Universidade Federal da Paraíba
Centro de Ciências Exatas e da Natureza
Coordenação dos Cursos de Graduação em Física

Trabalho de Conclusão de Curso

Método de aplicação de Redes Neurais Convolucionárias (CNN) para identificação de fusões de galáxias em dados simulados pelo Projeto *Illustris*

Natali Cristina Moreira de Almeida

João Pessoa - PB
25/10/2024

Natali Cristina Moreira de Almeida

Método de aplicação de Redes Neurais Convolucionárias (CNN) para identificação de fusões de galáxias em dados simulados pelo Projeto *Illustris*

Trabalho de Conclusão do Curso de Graduação em Física do Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba como requisito parcial para a obtenção do título de Bacharel em Física.

Orientador: Prof. Dr. Hugo Leonardo Davi De Souza Cavalcante

João Pessoa - PB

25/10/2024

Universidade Federal da Paraíba
Centro de Ciências Exatas e da Natureza
Coordenação dos Cursos de Graduação em Física

Ata da Sessão Pública da Defesa do Trabalho de Conclusão de Curso de Bacharelado em Física, da discente Natali Cristina Moreira de Almeida.

Aos 25 dias do mês de outubro do ano de 2024, às 10h, na sala 201 do Departamento de Física, CCEN-UFPB, realizou-se a Sessão Pública da Defesa do Trabalho de Conclusão de Curso de Bacharelado em Física, da discente Natali Cristina Moreira de Almeida, sendo a Banca Examinadora constituída pelos docentes Prof. Dr. Hugo Leonardo Davi de Souza Cavalcante (UFPB), orientador e presidente da banca, Prof. Drª. Thaís Gaudêncio do Rego (UFPB) e Prof. Dr. Fábio Leal de Melo Dahia (UFPB). Dando início aos trabalhos, o professor orientador e presidente da banca examinadora comunicou aos presentes a finalidade da reunião. A seguir, concedeu a palavra à discente para que fizesse a explanação de seu Trabalho de Conclusão de Curso, intitulado "*Método de aplicação de Convolutional Neural Networks (CNN) para identificação de fusões de galáxias em dados simulados pelo projeto Illustris*". Concluída a exposição, a discente foi arguida pelos membros presentes da Banca Examinadora. Após as arguições, a Banca, de comum acordo, declarou que o Trabalho apresentado foi aprovado com nota 8,3. E para constar, encerrada a sessão, lavrou-se esta ata que será assinada pelos presentes. João Pessoa, 25 de outubro de 2024.

Prof. Dr. Hugo Leonardo Davi de Souza Cavalcante
UFPB – Orientador

Prof. Drª. Thaís Gaudêncio do Rego

Prof. Dr. Fábio Leal de Melo Dahia

Catalogação na publicação
Seção de Catalogação e Classificação

A444m Almeida, Natali Cristina Moreira de.
Método de aplicação de Redes Neurais Convolucionárias (CNN) para identificação de fusões de galáxias em dados simulados pelo Projeto Illustris / Natali Cristina Moreira de Almeida. - João Pessoa, 2024.
51 p. : il.

Orientação: Hugo Leonardo Davi de Souza Cavalcante.
TCC (Curso de Bacharelado em Física) - UFPB/CCEN.

1. Redes Neurais Convolucionais - CNN. 2. Fusão de galáxias. 3. Classificação de imagens. 4. Projeto Illustris. 5. Astrofísica computacional. I. Cavalcante, Hugo Leonardo Davi de Souza. II. Título.

UFPB/CCEN CDU 53(043.2)

Elaborado por Josélia Maria Oliveira da Silva - CRB-15/113

Dedico esse trabalho a minha falecida avó Celma Moreira Rodrigues, a pessoa que acreditou no meu sonho antes de mim mesma.

Agradecimentos

Eu agradeço, primeiramente, a mim mesma por ter conseguido concluir o curso dando o meu melhor, apesar do meu autismo, dos meus gaps de conhecimento, das vezes em que adoeci física e emocionalmente.
Agradeço a minha família que me forneceu suporte emocional e financeiro durante a graduação. Especialmente, minha irmã, Stefani Almeida, por ser meu suporte emocional e fonte das ideias mais absurdas, que eu abracei por serem divertidas.
Agradeço as minhas psicóloga e psiquiatra, Maria da Luz e Maria Edilma, que cuidaram do meu psicológico, contornaram as situações de crise e me ajudaram a superar os limites que eu acreditava ser intransponíveis.
Agradeço ao meu tutor do PET-Física, Charlie Salvador, que me ajudou a crescer como pessoa e como profissional.
Também agradeço a cada um dos meus colegas do PET-Física que me proporcionam momentos de estresse, mas também de muitos risos.
Agradeço ao técnico Bruno César e ao seu filho Dayvison Gomes, que me serviram de suporte técnico para a execução desse projeto ambicioso de minha parte.
Agradeço atual coordenador do curso, Jansen Brasileiro Formiga, pela apoio na parte burocrática desse TCC.

Resumo

Desde sua origem, o universo está em constante evolução, a qual inclui a formação e destruição de buracos negros, galáxias, estrelas e outras estruturas cosmológicas. Observar esta evolução permite compreender, descobrir e testar modelos e teorias relacionadas aos fundamentos da física, como o Big Bang, a relatividade, a existência de matéria e de energia escuras, etc. Além da observação experimental, os modelos teóricos podem ser testados em simulações computacionais, nas quais as variáveis teóricas podem ser controladas diretamente e o tempo acelerado ou invertido, permitindo análises impossíveis de se realizar em escalas reais. Os modelos recentes, entretanto, produzem uma vasta quantidade de dados, o que dificulta sua análise manual. O mesmo acontece com a vasta quantidade de imagens adquiridas pelos Telescópios Espaciais Hubble (HST) e James Webb (JWST). Para facilitar esta análise, demonstramos aqui o uso de Redes Neurais Convolucionais (CNNs) para identificar fusões de galáxias em dados simulados pelo projeto Illustris. Reproduzimos com sucesso alguns resultados da literatura, aperfeiçoando os hiper parâmetros da rede neural para permitir uma convergência mais rápida e mais confiável. A coleta de dados foi realizada em plataformas de acesso livre, GitHub e Instituto de Ciência do Telescópio Espacial (STSI). O código-fonte desenvolvido também está disponibilizado à comunidade, para que possa ser usado neste e outros problemas, e aprimorado. Esperamos assim contribuir com o uso de redes neurais e inteligência artificial não apenas na área de astrofísica, mas em outros problemas que necessitem de classificação de imagens e que disponham de uma grande base de dados.

Palavras-chave: Redes Neurais Convolucionais, Fusão de Galáxias, Projeto Illustris, Classificação de Imagens, Astrofísica Computacional.

Abstract

Since its origin, the universe has been in constant evolution, which includes the formation and destruction of black holes, galaxies, stars, and other cosmological structures. Observing this evolution allows us to understand, discover, and test models and theories related to the foundations of physics, such as the Big Bang, relativity, the existence of dark matter and dark energy, etc. In addition to experimental observation, theoretical models can be tested in computer simulations, in which theoretical variables can be directly controlled and time accelerated or reversed, allowing analyses that are impossible to perform on real scales. Recent models, however, produce a vast amount of data, which makes their manual analysis difficult. The same is true for the vast amount of images acquired by the Hubble Space Telescope (HST) and James Webb Space Telescope (JWST). To facilitate this analysis, we demonstrate here the use of convolutional neural networks (CNNs) to identify galaxy mergers in data simulated by the Illustris project. We successfully reproduced some results from the literature, improving the hyperparameters of the neural network to allow faster and more reliable convergence. Data collection was performed on open-access platforms, GitHub, and the Space Telescope Science Institute (STSI). The source code developed is also made available to the community so that it can be used in this and other problems and improved. We hope to contribute to the use of neural networks and artificial intelligence not only in astrophysics but also in other problems that require image classification and that have a large database.

Keywords: Convolutional Neural Networks, Galaxy Merger, Illustris Project, Image Classification, Computational Astrophysics.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

A Inteligência Artificial (IA) e as Redes Neurais estão se consolidando como ferramentas-chave em diversas áreas da ciência, culminando, em 2024, na premiação do Nobel de Química e Física a pesquisadores pioneiros nessas áreas. David Baker, John Jumper e Demis Hassabis receberam o Nobel de Química por suas contribuições para a previsão da estrutura de proteínas utilizando IA, enquanto John Hopfield e Geoffrey Hinton foram reconhecidos com o Nobel de Física por seus trabalhos em Redes de Hopfield e no desenvolvimento de sistemas de aprendizagem profunda para a identificação de padrões em dados. Esse reconhecimento sublinha a importância e o potencial dessas tecnologias para impulsionar a pesquisa científica.

O advento de telescópios espaciais como o James Webb (JWST), lançado em 2021 e posicionado no segundo ponto de Lagrange (L2), embora represente um avanço monumental para a astronomia, impõe desafios significativos para a análise e interpretação da imensa quantidade de dados complexos gerados diariamente. As observações do JWST sobre o espaço profundo exigem métodos automatizados e eficientes de processamento e classificação, superando as limitações da análise manual, inerentemente lenta, trabalhosa e suscetível a erros e vieses humanos.

Este trabalho foca na análise de imagens de galáxias, especificamente no processo de fusão galáctica, um fenômeno com características morfológicas distintas passíveis de identificação e classificação automatizada. A comparação entre as imagens do JWST, dados do Telescópio Espacial Hubble (HST) e simulações cosmológicas do Projeto Illustris proporciona uma oportunidade singular para o aprimoramento da compreensão das complexidades desse processo.

Embora as Redes Neurais tenham demonstrado grande potencial, sua aplicação na pesquisa científica ainda encontra resistência por parte da comunidade acadêmica, que, por vezes, as considera “caixas pretas” devido à complexidade de seus processos internos e à potencial influência de vieses nos dados de treinamento. No entanto, a utilização criteriosa e supervisionada de Redes Neurais, aliada à validação rigorosa, configura-se como uma ferramenta poderosa para a análise de dados, permitindo validar ou refutar modelos físicos com agilidade e precisão.

O objetivo geral deste trabalho é aplicar Redes Neurais Convolucionais (CNNs) para a classificação de imagens de fusões de galáxias a partir de dados simulados pelo Projeto Illustris, utilizando a linguagem de programação Python. Os objetivos específicos incluem a replicação de resultados da literatura e a otimização dos hiper parâmetros empregados no treinamento das CNNs, visando maximizar a performance do modelo. Os dados foram obtidos de plataformas de acesso aberto, como o GitHub e o repositório do Space Telescope Science Institute (STSI).

A abordagem metodológica adotada é quantitativa, com ênfase na análise comparativa dos resultados. Uma metodologia descritivo-explicativa será utilizada para apresentar os fundamentos

teóricos das CNNs, abrangendo a arquitetura da rede, as funções de ativação e a função de custo. Os resultados do treinamento da CNN serão comparados com as expectativas teóricas para avaliar a eficácia do modelo.

Este trabalho está estruturado da seguinte forma: Capítulo 1 - Introdução; Capítulo 2 – Revisão de literatura e conceitos gerais; Capítulo 3 - Metodologia; Capítulo 4 – Apresentação e análise dos resultados; Capítulo 5 – Conclusões.

2 Revisão de literatura e conceitos gerais

Esse capítulo será dividido em: apresentação do Projeto *Illustris*; apresentação de conceitos básicos de Redes Neurais Artificiais (RNAs); detalhamento de alguns passos do processo de construção das RNAs e a definição e características das CNNs.

2.1 Projeto *Illustris*

O Projeto *Illustris* é um conjunto de simulações cosmológicas de formação de galáxias, em larga escala, que se baseia no cálculo de rastreamento da expansão do universo e na análise da "hidrodinâmica" do gás cósmico, das estrelas e dos buracos negros. Todas essas simulações são baseadas no modelo padrão da cosmologia *Lambda Cold Dark Matter* (ΛCDM), que postula sobre a relação entre densidade de massa-energia do Universo (1).

Essas simulações são baseadas nos dados astronômicos já coletados do pós-*Big Bang*, em que os fótons começaram a ser emitidos, recorte do universo mais jovem a que temos acesso, cerca de 3 bilhões de anos, até cerca de 13,8 bilhões, um universo mais recente, que possui as estruturas organizadas da forma com que conhecemos (1).

Nas imagens abaixo, temos dois exemplos de simulação realizados pelo Projeto *Illustris*. A primeira representa o que os cosmólogos chamam de "Teia Cósmica", uma interação entre matéria e energia bariônica com matéria e energia escura, as quais são responsáveis por modelar as estruturas da forma com que são observadas pelos telescópios espaciais. Já a segunda representa nosso objeto de estudo, as unidades básicas da estrutura cósmica, galáxias, que podem ou não ter forma fixa a depender da existência do processo de fusão (1).

Essas simulações servem para testar teorias cosmológicas, entender a formação de galáxias e explicar as propriedades observadas. Então, por se tratarem de um catálogo de simulações, de parceria internacional e de acesso gratuito, alguns cientistas utilizam essas simulações como um tipo de dados comparativos para os dados obtidos a bordo dos telescópios espaciais HST e JWST, cujos dados servem para corroborar ou descartar hipóteses.

Essa análise comparativa entre os dados simulados e reais também possibilita que o Projeto *Illustris* aprimore suas simulações à medida que novos dados cosmológicos vão sendo obtidos. Associado a isso, há a evolução computacional que permite simulações cada vez mais detalhadas e robustas, que antes eram tecnologicamente impossíveis.

