Universidade Federal de Ouro Preto
Escola de Minas
CECAU - Colegiado do Curso de
Engenharia de Controle e Automação

Ramon do Rosário de Lima

Aplicação de Redes Neurais Recorrentes na Predição de Relações Estrutura–Atividade Química com Múltiplos Alvos Biológicos

Monografia de Graduação em Engenharia de Controle e Automação

Ouro Preto, 2024

Ramon do Rosário de Lima

Aplicação de Redes Neurais Recorrentes na Predição de Relações Estrutura–Atividade Química com Múltiplos Alvos Biológicos

Trabalho apresentado ao Colegiado do Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Universidade Federal de Ouro Preto

Orientador: Prof. Dr. Jadson Castro Gertrudes

Coorientador: Prof. Agnaldo Jose da Rocha Reis

Ouro Preto

2024

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E AUTOMACAO

FOLHA DE APROVAÇÃO

Ramon do Rosário de Lima

Aplicação de Redes Neurais Recorrentes na Predição de Relações Estrutura–Atividade Química com Múltiplos Alvos Biológicos

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 17 de outubro de 2024

Membros da banca

[Doutor] - Jadson Castro Gertrudes - Orientador (Universidade Federal de Ouro Preto)
[Doutor] - Agnaldo José da Rocha Reis - (Universidade Federal de Ouro Preto)
[Mestre] - Hugo Eduardo Ziviani - (ATECH, Brasil)
[Graduado, Mestrando] - Marcos Felipe Pontes Rezende - (Universidade Federal de Ouro Preto)

Jadson Castro Gertrudes, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 24/10/2024

Documento assinado eletronicamente por **Jadson Castro Gertrudes**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 25/10/2024, às 10:01, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0800266** e o código CRC **38B3B044**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591533 - www.ufop.br

Resumo

O presente estudo abordou a aplicação de Redes Neurais Recorrentes (LSTM e BILSTM) para estabelecer relações entre estruturas químicas e atividades biológicas. O projeto visou replicar experimentos anteriores, desenvolver um modelo para detecção dessas relações. A segunda etapa do trabalho envolveu o uso do banco de dados TOX21, concentrando-se em amostras que evidenciavam reações positivas, e aplicando a técnica SMILES para representação molecular. Os experimentos demonstraram que aumentar a capacidade da rede neural, como o número de células e a combinação de camadas, não resultou em melhorias significativas no aprendizado, sugerindo que a dimensão do modelo não é a principal causa do overfitting observado. Uma hipótese levantada foi a qualidade e o volume dos dados, considerando a baixa proporção de amostras de sucesso. A pesquisa propôs a utilização de métodos mais avançados de tratamento de dados para alcançar um balanceamento adequado entre as classes de cada alvo biológico.

Palavras-chaves: Química Medicinal.Smiles. Predição. Múltiplos Alvos. Redes Neurais artificiais. Long Short-term Memory.

Abstract

This study looked at Recurrent Neural Networks (LSTM and BILSTM) applications to establish relationships between chemical structures and biological activities. The project aimed to replicate previous experiments and develop a model for detecting these relationships. The second stage of the work involved using the TOX21 database, focusing on samples that showed positive reactions, and applying the SMILES technique for molecular representation. The experiments showed that increasing the neural network's capacity, such as the number of cells and the combination of layers, did not significantly improve learning, suggesting that the model's size is not the leading cause of the observed overfitting. One hypothesis raised was the quality and volume of the data, considering the low proportion of successful samples. The research proposed using more advanced data processing methods to achieve an adequate balance between the classes of each biological target.

Key-words: Medicinal Chemistry. SMILES. Prediction. Multiple Targets. Artificial Neural Networks. Long Short-Term Memory.

Lista de ilustrações

Lista de abreviaturas e siglas

IA	Inteligência Artificial
RNA	Redes Neurais Artificiais
DP	Deep Learning
LSTM	Long Short Term Memory
BILSTM	Long Short Term Memory Bidirecional
SAR	Structure-Activity Relationship.
FFNN	Feed Forward Neural Network
MLP	Multi-Layer Perceptron
SMILES	Simplified Molecular Input Line Entry System
NLTK	Natural Language Toolkit

Sumário

1 Introdução

Redes neurais artificiais são um tipo de modelo computacional inspirado no funcionamento do cérebro humano, isto é, são capazes, a partir de um tratamento, uma interpretação e um treinamento sobre um conjunto de dados, fazer previsões ou classificações a respeito de algum assunto de interesse. Com o desenvolvimento de técnicas de aprendizado profundo nos últimos anos, as redes neurais passaram a ser capazes de realizar tarefas cada vez mais complexas, como reconhecimento de fala, reconhecimento de imagens e tradução automática de idiomas (FACELI et al., 2011).

A inteligência artificial, hoje, também está aplicada nas pesquisas na área de Química Medicinal com o objetivo de desenvolver e descobrir novos compostos químicos que possam ser utilizados como medicamentos. As técnicas de aprendizado profundo e mineração de dados otimizam esse processo por meio da redução de custos e de tempo (YOUNG, 2009). Sob essa perspectiva, diferente da descoberta da penicilina, um evento considerado um resultado do acaso, os algoritmos computacionais aplicados à pesquisa medicinal garantem maior previsibilidade (SILVA, 2023).

Nessa área de pesquisa, a análise das relações entre as estruturas químicas e o alvo biológico [1] - Relação Estrutura - Atividade (SAR) - e a triagem virtual de moléculas (TROSSINI; MALTAROLLO; SCHMIDT, 2014) são os dois métodos principais para a busca de uma molécula candidata a um fármaco, uma vez que o alvo biológico já foi identificado. Para aquela primeira forma de busca, utiliza-se um banco de dados de moléculas em que há informações sobre a sua interação com outras estruturas químicas e, a partir disso, aplica-se métodos para estabelecer as relações SAR, basicamente, modelos que estabelecem o grau de afinidade entre as formas químicas. Já o modelo de triagem virtual, consiste em uma técnica que faz uso de simulação computacional. Ocorre que, previamente, é feita uma seleção de moléculas candidatas a fármacos para fazer simulações, de modo a medir a afinidade com outros arranjos químicos antes de realizar testes em laboratório, o que reduz o número de experimentos a serem realizados.

A contribuição da técnica, que relaciona a estrutura e a atividade química, é de extrema importância para a pesquisa e o desenvolvimento de novas substâncias com potencial farmacológico (SERAFIM et al., 2021). Nesse contexto, pode-se, a partir de modelos SAR, promover antecipações sobre as propriedades biológicas dos compostos antes de sua síntese (PINGAEW et al., 2022). Bem como, identificar em meio a um conjunto de substâncias, aquelas que apresentam características promissoras, com alto grau de sucesso para o desenvolvimento de algum inibidor (DEMBITSKY et al., 2022). Dessa forma, a confia-

[1] São definidos como moléculas, macromoléculas ou estruturas dentro de um organismo que interagem com substâncias bioativas, como medicamentos, para produzir uma resposta fisiológica ou terapêutica.

bilidade, a análise e o tratamento de dados são imperativos para garantir a qualidade dos resultados (TROPSHA, 2010), pois podem resultar em cálculos incorretos e, por conseguinte, projetar modelos incorretos. De acordo com (TROPSHA, 2010), em média, há dois erros estruturais por cada publicação de Química Medicinal, com uma taxa de erro em torno de 8% para dados indexados no banco de dados *WOMBAT* (SCHREIBER; KAPOOR; WESS, 2007).

Visto que a análise SAR é menos custosa financeiramente e computacionalmente, com relação a triagem virtual, torna-se mais atrativo fazer o uso de redes neurais para propor uma abordagem a essa técnica. Afinal, por meio do aprendizado profundo, alguma arquitetura de rede recorrente, multicamada e com retropropagação ou convolucionais, entre outras. Além disso, essas arquiteturas de redes neurais, podem ser treinadas e validadas por alguma métrica, como a acurácia, para identificar padrões e prever atividades biológicas de novos compostos.

Assim, a aplicação de redes neurais em Química Medicinal pode contribuir significativamente para a descoberta e o desenvolvimento de novos fármacos, pois tornam o processo mais eficiente e rápido, já que fornecem a capacidade de fazer predições, testes e identificações sob um determinado parâmetro de confiança (YOUNG, 2009; VERÍSSIMO et al., 2022). Como resultado, permite uma exploração aumentada no vasto espaço de potenciais farmacológicos, visto que (POLISHCHUK; MADZHIDOV; VARNEK, 2013) identificaram existir, de forma estimada, de 10^{23} a 10^{60} moléculas potencialmente sintetizáveis.

1.1 Justificativas e Relevância

A Química Medicinal é uma área de pesquisa que se concentra na descoberta e desenvolvimento de novos compostos com atividade farmacológica. Uma das principais ferramentas utilizadas pelos pesquisadores para entender a estrutura-atividade dos compostos é a análise SAR. Esta técnica envolve a correlação entre as propriedades químicas dos compostos e a atividade biológica que eles exibem.

A análise SAR tradicional é um processo intensivo em trabalho e consome muito tempo, com isso, pode-se levar anos para analisar completamente um conjunto de compostos. É aqui que as redes neurais podem oferecer uma solução eficaz para acelerar esse processo em função da possibilidade de processar informações a fim de gerar modelos lógicos que forneçam resultados com certo grau de confiabilidade em intervalo de tempo menor do que a técnica tradicional.

1.2 Objetivos

Promover o uso de técnicas de aprendizado profundo, em especial a arquitetura de rede neural LSTM, para a análise SAR. Os objetivos específicos deste projeto são:

- Analisar, por meio de revisão, a existência de outras arquiteturas de redes neurais recorrentes aplicadas na análise SAR;
- Investigar e adaptar arquiteturas LSTM para o cenário estudado, que consiste em promover uma classificação da atividade biológica dos compostos químicos para múltiplos alvos.

2 Revisão Bibliográfica

2.1 Análise de relações entre estruturas químicas e atividade biológica

A análise de relações estrutura-atividade (SAR) é uma abordagem empregada para identificar e prever a atividade biológica de compostos químicos com base em suas estruturas moleculares. Esta técnica é fundamental na descoberta de novos fármacos, permitindo a otimização de compostos com propriedades desejáveis. Neste contexto, a aplicação de técnicas de aprendizado de máquina e redes neurais tem se mostrado promissora para aumentar a eficiência e precisão dessas análises. Um trabalho que envolve a técnica de análise SAR foi desenvolvido por (VERÍSSIMO et al., 2022). Neste projeto, os autores utilizaram a arquitetura KGCN, uma rede convolucional baseada em grafos, implementada para produzir modelos de classificação a fim de selecionar potenciais inibidores de *S. aureus FabI*, cujos resultados foram expressivos.

Com o uso da arquitetura de rede LSTM, um tipo de rede neural recorrente especializada em lidar com sequências de dados, (CHAKRAVARTI; ALLA, 2019) usa a representação da molécula para realizar a análise SAR em um conjunto de dados de três terminais: mutagenicidade de Ames, inibição de *P. falciparum Dd2* e inibição de vírus da Hepatite C, com conjuntos de treinamento variando de 7.866 a 31.919 compostos. Para isso, utilizaram a técnica SMILES[1], uma forma de representar as estruturas químicas em forma de texto, por meio de símbolos e códigos específicos, gerando resultados satisfatórios e com maior grau de transparência. Por meio dos resultados experimentais obtidos pela validação cruzada, a LSTM mostrou um desempenho expressivo, pois, no processo de teste, com dados diferentes do conjunto de treinamento, obteve previsões com elevada eficiência em relação aos modelos tradicionais de modelagem SAR. Concluíram, portanto, que é possível construir modelos SAR utilizando LSTMs sem a necessidade de dados pré-computados. Como consequência, abre-se a perspectiva de construir SARs de boa qualidade para um grande conjunto de dados, sem antes usar algum descritor pré-calculado.

