow para estimar essa entropia condicionada:

$$\hat{H}(X|X \in S_2) = -\sum_{i \ t.q. \ n_i > \lambda} \frac{n_i}{N_{S_3}} \log\left(\frac{n_i}{N_{S_3}}\right) + \frac{|S_3| - 1}{2N_{S_3}}, \tag{2.43}$$

em que N_{S_3} é a quantidade de instâncias que resultaram em símbolos pertecentes a esse conjunto, dada por $N_{S_3} = \sum_{i \ t.q. \ n_i > \lambda} n_i$. Assim, pode-se combinar as Equações 2.40 e 2.43 para obter uma estimativa $\hat{P}(S_3)\hat{H}(X|X \in S_3)$ do quarto e último termo.

Finalmente, é possível substituir as estimativas de cada um dos quatro termos na Equação 2.37 para obter uma estimativa final para a entropia.

O estimador proposto tem dois hiperparâmetros de fácil interpretabilidade. O limiar que separa os subconjuntos S_2 e S_3 foi ajustado de maneira empírica, e fixado em $\lambda = 3$. O parâmetro a, por sua vez, determina o fator de amplificação da amostra usado para estimar a quantidade de símbolos não vistos. Quanto maior for o valor de a, maior a chance dessa estimativa englobar todos os símbolos não vistos. Porém, aumentar a também implica um aumento na variância do estimador [39]. Assim, para escolher a, apelamos

para a simples intuição de que se a cobertura amostral for baixa, precisamos de uma amplificação maior para cobrir toda a massa de probabilidade dos símbolos não observados. Dessa forma, a é escolhido em função de $\hat{m}_0$ (a estimativa da massa de probabilidade faltante) da seguinte forma:

$$a = \begin{cases} 4 \times 10^5, \text{ se } \hat{m}_0 \geq 0.8, \\ 100, \text{ se } 0.7 \leq \hat{m}_0 < 0.8, \\ 8, \text{ se } 0.55 \leq \hat{m}_0 < 0.7, \\ 5, \text{ se } 0.4 \leq \hat{m}_0 < 0.55, \\ 2, \text{ se } 0.3 \leq \hat{m}_0 < 0.4, \\ 1.5, \textit{ se } 0.15 \leq \hat{m}_0 < 0.3, \\ 1, \textit{ se } \hat{m}_0 < 0.15. \end{cases} \tag{2.44}$$

Essa *look-up table*, usada para determinar o valor de a, foi ajustada empiricamente a partir de amostras de uma distribuição uniforme, comparando a entropia estimada com entropia verdadeira. Após o ajuste, ela foi mantida fixa como parte do método em todos os experimentos com diferentes distribuições, para garantir a capacidade de generalização do método. Esses experimentos foram apresentados no artigo publicado em [22], incluindo comparações experimentais que mostram que o método proposto possui um desempenho superior ao de estimadores clássicos em regimes de escassez de dados, e comparável aos estimadores que são considerados o estado da arte, como os das referências [42, 43].

A implementação do estimador, junto com a avaliação experimental usando diferentes combinações de distribuições, tamanho de amostra e cardinalidade do suporte, está disponível no repositório https://github.com/gabrielfab0022/Partitioning-the-Sample-Space-for-a-More-Precise-Shannon-Entropy-Estimation.

2.6 Estimação de informação mútua entre variáveis contínuas

Análogo ao caso discreto, o problema de estimar a informação mútua entre duas variáveis aleatórias contínuas X e Y a partir de amostras do par (X, Y) se reduz ao problema de estimação das entropias diferenciais $h(X)$, $h(Y)$ e $h(X, Y)$ — Equação 2.24.

Da mesma maneira, um método bastante natural de estimar a entropia diferencial de uma V.A. X é substituir sua verdadeira densidade de probabilidade por uma estimativa não paramétrica $\hat{f}_X$ da mesma na definição — Equação 2.8 —, e então estimar a entropia diferencial por integração numérica [44].

No entanto, dadas as dificuldades associadas ao processo de integração numérica, uma maneira ainda mais eficiente (e mais usada) de estimar a entropia diferencial, que

é definida como o valor esperado da quantidade $\log(1/f_X(x))$, é através da estimação empírica dessa média, conforme

$$\hat{h}(X^N) = -\frac{1}{N}\sum_{i=1}^{N}\log \hat{f}_X(x_i), \tag{2.45}$$

em que $X^N = (x_1, x_2, \ldots, x_N)$ é um conjunto de N instâncias independentes da variável X.

Resta, então, determinar $\hat{f}_X$ através de estimadores não paramétricos de densidades de probabilidade. Estes, por sua vez, são tipicamente baseados na hipótese de que em uma pequena região $\mathcal{R}$ em torno de um ponto x, a densidade de probabilidade f_X é aproximadamente constante, de modo que a probabilidade p de que uma determinada instância de X pertença a $\mathcal{R}$ pode ser aproximada por

$$p = \int_{\mathcal{R}} f_X(x')dx' \approx f_X(x)V, \tag{2.46}$$

em que x' é uma variável de integração e V é o volume da região $\mathcal{R}$ [45]. Considerando N realizações independentes de X, a quantidade k de amostras que pertencem a $\mathcal{R}$ é uma variável do tipo binomial com parâmetros N e p. Portanto, seu valor esperado é dado por $\mathbb{E}[k] = Np$ e sua variância é dada por $\text{var}[k] = Np(1-p)$. Da mesma maneira, temos que $\mathbb{E}[k/N] = p$ e $\text{var}[k/N] = p(1-p)/N$. Ou seja, à medida que o tamanho da amostra N cresce, a probabilidade da razão k/N se concentra quase que inteiramente em p [3]. Logo, segue que

$$p \approx f_X(x)V \approx k/N, \tag{2.47}$$

e portanto uma boa estimativa para $f_X(x)$ é dada por

$$\hat{f}_X(x) = \frac{k}{NV}. \tag{2.48}$$

Uma maneira de explorar o desenvolvimento acima é fixando um volume e, para cada ponto de interesse, encontrar o valor de k usando o conjunto de dados X^N. Essa abordagem caracteriza os estimadores baseados em *kernels* — em inglês, *kernel density estimators*, KDE. Por exemplo, a região $\mathcal{R}$ pode ser um hipercubo centrado em x, de modo que a estimativa é dada por

$$\hat{f}_X(x) = \frac{1}{Nh^d}\sum_{i=1}^{N}\phi\left(\frac{x - x_i}{h}\right), \tag{2.49}$$

em que d é a dimensão do espaço ambiente, h controla a aresta do hipercubo e define, implicitamente, a escala de observação dos dados. Além disso, ϕ é o *kernel*, definido por

$$\phi(u) = \begin{cases} 1, \ \forall \ \|u\|_\infty \leq 1/2 \\ 0, \ \forall \ \|u\|_\infty \ > 1/2, \end{cases} \tag{2.50}$$

[3] Como pode ser visto pelo fato de que a variância de k/N tende para 0.

em que $\|\cdot\|_\infty$ é a norma do supremo. Para esse *kernel*, o somatório na Equação 2.49, funciona como um simples contador de quantas observações pertencem ao hipercubo de aresta h centrado em x.

A estimativa descrita acima sofre com o fato de que a densidade de probabilidade é estimada como uma soma de funções descontínuas. No entanto, pode ser demonstrado que se o *kernel* satisfizer a definição de uma função densidade de probabilidade, então a própria função $\hat{f}_X(x)$ irá convergir para uma verdadeira função densidade de probabilidade [45]. Modelos mais suaves podem ser obtidos usando outros *kernels*, como uma distribuição Gaussiana, que é centrada em cada ponto de interesse e as contribuições são somadas ao longo de todo o conjunto de dados.

Ao longo dos anos, diversos estimadores de entropia diferencial foram propostos na literatura usando estimadores de densidade de probabilidades baseados em *kernels* [44, 46, 47, 48].

Outra maneira de explorar a Equação 2.48 é fixando o valor de k e aumentando o volume da região centrada em x até que ela compreenda exatamente k observações. Essa abordagem caracteriza os métodos baseados em k vizinhos mais próximos — k *nearest neighbors*, $k-$NN. O parâmetro k controla a escala de análise do estimador. No entanto, diferente do parâmetro h nos estimadores baseados em *kernel*, a escala imposta pelos métodos do tipo $k-$NN são por natureza, adaptável aos dados, de modo que, para um valor fixo de k, resoluções mais finas são usadas em regiões de maior densidade.

A literatura também é repleta de uma grande variedade de estimadores de entropia diferencial baseados em estimadores de densidades do tipo $k-$NN [49, 50]. Em particular, para o propósito de estimação de informação mútua, um trabalho relevante é o de Alexander Kraskov, Harald Stögbauer e Peter Grassberger [51], cujo o estimador proposto — conhecido como estimador KSG — é amplamente usado em aplicações e permanece, até os dias atuais, como um dos mais eficientes para a tarefa, especialmente à medida que a dimensão cresce [52].

2.6.1 O estimador KSG

O estimador KSG, proposto em [51], é baseado em argumentos semelhantes ao estimador de entropia diferencial proposto por Kozachenko-Leonenko, em [49]. Mais especificamente, considere novamente um conjunto $X^N = (x_1, \ldots, x_N)$ de N realizações independentes da variável aleatória $d-$dimensional X. Para um ponto qualquer x, sua verossimilhança $f_X(x)$ pode ser estimada através do estimador $k-$NN por

$$\hat{f}_X(x) = \frac{k}{NV} = \frac{k}{Nc_d r^d}, \tag{2.51}$$

em que c_d é o volume da bola unitária de dimensão d, e r é a distância entre x e o seu k-ésimo vizinho mais próximo em X^N. Note que o volume c_d está associado à escolha de uma métrica, usada como distância entre pares de pontos. Por exemplo, para a norma do supremo temos $c_d = 1$, enquanto que para a norma euclidiana temos $c_d = \frac{\pi^{d/2}}{\Gamma(1+d/2)}$.

A partir deste ponto, consideramos a entropia diferencial em unidades naturais (nats), de modo que objetivo é estimar $-\mathbb{E}[\ln f_X(x)]$, que pode ser aproximada por

$$-\mathbb{E}[\ln f_X(x)] \approx -\mathbb{E}[\ln \hat{f}_X(x)] \tag{2.52}$$

Note que a quantidade de pontos na amostra X^N pertencentes a uma bola de raio unitário centrada em x é uma V.A. do tipo $\text{bin}(N, \hat{f}_X(x)c_d)$, assumindo que a densidade é constante nessa região, e que pode ser aproximada por uma variável de Poisson com parâmetro $N\hat{f}_X(x)c_d$, sempre que N for grande o suficiente e $\hat{f}_X c_d$ for pequeno o suficiente. Dessa forma, o volume mínimo necessário para abranger exatamente k vizinhos de x é modelado como uma V.A.

$$c_d r^d \sim \text{Gama}(k, N\hat{f}_X(x)). \tag{2.53}$$

Logo, a esperança de $\ln(c_d r^d)$ é dada por

$$\ln c_d + d \cdot \mathbb{E}[\ln r] = \psi(k) - \ln N - \ln \hat{f}_X(x), \tag{2.54}$$

e portanto

$$-\ln \hat{f}_X(x) = -\psi(k) + \ln N + \ln c_d + d \cdot \mathbb{E}[\ln r], \tag{2.55}$$

em que ψ é a função digama [4]. Todos os termos no lado direito da Equação 2.55 são constantes, logo

$$-\mathbb{E}[\ln \hat{f}_X(x)] = -\psi(k) + \ln N + \ln c_d + d \cdot \mathbb{E}[\ln r]. \tag{2.56}$$

Finalmente, uma estimativa para a entropia diferencial pode ser obtida substituindo $\mathbb{E}[\ln r]$ na equação acima por uma estimativa empírica dessa média usando a amostra X^N. Para isso, é necessário calcular a distância entre cada observação x_i e o seu k-ésimo vizinho mais próximo dentre as $N-1$ amostras restantes. Assim, obtém-se

$$\hat{h}(X^N) = -\psi(k) + \ln(N-1) + \ln c_d + \frac{d}{N}\sum_{i=1}^{N} \ln r_i, \tag{2.57}$$

em que r_i é a distância entre x_i e o seu k-ésimo vizinho mais próximo em X^N (excluindo o próprio x_i).

[4] A função digama é definida como a derivada logarítmica da função gama. Se $X \sim \text{Gama}(\alpha, \lambda)$, então $\mathbb{E}[\ln X] = \psi(\alpha) - \ln \lambda$.

Uma maneira imediata de estimar a informação mútua $I(X;Y)$ seria estimar separadamente, usando a Equação 2.57, as entropias $h(X)$, $h(Y)$ e $h(X,Y)$ e substituí-las na Equação 2.24.

No entanto, em [51], os autores argumentam que o viés da estimativa na Equação 2.57 se dão, majoritariamente, devido a não uniformidade da densidade f_X na bola centrada em cada ponto x_i, cujos efeitos dependem da resolução usada pelo estimador, que por sua vez é determinada pela distância até o k-ésimo vizinho mais próximo. Dessa forma, as estimativas $\hat{h}(X^N), \hat{h}(Y^N)$ e $\hat{h}(X^N, Y^N)$ teriam vieses muito diferentes, que não se cancelariam quando aplicados na Equação 2.24.

Para evitar isso, os autores usam dois "truques". No primeiro deles, a estimação da entropia diferencial conjunta no espaço $Z = (X, Y)$ é feita usando a distância definida por

$$\|z - z'\| = \max\{\|x - x'\|, \|y - y'\|\}, \tag{2.58}$$

em que qualquer métrica pode ser usada para $\|x-x'\|$ e $\|y-y'\|$. Ao longo do restante desta seção, a distância entre uma amostra z_i e o seu k-ésimo vizinho mais próximo é denotada por r_i, enquanto que a distância entre as projeções desses mesmos pontos nos espaços de X e Y são denotadas por r_{x_i} e r_{y_i}, respectivamente, de modo que $r_i = \max\{r_{x_i}, r_{y_i}\}$.

Dessa forma, uma estimativa para a entropia diferencial conjunta pode ser obtida usando a Equação 2.57, em que $d = d_X + d_Y$ e $c_d = c_{d_X} c_{d_Y}$:

$$\hat{h}(Z^N) = -\psi(k) + \ln(N-1) + \ln c_{d_X} + \ln c_{d_Y} + \frac{d_X + d_Y}{N} \sum_{i=1}^{N} \ln r_i. \tag{2.59}$$

O segundo "truque" é garantir que as contribuições de cada observação para a estimação das entropias marginais serão calculadas usando a mesma resolução da estimação da entropia conjunta. Para isso, note que a estimativa de densidade de probabilidade na Equação 2.48, e consequentemente a estimativa de entropia diferencial na Equação 2.57, não requer que o mesmo k seja usado para todo x.

Assim, é possível estimar as entropias condicionais variando o k, de modo que a escala usada para cada ponto seja a mesma do espaço Z. Especificamente, suponha, sem perda de generalidade, que $r_i = r_{x_i}$ (i.e. $r_{x_i} \geq r_{y_i}$) e que existem n_{x_i} observações que satisfazem $\|x_i - x\| < r_i$. Então, r_i é a distância para o $(n_{x_i} + 1)-$ésimo vizinho mais próximo de x_i e, portanto, pode-se obter

$$\hat{h}(X^N) = -\frac{1}{N} \sum_{i=1}^{N} \psi(n_{x_i} + 1) + \ln(N-1) + \ln c_{d_X} + \frac{d_X}{N} \sum_{i=1}^{N} \ln r_i. \tag{2.60}$$

Quanto ao caso da variável Y, r_i não é exatamente a distância para o $(n_{y_i} + 1)-$ésimo vizinho mais próximo de y_i. Ainda assim, os autores consideram que uma boa

estimativa pode ser obtida através de uma expressão análoga

$$\hat{h}(Y^N) = -\frac{1}{N}\sum_{i=1}^{N}\psi(n_{y_i}+1) + \ln(N-1) + \ln c_{d_Y} + \frac{d_Y}{N}\sum_{i=1}^{N}\ln r_i. \tag{2.61}$$

.

Com isso, pode-se estimar a informação mútua subsituindo as Equações 2.59, 2.60 e 2.61 na Equação 2.24:

$$\hat{I}(X^N;Y^N) = -\frac{1}{N}\sum_{i=1}^{N}(\psi(n_{x_i}+1)+\psi(n_{y_i}+1)) + \psi(K) + \ln(N-1). \tag{2.62}$$

2.7 Informação mútua pontual

Até aqui, foram apresentadas medidas de dependência entre variáveis aleatórias (discretas ou contínuas) baseadas no conceito de informação mútua. Em muitas situações, é de interesse medir a associação entre eventos (subconjuntos do espaço amostral de um experimento aleatório) e não entre variáveis. Nesse contexto, surge o conceito de informação mútua pontual [53], definida como

$$\mathrm{PMI}(x,y) = \log\frac{\Pr(x\cup y)}{\Pr(x)\cdot\Pr(y)}. \tag{2.63}$$

Aqui, a noção de associação entre dois eventos x e y refere-se a uma medida do quanto a ocorrência de um deles informa sobre a ocorrência do outro. Usando a definição de informação mútua pontual, essa associação é quantificada através da comparação entre a probabilidade de co-ocorrência desses dois eventos com o que seria essa probabilidade caso eles fossem eventos independentes. Se eles de fato forem independentes, então $\mathrm{PMI}(x,y) = 0$.

O conceito de informação mútua é amplamente usado em processamento de linguagem natural (*natural language processing*, NLP). Por exemplo, os populares *word embeddings* apelam para a intuição de que palavras semelhantes aparecem em contextos semelhantes [54]. Logo, a forma mais natural de se medir similaridade entre duas palavras é verificando o quão frequente elas co-ocorrem no mesmo contexto comparado ao que seria esperado se elas co-ocorressem por mero acaso [55]. Em NLP, contextos são definidos como janelas de palavras consecutivas em um texto corrido. No entanto, esse princípio pode ser aplicado em outros domínios, como em visão computacional, em que o contexto pode ser definido como uma vizinhança de pixels (um *patch*) [56].

Para finalizar esta passagem, estabelecemos uma conexão entre os conceitos de informação mútua e informação mútua pontual. A saber, se X e Y são duas variáveis aleatórias, então

$$I(X;Y) = \mathbb{E}[\mathrm{PMI}(X=x, Y=y)]. \tag{2.64}$$

2.8 Síntese e implicações para a continuidade do trabalho

Como mencionado anteriormente, a razão de ser deste capítulo está associada ao fato de que, para os estudos apresentados nos capítulos subsequentes, obter estimativas de informação mútua entre pares de variáveis aleatórias é uma etapa obrigatória. Dessa forma, fez-se necessário adotar um estimador para a continuidade do trabalho. No entanto, a partir do estudo conduzido ao longo deste capítulo, é evidente que estimar informação mútua entre variáveis aleatórias é um desafio considerável, tanto no caso discreto, quanto no caso contínuo.

Por exemplo, no caso discreto, foi constatado que a literatura é repleta de métodos para a compensação do viés típica do estimador *plug in*, com um deles, inclusive, sendo resultado deste trabalho. Por outro lado, nenhuma das soluções é considerada universal e funciona em qualquer regime. Para o caso contínuo, o desafio é igualmente delicado, uma vez que envolve uma etapa de estimação de densidades de probabilidade, que também não é trivial.

Ao longo do restante do trabalho, as bases de dados utilizadas são de natureza contínua, ou discretas como consequência de um processo de quantização, a exemplo das imagens digitais. Por esse motivo, escolheu-se trabalhar com o estimador KSG, que se consolidou como um padrão *de facto* para a estimação não paramétrica da informação mútua entre variáveis aleatórias contínuas, em virtude de sua robustez, eficiência computacional e bom desempenho empírico em uma ampla variedade de aplicações [57, 52]. Nos experimentos reportados durante o restante deste trabalho, o parâmetro k vizinhos mais próximos foi fixado em $k = 5$, a menos que se especifique o contrário. Entretanto, foi verificada em alguns dos experimentos uma robustez considerável à escolha dos valores de k, indicando que os resultados finais (dos experimentos nos quais esses testes foram feitos) não são significativamente alterados para valores de k alterados dentro do intervalo entre 3 e 10.

É válido ressaltar também que, embora o problema de estimação de informação mútua entre fontes contínuas seja reconhecidamente complexo, os maiores gargalos surgem quando as variáveis envolvidas são de alta dimensão [52]. Nesse sentido, o presente trabalho opera no regime mais favorável possível, uma vez que todos os métodos aqui utilizados são baseados em medidas dependência entre pares de variáveis unidimensionais, de modo que altas dimensões nunca sejam um problema durante a etapa de estimação de informação mútua.

3 Agregação entre variáveis dependentes

Antes de adentrar em estudos mais formais sobre os efeitos de agregações entre variáveis dependentes no contexto das redes neurais artificiais, o objetivo deste capítulo é ilustrar, através de pequenos exemplos sintéticos, como elas podem produzir efeitos desejáveis no tratamento de conjuntos de dados. Mais especificamente, suponha, por exemplo, que $X_1, X_2, \ldots, X_n$ sejam n variáveis que modelam sensores que medem uma mesma quantidade de interesse Z, e que estão sendo perturbados por ruídos independentes. Nesse caso, se assumirmos ruídos aditivos independentes, temos que $X_i = g_i(Z) + N_i$, em que $N_i \perp N_j$ para todo $i \neq j$[1]. Partimos então da conjectura que agregações da forma $Y = f(X_1, X_2, \ldots, X_n)$ podem produzir representações mais robustas da grandeza latente de interesse Z, através do cancelamento dos ruídos que afetam de maneira independente as representações $X_1, \ldots, X_n$. Esse fenômeno é ilustrado através dos exemplos a seguir.

Exemplo: : Agregação entre variáveis correlacionadas usando análise de componentes principais

Considere um modelo generativo no qual uma V.A. multivariada $X = (X_1, X_2, X_3, X_4, X_5)$, com 5 dimensões, é definida a partir de duas variáveis independentes $Z_1 \sim U(0,1)$ e $Z_2 \sim U(0,1)$:

$$\begin{aligned} X_1 &= -0.6Z_1 \\ X_2 &= 0.5Z_2 \\ X_3 &= 1.5Z_2 \\ X_4 &= 0.1Z_1 \\ X_5 &= 0.5Z_1. \end{aligned} \tag{3.1}$$

Claramente, existem dois grupos de variáveis perfeitamente correlacionadas $\{X_1, X_4, X_5\}$ e $\{X_2, X_3\}$, ao mesmo tempo que a correlação é nula entre quaisquer duas variáveis em grupos diferentes. Além disso, todas as relações de dependência entre variáveis são lineares, fazendo da correlação uma medida ideal de dependência para esse caso.

Para esse exemplo, foi sintetizada uma base de dados $X^N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, em que $\mathbf{x}_i \in \mathbb{R}^5$, com $N = 10000$ instâncias independentes de X. Em seguida, cada uma dessas observações foi corrompida com uma instância independente de um ruído aditivo $N \sim \mathcal{N}(0, 0.2\mathbf{I})$, em que $\mathbf{I}$ é a matriz identidade. Com isso, foi produzida uma nova base de dados $Y^N = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, de modo que

[1] O símbolo $\perp$, nesse caso, é usado para indicar que duas variáveis aleatórias são independentes.

$\mathbf{y}_i = \mathbf{x}_i + \mathbf{n}_i$. A relação sinal-ruído da variável Y pode então ser estimada por

$$\text{SNR} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\mathbf{x}_i\|^2}{\|\mathbf{y}_i - \mathbf{x}_i\|^2}. \tag{3.2}$$

Para a base sintetizada, essa relação sinal ruído foi estimada em aproximadamente 5.24.

Aplicando a análise de componentes principais ao conjunto de dados Y^N, é possível reter pouco mais de 93% da variância total usando duas componentes principais, o que é consistente com o fato de que apenas duas variáveis latentes governam cada observação multivariada, a menos do ruído, através de uma transformação linear. Dessa forma, cada observação $\mathbf{y}_i$ pode ser projetada no $\mathbb{R}^2$ a partir da aplicação de uma matriz $\mathbf{P} \in \mathbb{R}^{2\times 5}$, contendo as duas primeiras componentes principais em suas linhas, ao passo que uma versão filtrada (no sentido em que o ruído é parcialmente removido) de $\mathbf{y}_i$ pode ser reconstruída através da autocodificação $\hat{\mathbf{y}}_i = \mathbf{P}^t\mathbf{P}\mathbf{y}_i$. Truncando os coeficientes da matriz $\mathbf{P}^t\mathbf{P}$ em duas casas decimais, essa transformação pode ser escrita como

$$\begin{aligned}
\hat{Y}_1 &= 0.58Y_1 - 0.11Y_4 - 0.48Y_5 \\
\hat{Y}_2 &= 0.10Y_2 + 0.31Y_3 \\
\hat{Y}_3 &= 0.31Y_2 + 0.9Y_3 \\
\hat{Y}_4 &= -0.11Y_1 + 0.02Y_4 + 0.09Y_5 \\
\hat{Y}_5 &= -0.48Y_1 + 0.09Y_4 + 0.40Y_5.
\end{aligned} \tag{3.3}$$

Como pode ser visto nas equações acima, este processo se resume a um mero conjunto de agregações entre variáveis correlacionadas, e o seu efeito positivo pode ser evidenciado substituindo $\mathbf{y}_i$ por $\hat{\mathbf{y}}_i$ na Equação 3.2, resultando em uma relação sinal-ruído de aproximadamente 12.55. Isto é, as agregações lineares entre variáveis correlacionadas, expressas na Equação 3.3, produziram um aumento de 5.13 para 12.55 na relação sinal-ruído.

De fato, a filtragem de ruídos é uma das principais aplicações da análise de componentes principais, que, em seu nível mais básico, pode ser interpretada sob o ponto de vista de agregação entre variáveis correlacionadas, como ilustrado no exemplo acima.

Exemplo: Agregação versus não agregação

Considere agora um cenário hipotético em que duas constantes desconhecidas z_1 e z_2 são medidas, respectivamente, por k_1 e k_2 canais paralelos que estão impregnados com ruídos independentes e normalmente distribuídos, com média nula e variância unitária. Ou seja, temos k_1 variáveis $X_1, \ldots, X_{k_1} \sim \mathcal{N}(z_1, 1)$

que medem a constante z_1, e k_2 variáveis $Y_1, \ldots, Y_{k_2} \sim \mathcal{N}(z_2, 1)$. que medem a constante z_2.

Com o intuito de estimar o valor de cada medição ruidosa, são tomadas N medições independentes ao longo dos $k_1 + k_2$ canais paralelos. Assim, é possível montar um experimento mental em dois modos. No primeiro modo, não se sabe que há dois subconjuntos de variáveis que compartilham um componente em comum (a média). Nesse caso, a melhor estimativa possível seria a média do conjunto de N medidas de cada canal de medição, de maneira independente. Isso resultaria em um estimador centrado em z_1 para os k_1 canais correspondentes, e em z_2 para os outros k_2 canais. Em todos os casos, a variância do estimador seria $1/N$ por canal.[2]

No segundo modo, são conhecidos os dois conjuntos de variáveis que compartilham a informação de interesse, e, portanto, elas podem ser agregadas por uma simples média. Assim, pode-se usar o fato de que $X_1, \ldots, X_{k_1}$ medem a mesma quantidade, e extrair a média das $k_1 \cdot n$ medições desse conjunto de canais. Da mesma forma, extrai-se a média das $k_2 \cdot n$ medições dos canais $Y_1, \ldots, Y_{k_2}$.

Como resultado, agora temos apenas duas médias empíricas, e a quantidade de dados para cada canal foi multiplicada (por k_1 para os do primeiro subconjunto de variáveis, e por k_2 para os do segundo). Consequentemente, agora, as variâncias dos estimadores do primeiro grupo de canais são reduzidas para $1/(nk_1)$, enquanto as do segundo grupo de canais são reduzidas para $1/(nk_2)$.

Esse exemplo traz luz para uma das principais motivações para estudar a agregação entre variáveis dependentes. Isto é, se duas variáveis compartilham um componente em comum, ao mesmo tempo em que são distorcidas por ruídos independentes, uma agregação adequada entre essas duas variáveis pode reforçar o componente comum entre elas, e descartar parte do ruído. Obviamente, um estimador baseado em redes neurais, treinado de maneira supervisionada, poderia *descobrir* por conta própria essa regularidade nos dados, e explorar os dois grupos de variáveis dependentes sem que restrições manuais fossem impostas. No entanto, também é verdade que, em cenários mais complexos, a imposição de restrições manuais adequadas resulta em ganhos significativos, como é o caso das CNNs, que são o objeto utilizado no próximo exemplo.