Sendo assim, esse projeto passou por evoluções, com características próprias, listadas abaixo:

- ***Illustris*-1:** A primeira versão do projeto foi focada em criar uma simulação cosmológica

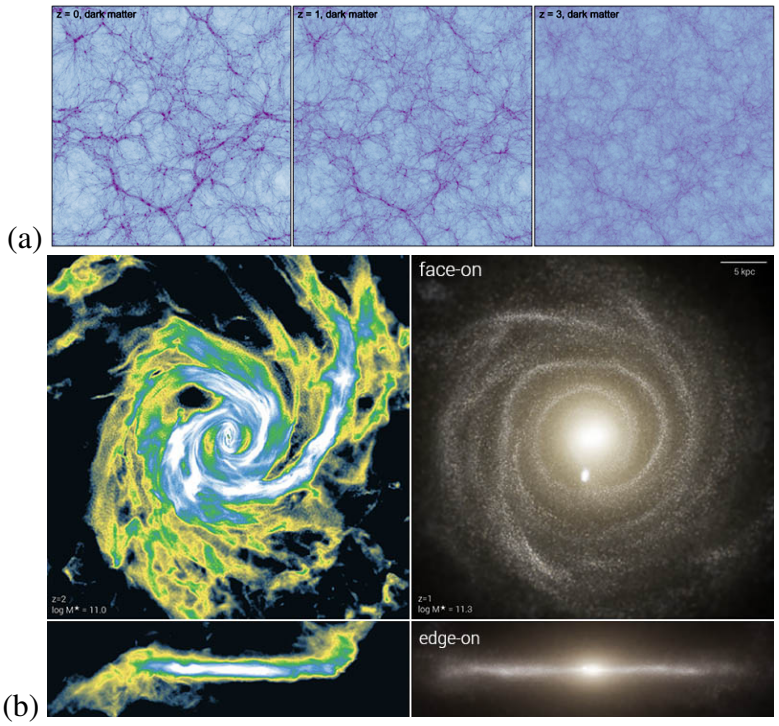


Figura 1 – Simulações respectivas de (a) "Teia Cósmica" e de (b) modelagem de Galáxia (1)

detalhada da formação e da evolução das galáxias. Apesar de sua alta resolução, ainda lidava com limitações nos modelos de física galáctica, pois só incluía processos básicos de formação estelar, de supernovas e do crescimento de buracos negros. Algumas dessas limitações eram devido a simplificações em alguns mecanismos de retroalimentação, o que gerava uma série de deficiências na formação de galáxias de baixa massa e na reprodução de propriedades específicas dessas.

- ***Illustris*-2 ou A próxima geração (TNG):** Essa versão revisou e aprimorou os modelos físicos utilizados, além de gerar uma modelagem mais precisa dos processos astrofísicos. O aprimoramento se deu no aumento da resolução do volume simulado, no aumento da variedade de ambientes galácticos e cosmológicos, da inclusão de campos magnéticos e de melhorias na modelagem da evolução química dos gases cósmicos.
- ***Illustris*-3:** continuou a expandir a escala das simulações e a complexidade dos modelos físicos, abordando questões não resolvidas das versões anteriores. Também passou a explorar novas áreas da cosmologia e astrofísica devido ao estabelecimento de colaborações científicas entre universidades que possuem ou têm parcerias com supercomputadores, pois essas cedem tempo de processamento e espaço de armazenamento para os dados simulados. (1)

Baseado no modelo de simulação cósmica *Illustris*-1, focaremos na análise das assinaturas características das galáxias com ou sem processo de fusão.

2.2 Redes Neurais Artificiais (RNAs)

O primeiro modelo teórico de rede neural artificial foi proposto, em 1943, pelo neurofisiologista Warren McCulloch (1898-1969) e pelo matemático Wallter Pitts (1923–1969), através do artigo "Um Cálculo Lógico das Ideias Imanentes à Atividade Nervosa", no qual dissertavam sobre como seria tentar simular o funcionamento do cérebro humano em uma máquina através de uma rede sofisticada de nós sinápticos chamada de *Psychon* (13).

A capacidade de processamento do cérebro é de grande interesse aos cientistas da computação devido à sua não linearidade, execução paralela de atividades, capacidade de processamento, de aprendizado, de reestruturação e de reconstrução de informações. Podemos tratá-lo como uma Rede Neural Biológica (RNB), cujo processamento da informação dos órgãos sensoriais se dá através da transformação dos estímulos físicos (calor, som, luz, etc.) em sinais eletroquímicos (14).

O cérebro é um órgão composto por células nervosas, chamadas neurônios, os quais recebem informações de outras células nervosas ou de órgãos sensoriais, como na imagem da figura 2 abaixo.

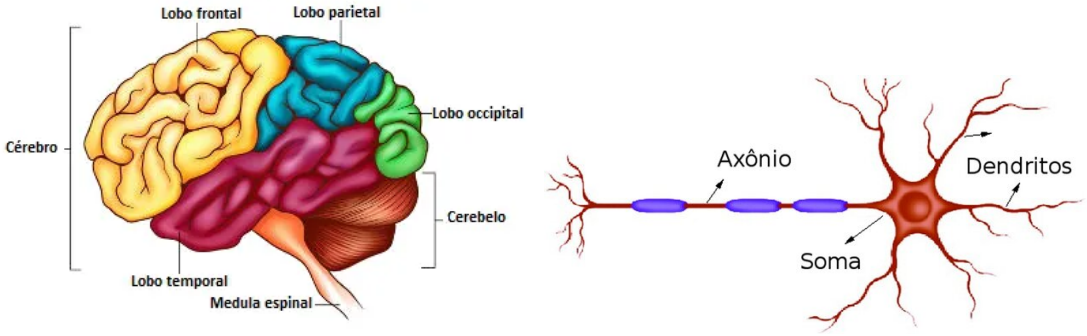


Figura 2 – Rede Neural Biológica (RNB) (2) (3)

As informações, provenientes de um estímulo nervoso, chegam ao neurônio a partir dos axônios, são processadas pelo corpo celular, estrutura em forma de árvore, que representa o ponto de entrada dessas informações. Após esse processamento, essas são propagadas de um neurônio para outro através das sinapses, que podem ser de natureza excitatória ou inibitória (14).

Segundo a neurociência, um recém-nascido possui a quantidade máxima de neurônios disponíveis para o processamento de informações, com uma alta taxa de elasticidade e de adaptabilidade. Isso se deve, pois as zonas de captação, processamento e emissão ainda não são especializadas, ou seja, não há conexões reforçadas por estímulos (14). Toda via, ao longo da vida ocorrem processos de poda neural, fixação de conexões, criação de vieses, etc., que geram uma redução exponencial das probabilidades de caminho para os impulsos nervosos (3).

Tendo em vista essa descrição, podemos dizer que Redes Neurais Artificiais (RNAs) são um

mecanismo de processamento, análise e respostas aos sinais de entrada binários propagados em circuitos eletrônicos. A propagação dessas informações se dá através das suas interconexões, que são capazes de extrair padrões, hierarquizá-los e detectar similaridades.

Um neurônio artificial (NA), representado na figura abaixo, pode ser definido como uma unidade de processamento de informações fundamental para a operação de uma rede neural. O NA é composto por:

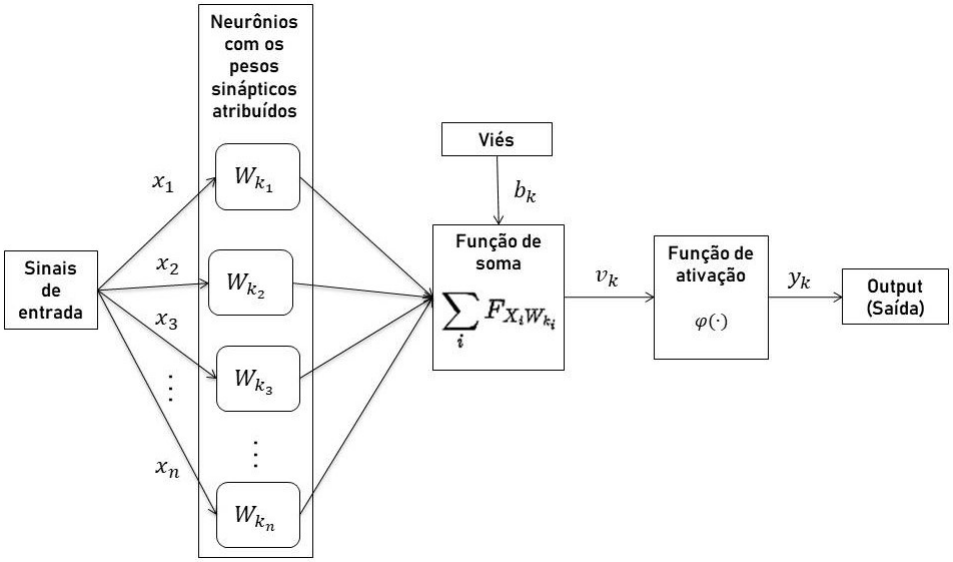


Figura 3 – Modelo de um neurônio artificial (4)

1. **Conjunto de sinapses:** caracterizado por um sinal de entrada do tipo x_j, em que cada sinapse j é conectada a um neurônio k. Quando esse sinal chega ao neurônio, ele é multiplicado pelo peso sináptico w_{x_j}, que pode incluir valores negativos e positivos de uma determinada faixa.
2. **Somador:** realiza soma ponderada dos sinais de entrada pelas respectivas sinapses do neurônio, cujas operações constituem uma combinação linear.
3. **Viés:** é responsável por deslocar o nível de referência para aumentar ou diminuir a sensibilidade da função de ativação.
4. **Função de ativação:** responde à soma dos estímulos, pesos e vieses associados e limita sua amplitude da saída. Isso ocorre, pois comprime a faixa de amplitude permitida do sinal de saída para algum valor finito. (14)

Podemos escrever a forma matemática da função de saída, de forma generalizada, através desse modelo para X_k sinais de entrada, com pesos sinápticos W_k, da seguinte forma: (14)

$$u_k = \sum_{j=1}^{m} W_{kj} X_j, \tag{2.1}$$

Em que u_k representa a combinação linear das saídas. Então, ao aplicarmos o viés b_k na

transformação de saída afim u_k, a combinação linear será: (14)

$$v_k = u_k + b_k, \tag{2.2}$$

Em que b_k serve como nível de referência e pode alterar a relação entre o potencial de ativação v_k e a saída da combinação linear u_k, descrita por:

$$\begin{aligned} y_k &= \varphi(u_k + b_k), \\ y_k &= \varphi(v_k). \end{aligned} \tag{2.3}$$

que gera um sinal de saída y_k. (14)

Uma vez compreendido o que são RNAs e como funciona o processamento de um NA, podemos refletir que a tentativa de se reproduzir uma rede neural, de forma artificial, é reconhecer a capacidade computacional não linear do cérebro humano. Esse é capaz de prever padrões na natureza de forma probabilística e tomar decisões em frações de segundo, enquanto os computadores estão limitados à lógica Booleana, 0 ou 1, em que se pode atribuir um caráter preditivo, baseado na Lógica Bayesiana, mas que para isso são necessárias técnicas mais rebuscadas de análise e classificação de dados, com mais camadas e interconexões.

2.2.1 Construção da RNA

Definir o tipo de dado que será tratado é de suma importância para determinar a escolha de cada atributo que comporá a nossa RNA, uma vez que esse influencia diretamente na escolha da arquitetura da rede, das funções de ativação e da função de perda. Assim, compreender o tipo de dado permite que você escolha os atributos corretos para criar uma rede neural artificial eficaz.

Nosso foco está em dados de imagem baseados nas simulações computacionais do Projeto *Illustris*, então a literatura já nos fornece pistas de que precisamos utilizar a técnica Redes Neurais Convolucionais (CNNs) para processar imagens e extrair características relevantes.

Antes de aplicarmos nossos dados à CNN, focaremos em descrever formas de processamento de dados, as principais características da modelagem, dos hiper parâmetros e dos mecanismos de monitoramento.

2.2.2 Processamento de dados

Uma das formas de se processar os dados de forma mais eficiente é dividi-los em lotes. O tamanho dos lotes caracteriza o número de amostras do conjunto de dados de treinamento que o modelo processa antes de atualizar seus parâmetros. Dessa forma, ao invés de ajustar os parâmetros a cada amostra individual, o modelo calcula a média do erro sobre o lote de amostras e usa essa média para ajustar os parâmetros.

Em aprendizado de máquina, uma época representa uma única passagem por todo o conjunto de dados de treinamento. Durante essa passagem, o modelo processa cada exemplo de dado uma

vez, realizando atualizações nos parâmetros com base nos erros encontrados em cada mini-lote de dados. Nesse trabalho, utilizaremos a técnica de Descida de Gradiente em Lote (BGD), em que o lote recebe um valor intermediário (32 e 256), que permite a atualização dos parâmetros do modelo várias vezes por época.

Um fato que devemos ter em mente é que o tamanho do lote afeta a eficiência e a estabilidade do treinamento. Lotes menores podem oferecer maior variação nas atualizações, enquanto lotes maiores geralmente proporcionam atualizações mais estáveis. Essa estabilidade dos lotes maiores é mais eficiente para processamento em paralelo.

Para algoritmos como redes neurais, o treinamento geralmente ocorre em múltiplas épocas, que permite o ajuste dos parâmetros do modelo de forma gradual, de forma a reduzir o erro das épocas até se atingir um nível de precisão satisfatório. Logo, a sua quantidade ideal varia de acordo com o problema.

Portanto, utilizar a quantia de épocas corretas é importante, pois valores muito altos podem levar ao sobreajuste, em que o modelo se ajusta demais aos dados de treinamento; já valores muito baixos podem resultar em modelos subajustados, com baixa precisão.

2.2.2.1 Número de camadas das RNAs

Definir o número de camadas em uma rede neural é multifatorial e depende da complexidade do problema, da quantidade de dados disponíveis e da capacidade computacional.

Para problemas simples, uma rede neural com uma ou duas camadas pode ser suficiente para encontrar padrões básicos nos dados. Já para problemas complexos, são exigidas redes neurais mais profundas com várias camadas ocultas. Associado a isso está a quantidade de dados disponíveis, uma vez que com uma quantidade de dados limitados, redes com poucas camadas tendem a se sair melhor, evitando o risco de sobreajuste. Já para uma grande quantidade de dados, as redes mais profundas são melhores para se aprender padrões mais complexos e obter melhores resultados.

As camadas ocultas se comportam como os "cérebros da rede neural", pois servem para executar tarefas complexas, como: capturar padrões e características não facilmente perceptíveis; criação de hierarquias de informação, e distribuição desses padrões com base nas de camadas anteriores.

O tipo de dados também ajuda a definir as arquiteturas de rede. Para dados de imagem, como os que iremos utilizar nesse trabalho, são necessárias redes neurais profundas, como as CNNs. Esse tipo de dado exige um grande poder computacional, então, para processadores limitados, é preferível usar redes com menor número de camadas.