Fooshee et al. (2018) desenvolveram um trabalho em que usam as redes recorrentes, LSTM, para fazer a predição de reações químicas. Ou seja, o autor utilizou o aprendizado profundo para prever e classificar reações elementares por meio da identificação de fontes e consumidores de elétrons. Foi proposto um modelo de solução para um problema da química sintética: a identificação de produtos desconhecidos observados por espectrometria de massa. Para isso, o conjunto de dados, assim como outros autores já citados fizeram,

[1] Sistema de Entrada de Linha de Codificação Simplificada de Moléculas.

foram representados utilizando a técnica SMILES e testaram esses dados na arquitetura LSTM e MLP[2]. Como resultado, a primeira foi mais eficiente, em função da sua capacidade de considerar todo o contexto dos reagentes durante a previsão, enquanto o modelo MLP foi menos preciso, pois não considera informações contextuais para realizar a predição. Assim, os autores concluíram que o primeiro preditor pode ser considerado uma ferramenta poderosa, pois, sob um conjunto de dados de treinamento expandido, produziu avanços significativos em velocidade e precisão preditiva.

Outro trabalho que relaciona o uso de redes neurais e aprendizado profundo para determinar as relações entre compostos químicos foi desenvolvido por (RAMOS, 2022). O autor tinha como objetivo estabelecer um modelo de aprendizado profundo capaz de avaliar a toxicidade de compostos químicos em aves. Nesse contexto, foi desenvolvido um conjunto de dados de compostos com informações relativas às propriedades toxicológicas experimentais para um determinado conjunto de espécies de aves. Ao final, modelos *multitasks* baseados em redes neurais, do tipo FFNN, foram capazes de predizer a toxicidade aguda de agrotóxicos em aves. Por fim, o autor encontrou resultados significativos, já que obteve-se valores de Coeficiente de Correlação de Pearson (**r**) entre 0,59–0,80 para predizer a toxicidade aguda de diversas espécies pDL50.[3]

2.2 SMILES

No estudo da química, é de fundamental importância estabelecer as regras corretas para identificação e caracterização dos compostos químicos, pois a partir dessas fórmulas é possível conhecer as propriedades de uma molécula. Nesse contexto, existem as fórmulas moleculares - que indicam o número real de átomos de uma molécula - e as fórmulas estruturais - desenhos em perspectiva para representar a forma tridimensional de uma molécula (THEODORE L BROWN H. EUGENE LEMAY, 2005).

A organização dos caracteres que representam cada elemento de uma substância em uma fórmula molecular obedece a uma regra lógica, de maneira que seja compreendida universalmente. A organização estrutural não é diferente; os caracteres e as formas geométricas que representam as ligações químicas, assim como a sua disposição no espaço, são uma referência normativa nessa área do conhecimento (PINGARRÓN et al., 2020). Para tanto, a IUPAC[4], é responsável por promover esse intercâmbio e a manutenção dessas informações. Sob essa perspectiva, destaca-se que existem regras diferentes para estabelecer um padrão que representa uma substância inorgânica e uma substância orgânica - química dos carbonos - para o desenvolvimento de drogas com potencial farmacológico,

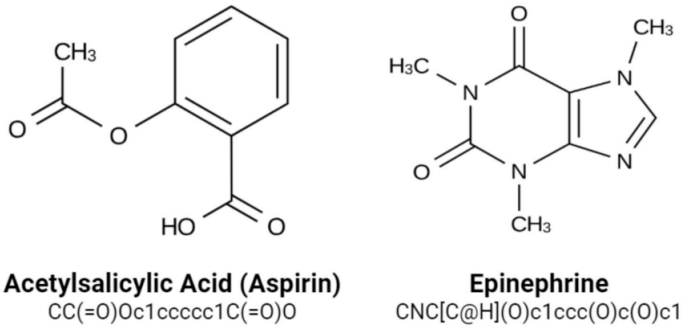
[2] Arquitetura de rede que propaga as informações em uma direção
[3] Escala logarítmica para a dose letal que induz a morte de 50% de uma população de animais.
[4] Órgão Internacional de Química Pura e Aplicada

é imperativo utilizar essas representações, pois a síntese de uma molécula errada pode gerar efeitos colaterais e prejuízos financeiros (RAJAN; ZIELESNY; STEINBECK, 2021).

Para desenvolver algoritmos de *machine learning* que trabalhem com moléculas químicas, a representação SMILES surge como uma facilitadora, representando um composto por uma série de caracteres alfanuméricos (SILVA, 2023) (RAJAN; ZIELESNY; STEINBECK, 2021). Existem algumas regras para fazer a transcrição para a notação SMILES, por exemplo, utiliza-se a notação da tabela periódica para descrever cada átomo, a ordenação dos átomos acontece de acordo com sua conectividade com outros átomos na molécula, o direcionamento das ramificações para os átomos com menor rótulo e, em fechamento de anéis e estruturas cíclicas, evita-se selecionar uma ligação múltipla para o fechamento (WEININGER; WEININGER; WEININGER, 1989).

É importante destacar alguns símbolos utilizados nessa notação: as ligações duplas são apresentadas pelo sinal de "=", já as ligações triplas são representadas pelo caractere "#". Por praticidade, as ligações simples são omitidas. Além disso, os parênteses podem ser utilizados para a identificação de estruturas cíclicas ou de ramificações, dependendo da posição dos átomos dentro dos parênteses e do contexto geral da representação da molécula (WEININGER; WEININGER; WEININGER, 1989; SILVA, 2023). Sob essa perspectiva, a Figura 1, demonstra um exemplo da notação SMILES para as moléculas do ácido acetilsalicílico e da adrenalina, nas quais ocorre a transcrição das cadeias aromáticas, das ligações simples, duplas e ramificações. Visto isso, deve-se observar a validade da SMILES gerada; é possível que uma cadeia esteja correta sintaticamente, mas seja inválida. A exemplo, uma notação para uma cadeia cíclica extensa, em função de restrições estéricas, pode não existir no mundo físico. Alterar a posição de algum caractere também pode tornar a cadeia inválida, assim como a descrição de átomos flutuantes, sem ligações ou com ligações quebradas (SILVA, 2023).

Figura 1 – Exemplo de uma notação SMILES.



Fonte: (YASONIK, 2020).

2.3 Processamento de Texto

2.3.1 *Word Embedding*

A transcrição de um composto químico para sequência de caracteres facilita o tratamento computacional. Após importar esses dados, o próximo passo é transformá-los em uma forma legível para máquinas. O *Token* consiste em segmentos de um texto identificado como unidades significativas, podendo ser palavras, sub palavras ou até mesmo unidades estilizadas (GREFENSTETTE, 1999). A tokenização é o processo de dividir o texto em pedaços menores, remover caracteres indesejados com a finalidade de realizar análises sintáticas e semânticas para alimentar modelos de aprendizagem de máquina (A. MULLEN et al., 2018).

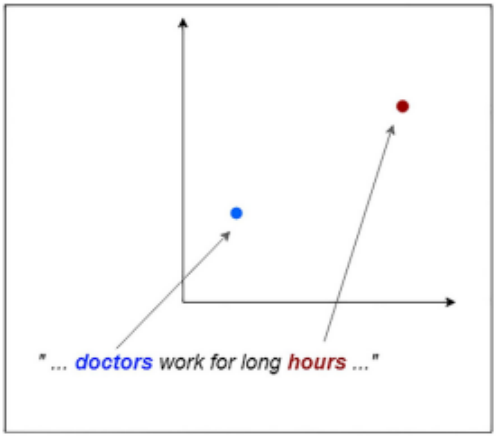
No roteiro para o tratamento de texto, após a fragmentação da cadeia de caracteres conforme as regras exigidas pelo projeto, deve-se quantificar o número de símbolos diferentes, determinar o vocabulário e transcrever cada *token* em um valor numérico de maneira que preserve a semântica (KHATTAK et al., 2019). Neste processo, é necessário garantir o mesmo tamanho das sequências que representam a sentença textual. Para tanto, utiliza-se o *Padding*, o preenchimento, uma maneira de fixar e padronizar as sequências textuais do conjunto de dados. Este preenchimento é realizado conforme o tamanho da maior sequência de texto encontrada no *dataset*, assim, aquela sequência que apresenta menor número de *tokens* é preenchida no início ou no final da sentença com o valor $< PAD >$, sem carga semântica, o que não prejudica a interpretação da sentença (DENG; LIU, 2018).

O *Word Embedding*, portanto, consiste essencialmente em atribuir um valor numérico ao conjunto de caracteres conforme o contexto. Desse modo, cada palavra é mapeada de acordo com o vocabulário, no qual as sentenças em um mesmo cenário possuem significados semelhantes e, portanto, um valor numérico próximo. Assim, essa relação de semelhança ou significado entre os fragmentos de texto, *tokens*, depende inteiramente dos dados, dos quais essa incorporação de valor numérico é resultado (KHATTAK et al., 2019). A fim de exemplificação, a Figura 2, expressa, em um espaço vetorial, a relação semântica das palavras **doctors** e **hours**, as quais não possuem valores próximos conforme o mapeamento realizado nos dados de texto.

2.4 Redes Neurais Artificiais

Em 1943, McCullock e Pitts propuseram o primeiro modelo matemático que representa o comportamento de um neurônio biológico, a partir do qual, iniciou-se as pesquisas e o desenvolvimento de trabalhos relacionados à inteligência artificial e às redes neurais. Com o objetivo de replicar o modelo de comportamento do neurônio biológico, a arquitetura proposta por Mccullock e Pitts envolve, conforme a Figura 3, um conjunto de dados

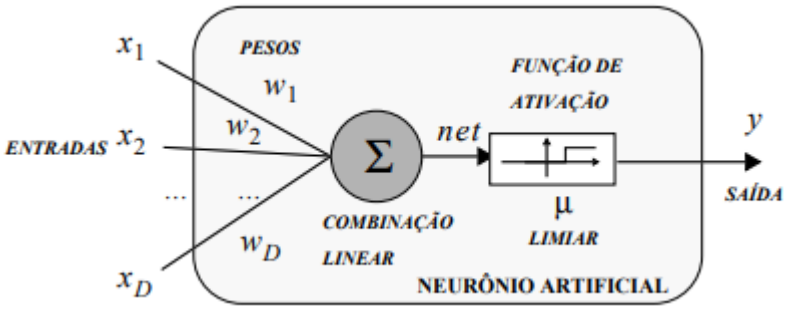
Figura 2 – Espaço vetorial 2D da representação de duas palavras.



Fonte: (KHATTAK et al., 2019).

de entrada (x) multiplicada por um número de pesos (w), os quais, posteriormente, são somados em um nó, cujo o resultado, de acordo com a Equação 2.9, é processado por uma função de ativação, que por fim, gera uma resposta na saída do modelo (LIMA; PINHEIRO; SANTOS, 2014).

