# Revisão das Tecnologias de Inteligência Artificial e Machine/Deep Learning: Restrições, Oportunidades, Estado da Arte e Desafios

Hugo G. Machado & Kleber Mundim

A utilização de algoritmos de aprendizagem de máquina tem aumentado exponencialmente na pesquisa científica, especialmente devido a avanços recentes em técnicas de aprendizado profundo. Aqui, serão discutidas aplicações desses algoritmos na química e em outras áreas da ciência, com foco em redes neurais artificiais. Essas redes têm a capacidade de automatizar todas as etapas do processo de aprendizado de máquina, incluindo a classificação e a predição de propriedades químicas. Será fornecida uma visão histórica do desenvolvimento desses algoritmos, desde a década de 1940 até os dias atuais, com destaque para aplicações em áreas como desenvolvimento de medicamentos, ciência de materiais e técnicas de análise autônomas. Aspectos importantes desses algoritmos serão discutidos em detalhes. Além disso, será abordado o processo de vetorização molecular, essencial para o tratamento de dados químicos, e alguns caracterizadores moleculares serão discutidos em particular. Em conclusão, será fornecida uma visão abrangente das aplicações dos algoritmos de aprendizado de máquina na química, juntamente com suas limitações e desafios associados à sua implementação, destacando seu potencial transformador quando utilizado de maneira responsável e ética.

**Palavras-chave:** *aprendizagem de máquina; química; redes neurais artificiais.*

The use of machine learning algorithms has exponentially increased in scientific research, especially due to recent advances in deep learning techniques. In this text, the applications of these algorithms in chemistry and other scientific fields will be discussed, with a focus on artificial neural networks. These networks can automate all stages of the machine learning process, including the classification and prediction of chemical properties. A historical overview of the development of these algorithms, from the 1940s to the present day, will be provided, highlighting their various applications in areas such as drug development, materials science, and autonomous analysis techniques. Important aspects of these algorithms will be discussed in detail. Additionally, the essential process of molecular vectorization for the treatment of chemical data will be discussed, along with several molecular featurizers. In conclusion, a comprehensive overview of the applications of machine learning algorithms in chemistry will be provided, along with their associated limitations and challenges in implementation, emphasizing the transformative potential of these algorithms when used in a responsible and ethical manner.

**Keywords:** *machine learning; chemistry; artificial neural networks.*

# Introdução

Modelos de aprendizagem estatística baseadas em Aprendizado de Máquina (ML, do inglês: *Machine Learning*) vem sendo utilizados a décadas na química, por exemplo: na modelagem molecular com a técnica de Relação Quantitativa entre Estrutura e Atividade (QSAR, do inglês: *Quantitative Structure Activity Relationship*)[1–4]; na estimativa de parâmetros fenomenológicos;[5] e na descrição de superfícies de energia potencial.[6]

Entre os diferentes algoritmos de ML, a Aprendizagem Profunda (DL, do inglês: *Deep Learning*) é a área que mais cresce e recentemente tem conquistado a ciência devido ao crescimento imprevisível, tanto da habilidade em analisar grandes conjuntos de dados e extrair informações significativas, quanto no progresso feito nas tecnologias de hardware como as Unidades de Processamento Gráfico (GPU, do inglês: *Graphics Processing Unit*) e computação de alto desempenho. O DL é uma subárea de ML e inspira-se nos padrões de processamento de informações encontrados no cérebro humano (Figura 1).

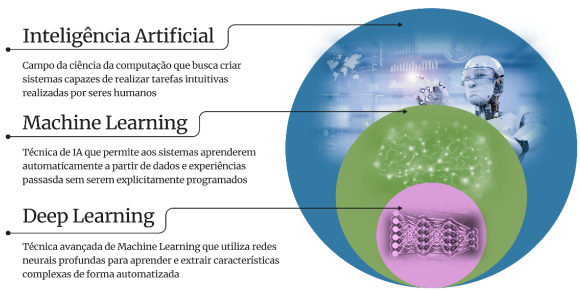


**Figura 1.** Grande área Inteligência Artificial e subáreas Machine Learning e Deep Learning

Algoritmos de ML convencionais podem desempenhar diversas tarefas como regressão e classificação, mas para isso são necessárias várias etapas sequenciais, como: pré-processamento dos dados, extração de características, seleção ou escolha de características, aprendizado por treinamento e classificação. Nesses algoritmos a escolha correta das características utilizadas na predição tem um grande impacto no desempenho do modelo, uma seleção enviesada pode levar à um modelo ruim. Por outro lado, algoritmos de DL são capazes de automatizar todos esses passos, desde o pré-processamento de dados e extração de características até o treinamento e classificação.[7] Algoritmos de DL não requerem nenhuma regra projetada por humanos para operar, são projetados com diversas camadas nas quais cada uma fornece uma interpretação diferente dos dados de entrada, dessa maneira, são capazes de mapear características presentes em grandes conjuntos de dados.

Os principais algoritmos utilizados em DL são baseados nas Redes Neurais Artificiais (RNA) empregando transformações e tecnologias de grafos simultaneamente para construir modelos de aprendizagem multicamadas. Técnicas mais recentes de DL baseadas em RNA's tem obtido um desempenho excelente em uma variedade de aplicações, como o processamento de áudio e fala, visão computacional e processamento de linguagem natural.[8–11] O uso de DL também tem se tornado rotina em diversas aplicações químicas e biológicas, como no desenvolvimento de drogas,[3,12–14]; na predição de propriedades químicas,[15–18] e em cálculos quânticos.[19–22]

Em tecnologias baseadas em ML, normalmente a eficácia dos modelos é altamente dependente da integridade da representação dos dados e por muitos anos a pesquisa em ML apresentou uma tendência de abordagem visando construir novos descritores e estratégias de representação de dados a partir de dados brutos. Neste sentido houve um esforço considerável para a construção bancos de dados, dentre eles pode-se destacar aqueles com: dados de mecânica quântica como o GDB-13[23]; dados físico-químicos como o ESOL[17], FreeSolv[18] e CheMBL[24]; dados biofísicos como o PubChem[25], LNCS[26], PDBbind[27]; e dados fisiológicos como o Tox21[28,29] e SIDER.[30]

Em contraste com métodos precedentes de ML, a extração e classificação de característica em algoritmos de DL é realizada de maneira automática. Por isso técnicas de DL tem atraído atenção dos pesquisadores ao exigir um menor de esforço humano e conhecimento de campo. Esses algoritmos multicamadas possuem uma alta capacidade de generalização, conseguindo extrair características

de baixo nível desde a suas primeiras camadas até características de alto nível em suas últimas camadas. É importante notar que originalmente a IA inspirou-se nesse tipo de arquitetura que simula o processo que ocorre no cérebro humano, ao extrair dados, representá-los e classificá-los automaticamente. Mais especificamente, o resultado (ou saída) desse processo são os objetos classificados enquanto as informações visuais e sensoriais recebidas em diferentes situações representam os dados de entrada.