Em resumo, o número de camadas é definido pela escolha de todos os parâmetros anteriormente explicados, de forma que se garanta um bom desempenho da rede, sem a presença de sobreajuste e considerando os recursos computacionais disponíveis.

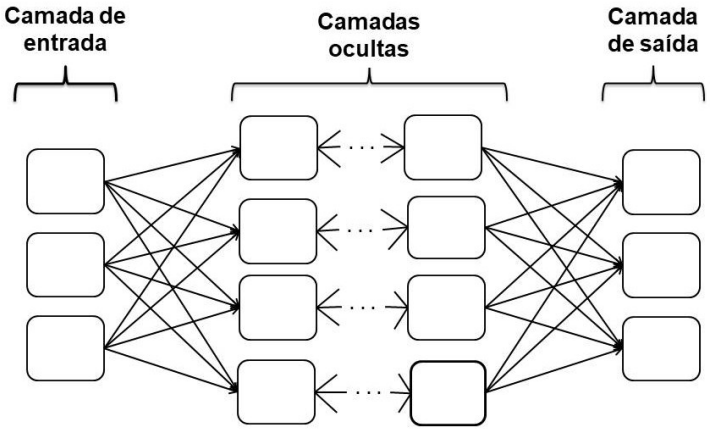


Figura 4 – Exemplo da interconexão entre camadas ocultas (4)

2.2.3 Funções de ativação para RNAs

A função de ativação e suas derivadas são fundamentais para o funcionamento de uma rede neural artificial, pois desempenham papéis cruciais no processo de aprendizado, permitindo que a rede aprenda padrões de dados complexos para realizar tarefas como classificação e regressão.

A escolha da função de ativação depende do tipo de informação que a rede está processando, já que essas são responsáveis pela introdução da não-linearidade da RNA, as quais possibilitam o aprendizado dos padrões complexos. Além disso, ela determina o sinal de saída de cada neurônio, definindo se a informação é ativada ou não.

Já a derivada dessa função de ativação é utilizada, pelo processo de retropropagação, para calcular o gradiente da função de custo em relação aos pesos da rede. Dessa forma, durante o aprendizado, o gradiente é usado para atualizar os pesos da rede, ajustando os parâmetros para minimizar a função de custo. Por isso, as funções devem possuir uma derivada bem comportada, pois o cálculo preciso do gradiente garante que a rede aprenda eficientemente e encontre a melhor solução.

Em nosso trabalho, utilizamos apenas três funções de ativação. Na camada convolucionária foi aplicada a função Unidade Linear Exponencial (ELU), na camada de achatamento foi utilizada a função Unidade Linear Retificada (RELU) e na camada densa foi aplicada a função Sigmoide.

Escolhemos a função ELU para as camadas convolucionárias, pois ela possui uma característica regularizadora. Consequentemente, não sofre com o problema da dissipação do gradiente e tem uma característica regularizadora, o que melhora a estabilidade da rede. Sua forma matemática é dada por:

$$\sigma(z) = \text{ELU}(x, \alpha) = \begin{cases} z, & \text{se } z \geq 0 \\ \alpha(e^z - 1), & \text{se } z < 0 \end{cases}, \tag{2.4}$$

Sua derivada é dada por:

$$\sigma'(z) = \begin{cases} 1, & \text{se } z \geq 0 \\ \text{ELU}(z, \alpha) + \alpha, & \text{se } z < 0 \end{cases}, \tag{2.5}$$

em que se pode perceber que essas saturam na parte negativa do seu domínio, mas isso não afeta sua eficácia na prática (15).

Na camada de achatamento, utilizamos a função ReLU, pois leva a uma convergência mais rápida durante o treinamento. Sua forma matemática é descrita por:

$$\sigma(z) = \text{ReLU}(z) = \max[0, z], \tag{2.6}$$

em que essa é bem próxima da função identidade, produzindo zero em metade do seu domínio. Sua derivada é do tipo: (15)

$$\sigma'(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0, \end{cases} \tag{2.7}$$

sendo não definida em 0, mas que pode ser implementada como sendo 0 ou 1 sem problemas. Esse processo gera derivadas grandes e estáveis, sendo 1 quando $x > 0$ e 0 quando $x < 0$. Assim, sua desvantagem está no fato de que essas unidades tendem a 'morrer' durante o treinamento (neurônios que produzem apenas saídas zeros), o que acontece pois a soma ponderada antes da aplicação dessa se torna negativa, fazendo com que a unidade produza zero (15).

Já para a camada densa, escolhemos a Função Sigmoide, pois ela comprime a saída para um intervalo entre 0 e 1, seu intervalo de validade. Sua representação matemática é:

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \tag{2.8}$$

que representa uma função contínua e capaz de preservar informações sobre a magnitude da pré-ativação, principalmente no intervalo próximo de $z = 0$, em que é quase linear (15).

Sua derivada é dada por:

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)), \tag{2.9}$$

que gera uma derivada do tipo $\sigma'(z) < 1$. Esse resultado pode gerar um problema de desaparecimento do gradiente para dados muito grandes, pois faz com que $\sigma(z) \to 0$, que implica no gradiente gerar uma baixa ou nenhuma aprendizagem (15).

2.2.4 Taxa de aprendizagem

A taxa de aprendizado é um parâmetro crucial no treinamento de modelos de aprendizado de máquina, pois determina a velocidade com que o modelo ajusta seus pesos e vieses em direção ao mínimo da função de perda, como ilustrado na figura 5 abaixo. A faixa ideal da taxa de aprendizado varia dependendo do problema, modelo, tamanho do conjunto de dados e complexidade do modelo, tendo como um intervalo típico $[10^{-5}, 10^{-1}]$.

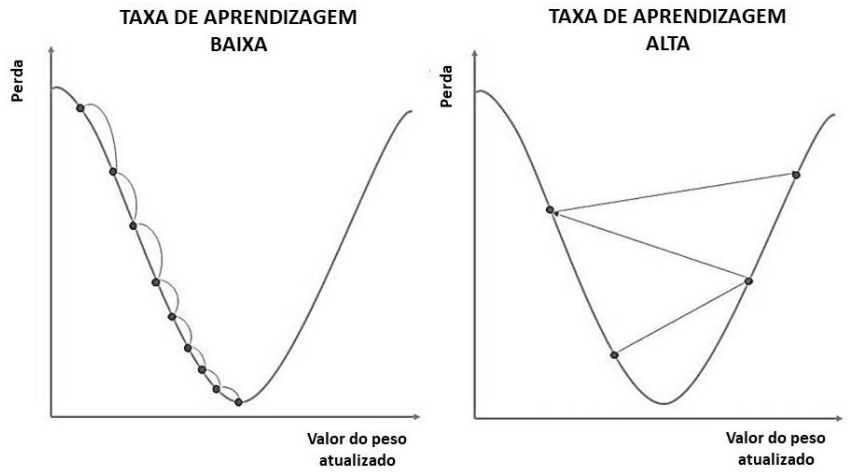


Figura 5 – Diferença da descida do gradiente para uma taxa de aprendizado alta e uma baixa (5)

Se o treinamento começar com uma taxa de aprendizado pequena (”pequenos passos”), o custo acaba aumentando, pois o tempo de processamento será maior, o que compromete a eficiência geral, gerando uma convergência para um mínimo local. Entretanto, sua vantagem é a sua alta precisão no monitoramento do desempenho do modelo, além de ser de fácil ajuste para valores maiores.

Já se começa com uma taxa de aprendizado muito alta (”grandes passos”), o custo acaba diminuindo, pois o tempo de processamento será menor, mas pode levar ao risco da função de ultrapassar o mínimo, o que pode gerar divergências.

2.2.4.1 Função de custo

Uma “função de perda” mede o erro entre uma previsão do modelo e o valor verdadeiro para um único ponto dos dados, enquanto uma “função de custo” é o valor agregado de todas as funções de custo através de todo o conjunto de treinamento, representando o erro global do modelo sobre todo o conjunto de dados. Usualmente, a função de custo usada é o Erro Quadrático Médio (MSE).

Treinar a rede significa encontrar o conjunto de parâmetros, pesos e vieses que minimiza a função de custo, ou seja, a diferença entre as previsões e os valores corretos, como podemos ver na figura 6 abaixo.

Tipicamente, os algoritmos de treinamento iteram ao longo da direção do gradiente negativo da função de custo até que esta se aproxime de um mínimo local ou global, quando o modelo parará de aprender. Logo, a melhora da eficiência do aprendizado de máquina se dá através do fornecimento do *feedback* de forma a minimizar o erro e encontrar o mínimo global ou menor mínimo local.

Inicialmente, a função de custo é calculada após fazer uma hipótese com parâmetros iniciais e depois estes parâmetros serão modificados seguindo uma regra iterativa em que o vetor de

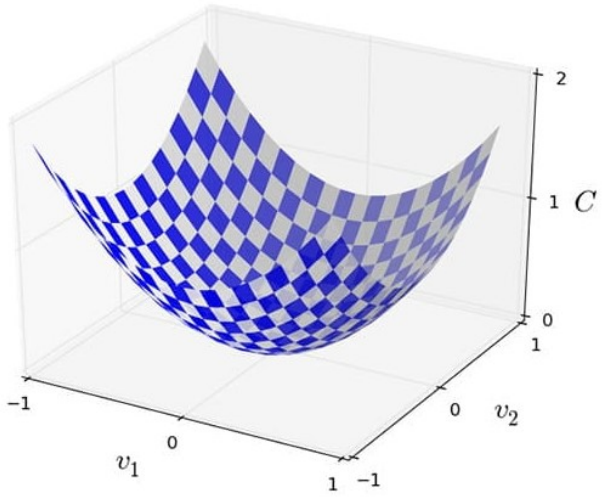


Figura 6 – Modelo de função de custo (6)

pesos w_0 e o ajuste a ser aplicado podem ser definidos como:

$$\Delta w \propto -\nabla J(w), \tag{2.10}$$

em que Δw é a variação do peso e ∇J é o gradiente da função de custo. Consequentemente, como o ajuste dos pesos depende do cálculo do gradiente, a equação da rede (ou função objetivo) deve ser diferenciável e contínua, isso é um dos motivos que ocasiona o Desaparecimento do Gradiente (6).

O método do gradiente descendente realiza o ajuste dos pesos no sentido contrário ao vetor gradiente da função de custo, com o objetivo de minimizá-la. Assim, podemos defini-lo como um conjunto de treinamento formado pelos pares (x_i, d_i), onde x_i é o i-ésimo vetor de entrada e d_i é a i-ésima saída desejada, com n sendo o número de elementos do conjunto de treinamento. A função de custo a ser minimizada será a soma dos quadrados dos erros, descrita pela seguinte equação:

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})^2 \tag{2.11}$$

em que d_i é a i-ésima saída desejada e y_i é a i-ésima saída da RNA (6).

2.2.5 Método de Descida do Gradiente

Baseado no Teorema de Cauchy, o método de descida do gradiente (BGD) é um dos algoritmos de otimização iterativa mais comuns no aprendizado de máquina. Sua função é encontrar o mínimo local da função de custo para o modelo.

Para compreender o funcionamento do método de descida do gradiente, recordemos que o vetor gradiente de uma função escalar diferenciável em um espaço de dimensão arbitrária "aponta" na direção de "mais rápido crescimento" da função. Assim, ao percorrermos o espaço

das variáveis independentes na direção oposta à do vetor gradiente, estaremos nos movendo na direção de um possível mínimo desta função, caso este exista.

Este método está ilustrado, para o caso unidimensional, na figura 7 abaixo.

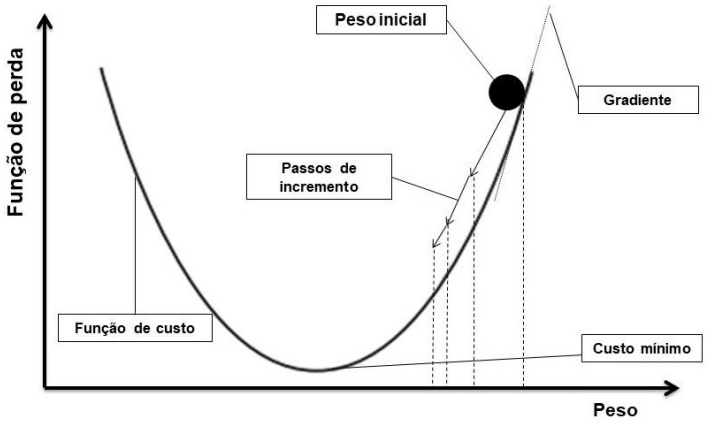


Figura 7 – Aplicação do Método do Gradiente Descendente em Aprendizado de Máquina (5)

O ponto de partida usado para avaliar o desempenho é arbitrário, então, ao escolhermos esse, deve-se realizar a primeira derivada, que gera a inclinação da linha tangente, a qual fornecerá as atualizações dos parâmetros pesos w e viés b. Essa inclinação é mais íngreme no ponto inicial, mas sempre que novos parâmetros são gerados, vai diminuindo gradualmente até o ponto mais baixo, que é chamado de ponto de convergência.

Para minimizar a função de custo, são necessários analisar a direção do gradiente e a taxa de aprendizagem, a qual é definida como o tamanho do passo tomado para atingir o ponto mínimo local ou global.

Como já explicado em sessão anterior, aplicaremos o método de BGD. A representação matemática para um mini-lote de dados de entrada $x^{(i:i+n)}$, com um mini-lote de rótulos correspondentes $y^{(i:i+n)}$, será atualizada para n exemplos de treinamento, da seguinte forma: (16)

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)}). \tag{2.12}$$

em que θ representa os parâmetros do modelo (w_i e b_i), η é a taxa de aprendizado, que controla o tamanho do passo de atualização, $\nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)})$ é o gradiente da função de perda J em relação aos parâmetros θ.

Dessa forma, as vantagens de utilização desse método são: (a) reduz a variância das atualizações de parâmetros, o que pode levar a uma convergência mais estável; e (b) pode utilizar otimizações de matriz altamente otimizadas, comuns em bibliotecas de aprendizado profundo de última geração, que tornam o cálculo do gradiente em relação a um mini-lote muito eficiente (16).