Figura 3 – Arquitetura de um neurônio artificial



Fonte: (RAUBER, 2005)

$$net = w_1 x_1 + w_2 x_2 + \cdots + w_D x_D = \sum_{j=1}^{D} w_j x_j - \mu \tag{2.1}$$

Na abstração proposta pela Figura 3 os pesos possuem um papel importante, já que simulam as conexões de um neurônio biológico. Isto é, os valores ajustados para os pesos no processo de aprendizado, podem ser positivos ou negativos. Com isso, a arquitetura, relacionada ao número de unidades de processamento (neurônios) e o aprendizado, vinculado às normas para o ajuste dos pesos, caracterizam um rede neural artificial e determinam o seu próprio aprendizado (FACELI et al., 2011). Este, por sua vez, acontece através de processos iterativos durante o treinamento sob o conjunto de dados. O que

acontece, a partir do momento que a rede neural passa a entregar resultados generalizados para algum problema em análise, significa que a rede extraiu padrões relevantes que contribuem para produção de representações própria do problema, em outra análise, pode-se dizer a rede adquiriu conhecimento sobre o ambiente em que está trabalhando (FLECK et al., 2016).

Além dos pesos, as funções de ativação também são de fundamental importância, uma vez que restringem a amplitude do valor e fornecem uma componente de não linearidade a resposta de um neurônio ou da saída total de uma rede (FLECK et al., 2016). Ocorre que, é necessário a quantificação da influência de cada dado na entrada da unidade de ativação para definir o resultado final, isto é feito, por meio de funções matemáticas como a limiar, Equação 2.2, que restringe a saída da RNA a valores binário (0- negativo e 1- positivo); a função linear (Equação 2.3); a função sigmoidal, assume um valor entre 0 e 1 com um parâmetro a de inclinação conforme Equação 2.4; função tangente hiperbólica, uma variação da função sigmoide cujo o intervalo varia entre (-1,1) de Equação 2.5 (FLECK et al., 2016). Agora recentemente, a função ReLU (ativição linear retificada), que passou ser utilizada com frequência, visto a popularização do *deep learning*, essa função retorna 0 se recebe um valor negativo, caso contrario, retorna o próprio valor de Equação 2.6 (FACELI et al., 2011) (FLECK et al., 2016).

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0, \\ 0 & \text{se } x < 0. \end{cases} \tag{2.2}$$

$$f(x) = x \tag{2.3}$$

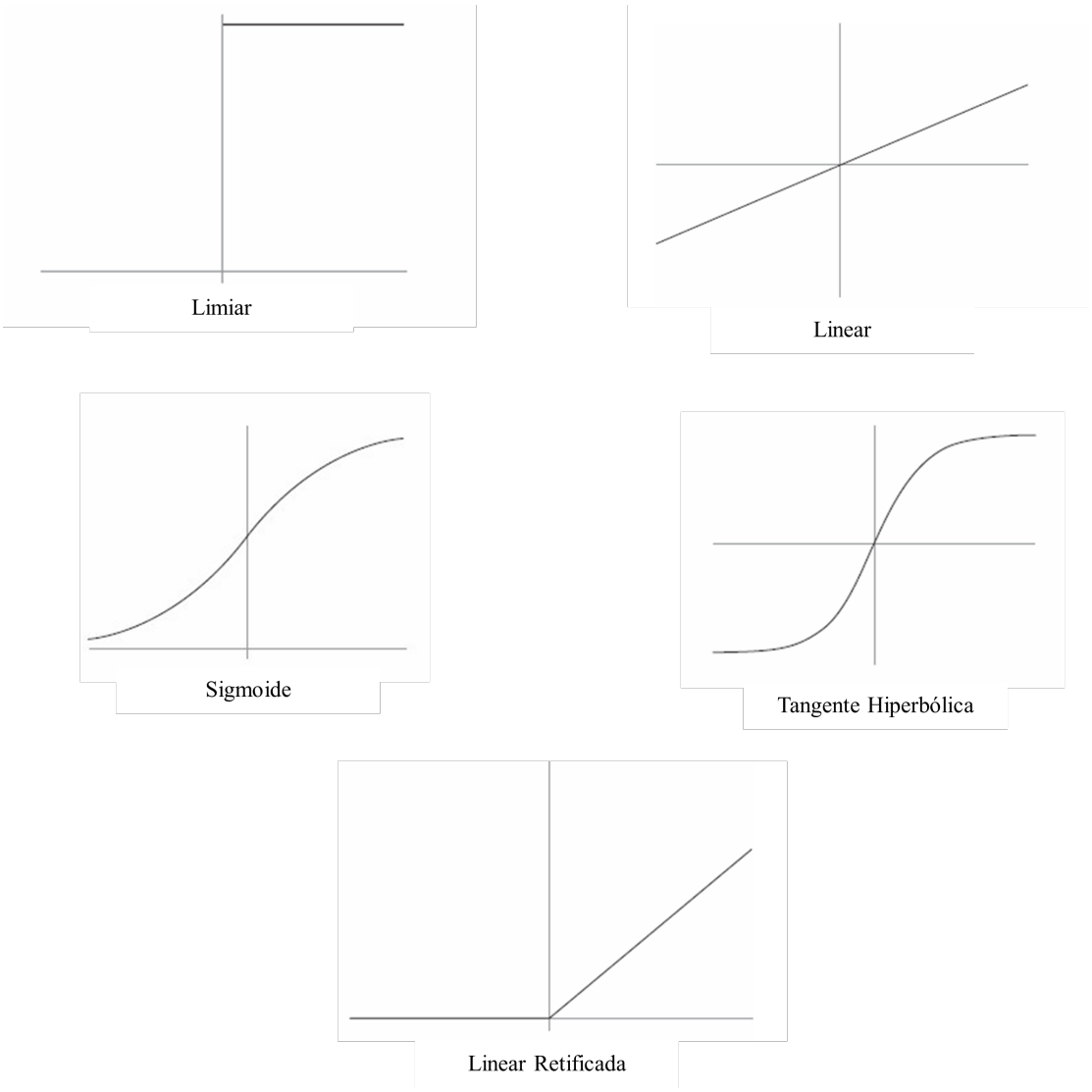
$$f(x) = \frac{1}{1 + e^{-ax}} \tag{2.4}$$

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-ax}} \tag{2.5}$$

$$f(x) = \max(0, x) \tag{2.6}$$

As redes Perceptron, que apresentam somente uma camada de neurônios, de acordo com o pesquisador Frank Rosenblatt, é capaz de classificar qualquer conjunto de dados que possua uma relação de linearidade. Essa característica, foi observado como uma limitação das redes de uma camada, já que não conseguem resolver problemas que não são linearmente separáveis, a exemplo, o operador lógico *XOR* que não possui uma reta linear que separe os valores de saída. Visto isso, as redes de múltiplas camadas, foram

Figura 4 – Funções de Ativação.



Fonte: Autor.

introduzidas por Minsky e Papert em 1969, as quais, a partir da adição de outra camada com mais nós de processamento -neurônios- , foi possível resolver o problema da ou-exclusiva e, portanto, trabalhar problemas não lineares. Nesse contexto, o resultado de um dado neurônio, de uma dada camada, é uma combinação das funções realizadas por neurônios de camadas anteriores, o que, aumenta a complexidade da rede neural, assim como, a capacidade de modelagem de problemas lineares ou não (FACELI et al., 2011). Dessa forma, a arquitetura de uma rede neural com múltiplas camadas, consiste em uma camada de entrada, camadas intermediarias ou ocultas e camada de saída, que podem estar complemente, parcialmente ou localmente conectadas.

2.5 Redes Neurais Recorrentes

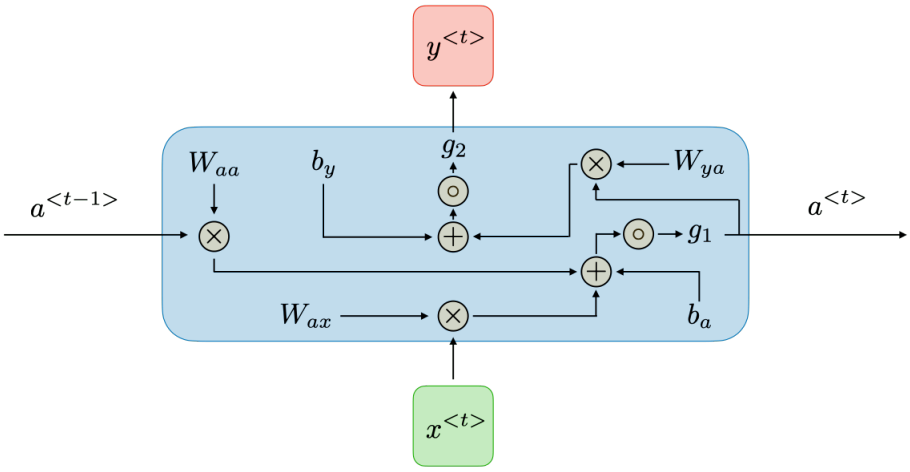
As redes neurais do tipo MLP, ao longo do desenvolvimento de problemas por meio de algoritmos de máquina, apresentaram limitações ao lidar com sequências (texto, vídeo, partitura musical), isto é, guardar instâncias anteriores para o processamento. Nesse cenário, foram desenvolvidas as Redes Neurais Recorrentes - RNN - uma classe que permite que saídas anteriores sejam usadas como entradas enquanto possuem estados ocultos. Isto é, a camada recorrente é capaz de representar a relação entre os elementos em uma atualização na memória interna, estabelecendo uma relação de dependência temporal (DENG; LIU, 2018).

A RNN pode ser representada, de forma resumida, pelas Equações 2.7 e 2.8, atualizadas a cada intervalo de tempo, $\mathbf{t}$. Onde, W_{ax}, W_{aa}, W_{ya}, b_a e b_y são coeficientes compartilhados temporalmente e g_1, g_2 são funções de ativação.

$$a_t = g_1(W_{aa}a_{t-1} + W_{ax}x_t + b_a) \tag{2.7}$$

$$y_t = g_2(W_{ya}a_t + b_y) \tag{2.8}$$

Figura 5 – Estrutura de uma Rede Neural Recorrente.



Fonte: (AMIDI, 2024).

Com isso, a RNN possui aplicações que solucionam problemas cuja entrada é uma sequência e o resultado é uma determinada inferência. Ademais, trata problemas em que as informações de entrada é um elemento único e a saída é uma sequência, assim como soluciona problemas em que a entrada e a saída são sequências diretamente relacionadas pelo tempo, ou seja, sincronizadas ou não (FACELI et al., 2011). Sob essa ótica, pode-se destacar como pontos positivos desse algoritmo a possibilidade de processar entradas de

qualquer tamanho, quer dizer, a dimensão do modelo não aumenta com o tamanho da entrada e, além disso, considera informações históricas.

Por outro lado, as RNNs apresentam a dificuldade de acessar informações distantes temporalmente e a incapacidade de considerar entradas futuras para o estado atual como aspectos negativos. Além disso, um aspecto característico dessa classe de rede neural é a explosão do gradiente[5] (*Vanishing Gradient*), em função da dificuldade de estabelecer relações de longo prazo do gradiente que pode aumentar ou diminuir de forma exponencial, como resultado, desacelera o processo de aprendizado e impede que a rede interprete padrões de maior complexidade no conjunto de dados.