Neste artigo de revisão, serão abordados aspectos essenciais a respeito das RNA's diretas de regressão e classificação de dados, como os neurônios matemáticos, técnicas de treinamento e métricas de validação de modelos, além de abordar diferentes descritores moleculares para uso destes algoritmos na área de química. Antes disso será feito uma breve abordagem histórica para contextualizar os efeitos e impactos da IA na vida das pessoas e na pesquisa científica.

# Contexto Histórico

Em 1943 Mcculoch e colaboradores[31] foram os pioneiros no desenvolvimento de algoritmos computacionais baseados no funcionamento do cérebro humano, mais especificamente dos neurônios: os autores criaram um modelo computacional chamado "Lógica do Limiar", do inglês: *threshold logic*. Em 1958 a lógica do limiar é aprimorada com a criação do conceito do perceptron[32] um neurônio matemático baseado em uma rede neural computacional de duas camadas. Este modelo abriu caminho para abordagens focadas na aplicação de RNA's em IA.

Nas décadas seguintes diversos algoritmos baseados em RNA's foram propostos[33,34], mas a escassez de dados e o alto custo computacional foram cruciais na lenta evolução desta área. Em 1986, um artigo publicado na revista Nature[35] revolucionou o treinamento de redes neurais com a técnica *"back-propagation"*. Esta técnica consiste em propagar o erro no sentido contrário do fluxo de dados na rede, quantificar a influência de cada neurônio no erro total e atualizar os pesos de modo a diminuir este erro, atribuindo pesos mais altos a neurônios com menores erros.

Esta técnica reacendeu o interesse de pesquisadores pelo tema e já em 1989 tem-se a publicação da primeira "Rede Neural Profunda" (DNN, do inglês: *Deep Neural Network*) que reconhecia dígitos escritos a mão[36]. Em 1992 pesquisadores publicam a Cresceptron[37], uma rede que reconhece objetos 3-D automaticamente. Em 1995 surgem as "Máquinas de Vetor de Suporte" (SVM's, do inglês: *Support Vector Machines*)[38], algoritmos de reconhecimento e mapeamento não supervisionado de dados. Em 1997 publica-se a tradicional "Memória Longa de Curto Prazo" (LSTM, do inglês: *Long short-term memory*)[39] a principal rede neural recorrente que até hoje é a base utilizada em softwares de reconhecimento de voz em smartphones.[40] Em 1998 Yan LeCun, o criador da técnica back-propagation, publica uma nova técnica chamada "Aprendizado Baseado em Gradiente" (GBL, do inglês: *Gradient-based Lerning*)[41], que consiste em um algoritmo estocástico que utiliza o gradiente descendente aliado ao back-propagation para agilizar e refinar o treinamento de DNN's.

Apesar do rápido desenvolvimento de algoritmos baseados em redes neurais em meados da década de 1980, a capacidade de processamento dos computadores ainda avançava lentamente e por volta do ano 2000 ainda era muito baixa: os tempos de treinamento eram contados em dias ou semanas inviabilizando a utilização dos algoritmos de forma aplicada. Foi a partir do desenvolvimento e melhoramento de hardwares como GPU's e a ascensão da computação de alto desempenho na década seguinte, que os algoritmos de DL baseados em redes neurais começam a fazer parte do nosso dia a dia.

Em 2009 foi criada a ImageNet[42] um banco de dados com mais de 14 milhões de imagens rotuladas, disponíveis para pesquisadores, professores e estudantes.

Em 2012 foi criado um desafio[43] para desenvolvimento de algoritmos de classificação de imagens baseados na ImageNet. Vários algoritmos foram desenvolvidos, como: a AlexNet[44], as VGG's[45] e as ResNet's[46], alguns deles com margem de erro menor do que de seres humanos. Uma análise realizada em 2017[47] definiu as ResNet's como o estado da arte do reconhecimento de imagens devido a sua maior acurácia e densidade de acurácia dentre todas as redes observadas.

# Redes Neurais Artificiais

As redes neurais artificiais (RNA) são algoritmos inspirados no cérebro humano, que utilizam neurônios e suas interconexões. Esses algoritmos podem apresentar perceptrons (neurônios artificiais) e as redes com várias camadas de neurônios são consideradas aproximadores universais, como as Multilayer Perceptrons[48]. Dentre os diversos tipos de RNA's, será dado ênfase as Redes Neurais Diretas (do inglês: feedforward network), que possuem um fluxo de informação em apenas uma direção: da camada inicial até a camada final, como representado na Figura 2. Neste tipo de rede não há loops ou ciclos, como acontece nas redes recorrentes (como a LSTM[39]) que necessitam da retroalimentação. Uma RNA direta pode ser definida como um conjunto de neurônios de entrada, um conjunto de neurônios ocultos e um conjunto de neurônios de saída (Figura 2).

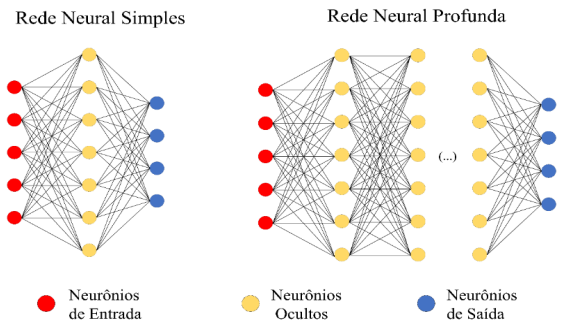


**Figura 2.** Representações genéricas de RNA's diretas

Dentre as diversas tarefas que uma RNA pode desempenhar, daremos destaque para as redes de regressão (que fazem inferência de um valor pretendido de acordo com os dados de entrada) e classificação (que identificam presença ou ausência de características nos dados de entrada).