2.2.5.1 Algoritmo de otimização de descida de gradiente

Outro método de otimização do modelo frequentemente usado é o Método de Estimação de Momentos Adaptativos (Adam), que calcula as taxas de aprendizado adaptativas para cada parâmetro, e também armazena uma média exponencialmente decrescente de gradientes quadrados anteriores v_t, matematicamente descrita como:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{2.13}$$

em que v_t é a variância não centrada dos gradientes e g_t representa o gradiente, por conveniência da notação (16).

Como v_t são inicializadas como vetores de zeros, é possível observar que o método é tendencioso em relação a zero, especialmente durante os passos de tempo iniciais, quando as taxas de decaimento são pequenas, ou seja, β_1 e β_2 estão próximos de 1. (16)

O Adam também atua para contrabalancear os vieses, através do cálculo das estimativas de primeiro e segundo momento corrigidas por viés: (16)

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \tag{2.14}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \tag{2.15}$$

em que essas estimativas são utilizadas para atualizar os parâmetros, o que gera sua regra de atualização:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t, \tag{2.16}$$

onde ϵ é um parâmetro de regularização, que impede divergências e instabilidade numérica quando $\hat{v}_t$ fica pequeno (16).

Os autores propõem valores padrão de 0,9 para β_1, 0,999 para β_2 e 10^{-8} para ϵ. Eles mostram empiricamente que Adam funciona bem na prática e se compara favoravelmente a outros algoritmos de método de aprendizado adaptativo (16).

2.3 Redes Neurais Convolucionais(CNNs)

As Redes de Aprendizado Profundo são compostas por múltiplas camadas de neurônios interconectados, formando uma estrutura hierárquica, da qual extraem padrões complexos, através de um processo iterativo de treinamento, visando minimizar o erro entre as previsões e os valores reais. Baseado nesse tipo de arquitetura, foram desenvolvidas as Redes Neurais Convolucionais (CNNs) (17).

A escolha da arquitetura das CNNs se deu devido à sua capacidade de detectar padrões em conjuntos de dados grandes e complexos, o que é útil para classificar imagens astronômicas. Além disso, as CNNs não exigem características paramétricas pré-definidas dos objetos que estão sujeitos à medição ou classificação (7).

As CNNs possuem uma estrutura única que as torna altamente eficazes no processamento de imagens, ilustrada na imagem abaixo. Para entender esse processo, é necessário ter uma noção sobre as principais suas principais camadas:

1. **Camadas Convolucionais:** tem como objetivo extrair as características relevantes das imagens, através da aplicação de filtros (núcleos convolucionais) sobre a imagem de entrada, realizando uma operação de convolução. Uma vez que cada filtro detecta padrões específicos, como bordas, texturas ou cores, cujos filtros possuem pesos ajustáveis durante o treinamento. Logo, a saída dessas camadas é um mapa de ativação que representa a presença das características detectadas pelos filtros em diferentes regiões da imagem.
2. **Camadas de agrupamento**: tem como objetivo reduzir a dimensionalidade dos mapas de ativação, através de uma função de agrupamento máximo (*Max Pooling*) ou médio (*Average Pooling*), os quais simplificam a informação e tornam a rede mais eficiente sobre cada região do mapa de ativação. Seus benefícios estão em reduzir o número de parâmetros, diminuir o risco de superajuste e aumentar a invariância a pequenas translações e rotações na imagem.
3. **Camadas de achatamento**: A camada de achatamento em CNNs atua como um "achatador" de matrizes, transformando o mapa de ativação multidimensional de saída da última camada convolucional em um vetor unidimensional. Essa transformação é essencial para conectar as camadas convolucionais com as camadas densas, que exigem entrada em formato unidimensional. Elas permitem que as redes neurais aprendam relações complexas entre os recursos extraídos pelas convoluções, e ajudam a reduzir a dimensionalidade dos dados, simplificando o processamento e diminuindo a probabilidade de sobreajuste.
4. **Camadas completamente conectadas:** tem como objetivo classificar ou prever a saída final da rede, uma vez que conectam cada neurônio da camada anterior a todos os neurônios da camada atual, através de uma operação linear seguida de uma função de ativação. As camadas totalmente conectadas se assemelham às redes neurais tradicionais, mas operam sobre a representação de características extraídas pelas camadas convolucionais e as de agrupamento, que aumentam sua eficiência. (18)

Para que se possa entender o comportamento das CNN, é necessário que nos aprofundemos sobre as operações internas a cada uma dessas camadas, para que se possa prever os comportamentos de saída pós-extração dos dados.

2.3.1 Camadas de convolução

As CNN recebem esse nome pois se baseiam na operação matemática de convolução, que se trata de uma operação que combina duas funções para produzir uma terceira função que expressa como a forma de uma delas é modificada pela outra. No contexto das CNNs, uma função seria a imagem de entrada e a outra seria um filtro, que é essencialmente um "detector"de padrões, principalmente imagens.

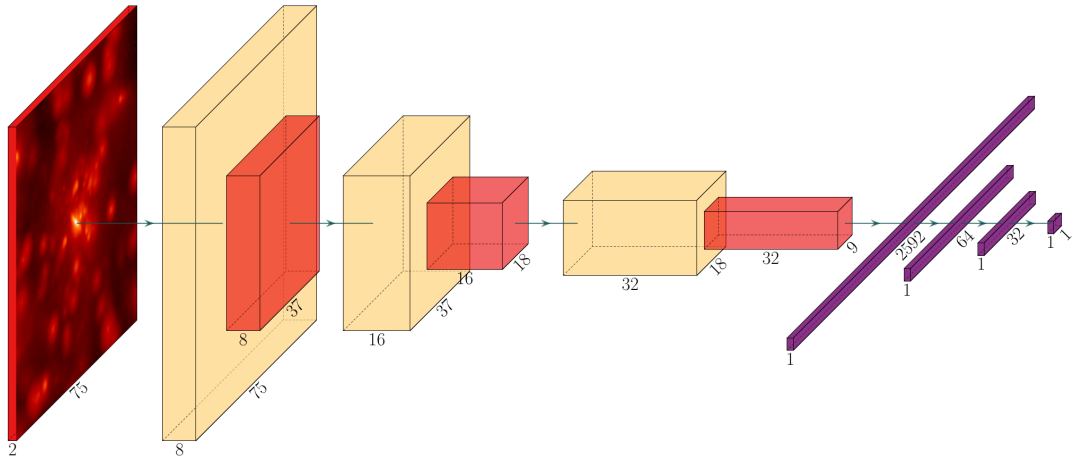


Figura 8 – Estrutura de uma CNN dividida em etapas (7)

2.3.1.1 A operação de convolução

Se assumirmos que uma imagem possui apenas um canal, em tons de cinza, podemos escrevê-la através da função:

$$I : \{1, ..., n_1\} \times \{1, ..., n_2\} \to W \subseteq \mathbb{R}, (i, j) \to I_{i,j}, \tag{2.17}$$

em que a imagem I pode ser representada por uma matriz de tamanho $n_1 \times n_2$ (19).

Escrevendo o filtro $K \in \mathbb{R}^{2h_1+1 \times 2h_2+1}$ na forma matricial, dado por:

$$K = \begin{pmatrix} K_{-h_1,-h_2} & \dots & K_{-h_1,h_2} \\ \vdots & K_{0,0} & \vdots \\ K_{h_1,-h_2} & \dots & K_{h_1,h_2} \end{pmatrix}, \tag{2.18}$$

em que esta representação matricial define a estrutura de um filtro de convolução 2D, comumente usado em processamento de imagens. Os elementos $K_{i,j}$ representam os pesos aplicados aos pixels da imagem em uma vizinhança local, centrada no pixel sendo processado, de tamanho $(2h_1 + 1) \times (2h_2 + 1)$ (19).

A aplicação deste filtro tipicamente envolve uma operação de convolução discreta, em que o filtro "desliza" sobre a imagem, calculando a soma ponderada dos produtos entre os elementos do filtro e os pixels correspondentes da imagem. A escolha dos pesos $K_{i,j}$ determina o efeito do filtro, como suavização, detecção de bordas, etc. (19)

Realizando uma convolução discreta da imagem I com o filtro K, dada por:

$$(I \times K)_{r,s} = \sum_{u=-h_1}^{h_1} \sum_{v=-h_2}^{h_2} K_{u,v} I_{r+u,s+v}, \tag{2.19}$$

em que para cada pixel (r, s) na imagem de saída é o resultado da convolução, calculado através da soma ponderada dos produtos entre os elementos do filtro K e os pixels correspondentes de I.

Essa operação possibilita que os índices u e v percorram a extensão do filtro e realizem futuras detecção e suavização de bordas (19).

Um cuidado que deve ser tomado é definir adequadamente o comportamento dessa operação em relação às bordas da imagem precisa. Assim, podemos aplicar um filtro de suavização, como no caso do filtro Gaussiano discreto $K_G(\sigma)$, definido por:

$$(K_G(\sigma))_{r,s} = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{r^2+s^2}{2\sigma^2}\right), \tag{2.20}$$

em que σ é o desvio padrão da distribuição Gaussiana (19).

Essa operação, definida pela operação (2.19) é semelhante à equação (2.21) que será apresentada na subseção a seguir, por isso essa camada é chamada de convolucional.

2.3.1.2 Operações na camada convolucional

Seja a camada l uma camada convolucional, em que a entrada da camada l comprime $m_1^{(l-1)}$ mapas de características da camada anterior, com cada um tendo tamanho $m_2^{(l-1)} \times m_3^{(l-1)}$. Generalizando esse resultado, teremos que o i-ésimo mapa de características na camada l, denotado por $Y_i^{(l)}$, é calculado como:

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{i,j}^{(l)} \times Y_j^{(l-1)}, \tag{2.21}$$

Em que $B_i^{(l)}$ é uma matriz de viés, $K_{i,j}^{(l)}$ é o filtro de tamanho $2h_1^{(l)}+1 \times 2h_2^{(l)}+1$, que conecta o j-ésimo mapa de características na camada $(l-1)$ com o i-ésimo mapa de características na camada l (19).

O tamanho dos mapas de característica $m_2^{(l)} \times m^{(l)_3}$ é influenciado por efeitos de borda. Então, ao aplicar a convolução discreta, apenas na chamada região válida do mapa de característica, apenas para pixels onde a soma da equação é definida adequadamente, os mapas de características de saída têm tamanho: $m_2^{(l)} = m_2^{(l-1)} - 2h_1^{(l)}$ e $m_3^{(l)} = m_3^{(l-1)} - 2h_2^{(l)}$ (19).

Para relacionar a camada convolucional com cada mapa de características $Y_i^{(l)}$, na camada l, com $m_2^{(l)} \cdot m_3^{(l)}$ unidades organizadas em uma matriz bidimensional, a unidade na posição (r,s) calcula a saída da seguinte forma:

$$(Y_i^{(l)})_{r,s} = (B_i^{(l)})_{r,s} + \sum_{j=1}^{m_1^{(l-1)}} (K_{i,j}^{(l)} \times Y_j^{(l-1)})_{r,s}, \tag{2.22}$$

$$(Y_i^{(l)})_{r,s} = (B_i^{(l)})_{r,s} + \sum_{j=1}^{m_1^{(l-1)}} \sum_{u=-h_1^{(l)}}^{h_1^{(l)}} \sum_{v=-h_2^{(l)}}^{h_2^{(l)}} (K_{i,j}^{(l)})_{u,v} (Y_j^{(l-1)})_{r+u,s+v}, \tag{2.23}$$

em que os pesos treináveis da rede podem ser encontrados nos filtros $K_{i,j}^{(l)}$ e nas matrizes de viés $B_i^{(l)}$ (19).

A subamostragem é usada para diminuir o efeito do ruído e das distorções da imagem, pois usa fatores de salto $s_1^{(l)}$ e $s_2^{(l)}$, cuja ideia básica é pular um número fixo de pixels, tanto na direção horizontal quanto na vertical, antes de aplicar o filtro novamente. Os fatores de salto podem ser dados por: $m_2^{(l)} = m_2^{(l-1)} - 2h_1^{(l)}s_1^{(l)} + 1$ e $m_3^{(l)} = m_3^{(l-1)} - 2h_2^{(l)}s_2^{(l)} + 1$, que representam o tamanho dos mapas de características de saída (19).

Aplicando uma camada l não linear, sua entrada será dada por $m_1^{(l)}$ mapas de características e sua saída terá $m_1^{(l)} = m_1^{(l-1)}$ mapas de características, cada uma de tamanho $m_2^{(l-1)} \times m_3^{(l-1)}$, tal que $m_2^{(l)} = m_2^{(l-1)}$ e $m_3^{(l)} = m_3^{(l-1)}$, dado por:

$$Y_i^{(l)} = f(Y_i^{(l-1)}), \tag{2.24}$$

em que f é a função de ativação usada na camada l e opera ponto a ponto (19).

Associado a essa, há os coeficientes de ganho adicionais, do tipo:

$$Y_i^{(l)} = g_i f(Y_i^{(l-1)}), \tag{2.25}$$

em que a união de todos esses processos constitui uma única camada convolucional (19).

2.3.1.3 Operação de Retificação

Seja a camada l uma camada de retificação, com entrada $m_1^{(l-1)}$, mapas de características de tamanho $m_2^{(l-1)} \times m_3^{(l-1)}$. O valor absoluto de cada componente dos mapas de características é dado por:

$$Y_i^{(l)} = |Y_i^{(l)}|, \tag{2.26}$$

em que o valor absoluto é calculado ponto a ponto, de modo que a saída consiste em $m_1^{(l)} = m_1^{(l-1)}$ mapas de características inalterados em tamanho (19).

2.3.1.4 Operação com camadas de Normalização de Contraste Local

A tarefa de uma camada de normalização de contraste local é impor competição local entre unidades adjacentes dentro de um mapa de características e unidades na mesma localização espacial em diferentes mapas de características.

Seja l uma camada de normalização de contraste, os dados $m_1^{(l-1)}$ dos mapas de características, de tamanho $m_2^{(l-1)} \times m_3^{(l-1)}$, terão uma saída que compreende $m_1^{(l)} = m_1^{(l-1)}$ mapas de características de tamanho inalterado. A operação de normalização subtrativa calculará:

$$Y_i^{(l)} = Y_i^{(l-1)} - \sum_{j=1}^{m_1^{(l-1)}} K_G(\sigma) \times Y_j^{(l-1)}, \tag{2.27}$$

em que $K_G(\sigma)$ é o filtro Gaussiano da equação (19).