2.5.1 *Long Short-Term Memory - LSTM*

A LSTM consiste em um modelo particular de rede neural, pois é indicada para trabalhar com informações em sistemas dinâmicos. Nessa configuração, as camadas recorrentes interpretam informações passadas e correntes por meio de conexões de *feedback* entre as unidades funcionais.

Em 1997, Hochreiter e Schmidhuber propuseram o primeiro modelo da unidade base de processamento da LSTM, a célula de memória. Nesse contexto, por meio dessa arquitetura, foi estabelecido o *forget gate* (portão do esquecimento), *input gate* (portão de entrada) e o *output gate* (portão de saída) (DENG; LIU, 2018).

O *input gate* é responsável por controlar o fluxo de entrada de novos dados. Dessa forma, sua estrutura é formada por x_t, dado de entrada no instante **t**, h_{t-1}, estado da camada oculta no instante ***t-1***, conforme a Equação 2.9, i_t é resultado da operação *sigmoid* no instante de tempo t, este valor é somado ao valor candidato à C'_t, o qual, de acordo com a Equação 2.10, é resultado de uma operação tangente hiperbólica. Os valores de **W** e **b** referem-se aos pesos e viés, respectivamente. Segundo a indicação da Figura 6, C_t é o resultado da célula após o intervalo de tempo i, cujo valor é determinado pela operação da Equação 2.11, onde f_t corresponde ao valor do *forget gate* no intervalo de tempo **t** (SILVA, 2023; DENG; LIU, 2018).

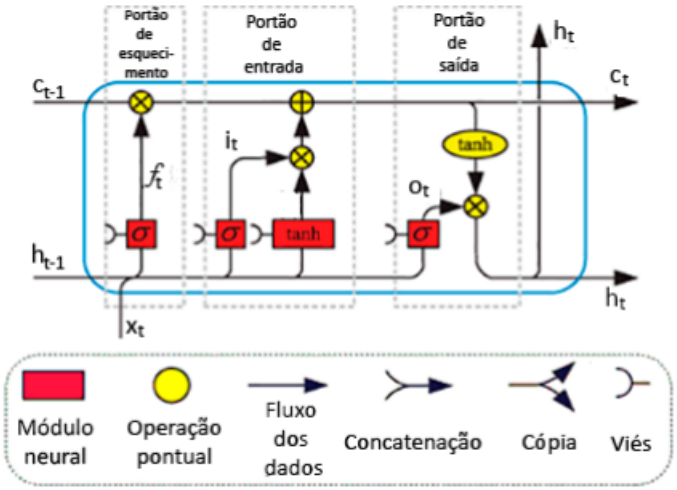
O *output gate*, por sua vez, controla o que a memória contribui para o bloco de ativação da saída. Em síntese, o_t denota o valor utilizado na saída do intervalo de tempo **t**, na multiplicação vetorial, de acordo com a Equação 2.14, um produto ponto a ponto com a tangente hiperbólica do valor do estado oculto da célula. Este resultado, em uma próxima interação, será utilizado para gerar um novo resultado para o estado oculto (SILVA, 2023; DENG; LIU, 2018).

Nessa arquitetura, a LSTM apresenta maior robustez ao problema de *Vanishing*

[5] Situação em que os pesos das camadas anteriores durante o treinamento diminuem de modo exponencial na medida em que se propaga na rede

Gradient em função do *forget gate*. Acontece que durante o cálculo dos pesos, a cascata de derivadas parciais e o valor da derivada *sigmoid* torna-se infinitamente menor que 1, portanto, indica que a rede parou de aprender. Visto isso, a Equação 2.11, determina o valor de saída do *forget gate*, por meio de derivadas parciais, caso $f = 1$, não ocorrerá o decaimento do gradiente de forma rápida, já que entradas anteriores serão lembradas. Com isso, o *forget gate* decide qual informação será mantida ou esquecida (DENG; LIU, 2018).

Figura 6 – Estrutura da LSTM.



Fonte: (SILVA, 2023).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.9}$$

$$C'_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{2.10}$$

$$C_t = f_t * C_{t-1} + i_t * C'_t \tag{2.11}$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2.12}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.13}$$

$$h_t = o_t * tanh(C_t) \tag{2.14}$$

Como a LSTM pode ser fragmentada em etapas intermediárias de aprendizado, essa arquitetura de rede pode ser empilhada, de maneira que, as informações produzidas

em uma etapa se tornem entra para outra. Além disso, pode-se estimar a complexidade computacional da LSTM por intermédio do número de operações e o número de parâmetros treináveis. Por exemplo, considere que uma rede receba vetores de tamanho m na entrada e, como resultado, forneça vetores de tamanho n, a complexidade de memória é expressa por$4(mn + n^2 + n)$. Ademais, o número de operações necessárias para executar uma etapa do treinamento, pode-se calcular da seguinte forma: $mn^2 + nm^2 - nm$, essa operação é feita para cada *forget gate* e para cada célula (PETROZZIELLO et al., 2022).

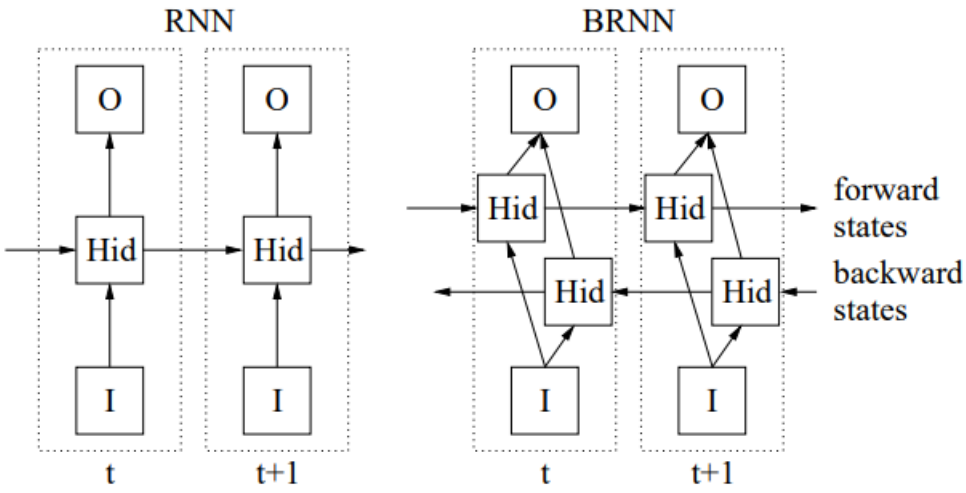
2.5.2 *Bidirectional Long Short-Term Memory - BILSTM*

A BILSTM, é uma arquitetura de rede recorrente e bidirecional, em outras palavras, é uma extensão da arquitetura LSTM padrão, cuja particularidade consiste em considerar o contexto passado e futuro para a modelagem de sequências (SHIRI et al., 2023). Com isso, na medida em que a LSTM processa informações em uma direção, a bidirecional é composta por duas camadas paralelas: uma promove o processamento no sentido para frente, ao passo que, a outra executa a sequência de entrada na direção inversa, como resultado, saída é uma combinação de ambas as camadas (SHIRI et al., 2023) (SUN et al., 2023).

As redes neurais do tipo RNN processam sequências em ordem temporal, isto é, não consideram o contexto futuro. Para tanto, poderia ser adicionado uma janela de tempo de contexto futuro, porém, aumentaria o número de pesos na entrada. Outra solução, é adicionar um atraso entre as entradas e os alvos, o que faz a rede resgatar a entrada original e o seu contexto durante o atraso, o que, não é atrativo. Nesse contexto, (SCHUSTER; PALIWAL, 1997), desenvolveram as redes neurais recorrentes bidirecionais, cuja a ideia básica é utilizar a sequência de treinamento no sentido direto e inverso à duas camadas ocultas recorrentes separadas e conectadas a mesma saída, conforme a Figura 7. Nesse modo, a rede consegue trabalhar as classificações em contexto que pondera passado e futuro de forma simétrica (KAWAKAMI, 2008), como resultado, é uma ferramenta que estima a probabilidade condicional de uma sequência dada e não a sequência de maior probabilidade (SCHUSTER; PALIWAL, 1997).

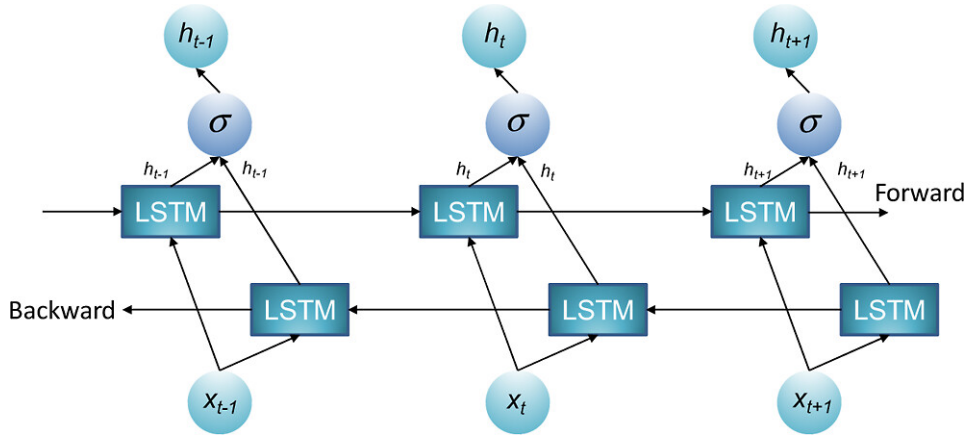
Na Figura 8 é possível observar um exemplo da arquitetura BILSTM. Segue que, de forma simplificada, o estado da camada oculta da rede LSTM direta é dado pela Equação 2.15, já a camada reversa, pode ser descrita pela Equação 2.16, em que h'_t e h'_{t-1} são respectivamente o estado da camada oculta reversa no tempo t e no tempo *t-1*, dessa forma, essas camadas atualizam seus estados internos com base na sequência de entrada e, portanto, de forma independente (SHIRI et al., 2023). Nessa configuração, incorporar informações em ambas as direções, fornece maior abrangência e capacidade de modelar dependências temporais em dados sequenciais (SUN et al., 2023).

Figura 7 – RNN padrão e bidirecional



Fonte: (KAWAKAMI, 2008).

Figura 8 – Arquitetura da BILSTM



Fonte: (SUN et al., 2023).

$$h_t = LSTM(x_t, h_{t-1}) \tag{2.15}$$

$$h'_t = LSTM(x_t, h'_{t-1}) \tag{2.16}$$

3 Metodologia

O desenvolvimento do código e a realização dos experimentos ocorreram no Google *Colab*, um ambiente de desenvolvimento compartilhado fornecido pelo Google. Para o desenvolvimento do preditor, foram utilizados os recursos disponibilizados pelo pacote *TensorFlow*.