## NEURÔNIO MATEMÁTICO

Um neurônio matemático, também conhecido como *perceptron* artificial, é um componente que recebe um ou mais sinais de entrada e retorna um sinal de saída único, como representado na Figura 3.

Os sinais $x_n$ recebidos por um neurônio matemático são análogos à estímulos recebidos por um neurônio biológico. Dentre esses estímulos, alguns irão causar maior ou menor excitação do neurônio receptor, essa medida de excitação é representada pelos pesos sinápticos $w_n$. Quanto maior o peso, mais excitatório é o estímulo. Quando os sinais de entrada $x_n$ chegam nos neurônios, são multiplicados pelos pesos sinápticos $w_n$ correspondentes, e então é feita uma soma ponderada. Um valor de polarização chamado bias ($b$) é geralmente incluído ao somatório com o intuito de aumentar o grau de liberdade da função e consequentemente a capacidade de aproximação (aprendizagem) da rede. Ao final deste processo um sinal de saída ($u$) é gerado. Este sinal de saída é enviado para a função de ativação $f(u)$ gerando o sinal de saída do neurônio ($y$)[49]. O processo total é descrito na Equação (1).

$$y = f(u) = f\left(\sum (w_n * x_n) + b\right) \quad (1)$$

*Perceptrons* quando associados em camadas em sequência dão origem as RNA's e podem descrever diversos tipos de comportamento utilizando um modelo de aproximação universal, análogo a uma soma de polinômios não lineares.

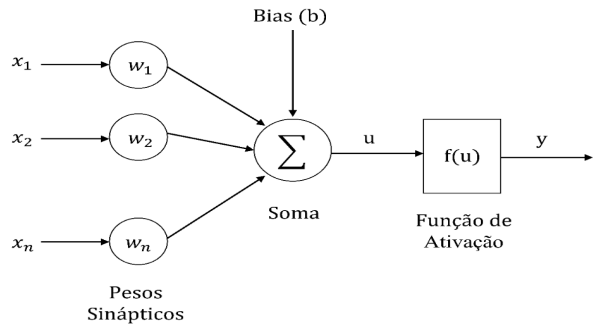


**Figura 3.** Representação de um neurônio matemático. $x_n$ são os sinais de entrada, $w_n$ são os pesos, ($u$) é o resultado da combinação linear, $f(u)$ é a função de ativação e ($y$) é o sinal de saída

## FUNÇÕES DE ATIVAÇÃO

Para que uma RNA seja um aproximador universal ela deve se adequar aos mais variados tipos de comportamentos. Se os neurônios realizassem apenas combinações lineares $u = [\sum(w_n * x_n) + b]$ a rede seria uma sequência de combinações lineares, e, portanto, também seria uma combinação linear. É para isto que existem as chamadas funções de ativação. Além de permitir a descrição de comportamentos não lineares, as funções de ativação diminuem o impacto causado por cada peso sináptico na saída final da rede, refinando a excitação dos neurônios e melhorando a capacidade de aproximação ou aprendizagem da RNA.

As funções de ativação são de extrema importância nas RNA's, pois elas basicamente decidem se um neurônio deve ser ativado ou não, ou seja, se a informação fornecida é relevante para o neurônio em questão ou deve ser ignorada. Além disso, são funções diferenciáveis e assim simplificam a determinação do erro pelo back-propagation ao possibilitarem a determinação de seu gradiente para cada peso, contribuindo para o refino do treinamento da rede.

A função de ativação mais simples é a função limiar, seu comportamento é descrito na Figura 4. Para esta função, o valor da saída y do neurônio será igual à 1 (ou ativado) se o valor de entrada u for igual ou maior que uma dada constante a, e será igual à 0 (ou não ativado) caso a entrada seja menor que esta mesma constante. A função limiar basicamente decide se um sinal é relevante ou deve ser ignorado, ou seja, se aquele neurônio foi ativado, ou não.

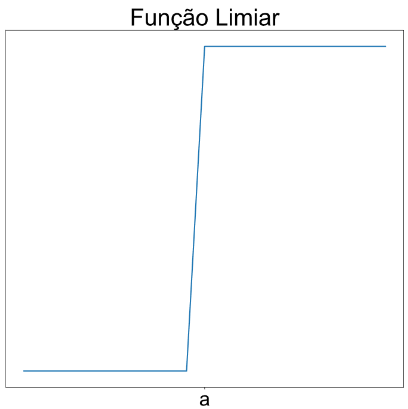


**Figura 4.** Função Limiar

Atualmente as funções de ativação mais utilizadas em RNA's são aquelas que retornam uma probabilidade de ativação: ao invés de retornar os valores 0 ou 1, retornam valores reais dentro de um intervalo, representando a relevância do sinal de entrada u ou a probabilidade de ativação. Dentre as mais comuns podemos citar as funções: linear; ReLU, sigmóide; e tangente hiperbólica, descritas nas Equações (2), (3), (4) e (5), respectivamente. Na Equação (5), p é um parâmetro arbitrário. Os comportamentos destas funções estão expressos na Figura 5.

$$f_{lin}(u) = au \quad (2)$$

$$f_{ReLU}(u) = \begin{cases} 0, se\ u < a \\ u, se\ u \geq a \end{cases} \quad (3)$$