Um esquema alternativo à normalização local é a normalização de brilho, proposta para ser usada na combinação de unidades lineares retificadas, cuja saída l é dada por:

$$(Y_i^{(l)})_{r,s} = \frac{(Y_i^{(l-1)})_{r,s}}{(\kappa + \mu \sum_{j=1}^{m_1^{(l-1)}} (Y_j^{(l-1)})_{r,s}^2)^{\mu}}, \tag{2.28}$$

em que κ, λ, μ são hiper parâmetros que podem ser definidos usando um conjunto de validação (19).

A soma dessa equação também pode percorrer subconjuntos dos mapas de características na camada $(l-1)$, como as camadas de normalização N_S e as de contraste local N_B (19).

2.3.2 Camada de Pooling

A motivação para subamostrar os mapas de características obtidos pelas camadas anteriores é a robustez ao ruído e distorções. Em geral, o *pooling* opera colocando janelas em posições não sobrepostas em cada mapa de características e mantendo um valor por janela, de modo que os mapas de características são subamostrados (19).

Seja l uma camada de *pooling*, com uma saída que compreende $m_1^{(l)} = m_1^{(l-1)}$ mapas de características de tamanho reduzido. Consideramos dois tipos:

- o *pooling* médio (P_A) ocorre quando se é aplicado um filtro chamado de *boxcar*;
- o *pooling* máximo (P_M) ocorre quando o valor máximo de cada janela é tomado, de forma a obter uma convergência mais rápida durante o treinamento. (19)

Portanto, ambas as técnicas podem ser aplicadas usando janelas sobrepostas, de tamanho $2p \times 2p$, colocadas a q unidades de distância, para que as janelas se sobreponham em $q < p$. Essa técnica é utilizada para reduzir a chance de superajuste do conjunto de treinamento (19).

2.3.3 Camadas de achatamento e camadas densas

As camadas de achatamento das CNNs não realizam operações matemáticas complexas, pois sua função principal é reordenar os elementos do tensor de entrada, sem alterar os valores. Esse reordenamento se dá através da concatenação e empilhamento das camadas, uma em cima da outra, para se formar uma única coluna, que depois será transformada em uma única linha (20).

É possível representar matematicamente esse reordenamento através de um mapeamento de índices. Como exemplo fictício, seja um tensor de entrada X com dimensões (N, H, W, C), onde: N é o número de exemplos no lote; H é a altura do mapa de ativação; W é a largura do mapa de ativação; C é o número de canais. A saída dessa camada, Y, terá dimensões $(N, H \times W \times C)$, representada da seguinte forma:

$$Y[n, i] = X[n, h, w, c], \tag{2.29}$$

em que n é o índice do exemplo, i é o índice do elemento no vetor de saída. Essa operação apenas garante que cada elemento do tensor de entrada seja mapeado para um elemento único no tensor de saída.

Portanto, a representação matemática da camada de achatamento é uma função de mapeamento de índices que transforma as coordenadas multidimensionais do tensor de entrada em um índice linear no vetor de saída. Esse passo é crucial para que essas informações possam ser inseridas nas camadas densas, que exigem vetores unidimensionais como entrada (20).

As camadas densas são capazes de aprender interações globais entre as camadas compactadas pelo achatamento, pois podem aprender representações de alto nível que capturam o significado global dos dados extraídas pelas convoluções (20).

2.3.4 Camada Totalmente Conectada

Seja a camada l uma camada totalmente conectada, ligada a uma camada $(l-1)$ também totalmente conectada, é possível se aplicar a função sigmoide. Caso contrário, a camada l espera $m_1^{(l-1)}$ mapas de características, de tamanho de entrada $m_2^{(l-1)} \times m_3^{(l-1)}$. A i-ésima unidade na camada l pode ser calculada, como:

$$y_i^{(l)} = f(z_i^{(l)}), \tag{2.30}$$

$$z_i^{(l)} = \sum_{j=1}^{m_1^{(l-1)}} \sum_{r=1}^{m_2^{(l-1)}} \sum_{s=1}^{m_3^{(l-1)}} w_{i,j,r,s}^{(l)} (Y_{j,r,s}^{(l-1)}), \tag{2.31}$$

em que $w_{i,j,r,s}^{(l)}$ denota o peso, conectando uma unidade da posição (r, s) no j-ésimo mapa de características, da camada $(l-1)$, a i-ésima unidade na camada l (19).

Portando, as camadas convolucionais são usadas para aprender uma hierarquia de características e uma ou mais camadas totalmente conectadas são usadas para fins de classificação com base nas características calculadas. Note que uma camada totalmente conectada já inclui as não linearidades, enquanto para uma camada convolucional as não linearidades são separadas em sua própria camada (19).

3 Metodologia

A metodologia será dividida em três etapas: na primeira, será descrita como foi modelada a CNN aplicada nesse projeto; na segunda, será relatado o processo de obtenção dos dados para a classificação de fusão de galáxias, passando por suas características e limitações, para então tentar prever os impactos desses dados na CNN; na terceira, serão apresentados os critérios para a análise de fusão ou não fusão de galáxias.

3.1 Descrição simplificada da CNN aplicada

O modelo que utilizamos neste trabalho é uma CNN, escrita na linguagem *Python*, em que são utilizadas as bibliotecas *Keras* e *TensorFlow*, para uma tarefa de classificação binária. A arquitetura dessa rede foi projetada para processar dados de imagem e otimizada para performance e generalização.

Sua entrada é um tensor com formato $(2, 75, 75)$, que indica dois canais de entrada (filtros ACS F814W e WFC3 F160W), em que cada um possui dimensões 75×75 pixels. Devido a esse formato de entrada, escolhemos *channels_first* para que o modelo leia primeiro os canais e depois as dimensões. Caso essa leitura não seja feita corretamente, o modelo pode ler que o tensor de entrada possui 75 canais de dimensão 2×75 pixels.

A arquitetura da rede é composta por dois blocos convolucionais, em que cada um consiste em uma camada *Conv2D*, seguida por *BatchNormalization*, *MaxPooling2D* e *Dropout*. O primeiro utiliza 32 filtros, com dimensão 5×5, enquanto o segundo bloco emprega 64 filtros, com dimensão 3×3. A função de ativação ELU é aplicada em ambas as camadas convolucionais. A normalização em lote auxilia na aceleração do treinamento e melhora a estabilidade do modelo, enquanto o *Max pooling* reduz a dimensionalidade espacial e o *Dropout*, com taxa de 0.4.

Após os blocos convolucionais, a saída é achatada utilizando a camada de achatamento, a qual é alimentada por duas camadas densas totalmente conectadas. A primeira camada densa possui 128 neurônios e a segunda possui 32 neurônios. Ambas utilizam a função de ativação ReLU, com regularização $L2$, de 0.05 e camadas de *Dropout*, aplicadas após cada camada densa.

A camada de saída consiste em um único neurônio com função de ativação sigmoid, que produz um valor entre 0 e 1, representando a probabilidade de pertencer à classe positiva na classificação binária.

O modelo é compilado utilizando o otimizador Adam, com uma taxa de aprendizado de 0.0001. A função de perda utilizada é a *binary_crossentropy*, apropriada para problemas de classificação binária. A acurácia é utilizada como métrica de avaliação do desempenho do modelo.

Em suma, a arquitetura da CNN descrita neste trabalho foi projetada para extrair características relevantes das imagens de entrada e realizar a classificação binária. A utilização de técnicas como normalização em lote, *Dropout* e regularização $L2$ para tentar gerar uma robustez no modelo e auxiliar na prevenção de sobreajuste. A escolha do otimizador Adam e da função de perda são consistentes com as práticas recomendadas para problemas de classificação binária.

3.2 Obtenção e extração dos dados de fusão de galáxias

As imagens utilizadas neste trabalho foram obtidas no STSI (21), disponibilizadas pelos autores (7), que realizaram a coleta desses dados em literaturas anteriores, baseadas na simulação cosmológica Illustris-1, usando imagens instantâneas do subconjunto de imagens de galáxias com deslocamento para o vermelho $z = 2$.

Nos dados selecionados por (7) foram aplicados dois filtros que simulam equipamentos embarcados no HST. Os filtros aplicados foram o ACS F814W, no infravermelho, com $\approx 1600nm$, e o WFC3 F160W, no infravermelho próximo ao espectro da luz visível, com $\approx 840nm$.

> Em Snyder et al. (2019), os autores modificam as imagens para refletir as qualidades observacionais do Telescópio Espacial Hubble (HST) e do Telescópio Espacial James Webb (JWST). Primeiro, as imagens de linha de base foram convolucionadas com uma função de espalhamento de ponto (PSF) modelo apropriada para cada filtro (nosso conjunto de dados "pristine"). [Tradução livre] (7)

As imagens e rótulos por nós extraídos são oriundos do arquivo do tipo Sistema de Transporte Flexível de Imagens (.fits), formato de arquivo padrão utilizado para armazenar dados astronômicos, especialmente imagens e dados espectroscópicos, com o rótulo *pristine*, imagens que não tiveram nenhuma interferência ou tratamento que possa afetar sua qualidade original.

Esse arquivo em específico é dividido em 02 cabeçalhos: o HDU[0], que contém 15426 imagens, com dois canais e dimensões 75 x 75; e o HDU[1], que contém 15426 rótulos, sendo 8120 que indicam fusão e 7306 que indicam não fusão. As imagens são extraídas como na figura 9 abaixo.

Sendo assim, são consideradas fusões as imagens que apresentarem as características apontadas na seção 2.2, em que os objetos são divididos em duas classes: positiva ("P"), em que há a fusão, e negativa ("N"), em que não há fusão. Apesar da delimitação do recorte de espaço-tempo simulado, qualquer algoritmo de classificação está passível de ocorrer alguns falsos positivos, não-fusões que parecem fusões, e falsos negativos, fusões que parecem não-fusões.

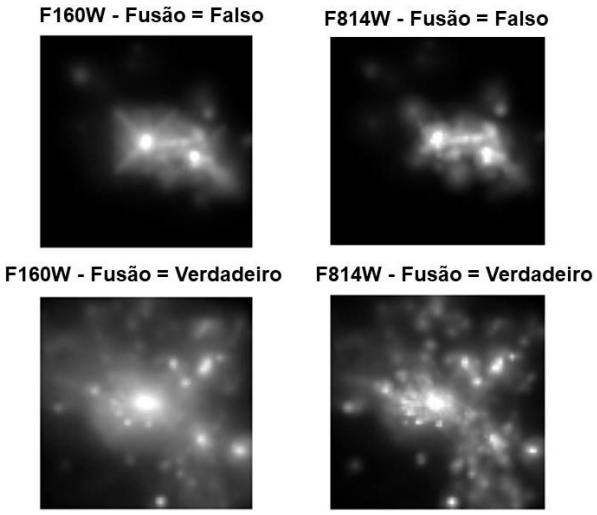


Figura 9 – Exemplos de imagens, de cada filtro, extraídas do arquivo

3.2.1 Matriz de confusão

Um classificador mapeia instâncias I de dados observados para classes previstas, podendo produzir uma saída contínua, como uma probabilidade de pertencer a uma classe, a qual pode ser usada para determinar a classe mais provável à qual pertence à instância observada. Classificadores discretos fornecem diretamente um rótulo de classe Y (previsão) que é elemento de um conjunto discreto.

Para um problema de classificação binária, cada instância I é classificada como pertencente a uma das duas classes: positiva P ou negativa N. No que segue, vamos representar o número de elementos em uma determinada classe ou com determinado rótulo por um símbolo em itálico. Por exemplo: P significa o número de instâncias pertencentes à classe P, N é o número de instâncias pertencentes à classe N.

Se for necessário, usaremos o símbolo ˆ para indicar uma previsão do classificador ou um número de previsões de determinada classe. Por exemplo: $\hat{P}$ significa número de previsões da classe P.

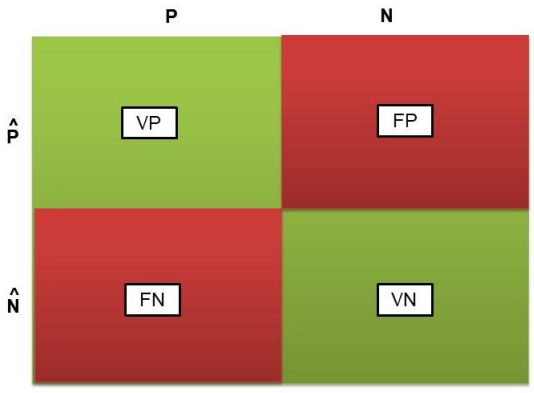


Figura 10 – Matriz de confusão teórica (8)

São gerados quatro resultados possíveis:

- instância positiva e classificação positiva, VP;
- instância positiva e classificação negativa, FN;
- instância negativa e classificação negativa, VN;
- instância negativa e classificação positiva, FP. (8)

Através dos números destas ocorrências, podemos extrair algumas métricas de desempenho (8):

- Taxa de FP: Representa a proporção de exemplos negativos que foram erroneamente classificados como positivos.

$$\text{Taxa de FP} = \frac{FP}{N} \tag{3.1}$$

- Taxa de VP, também chamada *sensibilidade* (*recall*): Representa a proporção de exemplos positivos que foram classificados corretamente como positivos.

$$\text{Taxa de VP} = \frac{VP}{P} \tag{3.2}$$

  Precisão: Indica a proporção de exemplos classificados como positivos que realmente são positivos.

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{3.3}$$

- Acurácia: Representa a proporção total de exemplos classificados corretamente.

$$\text{Acurácia} = \frac{VP + VN}{P + N} \tag{3.4}$$

- Sensibilidade (*Recall*): Indica a proporção de exemplos positivos que foram corretamente identificados.

$$\text{Taxa de VP} = \frac{VP}{P} \tag{3.5}$$

- Especificidade: Indica a proporção de exemplos negativos que foram corretamente identificados.

$$\text{Taxa de VN} = \frac{VN}{N} \tag{3.6}$$

- *F1-Score*: É a média harmônica entre precisão e sensibilidade, fornecendo um único valor que equilibra esses dois aspectos.

$$F1 = \frac{2}{\frac{1}{\text{precisão}} + \frac{1}{\text{sensibilidade}}} \tag{3.7}$$

Recorrentemente, as taxas principais são representadas na *matriz de confusão*, como mostrado na figura 10, na qual o eixo X representa os valores das classes reais das instâncias e o eixo Y as classes previstas pelo classificador.