As etapas metodológicas incluíram:

1. A replicação de ao menos um experimento realizado por (CHAKRAVARTI; ALLA, 2019);

2. Após a replicação do experimento, realizar o desenvolvimento de um modelo para realizar a detecção da relações entre estruturas químicas e a atividade biológica, avaliado pela acurácia e por técnicas de validação cruzada;

3. Avaliação dos resultados obtidos e discussão dos mesmos;

3.1 Replicação de Experimento - Inibidores do Vírus da Hepatite C (HCV)

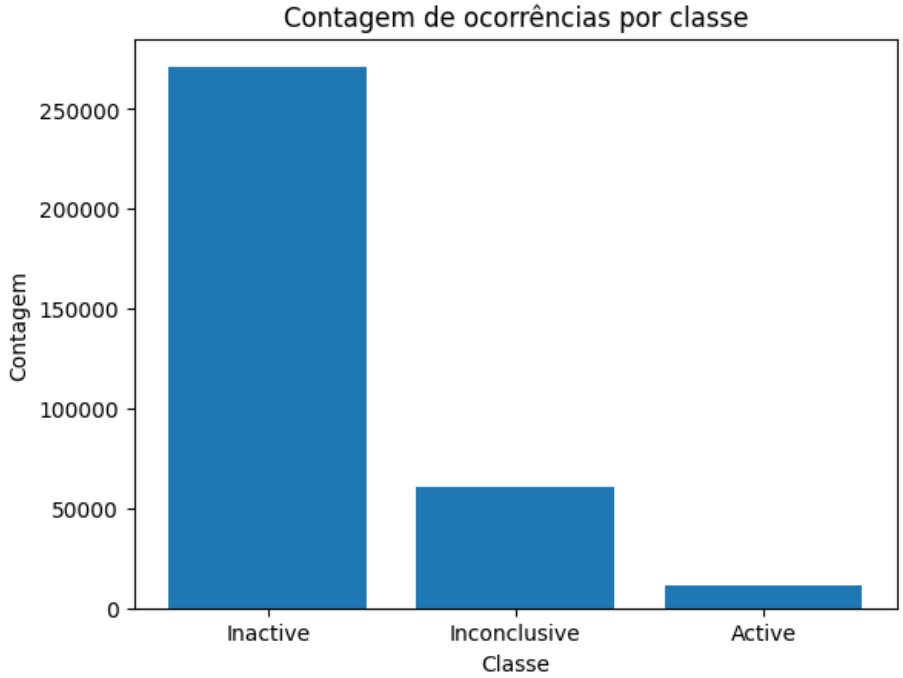
No trabalho desenvolvido por (CHAKRAVARTI; ALLA, 2019), foi realizada a aplicação da ténica SAR para predição, com o uso da rede LSTM, em diferentes experimentos de classificação. Foi realizado três experimentos, o primeiro, para a classificação da mutagenicidade de Ames [1], o segundo experimento, consistiu-se em utilizar uma base dados, disponível no *PubChem*: AID 651820 [2], que expressa a relação entre substâncias químicas e a relação de inibição do HCV e, o terceiro, foi a classificação do conjunto de dados (AID 2302 [3]) que expressa a relação entre a compostos químicos e a inibição do cresciemnto da *Plasmodium falciparum Dd2*, agente causador da malária.

O experimento escolhido para replicação foi a classificação dos inibidores HCV. Acontece que, este experimento foi aplicado a um alvo biológico e, com isso, procedeu-se com o tratamento do conjunto de dados, modelagem do preditor, treinamento e observação do desempenho na validação, por meio de métricas, como a acurácia. Nesse cenário, a base de dados AID 651820, utiliza a descrição das moléculas em forma de SMILES com 343.600 componentes, os autores, escolheram de forma aleatória, 9.935 moléculas ativas para inibição do vírus e 25.531 moléculas classificadas como inativas.

[1] Desenvolvido pelo cientista Bruce Ames na década de 1970 é um dos teste usado para detectar se uma substância tem a capacidade de causar mutações no DNA.
[2] AID 651820: PubChem BioAssay Record. Disponível: https://pubchem.ncbi.nlm.nih.gov/bioassay/651820
[3] AID 2302: PubChem BioAssay Record. Disponível: https://pubchem.ncbi. nlm.nih.gov/bioassay/2302

No experimento, foi modelado um preditor LSTM que obteve um valor de 87,30% de acurácia. Para tanto, foi utilizado 256 neurônios, 9.000 épocas, uma taxa de aprendizado de 0,0001 e um *batch size* de 256. Durante a replicação desse método, foi aplicado, um tratamento ao conjunto de dados que, utilizou-se uma amostra de 34.382 moléculas, coletadas de forma aleatória e distribuídas inicialmente, conforme a Figura 9. Observado o desbalanceamento, entre a as duas classes, foi feito um segundo tratamento nos dados, no qual, aplicou-se o método *sample ()*, disponível na biblioteca *Pandas*. Com o uso deste método, procurou-se aproximar o número de moléculas ativas e inativas. Em seguida, foi aplicado o tratamento textual às SMILES, promoveu-se a tokenização, a transcrição textual para sequência numérica e o preenchimento dos vetores textuais.

Figura 9 – Conjunto de Dados HCV



Fonte: Elaborado pelo Autor

3.2 Classificação de Multiplos Alvos

3.2.1 Tratamento dos Dados

A etapa de levantamento do conjunto de dados é crucial para o desenvolvimento do modelo computacional, com o objetivo de determinar as relações entre as estruturas químicas e a atividade biológica. Neste contexto, utilizou-se o banco de dados *TOX21*[4], composto por um conjunto de medições qualitativas de toxicidade para 12 alvos biológicos (Figura 10).

[4] Disponível para download em: https://moleculenet.org/datasets-1

Os experimentos estabelecem para cada molécula a reação com algum um dos alvos biológicos, sendo o valor 1 codificado como ativo para reação, o valor 0 para indicar que não ocorreu nenhum indicío de reação e, finalmente, -1 para expressar indiferença. Além disso, as estruturas de cada composto químico são representadas pelos SMILES, um conjunto de caracteres que representam o elemento químico, a ligação e a configuração geométrica da cadeia.

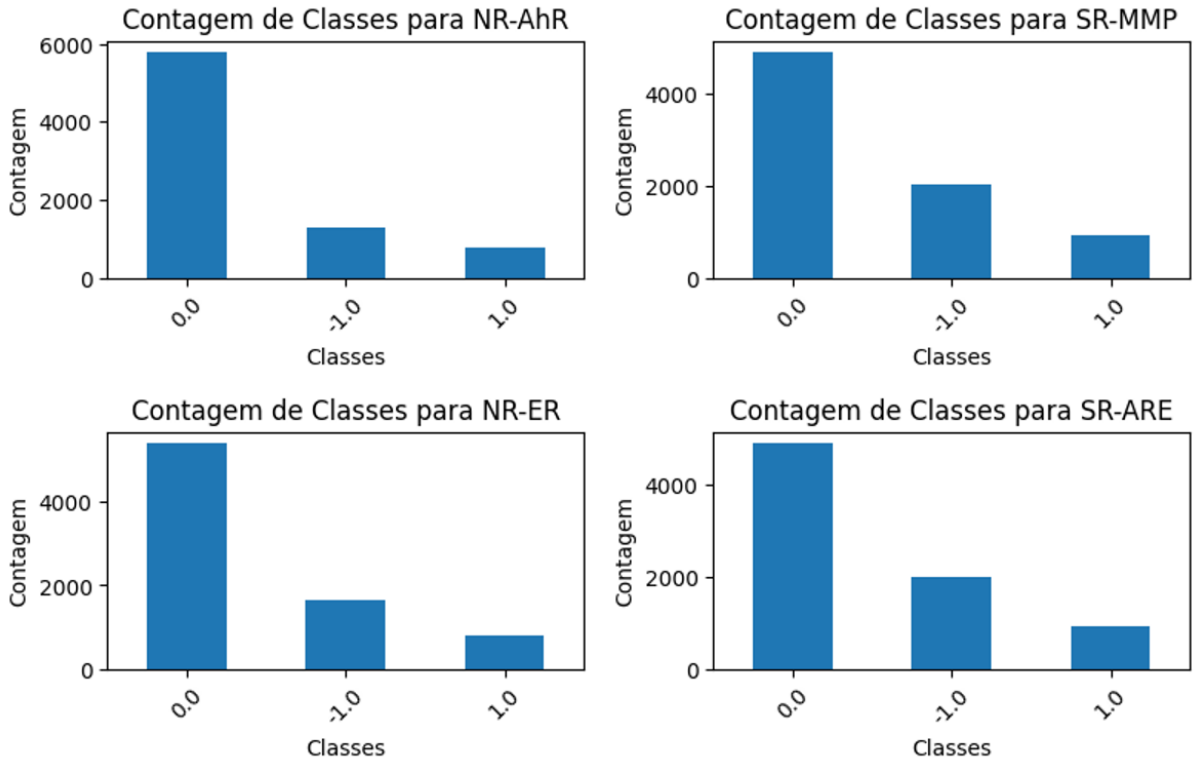
A análise da distribuição das informações permitiu otimizar o processo de treinamento e teste. Foram estratificadas as reações de cada alvo biológico para verificar sua relevância. Durante esse processo, moléculas com baixa, ou nenhuma, atividade biológica foram removidas por não contribuírem significativamente para o aprendizado do modelo.

Figura 10 – Organização do conjunto de dados.

	NR-AR	NR-AR-LBD	NR-AhR	NR-Aromatase	NR-ER	NR-ER-LBD	NR-PPAR-gamma	SR-ARE	SR-ATAD5	SR-HSE	SR-MMP	SR-p53	mol_id	smiles
0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	TOX3021	CCOc1ccc2nc(S(N)(=O)=O)sc2c1
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	TOX3020	CCN1C(=O)NC(c2ccccc2)C1=O
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	TOX3024	CC[C@]1(O)CC[C@H]2[C@@H]3CCC4=CCCC[C@@H]4[C@H]...
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	TOX3027	CCCN(CC)C(CC)C(=O)Nc1c(C)cccc1C
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	TOX20800	CC(O)(P(=O)(O)O)P(=O)(O)O

Fonte: Elaborado pelo autor.

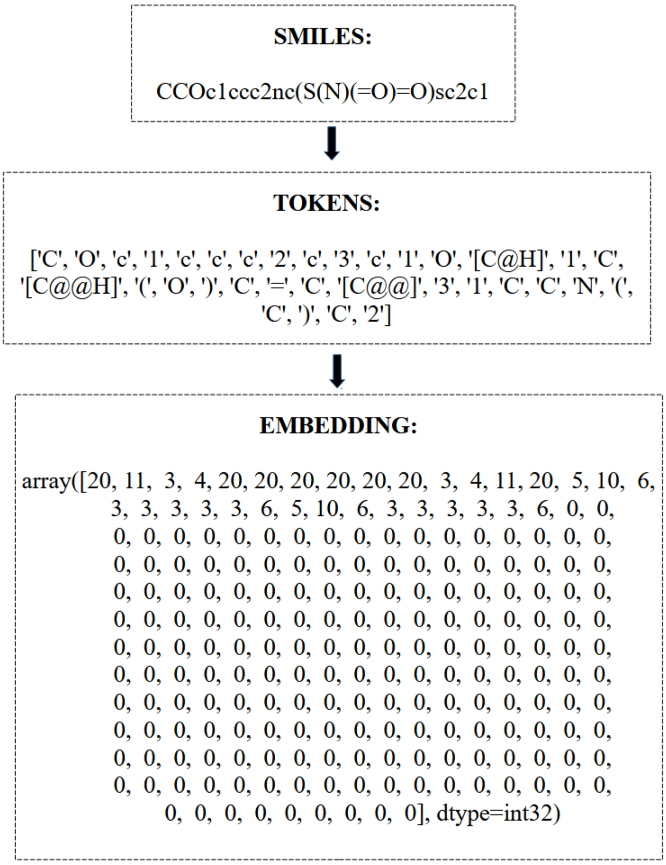
Figura 11 – Estratificação de 4 alvos biológicos com maior atividade química.