$$f_{sig}(u) = \frac{1}{1 + e^{-u}} \quad (4)$$

$$f_{tanh}(u) = \frac{e^{pu} - e^{-pu}}{e^{pu} + e^{-pu}} \quad (5)$$

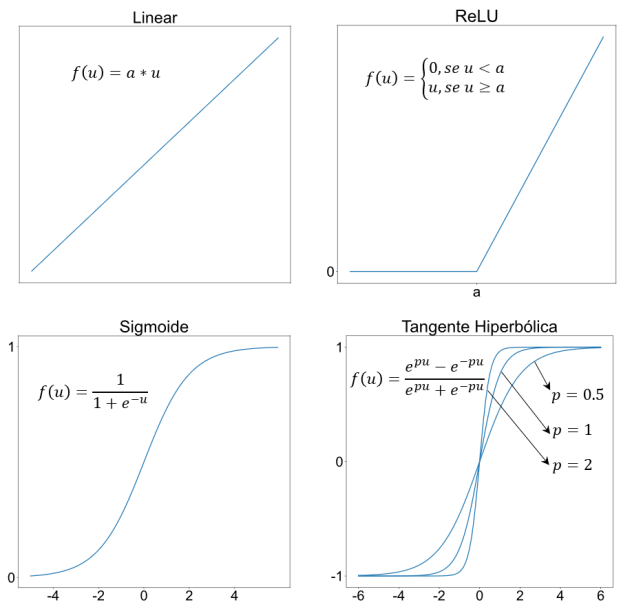


**Figura 5.** Comportamento das funções de ativação: linear, ReLU, sigmoide e tangente hiperbólica

# Treinamento e Hiper Parametrização

Um passo importante no desenvolvimento de RNA's é o treinamento, que consiste em modelar o comportamento do sinal de saída de uma rede para que possa gerar ou classificar informações. Para isto é preciso calibrar os pesos sinápticos . Neste processo os pesos são inicialmente aleatorizados ou herdados de uma outra rede já treinada para uma tarefa similar (transfer learning). A cada "lote" de dados que flui sobre a rede, os pesos  e o bias  (Equação (1) são atualizados visando minimizar o erro. Este processo de atualização dos pesos é realizado por otimizadores.

Os otimizadores são algoritmos utilizados para ajustar os pesos da rede durante o processo de treinamento e cada um possui seu próprio método para minimizar a função de erro. O Gradiente Descendente Estocástico (SGD) é um dos otimizadores mais simples, mas ainda amplamente utilizado, onde os pesos são atualizados em pequenos incrementos seguindo a direção oposta do gradiente da função de erro. O Adam[50] é outro otimizador comumente utilizado que adapta a taxa de aprendizagem para cada peso individualmente. Além desses, existem muitos outros otimizadores que utilizam técnicas como momentum[51], Nesterov accelerated gradient[52], Adagrad[53], RMSProp[54], GSA[5] entre outros. A escolha do otimizador depende do problema e da arquitetura da rede em questão, por exemplo, a técnica RMSProp é bastante utilizada em redes neurais para melhorar o desempenho em tarefas de reconhecimento de fala, enquanto o otimizador Adagrad é aplicado em tarefas de processamento de imagem para melhorar a precisão de detecção de objetos.

A função  é uma função erro bastante comum usada em problemas de regressão. Para problemas de classificação comumente se usa a função entropia cruzada.[55] Ao treinar uma rede, escolher corretamente os otimizadores e a função erro são de grande importância, mas sozinhos não garantem sucesso do treinamento. Existem uma série de parâmetros que devem ser ajustados ao problema em questão, como: a taxa de aprendizagem (do inglês: learning rate, é a grandeza com que os pesos serão atualizados: se for muito baixa serão necessárias demasiadas etapas de treinamento; se a for muito alta os pesos irão se alterar muito bruscamente); o número de camadas do modelo; a quantidade de neurônios em cada camada; e o números de ciclos de treinamento. Quanto mais complexo o problema, mais características os dados apresentam e um maior número de camadas e neurônios serão necessários.

Dois problemas mais comuns no treinamento de uma rede neural são o overfitting e o underfitting. O overfitting é um problema que ocorre quando o modelo se ajusta muito bem aos dados de treinamento, mas tem uma baixa capacidade de generalização para novos dados. Isso acontece quando a complexidade do modelo é muito alta para o tamanho do conjunto de dados de treinamento ou quando há um desequilíbrio entre o tamanho do conjunto de treinamento e o número de parâmetros do modelo. Já o underfitting é o oposto, ocorre quando o modelo é

muito simples para capturar a complexidade dos dados de treinamento, resultando em uma baixa capacidade de ajuste aos dados de treinamento e a novos dados. É importante encontrar um equilíbrio entre a complexidade do modelo e o tamanho do conjunto de treinamento para evitar tanto o overfitting quanto o underfitting.

# Classificadores Multi Classe e Multi Rótulo

Um algoritmo de classificação multi rótulo (do inglês: Multilabel Classification) é capaz de determinar à quais classes uma instância pertence, diferente de algoritmos de classificação binária multi classe, onde o objetivo é determinar à qual classe (única) aquela instância pertence. Nestes classificadores geralmente o sinal de saída da rede é um vetor de probabilidades (valores entre 0 e 1). Cada posição do vetor representa uma classe, enquanto o valor é a probabilidade de a classe estar presente na instância. Em classificação binária multi classe, a classe com maior probabilidade será o resultado da predição; em classificação multi rótulo é necessário um critério para definir quais classes estão presentes na instância, como por exemplo a determinação de limiares.

Essas técnicas são amplamente utilizadas em áreas como medicina, processamento de linguagem natural e reconhecimento de imagem. Um exemplo prático seria a aplicação em pré diagnóstico médico, onde um classificador multi rótulo pode ser utilizado para identificar múltiplas condições médicas em um paciente com base em diferentes sinais clínicos.

# Métricas de Validação

Para avaliar a eficiência de um modelo de classificação multi rótulo utiliza-se funções estatísticas como: acurácia [, Equação (6)], *recall* [, Equação (7)], precisão [, Equação (8)], especificidade [, Equação (9)] e pontuação F1 [, Equação (10)][56]. Cada uma dessas métricas avalia uma característica diferente da rede, iremos discutir todas elas nesta seção. Quando um modelo prediz corretamente uma classe contida na instância entende-se que ocorreu um Positivo Verdadeiro (, do inglês: *True Positive*) enquanto a predição de uma classe que não está contida na instância representa um Falso Positivo (, do inglês: False Positive). De forma análoga, quando uma classe presente na instância não é predita pelo modelo, tem-se um Falso Negativo (, do inglês: *False Negative*), enquanto a não detecção de uma classe que de fato não existe na instância representa um Negativo Verdadeiro (, do inglês: *True Negative*). Munidos destes conceitos iremos definir as métricas citadas anteriormente.[56] Essas métricas são calculadas item a item, classe por classe: a média das classes representa a métrica total da rede.

A acurácia, Equação (6), é a proporção das predições verdadeiras em relação ao total de predições realizadas. Geralmente não é suficiente para avaliar a eficiência de um classificador multi rótulo, por exemplo: considere um conjunto de dados em que 10% dos itens apresentem uma certa classe, se utilizarmos uma rede enviesada que não prevê esta classe nunca, ainda sim esta rede teria uma acurácia de 90%.