3.2.2 Características Operacionais do Receptor (ROC)

A curva ROC é uma ferramenta poderosa para avaliar o desempenho de classificadores binários. É particularmente útil para problemas com classes desbalanceadas, onde a acurácia pode ser enganosa, pois fornece uma visão abrangente do desempenho do classificador, considerando tanto a taxa de verdadeiros positivos quanto a taxa de falsos positivos. Já a Área Abaixo da Curva (AUC) é uma métrica importante para quantificar o desempenho geral do classificador (8).

Os gráficos ROC são gráficos bidimensionais em que a taxa de VP é plotada no eixo Y e a taxa de FP é plotada no eixo X. A relação entre elas descreve as compensações relativas entre benefícios VP e custos FP. Um classificador discreto é aquele que produz apenas um conjunto discreto de rótulos de classe. Cada classificador discreto produz um par correspondente a um único ponto no espaço ROC, como os presentes na figura 11:

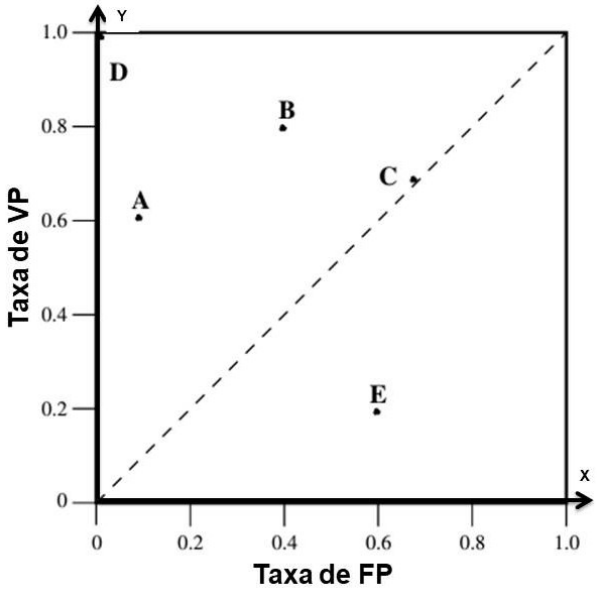


Figura 11 – Gráfico ROC básico com cinco classificadores discretos (8)

Vários pontos no espaço ROC são importantes de serem observados, sendo os principais:

- A coordenada (0,0) representa o ponto em que nunca é emitida uma classificação positiva;
- A coordenada (1,1) representa o ponto em que nunca é emitida uma classificação negativa;
- A coordenada (0,1) representa a classificação perfeita;
- Um ponto no espaço ROC é dito “melhor” que outro se a estiver “mais alto” na mesma abcissa (taxa $VP >$ taxa de FP). (8)

Quando um certo classificador discreto é aplicado a um conjunto de teste, produz uma única matriz de confusão, que por sua vez corresponde a um ponto no espaço ROC, de forma que um classificador discreto produz apenas um único ponto no espaço ROC. Esses valores podem ser probabilidades estritas ou podem ser pontuações gerais ou não calibradas, em que uma pontuação mais alta indica uma probabilidade mais alta, o que pode descrevê-lo como um ‘classificador probabilístico’, mesmo que sua saída não seja uma probabilidade no sentido estrito (8).

Observar a linha diagonal gerada abaixo da curva é essencial, pois ela indica se o classificador está classificando aleatoriamente, quando a taxa de verdadeiros positivos é igual à taxa de falsos positivos. Quanto mais pontos estiverem em cima ou próximo a ela, mais aleatório será o resultado, como o demonstrado na figura 12.

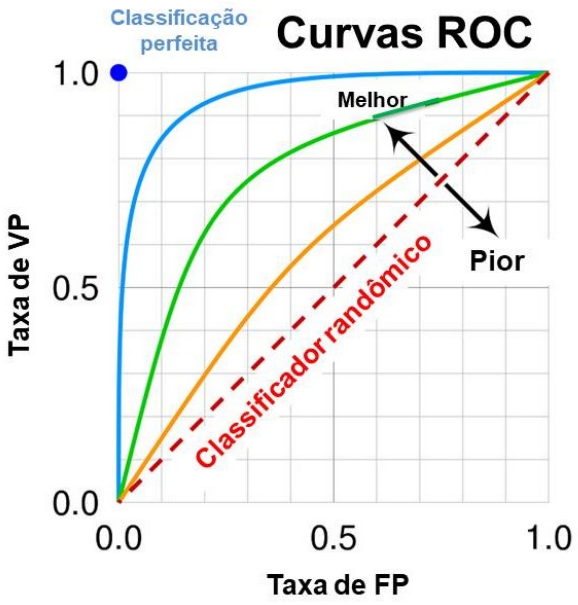


Figura 12 – Curvas ROC com seus parâmetros aplicados (9)

Outra métrica importante a ser analisada é a AUC, a qual mede o desempenho geral do classificador e representa a probabilidade de que o classificador classifique um exemplo positivo com uma pontuação maior do que um exemplo negativo. As AUC podem ser classificadas como:

- 1, que indica um classificador perfeito;
- 0.5, que indica um classificador aleatório;
- 0, que indica um classificador que classifica todos os exemplos incorretamente. (8)

3.3 Análise de fusão e não fusão de galáxias

Um período particularmente interessante para se detectar fusões de galáxias é conhecido como ”meio-dia cósmico”, pois, durante ele, as taxas de formação estelar foram as mais altas e havia uma quantidade significativa de massa estelar reunida em corpos galáticos, o que gerou um desvio para o vermelho $z \sim 2 - 3$. Nesse contexto, este período ainda não é totalmente compreendido, pois a taxa de ocorrência de eventos de fusão principais pode se tornar constante ou começar a diminuir durante o período $1 < z < 3$. Esse resultado discorda dos modelos teóricos, que preveem que as taxas de fusão principais continuam a aumentar durante este período (7).

Detectar fusões de galáxias em observações por métodos convencionais é bastante caro e demorado, pois depende da disponibilidade de dados profundos, de multi-comprimento de onda

de banda larga ou de espectroscópicos. Outro fator é que essa análise pode ser realizada por um grande número de pessoas, toda via esse processo se tornará cada vez mais demorado à medida que os volumes de dados aumentarem, além de estarem sujeitos aos vieses dos classificadores humanos (7).

Então, antes de se pensar em automatizar a análise e classificação de fusão de galáxias, é necessário que se compreendam as características básicas a serem rastreadas por uma rede neural especializada nesse processo.

3.3.0.1 Análise Morfológica

As primeiras classificações de galáxias se baseavam em sua forma visual, pelo Hubble, em 1926. Estas foram agrupadas em um sistema hierárquico, de acordo com características distintivas, dividido em galáxias espirais, elípticas e irregulares, como demonstrado na figura 13.

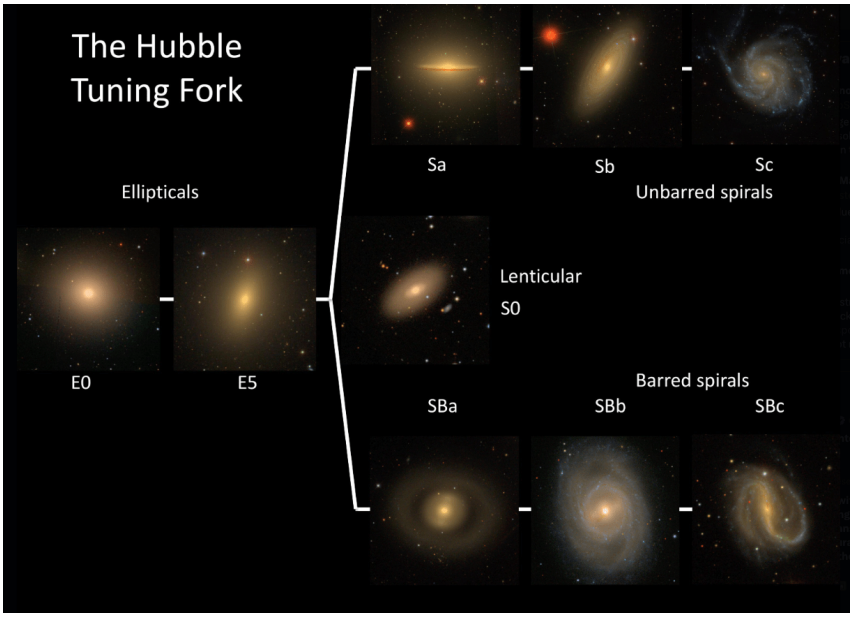


Figura 13 – "O diapasão de Hubble"ou Esquema de classificação de galáxias de Hubble (10)

Apesar de esse sistema servir como uma base fundamental e sólida para a análise morfológica de galáxias, ainda é insuficiente para capturar a complexidade de interações gravitacionais que ocorrem quando há o fenômeno de interação gravitacional entre galáxias, que pode gerar sua fusão, uma vez que esse fenômeno causa distorções significativas na matéria e no tecido do espaço-tempo.

3.3.0.2 Marcas de Interação

O estudo das fusões de galáxias tem sido impulsionado pela maior disponibilidade de super-computadores, atrelados a universidades, que utilizam modelos computacionais extremamente robustos para simular a evolução de galáxias através de sua interação (1).

Associado a isso, há o desenvolvimento de telescópios espaciais cada vez mais poderosos, pois antes tínhamos apenas os dados do HST e hoje já temos acesso aos dados do JWST, um equipamento mais recente e com filtros no infravermelho que permitem a sobreposição, enquanto se realiza a análise espectrográfica do objeto cósmico analisado, que em pouco tempo deve sanar uma série de questionamentos da cosmologia e da astrofísica, mas também gerar uma nova leva de hipóteses sobre a evolução cosmológica.

Logo, tais tecnologias, associadas às técnicas de processamento de imagens (remoção de ruído, aumento de contraste, zoom, ajuste de luminosidade, etc.), são ferramentas que possibilitaram a identificação de detalhes morfológicos, os quais auxiliam na compreensão da formação de estruturas complexas, como as caudas de maré, a presença de núcleos duplos e intensificação da formação estelar.

As Caudas de Maré, ilustradas na figura 14, são estruturas alongadas e filamentosas formadas pelo material estelar arrancado dos discos galácticos durante a interação gravitacional, principalmente os gases. Assim, através da análise espectrográfica dessas caudas, é possível estimar a força da interação, a distância entre as galáxias e a massa envolvida na fusão (22).

Figura 14 – Exemplo de processo de fusão de galáxias (11)

As distorções geradas por esse processo de interação gravitacional criam um padrão de irregularidades característico das fusões entre galáxias, seja em seu início, em que as caudas de maré são evidentes, seja em seu final, em que a nova galáxia formada ainda está ajustando sua forma e pode ter esse tipo de rastro até a conclusão desse processo.

Como a fusão entre galáxias não é um processo instantâneo, os núcleos de cada uma das galáxias envolvidas tendem a permanecer visíveis até que haja um colapso gravitacional entre eles, o que gera os famosos núcleos duplos. Assim, o estudo da evolução desses núcleos fornece pistas sobre a dinâmica interna da fusão e o tempo necessário para a integração completa das galáxias (22).

A formação estelar é um processo fundamental para a evolução de galáxias e as fusões são responsáveis por intensificar esse processo. O aumento da interação gravitacional cresce com r^2 com a diminuição do raio que separa os núcleos, fazendo com que a poeira e o gás galáctico sejam comprimidos a ponto de desencadearem uma reação em cadeia que causa um pico de formação estelar, como o ilustrado na figura 15.

Figura 15 – Processo de fusão de duas galáxias da constelação de Cetus capturado pelo JWST (12)

A consequência visível é que esse processo gera regiões de alta intensidade luminosa, em um núcleo ainda distorcido, que indica surtos de formação estelar. Mas também é possível realizar uma análise espectrográfica desse cenário, em que será possível observar a emissão em diferentes comprimentos de onda (infravermelho, ultravioleta e raios-x), que permite determinar as propriedades das estrelas em formação e estudar as condições físicas nas regiões de formação estelar (22).

Portanto, uma vez que conhecemos esses parâmetros, torna-se possível imaginar um processo de automatização da classificação dos dados de fusão de galáxias por meio da utilização de uma CNN. Todavia, é necessário que se compreenda o que são Redes Neurais Artificiais (RNAs) e suas características.

3.4 Descrição do modelo

O modelo que utilizamos neste trabalho é uma CNN, escrita na linguagem *Python*, em que são utilizadas as bibliotecas *Keras* e *TensorFlow*, para uma tarefa de classificação binária. A arquitetura dessa rede foi projetada para processar dados de imagem e otimizada para performance e generalização.

Sua entrada é um tensor com formato $(2, 75, 75)$, que indica dois canais de entrada (filtros ACS F814W e WFC3 F160W), em que cada um possui dimensões 75×75 pixels. Devido a esse formato de entrada, escolhemos *channels_first* para que o modelo leia primeiro os canais e depois as dimensões. Caso essa leitura não seja feita corretamente, o modelo pode ler que o tensor de entrada possui 75 canais de dimensão 2×75 pixels.

A arquitetura da rede é composta por dois blocos convolucionais, em que cada um consiste em uma camada *Conv2D*, seguida por *BatchNormalization*, *MaxPooling2D* e *Dropout*. O primeiro utiliza 32 filtros, com dimensão 5×5, enquanto o segundo bloco emprega 64 filtros, com dimensão 3×3. A função de ativação ELU é aplicada em ambas as camadas convolucionais. A

normalização em lote auxilia na aceleração do treinamento e melhora a estabilidade do modelo, enquanto o *Max pooling* reduz a dimensionalidade espacial e o *Dropout*, com taxa de 0,4.

Após os blocos convolucionais, a saída é achatada utilizando a camada de achatamento (*flatten*), a qual é alimentada por duas camadas densas totalmente conectadas. A primeira camada densa possui 128 neurônios e a segunda possui 32 neurônios. Ambas utilizam a função de ativação ReLU, com regularização $L2$ de 0,05 e camadas de *Dropout*, aplicadas após cada camada densa.