Exemplo: Impacto da agregação de variáveis nas redes convolutivas

[2] Como as N observações são independentes, segue que $\mathrm{Var}\left(\frac{X_i^{(1)}+\cdots+X_i^{(N)}}{N}\right) = \frac{1}{N^2}(\mathrm{Var}(X_i^{(1)} + \cdots + \mathrm{Var}(X_i^{(N)})) = \frac{N}{N^2} = \frac{1}{N}$.

Para esse exemplo, são usadas imagens do conjunto CIFAR-10, que contém 60000 imagens de baixa resolução (32×32) separadas em dez classes, sendo 10000 dessas amostras destinadas para teste. Essa base é frequentemente usada em pesquisas nas áreas de aprendizado de máquina e visão computacional [58].

Dentre as 10000 imagens de teste, foram selecionadas as imagens das classes "gato" e "cachorro" (rótulos 3 e 5, respectivamente), resultando em um total de 1953 imagens, das quais 16 delas são apresentadas na Figura 3.

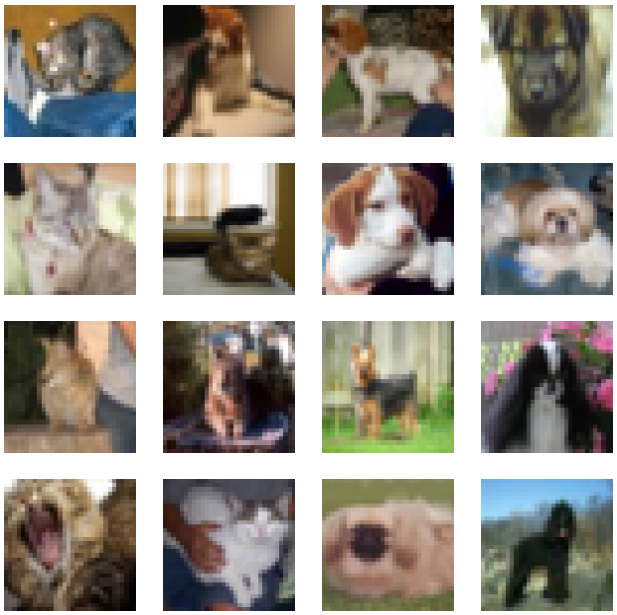
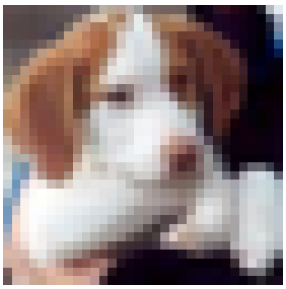


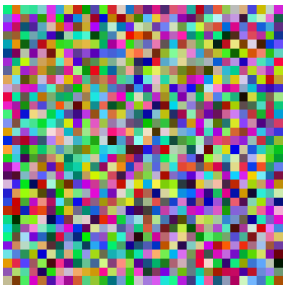
Figura 3 – Ilustração contendo 16 imagens dentre as 1953 selecionadas. As duas primeiras colunas pertencem à classe "gato", enquanto as duas restantes pertencem à classe "cachorro".

Em sequência, foi aplicada uma mesma permutação entre as 3072 variáveis (canais de cores dos 32×32 pixels) de cada uma das 1953 imagens, efetivamente desmontando todas as relações de vizinhança entre os pixels ao longo dos 3 canais de intensidade. A Figura 4 ilustra uma das imagens antes e depois da permutação.

Após isso, as duas versões da base foram transformadas por uma rede neural convolucional. No experimento, foi utilizada uma rede composta por duas camadas convolucionais intercaladas com funções de ativação tangente hiperbólica (Tanh) e camadas de *max pooling*. A primeira camada convolucional recebe imagens de entrada com 3 canais (RGB) e aplica 64 filtros de tamanho 3×3, seguida por uma função de ativação Tanh e uma camada de *max pooling* com janela 2×2 e *stride* 2. Em seguida, a segunda camada convolucional aplica



(a) .



(b)

Figura 4 – (a). Imagem da classe "cachorro" (b). Versão permutada da mesma imagem

48 filtros 3×3 com ativação Tanh, novamente seguida de uma camada de *max pooling* 2×2 e *stride* 2, resultando em saídas de ordem $8 \times 8 \times 48$. Logo, o espaço de características, após a transformação através da CNN, possui dimensão 3072 — a mesma do espaço original. Essa rede foi inicializada com pesos aleatórios, que foram mantidos constantes ao longo de todo o experimento (i.e. não houve nenhum tipo de treinamento).

Com isso, três espaços de características diferentes são obtidos, ambos de mesma dimensão aparente (3072):

- Espaço I: Contém as características originais (as intensidades de cada pixel ao longo dos três canais de cores), sem nenhum tipo de transformação
- Espaço II: Contém as características resultantes da aplicação da CNN nas imagens naturais, como a da Figura 4a.
- Espaço III. Contém as características resultantes da aplicação da CNN nas imagens após a permutação entre as variáveis, como a da Figura 4b.

Finalmente, os padrões em cada um dos três espaços foram usados para treinar um classificador linear, usando a entropia cruzada binária como critério de otimização. Para isso, as bases foram particionadas em 80% para treinamento e 20% para teste (a mesma partição é aplicada nos três espaços).

O experimento foi repetido cem vezes, com pesos diferentes para a CNN e partições diferentes para o classificador linear ao longo das execuções, e os resultados em cada um dos três espaços são apresentados na Tabela 1.

Como mencionado anteriormente, na introdução deste trabalho, as camadas da rede neural usada — tanto as camadas de convolução quanto as camadas de *max pooling* — realizam agregações entre pixels vizinhos.

Dessa forma o exemplo acima coloca em evidência a relevância dessas agregações entre variáveis, ao passo que a separabilidade linear entre as duas classes

	Acurácia
Espaço I	$54.70 \pm 2.25\%$
Espaço II	$65.35 \pm 2.13\%$
Espaço III	$59.45 \pm 2.23\%$

Tabela 1 – Taxa de desempenho do classificador linear em cada um dos três espaços, na base de teste.

é claramente maior nos espaços de características obtidos através da cadeia de agregações, mesmo que estas sejam realizadas com pesos aleatórios, sem nenhum treinamento. Além disso, é possível notar na Tabela 1 que a separabilidade entre classes no Espaço II, no qual as posições originais das variáveis são mantidas na entrada da rede, de modo que as relações de vizinhança sejam preservadas, é ainda maior do que no Espaço III, em que as variáveis são permutadas. Em um certo sentido, isso evidencia mais uma vez os efeitos das agregações entre as variáveis mais dependentes, uma vez que é conhecido e esperado que em imagens naturais existam relações de dependência local entre pixels vizinhos.

Além disso, considerando que, antes da permutação, a maioria das 3072 variáveis possui uma vizinhança imediata de 26 (9+9+8) variáveis, então, após permutação aleatória, a probabilidade de nenhuma das novas 26 variáveis vizinhas de cada variável não ser parte de sua vizinhança original é de aproximadamente 80% ($(1 - 26/3072)^{26}$). Assim, é de se esperar que aproximadamente 20% das vizinhanças originais sejam mantidas após permutação. Essa manutenção de parte das variáveis fortemente dependentes na vizinhança de agregação também deve contribuir para o desempenho intermediário obtido no Espaço III.

No Capítulo 4, as CNNs são estudadas sob essa perspectiva, e a agregação de variáveis dependentes (combinada com o compartilhamento de pesos) é colocada em evidência como suficiente à explicação do sucesso dessas estruturas no reconhecimento de imagens.

Por fim, um pequeno exemplo de exploração de medidas de dependência entre símbolos, e não entre variáveis, é mostrado a seguir.

Exemplo: Agregação entre símbolos dependentes
Considere a imagem formada por 28×28 pixels em níveis de cinza, mostrada na Figura 5.

Essa imagem foi sintetizada através da quantização de uma imagem do caractere 8, da base MNIST [59], em quatro níveis de intensidade: $I_1 = 50$ (mais escuro), $I_2 = 100$, $I_3 = 150$ e $I_4 = 200$ (mais claro). Em seguida, foi feita uma

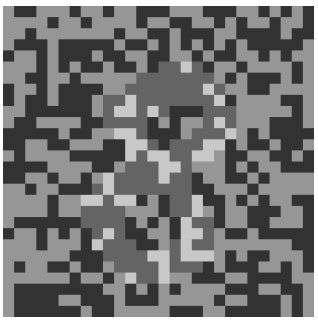


Figura 5 – Imagem após a permutação de intensidades

inversão entre as intensidades I_2 e I_3, em que todos pixels associados à intensidade I_2 foram atribuídos à intensidade I_3, e vice-versa. Ou seja, um ruído foi introduzido à imagem através de um simples *flip* entre duas intensidades subjacentes, de modo que a percepção de algum padrão visual seja dificultada.

No entanto, usando o conceito de informação mútua pontual, podemos mensurar a associação entre os pares de intensidade. Para isso, considere que P é um *patch* 2×2 da imagem, e denotemos por p_i o evento que indica a ocorrência da intensidade I_i em P (i.e. pelo pelo menos um pixel em P tem intensidade I_i). Dessa forma, podemos preencher a matriz

$$\mathbf{M} = \begin{bmatrix} 0.0 & 0.4973 & 1.1341 & 0.5219 \\ 0.4973 & 0.0 & 0.3359 & 3.3684 \\ 1.1341 & 0.3359 & 0.0 & 0.3533 \\ 0.5219 & 3.3684 & 0.3533 & 0.0 \end{bmatrix}, \tag{3.4}$$

em que $M_{ij} = e^{\text{pmi}(p_i, p_j)}$. Aqui, a exponencial cumpre o papel de converter as medidas de informação mútua pontual para uma escala não logarítmica. Ou seja M_{ij} mede a associação entre I_i e I_j. Observando a matriz $\mathbf{M}$, é possível notar que existem dois conjuntos disjuntos de intensidades fortemente associadas: $\{I_1, I_3\}$ e $\{I_2, I_4\}$.

Dessa forma, pode-se agregar os dois grupos de intensidade em um único representante para cada, através dos valores médios $\hat{I}_1 = \frac{I_1+I_3}{2}$ e $\hat{I}_2 = \frac{I_2+I_4}{2}$, resultando na imagem apresentada na Figura 6, em que se nota claramente o padrão familiar de um dígito manuscrito.

Ou seja, uma simples agregação entre símbolos fortemente associados (agrupamento, ou *clustering* de pontos que representam intensidades) possibilitou a segmentação entre o ruído de fundo e o objeto alvo.

Dessa forma, as quatro ilustrações apresentadas neste capítulo reforçam a conjectura que há uma relação a ser estudada entre agregações de variáveis dependentes

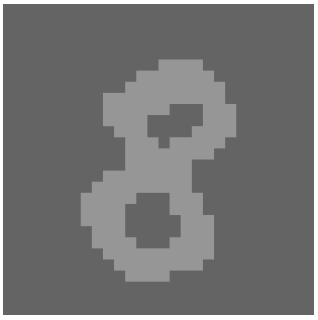


Figura 6 – Imagem após a agregação de intensidades

(ou símbolos dependentes) e a construção de representações mais robustas para padrões multivariados.

4 Agregação de variáveis nas redes convolutivas

4.1 Introdução do capítulo

Redes neurais convolucionais, ou, simplesmente, redes convolutivas, são um tipo especial de rede neural artificial designado para processar dados estruturados em um grid. Nesse contexto, a maior parte dos exemplos mais emblemáticos de aplicações destes modelos estão na área de visão computacional, uma vez que imagens podem ser interpretadas como funções definidas no espaço euclidiano bidimensional e amostradas em um grid.

De fato, as arquiteturas convolutivas são altamente inspiradas pelo processamento que ocorre no cortéx visual[1], como descrito no trabalho seminal de Hubel e Wiesel [60]. Uma das primeiras manifestações dessa inspiração biológica foi apresentada no trabalho de Fukushima, em [61], que propôs um modelo de aprendizado auto-organizado — sem a necessidade de um conjunto de imagens rotuladas para treinamento — chamado *Neocognitron*, no qual boa parte dos elementos que mais tarde guiaram o desenvolvimento das CNNs já podiam ser encontrados. Estas, por sua vez, foram introduzidas, e nomeadas, em seu formato moderno pela primeira vez por Yann Lecun, em 1989 [62]. Tal desenvolvimento foi acompanhado por uma sequência de trabalhos ao longo dos anos 90 que consolidaram as CNNs como o estado da arte na tarefa de reconhecimento de dígitos manuscritos [63, 64], e que eventualmente culminaram no desenvolvimento da arquitetura conhecida como LeNet5, em 1998 [59].

Apesar do sucesso na classificação de caracteres manuscritos, os anos subsequentes foram marcados por resultados decepcionantes em experimentos envolvendo redes neurais artificiais profundas e por uma retomada de protagonismo por parte de modelos baseados em extração manual de características e métodos clássicos para tarefas de reconhecimento de imagens [65]. No entanto, o interesse pelas arquiteturas profundas, incluindo, principalmente, as CNNs, foi renovado com a disponibilização de grandes volumes de dados e de *hardware* computacional igualmente poderoso para processar tais volumes. Em particular, um marco dessa retomada se deu em 2012, quando a arquitetura conhecida como AlexNet venceu o desafio anual da ImageNet por uma ampla margem [66]. Após isso, o desafio foi vencido por uma arquitetura convolutiva em todos os anos subsequentes (até ser descontinuado, em 2017), diversas novas arquiteturas foram propostas e as CNNs se tornaram, por muito tempo, o padrão-ouro em essencialmente todas as tarefas de visão

[1] O córtex visual é a região cerebral responsável por processar informações visuais, transformando sinais elétricos da retina em imagens e interpretando atributos como forma, cor, profundidade e movimento.

computacional e síntese de imagens [67, 68, 69][2]

É importante ressaltar, no entanto, que embora as imagens, que são dispostas em grids bidimensionais, sejam o exemplo mais emblemático, as redes convolutivas podem ser usadas para processar dados estruturados em grids de dimensão arbitrária, e foram aplicadas com sucesso em muitos outros domínios. Por exemplo, séries temporais podem ser interpretadas como sinais amostrados em um grid unidimensional, e nuvens de pontos podem ser representadas como sinais definidos em um grid tridimensional. Isso se dá pois as redes convolutivas exploram a estrutura espacial inerente à organização desses dados, incluindo as noções de localidade e vizinhança, ao contrário dos modelos densos em que cada instância é tratada como uma lista de atributos organizada em um vetor, ordenada de maneira fixa porém arbitrária.

Para o caso específico das imagens naturais, podemos sempre recorrer a expectativa de que pixels vizinhos apresentem forte dependência estatística entre si, e de que essas relações de dependência se tornem mais fracas ao compararmos pares de pixels mais distantes no domínio espacial. De fato, essa observação indica que as estruturas locais possuem padrões de interesse e é frequentemente apontada nos livros didáticos como parte da justificativa para o uso de estruturas convolutivas [70].

Neste capítulo da dissertação, a importância da existência de dependências locais e da agregação entre variáveis dependentes é colocada em evidência como uma componente fundamental ao sucesso das redes convolutivas. Nesse contexto, a investigação proposta consiste em avaliar o quão grande é o impacto das relações de dependência entre pares de pixels, e se o conhecimento dessas dependências é, por si só, suficiente para substituir o conhecimento do arranjo espacial dos pixels, tido como a estrutura fundamental a ser explorada por uma rede convolutiva. Mais especificamente, a seguinte questão é levantada: em um cenário hipotético, no qual o arranjo de pixels 2D em uma base de imagens é desconhecido, é possível alcançar desempenhos na tarefa de classificação comparáveis àqueles que seriam obtidos por uma CNN treinada com a base original, na qual esse arranjo é fornecido *a priori*?

Para responder tal pergunta, são desenvolvidos procedimentos algorítmicos e experimentais, cujos resultados apontam para duas conclusões que podem parecer surpreendentes, a saber:

(i) Usando a dependência entre variáveis, é possível recuperar, com um bom nível de fidelidade, a topologia 2D (assumida desconhecida) das imagens originais.

(ii) Mesmo quando o arranjo de pixels em um grid 2D não é fornecido *a priori*, o

[2] Atualmente, as CNNs continuam sendo a abordagem dominante na área. No entanto, nos últimos anos, elas competem em desempenho com os *Transformers*, que passaram atingir resultados expressivos em uma quantidade considerável dessas tarefas.

mapeamento das dependências entre variáveis permite usufruir dos benefícios da aplicação das CNNs em imagens, quando comparado às redes do tipo MLP.

Dessa forma, tais resultados também sugerem que a noção de dependência (e de agregação de variáveis dependentes) é potencialmente anterior às noções de vizinhança e localidade em imagens naturais, e, portanto, são usados como uma prova de conceito para dar suporte às hipóteses que foram levantadas na introdução deste trabalho, acerca dos benefícios de impor a agregação entre variáveis dependentes nas redes neurais artificiais.

O restante do capítulo está organizado da seguinte forma. Na Seção 4.2, os principais blocos constituintes das redes neurais convolucionais modernas são relembrados, bem como as motivações que justificam o seu uso, enquanto a Seção 4.3 menciona trabalhos anteriores encontrados na literatura que se relacionam com as investigações aqui propostas. Na Seção 4.4 é proposto um algoritmo para reorganizar o arranjo de pixels de uma base de imagens quando o arranjo original não é conhecido. A Seção 4.5 introduz uma discussão sobre a dimensão ideal para o arranjo de pixels. Resultados experimentais são reportados na Seção 4.6.

4.2 Elementos fundamentais das CNNs

4.2.1 Camadas convolucionais

O principal bloco constituinte das redes convolutivas, comum à todas as arquiteturas, são as camadas convolucionais. Curiosamente, no entanto, a operação matemática usada nas camadas convolucionais não corresponde precisamente à convolução, da maneira como esta é definida nas áreas de engenharia e matemática pura. Na verdade, a operação que é realmente implementada nas redes neurais artificiais é chamada de correlação cruzada, que, para o caso dos sinais discretos bidimensionais (como as imagens) é definida por

$$S[i,j] = (I \star K)[i,j] = \sum_m \sum_n I[i+m, j+n] K[m,n]. \tag{4.1}$$

No contexto do aprendizado profundo, é comum referir-se ao primeiro argumento I como a entrada da camada convolucional e ao segundo argumento K como o *kernel*. A saída S, por sua vez, é tipicamente chamada de mapa de características. A diferença está no rebatimento do *kernel*, uma vez que, na convolução, $K[m,n]$ seria substituído por $K[-m,-n]$. Matematicamente, o efeito do rebatimento está no fato de que a convolução é uma operação comutativa, enquanto a correlação cruzada não possui a mesma propriedade. No entanto, a comutatividade não é relevante no contexto das redes neurais

convolucionais, justificando o uso da correlação cruzada.[3] A Figura 7 ilustra visualmente a aplicação dessa operação.

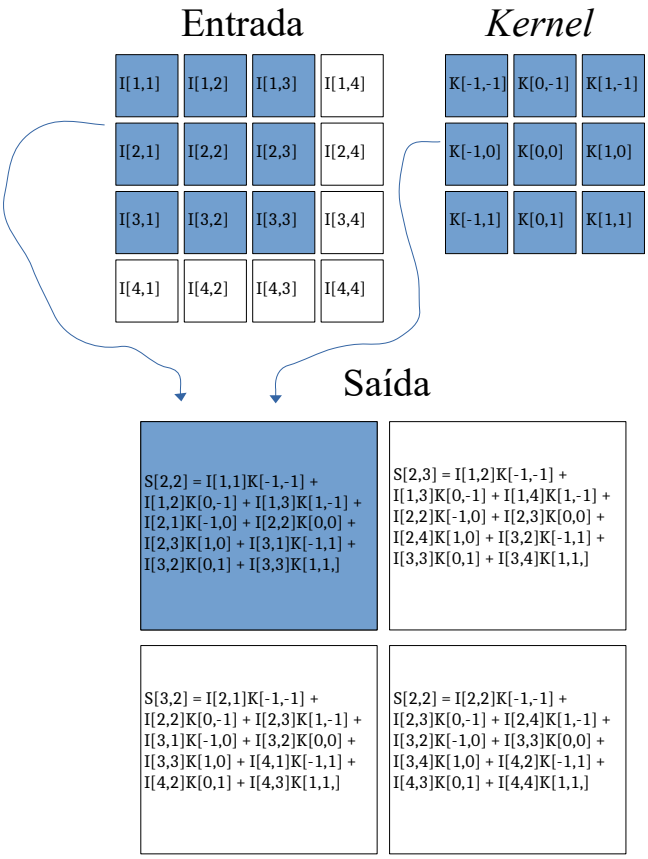


Figura 7 – Um exemplo da aplicação da correlação cruzada. As unidades pintadas em azul e as setas servem para indicar como a unidade $S[2,2]$ da saída é composta pelo produto escalar entre o *kernel* e o *patch* centralizado em $I[2,2]$ na entrada. Aqui, a saída foi restrita às posições em que o *kernel* se sobrepõe completamente à entrada. Em grande parte das implementações, no entanto, é comum que as bordas da entrada sejam preenchidas com zero para garantir que saída possuirá as mesmas dimensões.

Como pode ser visto, a correlação cruzada pode ser melhor entendida visualmente como uma operação na qual o *kernel* escaneia toda a entrada. Em cada passo, ele se sobrepõe com uma porção da entrada — um *patch* — e o resultado do produto escalar entre ambos corresponde a uma unidade na saída. Ou seja, cada uma dessas unidades corresponde a uma medida de similaridade entre o *kernel* e o *patch* correspondente na entrada. Dessa forma, o *kernel* pode ser interpretado como uma espécie de *template*, e a correlação cruzada pode ser interpretada como uma espécie de busca, ao longo da entrada, por padrões que se assemelhem a este *template*.

Em uma camada convolucional, após a aplicação da correlação cruzada, a saída é tipicamente somada com uma constante b, chamada de viés, ou *bias*, que é compartilhada

[3] É comum em diversas referências na área que a correlação cruzada seja chamada de convolução, apesar do desencaixe com a definição formal.

por todas as unidades do mapa de saída, e em seguida aplicada em uma não-linearidade φ (a função de ativação), sendo a $ReLU$ a mais comum delas.

Dessa forma, sob o ponto de vista deste trabalho, cada neurônio da saída é composto por uma agregação linear entre a unidade de entrada correspondente e os seus vizinhos, seguido pela adição de uma constante e aplicação de uma não-linearidade. Assim, em um certo sentido, é de se esperar que as camadas convolutivas constituam estruturas que impõem a agregação entre variáveis dependentes, como proposto neste trabalho originalmente. Uma propriedade interessante induzida pelo uso da correlação cruzada é o fato de que essas agregações se dão a pesos compartilhados. Isto é, os coeficientes da combinação linear entre as variáveis de entrada correspondentes são os mesmos para todas as unidades da saída.

Uma última observação relevante é a de que, se fixarmos o *kernel* k, a correlação cruzada é um operador linear em I e, portanto, pode ser expresso por uma multiplicação matricial. Para o caso bidimensional, a matriz associada a esta transformação é uma matriz bloco-circulante, na qual diversos pares de elementos estão restritos a serem iguais, e várias outras posições estão restritas a serem nulas quando o tamanho do *kernel* é muito menor que o do sinal de entrada [70]. Logo, qualquer função implementada por uma CNN pode ser também realizada por uma rede neural densa, mas a recíproca não é verdadeira. Ou seja, a escolha por uma CNN implica uma restrição no espaço de funções possíveis, configurando um viés no treinamento da rede. Na Seção 4.2.3, as razões pelas quais a imposição de tais restrições são tão eficazes em determinadas aplicações são elencadas.

4.2.2 Camadas de *pooling*

Outro bloco comumente utilizado nas CNNs são as camadas de *pooling*. Nessas camadas, o sinal de entrada também é escaneado usando janelas deslizantes, e cada unidade na saída corresponde a uma estatística que sumariza a janela correspondente na entrada. As estatísticas mais comuns são o máximo (*max pooling*) e a média (*average pooling*).

Um artifício frequentemente incorporado nas camadas de *pooling* é que o sinal de entrada seja escaneado de modo que não haja sobreposição entre janelas consecutivas (*pooling* com subamostragem), como ilustrado na Figura 8.[4].

Como pode ser visto, nas camadas de *pooling*, cada unidade da saída é resultado de uma agregação entre unidades vizinhas no sinal de entrada, tal qual nas camadas convolutivas. No entanto, nas camadas convolutivas, as agregações realizadas são compostas por combinações lineares a pesos ajustados durante o treinamento da rede, enquanto nas camadas de *pooling* não há parâmetros a serem ajustados.

[4] Embora menos comum, esse artifício também pode ser incorporado nas camadas convolutivas. O parâmetro que controla o tamanho do passo entre duas janelas (i.e. entre dois *patches*) é chamado de *stride*.

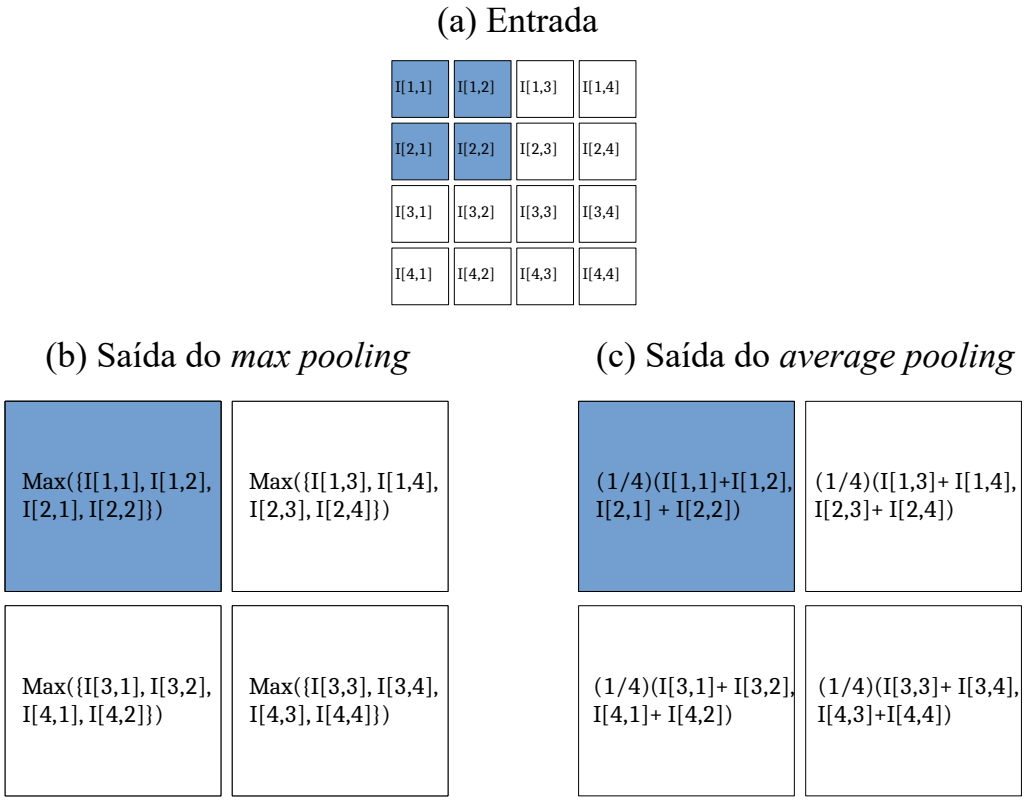


Figura 8 – Ilustração das operações de *pooling*

Na maioria das arquiteturas, é comum que camadas convolucionais sejam intercaladas com camadas de *pooling*. Quando o objetivo final é o de classificação, é comum também que a saída da última camada de convolução (ou *pooling*) seja usada como entrada para uma rede densa, cuja saída indica a classe predita.