O recall, Equação (7), representa a proporção de acertos de previsão realizadas (TP) em todas as instâncias que apresentam uma certa classe (TP+FN); enquanto a precisão, Equação (8), é a proporção de predições corretas (TP) dentre todas as vezes que uma classe foi predita (TP+FN): considere um conjunto de dados com 200 itens, onde 100 itens apresentem uma classe específica; e considere um algoritmo que detecta corretamente esta classe em 80 dos itens que realmente apresentam esta classe; ao mesmo tempo, o algoritmo detectou esta classe em outras 100 instâncias que originalmente não possuem a classe em questão. Neste caso, temos: temos: TP=80, FP=100, TN=20 e FN=0. O recall neste caso será 80%, enquanto a precisão será 44%. A especificidade, Equação (9), é análoga a precisão, mas avalia a proporção com que a rede identifica corretamente a ausência de certa classe, seu valor neste caso é de 16,66%. TP

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \quad (6)$$

$$Rec = \frac{TP}{TP + FN} \quad (7)$$

$$Prec = \frac{TP}{TP + FP} \quad (8)$$

$$Spec = \frac{TN}{TN + FP} \quad (9)$$

$$F1 - Score = \frac{2 * Prec * Rec}{Prec + Rec} \quad (10)$$

A importância de cada métrica depende do modelo e do problema abordado, por exemplo: qual o mais perigoso para uma rede treinada para identificação de câncer em pacientes: identificar um câncer não existente (*FP*), ou deixar de identificar um câncer existente (*FN*)? A resposta correta é provavelmente a segunda e neste caso devemos garantir uma boa medida para o *recall* desta rede; mas se a resposta correta fosse a primeira, deveríamos garantir que nossa rede tenha uma boa precisão e especificidade. Além disso, no geral quanto melhor o *recall*, precisão e especificidade, melhor tende a ser a acurácia.

A última e mais importante métrica dos classificadores é a *F1-Score*: uma relação entre *recall* a precisão. Seu valor máximo é 1 ou 100%, indicando *recall* e precisões perfeitos. Em suma, bons valores de *F1-Score* indicam que uma rede está falhando pouco em todos os quesitos. O *F1-Score* é geralmente utilizado para comparação de desempenhos entre redes. Entretanto, é necessário pontuar que apenas essa métrica não é suficiente para indicar a eficiência de uma rede, sendo necessário análises de várias métricas em um contexto geral para se chegar a esta conclusão.

# Redes Neurais Convolucionais

Uma Rede Neural Convolucional (CNN, do inglês: *Convolutional Neural Network*) é um RNA profunda que utiliza dados matriciais (que podem apresentar diferentes tamanhos) de entrada, atribuindo importância a aspectos destas matrizes, sendo, portanto, capaz de diferenciá-los. Imagens são matrizes de pixels que podem apresentar 1 ou mais canais de cores (3 canais no caso RGB, por exemplo) e foram o foco principal no desenvolvimento deste tipo de rede. As CNN's se diferem das RNA's convencionais devido a suas camadas iniciais de pré-processamento da imagem, que precedem as camadas de neurônios. Em métodos primitivos de IA o pré-processamento das imagens era algo complexo, pois os filtros utilizados para extrair características das imagens eram feitos a mão para cada conjunto de imagens de entrada. Já as CNN's, com treinamento suficiente são capazes de aprender esses filtros.

Dentre as operações de pré-processamento da imagem temos: o pooling (operação responsável por diminuir a dimensionalidade da imagem, camadas de pooling são essenciais para otimizar o tempo de treinamento da rede); achatamento ou flating (geralmente realizada pela camada final de pré-processamento da imagem onde as matrizes da imagem são "achatadas" e transformadas em vetores unitários que são enviados como sinais de entrada para primeira camada de neurônios); e as convoluções. Esta última é a operação mais importante, responsável pela extração de características na imagem, e será abordada com mais detalhes.

A operação de convolução é a aplicação de um filtro em uma imagem, sendo, portanto, responsável por extrair as características dessa imagem dando à rede a capacidade de diferenciar e classificar essas características. Matematicamente a operação de convolução é uma sequência de produtos escalares de matrizes, na qual a matriz do filtro utilizado (de tamanhos variados 3x3, 4x4 etc.) varre a matriz de pixels da imagem. A média do produto escalar para cada superposição entre o filtro e uma região da imagem da origem a um novo pixel da imagem gerada. Desta forma, cada filtro utilizado gera uma nova imagem, diferentes entre si, e cada uma dessas imagens resultantes carrega consigo uma característica da imagem inicial.[44,57]

Diferentes combinações entre camadas de pré-processamento e de neurônios dão origem a diversas arquiteturas avançadas de CNN's com diferentes capacidades

de aprendizagem. As redes VGG45, AlexNet44 e ResNet46 são exemplos de CNN's. Atualmente as CNN's são o estado da arte dos algoritmos de D L para processamento e classificação de imagens, como reconhecimento de objetos e detecção de anomalias.

# Codificação de Moléculas para Uso em ML e DL

Atualmente boa parte das pesquisas em química se dedicam a projetar novas moléculas com propriedades específicas, e, embora haja diversos trabalhos que visam novas estratégias de design, ainda é muito comum a utilização de métodos aleatórios para desenvolver moléculas de interesse. Um dos maiores objetivos da utilização de ML e DL na química é substituir estes métodos aleatórios por uma busca direcionada, no qual modelos de predição possam ser utilizados para prever propriedades moleculares e assim aumentar a eficiência no desenvolvimento de novos materiais e técnicas de análise.

Para isso, o primeiro passo é construir métodos que transformem moléculas em vetores numéricos que possam ser passados para os algoritmos de ML e DL, chamados "caracterizadores moleculares" (do inglês, *molecular featurizations*). Moléculas são entidades complexas, por isso existem diversas técnicas para caracterizá-las, incluindo vetores de descritores químicos, representações de gráficos 2D, representações 3D, representações de funções de base orbital e muito mais. Uma vez caracterizada, uma molécula pode ser "aprendida" por um modelo. Existem diversos tipos de descritores moleculares e alguns deles serão abordados detalhadamente a seguir.