A camada de saída consiste em um único neurônio com função de ativação sigmoide, que produz um valor entre 0 e 1, representando a probabilidade de pertencer à classe positiva na classificação binária.

O modelo é compilado utilizando o otimizador Adam, com uma taxa de aprendizado de 0,0001. A função de perda utilizada é a *binary_crossentropy*, apropriada para problemas de classificação binária. A acurácia é utilizada como métrica de avaliação do desempenho do modelo.

Em suma, a arquitetura da CNN descrita neste trabalho foi projetada para extrair características relevantes das imagens de entrada e realizar a classificação binária. Revelou-se importante a utilização de técnicas como normalização em lote, *Dropout* e regularização $L2$ para gerar uma robustez no modelo e auxiliar na prevenção de sobreajuste. A escolha do otimizador Adam e da função de perda são consistentes com as práticas recomendadas para problemas de classificação binária.

4 Apresentação e análise dos resultados

Neste capítulo, serão apresentados os resultados obtidos na tentativa de reprodução de alguns dos resultados da CNN proposta por (7), depositado no GitHub (23). Nos ateremos à capacidade de generalização do modelo, acurácia do modelo, matriz de confusão, ROC e exemplos de imagens classificadas usando a métrica da matriz de confusão.

Os códigos utilizados para a obtenção desses resultados estão disponíveis em: (24)

4.1 Capacidade de generalização do modelo e acurácia do modelo

O gráfico da figura 16 abaixo mostra a precisão e a perda durante o aprendizado da nossa CNN. O eixo x representa as épocas de treinamento, enquanto o eixo y esquerdo representa a precisão do modelo, em que as linhas azuis representam a precisão e a perda do conjunto de treinamento, enquanto as linhas laranja representam a precisão e a perda do conjunto de validação.

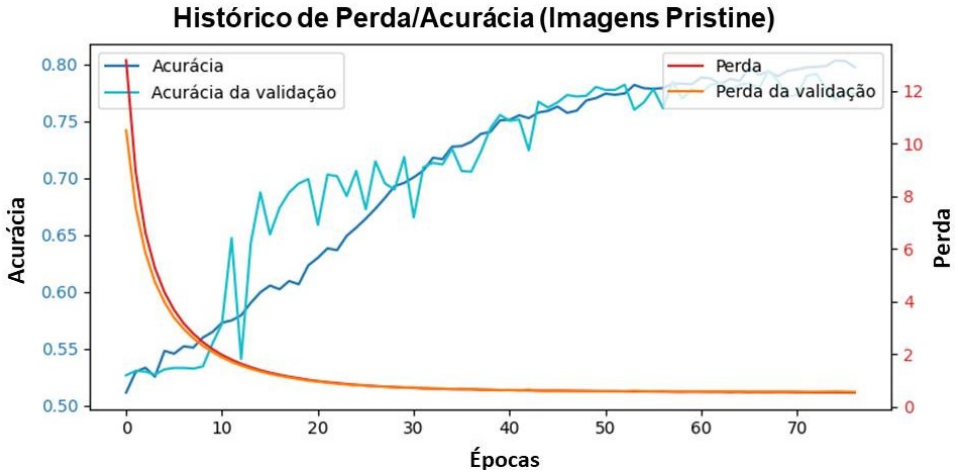
O gráfico de perda/acurácia do modelo treinado com imagens *pristine* (sem ruído), onde tanto a acurácia de treinamento quanto a de validação aumentam ao longo das épocas, chegando a aproximadamente 0,77 e 0,75, respectivamente. Concomitantemente, as perdas de treinamento e validação diminuem, estabilizando-se em torno de 0,5 e 1,0, respectivamente. A pequena diferença entre as métricas de treinamento e validação sugere uma boa generalização do modelo, com indícios de sobrejuste pequeno, indicando um treinamento eficaz. No final, a acurácia do modelo carregado foi de 79.39%.

A acurácia dos dados de teste foi de $\approx$ 79,39% e a perda de $\approx$ 0,56, que representam resultados consistentes com o desempenho observado no conjunto de validação durante o treinamento, que confirma a boa capacidade de generalização do modelo para dados não vistos. Esse resultado é semelhante ao obtido por (7), que foi de 79,25%.

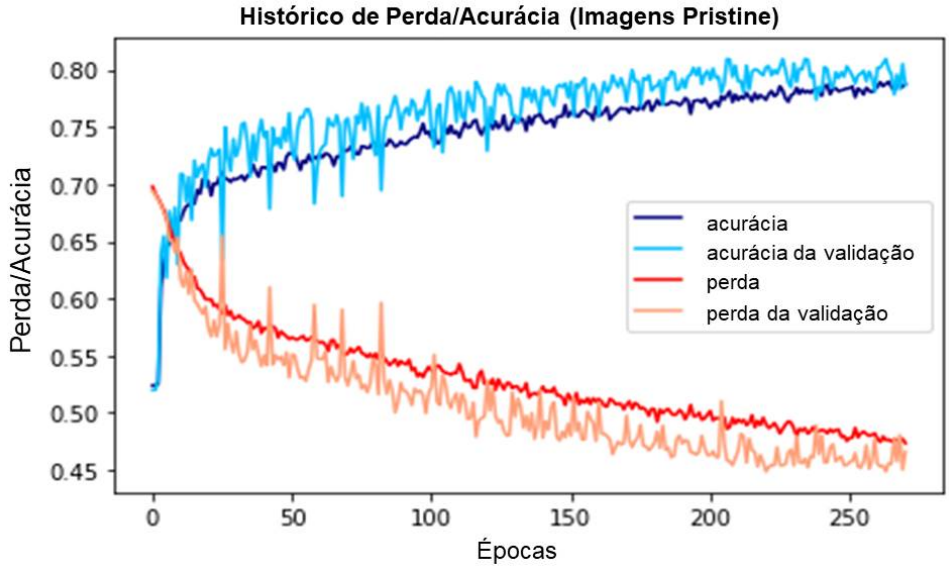
4.2 Análise da matriz de confusão

Para observar como os dados de teste foram classificados pelo modelo, utilizamos as matrizes de confusão mostradas na Figura 17, em que o eixo Y representa os valores das classes reais das instâncias e o eixo X as classes previstas pelo classificador.

A análise da matriz de confusão normalizada se deu da seguinte forma: para a classe 0, o modelo apresentou uma taxa de acerto de 80% de VN e uma taxa de erro de 20% de FP; já para a classe 1, o modelo obteve uma taxa de acerto de 79% de VP e uma taxa de erro de 21% de FN.



(a)



(b)

Figura 16 – Gráficos do treinamento dos modelos: (a) Nosso modelo, (b) Modelo de (7)

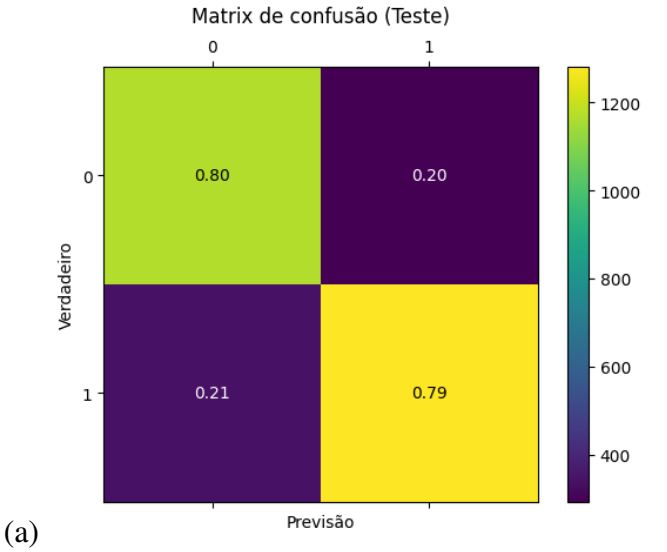
Esse resultado se assemelha ao obtido por (7), mas apresenta um pequeno desbalanceamento entre classes, gerado por uma maior presença de 0 (não fusões) do que de 1 (fusões).

Os demais parâmetros extraídos dessa análise estão organizados na Tabela 1, abaixo, que demonstra uma proximidade entre os nossos resultados e aqueles da literatura (7).

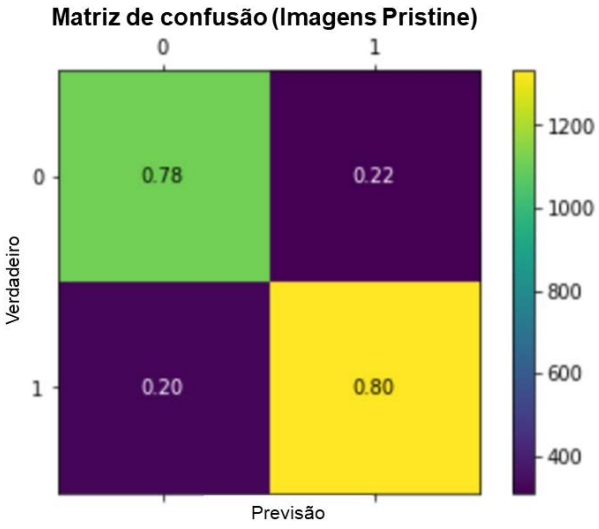
Tabela 1 – Comparação de desempenho entre o nosso modelo e o modelo de (7).

Parâmetros	**Nosso modelo**	**Modelo de** (7)
Acurácia	0.793908	0,792545
Precisão	0.813850	0,810597
Recall	0.788793	0,801807
F1 score	0.801126	0,806178
Brier score	0.147722	0,148574

Logo, analisando esses parâmetros, fica mais claro os pontos de sobreajuste do modelo que devem ser melhorados em trabalhos futuros.



(a)



(b)

Figura 17 – Matriz de confusão aplicada ao conjunto de teste: (a) Nossos resultados (b) resultados de (7)

4.3 Curva ROC

A curva ROC do modelo, figura 18 abaixo, indica um desempenho superior a um classificador aleatório, demonstrando sua capacidade de discriminar entre as classes. A ascensão inicial rápida da curva sugere a efetividade na captura de verdadeiros positivos, com poucos falsos positivos em limiares mais altos. A proximidade da curva ao canto superior esquerdo do gráfico reforça o bom desempenho geral do modelo.

Nossa AUC foi ≈ 0.8711, o que representa a atribuição de uma probabilidade 87,11% a amostras verdadeiramente positivas. Esse resultado é ligeiramente maior que a AUC de 0,8647 de (7).

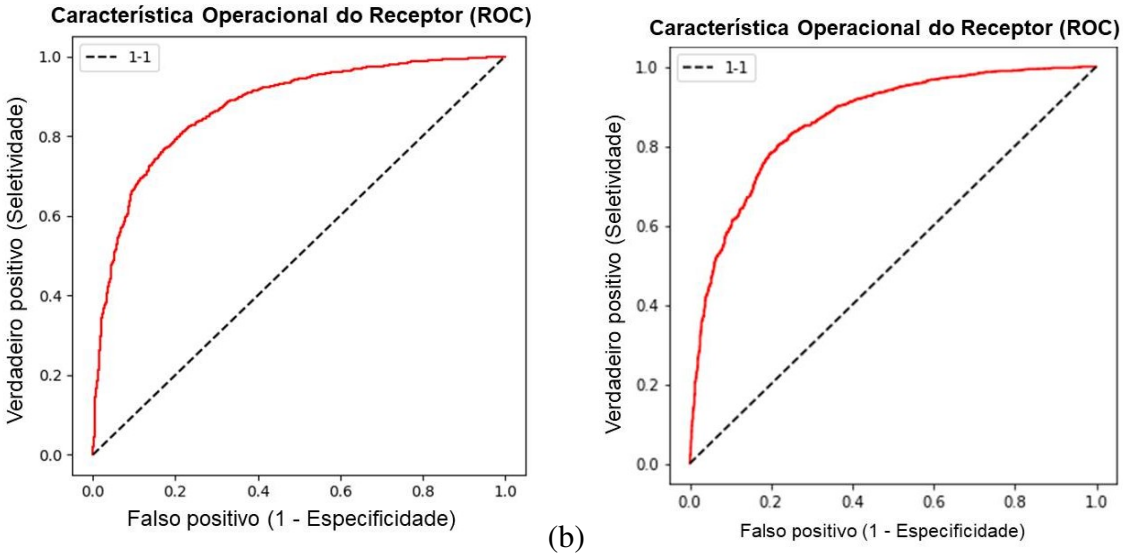


Figura 18 – Curva ROC: (a) nosso modelo, (b) modelo de (7)

4.4 Classificação e análise das galáxias classificadas

Antes de visualizar as imagens das galáxias e observar se elas correspondem com as características descritas na seção 2.2, selecionamos a *random.seed(2024)*, da qual extraímos os resultados da Tabela 2 abaixo.

Classificação	Número de imagens	Porcentagem do conjunto de teste
FN	293	9,5%
FP	343	11,1%
VN	1169	37,9%
TP	1281	41,5%

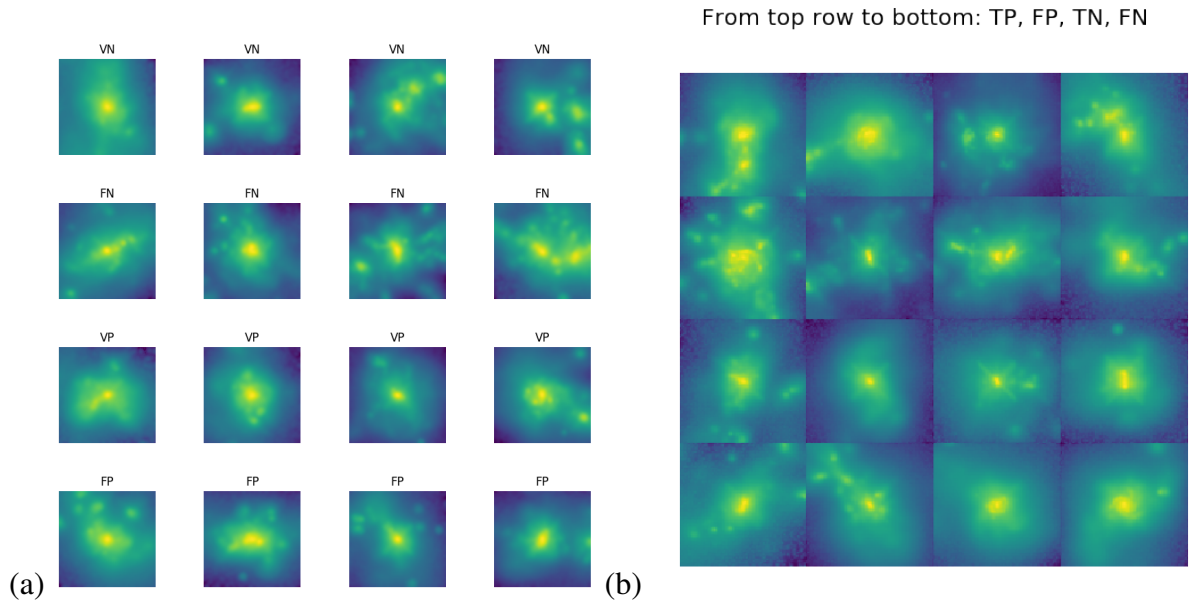
Tabela 2 – Exemplo da distribuição de classificação do nosso modelo para uma semente randômica

Após isso, extraímos o conjunto de imagens mostrado na figura 19, que possui a mesma quantidade de imagens e o mesmo mapa de cores utilizado por (7).