Fonte: Elaborado pelo autor.

Após a seleção dos alvos, procedeu-se com o processamento das SMILES. As moléculas selecionadas foram convertidas em uma lista de valores numéricos, formato ideal para interpretação computacional. Esse processo incluiu a tokenização, conversão do texto em sequência numérica, enquanto preserva a semântica, conforme exemplificado na Figura 12, sem aplicar, qualquer *embedding* previamente treinamento para atribuição de valores aos *tokens*, assim como o experimento de (CHAKRAVARTI; ALLA, 2019).

Figura 12 – Word Embedding da molécula do Omeprazol.



Fonte: Elaborado pelo autor.

3.2.2 Treinamento e Validação

Para as camadas da rede neural implementada, optou-se pela utilização da LSTM, devido à sua capacidade de considerar dependências de longo prazo nos dados. A primeira camada, *Embedding*, mapeia índices inteiros a vetores densos, com pesos inicializados de forma aleatória e ajustados via retropropagação. A segunda camada, uma LSTM bidirecional, processa a sequência de entrada de maneira direta e reversa.

A segregação dos dados em conjuntos de treinamento e validação é um passo importante no desenvolvimento de qualquer rede neural. Utilizou-se o *Train Test Split* para dividir os dados, com o intuito de identificar os melhores pesos para otimizar o desempenho

Figura 13 – Resumo do modelo implementado.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 213, 128)          25600

 bidirectional_1 (Bidirecti  (None, 213, 8)            4256
 onal)

 lstm_3 (LSTM)               (None, 4)                 208

 dense (Dense)               (None, 4)                 20

=================================================================
Total params: 30084 (117.52 KB)
Trainable params: 30084 (117.52 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fonte: Elaborado pelo autor.

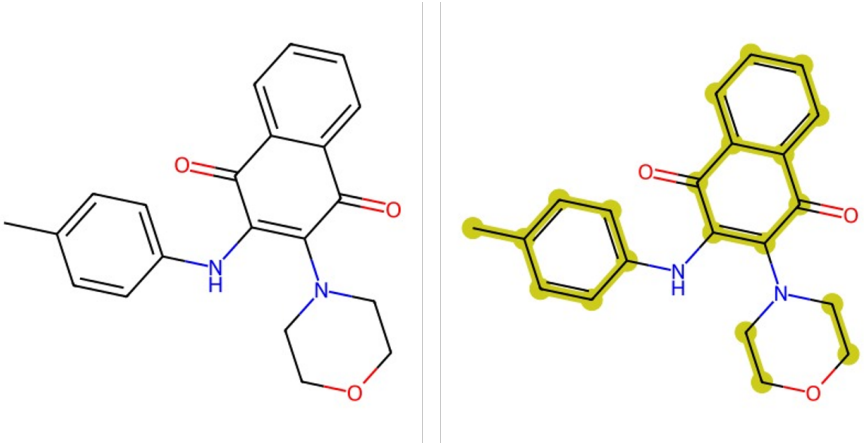
do modelo durante o treinamento, e posteriormente, avaliar o verdadeiro aprendizado por meio da validação. Dados exclusivos foram utilizados para cada conjunto, quer dizer, foi garantido que uma informação do conjunto de treinamento não estivesse presente, também, no conjunto de validação, isto foi feito, por meio da definição de parâmetros fornecidos pelo método de separação. Além de garantir essa exclusividade, foi estabelecido o tamanho de cada grupo, normalmente, o dados de treino são proporcionalmente maiores do que os dados de validação, já que durante o treinamento, acontece o processo de interpretação das relações entre os dados, para tanto, o conjunto de dados foi segmentado em 70% para o treinamento e 30% para a validação. Essa segregação permitiu testar a capacidade de generalização e previsão do modelo.

A definição de métricas, como a acurácia e a função de perda, também é um processo imperativo no desenvolvimento, pois, é responsável por avaliar a performance do algoritmo. A acurácia consiste na taxa de predições corretas e a função de perda, por sua vez, estima a qualidade da predição. Nesse contexto, valores elevados de perda indicam um desempenho insatisfatório, por outro lado, o valores próximos de zero indicam que estabeleceu-se um processo de aprendizado. Parâmetros adicionais como taxa de aprendizado, número de épocas e *dropout* foram definidos, por meio de teste de treinamento, a fim de para maximizar a acurácia de validação e evitar tanto o *Overfitting* quanto o *Underfitting*.

3.2.3 *LIME: Local Interpretable Model-Agnostic Explanations*

Integrou-se ao preditor a biblioteca LIME para elucidar as previsões realizadas durante a classificação. Essa ferramenta ajuda a identificar quais segmentos da molécula têm maior influência na classificação atribuída, oferecendo uma análise detalhada e enriquecida dos resultados. Isto é, contribui para mapear quais trechos e sequencias dos compostos químicos são o princípio ativo de alguma reação.

Figura 14 – LIME: Identificação de segmentos significativos de uma molécula.



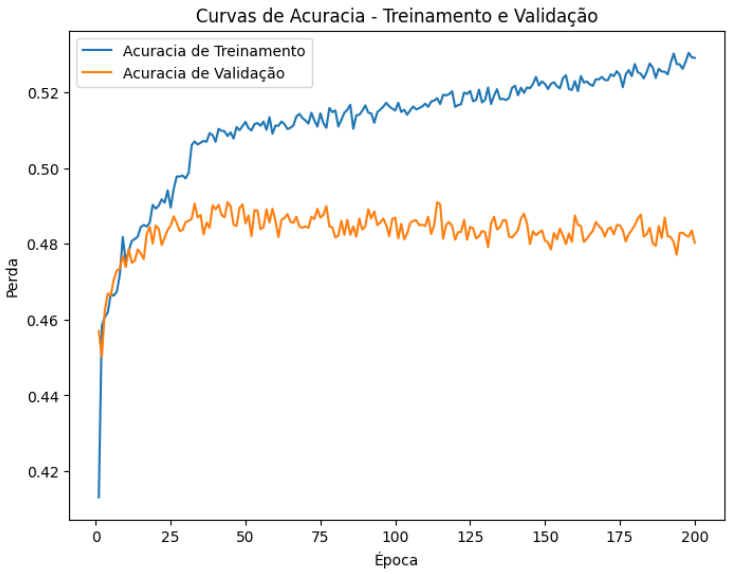
Fonte: Elaborado pelo autor.

4 Resultados

4.1 Replicação de Experimento - Inibidores do Vírus da Hepatite C (HCV)

Na replicação do experimento proposto por (CHAKRAVARTI; ALLA, 2019), foi realizado um treinamento com 200 épocas, diferente daquele valor que os autores estabeleceram, em função da limitação dos recursos de processamento. A rede LSTM utilizada, seguiu a mesma arquitetura apresentada no experimento, isto é, 256 neurônios, o tamanho do vocabulário igual a 34, *batch size* igual a 256 e um *learning rate* de 0,0001. Com isso, obteve-se os seguintes conforme as Figura 15 e Figura 16.

Figura 15 – Acurácia de Treinamento e Validação



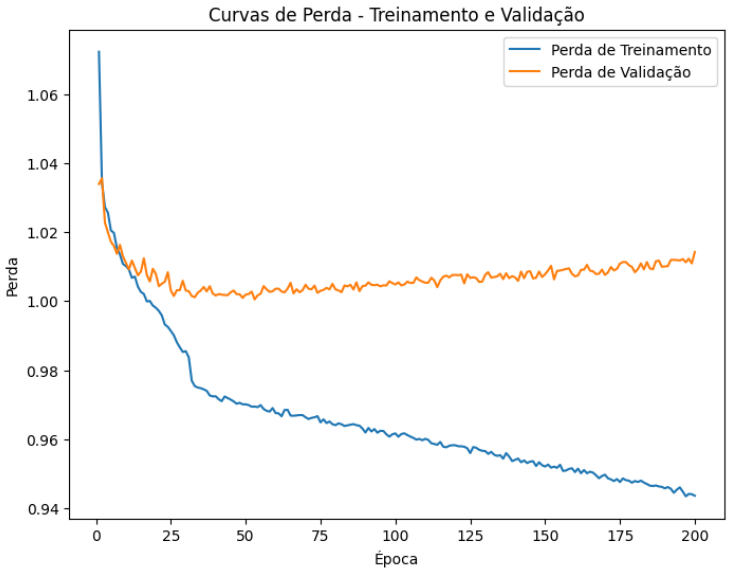
Fonte: Elaborado pelo autor.

4.2 Classificação de Múltiplos Alvos

Os resultados apresentados nesta seção foram obtidos após a estratificação do conjunto de dados e uma série de treinamentos com diferentes configurações do modelo. O objetivo foi determinar a arquitetura de maior eficácia diante do conjunto de dados disponível.

A análise do *dataset TOX21* revelou a distribuição de reações para cada um dos 12 alvos biológicos, destacando-se os alvos NR-AhR, NR-ER, SR-MMP e SR-ARE pela

Figura 16 – Perda de Treinamento e Validação



Fonte: Elaborado pelo autor.

sua maior relevância. Contudo, a distribuição também indicou que reações positivas correspondem a aproximadamente 6% do total de interações.

As Figuras 17, 18 e 19 apresentam os resultados para as diferentes arquiteturas testadas. A estratégia inicial consistiu na utilização da BILSTM, seguida pela LSTM, e posteriormente, uma combinação dessas arquiteturas. Ademais, cada configuração foi avaliada com base em sua acurácia e sua função de perda, de modo a estabelecer a melhor performance. Foi considerado em todos os modelos os parâmetros a seguir: função de ativação, a *sigmoide*; função de perda, a *Binary crossentropy*; otimizador, o Adam; tamanho da camada de *embedding*, dimensão equivalente ao maior vetor do cojunto de dados (213); número de épocas igual 1000; *learning rate* igual a 0,001 e um *batch size* igual a 64. Como resultado, garantiu-se uma padronização nos experimentos e estabeleceu-se a complexidade dos modelos como variável manipulada.

Figura 17 – Resultado BILSTM.