# Extended-Connectivity Fingerprints (ECFPs)

Impressões digitais químicas (do inglês, molecular fingerprints) são vetores contendo valores de 0 e 1 que representam a presença ou ausência de características específicas em uma molécula (Figura 6). Foram inicialmente desenvolvidas para facilitar a busca de estruturas em bancos de dados e atualmente são utilizadas em tarefas de ML para busca de similaridade e clusterização.[58]

ECFPs é uma metodologia mais recente[59] que visa descrever características moleculares relevantes para atividade química e é utilizada bastante em modelagem molecular. Podem ser calculadas rapidamente e transformam uma molécula de tamanho arbitrário em vetores de tamanho fixo, facilitando a comparação de similaridade entre moléculas uma vez que só é preciso comparar os elementos correspondentes dos vetores. Cada elemento do vetor representa a presença ou ausência de uma característica específica definida por um arranjo local de átomo, considerando propriedades atômicas como: elemento atômico e número de ligações covalentes.


**Figura 6.** Exemplo de um vetor ECFP

## MATRIZ DE COULOMB

Matriz de *Coulomb* é um caracterizador simples que descreve as interações eletrostáticas entre os átomos de uma molécula ao codificar cargas nucleares e coordenadas cartesianas correspondentes em uma matriz. A regra de formação para a matriz de *Coulomb* obedece a Equação (11):

$$M_{ij}^{Coulomb} = \begin{cases} 0.5Z_i^{2.4} & for\ i = j \\ \dfrac{Z_i Z_j}{R_{ij}} & for\ i \neq j \end{cases} \qquad (11)$$

na qual $i$ e $j$ são índices atômicos; $Z_i$ e $Z_j$ são cargas nucleares e $R_{ij}$ é a distância entre os átomos $i$ e $j$. Trabalhos recentes têm utilizado este caracterizador para predição de energia molecular.[60,61]

## GRID FEATURIZER

O Grid Featurizer é uma ferramenta poderosa para resumir as forças intermoleculares em pares proteína-ligante com base em suas estruturas detalhadas. Este descritor

incorpora impressões digitais de ambos os componentes da interação e fornece uma enumeração de interações como pontes salinas e ligações de hidrogênio. Além disso, o Grid Featurizer é altamente personalizável e permite a seleção de parâmetros específicos, tornando-o uma ferramenta valiosa para a análise de complexos proteína-ligante em estudos de docking virtual e design de fármacos. Sua robustez e eficácia foram comprovadas em várias aplicações em diferentes conjuntos de dados[62,63], tornando-o uma ferramenta promissora para a análise de interações proteína-ligante em diversos contextos biológicos e farmacêuticos.

**FUNÇÕES DE SIMETRIA**

Funções de simetria são descritores moleculares que se baseiam nas coordenadas cartesianas para preservar a simetria rotacional e de permutação do sistema. Elas podem introduzir uma série de funções de simetria radial e angular com diferentes limites de distância e ângulo. As funções radiais são usadas para capturar informações sobre a distribuição de átomos em torno de um átomo central, enquanto as funções angulares descrevem a relação entre átomos adjacentes. Além disso, os limites de distância e ângulo são ajustáveis, o que permite que o descritor seja personalizado para diferentes sistemas. As funções de simetria são particularmente úteis em sistemas com simetria esférica ou axial, onde a aplicação de outros descritores pode ser difícil. A eficácia das funções de simetria foi comprovada em várias aplicações[64,65] e são uma ferramenta valiosa para a descrição de sistemas moleculares simétricos em uma variedade de aplicações químicas e biológicas.

# Considerações Finais

A inteligência artificial é um campo em constante evolução, e o DL é uma das suas áreas mais promissoras. Com base em RNA's, essa técnica permite que as máquinas aprendam a partir de grandes quantidades de dados, sem a necessidade de uma programação específica. As RNA's foram desenvolvidas há décadas, mas somente com o aumento do poder de processamento e o surgimento de grandes bases de dados é que o DL se tornou viável. Ao utilizar algoritmos de aprendizagem profunda, essa técnica consegue identificar padrões em dados complexos, como imagens, áudio e texto, e realizar tarefas que antes só podiam ser executadas por seres humanos.

O interesse pelo DL na pesquisa em química tem crescido a cada ano, contribuindo no design de moléculas com propriedades específicas, onde modelos de predição podem ser utilizados para prever propriedades moleculares e aumentar a eficiência no desenvolvimento de novos materiais e técnicas de análise autônomas[66–69]. Para alcançar este objetivo, é essencial a compreensão acerca de caracterizadores moleculares, para que os dados possam ser lidos e aprendidos pelos algoritmos de ML e DL. A escolha correta do descritor molecular é uma etapa fundamental.

Apesar dos avanços, ainda há muito a ser explorado no campo do DL. Novas arquiteturas de redes neurais estão em constante desenvolvimento, como as Redes Neurais Adversárias Generativas[70], que são capazes de gerar imagens e vídeos de forma impressionante; e redes de processamento de linguagem natural como o GPT-3[71], capaz de conversar, agir e responder perguntas como se fosse um ser humano, também de forma impressionante. Além disso, é preciso considerar questões éticas e sociais relacionadas ao uso do DL, como a privacidade de dados e o impacto na empregabilidade. Porém, com uma abordagem responsável e ética, o DL pode revolucionar a maneira como interagimos com a tecnologia e como resolvemos problemas complexos em diversas áreas.

Em resumo, o DL é uma técnica que está mudando a forma como a inteligência artificial é aplicada e tem um enorme potencial para continuar transformando diversas áreas da sociedade. É importante investir em pesquisas e estudos nessa área, a fim de avançar ainda mais e desenvolver soluções cada vez mais sofisticadas e úteis.

# Referências


1. Aires-de-Sousa J, Hemmer MC, Gasteiger J. Prediction of 1H NMR chemical shifts using neural networks. Anal Chem. **2002**;74(1):80-90. doi:10.1021/ac010737m
2. Zupan J, Gasteiger J. Neural Networks in Chemistry and Drug Design. Published online **1999**:400. doi:3-527-29779-0
3. Kwon S, Bae H, Jo J, Yoon S. Comprehensive ensemble in QSAR prediction for drug discovery. BMC Bioinformatics. **2019**;20(1):1-12. doi:10.1186/s12859-019-3135-4
4. Balaban AT. Neural Networks in QSAR and Drug Design. J Chem Inf Comput Sci. **1997**;37(3).