4.2.3 Motivações

Nas seções 4.2.1 e 4.2.2 foram apresentadas breves ilustrações dos principais elementos arquitetônicos encontrados nas redes convolutivas. Um aspecto particularmente ressaltado foi o fato de que tanto as camadas convolucionais quanto as camadas de *pooling* são compostas por operadores orientados a vizinhança. Isto é, o valor de cada célula na saída é calculado através de uma agregação — paramétrica no caso das camadas convolucionais, e não-paramétrica no caso das camadas de *pooling* — de um conjunto de variáveis de entrada em uma vizinhança.

É importante ressaltar, entretanto, que essa apresentação omitiu alguns aspectos importantes da implementação prática dessas operações nas redes neurais, como, por exemplo, as extensões associadas ao parâmetro que controla o passo da janela deslizante (o *stride*) e o fato de que os sinais de entrada e de saída em cada camada possuem múltiplos canais, uma vez que nas ilustrações acima apenas sinais com um único canal foram

considerados. Além disso, não foram mencionados outros elementos tipicamente usados nas CNNs modernas, como estratégias de regularização (e.g., *dropout* [71]) e técnicas de normalização (e.g., camadas de *batch normalization* [72]). O leitor interessado em revisitar tais conceitos e os seus detalhes de implementação é direcionado para [73].

O restante desta seção se concentra em listar as principais razões, frequentemente encontradas em materiais didáticos, que justificam a sinergia entre as arquiteturas convolutivas (incluindo múltiplas camadas convolucionais e de *pooling*, em sequência) e as tarefas de reconhecimento de padrões envolvendo imagens (e.g., classificação, segmentação, detecção de objetos, super-resolução). Os principais argumentos podem ser sintetizados conforme a seguinte lista:

(i) **Eficiência estatística:** Nas redes neurais densas, cada camada é composta por uma multiplicação matricial entre o padrão de entrada e uma matriz de parâmetros, de modo que todas as variáveis de saída interagem com todas as variáveis de entrada, com cada interação sendo descrita por um parâmetro separado. Por outro lado, nas camadas convolucionais, cada variável de saída interage com um número restrito de variáveis de entrada, que frequentemente é muito menor que a quantidade total de variáveis de entrada (desde que as dimensões do *kernel* sejam menores que as dimensões da entrada, como normalmente é o caso). Além disso, o compartilhamento de pesos induzido pela correlação cruzada impõe restrições adicionais, de modo que a combinação dessas duas propriedades reduz drasticamente a quantidade de parâmetros do modelo. Mais especificamente, em uma camada densa com uma entrada M-dimensional e uma saída N-dimensional, a quantidade de parâmetros livres, desconsiderando o vetor de vieses, é $M \cdot N$. Em contrapartida, em uma camada convolucional, com entradas e saídas de mesma dimensão, essa quantidade é reduzida para $\mathcal{O}(k)$, em que k é a dimensão dos *kernels*. Logo, as redes convolutivas são significativamente mais eficientes que as densas, tanto em termos de memória quanto em termos estatísticos, uma vez que a tendência do modelo a se *sobreajustar* aos dados de treinamento é muito menor devido à restrição do espaço de soluções.

(ii) **Viés indutivo espacial:** Diferente das redes densas, as CNNs não são alheias à estrutura espacial inerente às imagens. Ou seja, não se trata apenas de uma redução qualquer do número de parâmetros, mas uma que induz a função aprendida durante o treinamento a explorar esse arranjo espacial. Em particular, elas exploram o *a priori* de que as características de maior interesse para a análise das imagens estão nas estruturas locais, uma vez que a correlação cruzada com um *kernel* fixo pode ser vista como uma varredura do sinal de entrada em busca por um determinado *template*. Isto é, as restrições mencionadas no item (i) impõem um viés extremamente forte, reduzindo o espaço de soluções a funções que exploram a detecção de estruturas

locais.

(iii) **Aprendizado hierárquico de características:** Embora nas camadas convolucionais cada unidade seja resultado da agregação entre variáveis em uma pequena região da entrada, nas camadas mais profundas essas unidades interagem, indiretamente, com uma porção maior da entrada da rede (e não da camada em questão). Tal efeito é amplificado devido à subamostragem que ocorre com as operações de *pooling*. Esse tipo de análise multiescala faz com que esses modelos sejam capazes de descrever relações complicadas de interação entre muitas variáveis a partir de pequenos blocos que descrevem interações mais simples. De fato, diversos trabalhos na literatura observaram que os filtros das primeiras camadas tendem a detectar características simples como bordas, cores e texturas, enquanto as camadas seguintes combinam essas informações para detectar características progressivamente mais abstratas, como formas e objetos [74].

(iv) **Equivariância a translações:** O compartilhamento de pesos induzido pela correlação cruzada resulta nesta propriedade, implicando que, sempre que a entrada de uma camada convolucional passar por uma translação espacial, o efeito correspondente na saída será a mesma translação. Isto é, se g é uma função que translada o sinal de entrada, então $S = I \star K \implies g(S) = g(I) \star K$. Isso é particularmente útil pois garante que as características de interesse serão detectadas independente de sua localização na entrada. Por exemplo, ao processar imagens, se um determinado *kernel* se especializa em detectar bordas verticais, e essas bordas podem aparecer em diferentes regiões da imagem, então o compartilhamento de parâmetros é uma prática desejável.

(v) **Invariância a pequenas translações:** As camadas de *pooling* contribuem para que as representações internas aprendidas por uma CNN sejam aproximadamente invariantes a pequenas translações. Isso acontece pois, se a entrada passar por uma pequena translação, os valores de grande parte das unidades de saída nas camadas de *pooling* não são alterados. Essa propriedade contribui para a robustez e para a generalização do modelo, uma vez que as imagens de entrada são mapeadas em representações aproximadamente equivalentes, mesmo após pequenas translações que não alteram o seu conteúdo semântico.

Como pode ser visto, com exceção do item (i), todos os argumentos que justificam o sucesso das CNNs em tarefas envolvendo imagens estão diretamente associados à existência de uma estrutura espacial, e à exploração desta por parte do modelo. Por esse motivo, o entendimento predominante na área é de que, sem o conhecimento prévio da estrutura espacial, as CNNs perdem o seu viés indutivo e a sua aplicação leva a resultados decepcionantes [75]. Em contrapartida, a investigação aqui proposta mostra

que, usando noção de agregação entre variáveis dependentes, esse viés indutivo pode ser aprendido através dos dados, ao invés de ser fixo e dependente do conhecimento *a priori* da estrutura espacial.

4.3 Trabalhos relacionados

O trabalho que mais se aproxima da investigação proposta neste capítulo é o de Le Roux et al., em [76], que também se propõe a mostrar que a ordenação dos pixels pode ser recuperada a partir de exemplos usando medidas de dependência entre pares de variáveis. Algumas diferenças algorítmicas podem ser encontradas em relação ao processo de reorganização (comparado ao utilizado neste trabalho), tais como o uso de estatísticas da correlação de Pearson como medida de dependência, em vez das medidas de informação mútua, a forma de mapear as intensidades de cores no grid, e o uso do ISOMAP como método de projeção, em vez do t-SNE. Essa última diferença se mostrou relevante, uma vez que durante os experimentos realizados, a reconstrução obtida usando o ISOMAP não foi satisfatória, pelo menos para as bases de dados aqui utilizadas.

No entanto, a principal diferença do estudo proposto neste capítulo para aquele apresentado em [76] está na formulação da questão em relação a possibilidade de obter desempenhos equivalentes àqueles de uma CNN quando o arranjo original de pixels não é conhecido, e, consequentemente, nos experimentos envolvendo CNNs treinadas e a interpretação destas sob a perspectiva da agregação de variáveis dependentes. Adicionalmente, em [76], a discussão sobre a dimensão intrínseca ideal para o arranjo de pixels não foi considerada.

Uma ideia muito parecida também foi perseguida por Mahajabin Rahman e Ilya Nemenman, em um trabalho recente, no qual o problema de reconstrução do arranjo de pixels em imagens embaralhadas a partir de medidas de correlação também foi considerado [77]. Nesse caso, os autores estavam interessados no problema de restringir quais interações entre variáveis são permitidas na modelagem de sistemas complexos nos quais o número de variáveis envolvidas é maior que o número de observações. Para isso, imagens sintéticas nas quais as relações entre variáveis se assemelham àquelas de imagens naturais foram utilizadas como uma espécie de exemplo brinquedo para um princípio muito mais geral envolvendo sistemas complexos.

Os resultados apresentados nesses dois trabalhos indicam que as estatísticas lineares são suficientes, no caso das imagens, para a reconstrução espacial do arranjo de variáveis e a inferência das estruturas locais. No entanto, este trabalho mostra que as medidas de informação mútua também produzem o mesmo efeito, o que embora pareça ser óbvio do ponto de vista estritamente conceitual, é relevante do ponto de vista das dificuldades associadas ao processo de estimação, o que abre a porta para sua aplicação

em casos nos quais as interações não lineares são não desprezíveis.

O uso de informação mútua para quantificar dependências entre pares de variáveis e mapeá-las em um grid, por sua vez, foi explorado em [19, 78] para criar representações geométricas das componentes (idealmente independentes) obtidas usando o ICA, com o propósito, principalmente, de visualização das relações de dependências residuais.

No contexto da pesquisa acerca das CNNs, um trabalho relacionado é o de Ivan [75]. O autor realiza um estudo comparativo entre as acurácias de CNNs treinadas em imagens naturais e versões permutadas das mesmas, confirmando a queda de desempenho esperada como resultado da descaracterização das estruturas locais. Além disso, foi mostrado que, para o caso das imagens permutadas, uma variante das CNNs baseada na operação de convolução dilatada produz resultados superiores [79], devido à capacidade de agregar dependências de longo alcance. No entanto, diferentemente deste trabalho, o estudo comparativo proposto em [75] não considera a reconstrução do arranjo de pixels baseada em medidas de dependência. Até aqui, não foram encontradas publicações que buscam quantificar explicitamente o quanto do desempenho perdido com a permutação dos pixels pode ser recuperado através da agregação entre variáveis dependentes.

Em [56], a aplicação dos *transformers* em tarefas de classificação de imagens foi introduzida, através de uma arquitetura chamada *Vision Transformers* (ViT). Devido a construção dos *transformers*, que é baseada na definição de grandes contextos, essa arquitetura permite a agregação de variáveis espacialmente distantes. Ao contrário das CNNs, em que as agregações estão explicitamente restritas às pequenas vizinhanças, nos *transformers*, o mecanismo de atenção é responsável por determinar implicitamente, durante o treinamento, quais as agregações são relevantes dentro de um grande conjunto de variáveis [80]. Isto é, os *transformers* não possuem os viéses indutivos inerente às CNNs, como a extração local de características e a equivariância a translações. Em vez disso, os resultados obtidos em [56] revelam que os viéses podem ser aprendidos diretamente através dos dados, desde que estes estejam disponíveis em grandes volumes, para treinamento, como reconhecido pelos próprios autores em [56]. Em contrapartida, o trabalho aqui apresentado busca mostrar que a determinação de quais agregações são relevantes, independentemente da distância espacial, pode ser aprendida a partir de volumes modestos de dados, através do mapeamento explícito de dependências entre variáveis, mesmo quando a topologia 2D não é conhecida *a priori*.

Similarmente aos *transformers*, o trabalho apresentado em [81] também obtém resultados compatíveis com o estado da arte em *benchmarks* de classificação de imagens sem o uso explícito do viés indutivo das convoluções. Especificamente, os autores propõem uma arquitetura chamada *Mlp-mixer* que combina, paralelamente, dois tipos de blocos baseados em redes densas: um deles proporciona a agregação de *patches* não sobrepostos ao longo de toda a imagem, independente da distância espacial, enquanto o outro proporciona

a agregação entre variáveis restritas a um mesmo *patch*. Entretanto, o sucesso de tal arquitetura também está associado ao conhecimento *a priori* e a exploração da estrutura espacial das imagens, diferente do que é assumido neste trabalho.

4.4 Abordagem proposta

Para iniciar a investigação, consideremos a tarefa hipotética de desenvolver um classificador para uma base de imagens sem que o arranjo natural dos pixels em um grid 2D seja previamente conhecido. Especificamente, consideramos o caso em que temos acesso a uma base de imagens na qual a posição dos pixels foi submetida a uma permutação fixa (i.e. a mesma permutação foi aplicada em todas as imagens da base), porém desconhecida, como ilustrado na Figura 9.

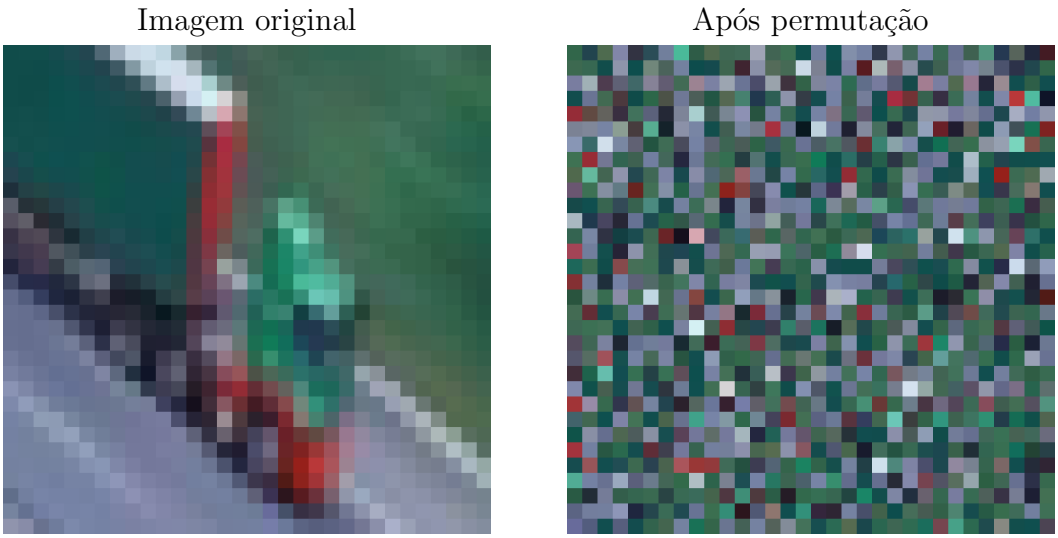


Figura 9 – Ilustração do efeito do embaralhamento dos pixels em uma imagem do conjunto CIFAR-10.

É de se esperar que o treinamento de uma CNN neste cenário resulte em desempenhos inferiores àqueles que seriam obtidos caso a mesma arquitetura fosse treinada usando a base original, uma vez que todas as estruturas locais são essencialmente descaracterizadas e, portanto, os pressupostos que justificam o uso das CNNs deixam de ser válidos com a permutação aleatória das posições dos pixels. De fato, tal expectativa já foi confirmada em estudos anteriores [75].

Por outro lado, se recorrermos à noção de agregação entre variáveis dependentes e às hipóteses levantadas ao longo deste trabalho, é natural sugerir que uma forma eficiente de melhorar o desempenho é reorganizar os pixels em um grid bidimensional, com as mesmas dimensões das imagens, de modo que pares de pixels que apresentam forte dependência estatística sejam mapeados em células vizinhas.

Para tal, o primeiro passo consiste em quantificar o grau de dependência estatística entre todos pares de pixels possíveis. Como mencionado anteriormente, neste trabalho, isso é feito usando estimativas de informação mútua, obtidas usando um conjunto de realizações conjuntas independentes de cada par. Mais especificamente, é usada a versão normalizada da informação mútua, conforme definida na Equação 2.25, de modo que a dependência entre dois pixels quaisquer seja mensurada por um valor entre 0 e 1, em que 1 indica uma relação determinística entre as duas variáveis e 0 indica que elas são independentes. Para a estimação, as imagens (embaralhadas) foram convertidas de RGB para escala de cinza, de modo a permitir trabalharmos com a informação mútua entre duas variáveis unidimensionais.

Dessa forma, dado um conjunto de imagens $M \times N$, em que os pixels estão dispostos de maneira arbitrária, cada posição no grid (i.e. cada pixel) é indexada por um inteiro entre 1 e MN, também de maneira arbitrária. Assim, para todo par de pixels (i, j), em que $i \neq j$ e $i, j \in \{1, \ldots, MN\}$, as imagens separadas para treinamento são usadas para extrair um conjunto correspondente de realizações conjuntas, que por sua vez é usado como entrada para o estimador KSG, conforme descrito na Seção 2.6.1. Com isso, é possível preencher uma matriz $\mathbf{C}$, em que a entrada C_{ij} é um valor entre 0 e 1 referente a uma estimativa da informação mútua normalizada entre as variáveis aleatórias correspondentes ao i-ésimo e ao j-ésimo pixel. Por definição, a diagonal principal da matriz $\mathbf{C}$ pode ser preenchida com 1's, uma vez que a relação entre uma variável e ela mesmo é determinística. Além disso, $\mathbf{C}$ deve ser simétrica, então, uma vez que a C_{ij} for obtida, não é necessária uma nova estimação de informação mútua para obtenção da entrada C_{ji}. Uma ilustração do mapa obtido usando as amostras de treino do conjunto CIFAR-10 é apresentada na Figura 10.

As entradas da matriz $\mathbf{C}$ estão associadas a uma noção de proximidade entre pares de pixels, uma vez que o objetivo é projetar pixels que possuem alta dependência em posições vizinhas no grid. Da mesma maneira, se cada entrada em $\mathbf{C}$ for submetida a uma transformação monotonicamente decrescente, ela pode ser convertida em uma matriz $\mathbf{D}$ cujas entradas emulam a noção de distância entre pares de pixels. Uma propriedade desejável desse mapa é que $C_{ij} = 1 \Rightarrow D_{ij} = 0$ (pois a distância de um pixel para si próprio é nula) e $C_{ij} \to 0 \Rightarrow D_{ij} \to \infty$ (i.e. dois pixels independentes devem ocupar posições distantes na imagem). Um mapa que satisfaz essas condições, e que é escolhido neste trabalho, é dado por

$$D_{ij} = -\ln(C_{ij}). \tag{4.2}$$

O gráfico apresentado na Figura 11 mostra o valor médio de D para cada valor de distância espacial entre pares de pixels.

Como pode ser visto, para pequenas distâncias (para aquelas menores do que dez, sobretudo), existe uma notável tendência de crescimento linear do valor médio de D em

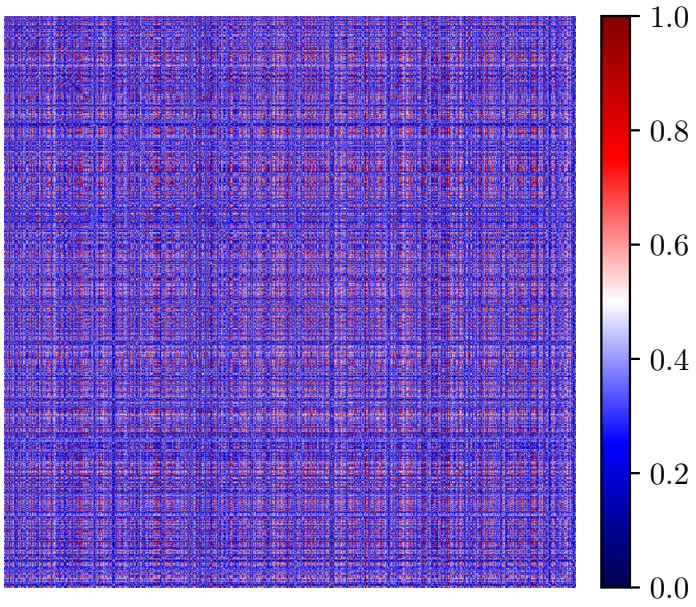


Figura 10 – Ilustração do mapa de informação mútua normalizada estimado usando as imagens do conjunto de treinamento do CIFAR-10. A dimensão do mapa é 1024×1024, uma vez que a resolução das imagens do CIFAR-10 é 32×32.

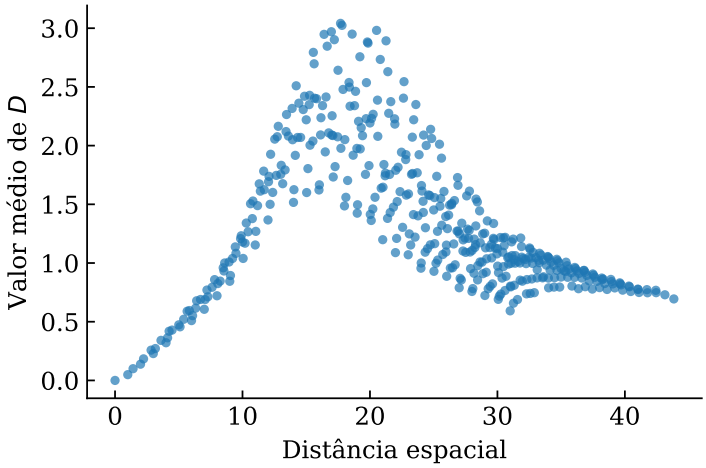


Figura 11 – Relação entre cada possível distância espacial entre pares de pixel e o valor médio de D para todos pares a essa distância.

função da verdadeira distância espacial entre os pixels. É importante ressaltar, no entanto, que os valores das verdadeiras distâncias entre os pixels são assumidos desconhecidos ao longo desta investigação, e, portanto, não devem ser usados durante o processo reorganização do arranjo. Não obstante, essa observação sugere que ao projetar cada pixel no $\mathbb{R}^2$ de modo que a distância entre as projeções do pixel i e do pixel j seja, aproximadamente, igual a D_{ij}, as estruturas locais e as noções de vizinhança características das imagens originais (anteriores às permutações) seriam restauradas, ao menos parcialmente.

A realização de tal projeção pode ser formulada como um problema de escalonamento multidimensional — *multidimensional scaling*, MDS —, em uma de suas variantes

tradicionais [82, 83, 84]. Isto é, o de encontrar uma configuração de pontos $\mathbf{x}_1, \ldots, \mathbf{x}_{MN} \in \mathbb{R}^2$ tal que as distâncias $\|\mathbf{x}_i - \mathbf{x}_j\|$ se aproximem de D_{ij} tão bem quanto possível.[5]

Entretanto, as variantes mais clássicas do MDS buscam a melhor preservação possível da estrutura global, de modo que a função de custo possa ser facilmente dominada pelas maiores distâncias [85], enquanto a preservação das estruturas locais parece ser mais interessante para os propósitos deste trabalho, uma vez que estas são mais relevantes do ponto de vista das redes convolutivas, que funcionam à base de agregações locais entre variáveis vizinhas.

Felizmente, a ideia de priorizar a preservação de estruturas locais já foi explorada em diversos algoritmos de *manifold learning*, como por exemplo o LLE [86] e o *Local MDS* [85]. Em particular, um algoritmo que se notabilizou, no contexto de visualização de dados, pela capacidade de preservar as vizinhanças foi o t-SNE, cujo funcionamento é revisado brevemente no parágrafo seguinte, uma vez que, durante os experimentos, este se mostrou o algoritmo mais adequado para a tarefa de reorganização do arranjo de pixels.

A construção algorítmica t-SNE consiste em converter uma matriz de distâncias (ou, de maneira mais genérica, de dissimilaridades, como a matriz $\mathbf{D}$ definida na Equação 4.2) em uma matriz estocástica, fazendo com que cada linha se transforme em uma distribuição de probabilidade com dispersão controlada por um hiperparâmetro, chamado de perplexidade, e de modo que pequenas distâncias sejam associadas a probabilidades altas. As projeções são obtidas ajustando suas coordenadas através de um processo de otimização, em que se minimiza a divergência de Kullback-Leibler entre as linhas da matriz estocástica induzida pelas distâncias de entrada, e as suas versões correspondentes induzidas pelas distâncias entre os pontos no espaço de projeções. Como pode ser visto na definição na Equação 2.7, a divergência de Kullback-Leibler penaliza os erros de maneira não uniforme, em que um peso maior é atribuído aos erros de aproximação das maiores probabilidades, que por sua vez estão associadas às menores distâncias. Ou seja, o processo de otimização usado no t-SNE, de fato, prioriza a preservação das vizinhanças.

A aplicação do t-SNE na matriz $\mathbf{D}$, com perplexidade fixa em 25, obtida usando o conjunto CIFAR-10, resulta na projeção apresentada na Figura 12, a qual possui 1024 pontos dispersos no $\mathbb{R}^2$, cada um correspondente a um pixel.

Dessa forma, para cumprir objetivo final de projetar os pixels em um grid $M \times N$ de modo que cada pixel fique próximo daqueles que são mais dependentes de si, dois detalhes precisam ser considerados: (i) A projeção do t-SNE busca garantir a preservação das vizinhanças, de modo que a orientação e a escala dos dados sejam arbitrárias (e de-

[5] É válido salientar que não há garantias de que a matriz $\mathbf{D}$ satisfaz as condições suficientes que definem uma verdadeira matriz de distâncias. Por exemplo, não há garantias de que a desigualdade triangular será satisfeita para todas as trincas da forma (D_{ij}, D_{jk}, D_{ik}). No entanto, isso não impede a aplicação do MDS como um problema de aproximação.

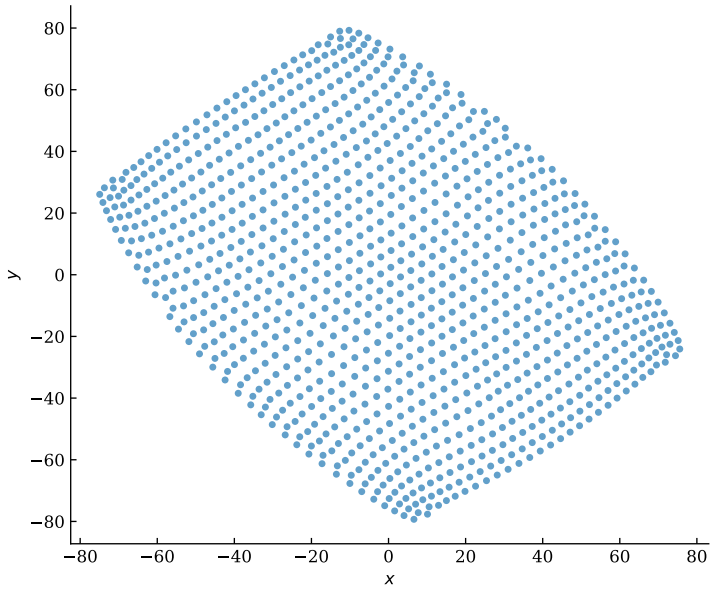


Figura 12 – Projeção dos pixels do CIFAR-10 no 2d usando a matriz **D** como entrada para o t-SNE.

pendentes da inicialização dos pontos). Logo, faz-se necessário aplicar uma rotação e um re-escalonamento nas projeções de modo que elas fiquem alinhadas, ao menos grosseiramente, com um grid $M \times N$. (ii) O t-SNE retorna projeções em um espaço de *embeddings* contínuo. Logo, após o alinhamento da nuvem de pontos com um grid, é necessário algum processo de discretização que atribua cada ponto a uma célula do grid.