Arquitetura	Formato	Acurácia Treinamento	Perda Treinamento	Acurácia Validação	Perda Validação
BILSTM	Embedding (None, 213, 128) BILSTM (None, 2) Dense (None,4)	56,76%	0,948	49,84%	1,002
	Embedding (None, 213, 128) BILSTM (None, 4) Dense (None,4)	58,21%	0,909	51,77%	0,980
	Embedding (None, 213, 128) BILSTM (None, 8) Dense (None,4)	56,68	0,900	51,13%	1,009
	Embedding (None, 213, 128) BILSTM (None, 16) Dense (None,4)	57,57%	0,901	48,23%	1,012
	Embedding (None, 213, 128) BILSTM (None, 32) Dense (None,4)	63,535	0,821	49,845	1,038
	Embedding (None, 213, 128) BILSTM (None, 64) Dense (None,4)	62,88%	0,809	53,70%	1,073

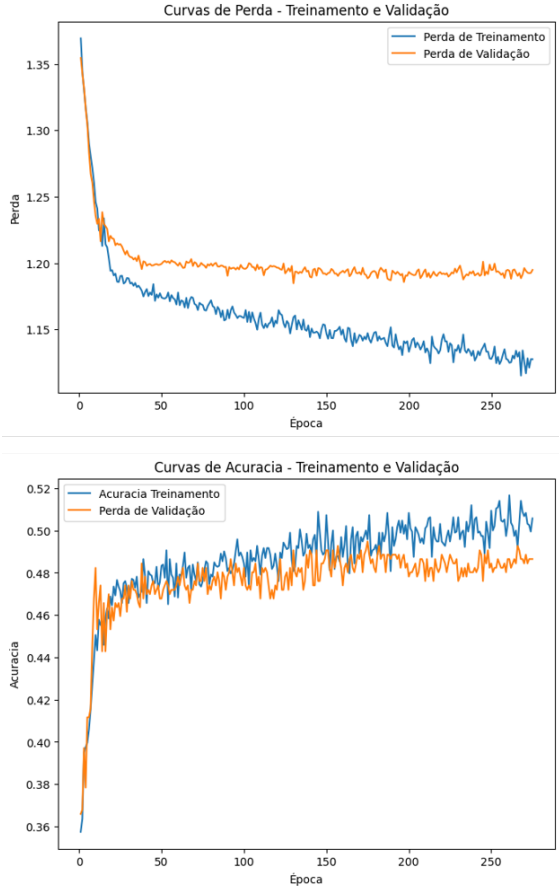
Fonte: Elaborado pelo autor.

Figura 18 – Resultado LSTM

Arquitetura	Formato	Acurácia Treinamento	Perda Treinamento	Acurácia Validação	Perda Validação
LSTM	Embedding (None, 213, 128) LSTM (None, 2) Dense (None,4)	55,39%	0,990	44%	1,056
	Embedding (None, 213, 128) LSTM (None, 4) Dense (None,4)	57,49%	0,972	51,77%	1,028
	Embedding (None, 213, 128) LSTM (None, 8) Dense (None,4)	56,12%	0,958	52,09%	1,002
	Embedding (None, 213, 128) LSTM (None, 16) Dense (None,4)	59,42%	0,884	52,09%	1,017
	Embedding (None, 213, 128)LSTM (None, 32) Dense (None,4)	54,43%	0,922	47,91%	1,031
	Embedding (None, 213, 128) LSTM (None, 64) Dense (None,4)	59,90%	0,865	52,09%	1,019

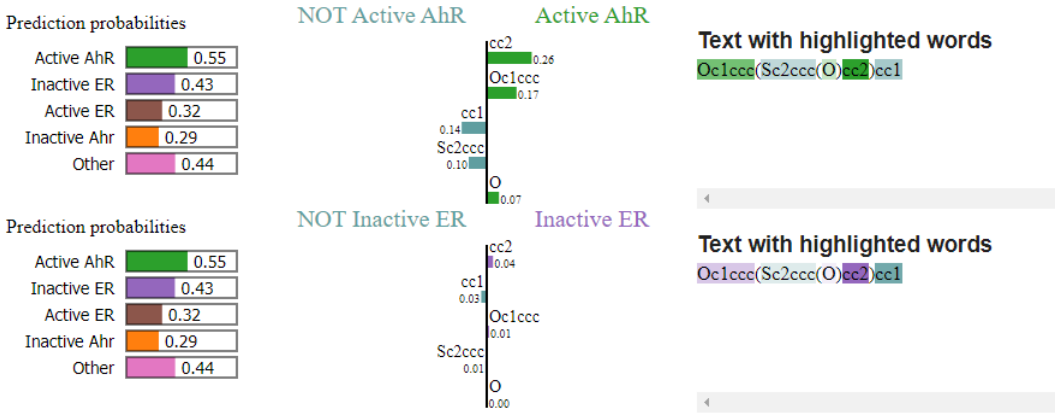
Fonte: Elaborado pelo autor.

Figura 19 – Arquitetura: BILSTM e LSTM.



Fonte: Elaborado pelo autor.

Figura 20 – Análise LIME para uma molécula do conjunto de validação.



Fonte: Elaborado pelo autor.

5 Discussão dos Resultados

Foi observado que a classe -1, indiferença, não contribuiu significativamente para a análise, como resultado, foi removida. Após a seleção dos alvos biológicos com maior relevância, optou-se pelo uso do SR-ARE e do SR-MMP. Acontece que, a perfomance da predição para os quatro alvos biológicos apresentou, ao longo dos treinamentos, dificuldades de aprendizagem. Com isso, além de estabelecer o experimento com os dois sistemas biológicos mais significantes, realizou-se uma codificação *One Hot Enconding*, através da combinação das possibilidades de resultados possíveis, conforme a figura Figura 21.

Figura 21 – Codificação dos rótulos.

	SR-ARE	SR-MMP	SR-ARE_SR-MMP(0,0)	SR-ARE_SR-MMP(1,0)	SR-ARE_SR-MMP(0,1)	SR-ARE_SR-MMP(1,1)
0	1	0	0	1	0	0
1	1	0	0	1	0	0
2	1	0	0	1	0	0
3	0	0	1	0	0	0
4	0	0	1	0	0	0
...	...	...	...	...	...	...
2865	1	1	0	0	0	1
2866	0	1	0	0	1	0
2867	0	0	1	0	0	0
2868	0	0	1	0	0	0
2869	1	0	0	1	0	0

2870 rows × 6 columns

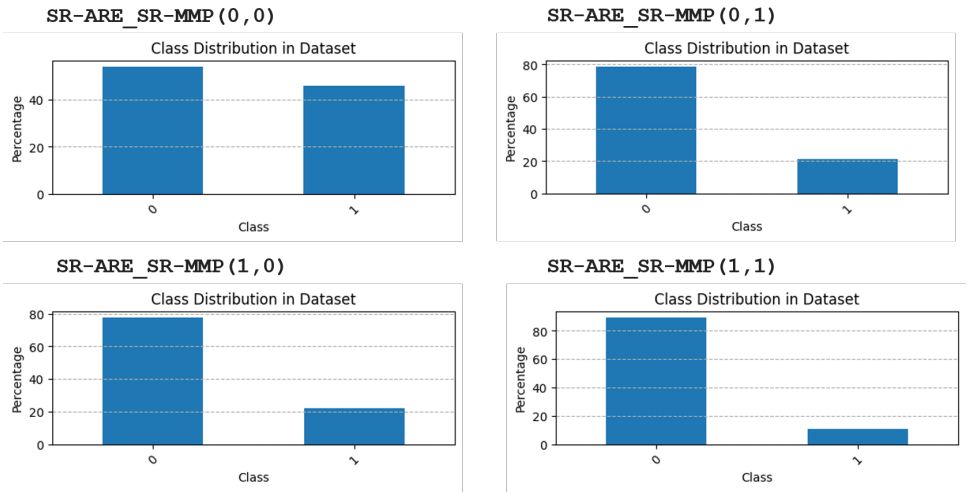
Fonte: Elaborado pelo autor.

A manutenção da proporção entre os rótulos nos conjuntos de treinamento e validação foi necessária para evitar a penalização de rótulos menos frequentes. A utilização do parâmetro *stratify* assegurou a constância na distribuição dos dados antes e após a segmentação pelo *Train Test Split*, conforme ilustrado nas Figuras 22 e 23. Com isso, procurou-se contornar a possibilidade de *Overfitting* ao longo do treinamento, uma vez que foi amostrado dados para classificação positiva e negativa.

Os resultados das diferentes arquiteturas revelaram um desempenho inferior ao apresentado por (CHAKRAVARTI; ALLA, 2019), na predição dos inibidores HCV, enquanto experimento de referência, apresentou uma acurácia de 87,30%. Além disso, observa-se que este resultado refere-se a uma classificação para um rótulo. A predição multirrótulo, por sua vez, possui maior complexidade, visto o maior número de combinações e resultados parciais, isto é, pode-se classificar, para um objeto, alguns rótulos de forma correta e outros de maneira incorreta, o que também exige outras métricas para avaliação (GONÇALVES, 2018),.

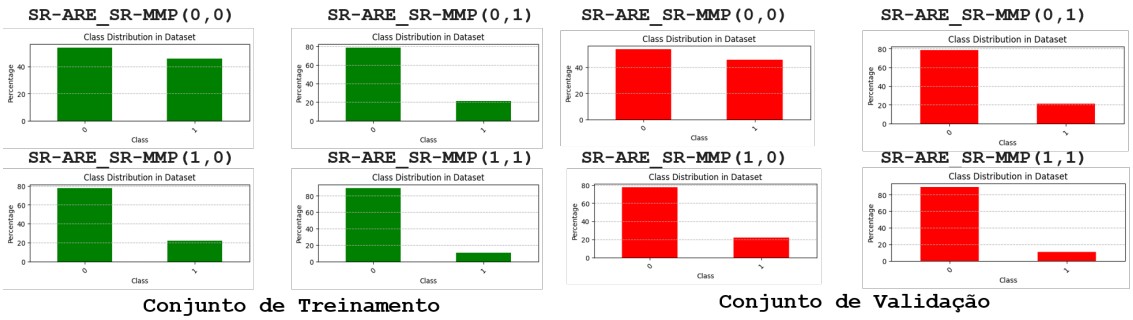
A exploração das configurações da BILSTM e da LSTM, e posteriormente, a combi-

Figura 22 – Distribuição dos rótulos no conjunto de dados final.



Fonte: Elaborado pelo autor.

Figura 23 – Distribuição de rótulos no treinamento e na validação.



Fonte: Elaborado pelo autor.

nação de ambas, não resultou em melhorias significativas de desempenho, o que sugere-se a necessidade de uma investigação mais aprofundada sobre as particularidades do conjunto de dados e possíveis ajustes no modelo. Além disso, conforme (GONÇALVES, 2018), com o objetivo de desenvolver o algoritmo de classificação multirrótulo, existem outras abordagens que podem ser implementadas conforme as características do problema. Nesse contexto, a exemplo, o *Binary Relevance* (BR), que consiste em decompor a base de dados original em dados binários para cada rótulo de destino Figura 24. Nessa configuração, é utilizado um preditor para cada classe, treinados de forma independente, o resultado final consiste na composição produzida por cada preditor.

O LIME, por sua vez, conforme a Figura 20, mostra como uma determinada molécula interagiu com um determinado alvo. Neste exemplo, de forma gráfica, podemos observar como um algoritmo preditor avaliou cada trecho do composto químico a fim de rotular o sucesso da reação. A partir desse resultado, observa-se que determinadas sequências da molécula química apresenta maior atividade biológica. Sob esse cenário,

Figura 24 – Transformação BR da base de dados de categorização de músicas.