5. Mundim KC, Tsallis C. Geometry optimization and conformational analysis through generalized simulated annealing. Int J Quantum Chem. **1996**;58(4):373-381. doi:10.1002/(sici)1097-461x(1996)58:4<373::aid-qua6>3.0.co;2-v

6. Behler J. Neural network potential-energy surfaces in chemistry: A tool for large-scale simulations. Physical Chemistry Chemical Physics. **2011**;13(40):17930-17955. doi:10.1039/c1cp21668f

7. Lecun Y, Bengio Y, Hinton G. Deep learning. Nature. **2015**;521(7553):436-444. doi:10.1038/nature14539

8. Adeel A, Gogate M, Hussain A. Contextual deep learning-based audio-visual switching for speech enhancement in real-world environments. Information Fusion. **2020**;59:163-170. doi:https://doi.org/10.1016/j.inffus.2019.08.008

9. Tian H, Chen SC, Shyu ML. Evolutionary Programming Based Deep Learning Feature Selection and Network Construction for Visual Data Classification. Information Systems Frontiers. **2020**;22(5):1053-1066. doi:10.1007/s10796-020-10023-6

10. Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing [Review Article]. IEEE Comput Intell Mag. **2018**;13(3):55-75. doi:10.1109/MCI.2018.2840738

11. Koppe G, Meyer-Lindenberg A, Durstewitz D. Deep learning for small and big data in psychiatry. Neuropsychopharmacology. **2021**;46(1):176-190. doi:10.1038/s41386-020-0767-z

12. Ramsundar B, Riley P, Webster D, Konerding D, Edu KS, Edu PS. Massively Multitask Networks for Drug Discovery. **2015**;(Icml).

13. Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V. Deep neural nets as a method for quantitative structure-activity relationships. J Chem Inf Model. **2015**;55(2):263-274. doi:10.1021/ci500747n

14. Hamza H, Salim N, Nasser M, Saeed F. AtomNet: A Deep Learning Neural Network for Bioactivity Prediction in Structure-based Drug Discovery. Published online **2020**:21-37. doi:10.5121/csit.**2020**.100203

15. Lusci A, Pollastri G, Baldi P. Deep architectures and deep learning in chemoinformatics: The prediction of aqueous solubility for drug-like molecules. J Chem Inf Model. **2013**;53(7):1563-1575. doi:10.1021/ci400187y

16. Mobley DL, Wymer KL, Lim NM, Guthrie JP. Blind prediction of solvation free energies from the SAMPL4 challenge. J Comput Aided Mol Des. **2014**;28(3):135-150. doi:10.1007/s10822-014-9718-2

17. Delaney JS. ESOL: Estimating aqueous solubility directly from molecular structure. J Chem Inf Comput Sci. **2004**;44(3):1000-1005. doi:10.1021/ci034243x

18. Mobley DL, Guthrie JP. FreeSolv: A database of experimental and calculated hydration free energies, with input files. J Comput Aided Mol Des. **2014**;28(7):711-720. doi:10.1007/s10822-014-9747-x

19. Rupp M, Tkatchenko A, Müller KR, Von Lilienfeld OA. Fast and accurate modeling of molecular atomization energies with machine learning. Phys Rev Lett. **2012**;108(5):1-5. doi:10.1103/PhysRevLett.108.058301

20. Montavon G, Rupp M, Gobre V, et al. Machine learning of molecular electronic properties in chemical compound space. New J Phys. **2013**;15:0-16. doi:10.1088/1367-2630/15/9/095003

21. McGibbon RT, Taube AG, Donchev AG, et al. Improving the accuracy of Møller-Plesset perturbation theory with neural networks. Journal of Chemical Physics. **2017**;147(16). doi:10.1063/1.4986081

22. Schütt KT, Arbabzadah F, Chmiela S, Müller KR, Tkatchenko A. Quantum-chemical insights from deep tensor neural networks. Nat Commun. **2017**;8(0):1-21. doi:10.1038/ncomms13890

23. Blum LC, Reymond JL. 970 Million druglike small molecules for virtual screening in the chemical universe database GDB-13. J Am Chem Soc. **2009**;131(25):8732-8733. doi:10.1021/ja902302h

24. Gaulton A, Kale N, Van Westen GJP, et al. A large-scale crop protection bioassay data set. Sci Data. **2015**;2. doi:10.1038/sdata.**2015**.32

25. Wang Y, Xiao J, Suzek TO, et al. PubChem's BioAssay database. Nucleic Acids Res. **2012**;40(D1). doi:10.1093/nar/gkr1132

26. Goos G, Hartmanis J, Van J, et al. LNCS 5342 - Structural, Syntactic, and Statistical Pattern Recognition.; **2008**.

27. Wang R, Fang X, Lu Y, Yang CY, Wang S. The PDBbind database: Methodologies and updates. J Med Chem. **2005**;48(12):4111-4119. doi:10.1021/jm048957q

28. Huang R, Xia M, Nguyen DT, et al. Tox21 challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs. Front Environ Sci. **2016**;3(JAN). doi:10.3389/fenvs.**2015**.00085

29. Mayr A, Klambauer G, Unterthiner T, Hochreiter S. DeepTox: Toxicity prediction using deep learning. Front Environ Sci. **2016**;3(FEB). doi:10.3389/fenvs.**2015**.00080

30. Kuhn M, Letunic I, Jensen LJ, Bork P. The SIDER database of drugs and side effects. Nucleic Acids Res. **2016**;44(D1):D1075-D1079. doi:10.1093/nar/gkv1075

31. Mcculloch WS, Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics. **1943**;5:115-133. doi:10.1007/bf02478259

32. Rosenblatt F. The Perceptron - A Perceiving and Recognizing Automaton. Report 85, Cornell Aeronautical Laboratory. Published online **1957**:460-461.

33. Minsky M. A Neural-Analogue Calculator Based upon a Probability Model of Reinforcement. Harvard University Psychological Laboratories,. Published online **1952**.

34. Widrow B. An Adaptive “Adaline” Neuron Using Chemical “Memistors.” Stanford Electronics Laboratories Technical Report. Published online **1960**:1553-2.

35. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. **1986**;323(6088):533-536. doi:10.1038/323533a0

36. LeCun Y, Boser B, Denker JS, et al. Handwritten Digit Recognition with a Back-Propagation Network. AT&T Bell Laboratories. **1989**;(07733):396-404.