Para resolver essas duas questões, foi proposta uma sequência de passos conforme descrito abaixo:

Primeiro passo: Denotemos por $\mathbf{x}_1, \ldots, \mathbf{x}_{M \cdot N}$ os pontos associados a cada um dos MN pixels, obtidos usando o t-SNE, e por x_{i1} e x_{i2} as coordenadas de $\mathbf{x}_i$. O primeiro passo para o alinhamento consiste em centralizar essa nuvem de pontos, para garantir que o centroide esteja na origem. Começar com esse passo previne a necessidade de que uma operação de translação seja necessária em passos posteriores. Logo, calcula-se o centroide conforme a equação

$$\mathbf{x}_{\text{centroide}} = \frac{\mathbf{x}_1 + \cdots + \mathbf{x}_{M \cdot N}}{M \cdot N}, \tag{4.3}$$

e então obtém-se uma nova nuvem de pontos $\mathbf{y}_1, \ldots, \mathbf{y}_{MN}$, em que

$$\mathbf{y}_i = \mathbf{x}_i - \mathbf{x}_{\text{centroide}}. \tag{4.4}$$

Segundo passo: O alinhamento começa com a determinação dos quatro pontos que devem estar alinhados com os vértices do grid retangular, aqui denotados por $\mathbf{y}^{\text{TL}}, \mathbf{y}^{\text{TR}}, \mathbf{y}^{\text{BL}}$ e $\mathbf{y}^{\text{BR}}$.[6] Dessa forma, os dois pontos cuja distância euclidiana entre eles é

[6] Os sobrescritos TL, TR, BL e BR referem-se a *top-left, top-right, bottom-left* e *bottom-right*.

máxima são tomados como um primeiro par de vértices, isto é

$$(\mathbf{y}^{\mathrm{TL}}, \mathbf{y}^{\mathrm{BR}}) = \arg\max_{(\mathbf{y}_i, \mathbf{y}_j)} \|\mathbf{y}_i - \mathbf{y}_j\|. \tag{4.5}$$

A reta que passa por $\mathbf{y}^{\mathrm{TL}}$ e $\mathbf{y}^{\mathrm{BR}}$ deve estar alinhada com uma das diagonais do grid e os seus coeficientes podem ser encontrados resolvendo um sistema de equações, cuja solução é dada por

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1^{\mathrm{TL}} & 1 \\ y_1^{\mathrm{BR}} & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} y_2^{\mathrm{TL}} \\ y_2^{\mathrm{BR}} \end{bmatrix}. \tag{4.6}$$

Da mesma forma, a outra diagonal, que conecta os outros dois vértices, é ortogonal à reta $y_2 - ay_1 - b = 0$. Logo, $\mathbf{y}^{\mathrm{TR}}$ é tomado como ponto de maior distância para essa reta, dentre aqueles que estão acima dela, enquanto $\mathbf{y}^{\mathrm{BL}}$ é tomado como o ponto de maior distância para essa reta, dentre aqueles que estão abaixo dela. Isto é,

$$\mathbf{y}^{\mathrm{TR}} = \arg\max_{\substack{\mathbf{y}_i \\ \text{t.q.} \\ y_{i2} > a \cdot y_{i1} + b}} |y_{i2} - ay_{i1} - b| \tag{4.7}$$

e

$$\mathbf{y}^{\mathrm{BL}} = \arg\max_{\substack{\mathbf{y}_i \\ \text{t.q.} \\ y_{i2} < a \cdot y_{i1} + b}} |y_{i2} - ay_{i1} - b|. \tag{4.8}$$

Terceiro passo: Assumindo, por exemplo, um grid $M \times N$ cujas coordenadas são uniformemente espaçadas entre os intervalos $[-N/2, N/2]$ e $[-M/2, M/2]$, o próximo passo é encontrar uma matriz $\mathbf{A}$ associada a uma transformação linear que satisfaz, aproximadamente, as condições $\mathbf{A}\mathbf{y}^{\mathrm{TL}} = \begin{bmatrix} -N/2 & M/2 \end{bmatrix}^t$, $\mathbf{A}\mathbf{y}^{\mathrm{TR}} = \begin{bmatrix} N/2 & M/2 \end{bmatrix}^t$, $\mathbf{A}\mathbf{y}^{\mathrm{BL}} = \begin{bmatrix} -N/2 & -M/2 \end{bmatrix}^t$ e $\mathbf{A}\mathbf{y}^{\mathrm{BR}} = \begin{bmatrix} N/2 & -M/2 \end{bmatrix}^t$, simultaneamente. Ou seja, $\mathbf{A}$ deve ser uma matriz que alinha (aproximadamente) cada um dos pontos encontrados no segundo passo com o seu respectivo vértice no grid. Assim, toma-se $\mathbf{A}$ como a solução aproximada ótima, sob o critério do erro quadrático médio, para o sistema

$$\mathbf{A} \cdot \begin{bmatrix} | & | & | & | \\ \mathbf{y}^{\mathrm{TL}} & \mathbf{y}^{\mathrm{TR}} & \mathbf{y}^{\mathrm{BL}} & \mathbf{y}^{\mathrm{BR}} \\ | & | & | & | \end{bmatrix} = \begin{bmatrix} -N/2 & N/2 & -N/2 & N/2 \\ M/2 & M/2 & -M/2 & -M/2 \end{bmatrix}, \tag{4.9}$$

que é dada por

$$\hat{\mathbf{A}} = \begin{bmatrix} -N/2 & N/2 & -N/2 & N/2 \\ M/2 & M/2 & -M/2 & -M/2 \end{bmatrix} \cdot \begin{bmatrix} | & | & | & | \\ \mathbf{y}^{\mathrm{TL}} & \mathbf{y}^{\mathrm{TR}} & \mathbf{y}^{\mathrm{BL}} & \mathbf{y}^{\mathrm{BR}} \\ | & | & | & | \end{bmatrix}^{\dagger}, \tag{4.10}$$

em que $(\cdot)^{\dagger}$ denota a pseudoinversa de Moore-Penrose.

Com isso, obtém-se uma nova configuração de pontos $\mathbf{z}_1, \ldots, \mathbf{z}_{M \cdot N}$, alinhada com o grid, definida como

$$\mathbf{z}_i = \hat{\mathbf{A}}\mathbf{y}_i \; \forall \; i \in \{1, 2, \ldots, M \cdot N\}. \tag{4.11}$$

Essa nova configuração, resultado da aplicação da matriz $\hat{\mathbf{A}}$ na configuração da Figura 12, é apresentada na Figura 13.

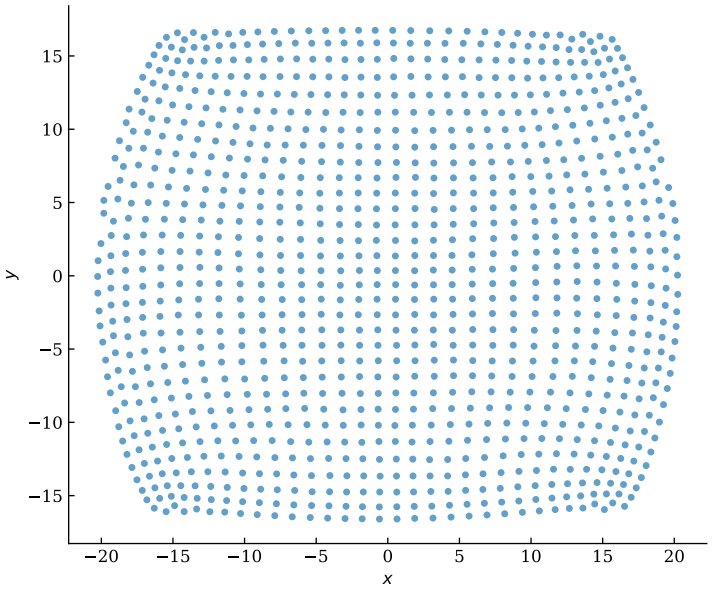


Figura 13 – Projeção dos pixels após a aplicação da matriz $\hat{\mathbf{A}}$.

Antes de prosseguir com o próximo passo, convém fazer a seguinte observação. Alguns dos leitores podem considerar que a solução mais natural e direta para remover a arbitrariedade da orientação, na nuvem de pontos obtida através do t-SNE (Figura 12 $\rightarrow$ Figura 13), seria a aplicação da transformada de Karhunen-Loève, que alinharia qualquer tendência em uma direção arbitrária com um dos eixos de coordenadas. De fato, essa solução foi a primeira a ser considerada. Entretanto, durante os experimentos, os resultados obtidos na reorganização das imagens de uma das bases de dados se mostraram insatisfatórios, ao passo que, após a discretização da nuvem de pontos, as relações de vizinhanças preservadas pelo t-SNE foram descaracterizadas. Logo, foi feita a opção pelo procedimento descrito até aqui.

Quarto passo: Nesse ponto, cada pixel está associado a um ponto em $\{\mathbf{z}_1, \ldots, \mathbf{z}_{M \cdot N}\}$, de modo que esse conjunto de pontos esteja alinhado, ao menos grosseiramente, com um grid retangular $M \times N$. Para discretizar essa nuvem de pontos, define-se o grid bidimensional discreto

$$\mathcal{G} = \{(p_i, q_j) | i = 0, \ldots, N-1, j = 0, \ldots M-1\}, \tag{4.12}$$

em que

$$p_i = p_{\min} + i\Delta_p,\ q_j = q_{\min} + j\Delta_q, \tag{4.13}$$

com espaçamentos dados por

$$\Delta_p = \frac{p_{\max} - p_{\min}}{N-1},\ \Delta_q = \frac{q_{\max} - q_{\min}}{M-1}. \tag{4.14}$$

Os extremos do domínio são

$$p_{\min} = \min_i z_{i1}, \quad p_{\max} = \max_i z_{i1}, \quad q_{\min} = \min_i z_{i2}, \quad q_{\max} = \max_i z_{i2}. \tag{4.15}$$

O último passo consiste em montar uma relação biunívoca que atribui cada ponto em $\{\mathbf{z}_1, \ldots, \mathbf{z}_{MN}\}$ a uma célula em $\mathcal{G} = \{\mathbf{g}_1, \ldots, \mathbf{g}_{MN}\}$, de modo que a distância total das associações seja mínima. Formalmente, isso é equivalente a encontrar uma permutação π de $\{1, \ldots, MN\}$ que minimiza $\sum_{i=1}^{MN} \|\mathbf{g}_i - \mathbf{z}_{\pi(\mathbf{i})}\|$.

Isso pode ser interpretado como o problema de encontrar o emparelhamento perfeito mínimo para um grafo bipartido da seguinte forma. Um grafo bipartido é um grafo no qual os vértices são particionados em dois subconjuntos disjuntos U e V de forma que não exista nenhuma conexão entre dois nós no mesmo subconjunto, e um emparelhamento perfeito consiste em um conjunto de arestas tal que nenhum vértice é incidente a mais de uma aresta do conjunto, e todos os nós em U e V estão pareados (quando possível). Desse modo, pode-se construir um grafo bipartido em que cada ponto $\mathbf{z}$ é um nó em U, e cada célula em $\mathcal{G}$ é um nó em V, enquanto uma aresta que conecta $\mathbf{g}_i$ a $\mathbf{z}_j$ é atribuída a um peso $\|\mathbf{z}_i - \mathbf{g}_j\|$. Por construção, encontrar um emparelhamento perfeito mínimo para esse grafo é equivalente a minimizar a função de custo definida no parágrafo acima. A solução ótima para esse problema pode ser obtida usando o algoritmo Húngaro (ou método de Kuhn-Munkres), conforme descrito em [87].

A sequência de passos descrita acima produz uma nova organização para os pixels em um grid $M \times N$, na qual as variáveis que compartilham alta informação mútua são colocadas em posições próximas, enquanto aquelas que compartilham baixa informação mútua são colocadas distantes. A Figura 14 apresenta um exemplo de comparação visual, usando uma imagem do conjunto CIFAR-10, entre a imagem original (que aqui é assumida desconhecida), a imagem permutada (entrada para o algoritmo) e a imagem obtida após a reorganização (saída do algoritmo).

Como pode ser visto, a metodologia proposta, baseada em estimativas de informação mútua, para reorganizar o arranjo de pixels, parece ser capaz de reconstruir, a menos da orientação, a topologia 2D das imagens, assumida desconhecida originalmente. Na Seção 4.6, essa propriedade é verificada de maneira mais formal para diferentes bases de dados. Com esse resultado, as noções de vizinhança e conectividade local voltam a ter o significado que lhes permitem serem exploradas pelas redes convolutivas. De fato,

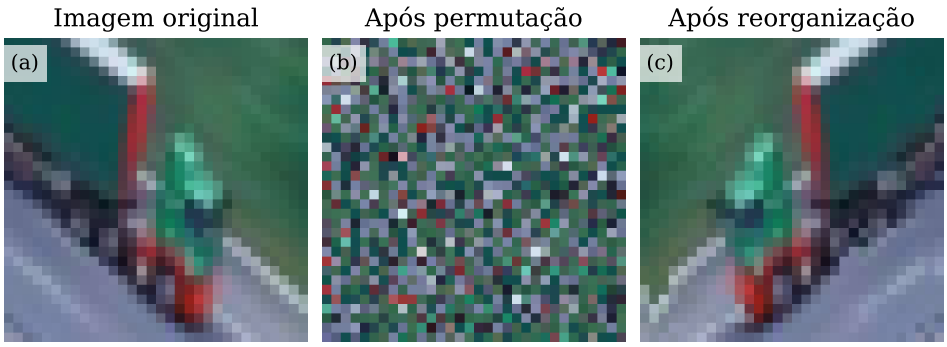


Figura 14 – Ilustração visual para uma imagem do conjunto CIFAR-10.

os experimentos apresentados na Seção 4.6 também corroboram isso, através de comparações entre os desempenhos de CNNs treinadas com cada uma das três versões da base, nas quais a reorganização baseada em medidas de dependência produzem resultados aproximadamente equivalentes em relação aqueles produzidos pelas imagens originais.

4.5 Sobre a dimensão intrínseca do arranjo de pixels

Até aqui, foi assumido que a dimensionalidade do grid no qual o arranjo de pixels deve estar disposto é dois, o que é bastante natural, uma vez que as imagens são planares. Uma forma, também natural, de estender esse estudo é investigando se a dimensionalidade ideal para o arranjo de pixels também pode ser inferida através das medidas de informação mútua, e não apenas as posições dos pixels em uma estrutura preestabelecida.

Para tal, é usado o conceito de dimensão intrínseca (DI), definido em [88] como o "*o número mínimo de variáveis livres necessárias para representar os dados sem perda de informação*". No contexto do aprendizado de máquina, a estimação da DI de um conjunto de observações D-dimensionais é uma ferramenta extremamente útil sempre que elas ocupam, aproximadamente, uma estrutura (*manifold*) de dimensão $d < D$, podendo guiar o projeto e a análise de modelos. Por exemplo, ao construir um *autoencoder*, o conhecimento (ou uma estimativa) de d pode ser útil à medida que sugere a dimensão adequada para um *embedding* de uma nuvem de pontos, de modo que cada instância possa ser representada sem perda de informação e com uma quantidade mínima de redundâncias.

Como conhecido na literatura, estimar a DI é um problema desafiador, uma vez que diversos estimadores existem e podem produzir resultados discrepantes para os mesmos conjuntos de dados [89, 90, 91]. Muitos desses estimadores usam apenas a informação das distâncias entre os pares de pontos, em vez das coordenadas de cada ponto. Essa é uma propriedade importante pois permite a aplicação no contexto deste trabalho, no qual não existe um conjunto de pontos em D dimensões como ponto de partida. Logo, é possível usar a matriz $\mathbf{D}$, cujas entradas codificam a noção de (pseudo) distâncias, para estimar a

dimensão ideal para um *embedding* do conjunto de pixels, de modo que todas as relações de dependências relevantes sejam apropriadamente acomodadas por meio de vizinhanças espaciais.

Dessa forma, devido a sua simplicidade e ampla validação na literatura, foi escolhido trabalhar com o estimador de Grassberger-Procaccia [92], como explicado na sequência.

4.5.1 O estimador de Grassberger-Procaccia

Dado um conjunto de N pontos $\mathbf{x}_1, \ldots, \mathbf{x}_N$ com matriz de distâncias $\mathbf{D}$, o estimador de Grassberger-Procaccia é baseado no conceito de dimensão de correlação, que por sua vez é baseado na integral de correlação, definida em [92], e que pode ser estimada como

$$C(r) = \frac{1}{N(N-1)} \sum_{i<j}^{N} u(D_{ij} - r), \tag{4.16}$$

em que $u(\cdot)$ é a função degrau. A função $C(r)$ pode ser interpretada como uma contagem de coincidências na amostra $\mathbf{x}_1, \ldots, \mathbf{x}_N$, com o evento coincidência sendo definido como a ocorrência de duas observações independentes a uma distância menor que r. Em [92], foi mostrado que, para pequenos valores de r e para N suficientemente grande, a integral de correlação se comporta, aproximadamente, conforme a relação

$$C(r) \propto r^d, \tag{4.17}$$

em que d é definida como a dimensão de correlação. Em [92], a dimensão de correlação foi usada para caracterizar atratores em sistemas dinâmicos não lineares. Em sistemas caóticos, esses atratores tipicamente possuem dimensões fractais. Mas, para estruturas mais comportadas, a dimensão de correlação coincide com a noção usual de dimensão. Por exemplo, a dimensão de correlação de um ponto é 0, enquanto a de um ciclo limite (uma curva fechada) é 1, e a de uma superfície é 2.

Para entender a relação 4.17, assuma que os pontos $\mathbf{x}_1, \ldots, \mathbf{x}_N$ foram amostrados de uma V.A. X cuja função densidade de probabilidade é $f(\mathbf{x})$. Note que, para pequenos valores de r, se os pontos estão sobre uma estrutura d-dimensional (potencialmente embutida em um espaço de dimensão maior), é possível tomar uma bola $\mathcal{B}_i$ (também d-dimensional) de raio r centrada em $\mathbf{x}_i$ tal que $f(\mathbf{x}) \approxeq f(\mathbf{x}_i)$ para todo $\mathbf{x} \in \mathcal{B}_i$. Logo, a probabilidade de que uma amostra aleatória esteja a uma distância inferior a r de $\mathbf{x}_i$ é dada por

$$p_{\mathcal{B}_i} = \Pr(X \in \mathcal{B}_i) \approx \alpha \cdot f(\mathbf{x}_i) \cdot r^d, \tag{4.18}$$

em que α é uma constante de proporcionalidade, que depende de d e da métrica utilizada.

Defina, então, g_i como

$$g_i = \sum_{j>i}^{N} u(r - d_{ij}) \tag{4.19}$$

e note que $g_i \sim \text{Bin}(N-i, p_{\mathcal{B}_i})$. Logo, segue que

$$\mathbb{E}[g_i] = (N-i) \cdot p_{\mathcal{B}_i}, \tag{4.20}$$

e, consequentemente,

$$\mathbb{E}[g_1 + \cdots + g_N] = \sum_{i=1}^{N} (N-i) \cdot p_{\mathcal{B}_i} \approx \left(\sum_{i=1}^{N} (N-i) \cdot f(\mathbf{x}_i) \right) \cdot \alpha \cdot r^d. \tag{4.21}$$

Isso implica $\mathbb{E}[g_1 + \cdots + g_N] \propto r^d$, o que explica a lei de potência na Equação 4.17, uma vez que $C(r) \propto g_1 + \cdots + g_N$, e que a variância de $C(r)$ tende para zero à medida que N cresce.

Da Equação 4.17, segue que

$$\log C(r) \approx d \log r + h, \tag{4.22}$$

em que h é o logarítmico da constante de proporcionalidade.

Isso sugere um método simples para estimação de d, conhecido como o método de Grassberger-Procaccia (GP), baseado na contagem de coincidências em um conjunto de observações para diferentes valores de r. Com isso, é possível obter uma coleção de pontos da forma $(\log r, \log C(r))$, em que o coeficiente angular da reta que melhor se encaixa a essa coleção é tomado como a estimativa final. Entretanto, na maioria das bases de dados reais a expectativa de uma única reta que explique a relação entre $\log r$ e $\log C(r)$ é frustrada, uma vez que é possível observar múltiplas tendências lineares com diferentes inclinações para diferentes intervalos de variação de r.

Como o método GP é desenvolvido assumindo valores pequenos de r, o coeficiente angular da primeira tendência linear, compreendendo a relação entre $\log r$ e $\log C(r)$ para $r \to 0$, pode ser tomado como a DI. No entanto, em [93], foi desenvolvida uma adaptação deste método, na qual mostrou-se que as estimativas obtidas usando um intervalo com valores maiores de r também são significativas. Mais especificamente, foi mostrado que a dimensão estimada depende da escala de observação dos dados, à medida que menores valores de r permitem a análise de DI em escalas mais finas, enquanto valores maiores de r correspondem a uma análise em escalas mais grosseiras.

Em particular, foi observado, em bases de dados reais e sintéticas, que nas escalas mais finas a estimativa é frequentemente máxima, uma vez que essas escalas são dominadas pelo ruído, que tende a corromper todas as dimensões de maneira independente. No outro extremo, valores muito altos de r tendem a englobar todos os pontos observados, fazendo com que a curva sature e a estimativa resultante seja 0, indicando que qualquer estrutura pode ser observada como um ponto desde que a escala de observação seja grosseira o suficiente. Por outro lado, nas escalas intermediárias, é possível observar a emergência de

tendências lineares que sugerem a dimensão da estrutura na qual está a informação de interesse nos dados.

Os achados em [93] revelam uma ambiguidade inerente ao processo de estimação de DI, que é a sensibilidade do resultado à escala de observação, uma vez que não existem métodos consolidados para uma escolha automática da mesma, como pontuado em [91]. Apesar disso, a adaptação proposta em [93] sugere que o estimador GP pode ser usado como uma ferramenta visual de análise multiescala da DI, na qual os resultados podem ser interpretados de acordo com o problema e aproveitando o raciocínio do parágrafo anterior. De fato, para o problema proposto neste trabalho, tal análise é feita na Seção 4.6.3 a partir da matriz $\mathbf{D}$, obtida conforme a Equação 4.2, com o objetivo de investigar se existe uma dimensão adequada para o arranjo de pixels (baseado nas dependências entre pares de pixels), e se esta confirma a expectativa de uma estrutura bidimensional, uma vez que se tratam de imagens planares.

4.6 Experimentos e resultados

Neste capítulo, três tipos de resultados experimentais são apresentados. O primeiro deles consiste em uma comparação entre as imagens naturais e as imagens obtidas após a aplicação do algoritmo de reorganização dos pixels. Isso inclui uma avaliação qualitativa, baseada em comparações visuais entre as duas imagens, semelhante à comparação da Figura 14, como também comparações quantitativas que buscam avaliar até que ponto as estruturas locais foram reconstruídas. Para a comparação quantitativa, o critério descrito a seguir foi proposto.

Sejam os conjuntos de pontos $\{\mathbf{x}_1, \ldots, \mathbf{x}_{M \cdot N}\} \subset \mathbb{R}^2$ e $\{\mathbf{y}_1, \ldots, \mathbf{y}_{M \cdot N}\} \subset \mathbb{R}^2$, em que $\mathbf{x}_i$ e $\mathbf{y}_i$ representam as posições do i-ésimo pixel (indexação arbitrária) nos grids das imagens originais e reorganizadas, respectivamente. Denote, então, por $N_k^x(i)$ e $N_k^y(i)$ os conjuntos de índices associados aos k vizinhos mais próximos de $\mathbf{x}_i$ (em $\{\mathbf{x}_1, \ldots, \mathbf{x}_{M \cdot N}\}$) e $\mathbf{y}_i$ (em $\{\mathbf{y}_1, \ldots, \mathbf{y}_{M \cdot N}\}$), respectivamente. O grau de reconstrução da vizinhança em torno do i-ésimo pixel pode ser quantificado através da interseção entre os dois conjuntos de pixels, de acordo com a equação

$$N_k(i) = \frac{|N_k^x(i) \cap N_k^y(i)|}{k}, \tag{4.23}$$

em que $|\cdot|$ representa a cardinalidade do conjunto. Dessa forma, a grau de reconstrução das estruturas locais nas imagens pode ser avaliado por

$$N_k = \sum_{i=1}^{MN} N_k(i). \tag{4.24}$$

Colocando em palavras, a quantidade N_k mede a fração média de vizinhos que são preservados nas vizinhanças contendo k pixels ao longo de toda a imagem e, portanto, é

usada na Seção 4.6.1 para complementar a comparação visual entre as imagens originais e as imagens obtidas após a reorganização do arranjo de pixels.

O segundo conjunto de experimentos consiste no treinamento de uma mesma arquitetura convolutiva usando as imagens originais, permutadas e reorganizadas. O objetivo desses experimentos é responder diretamente a questão que foi posta no início deste capítulo. Isto é, a de entender se é possível obter resultados próximos aos de uma CNN, treinada com o arranjo original dos pixels, usando as estimativas de dependências entre pares de variáveis e a reorganização das imagens para impor a agregação entre variáveis dependentes.

Finalmente, o último conjunto de experimentos consiste na investigação da dimensão intrínseca do arranjo de pixels baseado na matriz de pseudodistâncias $\mathbf{D}$ (Equação 4.2), que reflete as relações de dependências entre pares de variáveis. Essa investigação é feita usando a abordagem multiescala proposta em [93], e tem como objetivo questionar se, sob o ponto de vista das dependências entre variáveis, projetar o arranjo de pixels em um grid com as mesmas dimensões da imagem original é de fato a melhor estratégia, o que pode não ser o caso se as estimativas de DI obtidas convergirem para um arranjo de dimensão mais alta.

Em todos os experimentos, quatro bases de dados são utilizadas:

(i) CIFAR-10 [58], que é um conjunto de dados fundamental em visão computacional, composto por 60.000 imagens coloridas de 32×32 pixels, divididas em 10 classes perfeitamente balanceadas. Amplamente usado para treinar modelos de aprendizado de máquina e redes neurais convolucionais, ele contém 50000 imagens de treino e 10000 de teste.

(ii) CIFAR-100 [58], que também é um *benchmark* popular de visão computacional, composto por 60000 imagens coloridas de 32×32 pixels, divididas em 100 classes (600 imagens de cada). Ele inclui 50000 imagens de treinamento e 10000 de teste.

(iv) Tiny ImageNet, que é um subconjunto reduzido do conjunto de dados ImageNet [94], servindo como uma alternativa mais acessível para treinar e avaliar modelos de visão computacional, especialmente quando os recursos computacionais são limitados. Ele é composto por 100000 imagens para treino e 10000 imagens para teste, separadas em 200 classes perfeitamente balanceadas. Neste trabalho, para aliviar o custo computacional, as imagens foram subamostradas para uma resolução de 32×32 pixels, após uma filtragem *anti-aliasing* com frequência de corte na metade da banda.

(iii) EuroSat [95], é um conjunto bastante conhecido na área de sensoriamento remoto, usado principalmente para classificação de uso e cobertura do solo. Ele contém

27000 imagens coloridas de 64×64 pixels, separadas em 10 classes. As imagens são de resolução 64×64 e também foram subamostradas para uma resolução 32×32. Além disso, essa base foi particionada, com 70 % das imagens sendo destinadas para treinamento e o restante para teste.

Durante os experimentos, para cada uma das bases de dados, as estimativas de informação mútua foram sempre obtidas usando subconjuntos com 10000 imagens, dentre as que foram separadas para treinamento.

4.6.1 Reorganização do arranjo de pixels

O resultado da reconstrução do arranjo de pixels é apresentado na Figura 15, usando um exemplo de teste para cada uma das bases de dados.

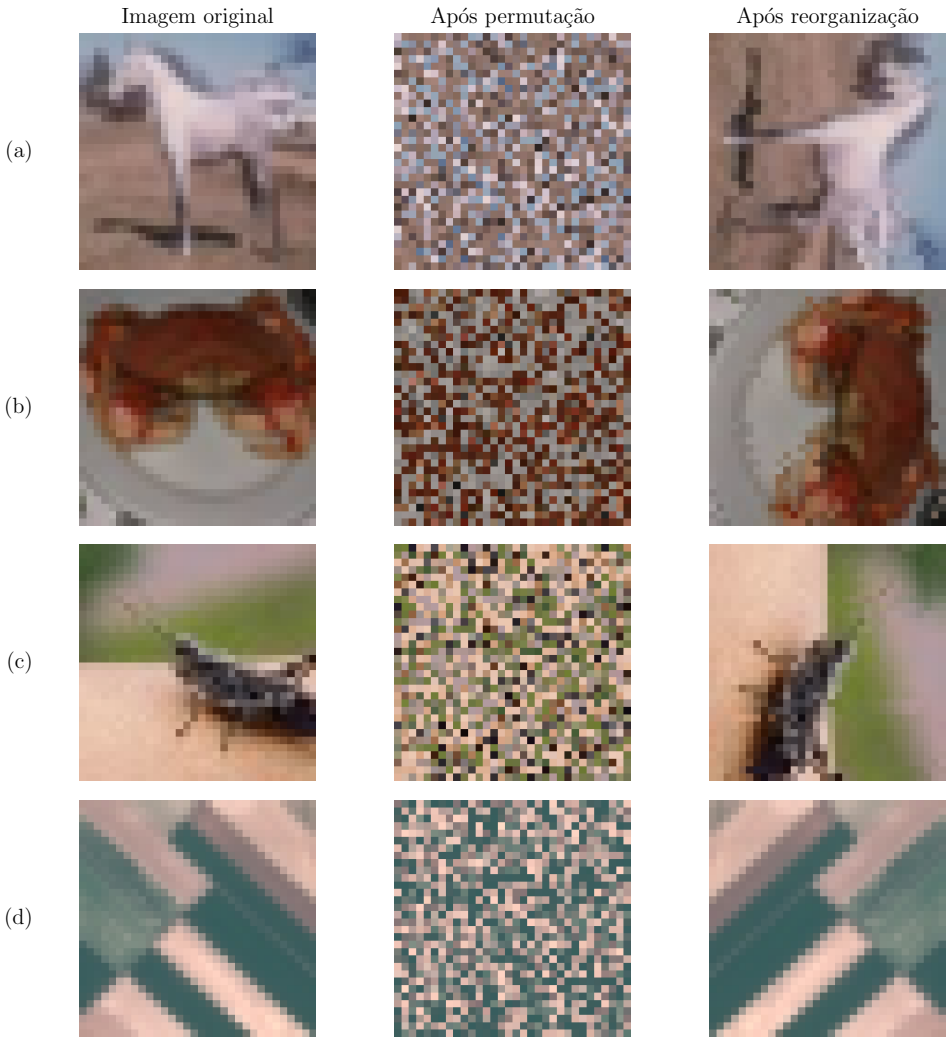


Figura 15 – Comparação visual entre a imagem original, imagem embaralhada e imagem reorganizada, para um exemplo de teste da base (a) CIFAR-10; (b) CIFAR-100; (c) Tiny ImageNet; (d) EuroSat;

Assim como na Figura 14, é possível observar que, em cada uma das quatro bases de dados utilizadas, o algoritmo proposto na Seção 4.4 é capaz de recuperar, com alta

precisão, as estruturas locais a partir de versões completamente descaracterizadas devido à permutação dos pixels. Como resultado, as imagens permutadas do conjunto são reordenadas de modo que seu conteúdo possa ser reconhecido pelo sistema visual humano, aproximadamente tão bem quanto nas imagens originais.