A escolha desse mapa de cores se deu pois sua gama de cores varia de forma uniforme do azul-esverdeado ao amarelo-esverdeado, através do mapeamento dos valores de dados, de forma que diferenças numéricas correspondam a diferenças visuais proporcionais, facilitando a interpretação de gradientes e padrões.

Para realizarmos uma análise mais concisa e detalhada, extraímos uma imagem de cada categoria, formando a figura 20 abaixo.

- **Imagem classificada como VN:** há um núcleo central bem definido, com morfologia regular e distribuição homogênea de estrelas. Apesar disso, há um ruído em todo, que o classificador não atribuiu uma leitura incorreta. Logo, foi corretamente classificada como não fusão.



(a) (b)

Figura 19 – Imagens extraídas pós-classificação em que foi aplicado o mapa de cores *Viridis*: (a) nosso resultado, (b) resultado de (7)

- **Imagem classificada como FN:** apesar de possuir uma simetria visual razoável, parece haver indícios do desaparecimento da cauda de maré; há também uma concentração de pontos amarelos no núcleo, com forma alongada, que podem indicar aumento de formação estelar. Logo, faz sentido a forma com que foi classificada, pois suas características podem confundir o sistema de classificação com uma provável fusão.
- **Imagem classificada como VP:** há a presença de distorção e irregularidade no formato da galáxia; há uma concentração de pontos amarelos intensos no núcleo, que indica uma intensa formação de estrelas. O conjunto indica que a galáxia está no final do processo de fusão, em que começa a se reorganizar. Logo, está corretamente classificada como Fusão.
- **Imagem classificada como FP:** há a presença de vários núcleos próximos, de amarelo intenso, mas sem os demais marcadores de uma fusão de galáxias. Logo, a distorção da imagem é tanta, que ela se torna inconclusiva.

Ao compararmos nosso modelo com o de Ćiprijanović *et al.* (2023) (7), que nos serviu como referência, evidenciou que tivemos uma relativa melhoria, apesar do sobreajuste evidente, apresentando resultados similares ou até mesmo superiores em algumas métricas, apesar das diferenças metodológicas e de arquitetura.

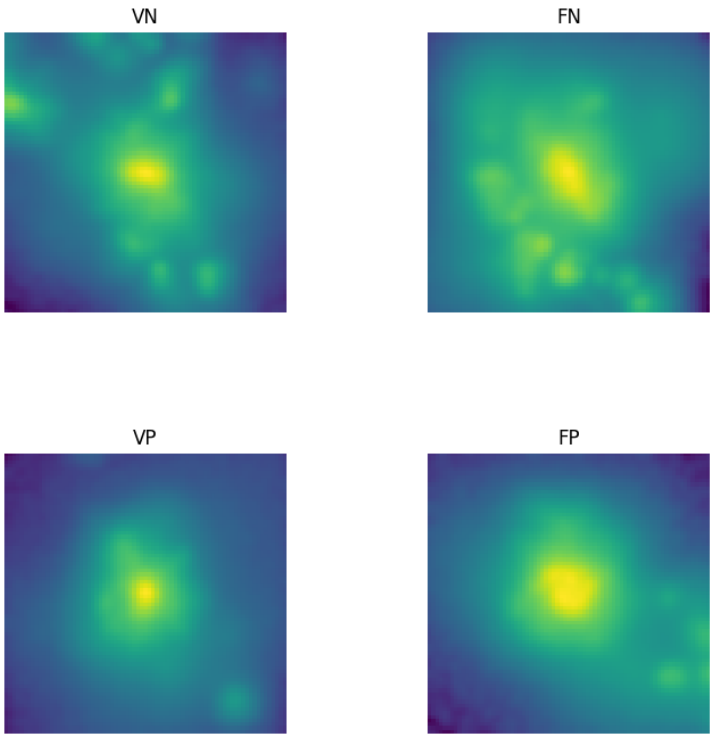


Figura 20 – Imagens extraídas para uma análise de fusão ou não fusão

5 Conclusões

Este trabalho explorou a aplicação de Redes Neurais Convolucionais (CNNs) na classificação de fusões de galáxias simuladas pelo Projeto Illustris. A partir da análise dos resultados obtidos, constatou-se a viabilidade e o potencial dessa abordagem para automatizar a análise, filtragem e categorização de grandes volumes de dados astrofísicos.

A escolha das CNNs como método de aprendizado de máquina se deu devido à extensa literatura que aborda sua aplicação para o reconhecimento de padrões em imagens. Essa robustez, para tal tipo de aplicação, se dá devido à arquitetura da rede neural, que é composta por camadas convolucionais, de agrupamento e totalmente conectadas. Essa composição permite a extração de características relevantes das imagens das galáxias, desde detalhes de rastros de poeira estelar até núcleos em processo de fusão espalhados pela região de alta interação gravitacional, o que contribui de forma significativa para a performance do modelo.

Alcançamos a acurácia de aproximadamente 79%, em conjunto com uma AUC de $\approx 0{,}87$, torna-se um pouco mais evidente a capacidade do modelo em aprender padrões complexos presentes no conjunto de imagens de galáxias utilizados e realizar a classificação com um desempenho promissor, apesar de um leve sobreajuste que prejudica a obtenção de um resultado melhor.

Esse sobrejauste é multifatorial, mas uma das causas foi termos descartado temporariamente um novo aumento do conjunto de dados, pois esses foram previamente extraídos de um catálogo, aumentados e organizados em um arquivo *.fits*, pelos autores do trabalho em que nos baseamos. Realizar um novo aumento interferiria na replicação dos resultados e nos tomaria mais tempo do que dispúnhamos.

A análise da matriz de confusão dos valores normalizados, forneceu pistas sobre o comportamento do modelo, apontando um certo desbalanceamento dos dados, devido a uma melhor classificação de 0 do que 1. Nesse sentido, a inspeção visual de casos individuais de classificações incorretas revelou alguns desafios, a começar pela baixa resolução das imagens, que gera um problema na classificação de galáxias com núcleo de luminosidade intensa, com presença de duas ou mais galáxias na imagem, e com a sobreposição de poeira estelar em áreas de galáxias de morfologia preservada.

Apesar dos resultados promissores, este trabalho representa apenas um passo inicial na exploração do potencial das CNNs para a classificação morfológica de um número maior de galáxias ou de outras estruturas astronômicas. Para trabalhos posteriores e complementares, há diversas oportunidades de melhoria e expansão que se apresentam para pesquisas futuras, como: o aumento de robustez do modelo, a utilização de um conjunto de dados maior e a incorporação de dados espectroscópicos ou medidas fotométricas. Esses acréscimos são capazes de enriquecer

o processo de aprendizado e aprimorar a precisão do modelo, o que permitirá, futuramente, avaliar cenários mais desafiadores e aproximar a sua utilização em aplicações práticas.

Em um contexto mais amplo, este trabalho se insere na crescente tendência de utilização de técnicas de aprendizado de máquina em astronomia e astrofísica. Devido à sua capacidade de analisar grandes volumes de dados, identificar padrões complexos e automatizar tarefas de classificação, tem o potencial de revolucionar a forma como a pesquisa científica é conduzida nessas áreas, reduzindo o tempo de análise e validação humana, possibilitando uma maior velocidade na depuração de dados e de testes comparativos entre teoria e dados reais.

A integração entre a experiência humana e a capacidade computacional das máquinas se configura como um caminho promissor para desvendar os segredos do universo e expandir as fronteiras da astronomia moderna. Portanto, o futuro da astronomia e da cosmologia estará intrinsecamente ligado ao desenvolvimento e à aplicação dessas tecnologias, abrindo caminho para uma nova era de descobertas e avanços científicos.

Referências

1 NELSON, D. et al. The illustris simulation: Public data release. *Astronomy and Computing*, v. 13, p. 12–37, nov. 2015.

2 SANTOS, V. S. dos. *Cérebro*. 2023. <https://mundoeducacao.uol.com.br/biologia/cerebro.htm>. Acessado: 27-09-2023.

3 BORGES, R. et al. Sincronização de disparos em redes neuronais com plasticidade sináptica. *Revista Brasileira de Ensino de Física*, v. 37, n. 2, p. 2310–1–2310–9, 2015. Disponível em: <https://www.researchgate.net/publication/282962512_Sincronizacao_de_disparos_em_redes_neuronais_com_plasticidade_sinaptica>.

4 ALVES, D. R. d. S.; SILVA, C. M. d. O.; OLIVEIRA, R. F. d. Redes neurais artificiais e suas aplicações em bioengenharia: uma revisão. *Revista Brasileira de Engenharia Biomédica*, Associação Brasileira de Engenharia Biomédica, v. 24, n. 2, p. 111–123, 2008. Disponível em: <https://www.maxwell.vrac.puc-rio.br/37156/37156_5.PDF>.

5 GRADIENT Descent in Machine Learning. 2023. <https://www.javatpoint.com/gradient-descent-in-machine-learning>. Acessado: 27-09-2023.

6 MOREIRA, G. *Redes Neurais Artificiais: Fundamentos e Aplicações*. [S.l.]: Escola de Engenharia da Universidade Federal de Minas Gerais, 2006. <https://www.est.ufmg.br/portal/wp-content/uploads/2023/01/96-GuilhermeMoreira_2006.pdf>. Dissertação de Mestrado.

7 ĆIPRIJANOVIĆ, A. et al. DeepMerge: Classifying high-redshift merging galaxies with deep neural networks. *Monthly Notices of the Royal Astronomical Society*, v. 522, n. 1, p. 1153–1175, 2020.

8 FAWCETT, T. An introduction to roc analysis. *Pattern recognition letters*, Elsevier, v. 27, n. 8, p. 861–874, 2006.

9 ILYUREK, I. Roc curve and auc: Evaluating model performance. *Medium*, 2019. Acessado: 2023-10-26. Disponível em: <https://medium.com/@ilyurek/roc-curve-and-auc-evaluating-model-performance-c2178008b02>.

10 ROWDEN, P. *Citizen Scientists Re-Tune Hubble's Galaxy Classification*. Royal Astronomical Society, 2019. <https://ras.ac.uk/news-and-press/research-highlights/citizen-scientists-re-tune-hubbles-galaxy-classification>. Acessado 2023-10-26. Disponível em: <https://ras.ac.uk/news-and-press/research-highlights/citizen-scientists-re-tune-hubbles-galaxy-classification>.

11 BRIGGS, A. *What is a Galaxy?* EarthSky, 2023. <https://earthsky.org/astronomy-essentials/definition-what-is-a-galaxy/>. Acessado 2023-10-26. Disponível em: <https://www.youtube.com/@spaceodditieslive>.

12 ANDRADE, M. E. James webb: Fusão de duas galáxias é registrada por telescópio. *O Povo*, oct 2022. Acessado 2023-10-27. Disponível em: <https://www.opovo.com.br/noticias/curiosidades/2022/10/25/james-webb-fusao-de-duas-galaxias-e-registrada-por-telescopio.html>.

13 BABINI, M.; MARRANGHELLO, N. *Introdução às redes neurais artificiais*. São Paulo: Cultura Acadêmica, 2007. 61p. : il. ; v. 2) p. São José do Rio Preto, SP: Laboratório Editorial do IBILCE, UNESP. ISBN 978-85-98605-21-0.

14 HAYKIN, S. S. *Neural networks : a comprehensive foundation*. Upper Saddle River, N.J.: Prentice Hall, 1999.

15 FACURE, M. *Funções de ativação em redes neurais*. 2017. Acessado: 27-09-2023. Disponível em: <https://matheusfacure.github.io/2017/07/12/activ-func/>.

16 KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1609.04747*, 2016.

17 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

18 SAHA, S. A comprehensive guide to convolutional neural networks: The eli5 way. *Towards Data Science*, 2018. Acessado: 27-09-2023.

19 STUTZ, D. *Seminar: A Robust Statistical Approach to Deep Learning*. 2014. <https://davidstutz.de/wordpress/wp-content/uploads/2014/07/seminar.pdf>. Acessado: 2023-10-27.

20 YAMASHITA, R. et al. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, Springer, v. 9, n. 4, p. 611–629, 2018.

21 INSTITUTE, S. T. S. *Deep Merge*. [S.l.]: HLSP, 2023. <https://archive.stsci.edu/hlsp/deepmerge>. Acessado: 27-09-2023.

22 LEE, B. et al. Candels: The correlation between galaxy morphology and star formation activity at $z\tilde{2}$. *Astrophysical Journal*, 2013. To appear in Astrophysical Journal. Disponível em: <https://esahubble.org/static/archives/releases/science_papers/heic1315a.pdf>.

23 ĆIPRIJANOVIĆ, A. *Deepmerge-public*. [S.l.]: GitHub, 2023. <https://github.com/AleksCipri/deepmerge-public>.

24 ALMEIDA, N. C. M. *CNN para classificar fusões de galáxias*. 2024. <https://github.com/NataliCMAlmeida/CNN-para-classificar-fus-o-de-galaxias>.