37. Weng J, Ahuja N, Huang TS. Cresceptron: A Self-Organizing Neural Network Which Grows Adaptively. RN. **1992**;63(2):576-581. doi:10.1109/IJCNN.1992.287150

38. Hearst MartiA, Scholkopf Bernhard, Dumais Susan, Osuna Edgar, Platt J. Supprot vector machines. IEEE Intelligent Systems and their Applications. **1998**;13(4):18-28.

39. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. **1997**;(9):1735–1780. doi:10.1162/neco.1997.9.8.1735

40. Graves A, Mohamed A rahman, Hinton G. Speech Recognition With Deep Recurrent Neural Networks. IEEE. **2013**;(3). doi:https://doi.org/10.1109/ICASSP.2013.6638947

41. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based Learning Applied to Document Recognition. proc OF THE IEEE. Published online **1998**. http://ieeexplore.ieee.org/document/726791/#full-text-section

42. Fei-Fei L, Deng J, Li K. ImageNet: A Large-Scale Hierarchical Image Database. Journal of Vision - IEEE. **2009**;9(8):1037-1037. doi:https://doi.org/10.1109/CVPR.2009.5206848

43. Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge. Published online **2015**.

44. Krizhevsky A, Sutsekever I, Hinton GE. ImageNet Classification with Deep Convolution Neural Networks. Adv Neural Inf Process Syst. **2012**;60(6):84-90. doi:http://dx.doi.org/10.1145/3065386

45. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. Published online **2015**:1-14.

46. Targ S, Almeida D, Lyman K. Resnet in Resnet: Generalizing Residual Architectures. Published online **2016**:1-7. http://arxiv.org/abs/1603.08029

47. Canziani A, Culurciello E, Paszke A. An Analysys of Deep Neural Netowrk Models for Practical Applications. Published online **2017**:1-7.

48. Ramsundar B, Eastman P, Walters P, Pande V. Deep Learning for the Life Science. 1st ed. (Tache N, Loukides M, Tozer K, Head R, eds.). O’Reilly Media; **2019**.

49. Simon S. Haykin. Neural Networks and Learning Machines. Vol 10. Prentice Hall; **2009**.

50. Kingma DP, Ba JL. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. ; **2015**:1-15.

51. Sutskever I, Martens J, Dahl G, Hinton G. On the importance of initialization and momentum in deep learning. In: Dasgupta S, McAllester D, eds. Proceedings of the 30th International Conference on Machine Learning. PMLR; **2013**:1139-1147.

52. Nesterov Y. A method for solving the convex programming problem with convergence rate O(1/k^2). Proceedings of the USSR Academy of Sciences. **1983**;269:543-547.

53. Duchi J, Hazan E, Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research. **2011**;12(61):2121-2159.

54. Tieleman T, Hinton G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning. **2012**;4(2):26-31.

55. Mannor S, Peleg D, Rubinstein R. The Cross Entropy Method for Classification. In: Proceedings of the 22nd International Conference on Machine Learning. Association for Computing Machinery; **2005**:561–568.

56. Goutte C, Gaussier E. Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In: Proceedings of the 27th European Conference on Advances in Information Retrieval Research. Springer-Verlag; **2005**:345–359.

57. Henaff M, Bruna J, LeCun Y. Deep Convolutional Networks on Graph-Structured Data. Published online **2015**:1-10. http://arxiv.org/abs/1506.05163

58. Mcgregor MJ, Pallai P v. Clustering of Large Databases of Compounds: Using the MDL “Keys” as Structural Descriptors. J Chem Inf Comput Sci. **1997**;37(3):443-448. doi:https://doi.org/10.1021/ci960151e

59. Rogers D, Hahn M. Extended-connectivity fingerprints. J Chem Inf Model. **2010**;50(5):742-754. doi:10.1021/ci100050t

60. Hazra R, Hazra P. Prediction of molecular energy using Coulomb matrix and Graph Neural Network. J Emerg Investig. **2022**;5.

61. Elton DC, Boukouvalas Z, Butrico MS, Fuge MD, Chung PW. Applying machine learning techniques to predict the properties of energetic materials. Sci Rep. **2018**;8(1). doi:10.1038/s41598-018-27344-x

62. Pandey M, Radaeva M, Mslati H, et al. Ligand Binding Prediction Using Protein Structure Graphs and Residual Graph Attention Networks. Molecules. **2022**;27(16). doi:10.3390/molecules27165114

63. Wu G, Robertson DH, Brooks CL, Vieth M. Detailed analysis of grid-based molecular docking: A case study of CDOCKER - A CHARMm-based MD docking algorithm. J Comput Chem. **2003**;24(13):1549-1562. doi:10.1002/jcc.10306

64. Bartók AP, De S, Poelking C, et al. Machine Learning Unifies the Modeling of Materials and Molecules.; **2017**. doi:https://doi.org/10.1063/1.5126336

65. Zhang L, Han J, Wang H, Saidi WA, Car R. End-to-end Symmetry Preserving Inter-atomic Potential Energy Model for Finite and Extended Systems. In: 32nd Conference on Neural Information Processing Systems. ; **2018**.

66. Lavecchia A. Deep learning in drug discovery: opportunities, challenges and future prospects. Drug Discov Today. **2019**;24(10):2017-2032. doi:10.1016/j.drudis.**2019**.07.006

67. Choudhary K, DeCost B, Chen C, et al. Recent advances and applications of deep learning methods in materials science. NPJ Comput Mater. **2022**;8(1). doi:10.1038/s41524-022-00734-6

68. Chen H, Engkvist O, Wang Y, Olivecrona M, Blaschke T. The rise of deep learning in drug discovery. Drug Discov Today. **2018**;23(6):1241-1250. doi:10.1016/j.drudis.**2018**.01.039

69. Goh GB, Hodas NO, Vishnu A. Deep learning for computational chemistry. J Comput Chem. **2017**;38(16):1291-1307. doi:10.1002/jcc.24764

70. Goodfellow IJ, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Networks. Published online June 10, **2014**. http://arxiv.org/abs/1406.2661

71. Brown TB, Mann B, Ryder N, et al. Language Models are Few-Shot Learners. Published online May 28, **2020**. http://arxiv.org/abs/**2005**.14165

## Hugo G. Machado* & Kleber Mundim

Universidade Federal de Goiás (UFG), Av. Esperança, s/n - Chácaras de Recreio Samambaia, Goiânia - GO, 74690-900

*E-mail: hugogontijomachado@gmail.com