Para reforçar essa percepção, na Tabela 2 são apresentados o valores de N_k, conforme definido no início desta seção, para diferentes valores de k.

Tabela 2 – Valores de N_k, após a reorganização do arranjo, para diferentes valores de k em cada conjunto de dados

Dataset	k=10	k=20	k=30	k=40	k=50
CIFAR-10	0.8711	0.9134	0.9164	0.9355	0.9368
CIFAR-100	0.8656	0.9082	0.9125	0.9294	0.9304
Tiny ImageNet	0.9066	0.9385	0.9413	0.9546	0.9561
EuroSat	1.0000	1.0000	1.0000	1.0000	1.0000

Como pode ser visto, em todos os casos, após a aplicação do algoritmo, uma grande fração das relações de vizinhança existentes nas imagens originais são recuperadas, com uma tendência dessa fração ser ainda maior à medida que se adota uma vizinhança maior (i.e., cada pixel possui um número maior de vizinhos). Um caso extremo, e surpreendente, é observado nos resultados correspondentes ao conjunto EuroSat, nos quais é possível observar que 100% das vizinhanças são preservadas, para todos os cinco valores de K utilizados, indicando que, para este conjunto em específico, as medidas de informação mútua estimadas permitiram a reconstrução perfeita do arranjo de pixels (a menos de uma reflexão).

4.6.2 Experimentos de classificação

No início deste capítulo, foi posto um questionamento acerca da possibilidade de, partindo apenas do mapa de estimativas de informação mútua entre pares de pixels, obter desempenhos de classificação comparáveis aos de uma CNN treinada quando a ordenação espacial dos pixels é conhecida previamente. Para respondê-lo, foi proposto um algoritmo, conforme explicado na Seção 4.4, cujo objetivo é reorganizar o arranjo espacial de uma base na qual os pixels estão ordenados de maneira aleatória.

Com isso, foram obtidas três versões para cada base de dados, com a diferença entre elas sendo a ordem dos pixels. Na primeira delas, eles estão ordenados naturalmente, correspondendo ao cenário usual no reconhecimento de imagens. Na segunda versão, os pixels estão ordenados de maneira aleatória, enquanto, na terceira, eles estão ordenados conforme o arranjo produzido pelo algoritmo proposto na Seção 4.4.

Finalmente, foi definido um modelo fixo, inspirado na arquitetura VGG [96], e este foi treinado usando cada uma das três versões explicadas acima. Mais especificamente, a

arquitetura utilizada é descrita da seguinte forma: $\mathrm{C}_3^{64} - \mathrm{BN} - \mathrm{ReLU} - \mathrm{D} - \mathrm{C}_{64}^{64} - \mathrm{BN} - \mathrm{ReLU} - \mathrm{D} - \mathrm{MP} - \mathrm{C}_{64}^{128} - \mathrm{BN} - \mathrm{ReLU} - \mathrm{D} - \mathrm{C}_{128}^{128} - \mathrm{BN} - \mathrm{ReLU} - \mathrm{D} - \mathrm{MP} - \mathrm{C}_{128}^{256} - \mathrm{BN} - \mathrm{ReLU} - \mathrm{D} - \mathrm{C}_{256}^{256} - \mathrm{BN} - \mathrm{ReLU} - \mathrm{D} - \mathrm{MP} - \mathrm{D} - \mathrm{FC}_{n_{\text{classes}}}$. Aqui, C_i^o se refere a uma camada convolutiva com i filtros de entrada, o filtros de saída, *kernel* de tamanho $k = 3$ e *stride* $s = 1$; BN corresponde a uma camada de *batch normalization*; D corresponde a uma camada de *dropout* com $p = 0.3$; MP corresponde a uma camada de *max pooling* com *kernel* de tamanho $k = 2$ e *stride* $s = 2$; FC_o corresponde a uma camada densa com o neurônios em sua saída.

Dessa forma, em cada uma das base de dados, é possível responder diretamente o questionamento posto a partir da verificação do quanto o desempenho do classificador na terceira versão se aproxima do desempenho do mesmo quando treinado com a primeira versão. Adicionalmente, a comparação com a segunda versão permite avaliar o ganho de desempenho proporcionado pela imposição da agregação entre variáveis dependentes, quando comparado à agregação entre variáveis de maneira aleatória. Os resultados obtidos são apresentados nas Tabelas 3–6 [7].

Versão das imagens	Acurácia (%)	Acurácia top-3 (%)	Acurácia top-5 (%)	Acurácia top-10 (%)
Imagens originais	87.67 ± 0.41	97.84 ± 0.16	99.50 ± 0.08	—
Após permutação	57.63 ± 0.91	85.76 ± 0.57	94.59 ± 0.33	—
Após reorganização	85.32 ± 0.56	97.29 ± 0.20	99.30 ± 0.07	—

Tabela 3 – Experimentos com o conjunto CIFAR-10.

Versão das imagens	Acurácia (%)	Acurácia top-3 (%)	Acurácia top-5 (%)	Acurácia top-10 (%)
Imagens originais	60.83 ± 0.37	79.23 ± 0.34	85.60 ± 0.24	92.01 ± 0.25
Após permutação	29.43 ± 0.48	47.40 ± 0.58	56.56 ± 0.51	68.88 ± 0.54
Após reorganização	58.01 ± 0.49	76.87 ± 0.54	83.63 ± 0.51	90.54 ± 0.34

Tabela 4 – Experimentos com o conjunto CIFAR-100.

Versão das imagens	Acurácia (%)	Acurácia top-3 (%)	Acurácia top-5 (%)	Acurácia top-10 (%)
Imagens originais	37.57 ± 0.52	55.15 ± 0.59	63.06 ± 0.62	73.26 ± 0.53
Após permutação	14.67 ± 0.32	26.56 ± 0.38	33.64 ± 0.35	44.75 ± 0.39
Após reorganização	36.13 ± 0.37	53.50 ± 0.54	61.35 ± 0.49	71.71 ± 0.40

Tabela 5 – Experimentos com o conjunto Tiny ImageNet.

Versão das imagens	Acurácia (%)	Acurácia top-3 (%)	Acurácia top-5 (%)	Acurácia top-10 (%)
Imagens originais	86.95 ± 2.76	98.90 ± 0.40	99.85 ± 0.09	—
Após permutação	66.40 ± 3.96	92.73 ± 1.84	98.14 ± 0.82	—
Após reorganização	87.07 ± 3.22	99.05 ± 0.22	99.89 ± 0.05	—

Tabela 6 – Experimentos com o conjunto EuroSAT.

Como pode ser visto, em todos os casos, os resultados obtidos usando as imagens originais e as imagens permutadas confirmam aqueles reportados em [75], à medida que

[7] Os resultados da acurácia top-10 (%) não são reportados para os conjuntos CIFAR-10 e EuroSAT pois estas possuem apenas dez classes.

apontam para uma diferença significativa de desempenho entre os dois cenários. Mais do que isso, antes de qualquer experimento, tal diferença deve ser esperada por razões teóricas, uma vez que a permutação dos pixels descaracteriza a coerência espacial e as estruturas locais que justificam o uso das redes convolutivas.

Por outro lado, os resultados acima também confirmam que a maior parte dessa diferença é superada através da localização das dependências e da reorganização do arranjo espacial de modo a garantir que as camadas de uma CNN promovam a agregação entre variáveis dependentes. De fato, como pode ser visto nas tabelas acima, em todas as bases de dados, as acurácias das redes treinadas após a reorganização estão, consistentemente, apenas $1 \sim 2$ pontos percentuais abaixo daqueles obtidos quando a rede é treinada usando a organização original dos pixels.

Tal proximidade entre os dois conjuntos de resultados é significativa, uma vez que a narrativa dominante na literatura é a de que o sucesso das redes neurais convolucionais se deve ao viés indutivo que é imposto por conta da informação presente nas adjacências entre pixels, e que, sem essas informações, os resultados são comprometidos. No entanto, usando as motivações postas no início deste trabalho (envolvendo os benefícios da imposição da agregação entre variáveis dependentes), foi possível mostrar que a geometria espacial das imagens é identificável a partir de dependências entre pares de variáveis, e que, portanto, as informações que justificam o viés indutivo das CNNs podem ser aprendidas através dos dados. Ou seja, embora a exploração da estrutura espacial seja sempre apresentada como a justificativa para os ganhos expressivos no reconhecimento de imagens quando substituímos uma rede neural completamente conectada por uma rede neural convolucional, o conhecimento completo *a priori* dessa estrutura não é realmente necessário para se obter a maior parte desses ganhos.

Além disso, é de se destacar que o algoritmo proposto para reorganizar o arranjo de pixels não foi ajustado exaustivamente. Logo, é de se imaginar que melhorias adicionais podem levar a organizações espaciais que produzam acurácias ainda mais próximas das originais. A título de exemplo, o processo de discretização dos *embeddings* gerados pelo t-SNE em um grid poderia ser refinado usando as técnicas mais avançadas propostas em [97, 98], no contexto de visualização de dados. No entanto, o objetivo dessa investigação não era necessariamente obter resultados compatíveis com estado da arte, mas sim estudar o sucesso das CNNs sob o ponto de vista da agregação de variáveis dependentes através do questionamento proposto, funcionando assim como uma prova de conceito para as hipóteses enunciadas na introdução deste trabalho. Logo, os resultados obtidos com essa metodologia foram considerados satisfatórios.

4.6.3 Estimação da dimensão intrínseca

Nesta seção, são apresentados os resultados da análise de DI, sob o ponto de vista das dependências entre variáveis (e não das imagens como instâncias de dados), para as quatro bases de dados usadas neste capítulo. Dessa forma, na Figura 16 é possível observar os gráficos $\log r$ vs $\log C(r)$ para cada uma delas.

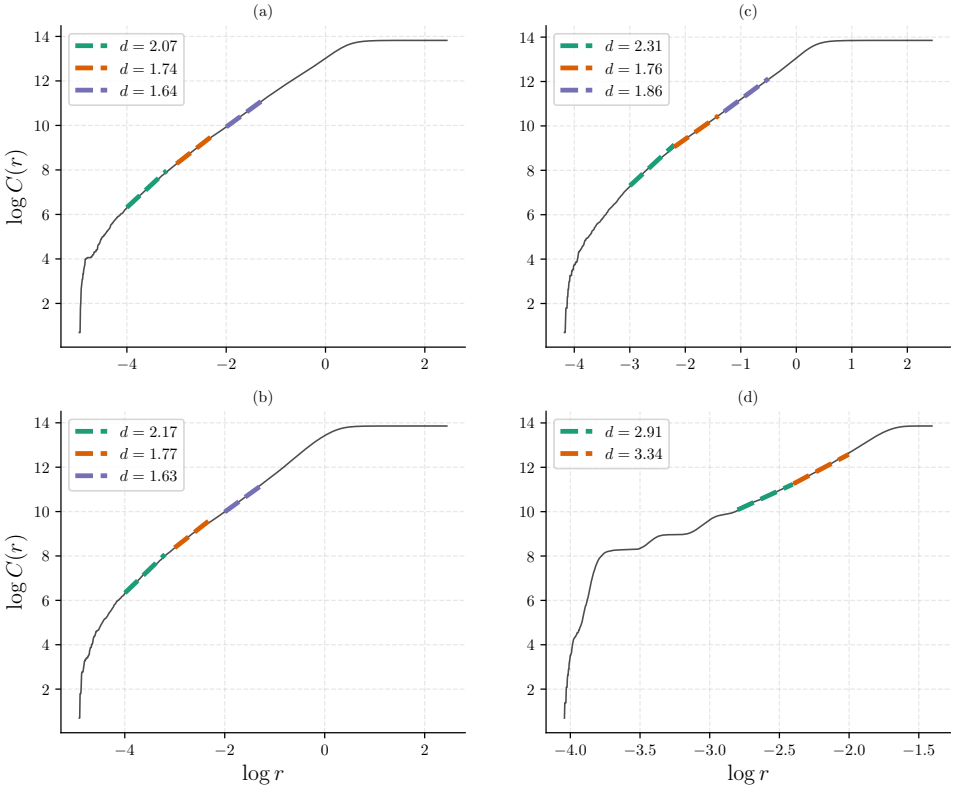


Figura 16 – Análise de DI usando a matriz **D** da base (a) CIFAR-10; (b) CIFAR-100; (c) Tiny ImageNet; (d) EuroSat;

Como pode ser visto, a expectativa de dimensões altas nas escalas mais finas, conforme explicado na Seção 4.6.3, é confirmada em todas as bases. É importante ressaltar que, no nosso contexto, essas estimativas não se dão por conta do ruído que corrompe diretamente as observações em espaços de alta dimensão, como é de se esperar no regime clássico de aplicação do método GP, uma vez que, aqui, essas observações nem sequer estão disponíveis inicialmente. No entanto, como partimos diretamente da matriz de pseudodistâncias, que são obtidas em função da informação mútua entre as variáveis, o "ruído" que induz as altas dimensões para pequenos valores de r é introduzido pelos erros associados ao estimador de informação mútua. Da mesma maneira, o regime de saturação para grandes valores de r também é observado em todas as bases de dados.

Assim, seguindo o raciocínio explicado na Seção 4.6.3, diversas estimativas em escalas intermediárias foram obtidas usando ajustes de mínimos quadrados, que correspondem aos segmentos de reta pontilhados na Figura 16.

Com isso, é possível notar que, para os conjuntos CIFAR-10, CIFAR-100 e Tiny Imagenet, todas as estimativas em escalas intermediárias oscilam consistentemente em torno de uma dimensão $d = 2$. Isso sugere que, além da ordenação dos pixels assumindo um grid bidimensional pré-estabelecido, a escolha da estrutura bidimensional como adequada para acomodar as relações de dependências entre os pixels (codificadas através das distâncias entre as células) também pode ser inferida através dos dados, para essas bases.

Entretanto, a expectativa da descoberta de uma estrutura bidimensional foi frustrada no conjunto EuroSAT, uma vez que as estimativas em escalas intermediárias sugerem uma estrutura tridimensional. Até aqui, não há uma explicação fechada para essa observação. Por um lado, ela abre a possibilidade para um questionamento acerca do grid 32×32 como a melhor escolha para reorganizar o arranjo de pixels. Embora essa seja a estrutura na qual a imagem é originalmente amostrada, as estimativas de DI obtidas podem revelar que, sob o ponto de vista da agregação de variáveis dependentes, um grid tridimensional pode ser mais adequado. No entanto, de uma forma geral, é possível observar que o gráfico da Figura 16 (d) segue um padrão muito diferente dos anteriores. Por exemplo, é possível notar múltiplas tendências lineares curtas com inclinações completamente diferentes, indicando mudanças abruptas na DI observada com pequenas alterações na escala de observação. Tal instabilidade também pode indicar que a matriz $\mathbf{D}$, obtida através das estimativas de informação mútua entre as variáveis do conjunto EuroSAT, não sugere uma estrutura bem comportada, com uma DI constante ao longo do espaço, como nas outras bases.

Por outro lado, isso se torna ainda mais surpreendente, tendo em vista que, de acordo com o critério utilizado na Seção 4.6.1, a reorganização dos pixels em um grid bidimensional para o conjunto EuroSAT gerou as imagens que reconstroem as versões originais com maior grau de fidelidade. Dessa forma, a explicação desse resultado permanece como uma questão em aberto nessa investigação.

4.7 Conclusões do capítulo

Neste capítulo da dissertação, foi proposta uma investigação acerca de qual desempenho de classificação é possível se alcançar com uma CNN partindo de um cenário hipotético no qual o arranjo espacial de pixels não é conhecido. Para tal, as imagens de cada base foram usadas para estimar a informação mútua entre todos os pares de variáveis possíveis, e foi proposto um algoritmo que recebe como entrada essas estimativas e constrói um arranjo artificial de modo que relações de alta dependência entre variáveis sejam convertidas em relações de vizinhança espacial.

Os principais achados dessa investigação revelam que o procedimento proposto é capaz de reconstruir o aspecto visual das imagens com alto grau de fidelidade (Seção

4.6.1) e que, usando o princípio de agregar variáveis dependentes, é possível obter resultados quase tão bons quanto os de uma CNN treinada no cenário em que o arranjo natural dos pixels é conhecido (Seção 4.6.2). Ou seja, no contexto das CNNs, é possível substituir, em grande parte, o conhecimento *a priori* da estrutura espacial pelo conhecimento das dependências entre variáveis. Invertendo esse raciocínio, é possível argumentar que a utilidade do arranjo espacial nas imagens vem do fato de que ele localiza as dependências entre variáveis — i.e. são as dependências que fazem com que as vizinhanças formem estruturas plausíveis de serem exploradas.

Tendo isso em mente, uma hipótese bastante razoável é a de que o princípio de localizar dependências pode ser explorado para equipar dados nos quais não existem nenhuma ordenação natural das variáveis com uma estrutura espacial, a exemplo de dados tabulares ou sinais definidos em grafos. Essa estrutura artificial, por sua vez, pode ser explorada para impor a agregação entre variáveis dependentes nas redes neurais artificiais, potencialmente levando a vieses indutivos significativos como no caso das redes convolutivas com imagens. De fato, investigar tal hipótese é o objetivo do capítulo seguinte desta dissertação, que se concentra em aplicar tais princípios em bases de dados tabulares.

Além disso, espera-se que os resultados e as conclusões apresentadas aqui abram o caminho para uma interpretação alternativa para o sucesso das CNNs, que privilegia o argumento da agregação entre variáveis dependentes como uma alternativa aos argumentos baseados na geometria espacial na qual esses sinais estão definidos. Espera-se ainda que essa perspectiva motive estudos posteriores no contexto das CNNs. A título de exemplo, um problema em aberto no projeto de redes convolutivas é a determinação do campo receptivo ideal de cada neurônio (i.e. de quantas variáveis de entrada são agregadas). Dessa forma, uma sugestão para trabalhos futuros consiste em investigar uma possível relação entre o tamanho ideal para o campo receptivo e o tamanho das vizinhanças que englobam variáveis altamente dependentes.

Por fim, neste capítulo, também foi conduzida uma investigação adicional acerca da dimensão intrínseca das relações de dependência entre pixels nas imagens, de modo a determinar qual dimensão do espaço no qual tais relações podem ser codificadas através de distâncias espaciais entre pontos. Com exceção da base EuroSAT, os resultados obtidos evidenciam que, usando o estimador de Grassberger-Procaccia, é possível recuperar (através das estimativas de informação mútua) a informação de que os pixels de uma imagem natural devem estar dispostos em um grid de dimensão $d = 2$. Logo, também é natural imaginar que tal ferramenta pode ser usada para auxiliar na construção de arranjos artificiais para estruturas no qual tal informação não está disponível *a priori*, como é o caso dos dados tabulares.

5 Viés indutivo para dados tabulares baseado em medidas de informação mútua

5.1 Introdução do capítulo

Como mencionado no capítulo anterior, a onda moderna de interesse científico e tecnológico pelas redes neurais artificiais e pelo aprendizado profundo foi impulsionada no início dos anos 2010s pelo sucesso obtido com o uso de arquiteturas convolutivas em aplicações envolvendo imagens, sons, séries temporais e outros tipos de dados, através da exploração da estrutura espacial ou temporal intrínseca a estes, conforme explicado na Seção 4.2.3. Posteriormente, esse interesse foi fortalecido devido ao desenvolvimento de arquiteturas baseadas em mecanismos de atenção e seus resultados impressionantes em aplicações envolvendo o processamento de linguagem natural e, especialmente, a síntese textual, também explorando a natureza sequencial dos textos [80].

Em contrapartida, tal nível de sucesso ainda não foi observado em problemas que envolvem dados tabulares, isto é, aqueles que são apresentados em formato de tabelas, nas quais cada linha corresponde a um padrão (uma instância de dado) e cada coluna corresponde a um atributo, de modo que estas estejam ordenadas de maneira arbitrária e possuam significados semânticos potencialmente distintos, ao contrário das imagens, por exemplo, em que cada atributo corresponde a intensidade luminosa em uma posição da imagem. Para esse tipo de dado, não existe na literatura nenhum *a priori* óbvio a ser explorado (tal qual a estrutura em grid das imagens e o *a priori* da equivariância a translação) e nenhum viés indutivo consolidado para possibilitar o desenvolvimento de arquiteturas de redes neurais dominantes. Na verdade, durante muitos anos, o estado da arte foi dominado por modelos baseados em comitês de árvores de decisão [99, 100].

Nesse contexto, o fenômeno descrito acima tem atraído a atenção de diversos pesquisadores ao longo dos últimos anos [101, 102]. Para além da curiosidade científica de explicar essas observações, existem muitos argumentos práticos que justificam a relevância de desenvolver arquiteturas especializadas para processar dados tabulares. A título de exemplo, alguns desses argumentos foram pontuados em [103], como o fato de que modelos baseados em árvores não dão suporte a sistemas de aprendizado contínuo (*continual learning*), nem ao ciclo completo de treinamento de sistemas multimodais, nos quais dados tabulares são combinados com dados estruturados (e.g. imagens ou textos).

Assim, muitos esforços foram concentrados no desenvolvimento de tais arquiteturas. Entre os anos de 2019 e 2023, a comunidade apostou fortemente na aplicação dos

mecanismos de atenção e dos *transformers* ao processamento tabular, como evidenciado pelo desenvolvimento de arquiteturas como a *TabNet* [104], a *TabTransformer* [103], as *DANets* [105] e diversas outras. Apesar do avanço, os resultados sempre foram mistos, e, em certo sentido, contraditórios, à medida que os métodos propostos frequentemente falharam quando testados em novas bases de dados, para além das reportadas nas respectivas publicações, conforme pontuado em [101]. Como resultado, se tratando de redes neurais artificiais, as MLPs, que constituem, provavelmente, a forma mais *rudimentar* de aprendizado profundo, continuam sendo um *baseline* relevante, e muitas vezes difícil de ser superado, quando os dados são tabulares. Mais recentemente, o artigo publicado por Hollmann et. al. [106], em 2025, na Nature, parece sugerir uma mudança de paradigma na comunidade, à medida que propõe um modelo fundacional para dados tabulares, no qual milhões ou bilhões de problemas sintéticos são apresentados para o pré-treinamento de um grande modelo, também baseado em *transformers*, que é posteriormente ajustado para tarefas específicas. Com isso, os resultados obtidos finalmente parecem ter alcançado (e, em alguns casos, rompido) o estado da arte.

Em contrapartida, um ponto de vista bastante razoável é o de que descobrir vieses indutivos adequados pode possibilitar o desenvolvimento de métodos mais eficientes sem a necessidade de treinar grandes modelos em volumes de dados igualmente extensos, a exemplo do que aconteceu com as imagens, em que a dominância das CNNs é anterior ao desenvolvimento de modelos fundacionais. No caso das imagens, o viés indutivo que impulsionou os ganhos de desempenho se dá pela exploração de campos receptivos locais construídos através das noções de vizinhança inerente a estrutura geométrica sobre a qual esses sinais são amostrados. Entretanto, a investigação conduzida no Capítulo 5 revela que, mesmo após uma permutação aleatória das posições dos pixels, que descaracteriza completamente os campos receptivos locais, estes podem ser reconstruídos com alto grau de fidelidade usando o princípio de localização das dependências entre pares de variáveis aleatórias que representam pixels nas imagens.

Tendo isso em mente, uma questão que surgiu naturalmente durante a execução deste trabalho, e que motivou a investigação apresentada no capítulo atual, foi acerca da possibilidade de, usando o mesmo princípio descrito acima, equipar dados tabulares genéricos com campos receptivos locais significativos e com vieses indutivos *similares* aos das CNNs para o caso das imagens. Aqui, colocamos ênfase no adjetivo “similar”, uma vez que, conforme explicado anteriormente, um ingrediente relevante do viés indutivo das CNNs está na equivariância a translação proporcionada pelo compartilhamento de pesos. Sob o ponto de vista estatístico, tal prática se justifica devido ao pressuposto de estacionariedade, no sentido de que as relações de dependência seguem distribuições aproximadamente iguais ao longo de todos os campos receptivos locais, e, portanto, é razoável que elas sejam modeladas através de pesos compartilhados. Obviamente, em

cenários mais genéricos, tal pressuposto não deve ser assumido, fazendo com que o viés indutivo possa ser generalizado apenas de maneira parcial.

Assim, a principal contribuição deste capítulo não é a proposição de uma nova arquitetura, mas sim uma investigação acerca de vieses indutivos baseados em dependência estatística entre variáveis para dados tabulares. Para tal, a abordagem proposta consiste em equipar esses dados com uma ordenação baseada em dependências (i.e. variáveis que compartilham muita informação são postas em posições adjacentes), seguida pela aplicação de camadas localmente conectadas, que se assemelham as camadas convolutivas, a menos do compartilhamento de pesos.

Na literatura, a linha de pesquisa que mais se aproxima das motivações e da abordagem adotadas neste trabalho é a primeira geração das redes neurais em grafos modernas. Por essa razão, uma seção específica é dedicada a esses trabalhos, na qual são apresentadas uma breve revisão, conexões com o presente trabalho e as principais diferenças entre as abordagens.

Neste trabalho, por sua vez, os efeitos práticos dos campos receptivos locais construídos artificialmente são testados na tarefa de redução dimensional através de *autoencoders* [107]. Para fins de recapitulação, no contexto das redes neurais artificiais, os *autoencoders* são modelos especializados que mapeiam os dados de entrada em um espaço latente de dimensão reduzida (etapa de codificação), e, a partir dessa representação latente comprimida, reconstrói os dados de entrada com a maior acurácia possível (etapa de decodificação).

A abordagem originalmente escolhida para este trabalho foi a de aplicar a metodologia proposta em redes neurais com o objetivo final de classificação. No entanto, dois fatores motivaram a opção pela tarefa de autocodificação. O primeiro deles é que, na tarefa de classificação, os pares de variáveis que apresentam alta dependência estatística nem sempre são aqueles que, em conjunto, apresentam alto poder preditivo, e vice-versa. Um contra-exemplo emblemático disso é o caso do Ou-Exclusivo (XOR), quando aplicado a duas variáveis aleatórias binárias independentes. Nesse caso, as duas variáveis de entrada, em conjunto, determinam o rótulo sem ambiguidades, apesar de serem independentes. Dessa forma, para criar campos receptivos locais adequados para a tarefa de classificação, seria necessário abandonar as medidas de informação mútua pura entre pares de variáveis e usar medidas que levam em conta uma terceira variável (o rótulo). A segunda motivação reside no fato de que, no contexto do aprendizado de máquina moderno, as tarefas de extração automática de características e de aprendizado de representações (*representation learning*) são tão relevantes quanto, ou até mais relevantes que, o desenvolvimento de classificadores. Nesse contexto, a figura do *autoencoder* se torna central, sendo crucial em diversos temas atuais, tais como a adaptação de domínio (*domain adaptation*), o aprendizado auto-supervisionado (*self-supervised learning*) e os modelos generativos.