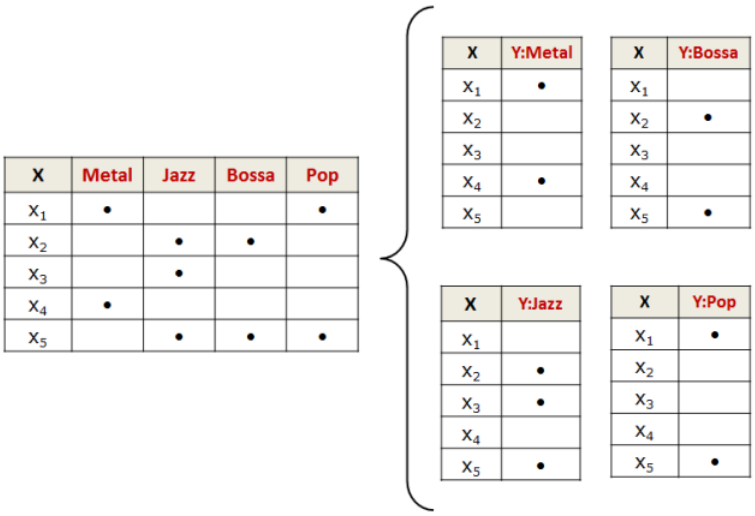


Figura 3.15. Transformação BR da base de dados de categorização de músicas

Fonte: (GONÇALVES, 2018).

verifica-se que o processo de segmentação da molécula em *tokens*, ao longo do processo de tratamento dos dados, contribui de forma positiva, uma vez que permite ao preditor estabelecer relações entre esses segmentos de molécula e os alvos biológicos.

6 Conclusão

A LSTM, conforme apresentado por (A. MULLEN et al., 2018) e evidenciado pelos resultados dos treinamentos, demonstra robustez no contexto de *Deep Learning*. Observou-se que, a partir dos experimentos com diferentes configurações do modelo neural, o aumento na capacidade da rede, como o número de células e a combinação de camadas, não resultou em melhorias significativas no aprendizado. Assim, pode-se inferir, que a dimensão do preditor não constitui a causa primária do *Overfitting*.

Outra hipótese que requer investigação, dada a observação dos níveis de acurácia e de perda obtidos, relaciona-se ao volume e à qualidade dos dados disponíveis. A presença de um número limitado de amostras com reações positivas, em comparação ao total, manteve-se como uma característica desafiadora nas abordagens adotadas para o tratamento dos dados. Por conseguinte, sugere-se a exploração de técnicas mais avançadas para o processamento do conjunto de dados, com o objetivo de alcançar um balanceamento e uma representatividade adequada entre as classes de cada alvo.

Ademais, pode-se observar a necessidade de explorar outras abordagens para o tratamento de problemas multirrótulos. Conforme observado, os experimentos aplicados à classificação para um alvo biológico, por meios convencionais, apresentou valores de acurácia significativos, conforme apresentam (CHAKRAVARTI; ALLA, 2019). No entanto, foi observado que a classificação para mais alvos biológicos, de forma simultânea, necessita-se também, da aplicação de outras estrategias de modelagem alinhadas ao tipo de problema. Sob essa perspectiva, conforme aponta (GONÇALVES, 2018) é necessário aplicar métricas que vão ao encontro do contexto aplicado, uma vez que, as características da base de dados também influenciam o preditor.

Com o desenvolvimento deste trabalho, conclui-se que a aplicação de técnicas de aprendizado profundo é viável para determinar a relação entre estruturas químicas e múltiplos alvos biológicos. Contudo, torna-se imperativa a análise das causas subjacentes que influenciam o desempenho subótimo da rede neural. Adicionalmente, a integração da ferramenta LIME, destinada a identificar segmentos moleculares de maior relevância para os resultados de classificação, enriquece a interpretação das predições. Assim, espera-se que, com aprimoramentos no modelo neural que elevem sua precisão e sua capacidade de generalização, a abordagem proporcione insights mais detalhados sobre a classificação de compostos químicos em relação a múltiplos alvos.

Referências

A. MULLEN, Lincoln et al. Fast, consistent tokenization of natural language text. *Journal of Open Source Software*, The Open Journal, v. 3, n. 23, p. 655, 2018. Citado 2 vezes nas páginas 14, 36.

AMIDI, Shervine. *CS 230 - Recurrent Neural Networks Cheatsheet*. 2024. Acesso em: 23 out. 2024. Disponível em: https://web.stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks. Citado 0 vez na página 18.

CHAKRAVARTI, Suman K; ALLA, Sai Radha Mani. Descriptor free QSAR modeling using deep learning with long short-term memory neural networks. *Frontiers in artificial intelligence*, Frontiers Media SA, v. 2, p. 17, 2019. Citado 7 vezes nas páginas 11, 23, 26, 29, 33, 36.

DEMBITSKY, Valery M et al. QSAR study of some natural and synthetic platelet aggregation inhibitors and their pharmacological profile. *Journal of Applied Pharmaceutical Science*, v. 12, n. 5, p. 039–058, 2022. Citado 1 vez na página 8.

DENG, Li; LIU, Yang. *Deep learning in natural language processing*. Springer, 2018. Citado 6 vezes nas páginas 14, 18–20.

FACELI, Katti et al. Inteligência artificial: uma abordagem de aprendizado de máquina, 2011. Citado 5 vezes nas páginas 8, 15–18.

FLECK, Leandro et al. Redes neurais artificiais: Princípios básicos. *Revista Eletrônica Científica Inovação e Tecnologia*, v. 1, n. 13, p. 47–57, 2016. Citado 4 vezes na página 16.

FOOSHEE, David et al. Deep learning for chemical reaction prediction. *Molecular Systems Design & Engineering*, Royal Society of Chemistry, v. 3, n. 3, p. 442–452, 2018. Citado 1 vez na página 11.

GONÇALVES, Eduardo Corrêa. Introdução à Classificação Multirrótulo. *Sociedade Brasileira de Computação*, 2018. Citado 3 vezes nas páginas 33–36.

GREFENSTETTE, Gregory. Tokenization. In: SYNTACTIC wordclass tagging. Springer, 1999. P. 117–133. Citado 1 vez na página 14.

KAWAKAMI, Kazuya. *Supervised sequence labelling with recurrent neural networks*. 2008. Tese (Doutorado) – Ph. D. thesis. Citado 1 vez nas páginas 21, 22.

KHATTAK, Faiza Khan et al. A survey of word embeddings for clinical text. *Journal of Biomedical Informatics*, Elsevier, v. 100, p. 100057, 2019. Citado 2 vezes nas páginas 14, 15.

LIMA, I.; PINHEIRO, C.A.M.; SANTOS, F.A.O. *Inteligência Artificial*. ELSEVIER EDITORA, 2014. ISBN 9788535278088. Disponível em: https://books.google.com.br/books?id=IEFNvgAACAAJ. Citado 1 vez na página 15.

PETROZZIELLO, Alessio et al. Deep learning for volatility forecasting in asset management. *Soft Computing*, Springer, v. 26, n. 17, p. 8553–8574, 2022. Citado 1 vez na página 21.

PINGAEW, Ratchanok et al. Anticancer activity and QSAR study of sulfur-containing thiourea and sulfonamide derivatives. *Heliyon*, Elsevier, v. 8, n. 8, 2022. Citado 1 vez na página 8.

PINGARRÓN, José M et al. Terminology of electrochemical methods of analysis (IUPAC Recommendations 2019). *Pure and Applied Chemistry*, De Gruyter, v. 92, n. 4, p. 641–694, 2020. Citado 1 vez na página 12.

POLISHCHUK, Pavel G; MADZHIDOV, Timur I; VARNEK, Alexandre. Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of computer-aided molecular design*, Springer, v. 27, p. 675–679, 2013. Citado 1 vez na página 9.

RAJAN, Kohulan; ZIELESNY, Achim; STEINBECK, Christoph. STOUT: SMILES to IUPAC names using neural machine translation. *Journal of Cheminformatics*, BioMed Central, v. 13, n. 1, p. 1–14, 2021. Citado 2 vezes na página 13.

RAMOS, Gabrielle Santos. Modelos de aprendizado profundo para avaliação de toxicidade aguda de compostos químicos em aves. Universidade Federal de Goiás, 2022. Citado 1 vez na página 12.

RAUBER, Thomas Walter. Redes neurais artificiais. *Universidade Federal do Espírito Santo*, v. 29, 2005. Citado 0 vez na página 15.

SCHREIBER, Stuart L; KAPOOR, Tarun M; WESS, Günther. Chemical biology: from small molecules to systems biology and drug design. *(No Title)*, 2007. Citado 1 vez na página 9.

SCHUSTER, M.; PALIWAL, K.K. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, v. 45, n. 11, p. 2673–2681, 1997. DOI: 10.1109/78.650093. Citado 2 vezes na página 21.

SERAFIM, Mateus S.M. et al. Knowing and combating the enemy: a brief review on SARS-CoV-2 and computational approaches applied to the discovery of drug candidates. *Bioscience Reports*, v. 41, n. 3, bsr20202616, mar. 2021. ISSN 0144-8463. DOI: 10.1042/BSR20202616. eprint: https://portlandpress.com/bioscirep/article-pdf/41/3/BSR20202616/906000/bsr-2020-2616c.pdf. Disponível em: https://doi.org/10.1042/BSR20202616. Citado 1 vez na página 8.

SHIRI, Farhad Mortezapour et al. A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. *arXiv preprint arXiv:2305.17473*, 2023. Citado 3 vezes na página 21.

SILVA, Rodney. Aprendizado de máquina construtivo e classificação hierárquica multirrótulo aplicados à geração de moléculas. Universidade Federal de São Carlos, 2023. Citado 6 vezes nas páginas 8, 13, 19, 20.

SUN, Youzhuang et al. Bidirectional long short-term neural network based on the attention mechanism of the residual neural network (ResNet–BiLSTM–Attention) predicts porosity through well logging parameters. *ACS omega*, ACS Publications, v. 8, n. 26, p. 24083–24092, 2023. Citado 2 vezes nas páginas 21, 22.

THEODORE L BROWN H. EUGENE LEMAY, Bruce E. Bursten. *Quimíca, a ciência central*. Pearson, 2005. Citado 1 vez na página 12.

TROPSHA, Alexander. Best practices for QSAR model development, validation, and exploitation. *Molecular informatics*, Wiley Online Library, v. 29, n. 6-7, p. 476–488, 2010. Citado 2 vezes na página 9.

TROSSINI, Gustavo HG; MALTAROLLO, Vinícius G; SCHMIDT, Thomas J. Hologram QSAR studies of antiprotozoal activities of sesquiterpene lactones. *Molecules*, Multidisciplinary Digital Publishing Institute, v. 19, n. 7, p. 10546–10562, 2014. Citado 1 vez na página 8.

VERÍSSIMO, Gabriel Corrêa et al. GCN-Based Structure-Activity Relationship and DFT Studies of Staphylococcus aureus FabI Inhibitors. *International Journal of Quantitative Structure-Property Relationships (IJQSPR)*, IGI Global, v. 7, n. 1, p. 1–16, 2022. Citado 2 vezes nas páginas 9, 11.

WEININGER, David; WEININGER, Arthur; WEININGER, Joseph L. SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of chemical information and computer sciences*, ACS Publications, v. 29, n. 2, p. 97–101, 1989. Citado 2 vezes na página 13.

YASONIK, Jacob. Multiobjective de novo drug design with recurrent neural networks and nondominated sorting. *Journal of Cheminformatics*, Springer, v. 12, n. 1, p. 14, 2020. Citado 0 vez na página 13.

YOUNG, David C. *Computational drug design - A Guide for Computational and Medicinal Chemists*. Wiley, 2009. Citado 2 vezes nas páginas 8, 9.