Assim, foram treinados *autoencoders* baseados na ordenação proposta e nas camadas localmente conectadas em diversos *benchmarks* tabulares. Para verificar a relevância dos campos receptivos baseados em dependência, os resultados foram comparados com a aplicação de estruturas localmente conectadas equivalentes em campos receptivos arbitrários (i.e., baseados em ordenações aleatórias de variáveis). E, para testar os seus potenciais benefícios em relação à abordagem padrão, os resultados foram comparados com aqueles de *autoencoders* densos, do tipo MLP. No primeiro caso, os resultados evidenciaram a significância dos campos receptivos locais, com a ordenação baseada em medidas de informação mútua produzindo reconstruções superiores aos de uma ordenação qualquer em essencialmente todos os experimentos. Quanto as comparações com arquiteturas densas, os resultados foram mistos. Em alguns *datasets*, as estruturas localmente conectadas produziram resultados superiores (do ponto de vista de qualidade de reconstrução), enquanto em outros os resultados foram equivalentes ou ligeiramente inferiores aos de uma arquitetura densa com mesma capacidade paramétrica. Em posse desses resultados, são feitas análises e levantadas hipóteses acerca de quais direções são promissoras para investigações futuras, como base nos achados.

As contribuições resumidas nos parágrafos acima estão diluídas ao longo do restante deste capítulo da seguinte forma. Na Seção 5.2, as primeiras arquiteturas de redes neurais em grafos modernas são explicadas, com foco nas conexões com a proposta deste capítulo da dissertação. Na Seção 5.3 são apresentados os detalhes algorítmicos da abordagem proposta, com suas devidas justificativas. A Seção 5.4 apresenta os experimentos e as discussões pertinentes, enquanto as conclusões finais são apresentadas na Seção 5.5;

5.2 Relações com as redes neurais em grafos

As redes neurais em grafos (*Graph Neural Networks*, GNNs) se tornaram um tema bastante relevante no aprendizado de máquina moderno, tendo ganhado tração ao longo dos últimos 15 anos [108]. Nos anos recentes, elas se tornaram o ferramental padrão para tarefas de aprendizado envolvendo grafos, como classificação de nós e predição de links, e encontraram aplicações em diversos domínios, e.g., predição de reações químicas, sistemas de recomendação, redes de tráfego etc [105, 109, 110].

Em um sentido amplo, as GNNs são tipicamente categorizadas em GNNs espectrais e *message passing neural networks* (MPNNs), refletindo dois paradigmas distintos que se desenvolveram ao longo dos anos [111]. O primeiro deles (as GNNs espectrais) são baseados em transformações que operam na representação espectral de sinais definidos em grafos, conforme explicado na Seção 5.2.1. As MPNNs, por sua vez, são baseadas em operadores que agem diretamente na topologia dos grafos (sem a necessidade da obtenção do espectro), e eventualmente se tornaram a abordagem dominante e mais estudada,

através de trabalhos como aqueles publicados em [112, 113], e do advento de arquiteturas que alcançam resultados correspondentes ao estado da arte nas aplicações mencionadas no parágrafo acima.

Em contrapartida, ao analisar as primeiras referências dessa onda moderna [114, 115, 116, 108], que são, em grande parte, dominadas pelas GNNs espectrais, torna-se evidente que o foco ainda não estava nessas aplicações, mas sim em uma motivação que carrega muitas conexões, teóricas e algorítmicas, com a investigação conduzida neste capítulo da dissertação: a de generalizar o sucesso encontrado pelas CNNs — em sinais amostrados em grids — para domínios não estruturados.

Nessas referências, que foram publicadas entre 2013 e 2017, os grafos emergiam, de maneira artificial, apenas como um meio comum para definir sinais que não são naturalmente equipados com uma estrutura espacial. Embora a literatura tenha, posteriormente, migrado para domínios nos quais estruturas de grafos aparecem de maneira inerente, o objetivo desta seção é descrever brevemente as primeiras arquiteturas de GNNs (as espectrais), e colocar em contexto como elas se relacionam com as motivações teóricas e as abordagens práticas adotadas neste capítulo.

Para tal, adotamos as seguintes notações e definições. Um grafo ponderado G é definido como um par $(\Omega, \mathbf{W})$, em que $\Omega = \{1, \ldots, m\}$ é um conjunto de nós e $\mathbf{W} \in \mathbb{R}^{m \times m}$ é a matriz simétrica de adjacências ponderadas, que satisfaz $W_{ii} = 0$ para todo $i = 1, \ldots, m$, e $W_{ij} \geq 0$ para todo $i \neq j$. Consideramos ainda um sinal definido no grafo G como uma função que associa a cada nó de G um valor (um escalar) ou um vetor (como em um sinal de múltiplos canais). Isto é, da mesma forma que em um sinal definido no domínio do tempo cada amostra é indexada por um instante de tempo, e em uma imagem cada amostra é indexada por uma coordenada espacial, em um sinal definido em G cada observação é indexada por um nó em Ω.

Naturalmente, uma estrutura em grid pode ser interpretada como um grafo, em que cada célula do grid corresponde a um nó, e a matriz $\mathbf{W}$ reflete as próprias adjacências da estrutura. Dessa forma, as definições adotadas acima comportam os sinais definidos em grids, nos quais as CNNs são aplicadas, enquanto também podem ser usadas para definir sinais oriundos de domínios mais genéricos, sem uma estrutura espacial (ou sem o uso dela, caso exista). É de se imaginar, portanto, que a definição de sinais em grafos emerge como um suporte natural para estudos acerca da generalização das CNNs para domínios não estruturados, que por sua vez é posta de maneira explícita como a motivação principal para as GNNs nas referências citadas ao longo desta seção.

5.2.1 GNNs espectrais

A primeira tentativa moderna de generalizar as CNNs para sinais definidos em grafos, em [114], teve suas raízes na comunidade do processamento de sinais em grafos [117], através das GNNs espectrais. Elas são baseadas na ideia de que, dispondo de uma generalização adequada para a operação de convolução, é possível definir camadas convolucionais em grafos que engendrem as mesmas propriedades de aprendizado que as CNNs já engendram para sinais definidos em grids.

Nesse contexto, como mencionado em [117], grande parte dos esforços para o desenvolvimento de ferramentas para processar sinais definidos em grafos são concentrados na generalização dos operadores fundamentais do processamento digital de sinais (PDS) convencional, a exemplo da transformada de Fourier e da operação de convolução. O primeiro fornece uma representação espectral (i.e. no domínio da frequência) para um sinal e está no centro da análise de sinais, enquanto a convolução é o operador que descreve a classe mais importante de filtros no processamento digital de sinais.

Em particular, no PDS clássico, esses dois conceitos estão intrinsecamente relacionados, através do teorema da convolução [118], o qual afirma que a transformada de Fourier da convolução entre dois sinais (e.g. no domínio do tempo) equivale ao produto entre eles no domínio da frequência, conforme descrito na equação

$$\mathbf{x} * \mathbf{y} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{x}) \odot \mathcal{F}(\mathbf{y})), \tag{5.1}$$

em que $\mathbf{x}$ e $\mathbf{y}$ são dois sinais, $*$ denota a convolução (circular, para o caso discreto), $\odot$ denota o produto de Hadamard, $\mathcal{F}(\cdot)$ e $\mathcal{F}^{-1}(\cdot)$ denotam a transformada de Fourier (discreta, para o caso discreto) e a sua inversa, respectivamente.

Invertendo esse raciocínio, a convolução pode ser generalizada para sinais definidos em grafos através do produto entre as representações espectrais dos mesmos, uma vez que se disponha de uma generalização pertinente para a noção de espectro. O conceito de frequência, por sua vez, pode ser generalizado da seguinte forma. Se $\mathbf{x} = (x_1, \ldots, x_m)$ é um sinal definido em $G = (\Omega, \mathbf{W})$, uma medida natural para a frequência de $\mathbf{x}$ é dada por

$$\|\nabla \mathbf{x}\|^2 = \sum_{i=1}^{m} \sum_{j=1}^{m} W_{ij}(x_i - x_j)^2. \tag{5.2}$$

Ou seja, para sinais de baixa frequência, duas amostras x_i, x_j indexadas por nós adjacentes (i.e., W_{ij} possui valor elevado) tendem a terem valores semelhantes. Sob essa definição, os sinais de menor frequência, como é de se esperar, são os sinais constantes, em que $x_1 = \cdots = x_m$.

Assim, para generalizar a transformada de Fourier para sinais definidos em G, um caminho natural é encontrar uma base ortonormal de sinais $\mathbf{v}_1, \ldots, \mathbf{v}_m$, com frequências

$\|\nabla \mathbf{v}_1\|^2 < \cdots < \|\nabla \mathbf{v}_m\|^2$.

Da teoria espectral de grafos [119, 120], sabe-se que uma base que satisfaz essas propriedades é dada pelos autovetores do grafo laplaciano, um operador que generaliza o Laplaciano em grids, definido por $\mathbf{L} = \mathbf{D} - \mathbf{W}$, em que $\mathbf{D}$ é uma matriz diagonal cujas entradas são dadas por $D_{ii} = \sum_{j=1}^{m} W_{ij}$. De fato, se $\mathbf{v}_1, \ldots, \mathbf{v}_m$ forem escolhidos como os autovetores (normalizados) de $\mathbf{L}$, eles satisfazem as relações da forma [1]

$$\mathbf{v}_i = \arg \min_{\substack{\mathbf{x} \in \mathbb{R}^m \\ \|\mathbf{x}\|=1 \\ \mathbf{x} \perp \{\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}\}}} \|\nabla \mathbf{x}\|^2, \tag{5.3}$$

enquanto os autovalores podem ser interpretados como as frequências, i.e., $\|\nabla \mathbf{v}_i\|^2 = \lambda_i$. Logo, a transformada de Fourier de um sinal $\mathbf{x}$, definido em G, e a sua inversa, podem ser convenientemente definidas por

$$\tilde{\mathbf{x}} = \mathbf{F}\mathbf{x}, \tag{5.4}$$

$$\mathbf{x} = \mathbf{F}^t \tilde{\mathbf{x}}, \tag{5.5}$$

respectivamente. Aqui, $\mathbf{F}$ é a matriz cujas linhas correspondem aos autovetores do Laplaciano do grafo G, enquanto $\tilde{\mathbf{x}}$ é o espectro de $\mathbf{x}$, de modo que $\tilde{x}_i$ corresponde ao coeficiente associado à componente de frequência λ_i.

Assim como ocorre na transformada de Fourier discreta (DFT) para sinais definidos no domínio do tempo, a primeira linha da matriz $\mathbf{F}$ é o vetor constante, associado à componente $\lambda_1 = 0$. [2]. Uma outra semelhança inerente a essa construção está no fato de que, no PDS clássico, a base ortonormal que define a DFT é composta pelos autovetores de operadores circulantes [3], incluindo o Laplaciano no grid. Aqui, a base que define a transformada de Fourier em G também é composta pelos autovetores do operador Laplaciano em G.

Dando um passo adiante, a convolução entre dois sinais $\mathbf{x}$ e $\mathbf{y}$ pode ser convenientemente definida por

$$\mathbf{x} * \mathbf{y} = \mathbf{F}^t \cdot (\mathbf{F}\mathbf{x} \odot \mathbf{F}\mathbf{y}), \tag{5.6}$$

em que $\odot$ representa o produto de Hadamard.

Essa foi exatamente a definição da convolução em grafos que foi adotada em [114], artigo no qual foi publicada a primeira instanciação das GNNs espectrais. Mais especificamente, esse trabalho define uma camada convolutiva, que mapeia um sinal de entrada

[1] A verificação dessas relações pode ser encontrada nas referências [119, 120].

[2] Isso emerge como uma consequência natural da definição grafo Laplaciano, uma vez que a soma das linhas da matriz $\mathbf{L}$ é 0, o que implica na existência de um autovalor nulo, cujo autovetor associado é o vetor constante.

[3] Operadores circulantes são operadores lineares invariantes a deslocamentos circulares, representados por matrizes circulantes. Em Processamento Digital de Sinais, eles descrevem a operação de convolução circular e possuem a propriedade de serem diagonalizados pela DFT (ver capítulo 8 em [121]).

$\mathbf{x} \in \mathbb{R}^{m \times k_i}$ definido em G, com k_i canais, em um sinal de saída $\mathbf{y} \in \mathbb{R}^{m \times k_o}$, também definido em G, com k_o canais, através da equação

$$y_{:,j} = \mathbf{F}^t \left(\sum_{i=1}^{k_i} \tilde{k}_{:,i}^{(j)} \odot (\mathbf{F} x_{:,i}) \right) ; \; j = 1, \ldots, k_o. \tag{5.7}$$

Na equação acima, $x_{:,i}$ e $y_{:,j}$ representam o i-ésimo e o j-ésimo canal dos sinais $\mathbf{x}$ e $\mathbf{y}$, respectivamente, enquanto $\tilde{k}_{:,i}^{(j)}$ representa o espectro do i-ésimo canal no j-ésimo filtro. Sob essa definição, os parâmetros a serem ajustados durante o treinamento são os coeficientes dos espectros de cada um dos k_0 filtros.

Uma diferença notável entre essa generalização e as CNNs convencionais está no fato de que a quantidade de coeficientes em cada filtro é igual à quantidade de coeficientes do sinal de entrada, enquanto que nas camadas convolutivas convencionais essa quantidade de coeficientes é fixa, uma vez que os filtros são localizados no espaço, e portanto, restritos a um suporte compacto. Obviamente, essa é uma diferença relevante, pois ela diz respeito a dois dos principais argumentos para o uso das CNNs: a eficiência paramétrica, e a extração local de características.

Em [114], esse desencaixe é contornado através da generalização do princípio da dualidade, que também é conhecido do processamento de sinais definidos em grids. Especificamente, sinais cujos coeficientes decaem de maneira abrupta, o que ocorre nos filtros locais, possuem espectros suaves (i.e., localidade no grid implica espalhamento no domínio espectral). Assim, os autores propõem que a localidade das CNNs convencionais seja generalizada para as GNNs espectrais através da imposição de suavidade no espectro dos filtros. Assim, apenas um conjunto subamostrado de coeficientes espectrais são ajustados durante o treinamento, enquanto o restante deles é determinado através de um *kernel* de interpolação, que impõe suavidade.

Após a primeira GNN espectral, publicada em 2013, diversas extensões foram propostas ao longo dos anos subsequentes. Por exemplo, em [116], Henaff et. al. propôs uma maneira de introduzir as camadas de *pooling* nas GNNs espectrais, usando *clustering* hierárquico de nós para possibilitar a redefinição dos sinais em múltiplas escalas. Ainda na mesma linha, um outro trabalho notável foi o de Defferrard et. al., em [115]. Nele, a principal inovação foi uma nova forma de impor a localidade espacial nos filtros espectrais, na qual a resposta em frequência destes são representadas por polinômios de Chebyshev, cujos coeficientes são determinados durante o treinamento. Essa nova representação permitiu implementações mais eficientes das GNNs espectrais, aumentando a aplicabilidade dessas arquiteturas.

5.2.2 Conexões e distinções com este trabalho

A conexão entre as GNNs espectrais e este trabalho pode ser articulada da seguinte forma. Considere uma base de dados de natureza genérica contendo padrões multivariados pertencentes ao $\mathbb{R}^d$. Ou seja, cada padrão representa uma instância de um vetor aleatório[4] $(X_1, \ldots, X_d)$. Dada uma medida de associação, denotada por $A(i, j)$, que quantifica algum aspecto pertinente de relação entre o par de variáveis (X_i, X_j), é possível definir um grafo $G = (\Omega, \mathbf{W})$, em que $\Omega = \{1, \ldots, d\}$ e $W_{ij} = A(i, j)$.

Dessa definição, decorre que qualquer base de padrões multivariados pode ser reinterpretada como uma coleção de sinais definidos em G, e portanto apropriada para o processamento por meio de GNNs. De fato, esse tipo de reinterpretação era comum nas primeiras referências sobre GNNs espectrais, pois, como mencionado anteriormente, os grafos eram definidos artificialmente, e não parte da estrutura intrínseca dos *benchmarks* adotados nos experimentos. Por exemplo, a base de dados MNIST é utilizada nas referências [114, 115], mostrando resultados competitivos com os de CNNs convencionais, enquanto as referências [115, 116] utilizaram bases de documentos em texto, representadas usando o modelo *bag-of-words* [122].

Sob o ponto de vista deste trabalho, um caso de interesse particular se dá quando a medida de associação usada reflete o grau de dependência estatística entre os pares de variáveis, a exemplo das medidas de informação mútua usadas ao longo deste trabalho, ou das medidas de correlação usadas em [76, 77]. Nesse caso, fica evidente que as arquiteturas de GNNs são modelos que impõem a agregação de variáveis dependentes, reforçando a conexão com este trabalho.

No entanto, a exceção de [123], os trabalhos da época não se utilizaram explicitamente de medidas de dependência entre variáveis. Por exemplo, em [115], os autores constroem o grafo para a base de textos através da distância entre os *word embeddings* de cada palavra[5], e das distâncias entre as posições dos pixels para a base MNIST, enquanto [116] propõe métodos supervisionados para a determinação do grafo. De modo geral, é possível afirmar que nenhum dos trabalhos encontrados é posicionado como uma investigação acerca de vieses indutivos baseados em medidas de dependência para dados tabulares, nem tampouco focam no aprendizado de representações e na autocodificação destes, à medida que os modelos eram sempre treinados para as tarefas de classificação ou regressão.

Adicionalmente, as GNNs espectrais, que essencialmente replicam as CNNs se o grafo for definido conforme o grid convencional das imagens, implicitamente engendram o

[4] Uma lista ordenada de variáveis aleatórias

[5] Aqui vale destacar a existência de uma relação implícita, uma vez que os *word embeddings* são determinados através das contagens de co-ocorrência, que, como argumentado na Seção 2.7 refletem a dependência entre eventos.

mecanismo de compartilhamento de pesos, fazendo delas inadequadas para estudos com dados tabulares genéricos. Como argumentado anteriormente, nesses cenários, a hipótese de estacionariedade ao longo do grafo não deve ser assumida de maneira automática, o que motivou a opção por camadas localmente conectadas no presente trabalho.

5.3 Abordagem proposta

Nesta seção, são explicados o passo a passo da abordagem proposta e os seus detalhes algorítmicos. Mais especificamente, nas Seções 5.3.1 e 5.3.2 são descritos os procedimentos para a criação de um arranjo espacial artificial para dados tabulares, baseado nas dependências entre pares de variáveis, enquanto a Seção 5.3.3 introduz a implementação das camadas localmente conectadas. É importante destacar que, sob o ponto de vista deste trabalho, as implementações e os procedimentos aqui utilizados são vistos como uma simples instanciação dos princípios que motivam esta investigação, isto é, a criação de estruturas locais baseadas em dependência e a agregação de variáveis dependentes como um viés indutivo para arquiteturas profundas. Dessa forma, tais procedimentos não foram exaustivamente ajustados com o objetivo de produzir resultados superiores, mas permitiram a experimentação acerca das ideias propostas e testar as hipóteses subjacentes.

5.3.1 Construção do mapa de informação mútua

Para começar, considere um conjunto de dados $\mathcal{X}$ com N padrões no $\mathbb{R}^d$. Isto é, cada instância de dado é representada por um vetor de d variáveis $X_1, \ldots, X_d$. O objetivo final é usar esse conjunto de dados para treinar uma estrutura de auto-codificação capaz de projetar qualquer amostra instanciada a partir da mesma distribuição que $\mathcal{X}$ em um espaço de dimensão reduzida $k < d$, de modo que a entrada possa ser reconstruída com a menor distorção possível.

Assim como no Capítulo 4, o primeiro passo para a exploração da agregação de variáveis dependentes como um viés indutivo para os *autoencoders* é quantificar o grau de dependência estatística entre todos os pares de variáveis possíveis. Para tal, novamente foram usadas estimativas de informação mútua obtidas usando todas as amostras dos pares em $\mathcal{X}$ como entradas para o estimador KSG. Adicionalmente, para trabalhar com medidas normalizadas, entre 0 e 1, a informação mútua estimada para cada par foi dividida pela informação mútua estimada máxima, dentre todos os pares.[6] Em posse de todas essas

[6] A escolha por esse tipo de normalização se deu pois foi verificado que a medida de informação mútua normalizada para variáveis aleatórias contínuas, usada no Capítulo 4, distorcia as diferenças entre diferentes pares de modo que dificultasse as etapas posteriores.

estimativas, é possível preencher uma matriz $\mathbf{C}$ em que cada entrada

$$C_{ij} = \frac{\hat{I}(X_i; X_j)}{\max\limits_{i,j} \hat{I}(X_i; X_j)} \tag{5.8}$$

representa uma estimativa do grau de dependência entre X_i e X_j. Como pode ser visto, essa matriz é análoga à matriz $\mathbf{C}$ que foi preenchida na Seção 4.6.1.

5.3.2 Ordenação baseada em informação mútua

Em posse de todas as estimativas de informação mútua, e, consequentemente, da matriz $\mathbf{C}$ definida na seção anterior, o próximo passo consiste em construir um arranjo espacial para as d variáveis, no qual pares de variáveis que compartilham alta dependência estatística são postos em posições vizinhas. Isso também remete ao que foi feito no capítulo anterior, quando variáveis que representam pixels em um conjunto de imagens permutadas foram mapeadas em pontos no $\mathbb{R}^2$. Para o caso de dados genéricos, em que não existe uma escolha óbvia para a dimensão, optamos por um arranjo unidimensional, em contraste às duas dimensões usadas para remontar as adjacências das imagens.

Assim como no Capítulo 4, a definição de tal arranjo pode ser interpretada como análoga ao problema de encontrar *embeddings* para os nós de um grafo ponderado, uma vez que cada variável pode ser interpretada como um nó e a matriz $\mathbf{C}$ pode ser interpretada como a matriz de adjacências de um grafo. Novamente, diversas abordagens podem ser usadas aqui, como aquelas baseadas em escalamento multidimensional, ou aquelas baseadas na teoria espectral de grafos [124]. Usando qualquer uma delas, o próximo passo para possibilitar o processamento através das camadas localmente conectadas seria a discretização desses *embeddings* contínuos em uma estrutura em grid.

No entanto, dado a escolha por um arranjo unidimensional, os dois passos descritos no parágrafo acima podem ser interpretados como análogos ao problema de encontrar uma reindexação das d variáveis de modo que índices adjacentes correspondam a variáveis que compartilhem alta dependência estatística. Neste trabalho, essa segunda interpretação foi preferida, por questões de simplicidade.

Novamente, diversas abordagens podem produzir ordenações com as propriedades desejadas. Em particular, as implementações de algoritmos de *clustering* hierárquico acompanham rotinas de ordenação das entradas de modo que não ocorra interseção entre os ramos no dendrograma [7], constituindo uma solução suficientemente simples e direta para esse problema de reindexação.

Para os leitores com pouca familiaridade, relembramos que os algoritmos de *clustering* hierárquico organizam os elementos de entrada em uma árvore binária que agrupa

[7] O dendrograma é um diagrama de árvore que exibe os grupos formados por agrupamento (*clustering*) de observações em cada passo e em seus níveis de similaridade.

elementos similares em conjunto. O processo começa com a determinação de uma medida de dissimilaridade entre todos os pares de elementos de entrada. Inicialmente, cada elemento corresponde a um *cluster* diferente. Em cada passo, os dois *clusters* com maior similaridade são aglutinados, e o processo é repetido até que todos os elementos sejam agrupados em um único *cluster*. O Dendrograma, por sua vez, é a representação gráfica da árvore gerada, e, portanto, um algoritmo de ordenação que evita a interseção entre ramos, naturalmente, posiciona elementos similares, que tendem a ser agrupados nos primeiros passos da hierarquia, como vizinhos. Uma descrição mais detalhada dos algoritmos de *clustering* hierárquico pode ser encontrada em [125].

Dessa forma, pode-se definir uma matriz de dissimilaridades $\mathbf{D}$, com entradas $D_{ij} = 1 - C_{ij}$ que correspondem a uma medida de dissimilaridade baseada em dependência estatística para as variáveis X_i e X_j. Assim, o conjunto de d variáveis pode ser usado como entrada para um algoritmo de *clustering* hierárquico, e uma rotina de ordenação pode ser usada para reindexá-las. Em particular, foi utilizado um algoritmo de ordenação baseado em *average link*, no qual a dissimilaridade entre dois *clusters* é definida como a média de todas as dissimilaridades entre os elementos de um *cluster* e os elementos do outro. A ordenação, por sua vez, foi obtida usando o algoritmo de Bar Joseph, proposto em [126]. Como uma ilustração dos efeitos da ordenação, consideremos o exemplo a seguir.

Exemplo: Seja $\{X_1, X_2, X_3, X_4, X_5\}$ uma coleção de variáveis para o qual a matriz $\mathbf{D}$, conforme definida anteriormente, é dada por

$$\mathbf{D} = \begin{bmatrix} 0 & 0.6 & 0.15 & 0.5 & 0.18 \\ 0.6 & 0 & 0.7 & 0.1 & 0.5 \\ 0.15 & 0.7 & 0 & 0.6 & 0.2 \\ 0.5 & 0.1 & 0.6 & 0 & 0.4 \\ 0.18 & 0.5 & 0.2 & 0.4 & 0 \end{bmatrix}. \tag{5.9}$$

Olhando para matriz, é possível perceber a existência de dois grupos naturais $\{X_2, X_4\}$ e $\{X_1, X_3, X_5\}$ de variáveis que compartilham informação, e que portanto resultam em baixos valores de D. De fato, ao aplicar o *average link*, $\{X_2\}$ e $\{X_4\}$ são agrupados no primeiro passo da hierarquia, com uma dissimilaridade $D = 0.1$, enquanto $\{X_1\}$ e $\{X_3\}$ são agrupados no segundo passo com $D = 0.15$. Em seguida, os grupos $\{X_1, X_3\}$ e $\{X_5\}$ são agrupados com uma dissimilaridade média $D = 0.19$, até que, finalmente, todas as variáveis são agrupadas em um único *cluster*.

Essa sequência hierárquica de agrupamentos é ilustrada através do dendrograma na Figura 17, no qual as cinco variáveis estão ordenadas conforme a ordenação obtida usando o algoritmo de Bar Joseph [126].

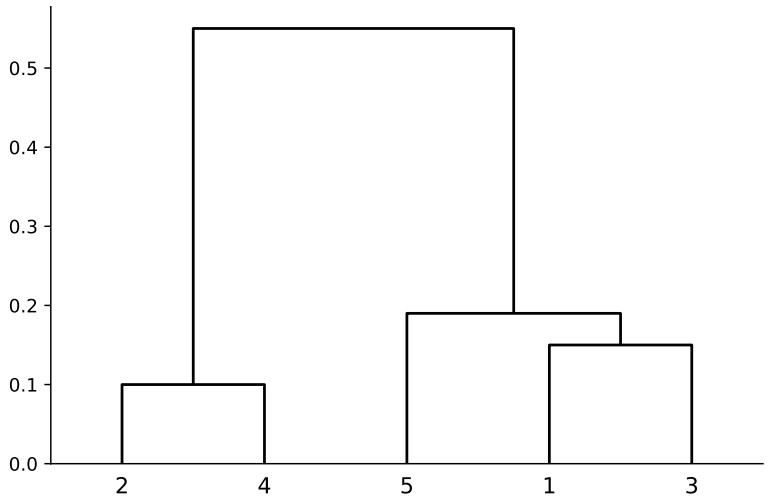


Figura 17 – Dendrograma para o *average link* aplicado ao conjunto de 5 variáveis cuja matriz **D** é definida na Equação 5.9

> Como pode ser visto, a sequência original $(1, 2, 3, 4, 5)$, para a qual haveria cruzamento de ramos, foi substituída pela sequência $(2, 4, 5, 1, 3)$. Diversas outras sequências também iriam garantir a ausência de cruzamentos entre ramos da árvore, como, por exemplo, $(4, 2, 3, 5, 1)$ ou $(2, 4, 5, 1, 3)$. No entanto, o algoritmo de Bar Joseph produz a ordenação ótima sob o critério do menor somatório das dissimilaridades entre pares de variáveis adjacentes. Por esse motivo, a variável X_1 é posicionada entre as variáveis X_3 e X_5, com $D_{13} = 0.15$ e $D_{15} = 0.18$, que são menores que $D_{35} = 0.2$. Da mesma maneira, a ordem $(2, 4)$ é preferida em relação à ordem $(4, 2)$, pois X_4 e X_1 possuem a menor dissimilaridade entre variáveis dos dois grupos.

Embora o passo a passo do algoritmo implementado para minimizar tal critério tenha sido omitido deste documento, os leitores interessados são direcionados para a referência [126]. Para os propósitos desta dissertação, o exemplo acima cumpre o objetivo de ilustrar como esse método de ordenação equipa um conjunto de variáveis genéricas com estruturas locais nas quais variáveis adjacentes compartilham altos valores de informação mútua. Dessa forma, a aplicação das camadas localmente conectadas — a serem explicadas na próxima seção — permite investigar os efeitos da imposição da agregação de variáveis dependentes como um viés indutivo para variáveis genéricas.

5.3.3 Camadas localmente conectadas

Camadas localmente conectadas são um tipo de mecanismo utilizado nas redes neurais artificiais em que, assim como nas camadas convolucionais, cada unidade na saída está conectada apenas a um pequeno subconjunto das unidades de entrada. Entretanto, diferente das camadas convolucionais, não há compartilhamento de parâmetros. Ou seja,

filtros independentes são aprendidos para cada *patch* do padrão de entrada, configurando assim um meio termo entre as camadas convolucionais e as camadas densas.

No contexto deste trabalho, o uso de tais camadas, em combinação com o arranjo espacial proposto na seção anterior, permite investigar os efeitos da imposição da agregação entre variáveis dependentes para dados tabulares, sem as restrições adicionais que são impostas pelo compartilhamento de pesos nas CNNs. Dessa forma, o objetivo desta seção é apresentar a implementação aqui proposta para as camadas localmente conectadas.

É importante ressaltar, no entanto, que o uso desse tipo de estrutura não constitui uma inovação. Na verdade, as primeiras arquiteturas neurais para sistemas de visão eram baseadas apenas em conectividade local [61], enquanto o compartilhamento de pesos foi popularizado posteriormente, com a arquitetura LeNet [59]. Mesmo em algumas arquiteturas modernas, camadas localmente conectadas ainda são aplicadas, como é o caso da *DeepFace* [127], publicada em 2014. A justificativa, apresentada de maneira explícita pelos autores, é a de que o sistema proposto para reconhecimento facial envolve um primeiro passo de alinhamento dos rostos, e, portanto, as incertezas causadas por translações são essencialmente removidas. Logo, a arquitetura aplicada utiliza-se de convoluções nas primeiras camadas para extrações de características de baixo nível (como bordas), enquanto o compartilhamento de parâmetros é abandonado nas camadas posteriores para obter extratores de características especializados para cada região do rosto.

Apesar desse histórico, a ideia de explorar tais estruturas em arranjos espaciais artificiais baseados em dependência como um viés indutivo para dados tabulares não foi explorada anteriormente e, portanto, constitui um caráter inovador dessa investigação, até onde é de conhecimento do autor.

Para a implementação proposta, assim como nas camadas convolutivas, o padrão de entrada é escaneado usando janelas deslizantes. Aqui dois parâmetros importantes são o tamanho da janela w e o *stride* s, que controla a sobreposição entre *patches* consecutivos. Com isso, cada patch de entrada é mapeado em uma janela correspondente na saída contendo k unidades através de uma projeção afim. O passo a passo desse simples processo é apresentado no Algoritmo 1.

5.4 Experimentos e resultados

5.4.1 Bases de dados

Nesta seção, *autoencoders* construídos à base de camadas localmente conectadas são empiricamente avaliados usando uma coleção de sete *benchmarks* tabulares, que está descrita conforme a Tabela 7.

Nesta tabela, o $*$ que acompanha os nomes de algumas das bases indica que uma

Algoritmo 1 Propagação direta de uma camada localmente conectada.

Hiperparâmetros: Tamanho da janela deslizante: w, Unidades de saída por janela deslizante: j, *Stride*: s
Parâmetros do modelo: Pesos: $\mathbf{W}_1, \ldots, \mathbf{W}_l \in \mathbb{R}^{w \times k}$, Vieses: $\mathbf{b} \in \mathbb{R}^m$
Entrada: $\mathbf{x} \in \mathbb{R}^n$
Saída: $\mathbf{y} \in \mathbb{R}^m$

$l \leftarrow \left\lceil \frac{n-j}{s} \right\rceil$ ▷ Quantidade de janelas deslizantes
$n_p \leftarrow (l-1)s + w$ ▷ Dimensão da entrada após o *padding*
$p \leftarrow n_p - n$ ▷ Tamanho do *padding*
$\tilde{\mathbf{x}} \leftarrow [\, \mathbf{x} \; \mathbf{0}_p \,]$ ▷ *Padding*
$m \leftarrow lp$ ▷ Dimensão da saída
$\mathbf{y} \leftarrow \mathbf{0}_m$ ▷ Inicialização da saída
$\mathbf{y} \leftarrow \mathbf{y} + \mathbf{b}$

Tabela 7 – Lista de bases de dados utilizadas.

Base de dados	# amostras	# atributos	# classes
isolet [128]	7795	617	26
christine* [129, 130]	5418	1503	2
gina [131, 130]	3468	970	2
dilbert [132, 130]	10000	2000	5
scene [133, 45]	2407	294	2
gisette* [134, 130]	7000	1600	2
guillermo* [135, 130]	20000	2000	2

etapa de filtragem de atributos foi realizada antes dos experimentos. Especificamente, foram removidos atributos cuja distribuição apresentava concentração em um número reduzido de valores distintos, caracterizando variáveis de natureza discreta. Essa filtragem foi realizada devido à restrição imposta pelo estimador de informação mútua adotado neste trabalho, que é formulado para estimar a informação mútua entre pares de variáveis contínuas.

Consequentemente, o número de atributos reportado na tabela para essas bases corresponde às versões filtradas utilizadas nos experimentos e difere do número de atributos presente nas versões originais das bases, conforme descrito nas referências correspondentes.

5.4.2 Protocolo experimental

O protocolo experimental utilizado para testar as hipóteses propostas é descrito conforme a seguinte sequência.

(i) Cada uma das bases de dados é particionada aleatoriamente, de modo que 20% das amostras são reservadas para teste, enquanto as amostras restantes são utilizadas

para treino/validação.

(ii) Os atributos de ambos os conjuntos são ordenados conforme o passo a passo descrito na Seção 5.3, que são obtidos através de medidas de informação mútua estimadas usando apenas os dados separados para treinamento/validação. Ou seja, em nenhum momento os dados de teste são usados para o aprendizado da estrutura espacial artificial. Após a filtragem, os dados foram escalonados conforme as estatísticas da base de treino/validação (média e desvio padrão).

(iii) Modelos de *autoencoders* construídos a base de camadas localmente conectadas foram treinados. Ajustes foram feitos usando a base de treinamento/validação para determinar o tamanho ideal para os campos receptivos (i.e. o tamanho da janela deslizante em cada camada). Após a determinação da arquitetura, como *baseline*, *autoencoders* densos (do tipo MLP) com aproximadamente a mesma quantidade de parâmetros foram treinados, com o objetivo de verificar se o viés indutivo baseado em dependências apresenta ganhos em relação a abordagem convencional. Adicionalmente, a mesma arquitetura localmente conectada é treinada usando um arranjo espacial com atributos ordenados aleatoriamente. Essa última comparação é feita com o objetivo de verificar a relevância dos campos receptivos baseados em dependência, isolando a hipótese de que os potenciais ganhos resultam apenas da imposição de um determinado padrão de esparsidade.

(iv) Durante os experimentos, as configurações e os hiperparâmetros associados ao treinamento dos modelos foram mantidos fixos. Especificamente, a função de custo utilizada foi o erro quadrático médio (EQM), o otimizador utilizado foi Adam [136], com taxa de aprendizado $\alpha = 0.001$. O treinamento de cada um dos modelos se deu ao longo de 50 épocas, e, para a avaliação, foram consideradas as métricas extraídas após a última época.

(v) Como dito anteriormente, a tarefa considerada é a de redução dimensional. Logo, para avaliar o mérito de cada modelo de autocodificação, a principal métrica utilizada são os valores de EQM na base de teste. Adicionalmente, são reportados os valores de acurácia balanceada para classificadores treinados com os dados projetados no espaço latente. Para a tarefa de classificação no espaço latente, foi adotado um classificador baseado em SVM (máquina de vetores de suporte). Cada modelo foi treinado 20 vezes, em cada um dos *datasets*, com o objetivo de analisar como as métricas de avaliação se distribuem ao longo de múltiplas execuções independentes.

5.4.3 Resultados

Para cada um dos modelos de *autoencoders* treinados, os erros de reconstrução estão apresentados nas Tabelas 8, 10, 12, para dimensões latentes $k = 10, 20, 30$, respecti-

vamente. Da mesma maneira, as acurácias obtidas pelas SVMs nos espaços latentes estão exibidas nas Tabelas 9, 11, 13.

Tabela 8 – Erro quadrático médio (± desvio padrão) de reconstrução obtidos pelos diferentes *autoencoders* ($k = 10$) em cada conjunto de dados.

Dataset	AE denso	AE-LC + ord. aleatória	AE-LC + ord. proposta
isolet	$\underline{0.3676 \pm 0.0013}$	0.3967 ± 0.0026	$\mathbf{0.3607 \pm 0.0022}$
christine	$\underline{0.6527 \pm 0.0008}$	0.6606 ± 0.0028	$\mathbf{0.6487 \pm 0.0010}$
gina	$\underline{0.6091 \pm 0.0046}$	0.7061 ± 0.0056	$\mathbf{0.5630 \pm 0.0046}$
dilbert	$\underline{0.3283 \pm 0.0016}$	0.3416 ± 0.0031	$\mathbf{0.2887 \pm 0.0013}$
scene	$\mathbf{0.4589 \pm 0.0029}$	0.5051 ± 0.0048	$\underline{0.4687 \pm 0.0023}$
gisette	$\underline{0.4713 \pm 0.0017}$	0.5273 ± 0.0032	$\mathbf{0.4051 \pm 0.0037}$
guillermo	$\mathbf{0.5460 \pm 0.0016}$	0.5924 ± 0.0031	$\underline{0.5566 \pm 0.0025}$

Tabela 9 – Acurácia (% ± desvio padrão) obtida pelos diferentes *autoencoders* ($k = 10$) em cada conjunto de dados.

Dataset	AE denso	AE-LC + ord. aleatória	AE-LC + ord. proposta
isolet	$\mathbf{87.60 \pm 0.52}$	87.25 ± 0.75	$\mathbf{87.85 \pm 0.88}$
christine	$\mathbf{71.57 \pm 0.59}$	$\mathbf{71.61 \pm 0.43}$	$\mathbf{71.76 \pm 0.61}$
gina	$\mathbf{81.89 \pm 1.96}$	78.99 ± 3.31	$\mathbf{81.72 \pm 2.33}$
dilbert	$\underline{87.37 \pm 0.81}$	$\underline{87.75 \pm 0.64}$	$\mathbf{90.73 \pm 0.67}$
scene	$\underline{83.84 \pm 1.02}$	$\underline{83.18 \pm 1.44}$	$\mathbf{84.49 \pm 0.89}$
gisette	$\mathbf{97.42 \pm 0.21}$	96.98 ± 0.26	$\mathbf{97.28 \pm 0.32}$
guillermo	$\mathbf{44.41 \pm 0.42}$	$\mathbf{44.42 \pm 0.37}$	43.90 ± 0.36

Tabela 10 – Erro quadrático médio (± desvio padrão) de reconstrução obtidos pelos diferentes *autoencoders* ($k = 20$) em cada conjunto de dados.

Dataset	AE denso	AE-LC + ord. aleatória	AE-LC + ord. proposta
isolet	$\underline{0.2988 \pm 0.0022}$	0.3654 ± 0.0051	$\mathbf{0.2934 \pm 0.0022}$
christine	$\underline{0.6151 \pm 0.0008}$	0.6451 ± 0.0075	$\mathbf{0.6114 \pm 0.0028}$
gina	$\underline{0.5240 \pm 0.0037}$	0.6693 ± 0.0043	$\mathbf{0.4594 \pm 0.0064}$
dilbert	$\underline{0.3051 \pm 0.0029}$	0.3225 ± 0.0062	$\mathbf{0.2586 \pm 0.0013}$
scene	$\mathbf{0.3669 \pm 0.0020}$	0.4498 ± 0.0063	$\underline{0.3798 \pm 0.0042}$
gisette	$\underline{0.4305 \pm 0.0026}$	0.5090 ± 0.0051	$\mathbf{0.3488 \pm 0.0049}$
guillermo	$\mathbf{0.5141 \pm 0.0014}$	0.5701 ± 0.0029	$\underline{0.5303 \pm 0.0032}$

Nas tabelas acima, para cada um dos experimentos, os melhores resultados estão destacados em negrito, enquanto o segundo melhor resultado de cada experimento está sublinhado.

Para todas as comparações entre modelos, foram realizados testes dos postos sinalizados de Wilcoxon [137], com nível de significância de 5% ($\alpha = 0.05$). Esse teste de hipótese é usado para comparar duas amostras emparelhadas, testando a hipótese nula de que a diferença entre as duas variáveis possui mediana 0. Logo, um modelo só é considerado estatisticamente superior ao outro, com respeito a uma determinada métrica, se

Tabela 11 – Acurácia (% ± desvio padrão) obtida pelos diferentes *autoencoders* ($k = 20$) em cada conjunto de dados.

Dataset	AE denso	AE-LC + ord. aleatória	AE-LC + ord. proposta
isolet	$\mathbf{92.67 \pm 0.50}$	91.44 ± 0.66	$\mathbf{92.53 \pm 0.67}$
christine	$\mathbf{73.29 \pm 0.64}$	72.08 ± 0.82	$\mathbf{73.20 \pm 0.62}$
gina	87.69 ± 1.05	83.15 ± 2.14	$\mathbf{88.69 \pm 0.99}$
dilbert	92.75 ± 0.33	92.34 ± 0.88	$\mathbf{95.75 \pm 0.41}$
scene	$\mathbf{85.56 \pm 0.99}$	84.07 ± 1.34	$\mathbf{85.38 \pm 1.05}$
gisette	97.73 ± 0.31	97.38 ± 0.31	$\mathbf{98.32 \pm 0.18}$
guillermo	$\mathbf{41.64 \pm 0.36}$	$\mathbf{41.86 \pm 0.52}$	41.11 ± 0.53

Tabela 12 – Erro quadrático médio (± desvio padrão) de reconstrução para os diferentes *autoencoders* ($k = 30$) em cada conjunto de dados.

Dataset	AE denso	AE-LC + ord. aleatória	AE-LC + ord. proposta
isolet	0.2631 ± 0.0027	0.3515 ± 0.0042	$\mathbf{0.2559 \pm 0.0026}$
christine	$\mathbf{0.5969 \pm 0.0007}$	0.6433 ± 0.0063	0.6064 ± 0.0052
gina	0.4916 ± 0.0042	0.6489 ± 0.0076	$\mathbf{0.4202 \pm 0.0063}$
dilbert	0.2842 ± 0.0009	0.3257 ± 0.0063	$\mathbf{0.2510 \pm 0.0042}$
scene	$\mathbf{0.3186 \pm 0.0025}$	0.4348 ± 0.0080	0.3385 ± 0.0038
gisette	0.4171 ± 0.0031	0.5024 ± 0.0055	$\mathbf{0.3241 \pm 0.0039}$
guillermo	$\mathbf{0.4950 \pm 0.0008}$	0.5633 ± 0.0044	0.5226 ± 0.0041

Tabela 13 – Acurácia (% ± desvio padrão) obtida pelos diferentes *autoencoders* ($k = 30$) em cada conjunto de dados.

Dataset	AE denso	AE-LC + ord. aleatória	AE-LC + ord. proposta
isolet	$\mathbf{94.19 \pm 0.24}$	92.74 ± 0.53	93.91 ± 0.40
christine	$\mathbf{74.22 \pm 0.62}$	72.38 ± 0.34	73.67 ± 0.78
gina	$\mathbf{89.55 \pm 0.71}$	85.14 ± 1.90	$\mathbf{90.13 \pm 0.92}$
dilbert	94.91 ± 0.29	92.66 ± 1.11	$\mathbf{96.50 \pm 0.40}$
scene	$\mathbf{86.29 \pm 0.67}$	83.91 ± 1.37	85.03 ± 1.23
gisette	98.06 ± 0.21	97.60 ± 0.36	$\mathbf{98.39 \pm 0.24}$
guillermo	39.72 ± 0.48	$\mathbf{40.75 \pm 0.36}$	39.94 ± 0.56

o valor médio do seu desempenho for superior, e se a hipótese nula for rejeitada. Por esse motivo, nas tabelas de acurácias, é possível ver que em diversos experimentos, múltiplos modelos são considerados equivalentes, apesar dos valores médios serem diferentes.

Na tabela 14, são mostrados os modelos aplicados em cada um dos *datasets*. Para isso, utiliza-se a seguinte notação: FC_h representa uma camada densa com h unidades de saída, enquanto LC_w^j representa uma camada localmente conectada com janela deslizante de tamanho w e j unidades de saída por janela. Nas camadas localmente conectadas, o parâmetro s, referente ao *stride*, foi omitido, pois este sempre foi escolhido de modo que não houvesse sobreposição entre janelas consecutivas (i.e., $s = w$). Além disso, funções de ativação do tipo ReLU foram utilizadas em cada um das camadas (densas ou localmente conectadas), com exceção da última camada do codificador e última camada do

decodificador, nas quais funções de ativação não foram utilizadas.

Tabela 14 – Arquitetura dos modelos de autocodificação utilizados para cada uma das bases de dados. Nessa tabela, apenas a arquitetura dos codificadores estão exibidas, pois os decodificadores foram construídos de maneira simétrica.

Dataset	Arquitetura densa	Arquitetura localmente conectada
isolet	FC_{70}-FC_{k}	LC^{30}_{15}-LC^{10}_{60}-LC^{50}_{210}-LC^{k}_{50}
christine	FC_{76}-FC_{k}	LC^{40}_{20}-LC^{10}_{80}-LC^{50}_{380}-LC^{k}_{50}
gina	FC_{92}-FC_{k}	LC^{15}_{5}-LC^{15}_{30}-LC^{6}_{30}-LC^{50}_{294}-LC^{k}_{50}
dilbert	FC_{44}-FC_{k}	LC^{20}_{40}-LC^{20}_{40}-LC^{50}_{500}-LC^{k}_{50}
scene	FC_{66}-FC_{k}	LC^{20}_{10}-LC^{10}_{40}-LC^{50}_{150}-LC^{k}_{50}
gisette	FC_{64}-FC_{k}	LC^{20}_{10}-LC^{10}_{40}-LC^{10}_{20}-LC^{50}_{400}-LC^{k}_{50}
guillermo	FC_{28}-FC_{k}	LC^{10}_{20}-LC^{10}_{20}-LC^{50}_{500}-LC^{k}_{50}

Nessa tabela, é possível notar que todas as arquiteturas densas são compostas de apenas uma camada escondida no codificador. A explicação está no fato de que foi verificado, durante os experimentos, que arquiteturas mais profundas não resultaram em ganhos de desempenho. Por fim, destaca-se que, nas últimas camadas das arquiteturas localmente conectadas, o tamanho da janela compreende todas as unidades de entrada, fazendo com que estas funcionem, essencialmente, como camadas densas. Isso se assemelha a forma como a redução dimensional é feita usando *autoencoders* convolutivos, nos quais as primeiras camadas são convolutivas, enquanto a projeção no espaço latente é feita através de camadas densas, como pode ser visto em [138].

5.4.4 Disponibilização dos códigos

Os dados e os códigos necessários para reproduzir os experimentos deste capítulo estão disponíveis no seguinte repositório: `https://github.com/gabrielfab0022/Investigating-a-Dependence-Based-Inductive-Bias-for-Tabular-Representation-Learning`.

5.4.5 Discussões

Analisando os resultados apresentados na seção anterior, a observação mais relevante está no fato de que, ao longo de todos os experimentos, — considerando as diferentes combinações entre *datasets* e dimensões latentes — a aplicação de camadas localmente conectadas em arranjos artificiais baseados em dependência estatística produziu resultados estatisticamente superiores àqueles produzidos pelas mesmas arquiteturas quando aplicadas à arranjos aleatórios, com respeito ao erro de reconstrução. Sob o ponto de vista deste trabalho, esses resultados evidenciam a possibilidade de induzir a noção de localidade em dados tabulares, a partir de campos receptivos locais nos quais vizinhanças espaciais codificam relações de alta dependência entre variáveis.

Também com respeito ao erro de reconstrução, as arquiteturas localmente conectadas, quando combinadas com a ordenação proposta, produziu resultados estatisticamente superiores aos de redes neurais densas com capacidade paramétrica equivalentes em 14 dos 21 experimentos. Sob o ponto de vista deste trabalho, essa observação indica que o viés indutivo proposto, que é inspirado nas CNNs e na localização de dependência, pode produzir, em certos conjuntos de dados, ganhos em relação à abordagem convencional para a autocodificação de dados tabulares.

Em resumo, com relação ao erro de reconstrução, a agregação local de variáveis dependentes se mostrou claramente superior a agregação local de variáveis quaisquer, assim como acontece nas CNNs, e se mostrou também superior a agregação global de variáveis em uma quantidade considerável de cenários.

Quanto aos experimentos de classificação, os resultados foram mistos, à medida que a combinação dos *autoencoders* localmente conectados com a ordenação proposta produziu resultados estatisticamente equivalentes àqueles dos *autoencoders* densos em uma parcela considerável dos experimento, enquanto a combinação dos *autoencoders* localmente conectados com uma ordenação qualquer parece ter produzido acurácias inferiores, a menos do *dataset* guillermo.

Uma outra observação notável, ao analisar os resultados, é a de que, em muitas das bases de dados, as representações latentes que resultam nos menores erros de reconstrução não são necessariamente as mesmas em que há maior separabilidade entre classes. Esse fenômeno não chega a ser surpreendente, uma vez que vários *datasets* reais possuem componentes que não contribuem para o problema de classificação, (e.g. ruídos correlacionados, variáveis distratoras) mas que, potencialmente, são igualmente importantes para o propósito de minimização do erro de reconstrução.

Entretanto, para os fins dessa investigação, consideramos que o erro de reconstrução é a métrica mais informativa, uma vez que ela diz respeito ao quão bem um determinado modelo cumpre o objetivo para o qual ele foi ajustado. Nesse sentido, o viés indutivo baseado em dependências estatísticas e vizinhanças artificiais se mostrou bastante promissor, tendo em vista os desempenhos com respeito ao erro de reconstrução. Por outro lado, também é válido ressaltar que, em muitos cenários práticos, a separabilidade no espaço latente é o objetivo final, de modo que uma representação latente é apenas tão boa quanto o desempenho que ela induz em tarefas posteriores, enquanto que a reconstrução assume o papel de uma tarefa auxiliar.

5.5 Conclusões do capítulo

Neste capítulo da dissertação, foi proposta uma investigação acerca de um viés indutivo baseado em medidas de informação mútua aplicado ao aprendizado tabular de representações, via autocodificação. A investigação foi conduzida através de uma sequência de procedimentos que se assemelham àqueles já utilizados no Capítulo 4, no qual a tarefa hipotética de classificar bases de imagens permutadas foi considerada como prova de princípio acerca da possibilidade de interpretar o viés indutivo das CNNs sob a perspectiva da agregação de variáveis dependentes. Para a extensão do princípio para dados tabulares proposta neste capítulo, as principais diferenças estão na tarefa considerada, que foi a de autocodificação, em vez da classificação de padrões, no fato de que a reconstrução da topologia 2D das imagens foi substituída por uma mera ordenação de *features*, e de que o compartilhamento de pesos das CNNs deu lugar aos filtros não estacionários das camadas localmente conectadas.

Através dos experimentos, foram coletadas evidências de que a imposição da agregação entre variáveis dependentes também podem constituir um viés indutivo relevante para dados tabulares. Em particular, a aplicação de camadas localmente conectadas a variáveis ordenadas de acordo com medidas de informação mútua levou a erros de reconstrução consistentemente menores que os das mesmas arquiteturas aplicadas a variáveis ordenadas aleatoriamente. Além disso, na comparação com arquiteturas densas, apesar dos resultados serem mistos, eles também parecem indicar que a implementação desse viés indutivo pode levar a ganhos de desempenho em relação a abordagem convencional.

Logo, a principal conclusão deste capítulo está no fato de que, usando medidas dependências entre pares de variáveis, é possível induzir noções de vizinhança e de campos receptivos locais para dados tabulares, e a exploração dessas estruturas locais por meio de arquiteturas localmente conectadas pode ser uma alternativa promissora para o avanço do aprendizado profundo no domínio dos dados tabulares.

É relevante destacar, no entanto, que durante a comparação entre modelos, os resultados foram analisados com ênfase em testes estatísticos. Logo, os ganhos de desempenho de um determinado modelo em relação aos outros foram considerados com respeito às suas relevâncias estatísticas, de modo que nada tenha sido dito sobre a relevância desses ganhos para aplicações práticas. Dessa forma, futuramente, uma investigação natural seria a avaliação das arquiteturas localmente conectadas em aplicações envolvendo dados tabulares nas quais *autoencoders* são tipicamente usados (e.g., filtragem de ruído, detecção de anomalias, estimação de dados faltantes etc [139, 140]).

Além disso, a abordagem proposta teve como objetivo preservar a simplicidade e isolar as hipóteses testadas. Nesse sentido, é de se considerar que, sob um ponto de vista de ganhos de desempenho, os resultados obtidos deixam bastante espaço para me-

lhorias posteriores. Por exemplo, a escolha do arranjo unidimensional para as variáveis pode ser revisitada, e estimativas de dimensão intrínseca (semelhante ao que foi feito no Capítulo 4) podem ser usadas para auxiliar na determinação de uma estrutura espacial ideal para acomodar as relações de dependência existentes em um determinado conjunto de dados. Adicionalmente, melhorias também podem ser feitas na implementação das camadas localmente conectadas. A estrutura aqui utilizada é baseada em campos receptivos de tamanhos fixos (assim como nas CNNs). Porém, uma abordagem mais flexível pode ser adotada, na qual os campos receptivos se adaptam naturalmente aos grupos de variáveis dependentes existentes numa base de dados (que podem ser obtidos usando *clustering* de variáveis, por exemplo [141]), com tamanhos potencialmente diferentes.

Outro comentário relevante está associado ao uso de medidas de informação mútua para quantificar a dependência estatística entre pares de variáveis. No início deste trabalho de mestrado, essa escolha foi motivada pela perspectiva de identificar qualquer tipo de dependência, incluindo as dependências não lineares para as quais o coeficiente de correlação pode até ser nulo. Entretanto, os experimentos desse capítulo também foram repetidos aplicando as arquiteturas localmente conectadas em arranjos de variáveis baseados em medidas de correlação. Os resultados obtidos não foram reportados, para se manter fiel ao escopo do trabalho. Não obstante, mencionamos que os resultados obtidos através de medidas de correlação se mostraram equivalentes, e em alguns casos até superiores, àqueles obtidos através de medidas de informação mútua.

Esses resultados são intrigantes, uma vez que eles indicam que, para as bases de dados aqui utilizadas, a maior generalidade da medida de dependência não foi convertida em melhores desempenhos. Embora essa observação não possa ser tratada como um resultado genérico, que se sustentaria em qualquer cenário, ela aponta para mais um foco importante em estudos subsequentes. Especificamente, a maior generalidade da medida de informação mútua é acompanhada de uma grande dificuldade no problema de estimação, tanto com respeito ao custo computacional quanto a questões de robustez. As medidas de correlação, por sua vez, costumam ser confiavelmente estimadas empiricamente, mesmo diante de quantidades moderadas de dados. Dessa forma, é importante caracterizar em quais situações de interesse prático o uso de medidas de informação mútua se justifica, uma vez que para os conjuntos de dados usados neste capítulo o coeficiente de correlação mostrou suficiente (e em alguns casos, até melhor).

6 Considerações finais

Neste trabalho, foi proposta uma investigação acerca das implicações práticas de se impor a agregação de variáveis dependentes nas redes neurais, através de campos receptivos locais artificiais ocupados por pares de variáveis que compartilham alta dependência estatística. Para tal, uma etapa imprescindível em todos os métodos utilizados foi a construção de uma matriz de dependências, contendo medidas de informação mútua entre pares de variáveis aleatórias.

Tendo isso em mente, o primeiro passo deste trabalho foi um estudo aprofundado sobre o problema de estimação de informação mútua para fontes discretas e contínuas, que culminou na escolha do estimador KSG como a ferramenta para a construção de tais matrizes. Durante o percurso, no entanto, sugiram questões relativas ao próprio problema de estimação, que motivaram o desenvolvimento de um estimador para fontes discretas, cujos resultados empíricos, apresentados em [22], se mostraram competitivos com aqueles dos melhores estimadores disponíveis na literatura. Assim, uma continuação natural desta etapa seria complementar a avaliação empírica do estimador com uma caracterização teórica do mesmo, desenvolvendo limites teóricos para o viés e a variância de suas estimativas. Esse tipo de análise é fundamental, uma vez que traz luz sobre as capacidades e as limitações de um determinado estimador, contribuindo para a determinação de quais regimes práticos em que ele se apresenta como uma ferramenta adequada.

No que diz respeito a imposição da agregação entre variáveis dependentes nas redes neurais, dois momentos se destacam. No primeiros deles, conforme apresentado no Capítulo 4, foi verificado, através de uma prova de conceito, que as CNNs, quando aplicadas em imagens, são uma instanciação desse princípio. Isso decorre do fato de que, as matrizes de dependências obtidas, em combinação com o compartilhamento de pesos, são suficientes para caracterizar as noções de localidade e vizinhanças em imagens, e portanto para impor o viés indutivo característico das CNNs, mesmo sem o conhecimento prévio da estrutura espacial das imagens.

O segundo momento diz respeito à investigação acerca da exploração de localidades/vizinhanças artificiais baseadas em medidas de informação mútua para dados tabulares, nos quais essas noções não estão presentes de maneira natural. Esse estudo foi conduzido no contexto da tarefa de redução dimensional por meio de *autoencoders*, e os resultados também revelaram o potencial da agregação entre variáveis dependentes como viés indutivo para essa tarefa, tendo inclusive induzido erros de reconstrução menores que *baselines* densos na maioria dos experimentos.

As seções finais dos capítulos 4 e 5 já trazem conclusões e sugestões para trabalhos

futuros específicas de cada um desses momentos do trabalho. Entretanto, sob o ponto de vista geral do trabalho, algumas questões mais gerais também merecem reflexões.

Por exemplo, os procedimentos utilizados em ambos os capítulos seguem a linha de construir uma matriz que sintetize as dependências entre pares de variáveis em um determinado conjunto de dados, e em seguida, representar essa estrutura de dependências através de uma geometria (e.g., a geometria do grid), na qual sinais podem ser definidos, campos receptivos locais podem ser explorados.

Construções como essas abrem portas para um princípio potencialmente mais genérico. Isto é, sempre que houver alguma medida de associação entre variáveis, — não necessariamente voltada à dependência estatística — é possível induzir uma geometria, e esta pode ser explorada para o desenvolvimento de arquiteturas de RNAs. Esse tipo raciocínio pode inspirar o desenvolvimento de uma nova família de métodos, ao longo de diversos domínios de aplicações.

Além disso, durante a introdução, foi mencionado que a escolha por medidas de informação mútua para quantificar o grau de dependência entre variáveis decorre da maior generalidade que ela permite, em comparação às medidas de correlação de Pearson. No entanto, um resultado surpreendente deste trabalho foi o de que em nenhum momento essa maior generalidade se mostrou necessária durante os experimentos.

No Capítulo 4, foi dito que estudos anteriores em [76, 77] já mostravam a reconstrução de imagens permutadas através de medidas de correlação. A maior surpresa, no entanto, decorre dos resultados do Capítulo 5, em que, como mencionado, os experimentos com as medidas de correlação (cujos resultados específicos não foram reportados) se mostraram equivalentes, e em alguns casos melhores, que aqueles obtidos quando as medidas de informação mútua foram utilizadas.

Esses resultados merecem uma reflexão, que pode ser explorada em trabalhos futuros. Em particular, é pertinente entender se isso diz respeito a uma característica das bases de dados utilizadas neste trabalho, ou se remete a um princípio mais geral, de que a forma como a agregação de variáveis é feita nas redes neurais não permite tirar proveito de relações de dependências mais complexas do que aquelas já capturadas pelo coeficiente correlação.

Tal reflexão se mostra particularmente relevante ao considerar o alto custo computacional associado à estimação de informação mútua. Considerando uma base de dados com N observações e d atributos, preencher a matriz de informação mútua (ou de correlação) requer $d(d-1)/2$ estimativas. Usando o estimador KSG, o principal gargalo computacional está associado às buscas por vizinhos mais próximos no espaço conjunto, cuja complexidade é $\mathcal{O}(N^2)$ em uma implementação ingênua, podendo ser reduzida para $\mathcal{O}(N \log N)$ em uma implementação mais eficiente. Assim, considerando os cálculos para

todos os pares de atributos, o preenchimento da matriz de informação mútua apresenta uma complexidade de $\mathcal{O}(d^2 N \log N)$. Em contrapartida, a complexidade para o preenchimento da matriz de correlação é $\mathcal{O}(d^2 N)$, de modo que o preenchimento da matriz de informação mútua seja computacionalmente mais custosa por um fator de $\log N$.

Finalmente, é necessário pontuar uma limitação inerente à forma como o estudo foi conduzido. Os procedimentos baseados na construção e exploração de uma matriz de informação mútua para um determinado conjunto de variáveis, tal qual a matriz de covariância típica do PCA, refletem uma escolha de modelagem, na qual toda a estrutura de dependência entre variáveis é resumida às interações de segunda ordem, enquanto potenciais interações mais complexas — novamente, remetemos ao exemplo do Ou-Exclusivo (Seção 5.1) — são desprezadas.

Essa escolha de modelagem é uma conveniência adotada em diversas aplicações envolvendo dados ou fenômenos físicos, a exemplo do próprio PCA, e que se mostra pertinente, uma vez que, em muitas situações, ela permite que os fenômenos sejam modelados de maneira tratável e não representa prejuízos significativos para a precisão do modelo. Entretanto, ela também não deve ser ignorada enquanto limitação, podendo motivar estudos futuros que expliquem alguns dos resultados obtidos no Capítulo 5, e eventuais extensões dos métodos propostos que levem em consideração interações de ordens mais altas.

Bibliografia

[1] Hans-Dieter Block. “The perceptron: A model for brain functioning. i”. Em: *Reviews of Modern Physics* 34.1 (1962), p. 123.

[2] Chiyuan Zhang et al. “Understanding deep learning requires rethinking generalization”. Em: *arXiv preprint arXiv:1611.03530* (2016).

[3] Simon JD Prince. *Understanding deep learning*. MIT press, 2023.

[4] Pratik Prabhanjan Brahma, Dapeng Wu e Yiyuan She. “Why deep learning works: A manifold disentanglement perspective”. Em: *IEEE transactions on neural networks and learning systems* 27.10 (2015), pp. 1997–2008.

[5] Henry W Lin, Max Tegmark e David Rolnick. “Why does deep and cheap learning work so well?” Em: *Journal of Statistical Physics* 168.6 (2017), pp. 1223–1247.

[6] Daniel A Roberts, Sho Yaida e Boris Hanin. *The principles of deep learning theory*. Vol. 46. Cambridge University Press Cambridge, MA, USA, 2022.

[7] Daniel Soudry et al. “The implicit bias of gradient descent on separable data”. Em: *Journal of Machine Learning Research* 19.70 (2018), pp. 1–57.

[8] Behnam Neyshabur, Ryota Tomioka e Nathan Srebro. “In search of the real inductive bias: On the role of implicit regularization in deep learning”. Em: *arXiv preprint arXiv:1412.6614* (2014).

[9] Feng-Lei Fan et al. “Towards NeuroAI: introducing neuronal diversity into artificial neural networks”. Em: *Med-X* 3.1 (2025), p. 2.

[10] Adriano Baldeschi, Raffaella Margutti e Adam Miller. “Deep Learning: a new definition of artificial neuron with double weight”. Em: *arXiv preprint arXiv:1905.04545* (2019).

[11] Giovanni Pellegrini et al. “Learning aggregation functions”. Em: *arXiv preprint arXiv:2012.08482* (2020).

[12] Fionn Murtagh. “Multilayer perceptrons for classification and regression”. Em: *Neurocomputing* 2.5-6 (1991), pp. 183–197.

[13] Keiron O’shea e Ryan Nash. “An introduction to convolutional neural networks”. Em: *arXiv preprint arXiv:1511.08458* (2015).

[14] Larry R Medsker, Lakhmi Jain et al. “Recurrent neural networks”. Em: *Design and applications* 5.64-67 (2001), p. 2.

[15] Alex Graves. “Long short-term memory”. Em: *Supervised sequence labelling with recurrent neural networks* (2012), pp. 37–45.

[16] Tianyang Lin et al. “A survey of transformers”. Em: *AI open* 3 (2022), pp. 111–132.

[17] Marina Meilă e Hanyu Zhang. “Manifold learning: What, how, and why”. Em: *Annual Review of Statistics and Its Application* 11.1 (2024), pp. 393–417.

[18] Alexandru Telea, Alister Machado e Yu Wang. “Seeing is learning in high dimensions: The synergy between dimensionality reduction and machine learning”. Em: *SN Computer Science* 5.3 (2024), p. 279.

[19] Aapo Hyvärinen, Patrik O Hoyer e Mika Inki. “Topographic independent component analysis”. Em: *Neural computation* 13.7 (2001), pp. 1527–1558.

[20] Claude E Shannon. “A mathematical theory of communication”. Em: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[21] Liam Paninski. “Estimation of entropy and mutual information”. Em: *Neural computation* 15.6 (2003), pp. 1191–1253.

[22] Gabriel FA Bastos e Jugurta Montalvão. “Partitioning the Sample Space for a More Precise Shannon Entropy Estimation”. Em: *2026 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*. IEEE. 2026, pp. 1–6.

[23] Daniel L Ruderman. “The statistics of natural images”. Em: *Network: computation in neural systems* 5.4 (1994), p. 517.

[24] Aryaman Arora, Clara Meister e Ryan Cotterell. “Estimating the entropy of linguistic distributions”. Em: *arXiv preprint arXiv:2204.01469* (2022).

[25] Anqi Mao, Mehryar Mohri e Yutao Zhong. “Cross-entropy loss functions: Theoretical analysis and applications”. Em: *International conference on Machine learning*. pmlr. 2023, pp. 23803–23828.

[26] Yuriy L Orlov e Nina G Orlova. “Bioinformatics tools for the sequence complexity estimates”. Em: *Biophysical reviews* 15.5 (2023), pp. 1367–1378.

[27] Jugurta Montalvão et al. “On the representation of sparse stochastic matrices with state embedding”. Em: *Pattern Recognition Letters* (2025).

[28] Laurens van der Maaten e Geoffrey Hinton. “Visualizing data using t-SNE”. Em: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[29] Krzysztof Ciesielski. “How good is Lebesgue measure?” Em: *The Mathematical Intelligencer* 11.2 (1989), pp. 54–58.

[30] Carl Erik Särndal. “A comparative study of association measures”. Em: *Psychometrika* 39.2 (1974), pp. 165–187.

[31] Harry Joe. “Relative entropy measures of multivariate dependence”. Em: *Journal of the American Statistical Association* 84.405 (1989), pp. 157–164.

[32] Sergio Verdú. "α-mutual information". Em: *2015 Information Theory and Applications Workshop (ITA)*. IEEE. 2015, pp. 1–6.

[33] Artur Luczak. "Entropy of neuronal spike patterns". Em: *Entropy* 26.11 (2024), p. 967.

[34] George Miller. "Note on the bias of information estimates". Em: *Information theory in psychology: Problems and methods* (1955).

[35] Richard O Duda, Peter E Hart et al. *Pattern classification*. John Wiley & Sons, 2006.

[36] Irving J Good. "The population frequencies of species and the estimation of population parameters". Em: *Biometrika* 40.3-4 (1953), pp. 237–264.

[37] Seongmin Lee e Marcel Böhme. "How much is unseen depends chiefly on information about the seen". Em: *arXiv preprint arXiv:2402.05835* (2024).

[38] Irving J Good e George H Toulmin. "The number of new species, and the increase in population coverage, when a sample is increased". Em: *Biometrika* 43.1-2 (1956), pp. 45–63.

[39] Alon Orlitsky, Ananda Theertha Suresh e Yihong Wu. "Optimal prediction of the number of unseen species". Em: *Proceedings of the National Academy of Sciences* 113.47 (2016), pp. 13283–13288.

[40] Yi Hao e Ping Li. "Optimal prediction of the number of unseen species with multiplicity". Em: *Advances in Neural Information Processing Systems* 33 (2020), pp. 8553–8564.

[41] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[42] Anne Chao e Tsung-Jen Shen. "Nonparametric estimation of Shannon's index of diversity when there are unseen species in sample". Em: *Environmental and ecological statistics* 10.4 (2003), pp. 429–443.

[43] Gregory Valiant e Paul Valiant. "Estimating the unseen: improved estimators for entropy and other properties". Em: *Journal of the ACM (JACM)* 64.6 (2017), pp. 1–41.

[44] Yu G Dmitriev e Felix P Tarasenko. "On the estimation of functionals of the probability density and its derivatives". Em: *Theory of Probability & Its Applications* 18.3 (1974), pp. 628–633.

[45] Christopher M Bishop e Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[46] Ibrahim Ahmad e Pi-Erh Lin. "A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.)" Em: *IEEE Transactions on Information Theory* 22.3 (1976), pp. 372–375.

[47] Hadi Alizadeh Noughabi. "A new estimator of entropy and its application in testing normality". Em: *Journal of Statistical Computation and Simulation* 80.10 (2010), pp. 1151–1162.

[48] Havva Alizadeh Noughabi e Reza Alizadeh Noughabi. "On the entropy estimators". Em: *Journal of Statistical Computation and Simulation* 83.4 (2013), pp. 784–792.

[49] Leonenko Kozachenko. "Sample estimate of the entropy of a random vector". Em: *Probl. Pered. Inform.* 23 (1987), p. 9.

[50] Thomas Benjamin Berrett. "Modern k-nearest neighbour methods in entropy estimation, independence testing and classification". Tese de dout. 2017.

[51] Alexander Kraskov, Harald Stögbauer e Peter Grassberger. "Estimating mutual information". Em: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 69.6 (2004), p. 066138.

[52] Mbanefo S Madukaife e Ho Dang Phuc. "Estimation of Shannon differential entropy: An extensive comparative review". Em: *arXiv preprint arXiv:2406.19432* (2024).

[53] Daniel Jurafsky e James H Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*.

[54] Zellig S Harris. "Distributional structure". Em: *Word* 10.2-3 (1954), pp. 146–162.

[55] Jeffrey Pennington, Richard Socher e Christopher D Manning. "Glove: Global vectors for word representation". Em: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[56] Alexey Dosovitskiy. "An image is worth 16x16 words: Transformers for image recognition at scale". Em: *arXiv preprint arXiv:2010.11929* (2020).

[57] Weihao Gao, Sewoong Oh e Pramod Viswanath. "Demystifying fixed k-nearest neighbor information estimators". Em: *IEEE Transactions on Information Theory* 64.8 (2018), pp. 5629–5661.

[58] Alex Krizhevsky, Geoffrey Hinton et al. *Learning multiple layers of features from tiny images.(2009)*. 2009.

[59] Y. Lecun et al. "Gradient-based learning applied to document recognition". Em: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[60] D. H. Hubel AD T. N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". Em: *The Journal of physiology* (1962).

[61] Kunihiko Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. Em: *Biological cybernetics* 36.4 (1980), pp. 193–202.

[62] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. Em: *Neural computation* 1.4 (1989), pp. 541–551.

[63] Léon Bottou et al. “Comparison of classifier methods: a case study in handwritten digit recognition”. Em: *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5)*. Vol. 2. IEEE. 1994, pp. 77–82.

[64] Yann LeCun et al. “Comparison of learning algorithms for handwritten digit recognition”. Em: *International conference on artificial neural networks*. Vol. 60. 1. Perth, Australia. 1995, pp. 53–60.

[65] Cong Geng e Xudong Jiang. “Face recognition using sift features”. Em: *2009 16th IEEE international conference on image processing (ICIP)*. IEEE. 2009, pp. 3313–3316.

[66] Alex Krizhevsky, Ilya Sutskever e Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. Em: *Advances in neural information processing systems* 25 (2012).

[67] Olaf Ronneberger, Philipp Fischer e Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. Em: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[68] Sora Kim, Sungho Suh e Minsik Lee. “Rad: Region-aware diffusion models for image inpainting”. Em: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 2439–2448.

[69] Fnu Neha et al. “From classical techniques to convolution-based models: A review of object detection algorithms”. Em: *2025 IEEE 6th International Conference on Image Processing, Applications and Systems (IPAS)*. IEEE. 2025, pp. 1–6.

[70] Ian Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[71] Geoffrey E. Hinton et al. “Improving neural networks by preventing co-adaptation of feature detectors”. Em: *CoRR* abs/1207.0580 (2012). arXiv: 1207.0580. URL: http://arxiv.org/abs/1207.0580.

[72] Sergey Ioffe e Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. Em: *International conference on machine learning*. pmlr. 2015, pp. 448–456.

[73] Aston Zhang et al. *Dive into Deep Learning*. https://D2L.ai. Cambridge University Press, 2023.

[74] Matthew D. Zeiler e Rob Fergus. "Visualizing and Understanding Convolutional Networks". Em: *Computer Vision – ECCV 2014*. Ed. por David Fleet et al. Cham: Springer International Publishing, 2014, pp. 818–833. ISBN: 978-3-319-10590-1.

[75] Cristian Ivan. "Convolutional Neural Networks on Randomized Data." Em: *CVPR Workshops*. 2019, pp. 1–8.

[76] Nicolas Roux et al. "Learning the 2-D topology of images". Em: *Advances in Neural Information Processing Systems* 20 (2007).

[77] Mahajabin Rahman e Ilya Nemenman. "Inferring local structure from pairwise correlations". Em: *Physical Review E* 108.3 (2023), p. 034410.

[78] Samer Abdallah e Mark Plumbley. *Geometric dependency analysis*. Rel. técn. Tech. rep, 2006.

[79] Fisher Yu e Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions". Em: *arXiv preprint arXiv:1511.07122* (2015).

[80] Ashish Vaswani et al. "Attention is all you need". Em: *Advances in neural information processing systems* 30 (2017).

[81] Ilya O Tolstikhin et al. "Mlp-mixer: An all-mlp architecture for vision". Em: *Advances in neural information processing systems* 34 (2021), pp. 24261–24272.

[82] Ingwer Borg e Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 2005.

[83] John W Sammon. "A nonlinear mapping for data structure analysis". Em: *IEEE Transactions on computers* 100.5 (1969), pp. 401–409.

[84] Joseph B Kruskal. "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis". Em: *Psychometrika* 29.1 (1964), pp. 1–27.

[85] Jarkko Venna e Samuel Kaski. "Local multidimensional scaling". Em: *Neural Networks* 19.6-7 (2006), pp. 889–899.

[86] Sam T Roweis e Lawrence K Saul. "Nonlinear dimensionality reduction by locally linear embedding". Em: *science* 290.5500 (2000), pp. 2323–2326.

[87] Harold W Kuhn. "The Hungarian method for the assignment problem". Em: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[88] Francesco Camastra e Alessandro Vinciarelli. "Estimating the intrinsic dimension of data with a fractal-based method". Em: *IEEE Transactions on pattern analysis and machine intelligence* 24.10 (2002), pp. 1404–1407.

[89] Zaher Joukhadar et al. "A Bayesian framework for robust local intrinsic dimensionality estimation". Em: *Information Systems* (2026), p. 102668.

[90] Elena Facco et al. “Estimating the intrinsic dimension of datasets by a minimal neighborhood information”. Em: *Scientific reports* 7.1 (2017), p. 12140.

[91] Francesco Camastra e Antonino Staiano. “Intrinsic dimension estimation: Advances and open problems”. Em: *Information Sciences* 328 (2016), pp. 26–41.

[92] Peter Grassberger e Itamar Procaccia. “Measuring the strangeness of strange attractors”. Em: *Physica D: nonlinear phenomena* 9.1-2 (1983), pp. 189–208.

[93] Jugurta Montalvão, Jânio Canuto e Luiz Miranda. “Bias-compensated estimator for intrinsic dimension and differential entropy: a visual multiscale approach”. Em: *Journal of Communication and Information Systems* 35.1 (2020), pp. 300–310.

[94] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. Em: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[95] Patrick Helber et al. “Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification”. Em: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 12.7 (2019), pp. 2217–2226.

[96] Karen Simonyan e Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. Em: *arXiv preprint arXiv:1409.1556* (2014).

[97] Gladys M Hilasaca et al. “A grid-based method for removing overlaps of dimensionality reduction scatterplot layouts”. Em: *IEEE Transactions on Visualization and Computer Graphics* 30.8 (2023), pp. 5733–5749.

[98] Ohad Fried et al. “IsoMatch: Creating informative grid layouts”. Em: *Computer graphics forum*. Vol. 34. 2. Wiley Online Library. 2015, pp. 155–166.

[99] Tianqi Chen e Carlos Guestrin. “Xgboost: A scalable tree boosting system”. Em: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[100] Liudmila Prokhorenkova et al. “CatBoost: unbiased boosting with categorical features”. Em: *Advances in neural information processing systems* 31 (2018).

[101] Léo Grinsztajn, Edouard Oyallon e Gaël Varoquaux. “Why do tree-based models still outperform deep learning on typical tabular data?” Em: *Advances in neural information processing systems* 35 (2022), pp. 507–520.

[102] Ravid Shwartz-Ziv e Amitai Armon. “Tabular data: Deep learning is not all you need”. Em: *Information fusion* 81 (2022), pp. 84–90.

[103] Xin Huang et al. “Tabtransformer: Tabular data modeling using contextual embeddings”. Em: *arXiv preprint arXiv:2012.06678* (2020).

[104] Sercan Ö Arik e Tomas Pfister. "Tabnet: Attentive interpretable tabular learning". Em: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 8. 2021, pp. 6679–6687.

[105] Jintai Chen et al. "Danets: Deep abstract networks for tabular data classification and regression". Em: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 4. 2022, pp. 3930–3938.

[106] Noah Hollmann et al. "Accurate predictions on small data with a tabular foundation model". Em: *Nature* 637.8045 (2025), pp. 319–326.

[107] Firas Laakom et al. "Reducing redundancy in the bottleneck representation of autoencoders". Em: *Pattern Recognition Letters* 178 (2024), pp. 202–208.

[108] Michael M Bronstein et al. "Geometric deep learning: going beyond euclidean data". Em: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[109] Tinglin Huang et al. "Mixgcf: An improved training method for graph neural network-based recommender systems". Em: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2021, pp. 665–674.

[110] Weiyang Kong, Ziyu Guo e Yubao Liu. "Spatio-temporal pivotal graph neural networks for traffic flow forecasting". Em: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 38. 8. 2024, pp. 8627–8635.

[111] Antonis Vasileiou et al. "Position: Message-passing and spectral GNNs are two sides of the same coin". Em: (2026).

[112] Thomas N Kipf e Max Welling. "Semi-supervised classification with graph convolutional networks". Em: *arXiv preprint arXiv:1609.02907* (2016).

[113] Justin Gilmer et al. "Neural message passing for quantum chemistry". Em: *International conference on machine learning*. Pmlr. 2017, pp. 1263–1272.

[114] Joan Bruna et al. "Spectral networks and locally connected networks on graphs". Em: *arXiv preprint arXiv:1312.6203* (2013).

[115] Michaël Defferrard, Xavier Bresson e Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering". Em: *Advances in neural information processing systems* 29 (2016).

[116] Mikael Henaff, Joan Bruna e Yann LeCun. "Deep convolutional networks on graph-structured data". Em: *arXiv preprint arXiv:1506.05163* (2015).

[117] Antonio Ortega et al. "Graph signal processing: Overview, challenges, and applications". Em: *Proceedings of the IEEE* 106.5 (2018), pp. 808–828.

[118] Clare D McGillem e George R Cooper. "Continuous and discrete signal and system analysis". Em: *(No Title)* (1991).

[119] Fan RK Chung. *Spectral graph theory*. Vol. 92. American Mathematical Soc., 1997.

[120] Ulrike Von Luxburg. "A tutorial on spectral clustering". Em: *Statistics and computing* 17.4 (2007), pp. 395–416.

[121] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[122] Yin Zhang, Rong Jin e Zhi-Hua Zhou. "Understanding bag-of-words model: a statistical framework". Em: *International journal of machine learning and cybernetics* 1.1 (2010), pp. 43–52.

[123] Yotam Hechtlinger, Purvasha Chakravarti e Jining Qin. "A generalization of convolutional neural networks to graph-structured data". Em: *arXiv preprint arXiv:1704.08165* (2017).

[124] Mikhail Belkin e Partha Niyogi. "Laplacian eigenmaps for dimensionality reduction and data representation". Em: *Neural computation* 15.6 (2003), pp. 1373–1396.

[125] Michael B Eisen et al. "Cluster analysis and display of genome-wide expression patterns". Em: *Proceedings of the National Academy of Sciences* 95.25 (1998), pp. 14863–14868.

[126] Ziv Bar-Joseph, David K Gifford e Tommi S Jaakkola. "Fast optimal leaf ordering for hierarchical clustering". Em: *Bioinformatics* 17.suppl_1 (2001), S22–S29.

[127] Yaniv Taigman et al. "Deepface: Closing the gap to human-level performance in face verification". Em: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1701–1708.

[128] Ron Cole e Mark Fanty. *ISOLET*. UCI Machine Learning Repository. DOI: https://doi.org/10.
1991.

[129] ChaLearn. *AutoML Challenge Dataset*. Dataset do OpenML ID 41142. URL: `https://www.openml.org/d/41142` (acesso em 30/06/2026).

[130] Isabelle Guyon et al. "Design of the 2015 chalearn automl challenge". Em: *2015 International joint conference on neural networks (IJCNN)*. IEEE. 2015, pp. 1–8.

[131] OpenML. *OpenML Dataset 41158*. Dataset do OpenML ID 41158. URL: `https://www.openml.org/d/41158` (acesso em 30/06/2026).

[132] ChaLearn. *AutoML Challenge Dataset*. Dataset do OpenML ID 41163. URL: `https://www.openml.org/d/41163` (acesso em 30/06/2026).

[133] OpenML. *OpenML Dataset 312*. Dataset do OpenML ID 312. URL: `https://www.openml.org/d/312` (acesso em 30/06/2026).

[134] Isabelle Guyon et al. "Result analysis of the nips 2003 feature selection challenge". Em: *Advances in neural information processing systems* 17 (2004).

[135] ChaLearn. *AutoML Challenge Dataset 41159*. Dataset do OpenML ID 41159. URL: `https://www.openml.org/d/41159` (acesso em 30/06/2026).

[136] Diederik P Kingma e Jimmy Ba. "Adam: A method for stochastic optimization". Em: *arXiv preprint arXiv:1412.6980* (2014).

[137] Frank Wilcoxon. "Individual comparisons by ranking methods". Em: *Breakthroughs in statistics: Methodology and distribution*. Springer, 1992, pp. 196–202.

[138] Xifeng Guo et al. "Deep clustering with convolutional autoencoders". Em: *International conference on neural information processing*. Springer. 2017, pp. 373–382.

[139] Timur Sattarov, Marco Schreyer e Damian Borth. "Diffusion-Scheduled Denoising Autoencoders for Anomaly Detection in Tabular Data". Em: *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 2025, pp. 2478–2489.

[140] Tianyu Du, Luca Melis e Ting Wang. "Remasker: Imputing tabular data with masked autoencoding". Em: *International Conference on Learning Representations*. Vol. 2024. 2024, pp. 9002–9024.

[141] Evelyne Vigneau e El Mostafa Qannari. "Clustering of variables around latent components". Em: *Communications in Statistics-Simulation and Computation* 32.4 (2003), pp. 1131–1150.